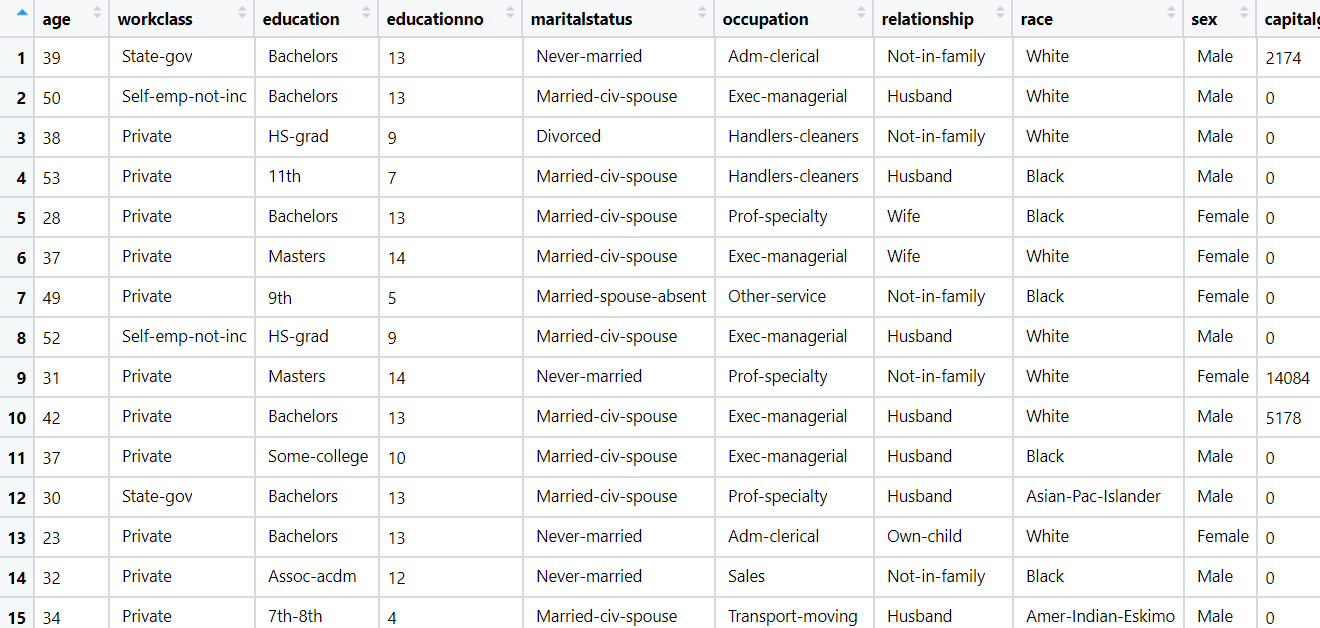
# Topic: Support Vector Machine (SVM)

