DS310 COVID Data Project Statistical Methods

Reetom Gangopadhyay, Gui Marques, Ethan Chang, Dhruv Gandhi, Neeza Singh, Austin Bellis

Data Cleaning

Read in data and subset for ISO3=NZL and SWE

```
df <- read.csv("confirmed.csv")

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
nzl_data <- df %>% filter(ISO3 == "NZL")
swe_data<- df %>% filter(ISO3 == "SWE")
```

This box creates "Date" variables in one column for both Sweden and New Zealand along with a "CasesPerDay" variable which counts number of new cases per day.

```
library(dplyr)

nzl_data$Date <- as.Date(paste(nzl_data$Year, nzl_data$Month, nzl_data$Day, sep = '-'), format = '%Y-%B-
nzl_data <- nzl_data %>% arrange(Date)

nzl_data <- nzl_data %>%
  group_by(ISO3) %>%
  mutate(CasesPerDay = Sum.of.Confirmed - lag(Sum.of.Confirmed, default = 0)) %>%
  ungroup()

swe_data$Date <- as.Date(paste(swe_data$Year, swe_data$Month, swe_data$Day, sep = '-'), format = '%Y-%B-
swe_data <- swe_data %>% arrange(Date)

swe_data <- swe_data %>%
  group_by(ISO3) %>%
  mutate(CasesPerDay = Sum.of.Confirmed - lag(Sum.of.Confirmed, default = 0)) %>%
  ungroup()
```

```r
#head(nzl_data)
#head(swe_data)
```

Time Series Methods:

Unit Root Test

The use of Unit Root tests will tell us whether we need to difference the data when creating ARIMA models.

```r
library(tseries)
kpss.test(nzl_data$CasesPerDay)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  nzl_data$CasesPerDay
## KPSS Level = 0.7446, Truncation lag parameter = 5, p-value = 0.01
```

```r
library(tseries)
kpss.test(swe_data$CasesPerDay)
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  swe_data$CasesPerDay
## KPSS Level = 4.63, Truncation lag parameter = 5, p-value = 0.01
```

Discussion:

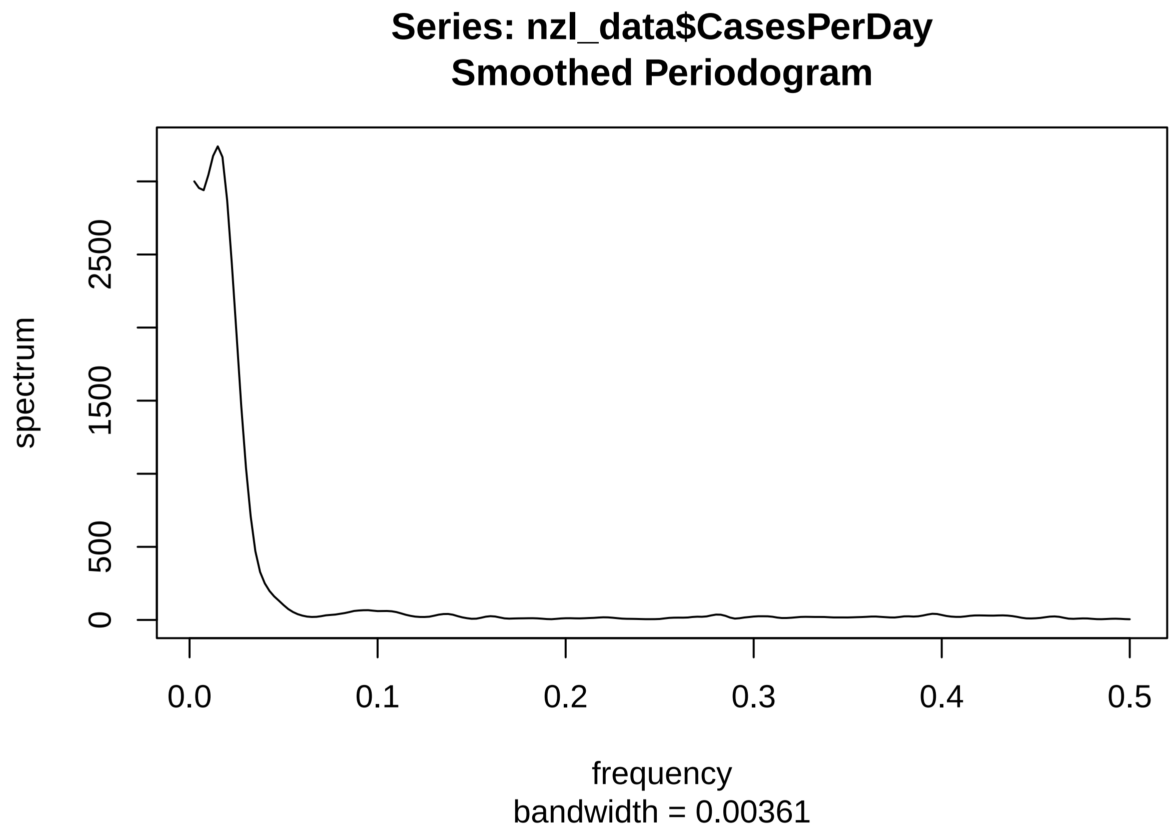
H_0: The series does not contain a unit root

