Topic 16 Numerical and Graphical Summaries of Data

It is difficult to get an overall picture of a large set of observations by simply looking at the list of numbers. It is good to organize the data in a picture or to summarize the data with sample statistics in order to get an overall picture. In general it helps to have some estimate of central tendency for univariate data and a measure of how spread out the data is. We also like to see the shape of the data, whether it is symmetric around the center or skewed.

In this section we will use three data sets; a larger set(than in the previous lecture) from the NFL combine for Notre Dame Players who later played Professional Football stored in `CombineND.csv`, a data set with points per minute per game for Steven Curry and Jeremy Lin for the 2015_2016 season `PPM.csv`, and a data set containing data on all NFL Players for the year 2014 stored in `NFL.csv`.

You should Load all three data sets

```
> CombineND<-read.csv("combineND.csv",header = TRUE)
> PPM<-read.csv("PPM.csv",header = TRUE)
> NFL<-read.csv("NFL.csv",header = TRUE)
```

Measure of center the most common measures of center are the mean and the median.

The Sample Mean: The sample mean is just the average of the set of data;

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{n},$$

where our data set is $\{x_1, x_2, \ldots, x_n\}$.
We can calculate it with `R` in either of 2 ways, we can take the sum of the observations divided by the number of them or we can use the inbuilt `mean()` function. Make sure you set `na.rm = TRUE`.

```
> head(CombineND)
```

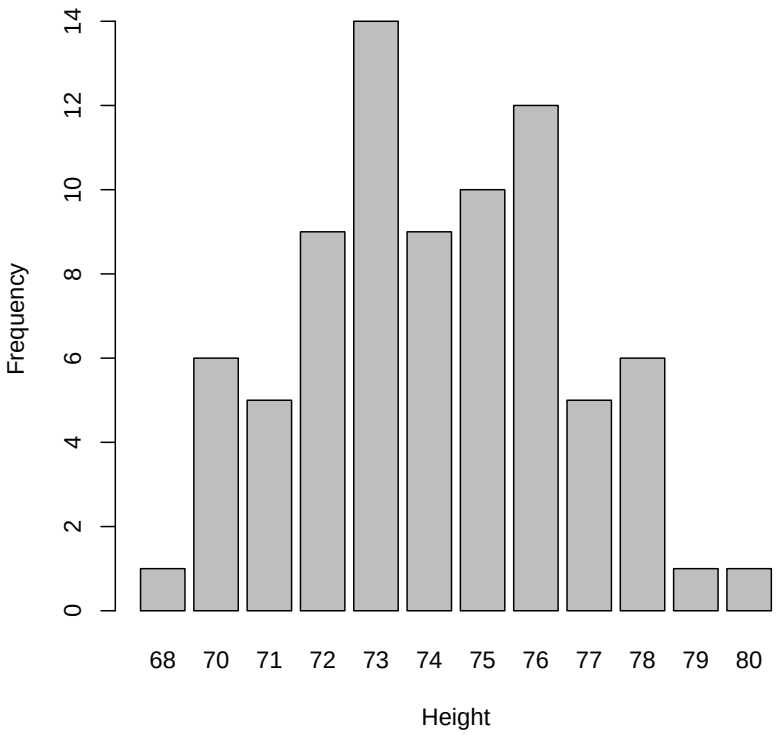
```
  Year           Player Pos Height  Wt FortyYD Vertical BenchReps BroadJump
1 2016      Sheldon Day  DT     73 293    5.07     30.0        21       102
2 2016 Keivarae Russell  CB     71 192    4.49       NA        17        NA
3 2016     C.J. Prosise  RB     72 220    4.48     35.5        NA       121
4 2016      Will Fuller  WR     72 186    4.32     33.5        10       126
5 2016   Ronnie Stanley  OT     78 312    5.20     28.5        NA        NA
6 2015     Kyle Brindza   K     73 236    5.17       NA        14        NA
  ThreeCone Shuttle            Draftedby
1      7.44    4.50 Jacksonville Jaguars
```

```
2        NA      NA   Kansas City Chiefs
3        NA      NA     Seattle Seahawks
4      6.93    4.27       Houston Texans
5      8.03    4.90     Baltimore Ravens
6        NA      NA
```

```
> mean(CombineND$Height)

[1] 74.08861

> mean(CombineND$BenchReps)

[1] NA

> mean(CombineND$BenchReps,na.rm = TRUE)

[1] 21.375
```

The mean gives us a measure of center in the following sense. If we draw a barchart of discrete data, where we draw a bar above each observation with height equal to the frequency of that observation; the mean is the balance point of the picture. It is the point under which I should hold my finger (on the horizontal axis) to balance the distribution. To create a barplot, we first make a table showing the frequency of each value in the data with the `table()` function.

```
> Height.freq<-table(CombineND$Height)
> Height.freq

68 70 71 72 73 74 75 76 77 78 79 80
 1  6  5  9 14  9 10 12  5  6  1  1

> barplot(Height.freq, xlab="Height",ylab="Frequency", )
```



We calculated a mean height of 74.08861 for the players in the data set which is the balance point of this picture. We can also calculate the mean from a summary of the frequency distribution. We have

$$\bar{x} = \frac{y_1 \cdot f_1 + y_2 \cdot f_2 + \cdots + y_k \cdot f_k}{N}$$

where the set $\{y_1, y_2, \ldots, y_k\}$ is a full set of representatives for the observations in our data set and f_i is the frequency with which y_i occurs in the data set. The number of observations in the data set is N. From our summary table of `CombineND$Height` above, we see that we can calculate the mean in the following way:

```
> reps<-c(68,70:80)
> reps

 [1] 68 70 71 72 73 74 75 76 77 78 79 80

> freq<-c(1 , 6 , 5,  9, 14,  9, 10, 12,  5,  6,  1,  1)
> freq
```

```
 [1]  1  6  5  9 14  9 10 12  5  6  1  1
```

```
> m<-sum((reps*freq)/sum(freq))
> m
```

```
[1] 74.08861
```

Deviations from the mean. The deviation of a data point x_i from the mean$\bar{x}$ is $x_i - \bar{x}$. If the deviation is negative, the data point falls below the mean and if it is positive, the data point falls above the mean. The sum and hence the average of all deviations from the mean should be 0.

```
> devH<-CombineND$Height-mean(CombineND$Height)
> head(devH)
```

```
[1] -1.088608 -3.088608 -2.088608 -2.088608  3.911392 -1.088608
```

```
> sum(devH)
```

```
[1] 3.836931e-13
```

