Adidas_Sales_Forecast.R

DELL

Fri Dec 07 14:06:29 2018

```
#Loading the dataset
library(ggplot2)
library(readr)
library(fpp)
```

```
## Loading required package: forecast
```

```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: tseries
```

```
library(fpp2)
```

```
##
## Attaching package: 'fpp2'
```

```
## The following objects are masked from 'package:fpp':
##
##     ausair, ausbeer, austa, austourists, debitcards, departures,
##     elecequip, euretail, guinearice, oil, sunspotarea, usmelec
```

```
#reading data from csv file
data = read.csv("C:/Users/DELL/Downloads/adidas_revenue1.csv")
data
```

```
##      date Revenue   US.GDP Europe.GDP CHN.GDP.US Price.Index   NIKE
## 1  2000Q1    1517 12359.09   1728.244   257.6491       246.5 2161.6
## 2  2000Q2    1248 12592.53   1749.600   290.4375       326.4 2272.7
## 3  2000Q3    1677 12607.68   1769.259   310.5510       322.5 2636.7
## 4  2000Q4    1393 12679.34   1789.253   352.6953       321.0 2198.7
## 5  2001Q1    1558 12643.28   1819.140   290.9745       329.4 2170.1
## 6  2001Q2    1368 12710.30   1833.713   322.9186       329.8 2483.3
## 7  2001Q3    1790 12670.11   1845.879   342.3183       313.3 2613.7
## 8  2001Q4    1396 12705.27   1861.936   383.2037       302.1 2336.8
## 9  2002Q1    1638 12822.26   1878.783   317.6722       296.5 2260.3
## 10 2002Q2    1507 12893.00   1893.801   352.7202       301.0 2682.2
## 11 2002Q3    1868 12955.77   1914.837   377.6277       298.9 2796.3
## 12 2002Q4    1510 12964.02   1927.341   422.4808       292.2 2514.7
## 13 2003Q1    1669 13031.17   1933.872   360.3185       294.0 2400.9
## 14 2003Q2    1392 13152.09   1943.995   393.0824       296.3 2985.1
## 15 2003Q3    1853 13372.36   1969.849   426.3864       300.7 3024.9
## 16 2003Q4    1353 13528.71   1988.492   480.4693       309.2 2837.1
## 17 2004Q1    1623 13606.51   2010.120   417.3614       327.1 2904.0
## 18 2004Q2    1401 13706.25   2031.765   467.5954       333.0 3487.1
## 19 2004Q3    1758 13830.83   2046.182   505.7028       332.7 3561.8
## 20 2004Q4    1078 13950.38   2065.015   564.7315       321.3 3148.3
## 21 2005Q1    1674 14099.08   2076.923   488.7694       310.3 3308.2
## 22 2005Q2    1516 14172.69   2101.559   541.2070       311.3 3721.4
## 23 2005Q3    1924 14291.76   2124.553   593.7716       304.9 3862.0
## 24 2005Q4    1522 14373.44   2153.840   669.4382       297.6 3474.7
## 25 2006Q1    2459 14546.12   2179.109   587.2396       292.9 3612.8
## 26 2006Q2    2428 14589.58   2214.849   658.7748       295.3 4005.4
## 27 2006Q3    2949 14602.63   2240.239   708.9029       295.9 4194.1
## 28 2006Q4    2248 14716.93   2275.903   814.7579       297.4 3821.7
## 29 2007Q1    2538 14726.02   2314.019   739.2749       299.8 3926.9
## 30 2007Q2    2400 14838.66   2338.311   851.0275       301.4 4383.2
## 31 2007Q3    2941 14938.47   2362.323   925.6537       305.6 4655.1
## 32 2007Q4    2420 14991.78   2392.426  1077.6908       310.7 4339.5
## 33 2008Q1    2621 14889.45   2416.113   988.8873       322.3 4544.4
## 34 2008Q2    2521 14963.36   2419.735  1148.3868       334.9 5088.0
## 35 2008Q3    3083 14891.64   2411.679  1210.5950       335.8 5432.2
## 36 2008Q4    2574 14576.99   2382.319  1299.1836       332.0 4590.1
## 37 2009Q1    2577 14375.02   2317.899  1083.2955       323.0 4440.8
## 38 2009Q2    2457 14355.56   2310.312  1229.2481       320.8 4713.0
## 39 2009Q3    2888 14402.48   2320.823  1318.1139       326.8 4798.5
## 40 2009Q4    2458 14541.90   2339.329  1479.6432       332.5 4405.6
## 41 2010Q1    2674 14604.84   2349.078  1283.5211       338.0 4733.0
## 42 2010Q2    2917 14745.93   2377.207  1465.6673       347.7 5076.9
## 43 2010Q3    3468 14845.46   2396.736  1585.3964       354.0 5175.0
## 44 2010Q4    2931 14939.00   2415.849  1806.4896       431.9 4842.0
## 45 2011Q1    3273 14881.30   2439.892  1596.0131       567.3 5079.0
## 46 2011Q2    3064 14989.56   2445.959  1841.4921       655.8 5766.0
## 47 2011Q3    3744 15021.15   2454.501  1998.1746       576.6 6081.0
## 48 2011Q4    3241 15190.25   2455.081  2198.1463       505.5 5546.0
## 49 2012Q1    3824 15291.03   2459.268  1868.2618       431.7 5656.0
## 50 2012Q2    3517 15362.42   2457.575  2081.9618       410.7 6236.0
## 51 2012Q3    4173 15380.80   2461.767  2186.1221       379.0 6474.0
## 52 2012Q4    3369 15384.25   2459.338  2424.1446       369.2 5955.0
```

```
## 53 2013Q1    3751 15491.88   2462.191  2067.9449       370.9 6187.0
## 54 2013Q2    3383 15521.56   2480.438  2330.3550       389.3 6697.0
## 55 2013Q3    3879 15641.34   2492.943  2487.0690       387.0 6971.0
## 56 2013Q4    3391 15793.93   2503.542  2763.2571       378.1 6431.0
## 57 2014Q1    3480 15757.57   2523.399  2285.6911       382.6 6972.0
## 58 2014Q2    3400 15935.83   2528.024  2542.9235       390.0 7425.0
## 59 2014Q3    4044 16139.51   2545.888  2693.4092       380.3 7982.0
## 60 2014Q4    3610 16220.22   2567.073  2960.9904       360.6 7380.0
## 61 2015Q1    4083 16349.97   2600.116  2458.1909       336.7 7460.0
## 62 2015Q2    3907 16460.89   2616.887  2756.1993       333.2 7779.0
## 63 2015Q3    4758 16527.59   2637.050  2777.8913       333.5 8414.0
## 64 2015Q4    4167 16547.62   2656.240  2970.8388       331.3 7686.0
## 65 2016Q1    4769 16571.57   2672.910  2498.3069       329.4 8032.0
## 66 2016Q2    4199 16663.52   2683.968  2720.2047       324.1 8244.0
## 67 2016Q3    5413 16778.15   2699.526  2856.3504       322.4 9061.0
## 68 2016Q4    4687 16851.42   2725.422  3040.1019       325.4 8180.0
## 69 2017Q1    5671 16903.24   2745.103  2622.0142       344.6 8432.0
```

```
#converting into time series data
data<-ts(data, start=c(2000,1), end=c(2017,1), frequency = 4)
data
```

```
##         date Revenue   US.GDP Europe.GDP CHN.GDP.US Price.Index   NIKE
## 2000 Q1    1    1517 12359.09   1728.244   257.6491       246.5 2161.6
## 2000 Q2    2    1248 12592.53   1749.600   290.4375       326.4 2272.7
## 2000 Q3    3    1677 12607.68   1769.259   310.5510       322.5 2636.7
## 2000 Q4    4    1393 12679.34   1789.253   352.6953       321.0 2198.7
## 2001 Q1    5    1558 12643.28   1819.140   290.9745       329.4 2170.1
## 2001 Q2    6    1368 12710.30   1833.713   322.9186       329.8 2483.3
## 2001 Q3    7    1790 12670.11   1845.879   342.3183       313.3 2613.7
## 2001 Q4    8    1396 12705.27   1861.936   383.2037       302.1 2336.8
## 2002 Q1    9    1638 12822.26   1878.783   317.6722       296.5 2260.3
## 2002 Q2   10    1507 12893.00   1893.801   352.7202       301.0 2682.2
## 2002 Q3   11    1868 12955.77   1914.837   377.6277       298.9 2796.3
## 2002 Q4   12    1510 12964.02   1927.341   422.4808       292.2 2514.7
## 2003 Q1   13    1669 13031.17   1933.872   360.3185       294.0 2400.9
## 2003 Q2   14    1392 13152.09   1943.995   393.0824       296.3 2985.1
## 2003 Q3   15    1853 13372.36   1969.849   426.3864       300.7 3024.9
## 2003 Q4   16    1353 13528.71   1988.492   480.4693       309.2 2837.1
## 2004 Q1   17    1623 13606.51   2010.120   417.3614       327.1 2904.0
## 2004 Q2   18    1401 13706.25   2031.765   467.5954       333.0 3487.1
## 2004 Q3   19    1758 13830.83   2046.182   505.7028       332.7 3561.8
## 2004 Q4   20    1078 13950.38   2065.015   564.7315       321.3 3148.3
## 2005 Q1   21    1674 14099.08   2076.923   488.7694       310.3 3308.2
## 2005 Q2   22    1516 14172.69   2101.559   541.2070       311.3 3721.4
## 2005 Q3   23    1924 14291.76   2124.553   593.7716       304.9 3862.0
## 2005 Q4   24    1522 14373.44   2153.840   669.4382       297.6 3474.7
## 2006 Q1   25    2459 14546.12   2179.109   587.2396       292.9 3612.8
## 2006 Q2   26    2428 14589.58   2214.849   658.7748       295.3 4005.4
## 2006 Q3   27    2949 14602.63   2240.239   708.9029       295.9 4194.1
## 2006 Q4   28    2248 14716.93   2275.903   814.7579       297.4 3821.7
## 2007 Q1   29    2538 14726.02   2314.019   739.2749       299.8 3926.9
## 2007 Q2   30    2400 14838.66   2338.311   851.0275       301.4 4383.2
## 2007 Q3   31    2941 14938.47   2362.323   925.6537       305.6 4655.1
## 2007 Q4   32    2420 14991.78   2392.426  1077.6908       310.7 4339.5
## 2008 Q1   33    2621 14889.45   2416.113   988.8873       322.3 4544.4
## 2008 Q2   34    2521 14963.36   2419.735  1148.3868       334.9 5088.0
## 2008 Q3   35    3083 14891.64   2411.679  1210.5950       335.8 5432.2
## 2008 Q4   36    2574 14576.99   2382.319  1299.1836       332.0 4590.1
## 2009 Q1   37    2577 14375.02   2317.899  1083.2955       323.0 4440.8
## 2009 Q2   38    2457 14355.56   2310.312  1229.2481       320.8 4713.0
## 2009 Q3   39    2888 14402.48   2320.823  1318.1139       326.8 4798.5
## 2009 Q4   40    2458 14541.90   2339.329  1479.6432       332.5 4405.6
## 2010 Q1   41    2674 14604.84   2349.078  1283.5211       338.0 4733.0
## 2010 Q2   42    2917 14745.93   2377.207  1465.6673       347.7 5076.9
## 2010 Q3   43    3468 14845.46   2396.736  1585.3964       354.0 5175.0
## 2010 Q4   44    2931 14939.00   2415.849  1806.4896       431.9 4842.0
## 2011 Q1   45    3273 14881.30   2439.892  1596.0131       567.3 5079.0
## 2011 Q2   46    3064 14989.56   2445.959  1841.4921       655.8 5766.0
## 2011 Q3   47    3744 15021.15   2454.501  1998.1746       576.6 6081.0
## 2011 Q4   48    3241 15190.25   2455.081  2198.1463       505.5 5546.0
## 2012 Q1   49    3824 15291.03   2459.268  1868.2618       431.7 5656.0
## 2012 Q2   50    3517 15362.42   2457.575  2081.9618       410.7 6236.0
## 2012 Q3   51    4173 15380.80   2461.767  2186.1221       379.0 6474.0
## 2012 Q4   52    3369 15384.25   2459.338  2424.1446       369.2 5955.0
```

```
## 2013 Q1   53    3751 15491.88   2462.191  2067.9449       370.9 6187.0
## 2013 Q2   54    3383 15521.56   2480.438  2330.3550       389.3 6697.0
## 2013 Q3   55    3879 15641.34   2492.943  2487.0690       387.0 6971.0
## 2013 Q4   56    3391 15793.93   2503.542  2763.2571       378.1 6431.0
## 2014 Q1   57    3480 15757.57   2523.399  2285.6911       382.6 6972.0
## 2014 Q2   58    3400 15935.83   2528.024  2542.9235       390.0 7425.0
## 2014 Q3   59    4044 16139.51   2545.888  2693.4092       380.3 7982.0
## 2014 Q4   60    3610 16220.22   2567.073  2960.9904       360.6 7380.0
## 2015 Q1   61    4083 16349.97   2600.116  2458.1909       336.7 7460.0
## 2015 Q2   62    3907 16460.89   2616.887  2756.1993       333.2 7779.0
## 2015 Q3   63    4758 16527.59   2637.050  2777.8913       333.5 8414.0
## 2015 Q4   64    4167 16547.62   2656.240  2970.8388       331.3 7686.0
## 2016 Q1   65    4769 16571.57   2672.910  2498.3069       329.4 8032.0
## 2016 Q2   66    4199 16663.52   2683.968  2720.2047       324.1 8244.0
## 2016 Q3   67    5413 16778.15   2699.526  2856.3504       322.4 9061.0
## 2016 Q4   68    4687 16851.42   2725.422  3040.1019       325.4 8180.0
## 2017 Q1   69    5671 16903.24   2745.103  2622.0142       344.6 8432.0
```

```
#extracting sales data
y=data[,2]
y
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2000 1517 1248 1677 1393
## 2001 1558 1368 1790 1396
## 2002 1638 1507 1868 1510
## 2003 1669 1392 1853 1353
## 2004 1623 1401 1758 1078
## 2005 1674 1516 1924 1522
## 2006 2459 2428 2949 2248
## 2007 2538 2400 2941 2420
## 2008 2621 2521 3083 2574
## 2009 2577 2457 2888 2458
## 2010 2674 2917 3468 2931
## 2011 3273 3064 3744 3241
## 2012 3824 3517 4173 3369
## 2013 3751 3383 3879 3391
## 2014 3480 3400 4044 3610
## 2015 4083 3907 4758 4167
## 2016 4769 4199 5413 4687
## 2017 5671
```