H_A: The series contains a unit root

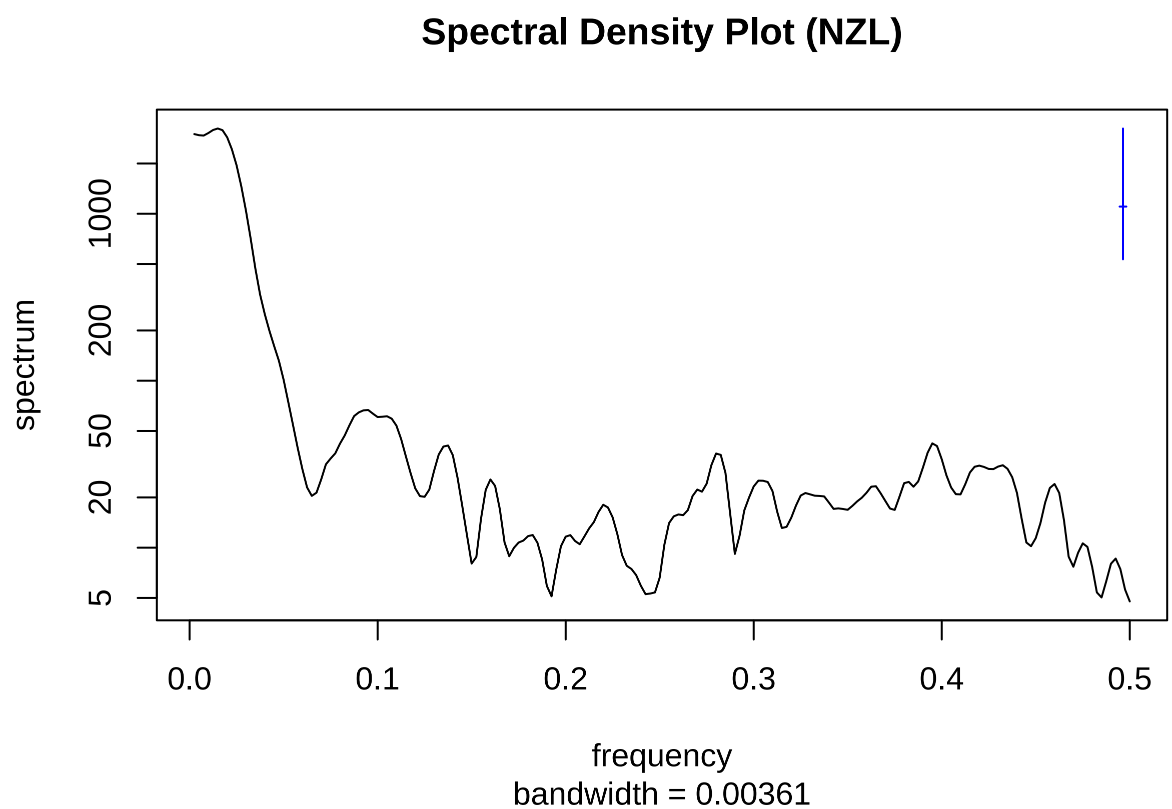
Reject null for both datasets meaning that you need to difference the data to remove unit root which invalidates forecasting results.

Spectral Analysis:

