Introduction to Probability

1 October 2007

1 What are probabilities?

There are two basic schools of thought as to the philosophical status of probabilities. One school of thought, the *frequentist* school, considers the probability of an event to be its asymptotic frequency over an arbitrarily large number of repeated trials. For example, to say that the probability of a toss of a fair coin landing as Heads is 0.5 (ignoring the possibility that the coin lands on its edge) means to a frequentist that if you tossed the coin many, many times, the proportion of Heads outcomes would approach 50%.

The second, *Bayesian* school of thought considers the probability of an event E to be a principled measure of the strength of one's belief that E will result. For a Bayesian, to say that $P(\text{Heads})$ for a fair coin is 0.5 (and thus equal to $P(\text{Tails})$) is to say that you believe that Heads and Tails are equally likely outcomes if you flip the coin.

The debate between these interpretations of probability rages, and we're not going to try and resolve it in this class. Fortunately, for the cases in which it makes sense to talk about both reasonable belief and asymptotic frequency, it's been proven that the two schools of thought lead to the same rules of probability. If you're further interested in this, I encourage you to read Cox (1946), a beautiful, short paper.

2 Sample Spaces

The underlying foundation of any probability distribution is the SAMPLE SPACE—a set of possible OUTCOMES, conventionally denoted Ω.For example, if you toss two coins, the event space is

$$\Omega = \{hh, ht, th, hh\}$$

where h is Heads and t is Tails. Sample spaces can be finite, countably infinite (e.g., the set of integers), or uncountably infinite (e.g., the set of real numbers).

3 Events and probability spaces

An EVENT is simply a subset of a sample space.

> What is the sample space corresponding to the roll of a single six-sided die? What is the event that the die roll comes up even?

It follows that the negation of an event E (that is, E not happening) is simply $\Omega - E$.

A PROBABILITY SPACE P on Ω is a function from events in Ω to real numbers such that the following three properties hold:

1. $P(\Omega) = 1$.
2. $P(E) \geq 0$ for all $E \subset \Omega$.
3. If E_1 and E_2 are disjoint, then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$.

4 Conditional Probability and Independence

We'll use an example to illustrate conditional independence. In Old English, the object in a transitive sentence could appear either preverbally or postverbally. Suppose that amoung transitive sentences in a corpus, the frequency distribution of object position and pronominality is as follows:

(1)

	Pronoun	**Not Pronoun**
Object **Preverbal**	0.224	0.655
Object **Postverbal**	0.014	0.107

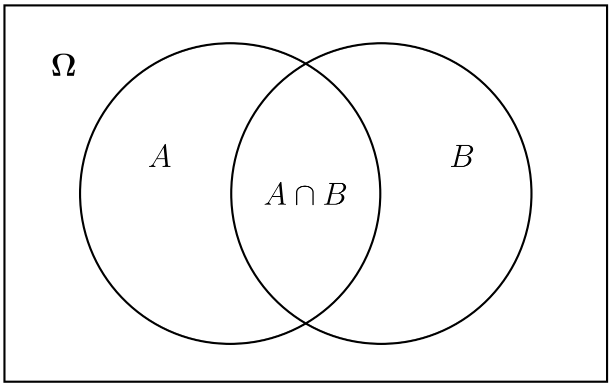


Figure 1: Conditional Probability (after Manning and Schütze (1999))

Let's interpret these frequencies as probabilities. What is the CONDITIONAL PROBABILITY of pronominality given that an object is postverbal?

The conditional probability of event B given that A has occurred/is known is defined as follows:

$$P(B|A) \triangleq \frac{P(A \cap B)}{P(A)}$$

In our case, event A is **Postverbal**, and B is **Pronoun**. The quantity $P(A \cap B)$ is already listed explicity in the lower-right cell of table (1): 0.014. We now need the quantity $P(A)$. For this we need to calculate the MARGINAL TOTAL of row 2 of Table (1): $0.014 + 0.107 = 0.121$. We can then calculate:

$$P(\mathbf{Pronoun}|\mathbf{Postverbal}) = \tfrac{0.014}{0.014+0.107} = 0.116$$

4.1 (Conditional) Independence

Events A and B are said to be CONDITIONALLY INDEPENDENT GIVEN C if

$$P(A \cap B|C) = P(A|C)P(B|C)$$

A more philosophical way of interpreting conditional independence is that if we are in the state of knowledge denoted by C, then conditional independence of A and B means that knowing A tells us nothing more about the probability of B, and vice versa. You'll also see the term we are in the state of "not knowing anything at all" ($C = \emptyset$) then we would simply say in this case that A and B are CONDITIONALLY INDEPENDENT.

It's crucial to keep in mind that if A and B are conditionally independent given C, that does not guarantee they will be conditionally independent given some other set of knowledge C'.

5 Random Variables

Technically, a RANDOM VARIABLE X is a function from Ω to the set of real numbers ($\mathbb{R}$). You can think of a random variable as an "experiment" whose outcome is not known in advance. In fact, the OUTCOME of a random variable is a technical term simply meaning which number resulted from the "experiment".

The relationship between the sample space Ω, a probability space P on Ω, and a random variable X on Ω can be a bit subtle so I'll explain it intuitively, and also with an example. In many cases you can think of a random variable as a "partitioning" of the sample space into the distinct classes of events that you (as a researcher, or as a person in everyday life) care about. For example, suppose you are trying to determine whether a particular coin is fair. A natural thing to do is to flip it many times and see how many times it comes out heads. Suppose you decide to flip it eight times. The sample space Ω is then all possible sequences of length eight whose members are either H or T. The coin being fair corresponds to the probability space P in which each point in the sample space has equal probability $\frac{1}{2^8}$. Now suppose you go ahead and flip the coin eight times, and the outcome is

TTTTTTHT

Intuitively, this is a surprising result. But under P, all points in Ω are equiprobable, so there is nothing about the result that is particularly surprising.

The key here is that you as an investigator of the coin's fairness are not interested in the particular H/T sequence that resulted. You are interested in *how many of the tosses came up heads.* This quantity is your random variable of interest—let's call it X. The logically possible outcomes of X are the integers $\{0, 1, \cdots, 8\}$. The actual outcome was $X = 1$—and there were seven other possible points in Ω for which $X = 1$ would be the outcome! We can use this to calculate the probability of this outcome of our random variable under the hypothesis that the coin is fair:

$$P(X = 1) = \frac{1}{2^8} \times 8 = \frac{1}{2^5} = 0.03125$$

So seven tails out of eight is a pretty surprising result. Incidentally, there's a very interesting recent paper (Griffiths and Tenenbaum, 2007) that deals with how humans actually *do* ascribe surprise to a "rare" event like a long sequence of heads in a coin flip.

6 Basic data visualization

We'll illustrate two basic types of data visualization—histograms and plots—by briefly investigating the relationship between word lengths and word frequencies.

6.1 Histograms

The file `brown-counts-lengths` contains the length and frequency of every word type appearing in the parsed Brown corpus. We'll start by visualizing the distributions of frequency and length counts.

```
# header=T reads off the
> x <- read.table("brown-counts-lengths",header=T)
> head(x,n=20)
   Count        Word Length
1    163           '      1
2     58           $      1
3     24           &      1
4      4           %      1
5      1           0      1
6      1        0600      4
7     27           1      1
8     16          10      2
9      8         100      3
10     1        1000      4
11     3       1,000      5
12     4      10,000      6
13     3     100,000      7
```

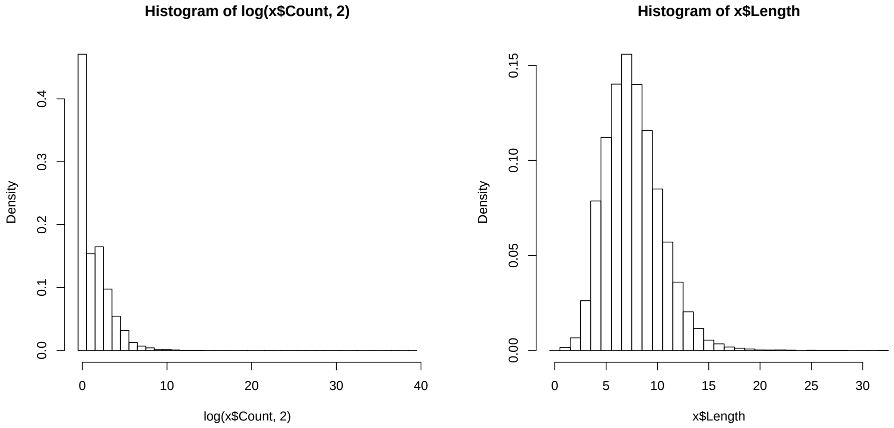


Figure 2: Histograms of log-frequency counts and word lengths for Brown corpus

```
14     1   1,000,000      9
15     1  10,000,000     10
16     1        1020      4
17     1 105-degrees     11
18     1         106      3
19     1         108      3
20     1      10-day      6
> hist(log(x$Count,2),breaks=seq(-0.5,39.5,by=1),prob=T)
> hist(x$Length,breaks=seq(-0.5,32.5,by=1),prob=T)
# results shown in Figure
```

Most of the word types (nearly 50%) are frequency 1—a word with a single instance in a given corpus is sometimes called a *hapax legomenon.* We can look up the ten most frequent words in the Brown corpus using the `order()` command:

```
> x[order(x$Count,decreasing=T)[1:10],]
      Count Word Length
26280 22244  the      3
17904 10964   of      2
1264  10661  and      3
26681  9778   to      2
```

```
337    8960    a      1
13265  6575   in      2
28599  5499  was      3
12187  4195   he      2
13001  4131    I      1
26270  4079 that      4
```

The longest word in the Brown corpus is:

```
> x[order(x$Length,decreasing=T)[1],]
      Count                            Word Length
17480     1 nnuolapertar-it-vuh-karti-birifw     32
```

Not surprisingly, this is *hapax legomenon*.

6.2 Plots

We can investigate the *relationship* between length and frequency by plotting them against each other:

```
> plot(x$Length,log(x$Count,2))
> lines(lowess(x$Length, log(x$Count,2)))
# results in Figure
```

The `lowess()` function is a kind of smoother that estimates the y-value of a function for a value of y, given a full dataset of (x, y) points. You can use it to graphically explore the relationship between two variables on a relatively informal basis.

Finally, you can use `identify()` to interactively inspect the points on a plot.

```
> ?plot
> plot(x$Length,log(x$Count,2))
> identify(x$Length,log(x$Count,2),labels=x$Word)
[1] 17480 24298 26270 26280
```

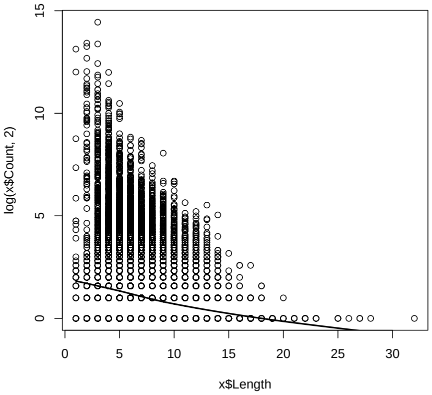


Figure 3: Word length versus log-frequency in the Brown corpus

References

Cox, R. T. (1946). Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13.

Griffiths, T. L. and Tenenbaum, J. B. (2007). From mere coincidences to meaningful discoveries. *Cognition*, 103(2):180–226.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Lecture 2: Counting methods, binomial, geometric and Poisson distributions, CDFs and quantile functions, comparing probability distributions

October 8, 2007

In this lecture we'll learn the following:

1. Why, mathematically, is a fair coin more likely to come up heads roughly half the time than almost never?
2. the binomial, geometric, and Poisson distributions
3. probability density functions, cumulative distribution functions, and quantiles

1 Counting methods

We encountered counting methods briefly in lecture 1. When we flipped a fair coin eight times, we needed to calculate the probability of obtaining exactly one heads if the coin is fair. We talked about this in the context of a random variable X mapping from the set of possible results of 8 coin flips (we'll call this $\{h/t\}^8$) to the integers $\{0, 1, \cdots, 8\}$. Therefore we wanted the quantity $P(X = 1)$. The simple solution to calculating this quantity was that there are eight different points in the sample space that X maps to 1, and each point had probability $\frac{1}{2^8}$. The answer was thus $\frac{8}{2^8}$.

This is a simple example of a *counting method* for calculating the probability of an event, and it is useful for countable and particularly finite sample spaces. That is, to calculate the probability of an event A, just sum the probabilities of each atomic subevent in A:[1]

$$P(A) = \sum_{a \in A} P(\{a\})$$

Conrrespondingly, we have

$$P(X = 1) = \sum_{a : X(\{a\}) = 1} P(\{a\})$$

Counting methods are particularly useful in two similar cases:

1. When all the points in the sample space that map to a given value of a random variable have the same probability p^*. In these cases, all you need to do to calculate $P(X = x)$ is calculate the number of points that X maps to X:[2]

$$P(X = x) = p^* |\{a : X(\{a\}) = 1\}|$$

2. When the sample space is finite and all points in it have equal probability. Then you just need to calculate the size of the sample space and the number of points that the random variable maps to the relevant value:

$$P(X = x) = \frac{|\{a : X(\{a\}) = 1\}|}{|\Omega|}$$

Note that case (2) is a special case of (1).

1.1 Sampling with replacement

Sometimes we need to calculate the probability of obtaining a certain collection of items when drawing N items from a larger pool. These probabilities come up often in simple models of texts called "bag of words" models. We

[1] Note that we are being a bit sloppy here and equating the "probability of a point a in Ω" with the probability of the event $\{a\}$. But I think that's excusible.

[2] Read $|\{a : X(\{a\}) = 1\}|$ as "the cardinality of (number of elements contained in) the set of all points that X maps to 1".

can model the generative process of collections like this as drawing K items, one after another, from a "bag" (more classicly, an *urn*) that contains many different items in it. A crucial choice point is whether drawing item w_i at pick j affects the probability of drawing w_i again at pick $j+1$. In the simplest models, there is no such interdependence; this is called **sampling with replacement**, as if you when you drew w_i, you marked its identity and then put it back in the bag.

Calculating the probability of getting R heads in N coin tosses is this kind of situation. This falls in the counting method of type (1) above. This is also known as the question of how many **combinations** of R objects can be selected from a collection of N objects? (In this case, the N objects are coin flips 1 through 8.) It turns out that this quantity, which is denoted $\binom{N}{R}$, is equal to:[3]

$$\binom{N}{R} = \frac{N!}{R!(N-R)!}$$

For a given N, this quantity is larger the closer R is to $\frac{N}{2}$ (check it yourself!), which is the reason why a fair coin is more likely to come up heads roughly half the time than almost never.

In language, we will often be interested in how many ways N items can be arranged into K classes of sizes $R_1, \cdots, R_K$ (that is, so $N = \sum_{k=1}^{K} R_k$). This quantity is sometimes denoted $\binom{N}{R_1, \cdots, R_K}$ and is equal to

$$\binom{N}{R_1, \cdots, R_K} = \frac{N!}{R_1! \cdots R_K!}$$

For example, if we have a sequence of 9 items and we want know how many ways it could be the case that 4 are nouns, 2 are determiners, 2 are verbs, and one is the word *and*, we would have

$$\frac{9!}{4!2!2!1!} = \frac{9 \times 8 \times 7 \times 6 \times 5 \times \cancel{4} \times \cancel{3} \times \cancel{2} \times \cancel{1}}{(\cancel{4} \times \cancel{3} \times \cancel{2} \times \cancel{1})(2 \times 1)(2 \times 1)(1)}$$

[3] $N!$ is the number of ways in which a collection of N items can be arranged in a line—that is, the number of **permutations** of N objects—and is equal to

$$N \times (N-1) \times \cdots \times 2 \times 1$$

$$
\begin{aligned}
&= \frac{9 \times \cancel{8}2 \times 7 \times 6 \times 5}{(\cancel{2} \times 1)(\cancel{2} \times 1)} \\
&= 9 \times 2 \times 7 \times 6 \times 5 \\
&= 3780
\end{aligned}
$$

Now think about how many ways it's possible to order these words within the constraints of English syntax!

2 The binomial distribution

The foregoing section has set the stage for our first probability distribution, the **binomial distribution**. In statistics, the term PROBABILITY DISTRIBUTION is generally used to describe a family of probability spaces (see lecture 1!) that can be succinctly described with a few parameters. One of the problems at the heart of statistics is how to infer what parameters for a given probability distribution are best justified by a given set of data.

You can think of the binomial distribution as the repeated flipping of a coin that is not necessarily fair. It is characterized by two parameters, n and p. The n parameter is how many coin flips. The p parameter is the probability that any given coin flip comes up heads. However, instead of "heads" and "tails" the conventional terminology used is SUCCESS (with probability p) and FAILURE (with probability $1-p$). The random variable X for the binomial distribution with parameters n, p is the number of successes.

For a binomial distribution with parameters n, p we can use the counting method of type (1) above to calculate the probability $P(X = r)$ for $r \in \{0, \cdots, n\}$. First, there are $\binom{n}{r}$ ways of getting r successes. Now, what is the probability of each such point in the sample space? Each success has probability p, each failure has probability $1-p$. Therefore each point in the sample space mapped by X to r has probability $p^r(1-p)^{n-r}$. This gives us the fundamental function, called the PROBABILITY DENSITY FUNCTION, for a binomially-distributed random variable X with parameters n, p:[4] Baayen page 51

(1)

$$P(X = r; n, p) = \binom{n}{r} p^r (1-p)^{n-r}$$

[4] Read $P(X = r; n, p)$ as "probability of X being r for the parameter values n and p".

Sometimes we will use abbreviated forms for $P(X = r; n, p)$ and simply say $P(X = r)$, $P(r; n, p)$, or even $P(r)$.

3 Probability density, cumulative distribution, and quantile functions

This is a good time to go over some more fundamental ideas in probability. The probability density function given for the binomial distribution in (1) above assigns a DENSITY to the real numbers. We can use the `dbinom()` function in `R` to plot an example of this density function for the previous example of 8 coin flips:

```
> n <- 8
> p <- 0.5
> # set up new plot but don't put any points down
> plot(c(),c(),xlim=c(0,n),ylim=c(0,0.2),xlab="r",ylab="binomial pdf")
> for(r in 0:n) {
>   xs <- c(r,r)
>   ys <- c(0,dbinom(r,n,p))
>   lines(xs,ys) # draws a vertical line from (r,0)
>                # as high as the density at r
> }
```

The result, for both $p = 0.5$ and $p = 0.8$, is shown in Figure 1.

More often than not, however, a fundamental quantity of interest is not how likely a particular outcome is, but how likely an outcome is to fall in a *region* of the real number line. For example, let's revisit the question of how to tell whether a coin is fair. We'll simplify the picture by assuming that the coin is either fair or weighted toward tails, but is definitely not weighted toward heads. Now we flip it n times and get r successes (i.e., heads), and want to know how surprised we'd be if the coin is fair. Suppose we flip it 20 times and get 7 heads. We can find out the probability density of this result: `dbinom(7,20,0.5)=0.0739`. This seems like a pretty unlikely event. On the other hand, the probability density of getting 10 heads isn't that high either: `dbinom(10,20,0.5)=0.176`. The key here is that we're not really that interested in getting *exactly* 7 heads. A quantity of considerably more interest would be the probability of getting *at most* 7 heads, that is:

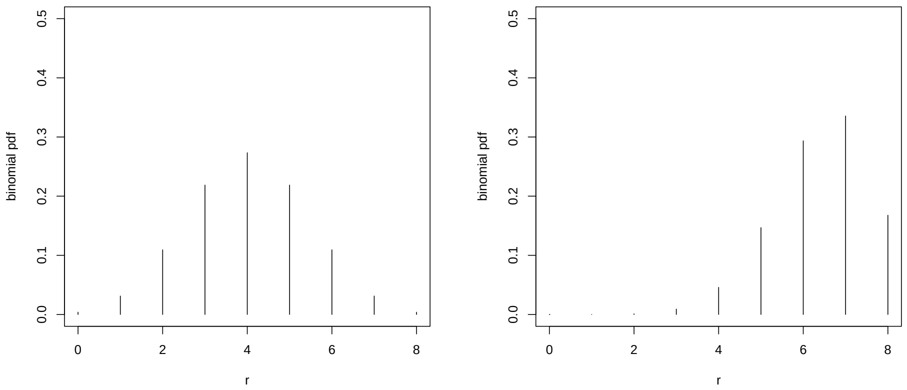


Figure 1: The probability density function for the binomial distribution with $n = 8$ and $p = 0.5$ (left) or $p = 0.8$ (right)

$P(r \leq 7)$. This is called the CUMULATIVE DISTRIBUTION FUNCTION and can be calculated for the binomial distribution using the `pbinom()` function:

Baayen pp. 53, 55

```
> pbinom(7,20,0.5)
[1] 0.1315880
> pbinom(10,20,0.5)
[1] 0.5880985
```

Here we can see a big difference between the two outcomes.

We can plot the cumulative density function for $n = 20, p.0 = 5$, which appears in the right side of Figure 2, using the following code:

```
n <- 20
p <- 0.5
# set up new plot but don't put any points down
plot(c(),c(),xlim=c(0,n),ylim=c(0,1),xlab="r",ylab="binomial cdf")
for(r in 0:n) {
  xs <- c(r,r)
  ys <- c(0,pbinom(r,n,p))
  lines(xs,ys) # draws a vertical line from (r,0)
               # as high as the density at r
}
abline(0.05,0,lty=2)
```

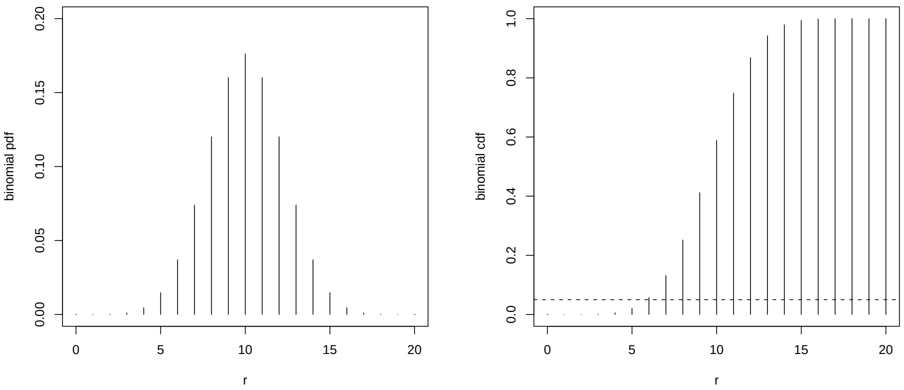


Figure 2: Probability density function (pdf) and cumulative distribution function (cdf) for $n = 20, p = 0.5$

The dashed line indicates the threshold for which the region $[0, r]$ contains more than 5% of the probability mass. We can see that we'd need 5 or less successes to be in this result. We'll foreshadow the hypothesis testing part of the course by saying that we'd need 5 or fewer successes on 20 coin flips to achieve a p-value of 0.05 on a ONE-TAILED TEST of the hypothesis that the coin is fair. We could also calculate how many successes we'd need by using the *inverse* of the CDF: the QUANTILE function, which we can access by `qbinom()`:

Baayen p. 55

```
> qbinom(0.05,20,0.5)
[1] 6
```

The output is 6 because the region $[0, 6]$ contains *at least* a probability mass of 0.05.

3.1 Quantile-quantile plots

One of the key parts of practicing statistics is determining whether a particular probability distribution (*family* of probability functions) represents a given dataset reasonably well. One useful tool in this enterprise is the QUANTILE-QUANTILE PLOT, or Q-Q plot. A Q-Q plot is constructed by taking values ranging between 0 and 1 and mapping them to $\langle x, y \rangle$ points where

x is the theoretical real-valued outcome of the random variable mapped to by the relevant quantile function (e.g., `qbinom()`), and y is the corresponding empirical quantile (accessible via `R`'s `quantile()` function). We'll first do a simple example and then unpack Baayen's example. Let's choose a binomial distribution with parameters $50, 0.3$. We artificially generate 1000 data points using `rbinom()` as follows:

```
n <- 50
p <- 0.3
dat <- rbinom(1000,n,p)
```

The x-axis of a Q-Q plot is the theoretical quantiles of the distribution, and the y-axis is the quantiles of the empirical data. We generate $\langle x, y\rangle$ points as described above:

```
quants <- seq(0.01,0.99,by=0.01)
x <- qbinom(quants,n,p)
y <- quantile(dat, probs=quants)
plot(x,y,xlab="theoretical",ylab="sampled",xlim=c(0,30),ylim=c(0,30))
abline(0,1)
```

The result is shown in Figure 3. This is a good-looking Q-Q plot.

Now let's take a look at the one from Baayen. It's easier to interpret Q-Q plots if you scale the x and y axes identially, and draw the ideal-distribution line explicitly:

Baayen, p. 56

```
n <- 1000
p <- mean(havelaar$Frequency / n)
plot(qbinom(quants,n,p), quantile(havelaar$Frequency,quants),xlim=c(3,35),ylim=c(3
abline(0,1)
```

4 The geometric distribution

Let's return to coin flipping, but use a different process to generate a sequence of coin flips. Suppose I start flipping a coin, and every time it comes up tails I keep on flipping, but the first time it comes up heads I stop. The random variable in this case is the length of the sequence of coin flips. The

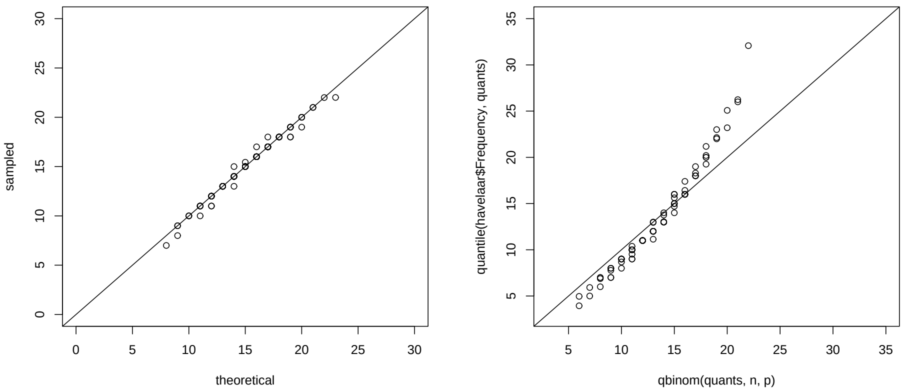


Figure 3: Theoretical and empirical case-study Q-Q plot

Figure 4: Distribution of article lengths—empirical and MLE geometric estimate—in Wall Street Journal, 1987–1989

geometric distribution characterizes the probability density on this random variable, with the single parameter p, the probability of a success in a single coin flip:

$$P(k;p) = (1-p)^{k-1}p, k \in 1, \cdots$$

The geometric distribution is of interest in the study of language for things like the length of words or texts. For example, Figure 4 shows the distribution of article lengths (measured by number of sentences) in a collection of Wall Street Journal articles from 1987 through 1989. The points superimposed on the graph are the predicted frequencies for a geometric distribution with p set to the value that maximizes the joint probability of all the observed article lengths. This way of choosing the parameter value $\hat{p}$ is known as the maximum likelihood estimate (MLE), and for this collection of articles the MLE is $\hat{p} = 0.04785$. We'll revisit the MLE later in the course.

Maximum Likelihood Estimate

Figure 4 is a somewhat reasonable fit, though there are more fairly short articles and fewer medium-length articles than the geometric distribution would predict.

There is a generalization of the geometric distribution called the negative binomial distribution parameterized by p plus a second parameter r, in which you stop after r successes.

5 The Poisson distribution

The Poisson distribution is the last important discrete distribution we'll look at. It is a bit more work to motivate the Poisson distribution from first principles: the distribution is an approximation of the binomial distribution for large n and small p, using a single parameter λ, which can be interpreted as the mean number of successes or np. The single parameter is λ and the resulting Poisson distribution is:

$$P(X = k; \lambda) = \frac{\lambda^k}{k!} e^{-\lambda}, k \in \{0, 1, \cdots\}$$

This formula looks more intimidating, so let's try to unpack it a bit. First, note that the $e^{-\lambda}$ term is constant for different values of k. You can think of this term as a **normalizing constant** C that ensures the result is a valid probability distribution (i.e., sums to 1).

Second, if we substitute in $\lambda = np$ we get

$$P(X = k; \lambda) = \frac{n^k p^k}{k!} e^{-\lambda}$$

Because n is large and p is small, we can make the approximations $1 - p \approx 1$ and $(n - k)! \approx n^k$. This gives us

$$P(X = k; \lambda) \approx \frac{(n-k)!}{k!} p^k (1-p)^{n-k} C$$

which indeed is the binomial distribution for n, p.

We can compare two different situations where the Poisson approximates the binomial:

```
# n = 50, p=0.3
n <- 20
p <- 0.4
x <- 1:50
Plot(x,dbinom(x,n,p),type="b",pch=19)
points(x,dpois(x,n*p),pch=21)
lines(x,dpois(x,n*p),lty=2)
# then try with n=500,p=0.02
```

References

Lecture 3: Continuous distributions, expected value & mean, variance, the normal distribution

8 October 2007

In this lecture we'll learn the following:

1. how continuous probability distributions differ from discrete
2. the concepts of expected value and variance
3. the normal distribution

1 Continuous probability distributions

Continuous probability distributions (CPDs) are those over random variables whose values can fall anywhere in one or more continua on the real number line. For example, the amount of time that an infant has lived before it hears a parasitic gap in its native-language environment would be naturally modeled as a continuous probability distribution.

With discrete probability distributions, the probability density function (PDF, often called the PROBABILITY MASS FUNCTION for discrete random variables) assigned a non-zero probability to points in the sample space. That is, for such a probability space you could "put your finger" on a point in the sample space and there would be non-zero probability that the outcome of the r.v. would actually fall on that point. For cpd's, this doesn't make any sense. Instead, the pdf is a true *density* in this case, in the same way as a point on a physical object in pre-atomic physics doesn't have any mass, only density—only *volumes* of space have mass.

As a result, the cumulative distribution function (CDF; $P(X \leq x)$ is of primary interest for cpd's, and its relationship with the probability density function $p(x)$ is defined through an integral:

$$P(X \leq x) = \int_{-\infty}^{x} p(x)$$

Note that the notation $f(x)$ is often used instead of $p(x)$.

We then become interested in the probability that the outcome of an r.v. will fall into a *region* $[x, y]$ of the real number line, and this is defined as:

$$\begin{aligned} P(x \leq X \leq y) &= \int_{x}^{y} p(x) \\ &= P(X \leq y) - P(X \leq x) \end{aligned}$$

2 The uniform distribution

The simplest cpd is the UNIFORM DISTRIBUTION, defined over a bounded region $[a, b]$ within which the density function $f(x)$ is a constant value $\frac{1}{b-a}$.

3 Expected values and variance

We now turn to two fundamental quantities of probability distributions: EXPECTED VALUE and VARIANCE.

3.1 Expected value

The expected value of a random variable X, denoted $E(X)$ or $E[X]$, is also known as the MEAN. For a discrete random variable X under probability distribution P, it's defined as

$$E(X) = \sum_{i} x_i P(x_i)$$

For a continuous random variable X under cpd p, it's defined as

$$E(X) = \int_{-\infty}^{\infty} x \; p(x) dx$$

> What is the mean of a binomially-distributed r.v. with parameters n, p? What about a uniformly-distributed r.v. on $[a, b]$?

Sometimes the expected value is denoted by the Greek letter μ, "mu".

3.1.1 Linearity of the expectation

Linearity of the expectation can expressed in two parts. First, if you *rescale* a random variable, its expectation rescales in the exact same way. Mathematically, if $Y = a + bX$, then $E(Y) = a + bE(X)$.

Second, the expectation of the sum of random variables is the sum of the expectations. That is, if $Y = \sum_i X_i$, then $E(Y) = \sum_i E(X_i)$.

We can put together these two pieces to express the expectation of a linear combination of random variables. If $Y = a + \sum_i b_i X_i$, then

$$E(Y) = a + \sum_i b_i E(X_i)$$

This is incredibly convenient. For example, it is intuitively obvious that the mean of a binomially distributed r.v. Y with parameters n, p is pn. However, it takes some work to show this explicitly by summing over the possible outcomes of Y and their probabilities. On the other hand, Y can be re-expressed as the sum of n BERNOULLI RANDOM VARIABLES X_i. (A Bernoulli random variable is a single coin toss with probability of success p.) The mean of each X_i is trivially p, so we have:

$$E(Y) = \sum_i^n E(X_i) \sum_i^n p = pn$$

3.2 Variance

The variance is a measure of how broadly distributed the r.v. tends to be. It's defined in terms of the expected value:

$$\mathrm{Var}(X) = E[(X - E(X))^2]$$

The variance is often denoted σ^2 and its positive square root, σ, is known as the STANDARD DEVIATION.

As an exercise, we can calculate the variance of a Bernoulli random variable X with parameter p—we already saw that its expectation is p:

$$
\begin{aligned}
\mathrm{Var}(X) &\triangleq E[(X - E(X))^2] && (1)\\
&= E[(X - p)^2] && (2)\\
&= \sum_{x \in \{0,1\}} (x - p)^2 P(x) && (3)\\
&= (0 - p)^2(1 - p) + (1 - p)^2 p && (4)\\
&= p^2(1 - p) + p(1 - p)^2 && (5)\\
&= p(1 - p) && (6)
\end{aligned}
$$

Interestingly, the variance is 0 when p is 0 or 1, and is maximized when $p = \frac{1}{2}$. This foreshadows the use of generalized linear models, particularly logistic regression, instead of straightforward linear regression/ANOVA for statistical modeling of categorical data. We'll cover this in week 6.

