

NORMAL DISTRIBUTIONS AND RELATED TOPICS

Today we'll review topics related to the normal distribution and get started computing with R.

Readings. KNNL Appendix A (pp. 1297–1305) and Section 2.11 (pp. 78–80)

Getting started with R. R should already be available on most Department of Statistics computers. You can also install it for free on your own computer. To get started, create a project directory (i.e., a folder) where you want to store your data files. On my personal Windows computer, I created a directory for this course called

```
d:\jls\stat511\2007\lectures
```

Within that directory, I created a shortcut to the R executable program, which on my computer is

```
C:\Program Files\R\R-2.3.1\bin\Rgui.exe
```

Then I right-clicked on the shortcut and modified its properties so that it starts in the project directory indicated above. Double-clicking on that shortcut will begin a new R session.

If you want to use R from a Unix or Linux machine, you should first create a project directory (e.g. by using the `mkdir` command) and go into that directory (e.g. by using `cd`). Then you can begin an R session by typing `R`.

Univariate normal distribution. A random variable Y is said to have a Gaussian or normal distribution with mean $E(Y) = \mu$ and variance $V(Y) = \sigma^2$,

$$Y \sim N(\mu, \sigma^2),$$

if its probability density function (pdf) is

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(y-\mu)^2}{2\sigma^2}\right\}$$

over the real line $-\infty < y < \infty$. This is a “location and scale” family, which means that

$$a + bY \sim N(a + b\mu, b^2\sigma^2),$$

i.e. normality is preserved under linear (or affine) transformations.

If $Y \sim N(\mu, \sigma^2)$, then $Z = (Y - \mu)/\sigma$ is said to have a standard normal distribution,

$$Z \sim N(0, 1).$$

The standard normal has the pdf

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2},$$

and the cumulative distribution function (cdf)

$$\Phi(z) = P(Z \leq z) = \int_{-\infty}^z \phi(\eta) d\eta.$$

This integral must be evaluated numerically. If $\Phi(z) = p$ then $\Phi^{-1}(p) = z$, where Φ^{-1} is the inverse-cdf or quantile function. In the old days, we obtained values of Φ and Φ^{-1} from tables; now we typically use computers. In R, values of ϕ , Φ and Φ^{-1} can be obtained using the functions `dnorm`, `pnorm` and `qnorm`. Here is a part of the R help file on these and related functions, which you can see if you type `help(dnorm)`, `help(pnorm)` or `help(qnorm)`.

The Normal Distribution

Description:

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to 'mean' and standard deviation equal to 'sd'.

Usage:

```
dnorm(x, mean=0, sd=1, log = FALSE)
pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean=0, sd=1)
```

Arguments:

`x,q`: vector of quantiles.

`p`: vector of probabilities.

`n`: number of observations. If '`length(n) > 1`', the length is taken to be the number required.

`mean`: vector of means.

```

sd: vector of standard deviations.

log, log.p: logical; if TRUE, probabilities p are given as log(p).

lower.tail: logical; if TRUE (default), probabilities are P[X <= x],
otherwise, P[X > x].

```

Value:

'dnorm' gives the density, 'pnorm' gives the distribution function, 'qnorm' gives the quantile function, and 'rnorm' generates random deviates.

Here are some examples of how to use these functions.

```

> pnorm(-1.96) # area under standard normal curve to the left of -1.96
[1] 0.02499790

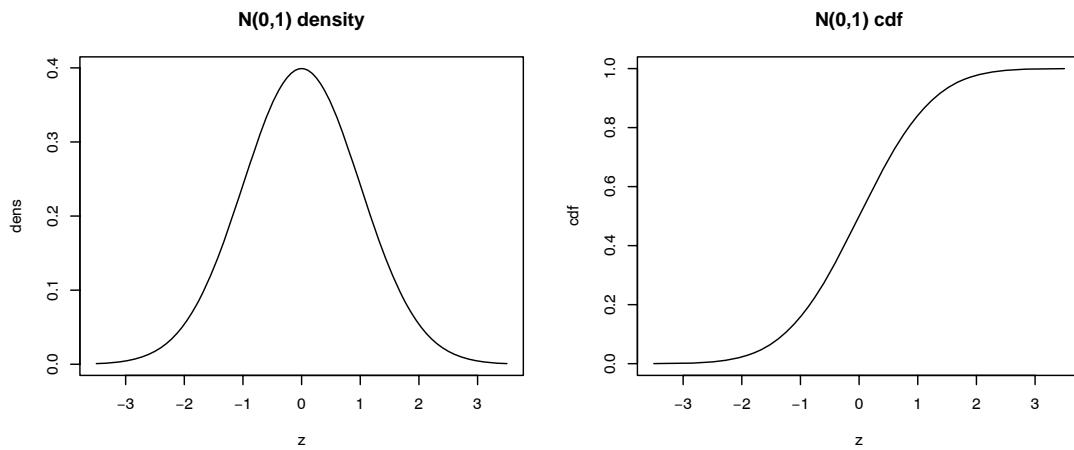
> pnorm(3) - pnorm(-3) # area between -3 and +3
[1] 0.9973002

> qnorm(.25) # 25th percentile
[1] -0.6744898

> z <- seq( from=-3.5, to=+3.5, by=.01) # grid of Z-values
> z
[1] -3.50 -3.49 -3.48 -3.47 -3.46 -3.45 -3.44 -3.43 -3.42 -3.41 -3.40 -3.39
[13] -3.38 -3.37 -3.36 -3.35 -3.34 -3.33 -3.32 -3.31 -3.30 -3.29 -3.28 -3.27
-- lines omitted --
[685] 3.34 3.35 3.36 3.37 3.38 3.39 3.40 3.41 3.42 3.43 3.44 3.45
[697] 3.46 3.47 3.48 3.49 3.50

> dens <- dnorm(z)      # density values
> cdf <- pnorm(z)      # cdf values
> plot( z, dens, type="l", main="N(0,1) density" ) # plot the density
> plot( z, cdf, type="l", main="N(0,1) cdf" )        # plot the cdf

```



Data objects in R. Data objects created during the R session are stored in volatile memory (RAM) for the duration of the session. You can see the objects in your session at any time by typing `objects()`.

```
> objects()
[1] "bp"    "cdf"   "dens"  "dias"  "sys"   "y"     "z"
```

You can delete objects by using the function `remove`.

```
> remove(bp)  # remove bp
> objects()
[1] "cdf"   "dens"  "dias"  "sys"   "y"     "z"

> remove( list=objects() )  # remove all objects in the session
> objects()                 # now no objects are left
character(0)
```

When you exit the R session by typing `q()`, you will be asked if you want to save the workspace image. If you say “No,” then all objects will be wiped out. If you say “Yes,” then they will be stored on the computer’s disk in a file called `.Rdata`. The next time you start an R session in that directory, all the objects stored in that file will be

read back in and will be available to you again.

R scripts. If you are doing anything more than a very simple analysis in R, you will probably want to save your R code in a script file. A script file is just an ASCII (text) file containing R commands.

To create a script file, go to the R menu and select **File -> New script**. This will open an R text editor Window. You can type commands in this window.

- To execute a single line from the editor window, move the cursor to that line and type Ctrl+R. This is equivalent to copying that line from the editor window (Ctrl+C) and pasting it into the R session (Ctrl+V).
- To execute multiple lines, use your mouse to highlight them in the editor window and type Ctrl+R.
- To save the commands in the editor window to a script file, select **File -> Save as....** By default, the name of the script file will have an **.R** extension.
- To execute all of the commands in a script file, go to the R command line and type **source("filename.R")**.

You may also choose to edit your R scripts in another text editor program. If you do, however, you cannot execute single or multiple lines using Ctrl+R. Rather, you will have to manually copy the lines of code from the editor

window and paste them into the R console window.

Chisquare distribution. If Z_1, Z_2, \dots, Z_n are independent and identically distributed (iid) $N(0, 1)$ random variables, then

$$Y = Z_1^2 + Z_2^2 + \cdots + Z_n^2$$

is said to have a chisquare distribution with n degrees of freedom, and we write $Y \sim \chi_n^2$.

The chisquare is a special case of the gamma distribution. Deriving the chisquare density is a straightforward exercise in probability theory, but we will never need to work with the χ^2 density in this course. Values of the pdf, cdf and inverse-cdf and random variates are available in R through the functions `dchisq`, `pchisq`, `qchisq` and `rchisq`.

The mean of the chisquare is its degrees of freedom, $E(Y) = n$, because

$$E(Y) = E(Z_1^2 + \cdots + Z_n^2) = nE(Z^2)$$

where $Z \sim N(0, 1)$, and $E(Z^2) = V(Z) + (E(Z))^2 = 1$. The variance of the chisquare is twice the degrees of freedom, $V(Y) = 2n$, because

$$V(Y) = V(Z_1^2 + \cdots + Z_n^2) = nV(Z^2),$$

and

$$V(Z^2) = E(Z^4) - (E(Z^2))^2 = E(Z^4) - 1,$$

and the fourth moment of the standard normal is $E(Z^4) = 3$. In an abuse of notation, we will write $E(\chi_n^2) = n$ and $V(\chi_n^2) = 2n$. (This is an abuse of notation because χ_n^2 is a distribution, not a random variable.)

The chisquare distribution is skewed to the right (positively skewed), especially for small degrees of freedom. As $n \rightarrow \infty$ it begins to look more symmetric and normal as a result of the Central Limit Theorem.

If we divide a chisquare by its degrees of freedom, we get a random variable that is sometimes called a mean square. If $Y \sim \chi_n^2$, then Y/n is said to be distributed as χ_n^2/n (another abuse of notation). It's easy to see that

$$E(\chi_n^2/n) = 1 \quad \text{and} \quad V(\chi_n^2/n) = 2/n.$$

We can think of χ_n^2/n as the average of n squared standard normals, $(Z_1^2 + \dots + Z_n^2)/n$. It follows from the law of large numbers that χ_n^2/n converges in probability to $E(Z^2) = 1$. As n becomes large, the density of χ_n^2/n approaches a spike at 1.

Student's t-distribution. Suppose that $Z \sim N(0, 1)$, $Y \sim \chi_n^2$ and the two are independent. If we let $X = Z/\sqrt{Y/n}$ then X is said to have a t-distribution with n degrees of freedom, $X \sim t_n$. In yet another abuse of notation, we can write

$$t_n = \frac{N(0, 1)}{\sqrt{\chi_n^2/n}}. \tag{1}$$

The t-distribution is symmetric and bell-shaped. It resembles the standard normal except that it is wider (greater variance) and has heavier tails. The k th moment, $E(X^k)$, does not exist if $k \geq n$. The mean $E(X)$ is zero when it exists (i.e. when $n > 1$). The variance, when it exists (i.e. when $n > 2$) is $n/(n - 2)$.

When $n \rightarrow \infty$, the mean square random variable in the denominator of (1) converges to 1, so $t_n \rightarrow N(0, 1)$. For our purposes, there is not much difference between t_n and $N(0, 1)$ if $n \geq 30$.

The special case of t_n with $n = 1$ degree of freedom is called a Cauchy distribution. The Cauchy has extremely heavy tails, so heavy that the mean does not exist. The distribution is still bell-shaped and symmetric about zero, so the median and mode are still zero even though the mean is undefined.

Here is R code that generates a plot that compares the pdf's for t-distributions with different degrees of freedom.

```
> x <- seq( from=-4.5, to=+4.5, by=.01) # grid of x-values
> zdens <- dnorm(x)
> tdens.1 <- dt(x, df=1)
> tdens.2 <- dt(x, df=2)
> tdens.4 <- dt(x, df=4)
> tdens.10 <- dt(x, df=10)

> # set up a blank plotting region
> plot( x=c(-4.5,4.5), y=c(0,.4), type="n",
+       main="Comparison of t densities", xlab="x", ylab="density", )
> # The function lines() adds lines to an existing plot.
```

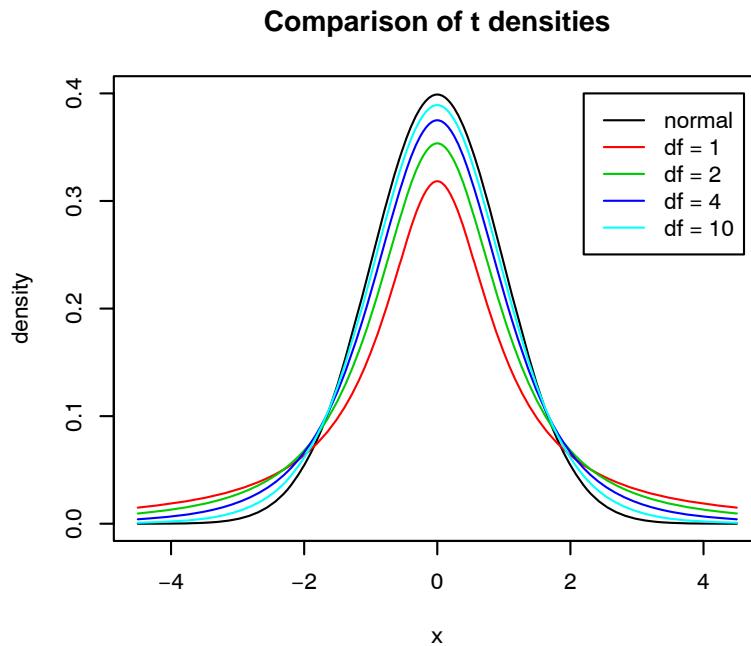
```

> # The arguments col= change the line colors. Or you could use lty=
> # instead to change the line types (solid, dashed, dotted, etc.)
> lines( x, zdens, col=1)
> lines( x, tdens.1, col=2)
> lines( x, tdens.2, col=3)
> lines( x, tdens.4, col=4)
> lines( x, tdens.10, col=5)

> # This function adds a legend to the plot.
> # The x and y arguments are the position of the legend, which
> # you need to choose by trial and error.
> # All the plotted lines are solid (lty=1), but they
> # have different colors (col=1,...,5)

> legend( x=2.5, y=.4,
+   lty = c(1,1,1,1,1),    col = c(1,2,3,4,5),
+   legend = c("normal", "df = 1", "df = 2", "df = 4", "df = 10"))

```



Fractional degrees of freedom. Notice that we have defined χ_n^2 and t_n in such a way that n must be a positive integer. If we derive the density functions based on these definitions, however, we find that they are still valid

densities for any positive real value of n . Fractional degrees of freedom do not arise much in the exact theory of linear models, but they are useful for statistical approximations. For example, in the test of equality of means from two independent samples (the two-sample t-test) that does not assume equal variances, the null behavior of the test statistic is often approximated by a t-distribution with fractional degrees of freedom (e.g. Satterthwaite's rule). The R functions for the t-distribution (`dt`, `pt`, `qt` and `rt`) can accept fractional degrees of freedom.

F-distribution. If $X_1 \sim \chi_m^2$, $X_2 \sim \chi_n^2$, and the two are independent, then

$$Y = \frac{X_1/m}{X_2/n}$$

is said to have an F-distribution with (m, n) degrees of freedom, and we write $Y \sim F_{m,n}$. We will refer to m and n as the numerator and denominator degrees of freedom, respectively.

We often encounter F distributions for which m is small and n is large. If $n \rightarrow \infty$ but m remains fixed, then the denominator converges to 1, and $F_{m,n} \rightarrow \chi_m^2/m$. For this reason, we can regard the F-distribution as an overdispersed version of the chisquare, just as we can regard the t-distribution as an overdispersed normal.

In the special case of $m = 1$, we can write

$$F_{1,n} = \frac{\chi_1^2}{\chi_n^2/n} = \frac{[N(0,1)]^2}{\left[\sqrt{\chi_n^2/n}\right]^2} = t_n^2,$$

so an F random variable with one degree of freedom in the numerator is the same as a squared t random variable.

Here is some R code that creates a table comparing the percentiles of χ_1^2 and $F_{1,n}$ for various values of n .

```
> # first create a matrix of missing values with
> # six rows and five columns
> result <- matrix( NA, 6, 5)
>
> # assign row and column names
> dimnames(result) <- list(
+   c("F(1,1)", "F(1,5)", "F(1,10)", "F(1,25)", "F(1,100)", "Chisquare(1)  "),
+   c("50%", "75%", "90%", "95%", "99%") )
>
> # fill the table with percentiles
> p <- c(.50, .75, .90, .95, .99 )
> result[1,] <- qf( p, df1=1, df2=1)
> result[2,] <- qf( p, df1=1, df2=5)
> result[3,] <- qf( p, df1=1, df2=10)
> result[4,] <- qf( p, df1=1, df2=25)
> result[5,] <- qf( p, df1=1, df2=100)
> result[6,] <- qchisq( p, df=1)
>
> # round off to two decimal places and print table
> result <- round( result, 2)
>
> result
      50%   75%   90%   95%   99%
F(1,1)     1.00  5.83 39.86 161.45 4052.18
F(1,5)     0.53  1.69  4.06   6.61   16.26
F(1,10)    0.49  1.49  3.29   4.96   10.04
F(1,25)    0.47  1.39  2.92   4.24   7.77
F(1,100)   0.46  1.34  2.76   3.94   6.90
Chisquare(1) 0.45  1.32  2.71   3.84   6.63
```

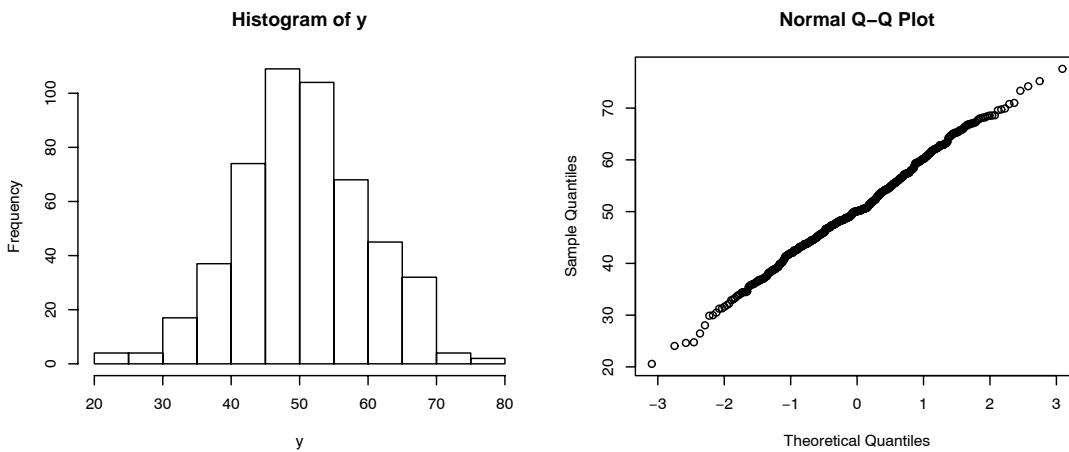
Next two lectures: Tests for normality, log transformations, Box-Cox and variance stabilizing transformations (KNNL Sections 3.4, 3.9 and 18.5).

ASSESSING NORMALITY AND TRANSFORMATIONS

Readings. Some material related to this lecture is found in KNNL Sections 3.4, 3.9 and 18.5.