1) Prepare a classification model using SVM for salary data



**Ans: Support Vector Machines for classification model**

**Analyzing the input and output variables:**

* Output Variable(y) - Salary
* Input Variable (x) - Other Factors

**Importing data and Data Preprocessing:**

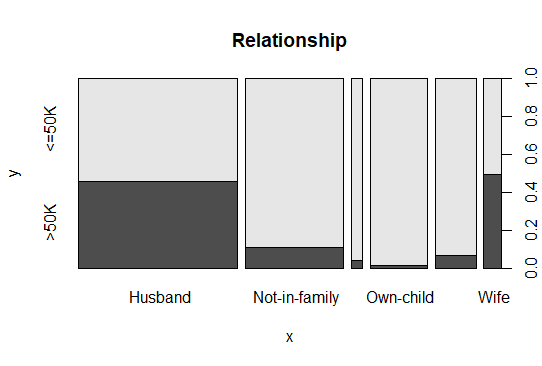
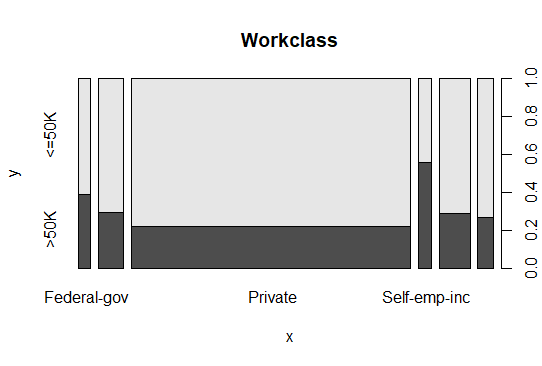
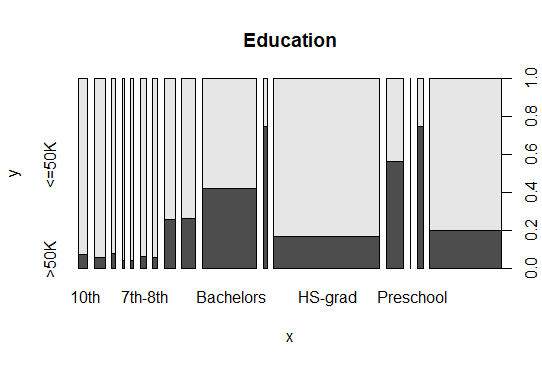
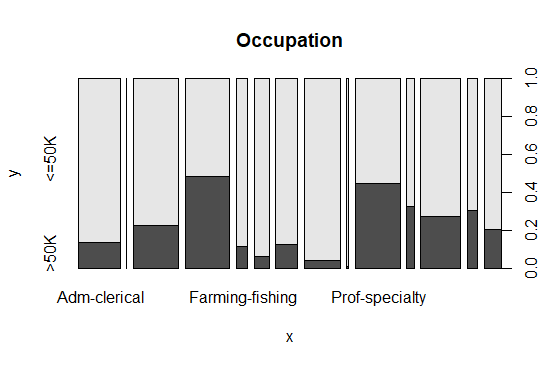
* Data file is imported by Text(base) to convert strings into factors.
* As there are no missing values and no imputation of data required.

**Normalizing the data:**

* Using the normalization function, the data is normalized and the data will come under same level.

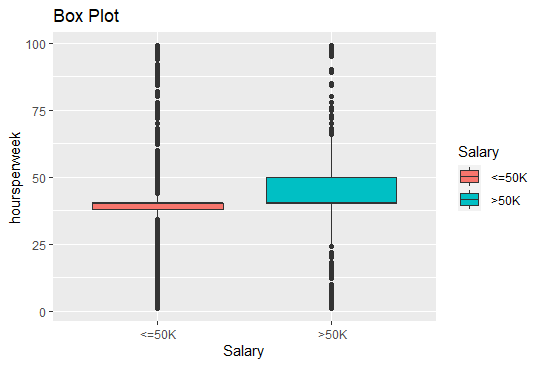
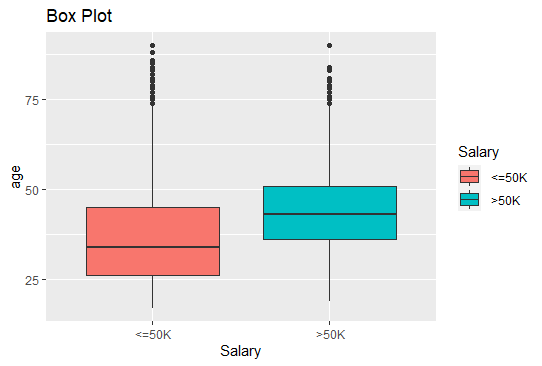
**Exploratory Data Analysis:**

