Lab5_Regression_2_a24kimwu

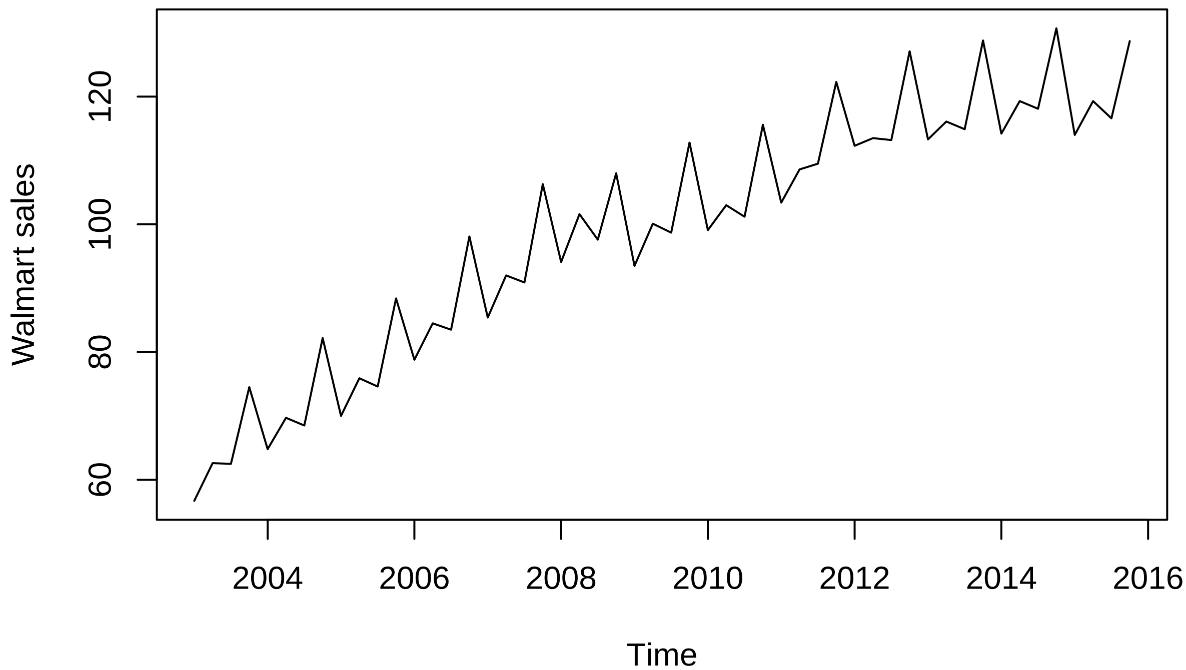
2025-10-01

Advanced regression modelling

```r
x <- ts(read.csv("./walmart.csv"),frequency=4,start=c(2003,1))
# Print the first 10 rows
x[1:10,]
```

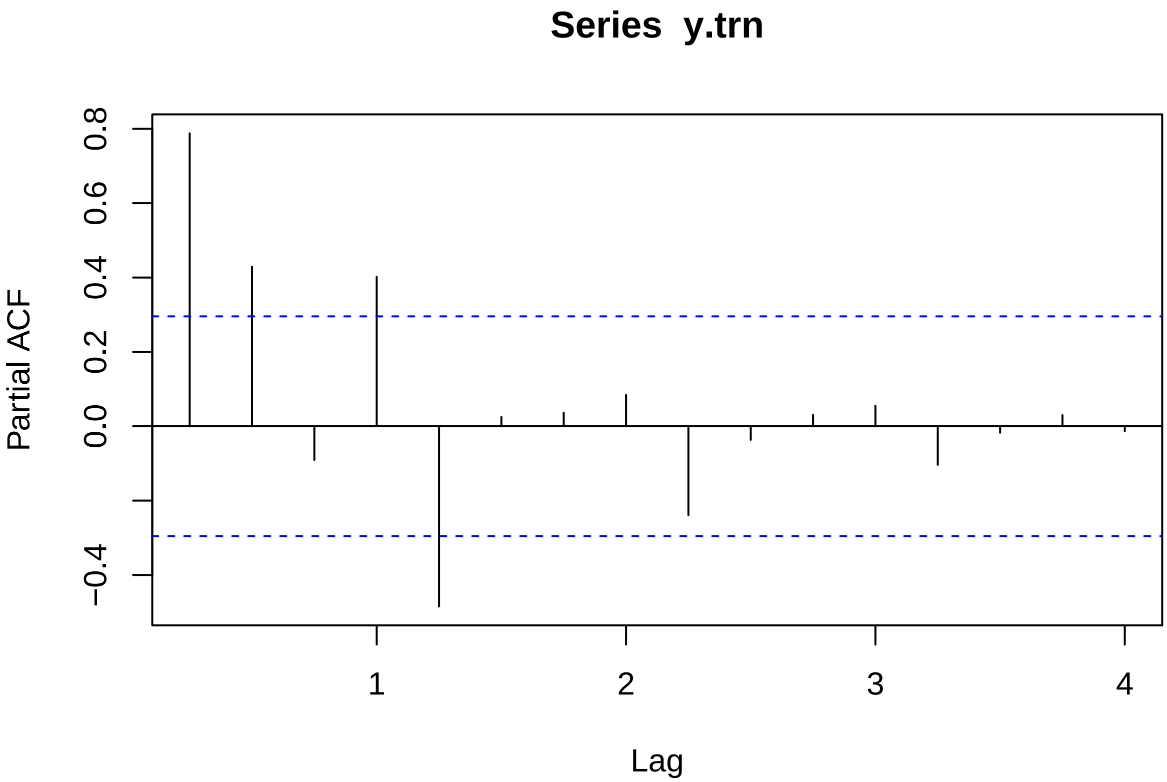
```
##       sales     gdp
##  [1,]  56.7 11230.1
##  [2,]  62.6 11370.7
##  [3,]  62.5 11625.1
##  [4,]  74.5 11816.8
##  [5,]  64.8 11988.4
##  [6,]  69.7 12181.4
##  [7,]  68.5 12367.7
##  [8,]  82.2 12562.2
##  [9,]  70.0 12813.7
## [10,]  75.9 12974.1
```

```r
plot(x[,1],ylab="Walmart sales")
```

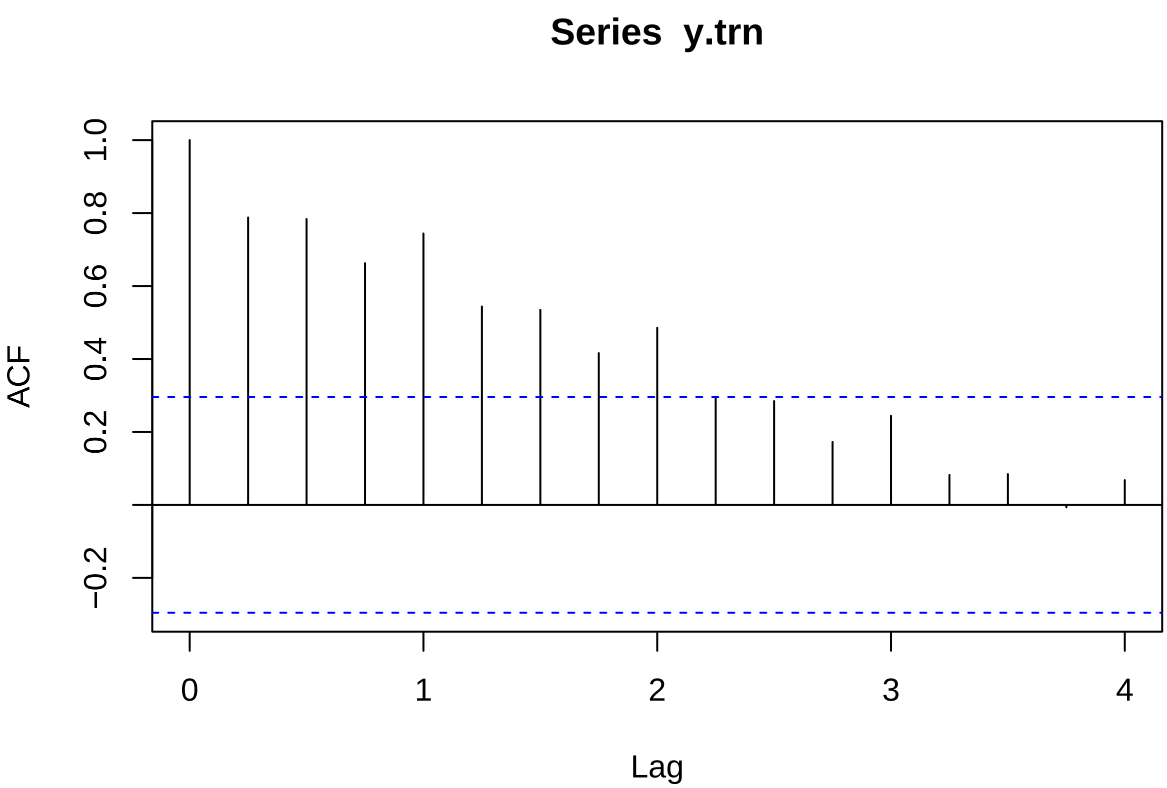



```r
y.trn <- window(x[,1],end=c(2013,4))
y.tst <- window(x[,1],start=c(2014,1))
```

```r
pacf(y.trn)
```




```r
acf(y.trn)
```



Construct Lags

```r
n<-length(y.trn)
n
```

```
## [1] 44
```

```r
 X<-array(NA,c(n,6))
```

```r
 #We start a loop, which will iterate for all values of i=1,2,3,4,5,6
for(i in 1:6){
 #We tell it to place the data in the i th column, from observation i till the end.
 #We place the data from the beginning towards as much as we can fit to the array (the n-i+1bit).
 X[i:n,i]<-y.trn[1:(n-i+1)]
}
 #Name the columns
 #paste0("lag",1:5) creates names lag1, lag2,lag3,lag4,lag5
 colnames(X)<-c("y",paste0("lag",1:5))
 #Let us see how the resulting array looks like (the first 10 observations)
 X[1:10,]
```