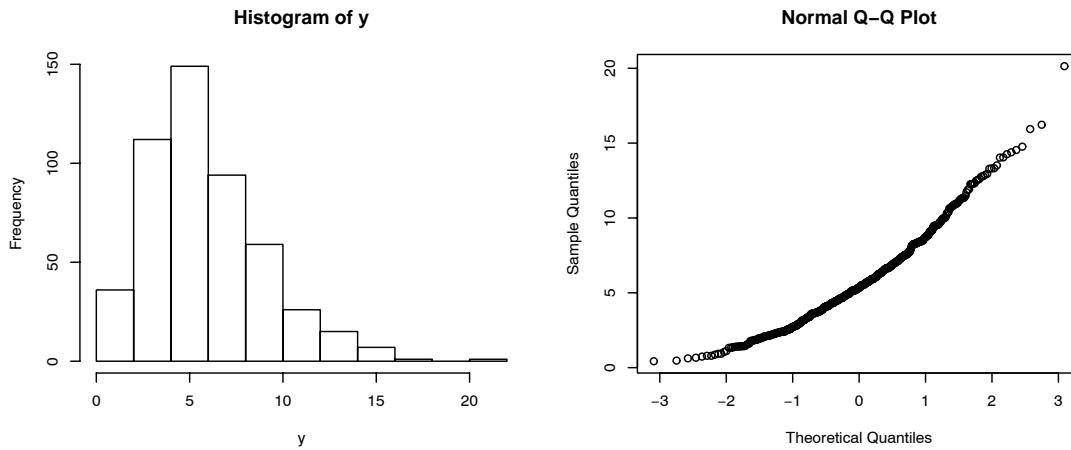
Evaluating normality. Given a sample y_1, \dots, y_n , we can evaluate their normality by a histogram or a normal probability plot.

```
> y <- rnorm(500, 50, 10) # generate 500 random variates with mean=50, sd=10
> hist(y)
> qqnorm(y)
```



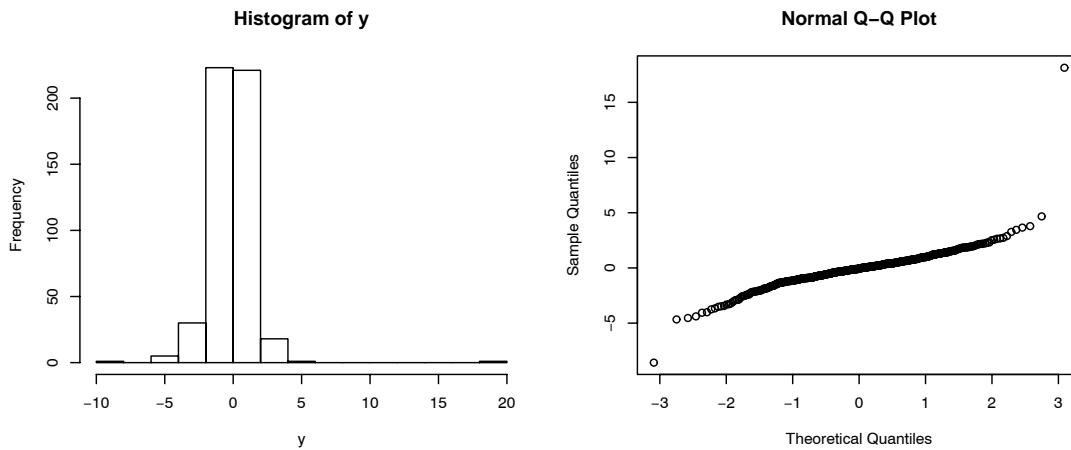
Example of right-skewed data:

```
> y <- rchisq(500, df=6) # chisquare with 6 degrees of freedom
> hist(y)
> qqnorm(y)
```



Data with heavier-than-normal tails:

```
> y <- rt(500, df=4) # Student's t with 4 degrees of freedom
> hist(y)
> qqnorm(y)
```



Now let's try this on some real data. The file `bp.dat` is an ASCII (plain text) file that contains blood pressure (systolic and diastolic) measurements on $n = 319$ subjects. The file looks like this.

```
BPSYS BPDIAS
147 60
121 78
121 74
-- lines omitted --
124 77
111 59
```

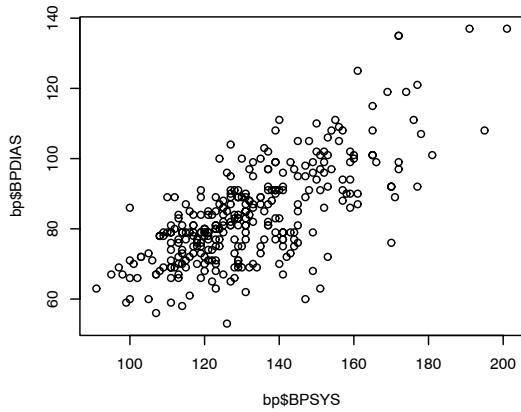
We'll read it into R using the function `read.table`. The result is a data frame. A data frame is a rectangular data set (essentially a data matrix) with rows corresponding to subjects and columns corresponding to variables.

```
> bp <- read.table( "bp.dat", header=T ) # read in data and create a data frame

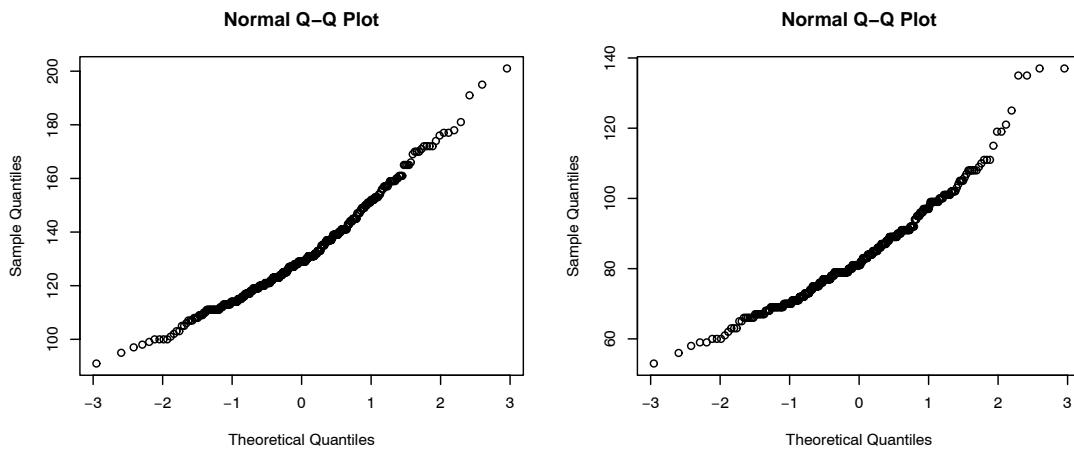
> names(bp) # display the variable names
[1] "BPSYS"  "BPDIAS"

> bp[1:10,] # display the first ten rows
   BPSYS BPDIAS
1     147      60
2     121      78
3     121      74
4     127      91
5     122      84
6     121      85
7     127      89
8     108      78
9     126      96
10    170      92

> plot( bp$BPSYS, bp$BPDIAS ) # the "$" refers to a variable in the data frame
```

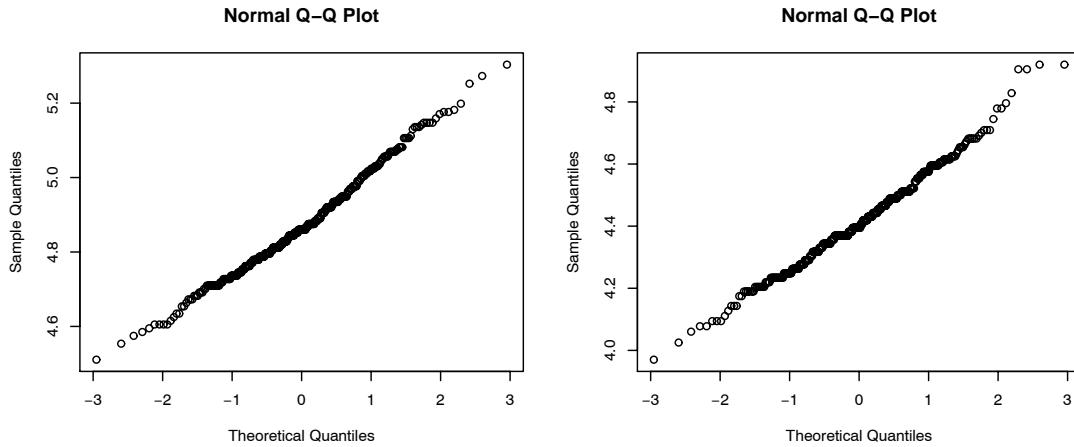


```
> qqnorm(bp$BPSYS)
> qqnorm(bp$BPDIAS)
```



Both of these measurements are skewed to the right. On the log scale, however, they appear to be nearly normally distributed.

```
> bp$LOGSYS <- log( bp$BPSYS )
> bp$LOGDIAS <- log( bp$BPDIAS )
> qqnorm( bp$LOGSYS )
> qqnorm( bp$LOGDIAS )
```



Tests for normality. There are many ways to test the null hypothesis that the population from which a sample is drawn is normally distributed. Tests have been developed based on the sample coefficient of skewness, the sample kurtosis, and combinations of the two. There are also many tests based on the correlation seen in the normal probability plot, i.e. the correlation between the actual sample quantiles and their theoretical values expected under a normal distribution. Perhaps best known example of the latter is the Wilk-Shapiro test. The test has been implemented in R in the function `shapiro.test()`.

The Wilk-Shapiro test is very powerful, and can detect even trivial departures from normality when the sample size is large. Let's apply it to the $n = 319$ measurements of systolic and diastolic blood pressure:

```
> shapiro.test( bp$BPSYS )
```

```
Shapiro-Wilk normality test
```

```
data: bp$BPSYS
W = 0.9672, p-value = 1.253e-06

> shapiro.test( bp$BPDIAS )

Shapiro-Wilk normality test

data: bp$BPDIAS
W = 0.9557, p-value = 3.137e-08
```

Both of the p-values are close to zero, so we can strongly reject the hypothesis of normality. This is to be expected, because the samples were obviously skewed. Now let's apply it to the logged versions:

```
> shapiro.test( bp$LOGSYS )

Shapiro-Wilk normality test

data: bp$LOGSYS
W = 0.9901, p-value = 0.03016

> shapiro.test( bp$LOGDIAS )

Shapiro-Wilk normality test

data: bp$LOGDIAS
W = 0.9896, p-value = 0.02328
```

We can still reject the null hypothesis at the $\alpha = .05$ level, even though the normal probability plots indicate that the transformed variables are nearly normal.

In practice, I do not find these tests useful for the following reasons.

- Naturally occurring data are never precisely normally distributed. We can always detect departures from normality if n is large enough.

- The important question is not whether the data are normal (they aren't) but whether they are close enough that it's ok to treat them as normal. And the answer to that question varies depending on the sample size and the procedures that will be applied.

Some normal-theory procedures (e.g., t-tests regarding means) are quite robust to departures from normality, especially if the sample is reasonably large. Other procedures (e.g., the F test for equality of variances) are so sensitive that small departures from normality—departures so small that they may never be detected by a formal hypothesis tests—can have disastrous effects.

If we are going to apply analytic procedures that assume the data are normally distributed (and we often will), then we need to have some idea of how these procedures are affected by non-normality.

Example: The effects of non-normality on confidence intervals about means and variances.

Suppose y_1, \dots, y_n is a random sample from a normal distribution with mean μ and variance σ^2 . Taking \bar{y} and S^2 to be the usual sample mean and variance, the exact 95% confidence interval for μ is

$$\left[\bar{y} - t_{.975, n-1} \frac{S}{\sqrt{n}}, \bar{y} + t_{.975, n-1} \frac{S}{\sqrt{n}} \right], \quad (1)$$

where $t(p, \nu)$ denotes the p th quantile of a t-distribution

with ν degrees of freedom. Moreover, an exact 95% confidence interval for σ^2 is

$$\left[\frac{(n-1)S^2}{\chi_{.975,n-1}^2}, \frac{(n-1)S^2}{\chi_{.025,n-1}^2} \right], \quad (2)$$

where $\chi_{p,\nu}^2$ denotes the p th quantile of χ_ν^2 . These intervals follows from the well-known results that

$$\frac{\bar{y} - \mu}{S/\sqrt{n}} \sim t_{n-1} \quad (3)$$

and

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi_{n-1}^2. \quad (4)$$

Both of these intervals are based on an assumption of normality. However, (1) works well for non-normal populations, especially when n is large. As $n \rightarrow \infty$, the distribution of \bar{y} approaches a normal because of the Central Limit Theorem. On the other hand, (2) is more sensitive to departures from normality; its performance depends on the forth moment (kurtosis) of the population.

Let's run a few simulations to see how these intervals perform with samples from normal and non-normal populations. This will also give us some more practice in programming with R. In the first simulation, we will draw samples of size $n = 319$ from a normal distribution with mean $\mu = 132$ and variance $\sigma^2 = 361$, which are the mean and variance of the 319 systolic blood pressure measurements in our sample:

```
> mean(bp$BPSYS)
[1] 132.0031
> var( bp$BPSYS )
[1] 361.1101
```

Let's draw a sample of $n = 319$ measurements from $N(132, 361)$, compute \bar{y} and S^2 from the sample, and see whether the confidence intervals given by (1) and (2) cover the true values of μ and σ^2 . We will repeat the whole process 10,000 times. Instead of saving the 10,000 samples, we will only save the 10,000 values of \bar{y} and S^2 and the logical indicators for whether each interval covered the true parameter. Here is some R code that will do it.

```
# define the parameters in the simulation
mu <- 132
sigma2 <- 361
n <- 319
nrep <- 10000

ybar <- numeric(nrep)    # create a vector to hold the ybar's
s2 <- numeric(nrep)      # another vector to hold the S2's
cover.mu     <- logical(nrep)  # logical vector
cover.sigma2 <- logical(nrep) # another one

set.seed(1432)  # set generator seed so that you can reproduce my results

for( i in 1:nrep ){

  # draw the sample
  y <- rnorm( n, mean=mu, sd=sqrt(sigma2) )

  # compute and store the sample mean and variance
  ybar[i] <- mean(y)
  s2[i] <- var(y)
  s <- sqrt(var(y))

  # see whether the interval for mu covers the true value
  cover.mu[i] <- abs( ybar[i] - mu ) <= qt(.975, n-1) * s/sqrt(n)}
```

```
# and whether the interval for sigma2 covers the true value
cover.sigma2[i] <- ( (n-1)*s2[i] / qchisq(.975,n-1) <= sigma2 ) &
( sigma2 <= (n-1)*s2[i] / qchisq(.025,n-1) ) }
```

This code has been stored in a file called `lec2sim1.R`. We can run it in the R session using the `source` command:

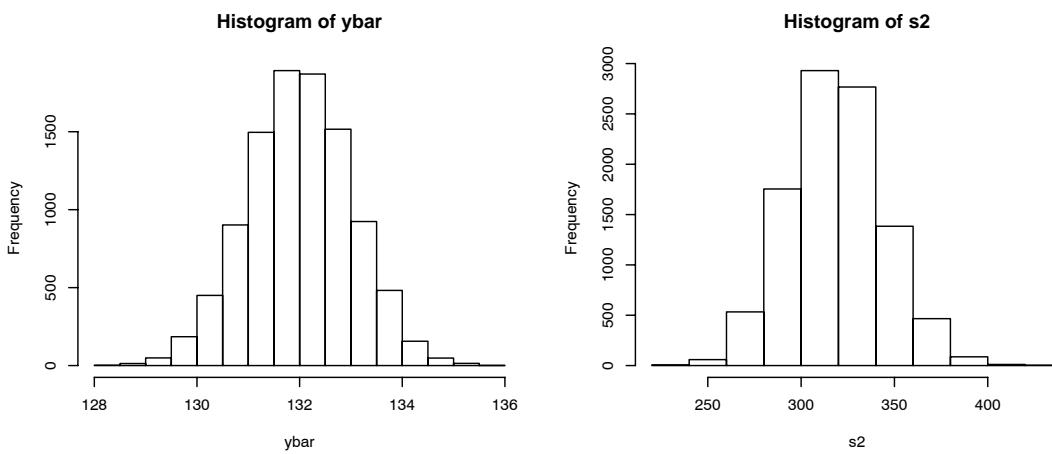
```
> source("lec2sim1.R")
```

Alternatively, you can open the file in the R script editor window, select everything (Ctrl+A) and run it (Ctrl+R). After running, you can inspect the 10,000 sample values of \bar{y} and S^2 :

```
> mean(ybar)
[1] 132.0019
> var(ybar)
[1] 1.152810

> mean(s2)
[1] 360.6145
> var(s2)
[1] 824.9867

> hist(ybar)
> hist(s2)
```



Notice that the average values of \bar{y} and S^2 over the 10,000 reps are very close to the true values of $\mu = 132$ and $\sigma^2 = 361$, as they should be, because both of these estimators are unbiased. How well did the intervals perform?

```
> table( cover.mu )
cover.mu
FALSE  TRUE
 518  9482

> table( cover.sigma2 )
cover.sigma2
FALSE  TRUE
 490  9510
```

94.8% of the simulated intervals for μ and 95.1% of the simulated intervals of σ^2 covered their respective true values. This is what we expected, because the samples are drawn from a normal population.

Now let's run a second simulation to see how the intervals in (1) and (2) perform when the data are drawn from a lognormal distribution. I chose a lognormal because the normal probability plots suggest that the natural log of the systolic blood pressure is approximately normally distributed. This should be a very realistic test of how reliable the procedures would be for data like these. If $X \sim N(\mu, \sigma^2)$, then $Y = e^X$ is said to have a lognormal distribution. The mean and variance of the lognormal are

$$\begin{aligned} E(Y) &= e^{\mu + \sigma^2/2}, \\ V(Y) &= (e^{\sigma^2} - 1) e^{2\mu + \sigma^2}. \end{aligned}$$

Back-solving for μ and σ^2 gives

$$\begin{aligned}\mu &= \log(E(Y)) - \frac{1}{2} \log\left(1 + \frac{V(X)}{(E(X))^2}\right), \\ \sigma^2 &= \log\left(1 + \frac{V(X)}{(E(X))^2}\right).\end{aligned}$$

(I got these facts from the Wikipedia article on the lognormal distribution.) Here is some R code which I've placed in a file called `lec2sim2.R`.

```
# define the parameters in the simulation
mu <- 132
sigma2 <- 361
n <- 319
nrep <- 10000

ybar <- numeric(nrep) # create a vector to hold the ybar's
s2 <- numeric(nrep) # another vector to hold the S2's
cover.mu <- logical(nrep) # logical vector
cover.sigma2 <- logical(nrep) # another one

# find corresponding mean and variance for the normal variates
mu.normal <- log(mu) - 0.5 * log(1 + sigma2/mu^2)
sigma2.normal <- log(1 + sigma2/mu^2)

set.seed(9876) # set generator seed so that you can reproduce my results

for( i in 1:nrep ){

  # draw the sample
  y <- exp( rnorm( n, mean=mu.normal, sd=sqrt(sigma2.normal) ) )

  # compute and store the sample mean and variance
  ybar[i] <- mean(y)
  s2[i] <- var(y)
  s <- sqrt(var(y))

  # see whether the interval for mu covers the true value
  cover.mu[i] <- (ybar[i] - 1.96 * s >= mu) & (ybar[i] + 1.96 * s <= mu)
  cover.sigma2[i] <- (s2[i] - 1.96 * s >= sigma2) & (s2[i] + 1.96 * s <= sigma2)
}
```

```

cover.mu[i] <- abs( ybar[i] - mu ) <= qt(.975, n-1) * s/sqrt(n)

# and whether the interval for sigma2 covers the true value
cover.sigma2[i] <- ( (n-1)*s2[i] / qchisq(.975,n-1) <= sigma2 ) &
( sigma2 <= (n-1)*s2[i] / qchisq(.025,n-1) )

```

Running this program, we find that the performance of the intervals is not bad:

```

> source("lec2sim2.R")
> table(cover.mu)
cover.mu
FALSE  TRUE
 507  9493
> table(cover.sigma2)
cover.sigma2
FALSE  TRUE
 693  9307

```

The simulated coverage of the intervals for μ is still very close to 95%, and the coverage for σ^2 has only deteriorated a tiny bit to 93%. In this case, I would say that the intervals for μ and σ^2 would be quite trustworthy if we applied them to the actual systolic blood pressure data.

For a third simulation, let's see what happens when we use a population with heavier tails. In Lecture 1, we learned that a t-distribution with ν degrees of freedom is $\nu/(\nu - 2)$. Therefore, if we want a random variate from a t-distribution with mean μ , variance σ^2 and degrees of freedom ν , we could draw $Z \sim t_\nu$ and take

$$Y = \mu + Z \sqrt{\sigma^2 \left(\frac{\nu - 2}{\nu} \right)}.$$

Code for a simulation with $\nu = 5$ has been placed in

another file called `lec2sim3.R`, which I will not show here. When we run it, this is the result:

```
> source("lec2sim3.R")
> table(cover.mu)
cover.mu
FALSE  TRUE
 479  9521
> table(cover.sigma2)
cover.sigma2
FALSE  TRUE
 1918  8082
```

The intervals for μ still work great, but the performance for σ^2 is terrible. Only 81% of the intervals cover the true value. This means for an $\alpha = .05$ -level hypothesis test, the actual Type 1 error rate is nearly four times as high as it should be (19% versus 5%). As a rule-of-thumb, we can say that if the coverage of a nominal 95% interval drops below 90%—which corresponds to a doubling of the Type 1 error rate—then the procedure is badly flawed.

What is my point? Many data analysts will assess normality with hypothesis tests. If normality is rejected at the .05 level, they might abandon normal-theory procedures and apply transformations, nonparametric methods, etc. Whether one can reject the null hypothesis of normality with a given sample, however, is usually irrelevant to the question of whether a normal model is appropriate. (With the diastolic blood pressures, we strongly rejected the null hypothesis of normality, but we found that normal-theory intervals for both the mean and variance seemed ok.) Rather than testing the null

hypothesis of normality—which is false anyway—we should focus on how departures from normality exhibited by our data are serious enough to adversely impact the procedures that we want to use.

Power transformations. Suppose we have a sample y_1, y_2, \dots, y_n from a distribution with a long right tail (positively skewed). This means that the largest observations are larger than one would expect under normality. We can “pull in” that tail with a decelerating transformation.

Some common decelerating transformations, in increasing order of strength, are

$$\sqrt{y}, \quad \log y, \quad -1/\sqrt{y}, \quad -1/y.$$

The negative signs on the last two preserve the original ordering of the observations. These are all examples of power transformations, which are of the form

$$g(y) = ay^\lambda + b$$

for some values of a , b and λ . (Although the log transformation is not strictly of this form, it can be regarded as the limit of a power transformation as $\lambda \rightarrow 0$, as we shall see.) If $\lambda < 1$ then the transformation is decelerating, reducing the skewness of positively skewed data. If $\lambda > 1$ then the transformation is accelerating, and will make the largest observations larger relative to the

rest. Negatively skewed data are rare, so accelerating transformations are not common in practice.

