Stat 153 Midterm 2

Samba Njie Jr. (SID: 23075185), Veronika Yang (SID: 3032577302)

4/7/2017

Report

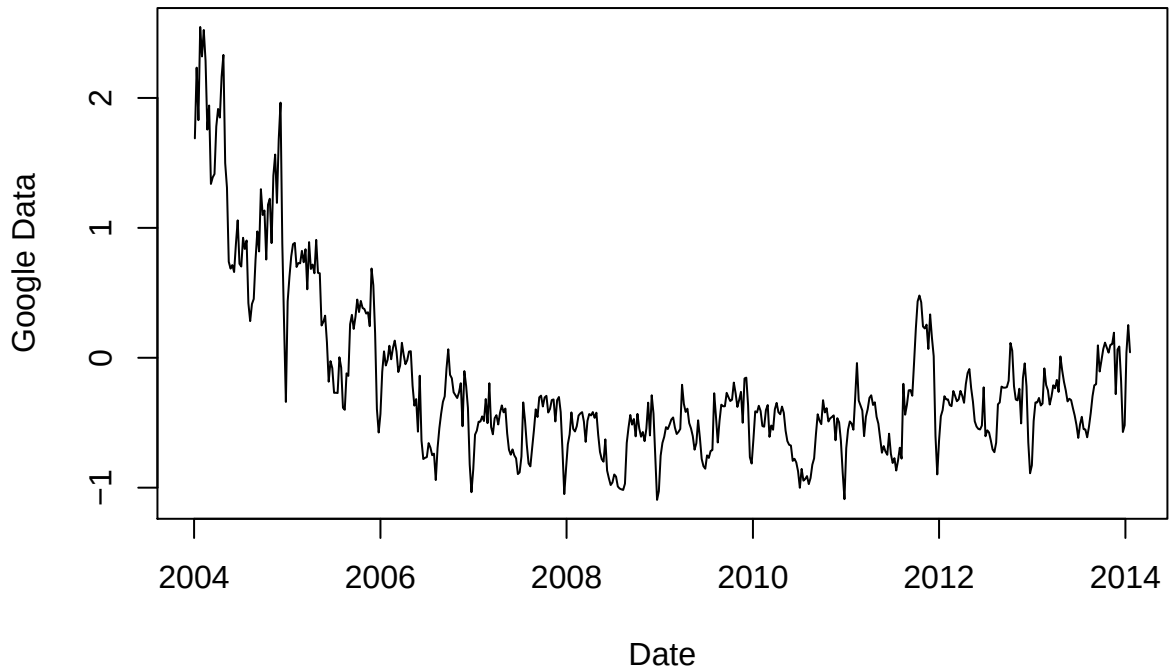
In this project, we are tested in our understanding of time-domain based methods for performing predictions for 5 time series Google Trends data sets, downloaded on April 7, 2017. At the culmination of our time-domain methods module, we have packed multiple time series analysis tools in our modeling arsenal, such as exploratory data analysis methods such as ACF/PACF analysis and detrending via differencing, diagnostic tests to determine strong candidate models with low statistics such as standardized residual analysis AIC, BIC, k-fold cross validation, and forecasting plots, as well as predictive methods to determine which time series model would produce the lowest test MSE against the true test set of our data.

Per report parameters, we chose to analyze the fifth data set (labeled `q5train.csv`) as a basis for interpretation. While specifics about transformation and model selection methodologies vary between different data sets, the same principles can be applied to other data sets, and you can observe our code for training sets 1,2,3, and 5.

Read in data sets:

Exploratory Data Analysis

Figure 1: Original Question 5 Training Set Plot



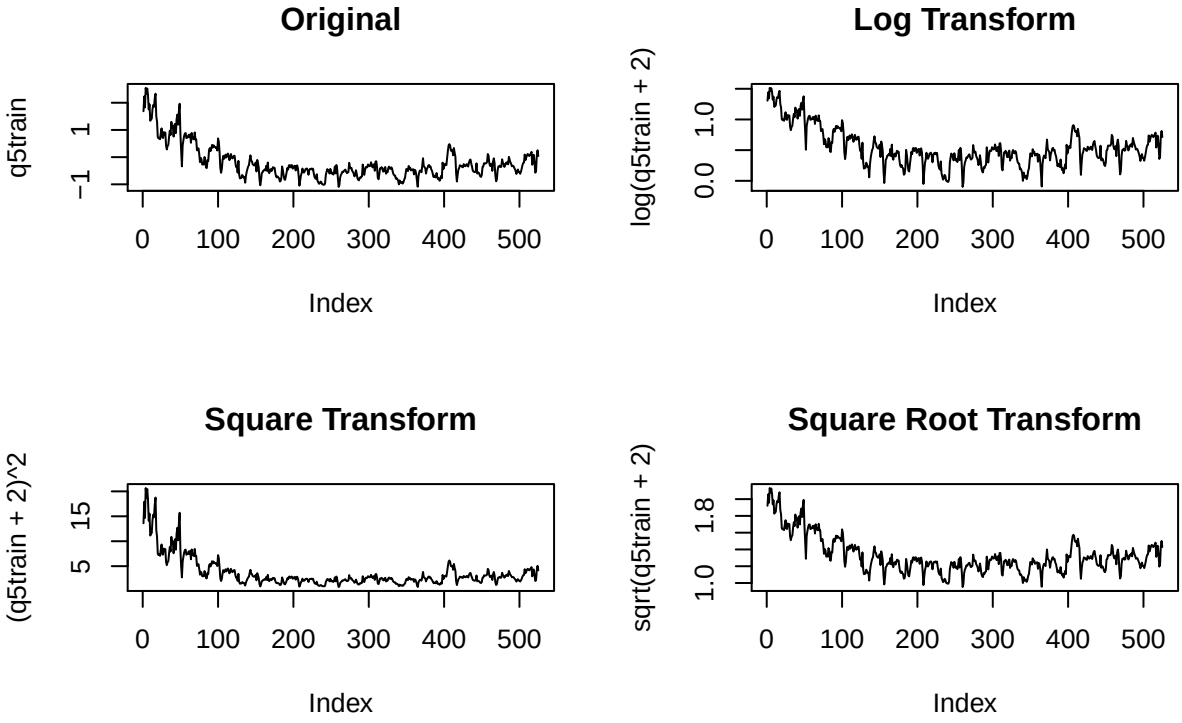
Per Figure 1, we can immediately see that we have to perform initial transformations and detrending to make the data stationary. Observe that the data set has some form of periodicity to it, which we attribute to seasonality, where there are four peaks for every two years, and the end of each two-peak consecutive period has a sharp decline after the second peak. This infers that each season may have two peaks, ending at a

sharp peak, as can be seen in 2005, 2006, 2007, etc. As such, we assume that for the data set gathered weekly, we have each period of length of a year, so we infer a seasonality of $s = 52$.

