Ćwiczenia z analizy danych w R - Spotkanie 2.

mmcharchuta

2025-04-08

ĆWICZENIE NR 1: Wczytaj i zbadaj zbiór danych

```
# Load necessary libraries
library(tidyverse)

# Load the dataset
url <- "https://raw.githubusercontent.com/mwaskom/seaborn-data/master/tips.csv"
tips <- read_csv(url)

# Display the structure of the dataset
str(tips)
```

```
## spc_tbl_ [244 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ total_bill: num [1:244] 17 10.3 21 23.7 24.6 ...
##  $ tip       : num [1:244] 1.01 1.66 3.5 3.31 3.61 4.71 2 3.12 1.96 3.23 ...
##  $ sex       : chr [1:244] "Female" "Male" "Male" "Male" ...
##  $ smoker    : chr [1:244] "No" "No" "No" "No" ...
##  $ day       : chr [1:244] "Sun" "Sun" "Sun" "Sun" ...
##  $ time      : chr [1:244] "Dinner" "Dinner" "Dinner" "Dinner" ...
##  $ size      : num [1:244] 2 3 3 2 4 4 2 4 2 2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   total_bill = col_double(),
##   ..   tip = col_double(),
##   ..   sex = col_character(),
##   ..   smoker = col_character(),
##   ..   day = col_character(),
##   ..   time = col_character(),
##   ..   size = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
# Count unique values in the 'day' column
unique_days <- n_distinct(tips$day)
cat("Liczba unikalnych wartości w kolumnie 'day':", unique_days, "\n")
```

```
## Liczba unikalnych wartości w kolumnie 'day': 4
```

ĆWICZENIE NR 2: Tworzenie prostego wykresu z ggplot2

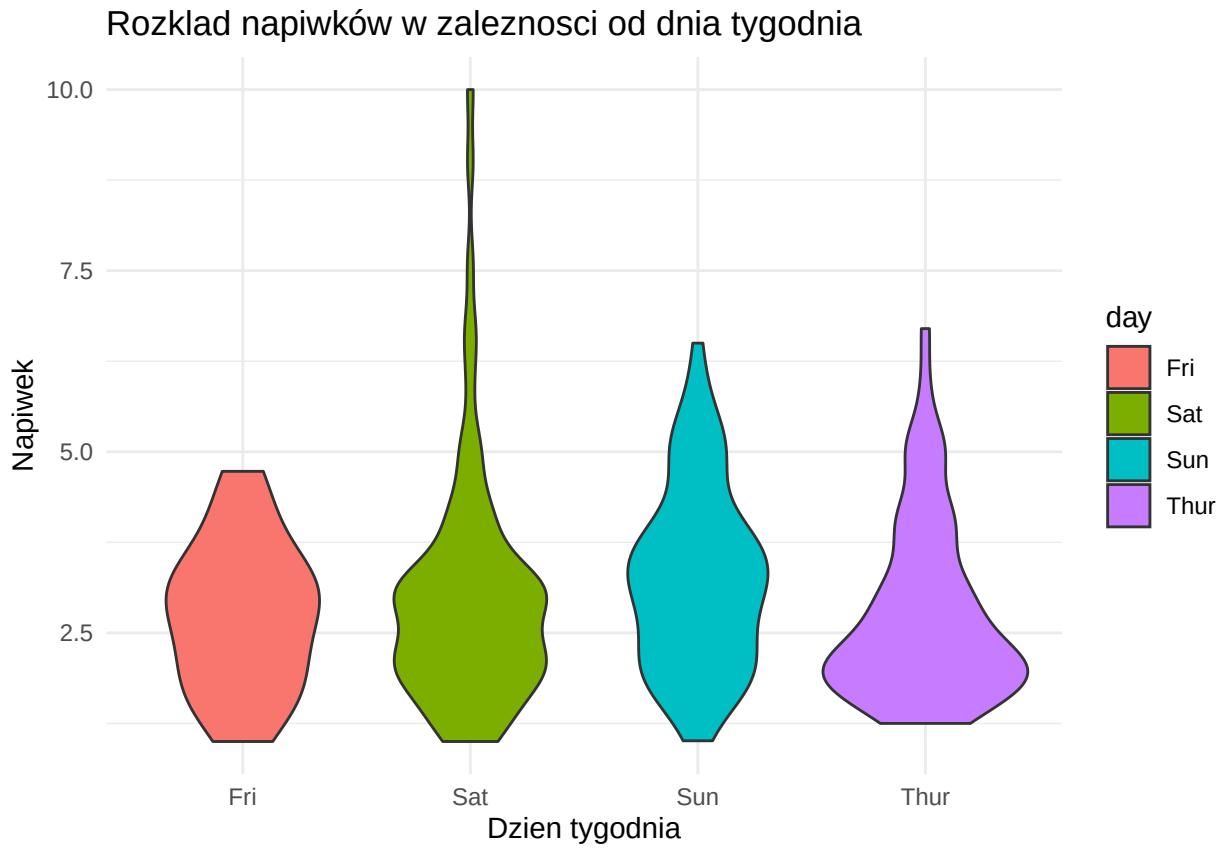
```
# Load ggplot2
library(ggplot2)
```

```r
# Find the day with the highest average tip
highest_tips_day <- tips %>%
  group_by(day) %>%
  summarise(avg_tip = mean(tip)) %>%
  arrange(desc(avg_tip)) %>%
  slice(1)

cat("Dzień tygodnia z najwyższymi napiwkami:", highest_tips_day$day, "\n")
```

```
## Dzień tygodnia z najwyższymi napiwkami: Sun
```

```r
# Create a violin plot for tips by day
ggplot(tips, aes(x = day, y = tip, fill = day)) +
  geom_violin() +
  labs(title = "Rozkład napiwków w zależności od dnia tygodnia",
       x = "Dzień tygodnia",
       y = "Napiwek") +
  theme_minimal()
```



