Statistics for Linguists
Hale
February 1, 2010

1 Central tendency: mode, mean and median

It's hard to understand data if you have to look at it all. Descriptive statistics are numbers you can calculate from data which summarize salient qualities in a shorter, more manageable way.

Modal value

The **mode** is the value which occurs most frequently. For instance, if these are the ages of audience members at a very poorly-attended Miley Cyrus concert.

$$9, 12, 15, 15, 15, 16, 16, 20, 26$$

The modal value is 15, occuring three times. The most well-represented age level at the concert was fifteen.

```
> miley <- c(9, 12, 15, 15, 15, 16, 16, 20, 26)
> table(miley)

miley
 9 12 15 16 20 26
 1  1  3  2  1  1
```

Mean

Your old friend the average.

$$\bar{y} = (y_1 + y_2 + y_3 + ... + y_n)/n = \frac{\sum_{i=1}^{n} y_i}{n}$$

The Greek uppercase letter sigma ($\sum$) means add up each measurement y; often the boundaries $i = 1$ and n are omitted since there are no special restrictions – just add up everything and divide by the number of observations.

For instance, if the ages of the children currently in McDonalds' Playland at the moment are: $3, 4, 4, 5, 6, 7$ and 10, we can summarize the situation by saying:

$$\text{mean age of playland visitor} = \frac{3+4+4+5+6+8+10}{7} = \frac{40}{7} = 5.7$$

It would misrepresent the precision of our observations (i.e. to the nearest year) to go beyond two significant digits in reporting the mean age. Could you write this averaging function in R? Sure, taking advantage of `sum` being vectorized. It "knows" how long the vector of measurements is. In the following function `x` would be called a *formal parameter* of the homemade `arithmetic.mean` function.

```
> arithmetic.mean <- function(x) {
+     sum(x)/length(x)
+ }
> arithmetic.mean(c(3, 4, 4, 5, 6, 8, 10))

[1] 5.714286
```

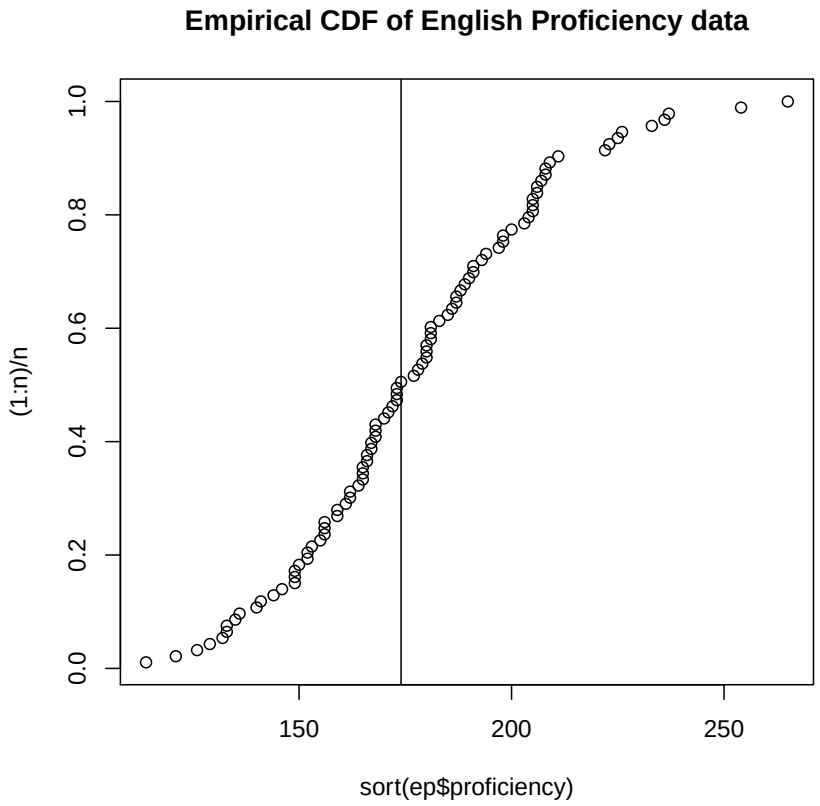
Median

Remember the empirical CDF? Dalgaard explains the empirical CDF by saying

> the fraction of data smaller than or equal to x. That is, if x is the k^{th} smallest observation, then the proportion k/n of the data smaller than or equal to x

Even though we can use the empirical CDF to talk about any such proportion, the median was the name we gave to the proportion $k/n = \frac{1}{2}$. Consider some imaginary results obtained on the Cambridge English Proficiency Exam.

```
> ep <- read.table("english.proficiency", header = T)
> n <- length(ep$proficiency)
> plot(sort(ep$proficiency), (1:n)/n, main = "Empirical CDF of English Proficiency data")
> abline(v = 174)
```



Half the data is smaller than or equal to the median score of 174. It is the midpoint of the sorted list of data values.

