R Studio Tutorial (index.html)

Basic Data Analysis through R/R Studio

In this tutorial, I 'll 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. Following steps will be performed to achieve our goal.

1. Downloading/importing data in R
2. Transforming Data / Running queries on data
3. Basic data analysis using statistical averages
4. Plotting data distribution

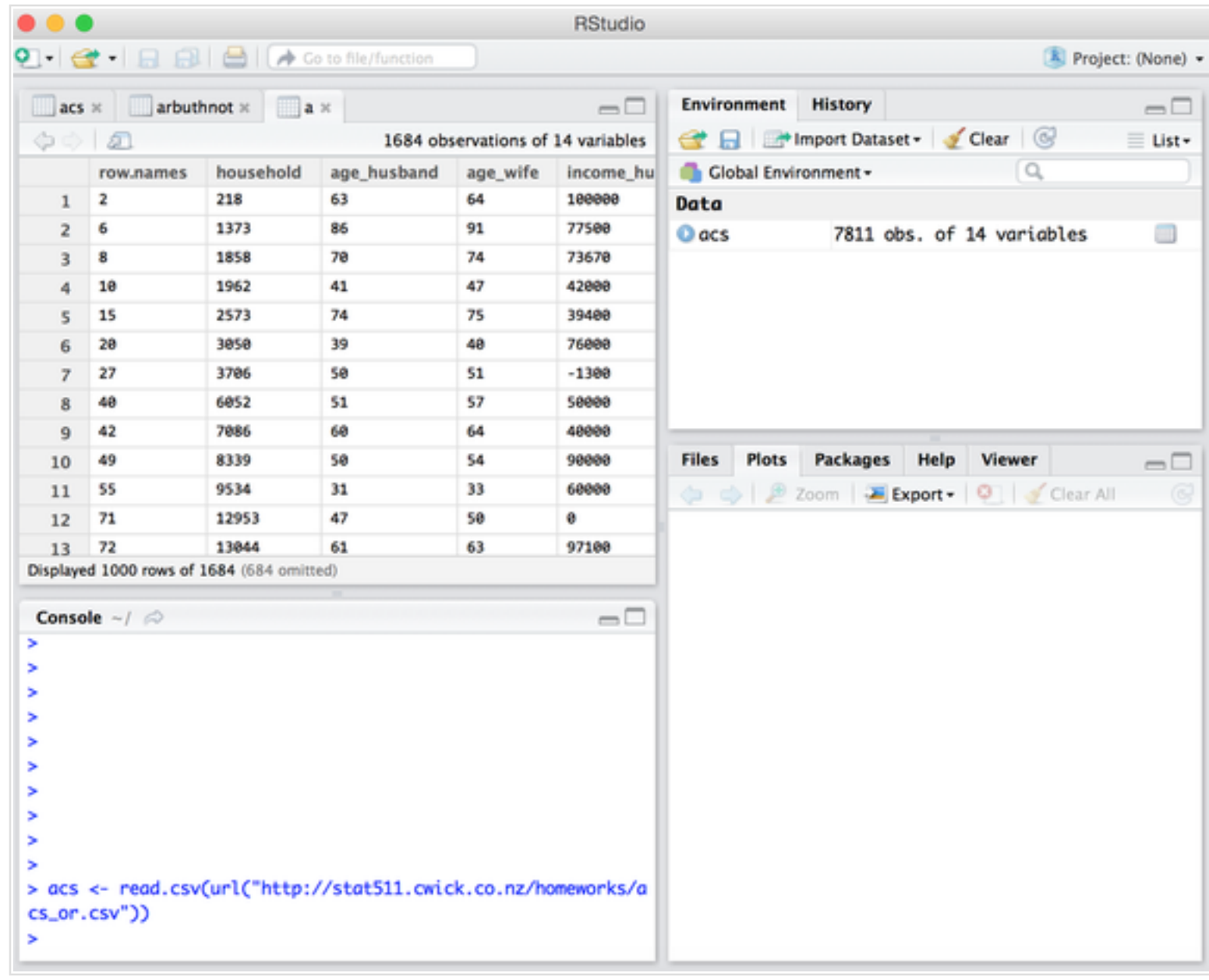
Let's go over the tutorial by performing one step at a time.

