Simple-data-analysis_plots-and-SLR

Basic Data Analysis, Plots and Simple Linear Regression

In these notes we will do a basic data analysis.

We will plot some data and then use simple linear regression to look for a linear relationship.

Reading in the Data

```
# data is read into a data.frame
hd = read.csv("http://www.rob-mcculloch.org/data/midcity.csv")
dim(hd) # number of rows and number of columns
```

```
## [1] 128   8
```

```
names(hd) # variable names
```

```
## [1] "Home"      "Nbhd"      "Offers"    "SqFt"      "Brick"     "Bedrooms"
## [7] "Bathrooms" "Price"
```

Each observation (row) corresponds to a house. We have data on 128 houses.

Each column corresponds to a variable, something different we have measured about each house.

Our goal is to relate the price of a house (the dependent variable) to characteristics of the house.

Price and Size

As a simple first pass, let's just relate the price of house to its size. We'll make a data.frame with just these two variables.

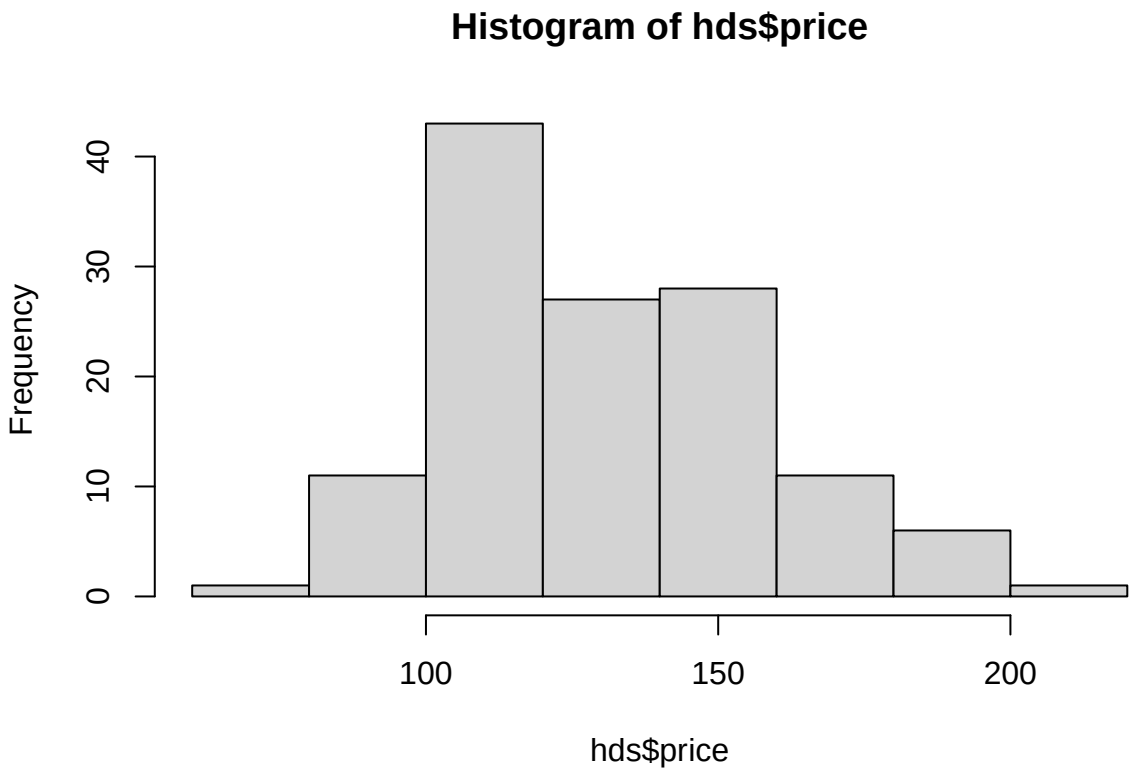
```
hds = data.frame(price = hd$Price, size = hd$SqFt)
# lets rescale the data so that the units are thousands of dollars and thousands of square feet
hds$price = hds$price/1000
hds$size = hds$size/1000
summary(hds)
```

```
##      price            size
##  Min.   : 69.1   Min.   :1.450
##  1st Qu.:111.3   1st Qu.:1.880
##  Median :126.0   Median :2.000
##  Mean   :130.4   Mean   :2.001
##  3rd Qu.:148.2   3rd Qu.:2.140
##  Max.   :211.2   Max.   :2.590
```

