COMP2501 Assignment1

## Requirements

**Submission deadline: Feb 28th, 2023 at 23:59.**

**Full mark of assignment 1: 33.**

For the following questions, please:

1. Replace all [Input here] places with your information or your answer (for multiple choice).
2. Complete the code block by adding your own code to fulfill the requirements in each question. Please use the existing code block and do not add your own code block.

Please make sure your Rmd file is a valid Markdown document and can be successfully knitted.

For assignment submission, please knit your final Rmd file into a Word document, and submit both your **Rmd** file and the knitted **Microsoft Word** document file to Moodle. You get 0 score if 1) the Rmd file you submitted cannot be knitted, and 2) you have not submitted a Word document. For each visualization question, please make sure that the generated plot is shown in-place with the question and after the code block.

## Name and UID

Name: Li Hanwen

UID: 3035719519

### Environmental setup

You need to have both the dslabs and dplyr packages installed. If not yet, please run install.packages(c("dslabs", "dplyr")) in your R environment. If you have installed the tidyverse package, dplyr is installed by default.

# Load the packages and dataset.  
library(dslabs)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

data("murders")

### Example question. Print the first 2 records of the murders dataset.

head(murders, 2)

## state abb region population total  
## 1 Alabama AL South 4779736 135  
## 2 Alaska AK West 710231 19

### 1. (1 points) Given matrix mat, print the entries of row 2, and columns 2 to 4.

mat <- matrix(1:16, nrow = 4)  
mat[2, 2:4]

## [1] 6 10 14

### 2. (2 points) Write a function compute\_s\_n that for any given n, computes the . Print the with .

compute\_s\_n <- function(n){  
 n \* sqrt(n+9) \* log10(n)  
}  
print(compute\_s\_n(500))

## [1] 30445.77

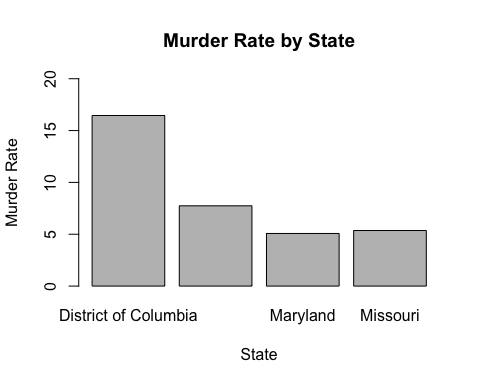
### 3. (2 points) Compute the murder rate per 100,000 people for each state and store it in an object called murder\_rate. Then use logical operators to find which state has a murder rate per 100,000 people higher than 5. Find these states, print their names and murder rate per 100,000 people.

murder\_rate <- murders |> group\_by(state) |> summarize(murder.rate = sum(total) / sum(population) \* 100000)  
murder\_rate\_filtered = murder\_rate |> filter(murder.rate > 5)  
murder\_rate\_filtered

## # A tibble: 4 × 2  
## state murder.rate  
## <chr> <dbl>  
## 1 District of Columbia 16.5   
## 2 Louisiana 7.74  
## 3 Maryland 5.07  
## 4 Missouri 5.36

### 4. (2 points) For all states having a murder rate per 100,000 people higher than 5, use the barplot function to create a barplot with the x-axis being the state name, and the y-axis being the murder rate per 100,000 people of each state. (Hint: check some barplot examples at <https://r-graph-gallery.com/210-custom-barplot-layout.html>)

barplot(  
 height = murder\_rate\_filtered$murder.rate,   
 names.arg = murder\_rate\_filtered$state,  
 ylim = c(0,20),  
 xlab = "State",  
 ylab = "Murder Rate",  
 main = "Murder Rate by State"  
)



### 5. (1 points) Examine the built-in dataset Orange. Which of the following is true?

* 1. Orange is tidy data: it has one observation for each row.
  2. Orange is not tidy: we need at least one column with a character vector.
  3. Orange is not tidy: it is a matrix instead of a data frame.
  4. Orange is tidy data: all small datasets are tidy by definition.