The log transformation. The most popular of these transformations is the log. By this we mean the natural (base- e) log, not the base-10 log. In R, the function `log()` computes natural logs by default.

The log of a non-positive number is not defined, so the log transformation should be applied only to variables whose values are strictly positive. Values of the original variable between zero and one become negative on the log scale. Therefore, the log transformation can be useful for predicting outcomes that are required to be positive. If we build a linear regression model for $\log y$, the predicted values for $\log y$ may be positive or negative. But when we exponentiate the fitted values to obtain predictions on the original y scale, those predictions are guaranteed to be positive.

Coefficients from a linear regression model for $\log y$ are easy to interpret. A standard linear regression model for $\log y$,

$$\log y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \epsilon \quad (5)$$

implies that

$$y = \gamma_0 \gamma_1^{x_1} \gamma_2^{x_2} \cdots \xi,$$

where $\gamma_j = \exp(\beta_j)$ and $\xi = \exp(\epsilon)$. An additive model on the log scale corresponds to a multiplicative model on the original scale. Increasing x_j by one unit, holding the other

x 's constant, will multiply the predicted value of y by γ_j . When interpreting a linear model to $\log y$, therefore, it is helpful to exponentiate the coefficients.

When a coefficient of a linear model for $\log y$ is close to zero, we can interpret the coefficient directly without exponentiating it, because $\exp(\beta) \approx 1 + \beta$ in the neighborhood of $\beta = 0$. For example, if $\beta = .05$ then $\exp(\beta) = 1.051$. A coefficient of $\beta_j = .05$ means that a one-unit increase in x_j corresponds to a 5% increase in the predicted value of y . If $\beta = -.05$ then $\exp(\beta) = 0.95$, so a coefficient of $\beta_j = -.05$ means that a one-unit increase in x_j corresponds to a 5% decrease in the predicted value of y . This approximation works very well when $|\beta| \leq 0.10$.

Finally, note that model (5) implies that the mean of $\log y$ is a linear function of the x_j 's. It is a linear model for **the mean of the log-response**. Alternatively, one could define $\mu = E(y)$ and assume

$$\log \mu = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots, \quad (6)$$

which is a linear model for **the log of the mean response**. These two models are not the same, because the log of the mean is not the same as the mean of the log. (In fact, $E(\log y) < \log E(y)$, as a result of Jensen's inequality.) Model (5) is called a linear regression model for $\log y$. But model (6) is called a loglinear model, and it belongs to a class of called “generalized linear models” (GLIM's). Loglinear models and GLIM's have become

increasingly popular in recent years. GLIM's keep the response variable on the original scale, whereas (5) redefines the response.

Prior to the 1980's, when linear regression was the only tool available, data analysts were taught to transform variables to make them fit into the framework that they knew. They were taught to change the data to accommodate the model. Nowadays, we have a greater variety of models and software available, so we can now change the model to accommodate the data. Generally speaking, I think that it's better to change the model to fit the data, rather than to change the data to fit the model. In the first part of Stat 511, as will focus on linear regression, we will often transform variables to make the model reasonable. By the end of this course, however, we will learn how to fit GLIM's and build reasonable models on the original scale.

MORE ABOUT TRANSFORMATIONS

Power transformations. Power transformations of a variable y are of the form

$$g_\lambda(y) = a + b y^\lambda$$

for some constants a , b and λ . From a standpoint of normality, the values of a and b don't matter. The crucial choice is λ . Power transformations are designed to correct skewness. If data are positively skewed, values of $\lambda < 1$ will tend to reduce the skewness. If the data are negatively skewed (which is rare), values of $\lambda > 1$ are appropriate.

In practice, analysts often choose λ by trial and error. First, we could try \sqrt{y} , which is mildly decelerating. If the data are still positively skewed, we could go to transformations that are successively stronger: $y^{1/3}$, $\log y$ (which can be regarded as taking $\lambda = 0$), $-1/y^{1/3}$, $-1/\sqrt{y}$, $-1/y$, $-1/y^2$, and so on.

These transformations work properly only when y is positive. If the variable to be transformed contains 0's or negative values, data analysts will sometimes replace y by $y + c$, where c is a positive constant. This can be

problematic, however, because the choice of c is somewhat arbitrary, but it affects the results. The distribution of $g_\lambda(y + c_1)$ looks different from the distribution of $g_\lambda(y + c_2)$ if $c_1 \neq c_2$.

Box-Cox transformations. In their pioneering article on transformations, Box and Cox (1964) defined a power-transformation family

$$g_\lambda(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{for } \lambda \neq 0, \\ \log y & \text{for } \lambda = 0. \end{cases}$$

This family is continuous in λ . Moreover, $g_\lambda(y)$ is monotonically increasing for every λ , so that the original order of the observations is preserved. That is, if $y_1 > y_2$ then $g_\lambda(y_1) > g_\lambda(y_2)$.

Box and Cox (1964) show that λ can be estimated by maximum likelihood (ML). Suppose that the transformed sample, $g_\lambda(y_i)$, $i = 1, \dots, n$, is distributed as $N(\mu, \sigma^2)$ for some λ . This is a parametric family with unknown parameters $\theta = (\mu, \sigma^2, \lambda)$. The principles of ML suggest that we find the θ for which the loglikelihood function $l(\mu, \sigma^2, \lambda)$ is highest. The three score equations

$$\frac{\partial}{\partial \mu} l(\mu, \sigma^2, \lambda) = 0, \tag{1}$$

$$\frac{\partial}{\partial \sigma^2} l(\mu, \sigma^2, \lambda) = 0, \tag{2}$$

$$\frac{\partial}{\partial \lambda} l(\mu, \sigma^2, \lambda) = 0, \quad (3)$$

are difficult to solve simultaneously. It is easy, however, to solve the first two equations (1)–(2) for any fixed value of λ .

For a fixed value λ , let us denote the values of μ and σ that maximize l by $\mu(\lambda)$ and $\sigma^2(\lambda)$, respectively. Plugging the closed-form expressions for $\mu(\lambda)$ and $\sigma^2(\lambda)$ into the loglikelihood gives a new function

$$l^*(\lambda) = l(\mu(\lambda), \sigma^2(\lambda), \lambda) \quad (4)$$

which is a function of the single parameter λ . This new function l^* is called the **profile loglikelihood**. If we are able to find the value $\hat{\lambda}$ that maximizes l^* , then the maximizer of l is

$$\hat{\theta} = (\mu(\hat{\lambda}), \sigma^2(\hat{\lambda}), \hat{\lambda}).$$

Box and Cox (1964) show that the profile loglikelihood for this problem, except for irrelevant constant terms, is given by

$$l^*(\lambda) = -\frac{n}{2} \log \left\{ \sum_i \left(y_i^{(\lambda)} - \bar{y}_i^{(\lambda)} \right)^2 \right\} + (\lambda - 1) \sum_i \log y_i,$$

where $y_i^{(\lambda)}$ is the transformed observation

$$y_i^{(\lambda)} = g_\lambda(y_i) = \begin{cases} \frac{y_i^\lambda - 1}{\lambda} & \text{for } \lambda \neq 0, \\ \log y_i & \text{for } \lambda = 0, \end{cases}$$

and

$$y_i^{(\bar{\lambda})} = \frac{1}{n} \sum_i y_i^{(\lambda)}$$

is the average of the transformed observations.

This profile loglikelihood is typically maximized by a grid search. That is, we specify a grid of λ values, compute $l^*(\lambda)$ for each λ in the grid, and take the λ for which $l^*(\lambda)$ is highest as the approximate value of $\hat{\lambda}$. In R, it is not difficult to write code that will do this. But we don't have to, because someone else has already done it. There is a function called `boxcox` in the library called MASS, which was written by Venables and Ripley. (A library is a package of functions that extends the basic R language. It's easy for statisticians to create their own R libraries.) MASS is an acronym for *Modern Applied Statistics with S-PLUS*, the title of the book by Venables and Ripley.

To use the functions in MASS, you first have to attach the library. Do this by issuing the

```
> library(MASS)
```

in the R session. Once you do this, you have access to the functions in MASS. To see what these functions are, type:

```
> library(help=MASS)
```

To get information about `boxcox`, type:

```
> help(boxcox)
```

The main arguments to `boxcox` are

- a formula for a regression model involving one or more variables,
- a data frame in which the variables reside, and
- a vector of λ values over which to perform the grid search.

A formula is something like this,

$$Y \sim X_1 + X_2 + X_3$$

where Y is the response variable, and X_1 , X_2 and X_3 are predictors that influence the mean of the response. The symbol \sim means “is modeled as.” In the future, when we perform regression analyses in R, we will include predictors in our formulas, and the mean parameter μ will be replaced by a vector of coefficients β . For now, we will use the formula

$$Y \sim 1$$

which specifies a regression with no predictors (i.e. a model with an intercept only).

Unless your data are very highly skewed, the ML estimate of λ will usually lie somewhere in the interval $(-2, 1)$. Therefore, we will use as our grid the sequence of values

$$-2.00, -1.99, -1.98, \dots, 0.98, 0.99, 1.00,$$

which in R can be created by the expression

```
seq( from=-2, to=1, by=.01).
```

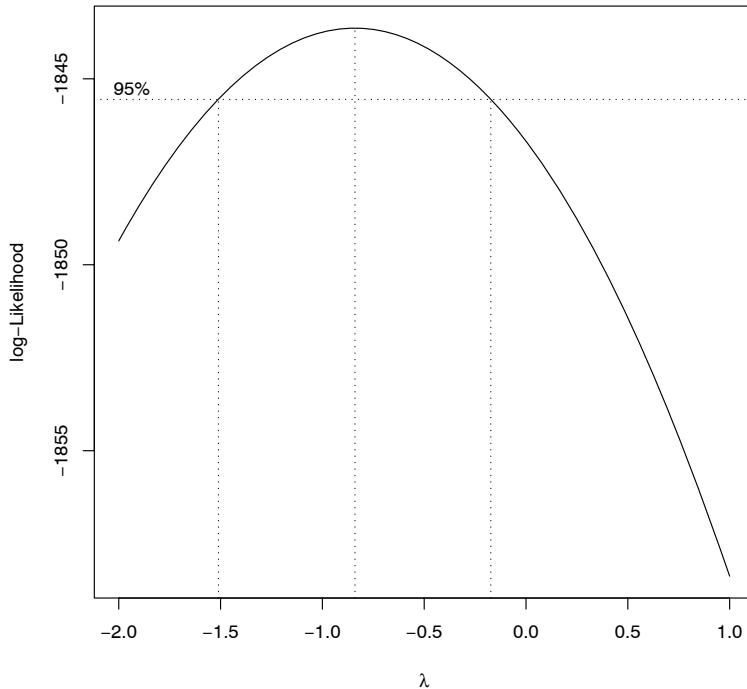
Let's apply the procedure to our systolic blood pressure measurements that we first examined in Lecture 2:

```
# attach the MASS library
library(MASS)

# read the data file
bp <- read.table( "bp.dat", header=T)

# apply the boxcox procedure and save the result
result <- boxcox( BPSYS ~ 1, data=bp, lambda=seq(from=-2,to=1, by=.01 ))
```

When you do this, the following plot appears.



This indicates that the ML estimate of λ is somewhere around -0.8 , and the limits of an approximate 95% confidence interval are approximately -1.5 and -0.2 . The results from the `boxcox` procedure, which we have stored in

an object called **result**, is a data frame with two variables: **x**, which contains the grid of λ values, and **y**, which contains the corresponding values of l^* . You can find the value of λ in the grid for which l^* is highest like this:

```
> result$x[ result$y==max(result$y) ]  
[1] -0.84
```

This tells us that the power transformation that makes our sample of systolic blood pressures most nearly normal is $-1/y^{0.84}$, which is quite a bit stronger than the log transformation.

Does this mean that we should transform our systolic blood pressures as $-1/y^{0.84}$ before modeling them? No! Box and Cox (1964) advocated the ML procedure only as a diagnostic tool for suggesting a range of transformations that could be useful. Changing the scale of the data by any power transformation will change many aspects of how to interpret the results of an analysis (e.g., the meaning of regression coefficients). Another analyst with another sample of systolic blood pressure measurements would have obtained a value of λ other than -0.84 . If everyone analyzes this variable on a different scale, it is difficult to compare results.

In general, it's good to use simple transformations like $\lambda = 1/2$ (the square root), $\lambda = 0$ (the natural log), and $\lambda = -1$ (the reciprocal) because these are intuitive and easier to understand than values like $\lambda = -0.84$ which are determined by a single dataset. If I were to transform

these blood pressure measurements, I would probably use the natural log ($\lambda = 0$), even though $\lambda = 0$ lies outside the 95% confidence interval, for reasons explained in Lecture 2. The fact that the null hypothesis $H_0 : \lambda = 0$ can be rejected does not mean that this transformation is unacceptable. As the normal probability plot shows, the log-transformed measurements are very nearly normally distributed, and so the log is good enough for most purposes. For many purposes, even the un-transformed raw measurements are good enough.

Variance-stabilizing transformations. The power transformations that we have discussed are purely empirical, suggested by the data themselves. A different kind of transformation, called a variance-stabilizing transformation, is suggested by theoretical considerations.

One feature of the normal model is that the mean μ and the variance σ^2 are distinct parameters. We can choose any value of μ on the real line and any positive value of σ^2 and stay within the distributional family. For some types of data, however, the mean and variance are functionally linked. Suppose, for example, that the responses are p_1, \dots, p_n where which p_i is a sample proportion. That is, p_i is the proportion of “successes” observed in n_i repeated trials with success probability π_i . In that case, the mean of p_i is π_i , and the variance of p_i is $\pi_i(1 - \pi_i)/n_i$. A plot of residuals versus fitted values from the linear regression

of p_i on one or more predictors will probably reveal heteroscedasticity. The variance of the residuals will be greatest when the fitted values are near 0.5, and the variance will be smallest for fitted values near 0 and 1.

More generally, suppose we have a response variable y for which theoretical considerations lead us to believe that the variance $V(y)$ is a function of the mean. That is, suppose that the variance is

$$V(y) \propto v(\mu),$$

where $\mu = E(y)$ and v is the so-called “variance function.” Then it may be possible to find a transformation $g(y)$ for which $V(g(y))$ is approximately constant. If so, the function g is called a variance-stabilizing transformation, and a linear regression analysis of the transformed response $g(y)$ may be more reasonable than a linear regression model for y itself.

Variance-stabilizing transformations are found by the following argument. If $g(y)$ is a smooth function of y , then it can be approximated by a Taylor expansion about μ ,

$$g(y) = g(\mu) + (y - \mu) g'(\mu) + \frac{(y - \mu)^2}{2} g''(\mu) + \dots,$$

where g' , g'' , ... are the derivatives of g . If the curvature of $g(y)$ in the vicinity of $y = \mu$ is not large relative to the standard deviation of y , then in the region where y lies with high probability, the transformation is approximately

linear,

$$g(y) \approx g(\mu) + (y - \mu) g'(\mu).$$

Taking variances of both sides gives the approximation

$$V(g(y)) \propto [g'(\mu)]^2 v(\mu),$$

where v is the variance function. For $V(g(y))$ to be approximately constant with respect to μ , therefore, we need to choose g such that

$$g'(\mu) \propto \frac{1}{\sqrt{v(\mu)}}.$$

Integrating both sides gives the variance-stabilizing transformation

$$g(\mu) \propto \int \frac{1}{\sqrt{v(\mu)}} d\mu.$$

Let's apply this argument to the binomial proportion. If y is a binomial proportion with mean π , then $V(y) \propto \pi(1 - \pi)$. The integral is

$$\int \frac{1}{\sqrt{\mu(1 - \mu)}} d\mu = 2 \sin^{-1} \sqrt{\mu},$$

where \sin^{-1} denotes the arcsine. Therefore, the variance-stabilizing transformation for a binomial proportion is the arcsine square root. In the old days, people used to analyze proportions by fitting normal linear regression models to the arcsine square root scale. When they did so, the residual plots would look fine, but the

regression coefficients would be very difficult to interpret.

Nowadays, it is much more common to apply logistic regression, which is based on the binomial likelihood.

Another frequently encountered situation is where the response is a frequency or count, i.e. a number of occurrences in a span of time or region of space. If y is a count, it is natural to suppose that it may arise from a Poisson process, and therefore it would have a Poisson distribution. If y is distributed as Poisson with mean μ , then $V(y) = \mu$, so the variance is proportional to the mean. The variance function is $v(\mu) = \mu$, and the variance-stabilizing transformation is

$$g(\mu) = \int \frac{1}{\sqrt{\mu}} d\mu$$

which gives $g(\mu) = \sqrt{\mu}$. Therefore, the variance-stabilizing transformation for a Poisson count is the square root. In the old days, people used to analyze count data by linear regression on the square-root scale. Nowadays, analysts are much more likely to apply techniques of loglinear modeling, which are based on the Poisson likelihood.

Finally, note that a variance-stabilizing transformation is designed to make the variance approximately independent of the mean. It is an attempt to get rid of heteroscedasticity. The purpose of a power transformation, on the other hand, is intended to reduce skewness. The two criteria are not the same. A transformation to reduce

skewness usually tends to reduce heteroscedasticity, and vice versa. But it is not necessarily so. Symmetry and homoscedasticity are two different properties of the normal distribution. There is no reason why the optimal transformation to correct departures from normality in one direction will correct departures in other directions.

COVARIANCE, CORRELATION AND NORMALITY

Covariance and correlation. Relationships between numeric variables are often described in terms of covariance and correlation. Suppose X and Y are random variables with means $E(X) = \mu_X$ and $E(Y) = \mu_Y$, respectively. The covariance between them is defined as

$$\text{Cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)].$$

The covariance between X and Y is often denoted by the symbol σ_{XY} .

Because $\text{Cov}(X, X) = V(X)$, covariance is a generalization of variance. The well known properties of variance,

$$\begin{aligned} V(X) &= E(X^2) - \mu_X^2, \\ V(a + bX) &= b^2 V(X), \end{aligned}$$

extend to covariance as

$$\begin{aligned} \text{Cov}(X, Y) &= E(XY) - \mu_X \mu_Y, \\ \text{Cov}(a + bX, c + dY) &= bd \text{Cov}(X, Y) \end{aligned}$$

(here, a , b , c and d denote constants). If X and Y are independent, then $\text{Cov}(X, Y) = 0$, but zero covariance

does not imply independence. Other useful properties of covariance include $\text{Cov}(c, X) = 0$ for any constant c , and

$$\begin{aligned}\text{Cov}(X + Y, W + Z) &= \text{Cov}(X, W) + \text{Cov}(X, Z) \\ &\quad + \text{Cov}(Y, W) + \text{Cov}(Y, Z).\end{aligned}$$

A large positive covariance indicates that large values of X and large values of Y tend to occur together. A large negative covariance indicates that large values of X are associated with small values of Y , and vice-versa. The problem with covariance is that its numerical value is difficult to interpret. The units of covariance—which are the units of X times units of Y —are not easy to understand. To aid understanding, we usually transform the covariance to make it scale-free. The correlation coefficient or correlation is

$$\text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sqrt{V(X) V(Y)}}.$$

The correlation between X and Y is sometimes denoted by ρ_{XY} or simply ρ .

Linear transforming random variables does not change their correlation,

$$\text{Corr}(a + bX, c + dY) = \text{Corr}(X, Y).$$

The correlation must lie between -1 and 1 . The extreme

values arise when Y is a linear transformation of X ,

$$\text{Corr}(X, a + bX) = \frac{b \text{Cov}(X, X)}{\sqrt{V(X) b^2 V(X)}} = \frac{b}{|b|} = \pm 1.$$

Much of our intuition about correlation is based on properties of the bivariate normal distribution. Before saying more about the correlation, let's briefly review random vectors and the multivariate normal distribution.

Random vectors. Let $Y = (Y_1, Y_2, \dots, Y_p)^T$ be a $p \times 1$ column vector of random variables, i.e. a random vector. The mean or expectation of Y is defined as the vector of expectations,

$$\begin{aligned} E(Y) &= E(Y_1, Y_2, \dots, Y_p)^T \\ &= (E(Y_1), E(Y_2), \dots, E(Y_p))^T \\ &= (\mu_1, \mu_2, \dots, \mu_p)^T. \end{aligned}$$

The covariance matrix of Y is

$$V(Y) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{bmatrix},$$

where $\sigma_{jk} = \text{Cov}(Y_j, Y_k)$. The variances σ_{jj} are sometimes written as σ_j^2 , and the standard deviations are sometimes written as $\sigma_j = \sqrt{\sigma_{jj}}$.

