**ASSIGNMENT NO: 10**

**TITLE:** Download the Iris flower dataset. Compute and Summarize statistics for features of dataset, perform data visualization and plot histogram as well as boxplot for the same.

**PROBLEM STATEMENT: Data Visualization III**

Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., https://archive.ics.uci.edu/ml/datasets/Iris ). Scan the dataset and give the inference as:

1. List down the features and their types (e.g., numeric, nominal) available in the dataset.
2. Create a histogram for each feature in the dataset to illustrate the feature distributions.
3. Create a boxplot for each feature in the dataset.

Compare distributions and identify outliers.

**OBJECTIVES:** learn classification techniques how to plot histogram and boxplot for given dataset.

**PRE-REQUISITES:** R-programming, Rstudio.

**THEORY:** Let’s design a basic data analysis program in R using R Studio by utilizing the features of R Studio to create some visual representation of that data.

**How to Install RStudio?**

We have to follow three basic steps in same order to run R and R Studio on your system.

* + Install R (Download the binary setup file for R from the following link. Open the downloaded .exe file and Install R.)
  + Install RStudio
  + Install R Packages

Following steps will be performed to achieve our goal.

* + Downloading/importing data in R
  + Transforming Data / Running queries on data
  + Basic data analysis using statistical averages
  + Plotting data distribution

# Typical data analysis process

Data analysis deals with collecting, inspecting, cleansing, transforming and modelling data to glean valuable insights and support better decision making in an organization. The various steps involved in the data analysis process include –

# Data Exploration –

Having identified the business problem, a data analyst has to go through the data provided by the client to analyse the root cause of the problem.

# Data Preparation

This is the most crucial step of the data analysis process wherein any data anomalies (like missing values or detecting outliers) with the data have to be modelled in the right direction.

# Data Modelling

The modelling step begins once the data has been prepared. Modelling is an iterative process wherein the model is run repeatedly for improvements. Data modelling ensures that the best possible result is found for a given business problem.

# Validation

In this step, the model provided by the client and the model developed by the data analyst are validated against each other to find out if the developed model will meet the business requirements.

# Implementation of the Model and Tracking

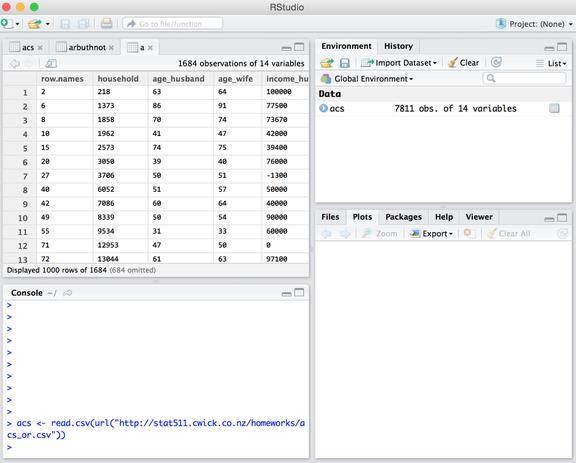
This is the final step of the data analysis process wherein the model is implemented in production and is tested for accuracy and efficiency.

1. Importing Data in R Studio:-

For this tutorial we will use the sample census data set [ACS](http://stat511.cwick.co.nz/homeworks/acs_or.csv) . There are two ways to import this data in R. One way is to import the data programmatically by executing the following command in the console window of R Studio

*acs <- read.csv(url(*[*"http://*](http://stat511.cwick.co.nz/homeworks/acs_or.csv)*s*[*tat511.cwick.co.nz/homeworks/acs\_or.csv*](http://stat511.cwick.co.nz/homeworks/acs_or.csv)*"))*

Once this command is executed by pressing Enter, the dataset will be downloaded from the internet, read as a *csv* file and assigned to the variable name *acs.*



