Graphs and Summary Statistics: Sections 7-9

How to Create Basic Graphs: Barplots and Pie Charts

How to Create a Histogram and Calculate Summary Statistics

How to Do Pairs of Graphs: Two Histograms or Two Boxplots on One Graph

Section 7: How to Create Basic Graphs -- Barplots and Pie Charts

(Uses data file: Hospitals.txt)

First get the data into R by the method described in the section called: "How to Get Data into R." Then you can use the data to have R create graphs.

The first one is a barplot that shows the bed counts by hospital.
The second one is a barplot that shows the percentages of the total beds by hospital.
The third one is a pie chart that shows relative capacity by hospital.

The parts of the commands are:

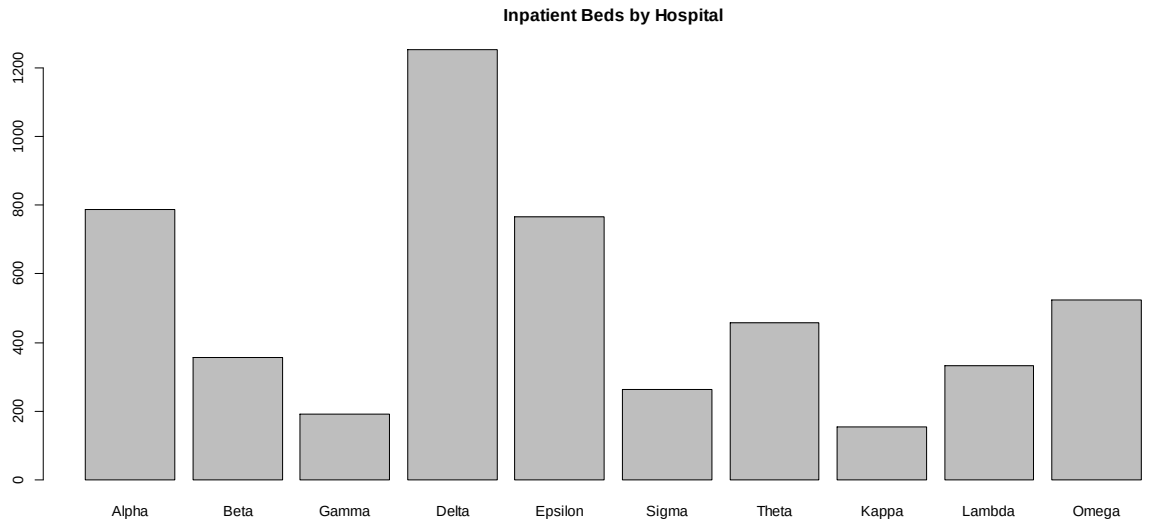
1. The type of graph you want.

2. The first item inside the parentheses tells R to use the column No.Beds for the frequencies or percentages. Note that the second version of this command changes from counts to percentages by dividing by the sum of the beds in the table. The change is reflected in the second graph.

3. The second item inside the parentheses of the barplot command says "names.arg = Hospital"; this tells R to label the horizontal axis with the names in the column called Hospital from the data set. In the pie chart, the "names.arg" part of the command is replaced by " labels = Hospital."

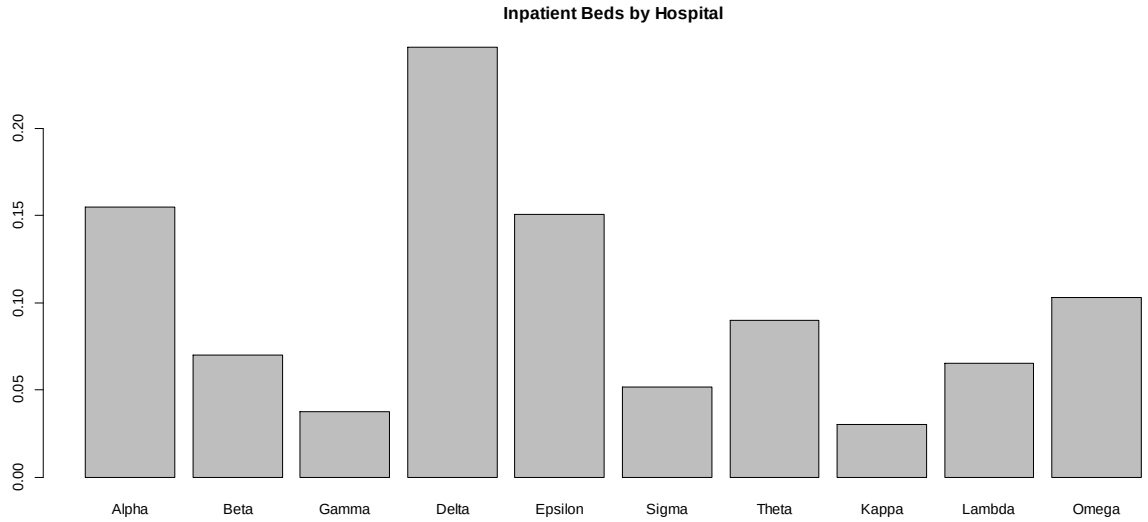
4. The last item in the command, where it says
   main = "something in quotes"
   tells R the heading that you want for the whole graph.

```
> barplot (No.Beds, names.arg = Hospital, main = "Inpatient Beds by Hospital")
```



