DATA 624: Project 1 - Part B

Sang Yoon (Andy) Hwang

October 22, 2019

Contents

1 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.

1.1 Data Exploration

Explore data.

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

2 Data preprocessing

Transformed data into time-series with freq - 12.

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

3 EDA - mean imputation, seasonal plots, STL decomposition, Acf graphs, summary statistics

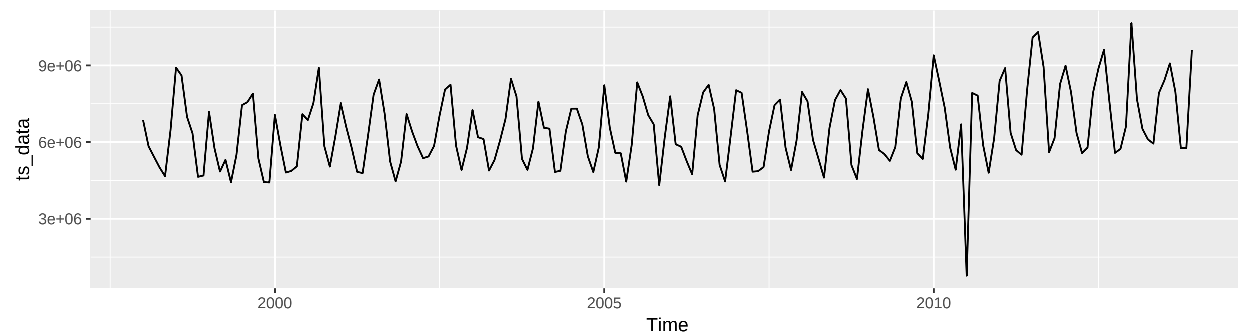
Box-Ljung test shows the series is not white noise (non-stationary with a weak positive trend and strong seasonality). 2008-Sep is missing and it was handled by mean imputation of all Septembers. On Jul 2010, we see that KWH suddenly drops dramatically (indeed outlier) - it could be due to input error but we are not so sure so we will keep it. During summer and winter time, we see the usage is usually higher. Seasonplot and ggAcf show that seasonality is pretty much consistent every year.

```r
# Missing data detected
ts_data
```