ĆWICZENIE NR 3: Interaktywny wykres z plotly

```r
# Load plotly
library(plotly)
```

```r
# Modify the 'sex' column to use custom labels
polish_tips <- tips %>%
  mutate(sex = recode(sex, "Female" = "Kobieta", "Male" = "Mężczyzna"))

# Calculate the correlation between total_bill and tip for all clients
correlation <- cor(tips$total_bill, tips$tip, method = "pearson")
cat("Współczynnik Korelacji Pearsona rachunku do napiwku (dla wszystkich):", correlation, "\n")
```

```
## Współczynnik Korelacji Pearsona rachunku do napiwku (dla wszystkich): 0.6757341
```

```r
# Calculate the correlation for female clients
correlation_female <- cor(tips$total_bill[tips$sex == "Female"], tips$tip[tips$sex == "Female"], method
cat("Współczynnik Korelacji Pearsona rachunku do napiwku (dla kobiet):", correlation_female, "\n")
```

```
## Współczynnik Korelacji Pearsona rachunku do napiwku (dla kobiet): 0.6829993
```

```r
# Calculate the correlation for male clients
correlation_male <- cor(tips$total_bill[tips$sex == "Male"], tips$tip[tips$sex == "Male"], method = "pea
cat("Współczynnik Korelacji Pearsona rachunku do napiwku (dla mężczyzn):", correlation_male, "\n")
```

```
## Współczynnik Korelacji Pearsona rachunku do napiwku (dla mężczyzn): 0.669753
```

```r
# Calculate the correlation between total_bill and tip for all clients
correlation <- cor(tips$total_bill, tips$tip, method = "spearman")
cat("Współczynnik Korelacji Spearmana rachunku do napiwku (dla wszystkich):", correlation, "\n")
```

```
## Współczynnik Korelacji Spearmana rachunku do napiwku (dla wszystkich): 0.6789681
```

```r
# Calculate the correlation for female clients
correlation_female <- cor(tips$total_bill[tips$sex == "Female"], tips$tip[tips$sex == "Female"], method
cat("Współczynnik Korelacji Spearmana rachunku do napiwku (dla kobiet):", correlation_female, "\n")
```

