PES University, Bangalore

Established under Karnataka Act No. 16 of 2013

UE20CS312 - Data Analytics - Worksheet 3a - Basic Forecasting Techniques
Designed by Vishruth Veerendranath, Dept. of CSE - vishruth@pesu.pes.edu

Time Series Data and Basic Forecasting Techniques

Time Series data is any data that is collected at regular time intervals, with changing observations at every time interval. Processing time series data effectively can help gain meaningful insights into how a quantity changes with time.

Forecasting a quantity into the future is an essential task, that predicts future values at any particular time. Forecasts can be made using various techniques like Exponential Smoothing to much more complex techniques such as Recurrent Neural Networks. Let's try to process time-series data and forecast values using basic techniques!

Prerequisites

- Revise the following concepts
  - Components of Time-Series Data
  - Single, Double and Triple Exponential Smoothing
  - Regression (Refer to worksheets and slides from Unit 2)
  - Croston's Forecasting
  - Time-Series Accuracy Metrics

Task

Let's imagine it is 2012 and you are in the market to buy an Orange Ultrabook Laptop for college. But this laptop is rare to find and expensive. You would want to put your Data Analytics skills to use, and predict the best time to buy your laptop, such that you can get it for the best price! You would also like to predict when the Orange Ultrabook would be in stock and when it would have high demand.

An electronics store collected sales data for their store weekly, from *February 2010* to *October 2012*, a period of 143 months. You have gotten your hands on this, and will use it to predict how the prices will change in the future.

The data for the following tasks can be downloaded from this Github Repository.

Data Dictionary

```
Date - Date on which data was collected (end of the week)
Sales - Weekly sales of the store (in $)
Holiday_Flag - Boolean Flag. 0 for normal week and 1 for holiday season
Temperature - Average temperature during the week
Fuel_Price - Average price during the week (in $/gallon)
CPI - Consumer Price Index
Unemployment - Average percentage of Unemployment in the city
Laptop_Demand - Number of Orange Ultrabook laptops sold during the week
```

Data Ingestion and Preprocessing

- Read the file into a `data.frame` object

```
df <- read.csv('sales.csv')
head(df)
```

```
##   X       Date   Sales Holiday_Flag Temperature Fuel_Price      CPI
## 1 0 05-02-2010 2135144            0       43.76      2.598 126.4421
## 2 1 12-02-2010 2188307            1       28.84      2.573 126.4963
## 3 2 19-02-2010 2049860            0       36.45      2.540 126.5263
## 4 3 26-02-2010 1925729            0       41.36      2.590 126.5523
## 5 4 05-03-2010 1971057            0       43.49      2.654 126.5783
## 6 5 12-03-2010 1894324            0       49.63      2.704 126.6043
##   Unemployment Laptop_Demand
## 1        8.623             0
## 2        8.623             0
## 3        8.623             0
## 4        8.623             0
## 5        8.623             1
## 6        8.623             0
```

- Pick out the `Sales` column in the `data.frame`. Most of our time-series analysis will be done on this column.

```
sales <- df$Sales
head(sales)
```

```
## [1] 2135144 2188307 2049860 1925729 1971057 1894324
```