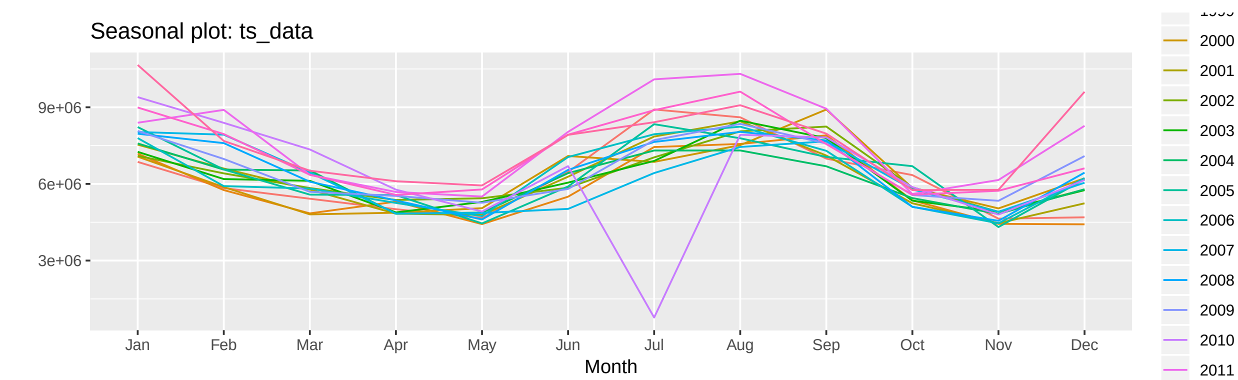
```
FALSE           Jan      Feb      Mar      Apr      May      Jun      Jul
FALSE 1998  6862583  5838198  5420658  5010364  4665377  6467147  8914755
FALSE 1999  7183759  5759262  4847656  5306592  4426794  5500901  7444416
FALSE 2000  7068296  5876083  4807961  4873080  5050891  7092865  6862662
FALSE 2001  7538529  6602448  5779180  4835210  4787904  6283324  7855129
FALSE 2002  7099063  6413429  5839514  5371604  5439166  5850383  7039702
FALSE 2003  7256079  6190517  6120626  4885643  5296096  6051571  6900676
FALSE 2004  7584596  6560742  6526586  4831688  4878262  6421614  7307931
FALSE 2005  8225477  6564338  5581725  5563071  4453983  5900212  8337998
FALSE 2006  7793358  5914945  5819734  5255988  4740588  7052275  7945564
FALSE 2007  8031295  7928337  6443170  4841979  4862847  5022647  6426220
FALSE 2008  7964293  7597060  6085644  5352359  4608528  6548439  7643987
FALSE 2009  8072330  6976800  5691452  5531616  5264439  5804433  7713260
FALSE 2010  9397357  8390677  7347915  5776131  4919289  6696292   770523
FALSE 2011  8394747  8898062  6356903  5685227  5506308  8037779 10093343
FALSE 2012  8991267  7952204  6356961  5569828  5783598  7926956  8886851
FALSE 2013 10655730  7681798  6517514  6105359  5940475  7920627  8415321
FALSE           Aug      Sep      Oct      Nov      Dec
FALSE 1998  8607428  6989888  6345620  4640410  4693479
FALSE 1999  7564391  7899368  5358314  4436269  4419229
FALSE 2000  7517830  8912169  5844352  5041769  6220334
FALSE 2001  8450717  7112069  5242535  4461979  5240995
FALSE 2002  8058748  8245227  5865014  4908979  5779958
FALSE 2003  8476499  7791791  5344613  4913707  5756193
FALSE 2004  7309774  6690366  5444948  4824940  5791208
FALSE 2005  7786659  7057213  6694523  4313019  6181548
FALSE 2006  8241110  7296355  5104799  4458429  6226214
FALSE 2007  7447146  7666970  5785964  4907057  6047292
FALSE 2008  8037137       NA  5101803  4555602  6442746
FALSE 2009  8350517  7583146  5566075  5339890  7089880
FALSE 2010  7922701  7819472  5875917  4800733  6152583
FALSE 2011 10308076  8943599  5603920  6154138  8273142
FALSE 2012  9612423  7559148  5576852  5731899  6609694
FALSE 2013  9080226  7968220  5759367  5769083  9606304
```

```r
# Mean imputation - with September
sept <- subset(power_data, grepl("Sep", power_data$`YYYY-MMM`))[3]
```

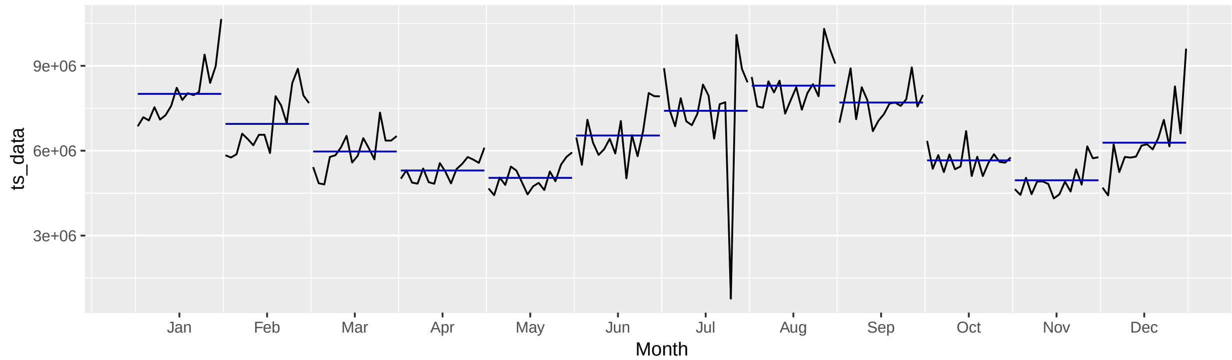
```r
sept_mean <- mean(sept$KWH, na.rm=TRUE)

# Apply mean to missing row
power_data$KWH[is.na(power_data$KWH) == TRUE]  <- sept_mean

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

# general series plot
autoplot(ts_data)
```