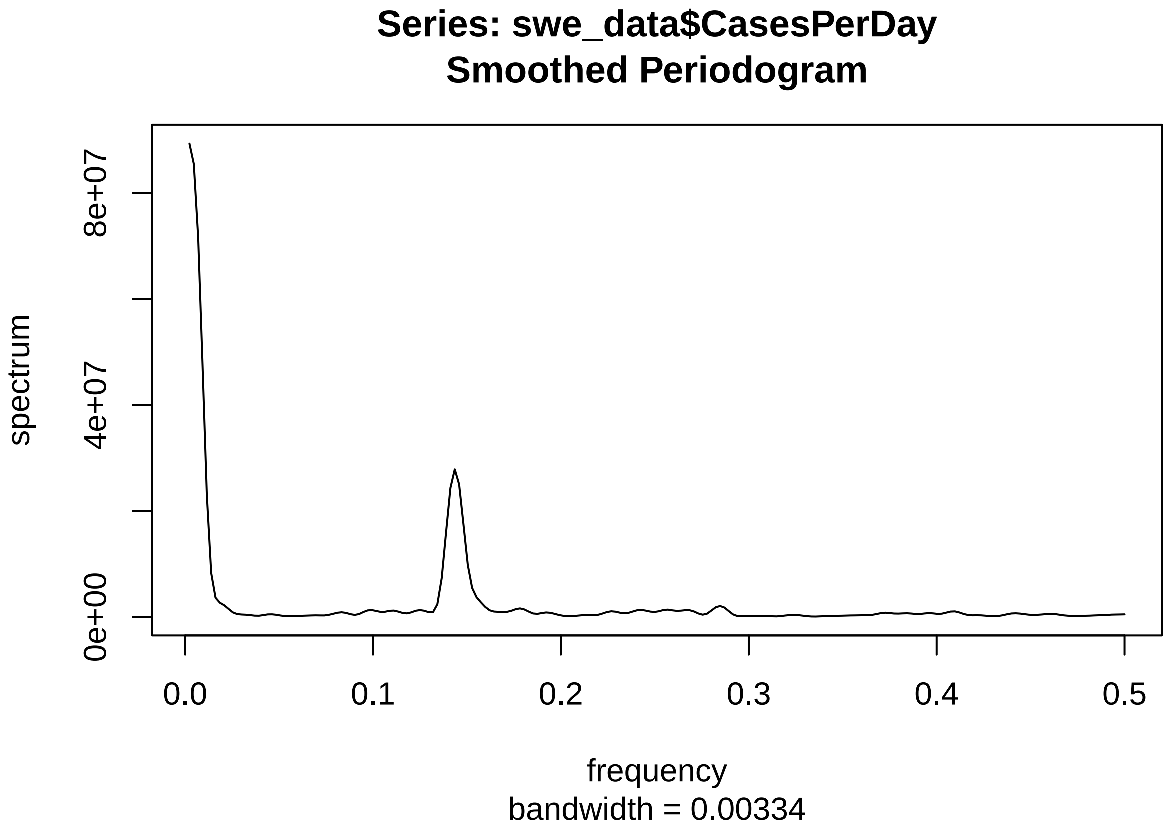
```r
spec_result <- spec.pgram(nzl_data$CasesPerDay, taper = 0, log = "no", span =c(3,5))
```



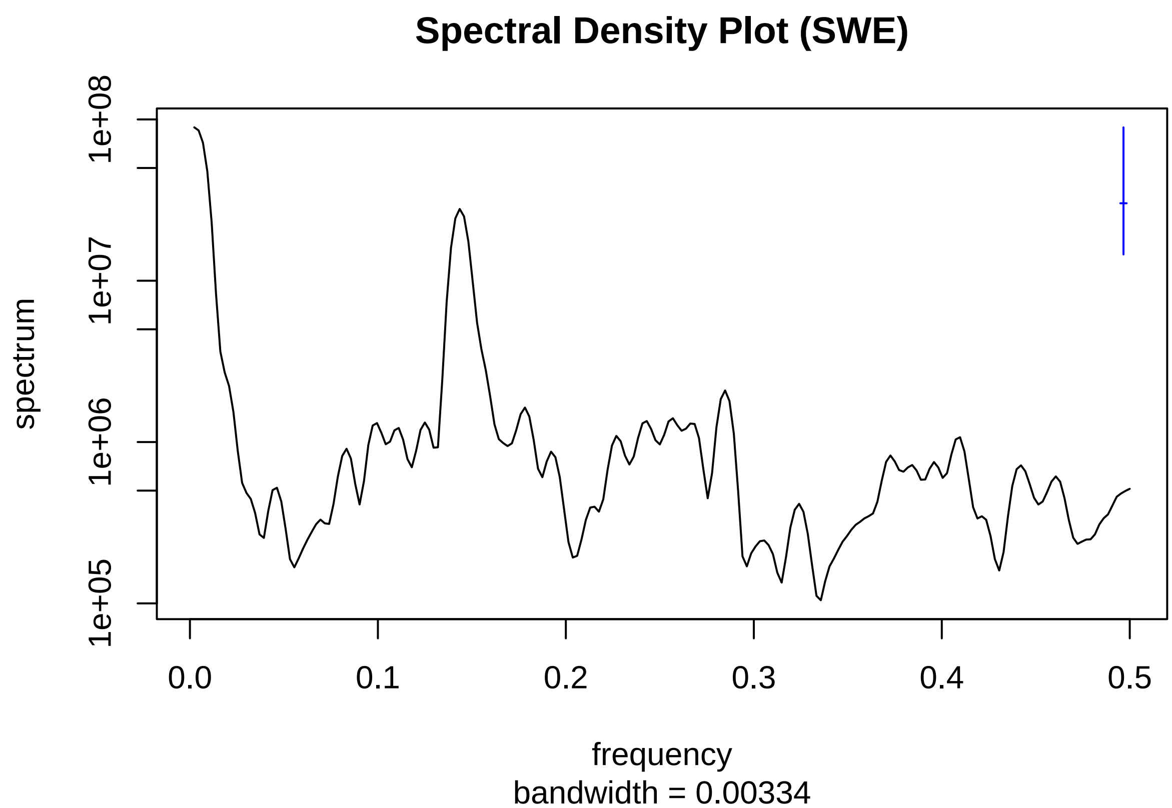

```r
plot(spec_result, main = "Spectral Density Plot (NZL)")
```



```r
spec_resultA <- spec.pgram(swe_data$CasesPerDay, taper = 0, log = "no", span =c(3,5))
```




```r
plot(spec_resultA, main = "Spectral Density Plot (SWE)")
```




```r
period <- 1/0.14

cat("Period of Sweden series:",period)
```

```
## Period of Sweden series: 7.142857
```

