COVID19_Data_Report

Israel Johnson

11/14/2021

Objective

This incident data report will provide specific analysis and results, based upon the following dataset: 'CSSE COVID19 Time Series Data', provided by Johns Hopkins University. To get started, we will need to install the following R packages for further use:

1.) tidyverse 2.) sessioninfo

Once installed, the following code will load key libraries from the required packages that will be used for this analysis.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyr)
library(dplyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(ggplot2)
```

Importing Data (Reproducibility)

We will need to import the required CSV datasets, which can be found at the following source: https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series. Once you are there, proceed with the following steps:

1.) The Github repository from the above source will contain the four covid19 datasets that are needed for further analysis. To obtain the data, first assign the following source to a variable (this will serve as the base link for all four datasets): https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/.

2.) Append the name of each data file to the base link and execute an operation to read in the data.

The following code completes this process and reads in the data in a CSV format.

```
covid_data_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/

files <- c("time_series_covid19_confirmed_US.csv",
           "time_series_covid19_confirmed_global.csv",
           "time_series_covid19_deaths_US.csv",
           "time_series_covid19_deaths_global.csv")

covid19_data <- str_c(covid_data_url, files)
US_cases <- read_csv(covid19_data[1])
```

```
## Rows: 3342 Columns: 682
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (676): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20,...
```

```
##
## 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.
```

```
global_cases <- read_csv(covid19_data[2])
```

```
## Rows: 280 Columns: 675
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (673): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20, ...
```

```
##
## 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.
```

```
US_deaths <- read_csv(covid19_data[3])
```

```
## Rows: 3342 Columns: 683
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (677): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24/...

##
## 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.
```

```
global_deaths <- read_csv(covid19_data[4])
```

```
## Rows: 280 Columns: 675

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (673): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20, ...

##
## 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.
```

Tidying and Transforming the Data

After reading in the required COVID19 Data, tidy and transform the data into a desired format for the analysis (leading to four main datasets: US_cases, US_deaths, global_cases, and global_deaths). For the global data, I transformed the corresponding datasets to focus on COVID19 cases within Germany and Italy.

In regards to the analysis, the main focus will be on answering the following questions:

1.) "How did the number of COVID19 cases in Germany and Italy change over time?" 2.) "Between Germany and Italy, which country has the higher number of COVID19-related deaths?"

Further visualizations and a data model will cover answers to the above questions.

```
US_cases <- US_cases %>%
  pivot_longer(cols = -(UID:Combined_Key),
               names_to = "date",
               values_to = "cases") %>%
  select(Admin2:cases) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat,Long_))

US_deaths <- US_deaths %>%
  pivot_longer(cols = -(UID:Combined_Key),
               names_to = "date",
               values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat,Long_))
```

```
## Warning: 3342 failed to parse.
```

```
global_cases <- global_cases %>%
  pivot_longer(cols = -c('Province/State',
                         'Country/Region', Lat, Long),
               names_to = "date",
               values_to = "cases") %>%
  select(-c(Lat,Long))

global_deaths <- global_deaths %>%
  pivot_longer(cols = -c('Province/State',
                         'Country/Region', Lat, Long),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat,Long))

global_cov19_data <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = `Country/Region`,
         Province_State = `Province/State`) %>%
  mutate(date = mdy(date))
```

```
## Joining, by = c("Province/State", "Country/Region", "date")
```

```
global_country_region <- global_cov19_data$Country_Region
global_dates <- global_cov19_data$date
specified_global_cases <- global_cov19_data$cases
specified_global_deaths <- global_cov19_data$deaths