As an aside, note that expected values are first-order terms (the r.v. enters into the sum/integral as itself) and that variances are second-order terms (the r.v. enters in its square). There are also higher-order terms of interest, notably *skewness* (the third-order term) and *kurtosis* (the fourth-order term). Skewness is a measure of the left-right asymmetry of a distribution, and kurtosis is a measure of the "peakedness" (independent of the variance). I don't think we'll really deal with these higher-order terms in the class.

4 The normal distribution

The NORMAL DISTRIBUTION is almost certainly the most common cpd you'll encounter. It is defined entirely in terms of its expected value μ and variance σ^2, and is characterized by an ugly-looking density function, denoted:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Normal distributions with different variances:

```
x <- seq(-10,10,by=0.01)
plot(x,dnorm(x,0,2),type="l")
lines(x,dnorm(x,0,5),lty=2)
```

There's something very important called the CENTRAL LIMIT THEOREM which states that when a large quantity of independent r.v.'s is added together, its sum approaches a normal distribution. We'll see this in lecture 4 by modeling a binomially distributed r.v. with a normal distribution.

Lecture 4: Joint probability distributions; covariance; correlation

10 October 2007

In this lecture we'll learn the following:

1. what joint probability distributions are;
2. visualizing multiple variables/joint probability distributions;
3. marginalization;
4. what covariariance and correlation are;
5. a bit more about variance.

1 Joint probability distributions

Recall that a basic probability distribution is defined over a random variable, and a random variable maps from the sample space to the real numbers ($\mathbb{R}$). What about when you are interested in the outcome of an event that is not naturally characterizable as a single real-valued number, such as the two formants of a vowel?

The answer is really quite simple: probability distributions can be generalized over multiple random variables at once, in which case they are called JOINT PROBABILITY DISTRIBUTIONS (jpd's). If a jpd is over N random variables at once then it maps from the sample space to $\mathbb{R}^N$, which is short-hand for real-valued VECTORS of dimension N. Notationally, for random variables $X_1, X_2, \cdots, X_N$, the joint probability density function is written as

$$p(X_1 = x_1, X_2 = x_2, \cdots, X_N = x_n)$$

or simply

$$p(x_1, x_2, \cdots, x_n)$$

for short.

Whereas for a single r.v., the cumulative distribution function is used to indicate the probability of the outcome falling on a segment of the real number line, the JOINT CUMULATIVE PROBABILITY DISTRIBUTION function indicates the probability of the outcome falling in a region of N-dimensional space. The joint cpd, which is sometimes notated as $F(x_1, \cdots, x_n)$ is defined as the probability of the set of random variables all falling at or below the specified values of X_i:[1]

$$F(x_1, \cdots, x_n) \triangleq P(X_1 \leq x_1, \cdots, X_N \leq x_n)$$

The natural thing to do is to use the joint cpd to describe the probabilities of rectangular volumes. For example, suppose X is the f_1 formant and Y is the f_2 formant of a given utterance of a vowel. The probability that the vowel will lie in the region $480\text{Hz} \leq f_1 530\text{Hz}, 940\text{Hz} \leq f_2 \leq 1020\text{Hz}$ is given below:

$$\begin{aligned} &P(480\text{Hz} \leq f_1 530\text{Hz}, 940\text{Hz} \leq f_2 \leq 1020\text{Hz}) = \\ &\quad F(530\text{Hz}, 1020\text{Hz}) - F(530\text{Hz}, 940\text{Hz}) - F(480\text{Hz}, 1020\text{Hz}) + F(480\text{Hz}, 940\text{Hz}) \end{aligned}$$

and visualized in Figure 1 using the code below.

[1] Technically, the definition of the multivariate cpd is then

$$F(x_1, \cdots, x_n) \triangleq P(X_1 \leq x_1, \cdots, X_N \leq x_n) = \sum_{\vec{x} \leq \langle x_1, \cdots, x_N \rangle} p(\vec{x}) \quad \text{[Discrete]} \tag{1}$$

$$F(x_1, \cdots, x_n) \triangleq P(X_1 \leq x_1, \cdots, X_N \leq x_n) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_N} p(\vec{x}) dx_N \cdots dx_1 \quad \text{[Continuous]} \tag{2}$$

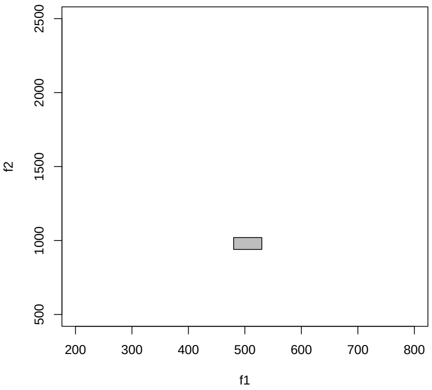


Figure 1: The probability of the formants of a vowel landing in the grey rectangle can be calculated using the joint cumulative distribution function.

```
plot(c(),c(),xlim=c(200,800),ylim=c(500,2500),xlab="f1",ylab="f2")
rect(480,940,530,1020,col=8)
```

1.1 Multinomial distributions as jpd's

We touched on the multinomial distribution very briefly in a previous lecture. To refresh your memory, a multinomial can be thought of as a random set of n outcomes into r distinct classes—that is, as n rolls of an r-sided die where the tabulated outcome is the number of rolls that came up in each of the r classes. This outcome can be represented as a set of random variables $X_1, \cdots, X_r$, or equivalently an r-dimensional real-valued vector. The r-class multinomial distribution is characterized by $r-1$ parameters, $p_1, p_2, \cdots p_{r-1}$, which are the probabilities of each die roll coming out as each class. The probability of the die roll coming out in the rth class is $1 - \sum_{i=1}^{r-1} p_i$, which is sometimes called p_r but is not a true parameter of the model. The probability mass function looks like this:

$$p(n_1, \cdots, n_r) = \binom{n}{n_1 \cdots n_r} \prod_{i=1}^{r} p_i$$

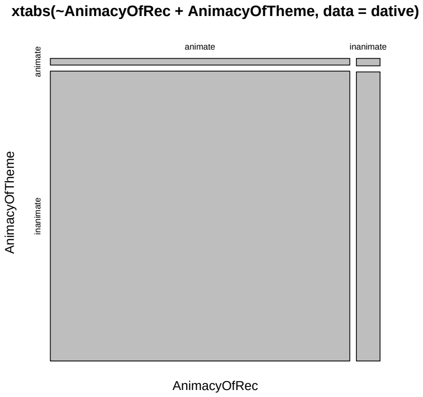


Figure 2: Distribution of animacy for recipient and theme

2 Visualizing joint probability distributions

We'll look at two examples of visiualizing jpd's. First is a discrete distribution one out of the `dative` dataset in `languageR`. We'll look at the distribution of animacy of theme and recipient in this dataset, shown in Figure 2 using the code below:

`xtabs()`, `mosaicplot()`

```
> xtabs(~ AnimacyOfRec + AnimacyOfTheme, data=dative)
            AnimacyOfTheme
AnimacyOfRec animate inanimate
   animate        68      2956
   inanimate       6       233
> mosaicplot(xtabs(~ AnimacyOfRec + AnimacyOfTheme, data=dative))
```

The second example is of the joint distribution of frequency and length in the Brown corpus. First we'll put together an estimate of the joint distribution.

`persp()`, `kde2d()`

```
persp(kde2d(x$Length,log(x$Count),n=50),theta=210,phi=15,
  xlab="Word Length",ylab="Word Frequency",zlab="p(Length,Freq)")
```

This gives a perspective plot of the joint distribution of length and frequency. Another way of visualizing the joint distribution is going along the length-axis slice by slice in a COPLOT:

`histogram()`

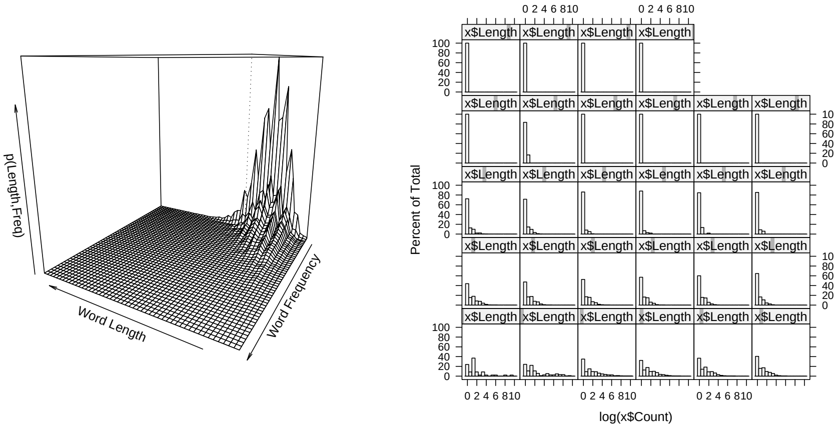


Figure 3: Perspective plot and coplot (i.e., slice-by-slice histogram plot) of the relationship between length and log-frequency in the Brown corpus

```
histogram(~ log(x$Count) | x$Length)
```

Both resulting plots are shown in Figure 3.

3 Marginalization

Often we have direct access to a joint density function but we are more interested in the probability of an outcome of a subset of the random variables in the joint density. Obtaining this probability is called MARGINALIZATION, and it involves taking a weighted sum[2] over the possible outcomes of the r.v.'s that are not of interest. For two variables X, Y:

$$
\begin{aligned}
P(X = x) &= \sum_y P(x, y) \\
&= \sum_y P(X = x|Y = y)P(y)
\end{aligned}
$$

In this case $P(X)$ is often called a *marginal probability* and the process of calculating it from the joint density $P(X, Y)$ is known as *marginalization*.

[2] or integral in the continuous case

4 Covariance

The COVARIANCE between two random variables X and Y is defined as follows:

$$Cov(X,Y) = E[(X - E(X))(Y - E(Y))]$$

Simple example:

(1)

		Coding for Y		
		0	1	
Coding for X		**Pronoun**	**Not Pronoun**	
0	Object **Preverbal**	0.224	0.655	.879
1	Object **Postverbal**	0.014	0.107	.121
		.238	.762	

Each of X and Y can be treated as a Bernoulli random variable with arbitrary codings of 1 for **Postverbal** and **Not Pronoun**, and 0 for the others. As a resunt, we have $\mu_X = 0.121$, $\mu_Y = 0.762$. The covariance between the two is:

$$\begin{array}{ll}
(0 - .121) \times (0 - .762) \times .224 & (0,0) \\
+(1 - .121) \times (0 - .762) \times 0.014 & (1,0) \\
+(0 - .121) \times (1 - .762) \times 0.0655 & (0,1) \\
+(1 - .121) \times (1 - .762) \times 0.107 & (1,1) \\
=0.0148 &
\end{array}$$

In R, we can use the `cov()` function to get the covariance between two random variables, such as word length versus frequency across the English lexicon:

```
> cov(x$Length,x$Count)
[1] -42.44823
> cov(x$Length,log(x$Count))
[1] -0.9333345
```

The covariance in both cases is *negative*, indicating that longer words tend to be less frequent. If we shuffle one of the covariates around, it eliminates this covariance:

`order()` plus `runif()` give a nice way of randomizing a vector.

```
> cov(x$Length,log(x$Count)[order(runif(length(x$Count)))])
[1] 0.006211629
```

The covariance is essentially zero now.

An important aside: the variance of a random variable X is just its covariance with itself:

$$\text{Var}(X) = \text{Cov}(X, X)$$

4.1 Covariance and scaling random variables

What happens to $Cov(X, Y)$ when you scale X? Let $Z = a + bX$. It turns out that the covariance with Y increases by b:[3]

$$\text{Cov}(Z, Y) = b\text{Cov}(X, Y)$$

As an important consequence of this, rescaling a random variable by $Z = a + bX$ rescales the variance by b^2: $\text{Var}(Z) = b^2\text{Var}(X)$.

4.2 Correlation

We just saw that the covariance of word length with frequency was much higher than with log frequency. However, the covariance cannot be compared directly across different pairs of random variables, because we also saw that random variables on different scales (e.g., those with larger versus smaller ranges) have different covariances due to the scale. For this reason, it is commmon to use the CORRELATION ρ as a standardized form of covariance:

[3] The reason for this is as follows. By linearity of expectation, $E(Z) = a + bE(X)$. This gives us

$$\begin{aligned}
Cov(Z, Y) &= E[(Z - a + bE(X))(Y - E(Y))] \\
&= E[((bX - bE(X))(Y - E(Y))] \\
&= E[b(X - E(X))(Y - E(Y))] \\
&= bE[(X - E(X))(Y - E(Y))] && \text{[by linearity of expectation]} \\
&= bCov(X, Y) && \text{[by linearity of expectation]}
\end{aligned}$$

$$\rho_{XY} = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}$$

If X and Y are independent, then their covariance (and hence correlation) is zero.

4.3 Multivariate normal distributions

We're now ready to deal with the multivariate normal distribution. Here we'll just work with a 2-dimensional, or bivariate, distribution. Whereas the univariate normal distribution was characterized by two parameters—mean μ and variance σ^2—the bivariate normal distribution is characterized by two mean parameters (μ_X, μ_Y), two variance terms (one for the X axis and one for the Y axis), and one *covariance term* showing the tendency for X and Y to go together. The three variance and covariance terms are often grouped together into a symmetric covariance matrix as follows:

$$\begin{bmatrix} \sigma^2_{XX} & \sigma^2_{XY} \\ \sigma^2_{XY} & \sigma^2_{YY} \end{bmatrix}$$

Note that the terms σ^2_{XX} and σ^2_{YY} are simply the variances in the X and Y axes (the subscripts appear doubled, XX, for notational consistency). The term σ^2_{XY} is the covariance between the two axes.

cbind(), function(), outer(), dmvnorm()

```
library(mvtnorm)
sigma.xx <- 4
sigma.yy <- 1
sigma.xy <- 0 # no covariance
sigma <- matrix(c(sigma.xx,sigma.xy,sigma.xy,sigma.yy),ncol=2)
old.par <- par(mfrow=c(1,2))
x <- seq(-5,5,by=0.25)
y <- x
f <- function(x,y)  {
  #cat("X: ", x,"\n")
  #cat("Y:", y," \n")
  xy <- cbind(x,y)
  #cat("XY: ", xy,"\n")
  dmvnorm(xy,c(0,0),sigma)
}
z <- outer(x,y,f)
persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue",
  ltheta = 120, shade = 0.75)
contour(x, y, z, method = "edge",xlab="X",ylab="Y")
par(old.par)
# do the same thing again with sigma.xy <- 1.
# Max sigma.xy for this case is 2
```

5 A bit more about variance, the binomial distribution, and the normal distribution

With covariance in hand, we can now express the variance of a sum of random variables. If $Z = X + Y$ then

$$Var(Z) = Var(X) + Var(Y) + 2Cov(X, Y)$$

As a result, if two random variables are independent then the variance of their sum is the sum of their variances.

As an application of this fact, we can now calculate the variance of a binomially distributed random variable X with parameters n, p. X can be expressed as the sum of n identically distributed Bernoulli random variables X_i, each with parameter p. We've already established that the variance of each X_i is $p(1 - p)$. So the variance of X must be $np(1 - p)$.

5.1 Normal approximation to the binomial distribution

Finally, we can combine these facts with the central limit theorem to show how the binomial distribution can be approximated with the normal distribution. The central limit theorem says that the sum of many independent random variables tends toward a normal distribution with appropriate mean and variance. In our case, the binomial distribution has mean np and variance $np(1 - p)$. Thus, when n is large, the binomial distribution should look approximately normal. We can test this:

```
p <- 0.3
```

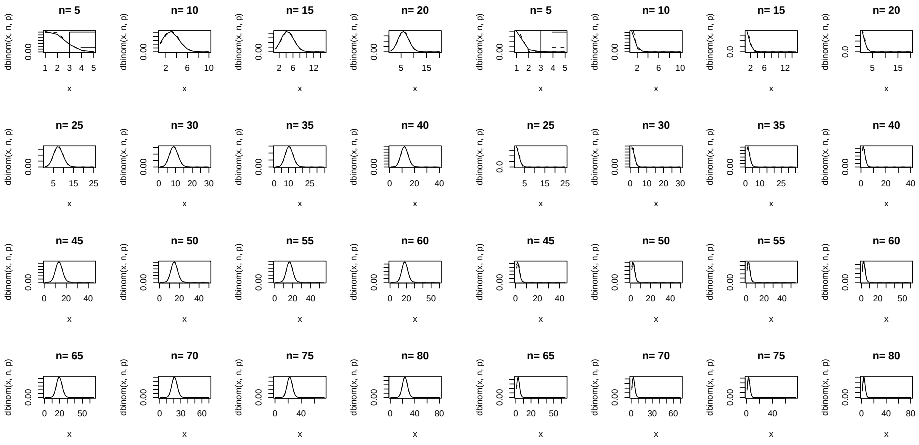


Figure 4: Approximation of binomial distribution with normal distribution for $p = 0.4$ (left) and $p = 0.05$ (right).

```
old.par <- par(mfrow=c(4,4))
for(n1 in 1:16) {
  n <- n1*5
  x <- 1:n
  plot(x, dbinom(x,n,p),type="l",lty=1,main=paste("n=",n))
  lines(x, dnorm(x,n*p,sqrt(n*p*(1-p))),lty=2)
  if(n1==1) {
    legend(3,0.35,c("Binomial","Normal"),lty=c(1,2),cex=1.5)
  }
}
par(old.par)
# then try with p=0.05
```

Lecture 5: Introduction to parameter estimation

15 October 2007

In this lecture we'll learn the following:

1. what parameter estimation is;
2. the basics of frequentist parameter estimation;
3. the method of maximum likelihood, and its strengths and weaknesses;
4. parameter estimation for the binomial and normal distributions;
5. the notions of *bias* in parameter estimation.

1 Parameter Estimation

Suppose I am a child learning my native language, English, and have just gotten to the point where I understand enough of the phonetics, phonology, morphology, syntax, and semantics to start making generalizations about when to use which variant of the ditransitive construction:

> Susan gave the ball to Fred [prepositional object; PO]
> Susan gave Fred the ball [double object; DO]

So far, I can reliably reconstruct seven instances of the DO and three instances of the PO in my linguistic experience. How can I make rational decisions about the following issues?

- How likely is the next ditransitive construction I hear to be a DO?
- If I need to construct a ditransitive clause of my own, should I make it a PO or a DO?

Although this is a relatively simple example, it already conjures several of the most fundamental problems of language acquisition:

1. In what formal framework can we adequately characterize linguistic input, and the resulting generalizations about the native language that are learned from the input?
2. What are the features of the linguistic input to which the learner attends?
3. What is the nature of the inductive bias?

Answering both sets of questions within the language of statistics requires two steps:

1. Choosing a family, or set of families, of probabilistic models to represent the probability of a PO versus DO ditransitive (given some intended meaning);
2. Coming to some conclusions about the model parameters on the basis of the data that have been observed.

The first of these two questions is the problem of MODEL SELECTION and we will postpone addressing it until a later date in the class, after we have introduced a wider variety of statistical model families applicable to this and other types of data. In principle, we would want these probabilities to be sensitive to a wide variety of factors potentially including (but not limited to) the type of meaning to be conveyed, the linguistic and extra-linguistic environment, and attributes of the individual speaker. We do not yet have the tools to include such a variety of factors in a sophisticated way, however, so for the moment we might simply choose to use the binomial distribution to model $P(\texttt{DO})$ (and correspondingly $P(\texttt{PO})$).

This choice does not, however, answer the second of these problems: given a choice of model family (families), what parameter setting(s) make sense given the data that are observed? This is the problem of PARAMETER ESTIMATION and we will occupy ourselves with it for the rest of this lecture.

2 Maximum Likelihood

Given a particular statistical model family with parameters θ and a dataset $\vec{x}$, the quantity $P(\vec{x}|\theta)$ can be viewed in two ways. One way is to viewed it as a function of $\vec{x}$, with θ fixed. Under this view it is the JOINT PROBABILITY of the dataset $\vec{x}$. The other view is as a function of θ, with $\vec{x}$ fixed. Under this view it is called the LIKELIHOOD of the parameter settings θ.

The LIKELIHOOD is a function of a set of parameters for a family of statistical models, given the dataset:

$$\mathrm{Lik}(\theta; \vec{x}) = P(\vec{x}|\theta)$$

We now reach the leading (frequentist) principle of parameter estimation, the METHOD OF MAXIMUM LIKELIHOOD. This method states that you should choose the parameter estimate $\hat{\theta}$ such that the likelihood with respect to your dataset $\vec{x}$ is maximized. This choice of parameters is called the MAXIMUM LIKELIHOOD ESTIMATE or MLE, and is often denoted $\hat{\theta}_{MLE}$. Mathematically speaking this is written:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \mathrm{Lik}(\theta; \vec{x})$$

Maximum-likelihood estimation is a highly general method and the MLE has many desirable properties, most notably that as you accumulate more and more data, the MLE will converge to to the true parameters of the underlying model (except under special circumstances not ordinarily encountered in our field).

Note that many cases, the likelihood can be infinitesimal and it is much easier to deal with the LOG-LIKELIHOOD:

$$l(\theta; \vec{x}) = \log \mathrm{Lik}(\theta; \vec{x})$$

When likelihood is maximized, log-likelihood is maximized, and vice versa.

2.1 An example

We'll take our initial example of PO vs. DO constructions. Call DO the "success" and PO the "failure". We've chosen to model this dataset as a binomial distribution with $n = 10$. Then the likelihood of our dataset with respect to p can be calculated as

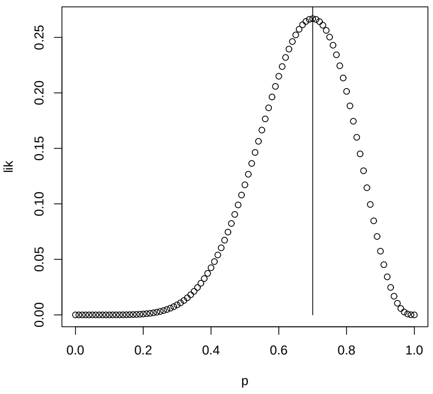


Figure 1: Likelihood of binomial parameter p for PO/DO dataset

$$\mathrm{Lik}(p; \vec{x}) = \binom{10}{7} p^7 (1-p)^3$$

This can be plotted as follows: (see Figure 1)

```
p <- seq(0,1,by=0.01)
lik <- dbinom(7,10,p)
plot(p,lik)
lines(c(0.7,0.7),c(0,0.3)) # draw a vertical line at p=0.7
```

Remarkably (or not), in this case the maximum-likelihood estimate of p is 0.7, or the relative frequency of the successful PO outcome. For estimating multinomial (including binomial) distributions, maximum-likelihood estimation turns out to be equivalent to setting the probabilities $\hat{p}_i$ equal to the relative frequencies of each outcome. This is also known as the *relative frequency estimate* or RFE.

3 Statistical bias

Let's stick with the method of maximum likelihood, and return to the example of estimating the weighting of a coin. We'll flip the coin n times. The true weighting of the coin is expressed by some hidden parameter value p, which we're in the business of trying to guess. If we have k successes in n coin flips, the maximum likelihood estimate is simply $\hat{p} = \frac{k}{n}$.

We can turn the tables and try to deconstruct the MLE by asking, for a given value of p, what is the expected value of $\hat{p}$? We write this down as

$$\begin{aligned}
E(\hat{p}) &= \sum_{k=0}^{n} \frac{k}{n} p(k) \\
&= \frac{1}{n} \sum_{k=0}^{n} k p(k) \\
&= \frac{1}{n} E(k) && \text{[Expectation of a binomial r.v.]} \\
&= \frac{1}{n} pn \\
&= p
\end{aligned}$$

We were able to take step 3 because $\sum_{k=0}^{n} kp(k)$ is simply the definition of the expected value of the binomial distribution, which we already determined to be pn. This is a very nice property: for a binomial distribution with known n, the MLE for $\hat{p}$ will on average give you back exactly p.

3.1 MLEs aren't perfect

Now we'll turn to a different problem. Suppose I am a sociolinguist studying bilingualism and language contact in the context of migration, and I become interested in knowing how long a particular speaker S who was peripherally involved in my study stayed in a community in a certain year. (S happened to be recorded in some of my studies, and his/her speech turned out to be particularly interesting.) However, the only data I have pertaining to S's presence is that every time S came to a certain café, the café manager (who worked as my informant) told me. Unfortunately, I can no longer get in touch with S. How can I estimate how long he/she stayed in this community?

We can use a simple statistical model to formulate this question. Let us assume that the time of each of S's visits to the café is uniformly distributed, and that each visit is independent. Let us further suppose that S actually arrived in the community at the beginning of day 1 of my study and left on the end of day 24, and visited the café four times, on days 4, 13, 15, and 20.

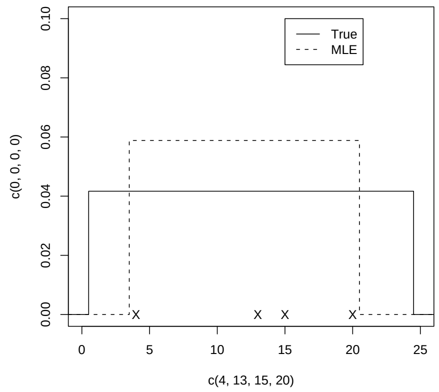


Figure 2: The MLE for a uniform distribution is biased.

Now consider the likelihood of these points under two distributions: (i) the true underlying distribution, and (ii) the uniform distribution with endpoints at $\langle 4, 20\rangle$. These are plotted as below, and shown in Figure 2:

```
plot(c(4,13,15,20),c(0,0,0,0),pch="X",xlim=c(0,25),ylim=c(0,0.1))
lines(c(-1,0.5,0.5,24.5,24.5,50),c(0,0,1/24,1/24,0,0),lty=1) # true
lines(c(-1,3.5,3.5,20.5,20.5,50),c(0,0,1/17,1/17,0,0),lty=2) # MLE
legend(15,0.1,c("True","MLE"),lty=c(1,2))
```

Note that each point has probability density 1/24 under the true distribution, but 1/17 under the second distribution. This latter distribution is in fact the MLE distribution—tightening the bounds any further will cause one of the points to have probability 0. Note how the MLE underestimates the true interval size. This is a general property of the MLE for uniform distributions—see Homework 3, problem 2.

This tendency of the MLE to underestimate the true interval size for a uniform distribution is an example of what is called "statistical bias". The statistical bias of an estimator is defined as the difference between the expected value of the parameter being estimated and its true value. We also say that an estimator is biased if its statistical bias is not equal to zero.

Finally, we'll briefly turn to the MLE for the mean and variance of a normal distribution given a sample of size n. It turns out that the MLEs are as follows:

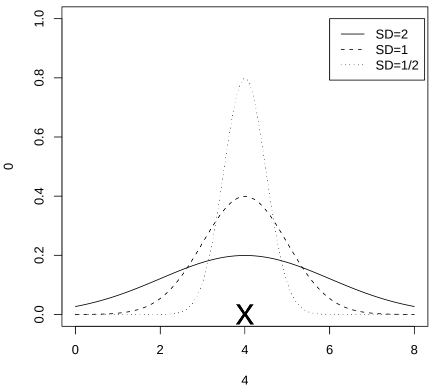


Figure 3: Bias in the MLE for σ in the normal distribution

$$\hat{\mu}_{MLE} = \frac{1}{n}\sum_i X_i \qquad \text{i.e., the sample mean}$$

$$\hat{\sigma}^2_{MLE} = \frac{1}{n}\sum_i (X_i - \hat{\mu})^2 \qquad \text{i.e., the sample variance divided by } n$$

While it turns out that $\hat{\mu}$ is unbiased, $\hat{\sigma}^2$ is not. You can see this graphically by imagining the MLE for a single observation, as in Figure 3. As $\hat{\sigma}^2$ shrinks, the likelihood of the observation will continue to rise, so that the MLE will push the estimated variance to be arbitrarily small.

```
plot(4,0,type="p",pch="x",xlim=c(0,8),ylim=c(0,1),cex=3)
x <- seq(0,8,by=0.01)
lines(x,dnorm(x,4,2))
lines(x,dnorm(x,4,1),lty=2)
lines(x,dnorm(x,4,1/2),lty=3)
legend(6,1,c("SD=2","SD=1","SD=1/2"),lty=c(1,2,3),cex=1)
```

It turns out that the this bias can be eliminated by adjusting the MLE by the factor $\frac{n}{n-1}$:

$$\begin{aligned}\hat{\sigma}^2 &= \frac{n}{n-1}\hat{\sigma}^2_{MLE} \\ &= \frac{1}{n-1}\sum_i (X_i - \hat{\mu})^2\end{aligned}$$

This is the most frequently used estimate of the underlying variance of a normal distribution from a sample, and it is what `R`'s `var()` function uses. `var()` A little simulation shows that it really isn't biased:

```
> v <- 9
> x <- c()
> n <- 5
> for(i in 1:10000) {
>   x <- c(x,var(rnorm(n,0,sqrt(v))))
> }
> y <- x * (n-1) / n
> hist(x)
> mean(x)
[1] 8.953537
> mean(y)
[1] 7.162829
```

Lecture 6: Bayesian parameter estimation; confidence intervals (Bayesian and frequentistic)

17 October 2007

In this lecture we'll learn the following:

1. what confidence intervals are;

2. how to calculate them;

3. how to interpret and use them.

1 Bayesian parameter estimation

Doing the full details of Bayesian parameter estimation can be rather involved, but I want to give you a quick example just to give you the flavor of it. Let's go back to the coin-flipping example. Bayesian statistics is characterized by placing a PRIOR DISTRIBUTION on the parameters θ. We'll denote the probability of success as π for this lecture. To be Bayesian, we need to place a prior distribution on π representing our beliefs in the absence of empirical coin-flip data. For example, we might place the prior probability

$$P(\pi) \propto \pi(1-\pi)$$

($\propto$ reads as "proportional to", which means "equal up to a multiplicative constant") which looks as follows (Figure 1):

```
p <- seq(0,1,by=0.01)
plot(p,p*(1-p),type="l")
```

Note that the mean, median, and mode of this curve are all at $\pi = 0.5$. This can be interpreted to mean that my average belief, my "halfway belief", and my "strongest belief" are all that the coin is fair.

When we are confronted with data, we use Bayes' rule to update our beliefs about π:

$$P(\pi|\vec{x}) = \frac{P(\vec{x}|\pi)P(\pi)}{P(\vec{x})}$$

Note that Bayes' rule is nothing new, it just follows from the definition of conditional probability! When we use Bayes' rule for statistical inference, we often just simplify the above expression to say

$$P(\pi|\vec{x}) \propto P(\vec{x}|\pi)P(\pi)$$

Now let's deconstruct this rule a bit more. We write it again, in annotated form:

$$P(\pi|\vec{x}) = \frac{\overbrace{P(\vec{x}|\pi)}^{\text{Likelihood in the model }\pi} \quad \overbrace{P(\pi)}^{\text{Prior belief}}}{\underbrace{P(\vec{x})}_{\text{Average data likelihood across possible models}}}$$

Once again, pay more atention to the numerator. The first term is the likelihood, which we learned about in the previous lecture. The second is our prior. When we combine these we get our posterior beliefs about $P(\pi|\vec{x})$, which are determined simply by multiplying the likelihood and the prior.

Suppose we saw seven successes and three failures. The likelihood is then

$$\begin{aligned} P(\vec{x}|\pi) &= \binom{10}{7}\pi^7(1-\pi)^3 \\ &\propto \pi^7(1-\pi)^3 \end{aligned}$$

Multiplying this by our prior beliefs we get

$$
\begin{aligned}
\overbrace{P(\pi|\vec{x})}^{\text{Posterior}} &\propto \overbrace{\pi^7(1-\pi)^3}^{\text{Likelihood}}\overbrace{\pi(1-\pi)}^{\text{Prior}} \\
&= \pi^8(1-p)^4
\end{aligned}
$$

and this is graphed as follows:

```
plot(p,p^8*(1-p)^4,type="l")
```

Note that this is *not* maximized at the MLE of $\hat{\pi} = 0.7$: `rank()`

```
rank(p^8*(1-p)^4)
  [1]   1.5   3.0   4.0   5.0   6.0   7.0   8.0   9.0  10.0  11.0  12.0  14.0  15.
 [17]  20.0  21.0  22.0  24.0  25.0  26.0  28.0  29.0  30.0  32.0  33.0  35.0  36.
 [33]  42.0  43.0  45.0  46.0  48.0  49.0  51.0  52.0  54.0  56.0  57.0  59.0  61.
 [49]  67.0  69.0  71.0  72.0  74.0  76.0  78.0  79.0  81.0  83.0  85.0  87.0  89.
 [65]  96.0  98.0 100.0 101.0  99.0  97.0  95.0  93.0  91.0  88.0  86.0  84.0  82.
 [81]  73.0  70.0  68.0  66.0  63.0  60.0  58.0  55.0  53.0  50.0  47.0  44.0  41.
 [97]  27.0  23.0  19.0  13.0   1.5
p[67]
[1] 0.66
abline(c(0.66,0.66),c(-1,1))
```

2 Confidence Intervals

In the previous lecture, we discussed how maximum-likelihood estimation does not give us information about the *certainty* associated with our parameter estimate. For example, if we want to estimate the parameter π of a binomial distribution from a sequence of coin tosses, the MLE is the same regardless of whether we observed 7 heads out of 10 or 700 heads out of 1,000.

