Workshop - KE

Time Series Forecasting (AR & MA) using R

```
Data Source:
1. Hotel Data - R.csv and t6-4 hotel_subset.csv (same data with and w
ithout dummy)
2. data_1995.csv
3. AmtrakBig.csv
4. cola-data.csv
5. Electricity raw data.csv
```

In [4]:

```
# setup the infrastructure
pacman::p_load(tidyverse, lubridate,zoo,forecast, fUnitRoots)
```

In [5]:

```
# set path to data folder on your computer
# pay attention to the '/'
setwd("C:/Users/isspcs/Desktop/workshop-data")
```

A. Linear Regression

```
    Let us use the hotel data we saw during lecture to quickly develo
p a linear regression model
```

In [78]:

```
data = read.csv('Hotel Data - R.csv')
```

In [79]:

```
# check the data types of all columns ?
sapply(data, class)
```

t
'integer'
Yt
'integer'
Yt_trans
'numeric'
t.1
'integer'
S1
'integer'
S2
'integer'
S3
'integer'
S4
'integer'
S5
'integer'
S6
'integer'
S7
'integer'
S8
'integer'
S9
'integer'
S10
'integer'
S11
'integer'

In [80]:

```
# get strucutre of data ?
str(data)
```

```
'data.frame':	168 obs. of  15 variables:
 $ t       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Yt      : int  501 488 504 578 545 632 728 725 585 542 ...
 $ Yt_trans: num  4.73 4.7 4.74 4.9 4.83 ...
 $ t.1     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ S1      : int  1 0 0 0 0 0 0 0 0 0 ...
 $ S2      : int  0 1 0 0 0 0 0 0 0 0 ...
 $ S3      : int  0 0 1 0 0 0 0 0 0 0 ...
 $ S4      : int  0 0 0 1 0 0 0 0 0 0 ...
 $ S5      : int  0 0 0 0 1 0 0 0 0 0 ...
 $ S6      : int  0 0 0 0 0 1 0 0 0 0 ...
 $ S7      : int  0 0 0 0 0 0 1 0 0 0 ...
 $ S8      : int  0 0 0 0 0 0 0 1 0 0 ...
 $ S9      : int  0 0 0 0 0 0 0 0 1 0 ...
 $ S10     : int  0 0 0 0 0 0 0 0 0 1 ...
 $ S11     : int  0 0 0 0 0 0 0 0 0 0 ...
```

In [81]:

```
# see some entries from begining ?
head(data, 3)
```

t	Yt	Yt_trans	t.1	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
1	501	4.731071	1	1	0	0	0	0	0	0	0	0	0	0
2	488	4.700077	2	0	1	0	0	0	0	0	0	0	0	0
3	504	4.738137	3	0	0	1	0	0	0	0	0	0	0	0

In [82]:

```
# model the data
model = lm(Yt_trans ~ t.1+S1+S2+S3+S4+S5+S6+S7+S8+S9+S10+S11, data=data)
```

In [83]:

```
# summary of model?
summary(model)
```

```
Call:
lm(formula = Yt_trans ~ t.1 + S1 + S2 + S3 + S4 + S5 + S6 + S7
+
    S8 + S9 + S10 + S11, data = data)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.068082 -0.018755  0.001425  0.018808  0.079133 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.807e+00  8.463e-03 568.070  < 2e-16 ***
t.1          3.515e-03  4.449e-05  79.009  < 2e-16 ***
S1          -5.247e-02  1.055e-02  -4.971 1.75e-06 ***
S2          -1.408e-01  1.055e-02 -13.342  < 2e-16 ***
S3          -1.071e-01  1.055e-02 -10.151  < 2e-16 ***
S4           4.988e-02  1.055e-02   4.728 5.05e-06 ***
S5           2.542e-02  1.055e-02   2.410   0.0171 *  
S6           1.902e-01  1.055e-02  18.031  < 2e-16 ***
S7           3.825e-01  1.055e-02  36.266  < 2e-16 ***
S8           4.134e-01  1.054e-02  39.201  < 2e-16 ***
S9           7.142e-02  1.054e-02   6.773 2.47e-10 ***
S10          5.064e-02  1.054e-02   4.803 3.66e-06 ***
S11         -1.419e-01  1.054e-02 -13.463  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0279 on 155 degrees of freedom
Multiple R-squared:  0.9884,    Adjusted R-squared:  0.9875 
F-statistic:  1098 on 12 and 155 DF,  p-value: < 2.2e-16
```

In [84]:

```
# since y_trans is y^0.25
data$predicted = (model$fitted.values)**4
```

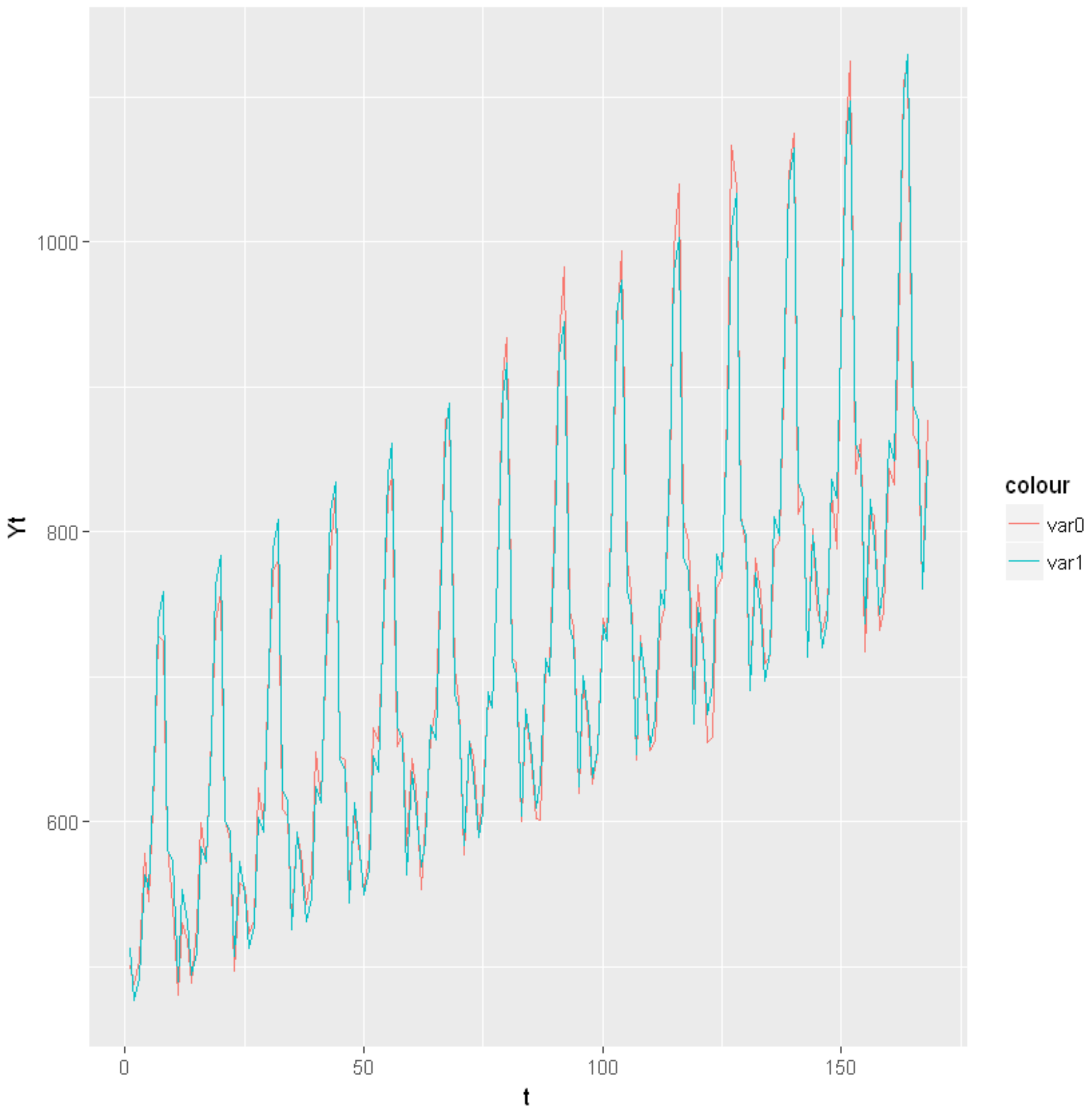
In [85]:

```
head(data)
```

t	Yt	Yt_trans	t.1	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	predic
1	501	4.731071	1	1	0	0	0	0	0	0	0	0	0	0	512.6
2	488	4.700077	2	0	1	0	0	0	0	0	0	0	0	0	477.0
3	504	4.738137	3	0	0	1	0	0	0	0	0	0	0	0	492.4
4	578	4.903227	4	0	0	0	1	0	0	0	0	0	0	0	563.0
5	545	4.831691	5	0	0	0	0	1	0	0	0	0	0	0	553.4
6	632	5.013942	6	0	0	0	0	0	1	0	0	0	0	0	634.3

In [86]:

```
ggplot(data, aes(t)) +
  geom_line(aes(y = Yt, colour = "var0")) +
  geom_line(aes(y = predicted, colour = "var1"))
```

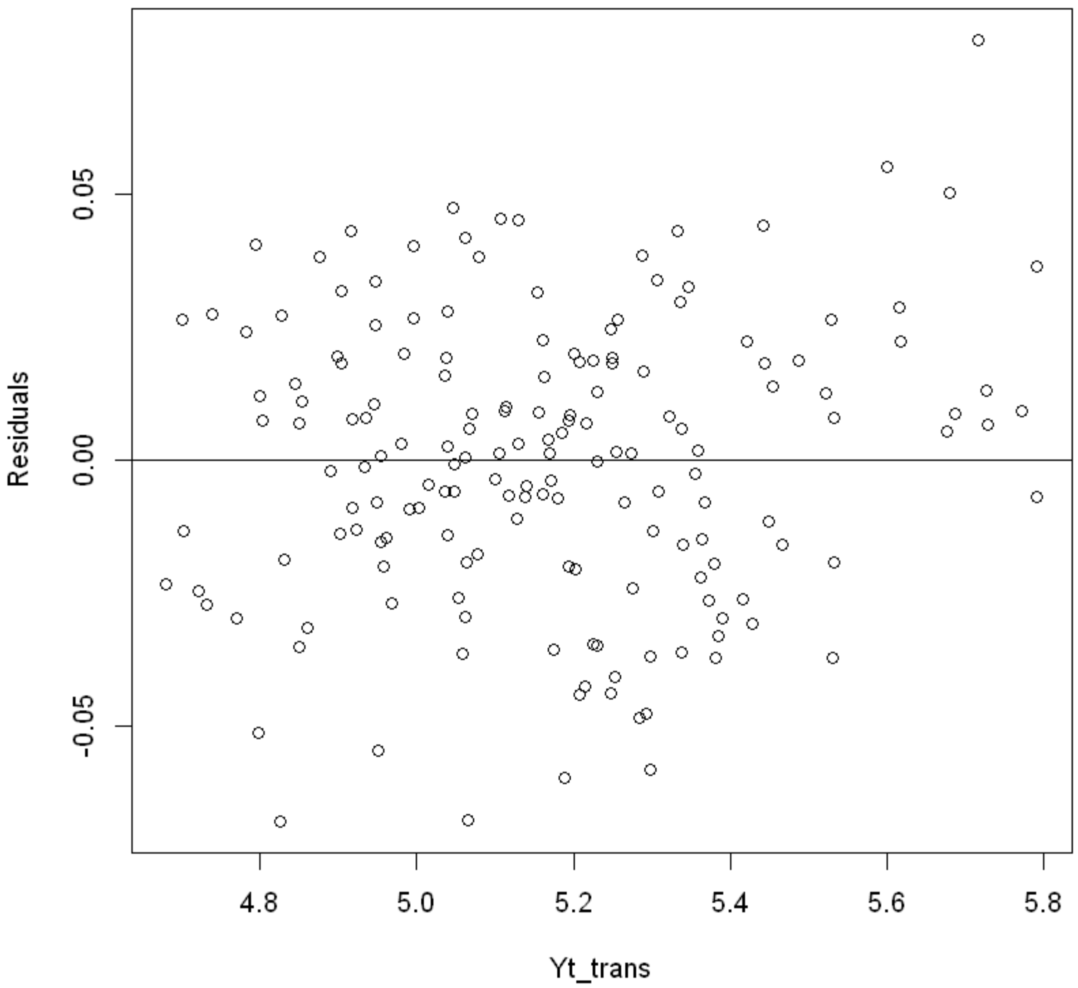


In [87]:

```
# check the residuals (errors)
res = residuals(model)
```

In [88]:

```
plot(jitter(res)~jitter(Yt_trans), ylab="Residuals", xlab="Yt_trans", data=da
abline(0,0)
```



Internal dummy

In [89]:

```
# let us use a file where R can create dummy internally
# file: t6-4 hotel_subset.csv

## Read file
data = read.csv('t6-4 hotel_subset.csv')

head(data, n=2)
#check the data types of all columns
sapply(data, class)
```

t	Month	Hotel.Occupancy..Y.	logY
1	Jan	501	2.700
2	Feb	488	2.688

t
'integer'
Month
'factor'
Hotel.Occupancy..Y.
'integer'
logY
'numeric'

In [90]:

```
data$newY = (data$Hotel.Occupancy..Y.)**0.25
model = lm(data$newY ~ t+Month, data=data)
summary(model)
```

