```
R version 4.4.2 (2024-10-31 ucrt) -- "Pile of Leaves"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> # Load required packages
> library(ggplot2)
Want to understand how all the pieces fit together? Read R for Data
Science: https://r4ds.hadley.nz/
> library(GGally)
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
> library(dplyr)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union

>
> # Load datasets
> data(iris)
> data("PlantGrowth")
>
> # 1a. Make a histogram of the variable Sepal.Width
> ggplot(iris, aes(x = Sepal.Width)) +
+   geom_histogram(binwidth = 0.2, fill = "blue", color = "black", alpha = 0.7) +
+   ggtitle("Histogram of Sepal Width") +
+   xlab("Sepal Width (cm)") +
+   ylab("Frequency")
>
> # 1b. Based on the histogram, which would be higher: mean or median? Why?
> # (Visual inspection of skewness can guide this answer)
>
> # 1c. Confirm the mean and median values
> mean_sepal_width <- mean(iris$Sepal.Width)
> median_sepal_width <- median(iris$Sepal.Width)
> print(paste("Mean Sepal Width:", mean_sepal_width))
[1] "Mean Sepal Width: 3.05733333333333"
> print(paste("Median Sepal Width:", median_sepal_width))
[1] "Median Sepal Width: 3"
>
> # 1d. Find the Sepal.Width value for which only 27% of flowers have a higher value
> threshold_27 <- quantile(iris$Sepal.Width, probs = 0.73)  # 73% below, 27% above
> print(paste("Only 27% of flowers have a Sepal.Width higher than", threshold_27, "cm"))
[1] "Only 27% of flowers have a Sepal.Width higher than 3.3 cm"
>
> # 1e. Scatterplots of each pair of numerical variables in iris
> ggpairs(iris, aes(color = Species)) +
+   ggtitle("Scatterplot Matrix of Iris Dataset Variables")
 plot: [1, 1] [=>------------------------------------------------]  4% est: 0s  plot: [1, 2] [==
=>--------------------------------------------]  8% est: 2s  plot: [1, 3] [=====>-------------
----------------------------] 12% est: 2s  plot: [1, 4] [=======>-------------------------------
-----------] 16% est: 2s  plot: [1, 5] [=========>----------------------------------------] 20% e
st: 2s  plot: [2, 1] [===========>--------------------------------------] 24% est: 2s  plot: [2,
 2] [=============>------------------------------------] 28% est: 2s  plot: [2, 3] [=============
==>----------------------------------] 32% est: 2s  plot: [2, 4] [=================>--------------
------------------] 36% est: 2s  plot: [2, 5] [===================>------------------------------
] 40% est: 2s  plot: [3, 1] [=====================>----------------------------] 44% est: 1s  pl
ot: [3, 2] [=======================>--------------------------] 48% est: 1s  plot: [3, 3] [======
===================>------------------------] 52% est: 1s  plot: [3, 4] [=========================
==>----------------------] 56% est: 1s  plot: [3, 5] [=============================>-------------
-------] 60% est: 1s  plot: [4, 1] [===============================>------------------] 64% est:
1s  plot: [4, 2] [=================================>----------------] 68% est: 1s  plot: [4, 3]
[===================================>--------------] 72% est: 1s  plot: [4, 4] [==================
===================>------------] 76% est: 1s  plot: [4, 5] [====================================
===>----------] 80% est: 0s  plot: [5, 1] [=========================================>--------] 84
% est: 0s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [5, 2] [===========================================>------] 88% est: 0s `stat_bin()` using
 `bins = 30`. Pick better value with `binwidth`.
 plot: [5, 3] [=============================================>----] 92% est: 0s `stat_bin()` using
 `bins = 30`. Pick better value with `binwidth`.
 plot: [5, 4] [===============================================>--] 96% est: 0s `stat_bin()` using
 `bins = 30`. Pick better value with `binwidth`.