Data Transformations

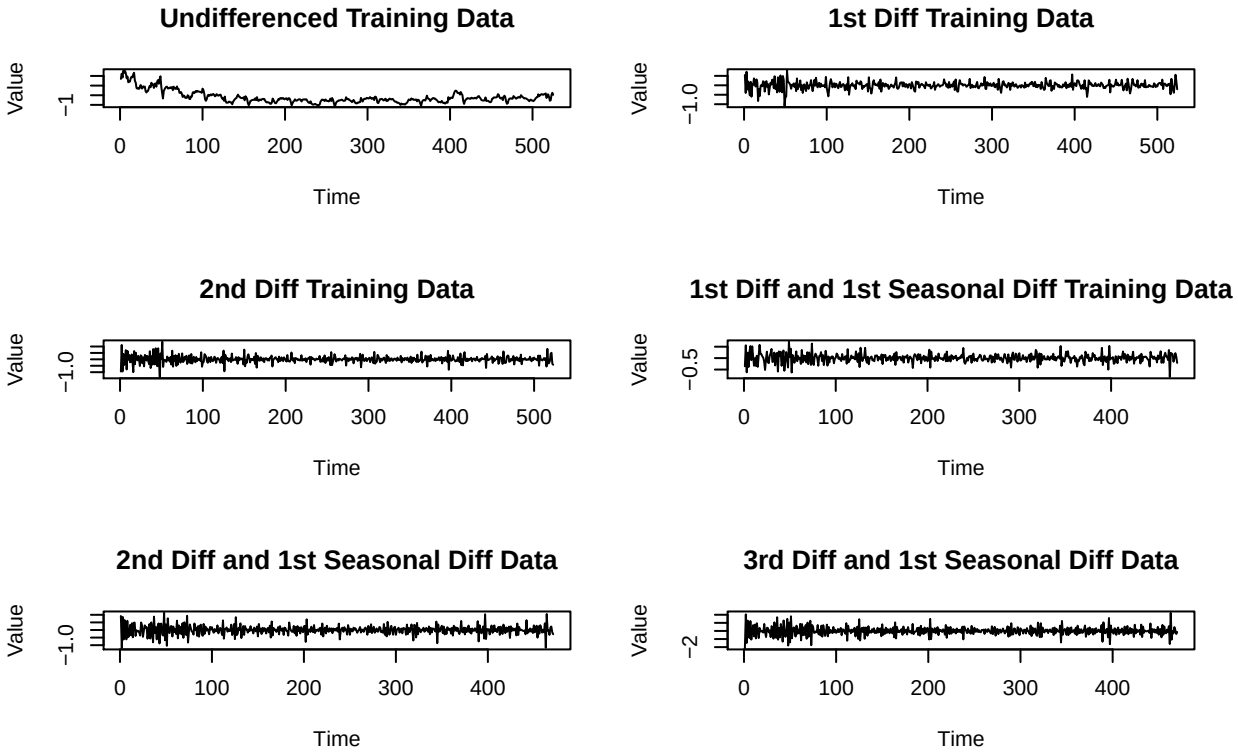
The first step is to remove heteroscedasticity, or time-dependent variance in the data set. Since stationarity requires a mean structure independent of time and autocovariance of the data only dependent on lag difference, then we want to make variance as consistent as possible. In Figure 2, We performed transformations which first shift the data up by 2 to remove negative terms, since the lowest negative term in `q5train` is -1.094, and allows us to do log and square root transformations. Notice that the log and square root transformations merely stretch the variance uniformly, while the square transformation, while shrinking the data set uniformly for time points 100 until 525, still retain high variance in the first 100 time points. As such, we've ignored transformations for this data set to minimize heteroscedasticity as much as possible.

Figure 2: Possible Transformations



Detrending via Differencing

Figure 3: Differenced Data



We heuristically infer using the standard deviation that the differenced data set with the smallest standard deviation is a good basis for model ideation. We also include the constraint that since the the original plot displays seasonality, it needs to be differenced seasonally, introducing parameters P, D, Q to create a SARIMA model of $ARIMA(p,d,q)(P,D,Q)_s$, with $s = 52$.

	Data	SD
1	Training Set	0.64
2	1st Diff	0.20
3	2nd Diff	0.28
4	Third Diff	0.49
5	1st Diff 1st Season	0.19
6	2nd Diff First Season	0.31
7	3rd Diff First Season	0.56

Table 1: Figure 4: Standard Deviation of Detrended Data

Since `q5diff1.52`, the data with first seasonal and first nonseasonal differencing, shows the lowest standard deviation, we observe this plot closely. Also, we can visually infer that after performing one nonseasonal and one seasonal differencing, we observe a stationary time series plot, which is further confirmed by the ACF plots below in Figure 4. Note that the 2nd row, 2nd column plot has an ACF that tails off more quickly than the other plots, and can visually see that most of the magnitudes of the ACF lags are on average smaller than higher-order differenced plots such as `q5diff2` and `q5diff3`, yet is more stationary than lower-order differencing such as nonseasonal n differencing and no seasonal differencing plots for $n = 1, 2, 3$. Also by Figure 4, notice that `q5diff1.52` or "1st Diff 1st Season" has the lowest standard deviation of `0.19`, leading us to choose these values of d and D for model interpretation.

Figure 5: ACF plots

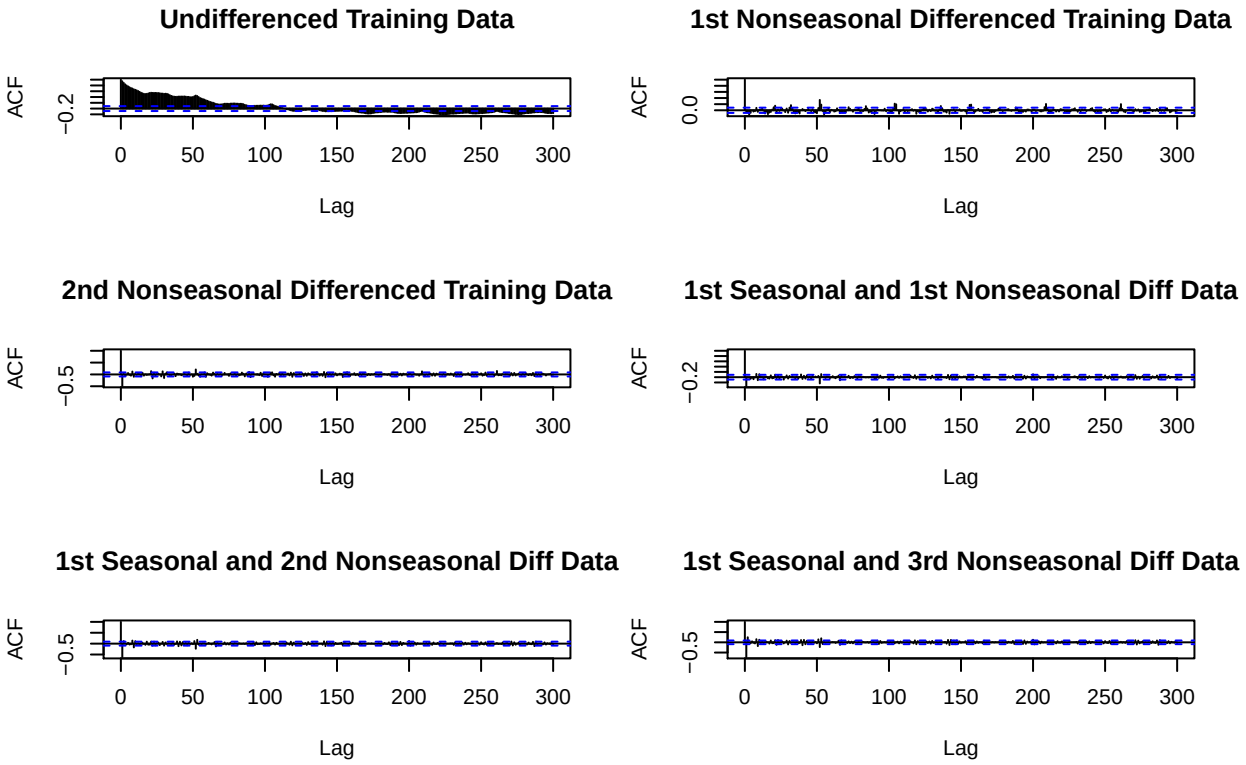
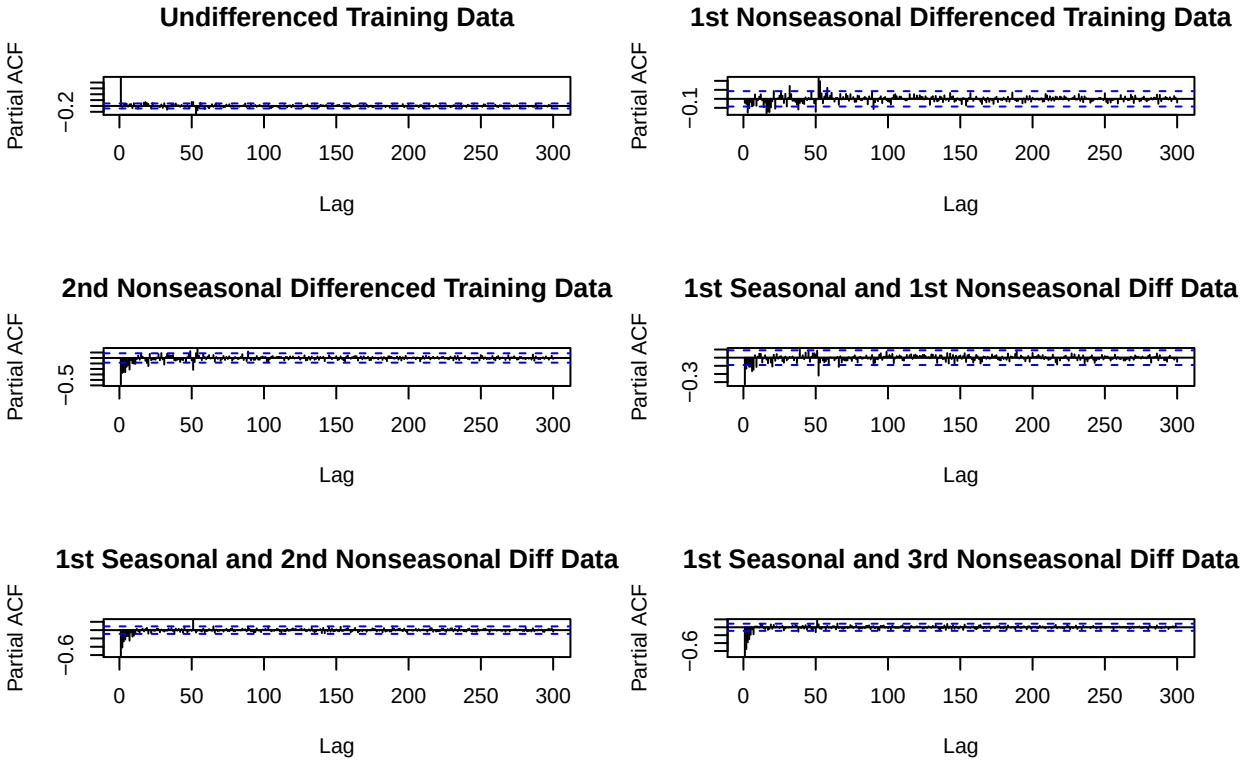