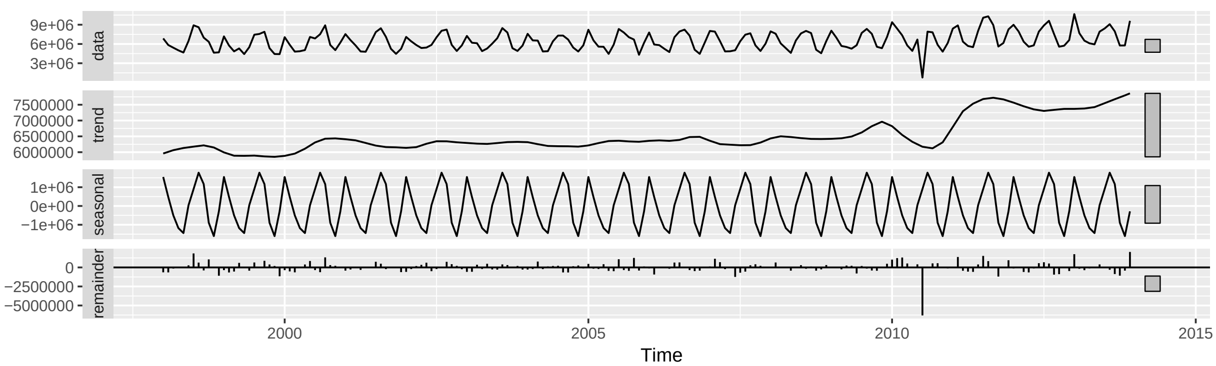

```r
# seasonal plot
ggseasonplot(ts_data)
```



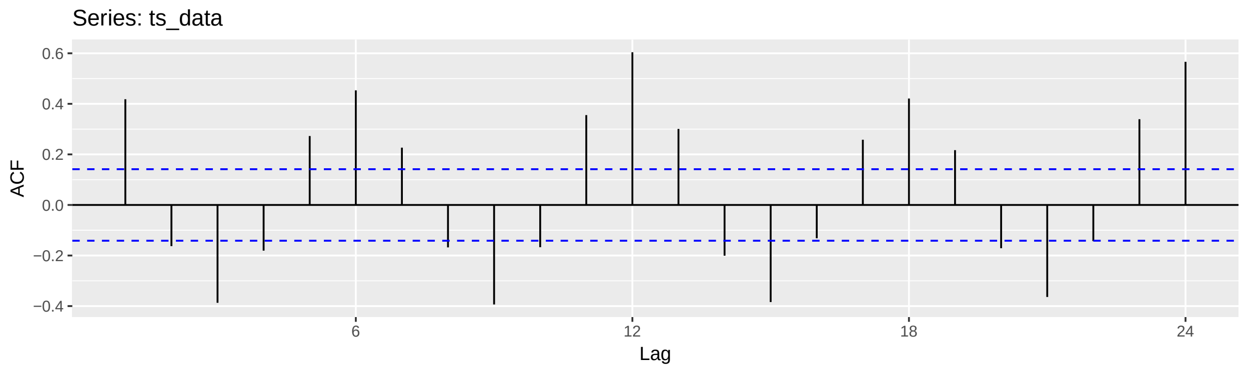

```r
# sub-seasonal plot
ggsubseriesplot(ts_data)
```




```
# STL decomposition
stl(ts_data, s.window = 'periodic') %>% autoplot()
```




```
# Autocorrelation
ggAcf(ts_data)
```




```
Box.test(ts_data, type = c("Ljung-Box"))
```

```
FALSE
FALSE   Box-Ljung test
FALSE
FALSE data:  ts_data
FALSE X-squared = 34.118, df = 1, p-value = 5.187e-09
```

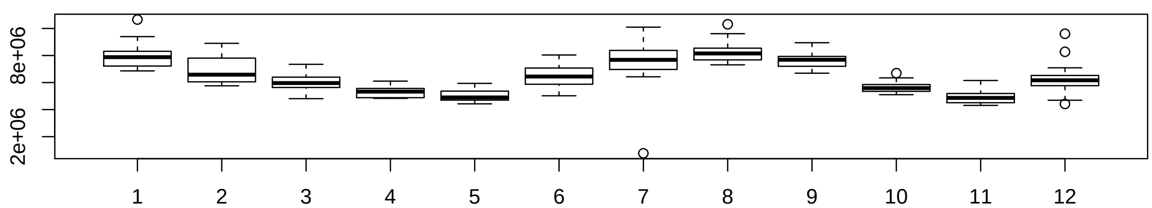
```r
# summary statistics
summary(ts_data)
```

```
FALSE     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
FALSE   770523  5434539  6314472  6508724  7649733 10655730
```

```r
summary(power_data)
```

```
FALSE   CaseSequence     YYYY-MMM              KWH          
FALSE  Min.   :733.0   Length:192         Min.   :  770523  
FALSE  1st Qu.:780.8   Class :character   1st Qu.: 5434539  
FALSE  Median :828.5   Mode  :character   Median : 6314472  
FALSE  Mean   :828.5                      Mean   : 6508724  
FALSE  3rd Qu.:876.2                      3rd Qu.: 7649733  
FALSE  Max.   :924.0                      Max.   :10655730
```

```r
# Boxplot
boxplot(ts_data~cycle(ts_data))
```