Could you write a median function in R? Sure, let the machine do the ranking for you.

```
> length(ep$proficiency)

[1] 93

> 93/2
```

```
[1] 46.5
```

```
> ceiling(46.5)
```

```
[1] 47
```

```
> sort(ep$proficiency)[47]
```

```
[1] 174
```

We have 93 observations, so the middle one would be the 47^{th}; the smallest integer greater than a number is its **ceiling** This exemplifies the case where we have an **odd number** of measurements. When there is an **even number** of measurements, the median is defined to be the average of the two values on either side of the middle.

```
> ep.even <- ep$proficiency[-1]
> length(ep.even)
```

```
[1] 92
```

```
> (sort(ep.even)[46] + sort(ep.even)[47])/2
```

```
[1] 173.5
```

Our homemade median function can then use modular division (%%) to check which method to use[1].

```
> med <- function(x) {
+     odd.even <- length(x)%%2
+     if (odd.even == 0)
+         (sort(x)[length(x)/2] + sort(x)[1 + length(x)/2])/2
+     else sort(x)[ceiling(length(x)/2)]
+ }
```

Of course you don't have to write these functions yourself, there are built-in R functions named **mean** and **median** (the 'mode' functions pertain instead to the runtime system of R, not the statistical notion).

Robustness

What if some really cocky hot-shot student came in and screwed it all up for everybody else by scoring 9000 on the English proficiency test?

```
> median(append(ep$proficiency, 9000))
```

```
[1] 175.5
```

```
> mean(append(ep$proficiency, 9000))
```

```
[1] 271.4574
```

The mean is radically thrown off by such an outlier, whereas the median is not. When a set of data is **skewed** these various descriptions of central tendency fail to coincide. Johnson's Figure 1.14 is an idealized exemple of such a skewed data set.

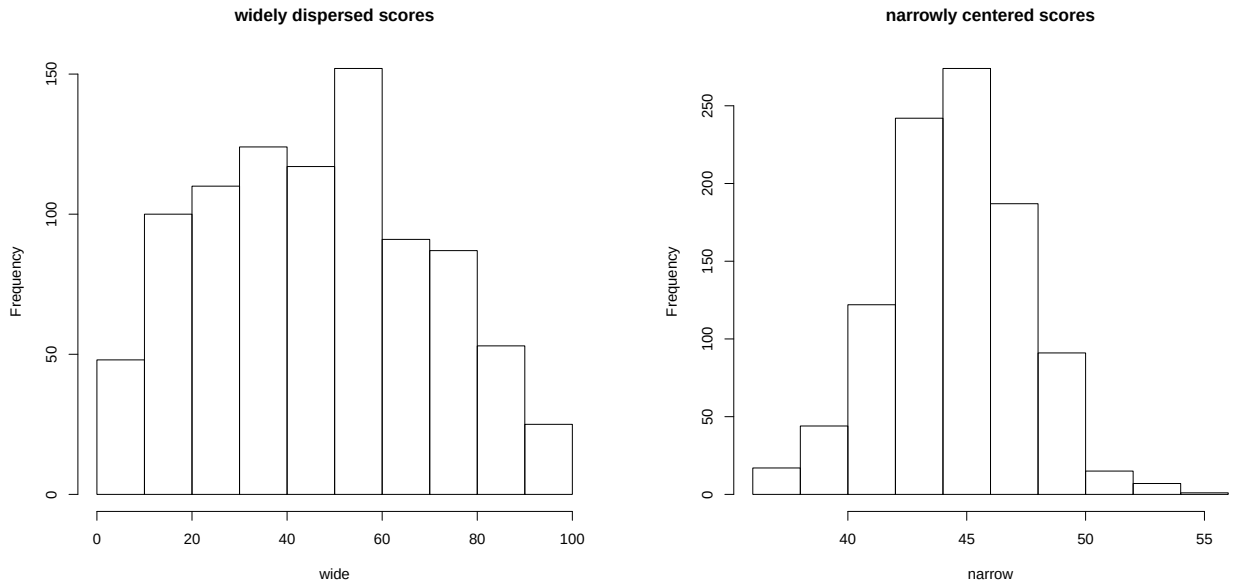
[1] Modular division of x by y yields the remainder of $\frac{x}{y}$. In the case where $y = 2$ the remainder will either be 0, which implies x was even, or 1, which implies y was odd. In general if $x\%\%y$ is zero, then x is a multiple of y.