All of the well-known properties of means and variances of random variables generalize to random vectors. Suppose that X and Y are $p \times 1$ random vectors with means μ_X and μ_Y and covariance matrices Σ_{XX} and Σ_{YY} , respectively. Then:

1. $E(X + Y) = \mu_X + \mu_Y$.
2. $V(X + Y) = \Sigma_{XX} + \Sigma_{YY}$ if every element of X is uncorrelated with every element of Y .
3. $E(a + BX) = a + B\mu_X$, where a is a $q \times 1$ vector and B is a $q \times p$ matrix of constants.
4. $V(a + BX) = B\Sigma_{XX}B^T$, where B is a $q \times p$ matrix of constants.

A covariance matrix must be symmetric and positive definite. A matrix Σ is positive definite ($\Sigma > 0$) if

$$a^T \Sigma a > 0$$

for every nonzero vector a . This is intuitively sensible, because $a^T \Sigma a$ is the variance of the random variable $a^T X$, and variances are positive. Symmetric, positive definite matrices have many other well known properties, including:

- Every element on the main diagonal is positive.
- Every eigenvalue is positive.
- The matrix is non-singular.

- Its determinant is positive.
- Its inverse is also positive definite.

Multivariate normal distribution. Introductory treatments of the multivariate normal distribution can be found in many texts on multivariate statistical analysis. A random vector Y is said to have a p -variate normal distribution with mean μ and covariance matrix Σ if its density is

$$f(y) = |2\pi\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2}(y - \mu)^T \Sigma^{-1} (y - \mu) \right\}.$$

In this case, we write $Y \sim N_p(\mu, \Sigma)$ or $Y \sim N(\mu, \Sigma)$. When $p = 1$, this reduces to the univariate normal distribution. The free parameters are p means, p variances and $p(p - 1)/2$ covariances. In the bivariate ($p = 2$) case, there are five parameters,

$$\mu_1, \mu_2, \sigma_{11}, \sigma_{22}, \sigma_{12}.$$

Sometimes we express the parameters as

$$\mu_1, \mu_2, \sigma_{11}, \sigma_{22}, \rho$$

where $\rho = \sigma_{12}/\sqrt{\sigma_{11}\sigma_{22}}$ is the correlation coefficient.

Here are some basic properties of the multivariate normal distribution.

1. If $Y = (Y_1, \dots, Y_p)^T \sim N(\mu, \Sigma)$, then the elements of Y are univariate normal, $Y_j \sim N(\mu_j, \sigma_{jj})$,

$$j = 1, \dots, p.$$

2. More generally, subvectors of a multivariate normal vector are multivariate normal. Suppose we partition a $p \times 1$ multivariate normal random vector $Y \sim N_p(\mu, \Sigma)$ into subvectors

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix},$$

where $\dim(Y_1) = p_1$, $\dim(Y_2) = p_2$ and $p_1 + p_2 = p$. And suppose we partition μ and Σ is a similar way,

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \text{and} \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix},$$

where $\dim(\mu_1) = p_1 \times 1$, $\dim(\mu_2) = p_2 \times 1$, $\dim(\Sigma_{11}) = p_1 \times p_1$, $\dim(\Sigma_{12}) = p_1 \times p_2$, $\dim(\Sigma_{21}) = p_2 \times p_1$, and $\dim(\Sigma_{22}) = p_2 \times p_2$. Then

$$Y_1 \sim N(\mu_1, \Sigma_{11}),$$

$$Y_2 \sim N(\mu_2, \Sigma_{22}),$$

3. Under the same conditions as 2, the conditional distribution of Y_2 given Y_1 is also multivariate normal,

$$Y_2 | Y_1 \sim N(\mu_{2 \cdot 1}, \Sigma_{22 \cdot 1}),$$

where $\mu_{2 \cdot 1} = E(Y_2 | Y_1)$ and $\Sigma_{22 \cdot 1} = V(Y_2 | Y_1)$ are

$$\mu_{2 \cdot 1} = \mu_2 + \Sigma_{21} \Sigma_{11}^{-1} (Y_1 - \mu_1),$$

$$\Sigma_{22.1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}.$$

4. Linear transformations of normal vectors are normal.

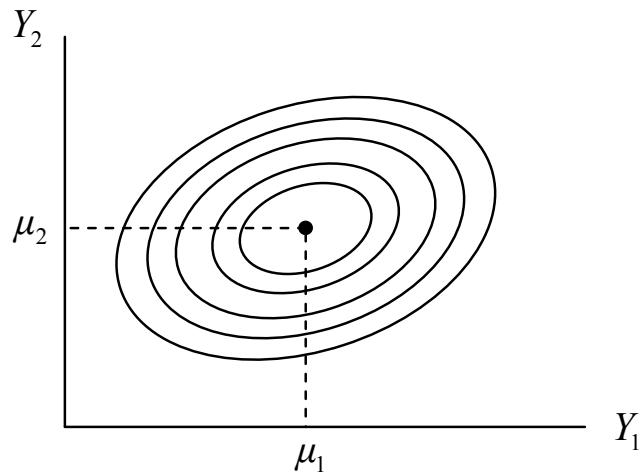
If $Y \sim N(\mu, \Sigma)$ then $a + BY \sim N(a + B\mu, B\Sigma B^T)$ for constants a and B with appropriate dimensions.

5. If $Y \sim N_p(\mu, \Sigma)$ then

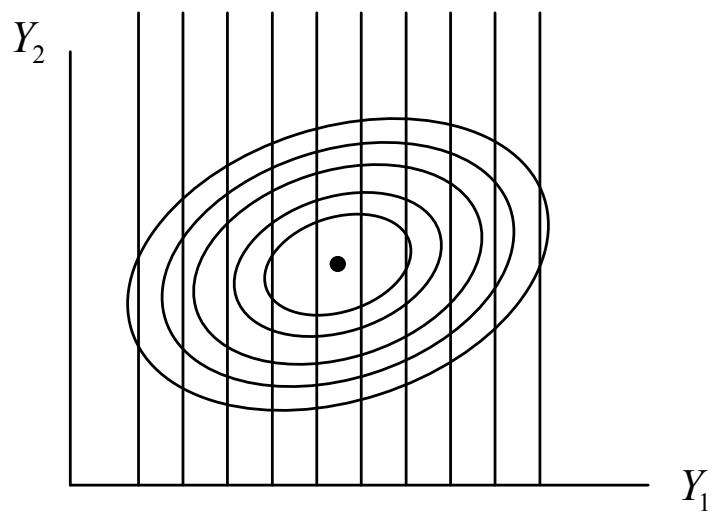
$$(Y - \mu)^T \Sigma^{-1} (Y - \mu) \sim \chi_p^2.$$

The bivariate normal distribution and correlation.

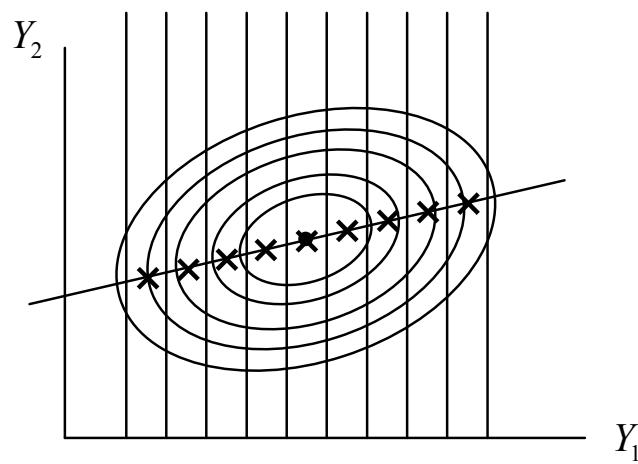
The contours of the bivariate normal distribution for $Y = (Y_1, Y_2)^T$ are concentric ellipses in the Y_1 - Y_2 plane, centered at the point $(\mu_1, \mu_2)^T$.



Suppose that we partition the plane into narrow vertical strips:



If the strips are very narrow, then the values of Y_1 within a strip are essentially constant. Within every strip of constant Y_1 , the values of Y_2 are normally distributed. For each constant value of Y_1 , suppose we compute and plot the average value of Y_2 . And then suppose we connect the points, like this.



The curve (in this case, the line) that goes through these points is $E(Y_2|Y_1)$ is called the regression function. If the distribution is bivariate normal, then the regression function is indeed a straight line,

$$E(Y_2|Y_1) = \mu_2 + \frac{\sigma_{12}}{\sigma_{11}} (Y_1 - \mu_1).$$

The variance of Y_2 within the vertical strips is

$$V(Y_2|Y_1) = \sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}}.$$

These formulas for the conditional mean and variance of Y_2 given Y_1 help us to interpret the correlation coefficient. Suppose that we express Y_1 and Y_2 in standard units, converting them to normal random variables with mean zero and variance 1,

$$\begin{aligned} Z_1 &= (Y_1 - \mu_1)/\sqrt{\sigma_{11}}, \\ Z_2 &= (Y_2 - \mu_2)/\sqrt{\sigma_{22}}. \end{aligned}$$

The correlation between Z_1 and Z_2 , which we call ρ , is the same as the correlation between the original variables, and thus

$$\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right).$$

The new variables Z_1 and Z_2 are scale-free. For $j = 1, 2$, Z_j measures the number of standard deviations that Y_j lies above its mean. The regression of Z_2 on Z_1 has a

particularly simple form,

$$E(Z_2 | Z_1) = 0 + \frac{\rho}{1}(Z_1 - 0) = \rho Z_1.$$

The regression of Z_2 on Z_1 is a straight line through the origin with slope ρ . This gives us one useful interpretation of the correlation coefficient: **The correlation is the slope of the regression line when both variables are expressed in standard units.**

The regression $E(Y_2 | Y_1)$ is a prediction equation. It gives us the “best” estimate of Y_2 if Y_1 becomes known. If Y_1 is unknown, our best guess as to what the value of Y_2 will be is just μ_2 . But if we are given a specific value for Y_1 , our best guess as to what the value of Y_2 will be is $E(Y_2 | Y_1)$. The correlation coefficient tells us how to make this prediction, if the distribution is bivariate normal. If Y_1 is known to be c standard deviations above its mean, then we would predict Y_2 to be $\rho \times c$ standard deviations above its mean. In terms of standard units, the predicted value of Y_2 is closer to its mean than Y_1 is to its mean. (This is called “regression to the mean.”) If $\rho = 0$, then knowledge of Y_1 gives us no additional information about Y_2 , and the best prediction will always be μ_2 .

Notice that $E(Z_2 | Z_1) = \rho Z_1$ but $E(Z_1 | Z_2) = \rho Z_2$. These two lines are not the same. If we plot them on the same graph, with Z_1 on the horizontal axis and Z_2 on the vertical axis, then both lines will pass through the origin. But the first line will have slope ρ , whereas the second line

will have slope $1/\rho$. The two lines will coincide only if $\rho = \pm 1$. The difference arises because $E(Y_2 | Y_1)$ is averaging within vertical strips, whereas $E(Y_1 | Y_2)$ is averaging within horizontal strips.

If we use the regression relationship $E(Y_2 | Y_1)$ to predict Y_2 from Y_1 , then we can also say something about the likely size of the prediction error. The variance of the prediction error is

$$\begin{aligned} V(Y_2 | Y_1) &= \sigma_{22} - \sigma_{12}^2 / \sigma_{11} \\ &= \sigma_{22} \left(1 - \frac{\sigma_{12}^2}{\sigma_{11} \sigma_{22}} \right) \\ &= \sigma_{22}(1 - \rho^2). \end{aligned}$$

Because $\rho^2 \leq 1$, it follows that

$$V(Y_2 | Y_1) \leq V(Y_2),$$

with equality if and only if $\rho = \pm 1$. This means that we can predict Y_2 with greater precision if we know Y_1 than if we do not, unless the two variables are uncorrelated.

Solving for ρ^2 gives

$$\rho^2 = \frac{\sigma_{22} - V(Y_2 | Y_1)}{\sigma_{22}}.$$

We can thus interpret the correlation coefficient in another way, as a measure of predictive ability. **The squared correlation coefficient is the proportion of variance in Y_2 that is explained by Y_1 .**

Notice that $V(Y_2 | Y_1) = \sigma_{22}(1 - \rho^2)$ does not depend on X_1 . Under the bivariate normal model, the variance of Y_2 within vertical strips is the same for all vertical strips.

This property is known as homoscedasticity. In many real bivariate distributions, this assumption is violated.

Consider the joint distribution of height and weight. The variance of weight among adults who are six feet tall will be much greater than the variance of weight among infants who are only 22 inches tall. If $V(Y_2 | Y_1)$ increases with Y_1 , we have several strategies for modeling.

- We can switch from a model of homoscedasticity to a model that explicitly allows $V(Y_2 | Y_1)$ to vary with Y_1 .
- We can apply a variance-stabilizing transformation to Y_2 , to try to make the variance of the response approximately independent of its mean.
- We can apply a power transformation to Y_2 , because it is probably skewed. Correcting for skewness in a response variable may also correct for heteroscedasticity.

Regression to the mean. In his pioneering work on correlation in the late 19th century, Sir Francis Galton made the following observation. The heights of fathers and sons are positively correlated, so taller-than-average fathers tend to have sons who are taller than average. But he

noticed that the sons of tall fathers were not quite as tall as their fathers. That is, if he selected sons whose fathers were 1 standard deviation above the mean, the average height of their sons would be less than one standard deviation above the mean. Similarly, among fathers whose heights are 1 standard deviation below the mean, the average height of their sons is less than one standard deviation below the mean. Sons of tall fathers are not as tall as their fathers, and sons of short fathers are not as short as their fathers, when expressed in standard units.

He named this effect “regression to mediocrity,” and it later became known as regression to the mean. (This is the reason why models of prediction are commonly called regression models.) This raises an interesting question. If sons of tall fathers are not as tall as their fathers, and sons of short fathers are not as short as their fathers, then shouldn’t the sons be more tightly clustered around the mean than the fathers? After a few generations, shouldn’t we notice a drop in the variance of heights?

Of course, we now know that Galton’s observation is exactly what one should see in data of this type. If fathers’ and sons’ heights are correlated with correlation coefficient ρ , then if we restrict our attention to fathers who are 1 standard deviation above (or below) average, their sons average heights will be ρ standard deviations above (or below) average. The reason why offspring over successive generations do not converge to the same height is that this

prediction principle also works in reverse. If we examine sons who are 1 standard deviation above (or below) average, their fathers tend to be ρ standard deviations above (or below) average. But if we look backward in time, previous generations do not have a range of height much narrower than the current generation. Regression to the mean is simply an observation that the slope of the regression line for predicting one variable from another in standard units is ρ rather than 1.

Regression to the mean has important implications for how we interpret the results of an intervention. Suppose that a doctor prescribes medication for all of his patients who have high blood pressure. Upon re-examination, he finds that, on average, the blood pressure of these patients has dropped. Does this mean that the medication is effective? Not at all. Blood pressure measurements contain a considerable amount of error. (The correlation between repeated measurements on the same individual is only about 0.6.) So even if the medication did nothing, we would expect the average blood pressure readings for the selected patients to drop due to regression to the mean.

Another way to understand this phenomenon is to think of each person's blood pressure measurement as consisting of a "true score" plus a "random error." If we select patients with high measurements, many of them will be included because their true scores are high, but others will be included because their chance errors are high. The next

time we measure them, not all of them will have high chance errors again, so the overall mean of this select group will drop merely because of chance.

A discrete version of regression to the mean was described in a humorous example by Robbins. Suppose that we toss 100 pennies, and 50 of them come up heads. Suppose we gather up the fifty pennies that came up heads and say to them, “You naughty pennies! Fifty percent of you were supposed to come up tails, but all of you were heads! Try to do better next time!” Then you toss the pennies again, and this time, 25 of them came up heads and the other 25 came up tails. Then you triumphantly announce that you “cured” this group of bad pennies. Of course, the effect is nothing but regression to the mean. In medical experiments, however, we can easily be misled by this type of effect. If we select our experimental subjects because they appear to be the sickest ones, then many of them will show improvement over time simply because of regression to the mean.

How do we overcome this problem in an intervention experiment? The solution is to include a **control group**. Before the experiment begins, randomly assign the subjects to receive either the medication, or to receive nothing. (Better yet, give the latter group a placebo.) Then the behavior of the control group will provide a baseline against which we can assess the improvement of the treatment group.

THE SAMPLE CORRELATION

Sample correlation. Last time, we discussed the interpretation of the population correlation coefficient ρ when the variables in question are normally distributed. Today we will discuss the sample correlation. Suppose that we have paired measurements (x_i, y_i) for each unit in a sample. The sample correlation is

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}.$$

where \bar{x} and \bar{y} are the sample means, sums are taken over the sample units $i = 1, \dots, n$.

Suppose we standardize each variable to have a mean of zero and a sample variance of one,

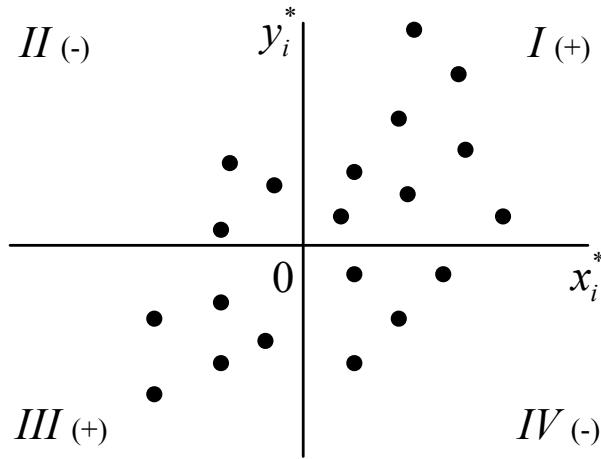
$$\begin{aligned} x_i^* &= (x_i - \bar{x})/\hat{\sigma}_x, \\ y_i^* &= (y_i - \bar{y})/\hat{\sigma}_y, \end{aligned}$$

where $\hat{\sigma}_x$ and $\hat{\sigma}_y$ are the sample standard deviations of x_i and y_i , computed with denominators of n rather than $n - 1$. The sample correlation can then be written as the

average product of the standardized variables,

$$r = \frac{1}{n} \sum_i x_i^* y_i^*.$$

Suppose that we plot the transformed data (x_i^*, y_i^*) , $i = 1, \dots, n$ on the x^*-y^* plane, like this.



The product $x_i^* y_i^*$ is positive in quadrant I, where the original variables x_i and y_i are both larger than average, and in quadrant III, where x_i and y_i are both smaller than average. And $x_i^* y_i^*$ is negative in quadrants II and IV, where x_i is large but y_i is small, and vice-versa. Therefore, a positive value of r indicates a tendency for high values of x_i and y_i to occur together, and for low values of x_i and y_i to occur together.

Vectors, distance, angles and projection in n -dimensional space. In elementary statistics courses, we are taught to create a scatterplot, which marks the

position of each pair of observations (x_i, y_i) on the X - Y plane. That is, we are taught to visualize the data as a set of n points in two-dimensional space. Mathematical statisticians, however, prefer to think of the data as two points in n -dimensional space,

$$\begin{aligned} x &= (x_1, x_2, \dots, x_n)^T, \\ y &= (y_1, y_2, \dots, y_n)^T. \end{aligned}$$

Many of the fundamental ideas of correlation, regression and linear models in general have a very nice interpretation in terms of n -dimensional space. To help us understand these ideas, let's briefly review some concepts of n -dimensional Euclidean geometry. Let

$$a = (a_1, a_2, \dots, a_n)^T$$

and

$$b = (b_1, b_2, \dots, b_n)^T$$

denote two sets of coordinates in n -dimensional space. The **squared distance** between these points is

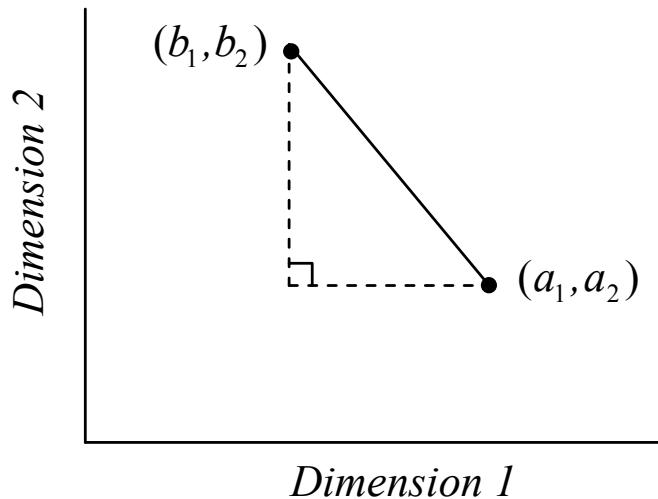
$$(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2,$$

and the **distance** is

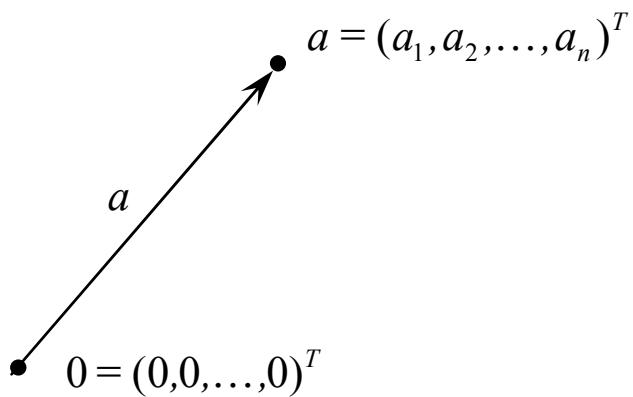
$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2}.$$

You can recognize this as a generalization of the Pythagorean Theorem. In two dimensions, the squared length of the line segment between two points, (a_1, a_2) and

(b_1, b_2) , is equal to squared length of the first leg,
 $(a_1 - b_1)^2$, plus the squared length of the second leg,
 $(a_2 - b_2)^2$, of the right triangle shown below.