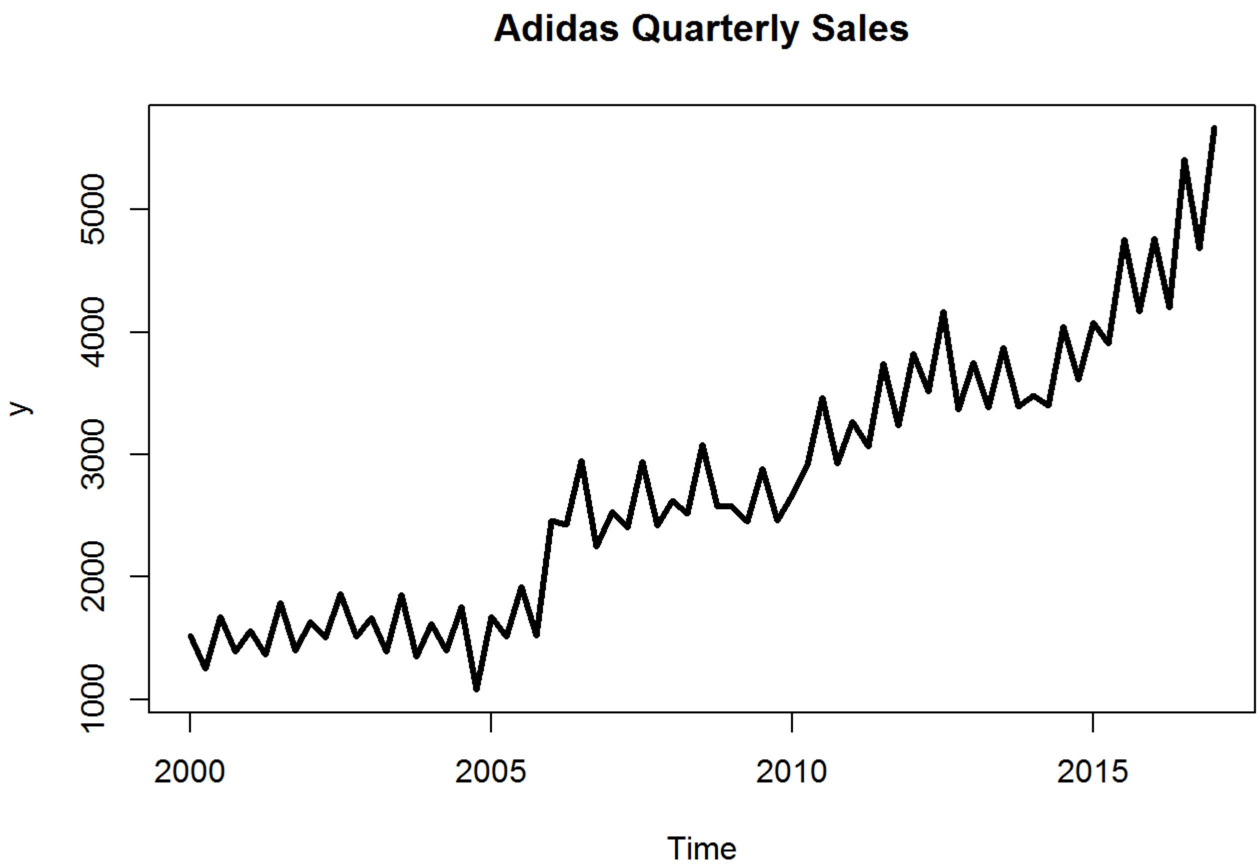
```
#Set training dataset and testing dataset
train_data = window(y,start=c(2000,1), end=c(2013,4))
test_data = window(y,start=2014)
train_data
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2000 1517 1248 1677 1393
## 2001 1558 1368 1790 1396
## 2002 1638 1507 1868 1510
## 2003 1669 1392 1853 1353
## 2004 1623 1401 1758 1078
## 2005 1674 1516 1924 1522
## 2006 2459 2428 2949 2248
## 2007 2538 2400 2941 2420
## 2008 2621 2521 3083 2574
## 2009 2577 2457 2888 2458
## 2010 2674 2917 3468 2931
## 2011 3273 3064 3744 3241
## 2012 3824 3517 4173 3369
## 2013 3751 3383 3879 3391
```

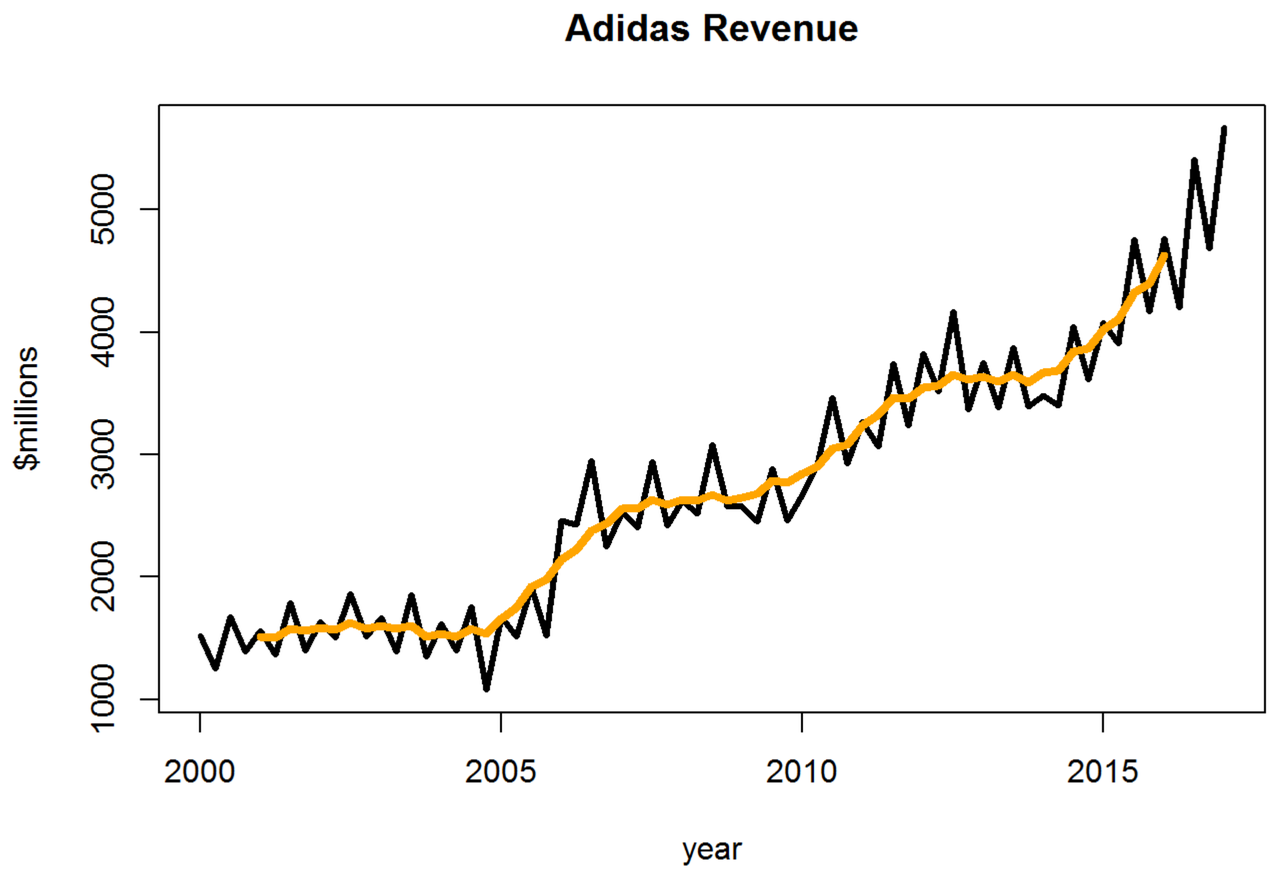
```
test_data
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 2014 3480 3400 4044 3610
## 2015 4083 3907 4758 4167
## 2016 4769 4199 5413 4687
## 2017 5671
```

