Shah Vaishakhi

2024-03-01

# Importing required libraries

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.5.0 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tibble)  
library(ggplot2)  
library (ggthemes)

# Exercise 5.1 : Movie Analysis

## Download the data from <http://becomingvisual.com/rfundamentals/movies.csv>

# 1. Getting to know the data

# a.Import the data

library(readr)  
movies <- read\_csv("movies.csv")

## Rows: 50 Columns: 8  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): Movie, Release\_Date, Distributor, Genre, MPAA  
## dbl (3): Rank, Gross\_Sales, Tickets\_Sold  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# b.View the data

#View(movies)

# c.Look at column names

names(movies)

## [1] "Rank" "Movie" "Release\_Date" "Distributor" "Genre"   
## [6] "MPAA" "Gross\_Sales" "Tickets\_Sold"

# d.Look at dimension of data (rows and columns)

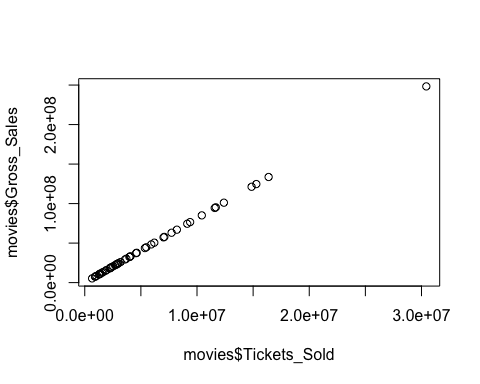
dim(movies)

## [1] 50 8

# 2. Scatter plots

# a. do scatter plot of Tickets Sold and Gross (Is the trend expected?)

plot(movies$Tickets\_Sold,movies$Gross\_Sales)



# Explanation: Is the trend expected?

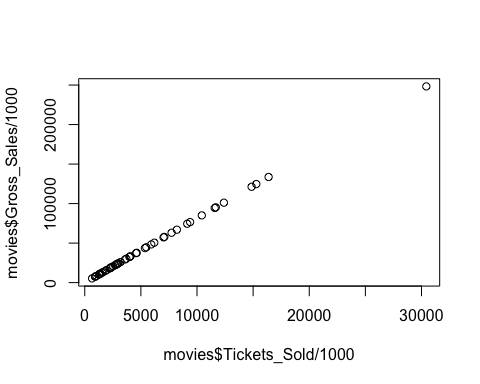
## Yes, the trend is expected : Based on common understanding, we might expect a positive correlation between tickets sold and gross sales.

## Typically, more tickets sold would result in higher gross sales, assuming each ticket sold contributes a fixed amount to the total gross.

## Below we are finding the correlation where it turns out, its exactly 1 which has detail explanation on the same to throw more light on the trend analysis.

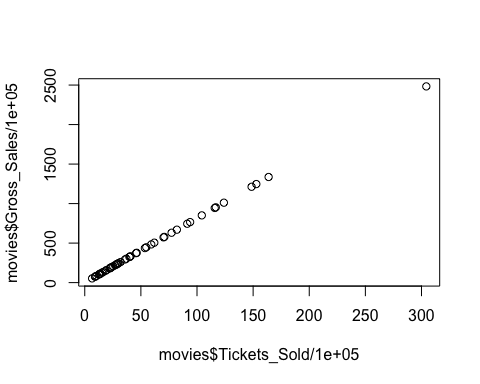
# b. redo scatter plot, adjusting scales, divide by 1000

plot(movies$Tickets\_Sold/1000 ,movies$Gross\_Sales/1000)



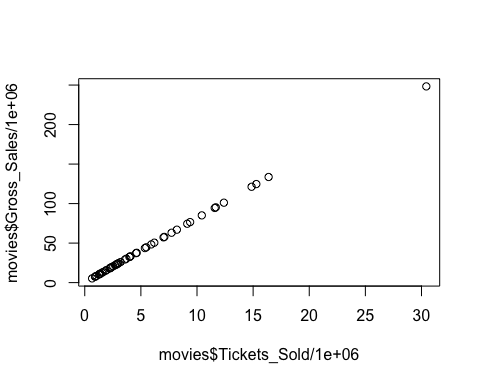
# c. redo scatter plot, adjusting scales, divide by 100,000

plot(movies$Tickets\_Sold/100000 ,movies$Gross\_Sales/100000)



# d. redo scatter plot, adjusting scales, divide by 1,000,000

plot(movies$Tickets\_Sold/1000000 ,movies$Gross\_Sales/1000000)



# 3. What is the correlation between tickets sold and sales? Is this expected?

cor(movies$Tickets\_Sold,movies$Gross\_Sales)

## [1] 1

# Explanation : Is this expected?

## A correlation coefficient of 1 represents a perfect positive linear relationship between two variables.

## In this case, a correlation coefficient of 1 between tickets sold and gross sales indicates that as the number of tickets sold increases, so do gross sales in a perfectly linear fashion.

## However, it is uncommon to find a correlation coefficient of exactly 1 in real-world data because other external factors frequently influencing the relationship between variables.

## It is possible as per this dataset as it is small and limited in scope, resulting in this perfect correlation.

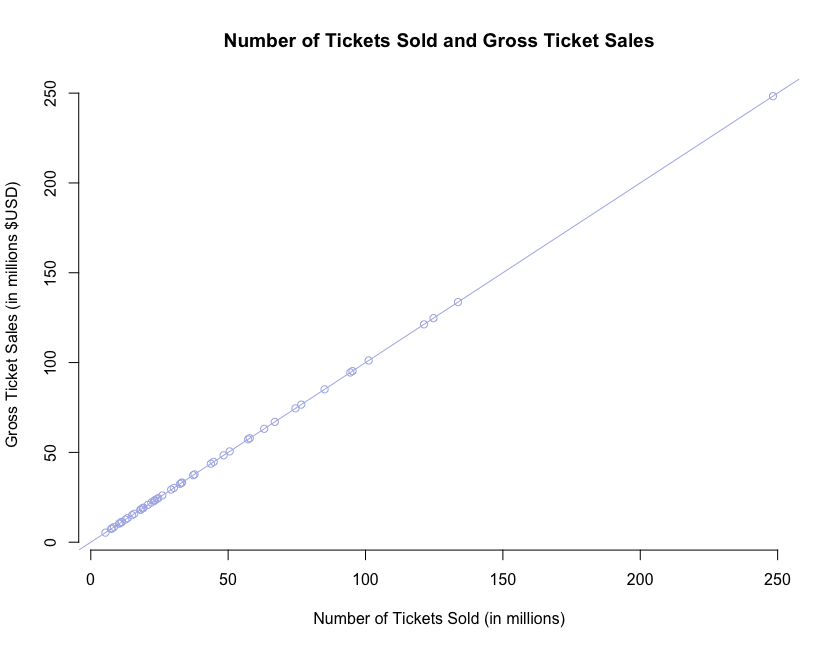
## In real-world scenarios, there can expect some variability and noise in the relationship between tickets sold and gross sales, so a correlation coefficient close to 1 may be more common.

# 4. Scatter plots with lines

# a. create scatter plot with millions scale, add a regression line

movies$Tickets\_Sold\_Million <- movies$Gross\_Sales/1000000  
movies$Gross\_Sales\_Million <- movies$Gross\_Sales/1000000

## Creating a scatter plot with millions scale  
plot(movies$Tickets\_Sold\_Million,movies$Gross\_Sales\_Million,main ="Number of Tickets Sold and Gross Ticket Sales", xlab = "Number of Tickets Sold (in millions)" , ylab = "Gross Ticket Sales (in millions $USD)" ,frame.plot = FALSE ,col ="#AEB6E9")  
## Adding a regression line  
  
abline(lm(movies$Gross\_Sales\_Million ~ movies$Tickets\_Sold\_Million), col ="#AEB6E9")

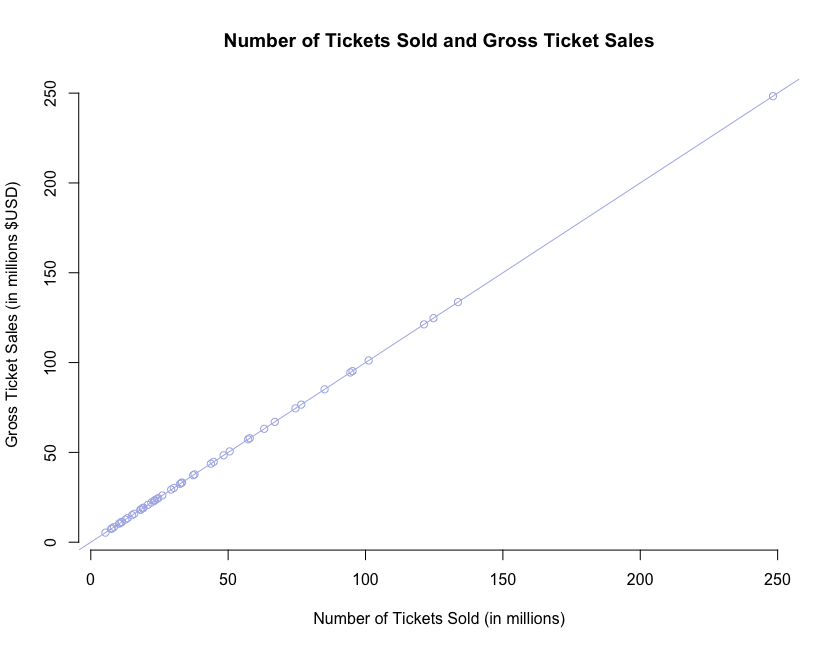


# calculating correlation  
  
cor(movies$Tickets\_Sold\_Million,movies$Gross\_Sales\_Million)

## [1] 1

# b. add label to x and y axis, add plot title label

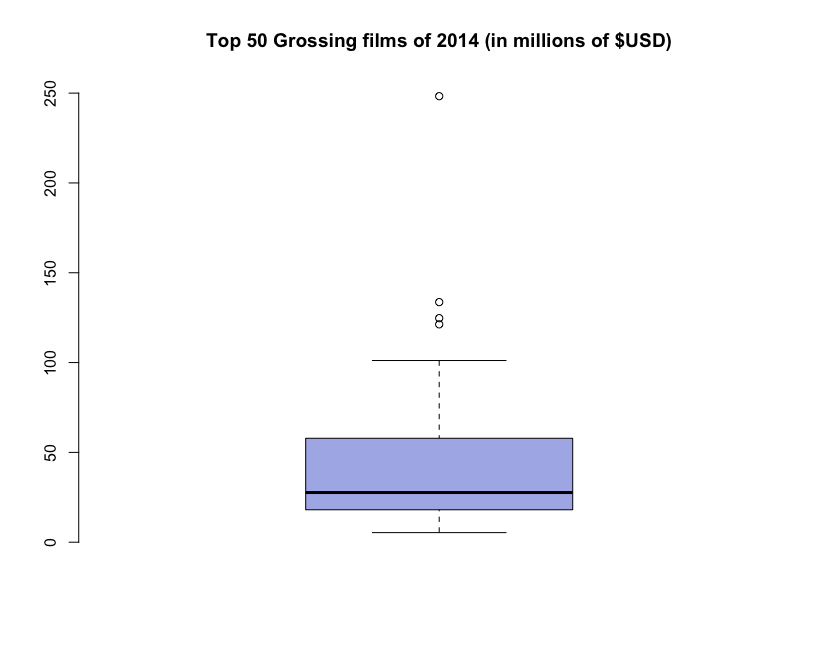
plot(movies$Tickets\_Sold\_Million,movies$Gross\_Sales\_Million,main ="Number of Tickets Sold and Gross Ticket Sales", xlab = "Number of Tickets Sold (in millions)" , ylab = "Gross Ticket Sales (in millions $USD)" ,frame.plot = FALSE ,col ="#AEB6E9")  
abline(lm(movies$Gross\_Sales\_Million ~ movies$Tickets\_Sold\_Million), col ="#AEB6E9")



# 5. Other plots

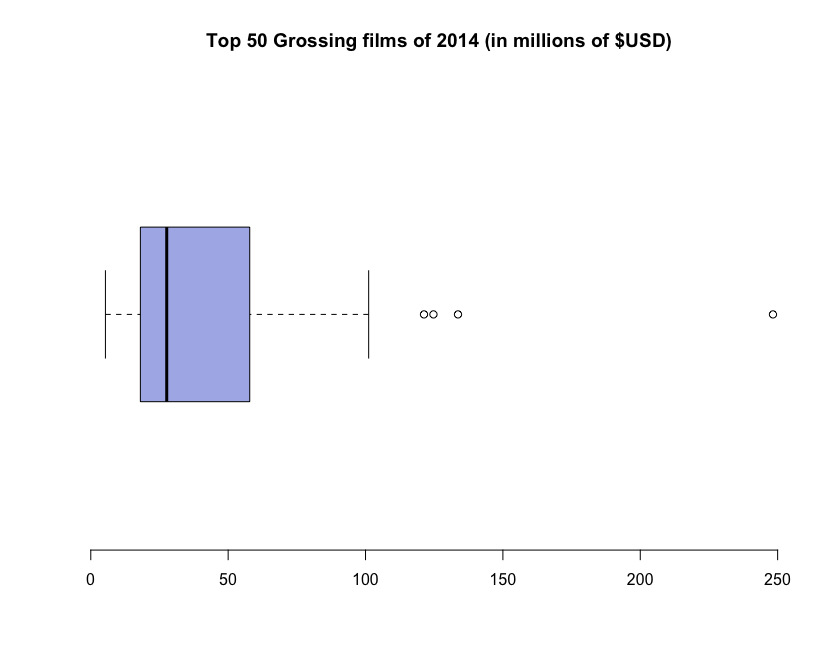
# a. do boxplot

boxplot(movies$Gross\_Sales\_Million,  
 main ="Top 50 Grossing films of 2014 (in millions of $USD)" ,   
 frame.plot = FALSE ,col ="#AEB6E9")