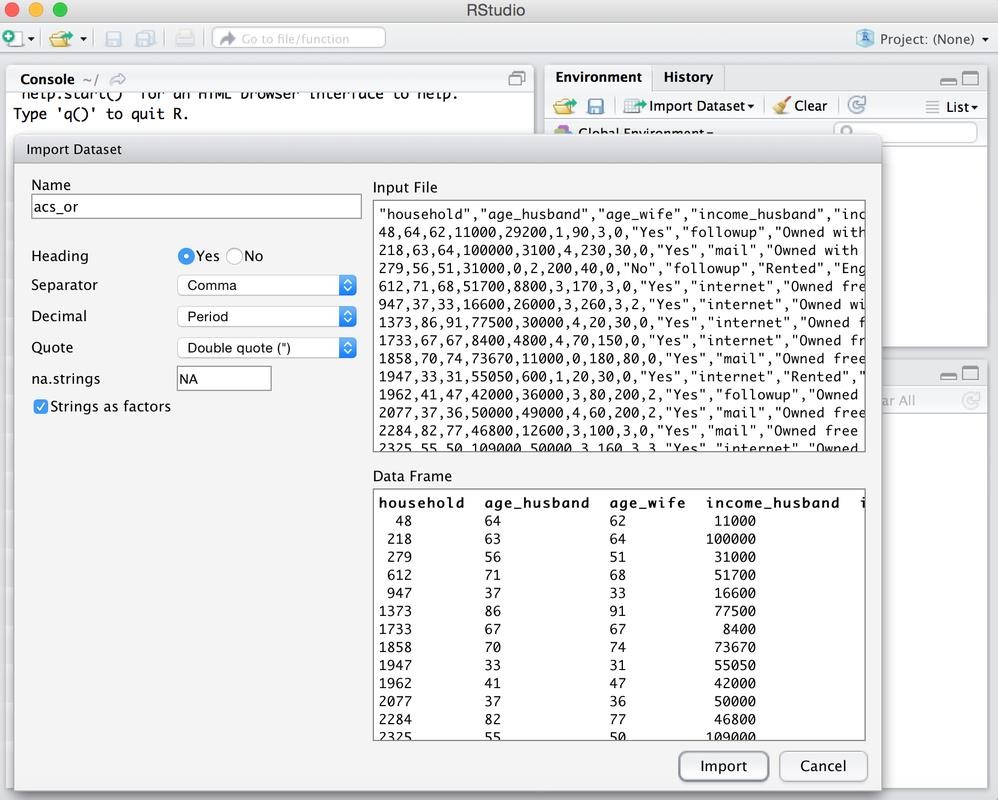
The second way to import the data set into R Studio is to first download it onto you local computer and use the *import dataset* feature of R Studio. To perform this follows the steps below

1. Click on the *import dataset* button in the top-right section under the environment tab. Select the file you want to import and then click open. The Import Dataset dialog will appear as shown below
2. After setting up the preferences of separator, name and other parameters, click on the Import button. The dataset will be imported in R Studio and assigned to the variable name as set before.

Any dataset can be viewed by executing the following line:

View(acs)

where acs is the variable dataset is assigned to.



2. Transforming Data

Once you are done with importing the data in R Studio, you can use various transformation features of R to manipulate the data. Let's learn few of the basic data access techniques

* To access a particular column, Ex. age\_husband in our case. acs$age\_husband
* To access data as a vector acs[1,3]
* To run some queries on data, you can use the *subset* function of R. Let's say I want those rows from the dataset in which the age\_husband is greater than age\_wife. For this we’ll run the following command in console a <- subset(acs , age\_husband > age\_wife)

The first parameter to the subset function is the dataframe you want to apply that function to and the second parameter is the boolean condition that needs to be checked for each row to be included or not. So the above statement will return the set the rows in which the age\_husband is greater than age\_wife and assign those rows to a

