HW3

Daniel Yoon

2024-02-08

```
library(tidyverse)
```

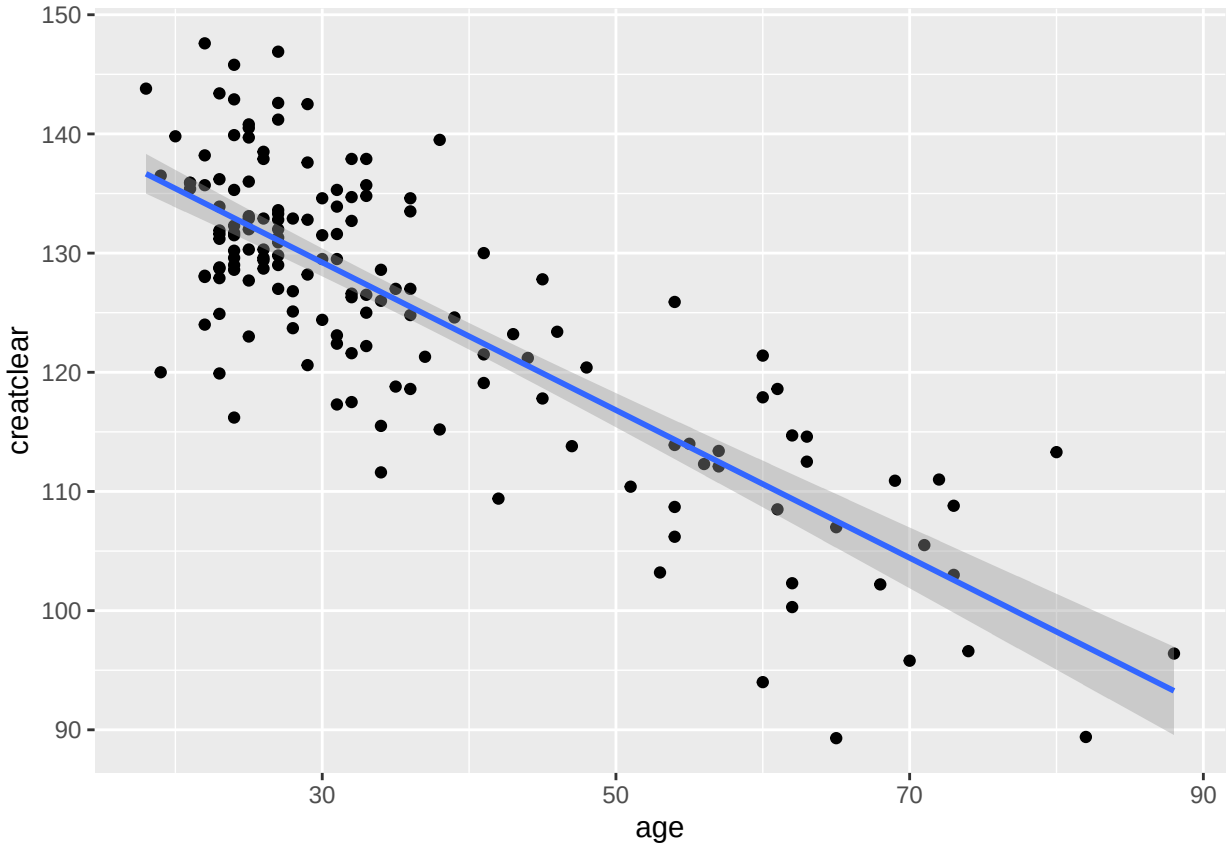
1.