 plot: [5, 5] [==================================================]100% est: 0s
                                                             >
> # 1f. Identify strongest and weakest relationships
> # (Observing the scatterplots helps determine correlation strengths)
>
> # ---------------------------
> # 2a. Histogram of PlantGrowth weight with breakpoints at every 0.3 from 3.3
> ggplot(PlantGrowth, aes(x = weight)) +
+   geom_histogram(breaks = seq(3.3, max(PlantGrowth$weight), by = 0.3),
+                  fill = "green", color = "black", alpha = 0.7) +
+   ggtitle("Histogram of Plant Weights") +
+   xlab("Weight") +
+   ylab("Frequency")
>
> # 2b. Boxplots of weight separated by group
> ggplot(PlantGrowth, aes(x = group, y = weight, fill = group)) +
+   geom_boxplot() +
+   ggtitle("Boxplots of Weight by Group") +
+   xlab("Group") +
+   ylab("Weight") +
+   theme_minimal()
>
> # 2c. Estimate percentage of "trt1" weights below the minimum "trt2" weight
> min_trt2_weight <- min(PlantGrowth$weight[PlantGrowth$group == "trt2"])
> percentage_trt1_below_trt2 <- mean(PlantGrowth$weight[PlantGrowth$group == "trt1"] < min_trt2_w
eight) * 100
> print(paste("Estimated percentage of 'trt1' weights below min 'trt2' weight:", percentage_trt1_
below_trt2, "%"))
[1] "Estimated percentage of 'trt1' weights below min 'trt2' weight: 80 %"
>
> # 2d. Exact percentage of "trt1" weights below minimum "trt2" weight
> exact_percentage <- sum(PlantGrowth$weight[PlantGrowth$group == "trt1"] < min_trt2_weight) /
+                     length(PlantGrowth$weight[PlantGrowth$group == "trt1"]) * 100
> print(paste("Exact percentage of 'trt1' weights below min 'trt2' weight:", exact_percentage, "%
"))
[1] "Exact percentage of 'trt1' weights below min 'trt2' weight: 80 %"
>
> # 2e. Barplot of group for plants with weight above 5.5
> ggplot(PlantGrowth %>% filter(weight > 5.5), aes(x = group, fill = group)) +
+   geom_bar() +
+   scale_fill_manual(values = heat.colors(3)) +
+   ggtitle("Barplot of Group for Plants with Weight > 5.5") +
+   xlab("Group") +
+   ylab("Count") +
+   theme_minimal()
> # Load required packages
> library(ggplot2)
> library(GGally)
> library(dplyr)
> 
> # Load datasets
> data(iris)
> data("PlantGrowth")
> 
> # 1a. Make a histogram of the variable Sepal.Width
> ggplot(iris, aes(x = Sepal.Width)) +
+   geom_histogram(binwidth = 0.2, fill = "blue", color = "black", alpha = 0.7) +
+   ggtitle("Histogram of Sepal Width") +
+   xlab("Sepal Width (cm)") +
+   ylab("Frequency")
> 
> # 1b. Based on the histogram, which would be higher: mean or median? Why?
> # (Visual inspection of skewness can guide this answer)
> 
> # 1c. Confirm the mean and median values
> mean_sepal_width <- mean(iris$Sepal.Width)
> median_sepal_width <- median(iris$Sepal.Width)
> print(paste("Mean Sepal Width:", mean_sepal_width))
[1] "Mean Sepal Width: 3.05733333333333"
> print(paste("Median Sepal Width:", median_sepal_width))
[1] "Median Sepal Width: 3"
> 
> # 1d. Find the Sepal.Width value for which only 27% of flowers have a higher value
> threshold_27 <- quantile(iris$Sepal.Width, probs = 0.73)  # 73% below, 27% above
> print(paste("Only 27% of flowers have a Sepal.Width higher than", threshold_27, "cm"))
[1] "Only 27% of flowers have a Sepal.Width higher than 3.3 cm"
> 
> # 1e. Scatterplots of each pair of numerical variables in iris
> ggpairs(iris, aes(color = Species)) +
+   ggtitle("Scatterplot Matrix of Iris Dataset Variables")
 plot: [1, 1] [=>--------------------------------------------]  4% est: 0s  plot: [1, 2] [==
=>------------------------------------------]  8% est: 1s  plot: [1, 3] [=====>-------------
---------------------------] 12% est: 1s  plot: [1, 4] [======>-----------------------------
----------] 16% est: 1s  plot: [1, 5] [========>-------------------------------------] 20% e
st: 1s  plot: [2, 1] [==========>-----------------------------------] 24% est: 1s  plot: [2,
 2] [============>---------------------------------] 28% est: 1s  plot: [2, 3] [============
==>-------------------------------] 32% est: 1s  plot: [2, 4] [================>------------
-----------------] 36% est: 1s  plot: [2, 5] [=================>----------------------------
] 40% est: 1s  plot: [3, 1] [===================>--------------------------] 44% est: 1s  pl
ot: [3, 2] [=====================>------------------------] 48% est: 1s  plot: [3, 3] [=====
==================>----------------------] 52% est: 1s  plot: [3, 4] [======================
==>---------------------] 56% est: 1s  plot: [3, 5] [===========================>-----------
-------] 60% est: 1s  plot: [4, 1] [============================>-----------------] 64% est:
1s  plot: [4, 2] [==============================>---------------] 68% est: 1s  plot: [4, 3]
[================================>-------------] 72% est: 1s  plot: [4, 4] [================
==================>-----------] 76% est: 0s  plot: [4, 5] [=================================
===>---------] 80% est: 0s  plot: [5, 1] [======================================>-------] 84
% est: 0s `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
 plot: [5, 2] [=======================================>------] 88% est: 0s `stat_bin()` using
 `bins = 30`. Pick better value with `binwidth`.
 plot: [5, 3] [=========================================>----] 92% est: 0s `stat_bin()` using
 `bins = 30`. Pick better value with `binwidth`.
 plot: [5, 4] [===========================================>--] 96% est: 0s `stat_bin()` using
 `bins = 30`. Pick better value with `binwidth`.
 plot: [5, 5] [==============================================]100% est: 0s
                                                             > q()
> ggplot(PlantGrowth %>% filter(weight > 5.5), aes(x = group, fill = group)) +
+   geom_bar() +
+   scale_fill_manual(values = heat.colors(3)) +
+   ggtitle("Barplot of Group for Plants with Weight > 5.5") +
+   xlab("Group") +
+   ylab("Count") +
+   theme_minimal()
> save.image("C:\\Users\\olive\\OneDrive\\Desktop\\assignment consol")
```

```
>
```