ISyE6402_Midterm

Jim Liu

10/14/2020

Question 1 (50 Points)

In this question we will be exploring the total monthly recreational visits to Yosemite National Park for the period between January 1986 and December 2016 . Each point represents total monthly visits to the park. We will be using analytical techniques to assess patterns to this data. Make no assumptions, all responses must cite evidence from your analytics for credit.

```
library(TSA)
```

```
##
## Attaching package: 'TSA'

## The following objects are masked from 'package:stats':
##
##     acf, arima

## The following object is masked from 'package:utils':
##
##     tar
```

```
library(mgcv)
```

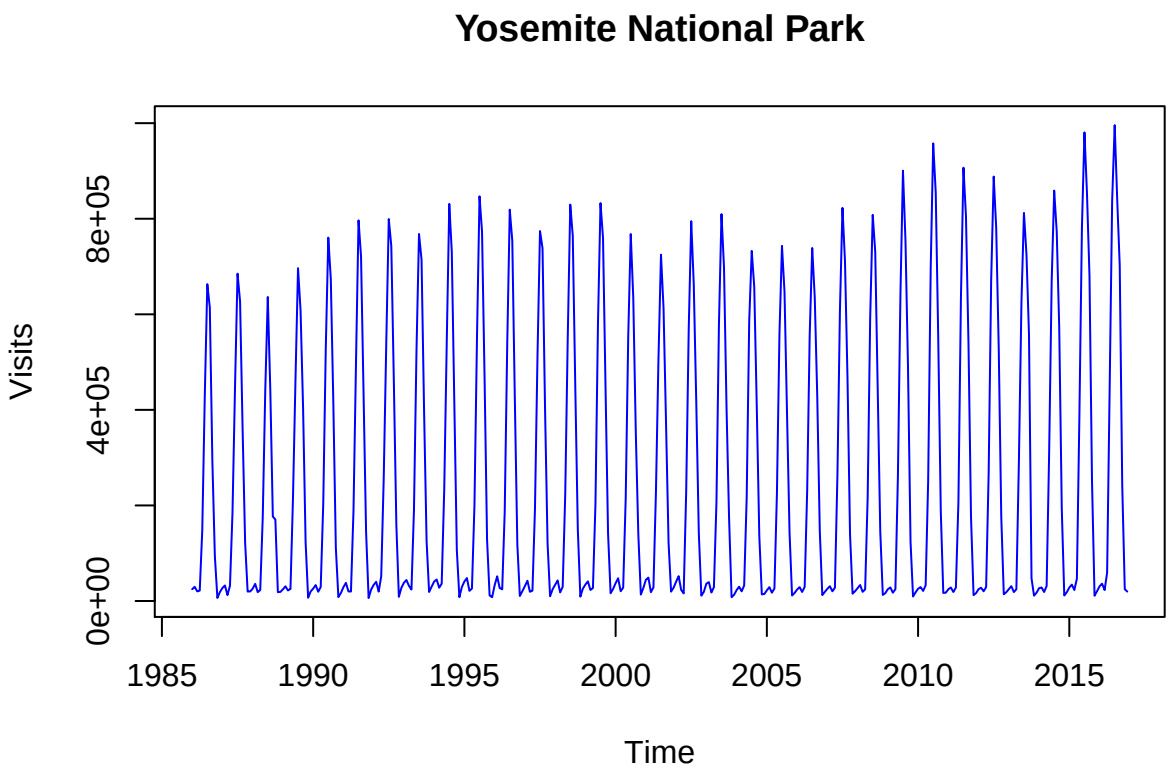
```
## Loading required package: nlme

## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.
```

```
#Load Data
library(TSA)
data <- read.csv("/Users/jim/Dropbox (GaTech)/Courses/ISyE6402/Midterm/YosemiteVisits.csv")
visits <- data$Recreation.Visits
#Convert to TS data
data.ts = ts(visits,start=c(1986, 01), frequency = 12)
```

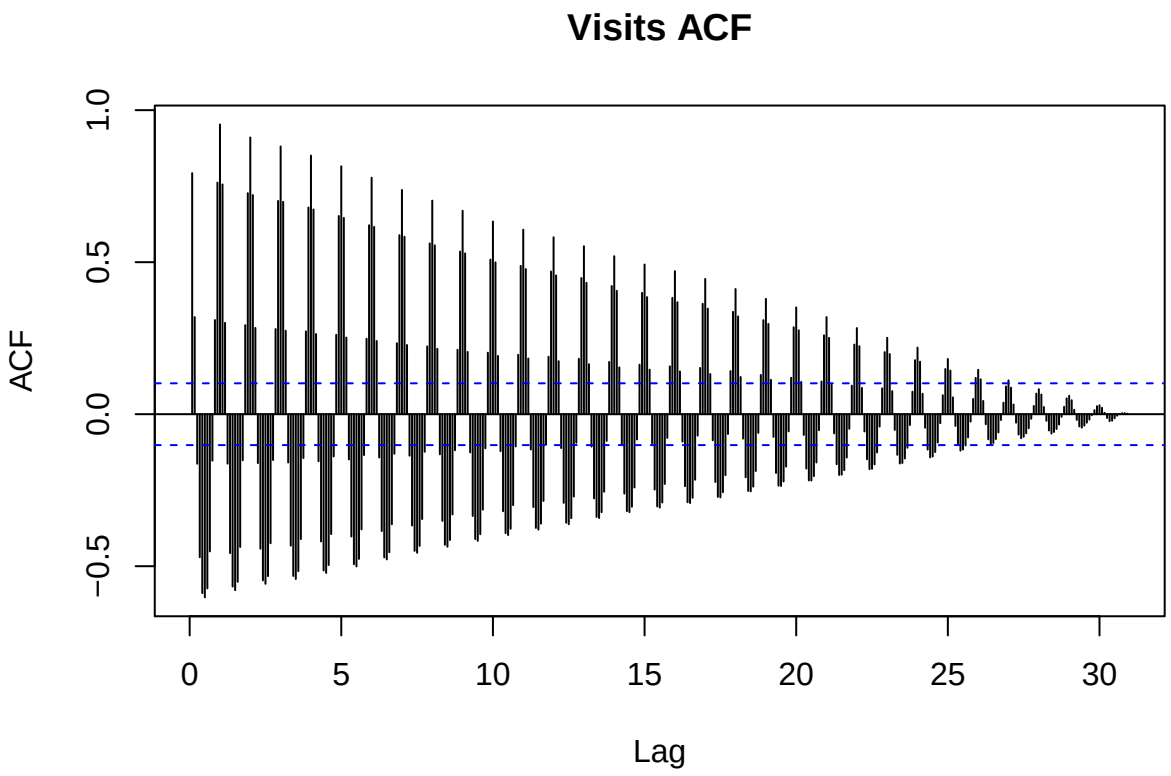
Part a. (6 points)

Plot the time series data as well as the ACF plot (use appropriate "lag.max" values to be able to draw relevant conclusions). Evaluate the assumptions of stationarity using graphical analysis. Explain which (if any) assumptions are violated.

```r
ts.plot(data.ts,main='Yosemite National Park',ylab='Visits', col='blue')
```




```r
acf(data.ts,main='Visits ACF', lag.max =12 * 48)
```



Evaluate the assumptions of stationarity using graphical analysis. Explain which (if any) assumptions are violated.

From the time series plot and ACF plot, we could see that it has seasonality. We could see a pattern that in some months, there are more visitors. So the assumption of autocovariance independent of time is violated. But, if lags are large, we could see there are no spikes outside the siginificant bands. As for the assumption of constant mean, in this plot, we could see there is a no clear increasing or decreasing trend.