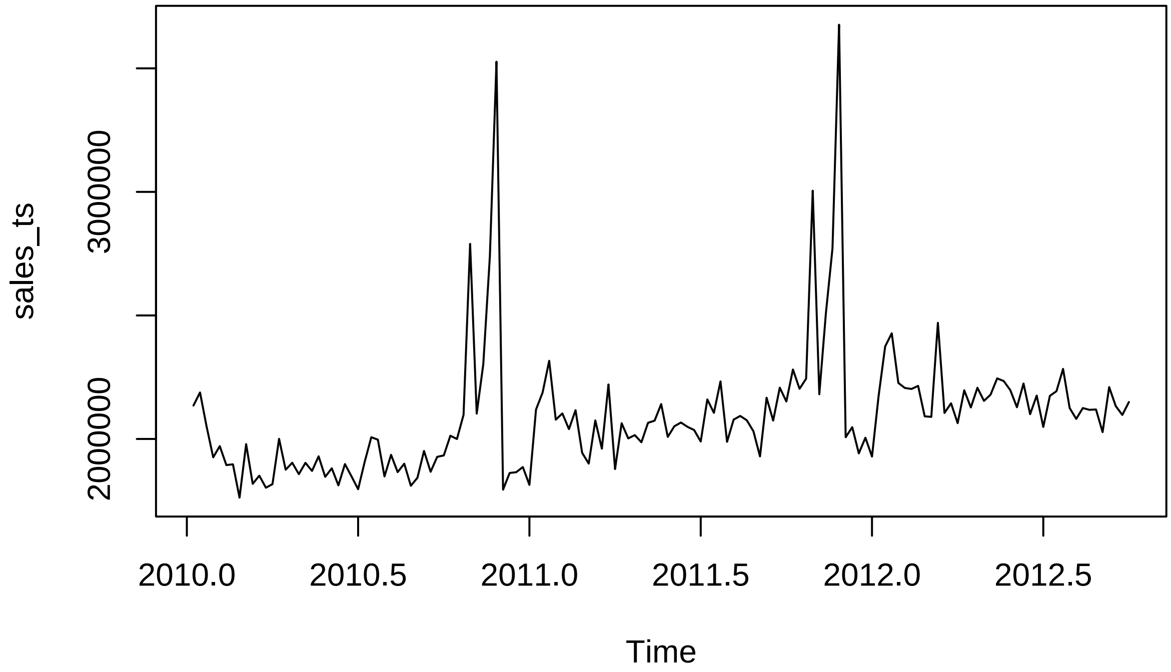
- The `ts` function is used to create the `ts` object in R. Frequency is 52 as it is weekly data. The start is specified like `start= c(y, m, d)` as we are dealing with weekly data. If it was monthly data we can omit the `d` and for yearly data we can omit the `m` as well.(`c` is the combine function in R)

```
sales_ts <- ts(sales, frequency = 52, start=c(2010, 2, 5))
sales_ts
```

```
## Time Series:
## Start = c(2010, 2)
## End = c(2012, 40)
## Frequency = 52
##   [1] 2135144 2188307 2049860 1925729 1971057 1894324 1897429 1762539 1979247
##  [10] 1818453 1851520 1802678 1817273 2000626 1875597 1903753 1857534 1903291
##  [19] 1870619 1929736 1846652 1881337 1812208 1898428 1848427 1796638 1907639
##  [28] 2007051 1997181 1848404 1935858 1865821 1899960 1810685 1842821 1951495
##  [37] 1867345 1927610 1933333 2013116 1999794 2097809 2789469 2102530 2302505
##  [46] 2740057 3526713 1794869 1862476 1865502 1886394 1814241 2119086 2187847
##  [55] 2316496 2078095 2103456 2039818 2116475 1944164 1900246 2074953 1960588
##  [64] 2220601 1878167 2063683 2002362 2015563 1986598 2065377 2073951 2141211
##  [73] 2008345 2051534 2066542 2049047 2036231 1989674 2160057 2105669 2232892
##  [82] 1988490 2078420 2093139 2075577 2031406 1929487 2166738 2074549 2207742
##  [91] 2151660 2281217 2203029 2243947 3004702 2180999 2508955 2771397 3676389
## [100] 2007106 2047766 1941677 2005098 1928721 2173374 2374661 2427640 2226662
## [109] 2206320 2202451 2214967 2091593 2089382 2470206 2105301 2144337 2064066
## [118] 2196968 2127661 2207215 2154138 2179361 2245257 2234191 2197300 2128363
## [127] 2224499 2100253 2175564 2048614 2174514 2193368 2283540 2125242 2081181
## [136] 2125105 2117855 2119439 2027620 2209835 2133026 2097267 2149594
```

- Visualize the Time-Series of `Sales` column

```
plot.ts(sales_ts)
```



Points

The problems in this worksheet are for a total of 10 points with each problem having a different weightage.

- *Problem 1*: 1 point
- *Problem 2*: 3 points
- *Problem 3*: 2 points
- *Problem 4*: 2 point
- *Problem 5*: 2 points

Problem 1 (1 Point)

Decompose the `Sales` column into trend, seasonal and random components. Plot these components as well (Hint: Look at the `decompose` function).

Problem 2 (3 Points)

- Perform forecasts using Single, Double and Triple Exponential Smoothing.
- Plot the forecasts of all three forecasts (different colours) against the true values. (Hint: use `lines`)
- **Use only one function needed for all 3 forecasts**, only changing parameters to get each of the 3 models (Hint: Think about alternate names)

Problem 3 (2 Points)

- Forecast `Sales` values by Regression using all other columns. Print summary of regression model.
- Plot the predicted values against original as well. (Hint: Regression model predictions will not be a Time Series, so need to get both to common index/x-axis)
- (Hint: Will not work unless one column is dropped/transformed before including it in the regression. Use the `lm` function to get linear model)

Note: This is Multiple Linear Regression, that is, using all the columns for regression

Problem 4 (2 Points)

Plot the `Laptop_Demand` column as a time series. Identify the forecasting required for this type of Time-series, and forecast the values for all 143 weeks (Hint: Look at functions in the `forecast` package)

Problem 5 (2 Points)

Evaluate the accuracy of all 3 Exponential Smoothing models (from Problem 2) and Regression models using the MAPE and RMSE metrics. Comment on which is the best Exponential Smoothing method, and if Regression is better than Exponential Smoothing? Provide a reasoning for why the best model is better suited for the `Sales` data (Bonus Point: reasoning for why the 2 other models perform similarly)