Energy in EU

25, July, 2019 Yen Chun, Liu

Package used

library(readxl)

library(knitr)

library(fpp2)

library(tseries)

library(portes)

Read in the data

```
data <- read_excel("DataSets.xlsx", sheet="Energy")
data$Windpower <-NULL
```

Data informaiton

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. For this analysis, use the Renewables" time series.

We will Split the data in a training set up to 2010 and a test set from 2011 up to 2016. Use the training set for estimation of the methods/models, and use the test set for assessing the forecast accuracy.

```
head(data)

## # A tibble: 6 x 2
##    Date Renewables
##   <dbl>      <dbl>
## 1  1990       481.
## 2  1991       487.
## 3  1992       483.
## 4  1993       409.
## 5  1994       405.
## 6  1995       528

tail(data)

## # A tibble: 6 x 2
##    Date Renewables
##   <dbl>      <dbl>
## 1  2011      3120.
## 2  2012      3366.
## 3  2013      3504.
## 4  2014      3398.
## 5  2015      3664.
## 6  2016      3916.
```

Change to time series format

```
to <- ts(data[,2], frequency = 1, start=c(1990))
```

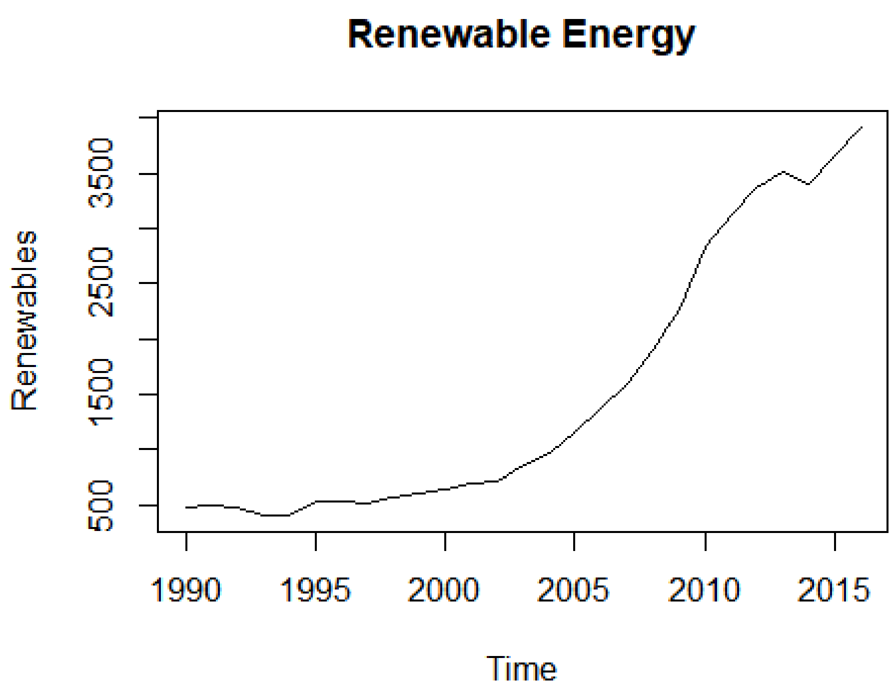
Split train and test

```
train <- window(to, start= c(1990), end= c(2010))
test <- window(to, start= c(2011),end= c(2016))
h = length(test)
```

Line plot

From the plot we can see that there's a upward trend from 1990 to 2016. There's no intensive fluctuation patterns.

```
plot(to, main="Renewable Energy")
```

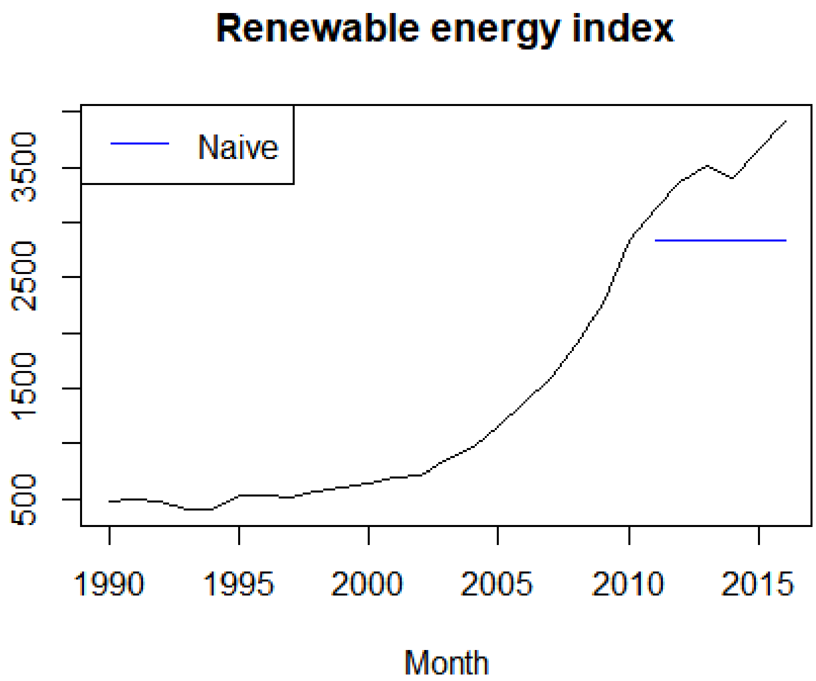


Naive method

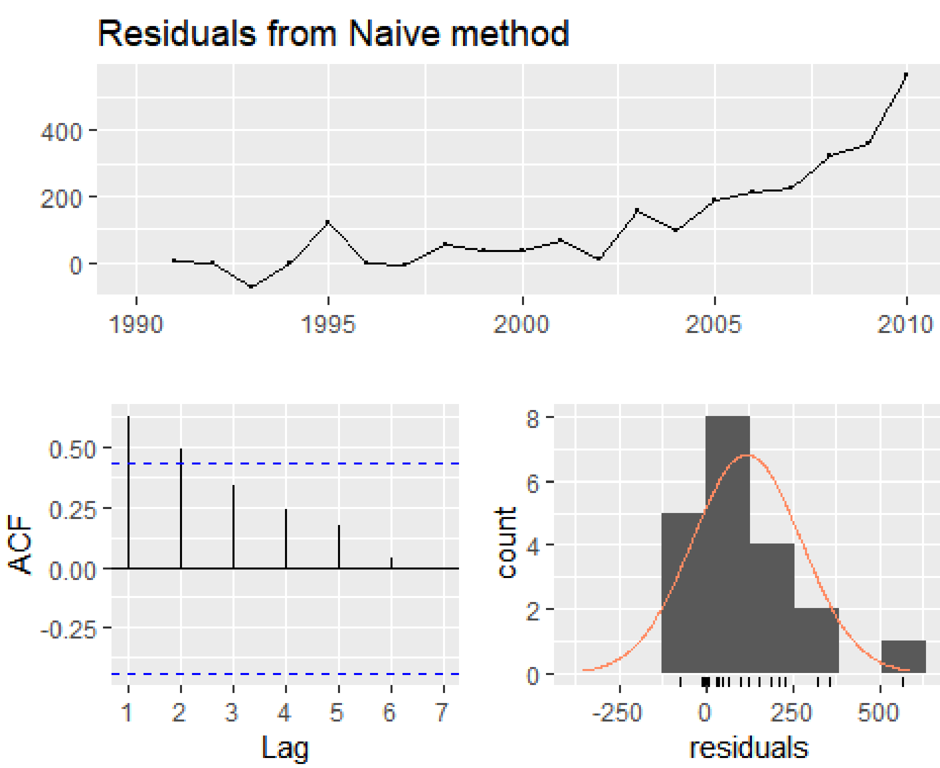
With the plot under we can see that naive predictions does not have a upward trend. However to judge the model performance we have to compare with other models by RMSE, MAE, MAPE and MAsE.

