Homework Assignment 02

Juan Cruz Ferreyra

2024-09-19

Section 1

a. Write a function that calculates the mean of any numeric vector you give it, without using the built-in mean() or sum() functions.

```
# Define custom sum function from scratch
custom_sum <- function(x) {
  result <- 0
  for (i in 1:length(x)) {
    result <- result + x[i]
  }

  return(result)
}

# Define custom mean function using custom sum function
custom_mean <- function(x) {
  return(custom_sum(x) / length(x))
}

v1 <- c(2, 4, 6, 8, 11)

print(paste0("The sum of the elements in the vector is: ", custom_sum(v1)))
```

```
## [1] "The sum of the elements in the vector is: 31"
```

```
print(paste0("The mean of the elements in the vector is: ", custom_mean(v1)))
```

```
## [1] "The mean of the elements in the vector is: 6.2"
```

b. Write a function that takes as its input a vector with four elements. If the sum of the first two elements is greater than the sum of the second two, the function returns the vector; otherwise it returns 0.

```r
# Define custom function
custom_function <- function(x) {
  if (length(x)!=4) {
    print(paste0("Vector should have 4 elements, ", length(x), " provided"))
    return()
  }

  if (sum(x[1:2]) > sum(x[3:4])) {
    return(x)
  }

  return(0)
}

print(custom_function(c(1, 2, 3, 4, 5)))
```

```
## [1] "Vector should have 4 elements, 5 provided"
## NULL
```

```r
print(custom_function(c(1, 2, 3, 4)))
```

```
## [1] 0
```

```r
print(custom_function(c(1, 4, 2, 3)))
```

```
## [1] 0
```

```r
print(custom_function(c(4, 3, 2, 1)))
```

```
## [1] 4 3 2 1
```

c. Write a function that calculates the Fibonacci sequence up to the nth element, where n is any number input into your function (its argument). The Fibonacci sequence is: 1, 1, 2, 3, 5, 8, 13, 21..., ie, each element is the sum of the previous two elements. One way to do this is to start off with the first two elements, c(1,1) and set an internal variable to this sequence. Then write a loop that counts up to n, where for each new element, you first calculate it by adding the last two elements of the growing sequence, and then stick that new number onto the growing sequence using c(). When the loop is finished, the function should return the final vector of Fibonacci numbers.

```r
# Define Fibonacci function
get_fibonacci <- function(x) {
  sequence <- c(1, 1)

  for (i in 3:x) {
    sequence <- c(sequence, sequence[i-2] + sequence[i-1])
  }

  return(sequence)
}

get_fibonacci(15)
```

```
##  [1]   1   1   2   3   5   8  13  21  34  55  89 144 233 377 610
```

d. Create a 4x4 matrix of the numbers 1 through 16. Use apply to apply your functions from (a) to each of the rows in your matrix.

```
# Define matrix
m1 <- matrix(1:16, 4, 4)

# Apply custom_sum and custom_mean functions
print(m1)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

```
print(paste0("Sums of each row of the matrix: ", paste0(apply(m1, 1, custom_sum), collapse = ", ")))
```

```
## [1] "Sums of each row of the matrix: 28, 32, 36, 40"
```

```
print(paste0("Means of each row of the matrix: ", paste0(apply(m1, 1, custom_mean), collapse = ", ")))
```

```
## [1] "Means of each row of the matrix: 7, 8, 9, 10"
```

Section 2

a. Using the airquality dataset, constuct an aggregated dataset which shows the maximum wind and ozone by month.

```
# Agreggate dataset
aggregate(cbind(Wind, Ozone) ~ Month, data=airquality, max)
```

Month	Wind	Ozone
5	20.1	115
6	20.7	71
7	14.9	135
8	15.5	168
9	16.6	96

b. Create the authors and books datasets following the example and data in the lecture, and then create a new data set by merging these two datasets by author, preserving all rows.

```r
# Create dataframes as in the lecture
authors <- data.frame(
    surname = c("Tukey", "Venables", "Tierney", "Ripley", "McNeil"),
    nationality = c("US", "Australia", "US", "UK", "Australia"),
    stringsAsFactors=FALSE)

books <- data.frame(
    name = c("Tukey", "Venables", "Tierney",
             "Ripley", "Ripley", "McNeil", "R Core"),
    title = c("Exploratory Data Analysis",
              "Modern Applied Statistics ...",
              "LISP-STAT",
              "Spatial Statistics", "Stochastic Simulation",
              "Interactive Data Analysis",
              "An Introduction to R"),
    stringsAsFactors=FALSE)
```

