W271 HW8

Subhashini R., Lei Yang, Ron Cordell

March 29, 2016

Build an univariate linear time series model (i.e AR, MA, and ARMA models) using the series in hw08_series.csv.

- Use all the techniques that have been taught so far to build the model, including date examination, data visualization, etc.
- All the steps to support your final model need to be shown clearly.
- Show that the assumptions underlying the model are valid.
- Which model seems most reasonable in terms of satisfying the model's underling assumption?
- Evaluate the model performance (both in- and out-of-sample)
- Pick your "best" models and conduct a 12-step ahead forecast. Discuss your results. Discuss the choice of your metrics to measure "best".

```r
# Load the libraries and tools
library(astsa)
library(zoo)
```

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

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

```r
library(forecast)
```

```
## Loading required package: timeDate

## This is forecast 6.2

##
## Attaching package: 'forecast'

## The following object is masked from 'package:astsa':
##
##     gas
```

```r
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

```r
# load the CSV file
df <- read.csv('hw08_series.csv')
str(df)
```

```
## 'data.frame':    372 obs. of  2 variables:
##  $ X: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ x: num  40.6 41.1 40.5 40.1 40.4 41.2 39.3 41.6 42.3 43.2 ...
```

The CSV file for the HW8 time series consists of two variables: an X variable that is the time interval and an x value corresponding to the time period. There is no information about the time interval or units of the values.

A time series object is created from the dataframe for further analysis.

```r
ts1 <- ts(df$x)
str(ts1)
```

```
##  Time-Series [1:372] from 1 to 372: 40.6 41.1 40.5 40.1 40.4 41.2 39.3 41.6 42.3 43.2 ...
```

```r
summary(ts1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   36.00   57.38   76.45   84.83  111.50  152.60
```

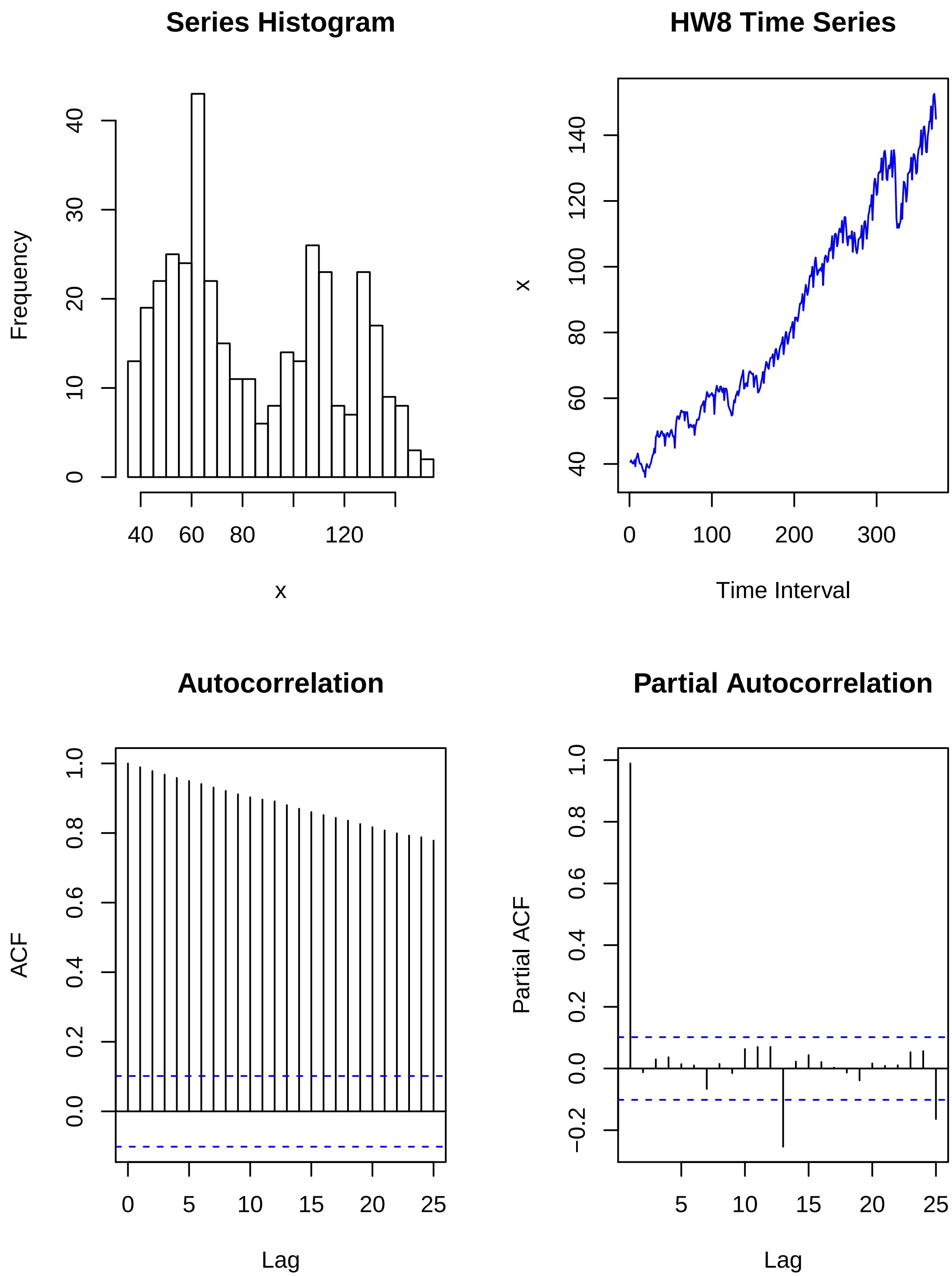
```r
head(ts1)
```