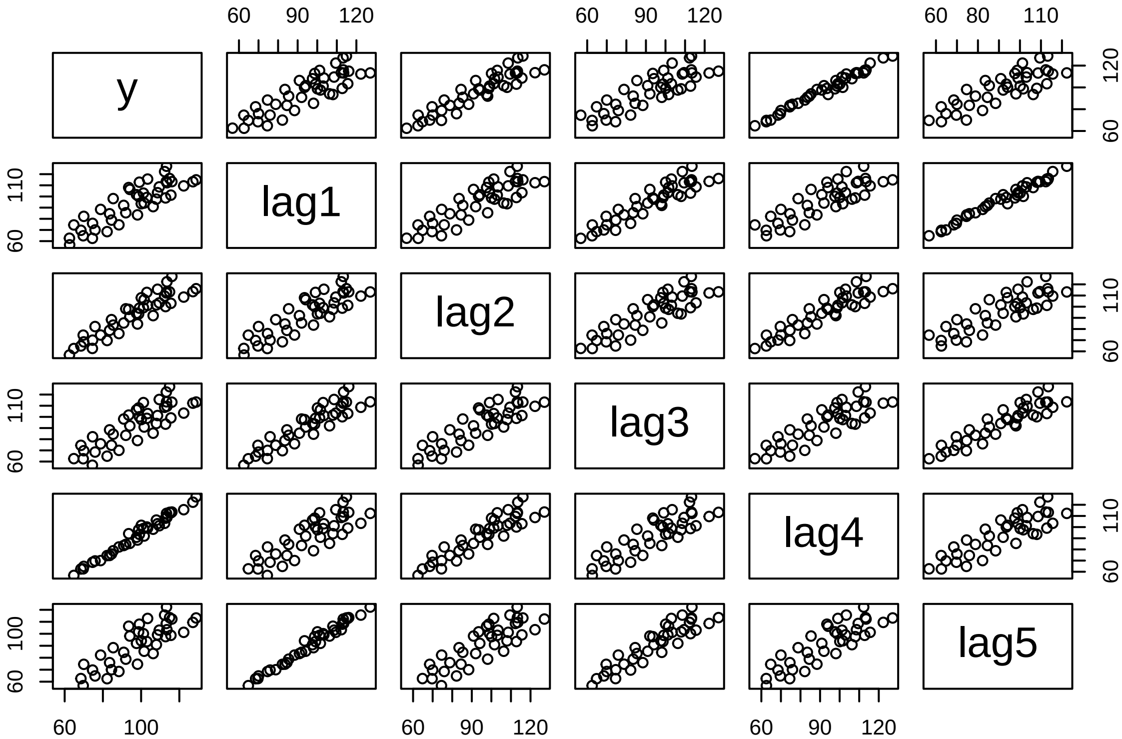
```
##          y lag1 lag2 lag3 lag4 lag5
##  [1,] 56.7   NA   NA   NA   NA   NA
##  [2,] 62.6 56.7   NA   NA   NA   NA
##  [3,] 62.5 62.6 56.7   NA   NA   NA
##  [4,] 74.5 62.5 62.6 56.7   NA   NA
##  [5,] 64.8 74.5 62.5 62.6 56.7   NA
##  [6,] 69.7 64.8 74.5 62.5 62.6 56.7
##  [7,] 68.5 69.7 64.8 74.5 62.5 62.6
##  [8,] 82.2 68.5 69.7 64.8 74.5 62.5
##  [9,] 70.0 82.2 68.5 69.7 64.8 74.5
## [10,] 75.9 70.0 82.2 68.5 69.7 64.8
```

```r
X[(n-9):n,] # Observe the use of parenthesis when I calculate locations in an array
```

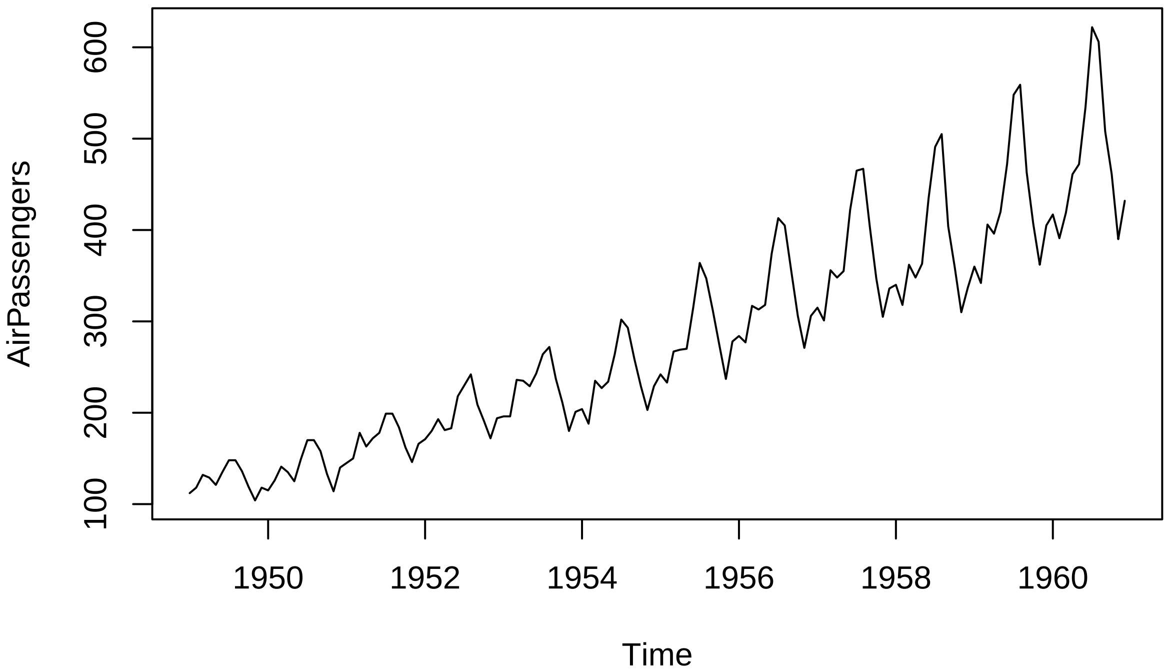
```
##           y  lag1  lag2  lag3  lag4  lag5
##  [1,] 109.5 108.6 103.4 115.6 101.2 103.0
##  [2,] 122.3 109.5 108.6 103.4 115.6 101.2
##  [3,] 112.3 122.3 109.5 108.6 103.4 115.6
##  [4,] 113.5 112.3 122.3 109.5 108.6 103.4
##  [5,] 113.2 113.5 112.3 122.3 109.5 108.6
##  [6,] 127.1 113.2 113.5 112.3 122.3 109.5
##  [7,] 113.3 127.1 113.2 113.5 112.3 122.3
##  [8,] 116.1 113.3 127.1 113.2 113.5 112.3
##  [9,] 114.9 116.1 113.3 127.1 113.2 113.5
## [10,] 128.8 114.9 116.1 113.3 127.1 113.2
```

```r
X <- as.data.frame(X)
```

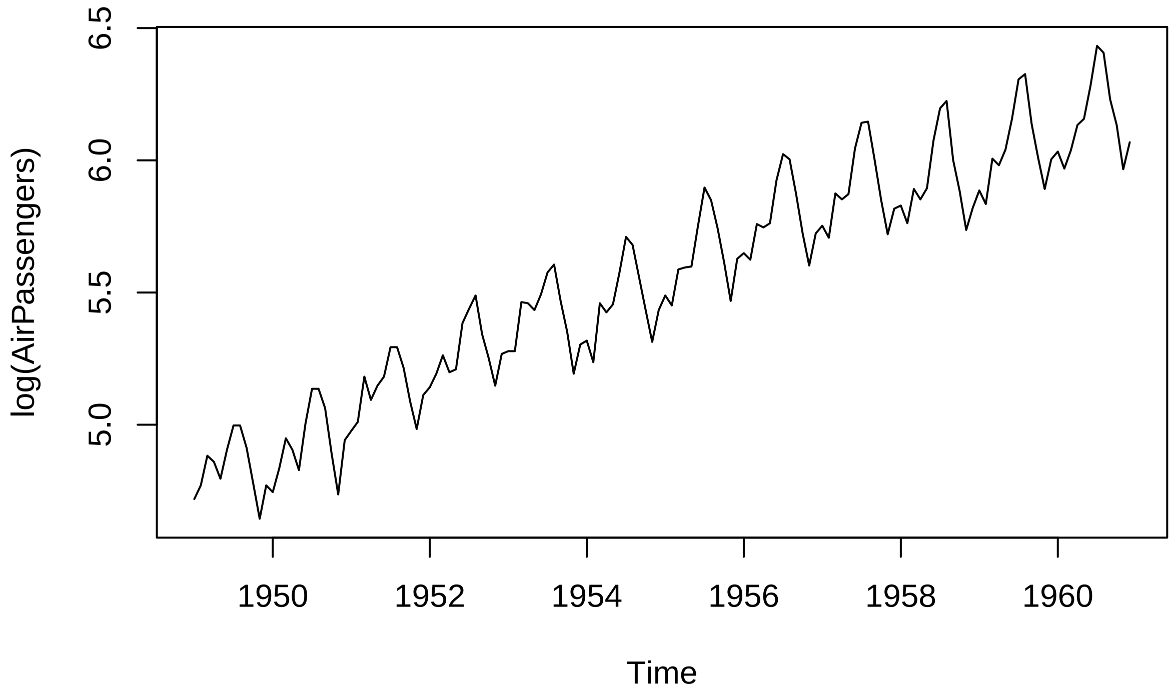
```r
plot(X)
```



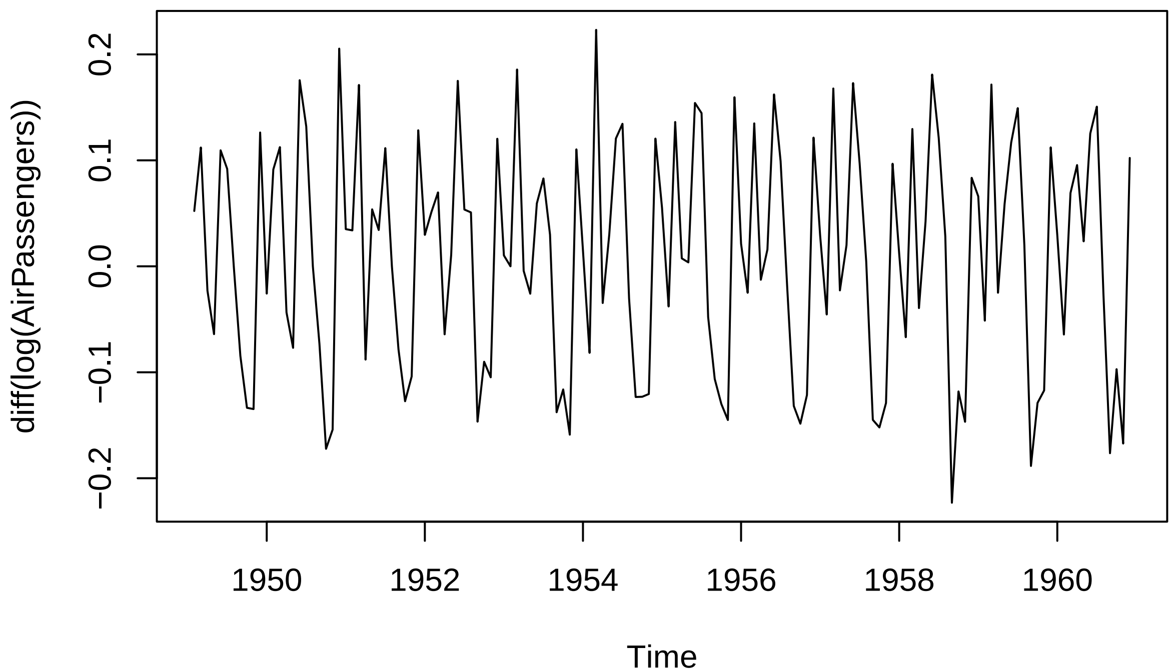

```r
plot(AirPassengers)
```




```r
plot(log(AirPassengers))
```




```
# Logs are calculated first, and then differences
plot(diff(log(AirPassengers)))
```




```
# The complete model
fit1 <- lm(y~.,data=X)
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ ., data = X)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3828 -1.0817  0.3289  1.2419  3.4923
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.94606    2.55986   1.932    0.062 .
## lag1         0.68880    0.12896   5.341 6.74e-06 ***
## lag2        -0.01486    0.04917  -0.302    0.764
## lag3        -0.02849    0.04952  -0.575    0.569
## lag4         0.99860    0.04920  20.297  < 2e-16 ***
## lag5        -0.67931    0.13025  -5.215 9.77e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.965 on 33 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.987,  Adjusted R-squared:  0.985
## F-statistic: 499.8 on 5 and 33 DF,  p-value: < 2.2e-16
```

```r
# The stepwise model
fit2 <- step(fit1)
```

```
## Start:  AIC=58.18
## y ~ lag1 + lag2 + lag3 + lag4 + lag5
##
##        Df Sum of Sq     RSS     AIC
## - lag2  1      0.35  127.79  56.286
## - lag3  1      1.28  128.71  56.567
## <none>               127.44  58.178
## - lag5  1    105.04  232.48  79.624
## - lag1  1    110.17  237.61  80.475
## - lag4  1   1590.97 1718.40 157.638
##
## Step:  AIC=56.29
## y ~ lag1 + lag3 + lag4 + lag5
##
##        Df Sum of Sq     RSS     AIC
## - lag3  1      1.51  129.29  54.743
## <none>               127.79  56.286
## - lag5  1    104.99  232.78  77.674
## - lag1  1    111.14  238.92  78.691
## - lag4  1   2717.34 2845.12 175.302
##
## Step:  AIC=54.74
## y ~ lag1 + lag4 + lag5
##
##        Df Sum of Sq     RSS     AIC
## <none>               129.29  54.743
## - lag1  1    110.09  239.39  76.766
## - lag5  1    116.20  245.50  77.749
## - lag4  1   2910.88 3040.17 175.888
```