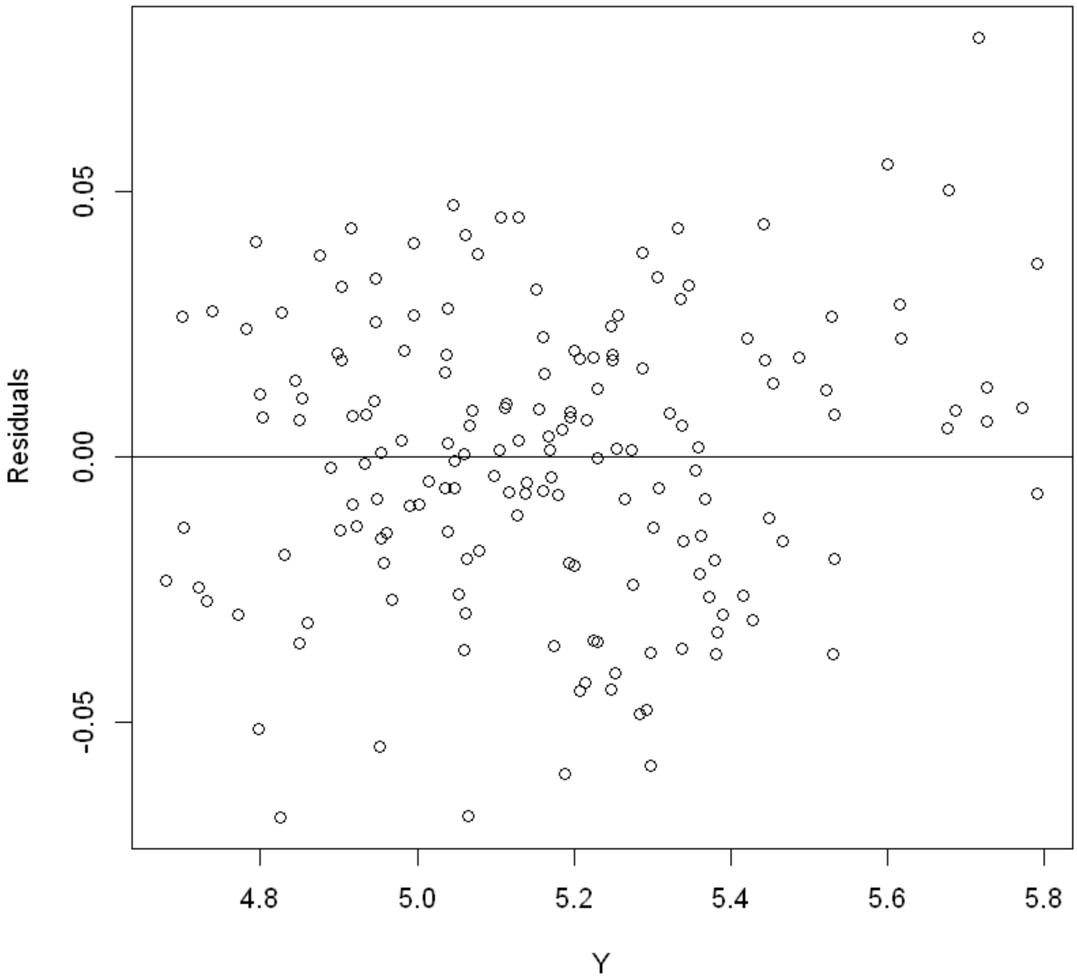
```
Call:
lm(formula = data$newY ~ t + Month, data = data)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.068082 -0.018755  0.001425  0.018808  0.079133 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.857e+00  8.300e-03 585.200  < 2e-16 ***
t            3.515e-03  4.449e-05  79.009  < 2e-16 ***
MonthAug     3.635e-01  1.054e-02  34.471  < 2e-16 ***
MonthDec    -4.988e-02  1.055e-02  -4.728 5.05e-06 ***
MonthFeb    -1.907e-01  1.054e-02 -18.084  < 2e-16 ***
MonthJan    -1.023e-01  1.054e-02  -9.707  < 2e-16 ***
MonthJul     3.326e-01  1.054e-02  31.541  < 2e-16 ***
MonthJun     1.403e-01  1.054e-02  13.305  < 2e-16 ***
MonthMar    -1.570e-01  1.054e-02 -14.889  < 2e-16 ***
MonthMay    -2.446e-02  1.054e-02  -2.320   0.0216 *  
MonthNov    -1.918e-01  1.055e-02 -18.186  < 2e-16 ***
MonthOct     7.593e-04  1.055e-02   0.072   0.9427    
MonthSep     2.154e-02  1.055e-02   2.042   0.0428 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0279 on 155 degrees of freedom
Multiple R-squared:  0.9884,	Adjusted R-squared:  0.9875 
F-statistic:  1098 on 12 and 155 DF,  p-value: < 2.2e-16
```

In [91]:

```
# observe that the base level is not January but April as R chooses it intern
```

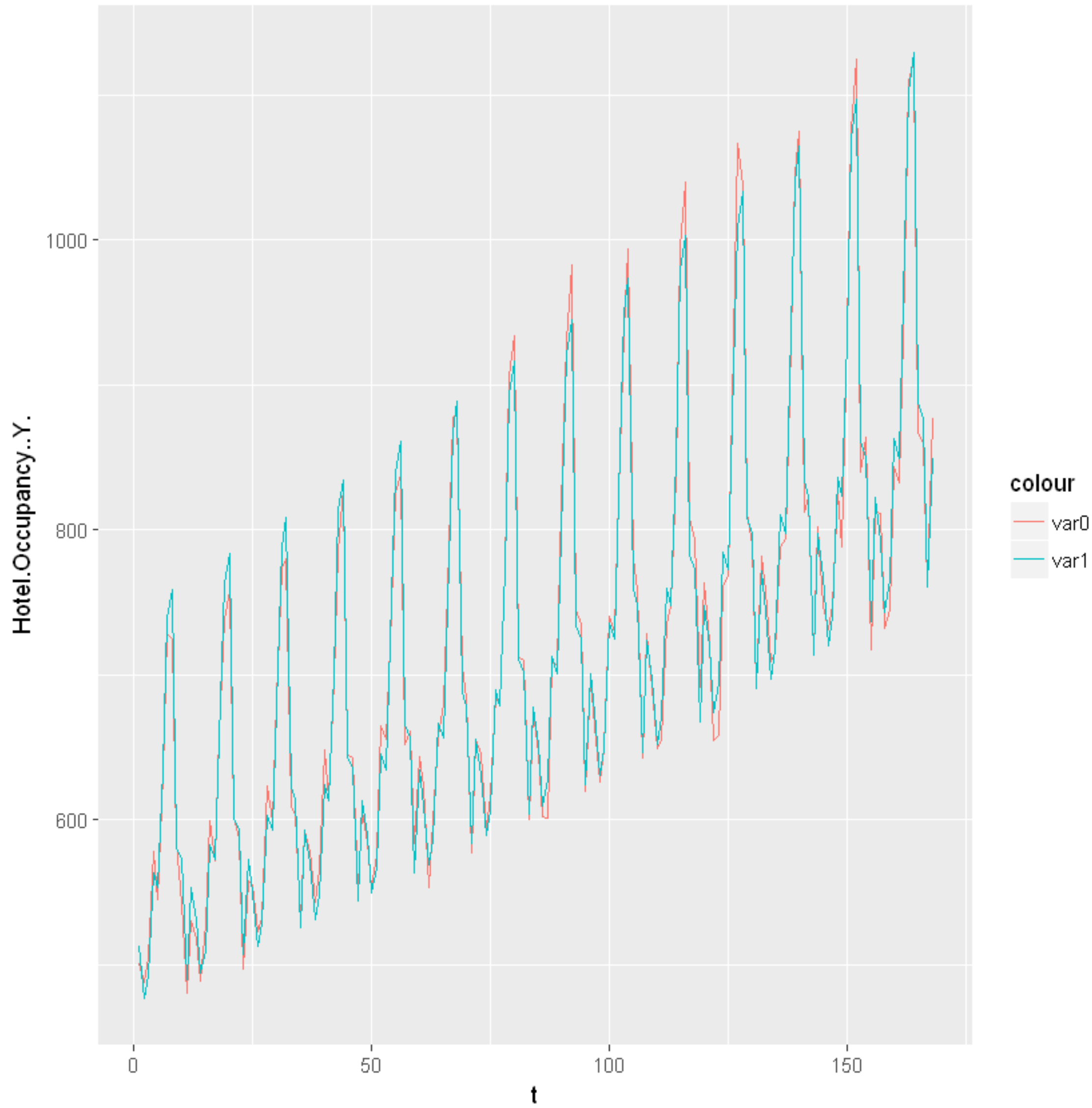
In [92]:

```
res <- residuals(model)
plot(jitter(res)~jitter(newY), ylab="Residuals", xlab="Y", data=data)
abline(0,0)
```



In [93]:

```
# since y_trans is y^0.25
data$predicted = (model$fitted.values)**4
ggplot(data, aes(t)) +
  geom_line(aes(y = Hotel.Occupancy..Y., colour = "var0")) +
  geom_line(aes(y = predicted, colour = "var1"))
```



B. Moving Averages

Let us use the data we saw during lecture for SP500 to perform a simple MA in R

Read in data

```
Data Import by R
R can import various data sources - txt, csv files and also has API t
o several data sources.
We illustrate with a csv data file.
The following reads in the data into a R variable 'dataframe' - named
 series.
```

In [94]:

```
series = read.csv('data_1995.csv')
```

In [95]:

```
# have a look at your data
head(series, n =4)
```

Date	SP500	Dividend	Earnings	Consumer.Price.Index	Long.Interest.Rate
1/1/1995	465.25	13.18	31.25	150.3	7.78
1/2/1995	481.92	13.18	31.90	150.9	7.47
1/3/1995	493.15	13.17	32.55	151.4	7.20
1/4/1995	507.91	13.24	33.18	151.9	7.06

In [96]:

```
# extract the first 2 columns and discard the rest for now
series = series[,1:2]
```

In [97]:

```
head(series)
```

Date	SP500
1/1/1995	465.25
1/2/1995	481.92
1/3/1995	493.15
1/4/1995	507.91
1/5/1995	523.81
1/6/1995	539.35

check if it is a valid time series object using function: is.ts()

```
    - if not then we need to do some intial data processing
```

In [98]:

```
is.ts(series)
```

FALSE

Since it is not a valid TS object we need to change it to a TS object

In [99]:

```
sp500 = ts(series$SP500, frequency=12, start=c(1995, 1))
```

In [100]:

```
# how it looks now internally
sp500
```

```
         Jan     Feb     Mar     Apr     May     Jun     Jul
  Aug     Sep
1995  465.25  481.92  493.15  507.91  523.81  539.35  557.37  5
59.11  578.77
1996  614.42  649.54  647.07  647.17  661.23  668.50  644.07  6
62.68  674.88
1997  766.22  798.39  792.16  763.93  833.09  876.29  925.29  9
27.24  937.02
1998  963.36 1023.74 1076.83 1112.20 1108.42 1108.39 1156.58 10
74.62 1020.64
1999 1248.77 1246.58 1281.66 1334.76 1332.07 1322.55 1380.99 13
27.49 1318.17
2000 1425.59 1388.87 1442.21 1461.36 1418.48 1461.96 1473.00 14
85.46 1468.05
2001 1335.63 1305.75 1185.85 1189.84 1270.37 1238.71 1204.45 11
78.50 1044.64
2002 1140.21 1100.67 1153.79 1111.93 1079.25 1014.02  903.59  9
12.55  867.81
2003  895.84  837.03  846.63  890.03  935.96  988.00  992.54  9
89.53 1019.44
         Oct     Nov     Dec
1995  582.92  595.53  614.57
1996  701.46  735.67  743.25
1997  951.16  938.92  962.37
1998 1032.47 1144.43 1190.05
1999 1300.01 1391.00 1428.68
2000 1390.14 1378.04 1330.93
2001 1076.59 1129.68 1144.93
2002  854.63  909.93  899.18
2003 1038.73 1049.90 1080.64
```

In [101]:

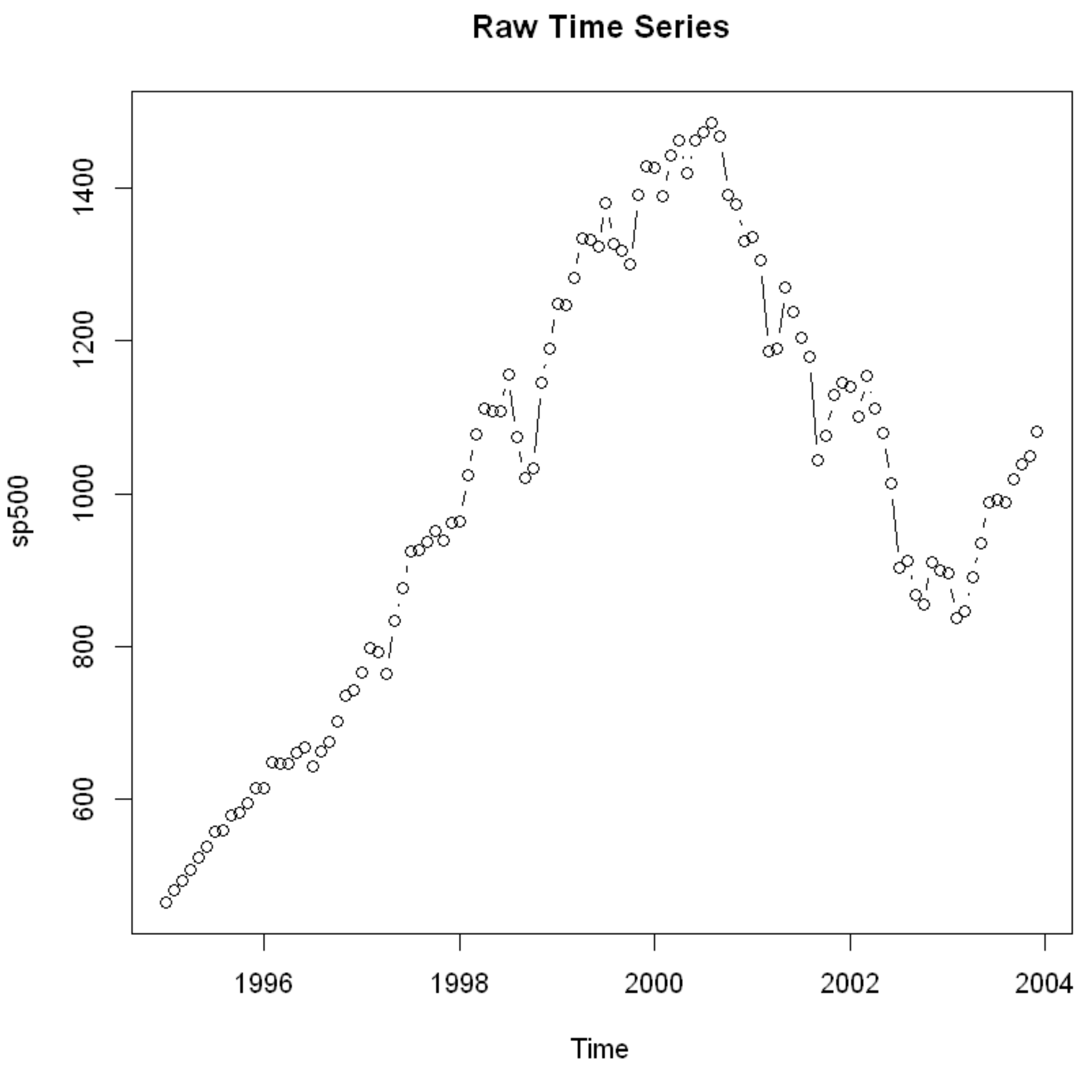
```
is.ts(sp500)
```

TRUE

plot your TS using ts.plot() command

In [102]:

```
ts.plot(sp500, main='Raw Time Series', type='b') # default type is 'l'
```



In [103]:

```
# Let us implement rolling mean
series2 = series %>%
  select(Date, SP500)%>%
  mutate(MA = rollmeanr(SP500, 2, fill = NA))
```

In [104]:

```
head(series2, n=4)
```

Date	SP500	MA
1/1/1995	465.25	NA
1/2/1995	481.92	473.585
1/3/1995	493.15	487.535
1/4/1995	507.91	500.530