2 Dispersion

We care about how widely dispersed data are because of its impact on our degree of certainty about where they might be coming from.

Ranges

Both of these histograms come from made-up datasets with 1000 observations with mean about 45.



The `summary` confirms the visual intuition that the data histogrammed on the left are more widely dispersed than the ones on the right.

```
> summary(wide)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00   28.00   46.00   46.46   63.00   99.00

> summary(narrow)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  37.00   43.00   45.00   45.03   47.00   55.00
```

The **interquartile range** $Q_3 - Q_1$ for each made-up dataset describes the difference between them.

25% quartile (Q_1) the median of the observations below the 'grand' median

75% quartile (Q_3) the median of the observations above the grand median.

```
> c(IQR(wide), IQR(narrow))

[1] 35  4
```

The narrowly-dispersed data (on right) has 50% of its observations clustered between 43 and 47 – that's really tight! In the more widely dispersed data (left), such a **50% central interval** lies between 28 and 63; much looser.

2.1 Variance

Does some data set have more or less dispersion, scatter, variability, etc than another? Consider the average lengths of syntactic phrases such as NP and VP in the hand-parsed Brown corpus (see appendix).

```
> np <- read.table(file = "brown-np-lengths", header = T)
> vp <- read.table(file = "brown-vp-lengths", header = T)
> mean(np$wordcount)

[1] 3.087473

> mean(vp$wordcount)

[1] 8.107505
```

we know that that average VP is longer than the average NP. Lets relativize the discussion to take that into account, looking at the **residuals**, $y_{\text{np}} - \bar{y}_{\text{np}}$, the differences between particular NP wordcounts and the average NP wordcount.

```
> head(np$wordcount - mean(np$wordcount), n = 20)

 [1] -1.08747349 -1.08747349 -0.08747349 -2.08747349 -2.08747349 -2.08747349
 [7] -2.08747349 -2.08747349  3.91252651 -1.08747349  0.91252651  6.91252651
[13]  3.91252651 -2.08747349 21.91252651  1.91252651 -0.08747349 -2.08747349
[19] -2.08747349 -2.08747349
```

2.1.1 Difficulty: negative residuals

A negative residual comes about when an NP is shorter than average. Summing up these residuals adds up to...nothing!

```
> sum(np$wordcount - mean(np$wordcount))

[1] 5.65592e-12
```

The problem is that positive and negative residuals are cancelling out. Vasishth provides a nice algebraic argument on page 3 of his notes. One solution is to square each residual, to ensure that account is taken of variability in both directions[2].

2.1.2 Difficulty: number of measurements affects it

The Brown corpus provides 142000 observations of NPs but only 83000 observations of VPs. The relative dispersion could be the same even though there are radically more attestations in one than the other. Our concept of dispersion should be relativized to the number of observations we actually have.

However, if you are going to put five numbers into a box, then you get to make up all five numbers. If instead I tell you what the average of those five must be, then you really only have the freedom to choose four of the numbers — the last is completely predictable. This is like using the mean $\bar{y}$ in calculating a measure of dispersion; data could be very scattered, but not scattered in way such that the residuals don't add up to zero. Thus, the variance has only $\mathbf{n-1}$ **degrees of freedom** because that last residual must be compatible with the mean.

$$\text{variance} = \frac{\text{sum of squared residuals}}{\text{degrees of freedom}}$$

$$s^2 = \frac{(y_1 - \bar{y})^2 + (y_2 - \bar{y})^2 + \cdots + (y_n - \bar{y})^2}{n-1} = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{1}$$

Equation 1 gives the definition of **variance**.

