Learning outcomes:

1. Understanding time series data and aggregating the data based on intervals
2. Decomposing the time series and analyzing the components
3. Adjusting for stationarity, trend and seasonality in the data
4. Building the following time series models
   - Simple moving Average
   - Weighted Moving Average
   - Exponential smoothing
   - Holt-Winters model
   - ARIMA models

Activity-1: Stock Value Forecasting

a) Read the HDFC stock data

```
hdfc<- read.csv("hdfc.csv")
```

b) Consider the Close Price as Stock Price and aggregate the data at **month** level

```
# Using lubridate package extract month and year for the Day(Date) attribute
library(lubridate)
# Converting the date field to appropriate type
day = mdy(hdfc$day)
# Extracting month from date field
mth = month(day)
# Extracting year from date field
yr = year(day)
# Group the data by month and year and considering the mean value of the stock
hdfc1 =  data.frame("Price"=hdfc$Close,yr,mth)
hdfc1 = aggregate(Price ~ mth + yr, hdfc1, mean)
```

c) Create a time series object and plot it

```
priceTimeSeries = ts(hdfc1$Price, start = c(2011,1), frequency = 12)
plot.ts(priceTimeSeries)
```

d) Decompose the time series and plot the components

```
decomposed_priceTimeSeries = decompose(priceTimeSeries)

par(mfrow=c(1,1))

plot(decomposed_priceTimeSeries)
```

e) Study ACF, PACF plots to find auto and partial auto correlations manually and check if data has predominant trend and seasonal components

```
par(mfrow=c(1,2))

acf(priceTimeSeries)

pacf(priceTimeSeries)

#shows significant trend, and no seasonality
```

f) Use moving averaging models I.e. SMA/WMA/EMA for smoothing the data

```
library(TTR)

par(mfrow=c(1,1))

sma_hdfc = SMA(priceTimeSeries,n=7)

wma_hdfc = WMA(priceTimeSeries,n=7)

ema_hdfc = EMA(priceTimeSeries,n=7)

par(mfrow=c(1,1))

plot(priceTimeSeries,type = 'l', col = "blue")

lines(sma_hdfc, col = "black")

lines(wma_hdfc, col = "red")

lines(ema_hdfc, col = "green")
```

g) Calculating errors on the averaging models

```
errorsma = mean(abs(priceTimeSeries[7:56]-sma_hdfc[7:56]))

errorwma = mean(abs(priceTimeSeries[7:56]-wma_hdfc[7:56]))

errorema = mean(abs(priceTimeSeries[7:56]-ema_hdfc[7:56]))

errorsma

errorwma

errorema
```

h) Apply Holt-Winters model for forecasting for next four time periods

```
# Building a Holt-Winter's model

# Train data to build the models

priceTimeSeries = ts(hdfc1$Price[1:52], start = c(2011,1), frequency = 12)

price_HW = HoltWinters(priceTimeSeries, gamma = FALSE)

# As there's no seasonality gamma is FASLE; As there is a trend it is left with its #default value – TRUE

#Forecasting for last 5 time periods

library(forecast)

price_hw_forecasts = forecast(price_HW,h=4)

hw_preds <- data.frame(price_hw_forecasts)$Point.Forecast

actuals <- hdfc1$Price[53:56]

#Compute error between actual and predicted values

mean(abs(hw_preds-actuals))

library(DMwR)

regr.eval(trues = actuals, preds = hw_preds)
```

i) Applying ARIMA models: We need stationary data to apply ARIMA models. As we have trend in the data, we difference the data to make it stationary, as a first step

```
# Method1:

Use diff function to difference, plot the difference data and decide on number of #differences to be done, manually:

par(mfrow=c(1,4))

plot.ts(priceTimeSeries,main="Actual Data")

#Differencing the time series and make it stationary; diffs = 1

timeseriesdiff1 <- diff(priceTimeSeries, differences=1)

plot.ts(timeseriesdiff1,main="Data with one difference")

#Differencing the time series and make it stationary; diffs = 2
```

```
timeseriesdiff2 <- diff(priceTimeSeries, differences=2)
plot.ts(timeseriesdiff2,main="Data with two differences")


#Differencing the time series and make it stationary; diffs = 3
Timeseriesdiff3 <- diff(priceTimeSeries, differences=3)
plot.ts(timeseriesdiff3,main="Data with two differences")
#From the plots we infer that we can set ndiffs at 1 or 2.
#Method2: Use ndiff() function to evaluate number of differences to be done
ndiffs(priceTimeSeries)
#Once number of diffs is finalized, difference the data and make it stationary
price_stationary_ts = diff(priceTimeSeries,differences = 1)
par(mfrow=c(1,1))
plot.ts(price_stationary_ts)
```

j) Inspect ACF and PACF plots of stationary data and choose p,d,q values;
(Note: As this is non-seasonal data, p,q are chosen from the plots for building ARIMA(p,d,q) models; If there's seasonality we need to choose P,Q values along with p,q values and then apply ARIMA(p,d,q)(P,D,Q))

```
par(mfrow=c(1,2))
acf(price_stationary_ts)
pacf(price_stationary_ts)
#From the PACF plot of differenced data, we infer that p = 0
#From the ACF plot of differenced data, we infer that q = 0
#Number of diffs we used to difference d=1
#There fore our ARIMA model is: ARIMA(0,1,0)
```

k) Build an ARIMA model on first 52 time periods and forecast for the last 4 time periods; evaluate the error metrics
NOTE: Stationary data is used to look at ACF & PACF plots and to choose q, and p values; Once they are chosen along with d, we give our normal (non-stationary) data as input to ARIMA function, along with the order(p,d,q)

```
#Model
price_arima <- arima(priceTimeSeries,order = c(0,1,0))
# Forecasting on the arima model
price_arima_forecasts <- forecast.Arima(price_arima, h=4)
# Creating a data frame using forecasted value
preds <- data.frame(price_arima_forecasts)$Point.Forecast
actuals <- hdfc1$Price[53:56]
#Error
library(DMwR)
regr.eval(trues = actuals, preds = preds)
```

l) Build an Auto-ARIMA model, forecast for the last 4 time periods, and evaluate the error
Reference:

```
# Building auto arima model
autoArimaModel <- auto.arima(priceTimeSeries)
# Forecasting for the last 4 months using auto arima model
autoarima_forecast <- data.frame(forecast(autoArimaModel,h=4))$Point.Forecast
regr.eval(trues = actuals, preds = autoarima_forecast)
```

m) Compare the results of Holt-Winters, ARIMA and Auto-ARIMA models, and choose the model which gives least error

Activity-2:

DeliveryRate.csv: Data represents the delivery rate of the product for each month.
Data consists of 3 years of data at month level from January 2013 to December 2015.

Build a monthly time series models to forecast the delivery rates at month level.

Steps:

1. Read the data DeliveryRate.csv into R.
2. Create a time series object by choosing the appropriate frequency value and plot it.
3. Decompose the time series and plot the components, identify the trend, seasonality and randomness in the data.
4. Study ACF, PACF plots to find auto and partial auto correlations manually and check if data has predominant trend and seasonal components.
5. Build the moving average models I.e. SMA/WMA/EMA for smoothing the data.
6. Calculate the error metrics for the moving average models.
7. Apply Holt-Winters model for forecasting for last four time periods, by identifying the appropriate parameters.
8. Applying ARIMA models on the data, by identifying the appropriate p,d,q values manually and by running auto.arima.
9. Compare the results of Holt-Winters, ARIMA and Auto-ARIMA models, and choose the model which gives least error.