Discussion:

The NZL data shows no dominant frequency, while the Sweden series shows a frequency of 0.14, which translates toa period of $1/0.14 = 7.1$. In other words, Sweden shows weekly seasonality.

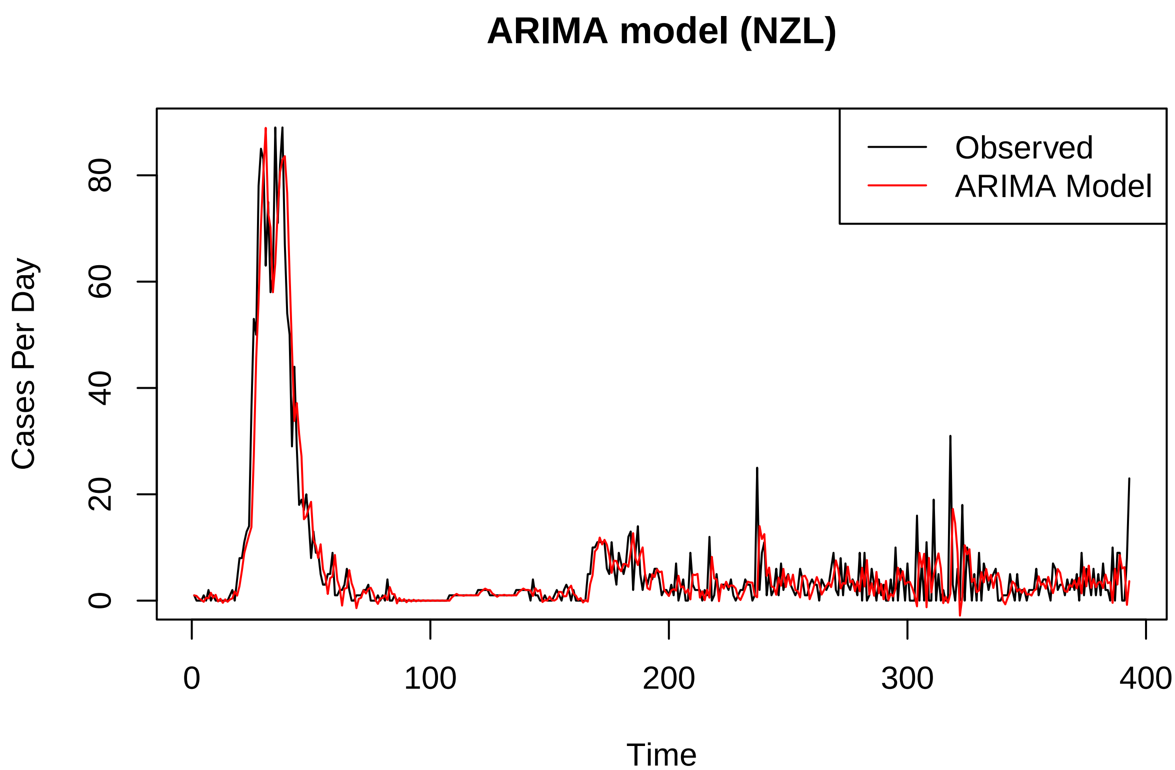
ARIMA/SARIMA Models and Forecasts:

We can fit ARIMA or SARIMA models using the auto.arima() function. Using the spectral analysis from before, we see that Sweden has a weekly period, so we can fit a "Seasonal ARIMA" (SARIMA) model.

Note: All models are fit on Cases per day, NOT sum of cases.

