pre-assignment

WoodyLIver

27 04 2021

Libraries

```r
library(openxlsx)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(forecast)
library(EnvStats)
```

```
##
## Attaching package: 'EnvStats'

## The following objects are masked from 'package:stats':
##
##     predict, predict.lm

## The following object is masked from 'package:base':
##
##     print.default
```

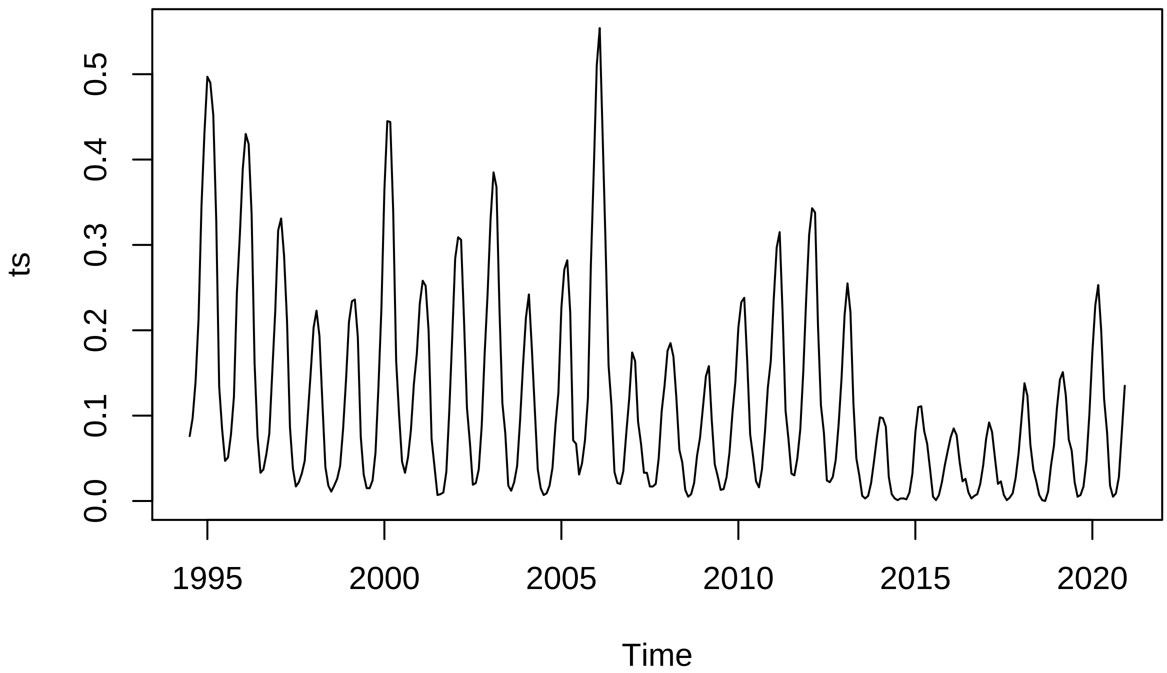
```r
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