In [105]:

```
is.ts(series2)
```

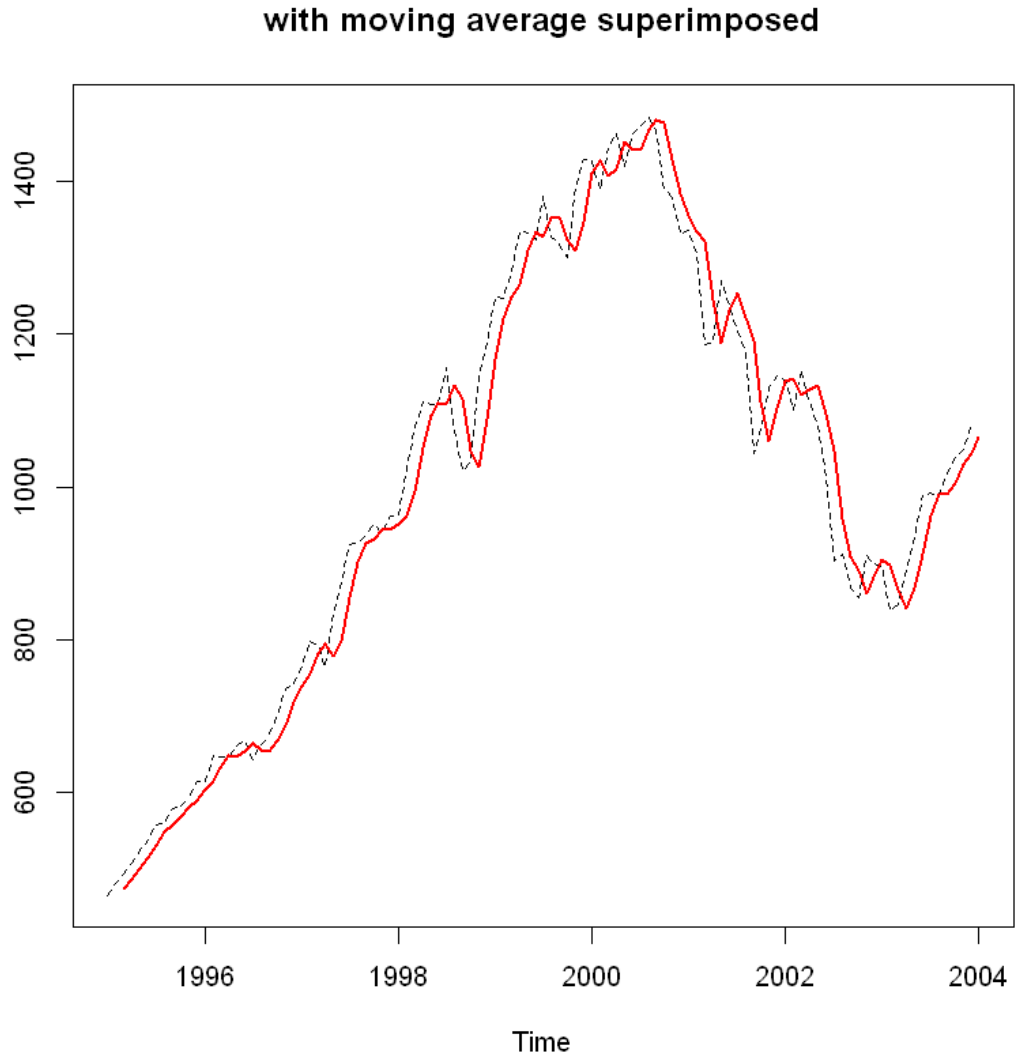
FALSE

In [106]:

```
#convert to a TS object
sp500_ma = ts(series2$MA, frequency=12, start=c(1995, 2))
```

In [107]:

```
ts.plot(sp500,sp500_ma, lty=2:1, col=1:2, lwd=1:2, main="with moving average
```



C. Forecating using AR and MA

```
Objectives of learning:
1. ARIMA (1,0,0) for AR
2. ARIMA (0,0,1) for MA
3. Sationarity test & ARIMA
```

In [108]:

```
## Read file "AmtrakBig.csv"
series = read.csv('AmtrakBig.csv')
head(series, n=4)
```

Month	Ridership	t	Season
Jan-91	1709	1	Jan
Feb-91	1621	2	Feb
Mar-91	1973	3	Mar
Apr-91	1812	4	Apr

In [109]:

```
# Once you have read the time series data into R,
# the next step is to store the data in a time series object in R,
# so that you can use R functions for analysing time series data.
# For monthly time series data, you set frequency=12,
# while for quarterly time series data, you set frequency=4
Rider = ts(series$Ridership, frequency=12, start=c(1991, 1))
Rider
```

```
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  De
c
1991 1709 1621 1973 1812 1975 1862 1940 2013 1596 1725 1676 181
4
1992 1615 1557 1891 1956 1885 1623 1903 1997 1704 1810 1862 187
5
1993 1705 1619 1837 1957 1917 1882 1933 1996 1673 1753 1720 173
4
1994 1563 1574 1903 1834 1831 1776 1868 1907 1686 1779 1776 178
3
1995 1548 1497 1798 1733 1772 1761 1792 1875 1571 1647 1673 165
7
1996 1382 1361 1559 1608 1697 1693 1836 1943 1551 1687 1576 170
0
1997 1397 1372 1708 1655 1763 1776 1934 2008 1616 1774 1732 179
7
1998 1570 1413 1755 1825 1843 1826 1968 1922 1670 1791 1817 184
7
1999 1599 1549 1832 1840 1846 1865 1966 1949 1607 1804 1850 183
6
2000 1542 1617 1920 1971 1992 2010 2054 2097 1824 1977 1981 200
0
2001 1683 1663 2008 2024 2047 2073 2127 2203 1708 1951 1974 198
5
2002 1760 1771 2020 2048 2069 1994 2075 2027 1734 1917 1858 199
6
2003 1778 1749 2066 2099 2105 2130 2223 2174 1931 2121 2076 214
1
2004 1832 1838 2132
```

In [110]:

```
#other useful commands
start(Rider)
end(Rider)

frequency(Rider)

#deltat(Rider)
cycle(Rider)

summary(Rider)
```

1991 1

2004 3

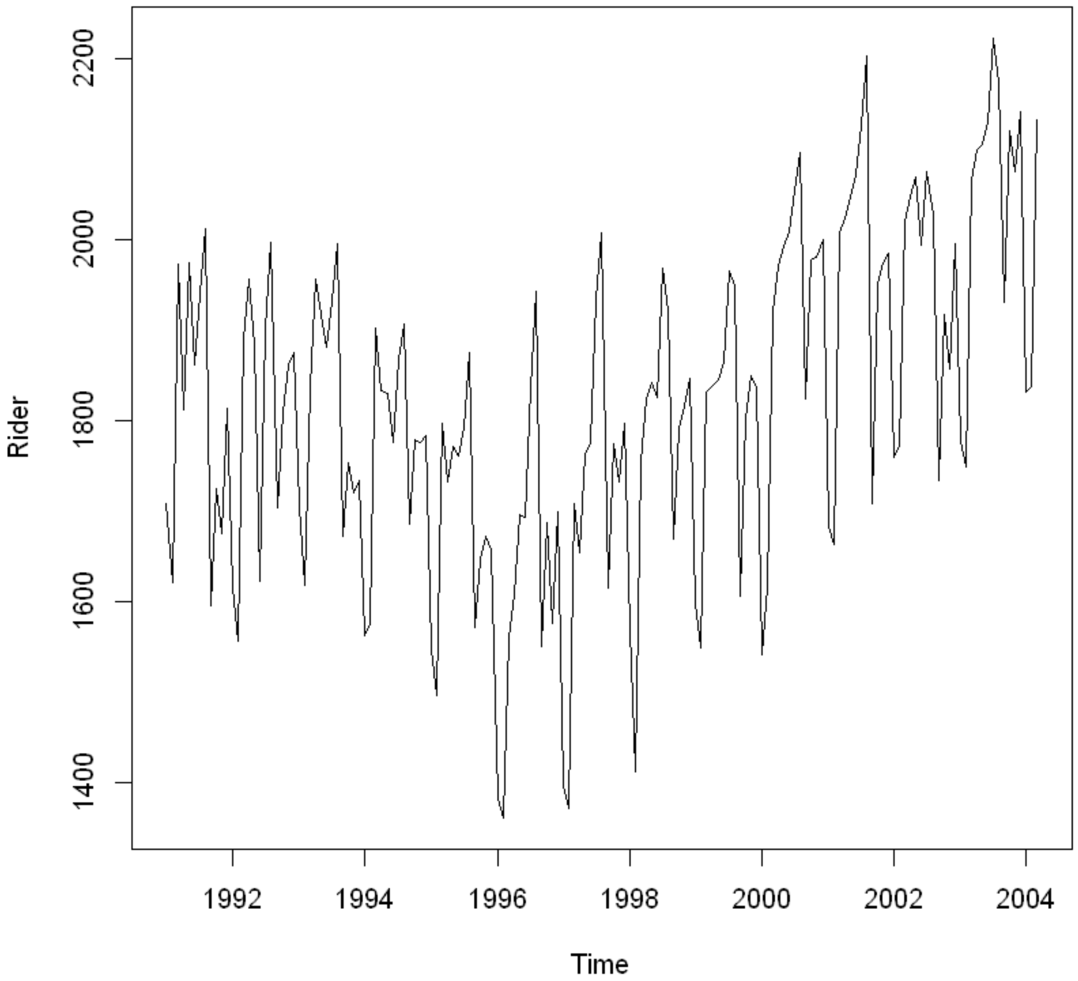
12

```
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1991   1   2   3   4   5   6   7   8   9  10  11  12
1992   1   2   3   4   5   6   7   8   9  10  11  12
1993   1   2   3   4   5   6   7   8   9  10  11  12
1994   1   2   3   4   5   6   7   8   9  10  11  12
1995   1   2   3   4   5   6   7   8   9  10  11  12
1996   1   2   3   4   5   6   7   8   9  10  11  12
1997   1   2   3   4   5   6   7   8   9  10  11  12
1998   1   2   3   4   5   6   7   8   9  10  11  12
1999   1   2   3   4   5   6   7   8   9  10  11  12
2000   1   2   3   4   5   6   7   8   9  10  11  12
2001   1   2   3   4   5   6   7   8   9  10  11  12
2002   1   2   3   4   5   6   7   8   9  10  11  12
2003   1   2   3   4   5   6   7   8   9  10  11  12
2004   1   2   3
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1361    1698    1831    1822    1967    2223
```

In [111]:

```
ts.plot(Rider)
```



In [ ]:

```
```

AR(1)

In [112]:

```
# AR using ARIMA

arima_100 = arima(Rider, order=c(1,0,0))
arima_100
```

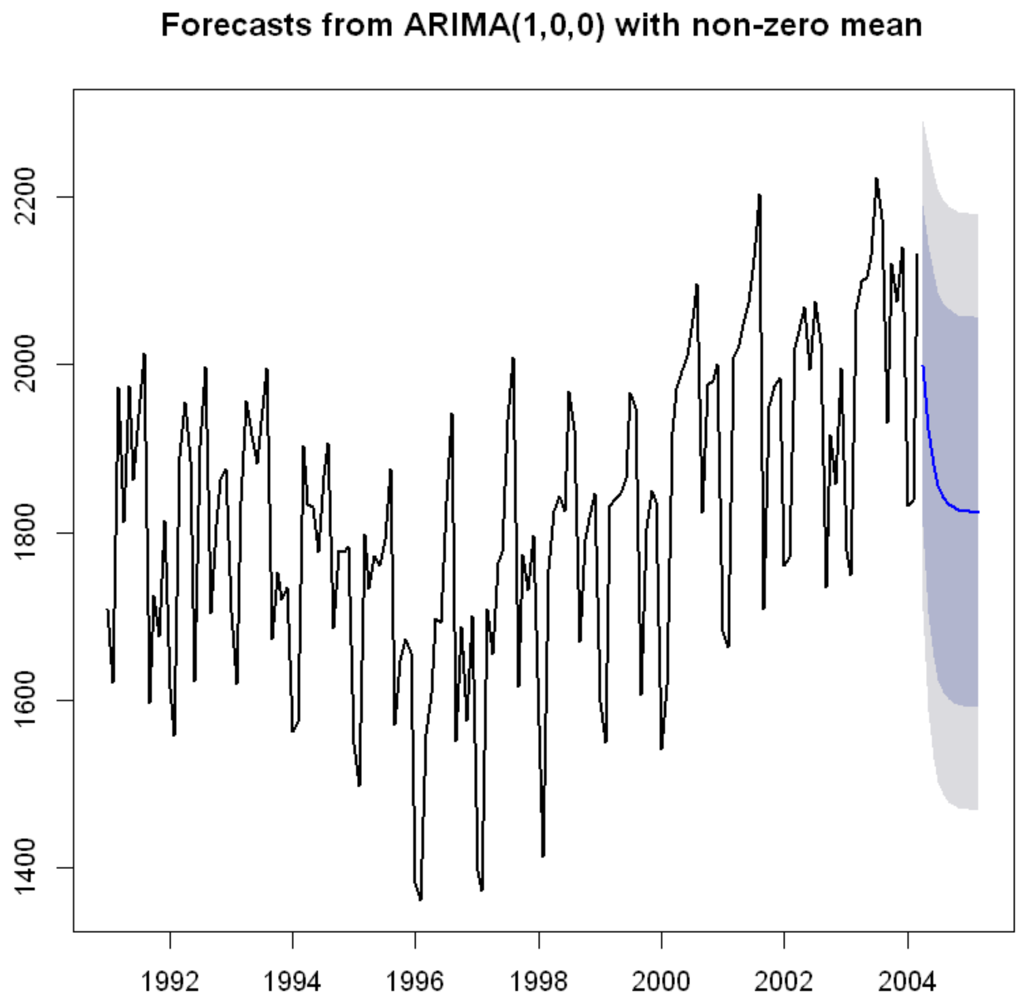
```
Call:
arima(x = Rider, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.5680  1823.8221
s.e.  0.0656    27.1868

sigma^2 estimated as 22285:  log likelihood = -1021.73,  aic =
2049.47
```

In [113]:

```
# forecast the values
forecasted_values = forecast(arima_100, 12)
forecasted_values
accuracy(forecasted_values)
plot(forecasted_values, lwd=2)
```

```
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Apr 2004       1998.880 1807.567 2190.193 1706.292 2291.468
May 2004       1923.263 1703.238 2143.287 1586.765 2259.760
Jun 2004       1880.308 1651.788 2108.828 1530.817 2229.800
Jul 2004       1855.909 1624.714 2087.104 1502.327 2209.491
Aug 2004       1842.049 1609.997 2074.100 1487.157 2196.940
Sep 2004       1834.176 1601.849 2066.503 1478.862 2189.489
Oct 2004       1829.703 1597.287 2062.119 1474.254 2185.153
Nov 2004       1827.163 1594.718 2059.607 1471.670 2182.656
Dec 2004       1825.720 1593.266 2058.174 1470.212 2181.227
Jan 2005       1824.900 1592.443 2057.357 1469.388 2180.412
Feb 2005       1824.434 1591.977 2056.892 1468.921 2179.948
Mar 2005       1824.170 1591.712 2056.628 1468.656 2179.684
```