```
## [1] 40.6 41.1 40.5 40.1 40.4 41.2
```

```r
tail(ts1)
```

```
## [1] 141.9 146.9 152.0 152.6 149.7 145.0
```

```r
par(mfrow=c(2,2))
hist(ts1, main='Series Histogram', xlab='x', breaks=20)
plot.ts(ts1, col='blue',
        xlab='Time Interval',
        ylab='x',
        main='HW8 Time Series')
acf(ts1, main='Autocorrelation')
pacf(ts1, main='Partial Autocorrelation')
```



The time series plot reveals that the HW8 time series is a persistently upward trending series and is not stationary. The autocorrelation shows a very long decay over more than 25 lags while the partial autocorrelation shows statistically significant results at lags 13 and 25, indicating a strong seasonal component that happens every 12 periods, in addition to the inter-period seasonality.

The ACF plot is indicative of an ARMA underlying model because of the long taper - most likely an ARMA(2,1) model.

```r
ts1.fit <- Arima(ts1, order=c(2,0,1))
ts1.fit
```

```
## Series: ts1
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1  intercept
##       1.4723  -0.4724  -0.6401    139.437
## s.e.  0.1244   0.1242   0.1032    574.662
##
## sigma^2 estimated as 6.89:  log likelihood=-890.15
## AIC=1790.3   AICc=1790.46   BIC=1809.89
```

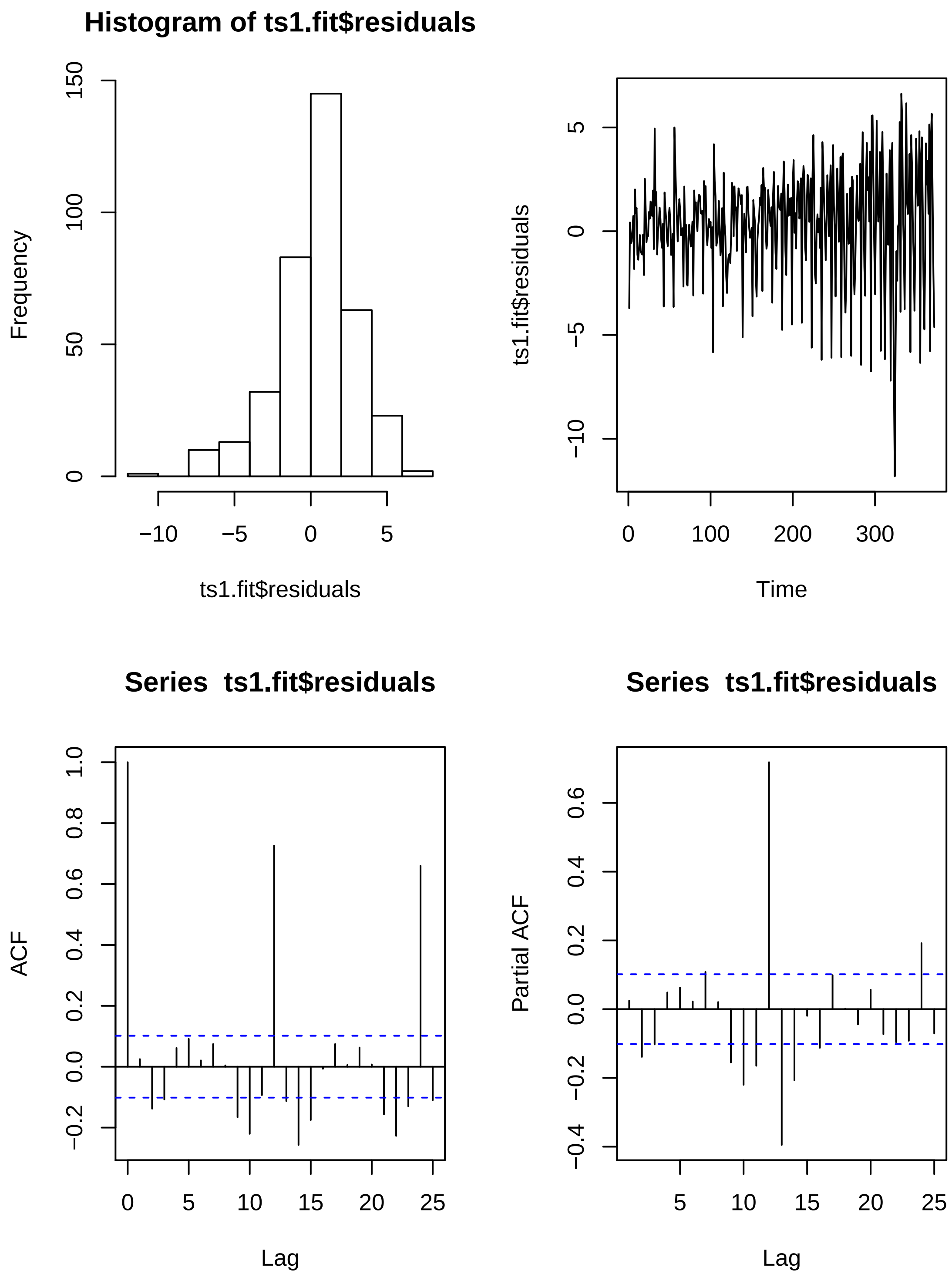
```r
summary(ts1.fit)
```