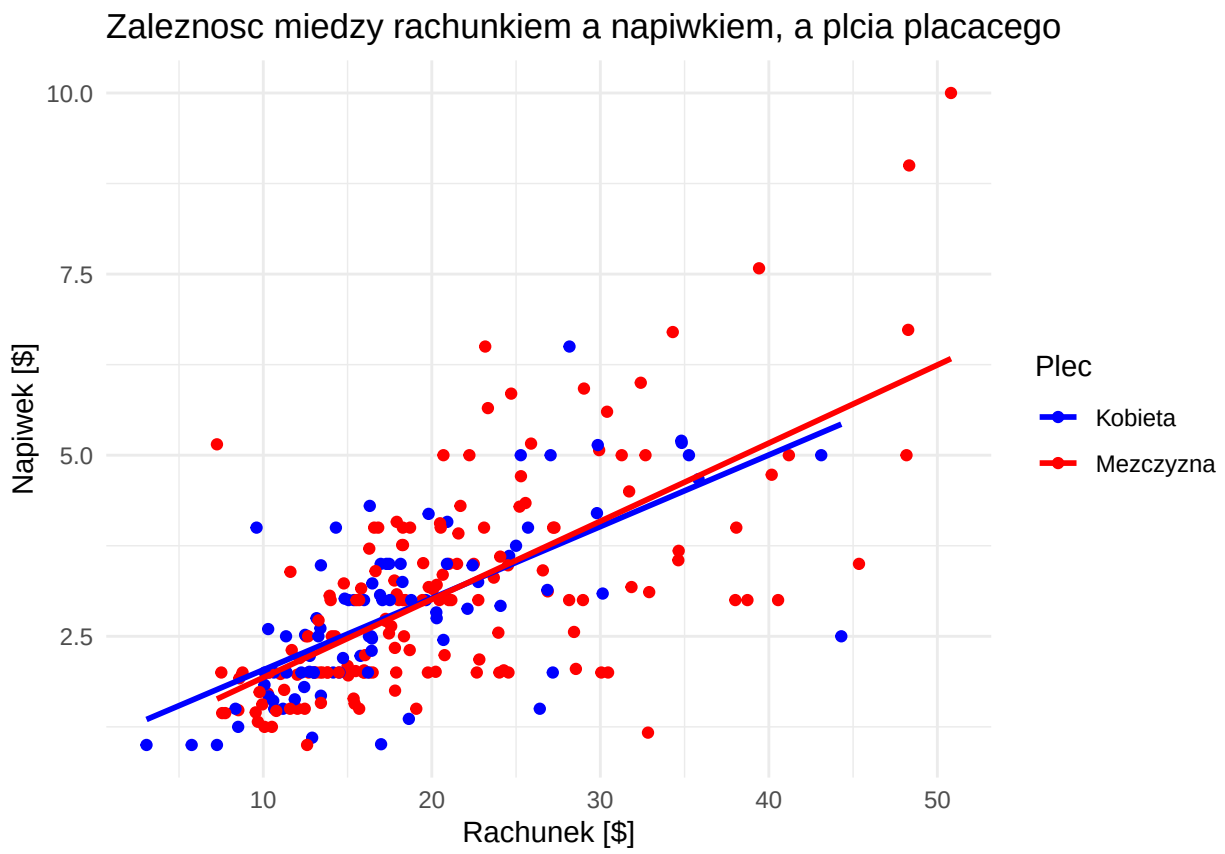
```
## Współczynnik Korelacji Spearmana rachunku do napiwku (dla kobiet): 0.6950734
```

```r
# Calculate the correlation for male clients
correlation_male <- cor(tips$total_bill[tips$sex == "Male"], tips$tip[tips$sex == "Male"], method = "sp
cat("Współczynnik Korelacji Spearmana rachunku do napiwku (dla mężczyzn):", correlation_male, "\n")
```

```
## Współczynnik Korelacji Spearmana rachunku do napiwku (dla mężczyzn): 0.6710884
```

```r
# Create the plot with separate trend lines for male and female
plot <- ggplot(polish_tips, aes(x = total_bill, y = tip, color = sex)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, aes(linetype = sex)) + # Add separate trend lines
  labs(
    title = "Zależność między rachunkiem a napiwkiem, a płcią płacącego",
    x = "Rachunek [$]",
    y = "Napiwek [$]"
  ) +
  scale_color_manual(
    name = "Płeć", # Custom legend title
    values = c("Kobieta" = "blue", "Mężczyzna" = "red") # Custom colors
  ) +
  scale_linetype_manual(
    name = "Płeć", # Custom legend title for line types
    values = c("Kobieta" = "solid", "Mężczyzna" = "solid") # Custom line types
  ) +
  theme_minimal()
```

```r
# Render Plotly for HTML and static ggplot for PDF
if (knitr::is_html_output()) {
  ggplotly(plot)
} else {
  print(plot)
}
```




```r
# Perform a t-test to compare tips given by men and women
t_test_result <- t.test(tip ~ sex, data = tips, alternative = "greater")

# Display the t-test result
cat("T-test result:\n")
```

```
## T-test result:
```

```r
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  tip by sex
## t = -1.4895, df = 215.71, p-value = 0.9311
## alternative hypothesis: true difference in means between group Female and group Male is greater than
## 95 percent confidence interval:
##  -0.5402706        Inf
## sample estimates:
## mean in group Female   mean in group Male
```

```
##             2.833448             3.089618
```

ĆWICZENIE NR 4: Zestaw danych z binarną zmienną zależną i niezależną

```r
# Create a binary dataset
binary_data <- tips %>%
  mutate(
    high_tip = ifelse(tip > median(tip), 1, 0), # Binary dependent variable
    weekend = ifelse(day %in% c("Sat", "Sun"), 1, 0) # Binary independent variable
  )

# Display the structure of the binary dataset
str(binary_data)
```

```
## tibble [244 x 9] (S3: tbl_df/tbl/data.frame)
##  $ total_bill: num [1:244] 17 10.3 21 23.7 24.6 ...
##  $ tip       : num [1:244] 1.01 1.66 3.5 3.31 3.61 4.71 2 3.12 1.96 3.23 ...
##  $ sex       : chr [1:244] "Female" "Male" "Male" "Male" ...
##  $ smoker    : chr [1:244] "No" "No" "No" "No" ...
##  $ day       : chr [1:244] "Sun" "Sun" "Sun" "Sun" ...
##  $ time      : chr [1:244] "Dinner" "Dinner" "Dinner" "Dinner" ...
##  $ size      : num [1:244] 2 3 3 2 4 4 2 4 2 2 ...
##  $ high_tip  : num [1:244] 0 0 1 1 1 1 0 1 0 1 ...
##  $ weekend   : num [1:244] 1 1 1 1 1 1 1 1 1 1 ...
```

```r
# Example analysis: Proportion of high tips on weekends vs weekdays
binary_summary <- binary_data %>%
  group_by(weekend) %>%
  summarise(
    proportion_high_tips = mean(high_tip)
  )

binary_summary
```

