R For Time Series

Ying Xu
Assistant Professor
Engineering Systems and Design (ESD)
Singapore University of Technology and Design

Outline

- *Time series forecasting in R*
  - *Moving average*
  - *Holt-Winter methods*
    - *Exponential Smoothing*
    - *Double exponential Smoothing*
    - *Holt-Winter*

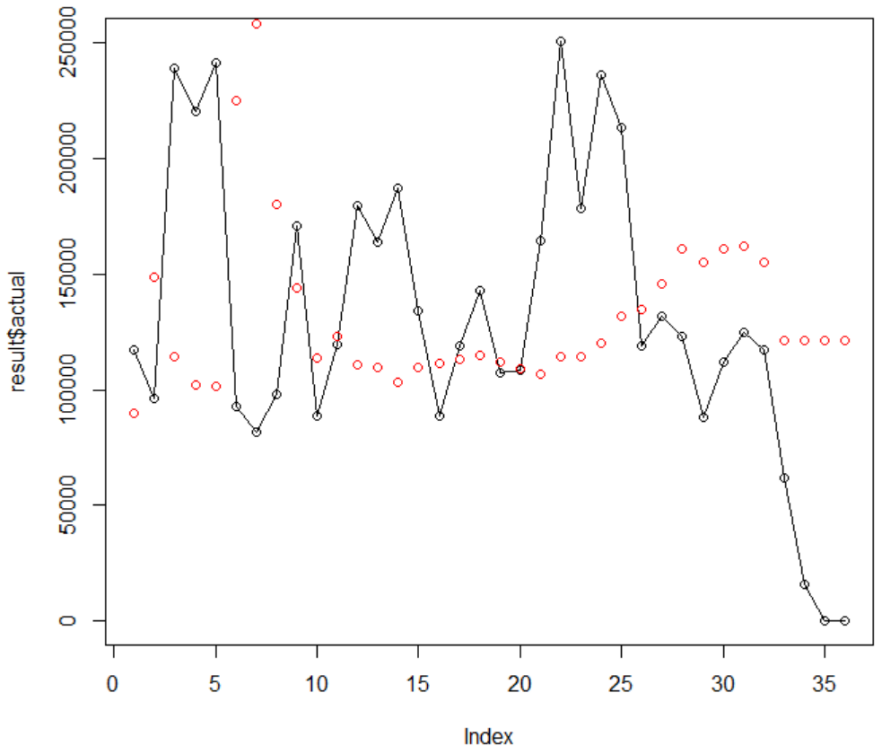
Read data

```
#### Prep: Load R package; Set working directory; Load the data to R
# Remove all variables from the R environment to create a fresh start
rm(list=ls())

# We next load some packages that are generally useful for data analytics.
#In each case, check first to see if a required package is installed. If not, then install it
# and invoke it as a library.
if(!require(stats)){
  install.packages("stats")
  library(stats)
}

# Similarly, we need to change the working directory to the directory where we want to read and save da
# Define a variable, wd, with a string describing the path to the directory where we want to read and s
# Change this to meet your needs.!!!!!!!!!
wd <- "C:\\Users\\xu_ying\\Google Drive\\Teaching\\DBA\\2022\\Lectures\\6\\Activity"
# Set the working directory to this path
setwd(wd)
# Test by displaying the current working directory
getwd()

# read the data from the data file
result <- read.csv(file = "Timeseries.csv")
head(result)
```

Compare the company's forecast with actual

```
# Note that the database has provided the company's forecast in the column "forecast"
# We first plot the actual demand against the company's forecast for comparison
forecasts <- result$forecast
  actuals <- result$actual
  plot(actuals)
  # Connect the points with lines so the sequence is clear
  lines(actuals)
  # Now add the corporate forecasts as red circles
  points(forecasts,col=c("red"))
  # How good are the forecasts? Perfection in the next plot would be a straight line angled at 45 degrees
  # You could also plot the actual against forecast by uncomment the command below
  # plot(actuals,forecasts)
  # So, the forecasts are not good.

# Let's develop a quantitative measure of forecast accuracy.
  errors <- forecasts-actuals
  # Take the absolute difference of forecasts versus actuals.
  abserrors <- abs(errors)
  # Compute the total of the absolute errors
  totalerror <- sum(abserrors)
  # Compute the total actual demand
  totalactual <- sum(actuals)
  # The ratio is the relative absolute error of forecast
  if (totalactual>0) {
      relativeabsoluteerror <- totalerror/totalactual
  } else {
    relativeabsoluteerror <- 0
  }
  # Practitioners multiply the relative absolute error this by 100 to get what they call the WAPE (Weighted Average Percent
  WAPE <- relativeabsoluteerror*100
  WAPE
  # So, the Weighted Average Percent Error is about 49%. That is pretty high.
  # And this is the largest selling product in the largest segment in the largest region!
  # The forecast errors for smaller selling products will likely be much worse.
```

