State specific time series analysis

Eric Olle

April 22, 2020

This is the basic markdown document for the analysis of the timeseries data used in the JAMA article submitted on April 20, 2020.

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Loading the data from file

Data use if from the nytimes github account.

```
## Direct method in comments because I already downloaded the file

# read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties.csv")

us_counties20apr <- read_csv("~/Documents/epid_co19/NY times data/us-counties20apr.csv")
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

```
unique(us_counties20apr$state)
```

```
##  [1] "Washington"                "Illinois"
##  [3] "California"                "Arizona"
##  [5] "Massachusetts"             "Wisconsin"
##  [7] "Texas"                     "Nebraska"
##  [9] "Utah"                      "Oregon"
## [11] "Florida"                   "New York"
## [13] "Rhode Island"              "Georgia"
## [15] "New Hampshire"             "North Carolina"
## [17] "New Jersey"                "Colorado"
## [19] "Maryland"                 "Nevada"
## [21] "Tennessee"                "Hawaii"
## [23] "Indiana"                  "Kentucky"
## [25] "Minnesota"                "Oklahoma"
## [27] "Pennsylvania"             "South Carolina"
## [29] "District of Columbia"     "Kansas"
## [31] "Missouri"                 "Vermont"
## [33] "Virginia"                 "Connecticut"
## [35] "Iowa"                     "Louisiana"
## [37] "Ohio"                     "Michigan"
## [39] "South Dakota"             "Arkansas"
## [41] "Delaware"                 "Mississippi"
## [43] "New Mexico"               "North Dakota"
## [45] "Wyoming"                  "Alaska"
## [47] "Maine"                    "Alabama"
## [49] "Idaho"                    "Montana"
## [51] "Puerto Rico"              "Virgin Islands"
## [53] "Guam"                     "West Virginia"
## [55] "Northern Mariana Islands"
```

Setting the constants that apply to the individual state

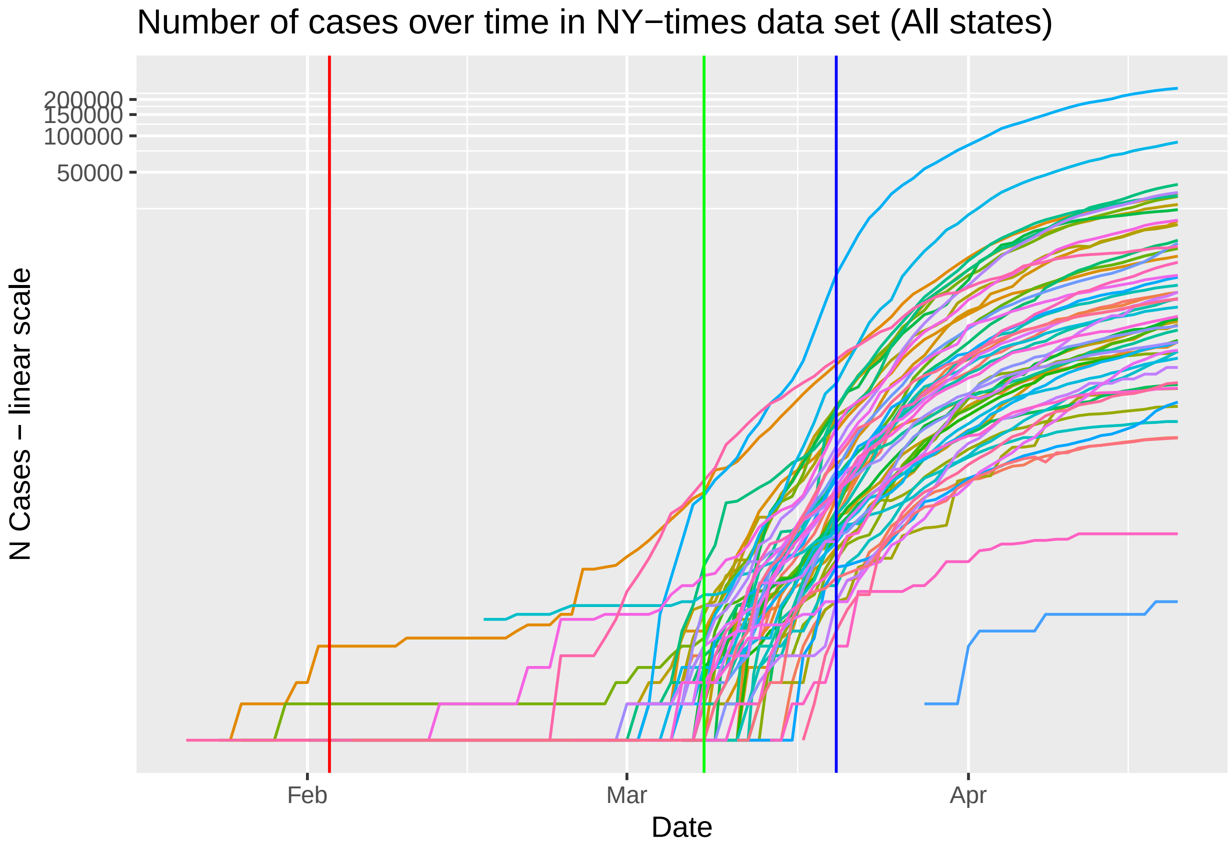
```r
# Which specific state?