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

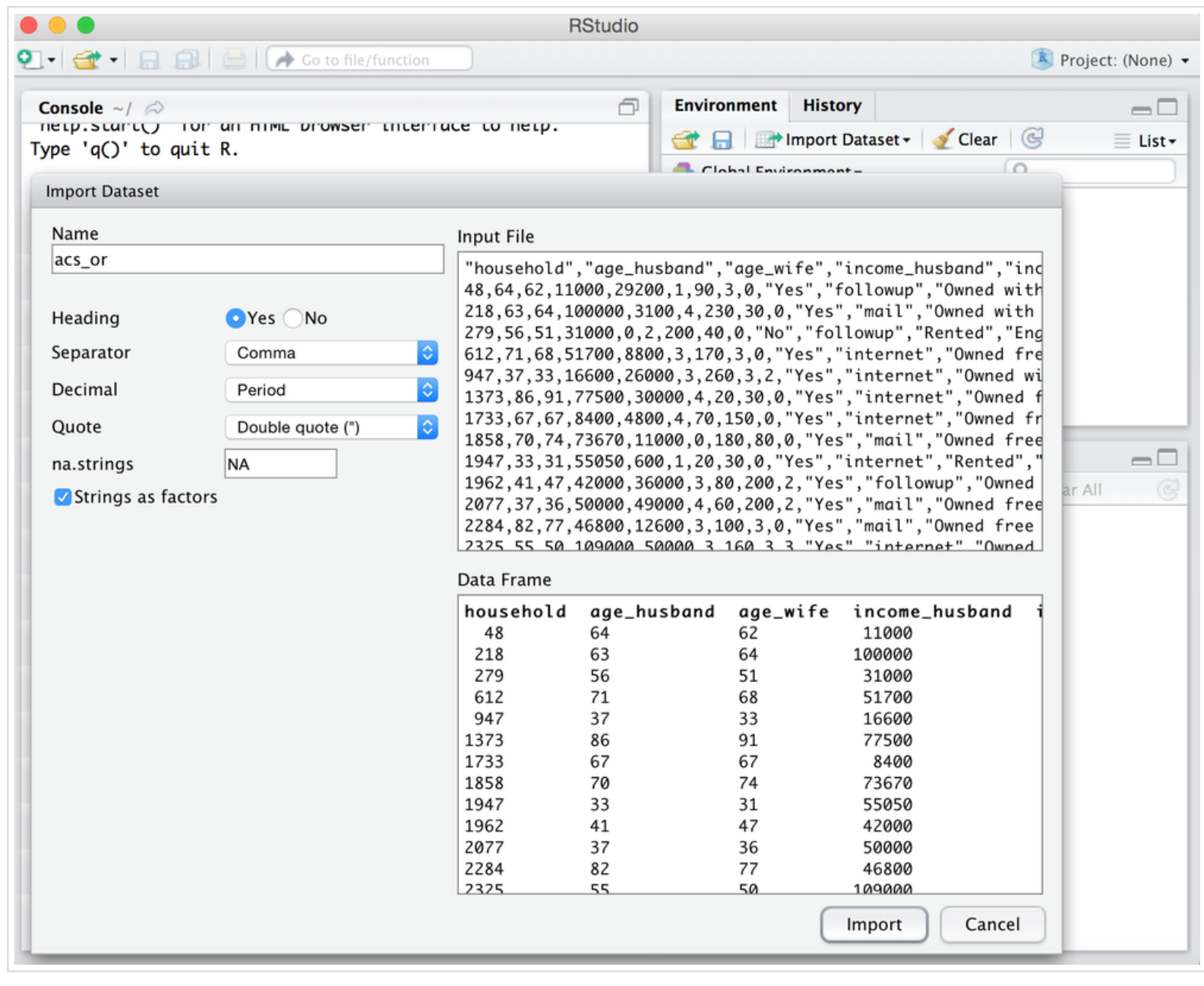
acs <- read.csv(url("http://stat511.cwick.co.nz/homeworks/acs_or.csv"))

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 follow 