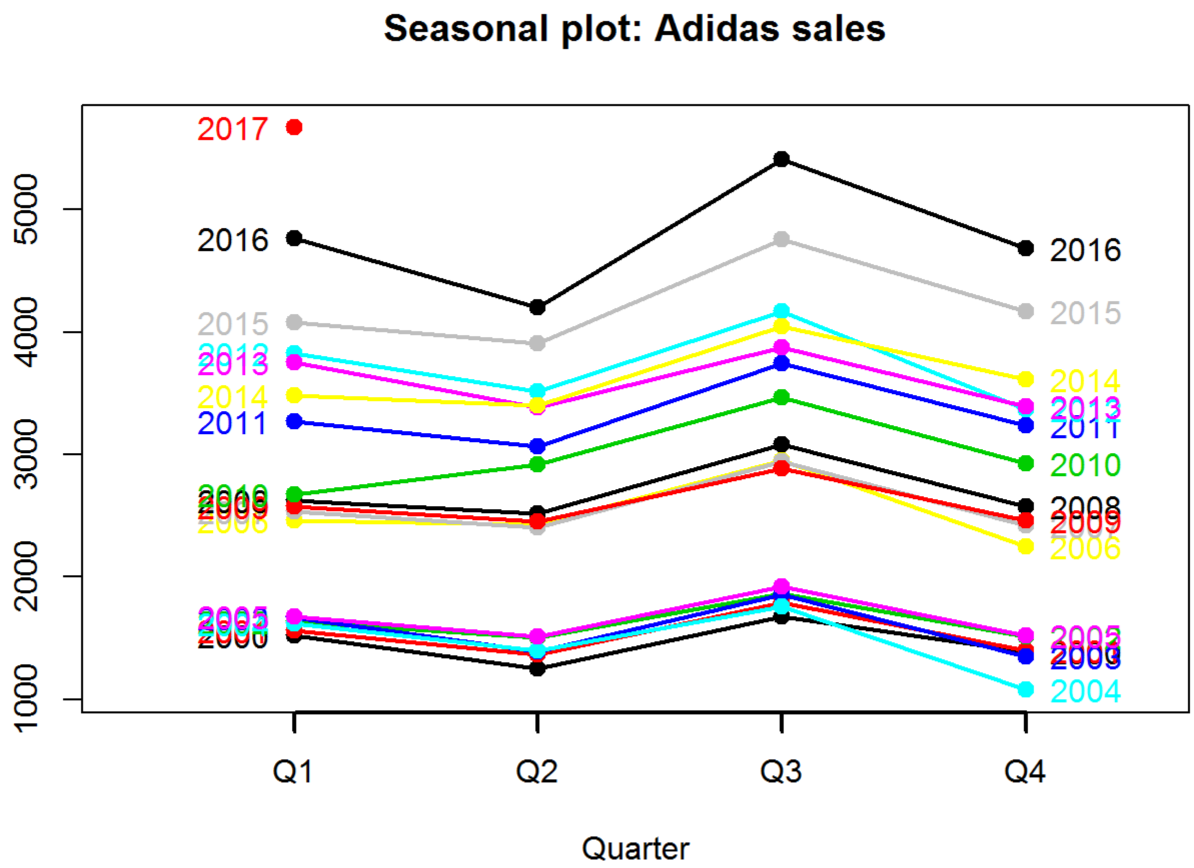
```
#plot raw dataset
plot(y,main="Adidas Quarterly Sales", lwd=3)
```



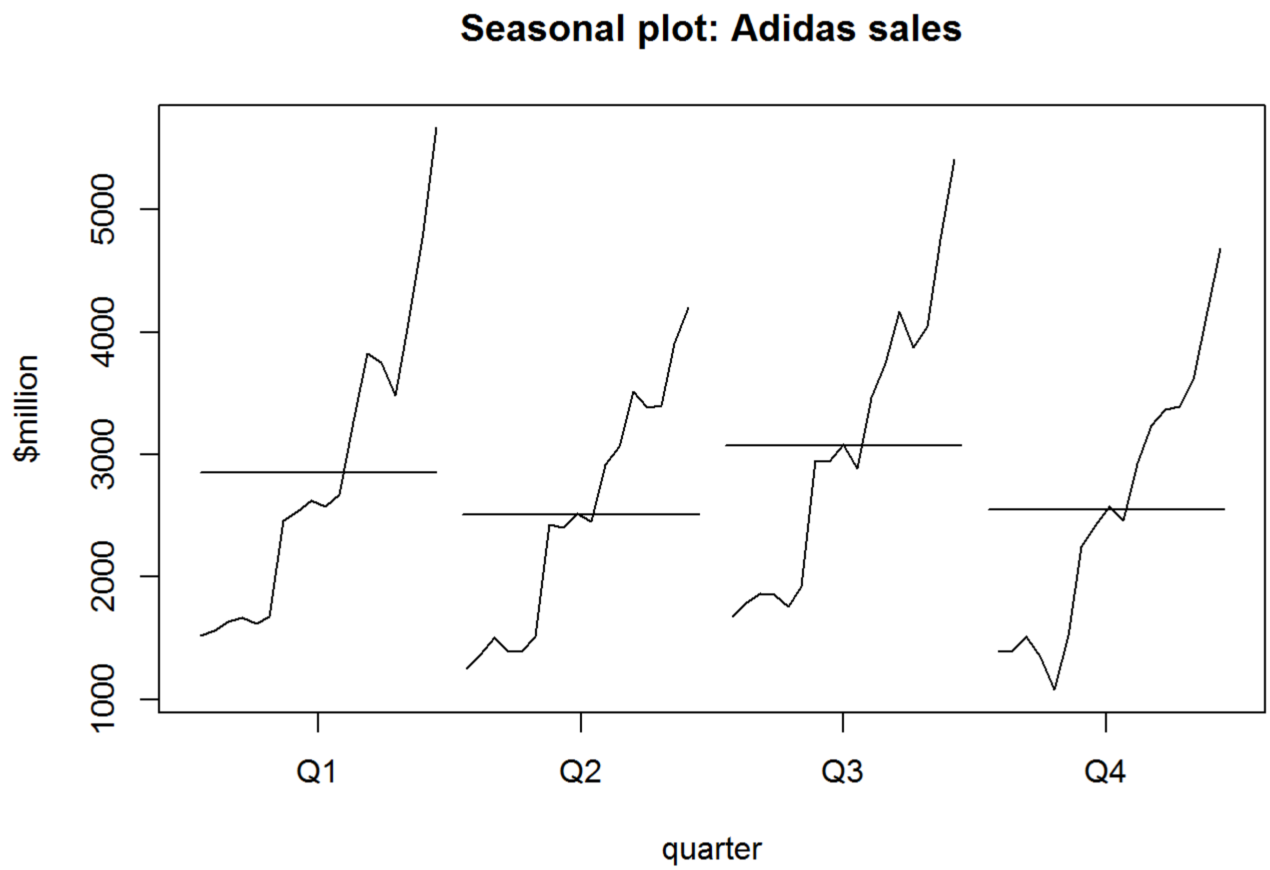

```
plot(y,main="Adidas Revenue", xlab="year", ylab="$millions", lwd=3)
lines(ma(y,9),col="orange",lwd=4)
```



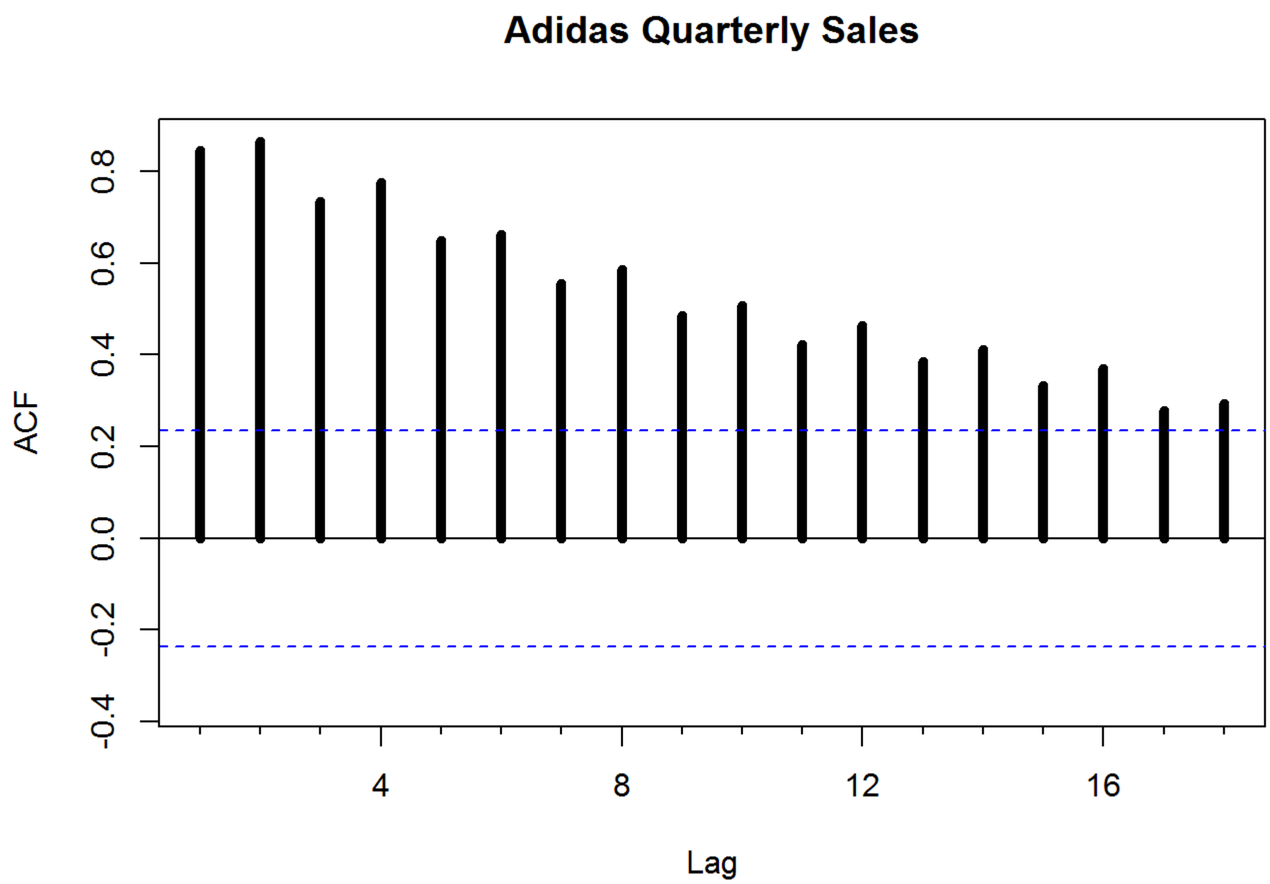

```
#seasonlity plot
seasonplot(y,main="Seasonal plot: Adidas sales",
          year.labels = TRUE, year.labels.left = TRUE,
          col=1:20, pch=19,lwd=2)
```



