Forecasting

Dr. B

Thursday, April 16, 2015

## Problem Overview

A public transportation company is expecting increasing demand for its services and is planning to acquire new buses and to extend its terminals. These investments require a reliable forecast of future demand which should be based on historic demand stored in the company's data warehouse.

For each 15-minute interval between 6:30 hours and 22 hours the number of passengers arriving at the terminal has been recorded and stored.

As a forecasting consultant you have been asked to forecast the number of passengers arriving at the terminal.

## [1] "LC\_COLLATE=English\_United States.1252;LC\_CTYPE=English\_United States.1252;LC\_MONETARY=English\_United States.1252;LC\_NUMERIC=C;LC\_TIME=English\_United States.1252"

### Data

The data were divided into a training set and a testing set. Demand data from March 1 to March 14 were used for the testing set.

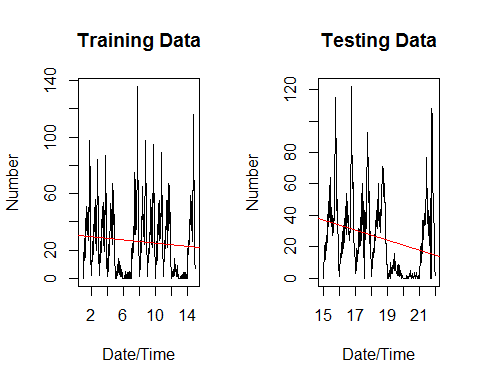
# Load the traing data and make it into time series  
train <- read.csv("D:/Data/bicup2006train.csv")  
#Convert to a timeseries using 63 periods (15 minutes) per day  
trainseries <- ts(train, frequency=63, start=c(1))  
summary(trainseries)

## DEMAND   
## Min. : 0.00   
## 1st Qu.: 4.00   
## Median : 23.00   
## Mean : 25.83   
## 3rd Qu.: 40.00   
## Max. :136.00

Demand data for March 15 to March 21 were used for the testing set.

# Load the test data and make it into time series  
test <- read.csv("D:/Data/bicup2006test.csv")  
#Convert to a timeseries using 63 periods (15 minutes) per day  
testseries <- ts(test, frequency=63, start=c(15))  
summary(testseries)

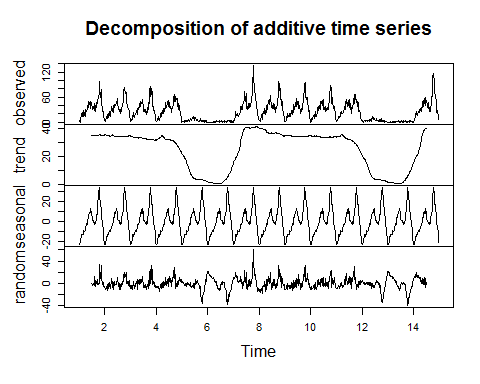
## DEMAND   
## Min. : 0.00   
## 1st Qu.: 4.00   
## Median : 22.00   
## Mean : 25.96   
## 3rd Qu.: 40.00   
## Max. :122.00

The data is shown below 

## Decomposing Seasonal Time Series

Decomposing a time series means separating it into its constituent components, which are usually a trend component and an irregular component, and if it is a seasonal time series, a seasonal component.

trainseriescomponents <- decompose(trainseries)  
plot(trainseriescomponents)



## Forecasting

#### Average method

The forecasts of all future values are equal to the mean of the historical data.

#Forcast 7 days  
meanfcast <- meanf(trainseries,441)

#### Naive Method

All forecasts are simply set to be the value of the last observation.

#Forcast 7 days  
naivefcast <- naive(trainseries,441)