2.1 Bayesian confidence intervals

Bayesian confidence intervals are very simple. A Bayesian $(1-\alpha)\%$ confidence interval is simply a continuous interval on θ such that the posterior probability mass contained in that interval is $1-\alpha$.

```
tot <- sum(p^8*(1-p)^4)
tot <-
[1] 0.01554002
P.subinterval <- seq(0.4,0.95,by=0.01)
subtot <- sum(p.subinterval^8*(1-p.subinterval)^4)
subtot/tot
[1] 0.9704187
```

So the region [0.4,0.95] is a 97% Bayesian confidence interval for π, given the prior and data as above.

Note that function `binom.bayes` in the `R` package `binom` can be used to auto-calculate a highest-probability density (HPD) interval:

```
> library(binom)
> binom.bayes(7,10,prior.shape1=1,prior.shape2=1)
  method x  n shape1 shape2      mean     lower     upper        sig
1  bayes 7 10      8      4 0.6666667 0.4120475 0.9066277 0.05000001
```

If the prior is strong and centered around the empirical results, then it tightens the confidence intervals. Note that the prior behaves exactly like "pseudo-data":

```
> binom.bayes(7,10,prior.shape1=70,prior.shape2=30)
  method x  n shape1 shape2 mean     lower     upper  sig
1  bayes 7 10     77     33  0.7 0.6140864 0.7838838 0.05
> binom.bayes(76,108,prior.shape1=1,prior.shape2=1)
  method  x   n shape1 shape2 mean     lower     upper  sig
1  bayes 76 108     77     33  0.7 0.6140864 0.7838838 0.05
```

2.2 Frequentist confidence intervals

To a frequentist, it does not make sense to say that "the true parameter θ lies between these points x and y with probability p^*." The parameter θ is a real property of the population from which the sample was obtained and is either in between x and y, or it is not. Remember, to a frequentist, the notion of probability as reasonable belief is not admitted! Therefore the Bayesian definition of a confidence interval—while intuitively appealing to many—is incoherent.

Instead, the frequentist uses more indirect means of quantifying their certainty about the estimate of θ. The issue is phrased thus: imagine that I were to repeat the same experiment—drawing a sample from my population—many times, and each time I repeated the experiment I constructed an interval I on the basis of my sample according to a fixed procedure `Proc`. Suppose it were the case that p percent of the intervals I thus constructed actually contained θ. Then for any given sample S, the interval I constructed by `Proc` is a $(1-p)\%$ confidence interval for θ.

If you think that is convoluted logic, well, you are right. **Frequentist confidence intervals are one of the most widely misunderstood constructs in statistics.** The Bayesian view is more intuitive to most people. Under some circumstances, there is a happy coincidence where Bayesian and frequentist confidence intervals look the same and you are free to misinterpret the latter as the former. In general, however, they do *not* necessarily look the same, and you need to be careful to interpret each correctly.

Here's an example, where we will explain the standard error of the mean. Suppose that we obtain a sample of n data points from a normal distribution. Some math shows that a confidence interval for the mean μ can be constructed as follows:[1]

$$\hat{\mu} = \frac{1}{n}\sum_i X_i \qquad \text{[maximum-likelihood estimate of the mean]}$$

$$S^2 = \frac{1}{n-1}\sum_i (X_i - \hat{\mu})^2 \quad \text{[empirical estimate of the standard deviation]}$$

$$P\left(|\hat{\mu} - \mu| < S/\sqrt{n}\; t_{n-1}(\alpha/2)\right) = 1 - \alpha \quad \text{[(1-}\alpha\text{)\% confidence interval for } \mu\text{]} \tag{1}$$

We can check this theoretical confidence interval with some simulation: `ifelse()`

```
mu <- 0
sigma <- 1
tot <- 0
```

[1] It isn't crucial that you know this math; for those who are interested, I put it up online as supplementary reading.

```
success <- c(0,0,0,0)
n <- 10
alpha <- c(0.001,0.01,0.05,0.1)
for(i in 1:10000) {
  x <- rnorm(n,mu,sigma)
  len <- sd(x)/sqrt(n) * (-1 * qt(alpha/2,n-1)) # calculates half the
                                                # length of the conf. interval
  tot <- tot + 1
  success <- success + ifelse(mean(x) < mu+len & mean(x) > mu-len, 1, 0)
}
success/tot
[1] 0.9990 0.9901 0.9524 0.9005
```

As you can see, this worked well!

> For a given choice of α, is the procedure denoted in (1) the only way to construct a confidence interval for μ?

Note that the quantity $S/\sqrt{n}$ is called the the STANDARD ERROR OF THE MEAN or simply the STANDARD ERROR. Note that this is different from the standard deviation of the sample, but related! (How?) When the number of observations n is large, the t distribution looks approximately normal, and as a rule of thumb, the symmetric 95% tail region of the normal distribution is about 2 standard errors away from the mean.

Another example: let's look at the the pb dataset, a classic study of the English vowel space (Peterson and Barney, 1952). (Thank you, Grant L.!) The distribution of the F1 formant for the vowel ϵ is roughly normally distributed:

```
pb <- read.table("pb.txt",header=T)
eh <- subset(pb,Vowel=="eh")
length(eh[,1]) # 152 data points
[1] 152
hist(eh$F1,breaks=20)
mean(eh$F1)
[1] 590.704
```

The 95% confidence interval can be calculated by looking at the quantity $S/\sqrt{n}\; t_{151}(0.025)$:

```
half.interval.length <- sd(eh$F1) / sqrt(length(eh$F1))
                          * (-1 * qt(0.025,length(eh$F1)))
half.interval.length
[1] 15.56314
c(mean(eh$F1) - half.interval.length, mean(eh$F1) + half.interval.length)
[1] 575.1408 606.2671
```

So our 95% confidence interval for the mean F1 is $[575, 606]$.

Finally, one more example, using the normal approximation to the binomial distribution. We'll compare 7 out of 10 successes to 700 out of 1000.

```
n <- 10
p <- 0.7
x <- c(rep(1,n*p),rep(0,n*(1-p)))
len <- sd(x)/sqrt(n) * (-1 * qt(0.25,n-1))
c(mean(x) - len, mean(x) + len)
[1] 0.5926574 0.8073426
# try again with n <- 1000
...
c(mean(x) - len, mean(x) + len)
[1] 0.6902173 0.7097827
```

Clearly we are much more certain about π with 1000 observations than we are with 10.

References

Peterson, G. E. and Barney, H. L. (1952). Control methods used in a study of the vowels. *Journal of the Acoustical Society of America*, 24(2):175–184.

Lecture 7: The χ^2 and t distributions, frequentist confidence intervals, the Neyman-Pearson paradigm, and introductory frequentist hypothesis testing

29 October 2007

In this lecture we'll learn the following:

1. about the χ^2 and t distributions;
2. how to compute and interpret frequentist confidence intervals;
3. what the Neyman-Pearson paradigm is;
4. and how to do simple frequentist hypothesis tests.

1 The χ^2 and t distributions

Before we learn properly about frequentist confidence intervals, it is necessary to learn about two other very important distributions, the χ^2 and t distributions.

First, we need to introduce the idea of a STANDARD NORMAL RANDOM VARIABLE. This is simply a normal random variable that has been rescaled such that its mean is zero and its standard deviation is 1. We sometimes write a normal distribution with mean μ and variance σ^2 as $\mathcal{N}(\mu, \sigma^2)$, so to say that a random variable X follows the standard normal distribution is sometimes written as

$$X \sim \mathcal{N}(0, 1)$$

(Read $\sim$ as "is distributed", so the whole thing above reads "X is normally distributed with mean 0 and variance 1".)

1.1 The χ^2 distribution

Now that we have the idea of a standard normal random variable, we can introduce the χ^2 distribution. If $X_1, \cdots, X_n$ are independent standard normal random variables, then the quantity $X_1^2 + \cdots + X_n^2$ is called the chi-squared distribution with n DEGREES OF FREEDOM and is written χ^2_n. The value n is the single parameter of the χ^2 family of distributions.

```
x <- seq(0,10,by=0.01)
plot(x,dchisq(x,1),type="l",ylim=c(0,0.8))
lines(x,dchisq(x,2),lty=2)
lines(x,dchisq(x,3),lty=3)
lines(x,dchisq(x,5),lty=4)
lines(x,dchisq(x,10),lty=5)
legend(3,0.75,c("1 d.f.","2 d.f.","3 d.f.","5 d.f.","10 d.f."),lty=1:5)
# *** add color!
```

The key place where χ^2 variables arise is as the distribution of variance of a normal distribution. If we sample n points from $\mathcal{N}(\mu, \sigma^2)$ (once again: that's a normal distribution with mean μ and variance σ^2), then the quantity

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

is distributed as χ^2_{n-1}. We can verify that with the code below (see Figure 1):

```
n <- 10
mu <- 0
sigma <- 2
y <- NULL
for(j in 1:10000) {
  y <- c(y, (1 / sigma^2) * var(rnorm(n,mu,sigma)) * (n-1) )
                      # *(n-1) because var() computes S^2 statistic
}
hist(y,breaks=20,prob=T)
```

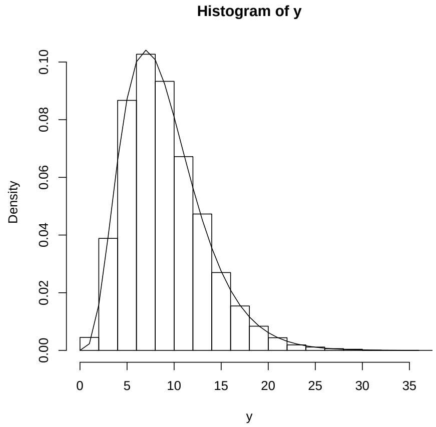


Figure 1: The sample variance for a normally-distributed random variable follows the χ^2 distribution.

```
x <- 0:200
lines(x,dchisq(x,n-1))
```

1.2 The t distribution

The t distribution is defined as the ratio of a standard-normal distributed random variable to a chi-squared random variable. In particular, let $Z \sim \mathcal{N}(0,1)$ and $U \sim \chi^2_n$, with U independent of Z. The t distribution with n degrees of freedom is defined by the distribution of $\frac{Z}{\sqrt{U/n}}$, and is denoted by t_n.

Figure 2 illustrates the t density for varying degrees of freedom, and also the standard normal distribution. Note that as the degrees of freedom n increases, the t distribution converges to the standard normal distribution! Keep this in mind as we move shortly to confidence intervals and hypothesis testing.

```
x <- seq(-5,5,by=0.01)
old.par <- par(lwd=2)
plot(x,dt(x,1),type="l",ylim=c(0,0.4),lty=2,col=2,ylab="t density")
lines(x,dt(x,2),lty=3,col=3)
lines(x,dt(x,10),lty=4,col=4)
lines(x,dt(x,20),lty=5,col=5)
```

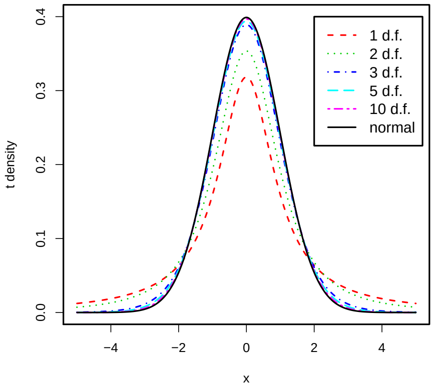


Figure 2: The probability density of the t distribution

```
lines(x,dt(x,50),lty=6,col=6)
lines(x,dnorm(x),lty=1,col=1)
legend(2,0.4,c("1 d.f.","2 d.f.","3 d.f.","5 d.f.","10 d.f.","normal"),
  lty=c(2:6,1),col=c(2:6,1),cex=1.2)
par(old.par)
```

2 Frequentist confidence intervals

To a frequentist, it does not make sense to say that "the true parameter θ lies between these points x and y with probability p^*." The parameter θ is a real property of the population from which the sample was obtained and is either in between x and y, or it is not. Remember, to a frequentist, the notion of probability as reasonable belief is not admitted! Therefore the Bayesian definition of a confidence interval—while intuitively appealing to many—is incoherent.

Instead, the frequentist uses more indirect means of quantifying their certainty about the estimate of θ. The issue is phrased thus: imagine that I were to repeat the same experiment—drawing a sample from my population—many times, and each time I repeated the experiment I constructed an interval I on the basis of my sample according to a fixed procedure `Proc`. Suppose it were the case that p percent of the intervals I thus constructed actually contained θ. Then for any given sample S, the interval I constructed by `Proc` is a $(1-p)\%$ confidence interval for θ.

If you think that is convoluted logic, well, you are right. **Frequentist confidence intervals are one of the most widely misunderstood constructs in statistics.** The Bayesian view is more intuitive to most people. Under some circumstances, there is a happy coincidence where Bayesian and frequentist confidence intervals look the same and you are free to misinterpret the latter as the former. In general, however, they do *not* necessarily look the same, and you need to be careful to interpret each correctly.

Here's an example, where we will explain the STANDARD ERROR OF THE MEAN. Suppose that we obtain a sample of n data points from a normal distribution. Some math shows that the following quantity follows the t_{n-1} distribution:[1]

$$\frac{\hat{\mu}-\mu}{\sqrt{S^2/n}} \sim t_{n-1} \tag{1}$$

where

$$\begin{aligned} \hat{\mu} &= \frac{1}{n}\sum_i X_i && \text{[maximum-likelihood estimate of the mean]} \\ S^2 &= \frac{1}{n-1}\sum_i (X_i-\hat{\mu})^2 && \text{[unbiased estimate of the standard deviation]} \end{aligned}$$

This means that a confidence interval for the mean μ can be constructed as follows:

$$P\left(|\hat{\mu}-\mu| < \frac{S}{\sqrt{n}} t_{n-1}(\alpha/2)\right) = 1-\alpha \quad \text{[(1-}\alpha\text{)\% confidence interval for }\mu\text{]} \tag{2}$$

We can visualize as follows:

```
n <- 5
```

[1] It isn't crucial that you know this math; for those who are interested, I put it up online as supplementary reading.

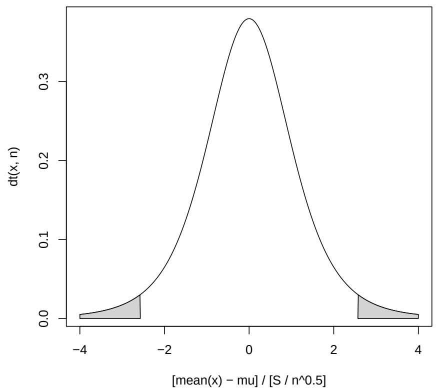


Figure 3: Visualizing confidence intervals with the t distribution

```
x <- seq(-4,4,by=0.01)
plot(x,dt(x,n),type="l",xlab="[mean(x) - mu] / [S / n^0.5]")
low.cutoff <- qt(0.025,n)
x.low <- subset(x,x < low.cutoff)
polygon(c(-4,x.low,low.cutoff),c(0,dt(x.low,n),0),col="lightgrey")
high.cutoff <- qt(0.975,n)
x.high <- subset(x,x >  high.cutoff)
polygon(c(high.cutoff,x.high,4),c(0,dt(x.high,n),0),col="lightgrey")
```

The results are shown in Figure 3 for $\alpha = 0.05$ (a 95% confidence interval). Most of the time, the "standardized" difference between $\hat{\mu}$ and μ is small and falls in the unshaded area. But 5% of the time, this standardized difference will fall in the shaded area—that is, the confidence interval won't contain μ. We can check this theoretical confidence interval with some simulation: `ifelse()`

```
mu <- 0
sigma <- 1
tot <- 0
success <- c(0,0,0,0)
n <- 10
alpha <- c(0.001,0.01,0.05,0.1)
for(i in 1:10000) {
  x <- rnorm(n,mu,sigma)
  len <- sd(x)/sqrt(n) * (-1 * qt(alpha/2,n-1)) # calculates half the
                                                # length of the conf. interval
  tot <- tot + 1
  success <- success + ifelse(mu < mean(x)+len & mu > mean(x)-len, 1, 0)
    # record a success in each case where mu falls inside the c.i.
}
success/tot
[1] 0.9990 0.9901 0.9524 0.9005
```

As you can see, this worked well!

> For a given choice of α, is the procedure denoted in (2) the only way to construct a confidence interval for μ?

Note that the quantity $S/\sqrt{n}$ is called the the standard error of the mean or simply the standard error. Note that this is different from the standard deviation of the sample, but related! (How?) When the number of observations n is large, the t distribution looks approximately normal, and as a rule of thumb, the symmetric 95% tail region of the normal distribution is about 2 standard errors away from the mean.

Another example: let's look at the the `pb` dataset, a classic study of the English vowel space (Peterson and Barney, 1952). (Thank you, Grant L.!) The distribution of the F1 formant for the vowel ɛ is roughly normally distributed (see Figure 4):

```
pb <- read.table("pb.txt",header=T)
eh <- subset(pb,Vowel=="eh")
length(eh[,1]) # 152 data points
[1] 152
mean(eh$F1)
[1] 590.704
sd(eh$F1)
[1] 97.11793
hist(eh$F1,breaks=20,prob=T)
x <- seq(350,900,by=1)
lines(x,dnorm(x,590.7,97.1))
```

The 95% confidence interval can be calculated by looking at the quantity $S/\sqrt{n}\; t_{151}(0.025)$:

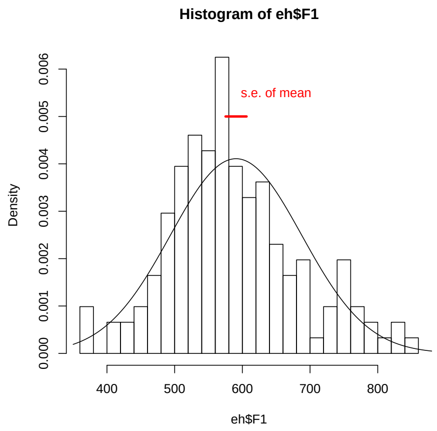


Figure 4: Distribution of F1 formant for ɛ

```
half.interval.length <- sd(eh$F1) / sqrt(length(eh$F1))
                          * (-1 * qt(0.025,length(eh$F1)))
half.interval.length
[1] 15.56314
c(mean(eh$F1) - half.interval.length, mean(eh$F1) + half.interval.length)
[1] 575.1408 606.2671
lines(c(575.1408,606.2671),c(0.005,0.005),lwd=3,col=2)
  # add in the standard error of the mean as a red line plus text
text(650,0.0055,"s.e. of mean",col=2)
```

So our 95% confidence interval for the mean F1 is $[575, 606]$.

Finally, one more example, using the normal approximation to the binomial distribution. We'll compare 7 out of 10 successes to 700 out of 1000.

```
n <- 10
p <- 0.7
x <- c(rep(1,n*p),rep(0,n*(1-p)))
len <- sd(x)/sqrt(n) * (-1 * qt(0.25,n-1)) # half-length of the c.i.
c(mean(x) - len, mean(x) + len) # calculate bounds of the c.i.
[1] 0.5926574 0.8073426
# try again with n <- 1000
...
c(mean(x) - len, mean(x) + len) # calculate bounds of the c.i.
```

```
[1] 0.6902173 0.7097827
```

Clearly we are much more certain about π with 1000 observations than we are with 10.

3 Introduction to frequentist hypothesis testing

In most of science, including areas such as psycholinguistics and phonetics, statistical inference is most often seen in the form of HYPOTHESIS TESTING within the NEYMAN-PEARSON PARADIGM. This paradigm involves formulating two hypotheses, the NULL HYPOTHESIS H_0 and the ALTERNATIVE HYPOTHESIS H_A (sometimes H_1). In general, there is an asymmetry such that H_A is more general than H_0. For example, let us take the coin-flipping example yet again. Let the null hypothesis be that the coin is fair:

$$H_0 : \pi = 0.5$$

]

The natural alternative hypothesis is simply that the coin may have any weighting:

$$H_A : 0 \leq \pi \leq 1$$

We then design a *decision procedure* on the basis of which we either ACCEPT or REJECT H_0 on the basis of some *experiment* we conduct. (Rejection of H_0 entails acceptance of H_A.) Now, within the Neyman-Pearson paradigm the true state of the world is that H_0 is either true or false. So the combination of the true state of the world with our decisions gives the following logically possible outcomes of an experiment:

(1)

		Null hypothesis	
		Accepted	Rejected
Null Hypothesis	True	Correct decision $(1-\alpha)$	Type I error (α)
	False	Type II error (β)	Correct decision $(1-\beta)$

As you can see in (1), there are two sets of circumstances under which we have done well:

1. The null hypothesis is true, and we accept it (upper left).
2. The null hypothesis is false, and we reject it (lower right).

This leaves us with two sets of circumstances under which we have made an error:

1. The null hypothesis is true, but we reject it. This by convention is called a TYPE I ERROR.
2. The null hypothesis is false, but we accept it. This by convention is called a TYPE II ERROR.

Let's be a bit more precise as to how hypothesis testing is done within the Neyman-Pearson paradigm. We know in advance that our experiment will result in the collection of some data $\vec{x}$. Before conducting the experiment, we decide on some TEST STATISTIC T that we will compute from $\vec{x}$.[2] We can think of T as a random variable, and the null hypothesis allows us to compute the distribution of T. Before conducting the experiment, we partition the range of T into an ACCEPTANCE REGION and a REJECTION REGION.[3]

(2) **Example:** a doctor wishes to evaluate whether a patient is diabetic. [Unbeknownst to all, the patient actually is diabetic.] To do this, she will draw a blood sample, $\vec{x}$, and compute the glucose level in the blood, T. She follows standard practice and designates the acceptance region as $T \leq 125\texttt{mg/dL}$, and the rejection region as $T > 125\texttt{mg/dL}$. The patient's sample reads as having 114`mg/dL`, so she diagnoses the patient as not having diabetes, committing a Type II error.

In this type of scenario, a Type I error is often called a FALSE POSITIVE, and a Type II error is often called a FALSE NEGATIVE.

The probability of Type I error is often denoted α and is referred to as the SIGNIFICANCE LEVEL of the hypothesis test. The probabilty of Type II error is often denoted β, and $1 - \beta$, which is the probability of correctly rejecting a false null hypothesis, is called the POWER of the hypothesis test. To calculate β and thus the power, however, we need to know the true model.

[2] Formally T is a function of $\vec{x}$ so we should designate it as $T(\vec{x})$, but for brevity we will just write T.

[3] For an unapologetic Bayesian's attitude about the Neyman-Pearson paradigm, read Section 37.1 of MacKay (2003).

Now we'll move on to another example of hypothesis testing in which we actually deploy some probability theory.

3.1 Hypothesis testing: a weighted coin

You decide to investigate whether a coin is fair or not by flipping it 16 times. As the test statistic T you simply choose the number of successes in 16 coin flips. Therefore the distribution of T under the null hypothesis H_0 is simply the distribution on the number of successes r for a binomial distribution with parameters $16, 0.5$, given below:

(3)

T	0	1	2	3	4	5	6	7	8
$p(T)$	0.0000153	0.000244	0.00183	0.00850	0.0278	0.0667	0.122	0.175	0.196
T	9	10	11	12	13	14	15	16	
$p(T)$	0.175	0.122	0.0667	0.0278	0.00854	0.00183	0.000244	0.0000153	

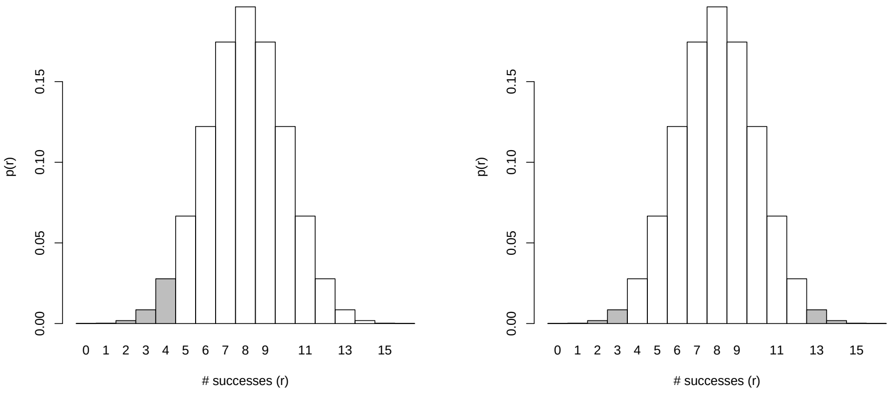
We need to start by partitioning the possible values of T into acceptance and rejection regions. The significance level α of the test will simply be the probability of landing in the rejection region under the distribution of T given in (3) above. Let us suppose that we want to achieve a significance level at least as good as $\alpha = 0.05$. This means that we need to choose as the rejection region a subset of the range of T with total probability mass no greater than 0.05. **Which values of T go into the rejection region is a matter of convention and common sense.**

Intuitively, it makes sense that if there are very few successes in 16 flips, then we should reject H_0. So we decide straight away that the values $T \leq 3$ will be in the rejection region. This comprises a probability mass of about 0.01:

```
> sum(dbinom(0:3,16,0.5))
[1] 0.01063538
```

We have probability mass of just under 0.04 to work with. Our next step now comes to depend on the alternative hypothesis we're interested in testing. If we are sure that the coin is not weighted towards heads but we think it may be weighted towards tails, then there is no point in putting high values of T into the rejection region, but we can still afford to add $T = 4$. We can't add $T = 5$, though, as this would put us above the $\alpha = 0.05$ threshold:

```
> sum(dbinom(0:4,16,0.5))
```



(a) One-tailed test

(b) Two-tailed test

Figure 5: Acceptance and rejection regions for one- and two-tailed tests for null hypothesis that a coin is fair. Acceptance regions are white; rejection regions are gray

```
[1] 0.03840637
> sum(dbinom(0:5,16,0.5))
[1] 0.1050568
```

Our rejection region is thus $T \leq 4$ and our acceptance region is $T \geq 5$. This is called a ONE-TAILED TEST and is associated with the alternative hypothesis $H_A : \pi < 0.5$. We can visualize the acceptance and rejection regions as follows:

```
x <- 0:16
colors <- c(rep(8,5),rep(0,12)) # color 8 is gray, 0 is white
barplot(dbinom(x,16,0.5),names.arg=x,space=0,col=colors,
  xlab="# successes (r)", ylab="p(r)")
  # barplot() allows more flexibility in coloring
```

Alternatively, we may have no reason to believe that the coin, if unfair, is weighted in a particular direction. In this case, symmetry demands that for every low value of T we include in the rejection region, we should include a corresponding high value of equal probability. So we would add the region $T \geq 13$ to our rejection region:

```
> sum(dbinom(0:3,16,0.5),dbinom(13:16,16,0.5))
[1] 0.02127075
```

We cannot add 4 and 12 to our rejection region because we would wind up with $\alpha > 0.05$:

```
> sum(dbinom(0:4,16,0.5),dbinom(12:16,16,0.5))
[1] 0.07681274
```

so we are finished and have the acceptance region $5 \leq T \leq 12$, with other values of T falling in the rejection region. This type of symmetric rejection region is called a TWO-TAILED TEST, which is associated with the alternative hypothesis $H_A : \pi \neq 0.5$.[4] We can visualize this as follows:

```
x <- 0:16
colors <- c(rep(8,4),rep(0,9), rep(8,4)) # color 8 is gray, 0 is white
barplot(dbinom(x,16,0.5),names.arg=x,space=0,col=colors,
  xlab="# successes (r)", ylab="p(r)")
  # barplot() allows more flexibility in coloring
```

In quantitative linguistics, you will nearly always see two-tailed tests rather than one-tailed tests, because the use of one-tailed test opens the door to post-hoc "explanations" of why, *a priori*, we don't expect to see deviations from the null hypothesis in the direction that we didn't see.

3.2 One-sample t-test

Finally, we cover the one-sample t-test. Suppose we believe we are sampling normally-distributed data and we want to test the null hypothesis that the mean of the data is a certain prespecified value μ_0. We can use the fact that the "standardized" mean of the distribution is t-distributed, as in Equation (1), to test $H_0 : \mu = \mu_0$. We can replicate an example given by Baayen for the distribution of duration for the Dutch prefix *ont-*. `t.test()`

[4] It is actually not just convention that associates these different alternative hypotheses with one- and two-tailed tests, but also the idea that the rejection region should be chosen so as to maximize the power of the test for a pre-specified choice of α. It turns out that these extreme-value choices of the rejection region accomplish this task, but explaining how is a more detailed discussion than we have time for.

```
t.test(durationsOnt$DurationPrefixNasal, mu = 0.053)
[...]
t = -1.5038, df = 101, p-value = 0.1358
```

The (two-tailed) p-value for the t-statistic is 0.1358, so that we cannot reject H_0 at the $\alpha = 0.05$ level (or even the ”marginal” $\alpha = 0.1$ level).

When you compare the means between two samples, the difference is also t-distributed but in a more complicated way. `t.test()` can also these two-sample tests.

3.3 Hypothesis testing: summary

- The Neyman-Pearson paradigm involves the formulation of two competing hypotheses: the null hypothesis H_0 and a more general alternative hypothesis H_A;

- H_0 and H_A are compared by choosing, in advance, a test statistic T to be calculated on the basis of data from an experiment, and partitioning the range of T into acceptance and rejection regions for H_0;

- Incorrectly rejecting H_0 when it is true is a Type I error (false positive); incorrectly accepting H_0 when it is false is a Type II error (false negative);

- The probability α of Type I error is the significance level of the hypothesis test;

- If we denote the probability of Type II error as β, then $1-\beta$ (the probability of correctly rejecting a false H_0) is the power of the hypothesis test.

References

MacKay, D. J. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge.

Peterson, G. E. and Barney, H. L. (1952). Control methods used in a study of the vowels. *Journal of the Acoustical Society of America*, 24(2):175–184.

Lecture 8: Frequentist hypothesis testing, and contingency tables

31 October 2007

In this lecture we'll learn the following:

1. what frequentist hypothesis testing is, and how to do it;
2. what contingency tables are and how to analyze them;
3. elementary frequentist hypothesis testing for count data, including the Chi-squared, likelihood-ratio, and Fisher's exact tests;

1 Introduction to frequentist hypothesis testing

In most of science, including areas such as psycholinguistics and phonetics, statistical inference is most often seen in the form of HYPOTHESIS TESTING within the NEYMAN-PEARSON PARADIGM. This paradigm involves formulating two hypotheses, the NULL HYPOTHESIS H_0 and the ALTERNATIVE HYPOTHESIS H_A (sometimes H_1). In general, there is an asymmetry such that H_A is more general than H_0. For example, let us take the coin-flipping example yet again. Let the null hypothesis be that the coin is fair:

$$H_0 : \pi = 0.5$$

The natural alternative hypothesis is simply that the coin may have any weighting:

$$H_A : 0 \leq \pi \leq 1$$

We then design a *decision procedure* on the basis of which we either accept or reject H_0 on the basis of some *experiment* we conduct. (Rejection of H_0 entails acceptance of H_A.) Now, within the Neyman-Pearson paradigm the true state of the world is that H_0 is either true or false. So the combination of the true state of the world with our decisions gives the following logically possible outcomes of an experiment:

(1)

		Null hypothesis	
		Accepted	Rejected
Null Hypothesis	True	Correct decision $(1-\alpha)$	Type I error (α)
	False	Type II error (β)	Correct decision $(1-\beta)$

As you can see in (1), there are two sets of circumstances under which we have done well:

1. The null hypothesis is true, and we accept it (upper left).

2. The null hypothesis is false, and we reject it (lower right).

This leaves us with two sets of circumstances under which we have made an error:

1. The null hypothesis is true, but we reject it. This by convention is called a Type I error.

2. The null hypothesis is false, but we accept it. This by convention is called a Type II error.

Let's be a bit more precise as to how hypothesis testing is done within the Neyman-Pearson paradigm. We know in advance that our experiment will result in the collection of some data $\vec{x}$. Before conducting the experiment, we decide on some test statistic T that we will compute from $\vec{x}$.[1] We can think of T as a random variable, and the null hypothesis allows us to compute the distribution of T. Before conducting the experiment, we partition the range of T into an acceptance region and a rejection region.[2]

[1] Formally T is a function of $\vec{x}$ so we should designate it as $T(\vec{x})$, but for brevity we will just write T.

[2] For an unapologetic Bayesian's attitude about the Neyman-Pearson paradigm, read Section 37.1 of **?**.

(2) **Example:** a doctor wishes to evaluate whether a patient is diabetic. [Unbeknownst to all, the patient actually is diabetic.] To do this, she will draw a blood sample, $\vec{x}$, and compute the glucose level in the blood, T. She follows standard practice and designates the acceptance region as $T \leq 125\texttt{mg/dL}$, and the rejection region as $T > 125\texttt{mg/dL}$. The patient's sample reads as having $114\texttt{mg/dL}$, so she diagnoses the patient as not having diabetes, committing a Type II error.

In this type of scenario, a Type I error is often called a FALSE POSITIVE, and a Type II error is often called a FALSE NEGATIVE.

The probability of Type I error is often denoted α and is referred to as the SIGNIFICANCE LEVEL of the hypothesis test. The probabilty of Type II error is often denoted β, and $1 - \beta$, which is the probability of correctly rejecting a false null hypothesis, is called the POWER of the hypothesis test. To calculate β and thus the power, however, we need to know the true model.

Now we'll move on to another example of hypothesis testing in which we actually deploy some probability theory.