state2select <- "West Virginia"

# What is the state specific lockdown?
# (follow the correct date format)

lockdown_st <- as.Date("2020-03-23")

# Known lockdown data

## Peoples Republic of China
lockdownprc <- as.Date("2020-02-03")
## Italy
lockdownitl <- as.Date("2020-03-08")
## USA - New York
lockdownnyc <- as.Date("2020-03-20")
```

Part 1. Time series analysis

In this section a basic time series plot of the full US along with the selected state will be done.

Plotting the full state data set.

```r
us_counties20apr %>%
  select(c(date, state, cases)) %>%
  group_by(date, state) %>%
  tally(cases) %>%
  ggplot(aes(x=date,y=n,colour=state,group=state)) +
  coord_trans(y="log") +
  scale_colour_discrete(guide = FALSE) +
  geom_line()+
  geom_vline(xintercept = as.numeric(lockdownnyc), color = "Blue") +
  geom_vline(xintercept = as.numeric(lockdownitl), color = "Green") +
  geom_vline(xintercept = as.numeric(lockdownprc), color = "Red") +
  ylab("N Cases - linear scale")+
  xlab("Date") +
  ggtitle(paste("Number of cases over time in NY-times data set (All states)"))
```



Selecting the state specific data from the NY Times dataset

From the unique state names call (above) set the state name by copy/paste or typing in with the quotation marks.

Selecting the state data from state2select var

The st data set was selected in case future geo-spatial on the rate of cases by county is needed.

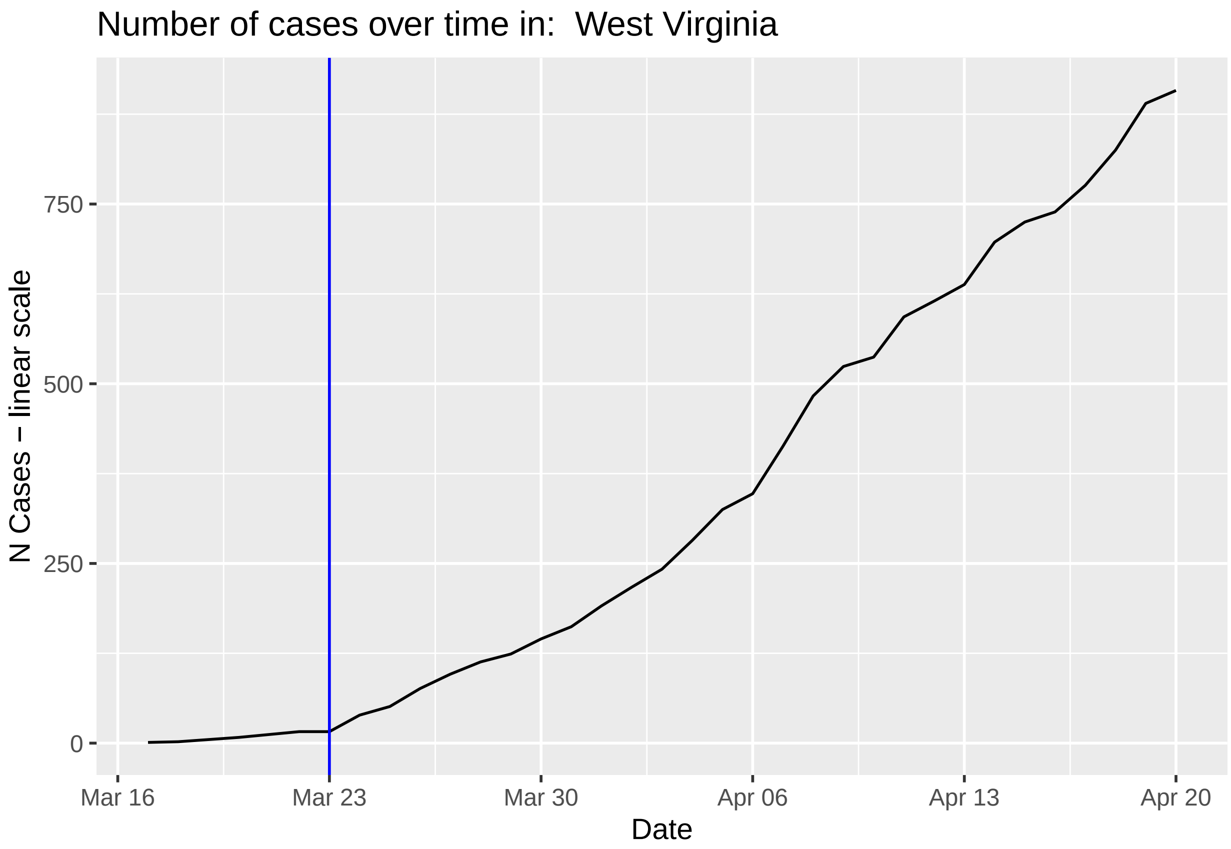
```
st <- us_counties20apr %>%
  select(c(date, state, county, fips, cases, deaths)) %>%
  filter(state == state2select)
```