#### Seasonal naive method

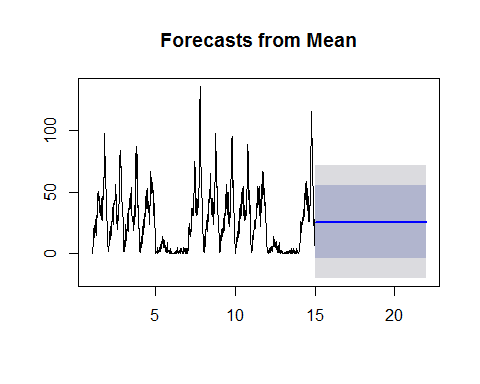
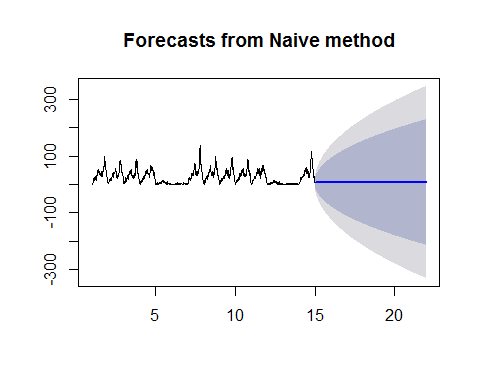
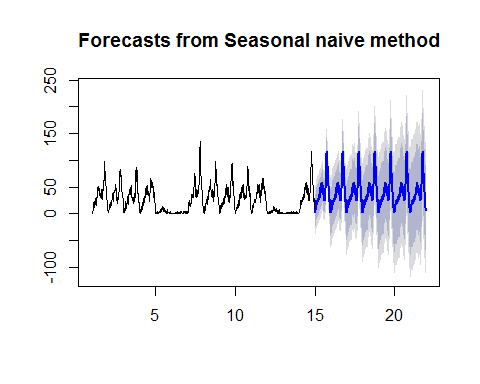
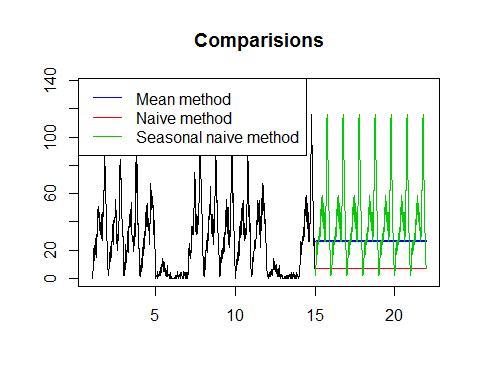
A similar method to the naive method. It is useful for highly seasonal data. In this case, we set each forecast to be equal to the last observed value from the same season of the year (e.g., the same month of the previous year).

#Forcast 7 days  
snaivefcast <- snaive(trainseries,441)

Of the methods used so far, the seasonal naive method seems to match the data the best.

#Forcast 7days  
fcastdata<-window(trainseries,start=1)  
fit1 <- meanf(fcastdata, h=441)  
fit2 <- naive(fcastdata, h=441)  
fit3 <- snaive(fcastdata, h=441)

#### Comparisions

#### Forecast Accuracy

The forecast error is simply actiual value less the projected value, which is on the same scale as the data. The two most commonly used scale-dependent measures are based on the Mean Absolute Error or the Root Mean Squared Error. Percentage errors have the advantage of being scale-independent. The most commonly used measure is Mean Absolute Percentage Error. Measures based on percentage errors have the disadvantage of being infinite or undefined if yi=0 for any i in the period of interest, and having extreme values when any yi is close to zero. Scaled errors were proposed as an alternative to using percentage errors. The Mean Absolute Scaled Error and the Mean Squared Scaled Error are two common scaled measures.

We compute the forecast accuracy measures for test period:

##### Average forecast method

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.0000000000000001821066 23.26008 18.98953 -Inf Inf 1.280140  
## Test set 0.1337868480725626363359 24.01361 19.43452 -Inf Inf 1.310138  
## ACF1 Theil's U  
## Training set 0.9364269 NA  
## Test set 0.9053621 NaN

##### Naive forecast method

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.006810443 8.232021 5.748014 -Inf Inf 0.3874906  
## Test set 18.961451247 30.596933 22.117914 -Inf Inf 1.4910339  
## ACF1 Theil's U  
## Training set -0.2051027 NA  
## Test set 0.9053621 NaN

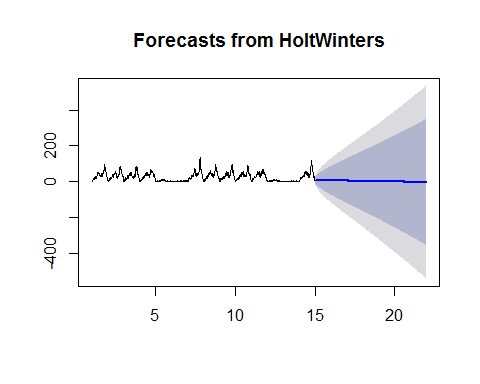
##### Seasonal naive method

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.3699634 23.62520 14.83394 -Inf Inf 1.000000 0.8998137  
## Test set -14.0226757 27.10515 17.30612 -Inf Inf 1.166657 0.8260613  
## Theil's U  
## Training set NA  
## Test set NaN