Part b. (10 points)

Fit both ANOVA and Sin-Cos (2nd Harmonic) seasonal models to the data. Print out the coefficients for both models. For both the models, plot the fitted values overlaid on the original data.

```
#Estimate seasonality using ANOVA approach
month = season(ts(visits,start=c(1986, 01), frequency = 12))
model2 = lm(visits~month-1)
summary(model2)
```

```
##
## Call:
## lm(formula = visits ~ month - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -270920   -9080   -1492    7245  263321
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## monthJanuary      30064      10231   2.938 0.003511 **
## monthFebruary     36766      10231   3.593 0.000372 ***
## monthMarch        20534      10231   2.007 0.045493 *
## monthApril        29349      10231   2.869 0.004366 **
## monthMay         234322      10231  22.902  < 2e-16 ***
## monthJune        574995      10231  56.200  < 2e-16 ***
## monthJuly        805934      10231  78.772  < 2e-16 ***
## monthAugust      711050      10231  69.498  < 2e-16 ***
## monthSeptember   446561      10231  43.647  < 2e-16 ***
## monthOctober     145513      10231  14.222  < 2e-16 ***
## monthNovember     12512      10231   1.223 0.222169
## monthDecember     20314      10231   1.985 0.047847 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56970 on 360 degrees of freedom
## Multiple R-squared:  0.9791, Adjusted R-squared:  0.9784
## F-statistic:  1405 on 12 and 360 DF,  p-value: < 2.2e-16
```

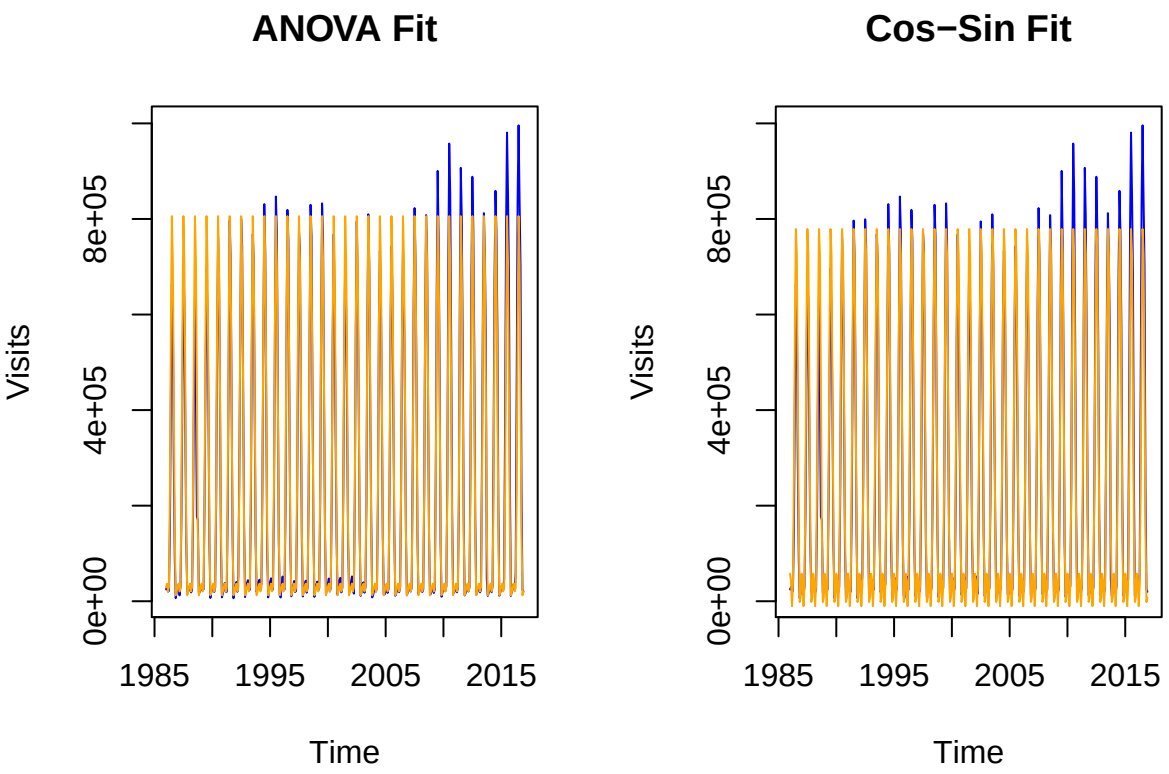
```
visit.fit.aov=model2$fitted
visit.fit.aov = ts(visit.fit.aov,start=c(1986, 01), frequency = 12)

#Estimate seasonality using cos-sin model
```

```r
har2=harmonic(ts(visits,start=c(1986, 01),frequency=12),2)
model4=lm(visits~har2)
summary(model4)
```

```
##
## Call:
## lm(formula = visits ~ har2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -285200  -27226   -1781   20475  264982
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       255660       3064   83.45   <2e-16 ***
## har2cos(2*pi*t)  -360684       4333  -83.24   <2e-16 ***
## har2cos(4*pi*t)   162622       4333   37.53   <2e-16 ***
## har2sin(2*pi*t)   -58804       4333  -13.57   <2e-16 ***
## har2sin(4*pi*t)    53804       4333   12.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59090 on 367 degrees of freedom
## Multiple R-squared:  0.9594, Adjusted R-squared:  0.959
## F-statistic:  2169 on 4 and 367 DF,  p-value: < 2.2e-16
```

```r
visit.fit.cossin = fitted(model4)
visit.fit.cossin = ts(visit.fit.cossin,start=c(1986, 01),frequency=12)
```

```r
#Plot and overlay Seasonality Fits
par(mfrow=c(1,2))
ts.plot(ts(visits,start=c(1986, 01), frequency = 12),main='ANOVA Fit',col='blue',ylab='Visits')
lines(visit.fit.aov,col='orange')
ts.plot(ts(visits,start=c(1986, 01), frequency = 12),main='Cos-Sin Fit',col='blue',ylab='Visits')
lines(visit.fit.cossin,col='orange')
```

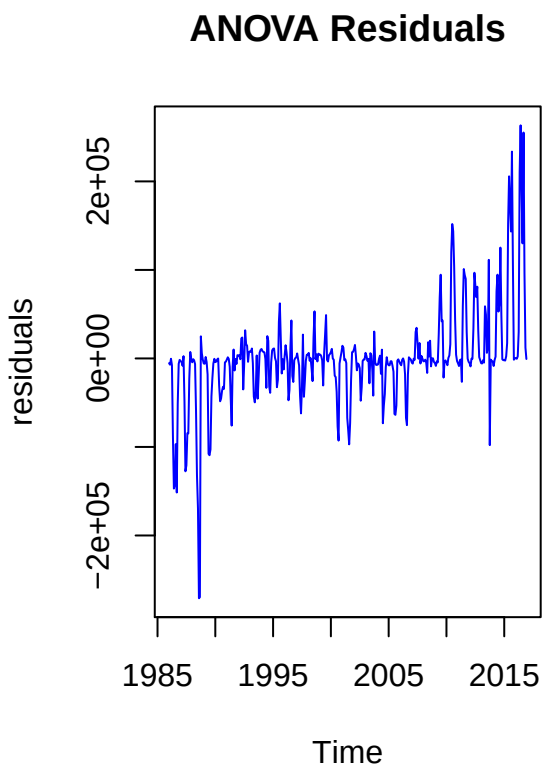


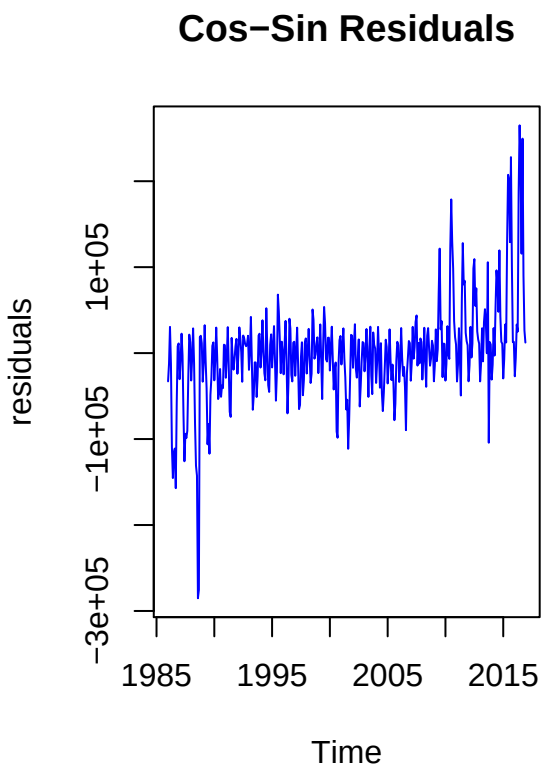
Part c. (6 points)

For each of the models above, plot the residuals and the residual ACF. Choose one model and study the stationarity of the residuals graphically. Provide your explanation on whether the stationarity assumptions hold.

