DATA 624: Project 1

Vinicio Haro
Sang Yoon (Andy) Hwang
Julian McEachern
Jeremy O'Brien
Bethany Poulin

October 22, 2019

Contents

Overview

We split the work into three sections for Project 1. Individual team members each took lead on individual problem. Jermey and Julian focused on Part A, Sang Yoon (Andy) and Vinicio worked on Part B, and Bethany took lead on Part C. Juliann created an overall format for the assignment to be used and all team members collectively worked together on reviewing and merging our finished product.

Dependencies

The following R libraries were used to complete this assignment:

```
library(easypackages)

libraries('knitr', 'kableExtra', 'default')

# Processing
libraries('readxl', 'tidyverse', 'janitor', 'imputeTS', 'tsoutliers')

# Timeseries
libraries('urca', 'forecast', 'timetk')

# Graphing
libraries('ggplot2', 'grid', 'gridExtra', 'ggfortify','ggpubr', 'scales')
```

Data

Data was stored within our group repository and imported below using the `readxl` package. Each individual question was solved within an R script and the data was sourced into our main report. For replication purposes, we also made our R scripts available within our appendix. All forecasts were exported and saved a `.csv` file in our [github repository]((https://github.com/JeremyOBrien16/CUNY_DATA_624/tree/master/Project%20One/) folder named forecasts.

```
# Data Aquisition
atm_data <- read_excel("data/ATM624Data.xlsx")
power_data <- read_excel("data/ResidentialCustomerForecastLoad-624.xlsx")
pipe1_data <- read_excel("data/Waterflow_Pipe1.xlsx")
pipe2_data <- read_excel("data/Waterflow_Pipe2.xlsx")

# Source Code
source('~/GitHub/CUNY_DATA_624/Project One/scripts/Part-A.R')
source('~/GitHub/CUNY_DATA_624/Project One/scripts/Part-B.R')
source('~/GitHub/CUNY_DATA_624/Project One/scripts/Part-C.R')
```

1 Part A: ATMs

Instructions: In part A, I want you to forecast how much cash is taken out of 4 different ATM machines for May 2010. The data is given in a single file. The variable `Cash` is provided in hundreds of dollars, other than that it is straight forward. I am being somewhat ambiguous on purpose. I am giving you data, please provide your written report on your findings, visuals, discussion and your R code all within a Word readable document, except the forecast which you will put in an Excel readable file. I must be able to cut and paste your R code and run it in R studio. Your report must be professional - most of all - readable, EASY to follow. Let me know what you are thinking, assumptions you are making! Your forecast is a simple CSV or Excel file that MATCHES the format of the data I provide.

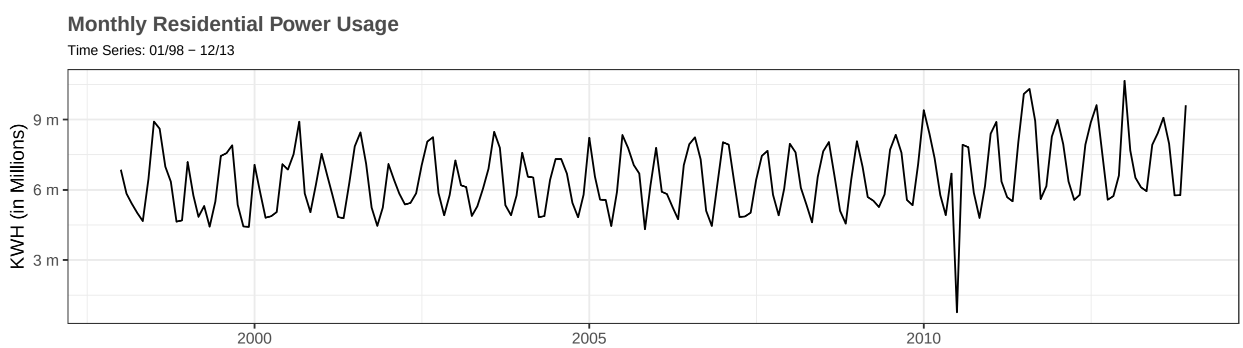
2 Part B: Forecasting Power

Instructions: Part B consists of a simple dataset of residential power usage for January 1998 until December 2013. Your assignment is to model these data and a monthly forecast for 2014. The data is given in a single file. The variable 'KWH' is power consumption in Kilowatt hours, the rest is straight forward. Add these to your existing files above - clearly labeled.