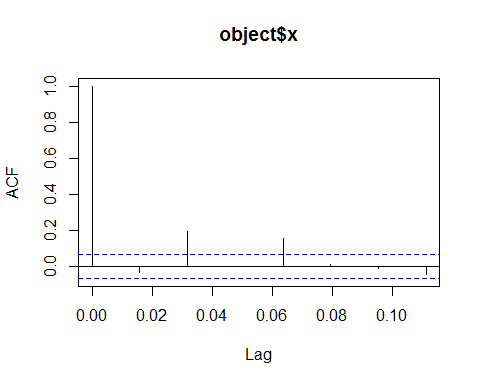
### Simple exponential smoothing

This method is suitable for forecasting data with no trend or seasonal pattern. In R, HoltWInters with gamma=FALSE computes the simple exponential smoothing forecast.

exptimeseries <- HoltWinters(trainseries, gamma=FALSE)  
expfcast <-forecast.HoltWinters(exptimeseries,441)



The 'forecast errors' are calculated as the observed values minus predicted values, for each time point. We can only calculate the forecast errors for the time period covered by our original time series. One measure of the accuracy of the predictive model is the sum-of-squared-errors (SSE) for the in-sample forecast errors.

If there are correlations between forecast errors for successive predictions, it is likely that the simple exponential smoothing forecasts could be improved upon by another forecasting technique. To determine if this is the case, we can obtain a correlogram of the in-sample forecast errors for lags 1-7. 

We can see from the sample correlogram that the several of the autocorrelation touch or exceed the significance bounds. To test whether there is significant evidence for non-zero correlations at lags 1-7, we run a Ljung-Box test.

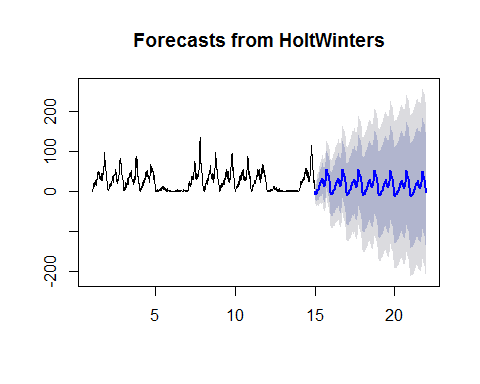
##   
## Box-Ljung test  
##   
## data: expfcast$residuals  
## X-squared = 57.726, df = 7, p-value = 0.0000000004288

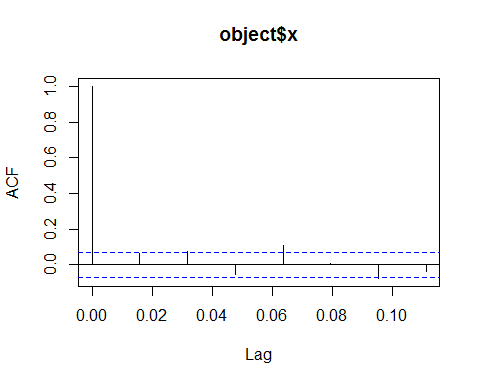
Here the p-value for the Ljung-Box test statistic is less than 0.05, so there is evidence of non-zero autocorrelations in the in-sample forecast errors at lags 1-7.

### Holt Winters' linear trend method

Holt (1957) extended simple exponential smoothing to allow forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend):

exptimeseries2 <- HoltWinters(trainseries)  
expfcast2 <-forecast.HoltWinters(exptimeseries2,441)



If there are correlations between forecast errors for successive predictions, it is likely that the simple exponential smoothing forecasts could be improved upon by another forecasting technique. To determine if this is the case, we can obtain a correlogram of the in-sample forecast errors for lags 1-7. 

We can see from the sample correlogram that the several of the autocorrelation touch or exceed the significance bounds. To test whether there is significant evidence for non-zero correlations at lags 1-7, we run a Ljung-Box test.

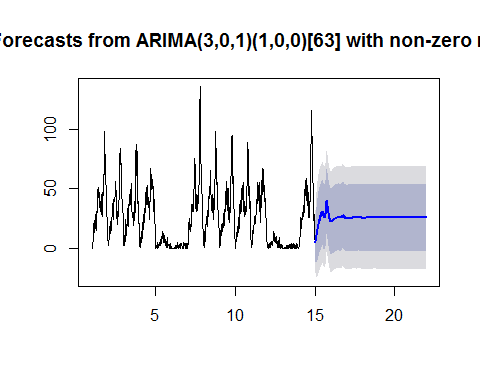
##   
## Box-Ljung test  
##   
## data: expfcast2$residuals  
## X-squared = 25.955, df = 7, p-value = 0.0005131

Here the p-value for the Ljung-Box test statistic is less than 0.05, so there is evidence of non-zero autocorrelations in the in-sample forecast errors at lags 1-7.

