ANN DESPAIN
5075
Computer Project 2, Completed in R

This report will examine the daily percent increase of covid cases in Utah during the period of March 6, 2020 through April 11, 2021. Source of data, collected at about 9:45 pm Mountain time each day as reported by The New York Times[1] and https://coronavirus.utah.gov/.

1: Import and Smooth the data

Here, I imported the data, renamed the second columns, and reformatted the date data.

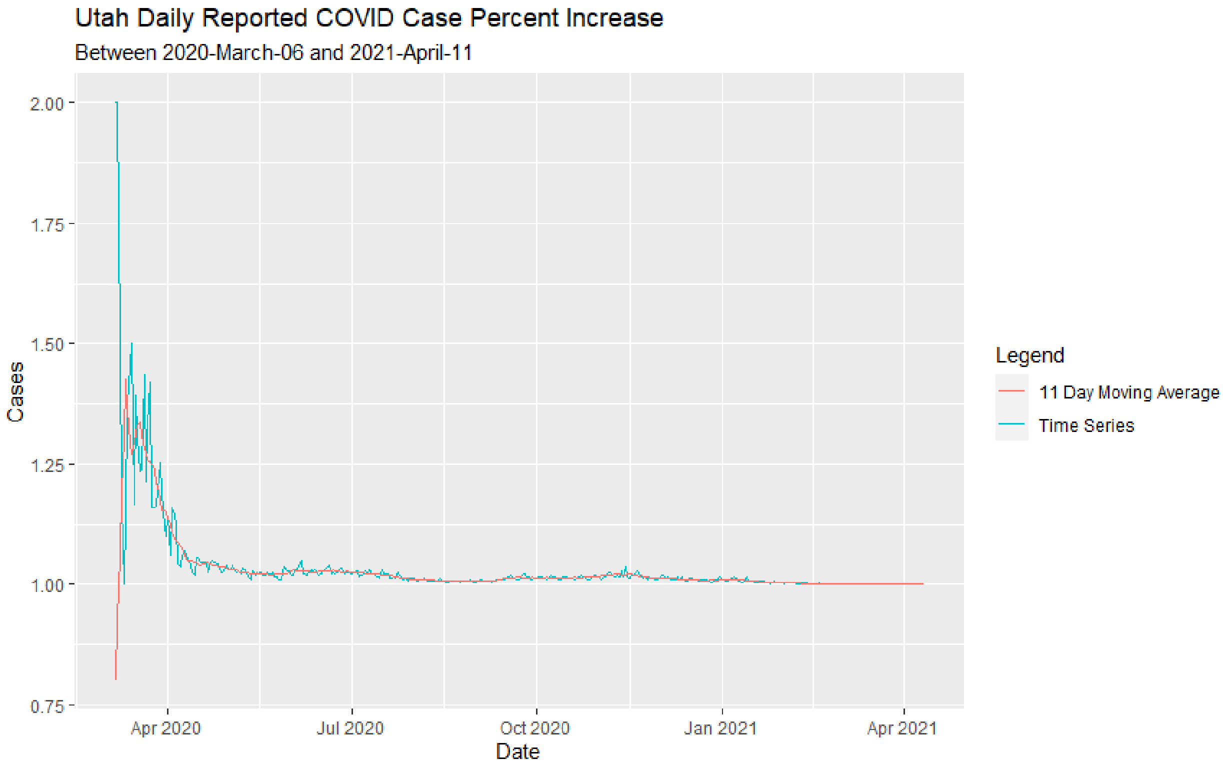
```
names(COVIDdata)[1] <- "date"
names(COVIDdata)[2] <- "x"
names(COVIDdata)[3] <- "y"
names(COVIDdata)[4] <- "V"
names(COVIDdata)[5] <- "i"
names(COVIDdata)[6] <- "US"
names(COVIDdata)[7] <- "UT"
names(COVIDdata)[8] <- "USP"
names(COVIDdata)[9] <- "UTP"

dates <- COVIDdata$date
dates <- as.Date(dates, format = "%Y-%m-%d")
COVIDdata$date2 = dates
```

To smooth the data, I used a rolling mean function, as suggested. To compensate for the first five and last five entries that do not have 11 observations to sum, I created a new data field, COVIDdata2 for $\tilde{x}_i$ that did not include those entries. Because of the volatile nature of the data, I was interested to see what a smaller rolling window would look like, so I also made provisions for that comparison.