New Zealand:

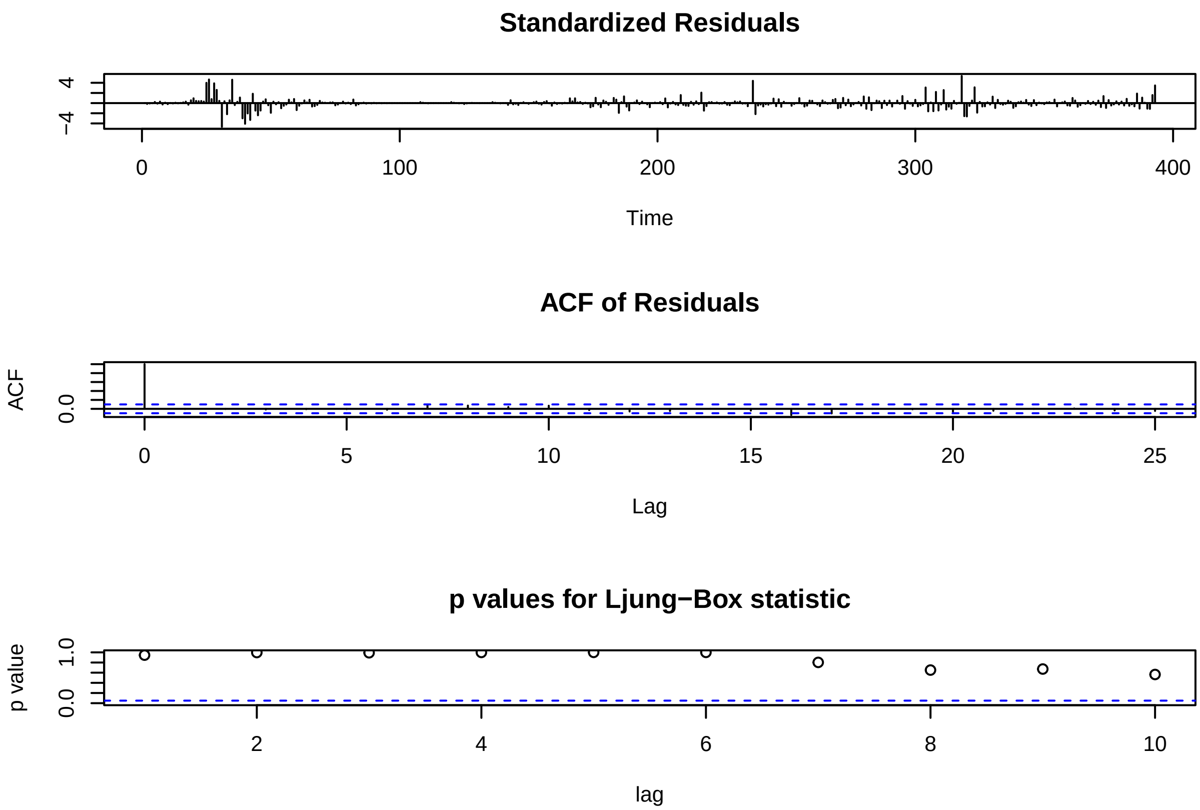
```r
# dayNzl <- auto.arima(nzl_data$CasesPerDay)

dayNzl <- Arima(nzl_data$CasesPerDay,order=c(3,1,3))
plot.ts(nzl_data$CasesPerDay, main="ARIMA model (NZL)",ylab="Cases Per Day")

lines(fitted(dayNzl), col = "red")
legend("topright", legend = c("Observed", "ARIMA Model"),
       col = c("black", "red"), lty = 1:1)
```




```
summary(dayNzl)
```