For white noise series, we expect each autocorrelation to be close to zero. Of course, they will not be exactly equal to zero as there is some random variation. For a white noise series, we expect 95% of the spikes in the ACF to lie within the blue dashed lines above. If one or more large spikes are outside these bounds, or if substantially more than 5% of spikes are outside these bounds, then the series is probably not white noise. If Ljung-Box test p-value is above 0.05 means accept as white noise. The residual diagnostics show that after the residuals of this method are not white noise.

```
f1 <-naive(train, h = h)
plot(to,main="Renewable energy index", ylab="",xlab="Month")
lines(f1$mean,col=4)
legend("topleft",lty=1,col=c(4),legend=c("Naive"))
```




```r
res <- residuals(f1)
checkresiduals(f1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Naive method
## Q* = 19.649, df = 4, p-value = 0.0005857
```

```
##
## Model df: 0.   Total lags used: 4
```

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

```
##  lags statistic df      p-value
##     1  9.098684  1 2.557935e-03
##     5 20.559475  5 9.808984e-04
##     9 21.718735  9 9.814429e-03
##    13 32.040088 13 2.369647e-03
##    17 61.908048 17 5.072205e-07
```

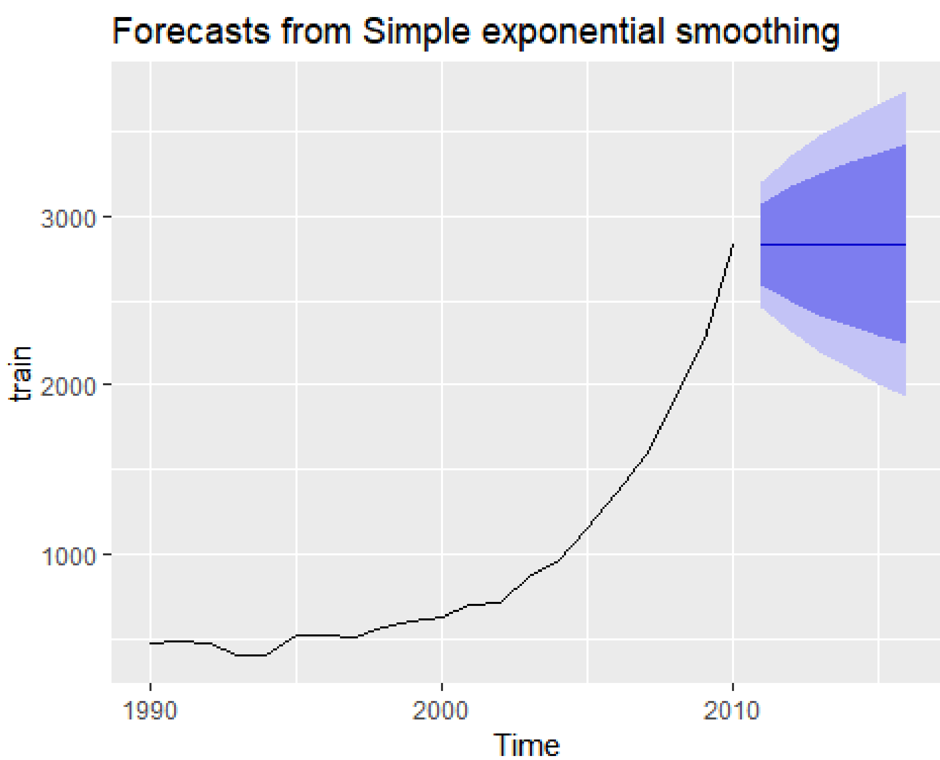
```
accuracy(f1)[,c(2,3,5,6)]
```

```
##      RMSE       MAE      MAPE      MASE
## 192.81377 126.63000  10.14226   1.00000
```

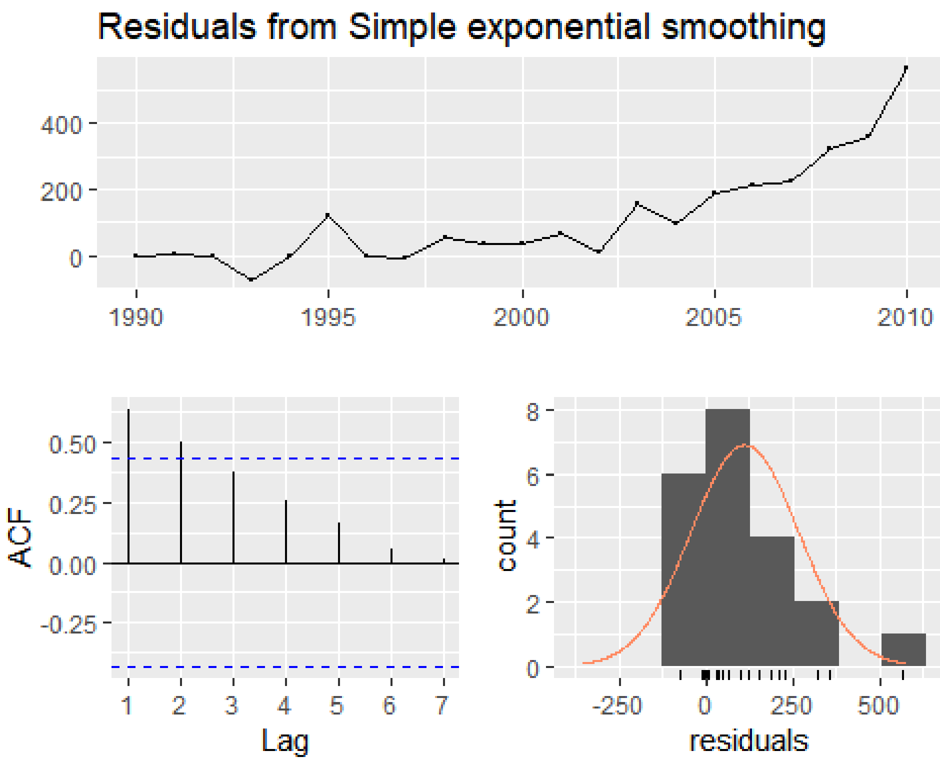
Exponential smoothing method

We will compare to other models to know how accuracy is performing. The residual diagnostics show that the residuals of this method are not white noise.

```
f2 <- ses(train, initial = "simple", h=h)
autoplot(f2)
```




```
res <- residuals(f2)
checkresiduals(f2)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Simple exponential smoothing
## Q* = 22.431, df = 3, p-value = 5.305e-05
##
## Model df: 2.   Total lags used: 5

res <- na.omit(res)
LjungBox(res, lags=seq(1,20,4), order=0)

##  lags statistic df      p-value
##     1  9.639086  1 1.904800e-03
##     5 22.431398  5 4.333999e-04
##     9 23.369324  9 5.418047e-03
##    13 31.902373 13 2.483623e-03
##    17 58.950858 17 1.564037e-06

accuracy(f2)[,c(2,3,5,6)]

##       RMSE        MAE       MAPE       MASE
## 188.166972 120.600000   9.659292   0.952381
```

ETS

ETS (Error, Trend, Seasonal) method is an approach method for forecasting time series. Based on the properties of the data, we estimate several ETS models with a trend and a seasonal component. We consider additive and multiplicative errors, and trends with and without damping. The first letter denotes the error type ("A", "M" or "Z"); the second letter denotes the trend type ("N","A","M" or "Z"); the third letter denotes the season type ("N","A","M" or "Z"). In all cases, "N"=none, "A"=additive, "M"=multiplicative and "Z"=automatically selected.

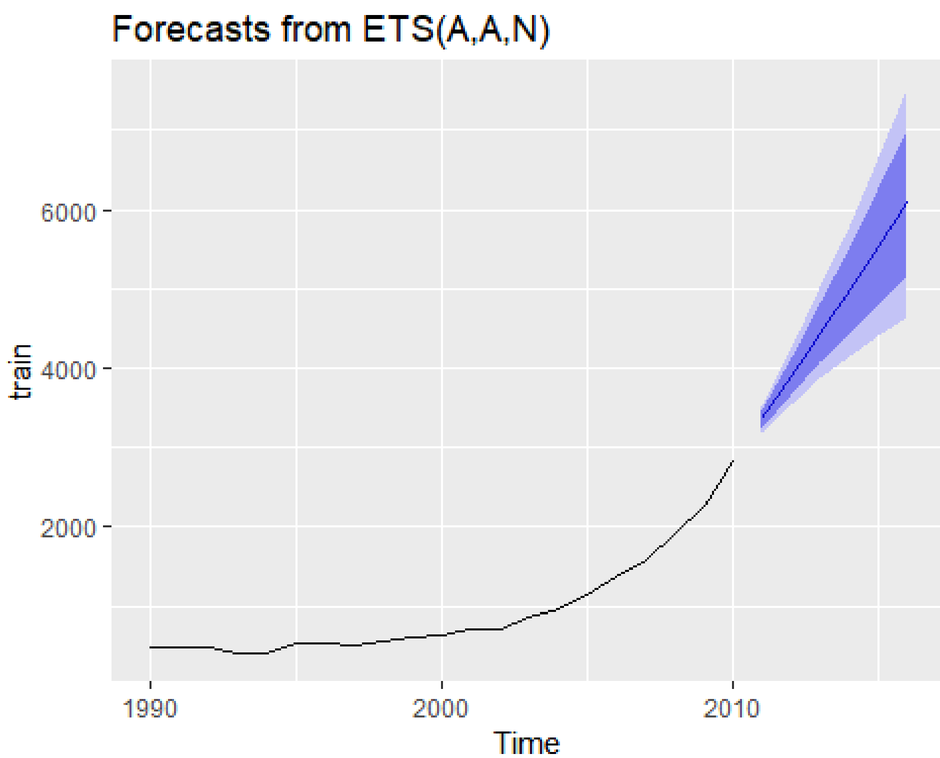
Due to the fact that we do not see a continues pattern of fluctuation. We will try MAN and MMN with damped and non damped, to compare with auto ets which auto ets choose among best AIC, but not accuracy.