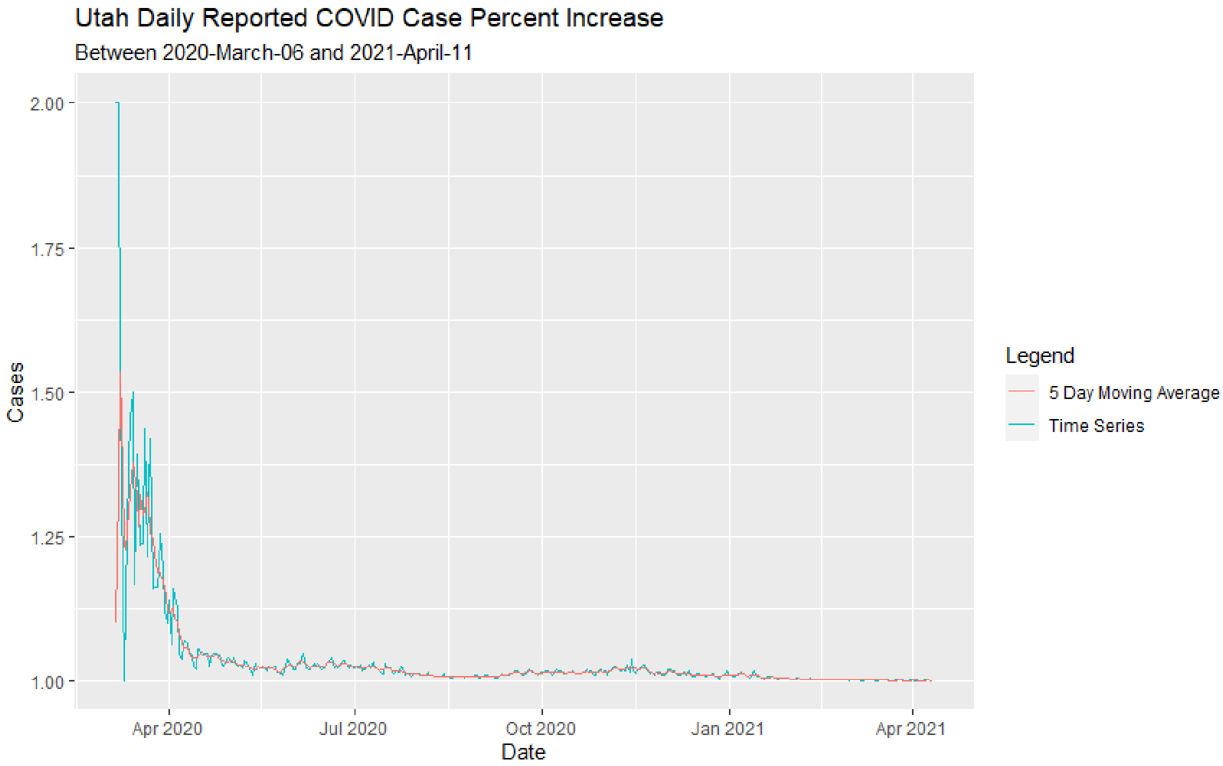
```
utp_tilde <- rollmean(COVIDdata$UTP , k=11, fill=0)
COVIDdata$utp_tilde <- utp_tilde

utp_tilde5 <- rollmean(COVIDdata$UTP , k=5, fill=0)
COVIDdata$utp_tilde5 <- utp_tilde5

COVIDdata2 <- COVIDdata[c(6:407),c(1:13)]
```

Here, we see the original data and the smoothed 11-day averaged data together.

```
ggplot(COVIDdata2) +
  geom_line(aes(date2,UTP, color="Time Series")) +
  geom_line(aes(date2,utp_tilde,
              color="11 Day Moving Average"
            )) +

  labs(title = "Utah Daily Reported COVID Case Percent Increase",
       subtitle = "Between 2020-March-06 and 2021-April-11",
       y = "Cases",
       color = "Legend",
  x = "Date")
```



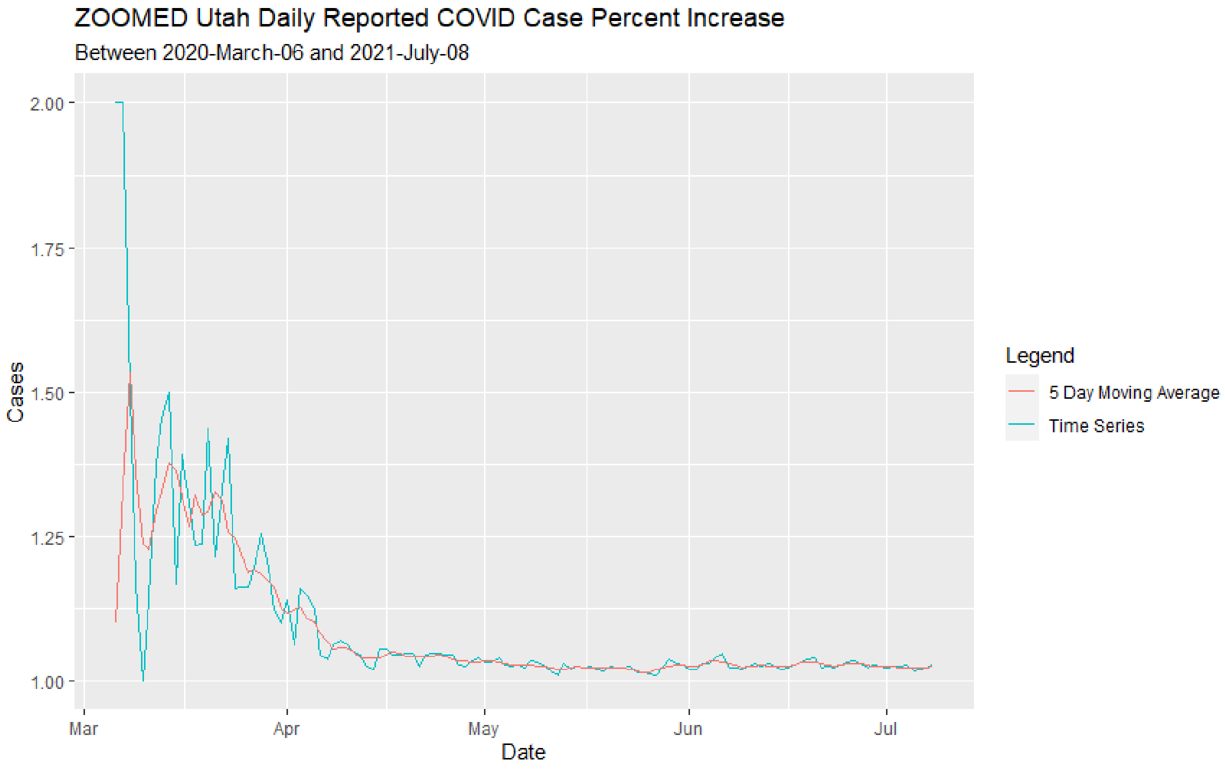
And the 5 day Moving Average:



```
ggplot(COVIDdata2) +
  geom_line(aes(date2,UTP, color="Time Series")) +
  geom_line(aes(date2,utp_tilde,
                color="11 Day Moving Average"
              )) +

  labs(title = "Utah Daily Reported COVID Case Percent Increase",
       subtitle = "Between 2020-March-06 and 2021-April-11",
       y = "Cases",
       color = "Legend",
       x = "Date")
```