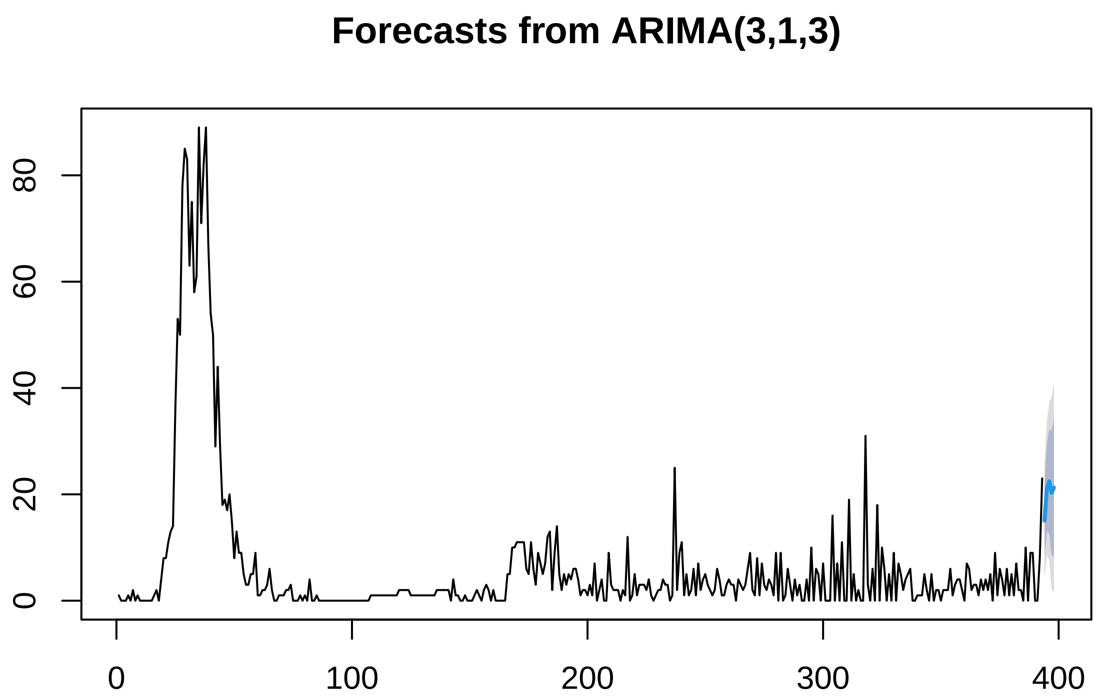
```
## Series: nzl_data$CasesPerDay
## ARIMA(3,1,3)
##
## Coefficients:
##           ar1      ar2      ar3     ma1     ma2     ma3
##       -0.9372  -0.2996  -0.1053  0.5165  0.0625  0.3204
## s.e.   0.2134   0.2650   0.1531  0.2058  0.1830  0.1185
##
## sigma^2 = 30.83:  log likelihood = -1225.41
## AIC=2464.82   AICc=2465.12   BIC=2492.62
##
## Training set error measures:
##                      ME     RMSE      MAE MPE MAPE      MASE         ACF1
## Training set 0.06295831 5.502854 3.116721 NaN  Inf 0.8701956 -0.003456786
```

```
tsdiag(dayNzl)
```




```
#acf(diff(nzl_data$CasesPerDay))
#pacf(diff(nzl_data$CasesPerDay))
```

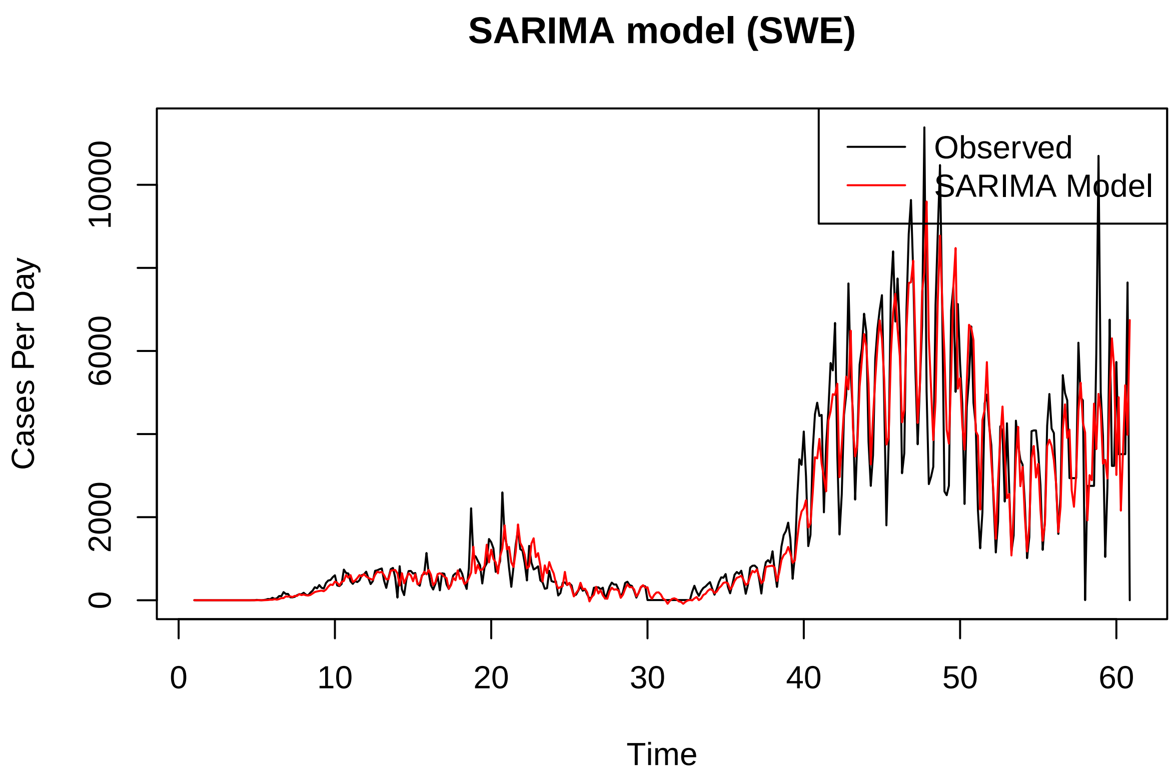
```
library(forecast)

nzl_cast <- forecast(dayNzl,h=5)

plot(nzl_cast)
```