[2]Remember that multiplying negative numbers by themselves yields a positive number, i.e. $(-3)^2 = (-3) \times (-3) = 9$.

2.2 Standard deviation and Z-scores

The **standard deviation** (equation 2) is just the square root s of the variance s^2.

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2}$$

The R function is called `sd`. The standard deviation is like a ruler for judging whether a particular data point is really whacko for this sample (or not).

```
> (10 - mean(np$wordcount))/sd(np$wordcount)

[1] 1.740146
```

An NP that is 10 words long is longer than your average 3-word NP. It is 1.7 standard deviations above the mean for NPs. Being ten words long is no great shakes if you are a VP in the Brown corpus, there, 10 words long is is only a quarter of a standard deviation above the mean.

```
> (10 - mean(vp$wordcount))/sd(vp$wordcount)

[1] 0.2485052
```

Analogous to being in school and being put in the gifted program if you are more than 2 SDs above the average on some test. It doesn't mean you're ready for college, just that you're probably bored with the work other children your age are doing. (This ensures that the school doesn't have to spend too many resources on "special" kids since by definition, most of the data will fall below that mark.)

The **Z-score** facilitates the comparison of different groups having different variances and means by relativizing observations by those quantities.

$$Z_y = \frac{y - \bar{y}}{s} \tag{3}$$

the resulting set of scores always has a mean of zero and a standard deviation of 1.

3 Now you try

- Implement your own variance function in R.
- An elite Ivy-league institution assigns beginning students to Chinese classes on the basis of their language aptitude test scores. Some students will have taken Aptitude Test A, which has a mean of 120 and a standard deviation of 12; the remainder will have taken Test B, which has a mean of 100 and a standard deviation of 15. (The means and std devs for both tests were obtained from large US Government studies).

student	Test A	Test B
P	132	—
Q	124	—
R	—	122
S	81	—
T	—	75
U	—	91

1. Calculate the standardized score for each of the students listed in the table (above) and rank the students according to their apparent ability, putting the best first.

2. The institution groups students into classes C,D,E or F according to their score on Test A as follows: those scoring at least 140 are assigned to class C; those with at least 120 but less than 140 are Class D; those with at least 105 but less than 120 are Class E. The remainder are assigned to Class F. In which classes would you place students R,T and U?

A Parsed corpora

The Linguistic Data Consortium's Catalog `http://www.ldc.upenn.edu/Catalog` lists all the corpora it distributes. The particular releases that we have in the Computational Linguistics lab are stored on a computer named `silver`. If you are a co-signer on the LDC license agreement, you can access it using sftp. Here is how to transfer the Penn Treebank to your local disk:

```
[colmerauer:~] john% sftp jth99@silver.compling
Connecting to silver.compling...
jth99@silver.compling's password:
sftp> cd silver2/ldc
sftp> dir
1995  2004  2005  2006  2009  95
sftp> cd 1995
sftp> dir
LDC95T7.zip
sftp> get LDC95T7.zip
Fetching /mnt/silver2/ldc/1995/LDC95T7.zip to LDC95T7.zip
/mnt/silver2/ldc/1995/LDC95T7.zip                          100%   14MB   6.8MB/s   00:02
sftp> quit
```

Use `unzip` to decompress this archive. Inside the directory `LDC95T7/COMBINED/WSJ` you will find a variety of numbered subdirectories. Files in these "WSJ" subdirectories contain text that was attested in the Wall Street Journal in 1988 or 1989. Each sentence is associated with a hand-corrected parse tree as shown below. "COMBINED" just means that the attested sentences are enriched both with part-of-speech tags with syntactic phrase annotations.

```
[colmerauer:~] john% cd LDC95T7
[colmerauer:~/LDC95T7] john% cat COMBINED/WSJ/00/WSJ_0001.MRG
( (S
    (NP-SBJ
      (NP (NNP Pierre) (NNP Vinken) )
      (, ,)
      (ADJP
        (NP (CD 61) (NNS years) )
        (JJ old) )
      (, ,) )
    (VP (MD will)
      (VP (VB join)
        (NP (DT the) (NN board) )
        (PP-CLR (IN as)
          (NP (DT a) (JJ nonexecutive) (NN director) ))
        (NP-TMP (NNP Nov.) (CD 29) )))
    (. .) ))
```