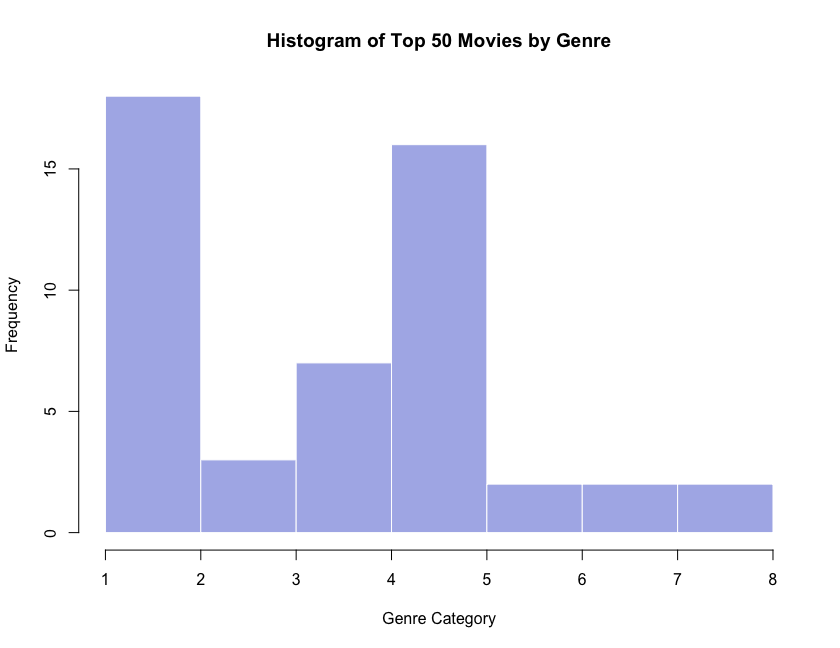
# b. do boxplot - horizontal

boxplot(movies$Gross\_Sales\_Million,  
 horizontal = TRUE,  
 main ="Top 50 Grossing films of 2014 (in millions of $USD)" ,   
 frame.plot = FALSE ,col ="#AEB6E9")



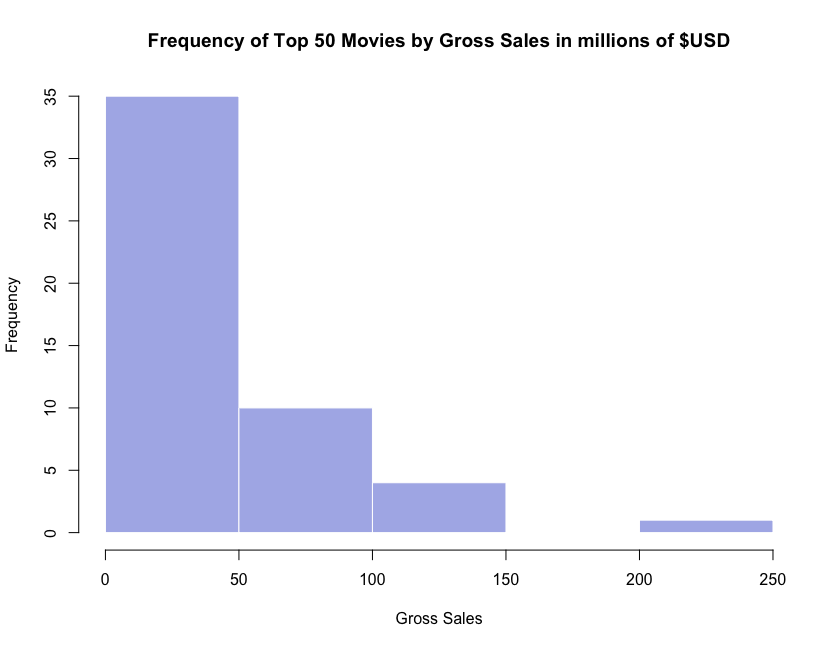
# c. do histogram for type of films

movies$Genre\_factor <- as.factor (movies$Genre)  
  
hist(as.numeric(movies$Genre\_factor),  
 main = "Histogram of Top 50 Movies by Genre",  
 xlab = "Genre Category",  
 ylab = "Frequency",  
 col ="#AEB6E9",  
 border = "#FFFFFF")



# d. do histogram of gross sales. How bins are shown by default?

hist(movies$Gross\_Sales\_Million,  
 main = "Frequency of Top 50 Movies by Gross Sales in millions of $USD",  
 xlab = "Gross Sales",  
 ylab = "Frequency",  
 col ="#AEB6E9",  
 border = "#FFFFFF")



# How bins are shown by default?

## The bins : By default, R selects the number breaks it sees fit. It ensures that the values on the x-axis are in logical intervals

## By default, the hist function in R automatically determines the number of bins to use based on the range and distribution of the data

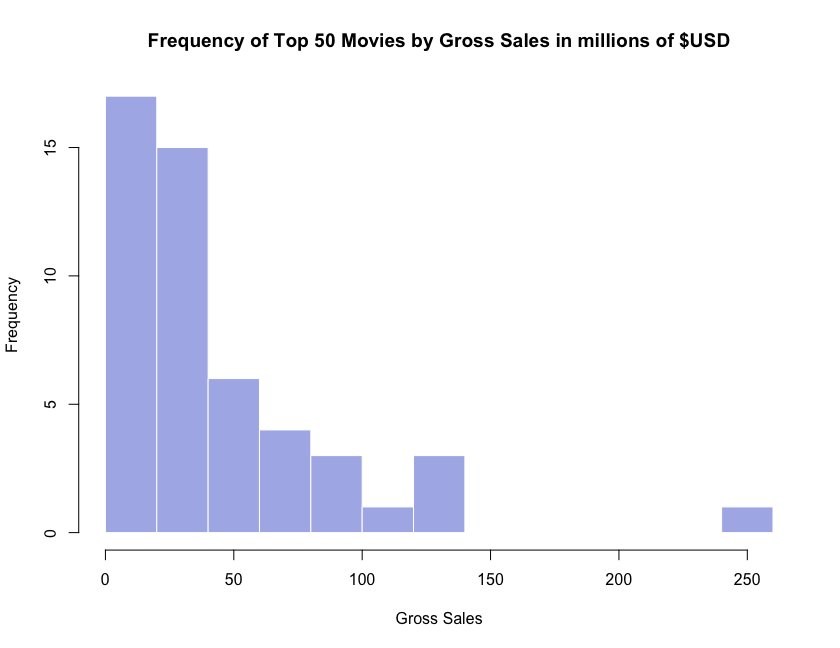
## When drawing histograms you need to determine where the breaks that separate the bins should be located and how many breaks there should be.

## However, to override this behavior, we can specify the number of bins manually using the two ways that can specify breaks: 1. How many breaks you want (e.g. breaks=3) , 2. Provide a vector that tells R exactly where the breaks should be placed

## In option 1, R treats it as a suggestion, rather than a command. This is odd for programming.

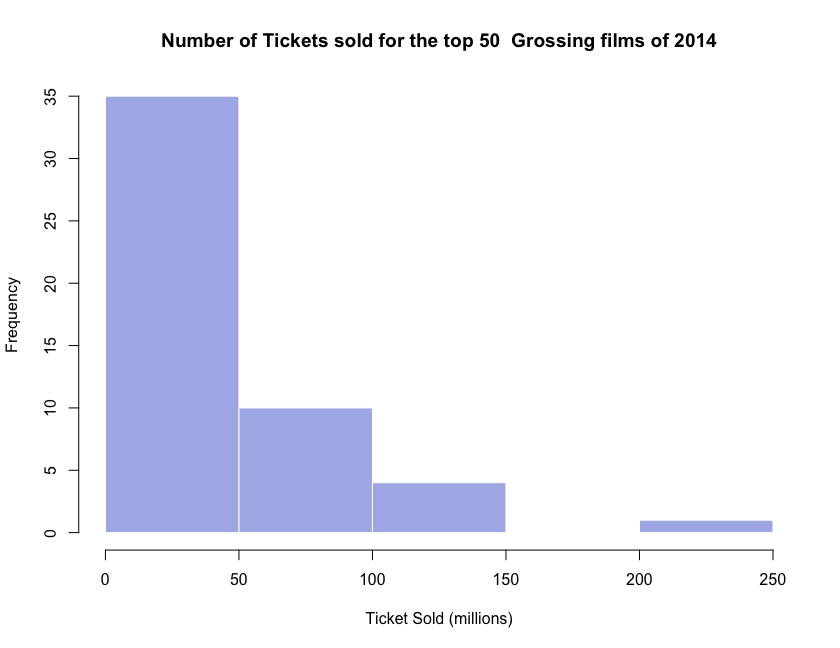
# e. do histogram of gross sales with 10 bins.

hist(movies$Gross\_Sales\_Million,  
 main = "Frequency of Top 50 Movies by Gross Sales in millions of $USD",  
 xlab = "Gross Sales",  
 ylab = "Frequency",  
 col ="#AEB6E9",  
 border = "#FFFFFF",  
 breaks = 10)

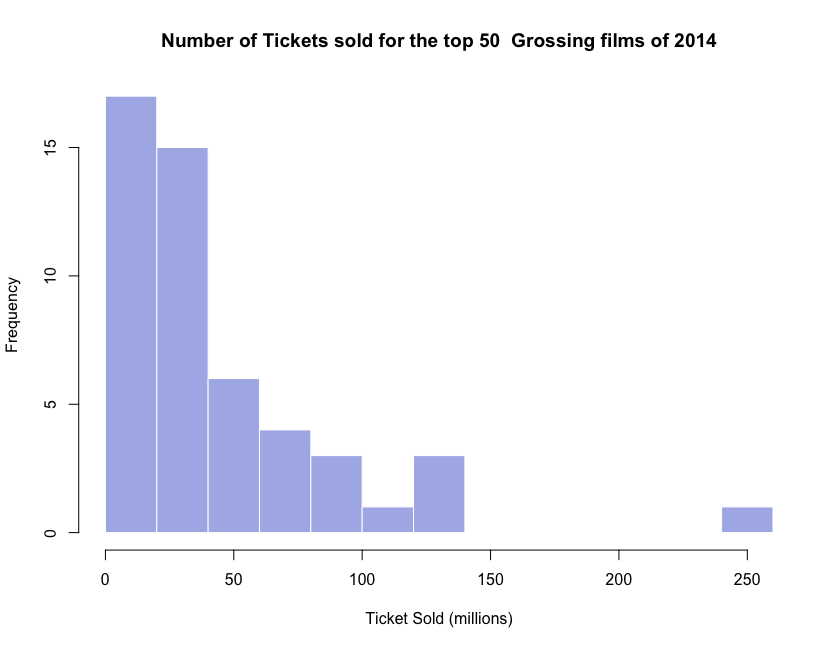


# f. do histogram of ticket sales. Try different bin numbers.

hist(movies$Tickets\_Sold\_Million,  
 main = "Number of Tickets sold for the top 50 Grossing films of 2014",  
 xlab = "Ticket Sold (millions)",  
 ylab = "Frequency",  
 col ="#AEB6E9",  
 border = "#FFFFFF")

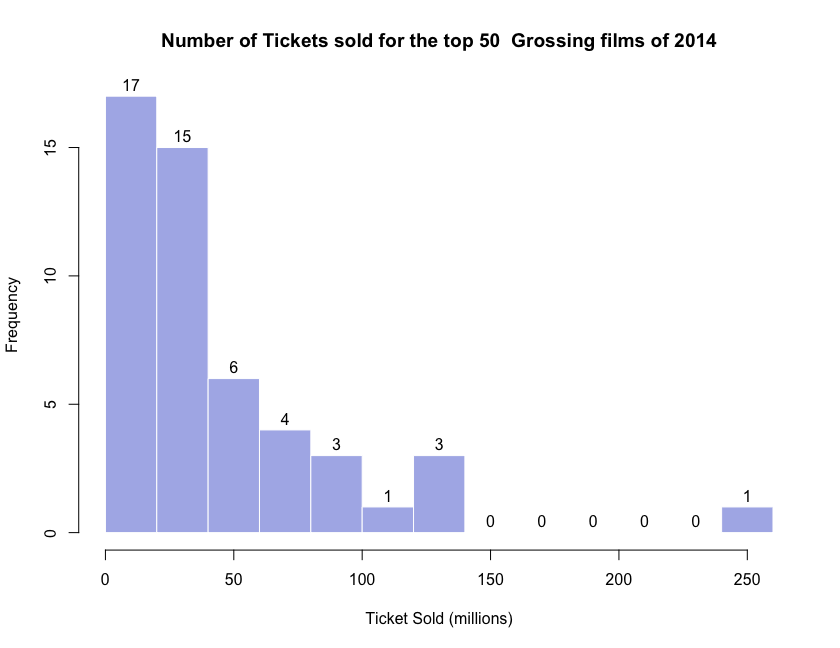


## Change the break - Try different bin numbers  
  
hist(movies$Tickets\_Sold\_Million,  
 breaks = 15,  
 main = "Number of Tickets sold for the top 50 Grossing films of 2014",  
 xlab = "Ticket Sold (millions)",  
 ylab = "Frequency",  
 col ="#AEB6E9",  
 border = "#FFFFFF")



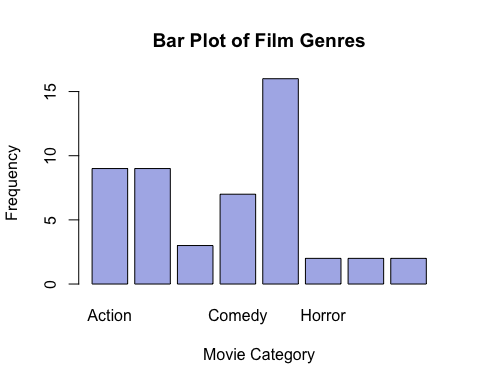
# g. do histogram of ticket sales (use millions unit). Add frequency count to top of bars. Add titles.

hist(movies$Tickets\_Sold\_Million,  
 breaks = 15,  
 labels = TRUE,  
 main = "Number of Tickets sold for the top 50 Grossing films of 2014",  
 xlab = "Ticket Sold (millions)",  
 ylab = "Frequency",  
 col ="#AEB6E9",  
 border = "#FFFFFF")



# h. do barplot of genre

barplot(table(movies$Genre),  
 main = "Bar Plot of Film Genres",  
 xlab = "Movie Category",  
 ylab = "Frequency",  
 col ="#AEB6E9",  
 border = "#000000")



# Exercise 5.2 : GDP and Life Expectancy Analysis

## In a R Markdown document, produce plots that describe the GDP and Life Expectancy during 2016 (see the Task below). You will need to create a new data frame with these columns.

