Statistics for Linguists
Hale
January 25, 2010

Administrative

Welcome to *Statistics for Linguists*, LING4476. The goal of this course is to give linguistics students statistical literacy. This is increasingly important as the profession in general becomes more receptive to numbers. The course is not a computational linguistics course, but we will be learning how to use a software tool called 'R' to do our calculations for us.

instructor John Hale is going to do his best to help you with your project, but his powers are necessarily limited. Free professional help is available from `http://www.cscu.cornell.edu`.

books Foundations of Statistics: a simulation-based approach.
By Shravan Vasishth and Michael Broe
`http://www.ling.uni-potsdam.de/~vasishth/SFLS.html` Draft dated July 17, 2009.

Quantitative Methods in Linguistics.
By Keith Johnson
`http://www.blackwellpublishing.com/quantmethods`

software The course uses the R system for statistical computing and graphics. It is a free version of Becker et al. [1988] 'S' language that offers "an integrated suite of software facilities for data manipulation, calculation and graphical display." It was originally created by Ross Ihaka and Robert Gentleman at the University of Auckland Statistics Department. It works on many computer operating systems including Unix, Windows and Mac OS. It is extendable in that it incorporates a functional programming language influenced by Lisp.

`http://www.r-project.org`

Please go to the R website and install it on your computer. Read the FAQ for installation tips. Take a look at the Introduction to R [Venables et al., 2006] posted on the distribution site under 'Manuals'.

evaluation Students in this class are graded on three things

participation	20 %
homework	30 %
final project	50 %

Participation measures students' intellectual engagement: was the student in class? Did he or she contribute questions about material not initially understood? Did he or she seek help at office hours or by email?

Homeworks are meant to apply ideas introduced in class and to strengthen students' analytical and computer skills. Distributed semi-weekly, they are graded either as pass/fail or on a 7-point scale.

The final project encourages reflection about the use of probability or statistics in a theoretical model or empirical study of the student's own choosing. This project could be, for instance:

empirical study detail and justify the analysis of an experiment to be run or observations to be collected

theoretical model demonstrate or disprove a property of some stochastic theory in linguistics.

A taste of R

R works like a calculator

One invokes R by double-clicking its icon, or by typing `R` at your operating system's command line prompt. From there on in, you type R *expressions* at the R prompt, a greater-than sign `>`. Expressions can be separated by either a newline or a semicolon. When you are done, issue the command `q()`. This gets you out of R and back to what you were doing before.

Expressions can use operators like `+` and `-` for addition, `^` for exponentiation. There are a variety of built-in facilities, like the constant `pi`, and functions such as `sqrt`, `exp` and `log`. To call a function, surround its argument with parentheses. Invoke the online help with `help(`*name*`)` or by prefixing the name of interest with a question mark.

```
> sqrt(2)

[1] 1.414214
```

The bracketed 1 included in R's response indicates that there was only one row of outputs needed to display the square root function's result. By contrast, if one had asked for fifteen random numbers between 0 and 1.0 the result would have taken up more space.

```
> runif(15)

 [1] 0.558328948 0.009271313 0.986948877 0.657304722 0.945634790 0.930848902
 [7] 0.891793896 0.085574300 0.907462089 0.242682589 0.116846437 0.156617086
[13] 0.390175087 0.450547917 0.135848000
```

Some functions take additional named arguments that are more like attributes.

```
> log(100, base = 10)

[1] 2
```

R has assignment

R is like those calculators that have nameable memory registers. Naming a value is called "assignment" because you are in effect re-writing the contents of cubbyhole somewhere in your computer's memory.

```
> assign("mycubbyhole", 27)
```

Once assigned, the contents of the cubbyhole can be retrieved by name.

```
> mycubbyhole + 3

[1] 30
```

The left-pointing arrow `name <- expression` is an evocative shorthand for assignment. The arrow is composed of two keyboard characters, the less-than symbol and the hyphen that together name one operator. You can make up whatever variable names you want as long as they start with a letter; case matters. Also, be careful not to choose names already in use by the R system like `q`, the quit command.

The example below assigns the numerical value of the square root of two to a symbolic name `sq2`

```
> sq2 <- sqrt(2)
```

No value is printed after an assignment; R assumes you will access the variable by name if you need the value. All the variables that have been created reside in a 'workspace'. To see which variables are defined within a workspace, use `ls()`. You can erase variables with `rm()`.

