Technical Appendix

Albert Nyarko-Agyei, Charles Ayson-Parrish, Mai-An Dang, Tim Leeman, Zhen Heng Low

Setup and Data

If you would like to run the code in this document yourself, you will need to install the following libraries: `devtools, tidyverse, dplyr, remotes, lubridate, readxl, rio, tidycovid19, ggpubr, zoo, MASS, splines, leaps, gridExtra, caret`. We have a script to do this automatically, load the package `devtools` and then run `source_url("https://raw.github.com/timleeman/ST344GroupProject/main/setup.R")` in your R console.

```
library(devtools)
#loads packages
source_url("https://raw.github.com/timleeman/ST344GroupProject/main/libraries.R")
#gets data
source_url("https://raw.github.com/timleeman/ST344GroupProject/main/getdata.R")
```

Explaining the setup scripts

The `getdata.R` script downloads the data we are using from our shared github repository, as well as produces a list of countries which we have chosen to analyse:

```
library(readr)
library(devtools)
#Reads the tidied version of the AuraVision dataset for lockdown dates
auravisionData <- read_csv(
"https://raw.github.com/timleeman/ST344GroupProject/main/Data/AuraVisionCleaned.csv")
#Imports a tibble which lists which continent of all of the countries
countries <- read_csv(
"https://raw.github.com/timleeman/ST344GroupProject/main/Data/Continents.csv")
#Imports a snapshot of the merged tidycovid19 data set created on 19/11/2020
covidData <- read_csv(
"https://raw.github.com/timleeman/ST344GroupProject/main/Data/TidyCovid19DataSet.csv",
  guess_max = 10000)

#Next section adds continent to the covidData dataset
countries <- filter(countries, country %in% covidData$country)
covidData <- left_join(covidData, countries, by = "country")
covidData <- mutate(covidData, country = factor(country), continent = factor(continent))
covidData$country <- as.factor(covidData$country)
#Getting just the relevant continents
covidData <- filter(covidData, continent == "Europe" |
 continent == "North America" | continent == "South America" | country == "Turkey")
#These loops get rid of countries which have more than a 1/3 of their GCMR data missing
co <- ""
#Remove countries with excessive amounts of data missing in gcmr_retail_recreation column
for (co in levels(covidData$country)){
  #100 can be changed to another threshold
```

```
  if(sum(is.na(filter(covidData, country == co)$gcmr_retail_recreation)) >= 100){
    covidData <- filter(covidData, country != co)
  }
}

#Remove countries with excessive amounts of data missing in gcmr_grocery_pharmacy column
for (co in levels(covidData$country)){
  #100 can be changed to another threshold
  if(sum(is.na(filter(covidData, country == co)$gcmr_grocery_pharmacy)) >= 100){
    covidData <- filter(covidData, country != co)
  }
}
#Getting rid of countries with populations less than 1,000,000
covidData <- filter(covidData, population > 1000000)
covidData <- droplevels(covidData)
chosenCountries <- levels(covidData$country)
covidData <- read_csv(
"https://raw.github.com/timleeman/ST344GroupProject/main/Data/TidyCovid19DataSet.csv",
guess_max = 10000)
countries <- filter(countries, country %in% covidData$country)
covidData <- left_join(covidData, countries, by = "country")
covidData <- mutate(covidData, country = factor(country), continent = factor(continent))
```

Cleaning the Auravision Dataset

We rely on a data set from AuraVision which includes start and end dates of various types of lockdowns across countries for much of our analysis. The data set was originally messy and what follows is the procedure for tidying it.

First we get the original version of the AuraVision Dataset.

```
auravisionData <- read_csv(
"https://raw.github.com/timleeman/ST344GroupProject/main/LockdownDates/AuravisionLockdownDates.csv")
```

Then we filter on our chosen countries, as defined by the getdata.R setup script.

```
auravisionData <- auravisionData %>% filter(Country %in% chosenCountries)
```

Cleaning our dataset

We rename our columns using CamelBack font type, with the first character capital. We did this to remove all spaces within variable names.

```
auravisionData <- rename(auravisionData, StartDate = 'Start date', EndDate = 'End date')
```

We do not make much use of non-national (e.g. city-wide or regional) lockdowns for our investigation. Therefore, we focussed our efforts on cleaning the rows of our dataset that pertain to national lockdowns.

Here, we check for countries with National lockdowns that have a missing start or end date.

```
auravisionData %>% filter(Level=='National' & (is.na(StartDate) | is.na(EndDate)))
```

After doing some (brief) research online, it seems that Panama is still in lockdown. Therefore, we can accept this missing end date; it is the result of an unknown end date, and not a missing data value.

Then, we check for countries with multiple National lockdowns reported in the dataset.

```
auravisionData %>%
  filter(Level=='National') %>%
  group_by(Country) %>%
  filter(n()>1) %>%
  arrange(Country, StartDate)
```

Honduras

The Honduras had a slightly staggered lockdown. Four municipalities implemented a lockdown and then within less than a week the rest of the country also went into lockdown. We shall be taking using the row in our dataset that refers to the country as a whole (and not to Place==“Rest of area”)

Mexico

After doing some (brief) research online, it seems that Mexico City remained in lockdown after the end of the national lockdown. The date at which the majority of the country's lockdown ended was 2020-06-01. Therefore we choose this as our end date.

Turkey

The source for Turkey's 'second' national lockdown (i.e. Place==“Turkey(second implementation)”) is no longer available. As it only spans four days, this may not have been a significant impact on the variables we are observing, therefore we shall ignore this row in our dataset. Turkey's 'first' national lockdown (i.e. Place==“Turkey(first implementation)”) also only spans two days. Therefore, for the same reasons, we dismiss this row and only observe the row where the variable 'Place' is missing.

```
# These are the rows for national lockdowns that we want,
# for our three countries with multiple national lockdowns.
auravisionData %>%
  filter((Country=='Honduras' & is.na(Place)) |
           (Country=='Mexico' & !is.na(Place)) |
           (Country=='Turkey' & is.na(Place)))

auravisionData <- subset(auravisionData,
        # Take any rows if not in our three countries.
       (!Country %in% c("Honduras", "Mexico", "Turkey")) |
         # Take any rows if not related to a national lockdown.
         Level != 'National' |
         (Country=='Honduras' & is.na(Place)) |
         (Country=='Mexico' & !is.na(Place)) |
         (Country=='Turkey' & is.na(Place))) %>% arrange(Country)
```

Exporting .csv file

Finally, we export this as a .csv so that it can be easily used by the whole team.

```
write_csv(auravisionData, "AuraVisionCleaned.csv", na="")
```

This `.csv` of the tidied dataset was then uploaded to our github repository here

Analysis

Investigating effect of Lockdowns

Setting up our functions

```
firstdiff <- function(x) {
  shifted <- c(0,x[1:(length(x)-1)])
  result = x-shifted
  which_negative = which(result<0)
  result[which_negative] = NA
  return(result)
}
```

Getting the OWID Data Set

The OWID Data Set is used to obtain the Reproduction Rate for COVID-19 for each country on each given day,

```
OWIDdata<- read.csv('https://covid.ourworldindata.org/data/owid-covid-data.csv')
```

Setting up the tidycovid19 data frame

```
covidData1<- covidData %>% filter(country %in% chosenCountries)

covidData1<- covidData1%>%
  mutate(national_lock = ifelse(country %in%
      filter(auravisionData, Level == 'National')$Country, 1, 0))%>%
  mutate(city_lock = ifelse(country %in%
      filter(auravisionData, Level == 'City')$Country, 1, 0))%>%
  mutate(region_lock = ifelse(country
       %in% filter(auravisionData, Level %in%
       c('Prefecture','Province','State','Region','Regional'))$Country, 1, 0))%>%
  mutate(confirmed_per_capita = confirmed/population) %>%
  mutate(daily_confirmed = firstdiff(confirmed)) %>%
  mutate(daily_deaths = firstdiff(deaths)) %>%
  mutate(deaths_per_capita = deaths/population)%>%
  mutate(income = factor(income, levels = c("High income","Upper middle income",
                                      "Lower middle income", "Low income")))
```

Setting up the OWID data frame to be merged with the main data frame

```
OWIDdata1<- OWIDdata %>%
  rename(country = location) %>%
  filter(country %in% chosenCountries)%>%
  dplyr::select(date, country, reproduction_rate)

OWIDdata1 <- OWIDdata1 %>%
  mutate(date = as.Date(parse_date_time(OWIDdata1$date, orders = 'ymd')))

covidData1<- right_join(covidData1, OWIDdata1, by = c('country','date'))
```

Plot of effect of a national lockdown on level of activity at retail and recreation centres.

```
plot_1<- filter(covidData1, national_lock %in% 1) %>%
  ggplot(aes(x = date, y = gcmr_retail_recreation)) +
  geom_line(aes(y = rollmean(gcmr_retail_recreation, 7, na.pad = TRUE)),
  colour = 'red', alpha= 0.3)+
  geom_smooth(aes(colour = 'Coutries with National Lockdowns'),
  data = filter(covidData1, national_lock %in% 1), se = FALSE)+
  geom_line(aes(y = rollmean(gcmr_retail_recreation, 7, na.pad = TRUE)),
  colour = 'blue', alpha= 0.3, data = filter(covidData1, national_lock %in% 0))+
  geom_smooth(aes(colour = 'Countries without National Lockdowns'),
  data = filter(covidData1, national_lock %in% 0), se = FALSE)+
  scale_colour_manual(name = 'Legend', values =
  c('Coutries with National Lockdowns'='red',
    'Countries without National Lockdowns' = 'blue' ))+
  labs(title = 'Effects of a National Lockdown on Retail and Recreation',
       x= "Date", y = 'Activity in Retail and Recreation Centres') +
  scale_x_date(breaks = 'months', date_labels = '%b')+
  theme(legend.position = "bottom")
```

Plot of effect of a national lockdown on level of activity at groceries and pharmacies.

```
plot_2<- filter(covidData1, national_lock %in% 1) %>%
  ggplot(aes(x = date, y = gcmr_grocery_pharmacy)) +
  geom_line(aes(y = rollmean(gcmr_grocery_pharmacy, 7, na.pad = TRUE)),
  colour = 'red', alpha= 0.3)+
  geom_smooth(aes(colour = 'Coutries with National Lockdowns'),
  data = filter(covidData1, national_lock %in% 1), se = FALSE)+
  geom_line(aes(y = rollmean(gcmr_grocery_pharmacy, 7, na.pad = TRUE)),
  colour = 'blue', alpha= 0.3, data = filter(covidData1, national_lock %in% 0))+
  geom_smooth(aes(colour = 'Countries without National Lockdowns'),
  data = filter(covidData1, national_lock %in% 0), se = FALSE)+
  scale_colour_manual(name = 'Legend', values =
  c('Coutries with National Lockdowns'='red',
  'Countries without National Lockdowns' = 'blue' ))+
  labs(title = 'Effects of a National Lockdown on Groceries and Pharmacies',
  x= "Date", y = 'Activity in Groceries and Pharmacies')+
  scale_x_date(breaks = 'months', date_labels = '%b')+
  theme(legend.position = "bottom")
```

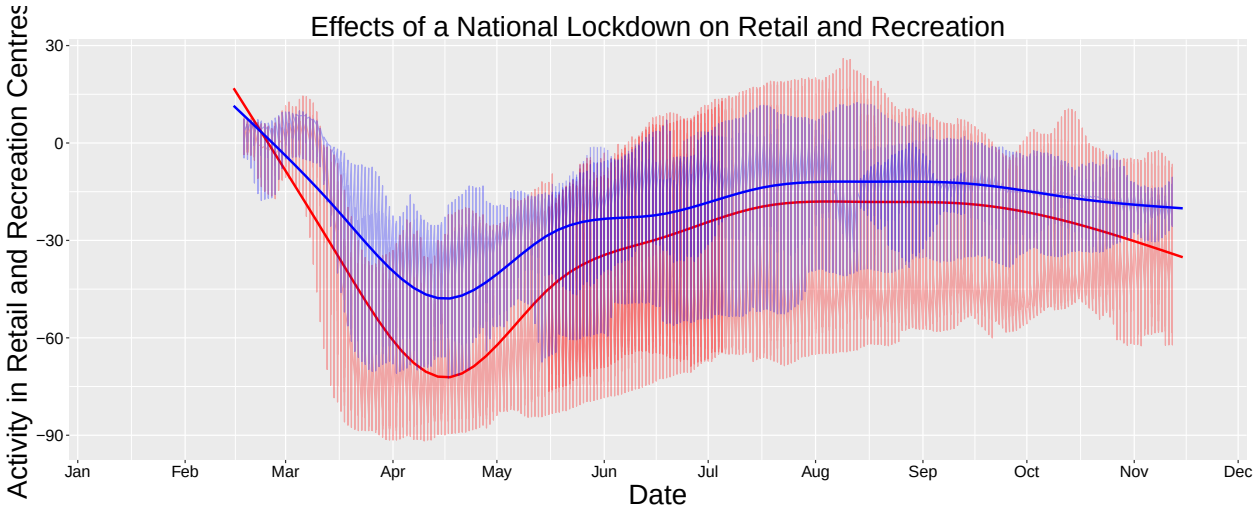
Plot of effect of a national lockdown on daily confirmed cases.

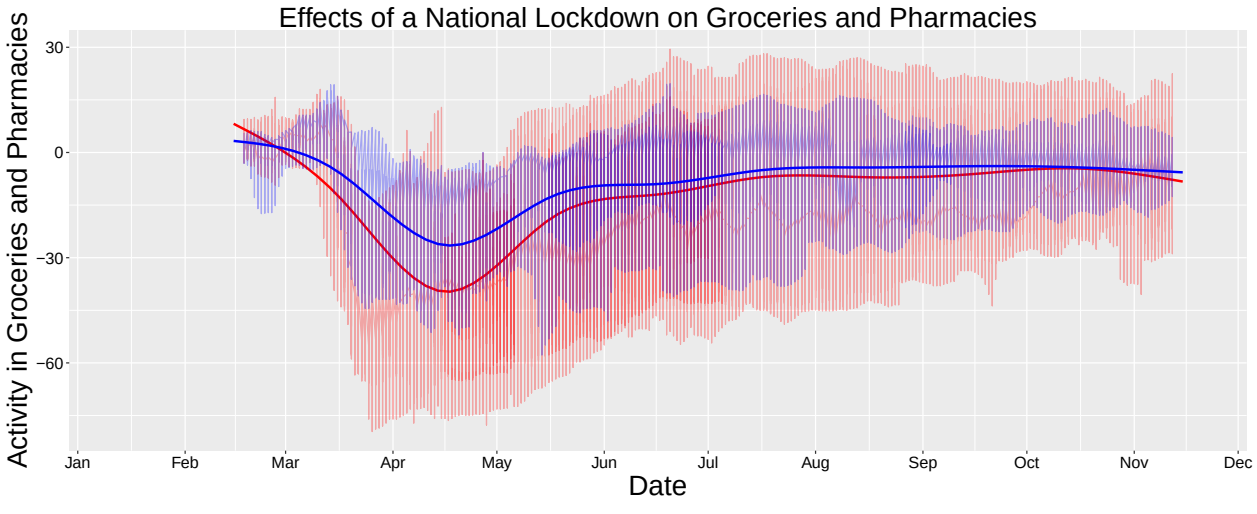
```
plot_3<- filter(covidData1, national_lock %in% 1) %>%
  ggplot(aes(x = date, y = daily_confirmed)) +
  geom_line(aes(y = rollmean(daily_confirmed, 7, na.pad = TRUE)),
  colour = 'red', alpha= 0.3)+
  geom_smooth(aes(colour = 'Coutries with National Lockdowns'),
  data = filter(covidData1, national_lock %in% 1), se = FALSE)+
  geom_line(aes(y = rollmean(daily_confirmed, 7, na.pad = TRUE)),
  colour = 'blue', alpha= 0.3, data = filter(covidData1, national_lock %in% 0))+
  geom_smooth(aes(colour = 'Countries without National Lockdowns'),
  data = filter(covidData1, national_lock %in% 0), se = FALSE)+
  scale_colour_manual(name = 'Legend',
```