## You can find the data here:

## GDP: <http://becomingvisual.com/rfundamentals/gdp.csv>

## Life Expectancy: <http://becomingvisual.com/rfundamentals/life_expectancy.csv>

## Initial steps

# Step1 - Importing GDP and Life Expectancy datasets  
  
library(readr)  
gdp <- read\_csv("gdp.csv")

## Rows: 264 Columns: 60  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): Country Name, Country Code  
## dbl (58): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

le <- read\_csv("life\_expectancy.csv")

## Rows: 264 Columns: 60  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): Country Name, Country Code  
## dbl (57): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...  
## lgl (1): 2017  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

#View(gdp)  
#View(le)

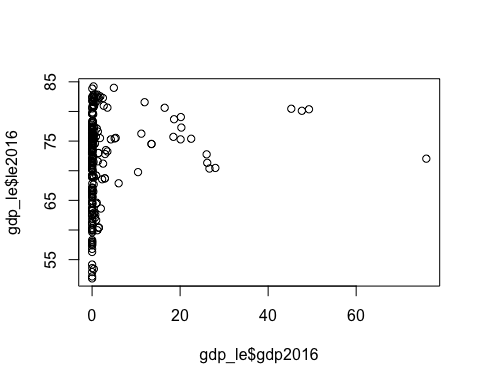
# Step2 - Format Data  
  
country <- gdp$"Country Name"  
gdp2016 <- gdp$"2016"/ 1000000000000  
le2016 <- le$"2016"

# Step3 - Create a new dataframe  
  
gdp\_le <- data.frame(country,gdp2016,le2016)  
  
# View(gdp\_le)

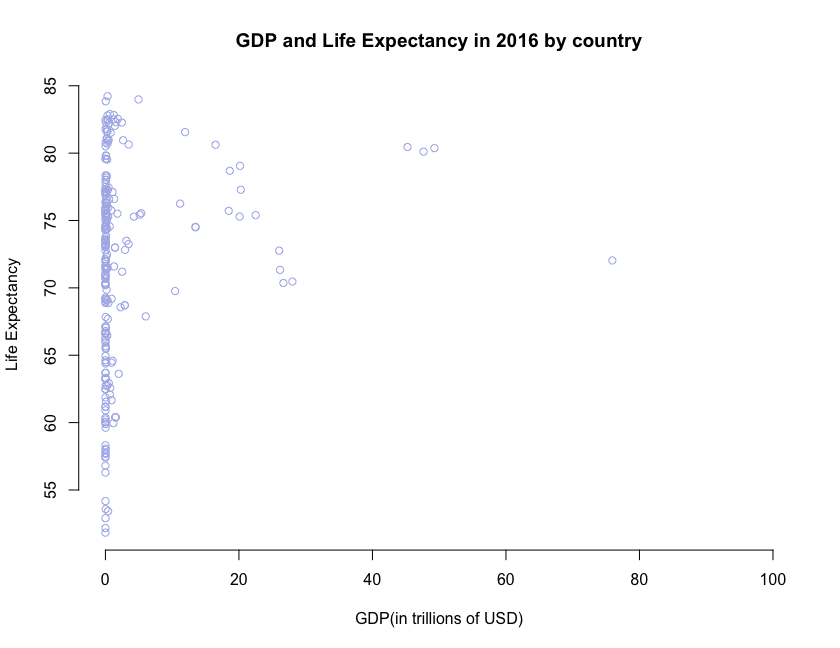
## Task:

# 1. Create a scatter plot of GDP to Life Expectancy

plot(gdp\_le$gdp2016,gdp\_le$le2016)

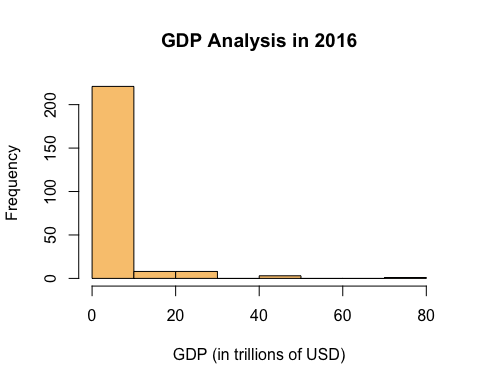


# Refining the chart   
  
plot(gdp\_le$gdp2016,gdp\_le$le2016,  
 main = "GDP and Life Expectancy in 2016 by country",  
 xlab = "GDP(in trillions of USD)",  
 ylab = "Life Expectancy",  
 frame.plot = FALSE,  
 col ="#AEB6E9",  
 xlim= c(min=0, max=100))



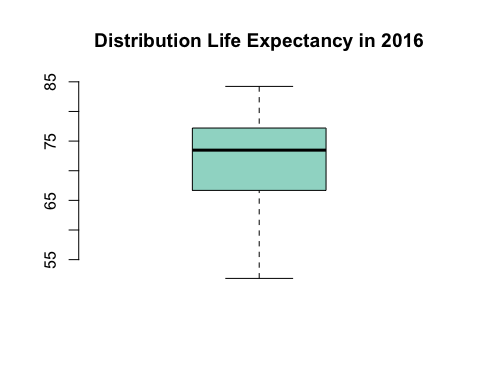
# 2. Create a histogram of GDP

hist(gdp\_le$gdp2016,  
 main = "GDP Analysis in 2016",  
 xlab = "GDP (in trillions of USD)",  
 ylab = "Frequency",  
 col ="#F9C77E")

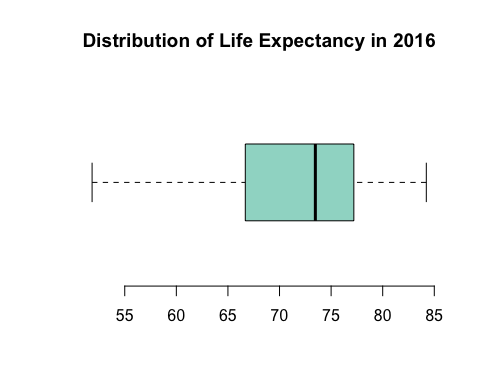


# 3. Create a box and whiskers plot of Life Expectancy

# Vertical   
  
boxplot(gdp\_le$le2016,  
 main = "Distribution Life Expectancy in 2016",  
 frame.plot = FALSE,  
 col = "#9ED9CC",  
 border = "#000000")



## Horizontal   
  
boxplot(gdp\_le$le2016,  
 horizontal = TRUE,  
 main = "Distribution of Life Expectancy in 2016",  
 frame.plot = FALSE,  
 col = "#9ED9CC",  
 border = "#000000")



## A box-and whisker plot provides a depiction of the median, the interquartile range, and the range of the data.

## The spacing between the different parts of the box help indicate the spread and skewness in the data.

## Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles

# Exercise 5.3 : Summer Winter Olympics Analysis

## Create an RMarkdown document to complete the following:

# 1. Getting to know the data

# a. Import the data (<http://becomingvisual.com/rfundamentals/summer_winter_olympics.csv>)

# Import the data   
  
library(readr)  
medals <- read\_csv("summer\_winter\_olympics.csv")

## New names:  
## Rows: 146 Columns: 17  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," chr  
## (1): Team..IOC.code. dbl (16): ...1, X..Summer, X, X.1, X.2, Total, X..Winter,  
## X.3, X.4, X.5, Tot...  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `` -> `...1`

#check class   
class(medals)

## [1] "spec\_tbl\_df" "tbl\_df" "tbl" "data.frame"

# convert to dataframe   
medal <- data.frame(medals)  
  
# how many variables re there in data frame ? (no. of columns)  
print(length(medals))

## [1] 17

# b. View the data

#View(medals)

# c. Look at column names

names(medals)

## [1] "...1" "Team..IOC.code." "X..Summer" "X"   
## [5] "X.1" "X.2" "Total" "X..Winter"   
## [9] "X.3" "X.4" "X.5" "Total.1"   
## [13] "X..Games" "X.6" "X.7" "X.8"   
## [17] "Combined.total"

# d. Look at dimension of data (rows and columns)

dim(medals)

## [1] 146 17

# 2. Dealing with Data

# a. Look at the column names and change names to more meaningful names.

names(medals)

## [1] "...1" "Team..IOC.code." "X..Summer" "X"   
## [5] "X.1" "X.2" "Total" "X..Winter"   
## [9] "X.3" "X.4" "X.5" "Total.1"   
## [13] "X..Games" "X.6" "X.7" "X.8"   
## [17] "Combined.total"

# b. The data represent, in order: (1) country (2) number of summer games played, gold, silver, bronze, total, (3) number of winter games played, gold, silver, bronze and total, total (4) total (Winter + Summer) games, gold, silver, bronze, total

# Renaming columns in the dataframe to more meaningful names.  
  
names(medals) = c("NA","country", "s\_games", "s\_gold", "s\_silver", "s\_bronze", "s\_total",  
 "w\_games", "w\_gold", "w\_silver", "w\_bronze", "w\_total",  
 "sw\_games", "sw\_gold", "sw\_silver", "sw\_bronze", "sw\_total")

# checking post changes(renaming)  
  
dim(medals)

## [1] 146 17

names(medals)

## [1] "NA" "country" "s\_games" "s\_gold" "s\_silver" "s\_bronze"   
## [7] "s\_total" "w\_games" "w\_gold" "w\_silver" "w\_bronze" "w\_total"   
## [13] "sw\_games" "sw\_gold" "sw\_silver" "sw\_bronze" "sw\_total"

# 3. Summary

attach(medals)

## The following object is masked \_by\_ .GlobalEnv:  
##   
## country

# a. use table() to find frequency of total summer games played

table(s\_games)

## s\_games  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26   
## 3 2 6 1 17 3 1 7 8 2 7 10 13 5 8 11 4 2 3 5 4 5 3 2 5 5   
## 27   
## 4

# b. explore the data with other variables

# calculate summary statistics for other variables  
  
summary(s\_gold)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 3.00 32.94 23.00 976.00

summary(s\_bronze)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 1.00 6.00 35.13 28.75 666.00

# get the basic descriptive statistics  
  
psych::describe(s\_gold)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 146 32.94 96.74 3 12.39 4.45 0 976 976 6.94 60.62 8.01

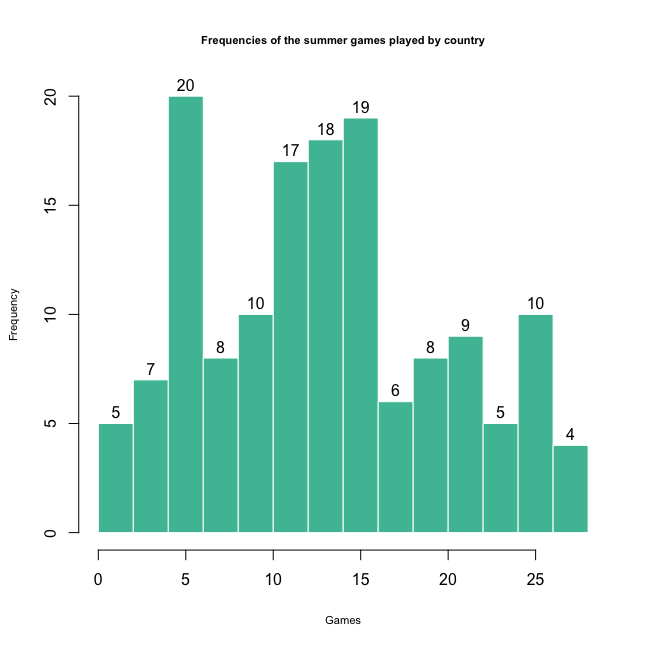
# calculate correlations between variables  
  
correlation\_matrix <- cor(medals[c("s\_games", "s\_gold", "s\_silver", "s\_bronze", "s\_total")])  
print(correlation\_matrix)

## s\_games s\_gold s\_silver s\_bronze s\_total  
## s\_games 1.0000000 0.2985213 0.3603901 0.3907429 0.3485493  
## s\_gold 0.2985213 1.0000000 0.9855432 0.9627301 0.9911633  
## s\_silver 0.3603901 0.9855432 1.0000000 0.9891444 0.9983057  
## s\_bronze 0.3907429 0.9627301 0.9891444 1.0000000 0.9894205  
## s\_total 0.3485493 0.9911633 0.9983057 0.9894205 1.0000000

# 4. Graphs

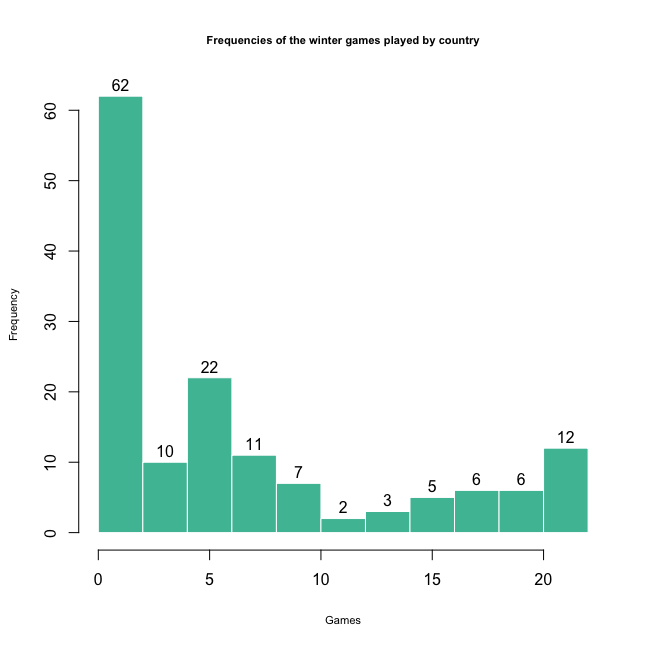
# a. do histogram of summer games (total)

hist(as.numeric(s\_games), breaks=10,   
 main = "Frequencies of the summer games played by country",   
 xlab = "Games", col="#4cbea3", labels=TRUE,   
 cex.lab = 0.7, cex.main = 0.7, border = "#FFFFFF")