R uses vectors

Much of R's functionality is automatically threaded across components of a structured object. For instance, consider a vector of textual strings created using the concatenate function `c`.

```
> mystuff <- c("linguists", "love", "statistics")
> length(mystuff)

[1] 3

> toupper(mystuff)

[1] "LINGUISTS"  "LOVE"       "STATISTICS"
```

Just as the built-in function `toupper` converts all three components of `mystuff` into uppercase, the division and exponentiation operators apply to corresponding elements to compute the body mass index of some made-up people (example from Dalgaard page 4).

```
> weight <- c(60, 72, 57, 90, 95, 72)
> height <- c(1.75, 1.8, 1.65, 1.9, 1.74, 1.91)
> bmi <- weight/height^2
> bmi

[1] 19.59184 22.22222 20.93664 24.93075 31.37799 19.73630
```

There are special logical values `T`rue and `F`alse which are often used as intermediate steps in wrangling data. For instance, we might want to know which patients have a body mass index greater than 25kg/m^2

```
> bmi > 25

[1] FALSE FALSE FALSE FALSE  TRUE FALSE
```

Vectors having regular structure can be created using `seq` which is abbreviated to just the colon.

```
> seq(4, 9)

[1] 4 5 6 7 8 9

> seq(0, 100, 10)

 [1]   0  10  20  30  40  50  60  70  80  90 100

> uptoFive <- 1:5
> uptoFive

[1] 1 2 3 4 5

> downfromFive <- 5:1
> downfromFive

[1] 5 4 3 2 1
```

It's worth looking at section 2.3 of Venables et al. [2006] to gain proficiency with commands that create vectors with systematic, repetitious structure.

R has graphics

The course webpage depicts Zipf's law, a controversial assertion whose truth you can easily check yourself. The claim is that the frequency of a word is inversely proportional to its rank. As Mandelbrot [1965] points out, it is definitional that rank and number of attestations vary inverse *directions*. Zipf's law encompasses that stronger claim that these two quantities in fact vary in inverse *proportion*.

To make the picture on the webpage, I used the Brown corpus [Kučera and Francis, 1967] a balanced collection of English texts available through the Linguistic Data Consortium. One form of the Brown corpus comes annotated with part-of-speech information, separated from the actual English word by a slash.

```
The/DT Fulton/NNP County/NNP Grand/NNP Jury/NNP said/VBD Friday/NNP an/DT investi-
gation/NN of/IN Atlanta/NNP 's/POS recent/JJ primary/JJ election/NN produced/VBD `` /`` `
no/DT evidence/NN '' /'' '' that/IN any/DT irregularities/NNS took/VBD place/NN ./.
```

From this corpus, one can extract a sorted list of words paired with their number of attestations. Doing so immediately shows that the definite article is the most common word.

```
[colmerauer:~/classes/ling4476/week1] john% gzcat BROWN-CORPUS-TAGGED.gz | ./counts.perl > /tmp/results
[colmerauer:~/classes/ling4476/week1] john% head /tmp/results
62481 "the"
58126 ","
49125 "."
35951 "of"
27850 "and"
25650 "to"
21811 "a"
19483 "in"
10297 "that"
10045 "is"
```

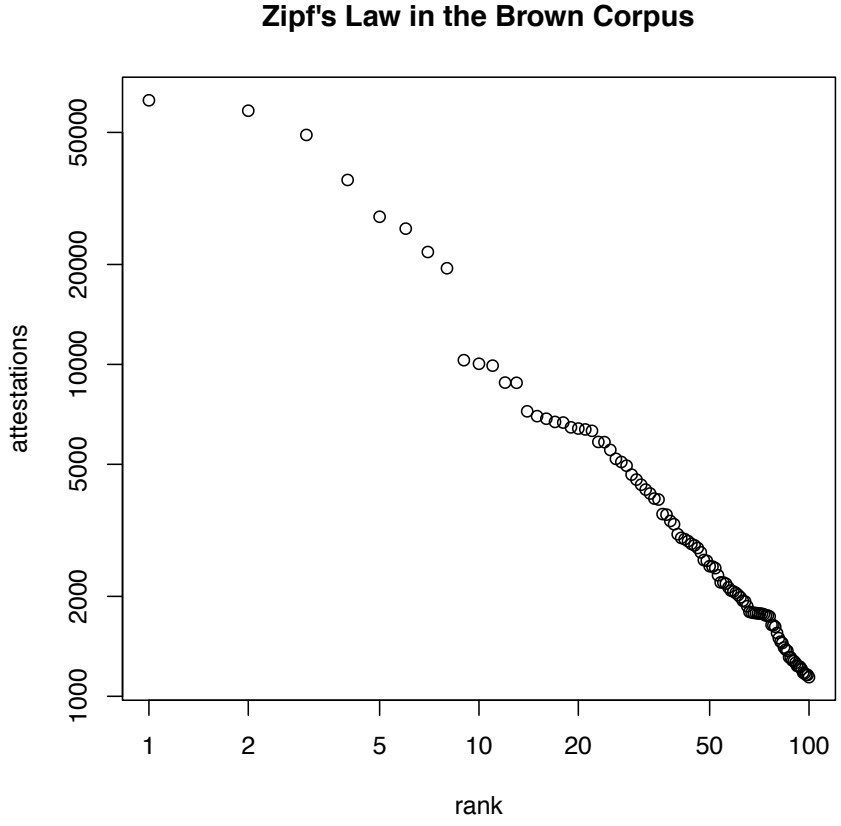
> Thus the most frequent word will occur approximately twice as often as the second most frequent word, which occurs twice as often as the fourth most frequent word, etc. For example, in the Brown Corpus "the" is the most frequently occurring word, and all by itself accounts for nearly 7% of all word occurrences. True to Zipf's Law, the second-place word "of" accounts for slightly over 3.5% of words, followed by "and". Only 135 vocabulary items are needed to account for half the Brown Corpus.
>
> http://en.wikipedia.org/wiki/Zipf's_law