If you want percentages instead of counts of numbers of beds, you change the first item in parentheses.

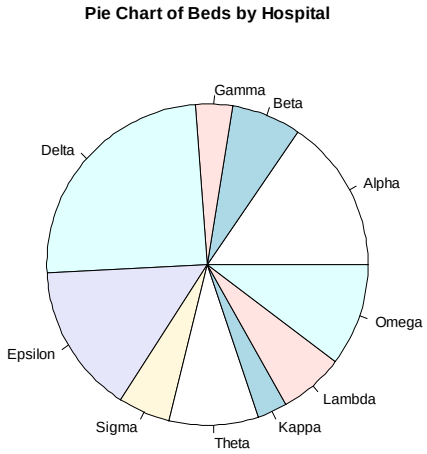
```
> barplot (No.Beds/sum(No.Beds), names.arg = Hospital, main = "Inpatient Beds by Hospital")
```



Note that this graph looks like the previous one, except that the scale on the vertical axis is no longer in terms of bed count. Now it is percentage of the total.

Finally, suppose you want a pie chart based on the number of beds. The command is similar to the "barplot" command, but the portion that said "names.arg" is replaced simply by "labels."

```
> pie (NoBeds, labels = Hospital, main = "Pie Chart of Beds by Hospital")
```



Section 8: How to Create a Histogram and Calculate Summary Statistics

(Uses data file: Hospitals.txt)

Read in the data file, attach it and then display it.

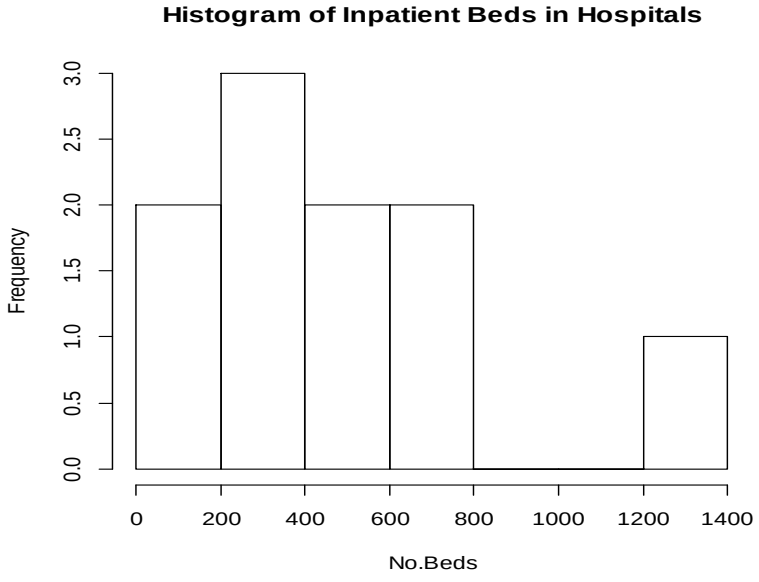
```
> Data = read.table ("E:/Data Files/Hospitals.txt", header = TRUE)
> attach (Data)
> Data
```

The file is as follows.

```
   Hospital  No.Beds
1    Alpha     787
2    Beta      356
3    Gamma     190
4    Delta    1252
5    Epsilon   767
6    Sigma     264
7    Theta     457
8    Kappa     154
9    Lambda    333
10   Omega     525
```

You can create a histogram using the column called No.Beds as input data. R will automatically group the data and count the frequencies. The second item inside the parentheses give the graph a title.

```
> hist (No.Beds, main= "Histogram of Inpatient Beds in Hospitals")
```



