3 Phonetics

Phoneticians have a long tradition of quantifying their observations of pronunciation and hearing. In this history both hypothesis testing and regression techniques have been extensively used. Obviously the full range of methods used in the quantitative analysis of phonetic data cannot be covered in a short chapter on the topic, however I do hope to extend the discussion of t-test and regression in interesting ways here, and to introduce factor analysis.

3.1 Comparing mean values

We saw in chapter 2 that we can test the hypothesis that a sample mean value $\bar{x}$ is the same as or different from a particular hypothesized population mean μ. Sometimes, this is a test that we are interested in. For example, we might want to know if the observed, sample mean $\bar{x}$ is reliably different from zero, but, in many cases the $\bar{x}$-μ comparison is not really what we want because we are comparing two sample means.

For example, the key question of interest with the Cherokee 1971/2001 data is the comparison of two sample means. Is the mean VOT in 1971 different from the mean VOT in 2001, as the boxplot in Figure 3.1 suggests ? This comparison will be the focal example of this section.

3.1.1 Cherokee Voice Onset time: $\mu_{1971} = \mu_{2001}$

We want to test whether the average VOT in 1971 was equal to the average VOT in 2001, because we think that for this speaker there may have been a slow drift in the aspiration of voiceless stops as a result of language contact. This question provides us with the null hypothesis that there was no reliable difference in the true, population, means for these two years - that is: H_0: $\mu_{1971} = \mu_{2001}$.

We can test this hypothesis with a *t* test similar to the "one sample" *t* test that was discussed in chapter 2. In that discussion, just to review, we tested the null hypothesis: H_0: $\mu_{1971} = \mu_{hyp}$, where we supplied the hypothesized population mean. Recall that the idea with the *t* test is that we expect the difference between means to be zero - the null hypothesis is that there is no difference - and we measure the magnitude of the observed difference relative to the magnitude of random or chance variation we expect in mean values (the standard error of the mean). If the difference between means is large, more than about 2 standard errors (a *t* value of 2 or -2), we are likely to conclude that the sample mean comes from a population that has a different mean than the hypothesized population mean.

$$t = \frac{\bar{x} - \mu}{SE}$$

the *t* statistic is the difference between observed and expected mean, divided by standard error of the observed mean.

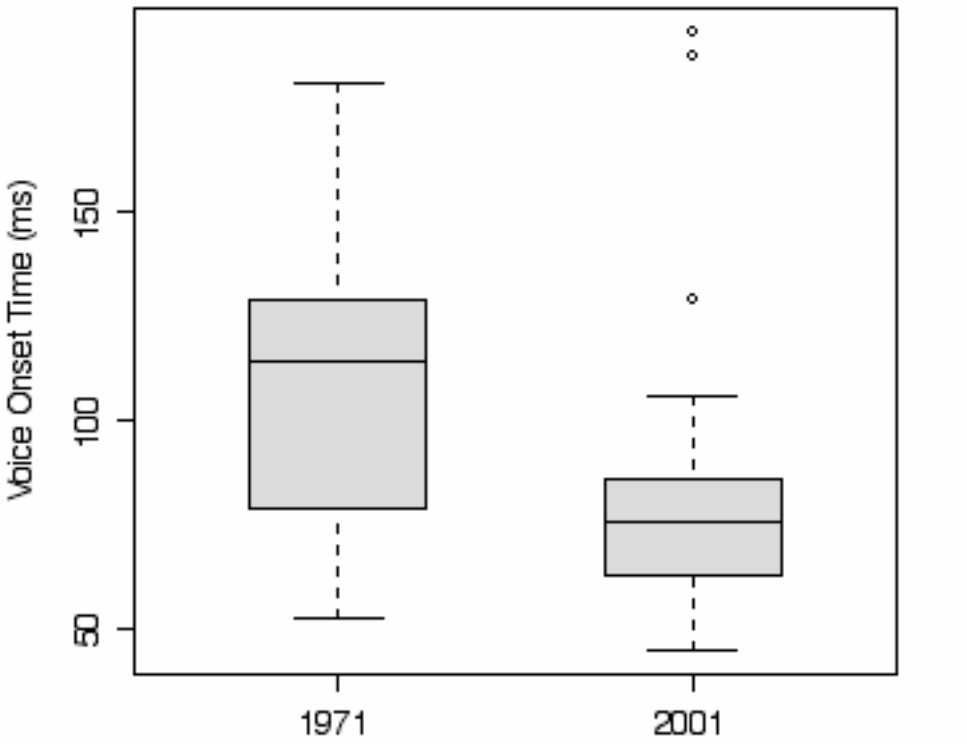

Figure 3.1. A boxplot of voice onset time measurements for Cherokee voiceless stops /t/ and /k/ produced by speaker DF in 1971 and in 2001.

So in testing whether the mean VOT in 1971 is different from the mean VOT in 2001 for this talker we are combining two null hypotheses:

H0: $\mu_{1971} = \mu$
H0: $\mu_{2001} = \mu$
H0: $\mu_{1971} = \mu_{2001}$

The expected man value of the 1971 sample is the same as the expected value of the 2001 sample,

so just as with a one-sample t-test the expected value of the difference is 0. Therefore we can compute a t statistic from the difference between the means of our two samples.

$$t = \frac{\bar{x}_{1971} - \bar{x}_{2001}}{SE}$$ the two-sample *t* value.

There is one little complication when we compare the two means in this computation of *t*. We have two samples of data, one from 1971 and one from 2001 and therefore we have two estimates of the standard error of the mean (SE). So, in calculating the *t* statistic we need to take information from both the 1971 data set and the 2001 data set when we compute the SE for this test.

R note. Before running t tests on the Cherokee data we need to read the data from a text file. The Cherokee data are in "cherokeeVOT.txt" and are arranged in three columns (I made this text file in a spreadsheet program using the "save as" command to save the data as "tab delimited text". The first column has the VOT measurement in milliseconds, the second column indicates that this VOT measurement is from a 1971 recording or a 2001 recording, and the third column indicates that the measurement comes from either a /k/ or a /t/. Use `read.delim()` to read the data file. I usually type the name of the data frame to see the column names and data values to be sure that the file was read correctly - here I show only the first five lines of the print out.

```
> vot <- read.delim("cherokeeVOT.txt")
> vot
   VOT year Consonant
1   67 1971         k
2  127 1971         k
3   79 1971         k
4  150 1971         k
5   53 1971         k
```

Everything is apparently fine, but when we look at a `summary()` to get some descriptive statistics of the variables we see a problem. This R function is convenient when you have a large data file and you want to quickly verify that your data have been read correctly.

```
summary(vot)
      VOT             year      Consonant
 Min.   : 45.00   Min.   :1971   k:21
 1st Qu.: 71.50   1st Qu.:1971   t:23
 Median : 81.50   Median :2001
 Mean   : 96.45   Mean   :1989
```

```
 3rd Qu.:120.25   3rd Qu.:2001
 Max.   :193.00   Max.   :2001
```

The data `summary()` indicates a problem because I intended "year" to be a nominal variable that has only two levels "2001" and "1971". R, quite naturally, saw that the column was filled with numbers and so treated the variable as continuous (either a ratio or interval variable). This needs to be changed explicitly by instructing R to treat the "year" column as a `factor()`. After entering the `factor()` command "year" shows up as a nominal variable with 18 data points from 1971 and 26 data points from 2001.

```
> vot$year <- factor(vot$year)
> summary(vot)
      VOT           year    Consonant
 Min.   : 45.00   1971:18   k:21
 1st Qu.: 71.50   2001:26   t:23
 Median : 81.50
 Mean   : 96.45
 3rd Qu.:120.25
 Max.   :193.00
```

If you wish to add a new column to the data.frame that has the factorized "year", rather than changing "year" in place, you can type the command as: `>vot$fyear <- factor(vot$year)`.

Use `attach(vot)` to instruct future lines to look in the "vot" spreadsheet/ data.frame for the variables that we mention (this is so we can refer to "year" instead of "vot$year". Now we can use `mean()` and `sd()` to get the means and standard deviations for the 1971 and 2001 data.

```
> mean(VOT[year=="1971"])
[1] 113.5
> mean(VOT[year=="2001"])
[1] 84.65385
> sd(VOT[year=="1971"])
[1] 35.92844
> sd(VOT[year=="2001"])
[1] 36.08761
```

The notation `VOT[year=="1971"]` specifies a subset of the VOT values for which the variable `year` is equal to "1971".

Oh, by the way. I made figure 3.1 with the `boxplot()` command. A boxplot is a very good way to see the central tendency and the range of variability of the data.

```
> boxplot(VOT~year,data=vot,col="lightgrey",ylab = "Voice Onset Time (ms)")
----------------------------------------
```

3.1.2 Samples have equal variance.

So we are testing the null hypothesis that one Cherokee speaker's average VOT was the same in 1971 and 2001. That is: H_0: $\mu_{1971} = \mu_{2001}$. We have two samples of data, one from each year and we want to use our sample data to calculate a value of *t* that will tell us whether it is likely that the null hypothesis is true.

$$t = \frac{\bar{x}_{1971} - \bar{x}_{2001}}{SE}$$ *t* from the deviation between population means

Our sample estimates of the means are easy - $\bar{x}_{1971}$ and $\bar{x}_{2001}$ are the least squares estimates of these parameters. What is our estimate of the standard error of the mean? With only one sample we used the standard deviation or the variance of the sample to estimate the standard error:

$$SE = \frac{s}{\sqrt{n}} = \sqrt{\frac{s^2}{n}}$$ the usual definition of sample standard error of the mean

With two samples, there are two estimates of variance, s^2_{1971} and s^2_{2001}. If we can assume that these two represent essentially the same value then we can pool them by taking the weighted average as our best estimate of SE.