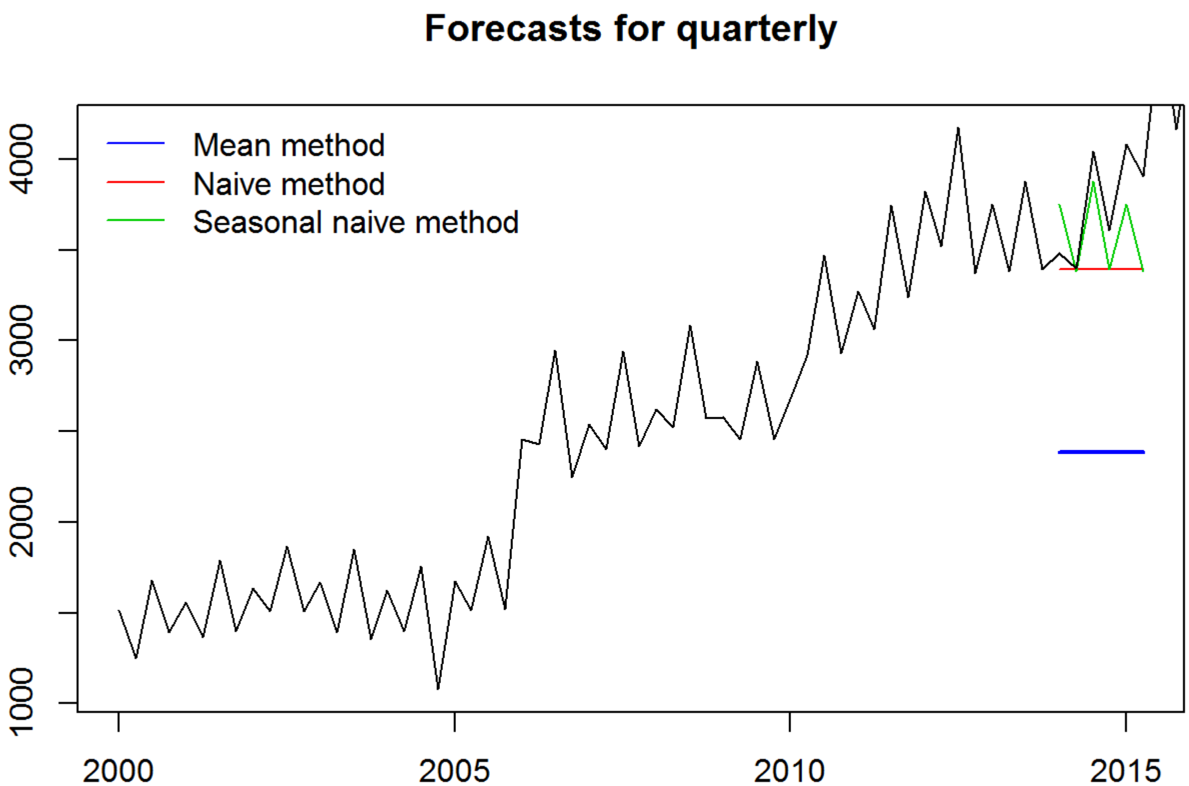

```
monthplot(y, main="Seasonal plot: Adidas sales",
          xlab="quarter", ylab = "$million")
```




```
#Autocorrelation
Acf(y, lwd=5, main="Adidas Quarterly Sales")
```



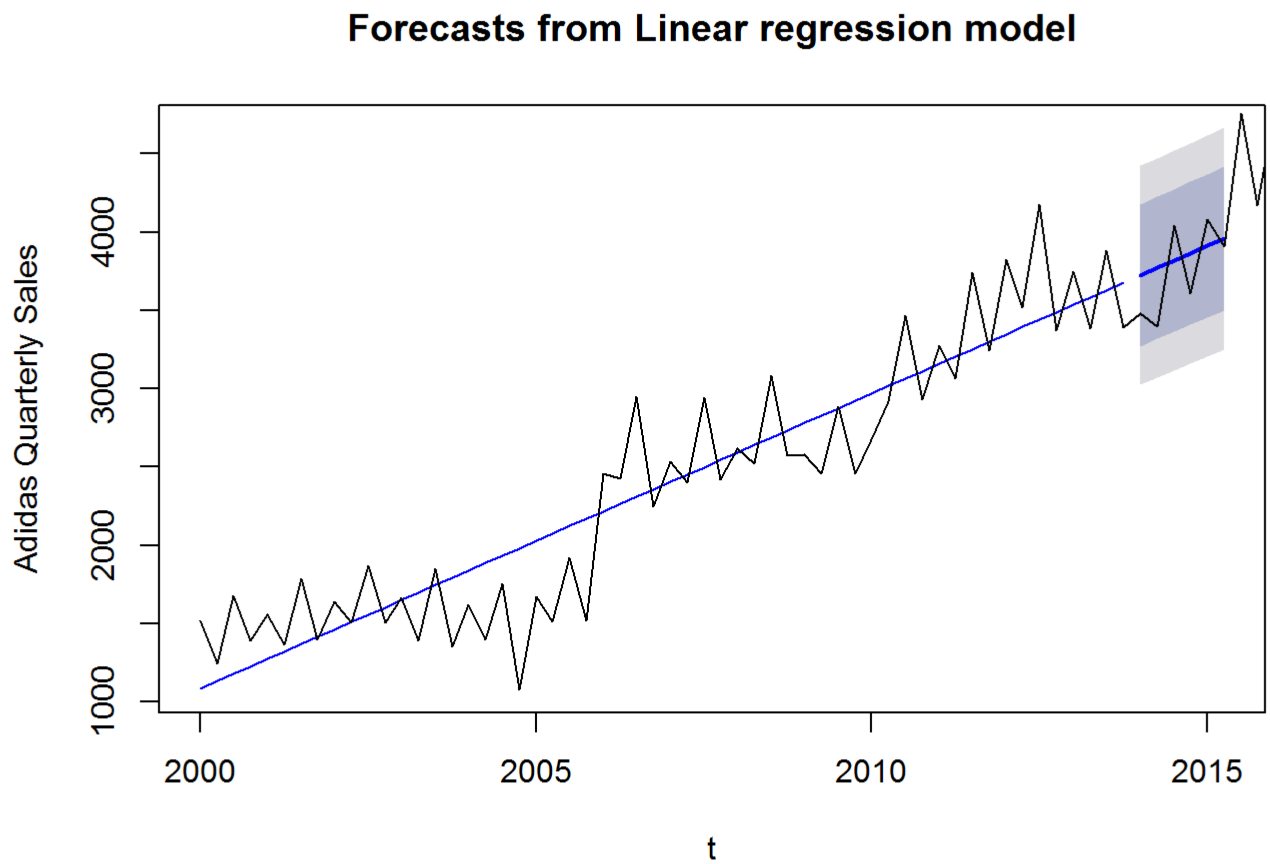

```
#Looking at the raw dataset, ADIDAS sales have a strong seasonal and increasing trend
#pattern. Out of four quarters, the third quarter generally has better performance.
#It also has a strong correlation with its lagged data.


#Simple forecasting methods:Mean,Naive, Seasonal Naive
h=6
fit.mean=meanf(train_data,h=h)
fit.naive=naive(train_data,h=h)
fit.snaive=snaive(train_data,h=h)

# plot with forecasts and actual values
plot(fit.mean, PI=FALSE,
     main="Forecasts for quarterly")
lines(fit.naive$mean,col=2)
lines(fit.snaive$mean,col=3)
lines(y)
legend("topleft",lty=1,col=c(4,2,3),
       legend=c("Mean method","Naive method","Seasonal naive method"),bty="n")
```



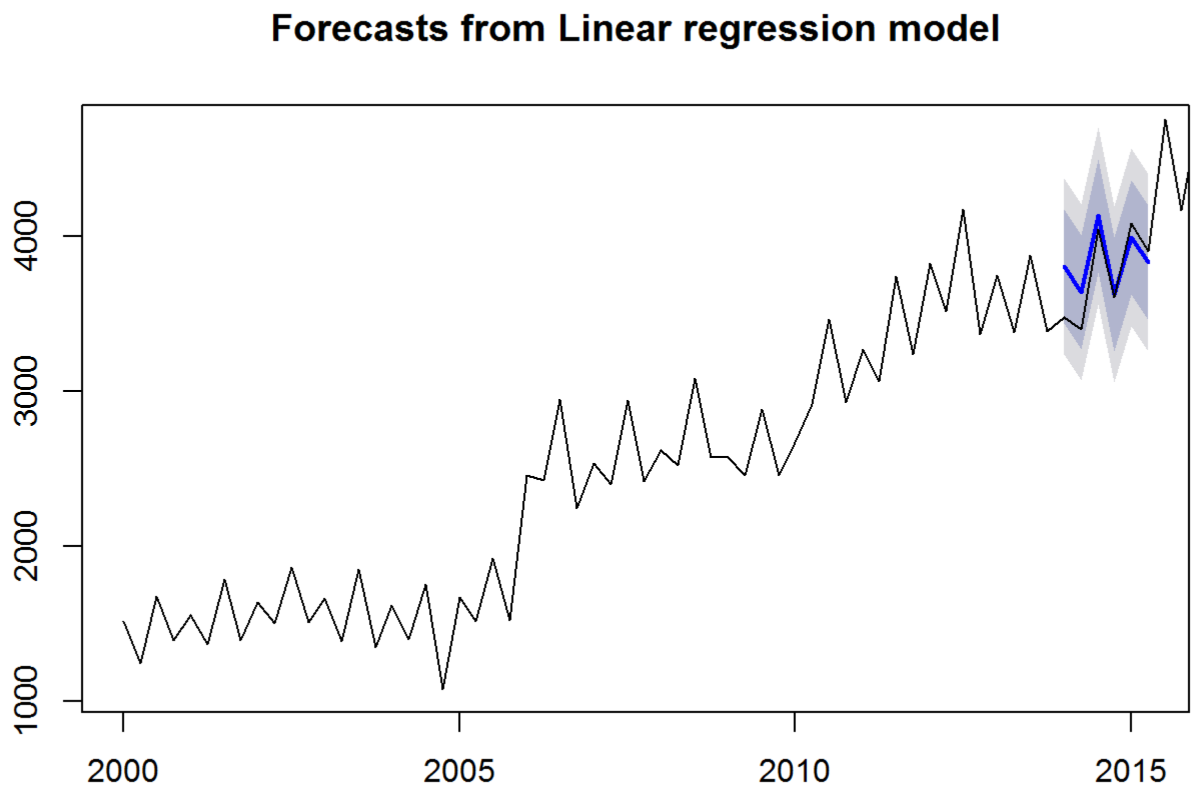

```
#Linear Trend model
reg <- tslm(train_data ~ trend)
fit.tslm=forecast(reg, h=h,level=c(80,95))
summary(fit.tslm)
```

```
##
## Forecast method: Linear regression model
##
## Model Information:
##
## Call:
## tslm(formula = train_data ~ trend)
##
## Coefficients:
## (Intercept)        trend
##     1040.73        47.06
##
##
## Error measures:
##                        ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -4.05727e-15 330.5585 264.7163 -2.184255 13.04242 1.158009
##                   ACF1
## Training set 0.1474735
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2014 Q1       3723.344 3270.858 4175.829 3024.130 4422.557
## 2014 Q2       3770.407 3317.087 4223.727 3069.903 4470.910
## 2014 Q3       3817.470 3363.288 4271.652 3115.635 4519.306
## 2014 Q4       3864.534 3409.463 4319.605 3161.324 4567.743
## 2015 Q1       3911.597 3455.610 4367.584 3206.973 4616.222
## 2015 Q2       3958.661 3501.731 4415.590 3252.579 4664.742
```

