Assigment 3

Rizny Mubarak

Contents

1. Load data and relevant packages 1
2. in- and out-of-sample and data exploration 2
3. Forecasting 10
    3.1 Selection of forecasts using information criteria 10
    3.2 Selection of forecasts using a validation set 11
4. Out-of-sample evaluation 15
5. Forecast combination with AIC weights 17

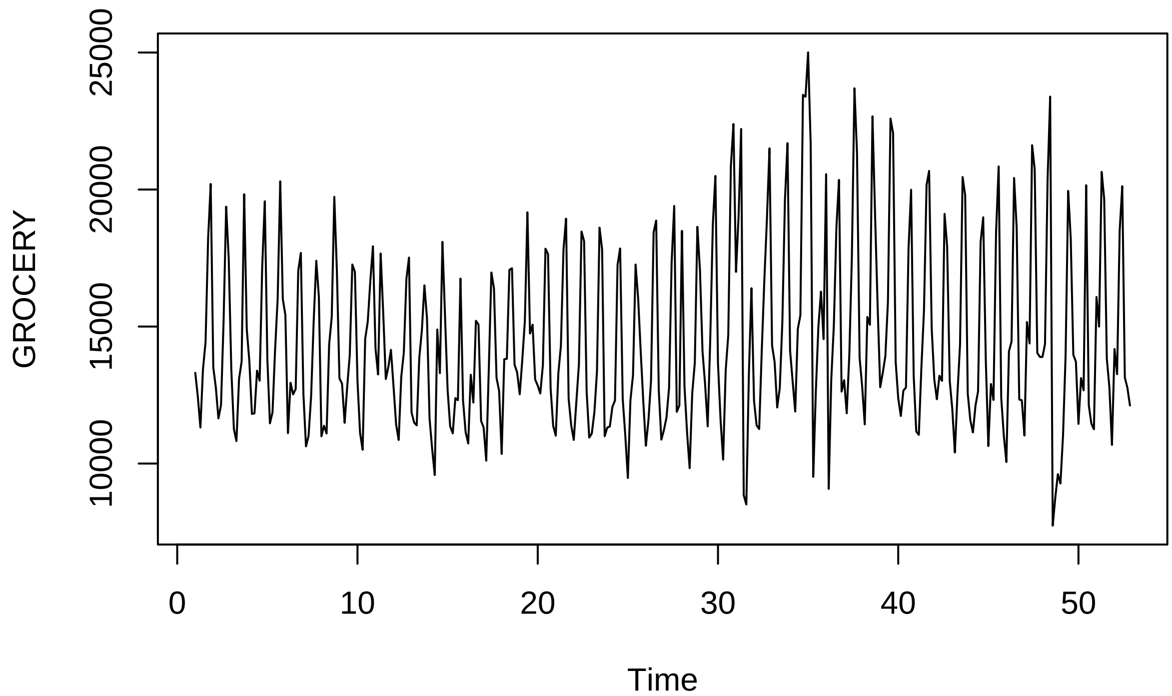
Exercise **19**

Question 1 19
1. in- and out-of-sample and data exploration 19
2. Forecasting 20
    2.1 Selection of forecasts using information criteria 20
    2.2 Selection of forecasts using a validation set 21
3. Out-of-sample evaluation 24
Question 2 26
1. in- and out-of-sample and data exploration 26
2. Forecasting 26
    2.1 Rolling origin testing 27

1. Load data and relevant packages

```
load("./grocery.Rdata")
```

```
plot(y)
```




```
n <- length(y)
n
```

```
## [1] 364
```

```
library(forecast)
```

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

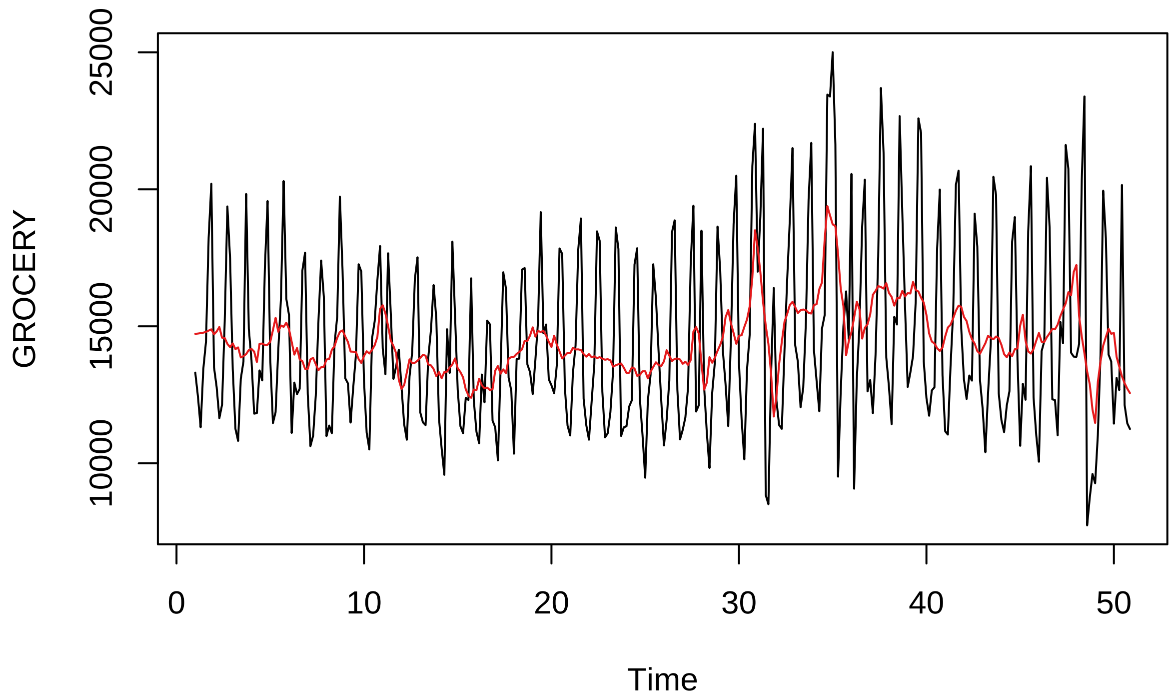
```
library(tsutils)
```

```
## Warning: package 'tsutils' was built under R version 4.2.3
```

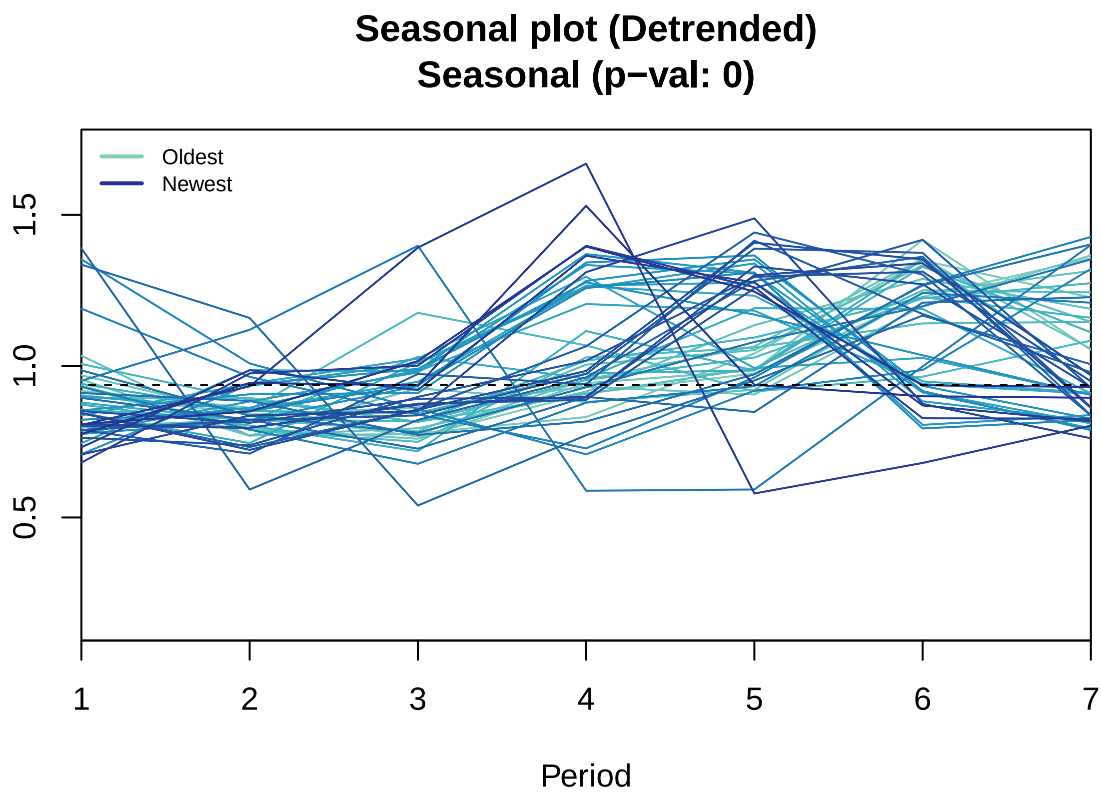
2. in- and out-of-sample and data exploration

```
y.trn <- head(y,7*50)
y.tst <- tail(y,7*2)
```

```r
cma <- cmav(y.trn, outplot=TRUE)
```