	ME	RMSE	MAE	MPE	MAPE	MASE	A(
Training set	0.5395504	149.2823	120.06	-0.6812322	6.784501	1.472945	-0.02920


Forecasts from ARIMA(1,0,0) with non-zero mean
2200
2000
1800
1600
1400
1992
1994
1996
1998
2000
2002
2004

In [ ]:

```

```

MA(1)

In [114]:

```
# AR using ARIMA

arima_001 = arima(Rider, order=c(0,0,1))
arima_001
```

```
Call:
arima(x = Rider, order = c(0, 0, 1))

Coefficients:
         ma1  intercept
      0.5067  1823.2098
s.e.  0.0686    18.5777

sigma^2 estimated as 24273:  log likelihood = -1028.48,  aic = 
2062.96
```

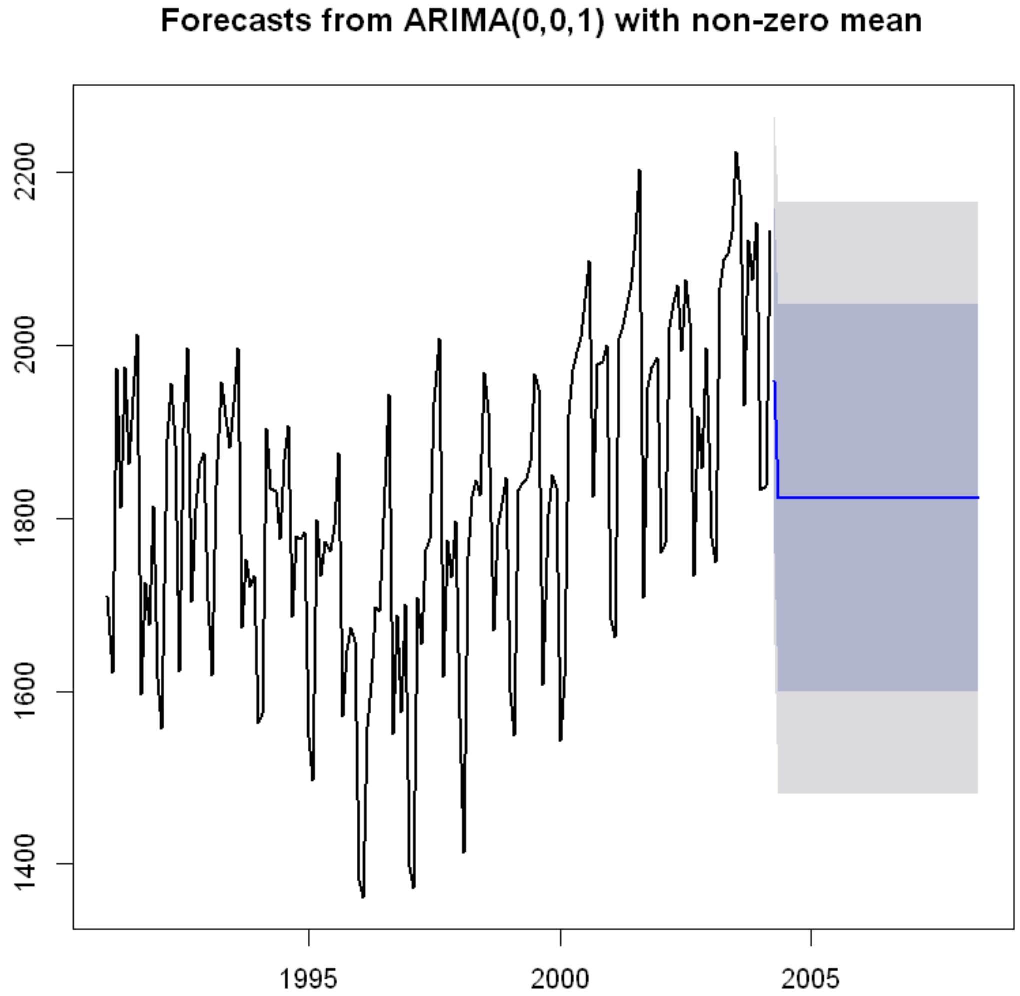
In [115]:

```
# forecast the values
forecasted_values = forecast(arima_001, 50)
forecasted_values
accuracy(forecasted_values)
plot(forecasted_values, lwd=2)
```

```
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Apr 2004       1959.529 1759.866 2159.192 1654.171 2264.887
May 2004       1823.210 1599.379 2047.041 1480.889 2165.530
Jun 2004       1823.210 1599.379 2047.041 1480.889 2165.530
Jul 2004       1823.210 1599.379 2047.041 1480.889 2165.530
Aug 2004       1823.210 1599.379 2047.041 1480.889 2165.530
Sep 2004       1823.210 1599.379 2047.041 1480.889 2165.530
Oct 2004       1823.210 1599.379 2047.041 1480.889 2165.530
Nov 2004       1823.210 1599.379 2047.041 1480.889 2165.530
Dec 2004       1823.210 1599.379 2047.041 1480.889 2165.530
Jan 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Feb 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Mar 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Apr 2005       1823.210 1599.379 2047.041 1480.889 2165.530
May 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Jun 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Jul 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Aug 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Sep 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Oct 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Nov 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Dec 2005       1823.210 1599.379 2047.041 1480.889 2165.530
Jan 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Feb 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Mar 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Apr 2006       1823.210 1599.379 2047.041 1480.889 2165.530
May 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Jun 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Jul 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Aug 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Sep 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Oct 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Nov 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Dec 2006       1823.210 1599.379 2047.041 1480.889 2165.530
Jan 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Feb 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Mar 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Apr 2007       1823.210 1599.379 2047.041 1480.889 2165.530
May 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Jun 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Jul 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Aug 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Sep 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Oct 2007       1823.210 1599.379 2047.041 1480.889 2165.530
```

```
Nov 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Dec 2007       1823.210 1599.379 2047.041 1480.889 2165.530
Jan 2008       1823.210 1599.379 2047.041 1480.889 2165.530
Feb 2008       1823.210 1599.379 2047.041 1480.889 2165.530
Mar 2008       1823.210 1599.379 2047.041 1480.889 2165.530
Apr 2008       1823.210 1599.379 2047.041 1480.889 2165.530
May 2008       1823.210 1599.379 2047.041 1480.889 2165.530
```

	ME	RMSE	MAE	MPE	MAPE	MASE	
Training set	-0.04981039	155.7977	128.715	-0.8087708	7.256321	1.579127	0.096



In [ ]:

Self Study: Additional Section

When AR or MA does not give desirable results

Try more sophisticated technique : ARIMA(p, d, q)

```
- ARIMA: autoregressive integrated moving average
- Key point: we need a stationary time series
- Differencing: method to remove trend. first order if we take differ
ence from the previous value.
- Apply ARIMA on a stationarised time series
- Use ACF and PACF curves to find p and q
- p : for AR (PACF chart)
- d : for differencing order
    subtract the trend from the series using differencing with the pr
evious values
- q : for MA (ACF chart)
```

Is it a stationary TS?

```
Let us test using Augmented Dickey-Fuller Test
what is my null hypothesis?
- it is a non-stionary time series
```

In [116]:

```
adfTest(Rider)
```

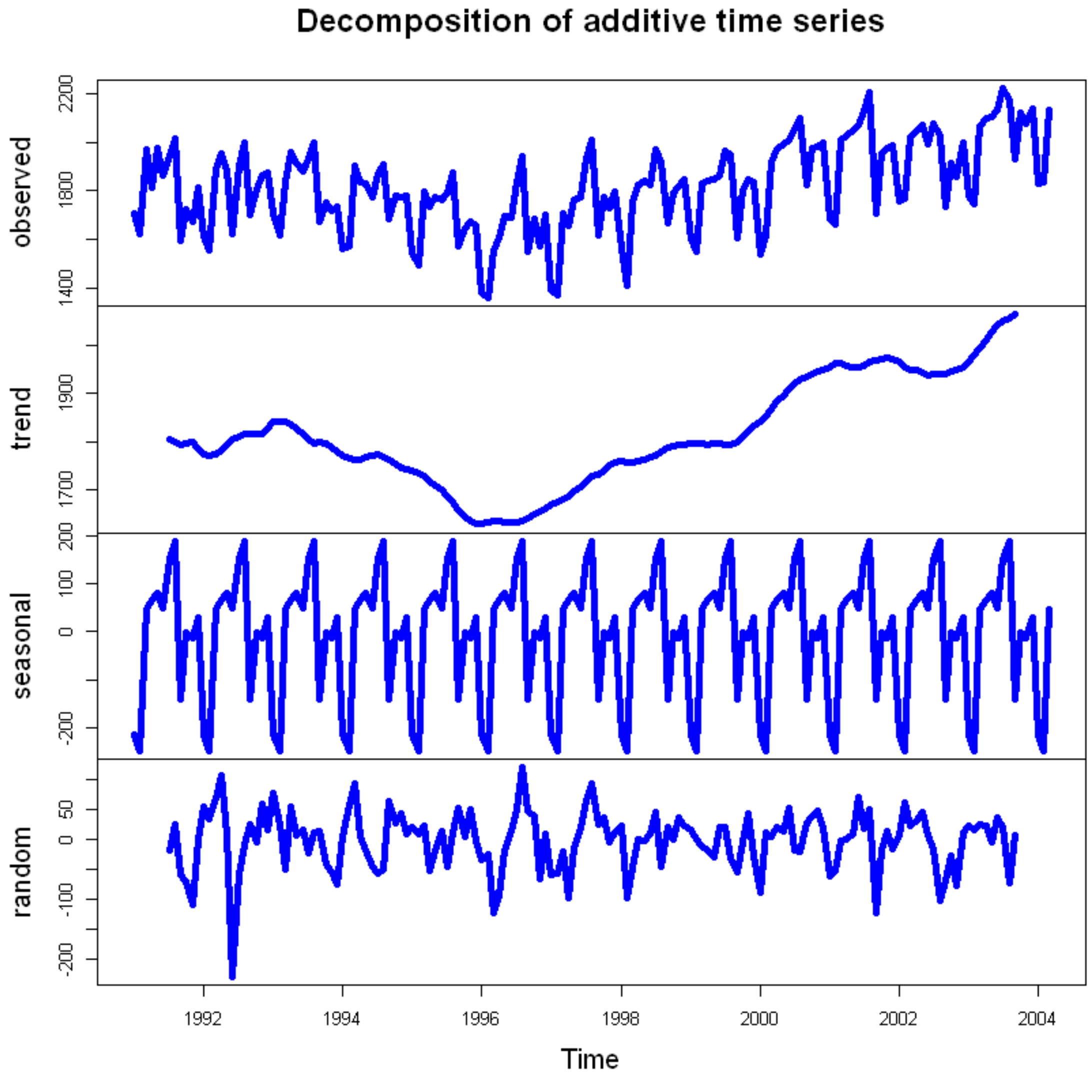
```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -0.1713
  P VALUE:
    0.5598

Description:
 Thu May 10 10:18:00 2018 by user: isspcs
```

Observations ?

```
- looking at the p-value we cannot reject the null hypothesis
```

In [117]:

```
# visualize the various components of TS
AmCom = decompose(Rider)
plot(AmCom, lwd=4, col="blue")
```

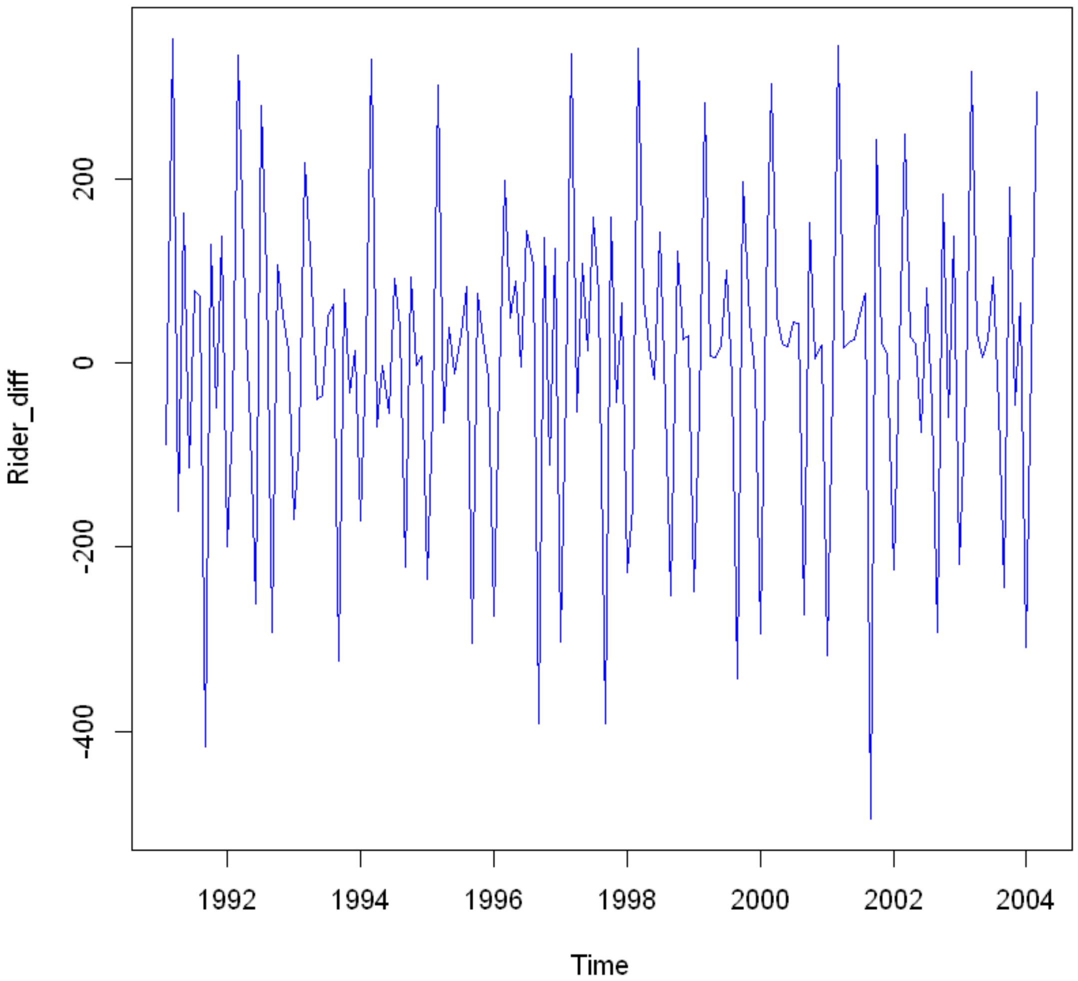


In [118]:

```
# as the TS appears non-stationary
# perform differencing of order 1
# to achieve stationarity
# using the "diff()" function in R
```