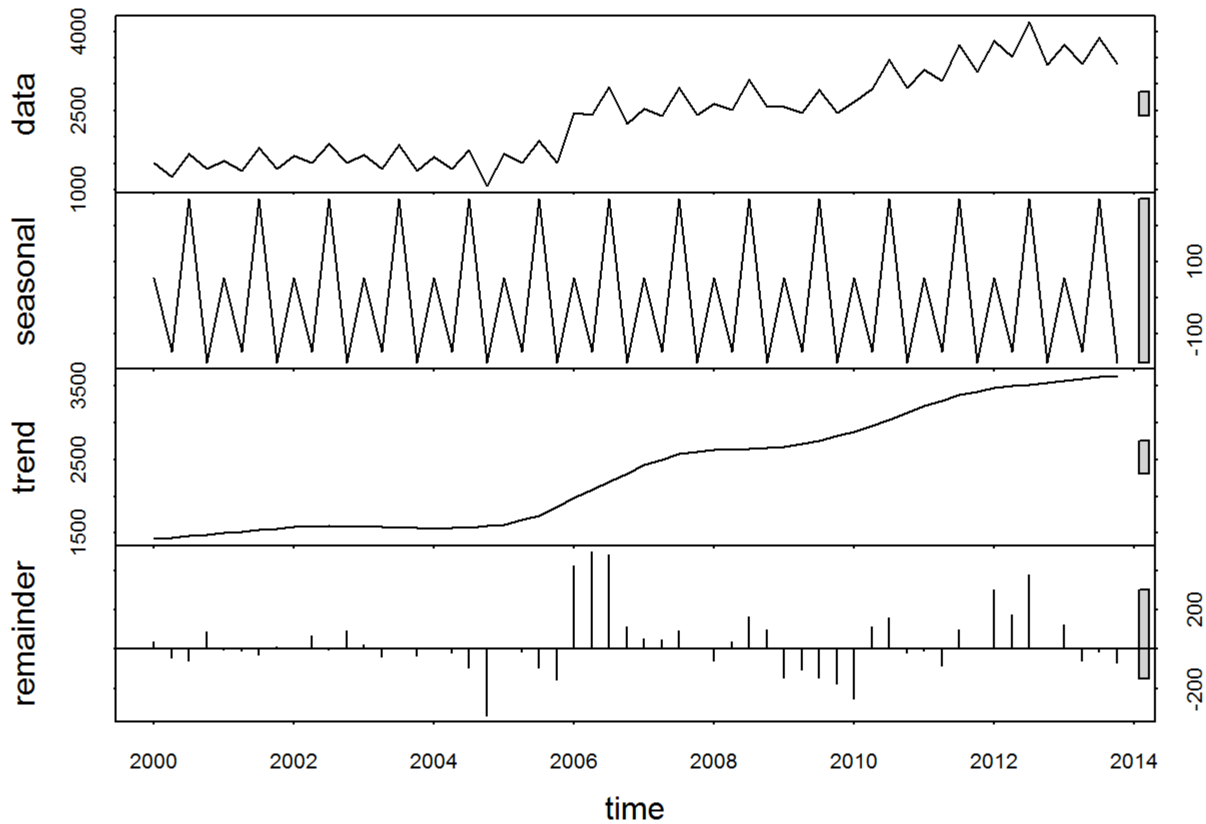
```
plot(fit.tslm, ylab="Adidas Quarterly Sales",
     xlab="t")
lines(fitted(reg),col="blue")
lines(y)
```




```
#trend & seasonal
ts <- tslm(train_data ~ trend + season)
fit.lmts = forecast(ts, h=h)
plot(fit.lmts)
lines(y)
```



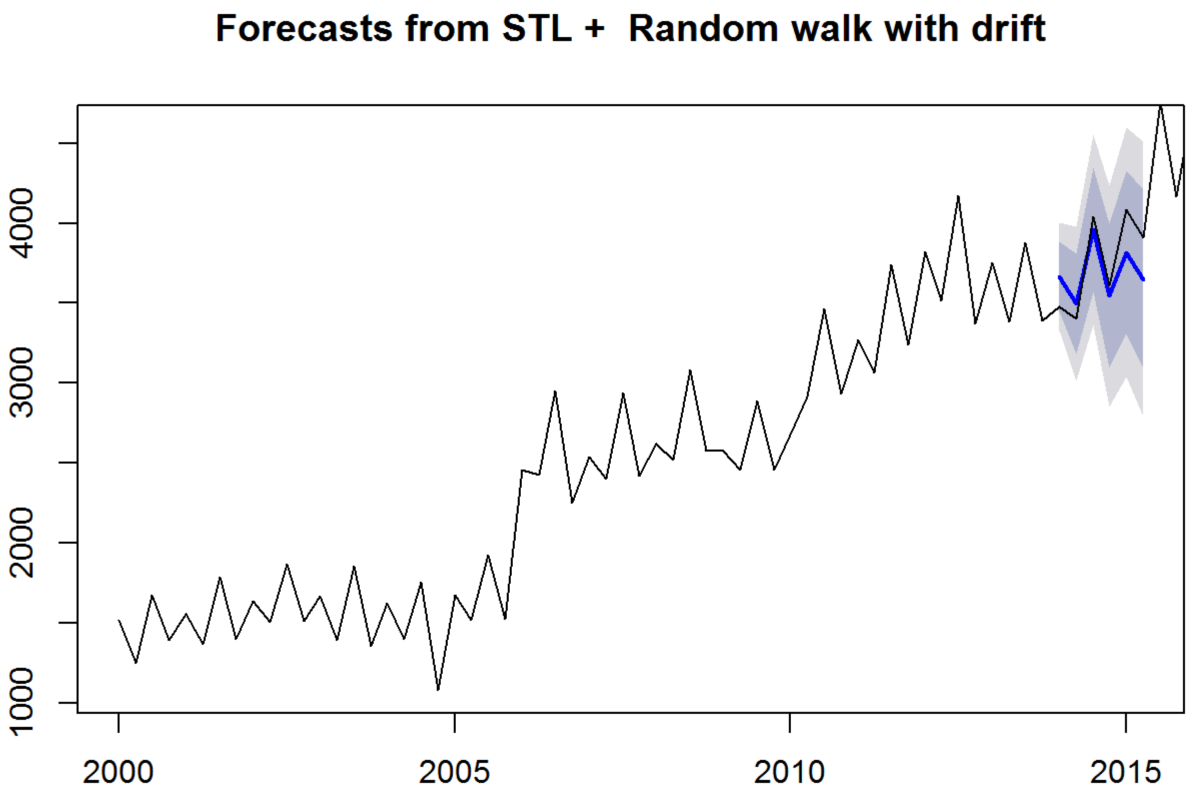

```
#STL Decomposition
y.stl <- stl(train_data, t.window=15, s.window="periodic", robust=TRUE)
plot(y.stl)
```




```
#STL Forecasting
fit.stl <- forecast(y.stl, method="rwdrift", h=h)
summary(fit.stl)
```

```
##
## Forecast method: STL +  Random walk with drift
##
## Model Information:
## Call: rwf(y = x, h = h, drift = TRUE, level = level)
##
## Drift: 38.3479  (se 22.7383)
## Residual sd: 170.1858
##
## Error measures:
##                         ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -7.974611e-12 168.6314 116.8932 -0.5536722 5.136756 0.5113525
##                    ACF1
## Training set -0.1616575
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2014 Q1       3664.484 3444.444 3884.524 3327.962 4001.006
## 2014 Q2       3496.375 3182.475 3810.276 3016.306 3976.445
## 2014 Q3       3960.785 3573.038 4348.531 3367.777 4553.792
## 2014 Q4       3544.392 3092.883 3995.900 2853.869 4234.915
## 2015 Q1       3817.876 3308.885 4326.866 3039.442 4596.309
## 2015 Q2       3649.767 3087.646 4211.888 2790.077 4509.457
```

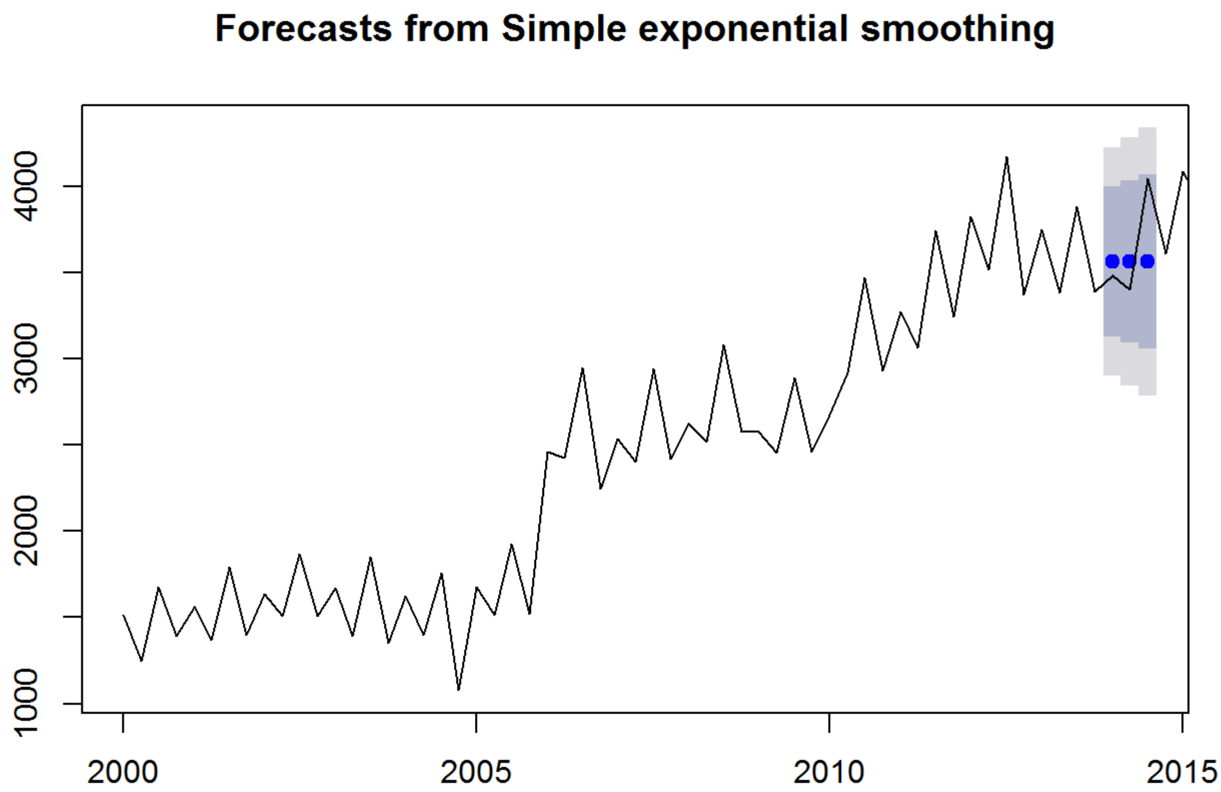
```
plot(fit.stl)
lines(y)
```




```
#SIMPLE EXPONENTIAL SMOOTHING
fit.expo <- ses(train_data, h = 3)
summary(fit.expo)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = train_data, h = 3)
##
##   Smoothing parameters:
##     alpha = 0.426
##
##   Initial states:
##     l = 1465.8592
##
##   sigma:  338.6906
##
##      AIC     AICc      BIC
## 881.7929 882.2544 887.8689
##
## Error measures:
##                    ME     RMSE      MAE      MPE     MAPE     MASE
## Training set 87.89642 332.5876 267.6454 1.968379 11.80674 1.170822
##                    ACF1
## Training set -0.3854825
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2014 Q1       3562.769 3128.719 3996.818 2898.947 4226.590
## 2014 Q2       3562.769 3090.974 4034.564 2841.220 4284.317
## 2014 Q3       3562.769 3056.032 4069.505 2787.782 4337.756
```