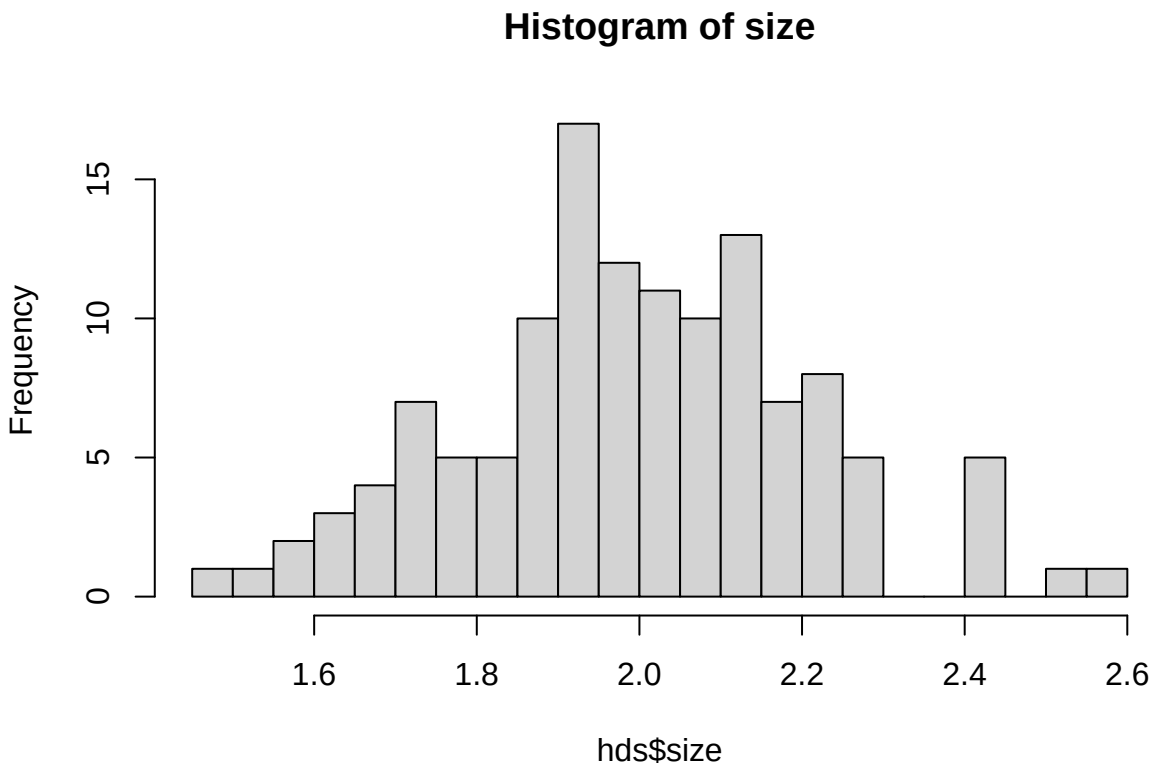
Histogram and Scatterplot

We can look at our data using the histogram and scatterplot.

```
hist(hds$price)
```



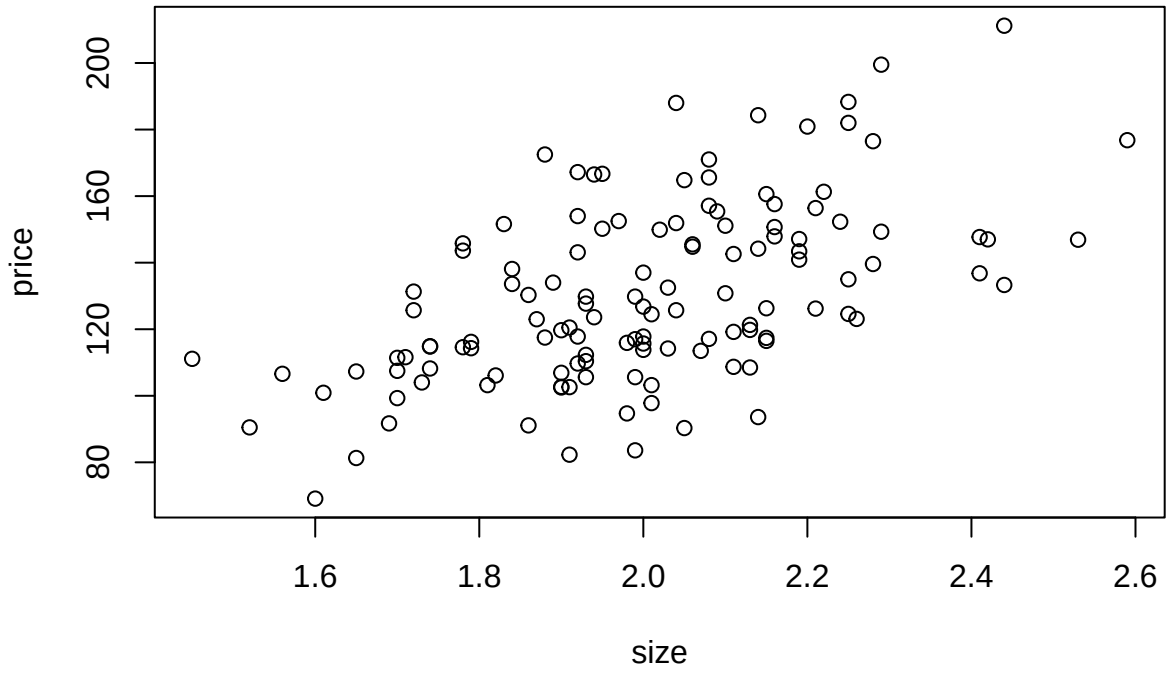

```
hist(hds$size, breaks=20, main="Histogram of size")
```



```
#breaks will choose allow us to choose the number of bins.
```