1.1 Hypothesis testing: a weighted coin

You decide to investigate whether a coin is fair or not by flipping it 16 times. As the test statistic T you simply choose the number of successes in 16 coin flips. Therefore the distribution of T under the null hypothesis H_0 is simply the distribution on the number of successes r for a binomial distribution with parameters $16, 0.5$, given below:

(3)

T	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$p(T)$	0.0000153	0.000244	0.00183	0.00850	0.0278	0.0667	0.122	0.175	0.196	0.175	0.122	0.0667	0.0278	0.00854	0.00183	0.000244	0.0000153

We need to start by partitioning the possible values of T into acceptance and rejection regions. The significance level α of the test will simply be the probability of landing in the rejection region under the distribution of T given in (3) above. Let us suppose that we want to achieve a significance level at least as good as $\alpha = 0.05$. This means that we need to choose as the rejection region a subset of the range of T with total probability mass no greater than 0.05. **Which values of T go into the rejection region is a matter of convention and common sense.**

Intuitively, it makes sense that if there are very few successes in 16 flips, then we should reject H_0. So we decide straight away that the values $T \leq 3$ will be in the rejection region. This comprises a probability mass of about 0.01:

```
> sum(dbinom(0:3,16,0.5))
[1] 0.01063538
```

We have probability mass of just under 0.04 to work with. Our next step now comes to depend on the alternative hypothesis we're interested in testing. If we are sure that the coin is not weighted towards heads but we think it may be weighted towards tails, then there is no point in putting high values of T into the rejection region, but we can still afford to add $T = 4$. We can't add $T = 5$, though, as this would put us above the $\alpha = 0.05$ threshold:

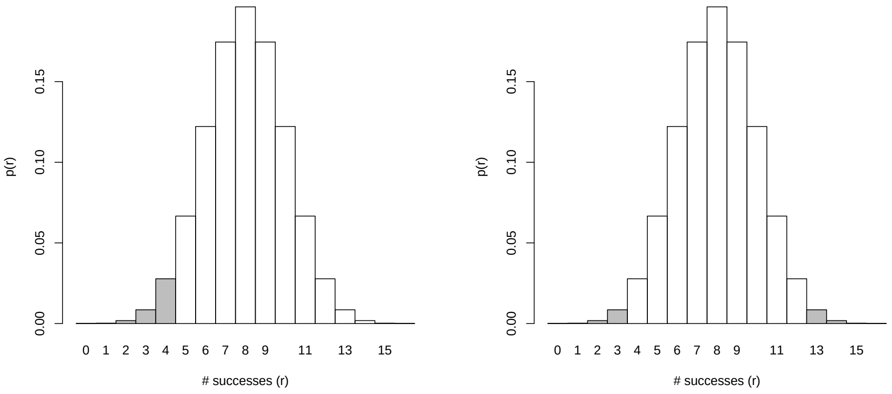
```
> sum(dbinom(0:4,16,0.5))
[1] 0.03840637
> sum(dbinom(0:5,16,0.5))
[1] 0.1050568
```

Our rejection region is thus $T \leq 4$ and our acceptance region is $T \geq 5$. This is called a ONE-TAILED TEST and is associated with the alternative hypothesis $H_A : \pi < 0.5$. We can visualize the acceptance and rejection regions as follows:

```
x <- 0:16
colors <- c(rep(8,5),rep(0,12)) # color 8 is gray, 0 is white
barplot(dbinom(x,16,0.5),names.arg=x,space=0,col=colors,
  xlab="# successes (r)", ylab="p(r)")
  # barplot() allows more flexibility in coloring
```

Alternatively, we may have no reason to believe that the coin, if unfair, is weighted in a particular direction. In this case, symmetry demands that for every low value of T we include in the rejection region, we should include a corresponding high value of equal probability. So we would add the region $T \geq 13$ to our rejection region:

```
> sum(dbinom(0:3,16,0.5),dbinom(13:16,16,0.5))
[1] 0.02127075
```



(a) One-tailed test

(b) Two-tailed test

Figure 1: Acceptance and rejection regions for one- and two-tailed tests for null hypothesis that a coin is fair. Acceptance regions are white; rejection regions are gray

We cannot add 4 and 12 to our rejection region because we would wind up with $\alpha > 0.05$:

```
> sum(dbinom(0:4,16,0.5),dbinom(12:16,16,0.5))
[1] 0.07681274
```

so we are finished and have the acceptance region $5 \leq T \leq 12$, with other values of T falling in the rejection region. This type of symmetric rejection region is called a TWO-TAILED TEST, which is associated with the alternative hypothesis $H_A : \pi \neq 0.5$.[3] We can visualize this as follows:

```
x <- 0:16
colors <- c(rep(8,4),rep(0,9), rep(8,4)) # color 8 is gray, 0 is white
barplot(dbinom(x,16,0.5),names.arg=x,space=0,col=colors,
  xlab="# successes (r)", ylab="p(r)")
  # barplot() allows more flexibility in coloring
```

[3] It is actually not just convention that associates these different alternative hypotheses with one- and two-tailed tests, but also the idea that the rejection region should be chosen so as to maximize the power of the test for a pre-specified choice of α. It turns out that these extreme-value choices of the rejection region accomplish this task, but explaining how is a more detailed discussion than we have time for.

In quantitative linguistics, you will nearly always see two-tailed tests rather than one-tailed tests, because the use of one-tailed test opens the door to post-hoc “explanations” of why, *a priori*, we don’t expect to see deviations from the null hypothesis in the direction that we didn’t see.

1.2 One-sample t-test

Finally, we cover the one-sample t-test. Suppose we believe we are sampling normally-distributed data and we want to test the null hypothesis that the mean of the data is a certain prespecified value μ_0. We can use the fact that the “standardized” mean of the distribution is t-distributed, as in Equation (**??**), to test $H_0 : \mu = \mu_0$. We can replicate an example given by Baayen for the distribution of duration for the Dutch prefix *ont-*.

`t.test()`

```
t.test(durationsOnt$DurationPrefixNasal, mu = 0.053)
[...]
t = -1.5038, df = 101, p-value = 0.1358
```

The (two-tailed) p-value for the t-statistic is 0.1358, so that we cannot reject H_0 at the $\alpha = 0.05$ level (or even the ”marginal” $\alpha = 0.1$ level).

When you compare the means between two samples, the difference is also t-distributed but in a more complicated way. `t.test()` can also these two-sample tests.

1.3 Hypothesis testing: summary

- The Neyman-Pearson paradigm involves the formulation of two competing hypotheses: the null hypothesis H_0 and a more general alternative hypothesis H_A;

- H_0 and H_A are compared by choosing, in advance, a test statistic T to be calculated on the basis of data from an experiment, and partitioning the range of T into acceptance and rejection regions for H_0;

- Incorrectly rejecting H_0 when it is true is a Type I error (false positive); incorrectly accepting H_0 when it is false is a Type II error (false negative);

- The probability α of Type I error is the significance level of the hypothesis test;

- If we denote the probability of Type II error as β, then $1-\beta$ (the probability of correctly rejecting a false H_0) is the power of the hypothesis test.

2 Contingency tables

There are many situations in quantitative linguistic analysis where you will be interested in the possibility of association between two categorical variables. In this case, you will often want to represent your data as a contingency table. Here's an example from my own research, on parallelism in noun-phrase coordination, [[NP1] and [NP2]]. Consider the following four noun phrases:

> The girl and the boy (parallel; no PPs)
> The girl from Quebec and the boy (not parallel)
> The girl and [the boy from Ottawa] (not parallel)
> The girl from Quebec and the boy from Ottawa (parallel; both with PPs)

I was interested in whether NP1 and NP2 tended to be similar to each other. As one instance of this, I looked at the patterns of PP modification in the Brown and Switchboard corpora, and came up with contingency tables like this:

			NP2					NP2		
	Brown		hasPP	noPP		Switchboard		hasPP	noPP	
(4)	NP1	hasPP	95	52	147	NP1	hasPP	78	76	154
		noPP	174	946	1120		noPP	325	1230	1555
			269	998	1267			403	1306	1709

From the table you can see that in both corpora, NP1 is more likely to have a PP postmodifier when NP2 has one, and NP2 is more likely to have a PP postmodifier when NP1 has one. But we would like to go beyond that and (a) *quantify* the predictive power of knowing NP1 on NP2; and (b) *test for significance* of the association.

3 Quantifying association: odds ratios

Given a contingency table of the form

		Y	
		y_1	y_2
X	x_1	n_{11}	n_{12}
	x_2	n_{21}	n_{22}

one of the things that's useful to talk about is how the value of one variable affects the distribution of the other. For example, the overall distribution of Y is

$$freq(y_1) = \tfrac{n_{11}+n_{21}}{n_{11}+n_{12}+n_{21}+n_{22}} \quad freq(y_2) = \tfrac{n_{12}+n_{22}}{n_{11}+n_{12}+n_{21}+n_{22}}$$

Alternatively we can speak of the overall *odds* ω^Y of y_1 versus y_2:

$$\omega^Y \equiv \frac{freq(y_1)}{freq(y_2)} = \frac{\frac{n_{11}+n_{21}}{n_{11}+n_{12}+n_{21}+n_{22}}}{\frac{n_{12}+n_{22}}{n_{11}+n_{12}+n_{21}+n_{22}}} = \frac{n_{11}+n_{21}}{n_{12}+n_{22}}$$

If $X = x_1$, then the odds for Y are just $\omega_1^Y = \frac{n_{11}}{n_{12}}$. If the odds of Y for $X = x_2$ are greater than the odds of Y for $X = x_1$, then the outcome of $X = x_2$ **increases** the chances of $Y = y_1$. We can express the effect of the outcome of X on the odds of Y by the **odds ratio** (which turns out to be symmetric between X, Y):

$$\mathcal{OR} = \frac{\omega_1}{\omega_2} = \frac{n_{11}/n_{12}}{n_{21}/n_{22}} = \frac{n_{11}n_{22}}{n_{12}n_{21}}$$

An odds ratio $\mathcal{OR} = 1$ indicates no association between the variables. For the Brown and Switchboard parallelism examples:

$$\mathcal{OR}_{Brown} = \tfrac{95\times946}{52\times174} = 9.93 \quad \mathcal{OR}_{Swbd} = \tfrac{78\times1230}{325\times76} = 3.88$$

So the presence of PPs in left and right conjunct NPs seems more strongly interconnected for the Brown (written) corpus than for the Switchboard (spoken).

4 Testing the significance of association

In frequentist statistics there are several ways to test the significance of the association between variables in a two-way contingency table. Although you may not be used to thinking about these tests as the comparison of two hypotheses in form of statistical models, they are!

4.1 Fisher's exact test

This test applies to a 2-by-2 contingency table:

$$\begin{array}{ccccc} & & & Y & \\ & & y_1 & y_2 & \\ X & x_1 & n_{11} & n_{12} & n_{1*} \\ & x_2 & n_{21} & n_{22} & n_{2*} \\ & & n_{*1} & n_{*2} & n \end{array}$$

H_0 is the model that all ***marginal totals*** are fixed, but that the individual cell totals are not – alternatively stated, that the individual outcomes of X and Y are independent. **This means that under H_0, the true odds ratio $\mathcal{OR}$is 1.** [H_0 has one free parameter – why?] H_A is the model that the individual outcomes of X and Y are not independent. With Fisher's exact test, you directly calculate the *exact* likelihood of obtaining a result as extreme or more extreme than the result that you got. [Since it is an *exact* test, you can use Fisher's exact test regardless of expected and actual cell counts.]

In R, you use the `fisher.test()` function to execute Fisher's exact test: fisher.test(),matrix(

```
> brown.nps <- matrix(c(95,174,52,946),2,2)
> fisher.test(brown.nps)

Fisher's Exact Test for Count Data

data:  brown.nps
p-value < 2.2e-16
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
  6.718106 14.737945
```

Notice how R told us that H_A is the hypothesis that the true odds ratio is not equal to 1.

4.2 Chi-squared test

This is probably the contingency-table test you have heard most about. It can be applied to arbitrary two-way $m \times n$ tables, if you have a model with

k parameters that predicts expected values E_{ij} for all cells. You calculate Pearson's X^2 statistic:

$$X^2 = \sum_{ij} \frac{[n_{ij} - E_{ij}]^2}{E_{ij}}$$

In the chi-squared test, H_A is the model that each cell in your table has its own parameter p_i in one big multinomial distribution. When the expected counts in each cell are large enough (the generally agreed lower bound is ≥ 5), the X^2 statistic is distributed as χ^2_{n-k-1}.

The most common way of using Pearson's chi-squared test is to test for the independence of two factors in a two-way contingency table. Take a $k \times l$ two-way table of the form:

$$\begin{array}{cccccc}
 & y_1 & y_2 & \cdots & y_l & \\
x_1 & n_{11} & n_{12} & \cdots & n_{1l} & n_{1*} \\
x_2 & n_{21} & n_{22} & \cdots & n_{2l} & n_{2*} \\
 & \vdots & \vdots & \ddots & \vdots & \vdots \\
x_l & n_{k1} & n_{k2} & \cdots & n_{kl} & n_{k*} \\
 & n_{*1} & n_{*2} & \cdots & n_{*l} & n
\end{array}$$

Our null hypothesis is that the x_i and y_i are independently distributed from one another. By the definition of probabilistic independence, that means that H_0 is:

$$P(x_i, y_j) = P(x_i)P(y_j)$$

In the chi-squared test we use the maximum-likelihood estimates $P(x_i) = P_{MLE}(x_i) = \frac{n_{i*}}{n}$ and $P(y_j) = P_{MLE}(y_j) = \frac{n_{*j}}{n}$. This gives us the formula for the expected counts:

$$E_{ij} = nP(x_i)P(y_j)$$

For the Brown corpus data in (4), we have

$$
\begin{align}
P(x_1) = \frac{147}{1267} && = 0.1160 P(y_1) && = \frac{269}{1267} = 0.2123 \tag{1}\\
P(x_2) && = 0.8840 P(y_2) && = 0.7877 \tag{2}\\
E_{11} = 31.2 \tag{3}\\
E_{12} = 115.8 \tag{4}\\
E_{21} = 237.8 \tag{5}\\
E_{21} = 882.2 \tag{6}\\
\tag{7}
\end{align}
$$

Comparing with (4), we get

$$
\begin{align}
X^2 = (95-31.2)^2/31.2 + (52-115.8)^2/115.8 + (174-237.8)^2/237.8 + (946-882.2)^2/882.2 \tag{8}\\
= 187.3445 \tag{9}
\end{align}
$$

We had 2 parameters in our model of independence, and there are 4 cells, so X^2 is distributed as χ^2_1 (4-2-1 = 1).

We can compare this with the built-in `chisq.test()` function:[4]

`chisq.test()`

```
> chisq.test(matrix(c(95,174,52,946),2,2),correct=F)

Pearson's Chi-squared test

data:  matrix(c(95, 174, 52, 946), 2, 2)
X-squared = 187.2482, df = 1, p-value < 2.2e-16
```

The exact numerical result is off because I rounded things off aggressively, but you can see where the result comes from.

4.3 Likelihood ratio test

With this test, the statistic you calculate for your data D is the ***likelihood ratio***

[4] There is something called a “continuity correction” in the chi-squared test which we don’t need to get into which is usable for 2 × 2 tables. The exposition we’re giving here ignores this correction.

$$\Lambda^* = \frac{\max P(D; H_0)}{\max P(D; H_A)}$$

that is: the ratio of the maximum data likelihood under H_0 to the maximum data likelihood under H_A. This requires that you explicitly formulate H_0 and H_A. $-2 \log \Lambda^*$ is distributed like a chi-squared with ***degrees of freedom*** equal to the difference in the the number of free parameters in H_A and H_0. [Danger: don't apply this test when expected cell counts are low, like < 5.]

The likelihood-ratio test gives similar results as the chi-squared for contingency tables, but is more flexible because it allows the comparison of arbitrary nested models.

Lecture 9: Bayesian hypothesis testing

5 November 2007

In this lecture we'll learn about Bayesian hypothesis testing.

1 Introduction to Bayesian hypothesis testing

Before we go into the details of Bayesian hypothesis testing, let us briefly review frequentist hypothesis testing. Recall that in the Neyman-Pearson paradigm characteristic of frequentist hypothesis testing, there is an *asymmetric* relationship between two hypotheses: the NULL hypothesis H_0 and the ALTERNATIVE hypothesis H_A. A decision procedure is devised by which, on the basis of a set of collected data, the null hypothesis will either be *rejected* in favor of H_A, or *accepted*.

In Bayesian hypothesis testing, there can be more than two hypotheses under consideration, and they do not necessarily stand in an asymmetric relationship. Rather, Bayesian hypothesis testing works just like any other type of Bayesian inference. Let us consider the case where we are considering only two hypotheses: H_1 and H_2. We know we will collect some data $\vec{x}$ but we don't yet know what that data will look like. We are interested in the posterior probabilities $P(H_1|\vec{x})$ and $P(H_2|\vec{x})$, which can be expressed using Bayes rule as follows:

$$P(H_1|\vec{x}) = \frac{P(\vec{x}|H_1)P(H_1)}{P(\vec{x})} \tag{1}$$

$$P(H_2|\vec{x}) = 1 - P(H_1|\vec{x}) \tag{2}$$

Crucially, the probability of our data $P(\vec{x})$ takes into account the possibility of each hypothesis under consideration to be true:

$$P(\vec{x}) = P(\vec{x}|H_1)P(H_1) + P(\vec{x}|H_2)P(H_2) \tag{3}$$

In other words, we are *marginalizing* over the possible hypotheses to calculate the data probability:

$$P(\vec{x}) = \sum_i P(\vec{x}|H_i)P(H_i) \tag{4}$$

As an example, we will return once more to the case of the possibly weighted coin as a case study. We will call hypothesis 1 the "fair coin" hypothesis, that the binomial parameter π is 0.5. In Bayesian statistics, model parameters have probabilities, so we state the fair coin hypothesis as:

$$H_1 : P(\pi|H_1) = \begin{cases} 1 & \pi = 0.5 \\ 0 & \pi \neq 0.5 \end{cases}$$

The probability above is a PRIOR PROBABILITY on the binomial parameter π.

Hypothesis 2 is the "weighted coin" hypothesis. For this hypothesis we must place a non-trivial probability distribution on π. Suppose that we did not entertain the possibility that the coin was two-headed or two-tailed, but we did consider it possible that the coin was weighted so that two out of three tosses turned up either heads or tails, and that each of these two possibilities was equally likely. This gives us:

$$H_2 : P(\pi|H_2) = \begin{cases} 0.5 & \frac{1}{3} \\ 0.5 & \frac{2}{3} \end{cases} \tag{5}$$

In order to complete the comparison in Equation (1), we need prior probabilities on the hypotheses themselves, $P(H_1)$ and $P(H_2)$. If we had strong beliefs one way or another about whether this coin was fair (e.g., from prior experience with the coin vendor), we might set one of these prior probabilities close to 1. For these purposes, we will use $P(H_1) = P(H_2) = 0.5$.

Now suppose we flip the coin six times and observe the sequence

`HHTHTH`

We can summarize this dataset as Does this data favor H_1 or H_2? We answer this question by completing Equation (1). We have:

$$
\begin{aligned}
P(H_1) &= 0.5 && (6)\\
P(\vec{x}|H_1) &= \binom{6}{4}(\frac{1}{2})^4(\frac{1}{2})^2 && (7)\\
&= \binom{6}{4}0.0156 P(H_2) \qquad = 0.5 && (8)
\end{aligned}
$$

Now to complete the calculation of $P(\vec{x})$ in Equation (3), we need $P(\vec{x}|H_2)$. To do this, we need to consider all possible values of π given H_2—that is, MARGINALIZE over π just as we are marginalizing over H to get the probability of the data. We have:

$$
\begin{aligned}
P(\vec{x}|H_2) &= \sum_i P(\vec{x}|\pi_i)P(\pi_i) && (9)\\
&= P(\vec{x}|\pi = \frac{1}{3})P(\pi = \frac{1}{3}) + && (10)\\
& P(\vec{x}|\pi = \frac{2}{3})P(\pi = \frac{2}{3}) + && (11)\\
&= \binom{6}{4}(\frac{1}{3})^4(\frac{2}{3})^2 \times 0.5 + \binom{6}{4}(\frac{2}{3})^4(\frac{1}{3})^2 \times 0.5 && (12)\\
&= \binom{6}{4} \times 0.0137 && (13)
\end{aligned}
$$

thus

$$
\begin{aligned}
P(\vec{x}) &= \overbrace{\binom{6}{4}0.0156}^{P(\vec{x}|H_1)} \times \overbrace{0.5}^{P(H_1)} + \overbrace{\binom{6}{4}0.0137}^{P(\vec{x}|H_2)} \times \overbrace{0.5}^{P(H_2)} && (14)\\
&= \binom{6}{4}0.01465 && (15)
\end{aligned}
$$

Therefore

$$
\begin{aligned}
P(H_1|\vec{x}) &= \frac{\binom{6}{4}0.0156 \times 0.5}{\binom{6}{4}0.01465} && (16)\\
&= 0.53 && (17)
\end{aligned}
$$

Note that even though the maximum-likelihood estimate of $\hat{\pi}$ from the data we observed hits one of the two possible values of π under H_2 on the head, our data actually supports the "fair coin" hypothesis H_1 – its support went up from a prior probability of $P(H_1) = 0.5$ to a posterior probability of $P(H_1|\vec{x}) = 0.53$.

1.1 More complex hypotheses

We might also want to consider more complex hypotheses than H_2 above as the "weighted coin" hypothesis. For example, we might think all possible values of π in $[0, 1]$ are equally probable *a priori*:

$$H_3 : P(\pi|H_2) = 1 \quad 0 \leq \pi \leq 1$$

(In Hypothesis 3, the probability distribution over π is continuous, not discrete, so H_3 is still a proper probability distribution.) Let us discard H_2 and now compare H_1 against H_3.

Let us compare H_3 against H_1 for the same data. To do so, we need to calculate the likelihood $P(\vec{x}|H_2)$, and to do this, we need to marginalize over π. Since π can take on a continuous range of values, this marginalization takes the form of an integral:[1]

[1] In general, the following relation holds:

$$
\begin{aligned}
\int_0^1 \pi^a (1-\pi)^b d\pi &= \frac{\Gamma(a+1)\Gamma(b+1)}{\Gamma(a+b+2)}\\
&= \frac{a!b!}{(a+b+1)!} && \text{when } a \text{ and } b \text{ are integers}
\end{aligned}
$$

The quantity $\frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b+1)}$ is also called the BETA FUNCTION with parameters $a+1$,$b+1$, accessible in `R` as `beta()`.

$$\begin{aligned} P(\vec{x}|H_2) &= \binom{6}{4} \int_0^1 \pi^4(1-\pi)^2 d\pi \\ &= \binom{6}{4} 0.0095 \end{aligned}$$

If we plug this result back in, we find that

$$\begin{aligned} P(H_1|\vec{x}) &= \frac{\binom{6}{4}0.0156 \times 0.5}{\binom{6}{4}0.01255} \\ &= 1.243 \end{aligned}$$

So H_3 fares even worse than H_2 against the fair-coin hypothesis H_1. Correspondingly, we would find that H_2 is favored over H_3.

> Would our hypothesis-testing results be changed at all if we did not consider the data as summarized by the number of successes and failures, and instead used the likelihood of the specific sequence `HHTHTH` instead?

1.2 Bayes factor

Sometimes we do not have strong feelings about the prior probabilities $P(H_i)$. Nevertheless, we can quantify how strongly a given dataset supports one hypothesis over another in terms of

$$\frac{P(\vec{x}|H_1)}{P(\vec{x}|H_2)}$$

that is, the LIKELIHOOD RATIO for the two hypotheses. This likelihood ratio is also called the BAYES FACTOR.

2 Learning contextual contingencies in sequences

Consider a sequence (e.g., phonemes) of length 20.

ABABBAAAAABBBABBBAAAA

Let us entertain two hypotheses. The first hypothesis H_1, is that the probability of an A is independent of the context. The second hypothesis, H_2, is that the probability of an A is dependent on the preceding token. How might this data influence the learner?

We can make these hypotheses precise in terms of the parameters that each entails. H_1 involves only one parameter $P(A)$, which we will call π. H_2 involves three parameters:

1. $P(A|\emptyset)$ (the probability that the sequence will start with A), which we will call $\pi_\emptyset$;

2. $P(A|A)$ (the probability that an A will appear after an A), which we will call π_A

3. $P(A|B)$ (the probability that an A will appear after an B), which we will call π_B.

Let us assume that H_1 and H_2 are equally likely; we will be concerned with the likelihood ratio between the two hypotheses. We will put a uniform prior distribution on all model parameters.

There are 21 observations, 12 of which are A and 9 of which are B . The likelihood of H_1 is therefore simply

$$\int_0^1 \pi^{12}(1-\pi^9)d\pi = 1.546441 \times 10^{-07} \tag{18}$$

To calculate the likelihood of H_2 it helps to break down the results into a table:

	Outcome	
	A	B
∅	1	0
A	7	4
B	4	5

So the likelihood of H_2 is

$$\int_0^1 \pi_\emptyset^1 d\pi_\emptyset \times \int_0^1 \pi_A^7(1-\pi_A^4)d\pi_A \times \int_0^1 \pi_B^4(1-\pi_B^5)d\pi_B \tag{19}$$

$$= 0.5 \times 0.00025 \times 0.00079 \tag{20}$$

$$= 1.002084 \times 10^{-07} \tag{21}$$

This dataset provides some support for the simpler hypothesis of statistical independence—the Bayes factor is about 1.5 in favor of H_1.

Introduction to Probability

1 October 2007

1 What are probabilities?

There are two basic schools of thought as to the philosophical status of probabilities. One school of thought, the *frequentist* school, considers the probability of an event to be its asymptotic frequency over an arbitrarily large number of repeated trials. For example, to say that the probability of a toss of a fair coin landing as Heads is 0.5 (ignoring the possibility that the coin lands on its edge) means to a frequentist that if you tossed the coin many, many times, the proportion of Heads outcomes would approach 50%.

The second, *Bayesian* school of thought considers the probability of an event E to be a principled measure of the strength of one's belief that E will result. For a Bayesian, to say that $P(\text{Heads})$ for a fair coin is 0.5 (and thus equal to $P(\text{Tails})$) is to say that you believe that Heads and Tails are equally likely outcomes if you flip the coin.

The debate between these interpretations of probability rages, and we're not going to try and resolve it in this class. Fortunately, for the cases in which it makes sense to talk about both reasonable belief and asymptotic frequency, it's been proven that the two schools of thought lead to the same rules of probability. If you're further interested in this, I encourage you to read Cox (1946), a beautiful, short paper.

2 Sample Spaces

The underlying foundation of any probability distribution is the SAMPLE SPACE—a set of possible OUTCOMES, conventionally denoted Ω.For example, if you toss two coins, the event space is

$$\Omega = \{hh, ht, th, hh\}$$

where h is Heads and t is Tails. Sample spaces can be finite, countably infinite (e.g., the set of integers), or uncountably infinite (e.g., the set of real numbers).

3 Events and probability spaces

An event is simply a subset of a sample space.

> What is the sample space corresponding to the roll of a single six-sided die? What is the event that the die roll comes up even?

It follows that the negation of an event E (that is, E not happening) is simply $\Omega - E$.

A probability space P on Ω is a function from events in Ω to real numbers such that the following three properties hold:

1. $P(\Omega) = 1$.
2. $P(E) \geq 0$ for all $E \subset \Omega$.
3. If E_1 and E_2 are disjoint, then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$.

4 Conditional Probability and Independence

We'll use an example to illustrate conditional independence. In Old English, the object in a transitive sentence could appear either preverbally or postverbally. Suppose that amoung transitive sentences in a corpus, the frequency distribution of object position and pronominality is as follows:

(1)

	Pronoun	**Not Pronoun**
Object **Preverbal**	0.224	0.655
Object **Postverbal**	0.014	0.107

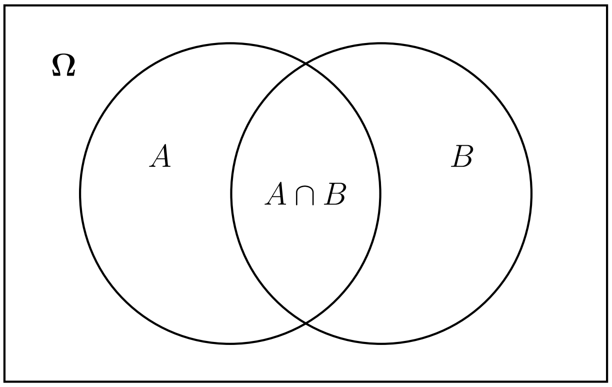


Figure 1: Conditional Probability (after Manning and Schütze (1999))

Let's interpret these frequencies as probabilities. What is the CONDITIONAL PROBABILITY of pronominality given that an object is postverbal?

The conditional probability of event B given that A has occurred/is known is defined as follows:

$$P(B|A) \triangleq \frac{P(A \cap B)}{P(A)}$$

In our case, event A is **Postverbal**, and B is **Pronoun**. The quantity $P(A \cap B)$ is already listed explicity in the lower-right cell of table (1): 0.014. We now need the quantity $P(A)$. For this we need to calculate the MARGINAL TOTAL of row 2 of Table (1): $0.014 + 0.107 = 0.121$. We can then calculate:

$$P(\textbf{Pronoun}|\textbf{Postverbal}) = \tfrac{0.014}{0.014+0.107} = 0.116$$

4.1 (Conditional) Independence

Events A and B are said to be CONDITIONALLY INDEPENDENT GIVEN C if

$$P(A \cap B|C) = P(A|C)P(B|C)$$

A more philosophical way of interpreting conditional independence is that if we are in the state of knowledge denoted by C, then conditional independence of A and B means that knowing A tells us nothing more about the probability of B, and vice versa. You'll also see the term we are in the state of "not knowing anything at all" ($C = \emptyset$) then we would simply say in this case that A and B are CONDITIONALLY INDEPENDENT.

It's crucial to keep in mind that if A and B are conditionally independent given C, that does not guarantee they will be conditionally independent given some other set of knowledge C'.

5 Random Variables

Technically, a RANDOM VARIABLE X is a function from Ω to the set of real numbers ($\mathbb{R}$). You can think of a random variable as an "experiment" whose outcome is not known in advance. In fact, the OUTCOME of a random variable is a technical term simply meaning which number resulted from the "experiment".

The relationship between the sample space Ω, a probability space P on Ω, and a random variable X on Ω can be a bit subtle so I'll explain it intuitively, and also with an example. In many cases you can think of a random variable as a "partitioning" of the sample space into the distinct classes of events that you (as a researcher, or as a person in everyday life) care about. For example, suppose you are trying to determine whether a particular coin is fair. A natural thing to do is to flip it many times and see how many times it comes out heads. Suppose you decide to flip it eight times. The sample space Ω is then all possible sequences of length eight whose members are either H or T. The coin being fair corresponds to the probability space P in which each point in the sample space has equal probability $\frac{1}{2^8}$. Now suppose you go ahead and flip the coin eight times, and the outcome is

TTTTTTHT

Intuitively, this is a surprising result. But under P, all points in Ω are equiprobable, so there is nothing about the result that is particularly surprising.

The key here is that you as an investigator of the coin's fairness are not interested in the particular H/T sequence that resulted. You are interested in *how many of the tosses came up heads.* This quantity is your random variable of interest—let's call it X. The logically possible outcomes of X are the integers $\{0, 1, \cdots, 8\}$. The actual outcome was $X = 1$—and there were seven other possible points in Ω for which $X = 1$ would be the outcome! We can use this to calculate the probability of this outcome of our random variable under the hypothesis that the coin is fair:

$$P(X = 1) = \frac{1}{2^8} \times 8 = \frac{1}{2^5} = 0.03125$$

So seven tails out of eight is a pretty surprising result. Incidentally, there's a very interesting recent paper (Griffiths and Tenenbaum, 2007) that deals with how humans actually *do* ascribe surprise to a "rare" event like a long sequence of heads in a coin flip.

6 Basic data visualization

We'll illustrate two basic types of data visualization—histograms and plots—by briefly investigating the relationship between word lengths and word frequencies.

6.1 Histograms

The file `brown-counts-lengths` contains the length and frequency of every word type appearing in the parsed Brown corpus. We'll start by visualizing the distributions of frequency and length counts.

```
# header=T reads off the
> x <- read.table("brown-counts-lengths",header=T)
> head(x,n=20)
   Count        Word Length
1    163           '      1
2     58           $      1
3     24           &      1
4      4           %      1
5      1           0      1
6      1        0600      4
7     27           1      1
8     16          10      2
9      8         100      3
10     1        1000      4
11     3       1,000      5
12     4      10,000      6
13     3     100,000      7
```

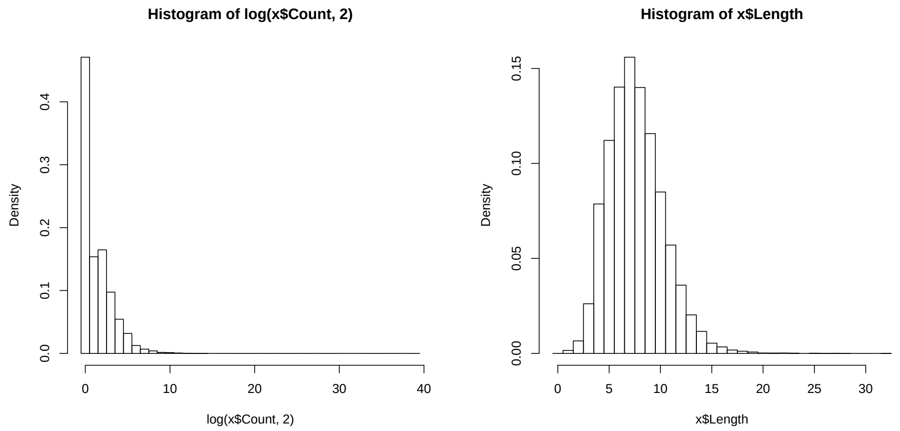


Figure 2: Histograms of log-frequency counts and word lengths for Brown corpus

```
14      1   1,000,000      9
15      1  10,000,000     10
16      1        1020      4
17      1 105-degrees     11
18      1         106      3
19      1         108      3
20      1      10-day      6
> hist(log(x$Count,2),breaks=seq(-0.5,39.5,by=1),prob=T)
> hist(x$Length,breaks=seq(-0.5,32.5,by=1),prob=T)
# results shown in Figure
```

Most of the word types (nearly 50%) are frequency 1—a word with a single instance in a given corpus is sometimes called a *hapax legomenon*. We can look up the ten most frequent words in the Brown corpus using the `order()` command:

```
> x[order(x$Count,decreasing=T)[1:10],]
      Count Word Length
26280 22244  the      3
17904 10964   of      2
1264  10661  and      3
26681  9778   to      2
```

```
337    8960    a      1
13265  6575   in      2
28599  5499  was      3
12187  4195   he      2
13001  4131    I      1
26270  4079 that      4
```

The longest word in the Brown corpus is:

```
> x[order(x$Length,decreasing=T)[1],]
      Count                           Word Length
17480     1 nnuolapertar-it-vuh-karti-birifw     32
```

Not surprisingly, this is *hapax legomenon*.