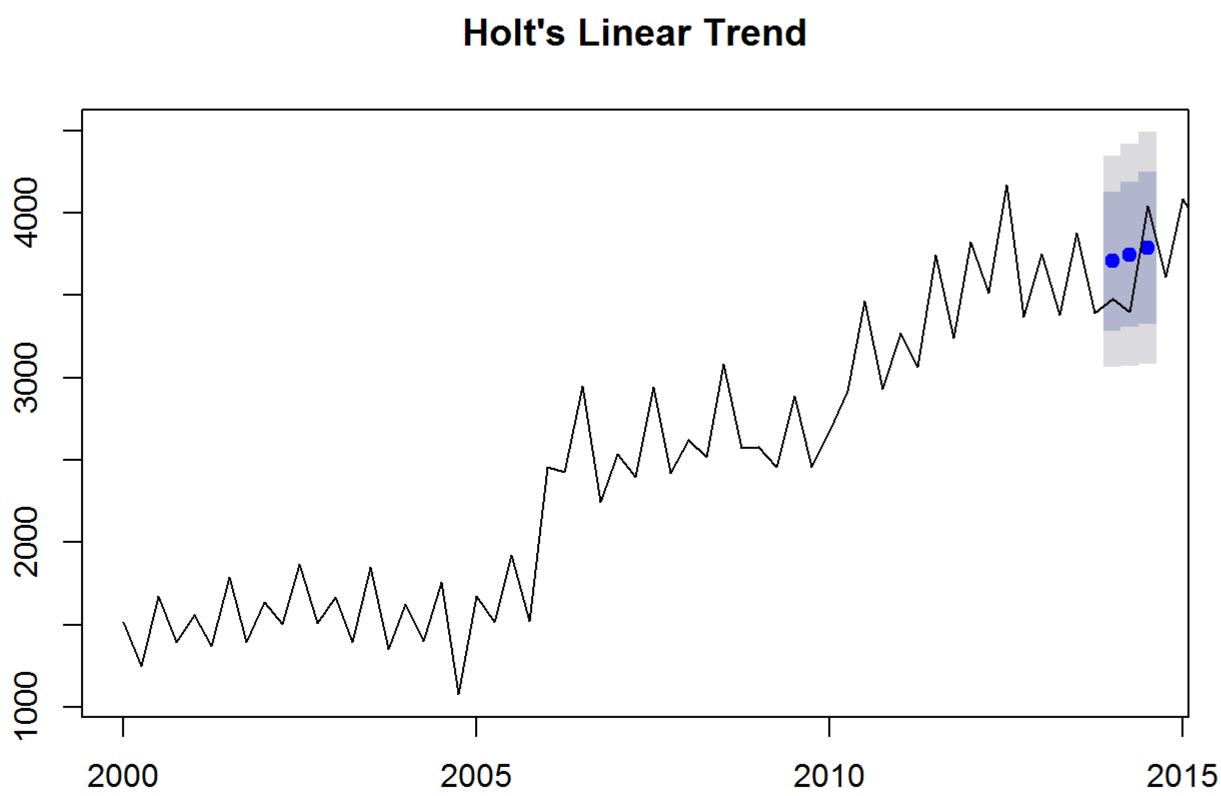
```
plot(fit.expo)
lines(y)
```



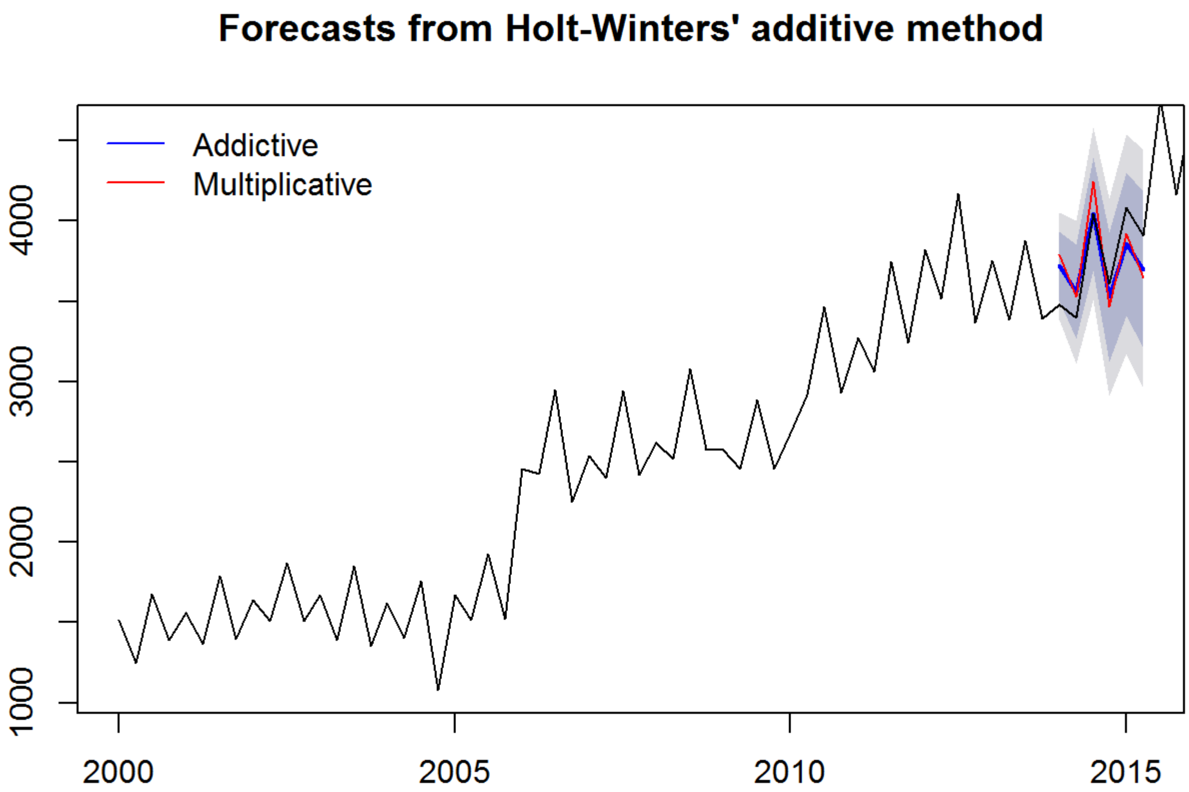

```
#HOLT-WINTERS LINEAR
fit.hlinear <- holt(train_data, h=3)
summary(fit.hlinear)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = train_data, h = 3)
##
##   Smoothing parameters:
##     alpha = 0.3108
##     beta  = 1e-04
##
##   Initial states:
##     l = 1423.3542
##     b = 40.7112
##
##   sigma:  328.1225
##
##      AIC     AICc      BIC
## 880.1290 881.3290 890.2558
##
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -1.989548 316.1868 258.6053 -2.498491 11.88473 1.131276
##                    ACF1
## Training set -0.2567311
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2014 Q1       3705.698 3285.193 4126.204 3062.590 4348.807
## 2014 Q2       3746.399 3306.039 4186.758 3072.927 4419.870
## 2014 Q3       3787.099 3327.732 4246.466 3084.557 4489.640
```

```
plot(fit.hlinear, main = "Holt's Linear Trend")
lines(y)
```