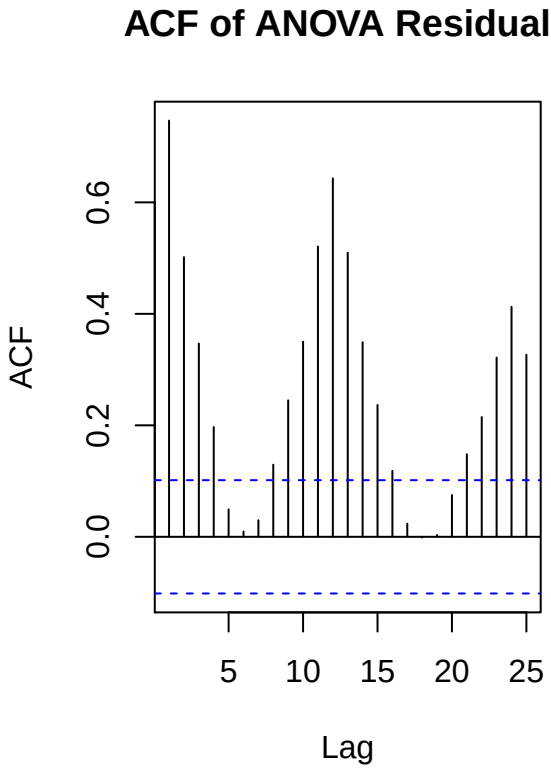
```
#Residual Analysis
resids.fit.aov=residuals(model2)
resids.fit.cossin=residuals(model4)
```

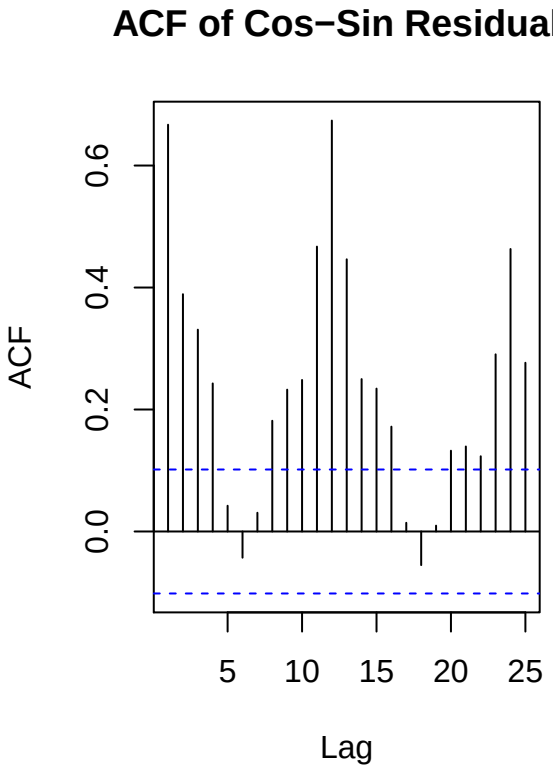
```
par(mfrow=c(1,2))
#Residuals plot
plot(ts(resids.fit.aov,start=c(1986, 01),frequency=12),main="ANOVA Residuals",cex=0.3,ylab="residuals",
plot(ts(resids.fit.cossin,start=c(1986, 01),frequency=12),main="Cos-Sin Residuals",cex=0.3,ylab="residua
```






```r
#Residual ACF
par(mfrow=c(1,2))
acf(as.numeric(resids.fit.aov),main="ACF of ANOVA Residuals",cex=0.3)
acf(as.numeric(resids.fit.cossin),main="ACF of Cos-Sin Residuals",cex=0.3)
```





Choose one model and study the stationarity of the residuals graphically. Provide your explanation on whether the stationarity assumptions hold.

Based on the Sin-Cos (2nd Harmonic) seasonal model, we could see its residuals still has the seasonal pattern. Therefore, we could see it violates the assumption of the autocovariance independent of time.

Part d. (4 points)

To assess fitting accuracy of both models, calculate Mean Absolute Percentage Error (MAPE) and Precision Measure. Compare the results.

```
### Mean Absolute Percentage Error (MAPE)
mean(abs(visit.fit.aov - visits)/visits)
```

```
## [1] 0.1722981
```

```
mean(abs(visit.fit.cossin - visits)/visits)
```

```
## [1] 0.4827031
```

```
### Precision Measure (PM)

# ANOVA
sum((visit.fit.aov-visits)^2)/sum((visits-mean(visits))^2)
```

```
## [1] 0.03699313
```

```
# Cos-sin
sum((visit.fit.cossin-visits)^2)/sum((visits-mean(visits))^2)
```

```
## [1] 0.0405809
```

MAPE gives us a model evaluation, because of its very intuitive interpretation in terms of relative error. For the prediction situation, we hope we could get smaller MAPE because from the definition if two values are close, the percentage would be smaller. For two models, ANOVA model performs better than cos-sin model since its MAPE 0.1722981 is smaller than 0.4827031

PM is also a similar indicator which use the sum of variance of observed values to show ratio between prediction and true values. If prediction is close to observed values, the value of PM would be close to 0. But, this value is not only close to 0 but also outside of 1. So it means most predictions are greater than one from the observed values to the mean. We know that the variance captures the spread of data. Therefore, if the values of PM is within 1, it shows that prediction is close to true values.

So in these two models, we could see ANOVA model has smaller value (0.036) than the cos-sin model (0.040).

Part e. (12 points)

Fit the following models to capture both trend and seasonality on the original time series:

Quadratic parametric trend regression on the data paired with ANOVA. Non-Parametric Regression using GAMs that use time.pts and month as predictors. For both the models, print coefficients and plot the residual plots.

```
# Equally spaced time points
time.pts = c(1:length(visits))
time.pts = c(time.pts - min(time.pts))/max(time.pts)

#Parametric quadratic polynomial
x1 = time.pts
x2 = time.pts^2
month = season(ts(visits,start=c(1986, 01),frequency=12))
lm.fit = lm(visits~x1+x2+month-1)
summary(lm.fit)
```