Before pooling the variances from our 1971 and 2001 samples we need to test the hypothesis that they do not differ (i.e. H0: $s^2_{1971} = s^2_{2001}$). This hypothesis can be tested using the F distribution - a theoretical probability distribution that gives probabilities for ratios of variances. Incidentally, this distribution is named after the eminent statistician Ronald Fisher, inventor of ANOVA and maximum likelihood. We'll come back to F in later chapters, so here we will only note that the expected value of F if the null hypothesis is true is 1. Because the two estimates of variance are based on independent samples, F is actually a family of distributions, the exact shape of which depends on the degrees of freedom of both variances. One caveat about using the F distribution to compare variances is that it is quite sensitive to whether the population distributions are normal (Hays,1973, pp. 450-1).

So, if we want to know if the two estimates of variance are equal to each other we can simply take their ratio and test the probability of this ratio, given the degrees of freedom that went into each variance estimate. We do this with the F distribution because this distribution lets us

specify degrees of freedom for the numerator and the denominator of the ratio.

$$F = \frac{s^2_{2001}}{s^2_{1971}} = \frac{36.0876^2}{35.9284^2} = \frac{1302.32}{1290.85} = 1.0089$$ F test of equality of variance

It is pretty obvious that the variances are not very different from each other (36.1 ms versus 35.9 ms), so the variances are also very similar in magnitude and thus the F ratio is close to one. We look up the probability of getting an F of 1.0089 or higher using the R `pf()` function:

```
> pf(1.0089,25,17,lower.tail=F)
[1] 0.5034847
```

In this function call I specified the degrees of freedom of the numerator (n_{2001} - 1 = 25) and of the denominator (n_{1971} - 1 = 17) for the two estimates of variance that went into the F ratio. I also specified that we are looking at the upper tail of the F distribution because, as is usually done, I put the larger of the two variances as the numerator. The probability of getting an F value of 1.0089 or higher when the variances are in fact equal is quite high p=0.5 so we have no reason to believe that the variance of the 1971 data is any different from the variance of the 2001 data.

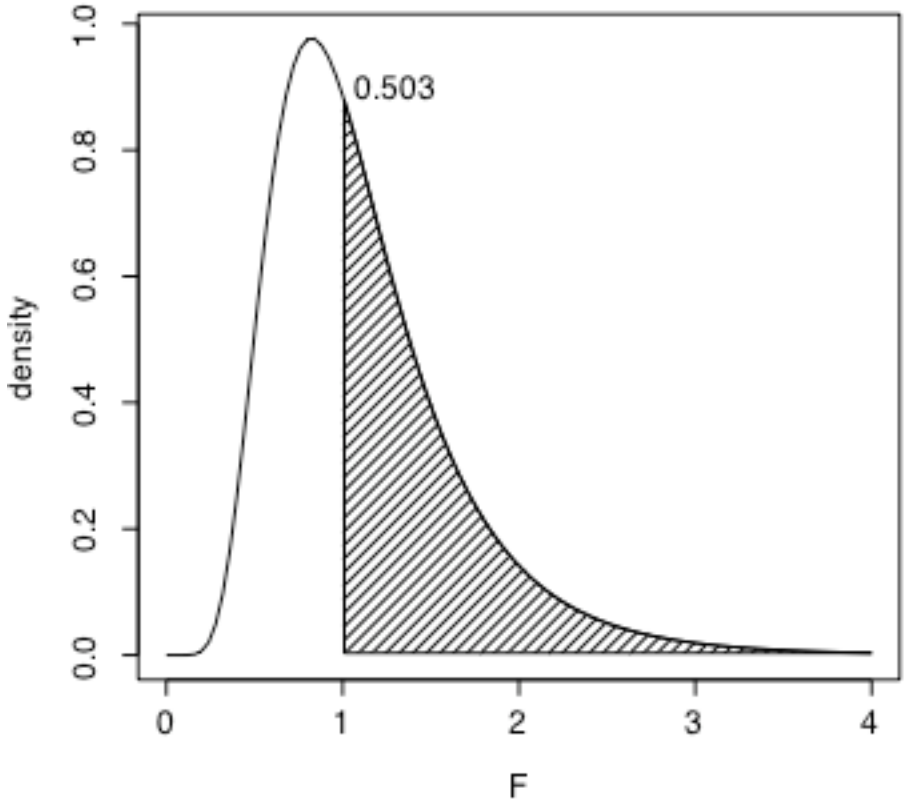

Figure 3.2. The F probability density distribution for 25 and 17 degrees of freedom showing that 0.503 of the area under the curve lies above the F value found by taking the ratio of variances from 2001 and 1971 in the Cherokee data.

Thus, because the sample variances in 2001 and 1971 are about the same, we can estimate SE for our test of whether VOT was different in 2001 than it was in 1971 by pooling the two sample variances. This is done using the weighted average of the variances where each variance is weighted by its degrees of freedom.

$$s_p^2 = \frac{(n_a - 1)s_a^2 + (n_b - 1)s_b^2}{(n_a - 1) + (n_b - 1)}$$ pooled variance, when the variances are roughly equal

The pooled variance for our Cherokee VOT data is thus 1297.7 and hence the pooled standard deviation is $s = 36.02$. We will also use the denominator in this weighted mean formula as the degrees of freedom for the *t* statistic that we calculate next.

The *t* statistic that we use then to compare two means uses the pooled variance from the two samples to estimate SE - the standard error of the mean(s), and *t* is a ratio of (1) the difference between the two means ($\bar{x}_a - \bar{x}_b$) with (2) SE calculated from the pooled variance. If the means differ from each other by much more than you would expect (the standard error) then we are likely to declare that the two samples (VOT 2001, and VOT 1971) have different means.

$$t = \frac{\bar{x}_a - \bar{x}_b}{\sqrt{s_p^2 / (n_a + n_b)}}$$

This works out to be t=2.6116 for our data, and the degrees of freedom is (18-1)+(26-1) = 42. So the probability of getting a *t* value this big if there is actually no difference between mean VOT in 1971 and 2001 (as compared with the hypothesis that VOT in 2001 was less than it was in 1971) is less than 1 in 100 ($p<0.01$). So we conclude that this speaker's VOT was shorter in 2001 than it was in 1971. This in itself is an interesting finding - perhaps indicating a gradual phonetic change that may be due to language contact. Of course, this result, based on one talker doesn't really tell us anything about Cherokee as a whole.

R note. I specified the *t* test contrasting Cherokee VOT in 1971 and 2001 with the command given below. I used the subset notation `VOT[year=="1971"]` to say that I wanted all of the VOT values for which the variable `year` is equal to "1971". I specified that the variance in the 1971 data is equivalent to the variance in the 2001 data with `var.equal=T`, and I specified that I expected VOT to be greater in 1971 than in 2001 because I hypothesized that this would be the direction of change if there was an impact of English on Cherokee.

```
> t.test(VOT[year=="1971"],VOT[year=="2001"], var.equal=T,
alternative="greater")

Two Sample t-test
data:  VOT[year == "1971"] and VOT[year == "2001"]
t = 2.6116, df = 42, p-value = 0.006223
alternative hypothesis: true difference in means is greater than 0  95 percent
confidence interval:
  10.2681     Inf
sample estimates:
mean of x mean of y
113.50000  84.65385
```

3.1.3 If the samples do not have equal variance.

What if we find that the variances of our two samples are not equal? Instead of using the pooled variance, we calculate the standard error of the mean as:

$$SE = \sqrt{(s_a^2/n_a) + (s_b^2/n_b)}$$ Standard error of the mean for unequal variances

The *t* value calculated with this estimated standard error ($t^* = \frac{\bar{x}_a - \bar{x}_b}{SE}$) follows the normal distribution if both samples have greater than 30 data points ($n_a > 30$ & $n_b > 30$). For smaller samples the t distribution is used with a degrees of freedom equal to:

$$df = \frac{U^2}{V^2/(n_a - 1) + W^2/(n_b - 1)} - 2$$

where

$V = s_a^2/n_a$,

$W = s_b^2/n_b$,

and

$U^2 = V + W$

The Welch correction of degrees of freedom

By adusting the degrees of freedom, this correction puts us on a more conservative version of the *t* distribution. Recall that *t* is a family of distributions and that the tails of the distribution grow as the df decreases - see figure 2.5. The Welch correction is done by default for two-sample *t* tests in R, so that in order to conduct the test with the assumption that the variances are equal you have to actually declare that `var.equal=TRUE`. In my experience, I've not noticed much difference between the two ways of calculating *t*, but maybe I haven't really dealt yet with a situation where the variances are strongly unequal.

R note. Below is the `t.test()` call when we do not assume that the variance in 2001 equals the variance in 1971. In this call, I used the function notation rather than specifying two vectors of numbers. You read the expression `VOT~year` as "VOT varies as a function of year". In formula notation, the "dependent variable" or the "criterion variable" appears on the left of the tilde (~) and the "independent variables" or the "predictive factors" appear on the right side of the tilde. Because `year` only has two levels in this data set (1971 and 2001) we can use this notation to describe the desired *t* test.

