TSA: Forecasting Competition

Yuxiang Ren and Karen Thornton

#packages

#Data ##temperature

```
#temperature
raw_temperature <- read_excel("Competition/Data/temperature.xlsx")
colnames(raw_temperature)

##transfer hourly to daily
temperature_day <- raw_temperature %>%
  group_by(date) %>%
  summarize(across(starts_with("t_ws"), mean, na.rm = TRUE))
##test result use first day of t_ws1
mean(raw_temperature$t_ws1[c(1:24)])
##save
write.csv(temperature_day, row.names = FALSE,
          file = "./Competition/Data/temperature_day.csv")
```

load

```
#load
raw_load <- read_excel("Competition/Data/load.xlsx")
colnames(raw_load)
## i guess h1 ~h24 mean hours,
## transfer hourly to daily
load_day_long <- raw_load %>%
  pivot_longer(cols = starts_with("h"),
               names_to = "hour",
               values_to = "energy")
load_day <- load_day_long %>%
  group_by(date) %>%
  summarize(energy = mean(energy))# use mean

##test
sum(load_day_long$energy[1:24])

##save
write.csv(load_day, row.names = FALSE,
          file = "./Competition/Data/load_day.csv")
```

##humidity

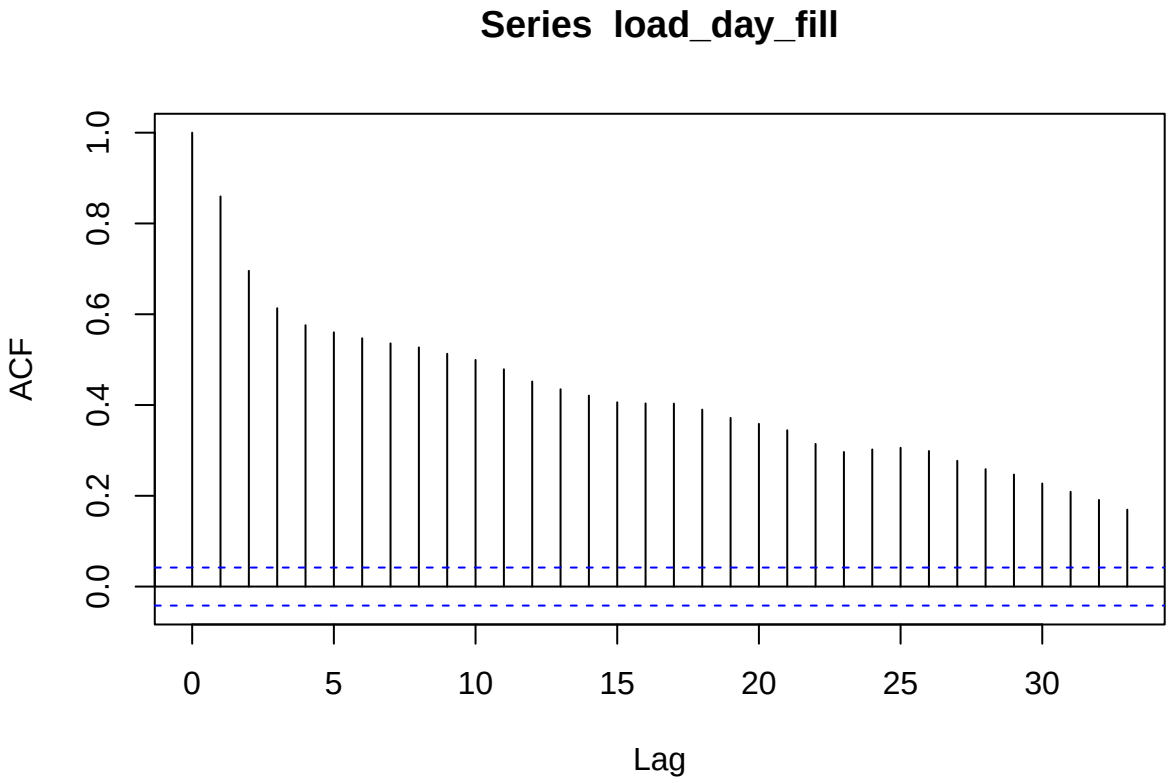
```r
#humidity
raw_humidity <- read_excel("Competition/Data/relative_humidity.xlsx")
colnames(raw_humidity)
##transfer hourly to daily
humidity_day <- raw_humidity %>%
  group_by(date) %>%
  summarize(across(starts_with("rh_ws"), mean, na.rm = TRUE))
##test result use first day of t_ws1
mean(raw_humidity$rh_ws1[c(1:24)])
##save
write.csv(humidity_day, row.names = FALSE,
          file = "./Competition/Data/humidity_day.csv")
```

##final data **to save space i save the result data, and use eval = FALSE in chunk obove.

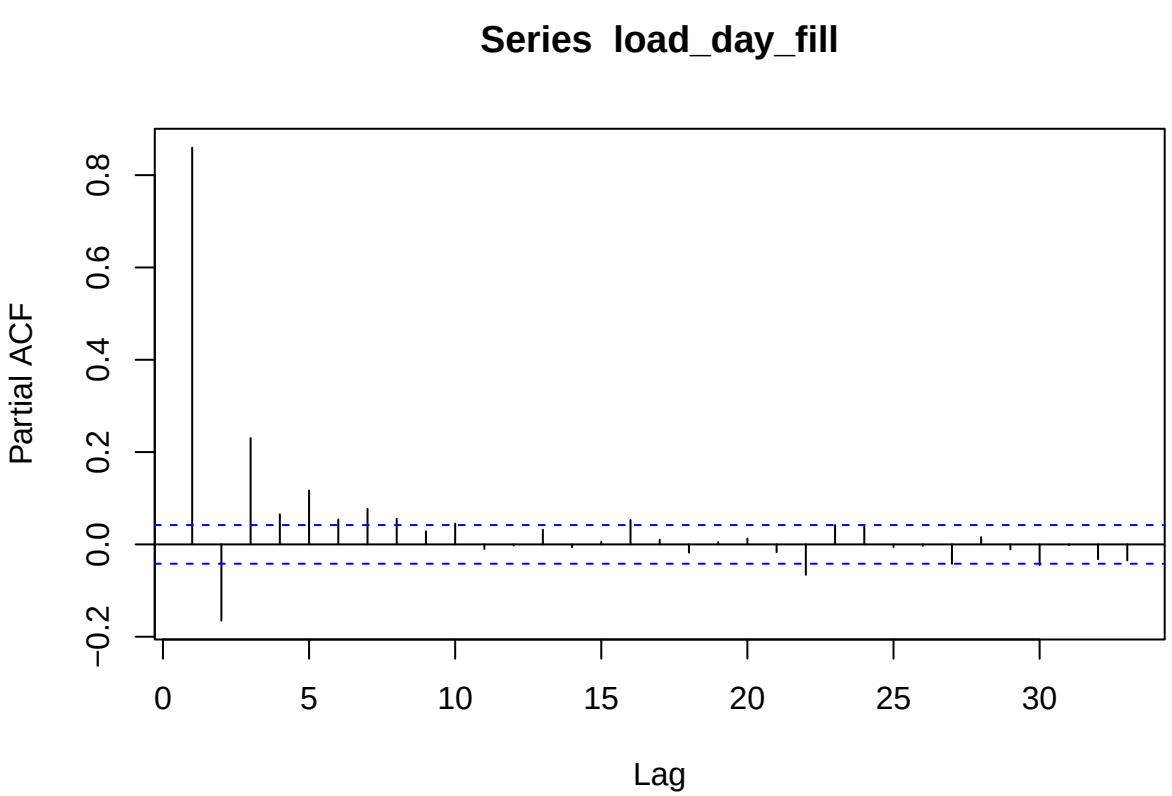
```r
load_day <- read.csv("./Competition/Data/load_day.csv")
temperature_day <- read.csv("./Competition/Data/temperature_day.csv")
humidity_day <- read.csv("./Competition/Data/humidity_day.csv")
```