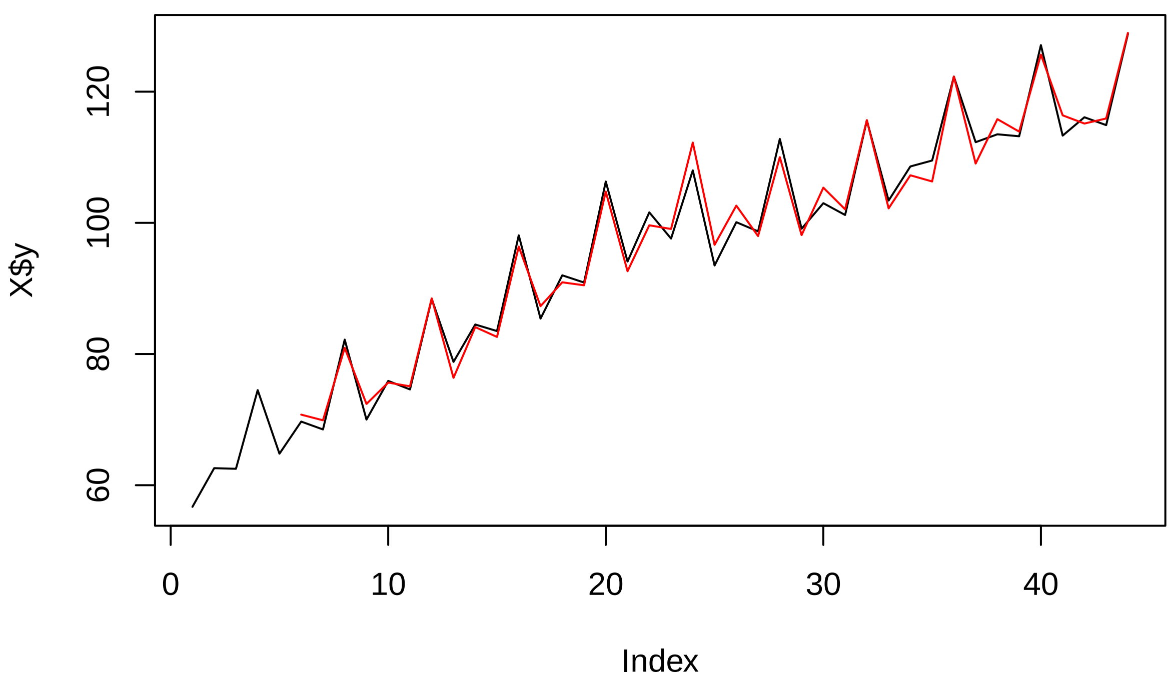
```r
summary(fit2)
```

```
##
## Call:
## lm(formula = y ~ lag1 + lag4 + lag5, data = X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2420 -1.2261  0.2523  1.3036  3.2640
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.5873     2.4497   1.873   0.0695 .
## lag1          0.6783     0.1242   5.459 3.99e-06 ***
## lag4          0.9824     0.0350  28.071  < 2e-16 ***
## lag5         -0.6927     0.1235  -5.609 2.53e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.922 on 35 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.9868, Adjusted R-squared:  0.9856
## F-statistic: 870.6 on 3 and 35 DF,  p-value: < 2.2e-16
```

```r
c(AIC(fit1),AIC(fit2))
```

```
## [1] 170.8552 167.4197
```

```r
# In-sample fit:
plot(X$y,type="l")
frc <- predict(fit2,X)
lines(frc,col="red")
```



```r
# I will take the last 5 values (remember: up to lag 5)
Xnew <- array(tail(y.trn,5),c(1,5))
colnames(Xnew) <- paste0("lag",5:1) # Note that I invert the order.
# I do that as the last value is lag1 and 5 values ago is lag 5.
# R is smart enough to pick the right element, just by looking at the names.
Xnew <- as.data.frame(Xnew)
Xnew
```

```
##    lag5  lag4  lag3  lag2  lag1
## 1 127.1 113.3 116.1 114.9 128.8
```

```r
predict(fit2,Xnew)
```

```
##        1
## 115.2038
```

```r
frc1 <- array(NA,c(8,1)) # 8 because the test set is 8 periods
```

```r
Xnew <- tail(y.trn,5)
Xnew <- Xnew[5:1]
Xnew
```

```
## [1] 128.8 114.9 116.1 113.3 127.1
```

```r
formula(fit2)
```

```
## y ~ lag1 + lag4 + lag5
```

```r
Xnew <- c(Xnew, frc1)
Xnew
```

```
##  [1] 128.8 114.9 116.1 113.3 127.1    NA    NA    NA    NA    NA    NA    NA
## [13]    NA
```

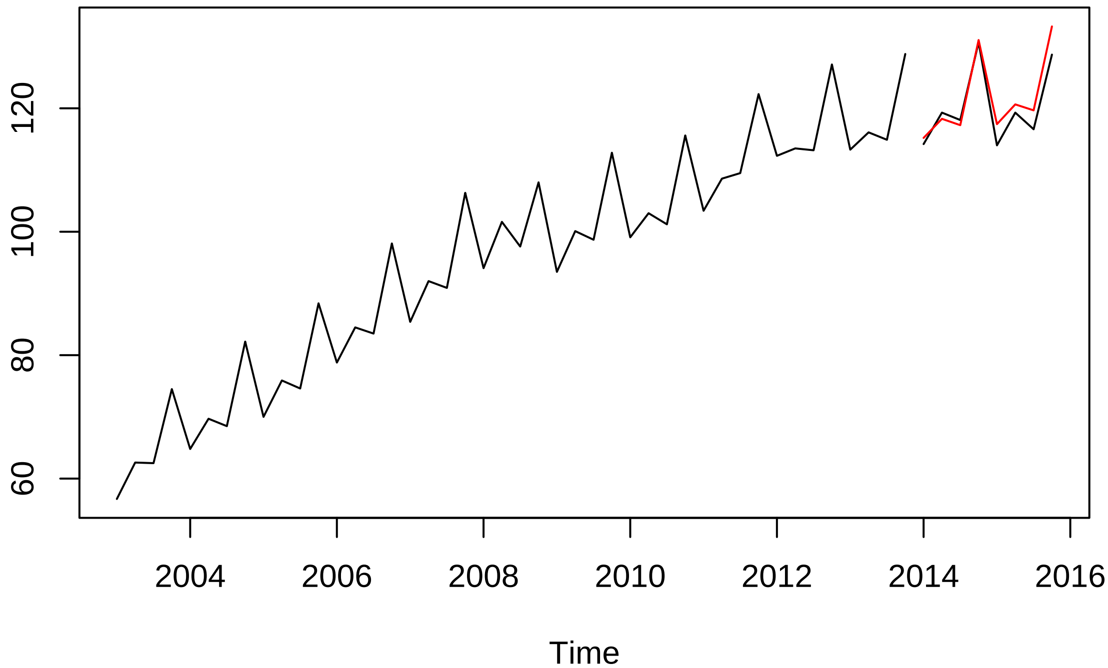
```r
frc1<-array(NA,c(8,1))
 for(i in 1:8){
 #For the Xnew we use the last five observations as before
 Xnew<-tail(y.trn,5)
 #Add to that the forecasted values
 Xnew<-c(Xnew,frc1)
 #Take the relevant 5 values. The index i helps us to get the right ones
 Xnew<-Xnew[i:(4+i)]
 #If i=1 then this becomes Xnew[1:5].
 #If i=2 then this becomes Xnew[2:6] - just as the example above.
 #Reverse the order
 Xnew<-Xnew[5:1]
 #Make Xnew an array and name the inputs
 Xnew<-array(Xnew,c(1,5))#c(1,5) are the dimensions of the array
 colnames(Xnew)<-paste0("lag",1:5)#I have already reversed the order
 #Convert to data.frame
 Xnew<-as.data.frame(Xnew)
 #Forecast
 frc1[i]<-predict(fit2,Xnew)
 }
 frc1
```

```
##          [,1]
## [1,] 115.2038
```

```
## [2,] 118.2922
## [3,] 117.2685
## [4,] 131.0602
## [5,] 117.4294
## [6,] 120.6364
## [7,] 119.6665
## [8,] 133.2663
```

```r
#Transform to time series, by copying the information from y.tst
frc1<-ts(frc1,frequency=frequency(y.tst),start=start(y.tst))
```

```r
ts.plot(y.trn,y.tst,frc1,col=c("black","black","red"))
```



