Project 3 COVID

Mia Rosini

Part I: Investigating the SIR Model

In this section, I experiment with different Beta and Gamma values to demonstrate the effect of increasing and decreasing Beta and Gamma.

```r
#Beta = 1.5, Gamma = 0.1
S <- 99000
I <- 1000
R <- 00
N <- S + I + R
pop <- c(0, S, I, R)
beta <- 1.5
gamma <- 0.1

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered

  # Add another row to pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
}
colnames(pop) <- c("day", "S", "I", "R")
pop1 <- pop %>% data.frame()
```

```r
#Increasing Beta & Gamma: Beta = 5, Gamma = 3
S <- 99000
I <- 1000
R <- 00
N <- S + I + R
pop <- c(0, S, I, R)
beta <- 5
gamma <- 3

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered
```

```r
  # Add another row tp pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
}
colnames(pop) <- c("day", "S", "I", "R")
pop2 <- pop %>% data.frame()
```

```r
#Decreasing Beta & Gamma: Beta = 0.3, Gamma = 0.05
S <- 99000
I <- 1000
R <- 00
N <- S + I + R
pop <- c(0, S, I, R)
beta <- 0.3
gamma <- 0.05

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered

  # Add another row tp pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
}
colnames(pop) <- c("day", "S", "I", "R")
pop3 <- pop %>% data.frame()
```
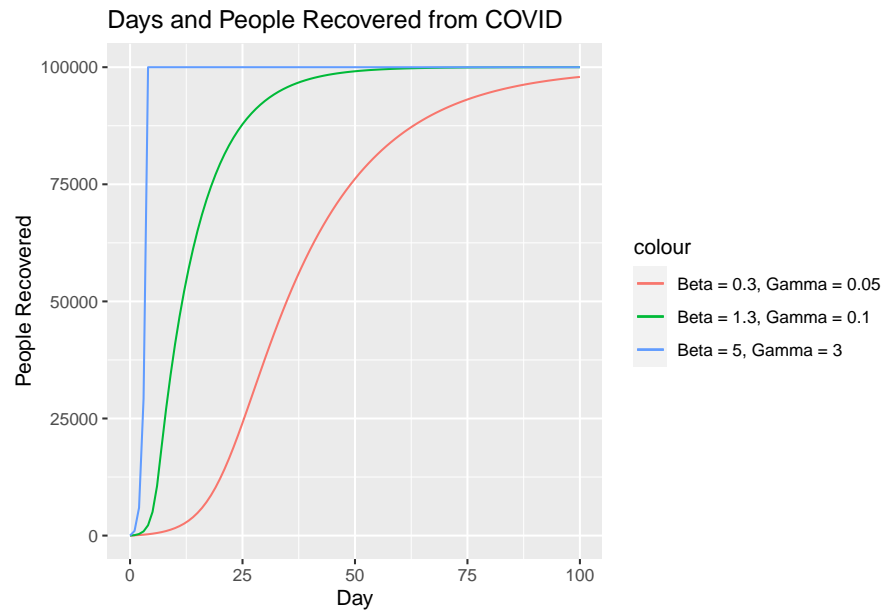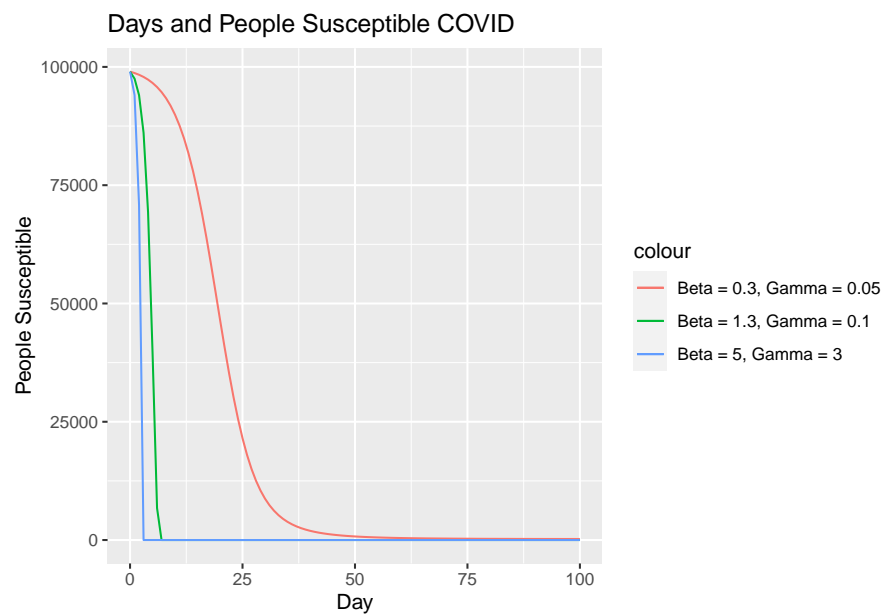
```r
#Combining all three scenarios into one plot
ggplot() +
  geom_line(data = pop3, aes(x = day, y = R, color = "Beta = 0.3, Gamma = 0.05")) +
  geom_line(data = pop1, aes(x = day, y = R, color = "Beta = 1.3, Gamma = 0.1")) +
  geom_line(data = pop2, aes(x = day, y = R, color = "Beta = 5, Gamma = 3")) +
  labs(x = "Day", y = "People Recovered", title = "Days and People Recovered from COVID")
```

Days and People Recovered from COVID


```
ggplot() +
  geom_line(data = pop3, aes(x = day, y = S, color = "Beta = 0.3, Gamma = 0.05")) +
  geom_line(data = pop1, aes(x = day, y = S, color = "Beta = 1.3, Gamma = 0.1")) +
  geom_line(data = pop2, aes(x = day, y = S, color = "Beta = 5, Gamma = 3")) +
  labs(x = "Day", y = "People Susceptible", title = "Days and People Susceptible COVID")
```