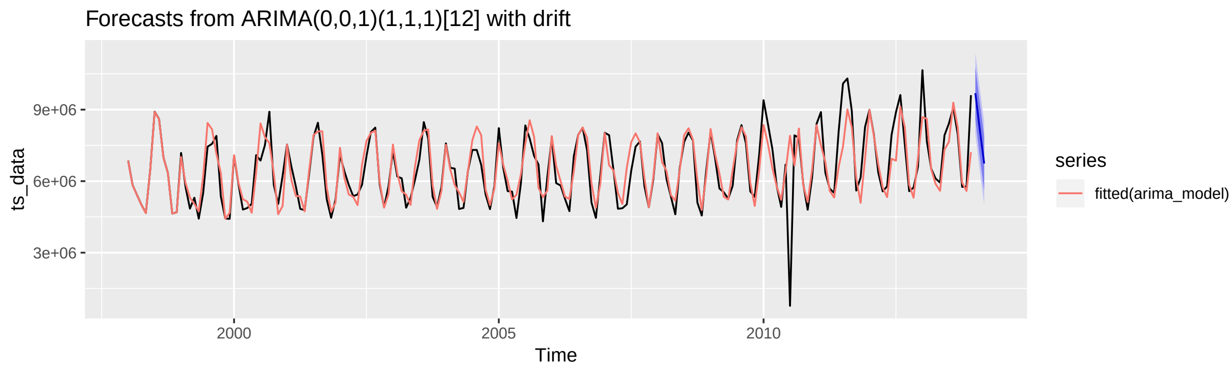


3.1 Data Model

From residual test (Box-Ljung), we found that ets - MNM is not reliable predictor as residuals are not white noise. Other models are all valid as residuals are all white noise (p > 0.05 from checkresiduals()). We will compare Arima and ets - AAN and ets - AAdN from stl decomposition in terms of RMSE on test set in the next section.

3.1.1 Model #1: ARIMA

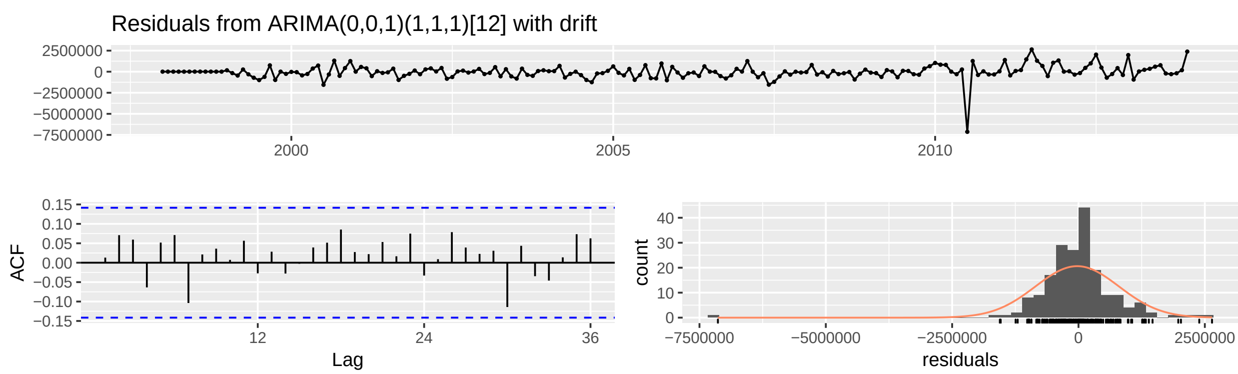
```r
# auto.arima
arima_model <- auto.arima(ts_data)

# forecast values
arima_model <- forecast(arima_model, h=3)

# forecast plot
autoplot(arima_model) + autolayer(fitted(arima_model))
```




```r
accuracy(arima_model)
```

```
FALSE                          ME     RMSE      MAE       MPE     MAPE      MASE
FALSE Training set -25755.56 823918.8 489803.5 -5.518168 11.63252 0.7141674
FALSE                     ACF1
FALSE Training set 0.0130951
```