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)

creatinine <- read.csv("creatinine.csv")
# Create a scatter plot with a linear regression line
ggplot(data = creatinine, aes(x = age, y = creatclear)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# linear regression model
model_creatinine <- lm(creatclear ~ age, data = creatinine)

# coefficients of the model
coef(model_creatinine)
```

```
## (Intercept)         age
## 147.8129158  -0.6198159
```

```
# Predict creatclear for age 55 using the model
new_data <- data.frame(age = 55)
predict(model_creatinine, newdata = new_data)
```

```
##       1
## 113.723
```

a) What creatinine clearance rate should we expect for a 55-year-old?

The expected creatinine clearance rate for a 55-year-old patient is 113.723mL/min. I found this value using a linear regression line. The equation for creatinine clearance rate is -0.6198 * x(age) + 147.8129. I put 55 in the age value so that I can find a specific value for a 55-year old.

b) How does creatinine clearance rate change with age?

Creatinine clearance rate changes -0.6198ml/minute per year, and I found it using linear regression line where x = age and y = clearance rate. The slope indicates the change of clearance rate in average as age increases by 1.

c) Whose creatinine clearance rate is healthier (higher) for their age: a 40-year-old with a rate of 135, or a 60-year-old with a rate of 112?

A 40-year-old with a rate of 135 person is healthier than 60-year-old with a rate of 112. It is because creatinine clearance rate of 135 is higher than the average rate which is -0.6198 * 40 + 147.8129 = 123.0209. However, a 60-year-old of 112 have an average creatinine clearance rate, which is -0.6198 * 60 + 147.8129 = 110.6249. Although there is a 60-year-old rate is little bit higher than the average, but it is a smaller difference than the 40-year-old case.

```
library(tidyverse)
library(ggplot2)
library(mosaic)
```

2.

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2

##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following object is masked from 'package:purrr':
##
##     cross

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
marketmodel <- read.csv('marketmodel.csv')
# Function to create and analyze linear models for each stock
analyze_stock <- function(stock_name) {
  # Plotting
  ggplot(marketmodel) +
    geom_point(aes(x = SPY, y = !!sym(stock_name))) +
    geom_smooth(aes(x = SPY, y = !!sym(stock_name)), method = 'lm')

  # Linear model
  model <- lm(as.formula(paste(stock_name, "~ SPY")), data = marketmodel)

  # Display coefficients and R-squared
  print(coef(model))
  print(summary(model)$r.squared)

  # Return coefficients and R-squared for later use
  return(tibble(
    Ticker = stock_name,
    Intercept = coef(model)[1],
    Slope = coef(model)[2],
    R_squared = summary(model)$r.squared
  ))
}

# List of stocks
stocks <- c("AAPL", "GOOG", "MRK", "JNJ", "WMT", "TGT")

# Analyze each stock and store results in a list
results_list <- lapply(stocks, analyze_stock)
```

```
## (Intercept)         SPY
## 0.009189277 1.065601182
## [1] 0.01336246
##  (Intercept)          SPY
## 0.0002330467 0.9967745749
## [1] 0.6483015
##   (Intercept)           SPY
## -0.0001540208  0.7136140905
## [1] 0.483701
##   (Intercept)           SPY
## -2.410714e-05  6.771930e-01
## [1] 0.501943
##  (Intercept)          SPY
## 0.0006781104 0.5189810644
## [1] 0.2853233
## (Intercept)         SPY
## 0.001583341 0.707648462
## [1] 0.2478813
```

```r
# Combine results into a tibble
regression_data <- bind_rows(results_list)

# Print the final regression data
print(regression_data)
```

