Basics of Plotting Data

Luke Chang

Last Revised July 16, 2010

One of the strengths of R over other statistical analysis packages is its ability to easily render high quality graphs. R uses vector based output (similar to illustrator), which produces very clean publication quality graphs. In addition, the graphs are fully customizeable, which gives R very powerful graphing capabilities (see http://addictedtor.free.fr/graphiques/ for some examples). This section will cover how to create and save simple graphs (e.g., histogram, scatterplot, barplot, and boxplots) as well as outline some of the basic user options to customize the graphs.

1 The 4 basic types of graphs

For this section will be using a subset of the 2009 sample adult health questionnaire dataset from the Centers for Disease Control.

The first step is to load the data.

```
> data<-read.table(paste(website,"SAMADULT_MH.csv",sep=""),sep=",",
    header=TRUE)
```

1.1 Histograms

Histograms provide a useful way to examine the distribution of the data. We use the `hist` function to plot the distribution of weight in America.

```
> hist(data$AWEIGHTP)
```

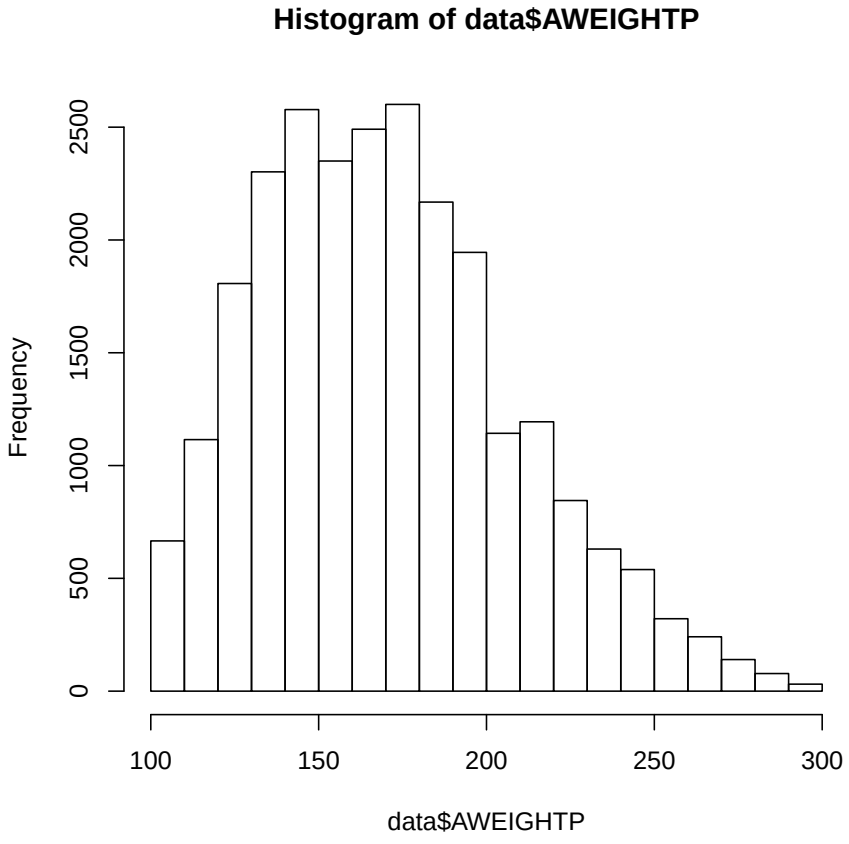

Figure 1: Histogram of the weight distribution of Americans

As you can see creating a histogram of a variable is very simple. Later in this section we will discuss how to customize the graph by labeling the axes, changing the colors, and adding a title.

1.2 Scatterplots

Often it is useful to examine the relationship between two variables using a scatterplot. We can use the generic `plot` function in `R` to examine the relationship between weight and body mass. There are two different ways to use the function. The first way is to specify the `x` and `y` variables in the graph.

```
> plot(data$AWEIGHTP,data$BMI)
```

The second way is to specify the linear equation `Y = X`.

```
> plot(BMI~AWEIGHTP,data=data)
```

Both methods produce the same plot.