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

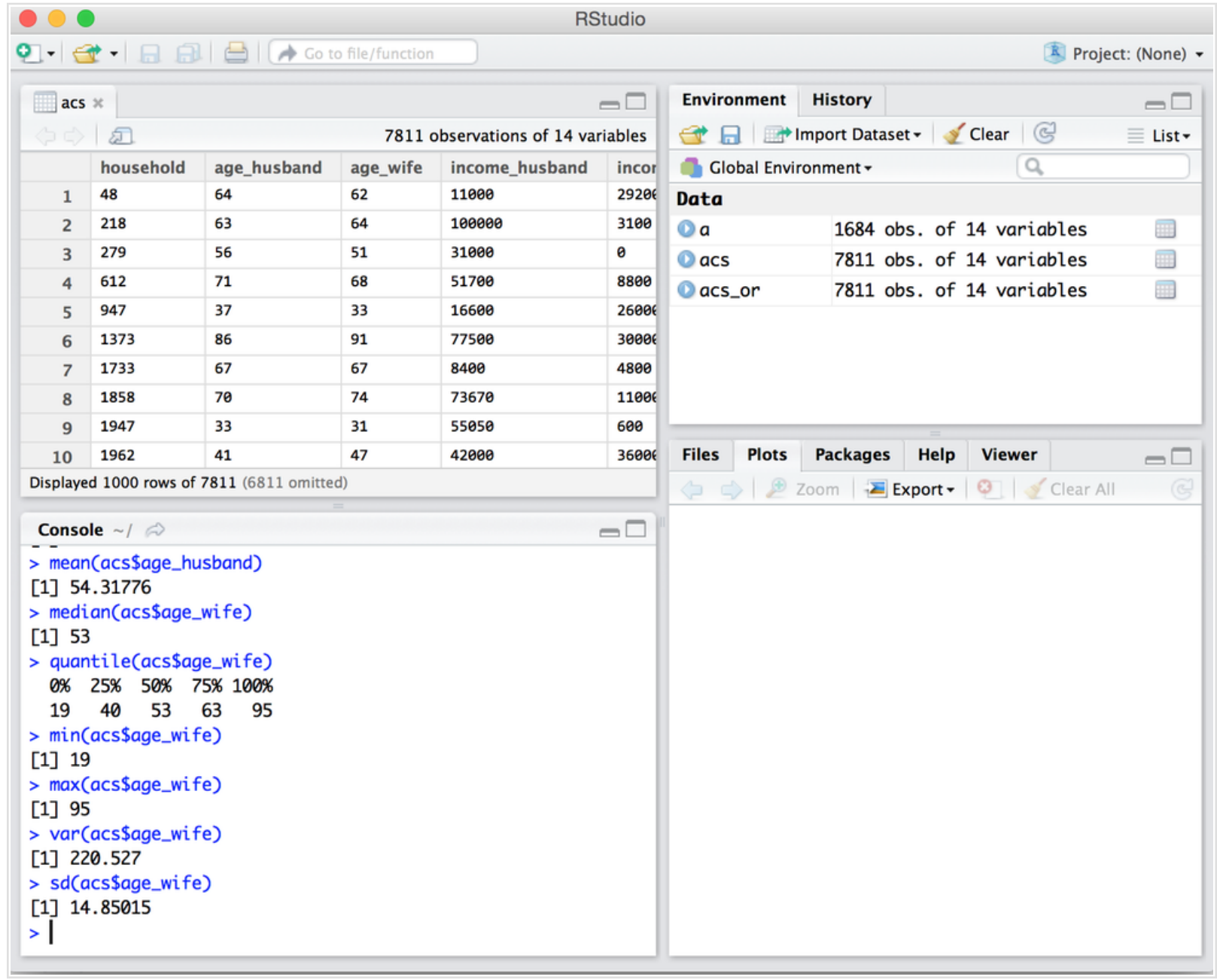
```
a <- subset(acs , age_husband > age_wife)
```

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

1. For mean of any column, run : mean(acs$age_husband)