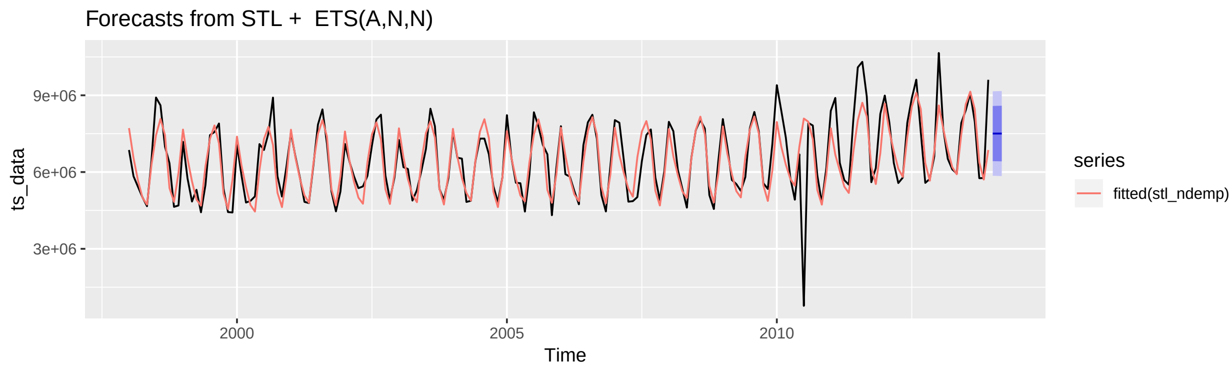
```r
checkresiduals(arima_model)
```



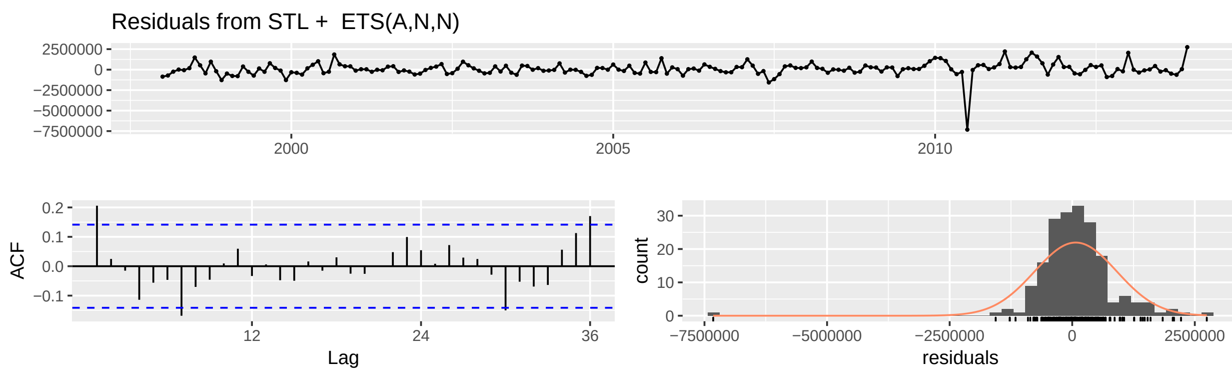

```
FALSE
FALSE   Ljung-Box test
FALSE
FALSE data:  Residuals from ARIMA(0,0,1)(1,1,1)[12] with drift
FALSE Q* = 12.619, df = 20, p-value = 0.8931
FALSE
FALSE Model df: 4.   Total lags used: 24
```

3.1.2 Model #2: STL (no-demped) - ANN

```r
#stlf - etsmodel estimation --- A,N,N is chosen.
stl_ndemp <- stlf(ts_data, damped=FALSE, s.window = "periodic", robust=TRUE, h = 3)

# forecast plot
autoplot(stl_ndemp) + autolayer(fitted(stl_ndemp))
```



