```
# JRodoni_HW06_script.R
# C:/Users/jackr/OneDrive/Desktop/Graduate School Courses/
#   STAT 604 - STAT Computation/Homeworks/JRodoni_HW06_script.R
# Created By: Jack Rodoni
# Creation Date: 09/27/2021
# Purpose: STAT 604 Homework 6
# Last Executed: 09/28/2021



# Prior to starting your script, execute in the console the function that will
display all the graphics
# parameters. Locate the parameter that defines the graph margin in inches.
Write down the margin
# values so that you can refer to them later in the assignment.

par()
par(mai = c(1.02,0.82,0.82,0.42)) #= C(bottom, left, top, right)
#mai = c(1.02,0.82,0.82,0.42) = C(bottom, left, top, right)
#mar = c(5.1,4.1,4.1,2.1)
#oma = c(0,0,0,0)
#omi = c(0,0,0,0)



# 1.) After the header, include housekeeping steps as you did in the previous
assignments.

Sys.time()

ls()
rm(list = ls())
library()
search()

# 2.) Write an expression in your script to load the workspace from the
previous assignment. Show
#     the contents of the workspace. Display a summary of the data frame
containing data as of
#     September 12.

load(paste("C:/Users/jackr/OneDrive/Desktop/Graduate School Courses/",
            "STAT 604 - STAT Computation/Homeworks/HW05.RData", sep = ""))
ls() # show the contents of the workspace

summary(Merged_df_Latest_NAsRemoved)

# 3.) On an assignment statement, use the with function to access the columns
in the September 12
#     data frame and create a new column containing the death rate. Death rate
is calculated as Total
#     Deaths divided by Total Cases then multiplied by 100 so it is displayed
as a number between 0
```

```
#     and 100. This expression will be one of the arguments in the with 
function. Write expressions
#     to show the minimum value and maximum value of the new column

Data_Latest = Merged_df_Latest_NAsRemoved
Data_Latest$DeathRate = with(Data_Latest, (TOTAL_DEATHS/TOTAL_CASES)*100)
min(Data_Latest$DeathRate)
max(Data_Latest$DeathRate)

# 4.) Use a line of code to direct all graphic output to your PDF document. 
Research the available
#     arguments for this function and set width to 11 and height to 8.5 so it 
will fit a normal size paper
#     in landscape orientation. (You may want to wait until you have your 
graphics working correctly
#     before you add the line to redirect to PDF so you can see the results in 
your R session.)

pdf(paste("C:/Users/jackr/OneDrive/Desktop/Graduate School Courses/",
          "STAT 604 - STAT Computation/Homeworks/JRodoni_HW6_graphics.pdf", 
sep = ""),
    width = 11, height = 8.5)


# 5.) Create a histogram of the death rate column you created above, forcing 
the cells to have a
#     width of 0.5. Start the breaks at the minimum death rate and continue to 
the next integer
#     above the maximum death rate. You may hard code the start and end values 
when setting up
#     your break points. (The term “hard coding” refers to entering an actual 
value like 50 in your
#     program code instead of using a formula.) Create the histogram in a 
manner that will facilitate
#     the addition of a distribution curve later. Label the X axis “Percent” 
and supply an appropriate
#     main title for the graph

hist(Data_Latest$DeathRate, breaks = seq(0,8,.5),
     freq = FALSE, xlim = c(0,8), ylim = c(0,.5), xlab = "Percent", main = 
"COVID Death Rates")


length(seq(min(Data_Latest$DeathRate), ceiling(max(Data_Latest$DeathRate)), by 
= 0.5))