6.2 Plots

We can investigate the *relationship* between length and frequency by plotting them against each other:

```
> plot(x$Length,log(x$Count,2))
> lines(lowess(x$Length, log(x$Count,2)))
# results in Figure
```

The `lowess()` function is a kind of smoother that estimates the y-value of a function for a value of y, given a full dataset of (x, y) points. You can use it to graphically explore the relationship between two variables on a relatively informal basis.

Finally, you can use `identify()` to interactively inspect the points on a plot.

```
> ?plot
> plot(x$Length,log(x$Count,2))
> identify(x$Length,log(x$Count,2),labels=x$Word)
[1] 17480 24298 26270 26280
```

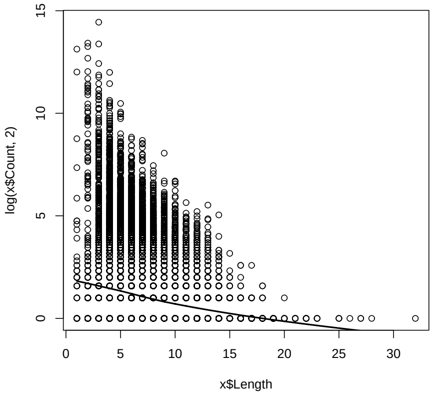


Figure 3: Word length versus log-frequency in the Brown corpus

References

Cox, R. T. (1946). Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13.

Griffiths, T. L. and Tenenbaum, J. B. (2007). From mere coincidences to meaningful discoveries. *Cognition*, 103(2):180–226.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Lecture 10: Introduction to linear models

7 November 2007

1 Linear models

Today and for much of the rest of the course we'll be looking at conditional probability distributions of the form

$$P(Y|X_1, X_2, \ldots, X_n)$$

that is, where we are interested in determining the distribution of a particular random variable Y given knowledge of the outcomes of a number of other random variables $X_1, \ldots, X_n$. Y is variously called the OUTCOME, RESPONSE, or DEPENDENT VARIABLE, and the Xi are called the PREDICTORS, INPUT VARIABLES, COVARIATES, or INDEPENDENT VARIABLES. In these notes I will tend to use the term RESPONSE for Y and PREDICTORS for the X_i. We will sometimes use $\vec{X}$ as a shorthand for $X_1, \ldots, X_n$.

In this lecture we will restrict our attention to models of a particular form:

$$P(Y|X) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon \tag{1}$$

for some fixed coefficients $\alpha, \beta_1, \beta_2, \ldots, \beta_n$.[1] Since the predictors X_i are known, then most of the right-hand side of Equation (1) is deterministic. However, ϵ, which is called the ERROR or NOISE term, is a random variable. Here, we assume that $\epsilon \sim \mathcal{N}(0, \sigma)$—that is, the error term is normally distributed with mean zero and some standard deviation σ.

[1] In some texts, the parameter α is instead denoted as β_0.

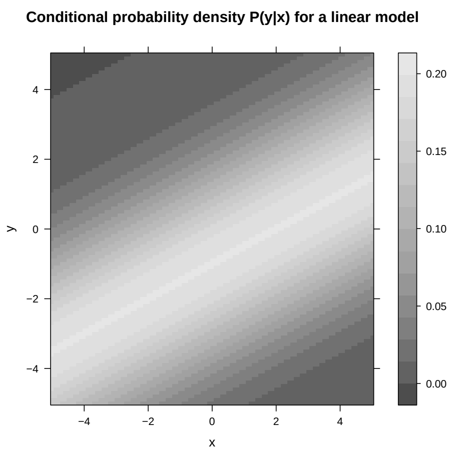


Figure 1: A plot of the probability density on the outcome of the Y random variable given the X random variable

This is what is called a LINEAR MODEL. Fitting a linear model to a dataset is called LINEAR REGRESSION. (If there is more than one predictor variable, it is often called MULTIPLE LINEAR REGRESSION.)

Before we get to fitting a linear model, let briefly us examine what type of distribution it captures. We assume the linear model with only one predictor variable, X_1, and set $\alpha = -1, \beta_1 = \frac{1}{2}, \sigma = 2$. This gives us the distribution:

$$Y = -1 + 2X_1 + \epsilon$$

We can visualize the probability density as in Figure 1, using the following code:

```
x <- seq(-5,5,by=0.1)
y <- seq(-5,5,by=0.1)
grid <- expand.grid(x=x,y=y)
grid$z <- dnorm(grid$y,mean=(1/2) * grid$x - 1, sd=2)
levelplot(z ~ x*y,grid, main="Conditional probability density P(y|x) for a linear
```

The lighter the region, the more probable for any given value of X that the observed data point will fall there. If you imagine a vertical line extending through the plot at $X = 0$, you will see that the plot along this line is lightest in color at $Y = -1 = \alpha$. This is the point at which ϵ takes its most probable value, 0. For this reason, α is also called the INTERCEPT parameter of the model, and the β_i are called the SLOPE parameters.

An important point to keep in mind is that the linear model is **NOT** a model of the distribution of X, but only the conditional distribution $P(Y|X)$.

2 Fitting a linear model

The process of estimating the parameters α and β_i of a linear model on the basis of some data (also called FITTING the model to the data) is called LINEAR REGRESSION. There are many techniques for parameter estimation in linear regression, but by far the most well-known and widespread technique is maximum-likelihood estimation.

Before we talk about exactly what the maximum-likelihood estimate looks like, we'll introduce some useful terminology. Suppose that we have estimated some model parameters $\hat{\alpha}$ and $\hat{\beta_i}$. This means that for each point $\langle x_j, y_j \rangle$ in our dataset $\vec{x}$, we can construct a PREDICTED VALUE for Y as follows:

$$\hat{y_j} = \alpha + \beta_1 x_{j1} + \dots \beta_n x_{jn}$$

where x_{ji} is the value of the i-th predictor variable for the j-th data point. (Note that no ϵ appears in the definition of $\hat{y_j}$, which is centered on the most likely outcome of Y in the predicted model.) We define the RESIDUAL of the j-th data point simply as

$$y_j - \hat{y_j}$$

—that is, the amount by which our model's prediction missed the observed value.

It turns out that for linear models with a normally-distributed error term ϵ, the likelihood of the model parameters with respect to $\hat{x}$ is monotonic in the sum of the squared residuals. This means that the maximum-likelihood estimate of the parameters is also the estimate that minimizes the the sum of the squared residuals.

2.1 Fitting a linear model: case study

The dataset `english` contains reaction times for lexical decision and naming of isolated English words, as well as written frequencies for those words. Reaction times are measured in milliseconds, and word frequencies are measured in appearances in a 17.9-million word written corpus. (All these variables are recorded in log-space) It is well-established that words of high textual frequency are generally responded to more quickly than words of low textual frequency. Let us consider a linear model in which reaction time RT depends on the log-frequency, F, of the word:

$$RT = \alpha + \beta_F F + \epsilon \tag{2}$$

This linear model corresponds to a FORMULA in R, which can be specified in either of the following ways:

An Introduction to R, section 11.1

```
RT ~ F
RT ~ 1 + F
```

The 1 in the latter formula refers to the intercept of the model; the presence of an intercept is implicit in the first formula.

You can fit a linear model with the `lm()`, `abline()` command. The code below displays the relationship between lexical-decision reaction time and written word frequency for the `english` dataset, as seen in Figure 2:[2]

lm()

```
> plot(exp(RTlexdec) ~ WrittenFrequency, data=english)
> rt.lm <- lm(exp(RTlexdec) ~ WrittenFrequency, data=english)
> rt.lm

Call:
lm(formula = exp(RTlexdec) ~ WrittenFrequency, data = english)

Coefficients:
     (Intercept)  WrittenFrequency
          843.58            -26.97

> abline(rt.lm,col=2,lwd=4)
```

The result of the linear regression is an intercept $\alpha = 843.58$ and a slope $\beta_F = -29.76$. The `WrittenFrequency` variable is in natural log-space, so the slope can be interpreted as saying that if two words differ in frequency by a

[2] We use `exp(RTlexdec)` because the variable `RTlexdec` is recorded in log-space.

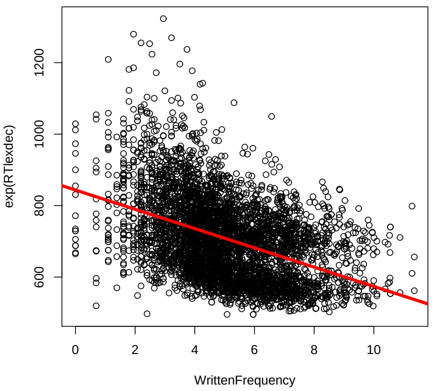


Figure 2: Lexical decision reaction times as a function of word frequency

factor of $e \approx 2.718$, then on average the more frequent word will be recognized as a word of English 26.97 milliseconds faster than the less frequent word. The intercept, 843.58, is the predicted reaction time for a word whose log-frequency is 0—that is, a word occurring only once in the corpus.

2.2 Fitting a linear model: a simple example

Let us now break down how the model goes about fitting data in a simple example.

Suppose we have only three observations of log-frequency/RT pairs:

$$\langle 4, 800 \rangle$$
$$\langle 6, 775 \rangle$$
$$\langle 8, 700 \rangle$$

Let use consider four possible parameter estimates for these data points. Three estimates will draw a line through two of the points and miss the third; the last estimate will draw a line that misses but is reasonably close to all the points. The code is below, and results in Figure 3:

```
x <- c(4,6,8)
```

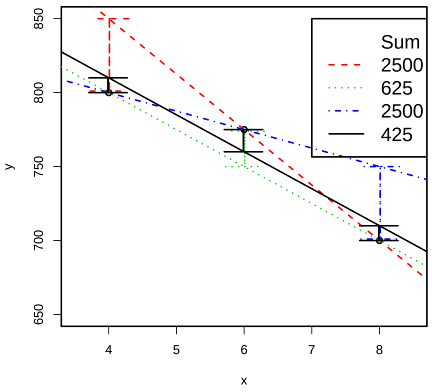


Figure 3: Linear regression with three points

```
y <- c(800,775,700)
old.par <- par(lwd=2)
plot(x,y,xlim=c(3.5,8.5),ylim=c(650,850))
abline(1000,-37.5, lty=2,col=2) # goes through points 2 & 3
errbar(4.01,801,4.01,850,lty=2,col=2)
abline(900,-25, lty=3,col=3) # goes through points 1 & 3
errbar(6.01,750,6.01,774,lty=3,col=3)
abline(850,-12.5, lty=4,col=4) # goes through points 2 & 3
errbar(8.01,750,8.01,701,lty=4,col=4)
abline(910,-25, lty=1,col=1) # goes through points 1 & 3
errbar(3.99,810,3.99,800,lty=1,col=1)
errbar(5.99,760,5.99,775,lty=1,col=1)
errbar(7.99,710,7.99,700,lty=1,col=1)
legend(7,850,c("Sum squared error","2500","625","2500","425"),
  lty=c(NA,2,3,4,1),col=c(NA,2,3,4,1),cex=1.5)
par(old.par)
```

First consider the solid black line, which has intercept 910 and slope -25. It predicts the following values, missing all three points:

x	$\hat{y}$	Residual ($\hat{y} - y$)
4	810	-10
6	760	15
8	710	-10

and the sum of its squared residuals is 425. Each of the other three lines has only one non-zero residual, but that residual is much larger, and in all three case, the sum of squared residuals is larger than for the solid black line. This means that the likelihood of the parameter values $\alpha = 910, \beta_F = -25$ is higher than the likelihood of the parameters corresponding to any of the other lines.

What is the MLE for α, β_F with respect to these three data points, and what are the residuals for the MLE? We can use `lm()` to fit the model and `resid()` to get at the residuals:

`resid()`

```
> rt3.lm <- lm(y ~ x)
> rt3.lm

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
      908.3        -25.0

> resid(rt3.lm)
        1         2         3
-8.333333 16.666667 -8.333333
> sum(resid(rt3.lm)^2)
[1] 416.6667
```

The MLE has the same slope as our solid black line, though the intercept is ever so slightly lower. The sum of squared residuals is slightly better too.

Take-home point: for linear regression, getting everything wrong by a little bit is better than getting a few things wrong by a lot.

2.3 Outliers and leverage

exclude???

3 Handling multiple predictors

In many cases, we are interested in simultaneously investigating the linear influence of two or more predictor variables on a single response. We'll discuss two methods of doing this: RESIDUALIZING and MULTIPLE LINEAR REGRESSION.

As a case study, consider naming reaction times from the `english` dataset, and now imagine that we're interested in the influence of orthographic neighbors. (An orthographic neighbor of a word w is one that shares most of its letters with w; for example, *cat* has several orthographic neighbors including *mat* and *rat*.) The `english` dataset summarizes this information in the `Ncount` variable, which measures ORTHOGRAPHIC NEIGHBORHOOD DENSITY as (I believe) the number of maximally close orthographic neighbors that the word has. How can we investigate the role of orthographic neighborhood while simultaneously taking into account the role of word frequency?

3.1 Residualizing

One approach would be a two-step process: first, construct a linear regression with frequency as the predictor and RT as the response. (This is commonly called "regressing RT against frequency".) Second, construct a new linear regression with neighborhood density as the predictor the *residuals from the first regression* as the response. The transformation of a raw RT into the residual from a linear regression is called RESIDUALIZATION. The whole process is illustrated below (see Figure 4):

```
english.young <- subset(english,AgeSubject=="young")
attach(english.young)
rt.freq.lm <- lm(exp(RTnaming) ~ WrittenFrequency)
rt.freq.lm

Call:
lm(formula = exp(RTnaming) ~ WrittenFrequency)

Coefficients:
     (Intercept)  WrittenFrequency
         486.506            -3.307
```

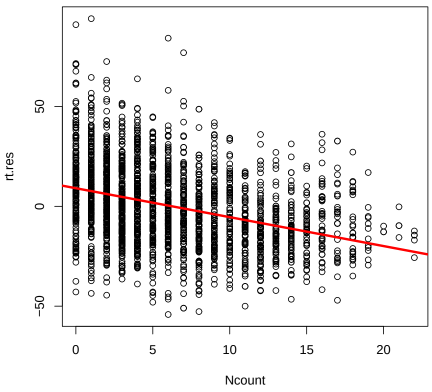


Figure 4: Plot of frequency-residualized word naming times and linear regression against neighborhood density

```
rt.res <- resid(rt.freq.lm)
rt.ncount.lm <- lm(rt.res ~ Ncount)
plot(Ncount, rt.res)
abline(rt.ncount.lm,col=2,lwd=3)
rt.ncount.lm

Call:
lm(formula = rt.res ~ Ncount)

Coefficients:
(Intercept)       Ncount
      9.080       -1.449

detach()
```

Even after linear effects of frequency have been accounted for by removing them from the RT measure, neighborhood density still has some effect – words with higher neighborhood density are named more quickly.

3.2 Multiple linear regression

The alternative is to build a single linear model with more than one predictor. A linear model predicting naming reaction time on the basis of both frequency F and neighborhood density D would look like this:

$$RT = \alpha + \beta_F F + \beta_D D + \epsilon$$

and the corresponding `R` formula would be either of the following:

```
RT ~ F + D
RT ~ 1 + F + D
```

Plugging this in gives us the following results:

```
attach(english)
rt.both.lm <- lm(exp(RTnaming) ~ WrittenFrequency + Ncount)
rt.both.lm

Call:
lm(formula = exp(RTnaming) ~ WrittenFrequency + Ncount)

Coefficients:
     (Intercept)  WrittenFrequency            Ncount
         602.367            -5.042            -1.780

detach()
```

Note that the results are qualitatively similar but quantitatively different than for the residualization approach: larger effect sizes have been estimated for both `WrittenFrequency` and `Ncount`.

Lecture 11: Confidence intervals and model comparison for linear regression; analysis of variance

14 November 2007

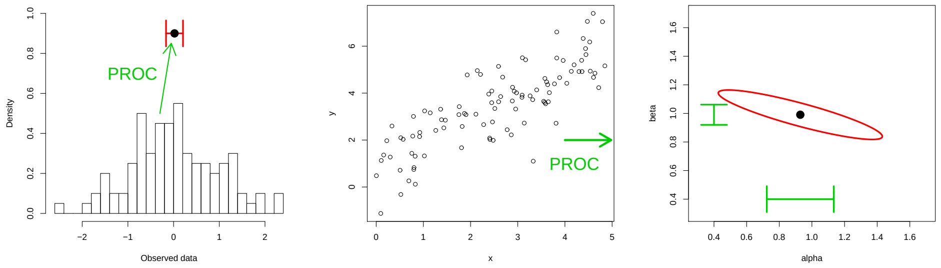
1 Confidence intervals and hypothesis testing for linear regression

Just as there was a close connection between hypothesis testing with the one-sample *t*-test and a confidence interval for the mean of a sample, there is a close connection between hypothesis testing and confidence intervals for the parameters of a linear model. We'll start by explaining the confidence interval as the fundamental idea, and see how this leads to hypothesis tests.

`ellipse()` from the `ellipse` package

```
#Sample mean
dat <- rnorm(100,0,1)
hist(dat,breaks=20,prob=T, ylim=c(0,1),xlab="Observed data",main="",cex=2)
arrows(t.test(dat)$conf.int[1],0.9,t.test(dat)$conf.int[2],0.9,
  code=3,angle=90,lwd=3,col=2)
points(mean(dat),0.9,pch=19,cex=2)
arrows(-0.3,0.5,-0.05,0.85,lwd=2,col=3)
text(-0.9,0.7,"PROC",cex=2,col=3)

#intercept and slope of a linear model
x <- runif(100,0,5)
y <- x + 1 + rnorm(100)
plot(x,y)
arrows(4,2,4.98,2,lwd=4,col=3)
```



(a) (b) (c)

Figure 1: The confidence-region construction procedure for (a) sample means and (b) parameters of a linear model. The black dots are the maximum-likelihood estimates, around which the confidence regions are centered.

```
text(4.2,1,"PROC",cex=2,col=3)
xy.lm <- lm(y ~ x)
plot(ellipse(vcov(xy.lm),centre=coef(xy.lm)),ylim=c(0.3,1.7),
  xlim=c(0.3,1.7),type="l",col=2,lwd=3,xlab="alpha",ylab="beta",cex=2)
points(coef(xy.lm)[1],coef(xy.lm)[2],cex=2,pch=19)
arrows(0.4,coef(xy.lm)[2]+sqrt(vcov(xy.lm)[2,2]),0.4,
  coef(xy.lm)[2]-sqrt(vcov(xy.lm)[2,2]),code=3,angle=90,lwd=3,col=3)
arrows(coef(xy.lm)[1]+sqrt(vcov(xy.lm)[1,1]),0.4,
  coef(xy.lm)[1]-sqrt(vcov(xy.lm)[1,1]),0.4,code=3,angle=90,lwd=3,col=3)
```

Figure 1 illustrates the procedures by which confidence intervals are constructed for a sample mean (one parameter) and for the intercept and slope of a linear regression with one predictor. In both cases, a dataset $\vec{x}$ is obtained, and a fixed procedure is used to construct boundaries of a CONFIDENCE REGION from $\vec{x}$. In the case of the sample mean, the "region" is in one-dimensional space so it is an interval. In the case of a linear regression model, the region is in two-dimensional space, and looks like an ellipse. The size and shape of the ellipse are determined by the VARIANCE-COVARIANCE MATRIX of the linear predictors, which is accessible via the `vcov()` function. If we collapse the ellipse down to only one dimension (corresponding to one of the linear model's parameters), we have a confidence interval on that parameter.[1]

`vcov()`

[1] Formally this corresponds to marginalizing over the values of the other parameters that you're collapsing over.

We illustrate this below for the linear regression model of frequency against word naming latency:

```
> attach(english)
> rt.lm <- lm(exp(RTnaming) ~ WrittenFrequency)
> summary(rt.lm)

Call:
lm(formula = exp(RTnaming) ~ WrittenFrequency)

Residuals:
     Min       1Q   Median       3Q      Max
-150.855  -97.030   -5.047   92.876  227.199

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      593.7061     4.3065 137.864  < 2e-16 ***
WrittenFrequency  -5.5376     0.8051  -6.878  6.9e-12 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 100.3 on 4566 degrees of freedom
Multiple R-Squared: 0.01025,Adjusted R-squared: 0.01004
F-statistic: 47.31 on 1 and 4566 DF,  p-value: 6.895e-12

> vcov(rt.lm) # The diagonals are the variances of the coeff. estimates!
                 (Intercept) WrittenFrequency
(Intercept)         18.54573       -3.2548502
WrittenFrequency    -3.25485        0.6482287
> plot(ellipse(vcov(rt.lm),centre=coef(rt.lm)),type="l")
> points(coef(rt.lm)[1],coef(rt.lm)[2],cex=5,col=2,pch=19)
```

Figure 2 shows the results. The model is quite certain about the parameter estimates; however, note that there is a correlation between the parameter estimates. According to the analysis, if we reran this regression many times on data drawn from the same population, whenever the resulting intercept (i.e., average predicted RT for the rarest class of word) is higher, the facilitative effect of written frequency would tend to be larger, and vice versa.

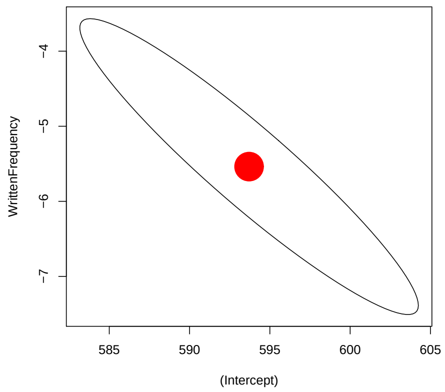


Figure 2: Confidence ellipse for parameters of regression of word naming latency against written frequency

2 Comparing models in multiple linear regression

Recall that the Neyman-Pearson paradigm involves specifying a null hypothesis H_0 and determining whether to reject it in favor of a more general and complex hypothesis H_A. In many cases, we are interested in comparing whether a more complex linear regression is justified by the data over a simpler regression. Under these circumstances, we can take the simpler model M_0 as the null hypothesis, and the more complex model M_A as the alternative hypothesis.

In these cases, a beautiful property of classical linear models is taken advantage of to compare M_0 and M_A. Recall that the VARIANCE of a sample is simply the sum of the square deviations from the mean:[2]

$$\mathrm{Var}(\vec{y}) = \sum_j (y_j - \bar{y})^2 \tag{1}$$

where $\bar{y}$ is the mean of the sample $\vec{y}$. For any model M that predicts values $\hat{y}_j$ for the data, the RESIDUAL VARIANCE of M is quantified in exactly the

[2] We're using $\vec{y}$ instead of $\vec{x}$ here to emphasize that it is the values of the response, and not of the predictors, that are relevant.

same way:

$$\mathrm{Var}_M(\vec{y}) = \sum_j (y_j - \hat{y_j})^2 \tag{2}$$

The beautiful thing about linear models is that the sample variance can be split apart, or *partitioned*, into (a) the component that is explained by M, and (b) the component that remains unexplained by M. This can be written as follows:

$$\mathrm{Var}(\vec{y}) = \overbrace{\sum_j (y_j - \hat{y_j})^2}^{\mathrm{Var}_M(\vec{y})} + \overbrace{\sum_j (\hat{y_j} - \bar{y})^2}^{\text{unexplained}} \tag{3}$$

Furthermore, if two models are nested (i.e., one is a special case of the other), then the variance can be futher subdivided among those two models. Figure 3 shows the partitioning of variance for two nested models.

2.1 Comparing linear models: the F test statistic

If M_0 has k_0 parameters and M_A has k_A parameters, then the F statistic, defined below, is widely used for testing the models against one another:[3]

$$F = \frac{\sum_j (\hat{y_j^A} - \hat{y_j^0})^2/(k_A - k_0)}{\sum_j (y_j - \hat{y_j^A})^2/(n - k_A - 1)} \tag{4}$$

where $\hat{y_j^A}$ and $\hat{y_j^0}$ are the predicted values for the j-th data point in models M_A and M_0 respectively.

The F statistic can also be written as follows:

$$F = \frac{\sum_j (y_j - \hat{y_j^0})^2 - \sum_j (y_j - \hat{y_j^A})^2/(k_A - k_0)}{\sum_j (y_j - \bar{y})^2 - \sum_j (y_j - \hat{y_j^A})^2/(n - k_A - 1)} \tag{5}$$

[3] The F statistic and the F distribution, which is a ratio of two χ^2 random variables, is named after Ronald A. Fisher, one of the founders of classical statistics, who worked out the importance of the statistic in the context of formulating the analysis of variance.

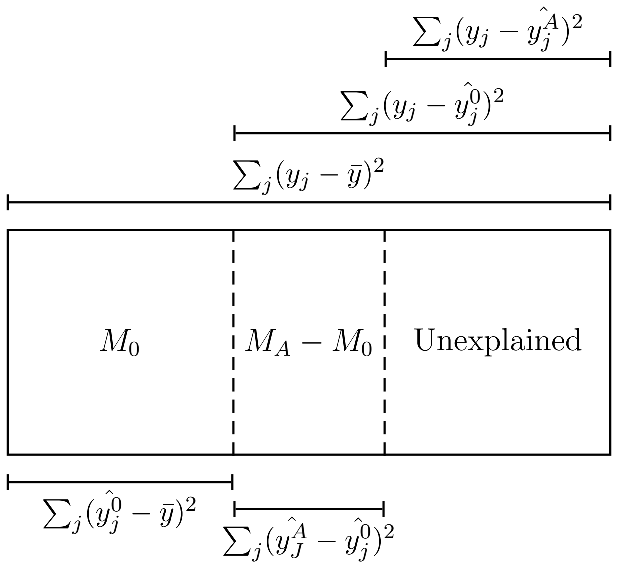

$\sum_j (y_j - \hat{y_j^A})^2$
$\sum_j (y_j - \hat{y_j^0})^2$
$\sum_j (y_j - \bar{y})^2$
M_0
$M_A - M_0$
Unexplained
$\sum_j (\hat{y_j^0} - \bar{y})^2$
$\sum_j (\hat{y_J^A} - \hat{y_j^0})^2$

Figure 3: The partitioning of residual variance in linear models. Symbols in the box denote the variance explained by each model; the sums outside the box quantify the variance in each combination of sub-boxes.

Take a look at the labels on the boxes in Figure 3 and convince yourself that the sums in the numerator and the denominator of the F statistic correspond respectively to the boxes $M_A - M_0$ and Unexplained. Because of this, using the F statistic for hypothesis testing is often referred to as evaluation of the RATIO OF THE SUMS OF SQUARES.

Because of its importance for linear models, the distribution of the F statistic has been worked out in detail and is accessible in `R` through the `{d,p,q,r}f()` functions.

`{d,p,q,r}f()`

2.2 Model comparison: case study

In the previous lecture we looked at the effect of written frequency and neighborhood density on word naming latency. We can use the F statistic to compare models with and without the neighborhood-density predictor `Ncount`. The `anova()` function is a general-purpose tool for comparison of nested models in `R`.

`anova()`

```
> attach(english)
> rt.mean.lm <- lm(exp(RTnaming) ~ 1)
> rt.both.lm <- lm(exp(RTnaming) ~ WrittenFrequency + Ncount)
> rt.freq.lm <- lm(exp(RTnaming) ~ WrittenFrequency)
> anova(rt.both.lm,rt.freq.lm)
Analysis of Variance Table

Model 1: exp(RTnaming) ~ WrittenFrequency + Ncount
Model 2: exp(RTnaming) ~ WrittenFrequency
  Res.Df      RSS   Df Sum of Sq      F    Pr(>F)
1   4565 45599393
2   4566 45941935   -1   -342542 34.292 5.075e-09 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> detach()
```

As you can see, model comparison by the F statistic rejects the simpler hypothesis that only written frequency has a predictive effect on naming latency.

3 Analysis of Variance

Recall that we just covered linear models, which are conditional probability distributions of the form

$$P(Y|X) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon \qquad (6)$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. We saw how this paradigm can be put to use for modeling the predictive relationship of continuous variables, such as word frequency, familiarity, and neighborhood density, on reaction times in word recognition experiments.

In many cases, however, the predictors of interest are not continuous. For example, for the `english` dataset in `languageR` we might be interested in how naming times are influenced by the type of the initial phoneme of the word. This information is coded by the `Frication` variable of the dataset, and has the following categories:

`burst`	the word starts with a burst consonant
`frication`	the word starts with a fricative consonant
`long`	the word starts with a long vowel
`short`	the word starts with a short vowel

It is not obvious how these categories might be meaningfully arranged on the real number line. Rather, we would simply like to investigate the possibility that the mean naming time differs as a function of initial phoneme type.

The most widespread technique used to investigate this type of question is the ANALYSIS OF VARIANCE (often abbreviated ANOVA). Although many books go into painstaking detail covering different instances of ANOVA, you can gain a firm foundational understanding of the core part of the method by thinking of it as a special case of multiple linear regression.

3.1 Dummy variables

Let us take the example above, where `Frication` is a categorical predictor. Categorical predictors are often called FACTORS, and the values they take are often called LEVELS. (This is also the nomenclature used in `R`.) In order to allow for the possibility that each level of the factor could have arbitrarily different effects on mean naming latency, we can create DUMMY PREDICTOR VARIABLES, one per level of the factor:

Level of `Frication`	X_1	X_2	X_3	X_4
`burst`	1	0	0	0
`frication`	0	1	0	0
`long`	0	0	1	0
`short`	0	0	0	1

(Variables such as these which are 0 unless a special condition holds, in which case they are 1, are often referred to as INDICATOR VARIABLES). We then construct a standard linear model with predictors X_1 through X_4:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \epsilon \tag{7}$$

When we combine the dummy predictor variables with the linear model in (6), we get the following equations for each level of `Frication`:

Level of `Frication`	Linear model
`burst`	$Y = \alpha + \beta_1 + \epsilon$
`frication`	$Y = \alpha + \beta_2 + \epsilon$
`long`	$Y = \alpha + \beta_3 + \epsilon$
`short`	$Y = \alpha + \beta_4 + \epsilon$

This linear model thus allows us to code a different predicted mean (and most-likely predicted value) for each level of the predictor, by choosing different values of α and β_i.

However, it should be clear from the table above that only four distinct means can be predicted in this linear model—one for each level of `Frication`. We don't need five parameters (one for α and four for the β_i) to encode four means; one of the parameters is redundant. This is problematic when fitting the model because it means that there is no unique maximum-likelihood estimate.[4] To eliminate this redundancy, we arbitrarily choose one level of the factor as the BASELINE level, and we don't introduce a dummy predictor for the baseline level. If we choose `burst` as the baseline level,[5] then we can eliminate X_4, and make X_1, X_2, X_3 dummy indicator variables for `frication`, `long`, and `short` respectively, giving us the linear model

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{8}$$

where predicted means for the four classes are as follows:[6]

Level of `Frication`	Predicted mean
`burst`	α
`frication`	$\alpha + \beta_1$
`long`	$\alpha + \beta_2$
`short`	$\alpha + \beta_3$

3.2 Analysis of variance as model comparison

Now that we have completed the discussion of using dummy variables to construct a linear model with categorical predictors (i.e., factors), we shall move on to discussing what analysis of variance actually *does*. Consider that we now have two possible models of how word-initial frication affects naming time. We have the model of Equation (8) above, in which each class of frication predicts a different mean naming time, with noise around the mean distributed the same way for each class. We might also consider a simpler

[4] For example, if $\alpha = 0, \beta_1 = \beta_2 = \beta_3 = \beta_4 = 1$ is a maximum-likelihood estimate, then $\alpha = 1, \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$ is as well because it encodes exactly the same model.

[5] By default, `R` chooses the first level of a factor as the baseline, and the first level of a factor is whatever level comes first alphabetically unless you specified otherwise when the factor was constructed—see the `levels` argument of the function `factor()` in the documentation.