Days and People Susceptible COVID


```
ggplot() +
  geom_line(data = pop3, aes(x = day, y = I, color = "Beta = 0.3, Gamma = 0.05")) +
  geom_line(data = pop1, aes(x = day, y = I, color = "Beta = 1.3, Gamma = 0.1")) +
  geom_line(data = pop2, aes(x = day, y = I, color = "Beta = 5, Gamma = 3")) +
  labs(x = "Day", y = "People Infected", title = "Days and People Infected from COVID")
```

Days and People Infected from COVID
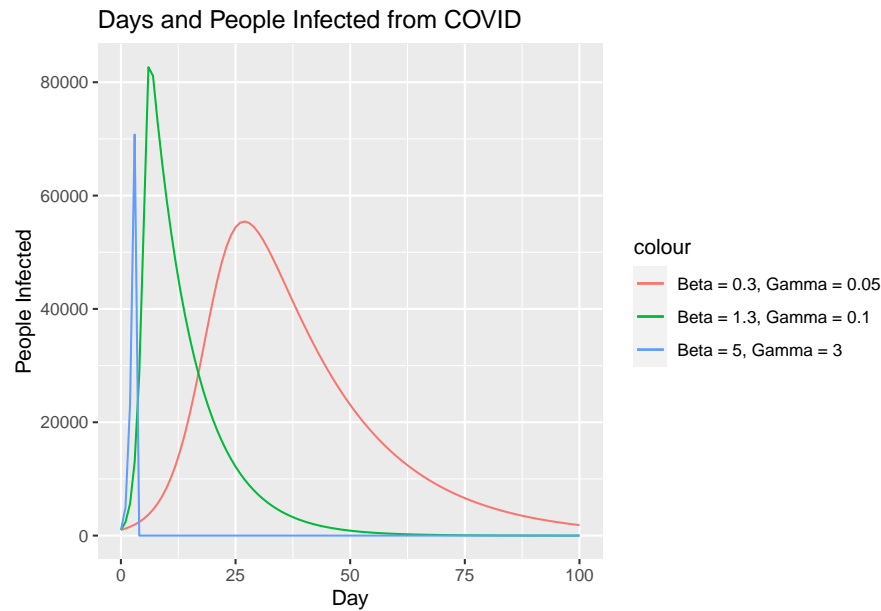
These graphs demonstrate that with higher Betas and Gammas, people recovered faster, the number of people susceptible is lower in less time, and the number of total people infected is lower.

Part 2: Basic Reproduction Ratio

Below: different R0s are used to create different SIR models, and then compared in a combined graph.

```
#Trial 1, R0 of 0.1
S <- 99000
I <- 1000
R <- 00
N <- S + I + R
pop <- c(0, S, I, R)
beta <- 0.2
gamma <- 2
R0 <- beta/gamma

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered

  # Add another row to pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
}
colnames(pop) <- c("day", "S", "I", "R")
popA1 <- pop %>% data.frame()
```

```r
#Trial 2, R0 of 1
S <- 99000
I <- 1000
R <- 00
N <- S + I + R
pop <- c(0, S, I, R)
beta <- 0.3
gamma <- 0.3
R0 <- beta/gamma

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered

  # Add another row to pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
}
colnames(pop) <- c("day", "S", "I", "R")
popA2 <- pop %>% data.frame()
```

```r
#Trial 3, R0 of 1.14
S <- 99000
I <- 1000
R <- 00
N <- S + I + R
pop <- c(0, S, I, R)
beta <- 0.8
gamma <- 0.7
R0 <- beta/gamma

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered

  # Add another row to pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
}
colnames(pop) <- c("day", "S", "I", "R")
popA3 <- pop %>% data.frame()
```
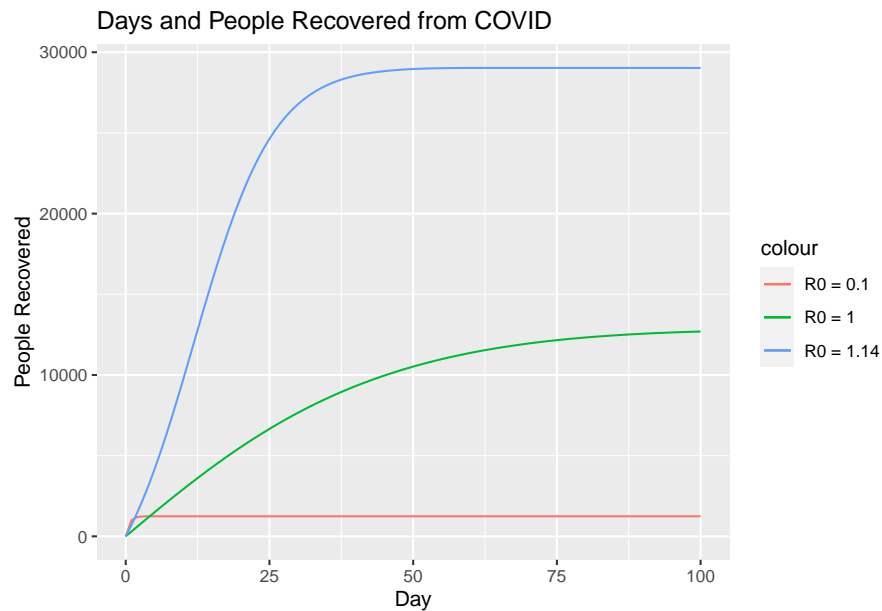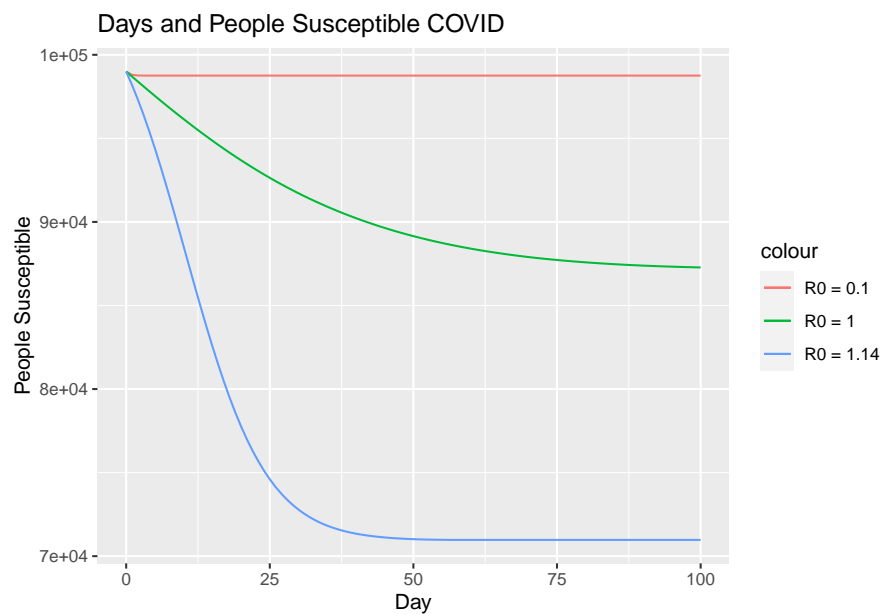
```
#Plotting and comparing
ggplot() + geom_line(data = popA1, aes(x = day, y = R, color = "R0 = 0.1")) +
  geom_line(data = popA2, aes(x = day, y = R, color = "R0 = 1")) +
  geom_line(data = popA3, aes(x = day, y = R, color = "R0 = 1.14")) +
  labs(x = "Day", y = "People Recovered", title = "Days and People Recovered from COVID")
```