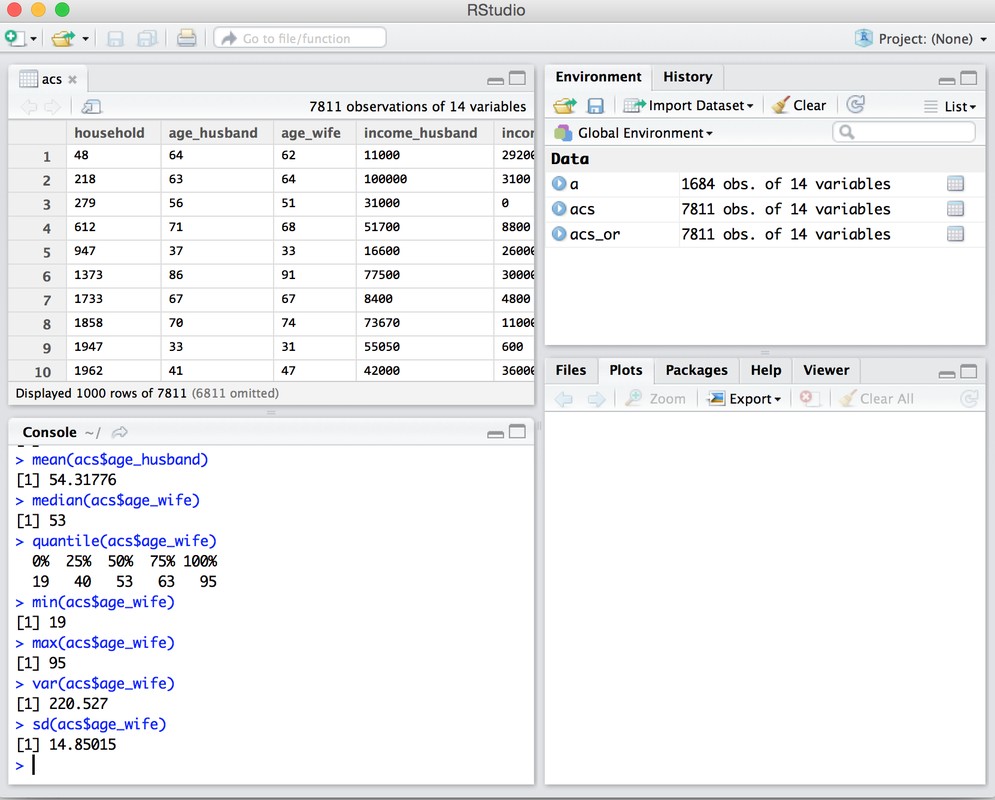
# Getting Statistical Averages from data

Following functions can be used to calculate the averages of the dataset

* For mean of any column, run : mean(acs$age\_husband)
* Median, run : median(acs$age\_husband)
* Quantile , run : quantile(acs$age\_wife)
* Variance , run : var(acs$age\_wife)
* Standard Deviation , run : sd(acs$age\_wife)

You can also get the statistical summary of the dataset by just running on either a column or the complete dataset