ETS AAN model have the best accuracy of MAE and MASE among ETS model for this situation.

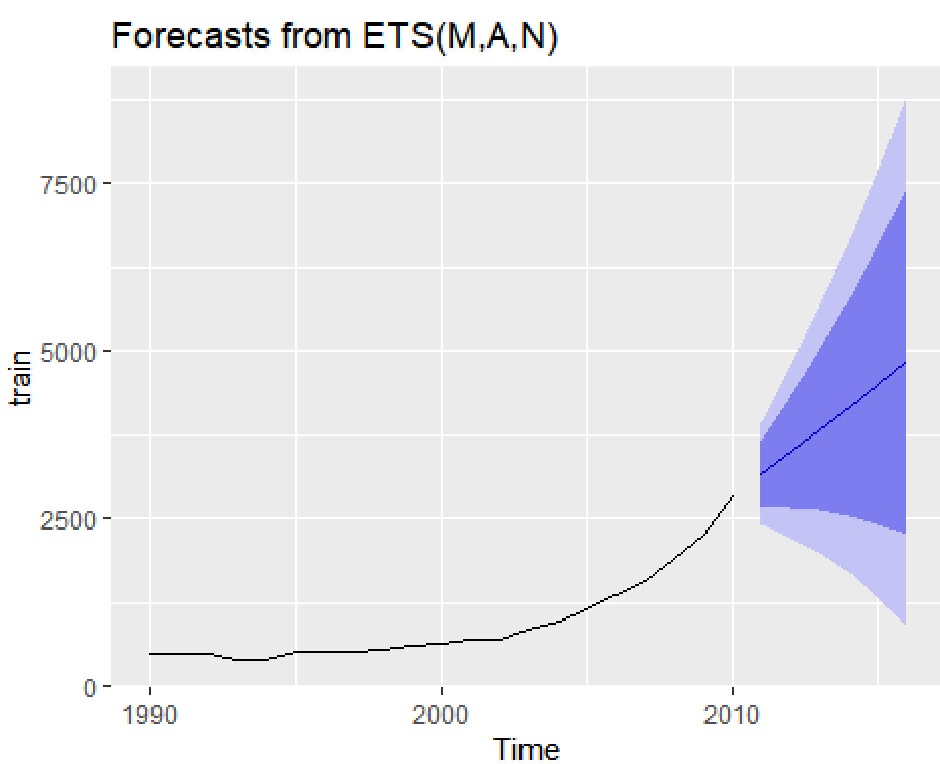
ETS MAN have the best MAPE. ETS MMN have the best RMSE.

The residual diagnostics also has an acceptable result.

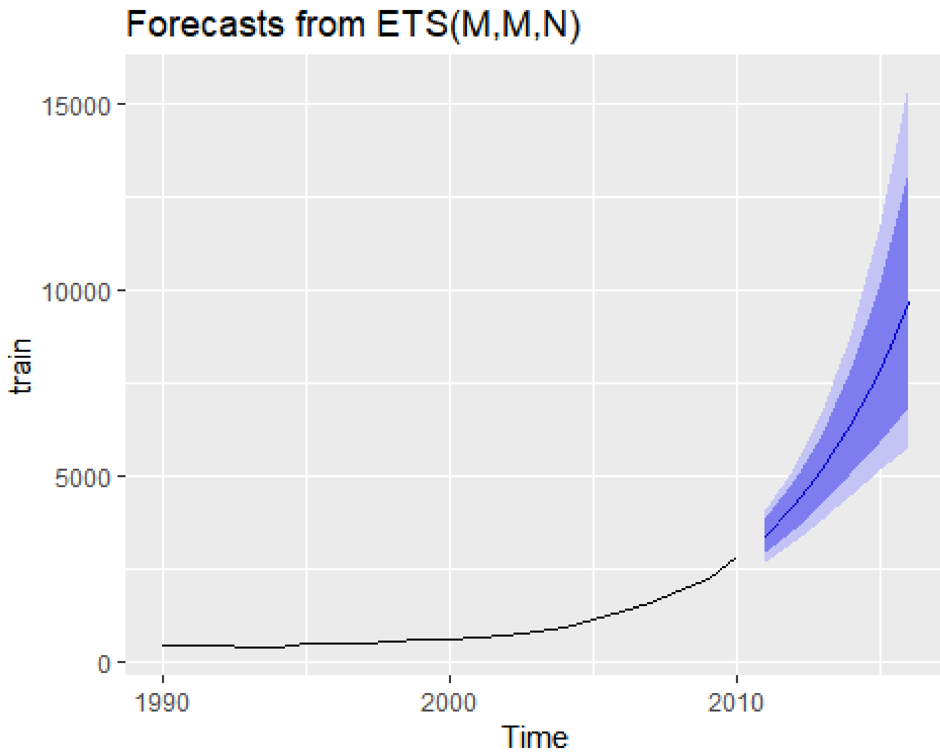
```
f8 <- forecast(ets(train, model = "ZZZ"), method="rwdrift", h=h)
autoplot(f8)
```



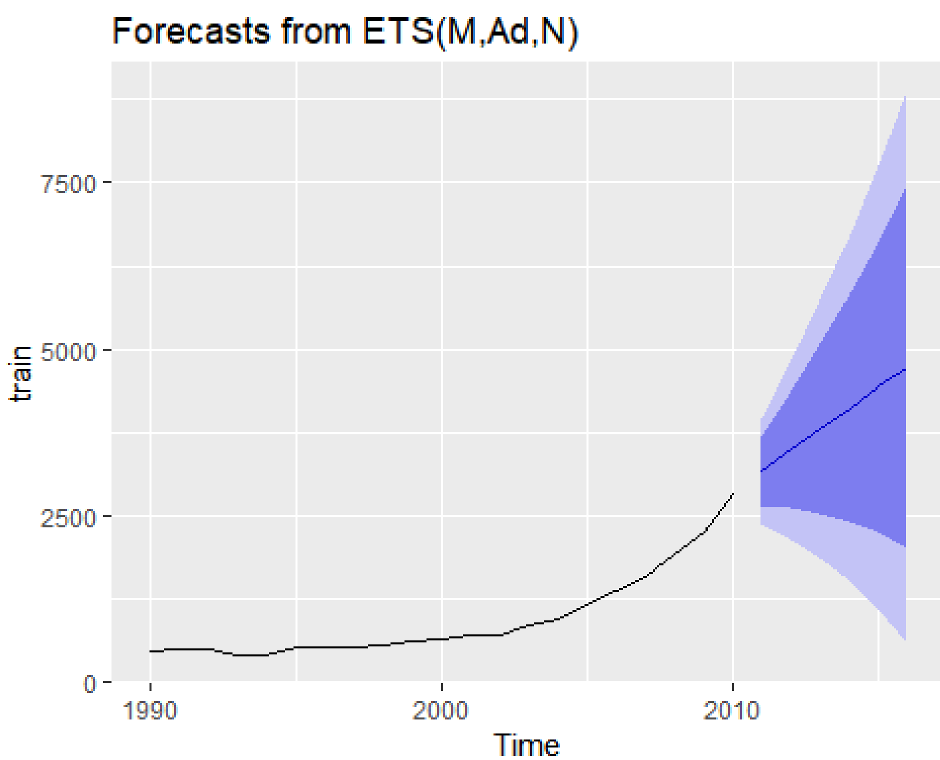

```
f9 <- forecast(ets(train, model = "MAN"), method="rwdrift", h=h)
autoplot(f9)
```