Trimmed Mean The mean can be heavily influenced by outliers. For Example:

```
> y<-c(1,2,3,4,5,6,7,8,9,1000)
> mean(y)
```

```
[1] 104.5
```

We see here that the mean is not representative of the majority of the data set. For this reason, people often use a trimmed mean, where they trim off a certain percentage of the highest and lowest observations before calculating the mean. (One often sees this in action in sports where a panel of judges give a rating or score for a performance and the highest and lowest scores are dropped before averaging.) To calculate the trimmed mean, trimming 10% (equal amounts from both ends) we use:

```
> mean(y,trim=0.10)
```

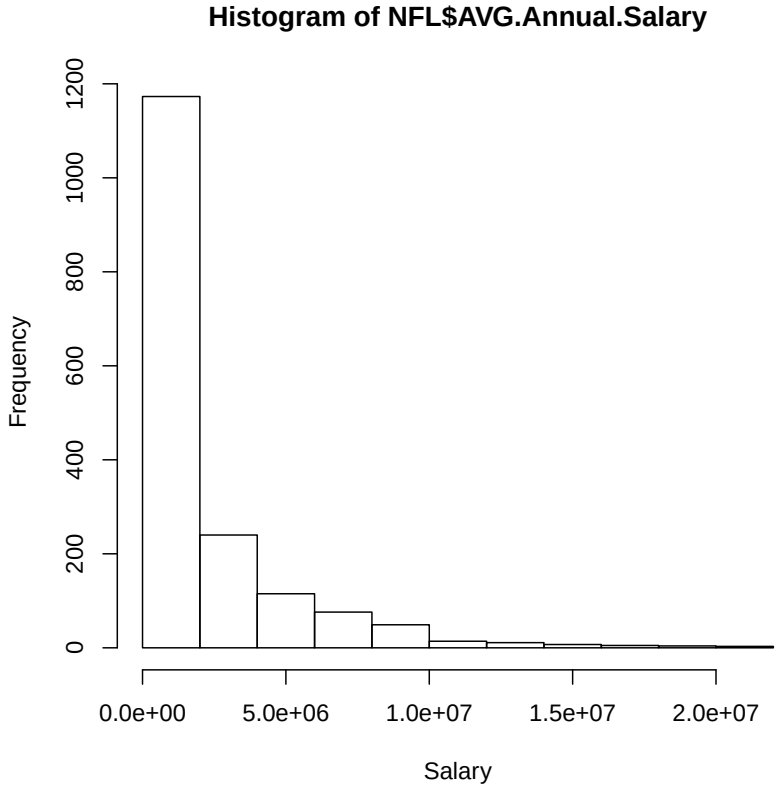
```
[1] 5.5
```

```
> mean(CombineND$Height,trim=0.10)
```

```
[1] 74.09231
```

It doesn't make a big difference in the second case since there are no outliers in the data. On the other hand, income distributions are quite often skewed. We can make a histogram to show the salaries which splits the range of salaries into several categories and shows the number of observations in each category.

```
> #NFLSalary<-table(NFL$AVG.Annual.Salary)
> #NFLSalary
> mean(NFL$AVG.Annual.Salary,na.rm=TRUE)

[1] 2269873

> max(NFL$AVG.Annual.Salary,na.rm=TRUE)

[1] 2.2e+07

> mean(NFL$AVG.Annual.Salary,na.rm=TRUE,trim=0.10)

[1] 1600828

> hist(NFL$AVG.Annual.Salary, xlab="Salary",ylab="Frequency" )
```



Clearly the income distribution is heavily skewed and the trimmed mean is less than the overall mean.

```
> quantile(NFL$AVG.Annual.Salary,0.90, na.rm=TRUE)

  90%
6e+06
```

The Median Because the mean is suceptible to the influence of outliers, the median is frequently used as a measure of center. The **Median** has the property that 50% of the data lies above it and 50% below it. If we have an odd number of observations, we can find the median by sorting them from smallest to largest and finding the middle observation. For example in the data set `y1` below which is already ordered, the median is the sixth observation, median = 5.We see here that the median is more representative than the mean since the mean is skewed by the outlier.

```
> y1<-c(0,1,2,3,4,5,6,7,8,9,1000)
> median(y1)

[1] 5

> mean(y1)

[1] 95
```

if we have an even number of observations, we sort from smallest to largest and the median is the average of the middle two observations. For example in the example below, the median is the average of the fifth and sixth observation which is 5.5.

```
> y<-c(1,2,3,4,5,6,7,8,9,1000)
> median(y)

[1] 5.5
```

The medain is not influenced by outliers, so trimming does not change it.

```
> median(CombineND$Height)

[1] 74

> mean(CombineND$Height)

[1] 74.08861

> median(NFL$AVG.Annual.Salary, na.rm=TRUE)
```

```
[1] 901250
```

```
> mean(NFL$AVG.Annual.Salary, na.rm=TRUE)
```

```
[1] 2269873
```

We see that the median is much more representative for players' salaries in the NFL, but for the Height of the players from ND in the NFL Combine data, it is very close to the mean. Since the median cuts the distrinution of the data in half, we see that when the barchart or histogram are roughly symetric about the center, the median is close to the mean, but when the data is skewed, they can be quite far apart.

Quartiles and Percentiles(quantiles): The p th percentile is a value for which p percent of the data is less than the value and (1 - p) percent of the data is above it. The median is a special case of this, it is the 50th percentile. There is some ambiguity in the definition and sometimes different formulas for the calculation of the pth percentile. Like the median, sometimes it coincides with a data point and sometimes it lies between two data points. Whatever definition is used to come up with the actual number, the pth percentile always has the same interpretation (given in boldface above).