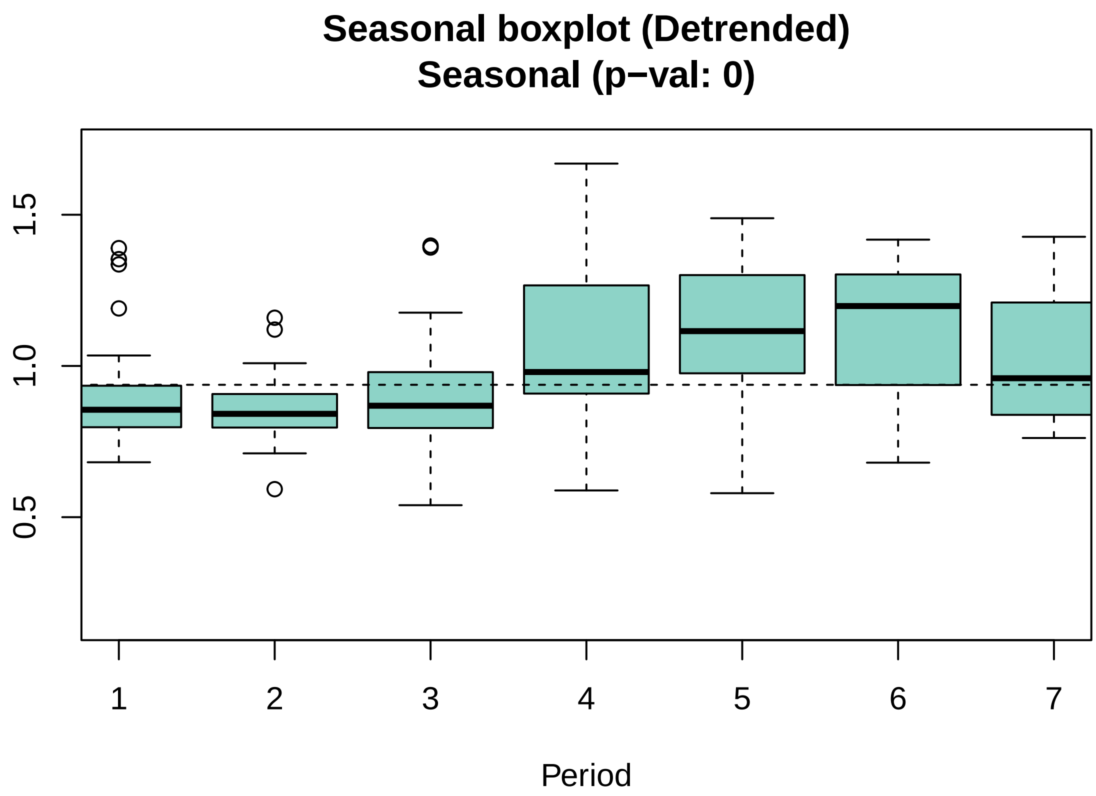



```r
seasplot(y.trn)
```



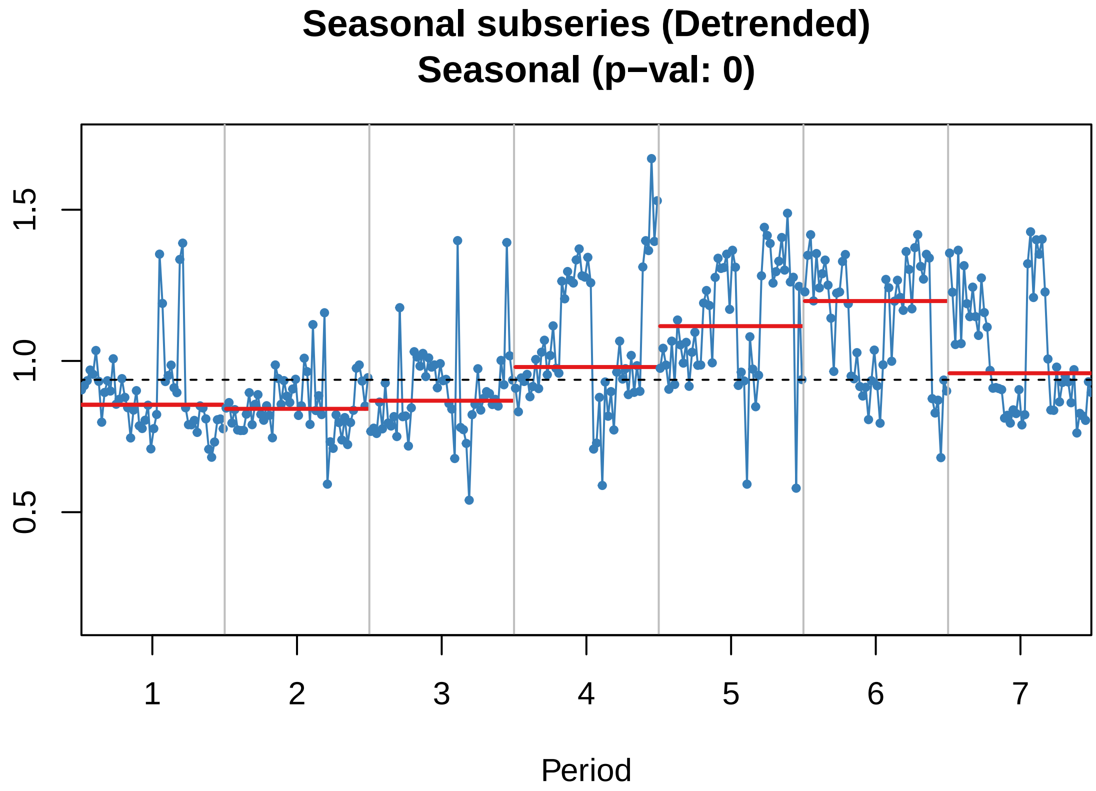

```
## Results of statistical testing
## Evidence of trend: TRUE  (pval: 0)
## Evidence of seasonality: TRUE  (pval: 0)
```

```
seasplot(y.trn,outplot=2)
```



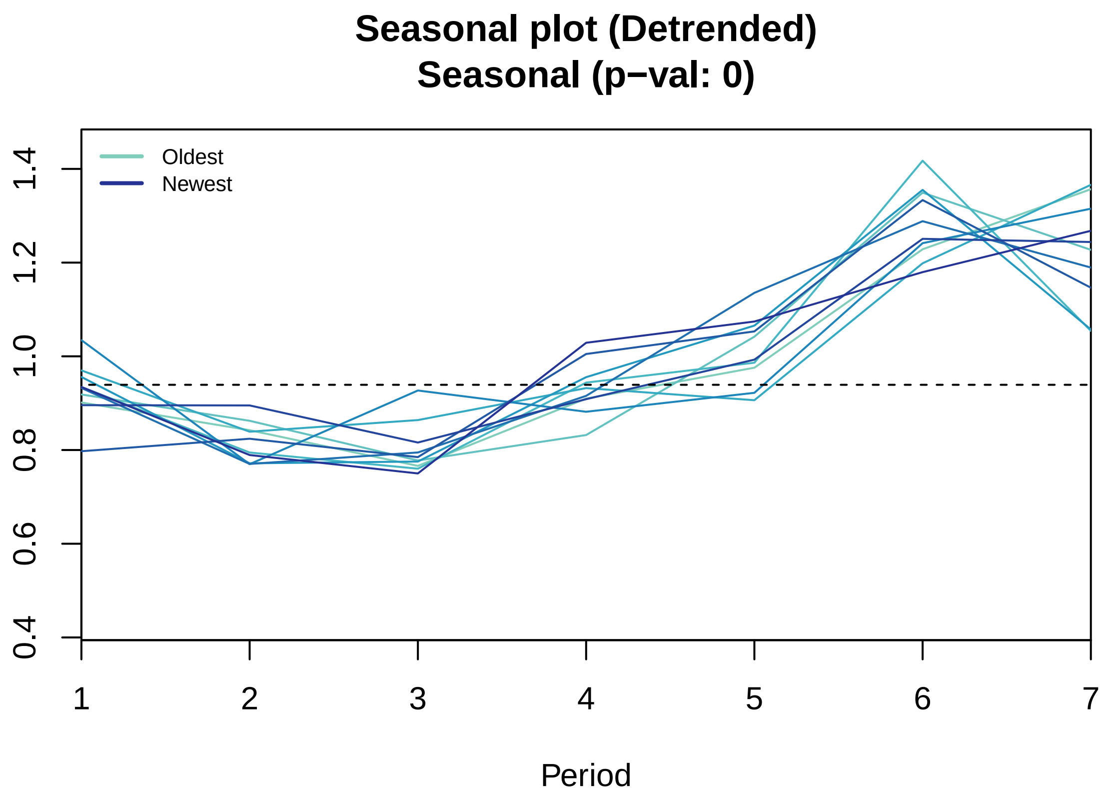

```
## Results of statistical testing
## Evidence of trend: TRUE  (pval: 0)
## Evidence of seasonality: TRUE  (pval: 0)
```

```
seasplot(y.trn,outplot=3)
```