global_vis1_data <- data.frame(global_country_region, global_dates, specified_global_cases, specified_gl
```

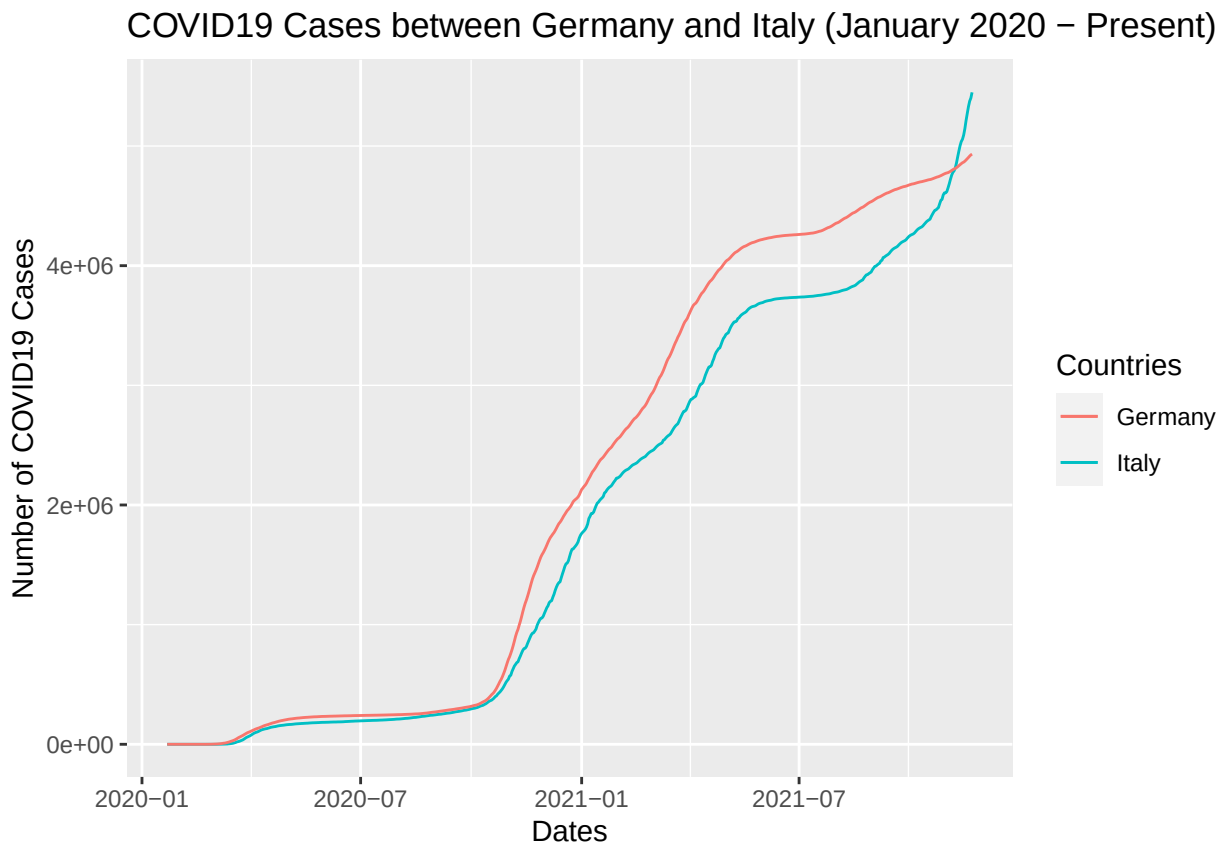
Data Visualization 01 : Number of COVID19 Cases Between Germany and Italy (January 2020 - Present)

The following visualization presents a line graph (each line representing a different country) and displays how the number of COVID19 cases for each country have changed over time. As a result, the visualization reveals that Germany had the higher number of COVID19 cases over time during the given time period.

```
italy_covid_data <- global_vis1_data %>%
  filter(global_country_region=='Italy')

germany_covid_data <- global_vis1_data %>%
  filter(global_country_region=='Germany')