Figure 6: PACF plots



Take Figure 3. We already know that there is a first order seasonal and first order nonseasonal differencing.

Also, since at lag-1, the magnitude of the acf is significantly higher than other acf lags and negative direction is an indicator of a nonseasonal moving average signature. We assume that this nonseasonal lag decays after the first lag, so there is at least an $MA(1)$ component to our model, however using forward stepwise selection, we can increment this value of q to observe more models. Also notice that there is a significant negative lag at arond the 52nd lag in the ACF plot for `q5diff1.52` and declines for every 52 lags or so for higher order lags assumes an MA component Q in the seasonal component of the model as well. For the 1st-order seasonal differenced data with 2nd and 3rd order nonseasonal differencing, the range of the lag-magnitudes is greater and so we see lags of higher magnitude for those plots, which shows signs of overdifferencing. On the other hand, the PACF, while sporadic, does not give us much information. We assume an $AR(0)$ because while the AR component can be described by a significant magnititude at lag-1 of the PACF, it is facing the negative direction, which does not tell us that there is an AR signature. As such we begin our inspection with an $ARIMA(0,1,1)(0,1,0)_{52}$. We will begin our model fitting then with a model that is simple yet describes the data set, then increment to higher orders of q, Q, and other degrees, performing forward stepwise selection to achieve the best model.

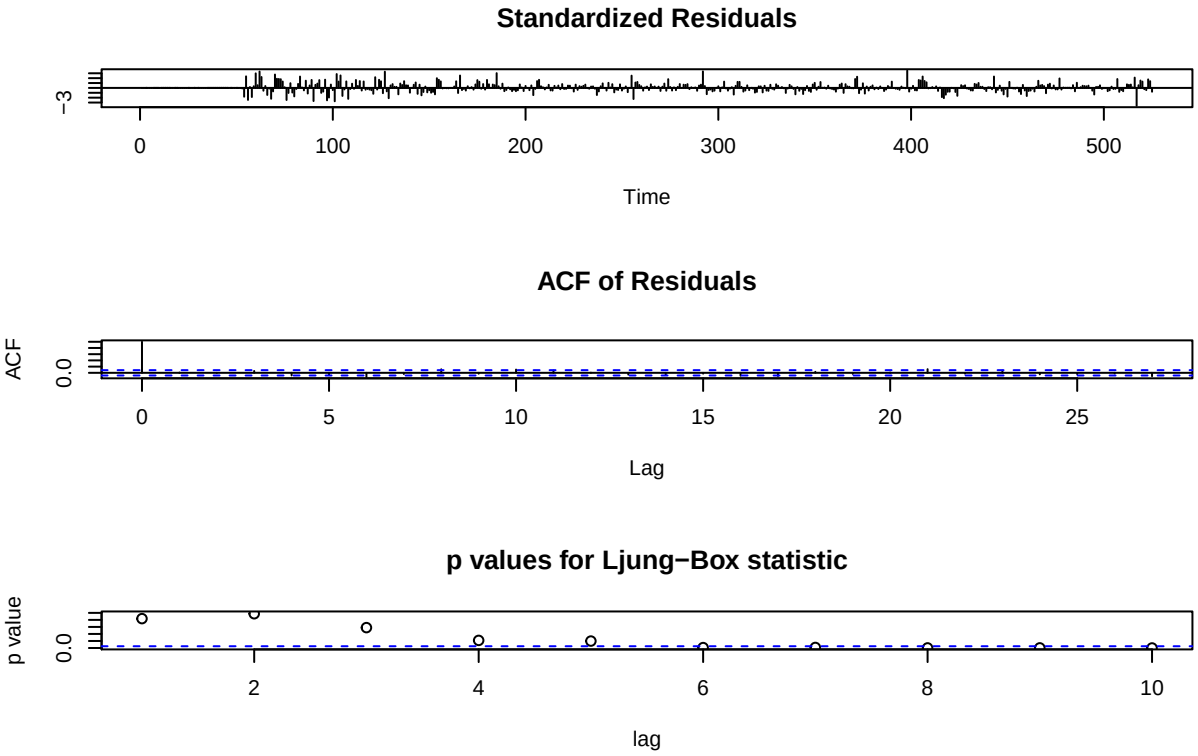
Also, there are other methods of detrending, such as smoothing and parametric detrending. While these are viable alternatives, they did not produce stationary results, so we stuck with differencing as our best detrending technique.

Model Fitting

We chose to increment a variety of models in our data set. We first began with the simplest case, an $ARIMA(0,1,0)(0,1,0)_{52}$ for `q5m1`, and then also tried performing diagnostic tests by incrementing q and Q where q, $Q \in \{0,1,2,3\}$. This brings us to a total of 16 models to increment, a stepwise approach, since reaching higher order models with overfit data can allow us to infer if more complex models are more likely to overfit based on the MSE and variance interpretations of our models.

Standardized Residuals, ACF, Ljung-Box Test Analysis

Figure 7: Residuals Analysis Plots



We now want to observe the residuals of each model as a goodness-of-fit test. The standardized residuals indicate if the models observed are stationary, i.e., if the residuals are "patternless" and look like white noise. If so, then we can extract only those that are white noise residuals to perform further analysis. The second plot, white noise ACF, also assists in the understanding if the ACF is purely white noise, which we would optimally want a significant lag-0 autocorrelation, and lag-n correlations that are within the confidence bands and ideally 0 or close to 0, for $n \geq 1$. The Ljung-Box Test (third plot) is a hypothesis test indicating if there is serial correlation in the data. We want it to be the case that the p-values are significant, since the null hypotheses state that there is no serial correlation. Most of the models have this decaying trend with significant lags, most notably all models from `q5m5` ($ARIMA(0,1,1)x(0,1,Q)_{52}$) and higher. So we consider these plots.

It seems that the last three models of $ARIMA(0,1,3)$ had significant Ljung-Box statistics where there was an asymptotic decay of lagged p-values for increasing value of lag-n. However, these are not relevant, since Ljung-Box test is a hypothesis test assessing the strength of the serial correlation in the data. For residuals that look like white noise (which is true for every model we have fit) then there is no serial autocorrelation to prove.