2. Median, run : median(acs$age_husband)
3. Quantile , run : quantile(acs$age_wife)
4. Variance , run : var(acs$age_wife)
5. 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)
```

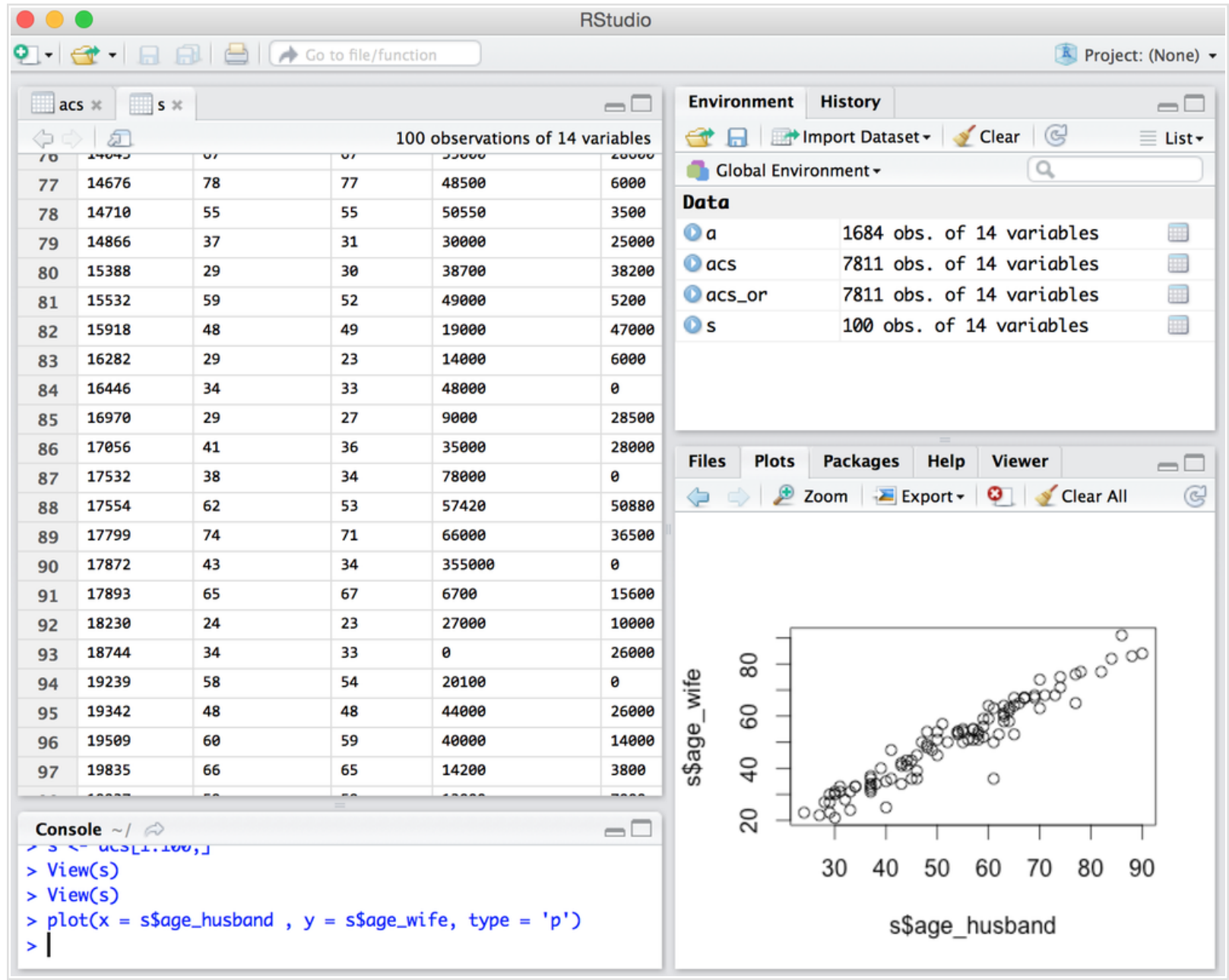
