Exercise 2

Deborah Kewon

August 10, 2019

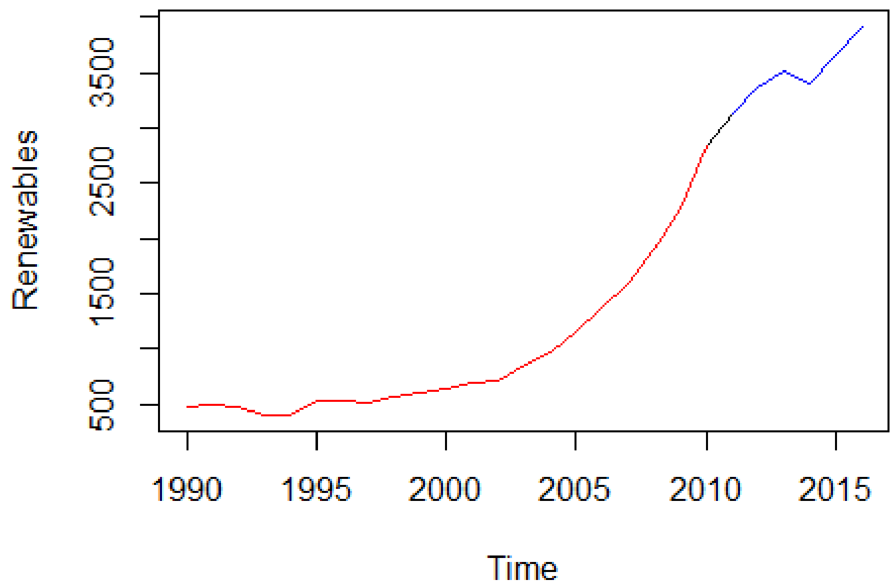
```
if (!require("fpp2")) install.packages("fpp2"); library(fpp2)
if (!require("portes")) install.packages("portes"); library(portes)
if (!require("readxl")) install.packages("readxl"); library(readxl)
if (!require("tseries")) install.packages("tseries"); library(tseries)
if (!require("lmtest")) install.packages("lmtest"); library(lmtest)
if (!require("forecast")) install.packages("forecast"); library(forecast)
if (!require("dplyr")) install.packages("dplyr"); library(dplyr)
options(digits=4, scipen=0)
```

Exercise 2

1. Exploring data

The data set Energy shows the yearly gross inland consumption of renewable energies (wind power and renewables) in the European Union, in thousand tonnes of oil equivalent (TOE) from 1990 up to 2016

```
setwd("C:\\Users\\dkewon\\Desktop\\retake\\final") # read the data
data <- read_excel("DataSets.xlsx", sheet = "Energy")
#using renewables
energy_consum <- ts(data['Renewables'], frequency = 1, start =
1990) # Split the data into training and test set
t_train <- window(energy_consum, end=2010)
t_test <- window(energy_consum, start=2011)
# Retrieve the length of the test set
h <- length(t_test)
# Plot the data
par(mfrow=c(1,1))
plot(energy_consum)
lines(t_train, col="red")
lines(t_test, col="blue")
```



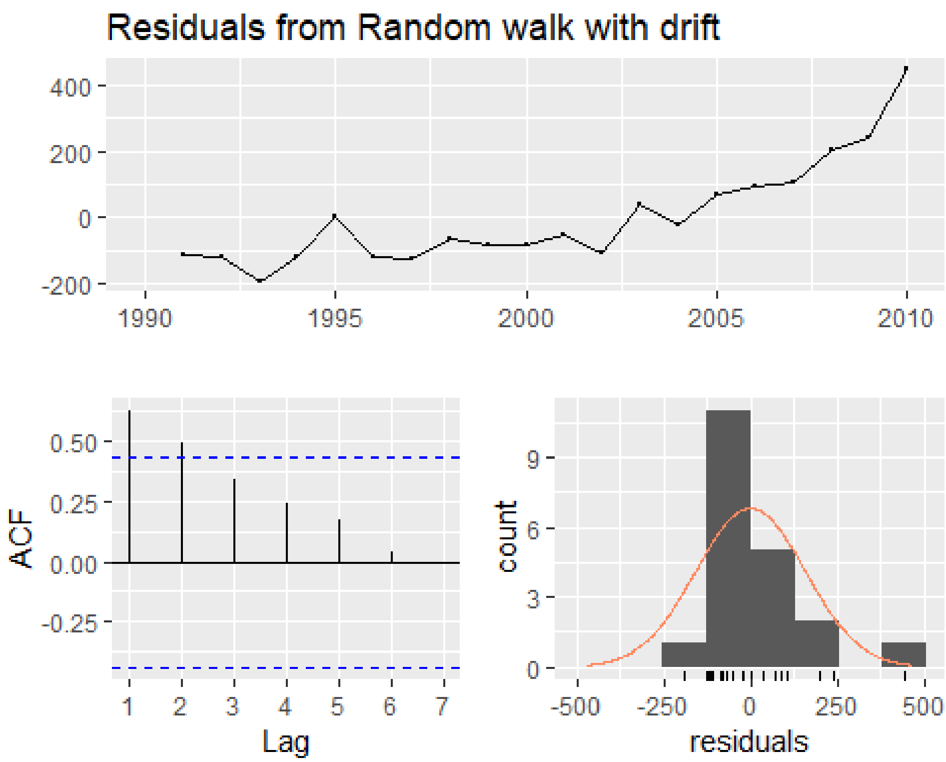
According to the graph above, the time series are not seasonal. The renewable energy consumption has been constantly increasing and exponentially since early 2000's.

2. Naive Method

As the time series are not seasonal and have constantly increased, we apply the naive method with drift method.

The residual analysis indicates that there is no white noise, which means there is still information we can capture.

```
fnaive <- rwf(t_train, drift=TRUE, h=length(t_test))
checkresiduals(fnaive)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Random walk with drift
## Q* = 20, df = 3, p-value = 2e-04
##
## Model df: 1.   Total lags used: 4
```

```r
res <- na.omit(fnaive$residuals)
LjungBox(res, lags=seq(1,20,4), order=0)
```

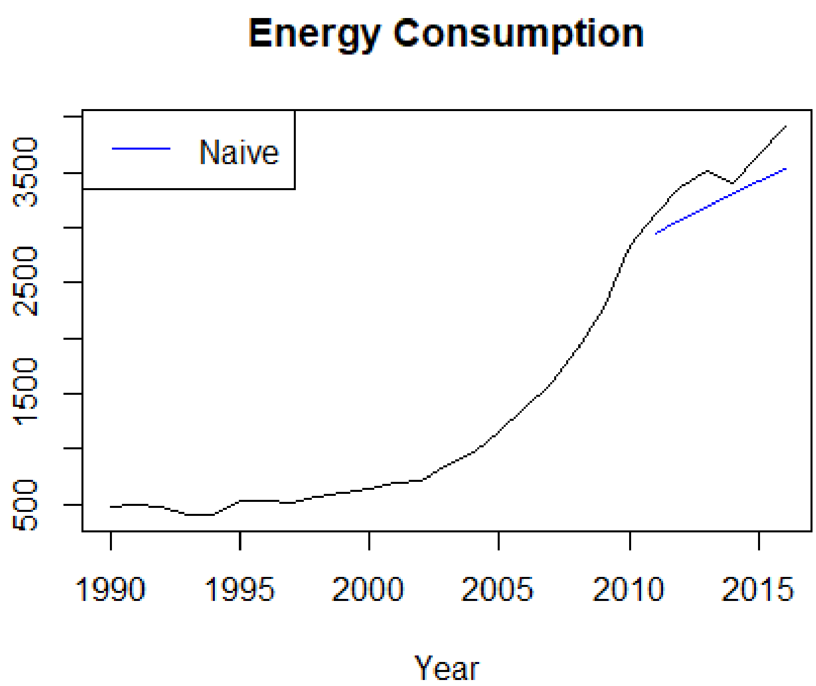
```
##  lags statistic df   p-value
##     1     9.099  1 2.558e-03
##     5    20.559  5 9.809e-04
##     9    21.719  9 9.814e-03
##    13    32.040 13 2.370e-03
##    17    61.908 17 5.072e-07
```

```r
accuracy(fnaive, t_test)[,c(2,3,5,6)]
```

```
##               RMSE   MAE   MAPE   MASE
## Training set 152.8 120.5 14.851 0.9513
## Test set     268.0 250.6  7.084 1.9792
```

```r
a_n<-accuracy(fnaive, t_test)[,c(2,3,5,6)]
a_train_n<-a_n[1,]
a_test_n <-a_n[2,]
plot(energy_consum,main="Energy Consumption", ylab="",xlab="Year")
```

```
lines(fnaive$mean, col=4)
legend("topleft",lty=1, col=c(4), legend = c("Naive"))
```