Days and People Recovered from COVID

```
ggplot() + geom_line(data = popA1, aes(x = day, y = S, color = "R0 = 0.1")) +
  geom_line(data = popA2, aes(x = day, y = S, color = "R0 = 1")) +
  geom_line(data = popA3, aes(x = day, y = S, color = "R0 = 1.14")) +
  labs(x = "Day", y = "People Susceptible", title = "Days and People Susceptible COVID")
```


Days and People Susceptible COVID

```r
ggplot() + geom_line(data = popA1, aes(x = day, y = I, color = "R0 = 0.1")) +
  geom_line(data = popA2, aes(x = day, y = I, color = "R0 = 1")) +
  geom_line(data = popA3, aes(x = day, y = I, color = "R0 = 1.14")) +
  labs(x = "Day", y = "People Infected", title = "Days and People Infected from COVID")
```
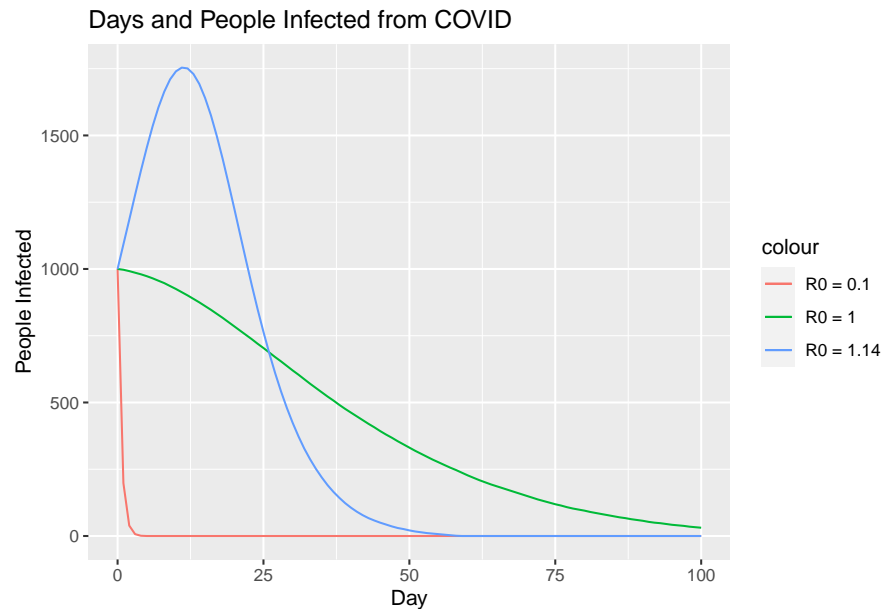

Above, I have plotted different curves for the recovered, infected, and susceptible populations for a R0 of 0.1, 1, and 1.14. It is difficult to make overarching conclusions about how curves change for different R0s because the beta and gamma values also have an influence. The next section will investigate what effect Beta and Gamma values have on the recovered, infection, and susceptible populations.

