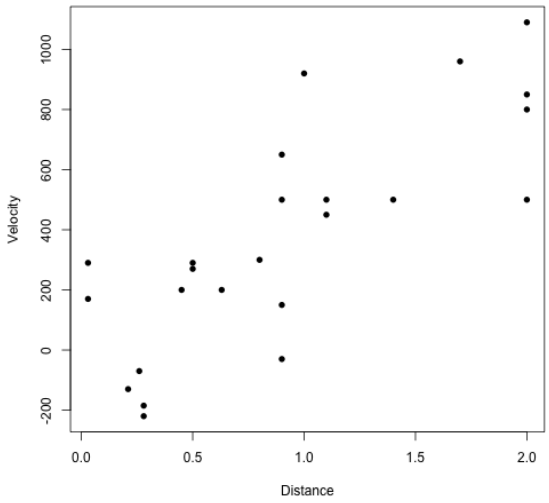
Chapter 5

Example 1: Let's return to the Hubble dataset. Let's load the dataset into R and then plot the data. Note the plot parameter **pch=16** fills in the data points for a nicer looking plot.

```
plot(Velocity~Distance,data=hubble, pch=16)
```



There appears to be a positive relationship between Distance and Velocity. We can calculate the correlation and covariance between the two variables as well.

```
cor(hubble$Distance,hubble$Velocity)

## [1] 0.789032

cov(hubble$Distance,hubble$Velocity)

## [1] 189.159
```

The correlation is +0.789 indicating a positive, moderately strong linear relationship between Distance and Velocity. The covariance is 189.159 which is less interpretable since covariance is dependent on the units of measurement. For this reason, we typically prefer to use correlation rather than covariance.

Example 2: The **mtcars** dataset is provided in the base version of R. To find out more info about this dataset, you can get details by calling **?mtcars**.

```
# get info on the mtcars dataset
?mtcars
```

In the side window of R Studio, it should display the following description after running **?mtcars**:
The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (197374 models).

Let's load the dataset and take a look at the first few values.

```
data(mtcars)
head(mtcars)

##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

In the code below, we extract just these two variables from **mtcars** and create a new variable **m**. From this new variable **m**, we can construct a two-way table of counts for the combinations of **cyl** and **gear**.

```
# extract the variables of interest
m = mtcars[,c("cyl","gear")]

# construct a two-way table of counts for cyl and gear
m.table = table(m); m.table

##    gear
## cyl  3  4  5
##   4  1  8  2
##   6  2  4  1
##   8 12  0  2
```

We can also construct the joint probability mass function by dividing our table by the total number of observations.

```
# calculate the number of total observations
n = sum(m.table); n

## [1] 32

# construct the joint pmf
m.pmf = table(m)/n; m.pmf

##    gear
## cyl       3       4       5
##   4 0.03125 0.25000 0.06250
##   6 0.06250 0.12500 0.03125
##   8 0.37500 0.00000 0.06250
```

Using the table, we can see that P(gear=3 and cyl=8) = 0.375 for example. What is P(gear=4 and cylinder = 4)? The middle entry of the table tells us this value is 0.125. In other words, 12.5% of the cars in this dataset have 4 gears and 4 cylinders.

Next, we are going to look at the joint probability mass function (PMF) of the variables **cyl** and **gear**. We can get the marginal PMFs by summing across rows and columns of the joint pmf.

```
# marginal pmf of gear
colSums(m.pmf)

##       3       4       5
## 0.46875 0.37500 0.15625

# marginal pmf of cyl
rowSums(m.pmf)

##       4       6       8
## 0.34375 0.21875 0.43750
```

From the tables above, you should see that P(gear=3)=0.46875 and P(cyl=4)=0.34375. Finally, we can get the correlation as follows. Note that gear and cyl have a negative, moderate linear relationship.

```
cor(m$cyl,m$gear)

## [1] -0.4926866
```