A point $a = (a_1, a_2, \dots, a_n)^T$ may also be visualized as a **vector**, i.e. a directed line segment, which begins at the origin $0 = (0, 0, \dots, 0)^T$ and ends at a .

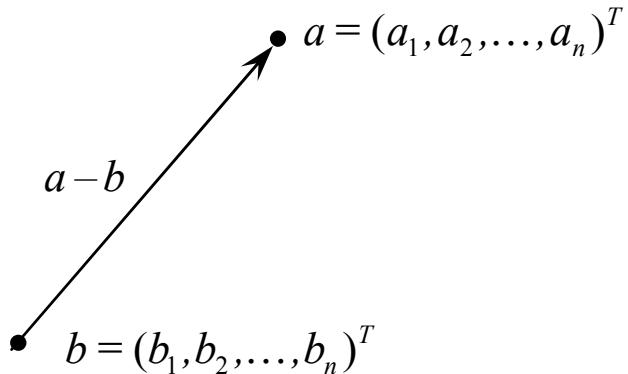


The **length** or **magnitude** of this vector is the distance

from 0 to a , which is written as

$$\|a\| = \sqrt{\sum_i (a_i - 0)^2} = \sqrt{\sum_i a_i^2} = \sqrt{a^T a}.$$

The difference between two vectors, $(a - b)$, can be regarded as a directed line segment starting at b and ending at a .



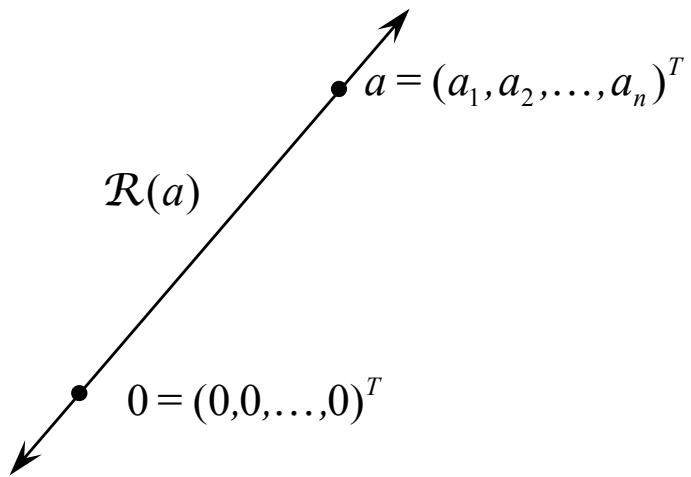
The distance between a and b is the magnitude of this difference,

$$\|a - b\| = \sqrt{\sum_i (a_i - b_i)^2} = \sqrt{(a - b)^T (a - b)}.$$

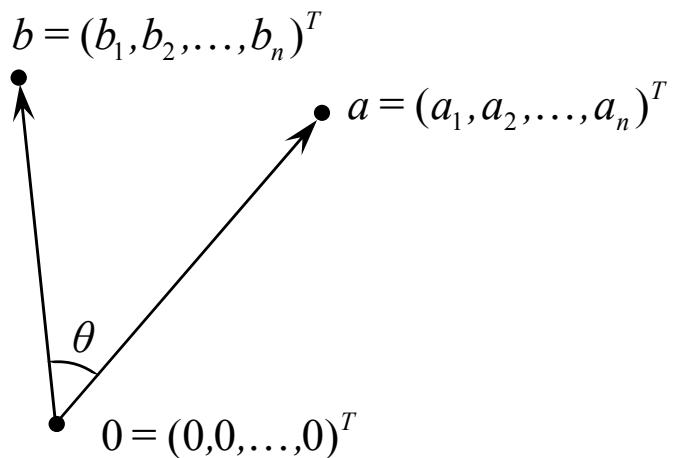
The **linear space** spanned by $a = (a_1, a_2, \dots, a_n)^T$, which we write as $\mathcal{R}(a)$, is the collection of points

$$\beta \cdot a = (\beta a_1, \beta a_2, \dots, \beta a_n)^T$$

for all real numbers β . We can visualize $\mathcal{R}(a)$ as a line passing through 0 and a and continuing indefinitely in both directions.



The **angle** between $a = (a_1, a_2, \dots, a_n)^T$ and $b = (b_1, b_2, \dots, b_n)^T$ can be visualized as the angle between the line segment connecting 0 to a , and the line connecting 0 to b .



The **cosine** of this angle is

$$\cos \theta = \frac{a^T b}{\sqrt{(a^T a)(b^T b)}}.$$

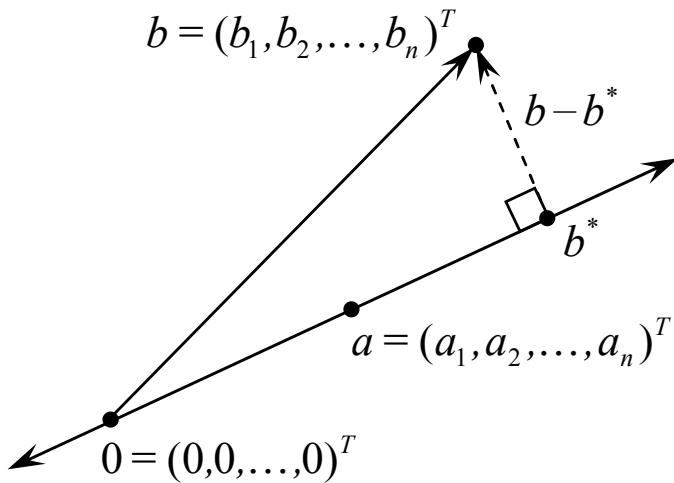
If the two vectors point in exactly the same direction (i.e., if $b = \beta a$ for some $\beta > 0$, then $\cos \theta = 1$). The two vectors are said to be **orthogonal** if $a^T b = 0$, in which case the cosine of the angle between them is 0.

The last major concept that we will introduce now is **projection**. The projection of a vector b onto a vector a —or, more accurately, the projection of b onto the linear space spanned by the vector a —is the point in $\mathcal{R}(a)$ that is closest to b in terms of Euclidean distance. This projection, which we call b^* , is given by the formula

$$b^* = \left(\frac{a^T b}{a^T a} \right) a. \quad (1)$$

Proving that this is indeed the point within $\mathcal{R}(a)$ that minimizes $\|b - b^*\|$ is a straightforward exercise in calculus and will be left as a homework exercise.

To visualize the projection, imagine dropping a line segment from b to $\mathcal{R}(a)$ that is perpendicular (i.e., orthogonal) to $\mathcal{R}(a)$. The point at which that line segment meets $\mathcal{R}(a)$ is b^* .



Using the formula (1), it is easy to show that the difference between the original vector and its projection, $b - b^*$, is indeed orthogonal to a . This will be left as another homework exercise.

A geometric interpretation of the sample correlation. Getting back to the correlation coefficient, let us suppose that we have a set of bivariate observations,

$$\{(x_i, y_i) : i = 1, \dots, n\},$$

which we now regard as two vectors in n -dimensional space,

$$\begin{aligned} x &= (x_1, x_2, \dots, x_n)^T, \\ y &= (y_1, y_2, \dots, y_n)^T. \end{aligned}$$

Now consider the n -dimensional unit vector

$$1 = (1, 1, \dots, 1)^T.$$

The space spanned by this vector, $\mathcal{R}(1)$, is the collection of all n -dimensional vectors of the form

$$k \cdot 1 = (k, k, \dots, k)^T$$

for some real number k . Using (1), the projection of the vector x onto $\mathcal{R}(1)$ is

$$x^* = \left(\frac{\sum_i x_i}{\sum_i 1^2} \right) \cdot 1 = \bar{x} \cdot 1 = (\bar{x}, \bar{x}, \dots, \bar{x})^T,$$

where \bar{x} is the sample mean of x_1, x_2, \dots, x_n . Similarly, the projection of y onto $\mathcal{R}(1)$ is

$$y^* = \bar{y} \cdot 1 = (\bar{y}, \bar{y}, \dots, \bar{y})^T,$$

where \bar{y} is the sample mean of y_1, y_2, \dots, y_n . The difference between x and its projection,

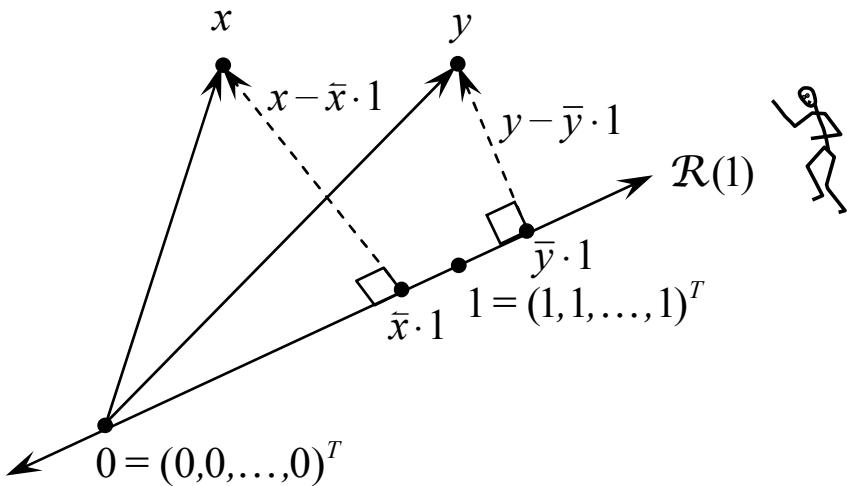
$$x - x^* = (x_1 - \bar{x}, x_2 - \bar{x}, \dots, x_n - \bar{x})^T,$$

and the difference between y and its projection,

$$y - y^* = (y_1 - \bar{y}, y_2 - \bar{y}, \dots, y_n - \bar{y})^T,$$

are just the original data vectors x and y shifted by a constant to have mean zero. They represent the deviations of the observations from their means. Now we are ready to interpret the sample correlation. **The correlation coefficient is the cosine of the angle between these two vectors of deviations.** Visually, imagine dropping a perpendicular from x onto $\mathcal{R}(1)$, and another perpendicular from y onto $\mathcal{R}(1)$. The correlation

coefficient is the cosine of the angle between these two perpendiculars, $x - \bar{x} \cdot 1$ and $y - \bar{y} \cdot 1$.



(Imagine viewing this angle from the position of the person in the drawing above.)

If the deviations of the x_i 's from their mean tend to rise and fall together with the deviations of the y_i 's from their mean, then these two deviation vectors will point in a similar direction; then the angle between them will be small, and the cosine of the angle will be close to 1. If the two variables are not related, the deviation vectors will tend to be nearly orthogonal to each other, and the cosine of the angle will be close to zero.

Inferences about the population correlation. The sample correlation coefficient r is a consistent estimate of the population correlation coefficient ρ under any reasonable bivariate distribution. (Consistency means that

r converges in probability to ρ as $n \rightarrow \infty$.) Beyond this point estimate, however, we may need to test the null hypothesis $H_0 : \rho = 0$ or compute a confidence interval for ρ .

One way to test $H_0 : \rho = 0$ is to convert r to a t -statistic. If we were to fit a simple linear regression of y_i on x_i , which assumes that

$$y_i | x_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2), \quad (2)$$

then a test of the null hypothesis that the slope is zero, $H_0 : \beta_1 = 0$, is equivalent to a test of zero correlation. In Lecture 4, we learned that the bivariate normal model,

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \sigma_{xx} & \rho \sigma_x \sigma_y \\ \rho \sigma_x \sigma_y & \sigma_{yy} \end{pmatrix}\right),$$

implies that the conditional distribution of y_i given x_i is given by (2), with

$$\begin{aligned} \beta_1 &= \rho \sigma_y / \sigma_x, \\ \beta_0 &= \mu_y - \beta_1 \mu_x, \\ \sigma^2 &= \sigma_{yy} (1 - \rho^2). \end{aligned}$$

Therefore, $\beta_1 = 0$ if and only if $\rho = 0$. (Notice that the simple linear regression model (2) is more general than bivariate normality, because it also covers situations where the predictor x_i is not normally distributed, e.g. when it is a binary indicator.)

The test for $H_0 : \beta_1 = 0$ in a simple linear regression model is based on a t -statistic,

$$t = \hat{\beta}_1 / SE(\hat{\beta}_1), \quad (3)$$

where $\hat{\beta}_1$ is the ordinary least-squares estimate of β_1 , and $SE(\hat{\beta}_1)$ is its standard error. This statistic is compared to a Student's t -distribution with $n - 2$ degrees of freedom. We will cover the theory of simple linear regression and give formulas for $\hat{\beta}_1$ and $SE(\hat{\beta}_1)$ in about one week. For now, we will simply note that there is a relationship between the sample correlation coefficient r and the t -statistic given by (3). The relationship is

$$r^2 = \frac{t^2}{t^2 + df}, \quad (4)$$

where $df = n - 2$ is the degrees of freedom. Solving for t^2 gives

$$t^2 = \left(\frac{r^2}{1 - r^2} \right) df.$$

Because r and $\hat{\beta}_1$ have the same sign, we can also write it as

$$t = \text{sign}(r) \sqrt{\left(\frac{r^2}{1 - r^2} \right) df}.$$

If we convert r to a t -statistic and find that t is greater than 2 or less than -2 , we can say that the correlation between the two variables is statistically significant. More formally, we can compute the p-value by finding the

cumulative distribution function at $-|t|$ and multiplying by 2 for a two-tailed test.

Let's do this in R for the blood pressure data we examined in Lecture 2. The function `cor` computes the correlation between two vectors.

```
> # read the data file
> bp <- read.table( "bp.dat", header=T)
>
> # find the correlation between systolic and diastolic blood pressure
> r <- cor( bp$BPSYS, bp$BPDIAS )
> r
[1] 0.7135145
>
> # convert to t-statistic
> n <- nrow(bp)
> df <- n - 2
> tstat <- sqrt( ( r^2 / (1-r^2) ) * df )
> tstat
[1] 18.13164
>
> # find the p-value
> pval <- 2 * pt( -tstat, df)
> pval
[1] 6.581913e-51
```

Because these blood pressure measurements are skewed, the assumptions of the normal linear regression model (2) are not quite satisfied. Nevertheless, this test is quite robust to departures from normality, especially when n is large. In this case, the t -statistic is so large that there is absolutely no doubt about the significance of the result.

The formula (4) also gives us a handy way to find an approximate critical value for the sample correlation, i.e. the value of r that is statistically significant from zero at

the .05 level. A correlation will be significant if $|t|$ is greater than about 2, i.e. if $t^2 > 4$. Therefore, the approximate critical value for r is $\sqrt{4/(4 + df)}$. In a sample of $n = 50$, for example, a correlation will be significant if it is greater than $\sqrt{4/(4 + 50 - 2)} = 0.28$.

MORE ABOUT CORRELATION

Confidence intervals for ρ . To test the null hypothesis $H_0 : \rho = 0$, we converted the sample correlation to a t -statistic using the formula

$$t = \text{sign}(r) \sqrt{\left(\frac{r^2}{1 - r^2} \right) df},$$

where $df = n - 2$. This statistic has a Student's t -distribution when $\rho = 0$. To develop a confidence interval, we need to understand the distributional properties of r for arbitrary values of ρ .

The sampling distribution of r under bivariate normality was first derived by Fisher in 1915 and is a bit complicated. It turns out, however, that when n is large, this distribution is approximately normal with mean ρ and variance

$$V(r) \approx \frac{(1 - \rho^2)^2}{n - 1}.$$

One way to form a confidence interval for ρ is to plug the sample correlation r into the variance formula above, and

take

$$r \pm 1.96 \frac{(1 - \rho^2)}{\sqrt{n - 1}}$$

as an approximate 95% confidence interval. An even better way is to apply a transformation to r to stabilize its variance. The formula above indicates that the variance of r is approximately proportional to $(1 - E(r)^2)^2$. This implies that the variance-stabilizing transformation for r is the inverse hyperbolic tangent,

$$g(r) = \tanh^{-1}(r) = \frac{1}{2} \log \left(\frac{1+r}{1-r} \right).$$

This transformation was first presented by Fisher (1921) and is commonly known as Fisher's z -transformation. Fisher showed that, under an assumption of bivariate normality, $g(r)$ is approximately normally distributed with mean $g(\rho)$ and variance $1/(n - 3)$. An approximate 95% confidence interval for $g(\rho)$ is simply

$$g(r) \pm 1.96 \sqrt{\frac{1}{n - 3}}.$$

If $z = g(r)$, then the inverse transformation is

$$r = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}.$$

Therefore, to get a confidence interval for ρ , we can find the interval for $g(\rho)$ and apply \tanh to its endpoints. In R, the hyperbolic tangent and its inverse are available as `tanh` and `atanh`. Let's apply this procedure to the blood

pressure data.

```
> # compute a confidence interval for r
> bp <- read.table( "bp.dat", header=T)
> r <- cor( bp$BPSYS, bp$BPDIAS )
> z <- atanh(r)
> n <- nrow(bp)
> z.low <- z - 1.96 / sqrt(n-3)
> z.high <- z + 1.96 / sqrt(n-3)
> r.low <- tanh( z.low )
> r.high <- tanh( z.high )
> c( r.low, r.high )
[1] 0.6550242 0.7635049
```

In the vicinity of $r = 0$, $\tanh^{-1}(r)$ is virtually a straight line through the origin with slope 1. For values of r near zero, $\tanh^{-1}(r) \approx r$. In fact, if $|r| \leq 0.24$, then $\tanh^{-1}(r)$ and r agree to two decimal places. Therefore, if the correlation coefficient is small, we can apply the quick-and-dirty approximation $r \sim N(\rho, (n - 3)^{-1})$ and use $r \pm 2/\sqrt{n - 3}$ as our approximate 95% confidence interval for ρ . If n is large, we can even use $r \pm 2/\sqrt{n}$.

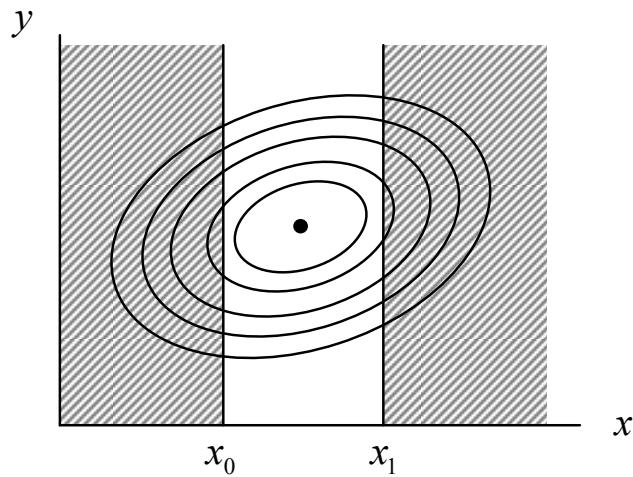
For example, suppose $r = 0.10$ in a sample of $n = 100$. A quick-and-dirty 95% confidence interval for ρ is

$$.10 \pm \frac{2}{\sqrt{100}} = (-.10, .30).$$

Because the interval covers zero, the correlation is clearly not significant at the 0.05 level.

Correlation and the distribution of x . The sample correlation r is a consistent estimate of ρ if the observations (x_i, y_i) , $i = 1, \dots, n$ are a simple random

sample from the population of interest. But if the sample is not representative of the population, then r can be a badly biased estimate. In particular, r will be biased if the sampling mechanism causes the x 's in the sample to be more or less variable than the x 's in the population. For example, suppose that we omit from the sample any unit for which $x < x_0$ or $x > x_1$. That is, we observe (x_i, y_i) only if the point lies in the unshaded region below. Then the value of r from the sample will tend to be considerably smaller than ρ .



