Laboratorium 4

Wstęp do Analizy Danych | Politechnika Krakowska

Jakub Kapała

Numer albumu: 151885

Data: 17.04.2025

Zadanie 1 - Warstwy wykresów ggplot2

Treść

1.1 Z biblioteki MASS otworzyć zestaw danych `mammals`. Przedstawić na wykresie z pakietu `ggplot2` zależność masy mózgu zawartych w tym zestawieniu ssaków od ich masy ciała

Potrzebne funkcje: `ggplot(, aes())`, `geom_point()`.

1.2 Jak właśnie przekonaliśmy się, liniowe skale na osiach układu współrzędnych słabo nadają się do przedstawienia tych danych. Użyć osi o skali logarytmicznej, tzn. przedstawić dane na wykresie typu log-log. Użyć na osiach znaczników będących potęgami 10.

Potrzebne funkcje: `scale_x_continuous(name, transform, breaks, labels)`, `scale_y_continuous()`.

Przydatne materiały: https://ggplot2.tidyverse.org/reference/scale_continuous.html

1.3 Dodać do powyzszego wykresu warstwę z linią trendu za pomocą funkcji `geom_smooth()`

1.4 Dodać do punktów na wykresie etykiety opisujące do jakiego gatunku zwierzęcia odnosi się dany punkt. Wykres staje się nieczytelny - wybrać zatem losową próbkę 10 zwierząt i jedynie dla nich nanieść na wykres te etykiety.

Potrzebne funkcje: `geom_text(, label)`, `rownames()`, `sample()`, `nrow()`

Przydatne materiały:

https://ggplot2.tidyverse.org/reference/geom_text.html

https://stackoverflow.com/questions/62524965/how-to-label-only-certain-points-in-ggplot2

Rozwiązanie

1.1. Utworzenie wykresu zależności masy mózgu od masy ciała.

Instaluje pakiet `ggplot2` poprzez instalację pakietu `tidyverse` w całości, bowiem będzie on potrzebny w zadaniu 4.3.:

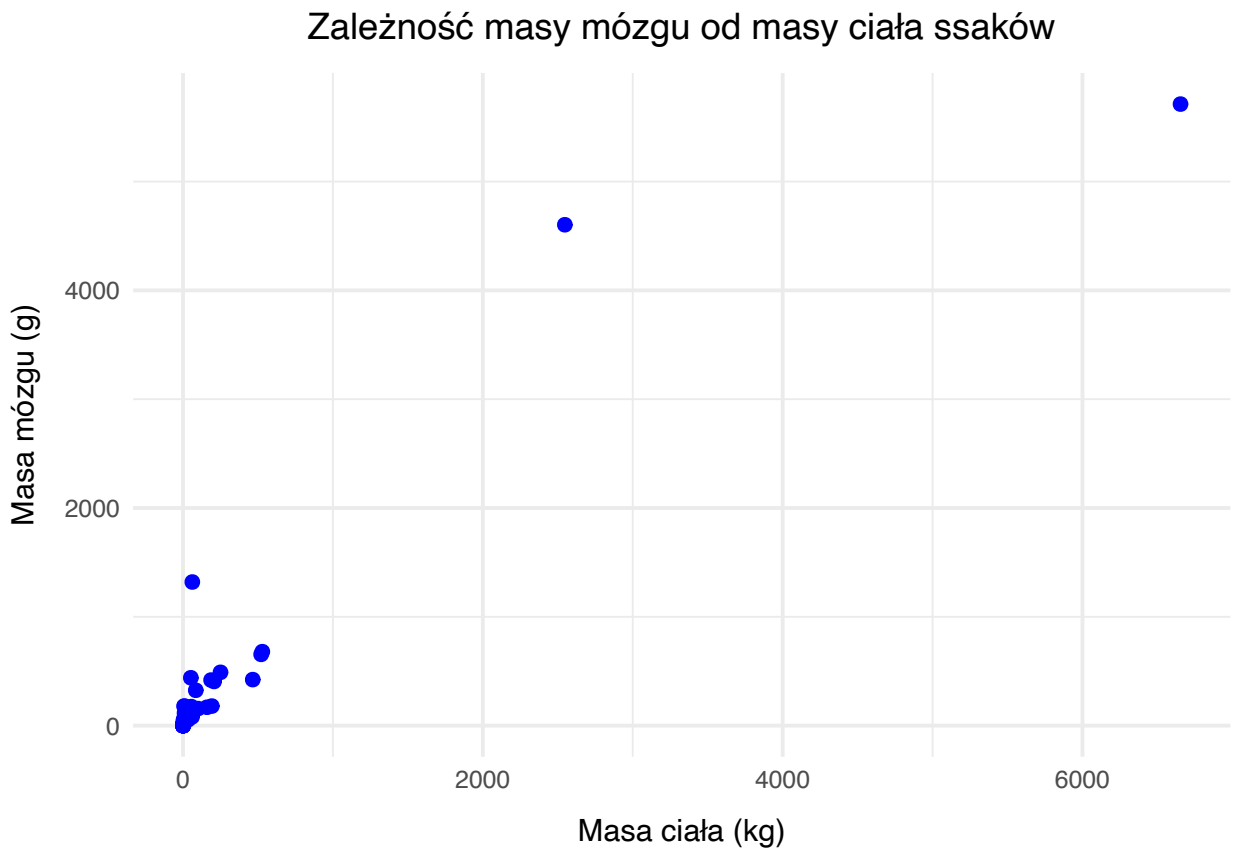
```
install.packages("tidyverse")
```

Wczytanie danych:

```r
library(MASS)
library(ggplot2)
data(mammals, package = "MASS")
```

Utworzenie wykresu:

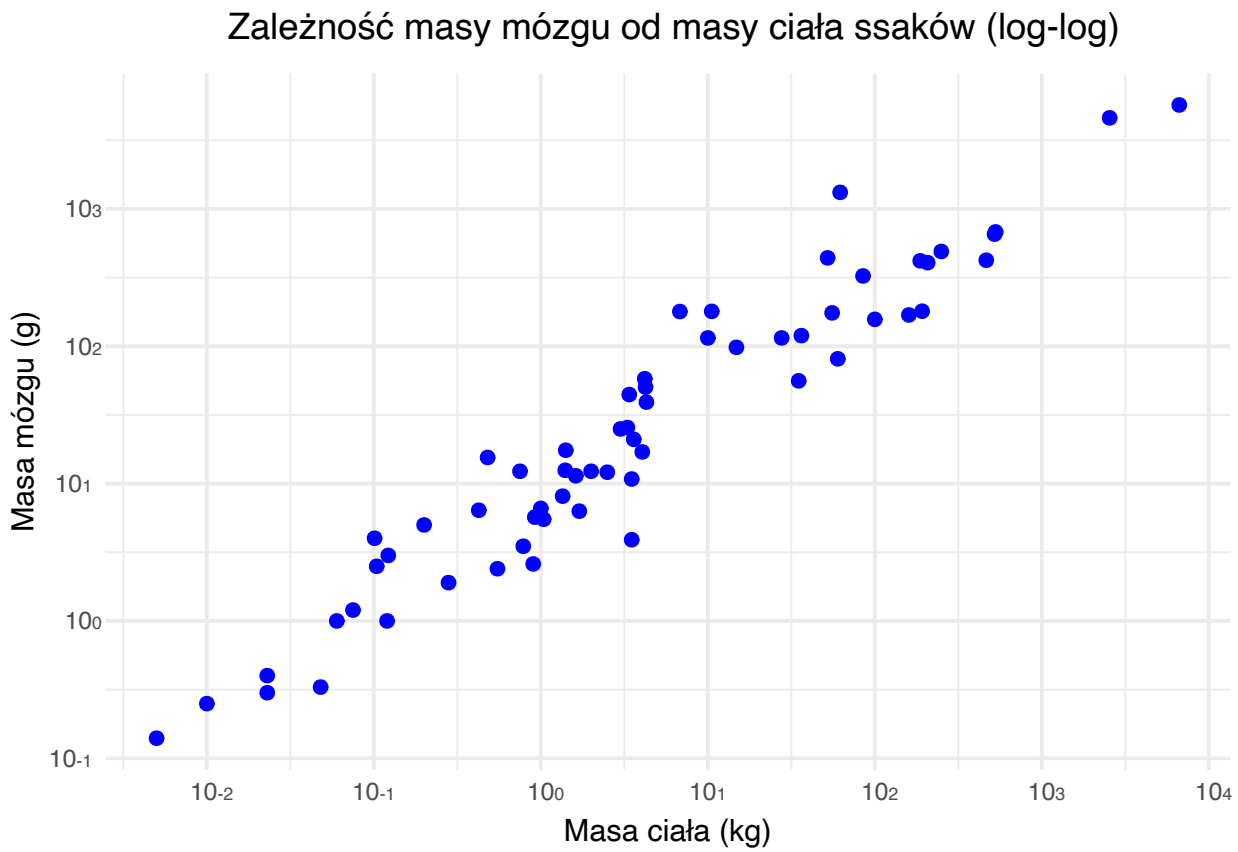
```r
ggplot(mammals, aes(x = body, y = brain)) +
  geom_point(color = "blue") +
  labs(
    title = "Zależność masy mózgu od masy ciała ssaków",
    x = "Masa ciała (kg)",
    y = "Masa mózgu (g)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, margin = margin(b = 10)),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10))
  )
```


Zależność masy mózgu od masy ciała ssaków
Masa mózgu (g)
4000
2000
0
0
2000
4000
6000
Masa ciała (kg)

1.2. Wykres typu log-log

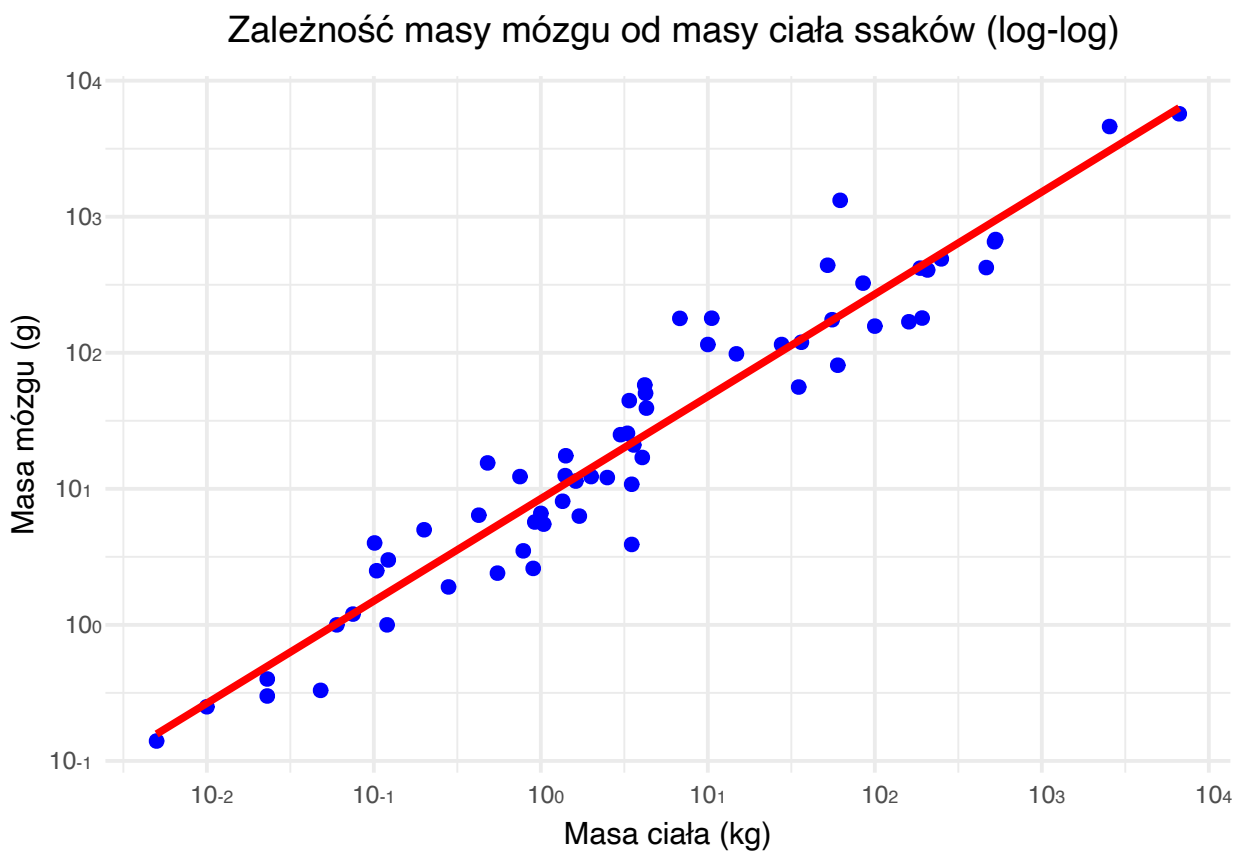
Tworzenie wykresu przy pomocy funkcji `scale_x_continuous()` i `scale_y_continuous()`:

```
ggplot(mammals, aes(x = body, y = brain)) +
  geom_point(color = "blue") +
  scale_x_continuous(
    name = "Masa ciała (kg)",
    trans = "log10",
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::trans_format("log10", scales::math_format(10^.x))
  ) +
  scale_y_continuous(
    name = "Masa mózgu (g)",
    trans = "log10",
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::trans_format("log10", scales::math_format(10^.x))
  ) +
  labs(
    title = "Zależność masy mózgu od masy ciała ssaków (log-log)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, margin = margin(b = 10)))
```



