Regression modelling

Rizny Mubarak

Contents

1. Lab Experiment 1
   1. Advanced regression modelling 1
   2. Construct lags 4
   3. Seasonality with dummy variables 12
   4. Modelling in differences (handling trends) 17
2. Exercises on regression 25
   1. Question 25
      1.1 Find lags 25
      1.2 Construct lags 28
      1.3 Lags - Models 30
      1.4 Trend - Models 33
   2. Question 2 41
      2.1 Exponential smoothing 41
   Answer 43

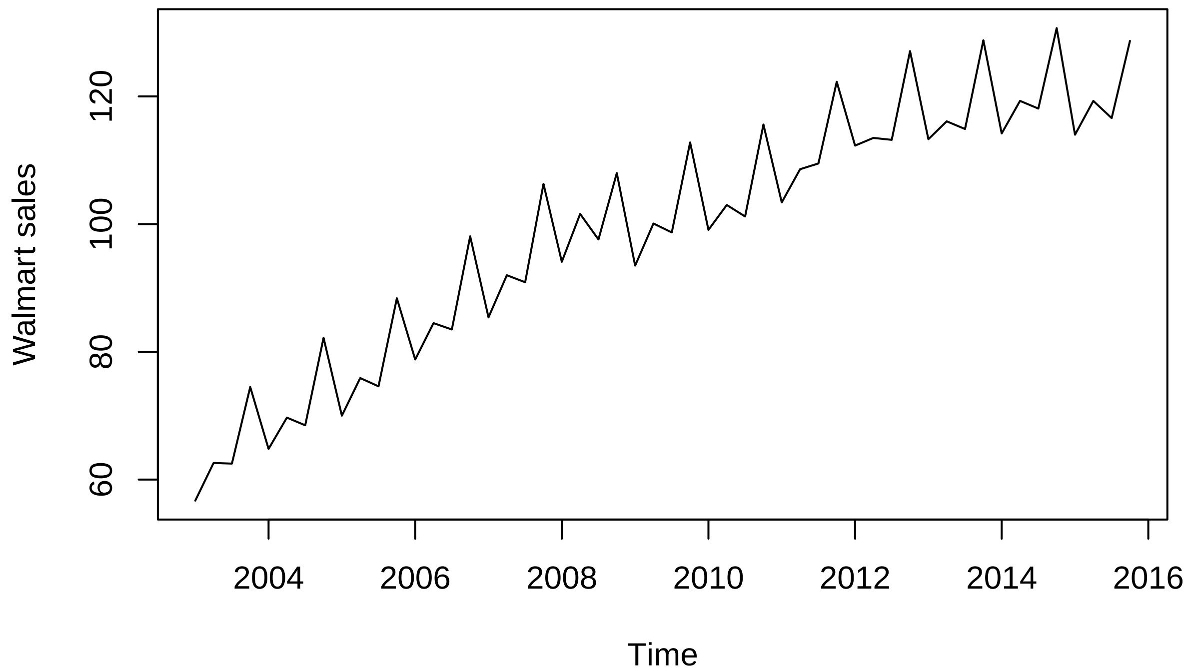
1. Lab Experiment

1. Advanced regression modelling

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

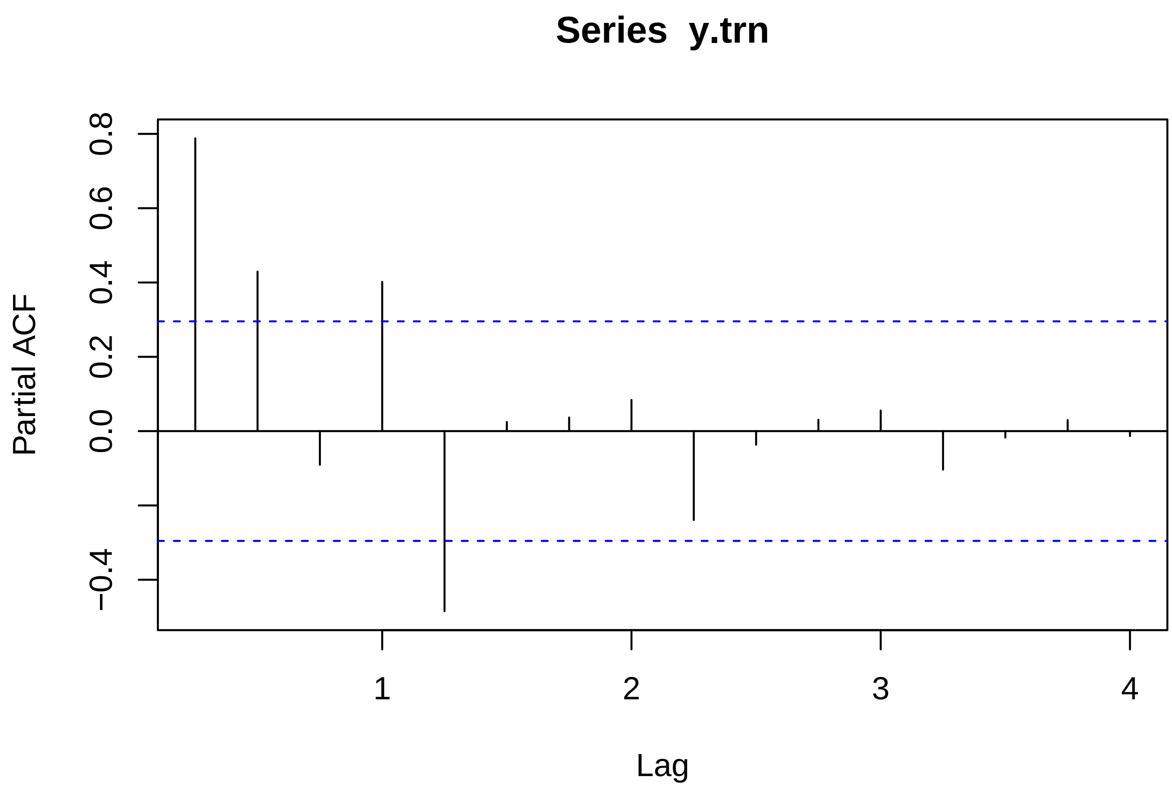
```
##       sales     gdp
##  [1,]  56.7 11230.1
##  [2,]  62.6 11370.7
##  [3,]  62.5 11625.1
##  [4,]  74.5 11816.8
##  [5,]  64.8 11988.4
##  [6,]  69.7 12181.4
##  [7,]  68.5 12367.7
##  [8,]  82.2 12562.2
##  [9,]  70.0 12813.7
## [10,]  75.9 12974.1
```

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

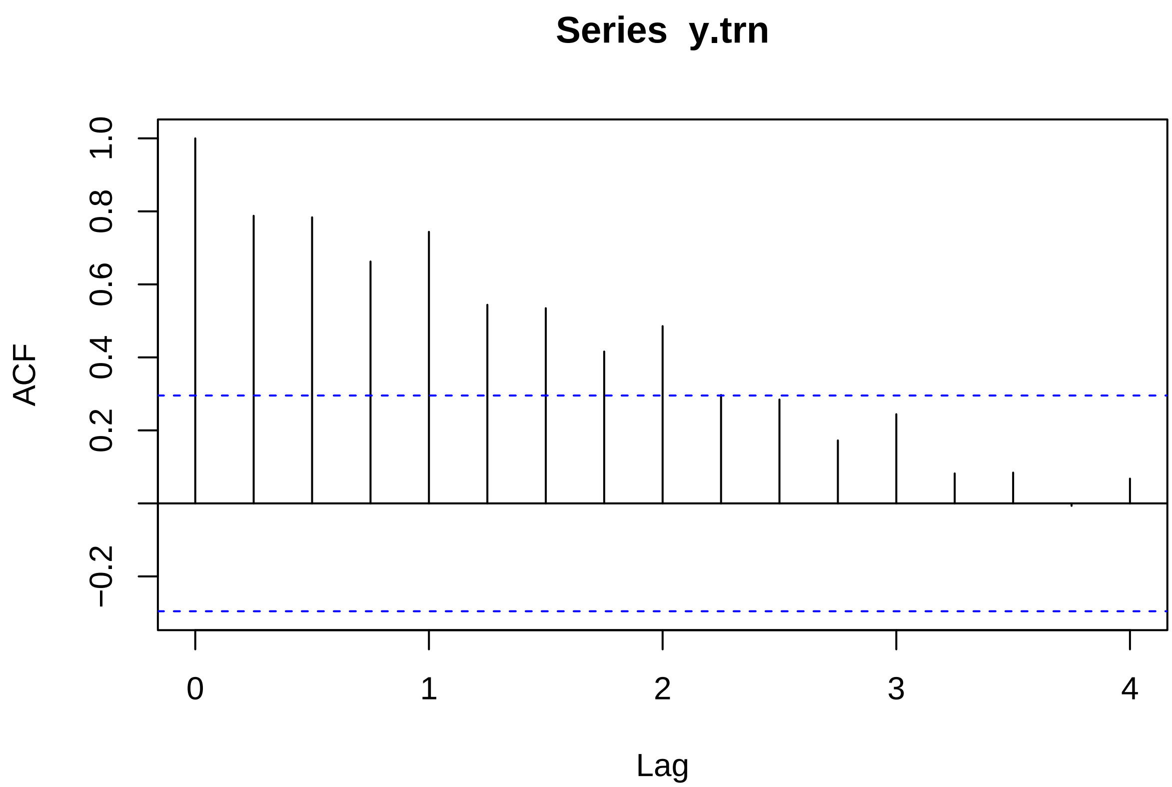



```r
y.trn <- window(x[,1],end=c(2013,4))
y.tst <- window(x[,1],start=c(2014,1))
```

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




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



2. Construct lags

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

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

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

```
for (i in 1:6){
X[i:n,i] <- y.trn[1:(n-i+1)]
}
# Name the columns
colnames(X) <- c("y",paste0("lag",1:5))
X[1:10,]
```