```r
library(sigmoid)
```

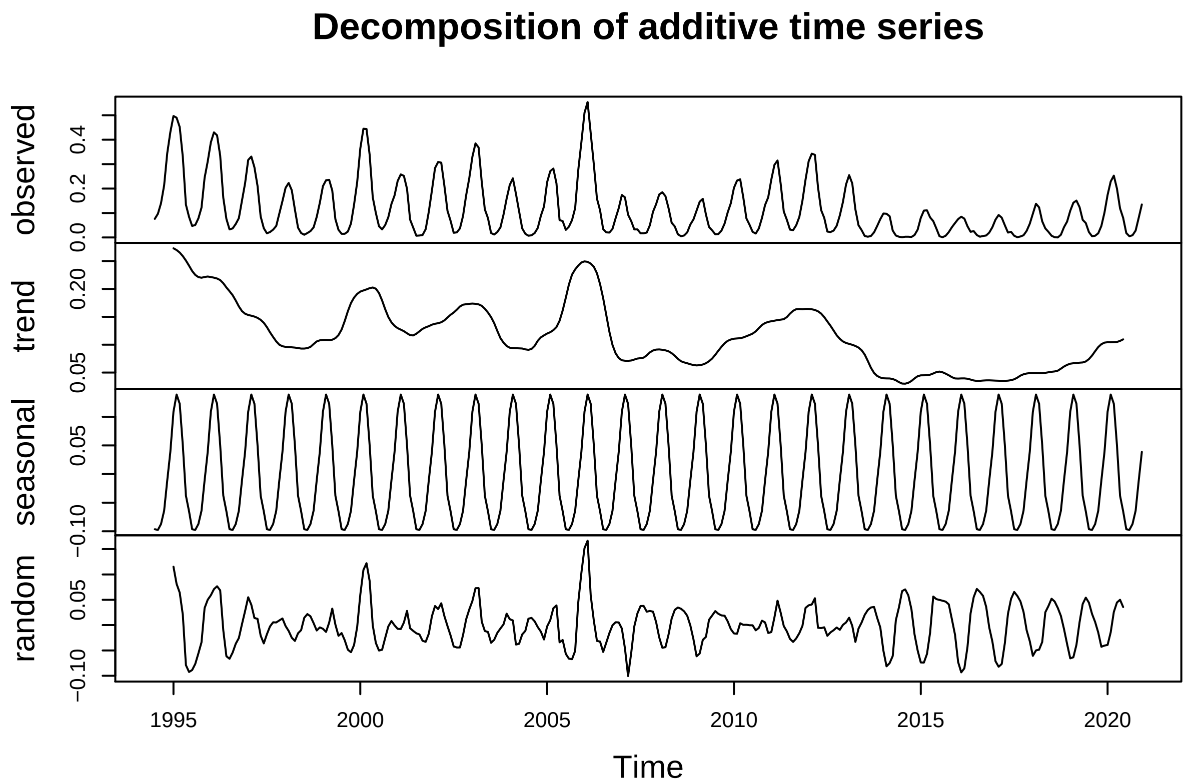
Read data

```r
data <- read.xlsx("Lions_Den_data.xlsx")
ts <- ts(unlist(data[2]), start=c(1994, 7), frequency=12)
plot(ts)
```



Decomposition

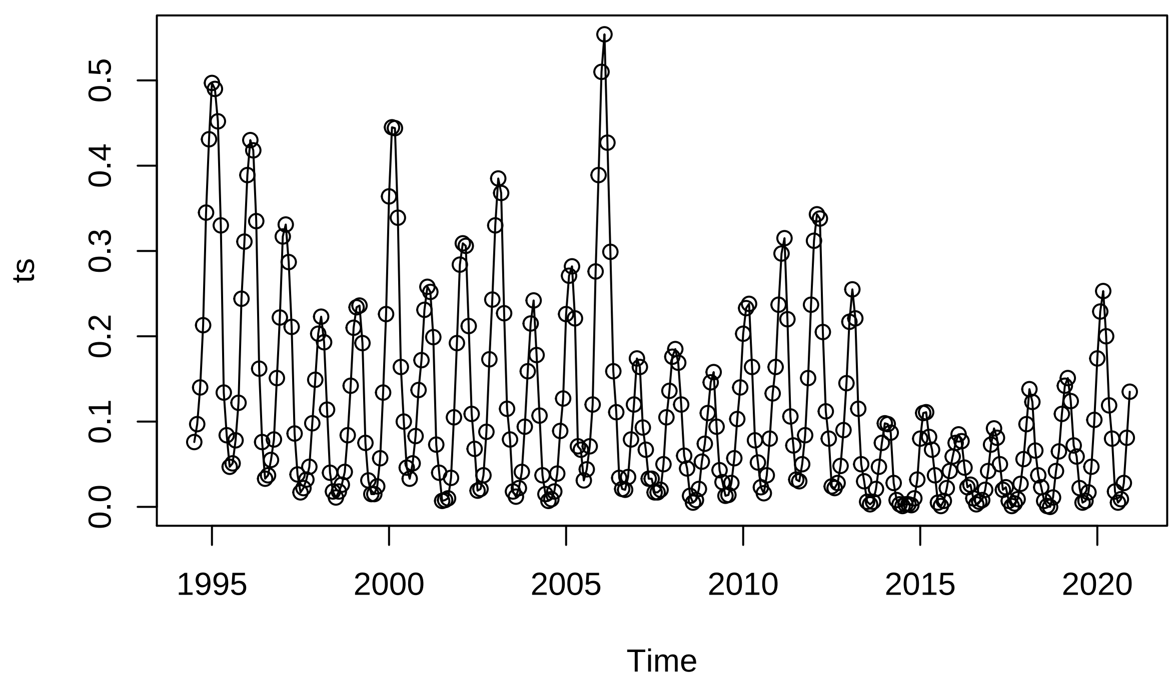
```r
decompose <- decompose(ts, "additive")
plot(decompose)
```



Outliers

Test on raw data that the winter 2006 does not match. We have tried to manipulate data(differentiation, decomposition, etc) in order to get more outliers, but there were no reasonable results.