The next series of commands calculates the mean, median, deviations, variance, standard deviation and range of the data column called No.Beds. Most of these only require single lines of R code. Whenever two lines are needed, the first line defines the function and the second displays the result. First find the mean number of beds.

```
> mean (No.Beds)
```

The output is:

```
 508.5
```

Next find the median number of beds.

```
> median (No.Beds)
```

The output is:

```
406.5
```

Find the deviations. These are defined to be the numbers of beds minus the mean.

```
> devs = No.Beds – mean (No.Beds)
> devs
```

The output is:

```
  278.5   -152.5   -318.5    743.5    258.5   -244.5    -51.5   -354.5   -175.5    16.5
```

Next find the variance.

```
> var (No.Beds)
```

The output is:

```
115672.3
```

Then find the standard deviation of the sample.

```
 sd (No.Beds)
```

The output is:

```
340.1063
```

Finally, if you want the range, it is defined to be the maximum minus the minimum.

```
> range = max (No.Beds) – min (No.Beds)
> range
```

The output is:

```
 1098
```

Section 9: How to Do Pairs of Graphs
Two Histograms or Two Boxplots on One Graph
(Uses data file: Solar Eclipses.txt)

Suppose you have two numeric data sets that you want to compare graphically. The most common comparisons are: (1) comparing their histograms, or (2) comparing their boxplots. Obviously, you could do one graph at a time and then look at the results. However, it may be easier to see the comparisons if you put both histograms on one graph, or both boxplots side-by-side. This section shows you how to do that.

The data set is: Solar Eclipses.txt. It contains the duration (in seconds) of a sample of annular and total solar eclipses, where duration is the length of time the shadow of the moon is completely in front of the sun.

As usual, read in the data set and attach it. Then you can display it if you wish.

```
> Data = read.table ("E:/Data Files/Solar Eclipses.txt", header = TRUE)
> attach (Data)
> Data
```

A portion of the data set appears as follows.

```
    Annular  Total
1       661    389
2       113    132
3       487    380
:
:
60       33    226
61      241    153
67      701    249
68      373    143
```

To put two histograms on the same graph, you first want to see which data set has a greater maximum value. That is because you will want to list that one first when you create the graphs, so that the scaling of the horizontal axis works out nicely.

```
> max (Annular); max (Total)
```

The output from this command is:

```
729
428
```

You can see that the annular eclipses in the data set have a larger maximum. Therefore, you will create the histogram of annular eclipse durations first. The first command does this. The second command overlays the histogram of total eclipse durations by “adding” it to the first graph.

A few comments are needed to explain these commands in the general case. The first command has the following syntax:

```
> hist (First variable, main = ”Title for Final Graph”, xlab = ”Labels for x-axis”)
```

NOTE: in xlab, plan ahead to specify that the frequencies of first variable you are graphing will show up as unshaded bars on the histogram, and the second will show up as shaded. A reader would not know that, so you have to explain it in the labeling. To see what this means, continue below to look at the example.

The second command is then:

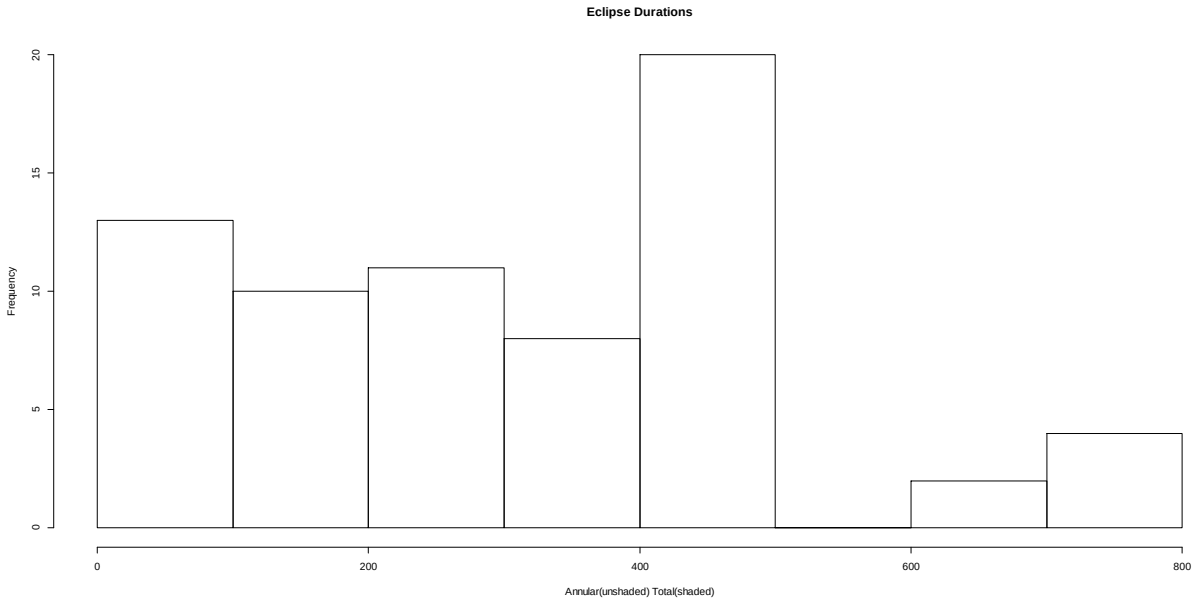
```
> hist (Second variable, density = 20, add = TRUE)
```