#create a time series

```r
ts_load_daily <- msts(load_day$energy,
                          seasonal.periods =c(7,365.25),
                          start=c(2005,1,1))

load_day_fill <- na.approx(load_day$energy)
ts_load_daily_fill <- msts(load_day_fill,
                          seasonal.periods =c(7,365.25),
                          start=c(2005,1,1))
acf(load_day_fill)
```




```
pacf(load_day_fill)
```



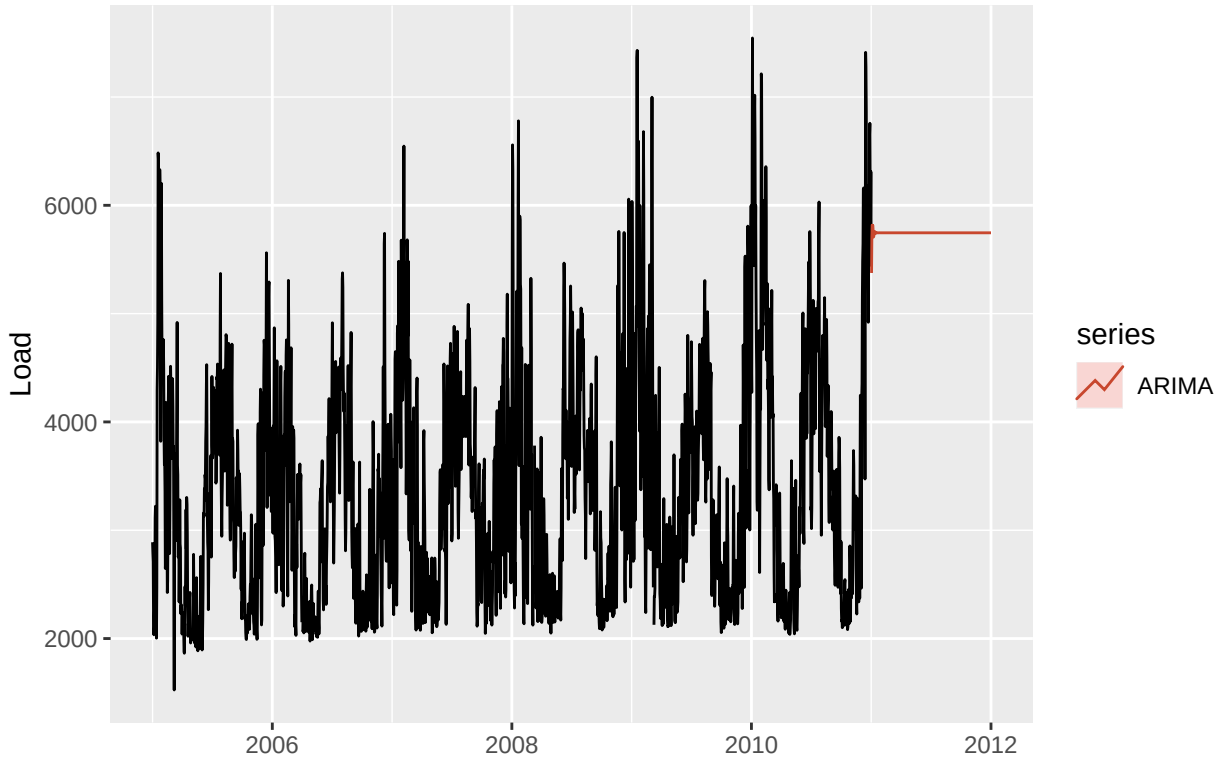
```r
#ts_temperature
##mean temperature
temp <- temperature_day %>%
  group_by(date) %>%
  summarise(T = mean(c_across(starts_with("t_ws")), na.rm = TRUE))

ts_temp <- msts(temp,seasonal.periods =c(7,365.25),
                           start=c(2005,1,1))
#for result generate
data_seq_YR <- seq(from = as.Date("2011-01-01"), to = as.Date("2011-02-28"), by = "day")
```

##create an arima

```r
#arima with only one factor
Simple_ARIMA <- auto.arima(ts_load_daily, seasonal=FALSE)
print(Simple_ARIMA)
```

```
## Series: ts_load_daily
## ARIMA(5,1,0)
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ar5
##       0.0350  -0.3419  -0.1377  -0.1551  -0.0895
## s.e.  0.0214   0.0211   0.0221   0.0211   0.0213
##
## sigma^2 = 270055:  log likelihood = -16756.07
## AIC=33524.15   AICc=33524.19   BIC=33558.3
```

```r
#simple arima forecast
Simple_ARIMA_for <- forecast(object = Simple_ARIMA, h=365)

#plot
autoplot(ts_load_daily) +
    autolayer(Simple_ARIMA_for,series="ARIMA",PI=FALSE) +
    ylab("Load") +
    xlab("")
```