2.1 Exploration

We observed there was a missing value in September 2008. We used imputation method called na.interpolation which performs a technique in numerical analysis which estimates a value from known data points. For our case, linear method using first order Taylor polynomial is used.

2.2 Time Series Plot



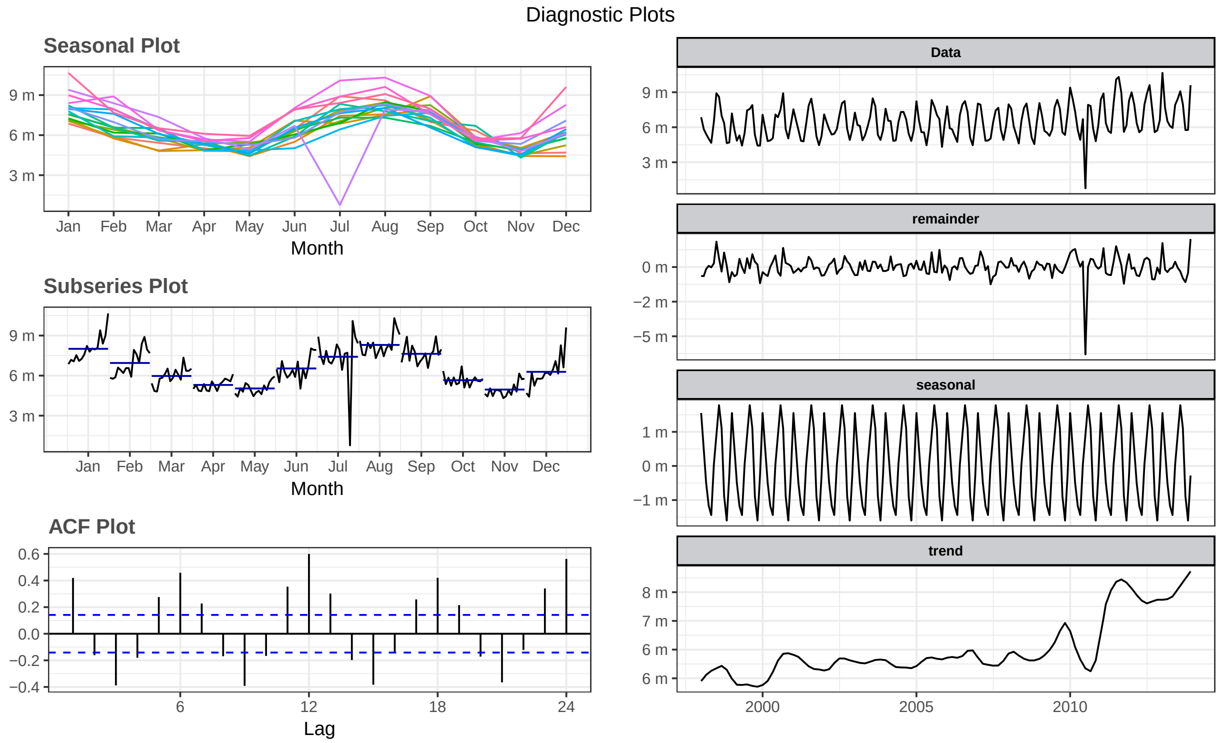
Our initial time series plot reveal annual seasonality within this time series. The box plot/seasonality plot actually reveals where power consumption fluctuations occur within each of the cycle positions. We can speculate that this could be due to there being no major Holidays that require power draining decor plus we assume minimal AC usage during the cold months.

2.3 Evaluation

We see power consumption increase between the months of June and August. This must be tied to AC usage during the warmer months of a year and finally power usage dips from September to Novemeber with a small spike in December. We speculate that thisis due to transitioning out of summer. The spike in December could be connected to the usage or Holiday lights being kept on.

Within the overall TS plot, we see a dip in July 2010. This could be due to a power outtage during a hot summer month. This can certainly be considered to be an outlier within this TS. Using TSOutliers, we can actually identify the index where our outliers may be. TSoutliers also replaces the outlier using Box-Cox. If set lambda=auto, then TSoutliers will automatically perform Box-Cox transformation.