```
st_cases <- st %>% select(c(date, cases)) %>%
    group_by(date) %>%
    tally(cases)
```

Plotting the state specific data

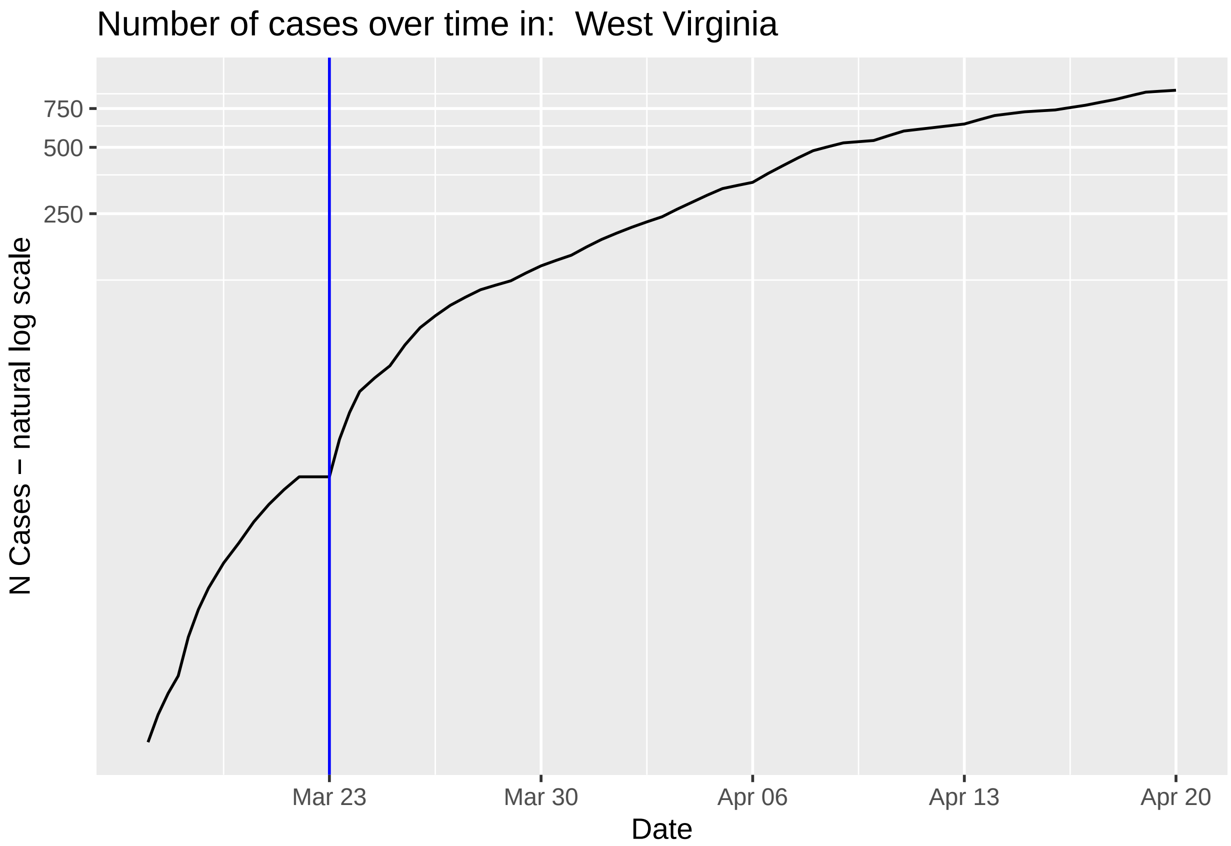
Once the data is selected and grouped by state cases a general total cases per day model can be developed. Depending on the state this should not select the NA's and have different start times. Alternative methods are possible by converting to a wide format to maintain early NA data.