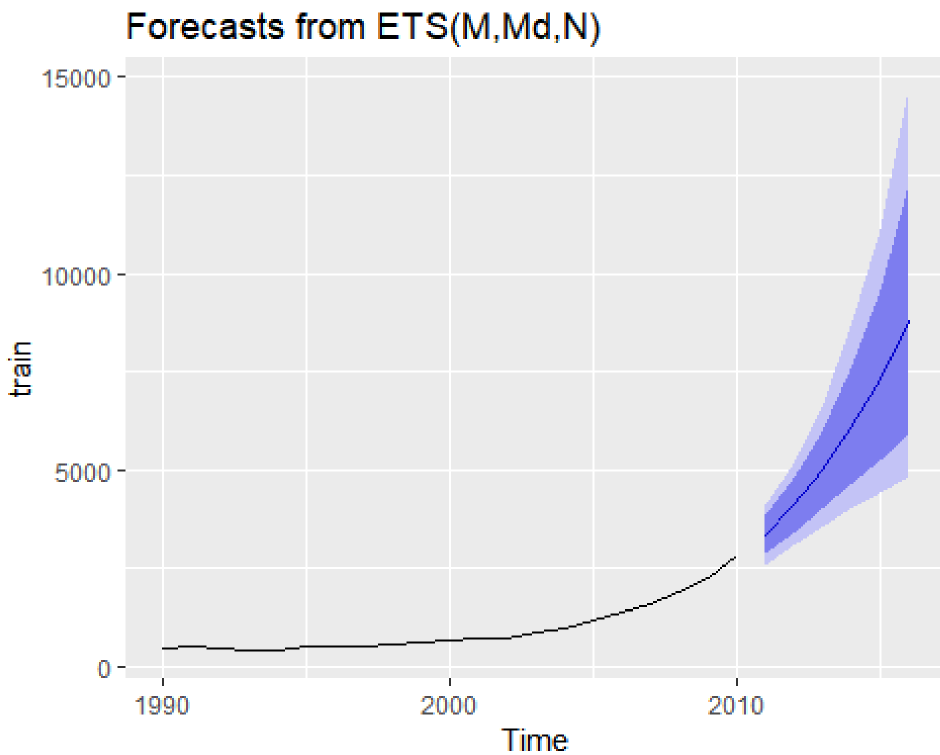
```r
f10 <- forecast(ets(train, model = "MMN"), method="rwdrift", h=h)
autoplot(f10)
```



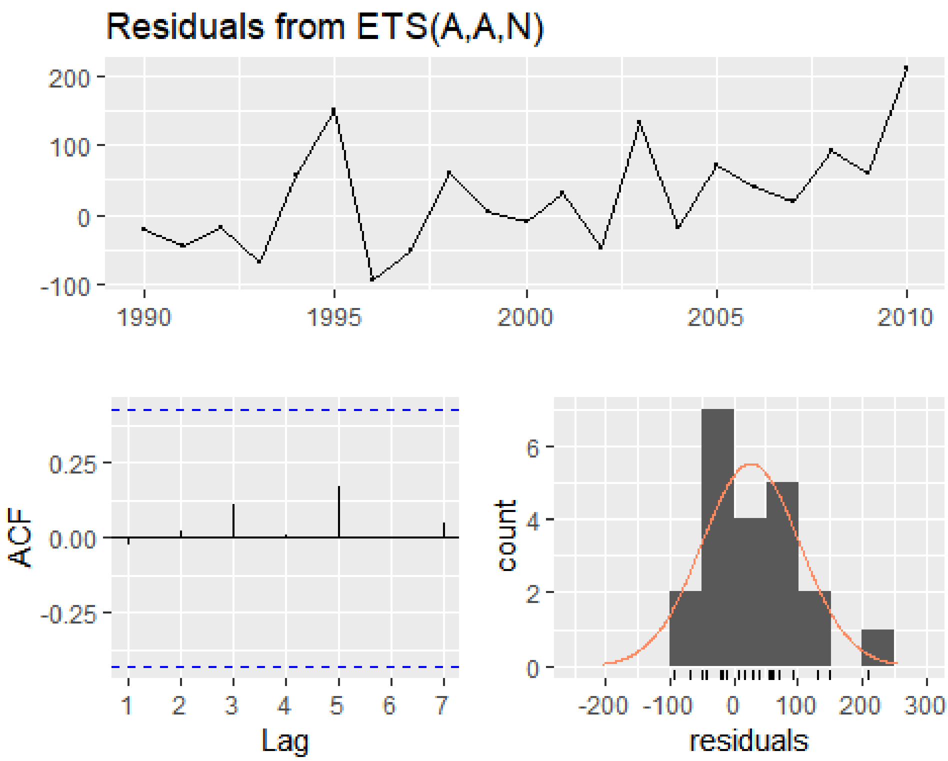

```r
f11 <- forecast(ets(train, model = "MAN",damped=TRUE), method="rwdrift", h=h)
autoplot(f11)
```



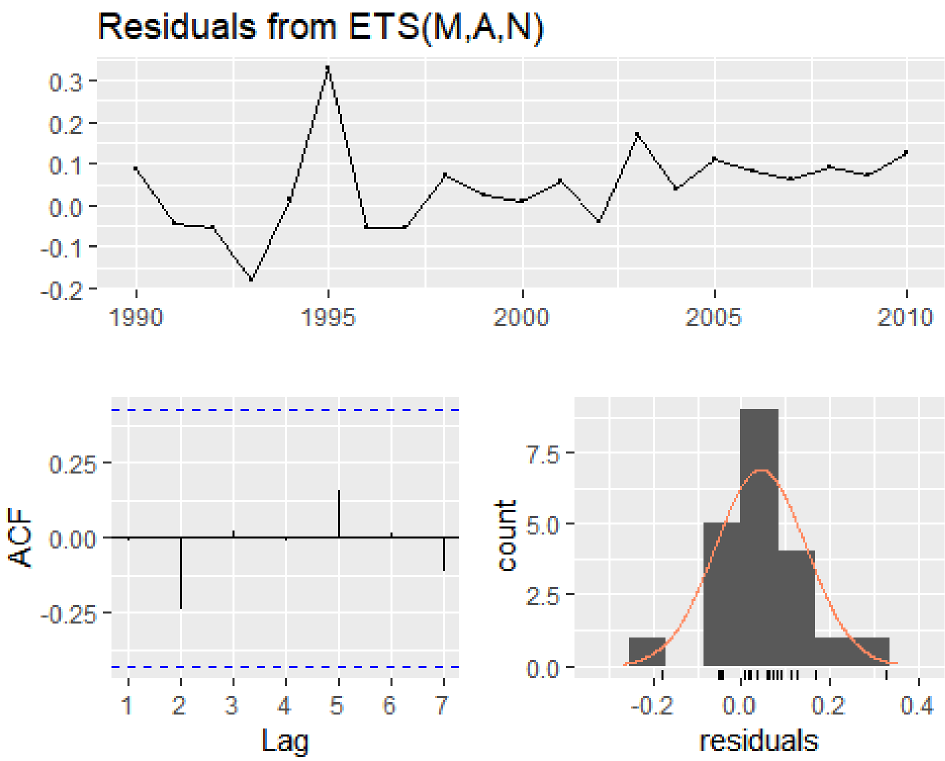

```r
f12 <- forecast(ets(train, model = "MMN",damped=TRUE), method="rwdrift", h=h)
autoplot(f12)
```



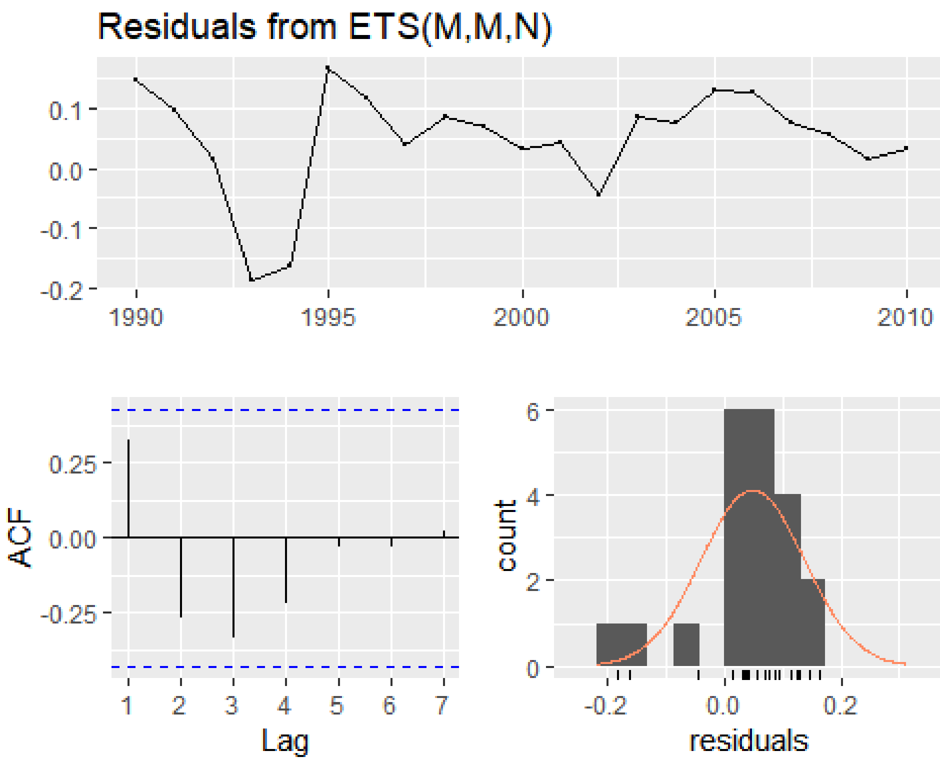

```
a_fc8 <- accuracy(f8)[,c(2,3,5,6)]
a_fc9 <- accuracy(f9)[,c(2,3,5,6)]
a_fc10 <- accuracy(f10)[,c(2,3,5,6)]
a_fc11 <- accuracy(f11)[,c(2,3,5,6)]
a_fc12 <- accuracy(f12)[,c(2,3,5,6)]
acc <- rbind(a_fc8, a_fc9, a_fc10, a_fc11, a_fc12)
rownames(acc) <- c("a_fc8", "a_fc9", "a_fc10", "a_fc11", "a_fc12")
acc

##              RMSE      MAE     MAPE      MASE
## a_fc8   79.11635 61.62783 8.073414 0.4866764
## a_fc9  105.11355 76.03939 7.843354 0.6004848
## a_fc10  74.54841 64.33515 8.255244 0.5080561
## a_fc11 106.15083 76.85439 7.932703 0.6069209
## a_fc12  78.87746 67.12608 8.372686 0.5300962

res <- residuals(f8)
checkresiduals(f8)
```