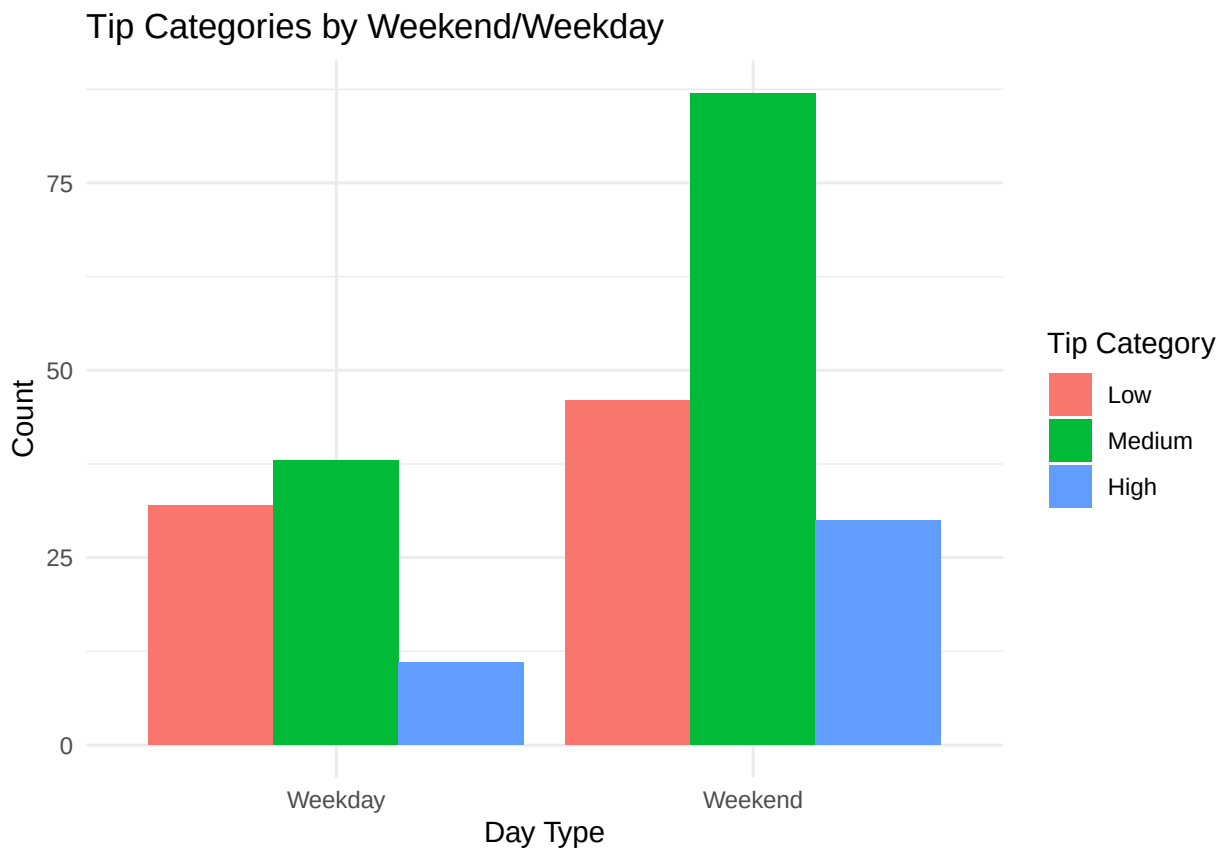
```
## # A tibble: 2 x 2
##   weekend proportion_high_tips
##     <dbl>                <dbl>
## 1       0                0.407
## 2       1                0.546
```