Because we see that the daily case percent increase steadies out and decreases over time (due to community isolation efforts and immunity that we will discuss later), I have zoomed in to look at the more volatile period at the beginning of the time period to get a better look at what is happening, there.

```
COVIDdata4 <- COVIDdata[c(6:87),c(1:13)]
ggplot(COVIDdata4) +
  geom_line(aes(date2,UTP, color="Time Series")) +
  geom_line(aes(date2,utp_tilde5,
                color="5 Day Moving Average"
  )) +

  labs(title = "ZOOMED Utah Daily Reported COVID Case Percent Increase",
       subtitle = "Between 2020-March-06 and 2021-July-08",
       y = "Cases",
       color = "Legend",
       x = "Date")
```



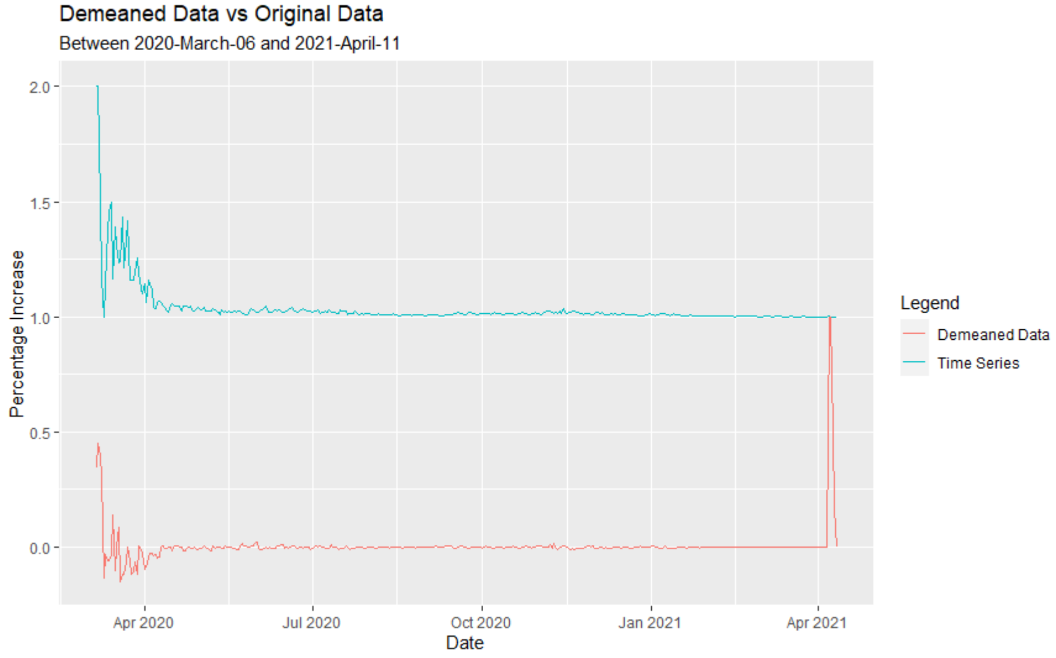
The smoothed moving average shows the general path of the original data without the extremes. Initially, the percentage increase in cases is at two percent. We see that there is a decreasing trend with the exception of the very volatile first forty or so days. We see that the data during that initial period is generally centered around 1.35, and then tapers off to hover just above one percent until the time of this writing (1.001% daily increase as of 2021-04-18).

This zoomed-in look at the data has me a little bit concerned. The shape, roughly a cone spread out to the left, might suggest that we have a heteroscedastic situation that is not well-suited for regression. But we will proceed and see if analyzing the residuals can confirm this as an issue.

2: Demean the Data

We define $\hat{x}_i = x_i - \bar{x}_n$. Essentially, we are subtracting the rolling average from our data (as it is not stationary). Here is the plotted series of the demeaned data, $\hat{x}_i$ AND the original data.

```
utp_hat <- (COVIDdata2$UTP-utp_tilde)
UTDemeaned <- utp_hat[c(6:407)]
UTDemeanedM <-as.matrix(DemeanedUT)

COVIDdata3 <- COVIDdata[c(6:407),c(1:12)]
COVIDdata3$UTDemeaned = UTDemeaned

ggplot(COVIDdata3) +
  geom_line(aes(date2,UTP, color="Time Series")) +
  geom_line(aes(date2,UTDemeaned,
                color="Demeaned Data", )) +

  labs(title = "Demeaned Data vs Original Data",
       subtitle = "Between 2020-March-06 and 2021-April-11",
       y = "Percentage Increase",
       color = "Legend",
       x = "Date")
```



We see that the data is now centered around zero. The demeaned data strangely reveals an outlier spike at the end of the time period. This cannot be confirmed by the original data or the rolling mean values there, and I am unsure why it appears. In an effort to keep the most current data for the immunization analysis I did later, I have let it remain.