```
st_cases %>%
  ggplot(aes(x=date,y=n)) +
  scale_colour_discrete(guide = FALSE) +
  geom_line()+
  geom_vline(xintercept = as.numeric(lockdown_st), color = "Blue") +
  ylab("N Cases - linear scale")+
  xlab("Date") +
  ggtitle(paste("Number of cases over time in: ", state2select))
```



Depending on total case numbers(i.e. greater than 1000-5000) a log scale maybe easier to show trends and see recent trends. If under 5000 cases total this may over-represent trends (up or down) that are only part of the standard variance of the data.

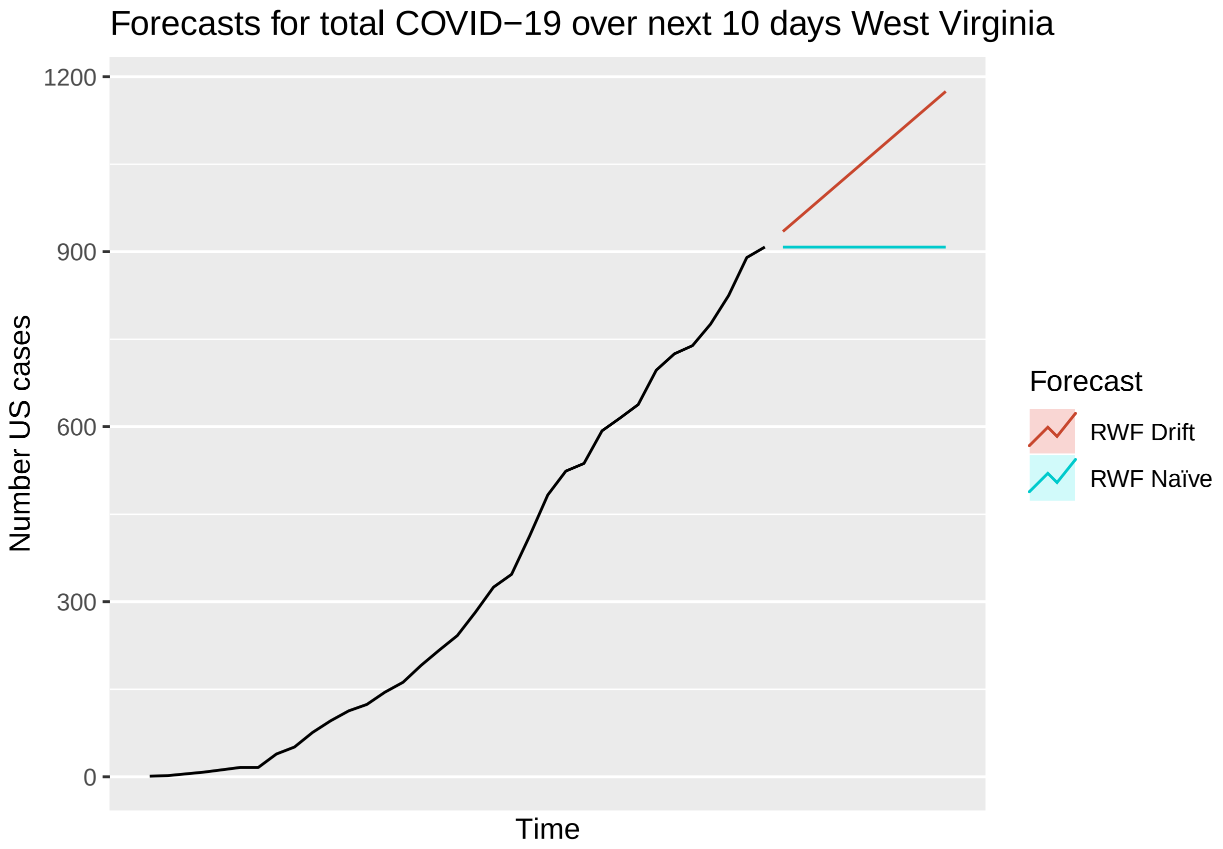
```
st_cases %>%
  ggplot(aes(x=date,y=n)) +
  scale_colour_discrete(guide = FALSE) +
  geom_line()+
  coord_trans(y="log") +
  geom_vline(xintercept = as.numeric(lockdown_st), color = "Blue") +
  ylab("N Cases - natural log scale")+
  xlab("Date") +
  ggtitle(paste("Number of cases over time in: ", state2select))
```



Basic forecast model for the next 10 days.

To do this you will need to convert the date data into a time-series using the Forecast and lubridate packages in R.