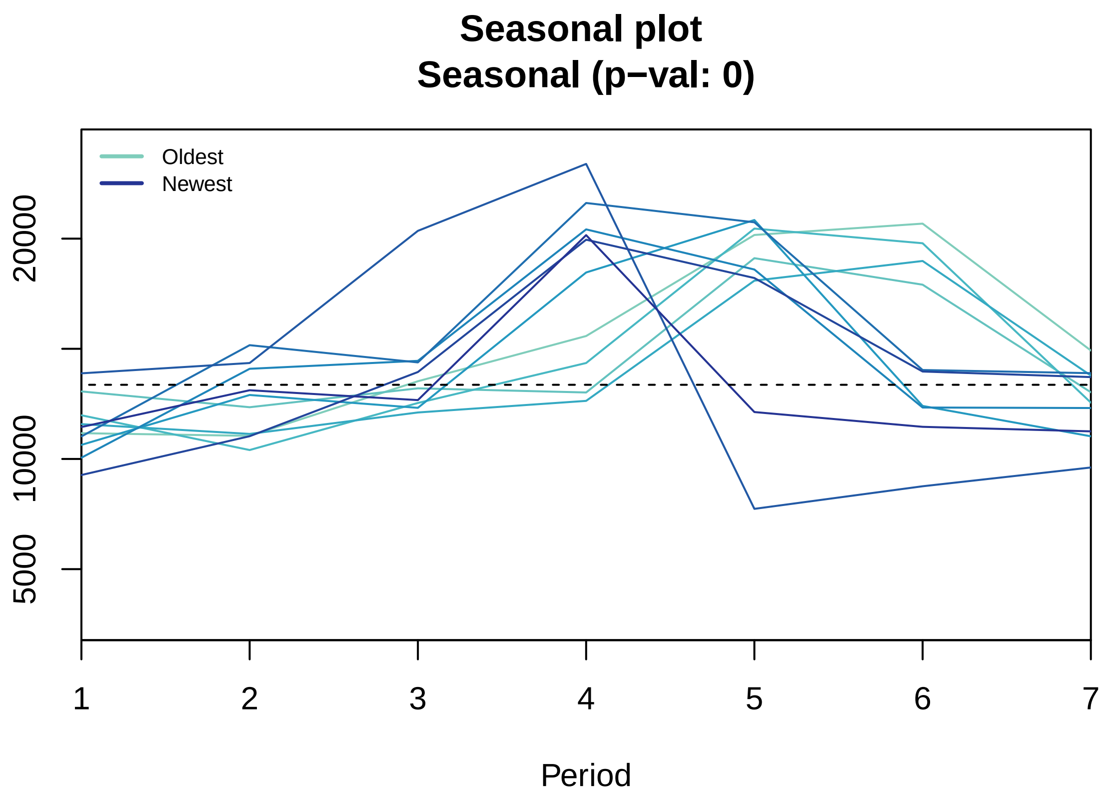

```
## Results of statistical testing
## Evidence of trend: TRUE  (pval: 0)
## Evidence of seasonality: TRUE  (pval: 0)
```

```
seasplot(head(y.trn,10*7))
```



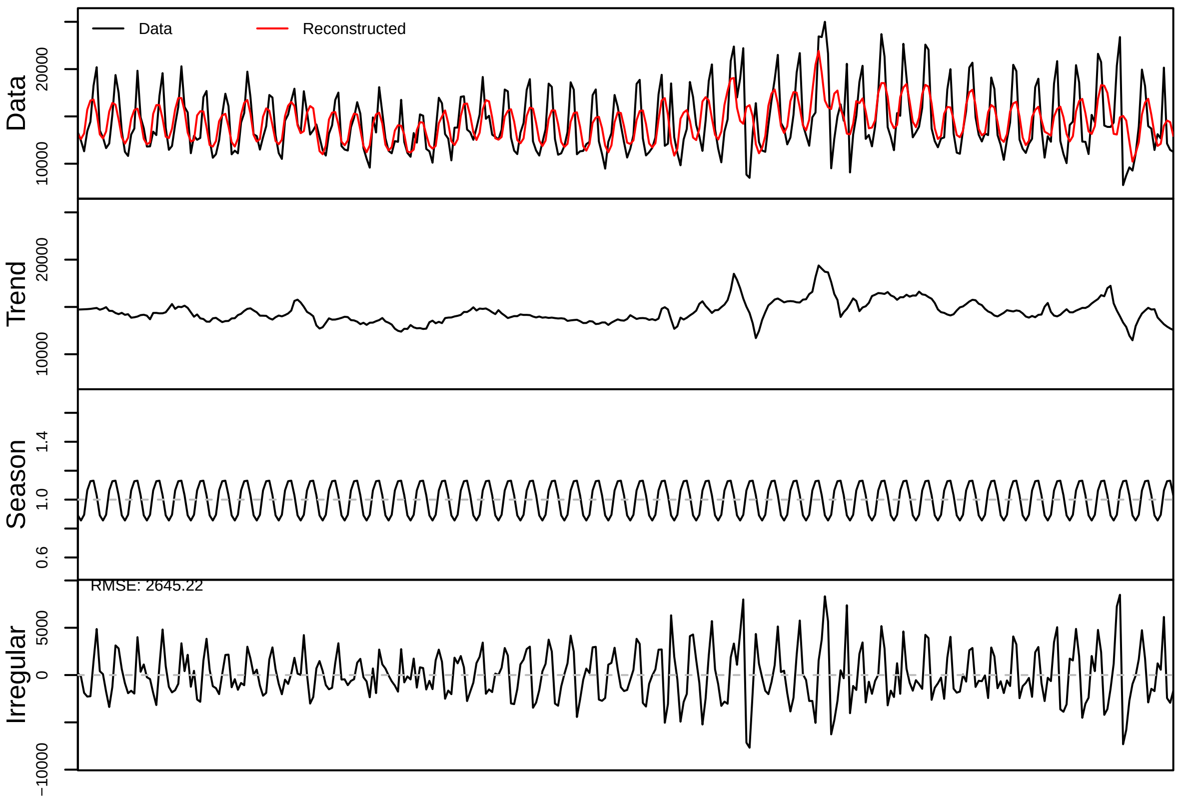

```
## Results of statistical testing
## Evidence of trend: TRUE  (pval: 0.001)
## Evidence of seasonality: TRUE  (pval: 0)
```

```
seasplot(tail(y.trn,10*7))
```

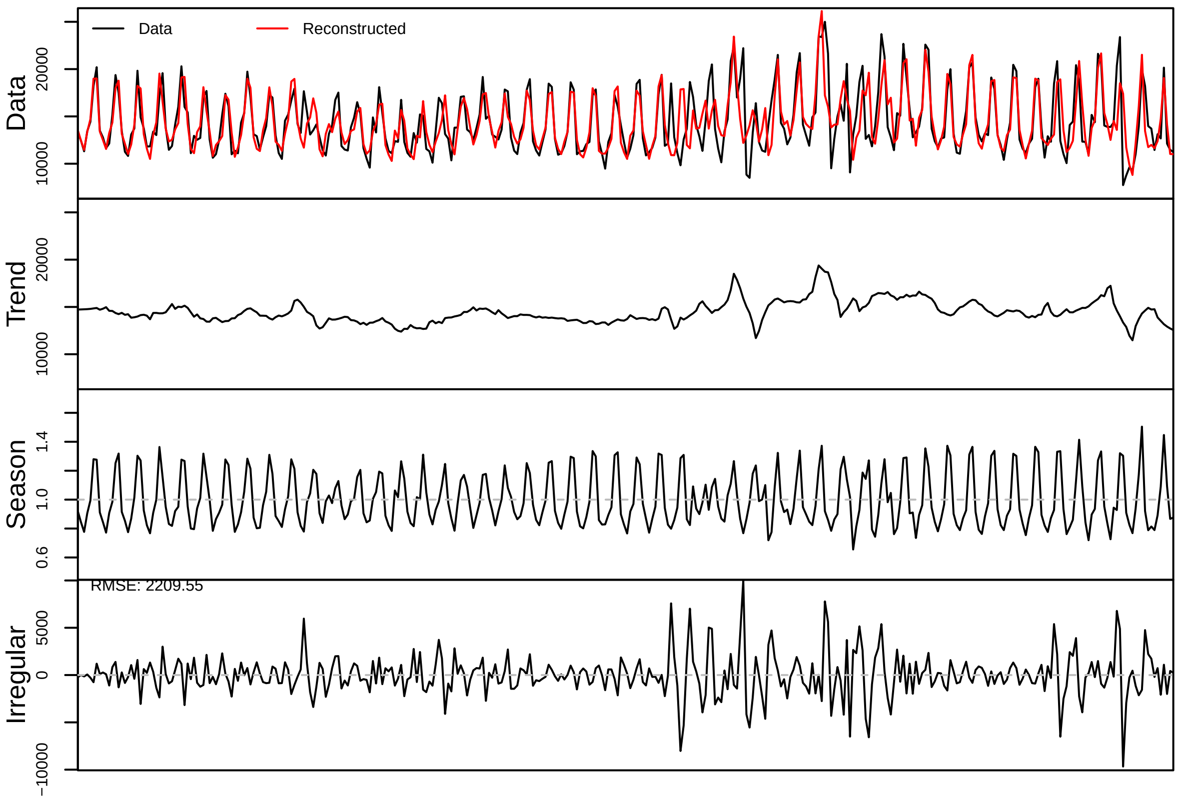



```
## Results of statistical testing
## Evidence of trend: FALSE  (pval: 0.045)
## Evidence of seasonality: TRUE  (pval: 0)
```

```
dc <- decomp(y.trn, outplot=TRUE)
```




```
dc <- decomp(y.trn, outplot=TRUE, type="pure.seasonal")
```