AIC, BIC and Cross-Validation (CV) Analysis

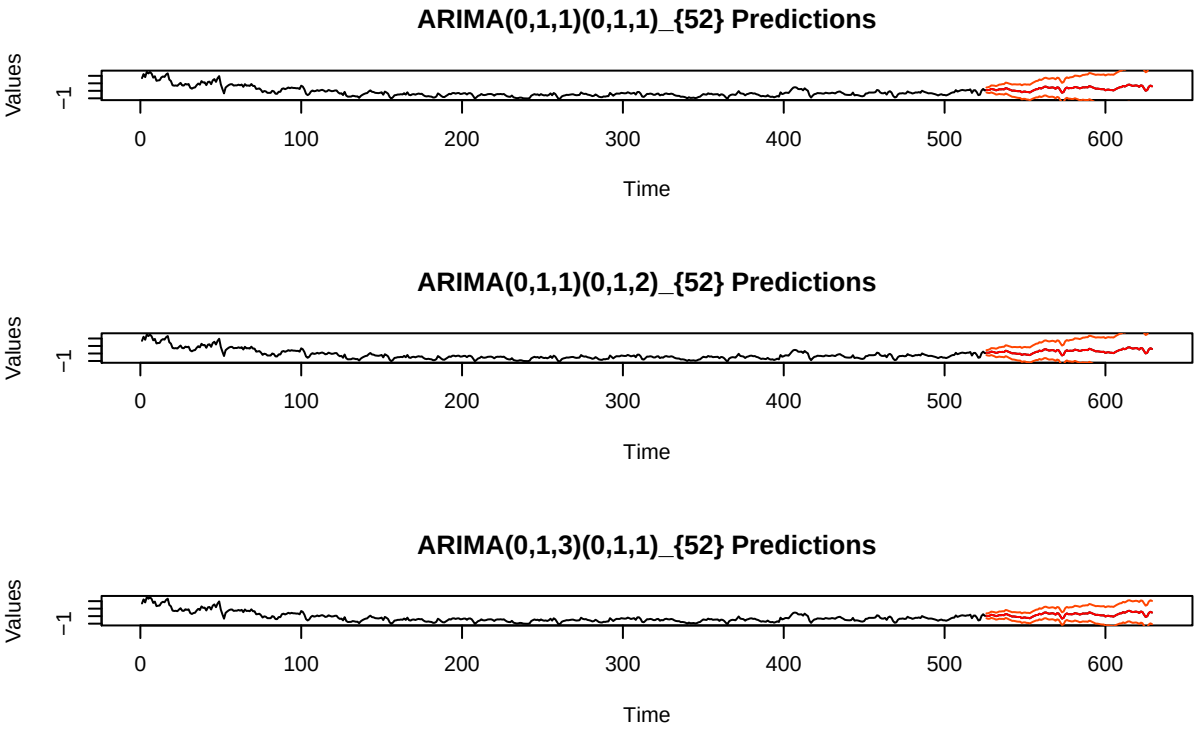
AIC and BIC are both penalized likelihood criteria, penalizing overfitting of models. Each new parameter increases likelihood of data, which in this case is increasing degrees of p, q, P, Q. AIC is better with more complex models and BIC with simpler models. We want a model that has minimal values for both. In this case, both AIC and BIC have that the 14th model `q5m14` ($ARIMA(0,1,3)(0,1,1)_{52}$) has the lowest AIC and lowest BIC amongst other models. Considering how both statistics penalize overfitting, this is a good approximation.

However, for cross validation, the 7th model `q5m7` ($ARIMA(0,1,1)(0,1,2)_{52}$) has the lowest cross validation mean MSE. Cross-validation is performed 5-fold, with each fold representing each season. For every model

that we are fitting, we find the MSE of each fold, which provides us a vector of MSE values for each fold for each model. Then we take the mean of each fold for each model, which provides us with the mean MSE values under the `CV_mean` column in Figure 7 above. We also observe that `q5m6` ($ARIMA(0,1,1)(0,1,1)_{52}$) is a similar model with the second lowest cross-validation mean MSE. Since cross validation is a good estimator because it tailors to the prediction problem considering seasonal terms represent recurring trends in the data. Since there is no hard and fast solution, we need to consider AIC, BIC, and k-fold cross validation into picking the best model, as well as the forecasting plots above. For now, we filter a balance between these three statistics and filter our `q5m6`, `q5m7`, and `q5m14` to be our chosen models, for their low AIC, BIC (for the 14th model) and low CV (for the 6th and 7th model) values.

Predictions

Figure 8: Forecasting Plots



	Name	Model	Variance	AIC	BIC	CV
1	q5m6	ARIMA(0,1,1)(0,1,1)[52]	0.0282707303145888	-327.068853408683	-314.597916451927	0.0856354994247
2	q5m7	ARIMA(0,1,1)(0,1,2)[52]	0.028259988282943	-325.173244777614	-308.545328835272	0.0800545729159
3	q5m14	ARIMA(0,1,3)(0,1,1)[52]	0.0271464580883575	-341.356087715552	-320.571192787624	0.1021398060897
4	Min		q5m14	q5m14	q5m14	q5m7

Table 2: Figure 9: Question 5 Statistics

Amongst these two models, `q5m14` ($ARIMA(0,1,3)x(0,1,1)_{52}$) has the lowest AIC and BIC statistic values, with also the lowest variance. However, it has the highest cross-validation mean value of the three, which means that it is not as subject to overfitting as the other data pointsm which may have overfitting since they have higher variance, and so if a model has higher variance but high MSE, then it is more likely that the high MSE is a result of higher variance than higher bias, unless they are both high. However, with low variance and high MSE such as for `qm14`, we most likely have the case of low variance, and high MSE due to high bias, so we may underfit the model more, but this has higher predictive power.

On the other hand,`q5m7` ($ARIMA(0,1,1)x(0,1,2)_{52}$) and `q5m6` ($ARIMA(0,1,1)x(0,1,1)_{52}$) has larger AIC, BIC values, and the cross-validation mean value is the lowest for the 7th model, as such overfitting is more likely to occur. As such, we choose a more complex model, due to likeliness of not overfitting due to the statistics shown, and it is also reflected in the forecasting plot, where we see `q5m14` seeming to follow the trend and the other two models already exhibiting higher variance. Hence, we choose `q5m14` ($ARIMA(0,1,3)x(0,1,1)_{52}$) as our best model for the fifth question.

Conclusion

As a result, our model is an $ARIMA(0,1,3)x(0,1,1)_{52}$, with seasonality of every year for weekly data $s = 52$ and first order differencing for the seasonal component and another first order differencing for the nonseasonal component. We began with a simple $ARIMA(0,1,0)x(0,1,0)_{52}$ model after witnessing that the ACF and PACF of the 1st nonseasonal differenced and 1st order seasonal differenced time series is stationary, and further differencing would over-difference the data set as evidenced by increasing standard deviation. Higher order differencing attributing to increased variance may also cause us to approximate a model with a higher order of d (nonseasonal differencing parameter) and higher D (seasonal differencing parameter). We performed model selection via AIC, BIC, residuals analysis, cross-validation, and forecast plotting, and holistically chose a more complex model with lower overfitting due to lower variance and consistently low statistics. In the end, we employed a myriad of time-domain data analysis and statistical prediction techniques.

To better optimize our results, it would be ideal to also eliminate outliers. We attempted to do this: the `detectIO()` function in the `TSA` package is tailor made to detect indices and values of outliers in the data, and the `arima()` function has an `io` argument for which we can exclude it for further analyses. However, it ran multiple errors, affecting our workflow for interpreting residuals analyses, so we decided to exclude it from the plot. Also, parameter estimation methods using `CSS` and `CSS-MLE` were used to approximate more complex models that could not have been approximated by excluding those arguments, and did not apply them to all models to minimize runtime and complexity. We also used 5-fold cross validation instead of higher folds for the seasons because we thought that the data is best captured with a smaller fold interval, and computing higher folds became too complex and ran into errors.