The 25th, 50th and 75th percentiles are also called quartiles. We can calculate these and all percentiles in `R` using the `quantile` command.

```
> quantile(CombineND$Height, 0.25, na.rm=TRUE)
```

```
25%
 72
```

```
> quantile(CombineND$Height, c(0.25,0.5,0.75), na.rm=TRUE)
```

```
25% 50% 75%
 72  74  76
```

```
> quantile(NFL$AVG.Annual.Salary,0.25, na.rm=TRUE)
```

```
   25%
578513
```

```
> quantile(NFL$AVG.Annual.Salary,0.9, na.rm=TRUE)
```

```
  90%
6e+06
```

To find which quantile a particular data point is at you can use the `ecdf()` function:

```
> ecdf(NFL$AVG.Annual.Salary)(1000000)

[1] 0.5309369

> ecdf(NFL$AVG.Annual.Salary)(2000000)

[1] 0.6912198

> ecdf(NFL$AVG.Annual.Salary)(3000000)

[1] 0.7672363
```

The Mode The mode is the most frequently observed value in the data set, it may not be unique. We can find the mode using the `table()` function we used above to summarize the data on heights.

```
> Height.freq<-table(CombineND$Height)
> Height.freq

68 70 71 72 73 74 75 76 77 78 79 80
 1  6  5  9 14  9 10 12  5  6  1  1

> which(Height.freq==max(Height.freq))

73
 5
```

Measures of Variability: It is impotant to have some measure of variability in the data.

The Range of a set of data is the largest measurement minus the smallest measurement. For example

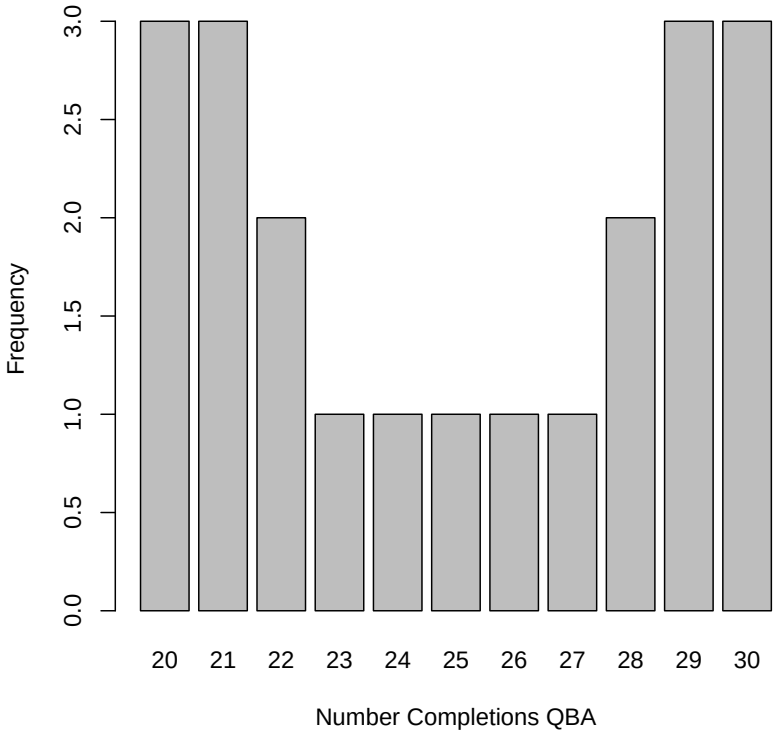
```
> x1<-c(1,22,25,25,25,25,28,50)
> x2<-c(1,2,3,4,47,48,49,50)
> max(x1)-min(x1)

[1] 49

> max(x2)-min(x2)

[1] 49
```

```
> range(x1)

[1]  1 50

> range(x2)

[1]  1 50
```

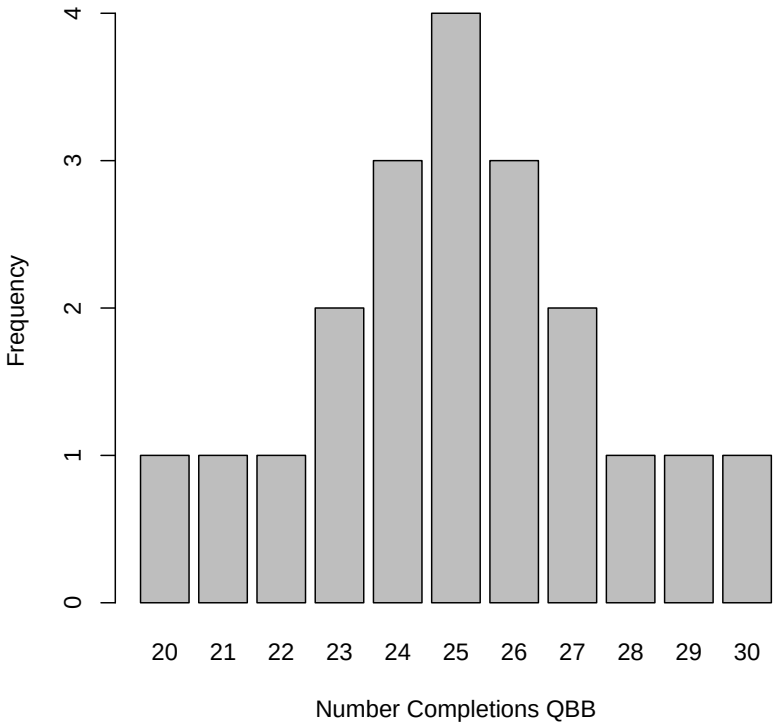
Although the range is easy to compute it is a crude measure of variability. The above data sets have the same range, 49, but obvious differences in the pattern of variability. Consider the following data showing the number of completions in recent games for two quaterbacks:

Game	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Quarterback A:	20	30	29	28	30	20	21	22	25	24	20	22	30	27	28	26	29	29	23	21	21
Quarterback B:	25	30	29	20	21	25	27	22	23	26	27	25	24	26	23	24	25	26	28	24	

```
> options(width=150)
> QBA<-c(20, 30, 29, 28,  30,  20, 21, 22, 25,  24, 20, 22, 30, 27, 28, 26,29,29,23,21,21)
> QBB<-c( 25, 30, 29, 20,  21,  25, 27, 22, 23,  26, 27, 25, 24, 26, 23, 24,25,26,28,24)
> mean(QBA)

[1] 25

> mean(QBB)

[1] 25

> range(QBA)

[1] 20 30

> range(QBB)

[1] 20 30

> A<-table(QBA)
> B<-table(QBB)

> barplot(A, xlab="Number Completions QBA",ylab="Frequency", )
```




```
> barplot(B, xlab="Number Completions QBB",ylab="Frequency", )
```



Both Quarterbacks have the same average number of completions over their recent games, $\bar{x} = 25$ and the same range. However we see that Quarterback A has a a more varied performance record than Quarterback B. Obviously one needs to take this difference in variability in the data into account when comparing the quarterbacks.