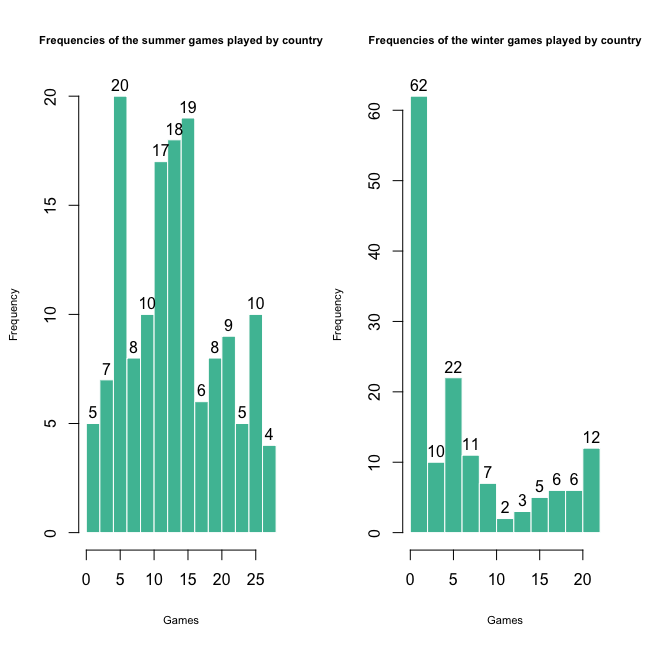
# b. do histogram of winter games (total)

hist(as.numeric(w\_games), breaks=10,  
 main = "Frequencies of the winter games played by country",   
 xlab = "Games", col="#4cbea3", labels=TRUE,  
 cex.lab = 0.7, cex.main = 0.7, border = "#FFFFFF")



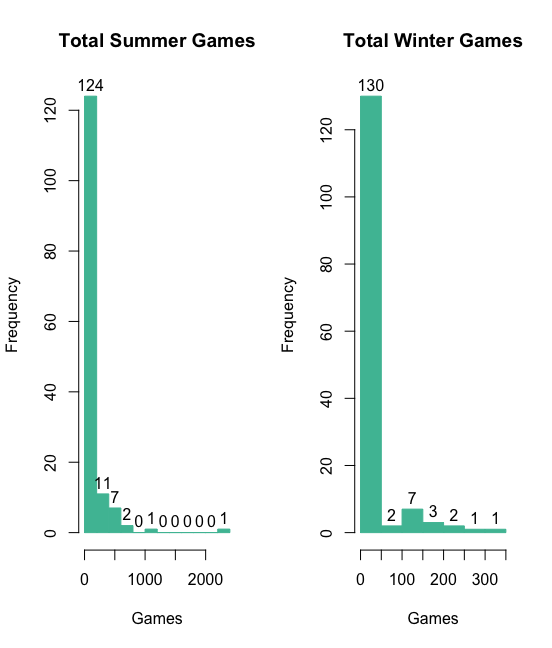
# c. put above two histograms on one page

par(mfrow = c(1,2))  
  
hist(as.numeric(s\_games), breaks=10,   
 main = "Frequencies of the summer games played by country",   
 xlab = "Games", col="#4cbea3", labels=TRUE,   
 cex.lab = 0.7, cex.main = 0.7, border = "#FFFFFF")  
hist(as.numeric(w\_games), breaks=10,  
 main = "Frequencies of the winter games played by country",   
 xlab = "Games", col="#4cbea3", labels=TRUE,  
 cex.lab = 0.7, cex.main = 0.7, border = "#FFFFFF")



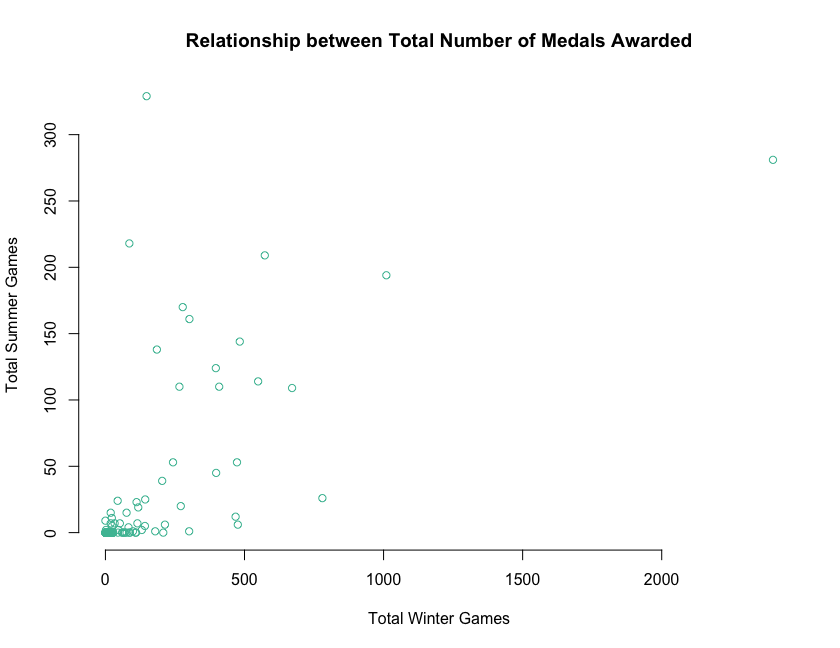
# d. do two histograms on one page: total summer, total winter medals won

par (mfrow = c( 1, 2))  
  
# Histogram for total summer games  
hist(s\_total, breaks = 10, main = "Total Summer Games",   
 xlab = "Games", col = "#4cbea3", labels = TRUE, border = "#4cbea3")  
  
# Histogram for total winter games  
hist(w\_total, breaks = 10, main = "Total Winter Games",   
 xlab = "Games", col = "#4cbea3", labels = TRUE, border = "#4cbea3")



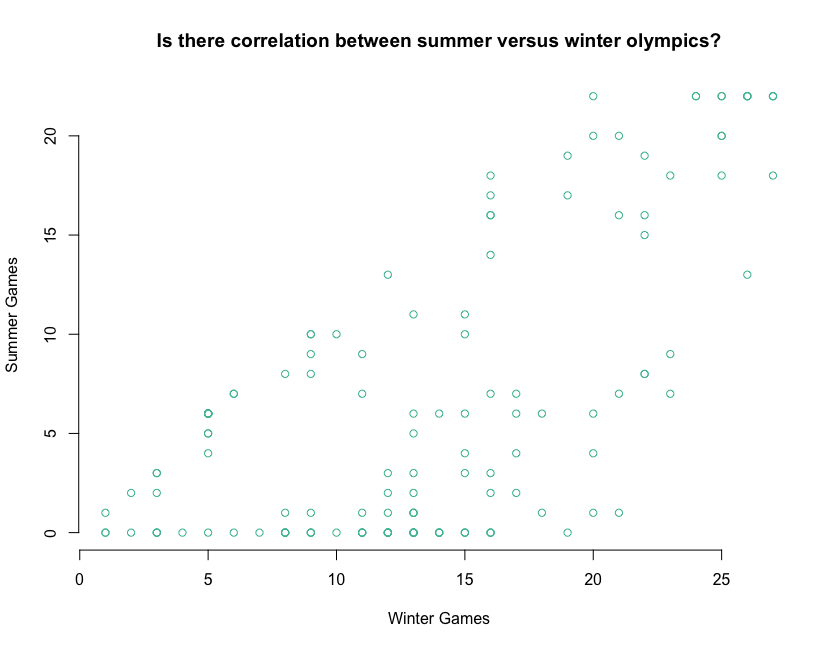
# e. is there a correlation between number of medals given out in winter and summer? (do plot)

#Reset to one graph per page  
  
par (mfrow=c(1,1))  
par (mfrow=c(1,1))  
  
# Plotting the relationship between total number of medals awarded in summer and winter games  
  
plot(s\_total, w\_total,   
 type = "p",   
 frame.plot = FALSE,   
 xlab = "Total Winter Games",   
 ylab = "Total Summer Games",   
 col = "#4cbea3",   
 main = "Relationship between Total Number of Medals Awarded")



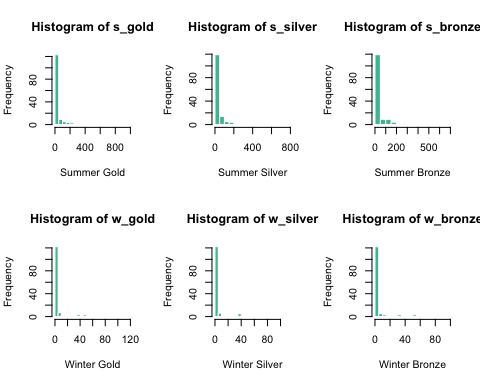
# f. how about number of games each country competes in. Is there correlation between winter and summer?

plot(s\_games, w\_games,   
 type = "p",   
 frame.plot = FALSE,   
 xlab = "Winter Games",   
 ylab = "Summer Games",   
 col = "#4cbea3",   
 main = "Is there correlation between summer versus winter olympics?")



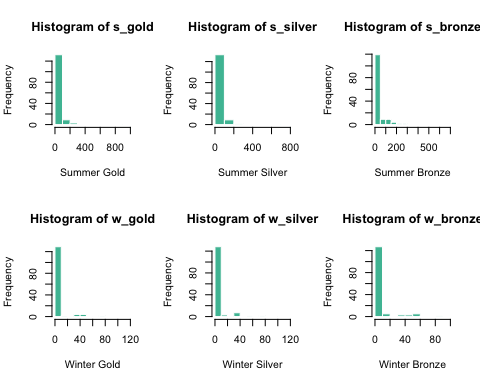
# g. look at distribution of each of the types of medals, by season (6 histograms on one page)

par(mfrow = c(2, 3))  
  
hist(s\_gold, breaks = 20, xlab = "Summer Gold", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of s\_gold")  
hist(s\_silver, breaks = 20, xlab = "Summer Silver", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of s\_silver")  
hist(s\_bronze, breaks = 20, xlab = "Summer Bronze", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of s\_bronze")  
hist(w\_gold, breaks = 20, xlab = "Winter Gold", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of w\_gold")  
hist(w\_silver, breaks = 20, xlab = "Winter Silver", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of w\_silver")  
hist(w\_bronze, breaks = 20, xlab = "Winter Bronze", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of w\_bronze")



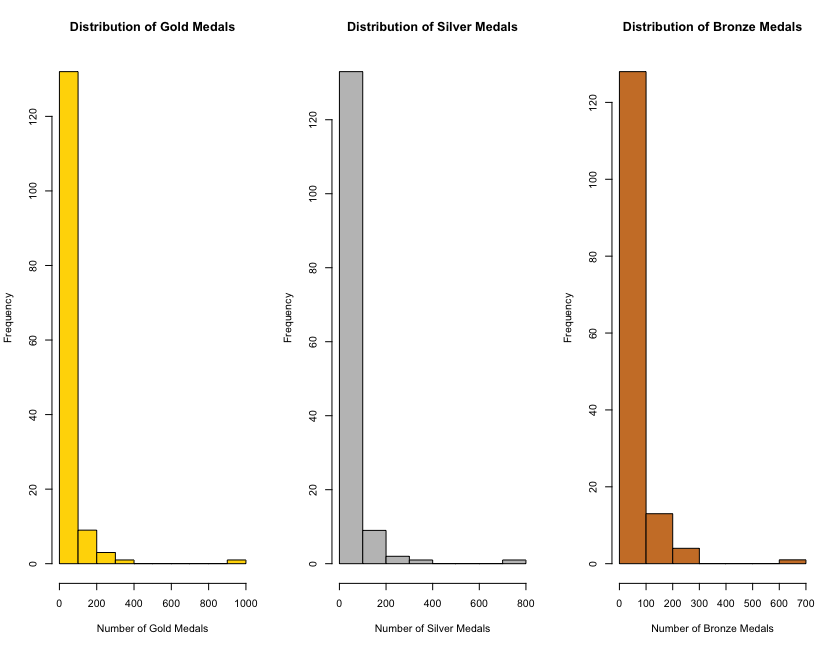
# h. redo g with different number of bins (10 instead of 20)

par(mfrow = c(2, 3))  
  
hist(s\_gold, breaks = 10, xlab = "Summer Gold", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of s\_gold")  
hist(s\_silver, breaks = 10, xlab = "Summer Silver", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of s\_silver")  
hist(s\_bronze, breaks = 10, xlab = "Summer Bronze", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of s\_bronze")  
hist(w\_gold, breaks = 10, xlab = "Winter Gold", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of w\_gold")  
hist(w\_silver, breaks = 10, xlab = "Winter Silver", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of w\_silver")  
hist(w\_bronze, breaks = 10, xlab = "Winter Bronze", col = "#4cbea3", border = "#FFFFFF", main = "Histogram of w\_bronze")