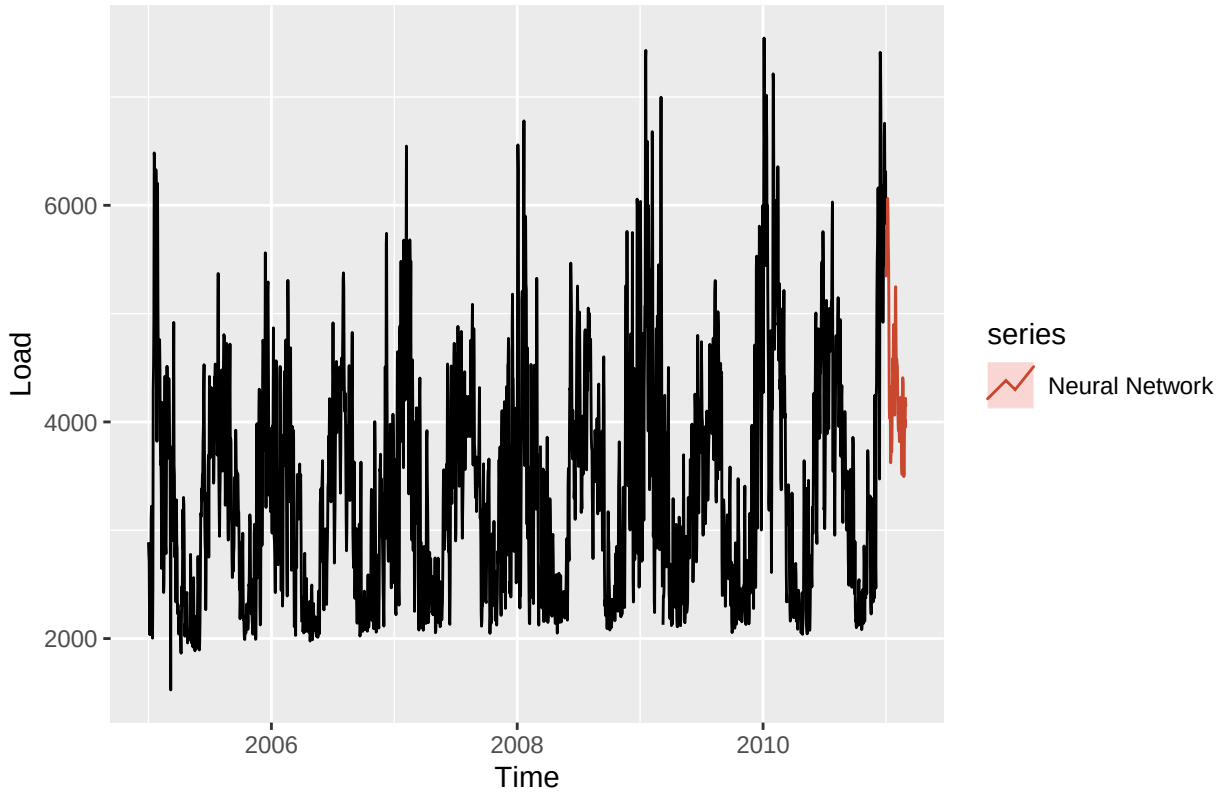

```
#save
write.csv(Simple_ARIMA_for, row.names = FALSE,
          file = "./Competition/Data/Simple.ARIMA.Forecast.mean.csv")
```

##Neural Network

```
NN_fit <- nnetar(ts_load_daily,p=1,P=0,xreg=fourier(ts_load_daily, K=c(2,12)))
```

```
## Warning in nnetar(ts_load_daily, p = 1, P = 0, xreg = fourier(ts_load_daily, :
## Missing values in x, omitting rows
```

```
NN_for <- forecast(NN_fit, h=59,xreg=fourier(ts_load_daily,
                                             K=c(2,12),h=59))

#plot
autoplot(ts_load_daily) +
  autolayer(NN_for, series="Neural Network",PI=FALSE)+
  ylab("Load")
```



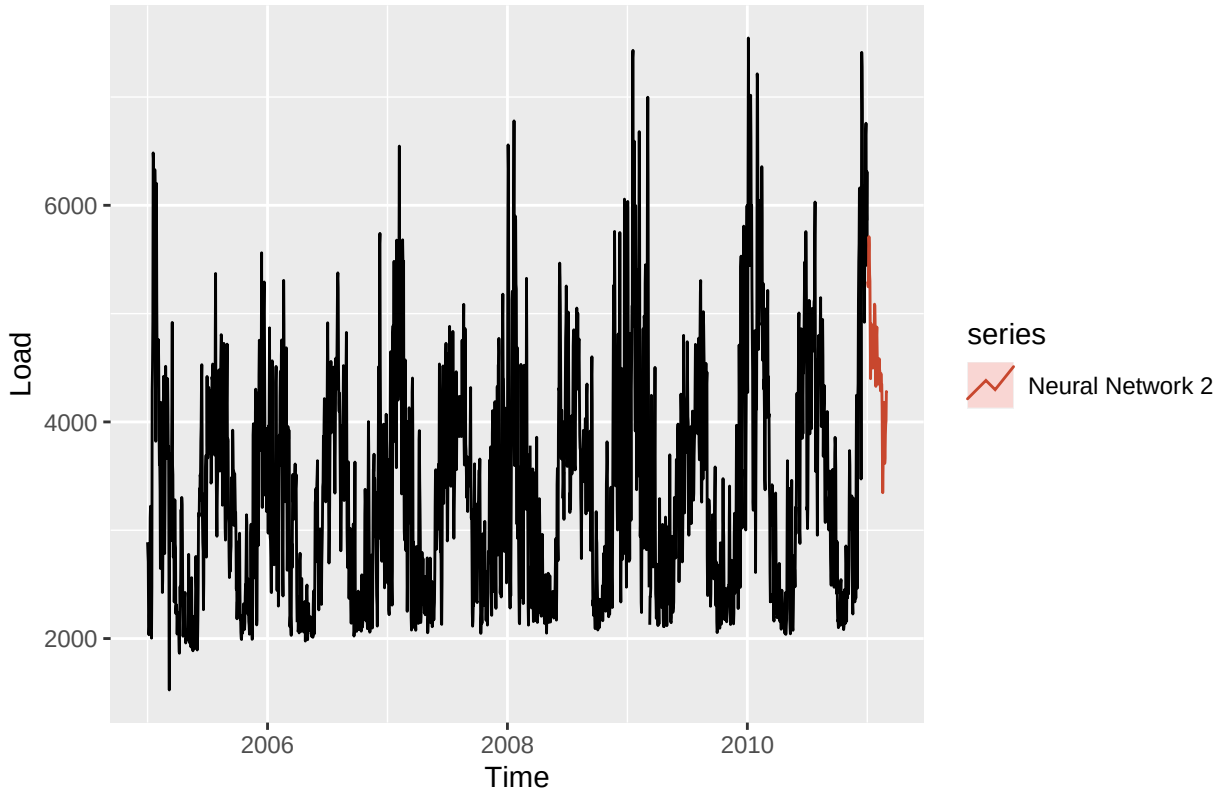

```
#save
write.csv(NN_for, row.names = FALSE,
          file = "./Competition/Data/Neural.Network.Forecast.csv")
```

##Neural Network 2

```
NN_fit_2 <- nnetar(ts_load_daily,p=1,P=0,xreg=fourier(ts_load_daily, K=c(2,7)))
```

```
## Warning in nnetar(ts_load_daily, p = 1, P = 0, xreg = fourier(ts_load_daily, :
## Missing values in x, omitting rows
```