[6] This choice of coding for the dummy variables is technically known as the choice of CONTRAST MATRIX. The choice of contrast matrix described here is referred to as the TREATMENT contrast matrix, or `contr.treatment` in `R`.

model in which frication has no effect on naming time. Such a model looks as follows:

$$Y = \alpha + \epsilon$$

Now look again at Figure 3 and think of the simpler model of Equation (**??**) as M_0, and the more complex model of Equation (8) as M_A. (Actually, the simpler model explains *no* variance because it just encodes the mean, but it still illustrates the general picture.) Because ANOVA is just another kind of linear model, we can perform a hypothesis test between M_0 and M_A by constructing an F statistic from the ratio of the amount of variance contained in the boxes $M_A - M_0$ and Unexplained. The simpler model has one parameter and the more complex model has four, so we use Equation (4) with $k_0 = 1, k_A = 4$ to construct the F statistic. Below is code to perform this hypothesis test using `lm()` just as we have before:

```
> m.0 <- lm(exp(RTnaming) ~ 1, english)
> m.A <- lm(exp(RTnaming) ~ 1 + Frication, english)
> anova(m.0, m.A)
Analysis of Variance Table

Model 1: exp(RTnaming) ~ 1
Model 2: exp(RTnaming) ~ 1 + Frication
  Res.Df      RSS   Df Sum of Sq      F    Pr(>F)
1   4567 46417913
2   4564 46076418    3    341494 11.275 2.287e-07 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

As we can see, the F statistic is high enough to easily reject the null-hypothesis model that frication has no effect on naming latency.

Classical ANOVA is encoded in `R` using the `aov()` function. We can `aov()` replicate the results above using this function:

```
> summary(aov(exp(RTnaming) ~ Frication, english))
              Df   Sum Sq  Mean Sq F value    Pr(>F)
Frication      3   341494   113831  11.275 2.287e-07 ***
Residuals   4564 46076418    10096
```

```
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Note that the F statistic reported is identical for the two approaches.

3.2.1 Interpreting parameter estimates in ANOVA

One thing that is often not taught in classical ANOVA is to inspect parameter estimates and how to interpret them. In fact, there is a tradition in software implementations of ANOVA to hide the parameter estimates from the user! This tradition is carried over into `aov()`. However, because ANOVA is just a linear model, we can use `lm()` to investigate the model parameters and interpret the differences between the factors.

```
>  summary(lm(exp(RTnaming) ~ 1 + Frication, english))

Call:
lm(formula = exp(RTnaming) ~ 1 + Frication, data = english)

Residuals:
     Min       1Q   Median       3Q      Max
-162.315  -97.418   -4.621   92.597  242.182

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         566.228      2.342 241.732  < 2e-16 ***
Fricationfrication    8.388      3.401   2.466 0.013696 *
Fricationlong        -9.141     10.964  -0.834 0.404469
Fricationshort      -14.909      3.973  -3.752 0.000177 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 100.5 on 4564 degrees of freedom
Multiple R-Squared: 0.007357,Adjusted R-squared: 0.006704
F-statistic: 11.28 on 3 and 4564 DF,  p-value: 2.287e-07
```

Recall that the baseline level is `burst`. This means that intercept parameter is the estimated mean naming latency for burst-initial words, and each of the parameter estimates for the three levels listed in the summary—`frication`,

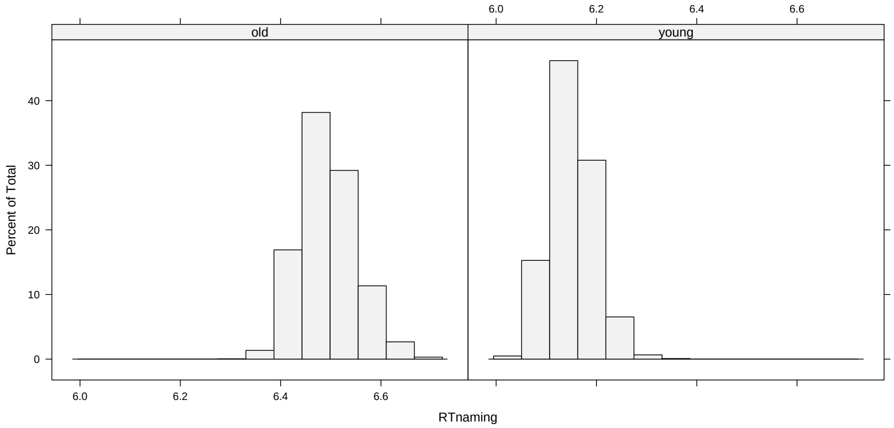


Figure 4: Histogram of naming latencies for young (ages ∼ 22.6) versus old (ages > 60 speakers)

`long`, and `short`—is to be interpreted as the *difference* between the mean naming latencies for bursts and for the level in question, as in Equation (8). That is, the estimated mean latency for fricative-initial words is higher than for bursts, at 574.616, whereas the estimated mean latencies for long and short vowels are shorter than for bursts. So vowel-initial words are in general named more quickly than consonant-initial words. In addition, the t values in the summary indicate the significance of the observed difference between the `burst` level and the level in question. So short vowel-initial words are highly significantly different in mean naming latency than bursts; fricatives are also significantly different,[7] but mean latency for initial long vowels is not significantly different from initial bursts.

3.3 Testing for interactions

The `english` dataset includes average naming latencies not only for college-age speakers but also for speakers age 60 and over. This degree of age difference turns out to have a huge effect on naming latency (Figure 4):

```
histogram(~ RTnaming | AgeSubject, english)
```

[7] We're omitting discussion of the problem of multiple hypothesis comparisons for now, which might affect conclusions about frication versus bursts. If you're interested in this issue, the Wikipedia article on the Bonferroni correction is a good place to start.

Clearly, college-age speakers are faster at naming words than speakers over age 60. We may be interested in including this information in our model. In Lecture 10 we already saw how to include both variables in a multiple regression model. Here we will investigate an additional possibility: that different levels of frication may have different effects on mean naming latency depending on speaker age. For example, we might think that fricatives, which our linear model above indicates are the hardest class of word onsets, might be even harder for elderly speakers than they are for the young. When these types of inter-predictor contingencies are included in a statistical model they are called interactions.

It is instructive to look explicitly at the linear model that results from introducing interactions between multiple categorical predictors. We will take `old` as the baseline value of speaker age, and leave `burst` as the baseline value of frication. This means that the "baseline" predictor set involves an old-group speaker naming a burst-initial word, and the intercept α will express the predicted mean latency for this combination. There are seven other logically possible combinations of age and frication; thus our full model will have to have seven dummy indicator variables, each with its own parameter. There are many ways to set up these dummy variables; we'll cover perhaps the most straightforward way. In addition to $X_{\{1,2,3\}}$ for the non-baseline levels of frication, we add a new variable X_4 for the non-baseline levels of speaker age (`young`). This set of dummy variables allows us to encode all eight possible groups, but it doesn't allow us to estimate separate parameters for all these groups. To do this, we need to add three more dummy variables, one for each of the non-baseline frication levels when coupled with the non-baseline age level. This gives us the following complete set of codings:

Frication	Age	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
burst	old	0	0	0	0	0	0	0	
frication	old	1	0	0	0	0	0	0	
long	old	0	1	0	0	0	0	0	
short	old	0	0	1	0	0	0	0	
burst	young	0	0	0	1	0	0	0	
frication	young	1	0	0	1	1	0	0	
long	young	0	1	0	1	0	1	0	
short	young	0	0	1	1	0	0	1	

We can test this full model against a strictly additive that allows for effects of both age and initial phoneme class, but not for interactions—that is, one with only $X_{\{1,2,3,4\}}$. In `R`, the formula syntax `Frication*AgeSpeaker` indicates that an interaction between the two variables should be included in the model.

```
> m.0 <- lm(exp(RTnaming) ~ Frication + AgeSubject, english)
> m.A <- lm(exp(RTnaming) ~ Frication * AgeSubject, english)
> anova(m.0,m.A)
Analysis of Variance Table

Model 1: exp(RTnaming) ~ Frication + AgeSubject
Model 2: exp(RTnaming) ~ Frication * AgeSubject
  Res.Df     RSS   Df Sum of Sq      F  Pr(>F)
1   4563 3977231
2   4560 3970213    3      7019 2.6871 0.04489 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Note that there are three more parameters in the model with interactions than in the additive model, which fits degrees of freedom listed in the analysis of variance table. As we can see, there is some evidence that frication interacts with speaker age. We get the same result with `aov()`:

(1)

```
> summary(aov(exp(RTnaming) ~ Frication * AgeSubject, english))
                       Df   Sum Sq  Mean Sq    F value  Pr(>F)
Frication               3   341494   113831   130.7414 < 2e-16 ***
AgeSubject              1 42099187 42099187 48353.1525 < 2e-16 ***
Frication:AgeSubject    3     7019     2340     2.6871 0.04489 *
Residuals            4560  3970213      871
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

3.4 ANOVA in its more general form

Although the picture in Figure 3 is the way that linear model comparison is classically done, and is appropriate for the ANOVA comparisons that we have looked at so far, the partitioning of the variance in ANOVA can get more complicated. The call to `aov()` we just made in (1) for the interaction between `Frication` and `AgeSubject` actually partitioned the variance as

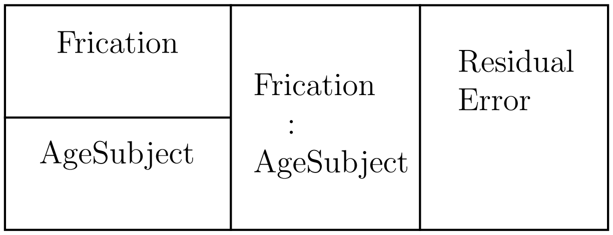


Figure 5: Partitioning the variance in a basic two-way ANOVA

shown in Figure 5. In each line of the summary for (1), the variance inside the box corresponding to the predictor of interest is being compared with the Residual Error box in Figure 5. The ratio of mean squares is F-distributed in all these cases. One of the somewhat counterintuitive consequences of this approach is that you can test for main effects of one predictor (say `Frication`) while accounting for idiosyncratic interactions of that predictor with another variable.

Lecture 12: Analysis of variance in real life

19 November 2007

1 Lecture 11 continued

Everything in this section really belongs in Lecture 11.

1.1 Testing for interactions

The `english` dataset includes average naming latencies not only for college-age speakers but also for speakers age 60 and over. This degree of age difference turns out to have a huge effect on naming latency (Figure 1):

```
histogram(~ RTnaming | AgeSubject, english)
```

Clearly, college-age speakers are faster at naming words than speakers over age 60. We may be interested in including this information in our model. In Lecture 10 we already saw how to include both variables in a multiple regression model. Here we will investigate an additional possibility: that different levels of frication may have different effects on mean naming latency depending on speaker age. For example, we might think that fricatives, which our linear model above indicates are the hardest class of word onsets, might be even harder for elderly speakers than they are for the young. When these types of inter-predictor contingencies are included in a statistical model they are called INTERACTIONS.

It is instructive to look explicitly at the linear model that results from introducing interactions between multiple categorical predictors. We will take `old` as the baseline value of speaker age, and leave `burst` as the baseline value of frication. This means that the "baseline" predictor set involves an

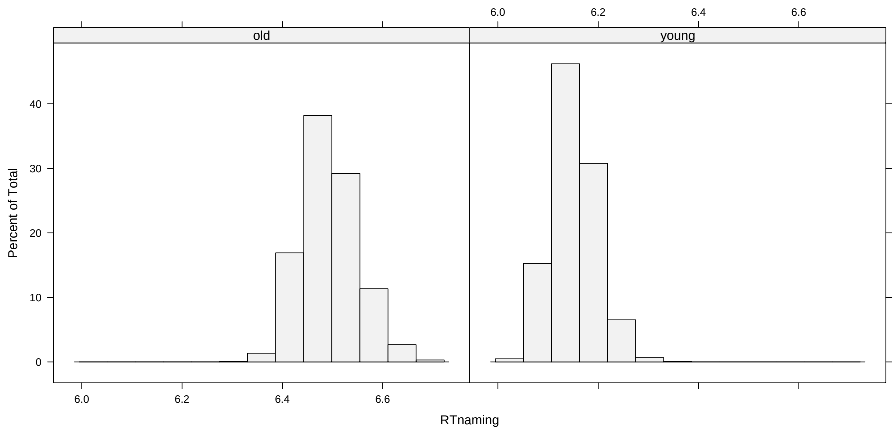


Figure 1: Histogram of naming latencies for young (ages ~ 22.6) versus old (ages > 60 speakers)

old-group speaker naming a burst-initial word, and the intercept α will express the predicted mean latency for this combination. There are seven other logically possible combinations of age and frication; thus our full model will have to have seven dummy indicator variables, each with its own parameter. There are many ways to set up these dummy variables; we'll cover perhaps the most straightforward way. In addition to $X_{\{1,2,3\}}$ for the non-baseline levels of frication, we add a new variable X_4 for the non-baseline levels of speaker age (`young`). This set of dummy variables allows us to encode all eight possible groups, but it doesn't allow us to estimate separate parameters for all these groups. To do this, we need to add three more dummy variables, one for each of the non-baseline frication levels when coupled with the non-baseline age level. This gives us the following complete set of codings:

Frication	Age	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
burst	old	0	0	0	0	0	0	0	
frication	old	1	0	0	0	0	0	0	
long	old	0	1	0	0	0	0	0	
short	old	0	0	1	0	0	0	0	
burst	young	0	0	0	1	0	0	0	
frication	young	1	0	0	1	1	0	0	
long	young	0	1	0	1	0	1	0	
short	young	0	0	1	1	0	0	1	

We can test this full model against a strictly ADDITIVE that allows for effects of both age and initial phoneme class, but not for interactions—that is, one with only $X_{\{1,2,3,4\}}$. In R, the formula syntax `Frication*AgeSpeaker` indicates that an interaction between the two variables should be included in the model.

```
> m.0 <- lm(exp(RTnaming) ~ Frication + AgeSubject, english)
> m.A <- lm(exp(RTnaming) ~ Frication * AgeSubject, english)
> anova(m.0,m.A)
Analysis of Variance Table

Model 1: exp(RTnaming) ~ Frication + AgeSubject
Model 2: exp(RTnaming) ~ Frication * AgeSubject
  Res.Df     RSS   Df Sum of Sq      F  Pr(>F)
1   4563 3977231
2   4560 3970213    3      7019 2.6871 0.04489 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Note that there are three more parameters in the model with interactions than in the additive model, which fits degrees of freedom listed in the analysis of variance table. As we can see, there is some evidence that frication interacts with speaker age. We get the same result with `aov()`:

(1)

```
> summary(aov(exp(RTnaming) ~ Frication * AgeSubject, english))
                       Df   Sum Sq  Mean Sq    F value  Pr(>F)
Frication               3   341494   113831   130.7414 < 2e-16 ***
AgeSubject              1 42099187 42099187 48353.1525 < 2e-16 ***
Frication:AgeSubject    3     7019     2340     2.6871 0.04489 *
Residuals            4560  3970213      871
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

1.2 ANOVA in its more general form

Although the picture in Figure **??** is the way that linear model comparison is classically done, and is appropriate for the ANOVA comparisons that we

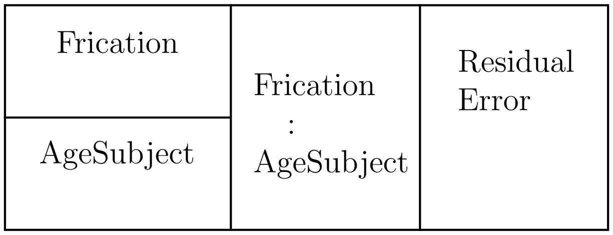


Figure 2: Partitioning the variance in a basic two-way ANOVA

have looked at so far, the partitioning of the variance in ANOVA can get more complicated. The call to `aov()` we just made in (1) for the interaction between `Frication` and `AgeSubject` actually partitioned the variance as shown in Figure 2. In each line of the summary for (1), the variance inside the box corresponding to the predictor of interest is being compared with the Residual Error box in Figure 2. The ratio of mean squares is F-distributed in all these cases. One of the somewhat counterintuitive consequences of this approach is that you can test for main effects of one predictor (say `Frication`) while accounting for idiosyncratic interactions of that predictor with another variable.

2 A bit more on the F distribution

By popular demand, here's a bit more about the F distribution. There's really not much to say about this distribution except that, crucially, it is the distribution of the ratio of two χ^2 random variables. Because the variance of a sample is distributed as a χ^2 random variable, the ratio of variances in linear models can be compared to the F distribution.

More formally, if $U \sim \chi^2_m$ and $V \sim \chi^2_n$, we have

$$F_{m,n} \sim \frac{U/m}{V/n} \tag{1}$$

It is useful to play a bit with the F distribution to see what it looks like. In general, the cumulative distribution is more interesting and pertinent than the probability density function (unless you have an anomalously low F statistic).

3 A case study

This is the outcome of a self-paced reading experiment conducted by Hannah Rohde, in collaboration with me and Andy Kehler.

The question under investigation is whether certain kinds of verbs (*implicit causality* (IC) *verbs*) such as "detest", which intuitively demand some sort of explanation, can affect readers' online syntactic attachment preferences.

(2) a. John **detests** the children of the musician who **is** generally arrogant and rude (IC,LOW)
b. John **detests** the children of the musician who **are** generally arrogant and rude (IC,HIGH)
c. John **babysits** the children of the musician who **is** generally arrogant and rude (NONIC,LOW)
d. John **babysits** the children of the musician who **are** generally arrogant and rude (NONIC,HIGH)

Hannah hypothesized that the use of an IC verb should facilitate reading of high-attached RCs, which are generally found in English to be harder to read than low-attached RCs (Cuetos and Mitchell, 1988). The reasoning here is that the IC verbs demand an explanation, and one way of encoding that explanation linguistically is through a relative clause. In these cases, the most plausible type of explanation will involve a clause in which the object of the IC verb plays a role, so an RC modifying the IC verb's object should become more expected. This stronger expectation may facilitate processing when such an RC is seen (Levy, 2007).

The stimuli for the experiment consist of 20 quadruplets of sentences of the sort above. Such a quadruplet is called an EXPERIMENTAL ITEM in the language of experimental psychology. The four different variants of each item are called the CONDITIONS. Since a participant who sees one of the sentences in a given item is liable to be strongly influenced in her reading of another sentence in the item, the convention is only to show each item once to a given participant. To achieve balance, each participant will be shown five items in each condition.

	Item					
Participant	1	2	3	4	5	...
1	ic,high	nonIc,high	ic,low	nonIc,low	ic,high	...
2	nonIc,low	ic,high	nonIc,high	ic,low	nonIc,low	...
3	ic,low	nonIc,low	ic,high	nonIc,high	ic,low	...
4	nonIc,high	ic,low	nonIc,low	ic,high	nonIc,high	...
5	ic,high	nonIc,high	ic,low	nonIc,low	ic,high	...
⋮	⋮	⋮	⋮	⋮	⋮	⋱

The experimental data will be analyzed for effects of verb type and attachment level, and more crucially for an *interaction* between these two effects. For this reason, we plan to conduct a two-way ANOVA.

In self-paced reading, the observable effect of difficulty at a given word often shows up a word or two downstream, so in this case we will focus on the first word after the disambiguator—i.e., “generally”. This is called the FIRST SPILLOVER REGION. First we read in the complete dataset, zoom in on the results at this region, and look at the distribution of reading times for each condition. A BOXPLOT (also known as a BOX-AND-WHISKERS diagram) is a good tool for this kind of visualization, and for identifying outliers.[1]

```
dat1 <- read.table("results.final.txt",
  quote="",sep="\t",header=T)
dat1 <- subset(dat1,subj != "subj2" & subj != "subj10"
             & subj != "subj50") # these subjects answered questions at chance
dat1$subj  <- factor(dat1$subj) # eliminate these levels from the factor
spillover.1 <- subset(dat1,expt==1 & correct ==1
         & crit=="RC_VERB+1") # focus on first spillover region,
                              # only correctly-answered sentences
```

[1] In a boxplot, the upper and lower bounds of the box are the first and third quartile of the data; the length of this box is called the INTER-QUARTILE RANGE, or IQR. The solid-line “whiskers” are placed at the farthest points that lie no more than 1.5 × IQR from the edges of the box. Any points that lie beyond these whiskers are considered “outliers” and plotted individually.

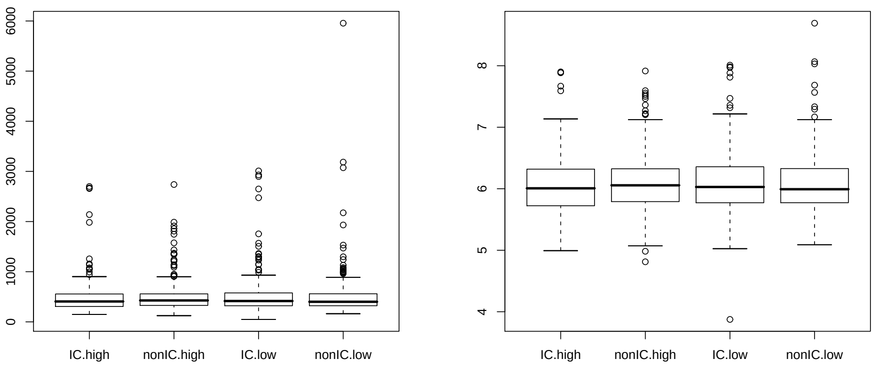


Figure 3: Boxplots for raw and log reading times at the first spillover region for the IC-RC experiment

```
spillover.1$verb <- factor(spillover.1$verb) # remove "NONE"
                                              # levels of factor
spillover.1$attachment <- factor(spillover.1$attachment)
boxplot(rt ~ verb*attachment,spillover.1)
boxplot(log(rt) ~ verb*attachment,spillover.1)
```

As you can see in Figure 3, there are lots of outliers – though the situation looks better in log-space. Histograms reveal similar results (Figure 4):

```
> spillover.1$cond <- factor(spillover.1$cond)
> histogram(~ rt | cond, spillover.1,breaks=30)
```

It is by no means obvious what to do in this kind of situation where there are so many outliers, and the raw response departs so severely from normality. In the case of self-paced reading, the dominant convention is to perform OUTLIER REMOVAL: use some relatively standardized criterion for identifying outliers, and deal with those outliers in some way.

There are several different ways outlier removal is handled in the literature; in this situation we shall apply one of the more common procedures. For each condition, we calculate for each point a Z-SCORE, which is a measure of how many sample standard deviations the point lies away from the sample mean. Points with a z-score of magnitude above 4 are simply thrown away:

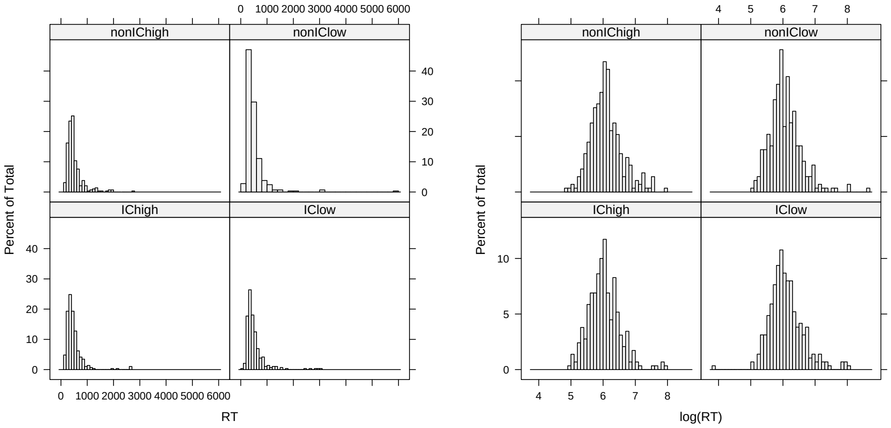


Figure 4: Histograms for raw and log reading times at the first spillover region for the IC-RC experiment

```
> cbind.list <- function(l) {
    result <- c()
    for(i in 1:length(l)) {
      result <- cbind(result,l[[i]])
    }
    result
  }
> get.z.score <- function(response,conds.list) {
    means <- tapply(response,conds.list,mean)
    sds <- tapply(response,conds.list,sd)
    (response - means[cbind.list(conds.list)]) /
          sds[cbind.list(conds.list)]
  }
> z <- with(spillover.1,get.z.score(rt,list(verb,attachment)))
> sum(abs(z) > 4) # 14 points are flagged this way as outliers
[1] 14
> length(z) # 14 out of 933 is a lot of outliers at 4sd!!!!
            # But this is typical for self-paced reading experiments
[1] 933
> spillover.1.to.analyze <- subset(spillover.1,abs(z) <= 4)
```

We take a quick look at what our resulting data look like (some fields of the data frame have been omitted):

```
> head(spillover.1.to.analyze)
   Subj Item  Verb Attachment      Crit     RT
9     1    1    IC       high RC_VERB+1 365.27
22   10    1 nonIC        low RC_VERB+1 616.43
35   11    1    IC        low RC_VERB+1 255.56
48   12    1 nonIC       high RC_VERB+1 626.26
61   13    1    IC       high RC_VERB+1 330.45
74   14    1 nonIC        low RC_VERB+1 434.66
```

and now we are ready to conduct our two-way ANOVA.

4 The comparisons to make

In this experiment, four factors characterize each stimulus: a particular subject reads a particular item that appears with particular values of the `verb` and `attachment` manipulations. `verb` and `attachment` have two levels each, so if we had m subjects and n items we would in principle need at least $2 \times 2 \times m \times n$ observations to consider a full classic linear model with interactions of all possible types. However, because each subject saw each item only once, we only have $m \times n$ observations. Therefore it is not possible to construct the full model.

For many years dating back to Clark (1973), the gold standard in this situation has been to construct two separate analyses: one for subjects, and one for items. In the analysis over subjects, we take as our individual data points the *mean* value of all the observations in each cell of Subject × Verb × Attachment—that is, we AGGREGATE, or average, across items. Correspondingly, in the analysis over items, we aggregate across subjects. We can use the function `aggregate()` to perform this averaging:

`aggregate()`

`with()`

```
sp.1.subj <- with(spillover.1.to.analyze,aggregate(list(rt=rt),
  list(subj=subj,verb=verb,attachment=attachment),mean))
sp.1.item <- with(spillover.1.to.analyze,aggregate(list(rt=rt),
  list(item=item,verb=verb,attachment=attachment),mean))
```

The view of the resulting data for the analysis over subjects can be seen in Table 1. This setup is called a WITHIN-SUBJECTS OR REPEATED-MEASURES

		Subject					
Verb	Attachment	1	2	3	4	5	...
IC	High	280.7	396.1	561.2	339.8	546.1	...
	Low	256.3	457.8	547.3	408.9	594.1	...
nonIC	High	340.9	507.8	786.7	369.8	453.0	...
	Low	823.7	311.4	590.4	838.3	298.9	...

Table 1: Repeated-measures (within-subjects) view of item-aggregated data for subjects ANOVA

design because each subject participates in each condition—or, in another manner of speaking, we take multiple measurements for each subject. Designs in which, for some predictor factor, each subject participates in only one condition are called between-subjects designs.

The way we partition the variance for this type of analysis can be seen in Figure 5. Because we have averaged things out so we only have one observation per Subject/Verb/Attachment combination, there will be no variation in the Residual Error box. Each test for an effect of a predictor sets of interest (`verb`, `attachment`, and `verb:attachment`) is performed by comparing the variance explained by the predictor set P with the variance associated with arbitrary random interactions between the subject and P. This is equivalent to performing a model comparison between the following two linear models, where i range over the subjects and j over the conditions in P:

$$rt_{ij} = \alpha + B_i \mathrm{Subj_i} + \epsilon_{ij} \qquad \text{(null hypothesis)} \qquad (2)$$

$$rt_{ij} = \alpha + B_i \mathrm{Subj_i} + \beta_j P_j + \epsilon_{ij} \qquad \text{(alternative hypothesis)} \qquad (3)$$

$$(4)$$

There is an added wrinkle here, which is that the B_i are not technically free parameters but rather are themselves assumed to be random and normally distributed. However, this difference does not really affect the picture here. (In a couple of weeks, when we get to mixed-effects models, this difference will become more prominent and we'll learn how to handle it in a cleaner and more unified way.)

Fortunately, `aov()` is smart enough to know to perform all these model comparisons in the appropriate way, by use of the `Error()` specification in your model formula. This is done as follows, for subjects:

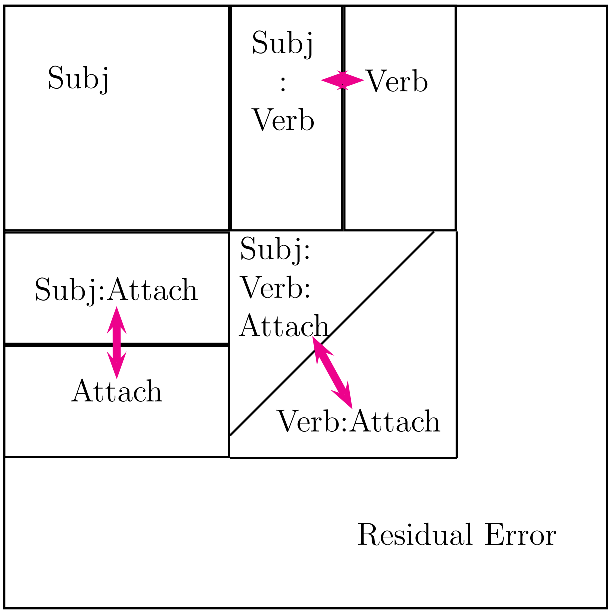


Figure 5: The picture for this 2 × 2 ANOVA, where Verb and Attachment are the fixed effects of interest, and subjects are a random factor

```
> summary(aov(rt ~ verb * attachment
    + Error(subj/(verb *attachment)), sp.1.subj))

Error: subj
          Df  Sum Sq Mean Sq F value Pr(>F)
Residuals 54 4063007   75241

Error: subj:verb
          Df Sum Sq Mean Sq F value  Pr(>F)
verb       1  48720   48720  7.0754 0.01027 *
Residuals 54 371834    6886
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Error: subj:attachment
           Df Sum Sq Mean Sq F value Pr(>F)
attachment  1    327     327  0.0406  0.841
Residuals  54 434232    8041
```

```
Error: subj:verb:attachment
                Df Sum Sq Mean Sq F value  Pr(>F)
verb:attachment  1  93759   93759  6.8528 0.01146 *
Residuals       54 738819   13682
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

and for items:

```
> summary(aov(rt ˜ verb * attachment
    + Error(item/(verb *attachment)), sp.1.item))

Error: item
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 19 203631   10717

Error: item:verb
          Df Sum Sq Mean Sq F value Pr(>F)
verb       1  21181   21181  3.5482  0.075 .
Residuals 19 113419    5969
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Error: item:attachment
           Df Sum Sq Mean Sq F value Pr(>F)
attachment  1    721     721   0.093 0.7637
Residuals  19 147299    7753

Error: item:verb:attachment
                Df Sum Sq Mean Sq F value  Pr(>F)
verb:attachment  1  38211   38211  5.4335 0.03092 *
Residuals       19 133615    7032
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Fortunately, the by-subjects and by-items analysis yield largely similar results: they both point towards (a) a significant main effect of verb type;

and (b) more interestingly, a significant interaction between verb type and attachment level. To interpret these, we need to look at the means of each condition. It is conventional in psychological experimentation to show the condition means from the aggregated data for the by-subjects analysis:

```
> with(sp.1.subj,tapply(rt,list(verb),mean))
      IC    nonIC
452.2940 482.0567
> with(sp.1.subj,tapply(rt,list(verb,attachment),mean))
          high      low
IC    430.4316 474.1565
nonIC 501.4824 462.6309
```

The first spillover region was read more quickly in the implicit-causality verb condition than in the non-IC verb condition. The interaction was a crossover interaction: in the high attachment conditions, the first spillover region was read more quickly for IC verbs than for non-IC verbs; but for the low attachment conditions, reading was faster for non-IC verbs than for IC verbs.

We interpreted this result to indicate that IC verbs do indeed facilitate processing of high-attaching RCs, to the extent that this becomes the preferred attachment level.

References

Clark, H. H. (1973). The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of Verbal Learning and Verbal Behavior*, 12:335–359.

Cuetos, F. and Mitchell, D. C. (1988). Cross-linguistic differences in parsing: Restrictions on the use of the Late Closure strategy in Spanish. *Cognition*, 30(1):73–105.

Levy, R. (2007). Expectation-based syntactic comprehension. *Cognition*. In press.

Lecture 13: Introduction to generalized linear models

21 November 2007

1 Introduction

Recall that we've looked at linear models, which specify a conditional probability density $P(Y|X)$ of the form

$$Y = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n + \epsilon \tag{1}$$

Linear models thus assume that the only stochastic part of the data is the normally-distributed noise ϵ around the predicted mean. Yet many (most?) types of data do not meet this assumption at all. These include:

- Continuous data in which noise is not normally distributed;
- Count data, in which the outcome is restricted to non-negative integers;
- Categorical data, where the outcome is one of a number of discrete classes.

One of the important developments in statistical theory over the past several decades has been the broadening of linear models from the classic form given in Equation (1) to encompass a much more diverse class of probabilistic distributions. This is the class of GENERALIZED LINEAR MODELS (GLMs). The next section will describe, step by step, how the generalization from classic linear models is attained.

1.1 Generalizing the classic linear model

The right-hand side of Equation (1) has two components: a deterministic component determining the *predicted mean*, and a stochastic component expressing the *noise distribution* around that mean:

$$Y = \overbrace{\alpha + \beta_1 X_1 + \cdots + \beta_n X_n}^{\text{Predicted Mean}} + \overbrace{\epsilon}^{\text{Noise}(\sim\mathcal{N}(0,\sigma^2))} \tag{2}$$

The first step from classic linear models to generalized linear models is to break these two components apart and specify a more indirect functional relationship between them. In the first step, we start with the idea that for any particular set of predictor variables $\{X_i\}$, there is a predicted mean μ. The probability distribution on the response Y is a function of that μ.[1] We'll review here what this means for linear models, writing both the abbreviated form of the model and the resulting probability density on the response Y:

$$Y = \mu + \epsilon \tag{3}$$

$$p(Y = y; \mu) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu)^2}{\sigma^2}} \tag{4}$$

$$\tag{5}$$

By choosing other functions to map from μ to $p(y)$, we can get to other probability distributions, such as the Poisson distribution over the non-negative integers (see Lecture 2, section 5):

$$P(Y = y; \mu) = \frac{e^{-\mu}}{y!}\mu^y \tag{6}$$

In the second step, we loosen the relationship between the predicted mean and the predictor variables. In the classic linear model of Equation (1), the predicted mean was a linear combination of the predictor variables. In generalized linear models, we call this linear combination η and allow the

[1] There is actually a further constraint on the functional relationship between μ and $f(y)$, which I'm not going into—see McCullagh and Nelder (1989) or Venables and Ripley (2002, Chapter 7) for more details.

predicted mean to be an invertible function of η. We call η the LINEAR PREDICTOR, and call the function relating μ to η the LINK FUNCTION:

$$\eta = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n \qquad \text{(linear predictor)} \tag{7}$$

$$l(\mu) = \eta \qquad \text{(link function)} \tag{8}$$

In classic linear regression, the link function is particularly simple: it is the identity function, so that $\eta = \mu$.