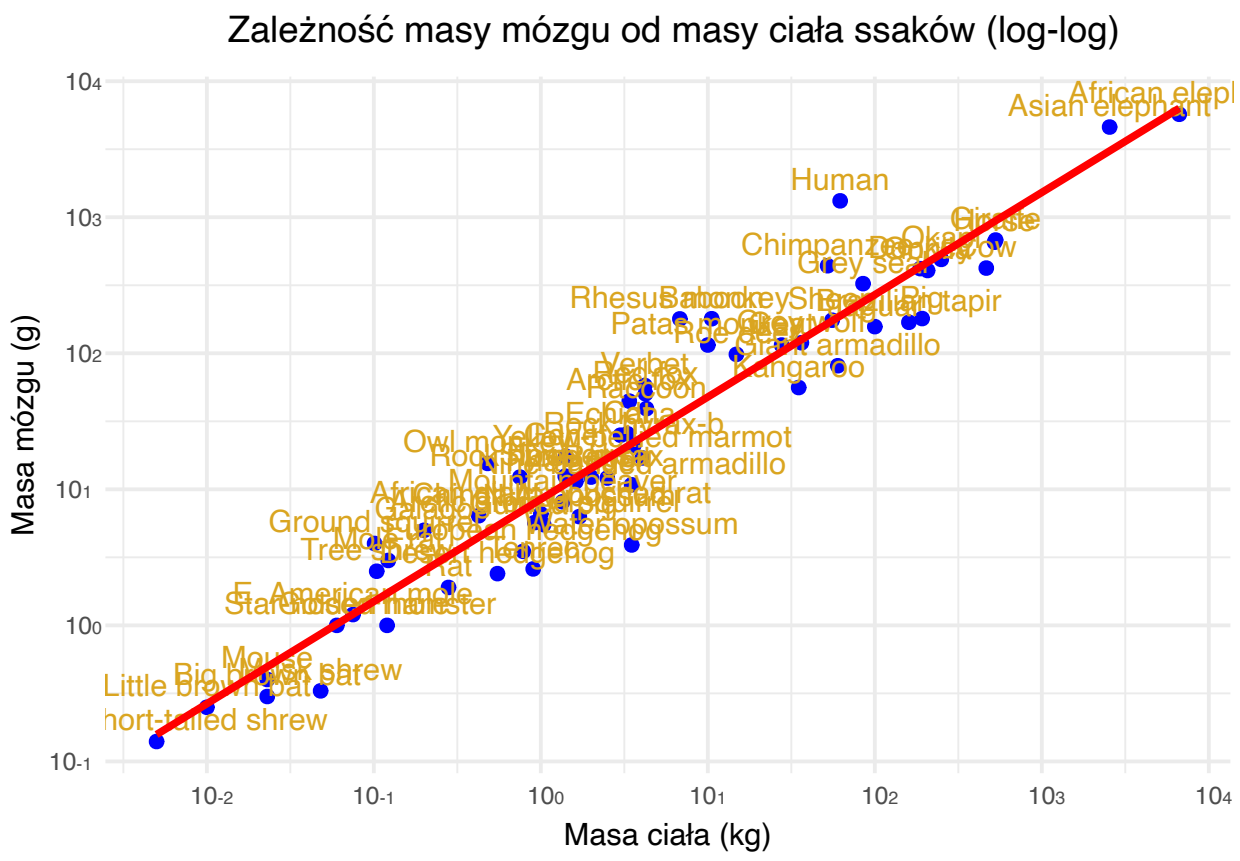
1.3. Dodanie linii trendu

```
ggplot(mammals, aes(x = body, y = brain)) +
  geom_point(color = "blue") +
  scale_x_continuous(
    name = "Masa ciała (kg)",
    trans = "log10",
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::trans_format("log10", scales::math_format(10^.x))
  ) +
  scale_y_continuous(
    name = "Masa mózgu (g)",
    trans = "log10",
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::trans_format("log10", scales::math_format(10^.x))
  ) +
  geom_smooth(method = "lm", formula = y ~ x, color = "red", se = FALSE) +
  labs(
    title = "Zależność masy mózgu od masy ciała ssaków (log-log)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, margin = margin(b = 10)))
```



1.4. Dodanie etykiet z gatunkami zwierząt + losowa próbka 10 zwierząt

```
ggplot(mammals, aes(x = body, y = brain)) +
  geom_point(color = "blue") +
  geom_text(
    aes(label = rownames(mammals)),
    vjust = -0.5,
    color = "goldenrod"
  ) +
  scale_x_continuous(
    name = "Masa ciała (kg)",
    trans = "log10",
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::trans_format("log10", scales::math_format(10^.x))
  ) +
  scale_y_continuous(
    name = "Masa mózgu (g)",
    trans = "log10",
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::trans_format("log10", scales::math_format(10^.x))
  ) +
  geom_smooth(method = "lm", formula = y ~ x, color = "red", se = FALSE) +
  labs(
    title = "Zależność masy mózgu od masy ciała ssaków (log-log)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, margin = margin(b = 10)))
```



Jak widać, wykres stał się nieczytelny, ponieważ jest na nim za dużo etykiet. Wylosujmy więc próbkę 10 zwierząt, dla których naniesiemy etykiety.

Najpierw ustawiam `seed` na swój numer albumu w celu zachowania powtarzalności wyników:

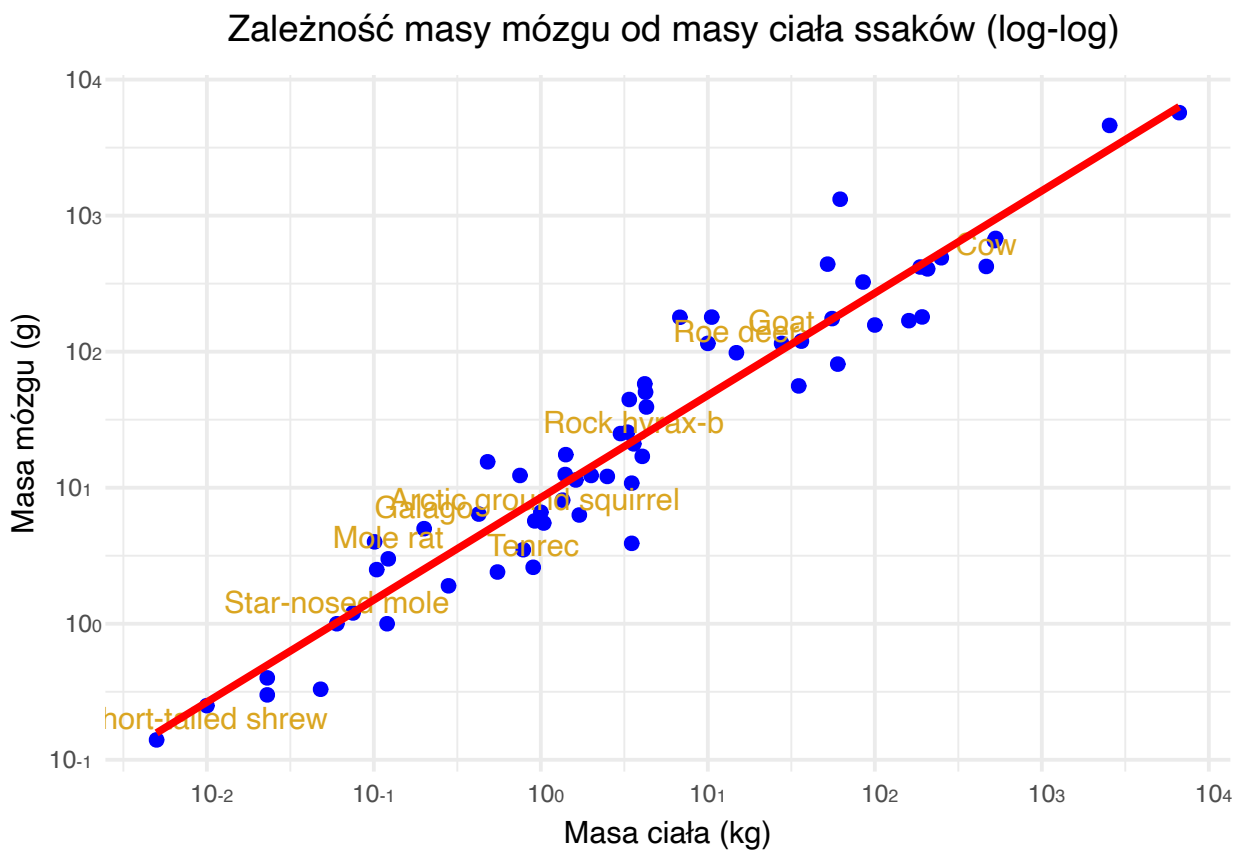
```
set.seed(151885)
```

Następnie losuję próbkę 10 zwierząt i tworzę nowy data frame z tymi zwierzętami:

```
sampled_indices <- sample(1:nrow(mammals), 10)
sampled_mammals <- mammals[sampled_indices, ]
```

Wykorzystując powyższą próbkę, nanosze etykiety na wykres:

```
ggplot(mammals, aes(x = body, y = brain)) +
  geom_point(color = "blue") +
  geom_text(
    data = sampled_mammals,
    aes(label = rownames(sampled_mammals)),
    vjust = -0.5,
    color = "goldenrod"
  ) +
  scale_x_continuous(
    name = "Masa ciała (kg)",
    trans = "log10",
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::trans_format("log10", scales::math_format(10^.x))
  ) +
  scale_y_continuous(
    name = "Masa mózgu (g)",
    trans = "log10",
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::trans_format("log10", scales::math_format(10^.x))
  ) +
  geom_smooth(method = "lm", formula = y ~ x, color = "red", se = FALSE) +
  labs(
    title = "Zależność masy mózgu od masy ciała ssaków (log-log)"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, margin = margin(b = 10)))
```



Zadanie 2 - Statystyki i geometrie w ggplot2

Treść

2.1 Z biblioteki MASS otworzyć zestaw danych `survey`. Przedstawić na wykresie z pakietu ggplot2 zależność rozpiętości dłoni używanej do pisania od rozpiętości drugiej dłoni dla studentów badanych w tej ankiecie.

a) Użyć wykresu punktowego.

b) Użyć wykresu z linią trendu.