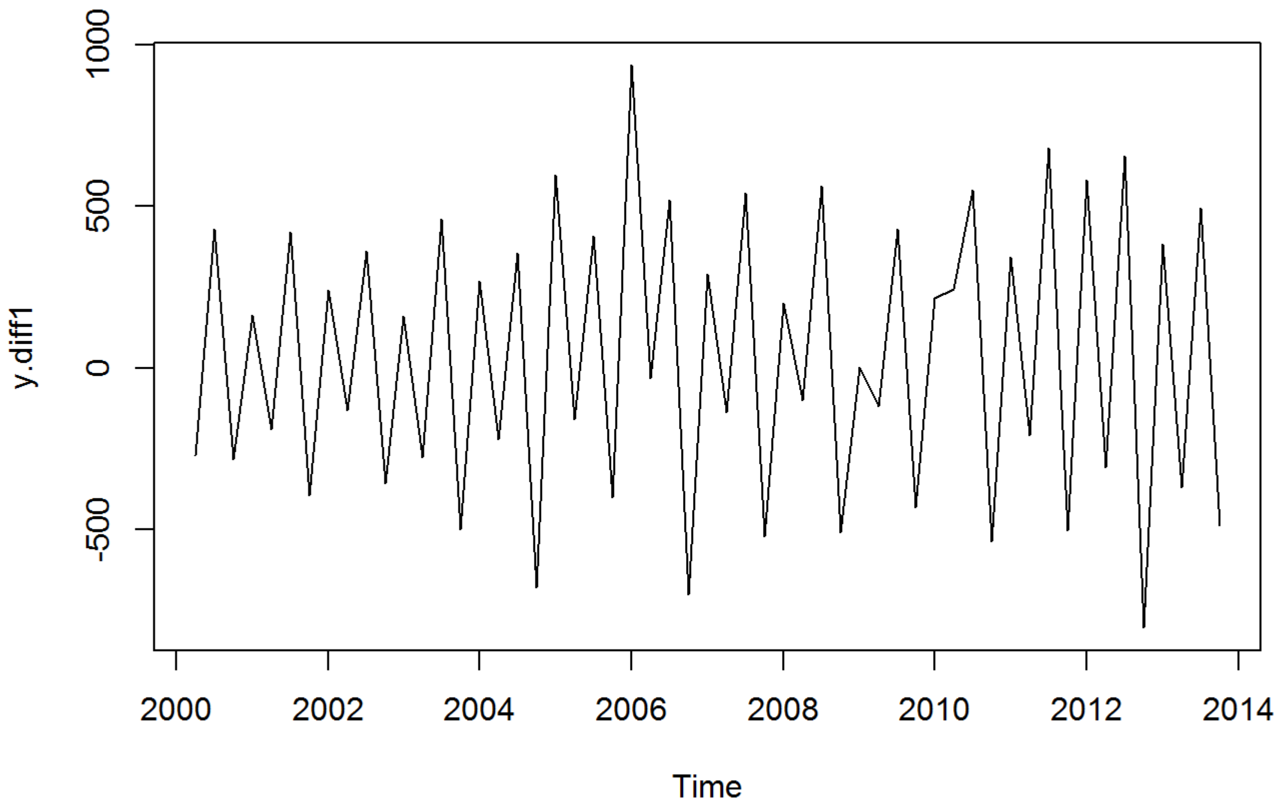

```
#HOLT-WINTERS SEASONAL METHOD:MULTIPLICATIVE AND ADDITIVE
fit.hwm <- hw(train_data, seasonal="multiplicative", h=h)
fit.hwa <- hw(train_data, seasonal="additive", h=h)
plot(fit.hwa, col = 4)
lines(fit.hwm$mean, col = 2)
lines(y)
legend("topleft",lty=1,col=c(4,2),
       legend=c("Addictive","Multiplicative"),bty="n")
```




```
#ARIMA
y.diff1 = diff(train_data, differences = 1)
adf.test(y.diff1, alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  y.diff1
## Dickey-Fuller = -2.4882, Lag order = 3, p-value = 0.3778
## alternative hypothesis: stationary
```

```
plot(y.diff1)
```

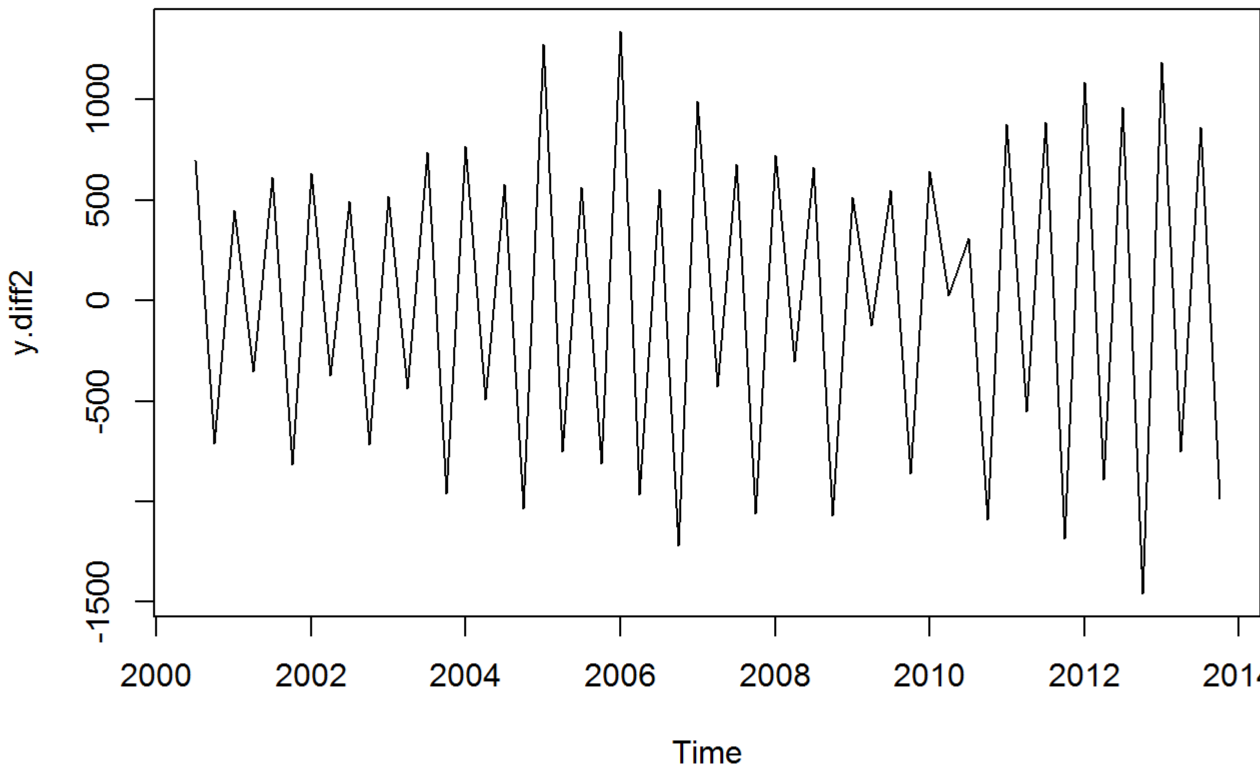



```
y.diff2 = diff(train_data, differences = 2)
adf.test(y.diff2, alternative = "stationary")
```

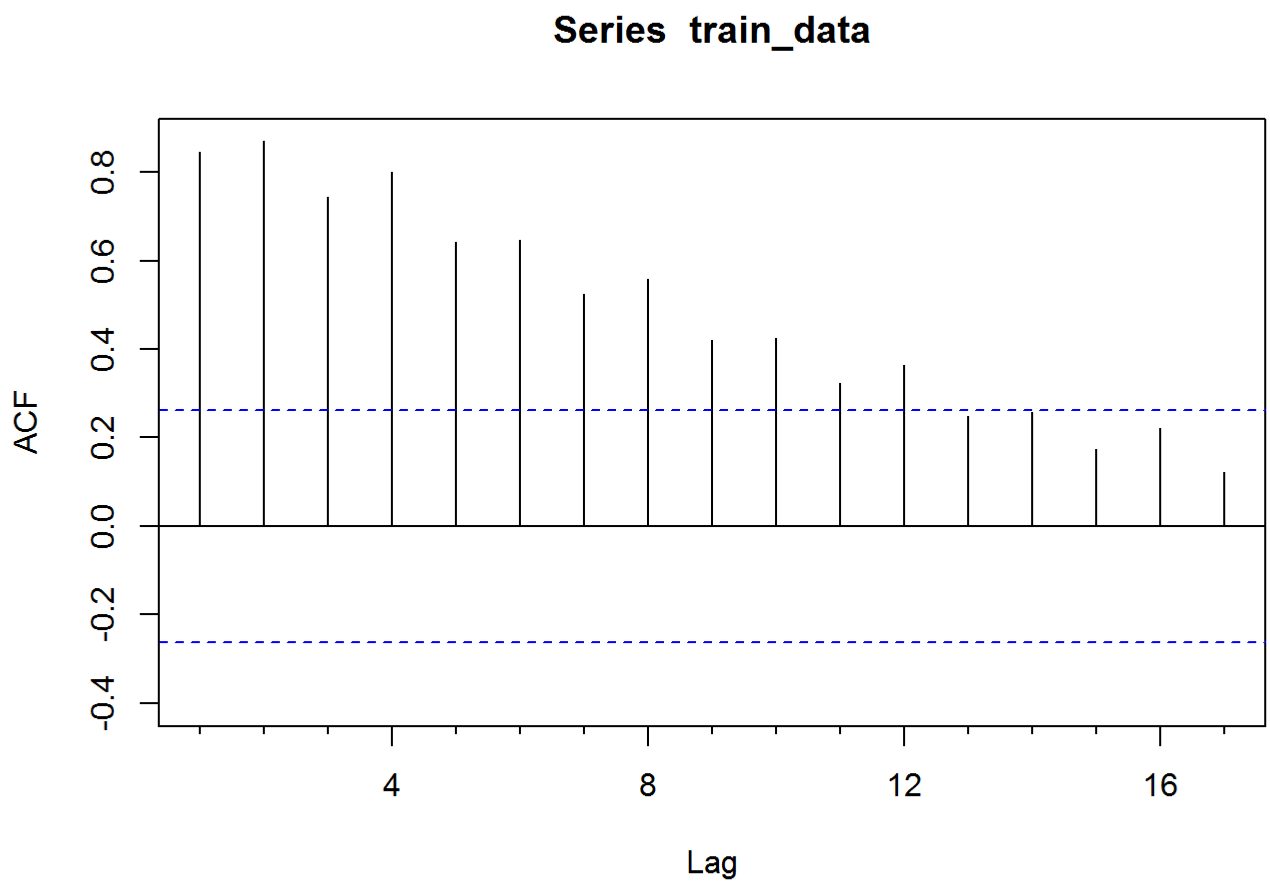
```
## Warning in adf.test(y.diff2, alternative = "stationary"): p-value smaller
## than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  y.diff2
## Dickey-Fuller = -5.1456, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

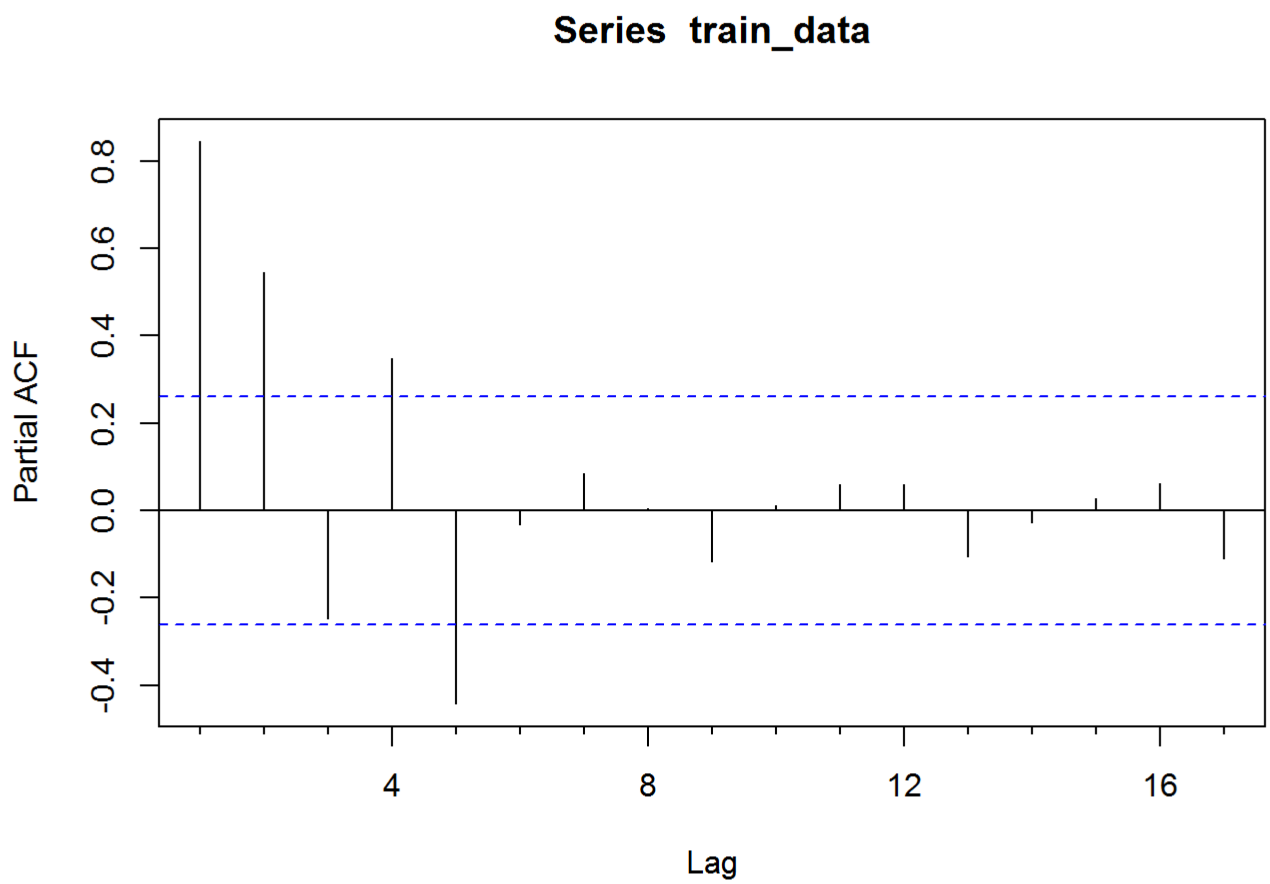
```
plot(y.diff2)
```