We notice that the accuracy is better for the training dataset (except MAPE). We are going to compare these results with other models later on.

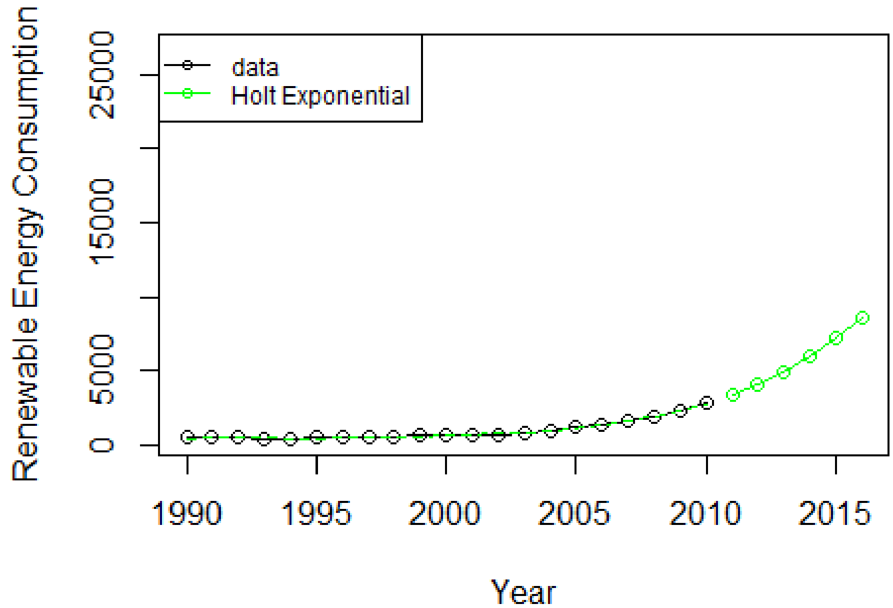
3. Exponential Smoothing Methods.

The renewable energy consumption has a positive trend and has constantly increased. For this reason, we choose the holt exponential smoothing method (damped).

Even though there is white noise, this model has the worse results than the previous model in terms of accuracy metrics especially for the testing dataset.

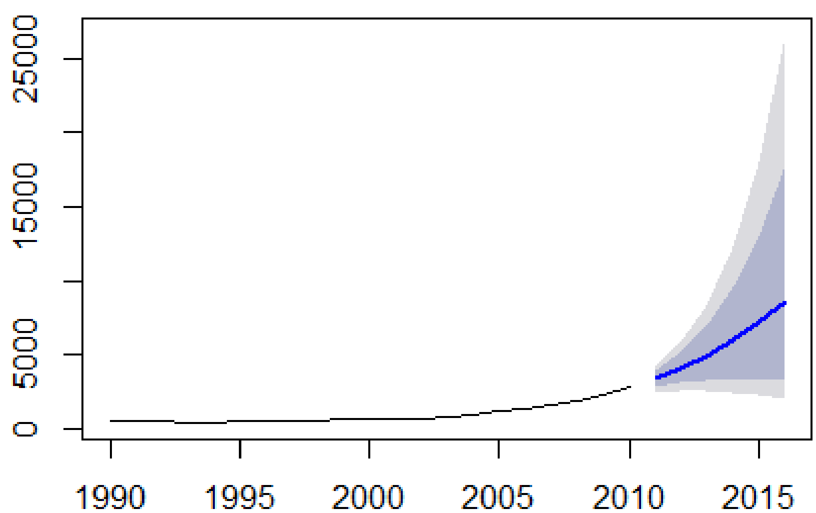
```
fit <- holt(t_train, exponential = TRUE, damped = TRUE,length(t_test))

plot(fit,ylab="Renewable Energy Consumption",
      shadecols = "white",
      type="o", fcol="white", xlab="Year")
 lines(fitted(fit), col="green", lty=2)
 lines(fit$mean, type="o", col="green")
 par_col <- c("black", "green", "green")
 legend("topleft",lty=1, pch=1, col=par_col,
      c("data","Holt Exponential"),cex = 0.75)
```

recasts from Damped Holt's method with exponentia

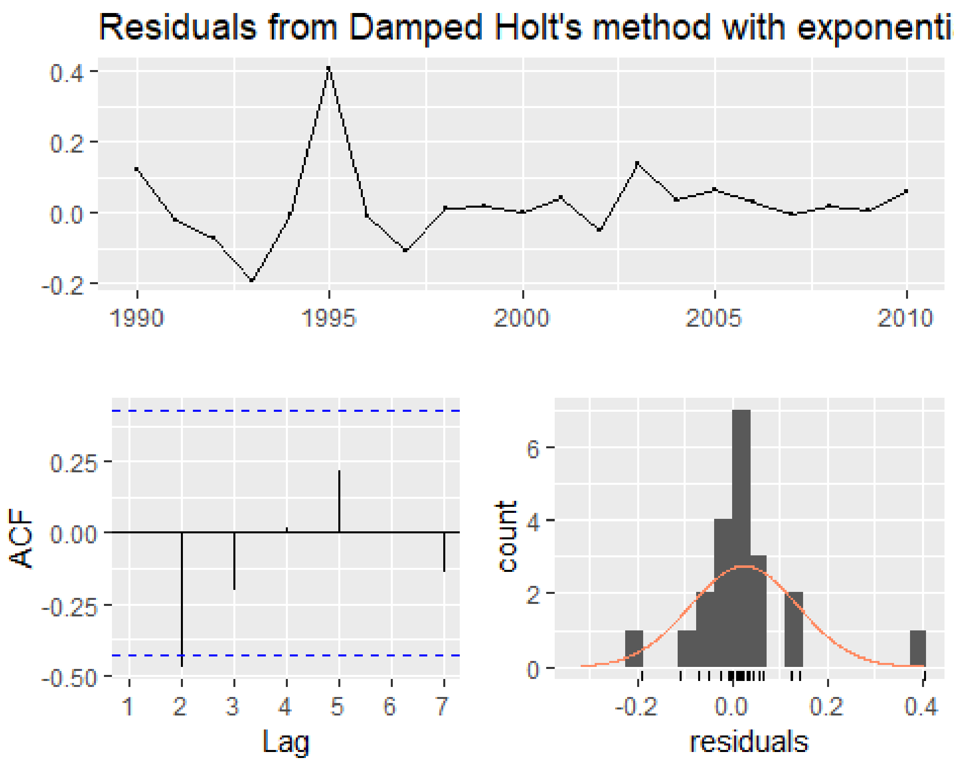



```
fcast<-forecast(fit, length(t_test))
plot(fcast)
```