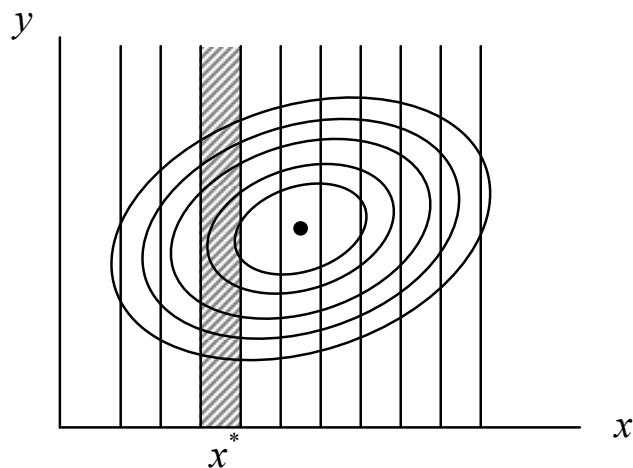
If we do not deterministically omit every observation in the shaded regions, but merely subsample them, then the variance of the x 's in our sample will still be lower than the variance of x in the population, and the sample correlation will still be biased toward zero. On the other hand, if our sample selection method produces x 's with greater variance than the population (e.g., by

oversampling units with extreme values of x) then the correlation in the sample will tend to be larger than the correlation in the population.

For a real-life example, suppose that we compute the correlation between $x =$ total score on the SAT exam and $y =$ first-year grade-point average among undergraduate students at Penn State. We may find that the correlation is unusually small. Would this indicate that SAT score is a poor predictor of academic achievement and should not be given much weight in the process of admissions? Not at all. The students enrolled at Penn State do not have the same distribution of x as all those who applied. Students with very low SAT scores are not admitted to Penn State. Students with very high SAT scores are admitted, but they often choose to attend more prestigious universities.

Therefore, the variance in SAT scores among students enrolled at Penn State will be lower than the variance among those who applied, and the correlation between x and y among students enrolled at Penn State will also tend to be lower than the correlation among those who applied. (For students who applied but did not come to Penn State, think of y as the hypothetical GPA that they would have had if they had enrolled.)

If the probability that a unit is included in the sample depends on x but not on y , then it is possible to estimate ρ in the greater population from the biased sample. Consider the vertical strip in the figure below where $x \approx x^*$.



If the sampling probabilities depend on x but not y , then the proportion of points falling within this strip in the sample may be very different from the proportion of points falling within this strip in the population. Within the vertical strip, however, the observed values of y are a representative sample from the population conditional distribution of y given that $x = x^*$. If the population is bivariate normal, then the mean of this conditional distribution is

$$E(y | x) = \beta_0 + \beta_1 x,$$

where $\beta_1 = \rho \sigma_y / \sigma_x$ and $\beta_0 = \mu_y - \beta_1 \mu_x$, and the conditional variance is

$$V(y | x) = \sigma_{yy|x} = \sigma_{yy}(1 - \rho^2).$$

The three parameters of this conditional distribution—the intercept β_0 , the slope β_1 , and the residual variance $\sigma_{yy|x}$ —can be consistently estimated from the simple

linear regression of y on x in the sample. Once these estimates are available, we can backsolve to get the implied estimate for ρ . With a little algebra, the result is

$$\rho = \sqrt{\frac{\beta_1^2 \sigma_{xx}}{\sigma_{yy \cdot x} + \beta_1^2 \sigma_{xx}}},$$

where σ_{xx} is the variance of x **in the population of interest** (not in the sample). To apply this formula, we need to know something about the variability of x in the population. We often do know this. In the SAT-GPA example, we would estimate σ_{xx} by the variance of the SAT scores among all those who applied to Penn State. Then we would estimate β_0 , β_1 and $\sigma_{yy \cdot x}$ by regressing GPA on SAT scores among students enrolled at Penn State.

We have not yet talked about how to compute the estimates of the parameters $(\beta_0, \beta_1, \sigma_{yy \cdot x})$ from the linear regression of y on x ,

$$y_i \mid x_i \sim N(\beta_0 + \beta_1 x_i, \sigma_{yy \cdot x}).$$

Formulas for these estimates will be given next week. Without knowing the formulas, however, we can still compute the estimates using statistical software. In R, there are several different functions for fitting a linear regression model. The most popular one is `lm` (for “linear model”). To illustrate, let’s use our blood pressure data to fit a linear regression to predict $y = \text{systolic blood pressure}$

from $x = \text{diastolic blood pressure}$:

```

> bp <- read.table( "bp.dat", header=T)
> y <- bp$BPSYS
> x <- bp$BPDIAS
> result <- lm( y ~ x ) # simple linear regression of y on x
> summary( result )

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-34.111 -8.652 -2.196  8.233 45.604 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 50.56202   4.55329   11.11  <2e-16 ***
x           0.97149   0.05358   18.13  <2e-16 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.34 on 317 degrees of freedom
Multiple R-Squared: 0.5091,    Adjusted R-squared: 0.5076 
F-statistic: 328.8 on 1 and 317 DF,  p-value: < 2.2e-16

```

In this example, the estimate of β_0 is 50.5620, the estimate of β_1 is 0.97149, and the estimate of $\sigma_{yy.x}$ is $13.34^2 = 177.96$.

Let's perform a small simulation to illustrate the effects of restricting the range of x values on the correlation between x and y . To start, let's generate a sample of $n = 100,000$ observations from the bivariate normal distribution

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} \sim N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right)$$

for $\rho = 0.75$. One easy way to do this is to note that

$$y_i | x_i \sim N(\rho x_i, (1 - \rho^2)),$$

so we can sample x_1, \dots, x_n from $N(0, 1)$, and then take

$$y_i = \rho x_i + \epsilon_i,$$

where $\epsilon_i \sim N(0, (1 - \rho^2))$.

```
> n <- 100000
> rho <- 0.75
> x <- rnorm(n)
> y <- rho*x + rnorm(n, sd=sqrt(1-rho^2))
```

The sample correlation is very close to ρ :

```
> cor(x,y)
[1] 0.7504039
```

Let's see what happens to the correlation if we omit from the sample any observation with $|x_i| > 1.25$.

```
> select <- ( abs(x) <= 1.25 )
> x.select <- x[ select ]
> y.select <- y[ select ]
> cor( x.select, y.select )
[1] 0.596116
```

Now let's use our procedure to estimate the population ρ from this biased sample. First, we regress y on x in the selected sample and save the estimated values of β_0 , β_1 and $\sigma_{yy \cdot x}$.

```
> result <- lm( y.select ~ x.select )
> summary( result )

Call:
lm(formula = y.select ~ x.select)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-2.955551	-0.446280	0.001145	0.446944	2.826302

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.002414	0.002365	-1.021	0.307
x.select	0.761104	0.003650	208.526	<2e-16 ***

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

```
Residual standard error: 0.6644 on 78882 degrees of freedom
```

```
Multiple R-Squared: 0.3554, Adjusted R-squared: 0.3553
```

```
F-statistic: 4.348e+04 on 1 and 78882 DF, p-value: < 2.2e-16
```

```
> beta.0 <- -0.002414
> beta.1 <- 0.761104
> sigma.yy.x <- 0.6644^2
```

Then we use our formula to obtain the bias-corrected estimate of ρ .

```
> sigma.xx <- var(x) # variance of x, not the variance of x.select
> rho <- sqrt( (beta.1^2 * sigma.xx) /
+   (sigma.yy.x + beta.1^2 * sigma.xx) )
> rho
[1] 0.7525565
```

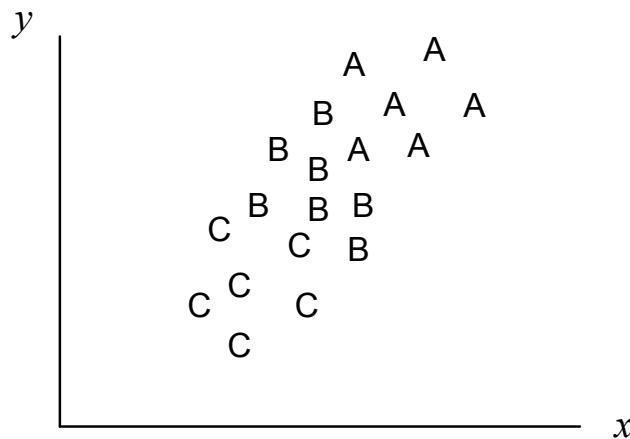
Ecological correlation. Sometimes the units in a sample are grouped into larger units. In educational research, for example, students are grouped into schools. If we want to investigate the relationship between two variables x and y , we can imagine estimating the correlation in two different ways.

- Compute the correlation between x and y among all units in the sample, ignoring the fact that they are clustered.

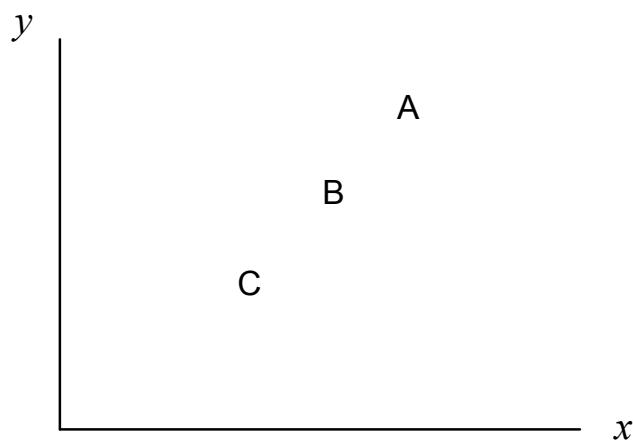
- Compute the correlation between the average value of x in each cluster and the average value of y in each cluster.

(There is also a third way: Remove the effect of cluster by computing a partial correlation. More about that later.)
In general, correlations computed by the two methods above will not agree.

To see why they might not agree, suppose that we sample children within three schools (A, B, C) and record their scores on a test of reading achievement (x) and a test of math achievement (y). And suppose that school A has a lot of high achievers, school C has a lot of low achievers, and school B is somewhere in between. The scatterplot of the individual student scores might look like this,



while the scatterplot of the school average scores would look like this.



The correlation in the second plot is clearly higher than the correlation in the first plot.

Sometimes researchers will have access to data only at the aggregate (e.g. school) level, and they may conclude that relationships seen at that higher level will also be present among units at a lower level (e.g. students within the schools). That false conclusion is known as **the ecological fallacy**. The two correlation coefficients may be very different. They could even have different signs.

There's a well known example of this in American politics. Individually, there is a positive relationship between income and the tendency to vote for Republican candidates. Persons with lower income are more likely to vote for Democrats, and persons with higher income are more likely to vote for Republicans. Looking at aggregate statistics for geographic areas (e.g., counties or congressional districts), however, the tendency is reversed.

Geographical areas with higher median income tend to support Democrats, and areas with lower median income tend to support Republicans. Therefore, either party could be characterized as “the party of the rich” or “the party of the poor.”

Partial correlation. More generally, the relationship between any two variables x and y could be influenced by their mutual associations with a third variable z . Suppose that three variables have a multivariate normal distribution,

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \sim N \left(\begin{pmatrix} \mu_x \\ \mu_y \\ \mu_z \end{pmatrix}, \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} \right).$$

The simple (marginal) correlation between x and y is

$$\rho_{xy} = \frac{\sigma_{xy}}{\sqrt{\sigma_{xx} \sigma_{yy}}} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}.$$

Using properties of the multivariate normal distribution (Lecture 4), we can also find the correlation between x and y at any fixed value of z . The conditional distribution of $(x, y)^T$ given z is bivariate normal with covariance matrix

$$\begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{pmatrix} - \begin{pmatrix} \sigma_{xz} \\ \sigma_{yz} \end{pmatrix} (\sigma_{zz})^{-1} \begin{pmatrix} \sigma_{xz} & \sigma_{yz} \end{pmatrix},$$

which becomes

$$\begin{pmatrix} \sigma_{xx \cdot z} & \sigma_{xy \cdot z} \\ \sigma_{yx \cdot z} & \sigma_{yy \cdot z} \end{pmatrix} = \begin{pmatrix} \sigma_{xx} - \sigma_{xz}^2 / \sigma_{zz} & \sigma_{xy} - \sigma_{xz}\sigma_{yz} / \sigma_{zz} \\ \sigma_{xy} - \sigma_{xz}\sigma_{yz} / \sigma_{zz} & \sigma_{yy} - \sigma_{yz}^2 / \sigma_{zz} \end{pmatrix}.$$

The correlation between x and y at a fixed value of z is

$$\rho_{xy \cdot z} = \frac{\sigma_{xy \cdot z}}{\sqrt{\sigma_{xx \cdot z} \sigma_{yy \cdot z}}}.$$

With a little algebraic manipulation, we can write it in terms of the simple correlations,

$$\rho_{xy \cdot z} = \frac{\rho_{xy} - \rho_{xz}\rho_{yz}}{\sqrt{(1 - \rho_{xz}^2)(1 - \rho_{yz}^2)}}.$$

This quantity is called the “partial correlation” between x and y given z , although a more appropriate name might be “conditional correlation.” It measures the relationship between x and y after accounting for their mutual associations with z . If $\rho_{xy \cdot z} = 0$, then we can say that the association between x and y is fully explained by z .

Notice that the partial correlation between x and y given z does not depend on z . Under the multivariate normal model, the correlation between any two variables given a third is the same for all values of the third. This is another aspect of the “homoscedasticity” property of the multivariate normal distribution. The multivariate normal model has no interactions. (An interaction is a relationship among three variables in which the correlation between two of them changes in relation to the third.)

Just by examining this formula, we see two interesting facts.

1. $\rho_{xy \cdot z} = \rho_{xy}$ if $\rho_{xz} = \rho_{yz} = 0$.
2. $\rho_{xy \cdot z} = 0$ if $\rho_{xy} = \rho_{xz}\rho_{yz}$.

Property 1 says that the simple correlation and the partial correlation are the same when the variable we're conditioning on is independent of both of the variables in question. Property 2 gives us the size of a spurious correlation between two variables when a third variable is ignored. Suppose that the relationship between x and y is fully explained by z ($\rho_{xy \cdot z} = 0$). If we ignore z , then the two variables will appear to be related, and the simple correlation between them (ρ_{xy}) will be equal to the product of their simple correlations with the third variable ($\rho_{xz}\rho_{yz}$). If $\rho_{xy} \neq \rho_{xz}\rho_{yz}$, then the relationship between x and y is not fully explained by z .

Recall that ρ_{xy}^2 , the squared simple correlation between x and y , can be interpreted as the proportion of the variance in y explained by x (or vice-versa). A similar argument can be used to interpret $\rho_{xy \cdot z}^2$. The variance in y explained by z is

$$V(y | z) = \sigma_{yy \cdot z} = \sigma_{yy}(1 - \rho_{yz}^2).$$

The variance in y explained by x and z is

$$V(y | x, z) = \sigma_{yy \cdot xz}.$$

With a little algebra, one can show that

$$\sigma_{yy \cdot xz} = \sigma_{yy \cdot z}(1 - \rho_{xy \cdot z}^2).$$

Therefore, $\rho_{xy \cdot z}^2$ is **the proportion of variance in y explained by x after accounting for the variance explained by z** . Exchanging the roles of y and x , we can also show that

$$\sigma_{xx \cdot yz} = \sigma_{xx \cdot z}(1 - \rho_{xy \cdot z}^2),$$

so $\rho_{xy \cdot z}^2$ is also **the proportion of variance in x explained by y after accounting for the variance explained by z** .

FINAL REMARKS ABOUT CORRELATION

Today we finish our dicussion of correlation and begin to talk about regression.

Discrepancies between simple and partial correlations. Last time, we defined the partial correlation between x and y given z ,

$$\rho_{xy.z} = \frac{\rho_{xy} - \rho_{xz}\rho_{yz}}{\sqrt{(1 - \rho_{xz}^2)(1 - \rho_{yz}^2)}}. \quad (1)$$

The simple correlation between x and y , which we write as ρ_{xy} , tells us little or nothing about $\rho_{xy.z}$. The partial correlation may be larger or smaller than the simple correlation. They may even have opposite signs. (With categorical variables, that phenomenon is called “Simpson’s paradox.”)

Suppose, for example, that you collect two variables,

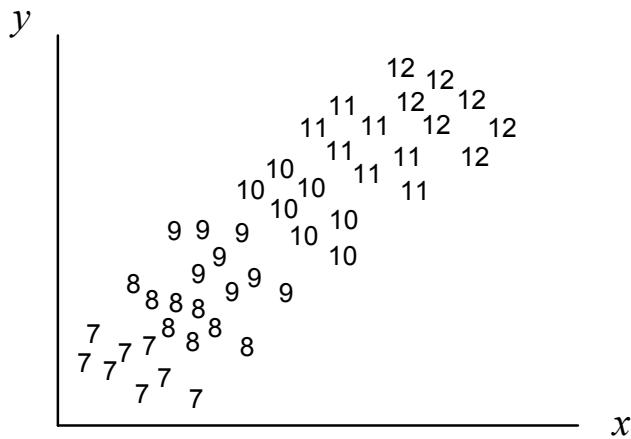
- x = score on a math achievement test, and
- y = number of alcoholic beverages consumed in the last 30 days

for a representative sample of American middle and high school students. It's quite possible that the correlation between these two variables will be *positive*. Students with higher math achievement scores will tend to have higher levels of alcohol consumption. Does this mean that math achievement is a risk factor for adolescent alcohol use? Of course not. Common sense tells us that the opposite should be true; higher levels of math achievement are probably associated with reduced levels of alcohol use.

The problem with the simple correlation is that it does not control for one obvious confounder: age. Suppose that, in addition to x and y , we also record

$$z = \text{grade in school } (7, 8, \dots, 12).$$

Within any grade, we are likely to find a significant negative correlation between x and y . Overall, however, the correlation between x and y may be positive because both are positively correlated with grade. Here is an example of how the data might look, with the effects greatly exaggerated.



Inferences about partial correlation. We may estimate the partial correlation by replacing each population correlation in the formula (1) by its corresponding sample correlation,

$$r_{xy \cdot z} = \frac{r_{xy} - r_{xz}r_{yz}}{\sqrt{(1 - r_{xz}^2)(1 - r_{yz}^2)}}. \quad (2)$$

For example, suppose we examine a sample of $n = 100$ children and record these three variables:

- $x =$ reading comprehension score
- $y =$ body weight
- $z =$ age

Suppose the sample correlation matrix looks like this.

$$\begin{pmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{pmatrix} = \begin{pmatrix} 1.000 & .616 & .827 \\ .616 & 1.000 & .732 \\ .827 & .732 & 1.000 \end{pmatrix}.$$

The high correlation between reading score and age ($r_{xz} = .827$) is to be expected, and so is the high correlation between weight and age ($r_{yz} = .732$). But what about the correlation between reading score and weight ($r_{xy} = .616$)? Does this suggest that children's reading will tend to improve if they gain weight? Notice that

$$r_{xz} \times r_{yz} = .827 \times .732 = .605,$$

which is very close to $r_{xy} = .616$. This suggests that the relationship between x and y might be explained by their mutual associations with z . The estimated partial correlation between x and y given z is

$$r_{xy.z} = \frac{.616 - (.827)(.732)}{\sqrt{(1 - .827^2)(1 - .732^2)}} = .028,$$

which is very close to zero.

Is this value significantly different from zero? The theory for the distribution of the partial correlation coefficient is remarkably similar to that for an ordinary correlation.

Testing the null hypothesis $H_0 : \rho_{xy.z} = 0$ is equivalent to

testing $H_0 : \beta_1 = 0$ in the normal linear regression model

$$y_i | x_i, z_i \sim N(\beta_0 + \beta_1 x_i + \beta_2 z_i, \sigma_{yy \cdot xz}).$$

(This linear regression model will hold if $(x, y, z)^T$ has a multivariate normal distribution. But it is more general, because it may also cover situations where x or z are not normal.) The test of $H_0 : \beta_1 = 0$ is based on the t -statistic

$$t = \hat{\beta}_1 / SE(\hat{\beta}_1),$$

where $\hat{\beta}_1$ is the ordinary least-squares estimate of β_1 , and $SE(\hat{\beta}_1)$ is its standard error. This statistic is compared to a Student's t -distribution with $n - 3$ degrees of freedom. (For a simple correlation, the degrees of freedom were $n - 2$. They are now $n - 3$, because we have estimated one additional parameter, β_2 .) The relationship between this t -statistic and $r_{xy \cdot z}$ is the same as before,

$$r_{xy \cdot z}^2 = \frac{t^2}{t^2 + df},$$

except that now $df = n - 3$. Solving for t gives

$$t = \text{sign}(r_{xy \cdot z}) \sqrt{\left(\frac{r_{xy \cdot z}^2}{1 - r_{xy \cdot z}^2} \right) df}.$$

Applying this to our hypothetical example, we found that $r_{xy \cdot z} = .028$ in a sample of $n = 100$. Converting this to a

t-statistic, we get

$$t = \sqrt{\left(\frac{.028^2}{1 - .028^2} \right) \times 97} = 0.276,$$

which is definitely not significant at the .05 level.

The method for constructing a confidence interval for a partial correlation is also remarkably similar to that of a simple correlation. For a simple correlation, we used Fisher's approximation

$$g(r_{xy}) \sim N\left(g(\rho_{xy}), \frac{1}{n-3}\right),$$

where $g(r) = \tanh^{-1}(r)$. For a partial correlation, the approximation is

$$g(r_{xy.z}) \sim N\left(g(\rho_{xy.z}), \frac{1}{n-4}\right).$$

An approximate confidence interval for $z = g(\rho_{xy.z})$ goes from

$$z_1 = g(r_{xy.z}) - 1.96 \sqrt{\frac{1}{n-4}}$$

to

$$z_2 = g(r_{xy.z}) + 1.96 \sqrt{\frac{1}{n-4}},$$

and the corresponding interval for $\rho_{xy.z}$ goes from $\tanh(z_1)$ to $\tanh(z_2)$. This is how you might do it in R.