**Plot Representation:**

****

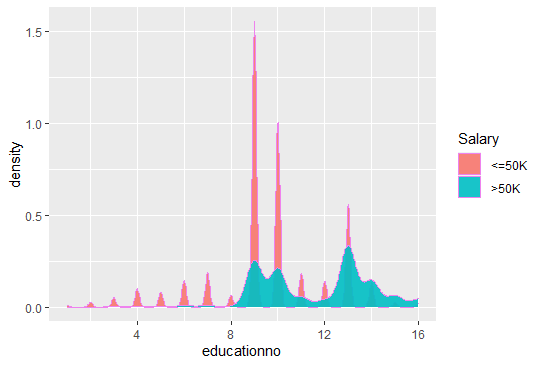
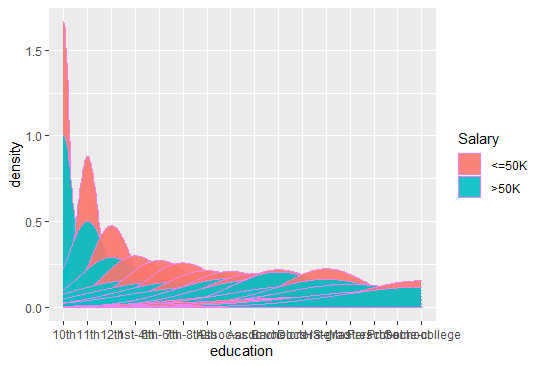
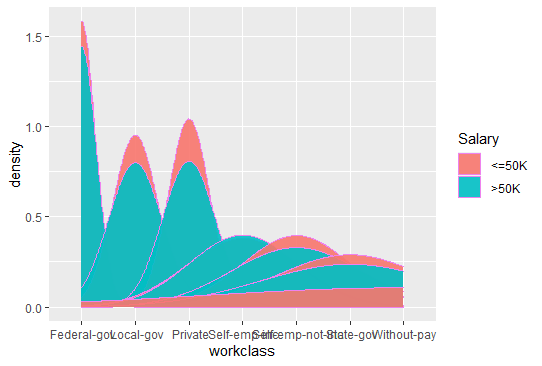
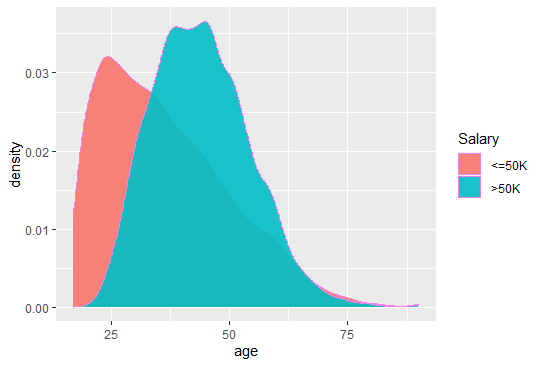
* From the above graphical representation, we can clearly see how the data is spread in each individual variable.

**Box Plot Representation:**

****

* The box plots represent that there exist outliers in the input and output variables.