We can see that the deviations from the mean here catch the variability, however when we average the deviations from the mean, we get 0 because of cancellation. Now the distance of a data point from the mean is $|x_i - \bar{x}|$ which is the absolute value of the deviation from the mean. The average distance of the data points from the mean is a reasonable measure of variability, however the more commonly used measure is the **standard deviation**, which is similar but not quite the same.

The Sample Variance The sample variance is almost the average squared distance of the data points from the mean and is given by

$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{n - 1}$$

where n is the number of observations in the sample. (The division by $n - 1$

gives us a better estimator of the population variance.) We can calculate this using the `var()` function in R.

```
> var(QBA)

[1] 14.6

> var(QBB)

[1] 6.421053
```

Clearly we see that the variation in the performance of Quarterback A is much larger than that of Quarterback B accourding to this measure of variability. One drawback is that we have squared the units in the process of calculation and this distorts our perception of the difference in performance. So in an effort to return to the original units, we take the square root of the variance as our measure of variability. The sample **standard deviation** is the square root of the sample variance:

$$s = \sqrt{s^2} = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \cdots + (x_n - \bar{x})^2}{n-1}}$$

```
> options(width=100)
> sd(QBA)

[1] 3.820995

> sd(QBB)

[1] 2.53398

> sd(CombineND$Height,na.rm=TRUE)

[1] 2.497126

> m<-mean(CombineND$FortyYD,na.rm=TRUE)
> m

[1] 4.795443

> s<-sd(CombineND$FortyYD,na.rm=TRUE)
> s

[1] 0.282102

> CombineND[which(CombineND$FortyYD<=m-(1.5)*s),]

  Year      Player Pos Height  Wt FortyYD Vertical BenchReps BroadJump ThreeCone Shuttle
4 2016 Will Fuller  WR     72 186    4.32     33.5        10       126      6.93    4.27
        Draftedby
4 Houston Texans
```

Interpretation of the Standard Deviation When presented with raw scores for performance, it is difficult to interpret their meaning without some measure of center and variability for the population from which they come. In any set of data, whether it is population data or a sample, observations that are more than 3 standard deviations from the mean are rare and exceptional. A theorem by Chebychev tells us that the proportion of values more than k standard deviations above or below the mean is no more than $1/k^2$. For example in any data set, less than 1/9 of the data is more than 3 standard deviations from the mean. Since this applies to data with any shape, it is a conservative estimate. For data that looks mound shaped or bell shaped, less than 0.3% of the data will be more than 3 standard deviations away from the mean.

Quite often when interpreting a data observation, such as height, weight, speed or salary we are interested in how it compares to the rest of the relevant population. Measures of relative standing describe the location of a particular measurement relative to the rest of the data. Percentiles give us some idea of the relative standing of a data point. We can also use z-scores which measure how many standard deviations an observation lies above or below the mean.

Z-Scores The z-score for a particular measurement in a set of data, measures how many standard deviations that measurement lies away from the mean. Recall the deviation $x_i - \bar{x}$ tells us how far above or below the mean the data point x_i lies.

The **z-score** for a data measurement, x_i is

$$z = \frac{x_i - \bar{x}}{s}$$

and it tells us how many standard deviations above or below the mean the data point x_i lies.

Example: we know that Ronnie Stanley was pretty tall at 78 inches = 6ft. 6in.. Now lets see where he stands relative to the population of NFL players in 2014, i.e. what the z-score of 78 is.

```
> (78-mean(NFL$HT, na.rm=TRUE))/sd(NFL$HT, na.rm=TRUE)

[1] 1.507512

> ecdf(NFL$HT)(78)

[1] 0.9670394
```

We see that he is 1.5 standard deviations above the average for height of football palyers in the NFL. We also see that 96 percent of the players have a height less than or equal to Ronnie Stanley's.

Example we can also use z-scores to check which event a player did better in relative to the other players in the CombineND data set. Since the forty yard dash and the three cone task have different distributions, it is difficult to figure out which one a player did better in from the raw performance scores. If we standardize the scores (calculate z-scores) then we can see in which event the athlete had a better relative performance. let's check out Golden Tate's performance in the FortyYD and the Three Cone task.

```
> options(width=100)
> CombineND[CombineND$Player=="Golden Tate",]

   Year      Player Pos Height  Wt FortyYD Vertical BenchReps BroadJump ThreeCone Shuttle
34 2010 Golden Tate  WR     70 199    4.42       35        17       120      7.12    4.34
          Draftedby
34 Seattle Seahawks

>
```

We see his time for the FortyYD was 4.42 and for the ThreeCone was 7.12. Let's compare his relative performance with z-scores:

```
> options(width=100)
> zfortyyd<-(4.42-mean(CombineND$FortyYD,na.rm=TRUE))/sd(CombineND$FortyYD,na.rm=TRUE)
> zfortyyd

[1] -1.330877

> zthreecone<-(7.12- mean(CombineND$ThreeCone,na.rm=TRUE))/sd(CombineND$ThreeCone,na.rm=TRUE)
> zthreecone

[1] -0.2556081
```

We see that although his performance was better than average in both cases, he did better in the Forty Yard Dash since he was 1.33 standard deviations below the mean.

We can compute the z-scores for a variable using the `scale()` function.

```
> Income<-scale(NFL$AVG.Annual.Salary)
> head(Income)

           [,1]
[1,] -0.6334155
[2,] -0.6334155
[3,] -0.6334155
[4,] -0.5272683
[5,] -0.7772278
[6,] -0.7772278

> ecdf(Income)(2)

[1] 0.9469652
```