3. Forecasting

3.1 Selection of forecasts using information criteria

```r
fit <- ets(y.trn)
fit
```

```
## ETS(M,N,M)
##
## Call:
##  ets(y = y.trn)
##
##   Smoothing parameters:
##     alpha = 0.1024
##     gamma = 0.3636
##
##   Initial states:
##     l = 14657.0119
##     s = 1.2821 1.2869 0.993 0.8979 0.7854 0.8323
##            0.9224
##
##   sigma:  0.1747
##
##      AIC     AICc      BIC
```

```
## 7537.494 7538.142 7576.073
```

```r
# Level model
fit1 <- ets(y.trn,model="ANN")
# Seasonal model
fit2 <- ets(y.trn,model="ANA")
# Linear trend model
fit3 <- ets(y.trn,model="AAN",damped=FALSE)
# Damped trend model
fit4 <- ets(y.trn,model="AAN",damped=TRUE)
# Trend seasonal model
fit5 <- ets(y.trn,model="AAA",damped=FALSE)
# Damped trend seasonal model
fit6 <- ets(y.trn,model="AAA",damped=TRUE)
```

```r
aicc <- c(fit1$aicc,fit2$aicc,fit3$aicc,fit4$aicc,fit5$aicc,fit6$aicc)

names(aicc) <- c("ANN","ANA","AAN","AAdN","AAA","AAdA")
aicc
```

```
##      ANN      ANA      AAN     AAdN      AAA     AAdA
## 7727.092 7573.495 7732.440 7767.566 7578.661 7580.221
```

```r
which.min(aicc)
```

```
## ANA
##   2
```

```r
fit$aicc
```

```
## [1] 7538.142
```

```r
fit2$aicc
```

```
## [1] 7573.495
```

3.2 Selection of forecasts using a validation set

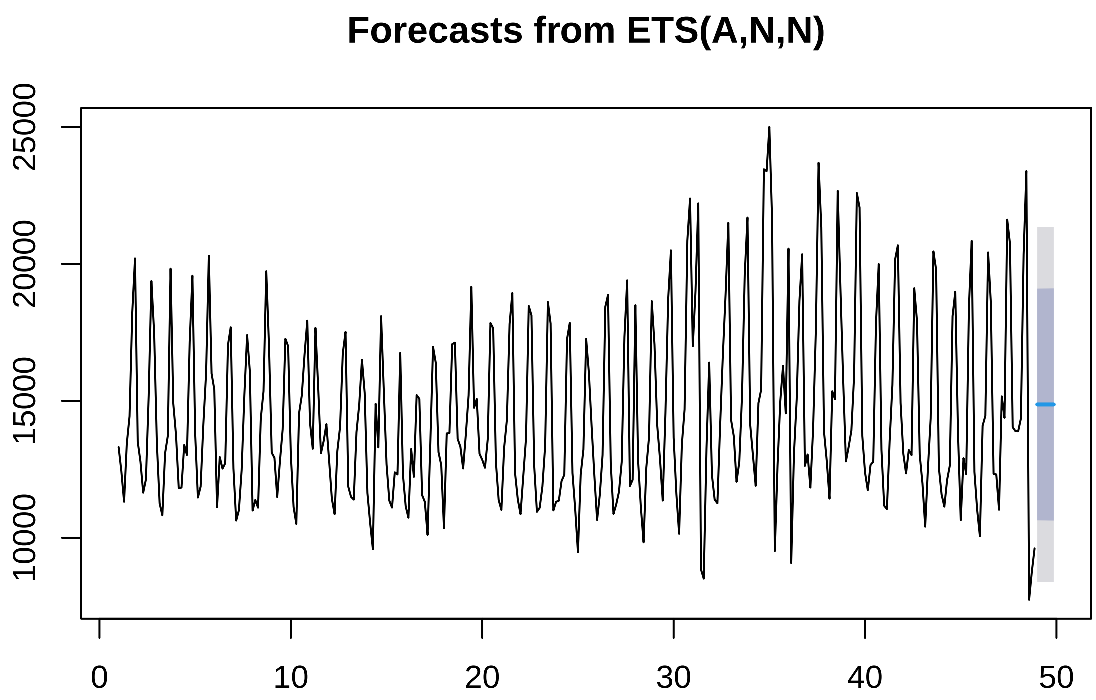
```r
y.ins <- head(y.trn,48*7)
y.val <- tail(y.trn,2*7)
```

```r
h <- 7
```

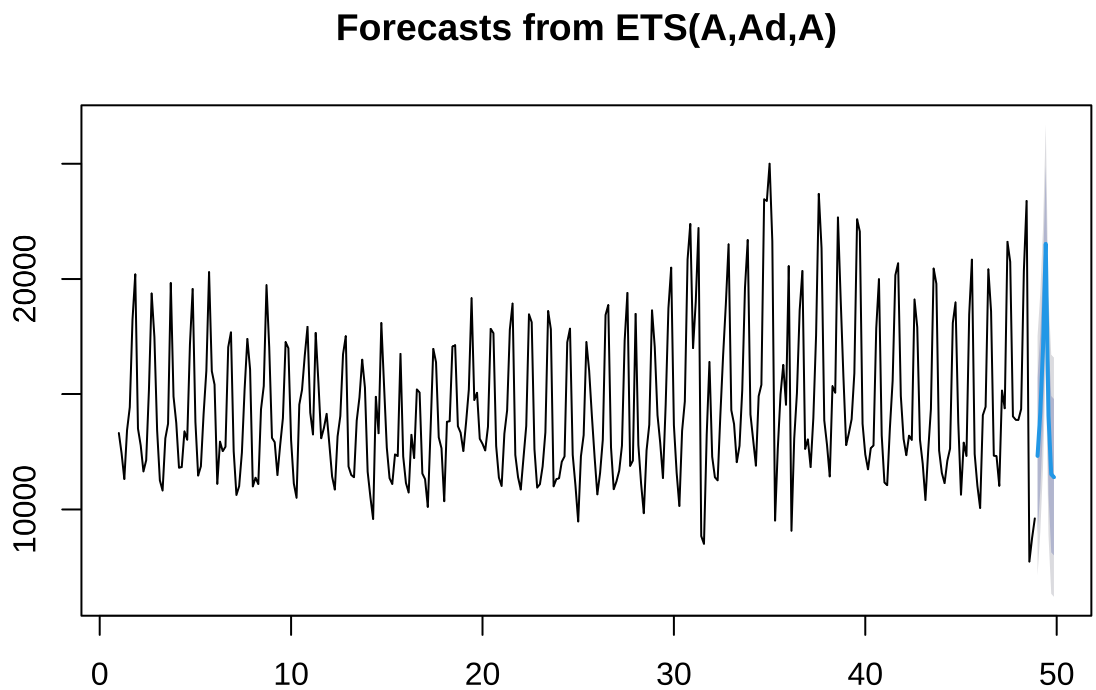
```r
fit1v <- ets(y.ins,model="ANN")
fit2v <- ets(y.ins,model="ANA")
fit3v <- ets(y.ins,model="AAN",damped=FALSE)
fit4v <- ets(y.ins,model="AAN",damped=TRUE)
fit5v <- ets(y.ins,model="AAA",damped=FALSE)
fit6v <- ets(y.ins,model="AAA",damped=TRUE)
fit7v <- ets(y.ins,model="MNM")
```

```r
frc1v <- forecast(fit1v,h=h)
frc2v <- forecast(fit2v,h=h)
frc3v <- forecast(fit3v,h=h)
frc4v <- forecast(fit4v,h=h)
frc5v <- forecast(fit5v,h=h)
frc6v <- forecast(fit6v,h=h)
frc7v <- forecast(fit7v,h=h)
frc8v <- tail(y.ins,frequency(y.ins))[1:h]
```

```r
plot(frc1v)
```




```r
plot(frc6v)
```




```r
err1v <- mean(abs(y.val[1:h] - frc1v$mean))
err2v <- mean(abs(y.val[1:h] - frc2v$mean))
err3v <- mean(abs(y.val[1:h] - frc3v$mean))
err4v <- mean(abs(y.val[1:h] - frc4v$mean))
err5v <- mean(abs(y.val[1:h] - frc5v$mean))
err6v <- mean(abs(y.val[1:h] - frc6v$mean))
err7v <- mean(abs(y.val[1:h] - frc7v$mean))

err8v <- mean(abs(y.val[1:h] - frc8v))
```

```r
errv <- c(err1v, err2v, err3v, err4v, err5v, err6v, err7v, err8v)
names(errv) <- c("ANN","ANA","AAN","AAdN","AAA","AAdA","MNM","Naive")
errv
```

```
##      ANN      ANA      AAN     AAdN      AAA     AAdA      MNM    Naive
## 2975.734 2822.253 3040.983 4786.245 2826.875 2822.616 2270.250 5367.263
```

```r
which.min(errv)
```

```
## MNM
##   7
```

```r
omax <- length(y.val) - h + 1
omax
```

```
## [1] 8
```

```r
models <- c("ANN", "ANA", "AAN", "AAN", "AAA", "AAA", "MNM", "Naive")
damped <- c(FALSE, FALSE, FALSE, TRUE, FALSE, TRUE, FALSE, FALSE)

err <- array(NA,c(omax,8))

frcs <- array(NA,c(h,8))
```

```r
for (o in 1:omax){
print(o)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
```

```r
# For each forecast origin
for (o in 1:omax){
  y.ins <- head(y.trn,48*7-1+o)
  y.val <- tail(y.trn,2*7-o+1)

  # Fit and forecast
  for (m in 1:7){
  fitTemp <- ets(y.ins,model=models[m],damped=damped[m])
  frcs[,m] <- forecast(fitTemp,h=h)$mean
  err[o,m] <- mean(abs(y.val[1:h] - frcs[,m]))
  }
  # seasonal naive
  frcs[,8] <- tail(y.ins,frequency(y.ins))[1:h]
  err[o,8] <- mean(abs(y.val[1:h] - frcs[,8]))
}
```

```r
colnames(err) <- c("ANN", "ANA", "AAN", "AAdN", "AAA", "AAdA", "MNM", "Naive")
err
```

```
##           ANN      ANA      AAN     AAdN      AAA     AAdA      MNM    Naive
## [1,] 2975.734 2822.253 3040.983 4786.245 2826.875 2822.616 2270.250 5367.263
## [2,] 2603.560 2459.703 2647.831 5340.174 2461.724 2452.621 2172.020 5019.013
## [3,] 2262.265 2055.915 1945.982 3877.711 2061.953 2050.249 2248.446 4842.480
## [4,] 2434.633 1988.818 2449.286 2160.198 1997.578 1988.870 2488.761 4109.314
## [5,] 2530.054 1843.592 2574.834 5254.436 1856.715 1847.040 2251.791 3647.554
## [6,] 2472.729 1813.034 2568.853 4881.620 1815.201 1811.906 1781.343 3019.866
## [7,] 2816.176 1684.053 2903.166 2208.285 1684.137 1684.079 1606.699 2634.483
## [8,] 3148.847 1546.037 3233.429 2379.490 1546.174 1545.162 1568.200 2400.111
```