**Density Plots:**

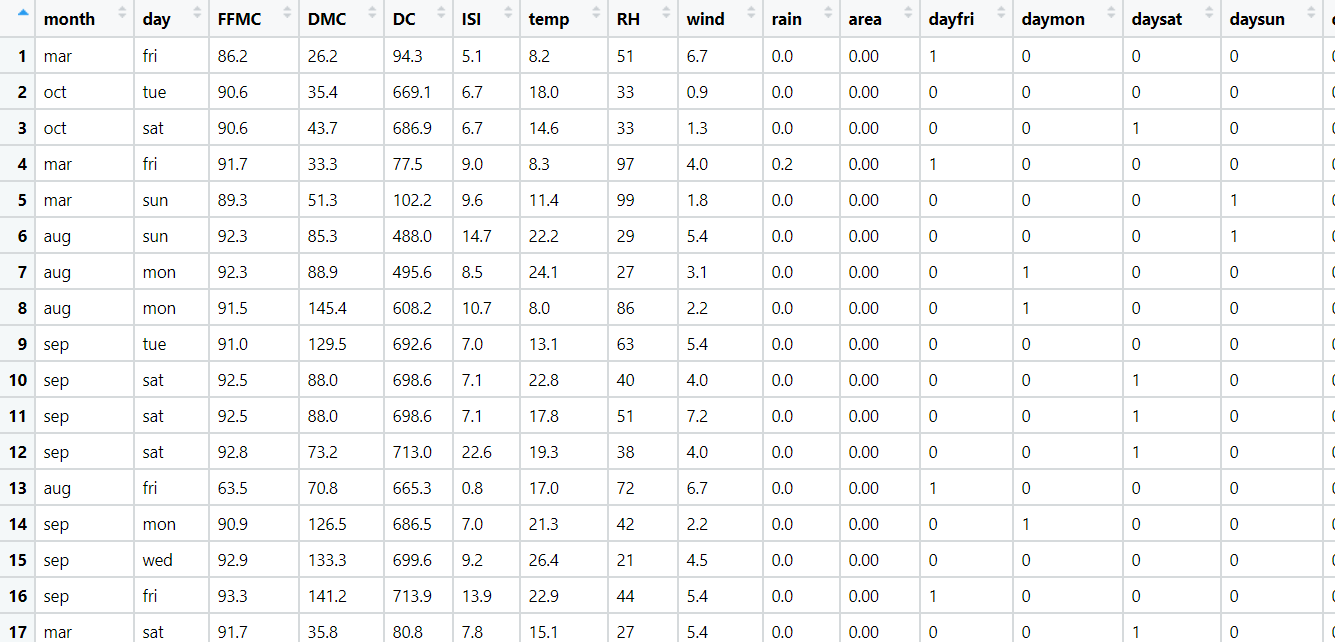
****

* The density plot observes the distribution of data of a variable.

**Model Building:**

* Building a model using a kernel parameter as “**vanilla dot**” and predicting the model on the test data
* The accuracy of the model from “**vanilla dot**” is 0.846
* Now changing the kernel parameter to “**rbfdot”** to improve the model performance.
* The accuracy of the model from “**rbfdot”** is 0.8544
* By comparing above 2 model the model with “**rbfdot”** kernel parameter is the best model as it gives more accuracy.

2.) Prepare svm model for classifying the area under fire for forest fires data



**Ans:**

**Analyzing the input and output variables:**

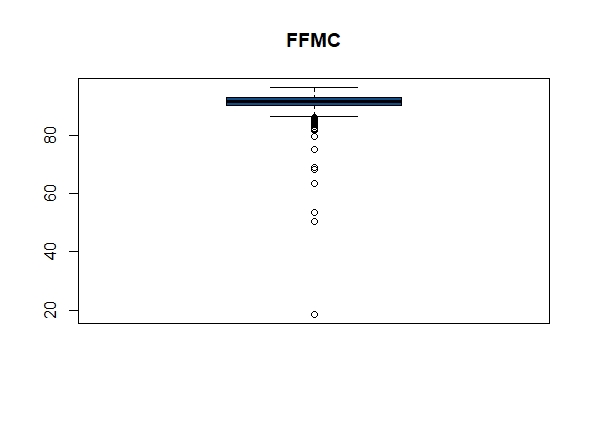
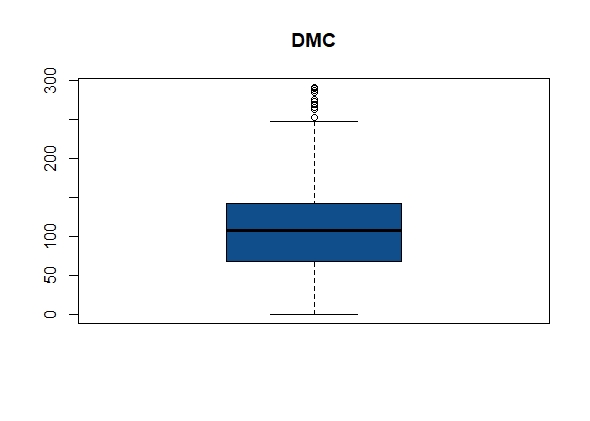
* Input Variables (x) = Other Factors influencing the output variable
* Output Variable(y) = area