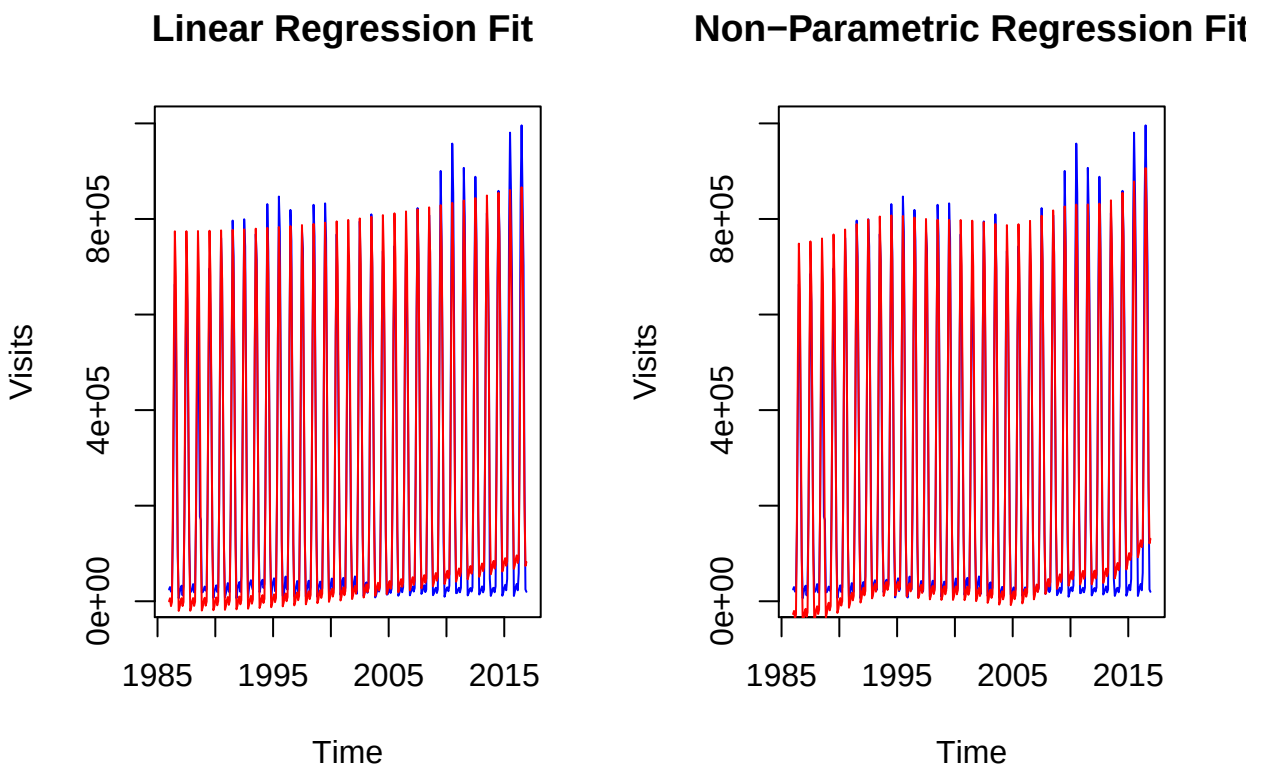
```
##
## Call:
## lm(formula = visits ~ x1 + x2 + month - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -239577  -25737    2304   25677  203070
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## x1                 -80.34   35322.94  -0.002   0.9982
## x2               95371.58   34286.62   2.782   0.0057 **
## monthJanuary      -165.49   11349.71  -0.015   0.9884
## monthFebruary     6286.97   11359.96   0.553   0.5803
## monthMarch      -10193.93   11370.00  -0.897   0.3706
## monthApril       -1630.49   11379.83  -0.143   0.8862
## monthMay        203089.03   11389.46  17.831   <2e-16 ***
## monthJune       543508.77   11398.89  47.681   <2e-16 ***
## monthJuly       774192.33   11408.11  67.863   <2e-16 ***
## monthAugust     679051.10   11417.13  59.477   <2e-16 ***
## monthSeptember 414303.94   11425.94  36.260   <2e-16 ***
## monthOctober    112995.85   11434.56   9.882   <2e-16 ***
## monthNovember   -20265.94   11442.97  -1.771   0.0774 .
## monthDecember   -12725.92   11451.17  -1.111   0.2672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49290 on 358 degrees of freedom
## Multiple R-squared:  0.9844, Adjusted R-squared:  0.9838
## F-statistic:  1618 on 14 and 358 DF,  p-value: < 2.2e-16
```

```
visit.fit.lm=ts(fitted(lm.fit),start=c(1986, 01),frequency=12)
```

```r
#Non-parametric model for trend with seasonality
gam.fit = gam(visits~s(time.pts)+month-1)
visit.fit.gam = ts(fitted(gam.fit),start=c(1986, 01),frequency=12)
summary(gam.fit)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## visits ~ s(time.pts) + month - 1
##
## Parametric coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## monthJanuary      32584       8316   3.918 0.000107 ***
## monthFebruary     38845       8314   4.672 4.25e-06 ***
## monthMarch        22169       8313   2.667 0.008008 **
## monthApril        30534       8312   3.674 0.000276 ***
## monthMay         235050       8311  28.281  < 2e-16 ***
## monthJune        575262       8311  69.219  < 2e-16 ***
## monthJuly        805733       8311  96.951  < 2e-16 ***
## monthAugust      710376       8311  85.473  < 2e-16 ***
## monthSeptember   445409       8312  53.588  < 2e-16 ***
## monthOctober     143876       8313  17.308  < 2e-16 ***
## monthNovember     10385       8314   1.249 0.212443
## monthDecember     17692       8316   2.128 0.034064 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df     F p-value
## s(time.pts) 7.839  8.651 21.89  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.975   Deviance explained = 98.7%
## GCV = 2.2617e+09  Scale est. = 2.1411e+09  n = 372
```

```r
#Plot fitted results
par(mfrow=c(1,2))
ts.plot(ts(visits,start=c(1986, 01),frequency=12),main='Linear Regression Fit ',ylab='Visits',col='blue
lines(visit.fit.lm,col='red')
ts.plot(ts(visits,start=c(1986, 01),frequency=12),main='Non-Parametric Regression Fit',ylab='Visits',co
lines(visit.fit.gam,col='red')
```



Part f. (4 points)

Similar to part d, calculate the Mean Absolute Percentage Error (MAPE) and Precision measures for the new models. Compare the measures in terms of model fit quality with the earlier seasonality-only models. Has incorporating trend modelling helped and why/why not?

```
### Mean Absolute Percentage Error (MAPE)
mean(abs(visit.fit.lm - visits)/visits)
```

```
## [1] 0.6633322
```

```
mean(abs(visit.fit.gam - visits)/visits)
```

```
## [1] 0.6319654
```

```
### Precision Measure (PM)
# lm
sum((visit.fit.lm-visits)^2)/sum((visits-mean(visits))^2)
```

```
## [1] 0.0275412
```

```
# gam
sum((visit.fit.gam-visits)^2)/sum((visits-mean(visits))^2)
```

```
## [1] 0.02387685
```

Compare the measures in terms of model fit quality with the earlier seasonality-only models. Has incorporating trend modelling helped and why/why not?

From MAPEs, it seems that there are more errors compared to seasonality-only models. But, for PMs, new models have smaller values, so it shows that new model data could fit better to within ratio of new varince to observed variance. And, they predict much better than seasonality-only models.

Has incorporating trend modelling helped and why/why not?

Yes. It captures some months which have higher values. From the plots, we could see there is an increasing trend and they capture the trend of hot seasons.