ĆWICZENIE NR 5: Binary vs. Categorical

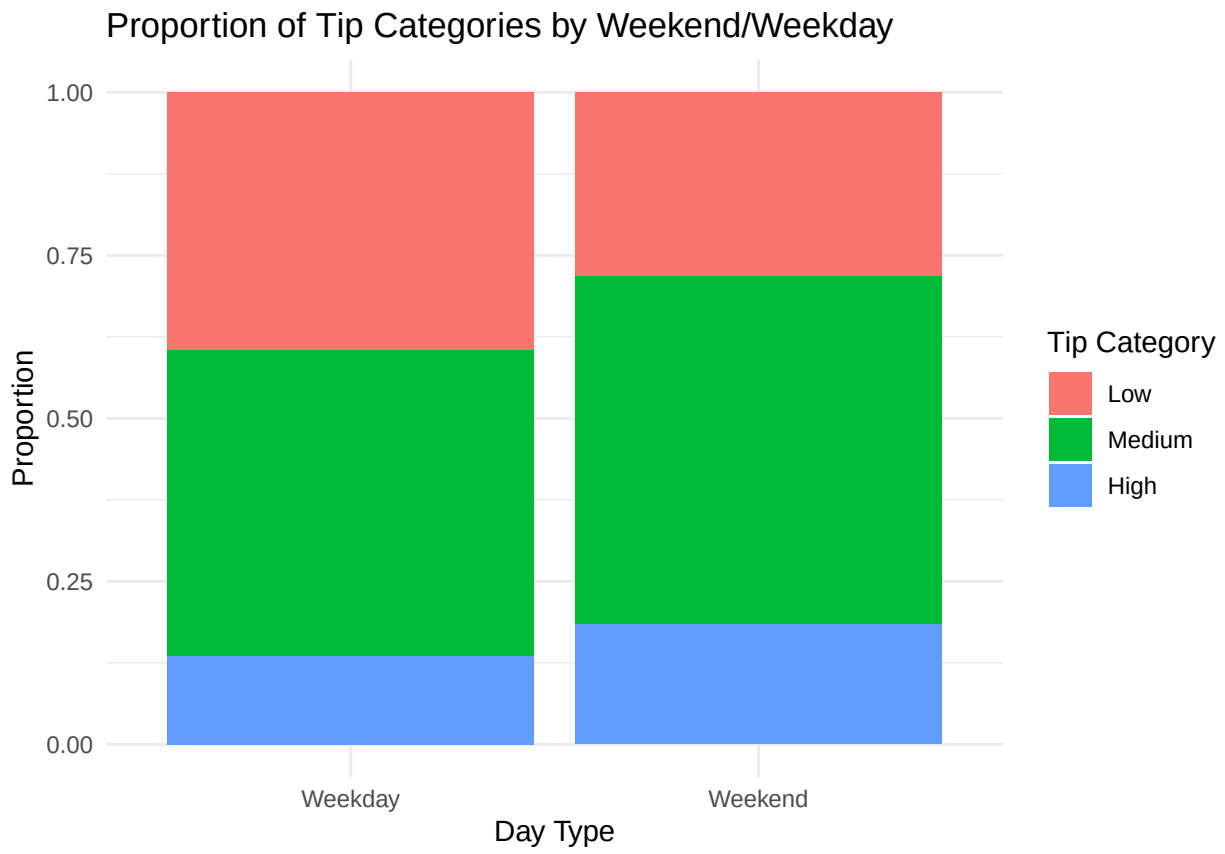
```r
# Create a dataset with a binary independent variable and a categorical dependent variable
binary_categorical_data <- tips %>%
  mutate(
    weekend = ifelse(day %in% c("Sat", "Sun"), "Weekend", "Weekday"), # Binary independent variable
    tip_category = cut(tip, breaks = c(0, 2, 4, Inf), labels = c("Low", "Medium", "High")) # Categorica
  )

# Visualization 1: Bar plot
ggplot(binary_categorical_data, aes(x = weekend, fill = tip_category)) +
  geom_bar(position = "dodge") +
```

```
  labs(title = "Tip Categories by Weekend/Weekday", x = "Day Type", y = "Count", fill = "Tip Category")
  theme_minimal()
```