Now let's plot `size` vs. `price` to see the relationship.

```
plot(hds$size,hds$price,xlab="size",ylab="price")
```



Simple Linear Regression

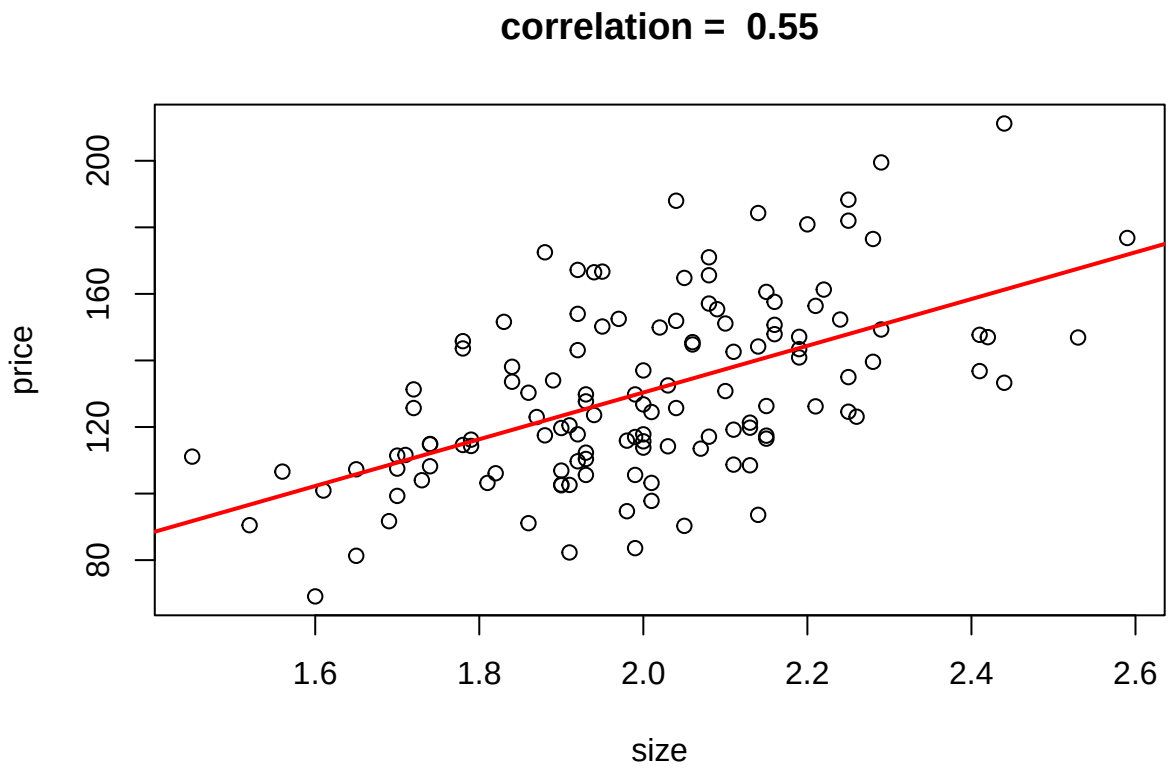
Definitely a relationship, and it looks linear.
Let's run the linear regression of `price` on `size`.

```
# regess price on size, pulling the variables from the data.frame hds.
hdreg = lm(price~size,hds)
summary(hdreg) # standard regression ouput
```

```
##
## Call:
## lm(formula = price ~ size, data = hds)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -46.59 -16.64  -1.61  15.12  54.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -10.091     18.966  -0.532    0.596
## size          70.226      9.426   7.450  1.3e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.48 on 126 degrees of freedom
## Multiple R-squared:  0.3058, Adjusted R-squared:  0.3003
## F-statistic:  55.5 on 1 and 126 DF,  p-value: 1.302e-11
```

Let's add the regression line to the plot.

```
plot(hds$size,hds$price,xlab="size",ylab="price")
abline(hdreg$coef,col="red",lwd=2) #lwd: line width
title(main=paste("correlation = ",round(cor(hds$price,hds$size),2)))
```



R packages

A major reason R is important in data science is that there are *many* R packages that do all kinds of modern statistics.