NOTE: The density controls the shading; you can experiment with larger or smaller values. Larger values make the shading denser. The instruction “add = TRUE” tells R to put the second histogram on the same graph as the first.

So for this example, you will proceed as follows.

```
> hist (Annular, main = "Eclipse Durations", xlab = "Annular(unshaded) Total(shaded)")
```

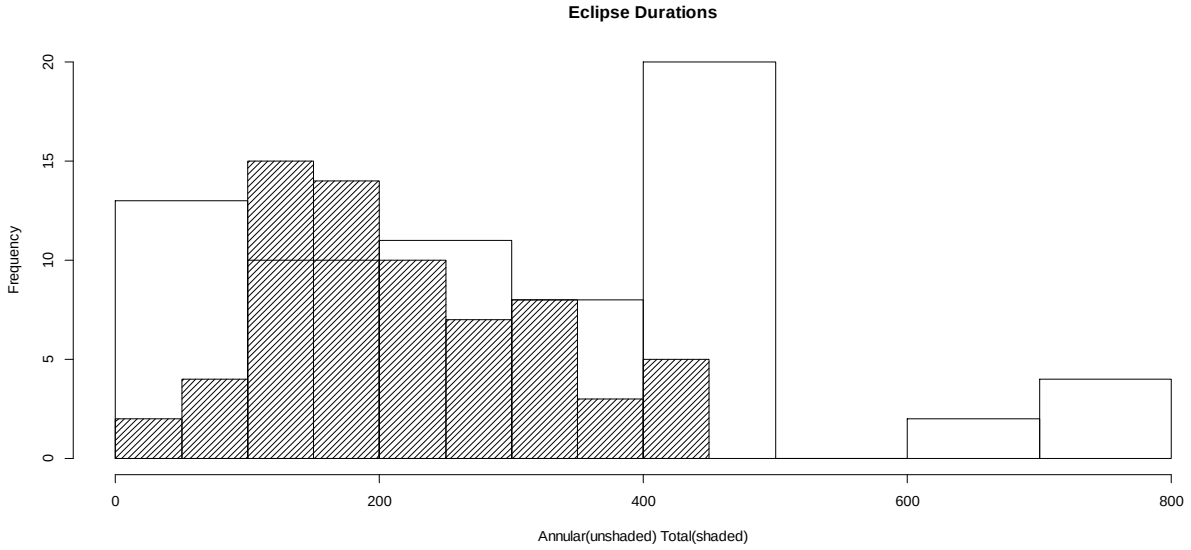
At this point, the histogram looks like this, showing only the annual eclipse durations. That is all the you have told R to produce so far, even though you gave it labelling for both types of eclipses. The labelling is really <u>just</u> a label, and does not get involved in producing the intervals used for the graph.



Now you overlay the histogram for the total eclipses.

```
> hist (Total, density = 20, add = TRUE)
```

Here is the resulting graph.



Getting two boxplots side by side is simpler; it only requires one command. You use the "boxplot" command, specify both variables in the order that you want them, and use the "names" subcommand to give R a list of the labels to use for each.

```
> boxplot (Annular, Total, names = c ("Annular Durations", "Total Durations") )
```

Here is graph that results. Now you can visually compare the two boxplots and see, for instance, that the median total duration is less than the median annular duration, but the first quartiles are almost the same.