Summary: generalized linear models are a broad class of models predicting the outcome of a response as a function of some linear combination of a set of predictors. To define a GLM, you need to choose (a) a link function relating the linear predictor to the predicted mean of the response; and (b) a function defining the "noise" or "error" probability distribution around that mean. For a classical linear model, the link function is the identity function

1.2 Logistic regression as a generalized linear model

Suppose we want a GLM that models binomially distributed data from n trials. We will use a slightly different formulation of the binomial distribution from what we introduced in Lecture 2: instead of viewing the response as the number of successful trials r, we view the response as the *proportion* of successful trials $\frac{r}{n}$; call this Y. Now, the mean number of successes for a binomial distribution is pn; hence the mean proportion is p. Thus p is the predicted mean μ of our GLM. This gives us enough information to specify precisely the resulting model:

$$P(Y = y; \mu) = \binom{n}{yn} \mu^{ny} (1-\mu)^{n(1-y)} \qquad \text{(or equivalently, replace } \mu \text{ with } p) \tag{9}$$

This should look familiar from Lecture 2, Section 2.

This is part (b) of designing a GLM: choosing the distribution on Y given the mean μ. Having done this means that we have placed ourselves in the BINOMIAL GLM FAMILY. The other part of specifying our GLM is (a): choosing a relationship between the linear predictor η and the mean μ. Unlike the case with the classical linear model, the identity link function is not a possibility, because η can potentially be any real number, whereas the mean proportion μ of successes can only vary between 0 and 1. There are many link functions that can be chosen to make this mapping valid, but here we will use the LOGIT link function (here we replace μ with p for simplicity):[2]

$$\log \frac{p}{1-p} = \eta \tag{10}$$

or equivalently,

$$p = \frac{e^{\eta}}{1+e^{\eta}} \tag{11}$$

When we plunk the full form of the linear predictor from Equation (7) back in, we arrive at the final formula for logistic regression:

Logistic regression formula:

$$p = \frac{e^{\alpha+\beta_1 X_1+\cdots+\beta_n X_n}}{1+e^{\alpha+\beta_1 X_1+\cdots+\beta_n X_n}} \tag{12}$$

This type of model is also called a LOGIT MODEL.

1.3 Fitting a simple logistic regression model

The most common criterion by which a logistic regression model for a dataset is exactly the way that we chose the parameter estimates for a linear regression model: the method of maximum likelihood. That is, we choose the parameter estimates that give our dataset the highest likelihood.

We will give a simple example using the `dative` dataset. The response variable here is whether the recipient was realized as an NP (i.e., the double-object construction) or as a PP (i.e., the prepositional object construction). This corresponds to the `RealizationOfRecipient` variable in the dataset. There are several options in `R` for fitting basic logistic regression models, including `glm()` in the `stats` package and `lrm()` in the `Design` package. In this case we will use `lrm()`. We will start with a simple study of the effect of recipient pronominality on the dative alternation. Before fitting a model, we examine a contingency table of the outcomes of the two factors:

[2] Two other popular link functions for binomial GLMs are the PROBIT link and the COMPLEMENTARY LOG-LOG link. See Venables and Ripley (2002, Chapter 7) for more details.

```
> xtabs(~ PronomOfRec + RealizationOfRecipient,dative)
               RealizationOfRecipient
PronomOfRec       NP   PP
  nonpronominal  600  629
  pronominal    1814  220
```

So sentences with nonpronominal recipients are realized roughly equally often with DO and PO constructions; but sentences with pronominal recipients are recognized nearly 90% of the time with the DO construction. We expect our model to be able to encode these findings.

It is now time to construct the model. To be totally explicit, we will choose ourselves which realization of the recipient counts as a "success" and which counts as a "failure" (although `lrm()` will silently make its own decision if given a factor as a response). In addition, our predictor variable is a factor, so we need to use dummy-variable encoding; we will satisfice with the `R` default of taking the alphabetically first factor level, `nonpronominal`, as the baseline level.

```
> response <- ifelse(dative$RealizationOfRecipient=="PP",
                1,0) # code PO realization as success, DO as failure
> lrm(response ~ PronomOfRec, dative)

Logistic Regression Model

lrm(formula = response ~ PronomOfRec, data = dative)



Frequencies of Responses
   0    1
2414  849

       Obs  Max Deriv Model L.R.       d.f.          P          C        Dxy
      3263      2e-12     644.08          1          0      0.746      0.492
     Tau-a         R2      Brier
      0.19      0.263      0.154

                       Coef    S.E.    Wald Z P
Intercept               0.0472 0.05707   0.83 0.4082
PronomOfRec=pronominal -2.1569 0.09140 -23.60 0.0000
```

The thing to pay attention to for now is the estimated coefficients for the intercept and the dummy indicator variable for a pronominal recipient. We can use these coefficients to determine the values of the linear predictor η and the predicted mean success rate p using Equations (7) and (12):

$$\eta_{--} = 0.0472 + (-2.1569) \times 0 \quad = 0.0472 \quad \text{(non-pronominal receipient)} \tag{13}$$

$$\eta_{+} = 0.0472 + (-2.1569) \times 1 \quad = -2.1097 \quad \text{(pronominal recipient)} \tag{14}$$

$$p_{\text{nonpron}} = \frac{e^{0.0472}}{1 + e^{0.0472}} \quad = 0.512 \tag{15}$$

$$p_{\text{pron}} = \frac{e^{-2.1097}}{1 + e^{-2.1097}} \quad = 0.108 \tag{16}$$

When we check these predicted probabilities of PO realization for nonpronominal and pronominal recipients, we see that they are equal to the proportions seen in the corresponding rows of the cross-tabulation we calculated above: $\frac{629}{629+600} = 0.518$ and $\frac{220}{220+1814} = 0.108$. This is exactly the expected behavior, because (a) we have two parameters in our model, α and β_1, which is enough to encode an arbitrary predicted mean for each of the cells in our current representation of the dataset; and (b) as we have seen before (Lecture 5, Section 2), the maximum-likelihood estimate for a binomial distribution is the relative-frequency estimate—that is, the observed proportion of successes.

1.4 Multiple logistic regression

Just as we were able to perform multiple linear regression for a linear model with multiple predictors, we can perform multiple logistic regression. Suppose that we want to take into account pronominality of both recipient and theme. First we conduct a complete cross-tabulation and get proportions of PO realization for each combination of pronominality status: `apply()`

```
> tab <- xtabs(~ RealizationOfRecipient + PronomOfRec + PronomOfTheme, dative)
> tab
, , PronomOfTheme = nonpronominal

                      PronomOfRec
```

```
RealizationOfRecipient nonpronominal pronominal
                    NP           583       1676
                    PP           512         71

, , PronomOfTheme = pronominal

                      PronomOfRec
RealizationOfRecipient nonpronominal pronominal
                    NP            17        138
                    PP           117        149
> apply(tab,c(2,3),function(x) x[2] / sum(x))
               PronomOfTheme
PronomOfRec     nonpronominal pronominal
  nonpronominal     0.4675799  0.8731343
  pronominal        0.0406411  0.5191638
```

Pronominality of the theme consistently increases the probability of PO realization; pronominality of the recipient consistently increases the probability of DO realization.

We can construct a logit model with independent effects of theme and recipient pronominality as follows:

```
> dative.lrm < - lrm(response ~ PronomOfRec + PronomOfTheme, dative)
> dative.lrm

Logistic Regression Model

lrm(formula = response ~ PronomOfRec + PronomOfTheme, data = dative)


Frequencies of Responses
   0    1
2414  849

       Obs  Max Deriv Model L.R.       d.f.          P          C        Dxy
      3263      1e-12    1122.32          2          0      0.827      0.654
     Tau-a         R2      Brier
     0.252      0.427      0.131
```

```
                            Coef    S.E.    Wald Z P
Intercept                   -0.1644 0.05999  -2.74 0.0061
PronomOfRec=pronominal      -2.8670 0.12278 -23.35 0.0000
PronomOfTheme=pronominal     2.9769 0.15069  19.75 0.0000
```

And once again, we can calculate the predicted mean success rates for each of the four combinations of predictor variables:

Recipient	Theme	η	$\hat{p}$
nonpron	nonpron	-0.1644	0.459
pron	nonpron	-3.0314	0.046
nonpron	pron	2.8125	0.943
pron	pron	-0.0545	0.486

In this case, note the predicted proportions of success are not the same as the observed proportions in each of the four cells. This is sensible – we cannot fit four arbitrary means with only three parameters. If we added in an interactive term, we would be able to fit four arbitrary means, and the resulting predicted proportions would be the observed proportions for the four different cells.

1.5 Multiplicativity of the odds

Let us consider the case of a dative construction in which both the recipient and theme are encoded with pronouns. In this situation, both the dummy indicator variables (indicating that the theme and recipient are pronouns) have a value of 1, and thus the linear predictor consists of the sum of three terms. From Equation (10) we can write

$$\frac{p}{1-p} = e^{\alpha+\beta_1+\beta_2} \tag{17}$$

$$= e^{\alpha}e^{\beta_1}e^{\beta_2} \tag{18}$$

The ratio $\frac{p}{1-p}$ is the odds of success, and in logit models the effect of any predictor variable on the response variable is multiplicative in the odds of success. If a predictor has coefficent β in a logit model, then a unit of that predictor has a multiplicative effect of e^{β} on the odds of success.

Unlike the raw coefficient β, the quantity e^{β} is not linearly symmetric—it falls in the range $(0, \infty)$. However, we can also perform the full reverse

Predictor	Coefficient	Factor Weight	Multiplicative effect on odds
Intercept	-0.1644	0.4590	0.8484
Pronominal Recipient	-2.8670	0.0538	0.0569
Pronominal Theme	2.9769	0.9515	19.627

Table 1: Logistic regression coefficients and corresponding factor weights for each predictor variable in the `dative` dataset.

Recip.	Theme	Linear Predictor	Multiplicative odds	P(PO)
–pron	–pron	-0.16	0.8484	0.46
+pron	–pron	$-0.16 - 2.87 = -3.03$	$0.85 \times 0.06 = 0.049$	0.046
–pron	+pron	$-0.16 + 2.98 = 2.81$	$0.85 \times 19.6 = 16.7$	0.94
+pron	+pron	$-0.16 - 2.87 + 2.98 = -0.05$	$0.85 \times 0.06 \times 19.63 = 0.947$	0.49

Table 2: Linear predictor, multiplicative odds, and predicted values for each combination of recipient and theme pronominality in the `dative` dataset. In each case, the linear predictor is the log of the multiplicative odds.

LOGIT TRANSFORM of Equation (11), mapping β to $\frac{e^\beta}{1+e^\beta}$ which ranges between zero and 1, and is linearly symmetric around 0.5. The use of logistic regression with the reverse logit transform has been used in quantitative sociolinguistics since Cedergren and Sankoff (1974) (see also Sankoff and Labov, 1979), and is still in widespread use in that field. In quantitative sociolinguistics, the use of logistic regression is often called VARBRUL (variable rule) analysis, and the parameter estimates are reported in the reverse logit transform, typically being called FACTOR WEIGHTS.

Tables 1 and 2 show the relationship between the components of the linear predictor, the components of the multiplicative odds, and the resulting predictions for each possible combination of our predictor variables.

2 Confidence intervals and model comparison in logit models

We'll close our introduction to logistic regression with discussion of confidence intervals and model comparison.

2.1 Confidence intervals for logistic regression

When there are a relatively large number of observations in comparison with the number of parameters estimated, the standardized deviation of the MLE for a logit model parameter θ is approximately normally distributed:

$$\frac{\hat{\theta} - \theta}{\text{StdErr}(\hat{\theta})} \overset{\text{approx}}{\widehat{\sim}} \mathcal{N}(0, 1) \tag{19}$$

This is called the Wald statistic[3]—note the close similarity with the t statistic that we were able to use for classic linear regression in Lecture 11 (remember that once the t distribution has a fair number of degrees of freedom, it looks a great deal like a standard normal distribution). If we look again at the output of the logit model we fitted in the previous section, we see the standard error, which allows us to construct confidence intervals on our model parameters.

```
                         Coef    S.E.    Wald Z P
Intercept                -0.1644 0.05999  -2.74 0.0061
PronomOfRec=pronominal   -2.8670 0.12278 -23.35 0.0000
PronomOfTheme=pronominal  2.9769 0.15069  19.75 0.0000
```

The Wald statistic can also be used for a frequentist test on the null hypothesis that an individual model parameter is 0. This is the source of the p-values given for the model parameters above.

2.2 Model comparison

Just as in the analysis of variance, we are often interested in conducting tests of the hypothesis that introducing *several* model parameters simultaneously leads to a better overall model. In this case, we cannot simply use a single Wald statistic for hypothesis testing. Instead, the most common approach is to use the likelihood-ratio test. A generalized linear model assigns a likelihood to its data as follows:

[3] It is also sometimes called the Wald Z statistic, because of the convention that standard normal variables are often denoted with a Z, and the Wald statistic is distributed approximately as a standard normal.

$$\text{Lik}(\vec{x};\hat{\theta}) = \prod_i P(x_i|\hat{\theta}) \tag{20}$$

Now suppose that we have two classes of models, M_0 and M_1, and M_0 is nested inside of M_1 (that is, the class M_0 is a "special case" of the class M_1). It turns out that if the data are generated from a model M_0 is the correct model, the ratio of the data likelihoods in the ML estimates for M_0 and M_1 is well-behaved. In particular, twice the log of the likelihood ratio is distributed as a χ^2 random variable with degrees of freedom equal to the difference k in the number of free parameters in the two models. This quantity is sometimes called the DEVIANCE:

$$2\log\frac{\text{Lik}_{M_1}(\vec{x})}{\text{Lik}_{M_0}(\vec{x})} = 2\left[\log \text{Lik}_{M_1}(\vec{x}) - \log \text{Lik}_{M_0}(\vec{x})\right] \qquad \sim \chi^2_k \tag{21}$$

As an example of using the likelihood ratio test, we will hypothesize a model in which pronominality of theme and recipient both still have additive effects but that these effects may vary depending on the modality (spoken versus written) of the dataset. We fit this model and our modality-independent model using `glm()`, and use `anova()` to calculate the likelihood ratio: `glm()`

```
> m.0 <- glm(response ~ PronomOfRec + PronomOfTheme,dative,family="binomial")
> m.A <- glm(response ~ PronomOfRec*Modality + PronomOfTheme*Modality,dative,famil
> anova(m.0,m.A)
Analysis of Deviance Table

Model 1: response ~ PronomOfRec + PronomOfTheme
Model 2: response ~ PronomOfRec * Modality + PronomOfTheme * Modality
  Resid. Df Resid. Dev   Df Deviance
1      3260    2618.74
2      3257    2609.67    3     9.07
```

We can look up the p-value of this deviance result in the χ^2_3 distribution:

```
> 1-pchisq(9.07,3)
[1] 0.02837453
```

Thus there is some evidence that we should reject a model that doesn't include modality-specific effects of recipient and theme pronominality.

3 Further reading

There are many places to go for reading more about generalized linear models and logistic regression in particular. The classic comprehensive reference on generalized linear models is McCullagh and Nelder (1989). For GLMs on categorical data, Agresti (2002) and the more introductory Agresti (2007) are highly recommended. For more information specific to the use of GLMs and logistic regression in `R`, Venables and Ripley (2002, Section 7), Harrell (2001, Chapters 10–12), and Maindonald and Braun (2007, Section 8.2) are all good places to look.

References

Agresti, A. (2002). *Categorical Data Analysis*. Wiley.

Agresti, A. (2007). *An Introduction to Categorical Data Analysis*. Wiley, second edition.

Cedergren, H. J. and Sankoff, D. (1974). Variable rules: Performance as a statistical reflection of competence. *Language*, 50(2):333–355.

Harrell, Jr, F. E. (2001). *Regression Modeling Strategies*. Springer.

Maindonald, J. and Braun, J. (2007). *Data Analysis and Graphics using R*. Cambridge, second edition.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman & Hall, second edition.

Sankoff, D. and Labov, W. (1979). On the uses of variable rules. *Language in Society*, 8:189–222.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, fourth edition.

Lecture 14: Introduction to generalized mixed-effects models

26 November 2007

In this lecture we'll learn the following:

1. an introduction to mixed-effects modeling;
2. the multivariate normal distribution (in the middle).

1 Introduction to mixed-effects modeling

1.1 High-level motivation

In the (generalized) linear models we've looked at so far, we've assumed that the observations are independent of each other given the predictor variables. However, there are many situations in which that type of independence does not hold. One major type of situation violating these independence assumptions is CLUSTER-LEVEL ATTRIBUTES: when observations belong to different clusters and each cluster has its own properties (different response mean, different sensitivity to each predictor). We'll now cover MIXED-EFFECTS (also called MULTI-LEVEL) models, which are designed to handle this type of mutual dependence among datapoints. Common instances in which mixed-effects models can be used include:

- Education-related observations (e.g., vocabulary) made of students have multiple clusters at the level of city, school, teacher, and student;
- Observations related to linguistic behavior are clustered at the level of the speaker, and speaker-specific attributes might include different

baseline reading rates, differential sensitive to construction difficulty, or preference for one construction over another;

- Different sentences or even words may have idiosyncratic differences in their ease of understanding or production, and while we may not be able to model these differences, we may be able model the fact that there is incidental variation at the sentence or word level.

This lecture will introduce mixed-effects models, building on the mathematical tools you have acquired throughout the course, and then will cover a case study of mixed-effects linear models.

1.2 Technical introduction

Recall that we've looked at linear models, which specify a conditional probability density $P(Y|X)$ of the form

$$Y = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n + \epsilon \tag{1}$$

and thus $\alpha + \beta_1 X_1 + \cdots + \beta_n X_n$ is deterministic given a choice of $\{X_i\}$. Furthermore, we moved from linear models to generalized linear models (GLMs) by specifying (a) a LINK FUNCTION between the predicted mean μ and the linear predictor η:

$$l(\mu) = \eta = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n \tag{2}$$

and (b) an ERROR DISTRIBUTION FUNCTION specifying a probability distribution around the predicted mean:

$$\begin{aligned} p(y;\mu) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu)^2}{2\sigma^2}} && \text{(Normal)} && (3)\\ P(y;\mu) &= e^{-\mu}\frac{\mu^y}{y!} && \text{(Poisson)} && (4)\\ P(y;\mu) &= \binom{n}{yn}\mu^{yn}(1-\mu)^{(1-y)n} && \text{(Binomial)} && (5) \end{aligned}$$

We will now add one more wrinkle. In the generalized linear models we have seen so far, the linear predictor is entirely deterministic given a choice of $\{X_i\}$. In multilevel modeling, we introduce a stochastic component to the linear predictor:

$$\eta = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n + b_0 + b_1 Z_1 + \cdots + b_m Z_m \tag{6}$$

where the $\{Z_i\}$ are predictor variables just like the $\{X_i\}$ (in fact, they can be and often are the same predictor variables), but the vector $\langle b_0, b_1, \dots, b_m \rangle \equiv \vec{\mathbf{b}}$ is stochastic and follows some joint probability distribution. The linear predictor is often split into two components: the fixed effects $\alpha + \beta_1 X_1 + \cdots + \beta_n X_n$, and the random effects $b_0 + b_1 Z_1 + \cdots + b_m Z_m$. In principle, $\vec{\mathbf{b}}$ can have any distribution, but most work considers only the situation in which it follows a multivariate normal distribution.

1.3 The multivariate normal distribution

We never got around to covering the multivariate normal distribution early on in class, but we need it for mixed-effect modeling so we'll do it now. We'll start with just a 2-dimensional, or bivariate, distribution. Whereas the univariate (1-dimensional) normal distribution was characterized by two parameters—mean μ and variance σ^2—the bivariate normal distribution is characterized by two mean parameters (μ_X, μ_Y), two variance terms (one for the X axis and one for the Y axis), and one *covariance term* showing the tendency for X and Y to go together. The three variance and covariance terms are often grouped together into a symmetric covariance matrix as follows:

$$\begin{bmatrix} \sigma^2_{XX} & \sigma^2_{XY} \\ \sigma^2_{XY} & \sigma^2_{YY} \end{bmatrix}$$

Note that the terms σ^2_{XX} and σ^2_{YY} are simply the variances in the X and Y axes (the subscripts appear doubled, XX, for notational consistency). The term σ^2_{XY} is the covariance between the two axes.

cbind(),
function(),
outer(),
dmvnorm()

```
library(mvtnorm)
sigma.xx <- 4
sigma.yy <- 1
```

Figure 1: Two-dimensional multivariate normal distributions, without & with X-Y correlation

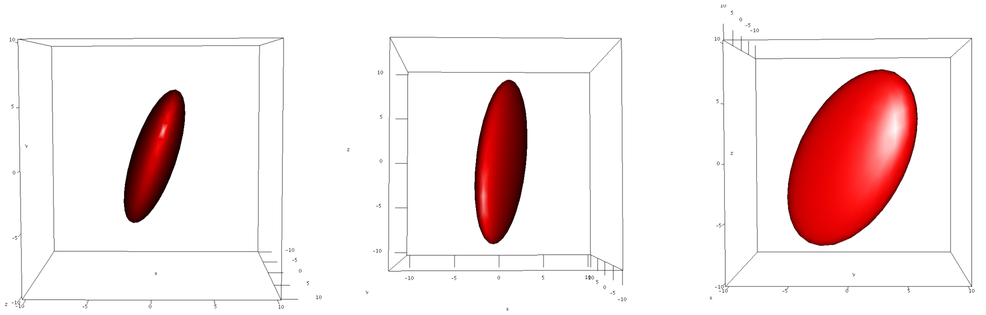
```
sigma.xy <- 0 # no covariance
sigma <- matrix(c(sigma.xx,sigma.xy,sigma.xy,sigma.yy),ncol=2)
old.par <- par(mfrow=c(1,2))
x <- seq(-5,5,by=0.25)
y <- x
f <- function(x,y)  {
  #cat("X: ", x,"\n")
  #cat("Y:", y," \n")
  xy <- cbind(x,y)
  #cat("XY: ", xy,"\n")
  dmvnorm(xy,c(0,0),sigma)
}
z <- outer(x,y,f)
persp(x, y, z, theta = 30, phi = 30, expand = 0.5, col = "lightblue",
  ltheta = 120, shade = 0.75)
contour(x, y, z, method = "edge",xlab="X",ylab="Y")
par(old.par)
# do the same thing again with sigma.xy <- 1.
# Max sigma.xy for this case is 2
```

1.3.1 Multivariate normal distributions in three dimensions

In three dimensions, the shape of a multivariate normal distribution is characterized a 3×3 symmetric positive-definite variance-covariance matrix, consisting of three variances (the diagonals on the matrix) and three pairwise covariances between the variables. For example, Figure 2 shows three views of an ellipse characterized by the following variance-covariance matrix:

$$\begin{bmatrix} 1 & 1.5 & 1 \\ 1.5 & 5 & 3 \\ 1 & 3 & 9 \end{bmatrix}$$

The code used to compute Figure 2 is given below:



(a) $x-y$ view (b) $x-z$ view (c) $y-z$ view

Figure 2: Three views of a 3d ellipsoid corresponding to equiprobable isoclines of a multivariate normal distribution.

```
> library(rgl)
> plot3d(ellipse3d(matrix(c(1,1.5,1,1.5,5,3,1,3,9),3,3)),
    xlim=lim, ylim=lim,zlim=lim,col=2)
> # spin the ellipsoid around with the mouse to get the correct angles
> rgl.snapshot("ellipse_3d_xy_view.png")
> rgl.snapshot("ellipse_3d_xz_view.png")
> rgl.snapshot("ellipse_3d_yz_view.png")
```

To calculate the pairwise correlations between the variables, we use the identity (see Lecture 4):

$$\rho_{XY} = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}} \tag{7}$$

giving us correlations $\rho_{XY} = 0.67, \rho_{XZ} = 0.33, \rho_{YZ} = 0.45$. Note that the "widest" 2D ellipse view is not the one that has the highest correlation coefficient. This is because the ratio of the variances in the different dimensions also plays an important role in determining the shape of the ellipse.

1.3.2 Multivariate normal distributions of higher dimension

Multivariate normal distributions of dimension n are characterized by an $n \times n$ variance-covariance matrix Σ similar to those used in Sections 1.3

and 1.3.2. If Σ is invertible then the multivariate normal probability density function is[1]

$$p(\mathbf{x};\mu,\Sigma) = \frac{1}{(\sqrt{2\pi})^{\frac{n}{2}}\left|\Sigma\right|^{-1}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)} \qquad (8)$$

In the one-dimensional case, $\Sigma = \sigma^2$ and $n = 1$; check for yourself that the multivariate normal reduceds to the univariate normal (Lecture 3, Section 4) in this situation.

1.4 Back to multilevel linear models

We'll start with a study of multilevel linear models. Substituting the linear predictor straight into Equation (3) gives us a model of the form

$$Y = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n + b_0 + b_1 Z_1 + \cdots + b_m Z_m + \ epsilon \qquad (9)$$

where $\vec{\mathbf{b}}$is multivariate-normal distributed around $\mathbf{0}$ with some covariance matrix Σ, and ϵ is normally distributed (independently of $\vec{\mathbf{b}}$) around 0 with some variance σ^2.

1.5 Multilevel linear models: a simple example

Let us consider the case where there are no predictor variables, just a fixed intercept and a random intercept. This type of model is easier to understand when rooted in a concrete example. Suppose that a phonetician conducts a longitudinal study of the pronunciation of the syllable “ba”, recruiting M native speakers of English and recording each speaker once a week for N weeks. In each case, the phonetician computes and records the F1 formant of the pronounced syllable. Now, no two recordings will be exactly alike, but different individuals will tend to pronounce the syllable in different ways—that is, there is both within-individual and between-individual variation in F1 formant from recording to recording. If we denote the F1 value for the jth recording of speaker i as y_{ij}, then our simple linear model looks as follows:

This gives us a model that looks like:

[1] $|\Sigma|$ denotes the determinant of Σ.

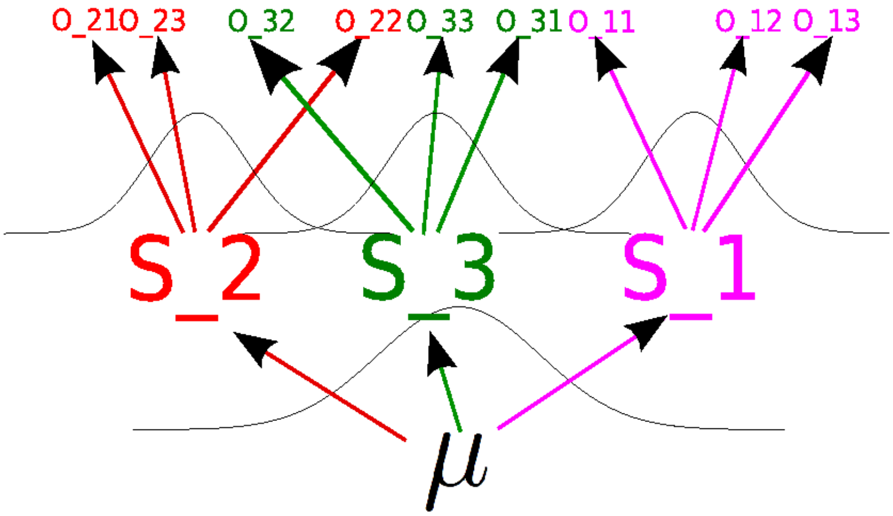


Figure 3: A multilevel linear model with two levels of normally-distributed noise around the mean. The `S_i` represent speaker-level means, and `O_ij` represent trial-level observations.

$$y_{ij} = \mu + b_i + \epsilon_{ij} \tag{10}$$

where the b_i are i.i.d. with variance σ_b^2, and the ϵ_{ij} are i.i.d. with variance σ_ϵ^2. Figure 3 gives a visualization of this two-level generative process. The individual speakers correspond to CLUSTERS of trial-level observations.

1.6 Fitting a multilevel model: theory

The multilevel model above is an example of a LATENT-VARIABLE model: we have postulated a set of speaker-specific mean formant frequencies at $\mu + b_1, \dots, \mu + b_M$, but we cannot observe those speaker-specific means. All we can do is to observe a number of noisy observations centered around these means. How we fit our multilevel model to a set of data depends on whether we are interested in recovering speaker-specific properties of trial clusters.

On the one hand, we may not be interested in estimating cluster-level model parameters b, but rather are primarily interested in the fixed effects β and the relative amount of variance associated with clusters as opposed to with individual observations—that is, estimating both $sigma^2$ and Σ. Circumstances under which this would be the case include if we are primarily interested in testing hypotheses regarding some subset of the fixed effects; or if we are interested in making predictions about data coming from new clusters that we have not yet observed. In this case, maximum-likelihood estimation is a reasonable strategy. However, because we are not interested in the b parameters we MARGINALIZE over them; hence the quantity we wish to maximize is the MARGINAL LIKELIHOOD of the fixed effects and variance terms:

$$\mathrm{Lik}(\beta, \sigma^2, \Sigma; \vec{x}) = \int_{-\infty}^{\infty} P(\vec{x}|b, \beta, \sigma^2, \Sigma) P(b|\sigma^2, \Sigma) \mathrm{d}b \tag{11}$$

On the other hand, we may be interested in estimating model parameters associated with specific clusters. This could be the case if, for example, our phonetician wanted to predict the next formant frequency that one of the study participants will produce. In this case we would be interested in simultaneously estimating all the effect parameters β and b, and also the covariances σ^2 and Σ. Unfortunately, this type of inference can be quite computationally intensive, and the necessary software is not as well developed. However, the `lme4` library has some facility for providing estimates of the b parameters as well.[2]

1.6.1 Fitting a multilevel linear model: practice

First we'll follow an example given in Baayen section 7.1. The question under investigation is whether there's support for an effect of trial on RT in a lexical decision experiment. We start using the `lexdec` dataset, pruning extremely high RTs and incorrect answers from the data, and then plot lowess fits of RT against trial number for each subject (Figure 4):

[2] We won't enter here into the details of how maximum-likelihood estimation is actually carried out for multilevel models: following it involves fairly sophisticated multivariate calculus. If you are interested, I suggest brushing up on your knowledge of matrix theory for linear regression (reading perhaps Healy, 2000 and then Rice, 1995, Section 14.3 before diving into Pinheiro and Bates, 2000, Chapter 2.

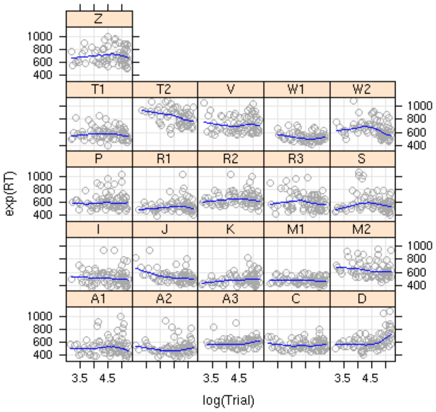


Figure 4: Result of `xylowess.fnc()` on reading times

```
> lexdec1 <- subset(lexdec,RT < 7 & Correct=="correct")
> xylowess.fnc(exp(RT) ~ Trial | Subject, lexdec1)
```

Now we fit a mixed-effects model. We start with random intercepts of word and subject:

```
> lexdec1.lmer <- lmer(exp(RT) ~ Trial + (1 | Subject) + (1 | Word),
    method="ML",data=lexdec1)
> lexdec1.lmer
Linear mixed-effects model fit by maximum likelihood
Formula: exp(RT) ~ Trial + (1 | Subject) + (1 | Word)
   Data: lexdec1
   AIC   BIC logLik MLdeviance REMLdeviance
 18889 18911  -9441      18881        18878
Random effects:
 Groups   Name        Variance Std.Dev.
 Word     (Intercept) 1836.7   42.857
 Subject  (Intercept) 7016.6   83.765
 Residual             9470.4   97.316
number of obs: 1557, groups: Word, 79; Subject, 21

Fixed effects:
             Estimate Std. Error t value
```

```
(Intercept) 613.17267   19.86768  30.863
Trial        -0.12174    0.05308  -2.294

Correlation of Fixed Effects:
      (Intr)
Trial -0.281
```

Hypothesis testing for fixed effects in mixed-effects models is still somewhat controversial, but the t-statistic of -2.294 gives fairly good support for an effect of trial.

