**Prepare a classification model using SVM for salary data.**

**Inferences from the Data Set:**

Data Set talks about the Salary dataof persons dataset given/provided as train and test data with respect to around 14 variables.

**Box plot:**

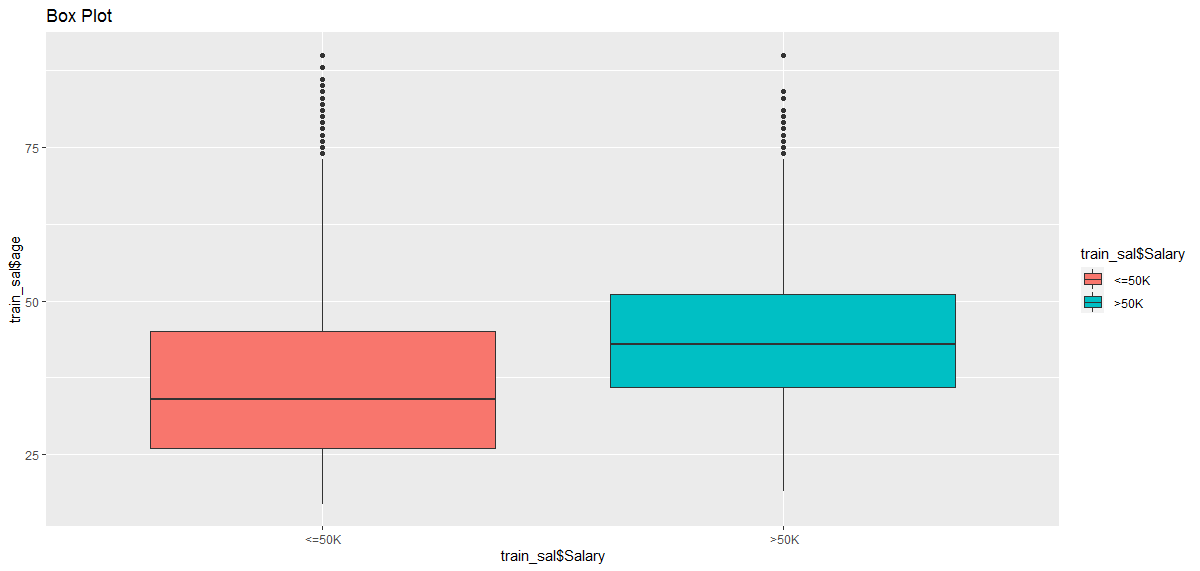
plot(train\_sal$sex,train\_sal$Salary)



The above plots explains as comparison between male and female and male persons are earning more income(>50k) than female persons.

Below Box plot shows between **age** and **salary**

And the persons having more than approximate age 44 are earning greater than 50k(>50k)



Measures of central tendency explains about mean, median and mode of the data set.

Measures of dispersion explains about the standard deviation variation and range of the data set.

Third moment business decision and Fourth moment business decision explains about the skewness and kurtosis of the data set.

Divide the dataset into train and test dataset with normal splitting of the data.(Dataset provided splitting the data into train and test data set)

Training a model on the data begin by training a simple linear SVM before that we have to install the packages called “**kernlab”**

In this kernel we will explore the data and try to predict if an adult earn more than 50k per year with a accuracy of more than 80%. To accomplish that, we will use the Support Vector Machine.

**Method/ Building Model**

We use the kernlab package and the ksvm() function therein to fit an SVM using a non-linear kernel. We can use the argument kernel = "rbfdot" for a radial basis and " vanilladot" for the complicated sounding hyperbolic tangentsigmoid.

Note the hugh amount of parameter customisation that is possible at this stage. For simplicity we use the default settings which will be far from optimal.

Using the simple defaults, the radial basis non-linear mapping for the SVM appears equivalent to the **vanilladot**, based on the lower training error; with the model slightly better. We should evaluate the model performance using the **predict()** function. In order to examine how well our classifier performed we need to compare our predicted size of the fire with the actual size in the test dataset.

**Evaluating model performance and predictions on testing dataset:**

We are going to predict the model using the model Salary data set and test data.

And by using the confusion matrix table we can predict the **Salary** of the dataset/person.

table(Salary\_prediction,test\_sal$Salary)

Salary\_prediction <=50K >50K

<=50K 10601 1554

>50K 759 2146

> prop.table(table(agreement))

agreement

FALSE TRUE

0.1535857 0.8464143

We can get the accuracy of the model by using the proportion of the table , predictions of the dataset to the Salary dataset of output variable**(Salary)**

We get the 86% of accuracy of model by using **kernel = "** **rbfdot ".**