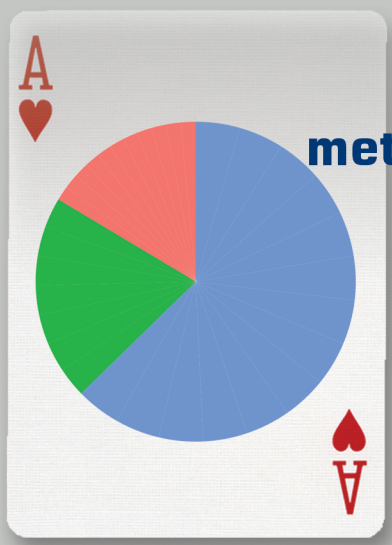
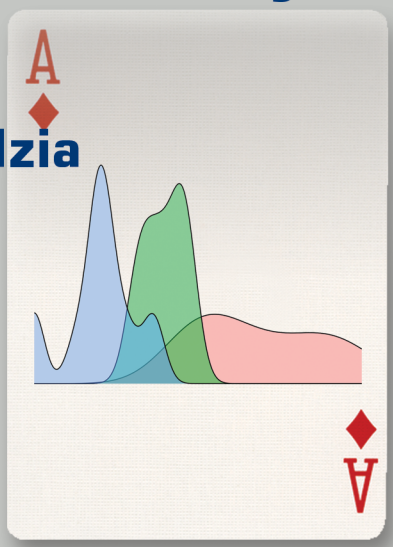
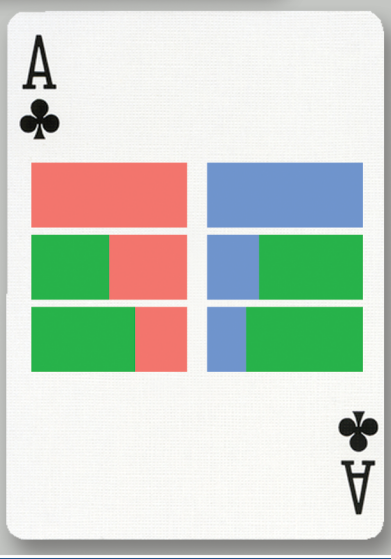
Grzegorz KOŃCZAK

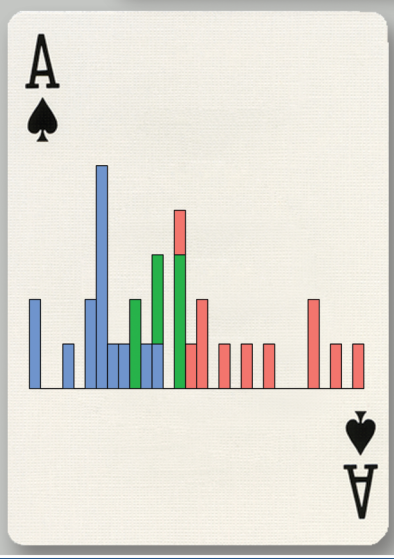
Wizualizacja wyników badań naukowych

Zasady, metody i narzędzia









Wydawnictwo Uniwersytetu Ekonomicznego
w Katowicach

Grzegorz Kończak

Wizualizacja wyników badań naukowych

Zasady, metody i narzędzia

Katowice 2024

Praca naukowa

Komitet redakcyjny

Janina Harasim (przewodnicząca), Monika Ogrodnik (sekretarz), Małgorzata Pańkowska, Jacek Pietrucha, Irena Pyka, Anna Skórska, Maja Szymura-Tyc, Artur Świerczek, Tadeusz Trzaskalik, Ewa Ziemba

Recenzent

Jacek Białek

Redakcja i korekta językowa

Alicja Bronder

Skład tekstu

Marzena Safian

Projekt okładki

Emilia Gumulak

Ilustracje na okładce dostarczone przez Autora

ISBN 978-83-7875-901-0

doi.org/10.22367/uekat.9788378759010

WYDAWNICTWO UNIWERSYTETU EKONOMICZNEGO W KATOWICACH
ul. 1 Maja 50, 40-287 Katowice, tel.: +48 32 257-76-33
http://www.wydawnictwo.ue.katowice.pl, e-mail: www.wydawnictwo.ue.katowice.pl
Facebook: @wydawnictwouekatowice

Spis treści

Jeden **obraz** bywa wart więcej niż tysiąc **słów**.

Przysłowie chińskie

Jeden **wykres** bywa wart więcej niż tysiąc **liczb**.

Parafraza przysłowia

Wprowadzenie

W ostatnich dziesięcioleciach, w wyniku znacznego postępu technicznego, nastąpił dynamiczny rozwój umiejętności wykorzystania komputerów i profesjonalnego oprogramowania w procesie dydaktycznym, badaniach naukowych, a w szczególności w prezentacji wyników takich badań. Początkowo była to możliwość gromadzenia znacznych zbiorów danych, przygotowywania raportów z badań oraz różnorodnych pomocy dydaktycznych. Z czasem rosnące możliwości obliczeniowe komputerów doprowadziły do rozwoju metod graficznej analizy danych. Możliwości uzyskania dobrej jakości kolorowych wydruków pozwoliły na upowszechnienie takiej formy prezentacji wyników badań. Interaktywne prezentacje są bardzo pomocne w przedstawianiu różnych złożonych zagadnień statystycznych, a rozwój technologii mobilnych umożliwia dotarcie z wiedzą statystyczną do szerokiego grona odbiorców. Przeprowadzenie wstępnych analiz graficznych pozwala na wskazanie ścieżek dalszych badań naukowych.

Metody statystyczne są często trudne do zrozumienia i przez to niekiedy źle postrzegane przez szersze grono odbiorców. W wielu środowiskach, niestety także naukowych, statystykę traktuje się jako naukę dostępną tylko dla wtajemniczonych. Wiadomości o wynikach badań przekazywane przez różne instytucje często nie są właściwie odbierane, a czasem nawet zupełnie przeciwnie, niż wskazują uzyskane wyniki. Informacje przekazywane w formie zestawień liczbowych okazują się trudne w odbiorze. Współczesne społeczeństwo w dużym stopniu korzysta z wytworów kultury obrazkowej. Wszędzie można spotkać różne oznaczenia, symbole, ikony, piktogramy oraz obrazy. Umiejętne wykorzystanie obecnych możliwości technicznych w zakresie graficznej prezentacji wyników badań statystycznych może zatem ułatwić dużym grupom odbiorców pierwsze zetknięcie się z metodami statystycznymi i zachęcić ich do pogłębiania wiedzy, co w konsekwencji powinno doprowadzić do właściwego odbioru wyników badań i podawanych komunikatów.

Zastosowanie metod graficznej analizy danych pozwala m.in. na oczyszczenie danych, określenie ich struktury, wykrycie wartości odstających oraz ekstremalnych, identyfikację trendów i skupisk obserwacji, dostrzeżenie lokalnych wzorców, ocenę wyników modelowania i prezentację rezultatów badania.

Wszystko to jest niezbędne w przypadku eksploracyjnej analizy danych i eksploracji danych (Unwin 2015, s. XI).

Wizualizacja danych to stosunkowo nowy termin. Wyraża on ideę, że chodzi o coś więcej niż tylko przedstawienie danych zawartych w tablicy w formie graficznej. Można powiedzieć, że jest to swoiste opowiadanie historii zawartej w danych (Knaflic 2015). Grafiki okazują się właściwe do pokazania struktury danych i przedstawienia wyników badań. Są one zwykle zdecydowanie łatwiejsze w interpretacji niż tabele, które pozostają niezbędne do podawania dokładnych wartości analizowanych charakterystyk, a także raportów statystycznych, pomocnych przy podawaniu szacunków i porównań, a także umożliwiających przekazanie większej porcji informacji o charakterze jakościowym. Informacje kryjące się za danymi powinny być również ujawnione w dobrej prezentacji; grafika powinna pomóc czytelnikom lub widzom w dostrzeżeniu struktury w danych (Chen, Härdle i Unwin 2008).

Określenie „wizualizacja danych" łączy się z potrzebą graficznego przedstawienia informacji dostępnych w różnych zbiorach danych. Obejmuje ono graficzną prezentację wszystkich rodzajów informacji, nie tylko danych, i jest ściśle związane z badaniami prowadzonymi przez statystyków i informatyków. Dotychczasowe prace w tej dziedzinie koncentrowały się raczej na prezentacji informacji niż na tym, co można z niej wywnioskować. Jednak metody graficznej prezentacji zmierzają do umożliwienia przeprowadzenia wnioskowania na podstawie dostępnych danych. Bliższe powiązanie grafiki z modelowaniem statystycznym może sprawić, że stanie się to bardziej widoczne – jest to obiecujący kierunek badań, który ułatwiają stale zwiększające się możliwości dostępnego oprogramowania komputerowego. Duża w tym rola naukowców, a w szczególności statystyków.

Celem prezentowanej monografii jest przedstawienie zasad konstrukcji prezentacji graficznych, metod wizualizacji danych oraz kluczowych narzędzi wykorzystywanych w takich prezentacjach. Realizacja tego celu wymaga wprowadzenia pewnej systematyki dla metod graficznych, a w szczególności powiązania doboru odpowiedniego wykresu z rodzajem i strukturą danych, a konkretniej ze skalą pomiarową analizowanych zmiennych. Wszystko to może być pomocne dla naukowców prowadzących badania naukowe w różnych dyscyplinach, ponieważ prezentowane metody i narzędzia związane z wizualizacją danych są uniwersalne. Ważnym założeniem poczynionych rozważań stało się dążenie do wypracowania u Czytelnika umiejętności stawiania pytań badawczych na podstawie przeprowadzonej wstępnej, graficznej analizy danych. Antony Unwin (2015) podkreśla, że najłatwiej tego dokonać poprzez przedstawienie odpowiednich przykładów. Takie przykłady, wykorzystujące dostępne w programie R zbiory danych, zostały zamieszczone w ostatnich rozdziałach książki. Metody wizualiza-

cji danych odgrywają coraz większą rolę także w dydaktyce (Zelazny 2005; Żądło i Kończak 2009) i popularyzacji wiedzy z różnych dyscyplin (Kończak 2014).

W książce wyróżniono sześć rozdziałów. W rozdziale pierwszym przytoczono wybrane fakty historyczne dotyczące graficznego przedstawienia różnych zjawisk. Zamieszczone przykłady grafik i wykresów mają zupełnie inny charakter niż obecnie konstruowane prezentacje, chociażby z tego powodu, że powstały na długo przed erą nowoczesnych technologii. W drugim rozdziale zaprezentowano podstawowe zasady związane z konstrukcją wykresów. W szczególności wskazano na powiązanie skali, na jakiej dokonano pomiaru, z możliwymi sposobami wizualizacji danych. W rozdziale trzecim ujęto zwięzłą charakterystykę wybranych metod graficznych. Przedstawiono w nim podstawowe informacje o różnych rodzajach wykresów i zasadach doboru wykresu do określonego typu danych oraz ich struktury. W kolejnym rozdziale opisano podstawowe zagadnienia dotyczące pracy z programem R. To środowisko programistyczne jest uznanym standardem w badaniach naukowych, a dodatkowo posiada znaczne możliwości w zakresie metod graficznej prezentacji danych. W rozdziałach piątym i szóstym zaprezentowano możliwości pakietu graficznego **`ggplot2`** oraz jego licznych rozszerzeń. Pakiet **`ggplot2`** jest powszechnie używany do graficznej prezentacji rezultatów badań i poniekąd w ostatnich latach stał się standardem w prezentacji wyników badań naukowych.

Dla zwiększenia przejrzystości tekstu w pracy zastosowano następujące oznaczenia (ze względów technicznych nie dotyczy to elementów graficznych):

- **`ggplot2`** – nazwy pakietów oznaczono pogrubioną czcionką Consolas,
- **mtcars** – nazwy zbiorów danych oznaczono pogrubioną czcionką tekstu głównego,
- *mpg* – zmienne oznaczono kursywą czcionką tekstu głównego,
- *`plot`* – funkcje oznaczono kursywą czcionką Consolas.

Kody w języku R zostały wyróżnione na szarym tle i zapisywane w pracy są w następujący sposób.

```
# To jest forma zapisu kodów w języku R
ggplot(mtcars,aes(wt,mpg))+
  geom_point()
```

Wyniki wykonania prezentowanych kodów poleceń przedstawiono w pracy następująco.

```
##  WYNIKI OBLICZEŃ
##  SUMMARY Variable Pop.1 Pop.2 n.1 n.2 Statistic Observed
##  STATISTICS     x     A     B   8   5 diff.mean  0.45025
##  HYPOTHESIS      Null Alternative P.value
##  TEST identical     shifted  0.0238
```

W książce zamieszczono wiele kodów w języku R pozwalających na obróbkę danych oraz na konstrukcję różnorodnych wykresów. Kody te, niekiedy w nieznacznie zmodyfikowanej postaci, a także z wieloma pomocnymi dodatkami, dostępne są pod adresem: http://stat.ue.katowice.pl/wwbn (Kończak 2024).

1 Rys historyczny metod wizualizacji danych

Wykresy są potężnym narzędziem do odkrywania
i rozumienia złożonych zjawisk.

Edward R. Tufte*

Symbole i znaki towarzyszyły człowiekowi od najdawniejszych czasów. Już w czasach prehistorycznych na ścianach jaskiń kreślono symbole pozwalające oznaczyć liczbę upolowanych zwierząt. W różnych społecznościach wytworzyły się inne systemy liczenia i związane z nimi znaki graficzne odpowiadające poszczególnym liczbom. W starożytności przedstawiano antyczny świat na ręcznie rysowanych mapach, twierdzenie Pitagorasa dowodzone było także z wykorzystaniem odpowiednich rysunków, a obserwatorzy nieba wykreślali linie odpowiadające przemieszczaniu się gwiazd i planet na nieboskłonie.

Początkowo grafika statystyczna była stosowana do prezentowania szerszemu gronu statystyk ekonomicznych i populacyjnych. Pionierem wykorzystania wykresów i diagramów do prezentacji danych statystycznych był brytyjski inżynier, ekonomista i statystyk Wiliam Playfair. Wiek XIX i początek wieku XX to rozwój myślenia statystycznego, gromadzenia danych i ich prezentacji. Druga połowa XX wieku to dynamiczny rozwój technologii i pojawienie się nowych możliwości w zakresie wizualizacji danych statystycznych, w tym prezentacji dynamicznych i interaktywnych zamieszczanych na stronach internetowych.

* BooKey (b.r.) – tłumaczenie własne.

1.1. Charakterystyka wizualizacji zdarzeń i zjawisk od najdawniejszych czasów

Graficzna prezentacja różnego rodzaju zjawisk ma bardzo długą historię. Już w starożytnych cywilizacjach podejmowano próby wizualizacji danych. Rysunki znajdowane na ścianach i sklepieniach jaskiń miały prawdopodobnie znaczenie symboliczne i duchowe dla ówczesnych ludzi. Dominowała w nich tematyka animalistyczna, z przewagą wizerunków zwierząt takich jak konie, bizony, żubry, łanie, a czasem również drapieżniki jak lwy. W jaskiniach znajdowano namalowane na ścianach sylwetki ludzkie, hybrydy zwierzęco-ludzkie, postacie fantastyczne, rośliny oraz różne symbole i figury geometryczne (Wikipedia. *Malarstwo*… b.r.).

W starożytnym Egipcie używano hieroglifów (Wikipedia. *Hieroglify* b.r.) do przedstawiania informacji i opisu zdarzeń. Mimo że były one głównie formą pisemną, można je również traktować jako pierwsze próby wizualizacji danych, gdzie symbole i obrazy były używane do przekazywania konkretnych informacji. Stosowano je do rejestrowania danych demograficznych, takich jak liczba ludności i wiek. Hieroglify były również używane do przedstawiania obiektów i zjawisk astronomicznych, takich jak ruch planet i gwiazd.

W starożytnych Chinach znane są przypadki użycia graficznych reprezentacji danych w różnych dziedzinach, takich jak religia, astronomia czy medycyna. Zastosowanie grafik w opisach takich zagadnień miało na celu zobrazowanie złożonych koncepcji oraz obserwowanych zjawisk. Chińczycy byli w stanie opracować dokładny kalendarz i przewidywać fazy Księżyca oraz zaćmienia Słońca. Do przedstawiania danych astronomicznych używali różnych rodzajów grafik.

W średniowieczu zaczęto konstruować wykresy i diagramy na potrzeby zaprezentowania danych. Pierwsze diagramy słupkowe i kolumnowe do przedstawienia prędkości stale przyspieszającego obiektu przedstawił Francuz Nicole Oresme w XIV wieku (JPowered b.r.). Średniowieczni kartografowie konstruowali mapy, które przedstawiały znane obszary geograficzne. Matematycy tego okresu wykorzystywali proste wykresy do przedstawienia danych liczbowych. Rysunki, diagramy i wykresy pojawiały się w opisie zjawisk astronomicznych. Mikołaj Kopernik (lata 1473-1543) w XVI wieku używał diagramów do przedstawienia swojej heliocentrycznej teorii układu słonecznego (Wikipedia. *Heliocentryzm* b.r.). Wszystkie te wczesne wykresy i diagramy były jednak rysowane ręcznie i nie były tak precyzyjne jak współczesne cyfrowe wizualizacje.

Edward R. Tufte (red. 2013) przedstawia graficzne prezentacje plam słonecznych pochodzące z pracy *De Maculis Solaribus* (z roku 1613) Christophera

Schneinera, piszącego pod pseudonimem Apelles. Nieco wcześniej, ale w tym samym roku Galileusz jako pierwszy zaobserwował plamy na Słońcu. Na podstawie dobrze opracowanych rysunków, poprzez elegancki łańcuch rozumowania wizualnego, uzasadnił, że obserwowane zjawiska faktycznie występują na Słońcu, a nie są to obiekty, które tylko przemieszczają się na tle tarczy słonecznej.

W czasie rewolucji przemysłowej w XIX wieku nastąpił znaczny rozwój metod ilościowych, a także wzrost znaczenia naukowego podejścia do prezentacji danych. W 1822 roku Charles Minard (lata 1781-1870) skonstruował słynną mapę ilustrującą przemieszczanie się wojsk Napoleona podczas jego kampanii w Rosji. Ta mapa jest uważana za jeden z pierwszych przykładów wykresu przepływu. Znaczny wpływ na powstawanie i rozwój metod graficznej prezentacji miały również badania i prace takich naukowców jak William Playfair (lata 1759-1823), Florence Nightingale (lata 1820-1910) oraz John Snow (lata 1813-1858). Wiliam Playfair (Playfair 2005; Aigner i in. 2011) wprowadził i upowszechnił zastosowania wykresów kolumnowych, kołowych oraz liniowych. Florence Nightingale użyła diagramów słupkowych i wykresów kołowych do prezentacji statystyk dotyczących śmiertelności wśród żołnierzy brytyjskich w czasie wojny krymskiej. Natomiast John Snow na podstawie opracowanej mapy dostrzegł koncentrację zachorowań i zgonów wokół studni znajdującej się w Londynie. Także w tym czasie została opublikowana pierwsza mapa statystyczna. Jej autorem był Joseph Fletcher, a sama publikacja miała miejsce w czasopiśmie statystycznym w 1849 roku (Toit, Steyn i Stumpf 1986).

Druga połowa XX wieku to czas wprowadzenia komputerów i technologii cyfrowej do powszechnego użytku. Miało to duży wpływ na rozwój metod graficznej prezentacji danych. W tym czasie powstały programy komputerowe i narzędzia, które umożliwiały szybkie tworzenie i upowszechnianie zaawansowanych wykresów i grafik. Obecnie metody graficznej prezentacji danych są niezwykle różnorodne i dostępne dla szerokiego spektrum użytkowników. Skonstruowano wiele narzędzi, programów komputerowych i bibliotek, które umożliwiają tworzenie profesjonalnych wykresów, diagramów, grafik interaktywnych, map i infografik. Mimo że w ciągu wieków metody graficznej prezentacji danych stale ewoluowały, to dopiero technologie cyfrowe otworzyły nowe możliwości i narzędzia dla twórców wizualizacji danych. Współcześnie coraz więcej osób korzysta z tych metod, aby lepiej zrozumieć i przedstawić informacje, co ma ogromne znaczenie w wielu dziedzinach naukowych, biznesie oraz dydaktyce.

1.2. Wybrane najważniejsze historyczne prezentacje graficzne

Od najdawniejszych czasów znaki i symbole graficzne pojawiały się wszędzie tam, gdzie tylko przebywał człowiek. Początkowo były to na przykład różne rysunki pozostawiane w jaskiniach skalnych. Z tak odległych czasów pochodzą również symbole ułatwiające zliczanie różnych elementów, jak choćby upolowanych zwierząt. W następnych okresach różnorodne znaki graficzne pomagały przekazać treści na zwojach papirusów. Jedna z najstarszych znanych grafik przedstawiająca układ zabudowań miejskich (pierwowzór mapy) pochodzi z LXII wieku przed naszą erą. Już w starożytności wykorzystywano metody graficzne między innymi do dowodzenia lub uzasadniania pewnych własności geometrycznych. Euklides w *Elementach* (Księga I) przytacza osiem dowodów twierdzenia Pitagorasa, które można zaprezentować w postaci graficznej (Jeleński 1974). Potrzeba zliczania i przedstawienia różnych wielkości w sposób czytelny, nawet dla osób niepotrafiących czytać, prowadziła do powstawania pierwszych typowych wykresów.

Antony Unwin, Martin Theus i Heike Hofmann (2006) oraz Chun-houh Chen, Wolfgang Härdle i Antony Unwin (2008) wśród wielu przykładów zaczerpniętych z historycznych prezentacji graficznych wymieniają:

– marsz wojsk napoleońskich na Moskwę z lat 1811-1812 (z 1865 roku),
– różę Nightingale (z 1857 roku),
– rozkład jazdy pociągów na trasie Paryż–Lyon (z 1885 roku),
– ilustrację rozprzestrzeniania się epidemii cholery w Londynie (z 1854 roku).

1.2.1. Wybrane prezentacje graficzne sprzed 1800 roku

Od najdawniejszych czasów ludzie próbowali oznaczać znany teren na mapach. Milestones Project (DataVis. *Milestones*… b.r.) wskazuje jako najstarszą znaną mapę pochodzący z około 6200 roku przed naszą erą rysunek przedstawiający domostwa pewnego miasta, które zapewne zostało pokryte lawą podczas wybuchu wulkanu (DataVis. *The Oldest Map* b.r.; zob. Ataman b.r.). To tragiczne zdarzenie najprawdopodobniej było inspiracją do graficznego zobrazowania terenu w formie mapy. Również w tym serwisie przedstawiona jest informacja o prawdopodobnie pierwszej mapie znanego ówczesnego świata (DataVis. *World Map* b.r.). Mapa ta pochodzi z około 550 roku przed naszą erą, a jej autorem jest Anaximander z Miltus. Nie zachował się oryginał tej mapy,

dostępny jest jedynie opis w księgach II i IV *Dziejów* Herodota. Kolejna interesująca grafika to mapa świata Ptolemeusza (Wikipedia. *Ptolemy's World Map* b.r.) znana społeczeństwom grecko-rzymskim w II wieku, a opisana w książce Ptolemeusza *Geografia*, która została napisana około roku 150. Opierając się na inskrypcji w kilku najwcześniejszych zachowanych manuskryptach, tradycyjnie przypisuje się ją Agathodaemonowi z Aleksandrii (Wikipedia. *Mapa Ptomeleusza* b.r.).

Najwcześniejsza znana próba graficznego przedstawienia zmieniających się w czasie wartości pochodzi z X wieku (Tufte 1983). Po wcześniejszych próbach graficznej prezentacji układów stałych za pomocą odpowiednich linii przedstawiono zmieniające się w czasie położenie wybranych ciał niebieskich (*Planetary Movements Map* b.r.), pozycje w ciągu roku Słońca, Księżyca i znanych wówczas planet. Z kolei pierwsza znana mapa pogody (DataVis. *Halley's Wind Map*… b.r.), pokazująca dominujące wiatry na mapie geograficznej Ziemi (DataVis. *Weather Map* b.r.), pochodzi z roku 1686. Mapa ta została opracowana przez Edmonda Halleya.

Jedna z najwcześniejszych wizualnych reprezentacji danych statystycznych została narysowana w 1644 roku przez Michaela Florenta van Langrena, flamandzkiego astronoma na hiszpańskim dworze (Tufte 2019). Autor przedstawił na wykresie 12 różnych szacunków odległości między Toledo a Rzymem. Mierzona w stopniach długości geograficznej skala lokalizuje Toledo, historyczne hiszpańskie miasto znajdujące się na południku 0°.

W roku 1786 wieku Wiliam Playfair opublikował książkę *The Commercial and Political Atlas*. Umieścił w niej opracowane przez siebie wykresy słupkowe i kołowe. Ponieważ zastosowane narzędzia odwoływały się do zasad znanych z kartografii, w tytule książki znalazło się określenie „atlas". Na początku XIX wieku znano już prawie wszystkie nowoczesne formy grafiki danych: wykres kołowy, liniowy szeregu czasowego i wykres słupkowy. Większość z tych kluczowych rozwiązań zaproponował Szkot William Playfair. Z tego też powodu często jest on nazywany ojcem nowoczesnych metod graficznych (Friendly, Wainer b.r.).

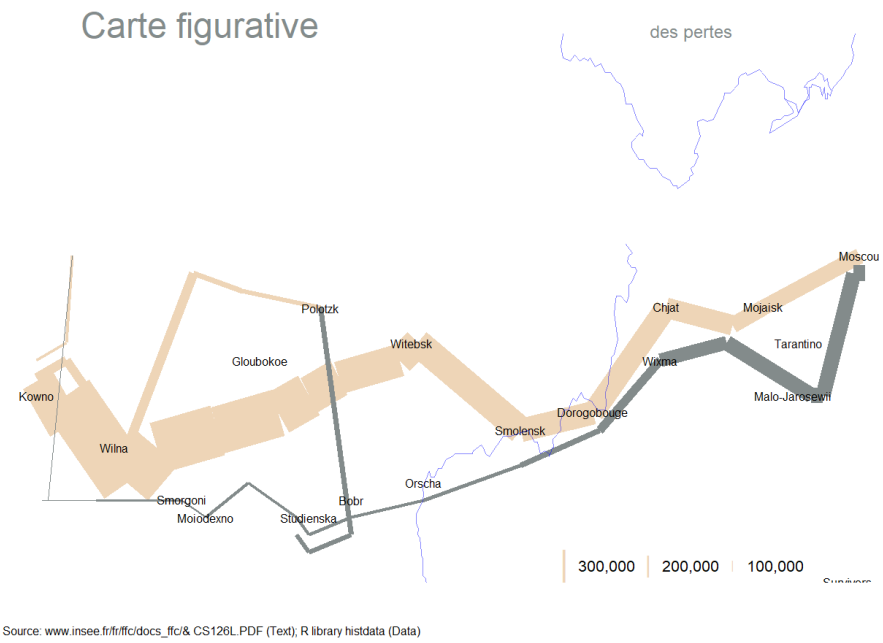
1.2.2. Wizualizacja marszu wojsk napoleońskich na Moskwę w latach 1812-1813

Charles J. Minard (lata 1781-1870) był inżynierem budownictwa. Uważany jest za pioniera w zakresie wykorzystania metod graficznych w inżynierii i statystyce. Michael Friendly przytacza ponad 70 prezentacji graficznych przedsta-

wionych przez Minarda (Friendly, 2024). Jedna z najczęściej przywoływanych takich prezentacji obrazuje marsz wojsk napoleońskich na Moskwę oraz ich dramatyczny odwrót. Opublikowana w 1869 roku – rok przed śmiercią Minarda – grafika ta wymownie podsumowuje katastrofalne przedsięwzięcie militarne Napoleona. Na mapie obejmującej obszar obecnych terenów Litwy, Białorusi i Rosji zwizualizował szczególnie wyrazistą zmienną statystyczną: gwałtowną i stałą utratę żołnierzy, jaką armia Napoleona poniosła w ciągu około sześciu miesięcy objętych grafiką (por. Cartographia b.r.; Martin Grandjean b.r.). Chociaż 420 tysięcy żołnierzy triumfalnie wyruszyło na Rosję w czerwcu 1812 roku, to gdy armia trzy miesiące później dotarła do Moskwy, była już znacznie zredukowana. Kiedy Napoleon nakazał wojskom wycofać się z Moskwy jesienią, wysyłał swoich ludzi na pewną śmierć, ponieważ musieli oni stawić czoła walce w obliczu niezwykle surowej zimy na rozległych równinach zachodniej Rosji (Rendgen 2018). Sytuację tę oddała omawiana mapa, przedstawiono na niej pięć zmiennych:

– wielkość armii (szerokość graficznego pasa),
– współrzędne geograficzne,
– kierunek marszu,
– położenie armii w określonym czasie,
– temperaturę.

Co można dostrzec, począwszy od lewej strony: na granicy polsko-rosyjskiej w pobliżu rzeki Niemen gruba brązowa linia pokazuje wielkość armii (422 tysiące ludzi), która ruszyła na Rosję w czerwcu 1812 roku, a także obraną przez nią drogę. W miejscu oznaczenia śmierci żołnierzy linia zwęża się; linia przepływu wskazuje liczbę pozostałych żołnierzy w danej pozycji na mapie. Pokazane są również ruchy oddziałów pomocniczych, które starały się chronić tyły i flanki nacierającej armii (Tufte 2006). Edward R. Tufte określił ten wykres jako najlepszy statystyczny wykres w całej historii (zob. Wills 2012). Wykres opracowany przez Charlesa Minarda jest dostępny w internecie (Wikimedia. *Carte Figurative*), a wykonana w programie R reprezentacja graficzna tego wykresu przedstawiona jest na rysunku 1.1. Wykres ten często uznaje się za jedną z najlepszych grafik statystycznych, jakie kiedykolwiek powstały.



Source: www.insee.fr/fr/ffc/docs_ffc/& CS126L.PDF (Text); R library histdata (Data)

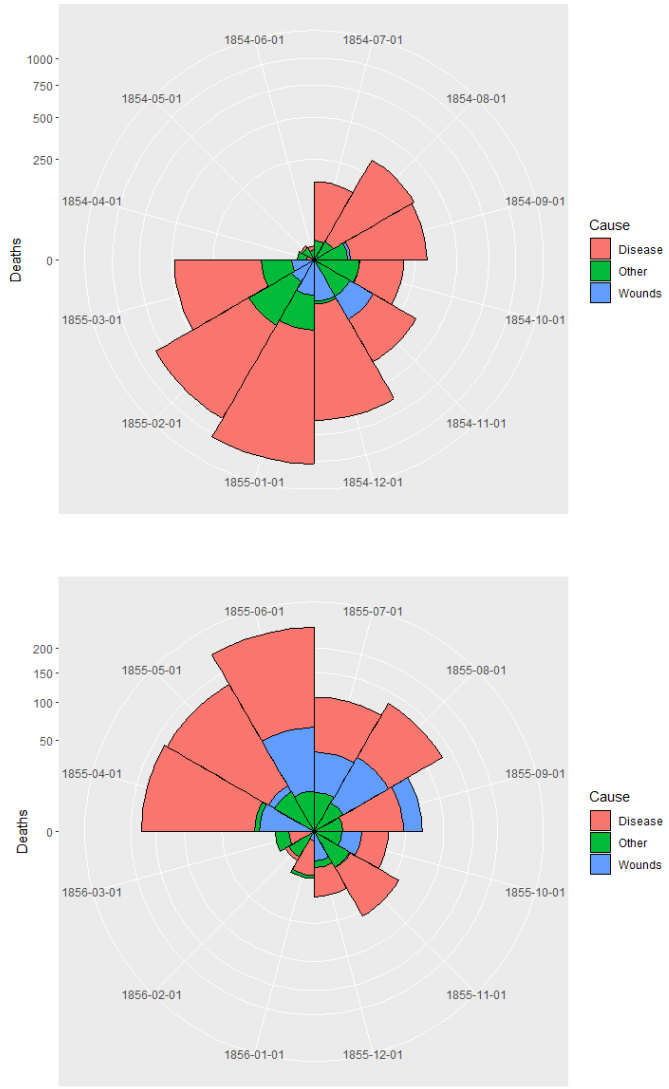
Rysunek 1.1. Marsz wojsk napoleońskich na Moskwę w latach 1812-1813

Źródło: opracowanie własne na podstawie pakietu HistData (b.r.).

1.2.3. Wykres róża Nightingale – wizualizacja śmiertelności żołnierzy brytyjskich uczestniczących w wojnie krymskiej

Florence Nightingale (lata 1820-1910) pochodziła z brytyjskiej arystokratycznej rodziny. Wbrew woli, a nawet przy sprzeciwie całej rodziny, zdecydowała się na podjęcie posługi pielęgniarskiej. W tamtych latach nie było to dobre zajęcie dla kobiety z arystokratycznego domu. W opinii najbliższych przyjęcie takiej pracy było wręcz haniebne. W roku 1854 Nightingale została wybrana do grona osób sprawujących opiekę nad rannymi w wojnie krymskiej żołnierzami brytyjskimi. Podjęła ona walkę o polepszenie opieki nad rannymi żołnierzami, upatrując w tym szansy na zmniejszenie ich śmiertelności. Musiała pokonywać uprzedzenia i sprzeciwy ze strony lekarzy, urzędników i oficerów. Swoją determinacją doprowadziła do poprawy fatalnego stanu sanitarnego szpitali polowych. Podjęła się skrupulatnej obserwacji i rejestracji danych dotyczących chorych i rannych żołnierzy. Rezultatem prowadzonych obserwacji i zapisów były między innymi obszerne zestawienia oraz stosowne wykresy pozwalające na diagnozę aktualnej sytuacji. Jeden z tych wykresów (DataVis. *The Causes of the Mortality*… b.r.) znany jest jako róża Nightingale lub wykres grzebieniowy

(coxcomb). W wyniku inicjatywy pielęgniarki w ciągu kilkunastu miesięcy śmiertelność wśród żołnierzy spadła z 43% do 2%. Doskonałe rezultaty opieki powszechnie przypisywano właśnie działaniom Florence Nightingale. Diagram sporządzony przez Nightingale jest dostępny w internecie (History of Information b.r.), a jego graficzną realizację w programie R prezentuje rysunek 1.2.

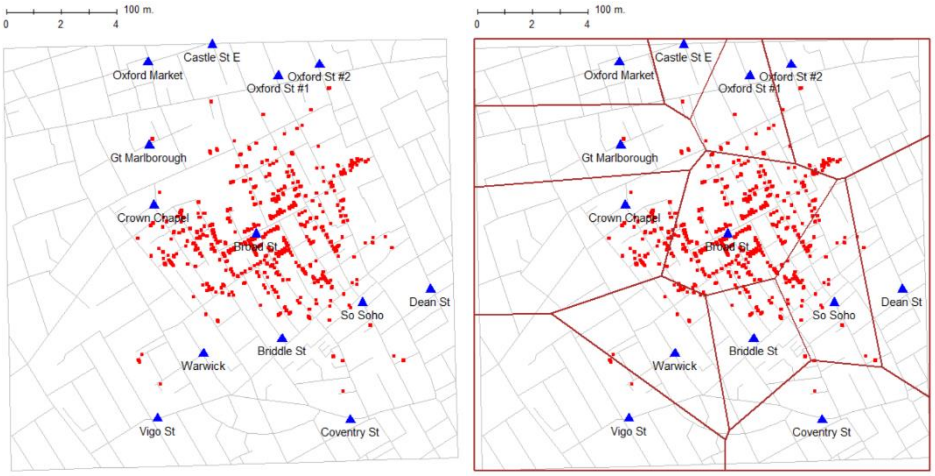


Rysunek 1.2. Zgony według przyczyn w armii brytyjskiej na Wschodzie od kwietnia 1854 roku do marca 1855 roku (góra) oraz od kwietnia 1855 roku do marca 1856 roku (dół)

Źródło: opracowanie własne na podstawie pakietu HistData (b.r.).

1.2.4. Epidemia cholery w Londynie w roku 1855 – wizualizacja zgonów na mapie miasta

Jeden z najstarszych przykładów zobrazowania na mapie rozprzestrzeniającej się epidemii pochodzi z roku 1855 (Chen, Härdle i Unwin 2008). W tym czasie w Londynie szerzyła się epidemia cholery. Doktor John Snow (lata 1813-1858) na podstawie opracowanej mapy dostrzegł koncentrację zachorowań i zgonów wokół studni znajdującej się przy Broad Street. Lokalizacja źródła zakażeń i wyłączenie pompy wodnej (usunięto uchwyt) pozwoliły na opanowanie epidemii, która jednak i tak pochłonęła blisko 500 ofiar. Dzięki wynikom z pracy nad tą epidemią John Snow jest uznawany za prekursora epidemiologii (Biecek 2014). Wykres dostępny jest w internecie (Wikipedia. *Snow Cholera Map* b.r.), a reprezentację graficzną przygotowaną w programie R przedstawia rysunek 1.3.

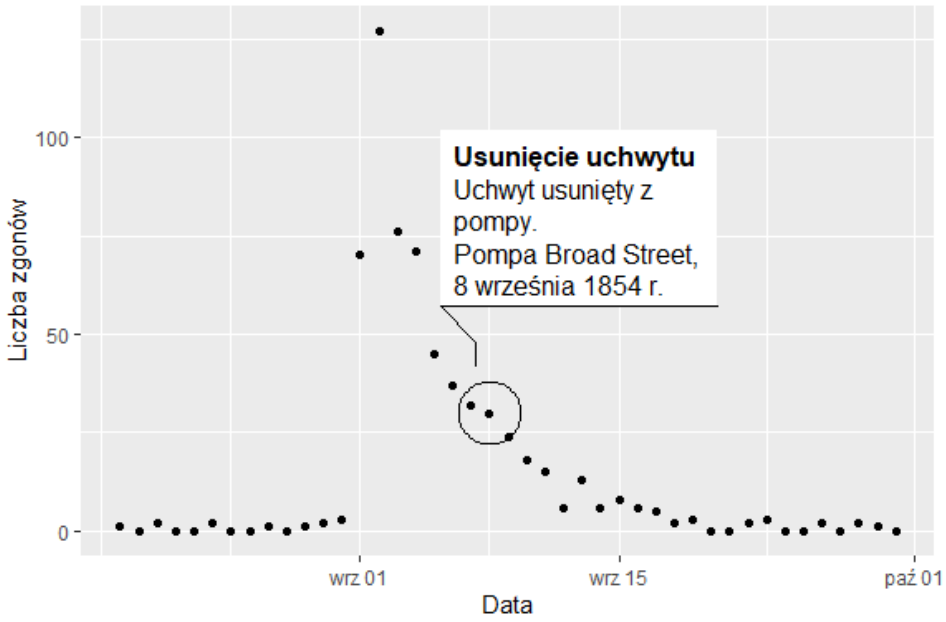


W prawej części dodatkowo zaznaczono obszary, w których mieszkańcy korzystali z danej studni.

Rysunek 1.3. Mapa rozprzestrzeniania się epidemii cholery w Londynie w 1855 roku na wzór mapy Johna Snowa

Źródło: opracowanie własne na podstawie danych z pakietu HistData (b.r.).

Doktor John Snow nie był w stanie ustalić przy pomocy analizy mikroskopowej ani chemicznej, co mogło być powodem występowania choroby. Jednak wyniki jego badań nad występowaniem cholery wystarczyły, aby przekonać lokalne władze do wyłączenia pompy poprzez usunięcie jej uchwytu. Na rysunku 1.4 przedstawiono liczbę zgonów z powodu epidemii cholery w Londynie w okresie od 19 sierpnia do 30 września 1854 roku. Na wykresie wyróżniono dzień 8 września 1854 roku, kiedy to usunięto uchwyt z pompy przy Broad Street.

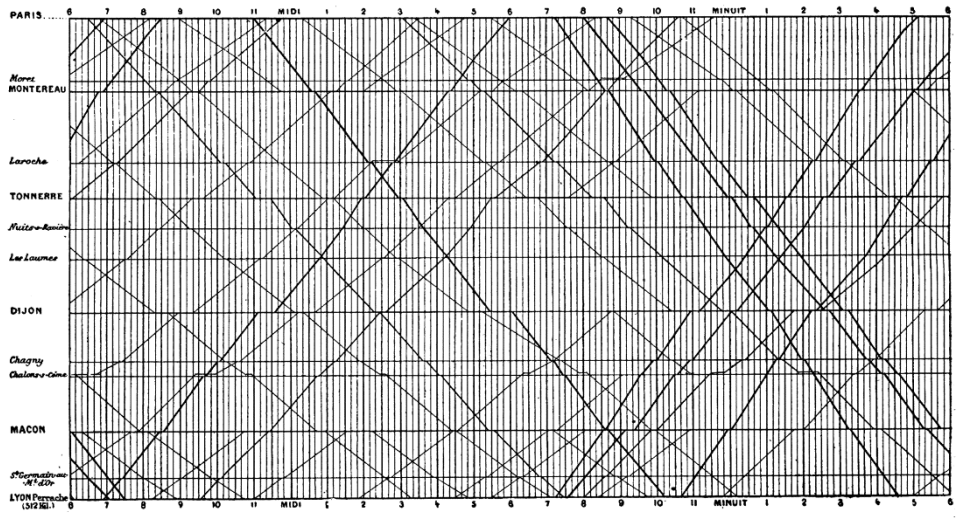


Rysunek 1.4. Liczba zgonów z powodu cholery w Londynie w okresie od 19 sierpnia do 30 września 1854 roku

Źródło: opracowanie własne na podstawie danych z pakietu HistData (b.r.).

1.2.5. Rozkład jazdy pociągów na trasie Paryż–Lyon z roku 1885 – praktyczna nietypowa wizualizacja

Zaprezentowanie wielu zmiennych na jednym czytelnym obrazie nie należy do prostych zadań. Etienne Jules Marey (lata 1830-1904), znany francuski naukowiec, zrealizował takie zadanie w bardzo nietypowej prezentacji. Przedstawił trasę, godziny odjazdów i przyjazdów, kierunek i prędkość jazdy, miejsce i czas postoju oraz miejsca mijania pociągów (por. rysunek 1.5) dla linii kolejowej Paryż–Lyon w roku 1885 (Marlena 2009).



Rysunek 1.5. Rozkład jazdy pociągów na trasie Paryż–Lyon w roku 1885

Źródło: Wikimedia (b.r.).

Przyjazdy i odjazdy ze stacji są zlokalizowane wzdłuż linii poziomej, czas trwania postoju na stacji jest opisany przez długość linii poziomej. Stacje są oddzielone proporcjonalnie do ich rzeczywistej odległości od siebie. Nachylenie linii odzwierciedla prędkość pociągu: im bardziej pionowa linia, tym szybszy pociąg. Przecięcie dwóch linii lokalizuje czas i miejsce, w którym mijają się pociągi jadące w przeciwnych kierunkach (Tufte 1983). Niezwykłe w swej istocie rozwiązanie do dziś zadziwia prostotą realizacji i skutecznością przekazu informacji (por. rysunek 1.5).

1.2.6. Inne wybrane przykłady historycznych prezentacji graficznych

Ujęte w rozważaniach przykłady z historii metod graficznych pokazują niekonwencjonalne podejście do prezentacji danych statystycznych. W przypadku wykresów dotyczących marszu wojsk napoleońskich na Moskwę, jak i w zobrazowaniu rozkładu jazdy pociągów na trasie Paryż–Lyon, imponująca jest ilość informacji przekazana za pomocą stosunkowo prostego rysunku. W wizualizacjach dotyczących opieki nad żołnierzami brytyjskimi i epidemii cholery odpowiednio przygotowane zestawienia i prezentacje graficzne prowadziły do podjęcia działań, które w konsekwencji uratowały życie wielu ludzi. Wskazane przykłady bardzo trafnie podsumowują słowa: „Statystyka jest bardziej sposobem myślenia

lub wnioskowania niż pęczkiem recept na młócenie danych w celu odsłonięcia odpowiedzi" (Rao 1994, s. 64). Myśl tę w odniesieniu do prezentacji graficznych zrealizował z początkiem XIX wieku William Playfair, publikując książkę *The Commercial and Political Atlas* (zob. 2005; pierwsze wydanie w roku 1801) i przedstawiając w niej wiele różnorodnych propozycji prezentacji graficznej danych statystycznych. Opisane przykłady zastosowania metod graficznych, a także wiele innych, przedstawia linia czasu prezentacji graficznych dostępna na witrynie *Milestones in the History of Thematic Cartography, Statistical Graphics, and Data Visualization* (Friendly i Denis b.r.). Na wspomnianej linii czasu ujęto między innymi tak ważne opracowania, książki, rodzaje wykresów jak:

- krzywa Lorenza z roku 1905,
- wykres sita z roku 1983,
- wprowadzenie systemu GIS w kartografii z roku 1960,
- propozycja wykresu gwiazdowego z roku 1971,
- propozycja wykresu twarzy Chernoffa z roku 1973,
- wprowadzenie wykresu mozaikowego z roku 1981,
- wprowadzenie wykresu współrzędnych równoległych z roku 1981,
- pierwsze wydanie książki *Grammar of Graphics* z roku 1999,
- prezentacja Gapminder z roku 2005,
- computational language graphics: **`ggplot2`** z roku 2006.

Na przywołanej witrynie wymieniono wiele innych ważnych faktów z historii dotyczącej graficznych metod wizualizacji danych.

2 Podstawowe określenia i zasady konstrukcji wykresów

Celem wizualizacji jest wgląd, a nie obrazy.

Ben Shneiderman*

Metody graficzne są powszechnie wykorzystywane w analizie danych. Takie wykresy jak histogram, wykres rozrzutu, wykres liniowy czy wykres pudełkowy to wręcz standardy pozwalające na przeprowadzenie wstępnej analizy dostępnych danych. Dla właściwego wykonania takiej analizy niezbędne staje się określenie podstawowych pojęć jak badanie statystyczne, populacja statystyczna oraz próba. Do wykonania prezentacji graficznej konieczny jest także zbiór danych. Pozyskane zbiory danych zwykle składają się z wielu zmiennych. Niektóre z tych zmiennych są wyrażane w postaci liczbowej, inne są charakteryzowane w postaci ciągów znaków lub napisów, a tylko dla niektórych z tych napisów można ustalić porządek. Wyniki pomiarów wiążą się z zastosowaną skalą pomiarową, ta zaś determinuje możliwości skorzystania w dalszej analizie z określonych metod statystycznych, a także pozwala wskazać na potencjalne zastosowania odpowiednich metod graficznych. W dalszej części rozdziału przybliżono najważniejsze rodzaje wykresów i zwrócono uwagę na ich typowe zastosowania, uwzględniając rodzaj przeprowadzanej analizy oraz skale pomiarowe, na których są rejestrowane badane zmienne.

* Inspiring Quotes (b.r.) – tłumaczenie własne.

2.1. Badanie statystyczne. Populacja i próba

Statystyka jest nauką, która bada i opisuje prawidłowości w zjawiskach masowych. Wykorzystuje w tym celu specyficzne metody badań, zbierania danych oraz prezentacji wyników (Sobczyk 2001). Wyniki mogą być przedstawiane w formie opisowej, algebraicznej (stosowne wzory), zestawu liczb, które zwykle ukazywane są w odpowiednio zbudowanych tablicach, a także w formie prezentacji graficznej.

Podstawowe pojęcia związane z badaniem statystycznym i w konsekwencji z analizą wyników takiego badania, a w szczególności z graficzną prezentacją takich wyników, to między innymi populacja, jednostka badania statystycznego, cecha statystyczna, próba oraz próba losowa (Sobczyk 2001; Wawrzynek 2007):

1. Populacja – określana również jako zbiorowość statystyczna lub uniwersum, to zbiór elementów, które posiadają co najmniej jedną wspólną cechę, różnicującą te elementy między sobą. Wyłącznie cecha, która ma przynajmniej kilka wariantów lub różnych wartości, jest interesująca dla przeprowadzającego badanie statystyczne. Zbiorowością statystyczną może być na przykład zbiór gospodarstw domowych na określonym terenie, państwa, powiaty, gminy, uczniowie szkół podstawowych, wszyscy zatrudnieni na umowę o pracę, pracownicy określonego przedsiębiorstwa;
2. Jednostka badania statystycznego – jednostką statystyczną jest dowolny element populacji. Jest to obiekt (na przykład osoba, rzecz, zjawisko) wyodrębniony do celów badań statystycznych;
3. Cecha statystyczna – cechami statystycznymi (inaczej zmiennymi) określa się właściwości, ze względu na które badana jest zbiorowość. Istnieje wiele kryteriów podziału cech statystycznych. Cechy mogą być stałe i zmienne. Cechy stałe przyjmują jeden poziom, wspólny dla wszystkich jednostek badanej zbiorowości. Cechy zmienne różnicują jednostki badania statystycznego. Inny podział wyróżnia:
   a) cechy ilościowe (metryczne, wyrażone w jednostkach miary),
   b) jakościowe (niemetryczne, warianty wyrażane jedynie słownie).

   Wśród cech ilościowych wyróżnia się:
   – skokowe (mogą przyjmować skończoną lub przeliczalną liczbę wartości),
   – ciągłe (mogą teoretycznie przyjmować dowolną wartość z pewnego przedziału);
4. Próba – stanowi podzbiór populacji generalnej. Na podstawie pobranej próby przeprowadzane są analizy opisowe i wnioskowanie statystyczne;
5. Próba losowa – to losowy podzbiór populacji. W wielu przypadkach badań statystycznych zaleca się właśnie losowy dobór próby. Pozwala to zwykle uniknąć sytuacji, kiedy pobrana próba nie stanowi dobrego odwzorowania populacji generalnej.

W przeprowadzanych badaniach kluczowy jest proces obserwacji statystycznej oraz pomiaru. Obserwacja statystyczna to proces zbierania danych zgodnie z przyjętymi procedurami. Proces ten polega na gromadzeniu informacji na temat określonych cech jednostek badanej populacji lub próby. Zwykle jedna obserwacja jest zapisywana w pojedynczym wierszu tabeli, a na podstawie przeprowadzonych pomiarów uzyskuje się odpowiednią tablicę danych. Cel obserwacji statystycznej sprowadza się do dostarczenia danych obrazujących rozkład badanych cech statystycznych w analizowanej zbiorowości (Czempas 2000). Pomiar to przyporządkowanie liczb właściwościom obiektów zgodnie z ustalonymi regułami tak, aby liczby odzwierciedlały relacje zachodzące pomiędzy tymi obiektami (Portal Statystyczny b.r.). Celem pomiaru jest takie przedstawienie treści dokonanych obserwacji statystycznych na jednostkach statystycznych, aby symbole były ze sobą związane tak samo, jak są połączone ze sobą analizowane jednostki, zdarzenia lub zjawiska opisujące te pojęcia. Pomiar stanowi podstawowe narzędzie w naukach społecznych, ekonomii, psychologii, medycynie i wielu innych dziedzinach, gdzie istnieje potrzeba zbierania i analizowania danych w celu zrozumienia istoty zjawisk, konstruowania prognoz, podejmowania decyzji czy weryfikowania hipotez.

2.2. Skale pomiarowe

Statystyka ma własny zestaw narzędzi do wizualizacji typowych i specyficznych zbiorów danych. Wykresy statystyczne można sklasyfikować na różne sposoby, w tym według formy graficznej prezentacji lub rodzaju danych przedstawionych na wykresie. Dane statystyczne prezentowane na wykresach są zwykle opisywane przez ich skalę: nominalną, porządkową lub liczbową (skokową lub ciągłą). Najważniejszą cechę odróżniającą grafikę statystyczną od innych metod statystycznych stanowi jej uniwersalność. Grafika statystyczna nie jest dostosowana tylko do jednego konkretnego zastosowania, ale przyjęte zasady obowiązują dla dowolnych danych mierzonych w odpowiednich skalach. W zależności od skali pomiarowej do prezentacji graficznej dostępna jest określona gama wykresów statystycznych. To właśnie skala pomiarowa danych w znacznej mierze determinuje możliwości odwołania się do konkretnych form graficznej prezentacji danych. W badaniach statystycznych wyróżnia się następujące cztery rodzaje skal pomiarowych (Domański, Pruska i Wagner 1998):

- nominalna,
- porządkowa,
- przedziałowa (interwałowa),
- ilorazowa (stosunkowa).

2.2.1. Skala nominalna

Wartości mierzone na skali nominalnej nie mają oczywistego uporządkowania (na przykład nazwy miejscowości, województw, określenia barw). Jedyną dozwoloną relacją porównującą dwie wartości na skali nominalnej jest równość. Wśród skal nominalnych wyróżnia się czasem skale dychotomiczne, gdzie zmienne przyjmują tylko dwie wartości, na przykład odpowiedź na pytania „tak" lub „nie". Dla pomiarów dokonanych dla zmiennych mierzonych na skali nominalnej dozwolone są następujące operacje:

– zliczanie,
– obliczanie frakcji (procent, odsetek całości),
– wyznaczenie dominanty (wartości najczęstszej),
– wykonanie binaryzacji (przypisanie wartości liczbowych, na przykład 0 i 1) dla zmiennych przyjmujących dwa warianty.

2.2.2. Skala porządkowa

Wartości na skali porządkowej (na przykład poziom wykształcenia) mają precyzyjnie określony porządek. Nie są jednak określone odległości pomiędzy nimi i z tego powodu nie można obliczać różnic pomiędzy takimi wielkościami. Oprócz operacji dozwolonych na skali nominalnej możliwe są następujące operacje:

– porównywanie, która wartość jest mniejsza, a która większa (ale bez możliwości określania różnicy),
– rangowanie i metody rangowe, w szczególności obliczanie współczynników korelacji rang Spearmana oraz tau Kendalla,
– wyznaczanie kwantyli,
– wyznaczanie minimum oraz maksimum.

2.2.3. Skala przedziałowa

Dla wartości określonych na skali przedziałowej różnice pomiędzy nimi mają sensowną interpretację, ale nie ich ilorazy. Dopuszczalne operacje dla danych mierzonych na skali przedziałowej to wszystkie dozwolone dla danych mierzonych na skali porządkowej, a ponadto:

– obliczanie średniej, wariancji, odchylenia standardowego,
– obliczanie wartości współczynnika korelacji liniowej Pearsona,

– wyznaczanie funkcji regresji,
– dodawanie i odejmowanie wartości, obliczanie różnic,
– mnożenie i dzielenie, ale wyłącznie przez stałą.

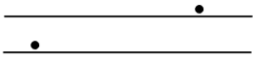
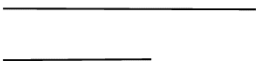
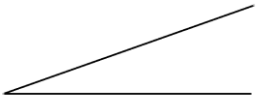
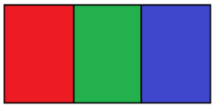
2.2.4. Skala ilorazowa

Dla wartości określonych na skali ilorazowej nie tylko różnice, ale także ilorazy wielkości mają interpretację. Przykładami są długość i masa (coś może być dwa razy dłuższe lub dwa razy cięższe). Na wielkościach mierzonych na tej skali można wykonywać wszystkie operacje dostępne do pomiarów na skali przedziałowej, a ponadto:
– obliczać zmiany względne (procentowe) w szeregu czasowym,
– mnożyć i dzielić wielkości interwałowe,
– logarytmować i potęgować,
– wyznaczać średnią kwadratową, harmoniczną i geometryczną.
Zastosowana skala pomiarowa w znacznej mierze wpływa na możliwość wyboru konkretnej formy prezentacji graficznej danych.

2.3. Podstawowe zasady konstrukcji wykresów

Wartości liczbowe można na wykresach przedstawiać w różnorodny sposób, na przykład wyrazić je za pomocą długości odcinków kątów nachylenia kolorów czy też wielkości powierzchni. William S. Cleveland i Robert McGill (1987) podają, że najdokładniej na wykresie odczytywane są wielkości przedstawione na wspólnej osi liczbowej. Nieco mniej zauważalne dla odbiorcy są wielkości zobrazowane na oddzielnych osiach. Stosunkowo dobrze są odbierane wielkości przedstawiane w postaci odcinków o długościach odpowiadających tym wielkościom. Nieco trudniej ocenić wielkości, które na wykresie przedstawiono w postaci wielkości kąta pomiędzy odcinkami. Kolejnym elementem pozwalającym na porównanie wielkości są pola figur płaskich. Zazwyczaj w tym celu wykorzystuje się prostokąty. Taka metoda przedstawienia wielkości okazuje się znacznie mniej poprawnie odbierana niż na przykład długości odcinków. Jeszcze trudniej ocenić wielkości przedstawiane graficznie jako objętości figur (zwykle prostopadłościany). Do przedstawienia wartości zmiennych mogą być również wykorzystane kolory, nie są one jednak w naturalny sposób kojarzone z uporządkowanymi wielkościami. W przypadku wykorzystania kolorów do przedstawienia wartości zmiennych niezbędne okazuje się załączenie odpowiedniej legendy.

Tabela 2.1. Hierarchia percepcji elementów graficznych

Kolejność postrzegania	Opis	Przykład
1	Pozycja względem wspólnej skali	
2	Pozycja względem oddzielnych skal	
3	Długość	
4	Kąt i nachylenie	
5	Pole	
6	Objętość	
7	Kolor	

Źródło: opracowanie własne na podstawie Wilkinson (2005).

W tabeli 2.1 przedstawiono hierarchię postrzegania elementów graficznych umieszczanych na wykresach w celu przekazania informacji o danych liczbowych (Cleveland i McGill 1987; Wilkinson 2005).

Dla zapewnienia właściwego odbioru zamieszczonych treści wykres powinien zawierać następujące informacje (Unwin 2015, s. 260; zob. Kassambara 2013):

1. Tytuł – powinien odnosić się do danych zamieszczonych na wykresie. Niekiedy tytuł może kierować uwagę na jakieś specyficzne informacje zamieszczone w grafice. W publikacjach jak artykuły lub książki tytuł wykresu nie jest zamieszczany w obrębie samego wykresu, ale zwykle bezpośrednio pod wykresem;

2. Podtytuł – w niektórych przypadkach uzasadnione jest dodanie na wykresie podtytułu;
3. Podpis – zamieszczony pod wykresem powinien objaśniać to, co jest pokazane na grafice. Pod wykresem zwykle należy podać również źródło. Podpisy powinny być na tyle szczegółowe, aby grafika mogła być właściwie odczytana. Bardziej rozbudowany podpis jest zwykle lepszy niż podpis zwięzły;
4. Obszar wykresu – obszar do przedstawienia danych, na podstawie których skonstruowano dany wykres;
5. Etykiety – na wykresie powinny być właściwie opisane osie. Często, jeżeli nie jest to jednoznacznie określone w podpisie, powinno się zaznaczyć jednostki, w których dane są wyrażone. Prawidłowo umieszczone etykiety ułatwiają skupienie się na grafice, ponieważ nie ma potrzeby szukania tych informacji w innym miejscu;
6. Skale – dobrze przedstawione skale (zwykle na osiach OX i OY) z właściwie dobranymi podziałkami liczbowymi, które mają znaczenie dla danych, pomagają czytelnikom zrozumieć, że dane są ważne. Ułatwiają również zrozumienie rzędów wielkości prezentowanych danych;
7. Legenda – jeśli na wykresie znajdują się różne grupy o różnych kolorach, rozmiarach lub kształtach, to legenda powinna je objaśnić;
8. Adnotacje – jeśli dana cecha ma być wyróżniona, na przykład wartość odstająca lub braki w danych, wówczas na samym wykresie można dodać adnotację;
9. Tekst objaśniający – wykresy powinny być omawiane w tekście towarzyszącym lub przynajmniej przywoływane. W opisie należy umieścić odwołania do rysunku wraz z podaniem numeru. Daje to możliwość skomentowania poszczególnych cech w sposób bardziej szczegółowy oraz dodanie dodatkowych uwag. Najlepiej, gdy opis wykresu i grafika znajdują się blisko siebie;
10. Źródło – w obszarze wykresu może być podane źródło, ale najczęściej jest ono zamieszczane pod wykresem. W przypadku, gdy jest to opracowanie autora pracy, dopuszczalne jest pominięcie źródła.

Nie wszystkie wymienione elementy muszą się znaleźć na wykresie. W opracowaniach zwartych tytuł i źródło zamieszcza się zwykle pod rysunkiem. Bardzo często nie ma potrzeby zamieszczania podtytułu oraz legendy. Dodatkowo na wykresie mogą zostać umieszczone inne elementy typu siatka lub linie referencyjne. Wprowadzenie takich dodatkowych elementów zależy jednak od kontekstu, czy na przykład autor chce zwrócić uwagę na pewne wielkości przedstawione na wykresie.

Antony Unwin (2015, s. 259) podaje następujące zalecenia dotyczące konstrukcji wykresów:

1. Na wykresie rozrzutu zmiennych powiązanych przyczynowo zmienna zależna jest rysowana na osi pionowej, a zmienna objaśniająca na osi poziomej;
2. Liczby na osiach rosną w prawo oraz w górę;
3. Oś OX przecina oś OY w punkcie y = 0. Jeżeli tak nie jest, to należy ten fakt wyraźnie zaznaczyć;
4. Skale są liniowe, a gdy nie są, to należy ten fakt wyraźnie wskazać;
5. Czas jest zwykle przedstawiany na osi poziomej, postępując od lewej do prawej;
6. Grafika jest zawsze rysowana tak, aby pokazać wszystkie dane. Jeśli niektóre przypadki znajdują się poza zakresem, należy to wyraźnie zaznaczyć;
7. Współczynniki proporcji (stosunek wysokości grafiki do jej szerokości) powinny być tak dobrane, aby nachylenie linii przekątnej wynosiło około 45°, co zostało po raz pierwszy dokładnie omówione w pracy (Cleveland i McGill 1987);
8. Punkty zwykle reprezentują poszczególne przypadki, a obszary – zliczenia lub wagi;
9. Pionowe słupki reprezentują częstości zmiennych ciągłych, gdy nie ma przerwy między nimi. Słupki z odstępami pomiędzy nimi reprezentują zmienne jakościowe lub ilościowe dyskretne;
10. Do reprezentacji grup należy używać wyraźnych kolorów, a cieniowanie lub ciągłe spektrum używa się do reprezentowania zmiennych wyrażanych na skalach ciągłych.

Przedstawiony zbiór zasad nie jest zbiorem zamkniętym. Przy konstrukcji różnych wykresów należy uwzględniać szereg innych wskazań i zaleceń. Należy unikać między innymi znacznych pustych obszarów na obrzeżach wykresu. Uzyskuje się to poprzez właściwy dobór skali, tak aby początek skali był nieco poniżej najmniejszych wartości, a koniec nieco powyżej największych wartości zmiennej. Jak podkreśla William J. Reichmann (1968, s. 37), jeśli wartości wykresu położone są bardzo wysoko, to wolno pominąć białą przestrzeń u podstawy wykresu, o ile tylko będzie wiadomo, że zostało to dokonane. W dalszej części Reichmann dodaje, że nie można podać ogólnych i jednoznacznych norm dla prezentacji graficznych. Jest to do pewnego stopnia zdeterminowane podejściem autora i chęcią zachowania artystycznych proporcji.

2.4. Gramatyka grafiki i jej realizacja w pakiecie ggplot2

Leland Wilkinson w roku 1999 zaproponował oryginalne zasady konstrukcji wykresów statystycznych (zob. Wilkinson 2005). Autor szczegółowo przedstawił wszystkie główne aspekty związane z efektywną wizualizacją danych. Gramatyka grafiki to idea (Moulik 2018), która umożliwia konstruowanie i opisywanie różnych rodzajów wykresów statystycznych poprzez deklaratywne specyfikacje. Ta gramatyka stanowi podstawę dla wielu narzędzi i bibliotek do tworzenia wykresów i wizualizacji danych, takich jak w szczególności dla pakietu **ggplot2** w języku R. Pakiet ten zostanie szerzej omówiony w dalszej części niniejszej pracy.

Gramatyka grafiki zaproponowana przez Wilkinsona pozwala użytkownikom tworzyć złożone wykresy poprzez kolejne zastosowanie drobniejszych elementów składowych (warstw) w spójny sposób. Użytkownicy mogą konstruować różnorodne wykresy, dostosowując je do swoich potrzeb w sposób deklaratywny, co oznacza, że skupiają się na tym, co chcą przedstawić, a nie na tym, jak to zrobić. Gramatyka grafiki zaproponowana przez Wilkinsona opiera się na następujących podstawowych elementach:

1. **Dane (data)**: wskazanie zbioru z danymi, które mają być przedstawione w formie graficznej na wykresie;
2. **Mapowanie (aesthetic mapping)**: określa, jakie właściwości i charakterystyki danych mają zostać zaprezentowane na wykresie oraz jaka ma być forma prezentacji (kolor, rozmiar, kształt);
3. **Geometria (geometry)**: określa rodzaj geometrii, który ma zostać użyty do przedstawienia danych. Przykładami rodzaju geometrii są punkty, linia, słupki i obszar;
4. **Transformacja statystyczna (statistical transformation)**: określa przekształcenia statystyczne, które mogą być stosowane do danych przed wygenerowaniem wykresu. Może to być wyznaczenie różnych mierników, jak na przykład obliczanie średniej, mediany, sumy, odchylenia standardowego;
5. **Panele (faceting)**: funkcja pozwala na tworzenie wielu paneli, które mogą przedstawiać określony podzbiór danych według wybranej cechy.