To use an R package you have to first install it on your computer with

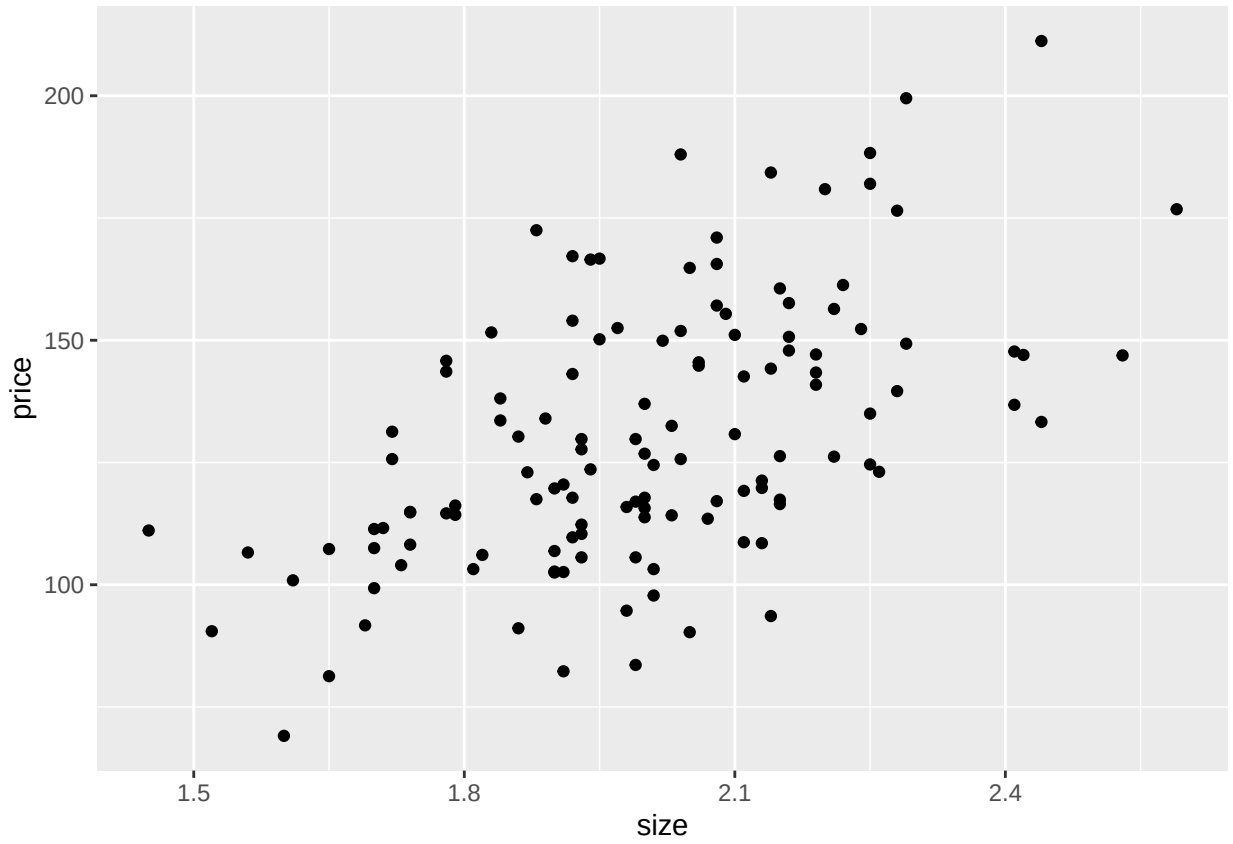
```
> install.packages('package name')
```

Or you can use the interactive package managment in R studio available in the Packages tab of the bottom left panel.

When you want to use an R package in an R session you have to use *library*.
Let's use ggplot2 which is a popular graphics package.

```
library(ggplot2)

p = ggplot(data=hds,aes(x=size,y=price)) + geom_point()
p
```


200
150
100
price
1.5
1.8
2.1
2.4
size