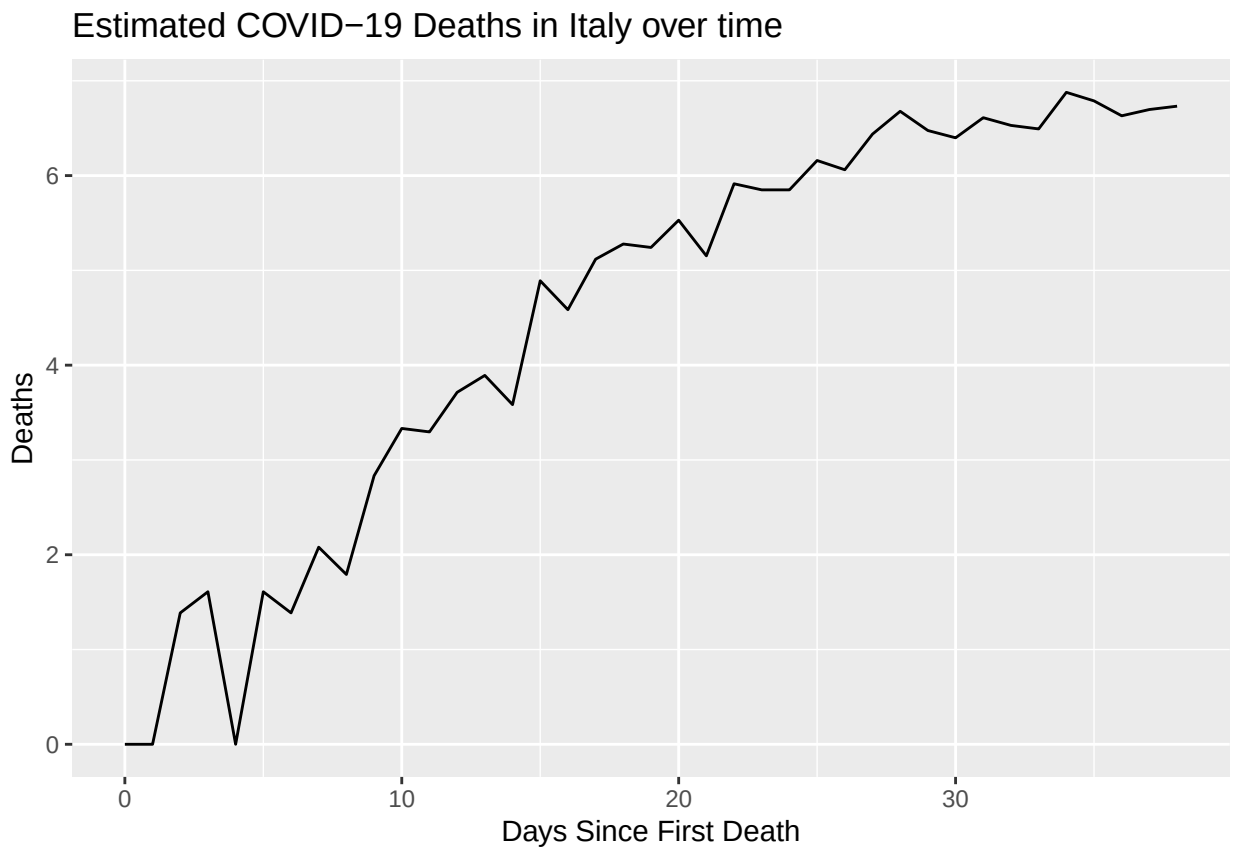
```
## # A tibble: 6 x 4
##   Ticker  Intercept Slope R_squared
##   <chr>       <dbl> <dbl>     <dbl>
## 1 AAPL    0.00919   1.07     0.0134
## 2 GOOG    0.000233  0.997    0.648
## 3 MRK    -0.000154  0.714    0.484
## 4 JNJ    -0.0000241 0.677    0.502
## 5 WMT     0.000678  0.519    0.285
## 6 TGT     0.00158   0.708    0.248
```

Beta is a measure of systematic risk which indicates percentage change in asset's return as market portfolio changes by 1%. Beta is calculated by dividing amount of each stock invested by the total invested amount and adding them up. The slope category of the table above indicates increase of each company's systematic risk as 1% of market portfolio increases, which is Beta. The intercept category indicates the constant value of systematic risk of each company and R-squared category indicates how the actual systematic risk are distributed from the predicted systematic risk. Among six stocks, AAPL has the highest systematic risk, while WMT has the lowest systematic risk.

```r
library(tidyverse)
covid <-read.csv('covid.csv')

italy <- subset(covid, country == "Italy")
spain <- subset(covid, country == "Spain")


# estimated death growth rate for Italy
ggplot(italy) +
  geom_line(aes(x = days_since_first_death, y = log(deaths))) +
  labs(title = "Estimated COVID-19 Deaths in Italy over time",
       x = "Days Since First Death",
       y = "Deaths")
```