The ACF plot shows that autocorrelations are well outside the significant space indicating the series is not white noise, non-stationary.



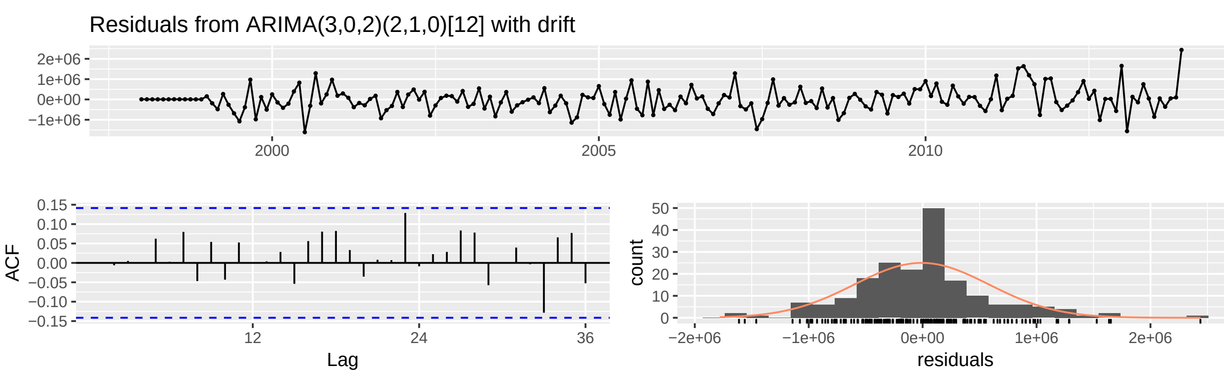
2.4 Data Model

Out of the models we built, we can make some preliminary observations. The residuals for each of our models does not have a major deviance from normality, however residuals of Model #1: ARIMA do not have an extended number of bins distorting the normality proximity but we can say it is still fairly normally distributed.

The residual ACF plots show residual autocorrelations for each of our models. Model #1: ARIMA has less autocorrelation than the other three models. Model 1 is well within the 95% limits indicated by the dotted blue lines.

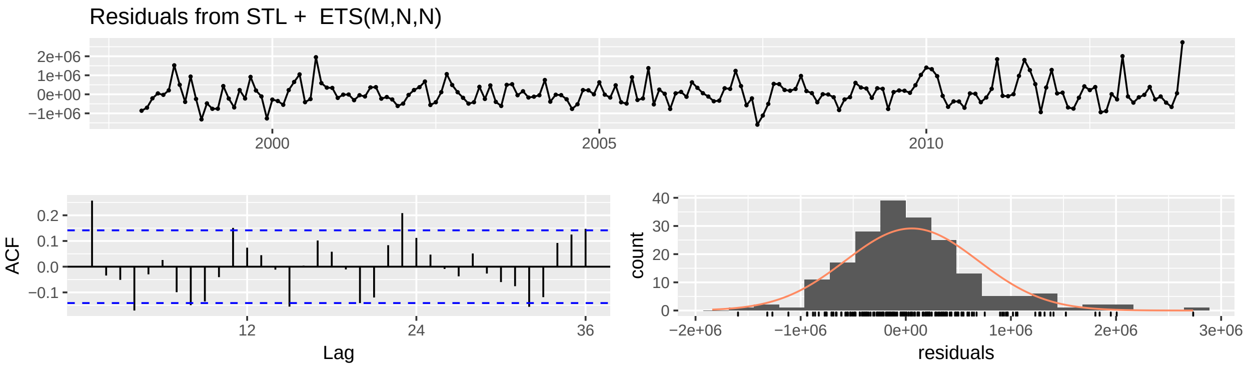
If we examine the Ljung-Box test results for our models, the only model with a p-value > 0.05 is Model #1: ARIMA. This implies that the residuals from other models are not independent, hence not white noise. The full model summary can be viewed in the appendix.