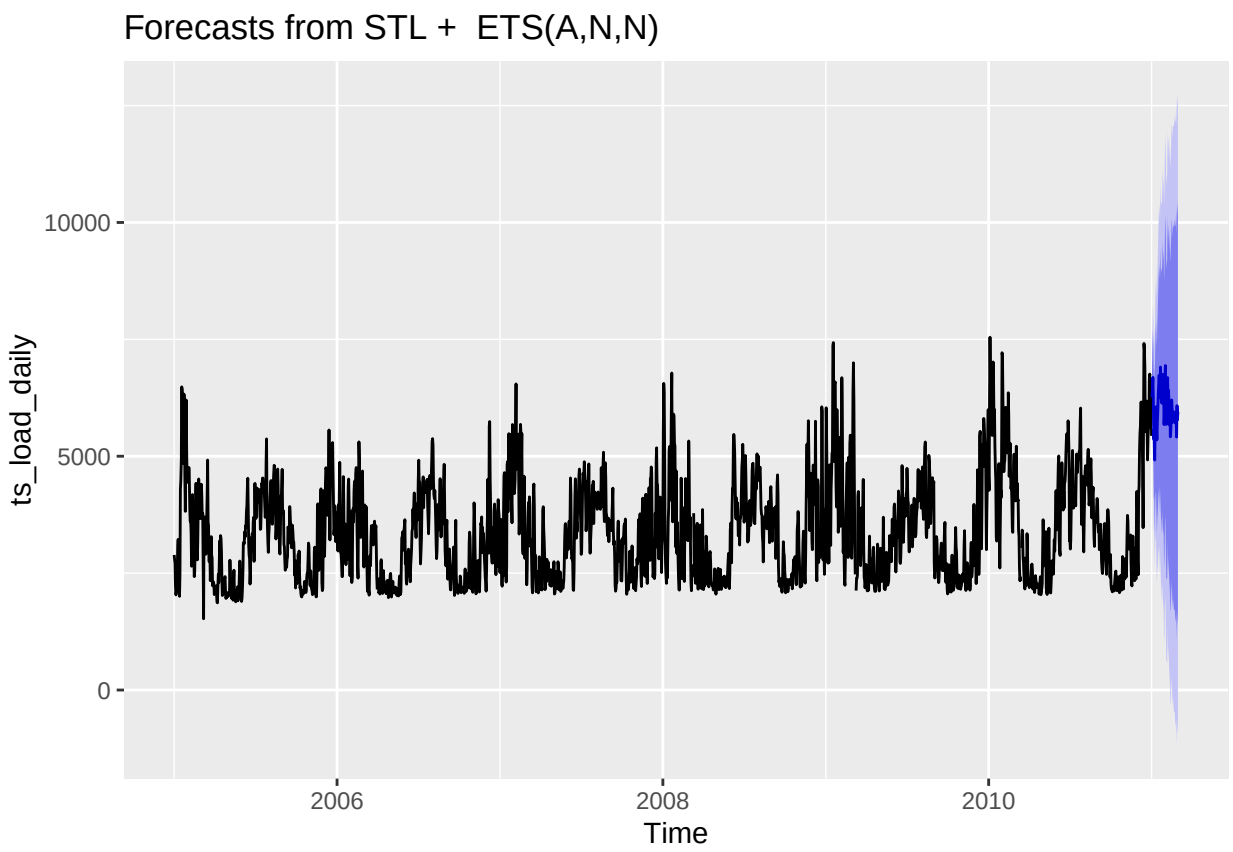
```
NN_for_2 <- forecast(NN_fit_2, h=59,xreg=fourier(ts_load_daily,
                                          K=c(2,7),h=59))

#plot
autoplot(ts_load_daily) +
  autolayer(NN_for_2, series="Neural Network 2",PI=FALSE)+
  ylab("Load")
```




```
#save
write.csv(NN_for_2, row.names = FALSE,
          file = "./Competition/Data/Neural.Network.Forecast.2.csv")
```

##STL + ETS

```
ETS_fit_YR <-  stlf(ts_load_daily,h=59)
autoplot(ETS_fit_YR)
```




```
STL_ETS_YR <- ETS_fit_YR$mean

#save

data_seq_YR <- seq(from = as.Date("2011-01-01"), to = as.Date("2011-02-28"), by = "day")
STL_ETS_YR_df <- data.frame(date = data_seq_YR, load = STL_ETS_YR)

write.csv(STL_ETS_YR_df, row.names = FALSE,
          file = "./Competition/Data/STL.ETS.Forecast.csv")
```

##ARIMA + FOURIER terms loop take long time, the best K values is (2,2)

```
min_AICc <- Inf
best_K <- c(0, 0)

for (k1 in 1:2) {
  for (k2 in 1:20) { # Change the range of k2 to a smaller value for testing
    ARIMA_Four_fit_temp <- auto.arima(ts_load_daily,
                                      seasonal=FALSE,
                                      lambda=0,
                                      xreg=fourier(ts_load_daily,
                                                   K=c(k1, k2))
                                      )
    current_AICc <- ARIMA_Four_fit_temp$aicc

    cat("k1: ", k1, " k2: ", k2, " AICc: ", current_AICc, "\n")
```

```
    if (current_AICc < min_AICc) {
      min_AICc <- current_AICc
      best_K <- c(k1, k2)
    }
  }
}

print(paste0("Best K values: (", best_K[1], ", ", best_K[2], ")"))
print(paste0("Minimum AICc: ", min_AICc))

ARIMA_Four_fit_YR <- auto.arima(ts_load_daily,
                            seasonal=FALSE,
                            lambda=0,
                            xreg=fourier(ts_load_daily,
                                         K=best_K)
                            )

ARIMA_Four_for_YR <- forecast(ARIMA_Four_fit_YR,
                          xreg=fourier(ts_load_daily,
                                       K=best_K,
                                       h=59),
                          h=59
                          )

#save
ARIMA_Four_YR <- ARIMA_Four_for_YR$mean
data_seq_YR <- seq(from = as.Date("2011-01-01"), to = as.Date("2011-02-28"), by = "day")
ARIMA_Four_YR_df <- data.frame(date = data_seq_YR, load = ARIMA_Four_YR)

write.csv(ARIMA_Four_YR_df, row.names = FALSE,
          file = "./Competition/Data/ARIMA.FOUR.Forecast.csv")

#re-test k = 2, 12
ARIMA_Four_fit2_YR <- auto.arima(ts_load_daily,
                            seasonal=FALSE,
                            lambda=0,
                            xreg=fourier(ts_load_daily,
                                         K=c(2,12))
                            )

ARIMA_Four_for2_YR <- forecast(ARIMA_Four_fit2_YR,
                          xreg=fourier(ts_load_daily,
                                       K=c(2,12),
                                       h=59),
                          h=59
                          )

#save
ARIMA_Four2_YR <- ARIMA_Four_for2_YR$mean
data_seq_YR <- seq(from = as.Date("2011-01-01"), to = as.Date("2011-02-28"), by = "day")
ARIMA_Four2_YR_df <- data.frame(date = data_seq_YR, load = ARIMA_Four2_YR)

write.csv(ARIMA_Four2_YR_df, row.names = FALSE,
```