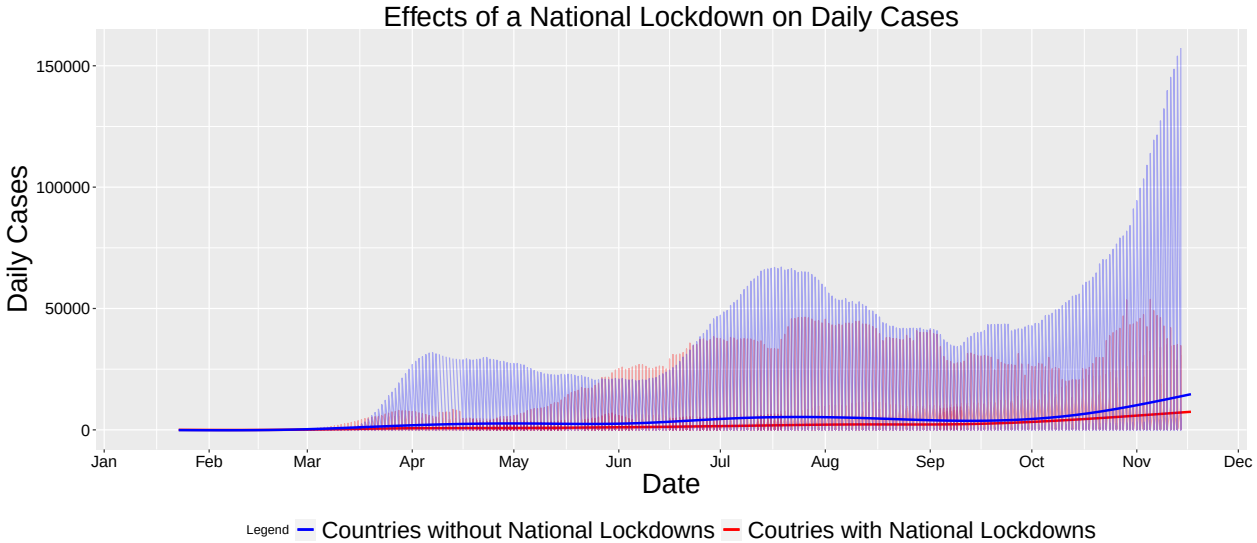
```
  values = c('Coutries with National Lockdowns'='red',
  'Countries without National Lockdowns' = 'blue' ))+
  labs(title = 'Effects of a National Lockdown on Daily Cases',
  x= "Date", y = 'Daily Cases')+
  scale_x_date(breaks = 'months', date_labels = '%b')+
  theme(legend.position = "bottom")
```

Final plots

```
plot_final<- ggarrange(plot_1, plot_2, plot_3, ncol = 1, nrow = 3)
plot_final
```







From the 2 plots above, we can see that with the presence of a national lockdown on our list of chosen

countries, the level of activity in gcmr_retail_recreation and gcmr_grocery_pharmacy have decreased dramatically compared to countries without a lockdown. This suggests that a national lockdown does hamper economic activity and thus affects the businesses by decreasing the number of customers.

Setting up step-wise regression analysis for gcmr_retail_recreation

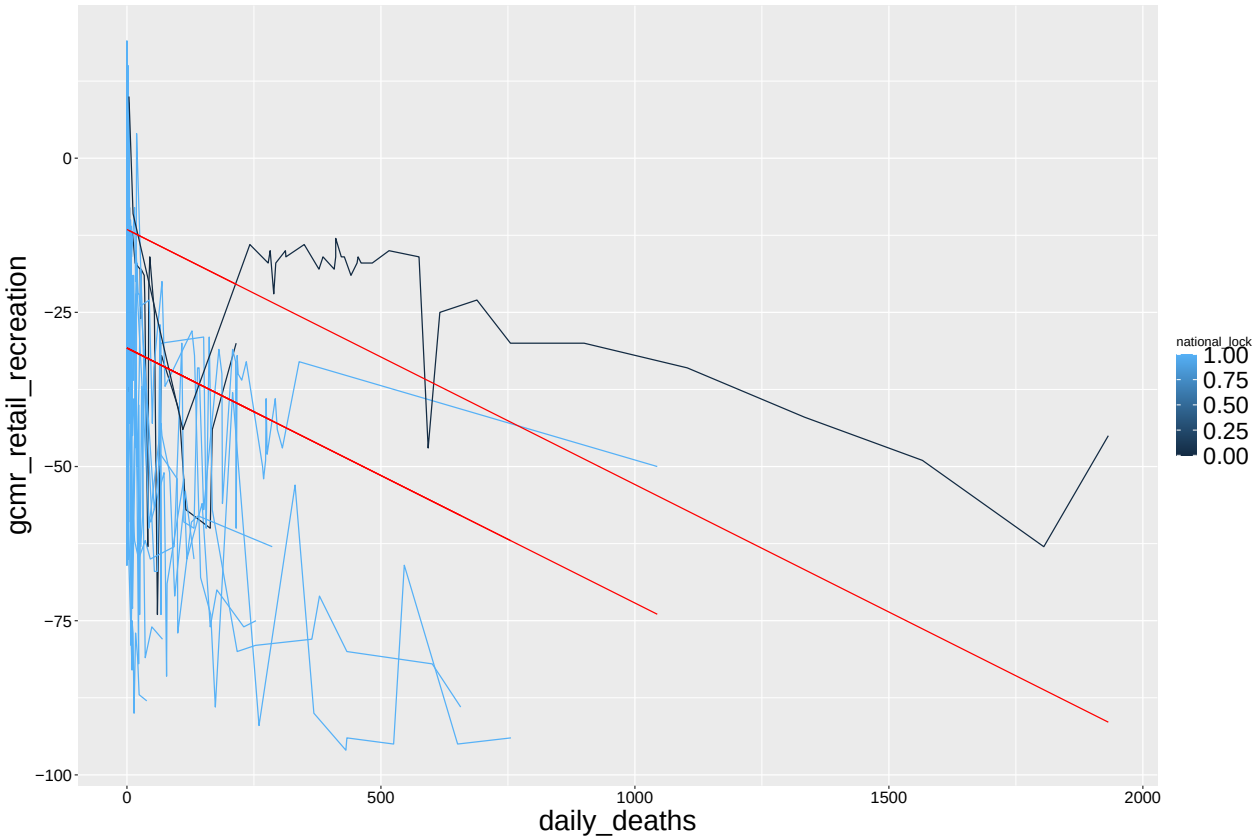
```
minModel <- glm(data = na.omit(covidData1),
                gcmr_retail_recreation ~ 1)
maxModel <- glm(data = na.omit(covidData1),
                gcmr_retail_recreation ~national_lock + confirmed_per_capita +
                  daily_deaths + deaths_per_capita + daily_confirmed)
autoBack <- step(maxModel, direction = "backward",
                 scope = list("lower" = minModel), trace = FALSE)
autoForward <- step(minModel, direction = "forward",
                    scope = list("upper" = maxModel), trace = FALSE)
autoBoth <- step(minModel, direction = "both",
                 scope = list("lower" = minModel, "upper" = maxModel), trace = FALSE)
```

```
model_1<- glm(data = na.omit(covidData1),
        gcmr_retail_recreation ~ national_lock + daily_deaths)
```

Finalised model for Model 1

```
na.omit(covidData1) %>%
ggplot(aes(x = daily_deaths, y = gcmr_retail_recreation,
       group = country, colour = national_lock))+
       geom_line()+
       geom_line(aes(y = predict(model_1)), colour = 'red')
```

Checking fit of Model 1



From the step-wise regression, we concluded that the most suitable model to explain the changes in gcmr_retail_recreation is shown as:

$$gcmr\_retail\_recreation = \beta + \alpha_1(national\_lock) + \alpha_2(daily\_deaths)$$

with values given by

$$\beta = -11.56, \alpha_1 = -19.22, \alpha_2 = -0.0414$$

Analysis of the Normal QQ plot shows that the residuals and the fitted values approximate a normal distribution, with most of the points falling on the line. A quick plot also shows that the model is an approximate fit to the points. This is in line with our initial findings where the presence of a national lockdown decreasing customer actvity in retail and recreation.

Setting up step-wise regression analysis for gcmr_grocery_pharmacy

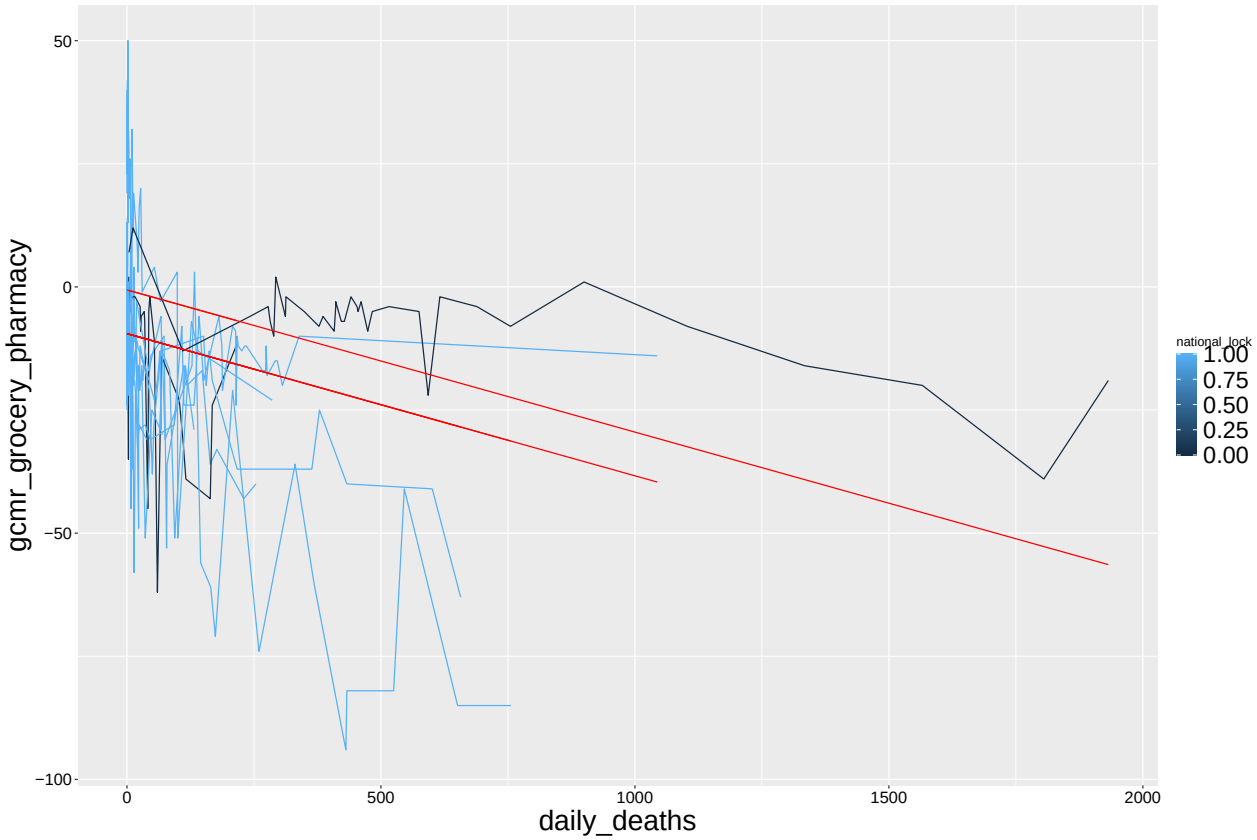
```
minModel <- glm(data = na.omit(covidData1),
                gcmr_grocery_pharmacy ~ 1)
maxModel <- glm(data = na.omit(covidData1),
                gcmr_grocery_pharmacy ~national_lock + confirmed_per_capita +
                  daily_deaths + deaths_per_capita + daily_confirmed)
autoBack <- step(maxModel, direction = "backward",
                 scope = list("lower" = minModel), trace = FALSE)
autoForward <- step(minModel, direction = "forward",
                    scope = list("upper" = maxModel), trace = FALSE)
autoBoth <- step(minModel, direction = "both",
                 scope = list("lower" = minModel, "upper" = maxModel), trace = FALSE)
```

```
model_2<- glm(data = na.omit(covidData1),
              gcmr_grocery_pharmacy~national_lock + daily_deaths)
```

Finalised model for Model 2

```
na.omit(covidData1) %>%
  ggplot(aes(x = daily_deaths, y = gcmr_grocery_pharmacy,
             group = country, colour = national_lock))+
  geom_line()+
  geom_line(aes(y = predict(model_2)), colour = 'red')
```

Checking fit of Model 2



From the step-wise regression, we concluded that the most suitable model to explain the changes in gcmr_grocery_pharmacy is shown as:

$$gcmr\_grocery\_pharmacy = \beta + \alpha_1(national\_lock) + \alpha_2(daily\_deaths) + \alpha_3(deaths\_per\_capita)$$

with values given by

$$\beta = 0.412, \alpha_1 = -8.62, \alpha_2 = -0.0286, \alpha_3 = -3715$$

Analysis of the Normal QQ plot shows that the residuals and the fitted values approximate a normal distribution, with most of the points falling on the line. A quick plot also shows that the model is an

approximate fit to the points. The findings here show that activity in groceries and pharmacies also respond negatively towards the presence of a national lockdown. However, deaths per capita seems to be the main driver towards the decrease of activity in groceries and pharmacies.

Setting up step-wise regression analysis for daily_confirmed

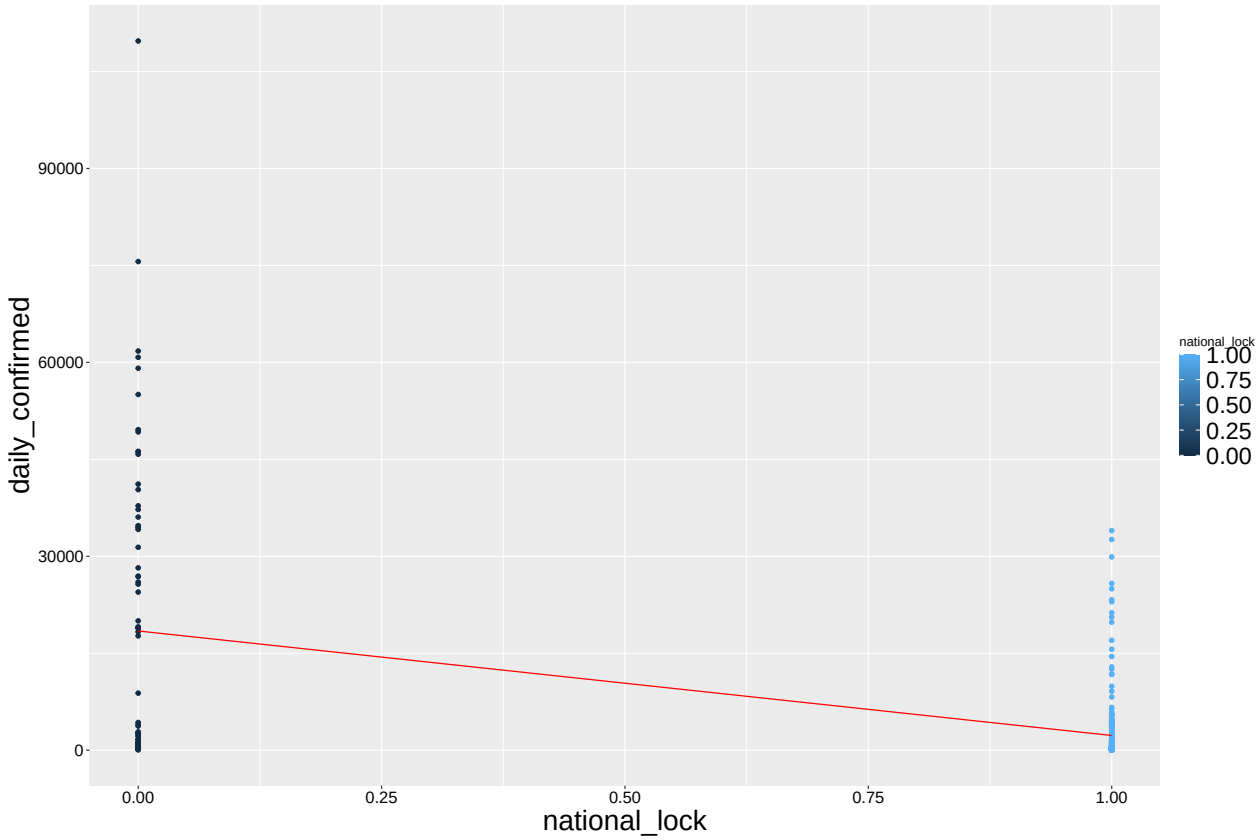
```
minModel <- glm(data = na.omit(covidData1),
                daily_confirmed ~ 1)
maxModel <- glm(data = na.omit(covidData1),
                daily_confirmed ~national_lock + reproduction_rate)
autoBack <- step(maxModel, direction = "backward",
                 scope = list("lower" = minModel), trace = FALSE)
autoForward <- step(minModel, direction = "forward",
                    scope = list("upper" = maxModel), trace = FALSE)
autoBoth <- step(minModel, direction = "both",
                 scope = list("lower" = minModel, "upper" = maxModel), trace = FALSE)
```

```
model_3<- glm(data = na.omit(covidData1), daily_confirmed ~ national_lock)
```

Finalised model for Model 3

```
na.omit(covidData1) %>%
  ggplot(aes(x = national_lock, y = daily_confirmed,
             colour = national_lock))+
  geom_point()+
  geom_line(aes(y = predict(model_3)), colour = 'red')
```

Checking fit of Model 3



From the step-wise regression, we conclude that the most suitable model to explain the changes in daily_confirmed is shown as:

$$daily\_confirmed = \beta + \alpha_1(national\_lock)$$

with values given by

$$\beta = 18449, \alpha_1 = -16168$$

Analysis of the Normal QQ plot shows that the residuals and the fitted values approximate a normal distribution, with most of the points falling on the line. A quick plot also shows that the model is an approximate fit to the points. From the step-wise regression, the reproduction rate seems to have little to no effect on the changes in daily confirmed cases, with a large p-value which suggests that it is insignificant to be included in our model on daily confirmed cases.

