EHSC P8321: Lab & Homework #1

Lizbeth Gomez (UNI: LG2982)

Due: April 6, 2020 by 3:59 pm

## PART I. Setting up RStudio & Reading Datasets in

### The first task we want to do after creating up our R Markdown document is to install and load any necessary packages.

### Create new dataframes in RStudio with the COVID-19 time series datasets.

confirmed = read\_csv("./data/time\_series\_covid19\_confirmed\_global.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## `Province/State` = col\_character(),  
## `Country/Region` = col\_character()  
## )

## See spec(...) for full column specifications.

deaths = read\_csv("./data/time\_series\_covid19\_deaths\_global.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## `Province/State` = col\_character(),  
## `Country/Region` = col\_character()  
## )  
## See spec(...) for full column specifications.

recovered = read\_csv("./data/time\_series\_covid19\_recovered\_global.csv")

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## `Province/State` = col\_character(),  
## `Country/Region` = col\_character()  
## )  
## See spec(...) for full column specifications.

## PART II. Descriptive Statistics

### What if we wanted to know the total number of confirmed cases, deaths, and those recovered from COVID-19 of a country? Let’s generate some summary statistics for Iran.

# Let's do some data tidying. First we want to convert the dataset from wide-to-long format using the pivot\_longer function. We can use the janitor::clean\_names() function to change the variable names into RStudio-friendly names. Next, we want to select only the variables of interest. Last, we can filter the dataset for cases in Iran only.  
iran\_confirmed = confirmed %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "cases") %>%  
 janitor::clean\_names() %>%   
 select(country\_region, lat, long, date, cases) %>%   
 filter (country\_region == "Iran")   
iran\_totalcases = sum(iran\_confirmed$cases)  
  
# We can follow the same tidying method for the deaths and recovered datasets.  
iran\_deaths = deaths %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "deaths") %>%  
 janitor::clean\_names() %>%   
 select(country\_region, lat, long, date, deaths) %>%   
 filter (country\_region == "Iran")  
iran\_totaldeaths = sum(iran\_deaths$deaths)  
  
iran\_recovered = recovered %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "recovered") %>%  
 janitor::clean\_names() %>%   
 select(country\_region, lat, long, date, recovered) %>%   
 filter (country\_region == "Iran")  
iran\_totalrecovered = sum(iran\_recovered$recovered)

To date, Iran has 2.343710^{4} total confirmed cases, 836 total deaths, and 4924 total recovered individuals.

### Your turn: Generate the total number of confirmed cases, deaths, and those recovered from COVID-19 in Ireland. Using inline R, include a summary sentence about Ireland’s statistics.

ireland\_confirmed = confirmed %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "cases") %>%  
 janitor::clean\_names() %>%   
 select(country\_region, lat, long, date, cases) %>%   
 filter (country\_region == "Ireland")   
ireland\_totalcases = sum(ireland\_confirmed$cases)  
  
# We can follow the same tidying method for the deaths and recovered datasets.  
ireland\_deaths = deaths %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "deaths") %>%  
 janitor::clean\_names() %>%   
 select(country\_region, lat, long, date, deaths) %>%   
 filter (country\_region == "Ireland")  
ireland\_totaldeaths = sum(ireland\_deaths$deaths)  
  
ireland\_recovered = recovered %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "recovered") %>%  
 janitor::clean\_names() %>%   
 select(country\_region, lat, long, date, recovered) %>%   
 filter (country\_region == "Ireland")  
ireland\_totalrecovered = sum(ireland\_recovered$recovered)

To date, Ireland has 53 total confirmed cases, 0 total deaths, and 0 total recovered individuals.