Next, I am trying the same R0 of 15 with different Betas and Gammas:

```r
#Trial 1: Beta = 1.5, Gamma = 0.1
S <- 99000
I <- 1000
R <- 00
N <- S + I + R
pop <- c(0, S, I, R)
beta <- 1.5
gamma <- 0.1
R0 <- beta/gamma

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered

  # Add another row to pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
```

```r
}
colnames(pop) <- c("day", "S", "I", "R")
popA <- pop %>% data.frame()


#Trial 2: Beta = 7.5, Gamma = 0.5
S <- 99000
I <- 1000
R <- 00
N <- S + I + R
pop <- c(0, S, I, R)
beta <- 7.5
gamma <- 0.5
R0 <- beta/gamma

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered

  # Add another row to pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
}
colnames(pop) <- c("day", "S", "I", "R")
popB <- pop %>% data.frame()


#Trial 3: Beta = 30, Gamma = 0.8
S <- 99000
I <- 1000
R <- 00
N <- S + I + R
pop <- c(0, S, I, R)
beta <- 30
gamma <- 0.8
R0 <- beta/gamma

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered

  # Add another row to pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
}
```
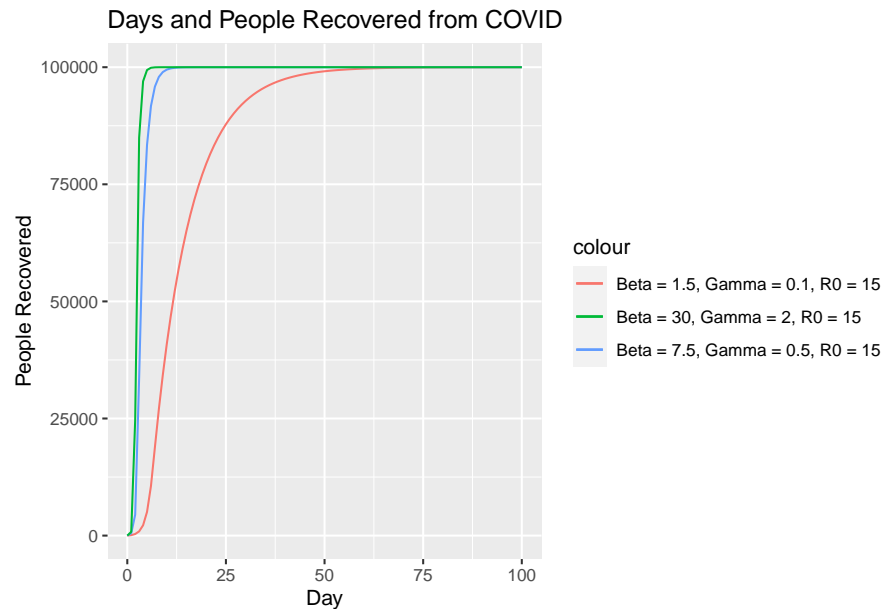
```
colnames(pop) <- c("day", "S", "I", "R")
popC <- pop %>% data.frame()
```

```
#Plotting and comparing
ggplot() +
  geom_line(data = popA, aes(x = day, y = R, color = "Beta = 1.5, Gamma = 0.1, R0 = 15")) +
  geom_line(data = popB, aes(x = day, y = R, color = "Beta = 7.5, Gamma = 0.5, R0 = 15")) +
  geom_line(data = popC, aes(x = day, y = R, color = "Beta = 30, Gamma = 2, R0 = 15")) +
  labs(x = "Day", y = "People Recovered", title = "Days and People Recovered from COVID")
```


Days and People Recovered from COVID

```
ggplot() +
  geom_line(data = popA, aes(x = day, y = S, color = "Beta = 1.5, Gamma = 0.1, R0 = 15")) +
  geom_line(data = popB, aes(x = day, y = S, color = "Beta = 7.5, Gamma = 0.5, R0 = 15")) +
  geom_line(data = popC, aes(x = day, y = S, color = "Beta = 30, Gamma = 2, R0 = 15")) +
  labs(x = "Day", y = "People Susceptible", title = "Days and People Susceptible COVID")
```

Days and People Susceptible COVID
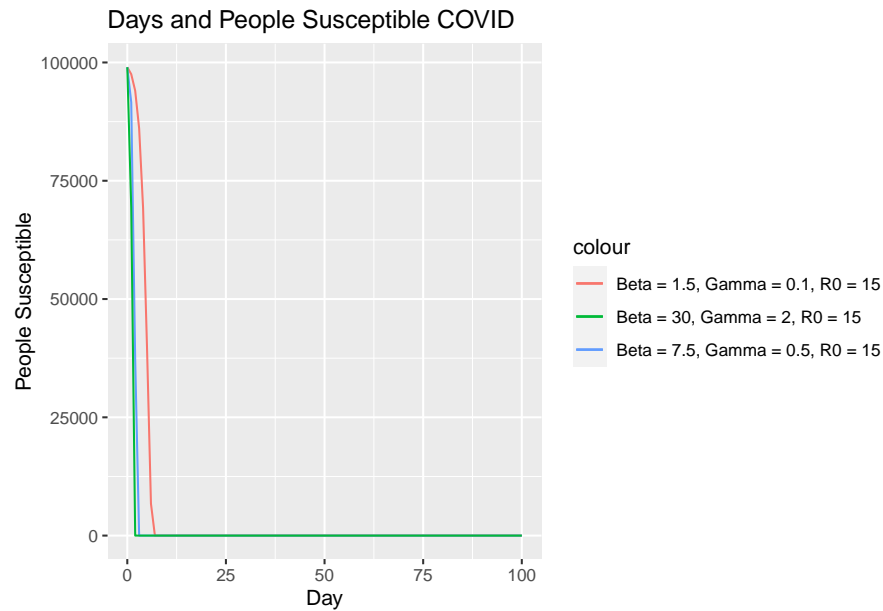

```
ggplot() +
  geom_line(data = popA, aes(x = day, y = I, color = "Beta = 1.5, Gamma = 0.1, R0 = 15")) +
  geom_line(data = popB, aes(x = day, y = I, color = "Beta = 7.5, Gamma = 0.5, R0 = 15")) +
  geom_line(data = popC, aes(x = day, y = I, color = "Beta = 30, Gamma = 2, R0 = 15")) +
  labs(x = "Day", y = "People Infected", title = "Days and People Infected from COVID")
```