Forecast:

Sweden

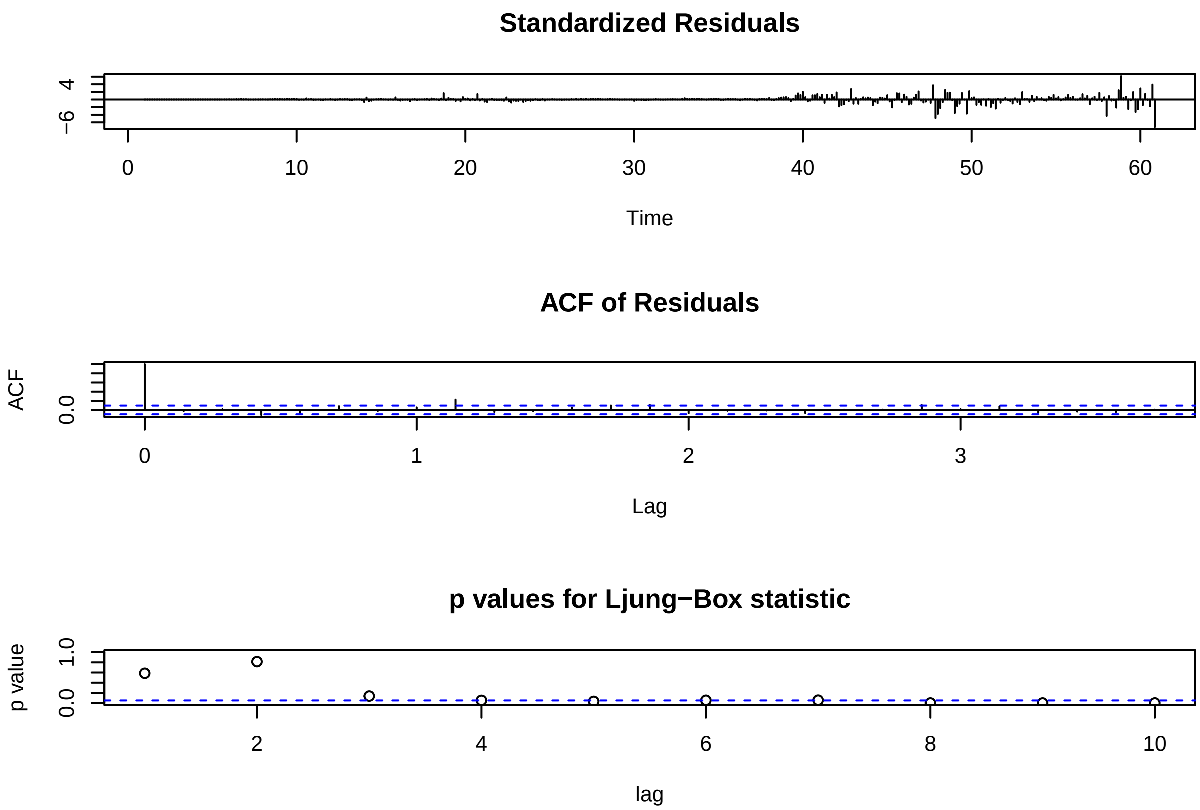
```
freqData <- ts(swe_data$CasesPerDay,frequency = 7)

daySwe <- auto.arima(freqData)

# daySwe <- Arima(freqData,order=c(2,1,1),
             #   seasonal = list(order = c(1,0,1), frequency = 7))

plot.ts(freqData,main="SARIMA model (SWE)",ylab="Cases Per Day")

lines(fitted(daySwe), col = "red")
legend("topright", legend = c("Observed", "SARIMA Model"),
       col = c("black", "red"), lty = 1:1)
```




```r
summary(daySwe)
```