## PART III. Creating Visualizations

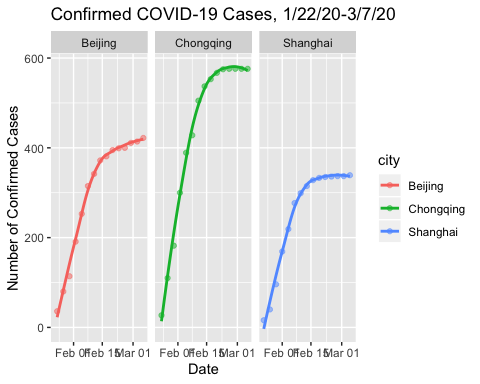
### Let’s create a scatterplot with confirmed case counts to date of the three largest Chinese cities by population (Shanghai, Beijing, and Chongqing).

china\_confirmed = confirmed %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "cases") %>%  
 janitor::clean\_names() %>%   
 select(province\_state, country\_region, lat, long, date, cases) %>%   
 rename ("city" = province\_state) %>%   
 filter (country\_region == "China", city == c("Shanghai", "Beijing", "Chongqing")) %>%   
 mutate(date = as.Date(date, format = "%m/%d/%y"))

## Warning in city == c("Shanghai", "Beijing", "Chongqing"): longer object length  
## is not a multiple of shorter object length

china\_confirmed %>%   
 ggplot(aes(x = date, y = cases, color = city)) +   
 geom\_point(alpha = 0.5) +  
 geom\_smooth(se = FALSE) +  
 facet\_grid(. ~ city) +  
 labs(title = "Confirmed COVID-19 Cases, 1/22/20-3/7/20",  
 x = "Date",  
 y = "Number of Confirmed Cases")

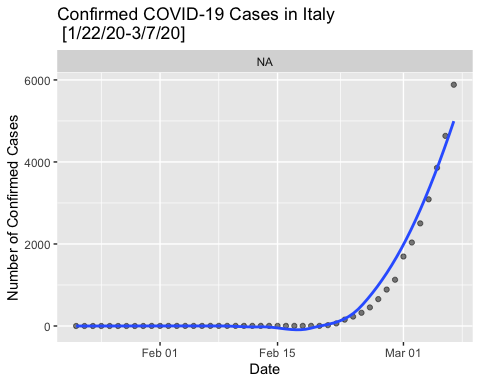
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



### Your turn: create a scatterplot of Italy’s confirmed case counts to date.

italy\_confirmed = confirmed %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "cases") %>%  
 janitor::clean\_names() %>%   
 select(province\_state, country\_region, lat, long, date, cases) %>%   
 rename ("city" = province\_state) %>%   
 filter (country\_region == "Italy") %>%   
 mutate(date = as.Date(date, format = "%m/%d/%y"))  
  
italy\_confirmed %>%   
 ggplot(aes(x = date, y = cases)) +   
 geom\_point(alpha = 0.5) +  
 geom\_smooth(se = FALSE) +  
 facet\_grid(. ~ city) +  
 labs(title = "Confirmed COVID-19 Cases in Italy \n [1/22/20-3/7/20]",  
 x = "Date",  
 y = "Number of Confirmed Cases")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

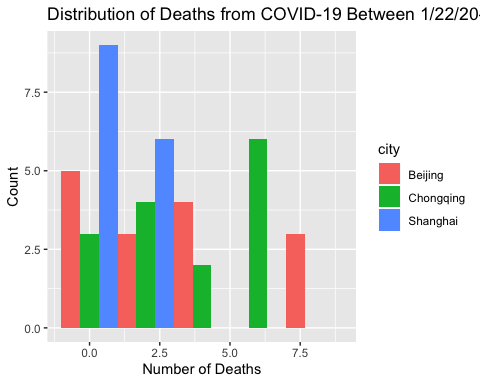


## What if we wanted to know if the distribution of the number of deaths from COVID-19 differed across the three main cities in China? Let’s create a histogram of China’s deaths from COVID-19 to date:

china\_deaths = deaths %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "deaths") %>%  
 janitor::clean\_names() %>%   
 select(province\_state, country\_region, lat, long, date, deaths) %>%   
 rename ("city" = province\_state) %>%   
 filter (country\_region == "China", city == c("Shanghai", "Beijing", "Chongqing")) %>%   
 mutate(date = as.Date(date, format = "%m/%d/%y"))