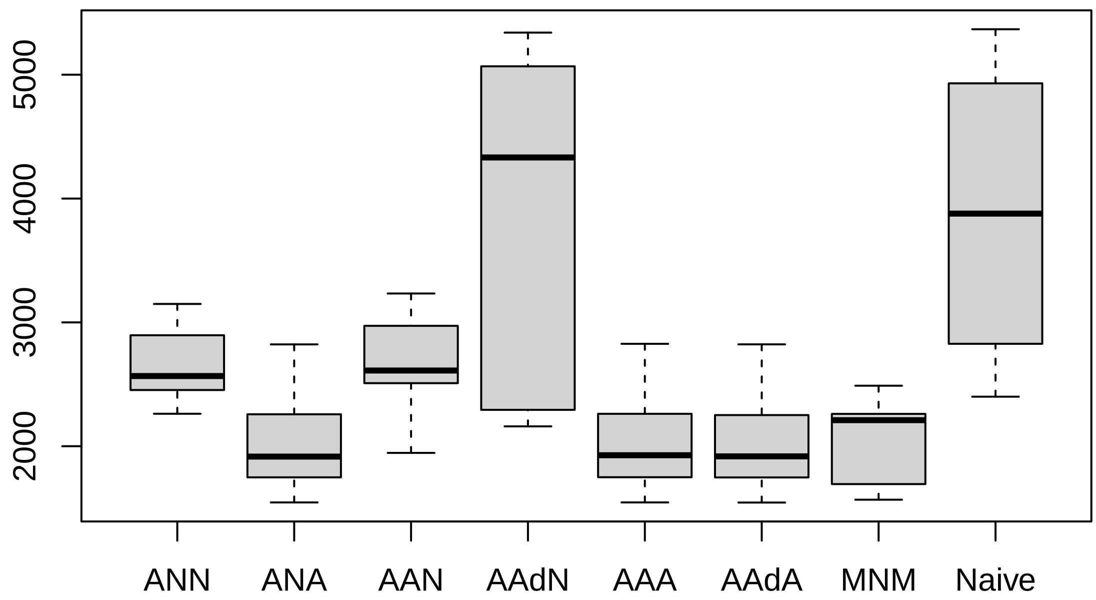
```r
errMean <- colMeans(err)
errMean
```

```
##      ANN      ANA      AAN     AAdN      AAA     AAdA      MNM    Naive
## 2655.500 2026.676 2670.545 3861.020 2031.295 2025.318 2048.439 3880.011
```

```r
which.min(errMean)
```

```
## AAdA
##    6
```

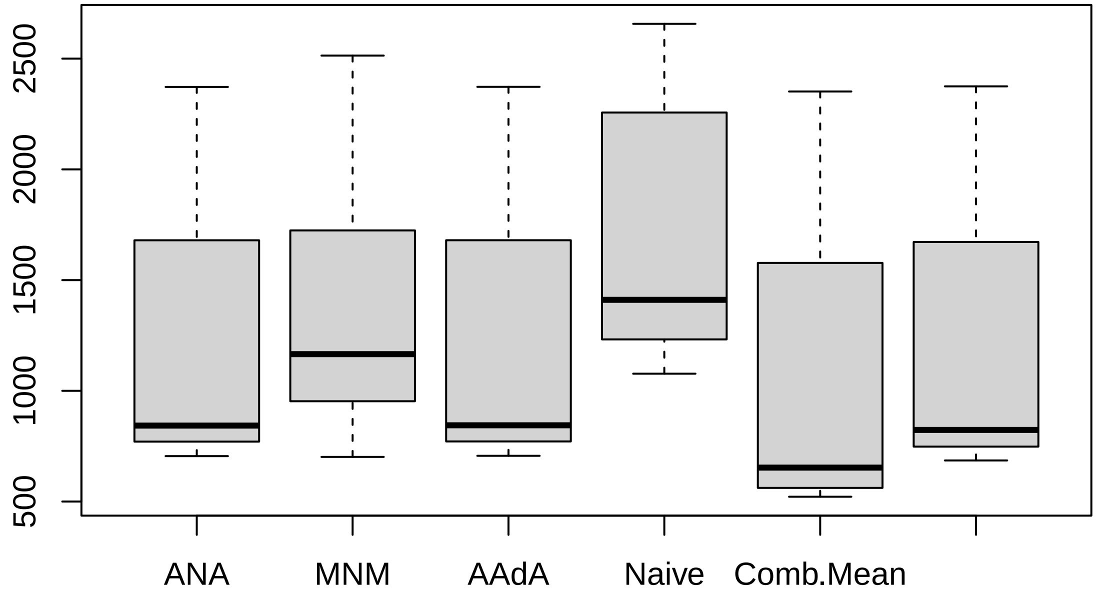
```r
boxplot(err)
```



4. Out-of-sample evaluation

```r
modelsTest <- c("ANA", "MNM", "AAA", "Naive", "CombMean", "CombMedian")
dampedTest <- c(FALSE, FALSE, TRUE)

# Pre-allocate memory
omaxTest <- length(y.tst) - h + 1
errTest <- array(NA,c(omaxTest,6))
frcsTest <- array(NA,c(h,6))
```

```r
# For each forecast origin
for (o in 1:omaxTest){

  y.trnTest <- head(y,50*7-1+o)
  y.tstTest <- tail(y,2*7-o+1)

  # Fit and forecast exponential smoothing models
  for (m in 1:3){
    fitTemp <- ets(y.trnTest,model=modelsTest[m],damped=dampedTest[m])
    frcsTest[,m] <- forecast(fitTemp,h=h)$mean
    errTest[o,m] <- mean(abs(y.tstTest[1:h] - frcsTest[,m]))
  }

  # Forecast using the seasonal naive
  frcsTest[,4] <- tail(y.trnTest,frequency(y.trnTest))[1:h]
  errTest[o,4] <- mean(abs(y.tstTest[1:h] - frcsTest[,4]))

  # Combinations
  # Mean
  frcsTest[,5] <- apply(frcsTest[,1:4],1,mean)
  errTest[o,5] <- mean(abs(y.tstTest[1:h] - frcsTest[,5]))

  # Median:
  frcsTest[,6] <- apply(frcsTest[,1:4],1,median)
  errTest[o,6] <- mean(abs(y.tstTest[1:h] - frcsTest[,6]))
}

# Assign names to errors
colnames(errTest) <- c("ANA","MNM","AAdA","Naive","Comb.Mean","Comb.Median")

# Summarise and plot errors
boxplot(errTest)
```




```r
errTestMean <- colMeans(errTest)
print(errTestMean)
```

```
##         ANA         MNM        AAdA       Naive   Comb.Mean Comb.Median
##    1208.019    1362.662    1208.888    1691.712    1057.314    1193.481
```

```r
which.min(errTestMean)
```

```
## Comb.Mean
##         5
```

5. Forecast combination with AIC weights

```r
y.trn <- window(AirPassengers,end=c(1959,12))
y.tst <- window(AirPassengers,start=c(1960,1))
```

```r
models <- c("ANN","AAN","MNM","MAM")
```

```r
fit <- list()
frc <- array(NA,c(12,4),dimnames=list(NULL,models))
```

```r
for (i in 1:4){
fit[[i]] <- ets(y.trn,model=models[i],damped=FALSE)
frc[,i] <- forecast(fit[[i]],h=12)$mean
}
```

```r
AIC <- unlist(lapply(fit,function(x){x$aic}))
AIC
```

```
## [1] 1558.920 1562.628 1297.518 1257.573
```