Effect of Lockdown on Covid Cases

The data for each country has been divided into their pre, intra and post lockdown dates. This used the functions `pre_lockdown_start date`, `lockdown_end_date` and `get_lockdown_data` and then each individual country's data was binded together. This was done using the Auravision data and then a rolling mean was calculated using the 3days before and after a specific day. This method means that although we were looking at data from a week before, the mean could only be calculated from four days before the start of lockdown and stopped three days before the end.

Getting the length of each country's lockdown from Auravision

These series of function pull the relevant dates from Auravision and countries from covidData. In order to compare countries, we aligned each country's data to the start of lockdown by creating the variable

days_in_lockdown. We also agreed to use a rolling mean for gcmr to reduce the effect of daily variations. The mean is calculated by averaging the gcmr of the three days before and after a specific day.

```
covidData<- covidData %>%
  mutate(daily_confirmed = firstdiff(confirmed))
selectData <- filter(covidData, country %in% chosenCountries)
pre_lockdown_start_date <- function(cou){
  a <- national_lockdowns %>%
    filter(country == cou) %>% dplyr::select(`StartDate`)
  return(a$`StartDate`[1] - 7)
}
lockdown_start_date <- function(cou){
  a <- national_lockdowns %>%
    filter(country == cou) %>% dplyr::select(`StartDate`)
  return(a$`StartDate`[1])
}
lockdown_end_date <- function(cou){
  a <- national_lockdowns %>%
    filter(country == cou) %>% dplyr::select(`EndDate`)
  return(a$`EndDate`[1])
}
get_lockdown_data <- function(cou){
   if(is.na(lockdown_end_date(cou))){
    lockdown_data <- selectData %>% filter(country == cou) %>%
       filter(date >= pre_lockdown_start_date(cou)) %>%
       mutate(days_in_lockdown =
                as.numeric(-difftime(lockdown_start_date(cou), date)/86400),
       rolling_gcmr_retail = zoo::rollmean(gcmr_retail_recreation, k = 7, fill = NA),
       rolling_gcmr_grocery = zoo::rollmean(gcmr_grocery_pharmacy, k = 7, fill = NA),
        peak_daily_cases = max(daily_confirmed, na.rm= T))
    return(lockdown_data)
  }
  lockdown_data <- selectData %>% filter(country == cou) %>%
       filter(date >= pre_lockdown_start_date(cou)) %>%
       filter(date < lockdown_end_date(cou)) %>%
      mutate(days_in_lockdown =
               as.numeric(-difftime(lockdown_start_date(cou), date)/86400),
       rolling_gcmr_retail = zoo::rollmean(gcmr_retail_recreation, k = 7, fill = NA),
       rolling_gcmr_grocery = zoo::rollmean(gcmr_grocery_pharmacy, k = 7, fill = NA),
       peak_daily_cases = max(daily_confirmed, na.rm= T))
  return(lockdown_data)
}
get_peak_daily_cases_day <- function(cou){
  most_in_day <- get_lockdown_data(cou)$peak_daily_cases[1]
  most_day <- get_lockdown_data(cou) %>%
    filter(daily_confirmed == most_in_day)
  most_day_date <- most_day$date[1]
  return(most_day_date)
}
lockdown_data <- tibble()
for(cou in national_lockdowns$country){
  a <- get_lockdown_data(cou)
  lockdown_data <- rbind(a, lockdown_data)
}
```

We calculated the R number for three periods in each country's epidemic. The first function uses the 28 days leading up to lockdown, it returns NA for countries that had a maximum of 10 daily cases during the period because the high R's calculated were not justifiable

```
before_lockdown_r <- function(cou){
  startday <- lockdown_start_date(cou) - 28
  endday <- lockdown_start_date(cou)
  key_data <- selectData %>%
    dplyr::select(country, date, daily_confirmed) %>%
    filter(date >= startday & date < endday) %>%
    filter(country == cou) %>% na.omit()
  if(max(key_data$daily_confirmed) < 10){
    return(NA)
  }
  daily_cases_model <- glm.nb(daily_confirmed ~
                              1 + date, data = key_data)
  key_data$pred_daily_cases <- predict(daily_cases_model)
  R <-  exp(4*(coef(daily_cases_model)[2] %>% as.numeric))
  return(R)
}
```

The second R uses the 28 days from the day with the most cases in lockdown

```
during_lockdown_r <- function(cou){
  startday <- get_peak_daily_cases_day(cou)
  endday <- startday + 28
  key_data <- selectData %>%
    dplyr::select(country, date, daily_confirmed) %>%
    filter(date >= startday & date < endday) %>%
    filter(country == cou) %>% na.omit()
  daily_cases_model <- glm.nb(daily_confirmed ~
                              1 + date, data = key_data)
  key_data$pred_daily_cases <- predict(daily_cases_model)
  R <-  exp(4*(coef(daily_cases_model)[2] %>% as.numeric))
  return(R)
}
```

This uses the 7 days after lockdown. Some countries were still in lockdown at the end of the dataset so if a lockdown end date was not found from Auravision, we use the during lockdown functions result for continuity.

```
post_lockdown_r <- function(cou){
  if(is.na(lockdown_end_date(cou))){
    return(during_lockdown_r(cou))
  }
  startday <- lockdown_end_date(cou)
  endday <- startday + 7
  key_data <- selectData %>%
    dplyr::select(country, date, daily_confirmed) %>%
    filter(date > startday & date <= endday) %>%
    filter(country == cou) %>% na.omit()
  daily_cases_model <- glm.nb(daily_confirmed ~
                                1 + date, data = key_data)
  key_data$pred_daily_cases <- predict(daily_cases_model)
  R <-  exp(4*(coef(daily_cases_model)[2] %>% as.numeric))
  return(R)
}
```

```
lst <- list(country= c(), before_lockdown_r= c(),
            during_lockdown_r= c(), post_lockdown_r= c())
Rdata <- as.data.frame(lst)
```

Calculates the three R values

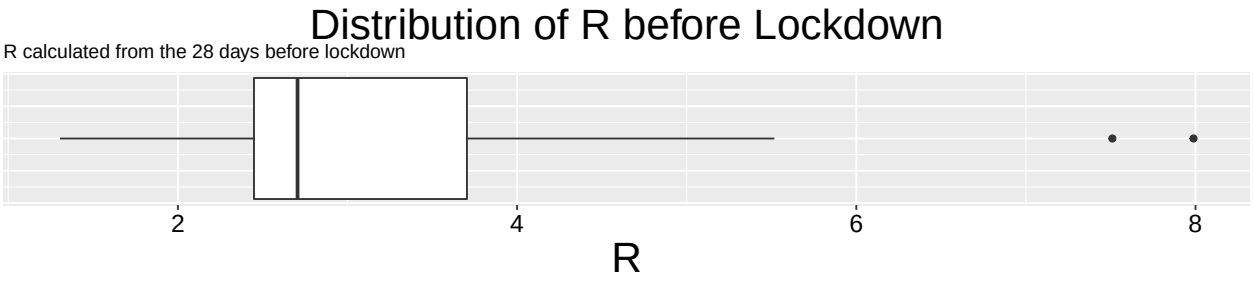
```
calculate_r <- function(x){
  a <- before_lockdown_r(x)
  b <- during_lockdown_r(x)
  c <- post_lockdown_r(x)
  row <- data.frame(country = x,
                    before_lockdown_r = a,
                    during_lockdown_r = b,
                    post_lockdown_r = c)
  return(row)
}
```

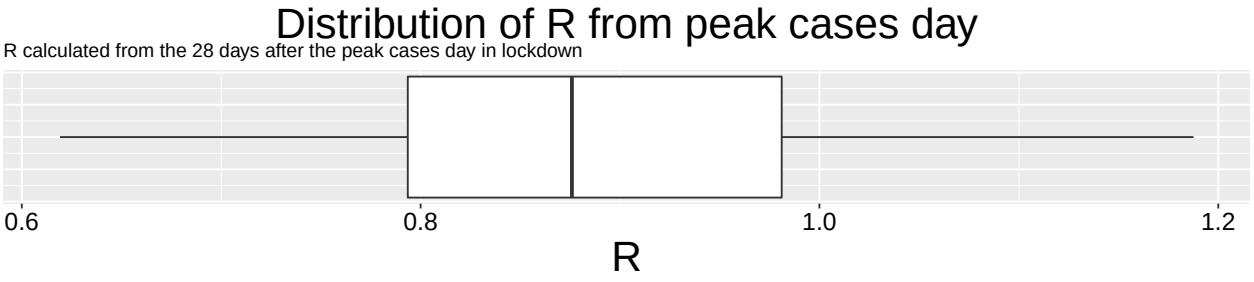
Binding the R data for each country in national lockdown

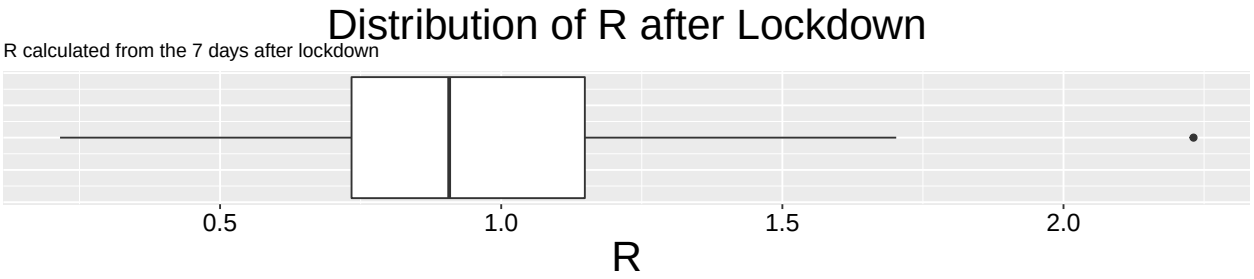
```
for(i in national_lockdowns$country){
  Rdata <- rbind(Rdata, calculate_r(i))
}
```

Exploring the R data

```
bp1 <- ggplot(Rdata, aes(x= before_lockdown_r)) + geom_boxplot() +
  labs(x= "R", title= "Distribution of R before Lockdown",
       subtitle= "R calculated from the 28 days before lockdown") +
       theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank())
bp2 <- ggplot(Rdata, aes(x= during_lockdown_r)) + geom_boxplot() +
  labs(x= "R", title= "Distribution of R from peak cases day",
       subtitle= "R calculated from the 28 days after the peak cases day in lockdown")+
       theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank())
bp3 <- ggplot(Rdata, aes(x= post_lockdown_r)) + geom_boxplot() +
  labs(x= "R", title= "Distribution of R after Lockdown",
       subtitle= "R calculated from the 7 days after lockdown") +
       theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank())
ggarrange(bp1, bp2, bp3, nrow = 3, ncol = 1)
```







The graphs above show that lockdown was effective in reducing the R in most countries as there is a clear shift to the left and then back to the right for the different periods across the three graphs. Even countries that had 'outlier' R numbers reduced their R significantly during lockdown. When deciding the period to calculate the R data from, we considered the trade-off between a large period but uncertainty on what the R reflected, the fact that linear models over long periods would not accurately predict the R because of fluctuations in daily confirmed cases. Also short periods might carry the risk of not being representative of the true R. In the end we decided to focus on the rising and falling part of each countries daily cases. For most countries there was an linear rise in the log of daily cases followed by a linear decrease in daily confirmed cases after the day with the most confirmed cases. After lockdown there was also a slight rise in R for some countries which intrinsically implies the damping effect of lockdown on R.

GCMR Modelling

In order to investigate the effect of length of lockdown, we use the data from AuraVision which includes the start and end dates of lockdowns to get a data set of countries which had a national lockdown and the length of those lockdowns. We once again choose the list of countries with sufficiently intact Google Mobility Trends data since this will be our main focus of investigation in examining the impact on relevant businesses.

```
national_lockdowns <- auravisionData %>%
  filter(Level== "National") %>% dplyr::select(-Level)
national_lockdowns <- national_lockdowns %>%
  mutate(national_lockdown_length =
           as.numeric(-difftime(`StartDate`, `EndDate`)))
colnames(national_lockdowns)[1] = "country"
national_lockdowns <- filter(national_lockdowns, country %in% chosenCountries)
head(national_lockdowns %>%
```

```
      dplyr::select(country, national_lockdown_length) %>%
      arrange(national_lockdown_length),60)
```

```
## # A tibble: 43 x 2
##    country  national_lockdown_length
##    <chr>                       <dbl>
##  1 Jamaica                        14
##  2 Turkey                         28
##  3 Austria                        29
##  4 Hungary                        37
##  5 Germany                        44
##  6 Norway                         46
##  7 Portugal                       46
##  8 Romania                        48
##  9 Slovakia                       51
## 10 Slovenia                       51
## # ... with 33 more rows
```

```
firstdiff <- function(x) {
  shifted <- c(0,x[1:(length(x)-1)])
  result = x-shifted
  which_negative = which(result<0)
  result[which_negative] = NA
  return(result) }
covidData <- covidData %>%
  mutate(daily_confirmed = firstdiff(confirmed))
```

Here we bind our data set which includes details of national lockdowns to our main data set which includes most of our data including `gcmr_retail_recreation` and `gcmr_grocery_pharmacy` which summarises changes in footfall in retail and grocery businesses respectively. We also create a rolling average of the gcmr data as the data tends to jump around due to lower activity on certain days (mostly on weekends, when certain stores are closed).

```
selectData <- filter(covidData, country %in% chosenCountries)
#Function to select country
pre_lockdown_start_date <- function(cou){
  a <- national_lockdowns %>% filter(country == cou) %>%
    dplyr::select(`StartDate`)
  return(a$`StartDate`[1] - 7)
}
lockdown_start_date <- function(cou){
  a <- national_lockdowns %>% filter(country == cou) %>%
    dplyr::select(`StartDate`)
  return(a$`StartDate`[1])
}
lockdown_end_date <- function(cou){
  a <- national_lockdowns %>% filter(country == cou) %>%
    dplyr::select(`EndDate`)
  return(a$`EndDate`[1])
}
get_lockdown_data <- function(cou){
  lockdown_data <- selectData %>% filter(country == cou, !is.na(daily_confirmed)) %>%
        filter(date >= pre_lockdown_start_date(cou)) %>%
        filter(date < lockdown_end_date(cou)) %>%
        mutate(days_in_lockdown =
```

```
                as.numeric(-difftime(lockdown_start_date(cou), date)/86400),
        rolling_gcmr_retail = zoo::rollmean(gcmr_retail_recreation, k = 7, fill = NA),
        rolling_gcmr_grocery = zoo::rollmean(gcmr_grocery_pharmacy, k = 7, fill = NA),
        rolling_daily_confirmed = zoo::rollmean(daily_confirmed, k = 7, fill = NA))
  return(lockdown_data)
}
lockdown_data <- tibble()
for(cou in national_lockdowns$country){
  a <- get_lockdown_data(cou)
  lockdown_data <- rbind(a, lockdown_data)
}
```