```r
# Merge dataframes keeping all the columns from both
authors_and_books <- merge(authors, books, by.x="surname", by.y="name", all.x=TRUE, all.y=TRUE)
authors_and_books
```

surname	nationality	title
McNeil	Australia	Interactive Data Analysis
R Core	NA	An Introduction to R
Ripley	UK	Spatial Statistics
Ripley	UK	Stochastic Simulation
Tierney	US	LISP-STAT
Tukey	US	Exploratory Data Analysis
Venables	Australia	Modern Applied Statistics ...

c. Take the following string and replace every instance of "to" or "To" with "2":

To be, or not to be – that is the question: Whether 'tis nobler in the mind to suffer The slings and arrows of outrageous fortune, Or to take arms against a sea of troubles, And by opposing end them. To die – to sleep – No more...

```r
original_txt <- "To be, or not to be -- that is the question:
Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles,
And by opposing end them. To die -- to sleep --
No more..."

# Change "To" or "to" strings to "2"
process_txt <- gsub("to", "2", original_txt, ignore.case = TRUE)

cat(process_txt)
```

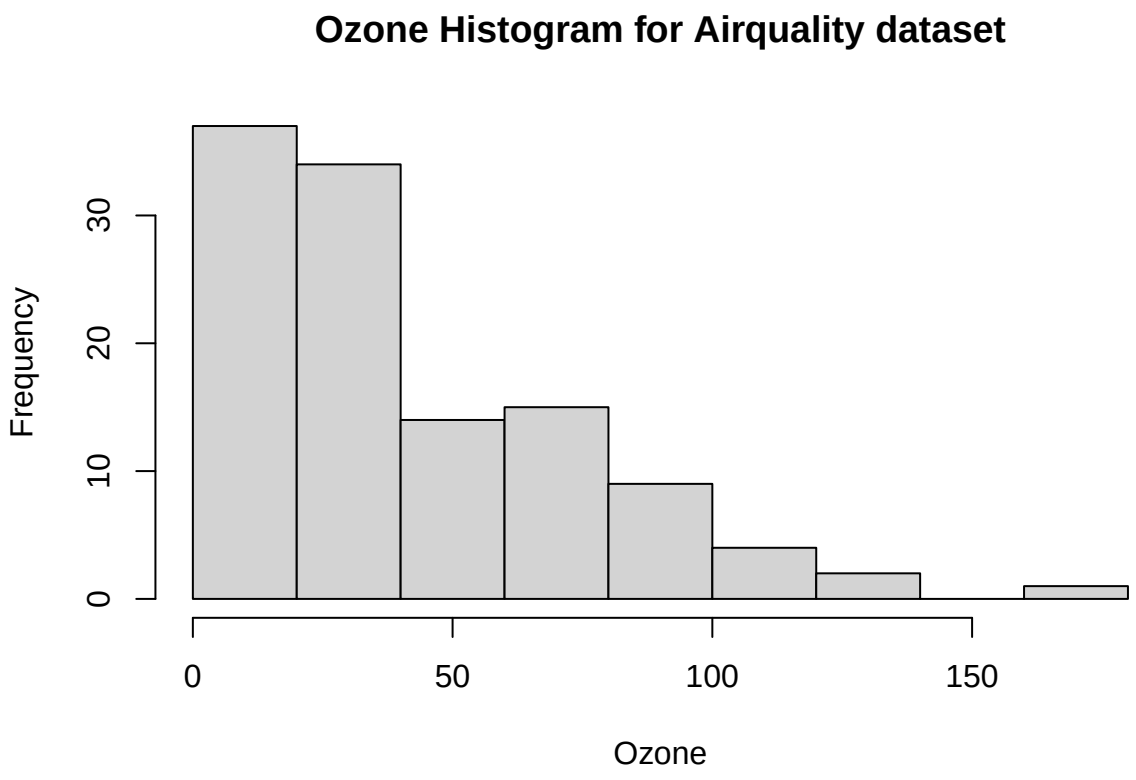
```
## 2 be, or not 2 be -- that is the question:
## Whether 'tis nobler in the mind 2 suffer
## The slings and arrows of outrageous fortune,
```