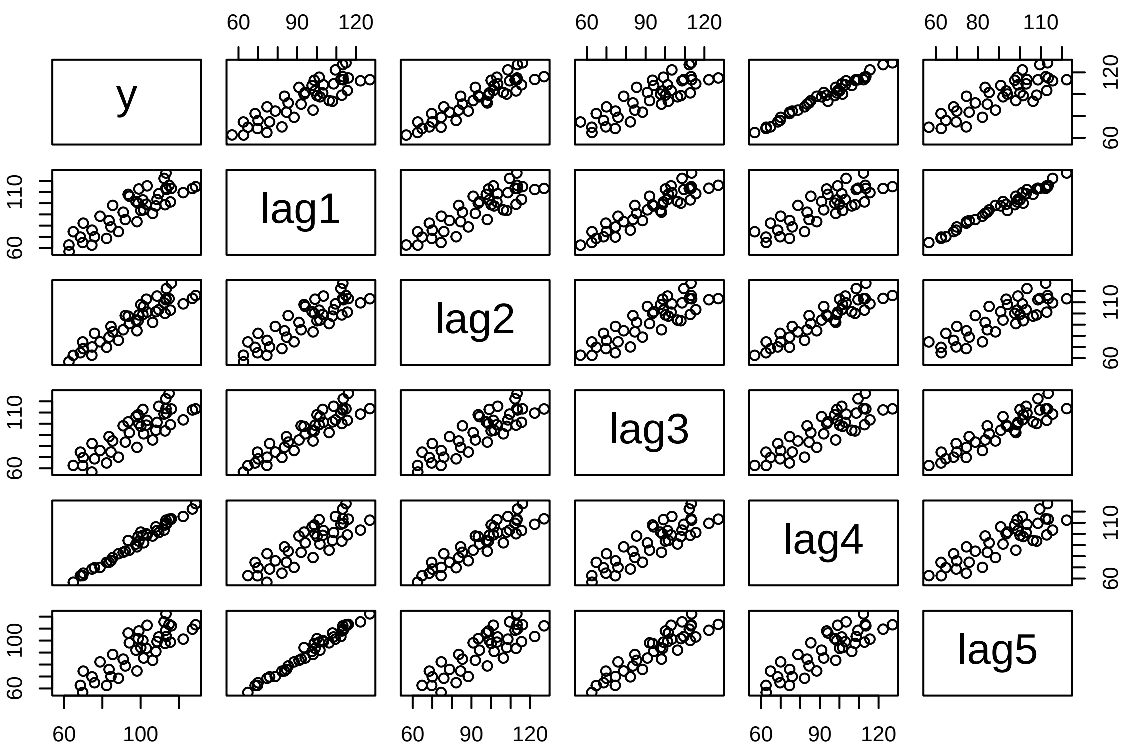
```
##         y lag1 lag2 lag3 lag4 lag5
##  [1,] 56.7   NA   NA   NA   NA   NA
##  [2,] 62.6 56.7   NA   NA   NA   NA
##  [3,] 62.5 62.6 56.7   NA   NA   NA
##  [4,] 74.5 62.5 62.6 56.7   NA   NA
##  [5,] 64.8 74.5 62.5 62.6 56.7   NA
##  [6,] 69.7 64.8 74.5 62.5 62.6 56.7
```

```
##   [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
```

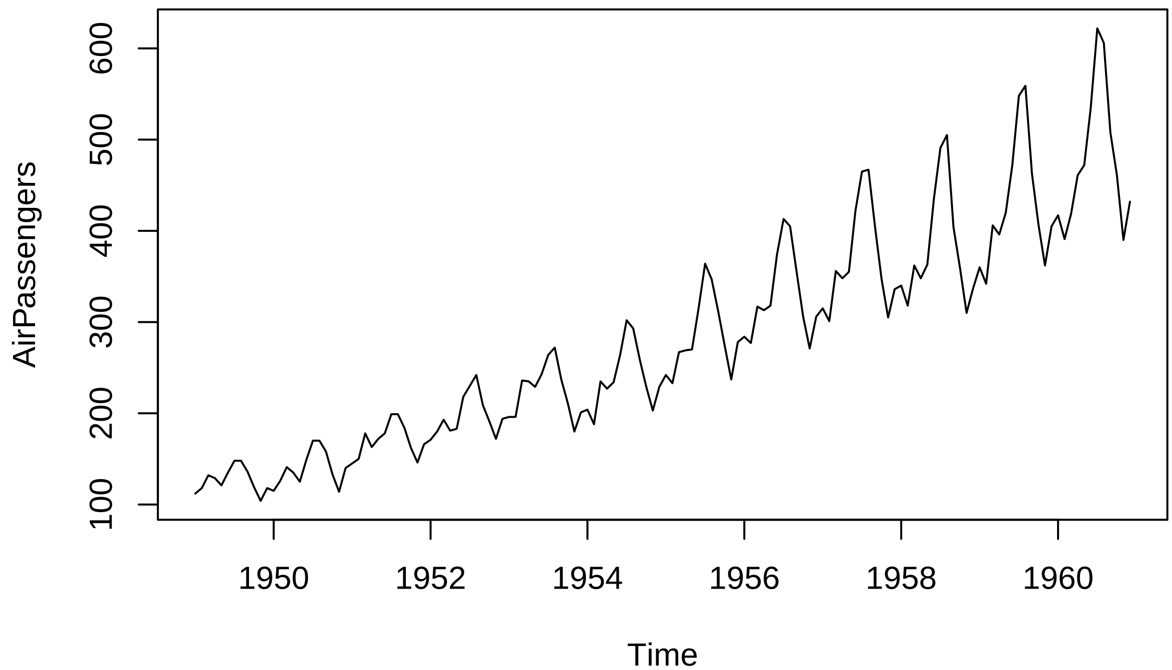
```
X[(n-9):n,]
```

```
##            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
```

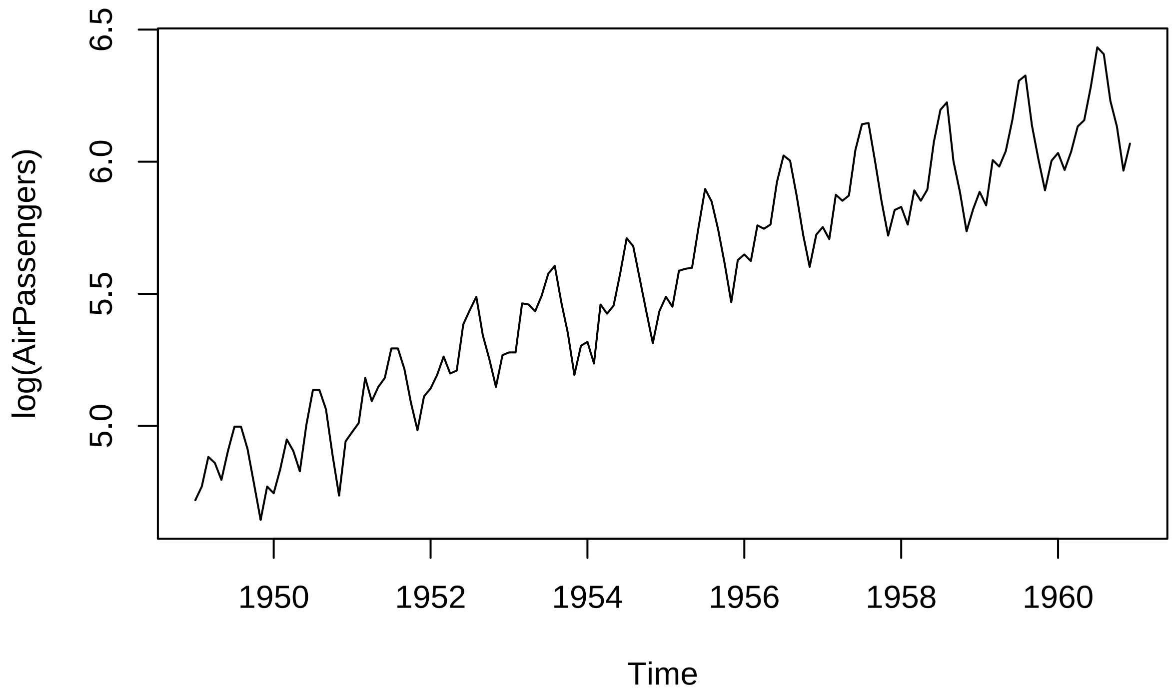
```
X <- as.data.frame(X)
plot(X)
```



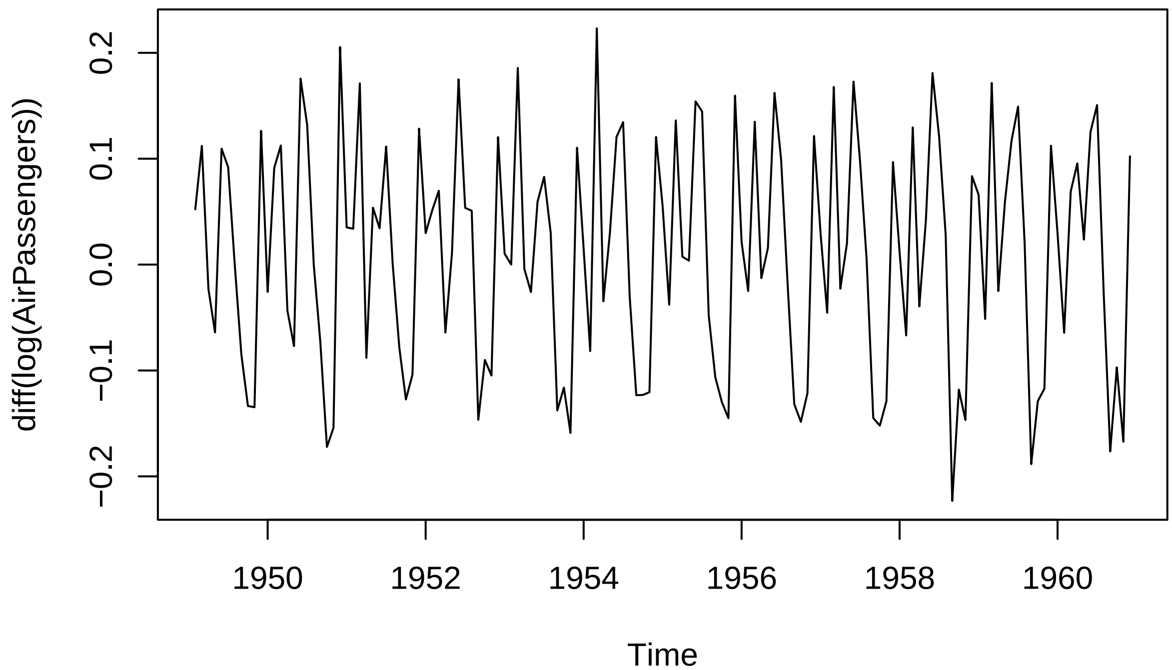

```
plot(AirPassengers)
```




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




```
plot(diff(log(AirPassengers)))
```




```r
# 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
```