### ARIMA Models

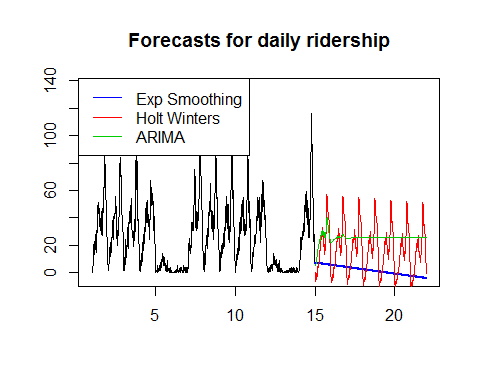
Exponential smoothing methods are useful for making forecasts, and make no assumptions about the correlations between successive values of the time series. However, if you want to make prediction intervals for forecasts made using exponential smoothing methods, the prediction intervals require that the forecast errors are uncorrelated and are normally distributed with mean zero and constant variance. While exponential smoothing methods do not make any assumptions about correlations between successive values of the time series, in some cases you can make a better predictive model by taking correlations in the data into account. Autoregressive Integrated Moving Average (ARIMA) models include an explicit statistical model for the irregular component of a time series, that allows for non-zero autocorrelations in the irregular component.

#fit <- Arima(trainseries, order=c(3,0,1), seasonal=c(0,1,2))  
#fit <- Arima(trainseries, order=c(0,1,3), seasonal=c(0,1,1))  
#res <- residuals(fit)  
#tsdisplay(res)  
#Box.test(res, lag=7, type="Ljung")  
arimatrainseries <- arima(trainseries, order=c(3,0,1), seasonal=c(1,0,0))  
arimafcast <-forecast.Arima(arimatrainseries,441)  
#res <residuals(arimafcast)



#### Comparison

fcastdata3<-window(trainseries,start=1)  
fit4 <- forecast.HoltWinters(exptimeseries,441)  
fit5 <- forecast.HoltWinters(exptimeseries2,441)  
fit6 <- forecast.Arima(arimatrainseries,441)



#### Forecast Accuracy

We compute the forecast accuracy measures for test period:

##### Exponential smoothing

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -0.3635496 8.132432 5.706266 NaN Inf 0.3841796 -0.03183686  
## Test set 24.4474549 33.798122 24.595156 NaN Inf 1.6580321 0.89962138  
## Theil's U  
## Training set NA  
## Test set NaN

##### Holt Winters'

## ME RMSE MAE MPE MAPE MASE  
## Training set -0.004451354 8.412059 6.127741 NaN Inf 0.4777325  
## Test set 8.718603039 21.691281 18.030915 NaN Inf 1.2155173  
## ACF1 Theil's U  
## Training set 0.06194977 NA  
## Test set 0.85476281 NaN

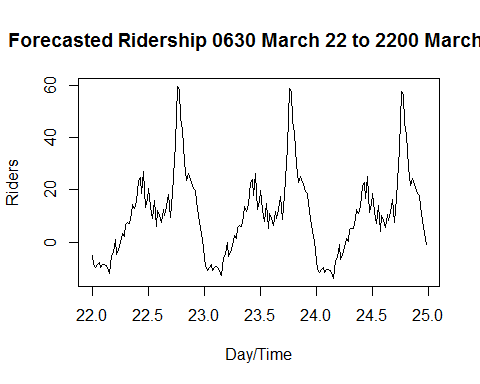
##### ARIMA

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.0336029 7.672115 5.492497 -Inf Inf 0.3702654 -0.0017719  
## Test set 0.7070695 23.090669 18.625949 -Inf Inf 1.2556303 0.9010195  
## Theil's U  
## Training set NA  
## Test set NaN

## Forecast for 22 - 24 March 2005

Using thee Holt Winter's method, the forecast for 22 -24 March would be:

complete<-rbind(train,test)  
timeseries <- ts(complete, frequency=63, start=c(1))  
exptimeseries3 <- HoltWinters(timeseries)  
expfcast3 <-forecast.HoltWinters(exptimeseries3,189)  
plot(expfcast3$mean, main="Forecasted Ridership 0630 March 22 to 2200 March 24", xlab="Day/Time", ylab="Riders")