```
plot(ts, type="o")
```



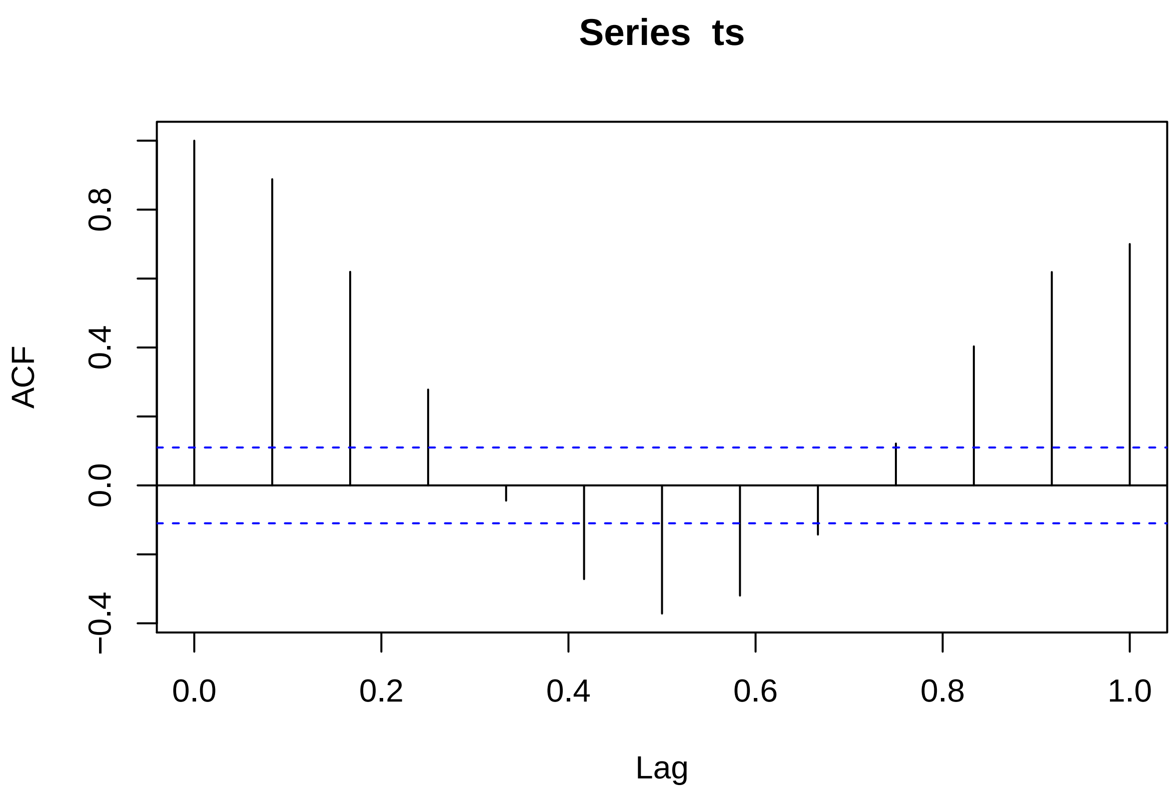
```r
rosnerTest(ts, k = 3)
```