This section adds data from Gapminder and Statista that we suscepted would be relevant to gcmr models

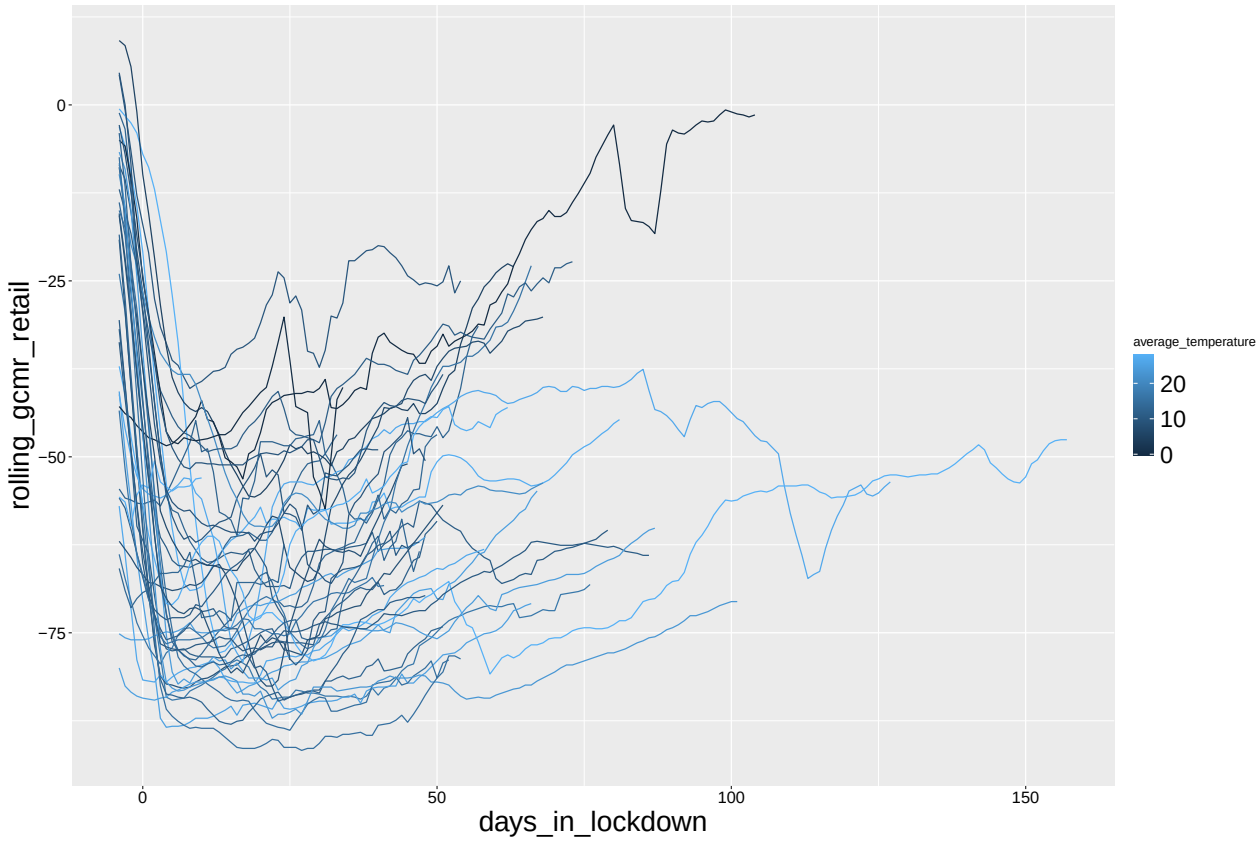
```
average_temperature <- import(
  "https://raw.github.com/timleeman/ST344GroupProject/main/Modelling/average_temperature_edit.xlsx")
modelData <- lockdown_data %>%
  dplyr::select(country, date, continent, gdp_capita, rolling_gcmr_retail,
    rolling_gcmr_grocery, gcmr_retail_recreation, gcmr_grocery_pharmacy,
    days_in_lockdown, income, daily_confirmed, rolling_daily_confirmed) %>%  distinct()
modelData <- left_join(modelData, average_temperature, by= "country")

modelData$gdp_capita <- scale(modelData$gdp_capita)
#modelData$average_temperature <- scale(modelData$average_temperature)
modelData <- modelData %>% na.omit()
modelData <- modelData %>% mutate(days_squared = days_in_lockdown^2)
population_20_39 <- import(
  "https://raw.github.com/timleeman/ST344GroupProject/main/Modelling/population_aged_20_39.xlsx")
population_20_39 <- dplyr::select(population_20_39, country, `2019`)
population_40_59 <- import(
  "https://raw.github.com/timleeman/ST344GroupProject/main/Modelling/population_aged_40_59.xlsx")
population_40_59 <- dplyr::select(population_40_59, country, `2019`)
population_20_59 <- left_join(population_20_39, population_40_59, by= "country")
population_20_59 <- population_20_59 %>%
  transmute(country, pop_20_59= `2019.x` + `2019.y` )
modelData <- left_join(modelData, population_20_59, by= "country")
head(modelData,500)
```

```
## # A tibble: 500 x 15
##    country date       continent gdp_capita[,1] rolling_gcmr_re~ rolling_gcmr_gr~
##    <chr>   <date>     <fct>              <dbl>            <dbl>            <dbl>
##  1 United~ 2020-03-20 Europe             0.914            -33.7             8
##  2 United~ 2020-03-21 Europe             0.914            -41.7             1.57
##  3 United~ 2020-03-22 Europe             0.914            -49.1            -5.29
##  4 United~ 2020-03-23 Europe             0.914            -56.7           -12.7
##  5 United~ 2020-03-24 Europe             0.914            -63.6           -18.3
##  6 United~ 2020-03-25 Europe             0.914            -68.3           -24
##  7 United~ 2020-03-26 Europe             0.914            -72.4           -28.6
##  8 United~ 2020-03-27 Europe             0.914            -76.6           -32
##  9 United~ 2020-03-28 Europe             0.914            -77             -32.9
## 10 United~ 2020-03-29 Europe             0.914            -77.3           -33.7
## # ... with 490 more rows, and 9 more variables: gcmr_retail_recreation <dbl>,
## #   gcmr_grocery_pharmacy <dbl>, days_in_lockdown <dbl>, income <chr>,
## #   daily_confirmed <dbl>, rolling_daily_confirmed <dbl>,
## #   average_temperature <dbl>, days_squared <dbl>, pop_20_59 <dbl>
```

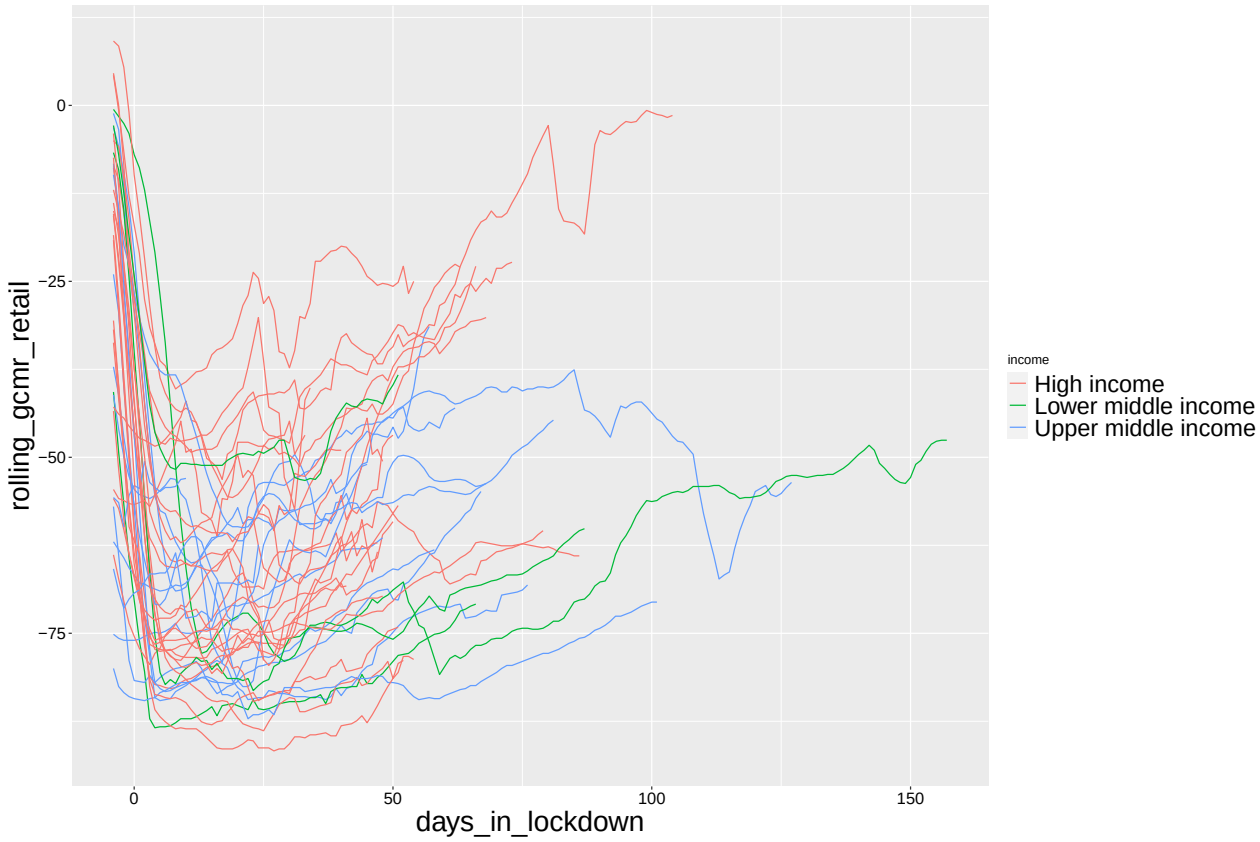
```
modelData <- left_join(modelData, national_lockdowns, by = "country")
modelData <- modelData %>%
  mutate(positive_rolling_gcmr_retail = -1*rolling_gcmr_retail,
         positive_rolling_gcmr_grocery = -1*rolling_gcmr_grocery)
```

View plots of gcmr against days in lockdown to assess the validating of a model.

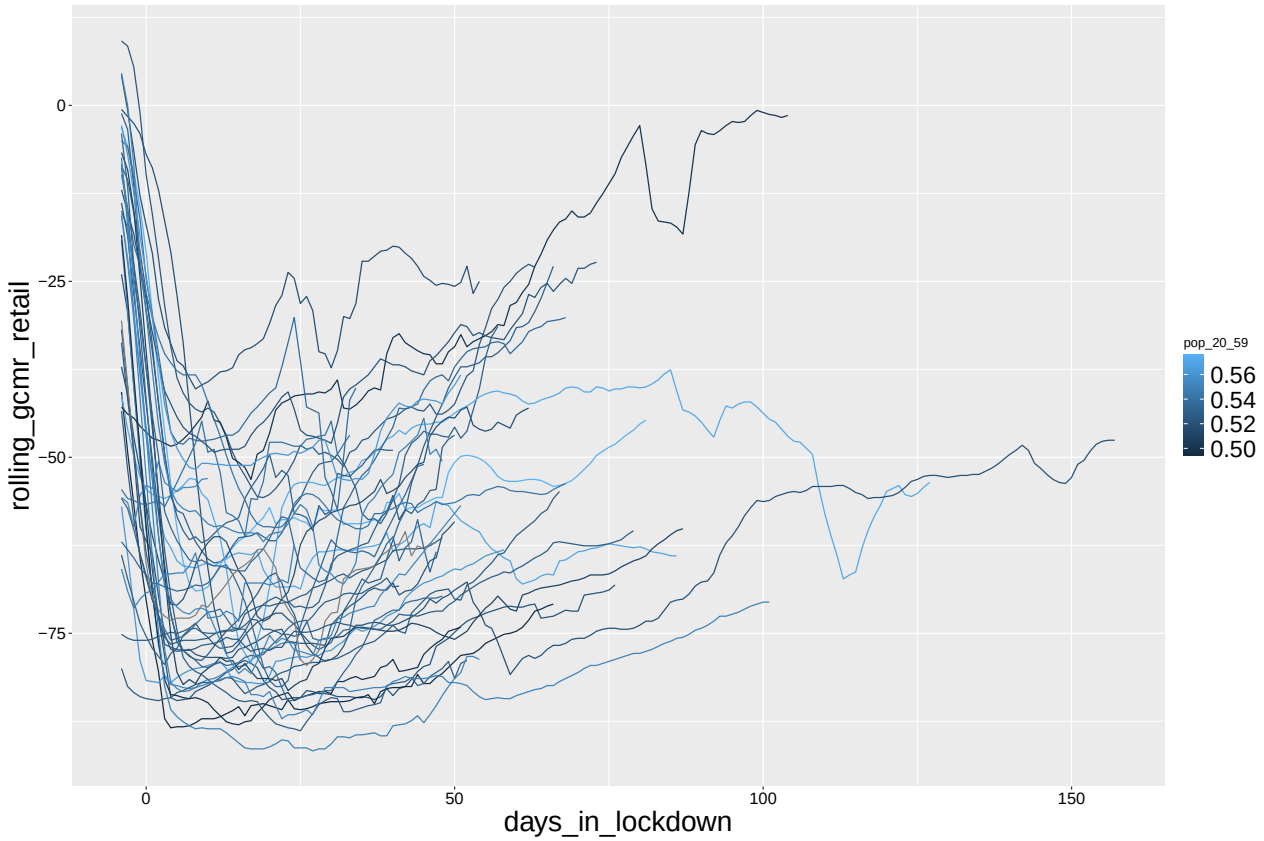
```
plot_country_lockdown_gcmr_retail <- function(cou){
  a <- modelData %>% filter(country == cou)
  ggplot(a, aes(x= date, y= rolling_gcmr_retail)) + geom_line()
}
#plot_country_lockdown_gcmr_retail("Germany")
ggplot(modelData, aes(x=days_in_lockdown, y= rolling_gcmr_retail,
                      group= country, col= average_temperature)) + geom_line()
```




```
ggplot(modelData, aes(x=days_in_lockdown, y= rolling_gcmr_retail,
                      group= country, col= income)) + geom_line()
```




```
ggplot(modelData, aes(x=days_in_lockdown, y= rolling_gcmr_retail,
                      group= country, col= pop_20_59)) + geom_line()
```



Here we fit a very basic linear model with the only predictor for the GCMR data being the length of national lockdown to see if we can learn anything about how lockdown lengths affect shopping habits.

```
initialModel1 <- glm(data = modelData, rolling_gcmr_retail ~ national_lockdown_length)
summary(initialModel1)
```

```
##
## Call:
## glm(formula = rolling_gcmr_retail ~ national_lockdown_length,
##     data = modelData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -30.338  -14.957   -3.186   11.689   69.961
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -64.97505    0.98098  -66.23  < 2e-16 ***
## national_lockdown_length   0.06205    0.01184    5.24 1.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 363.0489)
##
##     Null deviance: 965514  on 2633  degrees of freedom
## Residual deviance: 955545  on 2632  degrees of freedom
## AIC: 23005
##
```

```
## Number of Fisher Scoring iterations: 2
```

```
initialModel2 <- glm(data = modelData, rolling_gcmr_grocery ~ national_lockdown_length)
summary(initialModel2)
```

```
##
## Call:
## glm(formula = rolling_gcmr_grocery ~ national_lockdown_length,
##     data = modelData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -49.819  -14.553    1.177   12.870   45.495
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -26.01746    1.01710 -25.580  < 2e-16 ***
## national_lockdown_length  -0.05336    0.01228  -4.347 1.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 390.278)
##
##     Null deviance: 1034585  on 2633  degrees of freedom
## Residual deviance: 1027212  on 2632  degrees of freedom
## AIC: 23196
##
## Number of Fisher Scoring iterations: 2
```

Here we see that national_lockdown_length has a very low p-value which is good evidence for its usefulness in the model, but since its estimated coefficient is only 0.06256 we can see that while the length of a national lockdown does have an effect on how much people are visiting retail and grocery businesses, the effect appears to be small and inconsistent since we get a positive coefficient when modelling against data for retail and recreation but a negative coefficient when modelling against grocery and pharmacy. This would be interpreted as for two countries under the same conditions except for one had a longer lockdown by one day, we would predict that the one with a longer lockdown would see more people shopping in retail quanitfied by ~0.06 increase in gcmr and fewer people in grocery annd phramacy businesses quantified by a ~0.05 decrease in gcmr. These differences are small and it is important to take these conclusion with a pinch of salt: the gcmr data is based on a normal level of footfall in these businnesses in each country, so differences in how much difference there is in footfall under normal conditions between different countries is not accounted for.