In [119]:

```
Rider_diff = diff(Rider, differences = 1)
plot.ts(Rider_diff,col="blue")
```



In [120]:

```
# test again the differenced TS
adfTest(Rider_diff)
```

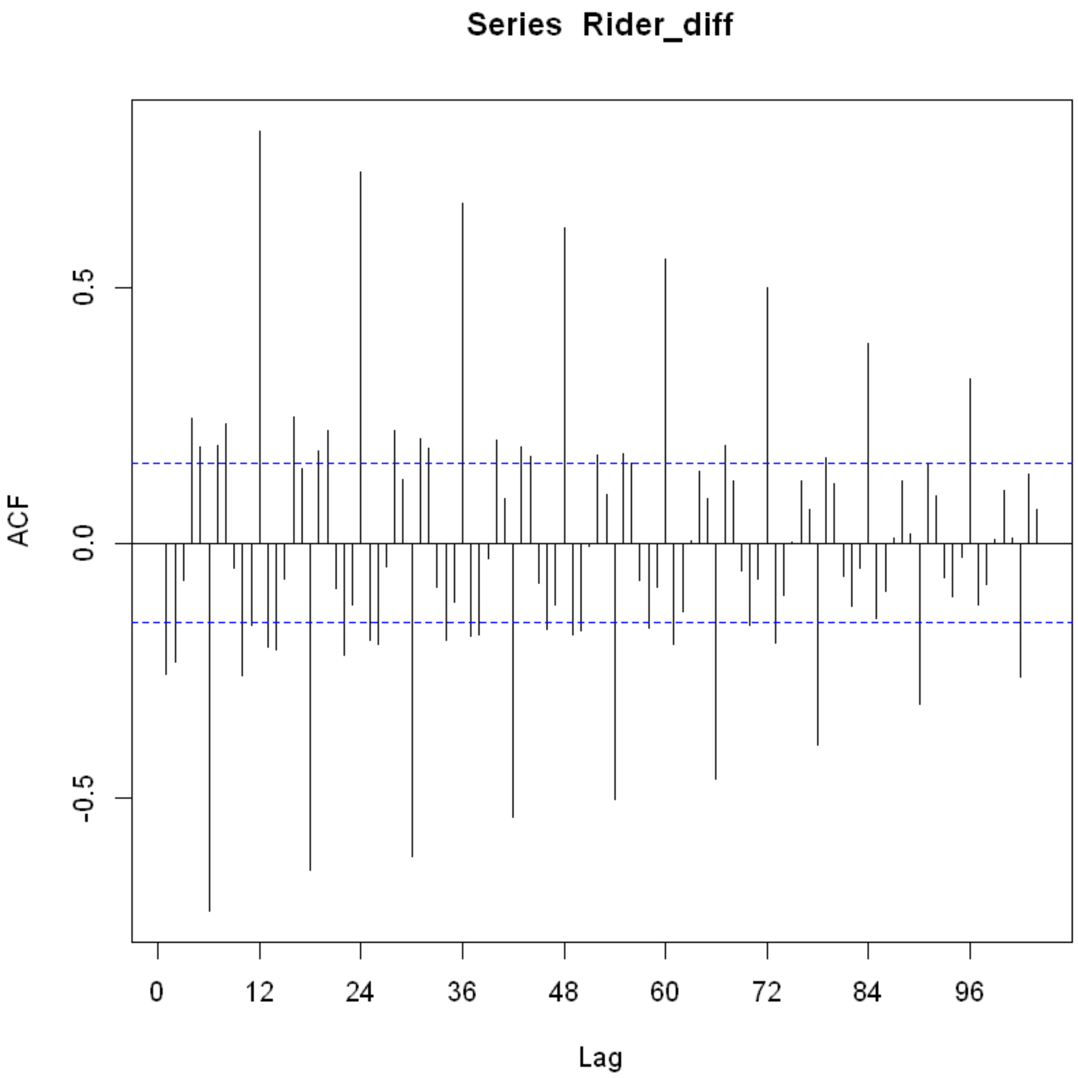
```
Warning message in adfTest(Rider_diff):
"p-value smaller than printed p-value"


Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -13.8351
  P VALUE:
    0.01

Description:
 Thu May 10 10:18:10 2018 by user: isspcs
```

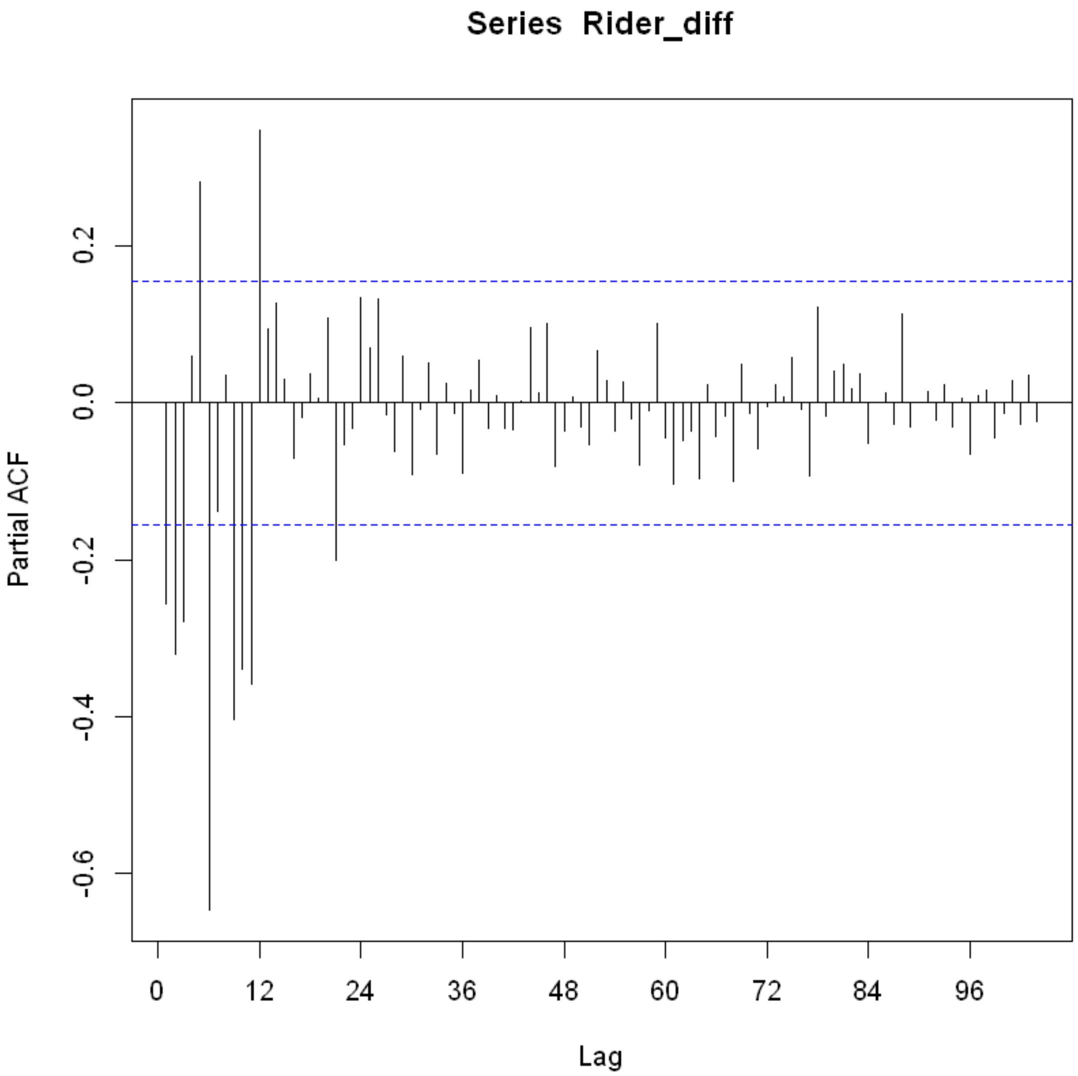
View auto correlation of time series data

ACF and PACF to discern p, q at d=1

In [121]:

```
Acf(Rider_diff, 104) # for p
Pacf(Rider_diff, 104)# for q
```




Series Rider_diff
Partial ACF
0.2
0.0
-0.2
-0.4
-0.6
0
12
24
36
48
60
72
84
96
Lag

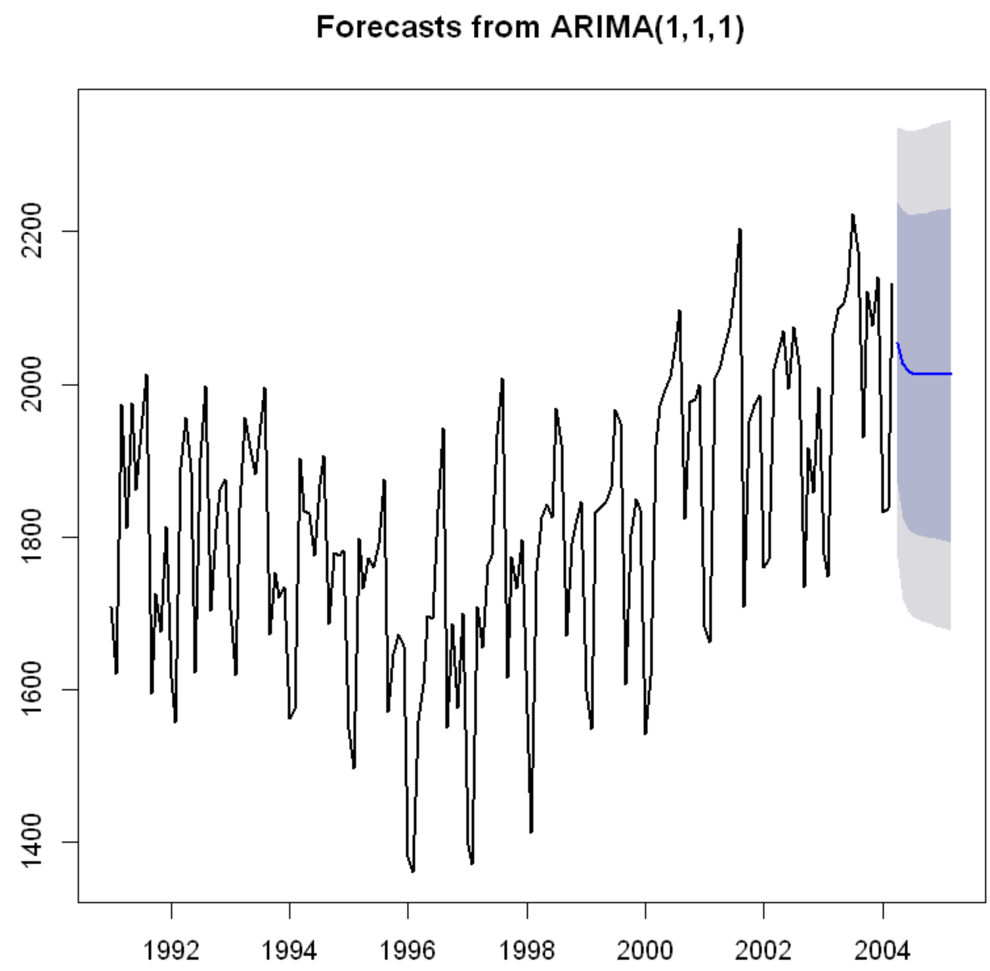
In [122]:

```
arima_111 = arima(Rider, order=c(1,1,1))
arima_111

# forecast the values
forecasted_values<-forecast(arima_111, 12)
forecasted_values
accuracy(forecasted_values)
plot(forecasted_values, lwd=2)
```

```
Call:
arima(x = Rider, order = c(1, 1, 1))

Coefficients:
         ar1      ma1
      0.3488  -0.9163
s.e.  0.0832   0.0294

sigma^2 estimated as 20714:  log likelihood = -1009.93,  aic =
2025.87
```

```
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Apr 2004       2054.032 1869.586 2238.478 1771.946 2336.118
May 2004       2026.836 1825.878 2227.795 1719.497 2334.176
Jun 2004       2017.350 1811.787 2222.914 1702.969 2331.732
Jul 2004       2014.042 1806.224 2221.860 1696.211 2331.872
Aug 2004       2012.887 1803.438 2222.336 1692.563 2333.212
Sep 2004       2012.485 1801.603 2223.366 1689.969 2335.000
Oct 2004       2012.344 1800.102 2224.587 1687.747 2336.942
Nov 2004       2012.295 1798.721 2225.869 1685.662 2338.929
Dec 2004       2012.278 1797.389 2227.168 1683.633 2340.924
Jan 2005       2012.272 1796.077 2228.468 1681.631 2342.914
Feb 2005       2012.270 1794.779 2229.762 1679.645 2344.895
Mar 2005       2012.270 1793.489 2231.050 1677.674 2346.865
```

	ME	RMSE	MAE	MPE	MAPE	MASE	
Training set	11.64368	143.4707	120.6574	0.003759092	6.787888	1.480274	0.024



Improve Further using AutoArima

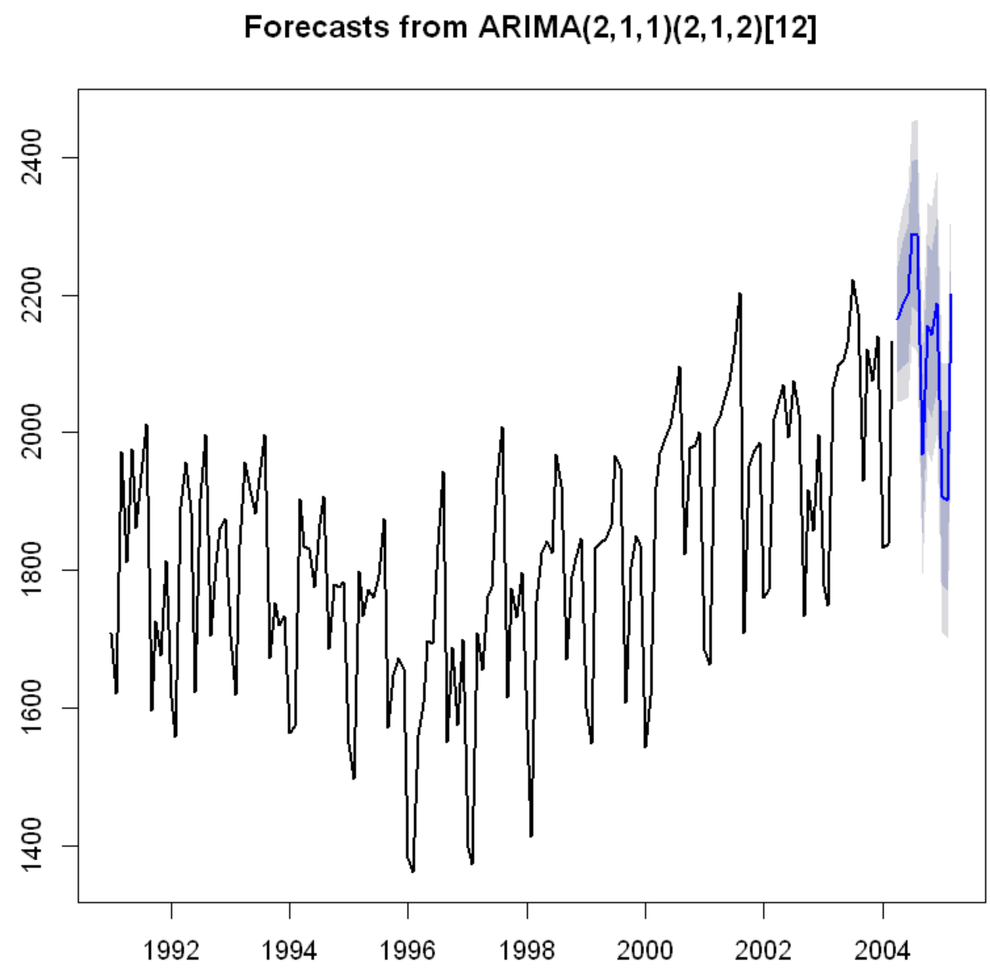
```
- R has an improved autoarima method
```