```
##
## Results of Outlier Test
## -------------------------
##
## Test Method:                     Rosner's Test for Outliers
##
## Hypothesized Distribution:       Normal
##
## Data:                            ts
##
## Sample Size:                     318
##
## Test Statistics:                 R.1 = 3.782021
##                                  R.2 = 3.487550
##                                  R.3 = 3.445420
##
## Test Statistic Parameter:        k = 3
##
## Alternative Hypothesis:          Up to 3 observations are not
##                                  from the same Distribution.
##
## Type I Error:                    5%
##
## Number of Outliers Detected:     1
##
##   i    Mean.i      SD.i Value Obs.Num    R.i+1 lambda.i+1 Outlier
## 1 0 0.1170692 0.1155284 0.554     140 3.782021   3.739949    TRUE
## 2 1 0.1156909 0.1130619 0.510     139 3.487550   3.739067   FALSE
## 3 2 0.1144430 0.1110335 0.497       7 3.445420   3.738181   FALSE
```

Autocorrelation and stationarity analysis

We can see that time-series is already stationary and is slightly correlated with itself 12 month earlier.

```r
#autocorrelation
acf(ts, lag.max = 12)
```




```r
#stationary test
adf.test(ts)
```

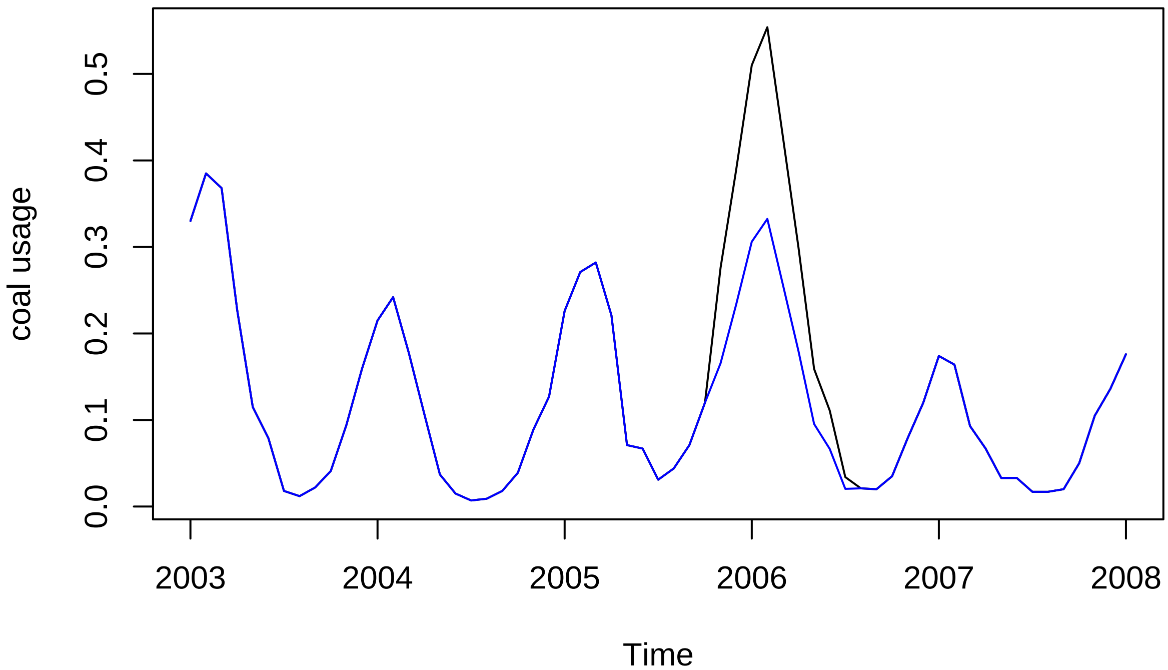
```
## Warning in adf.test(ts): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts
## Dickey-Fuller = -4.9571, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

PREDICTION

Delete outlier

```r
ts_ <- ts
plot(window(ts_, 2003, 2008), type="l", ylab="coal usage")
ts_[137:145] = ts_[137:145] * 0.6
lines(window(ts_, 2003, 2008), type="l", col="blue")
```