1.3 Barplots

Barplots are a useful way to display statistical summaries of several different variables at once. This can be done using the `barplot` function in `R` . Using this method requires first summarizing your data (e.g., mean, median, or mode) and then using the summary values as input into `barplot`.

1.3.1 Plotting Several Variables

To plot the average of several different variables, one simply has to take the mean of the variables. For example, to plot the average response to the mental health questions, we first need to calculate the mean of the data. To calculate the mean it is important to tell `R` to remove the missing values with `na.rm=TRUE`. We can then assign these values to a new variable.

```
> mh.data<-mean(data[10:14],na.rm=TRUE)
```

These questions are on a 5 point ordinal scale with 5 being "I have never experienced it". To make the graphs more readable, it might help to reverse code the data by subtracting it from 5.

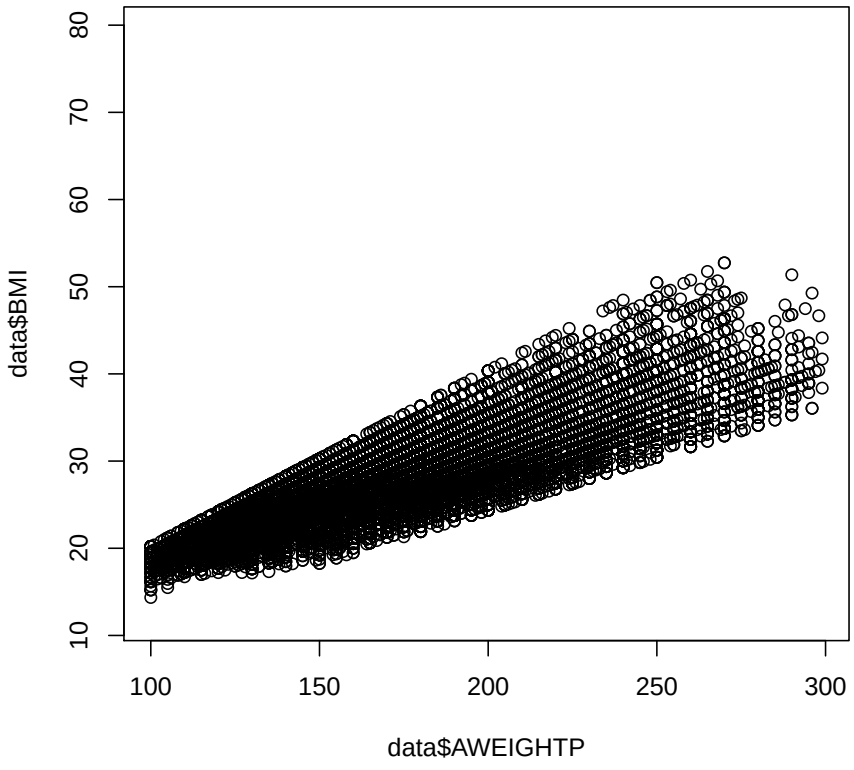

Figure 2: Scatterplot of the relationship between weight and BMI

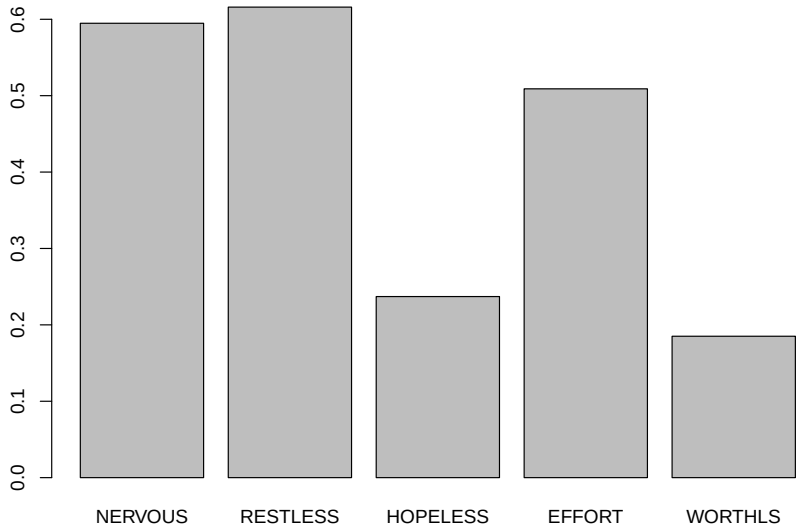

Figure 3: Barplot depicting the average response to the mental health questions

```
> mh.data<-5-mh.data
```

We then plot the summarized data using the `barplot` function.

```
> barplot(mh.data)
```

1.3.2 Plotting Levels of a Factor