```
## Or 2 take arms against a sea of troubles,
## And by opposing end them. 2 die -- 2 sleep --
## No more...
```

Section 3

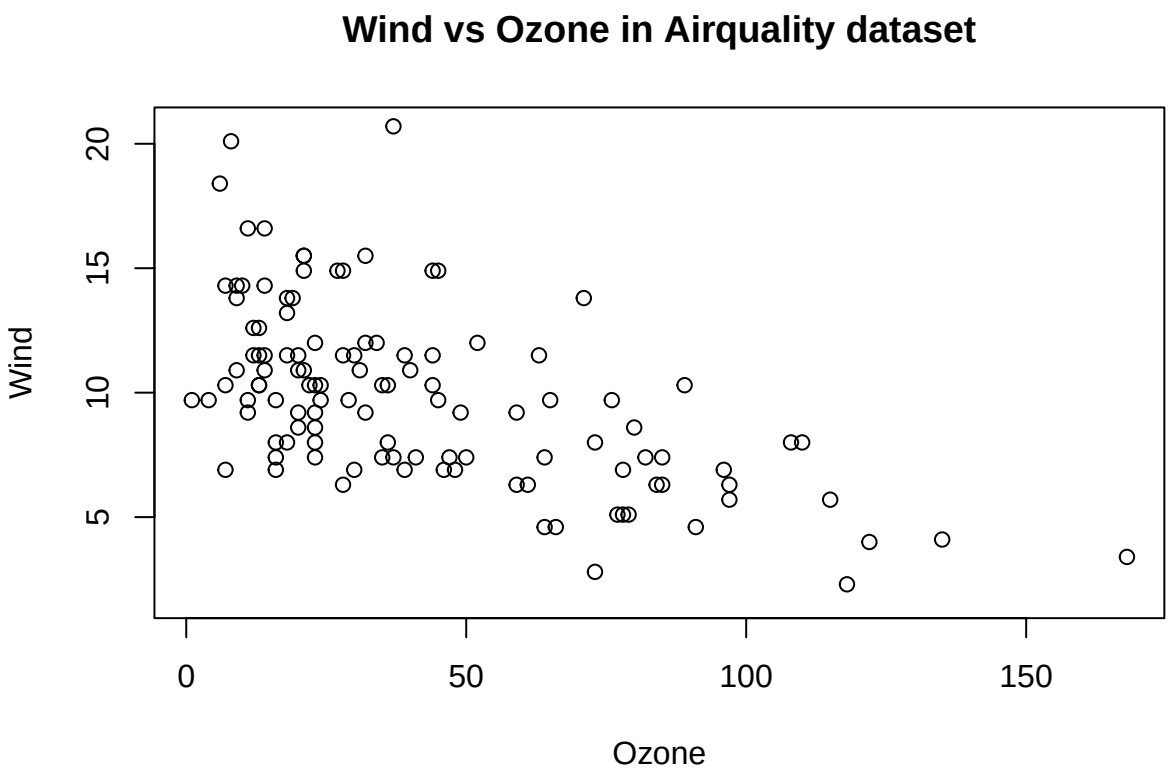
a. Create a histogram using the base R graphics using some dataset or variable other than the one in the lessons. Always make sure your graph has well-labeled x and y axes and an explanatory title.

```
# Plot histogram with base R
hist(airquality$Ozone, main="Ozone Histogram for Airquality dataset", xlab="Ozone")
```



b. Create a scatter plot using the base R graphics, again with some variable other than the one in the lessons.

```
# Plot scatterplot with base R
plot(airquality$Ozone, airquality$Wind, xlab="Ozone", ylab="Wind", main="Wind vs Ozone in Airquality da
```



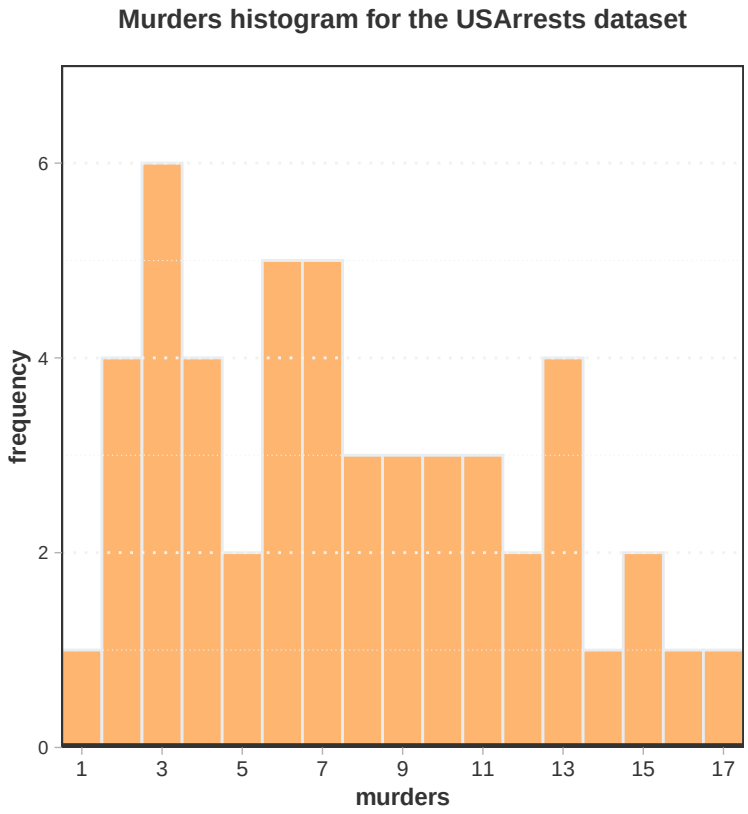
Load libraries