Potrzebne funkcje: `ggplot(, aes(color, shape))`.

2.2 Przypomnieć sobie, jak na jednych z minionych zajęć konstruowaliśmy wykresy pudełkowe (boxplot) dla wzrostu mężczyzn i kobiet z University of Adelaide. Niestety na ten wykres nie można było nanieść punktów odnoszących się do poszczególnych osób. Osięgnąć ten cel dzięki pakietowi `ggplot2`. Wykorzystać funkcje `stat_boxplot()` i `geom_jitter()`.

2.3 Przedstawić wykres słupkowy liczby studentów niepalących/palących okazjonalnie/regularnie/nałogowo. Słupki są porządkowane przez R alfabetycznie – uporządkować je w powyższej (logicznej) kolejności.

Potrzebne funkcje: `geom_bar()`, `factor()`.

Przydatne materiały:

https://guslipkin.medium.com/reordering-bar-and-column-charts-with-ggplot2-in-r-435fad1c643e

2.4 Przedstawić zależność zmierzonego pulsu od deklarowanego statusu palenia papierosów na wykresach typu `geom_violin()` oraz `geom_dotplot()`

2.5 Przedstawić związek pulsu ze wzrostem na wykresach z poziomicami gęstości rozkładu `geom_density_2d` oraz `geom_density_2d_filled`. Wykonać te wykresy osobno dla mężczyzn i dla kobiet.

Przydatne materiały:

https://r-charts.com/correlation/contour-plot-ggplot2/?utm_content=cmp-true

Rozwiązanie

2.1. Wykres porównujący rozpiętości dłoni

Wczytanie danych:

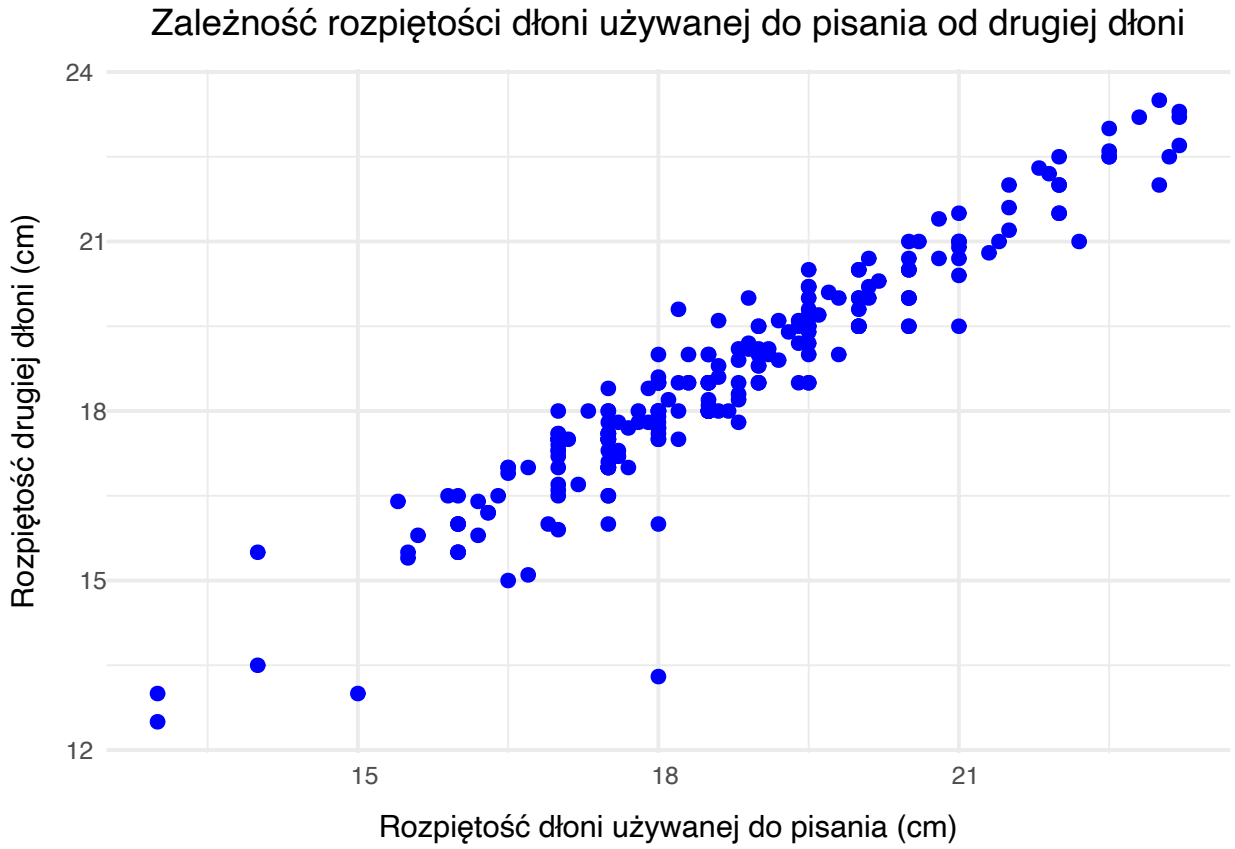
```
library(MASS)
library(ggplot2)
data(survey, package = "MASS")
```

Wyczyszczenie nieprawidłowych wartości rozpiętości dłoni:

```
survey_clean <- na.omit(survey[, c("Wr.Hnd", "NW.Hnd")])
```

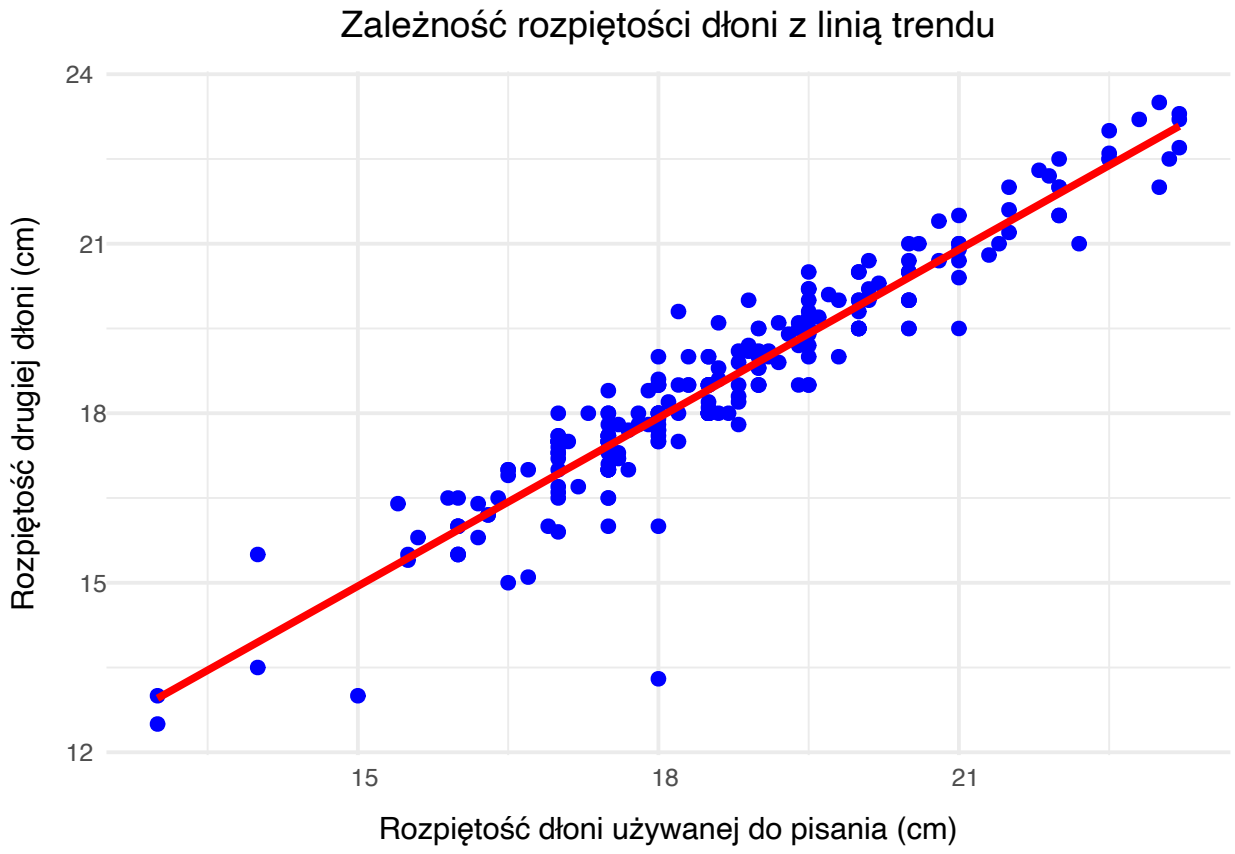
a) Wykres punktowy

```
ggplot(survey_clean, aes(x = Wr.Hnd, y = NW.Hnd)) +
  geom_point(color = "blue") +
  labs(
    title = "Zależność rozpiętości dłoni używanej do pisania od drugiej dłoni",
    x = "Rozpiętość dłoni używanej do pisania (cm)",
    y = "Rozpiętość drugiej dłoni (cm)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, margin = margin(b = 10)),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10))
  )
```



b) Wykres z linią trendu

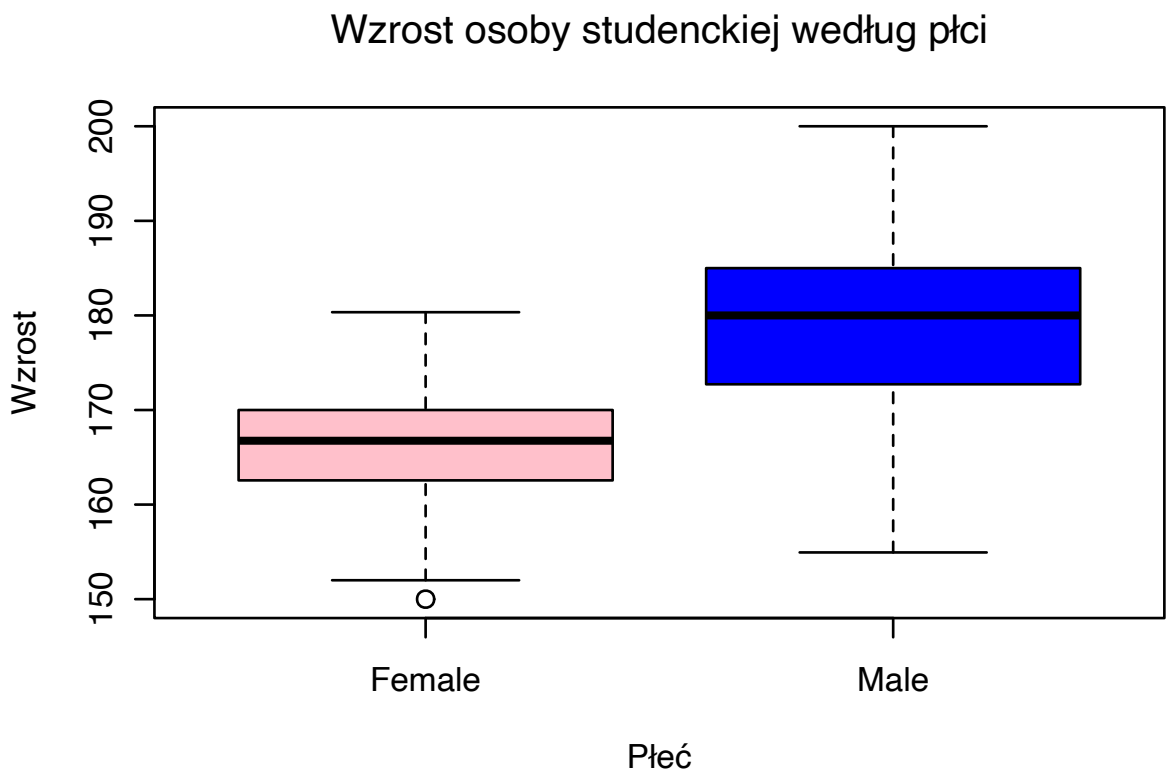
```
ggplot(survey_clean, aes(x = Wr.Hnd, y = NW.Hnd)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", formula = y ~ x, color = "red", se = FALSE) +
  labs(
    title = "Zależność rozpiętości dłoni z linią trendu",
    x = "Rozpiętość dłoni używanej do pisania (cm)",
    y = "Rozpiętość drugiej dłoni (cm)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, margin = margin(b = 10)),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10))
  )
```