Sometimes you may be interested in creating barplots for each level of a factor when your data is in the long format. For example, to plot the average weight split by sex, we have to calculate the mean weight for each level of sex. To do this we can use the `subset` function to select the data for each sex. For example, to select the males we tell `R` to grab the height of all the rows containing males using the logical operation `data$SEX==1` (i.e., `subset(data$AHEIGHT,data$SEX==1)`). We can then take the average of these rows making sure that we exclude the missing values (i.e., `mean(data,na.rm=TRUE)`). Finally, we can combine the mean of the average height of males and females into a single array using the `c()` function. When plotting this

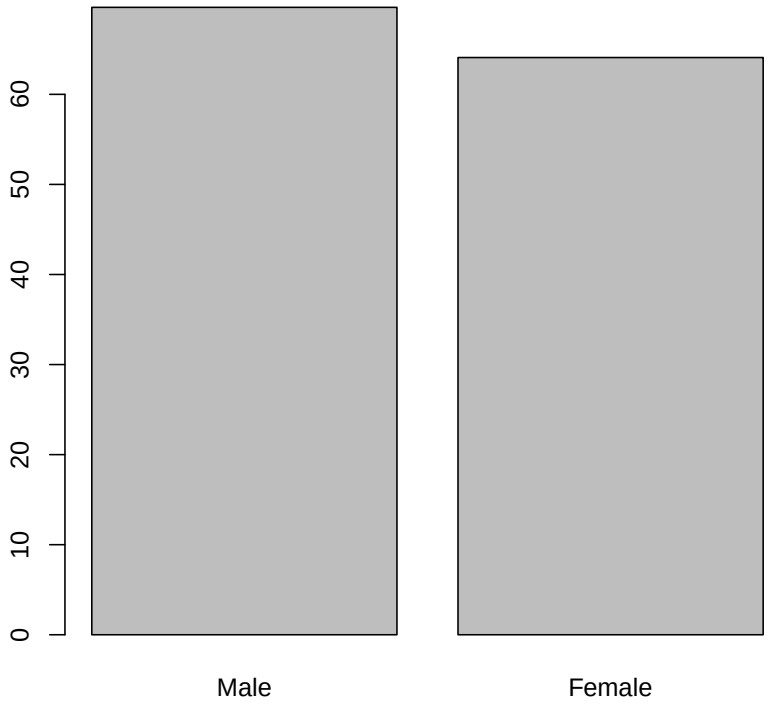

Figure 4: Barplot depicting the average weight of males and females

data with the `barplot` function, we will also need to label the categories with the `names.arg=c("Male","Female")` specifier.

```
> barplot(c(mean(subset(data$AHEIGHT,data$SEX==1),na.rm=TRUE),
    mean(subset(data$AHEIGHT,data$SEX==0),na.rm=TRUE)),
    names.arg=c("Male","Female"))
```

Unfortunately, for whatever reason it is not straightforward to easily add error bars to these barplots. However, there are many alternative methods to create barplots with errorbars using the `barplot2` function from `gplots` package or `ggplot2`, which will be discussed in more detail in the advanced graphics section.