```
Acf(train_data)
```



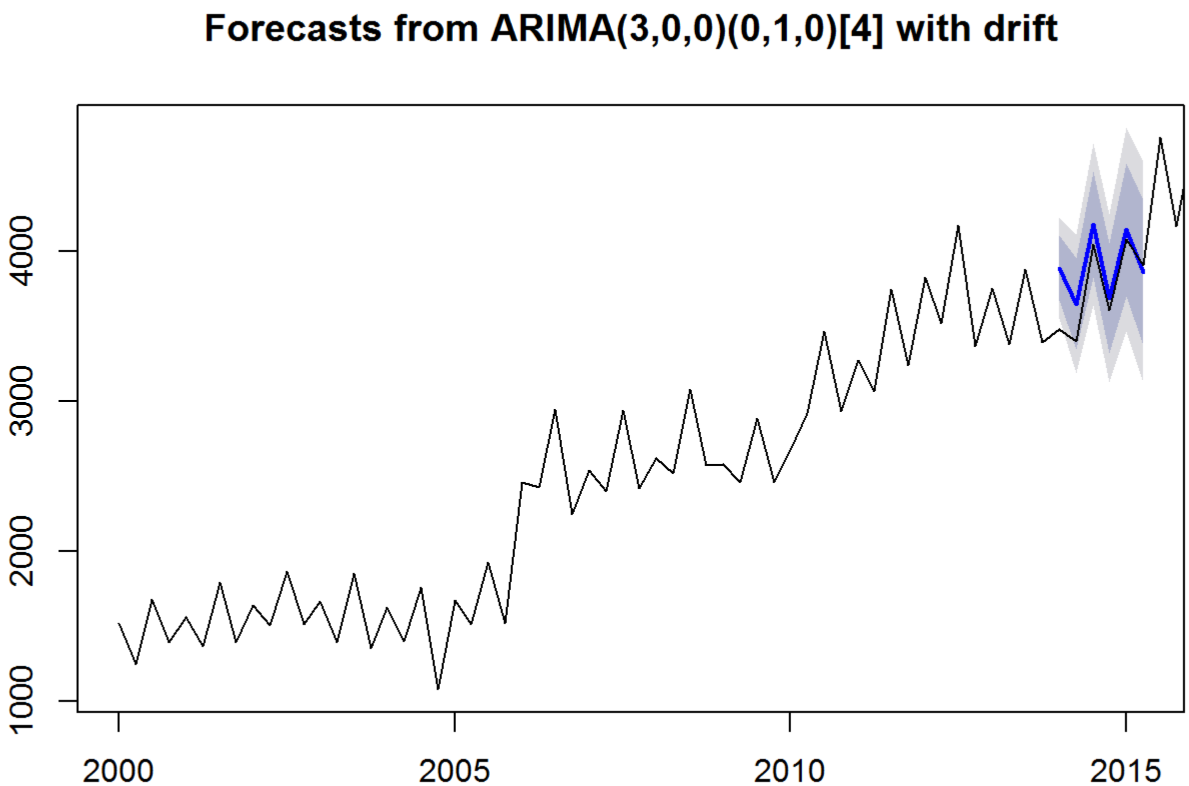

```
Pacf(train_data)
```




```
#Auto ARIMA
y.arima <- auto.arima(train_data)
fit.arima <- forecast(y.arima, h=h)
summary(fit.arima)
```

```
##
## Forecast method: ARIMA(3,0,0)(0,1,0)[4] with drift
##
## Model Information:
## Series: train_data
## ARIMA(3,0,0)(0,1,0)[4] with drift
##
## Coefficients:
##          ar1      ar2      ar3    drift
##       0.9518  -0.0746  -0.2485  42.2554
## s.e.  0.1348   0.1953   0.1367  15.2852
##
## sigma^2 estimated as 29069:  log likelihood=-339.59
## AIC=689.18   AICc=690.49   BIC=698.94
##
## Error measures:
##                    ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 1.006213 157.8494 123.0373 -0.5617156 5.473664 0.5382301
##                    ACF1
## Training set 0.02318496
##
## Forecasts:
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2014 Q1       3889.926 3671.426 4108.427 3555.759 4224.094
## 2014 Q2       3649.406 3347.757 3951.056 3188.074 4110.739
## 2014 Q3       4179.491 3827.375 4531.608 3640.976 4718.007
## 2014 Q4       3685.368 3318.475 4052.261 3124.254 4246.482
## 2015 Q1       4144.244 3699.497 4588.992 3464.062 4824.427
## 2015 Q2       3857.592 3375.351 4339.833 3120.068 4595.116
```

```
plot(fit.arima)
lines(y)
```




```
#Accuracy Summary
a.mean=accuracy(fit.mean,test_data)
a.naive=accuracy(fit.naive,test_data)
a.snaive=accuracy(fit.snaive,test_data)
a.linear=accuracy(fit.tslm,test_data)
a.st=accuracy(fit.lmts,test_data)
a.ses=accuracy(fit.expo, test_data)
a.stl=accuracy(fit.stl, test_data)
a.holt=accuracy(fit.hlinear, test_data)
a.multi=accuracy(fit.hwm, test_data)
a.add=accuracy(fit.hwa, test_data)
a.arima=accuracy(fit.arima, test_data)

a.table<-rbind(a.mean, a.naive, a.snaive, a.linear, a.st, a.ses, a.stl, a.holt, a.multi, a.add,
 a.arima)

row.names(a.table)<-c('Mean training','Mean test', 'Naive training', 'Naive test', 'S. Naive tra
ining', 'S. Naive test' ,'Linear training', 'Linear test','season-trend training', 'season-trend
 test', 'STL training', 'STL test',"ses training", "ses test","Holt's Linear training", "Holt's
 Linear test", 'Muti training', 'Multi test','add training', 'add test','ARIMA training', 'ARIMA
 test')

# order the table according to MASE
a.table<-as.data.frame(a.table)
a.table<-a.table[order(a.table$MASE),]
a.table
```

```
##                                ME      RMSE       MAE         MPE
## add training          5.936575e+00  156.5048  112.4352  -0.1179359
## Muti training         1.308620e+01  166.8032  112.6028  -0.1602071
## ses training         -7.974611e-12  168.6314  116.8932  -0.5536722
## ARIMA training        1.006213e+00  157.8494  123.0373  -0.5617156
## season-trend test    -8.451355e+01  176.0549  139.3619  -2.4960244
## add test              1.955155e+01  175.0263  152.6493   0.2612656
## ses test              6.505367e+01  178.4244  158.6734   1.4694336
## ARIMA test           -1.470048e+02  208.3561  163.4742  -4.1314244
## Multi test           -1.287246e+01  210.9563  200.8146  -0.5256939
## season-trend training  4.061359e-15  253.5765  207.0419  -1.1739508
## Linear test          -8.700208e+01  239.6415  219.6462  -2.7434039
## S. Naive training     1.647885e+02  333.3575  228.5962   5.9459772
## STL test              7.856461e+01  297.1688  242.2562   1.5780522
## S. Naive test         1.643333e+02  298.2493  254.6667   4.0670576
## Holt's Linear training -1.989548e+00  316.1868  258.6053  -2.4984905
## Linear training      -4.057270e-15  330.5585  264.7163  -2.1842548
## STL training          8.789642e+01  332.5876  267.6454   1.9683790
## Holt's Linear test   -1.050652e+02  281.0279  276.3328  -3.4403750
## Naive test            3.630000e+02  452.3074  363.0000   9.1985708
## Naive training        3.407273e+01  431.1145  384.2909  -0.5467924
## Mean training         6.473790e-14  829.4148  716.7819 -13.6030199
## Mean test             1.371964e+03 1398.2485 1371.9643  36.2157926
##                           MAPE      MASE         ACF1 Theil's U
## add training          5.257339 0.4918508 -0.007430929        NA
## Muti training         5.405616 0.4925838 -0.019884166        NA
## ses training          5.136756 0.5113525 -0.161657496        NA
## ARIMA training        5.473664 0.5382301  0.023184964        NA
## season-trend test     3.867160 0.6096424  0.545497353 0.3115035
## add test              4.121876 0.6677684  0.553473346 0.3667693
## ses test              4.181376 0.6941210  0.458407323 0.4051990
## ARIMA test            4.552960 0.7151221  0.378847178 0.3348381
## Multi test            5.376758 0.8784688  0.385913179 0.4257386
## season-trend training 10.909320 0.9057104  0.787718765        NA
## Linear test           6.009932 0.9608483 -0.438685321 0.5751233
## S. Naive training     9.220278 1.0000000  0.780659978        NA
## STL test              6.355203 1.0597564 -0.236360619 0.7812646
## S. Naive test         6.662843 1.1140462  0.359303768 0.6720790
## Holt's Linear training 11.884729 1.1312760 -0.256731105        NA
## Linear training      13.042421 1.1580085  0.147473485        NA
## STL training         11.806736 1.1708221 -0.385482533        NA
## Holt's Linear test    7.675478 1.2088252 -0.285759503 0.6549705
## Naive test            9.198571 1.5879532 -0.101774408 1.1757093
## Naive training       17.396986 1.6810909 -0.764711029        NA
## Mean training        34.890539 3.1355816  0.844401211        NA
## Mean test            36.215793 6.0016945 -0.101774408 3.4094174
```