In [123]:

```
fitAutoArima = auto.arima(Rider)
fitAutoArima


#acf(residuals(fitAutoArima), 24, lwd=2)
Box.test(residuals(fitAutoArima), lag=12, type="Ljung")

f3<-forecast(fitAutoArima, 12)
f3
accuracy(f3)
plot(f3, lwd=2)
```

```
Series: Rider
ARIMA(2,1,1)(2,1,2)[12]

Coefficients:
         ar1     ar2      ma1     sar1     sar2     sma1    sma2
      0.4372  0.1093  -0.8426  0.7065  -0.1721  -1.4182  0.5751
s.e.  0.1463  0.1014   0.1006  0.3406   0.1708   0.3417  0.2574

sigma^2 estimated as 3710:  log likelihood=-808.88
AIC=1633.76   AICc=1634.81   BIC=1657.63


        Box-Ljung test

data:  residuals(fitAutoArima)
X-squared = 4.8051, df = 12, p-value = 0.9642

         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Apr 2004       2163.965 2085.894 2242.036 2044.566 2283.364
May 2004       2186.394 2095.567 2277.222 2047.485 2325.304
Jun 2004       2201.974 2102.276 2301.672 2049.499 2354.449
Jul 2004       2288.869 2183.096 2394.642 2127.103 2450.635
Aug 2004       2286.234 2175.662 2396.807 2117.128 2455.340
Sep 2004       1969.172 1854.540 2083.803 1793.858 2144.486
Oct 2004       2154.715 2036.465 2272.964 1973.868 2335.562
Nov 2004       2142.919 2021.334 2264.503 1956.972 2328.866
Dec 2004       2187.248 2062.520 2311.976 1996.493 2378.002
Jan 2005       1906.287 1778.554 2034.020 1710.937 2101.637
Feb 2005       1900.663 1770.033 2031.294 1700.882 2100.445
Mar 2005       2201.317 2067.874 2334.760 1997.233 2405.401
```

	ME	RMSE	MAE	MPE	MAPE	MASE	/
Training set	6.020336	56.95005	43.41794	0.2691171	2.425085	0.5326688	-0.015C



Type *Markdown* and LaTeX: α^2

Some more additional implementations

Holt Winter's Additive model

In [130]:

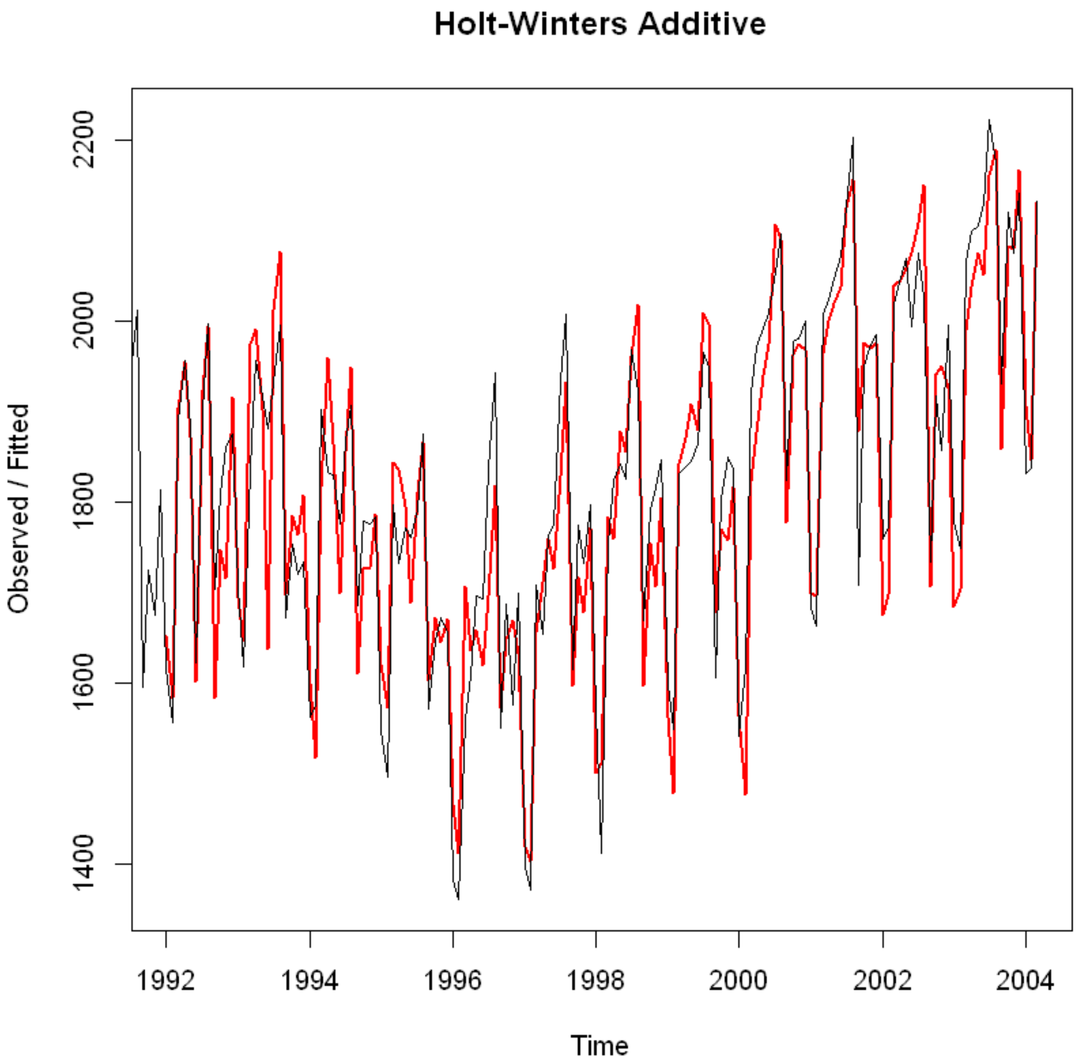
```
fitHW <- HoltWinters(Rider, seasonal="additive")

## Check the model summary and performance

fitHW
```

```
Holt-Winters exponential smoothing with trend and additive se
asonal component.

Call:
HoltWinters(x = Rider, seasonal = "additive")

Smoothing parameters:
 alpha: 0.3291363
 beta : 0.005983517
 gamma: 0.7483261

Coefficients:
            [,1]
a   2024.1666702
b      0.7482538
s1   122.4331329
s2   123.2120521
s3   113.9806811
s4   188.9536220
```

In [131]:

```
Box.test(residuals(fitHW), lag=12, type="Ljung")

## >>>> Plot the fitted Time Series

plot(fitHW, main="Holt-Winters Additive",lwd=2)
```

```
        Box-Ljung test

data:  residuals(fitHW)
X-squared = 24.08, df = 12, p-value = 0.01984
```

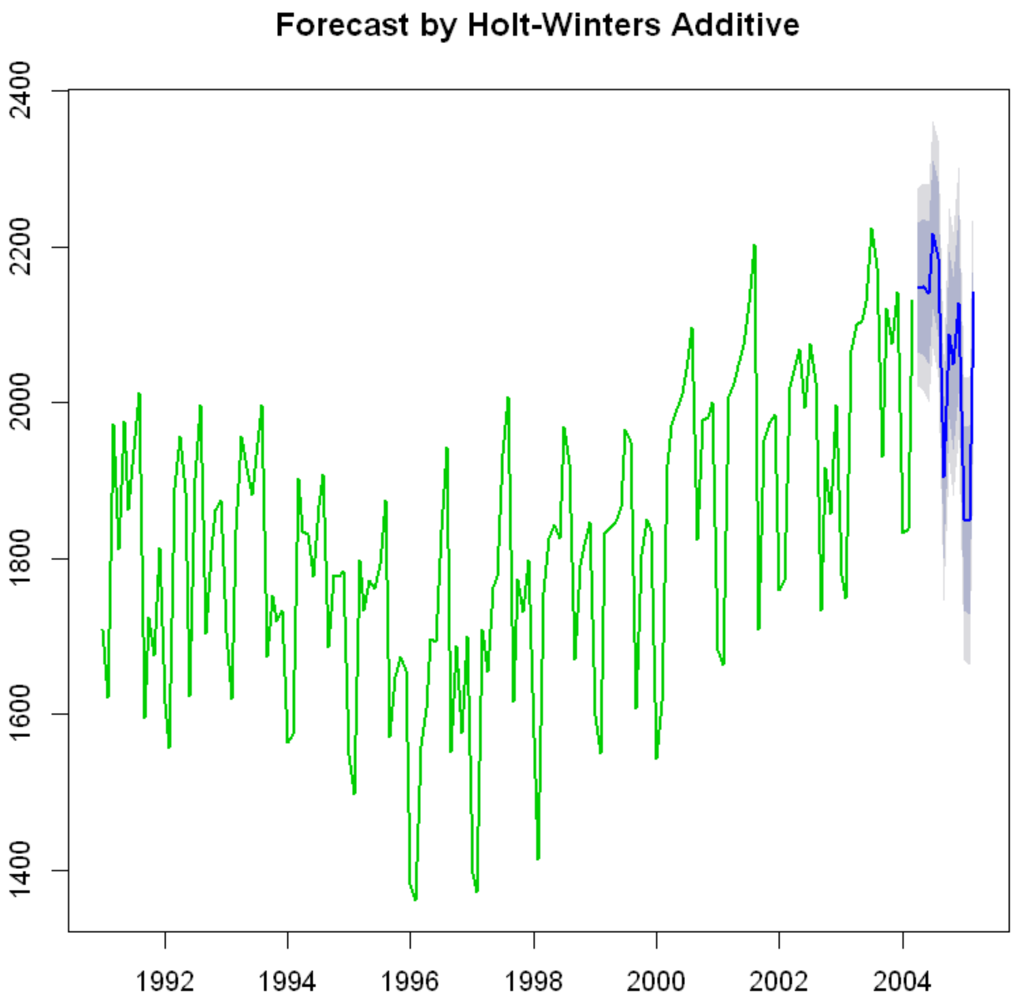


In [125]:

```
## Make forecast & check performance of model
f <-forecast(fitHW,12)
f
accuracy(f)
plot(f, lwd=2, col="green3", main="Forecast by Holt-Winters Additive")
```

```
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Apr 2004       2147.348 2064.724 2229.972 2020.985 2273.711
May 2004       2148.875 2061.840 2235.911 2015.766 2281.984
Jun 2004       2140.392 2049.109 2231.675 2000.787 2279.997
Jul 2004       2216.113 2120.725 2311.501 2070.230 2361.997
Aug 2004       2185.535 2086.166 2284.904 2033.563 2337.507
Sep 2004       1903.559 1800.318 2006.800 1745.666 2061.452
Oct 2004       2086.845 1979.829 2193.860 1923.179 2250.511
Nov 2004       2049.811 1939.108 2160.513 1880.506 2219.116
Dec 2004       2127.278 2012.967 2241.590 1952.454 2302.103
Jan 2005       1850.867 1733.018 1968.717 1670.632 2031.102
Feb 2005       1848.752 1727.430 1970.075 1663.205 2034.299
Mar 2005       2140.857 2016.121 2265.594 1950.089 2331.626
```

	ME	RMSE	MAE	MPE	MAPE	MASE	AC
Training set	6.746409	64.60549	51.18466	0.2987725	2.865657	0.6279541	0.2328