recasts from Damped Holt's method with exponentia



```r
# Residual
checkresiduals(fcast)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Damped Holt's method with exponential trend
## Q* = 9.6, df = 3, p-value = 0.02
##
## Model df: 5.   Total lags used: 8
```

```r
res <- na.omit(fcast$residuals)
LjungBox(res, lags=seq(1,20,4), order=1)
```

```
##  lags statistic df p-value
##     1 7.539e-05  0 0.00000
##     5 8.170e+00  4 0.08554
##     9 9.692e+00  8 0.28728
##    13 1.054e+01 12 0.56873
##    17 1.121e+01 16 0.79611
```

```r
# Accuracy
print ("Accuracy")
```

```
## [1] "Accuracy"
```

```r
accuracy(fit,t_test)[,c(2,3,5,6)]  # test set
```

```
##                  RMSE     MAE   MAPE   MASE
## Training set   64.62   46.48  6.355  0.367
## Test set     2686.47 2197.11 60.322 17.351
```

```
a_h <- accuracy(fit,t_test)[,c(2,3,5,6)]
a_train_h <- a_h[1,]
a_test_h <- a_h[2,]
```

4.ETS.

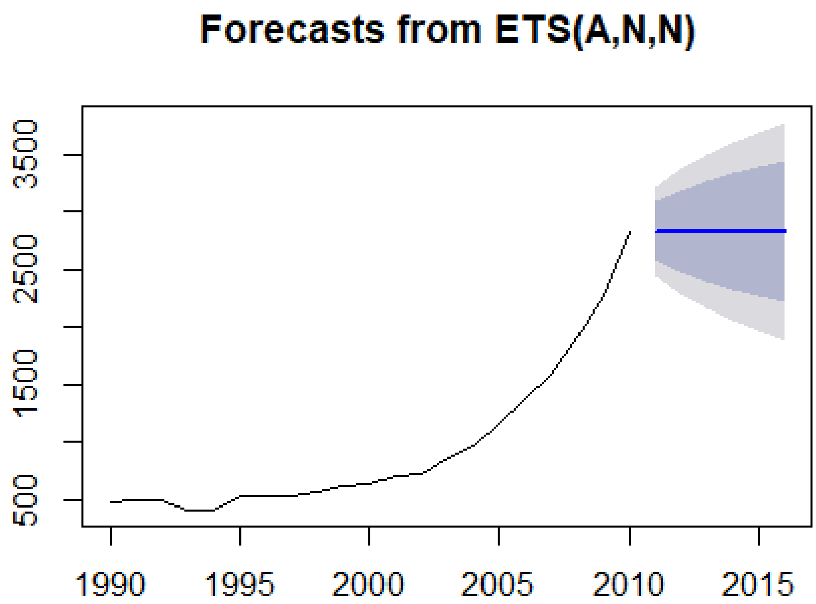
We test three ETS models such as ANN,MAN and MAdN for no seasonal time series and one auto ETS model(AAN).

The best fitting model is ETS(ANN) as it shows the lowest AIC.

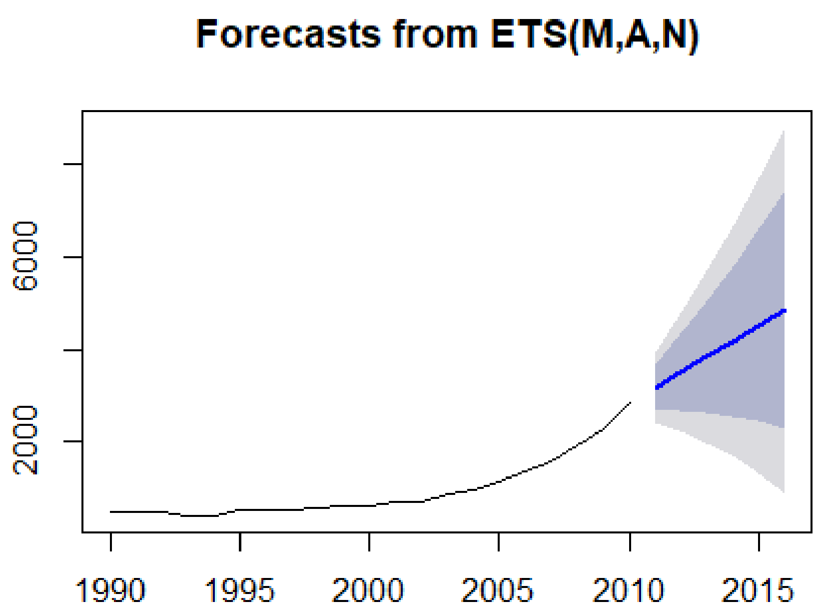
In terms of residual analysis,there is white noise for all the models except model ANN.

Considering the accuracy metrics, we choose the ETS(MAdN) model as a best model given the lowest RMSE and MASE in the testing dataset.

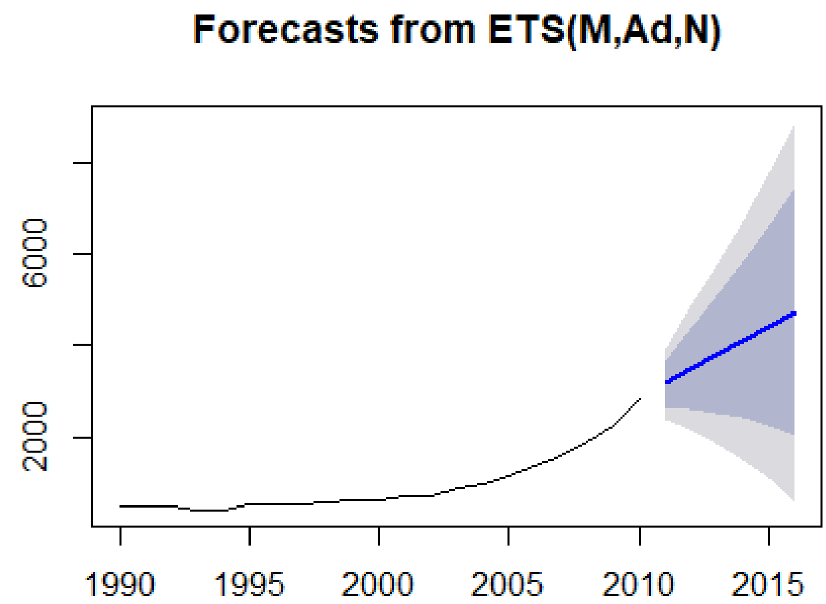
```
fit1<-ets(t_train,model = "ANN")
fit2<-ets(t_train,model = "MAN")
fit3<-ets(t_train,model = "MAN", damped = TRUE)
fit4<-ets(t_train)
fcast1<-forecast(fit1,h=length(t_test))
fcast2<-forecast(fit2,h=length(t_test))
fcast3<-forecast(fit3,h=length(t_test))
fcast4<-forecast(fit4,h=length(t_test))
plot(fcast1)
```