2.2. Boxplot dla wzrostu mężczyzn i kobiet z Uniwersytetu Adelaide

Dla przypomnienia, tak wyglądał wykres pudełkowy z zajęć laboratoryjnych numer 1:

```
boxplot(
  survey$Height ~ survey$Sex,
  main = "Wzrost osoby studenckiej według płci",
  xlab = "Płeć",
  ylab = "Wzrost",
  col = c("pink", "blue")
)
```



Teraz wyfiltrujmy dane, aby pozbyć się nieprawidłowych wartości:

```
survey_clean_height <- na.omit(survey[, c("Sex", "Height")])
```

Dzięki pakietowi `ggplot2` możemy dodać punkty do wykresu pudełkowego:

```
ggplot(survey_clean_height, aes(x = Sex, y = Height, fill = Sex)) +
  stat_boxplot(geom = "errorbar", width = 0.2) +
  geom_boxplot(outlier.shape = NA, alpha = 0.5) +
  geom_jitter(aes(color = Sex), width = 0.2, alpha = 0.7) +
  labs(
    title = "Wzrost studentów według płci",
    x = "Płeć",
    y = "Wzrost (cm)",
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, margin = margin(b = 10)),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10))
  )
```

2.3. Wykres słupkowy liczby studentów niepalących/palących okazjonalnie/regularnie/nałogowo

Wyfiltrujmy dane, aby pozbyć się nieprawidłowych wartości:

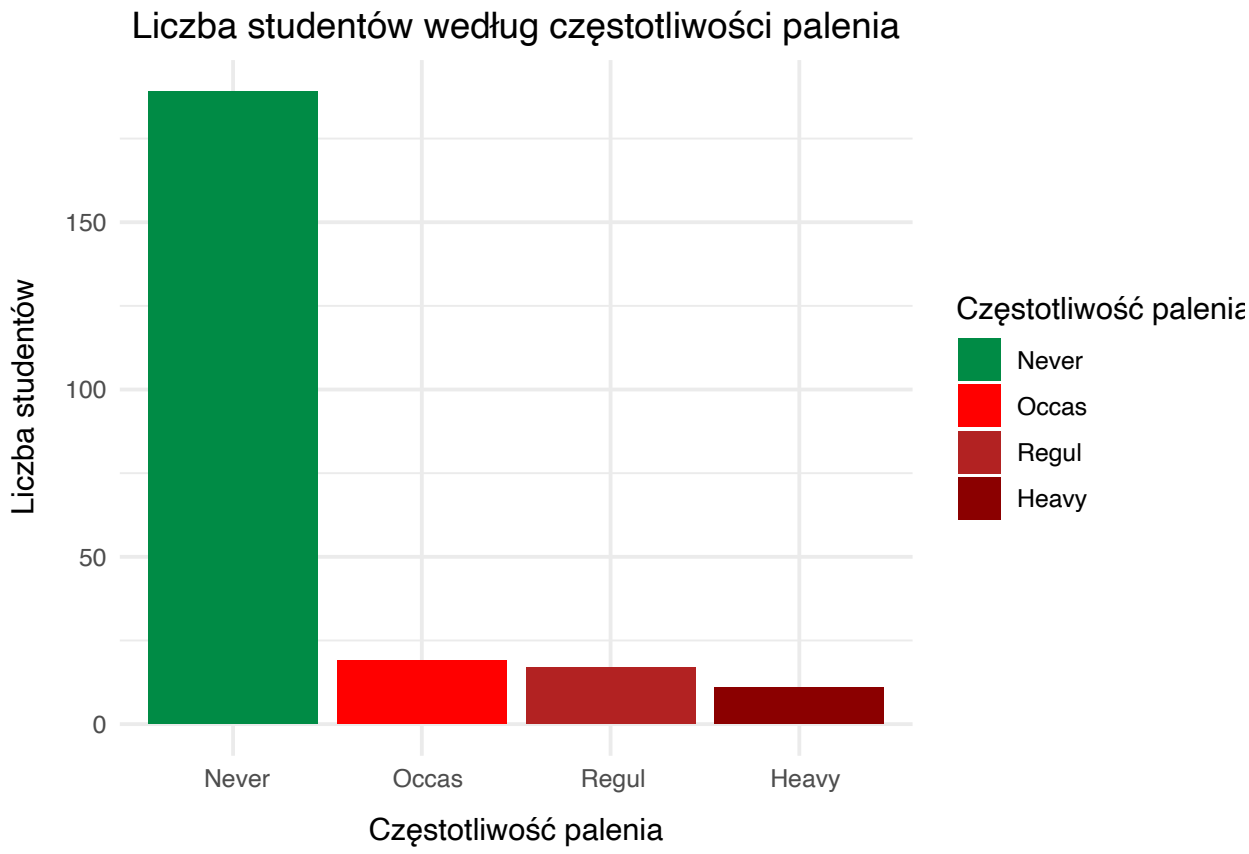
```
survey_clean_smoking <- na.omit(data.frame(Smoke = survey$Smoke))
```

Uporządkowanie poziomów palenia w odpowiedniej kolejności:

```
survey_clean_smoking$Smoke <- factor(
  survey_clean_smoking$Smoke,
  levels = c("Never", "Occas", "Regul", "Heavy")
)
```

Utworzenie wykresu:

```
ggplot(survey_clean_smoking, aes(x = Smoke, fill = Smoke)) +
  geom_bar() +
  scale_fill_manual(
    values = c("springgreen4", "red", "firebrick", "darkred")
  ) +
  labs(
    title = "Liczba studentów według częstotliwości palenia",
    x = "Częstotliwość palenia",
    y = "Liczba studentów",
    fill = "Częstotliwość palenia"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10))
  )
```



2.4. Zależność zmierzonego pulsu od statusu palenia papierosów

Wyfiltrujmy dane, aby pozbyć się nieprawidłowych wartości:

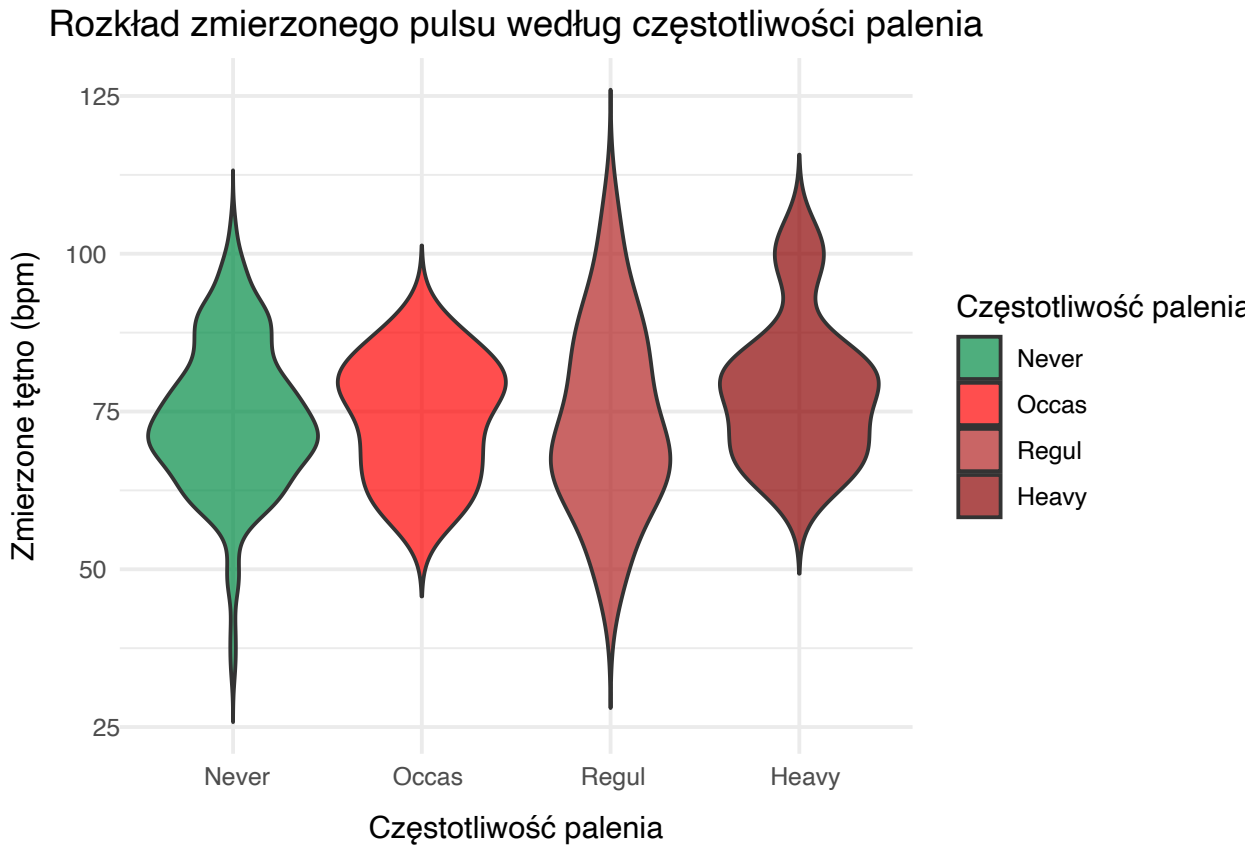
```
survey_clean_pulse <- na.omit(survey[, c("Pulse", "Smoke")])
```

Uporządkowanie częstotliwości palenia w odpowiedniej kolejności:

```
survey_clean_pulse$Smoke <- factor(
  survey_clean_pulse$Smoke,
  levels = c("Never", "Occas", "Regul", "Heavy")
)
```