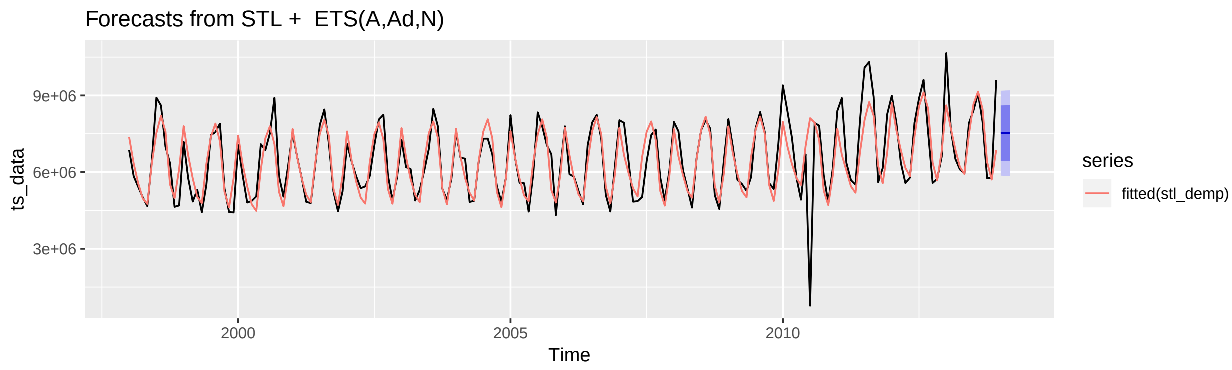

```r
checkresiduals(stl_ndemp)
```



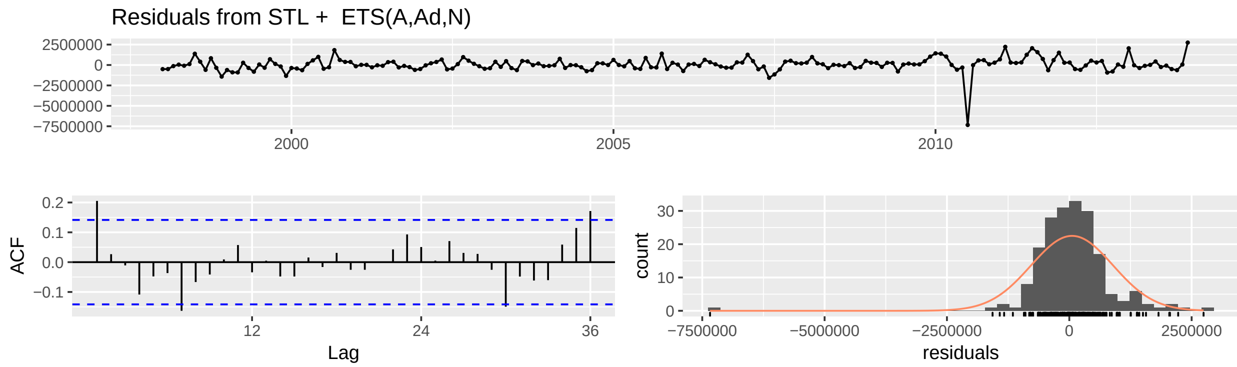

```
FALSE
FALSE   Ljung-Box test
FALSE
FALSE data:  Residuals from STL +  ETS(A,N,N)
FALSE Q* = 25.094, df = 22, p-value = 0.2926
FALSE
FALSE Model df: 2.   Total lags used: 24
```

3.1.3 Model #2-2: STL (demped) - AAdN

```r
#stlf - etsmodel estimation --- M, Ad, N is chosen.
stl_demp <- stlf(ts_data, damped=TRUE, s.window = "periodic", robust=TRUE, h = 3)

# forecast plot
autoplot(stl_demp) + autolayer(fitted(stl_demp))
```