This test reports that the Welch correction of degrees of freedom was used (notice that we now have df = 36.825, instead of df=42). The *t* statistic is also slightly different because the SE used in calculating *t* was different in this test than in the one for equal variance. The conclusion that we draw from this test is the same as the conclusion we arrived at when we assumed equal variances ($s^2_{1971} = s^2_{2001}$).

```
> t.test(VOT ~ year, alternative="greater")

      Welch Two Sample t-test

data:  VOT by year
t = 2.6137, df = 36.825, p-value = 0.006448
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 10.22436      Inf
sample estimates:
mean in group 1971 mean in group 2001
         113.50000           84.65385
---------------------------
```

3.1.4 Paired *t* test: Are men different from women?

There is a particularly powerful and easy-to-use kind of *t* test if you have observations that are meaningfully paired. What "meaningfully paired" means is that the observations naturally come in pairs. For example, there is no rational way to pair VOT measurements from 1971 with measurements taken in 2001. We could suggest that VOTs taken from /t/ should be matched with each other, but there is no meaningful way to choose which 2001 /t/ VOT should be paired with the first /t/ VOT on the 1971 list, for example. Now, if we had measurements from the same words spoken once in 1971 and again in 2001 it would be meaningful to pair the measurements for each of the words.

In chapter 2 though, we looked at a data set for which the observations are meaningfully paired. The first formant data in "F1_data.txt" was given for men and women for each language and vowel in the data set, so that it is natural to match, for example, the male F1 of /a/ in Sele with the female F1 of /a/ in Sele, the male F1 of /i/ in Sele with the female F1 of /i/ in Sele, and so on. Figure 3.3 shows that men and women tend to have systematically different vowel F1 frequency, but that the difference between vowels can be bigger than the overall male/female difference. To have a sensitive test of the male/female difference we need to control for the vowel differences. Pairing male/female differences by vowel provides just such control.

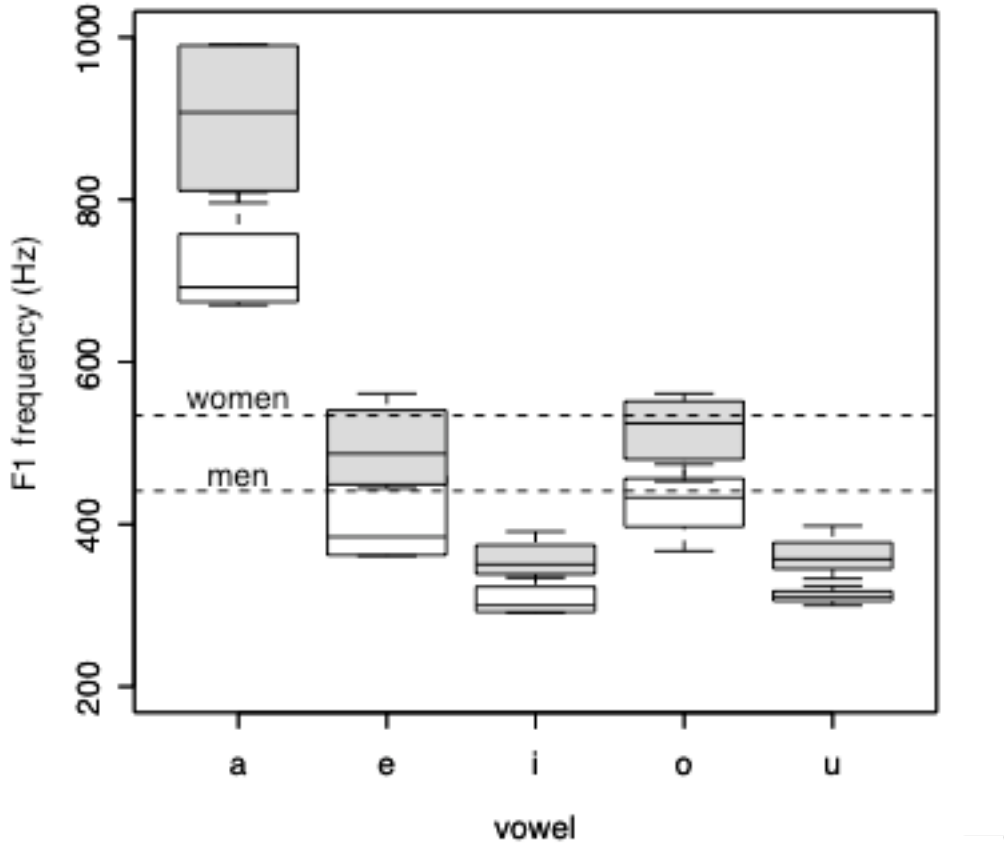

Figure 3.3. Boxplot comparing male (white boxes) and female (gray boxes) vowel F1 frequencies for five different vowels. The overall average F1 values for men and women are plotted with dashed lines.

With paired observations we can then define a derived variable - the difference between the paired observations

$d_i = x_{ai} - x_{bi}$ the difference between paired observations

Then calculate the mean and variance of the difference as we would for any other variable:

$$\overline{d} = \frac{\sum d_i}{n} \quad , \quad s_d^2 = \frac{\sum \left(d_i - \overline{d}\right)^2}{n-1}$$ the mean and variance of the differences

Now, we test the null hypothesis that there is no difference between the paired observations, i.e. that H_0: $\overline{d} = 0$. The *t* value, with degrees of freedom *n*-1, is;

$$t = \frac{\overline{d}}{\sqrt{s_d^2 / n}} \qquad \text{does } \overline{d} \text{ differ from zero?}$$

The beauty of the paired *t* test is that pairing the observations removes any systematic differences that might be due to the paired elements. For example, this measure of F1 difference is immune to any vowel or language influences on F1. So, if F1 varies as a function of language or vowel category these effects will be automatically controlled by taking paired F1 measurements from the same vowels spoken by speakers of the same language. In some other situations we are most interested in letting each person serve as his/her own control, because if we sample from a population of people, each of whom may differ from other members of the population in some systematic way, paired observations from the same person provides a control over individual sources of variation. Consequently, the paired *t* test tends to be much more sensitive (powerful) than the two sample *t* test.

The increased power of the paired t-test is seen in a comparison of "independent samples" and "paired comparisons" tests of the null hypothesis that male F1 frequencies are no different from female F1 frequencies. We'll contrast this hypothesis with the H_1, suggested by years of research, that female F1 is higher than male F1.

The independent samples comparison suggests that men and women are not reliably different from each other [$t(36) = 1.5$, $p = 0.067$], while the paired comparison results in a much higher t value [$t(18) = 6.1$, $p < 0.01$] showing a much more reliable difference. Evidently, in the independent samples comparison, the gender difference was swamped by variation due to vowel and perhaps language differences, and when we control for these sources of variation by contrasting men and women's F1 values for matched vowels and languages, the gender difference is significant. Note that in this report of the results, I'm using a standard style for reporting *t* test results. Put the *t* value and its associated degrees of freedom and probability value in square brackets. The degrees of freedom is in parentheses after the letter "t", and the probability is reported as p=0.06 if the value is above the type I error criterion α (0.01), or if the probability of *t* is less than α report p<0.01. It is preferable to round off *t* to one or two places after the decimal point.

R note. The two *t* tests contrasting vowel F1 frequencies for men and women in the "F1_data.txt" data set are shown below.

```
> f1 <-read.delim("F1_data.txt")
> attach(f1)

> t.test(female,male,alternative="greater",var.equal = T)

      Two Sample t-test

data:  female and male
t = 1.5356, df = 36, p-value = 0.0667
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -9.323753       Inf
sample estimates:
mean of x mean of y
 534.6316  440.8947

> t.test(female,male,paired=T,alternative="greater")

      Paired t-test

data:  female and male
t = 6.1061, df = 18, p-value = 4.538e-06
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 67.11652      Inf
sample estimates:
mean of the differences
               93.73684
----------------------
```

3.1.5 The sign test

A quick and easy variant of the paired *t* test works off of the signs of the differences - no calculation of the mean or standard deviation of the differences, just coding whether the difference for each pair is positive or negative. But keep in mind that the "sign test" is only valid if you have 25 or more pairs.

Here's how it works. Mark each paired observation according to whether d_i is greater than zero or less than zero. Let n^+ be the number of pairs with d_i greater than zero, and n^- be the number of pairs with d_i less than zero. The probability of the null hypothesis $\bar{x}_a = \bar{x}_b$ is given by the probability of having a smaller or larger (two-tailed) *z* score:

$$z = \frac{\left|n^+ - n^-\right| - 1}{\sqrt{n^+ + n^-}}$$ z score used in the sign test.

3.2 Predicting the back of the tongue from the front: Multiple regression

Suppose that you have a method for tracking the locations of 3 or 4 small gold pellets or electromagnetic transducers that are glued onto the upper surface of the tongue. Phoneticians do this (X-ray microbeam, EMMA). Such a point tracking system provides some very valuable information about speech production, but the root of the tongue is not represented.

Now, suppose that data about the location of points on the top surface of the tongue could be used to predict the location of a point on the root of the tongue. It seems to be reasonable that we could do this. After all, we are dealing with a physical structure with extrinsic tongue muscles that move the tongue up, front, and back as whole. Additionally, because the tongue is an incompressible mass, the intrinsic tongue muscles that reshape the tongue produce predictable effects (like squeezing a water balloon, squeeze the back of the tongue and the top goes up).

So the nature of the tongue suggests that we might be able to make good guesses about the root of the tongue (which can't be tracked in the current systems) from the information that we have about the front of the tongue. Multiple regression is the key.