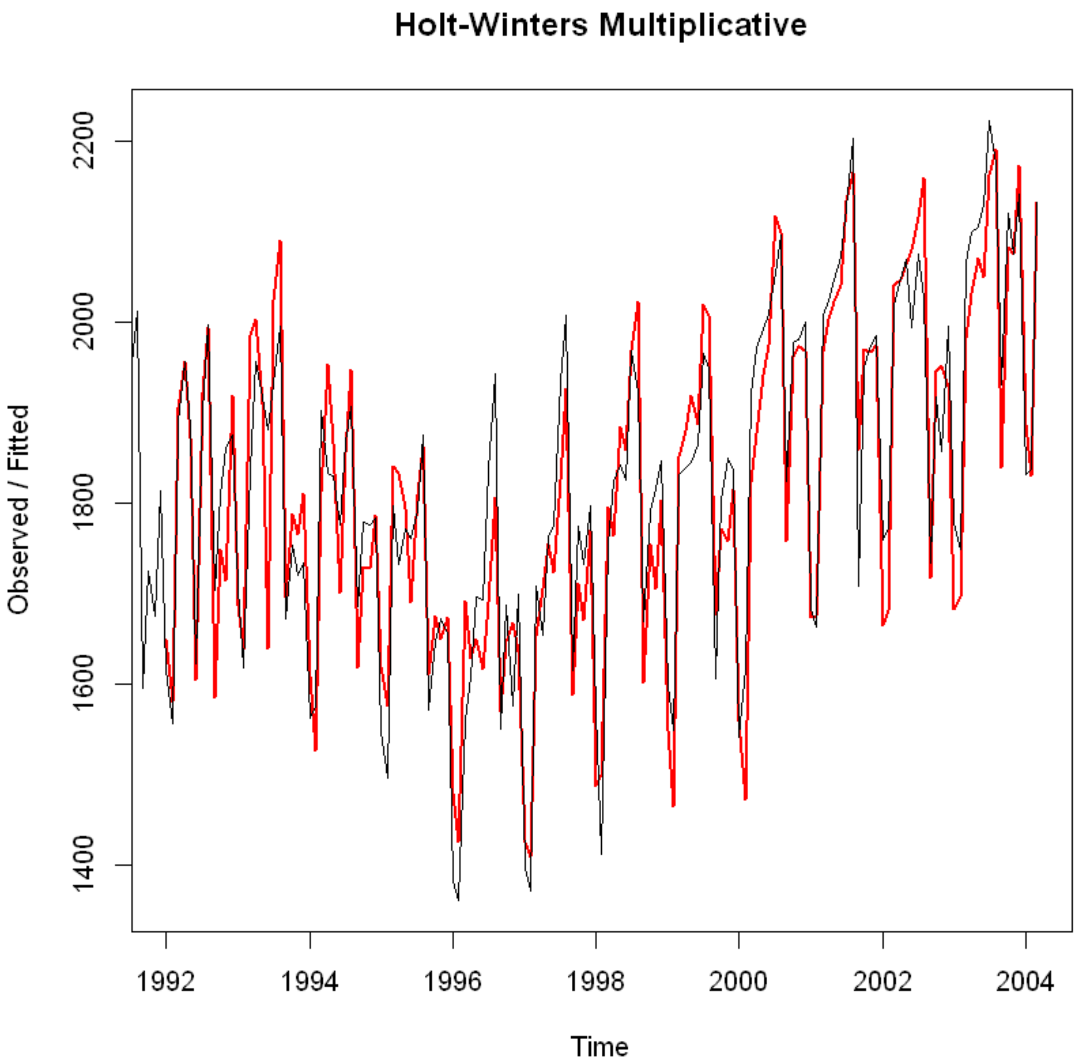


Fit a Holt Winter's Multiplicative model

In [126]:

```
fitHW1 <- HoltWinters(Rider, seasonal="multiplicative")
fitHW1

Box.test(residuals(fitHW1), lag=12, type="Ljung")
plot(fitHW1, main="Holt-Winters Multiplicative", lwd=2)
```

```
Holt-Winters exponential smoothing with trend and multiplicativ
e seasonal component.

Call:
HoltWinters(x = Rider, seasonal = "multiplicative")

Smoothing parameters:
 alpha: 0.3089791
 beta : 0.002171449
 gamma: 0.7283249

Coefficients:
            [,1]
a   2023.8929691
b     -0.3953454
s1     1.0618703
s2     1.0629445
s3     1.0585834
s4     1.0978202
s5     1.0833999
s6     0.9379491
s7     1.0305579
s8     1.0121733
s9     1.0517197
s10    0.9109065
s11    0.9071362
s12    1.0535303


        Box-Ljung test

data:  residuals(fitHW1)
X-squared = 29.849, df = 12, p-value = 0.002942
```


Holt-Winters Multiplicative
Observed / Fitted
2200
2000
1800
1600
1400
1992
1994
1996
1998
2000
2002
2004
Time

In [127]:

```
f1 <-forecast(fitHW1, 12)
f1
accuracy(f1)
plot(f1, lwd=2, col="green3", main="Forecast by Holt-Winters Multiplicative")
```

```
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Apr 2004       2148.692 2079.909 2217.474 2043.498 2253.886
May 2004       2150.445 2076.825 2224.066 2037.853 2263.038
Jun 2004       2141.204 2063.107 2219.301 2021.764 2260.644
Jul 2004       2220.135 2136.778 2303.491 2092.651 2347.618
Aug 2004       2190.544 2103.541 2277.546 2057.485 2323.603
Sep 2004       1896.084 1810.159 1982.008 1764.674 2027.493
Oct 2004       2082.887 1989.205 2176.569 1939.612 2226.161
Nov 2004       2045.329 1948.823 2141.835 1897.736 2192.922
Dec 2004       2124.826 2022.639 2227.013 1968.545 2281.107
Jan 2005       1839.976 1742.255 1937.697 1690.525 1989.427
Feb 2005       1832.002 1730.947 1933.056 1677.452 1986.551
Mar 2005       2127.234 1922.421 2332.048 1813.999 2440.470
```

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	7.096806	66.23209	52.73617	0.3245168	2.950346	0.6469885	0.2610309

In [ ]:

```
```

Fit an Exponential Smoothing Model

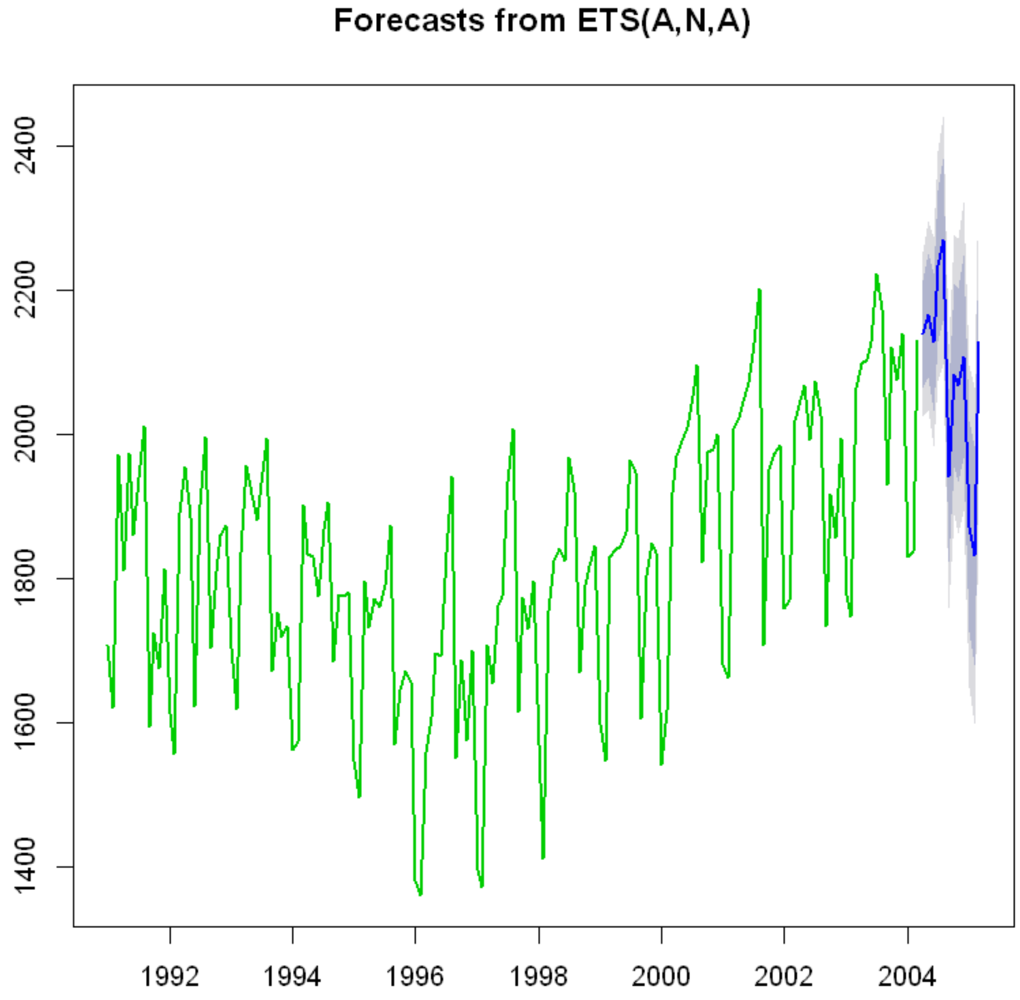
In [128]:

```
fitExp<-ets(Rider)#holt's
fitExp

Box.test(residuals(fitExp), lag=12, type="Ljung")

f2<-forecast(fitExp, 12)
f2
accuracy(f2)
plot(f2, lwd=2, col="green3")
```

```
ETS(A,N,A)

Call:
 ets(y = Rider)

  Smoothing parameters:
    alpha = 0.5625
    gamma = 1e-04

  Initial states:
    l = 1813.1983
    s=27.3613 -13.4603 1.85 -139.0846 189.3363 152.2884
           47.3404 85.0458 58.763 47.9488 -249.1373 -208.2517

  sigma:  58.216

     AIC     AICc      BIC
2113.703 2117.060 2159.737


        Box-Ljung test

data:  residuals(fitExp)
X-squared = 10.65, df = 12, p-value = 0.5591

         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Apr 2004       2140.417 2065.811 2215.024 2026.316 2254.518
May 2004       2166.705 2081.104 2252.306 2035.790 2297.620
Jun 2004       2129.000 2033.664 2224.335 1983.197 2274.802
Jul 2004       2233.944 2129.780 2338.108 2074.639 2393.249
Aug 2004       2270.993 2158.692 2383.294 2099.244 2442.742
Sep 2004       1942.575 1822.688 2062.461 1759.224 2125.925
Oct 2004       2083.511 1956.491 2210.531 1889.250 2277.771
Nov 2004       2068.201 1934.427 2201.975 1863.612 2272.790
Dec 2004       2109.026 1968.824 2249.228 1894.605 2323.447
Jan 2005       1873.405 1727.057 2019.754 1649.584 2097.227
Feb 2005       1832.528 1680.281 1984.776 1599.686 2065.371
Mar 2005       2129.613 1971.685 2287.541 1888.083 2371.143
```

	ME	RMSE	MAE	MPE	MAPE	MASE

	ME	RMSE	MAE	MPE	MAPE	MASE	
Training set	3.001498	55.59394	43.62935	0.08490419	2.436975	0.5352624	0.043



More Illustrations

Decomposition

In [6]:

```
## Read file "cola-data.csv"
cola_series = read.csv('Cola_data.csv')
head(cola_series, n=4)
```

Year	Month	Sales..Y.	Date
1	Jan	189	1/1/2000
NA	Feb	229	1/2/2000
NA	Mar	249	1/3/2000
NA	Apr	289	1/4/2000

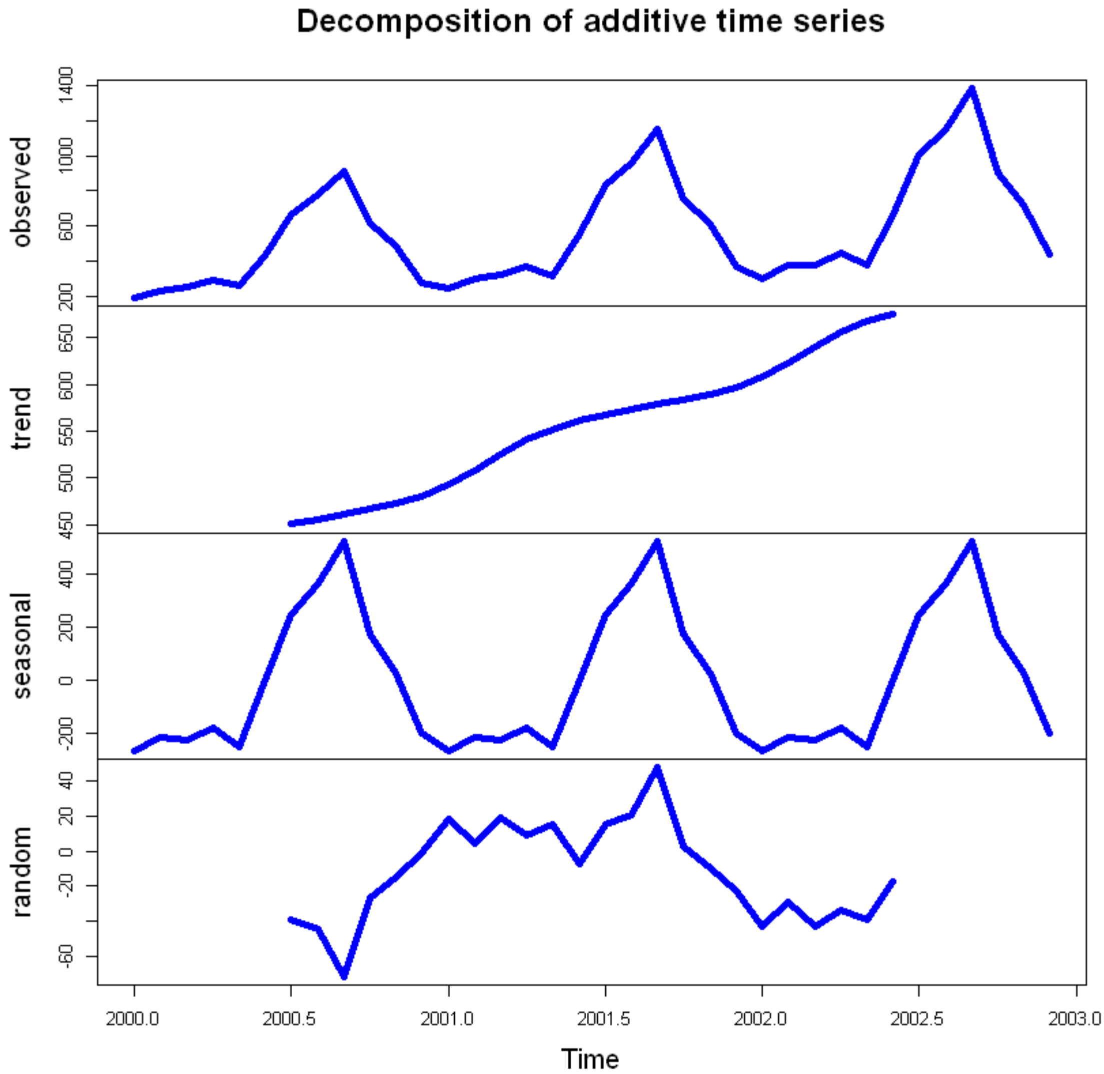
In [7]:

```
cola_ts = ts(cola_series$Sales..Y., frequency=12, start=c(2000, 1))
cola_ts
```

```
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  De
c
2000  189  229  249  289  260  431  660  777  915  613  485  27
7
2001  244  296  319  370  313  556  831  960 1152  759  607  37
1
2002  298  378  373  443  374  660 1004 1153 1388  904  715  44
1
```

In [8]:

```
# visualize the various components of TS
cola_com = decompose(cola_ts)
plot(cola_com, lwd=4, col="blue")
```



Arima

In [9]:

```
## Read file "Electricity raw data.csv"
elec_series = read.csv('Electricity raw data.csv')
head(elec_series, n=4)
```

Year	Month	Electricity
1	2005-01	448.1
1	2005-02	437.4
1	2005-03	480.0
1	2005-04	533.9