Let us now draw our attention from the fixed effects, which we understand relatively well already, to the random effects. What is reported is not the specific subject- and word-specific intercepts—remember, these are being marginalized over—but rather the variances. Our resulting model, for the RT of subject i responding to word j in trial t, is as follows:

$$\begin{aligned}
RT_{ij} &= 613.17 - 0.12t + b_{s_i} + b_{w_j} + \epsilon_{ij} && (12)\\
b_{s_i} &\sim \mathcal{N}(0, 7017) && (13)\\
b_{w_j} &\sim \mathcal{N}(0, 1837) && (14)\\
b_{\epsilon_{ij}} &\sim \mathcal{N}(0, 9470) && (15)\\
& && (16)
\end{aligned}$$

Recall that variance of independent random variables (as are the b_{s_i}, b_{w_j}, and ϵ_{ij}, by assumption) is additive, so that we can literally say that the contribution of subject- and word-specific effects to the stochastic variance in per-trial RT is nearly as large (8854) as the residual variance (9470).

1.6.2 Hypothesis testing for significance of random effects

We can use likelihood-ratio tests for testing the significance of a random-effects component in the model:

```
> lexdec1.subjonly.lmer <- lmer(exp(RT) ~ Trial + (1 | Subject),
    method="ML",data=lexdec1)
> anova(lexdec1.subjonly.lmer,lexdec1.lmer)
Data: lexdec1
Models:
```

```
lexdec1.subjonly.lmer: exp(RT) ~ Trial + (1 | Subject)
lexdec1.lmer: exp(RT) ~ Trial + (1 | Subject) + (1 | Word)
                      Df     AIC     BIC  logLik  Chisq Chi Df Pr(>Chisq)
lexdec1.subjonly.lmer  3 19021.0 19037.0 -9507.5
lexdec1.lmer           4 18889.3 18910.7 -9440.7 133.63      1  < 2.2e-16 ***
```

Not surprisingly, there is exceedingly strong support for random word-specific intercepts—after all, no word-specific properties have been included in the model, and the random intercept is the only way for the model to include an explanatory effect of this sort. Perhaps equally unsurprisingly, there is also substantial evidence for subject-specific intercepts:

```
> lexdec1.wordonly.lmer <- lmer(exp(RT) ~ Trial + (1 | Word),
    method="ML",data=lexdec1)
> anova(lexdec1.wordonly.lmer,lexdec1.lmer)
Data: lexdec1
Models:
lexdec1.wordonly.lmer: exp(RT) ~ Trial + (1 | Word)
lexdec1.lmer: exp(RT) ~ Trial + (1 | Subject) + (1 | Word)
                      Df     AIC     BIC  logLik  Chisq Chi Df Pr(>Chisq)
lexdec1.wordonly.lmer  3 19585.3 19601.4 -9789.6
lexdec1.lmer           4 18889.3 18910.7 -9440.7 697.97      1  < 2.2e-16 ***
```

This

More interestingly, we can test whether different subjects exhibit different dependencies on trial. We construct a new model with a random slope for trial:

```
> lexdec1.trialrandom.lmer <- lmer(exp(RT) ~ Trial + (1 + Trial | Subject)
    + (1 | Word) ,method="ML",data=lexdec1)
> lexdec1.trialrandom.lmer
Linear mixed-effects model fit by maximum likelihood
Formula: exp(RT) ~ Trial + (1 + Trial | Subject) + (1 | Word)
   Data: lexdec1
   AIC   BIC logLik MLdeviance REMLdeviance
 18856 18889  -9422      18844        18839
Random effects:
 Groups   Name        Variance   Std.Dev.  Corr
 Word     (Intercept) 1.8732e+03  43.28024
 Subject  (Intercept) 1.2253e+04 110.69270
          Trial       1.8858e-01   0.43425 -0.716
 Residual             9.0542e+03  95.15345
number of obs: 1557, groups: Word, 79; Subject, 21

Fixed effects:
            Estimate Std. Error t value
(Intercept) 615.0985    25.3626  24.252
Trial        -0.1380     0.1082  -1.275

Correlation of Fixed Effects:
      (Intr)
Trial -0.702
```

Note that the t-statistic for the fixed effect of Trial has dropped to the low value of -1.3. In this model there is no support for a fixed effect of trial. However, the subject-specific random slope of Trial has a non-trivial variance, and importantly is strongly negatively correlated with the subject-specific random intercept. First we use a likleihood-ratio test to check whether this is a significant improvement over the model with only a fixed effect of trial:

```
> anova(lexdec1.lmer,lexdec1.trialrandom.lmer)
Data: lexdec1
Models:
lexdec1.lmer: exp(RT) ~ Trial + (1 | Subject) + (1 | Word)
lexdec1.trialrandom.lmer: exp(RT) ~ Trial + (1 + Trial | Subject) + (1 | Word)
                         Df     AIC     BIC  logLik  Chisq Chi Df Pr(>Chisq)
lexdec1.lmer              4 18889.3 18910.7 -9440.7
lexdec1.trialrandom.lmer  6 18856.4 18888.5 -9422.2 36.913      2  9.646e-09 ***
```

Adding the random slope of Trial leads to a considerable improvement in the model. Now let us go back and interpret the negative correlation between the subject-specific intercepts and Trial slopes. The negative correlation means that when a subject reads slowly overall, they tend to speed up more over the course of the experiment. If we go back to Figure 4 we can see that this is a reasonable inference: the slowest readers are T2 and V; T2 shows the strongest evidence of speeding up throughout the experiment. J also shows

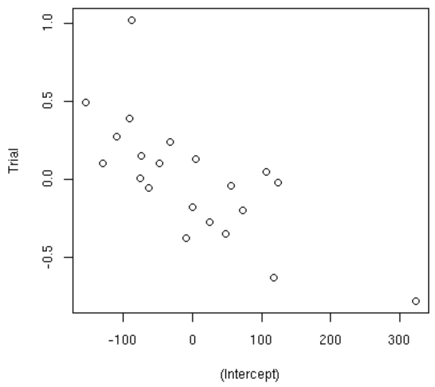


Figure 5: Random slopes plotted against random intercepts (BLUPs)

strong evidence of a speedup; and their reading overall is not the fastest either.

We can look more directly at the model's reconstruction of this by considering the Best Linear Unbiased Predictors (BLUPs) of the random effect, defined as the modes of the random effects $\vec{\mathbf{b}}$ conditioned on the estimates of the fixed effects (Figure 5):

$$\hat{\mathbf{b}} \equiv \arg\max_{\mathbf{b}} \mathrm{Lik}(\mathbf{b}|\vec{x}, \hat{\beta})$$

```
> plot(ranef(lexdec1.trialrandom.lmer)[[2]])
```

References

Healy, M. (2000). *Matrices for Statistics.* Oxford, second edition.

Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-Effects Models in S and S-PLUS.* Springer.

Rice, J. A. (1995). *Mathematical Statistics and Data Analysis.* Duxbury Press, 2 edition.

Lecture 15: mixed-effects logistic regression

28 November 2007

In this lecture we'll learn about mixed-effects modeling for logistic regression.

1 Technical recap

We moved from generalized linear models (GLMs) to multi-level GLMs by adding a stochastic component to the linear predictor:

$$\eta = \alpha + \beta_1 X_1 + \cdots + \beta_n X_n + b_0 + b_1 Z_1 + \cdots + b_m Z_m \tag{1}$$

and usually we assume the random effects vector $\vec{\mathbf{b}}$ is normally distributed with mean 0 and variance-covariance matrix Σ.

In a mixed-effects logistic regression model, we simply embed the stochastic linear predictor in the binomial error function (recall that in this case, the predicted mean μ corresponds to the binomial parameter p):

$$\begin{aligned} P(y;\mu) &= \binom{n}{yn} \mu^{yn} (1-\mu)^{(1-y)n} && \text{(Binomial error distribution)} \quad (2) \\ \log \frac{\mu}{1-\mu} &= \eta && \text{(Logit link)} \quad (3) \\ \mu &= \frac{e^\eta}{1+e^\eta} && \text{(Inverse logit function)} \quad (4) \end{aligned}$$

1.1 Fitting multi-level logit models

As with linear mixed models, the likelihood function for a multi-level logit model must marginalize over the random effects $\vec{\mathbf{b}}$:

$$\mathrm{Lik}(\beta, \Sigma|\vec{x}) = \int_{-\infty}^{\infty} P(\vec{x}|\beta, b)P(b|\Sigma)\mathrm{d}b \tag{5}$$

Unfortunately, this likelihood cannot be evaluated exactly and thus the maximum-likelihood solution must be approximated. You can read about some of the approximation methods in Bates (2007, Section 9). Laplacian approximation to ML estimation is available in the `lme4` package and is recommended. Penalized quasi-likelihood is also available but not recommended, and adaptive Gaussian quadrature is recommended but not yet available.

1.2 An example

We return to the `dative` dataset and (roughly) follow the example in Baayen Section 7.4. We will construct a model with all the available predictors (except for speaker), and with verb as a random effect. First, however, we need to determine the appropriate scale at which to enter the length (in number of words) of the recipient and theme arguments. Intuitively, both raw scales and log scales are plausible. If our response were continuous, a natural thing to do would be to look at scatterplots of each of these variables against the response. With a binary response, however, such a scatterplot is not very informative. Instead, we take two approaches:

1. Look at the empirical relationship between argument length and mean response, using a SHINGLE;

2. Compare single-variable logistic regressions of response against raw/log argument length and see which version has a better log-likelihood.

First we will define convenience functions to use for the first approach:

```
> tapply.shingle <- function(x,s,fn,...) {
  result <- c()
  for(l in levels(s)) {
    x1 <- x[s > l[1] & s < l[2]]
    result <- c(result, fn(x1,...))
  }
  result
}
> logit <- function(x) {
    log(x/(1-x))
  }
```

We then plot the mean response based on shingles (Figure 1):

```
> my.intervals <- cbind(1:29-0.5,1:29+1.5)
> response <- ifelse(dative$RealizationOfRecipient=="PP",1,0)
> recipient.x <- with(dative,tapply.shingle(LengthOfRecipient,
    shingle(LengthOfRecipient,my.intervals),mean))
> recipient.y <- with(dative,tapply.shingle(response,
    shingle(LengthOfRecipient,my.intervals),mean))
> plot(recipient.x,logit(recipient.y))
> theme.y <- with(dative,tapply.shingle(response,
    shingle(LengthOfTheme,my.intervals),mean))
> theme.x <- with(dative,tapply.shingle(LengthOfTheme,
    shingle(LengthOfTheme,my.intervals),mean))
> plot(theme.x,logit(theme.y))
```

These plots are somewhat ambiguous and could support either a linear or logarithmic relationship in logit space. (Keep in mind that (a) we're not seeing points where there are 100% of responses that are "successful" or "failures"; and (b) there are very few data points at the larger lengths.) So we resort to the logistic regression approach (recall that the deviance is simply -2 times the log-likelihood):

```
> summary(glm(response ~ LengthOfTheme,dative,
    family="binomial"))$deviance
[1] 3583.41
> summary(glm(response ~ log(LengthOfTheme),dative,
    family="binomial"))$deviance
[1] 3537.279
> summary(glm(response ~ LengthOfRecipient,dative,
```

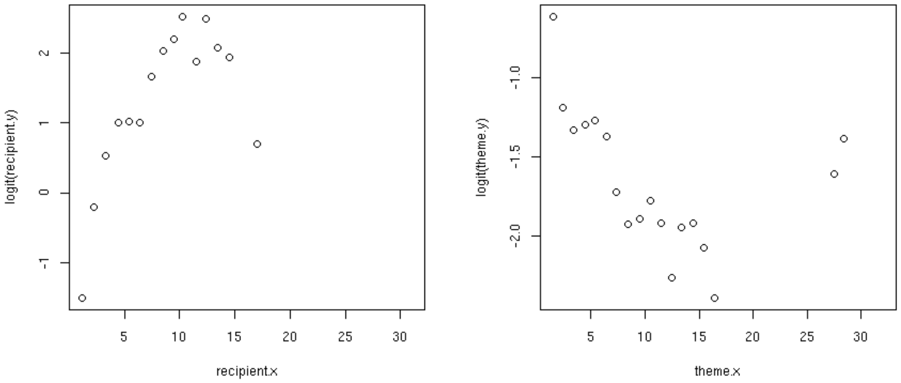


Figure 1: Responses of recipient and theme based on shingles

```
    family="binomial"))$deviance
[1] 3104.92
> summary(glm(response ~ log(LengthOfRecipient),dative,
    family="binomial"))$deviance
[1] 2979.884
```

In both cases the log-length regression has a lower deviance and hence a higher log-likelihood. So we'll enter these terms into the overall mixed-effects regression as log-lengths.

```
> dative.glmm <- lmer(RealizationOfRecipient ~
    log(LengthOfRecipient) + log(LengthOfTheme) +
    AnimacyOfRec + AnimacyOfTheme +
    AccessOfRec + AccessOfTheme +
    PronomOfRec + PronomOfTheme +
    DefinOfRec + DefinOfTheme +
    SemanticClass +
    Modality + (1 | Verb), dative,family="binomial",method="Laplace")
> dative.glmm

[...]

Random effects:
 Groups Name        Variance Std.Dev.
 Verb   (Intercept) 4.6872   2.165
number of obs: 3263, groups: Verb, 75

Estimated scale (compare to  1 )  0.7931773

Fixed effects:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)               1.9463     0.6899   2.821 0.004787 **
AccessOfThemegiven        1.6266     0.2764   5.886 3.97e-09 ***
AccessOfThemenew         -0.3957     0.1950  -2.029 0.042451 *
AccessOfRecgiven         -1.2402     0.2264  -5.479 4.28e-08 ***
AccessOfRecnew            0.2753     0.2472   1.113 0.265528
log(LengthOfRecipient)    1.2891     0.1552   8.306  < 2e-16 ***
log(LengthOfTheme)       -1.1425     0.1100 -10.390  < 2e-16 ***
AnimacyOfRecinanimate     2.1889     0.2695   8.123 4.53e-16 ***
AnimacyOfThemeinanimate  -0.8875     0.4991  -1.778 0.075334 .
PronomOfRecpronominal    -1.5576     0.2491  -6.253 4.02e-10 ***
PronomOfThemepronominal   2.1450     0.2654   8.081 6.40e-16 ***
DefinOfRecindefinite      0.7890     0.2087   3.780 0.000157 ***
DefinOfThemeindefinite   -1.0703     0.1990  -5.379 7.49e-08 ***
SemanticClassc            0.4001     0.3744   1.069 0.285294
SemanticClassf            0.1435     0.6152   0.233 0.815584
SemanticClassp           -4.1015     1.5371  -2.668 0.007624 **
SemanticClasst            0.2526     0.2137   1.182 0.237151
Modalitywritten           0.1307     0.2096   0.623 0.533008
```

(Incidentally, this model has higher log-likelihood than the same model with raw instead of log- argument length, supporting our choice of log-length as the preferred predictor.)

The fixed-effect coefficients can be interpreted as normal in a logistic regression. It is important to note that there is considerable variance in the random effect of verb. The scale of the random effect is that of the linear predictor, and if we consult the logistic curve we can see that a standard deviation of 2.165 means that it would be quite typical for the magnitude of this random effect to be the difference between a PO response probability of 0.1 and 0.5.

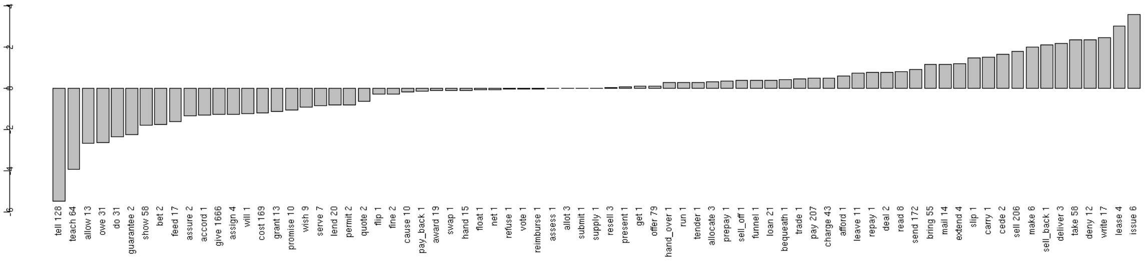


Figure 2: Random intercept for each verb in analysis of the `dative` dataset

Because of this considerable variance of the effect of verb, it is worth looking at the BLUPs for the random verb intercept:

```
> nms <- rownames(ranef(dative.glmm)$Verb)
> intercepts <- ranef(dative.glmm)$Verb[,1]
> support <- tapply(dative$Verb,dative$Verb,length)
> labels <- paste(nms,support)
> barplot(intercepts[order(intercepts)],names.arg=labels[order(intercepts)],
    las=3,mgp=c(3,-0.5,0),ylim=c(-6,4)) # mgp fix to give room for verb names
```

The results are shown in Figure 2. On the labels axis, each verb is followed by its SUPPORT: the number of instances in which it appears in the `dative` dataset. Verbs with larger support will have more reliable random-intercept BLUPs. From the barplot we can see that verbs including *tell*, *teach*, and *show* are strongly biased toward the double-object construction, whereas *send*, *bring*, *sell*, and *take* are strongly biased toward the prepositional-object construction.

This result is theoretically interesting because the dative alternation has been at the crux of a multifaceted debate that includes:

- whether the alternation is meaning-invariant;
- if it is not meaning-invariant, whether the alternants are best handled via constructional or lexicalist models;
- whether verb-specific preferences observable in terms of raw frequency truly have their locus at the verb, or can be explained away by other properties of the individual clauses at issue.

Because verb-specific preferences in this model play such a strong role despite the fact that many other factors are controlled for, we are on better footing to reject the alternative raised by the third bullet above that verb-specific preferences can be entirely explained away by other properties of the individual clauses. Of course, it is always possible that there are other explanatory factors correlated with verb identity that will completely explain away verb-specific preferences; but this is the nature of science. (This is also a situation where controlled, designed experiments can play an important role by eliminating the correlations between predictors.)

1.3 Model comparison & hypothesis testing

For nested mixed-effects logit models differing only in fixed-effects structure, likelihood-ratio tests can be used for model comparison. Likelihood-ratio tests are especially useful for assessing the significance of predictors consisting of factors with more than two levels, because such a predictor simultaneously introduces more than one parameter in the model:

```
> dative.glmm.noacc <- lmer(RealizationOfRecipient ~
    log(LengthOfRecipient) + log(LengthOfTheme) +
    AnimacyOfRec + AnimacyOfTheme +
    PronomOfRec + PronomOfTheme +
    DefinOfRec + DefinOfTheme +
    SemanticClass +
    Modality + (1 | Verb), dative,family="binomial",method="Laplace")
> anova(dative.glmm,dative.glmm.noaccessibility)
[...]
                  Df     AIC     BIC  logLik  Chisq Chi Df Pr(>Chisq)
dative.glmm.noacc 15 1543.96 1635.31 -756.98
dative.glmm       19 1470.93 1586.65 -716.46 81.027      4  < 2.2e-16 ***
> dative.glmm.nosem <- lmer(RealizationOfRecipient ~
    log(LengthOfRecipient) + log(LengthOfTheme) +
    AnimacyOfRec + AnimacyOfTheme +
    AccessOfRec + AccessOfTheme +
    PronomOfRec + PronomOfTheme +
    DefinOfRec + DefinOfTheme +
    Modality + (1 | Verb), dative,family="binomial",method="Laplace")
> anova(dative.glmm,dative.glmm.nosem)
```

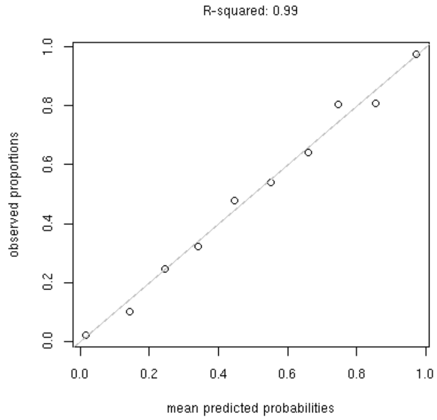


Figure 3: The fit between predicted and observed probabilities for each decile of predicted probability for `dative.glmm`

```
[...]
                   Df     AIC     BIC  logLik  Chisq Chi Df Pr(>Chisq)
dative.glmm.nosem 15 1474.55 1565.90 -722.27
dative.glmm       19 1470.93 1586.65 -716.46 11.618      4    0.02043 *
```

1.4 Assessing a logit model

When assessing the fit of a model whose response is continuous, a plot of the residuals is always useful. This is not a sensible strategy for assessing the fit of a model whose response is categorical. Something that is often done instead is to plot *predicted probability* against *observed proportion* for some binning of the data. This process is described in Baayen page 305, through the `languageR` function `plot.logistic.fit.fnc()`:

```
> plot.logistic.fit.fnc(dative.glmm,dative)
```

This is really a very good fit.

Finally, a slight word of warning: our model assumed that the random verb-specific intercepts are normally distributed. As a sanity check, we can use the Shapiro-Wilk test to check the distribution of BLUPs for the intercepts:

```
> shapiro.test(ranef(dative.glmm)$Verb[,1])

Shapiro-Wilk normality test

data:  intercepts
W = 0.9584, p-value = 0.0148
```

There is some evidence here that the intercepts are not normally distributed. This is more alarming given that the model has *assumed* that the intercepts are normally distributed, so that it is biased toward assigning BLUPs that adhere to a normal distribution.

2 Further Reading

There is good theoretical coverage (and some examples) of GLMMs in Agresti (2002, Chapter 12). There is a bit of `R`-specific coverage in Venables and Ripley (2002, Section 10.4) which is useful to read as a set of applie examples, but the code they present uses penalized quasi-likelihood estimation and this is outdated by `lme4`.

References

Agresti, A. (2002). *Categorical Data Analysis*. Wiley, second edition.

Bates, D. (2007). Linear mixed model implementation in lme4. Manuscript, University of Wisconsin, 15 May 2007.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, fourth edition.

Lecture 16: polynomial regression, splines, and kernel regression

3 December 2007

In this lecture we'll learn about improving regression modeling for continuous predictor variables with three available techniques:

- using polynomial functions of the predictor;
- using splines;
- using nonparametric/kernel methods.

1 Motivation

In the example data analysis of the dative alternation of Lecture 14, we had several categorical predictors for the categorical response variable of whether the double-object (DO) or prepositional-object (PO) alternative was used in a dative construction. In addition, we had two continuous response variables: the lengths (in words) of the theme and recipient arguments. In that lecture, we compared the options of modeling the effect of length on the linear predictor on two scales: raw length and log-length. As you can imagine, however, there is an infinite space of possible transformations that could be applied to a continuous predictor such as length (or, for example, frequency or familiarity) before it is entered into a regression. This lecture covers some of the options.

2 Nonlinear terms

The simplest way to create a nonlinear relationship between the predictor variable X and the linear predictor is to create extra predictor variables that are transforms of X. For example, if we add a predictor that is the square of X we get the model

$$\eta = \alpha + \beta_1 X + \beta_2 X^2$$

As an example, let us construct a linear regression on length of recipient for the `dative` dataset and compare it with observed sample proportions estimated from a shingle (Figure 1):

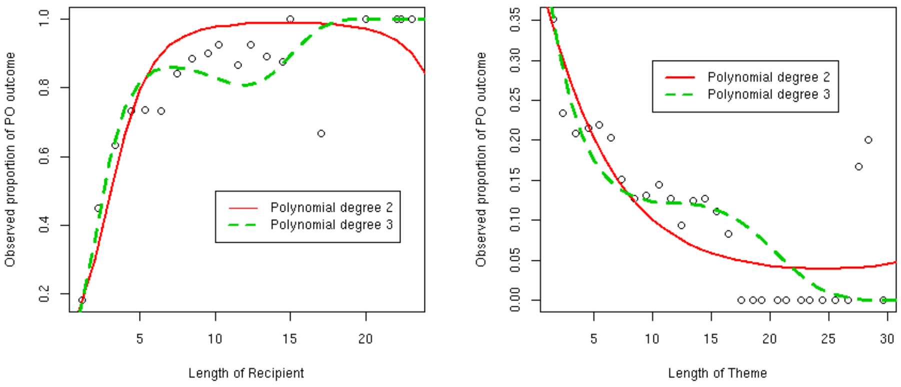


Figure 1: Quadratic polynomial regression for lengths of recipient and theme in `dative` dataset

We get the generalization that the effect of length flattens out with increasing length (consistent with the idea that log-space is the natural relation between length and linear predictor). However, extrapolation using polynomial regression is dangerous. This is particularly evident for the polynomial of degree 2, for which we get the wrong generalization—we would extrapolate that as the argument gets longer than the maximum observed length, its effect *reverses*.

```
my.intervals <- cbind(1:29-0.5,1:29+1.5)
response <- ifelse(dative$RealizationOfRecipient=="PP",1,0)
```

```
recipient.x <- with(dative,tapply.shingle(LengthOfRecipient,
  shingle(LengthOfRecipient,my.intervals),mean))
recipient.y <- with(dative,tapply.shingle(response,
  shingle(LengthOfRecipient,my.intervals),mean))
plot(recipient.x, recipient.y,xlab="Length of Recipient",
  ylab="Observed proportion of PO outcome")
dummy.data.frame <- data.frame(LengthOfRecipient=1:45)
dative.rec.glm.p2 <- glm(response ~ pol(LengthOfRecipient,2),dative,
  family="binomial")
lines(dummy.data.frame$LengthOfRecipient,invlogit(predict(dative.rec.glm.p2,
  dummy.data.frame)),col=2,lwd=2)
dative.rec.glm.p3 <- glm(response ~ pol(LengthOfRecipient,3),dative,
  family="binomial")
lines(dummy.data.frame$LengthOfRecipient,invlogit(predict(dative.rec.glm.p3,
  dummy.data.frame)),col=3,lwd=3,lty=2)
legend(10,0.5,c("Polynomial degree 2", "Polynomial degree 3"), lty=c(1,2),lwd=c(1,
# now repeat with LengthOfTheme rather than LengthOfRecipient
```

3 Restricted Cubic Splines

As mentioned in the previous section, polynomial regression is useful but potentially dangerous. For example, even a cubic regression creates strange predictions about highly frequent words (Figure 2, left):

```
> plot(exp(RTlexdec) ~ WrittenFrequency,english,pch=".")
> english.p3 <- lm(exp(RTlexdec) ~ pol(WrittenFrequency,3),english)
> x <- seq(0,12,by=0.1)
> dummy <- data.frame(WrittenFrequency = x)
> lines(x,predict(english.p3,dummy),lwd=2,col=2) # odd upturn at the end!
```

What is really happening in this case is that there is relatively little data at the right periphery; the flexibility granted by the cubic regression is being used to nudge the regression to an optimal shape in the large cloud. We can confirm this by comparing the results when we truncate the dataset at words of log written frequency above 8 for purposes of fitting the regression:

```
> english.p3.1 <- lm(exp(RTlexdec) ~ pol(WrittenFrequency,3),
    subset(english,WrittenFrequency < 8))
> lines(x,predict(english.p3.1,dummy),lwd=2,col=3,lty=2) # odd upturn even worse!
```

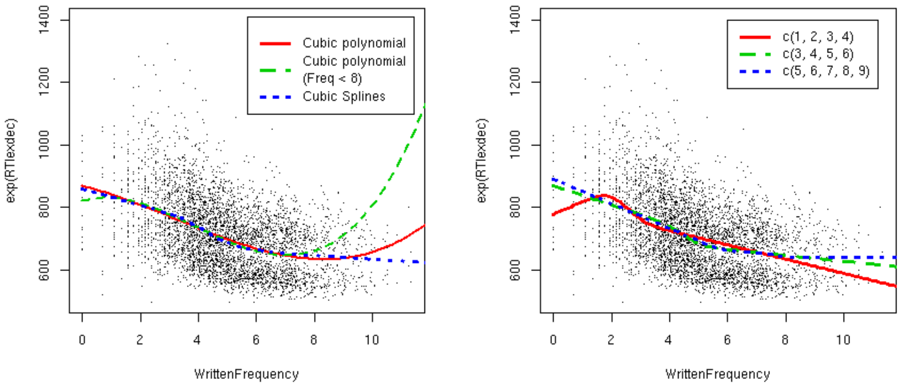


Figure 2: Modeling effects of written frequency on lexical-decision reaction time using polynomial regression and splines

The odd upturn got *much worse*, not better.

This is a difficult problem to deal with—extrapolation beyond what you have seen is never easy—but one approach that has become quite popular in the statistical literature is the use of RESTRICTED (also called NATURAL) CUBIC SPLINES. These are characterized as follows:

1. Place a set n of points—called KNOTS—on the real line: one at each of the maximum and minimum observed values of your predictor X, and the rest somewhere in between. Call these knots $\tau_1, \dots, \tau_n$.

2. The contribution of X to the linear predictor is determined as follows:

   (a) The i-th interval (the one lying between the knots τ_i and τ_{i+}) is characterized by a cubic curve, such that the contribution of X is determined by

   $$\beta_{i0} + \beta_{i1}(X - \tau_i) + \beta_{i2}(X - \tau_i)^2 + \beta_{i3}(X - \tau_i)^3$$

   for some choices of β_{ij};

   (b) Beyond the observed data (i.e., for $X < \tau_1$ and $X > \tau_n$), X contributes linearly to the linear predictor (i.e., as $\alpha + \beta(X - \tau_k)$);

(c) The spline parameters must be chosen such that the spline is smooth everywhere (technically, such that the spline and its first and second derivatives are continuous).

rcs()

In R we can use the rcs() function to introduce restricted cubic splines in our regression. [1] A simple example is given below:

```
> english.rcs <- ols(exp(RTlexdec) ~ rcs(WrittenFrequency),english)
> lines(x,predict(english.rcs,dummy),lwd=3,lty=3,col=4)
> legend(6,1300,c("Cubic polynomial","Cubic polynomial\n(Freq < 8)",
    "Cubic Splines"), lwd=3,lty=c(1,2,3),col=c(2,3,4))
```

Notice how the spline results have the desirable properties of the cubic polynomial regression (they give a more nuanced picture in the large cloud), but don't have the undesirable upturn at the right edge of the graph.

The rcs() function chooses knot positions for you. You can also choose your own (Figure 2, right):

```
> rcslines <- function(knots,i) {
  model <- ols(exp(RTlexdec) ~ rcs(WrittenFrequency,knots),english)
  lines(x,predict(model,dummy),lwd=3,lty=i,col=i+1)
}
> plot(exp(RTlexdec) ~ WrittenFrequency,english,pch=".",ylim=c(500,1400))
> knotsets <- list(c(1,2,3,4),c(3,4,5,6),c(5,6,7,8,9))
> rcslines(knotsets[[1]],1)
> rcslines(knotsets[[2]],2)
> rcslines(knotsets[[3]],3)
> legend(6,1400,as.character(knotsets),lwd=3,lty=1:3,col=2:4)
```

4 Nonparametric/locally weighted regression

The third option that we will consider here is the use of nonparametric or locally weighted methods to estimate the relationship between a predictor and the response.[2] We will look at one of the most popular such methods: loess locally weighted regression.

[1] **Warning:** the lm() function does not interact properly with rcs()—use the ols() function (ordinary least squares) from the Design package instead. You can also use the ns() function from the splines library for restricted cubic splines.

[2] The term "nonparametric" is heavily overloaded and means different things in different contexts. One of the common threads in its meaning is the idea that the complexity of the

4.1 Locally weighted regression with `loess()`

The LOESS regression model (Cleveland, 1979) is an instance of LOCALLY WEIGHTED REGRESSION: a linear regression model of standard parametric form is assumed, but each point in your dataset contributes differentially to the regression depending on its distance from the point at which you want to make a prediction. This idea is unpacked a bit more in the following description.

A loess model is characterized by (1) a classic linear model M, together with (2) a SPAN α, representing the proportion of points that have some contribution to the regression. Suppose you want to make a prediction at point x. Choose a proportion of your data α in a window around x. Find the most distant point x_{max} within this window. In order to predict the value of your response at x, use ordinary maximum-likelihood/least-squares regression to fit M on the data within your window, but each point x_i is weighted by the quantity

$$\left(1-\left(\frac{|x_i-x|}{|x_{max}-x|}\right)^3\right)^3$$

That is, points near the maximum distance will have almost no contribution to the regression. You can tune the severity of "smoothing" in the model by manipulating the span α.

In R, loess regression is implemented by the `loess()` function. Here's an example of loess regression on the RT/frequency relationship for lexical decision (Figure **??**):

`ols()`

```
> t.loess1 <- loess(exp(RTlexdec) ~ WrittenFrequency, english)
    # default span is 0.75
> plot(t.loess1,pch=".") # just plots the points
> x <- seq(0,12,by=0.01)
> lines(x,predict(t.loess1,x),lwd=3,col=2)
> t.loess2 <- loess(exp(RTlexdec) ~ WrittenFrequency, english, span = 0.4)
> lines(x,predict(t.loess2,x),lwd=3,col=3,lty=2)
> t.loess3 <- loess(exp(RTlexdec) ~ WrittenFrequency, english, span = 0.1)
```

structure of the model grows as the number of data points grows. This is in opposition to the "parametric" models we have been looking at up until now, in which the complexity of the model is bounded by the number of parameters specified for the model.

```
> lines(x,predict(t.loess3,x),lwd=3,col=4,lty=3)
> legend(5,1300,c("span=0.75","span=0.4","span=0.1"),lwd=3,col=2:4,lty=1:3)
```

5 Further Reading

Treatments of spline regression and related techniques can be found in many places, including Green and Silverman (1994), Venables and Ripley (2002, Section 8.7), Harrell (2001, Section 2.4), Hastie et al. (2001, Chapter 5), and Maindonald and Braun (2007, Sections 7.5 and 7.6).

References

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836.

Green, P. J. and Silverman, B. W. (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. London: Chapman & Hall.

Harrell, Jr, F. E. (2001). *Regression Modeling Strategies*. Springer.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.

Maindonald, J. and Braun, J. (2007). *Data Analysis and Graphics using R*. Cambridge, second edition.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, fourth edition.