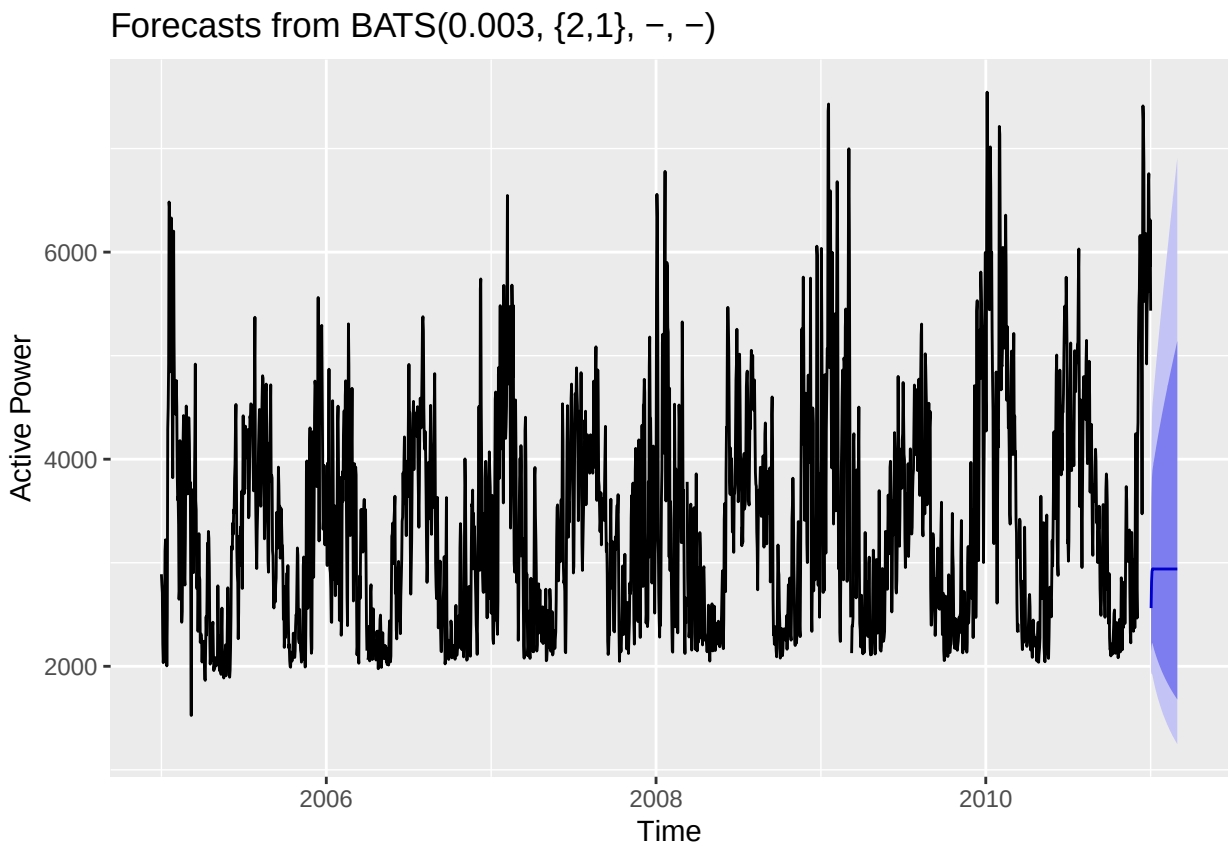
```
          file = "./Competition/Data/ARIMA.FOUR2.Forecast.csv")
```

##TBATS

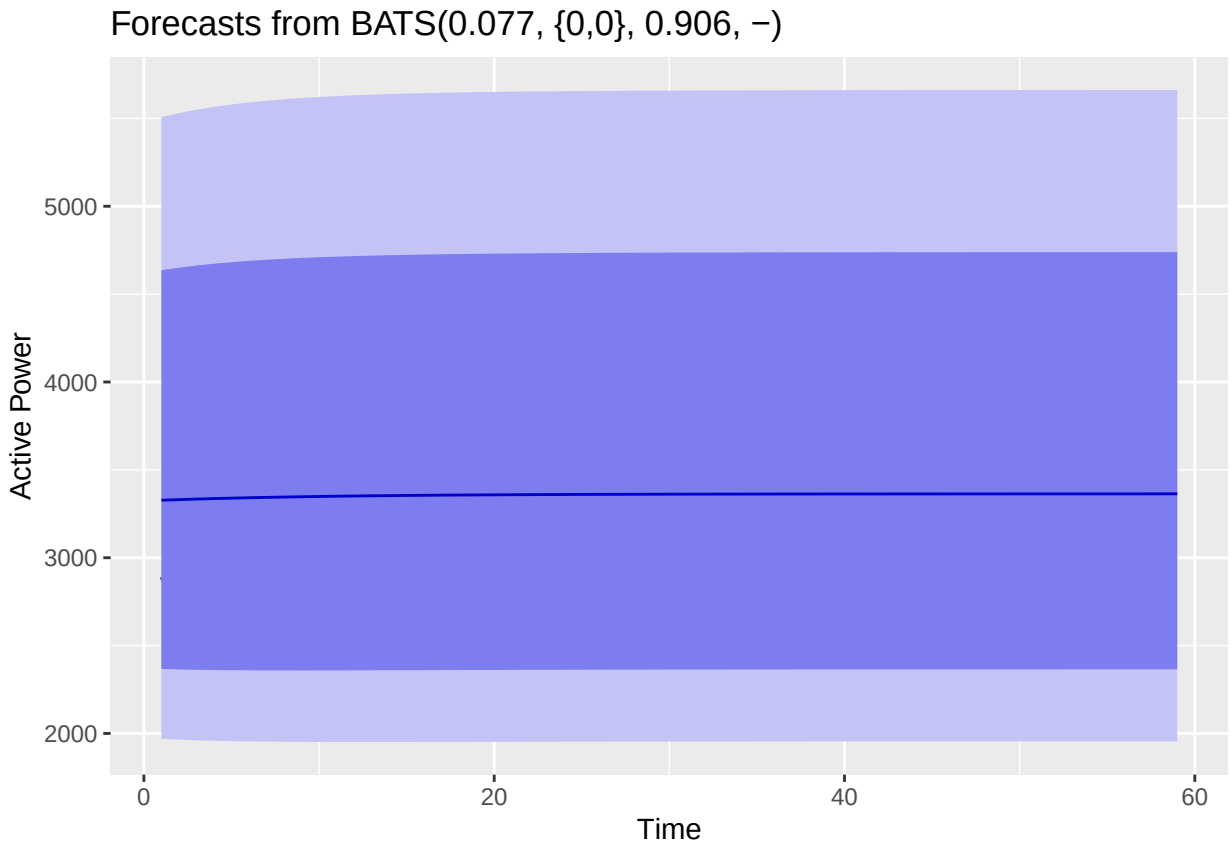
```
TBATS_fit_YR <- tbats(ts_load_daily)
```

```
## Warning in tbats(ts_load_daily): Missing values encountered. Using longest
## contiguous portion of time series
```

```
TBATS_for <- forecast(TBATS_fit_YR, h=59)
autoplot(TBATS_for) +
  ylab("Active Power") # too small, not report
```




```
#replace NA, might wrong
ts_full_YR <- approx(ts_load_daily, method = "linear")
TBATS_fit_YR <- tbats(ts_full_YR$y)
TBATS_for <- forecast(TBATS_fit_YR, h=59)
autoplot(TBATS_for) +
  ylab("Active Power")
```