2.4.1 Model #1: ARIMA



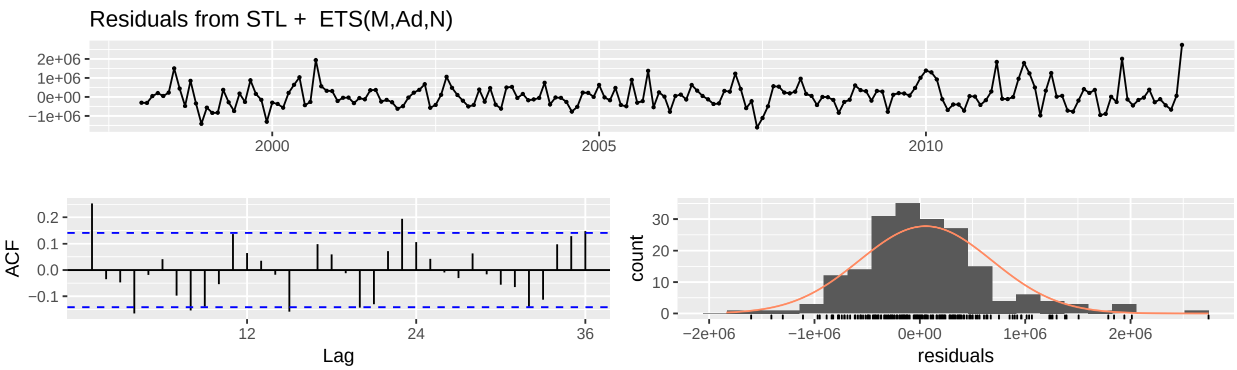
```
FALSE
FALSE   Ljung-Box test
FALSE
FALSE data:  Residuals from ARIMA(3,0,2)(2,1,0)[12] with drift
FALSE Q* = 12.555, df = 16, p-value = 0.705
FALSE
FALSE Model df: 8.   Total lags used: 24
```

2.4.2 Model #2: STL (no-demped) - MNN



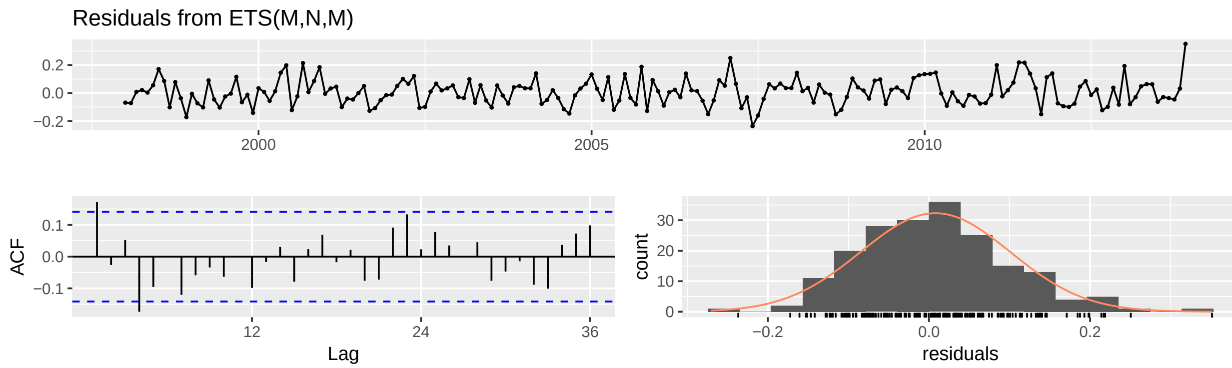

```
FALSE
FALSE   Ljung-Box test
FALSE
FALSE data:  Residuals from STL +  ETS(M,N,N)
FALSE Q* = 65.934, df = 22, p-value = 2.84e-06
FALSE
FALSE Model df: 2.   Total lags used: 24
```

2.4.3 Model #2-2: STL (demped) - MAdN




```
FALSE
FALSE   Ljung-Box test
FALSE
FALSE data:  Residuals from STL +  ETS(M,Ad,N)
FALSE Q* = 63.375, df = 19, p-value = 1.119e-06
FALSE
FALSE Model df: 5.   Total lags used: 24
```

