```
# JRodoni_HW05_script.R
# C:/Users/jackr/OneDrive/Desktop/Graduate School Courses/
#   STAT 604 - STAT Computation/Homeworks/JRodoni_HW04_script.R
# Created By: Jack Rodoni
# Creation Date: 09/20/2021
# Purpose: STAT 604 Homework 5
# Last Executed: 09/21/2021
Sys.time()

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


# 2.) Import the COVID Activity.csv file into an R data frame using the
appropriate function. DO NOT
#     include code to display the data frame upon creation as it will likely
overload the console due to
#     the amount of data.

#     (a) Show the structure of the new data frame.
COVID_Activity <- read.csv("C:/Users/jackr/OneDrive/Desktop/Graduate School
Courses/STAT 604 - STAT Computation/Rdata/COVID Activity.csv")
str(COVID_Activity)

#     (b) Some of the columns have very long names that could be shortened
without any
#         negative consequences. However, the column order has not always been
consistent in
#         the download of this data so we need to make the changes using a
value replacement
#         You can use the names function to access the column names as a
vector that you can
#         manipulate as you would any other vector. (Remember you are not
actually changing
#         anything unless you use an assignment statement.) Change the columns
shown in the
#         table below:

names(COVID_Activity)[c(1,7,12,13)] = c("TOTAL_CASES", "NEW_DEATHS",
"NEW_CASES", "TOTAL_DEATHS")

#     (c) Display the first 10 rows and all columns of the modified data frame
COVID_Activity[1:10,]

# 3.)  Create a new data frame that is a subset of the data frame created from
the CSV file. The subset
#      will contain only rows for the state of Texas. Use a list of column
numbers in your subscript so
#      the new data frame contains only the following columns in the order
shown: COUNTY_NAME,
#      REPORT_DATE, NEW_CASES, TOTAL_CASES, NEW_DEATHS, TOTAL_DEATHS. Display
in the
```

```
#      console the structure of the new data frame.

Covid_Texas = subset(COVID_Activity[,c(2,4,12,1,7,13)],
COVID_Activity$PROVINCE_STATE_NAME == "Texas")

# 4.) Write an expression to import the txt file into a data frame. You may
spread the expression
#     across multiple lines in your script so it does not get cut off when you
convert the script to pdf if
#     you will insert your breaks between elements of the expression or
function.


PopTable <- read.table("C:/Users/jackr/OneDrive/Desktop/Graduate School
Courses/STAT 604 - STAT Computation/RData/Master Location Pop Table.txt",
                       header = TRUE, sep = ":", quote = "\"")

#     (a) Display the structure of the new data frame
str(PopTable)

#     (b) Change the name of the column that contains population data to
POPULATION to be more concise
names(PopTable)[9] = "POPULATION"

#     (c) Display the structure again showing the modifications
str(PopTable)

#     (d) Display the first 10 rows of the modified data frame
head(PopTable, n = 10)

# 5.) Create a new data frame by combining the “Texas” data frame with the
“population” data frame
#     that you created in the previous step. When the “population” data frame
is referenced in your
#     expression to combine the data frames, use expressions for the rows and
columns so that only
#     rows from Texas are selected and only the COUNTY_NAME and POPULATION
columns. Include
#     non-matches in the resulting data frame. The new data frame should have
153,255 rows

Merged_df = merge(Covid_Texas,
            subset(PopTable[,c("POPULATION", "COUNTY_NAME")],
PopTable$PROVINCE_STATE_NAME == "Texas"),
            all = TRUE)

#     (a) Display a summary of the new data frame
summary(Merged_df)

#     (b) Display the first 50 rows of the new data frame
head(Merged_df, n = 50)

# 6.) Execute a function that will make the columns of the data frame
available to R directly by
```

```
#     column name to simplify coding in the modifications described below:

attach(Merged_df)

#     (a) Use a function to convert REPORT_DATE to an actual R date value and
assign it to a new
#         column in the data frame. Display a summary of the new date column.
Note: You
#         cannot refer to this column only by name because it did not exist
when you executed
#         the function to make the columns available.

ReportDate = as.Date(REPORT_DATE)
Merged_df = cbind(Merged_df, ReportDate)
summary(Merged_df$ReportDate)

#     (b) The COVID activity statistics are contained in four columns whose
names were changed as
#         instructed earlier in the assignment.  Create four new columns in
the data frame that
#         represent each of the statistics as a percentage of the population
of that county. This is
#         done by dividing the original column by the POPULATION column.
Include PCT in the
#         names of your new columns to differentiate them from the originals.
Leave the
#         percentage values in their raw format of a value between 0 and 1.
You will notice that
#         some of the percentages are so small they are displayed in
exponential notation

Merged_df = cbind(Merged_df, PCT_Total_CASES = Merged_df$TOTAL_CASES/
Merged_df$POPULATION,
                             PCT_NEW_DEATHS  = Merged_df$NEW_DEATHS/
Merged_df$POPULATION,
                             PCT_NEW_CASES   = Merged_df$NEW_CASES/
Merged_df$POPULATION,
                             PCT_TOTAL_DEATHS= Merged_df$TOTAL_DEATHS/
Merged_df$POPULATION)

#     (c) Display the structure of the updated data frame and its first 20
rows.