From here, we decided to use stepwise regression so that the model could choose between a few variables which it considered to be most useful for predicting overall trends in gcmr. This should allow us to draw the best conclusions on how different factors. In this case we are using a negative binomial model. Negative binomial is a suitable choice since the GCMR data is at its core based on changes in count data compared to a baseline, which would normally be modelled with a Poisson distribution, however since there is high variance in the data, a Poisson model is not an appropriate model, so we choose a negative binomial model which is often standard to use to model count data with high variance. We also include a log term so our models can better fit the curves.

```
modelData$Place <- NULL
minModel1 <- glm.nb(data =
          filter(modelData, positive_rolling_gcmr_retail > 0, days_in_lockdown > 0),
          round(positive_rolling_gcmr_retail) ~ 1)
maxModel1 <- glm.nb(data =
        filter(modelData, positive_rolling_gcmr_retail > 0, days_in_lockdown > 0),
```

```
       round(positive_rolling_gcmr_retail)
        ~ 1 + gdp_capita + average_temperature + national_lockdown_length +
           days_in_lockdown + I(log(days_in_lockdown)))
autoBack1 <- step(maxModel1, direction = "backward",
                  scope = list("lower" = minModel1), trace = FALSE)
autoForward1 <- step(minModel1, direction = "forward",
                     scope = list("upper" = maxModel1), trace = FALSE)
autoBoth1 <- step(minModel1, direction = "both",
                  scope = list("lower" = minModel1, "upper" = maxModel1), trace = FALSE)
summary(autoBack1)
```

```
##
## Call:
## glm.nb(formula = round(positive_rolling_gcmr_retail) ~ average_temperature +
##     national_lockdown_length + days_in_lockdown + I(log(days_in_lockdown)),
##     data = filter(modelData, positive_rolling_gcmr_retail > 0,
##         days_in_lockdown > 0), init.theta = 17.55237337, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.4237  -0.6935   0.1544   0.7136   1.4280
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 3.8822530  0.0301641 128.704  < 2e-16 ***
## average_temperature         0.0204704  0.0007532  27.177  < 2e-16 ***
## national_lockdown_length -0.0017430  0.0002320  -7.515 5.71e-14 ***
## days_in_lockdown         -0.0060352  0.0004312 -13.997  < 2e-16 ***
## I(log(days_in_lockdown))  0.0794591  0.0108865   7.299 2.90e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(17.5524) family taken to be 1)
##
##     Null deviance: 3611.9  on 2433  degrees of freedom
## Residual deviance: 2697.9  on 2429  degrees of freedom
## AIC: 20714
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  17.552
##           Std. Err.:  0.692
##
##  2 x log-likelihood:  -20702.418
```

```
summary(autoForward1)
```

```
##
## Call:
## glm.nb(formula = round(positive_rolling_gcmr_retail) ~ average_temperature +
##     days_in_lockdown + national_lockdown_length + I(log(days_in_lockdown)),
##     data = filter(modelData, positive_rolling_gcmr_retail > 0,
##         days_in_lockdown > 0), init.theta = 17.55234563, link = log)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.4237  -0.6935   0.1544   0.7136   1.4280
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.8822524  0.0301642 128.704  < 2e-16 ***
## average_temperature        0.0204705  0.0007532  27.178  < 2e-16 ***
## days_in_lockdown          -0.0060352  0.0004312 -13.997  < 2e-16 ***
## national_lockdown_length -0.0017430  0.0002320  -7.515  5.7e-14 ***
## I(log(days_in_lockdown))  0.0794591  0.0108865   7.299  2.9e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(17.5523) family taken to be 1)
##
##     Null deviance: 3611.9  on 2433  degrees of freedom
## Residual deviance: 2697.9  on 2429  degrees of freedom
## AIC: 20714
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  17.552
##           Std. Err.:  0.692
##
##  2 x log-likelihood:  -20702.418
```

```
summary(autoBoth1)
```

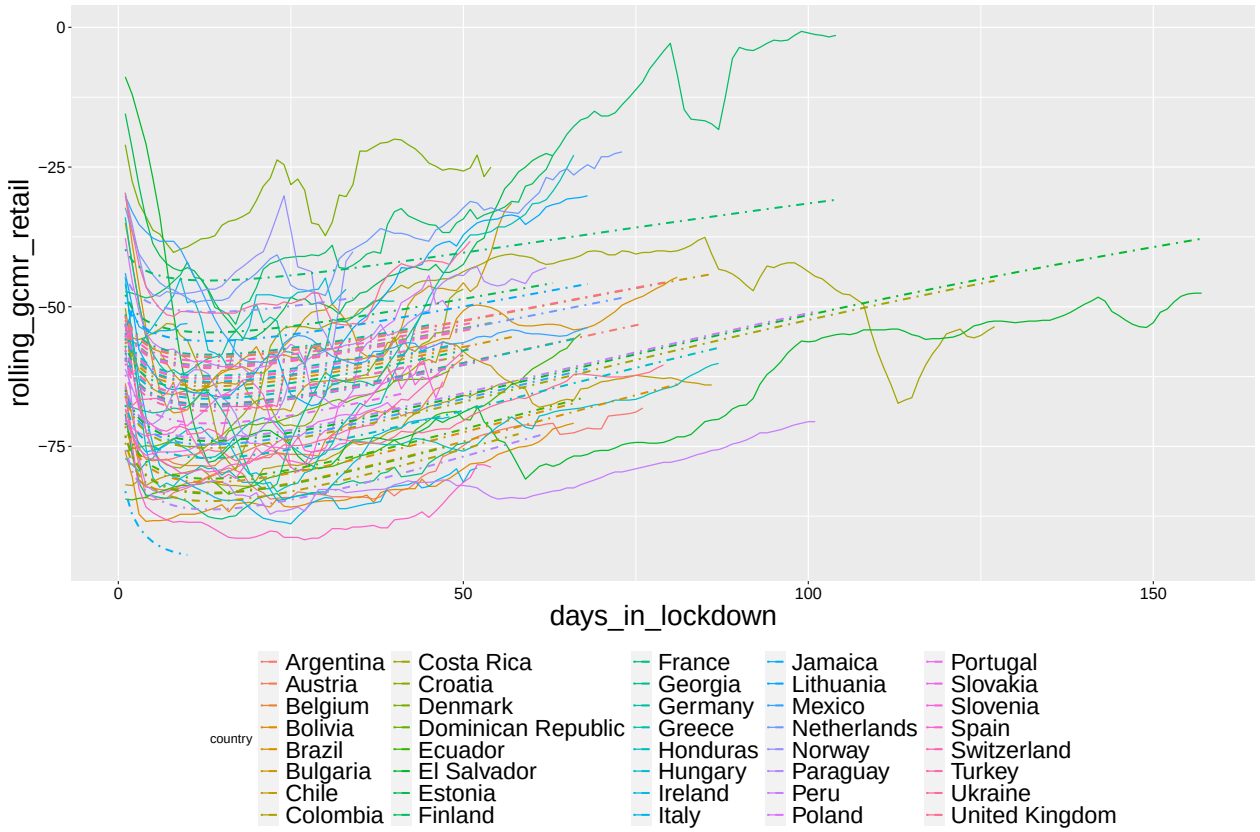
```
##
## Call:
## glm.nb(formula = round(positive_rolling_gcmr_retail) ~ average_temperature +
##     days_in_lockdown + national_lockdown_length + I(log(days_in_lockdown)),
##     data = filter(modelData, positive_rolling_gcmr_retail > 0,
##         days_in_lockdown > 0), init.theta = 17.55234563, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.4237  -0.6935   0.1544   0.7136   1.4280
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.8822524  0.0301642 128.704  < 2e-16 ***
## average_temperature        0.0204705  0.0007532  27.178  < 2e-16 ***
## days_in_lockdown          -0.0060352  0.0004312 -13.997  < 2e-16 ***
## national_lockdown_length -0.0017430  0.0002320  -7.515  5.7e-14 ***
## I(log(days_in_lockdown))  0.0794591  0.0108865   7.299  2.9e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(17.5523) family taken to be 1)
##
##     Null deviance: 3611.9  on 2433  degrees of freedom
```

```
## Residual deviance: 2697.9  on 2429  degrees of freedom
## AIC: 20714
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  17.552
##           Std. Err.:  0.692
##
##  2 x log-likelihood:  -20702.418
```

Forward, backward and both directions stepwise regression all choose the same model. Here we get the algorithm choosing both the length of national lockdown and the average temperature of a country and days in lockdown to be predictor variables for gcmr data. We see a similarly small effect from the length of national lockdown, as before, but we also interestingly see that the higher the average temperature of a country, the lower we would expect footfall in retail businesses to be on average. A lockdown one month longer would predict that gcmr for retail would be better by ~1.05 and an increase of one degree in average temperature would predict a lower gcmr overall by about -1.02. Generally over time, gcmr increases which is given by the coefficient of -0.0060352 for `days_in_lockdown`.

A plot of the model is shown against the data as below:

```
ggplot(data = filter(modelData, positive_rolling_gcmr_retail > 0, days_in_lockdown > 0),
  aes(x = days_in_lockdown, y = rolling_gcmr_retail, group = country, col = country)) +
  geom_line() +
  geom_line(aes(y = -exp(predict(autoBoth1))), linetype = "dotdash", size = 0.8) +
  theme(legend.position = "bottom")
```

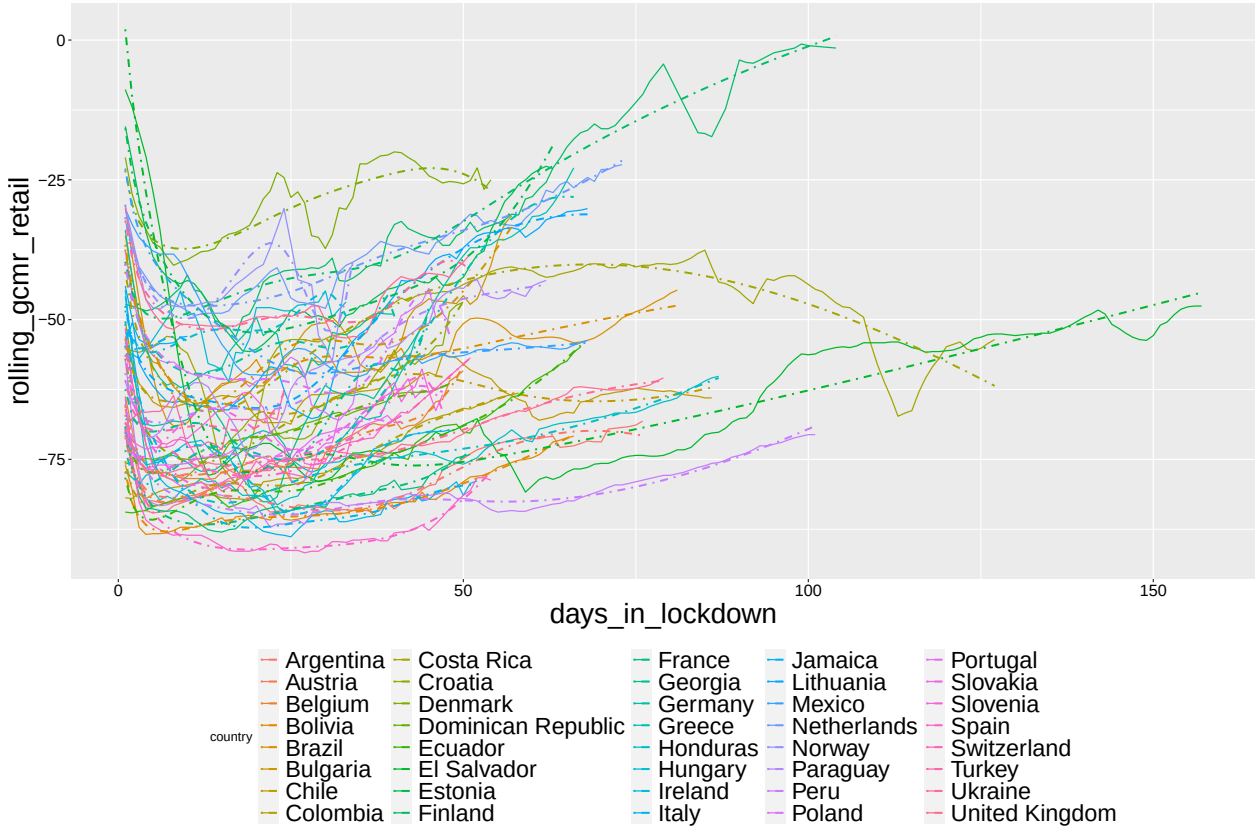


To fit the model more closely to the curves we use a spline model as follows, allowing the model to

```
splineModelRetail <-
  glm(data = filter(modelData, gcmr_retail_recreation < 0, days_in_lockdown > 0),
      rolling_gcmr_retail ~ days_in_lockdown +
        national_lockdown_length + country + country:ns(days_in_lockdown, df = 5) +
        average_temperature + I(log(days_in_lockdown)))
head(coef(splineModelRetail))
```

```
##             (Intercept)         days_in_lockdown national_lockdown_length
##            -168.1514568               0.2650759               1.2151424
##          countryAustria           countryBelgium           countryBolivia
##              61.5743218              36.8146011               5.6035997
```

```
ggplot(data = filter(modelData, gcmr_retail_recreation < 0, days_in_lockdown > 0),
  aes(x = days_in_lockdown, y = rolling_gcmr_retail, group = country, col = country)) +
  geom_line() +
  geom_line(aes(y = predict(splineModelRetail)), linetype = "dotdash", size = 0.8) +
  theme(legend.position = "bottom")
```



Here we can see that the model curves fit the data very well, and we get a coefficient of 1.215 for national lockdown length. This means this model "predicts" that for a national lockdown being one day longer means GCMR for retail will be on average 1.215 higher. We have to be careful about taking this coefficient too seriously though since spline models are not really made for prediction, and since this models differently over several countries it would be hard to make a conclusion about the overall trend since the data is overfitted.