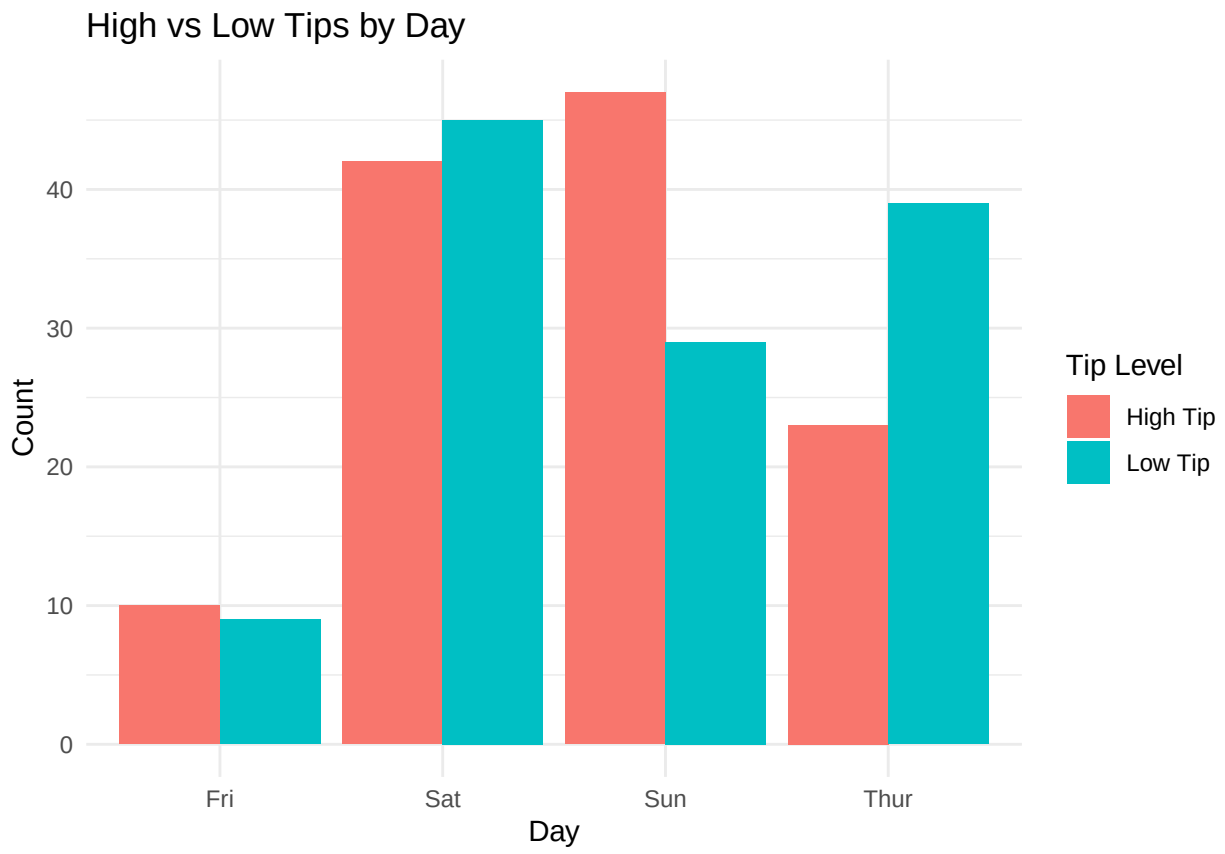

```
# Visualization 2: Stacked bar plot
ggplot(binary_categorical_data, aes(x = weekend, fill = tip_category)) +
  geom_bar(position = "fill") +
  labs(title = "Proportion of Tip Categories by Weekend/Weekday", x = "Day Type", y = "Proportion", fil
  theme_minimal()
```



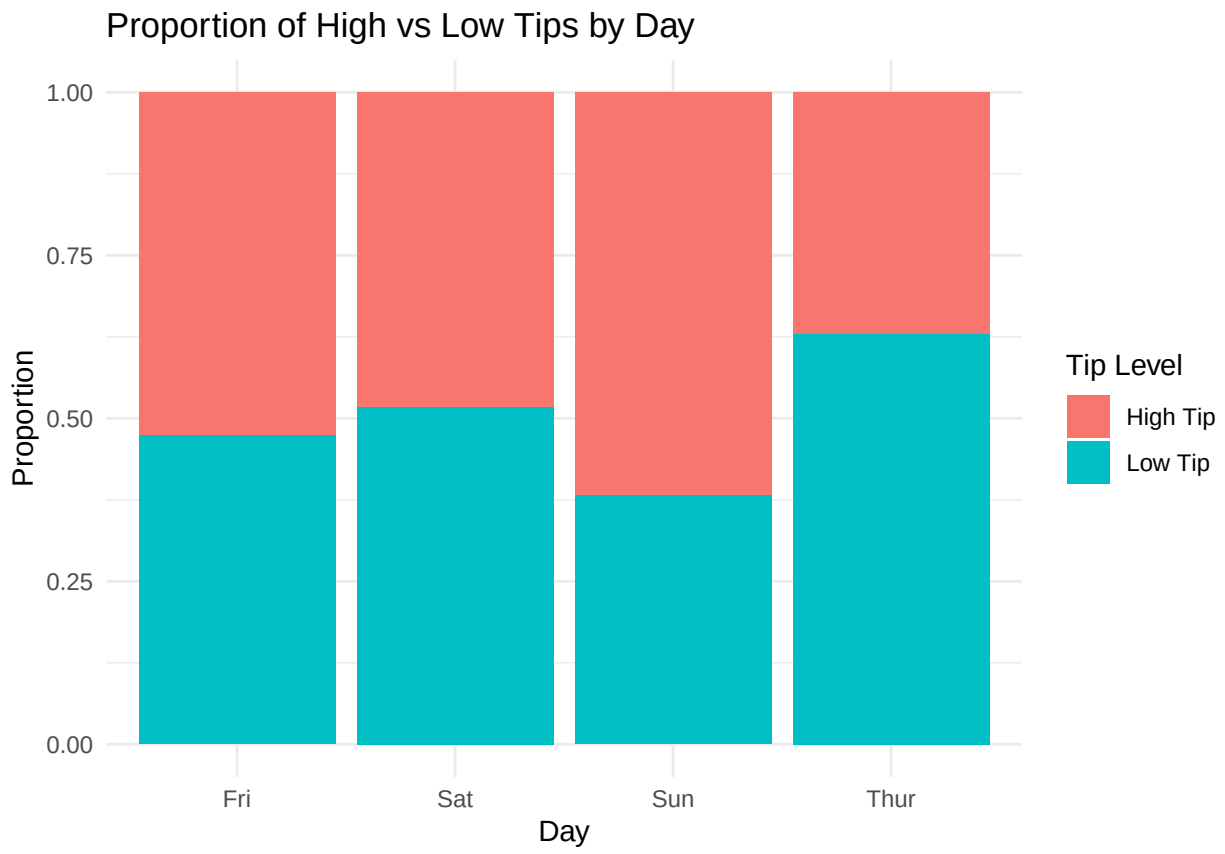
ĆWICZENIE NR 6: Categorical vs. Binary

```r
# Create a dataset with a categorical independent variable and a binary dependent variable
categorical_binary_data <- tips %>%
  mutate(
    high_tip = ifelse(tip > median(tip), "High Tip", "Low Tip") # Binary dependent variable
  )

# Visualization 1: Grouped bar plot
ggplot(categorical_binary_data, aes(x = day, fill = high_tip)) +
  geom_bar(position = "dodge") +
  labs(title = "High vs Low Tips by Day", x = "Day", y = "Count", fill = "Tip Level") +
  theme_minimal()
```