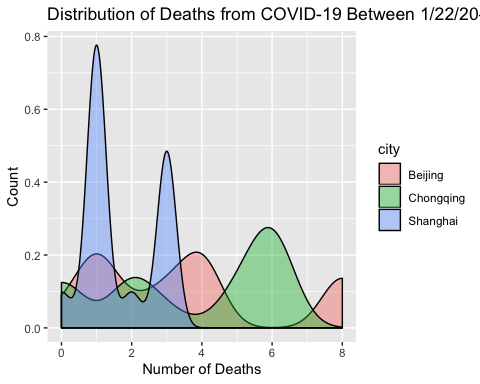
## Warning in city == c("Shanghai", "Beijing", "Chongqing"): longer object length  
## is not a multiple of shorter object length

china\_deaths %>%   
 ggplot(aes(x = deaths, fill = city)) +   
 geom\_histogram(position = "dodge", binwidth = 2) +  
 labs(title = "Distribution of Deaths from COVID-19 Between 1/22/20-3/7/20",  
 x = "Number of Deaths",  
 y = "Count")



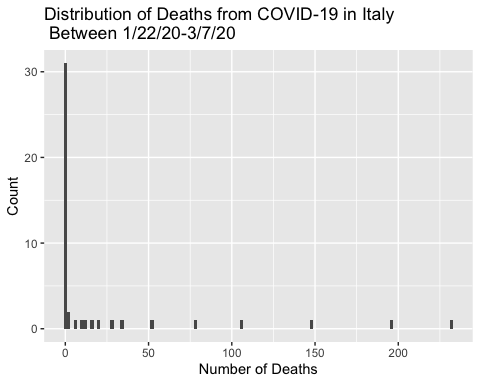
### While a histogram is a great way of showing the distribution of data, having the bars for each group side-by-side gets sort of hard to understand. We can often use density plots in place of histograms when comparing distributions across groups:

china\_deaths %>%   
 ggplot(aes(x = deaths, fill = city)) +   
 geom\_density(alpha = 0.4, adjust = 0.5, color = "black") +  
 labs(title = "Distribution of Deaths from COVID-19 Between 1/22/20-3/7/20",  
 x = "Number of Deaths",  
 y = "Count")

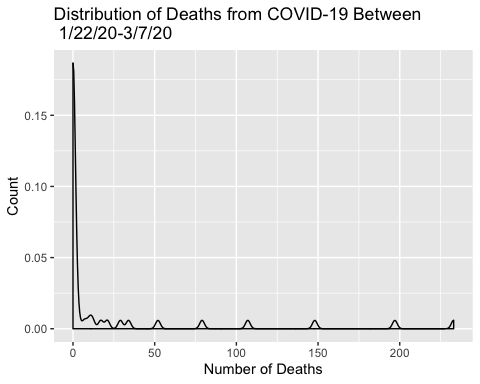


### Your turn: Create both a histogram and a density plot of Italy’s deaths from COVID-19 to date.

italy\_deaths = deaths %>%  
 pivot\_longer(  
 "1/22/20":"3/7/20",  
 names\_to = "date",  
 values\_to = "deaths") %>%  
 janitor::clean\_names() %>%   
 select(province\_state, country\_region, lat, long, date, deaths) %>%   
 filter (country\_region == "Italy") %>%   
 mutate(date = as.Date(date, format = "%m/%d/%y"))  
  
italy\_deaths %>%   
 ggplot(aes(x = deaths)) +   
 geom\_histogram(position = "dodge", binwidth = 2) +  
 labs(title = "Distribution of Deaths from COVID-19 in Italy \n Between 1/22/20-3/7/20",  
 x = "Number of Deaths",  
 y = "Count")



#density plot  
italy\_deaths %>%   
 ggplot(aes(x = deaths)) +   
 geom\_density(alpha = 0.4, adjust = 0.5, color = "black") +  
 labs(title = "Distribution of Deaths from COVID-19 Between \n 1/22/20-3/7/20",  
 x = "Number of Deaths",  
 y = "Count")



## Homework 1, Part 1: Finish generating the plots from the labs if you have not done so already. Knit this R Markdown document to a Word document and submit on Courseworks.