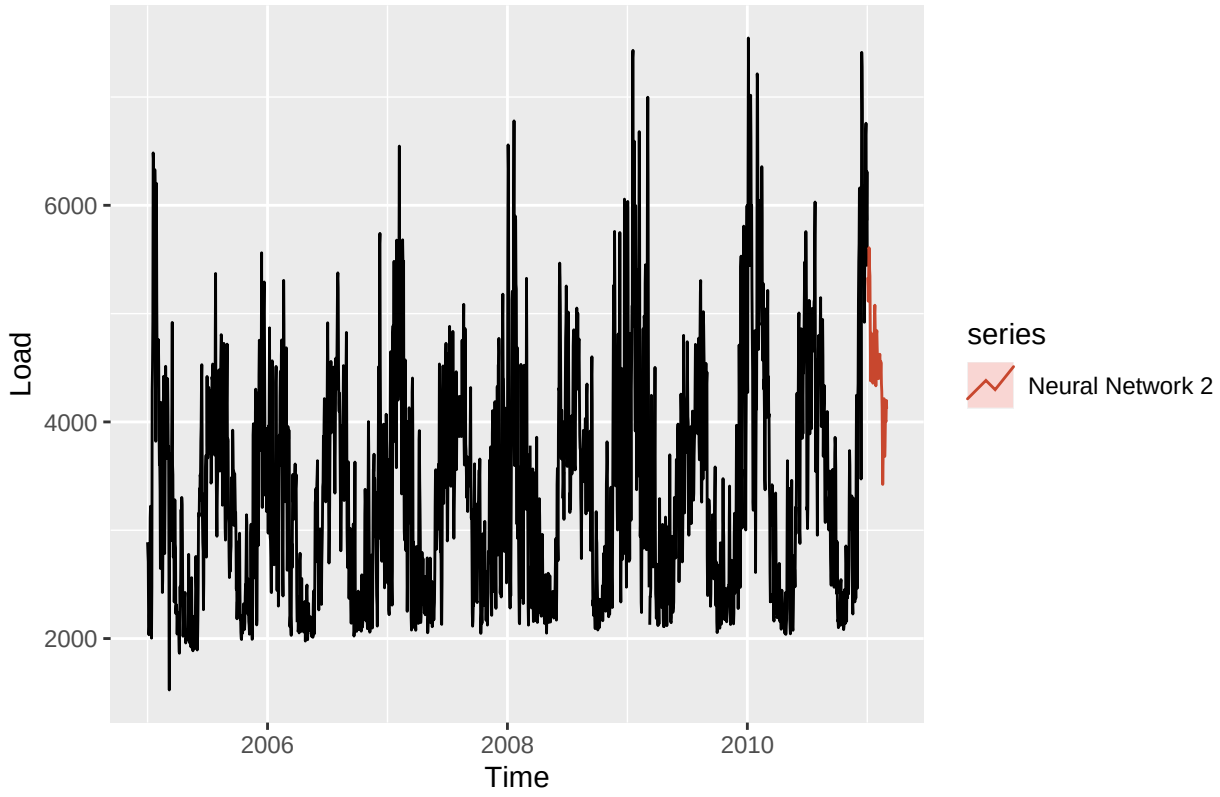


##Neural_Network same code as NN2, but set seed

```
set.seed(6787)
NN_fit_2_YR <- nnetar(ts_load_daily,p=1,P=0,xreg=fourier(ts_load_daily, K=c(2,7)))
```

```
## Warning in nnetar(ts_load_daily, p = 1, P = 0, xreg = fourier(ts_load_daily, :
## Missing values in x, omitting rows
```

```
NN_for_2_YR <- forecast(NN_fit_2_YR, h=59,xreg=fourier(ts_load_daily,
                                          K=c(2,7),h=59))

#plot
autoplot(ts_load_daily) +
  autolayer(NN_for_2_YR, series="Neural Network 2",PI=FALSE)+
  ylab("Load")
```



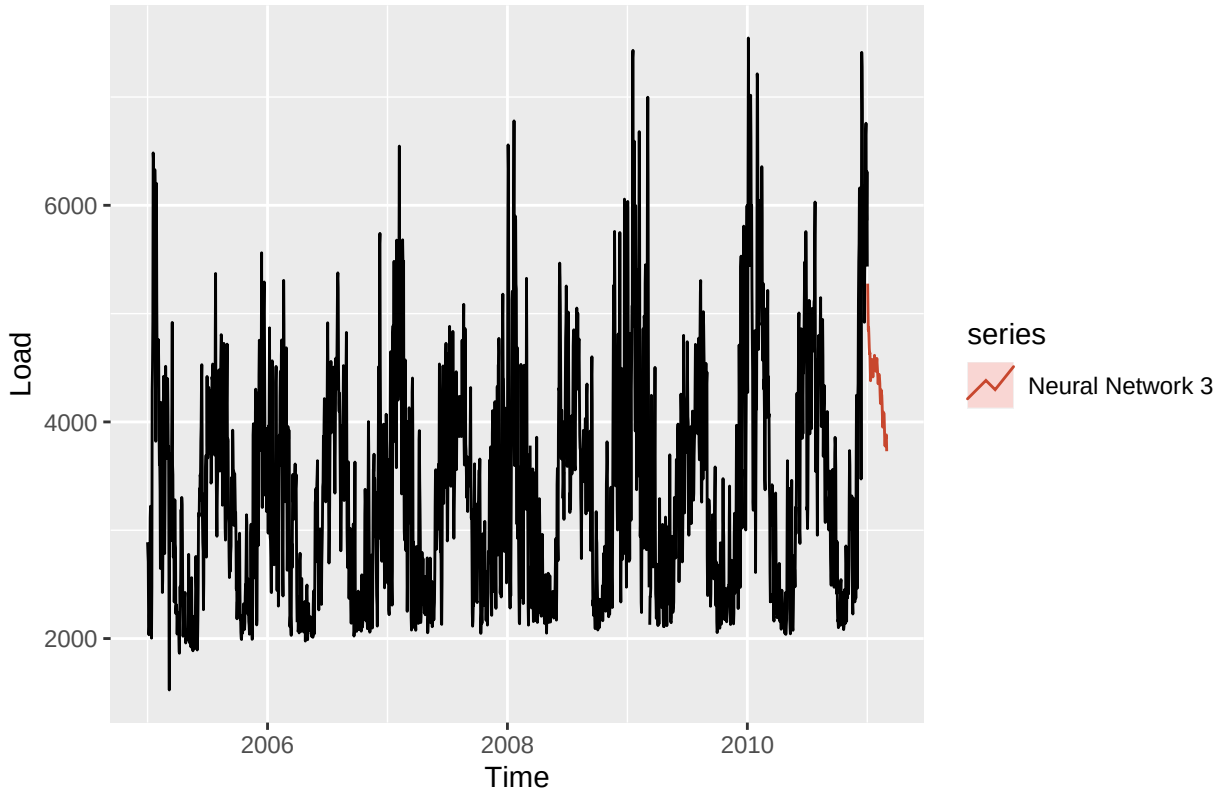

```
#save
Neural.Network.6787.YR_df <- data.frame(date = data_seq_YR, load = NN_for_2_YR$mean)

write.csv(Neural.Network.6787.YR_df, row.names = FALSE,
          file = "./Competition/Data/Neural.Network.Forecast.seed6787.2.csv")
```