We saw in Chapter 2 that we can define a regression line $y=a+bx$ that captures whatever linear relationship might exist between two variables, and that we can measure the strength of linear association between two variables with the Pearson's product moment correlation coefficient r.

Now we will extend this idea to consider regression equations that have more than one predictor variable , e.g. $y = a + b_1x_1 + b_2x_2 + ... + b_nx_n$. Such equations have some valuable practical applications in phonetics, one of which we will explore, but they also provide a formal basis for building models of causal explanations in many domains of scientific inquiry.

3.2.1 The covariance matrix.

We have some data that shows the top of the tongue as well as the root (Johnson et al., 1992). So we might be able to use this data to develop a regression equation that will let us predict the location of the back of the tongue from the front.

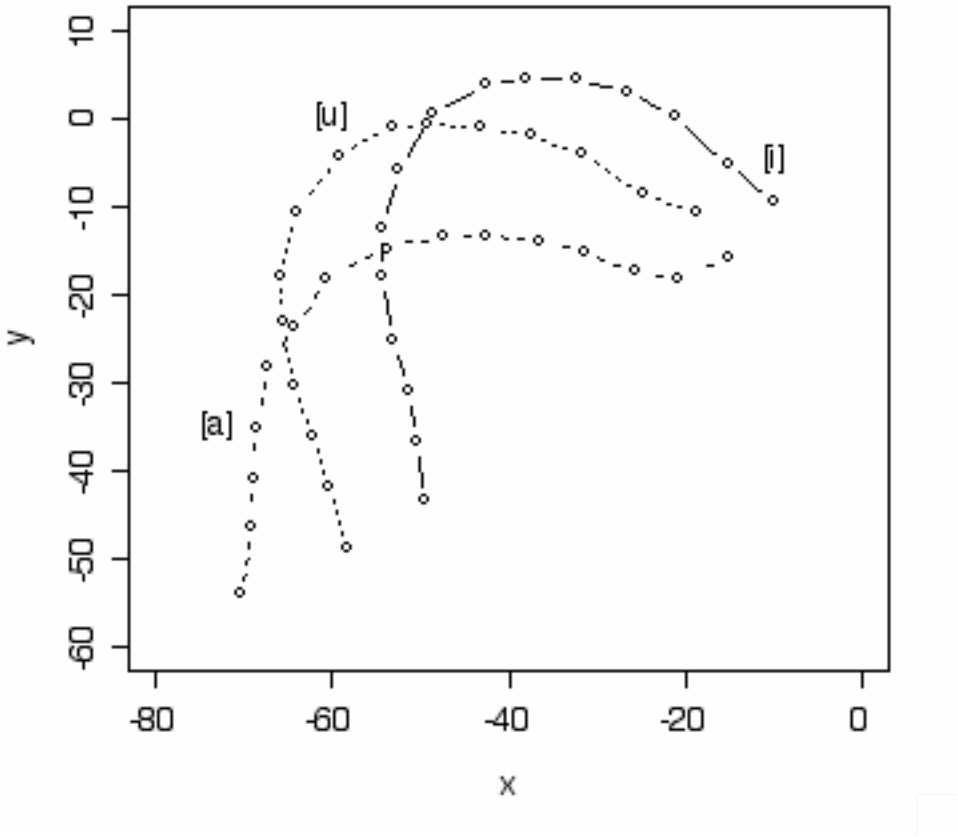

Figure 3.4. A chain of fifteen small gold pellets was draped over the tongue and the positions of the pellets were recorded from x-ray scans.

A chain of fifteen small gold pellets was draped over the tongue of one of the talkers in Johnson et al. 1992. This talker unlike the others was able to tolerate having the chain go very far back in the mouth extending back over the root of the tongue. X-ray images were taken of the tongue position during 11 different vowels, and figure 3.4 shows the tongue shape that was recorded in the *x* and *y* locations of the pellets for the corner vowels [i], [a], and [u].

The fifteen pellets in the chain each have an *x* and a *y* location in space, so we have 30 variables in this data set. Some of them are highly correlated with each other. For example, the vertical (*y*) location of the 15th pellet is correlated with the *y* location of the 14th pellet with $r = 0.9986$. So with a regression formula we could predict 99.7% of the variance of pellet 14's *y* location if we

know the *y* location of pellet 15. There is some indication also, that it may be possible to predict the *xy* location of one of the back pellets from one of the front pellets. For example, the correlation between the *y* locations of pellet 1 and pellet 14 is $r = 0.817$. However, the highest correlation between the *x* location of pellet 14 and any of the front pellets is with the *y* location of pellet 6 ($r = 0.61$).

We could explore this data set for quite some time calculating correlations among the 30 variables, but with so many correlations to look at (435 *r* values) and with so many of them falling at approximately the same level, it isn't clear how to pick and choose among the correlations to highlight comparisons that are especially meaningful. Of course we could show all 435 of the *r* values in a table, but that would be mean and a waste of paper.

R note. Perhaps it is mean and a waste of paper to show all 435 of the correlations among 30 variables in a table, but is sure is easy to produce such a table in R! In the data set "chaindata.txt", I named the *x* location of pellet 1, `x1`, and so on. `subset()` is used to extract data for talker PL1 (there are five other talkers in the data file, though only PL1 had 15 points in the chain of gold pellets - the others didn't tolerate the chain as far back on the tongue). If you use `subset()` to extract data for any of the other talkers you will need to specify which of the thirty variables to extract more laboriously (e.g. `x1:x12,y1:y12`) than I did in this command where I could specify x1 through y15 with `x1:y15`.

The correlation matrix is produced by `cor()`. Note that x1 is perfectly correlated ($r = 1$) with itself. Note also that the matrix is symmetrical - the correlation of x1 and x2 is the same in row x2, column x1 as it is in row x1, column x2 ($r = 0.9972439$).

```
> chain <- read.delim("chaindata.txt")
> PL1 <- subset(chain,talker=="PL1",x1:y15)
> cor(PL1)
```

```
           x1        x2        x3        x4        x5        x6
x1  1.0000000 0.9972439 0.9586665 0.9727643 0.9485555 0.9313837
x2  0.9972439 1.0000000 0.9737532 0.9839730 0.9647815 0.9452476
x3  0.9586665 0.9737532 1.0000000 0.9911912 0.9933240 0.9844554
x4  0.9727643 0.9839730 0.9911912 1.0000000 0.9939075 0.9706192
x5  0.9485555 0.9647815 0.9933240 0.9939075 1.0000000 0.9791682
x6  0.9313837 0.9452476 0.9844554 0.9706192 0.9791682 1.0000000
x7  0.8826467 0.9045604 0.9676897 0.9444815 0.9590505 0.9857084
x8  .....
```

The covariance matrix is also easy to produce, using `cov()`. Note that both of these functions

(`cor()` and `cov()`) are able to work with `PL1` because in the subset command I extracted continuous numeric vectors into the subset leaving out the `talker` and `vowel` columns of the original data set. If I had used `subset(chain,talker=="PL1")` each row would have been labeled with a vowel label copied from the original spreadsheet. Then the `x1:y15` columns would have to be specified explicitly in `cor()` and `cov()`.

```
> cov(PL1)

            x1         x2         x3        x4         x5         x6
x1   18.688494  20.257932  19.559742  20.71310  18.317303  16.912481
x2   20.257932  22.080714  21.595530  22.77404  20.251032  18.657079
x3   19.559742  21.595530  22.274955  23.04179  20.941653  19.516232
x4   20.713102  22.774042  23.041793  24.26056  21.867947  20.081253
x5   18.317303  20.251032  20.941653  21.86795  19.953692  18.372171
x6   16.912481  18.657079  19.516232  20.08125  18.372171  17.643432
x7   16.101781  17.936760  19.272782  19.63106  18.078109  17.471889
x8    .....
--------------------------
```

The patterns of interrelatedness, and perhaps causal relationships in the data are implicit in the correlation and covariance matrices and there are several techniques that can be used to discover and visualize these patterns (one of which will be the topic of section 3.3). In the remainder of this section though, we will examine a stepwise method to find the best linear combination of variables that can be used to predict the value of a criterion variable - in this case we want to predict the location of the root of the tongue (which is hard to observe directly) from the locations of pellets on the top of the tongue (which can be tracked without too much trouble).

3.2.2. More than one slope: the β_i

The best way to predict the *y* location of pellet 15 on the root of the tongue from the locations of pellets 2 through 6 on the top of the tongue is with the following formula:

$$y_{15} = -16.81 + 1.28y_2 + 3.85y_6 - 4.11x_5 + 1.47x_6 - 1.15x_5$$

This formula, a linear combination of some of the tongue pellet *xy* variables, produces an estimate of y_{15} that accounts for 98% of the variance of the *y* location of pellet 15.

Similarly, we can predict the *x* location of pellet 15 from the front pellets with this formula:

$$x_{15} = -51.69 - 0.97y_5 + 1.05x_2 - 4.04x_6 + 4.66x_4 + 0.61y_2 - 3.69x_3 + 2.66x_5 + 1.48y_4$$

This linear equation predicts 99% of the variance of the *x* location of pellet 15. With these two equations we should be able to make pretty good guesses about the location of the root of the tongue based on our knowledge of the top of the tongue. Of course, this could be a very valuable bit of knowledge as we attempt to model processes of speech articulation on the basis of x-ray microbeam, or EMMA data.

Now the method for coming up with these linear equations is not exactly magic, but it is still pretty neat.