2.4.4 Model #3: ets - MNM



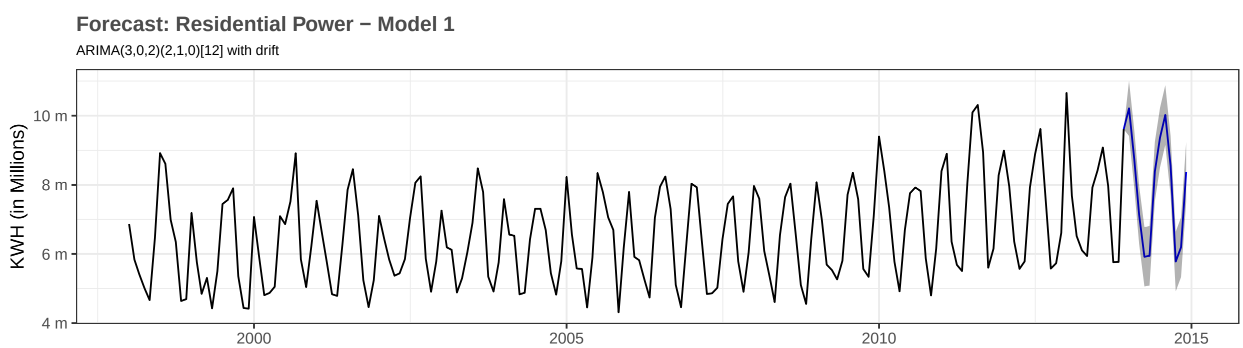

```
FALSE
FALSE 	Ljung-Box test
FALSE
FALSE data:  Residuals from ETS(M,N,M)
FALSE Q* = 32.042, df = 10, p-value = 0.000394
FALSE
FALSE Model df: 14.   Total lags used: 24
```

2.5 Forecast

The `auto.arima()` function performs cross validation on hyperparameter tuning to find the best model with parameters of `order` and `seasonal` that minimize `AIC`. This gave us **arima_model**: ARIMA$(3,0,2)(2,1,0)12$ with drift resulting `AIC` = 5332.24.

Since ARIMA is the only reliable model, as other models failed Ljung test, we will plot forecasts of ARIMA only. The forecasted values can be viewed in the appendix.

2.5.1 Model #1: ARIMA



2.6 Discussion

We implemented a cross validation method of testing for `h=12`. The process randomly chooses 12 points to measure and take the average of RMSEs. By definition, a lower RMSE on test set is attributed with a better forecast on unseen data.

Using Time series cross-validation, we compute RMSE on testset (`h=12`). We would have to pick the model with the lowest RMSE on test set as our final model if we had more than 1 model to compare. In our case, since we only have 1 model left after Ljung test, we have no choice but to pick seasonal ARIMA model as our final choice. Cross-validation test shows that RMSE on test is around 720k when RMSE on training is around 589k. We can conclude the model is not necessarily overfitted. Given that MAPE on training is less than 7, it is not a suprising result.

```
FALSE [1] "RMSE - train: 589381.7"
```

```
FALSE [1] "RMSE - test: 725175"
```

3 Part C: Waterflow

Instructions: Part C consists of two data sets. These are simple 2 columns sets, however they have different time stamps. Your optional assignment is to time-base sequence the data and aggregate based on hour (example of what this looks like, follows). Note for multiple recordings within an hour, take the mean. Then to test appropriate assumptions and forecast a week forward with confidence bands (80 and 95%). Add these to your existing files above - clearly labeled.

3.1 Pipes1 Forecast

3.2 Pipes2 Forecast

3.3 R Script

```
#Insert Script Here
```

4 Appendix

4.1 Part B

4.1.1 Model Summary

ARIMA:

```
FALSE Series: ts_data_o
FALSE ARIMA(3,0,2)(2,1,0)[12] with drift
FALSE
FALSE Coefficients:
FALSE           ar1      ar2     ar3     ma1     ma2     sar1     sar2     drift
FALSE       -0.5606  -0.2216  0.3284  0.8902  0.4827  -0.7249  -0.4152  9018.405
FALSE s.e.   0.3992   0.3382  0.0960  0.4120  0.4551   0.0797   0.0841  3027.685
FALSE
FALSE sigma^2 estimated as 3.878e+11:  log likelihood=-2657.12
FALSE AIC=5332.24   AICc=5333.3   BIC=5360.97
FALSE
FALSE Training set error measures:
FALSE                     ME     RMSE      MAE        MPE     MAPE      MASE
FALSE Training set -8455.077 589381.7 427752.5 -0.7944782 6.475365 0.6904053
FALSE                      ACF1
FALSE Training set 0.0006090194
```