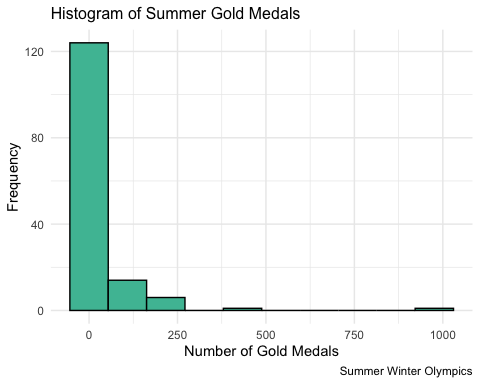


# i. explore data on your own

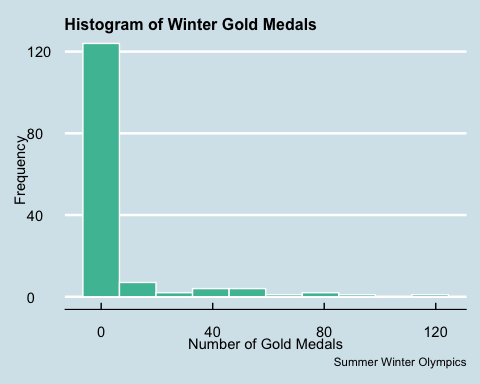
# 1. Histograms of Medals   
  
# Set up the layout for multiple plots  
par(mfrow = c(1, 3))  
  
# Histogram of Gold Medals  
hist(medals$s\_gold,   
 col = "#FFD700",   
 main = "Distribution of Gold Medals",   
 xlab = "Number of Gold Medals", ylab = "Frequency")  
  
# Histogram of Silver Medals  
hist(medals$s\_silver,   
 col = "#C0C0C0",  
 main = "Distribution of Silver Medals",  
 xlab = "Number of Silver Medals", ylab = "Frequency")  
  
# Histogram of Bronze Medals  
hist(medals$s\_bronze,   
 col = "#CD7F32",  
 main = "Distribution of Bronze Medals",   
 xlab = "Number of Bronze Medals", ylab = "Frequency")



# 2. Histogram of Summer and Winter Gold Medals- Using ggplot() function from the ggplot2 package.  
library(ggplot2)  
  
# Histogram of Summer Gold Medals  
ggplot(medals, aes(x = s\_gold)) +  
 geom\_histogram(fill = "#4cbea3", color = "#000000", bins = 10) +  
 labs(title = "Histogram of Summer Gold Medals", x = "Number of Gold Medals", y = "Frequency", caption= "Summer Winter Olympics")+  
 theme\_minimal() +  
 theme(plot.title = element\_text(size = 12))



# Histogram of Winter Gold Medals  
  
ggplot(medals, aes(x = w\_gold)) +  
 geom\_histogram(fill = "#4cbea3", color = "#FFFFFF", bins = 10) +  
 labs(title = "Histogram of Winter Gold Medals", x = "Number of Gold Medals", y = "Frequency",caption= "Summer Winter Olympics") +  
 theme\_economist() +  
 theme(plot.title = element\_text(size = 12))



# Exercise 5.4 : GDP\_Life Expectancy\_Employment Analysis

## Merge the columns for the year 2016 for GDP, Life Expectancy, and Employment into a new data frame and clean-up the new table.

# You can find the data here:

## GDP: <http://becomingvisual.com/rfundamentals/gdp.csv>

## Life Expectancy: <http://becomingvisual.com/rfundamentals/life_expectancy.csv>

## Employment: <http://becomingvisual.com/rfundamentals/employment.csv>

## Task:

## Step1 - Importing GDP, Life Expectancy and Employment datasets  
  
library(readr)  
gdp <- read\_csv("gdp.csv")

## Rows: 264 Columns: 60  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): Country Name, Country Code  
## dbl (58): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

le <- read\_csv("life\_expectancy.csv")

## Rows: 264 Columns: 60  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): Country Name, Country Code  
## dbl (57): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...  
## lgl (1): 2017  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

employment <- read\_csv("employment.csv")

## Rows: 264 Columns: 60  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): Country Name, Country Code  
## dbl (27): 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, ...  
## lgl (31): 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# View(gdp)  
# View(le)  
# View(employment)

# Step2 - Create a new dataframe  
  
countries\_2016 <- data.frame(gdp$"Country Name",gdp$"2016",le$"2016",employment$"2016")  
  