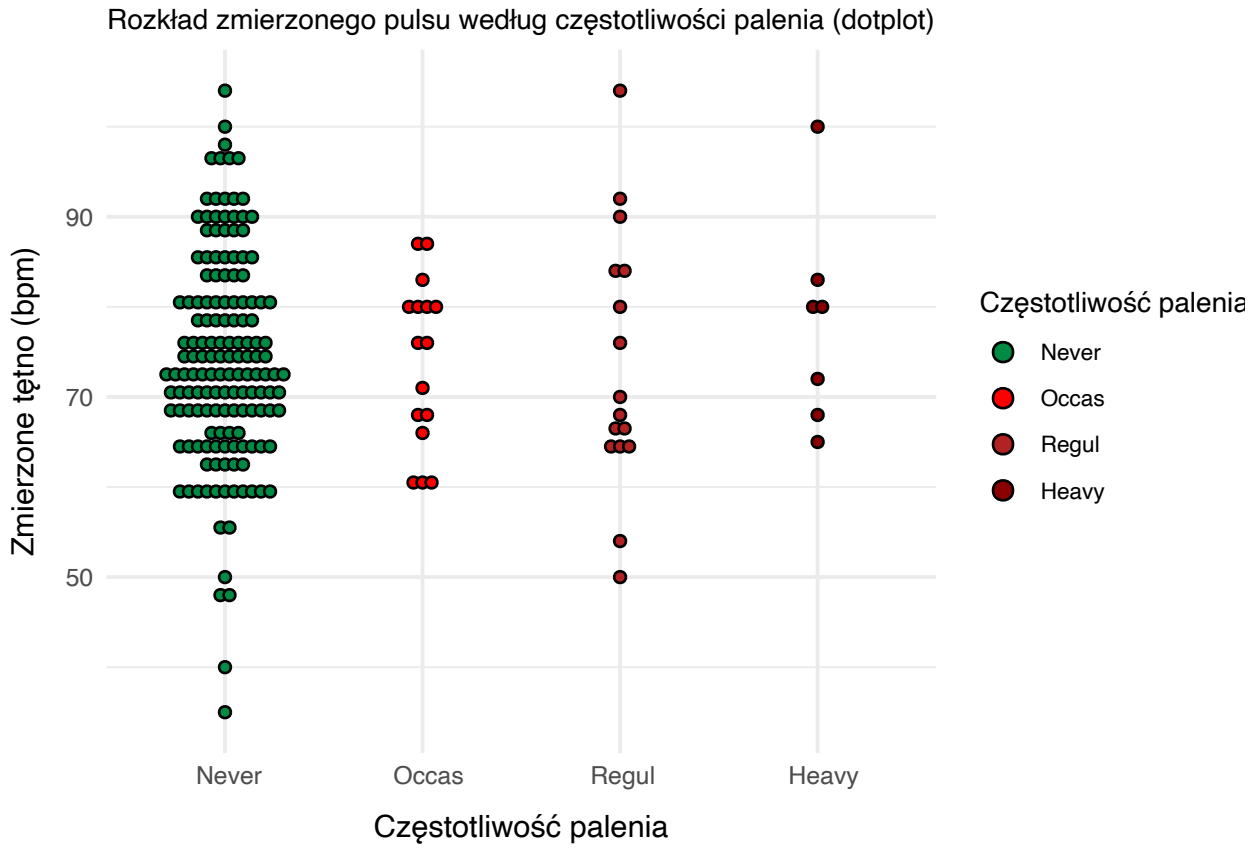
a) Wykres typu violin plot (skrzypcowy)

```r
ggplot(survey_clean_pulse, aes(x = Smoke, y = Pulse, fill = Smoke)) +
  geom_violin(trim = FALSE, alpha = 0.7) +
  scale_fill_manual(
    values = c("springgreen4", "red", "firebrick", "darkred")
  ) +
  labs(
    title = "Rozkład zmierzonego pulsu według częstotliwości palenia",
    x = "Częstotliwość palenia",
    y = "Zmierzone tętno (bpm)",
    fill = "Częstotliwość palenia"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10))
  )
```



b) Wykres typu dotplot (stacked dotted plot)

```
ggplot(survey_clean_pulse, aes(x = Smoke, y = Pulse, fill = Smoke)) +
  geom_dotplot(binaxis = "y", stackdir = "center", dotsize = 0.5, binwidth = 2) +
  scale_fill_manual(
    values = c("springgreen4", "red", "firebrick", "darkred")
  ) +
  labs(
    title = "Rozkład zmierzonego pulsu według częstotliwości palenia (dotplot)",
    x = "Częstotliwość palenia",
    y = "Zmierzone tętno (bpm)",
    fill = "Częstotliwość palenia"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10),
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10)),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 8)
  )
```



2.5. Związek pulsu ze wzrostem na wykresach z poziomicami gęstości rozkładu

Wyfiltrujmy dane, aby pozbyć się nieprawidłowych wartości:

```
survey_clean_height_pulse <- na.omit(survey[, c("Height", "Pulse", "Sex")])
```

Deklaracja funkcji wyświetlającej wykres z `geom_density_2d`:

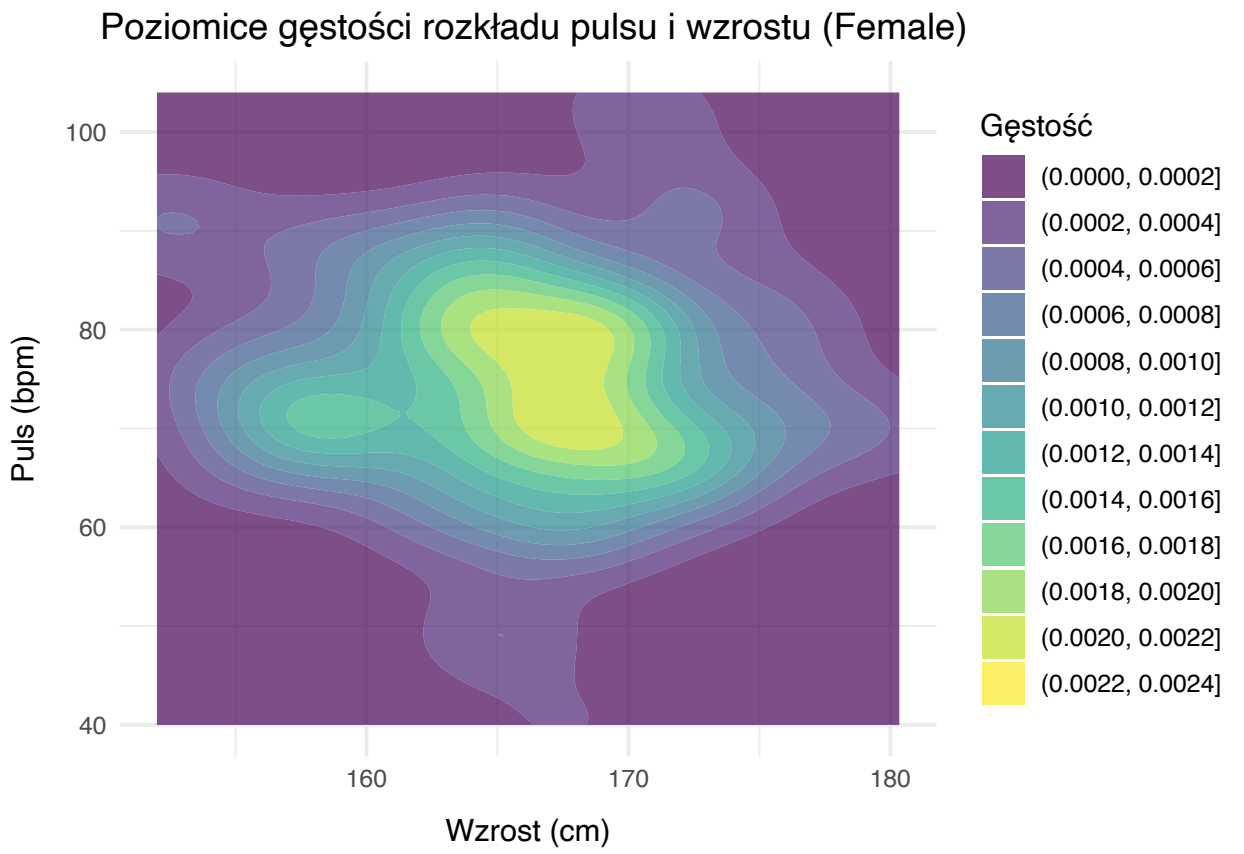
```
plot_height_pulse_density <- function(sex, color) {
  ggplot(
    subset(survey_clean_height_pulse, Sex == sex),
    aes(x = Height, y = Pulse)
  ) +
    geom_density_2d(color = color) +
    labs(
      title = paste("Poziomice gęstości rozkładu pulsu i wzrostu (", sex, ")", sep = ""),
      x = "Wzrost (cm)",
      y = "Puls (bpm)"
    ) +
    theme_minimal() +
    theme(
      plot.title = element_text(hjust = 0.5),
      axis.title.x = element_text(margin = margin(t = 10)),
      axis.title.y = element_text(margin = margin(r = 10))
    )
}
```

Deklaracja funkcji wyświetlającej wykres z `geom_density_2d_filled`:

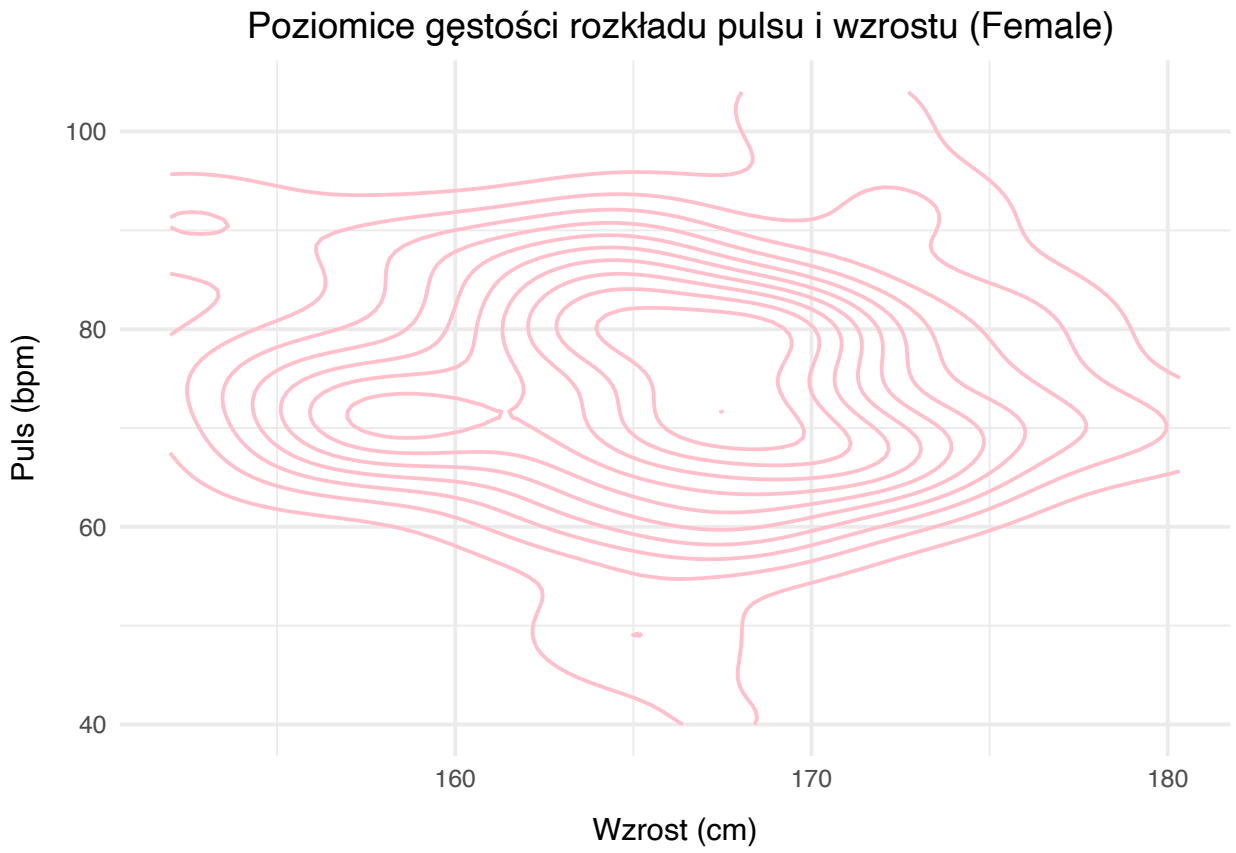
```
plot_height_pulse_density_filled <- function(sex) {
  ggplot(
    subset(survey_clean_height_pulse, Sex == sex),
    aes(x = Height, y = Pulse)
  ) +
    geom_density_2d_filled(alpha = 0.7) +
    scale_fill_viridis_d() +
    labs(
      title = paste("Poziomice gęstości rozkładu pulsu i wzrostu (", sex, ")", sep = ""),
      x = "Wzrost (cm)",
      y = "Puls (bpm)",
      fill = "Gęstość"
    ) +
    theme_minimal() +
    theme(
      plot.title = element_text(hjust = 0.5),
      axis.title.x = element_text(margin = margin(t = 10)),
      axis.title.y = element_text(margin = margin(r = 10))
    )
}
```

Wyświetlenie wykresów dla kobiet

```
plot_height_pulse_density_filled("Female")
```