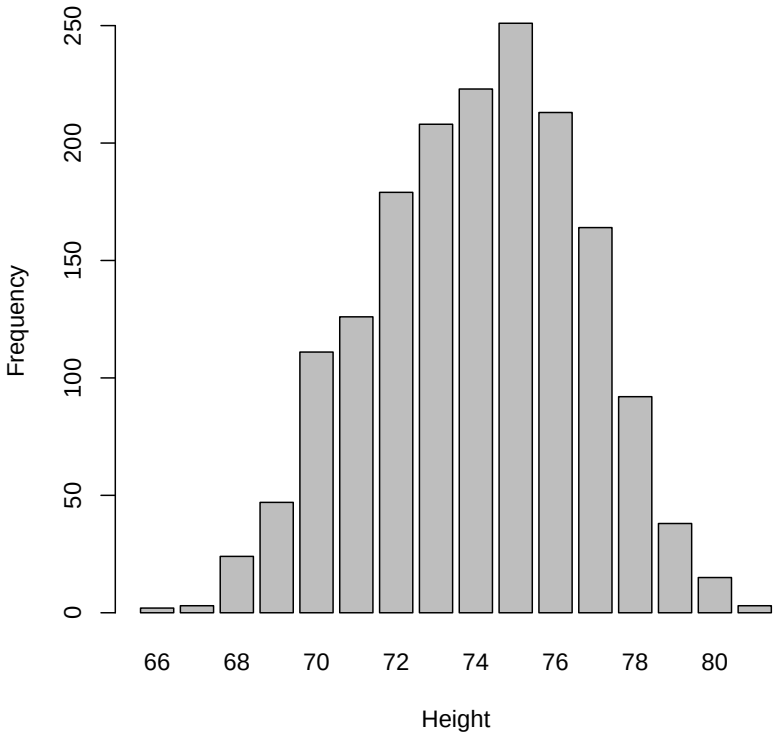
```
> ecdf(Income)(-2)

[1] 0

> ecdf(Income)(0)

[1] 0.7183265
```

We can see that the distribution is not symmetric since about 6% of the players have an income which is more than 2 standard deviations above the mean and no players have an income more than 2 standard deviations below the mean. In fact about 72 percent have an income below average.

Viewing The Shape of The Data We can draw pictures of our data.

Bargraph If the data is discrete and there are not too many values involved, we can draw a bargraph showing the frequencies of each observation in the data set with the `barplot()` function. You must first make a frequency table with the command `table()`.

```
> Ht<-table(NFL$HT)
> Ht

 66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81
  2   3  24  47 111 126 179 208 223 251 213 164  92  38  15   3

> barplot(Ht, xlab="Height",ylab="Frequency" )
```



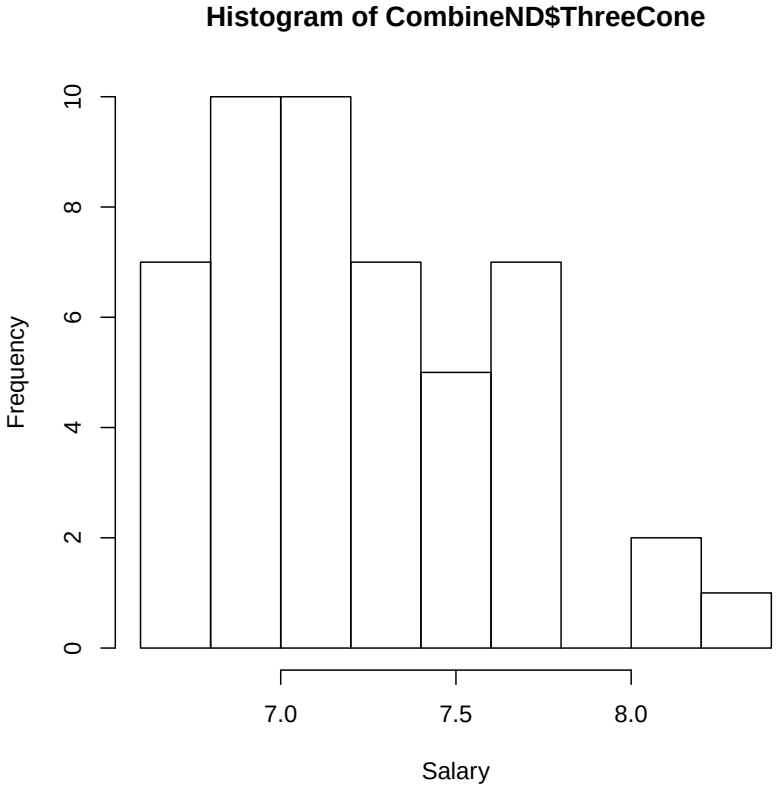
Stem and Leaf Plot A stem and leaf plot is often a useful way to get a picture of data.

```
> stem(CombineND$BenchReps)

  The decimal point is 1 digit(s) to the right of the |

  1 | 011124
  1 | 566677777999999
  2 | 0000011111122233444444
  2 | 5677999
  3 | 0011
  3 | 55
```

Here we see the each data point in the BenchReps data set is split into two parts, the stem and a leaf. The stem is shared by several data points. For example the data point 15 is represented in the top line with a 1 to the left of the divider and a 5 to the right. A data point 16 shares the same stem with a 6 appearing to the right of the 5. the three 6's in that row show that there were three 16's in the data set.

Histogram A histogram is similar to a bar graph but is more appropriate for continuous data or discrete data waith many values. To construct a histogram, we choose an interval within which all of the observations lie and which starts and ends close to the minimum and maximum observations respectively, but with more appealing endpoints if necessary. One important feature of Histograms and bar graphs is that the adhere to the **area principle**, that is the area of the bar devoted to an interval is proportional to the amount of data in the interval. We then split that interval into subintervals of equal length which do not overlap but which cover the entire interval when put together. The subintervals in R by default follow the right endpoint rule, that is they include the right end point but not the left of the subinterval. For example

```
> hist(CombineND$ThreeCone, xlab="Salary",ylab="Frequency" )
```



R automatically chooses the intervals for us unless we specify. We can find out where the cuts are made and hence what the sunintervals are by saving the histogram as an object and then printing it out as follows

```
> options(width=100)
> histinfo<-hist(CombineND$ThreeCone)
> histinfo
```