countries\_2016

## gdp..Country.Name. gdp..2016. le..2016.  
## 1 Aruba NA 75.86700  
## 2 Afghanistan 1.946902e+10 63.67300  
## 3 Angola 9.533720e+10 61.54700  
## 4 Albania 1.188368e+10 78.34500  
## 5 Andorra 2.877312e+09 NA  
## 6 Arab World 2.500164e+12 71.19846  
## 7 United Arab Emirates 3.570451e+11 77.25600  
## 8 Argentina 5.548609e+11 76.57700  
## 9 Armenia 1.054614e+10 74.61800  
## 10 American Samoa 6.580000e+08 NA  
## 11 Antigua and Barbuda 1.460145e+09 76.36400  
## 12 Australia 1.208039e+12 82.50000  
## 13 Austria 3.908000e+11 80.89024  
## 14 Azerbaijan 3.786752e+10 72.02600  
## 15 Burundi 3.007029e+09 57.48100  
## 16 Belgium 4.675455e+11 80.99268  
## 17 Benin 8.573160e+09 60.90700  
## 18 Burkina Faso 1.144878e+10 60.36100  
## 19 Bangladesh 2.214152e+11 72.48900  
## 20 Bulgaria 5.324089e+10 74.61463  
## 21 Bahrain 3.215269e+10 76.90000  
## 22 Bahamas, The 1.183880e+10 75.67500  
## 23 Bosnia and Herzegovina 1.691109e+10 76.91100  
## 24 Belarus 4.772266e+10 73.82683  
## 25 Belize 1.820159e+09 70.38400  
## 26 Bermuda NA 81.22707  
## 27 Bolivia 3.394113e+10 69.12500  
## 28 Brazil 1.793989e+12 75.50900  
## 29 Barbados 4.529050e+09 75.90600  
## 30 Brunei Darussalam 1.140065e+10 77.20300  
## 31 Bhutan 2.212639e+09 70.19700  
## 32 Botswana 1.564870e+10 66.79700  
## 33 Central African Republic 1.755468e+09 52.17100  
## 34 Canada 1.535768e+12 82.30051  
## 35 Central Europe and the Baltics 1.313067e+12 76.60313  
## 36 Switzerland 6.687453e+11 82.89756  
## 37 Channel Islands NA 81.10700  
## 38 Chile 2.500362e+11 79.52200  
## 39 China 1.119099e+13 76.25200  
## 40 Cote d'Ivoire 3.637485e+10 53.58200  
## 41 Cameroon 3.221754e+10 58.07300  
## 42 Congo, Dem. Rep. 3.499116e+10 59.62100  
## 43 Congo, Rep. 7.833509e+09 64.62500  
## 44 Colombia 2.800910e+11 74.38100  
## 45 Comoros 6.166545e+08 63.70100  
## 46 Cabo Verde 1.638927e+09 72.79800  
## 47 Costa Rica 5.698899e+10 79.83100  
## 48 Caribbean small states 6.778768e+10 73.19241  
## 49 Cuba NA 79.74200  
## 50 Curacao NA 77.87317  
## 51 Cayman Islands NA NA  
## 52 Cyprus 2.015387e+10 80.50800  
## 53 Czech Republic 1.953051e+11 78.33171  
## 54 Germany 3.477796e+12 80.64146  
## 55 Djibouti 1.764268e+09 62.46500  
## 56 Dominica 5.814840e+08 NA  
## 57 Denmark 3.068997e+11 80.70488  
## 58 Dominican Republic 7.234297e+10 73.86100  
## 59 Algeria 1.590491e+11 76.07800  
## 60 East Asia & Pacific (excluding high income) 1.350933e+13 74.49253  
## 61 Early-demographic dividend 1.043743e+13 69.76627  
## 62 East Asia & Pacific 2.251217e+13 75.39573  
## 63 Europe & Central Asia (excluding high income) 2.943297e+12 72.81472  
## 64 Europe & Central Asia 2.028071e+13 77.28145  
## 65 Ecuador 9.861397e+10 76.32700  
## 66 Egypt, Arab Rep. 3.329278e+11 71.48400  
## 67 Euro area 1.193474e+13 81.55993  
## 68 Eritrea NA 65.09200  
## 69 Spain 1.237255e+12 82.83171  
## 70 Estonia 2.333791e+10 77.73659  
## 71 Ethiopia 7.300098e+10 65.47500  
## 72 European Union 1.649186e+13 80.62040  
## 73 Fragile and conflict affected situations 7.240911e+11 62.56940  
## 74 Finland 2.386777e+11 81.78049  
## 75 Fiji 4.671313e+09 70.26900  
## 76 France 2.465134e+12 82.27317  
## 77 Faroe Islands NA 82.24146  
## 78 Micronesia, Fed. Sts. 3.298956e+08 69.19500  
## 79 Gabon 1.401428e+10 66.10500  
## 80 United Kingdom 2.650850e+12 80.95610  
## 81 Georgia 1.437802e+10 73.26100  
## 82 Ghana 4.280358e+10 62.74200  
## 83 Gibraltar NA NA  
## 84 Guinea 9.275886e+09 60.01500  
## 85 Gambia, The 9.627971e+08 61.19300  
## 86 Guinea-Bissau 1.178205e+09 57.40300  
## 87 Equatorial Guinea 1.125956e+10 57.68100  
## 88 Greece 1.926908e+11 81.03659  
## 89 Grenada 1.056189e+09 73.65900  
## 90 Greenland 2.705892e+09 NA  
## 91 Guatemala 6.866365e+10 73.40900  
## 92 Guam 5.793000e+09 79.57500  
## 93 Guyana 3.504024e+09 66.65000  
## 94 High income 4.928186e+13 80.37015  
## 95 Hong Kong SAR, China 3.208812e+11 84.22683  
## 96 Honduras 2.164394e+10 73.57500  
## 97 Heavily indebted poor countries (HIPC) 6.521759e+11 62.08175  
## 98 Croatia 5.133852e+10 78.02195  
## 99 Haiti 7.970649e+09 63.33000  
## 100 Hungary 1.258166e+11 75.56829  
## 101 IBRD only 2.602546e+13 72.76271  
## 102 IDA & IBRD total 2.800566e+13 70.46865  
## 103 IDA total 1.988820e+12 63.62072  
## 104 IDA blend 9.240355e+11 61.66245  
## 105 Indonesia 9.322565e+11 69.19100  
## 106 IDA only 1.060272e+12 64.58967  
## 107 Isle of Man NA NA  
## 108 India 2.274230e+12 68.56000  
## 109 Not classified NA NA  
## 110 Ireland 3.048190e+11 81.60732  
## 111 Iran, Islamic Rep. 4.189767e+11 75.95300  
## 112 Iraq 1.714890e+11 69.86200  
## 113 Iceland 2.030410e+10 82.46829  
## 114 Israel 3.177475e+11 82.40732  
## 115 Italy 1.859384e+12 82.54390  
## 116 Jamaica 1.405691e+10 75.97000  
## 117 Jordan 3.865473e+10 74.32900  
## 118 Japan 4.949273e+12 83.98488  
## 119 Kazakhstan 1.372783e+11 72.30000  
## 120 Kenya 7.087529e+10 67.03200  
## 121 Kyrgyz Republic 6.813092e+09 70.95122  
## 122 Cambodia 2.001675e+10 68.98100  
## 123 Kiribati 1.815515e+08 66.31700  
## 124 St. Kitts and Nevis 9.098546e+08 NA  
## 125 Korea, Rep. 1.414804e+12 82.02439  
## 126 Kuwait 1.109123e+11 74.69400  
## 127 Latin America & Caribbean (excluding high income) 4.263792e+12 75.28667  
## 128 Lao PDR 1.580571e+10 66.68300  
## 129 Lebanon 4.959883e+10 79.58400  
## 130 Liberia 2.101000e+09 62.50500  
## 131 Libya 3.225717e+10 71.93400  
## 132 St. Lucia 1.667079e+09 75.49600  
## 133 Latin America & Caribbean 5.360659e+12 75.54005  
## 134 Least developed countries: UN classification 9.296720e+11 64.44520  
## 135 Low income 5.017795e+11 62.93547  
## 136 Liechtenstein NA 82.65610  
## 137 Sri Lanka 8.178838e+10 75.28400  
## 138 Lower middle income 6.036799e+12 67.88221  
## 139 Low & middle income 2.666356e+13 70.36682  
## 140 Lesotho 2.291320e+09 54.17400  
## 141 Late-demographic dividend 1.846759e+13 75.71566  
## 142 Lithuania 4.277303e+10 74.32195  
## 143 Luxembourg 5.863132e+10 82.29268  
## 144 Latvia 2.757151e+10 74.52927  
## 145 Macao SAR, China 4.531088e+10 83.84900  
## 146 St. Martin (French part) NA 79.62195  
## 147 Morocco 1.036063e+11 75.82100  
## 148 Monaco NA NA  
## 149 Moldova 6.795742e+09 71.61000  
## 150 Madagascar 1.000119e+10 65.93200  
## 151 Maldives 4.222767e+09 77.33900  
## 152 Middle East & North Africa 3.142484e+12 73.49180  
## 153 Mexico 1.076912e+12 77.11800  
## 154 Marshall Islands 1.944979e+08 NA  
## 155 Middle income 2.615783e+13 71.33373  
## 156 Macedonia, FYR 1.074579e+10 75.70300  
## 157 Mali 1.403498e+10 57.96600  
## 158 Malta 1.127954e+10 81.79756  
## 159 Myanmar 6.322510e+10 66.61200  
## 160 Middle East & North Africa (excluding high income) 1.444679e+12 72.99192  
## 161 Montenegro 4.374131e+09 77.11600  
## 162 Mongolia 1.118346e+10 69.28700  
## 163 Northern Mariana Islands 1.242000e+09 NA  
## 164 Mozambique 1.101486e+10 58.31100  
## 165 Mauritania 4.739298e+09 63.23800  
## 166 Mauritius 1.223246e+10 74.39488  
## 167 Malawi 5.433039e+09 63.22300  
## 168 Malaysia 2.965359e+11 75.30000  
## 169 North America 2.016631e+13 79.05465  
## 170 Namibia 1.130923e+10 64.38800  
## 171 New Caledonia NA 76.96150  
## 172 Niger 7.606749e+09 60.05800  
## 173 Nigeria 4.046527e+11 53.42800  
## 174 Nicaragua 1.318499e+10 75.40400  
## 175 Netherlands 7.772275e+11 81.50976  
## 176 Norway 3.710752e+11 82.50976  
## 177 Nepal 2.113198e+10 70.25300  
## 178 Nauru 1.020601e+08 NA  
## 179 New Zealand 1.892860e+11 81.61244  
## 180 OECD members 4.765822e+13 80.11102  
## 181 Oman 6.682445e+10 77.02900  
## 182 Other small states 3.644192e+11 67.69503  
## 183 Pakistan 2.786546e+11 66.48100  
## 184 Panama 5.782092e+10 78.00100  
## 185 Peru 1.916397e+11 74.98300  
## 186 Philippines 3.048891e+11 69.09400  
## 187 Palau 3.026998e+08 NA  
## 188 Papua New Guinea 1.990481e+10 65.54400  
## 189 Poland 4.714003e+11 77.45122  
## 190 Pre-demographic dividend 1.207094e+12 59.96144  
## 191 Puerto Rico 1.050345e+11 79.78832  
## 192 Korea, Dem. People\xd5s Rep. NA 71.68500  
## 193 Portugal 2.051845e+11 81.12683  
## 194 Paraguay 2.742407e+10 73.12000  
## 195 West Bank and Gaza 1.342570e+10 73.47300  
## 196 Pacific island small states 9.027151e+09 70.89538  
## 197 Post-demographic dividend 4.525737e+13 80.45602  
## 198 French Polynesia NA 76.81300  
## 199 Qatar 1.524519e+11 78.18400  
## 200 Romania 1.878059e+11 75.01220  
## 201 Russian Federation 1.284728e+12 71.59293  
## 202 Rwanda 8.475682e+09 67.12900  
## 203 South Asia 2.903124e+12 68.71053  
## 204 Saudi Arabia 6.449355e+11 74.56100  
## 205 Sudan 9.558438e+10 64.48600  
## 206 Senegal 1.468375e+10 67.14600  
## 207 Singapore 3.097639e+11 82.79512  
## 208 Solomon Islands 1.232699e+09 70.72600  
## 209 Sierra Leone 3.556037e+09 51.83500  
## 210 El Salvador 2.391223e+10 73.51200  
## 211 San Marino 1.590708e+09 NA  
## 212 Somalia 6.752653e+09 56.29300  
## 213 Serbia 3.829985e+10 75.23902  
## 214 Sub-Saharan Africa (excluding high income) 1.511047e+12 60.39356  
## 215 South Sudan 2.904115e+09 56.81100  
## 216 Sub-Saharan Africa 1.512473e+12 60.39484  
## 217 Small states 4.412340e+11 68.87449  
## 218 Sao Tome and Principe 3.542381e+08 66.62400  
## 219 Suriname 3.278425e+09 71.40500  
## 220 Slovak Republic 8.976860e+10 76.56341  
## 221 Slovenia 4.470860e+10 80.77561  
## 222 Sweden 5.144600e+11 82.20488  
## 223 Eswatini 3.720649e+09 57.75400  
## 224 Sint Maarten (Dutch part) NA NA  
## 225 Seychelles 1.425929e+09 74.30976  
## 226 Syrian Arab Republic NA 70.31000  
## 227 Turks and Caicos Islands NA NA  
## 228 Chad 9.412034e+09 52.90300  
## 229 East Asia & Pacific (IDA & IBRD countries) 1.348339e+13 74.52766  
## 230 Europe & Central Asia (IDA & IBRD countries) 3.466036e+12 73.24905  
## 231 Togo 4.388570e+09 60.23200  
## 232 Thailand 4.117552e+11 75.30300  
## 233 Tajikistan 6.951657e+09 71.05100  
## 234 Turkmenistan 3.617989e+10 67.83500  
## 235 Latin America & the Caribbean (IDA & IBRD countries) 5.134442e+12 75.43740  
## 236 Timor-Leste 2.521008e+09 68.88100  
## 237 Middle East & North Africa (IDA & IBRD countries) 1.431082e+12 72.98599  
## 238 Tonga 4.015620e+08 73.02900  
## 239 South Asia (IDA & IBRD) 2.903124e+12 68.71053  
## 240 Sub-Saharan Africa (IDA & IBRD countries) 1.512473e+12 60.39484  
## 241 Trinidad and Tobago 2.232001e+10 70.67300  
## 242 Tunisia 4.206255e+10 75.73100  
## 243 Turkey 8.637216e+11 75.75500  
## 244 Tuvalu 3.657261e+07 NA  
## 245 Tanzania 4.738840e+10 65.67500  
## 246 Uganda 2.407893e+10 59.88900  
## 247 Ukraine 9.327048e+10 71.47634  
## 248 Upper middle income 2.010892e+13 75.29062  
## 249 Uruguay 5.268761e+10 77.49300  
## 250 United States 1.862448e+13 78.69024  
## 251 Uzbekistan 6.706757e+10 71.31400  
## 252 St. Vincent and the Grenadines 7.655556e+08 73.17900  
## 253 Venezuela, RB NA 74.54500  
## 254 British Virgin Islands NA NA  
## 255 Virgin Islands (U.S.) NA 79.26829  
## 256 Vietnam 2.052762e+11 76.25300  
## 257 Vanuatu 7.879426e+08 72.13300  
## 258 World 7.593681e+13 72.03534  
## 259 Samoa 7.863563e+08 75.01300  
## 260 Kosovo 6.715487e+09 71.64634  
## 261 Yemen, Rep. 1.821333e+10 64.95300  
## 262 South Africa 2.957627e+11 62.77400  
## 263 Zambia 2.095475e+10 61.87400  
## 264 Zimbabwe 1.661996e+10 61.16300  
## employment..2016.  
## 1 NA  
## 2 49.43900  
## 3 71.67600  
## 4 47.71300  
## 5 NA  
## 6 43.73156  
## 7 79.06200  
## 8 54.74900  
## 9 49.09500  
## 10 NA  
## 11 NA  
## 12 61.18100  
## 13 56.79600  
## 14 62.93500  
## 15 77.55700  
## 16 49.06400  
## 17 68.98900  
## 18 62.39100  
## 19 53.98000  
## 20 49.40700  
## 21 70.99200  
## 22 66.08200  
## 23 35.01700  
## 24 63.93500  
## 25 61.51200  
## 26 NA  
## 27 65.29700  
## 28 56.42400  
## 29 59.86000  
## 30 62.52900  
## 31 64.90500  
## 32 59.34800  
## 33 67.07800  
## 34 60.81300  
## 35 52.76415  
## 36 65.19100  
## 37 52.24200  
## 38 57.95300  
## 39 66.14100  
## 40 55.87700  
## 41 72.94200  
## 42 69.74800  
## 43 62.47400  
## 44 64.39900  
## 45 41.09300  
## 46 53.89100  
## 47 54.08100  
## 48 57.55373  
## 49 53.06900  
## 50 NA  
## 51 NA  
## 52 54.72600  
## 53 57.76700  
## 54 58.00000  
## 55 55.48100  
## 56 NA  
## 57 59.39400  
## 58 63.11800  
## 59 37.18400  
## 60 66.30836  
## 61 54.80050  
## 62 65.50412  
## 63 55.08223  
## 64 53.90948  
## 65 65.36300  
## 66 41.90900  
## 67 50.99243  
## 68 76.11100  
## 69 46.69600  
## 70 58.69200  
## 71 78.14400  
## 72 52.55013  
## 73 56.43361  
## 74 53.31500  
## 75 54.72200  
## 76 49.84500  
## 77 NA  
## 78 NA  
## 79 41.01400  
## 80 59.35200  
## 81 59.56000  
## 82 75.00000  
## 83 NA  
## 84 61.34400  
## 85 53.60700  
## 86 67.15800  
## 87 55.81200  
## 88 40.64500  
## 89 NA  
## 90 NA  
## 91 60.23000  
## 92 61.68600  
## 93 50.84200  
## 94 56.65457  
## 95 58.59600  
## 96 64.94900  
## 97 67.70086  
## 98 44.74200  
## 99 58.34700  
## 100 52.90300  
## 101 58.47836  
## 102 59.00001  
## 103 60.95081  
## 104 54.94984  
## 105 63.63800  
## 106 63.95577  
## 107 NA  
## 108 52.03900  
## 109 NA  
## 110 55.62000  
## 111 38.82400  
## 112 42.60700  
## 113 75.42200  
## 114 61.14400  
## 115 43.13600  
## 116 58.18100  
## 117 33.18100  
## 118 58.56400  
## 119 67.48100  
## 120 57.90400  
## 121 57.17500  
## 122 84.21300  
## 123 NA  
## 124 NA  
## 125 60.31800  
## 126 68.05300  
## 127 59.65700  
## 128 77.62900  
## 129 44.04200  
## 130 54.41600  
## 131 42.79500  
## 132 53.69100  
## 133 59.16340  
## 134 64.71419  
## 135 68.25517  
## 136 NA  
## 137 51.40900  
## 138 55.10973  
## 139 59.13388  
## 140 47.91500  
## 141 64.03702  
## 142 55.77000  
## 143 54.31900  
## 144 54.87700  
## 145 69.44900  
## 146 NA  
## 147 44.47300  
## 148 NA  
## 149 41.00700  
## 150 84.78900  
## 151 62.02900  
## 152 43.35142  
## 153 58.90200  
## 154 NA  
## 155 58.20892  
## 156 41.93100  
## 157 65.64300  
## 158 52.03400  
## 159 64.80000  
## 160 39.79076  
## 161 40.06200  
## 162 54.99100  
## 163 NA  
## 164 59.01500  
## 165 44.54800  
## 166 54.66100  
## 167 72.32800  
## 168 62.11000  
## 169 59.30812  
## 170 46.95500  
## 171 53.65400  
## 172 78.64900  
## 173 51.25700  
## 174 63.53400  
## 175 59.88400  
## 176 61.35900  
## 177 81.70700  
## 178 NA  
## 179 65.76800  
## 180 56.09797  
## 181 67.55700  
## 182 57.19620  
## 183 52.28200  
## 184 63.99800  
## 185 73.58300  
## 186 60.55800  
## 187 NA  
## 188 68.13600  
## 189 53.35800  
## 190 60.99151  
## 191 36.79900  
## 192 76.68100  
## 193 52.06500  
## 194 66.61800  
## 195 33.31200  
## 196 58.20431  
## 197 56.17611  
## 198 43.53000  
## 199 87.01900  
## 200 50.41500  
## 201 60.29600  
## 202 85.15200  
## 203 52.67405  
## 204 52.72700  
## 205 40.65100  
## 206 54.23100  
## 207 67.42900  
## 208 70.04500  
## 209 55.38100  
## 210 58.40600  
## 211 NA  
## 212 43.30800  
## 213 45.65100  
## 214 63.27177  
## 215 63.97400  
## 216 63.27177  
## 217 57.31549  
## 218 50.37000  
## 219 49.02800  
## 220 54.24600  
## 221 52.14300  
## 222 59.52100  
## 223 39.79600  
## 224 NA  
## 225 NA  
## 226 35.33000  
## 227 NA  
## 228 67.04700  
## 229 66.17902  
## 230 54.82683  
## 231 76.26400  
## 232 68.12200  
## 233 53.09100  
## 234 63.21200  
## 235 59.42001  
## 236 37.71000  
## 237 39.86063  
## 238 58.80300  
## 239 52.67405  
## 240 63.27177  
## 241 59.58700  
## 242 39.76000  
## 243 46.16000  
## 244 NA  
## 245 81.49800  
## 246 69.17400  
## 247 49.35300  
## 248 61.31300  
## 249 59.59300  
## 250 59.13300  
## 251 60.60500  
## 252 56.88000  
## 253 59.09600  
## 254 NA  
## 255 56.46400  
## 256 76.56000  
## 257 66.80300  
## 258 58.66954  
## 259 28.93200  
## 260 NA  
## 261 32.75400  
## 262 40.18000  
## 263 69.02500  
## 264 79.10700

# View(countries\_2016)

# 1. Rename the appropriate columns to “country”, “gdp”, “life\_expectancy”, and “employment”.

names(countries\_2016) <- c("country", "gdp", "life\_expectancy","employment")

# 2. Convert the employment number to percentages by dividing by 100

countries\_2016$employment <- countries\_2016$employment/100

# 3. Then round life expectancy to zero decimals and employment to two decimals

# round life expectancy to zero decimals  
  
countries\_2016$life\_expectancy <- round(countries\_2016$life\_expectancy,0)

# round employment to two decimals  
  
countries\_2016$employment <- round(countries\_2016$employment,2)

# Displaying post changes   
  
print(countries\_2016)