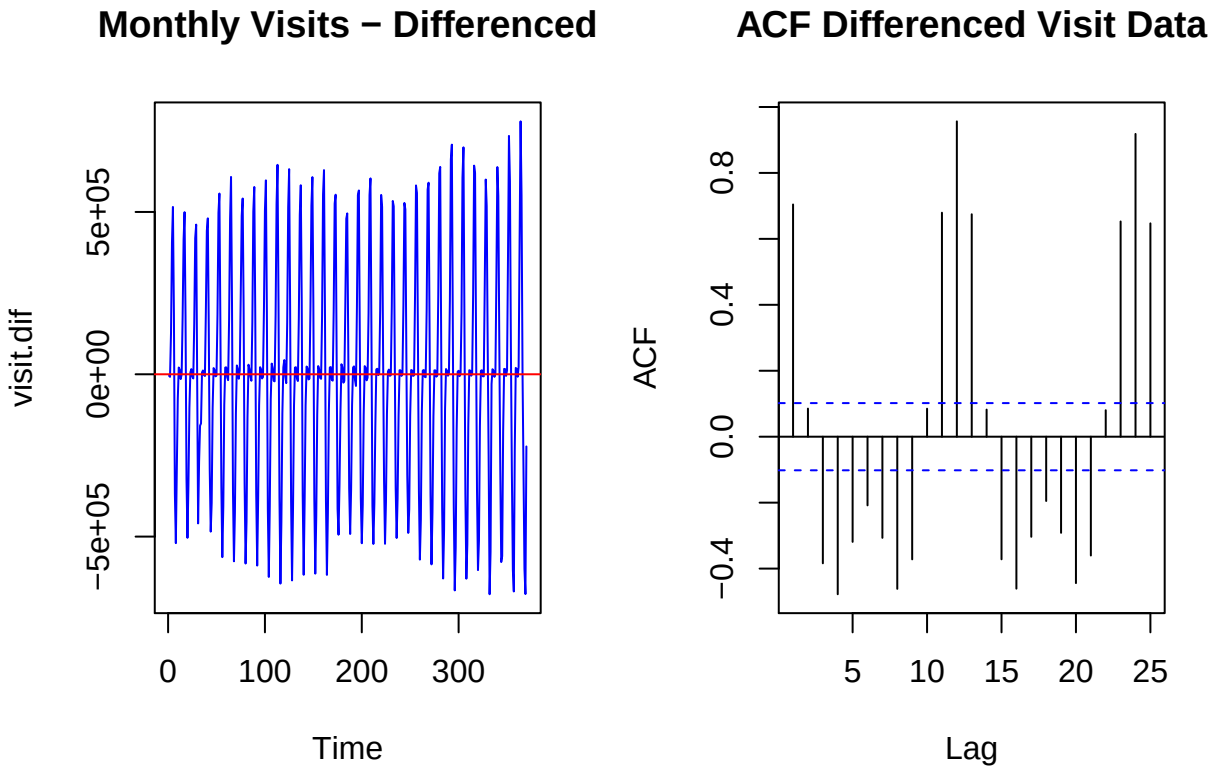
Part g. (8 points)

Apply differencing to the original time series by choosing the most appropriate "lag" parameter. By appropriate "lag" parameter we wish to choose a lag that most likely results in a stationary process. Which lag did you choose and why? Plot the ACF of this differenced time series.

```
orders <- orders[order(-orders$AIC),]
tail(orders)
```

```
##   p d q       AIC
## 7 0 5 0 10502.928
## 2 0 0 0 10420.933
## 6 0 4 0 10172.008
## 3 0 1 0 10061.726
## 5 0 3 0  9949.543
## 4 0 2 0  9904.169
```

```
par(mfrow=c(1,2))
visit.dif = diff(visits, lag=2)
visit.dif = c(visit.dif)
time.pts.diff = c(1:length(visit.dif))
time.pts.diff = c(time.pts.diff - min(time.pts.diff))/max(time.pts.diff)

ts.plot(visit.dif,main='Monthly Visits - Differenced',col='blue')
abline(a=mean(visit.dif),b=0,col='red')
acf(visit.dif,main='ACF Differenced Visit Data')
```



By appropriate "lag" parameter we wish to choose a lag that most likely results in a stationary process. Which lag did you choose and why? Plot the ACF of this differenced time series.

I chose lag = 2 since it has the smaller AIC value. But, for the differenced plot, we could see it still has a seasonality. So differenced data do not perform well.

Question 2 (50 Points)

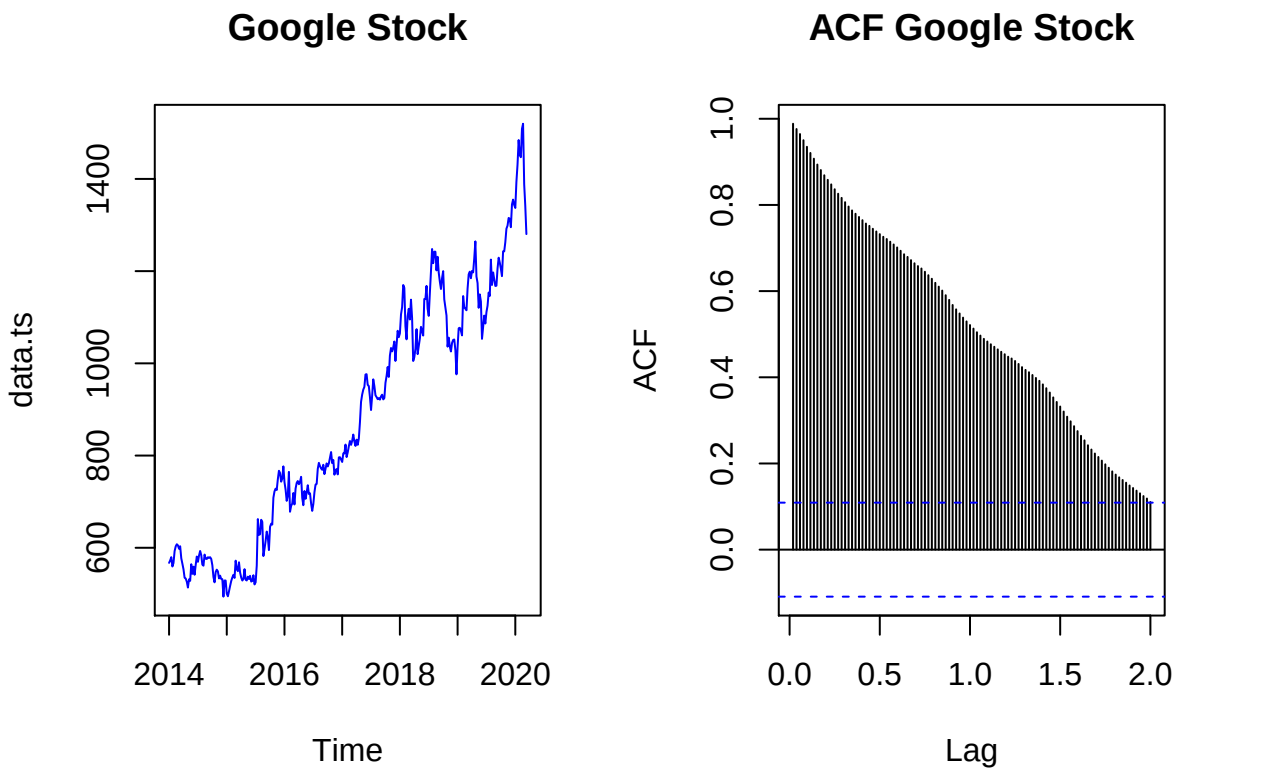
In this question we will explore using ARIMA forecasting on stock data. Specifically, we will be using weekly aggregated Google stock prices since 2014. Models will be built on training data from 2014-2019, and forecasts will be tested against the first 10 weekly prices in 2020. A helper template function is included for ARIMA order selection and root magnitude calculation.

```
library(TSA)
data <- read.csv("/Users/jim/Dropbox (GaTech)/Courses/ISyE6402/Midterm/GOOG.csv")
data <- data[2]
train <- data[(1:313),]
test <- data[(314:323),]
obs = test
train.ts <- ts(train,start=c(2014,1),freq=52)
data.ts <- ts(data,start=c(2014,1),freq=52)
```

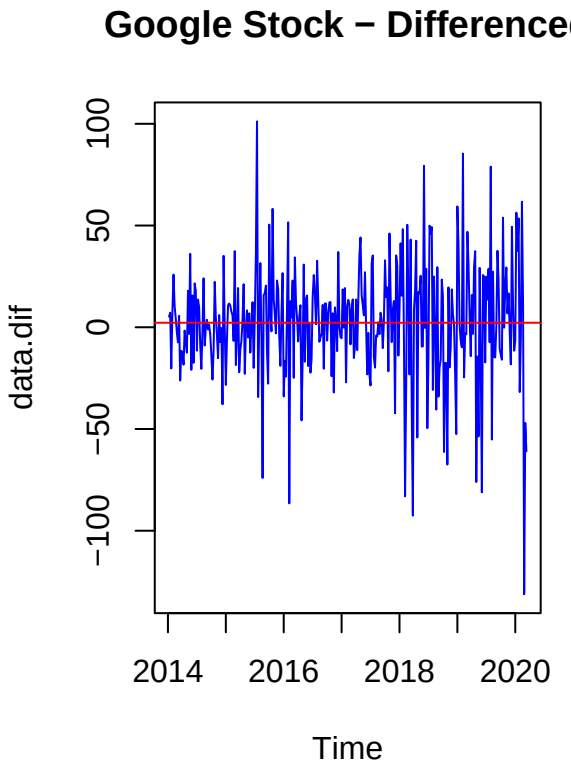
Part a. (8 Points)