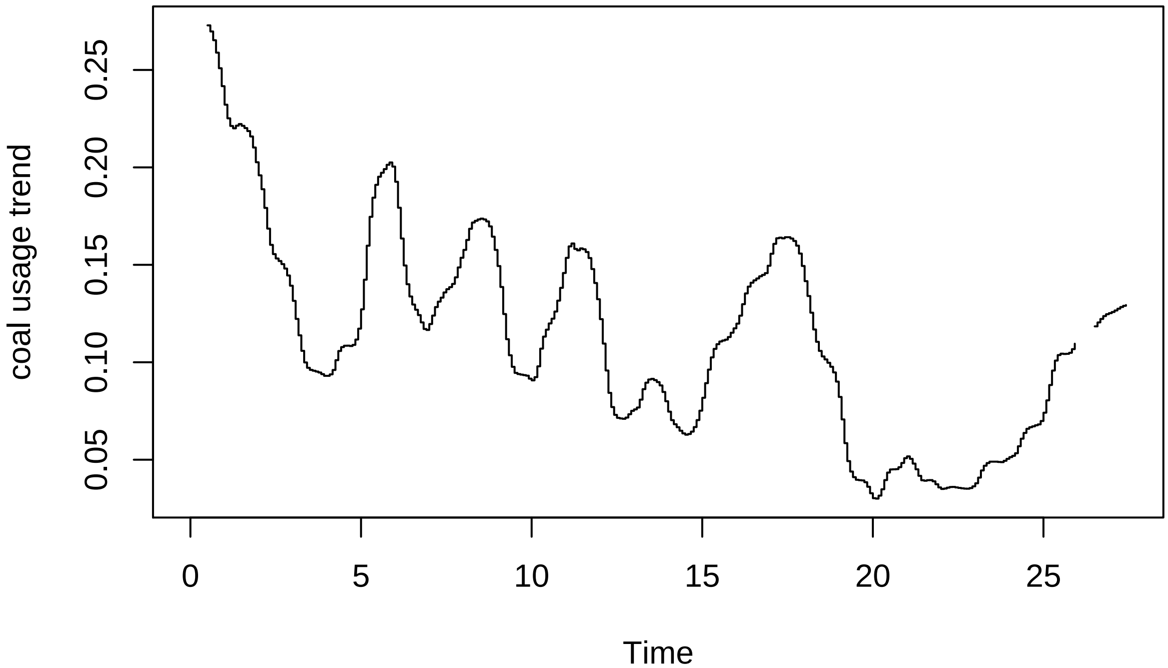


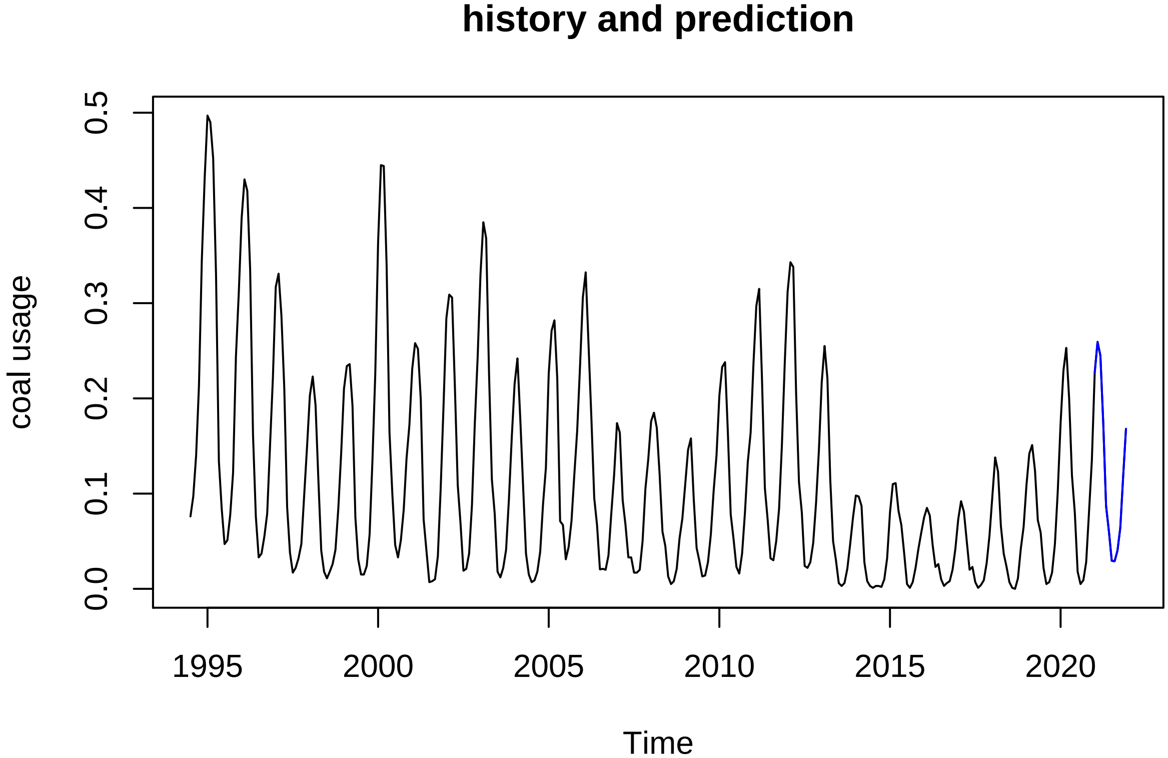
Create model Model performs decomposition into trend and seasonality, then trend is predicted by ARIMA. Result is a sum of trend prediction and seasonality pushed through relu.

```
model <- function (ts) {
  decompose_ <- decompose(ts, "additive")
  tsTrend <- ts(decompose_$trend, start=ts, frequency=12)
  fitARIMA <- arima(tsTrend, order=c(1,1,1),seasonal = list(order = c(1,0,0), period = 12),method="ML")
  yTrend <- predict(fitARIMA,n.ahead = 12)$pred
  cTrend <- ts(
    c(tsTrend, yTrend),
    start=start(tsTrend),
    frequency=12
  )
  plot(cTrend, type="s", ylab="coal usage trend")
  ySeasonal <- window(decompose$seasonal, start=end(ts)[1], end=end(ts))
  Y <- ts(as.numeric(yTrend) + as.numeric(ySeasonal),
      start=start(ySeasonal)[1]+1,
      frequency=12)
  Y <- relu(Y)
  TS <- ts(c(ts, Y), start=start(ts), frequency=12)
  plot(TS, type="l", ylab="coal usage",
      main="history and prediction", col="black")
  lines(Y, col="blue")
  print(Y)
  return(Y)
}
```