```
## Series: ts1
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1  intercept
##       1.4723  -0.4724  -0.6401    139.437
## s.e.  0.1244   0.1242   0.1032    574.662
##
## sigma^2 estimated as 6.89:  log likelihood=-890.15
## AIC=1790.3   AICc=1790.46   BIC=1809.89
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 0.3801334 2.624876 1.985887 0.3775082 2.362611 1.009543
##                    ACF1
## Training set 0.02475276
```

The coefficients of the ARMA(2,1) model are all statistically significant.

Checking the residuals…

```r
par(mfrow=c(2,2))
hist(ts1.fit$residuals)
plot.ts(ts1.fit$residuals)
acf(ts1.fit$residuals)
pacf(ts1.fit$residuals)
```



These residuals show a definite trend in that they become more volatile and larger over time. The distribution is slightly skewed. The autocorrelation shows correlations at lags 12 and 24 while the partial autocorrelation shows statistically significant effects at lag 2, 9-14 and 24. These indicate that the seasonality component remains and the series is not stationary.

Comparative statistics for the time series, fitted model and residuals shows that the model is reasonably close to the time series.

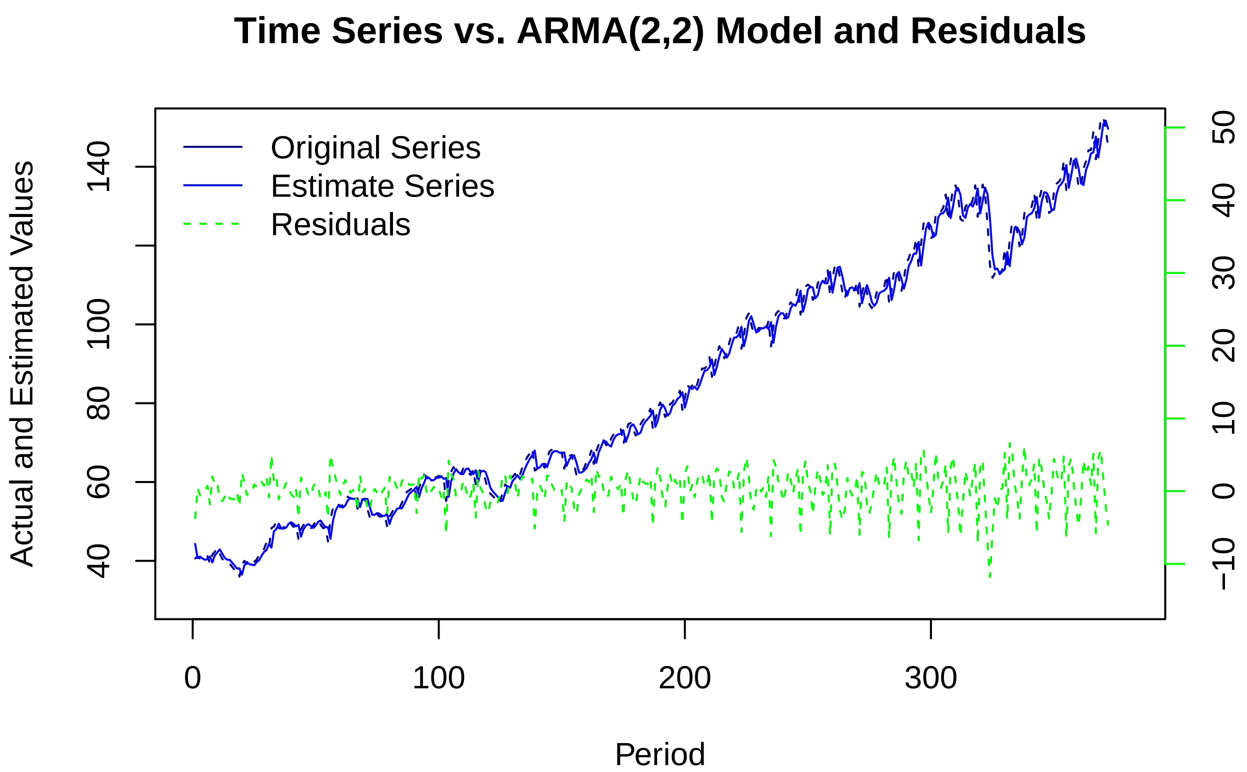
```r
fit.df <- data.frame(cbind(ts1, fitted(ts1.fit), ts1.fit$residuals))
class(df)
```