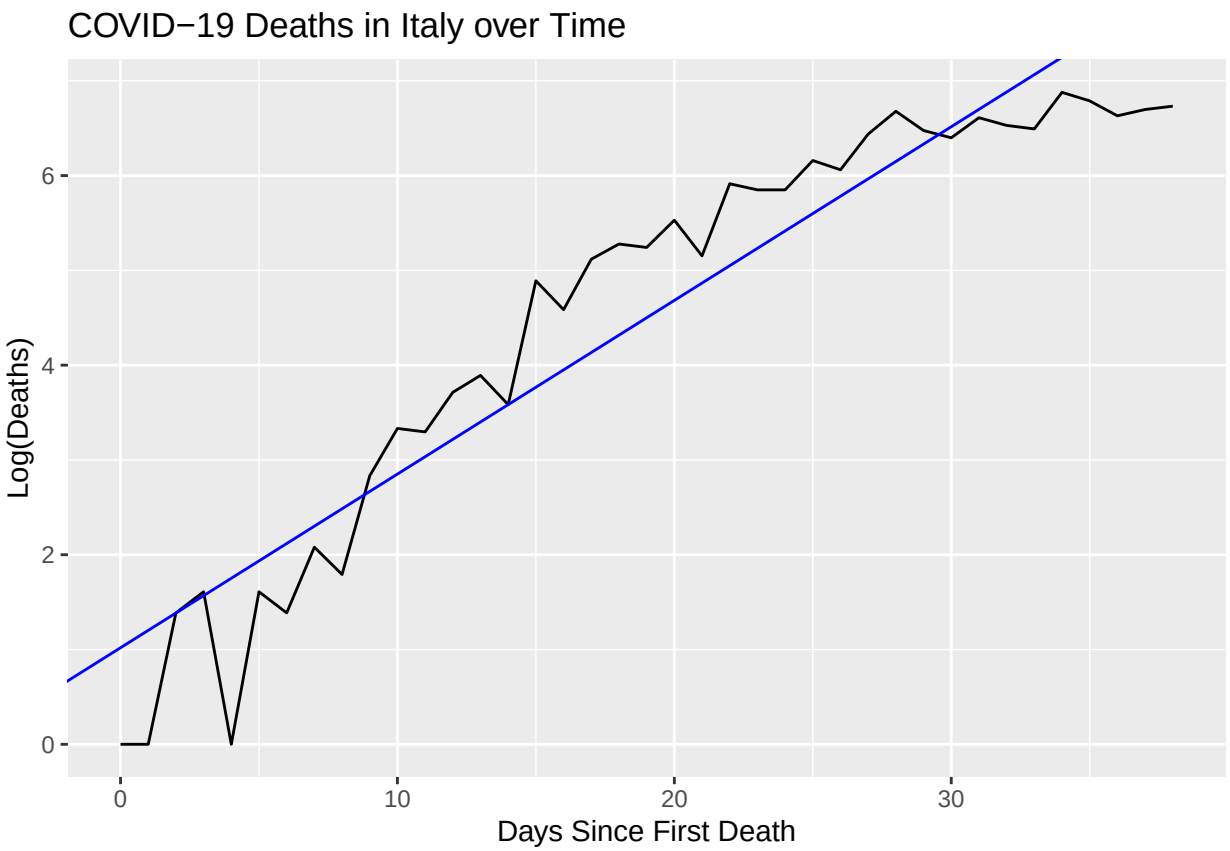
3.

```
lm_italy = lm(log(deaths) ~ days_since_first_death, data=italy)
coeff_Italy <- round(coef(lm_italy),3)
coeff_Italy
```