## APPENDIX

### Residuals

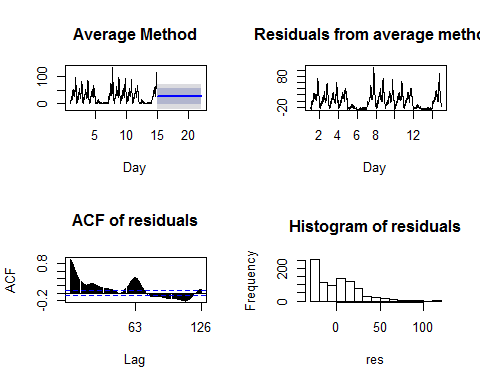
A residual in forecasting is the difference between an observed value and its forecast based on other observations. A good forecasting method will yield residuals with the following properties:

The residuals are uncorrelated. If there are correlations between residuals, then there is information left in the residuals which should be used in computing forecasts.  
  
 The residuals have zero mean. If the residuals have a mean other than zero, then the forecasts are biased.

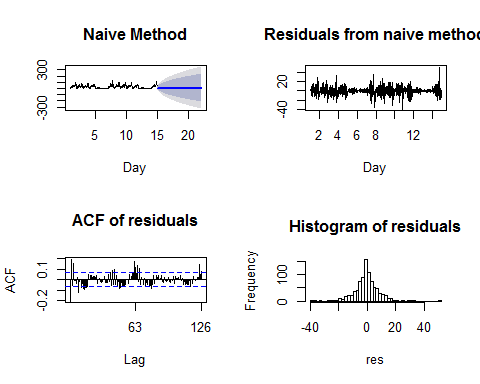
In addition to these essential properties, it is useful (but not necessary) for the residuals to also have the following two properties.

The residuals have constant variance.  
   
 The residuals are normally distributed.

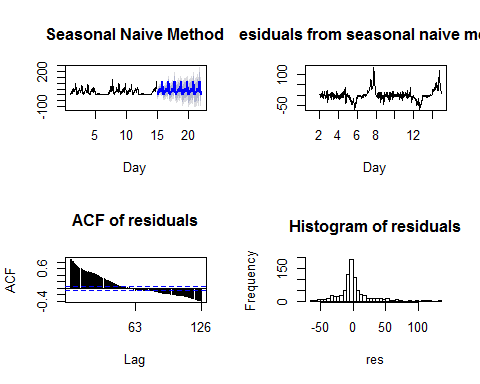
#### From Average Method



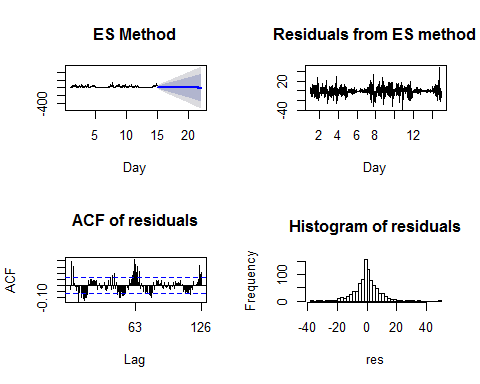
#### From Naive Method



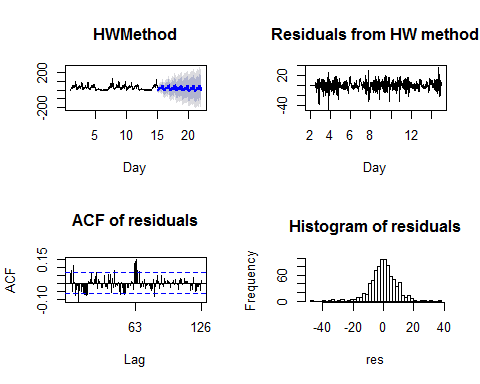
#### From Seasonal Naive Method



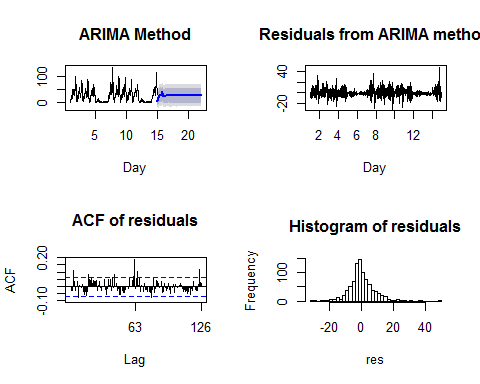
#### From Exponential Smoothing Method



#### From Holt Winters' Method



#### From ARIMA Method



This is an [R Markdown document](http://rmarkdown.rstudio.com). Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.