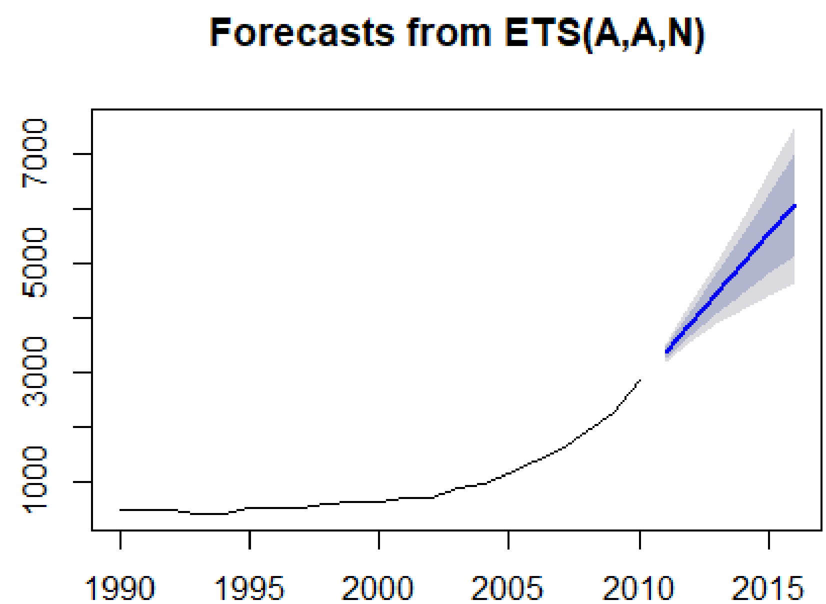

```
plot(fcast2)
```



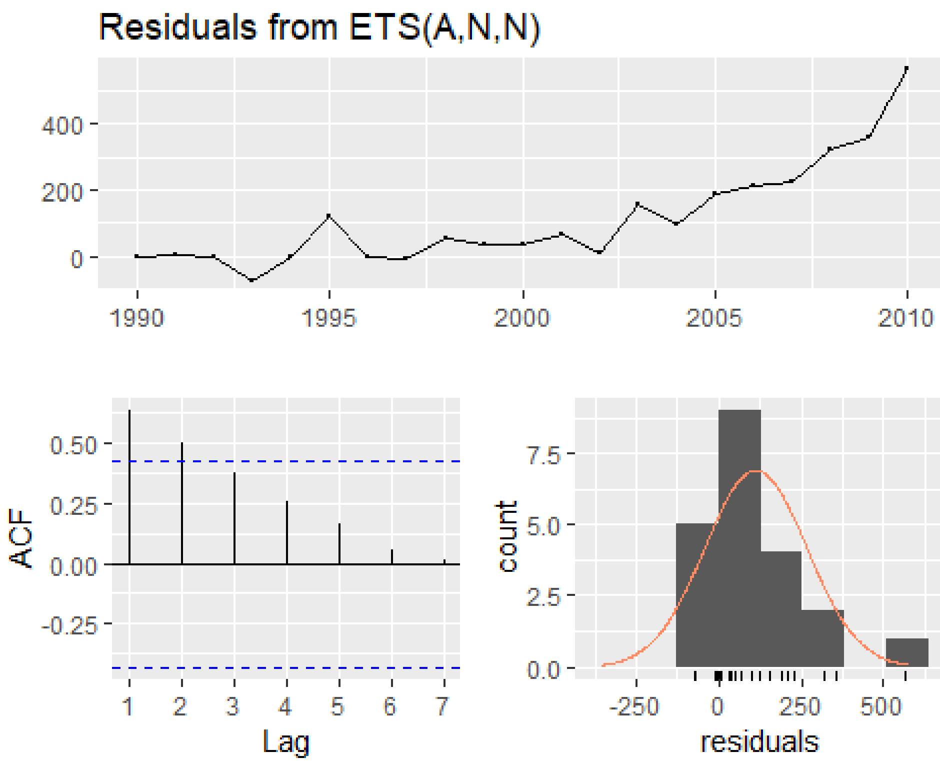

```
plot(fcast3)
```


Forecasts from ETS(M,Ad,N)
2000
6000
1990
1995
2000
2005
2010
2015

```
plot(fcast4)
```

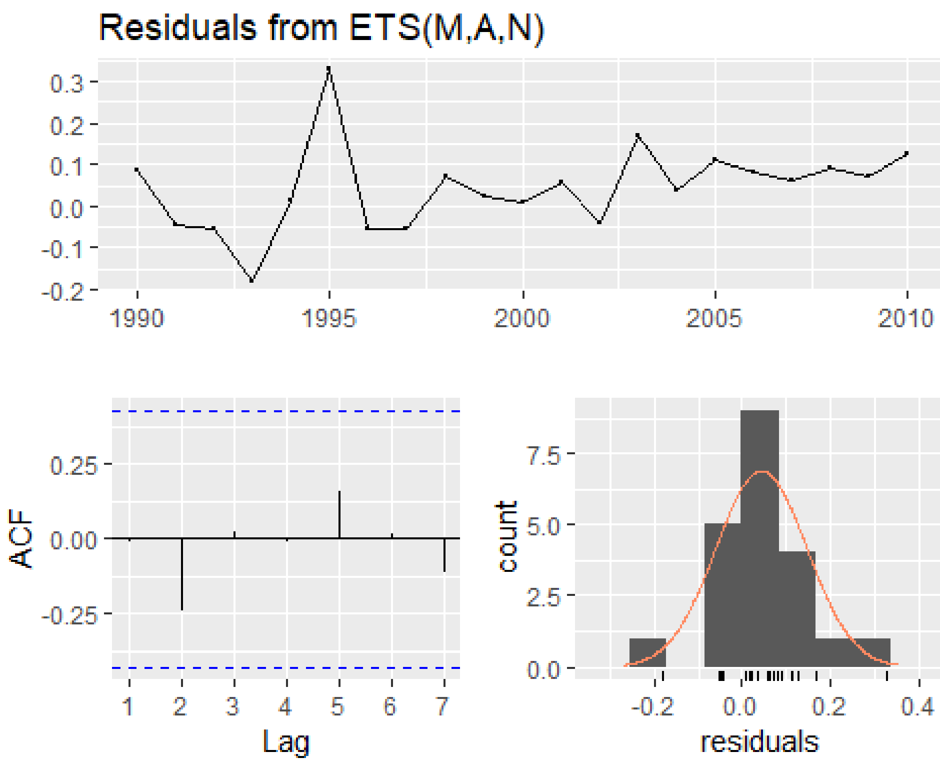

Forecasts from ETS(A,A,N)
1000
3000
5000
7000
1990
1995
2000
2005
2010
2015

```r
# Residual
checkresiduals(fcast1)
```