Seasonality with dummy variables

```r
D <- rep(1:4,11) # Replicate 1:4 11 times
D <- factor(D)
D
```

```
##  [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2
## [39] 3 4 1 2 3 4
## Levels: 1 2 3 4
```

```r
factor(rep(c("Q1","Q2","Q3","Q4"),11))
```

```
##  [1] Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1
## [26] Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
## Levels: Q1 Q2 Q3 Q4
```

```r
X2 <- cbind(X,D)
colnames(X2) <- c(colnames(X2)[1:6],"D")
X2
```

```
##        y  lag1  lag2  lag3  lag4  lag5 D
```

```
## 1   56.7    NA    NA    NA    NA    NA 1
## 2   62.6  56.7    NA    NA    NA    NA 2
## 3   62.5  62.6  56.7    NA    NA    NA 3
## 4   74.5  62.5  62.6  56.7    NA    NA 4
## 5   64.8  74.5  62.5  62.6  56.7    NA 1
## 6   69.7  64.8  74.5  62.5  62.6  56.7 2
## 7   68.5  69.7  64.8  74.5  62.5  62.6 3
## 8   82.2  68.5  69.7  64.8  74.5  62.5 4
## 9   70.0  82.2  68.5  69.7  64.8  74.5 1
## 10  75.9  70.0  82.2  68.5  69.7  64.8 2
## 11  74.6  75.9  70.0  82.2  68.5  69.7 3
## 12  88.4  74.6  75.9  70.0  82.2  68.5 4
## 13  78.8  88.4  74.6  75.9  70.0  82.2 1
## 14  84.5  78.8  88.4  74.6  75.9  70.0 2
## 15  83.5  84.5  78.8  88.4  74.6  75.9 3
## 16  98.1  83.5  84.5  78.8  88.4  74.6 4
## 17  85.4  98.1  83.5  84.5  78.8  88.4 1
## 18  92.0  85.4  98.1  83.5  84.5  78.8 2
## 19  90.9  92.0  85.4  98.1  83.5  84.5 3
## 20 106.3  90.9  92.0  85.4  98.1  83.5 4
## 21  94.1 106.3  90.9  92.0  85.4  98.1 1
## 22 101.6  94.1 106.3  90.9  92.0  85.4 2
## 23  97.6 101.6  94.1 106.3  90.9  92.0 3
## 24 108.0  97.6 101.6  94.1 106.3  90.9 4
## 25  93.5 108.0  97.6 101.6  94.1 106.3 1
## 26 100.1  93.5 108.0  97.6 101.6  94.1 2
## 27  98.7 100.1  93.5 108.0  97.6 101.6 3
## 28 112.8  98.7 100.1  93.5 108.0  97.6 4
## 29  99.1 112.8  98.7 100.1  93.5 108.0 1
## 30 103.0  99.1 112.8  98.7 100.1  93.5 2
## 31 101.2 103.0  99.1 112.8  98.7 100.1 3
## 32 115.6 101.2 103.0  99.1 112.8  98.7 4
## 33 103.4 115.6 101.2 103.0  99.1 112.8 1
## 34 108.6 103.4 115.6 101.2 103.0  99.1 2
## 35 109.5 108.6 103.4 115.6 101.2 103.0 3
## 36 122.3 109.5 108.6 103.4 115.6 101.2 4
## 37 112.3 122.3 109.5 108.6 103.4 115.6 1
## 38 113.5 112.3 122.3 109.5 108.6 103.4 2
## 39 113.2 113.5 112.3 122.3 109.5 108.6 3
## 40 127.1 113.2 113.5 112.3 122.3 109.5 4
## 41 113.3 127.1 113.2 113.5 112.3 122.3 1
## 42 116.1 113.3 127.1 113.2 113.5 112.3 2
## 43 114.9 116.1 113.3 127.1 113.2 113.5 3
## 44 128.8 114.9 116.1 113.3 127.1 113.2 4
```

```r
fit3 <- lm(y~.,data=X2)
summary(fit3)
```

```
##
## Call:
## lm(formula = y ~ ., data = X2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5499 -0.6431 -0.0694  0.7327  2.7217
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.84018    3.64761  -1.875 0.070522 .
## lag1         0.89964    0.18055   4.983 2.45e-05 ***
## lag2         0.09947    0.22994   0.433 0.668390
## lag3        -0.25396    0.22740  -1.117 0.272946
## lag4         0.23654    0.22898   1.033 0.309838
## lag5        -0.01125    0.17798  -0.063 0.950009
## D2          13.01788    5.16637   2.520 0.017302 *
## D3          12.21078    3.51534   3.474 0.001584 **
## D4          20.25475    5.12283   3.954 0.000433 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.567 on 30 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.9925, Adjusted R-squared:  0.9905
## F-statistic: 494.3 on 8 and 30 DF,  p-value: < 2.2e-16
```

```r
# Find NA in X2
idx <- is.na(X2)
# The result is logical TRUE/FALSE values
idx[1:10,]
```

```
##           y  lag1  lag2  lag3  lag4  lag5     D
##  [1,] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [2,] FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
##  [3,] FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE
##  [4,] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
##  [5,] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##  [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [7,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [8,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [9,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [10,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
idx <- rowSums(idx)
idx
```

```
##  [1] 5 4 3 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [39] 0 0 0 0 0 0
```

```r
idx <- idx == 0
idx
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [37]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```r
fit_temp<-lm(y~.,data=X2[idx,])
#fit_temp is the same as fit3, without the first NA part
fit4<-step(fit_temp)
```

```
## Start:  AIC=42.78
## y ~ lag1 + lag2 + lag3 + lag4 + lag5 + D
```

```
##
##        Df Sum of Sq     RSS    AIC
## - lag5  1     0.010  73.634 40.786
## - lag2  1     0.459  74.083 41.024
## - lag4  1     2.619  76.243 42.144
## - lag3  1     3.061  76.685 42.370
## <none>               73.624 42.781
## - D     3    53.812 127.436 58.178
## - lag1  1    60.931 134.555 64.298
##
## Step:  AIC=40.79
## y ~ lag1 + lag2 + lag3 + lag4 + D
##
##        Df Sum of Sq     RSS    AIC
## - lag2  1     0.507  74.141 39.054
## - lag3  1     3.206  76.840 40.449
## <none>               73.634 40.786
## - lag4  1     4.338  77.972 41.019
## - lag1  1    63.371 137.005 63.002
## - D     3   158.844 232.478 79.624
##
## Step:  AIC=39.05
## y ~ lag1 + lag3 + lag4 + D
##
##        Df Sum of Sq     RSS    AIC
## - lag3  1     2.704  76.845 38.451
## <none>               74.141 39.054
## - lag4  1     4.999  79.140 39.599
## - lag1  1   124.312 198.453 75.453
## - D     3   158.634 232.776 77.674
##
## Step:  AIC=38.45
## y ~ lag1 + lag4 + D
##
##        Df Sum of Sq     RSS    AIC
## - lag4  1     2.343  79.188 37.622
## <none>               76.845 38.451
## - D     3   168.652 245.498 77.749
## - lag1  1   155.276 232.121 79.564
##
## Step:  AIC=37.62
## y ~ lag1 + D
##
##        Df Sum of Sq    RSS     AIC
## <none>                79.2  37.622
## - D     3    3076.3 3155.5 175.340
## - lag1  1    8317.2 8396.4 217.508
```

```r
summary(fit4)
```

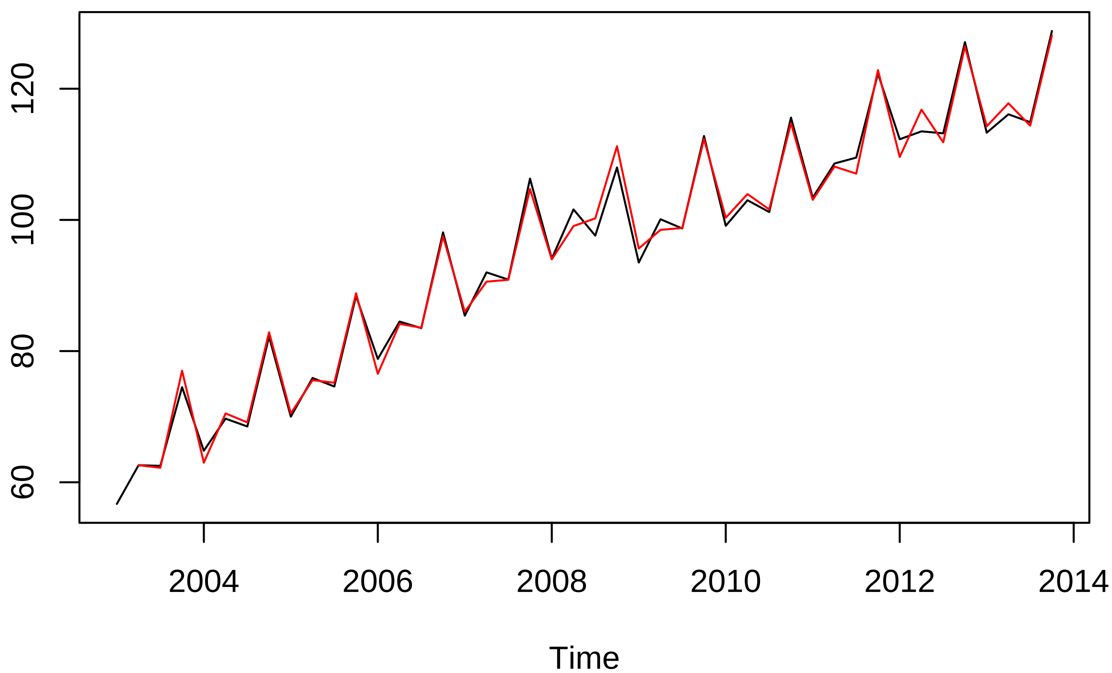
```
##
## Call:
## lm(formula = y ~ lag1 + D, data = X2[idx, ])
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -3.3091 -0.6497  0.0275  0.6699  2.7110
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.65240    1.81455  -5.319 6.61e-06 ***
## lag1         0.97499    0.01632  59.758  < 2e-16 ***
## D2          16.96995    0.74424  22.802  < 2e-16 ***
## D3          10.82574    0.72090  15.017  < 2e-16 ***
## D4          25.73473    0.72586  35.454  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.526 on 34 degrees of freedom
## Multiple R-squared:  0.9919, Adjusted R-squared:  0.9909
## F-statistic:  1041 on 4 and 34 DF,  p-value: < 2.2e-16
```

```
c(AIC(fit2),AIC(fit4))
```

```
## [1] 167.4197 150.2997
```

```
frc <- predict(fit4,X2)
 ts.plot(y.trn,frc,col=c("black","red"))
```



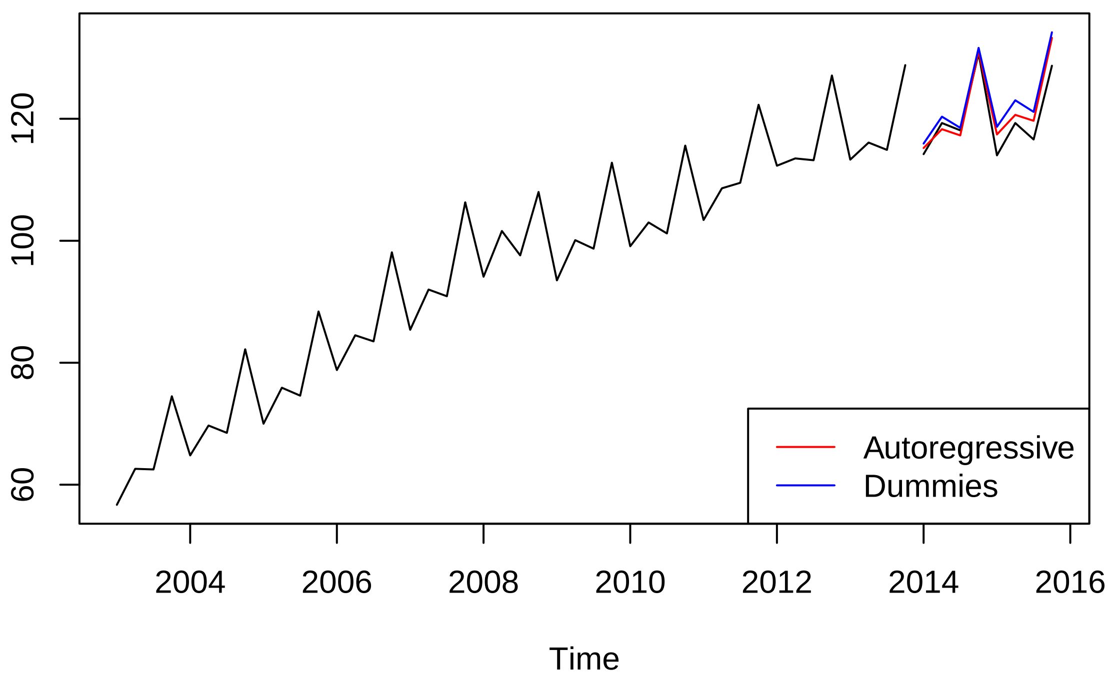

```
#Initialisefrc2tostoretheforecasts
frc2<-array(NA,c(8,1))
for(i in 1:8){
 #Create lags - same as before
 Xnew<-tail(y.trn,5)
 Xnew<-c(Xnew,frc2)
 Xnew<-Xnew[i:(4+i)]
 Xnew<-Xnew[5:1]
 Xnew<-array(Xnew,c(1,5))
 colnames(Xnew)<-paste0("lag",1:5)
```

```r
  Xnew<-as.data.frame(Xnew)
  #Xnew contains all the lags
  #Create the value of the dummy
  D<-as.factor(rep(1:4,2)[i])
  #The logic is that I create the dummy for all 8
  #periods and I pick the i th value. I start the
  #dummy from 1 because I know that the first period
  #is quarter 1. I should ammend this otherwise.
  Xnew<-cbind(Xnew,D)
  #Forecast
  frc2[i]<-predict(fit4,Xnew)
}
```

```r
cbind(frc1,frc2)
```

```
##             frc1     frc2
## 2014 Q1 115.2038 115.9265
## 2014 Q2 118.2922 120.3448
## 2014 Q3 117.2685 118.5085
## 2014 Q4 131.0602 131.6271
## 2015 Q1 117.4294 118.6829
## 2015 Q2 120.6364 123.0323
## 2015 Q3 119.6665 121.1288
## 2015 Q4 133.2663 134.1818
```

```r
 # Transform to time series
 frc2 <- ts(frc2,frequency=frequency(y.tst),start=start(y.tst))
 # Plot
 ts.plot(y.trn,y.tst,frc1,frc2,col=c("black","black","red","blue"))
 legend("bottomright",c("Autoregressive","Dummies"),col=c("red","blue"),lty=1)
```