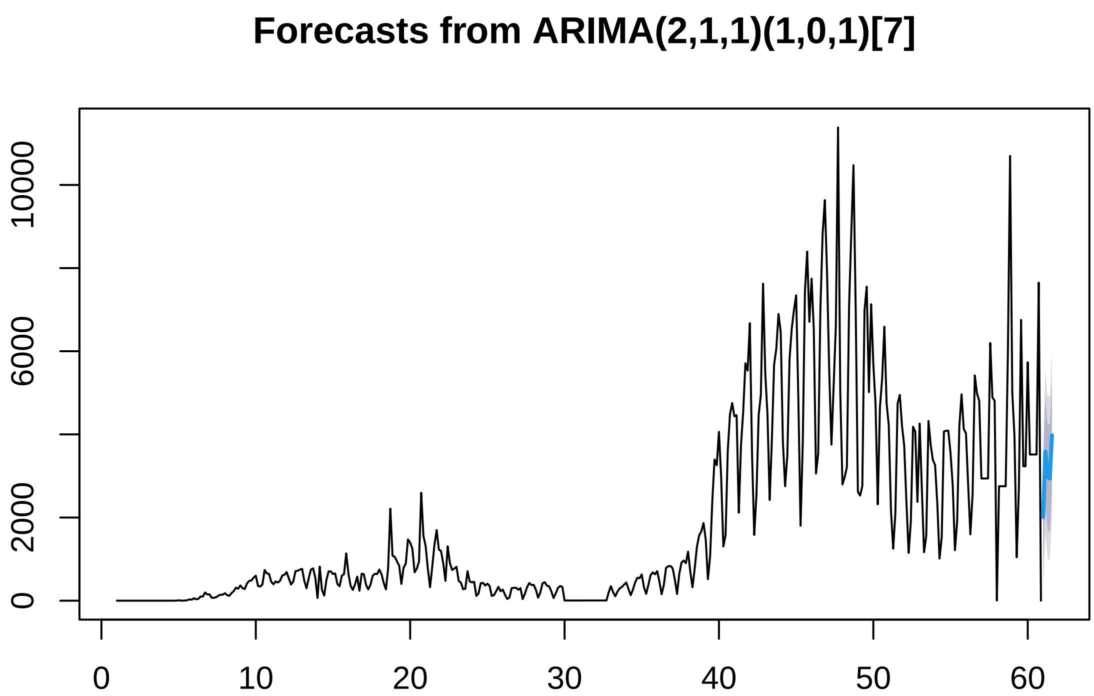
```
## Series: freqData
## ARIMA(2,1,1)(1,0,1)[7]
##
## Coefficients:
##          ar1      ar2      ma1    sar1     sma1
##       0.2410  -0.1818  -0.8584  0.8881  -0.5964
## s.e.  0.0579   0.0558   0.0294  0.0520   0.1122
##
## sigma^2 = 874192:  log likelihood = -3460.41
## AIC=6932.83   AICc=6933.03   BIC=6957.05
##
## Training set error measures:
##                    ME     RMSE      MAE MPE MAPE      MASE        ACF1
## Training set 11.64487 928.2801 468.3129 NaN  Inf 0.8510659 -0.02653019
```

```r
tsdiag(daySwe)
```




```
#acf(diff(swe_data$CasesPerDay))
#pacf(diff(swe_data$CasesPerDay))
```

```
library(forecast)

swe_cast <- forecast(daySwe,h=5)

plot(swe_cast)
```



Discussion:

The use of the ARIMA model and forecasting methods will help to identify when case spikes will occur, giving us a numerical approach to employing mitigation efforts.

Poisson Regression on Count Data:

The use of Poisson Regression for count data with an offset term of the population of each country will let us compare the rates of how quickly cases grew for two countries with populations that are very different from each other. To fit a Poisson Regression, we can use a Generalized Linear Model (GLM) with a poisson family and offset term.

```r
nzl_population <- 5000000
swe_population <- 10000000

nzl_data$log_nzl <- log(nzl_population)
swe_data$log_swe <- log(swe_population)

timeZl <- seq(nzl_data$CasesPerDay)
timeSw <- seq(swe_data$CasesPerDay)

# Create GLM models
model_nzl <- glm(Sum.of.Confirmed ~ timeZl, data = nzl_data,
                 family = poisson,offset = log_nzl)
model_swe <- glm(Sum.of.Confirmed ~ timeSw, data = swe_data,
                 family = poisson,offset = log_swe)

summary(model_nzl)
```

```
##
## Call:
## glm(formula = Sum.of.Confirmed ~ timeZl, family = poisson, data = nzl_data,
```

```
##     offset = log_nzl)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -43.507   -1.512    2.270    4.381   10.456
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.5812425  0.0028508 -3010.1   <2e-16 ***
## timeZl       0.0027734  0.0000111   249.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 122852  on 392  degrees of freedom
## Residual deviance:  58571  on 391  degrees of freedom
## AIC: 62093
##
## Number of Fisher Scoring iterations: 5
```

```r
summary(model_swe)
```

```
##
## Call:
## glm(formula = Sum.of.Confirmed ~ timeSw, family = poisson, data = swe_data,
##     offset = log_swe)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -188.381  -106.156    -7.717    66.434   151.173
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.152e+00  4.834e-04  -14798   <2e-16 ***
## timeSw       1.141e-02  1.399e-06    8152   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 105177193  on 419  degrees of freedom
## Residual deviance:   3744519  on 418  degrees of freedom
## AIC: 3749716
##
## Number of Fisher Scoring iterations: 4
```

Beta:

As shown by the slope estimates of 0.001141 for Sweden and 0.0027734 for New Zealand, the cases in New Zealand grew faster.