```
library(ggplot2)
library(tidyverse)
```

c. Create a histogram using ggplot, using some new data. In this and the later plots, please tinker with the settings using the examples in http://www.cookbook-r.com/Graphs/ to make it prettier.

```
# Create the histogram
ggplot(USArrests, aes(x=Murder)) +
  geom_histogram(binwidth=1, fill="#fdae61", color="#e9ecef", alpha=0.9) +
  scale_y_continuous(
    expand=c(0,0,0,0),
    limits=c(0, 7),
    breaks=seq(0, 10, by = 2),
    labels=seq(0, 10, by = 2)
  ) +
  scale_x_continuous(
    expand=c(0,0,0,0),
    breaks=seq(1, max(USArrests$Murder, na.rm=TRUE), by = 2)
  ) +
  geom_hline(yintercept=seq(1, 7, by = 2), color="gray95", linetype="dotted", linewidth=.2) +
  geom_hline(yintercept=seq(0, 7, by = 2), color="gray95", linetype="dotted", linewidth=.5) +
  geom_hline(yintercept=c(0), color="gray20", linewidth=1.5) +
  labs(title="Murders histogram for the USArrests dataset",
```

```
      fill=NULL,
      y="frequency",
      x="murders",
      caption=NULL) +
theme_light() +
theme(plot.margin=margin(20, 0, 10, 0),
       aspect.ratio=1,
       panel.border=element_rect(colour="gray20"),
       panel.grid.minor.x=element_blank(),
       panel.grid.major.x=element_blank(),
       panel.grid.minor.y=element_blank(),
       panel.grid.major.y=element_blank(),
       panel.background=element_blank(),
       plot.title=element_text(size=10,
                               face="bold",
                               hjust=.5,
                               vjust=4,
                               colour="gray20"),
       axis.title.x=element_text(size=8.5,
                                 colour="gray20",
                                 face="bold"),
       axis.text.x=element_text(size=7.5,
                                colour="gray20",
                                angle=0),
       axis.title.y=element_text(size=8.5,
                                 colour="gray20",
                                 face="bold"),
       axis.text.y=element_text(size=7,
                                colour="gray20"),
       legend.position="none"
)
```



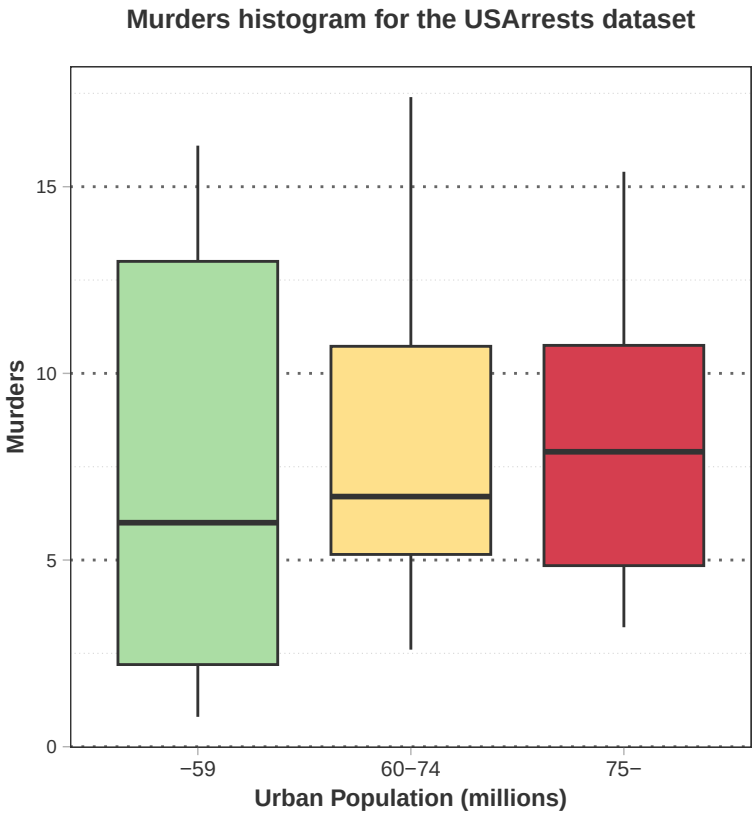
d. Create a box plot (with multiple categories) using ggplot, using some new data.