Predict

```
model(ts_)
```



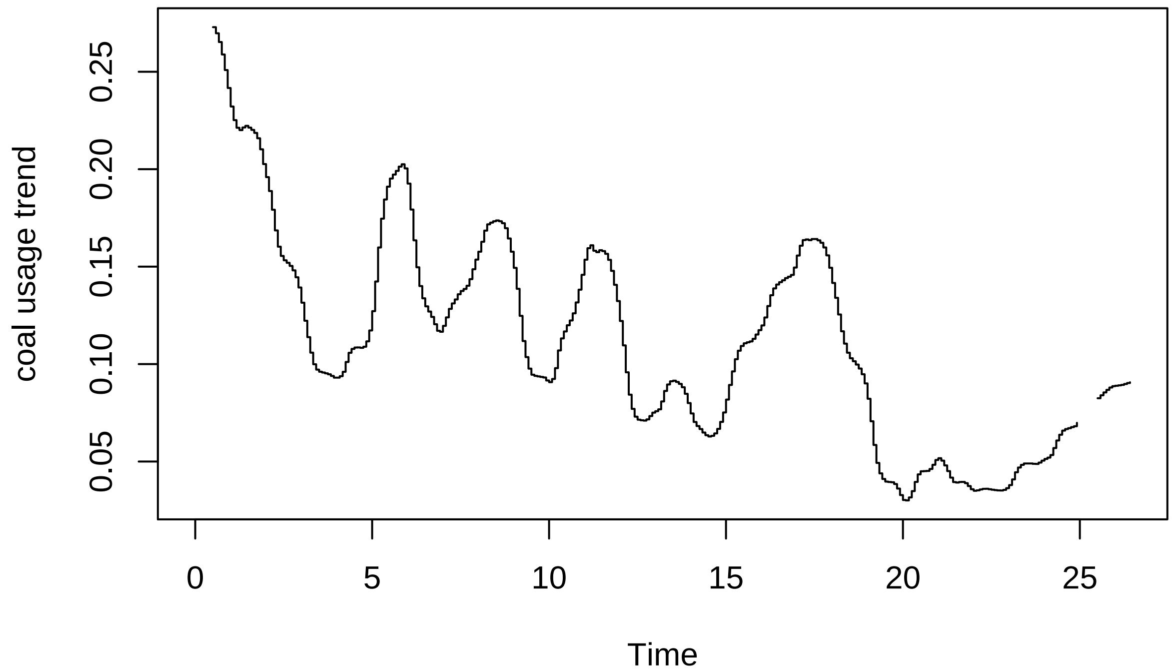


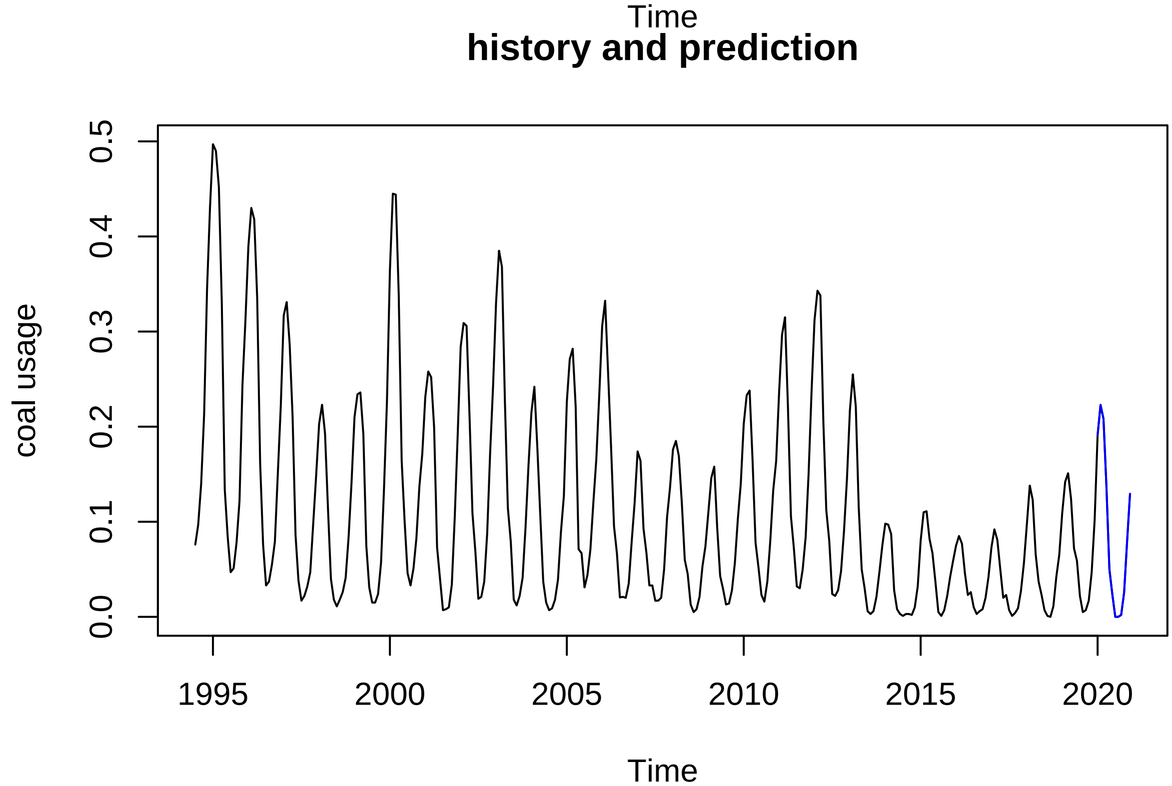

```
##               Jan        Feb        Mar        Apr        May        Jun
## 2021 0.22719068 0.25933252 0.24493153 0.17419903 0.08686344 0.06006777
##               Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.02937555 0.02900797 0.04012715 0.06425125 0.11783758 0.16806379

##               Jan        Feb        Mar        Apr        May        Jun
```

```
## 2021 0.22719068 0.25933252 0.24493153 0.17419903 0.08686344 0.06006777
##              Jul        Aug        Sep        Oct        Nov        Dec
## 2021 0.02937555 0.02900797 0.04012715 0.06425125 0.11783758 0.16806379
```

#Evaluate Both, plot and acuracy test shows that our model performs much better than naive method.