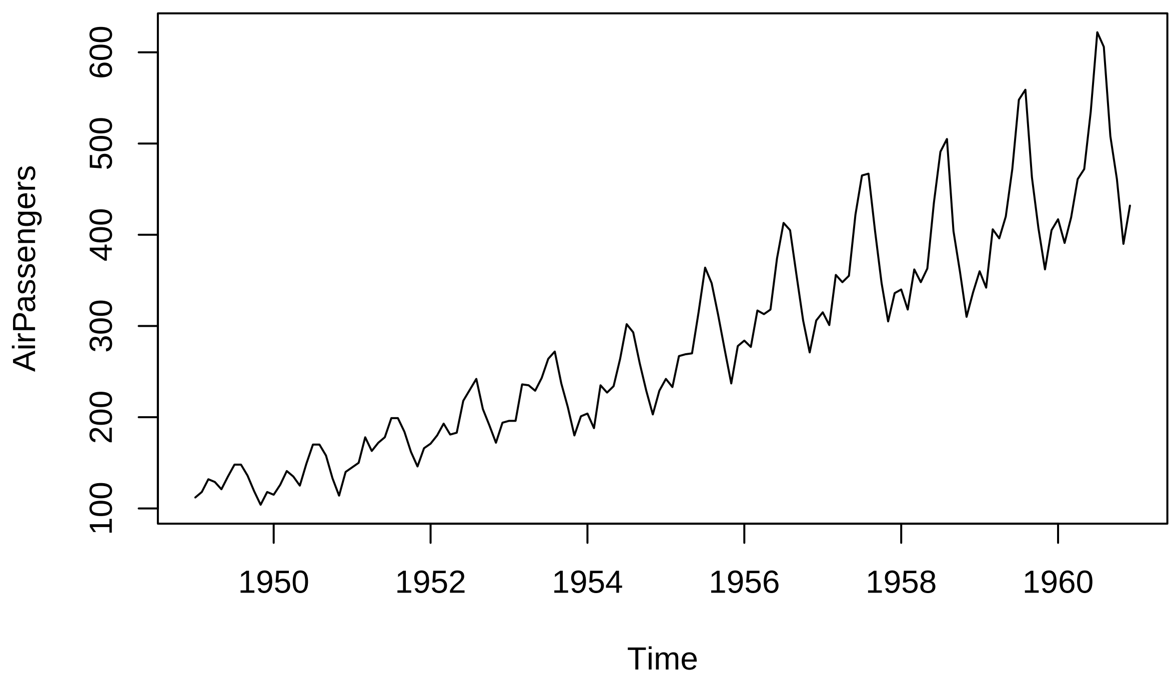
```r
dAIC <- AIC - min(AIC)
dAIC <- exp(-0.5*dAIC)
waic <- dAIC/sum(dAIC)
waic
```

```
## [1] 3.659296e-66 5.730754e-67 2.118497e-09 1.000000e+00
```

```r
round(waic,4)
```

```
## [1] 0 0 0 1
```

```r
plot(AirPassengers)
```



```r
# Prepare variables and models
fit2 <- list()
frc2 <- array(NA,c(12,6))
models <- rep(c("AAA","MAM","MMM"),2)
damped <- c(rep(FALSE,3),rep(TRUE,3))

# Fit models and generate forecasts
for (i in 1:6){
fit2[[i]] <- ets(y.trn,model=models[i],damped=damped[i])
frc2[,i] <- forecast(fit2[[i]], h = 12)$mean
}
#Extract AIC and calculate weights
AIC2 <- unlist(lapply(fit2,function(x){x$aic}))
dAIC2 <- AIC2 - min(AIC2)
dAIC2 <- exp(-0.5*dAIC2)
waic2 <- dAIC2/sum(dAIC2)
round(waic2,4)
```

```
## [1] 0.0000 0.0005 0.0001 0.0000 0.3481 0.6513
```

```r
# AIC weights
frcComb <- frc2 %*% cbind(waic2)

# Mean
frcComb <- cbind(frcComb, rowMeans(frc2))

# Median
frcComb <- cbind(frcComb, apply(frc2,1,median))

# Selection
frcComb <- cbind(frcComb, frc2[,which.min(AIC2)])
colnames(frcComb) <- c("Comb.AIC","Comb.Mean","Comb.Median","Selection")
```

```r
err <- matrix(rep(y.tst,4),ncol=4) - frcComb
MAE <- colMeans(abs(err))
round(MAE,2)
```

```
##    Comb.AIC   Comb.Mean Comb.Median   Selection
##       22.03       20.74       20.99       21.64
```

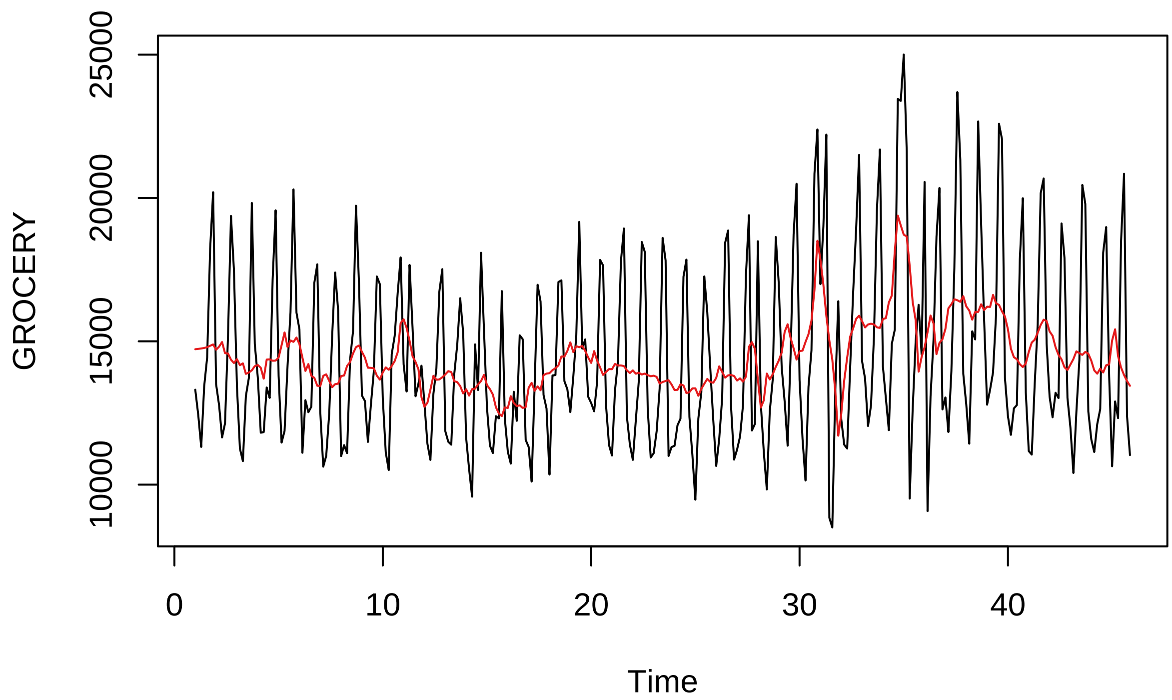
Exercise

Question 1

1. in- and out-of-sample and data exploration

```r
y.trn <- head(y,7*45)
y.tst <- tail(y,7*7)
```

```r
cma <- cmav(y.trn, outplot=TRUE)
```



2. Forecasting

2.1 Selection of forecasts using information criteria

```r
fit <- ets(y.trn)
fit
```

```
## ETS(M,N,M)
##
## Call:
##  ets(y = y.trn)
##
##   Smoothing parameters:
##     alpha = 0.0944
##     gamma = 0.3541
##
##   Initial states:
##     l = 14630.2616
##     s = 1.279 1.2932 0.9912 0.8979 0.7857 0.8307
##            0.9222
##
```

```
##   sigma:  0.1695
##
##      AIC     AICc      BIC
## 6732.795 6733.518 6770.320
```

```
# Level model
fit1 <- ets(y.trn,model="ANN")
# Seasonal model
fit2 <- ets(y.trn,model="ANA")
# Linear trend model
fit3 <- ets(y.trn,model="AAN",damped=FALSE)
# Damped trend model
fit4 <- ets(y.trn,model="AAN",damped=TRUE)
# Trend seasonal model
fit5 <- ets(y.trn,model="AAA",damped=FALSE)
# Damped trend seasonal model
fit6 <- ets(y.trn,model="AAA",damped=TRUE)
```

```
aicc <- c(fit1$aicc,fit2$aicc,fit3$aicc,fit4$aicc,fit5$aicc,fit6$aicc)

# Name Aicc vector
names(aicc) <- c("ANN","ANA","AAN","AAdN","AAA","AAdA")
aicc
```

```
##      ANN      ANA      AAN     AAdN      AAA     AAdA
## 6903.671 6755.773 6909.709 6934.329 6760.632 6762.468
```

```
which.min(aicc)
```

```
## ANA
##   2
```

```
fit$aicc
```

```
## [1] 6733.518
```

```
fit2$aicc
```

```
## [1] 6755.773
```