```
# Create breaks and labels to categorize the column UrbanPop
breaks <- c(0, 60, 75, Inf)
labels <- c("-59", "60-74", "75-")

# Apply the cut function with custom labels
USArrests$UrbanPopBins <- cut(USArrests$UrbanPop, breaks = breaks, labels = labels, right = FALSE)

# Create the boxplot
ggplot(USArrests, aes(x=UrbanPopBins, y=Murder, fill=UrbanPopBins)) +
  geom_boxplot(fill = c("#abdda4","#fee08b","#d53e4f")) +
  labs(title="Murders histogram for the USArrests dataset",
       fill=NULL,
       y="Murders",
       x="Urban Population (millions)",
       caption=NULL) +
  theme_light() +
  theme(plot.margin=margin(20, 0, 10, 0),
        aspect.ratio=1,
        panel.border=element_rect(colour="gray20"),
        panel.grid.minor.x=element_blank(),
        panel.grid.major.x=element_blank(),
        panel.grid.minor.y=element_line(color="gray85", linetype="dotted", linewidth=.2),
        panel.grid.major.y=element_line(color="gray40", linetype="dotted", linewidth=.5),
        panel.background=element_blank(),
```

```
        plot.title=element_text(size=10,
                                face="bold",
                                hjust=.5,
                                vjust=4,
                                colour="gray20"),
        axis.title.x=element_text(size=8.5,
                                  colour="gray20",
                                  face="bold"),
        axis.text.x=element_text(size=7.5,
                                 colour="gray20",
                                 angle=0),
        axis.title.y=element_text(size=8.5,
                                  colour="gray20",
                                  face="bold"),
        axis.text.y=element_text(size=7,
                                 colour="gray20"),
        legend.position="none"
  )
```



e. Create a scatter plot using ggplot, using some new data.

```
# Get extreme values to label those points in the scatterplot
USArrests$AssPopRatio <- USArrests$Assault / USArrests$UrbanPop

sampled_point_tail <- USArrests %>%
  arrange(AssPopRatio) %>%
```

```
  tail(4)
sampled_point_head <- USArrests %>%
  arrange(AssPopRatio) %>%
  head(4)
sampled_points <- rbind(sampled_point_tail, sampled_point_head)

# Create the scatterplot
ggplot(USArrests, aes(x=Rape, y=Assault, col=UrbanPop)) +
  geom_point() +
  scale_color_distiller(palette="Spectral") +
  geom_text(
    data=sampled_points,
    aes(label=row.names(sampled_points)),
    vjust=-.75,
    hjust=0.5,
    size=3,
    color="gray20",
    check_overlap = TRUE
    ) +
  labs(title="Assault vs Rape scatterplot by Urban Population",
       subtitle = "for the USArrests dataset",
       fill=NULL,
       y="Assaults",
       x="Rapes",
       caption=NULL) +
  theme_light() +
  theme(plot.margin=margin(20, 0, 10, 0),
        aspect.ratio=1,
        panel.border=element_rect(colour="gray20"),
        panel.grid.minor.x=element_line(color="gray85", linetype="dotted", linewidth=.2),
        panel.grid.major.x=element_line(color="gray40", linetype="dotted", linewidth=.5),
        panel.grid.minor.y=element_line(color="gray85", linetype="dotted", linewidth=.2),
        panel.grid.major.y=element_line(color="gray40", linetype="dotted", linewidth=.5),
        panel.background=element_blank(),
        plot.title=element_text(size=10,
                                face="bold",
                                hjust=.5,
                                vjust=4,
                                colour="gray20"),
        plot.subtitle=element_text(size=10,
                                face="bold",
                                hjust=.5,
                                vjust=4,
                                colour="gray20"),
        axis.title.x=element_text(size=8.5,
                                  colour="gray20",
                                  face="bold"),
        axis.text.x=element_text(size=7.5,
                                 colour="gray20",
                                 angle=0),
        axis.title.y=element_text(size=8.5,
                                  colour="gray20",
                                  face="bold"),
```

```
        axis.text.y=element_text(size=7,
                                 colour="gray20"),
        legend.position="right",
        legend.title=element_text(size = 8,
                                  colour = "gray20",
                                  face="bold"),
        legend.text=element_text(size = 7.5,
                                 colour = "gray20",
                                 margin=margin(c(0,0,0,0.5),unit="cm")),
        legend.key.size=unit(.6,"cm")
)
```