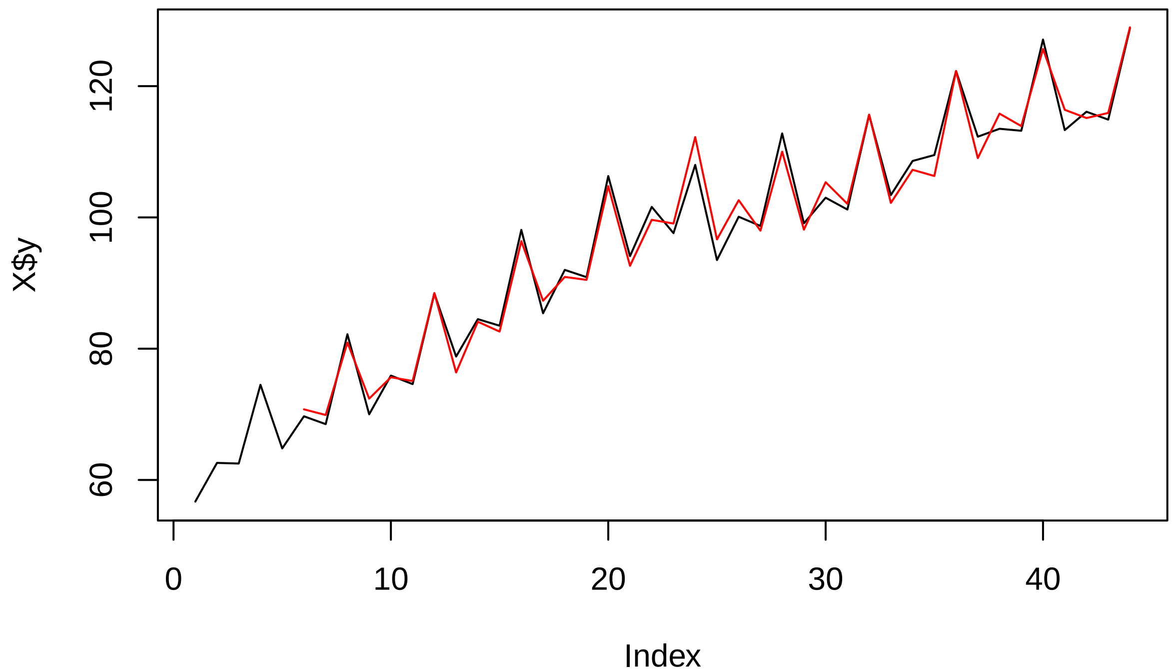
```r
# The stepwise model
fit2 <- step(fit1,trace = 0)
summary(fit2)
```

```
##
## 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")
```




```
Xnew <- array(tail(y.trn,5),c(1,5))
colnames(Xnew) <- paste0("lag",5:1) # Note that I invert the order.
Xnew <- as.data.frame(Xnew)
Xnew
```

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

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

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

```
frc1 <- array(NA,c(8,1))
```

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

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

```
formula(fit2)
```

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

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

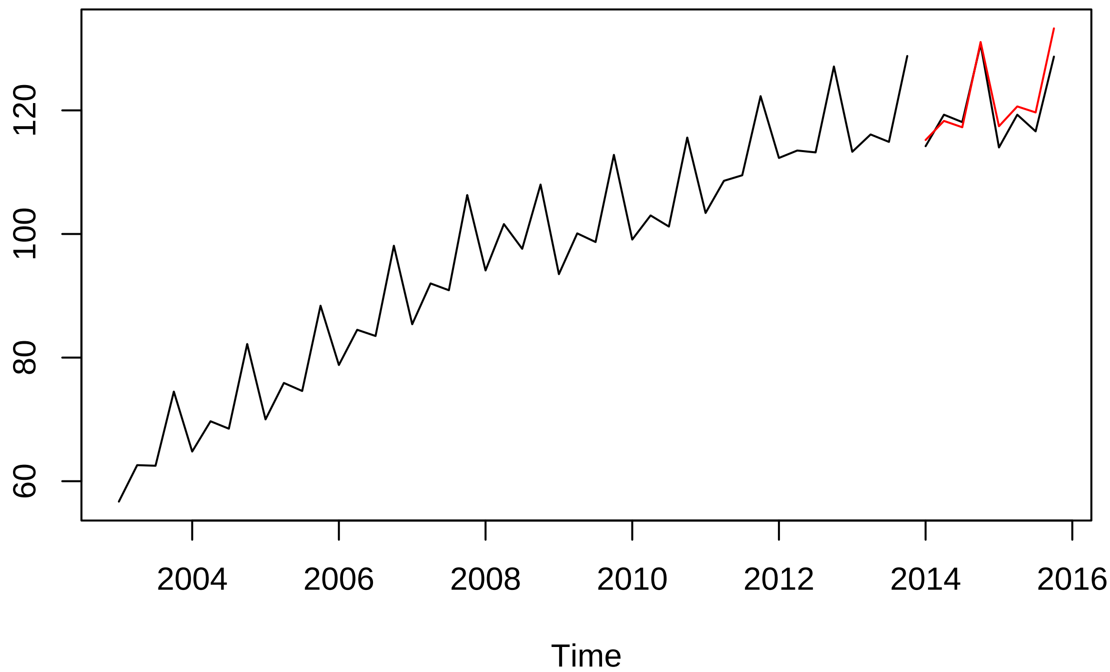
```
##  [1] 128.8 114.9 116.1 113.3 127.1    NA    NA    NA    NA    NA    NA    NA
## [13]    NA
```

```
frc1 <- array(NA,c(8,1))
for (i in 1:8){
Xnew <- tail(y.trn,5)
Xnew <- c(Xnew,frc1)
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
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
```

```
frc1 <- ts(frc1,frequency=frequency(y.tst),start=start(y.tst))
```

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



3. Seasonality with dummy variables

```
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
```

```
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
```

```
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
```

```
# 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
```

```
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
```

```
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
```

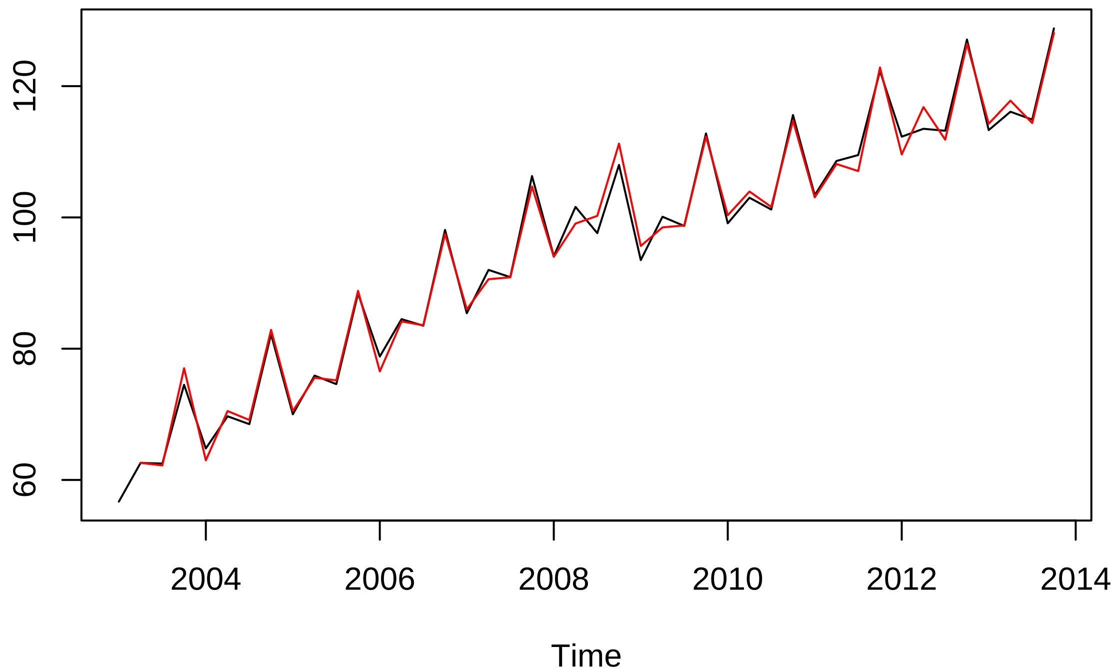
```r
fit_temp <- lm(y~.,data=X2[idx,])
# fit_temp is the same as fit3, without the first NA part
fit4 <- step(fit_temp,trace = 0)
summary(fit4)
```

```
##
## 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
```

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

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

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

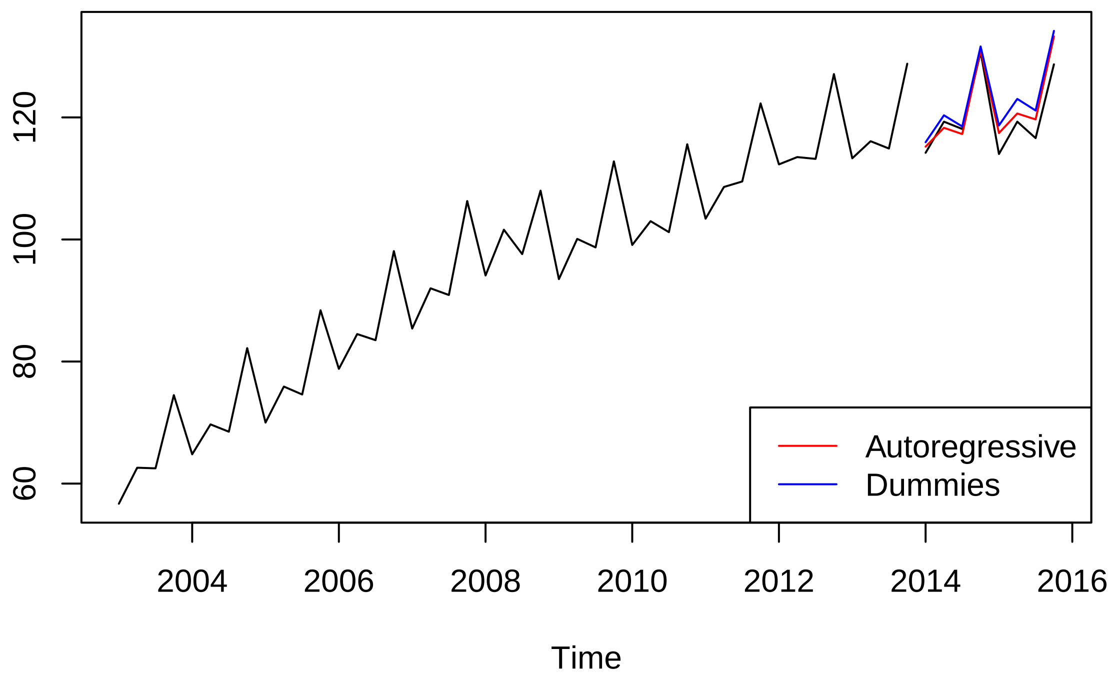



```
frc2 <- array(NA,c(8,1))
for (i in 1:8){
Xnew <- tail(y.trn,5)
Xnew <- c(Xnew,frc2)
Xnew <- Xnew[i:(4+i)]
Xnew <- Xnew[5:1]
Xnew <- array(Xnew, c(1,5))
colnames(Xnew) <- paste0("lag",1:5)
Xnew <- as.data.frame(Xnew)
D <- as.factor(rep(1:4,2)[i])
Xnew <- cbind(Xnew,D)
# Forecast
frc2[i] <- predict(fit4,Xnew)
}
```

```
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
```

```
# 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)
```



4. Modelling in differences (handling trends)

```
X3 <- X
```

```
# The function ncol() counts how many columns
for (i in 1:ncol(X3)){
X3[,i] <- c(NA,diff(X3[,i]))
20
}
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
```

```
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
```