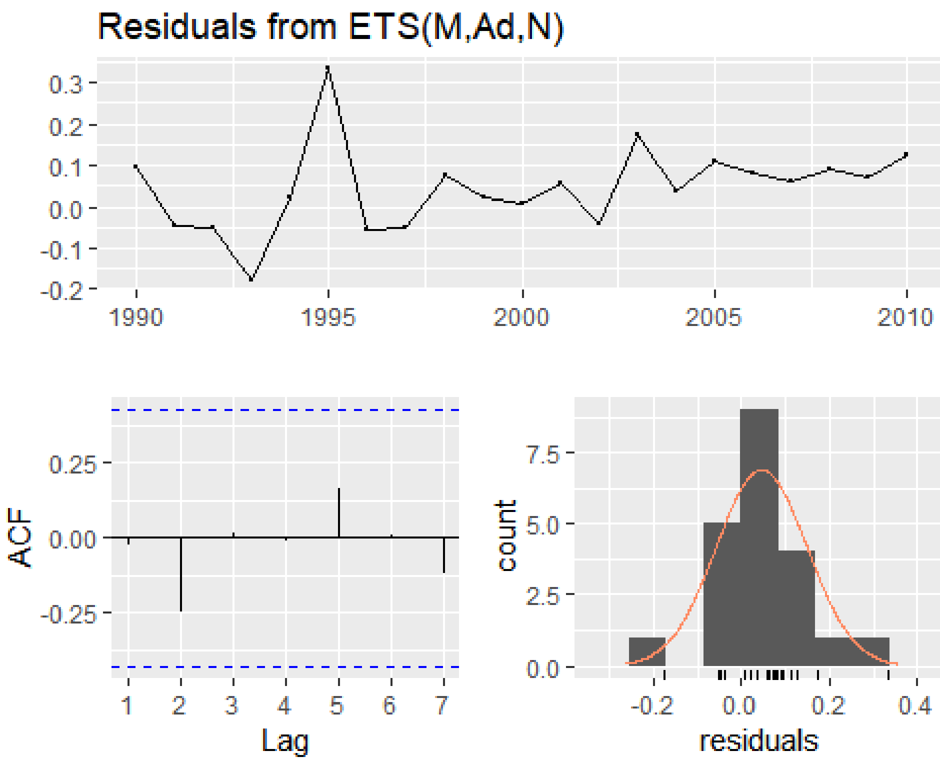

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,N)
## Q* = 1.342, df = 3, p-value = 0.7192
##
## Model df: 4.   Total lags used: 7

res <- na.omit(res)
LjungBox(res, lags=seq(1,20,4), order=0)

##  lags   statistic df   p-value
##     1  0.01652321  1 0.8977196
##     5  1.25153498  5 0.9398388
##     9  1.35077355  9 0.9981098
##    13  3.01246070 13 0.9978890
##    17 10.41692890 17 0.8852228

res <- residuals(f9)
checkresiduals(f9)
```



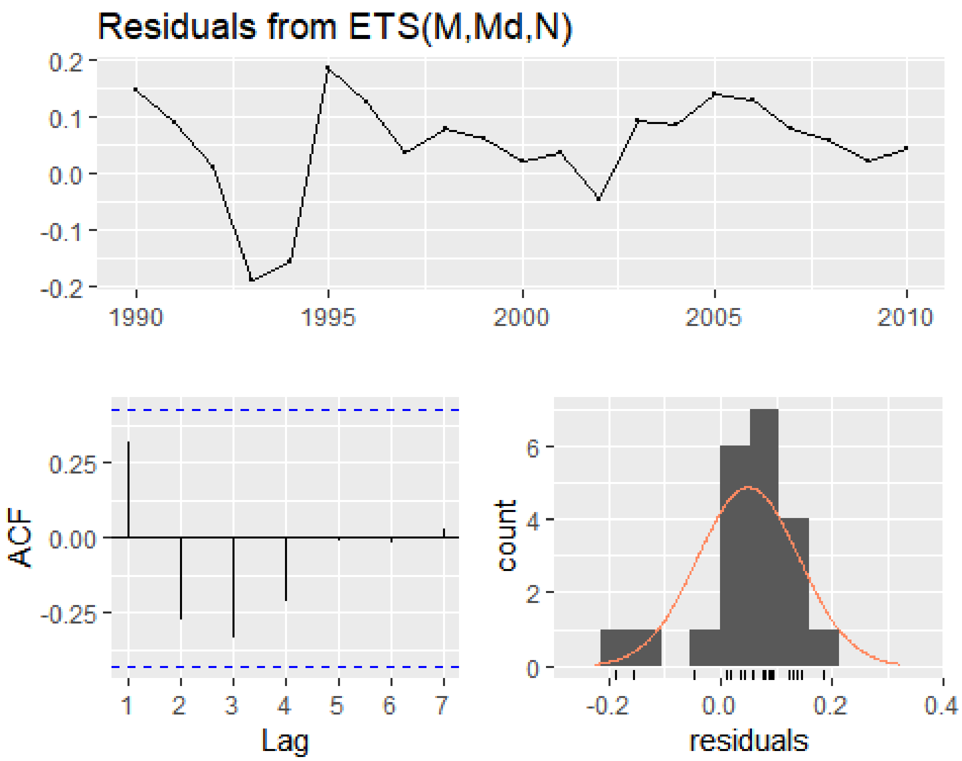

```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,A,N)
## Q* = 2.6619, df = 3, p-value = 0.4467
##
## Model df: 4.   Total lags used: 7

res <- na.omit(res)
LjungBox(res, lags=seq(1,20,4), order=0)

##  lags   statistic df   p-value
##     1 0.002728522  1 0.9583412
##     5 2.221098021  5 0.8177827
##     9 3.358357073  9 0.9483793
##    13 4.644134331 13 0.9822150
##    17 7.205695540 17 0.9807255

res <- residuals(f10)
checkresiduals(f10)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,M,N)
## Q* = 8.8356, df = 3, p-value = 0.03156
##
## Model df: 4.   Total lags used: 7

res <- na.omit(res)
LjungBox(res, lags=seq(1,20,4), order=0)

##  lags statistic df   p-value
##     1  2.593277  1 0.1073181
##     5  8.785843  5 0.1179171
##     9 11.889773  9 0.2195950
##    13 17.410177 13 0.1812251
##    17 19.985079 17 0.2749931

res <- residuals(f11)
checkresiduals(f11)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,Ad,N)
## Q* = 3.589, df = 3, p-value = 0.3094
##
## Model df: 5.   Total lags used: 8

res <- na.omit(res)
LjungBox(res, lags=seq(1,20,4), order=0)

##  lags  statistic df   p-value
##     1 0.01186708  1 0.9132532
##     5 2.41274984  5 0.7895739
##     9 3.60136861  9 0.9356403
##    13 4.85263793 13 0.9782953
##    17 7.27665126 17 0.9796754

res <- residuals(f12)
checkresiduals(f12)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,Md,N)
## Q* = 10.678, df = 3, p-value = 0.0136
##
## Model df: 5.   Total lags used: 8
```

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

