Basic Descriptive Statistics in R

MTHS 3020, Fall 2013

1 Getting an Excel file into R

Suppose we have a turning operation in a machine shop where we are turning pins to a diameter of $12.5 \pm .5$ mm. We have three different machines making the same part and throughout the course of a day we take six samples of pins from each machine to obtain the following diameter data:

Machine		
1	2	3
12.5	11.8	12.3
12.7	12.2	12.5
12.5	12.0	12.5
12.6	12.4	12.4
12.8	11.9	12.6
12.6	12.0	12.5

This dataset is contained in the spreadsheet `MachineData.xlsx`. I've found that it's easier to use Excel to save the data in a different format prior to reading them into `R`. For example, save them in a CSV or tab-delimited text file. Suppose we've saved it as the latter. Then, after changing the `R` directory to where the dataset is stored (File → Change dir...), I can read in the .txt file as follows:

```
dat= read.table("MachineData.txt", header= T)
```

Note that, since the text file has column headers, I'm telling `R` to read the first line of the file as column *names* and not part of the data set itself. The dataset is being stored in an object called `dat`. The data in each column can be accessed individually using the $ operator:

```
> dat$Machine
 [1] 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3
> dat$Diameter
 [1] 12.5 12.7 12.5 12.6 12.8 12.6 11.8 12.2 12.0 12.4 11.9 12.0 12.3 12.5 12.5
[16] 12.4 12.6 12.5
```

2 Basic Summary Statistics

Since the first column is just a identifying each machine, lets focus on the second column, the measured diameters. We can find the mean, median, etc. in `R` as follows:

```
> diams= dat$Diameter
> mean(diams)
[1] 12.37778
> median(diams)
[1] 12.5
> summary(diams)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  11.80   12.22   12.50   12.38   12.58   12.80
```

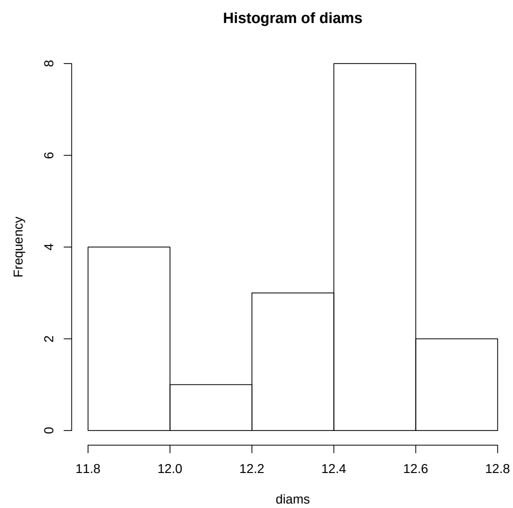
```
> var(diams)
[1] 0.08183007
> sd(diams)
[1] 0.2860595
```

3 Basic Plots

3.1 Histograms

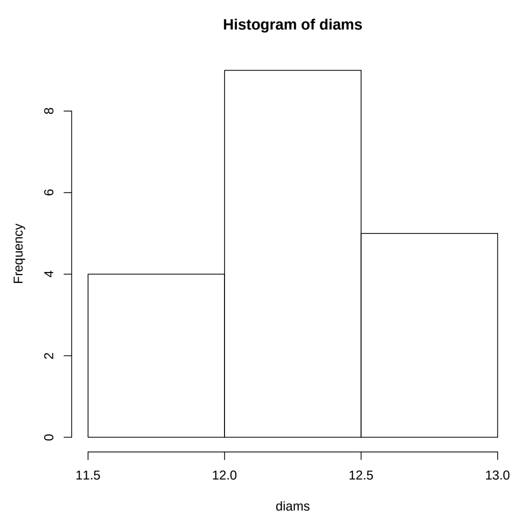
A histogram of the diameter values:

```
> hist(diams)
```

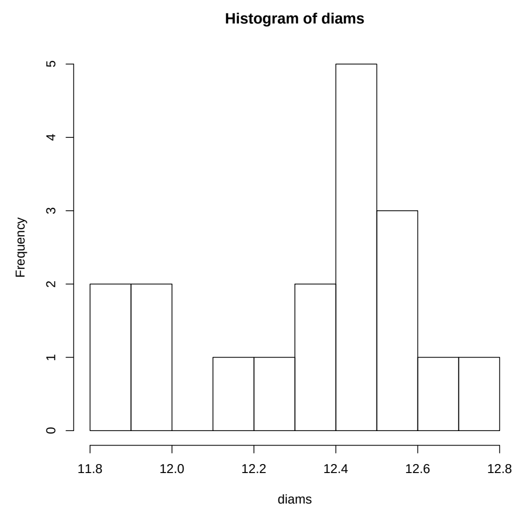


The number of bins can be adjusted using the "breaks" argument inside the function. Note that R only uses this as a 'suggested' number, and will try to make it close to that:

```
> hist(diams,breaks= 2)
```