## country gdp  
## 1 Aruba NA  
## 2 Afghanistan 1.946902e+10  
## 3 Angola 9.533720e+10  
## 4 Albania 1.188368e+10  
## 5 Andorra 2.877312e+09  
## 6 Arab World 2.500164e+12  
## 7 United Arab Emirates 3.570451e+11  
## 8 Argentina 5.548609e+11  
## 9 Armenia 1.054614e+10  
## 10 American Samoa 6.580000e+08  
## 11 Antigua and Barbuda 1.460145e+09  
## 12 Australia 1.208039e+12  
## 13 Austria 3.908000e+11  
## 14 Azerbaijan 3.786752e+10  
## 15 Burundi 3.007029e+09  
## 16 Belgium 4.675455e+11  
## 17 Benin 8.573160e+09  
## 18 Burkina Faso 1.144878e+10  
## 19 Bangladesh 2.214152e+11  
## 20 Bulgaria 5.324089e+10  
## 21 Bahrain 3.215269e+10  
## 22 Bahamas, The 1.183880e+10  
## 23 Bosnia and Herzegovina 1.691109e+10  
## 24 Belarus 4.772266e+10  
## 25 Belize 1.820159e+09  
## 26 Bermuda NA  
## 27 Bolivia 3.394113e+10  
## 28 Brazil 1.793989e+12  
## 29 Barbados 4.529050e+09  
## 30 Brunei Darussalam 1.140065e+10  
## 31 Bhutan 2.212639e+09  
## 32 Botswana 1.564870e+10  
## 33 Central African Republic 1.755468e+09  
## 34 Canada 1.535768e+12  
## 35 Central Europe and the Baltics 1.313067e+12  
## 36 Switzerland 6.687453e+11  
## 37 Channel Islands NA  
## 38 Chile 2.500362e+11  
## 39 China 1.119099e+13  
## 40 Cote d'Ivoire 3.637485e+10  
## 41 Cameroon 3.221754e+10  
## 42 Congo, Dem. Rep. 3.499116e+10  
## 43 Congo, Rep. 7.833509e+09  
## 44 Colombia 2.800910e+11  
## 45 Comoros 6.166545e+08  
## 46 Cabo Verde 1.638927e+09  
## 47 Costa Rica 5.698899e+10  
## 48 Caribbean small states 6.778768e+10  
## 49 Cuba NA  
## 50 Curacao NA  
## 51 Cayman Islands NA  
## 52 Cyprus 2.015387e+10  
## 53 Czech Republic 1.953051e+11  
## 54 Germany 3.477796e+12  
## 55 Djibouti 1.764268e+09  
## 56 Dominica 5.814840e+08  
## 57 Denmark 3.068997e+11  
## 58 Dominican Republic 7.234297e+10  
## 59 Algeria 1.590491e+11  
## 60 East Asia & Pacific (excluding high income) 1.350933e+13  
## 61 Early-demographic dividend 1.043743e+13  
## 62 East Asia & Pacific 2.251217e+13  
## 63 Europe & Central Asia (excluding high income) 2.943297e+12  
## 64 Europe & Central Asia 2.028071e+13  
## 65 Ecuador 9.861397e+10  
## 66 Egypt, Arab Rep. 3.329278e+11  
## 67 Euro area 1.193474e+13  
## 68 Eritrea NA  
## 69 Spain 1.237255e+12  
## 70 Estonia 2.333791e+10  
## 71 Ethiopia 7.300098e+10  
## 72 European Union 1.649186e+13  
## 73 Fragile and conflict affected situations 7.240911e+11  
## 74 Finland 2.386777e+11  
## 75 Fiji 4.671313e+09  
## 76 France 2.465134e+12  
## 77 Faroe Islands NA  
## 78 Micronesia, Fed. Sts. 3.298956e+08  
## 79 Gabon 1.401428e+10  
## 80 United Kingdom 2.650850e+12  
## 81 Georgia 1.437802e+10  
## 82 Ghana 4.280358e+10  
## 83 Gibraltar NA  
## 84 Guinea 9.275886e+09  
## 85 Gambia, The 9.627971e+08  
## 86 Guinea-Bissau 1.178205e+09  
## 87 Equatorial Guinea 1.125956e+10  
## 88 Greece 1.926908e+11  
## 89 Grenada 1.056189e+09  
## 90 Greenland 2.705892e+09  
## 91 Guatemala 6.866365e+10  
## 92 Guam 5.793000e+09  
## 93 Guyana 3.504024e+09  
## 94 High income 4.928186e+13  
## 95 Hong Kong SAR, China 3.208812e+11  
## 96 Honduras 2.164394e+10  
## 97 Heavily indebted poor countries (HIPC) 6.521759e+11  
## 98 Croatia 5.133852e+10  
## 99 Haiti 7.970649e+09  
## 100 Hungary 1.258166e+11  
## 101 IBRD only 2.602546e+13  
## 102 IDA & IBRD total 2.800566e+13  
## 103 IDA total 1.988820e+12  
## 104 IDA blend 9.240355e+11  
## 105 Indonesia 9.322565e+11  
## 106 IDA only 1.060272e+12  
## 107 Isle of Man NA  
## 108 India 2.274230e+12  
## 109 Not classified NA  
## 110 Ireland 3.048190e+11  
## 111 Iran, Islamic Rep. 4.189767e+11  
## 112 Iraq 1.714890e+11  
## 113 Iceland 2.030410e+10  
## 114 Israel 3.177475e+11  
## 115 Italy 1.859384e+12  
## 116 Jamaica 1.405691e+10  
## 117 Jordan 3.865473e+10  
## 118 Japan 4.949273e+12  
## 119 Kazakhstan 1.372783e+11  
## 120 Kenya 7.087529e+10  
## 121 Kyrgyz Republic 6.813092e+09  
## 122 Cambodia 2.001675e+10  
## 123 Kiribati 1.815515e+08  
## 124 St. Kitts and Nevis 9.098546e+08  
## 125 Korea, Rep. 1.414804e+12  
## 126 Kuwait 1.109123e+11  
## 127 Latin America & Caribbean (excluding high income) 4.263792e+12  
## 128 Lao PDR 1.580571e+10  
## 129 Lebanon 4.959883e+10  
## 130 Liberia 2.101000e+09  
## 131 Libya 3.225717e+10  
## 132 St. Lucia 1.667079e+09  
## 133 Latin America & Caribbean 5.360659e+12  
## 134 Least developed countries: UN classification 9.296720e+11  
## 135 Low income 5.017795e+11  
## 136 Liechtenstein NA  
## 137 Sri Lanka 8.178838e+10  
## 138 Lower middle income 6.036799e+12  
## 139 Low & middle income 2.666356e+13  
## 140 Lesotho 2.291320e+09  
## 141 Late-demographic dividend 1.846759e+13  
## 142 Lithuania 4.277303e+10  
## 143 Luxembourg 5.863132e+10  
## 144 Latvia 2.757151e+10  
## 145 Macao SAR, China 4.531088e+10  
## 146 St. Martin (French part) NA  
## 147 Morocco 1.036063e+11  
## 148 Monaco NA  
## 149 Moldova 6.795742e+09  
## 150 Madagascar 1.000119e+10  
## 151 Maldives 4.222767e+09  
## 152 Middle East & North Africa 3.142484e+12  
## 153 Mexico 1.076912e+12  
## 154 Marshall Islands 1.944979e+08  
## 155 Middle income 2.615783e+13  
## 156 Macedonia, FYR 1.074579e+10  
## 157 Mali 1.403498e+10  
## 158 Malta 1.127954e+10  
## 159 Myanmar 6.322510e+10  
## 160 Middle East & North Africa (excluding high income) 1.444679e+12  
## 161 Montenegro 4.374131e+09  
## 162 Mongolia 1.118346e+10  
## 163 Northern Mariana Islands 1.242000e+09  
## 164 Mozambique 1.101486e+10  
## 165 Mauritania 4.739298e+09  
## 166 Mauritius 1.223246e+10  
## 167 Malawi 5.433039e+09  
## 168 Malaysia 2.965359e+11  
## 169 North America 2.016631e+13  
## 170 Namibia 1.130923e+10  
## 171 New Caledonia NA  
## 172 Niger 7.606749e+09  
## 173 Nigeria 4.046527e+11  
## 174 Nicaragua 1.318499e+10  
## 175 Netherlands 7.772275e+11  
## 176 Norway 3.710752e+11  
## 177 Nepal 2.113198e+10  
## 178 Nauru 1.020601e+08  
## 179 New Zealand 1.892860e+11  
## 180 OECD members 4.765822e+13  
## 181 Oman 6.682445e+10  
## 182 Other small states 3.644192e+11  
## 183 Pakistan 2.786546e+11  
## 184 Panama 5.782092e+10  
## 185 Peru 1.916397e+11  
## 186 Philippines 3.048891e+11  
## 187 Palau 3.026998e+08  
## 188 Papua New Guinea 1.990481e+10  
## 189 Poland 4.714003e+11  
## 190 Pre-demographic dividend 1.207094e+12  
## 191 Puerto Rico 1.050345e+11  
## 192 Korea, Dem. People\xd5s Rep. NA  
## 193 Portugal 2.051845e+11  
## 194 Paraguay 2.742407e+10  
## 195 West Bank and Gaza 1.342570e+10  
## 196 Pacific island small states 9.027151e+09  
## 197 Post-demographic dividend 4.525737e+13  
## 198 French Polynesia NA  
## 199 Qatar 1.524519e+11  
## 200 Romania 1.878059e+11  
## 201 Russian Federation 1.284728e+12  
## 202 Rwanda 8.475682e+09  
## 203 South Asia 2.903124e+12  
## 204 Saudi Arabia 6.449355e+11  
## 205 Sudan 9.558438e+10  
## 206 Senegal 1.468375e+10  
## 207 Singapore 3.097639e+11  
## 208 Solomon Islands 1.232699e+09  
## 209 Sierra Leone 3.556037e+09  
## 210 El Salvador 2.391223e+10  
## 211 San Marino 1.590708e+09  
## 212 Somalia 6.752653e+09  
## 213 Serbia 3.829985e+10  
## 214 Sub-Saharan Africa (excluding high income) 1.511047e+12  
## 215 South Sudan 2.904115e+09  
## 216 Sub-Saharan Africa 1.512473e+12  
## 217 Small states 4.412340e+11  
## 218 Sao Tome and Principe 3.542381e+08  
## 219 Suriname 3.278425e+09  
## 220 Slovak Republic 8.976860e+10  
## 221 Slovenia 4.470860e+10  
## 222 Sweden 5.144600e+11  
## 223 Eswatini 3.720649e+09  
## 224 Sint Maarten (Dutch part) NA  
## 225 Seychelles 1.425929e+09  
## 226 Syrian Arab Republic NA  
## 227 Turks and Caicos Islands NA  
## 228 Chad 9.412034e+09  
## 229 East Asia & Pacific (IDA & IBRD countries) 1.348339e+13  
## 230 Europe & Central Asia (IDA & IBRD countries) 3.466036e+12  
## 231 Togo 4.388570e+09  
## 232 Thailand 4.117552e+11  
## 233 Tajikistan 6.951657e+09  
## 234 Turkmenistan 3.617989e+10  
## 235 Latin America & the Caribbean (IDA & IBRD countries) 5.134442e+12  
## 236 Timor-Leste 2.521008e+09  
## 237 Middle East & North Africa (IDA & IBRD countries) 1.431082e+12  
## 238 Tonga 4.015620e+08  
## 239 South Asia (IDA & IBRD) 2.903124e+12  
## 240 Sub-Saharan Africa (IDA & IBRD countries) 1.512473e+12  
## 241 Trinidad and Tobago 2.232001e+10  
## 242 Tunisia 4.206255e+10  
## 243 Turkey 8.637216e+11  
## 244 Tuvalu 3.657261e+07  
## 245 Tanzania 4.738840e+10  
## 246 Uganda 2.407893e+10  
## 247 Ukraine 9.327048e+10  
## 248 Upper middle income 2.010892e+13  
## 249 Uruguay 5.268761e+10  
## 250 United States 1.862448e+13  
## 251 Uzbekistan 6.706757e+10  
## 252 St. Vincent and the Grenadines 7.655556e+08  
## 253 Venezuela, RB NA  
## 254 British Virgin Islands NA  
## 255 Virgin Islands (U.S.) NA  
## 256 Vietnam 2.052762e+11  
## 257 Vanuatu 7.879426e+08  
## 258 World 7.593681e+13  
## 259 Samoa 7.863563e+08  
## 260 Kosovo 6.715487e+09  
## 261 Yemen, Rep. 1.821333e+10  
## 262 South Africa 2.957627e+11  
## 263 Zambia 2.095475e+10  
## 264 Zimbabwe 1.661996e+10  
## life\_expectancy employment  
## 1 76 NA  
## 2 64 0.49  
## 3 62 0.72  
## 4 78 0.48  
## 5 NA NA  
## 6 71 0.44  
## 7 77 0.79  
## 8 77 0.55  
## 9 75 0.49  
## 10 NA NA  
## 11 76 NA  
## 12 82 0.61  
## 13 81 0.57  
## 14 72 0.63  
## 15 57 0.78  
## 16 81 0.49  
## 17 61 0.69  
## 18 60 0.62  
## 19 72 0.54  
## 20 75 0.49  
## 21 77 0.71  
## 22 76 0.66  
## 23 77 0.35  
## 24 74 0.64  
## 25 70 0.62  
## 26 81 NA  
## 27 69 0.65  
## 28 76 0.56  
## 29 76 0.60  
## 30 77 0.63  
## 31 70 0.65  
## 32 67 0.59  
## 33 52 0.67  
## 34 82 0.61  
## 35 77 0.53  
## 36 83 0.65  
## 37 81 0.52  
## 38 80 0.58  
## 39 76 0.66  
## 40 54 0.56  
## 41 58 0.73  
## 42 60 0.70  
## 43 65 0.62  
## 44 74 0.64  
## 45 64 0.41  
## 46 73 0.54  
## 47 80 0.54  
## 48 73 0.58  
## 49 80 0.53  
## 50 78 NA  
## 51 NA NA  
## 52 81 0.55  
## 53 78 0.58  
## 54 81 0.58  
## 55 62 0.55  
## 56 NA NA  
## 57 81 0.59  
## 58 74 0.63  
## 59 76 0.37  
## 60 74 0.66  
## 61 70 0.55  
## 62 75 0.66  
## 63 73 0.55  
## 64 77 0.54  
## 65 76 0.65  
## 66 71 0.42  
## 67 82 0.51  
## 68 65 0.76  
## 69 83 0.47  
## 70 78 0.59  
## 71 65 0.78  
## 72 81 0.53  
## 73 63 0.56  
## 74 82 0.53  
## 75 70 0.55  
## 76 82 0.50  
## 77 82 NA  
## 78 69 NA  
## 79 66 0.41  
## 80 81 0.59  
## 81 73 0.60  
## 82 63 0.75  
## 83 NA NA  
## 84 60 0.61  
## 85 61 0.54  
## 86 57 0.67  
## 87 58 0.56  
## 88 81 0.41  
## 89 74 NA  
## 90 NA NA  
## 91 73 0.60  
## 92 80 0.62  
## 93 67 0.51  
## 94 80 0.57  
## 95 84 0.59  
## 96 74 0.65  
## 97 62 0.68  
## 98 78 0.45  
## 99 63 0.58  
## 100 76 0.53  
## 101 73 0.58  
## 102 70 0.59  
## 103 64 0.61  
## 104 62 0.55  
## 105 69 0.64  
## 106 65 0.64  
## 107 NA NA  
## 108 69 0.52  
## 109 NA NA  
## 110 82 0.56  
## 111 76 0.39  
## 112 70 0.43  
## 113 82 0.75  
## 114 82 0.61  
## 115 83 0.43  
## 116 76 0.58  
## 117 74 0.33  
## 118 84 0.59  
## 119 72 0.67  
## 120 67 0.58  
## 121 71 0.57  
## 122 69 0.84  
## 123 66 NA  
## 124 NA NA  
## 125 82 0.60  
## 126 75 0.68  
## 127 75 0.60  
## 128 67 0.78  
## 129 80 0.44  
## 130 63 0.54  
## 131 72 0.43  
## 132 75 0.54  
## 133 76 0.59  
## 134 64 0.65  
## 135 63 0.68  
## 136 83 NA  
## 137 75 0.51  
## 138 68 0.55  
## 139 70 0.59  
## 140 54 0.48  
## 141 76 0.64  
## 142 74 0.56  
## 143 82 0.54  
## 144 75 0.55  
## 145 84 0.69  
## 146 80 NA  
## 147 76 0.44  
## 148 NA NA  
## 149 72 0.41  
## 150 66 0.85  
## 151 77 0.62  
## 152 73 0.43  
## 153 77 0.59  
## 154 NA NA  
## 155 71 0.58  
## 156 76 0.42  
## 157 58 0.66  
## 158 82 0.52  
## 159 67 0.65  
## 160 73 0.40  
## 161 77 0.40  
## 162 69 0.55  
## 163 NA NA  
## 164 58 0.59  
## 165 63 0.45  
## 166 74 0.55  
## 167 63 0.72  
## 168 75 0.62  
## 169 79 0.59  
## 170 64 0.47  
## 171 77 0.54  
## 172 60 0.79  
## 173 53 0.51  
## 174 75 0.64  
## 175 82 0.60  
## 176 83 0.61  
## 177 70 0.82  
## 178 NA NA  
## 179 82 0.66  
## 180 80 0.56  
## 181 77 0.68  
## 182 68 0.57  
## 183 66 0.52  
## 184 78 0.64  
## 185 75 0.74  
## 186 69 0.61  
## 187 NA NA  
## 188 66 0.68  
## 189 77 0.53  
## 190 60 0.61  
## 191 80 0.37  
## 192 72 0.77  
## 193 81 0.52  
## 194 73 0.67  
## 195 73 0.33  
## 196 71 0.58  
## 197 80 0.56  
## 198 77 0.44  
## 199 78 0.87  
## 200 75 0.50  
## 201 72 0.60  
## 202 67 0.85  
## 203 69 0.53  
## 204 75 0.53  
## 205 64 0.41  
## 206 67 0.54  
## 207 83 0.67  
## 208 71 0.70  
## 209 52 0.55  
## 210 74 0.58  
## 211 NA NA  
## 212 56 0.43  
## 213 75 0.46  
## 214 60 0.63  
## 215 57 0.64  
## 216 60 0.63  
## 217 69 0.57  
## 218 67 0.50  
## 219 71 0.49  
## 220 77 0.54  
## 221 81 0.52  
## 222 82 0.60  
## 223 58 0.40  
## 224 NA NA  
## 225 74 NA  
## 226 70 0.35  
## 227 NA NA  
## 228 53 0.67  
## 229 75 0.66  
## 230 73 0.55  
## 231 60 0.76  
## 232 75 0.68  
## 233 71 0.53  
## 234 68 0.63  
## 235 75 0.59  
## 236 69 0.38  
## 237 73 0.40  
## 238 73 0.59  
## 239 69 0.53  
## 240 60 0.63  
## 241 71 0.60  
## 242 76 0.40  
## 243 76 0.46  
## 244 NA NA  
## 245 66 0.81  
## 246 60 0.69  
## 247 71 0.49  
## 248 75 0.61  
## 249 77 0.60  
## 250 79 0.59  
## 251 71 0.61  
## 252 73 0.57  
## 253 75 0.59  
## 254 NA NA  
## 255 79 0.56  
## 256 76 0.77  
## 257 72 0.67  
## 258 72 0.59  
## 259 75 0.29  
## 260 72 NA  
## 261 65 0.33  
## 262 63 0.40  
## 263 62 0.69  
## 264 61 0.79