Appendix : Code

```
# ================================================================================
# Question 1 -----------------------------------------------------------------
# ================================================================================

# Q1 : Exploratory Data Analysis ---------------------------------------------

## Read in Data for question 1
Q1Train <- data$Q1Train

plot(Q1Train, type = 'l', xlab = "Date", ylab = "Google Data")

## Log Transformation of Data to Remove Heteroscedasticity
Q1Train.Log <- data.frame(Date = Q1Train$Date, Activity = log(Q1Train$activity + 2))
plot(Q1Train.Log, type = 'l', xlab = "Date", ylab = "Log Google Data")
q1train.log <- Q1Train.Log$Activity

# Observe first and second differenced log data
firstdiff <- diff(q1train.log)
seconddiff <- diff(q1train.log, differences = 2)
thirddiff <- diff(q1train.log, differences = 3)
```

```r
firstdiff.52 <- diff(ts(firstdiff, frequency = 52))
firstdiff.52.2 <- diff(ts(firstdiff.52, frequency = 52))
zerodiff.52 <- diff(ts(q1train.log, frequency = 52))
zerodiff.52.2 <- diff(ts(zerodiff.52, frequency = 52))

plot(q1train.log, type = 'l', xlab = "Time", ylab = "Value", main = "Undifferenced Google Data")
plot(firstdiff, type = 'l', xlab = "Time", ylab = "Value", main =  "1st Diff Google Data"); plot(seconde
plot(firstdiff.52, type = 'l', xlab = "Time", ylab = "Value", main =  "1st Diff and 1st Seasonal Diff Go

# Observe acf of data of orders 1, 2
acf(q1train.log, lag.max = 300); acf(firstdiff, lag.max = 300); acf(seconddiff, lag.max = 300); acf(firs

pacf(q1train.log, lag.max = 300); pacf(firstdiff, lag.max = 300); pacf(seconddiff, lag.max = 300); pacf

# Lowest sd usually is the best indicator
which.min(sapply(list(q1train.log, firstdiff, seconddiff, firstdiff.52, firstdiff.52.2, zerodiff.52, zer


# Q1 : Model Fitting ------------------------------------------------------------
m1 <- arima(q1train.log, order = c(0, 1, 1))
m2 <- arima(q1train.log, order = c(0,1,2))
m3 <- arima(q1train.log, order = c(0,0,1))
m4 <- arima(q1train.log, order = c(0,1,1), seasonal = list(order = c(0, 1, 1), period = 52))
m5 <- arima(q1train.log, order = c(0,1,2), seasonal = list(order = c(0, 1, 1), period = 52))
m6 <- arima(q1train.log, order = c(0,1,1), seasonal = list(order = c(0, 1, 2), period = 52))
m7 <- arima(q1train.log, order = c(0,1,2), seasonal = list(order = c(0, 1, 2), period = 52))
m8 <- arima(q1train.log, order = c(0,2,2), seasonal = list(order = c(0, 1, 2), period = 52))
m9 <- arima(q1train.log, order = c(0,1,1), seasonal = list(order = c(0, 1, 0), period = 52))
m10 <- arima(q1train.log, order = c(0,1,1),
             seasonal = list(order = c(0, 1, 1), period = 52), method = "CSS-ML")
m11 <- arima(q1train.log, order = c(0,1,2), seasonal = list(order = c(0, 1, 0), period = 52))
m12 <- arima(q1train.log, order = c(0,1,1), seasonal = list(order = c(0, 2, 2), period = 52))
#m13 <- arima(q1train.log, order = c(1,1,1), seasonal = list(order = c(1, 1, 1), period = 52))
#m13 <- arima(q1train.log, order = c(2,1,2), seasonal = list(order = c(0,1,2), period = 52))
m13 <- arima(q1train.log, order = c(0,1,2), seasonal = list(order = c(0,2,1), period = 52))
m14 <- arima(q1train.log, order = c(0,1,2), seasonal = list(order = c(0,1,3), period = 52))
# run diagnostics:
tsdiag(m1) # only one significant value in Ljung Box Statistic
tsdiag(m2) # nearly all significant from 0 for Ljung
tsdiag(m3) # way off, ACFs all significant, trend in standardized residuals
tsdiag(m4) # Ljung has all values near 0, but has insignificant acf residuals
tsdiag(m5) # all but one Ljung insignificant
tsdiag(m6) # below zero, near 0 Ljung statistic values
tsdiag(m7) # teetering above 0 line a bit for Ljung
tsdiag(m8) # close below 0 for Ljung
tsdiag(m9)
tsdiag(m10) # too close to line for Ljung
tsdiag(m11) # teetering close to 0 pval for Ljung except for 1
tsdiag(m12) # nearly all near 0 pval
tsdiag(m13) # 1 teetering, others slightly significant
tsdiag(m14) # all significantly above 0

# AIC and BIC
```

```r
m <- list(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14)
sapply(m, AIC); which.min(sapply(m, AIC)); sapply(m, BIC); which.min(sapply(m, BIC))

# K-fold Cross Validation
CVmse1 <- function(ts, order.totry = c(0L, 0L, 0L),
                         seasorder.totry = c(0L, 0L, 0L), d = 52, num.seas = 5, method = NULL){
  # computes MSE
  MSE = numeric()
  len = length(ts)
  for (k in num.seas:1) {
    train.dt = ts[1:(len - d*k)]
    test.dt = ts[(len - d*k + 1):(len - d*(k - 1))] # split into training and test data
    mod = arima(train.dt, order = order.totry,
                seasonal = list(order = seasorder.totry, period = d, method = method)) # fit model into
    fcst = predict(mod, n.ahead = d) # because we are predicting seasons, so 52 points
    MSE[k] = mean(((exp(fcst$pred) - 2) - (exp(test.dt) - 2) )^2) # reverses our previous log transform
  }
  return(MSE)
}

MSE1 <- CVmse1(q1train.log, c(0,1,1), num.seas = 8)
MSE2 <- CVmse1(q1train.log, c(0,1,2), num.seas = 8)
MSE3 <- CVmse1(q1train.log, c(0,0,1), num.seas = 8)
MSE4 <- CVmse1(q1train.log, c(0,1,1), c(0,1,1), 52, num.seas = 8)
MSE5 <- CVmse1(q1train.log, c(0,1,2), c(0,1,1), 52, num.seas = 8)
MSE6 <- CVmse1(q1train.log, c(0,1,1), c(0,1,2), 52, num.seas = 8)
MSE7 <- CVmse1(q1train.log, c(0,1,2), c(0,1,2), 52, num.seas = 8)
MSE8 <- CVmse1(q1train.log, c(0,2,2), c(0,1,2), 52, num.seas = 8)
MSE9 <- CVmse1(q1train.log, c(0,1,1), c(0,1,0), 52, num.seas = 8)
MSE10 <- CVmse1(q1train.log, c(0,1,1), c(0,1,1), 52, num.seas = 8)
MSE11 <- CVmse1(q1train.log, c(0,1,2), c(0,1,0), 52, num.seas = 8)
MSE12 <- CVmse1(q1train.log, c(0,1,1), c(0,2,2), 52, num.seas = 8)
MSE13 <- CVmse1(q1train.log, c(0,1,2), c(0,2,1), 52, num.seas = 8)
MSE14 <- CVmse1(q1train.log, c(0,1,2), c(0,1,3), 52, num.seas = 8)

# Q1 : Predictions ------------------------------------------------------------

q1mse <- sapply(1:14, function(x) paste0("MSE",x))
q1meanMSE <- apply(ldply(q1mse, function(z) get(z)), 1, function(y) mean(y))
which.min(apply(ldply(q1mse, function(z) get(z)), 1, function(y) print(mean(y)))) # smallest MSE is mod

predictions6 <- exp(predict(m6, n.ahead = 104)$pred) - 2 # lowest MSE for CV ARIMA(0,1,1)x(0,1,2)_52
predictions7 <- exp(predict(m7, n.ahead = 104)$pred) - 2 # lowest BIC ARIMA(0,1,2)x(0,1,2)_52
predictions14 <- exp(predict(m14, n.ahead = 104)$pred) - 2 # lowest AIC ARIMA(0,1,2)x(0,1,3)_52

plotPred(exp(q1train.log) - 2, predictions6); plotPred(exp(q1train.log) - 2, predictions7); plotPred(ex