1.4 Boxplots

Another useful way to graphically summarize several different variables is the `boxplot` function. Boxplots depict the median, the upper 75 and lower 25 quartiles and any outliers. They are useful for displaying the distribution of a variable. Similar to barplots, boxplots can plot several different variables, or different levels of a factor.

1.4.1 Plotting Several Variables

Here we will illustrate how to plot the same data as the barplot example using the responses to the mental health questions. Don't forget that we are reverse coding the responses.

```
> boxplot(5-data[10:14])
```

As you can see, most respondents reported never experiencing any symptoms of mental health problems in the past 30 days. The graph reveals that the median is 0 and the data is highly positively skewed.

1.4.2 Plotting Levels of a Factor

Graphically examining the distribution of a variable at different levels of a factor can easily be accomplished using the `boxplot` function. We simply need to indicate the Y variable and the X factor, here being height and sex respectively.

```
> boxplot(AHEIGHT~SEX,data=data,names=c("Female","Male"))
```

1.5 Saving Graphs

By default `R` will print the graphics to your screen. To print them to a file, you must first tell `R` to direct the graphic output to a device. There are several different devices including: pdf and png. It is easy to create publication quality pdfs of your figure using the `pdf()` command. Alternatively, you could also write to other devices such as `png()`.

```
> pdf("FileName.pdf")
> dev.off()
```

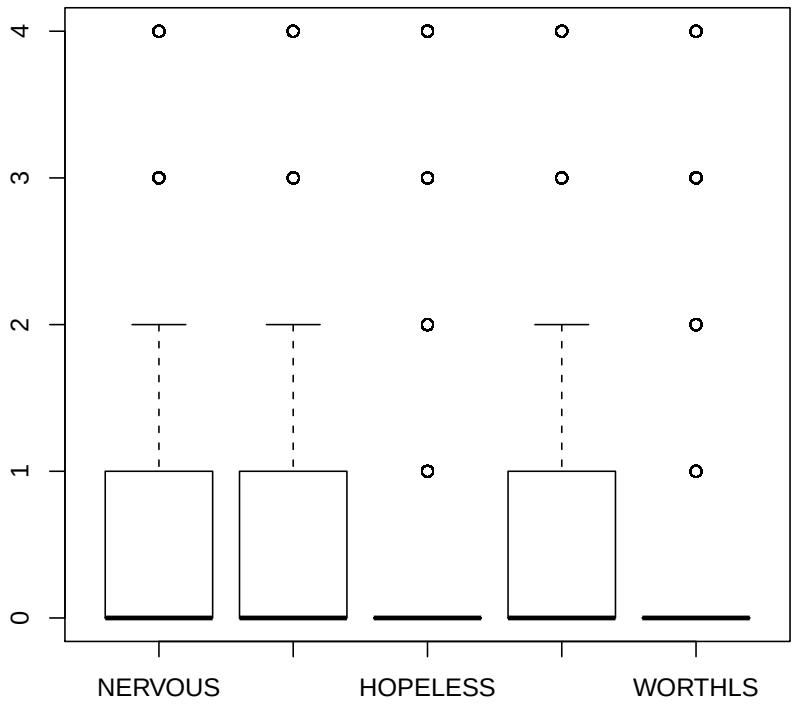

Figure 5: Boxplot depicting the distribution of responses on the mental health questions