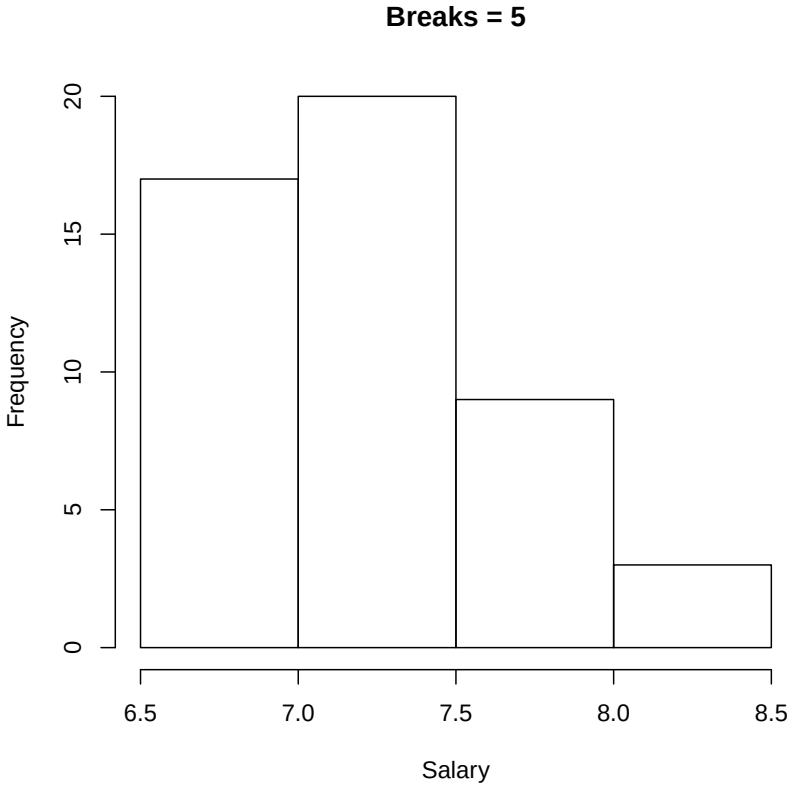
```
$breaks
 [1] 6.6 6.8 7.0 7.2 7.4 7.6 7.8 8.0 8.2 8.4

$counts
[1]  7 10 10  7  5  7  0  2  1

$density
[1] 0.7142857 1.0204082 1.0204082 0.7142857 0.5102041 0.7142857 0.0000000 0.2040816 0.1020408

$mids
[1] 6.7 6.9 7.1 7.3 7.5 7.7 7.9 8.1 8.3

$xname
[1] "CombineND$ThreeCone"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

From the list of breaks, we see that the the first subinterval is from 6.6 to 6.8 and since the default in R is to include the right end point, the interval is $(6.6, 6.8]$. From the list of counts, we see that there are 7 observations in this interval. You could call up these pieces of information individually

```
> histinfo$counts

[1]  7 10 10  7  5  7  0  2  1
```

You may want to choose your own intervals (called bins) and this is possible. You can choose the number of breaks with the argument `breaks` (the argument `main` below gives you the label).

```
> hist(CombineND$ThreeCone, xlab="Salary",ylab="Frequency",
+      breaks=5, main="Breaks = 5" )
```



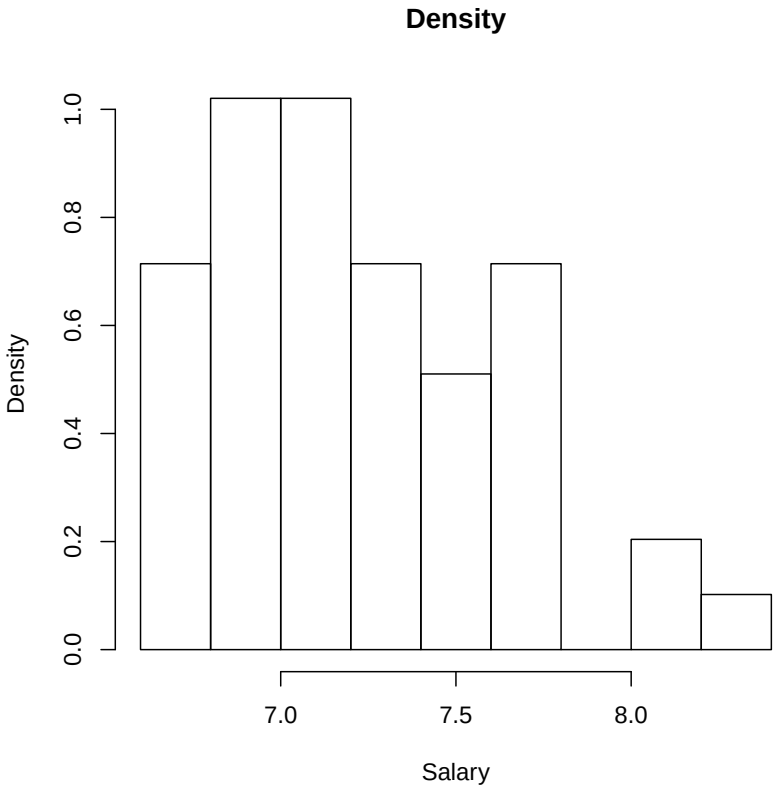
You can also feed R a vector of break points and we switch to intervals including the left end point below with `right=FALSE`.

```
> hist(CombineND$ThreeCone, xlab="Salary",ylab="Frequency",
+      breaks=c(seq(6.5,8.5,by=0.1)), main="vector of breaks",right=FALSE )
```

Here, the command `seq(6.5,8.5,by=0.1)` creates a vector starting at 6.5, ending at 8.5 increasing in increments of 0.1.

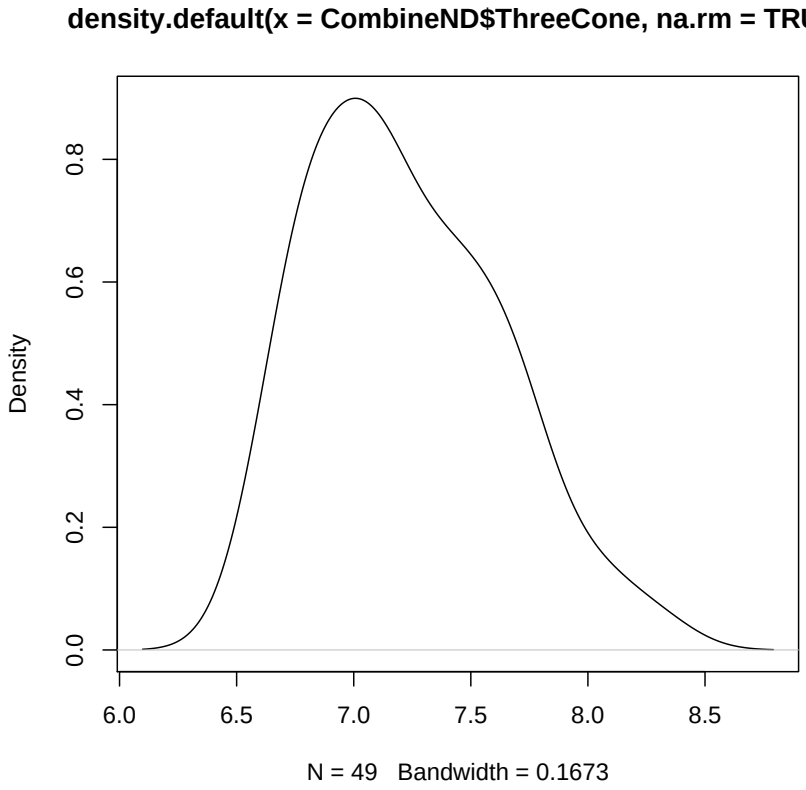
Density Plots: If we use the argument the settings `freq= FALSE` or `probability =TRUE` for our histogram, we get a histogram where the total area of the bars shown is 1 and the area of each bar gives the probability of getting an observation in that interval if we pick a data point at random.