Output



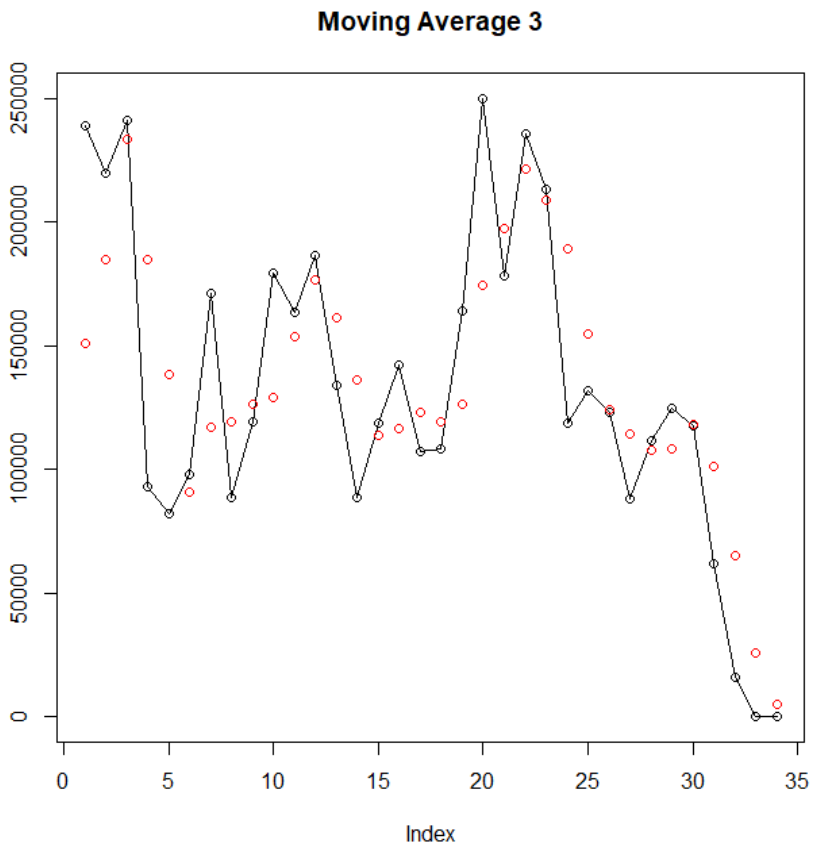

```
>    WAPE <- relativeabsoluteerror*100
>    WAPE
[1] 49.05694
```

Moving Average 3

```
# Next we try different forecasting methods in R
############################ Moving Average Forecasting Starts  ################################################
  # The moving average technique takes a vector of weights over a certain number of periods. Let's do a moving average over
  weights <- rep(1/3,3)
  weights
  # There are many ways to implement moving average in R. The filter method is easy.
  # The argument "Sides=1" says to just use data to the left of point to be forecast.
  maforecasts <- round(filter(actuals,weights,method="convolution",sides=1,circular=FALSE))
  # The first few fitted values are NA because there is insufficient history to calculate the moving average.
  # Omit NA entries.
  maforecasts <- na.omit(maforecasts)
  # The resulting structure is a time series. We just want the fitted values
  class(maforecasts)
  forecasts <- as.numeric(maforecasts)
  str(forecasts)
```

Moving Average 3: Output

```
# Plot the data
  # Omit the first few entries in actuals which has not predictions
  actualsnew <- tail(actuals, length(forecasts))
  plot(actualsnew)
  # Connect the points with lines so the sequence is clear
  lines(actualsnew)
  # Now add the corporate forecasts as red circles
  points(forecasts,col=c("red"))
  title("Moving Average 3")

# Calculate the quantitative measure of forecast accuracy.
  errors <- forecasts-actualsnew
  # Take the absolute difference of forecasts versus actuals.
  abserrors <- abs(errors)
  # Compute the total of the absolute errors
  totalerror <- sum(abserrors)
  # Compute the total actual demand
  totalactual <- sum(actualsnew)
  # The ratio is the relative absolute error of forecast
  if (totalactual>0) {
      relativeabsoluteerror <- totalerror/totalactual
  } else {
    relativeabsoluteerror <- 0
  }
  # Practitioners multiply the relative absolute error this by 100 t
  WAPE <- relativeabsoluteerror*100
  WAPE
```




```
>    WAPE <- relativeabsoluteerror*100
>    WAPE
[1] 22.01729
```