We do the same thing, this time modelling grocery and pharmacy

```
minModel2 <- glm.nb(data =
        filter(modelData, positive_rolling_gcmr_grocery > 0, days_in_lockdown > 0),
        round(positive_rolling_gcmr_grocery) ~ 1)
```

```
maxModel2 <- glm.nb(data =
        filter(modelData, positive_rolling_gcmr_grocery > 0, days_in_lockdown > 0),
        round(positive_rolling_gcmr_grocery)
        ~ 1 + gdp_capita + average_temperature + national_lockdown_length +
        days_in_lockdown + I(log(days_in_lockdown)))
autoBack2 <- step(maxModel2, direction = "backward",
                  scope = list("lower" = minModel2), trace = FALSE)
autoForward2 <- step(minModel2, direction = "forward",
                     scope = list("upper" = maxModel2), trace = FALSE)
autoBoth2 <- step(minModel2, direction = "both",
                  scope = list("lower" = minModel2, "upper" = maxModel2), trace = FALSE)
summary(autoBack2)
```

```
##
## Call:
## glm.nb(formula = round(positive_rolling_gcmr_grocery) ~ 1 + gdp_capita +
##     average_temperature + national_lockdown_length + days_in_lockdown +
##     I(log(days_in_lockdown)), data = filter(modelData, positive_rolling_gcmr_grocery >
##     0, days_in_lockdown > 0), init.theta = 5.25188847, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7350  -0.7280   0.0354   0.5805   2.9781
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.0044544  0.0574099  52.333  < 2e-16 ***
## gdp_capita                -0.1994763  0.0133827 -14.905  < 2e-16 ***
## average_temperature        0.0299884  0.0017579  17.059  < 2e-16 ***
## national_lockdown_length -0.0014774  0.0004155  -3.556 0.000377 ***
## days_in_lockdown         -0.0076705  0.0007685  -9.981  < 2e-16 ***
## I(log(days_in_lockdown))  0.1037154  0.0195947   5.293  1.2e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(5.2519) family taken to be 1)
##
##     Null deviance: 3746.9  on 2324  degrees of freedom
## Residual deviance: 2535.1  on 2319  degrees of freedom
## AIC: 18976
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  5.252
##           Std. Err.:  0.188
##
##  2 x log-likelihood:  -18962.057
```

```
summary(autoForward2)
```

```
##
## Call:
## glm.nb(formula = round(positive_rolling_gcmr_grocery) ~ gdp_capita +
```

```
##     average_temperature + days_in_lockdown + I(log(days_in_lockdown)) +
##     national_lockdown_length, data = filter(modelData, positive_rolling_gcmr_grocery >
##     0, days_in_lockdown > 0), init.theta = 5.251888471, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7350  -0.7280   0.0354   0.5805   2.9781
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.0044544  0.0574099  52.333  < 2e-16 ***
## gdp_capita                -0.1994763  0.0133827 -14.905  < 2e-16 ***
## average_temperature        0.0299884  0.0017579  17.059  < 2e-16 ***
## days_in_lockdown          -0.0076705  0.0007685  -9.981  < 2e-16 ***
## I(log(days_in_lockdown))   0.1037154  0.0195947   5.293  1.2e-07 ***
## national_lockdown_length -0.0014774  0.0004155  -3.556 0.000377 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(5.2519) family taken to be 1)
##
##     Null deviance: 3746.9  on 2324  degrees of freedom
## Residual deviance: 2535.1  on 2319  degrees of freedom
## AIC: 18976
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  5.252
##           Std. Err.:  0.188
##
##  2 x log-likelihood:  -18962.057
```

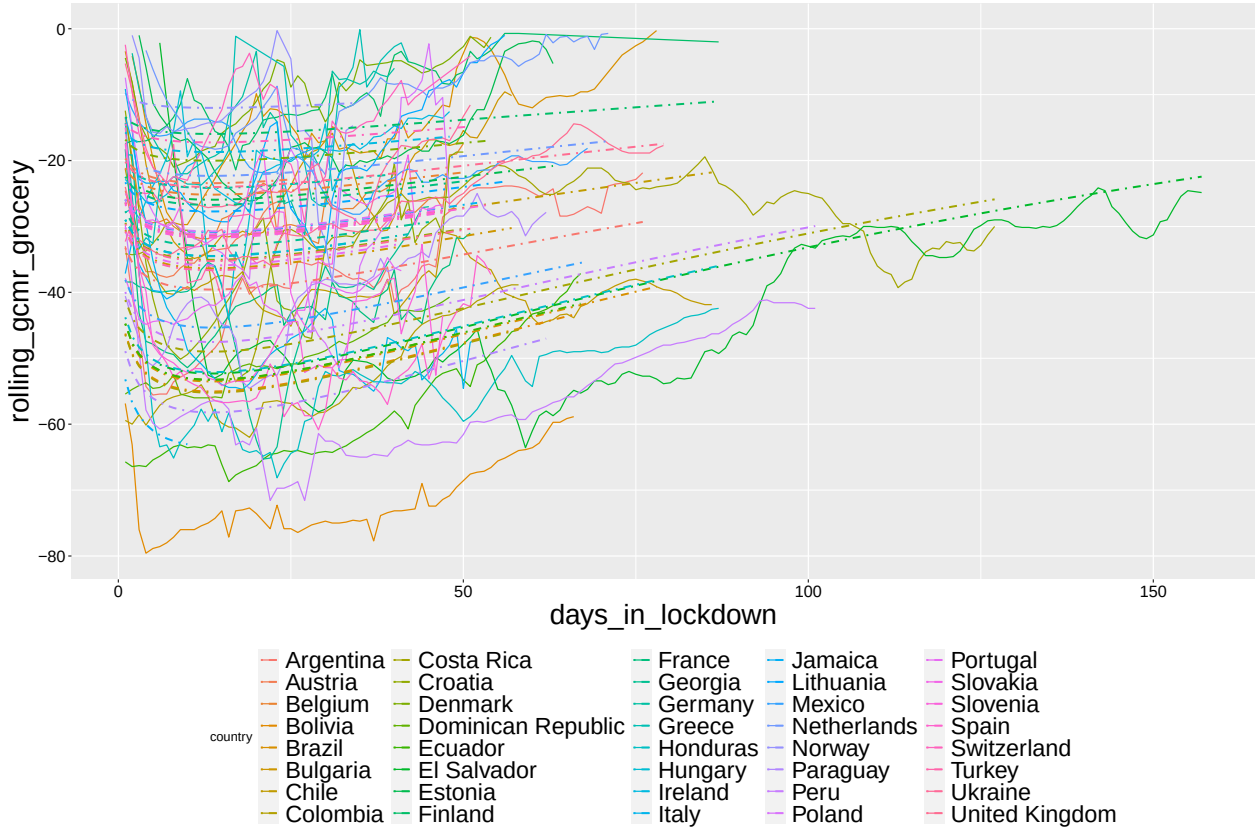
```
summary(autoBoth2)
```

```
##
## Call:
## glm.nb(formula = round(positive_rolling_gcmr_grocery) ~ gdp_capita +
##     average_temperature + days_in_lockdown + I(log(days_in_lockdown)) +
##     national_lockdown_length, data = filter(modelData, positive_rolling_gcmr_grocery >
##     0, days_in_lockdown > 0), init.theta = 5.251888471, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7350  -0.7280   0.0354   0.5805   2.9781
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                3.0044544  0.0574099  52.333  < 2e-16 ***
## gdp_capita                -0.1994763  0.0133827 -14.905  < 2e-16 ***
## average_temperature        0.0299884  0.0017579  17.059  < 2e-16 ***
## days_in_lockdown          -0.0076705  0.0007685  -9.981  < 2e-16 ***
## I(log(days_in_lockdown))   0.1037154  0.0195947   5.293  1.2e-07 ***
## national_lockdown_length -0.0014774  0.0004155  -3.556 0.000377 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(5.2519) family taken to be 1)
##
##     Null deviance: 3746.9  on 2324  degrees of freedom
## Residual deviance: 2535.1  on 2319  degrees of freedom
## AIC: 18976
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  5.252
##           Std. Err.:  0.188
##
##  2 x log-likelihood:  -18962.057
```

Forward, backward and both directions stepwise regression all choose the same model. Hence, a plot of the model is shown against the data as below:

```
ggplot(data = filter(modelData, positive_rolling_gcmr_grocery > 0, days_in_lockdown > 0),
 aes(x = days_in_lockdown, y = rolling_gcmr_grocery, group = country, col = country)) +
  geom_line() +
  geom_line(aes(y = -exp(predict(autoBoth2))), linetype = "dotdash", size = 0.8) +
  theme(legend.position = "bottom")
```



In this case the stepwise regression picks a model with three predictor variables: gdp per capita, average temperature and the length of national lockdown. Average temperature and national lockdown length have a similar effects in this model, with footfall to grocery and pharmacy businesses being higher with national lockdown lengths and lower with greater average temperatures. In this case gdp per capita is also chosen as a

good predictor for the gcmr data. It indicates that countries with a better performing economy tended to have significantly more activity in grocery and pharmacy stores as given by the coefficient of -0.199 (remembering that the model is fitting gcmr values times -1 to make them positive)

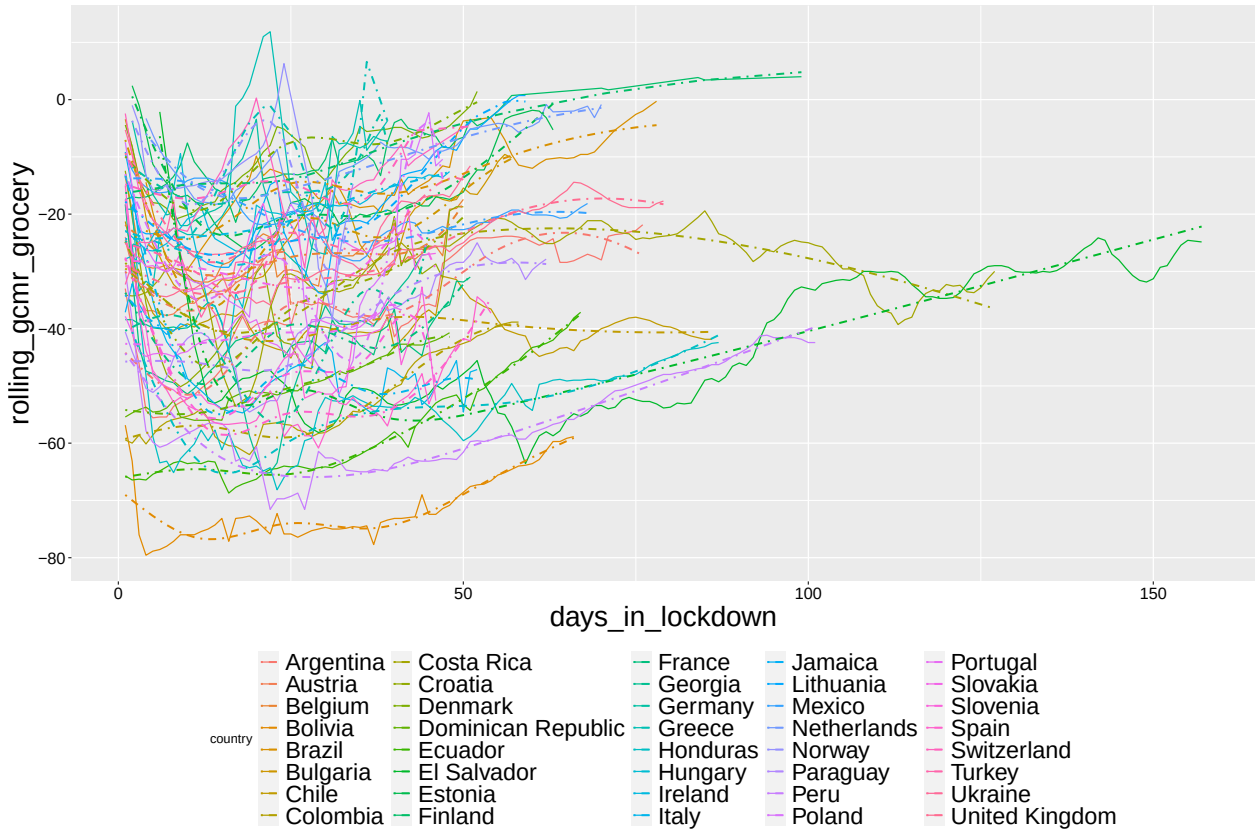
The tendency in both of the modelling situation for national lockdown length having a positive coefficient may be explained by confidence in the public in the safety of shopping. Whether countries had national lockdowns or not we saw an overall trend in February-April of 2020 where footfall to all businesses suddenly plummeted which in part would be explained by people's reluctance to have contact with other people for fear of contracting Covid, but as many businesses took steps to make shopping safer (such as providing hand sanitiser at entrances and enforcing the wearing of face-masks) the public's willingness to go shopping would have increased due to these measures, so we see that in long lockdowns, there seems to be less fear of Covid as time goes on. There is of course no way of proving this effect with our data but it offers a rational explanation of why longer lockdowns seem to indicate more activity in both retail and grocery businesses.

Once again we try a spline model to get the best fit to the data as we did before:

```
splineModelGrocery <- glm(data =
            filter(modelData, gcmr_grocery_pharmacy < 0, days_in_lockdown > 0),
            rolling_gcmr_grocery ~ days_in_lockdown + national_lockdown_length
            + country + country:ns(days_in_lockdown, df = 5))
head(coef(splineModelGrocery))
```

```
##              (Intercept)         days_in_lockdown national_lockdown_length
##              -449.8562642              -0.3525649                5.0749427
##            countryAustria           countryBelgium           countryBolivia
##               272.9609823              154.2853063               25.9262965
```

```
ggplot(data = filter(modelData, gcmr_grocery_pharmacy < 0, days_in_lockdown > 0),
  aes(x = days_in_lockdown, y = rolling_gcmr_grocery, group = country, col = country)) +
  geom_line() +
  geom_line(aes(y = predict(splineModelGrocery)), linetype = "dotdash", size = 0.8) +
  theme(legend.position = "bottom")
```



In this case we get a coefficient of 5.075 for national lockdown length so an increase of 1 day in national lockdown length would supposedly lead to an overall 5% increase in footfall to grocery and pharmacy businesses. Again, this coefficient cannot be taken particularly seriously for the same reasons as with the original spline model.

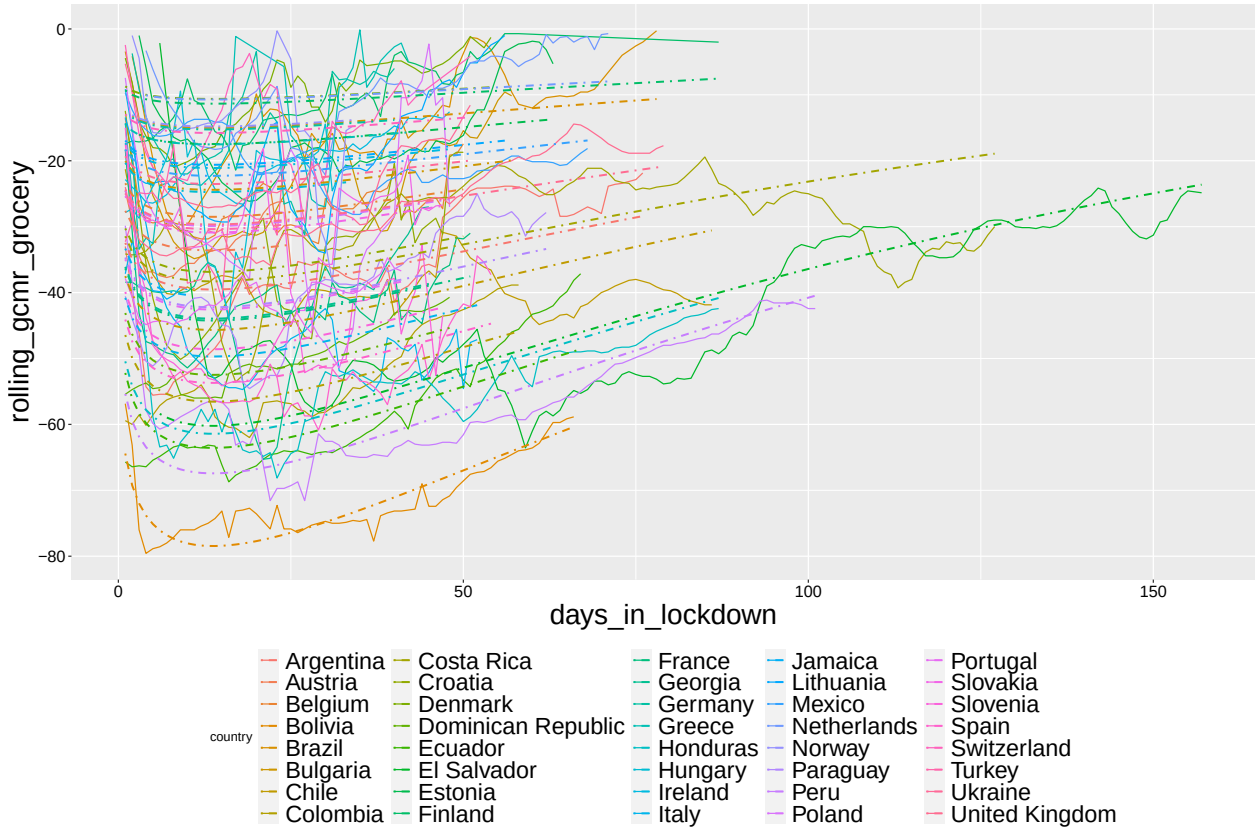
We consider allowing country to be a variable in our previous models (ones without splines):

```
minModel2 <- glm.nb(data =
     filter(modelData, positive_rolling_gcmr_grocery > 0, days_in_lockdown > 0),
     round(positive_rolling_gcmr_grocery) ~ 1)
maxModel2 <- glm.nb(data =
    filter(modelData, positive_rolling_gcmr_grocery > 0, days_in_lockdown > 0),
    round(positive_rolling_gcmr_grocery)
    ~ 1 + gdp_capita + average_temperature + national_lockdown_length +
    days_in_lockdown + I(log(days_in_lockdown)) + country)
autoBack2 <- step(maxModel2, direction = "backward",
                  scope = list("lower" = minModel2), trace = FALSE)
autoForward2 <- step(minModel2, direction = "forward",
                     scope = list("upper" = maxModel2), trace = FALSE)
autoBoth2 <- step(minModel2, direction = "both",
        scope = list("lower" = minModel2, "upper" = maxModel2), trace = FALSE)
```

Forward, backward and both directions stepwise regression all choose the same model. Hence, a plot of the model is shown against the data as below:

```
ggplot(data =
    filter(modelData, positive_rolling_gcmr_grocery > 0, days_in_lockdown > 0),
    aes(x = days_in_lockdown, y = rolling_gcmr_grocery,
    group = country, col = country)) +
```

```
    geom_line() +
    geom_line(aes(y = -exp(predict(autoBoth2))),
    linetype = "dotdash", size = 0.8) +
    theme(legend.position = "bottom")
```



This informs us that when split over countries, the model fits quite accurately so we can be confident that our negative binomial models for gcmr are giving us fairly good predictions and information.