# 6.) Add to the graph a line that shows the normal distribution density of 
death rate values. Include
#     arguments that will ensure calculations are made even when there are 
missing values in the
#     data. Use a hex value to “mix” a color for the line that has a Red 
amount of 22, a Green amount
#     of A0 and a Blue amount of EE.

x = seq(from = 0, to = 8, by = .001)
```

```
y = dnorm(x, mean = mean(Data_Latest$DeathRate), sd =
sd(Data_Latest$DeathRate))
lines(x,y, col = "#22A0EE")

#### still need to fix the density line, not sure what he wants

# 7.) Draw a vertical line at the mean death rate value. Use the second color
in the R palette as the
#     color of the line. Use a function to determine the position of the line
instead of hard coding the
#     current mean value. Include an argument to ensure the mean is calculated
even if there are
#     missing values. Draw a line at the median in the same manner except use
the color name
#     green1 to specify the line color

abline(v = mean(Data_Latest$DeathRate, na.rm = TRUE), col = "#DF536B")
abline(v = median(Data_Latest$DeathRate, na.rm = TRUE), col = "green1")

# 8.) Display in the console the names of all available R colors

colors()

# 9.) We want to observe the correlation between the total number of cases and
the total number of
#     deaths from each county in the September 12 data. Plot a point for each
county with data using
#     total cases for the x axis and total deaths for the y axis. Use the
diamond plot character (◊).
#     Pick an unusual name that sounds interesting to you from the list of
colors as the color of your
#     points. Any color is acceptable if the points show up well. Supply
appropriate labels for the
#     axes and an appropriate title for the graph

plot(x = Data_Latest$TOTAL_CASES, y = Data_Latest$TOTAL_DEATHS, pch = 5, col =
"darkturquoise",
     xlab = "Total Cases", ylab = "Total Deaths", xaxt = "n",
     main = "Total Cases Vs Total Deaths")
axis(1, at = seq(0,500000,100000), labels =
c("0","100k","200k","300k","400k","500k"))

# 10.) Add a fit line to the plot

lm1 = lm(Data_Latest$TOTAL_DEATHS~Data_Latest$TOTAL_CASES)
abline(lm1)

# 11.) Use functions to imbed text showing the date and time of creation in
the upper left-hand corner
#      of the graph area. The exact value of the y coordinate for the time
stamp location is not critical
#      if the time stamp is near the corner. You may hard code the coordinates
but use 0 as the x
```

```
#      coordinate and use an alignment value so the text starts at 0. The date
and time must
#      automatically change each time the script is run.

text(0,7500, Sys.time(), adj = 0)
# legend("topleft", legend = Sys.time())

# 12.)  Use logic expressions as an index parameter to create a new data frame
that is a subset of the
#       Texas COVID data frame where the population of the county is not
missing and is greater than
#       500 thousand and the value of the date column created in the previous
assignment is greater
#       than March 14, 2020. When you hard code the date value in your
comparison statement,
#       coerce it to a date so you can be sure R is comparing two values of
the Date class. Include all
#       columns in the subset. Display a summary of the new data frame. Use
the tapply function to
#       display a table showing the median number of New Cases for each county
in the data frame.
#       There should be 12 Counties displayed and the value for Bexar should
be 171.


NewDf = subset(Merged_df, is.na(POPULATION) == FALSE & POPULATION > 500000 &
ReportDate > as.Date("2020-03-14"))
summary(NewDf)

with(NewDf, tapply(NEW_CASES, COUNTY_NAME, median))

# 13.) Increase the bottom and left margins to be one-half of an inch larger
than their default values
#      recorded at the beginning of this assignment. Create a boxplot of the
number of New Cases
#      grouped by county using the data frame of large counties created in the
previous step. Supply
#      an appropriate Y axis label and a main title for the chart. Remove the
X axis label by using two
#      quotes with nothing inside them as the value for this label. The inside
of the boxes is maroon.
#      Supply an argument that will cause the whiskers of the plot to be 4
times the interquartile
#      range. Add the argument las=2 to cause the county names to be displayed
vertically.

# mai = c(1.02,0.82,0.82,0.42) = C(bottom, left, top, right)
par(mai = c(1.52, 1.32, 0.82, 0.42))

boxplot(NewDf$NEW_CASES ~ NewDf$COUNTY_NAME,
        xlab = "", ylab = "New Cases", range = 4, las = 2, col = "maroon")

dev.off()
# 14.)
# a.) The maximum number of new cases on Sep 12 was 1030
# b.) Not normally distributed because it seems to be skewed right.
# c.) There seems to be a strong positive relationship between the number of
#     total cases and the total deaths in a county.
# d.) Dec 13,2020
# e.) 142
# f.) Harris county, approximately 14000
```