Your answer is: [a]

### 6. (3 points) Base on the murders dataset, create a table called my\_states that contains rows for states satisfying two conditions: 1) it is in either West or South, and 2) the murder rate per 100,000 people is less than 2.0. Use select to show only the state name, the region and the murder rate, and use top\_n function to find the 3 safest states among them.

my\_states = murders |> mutate(murder.rate = total \* 100000 / population)  
my\_states = my\_states |> filter(region %in% c("West", "East"), murder.rate < 2.0) |> select(state, region, murder.rate)  
top\_n(my\_states, -3, murder.rate)

## state region murder.rate  
## 1 Hawaii West 0.5145920  
## 2 Idaho West 0.7655102  
## 3 Utah West 0.7959810

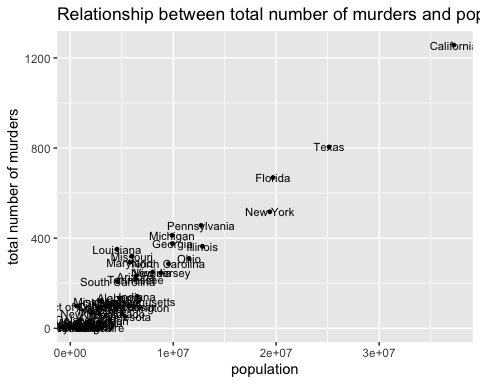
### 7. (2 points) By using the murders dataset, compute the average murder rate per 100,000 people in the four regions respectively of the U.S., and sort the results by murder rate in ascending order.

murders\_group = murders |> group\_by(region) |> summarize(avg\_murder\_rate = sum(total) / sum(population) \* 100000) |> arrange(avg\_murder\_rate)  
murders\_group

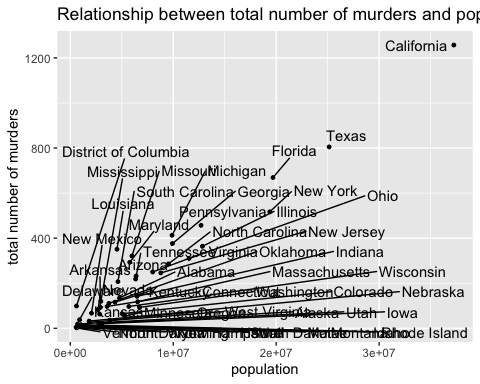
## # A tibble: 4 × 2  
## region avg\_murder\_rate  
## <fct> <dbl>  
## 1 Northeast 2.66  
## 2 West 2.66  
## 3 North Central 2.73  
## 4 South 3.63

### 8. (3 points) Use the ggplot2 package to create a scatterplot from the murders dataset, where the x-axis is the number of population, the y-axis is the total number of murders, and each point in the scatterplot is labeled with the state name. Please add an appropriate title, and axis labels to the plot.

library(ggplot2)  
ggplot(data = murders, aes(x = population, y = total)) +  
 geom\_point(size = 1) +  
 ggtitle("Relationship between total number of murders and population") +  
 geom\_text(aes(label = state), cex = 3) +  
 labs(x = "population", y = "total number of murders")



library(ggrepel)  
options(ggrepel.max.overlaps = Inf)  
ggplot(data = murders, aes(x = population, y = total, label = state)) +  
 geom\_point(size = 1) +  
 geom\_text\_repel() +  
 ggtitle("Relationship between total number of murders and population") +  
 labs(x = "population", y = "total number of murders")



### 9. (17 points) Explore the tidyverse with the COVID-19 dataset (<http://www.bio8.cs.hku.hk/comp2501/covid.csv>), and answer the following questions.