Investigating lockdown recovery

Preparing the data for realignment based on lockdown end

```
source_url("https://raw.github.com/timleeman/ST344GroupProject/main/getdata.R")
```

```
## SHA-1 hash of file is cf65a1bd6f6310fc5c5fd8b39b0b5fd2b52d8f24
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Country = col_character(),
##   Place = col_character(),
##   StartDate = col_date(format = ""),
##   EndDate = col_date(format = ""),
##   Level = col_character(),
##   url = col_character(),
##   update = col_date(format = ""),
##   Confirmed = col_logical()
```

```
## )
##
## -- Column specification --------------------------------------------------------
## cols(
##   continent = col_character(),
##   country = col_character()
## )
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   iso3c = col_character(),
##   country = col_character(),
##   date = col_date(format = ""),
##   tests_units = col_character(),
##   region = col_character(),
##   income = col_character(),
##   timestamp = col_datetime(format = "")
## )
## i Use `spec()` for the full column specifications.
##
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   iso3c = col_character(),
##   country = col_character(),
##   date = col_date(format = ""),
##   tests_units = col_character(),
##   region = col_character(),
##   income = col_character(),
##   timestamp = col_datetime(format = "")
## )
## i Use `spec()` for the full column specifications.
```

```
covidData$date = as.Date(parse_date_time(covidData$date,orders=c("y","ym","ymd")))

TidyData <- covidData %>% group_by(country)
TidyData <- mutate(TidyData, country = factor(country), continent = factor(continent))

national_lockdowns <- auravisionData %>% filter(Level== "National") %>%
  dplyr::select(-Level)
national_lockdowns <- national_lockdowns %>%
  mutate(national_lockdown_length =
           as.numeric(-difftime(`StartDate`, `EndDate`)))
colnames(national_lockdowns)[1] = "country"
```

Rearranging so day 0 is the end of each country's respective lockdowns.

```
selectData <- filter(covidData, country %in% chosenCountries)
#Function to select country
lockdown_start_date <- function(cou){
  a <- national_lockdowns %>% filter(country == cou) %>%
```

```
    dplyr::select(`StartDate`)
  return(a$`StartDate`[1])
}
lockdown_end_date <- function(cou){
  a <- national_lockdowns %>% filter(country == cou) %>%
    dplyr::select(`EndDate`)
  return(a$`EndDate`[1])
}
```

Adding columns for moving averages of gcmr retail/recreation, grocery/pharmacy and daily confirmed cases

```
selectData <- selectData %>% mutate(daily_confirmed = firstdiff(confirmed))
get_lockdown_data <- function(cou){
  lockdown_data <- selectData %>% filter(country == cou) %>%
    filter(date >= lockdown_end_date(cou))  %>%
    mutate(days_out_lockdown = as.numeric(-difftime(lockdown_end_date(cou), date)/86400),
    rolling_gcmr_retail = zoo::rollmean(gcmr_retail_recreation, k = 7, fill = NA),
    rolling_gcmr_grocery = zoo::rollmean(gcmr_grocery_pharmacy, k = 7, fill = NA),
    rolling_dc = zoo::rollmean(daily_confirmed, k = 7, fill = NA))
  return(lockdown_data)
}
lockdown_data <- data_frame()
```

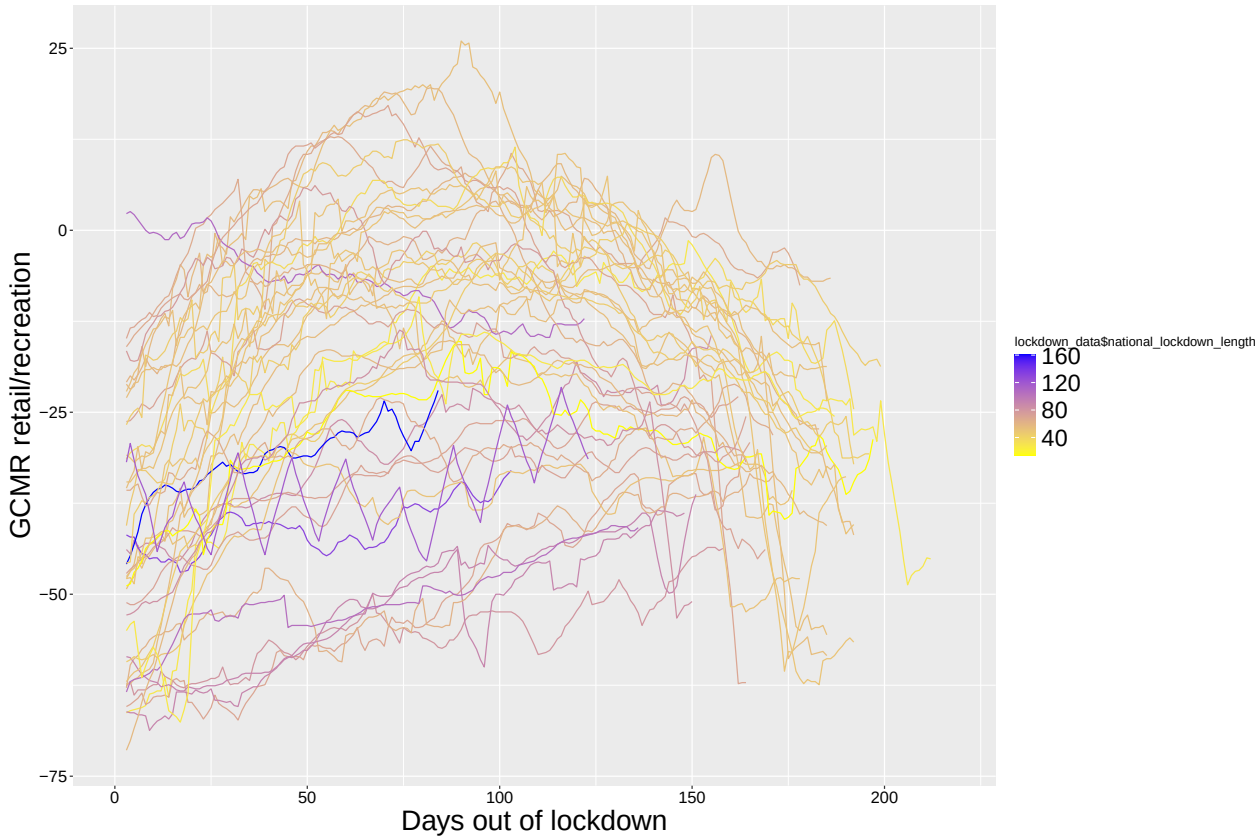
```
## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
for(cou in national_lockdowns$country){
  a <- get_lockdown_data(cou)
  lockdown_data <- rbind(a, lockdown_data)
}
lockdown_data <- left_join(lockdown_data, national_lockdowns, by = "country")
```

Plotting (rolling) gcmr recovery with gradient based on total length - (just retail/recreation)

```
ggplot(lockdown_data, aes(x=days_out_lockdown, y= rolling_gcmr_retail, group= country)) +
  geom_line(aes(colour = lockdown_data$national_lockdown_length ))+
  scale_colour_gradient(low = "yellow", high = "blue")+
  labs( x = "Days out of lockdown", y = "GCMR retail/recreation")
```

```
## Warning: Removed 410 row(s) containing missing values (geom_path).
```



Adding average temp for use as a possible explanatory variable & creating datafram with the relevant explanatory variables for modelling

```
average_temperature <- import("https://raw.github.com/timleeman/ST344GroupProject/main/Modelling/averag
modelData <- lockdown_data %>%
  dplyr::select(country, daily_confirmed, date, continent,
                gdp_capita, gcmr_retail_recreation, gcmr_grocery_pharmacy,
                days_out_lockdown, income, national_lockdown_length,
                rolling_gcmr_retail,
                rolling_gcmr_grocery, population, rolling_dc) %>%
                distinct()
modelData <- left_join(modelData, average_temperature, by= "country")
modelData$gdp_capita <- scale(modelData$gdp_capita)
```

Modeling rolling average of retail/recreation gcmr with some explanatory variables

```
modelData <- na.omit(modelData)

minModel <- glm(data = modelData, rolling_gcmr_retail ~ 1)
maxModel <- glm(data = modelData,
                rolling_gcmr_retail ~ gdp_capita + days_out_lockdown +
                  national_lockdown_length + average_temperature)
autoBack <- step(maxModel, direction = "backward",
                 scope = list("lower" = minModel), trace = FALSE)
autoForward <- step(minModel, direction = "forward",
                    scope = list("upper" = maxModel), trace = FALSE)
```

```r
autoBoth <- step(minModel, direction = "both",
                 scope = list("lower" = minModel, "upper" = maxModel), trace = FALSE)
summary(autoBack)
```

```
##
## Call:
## glm(formula = rolling_gcmr_retail ~ gdp_capita + days_out_lockdown +
##     national_lockdown_length + average_temperature, data = modelData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -53.209   -8.359    1.125   10.912   44.189
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.699000   0.858824   5.471 4.63e-08 ***
## gdp_capita                -1.235225   0.249235  -4.956 7.37e-07 ***
## days_out_lockdown          0.030661   0.003894   7.874 3.98e-15 ***
## national_lockdown_length -0.135081   0.008799 -15.351  < 2e-16 ***
## average_temperature      -1.404116   0.034332 -40.898  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 255.4531)
##
##     Null deviance: 2533300  on 6677  degrees of freedom
## Residual deviance: 1704639  on 6673  degrees of freedom
## AIC: 55975
##
## Number of Fisher Scoring iterations: 2
```

```r
summary(autoForward)
```

```
##
## Call:
## glm(formula = rolling_gcmr_retail ~ average_temperature + national_lockdown_length +
##     days_out_lockdown + gdp_capita, data = modelData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -53.209   -8.359    1.125   10.912   44.189
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.699000   0.858824   5.471 4.63e-08 ***
## average_temperature      -1.404116   0.034332 -40.898  < 2e-16 ***
## national_lockdown_length -0.135081   0.008799 -15.351  < 2e-16 ***
## days_out_lockdown          0.030661   0.003894   7.874 3.98e-15 ***
## gdp_capita                -1.235225   0.249235  -4.956 7.37e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 255.4531)
##
```

```
##     Null deviance: 2533300  on 6677  degrees of freedom
## Residual deviance: 1704639  on 6673  degrees of freedom
## AIC: 55975
##
## Number of Fisher Scoring iterations: 2
```

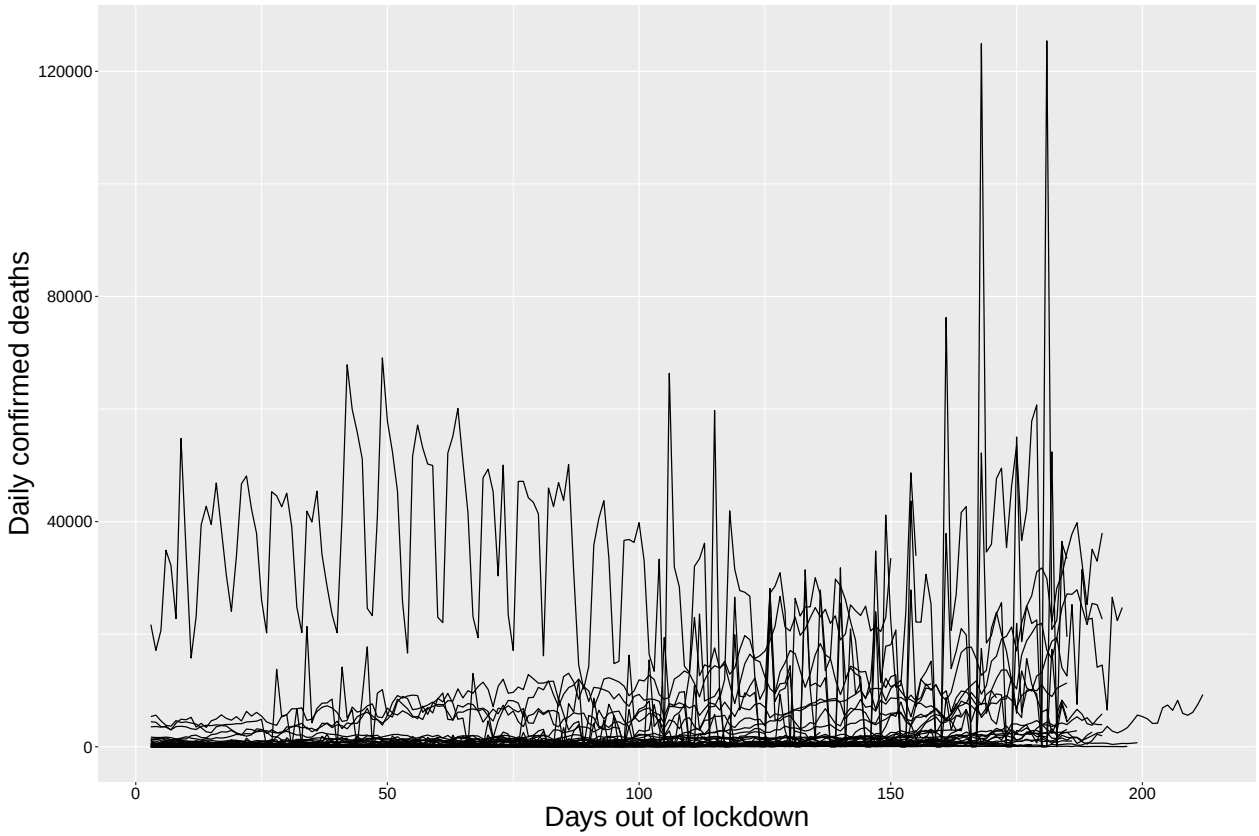
```
summary(autoBoth)
```

```
##
## Call:
## glm(formula = rolling_gcmr_retail ~ average_temperature + national_lockdown_length +
##     days_out_lockdown + gdp_capita, data = modelData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -53.209   -8.359    1.125   10.912   44.189
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.699000   0.858824   5.471 4.63e-08 ***
## average_temperature       -1.404116   0.034332 -40.898  < 2e-16 ***
## national_lockdown_length -0.135081   0.008799 -15.351  < 2e-16 ***
## days_out_lockdown          0.030661   0.003894   7.874 3.98e-15 ***
## gdp_capita                -1.235225   0.249235  -4.956 7.37e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 255.4531)
##
##     Null deviance: 2533300  on 6677  degrees of freedom
## Residual deviance: 1704639  on 6673  degrees of freedom
## AIC: 55975
##
## Number of Fisher Scoring iterations: 2
```

Step-wise regression all choose the same model. Model chooses average temp, lockdown length and days out of lockdown as variables. Intuitively the only coefficient that does not make sense is the average temperature. However, it also has the largest effect so may need to reconsider.