Zrozumienie gramatyki pakietu **ggplot2** ma kluczowe znaczenie dla efektywnego korzystania z niego. Kluczową koncepcją uwzględnioną w tworzeniu wykresów za pomocą pakietu **ggplot2** jest warstwowanie. Oznacza to, że **ggplot2** daje użytkownikom swobodę myślenia o grafice, którą chcieliby stworzyć w wysoce konfigurowalnej strukturze. Realizacja grafiki w **ggplot2** ma ważną właściwość polegającą na umożliwieniu użytkownikom korzystania z modułowych fragmentów kodu w celu tworzenia pięknych wykresów dokładnie

według specyfikacji pożądanej przez użytkownika. Zasady konstrukcji wykresów z wykorzystaniem gramatyki grafiki w pakiecie **ggplot2** zostaną szczegółowo przedstawione w następnych rozdziałach niniejszej pracy.

3 Charakterystyka wybranych metod graficznych stosowanych w analizie wyników badań naukowych

Przede wszystkim pokazuj dane.

Edward R. Tufte*

W literaturze znanych jest wiele różnego rodzaju form graficznej prezentacji danych. W praktyce szerokie zastosowanie znalazły wykresy słupkowe, kołowe, histogramy, wykresy punktowe, wykresy pudełkowe, wykresy mozaikowe i wiele innych sposobów wizualizacji danych. Wybór postaci wykresu zależy od rodzaju prezentowanych danych (liczby zmiennych, skal pomiarowych) oraz od tego, na co autor analizy zamierza zwrócić uwagę. Niewłaściwy wybór typu wykresu może całkowicie wypaczyć przekaz zawarty w danych. Nie zawsze jednak istnieje jedyny i optymalny wybór formy graficznej, a nawet zwykle możliwe są do wyboru różne formy takiej prezentacji. Wybór przyjętych domyślnych rozwiązań w określonym programie komputerowym nie zawsze okazuje się dobrym rozwiązaniem. Duża tu rola osoby wykonującej prezentację graficzną.

O ile wybrano już odpowiedni typ wykresu, o tyle nadal istnieje wiele opcji (ustawienia różnych parametrów) do rozważenia. W tym rozdziale zaprezentowano zwięzłą charakterystykę najczęściej stosowanych typów wykresów. Przedstawiony zestaw, mimo że jest dość obszerny, nie wyczerpuje wszystkich rodzajów wykresów spotykanych w literaturze, a ujmuje te najczęściej wykorzystywane w praktyce badań naukowych.

* Tufte (1983, s. 92) – tłumaczenie własne.

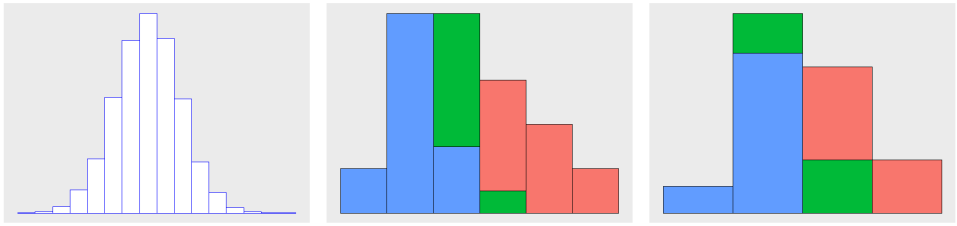
3.1. Charakterystyka wybranych typów wykresów*

Możliwości prezentacji graficznych dla określonego zbioru są w dużej mierze zdeterminowane używanym oprogramowaniem. W niniejszej pracy skoncentrowano się na programie R. Jednak w podstawowym zakresie wykresy opisane w tej części są dostępne w różnych programach przeznaczonych do analizy danych statystycznych i umożliwiających graficzną prezentację danych. W programie R użytkownik może skorzystać z wielu sposobów wizualizacji danych. Funkcje graficzne dostępne w podstawowych bibliotekach instalowanych wraz z programem R pozwalają na odwołanie się do różnorodnych wykresów. Po zainstalowaniu dodatkowych pakietów, takich jak np. **lattice**, **vcd**, **plotrix**, **iplots**, **playwith**, **plotly**, **ggvis**, **ggraph**, **gganimate**, **ggplot2**, a także wielu innych, możliwości graficznej prezentacji zostają znacznie rozszerzone. W dalszej części skoncentrowano się na funkcjach dostępnych w pakietach podstawowych, a szczególną uwagę poświęcono przedstawieniu możliwości pakietu **ggplot2** oraz wybranych pakietów rozszerzających jego możliwości.

W prezentowanym zestawieniu najczęściej stosowanych wykresów nie ujęto wszystkich typów grafik, ograniczając się tylko do tych częściej wykorzystywanych do wizualizacji danych i jednocześnie wykorzystanych w dalszej części niniejszej książki. Scharakteryzowane typy wykresów wystarczą do przeprowadzenia analizy graficznej praktycznie we wszystkich sytuacjach, a tylko w wyjątkowych przypadkach będzie konieczność odwołania się do innych, nietypowych form wykresów.

* W podrozdziale przy omówieniu poszczególnych typów wykresów zamieszczono ich graficzne prezentacje – stanowią one opracowanie autorskie.

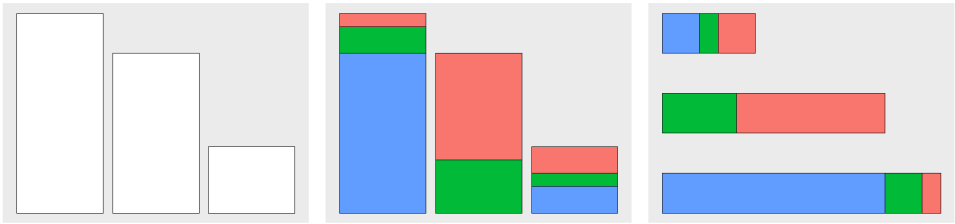
3.1.1. Histogram



Histogram to wykres, który za pomocą odpowiednio rozmieszczonych słupków przedstawia rozkład zmiennej ciągłej. Podczas konstrukcji należy odpowiednio podzielić zakres zmienności wartości danych na określoną liczbę przedziałów. Dla każdego takiego przedziału zlicza się znajdujące się w nim obserwacje. Dla przedziałów o jednakowej długości wielkość zjawiska jest wizualizowana jako kolumna o wysokości odpowiadającej liczbie obserwacji. Jeśli przedziały nie są takiej samej długości, to wysokość słupków odpowiada gęstościom wyznaczanym jako iloraz liczebności i długości danego przedziału. W obu przypadkach o wielkości zjawiska informuje pole danego słupka.

Histogram jest przydatny do analizy rozkładu danych, identyfikacji tendencji centralnej, poziomu zmienności danych oraz asymetrii rozkładu. Pozwala także na wykrywanie odstępstw, wartości skrajnych i ekstremalnych. Histogramy są często używane w statystyce i analizie danych, aby uzyskać w prosty i czytelny sposób obraz rozkładu danej zmiennej. Na podstawie histogramu można skonstruować wielobok liczebności. Jest to łamana, która łączy środki górnych podstaw histogramu. Niekiedy w analizach wykreśla się histogram dla szeregu skumulowanego. W takim przypadku także można skonstruować wielobok liczebności (diagram) dla szeregu skumulowanego.

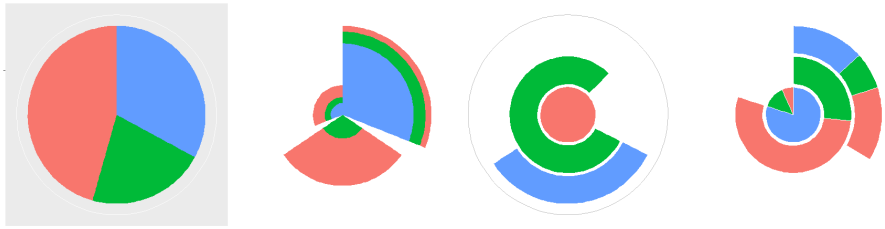
3.1.2. Wykres słupkowy



Wykres słupkowy (kolumnowy) jest wizualizacją danych, w której każda wartość jest reprezentowana przez wysokość słupka. Wykres ten może być przedstawiony także w orientacji poziomej. W takim przypadku długość słupka reprezentuje odpowiednią wartość. Słupki są rysowane obok siebie, co pozwala na porównywanie wartości różnych kategorii lub grup. Wykresów słupkowych często się używa, aby wizualnie przedstawić wyniki badań ankietowych i inne dane ilościowe, w których można wyróżnić kategorie lub grupy. Wykresy te są łatwe do zrozumienia i stanowią skuteczny sposób na przedstawienie danych w sposób jasny i czytelny. Mogą być używane do wizualizacji danych jednostkowych lub już podsumowanych.

Uprzednio przedstawiony histogram również jest wykresem słupkowym. Należy jednak zaznaczyć, że nie każdy wykres słupkowy jest histogramem. Dla wykresów słupkowych wyróżnione kategorie mają zazwyczaj charakter jakościowy lub są to zmienne ilościowe dyskretne, a w przypadku histogramu zmienna przedstawiana na wykresie jest ciągła. O ile w histogramie sąsiadujące słupki przylegają do siebie, to dla wykresów słupkowych, gdzie wyróżnione są kategorie, wskazane jest, aby pomiędzy tymi słupkami występowała pewna przerwa. Szczególną formą wykresu słupkowego jest spinogram. W przypadku tego wykresu o wielkości zjawiska informuje nie wysokość ani długość, ale szerokość słupka.

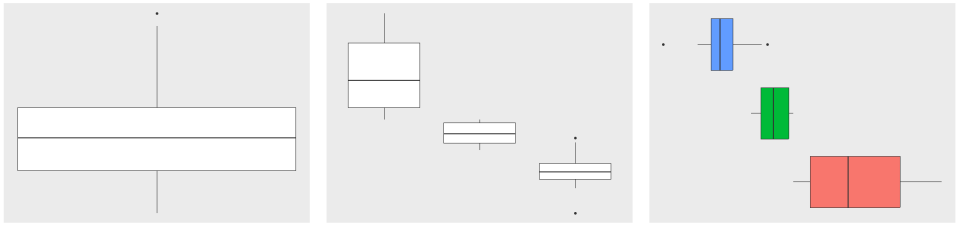
3.1.3. Wykresy kołowy i pierścieniowy



Wykres kołowy to rodzaj wykresu, z pomocą którego można pokazać proporcje lub udział poszczególnych elementów w całości. Za twórcę wykresu kołowego uznaje się Williama Playfaira (Biecek 2014). Pierwszy taki wykres został opublikowany w pracy Playfaira (1801 rok). William Playfair uważał, że wykresy przedstawiają dane znacznie lepiej niż tabele i pozwalają na szybkie przekazanie kluczowych faktów o opisywanej zbiorowości. Wykresy kołowe mogą być wykorzystane na przykład do przedstawiania proporcji podziału miejsc w parlamencie pomiędzy różne partie lub udziału różnych kategorii produktu w sprzedaży. Wielkości przedstawione na wykresie kołowym powinny się sumować do 100% lub do łącznej liczebności badanej zbiorowości. Wykres kołowy pozwala na przedstawienie struktury tylko jednej zbiorowości. Pewnym rozszerzeniem idei wykresów kołowych są wykresy pierścieniowe pozwalające na przedstawienie kilku struktur na jednym wykresie.

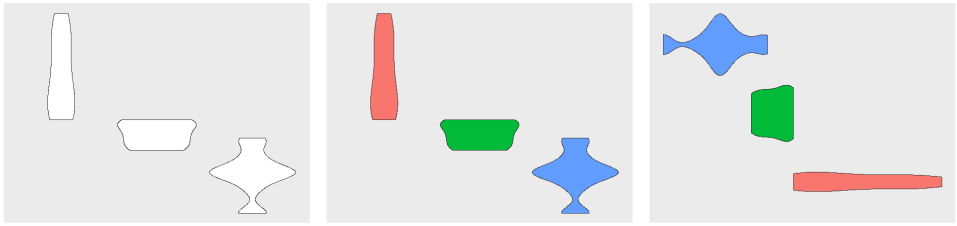
Wykres kołowy jest często wykorzystywany w prezentacjach o charakterze biznesowym lub popularnonaukowym. W analizach statystycznych zwykle lepiej odwołać się do wykresów słupkowych, ponieważ długości słupków są lepiej postrzegane od wielkości kątów. Formalnie wykres kołowy jest wykresem słupkowym wykreślonym we współrzędnych biegunowych.

3.1.4. Wykres pudełkowy



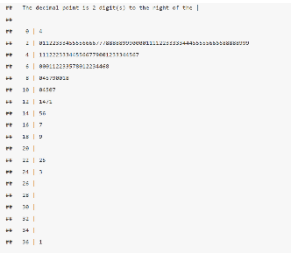
Wykres pudełkowy (inaczej: skrzynkowy, box plot lub Box-Whisker diagram) został zaproponowany przez Johna W. Tukeya (1977). Wykresu pudełkowego bardzo często używa się w analizie danych, aby przedstawić charakterystykę analizowanej zmiennej. Jest to rodzaj wykresu, który pokazuje rozkład danych mierzonych na skali mocnej. W szczególności na wykresie zaznaczone są mediana, kwartyle pierwszy i trzeci oraz wartości minimalna i maksymalna. Dodatkowo są zwykle wyróżniane wartości odstające i ekstremalne. Wykres pudełkowy składa się z pudełka, które reprezentuje obszar najczęstszych wartości, oraz dwóch „whiskerów" (wąsów) odpowiadających największej i najmniejszej wartości. Jeśli wartości są bardzo rozproszone, to można zobaczyć punkty reprezentujące wartości odstające (outlier). Wykres pudełkowy pozwala na ocenę nie tylko poziomu przeciętnego zróżnicowania badanej zmiennej, ale również kierunku i siły asymetrii. Może być wykreślany zarówno poziomo, jak i pionowo. Wykres ten może być prezentowany dla różnych kategorii jakościowych. Z tego powodu wykres pudełkowy jest bardzo pomocny przy przedstawieniu graficznego porównania charakterystyk kilku różnych zmiennych.

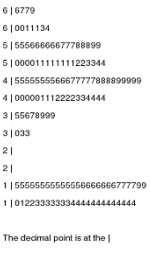
3.1.5. Wykres wiolinowy



Wykres wiolinowy (violin plot, kernel density plot), nazywany również wykresem skrzypcowym, w zakresie zastosowań jest podobny do wykresu pudełkowego. Stosuje się go dla ilościowych zmiennych ciągłych. W zasadzie stanowi pewne połączenie wykresu pudełkowego (box plot) oraz wykresu gęstości (density plot). Zamiast w formie prostokątnego pudełka rozkład zmiennej jest zobrazowany za pomocą dwóch połówek wiolinopodobnego kształtu, które przedstawiają estymator gęstości badanej zmiennej. W przeciwieństwie do wykresu pudełkowego, który może pokazać tylko statystyki zbiorcze, wykresy wiolinowe prezentują takie statystyki oraz gęstość każdej zmiennej. Połowa wiolinowa jest zwężona w miejscu, w którym znajduje się niewielka liczba obserwacji, a szeroka w miejscu, gdzie jest ich dużo. Na podstawie tego wykresu łatwo zidentyfikować takie charakterystyki, jak dominanta, mediana, rozproszenie czy asymetria rozkładu. Wykres wiolinowy może być wykreślony zarówno pionowo, jak i poziomo. Bardzo często jest wykorzystywany do porównania rozkładów kilku kategorii, które wyróżniono na podstawie zmiennej jakościowej lub numerycznej dyskretnej. Wykresu wiolinowego często używa się w analizie danych, aby pomóc w wizualizacji i interpretacji wyników, a także w porównywaniu kilku zestawów danych.

3.1.6. Wykres łodyga-liść

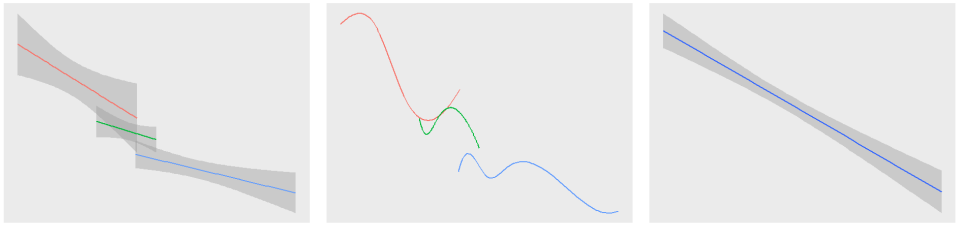




Wykres łodyga-liść (stem-and-leaf plot) to jedna z metod wizualizacji danych statystycznych, która umożliwia prezentację i analizę danych w sposób pozwalający zachować szczegółowe informacje o poszczególnych wartościach. Ten wykres to w zasadzie specyficzna tabela, w której każda wartość jest podzielona na „łodygę" (pierwszą cyfrę lub cyfry) i „liść" (zwykle ostatnią cyfrę). Wykres łodyga-liść nieco przypomina histogram, w którym słupki są rysowane poziomo. Jednak w przeciwieństwie do histogramu wykres łodyga-liść zachowuje oryginalne dane z dokładnością do co najmniej dwóch cyfr znaczących.

Wykres ten jest pomocny w wizualizacji kształtu rozkładu. Mimo że był znany już wcześniej, to powszechne zastosowania znalazł po opublikowaniu pracy Johna W. Tukeya (1977). Wykres ten dość często stosowano, gdy teksty były przygotowywane na maszynie do pisania. Obecnie, przy znacznie większych możliwościach technicznych, pojawia się już znacznie rzadziej. Wykres łodyga-liść pozwala zobaczyć rozkład danych, ocenić średnią, odchylenie standardowe i inne mierniki statystyczne. Jest to szczególnie przydatne do analizy danych o małej liczbie punktów lub do uzyskania szybkiego obrazu rozkładu danych. Konstrukcja tego wykresu nie jest wskazana przy bardzo dużej liczbie obserwacji.

3.1.7. Wykres liniowy

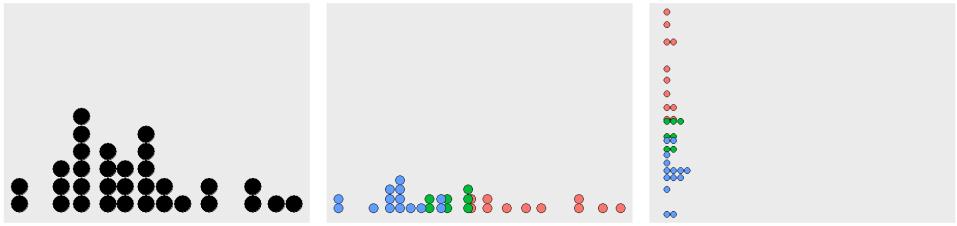


Wykres liniowy to rodzaj wykresu, który pokazuje zmiany w wartościach w czasie lub względem innej zmiennej. Punkty na wykresie są połączone liniami, co umożliwia łatwe odczytanie trendów i pojedynczych zmian w wartościach. Wykresu liniowego często używa się do prezentowania danych dotyczących wzrostu, spadku lub ogólniej: zmian zjawiska w czasie. Może być używany do wizualizacji danych historycznych, prognoz lub analizy wpływu określonej zmiennej na inną zmienną. Jest to prosty i łatwy do zrozumienia typ wykresu, który powszechnie stosuje się w różnych dziedzinach.

Na jednym wykresie liniowym może być przedstawionych kilka linii, co umożliwia przeprowadzenie porównań dotyczących na przykład zmian dla różnych obiektów (przedsiębiorstw, województw, państw). Za pomocą wykresów liniowych można przedstawić teoretyczne funkcje regresji różnej postaci (liniową, logarytmiczną, wykładniczą, wielomianową), jak również empiryczne funkcje regresji.

Wykresy liniowe były stosowane już bardzo dawno. Michael Friendly (Friendly i Denis b.r.) wskazuje, że zmiany w czasie na wykresach liniowych przedstawiał już między innymi William Playfair w XVIII wieku.

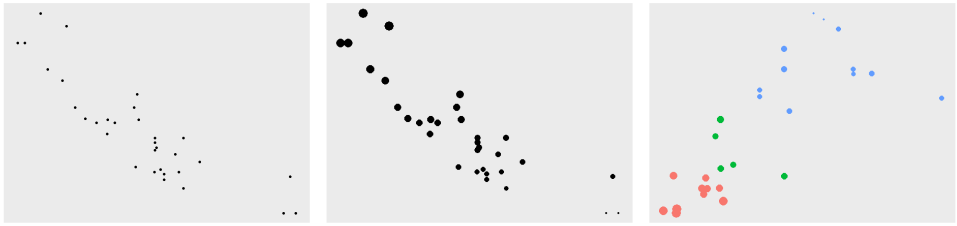
3.1.8. Wykres punktowy



Wykres punktowy lub kropkowy (dot plot) to wykres statystyczny składający się z punktów naniesionych na obszar wykresu, zazwyczaj przy użyciu wypełnionych kółek. Istnieją dwie popularne, choć bardzo różne, wersje wykresu punktowego. Pierwsza była używana już w ręcznie rysowanych (przed erą komputerów) wykresach do przedstawiania rozkładów. Druga wersja została opisana przez Williama S. Clevelanda (1993) jako alternatywa dla wykresu słupkowego, w której kropki są używane do przedstawiania wartości ilościowych (na przykład zliczeń) związanych ze zmiennymi jakościowymi.

Wykres punktowy pokazuje pojedyncze punkty danych, bez linii łączących punkty. Każdy punkt na wykresie reprezentuje jeden zestaw danych, składający się z dwóch wartości: jednej na osi OX i drugiej na osi OY. Punkty na tym wykresie są umieszczone w miejscu odpowiadającym ich wartościom na osiach OX i OY. Szczególnym przypadkiem wykresów punktowych są wykresy rozrzutu, które wykorzystuje się do oceny typu i siły zależności pomiędzy zmiennymi ilościowymi. Wykresy punktowe można także wykreślać na przykład dla trzech zmiennych na wykresach 3D, jednak postrzeganie wartości na takim wykresie staje się znacznie utrudnione. Często dla zwiększenia przejrzystości punkty są dodawane na wykresach innego typu, jak na przykład liniowych, pudełkowych, wiolinowych.

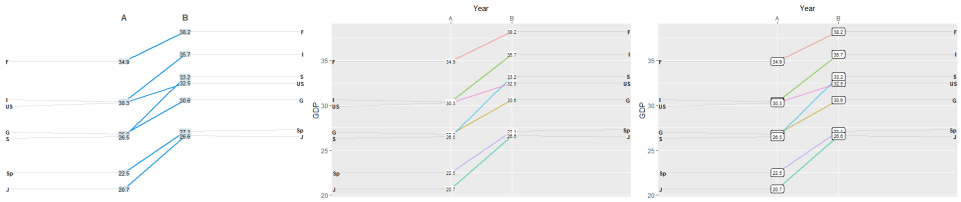
3.1.9. Wykres rozrzutu



Wykres rozrzutu (scatter plot) to szczególny przypadek wykresu punktowego. Na tym wykresie w układzie współrzędnych kartezjańskich umieszczane są punkty (kropki, wypełnione koła), których współrzędne odpowiadają wartościom dwóch zmiennych ilościowych. Na osi OX zwykle umieszcza się zmienną niezależną, a na osi OY zmienną zależną. Wykres rozrzutu pozwala na pokazanie związku między dwiema zmiennymi numerycznymi. Pomaga w identyfikacji ewentualnej zależności między dwiema zmiennymi oraz w określeniu typu związku pomiędzy nimi. Pozwala też w przypadku wystąpienia zależności liniowej na wskazanie kierunku zależności oraz odczytanie przybliżonej siły tej zależności. Na podstawie tego wykresu można uzyskać informacje o wartościach minimalnych, maksymalnych i odstających dla każdej ze zmiennych. Na tym wykresie można odczytać wartości dla wszystkich obserwacji.

Na wykresie rozrzutu współrzędne punktów odpowiadają dwóm zmiennym. Możliwe jest dodatkowo przypisanie wartości lub wariantów innych zmiennych do koloru, kształtu lub wielkości punktów na wykresie. Pozwala to na przedstawienie na jednym wykresie nie tylko dwóch, ale nawet pięciu, a niekiedy jeszcze większej liczby takich zmiennych. Wykres rozrzutu może być wykreślany dla trzech zmiennych w formie wykresu 3D, jednak taka forma zwykle utrudnia właściwy odbiór danych.

3.1.10. Wykres zmiany

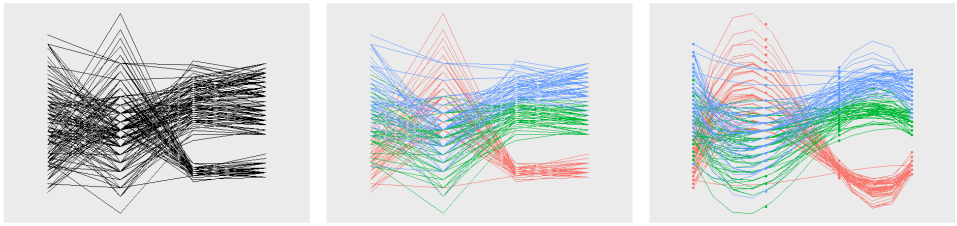


Wykres zmiany (slope plot), nazywany także wykresem różnicy, pozwala ukazać wielkość i kierunek zmian dla dwóch pomiarów tych samych obiektów (Biecek, Baranowska i Sobczyk 2019). Wykres tworzą dwie pionowe osie, po jednej dla każdego pomiaru oraz linii łączących pary pomiarów tego samego obiektu. Powiązane wartości są połączone odcinkami. Wykresu zmiany można użyć do pokazania wartości różnych zmiennych dla dwóch okresów lub stanów. Na tym wykresie można jednocześnie przedstawić zmiany dla wielu obiektów (państw, województw, przedsiębiorstw). Konstrukcja tego wykresu opiera się na wykresie liniowym, ale ważnym jego elementem są etykiety danych na końcach linii, co pozwala na szybką percepcję zmian wartości.

Na wykresie zmiany wartości mogą być przedstawiane nie dla dwóch różnych punktów czasowych, ale dla dwóch różnych kategorii jakościowych. Może to być przedstawienie na przykład dla wybranych państw poziomu bezrobocia oraz wartości PKB na osobę.

Wykres zmiany może być tworzony w różnych narzędziach do wizualizacji danych, takich jak arkusze kalkulacyjne, programy do tworzenia wykresów lub narzędzia programistyczne. Istnieje wiele wariantów i dostosowań tego typu wykresu, które pozwalają na dodanie uzupełniających informacji i cech, takich jak etykiety, kształty punktów danych.

3.1.11. Wykres współrzędnych równoległych

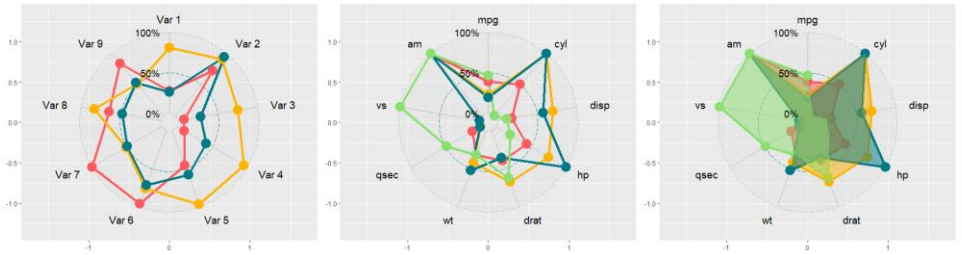


Wykres współrzędnych równoległych (parallel coordinates) może być traktowany jako rozszerzenie wykresu zmiany na większą liczbę okresów, zmiennych lub stanów. Wykres ten (Inselberg 1999) pozwala na jednoczesne wyświetlenie wartości znacznej liczby zmiennych ciągłych nawet dla wielu obiektów. Pozwala na porównanie cechy kilku pojedynczych obserwacji (serii) na zbiorze zmiennych liczbowych. Każdy pionowy słupek reprezentuje zmienną i często ma swoją własną skalę. Jednostki dla poszczególnych zmiennych mogą być różne. Wartości są następnie wykreślane jako serie linii połączonych w poprzek każdej osi. Ustalona linia reprezentuje wartości poszczególnych zmiennych dla określonego obiektu.

Wykres współrzędnych równoległych okazuje się szczególnie przydatny w identyfikowaniu wzorców, trendów i zależności w danych wielowymiarowych. Pozwala na zobrazowanie, jak zmienne są ze sobą skorelowane, czy istnieją grupy lub klastry obserwacji o podobnych wartościach oraz jakie są zakresy zmienności dla poszczególnych zmiennych.

Podstawowe zastosowania wykresu współrzędnych równoległych to: analiza danych statystycznych, badanie zbiorów danych wielowymiarowych, eksploracja danych i wykrywanie nietypowych wzorców lub obserwacji odstających.

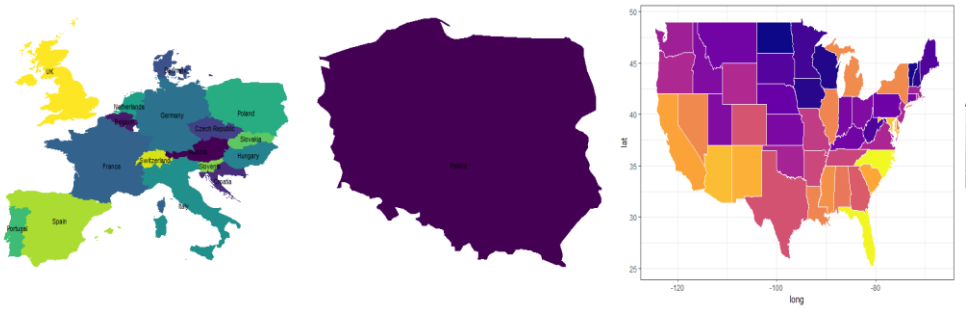
3.1.12. Wykres radarowy



Wykres radarowy (radar plot, spider plot, wykres gwiazdowy, wykres polarny) jest graficzną prezentacją dla danych wielowymiarowych w postaci dwuwymiarowego wykresu trzech lub więcej zmiennych ilościowych reprezentowanych na osiach wychodzących z tego samego punktu. Wykres radarowy stanowi pewną odmianę wykresu współrzędnych równoległych. Wykres współrzędnych równoległych jest wykreślony w kartezjańskim układzie kilku współrzędnych, a wykres radarowy wykreślony w układzie współrzędnych biegunowych. Podstawowa różnica pomiędzy tymi wykresami polega na tym, że w przypadku wykresu współrzędnych równoległych wszystkie współrzędne są równoległe, a w przypadku wykresu radarowego wszystkie współrzędne wychodzą z jednego, centralnego punktu.

Wykresy radarowe to użyteczny sposób wyświetlania obserwacji wielowymiarowych. Każda zamknięta linia (gwiazda) reprezentuje pojedynczą obserwację. Na wykresie radarowym można przedstawić jednocześnie wiele obiektów (gwiazd). Wykres radarowy ma pewne ograniczenia. Jego stosowanie jest zalecane do porównywania małej liczby zmiennych i obserwacji, ponieważ przy zbyt wielu zmiennych czy obserwacjach może stać się trudny do interpretacji. Ponadto równomierne skalowanie osi w przypadku różnych zmiennych może utrudniać porównywanie wartości.

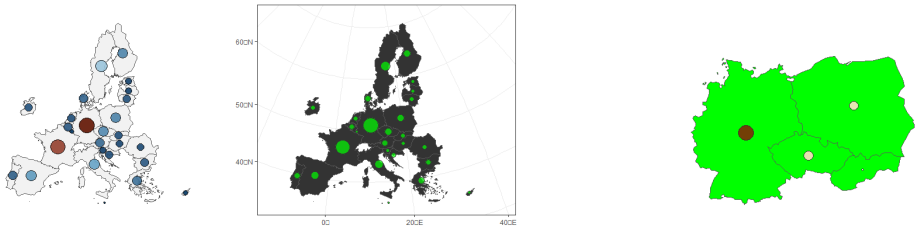
3.1.13. Wykres mapowy (kartogram)



Analiza danych geograficznych związana jest z wieloma trudnościami wynikającymi z występowania zależności przestrzennych. Właściwe graficzne przedstawienie takich danych może w tych przypadkach być bardzo pomocne. Graficzną prezentację danych przestrzennych umożliwiają wykresy mapowe (map plot, kartogram). Wykres mapowy wykorzystuje mapę geograficzną do prezentacji danych. Wykreślane są geograficzne kontury, granice, punkty i inne elementy mapy do przedstawienia stosownych informacji. Mogą to być dane dotyczące takich zmiennych jak na przykład populacja, gęstość zaludnienia, wskaźniki ekonomiczne lub inne informacje przypisane do konkretnych obszarów geograficznych, takich jak państwa, stany, województwa, powiaty lub miasta. Jeżeli do przedstawiania danych na wykresie mapowym wykorzystuje się słupki, koła, linie lub różne symbole graficzne, to otrzymuje się kartodiagram. Pozwala to wizualnie wyrazić poziom zjawiska lub wielu zjawisk, ich zmiany oraz zależności na określonych obszarach.

Wykresy mapowe mają wiele specyficznych zastosowań. Powszechnie są wykorzystywane w naukach społecznych, badaniach rynku, planowaniu przestrzennym oraz analizie danych biznesowych. Pozwalają na łatwe zrozumienie i wizualizację danych w kontekście geograficznym, co może pomóc w identyfikacji wzorców, tendencji i zależności w danych.

3.1.14. Kartodiagram

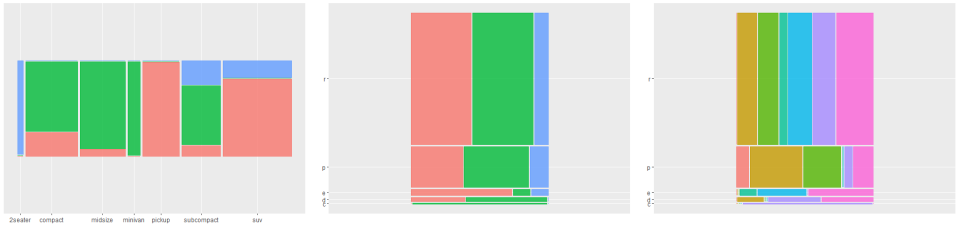


Kartodiagram to jedna z kartograficznych metod prezentacji, która jest mapą tematyczną przedstawiającą zmienność wybranych atrybutów obiektów przestrzennych za pomocą umieszczonych na niej punktów, diagramów lub wykresów. Ich lokalizacja odpowiada punktom pomiaru lub jednostkom przestrzennym, do których dane się odnoszą (na przykład miasta, województwa, powiaty). Na tej mapie przedstawiane są dane za pomocą wybranych symboli, punktów, linii, kół, histogramów bądź innych wykresów. Wielkość lub kolor odnoszą się do ustalonej wartości zmiennej geograficznej. Kartodiagramy są wykorzystywane do wizualizacji danych przestrzennych, takich jak rozkład populacji, zagęszczenie zdarzeń, rozkład badanych zmiennych.

Kartodiagramy pozwalają na łatwe porównywanie wielkości zjawiska między różnymi regionami, wygodną prezentację zależności przestrzennych, a także identyfikację obszarów z wyróżniającymi się wartościami badanych zmiennych.

Istnieje wiele rodzajów kartodiagramów, podobnie jak wiele jest typów diagramów i wykresów, które można umieścić na mapie: liniowe – wstęgowe i wektorowe, słupkowe, kwadratowe, kołowe, przestrzenne; a także proste, strukturalne.

3.1.15. Wykres mozaikowy



Wykresy mozaikowe (mosaic plot) pozwalają na prezentację danych z dwu- lub wielowymiarowych tablic wielodzielczych. Na tym wykresie obszar graficzny jest podzielony na prostokąty o rozmiarach proporcjonalnych do liczby obserwacji dla kombinacji zmiennych, które reprezentują. Wewnętrzne prostokąty odpowiadają podkategoriom, a ich położenie i kształt zależą od wielkości i proporcji podkategorii względem nadrzędnej kategorii. Kategorie danych są zwykle oznaczone różnymi kolorami, które pomagają w łatwej identyfikacji i porównywaniu kategorii. Wykres pozwala na przegląd danych i umożliwia rozpoznanie związków pomiędzy różnymi zmiennymi. Wykresy mozaikowe są przydatne w wizualizacji danych składających się z wielu poziomów kategorii i podkategorii. Pozwalają uzyskać szybki przegląd wielu danych na jednym wykresie i łatwo zauważyć proporcje oraz relacje między wyróżnionymi kategoriami.

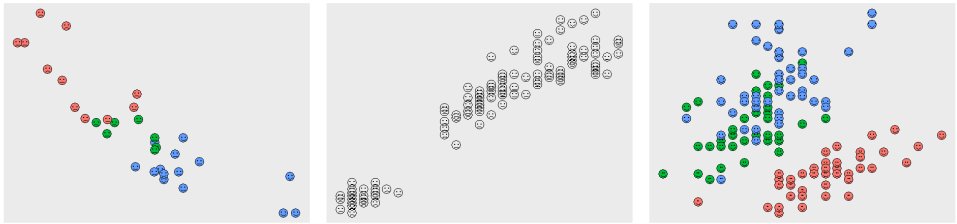
Dla dwóch zmiennych wykres mozaikowy stanowi dość specyficzny rodzaj wykresu słupkowego. W takim przypadku szerokość kolumn jest proporcjonalna do liczby obserwacji na każdym poziomie zmiennej wykreślonej na osi poziomej. Pionowa długość słupków jest proporcjonalna do liczby obserwacji drugiej zmiennej na każdym poziomie pierwszej zmiennej. Wykresy mozaikowe pomagają pokazać zależności i umożliwiają wizualne porównywania grup.

3.1.16. Wykres obrazkowy

Wykres obrazkowy (pictorial chart) to rodzaj wykresu, który używa obrazków lub ikon do reprezentacji danych i informacji. Zamiast tradycyjnych elementów graficznych, takich jak słupki czy linie, wykres obrazkowy wykorzystuje symbole, ikony, piktogramy, a nawet ilustracje. Taka forma przekazu pozwala łatwo rozpoznać przekazywane specyficzne informacje. Wykresy obrazkowe są przydatne przy prezentacji danych ilościowych jak liczba produktów, liczba osób czy udział procentowy określonego wariantu w całości. Pozwalają w sposób bardziej interesujący i dobrze zrozumiały dla odbiorców przekazać informację o skali zjawiska. W przeciwieństwie do innych typów wykresów, takich jak wykres słupkowy czy kołowy, wykres obrazkowy nie pokazuje wartości w skali, ale jedynie względne proporcje między elementami danych. Wykresy te są najbardziej efektywne, gdy ilość danych jest stosunkowo niewielka, a obrazki są czytelne i z łatwym do zrozumienia przekazem.

Wykresy obrazkowe mogą być tworzone ręcznie bądź przy użyciu narzędzi do grafiki lub specjalistycznych narzędzi i oprogramowania do wizualizacji danych. Wykresy te są stosunkowo rzadko wykorzystywane w typowych analizach statystycznych.

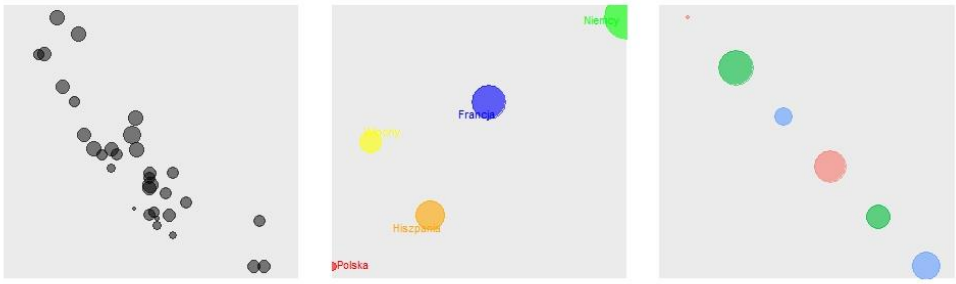
3.1.17. Twarze Chernoffa



Twarze Chernoffa (Chernoff faces), zaproponowane przez matematyka, statystyka i fizyka Hermana Chernoffa w 1973 roku, prezentują wielowymiarowe dane w kształcie ludzkiej twarzy. Poszczególne części, takie jak oczy, uszy, usta i nos, reprezentują wartości zmiennych poprzez swój kształt, rozmiar, rozmieszczenie i orientację. Ideą wykorzystania twarzy jest to, że człowiek łatwo rozpoznaje twarze i bez trudu zauważa zachodzące w nich niewielkie zmiany. Wykresy twarzy Chernoffa obsługują każdą zmienną w inny sposób. Ponieważ cechy twarzy różnią się pod względem postrzeganej ważności, sposób mapowania zmiennych na cechy powinien być starannie dobrany (na przykład stwierdzono, że rozmiar oczu i nachylenie brwi mają znaczącą wagę w postrzeganiu). Na wykresie twarzy Chernoffa każda cecha twarzy, taka jak kształt i rozmiar nosa, ust, oczu i tym podobne, jest zmapowana na jedną z wielu cech statystycznych, takich jak średnia, odchylenie standardowe lub kwantyle. Dla przykładu, wielkość nosa może być związana z wartością średniej dla danej zmiennej, a kształt ust z odchyleniem standardowym. W ten sposób wizualnie zrozumiały obraz twarzy przedstawia informacje statystyczne, co umożliwia łatwe i szybkie porównywanie danych.

Wykresy twarzy Chernoffa są często stosowane w wielu dziedzinach, takich jak badania marketingowe, finanse, biometria i inne, gdzie ważne jest szybkie i wizualne porównywanie dużych ilości danych.

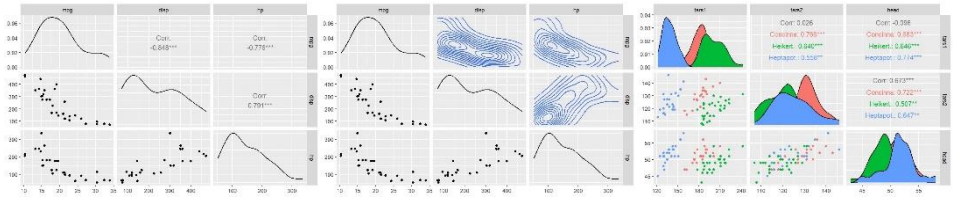
3.1.18. Wykres bąbelkowy



Wykres bąbelkowy (bubble chart) to szczególna forma wykresu rozrzutu. Na tym wykresie prezentowane są trzy zmienne. Pierwsze dwie zmienne (*x* i *y*) są mapowane na osie OX i OY tak jak przy konstrukcji wykresu rozrzutu. Trzecia zmienna jest połączona z wielkością kropki (bąbelka) na tak skonstruowanym wykresie. Możliwe jest także wprowadzenie dodatkowych wymiarów (zmiennych) związanych na przykład z kształtem lub kolorem punktów rozmieszczonych na wykresie.

Wykresy bąbelkowe znalazły szczególnie interesujące zastosowanie w wizualizacjach gapminder zaproponowanych przez Hansa Roslinga w ramach projektu Gapminder Foundation (Gapminder b.r.). W tych wizualizacjach w sposób dynamiczny i interaktywny prezentowane są zmiany w czasie dla różnych wskaźników społeczno-gospodarczych na całym świecie. Kolorem na tych prezentacjach oznacza się położenie geograficzne państw. Wykresy takie są szeroko stosowane w celu pokazywania trendów rozwojowych, takich jak PKB per capita, oczekiwana długość życia, stopa urodzeń i wiele innych, w zależności od dostępnych danych.

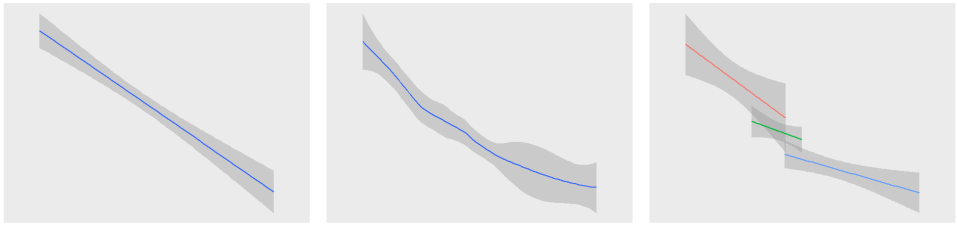
3.1.19. Macierzowy wykres rozrzutu



Macierzowe wykresy rozrzutu (scatter plot matrices) są naturalnym rozszerzeniem wykresów rozrzutu. O ile wykres rozrzutu jest konstruowany na podstawie dwóch zmiennych numerycznych, to macierzowy wykres rozrzutu pozwala na przedstawienie zależności pomiędzy większą liczbą takich zmiennych. Panele w macierzy pokazują wykresy rozrzutu dla wszystkich par zmiennych. Na wykresie dane są przedstawiane w postaci siatki, gdzie każda komórka siatki odpowiada parze zmiennych. Wykres taki pozwala na określenie, czy istnieje korelacja liniowa pomiędzy różnymi parami zmiennych. Staje się to szczególnie pomocne przy wskazywaniu konkretnych zmiennych, które mogą mieć istotne korelacje z innymi. Niektóre odmiany macierzowego wykresu rozrzutu pokazują tylko górny lub dolny trójkąt paneli, ponieważ w przeciwnym razie po drugiej stronie przekątnej pojawiają się (transponowane) te same układy par punktów. Na głównej przekątnej w zależności od rodzaju wykresu prezentowane mogą być na przykład nazwy poszczególnych zmiennych, histogramy lub oszacowania gęstości tych zmiennych.

Wykresy rozrzutu w układzie macierzowym mogą zawierać dodatkowe elementy, takie jak wartości współczynników korelacji, linie trendu, elipsy dopasowania lub kolorowanie punktów według innej zmiennej, co pozwala przedstawić na takim wykresie uzupełniające informacje.

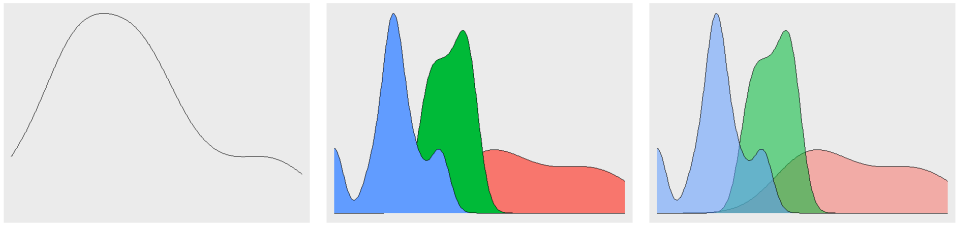
3.1.20. Wykres regresji



Wykres regresji (regression plot) wykorzystuje się w analizie statystycznej do wizualizacji zależności między dwiema zmiennymi. Przedstawia punkty danych na wykresie rozrzutu (scatter plot) i dodaje do niego linię regresji, która ilustruje trend lub wzorzec w danych. Linia regresji na wykresie jest tworzona na podstawie dopasowania modelu regresji do danych. Może to być prosty model regresji liniowej, w którym wykreślaną linią jest najlepiej dopasowana prosta do punktów danych, lub bardziej złożone modele regresji, takie jak regresja wielomianowa, regresja wykładnicza, logistyczna i tym podobne. Linia regresji reprezentuje ogólną zależność w danych i pozwala na przewidywanie wartości zmiennej objaśnianej na podstawie zmiennej objaśniającej.

Wykres regresji dostarcza informacji na temat siły, kierunku i istotności zależności między zmiennymi. Jeśli linia prosta regresji jest nachylona w górę, oznacza to dodatnią korelację między zmiennymi, czyli wzrost jednej zmiennej wiąże się ze wzrostem średniej wartości drugiej zmiennej. W przypadku nachylenia w dół mamy do czynienia z ujemną korelacją, gdzie wzrost jednej zmiennej łączy się ze spadkiem przeciętnych wartości drugiej zmiennej. Położenie danych punktów względem linii regresji może wskazywać na siłę zależności.

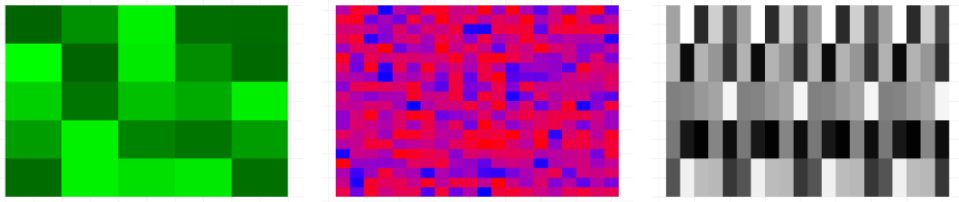
3.1.21. Wykres funkcji gęstości



Wykres funkcji gęstości (density plot) wykorzystuje się do wizualizacji rozkładu danych i prezentowania oszacowania funkcji gęstości prawdopodobieństwa. Jest szczególnie przydatny, gdy należy zobrazować kształt, skośność, asymetrię lub inne właściwości rozkładu danych. Wykres funkcji gęstości wykorzystuje nieparametryczną ocenę krzywej gęstości, która reprezentuje względną częstość występowania różnych wartości danych. Krzywa gęstości może być oparta na różnych modelach, takich jak rozkład normalny, rozkład jednostajny, rozkład gamma i tym podobne, w zależności od charakterystyki danych. Na wykresie funkcji gęstości oś pozioma reprezentuje wartości danych, a oś pionowa – wartości funkcji gęstości. Im wyższa wartość funkcji gęstości w danym miejscu, tym większe prawdopodobieństwo wystąpienia wartości w tym obszarze.

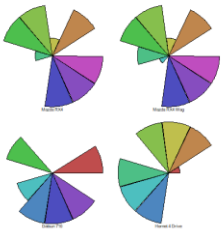
Wykres funkcji gęstości pozwala na porównywanie rozkładów między różnymi grupami lub kategoriami danych. Może być stosowany do porównywania jednocześnie wielu rozkładów, na przykład rozkładów wyników testów dla różnych grup uczniów lub rozkładów wynagrodzeń pracowników na różnych stanowiskach. Wykres taki umożliwia również przedstawienie porównania rozkładów empirycznych z określonymi postaciami rozkładów teoretycznych.

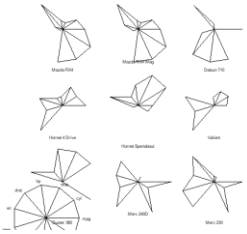
3.1.22. Wykres ciepła

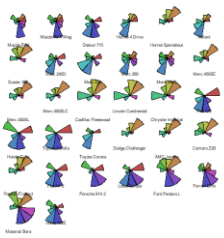


Wykres ciepła, znany również jako heatmap, jest używany do wizualizacji danych, które można przedstawić na siatce dwuwymiarowej. Staje się szczególnie przydatny, gdy należy zobrazować złożone zależności między dwiema zmiennymi oraz dodatkową trzecią zmienną, która na wykresie będzie reprezentowana przez dany kolor. Wykresy te mogą być pomocne przy graficznej prezentacji danych tabelarycznych, w szczególności na przykład macierzy współczynników korelacji. Mogą na nich być prezentowane dane z tabel o niewielkich wymiarach, ale również z tablic o bardzo dużych wymiarach. Są one skutecznym narzędziem wizualizacji wielowymiarowych szeregów czasowych. Mogą być z powodzeniem stosowane w kontekście przestrzennym. W tym przypadku mapy ciepła mogą być używane do przedstawienia rozkładu danych na mapie, gdzie intensywność koloru wskazuje na zagęszczenie danych w danym obszarze. Ważną zaletą tych wykresów jest możliwość ich wykorzystania do porównywania danych, gdzie na przykład kolejne „wiersze mapy” będą reprezentowały porównywane obiekty.

3.1.23. Wykres gwiazdowy



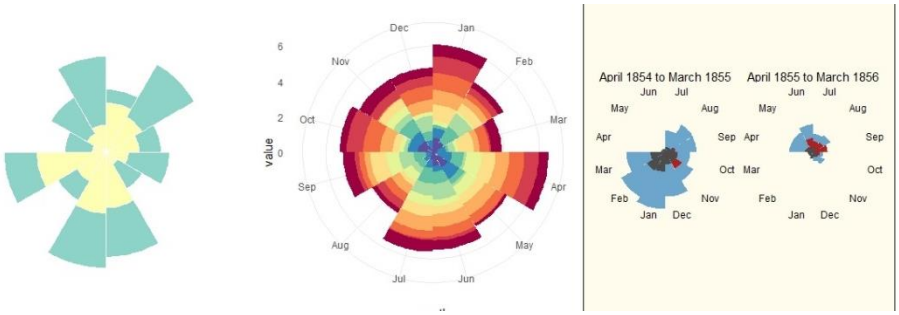




Wykres gwiazdowy (star plot) jest nieco podobny do wykresu radarowego, który wykorzystuje się do wizualizacji danych wielowymiarowych. Wykres ten składa się z osi promieniowych, z których każda odpowiada jednej zmiennej. Na każdej osi zaznacza się wartości danej zmiennej dla obserwowanego obiektu. Punkty na osiach są następnie łączone liniami, tworząc kształt gwiazdy. Każda obserwacja jest reprezentowana przez wykres kształtem przypominający gwiazdę, w którym każdy promień przedstawia jedną zmienną. Formy graficzne gwiazd mogą być dość różnorodne.

Wykres gwiazdowy okazuje się szczególnie przydatny w przypadku porównywania wielu obserwacji ze względu na kilka zmiennych. Pozwala on na łatwe porównanie wartości każdej zmiennej dla każdej obserwacji. W przypadku, gdy wykres star plot składa się z *n* promieni, każdy z nich reprezentuje jedną zmienną. Wartości każdej zmiennej są reprezentowane przez długość promienia, a kąt między promieniami odpowiada kolejnym zmiennym.

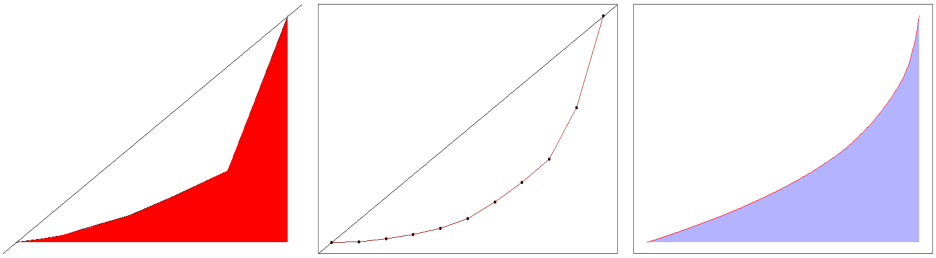
3.1.24. Wykres róża Nightingale



Wykres róża Nightingale został zaproponowany przez Florence Nightingale w 1858 roku. Znany jest również jako wykres grzebieniowy (coxcomb). Stanowi jeden z najbardziej znanych przykładów wczesnej wizualizacji danych. Nightingale, pielęgniarka, wykorzystała ten wykres do przedstawienia śmiertelności żołnierzy podczas wojny krymskiej w latach 1853-1856 (por. punkt 1.2.3 niniejszej monografii). Zastosowanie wizualizacji danych o śmiertelności żołnierzy miało znaczący wpływ na dostrzeżenie przyczyn zgonów i efekcie na zauważalną poprawę zdrowia leczonych żołnierzy.

Wartości liczbowe są reprezentowane przez pole powierzchni segmentów, a nie ich długość promienia, co stanowi różnicę w stosunku do klasycznych wykresów kołowych. Wykres róża Nightingale ma formę koła podzielonego na sektory, reprezentujące kolejne miesiące roku. Każdy sektor jest dalej podzielony na sekcje, z których każda reprezentuje daną przyczynę zgonu. Kolorowanie sektorów odpowiada różnym przyczynom zgonów. Każda przyczyna jest przypisana do określonego koloru, co ułatwia identyfikację dominujących źródeł śmiertelności. Długość sektorów wskazuje na liczbę zgonów w danym miesiącu z powodu danej przyczyny. Im dłuższy sektor, tym więcej zgonów.

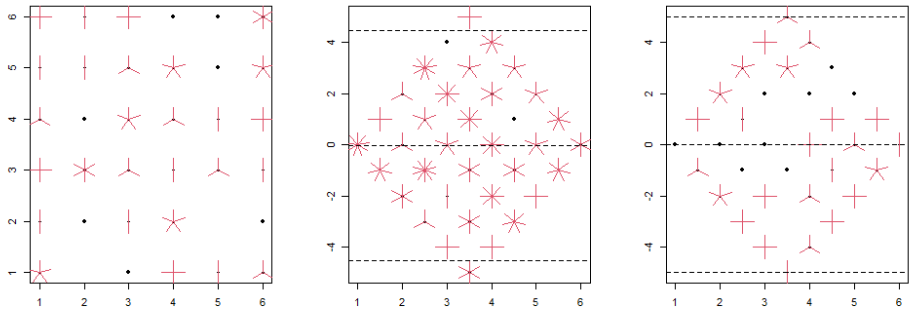
3.1.25. Krzywa Lorenza



Krzywa Lorenza, znana również jako krzywa koncentracji (nierównomierności podziału globalnego zasobu cechy), jest wykorzystywana do ilustrowania nierównomierności rozkładu dochodów w społeczeństwie. To krzywa wypukła, która jest wykreślana w kwadracie jednostkowym, i stanowi ilustrację sposobu, w jaki dochody są rozłożone w społeczeństwie. Końce tej krzywej to dolny lewy i górny prawy wierzchołek kwadratu jednostkowego.

Krzywą Lorenza najczęściej wykorzystuje się do opisu stopnia koncentracji (nierównomierności podziału globalnego zasobu cechy) jednowymiarowego rozkładu zmiennej losowej o wartościach nieujemnych. Najczęściej w praktyce badań ekonomicznych są to badania związane z pomiarem nierównomierności zarobków w badanym kraju. Może być też stosowana do określenia skali koncentracji (nierównomiernego podziału) także innych zmiennych, jak na przykład rozmieszczenie ludności w zależności od wielkości miast, pomiar nierównomierności wzrostu gospodarczego, analiza nierówności w zaopatrzeniu zdrowotnym czy też analiza nierównomierności rozkładu lokat terminowych w banku.

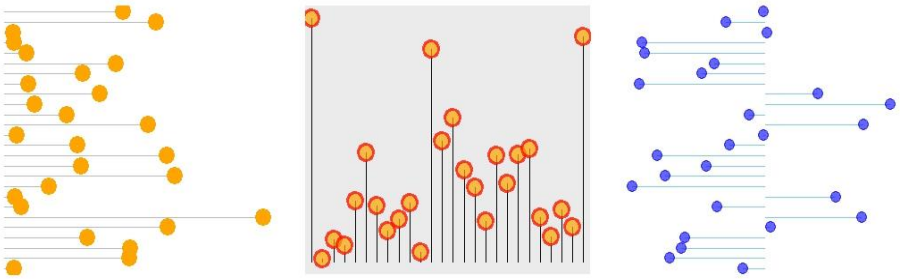
3.1.26. Wykres słonecznikowy



Wykres słonecznikowy (sunflower plot) to specyficzna forma wykresu punktowego. Jest on szczególnie przydatny do zobrazowania na wykresie obserwacji, które leżą dokładnie w tym samym miejscu. W takim przypadku punkty się na siebie nakładają i dla odbiorcy nie jest widoczne, ile obserwacji reprezentuje ustalony punkt. Wykres słonecznikowy rozwiązuje ten problem poprzez reprezentowanie wielokrotnych punktów danych jako „słoneczników" z wieloma „płatkami". Na wykresie słonecznikowym płaszczyzna OXY jest podzielona na siatkę regularnych prostokątów, a słonecznik jest umieszczony na przecięciach linii. Każdy płatek na wykresie słonecznikowym reprezentuje obserwację. Niektóre wersje wykresu słonecznikowego krotność wystąpienia danej obserwacji przedstawiają nie w formie wielu płatków, a za pomocą wielkości „tarczy słonecznika".

Z wykorzystaniem tego typu wykresu możliwe staje się wykonanie wykresu rozrzutu dla danych, gdzie obie zmienne są dyskretne. Może to być na przykład wykres dla reprezentacji 100-krotnego rzutu dwiema sześciennymi kostkami do gry.

3.1.27. Wykres lizakowy



Wykres lizakowy (lollipop chart) łączy w sobie elementy wykresu słupkowego (lub kolumnowego) oraz wykresu punktowego. Wizualnie przypomina słupki (lizaki) zakończone kropkami na osiach Y, które reprezentują wartości punktów danych. Wykresy lollipop są używane w różnych dziedzinach, aby porównać wielkości różnych kategorii danych w sposób bardziej wyraźny niż standardowy wykres słupkowy. Są szczególnie przydatne, gdy chcemy skupić się na konkretnych wartościach, ale jednocześnie zachować kontekst ogólnego rozkładu zmiennych.

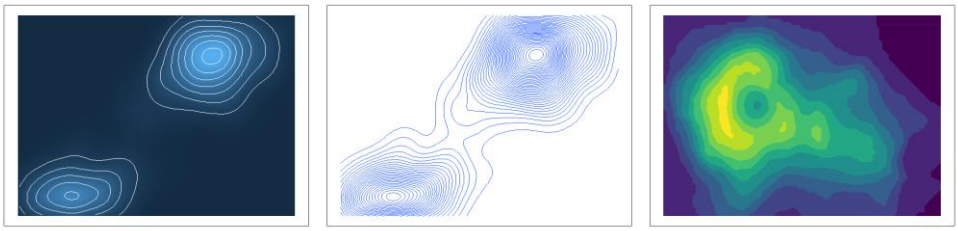
Wykres lizakowy zazwyczaj zawiera zmienne jakościowe na osi Y mierzone względem drugiej (ciągłej) zmiennej na osi X. Podobnie jak w przypadku wykresu punktowego główny nacisk kładziony jest na kropkę, aby zwrócić uwagę odbiorcy na konkretną wartość osi X osiągniętą dla każdej kategorii. Wykres lollipop w sposób bardzo czytelny prezentuje dane liczbowe. Zasadniczo zachowuje te same funkcje, co wykresy słupkowe (kolumnowe). Daje możliwości porównywania wielkości pomiędzy różnymi kategoriami. Stanowi dość atrakcyjną formę przekazu. Może być wykorzystany do przedstawienia zmian zjawiska w czasie.

3.1.28. Piramida wieku

Wykres piramida wieku (back-to-back histogram) to rodzaj wykresu demograficznego, który przedstawia strukturę populacji ze względu na wiek oraz płeć. Jest to graficzna reprezentacja procentowego udziału poszczególnych grup wiekowych w populacji, zazwyczaj przedstawiana w formie dwóch kolumn: jedna dla mężczyzn i druga dla kobiet. Piramida wieku umożliwia analizę proporcji osób między płciami oraz rozkładu wieku w populacji, co okazuje się istotne dla oceny dynamiki demograficznej i planowania społecznego. Na osi poziomej zazwyczaj umieszcza się procent lub bezwzględne wartości reprezentujące udział poszczególnych grup wiekowych w całej populacji. Na osi OX w prawo od punktu zerowego wyznaczane są liczebności kobiet, natomiast w lewo – mężczyzn. Osie pionowe reprezentują kolejne roczniki wieku lub wyróżnione grupy wiekowe, na przykład po 5 lub 10 lat.

Wykres back-to-back histogram może mieć także inne, nietypowe zastosowania wykraczające poza zagadnienia demograficzne. Generalnie powinny być dwie wyróżnione grupy (klasycznie jest to płeć) i ich struktura ze względu na pewną zmienną porządkową (klasycznie są to grupy wiekowe).