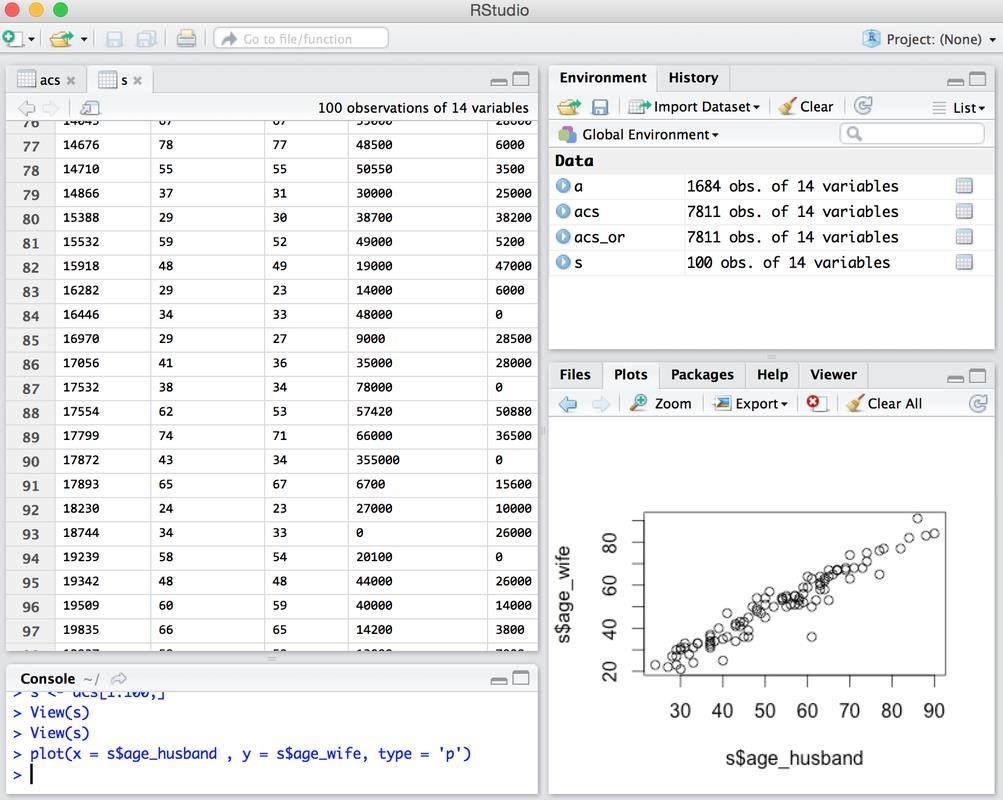
summary (acs)



# 4. Plotting Data

A very liked feature of R studio is its built in data visualizer for R. Any data set imported in R can visualized using the plot and several other functions of R. For Example

To create a scatter plot of a data set, you can run the following command in console plot(x = s$age\_husband , y = s$age\_wife, type = 'p')



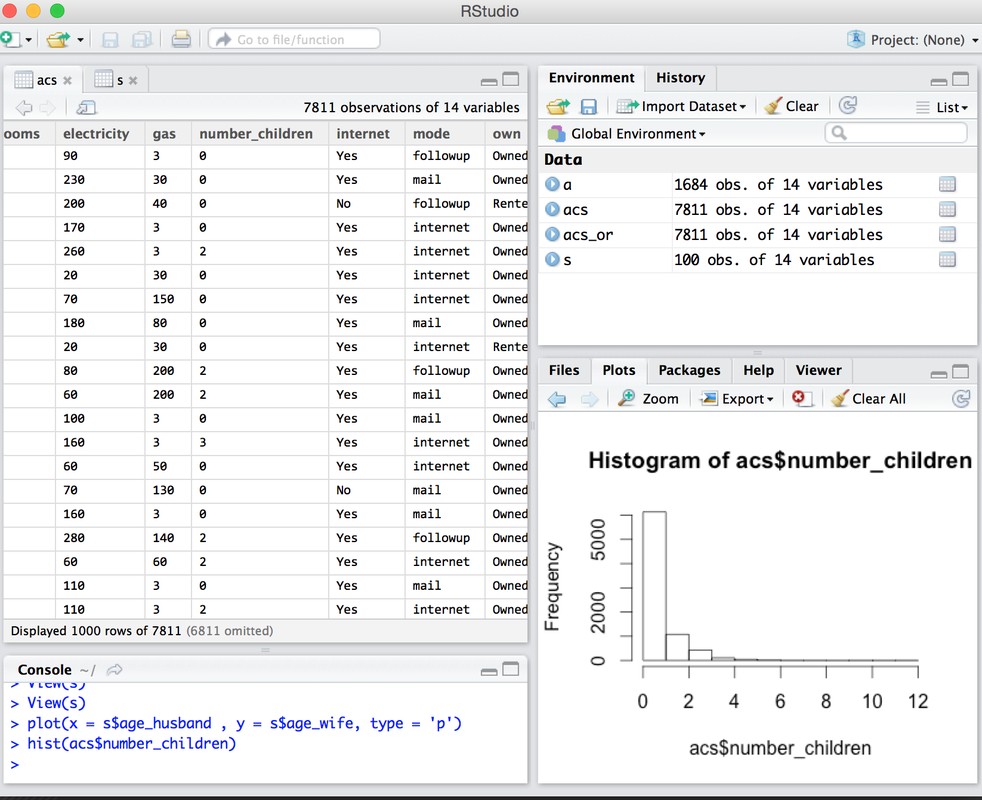
Where s is the subset of the original dataset and type 'p' set the plot type as point. You can aslo choose line and other change type variable to 'L' etc.