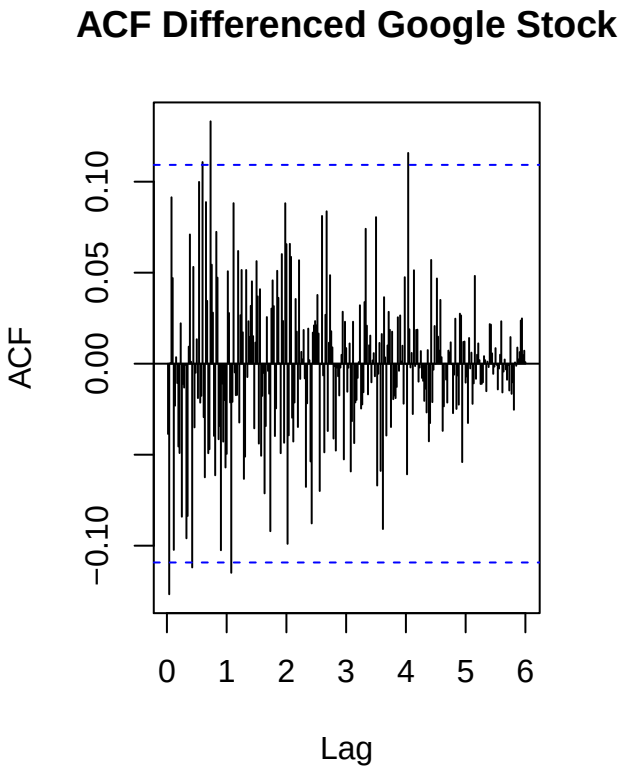
Plot the original data with ACF, what can you observe? Perform both 1st and 2nd differences on the original data, provide your explanation on whether the stationarity holds with relevant plots. Do you think ARIMA is a good type of model to use to model this data? Why or Why not?

```
par(mfrow=c(1,2))
ts.plot(data.ts,main='Google Stock',col='blue')
acf(data.ts,main='ACF Google Stock', lag.max = 52 * 2)
```

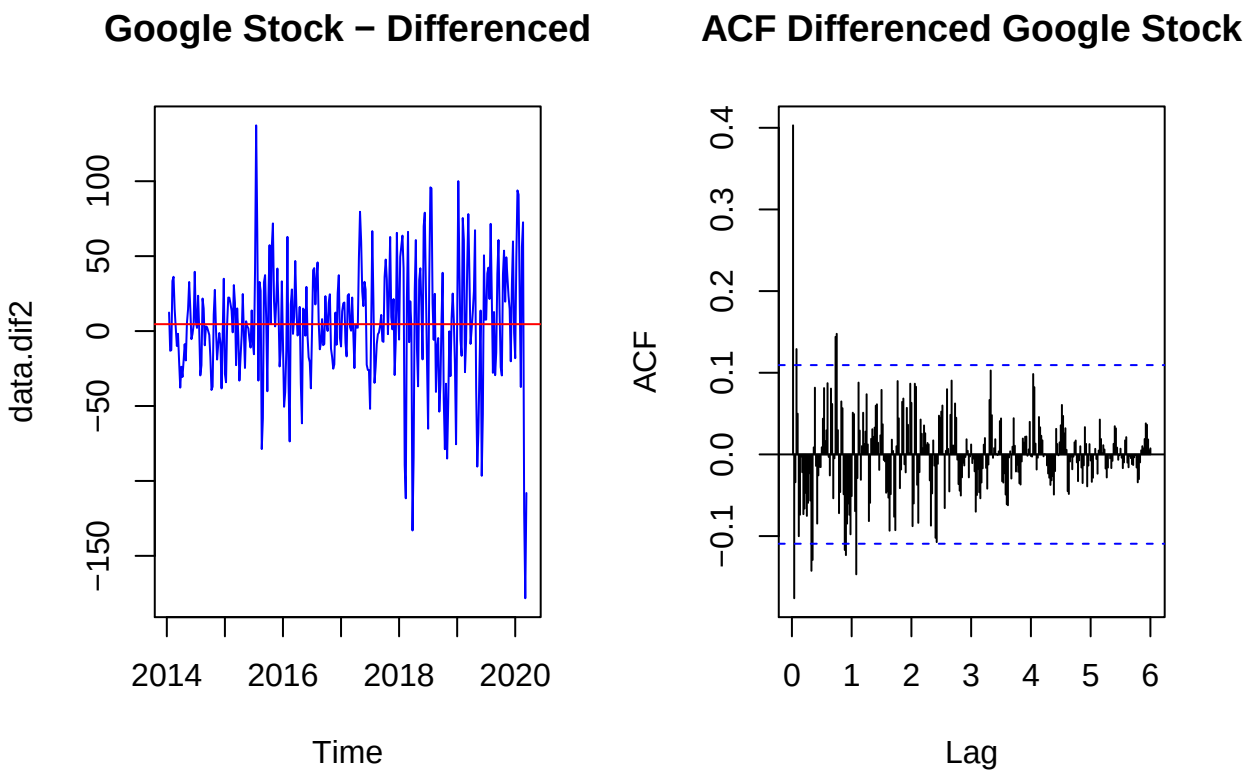



```
par(mfrow=c(1,2))
data.dif = diff(data.ts, lag=1)
ts.plot(data.dif,main='Google Stock - Differenced',col='blue')
abline(a=mean(data.dif),b=0,col='red')
acf(data.dif,main='ACF Differenced Google Stock', lag.max = 52*6)
```






```
par(mfrow=c(1,2))
data.dif2 = diff(data.ts, lag=2)
ts.plot(data.dif2,main='Google Stock - Differenced',col='blue')
abline(a=mean(data.dif2),b=0,col='red')
acf(data.dif2,main='ACF Differenced Google Stock', lag.max = 52*6)
```



Plot the original data with ACF, what can you observe? Perform both 1st and 2nd differences on the original data, provide your explanation on whether the stationarity holds with relevant plots. Do you think ARIMA is a good type of model to use to model this data? Why or Why not?

Plot the original data with ACF, what can you observe?

I could observe an increasing trend.

Perform both 1st and 2nd differences on the original data, provide your explanation on whether the stationarity holds with relevant plots.

(1) Constant mean: from the two plots, we could see there is a contant mean.
(2) Constant variance: from the two plots, we could see some years have much more variance, so this assumption does not hold.
(3) Autocovariance independent of time: This assumption does hold since with the lag greater than 3 or 4, most spikes that are inside of siginificant bands

---

Part b. (6 points)

Use the iterative approach to select ARIMA(p,d,q) order. Use max order (4,2,4) and select with the lowest AIC score. Report the order you select.

```
orders <- orders[order(-orders$AIC),]
tail(orders)
```

```
##    p d q      AIC
## 46 2 2 4 3071.968
```

```
## 57 3 2 1 3071.966
## 13 0 2 1 3071.685
## 15 0 2 3 3070.721
## 45 2 2 3 3070.479
## 43 2 2 1 3070.036
```

Report the order you select.

My selection is p=2, d=2 and q=1. That is, my model is ARIMA(2, 2, 1).

Part c. (12 points)

Fit ARIMA models with the order you choose from Part b as well as the order (0,2,3) on the training data. Display the coefficients of both models and evaluate how many are NOT significant at a 95% level of confidence. Interpret your findings.