```
> hist(CombineND$ThreeCone, xlab="Salary",ylab="Density",
+      probability=TRUE, main="Density" )
```



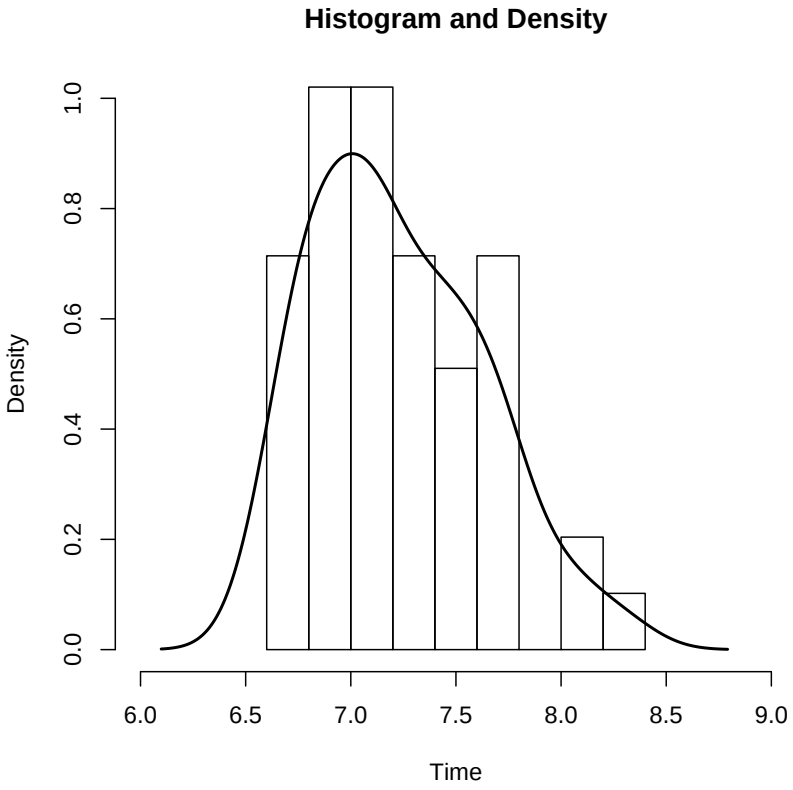
For a continuous variable such as the time it take an athlete to complete the three cone task, we could take a finer and finer mesh of intervals to cover the entire range of possible times that could occur in the data. if we had data for the entire population of football players who had participated in this event, the outline of the (probability) histogram would start to look more and more like a continuous curve as we took more and more intervals. For the limiting curve, the total area underneath it would be 1 and the area above any interval would be the probability that we would observe a time in that interval if we choose a record at random from the data. This limiting curve is called the density function of the variable. R will estimate the (probability) density function (pdf) for a variable from the sample data with the command `density()`.

```
> plot(density(CombineND$ThreeCone,na.rm=TRUE))
```

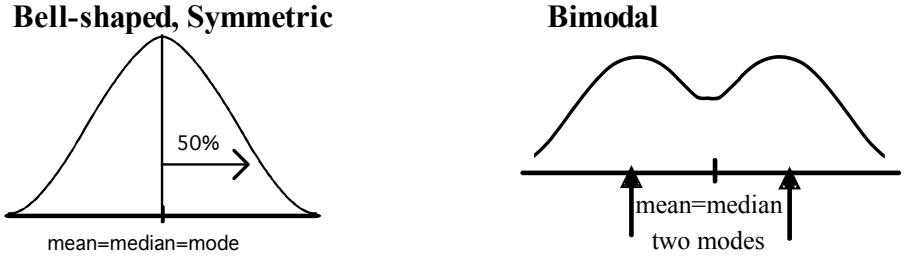


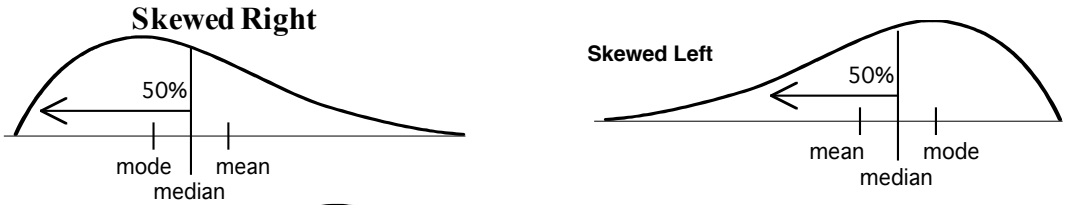
We can plot the scaled histogram and the approximate density on a single graph. Below we choose the window to show with `xlim` and `ylim` arguments and we add the density plot with the command `lines`. The argument `lwd=2` tells R to draw the density line with twice the width of the default line width.