```
## [1] "data.frame"
```

```r
stargazer(fit.df, type="text",title="Descriptive Statistics", digits=1)
```

```
##
## Descriptive Statistics
## ===============================================
## Statistic          N  Mean St. Dev.  Min   Max
## -----------------------------------------------
## ts1               372 84.8   32.0   36.0  152.6
## fitted.ts1.fit.   372 84.4   31.8   36.5  151.7
## ts1.fit.residuals 372 0.4    2.6    -11.8  6.6
## -----------------------------------------------
```

```r
par(mfrow=c(1,1))
plot.ts(ts1, main="Time Series vs. ARMA(2,2) Model and Residuals",
        ylab="Actual and Estimated Values", xlab="Period",
        col="navy", ylim=c(30,150), xlim=c(0,380), lty=2)
par(new=T)
plot.ts(fitted(ts1.fit), xlab='', ylab='', axes=F, col='blue',
        ylim=c(30,150), xlim=c(0,380), lty=1)
leg.txt <- c("Original Series", "Estimate Series", "Residuals")
legend("topleft",legend=leg.txt,lty=c(1,1,2),
      col=c("navy","blue","green"),
      bty='n', cex=1)
par(new=T)
plot.ts(ts1.fit$residuals, axes=F,xlab='',ylab='',col="green",
        lty=2, pch=1, col.axis='green', ylim=c(-15,50),
        xlim=c(0,380))
axis(side=4,col='green')
mtext("Residuals",side=4,line=2, col='green')
```



Forecast Model

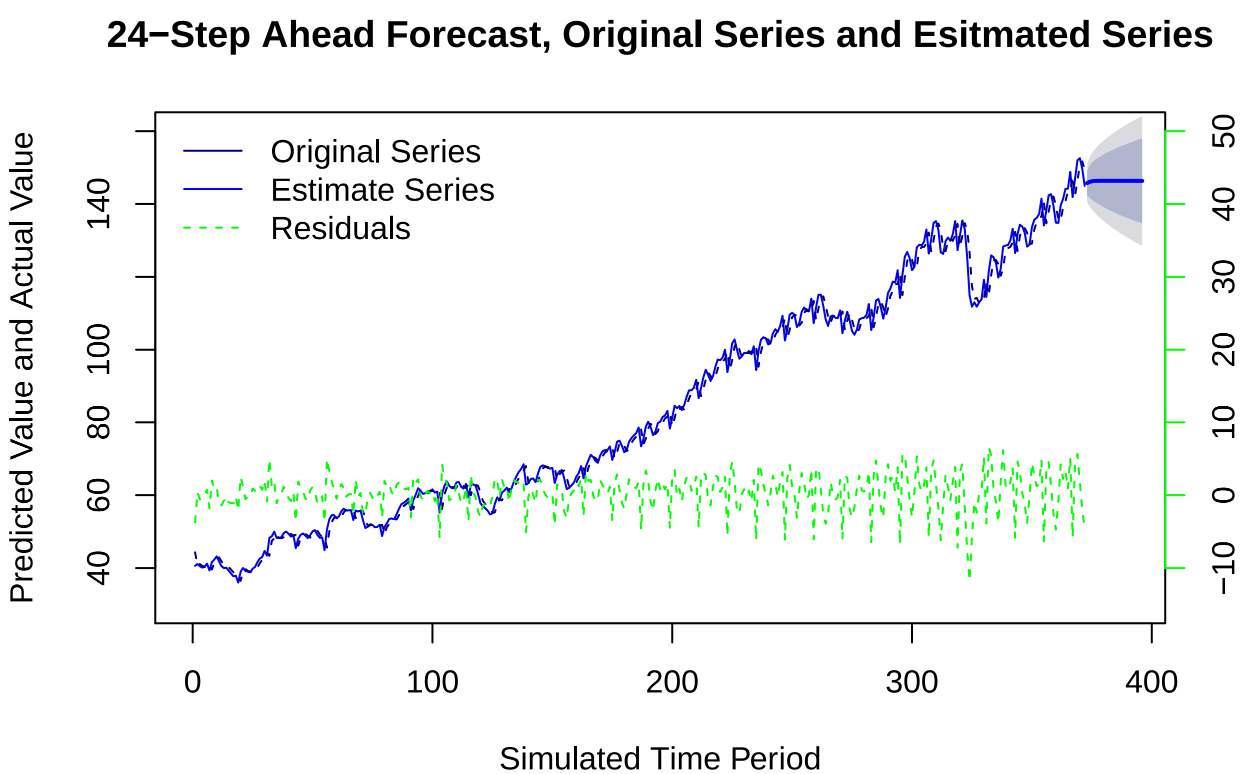
```
ts1.fcast <- forecast.Arima(ts1.fit, h=24)
ts1.fcast
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 373       145.7363 142.3724 149.1002 140.5917 150.8810
## 374       146.0831 141.7067 150.4595 139.3900 152.7763
## 375       146.2458 141.1895 151.3020 138.5129 153.9787
## 376       146.3215 140.7221 151.9208 137.7580 154.8850
## 377       146.3560 140.2853 152.4268 137.0716 155.6404
## 378       146.3712 139.8735 152.8689 136.4338 156.3085
## 379       146.3771 139.4835 153.2707 135.8343 156.9200
## 380       