First, let's recall that the equation for the regression line with one variable ($\hat{y} = A + Bx$) can be expressed as $\hat{z}_y = rz_x$ using the standardized *z* scores. Then from this we get the slope *B* from the correlation *r* and the ratio of standard deviations: $B = r\frac{s_x}{s_y}$. In determining the β coefficients in the multiple regression equation $\hat{z}_y = \beta_1 z_{x_1} + \beta_2 z_{x_2} .... \beta_n z_{x_n}$ we must take into account the correlations of the *x* predictor coefficients as well as the correlation between *x* and *y*. Unlike our calculation for r which is based on the product of *z* scores, values from the correlation matrix are used to compute the β values. The coefficient for predictor variable x_1 is based, naturally enough, on the correlation between variable x_1 and the variable (*y*) that we are trying to predict. But, the formula for β_1 removes from r_{y1} an amount that is based on how much x_2 can predict *y* and how closely correlated x_1 and x_2 are to each other.

$$\beta_1 = \frac{r_{y1} - r_{y2}r_{12}}{1 - r_{12}^2}$$ the coefficient that relates x1 to y.

This formula, and others like it for coefficients in larger models, are called beta weights and because they are coefficients for the standardized variables z_y, z_{x1}, z_{x2}, etc. they indicate the relative strength of an *x* variable in predicting the *y* variable. To get the non standardized coefficients, like those that I used in the predictive formulas for y_{15} and x_{15} above, you scale the β weights by the ratio of predicted and predictor standard deviations just like we did when there was only one *b* to be calculated.

$$b_i = \beta_i \frac{s_y}{s_i}$$ scaling by ratio of standard deviations to get *b* from β

R-note. The R function `lm()` finds the regression coefficients in linear models. For example, I used the following command to find the coefficients for the equation for y_{15} that started this section. Please read down the "Estimate" column and compare these numbers with the b_i values in the formula for y_{15}.

```
> summary(lm(y15 ~ y2 + y6 + y5 + x6 + x5, data = PL1))

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -16.8100     6.8275  -2.462  0.05708 .
y2            1.2785     0.1937   6.599  0.00120 **
y6            3.8458     0.6110   6.294  0.00149 **
y5           -4.1140     0.6104  -6.740  0.00109 **
x6            1.4703     0.8134   1.808  0.13048
x5           -1.1467     0.8165  -1.404  0.21919
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 1.304 on 5 degrees of freedom
Multiple R-Squared: 0.9826,   Adjusted R-squared: 0.9652
F-statistic: 56.46 on 5 and 5 DF,  p-value: 0.0002130
```

Similarly, the coefficients of the x15 model are given by a call to `lm()`.

```
> summary(lm(x15 ~ y5 + x2 + x6 + x4 + y2 + x3 + x5 + y4, data = PL1))

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -51.6910     8.7933  -5.878  0.02774 *
y5           -0.9702     0.5103  -1.901  0.19767
x2            1.0540     0.4120   2.558  0.12482
x6           -4.0357     0.3624 -11.135  0.00797 **
x4            4.6618     0.7939   5.872  0.02780 *
y2            0.6083     0.2932   2.075  0.17370
x3           -3.6891     0.5072  -7.273  0.01838 *
x5            2.6555     0.8803   3.017  0.09456 .
y4            1.4818     0.7193   2.060  0.17559
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.5194 on 2 degrees of freedom
Multiple R-Squared: 0.9989,   Adjusted R-squared: 0.9943
F-statistic: 220.1 on 8 and 2 DF,  p-value: 0.00453
```

The print out given by `summary()` reports t tests for each of the coefficients in the linear equation. The test compares the given coefficient value with zero and in general we want to include in the regression equation only coefficients that are reliably different from zero. However, the method that I used to select the regression models shown here relies on a measure

of the effectiveness of the overall model, and not just on the t tests of the coefficients. So, we have one case (x3 as a predictor of y15) of a coefficient that is not significantly different from zero on one test, but which is a reliable predictor when we compare different possible models with each other.

The `summary()` also reports "Multiple R-Squared" and "Adjusted R-squared".

3.2.3 Selecting a model.

So, in general that is how the regression coefficients (the b_i values in the regression equation $y = a + b_1x_1 + b_2x_2 + ... + b_nx_n)$ are found. There is an art though, to selecting a model. I restricted the model search for predicting the root of the tongue from the top of the tongue to models that use some combination of the x and y locations of pellets 2, 3, 4, 5 and 6. I felt that pellet 1 was sometimes not actually on the surface of the tongue (see the low back vowel [a] in figure 3.3), and I didn't want to go back further than pellet 6. Even with this restriction though there are 10 variables in the predictor set (5 pellets, *x* and *y* dimensions) so there are 2^{10} possible regression models. Choosing the best one from this set of possibilities is the subject of this section.

The procedure that is illustrated here is only one of several approaches that exist. These different approaches have their advocates and a discussion of the pros and cons of them is beyond the scope of this chapter - though we will return to this later. Perhaps one lesson to draw from the fact that there is more than one way to do this is that choosing the correct model is a bit of an art. The balancing act in this art is between high within-dataset accuracy on the one hand, and high predictive accuracy for new data on the other. So, with enough parameters you can account for all of the variation that is present in a set of data - particularly a tiny data set like the one we are looking at in this illustration of multiple regression. However, such "over-fitted" models are brittle and tend to break down when confronted with new data. So we are looking for a model that will use only a few parameters to fit the data - get as good a fit as possible with a minimum of predictive variables. For this, the `step()` function in R uses the Akaike Information Criterion. This criterion uses a log-likelihood measure of how well the model fits the data and adds a penalty for each new parameter (regression variable that is added to the model). So, to overcome the "added parameter" penalty the model predictions have to improve by a substantial amount. Tiny improvements don't make the cut.

Log-likelihood is a way of measuring model fit that is analogous to least-squares. We call the arithmetic average the least-squares estimate of central tendency because this measure minimizes the squared deviations. A log likelihood estimate seeks to maximize likelihood of the model and is a bit like an information measure. At this point, that's about all I'm going to say about it, but

we will be returning to this topic in the sociolinguistics chapter.

For now, I present the the formula for AIC acknowledging that it has a magical part L(M) the likelihood of model M. Realizing that this is a measure of model fit we can see that AIC is composed of two terms, the first is related to model fit and the second is related to model size, where n_m is the number of coefficients in the regression equation.

$$AIC = -2\log L(M) + 2n_M \qquad \text{Akaike Information Criterion}$$

R note. I used `step()` to select a model to predict the y location of pellet 15 in the `PL1chain` data set (which is assumed because of an earlier `attach(PL1chain)` command). The initial model has only one parameter - the intercept value. This is specified with `y15~1`. The largest model I want to consider has the *xy* locations for pellets 2, 3, 4, 5, and 6.

```
> summary(y.step <- step(lm(y15 ~ 1,data=PL1),y15~ x2+y2 + x3+y3+ x4+y4 +
x5+y5+ x6+y6))
Start:  AIC= 43.74
 y15 ~ 1

       Df Sum of Sq    RSS    AIC
+ y2    1    400.03  88.79  26.97
+ y3    1    323.32 165.50  33.82
+ y6    1    315.23 173.59  34.35
+ y4    1    301.93 186.90  35.16
+ y5    1    296.26 192.56  35.49
+ x4    1    121.77 367.06  42.58
+ x2    1    114.65 374.18  42.80
+ x5    1    105.75 383.08  43.05
+ x3    1     93.55 395.28  43.40
<none>              488.82  43.74
+ x6    1     56.78 432.05  44.38
```

What the print out shows is the first step in the stepwise search for the best combination of variables to include in the model. A plus sign in the first column indicates that the program considered what would happen if the variable named on that row was added to the model. The result of adding y_2 for example is that the sum of squared variance accounted for would go up leaving a small residual sum of squares and a smaller AIC value. The decision to add y_2 to the model at this step is taken because this model (with y_2 added) results in the lowest AIC as compared with any other model that is considered at this step. In particular, adding this variable

reduces the AIC when compared with no change in the model <none>. So, the program adds y_2 and considers adding other variables.

```
Step:  AIC= 26.97
 y15 ~ y2

       Df Sum of Sq    RSS    AIC
+ y3    1     20.84  67.95  26.03
+ y4    1     15.87  72.92  26.81
<none>               88.79  26.97
+ y5    1     10.68  78.11  27.56
+ x2    1      2.83  85.96  28.62
+ y6    1      2.10  86.69  28.71
+ x4    1      1.78  87.01  28.75
+ x3    1      0.93  87.86  28.86
+ x6    1      0.61  88.18  28.90
+ x5    1      0.50  88.29  28.91
- y2    1    400.03 488.82  43.74
```

So, the starting point for this step is the `y15 ~ y2` model, and the printout shows that the best possible thing to do now is to add y_3 to the model, and the worst possible thing to do is to remove y_2 (the row marked "`- y2`"). The "do nothing" approach <none> is also less good than adding y_3. One thing to notice here is that the ordering of the variables has changed from step 1 to step 2. At step 1 the second best thing we could have done would have been to add y6 to the model. At this step y6 appears in third place on the list. Evidently the correlation between y2 and y6 has resulted in a situation that when we add y2 to the model now y6 doesn't make such a large unique contribution in predicting the value of y15. Ready for step 3.

```
Step:  AIC= 26.03
 y15 ~ y2 + y3

       Df Sum of Sq     RSS     AIC
+ y6    1    32.429  35.525  20.896
+ y5    1    11.557  56.397  25.980
<none>               67.954  26.030
- y3    1    20.836  88.790  26.972
+ x2    1     3.243  64.711  27.492
+ y4    1     3.049  64.905  27.525
+ x5    1     2.612  65.342  27.599
+ x4    1     2.292  65.662  27.653
+ x3    1     2.175  65.779  27.672
+ x6    1     1.706  66.248  27.751
- y2    1    97.548 165.502  33.822
```

Now we can add y6! Notice that the option to remove y3 is pretty far up the list. One other thing to notice is that by optimizing AIC we give up on having the smallest possible residual error from our model. The idea is that we are seeking a stable model that produces a good fit to the data, and the AIC approach scores stability in terms of the information gain, per coefficient added to the model. This tends to keep out weak, unstable coefficients so that we avoid modeling the randomness of the data set and instead only capture robust relationships.