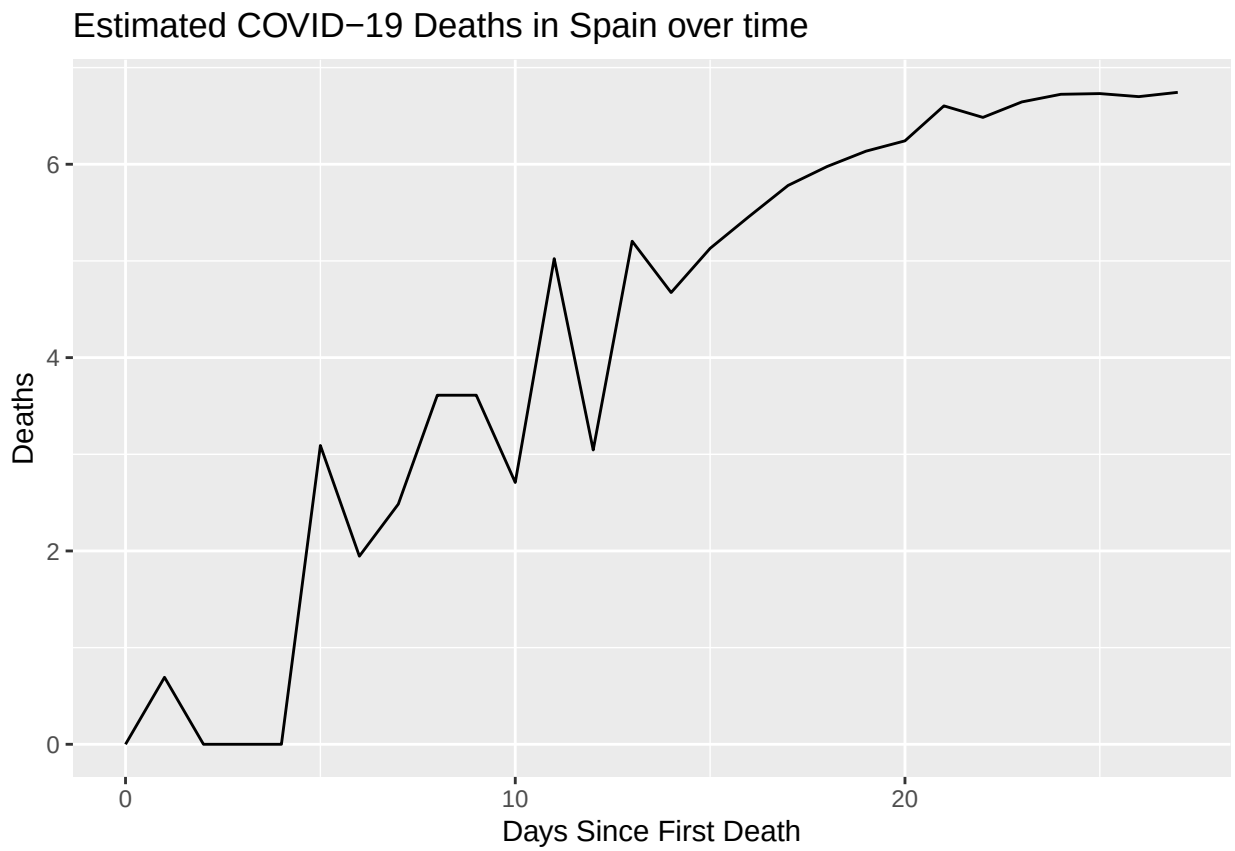
```
##            (Intercept) days_since_first_death
##                  1.019                  0.183
```

```
Italy_double <- log(2) / coef(lm_italy)[2]

ggplot(italy) +
  geom_line(aes(x = days_since_first_death, y = log(deaths))) +
  geom_abline(intercept = coef(lm_italy)[1], slope = coef(lm_italy)[2], color = 'blue') +
  labs(title = "COVID-19 Deaths in Italy over Time",
       x = "Days Since First Death",
       y = "Log(Deaths)")
```




```
#Estimated growth for Spain
ggplot(spain) +
  geom_line(aes(x = days_since_first_death, y = log(deaths))) +
  labs(title = "Estimated COVID-19 Deaths in Spain over time",
       x = "Days Since First Death",
       y = "Deaths")
```