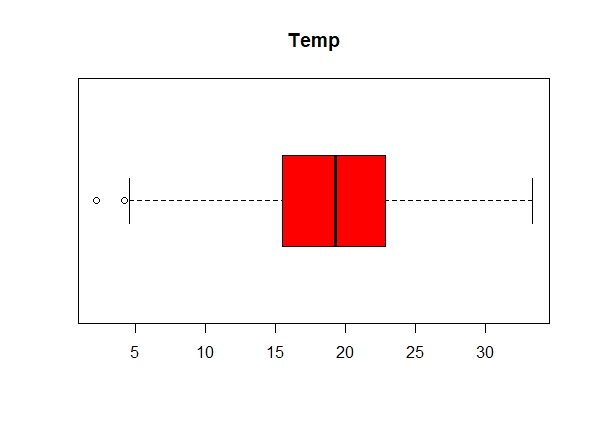
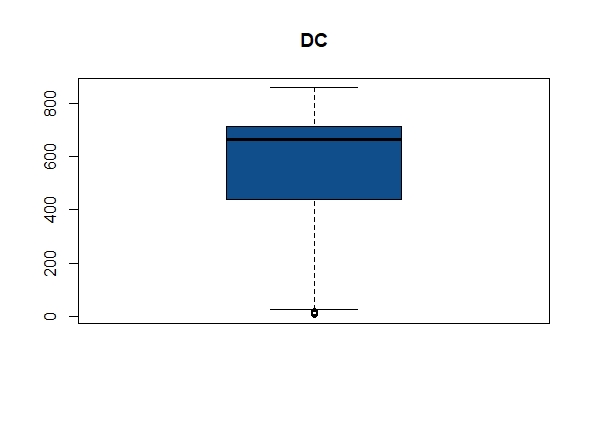
**Data Preprocessing:**

* To make the easy access of variables of input and output, columns are rearranged.
* In R , dummy variable are created automatically when object(x) is created as model. Matrix command has inbuild feature of converting dummy variables or we can factorize the variables whereas, in python we need to write label encoder () code for creation of dummy variables.
* Removing unnecessary columns and checking for the NA value