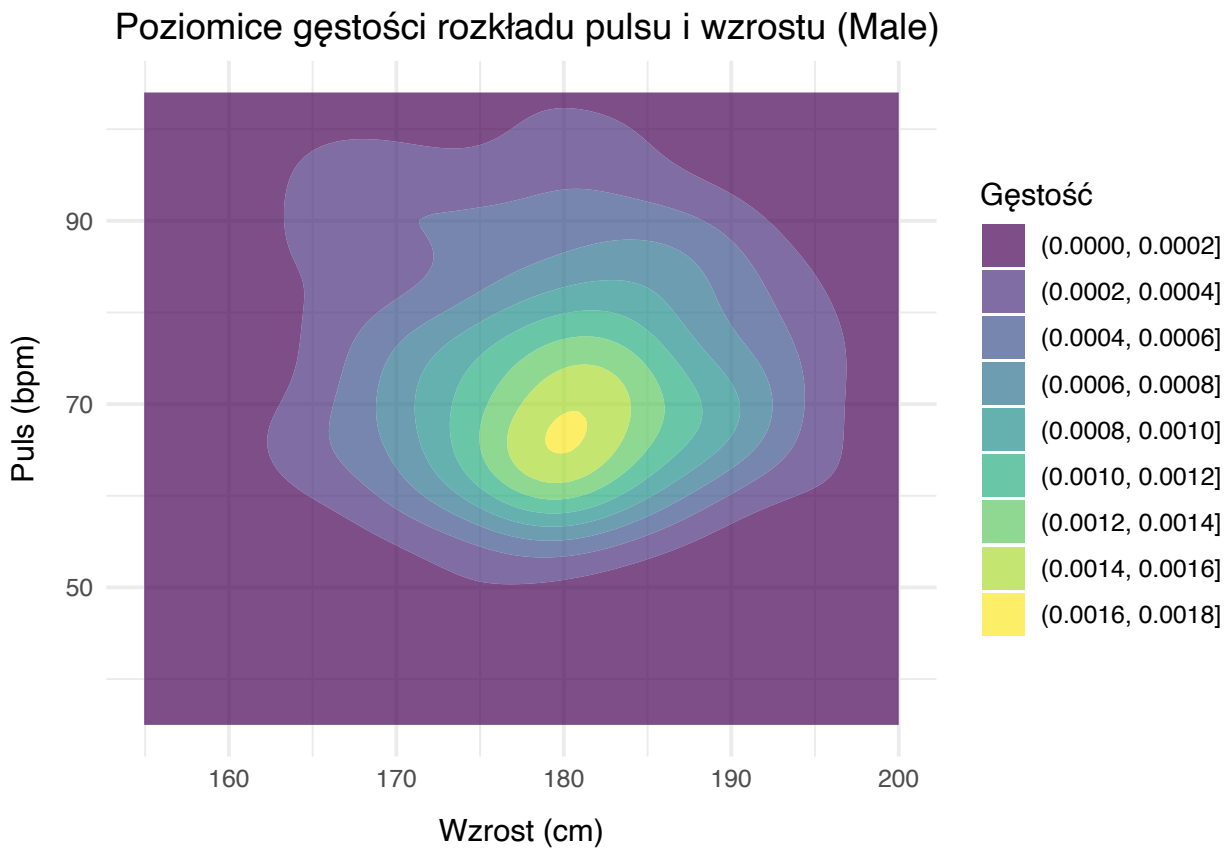


```
plot_height_pulse_density("Female", "pink")
```

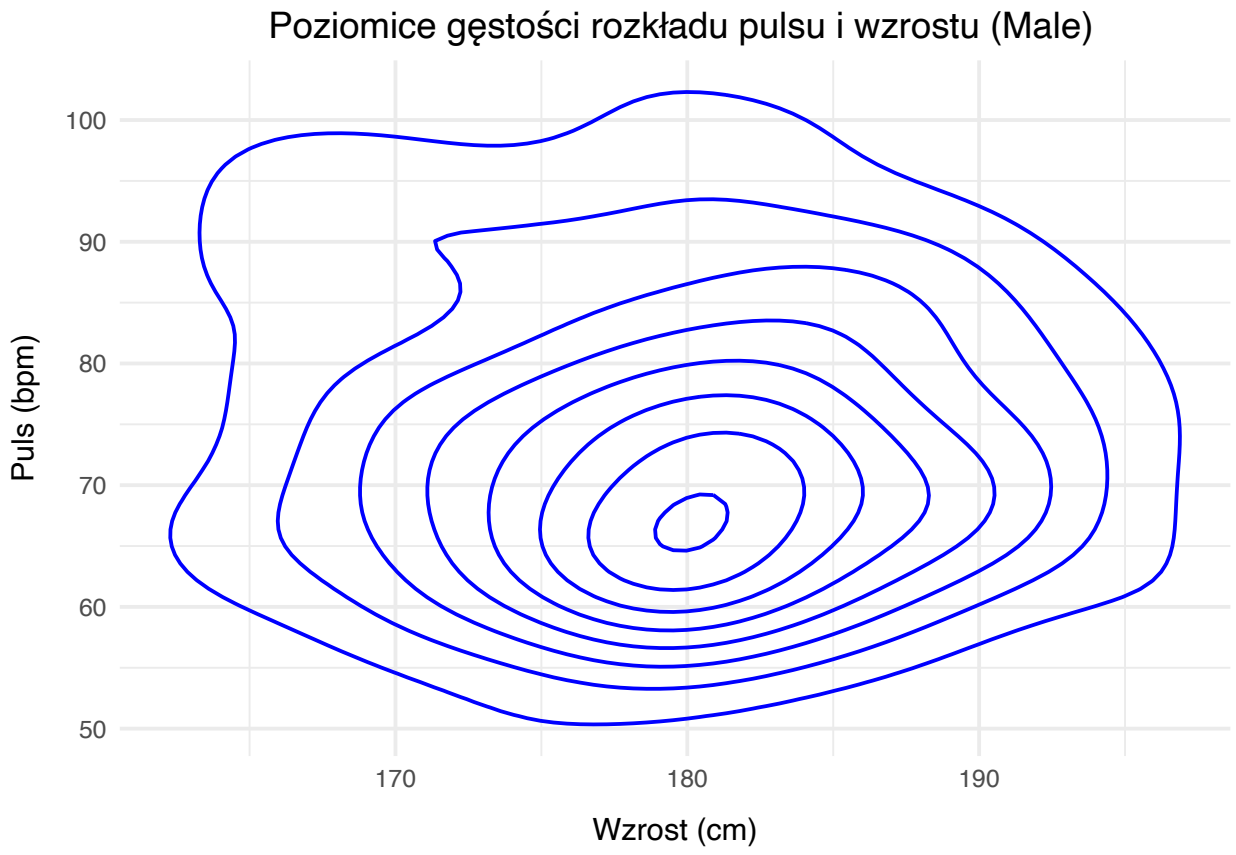


Wyświetlenie wykresów dla mężczyzn

```
plot_height_pulse_density_filled("Male")
```



```
plot_height_pulse_density("Male", "blue")
```



Zadanie 3 - Statystyki i geometrie w ggplot2

Treść

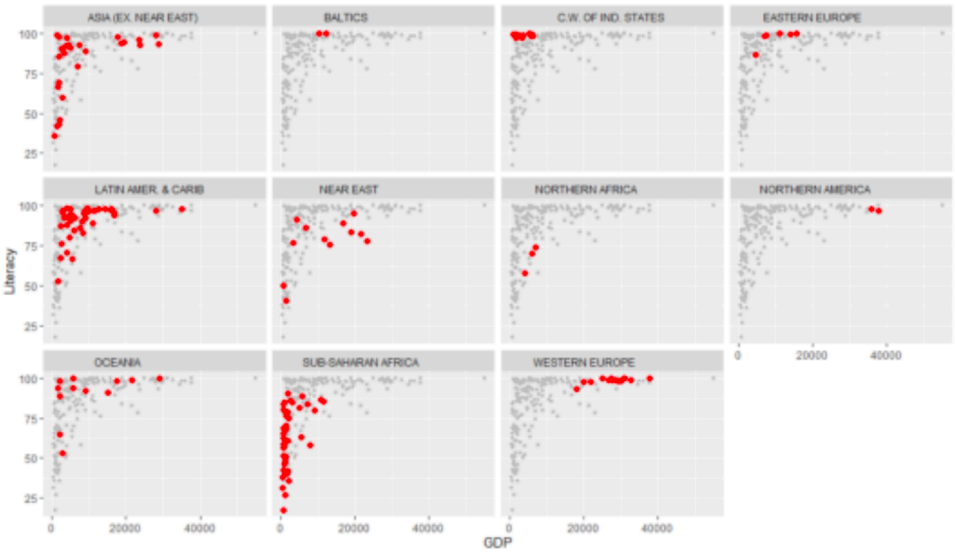
3.1 Ze strony https://github.com/bnokoro/Data-Science/blob/master/countries%20of%20the%20world.csv pobrać plik countries of the world.csv i zapisać go w R jako data frame countries. Uprościć nagłówki kolumn, aby łatwiej było się nimi później posługiwać.

3.2 Przedstawić na wykresie z pakietu `ggplot2` zależność piśmienności (oś pionowa) od PKB per capita (oś pozioma) dla wszystkich zawartych w tym data frame państw. Wynikiem tej części zadania powinien być wykres podobny do przedstawionego na Rys. 1, na którego poszczególnych panelach punkty odpowiadające wybranemu regionowi świata oznaczone są większymi czerwonymi kropkami, a te odnoszące się do reszty świata mniejszymi szarymi.

Potrzebne funkcje: `facet_wrap()`, `geom_point(size, color)`

Przydatne materiały:

https://pbiecek.github.io/Przewodnik/Wizualizacja/ggplot2/06_panele.html



Rysunek 1

3.3 W podobny sposób przedstawić zależność wypadkowej migracji od gęstości zaludnienia. Użyć skali logarytmicznej dla gęstości zaludnienia.

Rozwiązanie

3.1. Wczytanie danych

Pobranie pliku z danymi:

```
url <- "https://github.com/bnokoro/Data-Science/raw/master/countries%20of%20the%20world.csv"
download.file(url, destfile = "countries_of_the_world.csv", mode = "wb")
```

Wczytanie danych do R:

```
countries <- read.csv("countries_of_the_world.csv", stringsAsFactors = FALSE)
```

Uproszczenie nagłówków kolumn:

```
colnames(countries) <- gsub(" ", "_", colnames(countries))
colnames(countries) <- gsub("\\.", "", colnames(countries))
colnames(countries) <- tolower(colnames(countries))
```

Wyświetlenie pierwszy 5 wierszy:

```
head(countries, 5)
```

```
##           country                               region population areasqmi
## 1    Afghanistan        ASIA (EX. NEAR EAST)            31056997   647500
## 2        Albania  EASTERN EUROPE                           3581655    28748
## 3        Algeria  NORTHERN AFRICA                         32930091  2381740
## 4 American Samoa  OCEANIA                                    57794      199
## 5        Andorra  WESTERN EUROPE                            71201      468
##   popdensitypersqmi coastlinecoastarearatio netmigration
## 1              48,0                    0,00        23,06
## 2             124,6                    1,26        -4,93
## 3              13,8                    0,04        -0,39
## 4             290,4                   58,29       -20,71
## 5             152,1                    0,00          6,6
##   infantmortalityper1000births gdppercapita literacy phonesper1000 arable crops
## 1                       163,07          700     36,0           3,2  12,13  0,22
## 2                        21,52         4500     86,5          71,2  21,09  4,42
## 3                           31         6000     70,0          78,1   3,22  0,25
## 4                         9,27         8000     97,0         259,5     10    15
## 5                         4,05        19000    100,0         497,2   2,22     0
##   other climate birthrate deathrate agriculture industry service
## 1 87,65       1      46,6     20,34        0,38     0,24    0,38
## 2 74,49       3     15,11      5,22       0,232    0,188   0,579
## 3 96,53       1     17,14      4,61       0,101      0,6   0,298
## 4    75       2     22,46      3,27
## 5 97,78       3      8,71      6,25
```