# Load the required packages  
library(dplyr)  
library(ggplot2)

#### a. (2 points) Read the CSV formatted dataset. Find out how many observations (rows) and variables (columns) are in the dataset. Print the names of all variables.

df = read.csv("http://www.bio8.cs.hku.hk/comp2501/covid.csv")  
dim(df)

## [1] 47480 12

names(df)

## [1] "dateRep"   
## [2] "day"   
## [3] "month"   
## [4] "year"   
## [5] "cases"   
## [6] "deaths"   
## [7] "countriesAndTerritories"   
## [8] "geoId"   
## [9] "countryterritoryCode"   
## [10] "popData2019"   
## [11] "continentExp"   
## [12] "Cumulative\_number\_for\_14\_days\_of\_COVID.19\_cases\_per\_100000"

#### b. (1 points) List the observation with the largest Cumulative\_number\_for\_14\_days\_of\_COVID.19\_cases\_per\_100000.

top\_n(df, 1, Cumulative\_number\_for\_14\_days\_of\_COVID.19\_cases\_per\_100000)

## dateRep day month year cases deaths countriesAndTerritories geoId  
## 1 20/08/2020 20 8 2020 175 1 Aruba AW  
## countryterritoryCode popData2019 continentExp  
## 1 ABW 106310 America  
## Cumulative\_number\_for\_14\_days\_of\_COVID.19\_cases\_per\_100000  
## 1 1058.226

# df |> arrange(desc(Cumulative\_number\_for\_14\_days\_of\_COVID.19\_cases\_per\_100000)) |> head(1)

#### c. (2 points) How many unique countriesAndTerritories are in the dataset? How many unique continentExp are in the dataset?

length(unique(df$countriesAndTerritories))

## [1] 210

length(unique(df$continentExp))

## [1] 6

#### d. (3 points) For 1) the whole dataset, 2) different countriesAndTerritories, and 3) different continentExp, compute both i) the sum of cases, and ii) the sum of deaths. Sort the results by the sum of cases descendingly. Use head() if there are too many rows in the results.

df |> summarize(case\_sum = sum(cases), death\_sum = sum(deaths)) |> head()

## case\_sum death\_sum  
## 1 35848254 1048181

df |> group\_by(countriesAndTerritories) |> summarize(case\_sum = sum(cases), death\_sum = sum(deaths)) |> arrange(desc(case\_sum)) |> head()

## # A tibble: 6 × 3  
## countriesAndTerritories case\_sum death\_sum  
## <chr> <int> <int>  
## 1 United\_States\_of\_America 7501612 210909  
## 2 India 6757131 104555  
## 3 Brazil 4969141 147494  
## 4 Russia 1237504 21663  
## 5 Colombia 869808 27017  
## 6 Peru 832929 32914

df |> group\_by(continentExp) |> summarize(case\_sum = sum(cases), death\_sum = sum(deaths)) |> arrange(desc(case\_sum)) |> head()

## # A tibble: 6 × 3  
## continentExp case\_sum death\_sum  
## <chr> <int> <int>  
## 1 America 17445678 578079  
## 2 Asia 11233759 203583  
## 3 Europe 5605508 228689  
## 4 Africa 1528213 36828  
## 5 Oceania 34400 995  
## 6 Other 696 7

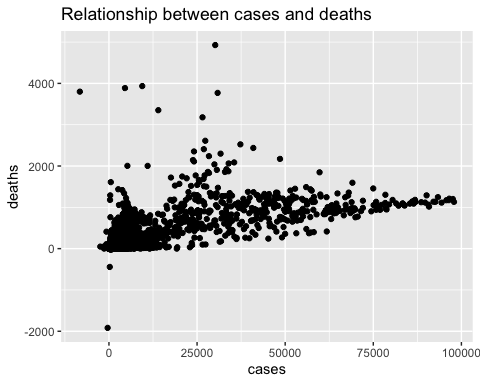
#### e. (2 points) Add a new column date with the standard date format “YYYY-MM-DD” to the data table according to the dateRep column. Be reminded the format of dateRep is “DD/MM/YYYY”. Please use head() to show the result.

df = df |> mutate(date = as.Date(as.character(dateRep), "%d/%m/%Y"))  
df |> head()