```
> rxy <- cor(x,y)
> rxz <- cor(x,z)
> ryx <- cor(y,z)
```

```

> rxy.z <- ( rxy - rxz*ryz ) / sqrt( ( 1-rxz^2 ) * ( 1-ryz^2 ) )
> z.low <- atanh(rxy.z) - 1.96 * sqrt( 1 / (n-4) )
> z.high <- atanh(rxy.z) + 1.96 * sqrt( 1 / (n-4) )
> r.low <- tanh(z.low)
> r.high <- tanh(z.high)

```

Partial correlation and residuals. Another way to compute the sample partial correlation $r_{xy.z}$ is

- regress x on z and save the residuals;
- regress y on z and save the residuals; then
- compute the simple correlation between the two sets of residuals.

If you have already taken a course that involves regression, you know what the residuals are. If you haven't, don't worry; we will learn about residuals very soon. Here is how you would do it in R.

```

> result <- lm( x ~ z )
> res.1 <- result$residuals
> result <- lm( y ~ z )
> res.2 <- result$residuals
> rxy.z <- cor( res.1, res.2 )

```

The value of $r_{xy.z}$ that you would get from this procedure is identical to what you would get from the formula (2).

Partial correlation given additional variables. The formula for partial correlation (1) also holds for conditioning on additional variables. For example, the partial correlation between x and y given z and w can be

written in terms of partial correlations given only w ,

$$\rho_{xy \cdot wz} = \frac{\rho_{xy \cdot w} - \rho_{xz \cdot w}\rho_{yz \cdot w}}{\sqrt{(1 - \rho_{xz \cdot w}^2)(1 - \rho_{yz \cdot w}^2)}}.$$

By repeatedly applying this principle, we can recursively compute partial correlations given any number of variables from the simple correlations.

In practice, however, statisticians do not compute partial correlations this way. Rather, they tend to compute partial correlations by fitting regression models and transforming the t -statistics to correlations by the formula

$$r = \text{sign}(t) \sqrt{\frac{t^2}{t^2 + df}},$$

where df is the sample size n minus the number of coefficients in the regression model. The resulting r is the partial correlation between the response variable and the predictor in question, given all the other predictors in the model. For example, suppose we fit the linear regression model

$$y = \beta_0 + \beta_1 x + \beta_2 w + \beta_3 z + \epsilon.$$

If we take the t -statistic for β_1 , which is $t = \hat{\beta}_1/\text{SE}(\hat{\beta}_1)$, and convert it to r using the formula above, the result value will be $r_{xy \cdot wz}$, the estimated partial correlation between x and y given z and w . (In this case, the degrees of freedom would be $df = n - 4$, because the regression model has four unknown β 's.)

Yet another way to compute the partial correlation between x and y given any set of variables is to

- regress x on that set of variables and save the residuals;
- regress y on that set of variables and save the residuals; then
- compute the simple correlation between the two sets of residuals.

Partial correlation given a grouping variable.

Sometimes we want to estimate the partial correlation between two variables, x and y , given a non-numeric grouping variable. For example, suppose you collected the answers to two questions from a sample of likely voters:

$x =$ In the next presidential election, how likely
are you to vote for the Republican candidate?
(1=very unlikely, . . . , 5=very likely)

$y =$ What was your income last year?

As we pointed out in the last lecture, the correlation between these two variables in your sample is likely to be positive. If you computed the average values of x and y within geographic areas, however, the correlation between these average scores would be negative. The discrepancy would arise because both of these variables are related to geography. People who live in the same area tend to have

similar incomes, and people who live in the same area also tend to have similar political leanings.

Suppose we wanted to answer the question, “What portion of the association between x and y is not explained by geography?” If $z = \text{geographic area}$ was a numeric variable, we could address this question by computing $r_{xy.z}$ as described above. But in this case, z consists of nominal categories. The partial correlation between x and y given these categories can be computed as follows.

- Subtract from each subject’s value of x the average value of x for his or her geographic area.
- Subtract from each subject’s value of y the average value of y for his or her geographic area.
- Compute the simple correlation between these de-meanned values of x and y .

This is how you might do it in R.