3.2. Wyświetlenie wykresu zależności piśmienności od PKB per capita

Czyszczenie danych:

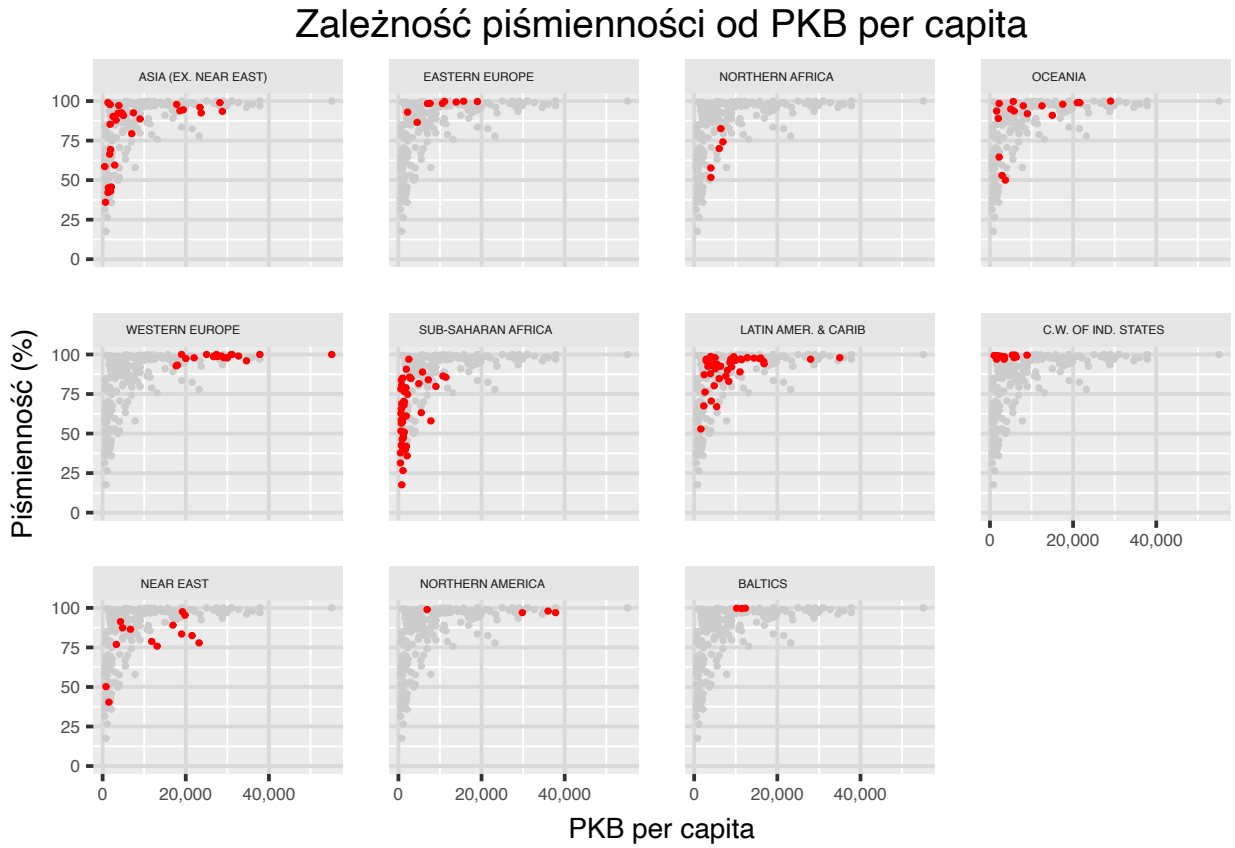
```
library(dplyr)
library(scales)

countries$gdppercapita <- as.numeric(gsub(",", ".", countries$gdppercapita))
countries$literacy <- as.numeric(gsub(",", ".", countries$literacy))
countries_clean <- na.omit(countries[, c("gdppercapita", "region", "literacy")])
```

Tworzenie nowego dataframe - każdy punkt pojawi się w każdym panelu:

```
all_regions <- unique(countries_clean$region)
plot_data <- expand.grid(facet_region = all_regions, id = 1:nrow(countries_clean)) %>%
  mutate(
    gdppercapita = countries_clean$gdppercapita[id],
    literacy = countries_clean$literacy[id],
    region = countries_clean$region[id],
    highlight = region == facet_region
  )
```

Wyświetlenie wykresu:

```
ggplot(plot_data, aes(x = gdppercapita, y = literacy)) +
  geom_point(data = subset(plot_data, !highlight), color = "grey80", size = 0.05) +
  geom_point(data = subset(plot_data, highlight), color = "red", size = 0.05) +
  facet_wrap(~ facet_region, scales = "fixed", ncol = 4) +
  scale_x_continuous(labels = comma) +
  scale_y_continuous(limits = c(0, 100)) +
  labs(
    title = "Zależność piśmienności od PKB per capita",
    x = "PKB per capita",
    y = "Piśmienność (%)"
  ) +
  theme_gray() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 6, margin = margin(r = 6)),
    strip.text = element_text(size = 4),
    strip.placement = "inside",
    strip.background = element_rect(fill = "grey90", color = NA),
    panel.grid.major = element_line(color = "grey85"),
    panel.spacing = unit(1.2, "lines"),
    axis.title.y = element_text(margin = margin(r = 6)),
    axis.title.x = element_text(margin = margin(t = 6))
  )
```



3.2. Wyświetlenie wykresu zależności wypadkowej migracji od zaludnienia

Czyszczenie danych:

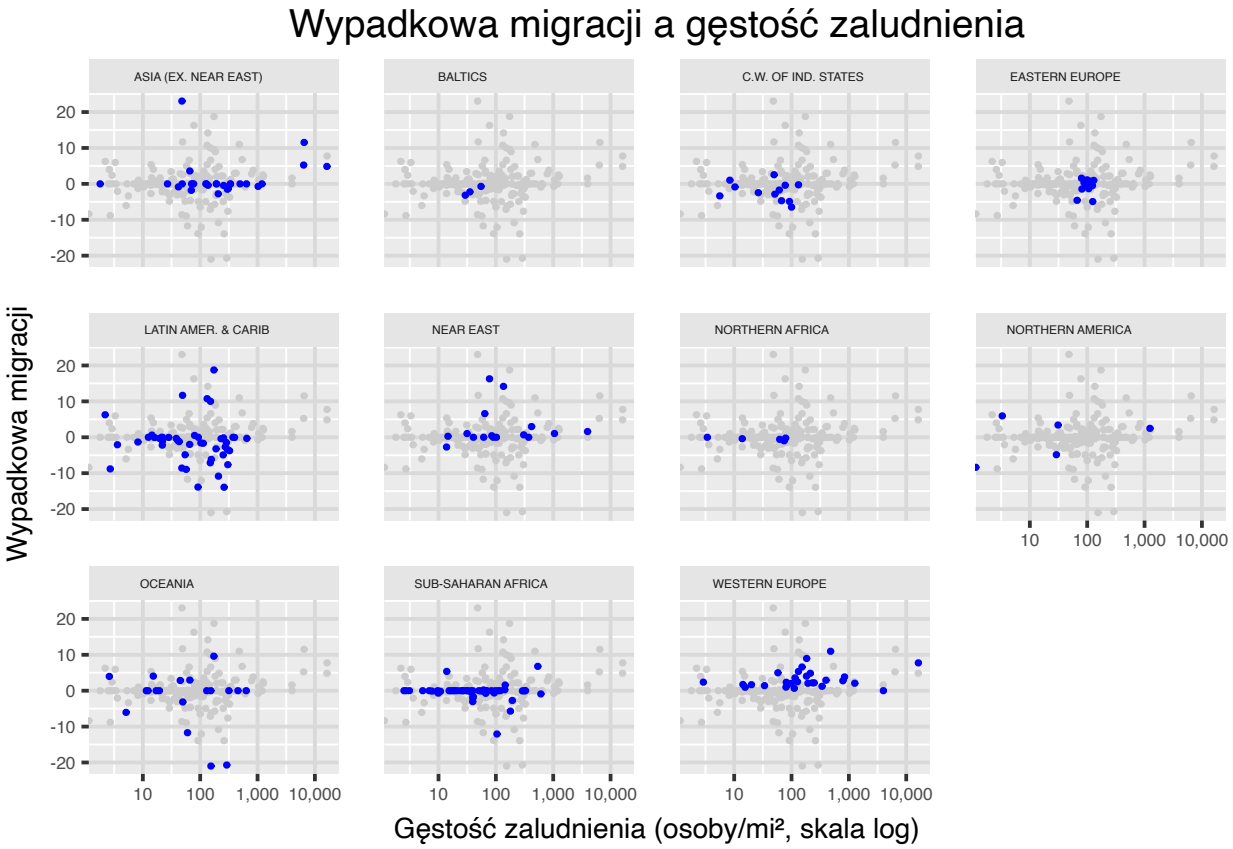
```
countries$netmigration <- as.numeric(gsub(",", ".", countries$netmigration))
countries$popdensitypersqmi <- as.numeric(gsub(",", ".", countries$popdensitypersqmi))
countries$region <- as.factor(countries$region)
countries_migration <- na.omit(countries[, c("popdensitypersqmi", "netmigration", "region")])
```

Tworzenie nowego dataframe - każdy punkt pojawi się w każdym panelu:

```
all_regions <- unique(countries_migration$region)
plot_data <- expand.grid(facet_region = all_regions, id = 1:nrow(countries_migration)) %>%
  mutate(
    netmigration = countries_migration$netmigration[id],
    popdensitypersqmi = countries_migration$popdensitypersqmi[id],
    region = countries_migration$region[id],
    highlight = region == facet_region
  )
```

Wyświetlenie wykresu:

```
ggplot(plot_data, aes(x = popdensitypersqmi, y = netmigration)) +
  geom_point(data = subset(plot_data, !highlight), color = "grey80", size = 0.05) +
  geom_point(data = subset(plot_data, highlight), color = "blue", size = 0.05) +
  facet_wrap(~ facet_region, scales = "fixed", ncol = 4) +
  scale_x_log10(labels = comma) +
  scale_y_continuous() +
  labs(
    title = "Wypadkowa migracji a gęstość zaludnienia",
    x = "Gęstość zaludnienia (osoby/mi², skala log)",
    y = "Wypadkowa migracji"
  ) +
  theme_gray() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 6, margin = margin(r = 6)),
    strip.text = element_text(size = 4),
    strip.placement = "inside",
    strip.background = element_rect(fill = "grey90", color = NA),
    panel.grid.major = element_line(color = "grey85"),
    panel.spacing = unit(1.2, "lines"),
    axis.title.y = element_text(margin = margin(r = 6)),
    axis.title.x = element_text(margin = margin(t = 6))
  )
```



Zadanie 4 - "Długi" format danych

Treść

4.1 Sprawdzić w R działanie linii kodu:

```
1000 %>% rnorm(mean = 0, sd = 5) %>% hist(breaks = 20)
```

Jakie jest znaczenie operatora `%>%`?