# ============================================================================
# Question 2 ------------------------------------------------------------------
# ============================================================================

# Q2 : Exploratory Data Analysis ----------------------------------------------
train2 = read.csv("q2_train.csv",header = T)
```

```r
data = train2$activity[1:525]
newtime = time(data)
plot(data, type='l', xlab="time",ylab = "Activity")
newd = log(data+2)
min(newd)
plot(newd, type = 'l', xlab = "time",ylab = "sqrtdata")
diff1 = diff(newd)
sd(diff1)
plot(diff1,type = "l", xlab = "time", ylab="diff1")
acf(diff1, lag.max = 100, type = "correlation", plot = T, main = "Correlogram of 1st Differenced Data",
pacf(diff1,lag.max = 100)
#x <- ts(diff1, frequency=52)
#fit <- tbats(x)
#seasonal <- !is.null(fit$seasonal)

#taking the seasonality out
diff3 = diff(diff1,52)
plot(diff3,type = "l", xlab = "time", ylab="diff3")
acf(diff3, lag.max = 200, type = "correlation", plot = T, main = "Correlogram of 3nd Differenced Data",
pacf(diff3,lag.max = 200)

auto.arima(data)

# Q2 : Model Fitting ----------------------------------------------------------

#jung box should be: higher the early one , lower the later one.
#only works correlation between lags. if your residuals are white noise. so no correlated lags.
#auto.arima(data)
#or = (0,1,2)
#ar=p =(1, 1, 2) seasonal = list(order= c(1,1))
#significant lag 1 for ma =0,1 in seasonal for acf
#significnat lag 1 for ar = 0,1,2 in seasonal pacf
#difference twice for the lower model, and seasonal difference for one model
#lower model you have pacf tails off instead of cut off
#while acf cuts off at lag1
#m0 <- arima(newd,order= c(0,1,1))
#tsdiag(m0)

m1 <- arima(newd,order = c(0,1,2),seasonal = list(order= c(0,1,1),period = 52))#trend existence +p valu
tsdiag(m1)
m2 <- arima(newd,order = c(0,1,2),seasonal = list(order= c(1,1,1),period = 52))
m2$sigma2 #0.00924647 #trend existence +p value decreases too quickly
tsdiag(m2)
m3 <- arima(newd,order = c(0,1,2),seasonal = list(order= c(2,1,1),period = 52)) #trend existence +p val
tsdiag(m3)
m4 <- arima(newd,order = c(0,1,2),seasonal = list(order= c(2,1,0),period = 52)) #trend existence +p val
tsdiag(m4)
m5 <- arima(newd,order = c(0,1,2),seasonal = list(order= c(1,1,0),period = 52)) #trend existence +p val
tsdiag(m5)
m6 <- arima(newd,order = c(0,1,2),seasonal = list(order= c(0,1,0),period = 52))#pvalue increases, too m
tsdiag(m6)
m7 <- arima(newd,order = c(1,1,2),seasonal = list(order= c(0,1,1),period = 52))#ACF p decreases too qui
m7$sigma2#0.009101095
```

```
tsdiag(m7)
#Does
m8 <- arima(newd,order = c(1,1,2),seasonal = list(order= c(1,1,1),period = 52)) #a trend
tsdiag(m8)
m9 <- arima(newd,order = c(1,1,2),seasonal = list(order= c(2,1,1),period = 52),method= 'CSS') #a trend
tsdiag(m9)
m10 <- arima(newd,order = c(1,1,2),seasonal = list(order= c(2,1,0),period = 52),method='CSS') #ideal bu
tsdiag(m10)
m11 <- arima(newd,order = c(1,1,2),seasonal = list(order= c(1,1,0),period = 52)) #a trend
tsdiag(m11)
m12 <- arima(newd,order = c(1,1,2),seasonal = list(order= c(0,1,0),period = 52))#standardize residuals
tsdiag(m12)

#all of them does not have significance lag in acf.

#7 is appears as the best model under AIC and BIC due to low value. m8 is the second best one for AIC a
#and m1 ,m8 is third best for AIC

sapply(list(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12), AIC); which.min(sapply(list(m1,m2,m3,m4,m5,m6,m7,m

#not working
sapply(list(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12), BIC); which.min(sapply(list(m1,m2,m3,m4,m5,m6,m7,m

computeCVmse <- function(order.totry = c(0L, 0L, 0L),
                         seasorder.totry = c(0L, 0L, 0L),method = NULL){
  MSE = numeric()
  len = length(newd)
  for(k in 9:1) {
    train.dt = newd[1:(len - 52*k)]
    test.dt = newd[(len - 52*k + 1):(len - 52*(k-1))]
    mod = arima(train.dt, order = order.totry,
                seasonal = list(order = seasorder.totry, period = 52))
    fcst = predict(mod, n.ahead = 52)
    MSE[k] = mean(((exp(fcst$pred)-2)- (exp(test.dt)-2))^2)
  }
  return(MSE)
}

MSE1 <- computeCVmse(c(0,1,2),c(0,1,1))
MSE2 <- computeCVmse(c(0,1,2),c(1,1,1),'CSS')#not working
MSE3 <- computeCVmse(c(0,1,2),c(2,1,1),'CSS')#not working
MSE4<-computeCVmse(c(0,1,2),c(2,1,0),'CSS')#not working
MSE5 <- computeCVmse(c(0,1,2), c(1,1,0),'CSS')#not working
MSE6<-computeCVmse(c(0,1,2),c(0,1,0))
MSE7 <- computeCVmse(c(1,1,2),c(0,1,1),'CSS')#not working
MSE8 <- computeCVmse(c(1,1,2),c(1,1,1)) #not working
MSE9 <- computeCVmse(c(1,1,2),c(2,1,1), method ='CSS') #not working
MSE10<-computeCVmse(c(1,1,2),c(2,1,0),method ='CSS') #not working
MSE11 <- computeCVmse(c(1,1,2), c(1,1,0)) #not working
MSE12<-computeCVmse(c(1,1,2),c(0,1,0), method = 'CSS')#not working
#only 1 and 6 are working

a1=mean(MSE1)
```

```r
a6=mean(MSE6)


#the lowest one is a1, with MSE of 0.1914. This does not matche the AIC and BIC results.But since m1 ma
#might be the the best model, m7 has large MSE compared with a1, which is 0.28546. The second lowest on
#to m1, m6 and m7
min(c(a1,a6))
# Q2 : Predictions ----------------------------------------------------------------

predictions1 <- exp(predict(m1, n.ahead = 104)$pred) - 2
predictions2 <- exp(predict(m2, n.ahead = 104)$pred) - 2
predictions3 <- exp(predict(m3, n.ahead = 104)$pred) - 2
predictions4 <- exp(predict(m4, n.ahead = 104)$pred) - 2
predictions5 <- exp(predict(m5, n.ahead = 104)$pred) - 2
predictions6 <- exp(predict(m6, n.ahead = 104)$pred) - 2
predictions7 <- exp(predict(m7, n.ahead = 104)$pred) - 2
predictions8 <- exp(predict(m8, n.ahead = 104)$pred) - 2
predictions9 <- exp(predict(m9, n.ahead = 104)$pred) - 2
predictions10 <- exp(predict(m10, n.ahead = 104)$pred) - 2
predictions11 <- exp(predict(m11, n.ahead = 104)$pred) - 2
predictions12 <- exp(predict(m12, n.ahead = 104)$pred) - 2


plotPred <- function(ts, pred) {
  plot(1:(length(ts) + length(pred)), c(ts, pred), type = 'l', col = 1); points((length(ts) + 1) : (leng
}

plotPred(exp(newd) - 2, predictions1)
plotPred(exp(newd) - 2, predictions2)
plotPred(exp(newd) - 2, predictions3)
plotPred(exp(newd) - 2, predictions4)
plotPred(exp(newd) - 2, predictions5)
plotPred(exp(newd) - 2, predictions6)
plotPred(exp(newd) - 2, predictions7)
plotPred(exp(newd) - 2, predictions8)
plotPred(exp(newd) - 2, predictions9)
plotPred(exp(newd) - 2, predictions10)
plotPred(exp(newd) - 2, predictions11)
plotPred(exp(newd) - 2, predictions12)


writeData(predictions7, q.num = 2, "Veronika", "Yang", 3032577302)

#so the model cut down to either m2, and m7. lets compared with all the properties.
#m2, has higher sigma value which is 0.00924647 compared with m7 which is 0.0091. While m2 has MSE of 0