```
##  lags statistic df   p-value
##     1  2.498949  1 0.1139223
##     5  8.648340  5 0.1239399
##     9 11.661534  9 0.2330698
##    13 17.183265 13 0.1910611
##    17 19.107289 17 0.3224088
```

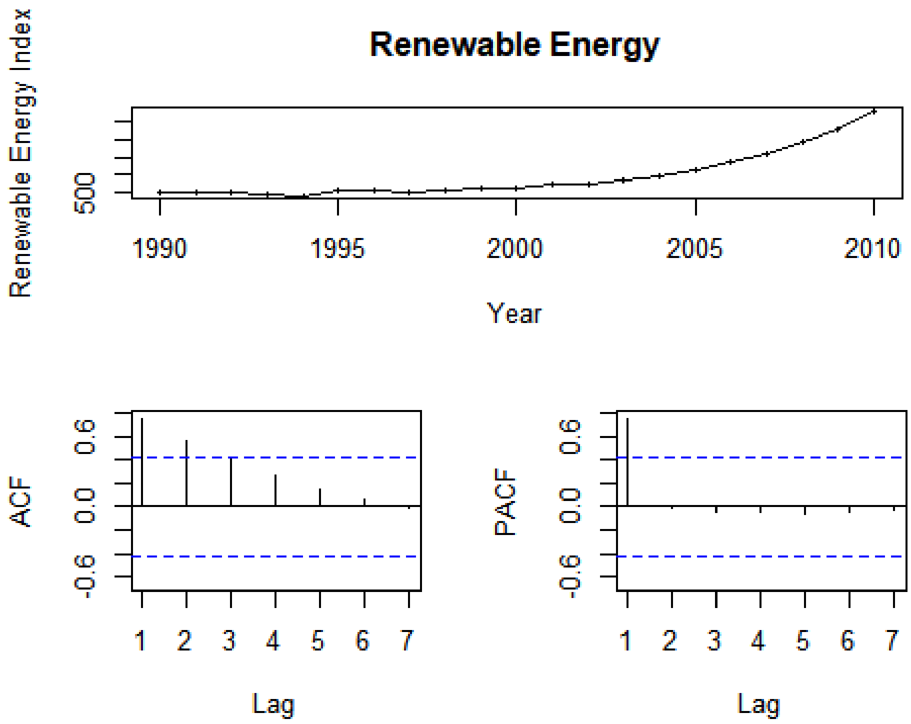
ARIMA

The ACF shows that nonstationarity is mainly caused by trend, and to a lesser extent by the seasonality. The auto.arima procedure results in an ARIMA 020 model. The ACF shows that nonstationarity is mainly caused by trend, and not by the seasonality. This model shows satisfactory diagnostics. We will explore some variations starting from this model, and check model fit and forecast accuracy. The code allows us to gather the results of several models.

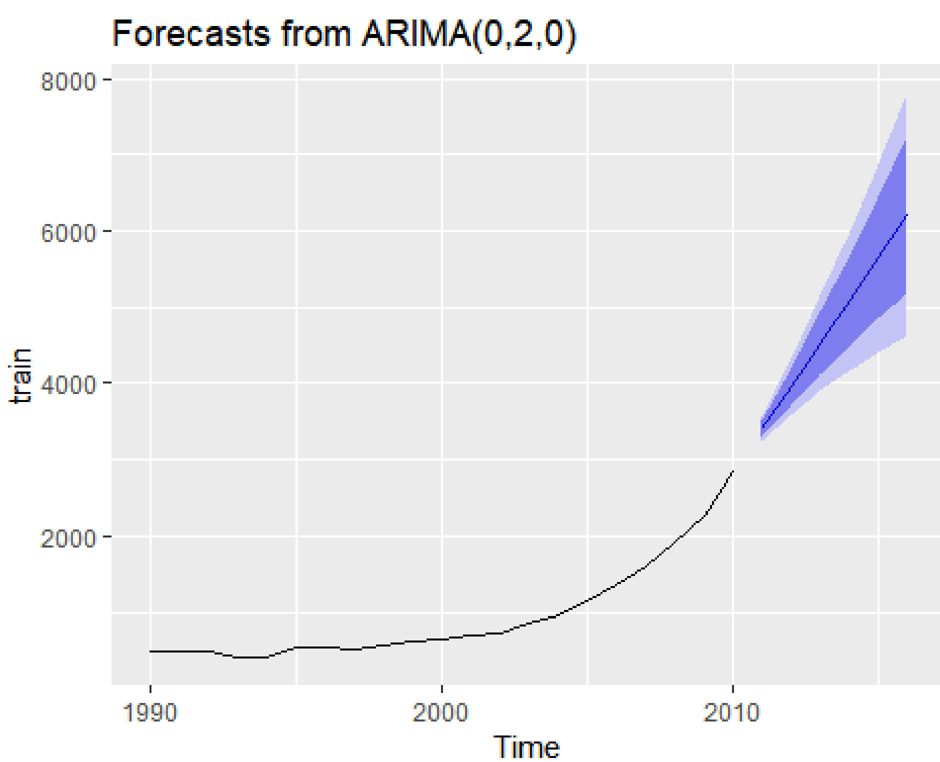
Arima 0 2 0 have the best AIC, where Arima 0 2 16 have the best accuracy.

Although the Arima auto does show an acceptable result of residual diagnostics, but it’s accuracy are not better than Arima 0 2 16 and it also has residual diagnostics results that is acceptable.

```
tsdisplay(train, main="Renewable Energy", ylab="Renewable Energy Index", xlab="Year")
```

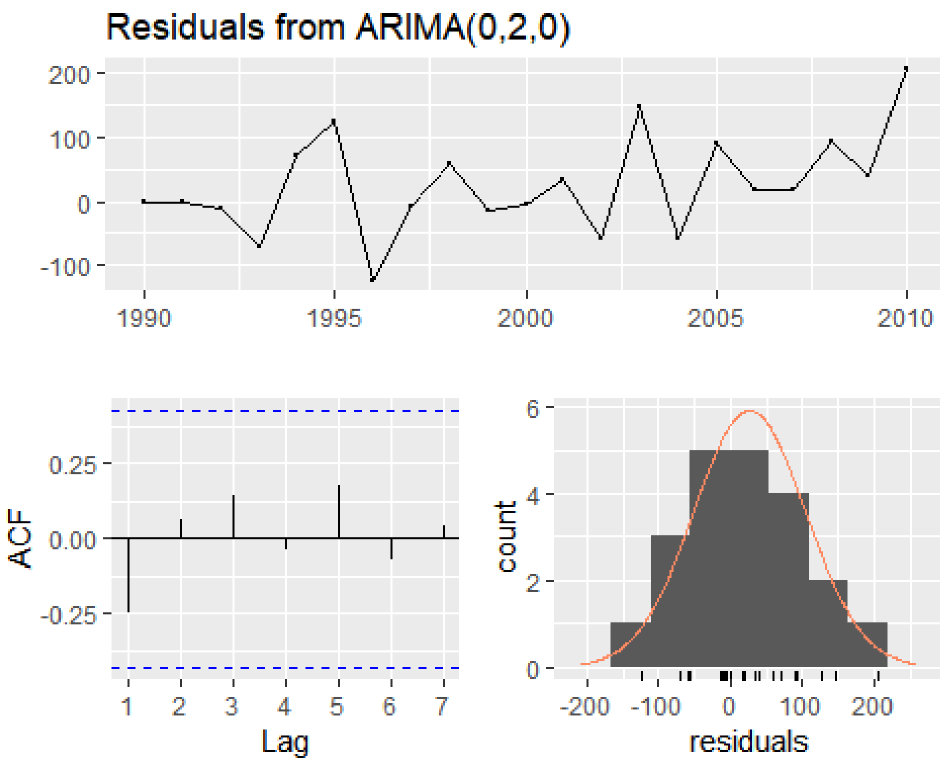



```
ndiffs(train)