**Exploratory Data Analysis:**

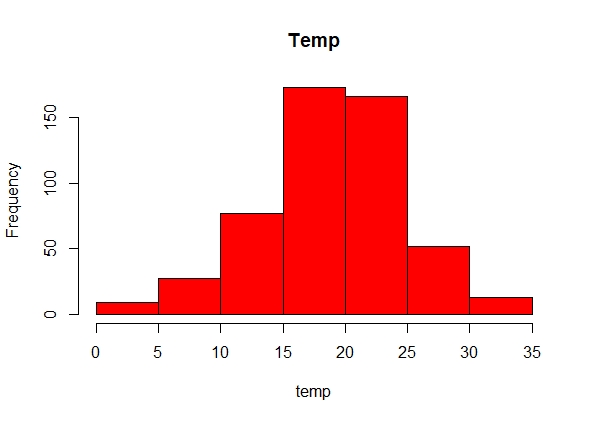
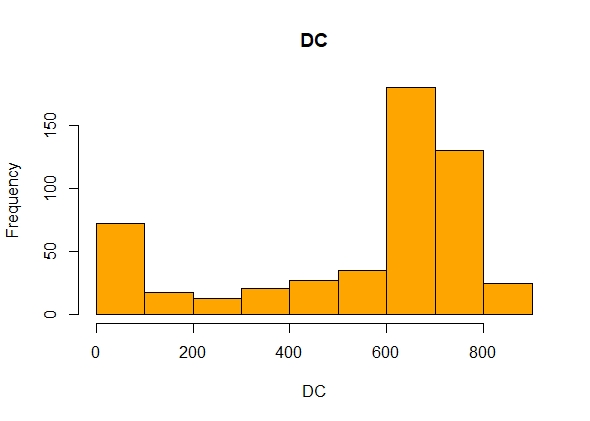
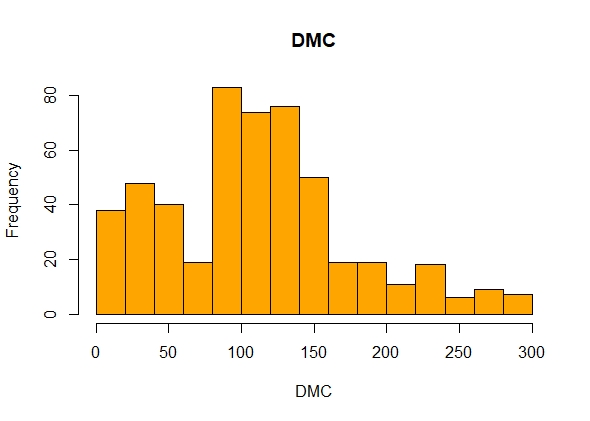
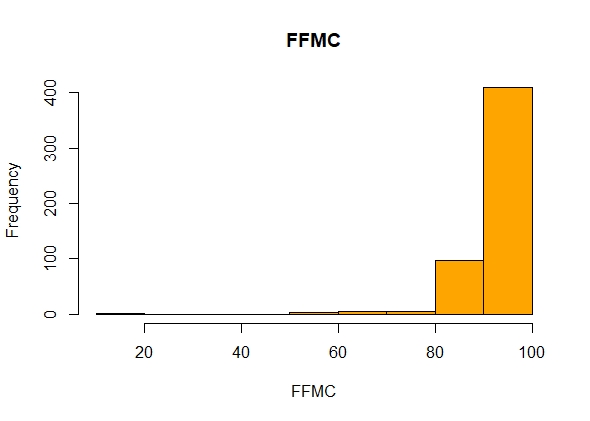
**Boxplot Representation:**

****

****

* The above boxplot represents there exist outlier in the individual variables

**Histogram Representation:**

****

* From the above graphs data is normally distributed in case of temp and DMC but incase of FFMG and DC the data is right skewed

**Normalizing the data:**

* Using the normalization function, the data is normalized and the data will come under same level.

**Converting output variable into categorical:**

* The output variable “**area**” is converting into 2 level categorical variable and the variable data is factorized.

**Model Building:**

* Now the model is built on train data by taking 400 observations into train data and building the model with different kernel methods to check the training error.
* Building a model using a kernel parameter as “**polydot**” and predicting the training error which gives as 0.3
* Building a model using a kernel parameter as “**rbfdot**” and predicting the training error which gives as 0.29
* Building a model using a kernel parameter as “**tanhdot**” and predicting the training error which gives as 0.47
* By comparing above 3 model the model with “**rbfdot”** kernel parameter gives less training error.
* Now with “**rbfdot**” kernel parameter, the model is tested on test data to find the accuracy.
* The accuracy rate of the test data is 0.74

**Hints:**

1. Business Problem
   1. Objective
   2. Constraints (if any)
2. Data Pre-processing

2.1 Data cleaning, Feature Engineering, EDA etc.

1. Model Building
   1. Partition the dataset
   2. Model(s) - Reasons to choose any algorithm
   3. Model(s) Improvement steps
   4. Model Evaluation
   5. Python and R codes
2. Deployment

4.1 Deploy solutions using R shiny and Python Flask.

1. Result Share the benefits/impact of the solution - how or in what way the business (client) gets benefit from the solution provided.

**Note:**

1. For each assignment the solution should be submitted in the format
2. Research and Perform all possible steps for improving the model(s) accuracy Ex: Feature Engineering, Hyper Parameter tuning etc.
3. All the codes (executable programs) are running without errors
4. Documentation of the module should be submitted along with R & Python codes, elaborating on every step mentioned here