STL - MNN:

```
FALSE
FALSE Forecast method: STL +  ETS(M,N,N)
FALSE
FALSE Model Information:
FALSE ETS(M,N,N)
FALSE
FALSE Call:
FALSE  ets(y = x, model = etsmodel, allow.multiplicative.trend = allow.multiplicative.trend)
FALSE
FALSE   Smoothing parameters:
FALSE     alpha = 0.1159
FALSE
FALSE   Initial states:
FALSE     l = 6317745.8917
FALSE
FALSE   sigma:  0.097
FALSE
FALSE      AIC     AICc      BIC
FALSE 6139.631 6139.758 6149.403
FALSE
FALSE Error measures:
```

```
FALSE                     ME     RMSE      MAE         MPE     MAPE      MASE
FALSE Training set 56926.03 633571.7 460713.4 -0.03288687 6.945185 0.7436052
FALSE                   ACF1
FALSE Training set 0.2570241
FALSE
FALSE Forecasts:
FALSE          Point Forecast   Lo 80    Hi 80   Lo 95    Hi 95
FALSE Jan 2014        8992609 8049591  9935628 7550387 10434831
FALSE Feb 2014        7908116 6958724  8857508 6456146  9360086
FALSE Mar 2014        7079434 6123709  8035158 5617779  8541088
FALSE Apr 2014        6435209 5473193  7397225 4963933  7906486
FALSE May 2014        6161593 5193326  7129860 4680756  7642430
FALSE Jun 2014        7728705 6754226  8703185 6238368  9219043
FALSE Jul 2014        8837980 7857327  9818633 7338201 10337759
FALSE Aug 2014        9376841 8390053 10363630 7867678 10886004
FALSE Sep 2014        8601001 7608114  9593888 7082511 10119490
FALSE Oct 2014        6670419 5671470  7669368 5142658  8198180
FALSE Nov 2014        6035845 5030870  7040821 4498868  7572822
FALSE Dec 2014        7189087 6178120  8200053 5642947  8735226
```

STL - MAdN:

```
FALSE
FALSE Forecast method: STL +  ETS(M,Ad,N)
FALSE
FALSE Model Information:
FALSE ETS(M,Ad,N)
FALSE
FALSE Call:
FALSE  ets(y = x, model = etsmodel, damped = TRUE, allow.multiplicative.trend = allow.multiplicative.tr
FALSE
FALSE   Smoothing parameters:
FALSE     alpha = 0.1233
FALSE     beta  = 1e-04
FALSE     phi   = 0.8
FALSE
FALSE   Initial states:
FALSE     l = 5615471.7851
FALSE     b = 173606.4508
FALSE
FALSE   sigma:  0.0972
FALSE
FALSE      AIC     AICc      BIC
FALSE 6143.452 6143.906 6162.997
FALSE
FALSE Error measures:
FALSE                     ME     RMSE      MAE         MPE     MAPE      MASE
FALSE Training set 54337.68 631081.9 458777.5 -0.07364717 6.937249 0.7404807
FALSE                   ACF1
FALSE Training set 0.2528558
FALSE
FALSE Forecasts:
FALSE          Point Forecast   Lo 80    Hi 80   Lo 95    Hi 95
FALSE Jan 2014        9007707 8060947  9954467 7559763 10455651
```

```
FALSE Feb 2014        7923348 6969325  8877372 6464295  9382401
FALSE Mar 2014        7094774 6133536  8056011 5624687  8564860
FALSE Apr 2014        6450635 5482232  7419038 4969591  7931680
FALSE May 2014        6177088 5201569  7152607 4685160  7669016
FALSE Jun 2014        7744256 6761668  8726843 6241518  9246993
FALSE Jul 2014        8853574 7863967  9843182 7340100 10367048
FALSE Aug 2014        9392471 8395890 10389052 7868332 10916609
FALSE Sep 2014        8616658 7613151  9620166 7081926 10151391
FALSE Oct 2014        6686100 5675711  7696488 5140843  8231356
FALSE Nov 2014        6051544 5034319  7068769 4495832  7607255
FALSE Dec 2014        7204799 6180782  8228817 5638700  8770899
```