146.3787 139.1130 153.6444 135.2667 157.4907
## 381       146.3783 138.7597 153.9969 134.7266 158.0299
## 382       146.3769 138.4217 154.3321 134.2104 158.5433
## 383       146.3750 138.0972 154.6527 133.7153 159.0347
## 384       146.3729 137.7850 154.9608 133.2389 159.5069
## 385       146.3707 137.4837 155.2577 132.7792 159.9622
## 386       146.3685 137.1923 155.5446 132.3348 160.4022
## 387       146.3662 136.9099 155.8225 131.9041 160.8283
## 388       146.3640 136.6358 156.0921 131.4860 161.2419
## 389       146.3617 136.3692 156.3542 131.0795 161.6439
## 390       146.3594 136.1096 156.6092 130.6837 162.0351
## 391       146.3571 135.8565 156.8578 130.2977 162.4165
## 392       146.3549 135.6093 157.1004 129.9210 162.7887
## 393       146.3526 135.3678 157.3374 129.5528 163.1523
## 394       146.3503 135.1316 157.5691 129.1927 163.5079
## 395       146.3480 134.9002 157.7959 128.8401 163.8560
## 396       146.3458 134.6735 158.0180 128.4946 164.1969
```

```r
summary(ts1.fcast$mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   145.7   146.3   146.4   146.3   146.4   146.4
```

```r
par(mfrow=c(1,1))
plot(ts1.fcast,
     main='24-Step Ahead Forecast, Original Series and Esitmated Series',
     xlab='Simulated Time Period',
     ylab='Predicted Value and Actual Value',
     ylim=c(30,160), xlim=c(0,390), lty=1, col='blue')
par(new=T)
plot.ts(fitted(ts1.fit), axes=F,
        ylab='', xlab='',
        col="navy", ylim=c(30,160), xlim=c(0,390), lty=2)
legend("topleft",legend=leg.txt,lty=c(1,1,2),col=c("navy","blue","green"),
      bty='n', cex=1)
par(new=T)
plot.ts(ts1.fit$residuals, axes=F,xlab='',ylab='',col="green",
        lty=2, pch=1, col.axis='green', ylim=c(-15,50),xlim=c(0,390) )
axis(side=4,col='green')
mtext("Residuals",side=4,line=2, col='green')
```



"‘