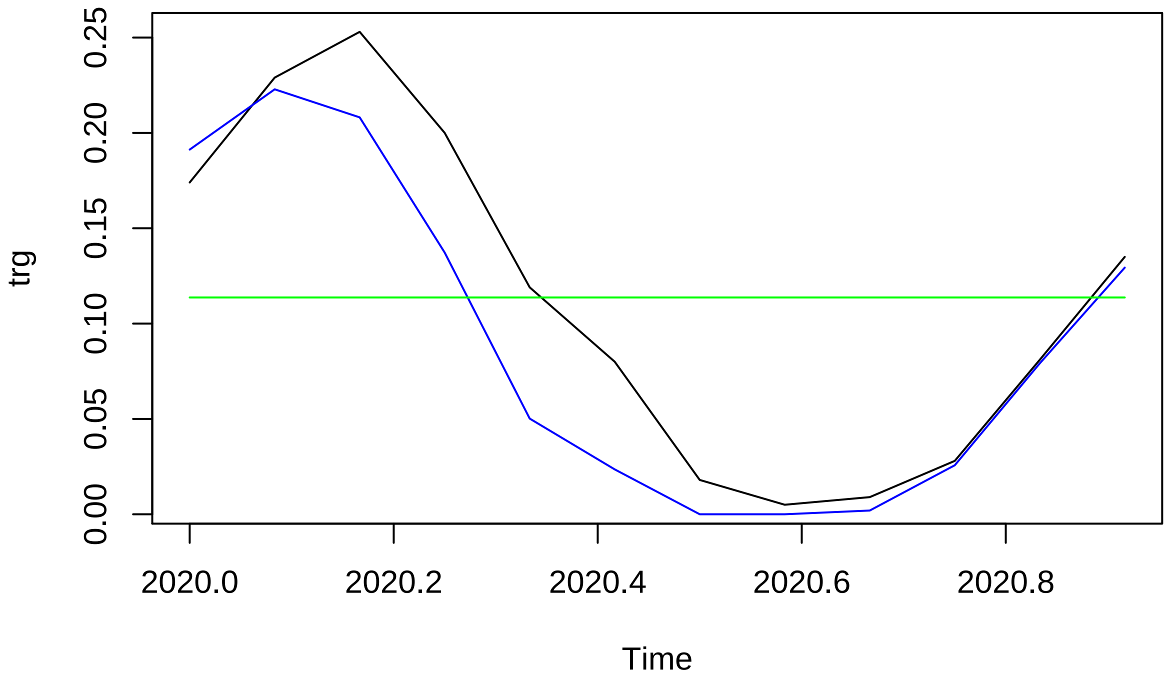
```
arg <- window(ts_, end=c(2019,12))
trg <- window(ts_, start=2020)
ret <- model(arg)
```





```
##              Jan         Feb         Mar         Apr         May         Jun
## 2020 0.191243014 0.222871632 0.208157240 0.137241529 0.050175519 0.023534018
##              Jul         Aug         Sep         Oct         Nov         Dec
## 2020 0.000000000 0.000000000 0.001965331 0.025700503 0.079150253 0.129369184
```

```
naive <- ts(rep(mean(arg), 12), start=2020, frequency=12)
plot(trg)
lines(ret, col="blue")
lines(naive, col="green")
```




```
accuracy(naive, trg)
```

```
##                     ME      RMSE        MAE       MPE     MAPE      ACF1
## Test set -0.002787255 0.0845331 0.07408333 -337.8586 371.7638 0.8297481
##          Theil's U
## Test set  7.305112
```

```
accuracy(ret, trg)
```

```
##                   ME       RMSE        MAE      MPE     MAPE      ACF1 Theil's U
## Test set 0.02179931 0.03493863 0.02467315 38.59319 40.24482 0.6629242 0.4960746
```