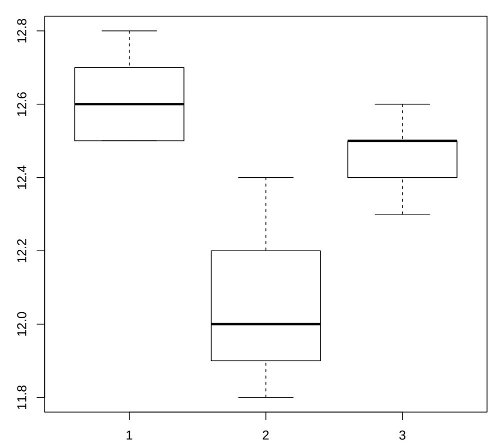
```
> hist(diams,breaks= 10)
```



3.2 Boxplots

Boxplots are especially useful for comparing groups of observations. For example, with the machine data, we may be interested in comparing the diameter observations between machines:

```
> boxplot(Diameter ~ Machine, data= dat)
```

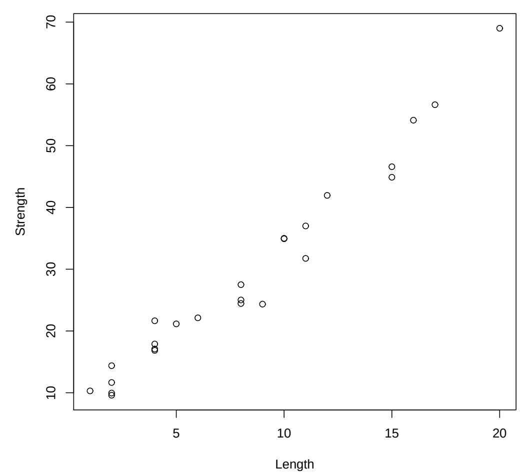


The argument in the function call is of the form <data> ~ <grouping variable>. Note that by identifying the dataset object with the `data=` argument, I can specify each variable name directly.

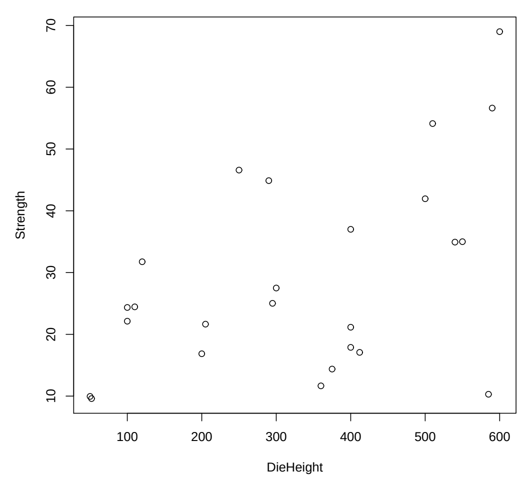
4 Scatterplots

In addition to plotting/obtatining information about a single variable, we are quite often interested in the relationship between two or more variables. The most basic tool for exploring such relationships is the scatterplot. To illustrate, consider a dataset on three variables that were collected in an observational study in a semiconductor manufacturing plant. (A completed semiconductor is wire bonded to a frame.) The variables in the dataset include pull strength (force required to break the bond), wire length, and die height. This dataset is contained in `WireBondData.xlsx`; I've again converted it to a different format prior to reading it into `R`:

```
> wireDat= read.table("WireBondData.txt", header= T)
> plot(Strength ~ Length, data= wireDat)
```

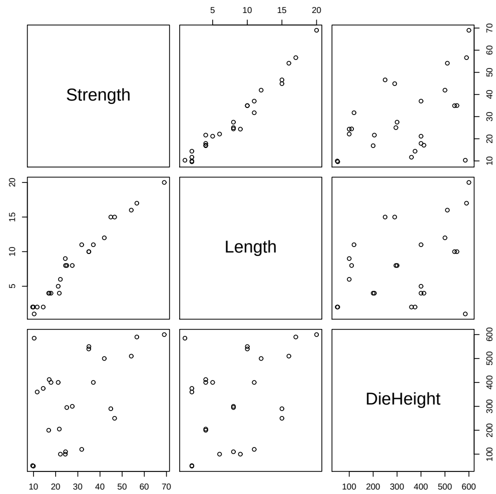



```
> plot(Strength ~ DieHeight, data= wireDat)
```



We can also create a matrix of scatter plots to see the relationships between all of the variables at the same time:

```
> plot(wireDat)
```



We can calculate the correlation between strength and the other variables using the `cor` function:

```
> cor(wireDat$Strength,wireDat$Length)
[1] 0.9818118
> cor(wireDat$Strength,wireDat$DieHeight)
[1] 0.4928666
```