Days and People Infected from COVID
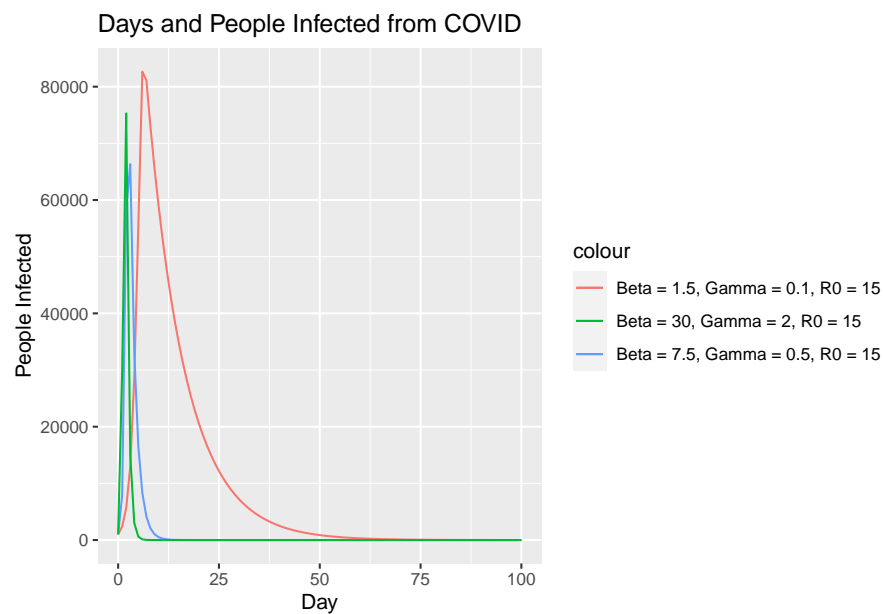
This section has demonstrated that the Beta & Gamma combination is more important than the R0 itself. With the same R0, the population that recovered faster, was less susceptible, and fewer infected was the population with the highest Beta and Gamma.
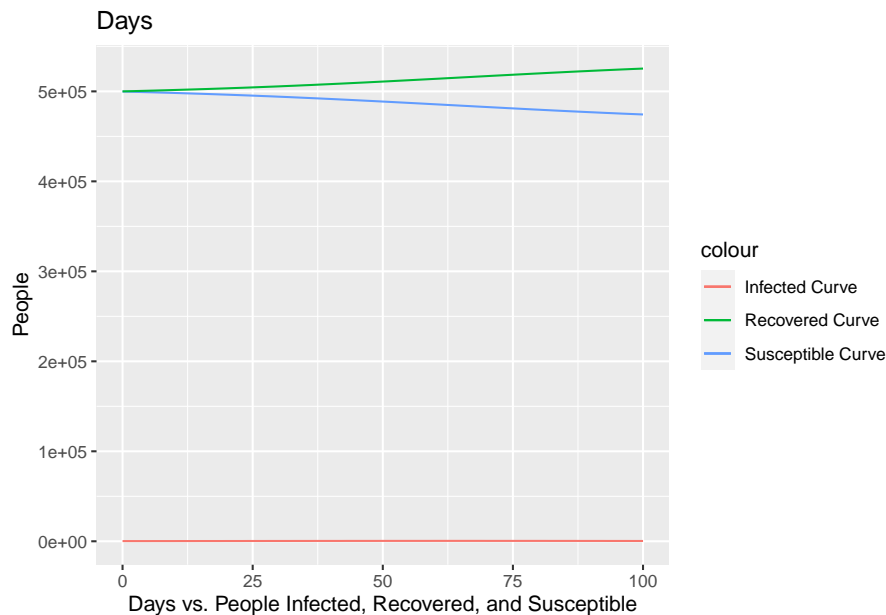
Part 3: Herd Immunity

```r
S <- 499800
I <- 200
R <- 500000
N <- S + I + R

pop <- c(0, S, I, R)
beta <- 1.45
gamma <- 0.7

# Repeat 100 times
for(t in 1:100){
  inf.prop <- I/N
  new.infected <- min(S, floor(beta * S * inf.prop))
  new.recovered <- min(I, ceiling(gamma * I))
  I <- I + new.infected - new.recovered
  S <- S - new.infected
  R <- R + new.recovered

  # Add another row to pop, with the current day t, and the current values
  # of S, I, and R
  pop <- rbind(pop, c(t, S, I, R))
}
colnames(pop) <- c("day", "S", "I", "R")
pop1 <- pop %>% data.frame()
ggplot() + geom_line(data = pop1, aes(x = day, y = S, color = "Susceptible Curve")) +
  geom_line(data = pop1, aes(x = day, y = I, color = "Infected Curve")) +
  geom_line(data = pop1, aes(x = day, y = R, color = "Recovered Curve")) +
  labs(x = "Days vs. People Infected, Recovered, and Susceptible", y = "People",
       title = "Days")
```


I found a population with beta 1.45 and a gamma of 0.7 demonstrates herd immunity. The recovered

population goes up around 10%, and the susceptible poppulation plateaus at around 50,000. I hypothesized that low beta values (because beta is the contact rate) and high gamma values (fration of infected people who recover) were required to achieve herd immunity, and thus tried different betas and gammas in that range. The Infected population slightly increases (although can't be seen in the graph).

Part 4: Reading in the Current Data

```
confirmed <-
  read.csv(
    "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_

confirmed_longer <- confirmed %>%
  pivot_longer(colnames(confirmed)[5:112], names_to = "Date", values_to = "Confirmed")

Confirmed_NewCases <- confirmed_longer %>%
  select(-Province.State) %>%
  group_by(Date,Country.Region) %>%
  mutate(SumConfirmed = sum(Confirmed)) %>%
  select(-Long, -Lat, -Confirmed) %>% unique() %>% ungroup() %>%
  group_by(Country.Region) %>%
  mutate(NewCases = (c(SumConfirmed, 0) -
                      c(0, SumConfirmed))[1:length(c(SumConfirmed, 0) -
                                                    c(0, SumConfirmed))-1])
```

```
days.since.jan.22 <- function(date.str){
  date.str <- substr(date.str, 2, str_length(date.str))
  comps <- strsplit(date.str, "[.]")[[1]]
  month <- as.numeric(comps[1])
  day <- as.numeric(comps[2])

  if (month == 1){
    day - 22
  }else if(month == 2){
    9 + day
  }else if(month == 3){
    9 + 29 + day
  }else if(month == 4){
    9 + 29 + 31 + day
  }else if(month == 5){
    9 + 29 + 31 + 30 + day
  }
}

Final <- Confirmed_NewCases %>%
  mutate(Days.Since.Jan22 = sapply(Date, FUN = days.since.jan.22))
```