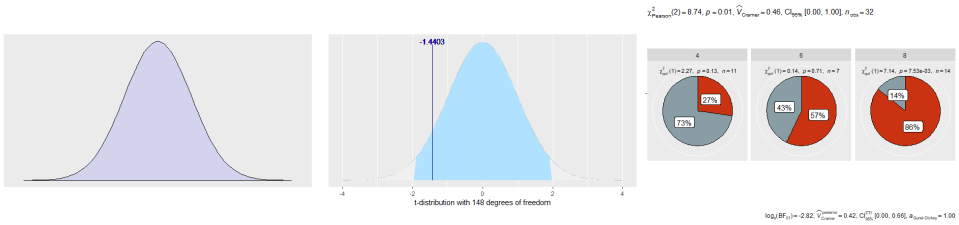
3.1.29. Wykres konturowy



Wykres konturowy (contour plot) to graficzna reprezentacja funkcji dwuwymiarowej, gdzie wartości funkcji są przedstawiane za pomocą linii łączących punkty o tej samej wartości. Wykres konturowy pokazuje „kształt" powierzchni utworzonej przez funkcję. Wykresy te są łatwe w interpretacji, pozwalają na szybką identyfikację punktów, dla których funkcja przyjmuje zbliżone wartości. Umożliwiają przedstawienie wizualizacji nawet złożonych funkcji dwuwymiarowych. O ile wykres trójwymiarowy przedstawia funkcję za pomocą trójwymiarowego kształtu, to wykres konturowy prezentuje funkcję za pomocą linii (konturów) na płaszczyźnie, gdzie te linie reprezentują punkty o takiej samej wartości funkcji (poziomy). Im linie są bliższe, tym większe zmiany wartości obserwowanej funkcji są w tym obszarze.

Wykresy konturowe powszechnie wykorzystuje się w geografii, meteorologii, fizyce, a także w inżynierii. Wykresy konturowe są bardzo często stosowane do graficznego przedstawienia topografii terenu, wzorców pogodowych oraz innych danych, które można przedstawić za pomocą ciągłej powierzchni. Linie na wykresie łączą punkty o równej wartości (na przykład wysokość nad poziomem morza, punkty o jednakowej wartości ciśnienia atmosferycznego czy o tej samej temperaturze powietrza), co pozwala obserwatorowi interpretować kształt i wzorce danych.

3.1.30. Wykres z wynikami wnioskowania



Wykresy statystyczne zdecydowanie bardziej kojarzą się z metodami statystyki opisowej niż z zagadnieniami wnioskowania statystycznego. Metody graficzne mogą być jednak bardzo pomocne przy estymacji parametrów populacji i przy weryfikacji hipotez statystycznych. Wykorzystując dobrze dobrane grafiki, można przedstawić podstawowe charakterystyki zmiennych losowych za pomocą wykresów funkcji gęstości lub dystrybuanty. Można również ukazać porównania charakterystyk z próby z rozkładami teoretycznymi.

Od dość dawna w analizach statystycznych wykorzystywane są wykresy qq-plot i qq-norm jako pomoc w testowaniu postaci rozkładu badanej zmiennej. Coraz częściej również inne formy wykresów są stosowane do wspomagania wnioskowania statystycznego o parametrach zmiennych losowych (Kończak 2020). Przy przeprowadzeniu w odpowiednim programie testu statystycznego na wykresie mogą być prezentowane podstawowe charakterystyki, a dodatkowo oznaczane wartości statystyki testowej, wartości krytyczne, obszar krytyczny oraz *p*-wartość. Przyjęcie takiego rozwiązania pozwala statystykowi nie tylko na ostateczne podjęcie decyzji związanej ze zweryfikowaną hipotezą, ale również na głębsze wyrobienie zdania dotyczącego analizowanego problemu.

3.2. Podstawowe zastosowania wykresów

W poprzednim punkcie przedstawiono zwięzłe charakterystyki najczęściej spotykanych typów wykresów. Nie można tego zestawienia traktować jako kompletny wykaz, ponieważ różnorodność typów wykresów jest bardzo duża i cały czas pojawiają się w tym zakresie nowe propozycje. Bardzo często będą to różne wykresy specjalistyczne, które okazują się szczególnie przydatne w analizach określonych zagadnień. Jednym z takich przykładów wykresów, które nie zostały omówione w poprzednim podpunkcie, są wykresy świecowe stosowane w analizach giełdowych. Danyel Fisher i Mariah Meyer (2018) wskazują między innymi na wykresy sieciowe (network visualizations), wykresy drzewa (tree view) czy chmury punktów (word cloud). W niniejszym opracowaniu nie są podejmowane kwestie związane z zagadnieniami, dla których wskazane są takie wykresy, dlatego nie ujęto ich w poprzednim punkcie.

Wykorzystując wykresy we wstępnej analizie danych, należy zawsze zwrócić uwagę na skalę pomiarową, na jakiej rejestrowana jest dana zmienna. Niektóre z wykresów wymagają, aby pomiary były dokonywane na skalach mocnych (przedziałowa lub ilorazowa), a inne z kolei determinują wyróżnienie odpowiednich kategorii. W tym drugim przypadku może to być skala nominalna lub porządkowa, a jeśli dokonano pomiarów na skalach mocnych, należy pogrupować obserwacje i wyodrębnić właściwe klasy. W przypadku wyróżniania klas w zbiorze danych możliwe są różne podziały. W konsekwencji mogą one prowadzić do bardzo różnych wyników, o bardzo innej wymowie i radykalnie innych prezentacji graficznych. To od konstruującego taki wykres należy oczekiwać, by prezentacja właściwie oddawała rzeczywisty charakter badanej zmiennej.

3.2.1. Zastosowania wykresów według ich typów

Różne rodzaje wykresów są używane w zależności od rodzaju danych, w tym skali pomiarowej, ich struktury oraz celu wizualizacji. Wykresy liniowe służą zwykle do prezentowania zmian zjawiska w czasie, wykresy słupkowe do porównywania wartości dla różnych kategorii, a wykresy kołowe do przedstawiania struktury składników pewnej całości. Wykresy rozrzutu umożliwiają analizę związków pomiędzy dwiema zmiennymi, a wykresy radarowe pozwalają na porównywanie struktur ze względu na wiele zmiennych. Wykresy pudełkowe prezentują rozkład analizowanych zmiennych, wskazując między innymi wartości odstające. Kartogramy są przydatne w analizie danych przestrzennych.

Zazwyczaj określony zbiór danych lub analizowane zjawisko można przedstawić graficznie na wiele różnych sposobów, gdzie każda z tych prezentacji będzie umożliwiała przekazanie dodatkowych specyficznych informacji dla odbiorcy. Wybór odpowiedniego typu wykresu staje się kluczowy dla skutecznej prezentacji danych, a zrozumienie charakteru danych i celu wizualizacji pomaga wybrać najbardziej adekwatny wykres, który ułatwi odbiorcom przyswojenie przedstawianych informacji.

W tabeli 3.1 wskazano najczęstsze zastosowania wybranych typów wykresów. Zwrócono również uwagę na szczególne wymagania dotyczące skali pomiarowej, o ile takie przy danym typie wykresu występują. Przedstawiono również zwięzłą charakterystykę związaną z zastosowaniami danego rodzaju wykresu.

Tabela 3.1. Wybrane rodzaje wykresów i ich typowe zastosowania

Typ wykresu	Zastosowania	Charakterystyka
1	*2*	*3*
Słupkowy	Rozkład zmiennej dyskretnej (ilościowej lub jakościowej)	– pozwala na szybką analizę struktury, – umożliwia porównania kilku zbiorowości, – sporządzany jest dla danych dyskretnych (ilościowych lub jakościowych), – podstawowe warianty: pionowe, poziome, 3D
Histogram	Estymacja gęstości rozkładu zmiennej ciągłej	– szczególna forma wykresu słupkowego, – pozwala na ocenę struktury jednej zmiennej ciągłej, – to wykres powierzchniowy, tzn. o wielkości zjawiska informuje powierzchnia słupka, – umożliwia porównanie z krzywą teoretyczną, np. rozkładu normalnego
Pudełkowy	Struktura zbiorowości zmiennej ciągłej, porównania rozkładów	– zawiera informacje o położeniu, rozrzucie i kształcie rozkładu, – umożliwia porównanie kilku rozkładów, a także identyfikację wartości odstających i ekstremalnych, – brak wartości dokładnych, – podobną charakterystykę mają wykresy wiolinowe
Łodyga-liść	Rozkład zmiennej numerycznej	– wskazuje dokładne wartości zmiennej, – dobry do zobrazowania wartości minimalnej i maksymalnej, a także zakresu zmienności
Liniowy	Analiza szeregu czasowego	– umożliwia analizę zmian zjawiska w czasie, – można przedstawić kilka linii na wykresie, – wskazuje minimum, maksimum i zakres zmienności, – warianty: wstęgowe, warstwowe, 3D
Radarowy	Dane wielowymiarowe numeryczne, analiza szeregu czasowego	– wartości poszczególnych kategorii są wykreślane wzdłuż osobnych osi rozpoczynających się w centrum wykresu i kończących się na zewnętrznym okręgu, – umożliwia zobrazowanie zmian cyklicznych w szeregu czasowym
Kołowy	Struktura zbiorowości z wyróżnionymi kategoriami	– pozwala na szybką analizę struktury jednej zbiorowości, – nie jest zalecany przy dużej liczbie kategorii
Pierścieniowy	Struktura jednej lub kilku zbiorowości z wyróżnionymi kategoriami	– pozwala na szybką analizę struktury kilku zbiorowości, – nie jest zalecany przy dużej liczbie kategorii, – formalnie jest to wykres słupkowy przedstawiony w układzie współrzędnych biegunowych

cd. tabeli 3.1

1	2	3
Rozrzutu	Zależność pomiędzy dwiema zmiennymi ilościowymi	– umożliwia analizę zależności dwóch zmiennych numerycznych, – pozwala na ocenę rodzaju zależności, siły i kierunku zależności liniowej, – umożliwia odczyt wartości minimalnych, maksymalnych i odstających
Mapowy	Prezentacja danych terytorialnych	– sposób przedstawienia wartości zjawiska w określonych jednostkach przestrzennych (np. administracyjnych)

Źródło: opracowanie własne na podstawie Kocimowski i Kwiatek (red., 1976); Kabacoff (2015); Wilke (2019); ArcGIS Pro (b.r.).

3.2.2. Zastosowania wykresów według rodzaju analizy

W tabeli 3.2 przedstawiono wybrane typowe możliwości zastosowania różnych rodzajów wykresów w zależności od rozważanego zagadnienia statystycznego i przeprowadzanej analizy. W kolumnie „Rodzaj analizy" ujęto tylko najczęściej stosowane zagadnienia związane z analizą danych. Dla różnych nieuwzględnionych typów analizy, jak na przykład „Analiza danych giełdowych", będą wykorzystywane różnorodne, często bardzo specyficzne metody prezentacji graficznych, które jednak nie będą rozważane w następnych rozdziałach.

Tabela 3.2. Wybór wykresu dla określonej analizy statystycznej

Rodzaj analizy	Cele analizy	Typ wykresu
1	2	3
Struktura zbiorowości (grup) dyskretnych	Określenie udziału poszczególnych kategorii w zbiorowości	– kołowy, – słupkowy, – punktowy, – łodyga-liść, – pudełkowy
Analiza zbiorowości	Określenie kolejności wielkości względem badanej zmiennej	– słupkowy, – liniowy, – punktowy, – łodyga-liść, – pudełkowy, – histogram (zmienna ciągła)
Porównanie struktur	Określenie udziału poszczególnych kategorii w zbiorowości, określenie kolejności wielkości względem badanej zmiennej	– słupkowy, – pierścieniowy, – skrzynkowy, – punktowy, – liniowy
Wizualizacja rozkładów ciągłych	Estymacja gęstości rozkładu zmiennej, określenie częstości dla poszczególnych klas zmiennej ciągłej	– histogram, – wykres gęstości, – liniowy, – pudełkowy, – wiolinowy, – QQ plot

cd. tabeli 3.2

1	2	3
Analiza szeregu czasowego	Określenie zmian wielkości zjawiska w czasie, wyodrębnienie wahań okresowych	– liniowy, – kolumnowy (tylko dla niewielkiej liczby okresów czasowych), – pudełkowy, – wiolinowy, – radarowy, – punktowy
Analiza zależności	Określenie rodzaju zależności dwóch zmiennych mierzalnych	– punktowy (rozrzutu), – macierzowy wykres rozrzutu, – liniowy (funkcja regresji, empiryczne linie regresji), – kolumnowy (dwa lub więcej szeregów), – mozaikowy
Analiza danych jakościowych	Ilustracja zależności dla zmiennych jakościowych	– mozaikowy, – wykres sita, – słupkowy
Analiza danych wielowymiarowych	Przedstawienie zależności dla wielu zmiennych	– macierzowy wykres rozrzutu, – wykres współrzędnych równoległych
Analiza przestrzenna	Przedstawienie skali zjawiska w ujęciu przestrzennym (terytorialnym)	– mapowy (kartogram), – słupkowy, – punktowy, – liniowy

Źródło: opracowanie własne na podstawie Sosulski (2019); ArcGIS Pro (b.r.).

3.2.3. Zastosowania wykresów według liczby zmiennych i skali pomiarowej

Przedstawione wcześniej zestawienia rodzajów wykresów ze względu ich typ (tabela 3.1) lub na rodzaj analizy (tabela 3.2) nie wyczerpują rozważań dotyczących możliwych klasyfikacji wykresów. Bardzo ważnym czynnikiem wpływającym na dobór formy wykresu jest skala pomiarowa, na jakiej dokonano pomiaru. W tabeli 3.3 dla rodzaju skali pomiaru (dyskretna lub ciągła) oraz liczby zmiennych wskazano możliwości zastosowania różnych typów wykresów.

Tabela 3.3. Wybór wykresu w zależności od liczby zmiennych i skali pomiarowej

Skala pomiaru	**Typ wykresu**
1	*2*
Jedna zmienna (X) dyskretna	– kołowy, – słupkowy, – punktowy
Jedna zmienna (X) ciągła	– histogram, – gęstości, – histogram z wykresem gęstości, – liniowy (diagram liczebności), – punktowy, – słupkowy

cd. tabeli 3.3

1	*2*
Dwie zmienne (X i Y), obie dyskretne	– punktowy, – mozaikowy, – heatmap
Dwie zmienne (X i Y), obie ciągłe	– rozrzutu, – warstwicowy, – gęstości trójwymiarowy, – heatmap
Dwie zmienne (X i Y): X dyskretna, Y ciągła	– pudełkowy, – wiolinowy, – punktowy, – liniowy, – słupkowy, – kołowy, – współrzędnych równoległych, – radarowy, – zmiany
Kilka zmiennych	– słupkowy, – pierścieniowy, – mozaikowy
Kilka zmiennych ilościowych	– macierzowy wykres rozrzutu, – współrzędnych równoległych, – liniowy, – radarowy, – twarze Chernoffa

Źródło: opracowanie własne na podstawie Kassambara (2013); Hilfiger (2016); ArcGIS Pro (b.r.).

4 Podstawy pracy z programem R

Największą wartością obrazu jest to,
że zmusza nas do zauważenia tego,
czego nigdy nie spodziewaliśmy się zobaczyć.

John Tukey*

Przez setki lat wszystkie obrazy, grafiki i wykresy musiały być przygotowywane ręcznie przez autora. W tym czasie powstało wiele różnorodnych prezentacji graficznych opartych na różnych danych liczbowych. Znalazły one aplikacje w ważnych i często interesujących zagadnieniach. Przygotowanie takich prezentacji nierzadko wymagało wielkiego wysiłku i długotrwałej pracy wykonawcy. Grafika taka mogła być indywidualnie zaprojektowana ze specjalnymi cechami dla poszczególnych zbiorów danych. Niektóre z takich prezentacji zaprezentowano w rozdziale 1 niniejszej monografii, a większy ich wybór przedstawiają Chun-houh Chen, Wolfgang Härdle i Antony Unwin (2008) oraz Michael Friendly (2000; 2005). Przykłady takich grafik są dostępne w różnych serwisach internetowych (DataViz Projekt b.r., FlowingData b.r.).

Obecnie grafika dla potrzeb analizy danych powstaje bez problemu z wykorzystaniem odpowiedniego oprogramowania. Upraszcza to w znacznym stopniu przygotowanie wykresów i upowszechnia ich zastosowanie. Oznacza to jednak, że pewne domyślne sposoby konstrukcji wykresów są przyjęte przez wielu jako oczywiste, bez możliwości zaawansowanej ingerencji w ich strukturę. Ogranicza to kreatywność w prezentacji, która dawniej była nieodłącznym elementem procesu graficznej prezentacji danych. W dalszej części rozważań zostaną przedstawione program R i podstawowe możliwości tego środowiska w zakresie graficznej prezentacji danych.

* Tukey (1977, s. vi) – tłumaczenie własne.

4.1. Ogólna charakterystyka programu R

Program R to język programowania i środowisko obliczeniowe, które jest szeroko stosowane w analizie danych, statystyce i badaniach naukowych. R został zaproponowany przez Rossa Ihakę i Roberta Gentlemana (1996) w latach 90. XX wieku, a obecnie jest utrzymywany i rozwijany przez międzynarodową społeczność programistów. Program R charakteryzuje się bogatym zestawem funkcji i bibliotek, które umożliwiają analizę danych, manipulację nimi, ich wizualizację oraz konstrukcję zaawansowanych modeli statystycznych. Dzięki swojej elastyczności R jest szczególnie popularny wśród statystyków, analityków danych oraz naukowców różnych dyscyplin (Kopczewska, Kopczewski i Wójcik 2009; Walesiak i Gatnar, red. 2009; Long i Teetor 2019).

Środowisko R dostarcza interaktywną konsolę, w której można wprowadzać polecenia i na bieżąco wykonywać obliczenia. Można również pisać skrypty R, które zawierają sekwencje poleceń i mogą być one uruchamiane jako programy. R obsługuje wiele typów danych, takich jak wektory, macierze, ramki danych i listy, co umożliwia efektywne przetwarzanie danych (Kończak 2012; 2016). Jednym z największych atutów R jest jego społeczność, która tworzy i udostępnia liczne pakiety rozszerzeń. Pakiety te zawierają gotowe funkcje i narzędzia do rozmaitych zastosowań, takich jak wnioskowanie statystyczne, testy permutacyjne, analizy wielowymiarowe, uczenie maszynowe, analiza sieci społecznościowych, przetwarzanie obrazów, analiza tekstu i wiele innych. Dzięki temu programiści mogą korzystać z istniejących rozwiązań i łatwo rozszerzać funkcjonalność języka R. Liczba dostępnych pakietów rozszerzających możliwości środowiska R dynamicznie rośnie. O ile w październiku 2012 roku dostępne były nieco ponad 4 tysiące takich bibliotek, a w październiku 2018 roku odnotowano ponad 13 tysięcy bibliotek, to w połowie czerwca 2024 roku ich liczba wyniosła już blisko 21 tysięcy (CRAN, b.r.).

Program R jest darmowy i dostępny na wielu platformach, w tym na systemy Windows, macOS i Linux. Ma również rozbudowaną dokumentację, wiele przykładów, przewodników i tutoriali, co ułatwia naukę i korzystanie z tego języka. W internecie dostępne są różne fora poświęcone zagadnieniom programowania w języku R. Wszystko to sprawia, że język ten jest aktualnie podstawowym narzędziem w analizie danych, a w szczególności wyznacza standardy w graficznej prezentacji danych.

Podstawowe biblioteki są dołączane bezpośrednio przy instalacji programu R. Biblioteki rozszerzające możliwości środowiska R wymagają dodatkowej instalacji. Thomas Rahlf (2017) wskazuje na następujące podstawowe biblioteki, które są dołączane podczas instalacji programu R:

- **base** – pakiet stanowiący podstawę środowiska R, zapewnia on ponad tysiąc funkcji dla wielu różnorodnych zagadnień statystycznych,
- **utils** – pakiet zawierający ponad 250 różnych funkcji,
- **stats** – podstawowy pakiet statystyczny zawierający ponad 600 funkcji do obliczeń statystycznych,
- **graphics** – zawiera prawie 100 funkcji zapewniających podstawowe operacje graficzne,
- **datasets** – zawiera zestaw przykładowych zbiorów danych,
- **methods** – zapewnia możliwości programowania obiektowego,
- **grDevices** – umożliwia operacje na urządzeniach graficznych,
- **grid** – pakiet zawierający około 200 funkcji składających się na alternatywny system graficzny R, który został opracowany przez Paula Murrella; pakiet **grid** jest podstawą zaawansowanych pakietów graficznych **lattice** i **ggplot**.

4.2. Podstawowe informacje o wykorzystywanych zbiorach danych

Zbiory danych mogą mieć różną strukturę. Mogą to być wektory, ramki danych, szeregi czasowe, tabele, macierze, a niekiedy będą to inne, nawet znacznie bardziej złożone struktury. Wektory należą do najprostszych struktur. Stanowią one uporządkowany ciąg wielkości, którymi zwykle są liczby, ale mogą to być również na przykład znaki, napisy czy wartości logiczne.

W niniejszej pracy korzystano z różnych zbiorów danych. W tabeli 4.1 przedstawiono krótką charakterystykę wybranych wykorzystywanych zbiorów, czego te dane dotyczą i jakiego typu jest to zbiór, a także informacje o liczbie zmiennych i obserwacji. Niniejsze zbiory reprezentują różne typy danych jak: wektor numeryczny, ramka danych, szereg czasowy, tabela wielowymiarowa oraz macierz. Wszystkie te zbiory są dostępne w programie R bez potrzeby instalowania dodatkowych bibliotek. Większość z tych zbiorów jest wykorzystywana tylko przy konstrukcji kilku wykresów. Zdecydowaną większość wszystkich zamieszczonych dalej prezentacji graficznych wykonano w oparciu o dane pochodzące ze zbioru **mtcars.** Z tego powodu zbiór ten będzie szerzej omówiony na początku rozdziału 5.

Tabela 4.1. Zbiory danych wykorzystane w pracy

Nazwa	Opis	Klasa (typ)	Liczba obserwacji/zmiennych
rivers	Długość (w milach) 141 największych rzek Ameryki Północnej	numeric	144 / 1
cars	Prędkość i odległość potrzebna do zatrzymania samochodu	data.frame	50 / 2
mtcars	Zużycie paliwa i 10 innych charakterystyk konstrukcji i osiągów samochodów	data.frame	32 / 11
AirPassengers	Miesięczna liczba pasażerów międzynarodowych linii lotniczych w latach 1949-1960	ts	144 / 1
Titanic	Dane z katastrofy statku Titanic	table	4 zmienne
chickwts	Dane o dodatkach paszowych i wadze kurcząt	data.frame	71 / 2
iris	Długości i szerokości kielicha oraz płatków dla 50 kwiatów z każdego z trzech gatunków irysa: setosa, versicolor i virginica	data.frame	150 / 5
diamonds	Cechy diamentów jak masa, jakość szlifu, kolor, szerokość górnej części, cena	data.frame	53940 / 10
VADeaths	Śmiertelność na 1000 ludności w Wirginii w 1940 roku	matrix array	5 x 4

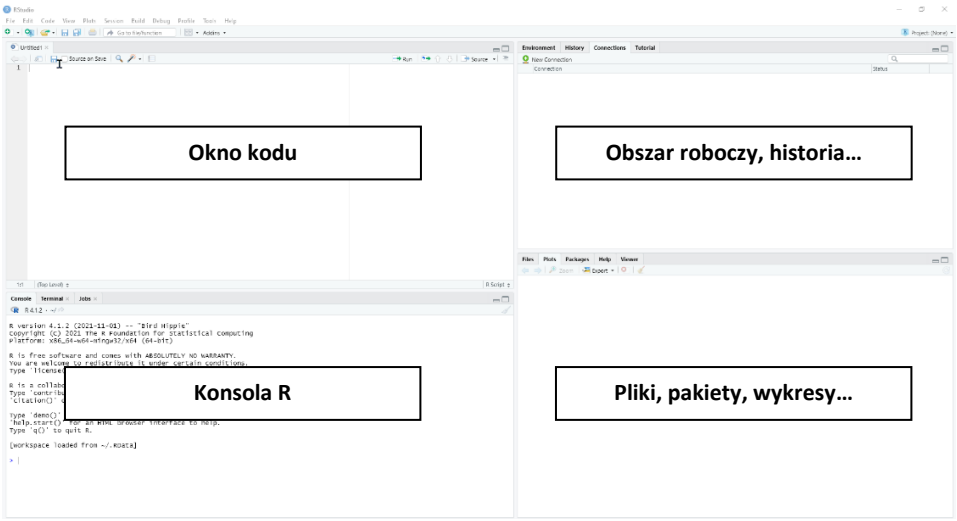
Źródło: CRAN (b.r.).

4.3. RStudio – charakterystyka

RStudio jest nakładką na program R. To zintegrowane środowisko programistyczne (IDE) zbudowane dla języka programowania R. RStudio umożliwia tworzenie, debugowanie i uruchamianie kodu R w łatwy sposób, dzięki czemu jest wygodnym narzędziem dla początkujących, jak również oferuje znaczne możliwości i wiele funkcjonalności dla zaawansowanych użytkowników języka R (Cirillo 2016). Umożliwia między innymi wygodną edycję kodu, zarządzanie pakietami, tworzenie i zarządzanie projektami, integrację z systemem kontroli wersji oraz wiele innych (Healy 2019). RStudio zdecydowanie ułatwia wyszukiwanie, instalowanie i zarządzanie pakietami CRAN, co umożliwia użytkownikom rozszerzanie funkcjonalności tej nakładki (Pimpler 2017). CRAN (Comprehensive R Archive Network) to repozytorium zawierające wiele tysięcy pakietów opracowanych dla języka programowania R. Dzięki CRAN i RStudio programiści i naukowcy zajmujący się analizą danych mogą szybko tworzyć i udostępniać swoje własne pakiety R, co przyspiesza proces tworzenia rozwiązań opartych na danych. Wersję instalacyjną RStudio można pobrać z witryny internetowej RStudio (RStudio b.r.).

Okno RStudio wraz z komentarzami przedstawiono na rysunku 4.1. Po uruchomieniu programu użytkownik ma do dyspozycji cztery obszary. Po lewej

u góry znajduje się okno z kodami skryptów. Użytkownik może pracować jednocześnie z wieloma skryptami. Kolejne skrypty zostaną umieszczone na następnych zakładkach. Po wprowadzeniu serii komend w oknie skryptu można uruchomić całość lub część tak przygotowanego programu. Komendy oraz wyniki zostaną wprowadzone do okna konsoli znajdującego się pod oknem skryptu. W konsoli użytkownik może także bezpośrednio wprowadzać komendy – tak jak w programie R.



Rysunek 4.1. Okno RStudio po uruchomieniu (wraz z objaśnieniami)

Źródło: opracowanie własne.

Dostępne po prawej stronie dwa okna mają po kilka zakładek. W zależności od wersji programu i zainstalowanych dodatków liczba zakładek może być większa. Umieszczone po prawej stronie u góry okno standardowo ma cztery następujące zakładki:

1. Environment (Środowisko) – zakładka ta przedstawia najważniejsze informacje dotyczące bieżącego środowiska R. Wyświetla listę wszystkich obiektów (zbiory danych, zmienne, funkcje i tym podobne), które są aktualnie załadowane do pamięci programu. Dla każdego obiektu podawane są jego nazwa, typ, rozmiar i wartość;
2. History (Historia) – przedstawia historię poleceń w konsoli R Studio. Umożliwia ponowne wykonanie wcześniejszych komend. Wykonane uprzednio komendy można także modyfikować;
3. Connections (Połączenia) – ułatwia zarządzanie połączeniami do różnych źródeł danych i baz danych. Umożliwia nawiązywanie i zamykanie połączeń

do baz danych, w szczególności do zbiorów danych przechowywanych poza środowiskiem R;
4. Tutorial (Samouczek) – zawiera zestaw samouczków i przykładowych projektów, które pomagają poznać podstawy programowania w języku R i korzystania z funkcji RStudio. Narzędzie to jest szczególnie przydane w początkowej fazie pracy z programem R.

Okno po prawej stronie u dołu standardowo ma pięć następujących zakładek:

1. Files (Pliki) – zakładka ta ułatwia dostęp do przeglądania i zarządzania plikami znajdującymi się na dysku w komputerze. Umożliwia sprawną nawigację po systemie plików, usuwanie, otwieranie i zapisywanie plików oraz tworzenie nowych katalogów. Daje możliwość ustawienia katalogu roboczego. Pozwala w prosty sposób importować i eksportować zbiory danych w różnych formatach;
2. Plots (Wykresy) – prezentuje wykresy utworzone podczas bieżącej sesji RStudio. Wykresy można przeglądać i zapisywać w różnych formatach graficznych;
3. Packages (Pakiety) – przedstawia listę zainstalowanych pakietów R na komputerze. Umożliwia wygodną instalację nowych pakietów oraz aktualizację lub odinstalowywanie istniejących pakietów;
4. Help (Pomoc) – zapewnia dostęp do dokumentacji i pomocy dla funkcji i pakietów R. Po wprowadzeniu nazwy funkcji lub pakietu zostaną wyświetlone związane z nią informacje oraz dostępne przykłady. Bardzo często jest możliwość uruchomienia wybranych przykładów z poziomu okna Help;
5. Viewer (Przeglądarka) – umożliwia wyświetlanie różnych rodzajów zawartości, takich jak pliki PDF, pliki HTML, strony internetowe i tym podobne. Po otwarciu lub utworzeniu takich plików zostaną one wyświetlone w tej zakładce.

4.4. Podstawy przekształcania danych i grafiki w programie R

Program R umożliwia konstrukcję różnorodnych wykresów. Wiele podstawowych wykresów można wykonać bezpośrednio po zainstalowaniu programu, wykorzystując funkcje pakietu **`graphics`**. Nie wymaga to instalowania dodatkowych bibliotek rozszerzających możliwości graficzne programu R. Znacznie większe możliwości uzyskania prezentacji graficznych otrzymuje się po zainstalowaniu dodatkowych bibliotek. Do najważniejszych pakietów związanych z graficzną prezentacją danych należy zaliczyć **`ggplot2`**, którego najważniejsze możliwości zostaną przedstawione w rozdziale 5.

4.4.1. Podstawy obróbki danych

Dla efektywnej pracy z danymi w programie R bardzo pomocne jest wykorzystanie pakietów pozwalających na podstawową obróbkę danych. Niezbędne są w szczególności dodawanie i usuwanie zmiennych lub obserwacji, konwertowanie zmiennych, filtrowanie, kategoryzacja danych oraz przekształcanie danych. Do przedstawionych zagadnień mogą być wykorzystane biblioteki jak na przykład: **`tidyverse`**, **`dplyr`**, **`xtable`**, **`data.table`**. Biblioteka **`tidyverse`** jest zbiorem wielu różnych pakietów, które zostały zbudowane w celu ułatwienia pracy z danymi i analizy danych. Biblioteka ta została zaprojektowana w oparciu o koncepcję tidy data oraz spójną koncepcję przetwarzania danych, co sprawia, że często jest wybierana przez użytkowników R. Skupia się na czytelności, spójności składni oraz efektywnym przetwarzaniu danych. W skład tej biblioteki wchodzą następujące pakiety:

- **`dplyr`** – umożliwia efektywne przetwarzanie i manipulację zmiennymi i zbiorami danych,
- **`tidyr`** – służy przekształcaniu danych w sposób zgodny z koncepcją tidy data, w tym zmianie układu danych na dłuższy lub szerszy format,
- **`readr`** – pozwala na wczytywanie danych z różnych formatów, takich jak pliki CSV, arkusza kalkulacyjnego Microsoft Excel czy pliki tekstowe,
- **`purrr`** – umożliwia manipulowanie strukturami danych, w tym mapowanie i filtrowanie danych,
- **`tibble`** – zapewnia skuteczną pracę ze zmodyfikowanymi ramkami danych,
- **`stringr`** – pozwala na manipulowanie ciągami znaków (tekstu), w tym na wyodrębnianie, łączenie i modyfikację danych tekstowych,
- **`forcats`** – umożliwia obsługę zmiennych o stałym i znanym zestawie wartości (czynników),
- **`ggplot2`** – pozwala na konstrukcję różnorodnych wykresów, których konstrukcja jest zgodna z gramatyką grafiki zaproponowaną przez Lelanda Wilkinsona (2005).

Pierwszy ze wskazanych pakietów – **`dplyr`** – pozwala na sprawną obróbkę i przetwarzanie danych. Umożliwia między innymi wydajne operacje na ramkach danych (data frames). Biblioteka dostarcza szeroki zestaw funkcji i narzędzi, które ułatwiają wykonywanie typowych operacji na danych, takich jak filtrowanie, wybieranie kolumn, ich modyfikację, grupowanie, agregowanie i łączenie. Do najważniejszych funkcji tego pakietu należą:

- *`filter()`* – pozwala na filtrowanie danych na podstawie określonych warunków,
- *`select()`* – służy do wybierania kolumn z ramki danych,

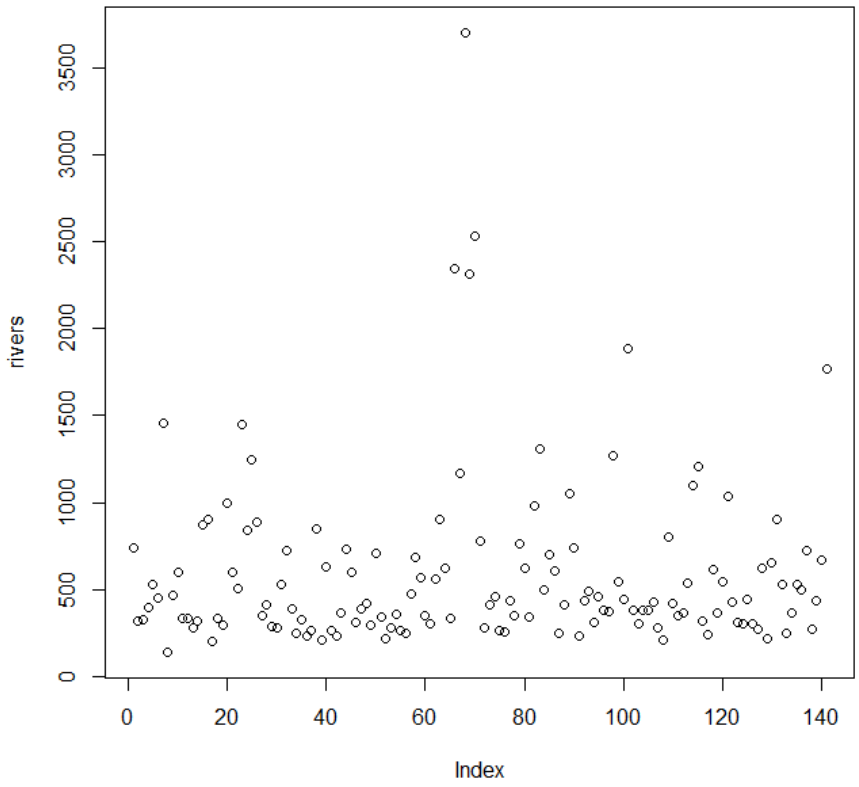
– *mutate()* – umożliwia dodawanie nowych kolumn lub modyfikację istniejących,
– *group_by()* – pozwala na grupowanie danych według określonych kategorii,
– *summarize()* – służy do agregowania danych w grupach i tworzenia podsumowań,
– *arrange()* – pozwala na sortowanie danych według określonych kolumn,
– *join()* – umożliwia łączenie danych z różnych źródeł na podstawie określonych kolumn.

4.4.2. Wykresy uzyskane z wykonania funkcji *plot*

Podstawowe zastosowanie funkcji *plot* to konstrukcja wykresu punktowego. Jednak w zależności od klasy (typu) zbioru wejściowego uruchamiane są różne metody i w konsekwencji otrzymuje się wykresy różnego rodzaju (Chang 2019). W podrozdziale przedstawiono wybrane możliwości funkcji *plot* w zastosowaniu do konstrukcji wykresów różnego typu. W przedstawionych przykładach zwracana jest uwaga na wykorzystywane metody graficzne. Dlatego tytuły wykresów (podpisy umieszczone pod rysunkami) są dwuczęściowe. W pierwszej kolejności odnoszą się do rodzaju (forma graficzna) zastosowanego wykresu, a w drugiej do prezentowanych na wykresie treści. W opracowaniach związanych z analizą danych tytuł powinien odnosić się wyłącznie do prezentowanych na wykresie treści.

Zastosowanie funkcji *plot* do zbioru danych postaci wektora *n* obserwacji prowadzi do wykreślenia wykresu punktowego, gdzie na osi OX jest numer obserwacji, a na osi OY wartość zmiennej. Komenda pozwalająca na uzyskanie graficznej prezentacji długości największych rzek Ameryki Północnej (por. rysunek 4.2) jest następująca:

```
plot(rivers)
```

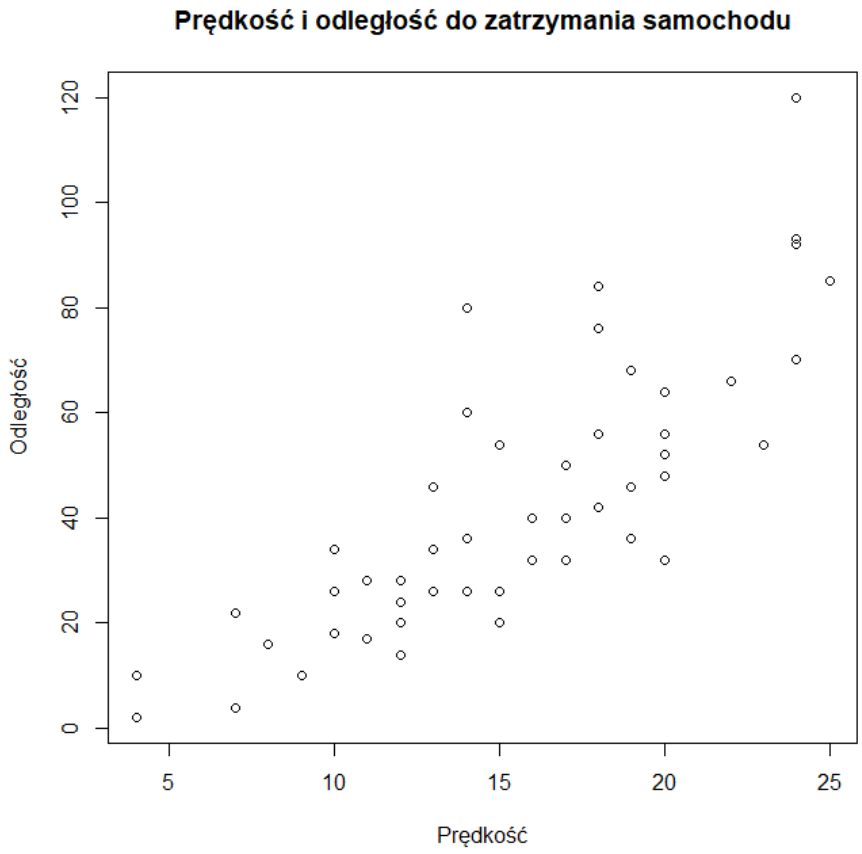


Rysunek 4.2. Wykres punktowy dla jednej zmiennej numerycznej. Długość w milach największych rzek Ameryki Północnej

Źródło: opracowanie własne w programie R.

Niniejsze zastosowanie funkcji *plot* pozwoliło na zaprezentowanie na wykresie jednej zmiennej. Znacznie częściej funkcja *plot* jest wykorzystywana do zobrazowania układu danych ze względu na dwie zmienne na wykresie rozrzutu. Zbiorem zawierającym dwie zmienne jest zbiór **cars** (por. tabela 4.1). Wywołanie funkcji *plot* dla tego zbioru prowadzi do uzyskania wykresu rozrzutu. Przy konstrukcji wykresu należy dodać opisy osi i ewentualnie tytuł wykresu. Realizuje to poniższy kod, a rezultat został przedstawiony na rysunku 4.3.

```
plot(cars, main='Prędkość i odległość do zatrzymania
samochodu',xlab='Prędkość', ylab='Odległość')
```

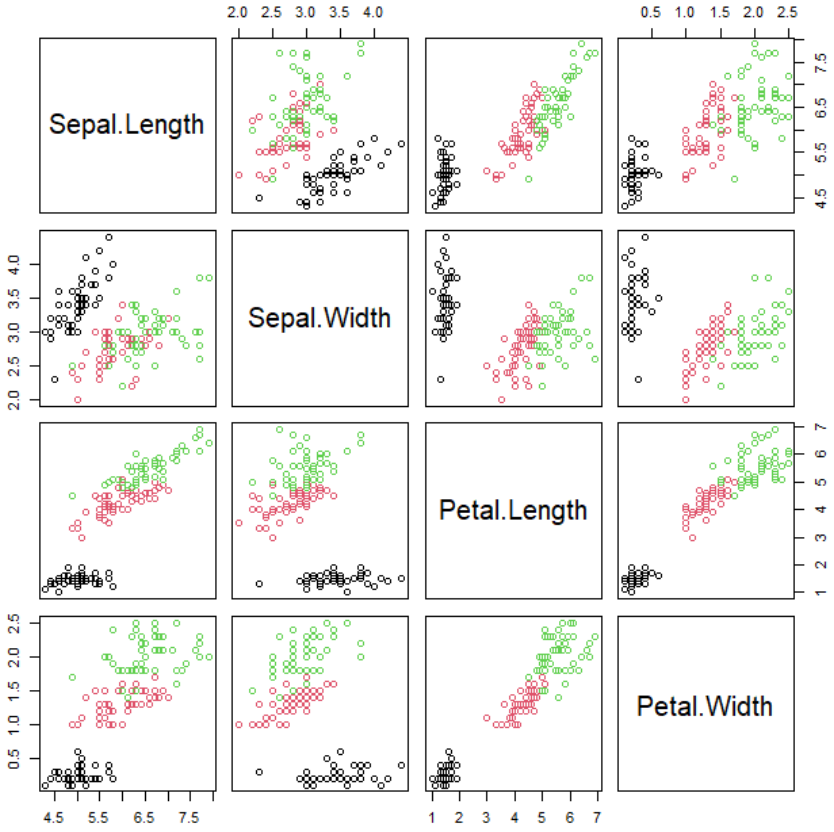


Rysunek 4.3. Wykres rozrzutu dla dwóch zmiennych ilościowych. Prędkość i odległość do zatrzymania samochodu

Źródło: opracowanie własne w programie R.

Jeżeli jako argument funkcji *plot* zostanie podany wielowymiarowy zbiór o więcej niż dwóch zmiennych liczbowych (ramka danych), to w wyniku otrzymuje się macierzowy wykres rozrzutu. Poniższy kod prowadzi do wykreślenia macierzowego wykresu rozrzutu dla czterech zmiennych ilościowych ze zbioru **iris**.

```
plot(iris[-5],col=iris[,5])
```



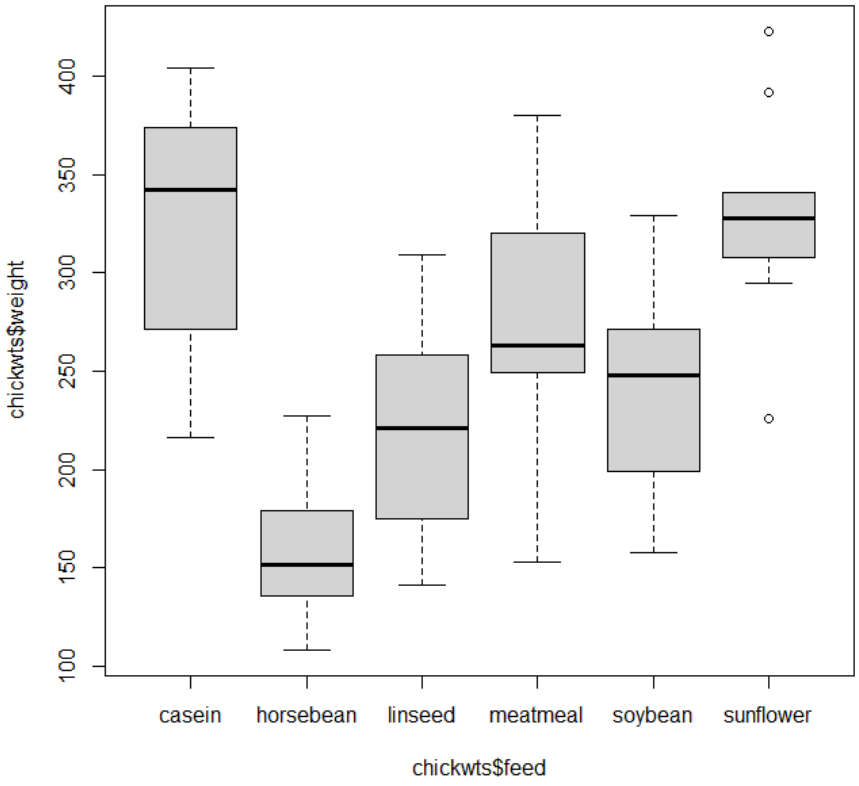
Rysunek 4.4. Macierzowy wykres rozrzutu dla czterech zmiennych. Długości i szerokości kielicha i płatków kwiatu irys (zbiór iris)

Źródło: opracowanie własne w programie R.

Rysunek 4.4 przedstawia macierzowy wykres rozrzutu dla zmiennych numerycznych zbioru **iris** (pierwsze cztery zmienne zbioru). Zmienna *Species* (jakościowa) określająca gatunek kwiatu kosaćca została wykorzystana jako wyróżnik koloru punktów.

Funkcja `plot` pozwala także uzyskać wykres pudełkowy. W takim przypadku jako argument należy wprowadzić formułę w formacie *y*~*x*, gdzie zmienna *y* jest ilościowa, a zmienna *x* dyskretna. Przykładowy kod z wykorzystaniem danych ze zbioru **chickwts** jest następujący:

```
plot(chickwts$weight~chickwts$feed)
```

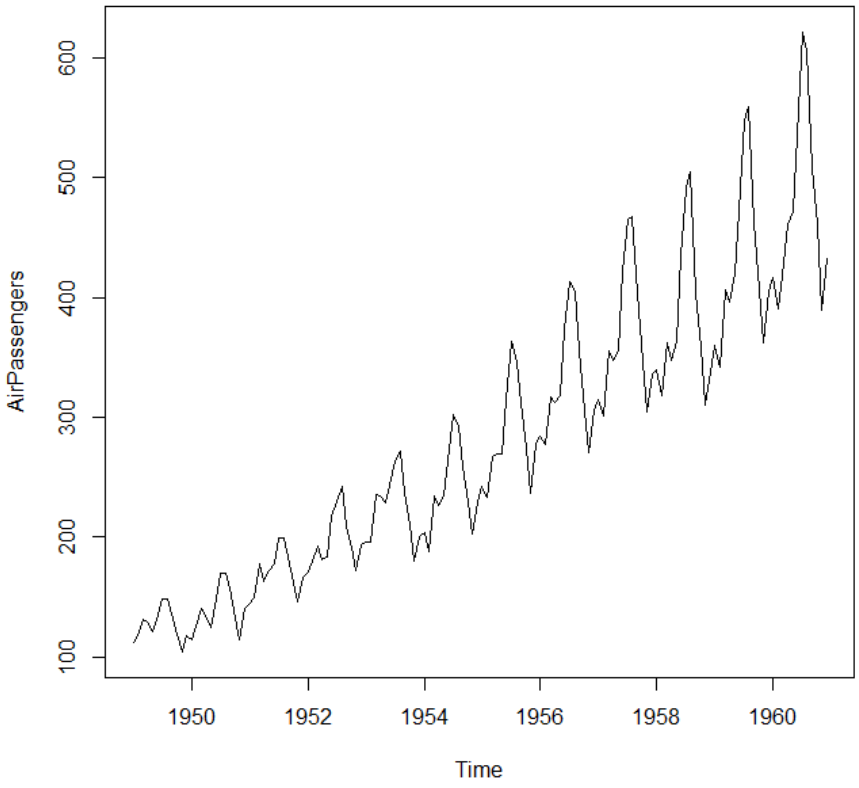


Rysunek 4.5. Wykres pudełkowy wykonany z wykorzystaniem funkcji `plot`. Waga piskląt kurczaków w zależności od rodzaju karmy

Źródło: opracowanie własne w programie R.

Na rysunku 4.5 przestawiono wykres pudełkowy uzyskany za pomocą funkcji `plot`. Wykres ten pozwala na łatwe porównanie wagi piskląt w zależności od zastosowania różnych wariantów diety (sześć wariantów). Zastosowanie funkcji `plot` do danych w postaci szeregu czasowego (*ts*) prowadzi do konstrukcji wykresu liniowego, gdzie na osi OX jest reprezentowany czas, a na osi OY znajdują się wartości danej zmiennej. Odpowiedni kod i jego rezultat (rysunek 4.6) z wykorzystaniem danych ze zbioru **AirPassengers** są następujące.

```
plot(AirPassengers)
```

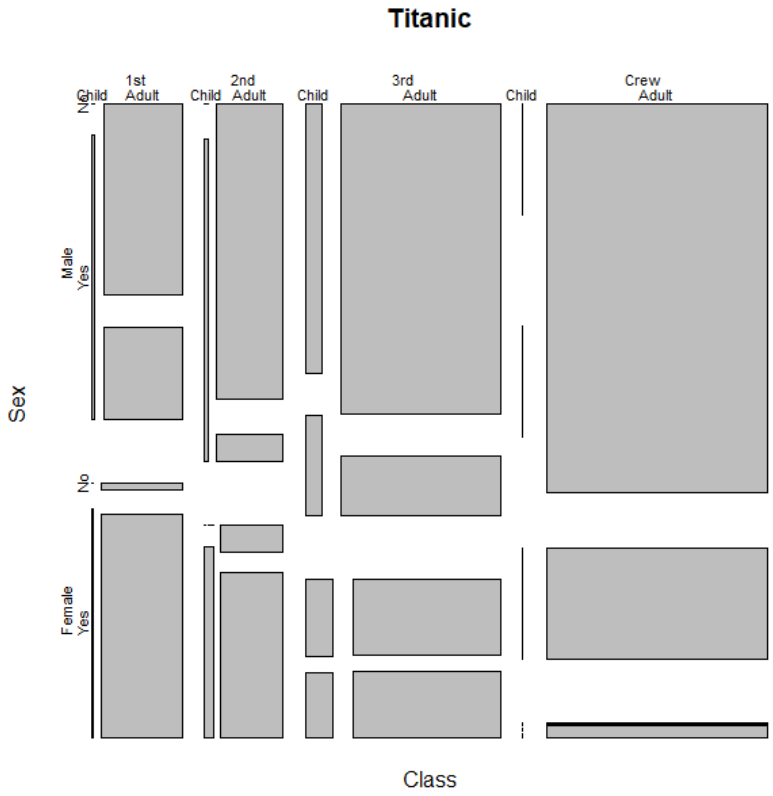


Rysunek 4.6. Wykres liniowy – dane w postaci szeregu czasowego. Liczba pasażerów w milionach linii lotniczych w latach 1948-1960

Źródło: opracowanie własne w programie R.

Przy wprowadzeniu argumentu w postaci tabeli do funkcji *plot* jako rezultat otrzymuje się wykres mozaikowy. Poniższy przykład prezentuje konstrukcję wykresu mozaikowego dla danych ze zbioru **Titanic**. Wynik realizacji kodu przedstawiono na rysunku 4.7.

```
plot(Titanic)
```

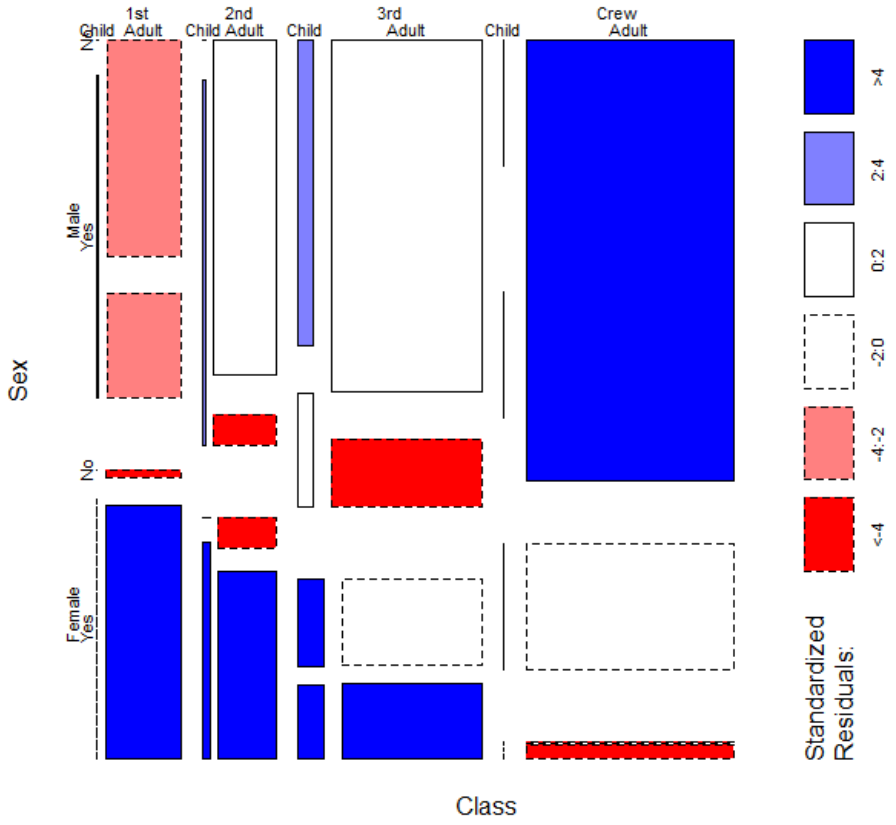


Rysunek 4.7. Wykres mozaikowy. Przeżycie katastrofy pasażerów Titanica w zależności od płci, wieku i klasy

Źródło: opracowanie własne w programie R.

Dodanie parametru shade, jak w poniższym kodzie, prowadzi do wykreślenia wykresu mozaikowego, na którym zaznaczono standaryzowane różnice pomiędzy liczebnościami obserwowanymi i oczekiwanymi (por. rysunek 4.8).

```
plot(Titanic,shade=TRUE)
```

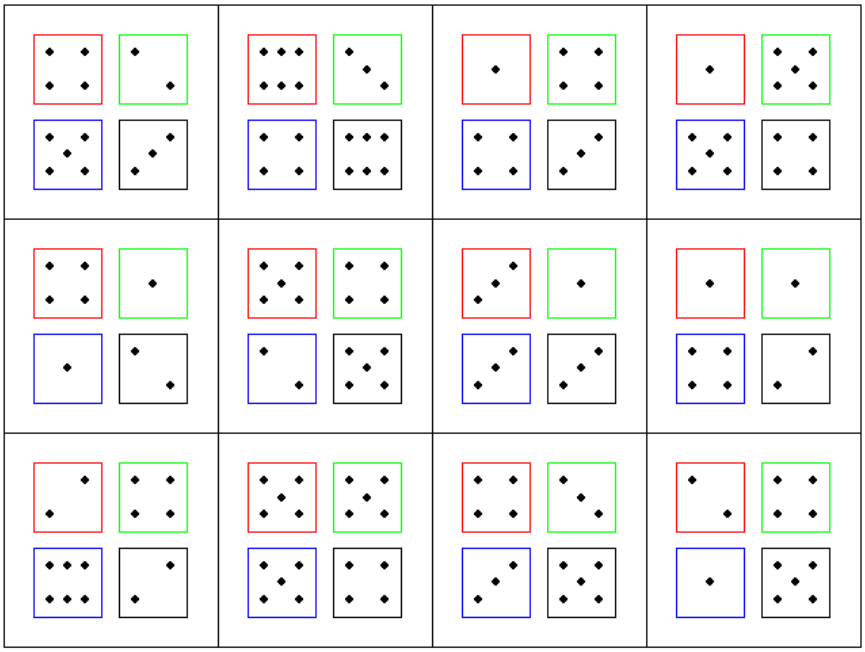


Rysunek 4.8. Wykres mozaikowy. Przeżycie katastrofy pasażerów Titanica w zależności od płci, wieku i klasy, ze wskazaniem standaryzowanych różnic pomiędzy liczebnościami obserwowanymi i oczekiwanymi

Źródło: opracowanie własne w programie R.

Przedstawione zastosowania funkcji *plot* do konstrukcji wykresów nie wyczerpują jej wszystkich możliwości w tym zakresie. Po instalacji pakietów rozszerzających możliwe jest dodanie nowych funkcjonalności do tej funkcji. Przykład takiego zastosowania z wykorzystaniem pakietu **TeachingDemos** przedstawiono na rysunku 4.9, który otrzymuje się w wyniku realizacji następującego kodu.

```
library(TeachingDemos)
plot(dice(12,4))
```



Rysunek 4.9. Graficzna prezentacja wyników rzutu kostką. Wyniki dwunastokrotnego rzutu czterema sześciennymi kostkami do gry

Źródło: opracowanie własne w programie R.

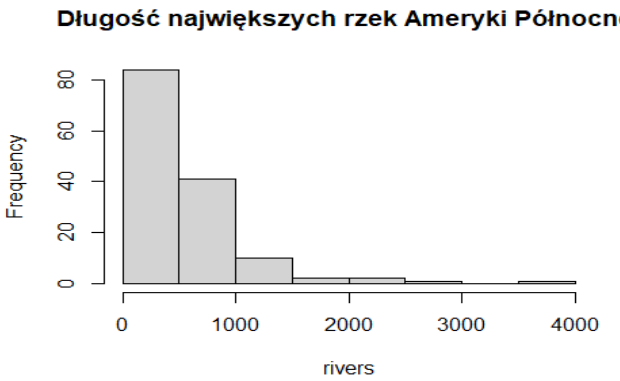
Na rysunku 4.9 przedstawiono w formie graficznej wyniki dwunastokrotnego rzutu czterema sześciennymi kostkami do gry. Uzyskanie takiego rezultatu było możliwe dzięki wcześniejszemu zainstalowaniu i załadowaniu pakietu **`TeachingDemos`**.

Wszystkie przedstawione w tym punkcie wykresy wykonano z wykorzystaniem funkcji ***`plot`***. Wyniki za każdym razem były zupełnie inne, a ostateczny rezultat był związany z typem zbioru danych.

4.4.3. Wybrane podstawowe funkcje graficzne

Do najczęściej wykorzystywanych wykresów w analizach statystycznych należy zaliczyć histogram. Do konstrukcji tego rodzaju wykresu w programie R wykorzystywana jest funkcja ***`hist`***. Wywołanie tej funkcji dla zbioru danych **rivers** przedstawia poniższy kod.

```
hist(rivers, main = 'Długość największych rzek Ameryki
Północnej')
```

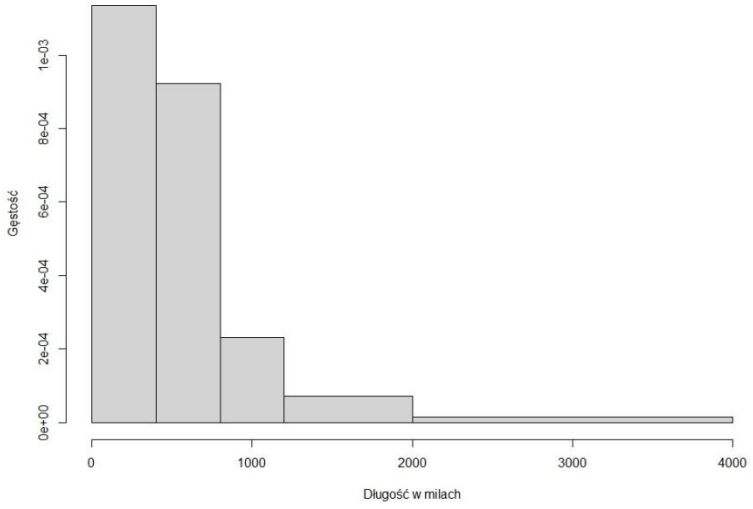


Rysunek 4.10. Histogram w podstawowej konstrukcji. Długość w milach największych rzek Ameryki Północnej

Źródło: opracowanie własne w programie R.

Na rysunku 4.10 z wykorzystaniem histogramu przedstawiono długość największych rzek Ameryki Północnej. Domyślnie histogram jest konstruowany dla przedziałów o jednakowej długości. Niekiedy dla dokładniejszego przedstawienia charakterystyki rozkładu zaleca się odwołanie się do konstrukcji histogramu z przedziałami różnej długości. Przykład takiej konstrukcji realizuje następujący kod.

```
hist(rivers,breaks=c(0,400,800,1200,2000,4000),xlab='Długość w
milach',ylab='Gęstość',main='')
```



Rysunek 4.11. Histogram z przedziałami klasowymi o niejednakowej długości. Długość w milach największych rzek Ameryki Północnej

Źródło: opracowanie własne w programie R.

Na rysunku 4.11 przedstawiono za pomocą histogramu długości największych rzek Ameryki Północnej. W przypadku wykresu na rysunku 4.10 histogram został skonstruowany dla przedziałów o jednakowej długości, a w drugim (rysunek 4.11) długości te były różne. Dodatkowo usunięto tytuł znad pola wykresu.

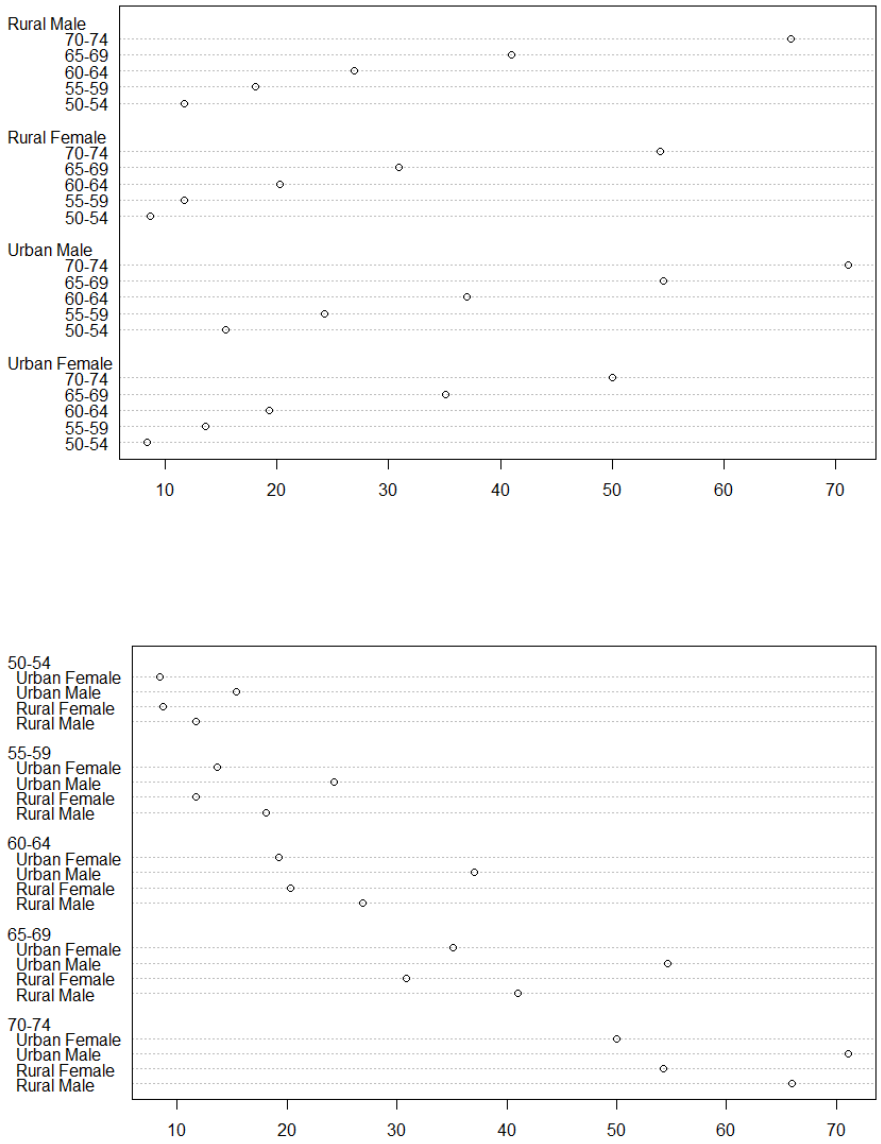
Na rysunku 4.2 pokazano konstrukcję wykresu punktowego z wykorzystaniem funkcji *plot*. Nieco inną prezentację graficzną wykresu punktowego umożliwia funkcja *dotchart*. Na rysunku 4.12 przedstawiono dane ze zbioru **VADeaths**. Zbiór ten ma postać macierzy. Przedstawia on współczynniki zgonów w stanie Virginia. W przypadku macierzy możliwe jest wykonanie transpozycji. Dane z tego zbioru oraz po operacji transpozycji zostały przedstawione na rysunku 4.13.

```
> VADeaths
      Rural Male Rural Female Urban Male Urban Female
50-54       11.7          8.7       15.4          8.4
55-59       18.1         11.7       24.3         13.6
60-64       26.9         20.3       37.0         19.3
65-69       41.0         30.9       54.6         35.1
70-74       66.0         54.3       71.1         50.0
> t(VADeaths)
             50-54 55-59 60-64 65-69 70-74
Rural Male    11.7  18.1  26.9  41.0  66.0
Rural Female   8.7  11.7  20.3  30.9  54.3
Urban Male    15.4  24.3  37.0  54.6  71.1
Urban Female   8.4  13.6  19.3  35.1  50.0
```

Rysunek 4.12. Zbiór VADeaths oraz ten sam zbiór po transpozycji

Źródło: opracowanie własne.

```
par(mfrow=c(2,1))
dotchart(VADeaths)
dotchart(t(VADeaths))
```



Rysunek 4.13. Wykres punktowy (dotchart) dla zbioru VADeaths i jego transpozycji. Współczynniki zgonów w stanie Virginia w 1940 roku

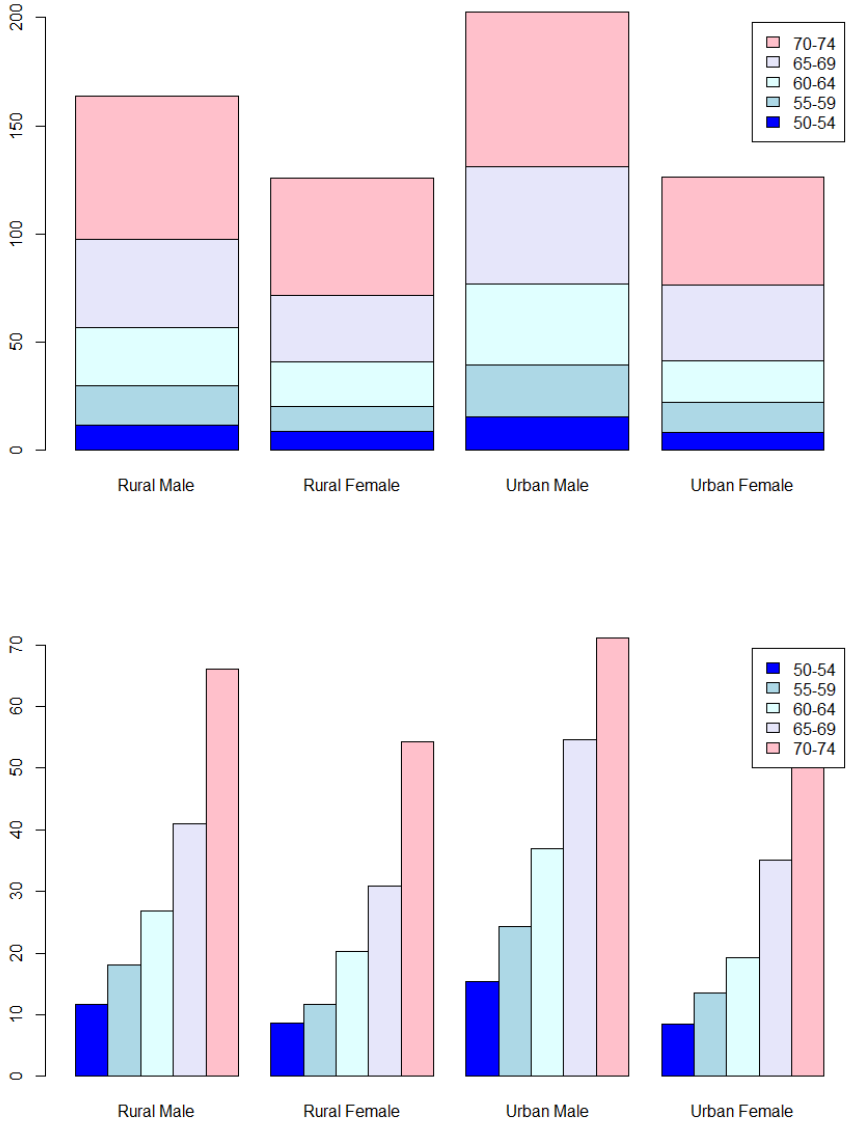
Źródło: opracowanie własne w programie R.