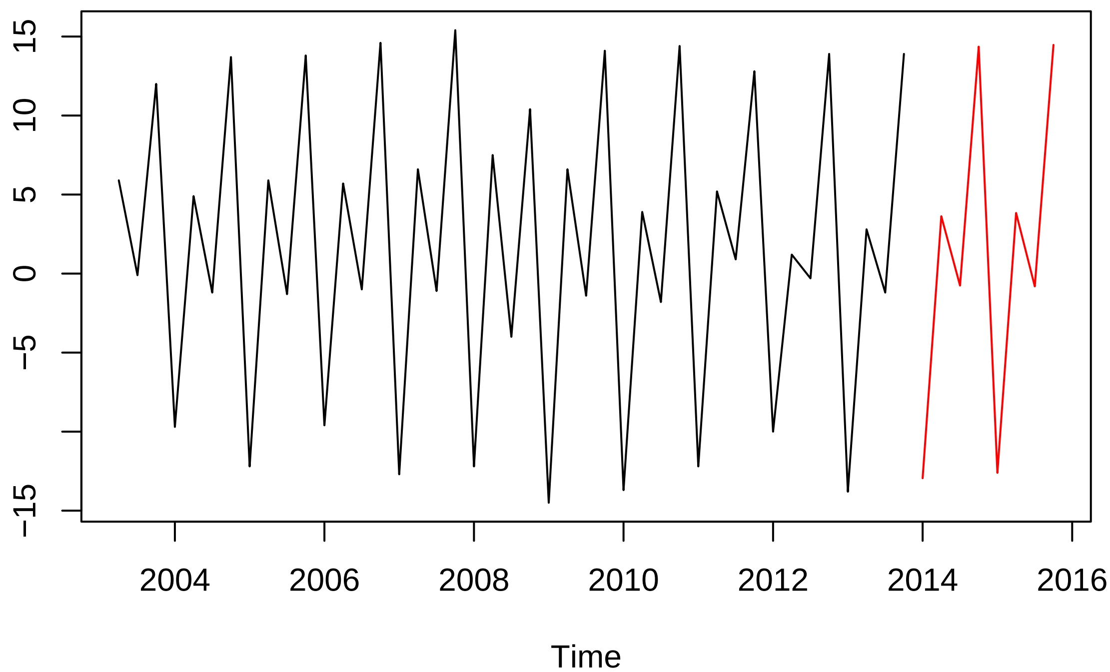
```
## lag2         -0.2588     0.1271  -2.036   0.0501 .
## lag3         -0.2716     0.1269  -2.141   0.0400 *
## lag4          0.7300     0.1313   5.560 3.89e-06 ***
## lag5         -0.1508     0.1818  -0.829   0.4130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual 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), trace = 0)
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
```

```r
frc3 <- array(NA,c(8,1))
for (i in 1:8){
y.diff <- diff(y.trn)
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)
}
```

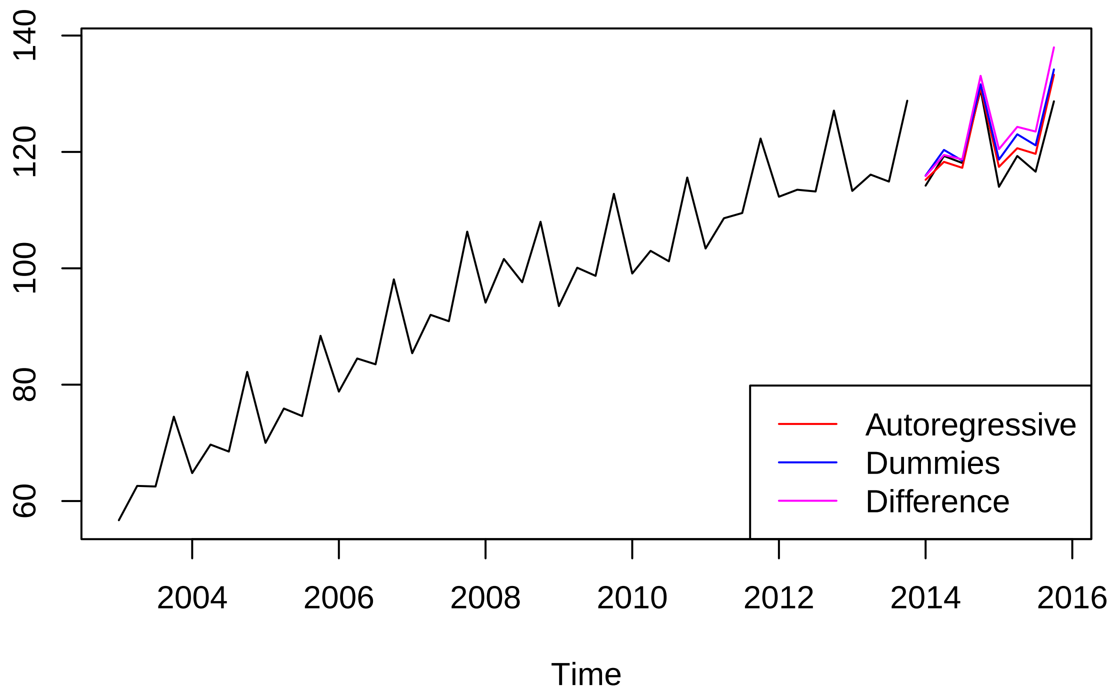
```r
# Transform to time series
frc3 <- ts(frc3,frequency=frequency(y.tst),start=start(y.tst))
# Plot
ts.plot(diff(y.trn),frc3,col=c("black","red"))
```




```r
frc3ud <- cumsum(c(tail(y.trn,1),frc3))
```

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

```r
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)
```



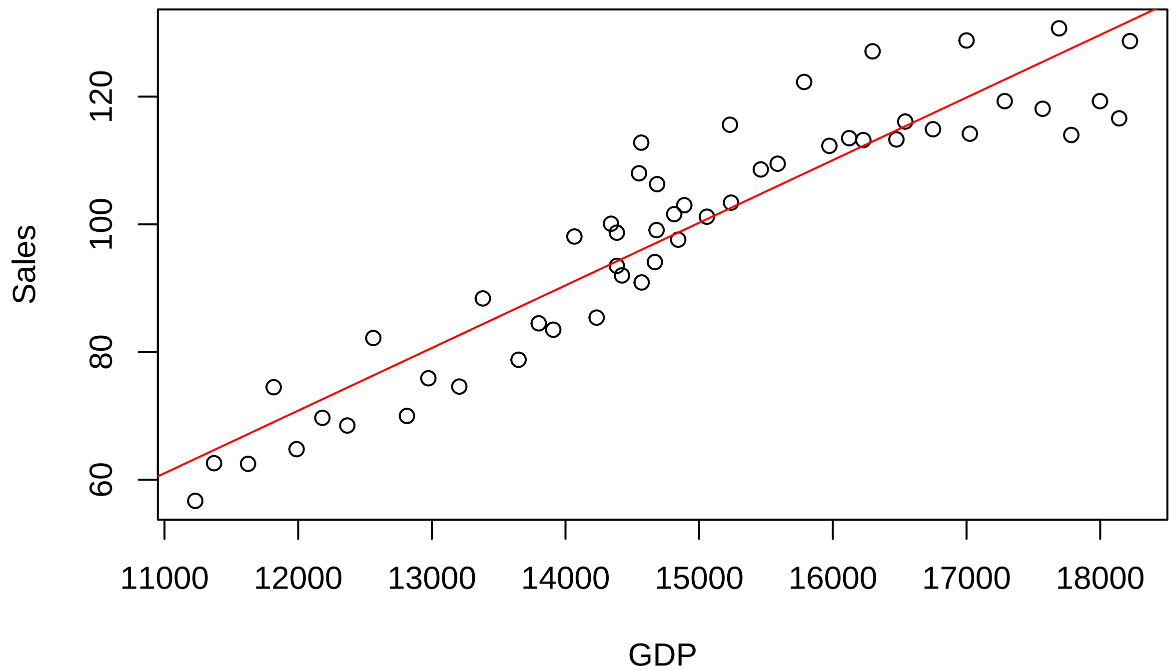

```r
actual <- matrix(rep(y.tst,3),ncol=3)
actual
```

```
##       [,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
```

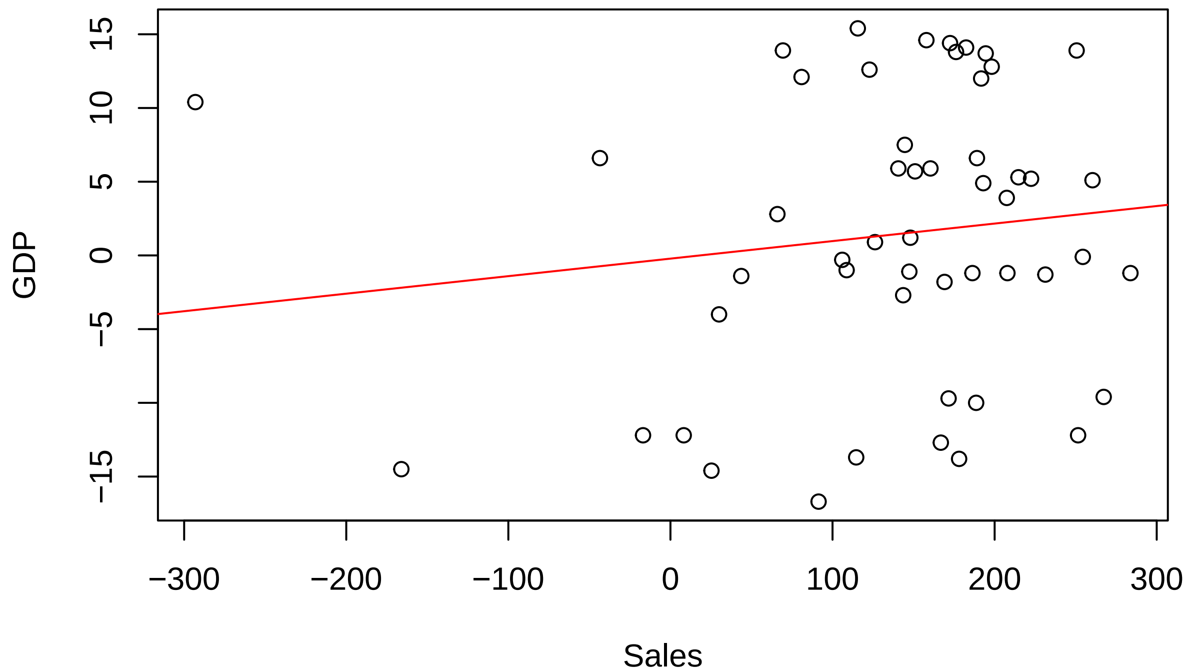
```r
error <- abs(actual - cbind(frc1,frc2,frc3ud))
MAE <- colMeans(error)
MAE
```

```
##     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")
```




```
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
gdp <- c(NA,diff(x[1:(length(x[,2])-8),2]))
# Construct inputs for regression
X4 <- cbind(X3,gdp)
fit6 <- step(lm(y~.,X4[-(1:6),]),trace = 0) # Remove NA
summary(fit6)
```