Quick plot of daily confirmed against days out of lockdown

```
ggplot(modelData, aes(x=days_out_lockdown,
                      y= daily_confirmed, group= country)) +
  geom_line()+
  labs( x = "Days out of lockdown", y = "Daily confirmed deaths")
```

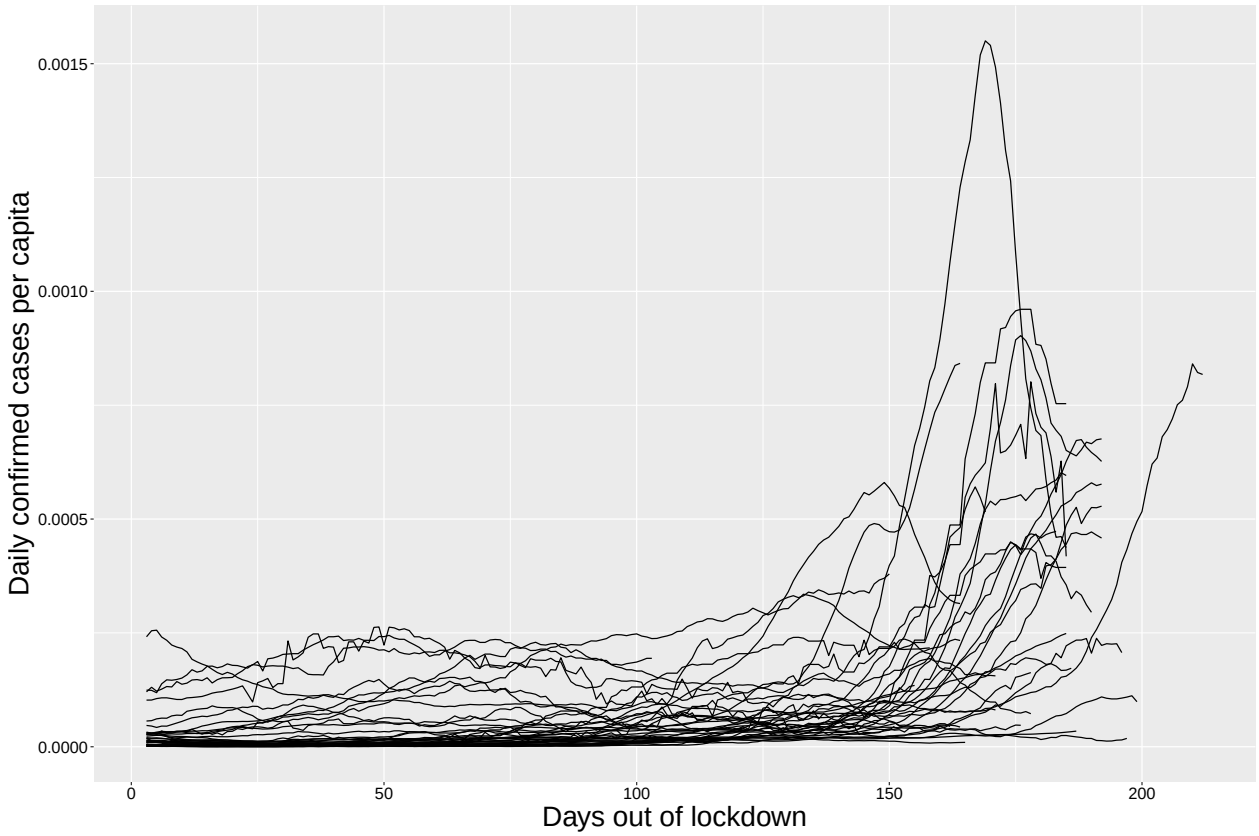


Adding per capita rolling daily cases to our modeling data

```
modelData <- modelData %>% mutate(dc_percapita = rolling_dc*(1/population))
```

Quick plot

```
ggplot(modelData, aes(x=days_out_lockdown,
                      y= dc_percapita, group= country)) +
  geom_line()+
  labs( x = "Days out of lockdown", y = "Daily confirmed cases per capita")
```



Modelling with rolling daily cases per capita

```
modelData <- na.omit(modelData)

minModel <- glm(data = modelData, dc_percapita ~ 1)
maxModel <- glm(data = modelData, dc_percapita ~ gdp_capita +
                  days_out_lockdown  +
                  national_lockdown_length + average_temperature)
autoBack <- step(maxModel, direction = "backward",
                 scope = list("lower" = minModel), trace = FALSE)
autoForward <- step(minModel, direction = "forward",
                    scope = list("upper" = maxModel), trace = FALSE)
autoBoth <- step(minModel, direction = "both",
                 scope = list("lower" = minModel, "upper" = maxModel), trace = FALSE)
summary(autoBack)
```

```
##
## Call:
## glm(formula = dc_percapita ~ gdp_capita + days_out_lockdown +
##     national_lockdown_length + average_temperature, data = modelData)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.008e-04  -7.017e-05  -2.317e-05   3.864e-05   1.358e-03
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)              -1.329e-04  6.638e-06 -20.020  < 2e-16 ***
## gdp_capita                7.218e-06  1.926e-06   3.747 0.000181 ***
## days_out_lockdown         1.432e-06  3.010e-08  47.589  < 2e-16 ***
## national_lockdown_length  1.105e-06  6.801e-08  16.243  < 2e-16 ***
## average_temperature       1.473e-06  2.654e-07   5.550 2.97e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.526149e-08)
##
##     Null deviance: 0.00013761  on 6677  degrees of freedom
## Residual deviance: 0.00010184  on 6673  degrees of freedom
## AIC: -101232
##
## Number of Fisher Scoring iterations: 2
```

```
summary(autoForward)
```

```
##
## Call:
## glm(formula = dc_percapita ~ days_out_lockdown + national_lockdown_length +
##     average_temperature + gdp_capita, data = modelData)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.008e-04  -7.017e-05  -2.317e-05   3.864e-05   1.358e-03
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.329e-04  6.638e-06 -20.020  < 2e-16 ***
## days_out_lockdown         1.432e-06  3.010e-08  47.589  < 2e-16 ***
## national_lockdown_length  1.105e-06  6.801e-08  16.243  < 2e-16 ***
## average_temperature       1.473e-06  2.654e-07   5.550 2.97e-08 ***
## gdp_capita                7.218e-06  1.926e-06   3.747 0.000181 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.526149e-08)
##
##     Null deviance: 0.00013761  on 6677  degrees of freedom
## Residual deviance: 0.00010184  on 6673  degrees of freedom
## AIC: -101232
##
## Number of Fisher Scoring iterations: 2
```

```
summary(autoBoth)
```

```
##
## Call:
## glm(formula = dc_percapita ~ days_out_lockdown + national_lockdown_length +
##     average_temperature + gdp_capita, data = modelData)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.008e-04  -7.017e-05  -2.317e-05   3.864e-05   1.358e-03
```

```
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1.329e-04  6.638e-06 -20.020  < 2e-16 ***
## days_out_lockdown          1.432e-06  3.010e-08  47.589  < 2e-16 ***
## national_lockdown_length   1.105e-06  6.801e-08  16.243  < 2e-16 ***
## average_temperature        1.473e-06  2.654e-07   5.550 2.97e-08 ***
## gdp_capita                 7.218e-06  1.926e-06   3.747 0.000181 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.526149e-08)
##
##     Null deviance: 0.00013761  on 6677  degrees of freedom
## Residual deviance: 0.00010184  on 6673  degrees of freedom
## AIC: -101232
##
## Number of Fisher Scoring iterations: 2
```

```
anova(autoBoth)
```

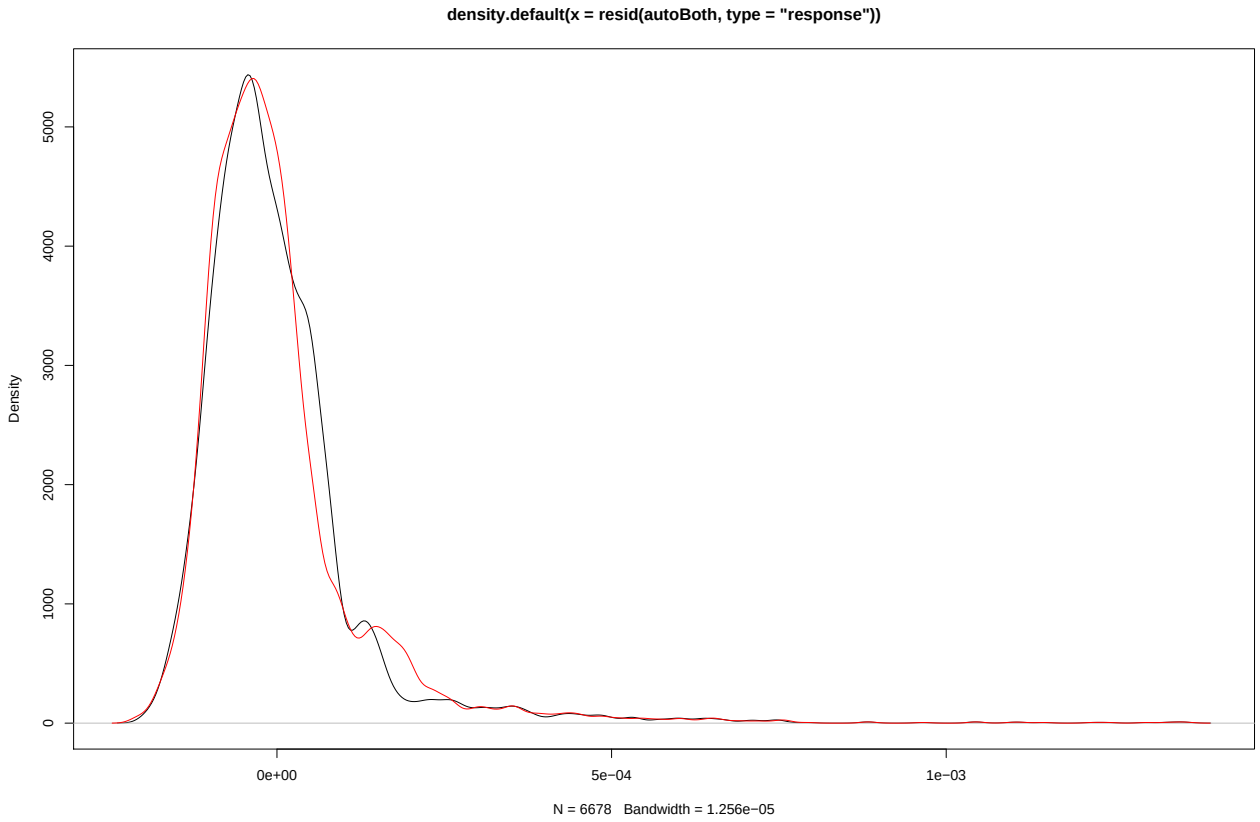
```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: dc_percapita
##
## Terms added sequentially (first to last)
##
##
##                           Df   Deviance Resid. Df Resid. Dev
## NULL                                        6677 0.00013761
## days_out_lockdown          1 3.0713e-05      6676 0.00010690
## national_lockdown_length   1 4.5829e-06      6675 0.00010231
## average_temperature        1 2.5900e-07      6674 0.00010205
## gdp_capita                 1 2.1430e-07      6673 0.00010184
```

```
with(summary(autoBoth), 1 - deviance/null.deviance)
```

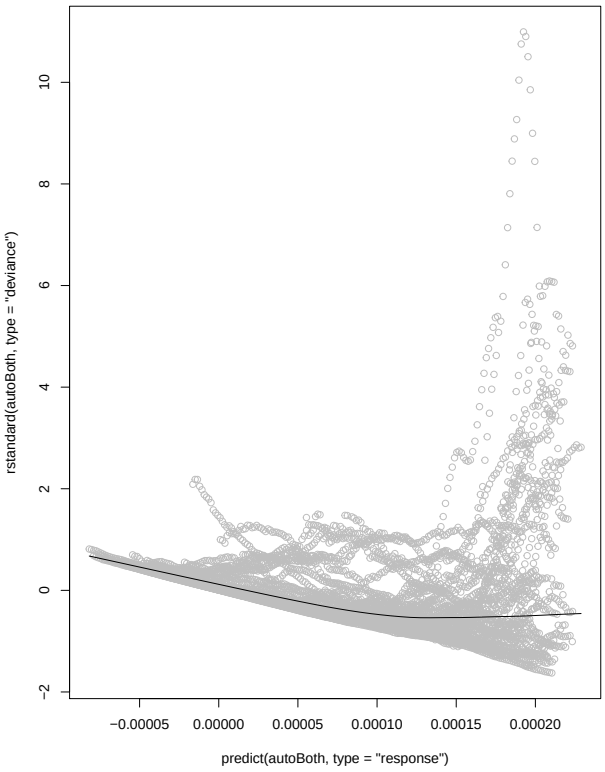
```
## [1] 0.2599359
```

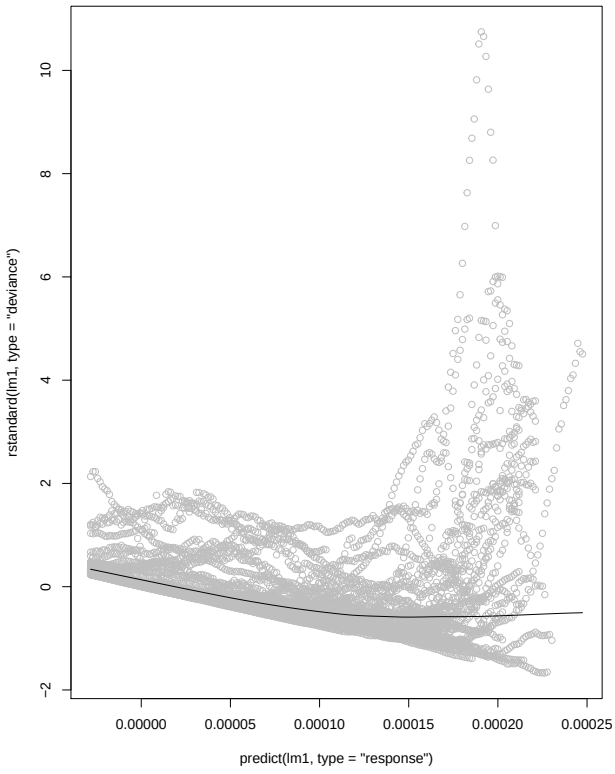
Step-wise regression all chose the same model. Model uses all explanatory variables afforded to it. Coefficients for lockdown length and days out of lockdown seem to be most important

```
lm1 <- glm(data = modelData, dc_percapita ~ days_out_lockdown)



#comparing residuals of iterative model with basic 'null model'- lm1
#residuals seem to have variance that is
#fairly constant however there is an extremely long tail

plot(density(resid(autoBoth, type='response')))
lines(density(resid(lm1, type='response')), col='red')
```




```
# Plotting residuals against fitted values of comparative model (lm1)
#discernible patterns with both models however looks
#stronger with stepwise regression model suggesting better fit

par(mfrow=c(1,2))
scatter.smooth(predict(autoBoth, type='response'),
               rstandard(autoBoth, type='deviance'), col='gray')
scatter.smooth(predict(lm1, type='response'),
               rstandard(lm1, type='deviance'), col='gray')
```


rstandard(autoBoth, type = "deviance")
predict(autoBoth, type = "response")
−2
0
2
4
6
8
10
−0.00005
0.00000
0.00005
0.00010
0.00015
0.00020


rstandard(lm1, type = "deviance")
predict(lm1, type = "response")
−2
0
2
4
6
8
10
0.00000
0.00005
0.00010
0.00015
0.00020
0.00025