```
start_cases <- min(st_cases$date)
end_cases <- max(st_cases$date)

stcases_ts <- ts(st_cases[,2], start = c(year(start_cases), yday(start_cases)),
               end = c(year(end_cases), yday(end_cases)), frequency = 365)


autoplot(stcases_ts) +
  autolayer(rwf(stcases_ts, h=10),
    series="RWF Naïve", PI=FALSE) +
  autolayer(rwf(stcases_ts, drift=TRUE, h=10),
    series="RWF Drift", PI=FALSE) +
  ggtitle(paste("Forecasts for total COVID-19 over next 10 days", state2select)) +
  xlab("Time") +
  ylab("Number US cases") +
  guides(colour=guide_legend(title="Forecast"))
```



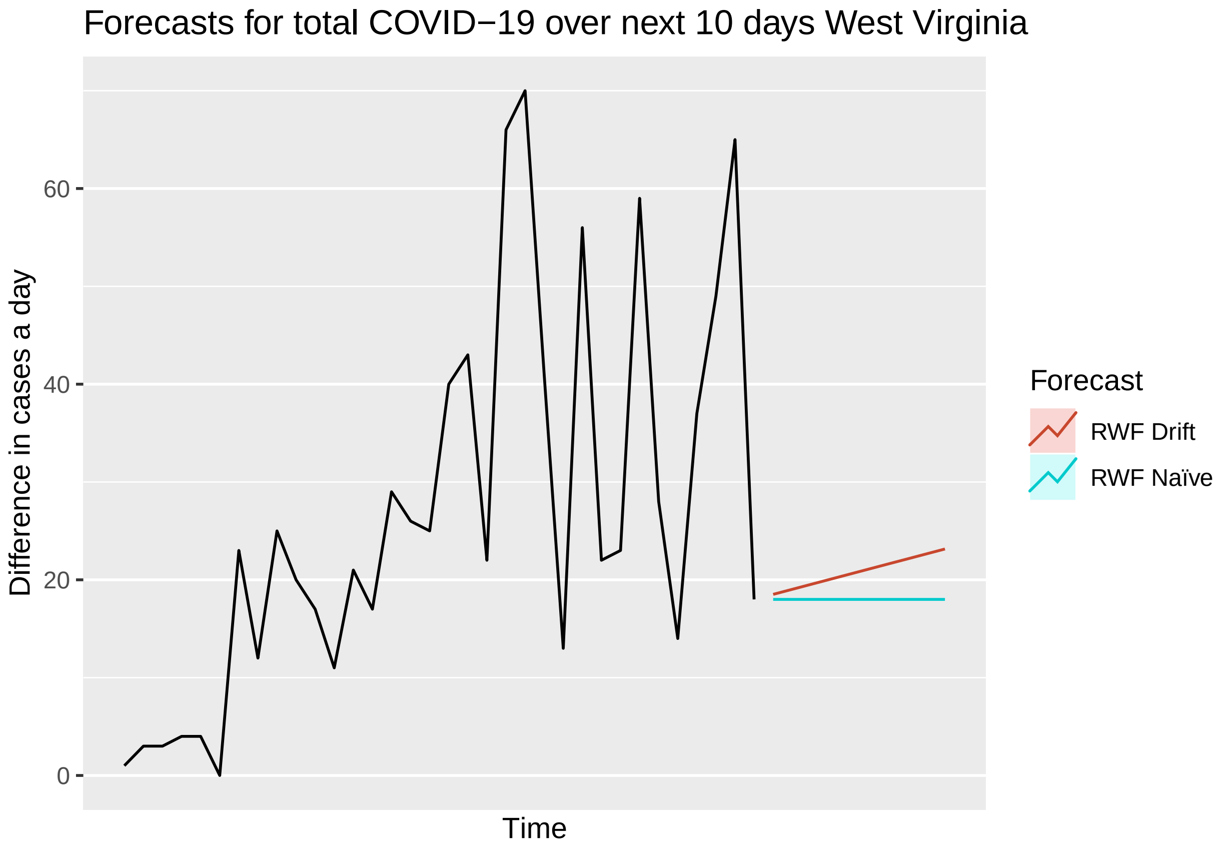
Part 2. Using linear modeling in daily difference to determine the trends

In this part it will be broken down into thre sections. First section will look at the standard model for the 10-days bofere and after a lock-down/stay in place order. Then a standard last 10-14 days compared to previous 10-14 days and both normalized to days 1 through 10 (or 14). Second part is the start of the difference model by looking at the previous 10-14 days and comparing to the time before that. This is appropriate in states that had not transistioned from lag-phase to exponetial growth phase (i.e. West Virginia). This shoud show a basic day-over-day trend. Third part is looking at the effect of the stay in place/lockdown order to determine if it had a measurable effect. NOTE: in states with minimal/no time in exponential growth this may not be an accurate measure and recommend that Part 2, section B be used (ie. comparing two different time frames).

Part 2. Section A

Comparing the last n days versus previous time frame using standard case number and difference modeling.