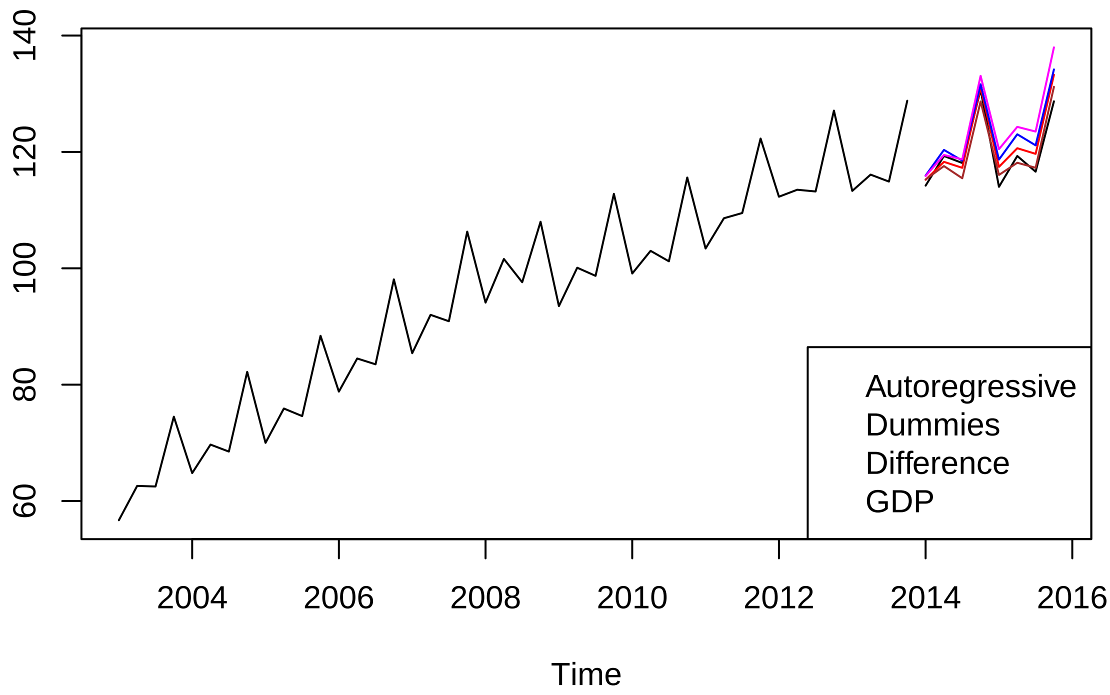
```
##
## Call:
## lm(formula = y ~ lag1 + lag2 + lag3 + lag4 + gdp, data = X4[-(1:6),
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4271 -1.2216  0.5818  1.4958  3.0880
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.532350   0.712472   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){
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]

Xgdp <- tail(gdp,9)

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
```

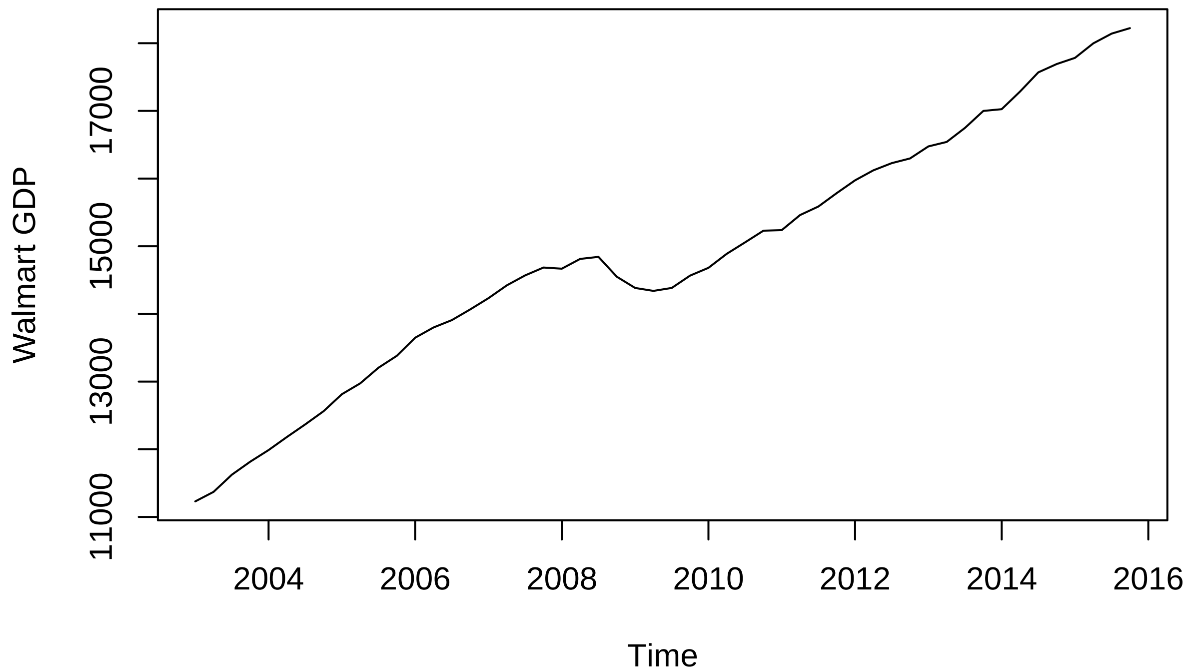
2. Exercises on regression

1. Question

1.1 Find lags

- Plot the data distribution

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

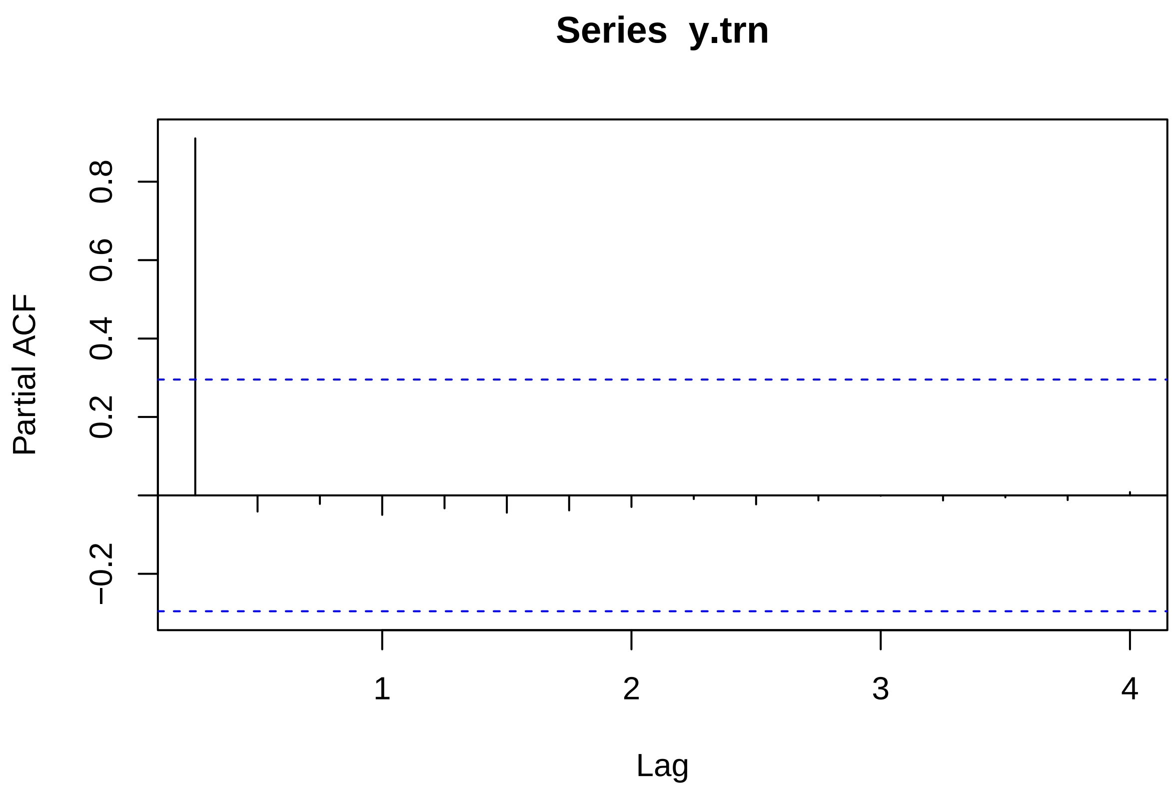


- Train & test split

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

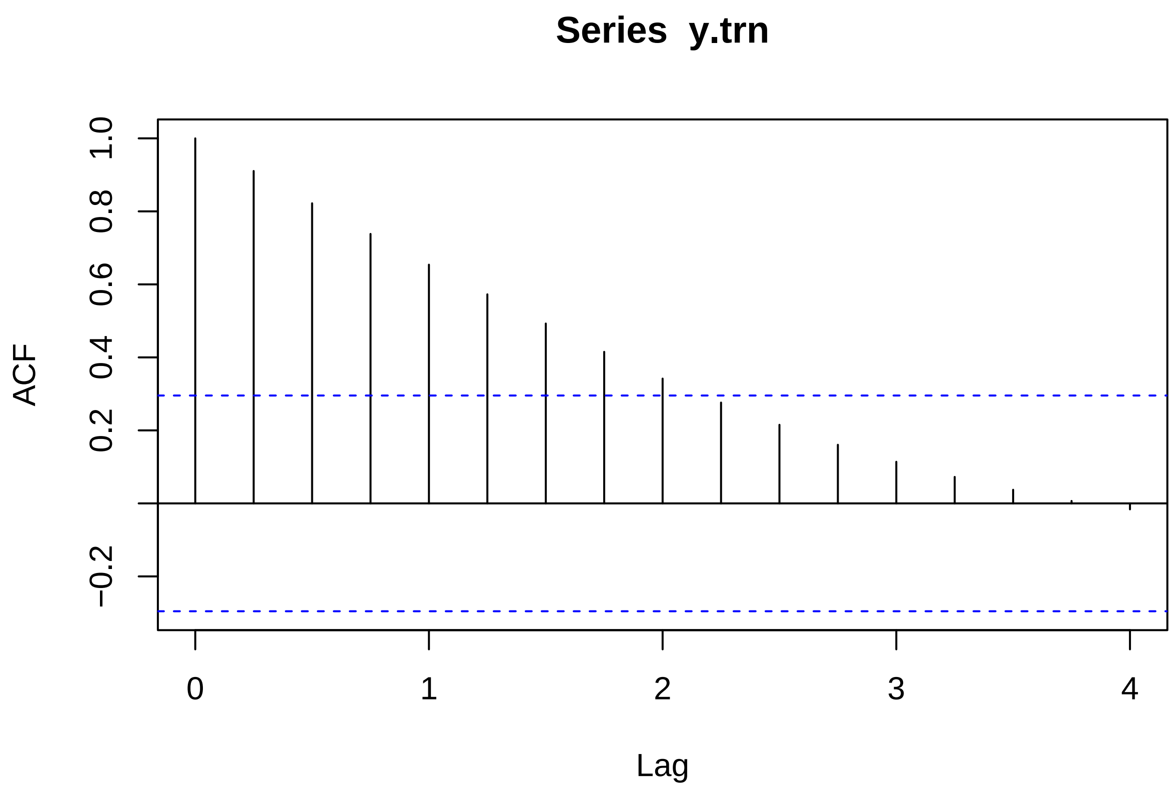
- PACF

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



- ACF

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




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

1.2 Construct lags

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

```r
X <- array(NA,c(n,2))
```

```r
for (i in 1:2){
X[i:n,i] <- y.trn[1:(n-i+1)]
}
# Name the columns
colnames(X) <- c("y",paste0("lag",1))
X[1:10,]
```

```
##             y    lag1
##  [1,] 11230.1      NA
##  [2,] 11370.7 11230.1
##  [3,] 11625.1 11370.7
##  [4,] 11816.8 11625.1
##  [5,] 11988.4 11816.8
```