Part 5: SIR Model and Country Data

1. Below, I write a function that displays on the same graph the number of new cases in a given country, and the number of new cases according to the SIR model, with parameters Beta and Gamma. My

function takes in the population, country, beta, and gamma.

```r
SIRModel <- function(pops, country, beta1, gamma1) {
  country$NewCases[country$Date == "X1.22.20"] = 0

  #calculate number of new cases for B and Gamma based on SIR model
  n <- pops
  R <- 00
  I <- 200
  S <- n - R - I
  N <- S + I + R
  beta <- beta1
  gamma <- gamma1
  pop <- c(0,S,I,R)

  # repeat 100 times
  for(t in 1:100){
    inf.prop <- I/N
    new.infected <- min(S, floor(beta * S * inf.prop))
    new.recovered <- min(I, ceiling(gamma * I))
    I <- I + new.infected - new.recovered
    S <- S - new.infected
    R <- R + new.recovered

    # Add another row to pop, with the current day t, and the current values
    # of S, I, and R
    pop <- rbind(pop, c(t, S, I, R))
  }
  colnames(pop) <- c("day", "S", "I", "R")
  pop.new <- pop %>% data.frame() %>%
    mutate(NewCases = (c(I, 0) - c(0, I))[1:length(c(I, 0) - c(0, I))-1])
  pop.new[pop.new < 0] <- 0
  pop.new

  #plot cases on graph
  ggplot(data = country, aes(x = Days.Since.Jan22, y = NewCases)) +
    geom_line(color = "black") +
    geom_line(data = pop.new, aes(x = day, y = NewCases, color = "blue"))
}
```

2. Below, I write a function that plots log of new cases vs log of predicted new cases with different betas and gammas

Social Distancing Start Dates: China: Jan 22, US: April 3 (roughly, dependent on state), Italy: March 9, South Korea: March 21