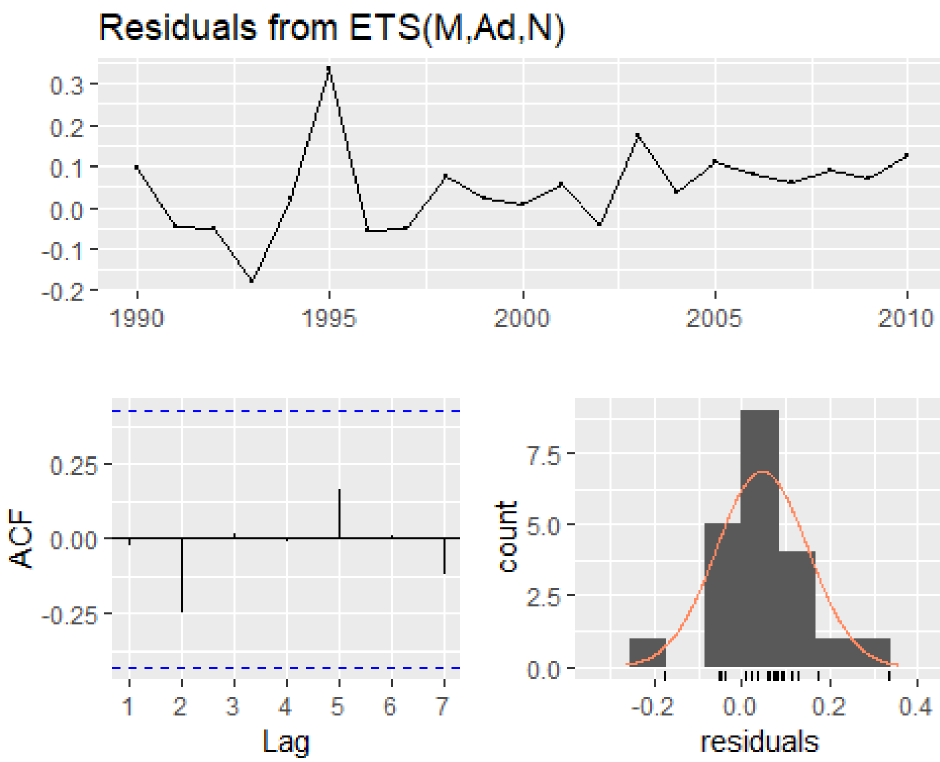

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,N,N)
## Q* = 22, df = 3, p-value = 5e-05
##
## Model df: 2.   Total lags used: 5
```

```r
checkresiduals(fcast2)
```

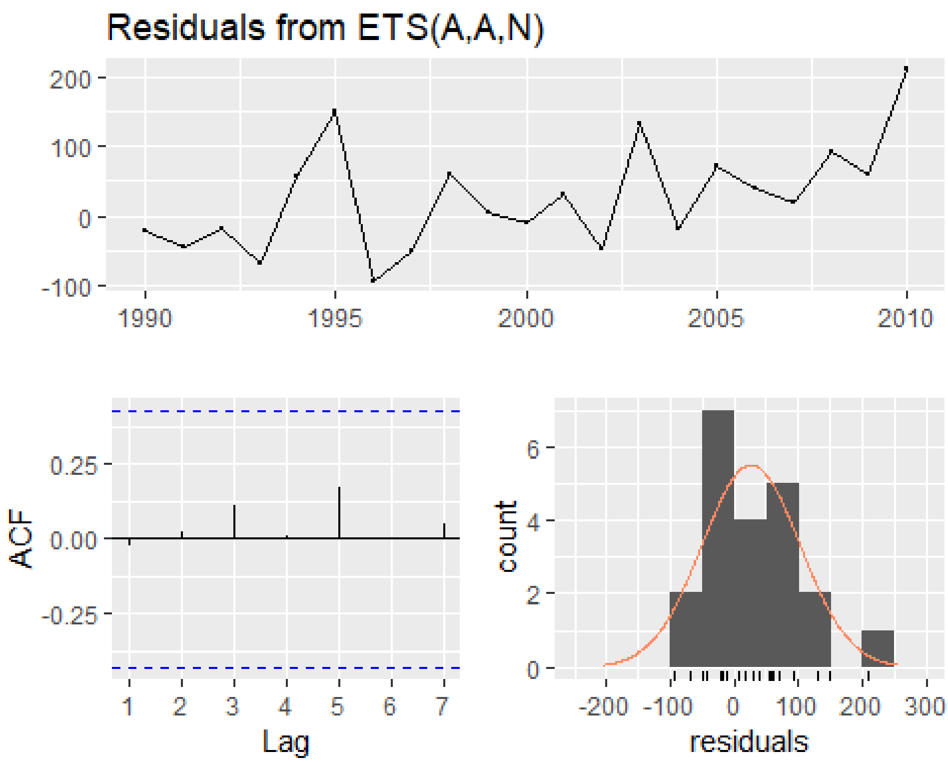



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,A,N)
## Q* = 2.7, df = 3, p-value = 0.4
##
## Model df: 4.   Total lags used: 7
```

```
checkresiduals(fcast3)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,Ad,N)
## Q* = 3.6, df = 3, p-value = 0.3
##
## Model df: 5.   Total lags used: 8

checkresiduals(fcast4)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,N)
## Q* = 1.3, df = 3, p-value = 0.7
##
## Model df: 4.   Total lags used: 7
```

```
# AIC
test1<- ets(t_test,fit1,use.initial.values = TRUE)  # test set
test2<- ets(t_test,fit2,use.initial.values = TRUE)
test3<- ets(t_test,fit3,use.initial.values = TRUE)
test4<- ets(t_test,fit4,use.initial.values = TRUE)
print(" AIC | AICc | BIC ")
```

```
## [1] " AIC | AICc | BIC "
```

```
round(c(test1$aic,test1$aicc,test1$bic),4)
```

```
## [1] 100.7 112.7 100.1
```

```
round(c(test2$aic,test2$aicc,test2$bic),4)
```

```
## [1] 126.5   Inf 125.5
```

```
round(c(test3$aic,test3$aicc,test3$bic),4)
```

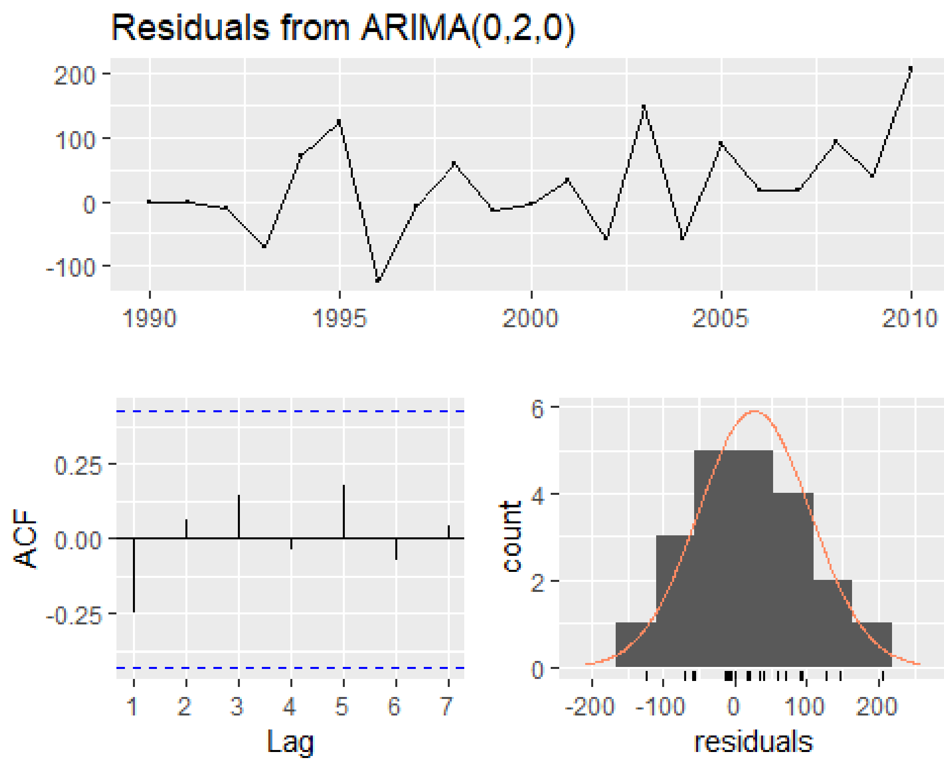
```
## [1] 128.57  44.57 127.32
```

```
round(c(test4$aic,test4$aicc,test4$bic),4)

## [1] 107.3   Inf 106.3

# Accuracy
round(accuracy(fcast1,t_test)[,c(2,3,5,6)],4)

##               RMSE   MAE   MAPE   MASE
## Training set 188.2 120.6  9.662 0.9525
## Test set     707.5 662.2 18.538 5.2292

round(accuracy(fcast2,t_test)[,c(2,3,5,6)],4)

##               RMSE    MAE   MAPE   MASE
## Training set 105.1  76.04  7.843 0.6005
## Test set     621.8 512.51 14.155 4.0473

round(accuracy(fcast3,t_test)[,c(2,3,5,6)],4)

##               RMSE    MAE   MAPE   MASE
## Training set 106.2  76.85  7.933 0.6069
## Test set     563.0 465.63 12.879 3.6771

round(accuracy(fcast4,t_test)[,c(2,3,5,6)],4)

##                 RMSE     MAE   MAPE   MASE
## Training set   79.12   61.63  8.073 0.4867
## Test set     1409.31 1220.55 33.837 9.6387

a_e3<-accuracy(fcast3,t_test)[,c(2,3,5,6)]
a_train_e3<-a_e3[1,]
a_test_e3<-a_e3[2,]
```