#than m7 which has the lowest one of 0.28546. this means that m2 or m7 are potentially not over fitting
#is the smallest among all the models. Based on the diagram. both of them are similar to each other- st
#ACF shows white noise exists (ideal) and jungbox are greatly larger than 0.05. indepedent lag.
# Although based on the prediction plot, m2 seems better than m7, but comparing all other models, we de


# ============================================================================
# Question 3 ----------------------------------------------------------------
```

```r
# ================================================================================

# Q3 : Exploratory Data Analysis ------------------------------------------------

train = read.csv("q3_train.csv",header = T)
data = train$activity[1:525]
newtime = time(data)
plot(data, type='l', xlab="time",ylab = "Activity")
newd = sqrt(data+1.5)
min(newd)
plot(newd, type = 'l', xlab = "time",ylab = "sqrtdata")
diff1 = diff(newd)
sd(diff1)
plot(diff1,type = "l", xlab = "time", ylab="diff1")
acf(diff1, lag.max = 100, type = "correlation", plot = T, main = "Correlogram of 1st Differenced Data",
pacf(diff1,lag.max = 100)
#x <- ts(diff1, frequency=52)
#fit <- tbats(x)
#seasonal <- !is.null(fit$seasonal)

#taking the seasonality out
diff3 = diff(diff1,52)
plot(diff3,type = "l", xlab = "time", ylab="diff3")
acf(diff3, lag.max = 200, type = "correlation", plot = T, main = "Correlogram of 3nd Differenced Data",
pacf(diff3,lag.max = 200)

#jung box should be: higher the early one , lower the later one.
#only works correlation between lags. if your residuals are white noise. so no correlated lags.
#auto.arima(data)
#ar=p =(3, 1, 1,) seasonal = list(order= c(1,1))
#significant lag 1 for ma =0,1,2 in seasonal for acf
#significnat lag 1 for ar = 0,1,2 in seasonal pacf
#difference twice for the lower model, and seasonal difference for one model
#lower model you have pacf tails off instead of cut off
#while acf cuts off at lag1

# Q3 : Model Fitting ------------------------------------------------------------

m1 <- arima(newd,order = c(3,1,1),seasonal = list(order= c(0,1,1),period = 52))#pvalue increases
#Standardized Residuals is more performed as white noise and the ACF of residuals are within the Confid
tsdiag(m1)
#m1$sigma2 = 0.006491645
m2 <- arima(newd,order = c(3,1,1),seasonal = list(order= c(1,1,0),period = 52),method = 'CSS')#pvalue i
tsdiag(m2)
m3 <- arima(newd,order = c(3,1,1),seasonal = list(order= c(0,1,2),period = 52),method = 'CSS')#p value
#Standardized Residuals is more performed as white noise and the ACF of residuals are within the Confid
#m3$sigma2 = 0.006698322
tsdiag(m3)
m4 <- arima(newd,order = c(3,1,1),seasonal = list(order= c(2,1,0),period = 52),method = 'CSS')#p value
tsdiag(m4)
m5 <- arima(newd,order = c(3,1,1),seasonal = list(order= c(2,1,1),period = 52),method = 'CSS')#p value
tsdiag(m5)
m6 <- arima(newd,order = c(3,1,1),seasonal = list(order= c(1,1,2),period = 52),method = 'CSS')#pvalue i
```

```r
tsdiag(m6)
m7<-arima(newd,order = c(3,1,1),seasonal = list(order= c(1,1,1),period = 52),method = 'CSS')#p value in
tsdiag(m7)
m8 <-arima(newd,order = c(3,1,1),seasonal = list(order= c(2,1,2),period = 52),method = 'CSS')#good, ACF
tsdiag(m8)
#m8$sigma2 = 0.00658078, MSE= sigma^2 +bias^2. so sigma is low and we may overfit.

#not working
sapply(list(m1,m2,m3,m4,m5,m6,m7,m8), AIC); which.min(sapply(list(m1,m2,m3,m4,m5,m6,m7,m8), AIC))
#not working
sapply(list(m1,m2,m3,m4,m5,m6,m7,m8), BIC); which.min(sapply(list(m1,m2,m3,m4,m5,m6,m7,m8), BIC))

#Due to css method, for both AIC and BIC we only generated one result for 1. it may not be applicable.
#so therefore, we decide to use CV to perform a further check.


computeCVmse <- function(order.totry = c(0L, 0L, 0L),
                         seasorder.totry = c(0L, 0L, 0L),method = NULL){
  MSE = numeric()
  len = length(newd)
  for(k in 5:1) {
    train.dt = newd[1:(len - 52*k)]
    test.dt = newd[(len - 52*k + 1):(len - 52*(k-1))]
    mod = arima(train.dt, order = order.totry,
                seasonal = list(order = seasorder.totry, period = 52))
    fcst = predict(mod, n.ahead = 52)
    MSE[k] = mean( ((fcst$pred^2-1.5)- ((test.dt)^2-1.5))^2 )
  }
  return(MSE)
}


MSE1 <- computeCVmse(c(3,1,1),c(0,1,1))
a1=mean(MSE1) #a1 = 0.0971
MSE3 <- computeCVmse(c(3,1,1),c(0,1,2))
a3=mean(MSE3)#a3 = 0.9765
a1=mean(MSE1) #a1 = 0.0971
MSE2 <- computeCVmse(c(3,1,1),c(1,1,0)) #data is not working
MSE3 <- computeCVmse(c(3,1,1),c(0,1,2))
a3=mean(MSE3)#a3 = 0.9765
MSE4<-computeCVmse(c(3,1,1),c(2,1,0))#not working
MSE5 <- computeCVmse(c(3,1,1), c(2,1,1))#not working
MSE6<-computeCVmse(c(3,1,1),c(1,1,2))#not working
MSE7<-computeCVmse(c(3,1,1),c(1,1,1))#not working
MSE8<-computeCVmse(c(3,1,1),c(2,1,2))#not working


a1=mean(MSE1)
a3=mean(MSE3)


min(c(a1,a2,a3,a4,a5,a6,a7,a8))
```

```r
#a1 seems like the best model out of 8.

# Q3 : Predictions ------------------------------------------------------------

predictions1 <- (predict(m1, n.ahead = 104)$pred)^2 - 1.5
predictions2 <- (predict(m2, n.ahead = 104)$pred)^2 - 1.5
predictions3 <- (predict(m3, n.ahead = 104)$pred)^2 - 1.5
predictions4 <- (predict(m4, n.ahead = 104)$pred)^2 - 1.5
predictions5 <- (predict(m5, n.ahead = 104)$pred)^2 - 1.5
predictions6 <- (predict(m6, n.ahead = 104)$pred)^2 - 1.5
predictions7 <- (predict(m7, n.ahead = 104)$pred)^2 - 1.5
predictions8 <- (predict(m8, n.ahead = 104)$pred)^2 - 1.5


plotPred <- function(ts, pred) {
  plot(1:(length(ts) + length(pred)), c(ts, pred), type = 'l', col = 1); points((length(ts) + 1) : (leng
}

plotPred((newd)^2 - 1.5, predictions1)
plotPred((newd)^2 - 1.5, predictions2)
plotPred((newd)^2 - 1.5, predictions3)
plotPred((newd)^2 - 1.5, predictions4)
plotPred((newd)^2 - 1.5, predictions5)
plotPred((newd)^2 - 1.5, predictions6)
plotPred((newd)^2 - 1.5, predictions7)
plotPred((newd)^2 - 1.5, predictions8)

#Comparing with all the method, we did not use AIC and BIC since it doesn't work under CSS method.We th
#In the tsdiagram, both of m3 and m1 does not against the null hypothesis under all the lags are indepe
#the standadize residuals does not show any trend and also there are no significant lags present in the
#comparing with m3, m1 is choosing as the best model. The reason is given as follwoing. m1 has lower si
#However, m1 slightly higher MSE of 0.0759 than m3 0.0755, this means that m3 is overfitting while m1 m
#further validates my guess of overfitting, and therefore we decided to choose m1.
#the prediction plots of m1 and m3 are similar to each other, both of them follow the initial trend wit