4. Plotting Data

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

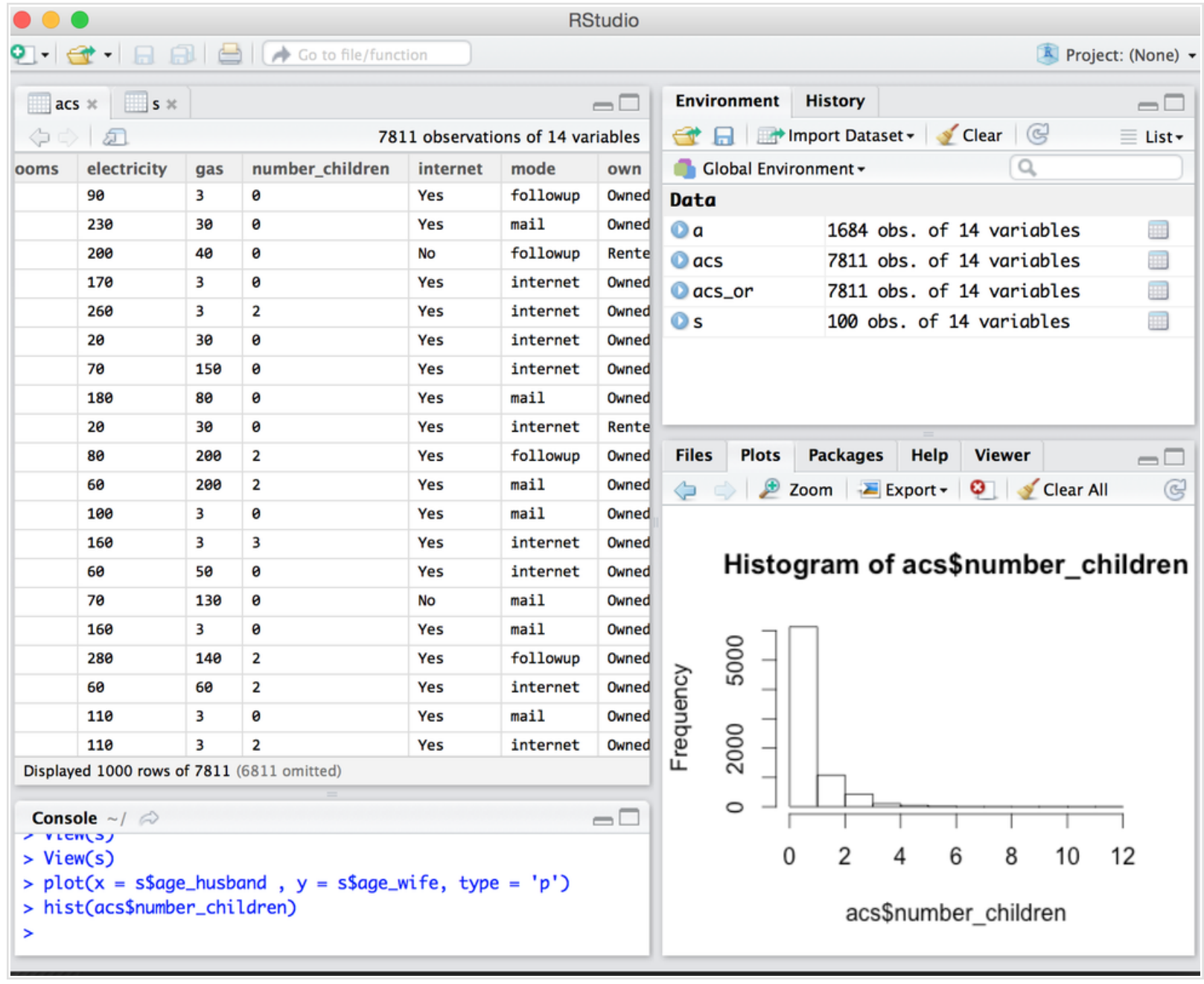
```
plot(x = s$age_husband , y = s$age_wife, type = 'p')
```

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.