## dateRep day month year cases deaths countriesAndTerritories geoId  
## 1 07/10/2020 7 10 2020 62 2 Afghanistan AF  
## 2 06/10/2020 6 10 2020 145 5 Afghanistan AF  
## 3 05/10/2020 5 10 2020 44 0 Afghanistan AF  
## 4 04/10/2020 4 10 2020 7 4 Afghanistan AF  
## 5 03/10/2020 3 10 2020 5 0 Afghanistan AF  
## 6 02/10/2020 2 10 2020 17 0 Afghanistan AF  
## countryterritoryCode popData2019 continentExp  
## 1 AFG 38041757 Asia  
## 2 AFG 38041757 Asia  
## 3 AFG 38041757 Asia  
## 4 AFG 38041757 Asia  
## 5 AFG 38041757 Asia  
## 6 AFG 38041757 Asia  
## Cumulative\_number\_for\_14\_days\_of\_COVID.19\_cases\_per\_100000 date  
## 1 1.0593622 2020-10-07  
## 2 1.0830204 2020-10-06  
## 3 0.7807210 2020-10-05  
## 4 0.6650587 2020-10-04  
## 5 0.9752441 2020-10-03  
## 6 1.0856491 2020-10-02

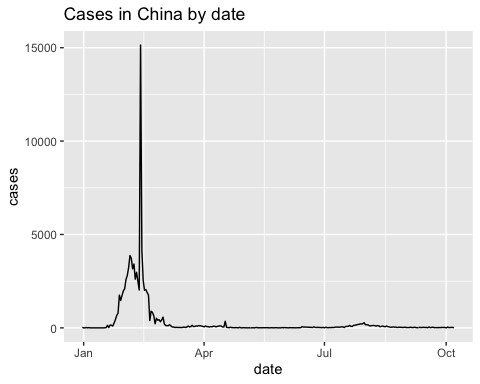
#### f. (1 points) Create a scatterplot showing cases vs. deaths. Set an appropriate plot title and axis titles.

ggplot(data = df, aes(x = cases, y = deaths)) +  
 geom\_point() +  
 ggtitle("Relationship between cases and deaths") +  
 labs(x = "cases", y = "deaths")



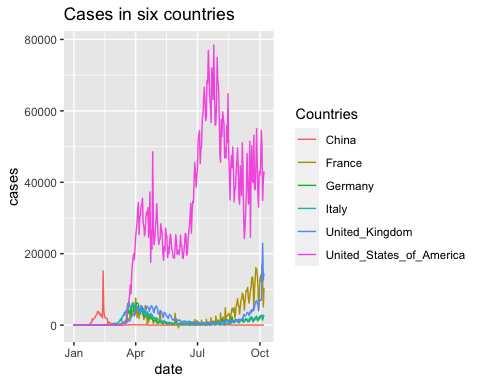
#### g. (2 points) Create a line plot using data with countriesAndTerritories=="China", showing date on the x-axis and cases per day on the y-axis. Set an appropriate plot title and axis titles.

df\_china = df |> filter(countriesAndTerritories=="China")  
ggplot(data = df\_china, aes(x = date, y = cases)) +  
 geom\_line() +  
 ggtitle("Cases in China by date") +  
 labs(x = "date", y = "cases")



#### h. (2 points) Similar to above, create a line plot using the data of six countries including “China”, “United\_States\_of\_America”, “United\_Kingdom”, “France”, “Germany”, and “Italy”. Use different line colors for each country. Set an appropriate plot title and axis titles.

df\_sub = df |> filter(countriesAndTerritories %in% c("China", "United\_States\_of\_America", "United\_Kingdom", "France", "Germany", "Italy"))  
ggplot(data = df\_sub, aes(x = date, y = cases)) +  
 geom\_line(aes(color = countriesAndTerritories)) +  
 ggtitle("Cases in six countries") +  
 labs(x = "date", y = "cases", color = "Countries")



#### i. (2 points) Similar to question h, create a boxplot instead.

df\_sub = df |> filter(countriesAndTerritories %in% c("China", "United\_States\_of\_America", "United\_Kingdom", "France", "Germany", "Italy"))  
ggplot(data = df\_sub, aes(x = countriesAndTerritories, y = cases)) +  
 geom\_boxplot(aes(color = countriesAndTerritories)) +  
 ggtitle("Cases in six countries") +  
 labs(x = "country", y = "cases", color = "Countries") +  
 theme(axis.text.x = element\_text(angle = 30, vjust = .8))