Kolejną funkcją pozwalającą na konstrukcję często stosowanych wykresów słupkowych jest funkcja *barplot*. Umożliwia ona konstrukcję wykresów słupkowych oraz słupkowych skumulowanych. Wynik poniższego kodu został zaprezentowany na rysunku 4.14.

```
par(mfrow=c(2,1))
barplot(VADeaths,
         col = c("blue", "lightblue", "lightcyan",
                 "lavender", "pink"),
        legend = rownames(VADeaths))
barplot(VADeaths,
         col = c("blue", "lightblue", "lightcyan",
                 "lavender", "pink"),
        legend = rownames(VADeaths), beside = TRUE)
```

Przy konstrukcji wykresu słupkowego z wykorzystaniem funkcji *barplot* użytkownik może podać wartości różnych parametrów. Do najbardziej przydatnych należą:

- **height** – wektor lub macierz określający wysokość każdego słupka na wykresie,
- **width** – wektor określający szerokość każdego słupka na wykresie,
- **space** – wektor określający odstępy między słupkami,
- **names.arg** – wektor zawierający etykiety dla każdego słupka na osi OX,
- **horiz** – określa, czy układ wykresu ma być poziomy czy pionowy,
- **col** – określa kolor lub kolory dla słupków,
- **border** – określa kolor lub kolory dla obramowania słupków,
- **main** – tytuł główny wykresu,
- **xlab**, **ylab** – etykieta osi OX i OY,
- **xlim**, **ylim** – zakres osi OX i OY,
- **beside** – słupki danych mogą być obok siebie czy nałożone,
- **legend.text** – etykiety do umieszczenia w legendzie.

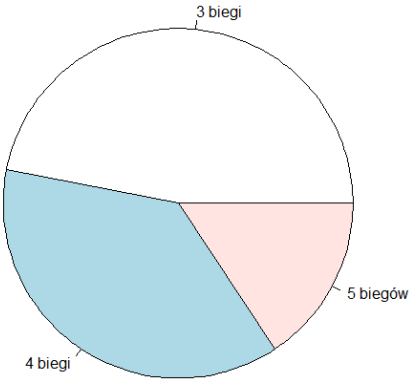


Rysunek 4.14. Wykresy słupkowe skumulowane (u góry) i słupkowe (na dole). Współczynniki zgonów w stanie Virginia

Źródło: opracowanie własne w programie R.

Bardzo często w prezentacjach graficznych do przedstawienia struktury zbiorowości wykorzystuje się wykresy kołowe. Przykład konstrukcji wykresu kołowego z wykorzystaniem funkcji ***pie*** przedstawia następujący kod.

```
pie(table(mtcars$gear),labels=c('3 biegi','4 biegi',
'5 biegów'),main='')
```



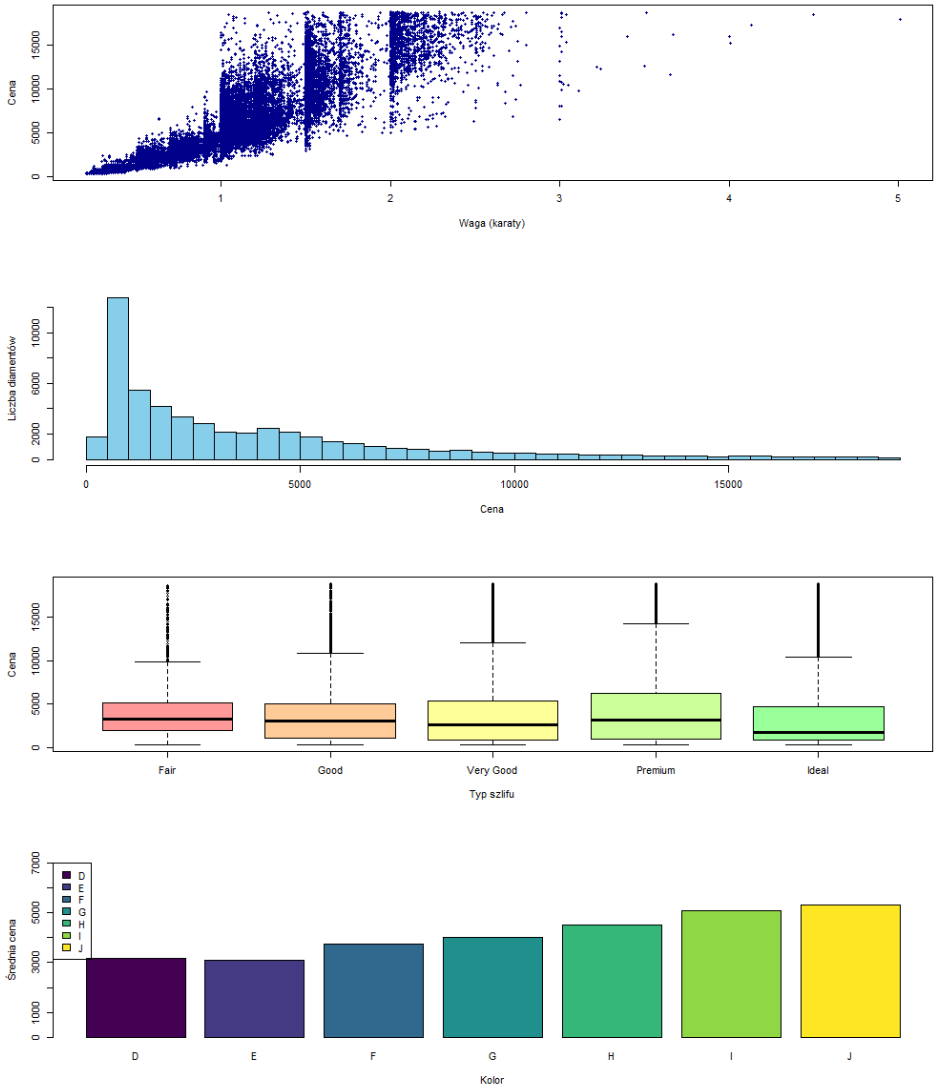
Rysunek 4.15. Wykres kołowy. Struktura samochodów ze względu na liczbę biegów w samochodzie

Źródło: opracowanie własne w programie R.

Na rysunku 4.15 przedstawiono strukturę samochodów ze zbioru **mtcars** ze względu na liczbę biegów.

Niekiedy warto w jednym obszarze graficznym umieścić kilka wykresów. Taka możliwość została już wykorzystana przy konstrukcji na rysunkach 4.13 i 4.14. Poniższy kod pozwala na umieszczenie czterech wykresów w układzie pionowym w jednym obszarze graficznym. Określa on konstrukcję kolejno wykresu rozrzutu, histogramu, wykresu pudełkowego oraz wykresu słupkowego dla danych ze zbioru **diamonds**. Rezultat realizacji kodu został przedstawiony na rysunku 4.16.

```
# Umieszczenie kilku wykresów w jednym obszarze graficznym.
par(mfrow=c(4,1))

plot(price ~ carat, data = diamonds,  xlab = "Waga (karaty)",
ylab = "Cena", col = "darkblue", pch = 20)

hist(diamonds$price, breaks = 30, col = "skyblue",  main='',
xlab = "Cena", ylab = "Liczba diamentów")

boxplot(price ~cut , data = diamonds,  xlab = "Typ szlifu", ylab
= "Cena", col = c("#FF9999", "#FFCC99", "#FFFF99", "#CCFF99",
"#99FF99"))

avg_price <- tapply(diamonds$price, diamonds$color, mean)
barplot(avg_price,  xlab = "Kolor", ylab = "Średnia cena",
ylim=c(0,7000),col = viridis::viridis(length(avg_price)))

legend("topleft", legend = names(avg_price), fill = virid-
is::viridis(length(avg_price)))
```



1. Waga i cena diamentów;
2. Cena diamentów;
3. Typ szlifu i cena diamentów;
4. Kolor i średnia cena diamentów.

Rysunek 4.16. Cztery wykresy w jednym obszarze graficznym. Zbiór diamonds

Źródło: opracowanie własne w programie R.

4.4.4. Kolory i palety kolorystyczne

Kolory są bardzo istotnym elementem składowym wykresów (Steele i Iliinsky, red. 2010). Właściwy dobór kolorów umożliwia wizualne odróżnienie różnych serii danych lub kategorii. Może pomóc albo utrudnić postrzeganie różnych serii danych bądź kategorii. Ważne jest, aby dobrać kontrastowe kolory, ograniczyć ich liczbę, a także zwrócić uwagę na kolor tła. Kolor jest najsilniejszym bodźcem wizualnym (Kirk 2019). Dokonany wybór będzie miał istotny wpływ na odbiór całej prezentacji. Różnorodność kolorów ma duże znaczenie. Wiele typów wykresów wykorzystuje atrybut koloru do reprezentowania wartości danych, niezależnie od tego, czy jest on używany do klasyfikowania skal ilościowych, czy do kojarzenia z dyskretnymi wartościami jakościowymi. Przy konstrukcji wykresów warto korzystać z wbudowanych palet kolorystycznych.

Paleta kolorystyczna to zestaw kolorów, które mogą być wykorzystywane do tworzenia grafik i wykresów. Dobrze dobrana paleta kolorów może pomóc w przekazywaniu informacji, a jednocześnie zwiększyć czytelność wykresu. Przy wyborze palety kolorów ważne jest, aby uwzględnić kontekst, w którym wykres będzie używany, a także rodzaj przedstawianych danych. Warto wybrać paletę kolorów, która będzie odpowiednia dla odbiorców oraz pozwoli na łatwe odczytanie i interpretację wykresu. Istnieją różne rodzaje palet kolorów, w tym:

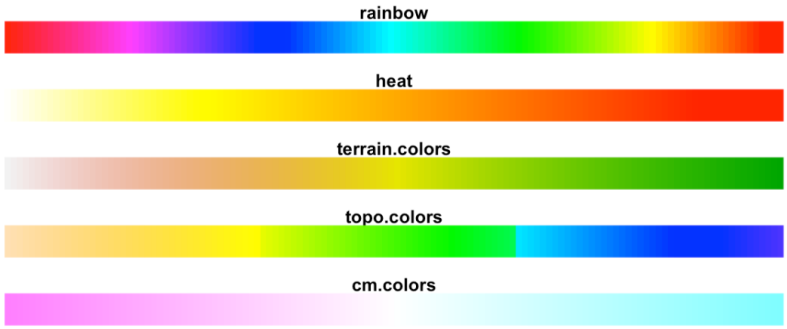
- Sekwencyjne (sequntial) – składają się z kolorów ułożonych w kolejności od jasnego do ciemnego. Takie palety są często stosowane do przedstawiania danych ciągłych, takich jak temperatura lub wartości numeryczne;
- Rozbieżne (divergent) – składają się z kolorów ułożonych w kolejności od dwóch przeciwnych kolorów do środka palety. Takie palety są często stosowane do przedstawiania danych, które mają dwie skrajne wartości, takie jak wyniki badań przed i po leczeniu;
- Jakościowe (qualitative) – składają się z kilku jaskrawych kolorów, które są używane do odróżniania kategorii danych. Takie palety są często stosowane do przedstawiania danych dyskretnych, takich jak grupy wiekowe lub płci.

Wybrane palety kolorystyczne dostępne w programie R przedstawiono w tabeli 4.2.

Tabela 4.2. Przykłady palet kolorystycznych wykorzystywanych w programie R

Pakiet	Liczba palet	Palety, wybrane informacje
`grDevices`	5	Palety: rainbow, heat, terrain.colors, topo.colors, cm.colors
`RColorBrewer`	35	RColorBrewer, od 8 do 12 poziomów kolorystycznych
`viridis`	4	Palety sekwencyjne: viridis, magma, plasma, inferno
`viridisLite`	4	Łatwy dostęp do palet kolorystycznych
`ggsci`	21	Palety inspirowane nauką oraz kolorami popularnych narzędzi i stron internetowych
`ggplot2`	wiele	Liczba palet zależna od wersji pakietu. Dostępne m.in. viridis, Brewer, manual
`wesanderson`	16	wesanderson, od 4 do 7 poziomów kolorystycznych
`scales`	wiele	Rozszerza możliwości `ggplot2` o dodatkowe skale kolorystyczne
`colorspace`	wiele	Zapewnia szeroki zestaw narzędzi do wybierania poszczególnych kolorów lub palet kolorów

Źródło: opracowanie własne na podstawie CRAN (b.r.).



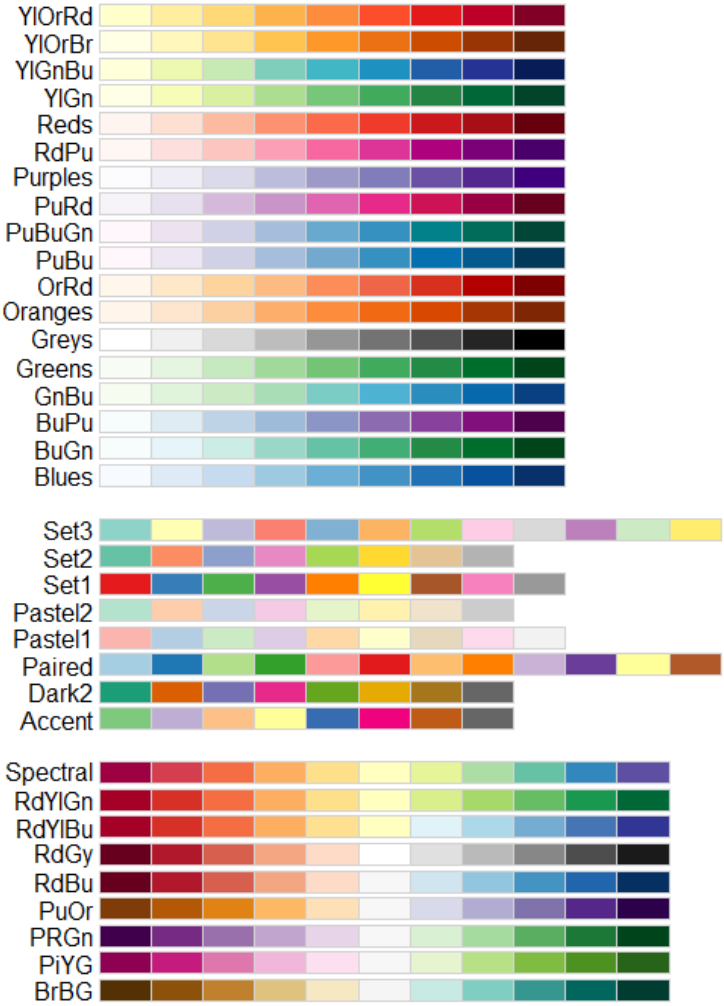
Rysunek 4.17. Palety dostępne w pakiecie `grDevices`

Źródło: opracowanie własne na podstawie Alboukadel (b.r.).

W oparciu o pakiet **`grDevices`** (por. rysunek 4.17) użytkownik może konstruować własne palety kolorystyczne. Przykłady takich konstrukcji są następujące.

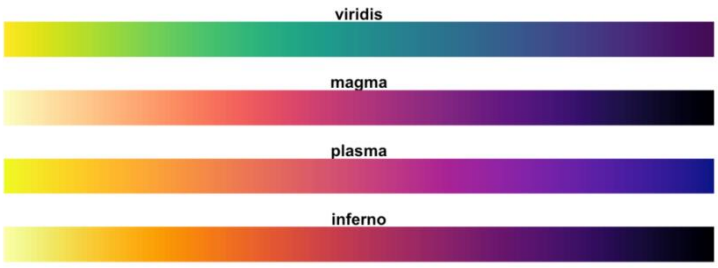
```
# Przykłady palet kolorystycznych
paleta1<- heat.colors(5)        # paleta z 5 kolorami
paleta2<- topo.colors(8)  # paleta z 8 kolorami
```

Pakiet **`RColorBrewer`** zawiera 35 palet kolorystycznych, na które składa się od 8 do 12 kolorów (por. rysunek 4.18). Palety te są szczególnie użyteczne w tworzeniu wykresów i grafik, gdzie potrzeba różnorodnych kolorów, aby odróżniać kategorie lub poziomy danych. W pakiecie **`RColorBrewer`** występują zarówno skale monochromatyczne, jak i wielobarwne. Skala ColorBrewer Blues (por. rysunek 4.18) jest skalą monochromatyczną, która zmienia się od ciemnego do jasnego błękitu, a skala Greens od jasnej do bardzo ciemnej zieleni (Chang 2019).



Rysunek 4.18. Palety kolorystyczne w pakiecie `RColorBrewer`

Źródło: opracowanie własne w programie R; CRAN R Project. *RColorBrewer* (b.r.).



Rysunek 4.19. Palety dostępne w pakiecie `viridis`

Źródło: opracowanie własne na podstawie Alboukadel (b.r.).

Pakiet **viridis** (por. rysunek 4.19) oferuje różne palety kolorystyczne, takie jak viridis, magma, plasma i inferno, które różnią się odcieniami i nasyceniem. Pozwala to wybrać odpowiednią paletę, która najlepiej pasuje do prezentowanych danych i rodzaju wykresu. Palety w pakiecie **viridis** zostały zaprojektowane tak, aby były czytelne dla osób z różnymi typami daltonizmu. Dzięki temu dane przedstawione na wykresach wizualizacyjnych mogą być bardziej zrozumiałe dla szerszego kręgu odbiorców.

W pakiecie **wesanderson** (rysunek 4.20) znajduje się 16 palet kolorystycznych. W paletach tych dostępnych jest od czterech do siedmiu kolorów.

Rysunek 4.20. Palety kolorystyczne w pakiecie wesanderson

Źródło: opracowanie własne na podstawie Alboukadel (b.r.); CRAN R Project. *Package 'wesanderson'* (b.r.).

Obszerny zestaw 2 759 palet kolorystycznych pochodzących z 75 różnych bibliotek zawiera pakiet **`paletteer`**. W pakiecie wyróżniono dwie grupy palet: dyskretne i ciągłe. Szczególnym rodzajem palet dyskretnych są palety dynamiczne, które pozwalają na uzyskanie żądanej przez użytkownika liczby wariantów kolorystycznych. Niektóre z wymienionych palet kolorystycznych będą wykorzystane przy konstrukcji wykresów w dalszej części pracy. Obszerny wykaz dostępnych palet kolorystycznych znajduje się w Paletteer Gallery (PMassicotte, b.r.).

5 Charakterystyka grafiki w ggplot2

Wykresy niczego nie udowadniają,
ale pozwalają dostrzec wyróżniające się cechy.

Ronald A. Fisher*

Do konstrukcji zaawansowanych wykresów w programie R najczęściej wykorzystywany jest pakiet **ggplot2**, którego autorem jest Hadley Wickham. Nazwa odnosi się do „Grammar of Graphics", czyli gramatyki grafiki, co oznacza, że konstrukcja wykresów z wykorzystaniem tego pakietu opiera się na zasadzie budowy wykresów za pomocą składni opisującej elementy graficzne. Gramatykę grafiki zaproponował Leland Wilkinson (2005). Idea zrealizowana w **ggplot2** prowadzi do eleganckiej składni, która jest konsekwentna i intuicyjna. W tej składni zastosowano warstwową strukturę oraz mapowanie danych na estetyki, co ułatwia zrozumienie i dostosowywanie wyglądu wykresów do potrzeb użytkownika. Biblioteka zapewnia wsparcie do konstrukcji różnorodnych typów wykresów. Poza podstawowymi rodzajami wykresów, jak histogram, wykres punktowy, liniowy, słupkowy, pudełkowy, możliwa jest konstrukcja wielu innych typów wykresów, a możliwości te znacznie się zwiększają po zainstalowaniu dodatkowych pakietów rozszerzających. Pakiet **ggplot2** posiada bardzo obszerną dokumentację i przyjazny system pomocy. Liczna jest też wspólnota aktywnych użytkowników, co znacznie ułatwia uzyskanie wsparcia w różnych problemach z prezentacjami graficznymi.

* Fisher (1925, s. 27) – tłumaczenie własne.

5.1. Podstawy pracy z pakietem ggplot2

Termin „ggplot" jest skrótem od „gramatyki grafiki dla wykresów" („grammar of graphics plot"). W 1999 roku Leland Wilkinson (2005) przedstawił propozycję ogólnego schematu wizualizacji danych. Implementację tego systemu w postaci pakietu **ggplot2** dla programu R przedstawił Hadley Wickham (2009). Od tego czasu system ten stał się praktycznie standardem w wizualizacji wyników badań w nauce. Funkcja *ggplot* w pakiecie **ggplot2** tworzy wykres z wykorzystaniem tej gramatyki (Aldrich i Rodriguez 2013). Tradycyjnie wykresy są klasyfikowane na wykresy punktowe, liniowe i słupkowe w zależności od wyglądu, ale **ggplot2** jest zaprojektowany do pracy w sposób warstwowy. Schemat ten dzieli wykresy na komponenty, jak na przykład skale i warstwy.

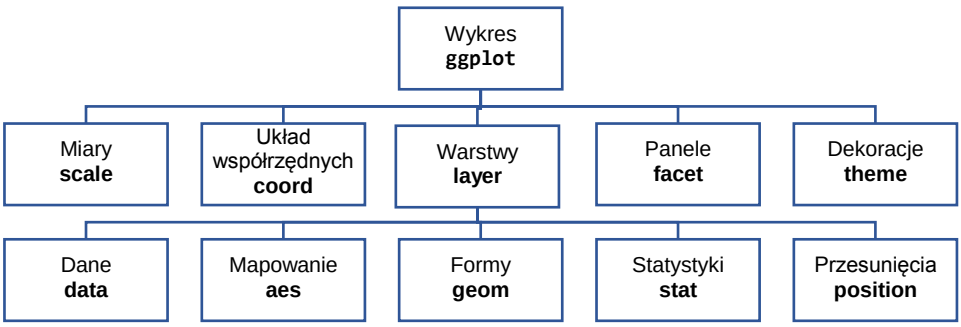
W systemie **ggplot2** na ostateczną postać grafiki składają się:

– zbiór danych oraz zmienne (data),
– zestaw mapowań zmiennych do estetyki (aes),
– jedna lub więcej warstw, z których każda składa się z obiektu geometrycznego (geom_*).

Dodatkowo dla kontroli rodzaju tworzonego wykresu (punkty, linie i tym podobne), statystycznej transformacji (stat_*) oraz regulacji pozycji mogą być wykorzystywane kolejne komponenty:

– skala – scale_* – do sterowania odwzorowaniem danych na atrybuty estetyczne; skale są wspólne dla wszystkich warstw, aby zapewnić spójne mapowanie od danych do estetyki,
– układ współrzędnych – coords_*,
– specyfikacja paneli – facet_* (grafika z panelami).

Schematyczny obraz układu gramatyki grafiki zrealizowany w pakiecie **ggplot2** przedstawiono na rysunku 5.1.



Rysunek 5.1. Gramatyka języka wizualizacji danych w ggplot2

Źródło: opracowanie własne na podstawie Biecek (2014).

Metody graficzne pozwalają pokazać strukturę danych i przedstawić wyniki analizy danych. Są one na ogół łatwiejsze w interpretacji niż tabele, które są dobre do podawania dokładnych wartości, a także raporty statystyczne, które z kolei okazują się odpowiednie do podawania szacunków, formalnych porównań i przekazują więcej informacji jakościowych. Antony Unwin (2015) podkreśla, że najprościej można pokazać potencjał prezentacji graficznej poprzez przedstawienie przykładów. Takie prezentacje pozwalają na wstępne spojrzenie na zbiór danych, a nie są kompletnymi analizami. W tym rozdziale przedstawione zostaną właśnie tego typu przykłady konstrukcji wybranych wykresów. Niekiedy konstrukcja taka będzie obejmowała kilka etapów. W omawianych przypadkach wykreślona zostanie graficzna prezentacja zbioru danych, aby ujawnić niektóre informacje w nim zawarte.

5.2. Przygotowanie do przeprowadzenia analizy graficznej

5.2.1. Pakiet ggplot2 i wybrane biblioteki rozszerzające

Konstrukcja wykresów z wykorzystaniem pakietu **`ggplot2`** musi być poprzedzona załadowaniem tej biblioteki. Przed rozpoczęciem pracy warto również załadować inne, przydatne w dalszej pracy biblioteki. Do uruchomienia kodów przedstawionych w tym rozdziale niezbędne staje się zainstalowanie bibliotek ujętych wraz z krótką charakterystyką w tabeli 5.1.

Tabela 5.1. Biblioteki wykorzystane w bieżącym rozdziale

Biblioteka	Opis
`ggplot2`	Konstrukcja eleganckich wizualizacji w oparciu o gramatykę grafiki
`patchwork`	Konstrukcja kompozycji wykresów
`ggpubr`	Konstrukcja wykresów do publikacji
`gganimate`	Gramatyka animowanej grafiki
`ggthemes`	Dodatkowe motywy, skale i geom dla `ggplot2`
`dplyr`	Gramatyka manipulacji danymi
`tidyverse`	Zestaw kilku pakietów ułatwiających pracę z danymi

Źródło: opracowanie własne na podstawie CRAN (b.r.).

W tabeli 3.3 wskazano typowe możliwości reprezentacji graficznej w zależności od liczby i skali pomiarowej analizowanych zmiennych. W pakiecie **`ggplot2`** przewidziano wiele różnych możliwych reprezentacji geometrycznych danych. Listę tych reprezentacji przedstawiono w tabeli 5.2. Należy jednak podkreślić, że po zainstalowaniu wybranych pakietów rozszerzających uzyskać

można wiele innych reprezentacji, jak na przykład geom_map, geom_mosaic i tym podobne. Niektóre z takich reprezentacji geometrycznych zostaną przedstawione w rozdziale 6.

Tabela 5.2. Reprezentacje geom w pakiecie `ggplot2`

geom	Opis
abline	Linia opisana przez współczynnik kierunkowy i wyraz wolny
area	Wykres powierzchniowy
bar	Wykres słupkowy
blank	Czyste pole wykresu
boxplot	Wykres pudełkowy
contour	Wykres konturowy
crossbar	Symbol znaku plus
density	Estymacja funkcji gęstości
density_2d	Konturowy wykres gęstości 2D
errorbar	Słupki błędów
histogram	Histogram
hline	Linia pionowa
interval	Podstawa dla konstrukcji wykresów z przedziałami
iitter	Rozrzucenie punktów zapobiegające ich nakładaniu się
line	Wykres liniowy
linerange	Przedział reprezentowany przez pionową linię
path	Połączenie obserwacji zgodnie z zadanym porządkiem
point	Punkt
pointrange	Przedział reprezentowany przez pionową linię z punktem w środku
poligon	Konstrukcja wielokątów
quantile	Dodanie linii kwantylowych z regresji kwantylowej
ribbon	Wykres wstęgi dla ciągłych wartości *x*
rug	Zaznaczenie obserwacji na brzegu wykresu
segment	Pojedyncze segmenty linii
smooth	Dodanie wygładzenia
step	Wykres schodkowy
text	Wprowadzenie adnotacji tekstowej
tile	Do konstrukcji wykresów kafelkowych
vline	Linia pozioma

Źródło: opracowanie własne na podstawie Wickham (2009).

W praktycznych zastosowaniach wybór formy wykresu wiąże się z doborem właściwej formy geometrycznej. Alboukadel Kassambara (2013) wskazuje

na następujące możliwości wykorzystania form prezentacji geometrycznych w zależności od liczby i typu zmiennych:

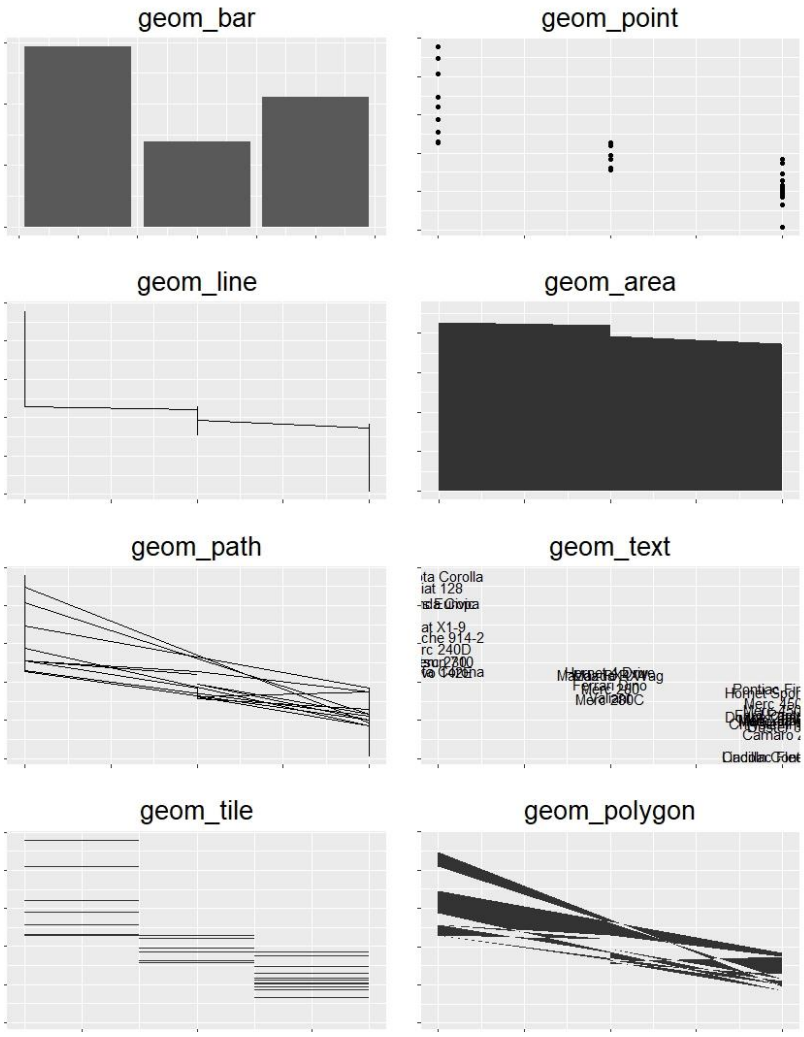
1. Dla jednej zmiennej ciągłej:
   - geom_area – dla wykresu powierzchniowego,
   - geom_density – dla wykresu gęstości,
   - geom_dotplot – dla wykresu punktowego,
   - geom_freqpoly – dla diagramu liczebności,
   - geom_histogram – dla histogramu,
   - stat_ecdf – dla wykresu empirycznej dystrybuanty,
   - stat_qq – dla wykresu kwantylowego;
2. Dla jednej zmiennej dyskretnej:
   - geom_bar – dla wykresu słupkowego;
3. Dla dwóch zmiennych ciągłych:
   - geom_point – dla wykresu punktowego,
   - geom_smooth – dla dodania wygładzonej linii regresji,
   - geom_quantile – dla dodania linii kwantylowej,
   - geom_rug – dla dodania wartości na brzegach,
   - geom_jitter – dla uniknięcia nakładania się punktów,
   - geom_text – dla dodania tekstu (np. etykiet obserwacji);
4. Dla zmiennej dyskretnej i ciągłej:
   - geom_boxplot – dla wykresu pudełkowego,
   - geom_violin – dla wykresu wiolinowego,
   - geom_dotplot – dla wykresu punktowego,
   - geom_jitter – dla uniknięcia nakładania się punktów,
   - geom_line – dla wykresu liniowego,
   - geom_bar – dla wykresu słupkowego;
5. Dla dwóch zmiennych ciągłych – rozkład empiryczny:
   - geom_point – dla wykresu punktowego,
   - geom_bin2d – dla wykresu heatmap,
   - geom_hex – dla wykresu hexagonalnego,
   - geom_density_2d – dla dwuwymiarowego wykresu gęstości;
6. Dla wykreślenia funkcji:
   - geom_area – dla wykresu powierzchniowego,
   - geom_line – dla wykresu liniowego,
   - geom_step – dla wykresu schodkowego.

Dla wybranej formy reprezentacji geometrycznej możliwe jest ustawienie wartości różnych parametrów, co pozwala na znaczne rozszerzenie możliwości prezentacji graficznej. Wybór właściwej reprezentacji geometrycznej (geom) okazuje się bardzo ważny. Zwykle jest on jednoznacznie wyznaczony celem analizy. Chcąc skonstruować histogram, należy wybrać reprezentację geom_histogram.

Niekiedy dla ustalonego zbioru danych można wykorzystać różne reprezentacje geometryczne i każda z nich przekaże nieco inne informacje odbiorcy. Na rysunku 5.2 przedstawiono osiem różnych reprezentacji geometrycznych dla jednego ustalonego wykresu (obiekt **p**) zadanego następującą komendą.

```
p<-ggplot(mtcars,aes(cyl,mpg,label=rownames(mtcars)))
```

Poszczególne pola wykresu na rysunku 5.2 niosą bardzo różny przekaz. Tylko niektóre z nich mogą dawać logiczny przekaz z sensowną interpretacją odnośnie do prezentowanego obiektu.



Rysunek 5.2. Wybrane reprezentacje geometryczne dla ustalonego obiektu p

Źródło: opracowanie własne w programie R.

5.2.2. Charakterystyka zbioru danych mtcars

Antony Unwin (2015) podkreśla, że najprościej pokazać możliwości prezentacji graficznej poprzez przedstawienie przykładów. Wszystkie przykłady wykresów ujęte w tym rozdziale oraz większość w kolejnym będą zaprezentowane na podstawie danych ze zbioru **mtcars**. Odniesienie się do danych z jednego zbioru pozwoli Czytelnikowi skoncentrować się na wykorzystywanych metodach prezentacji graficznej oraz na stosowanych narzędziach do wizualizacji danych. Zbiór **mtcars** jest dostępny dla użytkownika już po zainstalowaniu programu R. Pochodzi on z badań przeprowadzonych przez MotorTrend w latach 1973-1974. Podstawowe informacje dotyczące tego zbioru zamieszczono w tabeli 5.3.

Tabela 5.3. Charakterystyka zmiennych zbioru mtcars

Zmienna	Skala pomiarowa	Opis
mpg	Liczbowa	Liczba mil amerykańskich przejechanych na galonie paliwa
cyl	Porządkowa	Liczba cylindrów (wartości: 3, 4 i 5)
disp	Liczbowa	Pojemność silnika
hp	Liczbowa	Moc silnika
drat	Liczbowa	Przełożenie tylnej osi
wt	Liczbowa	Waga samochodu w tysiącach funtów
qsec	Liczbowa	Czas przejazdu, ¼ mili
vs	Binarna	Silnik (0 – V-kształtny, 1 – prosty)
am	Binarna	Skrzynia biegów (0 – automatyczna, 1 – ręczna)
gear	Porządkowa	Liczba biegów (bez wstecznego, wartości: 4, 6 i 8)
carb	Liczbowa	Liczba gaźników

Źródło: opracowanie własne na podstawie Henderson i Velleman (1981).

Dla wyświetlenia początkowych rekordów wybranych zmiennych zbioru danych można wykorzystać funkcję *head()*.

```
# Początkowe rekordy zbioru danych dla wybranych zmiennych
data(mtcars)
head(mtcars[,c(1,2,4,6,7,9,10)])
##                   mpg cyl  hp    wt  qsec am gear
## Mazda RX4         21.0   6 110 2.620 16.46  1    4
## Mazda RX4 Wag     21.0   6 110 2.875 17.02  1    4
## Datsun 710        22.8   4  93 2.320 18.61  1    4
```

```
## Hornet 4 Drive    21.4   6 110 3.215 19.44  0    3
## Hornet Sportabout 18.7   8 175 3.440 17.02  0    3
## Valiant           18.1   6 105 3.460 20.22  0    3
```

Zmienne *gear* (liczba biegów), *cyl* (liczba cylindrów) oraz *am* (rodzaj skrzyni biegów) w zbiorze są ujęte jako zmienne liczbowe. Wygodniej będzie rozważać je jako czynniki. Konwersja zmiennych numerycznych na zmienne jakościowe jest często stosowana w analizie danych, zwłaszcza gdy zmienne zapisane jako liczbowe mają charakterystykę nominalną lub porządkową, a jednocześnie nie są wykorzystywane w dalszych obliczeniach jako wielkości liczbowe. Poniższe komendy wykonują konwersję zmiennych numerycznych w ramce danych **mtcars** na zmienne jakościowe (factor) przy użyciu funkcji *factor*.

```
# Załadowanie zbioru danych
data(mtcars)
# Zmiana trzech zmiennych na czynniki (factors)
mtcars$gear <- factor(mtcars$gear,levels=c(3,4,5),labels=c('3
biegi', '4 biegi', '5 biegów'))
mtcars$am <- factor(mtcars$am, levels = c(0,1), labels =
c('automatyczna','ręczna'))
mtcars$cyl <- factor(mtcars$cyl,levels=c(4,6,8),labels=c('4 cy-
lindry', '6 cylindrów', '8 cylindrów'))

head(mtcars[,1:5])
##                    mpg         cyl disp  hp drat
## Mazda RX4         21.0 6 cylindrów  160 110 3.90
## Mazda RX4 Wag     21.0 6 cylindrów  160 110 3.90
## Datsun 710        22.8  4 cylindry  108  93 3.85
## Hornet 4 Drive    21.4 6 cylindrów  258 110 3.08
## Hornet Sportabout 18.7 8 cylindrów  360 175 3.15
## Valiant           18.1 6 cylindrów  225 105 2.76
# Podsumowanie dla wybranych zmiennych
summary(mtcars[,1:3])
##       mpg                 cyl          disp
##  Min.   :10.40   4 cylindry :11   Min.   : 71.1
##  1st Qu.:15.43   6 cylindrów: 7   1st Qu.:120.8
##  Median :19.20   8 cylindrów:14   Median :196.3
##  Mean   :20.09                    Mean   :230.7
##  3rd Qu.:22.80                    3rd Qu.:326.0
##  Max.   :33.90                    Max.   :472.0
```

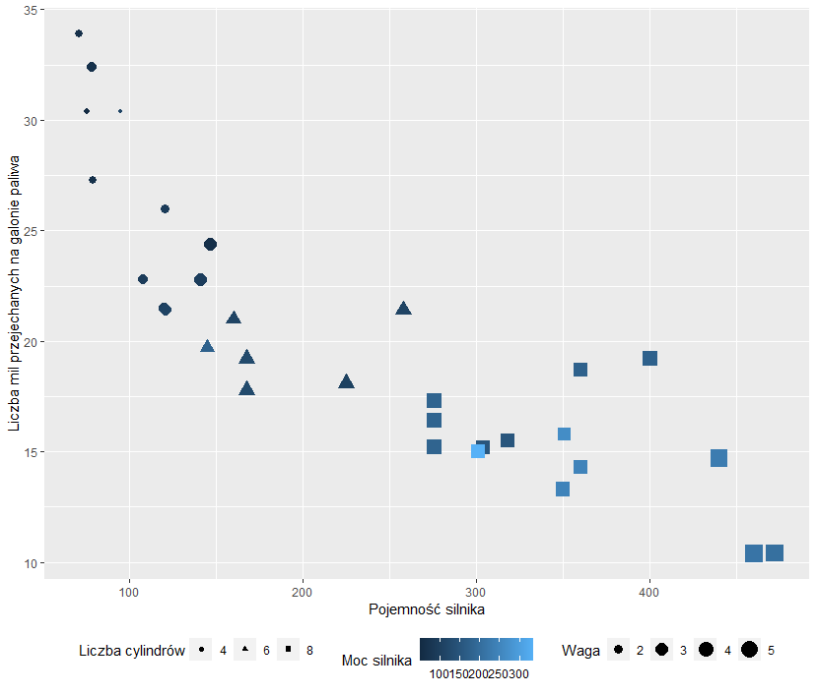
W przypadku konieczności przywrócenia oryginalnej postaci danych wystarczy wykonać komendę, która ładuje do pamięci pierwotną postać zbioru **mtcars**.

```
data(mtcars)
```

Na rysunku 5.3 przedstawiono liczbę mil przejechanych przez samochód na jednym galonie paliwa (*mpg*) w zależności od pojemności silnika (*disp*). Dodatkowo na wykresie zamieszczone zostały informacje dotyczące liczby cylindrów, wagi samochodu oraz mocy silnika. Do konstrukcji wykresu na rysunku 5.3 wykorzystano pięć następujących zmiennych:

– oś OX (pojemność skokowa silnika),
– oś OY (efektywność paliwowa),
– kolor punktów danych (moc silnika),
– wielkość danych (waga samochodu),
– kształt punktów danych (liczba cylindrów).

Cztery z pięciu zmiennych (pojemność silnika, zużycie paliwa, moc i masa) to zmienne numeryczne ciągłe. Ostatnią ze zmiennych (liczba cylindrów) można uznać za liczbową dyskretną lub jakościową rejestrowaną na skali porządkowej.



Rysunek 5.3. Liczba przejechanych mil na jednym galonie paliwa w zależności od pojemności skokowej silnika względem liczby cylindrów, mocy silnika i wagi samochodu (modele z lat 1973-1974)

Źródło: opracowanie własne w programie R.

5.3. Konstrukcja wybranych typów wykresów w pakiecie ggplot2

W tym punkcie zostaną przedstawione zasady konstrukcji wybranych wykresów z wykorzystaniem pakietu **ggplot2**. Na wstępie niezbędne jest załadowanie bibliotek, które będą wykorzystywane w tej części. Wykaz tych bibliotek wraz z krótką charakterystyką został zamieszczony w tabeli 5.1. Poniższe komendy powodują załadowanie bibliotek graficznych oraz wykorzystywanych do operacji na zbiorach danych.

```
library(ggplot2)
library(patchwork)
library(ggpubr)
library(gganimate)
library(ggthemes)
library(dplyr)
library(tidyverse)
```

W tym rozdziale pracy skoncentrowano się na zasadach konstrukcji wykresów zgodnie z ideą Grammar of Graphics (Wilkinson 2005). Z tego powodu tytuły wykresów dotyczą zasadniczo zastosowanej metody graficznej (rodzaju lub sposobu konstrukcji wykresu) prezentacji danych. Tylko w niektórych przypadkach zamieszczono drugą część tytułu. W takich sytuacjach ta część odnosi się do opisu prezentowanego na wykresie zjawiska. W przypadku przeprowadzania analizy danych nie umieszcza się pierwszej z tych części, ponieważ tytuł wykresu powinien odnosić się do zawartości obszaru wykresu. W opracowaniach jak książki, artykuły, publikacje tytuł wykresu nie jest umieszczany w obrębie pola wykresu, a jedynie jako podpis pod rysunkiem. Do wszystkich wykonanych wykresów podano kody w języku R.

5.3.1. Wykres punktowy

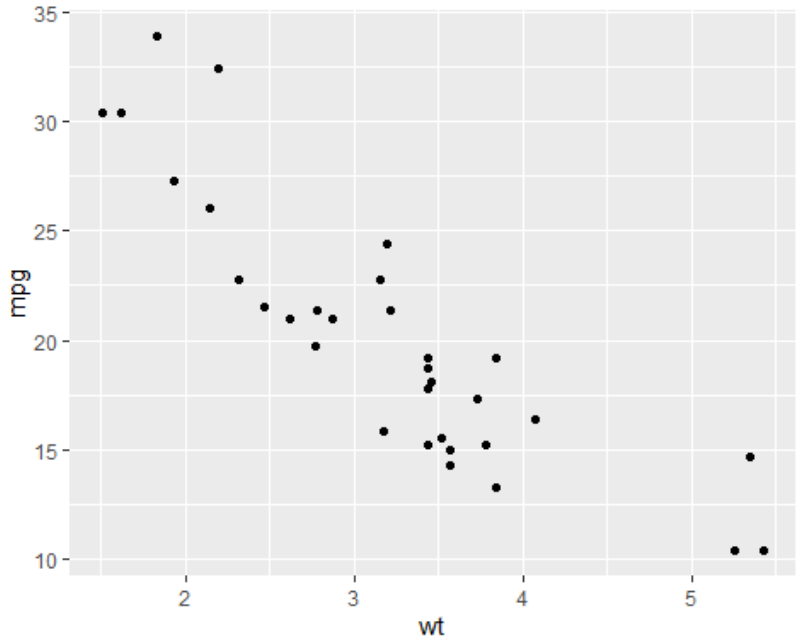
Funkcja *ggplot* wymaga wskazania zbioru danych (data), dla którego ma zostać skonstruowany wykres. Jako drugi argument (aes) należy podać zmienne lub zmienną, które mają być uwzględnione na wykresie. Parametr aes pozwala określić tak zwaną estetykę wykresu (aesthetics). Następnie po znaku „+" należy wskazać warstwę – sposób geometrycznej reprezentacji danych. Może to być na przykład reprezentacja (por. tabela 5.2): punkty (geom_point), linie (geom_line),

histogram (geom_histogram), boxplot (geom_boxplot) i tym podobne. Podstawowa konstrukcja dla wykresu rozrzutu (wykres punktowy dla dwóch zmiennych *wt* i *mpg* ze zbioru **mtcars**) jest następująca.

```
# Podstawowa konstrukcja wykresu w ggplot2
ggplot(data=mtcars,aes(x=wt,y=mpg))+geom_point()
```

Efektem powyższej komendy jest wykres przedstawiony na rysunku 5.4. Na wykresie na osiach rozmieszczono zmienne waga (*wt*) oraz liczba przejechanych mil na galonie paliwa (*mpg*). Układ punktów wskazuje na zależność liniową ujemną pomiędzy zmiennymi. Dokładnie taki sam efekt jak na rysunku 5.4 uzyska się bez podawania nazw parametrów funkcji: 'data=', '*x*=' i '*y*='. Komenda przyjmuje wówczas następującą, prostszą postać.

```
# Zapis bez podawania nazw parametrów 'data=', 'x=' i 'y='
ggplot(mtcars,aes(wt,mpg))+geom_point()
```



Rysunek 5.4. Rezultat konstrukcji wykresu z rysunku 5.3 za pomocą funkcji `ggplot`

Źródło: opracowanie własne w programie R.

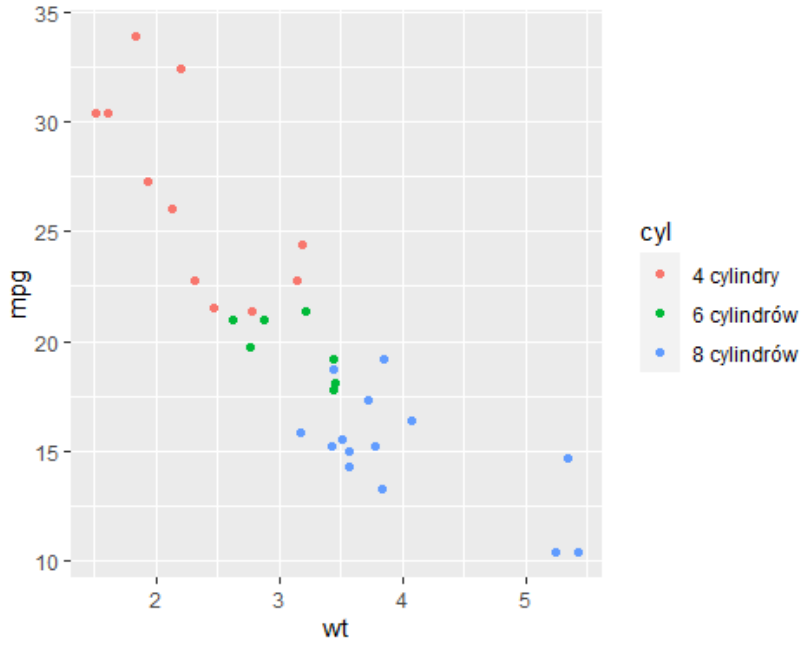
Dla przejrzystości zapisu wygodnie jest stosować konwencję, w której każda warstwa wykresu jest ujmowana w kolejnej linii. Taka forma zapisu będzie stosowana w dalszej części. W prezentowanym przypadku polecenie przybierze następującą formę.

```
# Konwencja zapisu warstwy w nowej linii
ggplot(mtcars,aes(wt,mpg))+
  geom_point()
```

Wynik tej komendy będzie taki sam jak zaprezentowany na rysunku 5.4.

Poza zmiennymi *x* i *y* w aes możliwe jest wskazanie dodatkowych zmiennych z przypisaniem do nich takich atrybutów jak kolor (color), rozmiar (size) i kształt (shape). Poniżej przedstawiono postać komendy wprowadzającą przypisanie różnych kolorów do punktów w zależności od liczby cylindrów.

```
# Wyróżnienie grup ze względu na liczbę cylindrów
ggplot(mtcars,aes(wt,mpg,color=cyl))+geom_point()
```

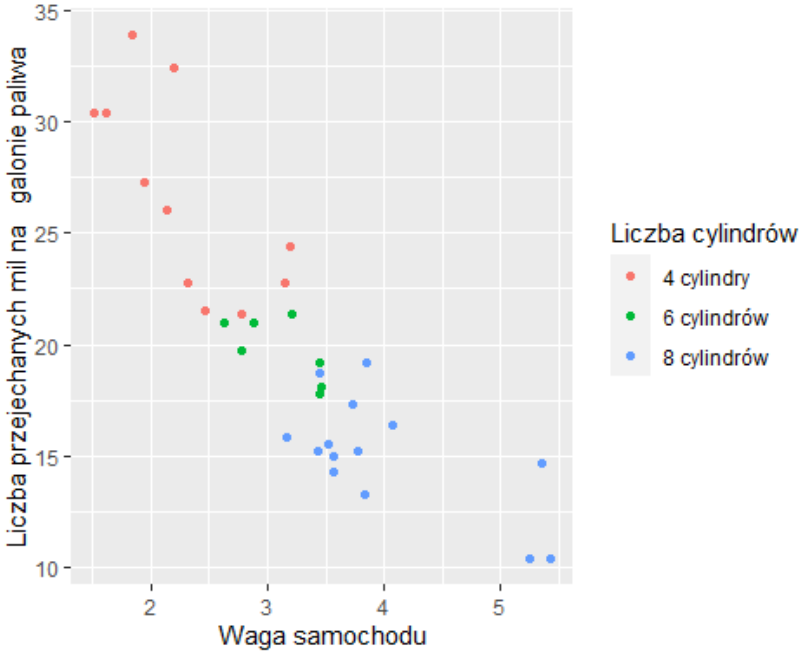


Rysunek 5.5. Dodanie do wykresu z rysunku 5.4 kolorów punktów w zależności od liczby cylindrów

Źródło: opracowanie własne w programie R.

Na rysunku 5.5 przedstawiono liczbę mil przejechanych na galonie paliwa w zależności od wagi samochodu z wyróżnieniem kolorami samochodów ze względu na liczbę cylindrów. Automatycznie została dołączona po prawej stronie legenda. Podobnie jak na poprzednim wykresie (rysunek 5.4) na osiach są zapisane nazwy zmiennych *wt* oraz *mpg*. Nie jest to czytelne dla odbiorcy. Wszystkie te opisy można zmienić, dodając po znaku „+" warstwę etykiet (labs) jak w poniższym kodzie.

```
# Wprowadzenie etykiet osi i legendy
ggplot(mtcars,aes(wt,mpg,color=cyl))+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba przejechanych mil na galonie
paliwa',color='Liczba cylindrów')
```



Rysunek 5.6. Wykres z rysunku 5.5 po dodaniu opisu osi oraz tytułu legendy

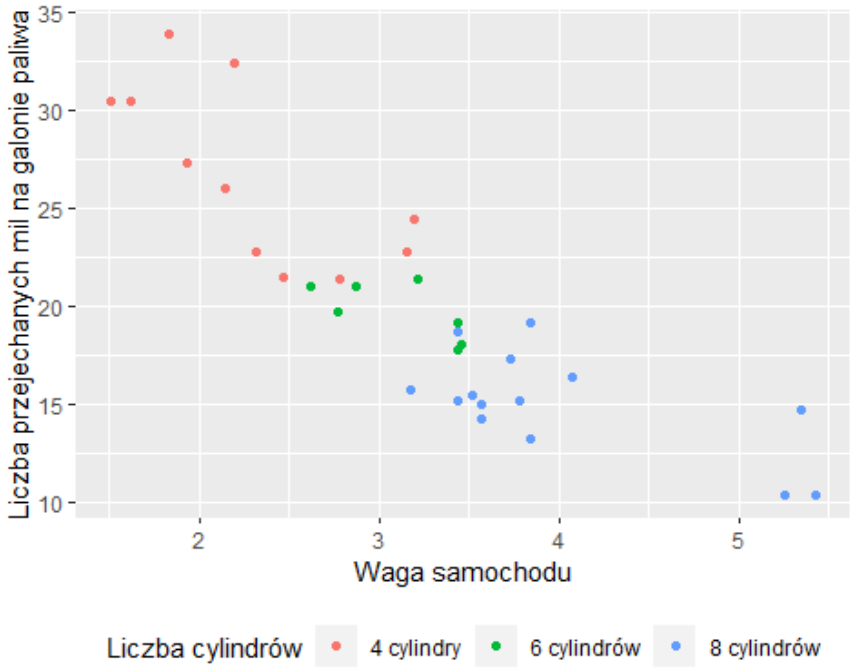
Źródło: opracowanie własne w programie R.

Wykonanie powyższych komend prowadzi do uzyskania wykresu jak na rysunku 5.6. Zarówno opisy osi, jak i legenda są dobrze czytelne dla odbiorcy.

Niekiedy dla zwiększenia czytelności wykresu wskazane jest umieszczenie legendy w innym miejscu niż po prawej stronie wykresu. Można to zrealizować poprzez dodanie warstwy motywu (theme) ze wskazaniem położenia legendy na przykład na dole (bottom).

```
# Umieszczenie legendy pod wykresem
ggplot(mtcars,aes(wt,mpg,color=cyl))+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba przejechanych mil na galonie
paliwa',color='Liczba cylindrów')+
  theme(legend.position='bottom')
```

W efekcie otrzymuje się wykres (por. rysunek 5.7), na którym na osi OX znajduje się *wt* (waga samochodu), a na osi OY *mpg* (liczba przejechanych mil na galonie paliwa). Kolor punktów jest zależny od wariantów zmiennej dyskretnej *cyl* (liczba cylindrów), co pozwala na łatwe rozróżnienie samochodów o różnej liczbie cylindrów. Tytuły osi OX i OY oraz legendy zostały nadane za pomocą funkcji labs. Pozycja legendy została ustawiona na dole za pomocą warstwy theme. Ta ostatnia warstwa pozwala nie tylko na ustawienie położenia legendy, ale także na modyfikację wielu innych parametrów, jak na przykład: obramowania pola legendy, zmianę kierunku tekstu w polu legendy lub przy osiach, ustawienie tła legendy, zmianę sposobu wyświetlania osi i tym podobne. Niektóre parametry tej warstwy zostaną wykorzystane w dalszej części.

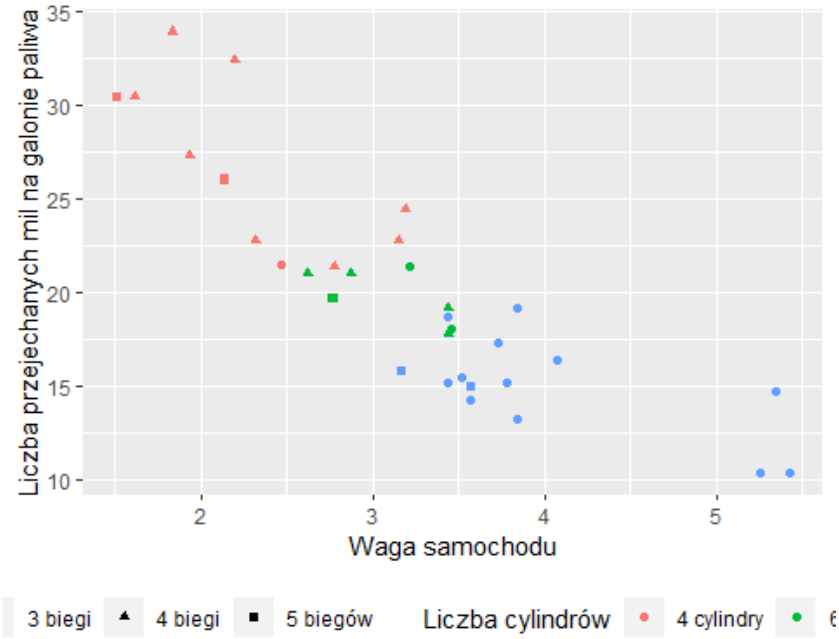


Rysunek 5.7. Wykres z rysunku 5.6 po zmianie położenia legendy

Źródło: opracowanie własne w programie R.

Wcześniej podkreślono, że jako parametry aes mogą zostać podane również shape oraz size. Możliwości te pokazują dwa kolejne fragmenty kodu, gdzie z tymi parametrami zostaną powiązane dodatkowo liczba biegów (zmienna dyskretna) oraz moc silnika (zmienna ciągła).

```
# Wyróżnienie liczby biegów poprzez kształt
ggplot(mtcars,aes(wt,mpg,color=cyl,shape=gear))+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba przejechanych mil na galonie
paliwa',color='Liczba cylindrów',shape='Liczba biegów')+
  theme(legend.position='bottom')
```

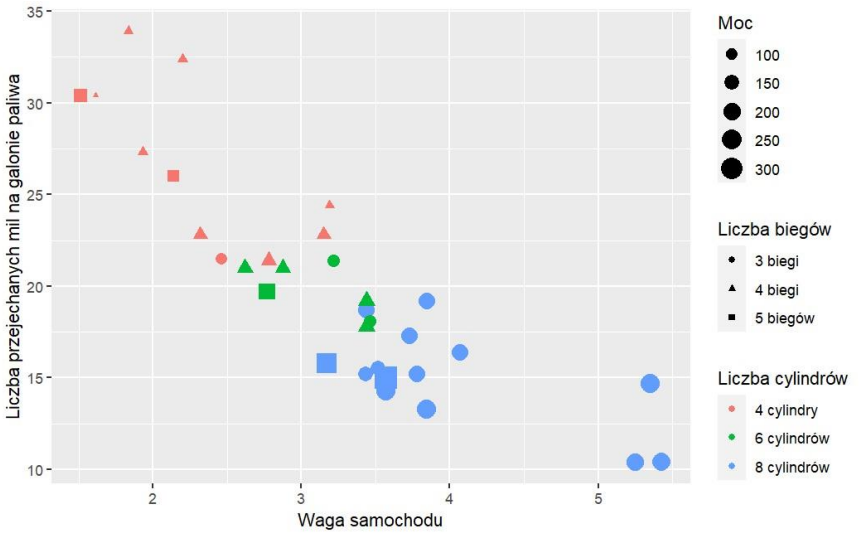


Rysunek 5.8. Wykres z rysunku 5.7 po wprowadzeniu kształtu punktów związanego z liczbą biegów

Źródło: opracowanie własne w programie R.

Na rysunku 5.8 rozróżnione poprzez kształt punktów zostały samochody o różnej liczbie biegów (*gear* – zmienna dyskretna), a na rysunku 5.9 dodatkowo wielkość punktów pozwala na rozróżnienie samochodów o różnej mocy silnika (*hp* – zmienna ciągła).

```
# Wyróżnienie mocy silnika poprzez rozmiar punktów
ggplot(mtcars,aes(wt,mpg,color=cyl,shape=gear,size=hp))+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba przejechanych mil na galonie
paliwa',color='Liczba cylindrów',shape='Liczba
biegów',size='Moc')+
  theme(legend.position='right')
```



Rysunek 5.9. Wykres z rysunku 5.8 po zmianie kształtu punktów związanego z liczbą biegów (zmienna dyskretna) oraz rozmiaru punktów związanego z mocą silnika (zmienna ciągła)

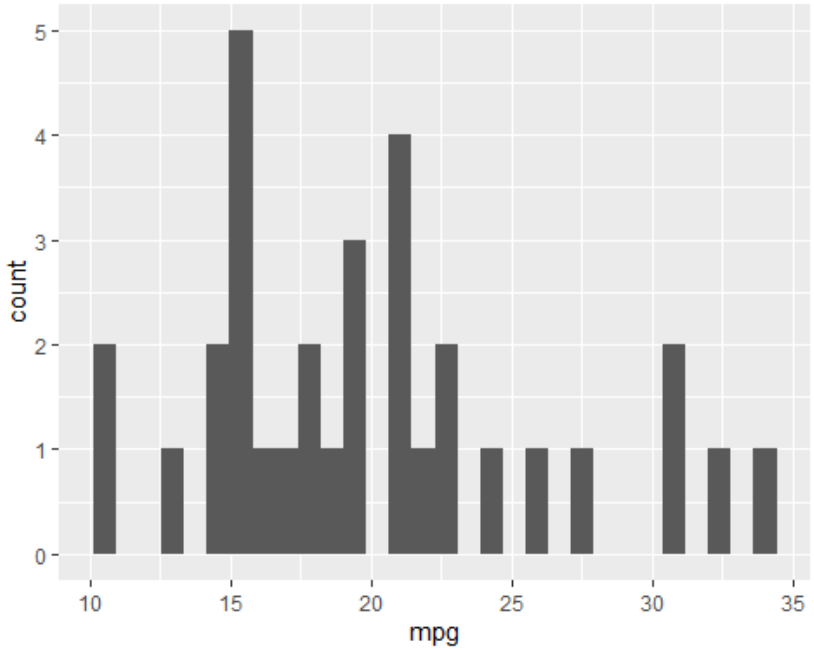
Źródło: opracowanie własne w programie R.

