Lag Metrics

Dileka Gunawardana

4/13/2020

```
setwd("C:/Users/Dilek/D2K Covid-19")
cases <- read.csv("Confirmed Cases.csv")
deaths <- read.csv("Deaths.csv")
```

**METRIC 1: Difference Between Cases and Deaths

According to a study, “Predicition and Analysis of Coronavirus Disease 2019” by Jia et. al (https://arxiv.org/ftp/arxiv/papers/2003/2003.05447.pdf (https://arxiv.org/ftp/arxiv/papers/2003/2003.05447.pdf)), the best way to model both the deaths and the cases is via a logistic regression. According to a study, “Estimates of the Severity of Coronavirus Disease 2019: a Model-Based Analysis” by Verity et. al (https://www.thelancet.com/action/showPdf?pii=S1473-3099%2820%2930243-7 (https://www.thelancet.com/action/showPdf?pii=S1473-3099%2820%2930243-7)), the average length of time from the onset of symptoms to death was relatively consistent - a 95% confidence interval from 16.9 - 19.2 days with a mean of 17.8 days. However, a Canadian epidimiologist, Dr. David Fisman claimed that period might be closer to 30 days.

When examining a typical COVID-19 patient who, unfortunately, passes away, the following timeline can be constructed:

1. Onset of patient’s symptoms (only noticetable in ~50% of cases)
2. COVID-19 test administered
3. COVID-19 test results recieved –> Cases Curve
4. Death of the patient –> Deaths Curve

If we assume that the length of time between (1) and (4) is some constant like 17.8 or 30 days and consider “lag” to be the time between (1) and (3), then “lag” can be calculated for different communities by subtracting from 17.8 the difference between the logistic regressions for the cases and deaths. The date that a patient dies is not going to be affected by the lag time, but the date that their case was reported will be. Therefore, if the assumptions above are satisfied, a metric to consider for lag is the difference in days between the peak cases and the peak deaths. The larger these lag metrics are, the longer the lag period was.

First, wrote function to extract data for each country from the main file to be plottable:

```
per_county <- function(n,data){
  # n is n'th row of the data that corresponds to the n'th county
  # Data contains either number of cases or deaths per day
  # Output is a data frame with the number of days since March 8th 2020 and the number of casse
 for each of those time frames
  a <- c()
  b <- c()
  count = 1
  for (num in data[n,]){
    a <- append(a, count)
    b <- append(b, num)
    count = count + 1
  }
  return(list("days"=a,"numbers"=b))
}
```

Created a function where, given a certain state's name, will calculate the lag metric for each county within that state.

```
setwd("C:/Users/Dilek/D2K Covid-19")
cases <- read.csv("Confirmed Cases.csv")
deaths <- read.csv("Deaths.csv")

#Type in any state for subsetting, for example, Texas.
# cases <- subset(cases,Province_State=="Texas",select=X3.8.2020:X4.8.2020)
# deaths <- subset(deaths,Province_State=="Texas",select=X3.8.2020:X4.8.2020)

n <- nrow(cases)
cases_v_deaths_lag <- matrix(NA,nrow=n,ncol=2)
for (i in 1:n){
  #First check if the county has any deaths
  if (deaths[i,32] != 0){
    #Cases
    county_cases <- data.frame(per_county(i,cases))
    info_case <- summary(glm(county_cases$numbers/max(county_cases$numbers) ~ county_cases$days,
family=quasibinomial))
    b0_case <- coef(info_case)[1]
    b1_case <- coef(info_case)[2]
    #Deaths
    county_deaths <- data.frame(per_county(i,deaths))
    info_death <- summary(glm(county_deaths$numbers/max(county_deaths$numbers) ~ county_deaths$d
ays,family=quasibinomial))
    b0_death <- coef(info_death)[1]
    b1_death <- coef(info_death)[2]
    if (abs(b0_case) < 300 & abs(b0_death) < 300){ #checking if glm converged
      #Difference in values for b0
      cases_v_deaths_lag[i,1] <- abs(abs(b0_death)-abs(b0_case))
      #Difference in days for when 50% of cases/ deaths reached
      cases_v_deaths_lag[i,2] <- ((log(1)-b0_death)/b1_death)-((log(1)-b1_case)/b0_case)
    }
    else{
      #Not enough cases/ deaths to model
      cases_v_deaths_lag[i,] <- 0
    }
  }
  else{
    #There aren't any deaths, so there's no metric for lag
    cases_v_deaths_lag[i,] <- 0
  }
}
```

Another potential metric to examine could be calculating the difference in days between when the maximum of the peaks are reached (i.e. through the Newton-Raphson method or some other optimization)