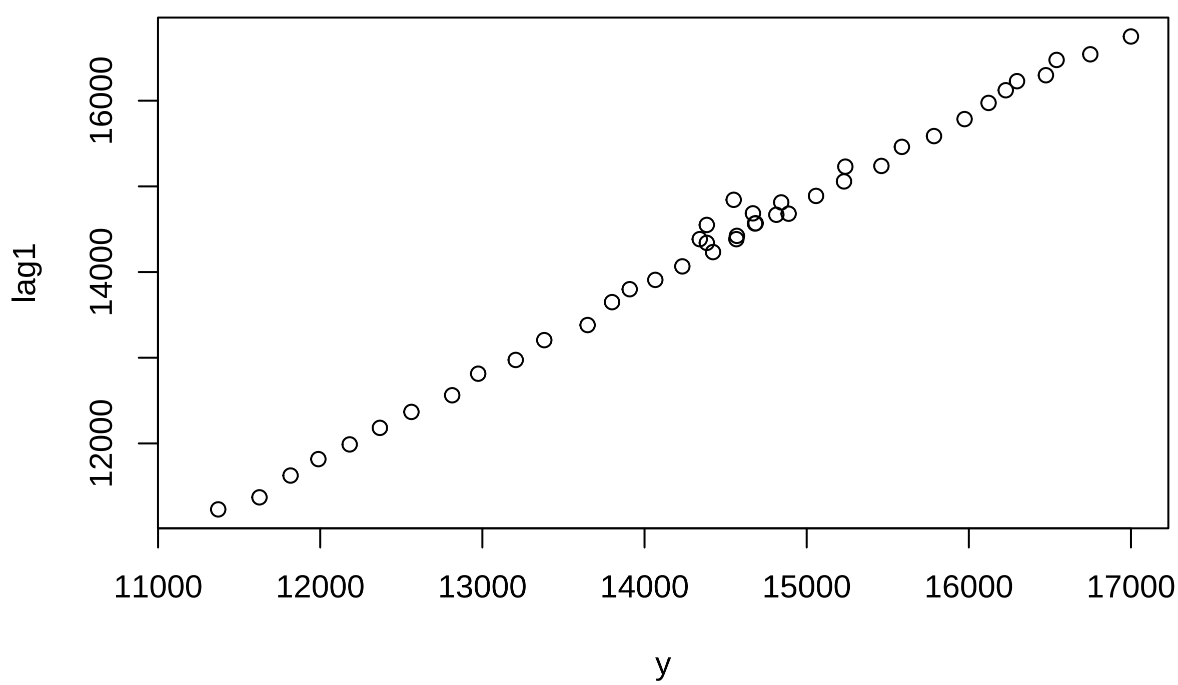
```
##   [6,] 12181.4 11988.4
##   [7,] 12367.7 12181.4
##   [8,] 12562.2 12367.7
##   [9,] 12813.7 12562.2
##  [10,] 12974.1 12813.7
```

```r
X[(n-9):n,]
```

```
##              y    lag1
##   [1,] 15587.1 15460.9
##   [2,] 15785.3 15587.1
##   [3,] 15973.9 15785.3
##   [4,] 16121.9 15973.9
##   [5,] 16227.9 16121.9
##   [6,] 16297.3 16227.9
##   [7,] 16475.4 16297.3
##   [8,] 16541.4 16475.4
##   [9,] 16749.3 16541.4
##  [10,] 16999.9 16749.3
```

- Plot lags correlations

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



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

1.3 Lags - Models

```
##
## Call:
## lm(formula = y ~ ., data = X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -417.85  -23.25   17.57   63.27  157.34
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 370.01411  159.16321   2.325   0.0251 *
## lag1          0.98348    0.01109  88.660   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 107.2 on 41 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9948, Adjusted R-squared:  0.9947
## F-statistic:  7861 on 1 and 41 DF,  p-value: < 2.2e-16
```

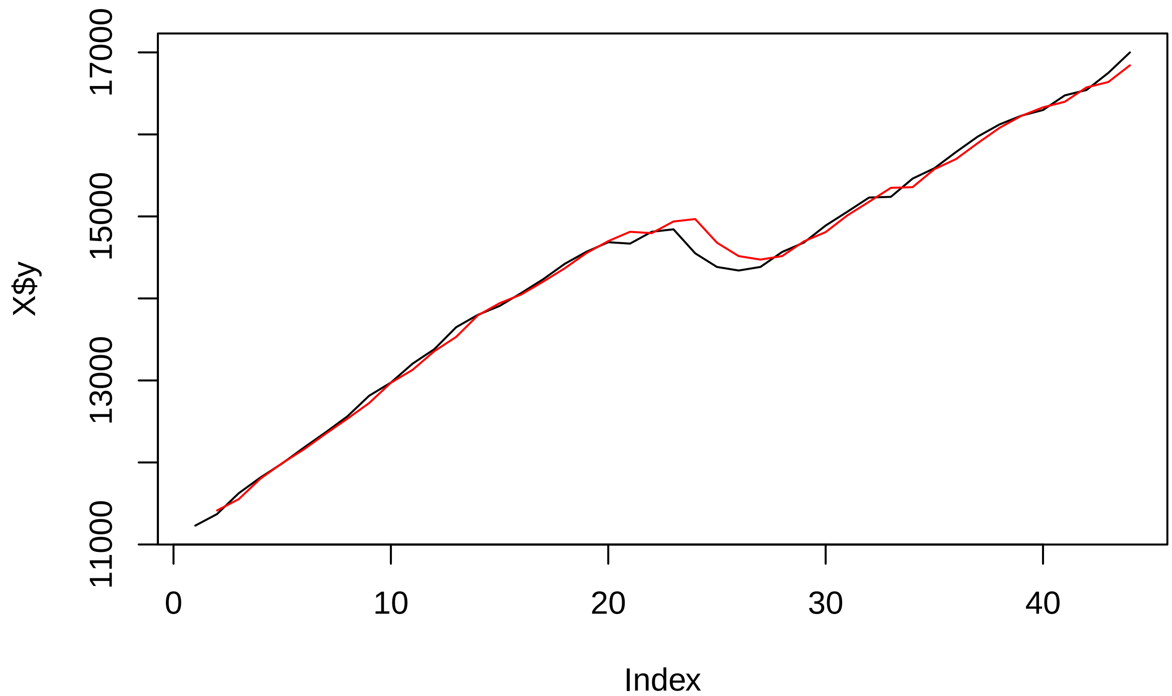
- AIC

```r
AIC(fit1)
```

```
## [1] 528.0349
```

- In-sample vs Fit

```r
# In-sample fit:
plot(X$y,type="l")
frc <- predict(fit1,X)
lines(frc,col="red")
```



- Hold last observation for forecasting

```
Xnew <- array(tail(y.trn,1),c(1,1))
colnames(Xnew) <- paste0("lag",1)
Xnew <- as.data.frame(Xnew)
Xnew
```

```
##      lag1
## 1 16999.9
```

- Predict

```
predict(fit1,Xnew)
```

```
##        1
## 17089.01
```

- Forecast

```
frc1 <- array(NA,c(8,1))
```

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

```
## $lag1
## [1] 16999.9
##
## [[2]]
## [1] NA
##
## [[3]]
## [1] NA
##
## [[4]]
## [1] NA
##
## [[5]]
## [1] NA
##
## [[6]]
## [1] NA
##
## [[7]]
## [1] NA
##
## [[8]]
## [1] NA
##
## [[9]]
## [1] NA
```

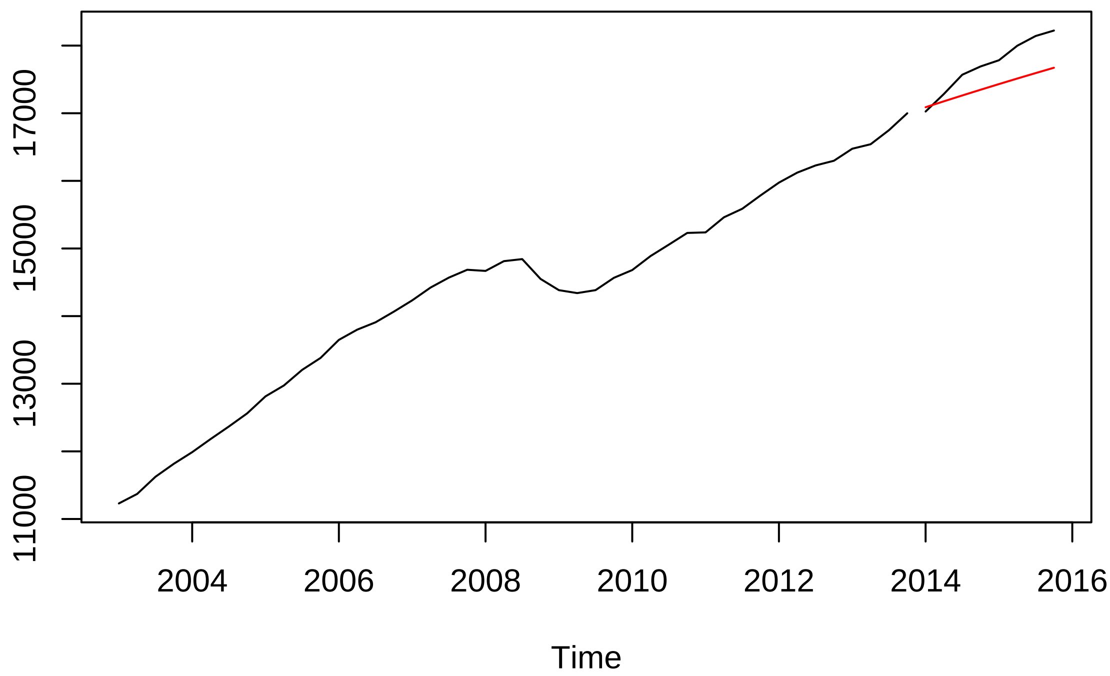
```r
frc1 <- array(NA,c(8,1))
for (i in 1:8){
Xnew <- tail(y.trn,1)
Xnew <- c(Xnew,frc1)
Xnew <- Xnew[i:(0+i)]
Xnew <- array(Xnew, c(1,1))
colnames(Xnew) <- paste0("lag",1)

# Convert to data.frame
Xnew <- as.data.frame(Xnew)
# Forecast
frc1[i] <- predict(fit1,Xnew)
}
frc1
```

```
##          [,1]
## [1,] 17089.01
## [2,] 17176.66
## [3,] 17262.85
## [4,] 17347.62
## [5,] 17430.99
## [6,] 17512.98
## [7,] 17593.62
```

```
## [8,] 17672.92
```

- Plot forecast and test

```
frc1 <- ts(frc1,frequency=frequency(y.tst),start=start(y.tst))
ts.plot(y.trn,y.tst,frc1,col=c("black","black","red"))
```




```
X1 <- X
```

```
for (i in 1:ncol(X1)){
X1[,i] <- c(NA,diff(X1[,i]))
}
print(X1)
```