Modelling in differences (handling trends)

```r
X3 <- X
```

```r
 # The function ncol() counts how many columns
 for (i in 1:ncol(X3)){
 X3[,i] <- c(NA,diff(X3[,i]))
 }
print(X3)
```

```
##        y  lag1  lag2  lag3  lag4  lag5
## 1     NA    NA    NA    NA    NA    NA
## 2    5.9    NA    NA    NA    NA    NA
## 3   -0.1   5.9    NA    NA    NA    NA
## 4   12.0  -0.1   5.9    NA    NA    NA
## 5   -9.7  12.0  -0.1   5.9    NA    NA
## 6    4.9  -9.7  12.0  -0.1   5.9    NA
## 7   -1.2   4.9  -9.7  12.0  -0.1   5.9
## 8   13.7  -1.2   4.9  -9.7  12.0  -0.1
## 9  -12.2  13.7  -1.2   4.9  -9.7  12.0
## 10   5.9 -12.2  13.7  -1.2   4.9  -9.7
## 11  -1.3   5.9 -12.2  13.7  -1.2   4.9
## 12  13.8  -1.3   5.9 -12.2  13.7  -1.2
## 13  -9.6  13.8  -1.3   5.9 -12.2  13.7
## 14   5.7  -9.6  13.8  -1.3   5.9 -12.2
## 15  -1.0   5.7  -9.6  13.8  -1.3   5.9
## 16  14.6  -1.0   5.7  -9.6  13.8  -1.3
## 17 -12.7  14.6  -1.0   5.7  -9.6  13.8
## 18   6.6 -12.7  14.6  -1.0   5.7  -9.6
## 19  -1.1   6.6 -12.7  14.6  -1.0   5.7
## 20  15.4  -1.1   6.6 -12.7  14.6  -1.0
## 21 -12.2  15.4  -1.1   6.6 -12.7  14.6
## 22   7.5 -12.2  15.4  -1.1   6.6 -12.7
## 23  -4.0   7.5 -12.2  15.4  -1.1   6.6
## 24  10.4  -4.0   7.5 -12.2  15.4  -1.1
## 25 -14.5  10.4  -4.0   7.5 -12.2  15.4
## 26   6.6 -14.5  10.4  -4.0   7.5 -12.2
## 27  -1.4   6.6 -14.5  10.4  -4.0   7.5
## 28  14.1  -1.4   6.6 -14.5  10.4  -4.0
## 29 -13.7  14.1  -1.4   6.6 -14.5  10.4
## 30   3.9 -13.7  14.1  -1.4   6.6 -14.5
## 31  -1.8   3.9 -13.7  14.1  -1.4   6.6
## 32  14.4  -1.8   3.9 -13.7  14.1  -1.4
## 33 -12.2  14.4  -1.8   3.9 -13.7  14.1
## 34   5.2 -12.2  14.4  -1.8   3.9 -13.7
## 35   0.9   5.2 -12.2  14.4  -1.8   3.9
## 36  12.8   0.9   5.2 -12.2  14.4  -1.8
## 37 -10.0  12.8   0.9   5.2 -12.2  14.4
## 38   1.2 -10.0  12.8   0.9   5.2 -12.2
## 39  -0.3   1.2 -10.0  12.8   0.9   5.2
## 40  13.9  -0.3   1.2 -10.0  12.8   0.9
## 41 -13.8  13.9  -0.3   1.2 -10.0  12.8
## 42   2.8 -13.8  13.9  -0.3   1.2 -10.0
## 43  -1.2   2.8 -13.8  13.9  -0.3   1.2
```

```
## 44  13.9  -1.2   2.8 -13.8  13.9  -0.3
```

```r
summary(lm(y~.,X3))
```

```
##
## Call:
## lm(formula = y ~ ., data = X3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1629 -1.5089  0.3572  1.3891  2.8476
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3341     0.7653   1.743   0.0909 .
## lag1         -0.1118     0.1754  -0.638   0.5282
## lag2         -0.2588     0.1271  -2.036   0.0501 .
## lag3         -0.2716     0.1269  -2.141   0.0400 *
## lag4          0.7300     0.1313   5.560 3.89e-06 ***
## lag5         -0.1508     0.1818  -0.829   0.4130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.045 on 32 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.9615, Adjusted R-squared:  0.9555
## F-statistic:   160 on 5 and 32 DF,  p-value: < 2.2e-16
```

```r
fit5 <- step(lm(y~.,X3))
```

```
## Start:  AIC=59.85
## y ~ lag1 + lag2 + lag3 + lag4 + lag5
##
##        Df Sum of Sq    RSS    AIC
## - lag1  1     1.702 135.57 58.332
## - lag5  1     2.878 136.75 58.660
## <none>              133.87 59.852
## - lag2  1    17.344 151.21 62.481
## - lag3  1    19.175 153.04 62.939
## - lag4  1   129.322 263.19 83.541
##
## Step:  AIC=58.33
## y ~ lag2 + lag3 + lag4 + lag5
##
##        Df Sum of Sq    RSS    AIC
## <none>              135.57 58.332
## - lag5  1    15.080 150.65 60.340
## - lag2  1    15.642 151.21 60.481
## - lag3  1    17.488 153.06 60.942
## - lag4  1   158.014 293.58 85.694
```

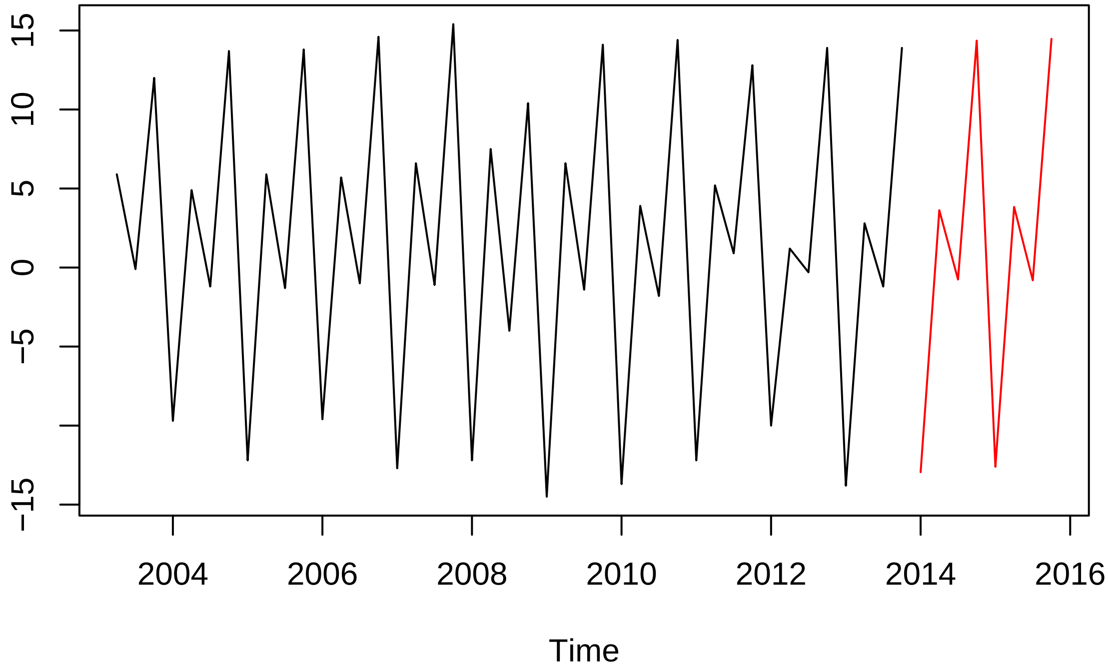
```r
summary(fit5)
```

```
##
## Call:
## lm(formula = y ~ lag2 + lag3 + lag4 + lag5, data = X3)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1763 -1.6582  0.1921  1.4694  2.9309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2013     0.7297   1.646   0.1092
## lag2         -0.2334     0.1196  -1.951   0.0596 .
## lag3         -0.2453     0.1189  -2.063   0.0470 *
## lag4          0.7586     0.1223   6.202 5.33e-07 ***
## lag5         -0.2355     0.1229  -1.916   0.0641 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.027 on 33 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.961,  Adjusted R-squared:  0.9563
## F-statistic: 203.6 on 4 and 33 DF,  p-value: < 2.2e-16
```

```
frc3 <- array(NA,c(8,1))
for (i in 1:8){
 # Calculate the differences of the in-sample data
 y.diff <- diff(y.trn)
 # Create lags- same as before
 Xnew <- tail(y.diff,5)
 Xnew <- c(Xnew,frc3)
 Xnew <- Xnew[i:(4+i)]
 Xnew <- Xnew[5:1]
 Xnew <- array(Xnew, c(1,5))
 colnames(Xnew) <- paste0("lag",1:5)
 Xnew <- as.data.frame(Xnew)
 # Forecast
 frc3[i] <- predict(fit5,Xnew)
}
```