Population: China: 1.4 billion, U.S: 328 million, Italy: 60 million, South Korea: 52 million

```r
#Subsetting by Region, post social distancing dates
China <- Final %>% filter(Country.Region == "China")
US <- Final %>% filter(Country.Region == "US") %>% filter(Days.Since.Jan22 >= 72)
Italy <- Final %>% filter(Country.Region == "Italy") %>% filter(Days.Since.Jan22 >= 47)
SK<- Final %>% filter(Country.Region == "Korea, South") %>% filter(Days.Since.Jan22 >= 59)
```

```r
LogFunc <- function(pops, country, beta1, gamma1) {
  country <- country
  country <- country %>% mutate(logcases = log(1 + NewCases))

  #change I based on country
  if (country$Country.Region[1] == factor("US")) {
    I <- 500000
  }else if (country$Country.Region[1] == factor("Italy")) {
    I <- 50000
  }else if (country$Country.Region[1] == factor("Korea, South")){
    I <- 25000
  }else {
    I <- 10000
  }

  #variables
  n <- pops
  R <- 00
  S <- n - R - I
  N <- S + I + R
  I.new <- 0
  beta <- beta1
  gamma <- gamma1
  pop <- c(0,S,I,R, I.new)

  # repeat 100 times
  for(t in 1:100){
    inf.prop <- I/N
    new.infected <- min(S, floor(beta * S * inf.prop))
    new.recovered <- min(I, ceiling(gamma * I))
    I <- I + new.infected - new.recovered
    S <- S - new.infected
    R <- R + new.recovered
    I.new <- new.infected
    # Add another row to pop, with the current day t, and the current values
    # of S, I, and R
    pop <- rbind(pop, c(t, S, I, R, I.new))
  }
  colnames(pop) <- c("day", "S", "I", "R", "I.new")
  pop.new <- pop %>%
    data.frame()
  pop.new$I.new[1] = pop.new$I[1]
  pop.new <- pop.new %>%
    mutate(PREDlogcases = log(1 + I.new))
  #pop.new

    #filtering for post-social distancing days
  if (country$Country.Region[1] == factor("US")) {
    pop.new <- pop.new %>% filter(day >= 72)
  }else if (country$Country.Region[1] == factor("Italy")) {
    pop.new <- pop.new %>% filter(day >= 47)
  }else if (country$Country.Region[1] == factor("Korea, South")){
    pop.new <- pop.new %>% filter(day >= 59)
```

```
  }else {
    pop.new <- pop.new %>% filter(day >= 1)
  }

  #calculate sum of squared differences
  pop.new1 <- pop.new %>% ungroup() %>% select(day, PREDlogcases)
  country1 <- country %>%
    mutate(logcases = log(1 + NewCases)) %>% ungroup() %>%
    select(Days.Since.Jan22, logcases) %>% rename(day = Days.Since.Jan22)
  Both <- merge(pop.new1, country1)
  SumSquared <- Both %>% ungroup() %>%
    mutate(SSE_indv = (logcases - PREDlogcases)**2) %>%
    mutate(SSE = sum(SSE_indv)) %>%
    select(-SSE_indv)
  cat(as.character(country$Country.Region[1]),": SSE is", SumSquared$SSE[1])

  #plot cases on graph
  ggplot(data = country, aes(x = Days.Since.Jan22, y = logcases)) +
    geom_line(aes(color = "Reported New Cases")) +
    geom_line(data = pop.new, aes(x = day, y = PREDlogcases, color = "Predicted New Cases")) +
    scale_color_manual(values = c("red", "black")) +
    labs(x = "Days since Jan 22", y = "New Cases (logged)",
         title =  as.character(country$Country.Region[1]))
}

LogFunc(1.4e+09, China, 0.1999, 0.2533)
```
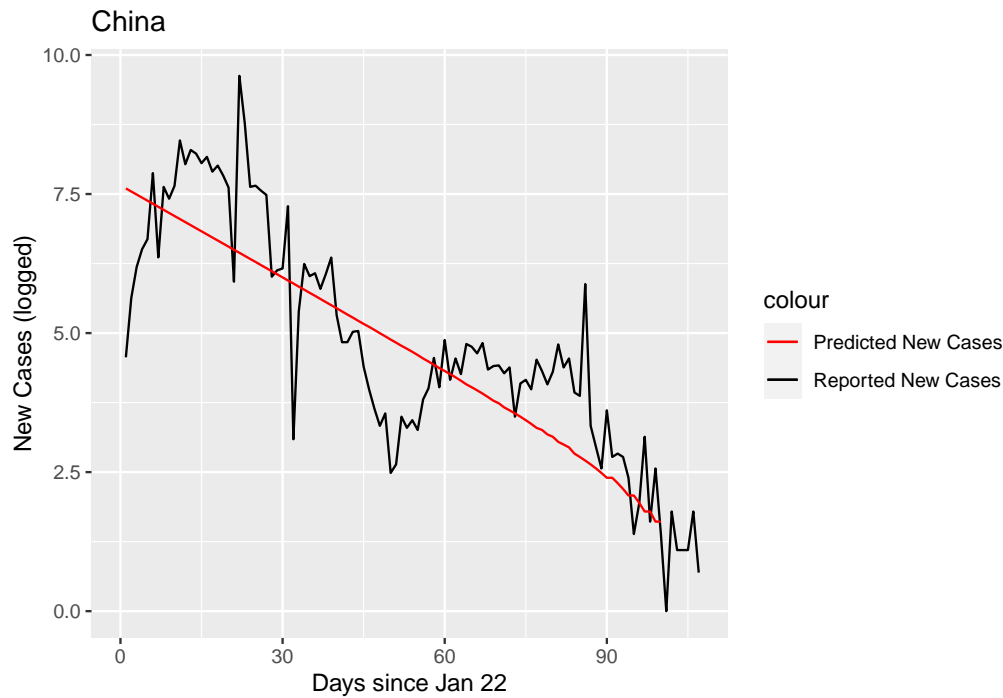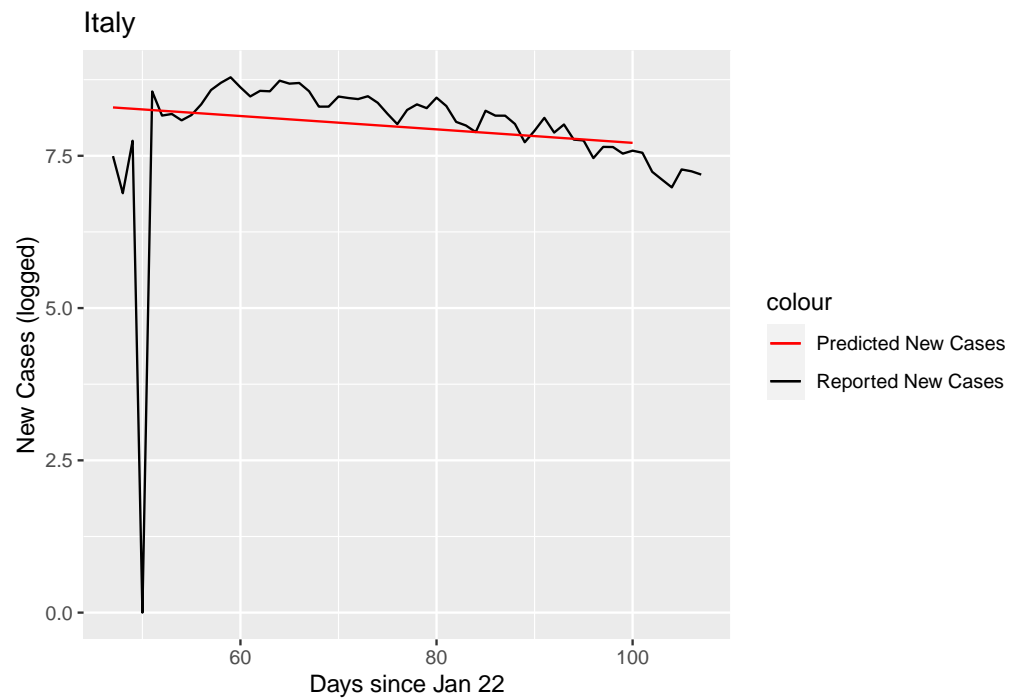
```
## China : SSE is 4388.294
```
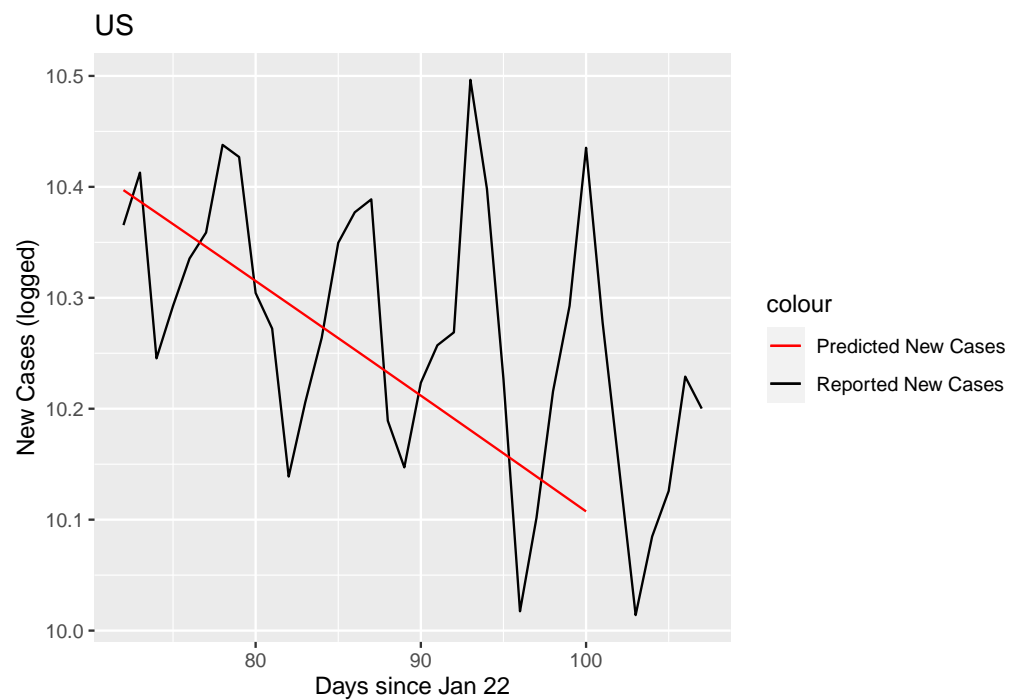