##Neural_Network, k= 2,2 seed 6787 k 22

```
set.seed(6787)
NN_fit_3_YR <- nnetar(ts_load_daily,p=1,P=0,xreg=fourier(ts_load_daily, K=c(2,2)))
```

```
## Warning in nnetar(ts_load_daily, p = 1, P = 0, xreg = fourier(ts_load_daily, :
## Missing values in x, omitting rows
```

```
NN_for_3_YR <- forecast(NN_fit_3_YR, h=59,xreg=fourier(ts_load_daily,
                                          K=c(2,2),h=59))

#plot
autoplot(ts_load_daily) +
  autolayer(NN_for_3_YR, series="Neural Network 3",PI=FALSE)+
  ylab("Load")
```



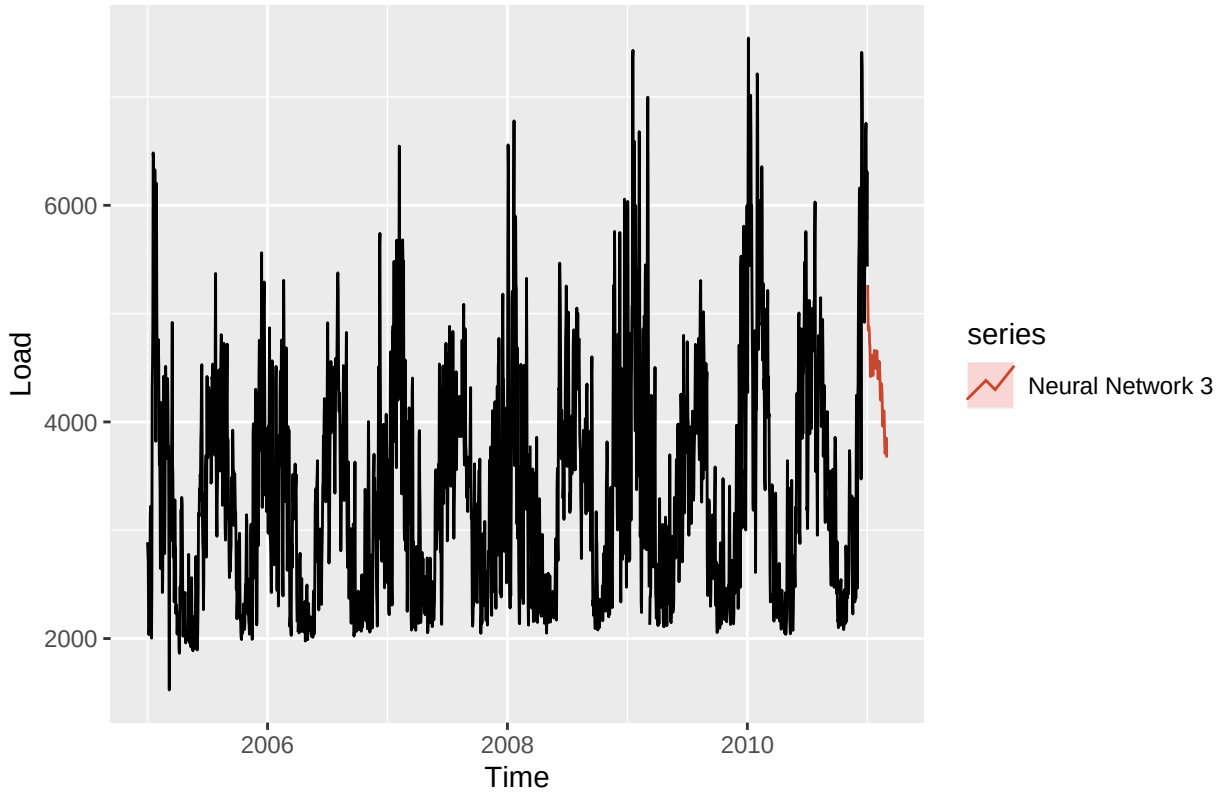

```
#save
Neural.Network.22.6787.YR_df <- data.frame(date = data_seq_YR, load = NN_for_3_YR$mean)

write.csv(Neural.Network.22.6787.YR_df, row.names = FALSE,
          file = "./Competition/Data/Neural.Network.Forecast.22.seed6787.csv")
```

##Neural_Network, k= 2,2 seed = 1234

```
set.seed(1234)
NN_fit_4_YR <- nnetar(ts_load_daily,p=1,P=0,xreg=fourier(ts_load_daily, K=c(2,2)))
```

```
## Warning in nnetar(ts_load_daily, p = 1, P = 0, xreg = fourier(ts_load_daily, :
## Missing values in x, omitting rows
```

```
NN_for_4_YR <- forecast(NN_fit_4_YR, h=59,xreg=fourier(ts_load_daily,
                                          K=c(2,2),h=59))

#plot
autoplot(ts_load_daily) +
  autolayer(NN_for_4_YR, series="Neural Network 3",PI=FALSE)+
  ylab("Load")
```




```
#save
Neural.Network.22.1234.YR_df <- data.frame(date = data_seq_YR, load = NN_for_4_YR$mean)

write.csv(Neural.Network.22.1234.YR_df, row.names = FALSE,
          file = "./Competition/Data/Neural.Network.Forecast.22.seed1234.csv")
```

##Neural_Network, find the best pPK. this code can not work

```
#replace NA
load_day_fill <- na.approx(load_day$energy)
ts_load_daily_fill <- msts(load_day_fill,
                           seasonal.periods =c(7,365.25),
                           start=c(2005,1,1))
ts_data_train <- window(ts_load_daily_fill, start=c(2005, 1), end=c(2009, 365))

ts_data_test <- window(ts_load_daily_fill, start=c(2010, 1), end=c(2010, 365))

min_MSE <- Inf
best_p <- 0
best_P <- 0
# Loop through different combinations of p, P, and K
for (p in 0:3) {
  for (P in 0:3) {
        # Fit the model with the current combination of parameters
       NN_fit <- nnetar(ts_data_train, p=p, P=P, xreg=fourier(ts_data_train, K=c(2,12)))
       NN_for_temp <- forecast(NN_fit, h = 365, xreg=fourier(ts_data_test, K = c(2,12), h = 365))

       current_MSE <- mean((ts_data_test - NN_for_temp$mean)^2)
```

```
        if (current_MSE < min_MSE) {
          min_MSE <- current_MSE
          best_p <- p
          best_P <- P
        }
      }
    }

cat("Best p:", best_p, "\n")
cat("Best P:", best_P, "\n")
cat("Minimum MSE:", min_MSE, "\n")
```

