R Workflow: Plotting in R

Manas Patnayakuni : BS2017

August 15, 2022

1 Introduction to Plotting

When it comes to interpreting large amounts of data, data visualisation is the key to understanding the distribution of our data. Plotting can help us visualize data in a much more refined form rather than making sense of huge tables in excel sheets. With the help of the plot() function in R, we can take in few parameters and plot X-Y graphs. Previously we discussed how we can download the "faraway" package in R. We also showed that it comes inbuilt with a data set called "pima". Let us try to use the plot() function for this data.

2 The Problem in Pima Dataset

A common obstacle that we face when we collect data is that we don't always get all the required data points for every entry. This mainly occurs in huge data sets where accurate collection of all the data points for thousands of entries becomes very difficult. Let us take an example of the data set "pima" which comes from a study on 768 adult female Pima Indians living near Phoenix conducted by the National Institute of Diabetes and Digestive and Kidney Diseases.

After downloading the package "faraway", we have to open by using the following code. The data set "pima" already comes with the "faraway" package so we don't need to download it again.

```
> library("faraway")
```

Now open pima and check the data points for the variable "diastolic". This contains the values of the diastolic blood pressure (mm Hg) for the 768 females.

```
> pima$diastolic
  [1]  72  66  64  66  40  74  50   0  70  96  92  74  80  60  72   0  84  74
 [40]  72  64  84  92 110  64  66  56  70  66   0  80  50  66  90  66  50  68
 [79]   0  66  44   0  78  65 108  74  72  68  70  68  55  80  78  72  82  72
[118]  48  60  76  76  64  74  80  76  30  70  58  88  84  70  56  64  74  68
[157]  52  56  74  72  90  74  80  64  88  74  66  68  66  90  82  70   0  60
[196]  84  58  62  64  60  80  82  68  70  72  72  76 104  64  84  60  85  95
[235]  68  72  84  90  84  76  64  70  54  50  76  85  68  90  70  86  52  84
[274]  78  70  70  60  64  74  62  70  76  88  86  80  74  84  86  56  72  88
[313]  74  50  80  68  80  74  66  78  60  74  70  90  75  72  64  70  86  70
[352]  84  82  62  78  88  50   0  74  76  64  70 108  78  74  54  72  64  86
[391]  66  76  64  72  78  58  56  66  70  70  64  61  84  78  64  48  72  62
[430]  82   0  74  74  75  68   0  85  75  70  88 104  66  64  70  62  78  72
[469]   0  78  82  70  66  90  64  84  80  76  74  86  70  88  58  82   0  68
[508]  60  50  78  72  62  68  62  54  70  88  86  60  90  70  80   0  70  58
[547]  76  68  82 110  70  68  88  62  64  70  70  76  68  74  76  66  68  60
[586]  56  66  66  86   0  84  78  80  52  72  82  76  24  74  38  88   0  74
[625]  64  88  68  78  80  65  64  78  60  82  62  72  74  76  76  74  86  70
[664]  80  60  80  82  70  58  78  68  58 106 100  82  70  86  60  52  58  56
[703]  88   0  76  80   0  46  78  64  64  78  62  58  74  50  78  72  60  76
[742]  44  58  94  88  84  94  74  70  62  70  78  62  88  78  88  90  72  76
```

We notice that there are some data points of the variable "diastolic" which are collected as 0. This does not make any sense as the diastolic blood pressure can never be zero for a living human being! However, we know the reason that they are given 0 is because the data points for these entries were never taken due to some circumstances.

The problem that arises with this obstacle is that R cannot tell the difference between the value 0 and the fact that this data point was never collected. It will simply assume that 0 is a valid diastolic pressure. This makes the data invalid and we won't be able to perform any operations to analyse it.

3 The Solution

To fix this problem,we just have to replace every 0 with NA which stands for "Not Available". This way R will consider those NAs as missing values and hence not affecting the data analysis.

We first create a variable called "zeros" which will store the Boolean values of the condition "diastolic=0".

```
> zeros=(pima$diastolic==0)
```

This means that "zeros" will contain values: TRUE or FALSE depending on whether the value of diastolic pressure is 0 or not respectively. We can check this by printing the data points of variable "diastolic" which give the value TRUE for the above Boolean condition.

```
 pima$diastolic[zeros]
[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

The above code basically printed the 0s of the variable "diastolic". Now all we need to do is equate these 0s to NA which can be done by the following code.

```
> pima$diastolic[zeros]=NA
```

To check whether all the 0s have been switched or not, we need to check the variable diastolic once again.