2.2 Selection of forecasts using a validation set

```
y.ins <- head(y.trn,35*7)
y.val <- tail(y.trn,10*7)
```

```
h <- 7
```

```r
fit1v <- ets(y.ins,model="ANN")
fit2v <- ets(y.ins,model="ANA")
fit3v <- ets(y.ins,model="AAN",damped=FALSE)
fit4v <- ets(y.ins,model="AAN",damped=TRUE)
fit5v <- ets(y.ins,model="AAA",damped=FALSE)
fit6v <- ets(y.ins,model="AAA",damped=TRUE)
fit7v <- ets(y.ins,model="MNM")
```

```r
frc1v <- forecast(fit1v,h=h)
frc2v <- forecast(fit2v,h=h)
frc3v <- forecast(fit3v,h=h)
frc4v <- forecast(fit4v,h=h)
frc5v <- forecast(fit5v,h=h)
frc6v <- forecast(fit6v,h=h)
frc7v <- forecast(fit7v,h=h)
frc8v <- tail(y.ins,frequency(y.ins))[1:h]
```

```r
err1v <- mean(abs(y.val[1:h] - frc1v$mean))
err2v <- mean(abs(y.val[1:h] - frc2v$mean))
err3v <- mean(abs(y.val[1:h] - frc3v$mean))
err4v <- mean(abs(y.val[1:h] - frc4v$mean))
err5v <- mean(abs(y.val[1:h] - frc5v$mean))
err6v <- mean(abs(y.val[1:h] - frc6v$mean))
err7v <- mean(abs(y.val[1:h] - frc7v$mean))

err8v <- mean(abs(y.val[1:h] - frc8v))
```

```r
errv <- c(err1v, err2v, err3v, err4v, err5v, err6v, err7v, err8v)
names(errv) <- c("ANN","ANA","AAN","AAdN","AAA","AAdA","MNM","Naive")
errv
```

```
##      ANN      ANA      AAN     AAdN      AAA     AAdA      MNM    Naive
## 3612.514 3523.516 3759.723 3612.566 3471.660 3521.552 3555.182 4660.434
```

```r
which.min(errv)
```

```
## AAA
##   5
```

```r
omax <- length(y.val) - h + 1
omax
```

```
## [1] 64
```

```r
# Define Model's
models <- c("ANN", "ANA", "AAN", "AAN", "AAA", "AAA", "MNM", "Naive")
damped <- c(FALSE, FALSE, FALSE, TRUE, FALSE, TRUE, FALSE, FALSE)

err <- array(NA,c(omax,8))

frcs <- array(NA,c(h,8))
```

```r
# For each forecast origin
for (o in 1:omax){
  y.ins <- head(y.trn,35*7-1+o)
  y.val <- tail(y.trn,10*7-o+1)

  # Fit and forecast
  for (m in 1:7){
  fitTemp <- ets(y.ins,model=models[m],damped=damped[m])
  frcs[,m] <- forecast(fitTemp,h=h)$mean
  err[o,m] <- mean(abs(y.val[1:h] - frcs[,m]))
  }
  # Forecast using the seasonal naive
  frcs[,8] <- tail(y.ins,frequency(y.ins))[1:h]
  err[o,8] <- mean(abs(y.val[1:h] - frcs[,8]))
}
```

```r
colnames(err) <- c("ANN", "ANA", "AAN", "AAdN", "AAA", "AAdA", "MNM", "Naive")
```

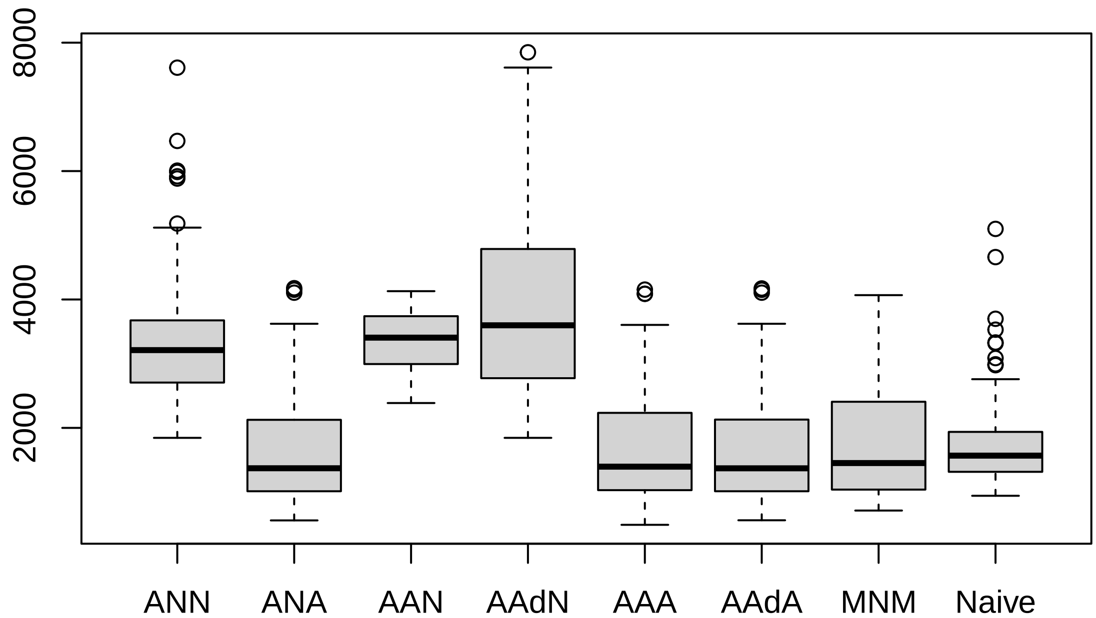
```r
errMean <- colMeans(err)
errMean
```

```
##      ANN      ANA      AAN     AAdN      AAA     AAdA      MNM    Naive
## 3513.851 1741.863 3354.955 3934.091 1756.752 1741.815 1785.405 1821.107
```

```r
which.min(errMean)
```

```
## AAdA
##    6
```

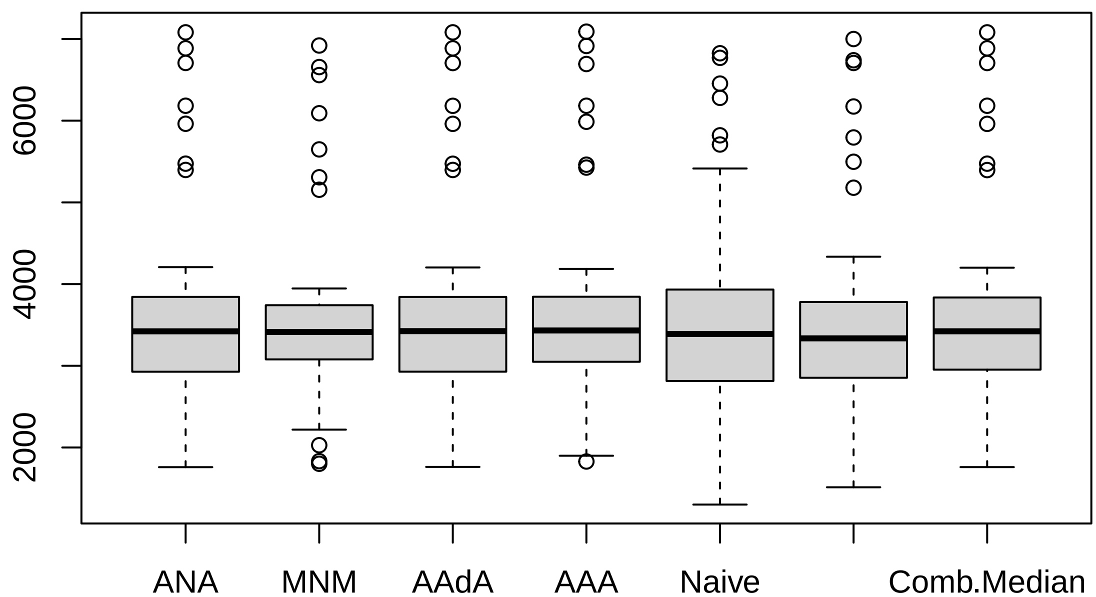
```r
boxplot(err)
```



3. Out-of-sample evaluation

```
modelsTest <- c("ANA", "MNM", "AAA","AAA", "Naive", "CombMean", "CombMedian")
dampedTest <- c(FALSE, FALSE, TRUE,FALSE)

# Pre-allocate memory
omaxTest <- length(y.tst) - h + 1
errTest <- array(NA,c(omaxTest,7))
frcsTest <- array(NA,c(h,7))

# For each forecast origin
for (o in 1:omaxTest){
  y.trnTest <- head(y,35*7-1+o)
  y.tstTest <- tail(y,10*7-o+1)

  # Fit and forecast
  for (m in 1:4){
    fitTemp <- ets(y.trnTest,model=modelsTest[m],damped=dampedTest[m])
    frcsTest[,m] <- forecast(fitTemp,h=h)$mean
    errTest[o,m] <- mean(abs(y.tstTest[1:h] - frcsTest[,m]))
  }

  # Forecast using the seasonal naive
  frcsTest[,5] <- tail(y.trnTest,frequency(y.trnTest))[1:h]
```

```r
  errTest[o,5] <- mean(abs(y.tstTest[1:h] - frcsTest[,5]))

  # Combinations
  # Mean
  frcsTest[,6] <- apply(frcsTest[,1:5],1,mean)
  errTest[o,6] <- mean(abs(y.tstTest[1:h] - frcsTest[,6]))

  # Median
  frcsTest[,7] <- apply(frcsTest[,1:5],1,median)
  errTest[o,7] <- mean(abs(y.tstTest[1:h] - frcsTest[,7]))
}
# Assign names to errors
colnames(errTest) <- c("ANA","MNM","AAdA","AAA","Naive","Comb.Mean","Comb.Median")

# Summarise and plot errors
boxplot(errTest)
```




```r
errTestMean <- colMeans(errTest)
print(errTestMean)
```

```
##         ANA         MNM        AAdA         AAA       Naive   Comb.Mean
##    3654.662    3623.760    3655.350    3683.399    3609.080    3562.348
## Comb.Median
##    3657.794
```

```r
which.min(errTestMean)
```

```
## Comb.Mean
##         6
```

Yes,

In the context of time series forecasting for grocery sales in a US supermarket store, two distinct forecasting methodologies were employed to evaluate the impact of varying data splits on forecasting outcomes. Initially, a data split consisting of 48 weeks for training, 2 weeks for validation, and 2 weeks for testing was utilized. Under this configuration, both automatic model selection and method-wise analysis consistently favored the MNM model for forecasting. Concurrently, the information criteria consistently indicated ANA as the preferred model choice, and validation through single iterations consistently resulted in the MNM model being selected. Furthermore, the rolling origin validation approach produced the AAdA model, while the rolling origin combined with the mean model exhibited distinct performance characteristics.