This process of adding variables, considering the possibility of making no change to the model, or even of removing variables that had been added at a previous step, is continued until the <none> option appears at the top of the list. Here's a summary of the steps that were taken on this run:

step 1	+ y2	
step 2	+ y3	
step 3	+ y6	
step 4	+ y5	
step 5	+ x6	
step 6	- y3	remove y3
step 7	+ x5	
step 8	<none>	finished

The resulting model, with the variables that survived the process of adding and removing variables (`y15 ~ y2 + y6 + y5 + x6 + x5`) is declared the winner because it produced the lowest AIC value.

The procedure for finding the best predictive model of x15 is exactly comparable to this.

```
> summary(x.step <- step(lm(x15 ~ 1,data=PL1),x15~ x2+y2 + x3+y3+ x4+y4 +
x5+y5+ x6+y6))
----------------------
```

How good is 99% variance accounted for? There are only 11 different vowels in the PL1 chain data, and we are using 8 parameters to estimate the 11 x15 data values and 6 parameters to estimate the 11 y15 values. So, it isn't too surprising that we can get very high variance accounted for in these cases. A more robust and realistic use of multiple regression would draw on many more observations providing presumably more variance to be accounted for and more variance in the values of the predictor variables as well.

Despite these caveats regarding this tongue position example, I decided to try the regression equations for predicting the location of pellet 15 by using the average pellet locations for pellets

2, 3, 4, 5 and 6 to predict pellet 15 (Figure 3.5). Though the regression never saw the average tongue shape it does an excellent job of predicting the average location of pellet 15 from the top of the tongue pellets.

This is a very informal evaluation of a model fit, more like a "gee whiz" demo. What is really needed is a dataset large enough to split into two pieces. Then the first piece could be used to produce the fit and the remainder could be used to make predictions and evaluate the model by measuring the differences between the predicted and the actual locations of pellet 15 for tokens that weren't used to build the model (see chapter 7 for an example of this method).

What about the other pellets? If we can predict the location of pellet 15 why not use the same method to predict 14, 13, 12, etc.? In other words, doesn't the covariance matrix contain information that would make it possible for us to characterize all of the pellets at once?

The answer is "yes" and this leads us to principal components analysis.

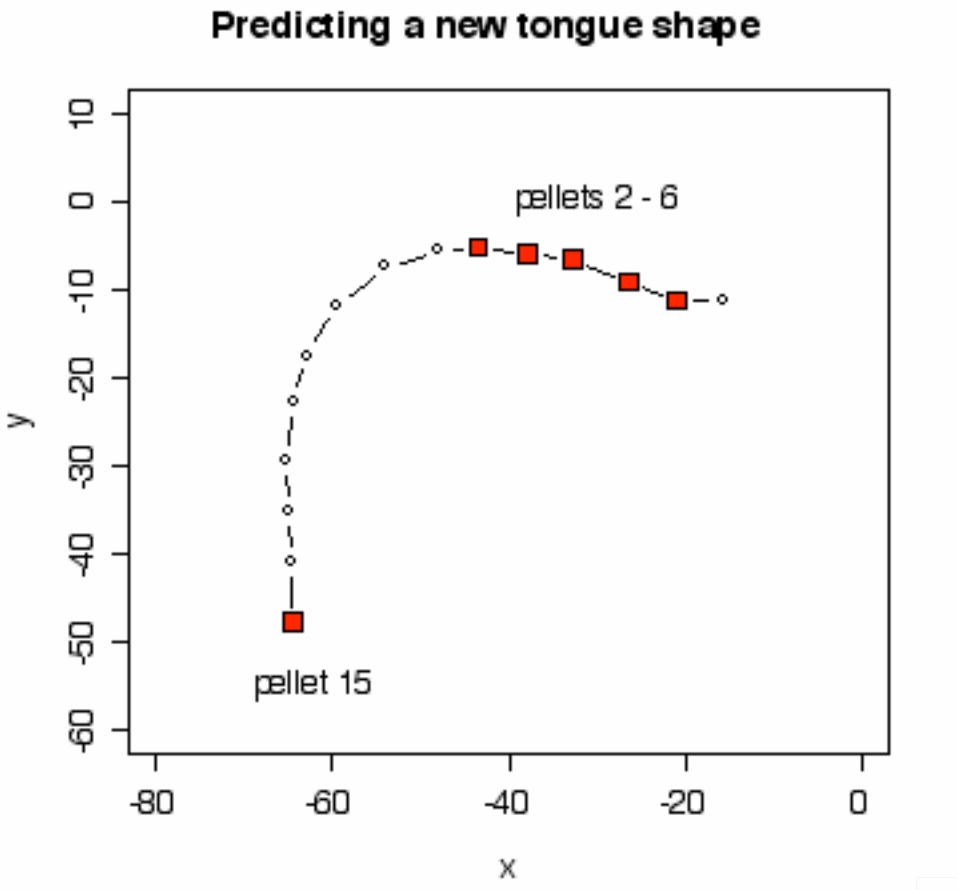

Figure 3.5. The average pellet locations - averaged across 11 vowels. The locations of five pellets on the top of the tongue were used to predict the location of pellet 15. .

3.3 Tongue shape factors: Principal components analysis

One of the primary aims of quantitative analysis in phonetics, as well as in other subareas in linguistics, is to discover underlying factors that in some sense "cause" the pattern of data that we observe in our studies.

As we saw in section 3.2, adjacent points on the tongue are not physically or statistically independent of each other. The natural next question is: how many independent parameters of tongue movement are there? To answer this question we will use principal components analysis (PCA) to extract independent factors of tongue movement from the covariance matrix.

The main idea with PCA is to use the correlations of raw data variables to define abstract components of correlated variation. With tongue motion, we want to find the main components of tongue shape variation. As we saw earlier, the second point and the third point on the tongue are highly correlated with each other, suggesting that we could account for both of these variables and probably others by a single abstract factor. The linguistic analysis of tongue position in vowels suggests that there may be only two factors - tongue height and tongue backness.

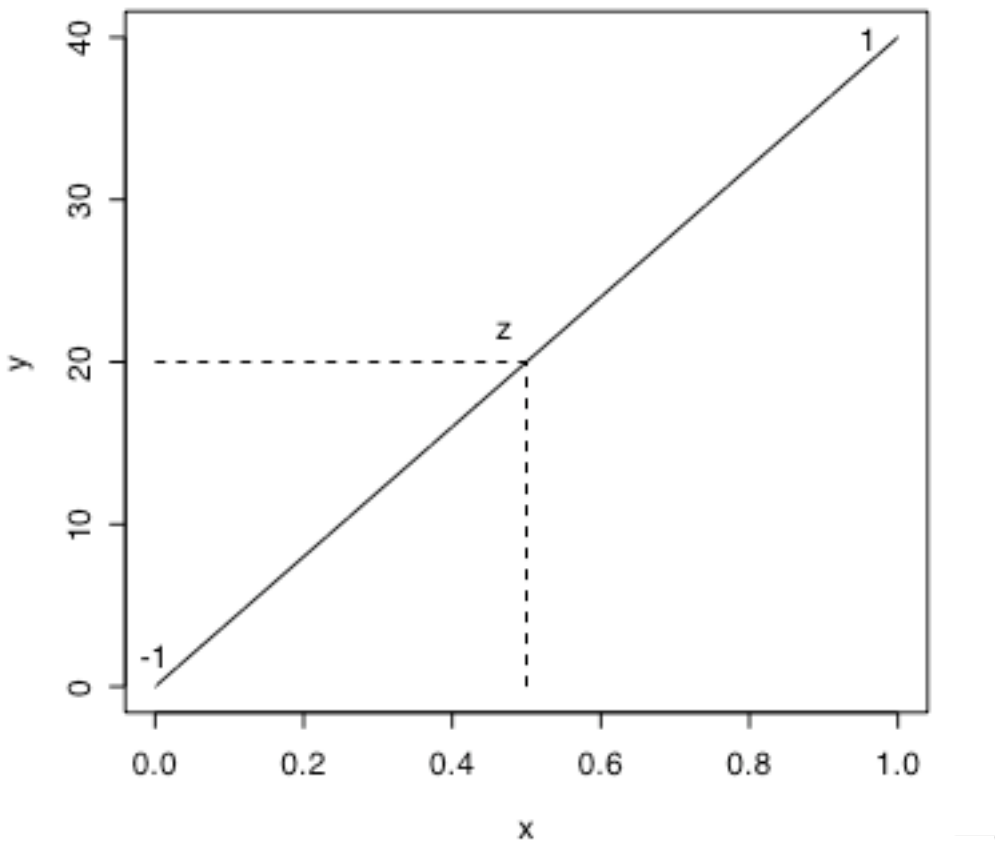

Figure 3.6. A derived factor is a line that relates x and y to each other. By knowing one value on the factor axis (z) you know two values (x and y) in the raw data.