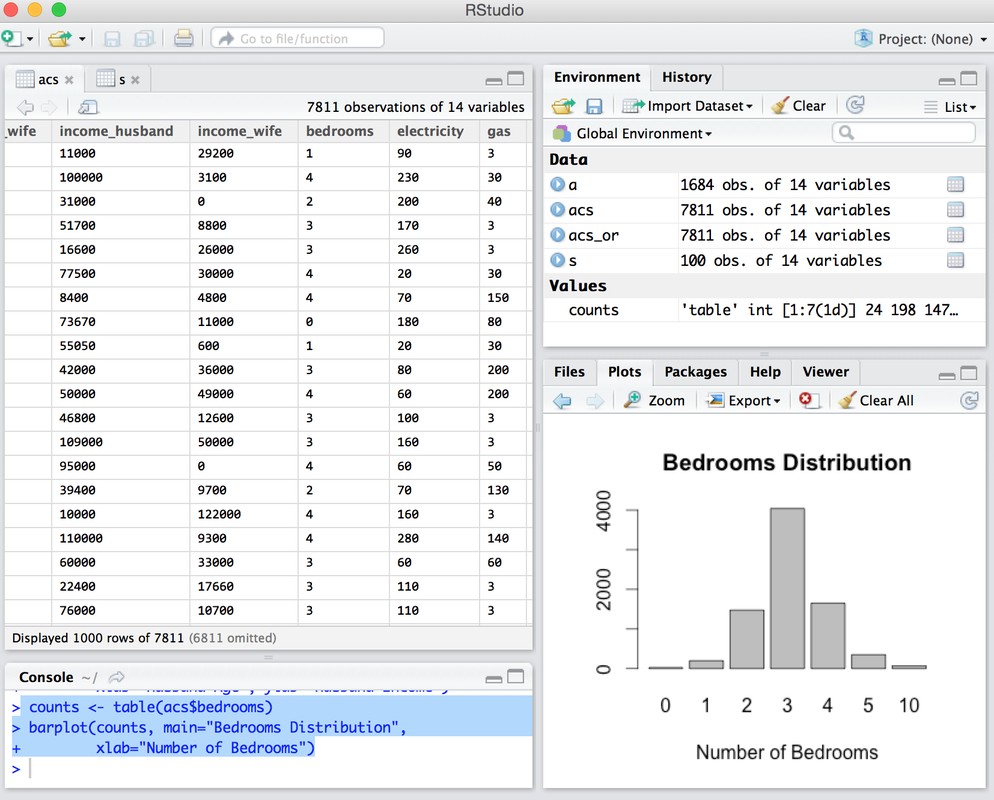
For data distribution plots, there are several features tools and packages available in R that you can use to draw any kind of distribution. For example

To draw a Histogram of a dataset, you can run the command hist(acs$number\_children)

Similarly for Bar Plots, run the following set of commands counts <- table(acs$bedrooms)

barplot(counts, main="Bedrooms Distribution", xlab="Number of Bedrooms")





I hope this will give you a basic idea on how to do simple statistics in R.

# CONCLUSION:

Thus we have learned how to use features of Rstudio when learning data visualization. How to create histogram for each feature of dataset, boxplot etc.

# Questions:-

1. Why do we need Analytics?
2. What is Business Analytics?
3. Why R and Who uses R?
4. Data types used in R.
5. What is the difference between Data Mining and Data Analysis?