## [1] 2

f13 <- forecast(auto.arima(train), h=h)
autoplot(f13)
```

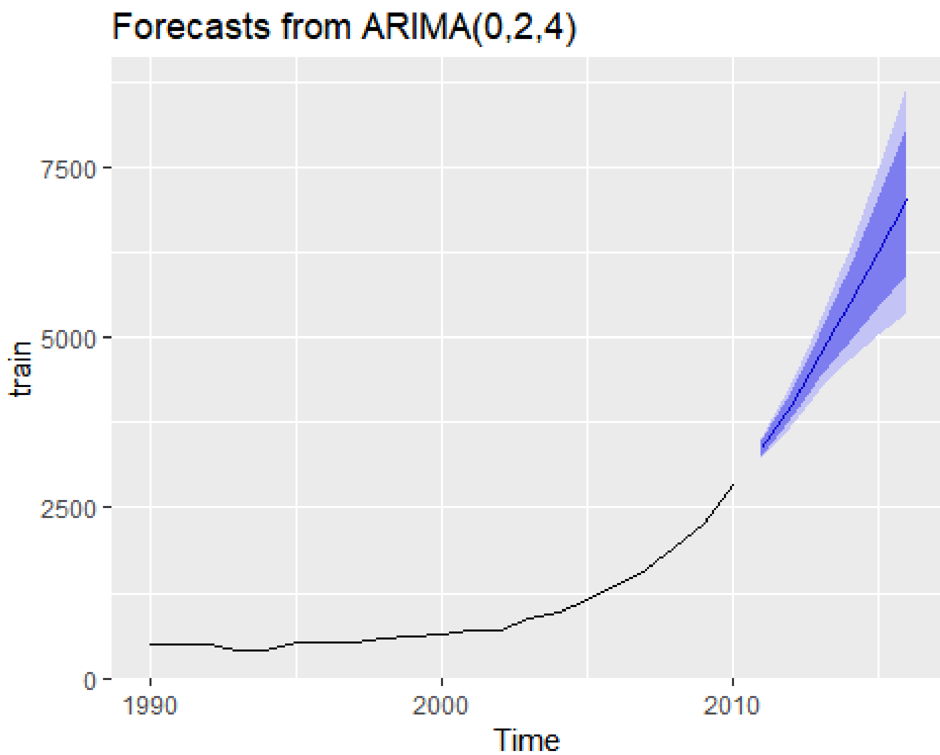



```
accuracy(f13)[,c(2,3,5,6)]
```

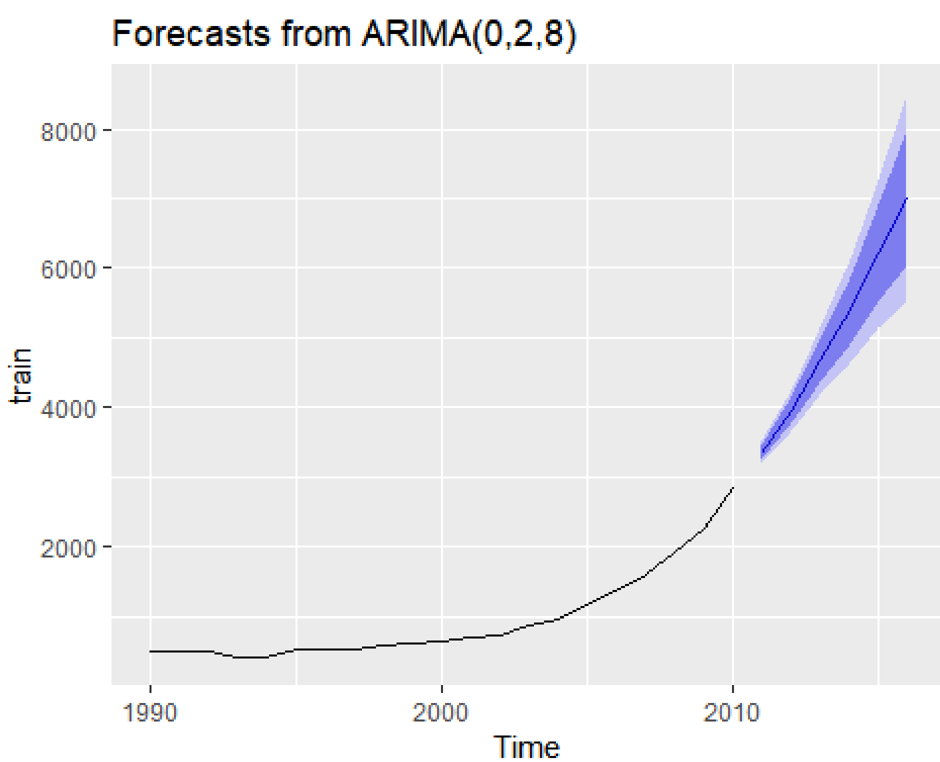
```
##        RMSE         MAE        MAPE        MASE
## 80.6009474 59.5070606  7.5832102  0.4699286

res <- residuals(f13)
checkresiduals(f13)
```



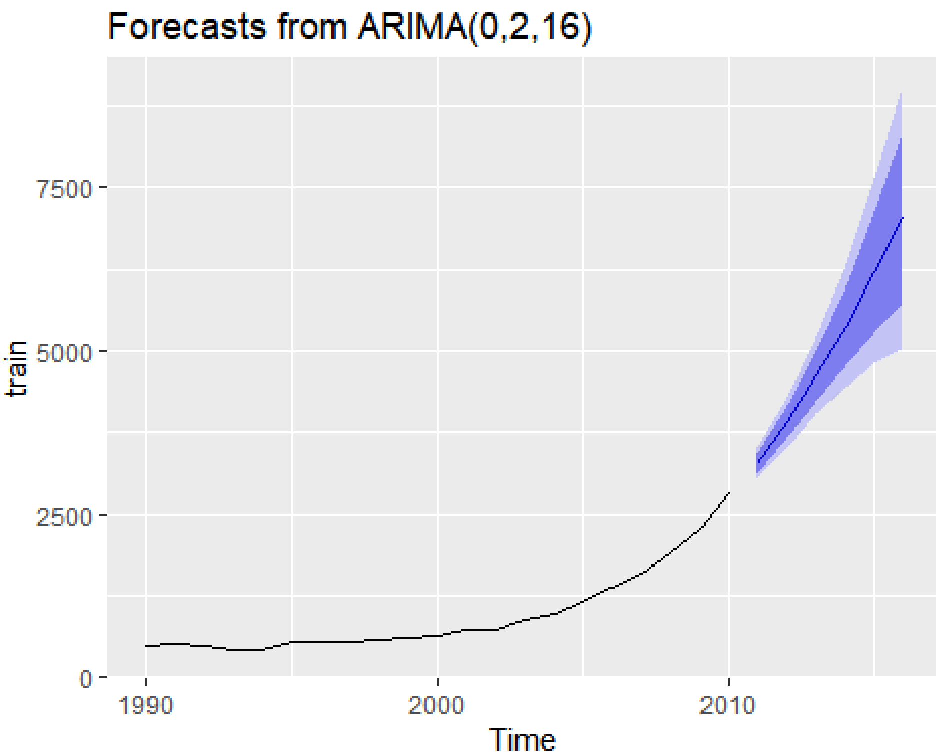

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,2,0)
## Q* = 2.2235, df = 4, p-value = 0.6947
##
## Model df: 0.   Total lags used: 4

res <- na.omit(res)
LjungBox(res, lags=seq(1,20,4), order=0)

##  lags statistic df   p-value
##     1  1.517832  1 0.2179478
##     5  3.153500  5 0.6763342
##     9  3.462382  9 0.9431179
##    13  4.078546 13 0.9903268
##    17 10.037774 17 0.9020187

f14 <- forecast(Arima(train, order=c(0,2,4)), h=h)
autoplot(f14)
```



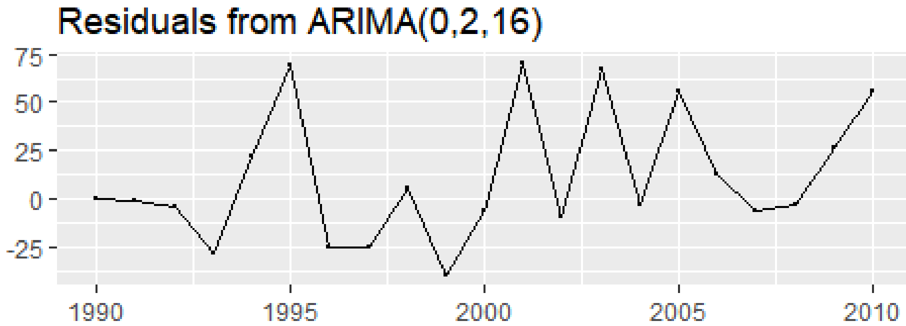

```
f15 <- forecast(Arima(train, order=c(0,2,8)), h=h)
autoplot(f15)
```