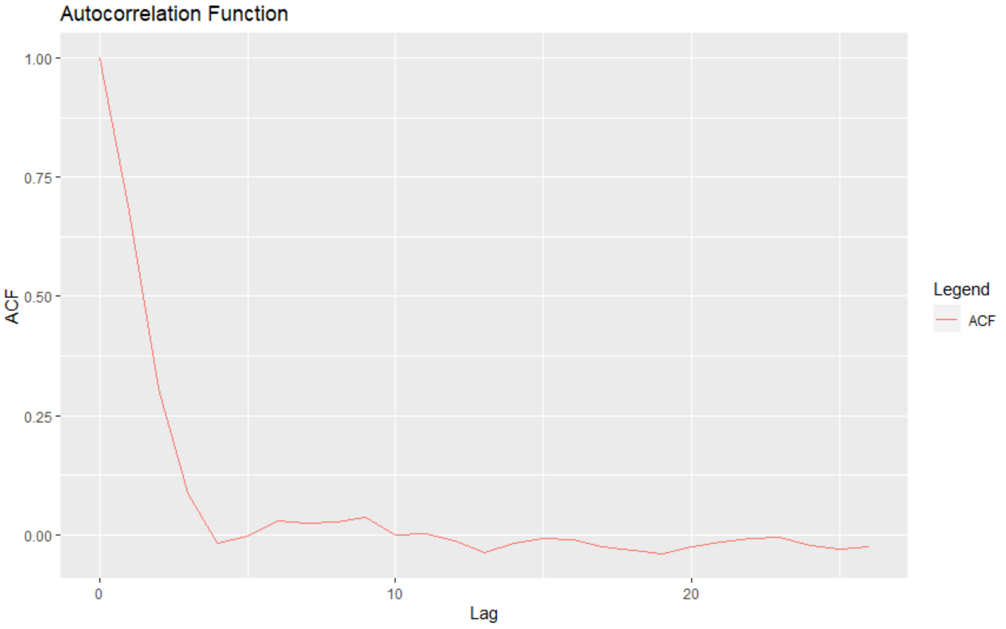
3: Plot ACF and PACF

We now use the demeaned data to plot the Auto Correlation Function (ACF) and the Partial Auto Correlation Function (PACF).

```
ACF <- acf(UTDemeaned,plot=TRUE)
PACF <- pacf(UTDemeaned,plot=TRUE)
ACF_df <- data.frame("acf_lag" <- ACF$lag,
                     "acf" <- ACF$acf)
names(ACF_df)[1] <- "lag"
names(ACF_df)[2] <- "acf"
PACF_df <- data.frame("pacf_lag" <- PACF$lag,
                      "pacf" <- PACF$acf)
names(PACF_df)[1] <- "lag"
names(PACF_df)[2] <- "pacf"
```

The ACF describes how well the present value of the series is related with its past values.

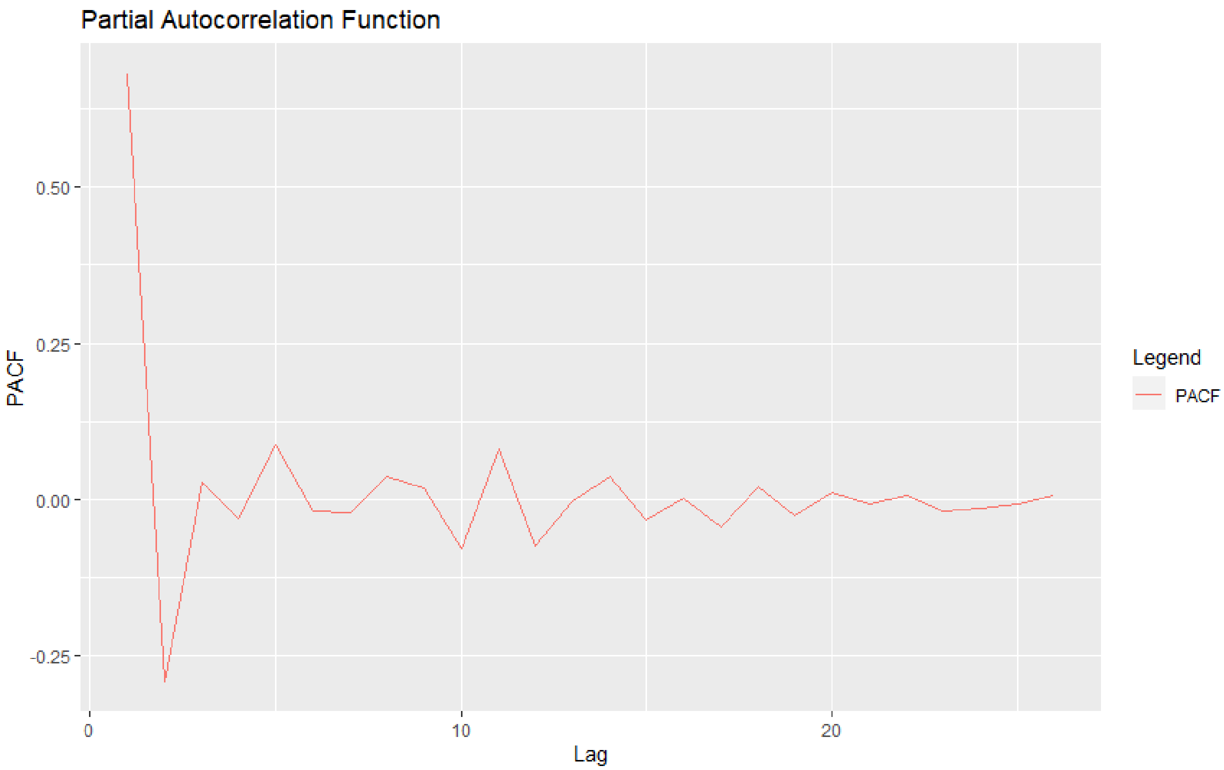
```
ggplot(ACF_df) +
  geom_line(aes(lag,acf,color="ACF")) +

  labs(title = "Autocorrelation Function",
       y = "ACF",
       color = "Legend",
       x = "Lag")
```