5.ARIMA.

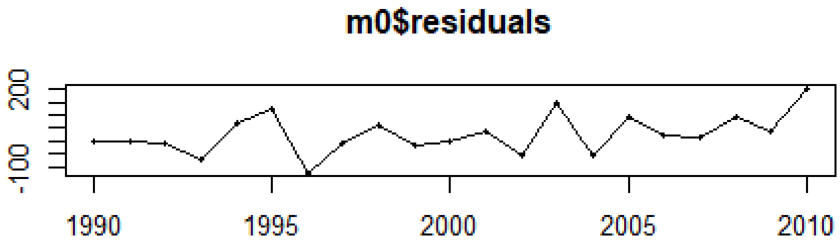
There is white noise; no information is left with the auto arima model Arima(1,2,1).We start by differencing the data; two differences are suggested

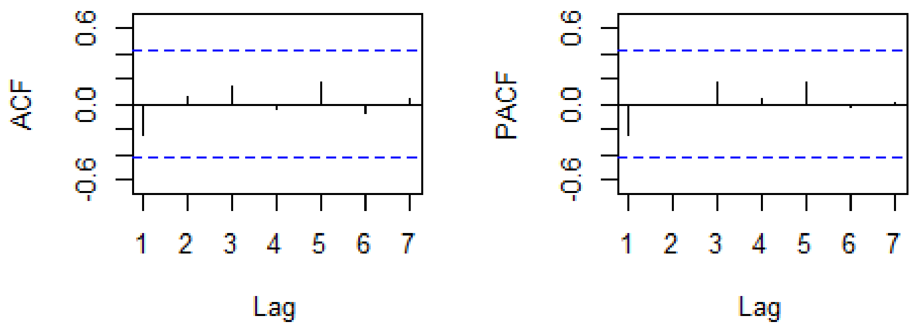
```
ndiffs(t_train)

## [1] 2

m0 <- auto.arima(t_train)
checkresiduals(m0)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,2,0)
## Q* = 2.2, df = 4, p-value = 0.7
##
## Model df: 0.   Total lags used: 4
```

```r
tsdisplay(m0$residuals)
```





Using the getinfo() function, we are going to compute more Arima models with 2 differences Arima(i,2,j).

```
getinfo <- function(x,h,...) {
  train.end <- time(x)[length(x)-h]
  test.start <- time(x)[length(x)-h+1]
  train <- window(x,end=train.end)
  test <- window(x,start=test.start)
  fit <- Arima(train,...)
  fc <- forecast(fit,h=h)
  a <- accuracy(fc,test)
  result <- matrix(NA, nrow=1, ncol=5)
  result[1,1] <- fit$aicc
  result[1,2] <- a[1,6]
  result[1,3] <- a[2,6]
  result[1,4] <- a[1,2]
  result[1,5] <- a[2,2]
  return(result)
}

mat <- matrix(NA,nrow=72, ncol=5)
modelnames <- vector(mode="character", length=72)
line <- 0
for (p in 1:6){
  for (q in 1:6){
    for (d in 1:2){
        line <- line+1
```

```r
        mat[line,] <- getinfo(energy_consum,h=h,order=c(p,d,q), method="ML")
        modelnames[line] <- paste0("ARIMA(",p,",",d,",",q,")")
    }
  }
}
colnames(mat) <- c("AICc", "MASE_train", "MASE_test", "RMSE_train", "RMSE_tes
t")
rownames(mat) <- modelnames
```

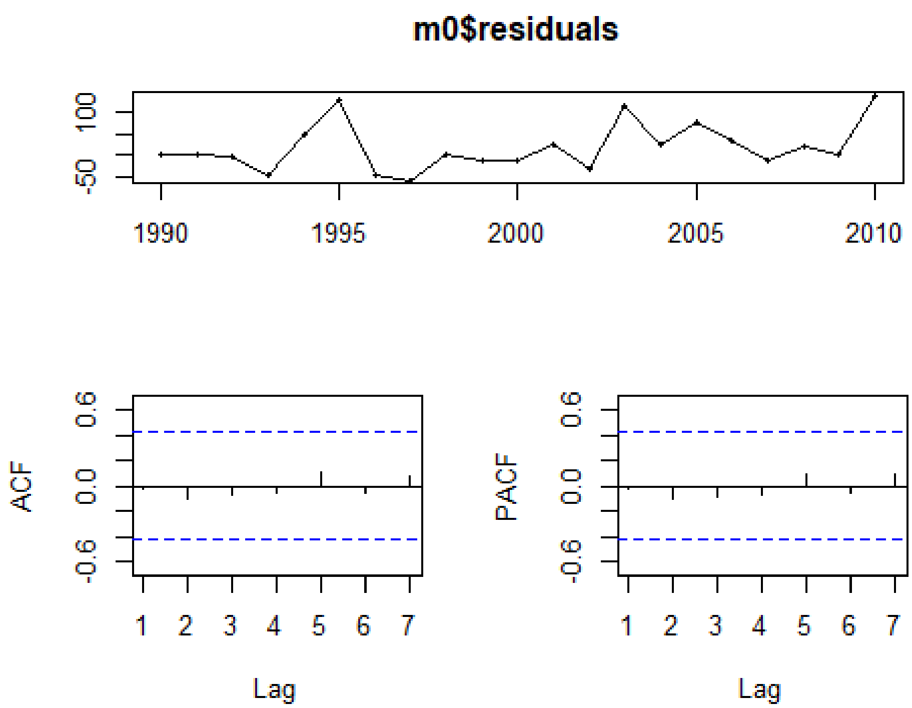
Then, we select the best models in terms of AICc, MASE and RMSE respectively. After that, we will compute each model separately more in detail.

```r
print("best AICc")

## [1] "best AICc"

which(mat[,1]==min(mat[,1]))

## ARIMA(1,2,2)
##            4

print("best MASE_train")

## [1] "best MASE_train"

which(mat[,2]==min(mat[,2]))

## ARIMA(5,2,4)
##           56

print("best MASE_test")

## [1] "best MASE_test"

which(mat[,3]==min(mat[,3]))

## ARIMA(3,1,1)
##           25

print("best RMSE_train")

## [1] "best RMSE_train"

which(mat[,4]==min(mat[,4]))

## ARIMA(6,2,6)
##           72

print("best RMSE_test")

## [1] "best RMSE_test"

which(mat[,5]==min(mat[,5]))
```

```
## ARIMA(3,1,1)
##            25
```