ggplot()+
  geom_line(germany_covid_data, mapping=aes(global_dates, specified_global_cases,
                                            color="red"))+
  geom_line(italy_covid_data, mapping=aes(global_dates, specified_global_cases,
                                          color="blue"))+
  scale_color_discrete(name = "Countries", labels = c("Germany", "Italy"))+
  labs(title='COVID19 Cases between Germany and Italy (January 2020 - Present)',
       x='Dates',
       y='Number of COVID19 Cases')
```



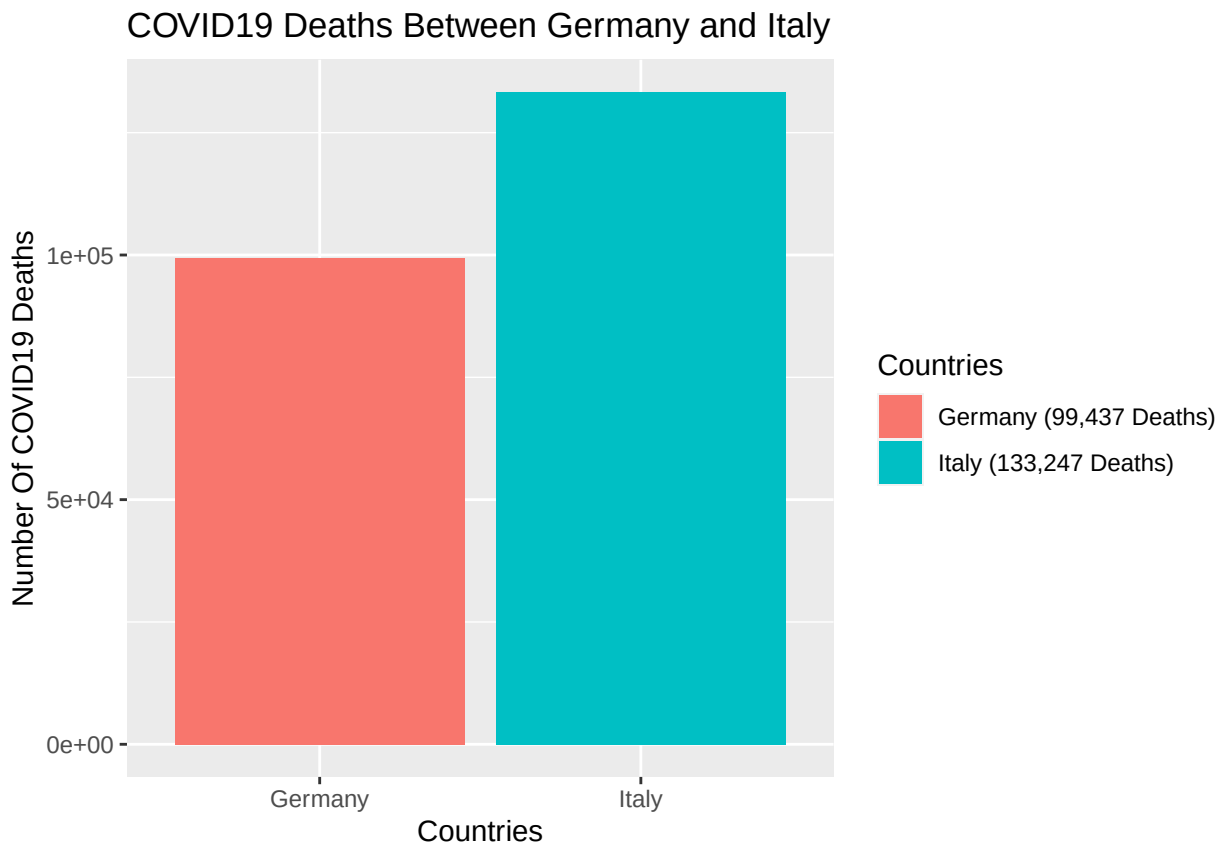
Data Visualization 02: COVID19 Deaths Between Germany and Italy (January 2020 - Present)

The second visualization presents a bar graph that displays the number of COVID19-related deaths (as of today) for Germany and Italy. As a result, Italy appears to have more COVID19-related deaths than Germany, even though Germany had a higher number of COVID19 cases over time. What additional data would be needed in order to determine why Italy has a higher number of COVID19-related deaths?

Additional types of data that could provide more insight on the high number of Italy COVID19 deaths could include the following: government policy information, hospital availability, COVID19 vaccine availability, number of COVID19 deaths among different age groups, and many other forms of data.

```
selected_global_covid_data <- global_vis1_data %>%
  filter(global_country_region==c("Germany", "Italy"))