We see that with zero lag, the value is one, as expected, with a sudden decrease afterward. This would suggest that the present values are more relevant than the past values. Note that Lag 1 = 0.6797.

Taking a look at the PACF:

```
ggplot(PACF_df) +
  geom_line(aes(lag,pacf,color="PACF")) +

  labs(title = "Partial Autocorrelation Function",
       y = "PACF",
       color = "Legend",
       x = "Lag")
```



The PACF does not consider the data that lies within our error bands, so instead of finding correlations of the present with lags like the ACF, it finds correlation of the residuals. Therefore, if there is any hidden information in the residual data which can be modeled by the next lag, we might get a good correlation and we would keep that next lag as a feature while modeling. As we have a high initial value (PACF lag 1 = 0.6797, the same as the lag 1 value for ACF) followed by a drastic drop, we would assume that the relevant information is found not with past residuals where the values of the PACF are close to zero, but with the more current information. This would suggest that an AR(1) would be a good fit.

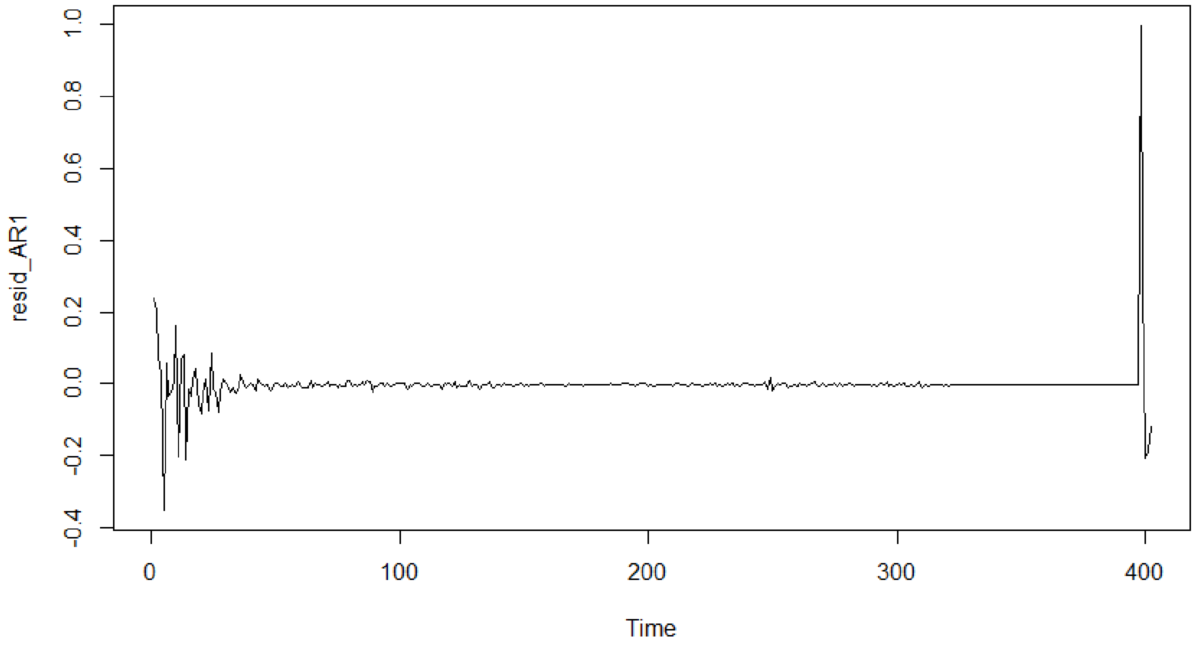
4: Fit an AR(1) model to $\hat{x}_t$ and compute residuals and s^2 as defined by the project.

To fit an AR(1), we use an ARMA (p,q) model in R, recalling that we use the ACF and PACF pair to help identify the orders p and q. The ACF tails off for AR(p) and cuts off after a certain lag MA(q) while the PACF cuts off the AR(p) after lag p and tails off an MA (q).