 writeData(predictions1, q.num = 3, "Veronika", "Yang", 3032577302)


# ============================================================================
# Question 5 ------------------------------------------------------------------
# ============================================================================
Q5Train <- data$Q5Train
q5train <- Q5Train$activity

# Q5 : Exploratory Data Analysis ----------------------------------------------
plot(Q5Train, type = 'l', xlab = "Date", ylab = "Google Data", main = "Figure 1: Original Question 5 Tra
par(mfrow = c(2,2)); plot(q5train, type = 'l', main = "Original"); plot(log(q5train + 2), type = 'l', ma

# log transform increases variance so we will not perform log transform
q5diff1 <- diff(q5train, differences = 1); q5diff2 <- diff(q5train, differences = 2); q5diff3 <- diff(q5
q5diff1.52 <- diff(q5diff1, lag = 52); q5diff2.52 <- diff(q5diff2, lag = 52); q5diff3.52 <- diff(q5diff3
par(mfrow = c(3,2)); plot(q5train, type = 'l', xlab = "Time", ylab = "Value", main = "Undifferenced Goog
which.min(sapply(list(q5train, q5diff1, q5diff2, q5diff1.52),
                 function(x) print(sd(x)))) # q5diff1.52 is lowest
```

```r
par(mfrow = c(3,2)); acf(q5train, lag.max = 300); acf(q5diff1, lag.max = 300); acf(q5diff2, lag.max = 3

par(mfrow = c(3,2)); pacf(q5train, lag.max = 300); pacf(q5diff1, lag.max = 300); pacf(q5diff2, lag.max =

# Q5 : Model Fitting ------------------------------------------------------------
q5m1 <- arima(q5train, order = c(0,1,0), seasonal = list(order = c(0,1,0), period = 52))
q5m2 <- arima(q5train, order = c(0,1,0), seasonal = list(order = c(0,1,1), period = 52))
q5m3 <- arima(q5train, order = c(0,1,0), seasonal = list(order = c(0,1,2), period = 52))
q5m4 <- arima(q5train, order = c(0,1,0), seasonal = list(order = c(0,1,3), period = 52))
q5m5 <- arima(q5train, order = c(0,1,1), seasonal = list(order = c(0,1,0), period = 52))
q5m6 <- arima(q5train, order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 52))
q5m7 <- arima(q5train, order = c(0,1,1), seasonal = list(order = c(0,1,2), period = 52))
q5m8 <- arima(q5train, order = c(0,1,1), seasonal = list(order = c(0,1,3), period = 52))
q5m9 <- arima(q5train, order = c(0,1,2), seasonal = list(order = c(0,1,0), period = 52))
q5m10 <- arima(q5train, order = c(0,1,2), seasonal = list(order = c(0,1,1), period = 52))
q5m11 <- arima(q5train, order = c(0,1,2), seasonal = list(order = c(0,1,2), period = 52))
q5m12 <- arima(q5train, order = c(0,1,2), seasonal = list(order = c(0,1,3), period = 52))
q5m13 <- arima(q5train, order = c(0,1,3), seasonal = list(order = c(0,1,0), period = 52))
q5m14 <- arima(q5train, order = c(0,1,3), seasonal = list(order = c(0,1,1), period = 52))
q5m15 <- arima(q5train, order = c(0,1,3), seasonal = list(order = c(0,1,2), period = 52))
q5m16 <- arima(q5train, order = c(0,1,3), seasonal = list(order = c(0,1,3), period = 52))
q5m <- sapply(1:16, function(x) paste0("q5m",x))
q5diag <- lapply(q5m, function(x) tsdiag(get(x))) # Residuals, ACF, Ljung-Box Test
CVmse5 <- function(ts, order.totry = c(0L, 0L, 0L),
                        seasorder.totry = c(0L, 0L, 0L), d = 52, num.seas = 5, method = NULL){
  # computes MSE
  MSE = numeric()
  len = length(ts)
  for (k in num.seas:1) {
    train.dt = ts[1:(len - d*k)]
    test.dt = ts[(len - d*k + 1):(len - d*(k - 1))] # split into training and test data
    mod = arima(train.dt, order = order.totry,
                seasonal = list(order = seasorder.totry, period = d, method = method)) # fit model into
    fcst = predict(mod, n.ahead = d) # because we are predicting seasons, so 52 points
    MSE[k] = mean((fcst$pred - test.dt )^2) # reverses our previous log transform
  }
  return(MSE)
}

q5MSE1 <- CVmse5(q5train, c(0,1,0), c(0,1,0),52) ; q5MSE2 <- CVmse5(q5train, c(0,1,0), c(0,1,1),52)
q5MSE3 <- CVmse5(q5train, c(0,1,0), c(0,1,2),52); q5MSE4 <- CVmse5(q5train, c(0,1,0), c(0,1,3),52)
q5MSE5 <- CVmse5(q5train, c(0,1,1), c(0,1,0),52); q5MSE6 <- CVmse5(q5train, c(0,1,1), c(0,1,1),52)
q5MSE7 <- CVmse5(q5train, c(0,1,1), c(0,1,2),52); q5MSE8 <- CVmse5(q5train, c(0,1,1), c(0,1,3),52)
q5MSE9 <- CVmse5(q5train, c(0,1,2), c(0,1,0),52); q5MSE10 <- CVmse5(q5train, c(0,1,2), c(0,1,1),52)
q5MSE11 <- CVmse5(q5train, c(0,1,2), c(0,1,2),52); q5MSE12 <- CVmse5(q5train, c(0,1,2), c(0,1,3),52)
q5MSE13 <- CVmse5(q5train, c(0,1,3), c(0,1,0),52); q5MSE14 <- CVmse5(q5train, c(0,1,3), c(0,1,1),52)
q5MSE15 <- CVmse5(q5train, c(0,1,3), c(0,1,2),52); q5MSE16 <- CVmse5(q5train, c(0,1,3), c(0,1,3),52)

q5mse <- sapply(1:16, function(x) paste0("q5MSE",x))

q5meanMSE <- apply(ldply(q5mse, function(z) get(z)), 1, function(y) mean(y))

#which.min(apply(ldply(q5mse, function(z) get(z)), 1, function(y) print(mean(y)))) # lowest mean MSE is
```

```r
#which.min(apply(ldply(list(MSE1, MSE2, MSE3,MSE4,MSE5,MSE6,MSE7,MSE8,MSE9, MSE10, MSE11, MSE12, MSE13,
AICBICCV <- as.matrix(data.frame(Name = sapply(1:16, function(x) paste0("q5m",x)),
                     Model= sapply(q5m, function(x) as.character(get(x))),
                     AIC = sapply(q5m, function(x) AIC(get(x))),
                     BIC = sapply(q5m, function(x) BIC(get(x))),
                     CV_mean = sapply(q5mse,function(x) mean(get(x)))))
DT::datatable(AICBICCV, caption = "Figure 7: Model Fitting Analysis")

# Q5 : Predictions ------------------------------------------------------------
q5pred6 <- predict(q5m6, n.ahead = 104)$pred
q5pred7 <- predict(q5m7, n.ahead = 104)$pred
q5pred14 <- predict(q5m14, n.ahead = 104)$pred

plotPred <- function(ts, pred, model, title = NULL, x = NULL, y = NULL) {
  plot(1:(length(ts) + length(pred)), c(ts, pred), type = 'l', col = 1, main = title, xlab = x,ylab = y
}

par(mfrow = c(3,1)); plotPred(q5train, q5pred6, title = "ARIMA(0,1,1)(0,1,1)_{52} Predictions", y = "Va

q5table <- as.matrix(data.frame(Name = c("q5m6","q5m7","q5m14"), Model = sapply(list(q5m6,q5m7,q5m14), a
                    Variance = as.numeric(c(q5m6$sigma2, q5m7$sigma2, q5m14$sigma2)),
                    AIC = as.numeric(c(sapply(list(q5m6,q5m7,q5m14), AIC))),
                    BIC = as.numeric(c(sapply(list(q5m6,q5m7,q5m14), BIC))),
                    CV = as.numeric(c(sapply(list(q5MSE6, q5MSE7, q5MSE14),mean)))) )
DT::datatable(q5table, caption = "Figure 9: Question 5 Statistics")

writeData(predictions14, q.num = 1, "Samba", "Njie", 23075185) #3.39752
writeData(predictions1, q.num = 2, "Veronika", "Yang", 3032577302)
writeData(predictions1, q.num = 3, "Veronika", "Yang", 3032577302)
writeData(q5pred14, q.num = 5, "Samba", "Njie", 23075185)
```