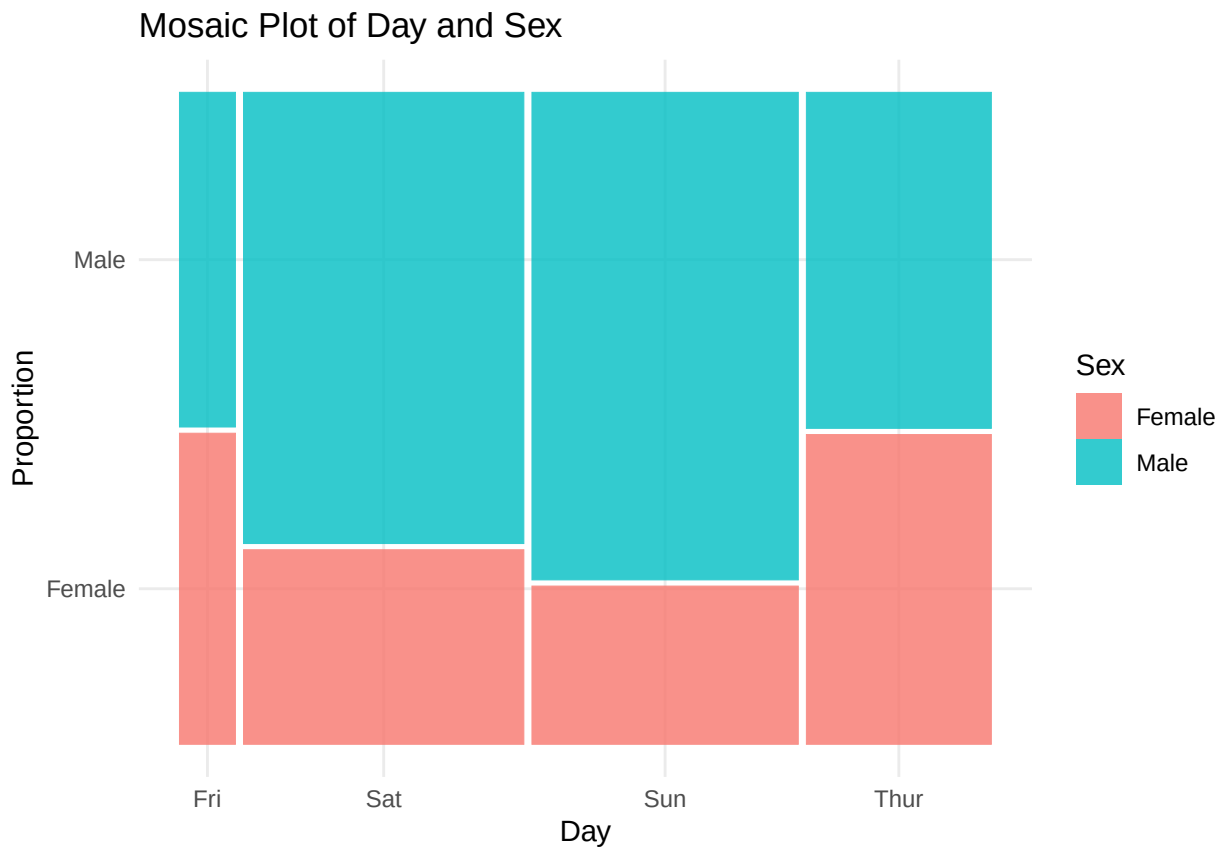

```
# Visualization 2: Proportional bar plot
ggplot(categorical_binary_data, aes(x = day, fill = high_tip)) +
  geom_bar(position = "fill") +
  labs(title = "Proportion of High vs Low Tips by Day", x = "Day", y = "Proportion", fill = "Tip Level")
  theme_minimal()
```



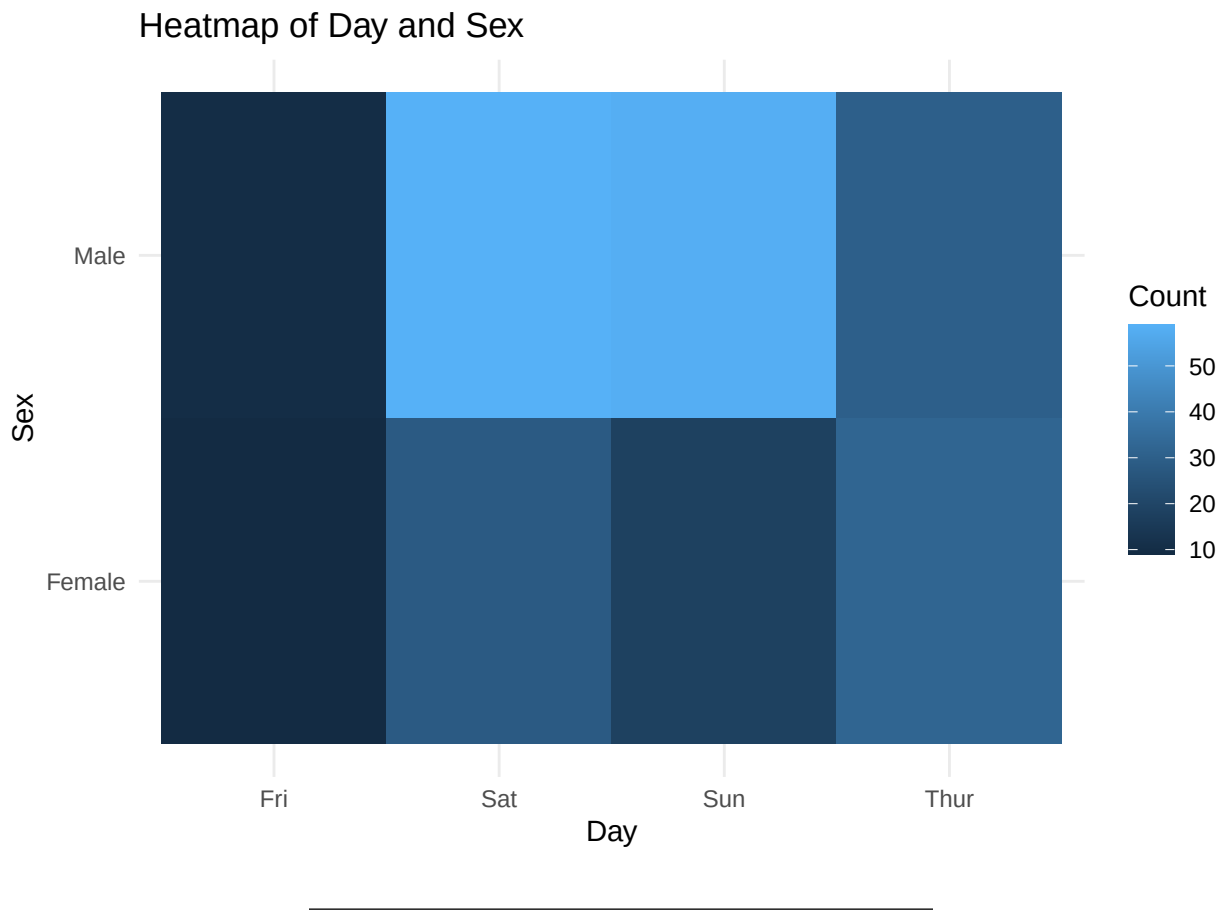
ĆWICZENIE NR 7: Categorical vs. Categorical

```r
# Create a dataset with two categorical variables
categorical_categorical_data <- tips

# Visualization 1: Mosaic plot
library(ggmosaic)
ggplot(data = categorical_categorical_data) +
  geom_mosaic(aes(weight = tip, x = product(day), fill = sex)) +
  labs(title = "Mosaic Plot of Day and Sex", x = "Day", y = "Proportion", fill = "Sex") +
  theme_minimal()
```




```
# Visualization 2: Heatmap
ggplot(categorical_categorical_data, aes(x = day, y = sex, fill = ..count..)) +
  geom_bin2d() +
  labs(title = "Heatmap of Day and Sex", x = "Day", y = "Sex", fill = "Count") +
  theme_minimal()
```


Heatmap of Day and Sex
Male
Female
Sex
Fri
Sat
Sun
Thur
Day
Count
50
40
30
20
10