5.3.2. Histogram i krzywa gęstości

Przedstawione w poprzednim punkcie sposoby konstrukcji grafiki dotyczyły wyłącznie wykresów punktowych, a konkretnie różnych wersji wykresów rozrzutu. Pakiet **ggplot2** umożliwia konstrukcję różnego typu prezentacji graficznych. Do najczęściej stosowanych wykresów w statystycznej analizie danych należy histogram. O ile przy konstrukcji wykresu rozrzutu należało podać dwie zmienne liczbowe, to histogram jest wykreślany dla jednej ciągłej zmiennej numerycznej. Podstawowa konstrukcja histogramu dla zmiennej *mpg* jest następująca.

```
# Konstrukcja histogramu dla zmiennej mpg
ggplot(mtcars,aes(mpg))+
  geom_histogram()
```

Po wykonaniu powyższej komendy otrzymuje się wykres jak na rysunku 5.10.

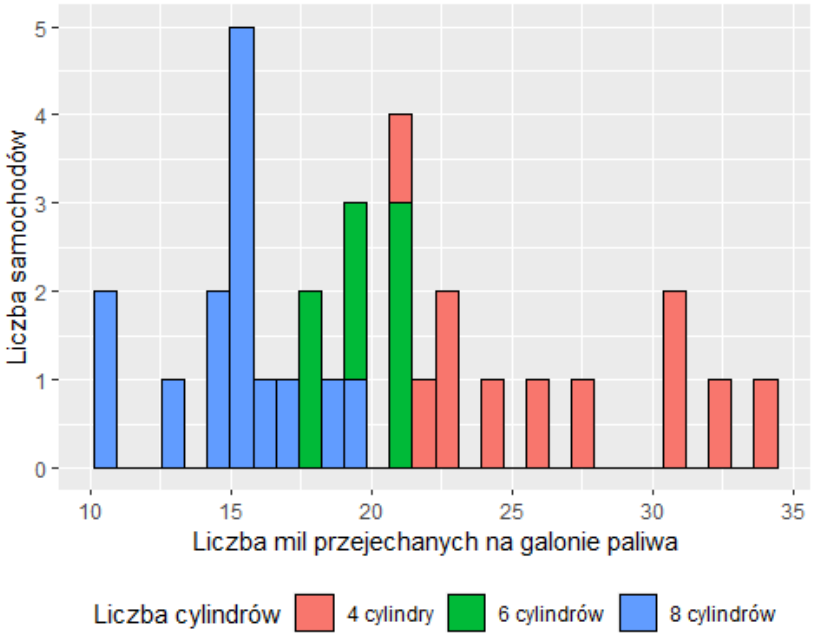


Rysunek 5.10. Histogram w podstawowej konstrukcji. Samochody według liczby mil przejechanych na galonie paliwa

Źródło: opracowanie własne w programie R.

Podobnie jak w przypadku konstrukcji wykresu punktowego, tak również przy konstrukcji histogramu można wyróżnić kolorem samochody z różną liczbą cylindrów. W przypadku wykresu rozrzutu wykorzystano parametr color. Możliwe jest zastosowanie tego samego parametru przy konstrukcji histogramu, ale spowoduje to zmianę koloru wyłącznie obramowania słupków. Dla zmiany koloru wypełnienia słupków należy odwołać się do parametru fill. Dodatkowo zostaną wprowadzone opisy osi i tytuł legendy. Rezultat poniższej komendy przedstawiono na rysunku 5.11.

```
# Wykorzystanie parametru fill w konstrukcji histogramu
ggplot(mtcars,aes(mpg,fill=cyl))+
  geom_histogram(color='black')+
  labs(fill='Liczba cylindrów',x='Liczba mil przejechanych na
galonie paliwa',y='Liczba samochodów')+
  theme(legend.position='bottom')
```

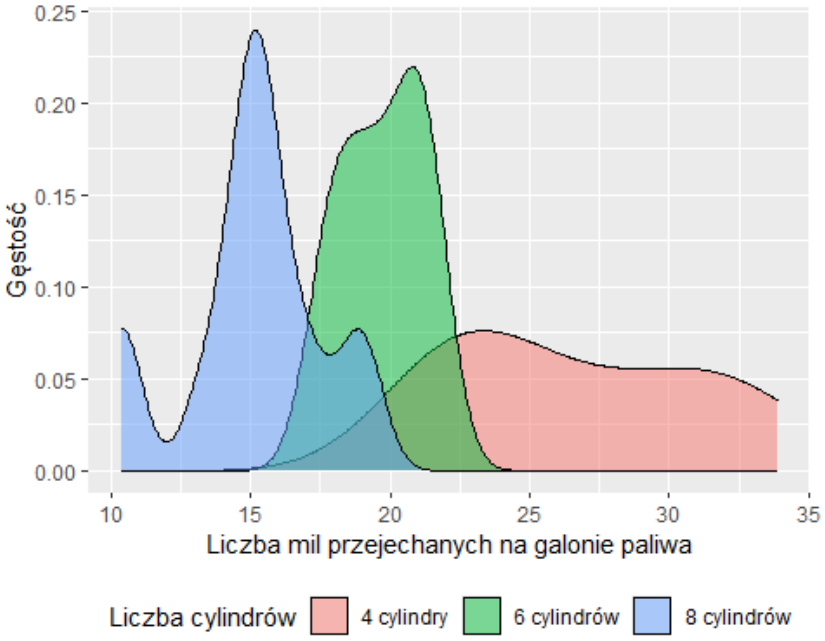


Rysunek 5.11. Histogram z wyróżnieniem kolorem w zależności od liczby cylindrów. Samochody według liczby mil przejechanych na galonie paliwa względem liczby cylindrów

Źródło: opracowanie własne w programie R.

Histogram jest estymatorem nieznanej teoretycznej funkcji gęstości (estymacja nieparametryczna) badanej zmiennej. Inną oceną funkcji gęstości jest jej empiryczna postać uzyskana na podstawie próby. Zmieniając warstwę reprezentacji geometrycznej 'histogram' na 'density' jak w poniższym kodzie, otrzymuje się wykres oceny funkcji gęstości jak na rysunku 5.12.

```
# Konstrukcja gęstości z wypełnieniem względem cyl
ggplot(mtcars,aes(mpg,fill=cyl))+
  geom_density(alpha=0.5)+
  labs(fill='Liczba cylindrów',x='Liczba mil przejechanych na
galonie paliwa',y='Gęstość')+
  theme(legend.position='bottom')
```



Rysunek 5.12. Estymacja gęstości z wyróżnieniem kolorem w zależności od liczby cylindrów. Samochody według liczby mil przejechanych na galonie paliwa względem liczby cylindrów

Źródło: opracowanie własne w programie R.

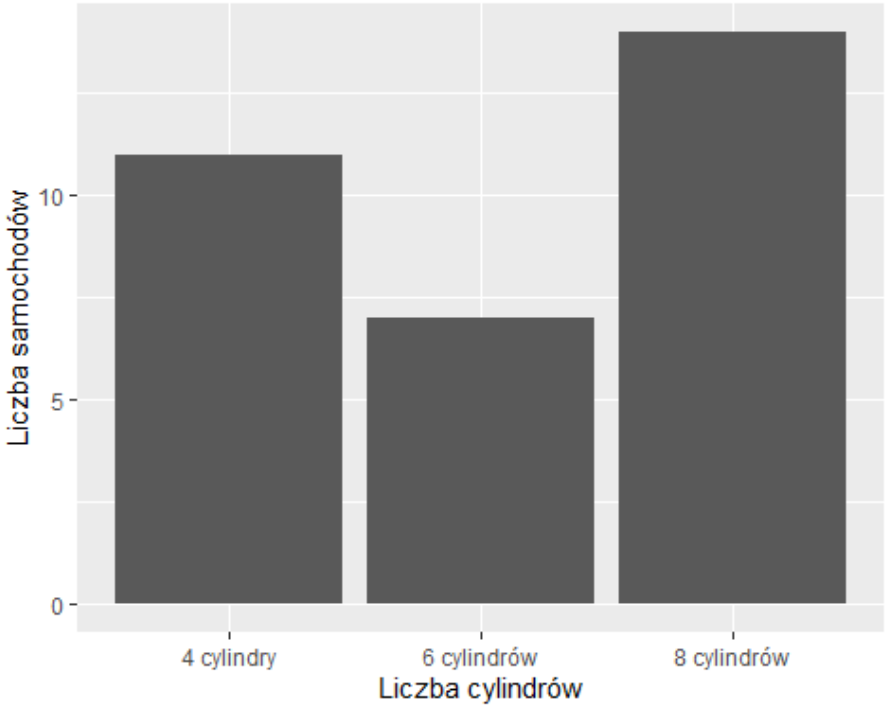
Przy konstrukcji wykresu 5.12 dodatkowo wykorzystano parametr alpha, przyjmujący wartości od 0 do 1, pozwalający na ustalenie poziomu przezroczystości tła empirycznej funkcji gęstości. Wartości parametru bliskie 0 powodują dużą przezroczystość, a bliskie 1 niewielką.

5.3.3. Wykres słupkowy

Histogram jest wykreślany dla zmiennej ciągłej. Jeśli zachodzi potrzeba przedstawienia na wykresie zmiennej liczbowej dyskretnej albo zmiennej nominalnej, to można wykorzystać wykres słupkowy. Dla otrzymania wykresu słupkowego należy wykorzystać warstwę geom_bar(). Konstrukcja wykresu słupkowego w **ggplot2** dla zmiennej *cyl* jest następująca.

```
# Wykres słupkowy dla zmiennej cyl
ggplot(mtcars,aes(cyl))+
  labs(x='Liczba cylindrów',y='Liczba samochodów')+
  geom_bar()
```

Wynik powyższej komendy przedstawiono na rysunku 5.13.

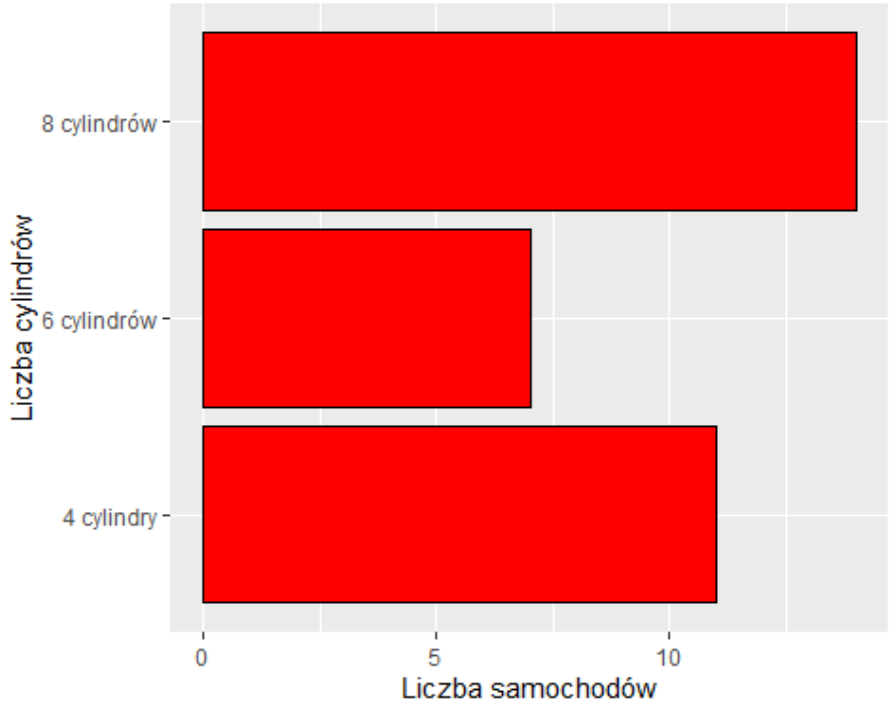


Rysunek 5.13. Wykres słupkowy w podstawowej konstrukcji. Samochody według liczby cylindrów

Źródło: opracowanie własne w programie R.

Dla otrzymania wykresu słupkowego w układzie poziomym do poprzedniej komendy należy dodać zamianę współrzędnych (coord_flip). Wynik dodania takiej warstwy, a także zmianę koloru wypełnienia słupków, przedstawiają poniższa komenda i rysunek 5.14.

```
# Wykres słupkowy poziomy - coord_flip
ggplot(mtcars,aes(cyl))+
  geom_bar(color='black',fill='red')+
  labs(x='Liczba cylindrów',y='Liczba samochodów')+
  coord_flip()
```

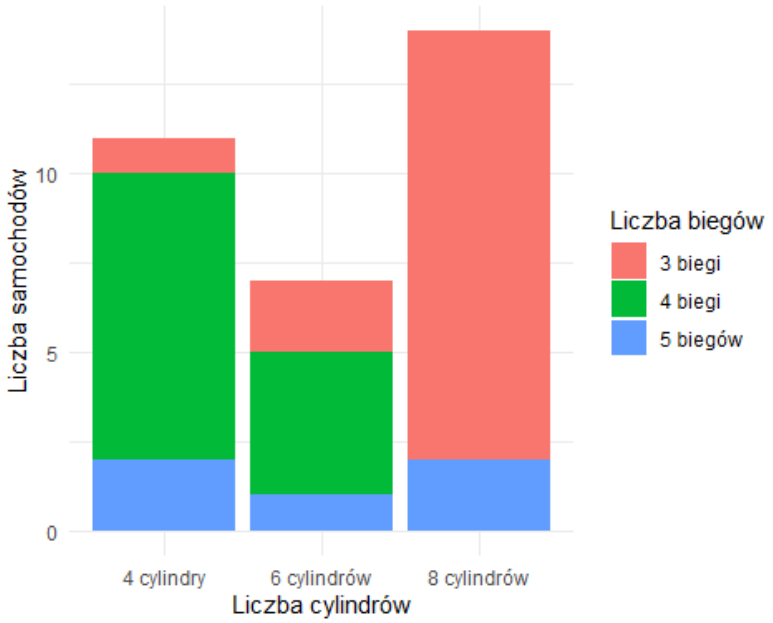


Rysunek 5.14. Wykres słupkowy po obrocie współrzędnych. Samochody według liczby cylindrów

Źródło: opracowanie własne w programie R.

Wykresy słupkowe mogą być konstruowane w różnej formie. Najczęściej obok przedstawionych w rozważaniach możliwości w analizach wykorzystuje się wykresy słupkowe nakładane i wykresy struktury. Konstrukcję wykresu słupkowego nakładanego realizuje poniższy kod.

```
# Konstrukcja wykresu słupkowego nakładanego
ggplot(mtcars, aes(x = factor(cyl), fill = factor(gear))) +
  geom_bar(position = "stack") +
  labs(x = "Liczba cylindrów", y = "Liczba samochodów") +
  scale_fill_discrete(name = "Liczba biegów") +
  theme_minimal()
```



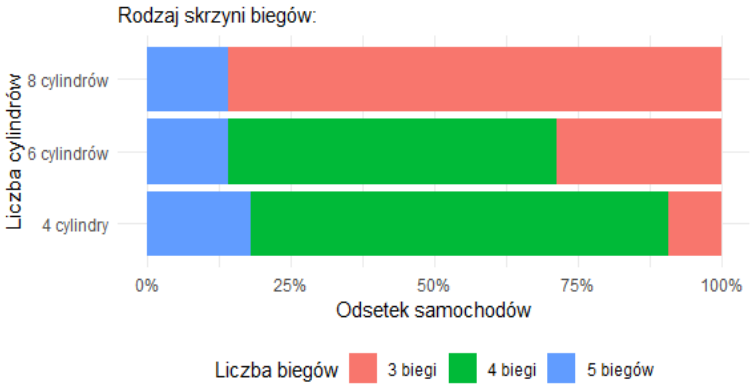
Rysunek 5.15. Wykres słupkowy nakładany. Samochody według liczby cylindrów i biegów

Źródło: opracowanie własne w programie R.

Na rysunku 5.15 przedstawiono wykres słupkowy nakładany. Poszczególne słupki przedstawiają, podobnie jak na rysunku 5.13, liczbę samochodów ze względu na liczbę cylindrów silnika. Jednak dodatkowo umieszczone kolory pozwalają na odczyt liczby samochodów w poszczególnych kategoriach ze względu na liczbę biegów.

Podobna jest konstrukcja kolejnego wykresu, który jednak zamiast liczby samochodów wskazuje na strukturę liczby samochodów dla trzech wyróżnionych grup. Realizuje to następujący kod, którego wynik przedstawia rysunek 5.16.

```
# Konstrukcja wykresu słupkowego struktury
ggplot(mtcars, aes(x = factor(cyl), fill = factor(gear))) +
  geom_bar(position = "fill") +
  labs(subtitle = "Rodzaj skrzyni biegów:",x = "Liczba cylin-
drów", y = "Odsetek samochodów") +
  scale_fill_discrete(name = "Liczba biegów") +
  coord_flip()+
  theme_minimal()+
  theme(legend.position='bottom')+
     scale_y_continuous(labels = scales::percent_format())
```

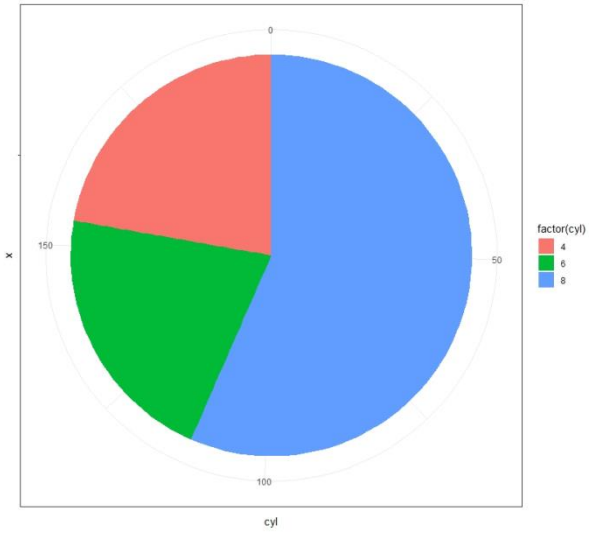


Rysunek 5.16. Wykres słupkowy struktury. Struktura samochodów ze względu na liczbę cylindrów i typ skrzyni biegów

Źródło: opracowanie własne w programie R.

Na rysunku 5.14 przedstawiono wykres słupkowy po rotacji współrzędnych. Natomiast po zmianie współrzędnych na współrzędne biegunowe (coord_poolar), jak w poniższym kodzie, uzyskuje się wykres kołowy (por. rysunek 5.17).

```
# Konstrukcja wykresu kołowego -współrzędne biegunowe
ggplot(mtcars,aes(x="",y=cyl,fill=cyl))+
  geom_col()+
  coord_polar(theta="y")
```



Rysunek 5.17. Wykres kołowy – wykres słupkowy we współrzędnych biegunowych. Struktura samochodów według liczby cylindrów

Źródło: opracowanie własne w programie R.

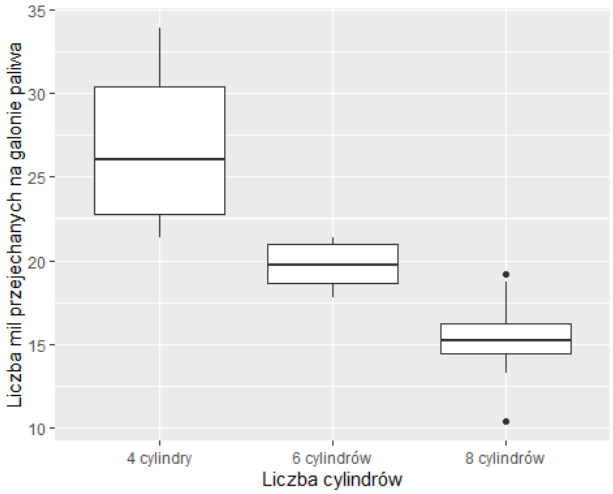
Rysunek 5.17 pozwala dostrzec, że tak często wykorzystywane w różnych prezentacjach wykresy kołowe faktycznie są reprezentacjami wykresów słupkowych we współrzędnych biegunowych.

5.3.4. Wykresy pudełkowe i wiolinowe

Bardzo często w analizie danych wykorzystywane są wykresy pudełkowe (box plot). Wykres ten jest wykreślany dla zmiennej liczbowej. Można na takim wykresie dodać zmienną jakościową jako wyróżnik kategorii, co pozwala na przeprowadzenie porównań. Dla uzyskania wykresu pudełkowego do wcześniej podanych konstrukcji należy wprowadzić warstwę geom_boxplot.

```
# Konstrukcja wykresu boxplot
ggplot(mtcars,aes(cyl,mpg))+
  geom_boxplot()+
  labs(x='Liczba cylindrów',y='Liczba mil przejechanych na
galonie paliwa')
```

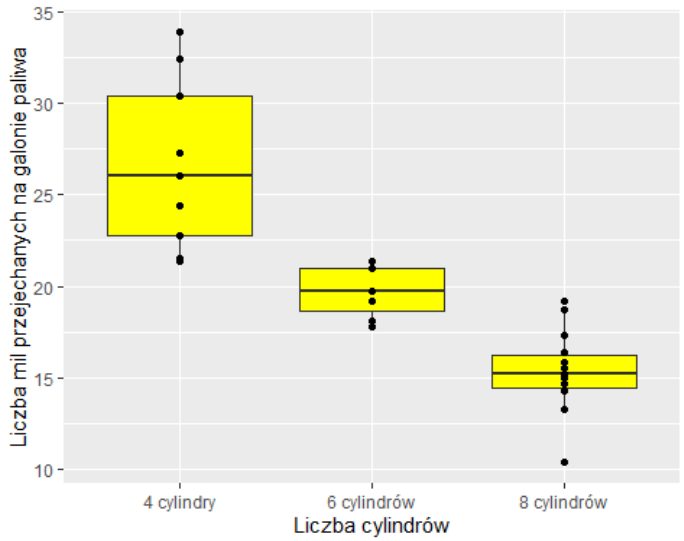
Na rysunku 5.18 przedstawiono wykres pudełkowy dla zmiennej *mpg* (zmienna ciągła) z wyróżnieniem kategorii ze względu na zmienną *cyl* (zmienna dyskretna). Do tego wykresu można dodać punkty reprezentujące poszczególne obserwacje. Uzyskuje się to poprzez dodanie warstwy geom_point, co przedstawia kolejny kod.



Rysunek 5.18. Wykres pudełkowy dla trzech wyróżnionych kategorii. Liczba mil przejechanych na galonie paliwa według liczby cylindrów w samochodzie

Źródło: opracowanie własne w programie R.

```
# Dodanie warstwy z obserwacjami
ggplot(mtcars,aes(cyl,mpg))+
  geom_boxplot(fill='yellow')+
  geom_point()+
  labs(x='Liczba cylindrów',y='Liczba mil przejechanych na
galonie paliwa')
```

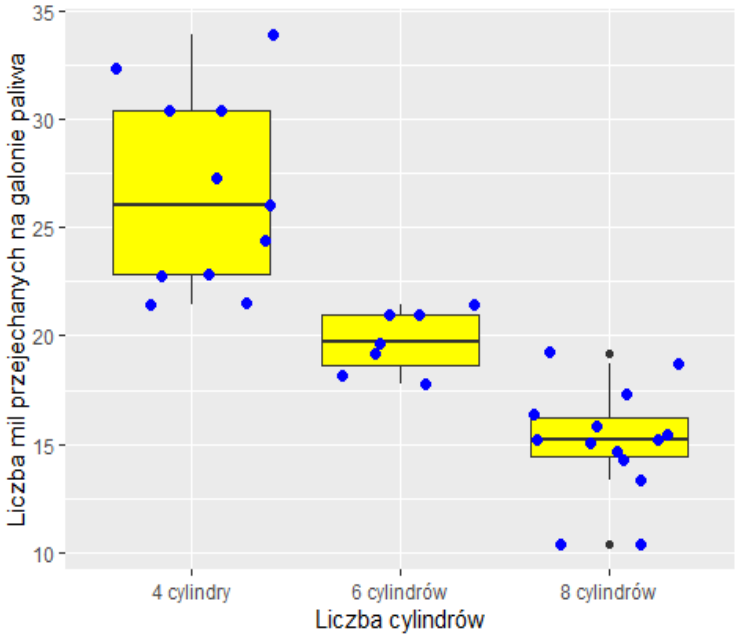


Rysunek 5.19. Wykres pudełkowy dla trzech wyróżnionych kategorii z zaznaczonymi punktami. Liczba mil przejechanych na galonie paliwa według liczby cylindrów w samochodzie

Źródło: opracowanie własne w programie R.

Na rysunku 5.19 przedstawiono na tle wykresu pudełkowego wszystkie obserwacje. Zastosowane rozwiązanie na jednym wykresie wprowadza dwie warstwy reprezentacji geometrycznej (geom_boxplot i geom_point). Można zauważyć, że w niektórych przypadkach punkty nakładają się na siebie i nie wszystkie obserwacje są dobrze widoczne. Można temu zaradzić, dodając zamiast warstwy geom_point warstwę geom_jitter. Dodanie takiej warstwy spowoduje rozrzucenie punktów w poziomie, dzięki czemu punkty nie będą już się nakładały na siebie (por. rysunek 5.20).

```
# "Rozrzucenie" obserwacji
ggplot(mtcars,aes(cyl,mpg))+
  geom_boxplot(fill='yellow')+
  geom_jitter(color='blue',size=2)+
  labs(x='Liczba cylindrów',y='Liczba mil przejechanych na
galonie paliwa')
```

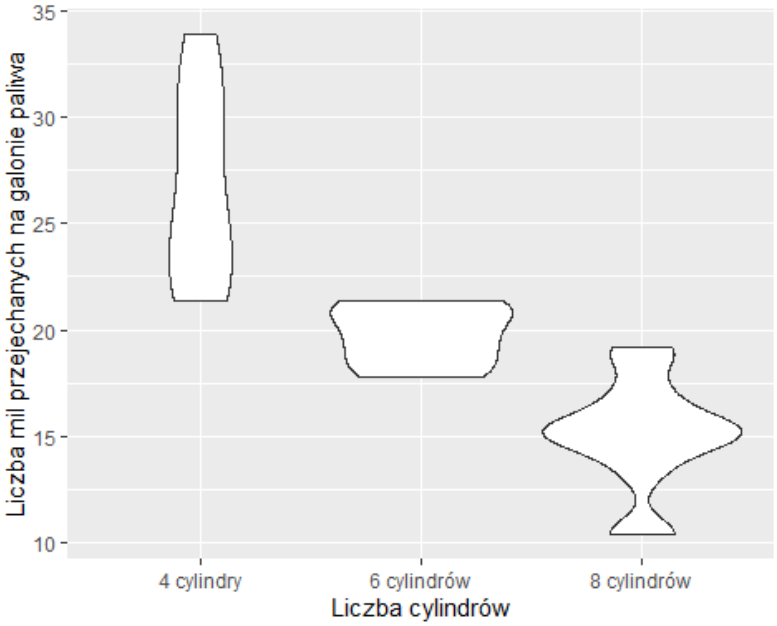


Rysunek 5.20. Wykres pudełkowy dla trzech wyróżnionych kategorii z rozrzuconymi punktami. Liczba mil przejechanych na galonie paliwa według liczby cylindrów w samochodzie

Źródło: opracowanie własne w programie R.

Pewną odmianą wykresów pudełkowych są wykresy wiolinowe. O ile wykres pudełkowy wyróżnia wszystkie kwartyle, o tyle wykres wiolinowy przekazuje kompleksowy obraz rozkładu badanej zmiennej (kształt oszacowania funkcji gęstości). Dla konstrukcji takich wykresów zamiast geom_boxplot należy użyć warstwy geom_violin. Takie rozwiązanie przedstawia następujący kod.

```
# Konstrukcja wykresu wiolinowego
ggplot(mtcars,aes(factor(cyl),mpg))+
  geom_violin()+
  labs(x='Liczba cylindrów',y='Liczba mil przejechanych na
galonie paliwa')
```

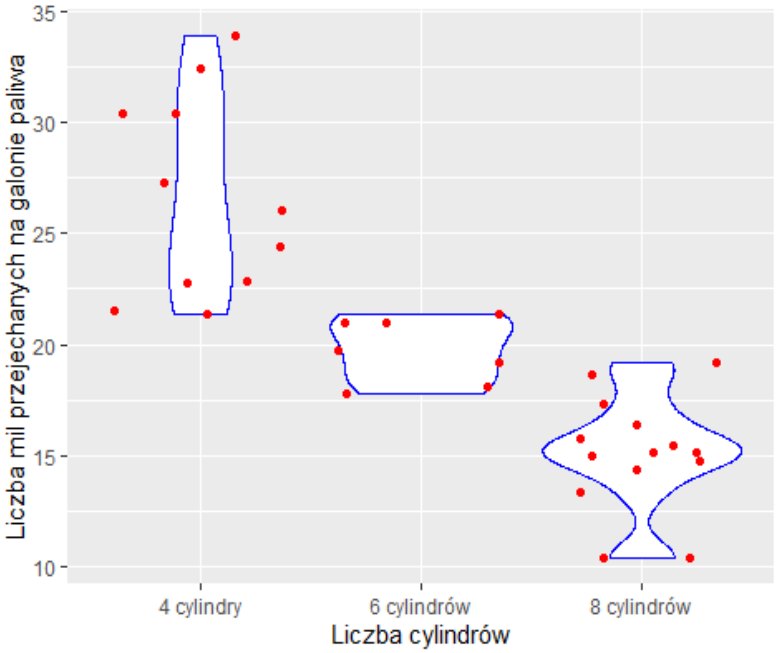


Rysunek 5.21. Wykres wiolinowy dla trzech wyróżnionych kategorii. Liczba mil przejechanych na galonie paliwa według liczby cylindrów w samochodzie

Źródło: opracowanie własne w programie R.

Rysunek 5.21 przedstawia wykres wiolinowy. Podobnie jak dla wykresu pudełkowego możliwe jest dodanie punktów na wykresie poprzez zastosowanie warstwy geom_point lub warstwy geom_jitter. Ten drugi wariant zastosowano poniżej.

```
# Wykres wiolinowy z „rozrzuconymi” punktami - jitter
ggplot(mtcars,aes(factor(cyl),mpg))+
  geom_violin(color='blue')+
  geom_jitter(color='red')+
  labs(x='Liczba cylindrów',y='Liczba mil przejechanych na
galonie paliwa')
```



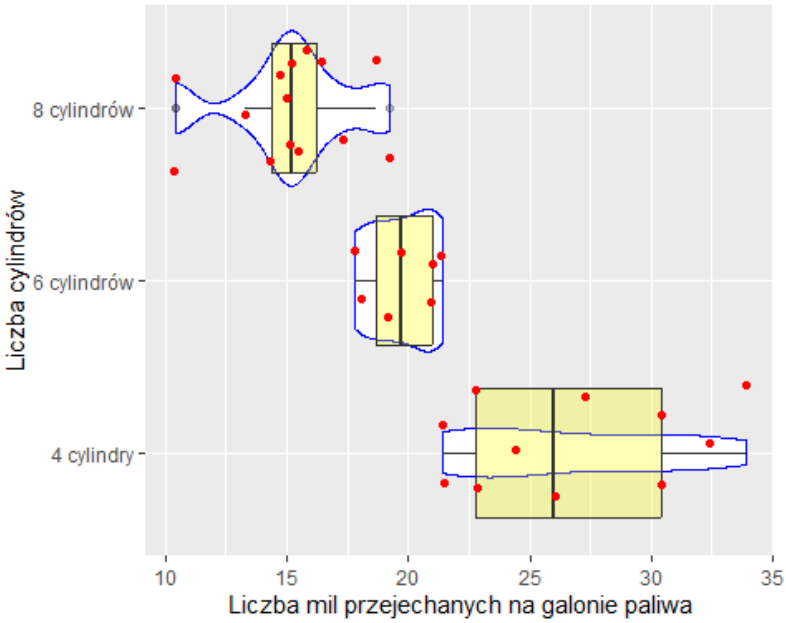
Rysunek 5.22. Wykres wiolinowy dla trzech wyróżnionych kategorii z zaznaczonymi rozrzuconymi punktami. Liczba mil przejechanych na galonie paliwa według liczby cylindrów w samochodzie

Źródło: opracowanie własne w programie R.

Rysunek 5.22 przedstawia wykres wiolinowy dla zmiennej ciągłej *mpg* z wyróżnionymi trzema kategoriami (*cyl*) oraz z rozrzuconymi punktami obserwacji (warstwa geom_jitter). Niekiedy dla właściwego zobrazowania badanego rozkładu zmiennej może być korzystne przedstawienie dwóch lub większej liczby warstw reprezentacji geometrycznej. Taką możliwość prezentuje poniższy kod. Na wykresie umieszczono trzy warstwy reprezentacji geometrycznej (geom_boxplot, geom_violin i geom_jitter).

```
# Wykres wiolinowy z „rozrzuconymi” punktami - jitter
ggplot(mtcars,aes(factor(cyl),mpg))+
  geom_violin(color='blue')+
    geom_boxplot(fill='yellow', alpha=0.3)+
  geom_jitter(color='red')+
  coord_flip()+
  labs(x='Liczba cylindrów',y='Liczba mil przejechanych na
galonie paliwa')
```

Wykres z trzema reprezentacjami geometrycznymi: wiolinową, wykresu pudełkowego oraz rozrzuconych punktów, jest przedstawiony na rysunku 5.23. Dodatkowo zastosowano orientację poziomą.



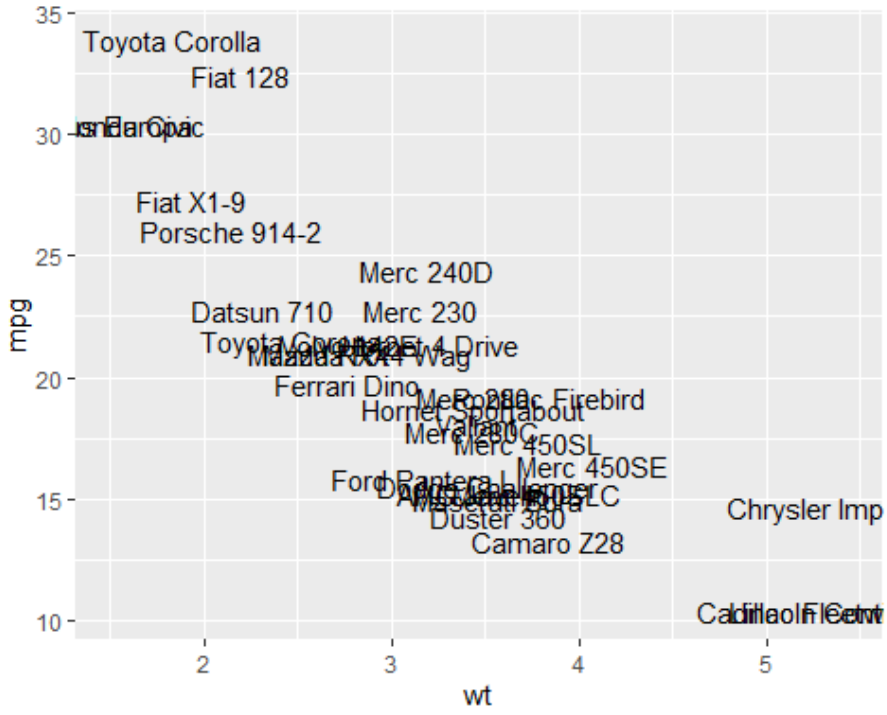
Rysunek 5.23. Wykres wiolinowy i pudełkowy z punktami rozrzuconymi. Liczba mil przejechanych na galonie paliwa według liczby cylindrów w samochodzie

Źródło: opracowanie własne w programie R.

5.3.5. Etykiety tekstowe w obszarze wykresu

W poprzednich konstrukcjach wykresów (pudełkowego i wiolinowego) do podstawowej prezentacji graficznej dodawano punkty. Podobnie do takich prezentacji: zamiast punktów można dodać etykiety tekstowe odpowiadające poszczególnym obiektom. Uzyskuje się to poprzez wprowadzenie do wykresu warstwy geom_text(). Taką operację realizuje następujący kod.

```
# Wprowadzenie etykiet tekstowych
ggplot(mtcars,aes(wt,mpg,label=row.names(mtcars)))+
  geom_text()
```

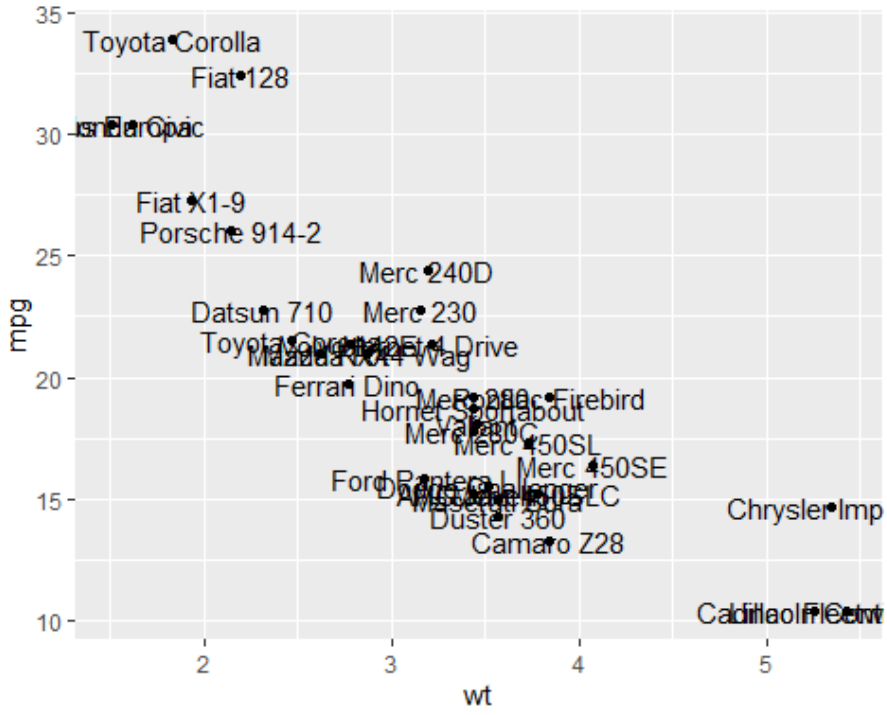


Rysunek 5.24. Wykres rozrzutu z etykietami tekstowymi zamiast punktów. Waga samochodu i liczba mil przejechanych na galonie paliwa

Źródło: opracowanie własne w programie R.

Rysunek 5.24 przedstawia wykres rozrzutu, na którym zamiast punktów zamieszczono etykiety obiektów. Oczywiście można pozostawić warstwę geom_point, aby na wykresie znalazły się jednocześnie i punkty, i etykiety obiektów. Kod w takim przypadku ujmuje dwie reprezentacje geometryczne i może być zapisany następująco.

```
# Etykiety tekstowe i punkty
ggplot(mtcars,aes(wt,mpg,label=row.names(mtcars)))+
  geom_text()+
  geom_point()
```

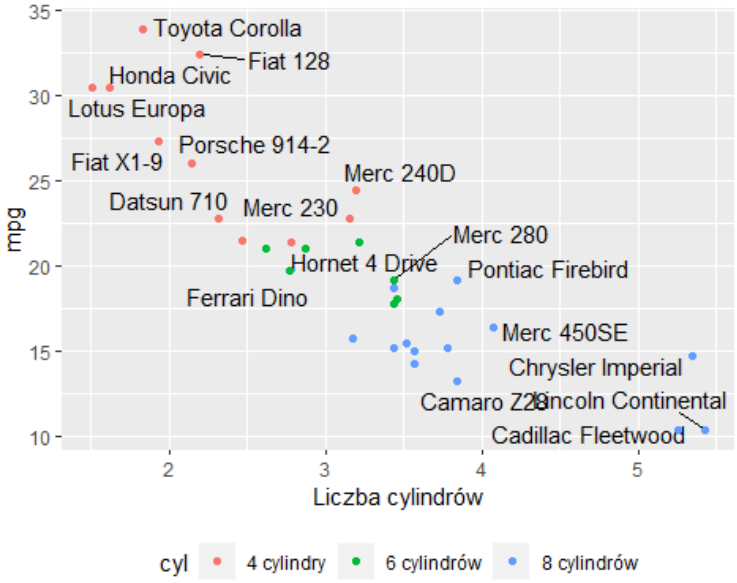


Rysunek 5.25. Wykres rozrzutu z punktami i etykietami tekstowymi. Waga samochodu i liczba mil przejechanych na galonie paliwa

Źródło: opracowanie własne w programie R.

Na rysunkach 5.24 i 5.25 zamieszczono etykiety tekstowe obiektów. Na obu wykresach w niektórych miejscach etykiety znacznie na siebie nachodzą, przez co nie są czytelne. Dobre rozwiązanie przynosi zastosowanie zamiast warstwy geom_text warstwy geom_text_repel. Taka reprezentacja geometryczna nie jest standardowym rozwiązaniem zawartym w **ggplot2** (por. tabela 5.2). Dla zapewnienia możliwości skorzystania z tej reprezentacji geometrycznej należy załadować bibliotekę **ggrepel** (por. rysunek 5.26).

```
# Wprowadzenie etykiet tekstowych i punktów
library(ggrepel)
p=ggplot(mtcars,aes(wt,mpg,label=row.names(mtcars)))+
  geom_point(aes(color=cyl))+
  labs(x='Waga samochodu')+
  theme(legend.position='bottom')
p+geom_text_repel()
```



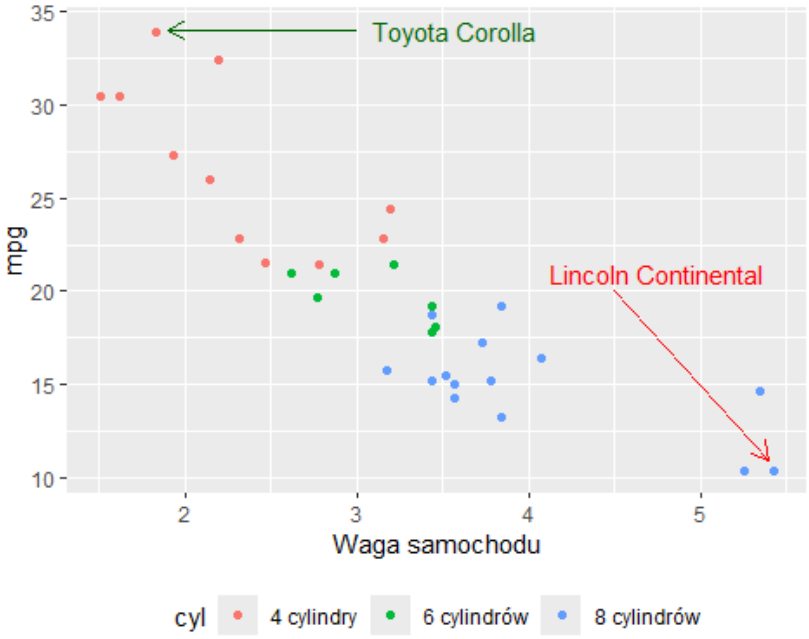
Rysunek 5.26. Wykres rozrzutu z punktami i etykietami z zastosowaniem warstwy geom_text_repel. Waga samochodu i liczba mil przejechanych na galonie paliwa

Źródło: opracowanie własne w programie R.

Na rysunku 5.26 widoczny jest efekt zastosowania jednocześnie reprezentacji geometrycznych geom_point oraz geom_text_repel. Ze względu na liczbę punktów (i etykiet tekstowych) nie wszystkie etykiety zostały zamieszczone, ale wykres stał się znacznie bardziej czytelny niż na rysunku 5.25.

Etykiety tekstowe można dodawać także w innej formie. Jedną z takich możliwości jest wykorzystanie funkcji *annotate*. Przykład dodania opisów oraz strzałek wskazujących na dwa wyróżnione punkty na wykresie przedstawia poniższy kod, a jego rezultat obrazuje wykres na rysunku 5.27.

```
# Dodanie do wykresu strzałek z opisami
p + annotate("segment", x = 3, y = 34, xend = 1.9, yend =
34,arrow = arrow(length = unit(0.3, "cm")), color = "darkgreen")
+
  annotate("text", x = 3, y = 34, label = "Toyota Corolla",
           hjust = -0.1, color = "darkgreen")+
  annotate("segment", x = 4.5, y = 20, xend = 5.4, yend =
11,arrow = arrow(length = unit(0.3, "cm")), color = "red") +
  annotate("text", x = 4, y = 21, label = "Lincoln Continen-
tal",hjust = -0.1, color = "red")
```



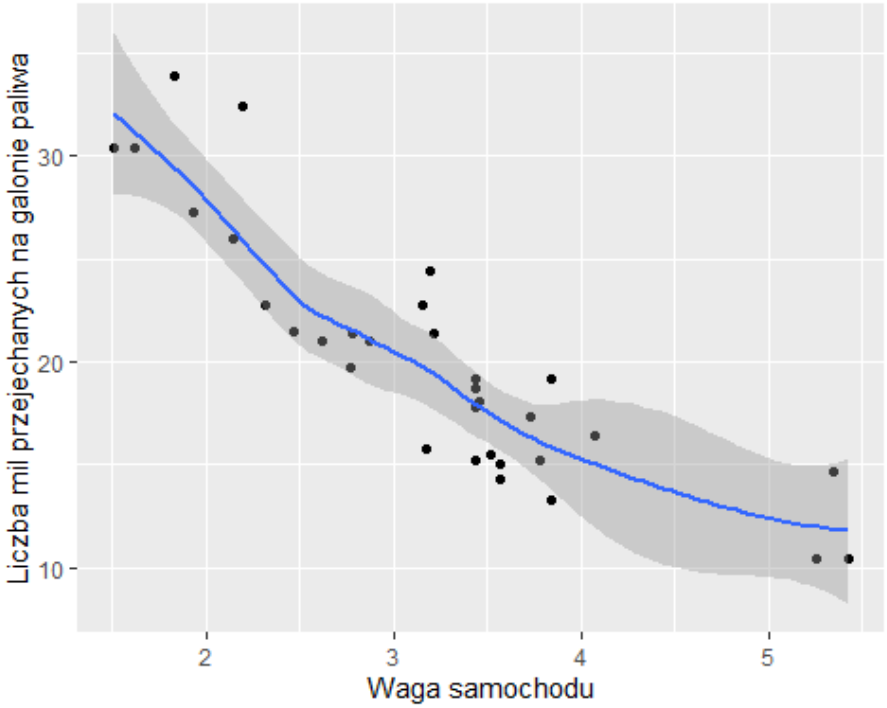
Rysunek 5.27. Wykres rozrzutu z dodanymi strzałkami i opisami obserwacji. Waga samochodu i liczba mil przejechanych na galonie paliwa

Źródło: opracowanie własne w programie R.

5.3.6. Graficzne przedstawienie funkcji regresji

Na wykresie rozrzutu możliwe jest dodanie różnych postaci linii regresji. Można to uzyskać poprzez wprowadzenie warstwy geom_smooth. Realizuje to poniższy kod, a wynik działania przedstawiono na rysunku 5.27.

```
# Konstrukcja wykresu funkcji regresji
ggplot(mtcars,aes(wt,mpg))+
  geom_point()+
  geom_smooth()+
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa')
```

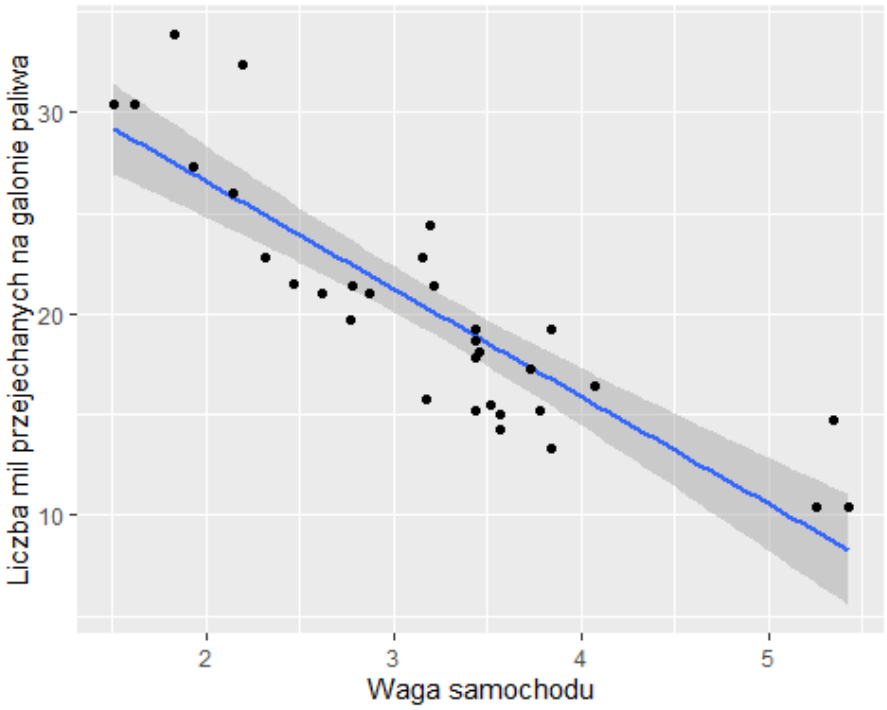


Rysunek 5.28. Wykres rozrzutu z punktami linią regresji. Waga samochodu i liczba mil przejechanych na galonie paliwa

Źródło: opracowanie własne w programie R.

Na rysunku 5.28 zaprezentowano linię regresji. Często celem może być otrzymanie wykresu funkcji regresji określonej postaci, na przykład liniowej funkcji regresji. W takim przypadku do komendy należy wprowadzić parametr określający postać funkcji regresji, na przykład jako model liniowy (method = 'lm') jak w poniższym kodzie.

```
# Regresja liniowa
ggplot(mtcars,aes(wt,mpg))+
  geom_smooth(method='lm')+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa')
```

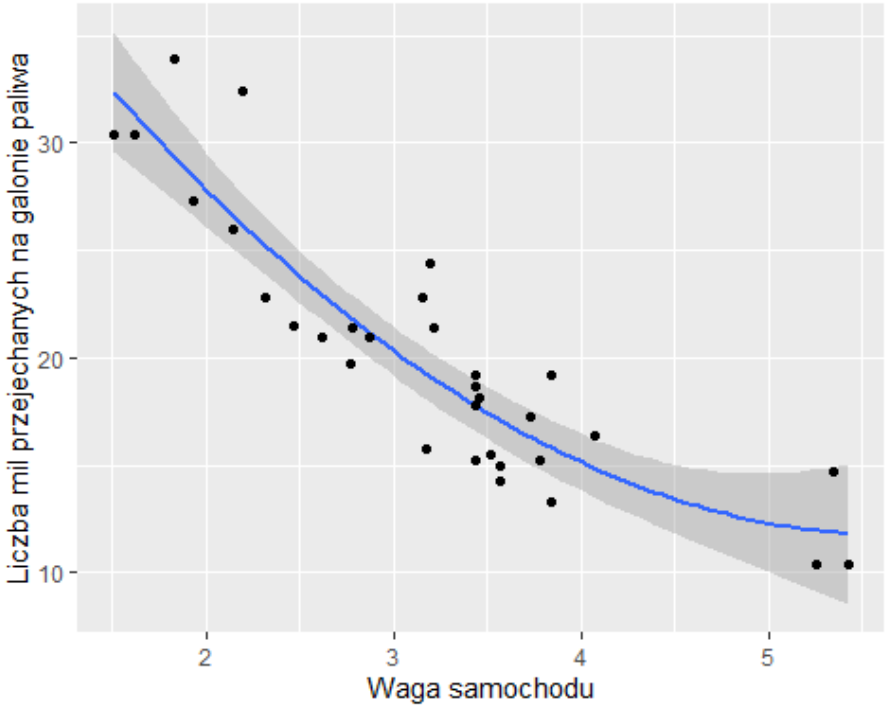


Rysunek 5.29. Wykres rozrzutu z liniową funkcją regresji. Waga samochodu i liczba mil przejechanych na galonie paliwa – regresja liniowa

Źródło: opracowanie własne w programie R.

Na rysunku 5.29 na wykresie rozrzutu zobrazowano dodatkowo liniową funkcję regresji. Dodanie parametru 'formula' w warstwie geom_smooth pozwala umieścić na wykresie inną postać funkcji regresji, na przykład wielomianową stopnia drugiego jak na rysunku 5.30.

```
# Regresja kwadratowa
ggplot(mtcars,aes(wt,mpg))+
  geom_smooth(method='lm',formula=y~poly(x,2))+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa')
```

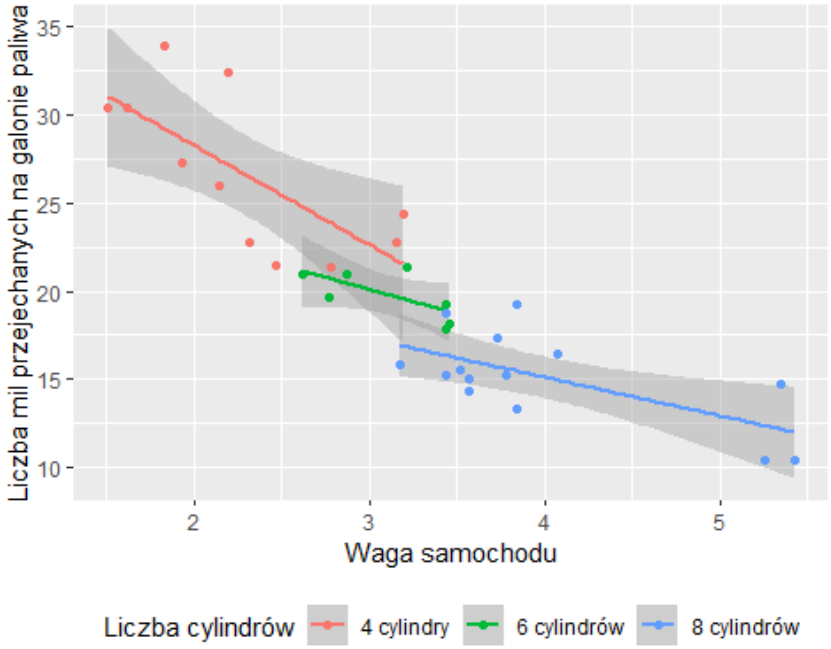


Rysunek 5.30. Wykres rozrzutu z wielomianową funkcją regresji stopnia drugiego. Waga samochodu i liczba mil przejechanych na galonie paliwa – funkcja regresji drugiego stopnia

Źródło: opracowanie własne w programie R.

Podobnie jak we wcześniej prezentowanych przykładach można poszczególne linie regresji wyróżnić kolorami w zależności od pewnej zmiennej dyskretnej, jak na przykład od liczby cylindrów (zmienna *cyl*) w samochodzie. Realizuje to poniższy kod, którego wynik zamieszczono na rysunku 5.31.

```
# Funkcja regresji - wyróżnienie kilku grup
ggplot(mtcars,aes(wt,mpg,color=cyl))+
  geom_smooth(method='lm')+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa',color='Liczba cylindrów')+
  theme(legend.position='bottom')
```



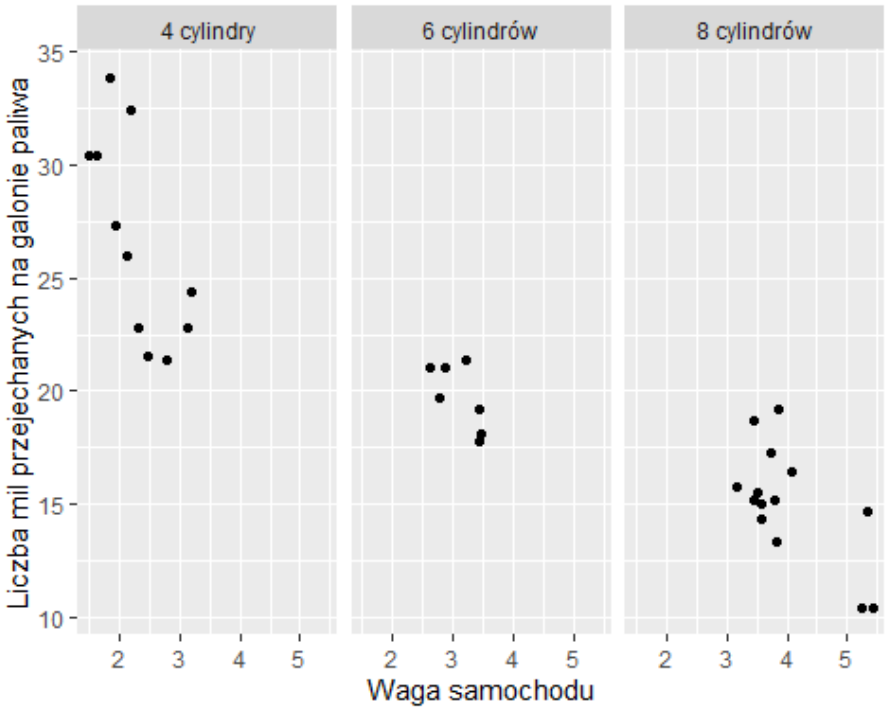
Rysunek 5.31. Wykres rozrzutu z liniowymi funkcjami regresji dla samochodów o zadanej liczbie cylindrów. Waga samochodu i liczba mil przejechanych na galonie paliwa – funkcje regresji dla ustalonej liczby cylindrów

Źródło: opracowanie własne w programie R.

5.3.7. Wykresy w panelach (facet) – idea oraz przykłady zastosowań

W analizie danych często dąży się do przeprowadzenia różnych porównań. Graficznie takie porównania można zrealizować z wykorzystaniem warstw paneli (facet_wrap oraz facet_grid). Dodanie do wykresu rozrzutu warstwy facet_wrap pozwala na uzyskanie kilku okien z wykresami rozrzutu dla wyróżnionych wariantów czynnika, na przykład dla różnych wariantów liczby cylindrów w samochodzie (por. rysunek 5.32).

```
# Panele w konstrukcji wykresów
ggplot(mtcars,aes(wt,mpg))+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa',color='Liczba cylindrów')+
  facet_wrap(~cyl)
```

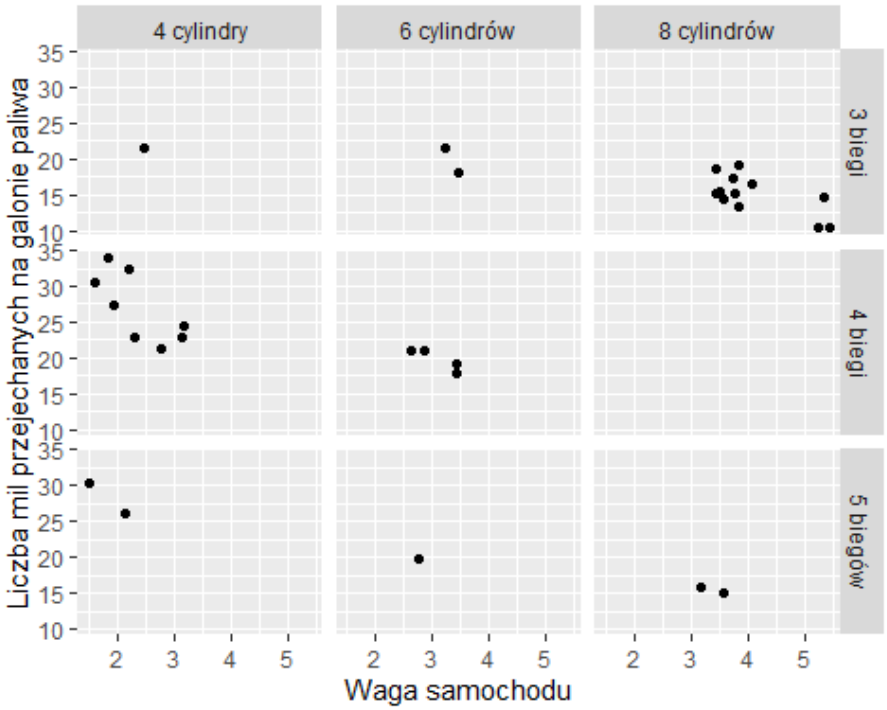


Rysunek 5.32. Wykres rozrzutu w układzie paneli. Waga samochodu i liczba mil przejechanych na jednym galonie paliwa według liczby cylindrów

Źródło: opracowanie własne w programie R.

Możliwe jest także uzyskanie prostokątnej siatki wykresów uwzględniającej warianty dwóch zmiennych dyskretnych (facet_grid). Taki wykres zaprezentowano na rysunku 5.33, gdzie wyróżniono dwie zmienne jakościowe: liczbę cylindrów (*cyl*) oraz liczbę biegów (*gear*). Wykres ten uzyskano na podstawie następującego kodu.

```
# Siatka dwuwymiarowa paneli
ggplot(mtcars,aes(wt,mpg))+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa',color='Liczba cylindrów')+
  facet_grid(gear~cyl)
```

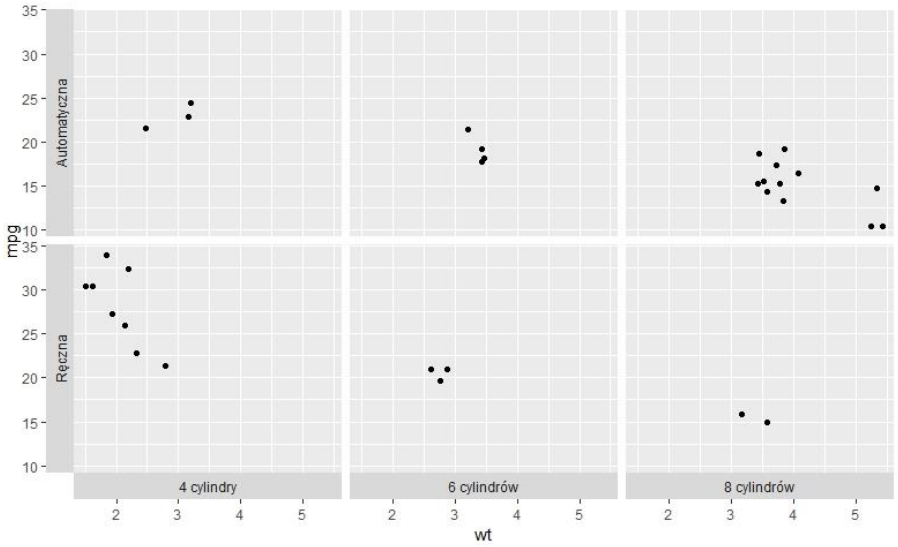


Rysunek 5.33. Wykres rozrzutu w siatce paneli. Waga samochodu i liczba mil przejechanych na jednym galonie paliwa według liczby cylindrów i biegów

Źródło: opracowanie własne w programie R.

Na rysunku 5.33 przedstawiono dwuwymiarową siatkę wykresów rozrzutu. Na każdym polu wykresu rozrzutu przedstawiono wagę samochodu (*wt*) oraz liczbę przejechanych kilometrów na galonie paliwa (*mpg*). Utworzone panele odpowiadają wariantom zmiennych *cyl* oraz *gear*. Nagłówki dotyczące wariantów tych zmiennych znajdują się u góry (zmienna *cyl*) oraz po prawej stronie (zmienna *gear*). Użytkownik ma jednak dużą swobodę modyfikacji odpowiednich ustawień i zmiany położenia takich opisów. Realizuje to poniższy kod.

```
# Zmiana położenia etykiet paneli
data <- transform(mtcars,
  am = factor(am, levels = 0:1, c("Automatyczna", "Ręczna")),
  cyl = factor(cyl, labels=c('4 cylindry', '6 cylindrów', '8
cylindrów')))
ggplot(data, aes(wt, mpg)) +
  geom_point() +
  facet_grid(am ~ cyl, switch = "both")
```

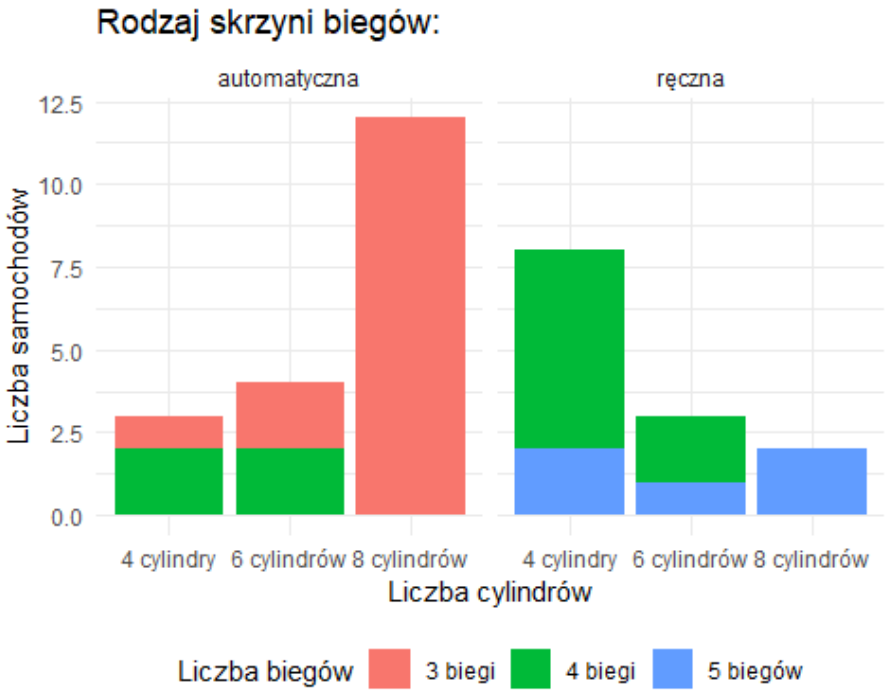


Rysunek 5.34. Wykres rozrzutu w siatce paneli przy zmianie położenia etykiet. Waga samochodu i liczba mil przejechanych na jednym galonie paliwa według liczby cylindrów i biegów

Źródło: opracowanie własne w programie R.