There are many factors that go into differences between the US, UK, Spain, and Italy, but we thought it might be interesting to use this lag metric calculation on their data. We assumed that the start-date of the disease would be the presence of the very first case and then looked at how the cases and deaths increased since then. Data is from: https://www.kaggle.com/imdevskp/corona-virus-report (https://www.kaggle.com/imdevskp/corona-virus-report)

```
country_data <- read.csv("Country_Data.csv")
UK <- subset(country_data,Country.Region == "United Kingdom" & Province.State=="")
uk_days <- c(1:nrow(UK))
uk_cases <- UK$Confirmed
uk_deaths <- UK$Deaths
Italy <- subset(country_data,Country.Region == "Italy" & Province.State=="")
ital_days <- c(1:nrow(Italy))
ital_cases <- Italy$Confirmed
ital_deaths <- Italy$Deaths
US <- subset(country_data,Country.Region == "US" & Province.State=="")
us_days <- c(1:nrow(US))
us_cases <- US$Confirmed
us_deaths <- US$Deaths
Spain <- subset(country_data,Country.Region =="Spain" & Province.State == "")
spain_days <- c(1:nrow(Spain))
spain_cases <- Spain$Confirmed
spain_deaths <- Spain$Deaths
France <- subset(country_data,Country.Region == "France" & Province.State == "")
france_days <- c(1:nrow(France))
france_cases <- France$Confirmed
france_deaths <- France$Deaths
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
plot <- function(days,in_data,y,title){
  dat <- data.frame(list("days"=days,"numbers"=in_data))
  plot <- ggplot(dat,aes(x=days,y=numbers/max(numbers)))+geom_point()+geom_smooth(method="glm",m
ethod.args=list(family="quasibinomial"))+labs(title=title,y=y,x="Days Since January 22nd")
  plot
}
#plot(uk_days,uk_deaths,"Deaths","UK")
#plot(uk_days,uk_cases,"Cases","UK")
#plot(us_days,us_deaths,"Deaths","US")
#plot(us_days,us_cases,"Cases","US")
#plot(spain_days,spain_deaths,"Deaths","Spain")
#plot(spain_days,spain_cases,"Cases","Spain")
#plot(ital_days,ital_deaths,"Deaths","Italy")
#plot(ital_days,ital_cases,"Cases","Italy")
#plot(france_days,france_cases,"Cases","France")
#plot(france_days,france_deaths,"Deaths","France")
```

I’m only going to look at the 50% metric because this seems to be the most reliable as it relies on both parameters of the logistic regression as opposed to focusing on the intercept parameter.

```
metric<- function(days,cases,deaths){
  dat1 <- data.frame(list("days"=days,"numbers"=cases))
  info_case <- summary(glm(dat1$numbers/max(dat1$numbers) ~ dat1$days,family="quasibinomial"))
  b0_case <- coef(info_case)[1]
  b1_case <- coef(info_case)[2]
  dat2 <- data.frame(list("days"=days,"numbers"=deaths))
  info_death <- summary(glm(dat2$numbers/max(dat2$numbers) ~ dat2$days,family="quasibinomial"))
  b0_death <- coef(info_death)[1]
  b1_death <- coef(info_death)[2]
  return(((log(1)-b0_death)/b1_death)-((log(1)-b1_case)/b0_case))
}
metric(ital_days,ital_cases,ital_deaths)
```

```
## [1] 66.78364
```

```
metric(spain_days,spain_cases,spain_deaths)
```

```
## [1] 70.02082
```

```
metric(france_days,france_cases,france_deaths)
```

```
## [1] 73.27836
```

```
metric(uk_days,uk_cases,uk_deaths)
```

```
## [1] 74.98183
```

```
metric(us_days,us_cases,us_deaths)
```

```
## [1] 75.12094
```

Important to note that these differences could be attributed to so many things; for example, a better medical system is going to prolong the time until death. I'm not sure why these numbers are so large - this is a different dataset, so perhaps the way that the cases/ deaths are numbered is different. This shouldn't matter, however, as long as the relationship between cases and deaths is the same.