I think of principal components analysis as a series of multiple regression analyses. We saw that in regression we can describe the relationship between two variables as a line so that for any one location on the regression line you know the value for both of the correlated variables that define the line. This is a kind of data reduction technique because we go from two axes, x and y, to one derived line, a new axis F. So, imagine (see figure 3.6) that our derived factor F is a line that captures the correlation between x and y, and we let values on this line range from -1 to 1. When we say that an observation has a value (z) on the factor axis - the regression line - we can expand this into a statement about the x and y values of this observation. In Figure 3.6, I put the value of z at 0 on the factor axis (halfway between -1 and 1). This corresponds to an x value of 0.5 and a y value of 20. In this way, principal components analysis uses correlations among raw data

variables to define the main components of variation in our data set and reduce the dimensions of description from many variables to only a few principal components of variation.

How it works. With two variables, x and y, it is fairly easy to imagine how principal components analysis works. We rotate the two dimensional space so that the slope of the best fitting regression line is zero and the y intercept is zero. Now we have a horizontal line - our new derived axis replaces the two raw variables x and y. No further reduction of dimensions is possible because we have reduced from two dimensions (x and y) to one.

With a larger number of variables, such as the x and y locations of pellets on the tongue, it is harder to visualize the principal components. The strategy that I and others have taken with tongue shape data is to plot the principal components (the "loadings" matrix, as discussed below) in the original data dimensions. For example, the first principal component of tongue shape usually corresponds to the high/low distinction and the second principal component corresponds to the front/back distinction.

Because a component of variation is a way of describing deviation from the mean, we start with the average values of the x and y variables in our tongue shape data for talker PL1. We only have 11 vowel shapes to work with, so we can have no more than 11 variables in the input (recall that with 15 pellets in two dimensional space we have 30 variables). Therefore, I choose five pellets to represent the general shape of the tongue. The average x and y locations of these five pellets is shown in figure 3.7.

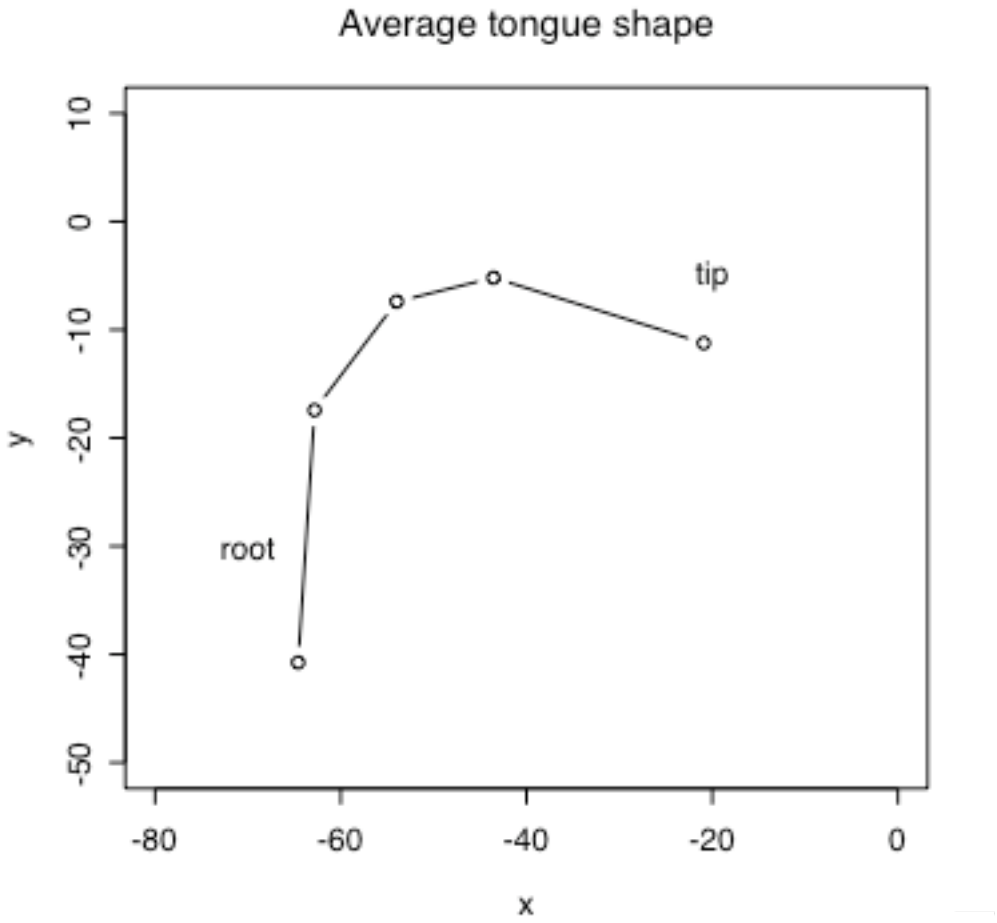

Figure 3.7. The average location of the five pellets that were analyzed.

The correlations among the 10 input variables (xy locations of 5 pellets) are analyzed in `princomp()` and a principal components analysis is calculated. This model has 10 principal components but the first two components account for most of the variance in the tongue shape data. As Figure 3.8 shows, the first principal component accounts for 66% of the variance and the first two components together account for 93% of the variance. Each additional component accounts for a very small amount of variance (5% for component 3, 2% for component 4, and less than 1% for all the others). Therefore, our most robust results are to be found in looking at the first two components. The higher components are likely to code minor quirks that are only present in these particular data that won't generalize to other similar data sets.

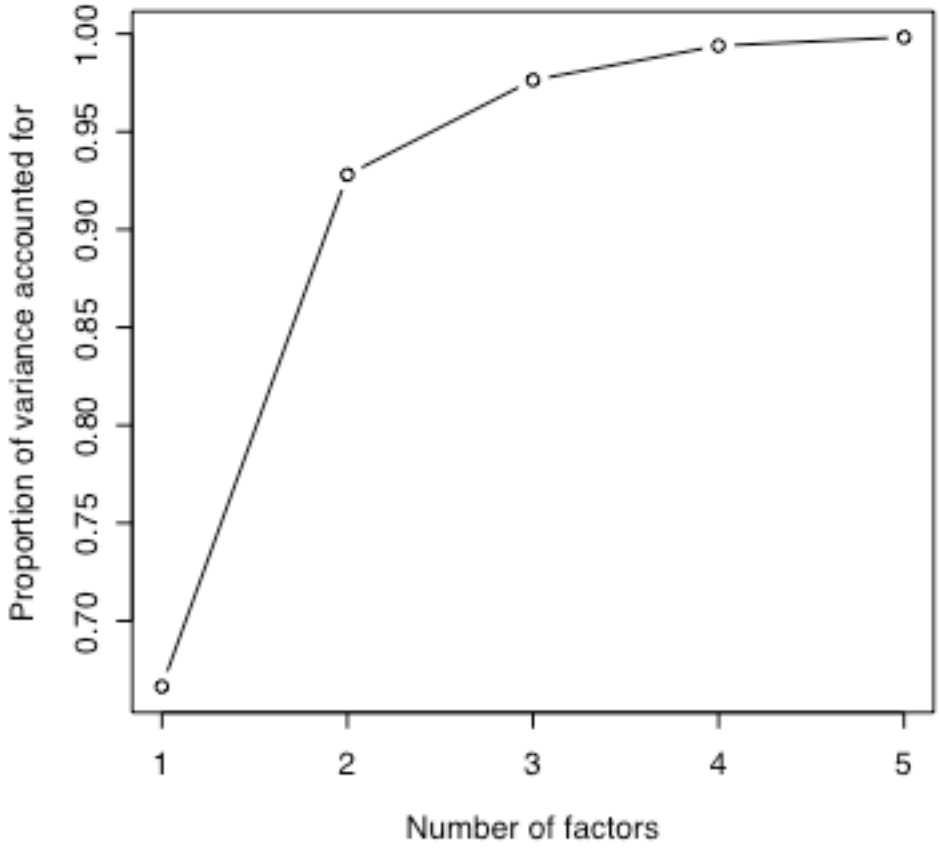

Figure 3.8. The cumulative proportion of variance accounted for by the first five principal components of tongue shape.

We are using principal components analysis to discover the main correlated patterns of variation that distinguish vowels in the tongue shape data set. We have found so far that the variation in these ten variables can be described in terms of just two dimensions of variation. Another way of thinking about this is to say that any particular tongue shape in the data set can be described as a combination of two basic shapes - two "principal components" of variation. This is because principal components analysis decomposes the data matrix X into two matrices V and U, such that $X=V^T * U$, where V is the set of principal components (also called the "loadings" matrix), and U is the set of "scores" on those components for each observation. You can reconstruct the data values X from the scores and loadings. The loadings matrix is an abstraction over the data set that captures a few main patterns of correlation that are present in the data set, and the scores

matrix indicates for each vowel how much of each principal component (positive or negative) can be found in the vowel's tongue shape. With two principal components each vowel will be represented with only two numbers - a score for principal component one and a score for principal component two. The PCA loadings will translate these scores into detailed tongue shapes.

To visualize the two main principal components of tongue shape we will plot values from the loadings matrix to draw pictures of the tongue shape differences that are captured in the first and second principal components. PCA loadings are z-scores, so to visualize them you take loading times standard deviation, then add that to the variable's mean value to put the loading back into the original measurement units. For instance, if principal component 1 has a loading of 0.5 on data variable x1, this means that if a vowel has a score of 1 for the first principal component then the predicted value of x1 for that vowel is the average tongue position plus 0.5 times the standard deviation associated with the first principal component.