ggplot(selected_global_covid_data, mapping=aes(global_country_region, specified_global_deaths,fill=globa
  geom_bar(stat = 'identity', position = 'dodge')+
  scale_fill_discrete(name = "Countries", labels = c("Germany (99,437 Deaths)", "Italy (133,247 Deaths)
  labs(title="COVID19 Deaths Between Germany and Italy", x="Countries", y="Number Of COVID19 Deaths")
```



Linear Regression Data Model

Using information and results from the visualizations, I generated a data model that reflects the COVID19 information related to Germany and Italy. This is a linear regression model where COVID19 cases for Germany and Italy will serve as a function of the series of dates. The independent variable will be the dates to indicate the change in time and the dependent variable will be the number of cases that have occurred between Germany and Italy. Once both variables are obtained, use them as well as the global COVID19 data to generate the model.

As part of the result and according to the the plot of the model, the predicted occurrences initially do not match with the actual occurrences for the year 2020. However, after July of 2021, the predicted values appear to be closer in accuracy with the actual case values of COVID19 cases of both Germany and Italy. Despite a high value for the residual standard error (which indicates a lack of model accuracy), the model appears to improve in accuracy over time in the case of 2021 for COVID19 cases of Germany of Italy. Additional input data and more adjustments to the model parameters can potentially improve upon the accuracy of the model.

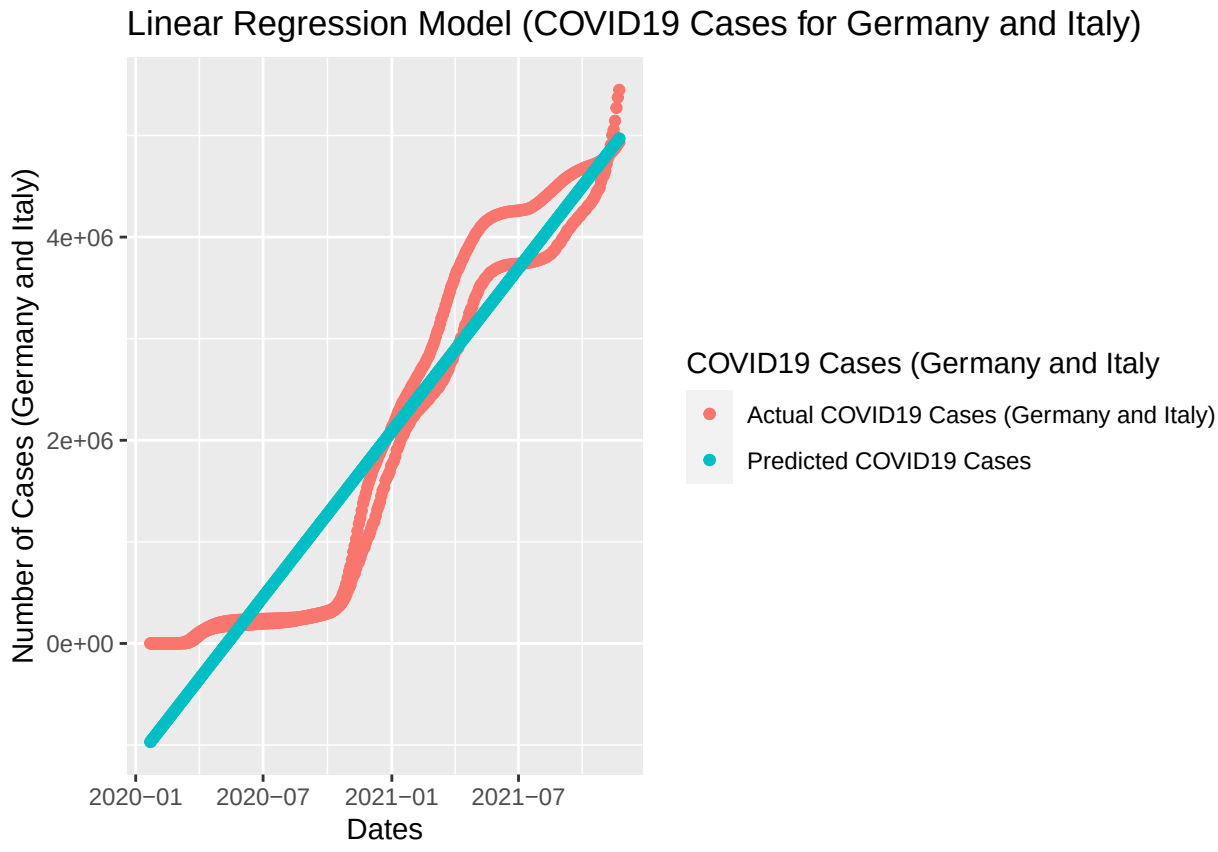
```
mod <- lm(selected_global_covid_data$specified_global_cases~selected_global_covid_data$global_dates, se
df_w_pred <- mutate(selected_global_covid_data, pred = predict(mod))

summary(mod)
```

```
##
## Call:
## lm(formula = selected_global_covid_data$specified_global_cases ~
```