This distributional information can be loaded into R with the `read.table` command.

```
> brown <- read.table("/tmp/results", header = F)
> names(brown) <- c("attestations", "word")
> head(brown)

  attestations word
1        62481  the
2        58126    ,
3        49125    .
4        35951   of
5        27850  and
6        25650   to
```

```
> plot(1:100, brown$attestations[1:100], main = "Zipf's Law in the Brown Corpus",
+     xlab = "rank", ylab = "attestations", log = "xy")
```



Measurement in linguistics

There is a distinction between an experiment and an observational study.

observational study A corpus study is typically observational; the linguist doesn't decide what people say, but merely records what is said. With this data, sometimes an association can be established e.g. repetition disfluencies tend to precede heavier constituents [Clark and Wasow, 1998].

(controlled) experiment The experimenter (randomly) assigns subjects to levels of some 'treatment' e.g. auditory or written stimuli. Then he or she measures some response — the 'dependent' variable. Typical dependent variables include number of mistakes on a test, reaction times, acceptability judgments etc.

Different kinds of data support different kinds of inference.

nominal Named properties which have no meaningful order on a scale of any type are called 'nominal'. Examples: What language is being observed? What dialect? Which word or variant, i.e. *going to* or *gonna*? What is the gender of the person being observed? There is no average fruit.

ordinal Orderable properties are called 'ordinal'. They aren't observed on a measurable scale, but this kind of property is transitive so that if $a < b$ and $b < c$ then also $a < c$. Example: Zipf's rank frequency of word-attestations, rating scales like Strongly Agree, Agree, Disagree, Strongly Disagree. The order of finishers in a cross-country race is ordinal. It doens't matter how many seconds ahead of the next runner over the finish line you were — they are still one rank behind you.

interval Properties measured on a scale that does not have a true zero value are called 'interval'. On an interval scale, the magnitude of differences of adjacent observations can be determined (unlike the adjacent items on an ordinal scale), but because the zero value on the scale is arbitrary the scale cannot be interpreted in any absolute sense. Examples: temperature (Fahrenheit or Centigrade scales), magnitude estimates such as "This soup is ten times hotter than that other soup."

ratio When measuring on a scale that does have an absolute zero value, the data are called 'ratio'. They get this name because ratios of these measurements are meaningful. For instance, a vowel that is 100 msec long is twice as long as a 50 msec vowel, and 200 msec is twice 100 msec. Contrast this with temperature where 80 degrees Fahrenheit is not twice as hot as 40 degrees. Examples: Acoustic measures like sound frequency, durations such as reaction time, distances e.g. centimeters.

Parametric tests assume that dependent variables are interval or ratio-scored. Non-parametric tests work with frequencies and rank-ordered scales.

Homework

1. Get R working on your computer. Take a look at Venables and Smith "An Introduction to R" the first document listed in the Manuals section under Documentation at `http://www.r-project.org`. Consider looking at chapter 2, as well as sections 6.3 and 7.1.

2. Download the file `cherokeeVOT.txt` from publisher's website for Keith Johnson's book. It's in the "phonetics" section, click on the Downloads tab first. Follow the directions Johnson provides on pp72–74. Write a paragraph explaining the output that you get when you type `vot$Consonant=='k'` at the R prompt.

Don't hesitate to use the R online help. The next class meeting will be a lab day to diagnose practical issues using R with linguistic data; bring your laptop if you have one.

References

Richard A. Becker, John M. Chambers, and Allan R. Wilks. *The New S language: a programming environment for data analysis and graphics*. Wadsworth and Brooks, 1988.

Herbert Clark and Thomas Wasow. Repeating words in spontanous speech. *Cognitive Psychology*, 37: 201–242, 1998.

Peter Dalgaard. *Introductory Statistics with R*. Springer, 2002.

Henry Kučera and W. Nelson Francis. *Computational Analysis of Present-day American English*. Brown University Press, 1967.

Benoit Mandelbrot. Information theory and psycholinguistics. In Benjamin B. Wolman and Ernst Nagel, editors, *Scientific psychologic: principles and approaches*, pages 550–562. Basic Books, 1965.

W.N. Venables, D.M. Smith, and the R Development Core Team. An introduction to R. Available at `http://www.r-project.org`, 2006.