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
final_221 = arima(data.ts, order = c(2,2,1), method = "ML")
final_023 = arima(data.ts, order = c(0,2,3), method = "ML")

coeftest(final_221)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error   z value Pr(>|z|)
## ar1 -0.0430912  0.0558047   -0.7722  0.44001
## ar2 -0.1286614  0.0559648   -2.2990  0.02151 *
## ma1 -0.9999996  0.0093967 -106.4203  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coeftest(final_023)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
```

```
## ma1 -1.042205   0.056941 -18.3031  < 2e-16 ***
## ma2 -0.069532   0.079403  -0.8757  0.38121
## ma3  0.111738   0.053502   2.0885  0.03676 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Display the coefficients of both models and evaluate how many are NOT significant at a 95% level of confidence. Interpret your findings.

For the ARIMA(2, 2, 1) model, there is one coefficient which is not significant. Other coefficients are smaller than 0.025, although ar2 seems to reach 0.025.

For the ARIMA(0, 2, 3) model, there is one coefficient which is not significant. Other coefficients are smaller than 0.025.

Part d. (10 points)

For both models, calculate the roots (round to the third decimal place) and evaluate if the process is Causal, Invertible, and/or Stationary. (Hint: you can solve either with R or by hand)

```r
#ARIMA root calculation template

#For model with order (p,d,q)
#AR Roots
round(abs(polyroot(c(1, coef(final_221)[1:2]) )),3)
```

```
## [1] 2.625 2.960
```

```r
#MA Roots
round(abs(polyroot(c(1, coef(final_221)[(2+1):(2+1)]) )),3)
```

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

```r
#For model with order (p,d,q)
#MA Roots
round(abs(polyroot(c(1, coef(final_023)[(0+1):(0+3)]) )),3)
```

```
## [1] 1.000 3.186 2.809
```

For these two models, Causal: ARIMA(2, 2, 1) is causal since they have roots greater than 1.

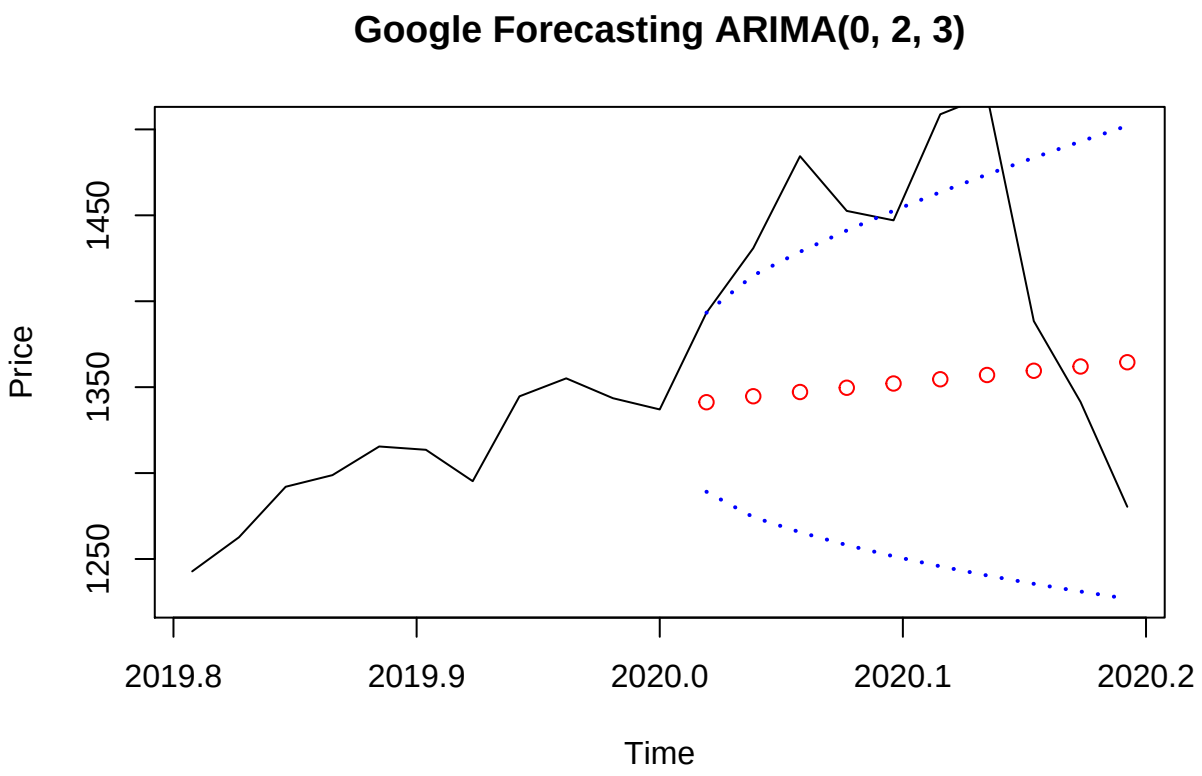
Invertible: ARIMA(2, 2, 1) is not invertible since they have roots greater than 1. ARIMA(0, 2, 3) is invertible since they have roots equal to 1.

Stationary: For both models, ARIMA(0, 2, 3) is stationary since MA(3) is always stationary. But, for ARIMA(2, 2, 1) is not stationary since its root is equal to 1.

Part f. (8 points)

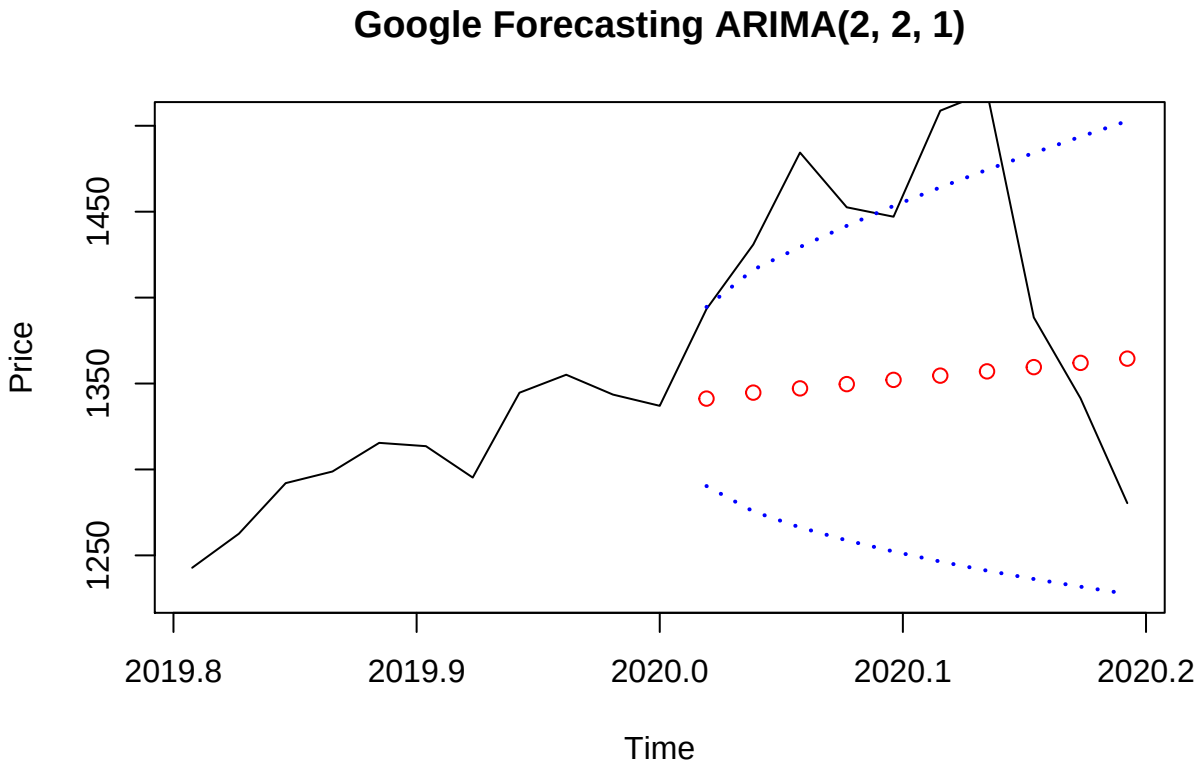
Use the forecasts from the model with order (0,2,3). overlay the points on plots of the actual values, including 95% confidence intervals. Comment on the plots.

```
n = length(data.ts)
nfit = n-10
outprice.023 = arima(data.ts[1:nfit], order = c(0,2,3),method = "ML")
outpred.023 = predict(outprice.023,n.ahead=10)
ubound.023 = outpred.023$pred+1.96*outpred.023$se
lbound.023 = outpred.023$pred-1.96*outpred.023$se
ymin = min(lbound.023)
ymax = max(ubound.023)
plot(time(data.ts)[(n-20):n], data.ts[(n-20):n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab="Price",
points(time(data.ts)[(nfit+1):n],outpred.023$pred,col="red")
lines(time(data.ts)[(nfit+1):n],ubound.023,lty=3,lwd= 2, col="blue")
lines(time(data.ts)[(nfit+1):n],lbound.023,lty=3,lwd= 2, col="blue")
```




```
n = length(data.ts)
nfit = n-10
outprice.221 = arima(data.ts[1:nfit], order = c(2,2,1),method = "ML")
outpred.221 = predict(outprice.221,n.ahead=10)
ubound.221 = outpred.221$pred+1.96*outpred.023$se
lbound.221 = outpred.221$pred-1.96*outpred.023$se
ymin = min(lbound.221)
ymax = max(ubound.221)
plot(time(data.ts)[(n-20):n], data.ts[(n-20):n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab="Price",
points(time(data.ts)[(nfit+1):n],outpred.023$pred,col="red")
```

```
lines(time(data.ts)[(nfit+1):n],ubound.221,lty=3,lwd= 2, col="blue")
lines(time(data.ts)[(nfit+1):n],lbound.221,lty=3,lwd= 2, col="blue")
```