$$\hat{\epsilon}_i = \hat{x}_i - \hat{\varphi}\hat{x}_{i-1}$$

$$s_1^2 = \frac{1}{n-2}\sum_{i=2}^{n}\hat{\epsilon}_i^2$$

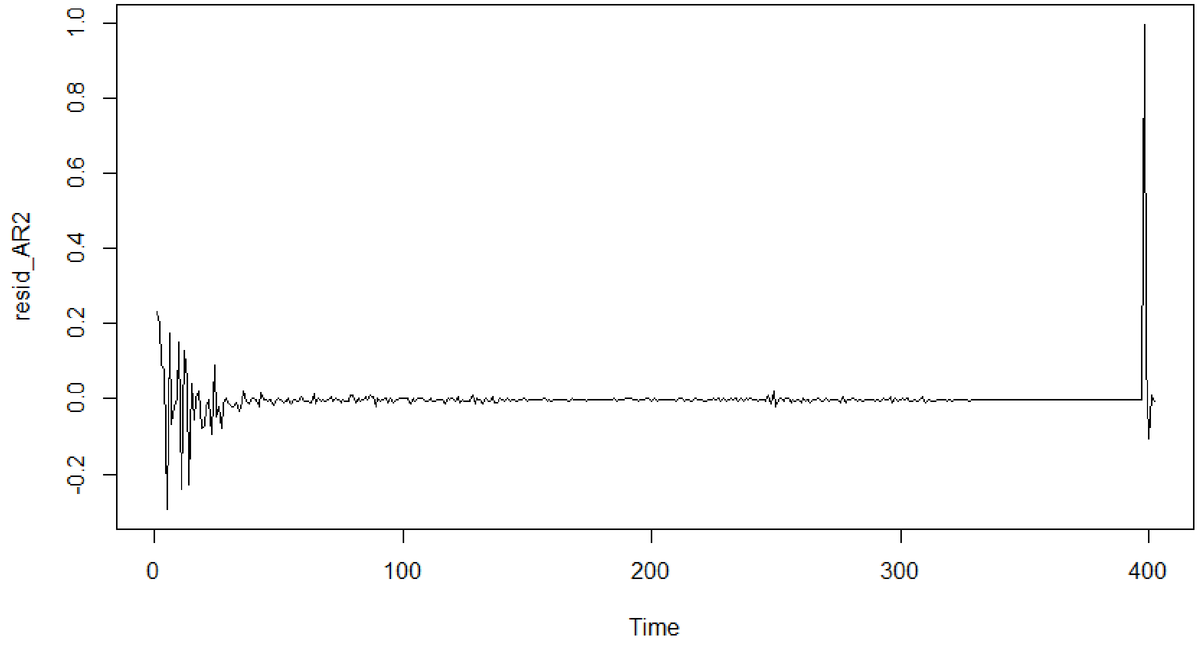
```
      AR1_fit <- arima(UTDemeaned,order = c(1,0,0))
      AR1_fit

      Call:
      arima(x = UTDemeaned, order = c(1, 0, 0))

      Coefficients:
               ar1  intercept
            0.7041     0.0078
      s.e.  0.0365     0.0105

sigma^2 estimated as 0.003917:  log likelihood = 543.25,  aic = -1080.51


resid_AR1 <- AR1_fit$resid
plot(resid_AR1)
```



There is that wonky outlier, again, showing up in the residual data.

Similarly, we fit an AR(2) model to $\hat{x}_i$ and compute s^2.

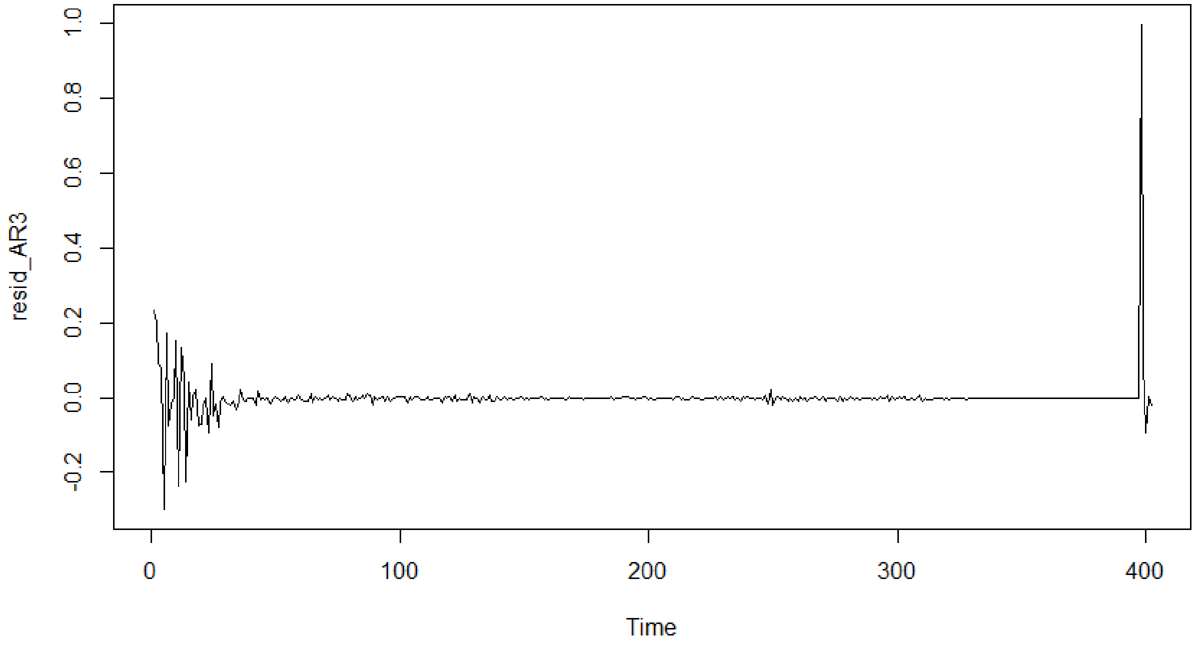
```
      AR2_fit <- arima(UTDemeaned,order = c(2,0,0))
      AR2_fit
Call:
arima(x = UTDemeaned, order = c(2, 0, 0))

Coefficients:
         ar1      ar2  intercept
      0.8962  -0.2921     0.0060
s.e.  0.0480   0.0499     0.0076

sigma^2 estimated as 0.003609:  log likelihood = 559.68,  aic = -1111.37
resid_AR2 <- AR2_fit$resid
plot(resid_AR2)
```