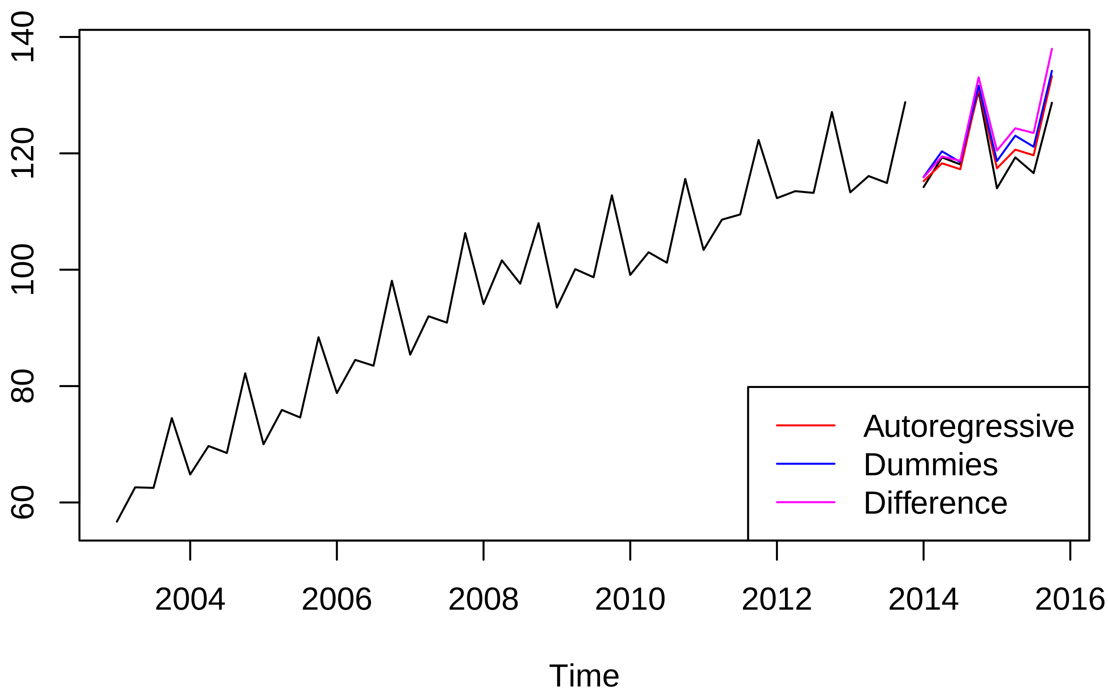
```
 # Transform to time series
 frc3 <- ts(frc3,frequency=frequency(y.tst),start=start(y.tst))
 # Plot
 ts.plot(diff(y.trn),frc3,col=c("black","red"))
```




```
frc3ud <- cumsum(c(tail(y.trn,1),frc3))
# The function cumsum() is the cumulative sum.
```

```
frc3ud <- frc3ud[-1]
```

```
frc3ud <- ts(frc3ud,frequency=frequency(y.tst),start=start(y.tst))
ts.plot(y.trn,y.tst,frc1,frc2,frc3ud,col=c("black","black","red","blue","magenta"))
legend("bottomright",c("Autoregressive","Dummies","Difference"),col=c("red","blue","magenta"),lty=1)
```




```
# Create an array with the actuals replicated three times
# to compare with the three forecasts in one go
```

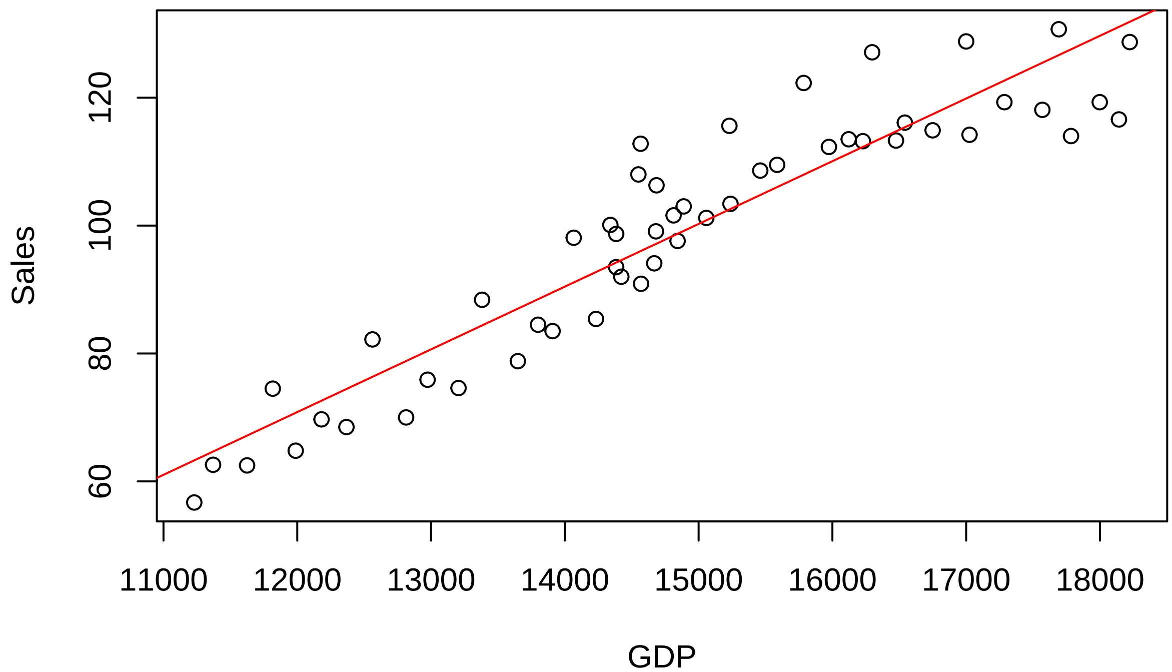
```r
actual <- matrix(rep(y.tst,3),ncol=3)
actual
```

```
##       [,1]  [,2]  [,3]
## [1,] 114.2 114.2 114.2
## [2,] 119.3 119.3 119.3
## [3,] 118.1 118.1 118.1
## [4,] 130.7 130.7 130.7
## [5,] 114.0 114.0 114.0
## [6,] 119.3 119.3 119.3
## [7,] 116.6 116.6 116.6
## [8,] 128.7 128.7 128.7
```

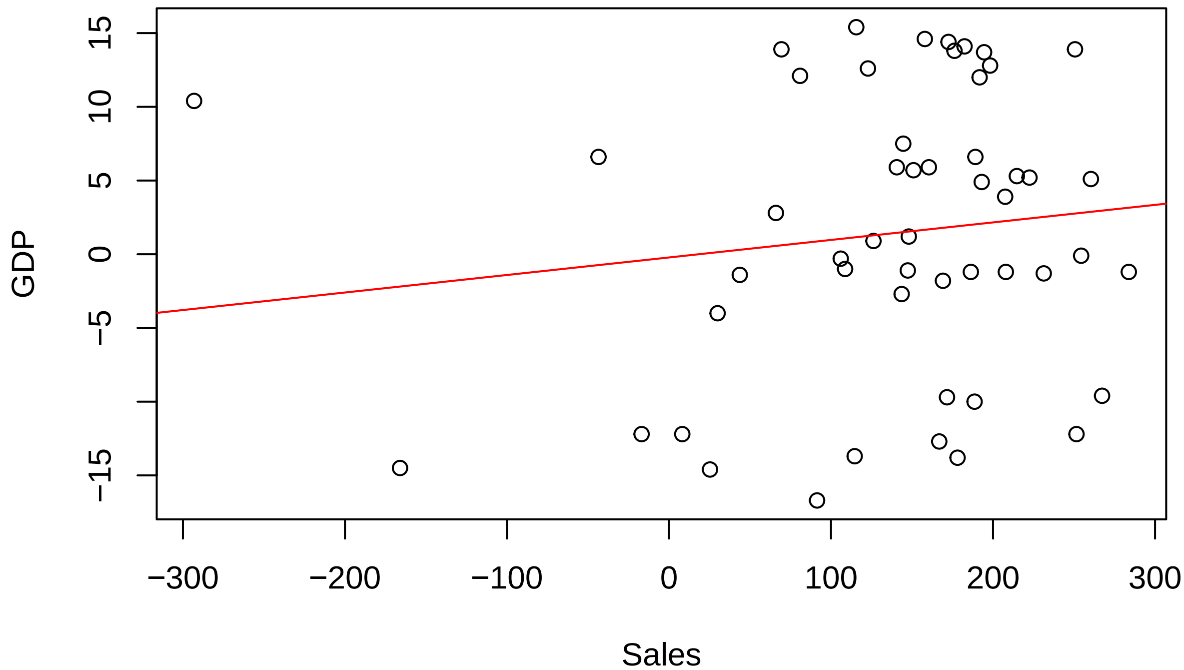
```r
error <- abs(actual- cbind(frc1,frc2,frc3ud))
MAE <- colMeans(error)
MAE
```

```
##     frc1     frc2   frc3ud
## 1.950239 2.816589 4.060461
```

```r
plot(as.vector(x[,2]),as.vector(x[,1]),ylab="Sales",xlab="GDP")
abline(lm(x[,1]~x[,2]),col="red")
```




```r
plot(as.vector(diff(x[,2])),as.vector(diff(x[,1])),xlab="Sales",ylab="GDP")
abline(lm(diff(x[,1])~diff(x[,2])),col="red")
```




```r
# Get gdp in differences after the test set is removed
gdp <- c(NA,diff(x[1:(length(x[,2])-8),2]))
X4 <- cbind(X3,gdp)
fit6 <- step(lm(y~.,X4[-(1:6),])) # Remove NA
```

```
## Start:  AIC=56.83
## y ~ lag1 + lag2 + lag3 + lag4 + lag5 + gdp
##
##        Df Sum of Sq    RSS    AIC
## - lag5  1     0.042 117.35 54.848
## <none>              117.31 56.835
## - lag1  1     8.527 125.84 57.501
## - gdp   1    16.558 133.87 59.852
## - lag2  1    17.762 135.07 60.192
## - lag3  1    20.926 138.24 61.072
## - lag4  1   125.653 242.96 82.502
##
## Step:  AIC=54.85
## y ~ lag1 + lag2 + lag3 + lag4 + gdp
##
##        Df Sum of Sq    RSS    AIC
## <none>              117.35 54.848
## - gdp   1    19.393 136.75 58.660
## - lag1  1    19.458 136.81 58.678
## - lag2  1    20.413 137.76 58.942
## - lag3  1    22.923 140.27 59.629
## - lag4  1   135.254 252.60 81.981
```

```r
summary(fit6)
```