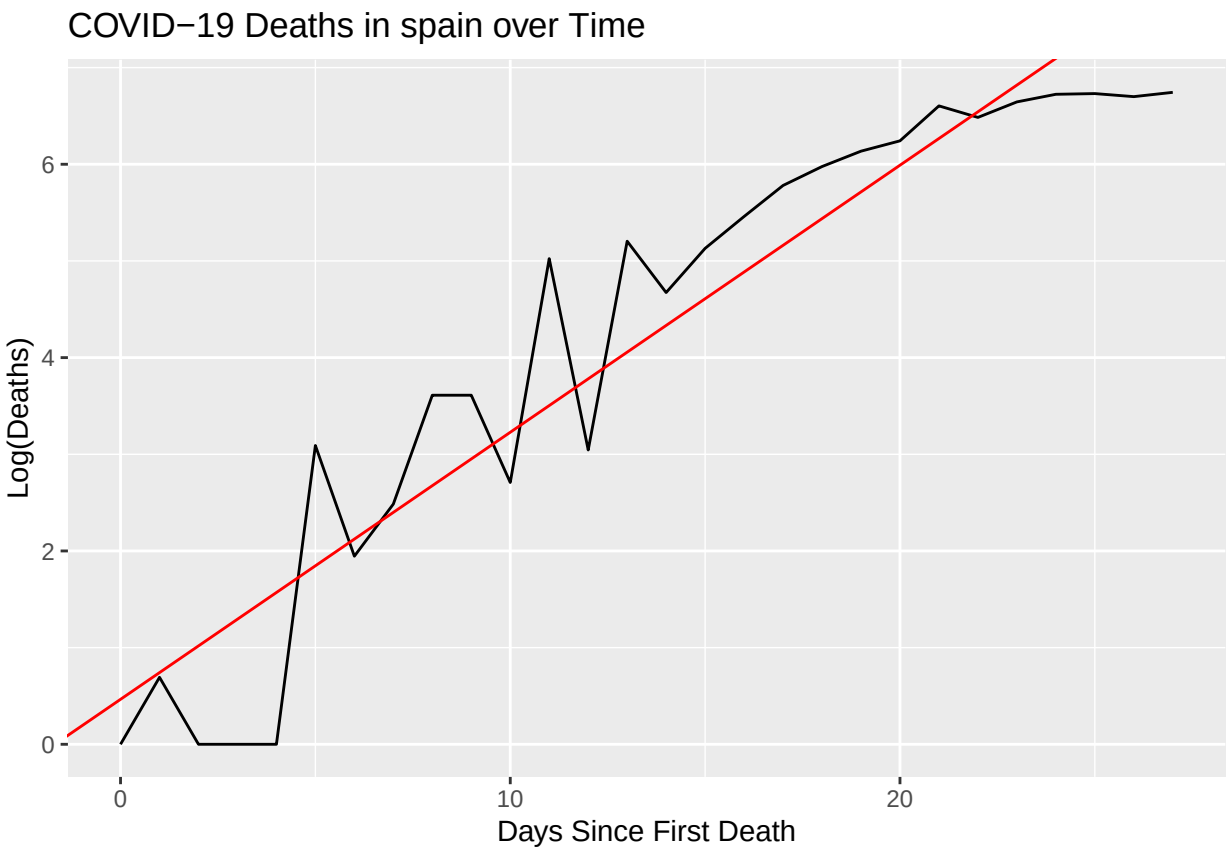



```
lm_spain = lm(log(deaths) ~ days_since_first_death, data=spain)
coeff_Spain <- round(coef(lm_spain),3)
coeff_Spain
```

```
##            (Intercept) days_since_first_death
##                  0.465                  0.276
```

```
spain_double <- log(2) / coef(lm_spain)[2]

ggplot(spain) +
  geom_line(aes(x = days_since_first_death, y = log(deaths))) +
  geom_abline(intercept = coef(lm_spain)[1], slope = coef(lm_spain)[2], color = 'red') +
  labs(title = "COVID-19 Deaths in spain over Time",
       x = "Days Since First Death",
       y = "Log(Deaths)")
```