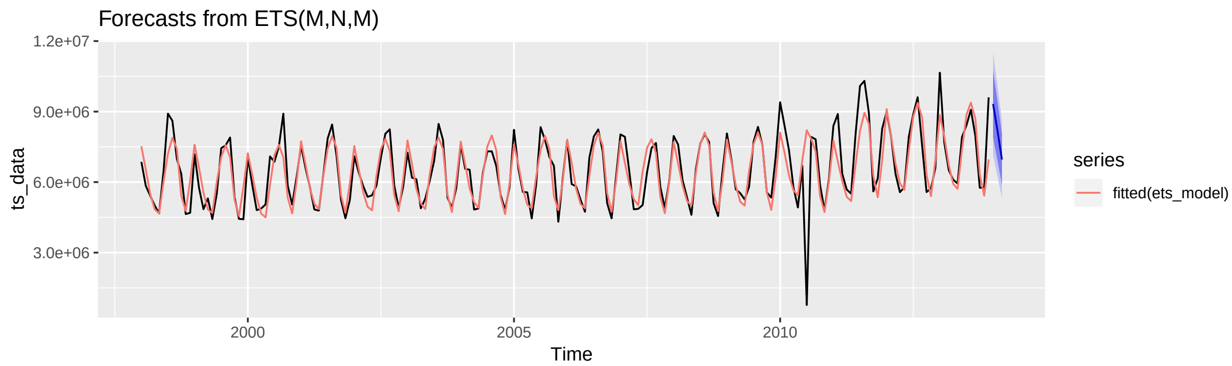

```
checkresiduals(stl_demp)
```



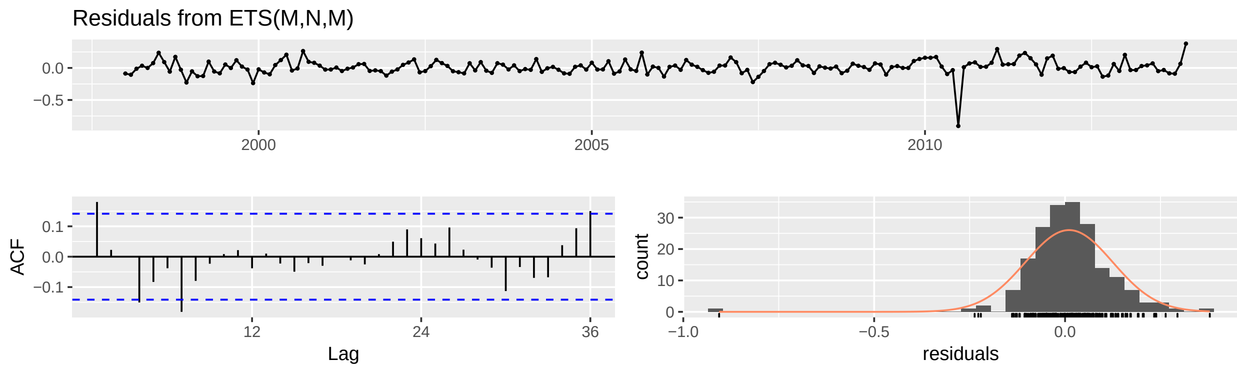

```
FALSE
FALSE   Ljung-Box test
FALSE
FALSE data:  Residuals from STL +  ETS(A,Ad,N)
FALSE Q* = 23.407, df = 19, p-value = 0.2199
FALSE
FALSE Model df: 5.   Total lags used: 24
```

3.1.4 Model #3: ets - MNM

```
# ETS models - MNM
ets_model <- ets(ts_data)

# forecast plot
autoplot(forecast(ets_model, h=3)) + autolayer(fitted(ets_model))
```




```
checkresiduals(ets_model)
```




```
FALSE
FALSE 	Ljung-Box test
FALSE
FALSE data:  Residuals from ETS(M,N,M)
FALSE Q* = 25.272, df = 10, p-value = 0.004853
FALSE
FALSE Model df: 14.   Total lags used: 24
```

3.2 Forecast accuracy

Using Time series cross-validation, we compute RMSE on testset (h=3). We will pick the model with the lowest RMSE on testset as our final model.

3.2.1 Model #1: ARIMA

```
arima_cv <- function(x, h){forecast(Arima(ts_data, order = c(0, 0, 1), seasonal = c(1, 1, 1),  include.
e <- tsCV(ts_data, arima_cv, h=3)

sqrt(mean(e^2, na.rm=TRUE))
```

```
FALSE [1] 2536394
```

3.2.2 Model #2: STL (no-demped) - ANN

```
e <- tsCV(ts_data, stlf, damped=FALSE, s.window = "periodic", robust=TRUE, h=3)
sqrt(mean(e^2, na.rm=TRUE))
```

```
FALSE [1] 1467209
```

3.2.3 Model #2-2: STL (demped) - AAdN

```
e <- tsCV(ts_data, stlf, damped=TRUE, s.window = "periodic", robust=TRUE, h=3)
sqrt(mean(e^2, na.rm=TRUE))
```

```
FALSE [1] 1473538
```

3.3 Discussion

From above, we found that ARIMA is the worst predictor and STL (demped) - AAdN is the best model as RMSE on testset is the lowest. We will pick Model #2-2.