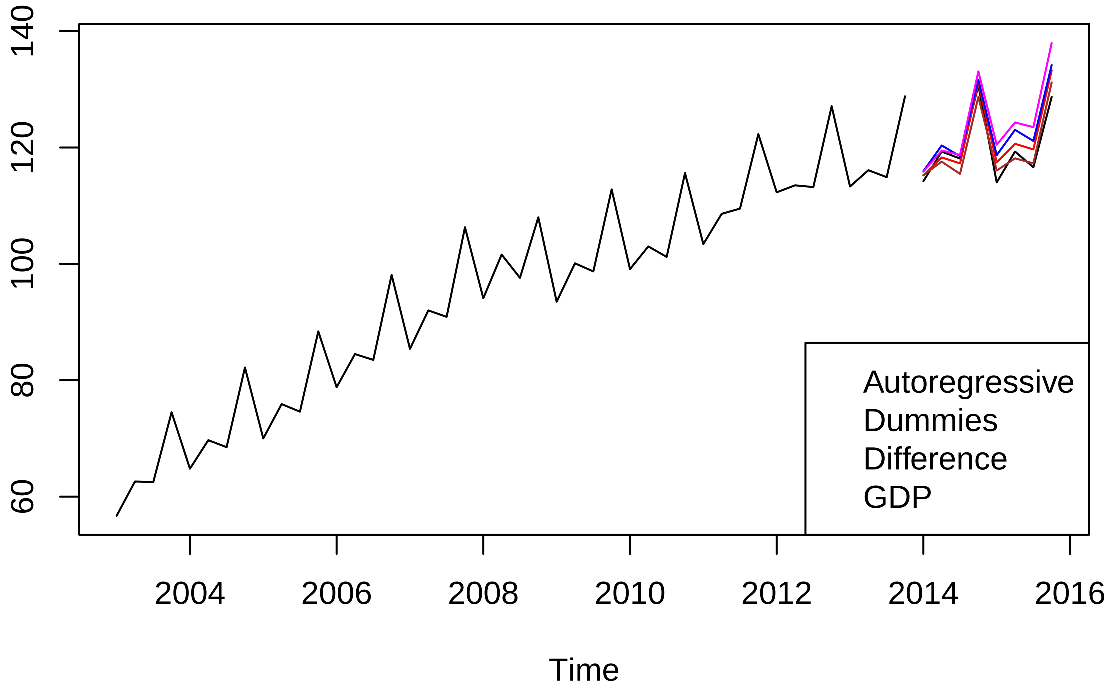
```
##
## Call:
## lm(formula = y ~ lag1 + lag2 + lag3 + lag4 + gdp, data = X4[-(1:6),
##     ])
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4271 -1.2216  0.5818  1.4958  3.0880
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.532350   0.712472   0.747   0.4604
## lag1        -0.261779   0.113647  -2.303   0.0279 *
## lag2        -0.265991   0.112742  -2.359   0.0246 *
## lag3        -0.287376   0.114944  -2.500   0.0177 *
## lag4         0.716369   0.117959   6.073 8.79e-07 ***
## gdp          0.006526   0.002838   2.300   0.0281 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.915 on 32 degrees of freedom
## Multiple R-squared:  0.9663, Adjusted R-squared:  0.961
## F-statistic: 183.4 on 5 and 32 DF,  p-value: < 2.2e-16
```

```
 frc4<-array(NA,c(8,1))
 for(i in 1:8){
 #-- Autoregressions are same as before-
#Calculate the differences of the in-sample data
 y.diff<-diff(y.trn)
 #Create lags - same as before
 Xnew<-tail(y.diff,5)
 Xnew<-c(Xnew,frc3)
 Xnew<-Xnew[i:(4+i)]
 Xnew<-Xnew[5:1]
 #Add differenced gdp information
 #We take the last 9 values,that is test set + 1
 Xgdp<-tail(gdp,9)
 #and calculate differences - this is why we needed the
 #one extra value, which is now removed from the differencing
 Xgdp<-diff(Xgdp)
 #Use only the i th value
 Xgdp<-Xgdp[i]
 #Bind to Xnew
 Xnew<-c(Xnew,Xgdp)
 #Name things
 Xnew<-array(Xnew,c(1,6))
 colnames(Xnew)<-c(paste0("lag",1:5),"gdp")
 Xnew<-as.data.frame(Xnew)
 #Forecast
 frc4[i]<-predict(fit6,Xnew)
}
```

```
frc4ud <- cumsum(frc4) + as.vector(tail(y.trn,1))
```

```
frc4ud <- ts(frc4ud,frequency=frequency(y.tst),start=start(y.tst))
ts.plot(y.trn,y.tst,frc1,frc2,frc3ud,frc4ud,col=c("black","black","red","blue","magenta","brown"))
legend("bottomright",c("Autoregressive","Dummies","Difference","GDP"),col=c("red","blue","magenta","brow
```




```r
c(MAE, mean(abs(y.tst-frc4ud)))
```

```
##     frc1     frc2   frc3ud
## 1.950239 2.816589 4.060461 1.726872
```

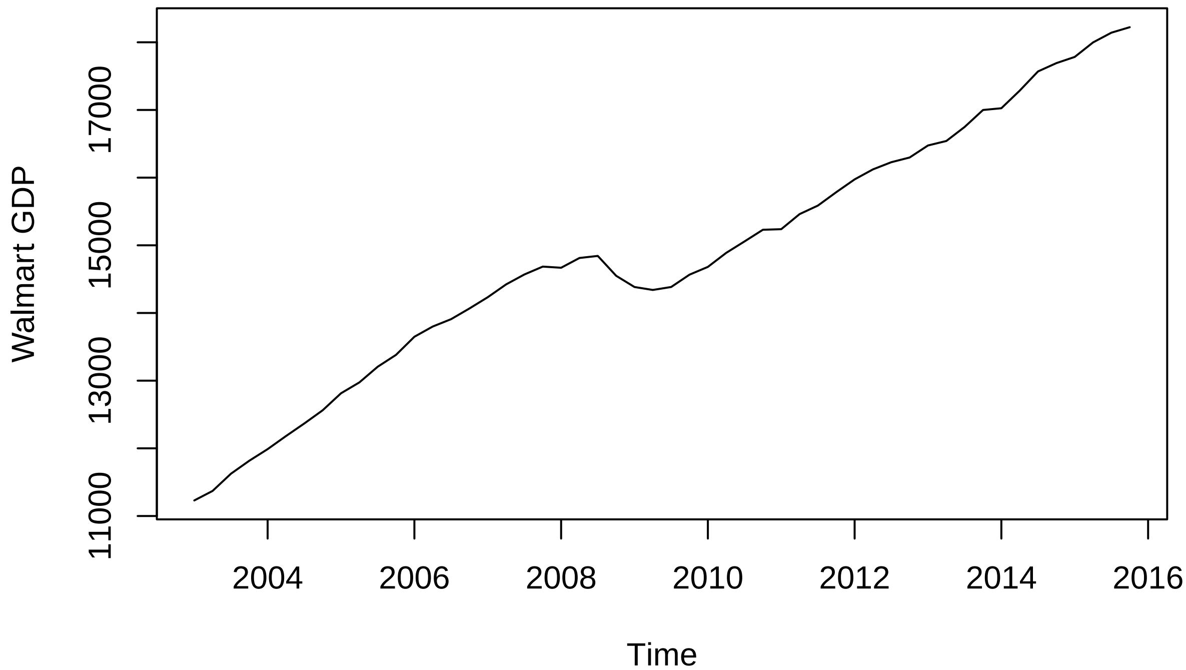
Exercises

Exercise 1

1. Develop a regression using lagged only values of GDP and forecast the next 8 quarters. Attempt the model in differences and on the original data.

Model on the original data:

```r
plot(x[,2],ylab="Walmart GDP")
```




```r
gdp.trn <- window(x[,2], end=c(2013,4))
gdp.tst <- window(x[,2], start=c(2014,1))
```

```r
print(length(gdp.tst))
```

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

```r
n<-length(gdp.trn)
n
```

```
## [1] 44
```

```r
X<-array(NA,c(n,6))
#Construct lags
for(i in 1:6){
 X[i:n,i]<-gdp.trn[1:(n-i+1)]
}

colnames(X)<-c("y",paste0("lag",1:5))

X[1:10,]
```

```
##             y    lag1    lag2    lag3    lag4    lag5
##  [1,] 11230.1      NA      NA      NA      NA      NA
##  [2,] 11370.7 11230.1      NA      NA      NA      NA
##  [3,] 11625.1 11370.7 11230.1      NA      NA      NA
##  [4,] 11816.8 11625.1 11370.7 11230.1      NA      NA
##  [5,] 11988.4 11816.8 11625.1 11370.7 11230.1      NA
##  [6,] 12181.4 11988.4 11816.8 11625.1 11370.7 11230.1
##  [7,] 12367.7 12181.4 11988.4 11816.8 11625.1 11370.7
##  [8,] 12562.2 12367.7 12181.4 11988.4 11816.8 11625.1
##  [9,] 12813.7 12562.2 12367.7 12181.4 11988.4 11816.8
## [10,] 12974.1 12813.7 12562.2 12367.7 12181.4 11988.4
```

```r
X <- as.data.frame(X)
```

```r
fit_lvl <- lm(y~., data = X)
summary(fit_lvl)
```

```
##
## Call:
## lm(formula = y ~ ., data = X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -377.68  -41.61   16.05   62.54  154.14
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 156.84499  198.74890   0.789    0.436
## lag1          1.47945    0.17730   8.344 1.22e-09 ***
## lag2         -0.34124    0.31280  -1.091    0.283
## lag3         -0.15505    0.32184  -0.482    0.633
## lag4         -0.04547    0.32014  -0.142    0.888
## lag5          0.05546    0.17803   0.312    0.757
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98.9 on 33 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.9945, Adjusted R-squared:  0.9936
## F-statistic:  1186 on 5 and 33 DF,  p-value: < 2.2e-16
```

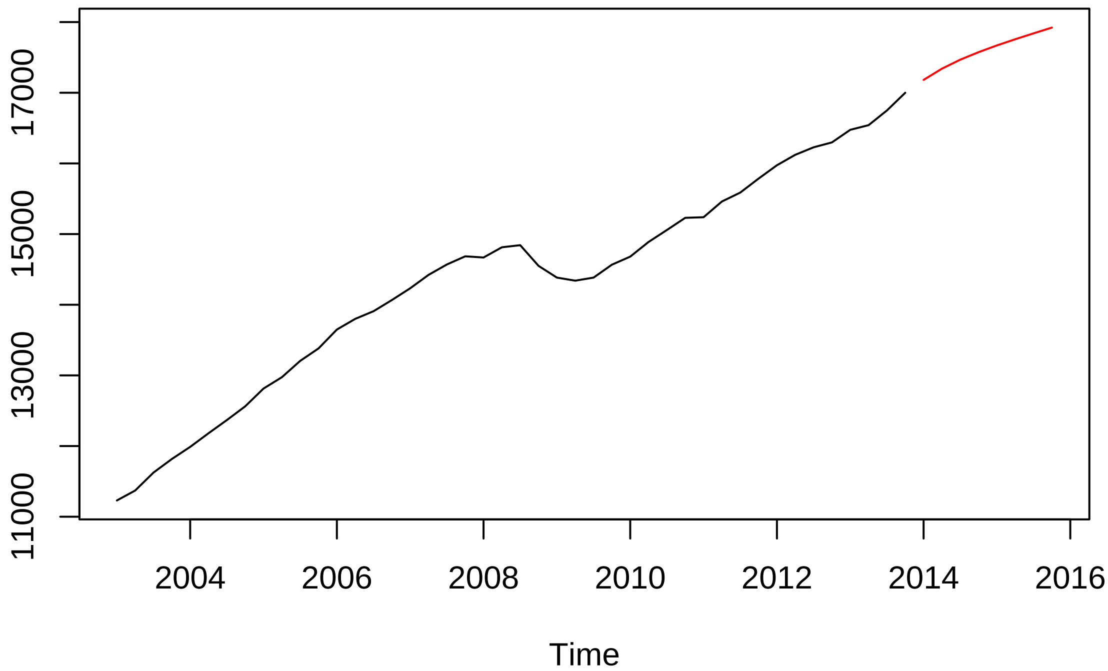
```r
frc_lvl <- array(NA,c(8,1))
 for(i in 1:8){

  Xnew<-tail(gdp.trn,5)
  Xnew<-c(Xnew,frc_lvl)
  Xnew<-Xnew[i:(4+i)]
  Xnew<-Xnew[5:1]
  Xnew<-array(Xnew,c(1,5))

  colnames(Xnew)<-paste0("lag",1:5)
  Xnew<-as.data.frame(Xnew)
  frc_lvl[i]<-predict(fit_lvl,Xnew)
 }
 frc_lvl
```

```
##          [,1]
## [1,] 17181.88
## [2,] 17340.24
## [3,] 17467.78
## [4,] 17574.35
## [5,] 17669.57
## [6,] 17757.19
## [7,] 17840.78
## [8,] 17922.02
```

```r
frc_lvl_ts <-ts(frc_lvl, frequency=frequency(gdp.tst), start=start(gdp.tst))
ts.plot(gdp.trn,frc_lvl_ts,col=c("black","red"))
```