1.4 Trend - Models

```
##          y   lag1
## 1       NA     NA
## 2    140.6     NA
## 3    254.4  140.6
```

```
## 4   191.7  254.4
## 5   171.6  191.7
## 6   193.0  171.6
## 7   186.3  193.0
## 8   194.5  186.3
## 9   251.5  194.5
## 10  160.4  251.5
## 11  231.3  160.4
## 12  176.2  231.3
## 13  267.3  176.2
## 14  150.9  267.3
## 15  108.7  150.9
## 16  157.9  108.7
## 17  166.8  157.9
## 18  189.1  166.8
## 19  147.4  189.1
## 20  115.6  147.4
## 21  -16.9  115.6
## 22  144.6  -16.9
## 23   30.0  144.6
## 24 -293.1   30.0
## 25 -166.0 -293.1
## 26  -43.5 -166.0
## 27   43.7  -43.5
## 28  182.4   43.7
## 29  114.6  182.4
## 30  207.5  114.6
## 31  169.1  207.5
## 32  172.5  169.1
## 33    8.2  172.5
## 34  222.5    8.2
## 35  126.2  222.5
## 36  198.2  126.2
## 37  188.6  198.2
## 38  148.0  188.6
## 39  106.0  148.0
## 40   69.4  106.0
## 41  178.1   69.4
## 42   66.0  178.1
## 43  207.9   66.0
## 44  250.6  207.9
```

- Build full regression

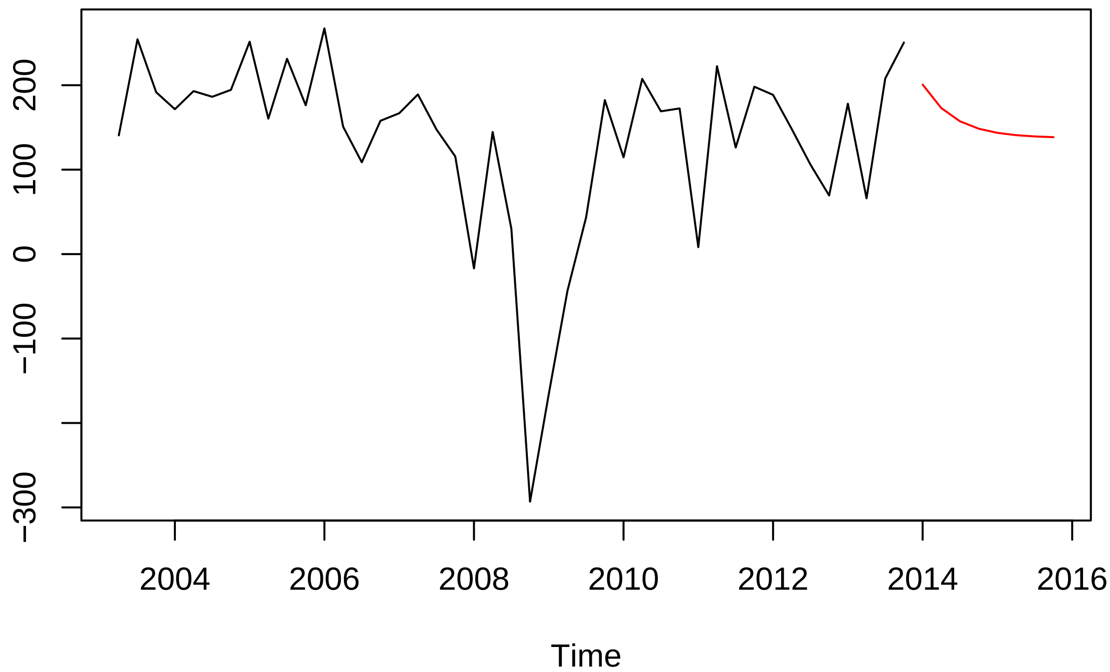
```
fit2 <- lm(y~.,X1)
summary(fit2)
```

```
##
## Call:
## lm(formula = y ~ ., data = X1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -370.34  -39.99    5.71   72.07  157.47
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  60.4395    22.6786   2.665  0.01105 *
## lag1          0.5600     0.1337   4.188  0.00015 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.93 on 40 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.3049, Adjusted R-squared:  0.2875
## F-statistic: 17.54 on 1 and 40 DF,  p-value: 0.0001501
```

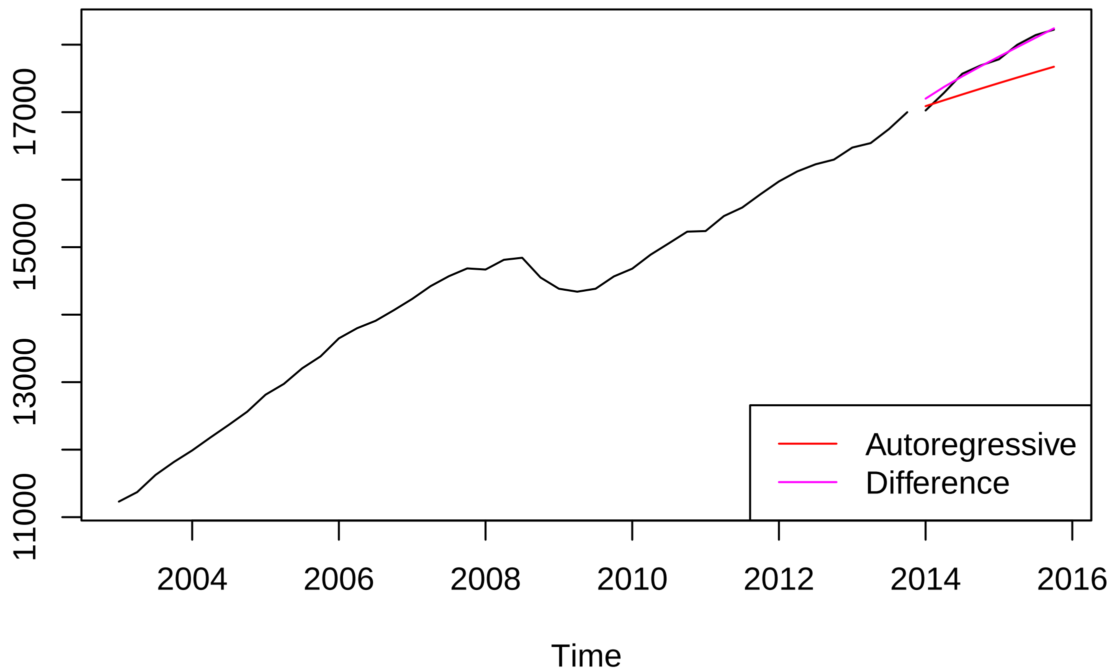
```r
frc2 <- array(NA,c(8,1))
for (i in 1:8){
y.diff <- diff(y.trn)

Xnew <- tail(y.diff,1)
Xnew <- c(Xnew,frc2)
Xnew <- Xnew[i:(0+i)]
Xnew <- Xnew[1]
Xnew <- array(Xnew, c(1,1))
colnames(Xnew) <- paste0("lag",1)
Xnew <- as.data.frame(Xnew)

# Forecast
frc2[i] <- predict(fit2,Xnew)
}
```

```r
# Transform to time series
frc2 <- ts(frc2,frequency=frequency(y.tst),start=start(y.tst))
# Plot
ts.plot(diff(y.trn),frc2,col=c("black","red"))
```




```
frc2ud <- cumsum(c(tail(y.trn,1),frc2))
frc2ud <- frc2ud[-1]
```

```
frc2ud <- ts(frc2ud,frequency=frequency(y.tst),start=start(y.tst))
ts.plot(y.trn,y.tst,frc1,frc2ud,col=c("black","black","red","magenta"))
legend("bottomright",c("Autoregressive","Difference"),col=c("red","magenta"),lty=1)
```



- Compare with the two forecasts

```
actual <- matrix(rep(y.tst,2),ncol=2)
actual
```

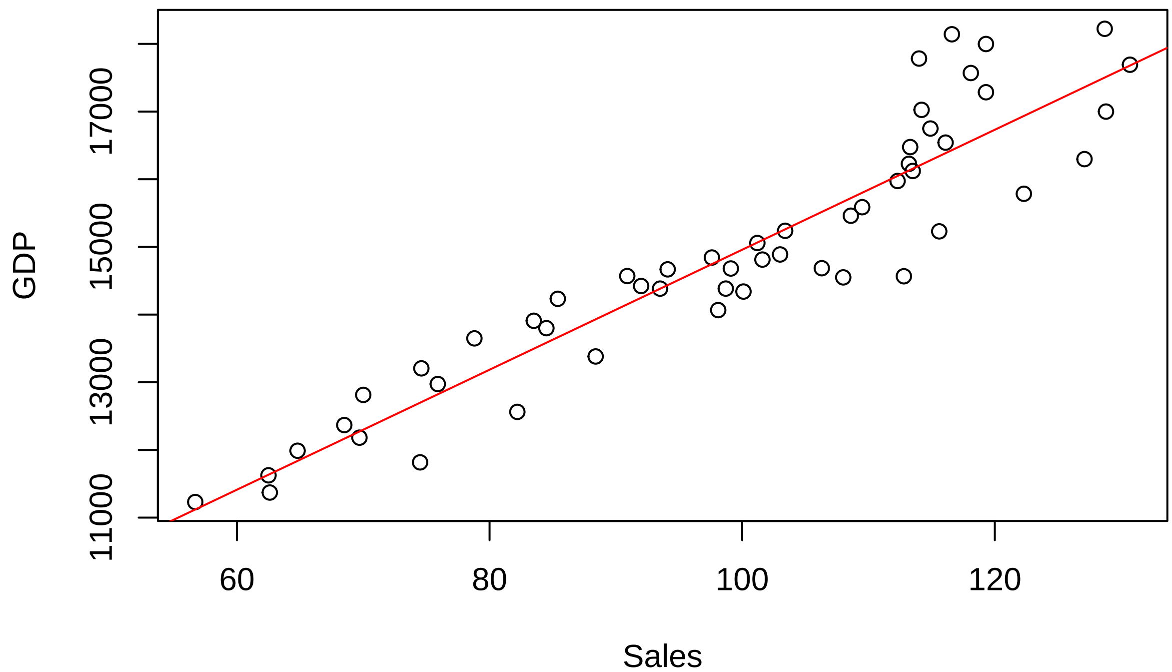
```
##          [,1]    [,2]
## [1,] 17025.2 17025.2
## [2,] 17285.6 17285.6
## [3,] 17569.4 17569.4
## [4,] 17692.2 17692.2
## [5,] 17783.6 17783.6
## [6,] 17998.3 17998.3
## [7,] 18141.9 18141.9
## [8,] 18222.8 18222.8
```