ets - MNM:

```
FALSE
FALSE Forecast method: ETS(M,N,M)
FALSE
FALSE Model Information:
FALSE ETS(M,N,M)
FALSE
FALSE Call:
FALSE  ets(y = ts_data_o)
FALSE
FALSE   Smoothing parameters:
FALSE     alpha = 0.1428
FALSE     gamma = 0.2119
FALSE
FALSE   Initial states:
FALSE     l = 6189149.8743
FALSE     s = 0.8984 0.7596 0.938 1.2229 1.2597 1.2396
FALSE            1.0059 0.7638 0.8078 0.8864 1.0269 1.191
FALSE
FALSE   sigma:  0.0967
FALSE
FALSE      AIC     AICc      BIC
FALSE 6144.033 6146.760 6192.895
FALSE
FALSE Error measures:
FALSE                     ME     RMSE      MAE         MPE     MAPE      MASE
FALSE Training set 45241.77 628252.5 481520.9 -0.04000239 7.277118 0.7771892
FALSE                   ACF1
FALSE Training set 0.1927438
FALSE
FALSE Forecasts:
FALSE          Point Forecast   Lo 80    Hi 80   Lo 95    Hi 95
FALSE Jan 2014        9917654 8689211 11146096 8038913 11796394
FALSE Feb 2014        8522973 7456477  9589469 6891908 10154038
FALSE Mar 2014        7012478 6126191  7898765 5657019  8367937
FALSE Apr 2014        6208601 5416196  7001006 4996722  7420480
FALSE May 2014        5928833 5164834  6692832 4760398  7097269
FALSE Jun 2014        7840532 6820624  8860440 6280717  9400347
FALSE Jul 2014        9115823 7919004 10312642 7285446 10946200
FALSE Aug 2014        9648549 8370229 10926869 7693527 11603571
FALSE Sep 2014        8553364 7409986  9696742 6804718 10302010
```

```
FALSE Oct 2014         6266745 5421655  7111835 4974291  7559199
FALSE Nov 2014         5938289 5130560  6746017 4702975  7173603
FALSE Dec 2014         8020901 6920610  9121192 6338151  9703651
```

4.1.2 R Script

```r
library(readxl)
library(tidyverse)
library(forecast)
library(imputeTS)
library(tsoutliers)

# load data
power_data <- read_excel("data/ResidentialCustomerForecastLoad-624.xlsx")

# Time Series
ts_data <- ts(power_data$KWH, frequency = 12, start = c(1998,1))

# Missing value imputation
ts_data <- na_interpolation(ts_data)

# STL decomposition
stl1 <- stl(ts_data, s.window = 'periodic')

# Handling outlier
outlier_func <- tsoutliers(ts_data, iterate = 2, lambda = "auto")

# Time Series - After outlier and imputation handeled
ts_data_o <- ts_data  # Let's treate outlier handled data seperatly for Modelling part.
ts_data_o[outlier_func$index] <- outlier_func$replacements

# Model#1: ARIMA
arima_auto <- auto.arima(ts_data_o)
arima_fc <- forecast(arima_auto, h=12)

# Model #2: STL (no-demped) - MNN
stl_ndemp <- stlf(ts_data_o, s.window = "periodic", robust=TRUE, h = 12)

# Model #2-2: STL (demped) - MAdN
stl_demp <- stlf(ts_data_o, damped=TRUE, s.window = "periodic", robust=TRUE, h = 12)

# Model #3: ets - MNM
ets_auto <- ets(ts_data_o)
ets_model <- forecast(ets_auto, h=12)

# tsCv - ARIMA -> it takes so much time. I got the results and saved them
##arima_cv <- function(x, h){forecast(Arima(x, order = c(3, 0, 2), seasonal = c(2, 1, 0), include.drift
##e <- tsCV(ts_data_o, arima_cv, h=12)

# RMSEs -> tsCV takes lot of time to process so just saved the output
#rmse_train_arima <- arima_auto[2]
#rmse_test_arima <- sqrt(mean(e^2, na.rm=TRUE))

rmse_train_arima <- 589381.7
rmse_test_arima <- 725175

# Save output
```

```r
write.csv(arima_fc, file="forecasts/POWER_ARIMA_FC.csv")
```