##Neural_Network, for non-seasonal i forget there should no k

```
## remove seansaol
stl_decomposition <- stl(ts_load_daily_fill, s.window = "periodic", robust = TRUE)
seasonal_component <- stl_decomposition$time.series[, "seasonal"]
deseasonalized_data <- ts_load_daily_fill - seasonal_component

set.seed(6787)
NN_fit_deseasonalized_212 <- nnetar(deseasonalized_data, p = 1, P = 0, K = c(2, 12))
NN_forecast_deseasonalized_212 <- forecast(NN_fit_deseasonalized_212, h = 59)

forecast_seasonal_component <- window(seasonal_component, start = c(2010, 1), end = c(2010, 59))


df <- data.frame(date = data_seq_YR, load1 = NN_forecast_deseasonalized_212$mean)
df2 <- data.frame(load2 = forecast_seasonal_component)

df <- data.frame(date = as.Date(data_seq_YR), load1 = coredata(df$load1))
df2 <- data.frame(load2 = coredata(df2$load2))

df3 <- cbind(df, df2)
df3$load <- df3$load1 + df3$load2
df3 <- df3[,c(1,4)]
write.csv(df3, row.names = FALSE,
          file = "./Competition/Data/NN.deseason.212.6768.csv")
```

##Neural_Network, for non-seasonal p= 2,P = 0

```
## remove seansaol
stl_decomposition <- stl(ts_load_daily_fill, s.window = "periodic", robust = TRUE)
seasonal_component <- stl_decomposition$time.series[, "seasonal"]
deseasonalized_data <- ts_load_daily_fill - seasonal_component

set.seed(6787)
NN_fit_deseasonalized_212 <- nnetar(deseasonalized_data, p = 2, P = 0)
NN_forecast_deseasonalized_212 <- forecast(NN_fit_deseasonalized_212, h = 59)

forecast_seasonal_component <- window(seasonal_component, start = c(2010, 1), end = c(2010, 59))
```

```
df <- data.frame(date = data_seq_YR, load1 = NN_forecast_deseasonalized_212$mean)
df2 <- data.frame(load2 = forecast_seasonal_component)
df3 <- cbind(df,df2)

df <- data.frame(date = as.Date(data_seq_YR), load1 = coredata(df$load1))
df2 <- data.frame(load2 = coredata(df2$load2))

df3 <- cbind(df, df2)
df3$load <- df3$load1 + df3$load2
df3 <- df3[,c(1,4)]
write.csv(df3, row.names = FALSE,
          file = "./Competition/Data/NN.deseason.p2.6768.csv")
```

##Arima+temp

```
set.seed(6787)
deseasonalized_data_temp <- cbind(deseasonalized_data, ts_temp)

arima_fit_deseasonalized_temp <- auto.arima(deseasonalized_data_temp[, 1],
                                            xreg = as.matrix(deseasonalized_data_temp[, 3]),
                                            seasonal = FALSE)
xreg <- deseasonalized_data_temp[(nrow(deseasonalized_data_temp)-58):nrow(deseasonalized_data_temp), 3]

forecast_deseasonalized_T <- forecast(arima_fit_deseasonalized_temp, xreg = xreg, h = 59)


df <- data.frame(date = data_seq_YR, load1 = forecast_deseasonalized_T$mean)
df2 <- data.frame(load2 = forecast_seasonal_component)
df3 <- cbind(df,df2)

df <- data.frame(date = as.Date(data_seq_YR), load1 = coredata(df$load1))
df2 <- data.frame(load2 = coredata(df2$load2))

df3 <- cbind(df, df2)
df3$load <- df3$load1 + df3$load2
df3 <- df3[,c(1,4)]
write.csv(df3, row.names = FALSE,
          file = "./Competition/Data/NN.deseason.Temp.6768.csv")
```

##Neural_Network, k= 2,2,p=2

```
set.seed(6787)
NN_fit_p2_YR <- nnetar(ts_load_daily,p=2,P=0,xreg=fourier(ts_load_daily, K=c(2,2)))


## Warning in nnetar(ts_load_daily, p = 2, P = 0, xreg = fourier(ts_load_daily, :
## Missing values in x, omitting rows

NN_forp2_YR <- forecast(NN_fit_p2_YR, h=59,xreg=fourier(ts_load_daily,
                                          K=c(2,2),h=59))


#save
```

```
Neural.Network.p2.22.6787.YR_df <- data.frame(date = data_seq_YR, load = NN_forp2_YR$mean)

write.csv(Neural.Network.p2.22.6787.YR_df, row.names = FALSE,
          file = "./Competition/Data/Neural.Network.Forecast.p2.22.seed6787.csv")
```

##Neural_Network, k= 2,2,p=1,P=1

```
set.seed(6787)
NN_fit_p1P1_YR <- nnetar(ts_load_daily,p=1,P=1,xreg=fourier(ts_load_daily, K=c(2,2)))
```

```
## Warning in nnetar(ts_load_daily, p = 1, P = 1, xreg = fourier(ts_load_daily, :
## Missing values in x, omitting rows
```

```
NN_for_p1P1_YR <- forecast(NN_fit_p1P1_YR, h=59,xreg=fourier(ts_load_daily,
                                          K=c(2,2),h=59))


#save
Neural.Network.p1P1.22.6787.YR_df <- data.frame(date = data_seq_YR, load = NN_for_p1P1_YR$mean)

write.csv(Neural.Network.p1P1.22.6787.YR_df, row.names = FALSE,
          file = "./Competition/Data/Neural.Network.Forecast.p1P1.22.seed6787.csv")
```

##Neural_Network, k= 2,2,p=1,P=4

```
set.seed(6787)
NN_fit_p1P4_YR <- nnetar(ts_load_daily,p=1,P=4,xreg=fourier(ts_load_daily, K=c(2,2)))
```

```
## Warning in nnetar(ts_load_daily, p = 1, P = 4, xreg = fourier(ts_load_daily, :
## Missing values in x, omitting rows
```

```
NN_for_p1P4_YR <- forecast(NN_fit_p1P4_YR, h=59,xreg=fourier(ts_load_daily,
                                          K=c(2,2),h=59))


#save
Neural.Network.p1P4.22.6787.YR_df <- data.frame(date = data_seq_YR, load = NN_for_p1P4_YR$mean)

write.csv(Neural.Network.p1P4.22.6787.YR_df, row.names = FALSE,
          file = "./Competition/Data/Neural.Network.Forecast.p1P4.22.seed6787.csv")
```