Na rysunku 5.34 etykiety wariantów zmiennych dyskretnych *cyl* i *gear* zostały umieszczone odpowiednio na dole oraz po lewej stronie wykresu. Uprzednio w rozważaniach przedstawiono sposoby konstrukcji wykresów słupkowych (por. rysunki 5.13-5.16). Oczywiście do takich wykresów również może być zastosowane rozmieszczenie w panelach. Przykład rozmieszczenia wykresów słupkowych nakładanych w dwóch panelach związanych z wariantami zmiennej *am* (rodzaj skrzyni biegów) realizuje następujący kod.

```
# Konstrukcja wykresu stosowanego słupków dla liczby cylindrów
dla dwóch grup skrzyń biegów
ggplot(mtcars, aes(x = factor(cyl), fill = factor(gear))) +
  geom_bar(position = "stack") +
  labs(title = "Rodzaj skrzyni biegów:",x = "Liczba cylindrów",
y = "Liczba samochodów") +
  scale_fill_discrete(name = "Liczba biegów") +
  facet_wrap(~am)+
  theme_minimal()+
  theme(legend.position='bottom')
```

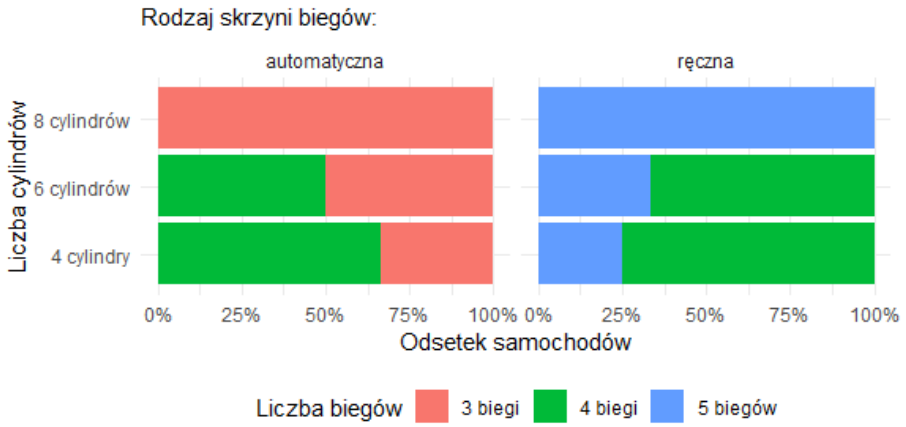


Rysunek 5.35. Wykres słupkowy nakładany z dwoma panelami. Liczba samochodów względem liczby cylindrów i rodzaju skrzyni biegów

Źródło: opracowanie własne w programie R.

Na rysunku 5.35 przedstawiono w układzie panelowym wykres słupkowy nakładany (por. rysunek 5.15). Ten wykres ujmuje trzy zmienne dyskretne: *cyl*, *gear* oraz *am*. Natomiast poniższy kod realizuje podobną prezentację graficzną, ale dla wykresu słupkowego struktury (por. rysunek 5.16).

```
# Tworzenie wykresu stosowanego słupków dla liczby cylindrów dla
dwóch grup skrzyń biegów
ggplot(mtcars, aes(x = factor(cyl), fill = factor(gear))) +
  geom_bar(position = "fill") +
  labs(subtitle = "Rodzaj skrzyni biegów:",x = "Liczba cylin-
drów", y = "Odsetek samochodów") +
  scale_fill_discrete(name = "Liczba biegów") +
  facet_wrap(~am)+
  coord_flip()+
  theme_minimal()+
  theme(legend.position='bottom')+
     scale_y_continuous(labels = scales::percent_format())
```



Rysunek 5.36. Wykres słupkowy struktury z dwoma panelami. Struktura liczby samochodów względem liczby cylindrów i rodzaju skrzyni biegów

Źródło: opracowanie własne w programie R.

Na rysunku 5.36 przedstawiono wykres słupkowy struktury na podstawie trzech zmiennych dyskretnych: *cyl*, *gear* oraz *am*.

5.3.8. Kompozycje wykresów – pakiety patchwork i ggpubr

Niekiedy dla przekazania w prezentacji graficznej szczególnych informacji wskazane jest umieszczenie dwóch lub większej liczby wykresów w obszarze pola graficznego (rysunku). Można to bardzo wygodnie zrealizować z wykorzystaniem takich pakietów jak **patchwork** oraz **ggpubr**. W tym celu w pierwszej kolejności należy utworzyć obiekty zawierające poszczególne rysunki, a następnie posługując się możliwościami, które udostępnia pakiet **patchwork**, odpowiednio rozmieścić wykonane wcześniej wykresy. Przykłady takich rozwiązań (kody oraz rysunki) zostały zaprezentowane w tym punkcie.

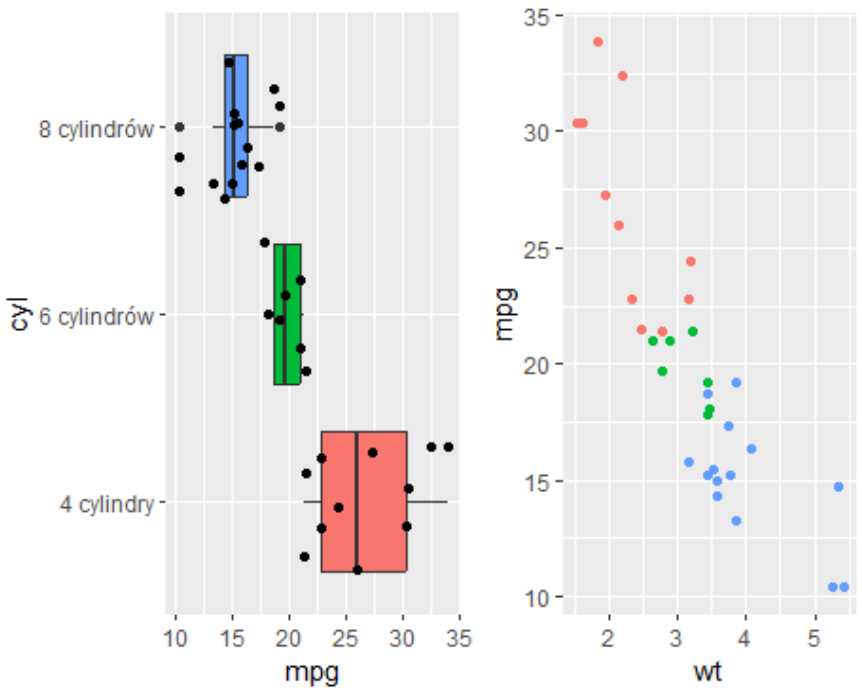
W pierwszej kolejności zostaną utworzone cztery wykresy i zapamiętane jako obiekty **r1**, **r2**, **r3** i **r4**. Załadowanie odpowiedniej biblioteki oraz przygotowanie tych rysunków przedstawia następujący kod.

```
# Załadowanie biblioteki i utworzenie wykresów r1-r4
library(patchwork)
library(ggpubr)
r1=ggplot(data=mtcars,aes(x=wt,y=mpg,color=cyl))+geom_point()+th
eme(legend.position='none')
r2=ggplot(mtcars,aes(mpg,fill=cyl))+
```

```
  geom_histogram()+
  theme(legend.position='none')
r3=ggplot(mtcars,aes(cyl,mpg,fill=cyl))+
  geom_boxplot()+
  geom_jitter()+
  theme(legend.position='none')+
  coord_flip()
r4=ggplot(mtcars,aes(wt,mpg,color=cyl))+
  geom_smooth(method='lm')+
  theme(legend.position='none')+
  geom_point()
```

Zastosowanie pakietu **patchwork** pozwala w bardzo prosty sposób umieścić wcześniej przygotowane rysunki obok siebie. Dla rozmieszczenia dwóch wykresów obok siebie należy wykonać poniższy kod.

```
# Umieszczenie dwóch wykresów obok siebie
r3+r1
```

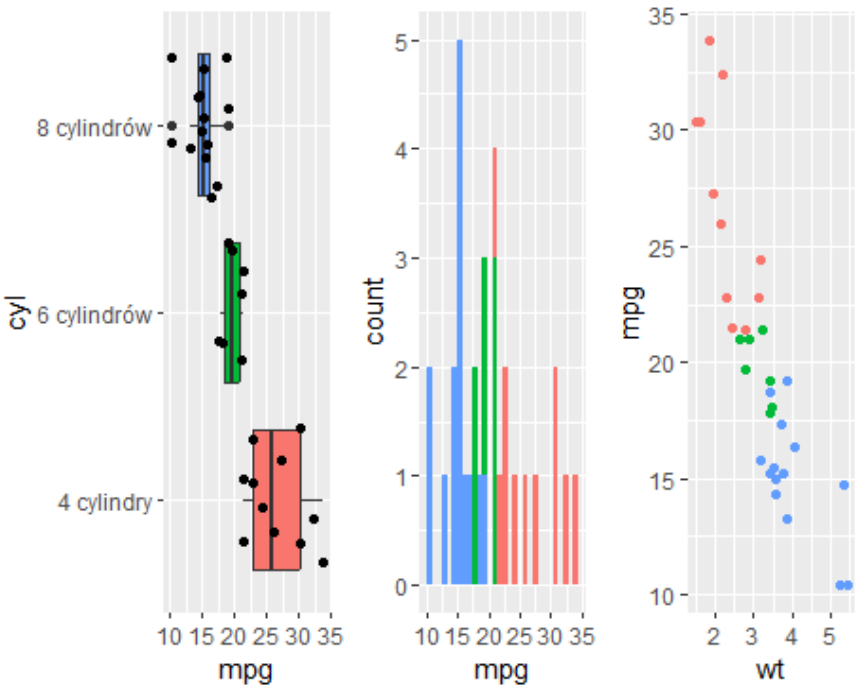


Rysunek 5.37. Umieszczenie dwóch wykresów obok siebie

Źródło: opracowanie własne w programie R.

Na rysunku 5.37 dwa wykresy wykonane w **ggplot2** zostały umieszczone obok siebie. Podobnie na rysunku 5.38 ujęto trzy takie wykresy w jednym wierszu. Jednak próba umieszczenia czterech wykresów jeden obok drugiego nie prowadzi do oczekiwanego rezultatu. Wykresy zostają umieszczone w układzie dwa w górnej części i dwa w dolnej części (por. rysunek 5.39).

```
# Umieszczenie trzech wykresów obok siebie
r3+r2+r1
```



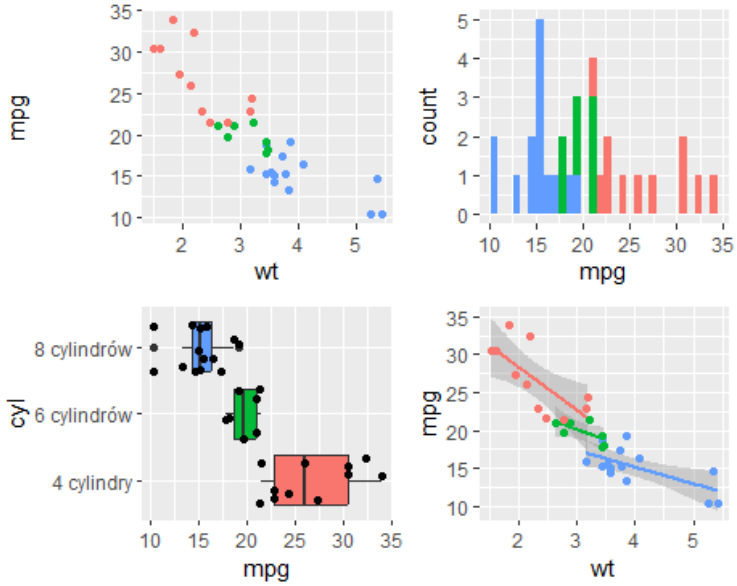
Rysunek 5.38. Umieszczenie trzech wykresów obok siebie

Źródło: opracowanie własne w programie R.

```
# Rozmieszczenie czterech wykresów
r1+r2+r3+r4
```

Dla wymuszenia odpowiedniego rozmieszczenia wykresów w jednej linii wystarczy pomiędzy obiektami zamiast znaku „+” wprowadzić znak „|”. Rezultat w tym przypadku będzie zgodny z wcześniejszymi oczekiwaniami. Odpowiedni kod jest przedstawiony poniżej, a wynik jego realizacji został zaprezentowany na rysunku 5.40.

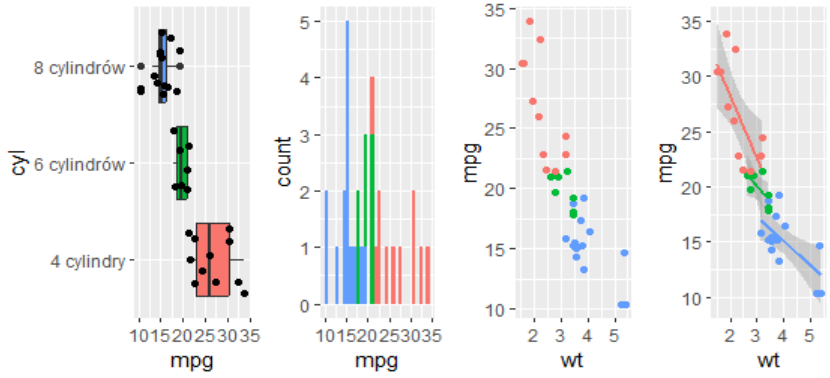
```
# Umieszczenie czterech wykresów obok siebie
r3|r2|r1|r4
```



Rysunek 5.39. Umieszczenie czterech wykresów

Źródło: opracowanie własne w programie R.

Znaki „+” oraz „|” wprowadzone pomiędzy symbolami obiektów powodują umieszczanie kolejnych wykresów jeden za drugim, z tą różnicą, że zastosowanie znaku „+” prowadzi do rozmieszczenia większej liczby obiektów w kolejnych liniach, natomiast wstawienie znaku „|” wymusza ustawianie kolejnych wykresów w tej samej linii.



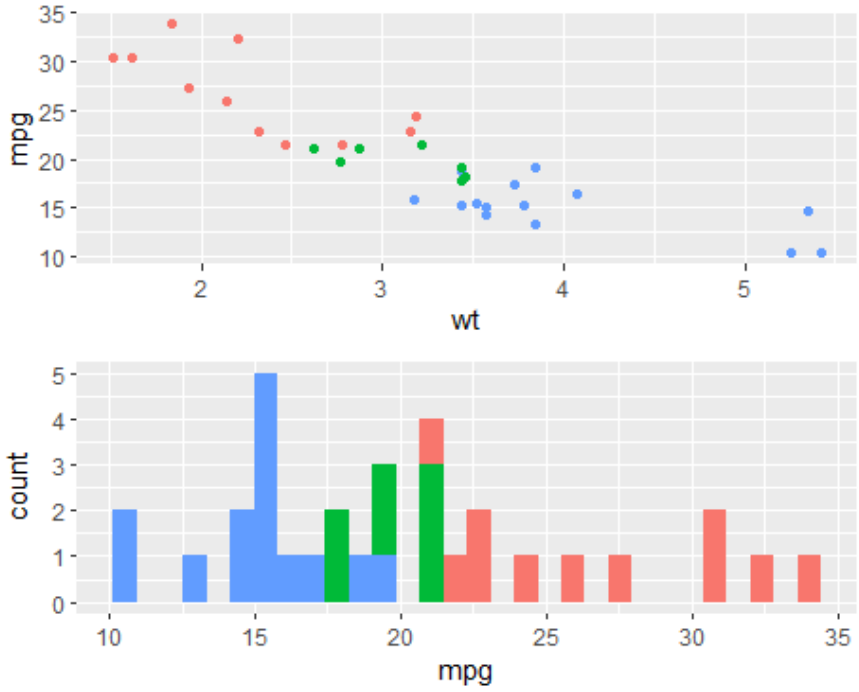
Rysunek 5.40. Umieszczenie czterech wykresów obok siebie

Źródło: opracowanie własne w programie R.

Pakiet **patchwork** umożliwia użytkownikowi rozmieszczanie obiektów w dość swobodny sposób. Dla przykładu: aby umieścić dwa wykresy jeden ponad drugim, należy pomiędzy obiektami wprowadzić znak „/”.

```
# Umieszczenie dwóch wykresów w układzie pionowym
r1/r2
```

W efekcie powyższej komendy uzyskuje się układ wykresów jak na rysunku 5.41.

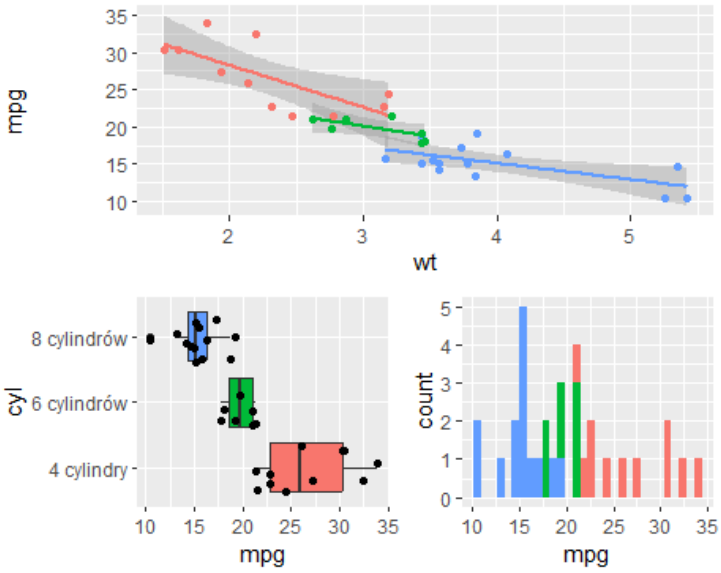


Rysunek 5.41. Umieszczenie dwóch wykresów w układzie pionowym

Źródło: opracowanie własne w programie R.

Rozmieszczenie dwóch wykresów w pierwszej linii oraz jednego wykresu w drugiej linii uzyskuje się w następujący sposób (por. rysunek 5.42).

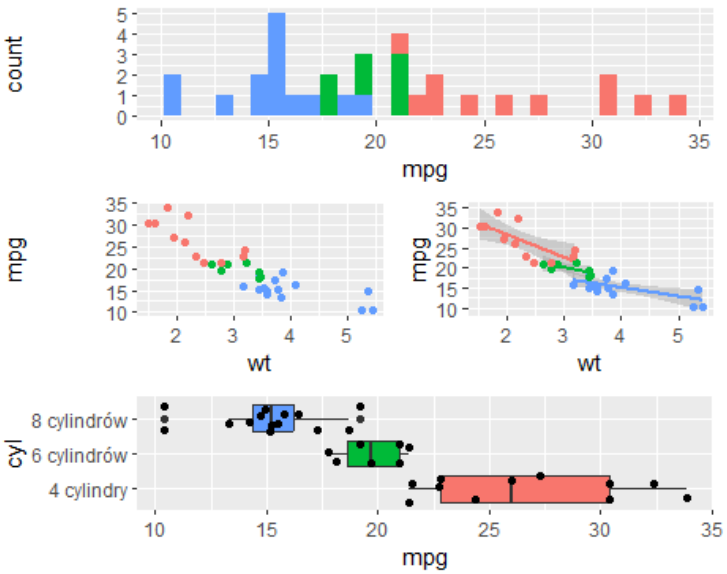
```
# Umieszczenie jednego wykresu u góry i dwóch na dole
r4/(r3+r2)
```



Rysunek 5.42. Umieszczenie trzech wykresów w układzie jeden u góry i dwa na dole

Źródło: opracowanie własne w programie R.

```
# Umieszczenie czterech wykresów w trzech liniach
r2/(r1+r4)/r3
```

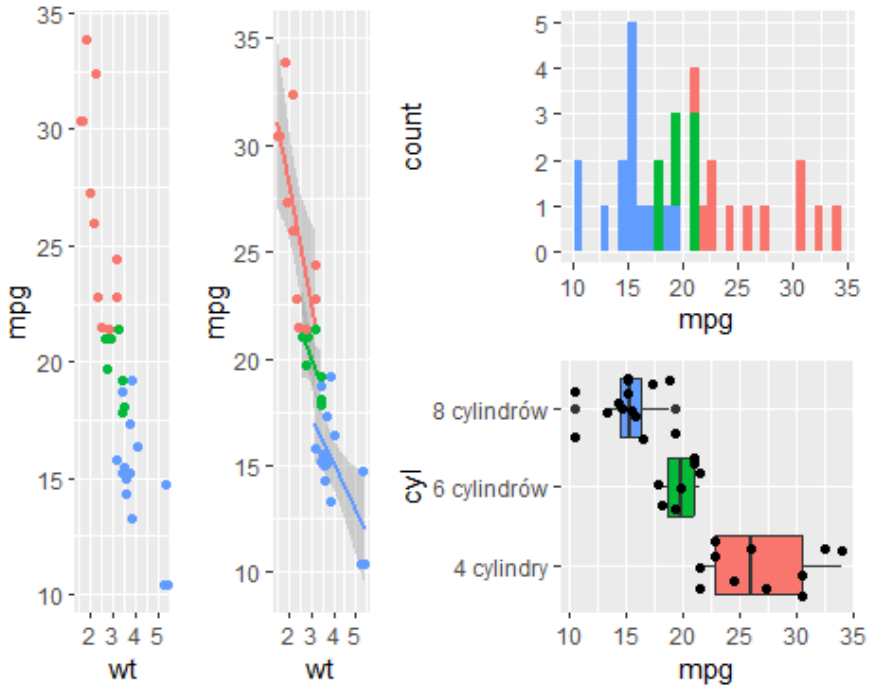


Rysunek 5.43. Umieszczenie czterech wykresów w układzie 1 / 2 / 1

Źródło: opracowanie własne w programie R.

```
# Umieszczenie czterech wykresów w trzech kolumnach
(r1+r4)|r2/r3
```

Na rysunkach 5.43 i 5.44 wskazano możliwości konstrukcji różnych układów położenia wcześniej przygotowanych wykresów.

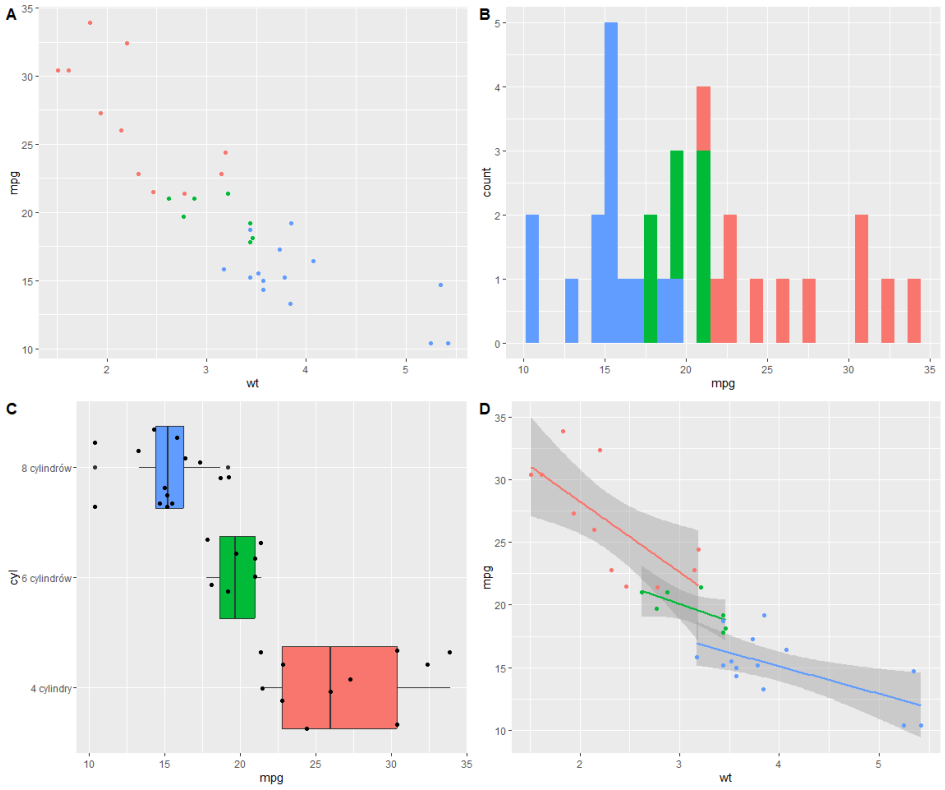


Rysunek 5.44. Umieszczenie czterech wykresów w układzie 1 / 1 / 2

Źródło: opracowanie własne w programie R.

Podobne rezultaty jak powyżej można uzyskać z wykorzystaniem funkcji *ggarrange* dostępnej w pakiecie **ggpubr**. Przykłady takich rozwiązań przedstawiają dwa poniższe kody, a ich rezultaty zamieszczono na rysunkach 5.45 i 5.46.

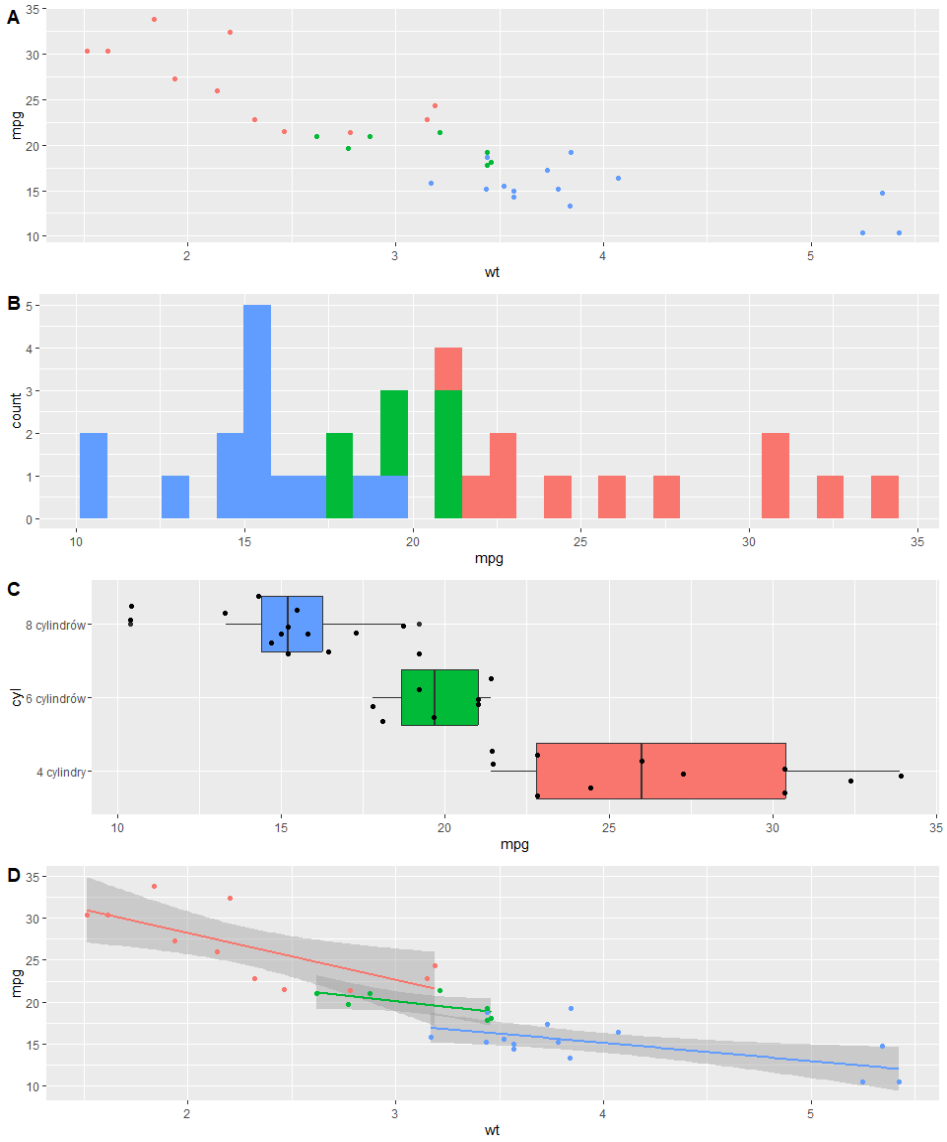
```
# Aranżacja wykresów
p <- ggarrange(r1, r2, r3, r4,
                          labels = c("A", "B", "C", "D"),
                          ncol = 2, nrow = 2)
print(p)
```



Rysunek 5.45. Aranżacja czterech wykresów w układzie 2 x 2 – pakiet ggpubr

Źródło: opracowanie własne w programie R.

```
# Aranżacja wykresów
p <- ggarrange(r1, r2, r3, r4,
                          labels = c("A", "B", "C", "D"),
                          ncol = 1, nrow = 4)
print(p)
```



Rysunek 5.46. Aranżacja czterech wykresów w układzie 4 x 1 – pakiet ggpubr

Źródło: opracowanie własne w programie R.

W pakiecie **ggplot2** dostępne są motywy (theme), które pozwalają nadać określony wygląd wcześniej przygotowanym wykresom. Motywy to predefiniowane zestawy estetyk i parametrów, które umożliwiają jednolite formatowanie wykresów. Można w ten sposób szybko zmieniać wygląd wszystkich elementów na wykresie (tło, osie, etykiety, tytuły i inne). Pakiet **ggplot2** dostarcza kilka predefiniowanych motywów, takich jak theme_gray (szary), theme_bw (biały)

i theme_minimal (minimalistyczny). Każdy z tych motywów ma swoje własne ustawienia estetyczne, co pozwala na uzyskanie różnorodnego wyglądu wykresów. Znacznie większe możliwości daje zainstalowanie specjalistycznych pakietów, jak na przykład pakietu **ggthemes**. Utworzenie obiektu **rys** prezentuje poniższy kod, a odpowiedni rezultat przedstawia rysunek 5.47.

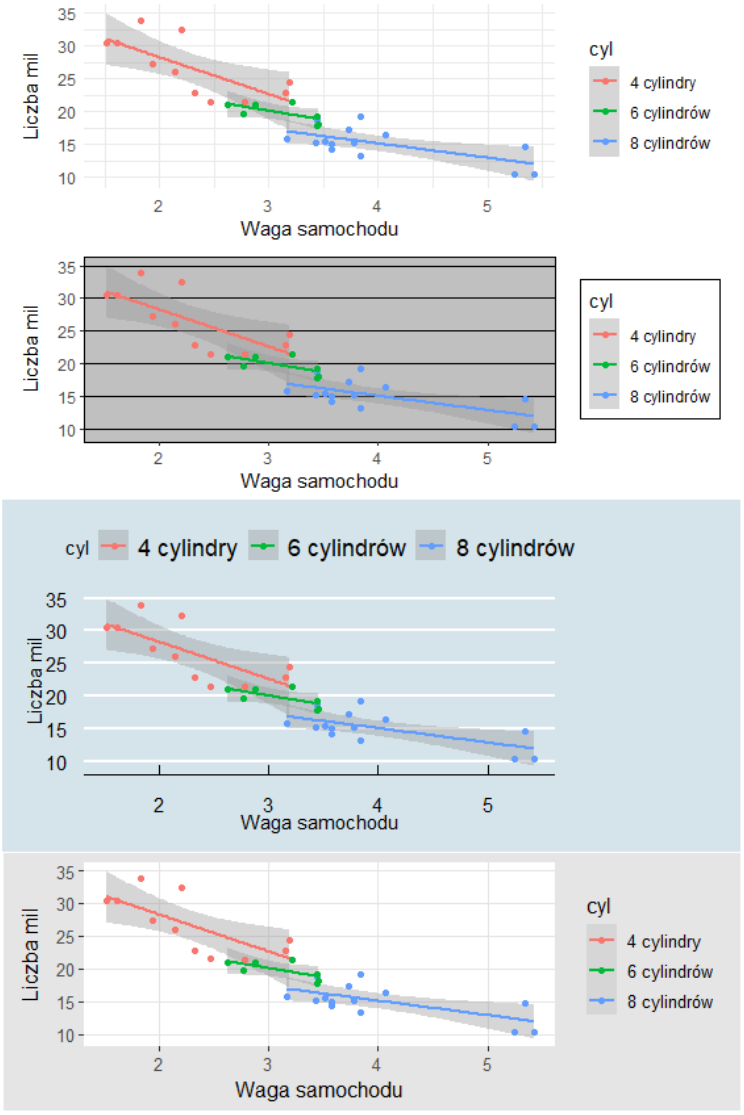
```
# Konstrukcja obiektu rys
rys=ggplot(mtcars,aes(wt,mpg,color=cyl))+
  geom_smooth(method='lm')+
  geom_point()+
  labs(x='Waga samochodu',y='Liczba mil przejechanych na ga-
lonie paliwa',color='Liczba cylindrów')
rys
```

Rysunek 5.47. Wykres – obiekt rys

Źródło: opracowanie własne w programie R.

Po utworzeniu obiektu **rys** możliwe jest przedstawienie go z wykorzystaniem różnych motywów. W tym celu niezbędne staje się załadowanie biblioteki **ggthemes**. Konstrukcję takich wykresów z wykorzystaniem tej biblioteki przedstawia poniższy kod, a rezultat widoczny jest na rysunku 5.48.

```
# Zastosowanie motywów do obiektu rys
library(ggthemes)
r1=rys+theme_minimal()
r2=rys+theme_excel()
r3=rys+theme_economist()
r4=rys+theme_igray()
r1/r2/r3/r4
```

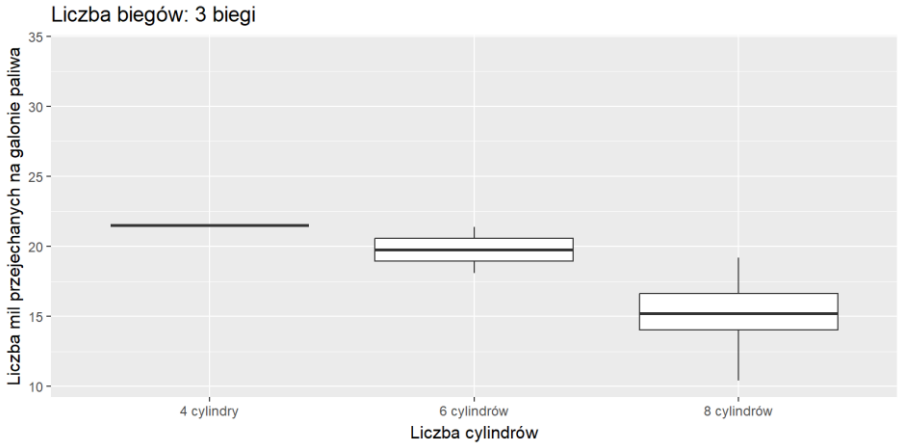


Rysunek 5.48. Wykres (obiekt rys) z zastosowaniem różnych motywów

Źródło: opracowanie własne w programie R.

Na wykresach można również wprowadzić animacje. Animacje takie nie będą oczywiście widoczne na finalnym wydruku, ale mogą zostać przedstawione na przykład na stronie internetowej lub w plikach umożliwiających zastosowanie takiej animacji, a więc również w dokumentach w formacie docx. Wykonanie prostej animacji dla danych ze zbioru **mtcars** przedstawia poniższy kod, a jego rezultat zamieszono na rysunku 5.49.

```
# Konstrukcja animacji
library(gganimate)
p=ggplot(mtcars, aes(factor(cyl), mpg)) +
  geom_boxplot() +
  transition_states(
    gear,
    transition_length = 1,
    state_length = 2) +
  labs(title = 'Liczba biegów: {closest_state}',x='Liczba cylin-
drów',y='Liczba mil przejechanych na galonie paliwa')
# Wykonanie animacji
animate(p, height = 4, width = 8, units = "in", res = 200)
```

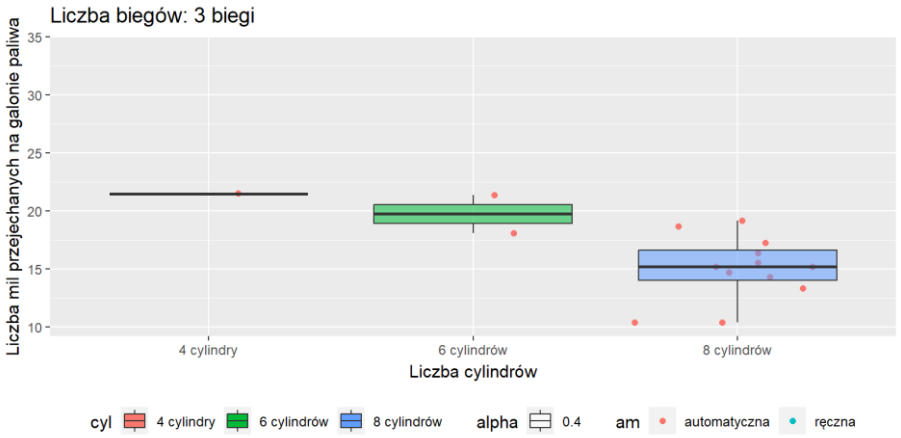


Rysunek 5.49. Wykres z animacją. Liczba mil przejechanych na galonie paliwa według liczby cylindrów

Źródło: opracowanie własne w programie R.

Do przedstawionej animacji poniższy kod wprowadza dodatkowe elementy, jak na przykład kolory dla wyróżnienia grup samochodów ze względu na liczbę cylindrów czy rozrzucone punkty z zaznaczeniem kolorem rodzaju skrzyni biegów (automatyczna lub ręczna). Realizację takiej prezentacji przedstawia niniejszy kod, a wynik został zobrazowany na rysunku 5.50.

```
# Konstrukcja animacji z dodatkowymi parametrami
p=ggplot(mtcars, aes(factor(cyl), mpg)) +
    geom_jitter(aes(color=am)) +
  geom_boxplot(aes(fill=cyl,alpha=0.4)) +
        theme(legend.position='bottom')+
  transition_states(
    gear,
    transition_length = 1,
    state_length = 2) +
  labs(title = 'Liczba biegów: {closest_state}',x='Liczba cylin-
drów',y='Liczba mil przejechanych na galonie paliwa')
# Wykonanie animacji
animate(p, height = 4, width = 8, units = "in", res = 200)
```



Rysunek 5.50. Wykres z animacją z dodatkiem kolorów. Liczba mil przejechanych na galonie paliwa według liczby cylindrów

Źródło: opracowanie własne w programie R.

5.3.9. Eksport wykresu do pliku

Pakiet **ggplot2** umożliwia nie tylko przygotowanie i wyświetlenie wysokiej klasy wizualizacji danych, ale również zapisanie grafiki w plikach o różnorodnych formatach. Do tego celu może być wykorzystana funkcja *ggsave*. Jako parametry wywołania tej funkcji podaje się nazwę pliku, wykres do zapisania, a także ewentualne dodatkowe parametry, jak na przykład szerokość (width) i wysokość (height) rysunku lub rozdzielczość (dpi).

```r
# Konstrukcja wykresu
p <- ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  labs(title = "Waga samochodu i liczba mil przejechanych na
galonie paliwa")

# Zapis wykresu w formacie PNG
ggsave("wykres.png", plot = p, width = 6, height = 4, dpi = 300)

# Zapis wykresu w formacie PDF
ggsave("wykres.pdf", plot = p, width = 6, height = 4)

# Zapis wykresu w formatach SVG, JPEG, TIFF i EPS
ggsave("wykres.svg", plot = p, width = 6, height = 4)
ggsave("wykres.jpg", plot = p, width = 6, height = 4, dpi = 300)
ggsave("myplot.tiff", plot = p)
ggsave("myplot.eps", plot = p)
```

6 Wybrane biblioteki rozszerzające możliwości pakietu ggplot2

Doskonałość graficzna jest prawie zawsze wielowymiarowa.

Edward R. Tufte*

W poprzednim rozdziale przedstawiono podstawowe możliwości graficzne pakietu **ggplot2**. W zaprezentowanych przykładach odwołano się również do pakietów **ggrepel**, **ggthemes**, **patchwork** oraz **gganimate**, które rozszerzają możliwości pakietu **ggplot2**. Takich pakietów jest jednak znacznie więcej i charakteryzują się one ogromną różnorodnością możliwych zastosowań. Ponad 120 rozszerzeń do pakietu **ggplot2** przedstawia serwis ggplot2Extensions (b.r.). Inne obszerne zestawienie takich rozszerzeń jest dostępne na stronie Awesome ggplot2 (Erik Gahner b.r.). W tym serwisie wyróżniono różne kategorie takich rozszerzeń, między innymi: Plot layers, Themes and aesthetics, Presentations, Interactive, Network, Spatial, Data and models. Szeroki wybór pakietów jest również przedstawiony na stronie R-charts (b.r.). Podobnie jak w poprzednim przypadku wyróżniono różne kategorie, w tym: Distribution, Correlation, Evolution, Spatial, Part of a Whole, Ranking, Flow oraz Miscelanous.

W tym rozdziale przedstawiono wybrane pakiety rozszerzające możliwości pakietu **ggplot2**. Ze względu na możliwości tej monografii zaprezentowano tylko kilka z wielu dostępnych pakietów rozszerzających.

* Tufte (1983, s. 51) – tłumaczenie własne.

6.1. Charakterystyka wybranych pakietów

W tabeli 6.1 przedstawiono wykaz wybranych pakietów rozszerzających możliwości pakietu **ggplot2** wykorzystanych w punktach 6.1-6.3 niniejszego rozdziału.

Tabela 6.1. Wybrane biblioteki rozszerzające możliwości pakietu ggplot2

Biblioteka	Opis
ggcorrplot	Wizualizacja macierzy korelacji za pomocą **ggplot2**
GGally	System graficzny oparty na Grammar of graphics
ggExtra	Zbiór funkcji i warstw rozszerzających dla **ggplot2**
ggside	Dodawanie graficznych informacji o jednym z paneli
ggridges	Wizualizacja zmian rozkładów w czasie lub przestrzeni
ggmosaic	Konstrukcja wykresów mozaikowych w **ggplot2**
ggmulti	Wizualizacja danych wielowymiarowych

Źródło: opracowanie własne na podstawie CRAN (b.r.).

Pakiety wykorzystane w tym rozdziale dotyczą głównie sposobów prezentacji zależności pomiędzy zmiennymi oraz wizualizacji danych wielowymiarowych, a w szczególności danych o charakterze jakościowym. W ostatnim punkcie odwołano się również do pakietów, które nie są rozszerzeniami **ggplot2**, ale mogą być pomocne w analizie danych wielowymiarowych.

6.2. Graficzna prezentacja zależności

Analiza zależności to proces badania relacji między różnymi zmiennymi w celu zrozumienia, czy i jakie związki występują między nimi. Jest to bardzo ważny krok w analizie danych, który pozwala odkryć wzorce, trendy i powiązania między zmiennymi. Analiza zależności ma ugruntowane miejsce w różnych dziedzinach i dyscyplinach nauki, takich jak ekonomia, finanse, nauki społeczne, nauki przyrodnicze, medycyna, marketing, psychologia i wiele innych. Cele i techniki wykorzystywane w takiej analizie w dużej mierze zależą od charakteru badanej dziedziny lub dyscypliny naukowej.

Przedstawienie możliwości omawianych pakietów rozszerzających należy rozpocząć od załadowania wymaganych pakietów (por. tabela 6.1). Realizowane jest to z wykorzystaniem następujących komend.

```
library(ggplot2)
library(patchwork)
library(ggcorrplot)
library(GGally)
library(ggExtra)
library(ggside)
library(ggridges)
library(ggmosaic)
library(ggmulti)
```

Do graficznego przedstawienia występujących zależności pomiędzy wybranymi zmiennymi przydatne są wartości współczynników korelacji liniowej Pearsona. Wyznaczenie macierzy współczynników korelacji dla wybranych zmiennych ilościowych (*mpg*, *hp*, *wt* oraz *qsec*) można zrealizować z pomocą funkcji *cor* następująco.

```
corr <- round(cor(mtcars[,c(1,4,6,7)]), 2)
corr
##        mpg    hp    wt  qsec
## mpg   1.00 -0.78 -0.87  0.42
## hp   -0.78  1.00  0.66 -0.71
## wt   -0.87  0.66  1.00 -0.17
## qsec  0.42 -0.71 -0.17  1.00
p.mat <- cor_pmat(mtcars[,c(1,4,6,7)])
p.mat
##               mpg           hp           wt         qsec
## mpg  0.000000e+00 1.787835e-07 1.293959e-10 1.708199e-02
## hp   1.787835e-07 0.000000e+00 4.145827e-05 5.766253e-06
## wt   1.293959e-10 4.145827e-05 0.000000e+00 3.388683e-01
## qsec 1.708199e-02 5.766253e-06 3.388683e-01 0.000000e+00
```

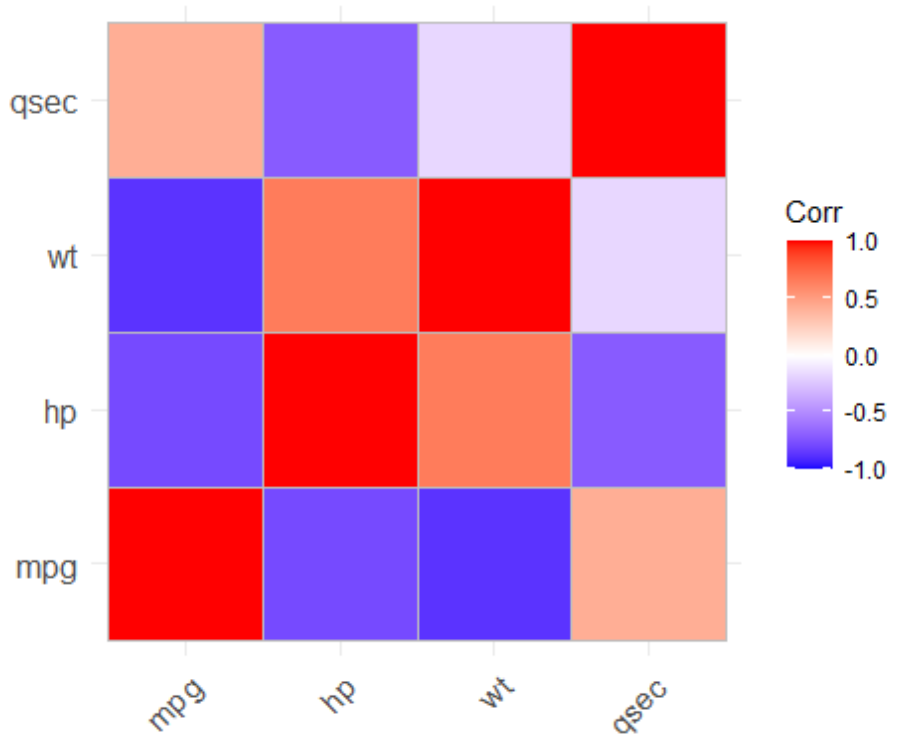
Dodatkowo powyższy kod pozwala na wyznaczenie poza wartościami współczynników korelacji liniowej Pearsona również *p*-wartości (obiekt **p.mat**), które także zostaną wykorzystane w dalszej części przy wizualizacji występujących zależności pomiędzy zmiennymi.

6.2.1. Pakiet ggcorrplot

Pakiet **ggcorrplot** umożliwia czytelną graficzną prezentację zależności pomiędzy zmiennymi na podstawie danej macierzy współczynników korelacji liniowej Pearsona. Jeżeli macierz współczynników korelacji jest umieszczona

w obiekcie **corr**, to wizualizację takich zależności uzyskuje się po wykonaniu następującej komendy.

```
# Ilustracja macierzy współczynników korelacji
ggcorrplot(corr)
```

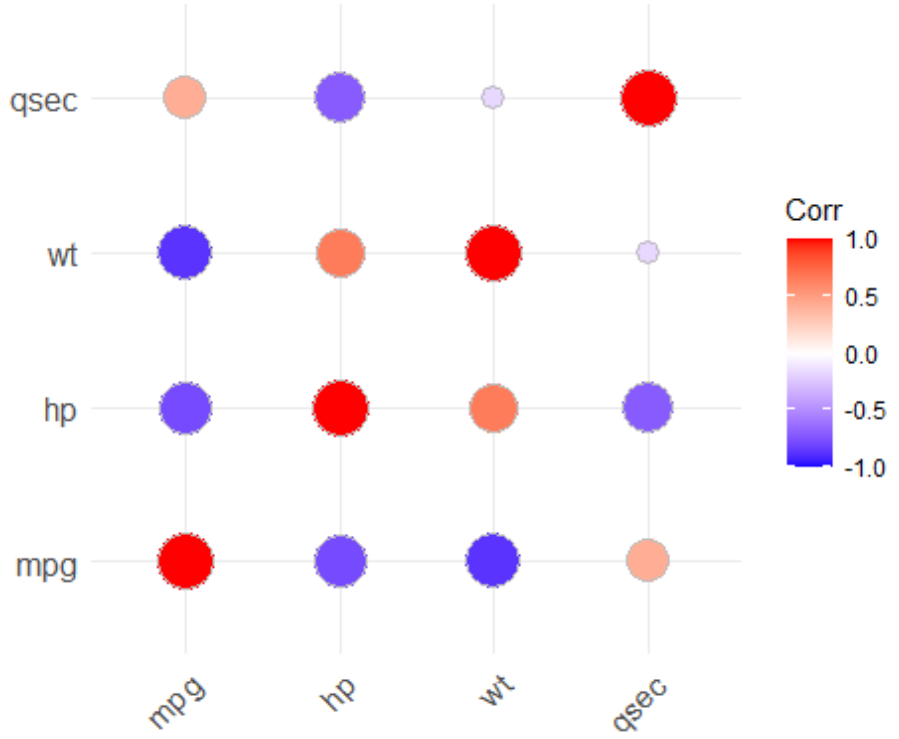


Rysunek 6.1. Siła zależności pomiędzy zmiennymi *mpg*, *hp*, *wt* i *qsec*

Źródło: opracowanie własne w programie R.

Na rysunku 6.1 przedstawiono graficzną prezentację siły zależności pomiędzy czterema analizowanymi zmiennymi. O sile zależności informuje kolor odpowiedniego pola zgodnie z legendą umieszczoną z prawej strony. Formę graficzną przekazu można zmieniać na różne sposoby. Tę samą informację co przedstawiona na rysunku 6.1, ale w nieco innej formie graficznej, przekazuje wykres na rysunku 6.2. Zamiast kolorowanych kwadratów wykorzystano kolorowane koła. O sile zależności informuje nie tylko kolor koła, ale także jego wielkość. Do takiego rezultatu prowadzi następująca komenda.

```
# Ilustracja macierzy współczynników korelacji - koła
ggcorrplot(corr, method = "circle")
```

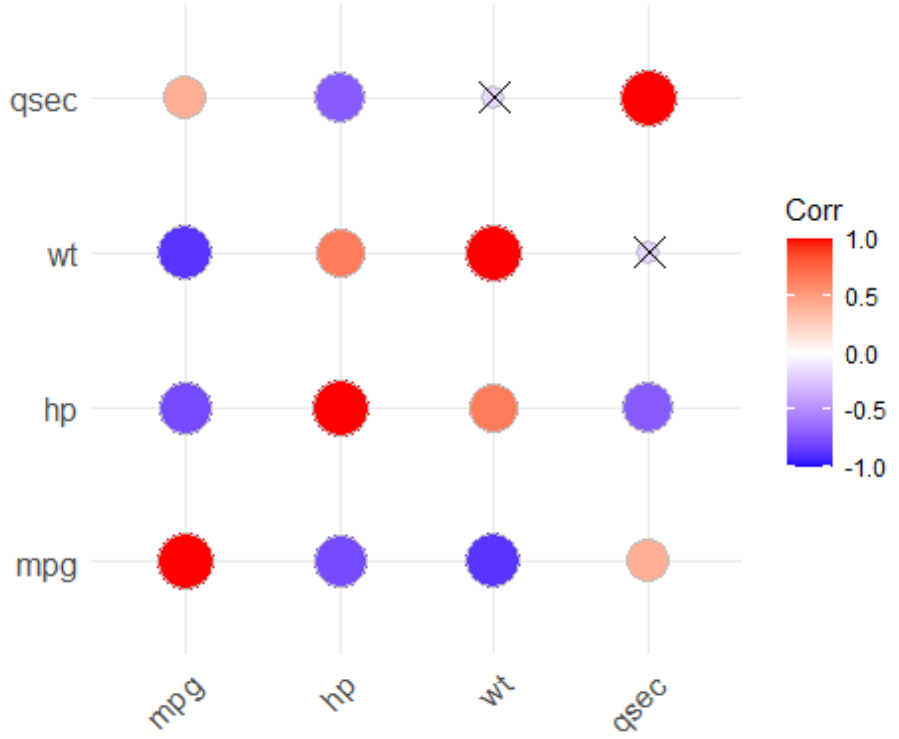


Rysunek 6.2. Siła zależności pomiędzy zmiennymi *mpg*, *hp*, *wt* i *qsec* (metoda circle)

Źródło: opracowanie własne w programie R.

W przypadku badania zależności pomiędzy zmiennymi ważne okazuje się nie tylko określenie siły tego związku, ale również zbadanie, czy zależność jest statystycznie istotna. Wyróżnienie na wykresie zależności, które nie są statystycznie istotne, można uzyskać poprzez wykonanie następującej komendy.

```
# Ilustracja macierzy współczynników korelacji - zaznaczenie
nieistotnych zależności
ggcorrplot(corr, p.mat = p.mat, method = "circle")
```

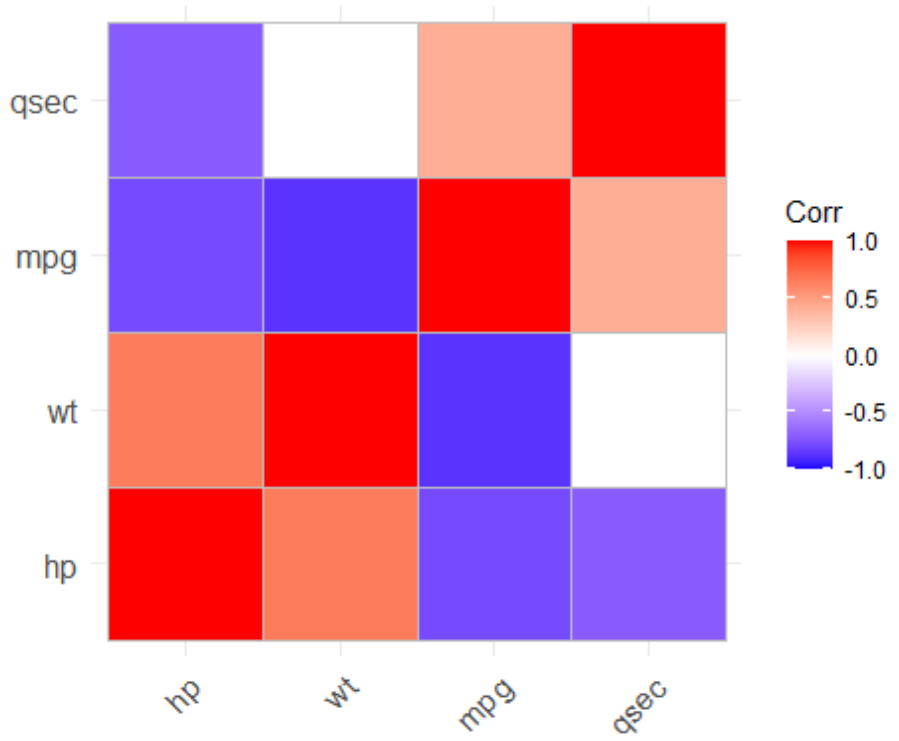


Rysunek 6.3. Siła zależności pomiędzy zmiennymi *mpg*, *hp*, *wt* i *qsec* z zaznaczeniem statystycznie nieistotnych zależności

Źródło: opracowanie własne w programie R.

Na rysunku 6.3 przedstawiono te same informacje co na dwóch poprzednich wykresach, ale dodatkowo wyróżnione zostały zależności (symbol „×"), które nie są statystycznie istotne. Nieco inną graficzną prezentację siły zależności z wyróżnieniem zależności statystycznie nieistotnych (parametr *insig*) realizuje poniższy kod, a odpowiedni wynik przedstawiono na rysunku 6.4.

```
# Ilustracja macierzy współczynników korelacji - nieistotne
zależności jako białe pola
ggcorrplot(corr, p.mat = p.mat,
  hc.order = TRUE, insig = "blank")
```

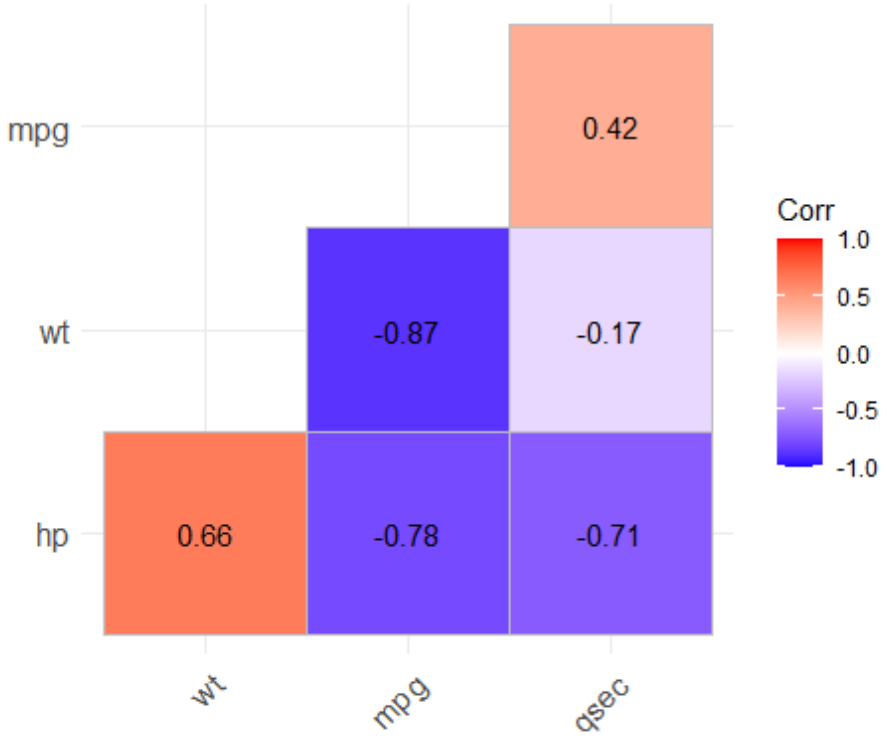


Rysunek 6.4. Siła zależności pomiędzy zmiennymi *mpg*, *hp*, *wt* i *qsec* z wykluczeniem zależności nieistotnych statystycznie (puste pola)

Źródło: opracowanie własne w programie R.

Graficzna prezentacja zazwyczaj nie przekazuje informacji o konkretnych wartościach mierników, w tym przypadku wartościach współczynników korelacji liniowej Pearsona. Poprzednie wykresy siłę zależności odwzorowywały jedynie za pomocą odpowiedniej tonacji kolorystycznej. Nie pozwala to użytkownikowi odczytać rzeczywistych wartości tych współczynników. Dla wskazania na wykresie dodatkowo wartości liczbowych wystarczy do poprzednich komend ustawić wartość parametru lab=TRUE, jak w kodzie poniżej. Odpowiedni rezultat przedstawiono na rysunku 6.5.

```
# Ilustracja macierzy współczynników korelacji z ich wartościami
ggcorrplot(corr, hc.order = TRUE,
           type = "lower", lab = TRUE)
```



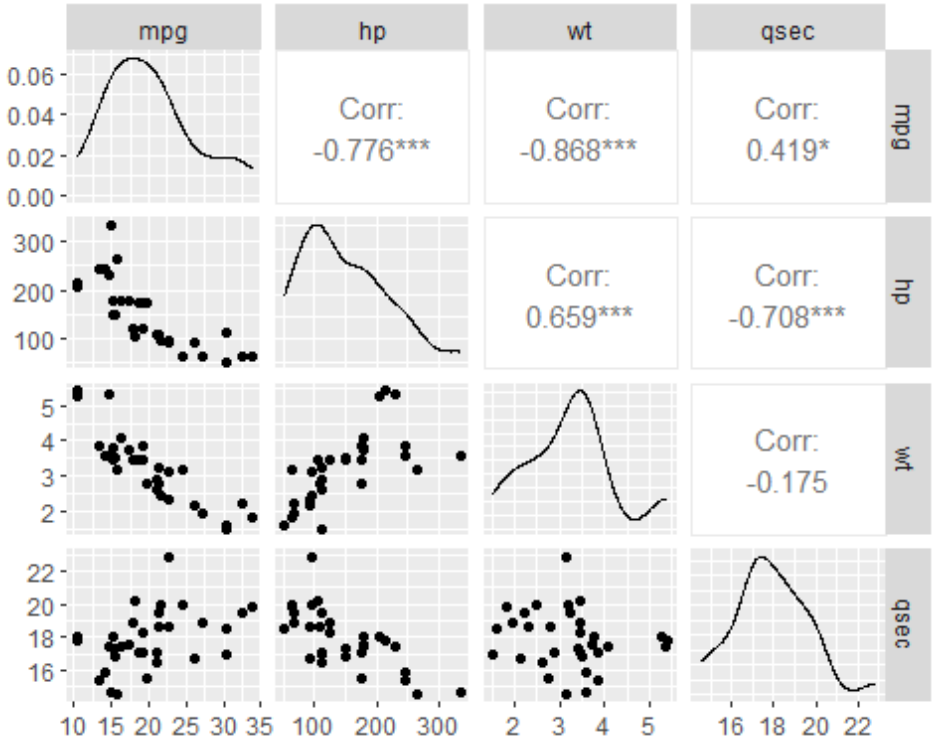
Rysunek 6.5. Siła zależności pomiędzy zmiennymi *mpg*, *hp*, *wt* i *qsec* z zaznaczeniem wartości współczynników korelacji liniowej Pearsona

Źródło: opracowanie własne w programie R.

6.2.2. Pakiet GGally

Przedstawiony w poprzednim punkcie pakiet **ggcorrplot** nie jest jedynym pozwalającym na wizualizację siły zależności pomiędzy badanymi zmiennymi. Kolejny taki pakiet to **GGally**. Funkcja *ggpairs* z tego pakietu umożliwia przedstawienie macierzowego wykresu rozrzutu. Dodatkowo ponad główną przekątną zostały zamieszczone wartości współczynników korelacji liniowej. Postać komendy przedstawiono poniżej, a jej rezultat zaprezentowano na rysunku 6.6.

```
# Macierzowy wykres rozrzutu
ggpairs(mtcars[,c(1,4,6,7)])
```



Rysunek 6.6. Macierzowy wykres rozrzutu z wartościami współczynników korelacji liniowej Pearsona

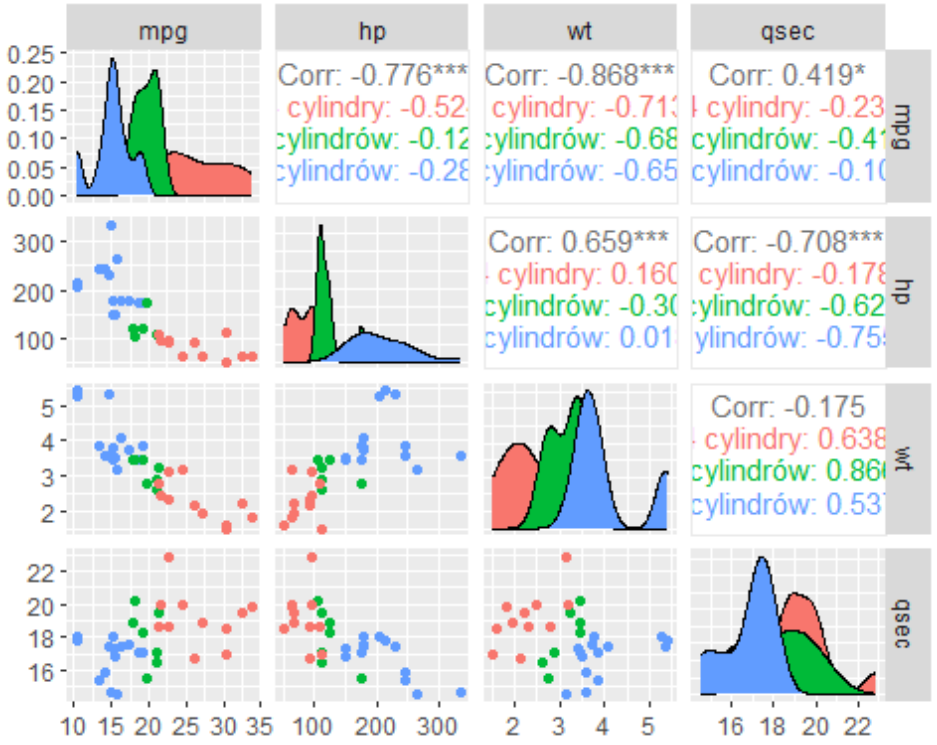
Źródło: opracowanie własne w programie R.