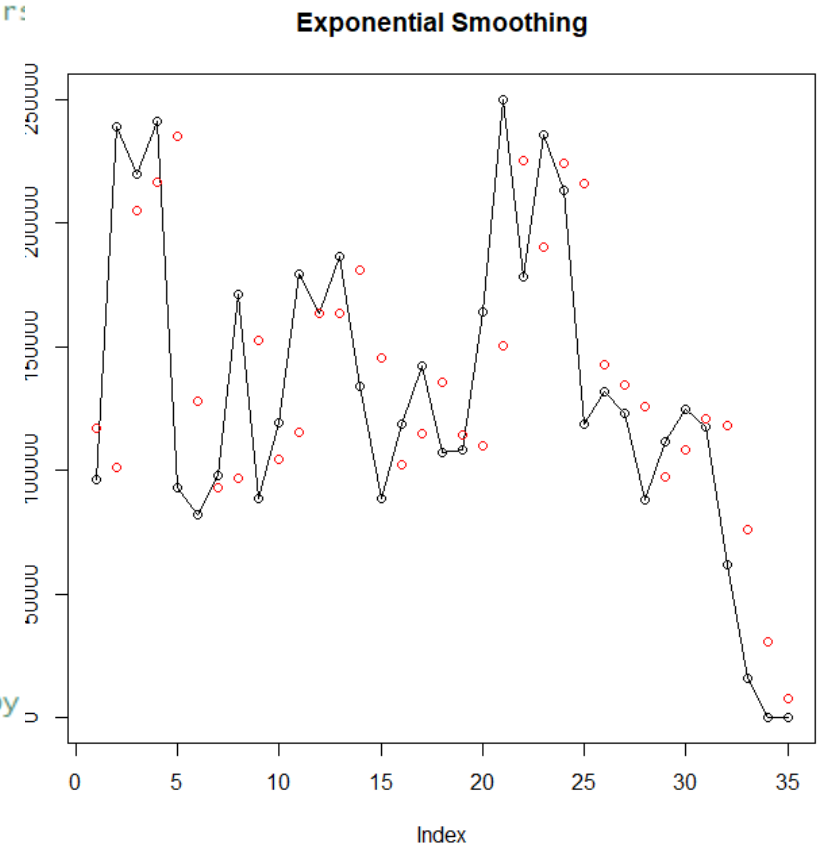
Exponential Smoothing

```
# We try another powerful technique Holt-Winters method.
############################  Holt-Winters method Starts ##################################################

################# Get read for Holt_Winters #################
  # Holt-Winters requires a time series structure. So we convert the actuals vector into a time series by specifying the freq
  tsactuals <- ts(actuals,frequency=12)
  str(tsactuals)
  plot(tsactuals)

# Here we try three Holt-Winters methods
# We first try Holt-Winters with just exponential smoothing (beta=False,gamma=FALSE)
################# Holt-Winters with just exponential smoothing (beta=False,gamma=FALSE) ################
  hw <- HoltWinters(tsactuals, beta=FALSE,gamma=FALSE)
  # THe Holt Winters data structure is complicated. We want to extract the first column of the fitted values.
  hwfitted <- fitted(hw)[,1]
  # The fitted values are a time series so we convert it to numeric (and round them to whole numbers for simplicity).
  forecasts <- round(as.numeric(fitted(hw)[,1]))
```

Exponential Smoothing: Output

```
# Plot the Holt-Winters forecasts
# The fitted values are missing the first entry, so we the firs
  actualsnew <- tail(actuals, length(forecasts))
  plot(actualsnew)
  # Connect the points with lines so the sequence is clear
  lines(actualsnew)
  # Now add the corporate forecasts as red circles
  points(forecasts,col=c("red"))
  title("Exponential Smoothing")

# Calculate the quantitative measure of forecast accuracy.
  errors <- forecasts-actualsnew
  # Take the absolute difference of forecasts versus actuals.
  abserrors <- abs(errors)
  # Compute the total of the absolute errors
  totalerror <- sum(abserrors)
  # Compute the total actual demand
  totalactual <- sum(actualsnew)
  # The ratio is the relative absolute error of forecast
  if (totalactual>0) {
      relativeabsoluteerror <- totalerror/totalactual
  } else {
    relativeabsoluteerror <- 0
  }
  # Practitioners multiply the relative absolute error this by
  WAPE <- relativeabsoluteerror*100
  WAPE
```

```
>    WAPE <- relativeabsoluteerror*100
>    WAPE
[1] 30.74229
```



Double Exponential Smoothing

```
# We secondly try Holt-Winters without seasonality (double exponential smoothing (gamma=FALSE)
################## Holt-Winters without seasonality (gamma=FALSE) #################
  # Now call the Holt-Winters function without seasonality (gamma=FALSE). This is equivalent to double exponential smoothing.
  hw <- HoltWinters(tsactuals, gamma=FALSE)
  # THe Holt Winters data structure is complicated. We want to extract the first column of the fitted values.
  hwfitted <- fitted(hw)[,1]
  # The fitted values are a time series so we convert it to numeric (and round them to whole numbers for simplicity).
  forecasts <- round(as.numeric(fitted(hw)[,1]))
```

Input your code for plotting and calculating WAPE (see answer below)

```
>    WAPE <- relativeabsoluteerror*100
>    WAPE
[1] 32.7282
```

Holt-Winters

Input your code for the forecasting, plotting and calculating WAPE
(see answer below)

```
>    WAPE <- relativeabsoluteerror*100
>    WAPE
[1] 31.79599
```

Which methods works the best?