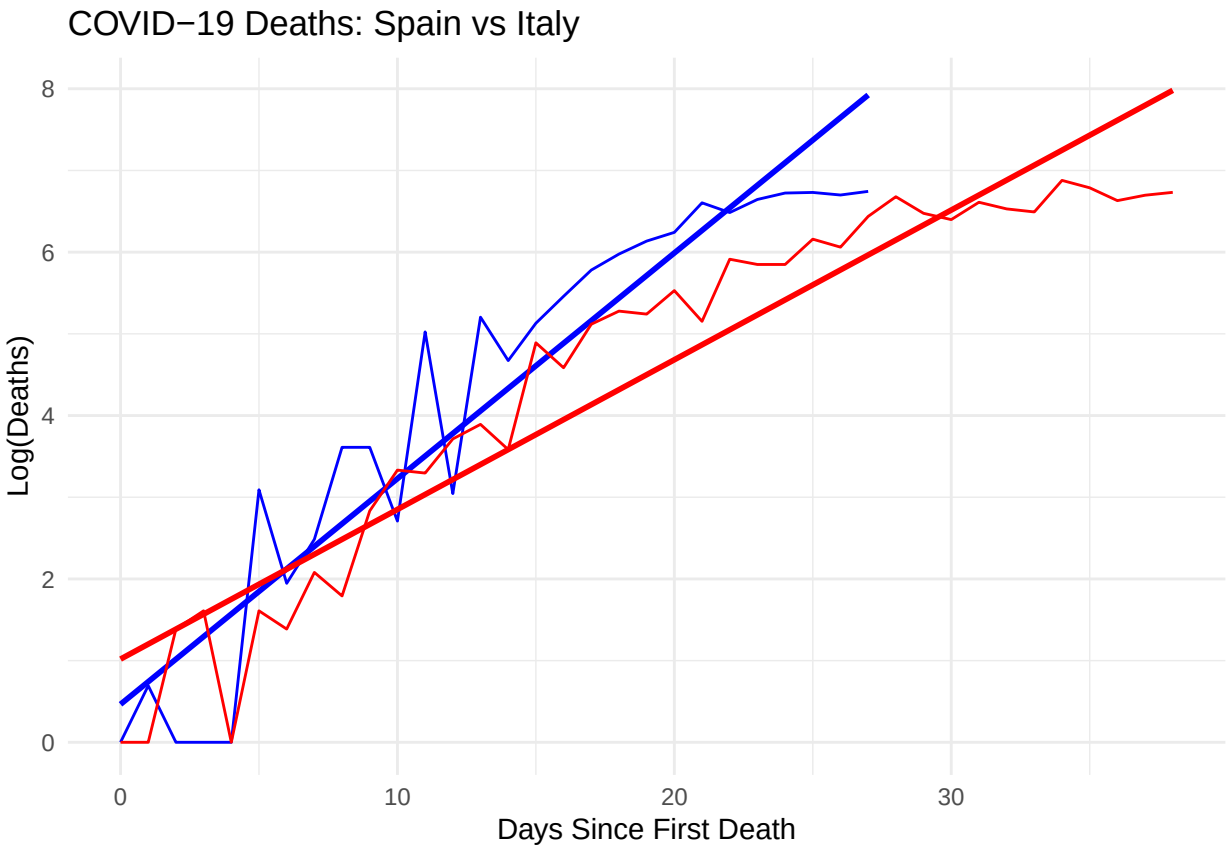



```
# Plotting Spain data
ggplot(spain) +
  geom_line(aes(x = days_since_first_death, y = log(deaths)), color = "blue") +
  geom_smooth(aes(x = days_since_first_death, y = log(deaths)), method = "lm", se = FALSE, color = "blu

# Adding Italy data
  geom_line(data = italy, aes(x = days_since_first_death, y = log(deaths)), color = "red") +
  geom_smooth(data = italy, aes(x = days_since_first_death, y = log(deaths)), method = "lm", se = FALSE

# Adjusting labels and theme
  labs(title = "COVID-19 Deaths: Spain vs Italy",
       x = "Days Since First Death",
       y = "Log(Deaths)") +
  scale_color_manual(values = c("blue", "red")) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



Between two countries, Italy and Spain, had different doubling time and estimated growth rate. Italy had rate of 0.183, while Spain had rate of 0.276. The doubling time for Italy is 4, while the doubling time for Spain is 3. Therefore, Spain has faster spread rate than Italy.