```
autoplot(diff(stcases_ts)) +
  autolayer(rwf(diff(stcases_ts), h=10),
    series="RWF Naïve", PI=FALSE) +
  autolayer(rwf(diff(stcases_ts), drift=TRUE, h=10),
    series="RWF Drift", PI=FALSE) +
  ggtitle(paste("Forecasts for total COVID-19 over next 10 days", state2select)) +
  xlab("Time") +
  ylab("Difference in cases a day") +
  guides(colour=guide_legend(title="Forecast"))
```

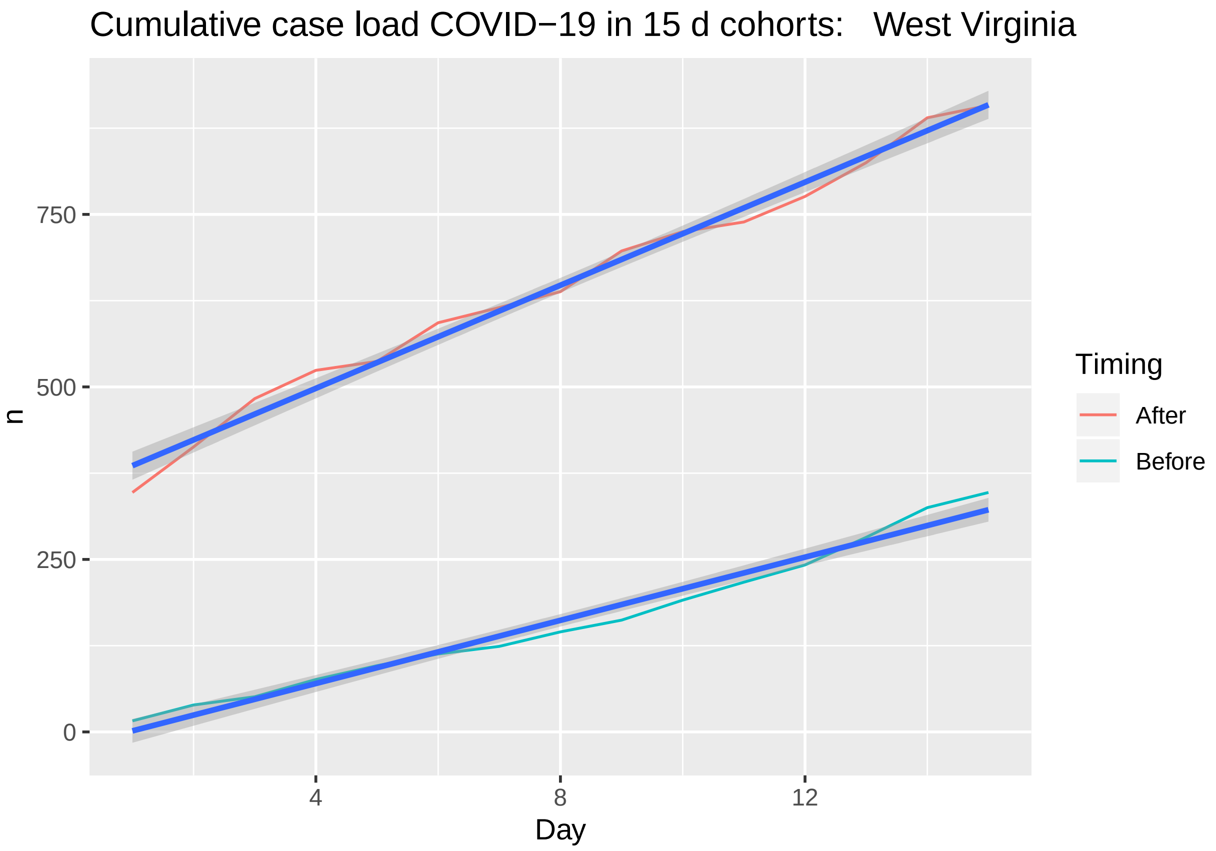


Above shows a basic difference model. If it appears that a form of stasis (i.e. random variation around an estimated mean) was achieved then it maybe possible to use ARIMA modeling for a better determination of actual Difference(cases).