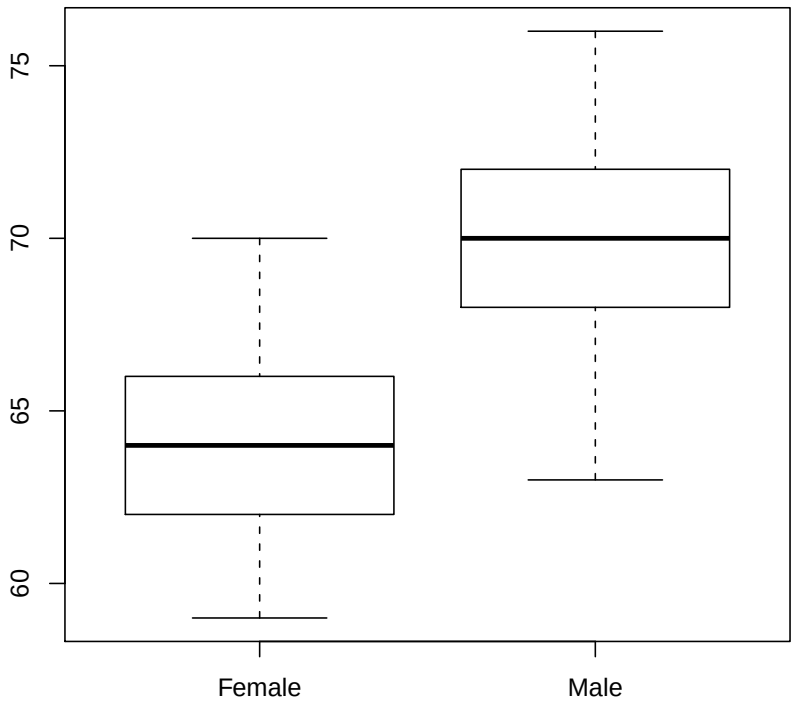

Figure 6: Boxplot depicting the different distributions of the weight of males and females

```
pdf
  2
```

After you are done creating your plot, it is necessary to close the device using the `dev.off()` command.

For example, to save the scatterplot we created earlier:

```
> pdf("Fig4_WeightXBMI.pdf")
> plot(BMI ~AWEIGHTP,data=data)
> dev.off()
```

```
pdf
  2
```

1.6 Customizing Graphs

Graphs can be customized with relative ease in `R` using the user options. There are a number of different points, colors, and sizes that can be used in your graphs.

1.6.1 Points

This example shows all 25 symbols that you can use to produce points in your graphs (`pch=?`). It is easy to vary the size using the `cex=?` specifier. It is also easy to vary the color using the `col=?` specifier.

1.6.2 Customizing the Scatterplot

In this section will demonstrate how to customize the appearance of the scatterplot that we created earlier. We will change the points to filled in circles (`pch=16`) and make them transparent to better reflect the density of the data (`col=rgb(0, 0, 0, .007)`). We will also make the points slightly bigger (`cex=4`). We will plot the sex by weight interaction by adding separate regression lines for males and females. We will split the data file into two new files by sex and then calculate a linear regression to find the intercept and slope parameters for each sex. These parameters can then be added on to the graph using the `abline` function. Line types can be manipulated with the `lty=?` specifer. We will also add a legend to indicate the color of each sex.