Subsequently, the data split was modified to comprise 35 weeks for training, 10 weeks for validation, and 7 weeks for testing, leading to a shift in the validation dynamics. Surprisingly, despite this change in data distribution, the automatic ETS model selection process continued to favor the MNM model, indicating its robustness across different data splits. The information criteria consistently pointed to ANA, reaffirming its suitability. However, the validation approach using single iterations notably shifted towards favoring the AAA model, suggesting sensitivity to the training-validation-test ratio. In contrast, the rolling origin validation approach remained relatively stable, consistently yielding the AAdA model as the chosen option. The utilization of a rolling origin approach in conjunction with the mean model persisted as a viable alternative.

The comparative analysis of the initial and revised results underscores the profound impact of altering the training, validation, and test split configurations on the performance and selection of forecasting models. It highlights the resilience of certain model choices, such as MNM, in the face of changing data distributions, while also revealing the sensitivity of other approaches, notably single iteration validation, to shifts in data distribution. These findings accentuate the critical importance of carefully considering and selecting data splits when engaging in time series forecasting tasks, as they wield significant influence over the choice and effectiveness of forecasting models. Ultimately, these insights guide decision-making processes in the adoption of specific forecasting methodologies for the context of grocery sales forecasting within a supermarket setting.

Question 2

For this question, I'm comparing AIC weight approach with Rolling origin approach,
So I'll be creating same trend, season model's ("AAA","MAM","MMM"), with and without damped models.
Also using the same -In and -Out sample split.

1. in- and out-of-sample and data exploration

```r
y <- AirPassengers
```

```r
y.trn <- head(y,11*12)
y.tst <- tail(y,1*12)
```

2. Forecasting

```r
fit <- ets(y.trn)
fit
```

```
## ETS(M,Ad,M)
##
## Call:
##  ets(y = y.trn)
##
##   Smoothing parameters:
##     alpha = 0.758
##     beta  = 0.0213
##     gamma = 1e-04
##     phi   = 0.98
##
##   Initial states:
##     l = 120.7483
##     b = 1.7632
##     s = 0.897 0.798 0.919 1.0587 1.2156 1.2251
##            1.1075 0.9782 0.9804 1.0207 0.8926 0.9073
##
##   sigma:  0.0378
##
##      AIC     AICc      BIC
## 1244.458 1250.511 1296.348
```

2.1 Rolling origin testing

```r
h <- 12
```

```r
omax <- length(y.tst) - h + 1
omax
```

```
## [1] 1
```

```r
models <- c("AAA", "MAM", "MMM", "AAA", "MAM", "MMM")
damped <- c(FALSE, FALSE, FALSE, TRUE, TRUE, TRUE)
err <- array(NA,c(omax,6))
frcs <- array(NA,c(h,6))
```

```r
# For each forecast origin
for (o in 1:omax){
  # Split training set
  y.trn <- head(y.trn,11*12-1+o)
  y.tst <- tail(y.tst,1*12-o+1)

  # Fit and forecast same 6 models
  for (m in 1:6){
    fitTemp <- ets(y.trn,model=models[m],damped=damped[m])
    frcs[,m] <- forecast(fitTemp,h=h)$mean
    err[o,m] <- mean(abs(y.tst[1:h] - frcs[,m]))
```

```r
  }
}
```

```r
colnames(err) <- c("AAA", "MAM", "MMM", "AAdA", "MAdM", "MMdM")
err
```

```
##           AAA      MAM     MMM     AAdA    MAdM     MMdM
## [1,] 40.59396 10.80856 19.1405 46.09029 22.8045 21.63728
```

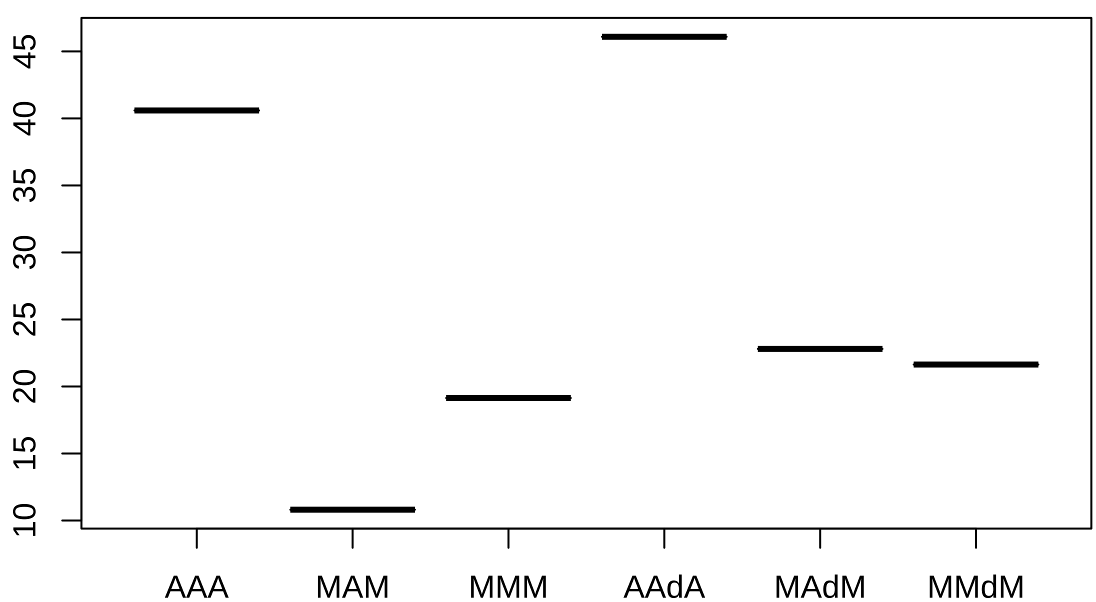
```r
errMean <- colMeans(err)
errMean
```

```
##      AAA      MAM      MMM     AAdA     MAdM     MMdM
## 40.59396 10.80856 19.14050 46.09029 22.80450 21.63728
```

```r
which.min(errMean)
```

```
## MAM
##   2
```

```r
boxplot(err)
```



Yes the results changed.

Approach 1, “Forecast Combination with AIC Weights”:

Under Approach 1, a combination of forecasting models was explored, incorporating various weighted averaging techniques. Specifically, this approach considered models such as the Combined Weighted Average, Combined Mean, Combined Median, and the Minimum AIC model.

The outcomes of Approach 1 indicated that the ensemble-based methodology led to relatively low forecast errors. Notably, the Minimum AIC model demonstrated competitive performance, exhibiting a forecast error of 21.64.

In this approach, the emphasis was on leveraging the principles of model combination and selection, with the goal of harnessing the strengths of multiple models to enhance forecasting accuracy.

Approach 2, “Rolling Origin”:

Approach 2, adopted a distinct strategy. This methodology involved the utilization of a rolling origin approach, where each individual forecasting model (AAA, MAM, MMM, AAdA, MAdM, MMdM) was independently tested.

The results obtained in Approach 2 revealed a discernible shift in forecasting performance when compared to Approach 1. Significantly, one specific individual model consistently outperformed the others in terms of forecast accuracy.

The standout model in Approach 2 was identified as the MAM (Model-Additive-Additive-Additive) model, which consistently yielded a notably low forecast error of 10.81 across the test periods.

Unlike Approach 1, which relied on model combination techniques, Approach 2 highlighted the superiority of a single, carefully chosen model in effectively capturing the underlying patterns present within the dataset.

Interpretation of Results:

The key insight derived from these results is that, within the context of forecasting for the provided dataset (AirPassengers), Approach 2, identified a single model, namely MAM, that consistently produced the most accurate forecasts.

This finding challenges the conventional belief that ensemble or combined forecasting models are universally superior. Instead, it underscores the critical importance of meticulous model selection and testing methodologies. Approach 2 demonstrates that an appropriately selected individual model, such as MAM, can surpass combined models when tailored to the specific characteristics of the dataset.

The success of MAM in Approach 2 underscores its potential as a valuable forecasting tool, offering enhanced forecast accuracy and operational efficiency for practical applications.

In summary, the results within the context of Approach 1 and Approach 2 highlight a substantial shift in forecasting performance. Approach 2, showcases the emergence of a single, superior model (MAM), emphasizing the significance of data-driven model selection and demonstrating that simplicity and meticulous model choice can profoundly impact forecasting accuracy in real-world scenarios.