```
> pima$diastolic
  [1]  72  66  64  66  40  74  50  NA  70  96  92  74  80  60  72  NA  84  74
 [19]  30  70  88  84  90  80  94  70  76  66  82  92  75  76  58  92  78  60
 [37]  76  76  68  72  64  84  92 110  64  66  56  70  66  NA  80  50  66  90
 [55]  66  50  68  88  82  64  NA  72  62  58  66  74  88  92  66  85  66  64
 [73]  90  86  75  48  78  72  NA  66  44  NA  78  65 108  74  72  68  70  68
 [91]  55  80  78  72  82  72  62  48  50  90  72  60  96  72  65  56 122  58
[109]  58  85  72  62  76  62  54  92  74  48  60  76  76  64  74  80  76  30
[127]  70  58  88  84  70  56  64  74  68  60  70  60  80  72  78  82  52  66
[145]  62  75  80  64  78  70  74  65  86  82  78  88  52  56  74  72  90  74
[163]  80  64  88  74  66  68  66  90  82  70  NA  60  64  72  78 110  78  82
[181]  80  64  74  60  74  68  68  98  76  80  62  70  66  NA  55  84  58  62
[199]  64  60  80  82  68  70  72  72  76 104  64  84  60  85  95  65  82  70
[217]  62  68  74  66  60  90  NA  60  66  78  76  52  70  80  86  80  80  68
[235]  68  72  84  90  84  76  64  70  54  50  76  85  68  90  70  86  52  84
[253]  80  68  62  64  56  68  50  76  68  NA  70  80  62  74  NA  64  52  NA
[271]  86  62  78  78  70  70  60  64  74  62  70  76  88  86  80  74  84  86
[289]  56  72  88  62  78  48  50  62  70  84  78  72  NA  58  82  98  76  76
[307]  68  68  68  68  66  70  74  50  80  68  80  74  66  78  60  74  70  90
[325]  75  72  64  70  86  70  72  58  NA  80  60  76  NA  76  78  84  70  74
[343]  68  86  72  88  46  NA  62  80  80  84  82  62  78  88  50  NA  74  76
```

As we can see all the 0s have changed to NA. This means R will consider those data points as missing and avoid including them in the data analysis.

4 Plots of Pima Dataset

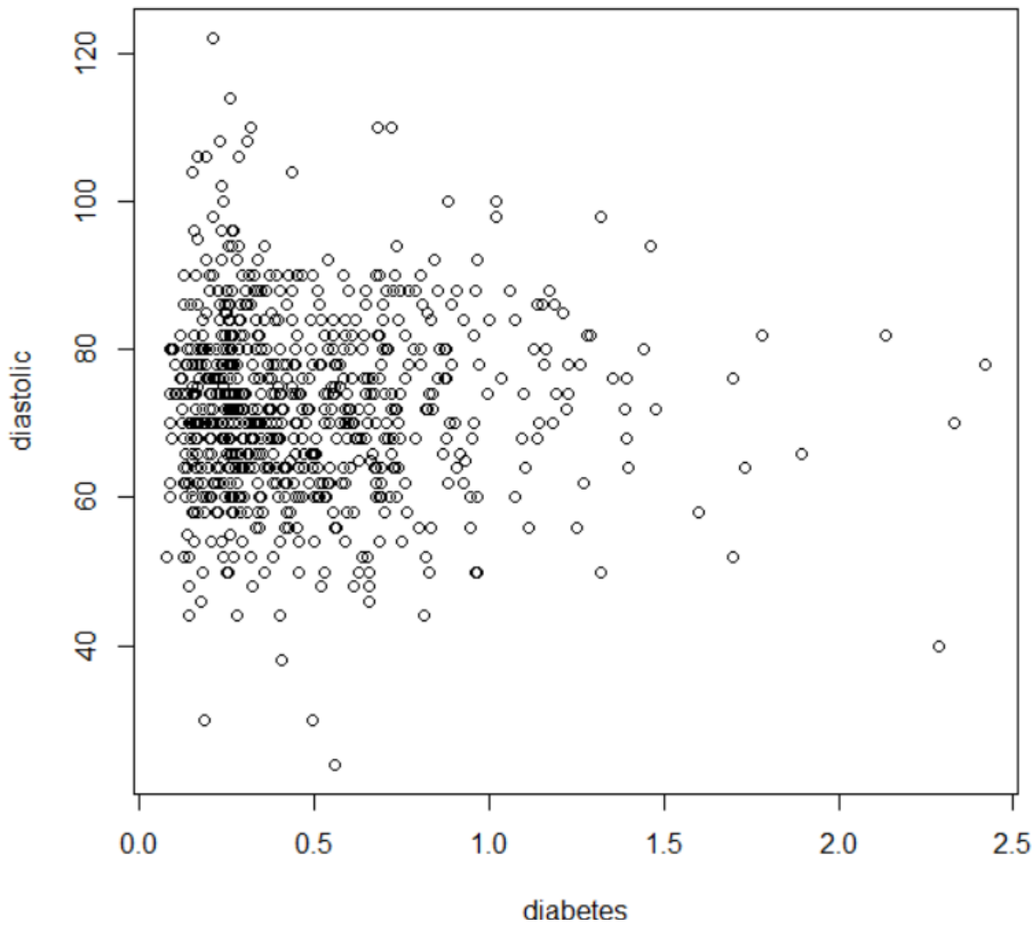
Since we have made the necessary changes to our data set, now we can plot our data and conduct the necessary analysis. Let us plot the variables "diastolic" vs "diabetes". The code to be used is as follows:

```
> plot(diastolic~diabetes,pima)
```

We can also simply write the code as:

```
> plot(pima$diastolic~pima$diabetes)
```

The resultant plot is:



We can also try a box-plot of "diastolic". The code for it is as follows:

```
boxplot(pima$diastolic,main="Boxplot of Diastolic Pressure",ylim=c(0,130)
```