Appendix

```
setwd("C:/Users/Dilek/D2K Covid-19")
cases <- read.csv("Confirmed Cases.csv")
deaths <- read.csv("Deaths.csv")
per_county <- function(n,data){
  # n is n'th row of the data that corresponds to the n'th county
  # Data contains either number of cases or deaths per day
  # Output is a data frame with the number of days since March 8th 2020 and the number of casse
 for each of those time frames
  a <- c()
  b <- c()
  count = 1
  for (num in data[n,]){
    a <- append(a, count)
    b <- append(b, num)
    count = count + 1
  }
  return(list("days"=a,"numbers"=b))
}
setwd("C:/Users/Dilek/D2K Covid-19")
cases <- read.csv("Confirmed Cases.csv")
deaths <- read.csv("Deaths.csv")

#Type in any state for subsetting, for example, Texas.
# cases <- subset(cases,Province_State=="Texas",select=X3.8.2020:X4.8.2020)
# deaths <- subset(deaths,Province_State=="Texas",select=X3.8.2020:X4.8.2020)

n <- nrow(cases)
cases_v_deaths_lag <- matrix(NA,nrow=n,ncol=2)
for (i in 1:n){
  #First check if the county has any deaths
  if (deaths[i,32] != 0){
    #Cases
    county_cases <- data.frame(per_county(i,cases))
    info_case <- summary(glm(county_cases$numbers/max(county_cases$numbers) ~ county_cases$days,
family=quasibinomial))
    b0_case <- coef(info_case)[1]
    b1_case <- coef(info_case)[2]
    #Deaths
    county_deaths <- data.frame(per_county(i,deaths))
    info_death <- summary(glm(county_deaths$numbers/max(county_deaths$numbers) ~ county_deaths$d
ays,family=quasibinomial))
    b0_death <- coef(info_death)[1]
    b1_death <- coef(info_death)[2]
    if (abs(b0_case) < 300 & abs(b0_death) < 300){ #checking if glm converged
      #Difference in values for b0
      cases_v_deaths_lag[i,1] <- abs(abs(b0_death)-abs(b0_case))
      #Difference in days for when 50% of cases/ deaths reached
      cases_v_deaths_lag[i,2] <- ((log(1)-b0_death)/b1_death)-((log(1)-b1_case)/b0_case)
    }
    else{
      #Not enough cases/ deaths to model
      cases_v_deaths_lag[i,] <- 0
    }
```

```
  }
  else{
    #There aren't any deaths, so there's no metric for lag
    cases_v_deaths_lag[i,] <- 0
  }
}
country_data <- read.csv("Country_Data.csv")
UK <- subset(country_data,Country.Region == "United Kingdom" & Province.State=="")
uk_days <- c(1:nrow(UK))
uk_cases <- UK$Confirmed
uk_deaths <- UK$Deaths
Italy <- subset(country_data,Country.Region == "Italy" & Province.State=="")
ital_days <- c(1:nrow(Italy))
ital_cases <- Italy$Confirmed
ital_deaths <- Italy$Deaths
US <- subset(country_data,Country.Region == "US" & Province.State=="")
us_days <- c(1:nrow(US))
us_cases <- US$Confirmed
us_deaths <- US$Deaths
Spain <- subset(country_data,Country.Region =="Spain" & Province.State == "")
spain_days <- c(1:nrow(Spain))
spain_cases <- Spain$Confirmed
spain_deaths <- Spain$Deaths
France <- subset(country_data,Country.Region == "France" & Province.State == "")
france_days <- c(1:nrow(France))
france_cases <- France$Confirmed
france_deaths <- France$Deaths
library(ggplot2)
plot <- function(days,in_data,y,title){
  dat <- data.frame(list("days"=days,"numbers"=in_data))
  plot <- ggplot(dat,aes(x=days,y=numbers/max(numbers)))+geom_point()+geom_smooth(method="glm",m
ethod.args=list(family="quasibinomial"))+labs(title=title,y=y,x="Days Since January 22nd")
  plot
}
#plot(uk_days,uk_deaths,"Deaths","UK")
#plot(uk_days,uk_cases,"Cases","UK")
#plot(us_days,us_deaths,"Deaths","US")
#plot(us_days,us_cases,"Cases","US")
#plot(spain_days,spain_deaths,"Deaths","Spain")
#plot(spain_days,spain_cases,"Cases","Spain")
#plot(ital_days,ital_deaths,"Deaths","Italy")
#plot(ital_days,ital_cases,"Cases","Italy")
#plot(france_days,france_cases,"Cases","France")
#plot(france_days,france_deaths,"Deaths","France")
metric<- function(days,cases,deaths){
  dat1 <- data.frame(list("days"=days,"numbers"=cases))
  info_case <- summary(glm(dat1$numbers/max(dat1$numbers) ~ dat1$days,family="quasibinomial"))
  b0_case <- coef(info_case)[1]
  b1_case <- coef(info_case)[2]
  dat2 <- data.frame(list("days"=days,"numbers"=deaths))
  info_death <- summary(glm(dat2$numbers/max(dat2$numbers) ~ dat2$days,family="quasibinomial"))
  b0_death <- coef(info_death)[1]
  b1_death <- coef(info_death)[2]
  return(((log(1)-b0_death)/b1_death)-((log(1)-b1_case)/b0_case))
```

```
}
metric(ital_days,ital_cases,ital_deaths)
metric(spain_days,spain_cases,spain_deaths)
metric(france_days,france_cases,france_deaths)
metric(uk_days,uk_cases,uk_deaths)
metric(us_days,us_cases,us_deaths)
```