```
> hist(CombineND$ThreeCone, main="Histogram and Density",xlab="Time",
+      probability=TRUE, xlim=range(c(6,9)),ylim=range(c(0,1)) )
> lines(density(CombineND$ThreeCone,na.rm=TRUE),lwd=2)
```



Shape of The Data With large sets of data and narrow class widths, the histogram looks roughly like a smooth curve. The mean, median and mode, have a graphical interpretation in this case. The mean is **the balance point of the histogram of the data**, whereas **the median is the point on the x-axis such that half of the area under the histogram lies to the right of the median and half of the area lies to its left.** The **mode occurs at the data point where the graph reaches its highest point.** This of course may not be unique.





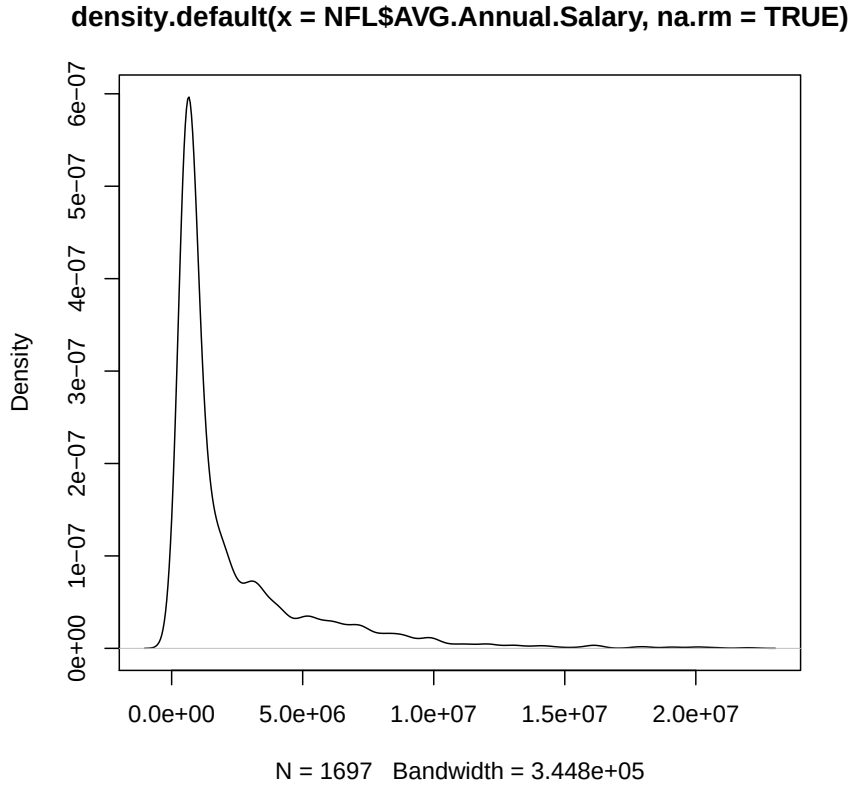
Skewed Data

Definition A data set is said to be **skewed** if one tail of the distribution has more extreme observations than the other tail.

The mean is sensitive to extreme observations, but the median is not(check out the example below).

Example Consider the data from the above example concerning the average yearly salary of NFL Players. here the density function looks like this

```
> plot(density(NFL$AVG.Annual.Salary,na.rm=TRUE))
```



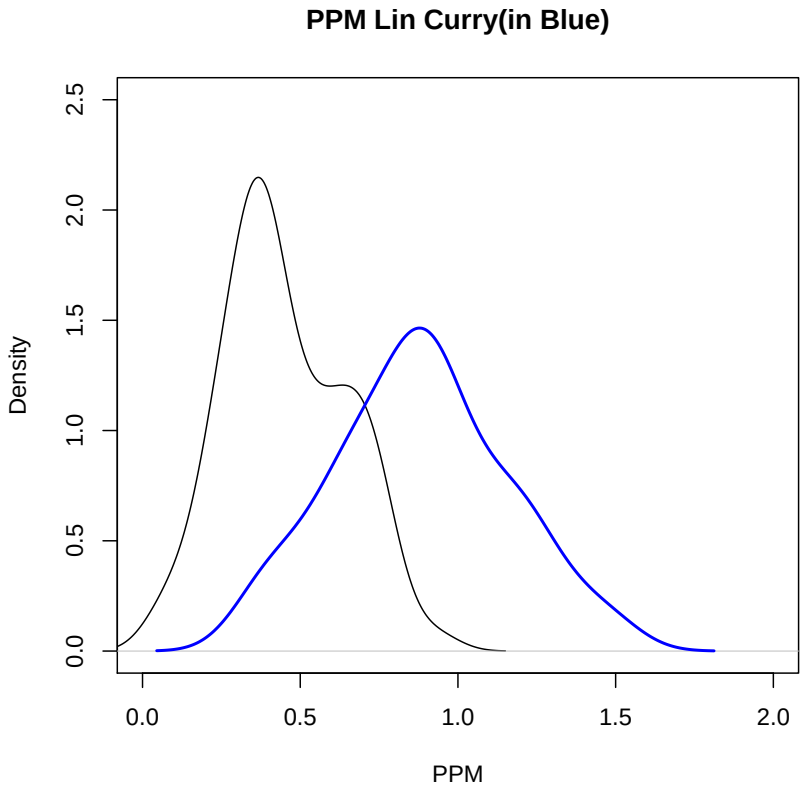
For data skewed to the right, the mean is larger than the median and for data skewed to left, the mean is less than the median.

```
> mean(NFL$AVG.Annual.Salary,na.rm=TRUE)

[1] 2269873

> median(NFL$AVG.Annual.Salary,na.rm=TRUE)

[1] 901250
```

Larger standard deviations mean the data is more spread out and tails are fatter. One might quote the mean as a measure of center here when recruiting, but perhaps quote the median if one is arguing in favour of increased benefits.

Example Let us compare the densities for the number of points per minute per game for Jeremy Lin and Stephen Curry for the year 2015_2016.

```
> plot(density(PPM$linptspermin,na.rm=TRUE),xlim=range(c(0,2)),
+      ylim=range(c(0,2.5)),main="PPM Lin Curry(in Blue)",xlab="PPM" )
> lines(density(PPM$Curryptspermin,na.rm=TRUE),lwd=2,col="blue")
```



These densities tell us that Curry's performance is more spread out and should have a higher standard deviation. It is roughly bell shaped and the mead should be close to the median at the center of the distribution. Lin's performance is a bit skewed so we would expect the mean to be higher than the median and less than the mean for Curry. It is also more concentrated and we would expect the standard deviation to be less than that for Curry. Although they both have roughly the same number of games where their points per minute are between 0.5 and 0.1, about half of Curry's game have PPM higher than 1 whereas the probability thia will happen for Lin is virtually 0. You can get a summary of the statistics in the file in a number of ways with the functions shown below:

```
> summary(PPM)

  linptspermin    Curryptspermin
 Min.   :0.0530   Min.   :0.3570
 1st Qu.:0.3217   1st Qu.:0.6815
 Median :0.4145   Median :0.8920
 Mean   :0.4513   Mean   :0.8827
 3rd Qu.:0.5982   3rd Qu.:1.0625
 Max.   :0.9350   Max.   :1.5000
                  NA's   :7

> sapply(PPM,mean,na.rm=TRUE)

  linptspermin Curryptspermin
     0.4512564      0.8827324

> sapply(PPM,sd,na.rm=TRUE)

  linptspermin Curryptspermin
     0.1906475      0.2710184

> sapply(PPM,median,na.rm=TRUE)

  linptspermin Curryptspermin
        0.4145         0.8920
```