str(Merged_df)
head(Merged_df, n = 20)

#     (d) Execute a function so that the column names of the data frame are no
longer available
#         in the R search path

detach(Merged_df)

# 7.) Create and display a new data frame that is a subset of the data frame
created in the previous
```

```
#     step. Use a logical test to subset the rows to only those where the
REPORT_DATE is the last
#     available and POPULATION is not missing. Determine the last date value
based on the summary
#     of the Date column from the previous step. Hard code this value into
your expression. Display
#     the structure of the new data frame.

Merged_df_Latest_NAsRemoved = subset(Merged_df, Merged_df$REPORT_DATE ==
"2021-09-12" & is.na(Merged_df$POPULATION) == FALSE)

# 8.) Use the colSums function to display the statewide totals of each of the
columns containing the
#     original Covid count statistics. Use the apply function to make the same
calculation. Include an
#     argument on your functions so that you will get a total even if there
are missing values for some
#     counties.


colSums(Merged_df_Latest_NAsRemoved[,c("TOTAL_CASES", "NEW_DEATHS",
"NEW_CASES", "TOTAL_DEATHS")])
apply(Merged_df_Latest_NAsRemoved[,c("TOTAL_CASES", "NEW_DEATHS", "NEW_CASES",
"TOTAL_DEATHS")], MARGIN = 2, FUN = sum)

# 9.) Using the last data frame created, display a list of County names,
TOTAL_CASES, POPULATION,
#     and percent of TOTAL_CASES, listed from the highest percentage to the
lowest.

Merged_df_Latest_NAsRemoved[order(Merged_df_Latest_NAsRemoved$PCT_Total_CASES,
decreasing = TRUE),
                            c("COUNTY_NAME","TOTAL_CASES",
"POPULATION","PCT_Total_CASES")]

# 10.) Display all data for counties whose names contain the letter V,
ignoring case.

Merged_df_Latest_NAsRemoved[grep("v",Merged_df_Latest_NAsRemoved$COUNTY_NAME,
ignore.case = TRUE),]

# 11.) Display the contents of the workspace

ls()

# 12.) Remove everything from the workspace except the data frame created
beginning in step 5
#      above and the data frame created in step 7. Display the contents of the
workspace again.

rm(list = setdiff(ls(), c("Merged_df", "Merged_df_Latest_NAsRemoved")))

# 13.)  Save the workspace in case we want to use it in the next assignment.
Name it HW05.RData.
```

```
#       You may save it initially using the R GUI but your script must contain
code to save the workspace
#       in case you submit the script again.


# 14.) After you have debugged your program and successfully executed it in a
new R session, use the
#      information in your console to answer the questions below in comment
lines at the bottom of
#      your script:

#     (a) How many observations were loaded from the CSV file?

#         2132949

#     (b) How many observations and variables are in the data frame loaded
from the txt file?

#         3483 observations of 10 variables

#     (c) What is one possible explanation for the minimum value of NEW_CASES
shown in the
#         summary from step 5a and what is your reaction to this value as an
analyst?

#         The minimum value could represent an adjustment to the previous
entry's number of new cases.
#         In other words, the new cases, minus adjustments made to the
previous entry is -1222.
#         As an analyst my first reaction would be to investigate this
further.

#     (d) Explain the difference in the summaries of the two date columns.
What are the
#         minimum and maximum dates in the data frame?

#         The original date column is a character vector, so the entries are
not interpreted by
#         r as dates, thus there are no numerical summaries available for the
original date column.
#         The minimum and maximum dates in the data frame are 01/21/2020 &
09/12/2021 respectively.

#     (e) What is the total number of COVID cases and deaths in the state of
Texas on the last
#         date reported?

#         Total Cases = 3815818, Total Deaths = 60357

#     (f) What is the name and population of the county with the lowest
percentage of cases as
#         of the last date reported?

#         County Name: King, Population: 272
```