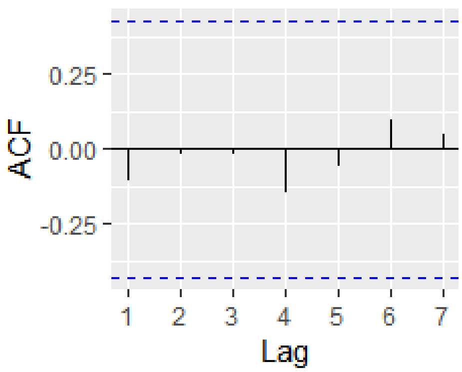


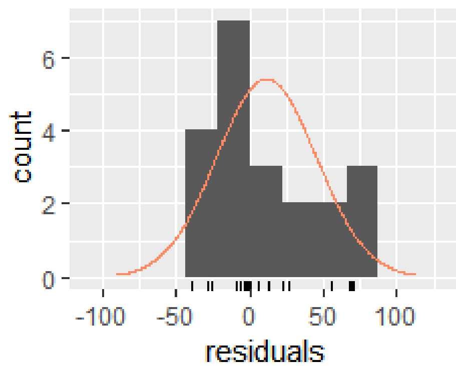

```
f16 <- forecast(Arima(train, order=c(0,2,16)), h=h)
autoplot(f16)
```




```r
accuracy(f13)[,c(2,3,5,6)]

##       RMSE        MAE       MAPE       MASE
## 80.6009474 59.5070606  7.5832102  0.4699286

accuracy(f14)[,c(2,3,5,6)]

##       RMSE        MAE       MAPE       MASE
## 65.8666205 47.9409224  6.1537196  0.3785906

accuracy(f15)[,c(2,3,5,6)]

##       RMSE        MAE       MAPE       MASE
## 52.7089396 39.5434752  5.2606424  0.3122757

accuracy(f16)[,c(2,3,5,6)]

##       RMSE        MAE       MAPE       MASE
## 34.9246080 25.4468864  3.4926272  0.2009546

res <- residuals(f16)
checkresiduals(f16)
```








```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,2,16)
## Q* = 7.7479, df = 3, p-value = 0.05152
##
## Model df: 16.   Total lags used: 19

res <- na.omit(res)
LjungBox(res, lags=seq(1,20,4), order=0)

##  lags statistic df   p-value
##     1 0.2631615  1 0.6079564
##     5 0.9702448  5 0.9649299
##     9 1.8405063  9 0.9937459
##    13 5.2743088 13 0.9686477
##    17 7.3853865 17 0.9779878
```

Conclusion

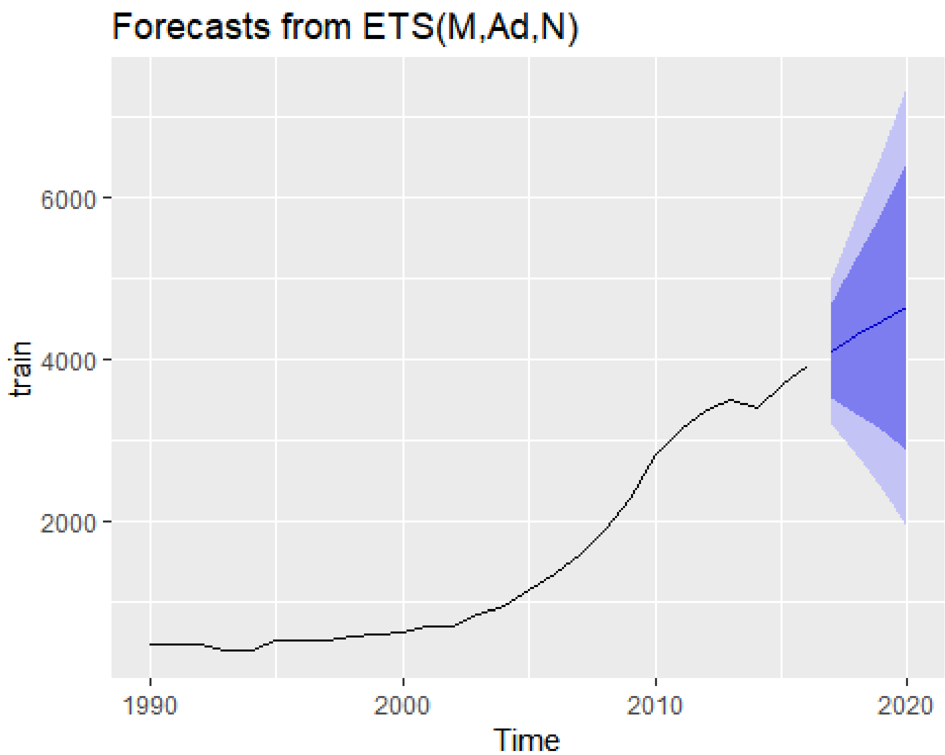
The Arima model 0 2 16 have been out perform other than all other models on training set. But on the test we can see that ETS MAN d test have the best performance of RMSE, MAE, MAPE and MASE. The residual diagnostics results of ETS MAN d test is also acceptable. Therefore we will use ETS MAN d as final to do forecast to 2020.

```
af1 = accuracy(f1, test)
af2 = accuracy(f2, test)
af8 = accuracy(f8, test)
af9 = accuracy(f9, test)
af10 = accuracy(f10, test)
af11 = accuracy(f11, test)
af12 = accuracy(f12, test)
af13 = accuracy(f13, test)
af14 = accuracy(f14, test)
af15 = accuracy(f15, test)
```

```
af16 = accuracy(f16, test)

a.table <- rbind(af1, af2, af8, af9, af10, af11, af12, af13, af14, af15, af16)
row.names(a.table)<-c("S. Naive training", 'S. Naive test',
                      'STL training', 'STL test',
                      'ETS auto training', 'ETS auto test',
                      'ETS MAN training', 'ETS MAN test',
                      'ETS MMN training', 'ETS MMN test',
                      'ETS MAN d training', 'ETS MAN d test',
                      'ETS MMN d training', 'ETS MMN d test',
                      'ARIMA Auto training', 'ARIMA Auto test',
                      'ARIMA 024 training', 'ARIMA 024 test',
                      'ARIMA 028 training', 'ARIMA 028 test',
                      'ARIMA 0216 training', 'ARIMA 0216 test')
a.table <- as.data.frame(a.table)
print(kable(a.table, caption="Forecast accuracy",digits = 2 ))

##
##
## Table: Forecast accuracy
##
##                                ME       RMSE        MAE       MPE     MAPE     MASE     ACF1   Theil's U
## --------------------  ---------  --------  --------  -------  ------  ------  ------  ----------
## S. Naive training        117.57    192.81    126.63     7.99   10.14    1.00    0.63          NA
## S. Naive test            662.12    707.48    662.12    18.54   18.54    5.23    0.27        3.51
## STL training             111.97    188.17    120.60     7.61    9.66    0.95    0.63          NA
## STL test                 662.12    707.48    662.12    18.54   18.54    5.23    0.27        3.51
## ETS auto training         26.10     79.12     61.63     1.23    8.07    0.49   -0.03          NA
## ETS auto test          -1220.55   1409.31   1220.55   -33.84   33.84    9.64    0.54        6.99
## ETS MAN training          54.18    105.11     76.04     3.28    7.84    0.60    0.38          NA
## ETS MAN test            -512.51    621.76    512.51   -14.16   14.16    4.05    0.57        3.08
## ETS MMN training          44.81     74.55     64.34     3.80    8.26    0.51    0.54          NA
## ETS MMN test           -2617.32   3243.62   2617.32   -71.71   71.71   20.67    0.50       15.97
## ETS MAN d training        55.37    106.15     76.85     3.45    7.93    0.61    0.38          NA
## ETS MAN d test          -465.63    563.02    465.63   -12.88   12.88    3.68    0.57        2.80
## ETS MMN d training        47.69     78.88     67.13     3.94    8.37    0.53    0.54          NA
## ETS MMN d test         -2276.46   2802.95   2276.46   -62.43   62.43   17.98    0.51       13.82
## ARIMA Auto training       26.59     80.60     59.51     1.74    7.58    0.47   -0.25          NA
## ARIMA Auto test        -1313.28   1505.78   1313.28   -36.44   36.44   10.37    0.54        7.47
## ARIMA 024 training        21.98     65.87     47.94     1.40    6.15    0.38   -0.06          NA
## ARIMA 024 test         -1647.86   1941.50   1647.86   -45.52   45.52   13.01    0.54        9.62
## ARIMA 028 training        17.59     52.71     39.54     1.23    5.26    0.31   -0.12          NA
## ARIMA 028 test         -1601.23   1904.54   1601.23   -44.16   44.16   12.64    0.54        9.43
## ARIMA 0216 training       11.21     34.92     25.45     0.91    3.49    0.20   -0.10          NA
## ARIMA 0216 test        -1578.71   1905.12   1578.71   -43.45   43.45   12.47    0.54        9.43
```

Final model

```
train <- window(to, start=c(1990))

f <- forecast(ets(train, model = "MAN",damped=TRUE), method="rwdrift", h=4)
autoplot(f)
```




```
f$mean

## Time Series:
## Start = 2017
## End = 2020
## Frequency = 1
## [1] 4105.536 4291.387 4473.522 4652.013
```