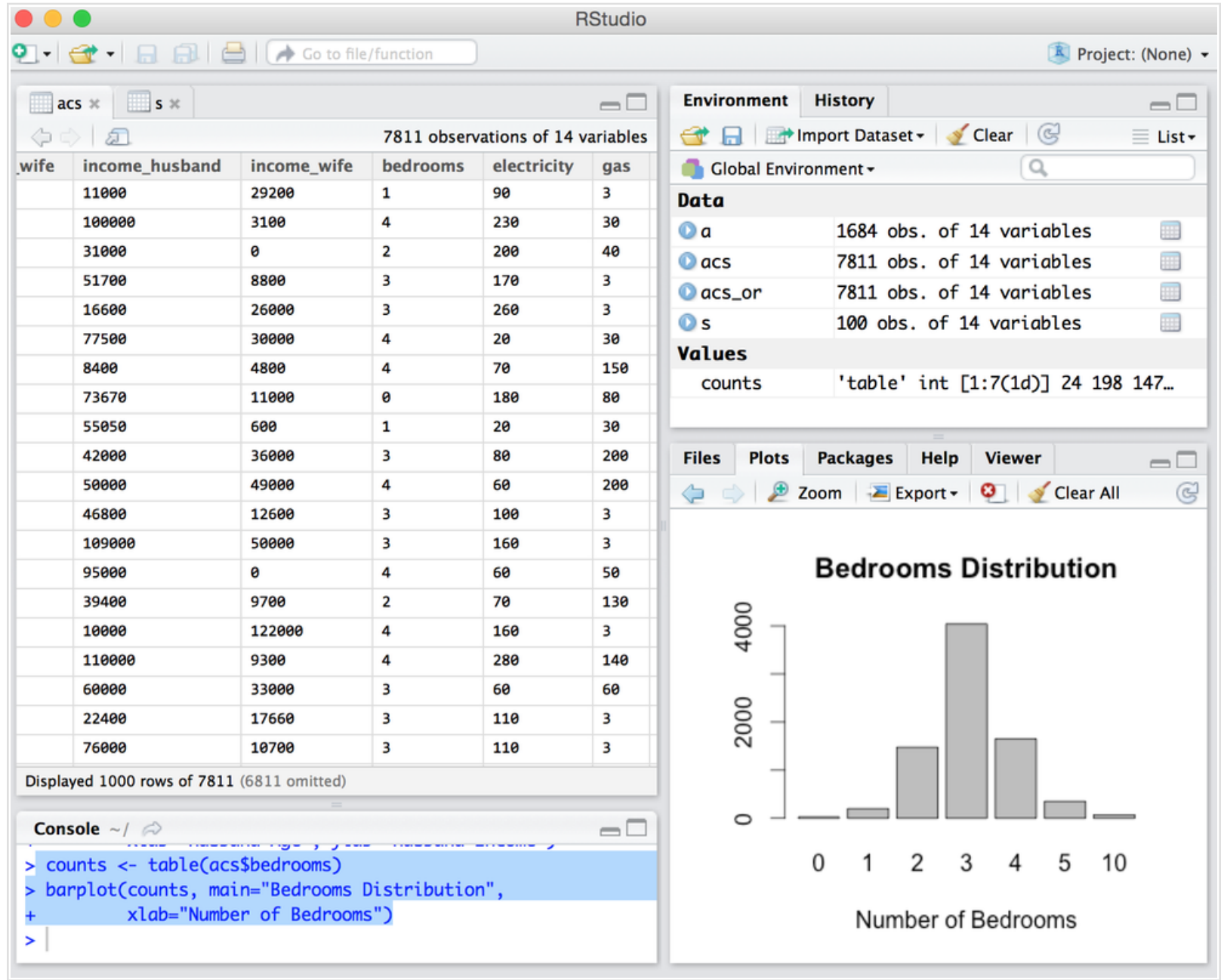
For data distribution plots, there are several features tools and packages available in R that you can use to draw any kind of distribution. For example

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.

Note

For any documentation or usage of the function in R Studio, just type the name of the function and then press *cntrl+space* to get the auto completion window.
You can use *?* before any function name to view the official documentation