Computing Auto Arima model: Arima(1,2,2)

```
m0<-Arima(t_train,order=c(1,2,2),method = 'ML')
coef(m0)

##    ar1    ma1    ma2
##  0.940 -1.683  1.000

LjungBox(m0$residuals, lags=seq(length(m0$coef),20,4), order=length(m0$coef))

##  lags statistic df p-value
##     3    0.4202  0  0.0000
##     7    1.1202  4  0.8911
##    11    1.3751  8  0.9946
##    15    9.6374 12  0.6477
##    19   12.2992 16  0.7231

tsdisplay(m0$residuals)
```




```
f0 <- forecast(m0, h=length(t_test))
```

Computing ARIMA(5,2,4)

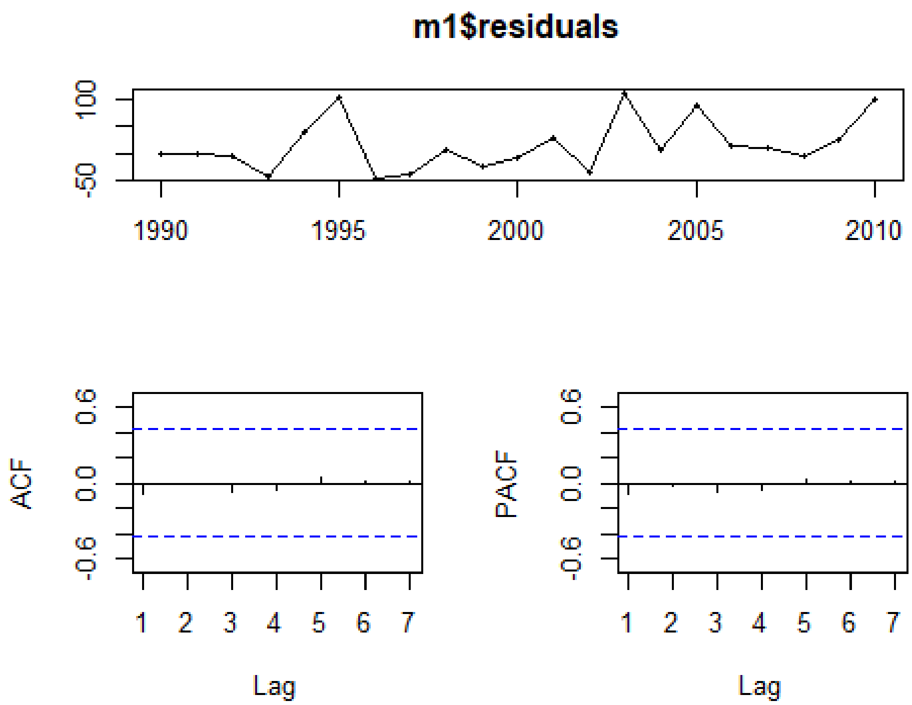
```
m1<-Arima(t_train,order=c(5,2,4))
coef(m1)
```

```
##       ar1      ar2      ar3      ar4      ar5      ma1      ma2      ma3
## -0.80521  0.72248  0.57690  0.07882  0.19424  0.33450 -1.28855  0.33433
##       ma4
##  0.99972
```

```
LjungBox(m1$residuals, lags=seq(length(m1$coef),20,4), order=length(m1$coef))
```

```
##  lags statistic df p-value
##     9    0.5823  0  0.0000
##    13    4.1051  4  0.3920
##    17    8.6373  8  0.3738
```

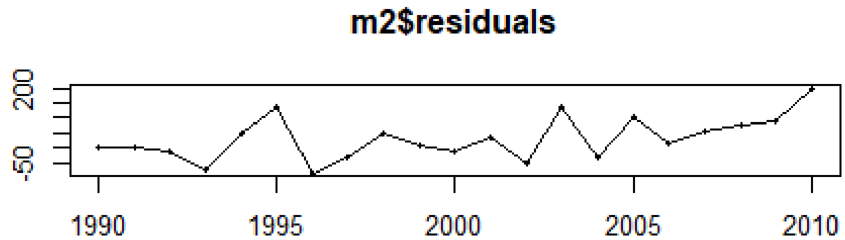
```
tsdisplay(m1$residuals)
```

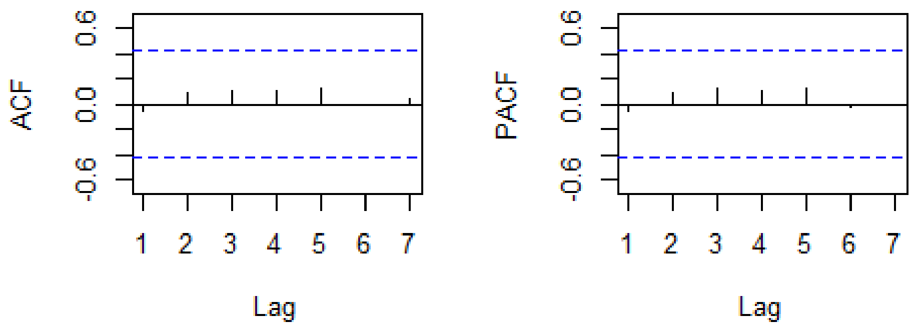



```
f1 <- forecast(m1, h=length(t_test))
```

Computing ARIMA(3,1,1)

```
m2 <- Arima(t_train, order=c(3,1,1), method = 'ML')
tsdisplay(m2$residuals)
```



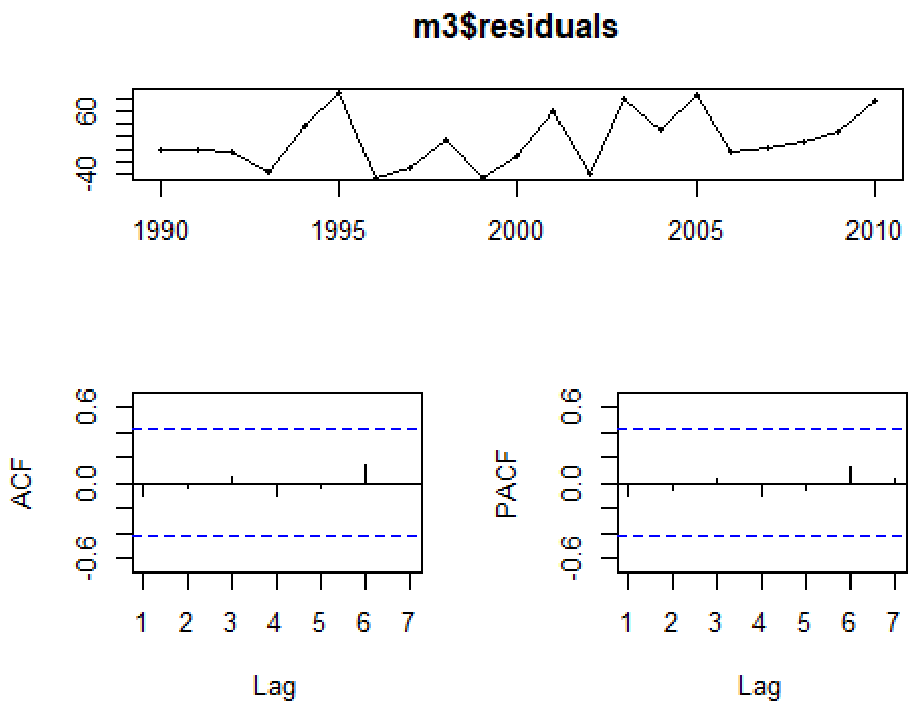



```
f2 <- forecast(m2, h=length(t_test))
```