Finally, we fit an AR(3) model to $\hat{x}_i$ and compute s^2.

```
AR3_fit <- arima(UTDemeaned,order = c(3,0,0))
AR3_fit

Call:
arima(x = UTDemeaned, order = c(3, 0, 0))

Coefficients:
         ar1      ar2     ar3  intercept
      0.9016  -0.3094  0.0219     0.0062
s.e.  0.0498   0.0662  0.0550     0.0078

sigma^2 estimated as 0.003607:  log likelihood = 559.76,  aic = -1109.53
resid_AR3 <- AR3_fit$resid
```

```
plot(resid_AR3)
```



A residual plot is typically used to find problems with regression. Some data sets are not good candidates for regression, including:

- Heteroscedastic data (cone-shaped points at widely varying distances from the line).
- Data that is non-linearly associated.
- Data sets with outliers.

These problems are more easily seen with a residual plot than by looking at a plot of the original data set. Ideally, residual values should be equally and randomly spaced around the horizontal axis.

It is not a good thing for residual plots if it is easy to predict where other data points might fall. These residual plots may indicate a problem. There can be several reasons why regression isn't suitable. From my research, it could be something simple, like:

- Missing higher-order variable terms that explain a non-linear pattern.
- Missing interaction between terms in your existing model.
- Missing variables.

Given the shape of the data and the shape of the residuals, I'm prone to say that this data set is not a great candidate for regression.

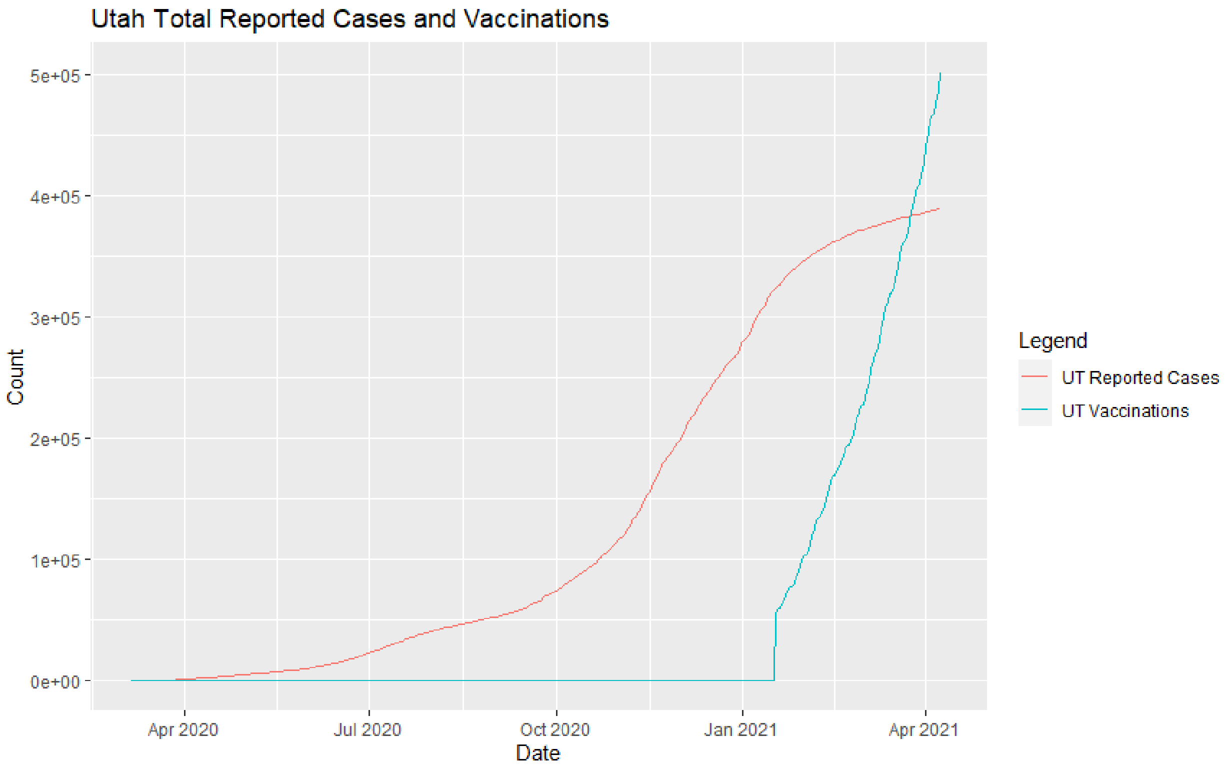
Already invested and forging ahead as if it were, there is not any detectable visible difference in the graphs of the residual data. Analyzing the coefficients we received by fitting the data to the AR(1), AR(2), and AR(3) models, along with what we know from the ACF and PACF, the AR(1) model appears to be the best fit. With coefficients from the AR(1) of 0.7041, we see these values are very close to our ACF and PACF lag 1 values of 0.6797.