- Calculate MAE

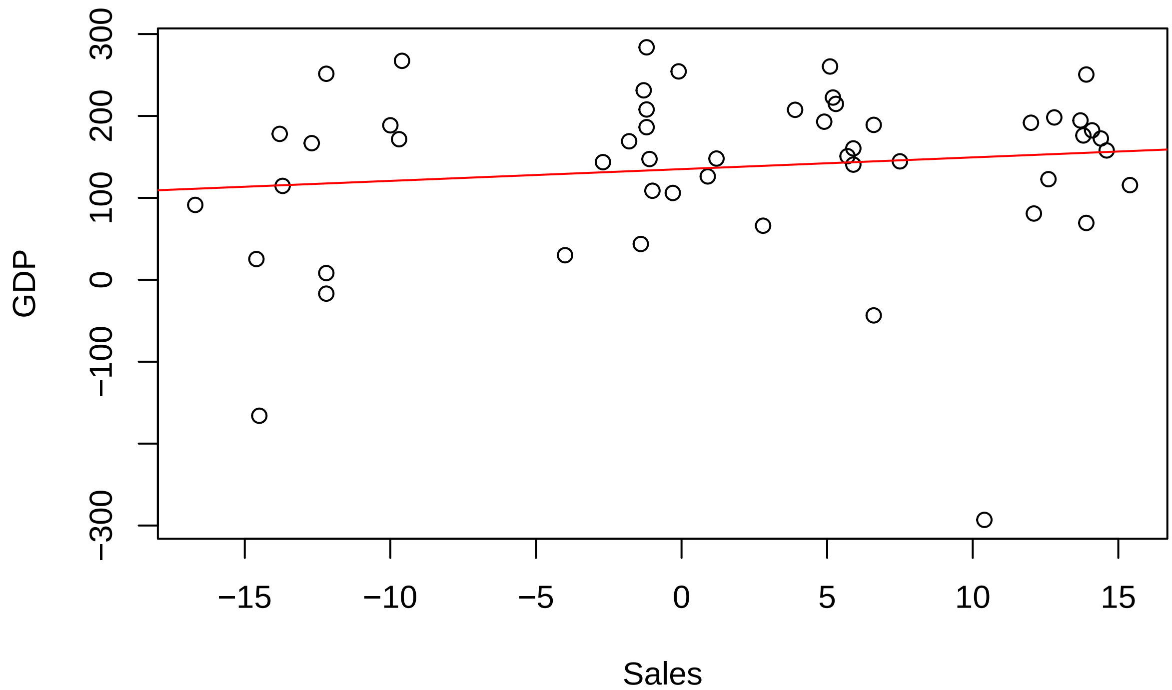
```
error <- abs(actual - cbind(Autoregressive=frc1,Difference=frc2ud))
MAE <- colMeans(error)
MAE
```

```
## Autoregressive     Difference
##      344.99667       55.79193
```

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




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




```r
sales <- c(NA,diff(x[1:(length(x[,1])-8),1]))
# Construct inputs for regression
X2 <- cbind(X1,sales)
fit3 <- lm(y~.,X2[-(2),]) # Remove NA
```

```r
summary(fit3)
```

```
##
## Call:
## lm(formula = y ~ ., data = X2[-(2), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -376.11  -37.98    9.21   65.15  154.70
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  60.1442    22.9363   2.622  0.01240 *
## lag1          0.5551     0.1357   4.089  0.00021 ***
## sales         0.5975     1.5448   0.387  0.70105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93.94 on 39 degrees of freedom
##   (1 observation deleted due to missingness)
```

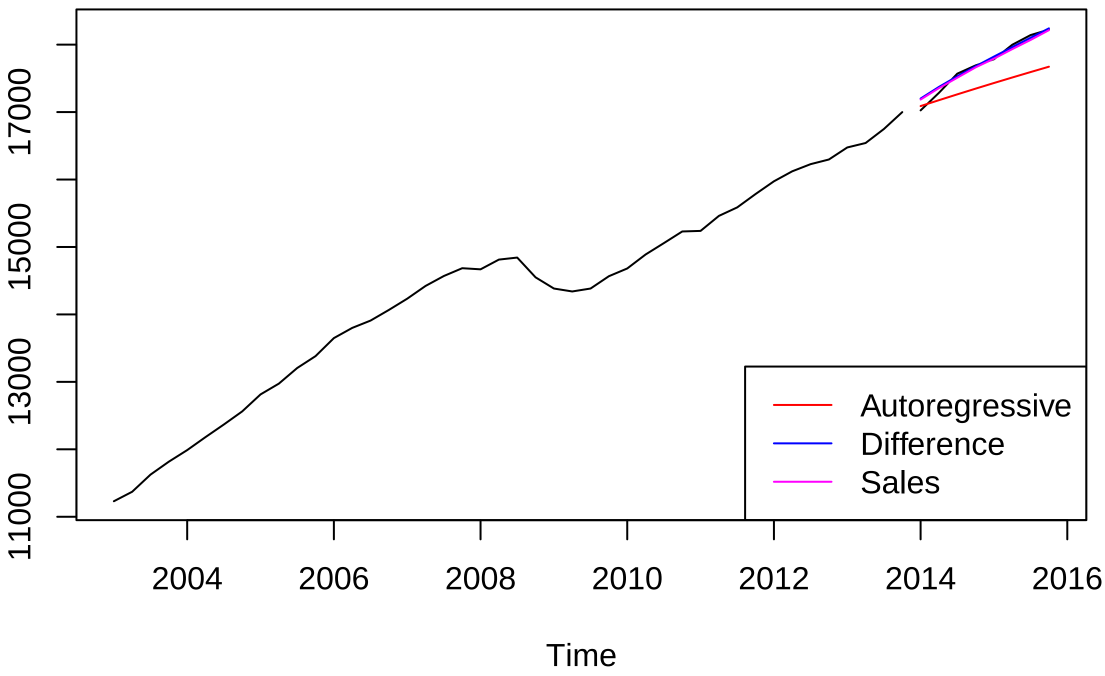
```
## Multiple R-squared:  0.3075, Adjusted R-squared:  0.272
## F-statistic:  8.66 on 2 and 39 DF,  p-value: 0.0007725
```

```
frc3 <- array(NA,c(8,1))
for (i in 1:8){

y.diff <- diff(y.trn)
# Create lags - same as before
Xnew <- tail(y.diff,1)
Xnew <- c(Xnew,frc3)
Xnew <- Xnew[i:(0+i)]


Xsales <- tail(sales,9)
Xsales <- diff(Xsales)
# Use only the i th value
Xsales <- Xsales[i]
# Bind to Xnew
Xnew <- c(Xnew,Xsales)
# Name things
Xnew <- array(Xnew, c(1,2))
colnames(Xnew) <- c(paste0("lag",1),"sales")
Xnew <- as.data.frame(Xnew)
# Forecast
frc3[i] <- predict(fit3,Xnew)
}
print(frc3)
```

```
##          [,1]
## [1,] 185.6248
## [2,] 169.8722
## [3,] 153.5405
## [4,] 153.8553
## [5,] 128.9963
## [6,] 141.6652
## [7,] 136.3897
## [8,] 144.8729
```

```
frc3ud <- cumsum(frc3) + as.vector(tail(y.trn,1))
```

```
frc3ud <- ts(frc3ud,frequency=frequency(y.tst),start=start(y.tst))
ts.plot(y.trn,y.tst,frc1,frc2ud,frc3ud,col=c("black","black","red","blue","magenta"))

legend("bottomright",c("Autoregressive","Difference","Sales"),col=c("red","blue","magenta"),lty=1)
```




```r
c(MAE, Sales=mean(abs(y.tst-frc3ud)))
```

```
## Autoregressive     Difference          Sales
##      344.99667       55.79193       59.14564
```

2. Question 2

```r
library(forecast)
```

2.1 Exponential smoothing

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

```r
class(y.trn)
```

```
## [1] "ts"
```

```
class(y.trn)
```

```
## [1] "ts"
```

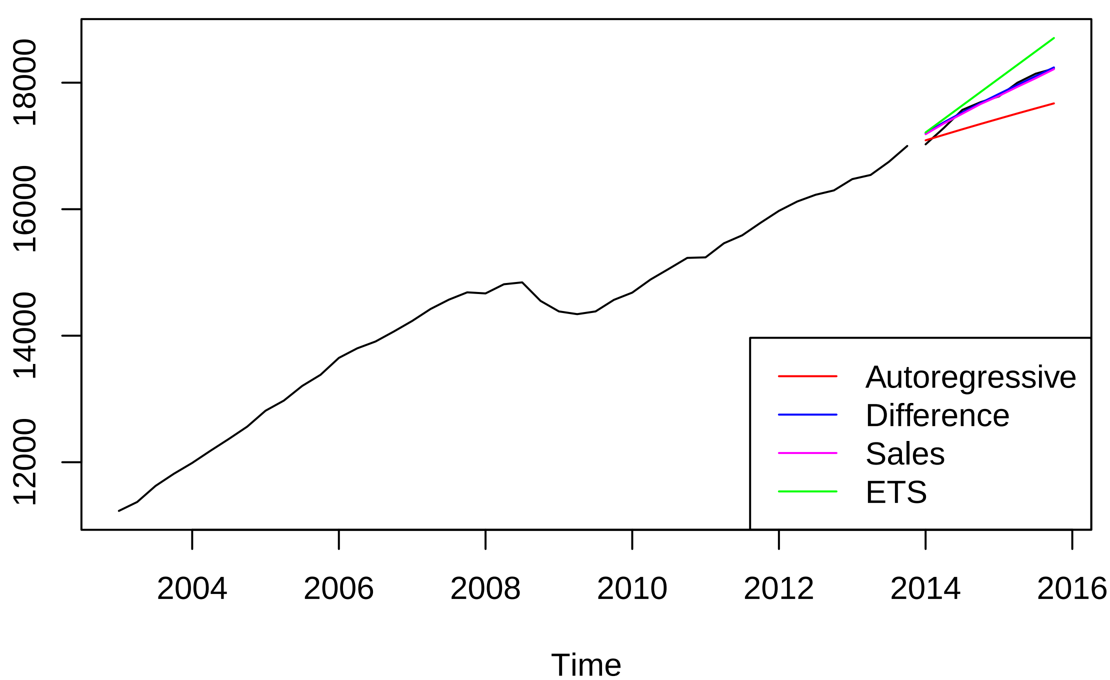
```
fit4 <- ets(y.trn)
class(fit4)
```

```
## [1] "ets"
```

```
frc4 <-forecast(fit4, h=8)
frc4 <- frc4$mean
print(frc4)
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 2014 17213.20 17426.52 17639.83 17853.14
## 2015 18066.45 18279.76 18493.08 18706.39
```

```
ts.plot(y.trn,y.tst,frc1,frc2ud,frc3ud,frc4,col=c("black","black","red","blue","magenta","green"))

legend("bottomright",c("Autoregressive","Difference","Sales","ETS"),col=c("red","blue","magenta","green
```



```r
c(MAE, Sales = mean(abs(y.tst-frc3ud)),ETS =mean(abs(y.tst-frc4)))
```

```
## Autoregressive     Difference          Sales            ETS
##      344.99667       55.79193       59.14564      244.92075
```

Answer

To support the study's conclusions, a comparative analysis was conducted between ETS and OLS regression models to determine the optimal forecasting model. Empirical evidence and evaluation metrics reveal that the **differenced OLS model** utilizing surpasses the ETS benchmark in this scenario, highlighting the practical and analytical advantages associated with this approach in modeling and forecasting observed data.

The study encompassed a comprehensive evaluation of various forecasting methodologies, including autoregressive models applied to lagged GDP estimates, spanning an eight-quarter period. It became evident that these models consistently outperformed Exponential Smoothing, as indicated by the Mean Absolute Error (MAE) statistic and visual inspections employed as evaluation criteria.