China

```
LogFunc(6e+07, Italy, 0.13, 0.14)
```

```
## Italy : SSE is 76.64315
```
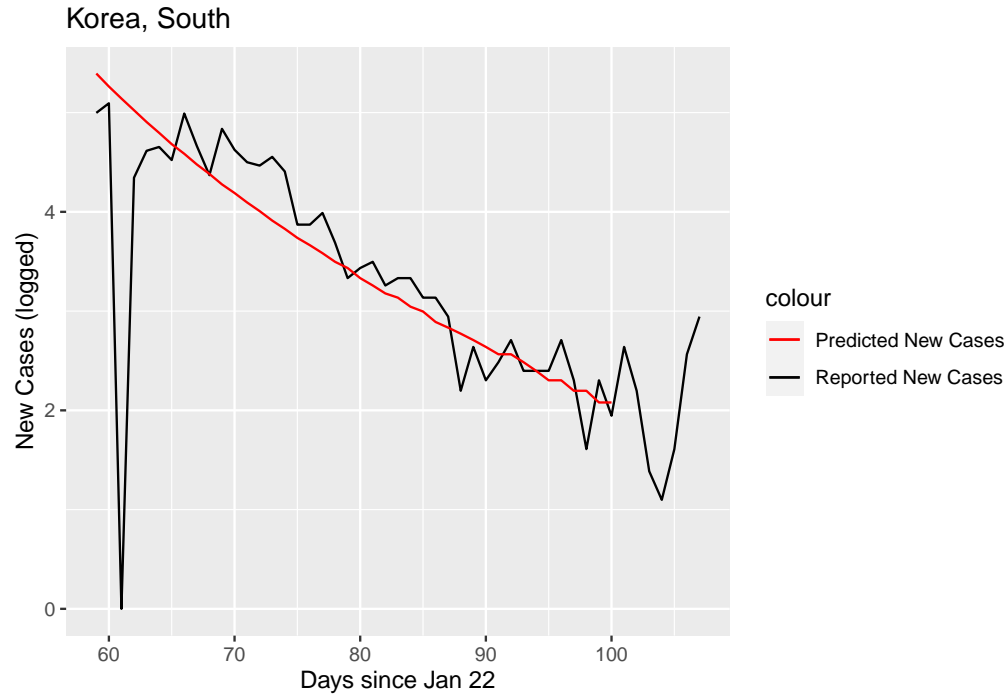


```
LogFunc(3.28e+08, US, 0.13, 0.1385)
```

```
## US : SSE is 0.4587726
```

```
LogFunc(5.2e+07, SK, 0.5, 0.05)
```

```
## Korea, South : SSE is 30.67856
```


Korea, South

For the US, Italy, China, and South Korea I attempted to find values of Beta and Gamma that predicted the reported number of new cases as well as possible. I did this by displaying curves for various Beta and Gammas (and I's) that plotted the days by the (logged) new cases, until I found a curve that matched the trend of the actual (logged) new cases in each respective country. These graphs show post-social distancing days by country.

I estimated the Beta and Gammas after social distancing measures were taken, and used google and estimation to find the date that social distancing began in each country. I found the best Beta and Gamma for China was 0.1999 and 0.2533, for Italy was 0.13 and 0.14, for the US was 0.13 and 0.1385, and for South Korea was 0.5 and 0.05. After finding the correct Beta and Gammas, I calculated the SSEs for each country. The model with the lowest SSE is the United States.

With the models I have created for China, Italy, US, and South Korea (with different pairs of Beta and Gamma), I predict that the number of new cases will continue to decrease in each of these countries. My models predict the number of new cases in China will decrease by around 3.75 (logged) new cases in the next month; the number of new cases in Italy will decrease by around 1.25 (logged) new cases in the next month; the number of new cases in the US will decrease around 0.15 (logged) new cases in the next month, and the number of new cases in South Korea will decrease around 2.5 (logged) new cases in the next month.

However, limitations of this model are that accurate results depend on accurately inputting the number of people initially infected and the total population. This is important to keep in mind when further conclusions are being made.

Sources: https://www.scmp.com/news/asia/east-asia/article/ 3078460/coronavirus-south-korea-extends-social-distancing-limit-new

https://www.axios.com/italy-coronavirus-timeline-lockdown-deaths-cases-2adb0fc7-6ab5-4b7c-9a55-bc6897494dc6.html