```
##     selected_global_covid_data$global_dates, data = selected_global_covid_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1056200  -273551     8178   361069   968974
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             -1.630e+08  1.907e+06  -85.49   <2e-16
## selected_global_covid_data$global_dates  8.863e+03  1.024e+02   86.54   <2e-16
##
## (Intercept)                             ***
## selected_global_covid_data$global_dates ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 515000 on 670 degrees of freedom
## Multiple R-squared:  0.9179, Adjusted R-squared:  0.9178
## F-statistic:  7490 on 1 and 670 DF,  p-value: < 2.2e-16
```

```
ggplot(df_w_pred)+
  geom_point(aes(x=global_dates, y=specified_global_cases, color="Actual COVID19 Cases (Germany and Ital
  geom_point(aes(x=global_dates, y=pred, color="Predicted COVID19 Cases"))+
  guides(color=guide_legend(title = "COVID19 Cases (Germany and Italy"))+
  labs(title="Linear Regression Model (COVID19 Cases for Germany and Italy)", y="Number of Cases (Germa
```



Identification of Bias and Conclusion (Communication of Results and Summary)

There is potential for different biases to occur here in the given data and analysis, but the most likely case for this process would be selection bias. The given data model and related plot are based upon COVID19 data for Germany and Italy for the given timeframe. The analysis could have shown results for alternate selections of COVID19 data (for other countries such as France, Spain, or others).

Another form of bias for the source data itself could be in the form of confirmation bias. It is possible that a misdiagnosis occurred in the case of a COVID19 diagnosis. For example, what was recorded as a COVID19 diagnosis at the time may have actually been a different illness, like the flu (especially if both illnesses share similar symptoms). There also could have been delays for a COVID19 diagnosis. If a COVID19 case occurred in May and the diagnosis was not recorded until June, this would result in an inaccurate entry of the COVID19 case for the dataset.

Overall, the analysis has been able to reveal key aspects of information, based upon the COVID19 dataset. The analysis (visualizations included) have been able to answer the following questions:

1.) How have COVID19 cases for Germany and Italy changed overtime between 2020 and 2021? 2.) Between Germany and Italy, which country has the highest recorded number of COVID19 deaths?

Additional data is required in order to find out why Italy has a high number of COVID19 deaths and what preventative measures can be taken to reduce the number of COVID19 deaths.

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7
##  [5] purrr_0.3.4     readr_2.0.2     tidyr_1.1.4     tibble_3.1.5
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7       assertthat_0.2.1 digest_0.6.28    utf8_1.2.2
##  [5] R6_2.5.1         cellranger_1.1.0 backports_1.3.0  reprex_2.0.1
##  [9] evaluate_0.14    highr_0.9        httr_1.4.2       pillar_1.6.4
## [13] rlang_0.4.12     curl_4.3.2       readxl_1.3.1     rstudioapi_0.13
## [17] rmarkdown_2.11   labeling_0.4.2   bit_4.0.4        munsell_0.5.0
## [21] broom_0.7.10     compiler_4.1.2   modelr_0.1.8     xfun_0.27
## [25] pkgconfig_2.0.3  htmltools_0.5.2  tidyselect_1.1.1 fansi_0.5.0
## [29] crayon_1.4.2     tzdb_0.2.0       dbplyr_2.1.1     withr_2.4.2
## [33] grid_4.1.2       jsonlite_1.7.2   gtable_0.3.0     lifecycle_1.0.1
## [37] DBI_1.1.1        magrittr_2.0.1   scales_1.1.1     cli_3.1.0
## [41] stringi_1.7.5    vroom_1.5.5      farver_2.1.0     fs_1.5.0
```

```
## [45] xml2_1.3.2       ellipsis_0.3.2   generics_0.1.1   vctrs_0.3.8
## [49] tools_4.1.2      bit64_4.0.5      glue_1.4.2       hms_1.1.1
## [53] parallel_4.1.2   fastmap_1.1.0    yaml_2.2.1       colorspace_2.0-2
## [57] rvest_1.0.2      knitr_1.36       haven_2.4.3
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.