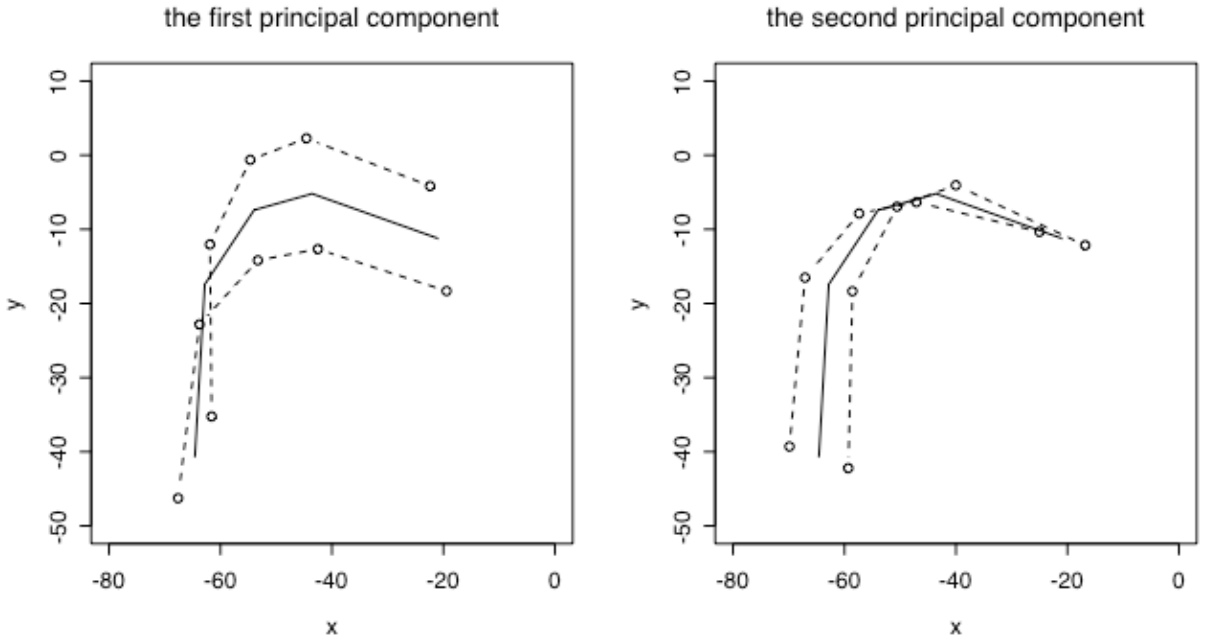

Figure 3.9. The first two principal components of variation in the tongue shape data of talker PL1.

The loadings for the first principal component (Table 3.1) are mainly on the y variables - this is a high/low component of tongue movement, while the second principal component loadings are mainly on the x variables - a front/back component.

Table 3.1 Loadings of the tongue x and y variables on the first two principal components of variation. PC1 is an abbrevation of principal component 1. Values near 0 are not listed in the table.

	x2	x6	x8	x10	x14	y2	y6	y8	y10	y14
PC1					.2	.47	.5	.45	.36	.37
PC2	.43	.37	.36	.44	.55		.12			-.15

We can visualize the principal components of tongue shape variation for this talker by graphing weighted deviations from the average tongue shape. The first two components are shown in figure 3.9. The principal components analysis of tongue shapes "decomposes" all tongue shapes into some combination of these two basic tongue shape components, one for the front/back location of the tongue and one for the high/low location of the tongue. This two component decomposition of tongue shape was not prespecified by me in setting up the analysis, but emerges from the data. To plot PC1 in that figure I took the average locations of the pellets and added the standard deviation of PC1 multiplied by the loading (written *w* in the formula below) to get one of the dotted lines and then subtracted the standard deviation times the loading to get the other dotted line. These lines show tongue shapes associated with the variation captured by PC1.

$x_{ji} = \bar{x}_i \pm s_j w_{ji}$ to visualize the variance of variable *i* encoded in PCj

---

R note. Computing the principal components analysis and plotting the results is actually pretty easy. We have some data in PL1 with *xy* locations of pellets on the tongue. I chose to use pellets 2, 6, 8, 10, and 14 for this analysis. Because the data set only has 11 vowels for this talker we can analyze only 11 or fewer variables.

```
> summary(pc <- princomp(~ x2+x6+x8+x10+x14+y2+y6+y8+y10+y14, data = PL1))

Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4      Comp.5
Standard deviation     15.3389055 9.6094120 4.13258604 2.48945962 1.230628457
Proportion of Variance  0.6664402 0.2615568 0.04837448 0.01755426 0.004289695
Cumulative Proportion   0.6664402 0.9279971 0.97637154 0.99392580 0.998215497
```

The summary produces an "Importance of components" printout. From this I used the standard deviations of components 1 and 2 in producing figure 3.9 (s_1 = 15.33, and s_2 = 9.6). Additionally the proportion of variance accounted for is reported. I plotted the cumulative proportion of

variance accounted for of the first 5 components in figure 3.8.

The object `pc` contains a number of other results. For example, the values returned by `pc$center` are the average values for each of the variables ($\bar{x}_i$ in the formula above). I plotted the average values in figure 3.7 and again as the solid lines in figure 3.9.

```
> pc$center
        x2         x6         x8        x10        x14         y2
-20.911655 -43.554792 -53.972281 -62.819160 -64.590483 -11.240615
        y6         y8        y10        y14
 -5.194050  -7.400377 -17.428613 -40.752770
```

The factor loadings (the w_{ji} in the formula above) are saved in `pc$center`. I reported these in table 3.1 and used them together with the averages and standard deviations to produce figure 3.9.

```
> pc$loadings

Loadings:
     Comp.1 Comp.2
x2           -0.427
x6           -0.369
x8           -0.355
x10          -0.443
x14  -0.201 -0.549
y2   -0.472
y6   -0.498 -0.119
y8   -0.452
y10  -0.360
y14  -0.368  0.152
```

In particular, the plot commands for the left side of figure 3.9 (of PC1) were:

```
> attach(pc)
> plot(center[1:5]+loadings[1:5,1]*15, center[6:10]+loadings[6:10,1]*15,
      ylim=c(-50,10),xlim=c(-80,0),xlab = "x", ylab = "y", type="b",lty=2)
> points(center[1:5]-loadings[1:5,1]*15, center[6:10]-loadings[6:10,1]*15,
      type="b",lty=2)
> lines(center[1:5],center[6:10])
```

To produce the right side of the figure (PC2) you replace the standard deviation 15 with the PC2 value 9.6, and refer to `loadings[,2]` instead of `loadings[,1]`.

Exercises

1. Pooled error variance. At one point I just declared that the pooled variance in the Cherokee VOT data is 1297.7. Show how this number was derived from the data. Why is this different from the overall variance of VOT in this data set (note: `sd(VOT)^2 = 1473.3`)?

2. Calculate the sign test contrasting male and female vowel first formant frequency from the data in "F1_data.txt".

3. Below is a matrix of correlations among three variables, *x1*, *x2*, and *y*. The standard deviations of these three variables are: $s_{x1} = 11.88$, $s_{x2} = 6.02$, and $s_y = 9.95$. Use these correlations and standard deviations to compute a regression formula to predict the value of *y* from *x1* and *x2*. Show your calculations. Use the regression equation to predict the value of *y* when *x1* is 30 and *x2* is 6.

```
          y         x1         x2
y  1.000000  0.7623830 -0.8292570
x1 0.762383  1.0000000 -0.3596465
x2 -0.829257 -0.3596465  1.0000000
```

4. Variations on the `t.test()` command. Read the "cherokeeVOT.txt" data with `read.delim()` and try these variations on the `t.test()` command illustrated in this chapter.

```
t.test(VOT[year=="1971"],VOT[year=="2001"], alternative="greater")
t.test(VOT~year, alternative="greater")
t.test(VOT~year)
```

What changes in the results? What do these changes suggest, as regards the hypothesis being tested and the conclusions that you can reach with the test? Hint: you can look at the help page `>help(t.test)` to see a list of the options taken by the `t.test()` function. Do the default values always produce the most conservative test?

5. Use the data file "regression.txt". Read this data into a data.frame called `reg`. Look at the results of `plot(reg)` and compare the graph there to the correlation matrix given by `cor(reg)`. Use lm() to fit three models y~x1, y~x2, y~x1+x2. Which do you prefer to account for variance in y? What is the regression equation? Use the regression equation to predict the value of *y* when *x1* is 30 and *x2* is 6.

6. Use the data file "chaindata.txt". Read this data into a data.frame called `chain`. Use `subset()` to extract the data for one of the other talkers in the data set (not PL1), and perform

a principal components analysis of the tongue shape for this talker. How many factors seem to be needed to model the variance in this talker's productions. See if you can produce graphs of the principal components like those in figure 3.9.

7. The F distribution. I said that it is common to put the larger of two variances as the numerator in an F ratio testing whether two variances are equal. Is this necessary or merely conventional? Use the function `shade.tails.df()`, which you will find on the course web page (don't forget to "source" the function) and find the F value and probability of a more extreme F for these values:

$s_a^2 = 300,\ s_b^2 = 1200,\ n_a = 15,\ n_b = 50$

Do this once with $F = s_a^2 / s_b^2$ and once with $F = s_b^2 / s_a^2$.