Model in differences:

```r
Xdiff <- X

for (i in 1:ncol(Xdiff)){
  Xdiff[,i] <- c(NA, diff(Xdiff[,i]))
}
print(Xdiff)
```

```
##          y  lag1  lag2  lag3  lag4  lag5
## 1       NA    NA    NA    NA    NA    NA
## 2    140.6    NA    NA    NA    NA    NA
## 3    254.4 140.6    NA    NA    NA    NA
## 4    191.7 254.4 140.6    NA    NA    NA
## 5    171.6 191.7 254.4 140.6    NA    NA
## 6    193.0 171.6 191.7 254.4 140.6    NA
## 7    186.3 193.0 171.6 191.7 254.4 140.6
## 8    194.5 186.3 193.0 171.6 191.7 254.4
## 9    251.5 194.5 186.3 193.0 171.6 191.7
## 10   160.4 251.5 194.5 186.3 193.0 171.6
## 11   231.3 160.4 251.5 194.5 186.3 193.0
## 12   176.2 231.3 160.4 251.5 194.5 186.3
## 13   267.3 176.2 231.3 160.4 251.5 194.5
## 14   150.9 267.3 176.2 231.3 160.4 251.5
## 15   108.7 150.9 267.3 176.2 231.3 160.4
## 16   157.9 108.7 150.9 267.3 176.2 231.3
## 17   166.8 157.9 108.7 150.9 267.3 176.2
## 18   189.1 166.8 157.9 108.7 150.9 267.3
## 19   147.4 189.1 166.8 157.9 108.7 150.9
```

```
## 20  115.6  147.4  189.1  166.8  157.9  108.7
## 21  -16.9  115.6  147.4  189.1  166.8  157.9
## 22  144.6  -16.9  115.6  147.4  189.1  166.8
## 23   30.0  144.6  -16.9  115.6  147.4  189.1
## 24 -293.1   30.0  144.6  -16.9  115.6  147.4
## 25 -166.0 -293.1   30.0  144.6  -16.9  115.6
## 26  -43.5 -166.0 -293.1   30.0  144.6  -16.9
## 27   43.7  -43.5 -166.0 -293.1   30.0  144.6
## 28  182.4   43.7  -43.5 -166.0 -293.1   30.0
## 29  114.6  182.4   43.7  -43.5 -166.0 -293.1
## 30  207.5  114.6  182.4   43.7  -43.5 -166.0
## 31  169.1  207.5  114.6  182.4   43.7  -43.5
## 32  172.5  169.1  207.5  114.6  182.4   43.7
## 33    8.2  172.5  169.1  207.5  114.6  182.4
## 34  222.5    8.2  172.5  169.1  207.5  114.6
## 35  126.2  222.5    8.2  172.5  169.1  207.5
## 36  198.2  126.2  222.5    8.2  172.5  169.1
## 37  188.6  198.2  126.2  222.5    8.2  172.5
## 38  148.0  188.6  198.2  126.2  222.5    8.2
## 39  106.0  148.0  188.6  198.2  126.2  222.5
## 40   69.4  106.0  148.0  188.6  198.2  126.2
## 41  178.1   69.4  106.0  148.0  188.6  198.2
## 42   66.0  178.1   69.4  106.0  148.0  188.6
## 43  207.9   66.0  178.1   69.4  106.0  148.0
## 44  250.6  207.9   66.0  178.1   69.4  106.0
```

```r
fit_diff <- lm(y~., data = Xdiff)
summary(fit_diff)
```

```
##
## Call:
## lm(formula = y ~ ., data = Xdiff)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -377.37  -42.98   14.96   58.08  146.79
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 57.04943   31.07761   1.836  0.07571 .
## lag1         0.49152    0.17863   2.752  0.00968 **
## lag2         0.14861    0.19937   0.745  0.46147
## lag3        -0.01339    0.20630  -0.065  0.94863
## lag4        -0.01328    0.20441  -0.065  0.94860
## lag5        -0.05226    0.18194  -0.287  0.77576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 100.5 on 32 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.3152, Adjusted R-squared:  0.2083
## F-statistic: 2.946 on 5 and 32 DF,  p-value: 0.0268
```

```r
frc_diff <- array(NA,c(8,1))
 for(i in 1:8){
```

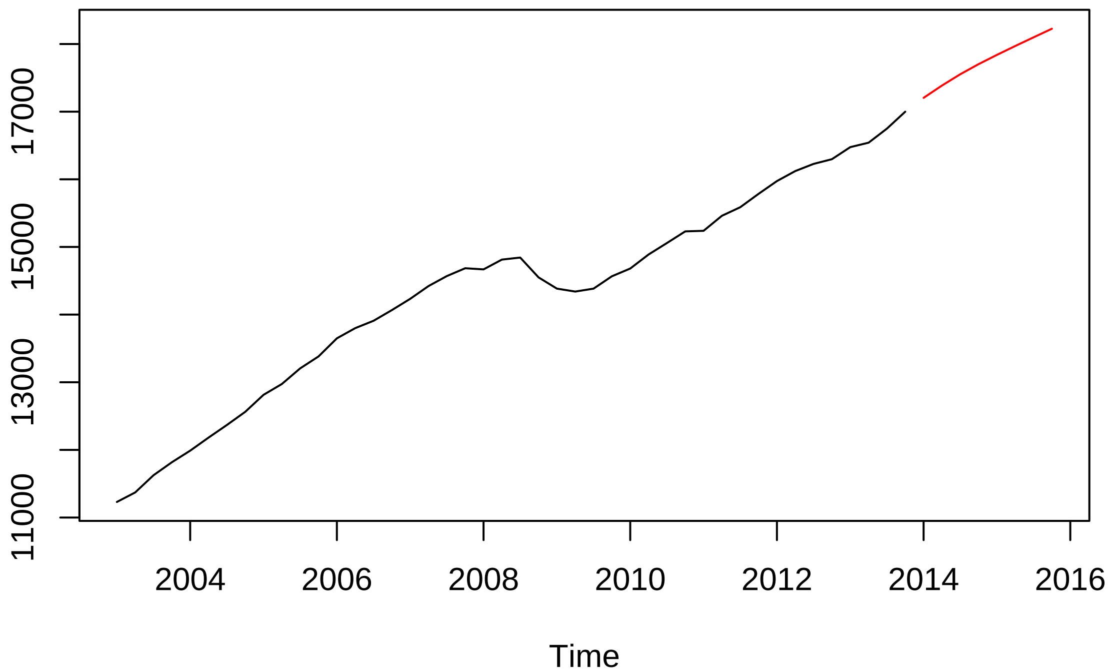
```r
  Xnew<-tail(diff(gdp.trn),5)
  Xnew<-c(Xnew,frc_diff)
  Xnew<-Xnew[i:(4+i)]
  Xnew<-Xnew[5:1]
  Xnew<-array(Xnew,c(1,5))

  colnames(Xnew)<-paste0("lag",1:5)
  Xnew<-as.data.frame(Xnew)

  frc_diff[i]<-predict(fit_diff,Xnew)
 }
 frc_diff
```

```
##          [,1]
## [1,] 204.2453
## [2,] 181.7132
## [3,] 167.1518
## [4,] 149.2835
## [5,] 137.0225
## [6,] 131.2575
## [7,] 128.2121
## [8,] 127.0210
```

```r
frc_diff_leveled <- cumsum(c(as.numeric(tail(gdp.trn,1)), frc_diff))[-1]
frc_diff_ts <-ts(frc_diff_leveled, frequency=frequency(gdp.tst), start=start(gdp.tst))
ts.plot(gdp.trn,frc_diff_ts,col=c("black","red"))
```



Exercise 2

2. Develop an exponential smoothing benchmark. Which model is better? OLS or ETS.

```r
library(forecast)
```

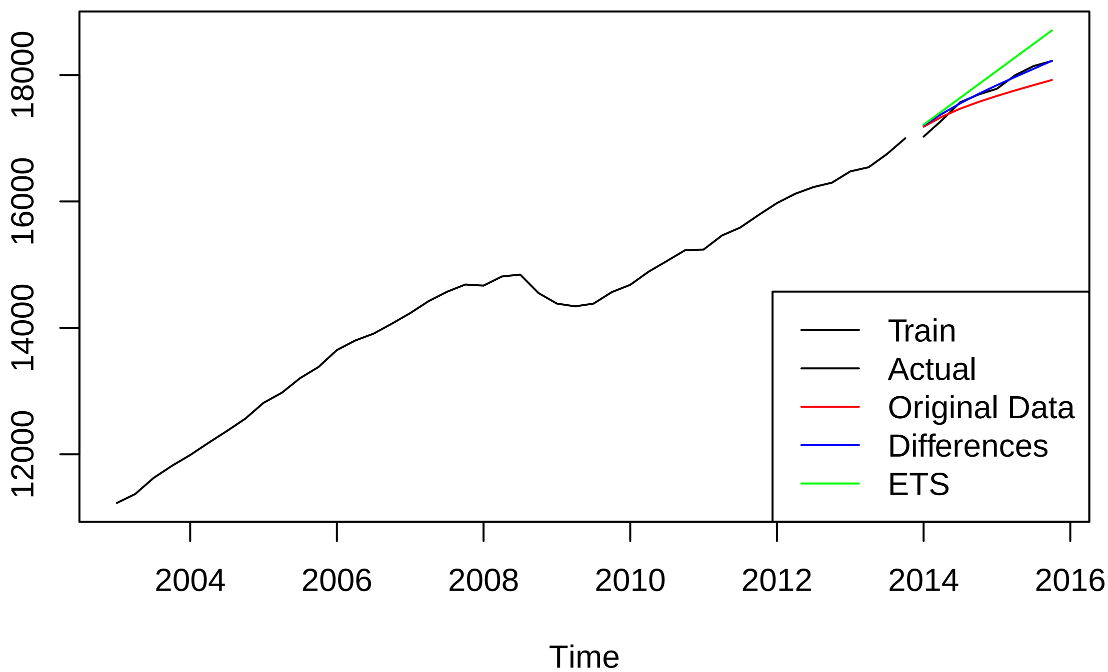
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
ets <- ets(gdp.trn)
ets_frc <- forecast(ets, h=8)
frc_ets_ts <- ets_frc$mean
```

```
#Calculate MAEs
actual <- as.numeric(gdp.tst)
MAE_lvl  <- mean(abs(actual - as.numeric(frc_lvl_ts)))
MAE_diff <- mean(abs(actual - as.numeric(frc_diff_ts)))
MAE_ets  <- mean(abs(actual - as.numeric(frc_ets_ts)))
c(MAE_lvl = MAE_lvl, MAE_diff = MAE_diff, MAE_ets = MAE_ets)
```

```
##   MAE_lvl  MAE_diff   MAE_ets
## 173.47883  54.40635 244.92075
```

```
# Plot forecasts
ts.plot(gdp.trn, gdp.tst, frc_lvl_ts, frc_diff_ts, frc_ets_ts, col=c("black","black","red","blue","greer
legend("bottomright", legend=c("Train","Actual","Original Data","Differences","ETS"), col=c("black","bla
```



Based on the MAE we can see that the OLS regression models functioned better than the ETS model. Especially the model trained on differences performed very well with a MAE of 54.4. The OLS model trained on original data only achieved an MAE of 173.5. The ETS model performed worse with a MAE of 244.9.

For this task we can clearly say that the OLS models performed better.