Identyczny efekt jak na rysunku 6.6 uzyskuje się, wskazując jako argumenty funkcji zbiór danych oraz kolumny (zmienne), na podstawie których należy wyznaczyć macierz współczynników korelacji. Postać komendy w tym przypadku jest następująca.

```
# Macierzowy wykres rozrzutu
ggpairs(mtcars,columns = c(1,4,6,7))
```

Niekiedy może być bardzo przydatne uwypuklenie na macierzowym wykresie rozrzutu różnych kategorii. Poniższy kod pozwala wyróżnić punkty odpowiadające obserwacjom ze względu na liczbę cylindrów w samochodzie.

```
# Macierzowy wykres rozrzutu z wyróżnieniem kategorii
ggpairs(mtcars, columns = c(1,4,6,7),aes(colour=factor(cyl)))
```

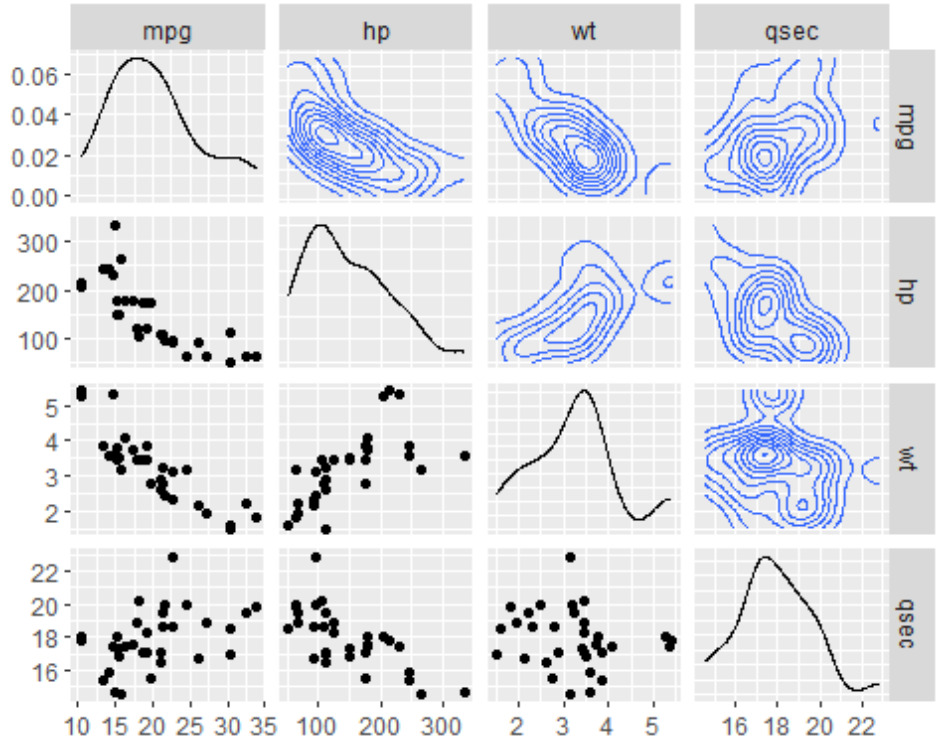


Rysunek 6.7. Macierzowy wykres rozrzutu z wartościami współczynników korelacji liniowej Pearsona i wyróżnieniem kategorii ze względu na liczbę cylindrów samochodu

Źródło: opracowanie własne w programie R.

Na rysunku 6.7 zaprezentowano macierzowy wykres rozrzutu dla tych samych danych co na rysunku 6.6. W tym przypadku dodatkowo kolorami zostały rozróżnione samochody ze względu na liczbę cylindrów. Zamiast wartości współczynników korelacji liniowej Pearsona ponad główną przekątną macierzy mogą zostać wykreślone funkcje gęstości, ewentualnie wykresy pudełkowe. Odpowiednie kody zostały przedstawione poniżej, a rezultaty ujęto na rysunkach 6.8 i 6.9.

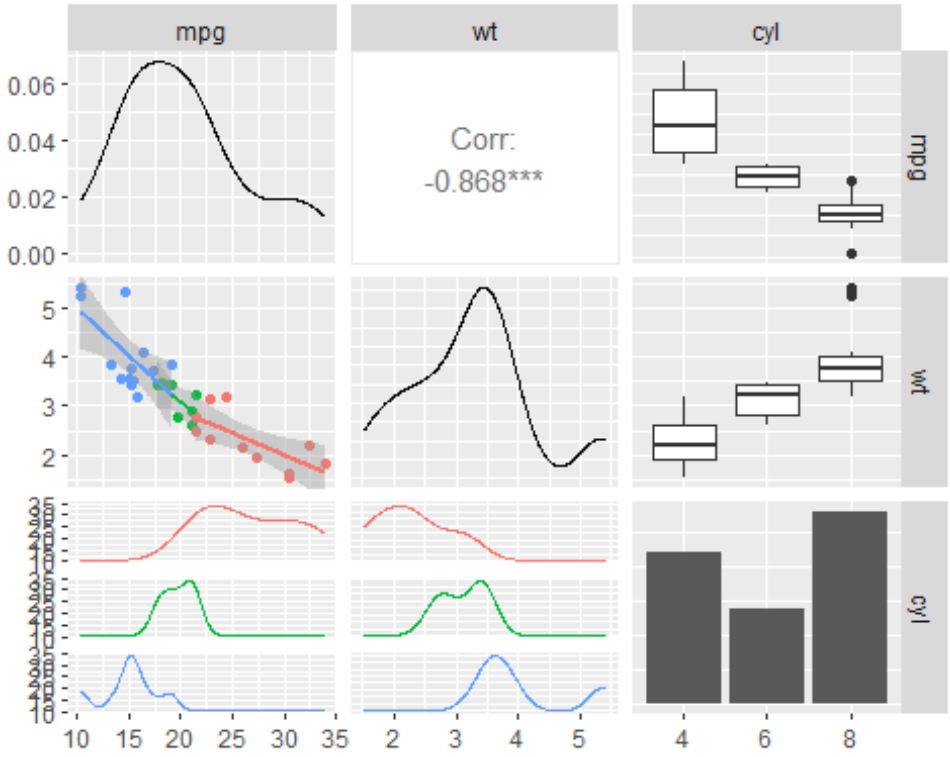
```
# Macierzowy wykres rozrzutu z wykresami gęstości
ggpairs(
  mtcars[, c(1, 4, 6,7)],
  upper = list(continuous = "density", combo = "box_no_facet"),
  lower = list(continuous = "points", combo ="dot_no_facet")
)
```



Rysunek 6.8. Macierzowy wykres rozrzutu z funkcjami gęstości ponad główną przekątną wykresu macierzowego

Źródło: opracowanie własne w programie R.

```
# Macierzowy wykres rozrzutu z wykresami pudełkowymi
data(mtcars)
mtcars$cyl=factor(mtcars$cyl)
ggpairs(
  mtcars, columns =  c("mpg", "wt", "cyl"),
  lower = list(
    continuous = "smooth",
    combo = "facetdensity",
    mapping = aes(color = cyl)
  )
)
```



Rysunek 6.9. Macierzowy wykres rozrzutu z wykresami pudełkowymi ponad przekątną wykresu macierzowego

Źródło: opracowanie własne w programie R.

Na rysunku 6.9 na przekątnej przedstawiono gęstości dla zmiennych ciągłych (*mpg* i *wt*) oraz wykres słupkowy dla zmiennej skokowej (*cyl*). Podobny wykres umieszczono na rysunku 6.10, ale gęstości i wykres słupkowy zostały wykreślone z uwzględnieniem wariantów zmiennej *cyl*.

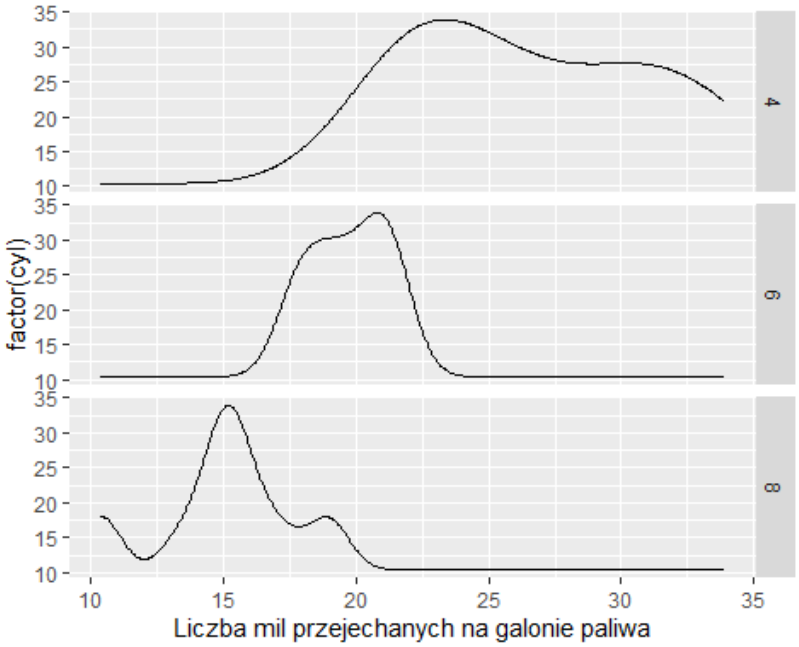
```
# Macierzowy wykres rozrzutu z wykresami pudełkowymi i słupkowym
oraz z wyróżnieniem kategorii
ggpairs(mtcars, columns = c("mpg", "wt", "cyl"), columnLabels =
c("mpg", "wt", "cyl"),aes(color=cyl))
```

Rysunek 6.10. Macierzowy wykres rozrzutu z funkcjami gęstości ponad przekątną macierzy i wyróżnionymi kategoriami ze względu na liczbę cylindrów

Źródło: opracowanie własne w programie R.

Rozkład zmiennej ciągłej może zostać przedstawiony w osobnych panelach ze względu na warianty zmiennej skokowej. Taki wykres realizuje poniższy kod, którego rezultat przedstawiono na rysunku 6.11.

```
# Oszacowania funkcji gęstości – wykres panelowy
ggally_facetdensity(mtcars[,c(1,2)],aes(x=mpg,y=cyl)) +
  labs(x="Liczba mil przejechanych na galonie paliwa")
```



Rysunek 6.11. Wykres panelowy. Liczba mil przejechanych na galonie paliwa w zależności od liczby cylindrów samochodu

Źródło: opracowanie własne w programie R.

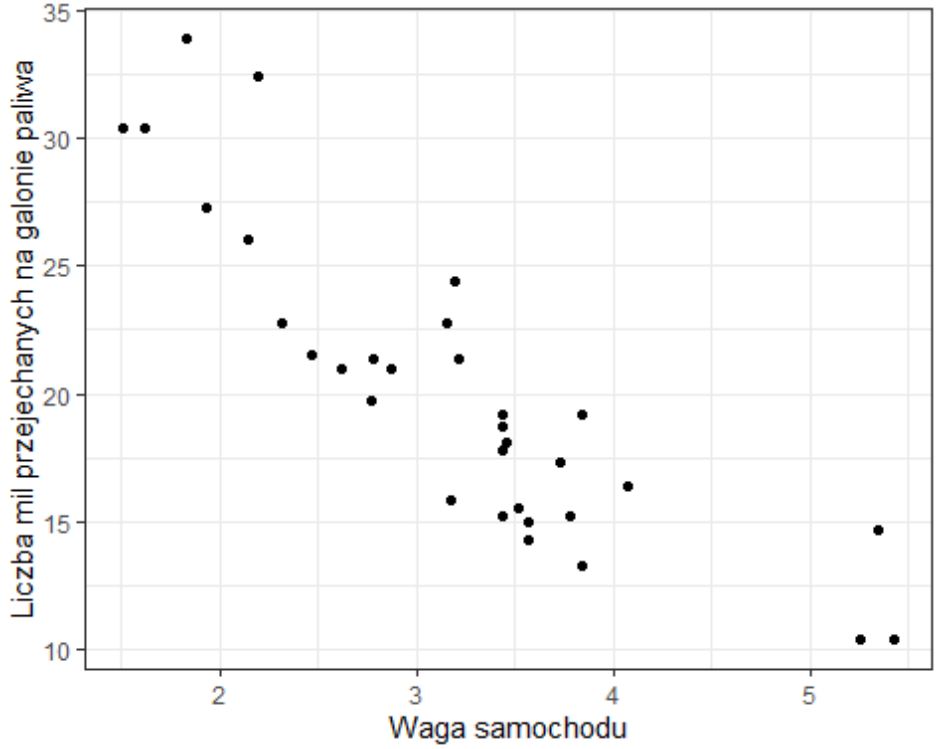
Na rysunku 6.11 przedstawiono oszacowania funkcji gęstości liczby mil przejechanych na galonie paliwa (*mpg*) z uwzględnieniem w oknach panelu samochodów o różnej liczbie cylindrów (*cyl*).

6.2.3. Pakiet ggExtra

Dodatkowe możliwości w zakresie prezentacji zależności pomiędzy zmiennymi zapewnia pakiet **ggExtra**. Pozwala on między innymi na wykreślenie rozkładów brzegowych dla zmiennych przedstawianych na wykresie rozrzutu. Dla zaprezentowania kluczowych możliwości pakietu **ggExtra** wygodnie najpierw skonstruować obiekt graficzny **p** w następujący sposób.

```
# Wykres rozrzutu – konstrukcja obiektu p
p <- ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa')+
  theme_bw()
p
```

Utworzony obiekt **p** to wykres rozrzutu dla dwóch zmiennych *wt* i *mpg*. Wykres uzyskany po realizacji powyższej komendy został przedstawiony na rysunku 6.12.

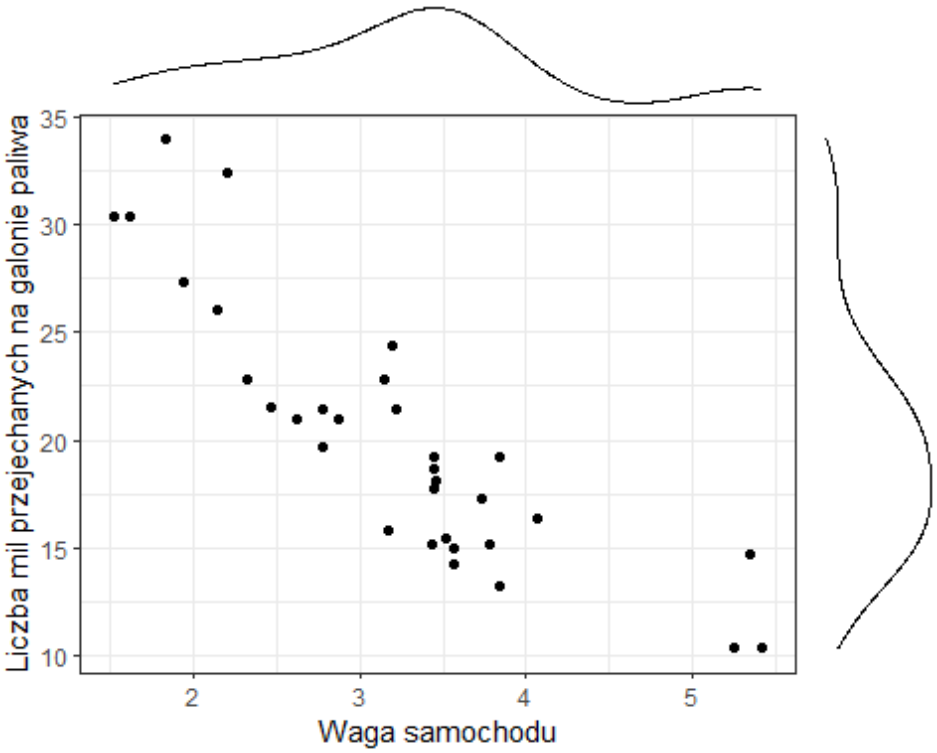


Rysunek 6.12. Obiekt p – wykres rozrzutu dla zmiennych *wt* i *mpg*

Źródło: opracowanie własne w programie R.

W oparciu o funkcję pakietu **ggExtra** obiekt **p** można w prosty sposób modyfikować. Przykładowe sposoby modyfikacji przedstawiają poniżej zaprezentowane kolejne komendy, a rezultaty wykonania tych komend zostały ukazane na wykresach na rysunkach 6.13 i 6.14.

```
Wykres rozrzutu z rozkładami brzegowymi
ggMarginal(p)
```

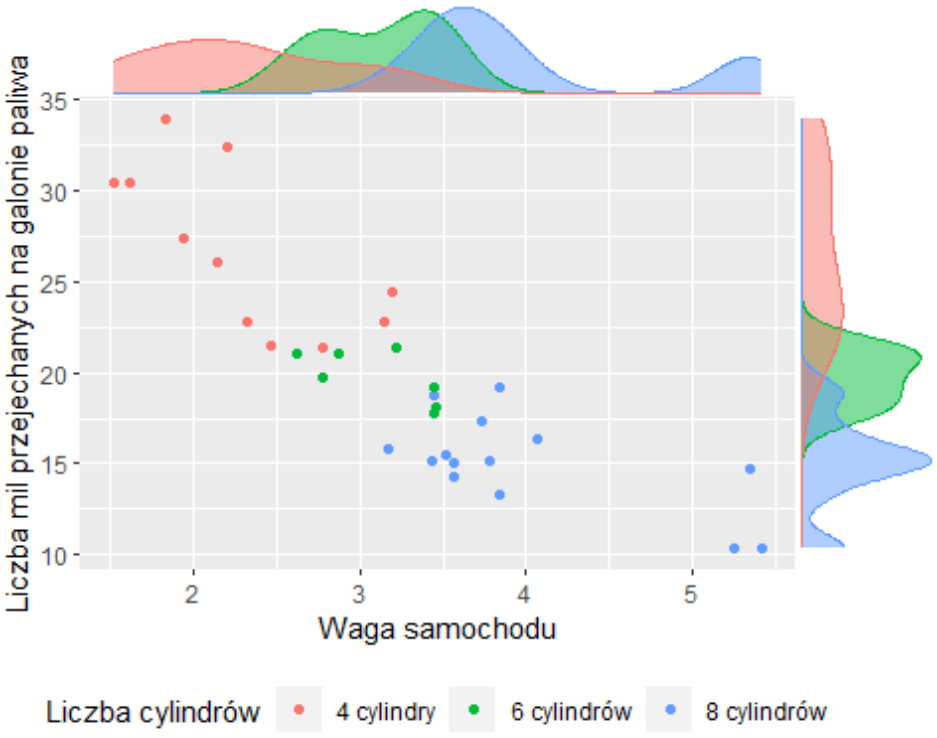


Rysunek 6.13. Obiekt p z dodanymi wykresami gęstości brzegowych dla zmiennych *wt* i *mpg*

Źródło: opracowanie własne w programie R.

Na rysunku 6.13 przedstawiono wykres rozrzutu z dodatkowymi brzegowymi gęstościami obu zmiennych. Na rysunku 6.14 dodano rozróżnienie punktów na wykresie i gęstości brzegowych ze względu na liczbę cylindrów w samochodzie. Uzyskano to w następujący sposób.

```
# Wykres rozrzutu z rozkładami brzegowymi z wyróżnieniem
kategorii
p <- ggplot(mtcars, aes(wt, mpg, colour = cyl)) +
  geom_point()+
    labs(x='Waga samochodu',y='Liczba mil przejechanych na
galonie paliwa',color='Liczba cylindrów')+
  theme(legend.position='bottom')
ggMarginal(p, groupColour = TRUE, groupFill = TRUE)
```

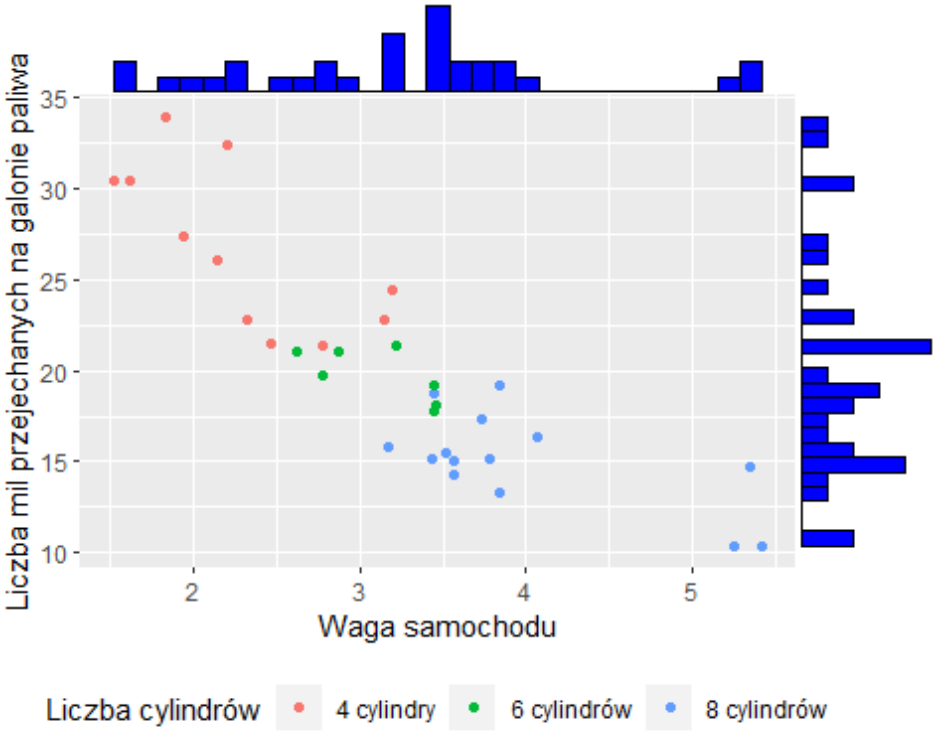


Rysunek 6.14. Obiekt p z dodanymi wykresami gęstości brzegowych dla zmiennych *wt* i *mpg* oraz z rozróżnieniem ze względu na liczbę cylindrów

Źródło: opracowanie własne w programie R.

Zamiast wykreślania rozkładów brzegowych w postaci funkcji gęstości można rozkłady brzegowe przedstawić w postaci histogramów lub wykresów pudełkowych. Odpowiednie komendy przedstawiono poniżej, a rezultaty ich wykonania znajdują się na rysunkach 6.15 i 6.16.

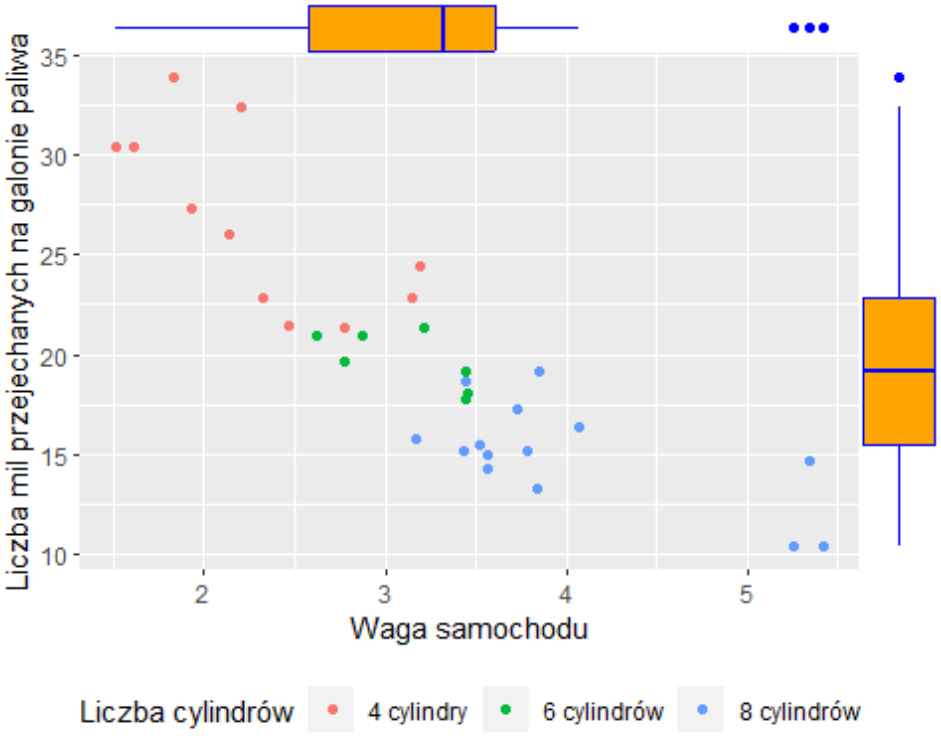
```
# Wykres rozrzutu z rozkładami brzegowymi w postaci histogramów
ggMarginal(p, type = "histogram",fill = "blue")
```



Rysunek 6.15. Obiekt p z dodanymi histogramami brzegowymi dla zmiennych *wt* i *mpg*

Źródło: opracowanie własne w programie R.

```
# Wykres rozrzutu z rozkładami brzegowymi w postaci wykresów
pudełkowych
ggMarginal(p,  size = 10, type = "boxplot",
           col = "blue", fill = "orange")
```



Rysunek 6.16. Obiekt p z dodanymi wykresami pudełkowymi brzegowymi dla zmiennych *wt* i *mpg*

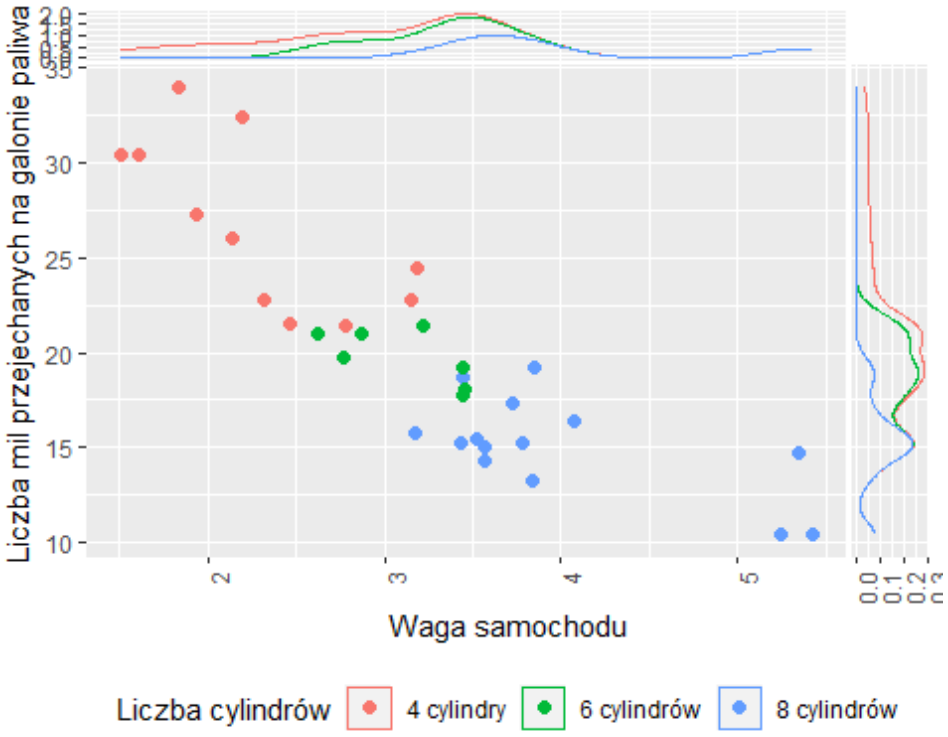
Źródło: opracowanie własne w programie R.

Wykresy przedstawione na rysunkach 6.15 i 6.16 pozwalają nie tylko ocenić kierunek i siłę zależności pomiędzy badanymi zmiennymi, ale również pokazać strukturę rozkładów obu analizowanych zmiennych.

6.2.4. Pakiet `ggsides`

Przedstawiony w poprzednim punkcie pakiet **`ggExtra`** pozwalał między innymi na dodanie do wykresów rozrzutu graficznej prezentacji rozkładów brzegowych analizowanych zmiennych. Podobne wykresy można uzyskać, wykorzystując bibliotekę **`ggsides`**. Przykład zastosowania tej biblioteki dla zbioru **mtcars** przedstawia poniższy kod, a wynik jest zaprezentowany na rysunku 6.17.

```
# Wykres rozrzutu z gęstościami brzegowymi i wyróżnionymi
kategoriami
ggplot(mtcars, aes(wt, mpg, colour = cyl)) +
  geom_point(size = 2) +
  geom_xsidedensity(aes(y = after_stat(density)), position =
"stack") +
  geom_ysidedensity(aes(x = after_stat(density)), position =
"stack") +
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa',colour='Liczba cylindrów')+
  theme(legend.position='bottom',axis.text.x =
element_text(angle = 90))
```



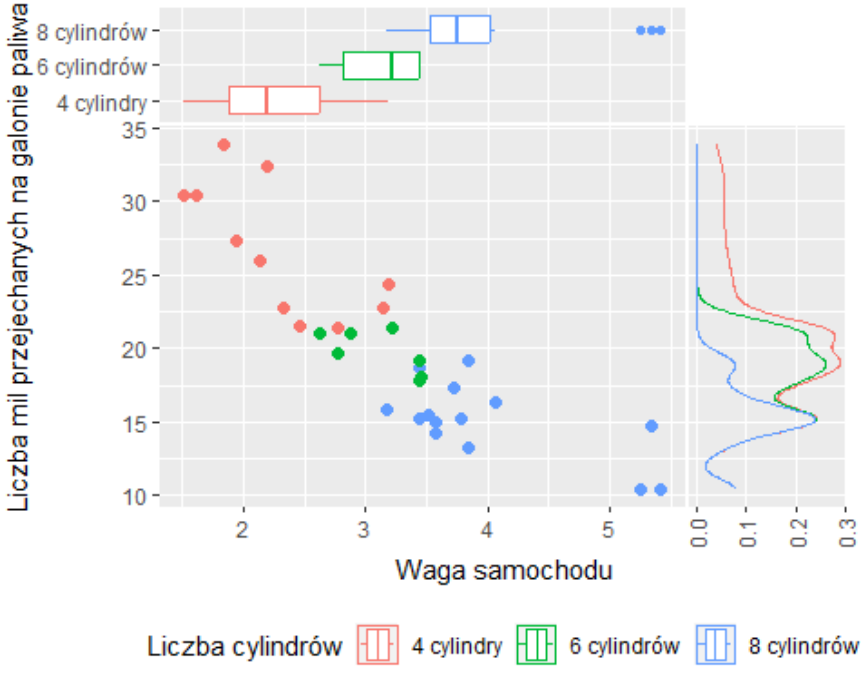
Rysunek 6.17. Wykres rozrzutu z brzegowymi rozkładami gęstości

Źródło: opracowanie własne w programie R.

Nieco inną formę prezentacji rozkładów brzegowych (por. rysunek 6.18) przedstawia poniższy kod.

```
# Wykres rozrzutu z brzegowymi gęstościami i wykresem pudełkowym
i wyróżnionymi kategoriami
ggplot(mtcars, aes(wt, mpg, colour = cyl)) +
  geom_point(size = 2) +
  geom_xsideboxplot(aes(y =cyl), orientation = "y") +
  scale_xsidey_discrete() + #In order to use xsideboxplot with a
main panel that uses
  geom_ysidedensity(aes(x = after_stat(density)), position =
"stack") +
  scale_ysidex_continuous(guide = guide_axis(angle = 90),
minor_breaks = NULL) +
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa',colour='Liczba cylindrów')+
  theme(legend.position='bottom',ggside.panel.scale = .3)
```

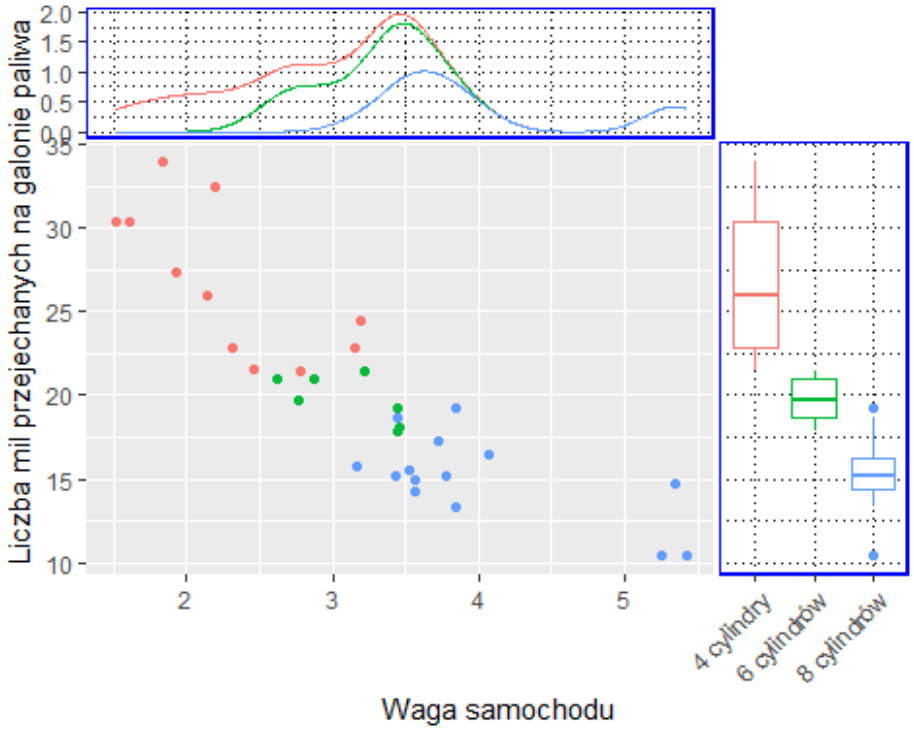
Na rysunku 6.18 przedstawiono wykres rozrzutu podobnie jak na rysunku 6.17, ale rozkłady brzegowe ujęto w formie wykresów gęstości i pudełkowych. Nieco inną realizację od strony graficznej zaprezentowano na rysunku 6.19.



Rysunek 6.18. Wykres rozrzutu z brzegowymi rozkładami gęstości i wykresu pudełkowego

Źródło: opracowanie własne w programie R.

```
# Wykres rozrzutu z rozkładami brzegowymi w formie gęstości i
pudełkowego oraz z wyróżnionymi kategoriami
ggplot(mtcars, aes(wt, mpg, colour = cyl)) +
  geom_point(aes(color = cyl)) +
  geom_xsidedensity(alpha = .3, position = "stack") +
  geom_ysideboxplot(aes(x = cyl), orientation = "x") +
  scale_ysidex_discrete(guide = guide_axis(angle = 45)) +
  labs(x='Waga samochodu',y='Liczba mil przejechanych na galonie
paliwa',colour='Liczba cylindrów')+
  theme(legend.position='bottom',ggside.panel.scale = .3,
        ggside.panel.border = element_rect(NA, "blue", linewidth
= 1),
        ggside.panel.grid = element_line("black", linewidth =
.1, linetype = "dotted"),
        ggside.panel.background = element_blank()) +
  guides(color = "none", fill = "none")
```



Rysunek 6.19. Wykres rozrzutu z brzegowymi rozkładami gęstości i wykresu pudełkowego bez legendy

Źródło: opracowanie własne w programie R.

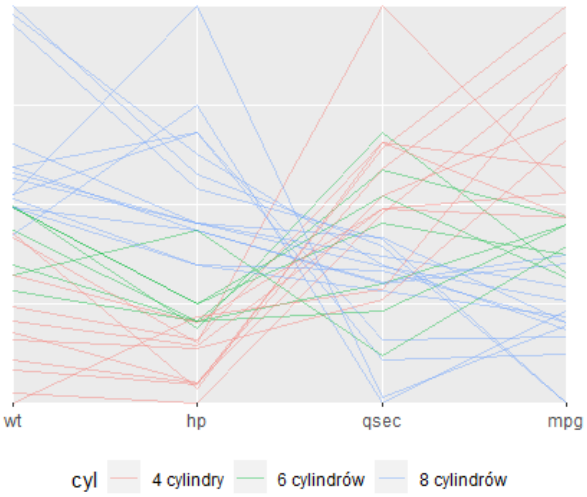
6.3. Graficzna prezentacja danych wielowymiarowych

W poprzednim punkcie skoncentrowano się na przedstawieniu zależności pomiędzy zmiennymi. Nieco inne możliwości analizowania danych wielowymiarowych przedstawiono w tym punkcie. Dla różnych prezentacji danych wielowymiarowych zostaną wykorzystane pakiety **ggmulti**, **ggridges** oraz **ggmosaic**.

6.3.1. Pakiet ggmulti

Pakiet **ggmulti** pozwala na graficzną prezentację danych wielowymiarowych. Poniższy kod przedstawia wizualizację wybranych zmiennych ze zbioru **mtcars** w formie wykresu o współrzędnych równoległych. Rezultat został zamieszczony na rysunku 6.20.

```
# Wykres współrzędnych równoległych
p <- ggplot(mtcars,
      mapping = aes(wt = wt,hp = hp,qsec = qsec,mpg = mpg,
                    colour = cyl)) +
  geom_path(alpha = 0.4) +
  theme(legend.position='bottom')+
  coord_serialaxes(axes.layout = "parallel",scaling =
"variable")
p
```

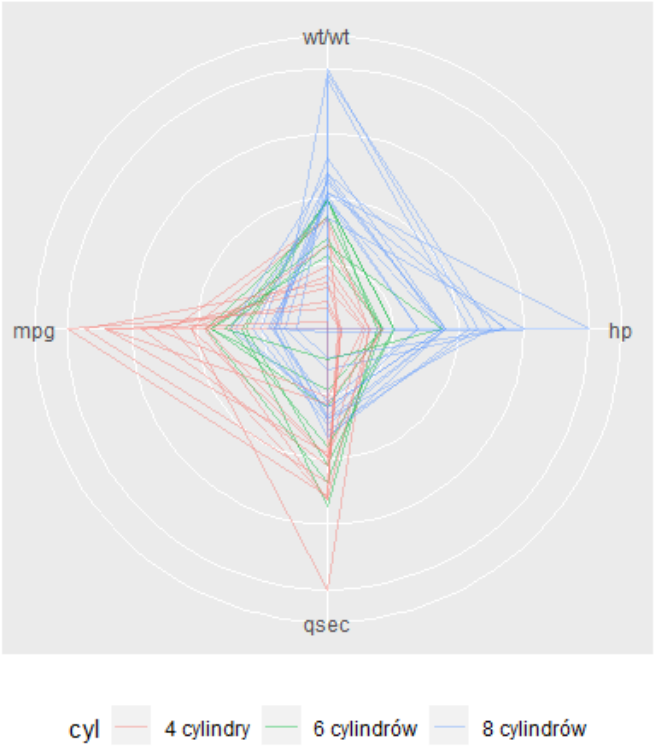


Rysunek 6.20. Wykres współrzędnych równoległych dla wybranych zmiennych ze zbioru mtcars

Źródło: opracowanie własne w programie R.

Kolejny kod oraz rysunek 6.21 przedstawiają te same informacje co na rysunku 6.20, ale w ujęciu współrzędnych biegunowych.

```
# Wykres radarowy
p+
  coord_serialaxes(axes.layout =  "radial",scaling = "variable")
```

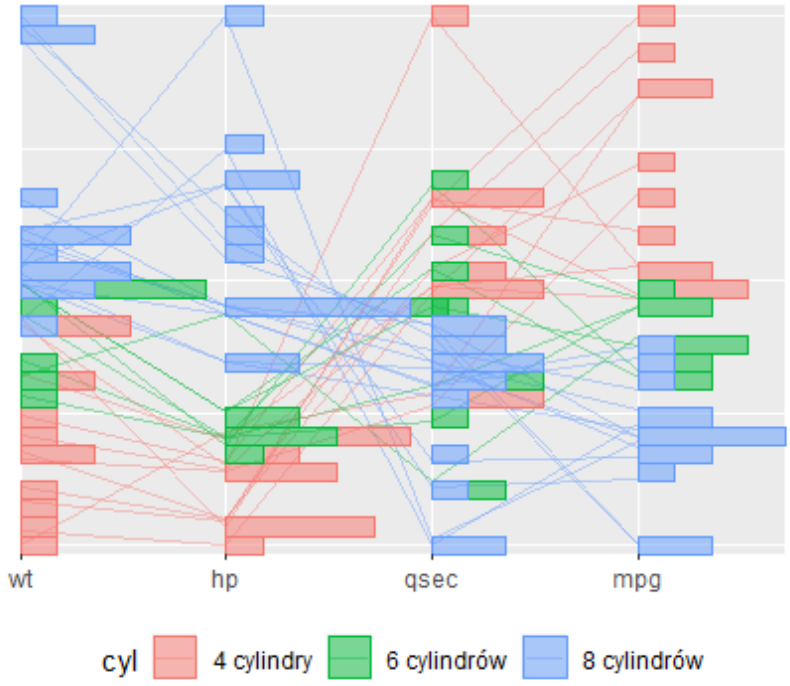


Rysunek 6.21. Wykres współrzędnych równoległych dla wybranych zmiennych ze zbioru mtcars we współrzędnych biegunowych

Źródło: opracowanie własne w programie R.

Na wykres przedstawiony na rysunku 6.20 można dodatkowo nanieść rozkłady brzegowe poszczególnych zmiennych. Poniższy kod i rysunek 6.22 przedstawiają przykład wykorzystania histogramów do prezentacji rozkładów brzegowych.

```
# Wykres współrzędnych równoległych z histogramami
p +
  geom_histogram(mapping = aes(fill = cyl), alpha = 0.5)
```



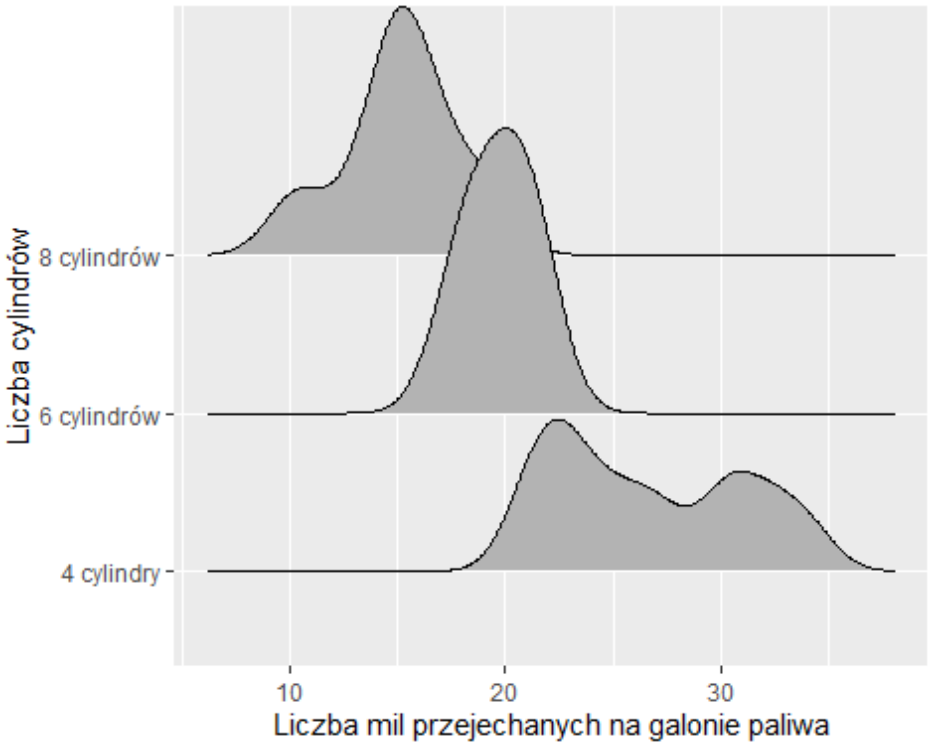
Rysunek 6.22. Wykres współrzędnych równoległych z rozkładami brzegowymi w formie histogramów dla wybranych zmiennych ze zbioru mtcars

Źródło: opracowanie własne w programie R.

6.3.2. Pakiet ggridges

Pakiet **ggridges** pozwala na konstrukcję częściowo nakładających się na siebie wykresów liniowych, które tworzą wrażenie pasma górskiego. Wykresy takie mogą być bardzo przydatne do wizualizacji zmian w rozkładach w czasie lub przestrzeni. Podstawową konstrukcję takiego wykresu dla danych ze zbioru **mtcars** przedstawia następujący kod.

```
# Wykres gęstości względem wyróżnionych kategorii
ggplot(mtcars, aes(x = mpg, y = cyl, group = cyl)) +
    labs(x='Liczba mil przejechanych na galonie
paliwa',y='Liczba cylindrów')+
  geom_density_ridges()
```

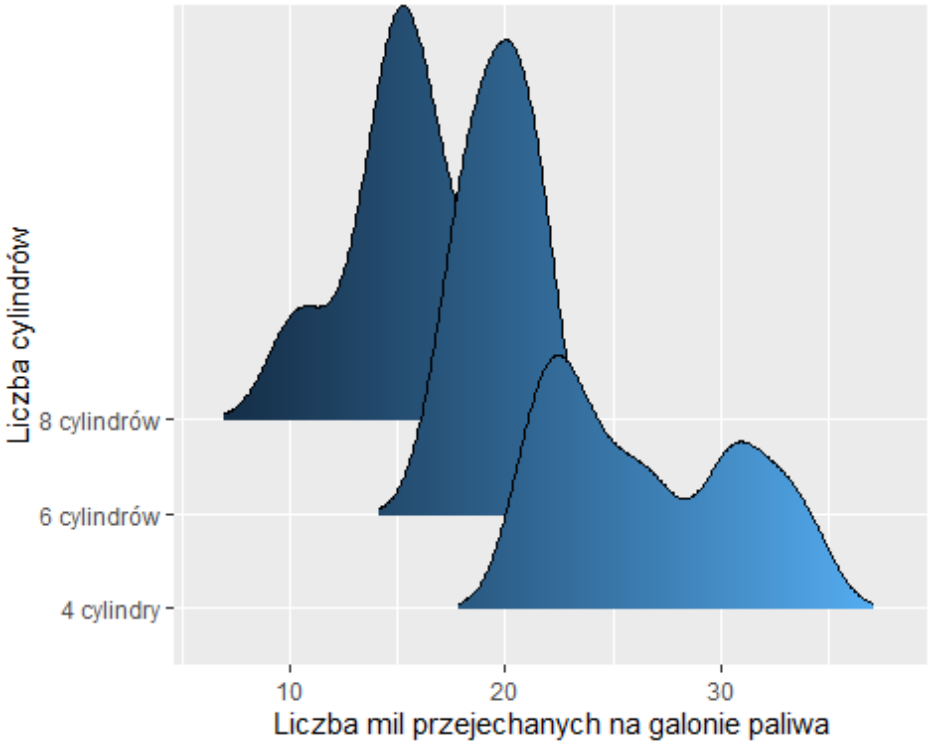


Rysunek 6.23. Linie ridges dla zmiennej *mpg* względem liczby cylindrów

Źródło: opracowanie własne w programie R.

Na rysunku 6.23 przedstawiono gęstości liczby mil przejechanych na jednym galonie paliwa dla trzech wyróżnionych grup ze względu na liczbę cylindrów. Poniższy kod do poprzedniego wykresu dodaje wypełnienie obszaru pod funkcją gęstości, którego tonacja zależy od wartości zmiennej *mpg*. Rezultat prezentuje rysunek 6.24.

```
# Wykres gęstości względem wyróżnionych kategorii z natężeniem
wartości
ggplot(mtcars, aes(x = mpg, y = cyl, fill = after_stat(x))) +
  geom_density_ridges_gradient(scale = 5, rel_min_height = 0.01)
+
  labs(x='Liczba mil przejechanych na galonie paliwa', y='Liczba
cylindrów')+
  theme(legend.position='none')
```



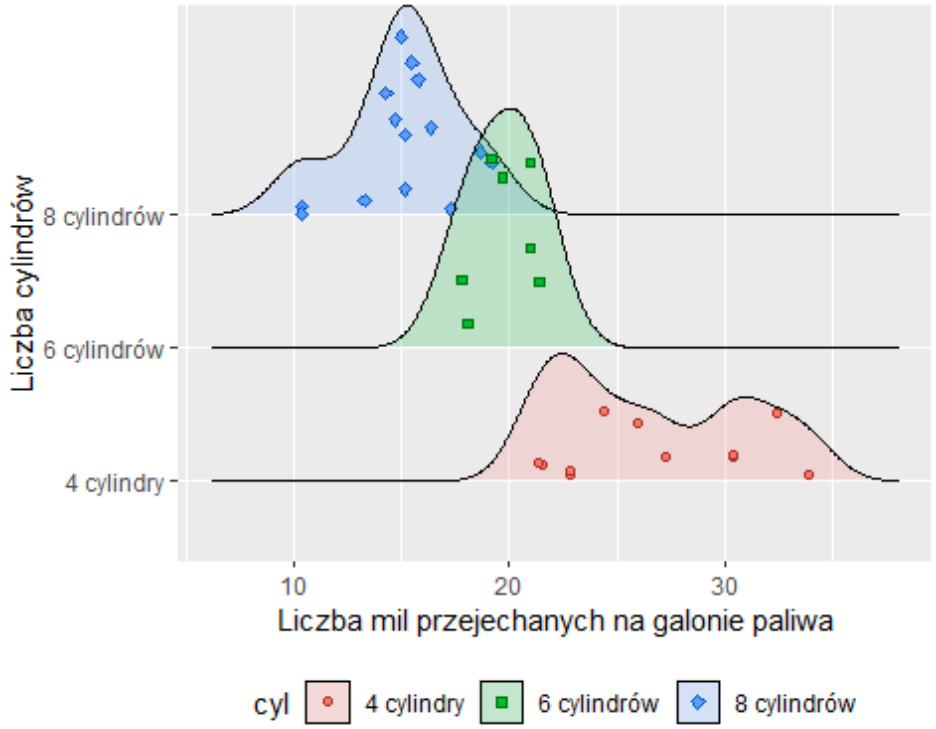
Rysunek 6.24. Linie ridges dla zmiennej *mpg* względem liczby cylindrów z zaznaczeniem intensywności zmiennej zależnej

Źródło: opracowanie własne w programie R.

Na wykresie można dodatkowo wprowadzić wszystkie obserwacje z reprezentacją punktów z wykorzystaniem niniejszego kodu.

```
# Wykres gęstości względem wyróżnionych kategorii z rozrzuconymi
obserwacjami
ggplot(mtcars, aes(x = mpg, y = cyl, fill = cyl)) +
  geom_density_ridges(
    aes(point_color = cyl, point_fill = cyl, point_shape = cyl),
    alpha = .2, point_alpha = 1, jittered_points = TRUE  ) +
  labs(x='Liczba mil przejechanych na galonie paliwa', y='Liczba
cylindrów')+
    theme(legend.position='bottom')+
  scale_point_color_hue(l = 40) +
  scale_discrete_manual(aesthetics = "point_shape", values =
c(21, 22, 23))
```

Wynik powyższego kodu zamieszczono na rysunku 6.25.

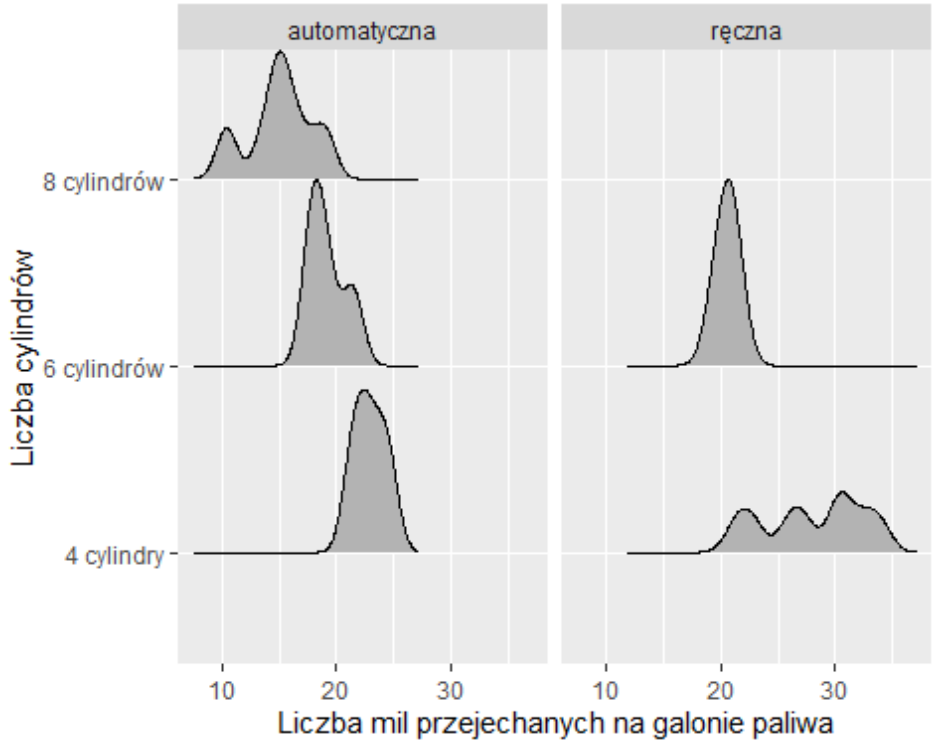


Rysunek 6.25. Linie ridges dla zmiennej *mpg* względem liczby cylindrów z wyróżnieniem pojedynczych obserwacji

Źródło: opracowanie własne w programie R.

Podobnie jak we wcześniej prezentowanych wykresach możliwe jest wyróżnienie paneli ze względu na wybraną zmienną dyskretną. Taką możliwość realizuje następujący kod.

```
# Panele wykresu gęstości względem wyróżnionych kategorii
ggplot(mtcars, aes(x = mpg, y = cyl)) +
  geom_density_ridges(scale = 1) +
    labs(x='Liczba mil przejechanych na galonie paliwa',
y='Liczba cylindrów')+
  facet_wrap(~am)
```



Rysunek 6.26. Linie ridges dla zmiennej *mpg* względem liczby cylindrów w ujęciu panelowym dla zmiennej *am*

Źródło: opracowanie własne w programie R.

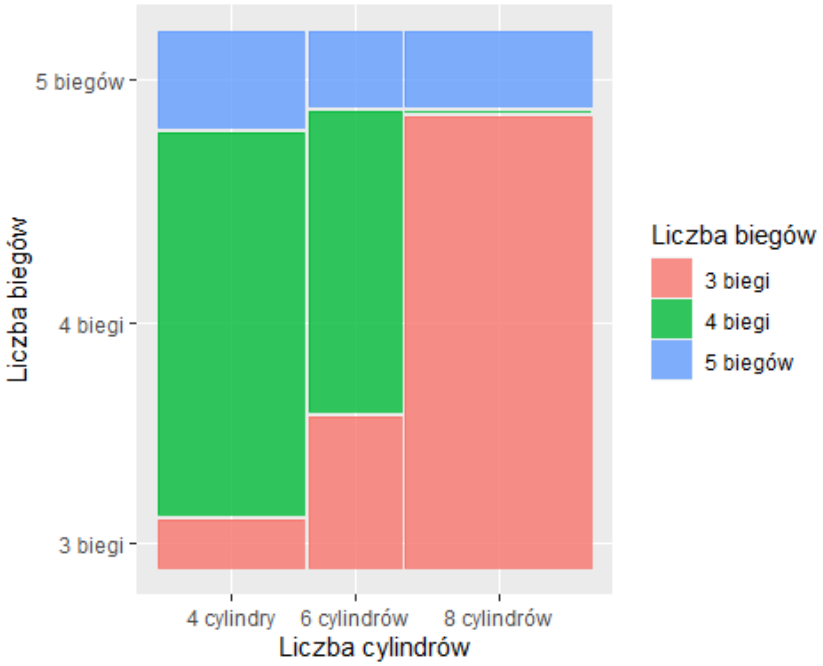
Na rysunku 6.26 przedstawiono empiryczne gęstości liczby przejechanych mil na galonie paliwa względem liczby cylindrów w ujęciu panelowym z uwagi na rodzaj skrzyni biegów.

6.3.3. Pakiet `ggmosaic`

Rezultaty analiz statystycznych często przedstawia się w tablicach wielodzielczych. Jeżeli dane są zamieszczone w tablicy dwuwymiarowej, to graficzna prezentacja może być ograniczona do odpowiednio skonstruowanych wykresów słupkowych (Kończak i Żądło 2010; Kończak i Kosińska 2023). Jeżeli jednak dane są zamieszczone w wielowymiarowej tablicy wielodzielczej, to do przedstawienia zależności pomiędzy zmiennymi jakościowymi mogą być wykorzystane różne wersje wykresu mozaikowego (Friendly 1994; Albert i Rizzo 2012). Wykres ten został opisany w punkcie 3.1.15 niniejszej monografii, a przykład takiego wykresu został zamieszczony na rysunkach 4.7 i 4.8. Dobrą praktyką

jest, aby na wykresie mozaikowym kolorem oznaczać warianty zmiennej zależnej. Pakiet **ggmosaic** umożliwia wykonanie wykresu jak na rysunku 6.27 z wykorzystaniem następującego kodu.

```
# Wykres mozaikowy
ggplot(mtcars) +
 geom_mosaic(aes(x = product(cyl), fill = gear))+
     labs(x='Liczba cylindrów', y='Liczba biegów',fill='Liczba
biegów')
```

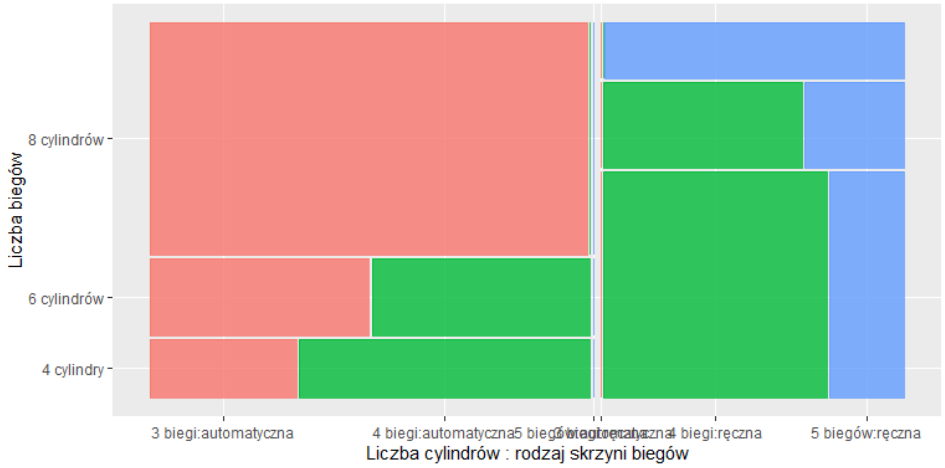


Rysunek 6.27. Wykres mozaikowy. Struktura liczby samochodów ze względu na liczbę cylindrów i biegów

Źródło: opracowanie własne w programie R.

Kolejny kod pozwala wprowadzić do dwóch wyróżnionych zmiennych kolejną – rodzaj skrzyni biegów (*am*).

```
# Wykres mozaikowy
ggplot(mtcars) +
  geom_mosaic(aes(x = product(cyl,am), fill = gear))+
       labs(x='Liczba cylindrów : rodzaj skrzyni biegów',
y='Liczba biegów')+
  theme(legend.position = 'none')
```

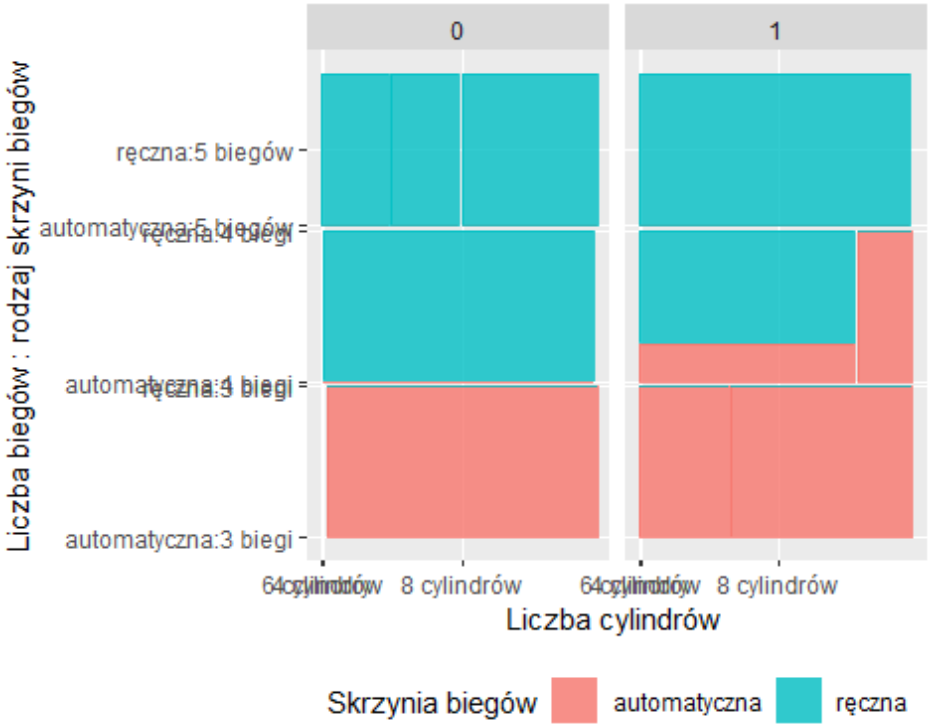


Rysunek 6.28. Wykres mozaikowy. Liczba cylindrów, biegów oraz rodzaj skrzyni biegów

Źródło: opracowanie własne w programie R.

Na rysunku 6.28 przedstawiono wykres mozaikowy, na którym zobrazowano strukturę liczby samochodów ze względu na trzy zmienne jakościowe: liczbę biegów, liczbę cylindrów oraz rodzaj skrzyni biegów. Podobnie jak w przypadku omawianych już wykresów możliwe jest wprowadzenie układu panelowego. Pozwala to dodać kolejną zmienną lub zwiększyć nieco czytelność wykresu. Konstrukcję takiego wykresu przedstawia poniższy kod, a rezultat zobrazowano na rysunku 6.29.

```
# Wykres mozaikowy w układzie panelowym
ggplot(mtcars) +
  geom_mosaic(aes(x = product(cyl), conds=product(gear), fill =
am))+
  labs(y='Liczba cylindrów ', x='Liczba biegów : rodzaj skrzyni
biegów',fill="Skrzynia biegów")+
  coord_flip()+
  facet_wrap(~vs)+
  theme(legend.position = 'bottom')
```



Rysunek 6.29. Wykres mozaikowy w układzie panelowym. Liczba cylindrów, biegów, rodzaj skrzyni biegów i kształt silnika

Źródło: opracowanie własne w programie R.

6.4. Inne wybrane reprezentacje geometryczne

W rozdziale 3 przedstawiono charakterystyki wielu różnych reprezentacji geometrycznych. W poprzednim rozdziale oraz w poprzednich punktach bieżącego rozdziału omówiono praktyczne zastosowania wielu z tych reprezentacji z wykorzystaniem gramatyki grafiki zaimplementowanej w pakiecie **`ggplot2`**. W tym punkcie przedstawiono wybrane, zwykle rzadko stosowane rodzaje wykresów i niekiedy nieposiadające implementacji we wspomnianym pakiecie. Ujęto oczywiście tylko wybrane reprezentacje geometryczne, które mogą być przydatne przy wizualizacji wyników badań naukowych, a w szczególności w analizie danych wielowymiarowych. Do realizacji kodów zamieszczonych w tym punkcie niezbędne jest załadowanie pakietów ujętych w tabeli 6.2, a także wskazanych uprzednio w tabeli 6.1.

Tabela 6.2. Biblioteki wykorzystywane w bieżącym punkcie

Biblioteka	Opis
`ggChernoff`	Konstrukcja wykresów opartych na twarzach Chernoffa
`HistData`	Zbiór danych ważnych w historii statystyki
`aplpack`	Umożliwia konstrukcję wielu specjalnych wykresów
`car`	Funkcje wspomagające analizę regresji
`reshape2`	Funkcje do przeorganizowywania i agregacji danych
`vcd`	Wizualizacja danych jakościowych

Źródło: opracowanie własne.