```
### This probably needs to be made into a function but this is the simpler way

diff_window <- 14 # Setting the number of days for 14

before_start <-(end_cases - (2*diff_window))

before_end <- (end_cases - diff_window)

diff_tib <- tibble(Day = c(1:diff_window))

### Adding the different rows
```

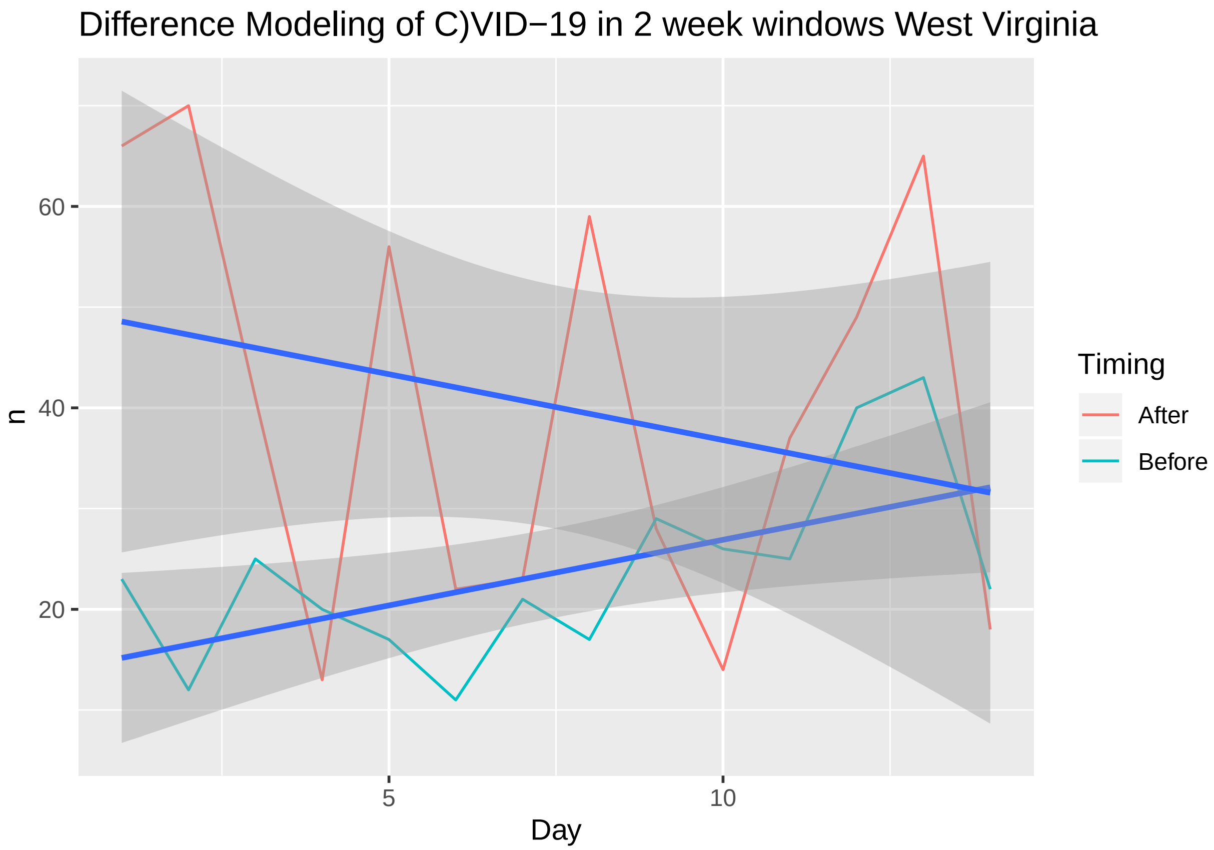
```
before_window <- st_cases %>%
  filter(date >= before_start & date <= before_end)