# 4. Create a frequency table for each variable

# frequency table for GDP variable   
  
gdp\_freq <- table(countries\_2016$gdp)  
  
print("Frequency Table for GDP:")

## [1] "Frequency Table for GDP:"

print(gdp\_freq)

##   
## 36572611.89 102060129.58 181551516.5 194497900   
## 1 1 1 1   
## 302699800 329895600 354238109.3 401562006.23   
## 1 1 1 1   
## 581484037.04 616654490.41 6.58e+08 765555555.56   
## 1 1 1 1   
## 786356314.8 787942567.41 909854629.63 962797146.5   
## 1 1 1 1   
## 1056188592.59 1178204501.44 1232699140.37 1.242e+09   
## 1 1 1 1   
## 1425929444.33 1460144703.7 1590707964.6 1638927335.62   
## 1 1 1 1   
## 1667078703.7 1755468136.87 1764268468.94 1820158550   
## 1 1 1 1   
## 2.101e+09 2212638830.39 2291319971.84 2521007678.83   
## 1 1 1 1   
## 2705891616.98 2877311946.9 2904114903.23 3007029030.4   
## 1 1 1 1   
## 3278425328.32 3504024213.08 3556036534.57 3720649374.58   
## 1 1 1 1   
## 4222767412.65 4374130530.97 4388569576.44 4529050000   
## 1 1 1 1   
## 4671313314.56 4739298311.39 5433038646.52 5.793e+09   
## 1 1 1 1   
## 6715486725.66 6752653098.65 6795741776.17 6813092065.84   
## 1 1 1 1   
## 6951657158.9 7606749314.04 7833508878.97 7970649131.23   
## 1 1 1 1   
## 8475681532.52 8573159696.61 9027150901.34 9275886117.43   
## 1 1 1 1   
## 9412034268.67 10001193315.2 10546135160.03 10745787406.45   
## 1 1 1 1   
## 11014862241.73 11183458130.81 11259559879.05 11279535398.23   
## 1 1 1 1   
## 11309232187.53 11400653959.84 11448781714.31 11838800000   
## 1 1 1 1   
## 11883682170.82 12232463655.57 13184989877.57 13425700000   
## 1 1 1 1   
## 14014278017.47 14034980333.66 14056908749.35 14378016732.16   
## 1 1 1 1   
## 14683747153.66 15648700274.48 15805707154.23 16619960400   
## 1 1 1 1   
## 16911088173.75 18213328571.43 19469022207.69 19904808311.62   
## 1 1 1 1   
## 20016747754.02 20153871681.42 20304098101.41 20954754378.14   
## 1 1 1 1   
## 21131983246.19 21643936938.91 22320008403.11 23337907618.52   
## 1 1 1 1   
## 23912227500 24078931744.41 27424071373.05 27571513793.4   
## 1 1 1 1   
## 32152686170.21 32217537942.66 32257171354.08 33941126193.92   
## 1 1 1 1   
## 34991160099.74 36179885714.29 36374849865.05 37867518957.2   
## 1 1 1 1   
## 38299854688.13 38654727746.48 42062549394.79 42773029835.32   
## 1 1 1 1   
## 42803583022.15 44708598648.86 45310877912.72 47388395823.4   
## 1 1 1 1   
## 47722657820.67 49598825982.16 51338524831.03 52687612261.54   
## 1 1 1 1   
## 53240893665.16 56988989896.64 57820916600 58631324559.45   
## 1 1 1 1   
## 63225097051.26 66824447334.2 67067565988.64 67787681466.08   
## 1 1 1 1   
## 68663653469.12 70875289605.38 72342967648.34 73000980433.95   
## 1 1 1 1   
## 81788375089.98 89768598023.39 93270479388.52 95337203468.12   
## 1 1 1 1   
## 95584380032.21 98613972000 103606321692.58 105034500000   
## 1 1 1 1   
## 110912280701.75 125816640420.57 137278320084.17 152451923076.92   
## 1 1 1 1   
## 159049096745.25 171489001692.05 187805922349.23 189285950470.75   
## 1 1 1 1   
## 191639655121.33 192690813126.86 195305084919.14 205184480409.02   
## 1 1 1 1   
## 205276172134.9 221415162445.65 238677672281.61 250036180921.05   
## 1 1 1 1   
## 278654637737.69 280090999648.12 295762685147.67 296535930381.12   
## 1 1 1 1   
## 304819018067.11 304889079564.68 306899653409.6 309763879840.75   
## 1 1 1 1   
## 317747542489.39 320881182123.86 332927833278.04 357045064669.84   
## 1 1 1 1   
## 364419151281.06 371075238095.24 390799991147.47 404652720164.9   
## 1 1 1 1   
## 411755164832.67 418976711586.86 441233983648.48 467545548764.57   
## 1 1 1 1   
## 471400273917.01 501779518531.24 514459972806.17 554860945013.62   
## 1 1 1 1   
## 644935541446.45 652175948756.27 668745279604.53 724091067411.03   
## 1 1 1 1   
## 777227541581.31 863721647958.68 924035470840.17 929671974133.33   
## 1 1 1 1   
## 932256495234.25 1060271611687.89 1076912039691.17 1207093555741.69   
## 1 1 1 1   
## 1208039015868.39 1237255019653.86 1284727602173.71 1313066988021.63   
## 1 1 1 1   
## 1414804158515.26 1431082428979.52 1444678670901.96 1511046712038.01   
## 1 1 1 1   
## 1512472520978.19 1535767736946.18 1793989048409.29 1859383610248.72   
## 2 1 1 1   
## 1988820003334.38 2274229710530.03 2465134297438.91 2500164034395.78   
## 1 1 1 1   
## 2650850178102.14 2903124297500.27 2943297166689.13 3142484003145.08   
## 1 2 1 1   
## 3466035965437.18 3477796274496.8 4263791849337.86 4949273341993.88   
## 1 1 1 1   
## 5134441961597.49 5360659105919.66 6036798782954.65 10437433411218.3   
## 1 1 1 1   
## 11190992550229.5 11934738646774.8 13483389369359.2 13509334237575.9   
## 1 1 1 1   
## 16491855791194.9 18467589315232.1 1.8624475e+13 20108923623800.1   
## 1 1 1 1   
## 20166305457752 20280711671665.5 22512169652487.2 26025458644283.5   
## 1 1 1 1   
## 26157827164876.7 26663561044210.5 28005663818401.2 45257369047361.3   
## 1 1 1 1   
## 47658224273925.7 49281856158945.9 75936811478760.2   
## 1 1 1

# frequency table for life\_expectancy variable   
  
life\_exp\_freq <- table(countries\_2016$life\_expectancy)  
  
print("Frequency Table for Life Expectancy:")

## [1] "Frequency Table for Life Expectancy:"

print(life\_exp\_freq)

##   
## 52 53 54 56 57 58 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79   
## 2 2 2 1 3 5 10 3 5 8 6 5 6 8 3 11 9 11 10 14 11 20 18 16 7 3   
## 80 81 82 83 84   
## 10 12 15 6 3

# frequency table for employment variable   
  
employment\_freq <- table(countries\_2016$employment)  
  
print("Frequency Table for Employment:")

## [1] "Frequency Table for Employment:"

print(employment\_freq)

##   
## 0.29 0.33 0.35 0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49   
## 1 3 2 2 1 1 6 5 2 5 4 2 2 2 2 6   
## 0.5 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62 0.63 0.64 0.65   
## 3 4 6 10 11 13 9 6 11 17 10 10 6 7 8 7   
## 0.66 0.67 0.68 0.69 0.7 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.81 0.82   
## 7 7 6 4 2 1 2 1 1 2 2 2 3 3 1 1   
## 0.84 0.85 0.87   
## 1 2 1

# 5. Draw histograms for each variable

# Removing missing values from dataset : non-finite values  
  
countries\_2016\_clean <- na.omit(countries\_2016)

# Dividing GDP by trillion  
  
countries\_2016\_clean$gdp <- countries\_2016\_clean$gdp/1000000000000

# Set up a multi-panel plot - 3 graphs in a single row on one page  
  
par(mfrow = c(1, 3))  
  
# GDP histogram  
  
hist(countries\_2016\_clean$gdp,  
 col = "#AEB6E9",  
 main = "Distribution of GDP (2016)",  
 xlab = "GDP (in trillions of USD)",  
 ylab = "Frequency")  
  
# Life expectancy histogram  
  
hist(countries\_2016\_clean$life\_expectancy,   
 breaks = "Sturges",  
 col = "#F9C77E",  
 main = "Distribution of Life Expectancy (2016)",  
 xlab = "Life Expectancy",  
 ylab = "Frequency")  
  
# Employment histogram  
  
hist(countries\_2016\_clean$employment,   
 breaks = "Sturges",  
 col = "#9ED9CC",  
 main = "Distribution of Employment (2016)",  
 xlab = "Employment (%)",  
 ylab = "Frequency")