Dla przedstawienia możliwości prezentowanych reprezentacji graficznych w pierwszym kroku należy załadować niezbędne biblioteki.

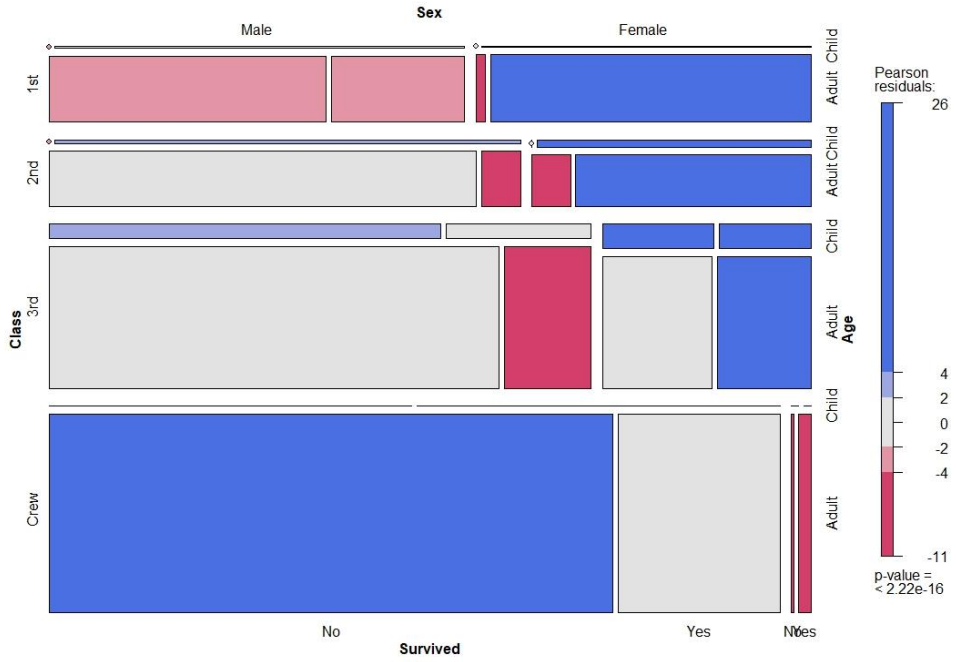
```
library(ggChernoff)
library(HistData)
library(aplpack)
library(car)
library(reshape2)
library(vcd)
```

6.4.1. Wykres mozaikowy

W punkcie 6.3.3 przedstawiono podstawowe zasady konstrukcji wykresów mozaikowych z pakietem **`ggmosaic`**. Warto jednak zaznaczyć, że nie jest to jedyny pakiet, który pozwala na konstrukcję wykresów mozaikowych. Jednym z takich pakietów jest **`vcd`**. Nie jest to pakiet zbudowany na idei Grammar of Graphics, nie stanowi też rozszerzenia pakietu **`ggplot2`**. Jednak ze względu na możliwości w zakresie wizualizacji danych jakościowych warto go przedstawić.

Poniższa komenda konstruuje wykres mozaikowy (mosaic plot) na podstawie danych ze zbioru **Titanic**.

```
# Konstrukcja wykresu mozaikowego
mosaic(Titanic,shade=TRUE)
```



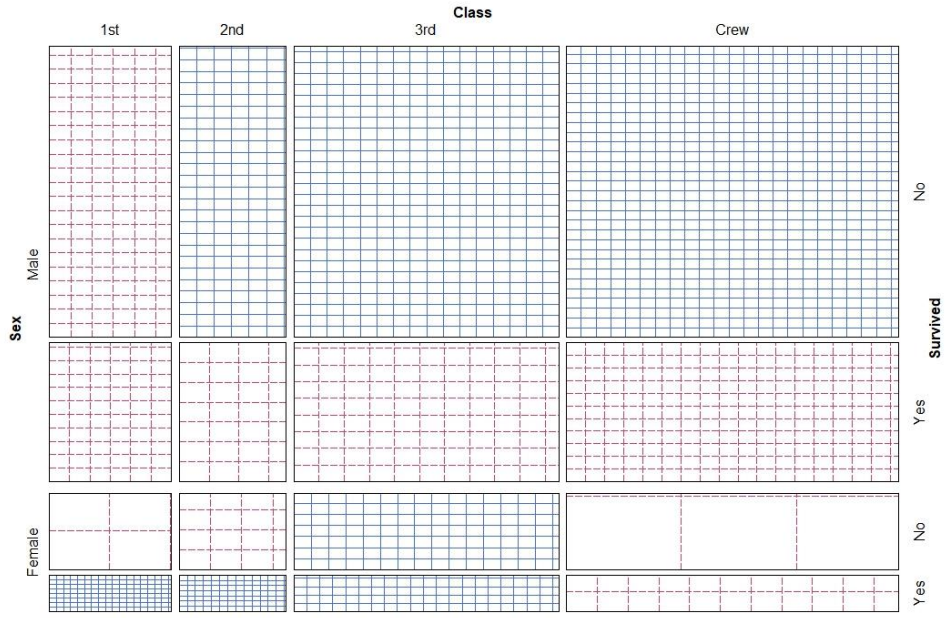
Rysunek 6.30. Wykres mozaikowy dla zbioru Titanic. Przeżycie katastrofy pasażerów Titanica w zależności od płci, wieku i klasy

Źródło: opracowanie własne w programie R.

Wykres przedstawiony na rysunku 6.30 prezentuje strukturę relacji między czterema zmiennymi jakościowymi zbioru **Titanic**. Ustawienie parametru shade=TRUE sprawia, że komórki są kolorowane. Intensywne kolory odpowiadają komórkom wielowymiarowej tablicy wielodzielczej, dla których występują istotne statystycznie różnice pomiędzy liczebnościami obserwowanymi a liczebnościami oczekiwanymi.

Kolejny kod pozwala na konstrukcję wykresu sita.

```
# Konstrukcja wykresu sita
tit <- margin.table(Titanic, c(2,1,4))
sieve(tit,  shade = TRUE)
```



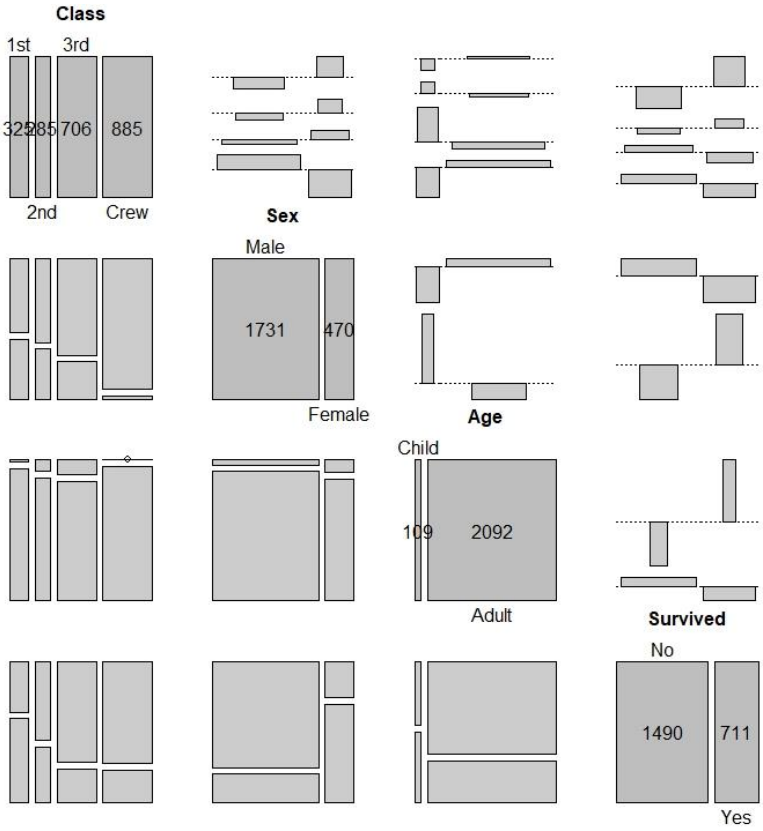
Rysunek 6.31. Wykres sita dla trzech zmiennych ze zbioru Titanic. Przeżycie katastrofy pasażerów Titanica w zależności od płci, wieku i klasy

Źródło: opracowanie własne w programie R.

Powyższa komenda utworzyła tabele brzegowe, które obejmowały relacje między klasą pasażera, płcią i przeżyciem w danych ze zbioru **Titanic**. Następnie przedstawiono wyniki na wykresie sita (rysunek 6.31). Ciemniejsze odcienie na tym wykresie oznaczają relatywnie większą liczbę obserwacji.

Poniższy kod konstruuje tablicę wykresów mozaikowych dla par wszystkich zmiennych ze zbioru **Titanic**. Rezultat został zamieszczony na rysunku 6.32.

```
# Konstrukcja wykresu asocjacji par
pairs(Titanic,upper_panel = pairs_assoc)
```



Rysunek 6.32. Asocjacje dla par zmiennych zbioru Titanic. Przeżycie katastrofy pasażerów Titanica w zależności od płci, wieku i klasy

Źródło: opracowanie własne w programie R.

Na rysunku 6.32 przedstawiono tablicę wykresów mozaikowych. W poszczególnych panelach pod przekątną znajdują się wykresy mozaikowe poszczególnych par zmiennych jakościowych. Ponad przekątną umieszczone zostały wykresy asocjacji, które wskazują standaryzowane różnice pomiędzy liczebnościami obserwowanymi a liczebnościami oczekiwanymi.

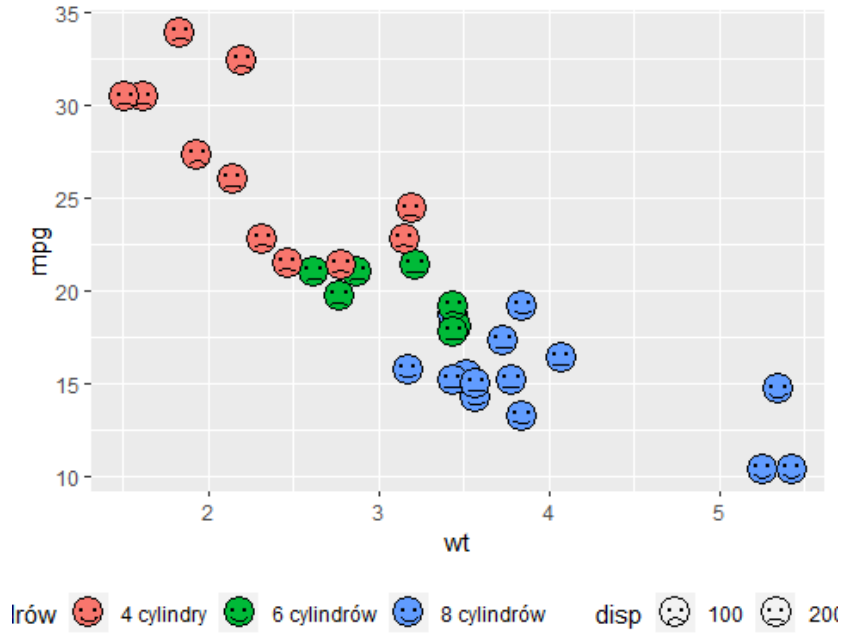
6.4.2. Twarze Chernoffa

W punkcie 3.1.17 przedstawiono charakterystykę wykresów Chernoff faces. Jest to stosunkowo rzadko wykorzystywana forma wykresu, ale niekiedy może okazać się bardzo pomocna przy analizach wielowymiarowych. Twarze Chernoffa prezentują wielowymiarowe dane w kształcie ludzkiej twarzy. Poszczególne części, takie jak oczy, uszy, usta i nos, reprezentują wartości zmiennych poprzez

swój kształt, rozmiar, rozmieszczenie i orientację. Uzasadnieniem wykorzystania twarzy na wykresie jest to, że człowiek łatwo rozpoznaje twarze i bez trudu zauważa w ich obrębie nawet niewielkie różnice. Wykresy twarzy Chernoffa obsługują każdą zmienną poprzez inną charakterystykę twarzy. Ponieważ cechy twarzy różnią się pod względem postrzeganej ważności, sposób mapowania zmiennych na cechy powinien być starannie dobrany. Sporządzenie wykresu tego typu umożliwia pakiet **ggChernoff**, a przykładowy kod wygląda następująco.

```
# Konstrukcja wykresu twarzy Chernoffa
ggplot(mtcars) +
  aes(wt, mpg, fill = factor(cyl),smile=disp) +
  labs(fill='Liczba cylindrów')+
  theme(legend.position='bottom')+
  geom_chernoff()
```

Rezultat wykonania kodu przedstawia rysunek 6.33. Kolor twarzy jest powiązany z liczbą cylindrów samochodu, a uśmiech (*smile*) z pojemnością silnika. Dodatkowo jest możliwość mapowania zmiennych na następujące cechy twarzy: nos (*nose*), brwi (*brow*) oraz oczy (*eyes*).



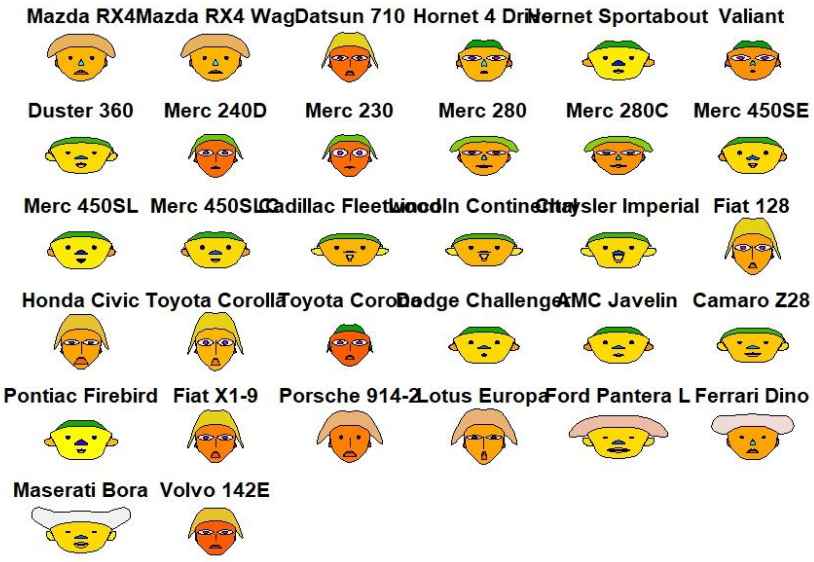
Rysunek 6.33. Twarze Chernoffa – reprezentacja czterech zmiennych. Waga samochodu, liczba mil przejechanych na galonie paliwa, liczba cylindrów oraz pojemność silnika

Źródło: opracowanie własne w programie R.

Znacznie większe możliwości w tym zakresie daje pakiet **aplpack**. Przykładowy kod z mapowaniem 16 zmiennych (niektóre ze zmiennych mapowane są dwukrotnie) ze zbioru **mtcars** jest następujący.

```
# Załadowanie zbioru i konstrukcja wykresu
data(mtcars)
faces(mtcars)
```

Wynik realizacji powyższego kodu przedstawiono na rysunku 6.34. Natomiast w tabeli 6.3 wskazano sposób mapowania zmiennych na cechy twarzy. Warto zauważyć, że w niektórych przypadkach ta tabela wykorzystuje te same zmienne (*mpg*, *cyl*, *disp*, *hp*) dla różnych elementów twarzy Chernoffa, co może wpływać na interpretację wizualizacji.



Rysunek 6.34. Twarze Chernoffa – reprezentacja jedenastu zmiennych dla zbioru mtcars

Źródło: opracowanie własne w programie R.

Tabela 6.3. Mapowanie elementów twarzy dla zmiennych zbioru mtcars

Element twarzy	Zmienna
1	*2*
Wysokość twarzy	*mpg*
Szerokość twarzy	*cyl*
Struktura twarzy	*disp*
Wysokość ust	*hp*
Szerokość ust	*drat*

cd. tabeli 6.3

1	2
Uśmiech	*wt*
Wysokość oczu	*qsec*
Szerokość oczu	*vs*
Wysokość włosów	*am*
Szerokość włosów	*gear*
Styl włosów	*carb*
Wysokość nosa	*mpg*
Szerokość nosa	*cyl*
Szerokość ucha	*disp*
Wysokość ucha	*hp*

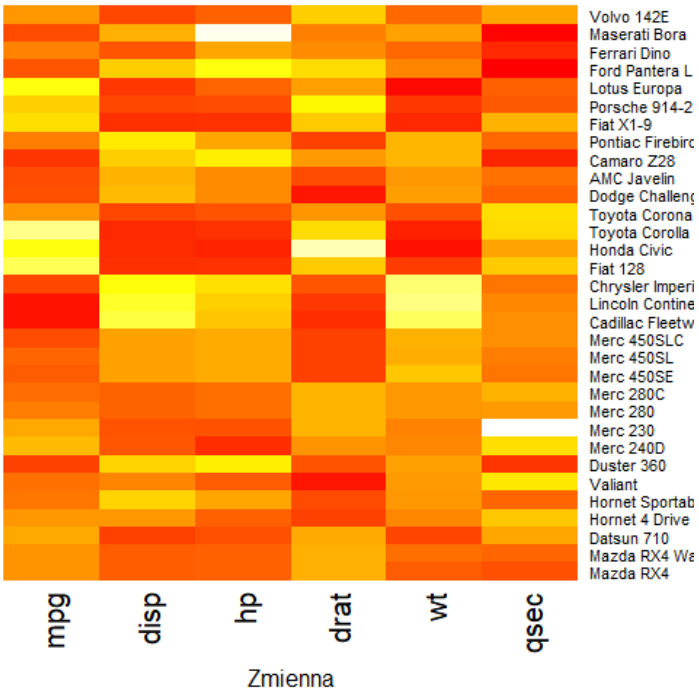
Źródło: opracowanie własne na podstawie CRAN R Project. *aplpack* (b.r.).

6.4.3. Wykres ciepła (heatmap)

Wykres ciepła (heatmap) to taka wizualizacja danych, która pozwala na przedstawienie wartości dla jednej zmiennej w zależności od dwóch innych zmiennych jako siatkę różnokolorowych prostokątów. Zmienne osi są podzielone na zakresy (kategorie lub przedziały), a kolor każdej komórki wskazuje wartość zmiennej głównej w odpowiadającym zakresie komórki. Wykresy takie wykorzystuje się do prezentacji związków między dwiema zmiennymi. Szczególną formą wykresów ciepła są wcześniej przedstawione na rysunkach 6.1 i 6.4 wykresy zbudowane na macierzy współczynników korelacji liniowej. Zmienne na obu osiach mogą być zarówno jakościowe, jak i liczbowe. Kolory komórek mogą odpowiadać różnym metrykom, takim jak częstotliwość punktów w każdym przedziale, lub statystykom podsumowującym, jak średnia bądź mediana dla trzeciej zmiennej. Konstrukcję wykresu ciepła można postrzegać jako swoistą tabelę lub macierz. Przykładową konstrukcję wykresu ciepła przedstawia poniższy kod.

```
# Wykres ciepła z użyciem funkcji heatmap
mtcars_matrix <- as.matrix(mtcars[,c(1,3,4,5,6,7)])
heatmap(mtcars_matrix, scale = "column", Colv = NA, Rowv = NA,
col = heat.colors(256), xlab = "Zmienna")
```

Rezultat powyższego kodu przedstawia rysunek 6.35.



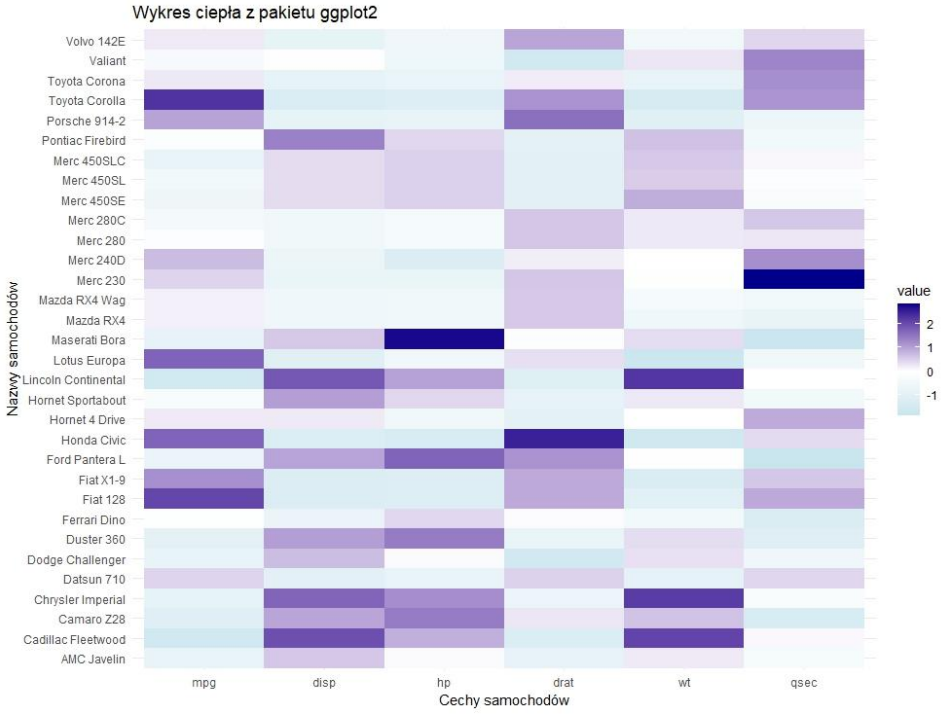
Rysunek 6.35. Wykres ciepła wybranych zmiennych z mtcars w zrealizowany pakiecie base

Źródło: opracowanie własne w programie R.

Rysunek 6.35 został wykonany z wykorzystaniem funkcji *heatmap*. W pakiecie **ggplot** także możliwa jest konstrukcja wykresu ciepła. W tym celu należy wykorzystać geometrię geom_tile jak w poniższym kodzie.

```
# Konstrukcja wykresu heatmap
data(mtcars)
mtcars_std <- mtcars %>%
  mutate_if(is.numeric, scale)
mtcars_std=mtcars_std[,c(1,3,4,5,6,7)]
mtcars_std$model=rownames(mtcars_std)
mtcars_melted <- melt(mtcars_std, id.vars = "model")
ggplot(mtcars_melted, aes(x = variable, y = model, fill =
value)) +
  geom_tile() +
  scale_fill_gradient2(low = "lightblue", high = "darkblue") +
  labs(title = "Wykres ciepła z pakietu ggplot2", x = "Cechy
samochodów", y = "Nazwy samochodów") +
  theme_minimal()
```

Wynik realizacji tego kodu przedstawia rysunek 6.36.



Rysunek 6.36. Wykres ciepła wybranych zmiennych z mtcars zrealizowany w pakiecie ggplot2

Źródło: opracowanie własne w programie R.

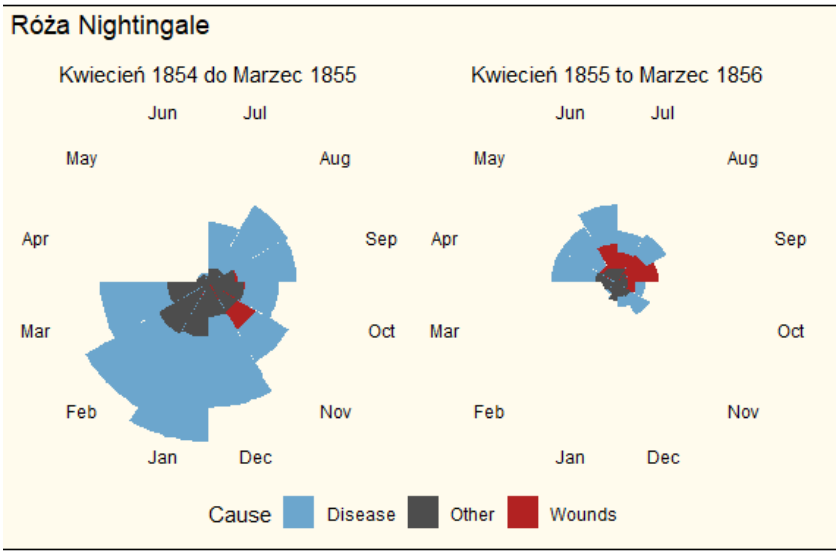
6.4.4. Wykres róża Nightingale

Wykres róża Nightingale należy do najczęściej przywoływanych historycznych prezentacji graficznych. Zwięzły opis dotyczący tego wykresu został przybliżony w punkcie 1.2.3. Poniżej przedstawiono kod pozwalający sporządzić taki wykres w formie zbliżonej do pierwowzoru postaci. W kodzie wykorzystano dane pochodzące z pakietu **`HistData`**.

```
# Konstrukcja wykresu róża Nightingale
Nightingale %>%
  select(Date, Month, Year, contains("rate")) %>%
  pivot_longer(cols = 4:6, names_to = "Cause", values_to =
"Rate") %>%
  mutate(Cause = gsub(".rate", "", Cause),
```

```
         period = ifelse(Date <= as.Date("1855-03-01"),
"Kwiecień 1854 do Marzec 1855", "Kwiecień 1855 to Marzec 1856"),
         Month = fct_relevel(Month, "Jul", "Aug", "Sep", "Oct",
"Nov", "Dec", "Jan", "Feb", "Mar", "Apr", "May", "Jun")) %>%
  ggplot(aes(Month, Rate)) +
  geom_col(aes(fill = Cause), width = 1, position = "identity")
+
  coord_polar() +
  facet_wrap(~period) +
  scale_fill_manual(values = c("skyblue3", "grey30",
"firebrick")) +
  scale_y_sqrt() +
  theme_void() +
  theme(axis.text.x = element_text(size = 9),
        strip.text = element_text(size = 11),
        legend.position = "bottom",
        plot.background = element_rect(fill = alpha("cornsilk",
0.5)),
        plot.margin = unit(c(10, 10, 10, 10), "pt"),
        plot.title = element_text(vjust = 5)) +
  ggtitle("Róża Nightingale")
```

Rezultat wykonania powyższego kodu został zamieszczony na rysunku 6.37.



Rysunek 6.37. Historyczny wykres róża Nightingale

Źródło: opracowanie własne w `ggplot2` na podstawie HistData (b.r.).

6.4.5. Wykres gwiazdowy

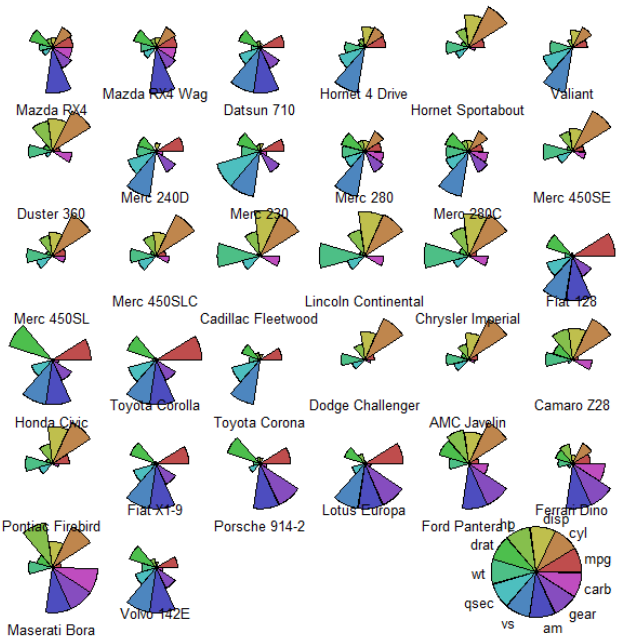
Wykresy gwiazdowe (star plot) mogą mieć bardzo różną postać. Wersja takich wykresów jest zaimplementowana w podstawowym zestawie pakietów instalowanych wraz z programem R. Konstrukcja dla zbioru **mtcars** wygląda następująco.

```
# Konstrukcja wykresu gwiazdowego
palette(rainbow(12, s = 0.6, v = 0.75))
stars(mtcars, draw.segments = TRUE, key.loc = c(13, 2.2)))
```

Wykres gwiazdkowy prezentujący wszystkie zmienne dla wszystkich obiektów (samochodów) zbioru **mtcars** przedstawia rysunek 6.38.

Przy dużej liczbie obiektów i zmiennych wykres może nie być dostatecznie czytelny. Tak właśnie jest na rysunku 6.38. Dobrym wyjściem jest prezentacja tylko wybranej grupy obiektów oraz wybranych zmiennych. Przykład kodu realizującego takie zadanie jest następujący.

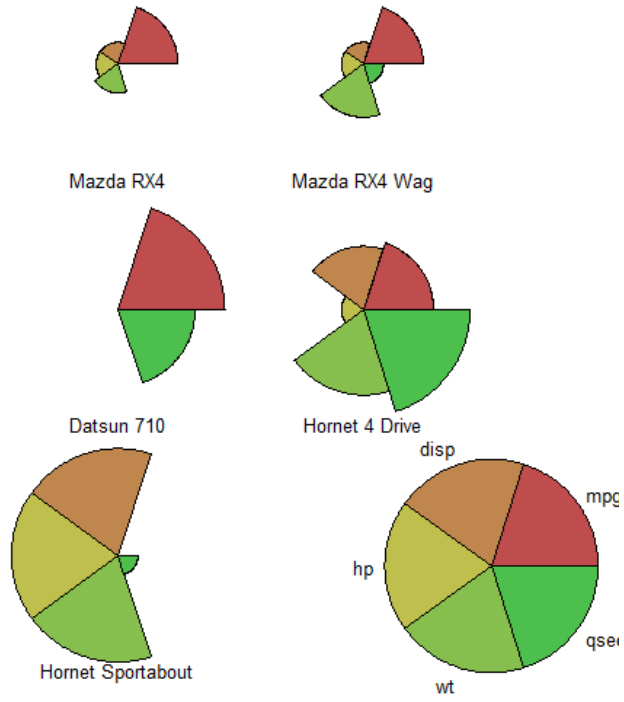
```
# Konstrukcja wykresu gwiazdowego
stars(mtcars[1:5,c(1,3,4,6,7)], draw.segments = TRUE, key.loc =
c(5.8, 2.2))
```



Rysunek 6.38. Wykres gwiazdowy. Charakterystyka wszystkich zmiennych dla wszystkich obserwacji zbioru mtcars

Źródło: opracowanie własne w programie R.

Na rysunku 6.39 przedstawiono charakterystykę czterech wybranych samochodów ze względu na pięć zmiennych (*mpg*, *disp*, *hp*, *wt*, *qsec*). Dodatkowo na wykresie została zamieszczona czytelna legenda.



Rysunek 6.39. Wykres gwiazdowy. Charakterystyka wybranych zmiennych wybranych obserwacji zbioru mtcars

Źródło: opracowanie własne w programie R.

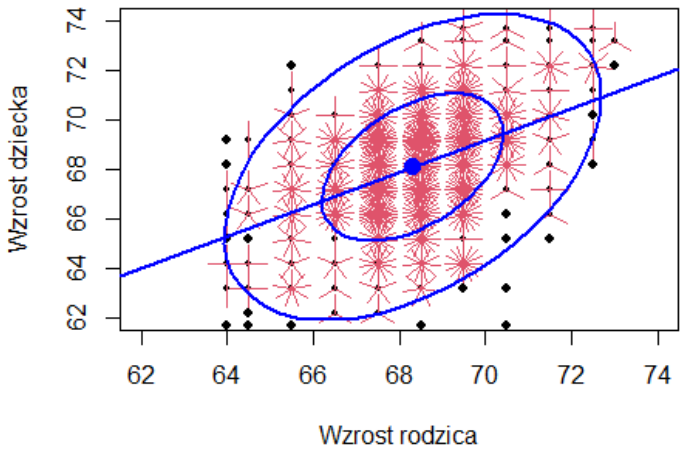
6.4.6. Wykres słonecznikowy

Francis Galton (lata 1822-1911) był brytyjskim naukowcem prowadzącym badania nad dziedzicznością cech fizycznych i psychicznych u ludzi. Jednym z jego najbardziej znanych eksperymentów było określenie postaci związku między wzrostem rodziców a wzrostem ich dzieci. W trakcie tego badania Galton zbadał kilkaset rodzin. Wynikiem tych prac było stworzenie koncepcji tak zwanej regresji do średniej. Galton odkrył, że dzieci, których rodzice mieli bardzo wysoki lub bardzo niski wzrost, miały tendencję do wykazywania średniego wzrostu bliższego do średniego wzrostu całej populacji niż u ich rodziców. Badania pokazały, że dzieci osób o ponadprzeciętnym wzroście zazwyczaj są niższe od swoich rodziców, podczas gdy dzieci osób o wzroście znacznie niższym od średniego zwykle są wyższe od swoich rodziców.

```
# Konstrukcja wykresu słonecznikowego
data(Galton)
with(Galton,
        {
        sunflowerplot(parent,child, xlim=c(62,74),
ylim=c(62,74),xlab='Wzrost rodzica',ylab='Wzrost dziecka')
        reg <- lm(child ~ parent)
        abline(reg, col='blue',lwd=2)
        dataEllipse(parent,child, plot.points=FALSE)
  })
```

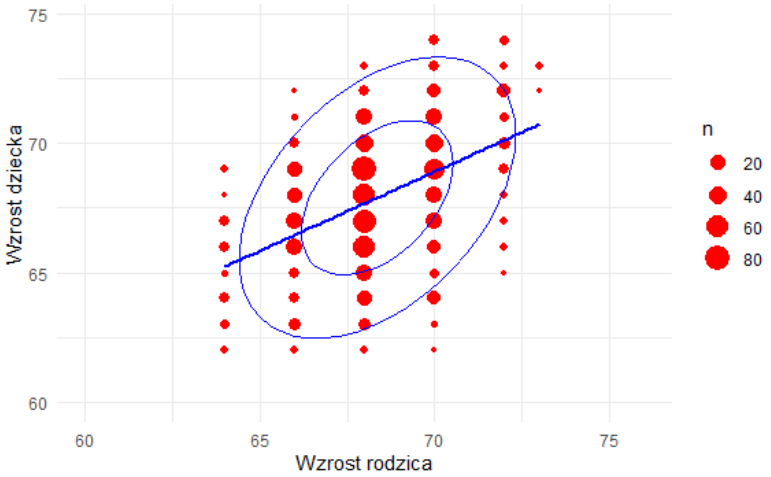
Na rysunku 6.40 przedstawiono wyniki pochodzące z badań Galtona na wykresie słonecznikowym. Podobny wykres, ale zrealizowany w **ggplot2** (rysunek 6.41), ma wyraźnie inną konstrukcję. Liczebności obserwacji w ustalonych punktach są reprezentowane przez wielkość kropki, a na wykresie słonecznikowym w klasycznej postaci były to „płatki" słonecznika.

```
# Konstrukcja wykresu słonecznikowego – wersja ggplot2
ggplot(Galton, aes(x = round(parent), y = round(child))) +
    geom_count(color='red') +
    stat_ellipse(type = "norm", level = 0.9,color='blue') +
    stat_ellipse(type = "norm", level = 0.5,color='blue') +
    geom_smooth(method = "lm", se = FALSE, color = "blue") +
    lims(x=c(60,76),y=c(60,76))+
    theme_minimal() +
    labs(x = "Wzrost rodzica",y = "Wzrost dziecka")
```



Rysunek 6.40. Wykres słonecznikowy. Wzrost rodzica i dziecka (w calach)

Źródło: opracowanie własne na podstawie HistData (b.r.).



Rysunek 6.41. Wykres słonecznikowy. Wzrost rodzica i dziecka (w calach) – opracowanie w ggplot2

Źródło: opracowanie własne na podstawie HistData (b.r.).

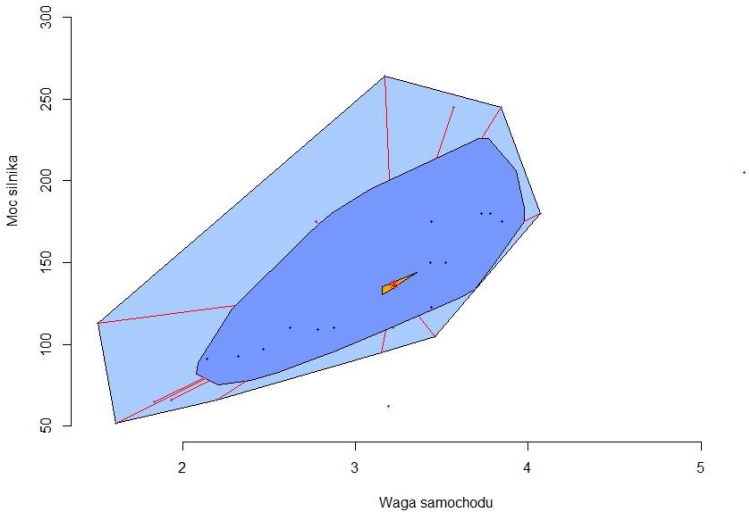
6.4.7. Wykres pudełkowy dwuwymiarowy

W punkcie 3.1.4 przedstawiono zwięzłą charakterystykę wykresu pudełkowego, a w punkcie 5.3.4 realizację takiego wykresu w pakiecie **ggplot2**. Wykresy pudełkowe standardowo są wykonywane dla jednej zmiennej liczbowej. Zostały jednak opracowane reprezentacje geometryczne, które w pewnej mierze rozszerzają tę konstrukcję na dwie zmienne liczbowe. Takie możliwości dają np. funkcje *bagplot* oraz *boxplot2D* z pakietu **aplpack**. Poniżej przedstawiono dwa kody dla wspomnianych funkcji.

```
# Konstrukcja wykresu pudełkowego 2D
bagplot(mtcars[,c(6,4)],factor=2.5,create.plot=TRUE,approx.limit
=300,xlab='Waga samochodu',ylab='Moc silnika')
```

Wynik realizacji kodu został przedstawiony na rysunku 6.42.

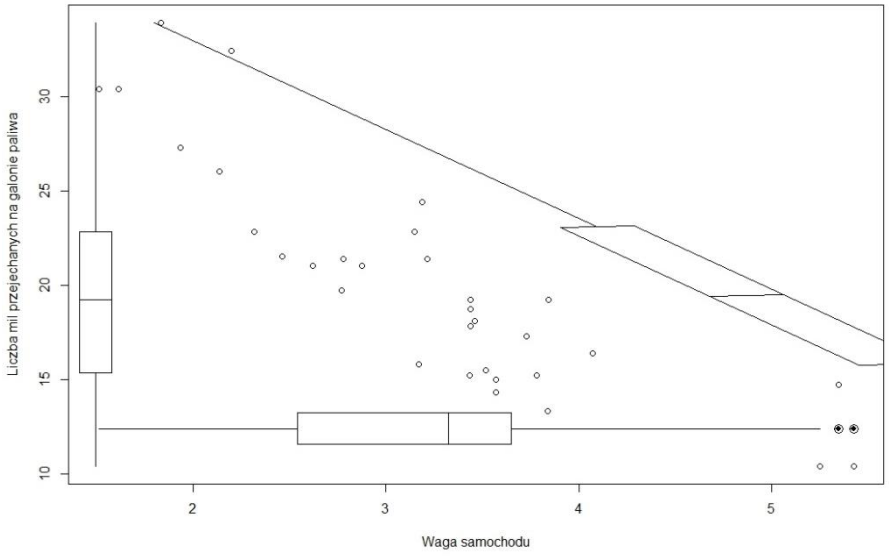
```
# Konstrukcja wykresu boxplot2D
mt=cbind(mtcars[,6],mtcars[,1])
plot(mt, xlab='Waga samochodu',ylab='Liczba mil przejechanych na
galonie paliwa')
boxplot2D(mt,box.shift=-40,angle=3,angle.typ=1)
boxplot2D(mt,box.shift=-110,angle=90,angle.typ=1)
boxplot2D(mt,box.shift=-40,angle=11.6,angle.typ=1)
```



Rysunek 6.42. Dwuwymiarowy wykres pudełkowy. Waga samochodu i moc silnika

Źródło: opracowanie własne w programie R.

Drugi ze wspomnianych kodów wykorzystuje funkcję *boxplot2D*, a odpowiedni rezultat jest zamieszczony na rysunku 6.43. Poza wykresami pudełkowymi dla rozkładów brzegowych możliwe jest wykreślenie wykresu pudełkowego dla kombinacji liniowej analizowanych zmiennych.



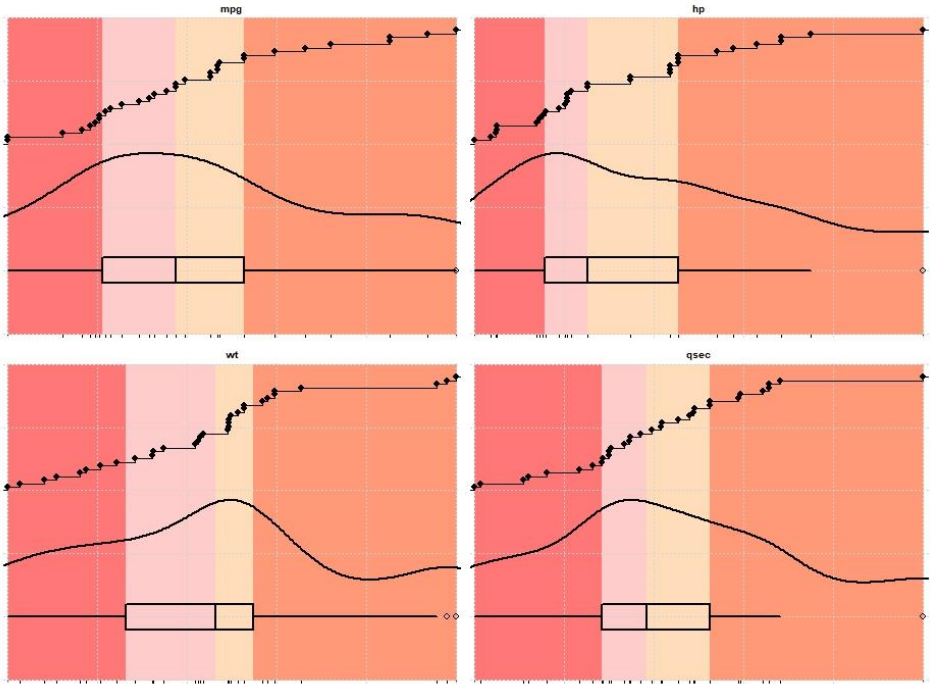
Rysunek 6.43. Boxplot 2D. Waga samochodu i liczba mil przejechanych na galonie paliwa

Źródło: opracowanie własne w programie R.

6.4.8. Wykres podsumowania zmiennych

W pakiecie **aplpack** dostępna jest także dość nietypowa funkcja pozwalająca na przedstawienie swoistego podsumowania graficznego wszystkich wskazanych zmiennych dla ustalonego zbioru danych. Kod w tym przypadku jest następujący.

```
# Konstrukcja wykresu podsumowania zmiennych
library(aplpack)
plotsummary(mtcars[,c(1,4,6,7)],types=c("ecdf", "density",
"boxplot"))
```



Rysunek 6.44. Podsumowanie zmiennych *mpg*, *hp*, *wt* oraz *qsec*

Źródło: opracowanie własne w programie R.

Na otrzymanym wykresie (por. rysunek 6.44) przedstawiona jest charakterystyka czterech wybranych zmiennych (*mpg*, *hp*, *wt* oraz *qsec*) ze zbioru **mtcars**. Graficznie przedstawiono na nim wykres empirycznej dystrybuanty, oszacowania funkcji gęstości oraz wykres pudełkowy.

Zakończenie

Niniejsza monografia została zaplanowana jako przewodnik po metodach graficznej prezentacji danych oraz praktycznych aspektach pracy z programem R, pakietem **ggplot2** i wybranymi rozszerzeniami tego pakietu. Celem monografii uczyniono przedstawienie podstawowych zasad dotyczących konstrukcji wykresów w praktyce badań naukowych, metod wykorzystywanych w wizualizacji danych oraz narzędzi pozwalających w profesjonalny sposób prezentować rezultaty badań naukowych.

W rozdziale pierwszym przytoczono wybrane historyczne metody wizualizacji danych, obejmując najwcześniejsze zastosowania graficznej prezentacji zdarzeń i zjawisk. Wskazano na znaczące kamienie milowe w wizualizacji danych od najdawniejszych czasów po ikoniczne prezentacje graficzne, takie jak marsz wojsk Napoleona na Moskwę, róża Nightingale czy mapa epidemii cholery autorstwa Johna Snowa. Ta część monografii pozwoliła poznać ewolucję technik reprezentacji wizualnej oraz ich kluczową rolę w zrozumieniu złożonych zjawisk i zbiorów danych.

Kolejny rozdział wprowadził podstawowe określenia związane z badaniami naukowymi, a w szczególności z dokonywanymi pomiarami, jak również podstawowe zasady konstrukcji wykresów. Zaprezentowano ideę gramatyki grafiki zaproponowaną przez Lelanda Wilkinsona (2005), którą później zaimplementowano w pakiecie **ggplot2**. Ta implementacja zapewniła naukowcom różnych dyscyplin solidne podstawy do tworzenia skutecznych wizualizacji. To niezbędna wiedza potrzebna do profesjonalnego i eleganckiego przedstawiania danych w formie graficznej.

W trzecim rozdziale skoncentrowano się na charakterystyce wybranych metod graficznych. Zaprezentowano różnorodne rodzaje wykresów wykorzystywane w wizualizacji danych, poczynając od bardzo często stosowanych histogramów, wykresów słupkowych czy rozrzutu po wykresy mapowe albo stosunkowo rzadkie wykresy mozaikowe. Zastosowania wykresów zostały omówione według ich typów, rodzaju analizy oraz skali pomiarowej, co powinno ułatwić Czytelnikowi zrozumienie praktycznych zastosowań. Wskazano na podstawowe przeznaczenie tych typów wykresów, kategoryzując je według rodzaju wykresu,

analizy oraz skali pomiarowej, oferując Czytelnikowi wgląd w wybór odpowiedniej techniki wizualizacji dla opisywanego zbioru danych.

Rozdział czwarty omawia tematykę wprowadzającą Czytelnika do pracy z programem R. Przedstawiono ogólną charakterystykę programu, zasady pracy z RStudio oraz podstawowe metody graficzne w programie R. Czytelnik może zdobyć tym samym umiejętności niezbędne do efektywnego korzystania z narzędzi programistycznych w analizie danych.

Znacząca część pracy jest poświęcona pakietowi **ggplot2** – ważnemu narzędziu do tworzenia złożonych i estetycznie przyjemnych wizualizacji w programie R. Pakiet ten jest poniekąd standardem w wizualizacji wyników badań naukowych w różnych dyscyplinach. Przedstawione zostały podstawy pracy z tym pakietem, przygotowanie zbioru danych do analizy graficznej oraz konstrukcja wybranych, a zarazem najczęściej wykorzystywanych w praktyce prezentowania wyników badań typów wykresów. Zamieszczone przykłady z wykresami z wykorzystaniem pakietu **ggplot2** pomogą Czytelnikowi zrozumieć zagadnienia związane z konstrukcją wykresów zgodnie z zasadami gramatyki grafiki.

W ostatnim rozdziale zaprezentowano wybrane biblioteki rozszerzające możliwości pakietu **ggplot2**. Przedstawiono charakterystykę tych pakietów, wskazując przykłady ich zastosowania w analizie zależności i opisie zjawisk wielowymiarowych.

Monografia może być źródłem wiedzy dla badaczy, analityków danych i każdego, kto interesuje się sztuką i nauką wizualizacji danych. Nie tylko obejmuje historyczny kontekst i aspekty teoretyczne reprezentacji wizualnej, ale również zapewnia praktyczne instrukcje i przykłady tworzenia skutecznych wizualizacji za pomocą R oraz pakietu **ggplot2**.

Bibliografia

Aigner W., Miksch S., Schumann H., Tominski C. (2011). *Visualization of Time-Oriented Data*. Springer. London. DOI: 10.1007/978-0-85729-079-3.

Albert J., Rizzo M. (2012). *R by Example*. Springer Science+Business Media. New York.

Aldrich J.O., Rodriguez H.M. (2013). *Building SPSS Graphs to Understand Data*. SAGE Publications.

Biecek P. (2014). *Odkrywać! Ujawniać! Objaśniać! Zbiór esejów o sztuce prezentowania danych*. Fundacja Naukowa SmarterPoland.pl. Warszawa.

Biecek P., Baranowska E., Sobczyk P. (2019). *Wykresy unplugged*. Fundacja Naukowa SmarterPoland.pl. Warszawa.

Chang W. (2019). *R Graphics Cookbook. Practical Recipes for Visualizing Data*. O'Reilly. Sebastopol.

Chen C., Härdle W., Unwin A. (2008). *Handbook of Data Visualization*, Springer-Verlag. Berlin.

Cirillo A. (2016). *RStudio for R. Statistical Computing Cookbook*. Packt Publishing. Livery Place.

Cleveland W.S. (1993). *Visualizing Data*. Hobart Press.

Cleveland W.S., McGill R. (1987). *Graphical Perception: The Visual Decoding of Quantitative Information on Graphical Displays of Data*. „Journal of the Royal Statistical Society. Series A (General)", nr 150(3), s. 192-229. DOI: 10.2307/2981473.

Czempas J. (2000). *Elementy statystyki. Podstawowe mierniki i metody*. Wyższa Szkoła Biznesu, Triada. Dąbrowa Górnicza.

Domański C., Pruska K., Wagner W. (1998). *Wnioskowanie statystyczne przy nieklasycznych założeniach*. Uniwersytet Łódzki, Łódź.

Fisher D., Meyer M. (2018). *Making Data Visual. A Practical Guide to Using Visualization for Insight*. O'Reilly. Sebastopol.

Fisher R.A. (1925). *Statistical Methods for Research Workers*. Oliver and Boyd. Edinburgh.

Friendly M. (1994). *Mosaic Displays for Multi-Way Contingency Tables*. „Journal of the American Statistical Association", nr 89, s. 190-200.

Friendly M. (2000). *Visualizing Categorical Data*. SAS Institute. Cary.

Friendly M. (2005). *Milestones in the History of Data Visualization: A Case Study in Statistical Historiography*. W: C. Weihs, W. Gaul (red.). *Classification: The Ubiquitous Challenge*. Springer. New York, s. 34-52.

Healy K. (2019). *Data Visualization. A Practical Introduction*. Princeton University Press.

Henderson H.V., Velleman P.F. (1981). *Building Multiple Regression Models Interactively*. „Biometrics", nr 37, s. 391-411.

Hilfiger J.J. (2016). *Graphing Data with R*. O'Reilly. Sebastopol.

Ihaka R., Gentleman R. (1996). *R: A Language for Data Analysis and Graphics*. „Journal of Computational and Graphical Statistics", nr 5(3), s. 299-314. DOI: 10.2307/1390807.

Inselberg A. (1999). *Don't Panic… Do It in Parallel*. „Computational Statistics", nr 14(1), s. 53-77.

Jeleński S. (1974). *Śladami Pitagorasa. Rozrywki matematyczne*. Wydawnictwa Szkolne i Pedagogiczne. Warszawa.

Kabacoff R.I. (2015). *R in Action. Data Analysis and Graphics with R*. Manning Publications. New York.

Kassambara A. (2013). *ggplot2: Guide to Create Beautiful Graphics in R*. STHDA.

Kirk A. (2019). *Data Visualisation: A Handbook for Data Driven Design*. SAGE.

Knaflic C.N. (2015). *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley.

Kocimowski K., Kwiatek J. (red.) (1976). *Wykresy i mapy statystyczne*. Główny Urząd Statystyczny. Warszawa.

Kończak G. (2012). *Wprowadzenie do symulacji komputerowych*. Uniwersytet Ekonomiczny. Katowice.

Kończak G. (2014). *Rola graficznych prezentacji danych w popularyzacji statystyki*. „Wiadomości Statystyczne", nr 7(LIX), s. 49-61.

Kończak G. (2016). *Testy permutacyjne. Teoria i zastosowania*. Uniwersytet Ekonomiczny. Katowice.

Kończak G. (2020). *Nieklasyczne metody statystyczne w badaniach ekonomicznych*. Uniwersytet Ekonomiczny. Katowice.

Kończak G., Kosińska M. (2023). *O testowaniu istotności różnic w strukturach populacji na podstawie prób o małych liczebnościach*. „Zeszyty Naukowe Uniwersytetu Ekonomicznego w Krakowie", nr 3(1001), s. 145-160. DOI: 10.15678/ZNUEK. 2023.1001.0308.

Kończak G., Żądło T. (2010). *Potrzeby przedsiębiorstw w zakresie analiz statystycznych i możliwości ich realizacji z wykorzystaniem arkusza kalkulacyjnego Excel*. W: M. Pilch (red.). *Nauczanie przedmiotów ilościowych a potrzeby rynku pracy*. Uniwersytet Łódzki. Łódź, s. 149-159.

Kopczewska K., Kopczewski T., Wójcik P. (2009). *Metody ilościowe w R. Aplikacje ekonomiczne i finansowe*. CeDeWu, Wydawnictwa Fachowe. Warszawa.

Long J.D., Teetor P. (2019). *R Cookbook. Proven Recipes for Data Analysis, Statistics and Graphics*. O'Reilly. Sebastopol.

Moulik T. (2018). *Applied Data Visualization with R and ggplot2*. Packt Publishing. Birmingham.

Pimpler E. (2017). *Data Visualization and Exploration with R. A Practical Guide to Using R, RStudio, and Tidyverse for Data Visualization, Exploration, and Data Science Applications*. GeoSpatial Training Services. Boerne.

Playfair W. (2005). *Playfair's Commercial and Political Atlas and Statistical Breviary*. Cambridge. London.

Rahlf T. (2017). *Data Visualization with R. 100 Examples*. Springer International Publishing. Cham.

Rao C.R. (1994). *Statystyka i prawda*. WN PWN. Warszawa.

Reichmann W.J. (1968). *Drogi i bezdroża statystyki*. PWN. Warszawa.

Rendgen S. (2018). *The Minard System. The Complete Statistical Graphics of Charles--Joseph Minard*. Princetown Architectural Press. New York.

Sobczyk M. (2001). *Statystyka*. WN PWN. Warszawa.

Sosulski K. (2019). *Data Visualization Made Simple: Insights into Becoming Visual*. Routledge, Taylor & Francis Group.

Steele J., Iliinsky N.P.N. (red.) (2010). *Beautiful Visualization: Looking at Data through the Eyes of Experts*. O'Reilly.

Toit S.H.C., Steyn A.G.W., Stumpf R.H. (1986). *Graphical Exploratory Data Analysis*. Springer International Publishing. New York.

Tufte E.R. (1983). *The Visual Display of Quantitative Information*. Graphics Press. Cheshire.

Tufte E.R. (2006). *Beautiful Evidence*. Graphics Press.

Tufte E.R. (2019). *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press.

Tufte E.R. (red.) (2013). *Envisioning Information*. Graphics Press.

Tukey J.W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

Unwin A. (2015). *Graphical Data Analysis with R*. A CRC Press Taylor & Francis Group. Boca Raton.

Unwin A., Theus M., Hofmann H. (2006). *Graphics of Large Datasets. Visualizing a Million*. Springer Science+Business Media. New York.

Walesiak M., Gatnar E. (red.) (2009). *Statystyczna analiza danych z wykorzystaniem programu R*. WN PWN. Warszawa.

Wawrzynek J. (2007). *Metody opisu i wnioskowania statystycznego*. Akademia Ekonomiczna. Wrocław.

Wickham H. (2009). *ggplot2. Elegant Graphics for Data Analysis*. Springer Science+ Business Media. New York.

Wilke C.O. (2019). *Fundamentals of Data Visualization. A Primer on Making Informative and Compelling Figures*. O'Reilly. Sebastopol.

Wilkinson L. (2005). *The Grammar of Graphics*. Springer Science+Business Media. New York.

Wills G. (2012). *Visualizing Time*. Springer. New York. DOI: 10.1007/978-0-387-77907-2.

Zelazny G. (2005). *Say It with Charts Workbook*. McGraw-Hill.

Żądło T., Kończak G. (2009). *Analizy statystyczne i graficzna prezentacja danych wykorzystaniem programu R w nauczaniu statystyki*. „Rola Informatyki w Naukach Ekonomicznych i Społecznych. Innowacje i Implikacje Dyscyplinarne", nr 2(2), s. 353-361.

Netografia

Alboukadel. *Top R Color Palettes to Know for Great Data Visualization*. Datanovia, https://www.datanovia.com/en/blog/top-r-color-palettes-to-know-for-great-data-visualization/ (dostęp: 12.02.2024).

ArcGIS Pro. *Types of Tabular Charts*, https://pro.arcgis.com/en/pro-app/latest/help/analysis/geoprocessing/charts/types-of-charts.htm (dostęp: 29.07.2023).

Ataman. *The Oldest Map of the World*, http://www.atamanhotel.com/catalhoyuk/oldest-map.html (dostęp: 22.03.2024).

BooKey. *30 Best Edward Tufte Quotes with Image*, https://www.bookey.app/quote-author/edward-tufte (dostęp: 24.06.2024).

Cartographia. *Carte Figurative*, http://cartographia.files.wordpress.com/2008/05/minard_napoleon.png (dostęp: 17.02.2024).

CRAN. *The R Project for Statistical Computing*, http://www.r-project.org (dostęp: 2.04.2024).

CRAN R Project. *aplpack*, https://cran.r-project.org/web/packages/aplpack/index.html (dostęp: 12.02.2024).

CRAN R Project. *Package 'wesanderson'*, https://cran.r-project.org/web/packages/wesanderson/wesanderson.pdf (dostęp: 5.05.2023).

CRAN R Project. *RColorBrewer*, https://cran.rproject.org/web/packages/RColorBrewer/index.html (dostęp: 5.05.2023).

DataVis. *Halley's Wind Map, Section 1 Detail*, https://www.datavis.ca/milestones/index.php?group=1600s#lightbox-gallery-49-2 (dostęp: 15.07.2023).

DataVis. *Milestones Project*, https://www.datavis.ca (dostęp: 22.07.2023).

DataVis. *The Causes of the Mortality in the Army in the East*, http://datavis.ca/milestones//admin/uploads/images/coxcomb3.jpg (dostęp: 17.02.2024).

DataVis. *The Oldest Map*, http://datavis.ca/milestones/uploads/images/oldest-map.jpg (dostęp: 22.07.2023).

DataVis. *Weather Map*, https://www.datavis.ca/milestones/index.php?group=1600s&mid=ms49 (dostęp: 22.03.2024).

DataVis. *World Map*, https://datavis.ca/milestones/uploads/images/worldmap.gif (dostęp: 17.02.2024).

DataViz Project, https://datavizproject.com (dostęp: 22.07.2023).

Erik Gahner. *Awesome ggplot2*, https://github.com/erikgahner/awesome-ggplot2 (dostęp: 27.07.2023).

FlowingData. *Statistical Visualization*, https://flowingdata.com/category/visualization/statistical-visualization/ (dostęp: 27.07.2023).

Friendly M. (2024). *The Graphic Works of Charles Joseph Minard*. DataVis, http://www.datavis.ca/gallery/minbib.php (dostęp: 22.03.2024).

Friendly M., Denis D.J. *Milestones in the History of Thematic Cartography, Statistical Graphics, and Data Visualization*. DataVis, http://datavis.ca/milestones/ (dostęp: 22.03.2024).

Friendly M., Wainer H. *Chapter 5. The Big Bang: William Playfair, the Father of Modern Graphics*. History of Data Visualization, https://friendly.github.io/HistDataVis/ch05-playfair.html (dostęp: 15.07.2023).

Gapminder, https://www.gapminder.org/ (dostęp: 12.02.2024).

ggplot2Extensions, https://exts.ggplot2.tidyverse.org/gallery/ (dostęp: 27.07.2023).

HistData, https://cran.r-project.org/web/packages/HistData/HistData.pdf (dostęp: 18.06.2024).

History of Information. *Diagram of the Causes of Mortality in the Army in the East*, https://www.historyofinformation.com/image.php?id=851 (dostęp: 15.07.2023).

Inspiring Quotes. *Ben Shneiderman Quotes*, https://www.inspiringquotes.us/author/8937-ben-shneiderman (dostęp: 24.06.2024).

JPowered. *History of Bar Charts and Graphs*, https://jpowered.com/graphs-and-charts/bar-chart-history.html (dostęp: 7.02.2024).

Kończak (2024). *Wizualizacja wyników badań naukowych*, http://stat.ue.katowice.pl/wwbn (dostęp: 6.06.2024).

Marlena (2009). *What is Data Visualization: Part 1 of 2 Characteristics of Excellent Visualizations*. Marlena's Blog, http://marlenacompton.com/?p=103 (dostęp: 22.03.2024).

Martin Grandjean. *Carte Figurative*, https://www.martingrandjean.ch/wp-content/uploads/2014/05/Minard1Vector.png (dostęp: 15.07.2023).

Planetary Movements Map, http://www.fi.uu.nl/wiskrant/artikelen/hist_grafieken/begin/images/planeten.gif (dostęp: 15.07.2023).

PMassicotte. *The Paletteer Gallery*, https://pmassicotte.github.io/paletteer_gallery/ (dostęp: 16.02.2024).

Portal Statystyczny, https://portalstatystyczny.pl/ (dostęp: 22.07.2023).

R-charts, https://r-charts.com/ (dostęp: 22.07.2023).

RStudio, https://www.rstudio.com/ (dostęp: 17.02.2024).

Wikimedia. *Carte Figurative*, https://upload.wikimedia.org/wikipedia/commons/2/29/Minard.png (dostęp: 15.07.2024).

Wikimedia. *Visual Train Schedule*, https://upload.wikimedia.org/wikipedia/commons/d/da/Ibry%27s_Visual_Train_Schedule.png (dostęp: 15.02.2024).

Wikipedia. *Heliocentryzm*, https://pl.wikipedia.org/wiki/Heliocentryzm (dostęp: 17.02.2024).

Wikipedia. *Hieroglify*, https://pl.wikipedia.org/wiki/Egipskie_hieroglify_%E2%80%93_okre%C5%9Blniki_i_ideogramy (dostęp: 22.03.2024).

Wikipedia. *Malarstwo jaskiniowe*, https://pl.wikipedia.org/wiki/Malarstwo_jaskiniowe (dostęp: 22.03.2024).

Wikipedia. *Mapa Ptomeleusza*, https://pl.wikipedia.org/wiki/Mapa_Ptolemeusza#/media/Plik:Ptolemy_Asia_detail.jpg (dostęp: 15.07.2023).

Wikipedia. *Ptolemy's World Map*, https://en.wikipedia.org/wiki/Ptolemy%27s_world_map (dostęp: 24.06.2024).

Wikipedia. *Snow Cholera Map*, https://upload.wikimedia.org/wikipedia/commons/2/27/Snow-cholera-map-1.jpg (dostęp: 15.07.2023).

Spis rysunków

Spis tabel

Celem monografii jest przedstawienie zasad konstrukcji prezentacji graficznych, metod wizualizacji danych oraz kluczowych narzędzi wykorzystywanych w takich prezentacjach. Realizacja tego celu wymaga wprowadzenia pewnej systematyki dla metod graficznych, w szczególności powiązania doboru odpowiedniego wykresu z rodzajem i strukturą danych, a konkretniej ze skalą pomiarową analizowanych zmiennych. Wszystko to może być pomocne dla naukowców prowadzących badania w różnych dyscyplinach, ponieważ prezentowane metody i narzędzia związane z wizualizacją danych są uniwersalne. Ważnym założeniem poczynionych rozważań stało się dążenie do wypracowania u Czytelnika umiejętności stawiania pytań badawczych na podstawie przeprowadzonej wstępnej, graficznej analizy danych. Zamiar ten można w znacznej mierze zrealizować poprzez przedstawienie odpowiednich przykładów. Takie przykłady, wykorzystujące dostępne w programie R zbiory danych, zostały zamieszczone w ostatnich rozdziałach książki. Metody wizualizacji danych odgrywają coraz większą rolę także w dydaktyce i popularyzacji wiedzy z różnych dyscyplin.

Prof. dr hab. Grzegorz Kończak jest pracownikiem Katedry Statystyki, Ekonometrii i Matematyki na Wydziale Zarządzania Uniwersytetu Ekonomicznego w Katowicach. Obszar jego zainteresowań naukowych pozostaje związany z wykorzystaniem metod symulacji komputerowej w badaniach ekonomicznych, metodami graficznej prezentacji i analizy danych, a także z zagadnieniami statystycznej kontroli jakości. Jest autorem lub współautorem ponad 100 publikacji naukowych, 13 książek oraz kilkudziesięciu prac niepublikowanych. Uczestniczył w ponad 80 konferencjach i seminariach naukowych krajowych i zagranicznych.

ISBN 978-83-7875-901-0

Uniwersytet Ekonomiczny w Katowicach