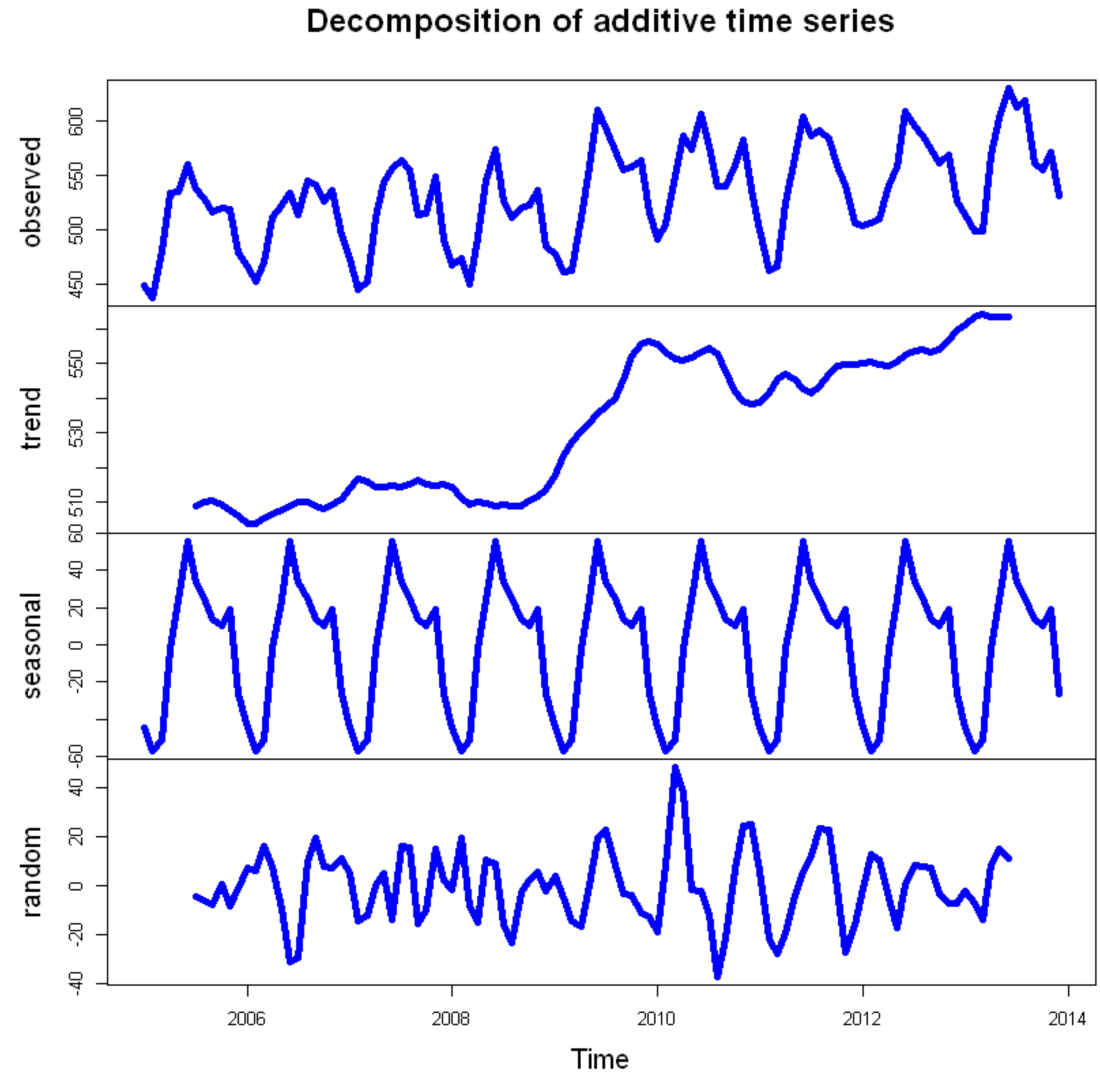
In [10]:

```
elec_ts = ts(elec_series$Electricity, frequency=12, start=c(2005, 1))
elec_ts
```

```
       Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oc
t   Nov   Dec
2005 448.1 437.4 480.0 533.9 535.3 560.6 537.7 529.1 516.7 520.
4 518.9 478.3
2006 465.9 452.7 470.4 511.8 522.4 533.8 513.8 544.9 542.0 526.
4 535.8 495.6
2007 474.2 445.0 452.4 511.7 543.5 556.8 563.9 555.8 514.4 515.
5 548.7 490.7
2008 467.9 473.6 450.2 492.3 544.8 573.5 526.7 510.9 520.0 523.
0 536.4 484.9
2009 477.2 461.3 462.0 511.4 557.5 610.5 593.6 573.1 555.8 558.
3 563.9 516.9
2010 491.5 504.8 548.5 586.0 574.5 607.0 576.2 540.6 540.1 558.
7 582.4 537.1
2011 500.2 462.6 466.6 525.3 566.0 603.8 587.0 591.6 583.6 558.
7 542.0 506.8
2012 504.2 505.9 509.5 541.8 558.1 608.9 595.8 586.5 574.2 561.
1 569.1 525.8
2013 514.0 498.9 499.1 569.2 602.7 630.6 612.9 618.6 561.6 555.
8 571.7 531.0
```

In [12]:

```
# visualize the various components of TS
elec_com = decompose(elec_ts)
plot(elec_com, lwd=4, col="blue")
```



In [13]:

```
arima_111 = arima(elec_ts, order=c(1,1,1))
arima_111

# forecast the values
forecasted_values<-forecast(arima_111, 12)
forecasted_values
accuracy(forecasted_values)
```

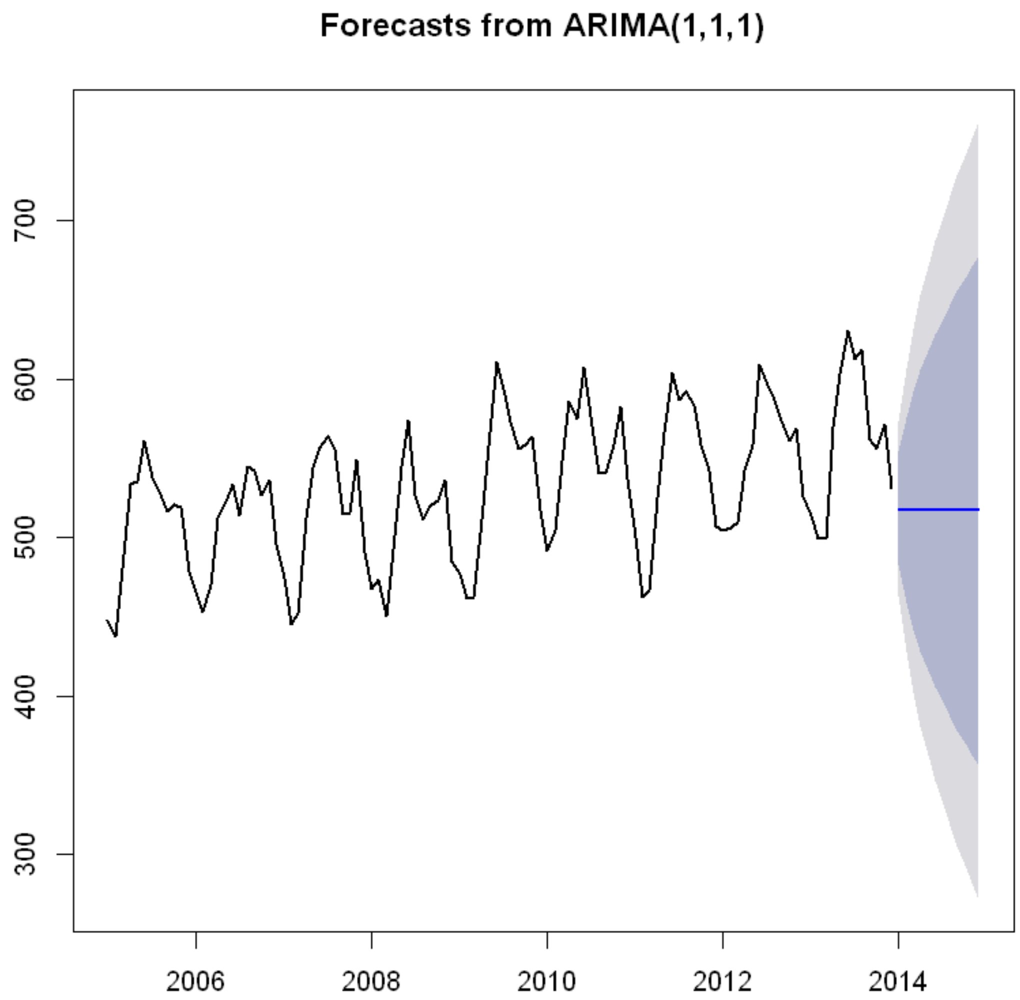
```
Call:
arima(x = elec_ts, order = c(1, 1, 1))

Coefficients:
         ar1     ma1
      0.1040  0.1880
s.e.  0.2943  0.2874

sigma^2 estimated as 778.4:  log likelihood = -508.03,  aic = 1
022.06
```

```
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2014       518.2974 482.5428 554.0521 463.6154 572.9794
Feb 2014       516.9761 458.5608 575.3914 427.6376 606.3147
Mar 2014       516.8387 441.6867 591.9907 401.9036 631.7737
Apr 2014       516.8244 427.9762 605.6725 380.9429 652.7058
May 2014       516.8229 416.1190 617.5267 362.8096 670.8362
Jun 2014       516.8227 405.5183 628.1271 346.5973 687.0481
Jul 2014       516.8227 395.8430 637.8024 331.8003 701.8451
Aug 2014       516.8227 386.8862 646.7592 318.1020 715.5434
Sep 2014       516.8227 378.5082 655.1372 305.2889 728.3565
Oct 2014       516.8227 370.6095 663.0359 293.2089 740.4365
Nov 2014       516.8227 363.1161 670.5293 281.7487 751.8967
Dec 2014       516.8227 355.9714 677.6740 270.8219 762.8235
```

	ME	RMSE	MAE	MPE	MAPE	MASE	
Training set	0.4864186	27.77006	22.0688	0.008373021	4.151499	1.024718	0.003

In [14]:

```
plot(forecasted_values, lwd=2)
```



Auto-Arima

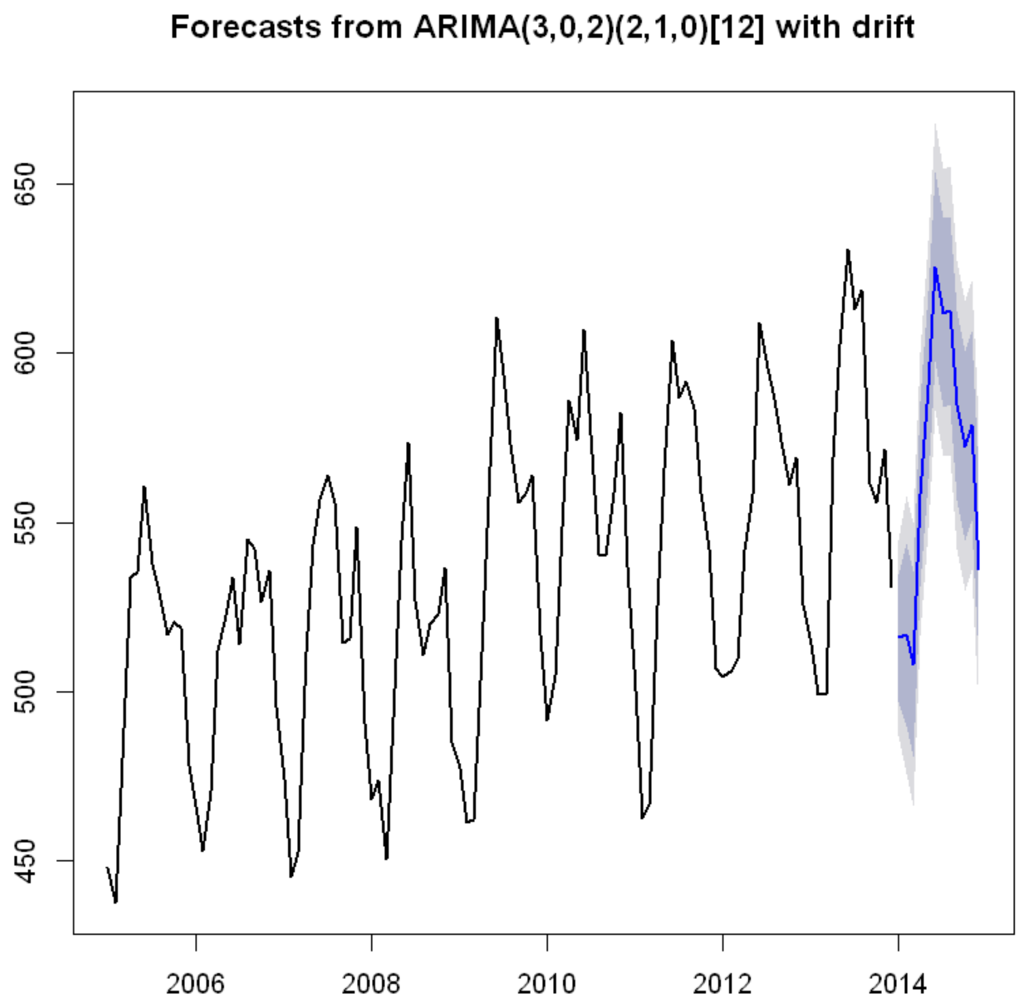
In [11]:

```
fitAutoArima = auto.arima(elec_ts)
fitAutoArima


#acf(residuals(fitAutoArima), 24, lwd=2)
Box.test(residuals(fitAutoArima), lag=12, type="Ljung")

f3<-forecast(fitAutoArima, 12)
f3
accuracy(f3)
plot(f3, lwd=2)
```

```
Series: elec_ts
ARIMA(3,0,2)(2,1,0)[12] with drift

Coefficients:
          ar1      ar2     ar3     ma1     ma2     sar1     sar
2   drift
      -0.3843  -0.2009  0.3264  1.4106  0.8780  -0.7045  -0.210
8  0.5925
s.e.   0.1205   0.1120  0.1321  0.0765  0.0733   0.1153   0.112
5  0.1737

sigma^2 estimated as 213.8:  log likelihood=-393.38
AIC=804.77   AICc=806.86   BIC=827.85


        Box-Ljung test

data:  residuals(fitAutoArima)
X-squared = 8.121, df = 12, p-value = 0.7756
```

```
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 2014       515.8311 497.0920 534.5702 487.1722 544.4900
Feb 2014       516.7737 489.9229 543.6246 475.7089 557.8386
Mar 2014       508.0988 480.7302 535.4675 466.2422 549.9555
Apr 2014       558.1334 530.7639 585.5028 516.2754 599.9914
May 2014       590.6335 562.7875 618.4794 548.0468 633.2202
Jun 2014       625.7235 597.8761 653.5709 583.1346 668.3124
Jul 2014       611.9949 584.1346 639.8553 569.3862 654.6037
Aug 2014       612.6727 584.7364 640.6089 569.9478 655.3975
Sep 2014       584.7389 556.7935 612.6842 542.0001 627.4776
Oct 2014       572.5568 544.6083 600.5053 529.8133 615.3003
Nov 2014       578.7264 550.7610 606.6919 535.9569 621.4959
Dec 2014       536.1626 508.1923 564.1330 493.3856 578.9397
```

	ME	RMSE	MAE	MPE	MAPE	MASE	
Training set	-0.09804067	13.19901	9.106145	-0.08277868	1.719975	0.4228246	0.


Forecasts from ARIMA(3,0,2)(2,1,0)[12] with drift
650
600
550
500
450
2006
2008
2010
2012
2014

In [ ]: