Homework1.Rmd

2024-10-10

List of required packages

packages <- c(“rlang”, “ggplot2”, “tidyverse”, “palmerpenguins”, “rmarkdown”, “tinytex”, “xtable”, “patchwork”, “gridExtra”, “dplyr”, “tidyr”)

Install missing packages and load them

for (pkg in packages) { if (!require(pkg, character.only = TRUE)) { install.packages(pkg) library(pkg, character.only = TRUE) } else { library(pkg, character.only = TRUE) } }

Task 1 - Data Frame

Load the mpg dataset as a Data Frame

df <- as.data.frame(mpg)

Check the Data Frame

head(df)

Create a frequency table for the ‘drv’ variable

freq_table <- as.data.frame(table(mpg$drv))

Calculate relative frequency and percentage

freq_table$rel_Freq <- round(freq_table$Freq / sum(freq_table$Freq), 2) freq_table$Percentage <- round(freq_table$rel_Freq * 100, 200)

Rename the columns for clarity

colnames(freq_table) <- c(“drv”, “Freq”, “rel_Freq”, “Percentage”)

Print the frequency table

print(freq_table)

Create a bar chart

ggplot(freq_table, aes(x = drv, y = Freq, fill = drv)) + geom_bar(stat = “identity”)

Create a pie chart

ggplot(freq_table, aes(x = “ “, y = Freq, fill = drv)) + geom_bar(stat =”identity”, width = 1) + coord_polar(theta = “y”) + scale_fill_manual(values = c(“4” = “red”, “f” = “green”, “r” = “blue”)) + # Custom colors labs(title = “Distribution of Drive Types”, fill = “Drive Type (drv)”) + theme_minimal()

Simple histogram with x-axis starting at 10

ggplot(data = mpg, aes(x = cty, y = hwy)) + geom_point()

ggplot(df, aes(x = hwy)) + geom_histogram(breaks = c(10, 15, 20, 25, 30, 35, 40, 45), fill = “grey”, color = “black”) + labs(title = “Histogram of Highway Mileage”, x = “Highway”, y = “Frequency”) + theme_classic(base_size = 15)

Histogram with x-axis labeled from 15 to 45 in steps of 5

ggplot(data = mpg, aes(x = cty, y = hwy)) + geom_point() ggplot(df, aes(x = hwy)) + geom_histogram(binwidth = 2, fill = “grey”, color = “black”) + labs(title = “Histogram of Highway Mileage”, x = “Highway”, y = “Frequency”) + scale_x_continuous(breaks = seq(15, 45, by = 5)) + theme_classic(base_size = 15)

Simple Boxplot

ggplot(data = mpg, aes(x = cty, y = hwy)) + geom_point()

ggplot(df, aes(x = hwy)) + geom_boxplot(fill = “grey”, color = “black”, outlier.shape = 21, outlier.fill = “white”, outlier.color = “black”, linetype = “solid”) + scale_x_continuous(breaks = seq(15, 45, by = 5), limits = c(10, 45), name = “Highway”) + theme_minimal() + theme(panel.border = element_rect(color = “black”, fill = NA), # Black border around the plot axis.line.x = element_line(color = “black”), # x-axis line axis.ticks.x = element_line(color = “black”), # x-axis ticks axis.text.y = element_blank(), # Remove y-axis labels panel.grid = element_blank() # Remove grid lines)

Scatter plot with adjusted y-axis limits, no grid, and white points with black borders

ggplot(data = mpg, aes(x = cty, y = hwy)) + geom_point()

ggplot(df, aes(x = displ, y = hwy)) + geom_point(color = “black”, fill = “white”, shape = 21) + # White points with black borders labs(x = “displ”, y = “Highway Mileage (mpg)”) + scale_y_continuous(breaks

= seq(15, 45, by = 5), limits = c(0, NA), expand = c(0, 0)) + # y-axis starts at 0, labels start at 15, no extra padding theme_minimal() + theme(panel.border = element_rect(color = “black”, fill = NA), # Black border around the plot panel.grid = element_blank() # Remove grid lines)

Horizontal boxplot showing the association between highway mileage (hwy) and drive system (drv)

ggplot(data = mpg, aes(x = cty, y = hwy)) + geom_point()

ggplot(df, aes(y = drv, x = hwy)) + geom_boxplot(fill = “grey”, color = “black”, outlier.shape = 21, outlier.fill = “white”, outlier.color = “black”) + labs(y = “Drive System (drv)”, x = “Highway Mileage (mpg)”) + scale_x_continuous(breaks = seq(15, 45, by = 5), limits = c(10, 45)) + # x-axis with labels from 15 to 45 theme_minimal() + theme(panel.border = element_rect(color = “black”, fill = NA), # Black border around the plot panel.grid = element_blank() # Remove grid lines)

Stacked bar chart showing the distribution of vehicle classes within each drive type

ggplot(data = mpg, aes(x = cty, y = hwy)) + geom_point() ggplot(df, aes(x = drv, fill = class)) + geom_bar(position = “stack”, color = “black”) + # Stacked bar chart with black borders labs(x = “Drive System (drv)”, y = “Count”, fill = “Vehicle Class”) + scale_fill_manual(values = c(“suv” = “magenta”, “subcompact” = “purple”, “pickup” = “blue”, “minivan” = “cyan”, “midsize” = “limegreen”, “compact” = “yellow”, “2seater” = “red”)) + # Custom colors theme_minimal() + theme(panel.border = element_rect(color = “black”, fill = NA), # Black border around the plot panel.grid = element_blank() # Remove grid lines)

Clustered bar chart showing association between drv and class

df_complete <- as.data.frame(table(mpgdrv, mpgclass)) colnames(df_complete) <- c(“drv”, “class”, “n”)

ggplot(df_complete, aes(x = drv, y = n, fill = class)) + geom_bar(stat = “identity”, position = “dodge”, color = “black”) + # Side-by-side bars with black borders labs(title = “Association between categorical variables”, x = “Drive System (drv)”, y = “Count”, fill = “Vehicle Class”) + scale_fill_manual(values = c(“2seater” = “red”, “compact” = “yellow”, “midsize” = “limegreen”, “minivan” = “cyan”, “pickup” = “blue”, “subcompact” = “purple”, “suv” = “magenta”)) + # Custom colors theme_minimal() + theme(plot.title = element_text(color = “blue”, size = 16, hjust = 0.5), # Blue title, centered panel.border = element_rect(color = “black”, fill = NA), # Black border around the plot panel.grid = element_blank() # Remove grid lines)

Stacked bar chart with proportions

df_proportions <- df_complete df_proportions$proportion < −df_proportions$n / ave(df_proportions$n, df_proportions$drv, FUN = sum)

ggplot(df_proportions, aes(x = drv, y = proportion, fill = class)) + geom_bar(stat = “identity”, position = “fill”, color = “black”) + # Stacked bar chart with black borders labs(x = “Drive System (drv)”, y = “Proportion”, fill = “Vehicle Class”) + scale_y_continuous(labels = scales::percent) + # y-axis as percentages scale_fill_manual(values = c(“2seater” = “red”, “compact” = “yellow”, “midsize” = “limegreen”,

“minivan” = “cyan”, “pickup” = “blue”, “subcompact” = “purple”, “suv” = “magenta”)) + # Custom colors theme_minimal() + theme(panel.border = element_rect(color = “black”, fill = NA), # Black border around the plot panel.grid = element_blank() # Remove grid lines)

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plo

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plo

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distributions: mosaic plot

Es fehlt Visualizing conditional distributions und Joint distribu-

Points sized for better visibility

geom_smooth(method = “lm”, linetype = “dashed”, color = “black”, se = FALSE) + # Line of best fit labs(title = “Association between Engine Displacement and Highway Mileage”, x = “Engine Displacement (L)”, y = “Highway Mileage (mpg)”, shape = “Fuel Type”, color = “Vehicle Class”) + theme_minimal()

Task 3: Comparison of geom_point() and geom_count()

Create geom_point plot

point_plot <- ggplot(data = mpg, aes(x = cty, y = hwy)) + geom_point() + labs(title = “Scatterplot using geom_point”, x = “City MPG”, y = “Highway MPG”)

Create geom_count plot

count_plot <- ggplot(data = mpg, aes(x = cty, y = hwy)) + geom_count() + labs(title = “Count Plot using geom_count”, x = “City MPG”, y = “Highway MPG”)

Display the plots side by side

grid.arrange(point_plot, count_plot, ncol = 2)

Add Interpretation

Describe the difference: The geom_count() plot shows larger points where there are multiple data points overlapping, whereas geom_point() displays each observation individually.

Task 4

Load the penguins dataset

data(penguins)

Initial bar plot for proportion

ggplot(data = penguins, aes(fill = island, x = species)) + geom_bar(aes(y = after_stat(prop))) + labs(title = “Proportion of Species by Island”, x = “Penguin Species”, y = “Proportion”)

Absolute counts with labels

ggplot(data = penguins, aes(fill = island, x = species)) + geom_bar(position = “stack”) + geom_text(stat = “count”, aes(label = ..count..), vjust = -0.5) + labs(title = “Counts of Species by Island”, x = “Penguin Species”, y = “Count”) + theme_minimal()

Add Interpretation

Explain that the improved version shows the absolute counts for each species and the labels make the exact values visible, improving readability.

Task 5

Load the diamonds dataset

data(diamonds)

Basic bar chart

bar_plot <- ggplot(diamonds, aes(x = cut, fill = clarity)) + geom_bar(position = “dodge”) + labs(title = “Bar Chart of Cut and Clarity”, x = “Cut”, y = “Count”) + theme_minimal()

Stacked bar chart

stacked_bar_plot <- ggplot(diamonds, aes(x = cut, fill = clarity)) + geom_bar(position = “stack”) + labs(title = “Stacked Bar Chart of Cut and Clarity”, x = “Cut”, y = “Count”) + theme_minimal()

Pie chart

pie_chart <- ggplot(diamonds, aes(x = “ “, fill = clarity)) + geom_bar(width = 1, position =”fill”) + coord_polar(“y”) + facet_wrap(~cut) + labs(title = “Pie Chart of Cut and Clarity”) + theme_minimal() + theme(axis.title.x = element_blank(), axis.title.y = element_blank())

Display the charts side by side

grid.arrange(bar_plot, stacked_bar_plot, pie_chart, ncol = 3)

Add Interpretation

Describe the advantages and disadvantages of each chart

The bar chart allows easy comparison of clarity within each cut.

The stacked bar chart provides an overview of the total while showing the internal distribution.

The pie chart illustrates proportions but may be harder to read compared to bar charts.

#stand 27.10 Aufgabe 1 fast alles gemacht ausser die zweiten letzten Seiten mosaic plot # Visualizing condition. Das hat mega viel arbeit gebeben und mann könnte immer noch alles verbessern, schaue doch mal drüber und verbessere je nach dem noch etwas. Aufgabe zwei und drei habe ich mithilfe von chatgpt gemacht, habe aber wirklich gar keine ahnung ob das irgendwie richtig ist, das unbedingt noch verbessern.