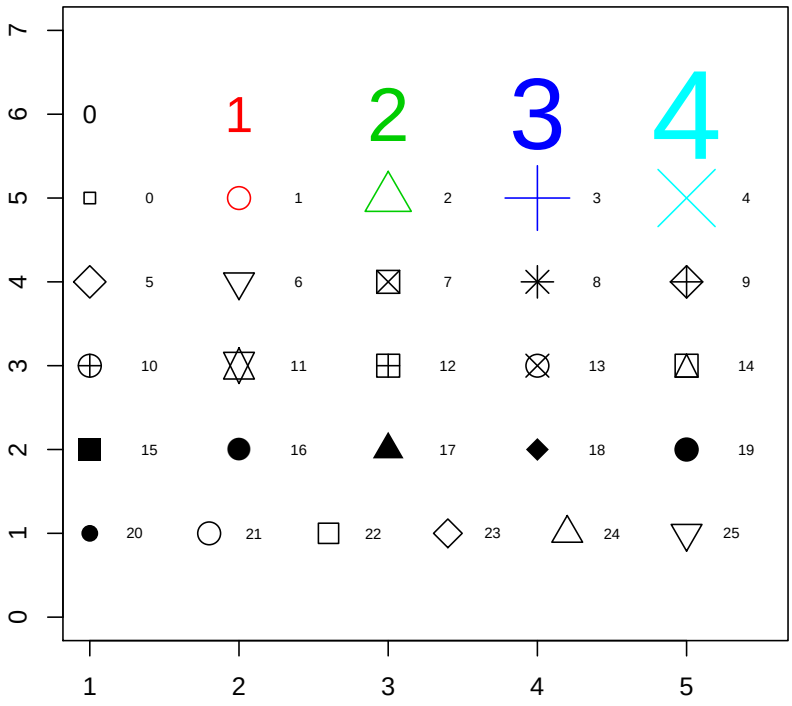

Figure 7: Variations of point styles

Finally, It is easy to relabel the axes (`xlab="Weight (lbs)"`) and the size of the text (`cex.lab=1.5`) as well as add a title (`main="Interaction Plot"`) .

```
> plot(BMI~AWEIGHTP, data=data,col=rgb(0, 0, 0, .007), pch=16,cex=4,
    ylab="Body Mass Index (BMI)",xlab="Weight (lbs)",cex.lab=1.25,
    main="Interaction between sex and weight in predicting BMI")
> m<-subset(data,data$SEX==1)
> f<-subset(data,data$SEX==0)
> mLine<-lm(BMI~AWEIGHTP,data=m)
> fLine<-lm(BMI~AWEIGHTP,data=f)
> abline(mLine,col="blue",lwd=6,lty=2)
> abline(fLine,col="red",lwd=6,lty=3)
> legend("topright", c("Female","Male"), pch=15, col=c("red","blue"),
    title="Sex", inset = .02)
```

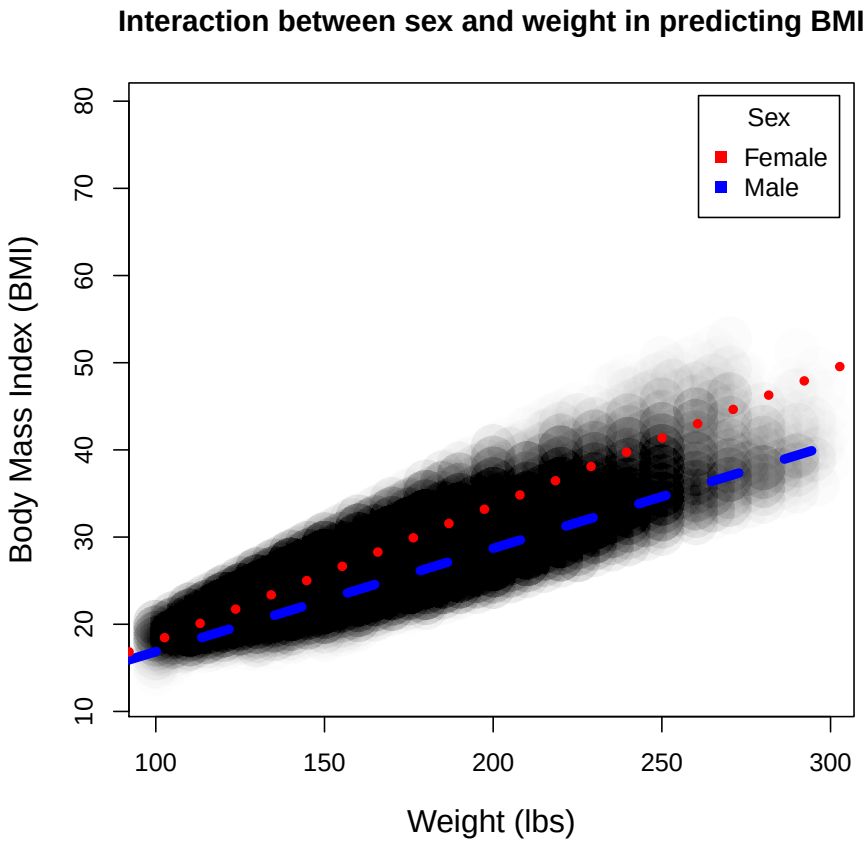

Figure 8: Scatterplot depicting degree of association between weight and BMI