4.2 Za pomocą funkcji `data.frame()` skonstruować "ręcznie" następujący data frame:

pokarm	weglowodany	bialko	tluszcz
chleb	49.0	9.0	3.2
jablko	14.0	0.3	0.2
szynka	1.5	21.0	6.0

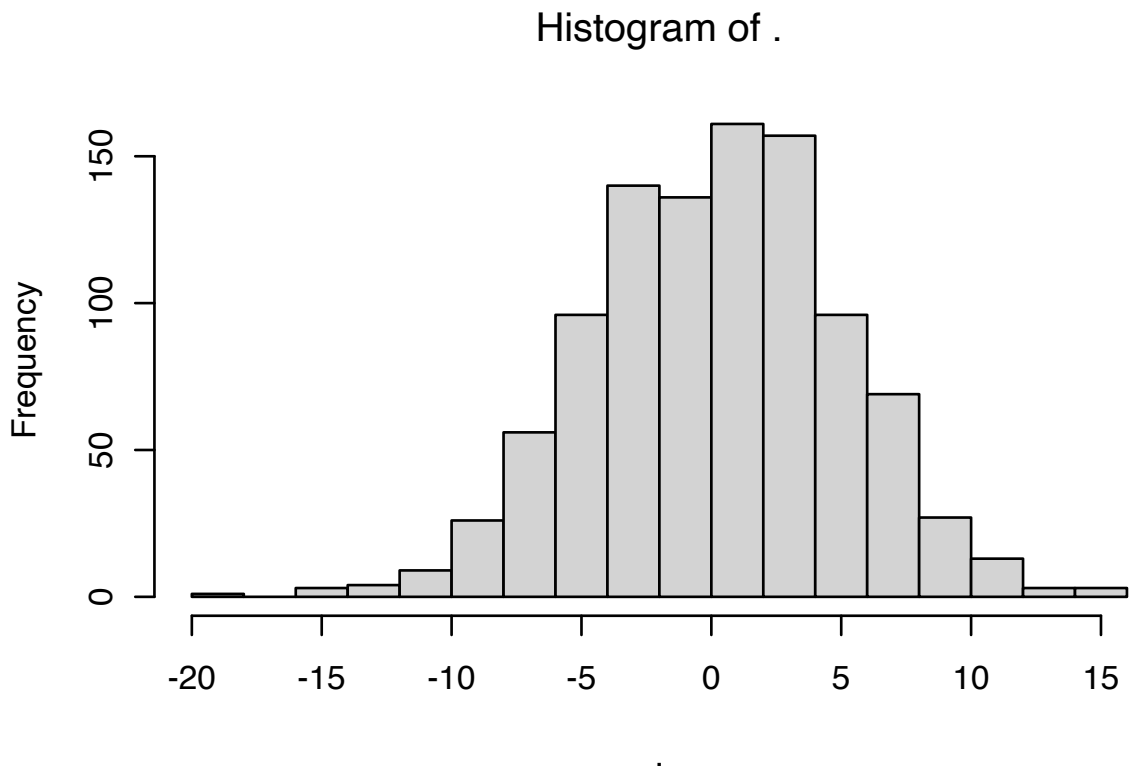
4.3 Zainstalować pakiet tidyverse (`install.packages("tidyverse")`). Korzystając z funkcji `pivot_longer()` tego pakietu przekształcić powyszy dataframe do postaci:

pokarm	wart odzywcza	zawartosc
chleb	weglowodany	49.0
chleb	bialko	9.0
chleb	tluszcz	3.2
jablko	weglowodany	14.0
jablko	bialko	0.3
...	...	...

Rozwiązanie

4.1. Znaczenie operatora %>%

```
1000 %>% rnorm(mean = 0, sd = 5) %>% hist(breaks = 20)
```



Operator %>% jest operatorem pipe, który pozwala na przekazywanie wyniku jednej funkcji jako argumentu do następnej funkcji. W tym przypadku, generuje 1000 losowych liczb z rozkładu normalnego o średniej 0 i odchyleniu standardowym 5, a następnie tworzy histogram tych danych z 20 przedziałami.

Przykładowym odpowiednikiem tego operatora będzie np. operator | (pipe) w bashu.

4.2. Konstrukcja dataframe - pokarm, węglowodany, białko, tłuszcz

```
food_df <- data.frame(
  pokarm = c("chleb", "jabłko", "szynka"),
  weglowodany = c(49, 14, 1.5),
  bialko = c(9, 0.3, 21),
  tluszcz = c(3.2, 0.2, 6)
)
food_df
```

```
##   pokarm weglowodany bialko tluszcz
## 1  chleb        49.0    9.0     3.2
## 2 jabłko        14.0    0.3     0.2
## 3 szynka         1.5   21.0     6.0
```

4.2. Przekształcenie dataframe

```
library(tidyverse)

food_pivoted <- pivot_longer(
  food_df,
  cols = c(weglowodany, bialko, tluszcz),
  names_to = "wart_odzywcza",
  values_to = "zawartosc"
)
food_pivoted
```

```
## # A tibble: 9 x 3
##   pokarm wart_odzywcza zawartosc
##   <chr>  <chr>             <dbl>
## 1 chleb  weglowodany        49
## 2 chleb  bialko              9
## 3 chleb  tluszcz             3.2
## 4 jabłko weglowodany        14
## 5 jabłko bialko              0.3
## 6 jabłko tluszcz             0.2
## 7 szynka weglowodany         1.5
## 8 szynka bialko             21
## 9 szynka tluszcz             6
```

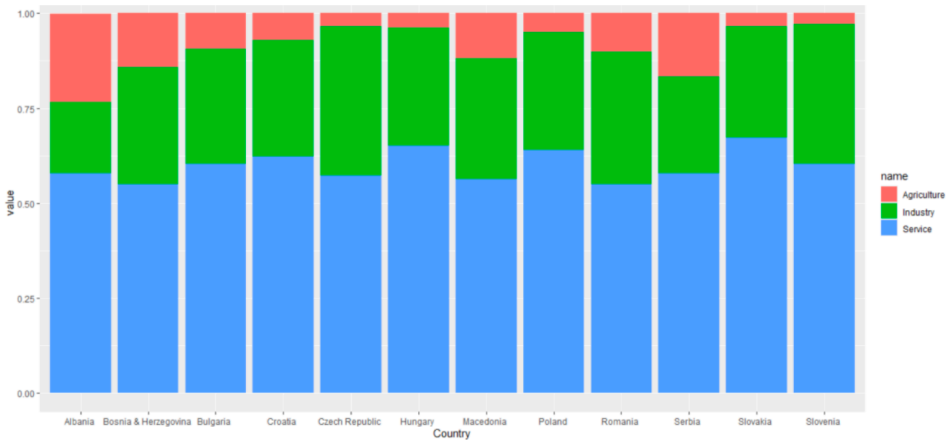
Zadanie 5

Treść

Przedstawić na wykresie słupkowym udział rolnictwa/przemysłu/usług w gospodarce poszczególnych krajów Europy Wschodniej, tak, jak przedstawiono to na Rys. 2.

Przydatne materiały

https://stackoverflow.com/questions/71359655/how-to-create-a-stacked-bar-chart-in-r-with-ggplot



Rysunek 2

Rozwiązanie Czyszczenie danych:

```
countries$agriculture <- as.numeric(gsub(",", ".", countries$agriculture))
countries$industry <- as.numeric(gsub(",", ".", countries$industry))
countries$service <- as.numeric(gsub(",", ".", countries$service))
countries$country <- as.factor(countries$country)
countries$region <- as.factor(countries$region)
countries_sectors <- na.omit(countries[, c(
  "agriculture", "industry", "service",
  "country", "region"
)])
```

Wybranie krajów Europy Wschodniej:

```
eastern_europe_sectors <- countries_sectors[
  grepl(
    "eastern europe",
    countries_sectors$region,
    ignore.case = TRUE
  ),
]
```

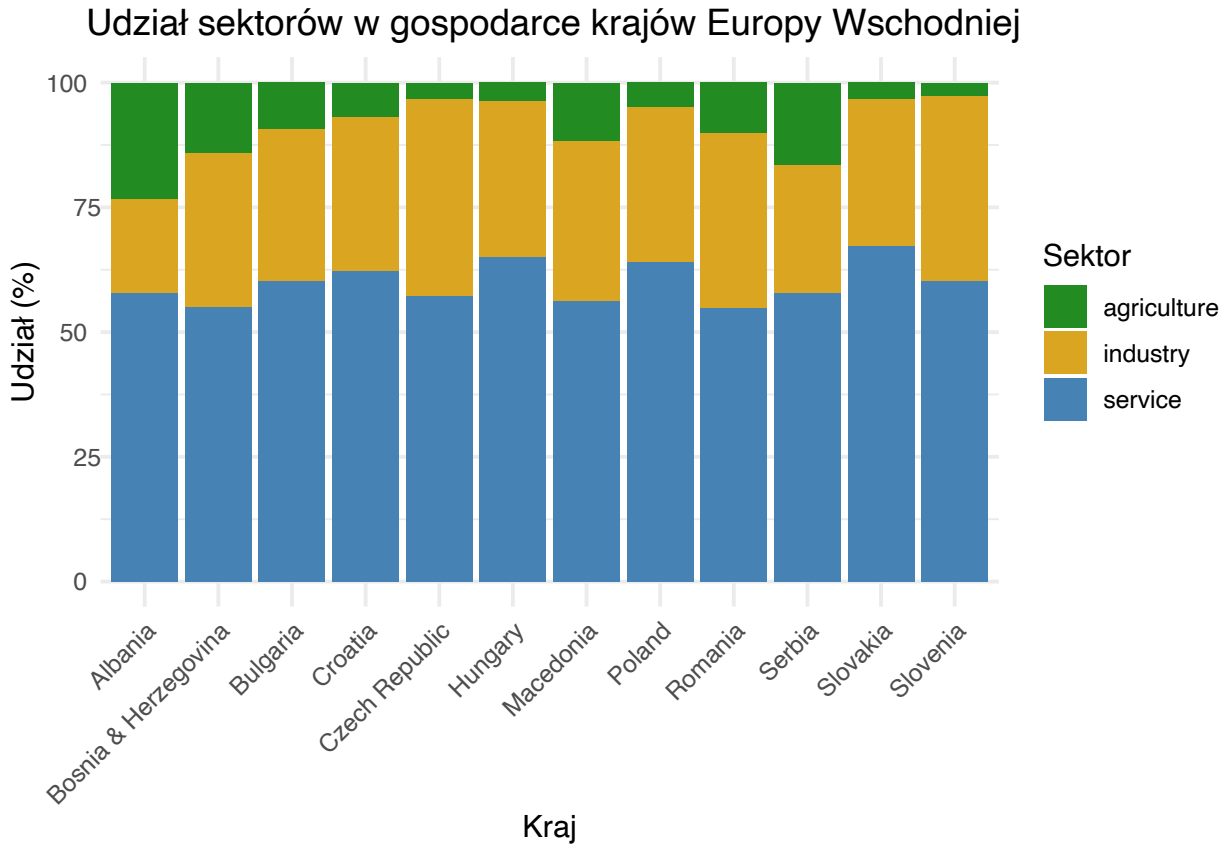
Przekształcenie do formatu długiego + przemnożenie przez 100:

```
eastern_europe_sectors_long <- pivot_longer(
  eastern_europe_sectors,
  cols = c(agriculture, industry, service),
  names_to = "sektor",
```

```
  values_to = "udzial"
) %>%
  mutate(udzial = udzial * 100)
```

Wyświetlenie wykresu:

```
ggplot(eastern_europe_sectors_long, aes(x = country, y = udzial, fill = sektor)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("forestgreen", "goldenrod", "steelblue")) +
  labs(
    title = "Udział sektorów w gospodarce krajów Europy Wschodniej",
    x = "Kraj",
    y = "Udział (%)",
    fill = "Sektor"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(hjust = 0.5)
  )
```



Zadanie 6

Treść

Przedstawić te dane na wykresach kołowych osobnych dla każdego z analizowanych krajów.

Przydatne materiały
https://r-graph-gallery.com/piechart-ggplot2.html

Rozwiązanie

Wyświetlenie wykresów:

```
ggplot(eastern_europe_sectors_long, aes(x = "", y = udzial, fill = sektor)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar("y") +
  facet_wrap(~ country, ncol = 4) +
  scale_fill_manual(values = c("forestgreen", "goldenrod", "steelblue")) +
  labs(
    title = "Sektory gospodarki krajów Europy Wschodniej",
    fill = "Sektor"
  ) +
  theme_void() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10, margin = margin(b = 10)),
    strip.text = element_text(size = 8),
    legend.position = "bottom"
  )
```