Part e. (10 points)

With both models, forecast 10 time points ahead. Calculate the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE) for both forecasts when compared to the test data. Compare the model performance, which one performs better?

Calculate the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Mean Squared Error (MSE) for both forecasts when compared to the test data.

```
## Compute Accuracy Measures
obs.221 = data[(nfit+1):n,]
pred.221 = outpred.221$pred
### Mean Absolute Prediction Error (MAE)
mean(abs(pred.221-obs.221))
```

```
## [1] 91.87206
```

```
### Mean Absolute Percentage Error (MAPE)
mean(abs(pred.221-obs.221)/obs.221)
```

```
## [1] 0.06341899
```

```r
### Mean Squared Error(MSE)
mean((pred.221-obs.221)^2)
```

```
## [1] 10602.61
```

```r
## Compute Acuracy Measures
obs.023 = data[(nfit+1):n,]
pred.023 = outpred.023$pred
### Mean Absolute Prediction Error (MAE)
mean(abs(pred.023-obs.023))
```

```
## [1] 92.38033
```

```r
### Mean Absolute Percentage Error (MAPE)
mean(abs(pred.023-obs.023)/obs.023)
```

```
## [1] 0.06376097
```

```r
### Mean Squared Error(MSE)
mean((pred.023-obs.023)^2)
```

```
## [1] 10712.8
```

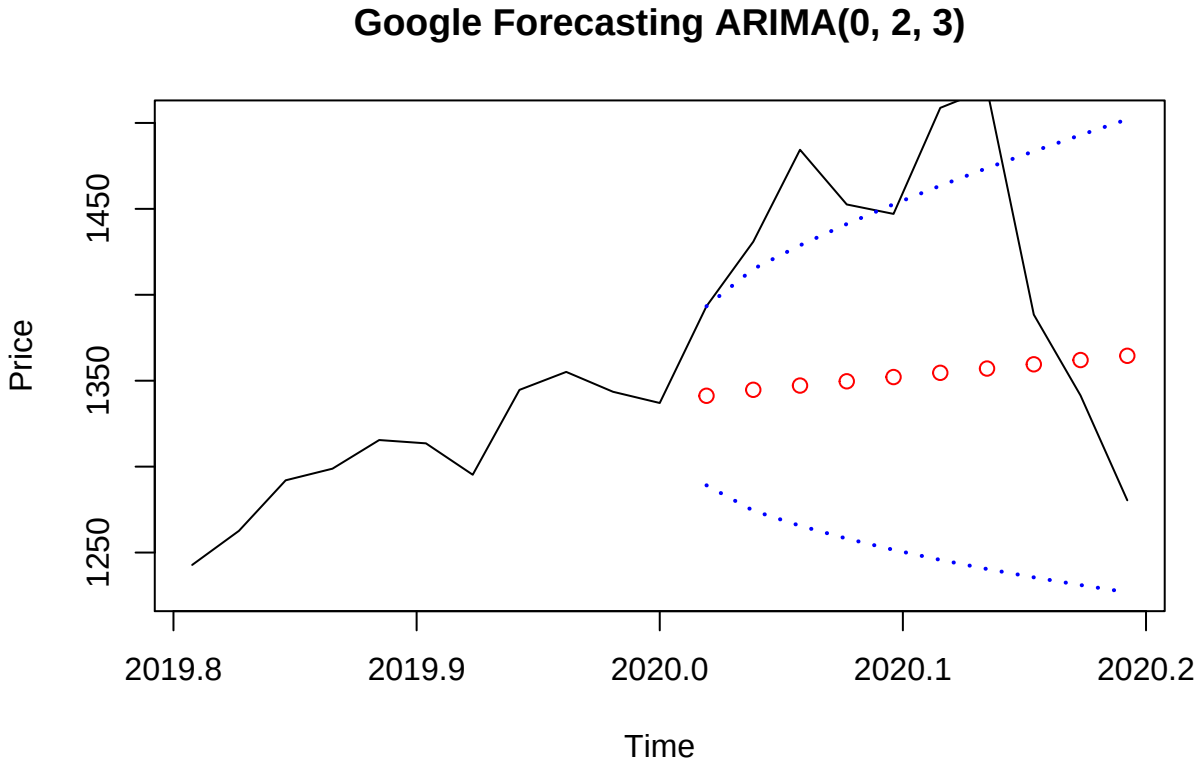
Compare the model performance, which one performs better?

ARIMA(2, 2, 1) performs better since it has smaller values compared to ARIMA(0, 2, 3)

Part f. (8 points)

Use the forecasts from the model with order (0,2,3). overlay the points on plots of the actual values, including 95% confidence intervals. Comment on the plots.

```r
n = length(data.ts)
nfit = n-10
outprice.023 = arima(data.ts[1:nfit], order = c(0,2,3),method = "ML")
outpred.023 = predict(outprice.023,n.ahead=10)
ubound.023 = outpred.023$pred+1.96*outpred.023$se
lbound.023 = outpred.023$pred-1.96*outpred.023$se
ymin = min(lbound.023)
ymax = max(ubound.023)
plot(time(data.ts)[(n-20):n], data.ts[(n-20):n],type="l", ylim=c(ymin,ymax), xlab="Time", ylab="Price",
points(time(data.ts)[(nfit+1):n],outpred.023$pred,col="red")
lines(time(data.ts)[(nfit+1):n],ubound.023,lty=3,lwd= 2, col="blue")
lines(time(data.ts)[(nfit+1):n],lbound.023,lty=3,lwd= 2, col="blue")
```



For prediction, it does not perform well. There is not increasing trend in the orginal dataset.

Part g. (6 points)

Reflection: Do ARIMA forecasts work well with stock data of this nature? What would you conclude about the viability of using ARIMA to predict stock values? To receive full credit you must cite your above results as well as general theory.

Answer: (1) Reflection: Do ARIMA forecasts work well with stock data of this nature?

No. From prediction accuracy, they are large [1] 92.38033, [1] 0.06376097, [1] 10712.8.

(2) What would you conclude about the viability of using ARIMA to predict stock values? Accuracy / theory like not stationary

I would not recommend it because from the accuracy metrics, we could see there are large values like MSE and MAE. Also, if a model is not stationary like there roots are equal to one.