5: Further Analysis

Given the current climate and the world's interest in the subject, I went on to further analyze the data. It is clear that the daily percentage increase has decreased to a steady rate of a little over one percent. This is due to the world's efforts to slow the spread of the virus, as well as immunity. Immunity has come from both vaccines and the natural immunity received by those who have recovered from the virus.

Here, we look at the total reported cases in Utah as well as the Vaccinations administered.

```
ggplot(COVIDdata2) +
  geom_line(aes(date2,y, color="UT Reported Cases"))+
  geom_line(aes(date2,V, color="UT Vaccinations"))+

  labs(title = "Utah Total Reported Cases and Vaccinations",

       y = "Count",
       color = "Legend",
       x = "Date")
```

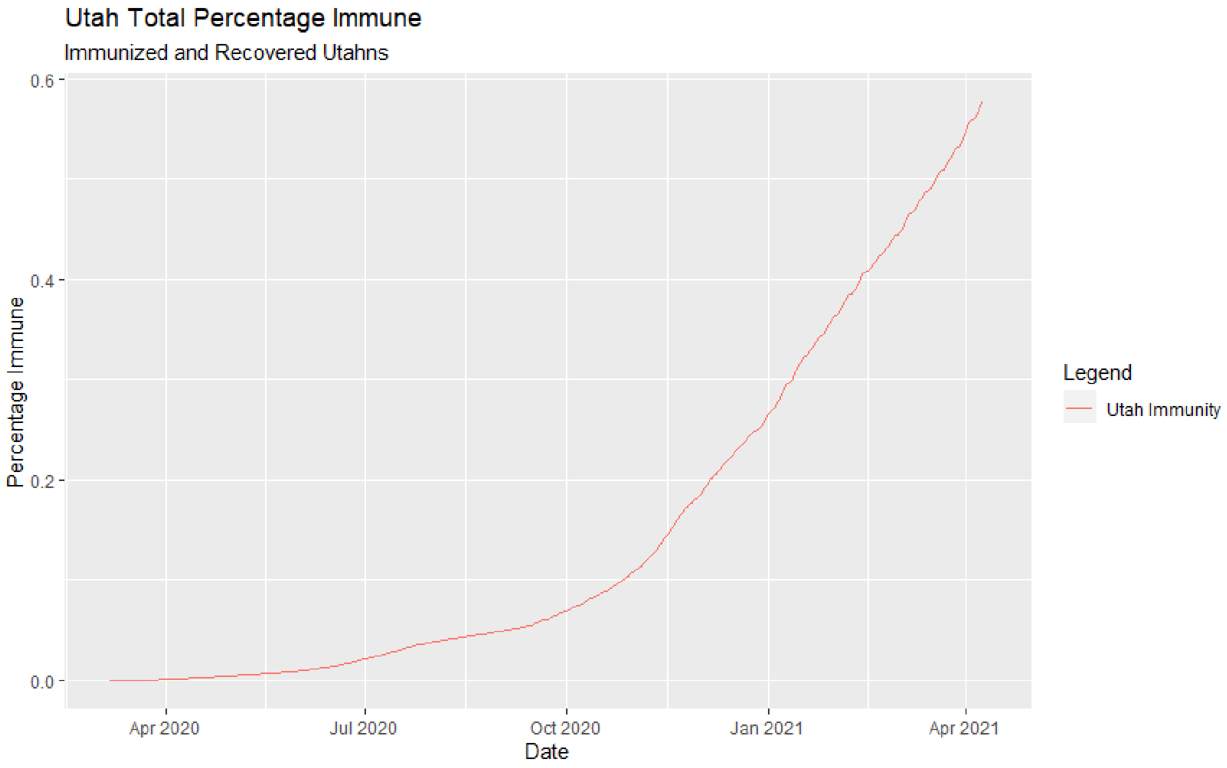


Utah's first reported case was March 5, 2020. Our first vaccines administered were on January 17th, 2021, by which date there had been 323,837 total cases reported in Utah. As expected, as vaccinations have begun and are steadily increasing, the reported cases are tapering. There are almost twice as many Utahns vaccinated than have had the virus! As of April 17, 2021, there have been 392,509 reported cases in Utah, and 584,492 fully vaccinated Utahns.

Using this information, I have calculated the percentage of COVID-immune people in Utah. Again, using the data reported by the NYTimes and Utah Corona Virus Response Team, this model assumes:

- 50% of people who got just a first dose are immune,
- 95% of people who got both first and second doses are immune,
- 70% of people who got the J&J vaccine are immune, and
- 100% of the people who already had covid are immune, AND
- It accounts for the chance that someone receiving the vaccine might already have recovered from Covid. It selects that probability based on what percentage of the population has gotten covid, a number which I assume is three times the count of people who had a positive test result.

```
ggplot(COVIDdata2) +
  geom_line(aes(date2,i, color="Utah Immunity"))+


labs(title = "Utah Total Percentage Immune",
     subtitle = "Immunized and Recovered Utahns",
     y = "Percentage Immune",
     color = "Legend",
     x = "Date")
```



As of April 17, 2021, according to this model, 61.9% of Utahns are immune to COVID-19.

Notes:

R libraries required to run the code:

```
require(zoo)
require(tidyverse)
require(hrbrthemes)
require(lubridate)
```

Utah COVID numbers come from: https://coronavirus.utah.gov/
That are also reported to:
[1]https://www-nytimes-com.cdn.ampproject.org/v/s/www.nytimes.com/interactive/2020/world/coronavirus-maps.amp.html?usqp=mq331AQFKAGwASA%3D&amp_js_v=0.1#us