after_window <- st_cases %>%
  filter(date >= before_end & date <= before_end + diff_window)

std_tib <- tibble(Day = c(1:15)) %>%
  mutate(Before = before_window$n) %>%
  mutate(After = after_window$n) %>%
  gather(key = Timing, value = n, - Day)


diff_tib <- diff_tib %>%
  mutate(Before = diff(before_window$n)) %>%
  mutate(After = diff(after_window$n)) %>%
  gather(key = Timing, value = n, - Day)
```

```
ggplot(data = std_tib, aes(x = Day, y = n)) +
  geom_line(aes(colour=Timing)) +
  geom_smooth(data = std_tib[1:15,], method = "lm") +
  geom_smooth(data = std_tib[16:30,], method = "lm") +
  ggtitle(paste("Cumulative case load COVID-19 in 15 d cohorts:  ", state2select))
```



After the days are set now its time to plot.

```r
ggplot(data = diff_tib, aes(x = Day, y = n)) +
  geom_line(aes(colour=Timing)) +
  geom_smooth(data = diff_tib[1:14,], method = "lm") +
  geom_smooth(data = diff_tib[15:28,], method = "lm") +
  ggtitle(paste("Difference Modeling of C)VID-19 in 2 week windows", state2select))
```



Checking the Linear Model of the before and after.

```r
before_lm <- lm(n ~ Day, data = diff_tib %>% filter(Timing == "Before"))


summary(before_lm)
```

```
##
## Call:
## lm(formula = n ~ Day, data = diff_tib %>% filter(Timing == "Before"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.688  -4.202  -1.446   6.267  12.189
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.8681     4.3140   3.215  0.00743 **
## Day           1.3033     0.5067   2.572  0.02444 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.642 on 12 degrees of freedom
## Multiple R-squared:  0.3554, Adjusted R-squared:  0.3017
## F-statistic: 6.617 on 1 and 12 DF,  p-value: 0.02444
```

```r
after_lm <- lm(n ~ Day, data = diff_tib %>% filter(Timing == "After"))


summary(after_lm)
```

```
##
## Call:
## lm(formula = n ~ Day, data = diff_tib %>% filter(Timing == "After"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.648 -16.687  -1.725  16.775  32.121
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   49.879     11.713   4.259  0.00111 **
## Day           -1.308      1.376  -0.951  0.36054
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.75 on 12 degrees of freedom
## Multiple R-squared:  0.07003,    Adjusted R-squared:  -0.007462
## F-statistic: 0.9037 on 1 and 12 DF,  p-value: 0.3605
```

Take home message is that an effective stay in place order should be judged by a reduction in day-over-day case load of around 10% the difference in cases. This would mean that WV would need to have a difference of around 4-cases/day when comparing the before and after. However, WV has only about a 2.6 to 2.7 case differnerce over the last 14 days compared to the previous. There is also an increase in varablity that indicates that either (i)stay in place order is not being followed, (ii) there is a chance that the outside the local community infections are occuring or (iii) the positive cases are based upon different testing technologies ad could be adding extra variance. It is very hard to tell because the difference between cases in the last 14 days has dramatically increased from the previous 14 day. Basicaly the last 14-days are showing a slight reduction but the variance between days is making any inference impossible and points to the community not following the stay in place order. While not alarming this is worrysome.