Computing ARIMA(6,2,6)

```
m3 <- Arima(t_train, order=c(6,2,6))
coeftest(m3)

##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1    -1.207      0.434   -2.78   0.0054 **
## ar2    -0.395      0.414   -0.96   0.3395
## ar3     0.137      0.292    0.47   0.6380
## ar4     0.930      0.322    2.89   0.0039 **
## ar5     0.895      0.400    2.24   0.0252 *
## ar6     0.110      0.410    0.27   0.7892
## ma1     0.725      0.506    1.43   0.1523
## ma2    -0.168      0.465   -0.36   0.7172
## ma3     0.145      0.400    0.36   0.7163
## ma4    -0.168      0.460   -0.37   0.7143
## ma5     0.725      0.551    1.32   0.1880
## ma6     1.000      0.508    1.97   0.0493 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

LjungBox(m3$residuals, lags=seq(length(m3$coef),20,4), order=length(m3$coef))
```

```
##   lags statistic df p-value
##     12      4.491  0  0.0000
##     16      7.581  4  0.1082
##     20      9.087  8  0.3350

tsdisplay(m3$residuals)
```




```
f3 <- forecast(m3, h=length(t_test))

a_m0 <- accuracy(f0,t_test)[,c(2,3,5,6)]
a_m1 <- accuracy(f1,t_test)[,c(2,3,5,6)]
a_m2 <- accuracy(f2,t_test)[,c(2,3,5,6)]
a_m3 <- accuracy(f3,t_test)[,c(2,3,5,6)]

print("Accuracy for train")

## [1] "Accuracy for train"

a_train_a <- rbind(a_m0[1,], a_m1[1,], a_m2[1,],a_m3[1,])
rownames(a_train_a) <- c("a_m0", "a_m1", "a_m2","a_m3")
a_train_a

##         RMSE   MAE  MAPE   MASE
## a_m0 58.11 40.62 5.449 0.3208
## a_m1 49.89 35.29 4.792 0.2787
## a_m2 77.85 59.01 7.134 0.4660
## a_m3 45.83 35.53 4.911 0.2806
```

```r
print("Accuracy for test")

## [1] "Accuracy for test"

a_test_a <- rbind(a_m0[2,], a_m1[2,], a_m2[2,],a_m3[2,])
rownames(a_test_a) <- c("a_m0", "a_m1", "a_m2","a_m3")
a_test_a

##      RMSE  MAE  MAPE   MASE
## a_m0 2000 1668 45.95 13.173
## a_m1 1909 1589 43.77 12.550
## a_m2 1178 1025 28.45  8.093
## a_m3 1880 1563 43.05 12.344
```

Arima model ARIMA(1,2,2) has the best AICc index-best fitting model Considering the residual analysis, all the Arima models computed perform well. In terms of accuracy, model m2 = ARIMA(3,1,1) has the best performance in the test dataset. Therefore, we choose m2 = ARIMA(3,1,1) as a best model.

6. Selection of final model

In terms of AIC and residual analysis, the best model is ARIMA(3,1,1)- time series fit very well However, considering the accuracy index on the trainset, naive method (random walk with drift) has the best performance. As a result, we choose the naive model as a best model.

```r
print("Accuracy - Train ")

## [1] "Accuracy - Train "

final_train <- rbind(a_train_n, a_train_h, a_train_e3, a_train_a[3,])
rownames(final_train) <- c("naive", "holt", "ETS(MAdN)", "ARIMA(3,1,1)")
final_train

##                RMSE    MAE   MAPE   MASE
## naive        152.82 120.46 14.851 0.9513
## holt          64.62  46.48  6.355 0.3670
## ETS(MAdN)    106.15  76.85  7.933 0.6069
## ARIMA(3,1,1)  77.85  59.01  7.134 0.4660

print("Accuracy - Test ")

## [1] "Accuracy - Test "

final_test <- rbind(a_test_n,  a_test_h,  a_test_e3, a_test_a[3,])
rownames(final_test) <- c("naive", "holt", "ETS(MAdN)", "ARIMA(3,1,1)")
final_test

##              RMSE    MAE   MAPE   MASE
## naive         268  250.6  7.084  1.979
## holt         2686 2197.1 60.322 17.351
```

```
## ETS(MAdN)      563  465.6 12.879  3.677
## ARIMA(3,1,1) 1178 1024.9 28.447  8.093
```

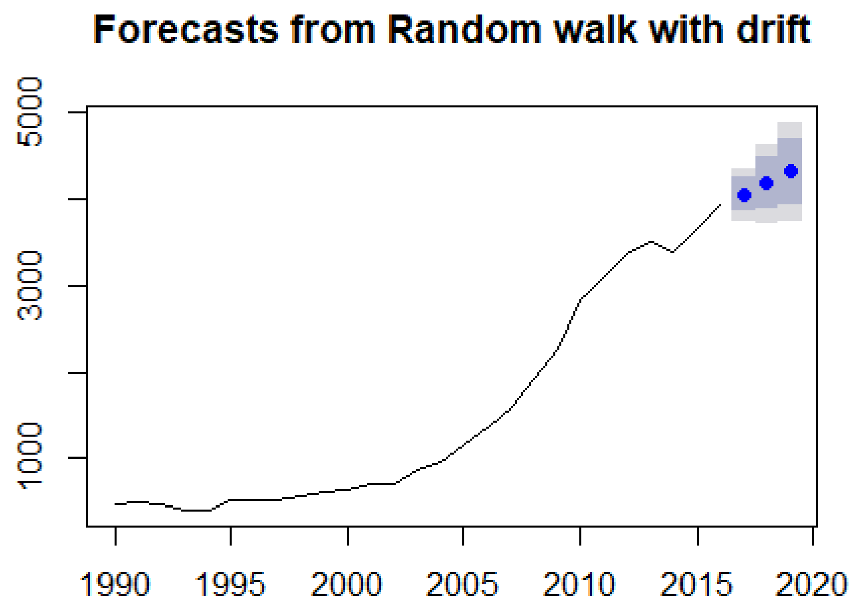
Forecast up to 2020

The naive (random walk with drift) model is corresponding to the trend.

```
model<-rwf(energy_consum, drift=TRUE, h=length(t_test))
summary(model)
```

```
##
## Forecast method: Random walk with drift
##
## Model Information:
## Call: rwf(y = energy_consum, h = length(t_test), drift = TRUE)
##
## Drift: 132.1038  (se 30.3256)
## Residual sd: 154.631
##
## Error measures:
##                    ME  RMSE   MAE    MPE  MAPE   MASE   ACF1
## Training set 1.53e-14 151.6 125.2 -7.913 13.26 0.8499 0.6183
##
## Forecasts:
##      Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
## 2017           4048  3846  4250  3739  4357
## 2018           4180  3884  4476  3727  4633
## 2019           4312  3937  4687  3739  4885
## 2020           4444  3998  4891  3761  5127
## 2021           4576  4062  5091  3790  5363
## 2022           4709  4129  5288  3823  5594
```

```
plot(forecast(model, h=3))
```


Forecasts from Random walk with drift
5000
3000
1000
1990
1995
2000
2005
2010
2015
2020