```
> n <- length(x)
> x.new <- numeric(n) # create a blank vector to hold the de-meanned x
> y.new <- numeric(n) # create a blank vector to hold the de-meanned y
> for( k in unique(z) ){
+   x.new[ z==k ] <- x[ z==k ] - mean( x[ z==k ] )
+   y.new[ z==k ] <- y[ z==k ] - mean( y[ z==k ] ) }
> rxy.z <- cor(x.new, y.new)
```

Another way to compute $r_{xy.z}$ is to regress x and y on a set of dummy variables defined by the levels of z , and then correlate the residuals. We will learn about this soon.

Introduction to regression. Regression is different from correlation. Correlation is symmetric in the following sense: The correlation between X and Y is the same as the correlation between Y and X . (We are now going to switch notation and denote random variables by capital letters.) In simple linear regression, however, we designate one of the variables (Y) to be a response and then predict it from one or more X 's. Correlation describes the joint distribution of X and Y , whereas regression describes the conditional distribution of Y given X .

In regression analysis, we treat the X variables (i.e., the predictors) as fixed constants. In many applications, the X 's are random variables not under the control of the investigators. Treating X as fixed is merely a technique to avoid specifying a model for the joint distribution of the predictors, which is usually regarded as a nuisance, to focus attention on the question of interest—namely, the “effect” of the predictors on the response. (Here we use the term “effect” very loosely. It does not mean that the relationship between X and Y is causal. The meaning of causality and techniques for causal inference will be discussed later in the semester.)

A typical linear regression model has the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon, \quad (3)$$

where ϵ is a normally distributed random error with mean zero and variance σ^2 . The unknown parameters are

β_0, \dots, β_p and σ^2 . For notational reasons, it will sometimes be more convenient to write the model as

$$Y = \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon. \quad (4)$$

When written this way, it is understood that in most cases the first predictor variable is actually a constant ($X_1 \equiv 1$). Using the latter notation, we can collect the predictors and coefficients into vectors,

$$\begin{aligned} X &= (X_1, X_2, \dots, X_p)^T, \\ \beta &= (\beta_1, \beta_2, \dots, \beta_p)^T, \end{aligned}$$

and write the model as

$$Y \sim N(X^T \beta, \sigma^2),$$

where conditioning on X is understood.

To derive methods of *exact* inference (exact tests and confidence intervals) about the β 's, we will need to assume that the error term ϵ is normally distributed. Although this seems to limit the usefulness and generality of regression, it turns out that this assumption is not crucial for most purposes. Many (but not all) of the procedures that we will cover are not sensitive to moderate departures from normality, especially if the sample size is large. It is fair to say that, in linear regression, the assumption of normality is the least important assumption that we will make. The other assumptions, which are more crucial, are

- the constancy of the error variance σ^2 , and

- the independence of the errors ϵ across units.

If the X 's are actually random variables, then we are also implicitly assuming that the ϵ 's are uncorrelated with the X 's. That is, we assume that ϵ has mean zero for all units regardless of their values of X .

The no-predictors model. Most textbooks on regression begin with a single predictor,

$$Y = \beta_0 + \beta_1 X + \epsilon, \quad (5)$$

which is called “simple linear regression,” and then move up to the model (3), which is called “multiple linear regression.” In this course, however, we will start with

$$Y = \beta + \epsilon, \quad (6)$$

which can be regarded as a special case of (4) with $p = 1$ and $X_1 \equiv 1$. This model, which we call the “no-predictors” model, is ridiculously simple. But the primary results and the intuition we develop about it will immediately generalize to the more complicated settings.

If the error ϵ is assumed to be normally distributed with mean 0 and variance σ^2 , then (6) is equivalent to $Y \sim N(\beta, \sigma^2)$. The data to fit this model will consist of n independent observations of Y , which we denote by y_1, y_2, \dots, y_n . If we collect these into a vector,

$$\mathbf{y} = (y_1, y_2, \dots, y_n)^T,$$

then the no-predictors model can be written in multivariate form,

$$\mathbf{y} \sim N(\mu, \sigma^2 I), \quad (7)$$

where $\mu = \beta \cdot \mathbf{1}$, and $\mathbf{1} = (1, 1, \dots, 1)^T$. Although the mean of \mathbf{y} is $E(\mathbf{y}) = \mu$, which is a vector of length n , it does not contain n unknown parameters. We have constrained μ to lie within $\mathcal{R}(1)$, the linear space spanned by the vector $\mathbf{1}$, so μ is a function of the single parameter β .

The variance σ^2 is often considered to be a nuisance parameter. Although it may not be of direct interest, we need to pay attention to it, because it will affect the precision of the estimate of β and the prediction of future observations of \mathbf{Y} .

In elementary statistics texts, inferences for β are usually presented first for the situation where σ^2 is known, and then for the situation where σ^2 is unknown. If σ^2 is known, we can use the fact that

$$\bar{y} = \frac{1}{n} \sum_i y_i \sim N(\beta, \sigma^2/n),$$

and thus

$$\frac{\bar{y} - \beta}{\sigma^2/n} \sim N(0, 1), \quad (8)$$

to construct tests and intervals. This result (8) is exactly true if the errors ϵ are normally distributed, and approximately true if they are not because of the Central Limit Theorem.

Of course, the situation where σ^2 is known rarely arises in practice. With real data, we need to estimate σ^2 along with β . The usual unbiased estimate of σ^2 is

$$S^2 = \frac{1}{n-1} \sum_i (y_i - \bar{y})^2.$$

To construct tests and intervals for β , we appeal to the following theorem.

Theorem. If y_1, y_2, \dots, y_n are independent and identically distributed as $N(\beta, \sigma^2)$, then

$$\bar{y} \sim N(\beta, \sigma^2/n), \quad (9)$$

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi_{n-1}^2, \quad (10)$$

and the quantities in (9) and (10) are independent.

It follows from (9) that

$$\frac{\bar{y} - \beta}{\sigma^2/n} \sim N(0, 1),$$

and if we divide this quantity by

$$\sqrt{\frac{S^2}{\sigma^2}} \sim \sqrt{\frac{\chi_{n-1}^2}{n-1}},$$

we get

$$\frac{\bar{y} - \beta}{S/\sqrt{n}} \sim t_{n-1}.$$

A test of $H_0 : \beta = \beta_0$ is based on the statistic

$$t = \frac{\bar{y} - \beta_0}{S/\sqrt{n}},$$

which, under H_0 , has a Student's t -distribution with $n - 1$ degrees of freedom. A confidence interval for β is $\bar{y} \pm TS/\sqrt{n}$, where T is an appropriate quantile from the t_{n-1} distribution.

This theorem is often quoted in undergraduate texts, but proofs are usually omitted. Demonstrating (9) is easy, but demonstrating (10) and the independence of (9) and (10) is more difficult. We will not prove this theorem either. But in the next lecture, we will restate it in a more elegant form that will easily generalize to models with additional predictors.

REGRESSION WITH NO PREDICTORS

Last time, we introduced the no-predictors model, which assumes that $y_i \sim N(\beta, \sigma^2)$ for $i = 1, \dots, n$. Collecting the responses into a vector, $y = (y_1, y_2, \dots, y_n)^T$, the model is

$$y \sim N(\mu, \sigma^2 I),$$

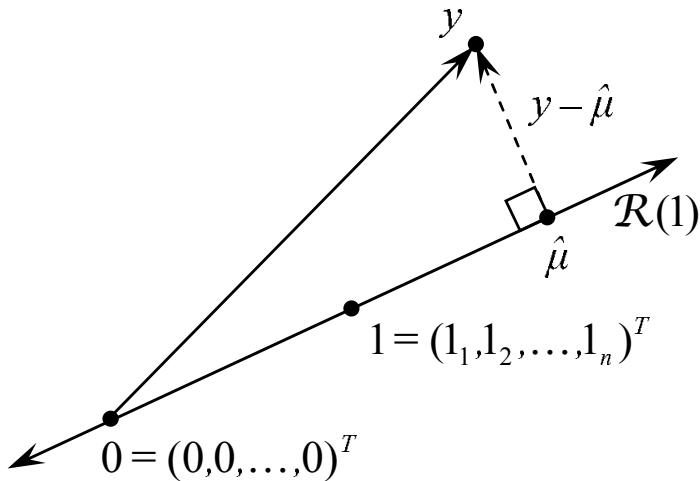
where $\mu = \beta \cdot 1$, and $1 = (1, 1, \dots, 1)^T$. The problem of estimating β is analogous to finding the single number that best represents the whole sample vector y . That is, we want to find an estimate $\hat{\mu} = \hat{\beta} \cdot 1$ that is as close as possible to y . Defining closeness in terms of Euclidean distance, we want to minimize

$$\|y - \beta \cdot 1\|^2 = (y - \beta \cdot 1)^T (y - \beta \cdot 1) = \sum_i (y_i - \beta)^2.$$

This distance is minimized by choosing $\hat{\mu} = \hat{\beta} \cdot 1$ to be the projection of y onto $\mathcal{R}(1)$. The projection is

$$\left(\frac{y^T 1}{1^T 1} \right) 1 = \left(\frac{\sum_i y_i}{n} \right) 1 = \bar{y} \cdot 1,$$

and the estimate of β is $\hat{\beta} = \bar{y}$.



It is legitimate to ask why we should minimize the Euclidean distance rather than some other distance function. Euclidean distance leads to elegant mathematical results and formulas. It also leads to the maximum-likelihood estimate of β under the normal model. But these considerations do not imply that Euclidean distance is always best. If the data are prone to outliers, then we might want to apply another criterion, such as the sum of the absolute deviations $|y_i - \beta|$. The theory and results for these alternative distance functions are messier, but in some cases the resulting estimates will have better properties; e.g., they may be more robust.

The projection $\hat{\mu} = \hat{\beta} \cdot 1$ is often denoted by \hat{y} , and is also called the vector of “fitted values” or “predicted values.” This vector differs from the true mean vector $\mu = \beta \cdot 1$, so we will include the true mean in our picture. Let us also

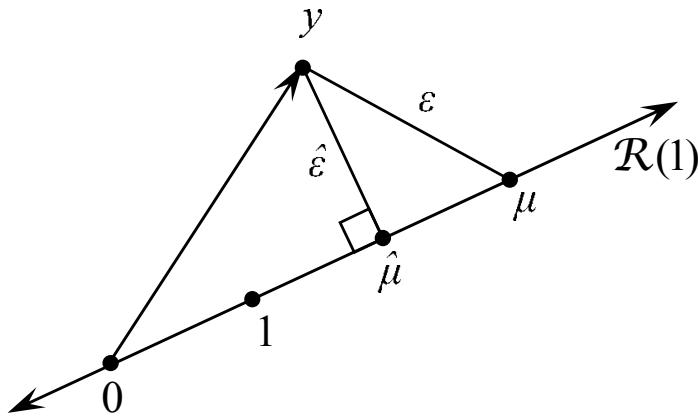
define the vector of true errors,

$$\begin{aligned}\epsilon &= (\epsilon_1, \epsilon_2, \dots, \epsilon_n)^T \\ &= (y_1 - \beta, y_2 - \beta, \dots, y_n - \beta)^T \\ &= y - \mu,\end{aligned}$$

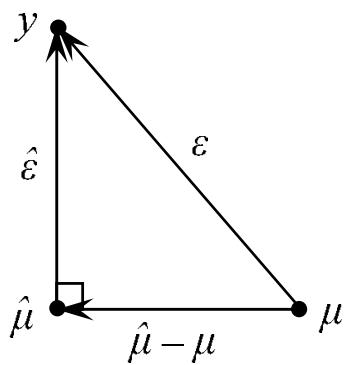
and the vector of estimated errors,

$$\begin{aligned}\hat{\epsilon} &= (\hat{\epsilon}_1, \hat{\epsilon}_2, \dots, \hat{\epsilon}_n)^T \\ &= (y_1 - \hat{\beta}, y_2 - \hat{\beta}, \dots, y_n - \hat{\beta})^T \\ &= y - \hat{\mu},\end{aligned}$$

and include them in the picture as well. The vectors ϵ and $\hat{\epsilon}$ are called the “true residuals” and the “estimated residuals,” respectively. The picture now looks like this.



The theorem that we presented at the end of Lecture 7 pertains to the lengths of the sides of the triangle connecting y , μ and $\hat{\mu}$.



Because this is a right triangle, the Pythagorean Theorem says

$$\|\epsilon\|^2 = \|\hat{\epsilon}\|^2 + \|\hat{\mu} - \mu\|^2.$$

Under the no-predictors model, $\|\epsilon\|^2 \sim \sigma^2 \chi_n^2$. This is easy to verify, because

$$\begin{aligned} \frac{1}{\sigma^2} \|\epsilon\|^2 &= (y - \mu)^T (y - \mu) / \sigma^2 \\ &= \sum_{i=1}^n \frac{(y_i - \beta)^2}{\sigma^2} \\ &= \sum_{i=1}^n z_i^2, \end{aligned}$$

where $z_i = (y_i - \beta) / \sigma \sim N(0, 1)$. The squared length of the hypotenuse, then, is $\sigma^2 \chi_n^2$. The theorem from Lecture 7 describes how this decomposes into the squared lengths of the two legs. We now restate the theorem as follows.

Theorem. Suppose $y = (y_1, y_2, \dots, y_n)^T$ is distributed as $N(\mu, \sigma^2 I)$, where $\mu = \beta \cdot 1$. Let $\hat{\mu} = \hat{\beta} \cdot 1$ denote the projection of y onto (1). Then

$$\|\hat{\mu} - \mu\|^2 \sim \sigma^2 \chi_1^2, \quad (1)$$

$$\|\hat{\epsilon}\|^2 \sim \sigma^2 \chi_{n-1}^2, \quad (2)$$

and the quantities in (1) and (2) are independent.

To see how (1) relates to the result stated in Lecture 7, note that

$$\|\hat{\mu} - \mu\|^2 = n(\bar{y} - \beta)^2,$$

and thus

$$\frac{1}{\sigma^2} \|\hat{\mu} - \mu\|^2 = \left(\frac{\bar{y} - \beta}{\sigma/\sqrt{n}} \right)^2 \sim \chi_1^2,$$

because $\sqrt{n}(\bar{y} - \beta)/\sigma \sim N(0, 1)$. For part (2), note that

$$\frac{1}{\sigma^2} \|\hat{\epsilon}\|^2 = \frac{1}{\sigma^2} \sum_{i=1}^n (y_i - \bar{y})^2 = (n-1)S^2/\sigma^2,$$

which, according to the theorem as stated last time, is distributed as χ_{n-1}^2 .

Inferences about β . If σ^2 were known, then inferences about β could be based on the squared length of the vector in (1). When σ^2 is unknown, we rely on the ratio of (1) to (2), because in the distribution of this ratio the unknown

σ^2 cancels out,

$$\frac{\|\hat{\mu} - \mu\|^2}{\|\hat{\epsilon}\|^2} \sim \frac{\chi_1^2}{\chi_{n-1}^2}.$$

Recall that the ratio of two independent mean-square random variables (a mean square is a χ^2 divided by its degrees of freedom) has an F distribution. Therefore,

$$\frac{\|\hat{\mu} - \mu\|^2}{\|\hat{\epsilon}\|^2/(n-1)} \sim F_{1,n-1}. \quad (3)$$

But $\|\hat{\mu} - \mu\|^2 = n(\hat{\beta} - \beta)^2$ and $\|\hat{\epsilon}\|^2/(n-1) = S^2$ so (3) implies that

$$\frac{(\hat{\beta} - \beta)^2}{S^2/n} \sim F_{1,n-1},$$

and that

$$\frac{\hat{\beta} - \beta}{S/\sqrt{n}} \sim t_{n-1}.$$

Therefore, a test of the null hypothesis $H_0 : \beta = \beta_*$ for any specific value β_* can be carried out by comparing the observed F -statistic

$$F = \frac{(\hat{\beta} - \beta_*)^2}{S^2/n}$$

to $F_{1,n-1}$, or by comparing the observed t -statistic

$$t = \frac{\hat{\beta} - \beta_*}{S/\sqrt{n}}$$

to t_{n-1} . The p-value for a two-tailed test is

$$\begin{aligned} p &= 2 P(t_{n-1} > |t|) \\ &= P(F_{1,n-1} > F). \end{aligned}$$

A measure of effect size. Notice that the F -statistic for testing $H_0 : \beta = \beta_*$ can be written as

$$F = df \times \frac{\|\hat{\mu} - \mu_*\|^2}{\|\hat{\epsilon}\|^2}$$

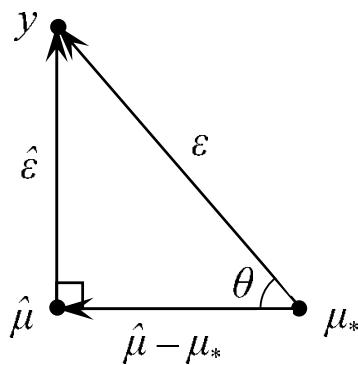
where $df = n - 1$ and $\mu_* = \beta_* \cdot 1$. If we divide F by df , we get

$$\eta = F/df = \frac{\|\hat{\mu} - \mu_*\|^2}{\|\hat{\epsilon}\|^2}.$$

We can regard η as a scale-free measure of “statistical distance” between the estimated mean vector $\hat{\mu}$ and the hypothesized true value μ_* . This distance does not depend on the units in which y_i is measured. A small value of η means that the hypothesized value μ_* is a good approximation to $\hat{\mu}$ and is well supported by the data, whereas a large value of η means that the hypothesized μ_* is contradicted by the data.

Unlike F , η does not tend to grow with the sample size. If we had two different samples from the same population, one with $n = 100$ and the other with $n = 1,000$, and if we tested the same null hypothesis with these two samples, we would expect the F -statistic from the larger sample to be about 10 times as large as the F -statistic from the smaller

sample. F measures the strength of the evidence against the null hypothesis, and the evidence becomes stronger as n grows. But we would expect the values of η from the two samples to be similar. In the social and behavioral sciences, a measure of discrepancy between the data and a null hypothesis that does not depend on n is called an **effect size**. The effect size η has a geometric interpretation as the squared cotangent of the angle θ in the triangle below.



$$\eta = (\cot \theta)^2 = \frac{1}{(\tan \theta)^2} = \frac{||\hat{\mu} - \mu_*||^2}{||\hat{\epsilon}||^2}.$$

Later, when we add predictors to the regression model, we will use trigonometric identities to relate F/df to a correlation coefficient, which can be regarded as another kind of effect size.

Inferences about σ^2 . The second part of our theorem, which says that

$$\|\hat{\epsilon}\|^2 = (n-1)S^2 \sim \sigma^2 \chi_{n-1}^2,$$

allows us to draw inferences about σ^2 . To test $H_0 : \sigma^2 = \sigma_*^2$, we would compare the test statistic

$$\frac{(n-1)S^2}{\sigma_*^2}$$

to χ_{n-1}^2 and reject the null hypothesis if this statistic is too large or too small. In the traditional .05-level, two-tailed test, we would reject H_0 if the test statistic is greater than $\chi_{.975, n-1}^2$ or less than $\chi_{.025, n-1}^2$, where $\chi_{p,\nu}^2$ denotes the p th quantile of χ_ν^2 . A 95% confidence interval for σ^2 would be

$$\left[\frac{(n-1)S^2}{\chi_{.975, n-1}^2}, \frac{(n-1)S^2}{\chi_{.025, n-1}^2} \right].$$

These procedures for σ^2 , unlike the procedures for β , are more sensitive to departures from normality, as we demonstrated in Lecture 2. In particular, if the population has heavier-than-normal tails, the actual Type I error rate for the test and the actual coverage probability for the interval can be substantially worse than their nominal values.

Prediction of a future value of Y . Let $y_* \sim N(\beta, \sigma^2)$ denote a future response drawn from the same population as y_1, \dots, y_n . Our best guess for what this future value will be is $\hat{\beta} = \bar{y}$. What can we say about the quality of this prediction? The prediction is unbiased, because

$$E(y_* - \hat{\beta}) = \beta - \beta = 0.$$

Moreover, if y_* is independent of the current sample y_1, \dots, y_n , then

$$\begin{aligned} V(y_* - \hat{\beta}) &= V(y_*) + V(\hat{\beta}) \\ &= \sigma^2 + \frac{\sigma^2}{n} \\ &= \sigma^2 \left(1 + \frac{1}{n}\right). \end{aligned}$$

Finally, the prediction error $y_* - \hat{\beta}$ is normally distributed, because it is a linear combination of the independent normal variates y_1, \dots, y_n and y_* . If we divide the standard normal variate

$$\frac{y_* - \hat{\beta}}{\sigma \sqrt{1 + \frac{1}{n}}} \sim N(0, 1)$$

by

$$\sqrt{S^2/\sigma^2} \sim \sqrt{\chi_{n-1}^2/(n-1)},$$

we get

$$\frac{y_* - \hat{\beta}}{S\sqrt{1 + \frac{1}{n}}} \sim t_{n-1}.$$

Therefore, the interval

$$\hat{\beta} \pm t_{.975, n-1} S\sqrt{1 + \frac{1}{n}}$$

will cover the future value y_* with probability 95%. This is called a **prediction interval** rather than a confidence interval, because it is intended to capture a future observation rather than a parameter.

The estimated variance of the prediction,

$$S^2 \left(1 + \frac{1}{n}\right) = S^2 + \frac{S^2}{n},$$

reflects our uncertainty about the mean of the population (S^2/n) and our uncertainty about how much the future observation will deviate from that mean (S^2). Inferences about the population mean are quite robust to departures from normality, because the Central Limit Theorem is working for us there. But an inference about how much the future observation will deviate from the mean is not robust, because it pertains to a single observation.

Therefore, we should not expect this prediction interval to work well for non-normal populations. If the observations y_1, \dots, y_n look non-normal, we should consider transforming them to approximate normality before using this procedure.

Regression with predictors. Having developed a thorough understanding of the no-predictors model, introducing predictors will now be a very small step. As before, the responses will be $y = (y_1, \dots, y_n)^T$. However, we will no longer assume that the mean of each y_i is the same. Rather, we will allow the means $E(y_i) = \mu_i$ to vary in relation to a set of predictor variables. Suppose that, for each y_i , we have a vector of p predictor variables,

$$x_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T,$$

where x_{ij} denotes the value of the j th variable for unit i . In most cases, the first “variable” will actually be a constant ($x_{i1} \equiv 1$). Except where specifically noted, however, all of the results that follow will apply whether or not the model includes a constant, and regardless of how the x ’s are distributed.

The model is

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i,$$

where $\epsilon_i \sim N(0, \sigma^2)$ independently for $i = 1, \dots, n$. An equivalent way to write the model is

$$y_i \sim N(x_i^T \beta, \sigma^2)$$

independently for $i = 1, \dots, n$, where

$$\beta = (\beta_1, \beta_2, \dots, \beta_p)^T.$$

It is common practice to collect the responses into an $n \times 1$ vector,

$$\mathbf{y} = (y_1, y_2, \dots, y_n)^T$$

and the predictors into an $n \times p$ matrix,

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}.$$

We can also write this matrix as

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix} = [X_1, X_2, \dots, X_p],$$

where x_i is the $p \times 1$ vector of predictors for unit i , and X_i is the $n \times 1$ vector containing the values of the p th predictor for units $i = 1, \dots, n$. Now the model can also be written as

$$\mathbf{y} \sim N(X\beta, \sigma^2 I).$$

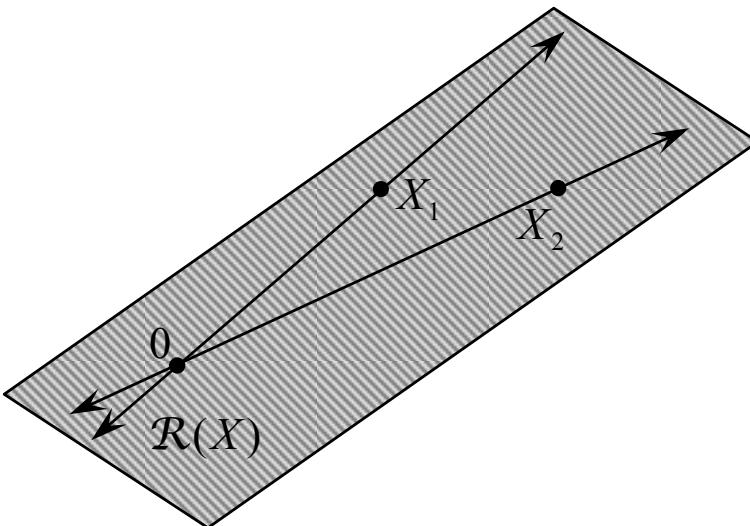
This model asserts that the mean vector $\mu = E(\mathbf{y})$ lies in **the linear space spanned by the columns of X** . This space, which is also known as the range space of X and denoted by $\mathcal{R}(X)$, is the set of all vectors that can be

written as

$$\beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p,$$

where $\beta_1, \beta_2, \dots, \beta_p$ are any real numbers.

For $p = 2$, we can think of $\mathcal{R}(X)$ as the plane that contains the line passing through 0 and X_1 , and the line passing through 0 and X_2 . In other words, it is the plane that contains the triangle with vertices 0, X_1 and X_2 .



If $p > 2$, then $\mathcal{R}(X)$ becomes a hyperplane.

If the columns of X are **linearly independent**, then the dimension of $\mathcal{R}(X)$ is p . The columns of X are said to be linearly independent if the only choice of β 's for which

$$\beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0$$

is $\beta_1 = \beta_2 = \cdots = \beta_p = 0$. If the columns of X are not linearly independent, then one or more columns of X are not needed; they are redundant, because they can be

replaced by linear combinations of other columns.

The set of vectors X_1, X_2, \dots, X_p that span any particular space are not unique. For example, each of these three pairs of vectors,

$$\begin{aligned} &X_1 \quad \text{and} \quad X_2, \\ &2X_1 \quad \text{and} \quad 5X_2, \\ &(X_1 + X_2) \quad \text{and} \quad (X_1 - X_2) \end{aligned}$$

span the same two-dimensional space. The reason is that any linear combination of $2X_1$ and $5X_2$, or any linear combination of $(X_1 + X_2)$ and $(X_1 - X_2)$, can also be written as a linear combination of $X_1 + X_2$. Therefore, a regression model that uses $2X_1$ and $5X_2$ as predictors, or $(X_1 + X_2)$ and $(X_1 - X_2)$ as predictors, will lead to the same predictions as the model that uses X_1 and X_2 . The β 's for the three versions will be different, but the predicted values of y will be the same.

Projection. The key idea of regression is to find the point within $\mathcal{R}(X)$ that closest to y . That is, we need to choose a value of $\beta = (\beta_1, \dots, \beta_p)^T$ to make

$$X\beta = \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

as close as possible to y . We need to find the $\hat{\beta}$ for which $\hat{y} = X\hat{\beta}$ is the projection of y onto $\mathcal{R}(X)$. The projection will minimize the Euclidean distance between y and \hat{y} , or

minimize the length of

$$\hat{\epsilon} = y - X\hat{\beta}.$$

The squared length of $\hat{\epsilon}$ is

$$\begin{aligned} \|\hat{\epsilon}\|^2 &= \hat{\epsilon}^T \hat{\epsilon} \\ &= (y - \hat{y})^T (y - \hat{y}) \\ &= \sum_{i=1}^n (y_i - \hat{y}_i)^2, \end{aligned}$$

where $\hat{y}_i = x_i^T \hat{\beta}$ is the predicted value for y_i . Using calculus, one can show that this projection is

$$\hat{y} = X(X^T X)^{-1} X^T y. \quad (4)$$

(Recall that the projection of the vector y onto the space spanned by a single vector x is $((x^T y)/(x^T x))x$; the formula (4) is a generalization of this.) The value of $\hat{\beta}$ for which $\hat{y} = X\hat{\beta}$ coincides with (4) is

$$\hat{\beta} = (X^T X)^{-1} X^T y,$$

which is called the ordinary least-squares (OLS) estimate for β .

REGRESSION WITH PREDICTORS

In Lecture 8, we introduced predictors into the regression model. Let y denote the $n \times 1$ vector of responses for units $i = 1, \dots, n$, and X the $n \times p$ matrix of predictor variables (the first column is usually a constant). The normal linear regression model assumes that

$$y \sim N(X\beta, \sigma^2 I),$$

where β is a $p \times 1$ vector of unknown coefficients. By setting $E(y)$ equal to $X\beta$, this model assumes that the mean of y lies within the linear space spanned by the columns of X . So another way to write the model is

$$\begin{aligned} y &\sim N(\mu, \sigma^2 I), \\ \mu &\in \mathcal{R}(X). \end{aligned}$$

The estimate of μ is the point within $\mathcal{R}(X)$ that is closest to y in terms of Euclidean distance. That point, which is called the projection of y onto $\mathcal{R}(X)$, is

$$\hat{y} = \hat{\mu} = X(X^T X)^{-1} X^T y.$$

In practice, \hat{y} is called the vector of predicted or fitted

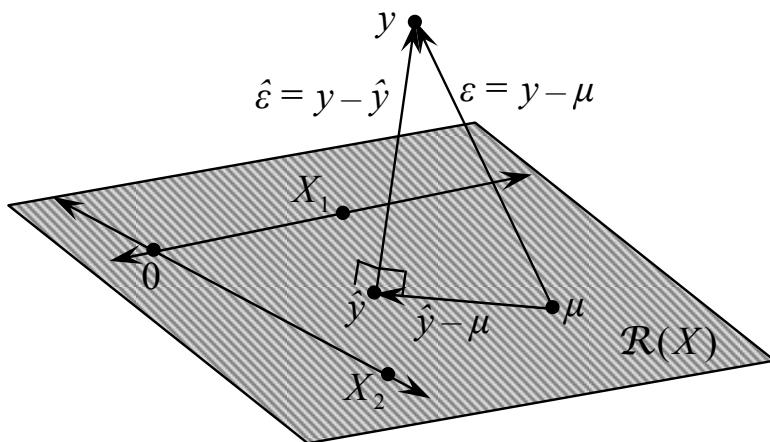
values. It can also be written as $\hat{y} = X\hat{\beta}$, where

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

is called the ordinary least-squares (OLS) estimate for β .

For visualization purposes, let us suppose that X has $p = 2$ columns, which we denote by X_1 and X_2 . The linear space $\mathcal{R}(X)$ is the plane passing through 0 , X_1 and X_2 . Except by a very rare accident, the data vector y will not lie within \mathcal{R} . The closest point to y within $\mathcal{R}(X)$ is $\hat{y} = \hat{\mu} = X\hat{\beta}$. The true mean, $\mu = X\beta$, is unknown, but it also lies within $\mathcal{R}(X)$, and it is farther away from y than \hat{y} is. Let's also define

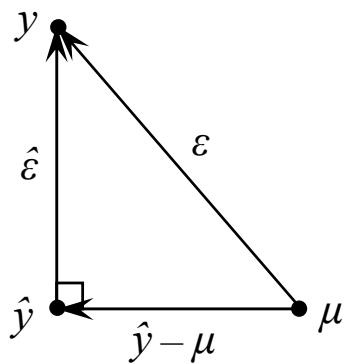
- $\hat{\epsilon} = y - \hat{y}$ to be the difference between y and \hat{y} , which we call the “estimated residuals,” and
- $\epsilon = y - \mu$ to be the difference between y and μ , which we call the vector of “true residuals.”



The vector $\hat{\epsilon} = y - \hat{y}$ is orthogonal to $\mathcal{R}(X)$, which means that it is orthogonal to every vector within $\mathcal{R}(X)$. In

particular, it is orthogonal to $\hat{y} - \mu$. (Both \hat{y} and μ lie within $\mathcal{R}(X)$, so the difference between them also lies within $\mathcal{R}(X)$.)

Let's focus attention on the right triangle with vertices y , \hat{y} and μ .



Now we are ready to state the “Fundamental Theorem of Regression,” which is a simple generalization of the theorem from our last lecture, and we state it without proof.

Theorem. Suppose $y = (y_1, y_2, \dots, y_n)^T$ is distributed as $N(\mu, \sigma^2 I)$, where $\mu = X\beta$ and X is an $n \times p$ matrix whose columns are linearly independent ($\text{rank}(X) = p$). Then

$$\|\hat{\mu} - \mu\|^2 \sim \sigma^2 \chi_p^2, \quad (1)$$

$$\|\hat{\epsilon}\|^2 \sim \sigma^2 \chi_{n-p}^2, \quad (2)$$

and the quantities in (1) and (2) are independent.

By the Pythagorean Theorem,

$$\|\epsilon\|^2 = \|\hat{y} - \mu\|^2 + \|\hat{\epsilon}\|^2,$$

and the additive property of chisquare random variates, it follows that

$$\|\epsilon\| \sim \sigma^2 \chi_n^2.$$

Estimation of σ^2 . The vector of unknown coefficients β is estimated by $\hat{\beta} = (X^T X)^{-1} X^T y$. What about σ^2 ? It follows from (2) that

$$\|\hat{\epsilon}\|^2 = \sum_{i=1}^n \hat{\epsilon}_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3)$$

is a random variable with mean $(n - p)\sigma^2$, so an unbiased estimate of σ^2 is

$$S^2 = \frac{1}{(n - p)} \sum_{i=1}^n (y_i - \hat{y}_i)^2. \quad (4)$$

In the terminology of regression,

- σ^2 is called the **residual variance**,
- (3) is called the **residual sum of squares** or **error sum of squares** or **sum of squared residuals**, and
- S^2 called the **mean squared error** or MSE.

The square root of the MSE,

$$S = \sqrt{\frac{1}{n-p} \sum_{i=1}^n (y_i - \hat{y}_i)^2},$$

is sometimes called the **residual standard error**.

If one were to apply the principle of maximum-likelihood (ML) estimation to this model, the ML estimate for σ^2 would be

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2,$$

which uses a denominator of n rather than $n - p$. The ML estimate is biased downward,

$$E(\hat{\sigma}^2) = \left(\frac{n-p}{n} \right) \sigma^2 < \sigma^2,$$

and thus

$$S^2 = \left(\frac{n}{n-p} \right) \hat{\sigma}^2$$

can be regarded as a bias-corrected version of the ML estimate. Because $n/(n-p) \rightarrow 1$ as $n \rightarrow \infty$, the difference between the two estimates becomes negligible as n grows relative to p .

The result (2) may be used to construct tests and confidence intervals for σ^2 . To test $H_0 : \sigma^2 = \sigma_*^2$, one may compare

$$\frac{\|\hat{\epsilon}\|^2}{\sigma_*^2} = \frac{(n-p)S^2}{\sigma_*^2}$$

to χ^2_{n-p} , and reject H_0 if this statistic is too large or too small. For a standard .05-level, two-tailed test, we would reject H_0 if the statistic was greater than $\chi^2_{.975,n-p}$ or less than $\chi^2_{.025,n-p}$. A 95% confidence interval for σ^2 is

$$\left[\frac{(n-p)S^2}{\chi^2_{.975,n-p}}, \frac{(n-p)S^2}{\chi^2_{.025,n-p}} \right].$$

The performance of this test and interval may suffer if the assumption of normality is violated.

In practice, a regression analysis rarely involves tests and intervals for σ^2 , because this parameter is usually regarded as a nuisance. We are usually more interested in drawing inferences about the coefficients in β .

Inferences about individual elements of β . Notice that the OLS estimate for β ,

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

has the form $\hat{\beta} = Ay$ where $A = (X^T X)^{-1} X^T$ is a $p \times n$ matrix of constants. This, combined with the fact that

$$y \sim N(X\beta, \sigma^2 I),$$

implies that $\hat{\beta}$ is multivariate normal with mean

$$\begin{aligned} E(\hat{\beta}) &= A E(y) \\ &= (X^T X)^{-1} X^T X \beta \\ &= \beta \end{aligned}$$

and covariance matrix

$$\begin{aligned} V(\hat{\beta}) &= A(\sigma^2 I)A^T \\ &= \sigma^2 AA^T \end{aligned} \tag{5}$$

$$\begin{aligned} &= \sigma^2 (X^T X)^{-1} X^T X (X^T X)^{-1} \\ &= \sigma^2 (X^T X)^{-1}. \end{aligned} \tag{6}$$

(To go from (5) to (6), we have used the fact that $(BC)^T = C^T B^T$ and the fact that $(X^T X)^{-1}$ is symmetric.) Therefore,

$$\hat{\beta} \sim N(\beta, \sigma^2 (X^T X)^{-1}). \tag{7}$$

The first part (1) of our Fundamental Theorem of Regression can be deduced from (7). Recall from Lecture 4 that, if a $p \times 1$ vector Y has a multivariate normal distribution with mean μ and covariance matrix Σ , then

$$(Y - \mu)^T \Sigma^{-1} (Y - \mu) \sim \chi_p^2.$$

Using this fact, (7) implies that

$$\frac{1}{\sigma^2} (\hat{\beta} - \beta)^T X^T X (\hat{\beta} - \beta) \sim \chi_p^2. \tag{8}$$

But

$$\begin{aligned} (X\hat{\beta} - X\beta)^T (X\hat{\beta} - X\beta) &= (\hat{\mu} - \mu)^T (\hat{\mu} - \mu) \\ &= \|\hat{\mu} - \mu\|^2, \end{aligned}$$

and thus (8) is equivalent to (1).

Result (7) can be used to construct tests and confidence intervals about individual coefficients. The j th element of

$$\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_p)^T$$

is distributed as

$$\hat{\beta}_j \sim N(\beta_j, \sigma^2(X^T X)_{jj}^{-1}),$$

where $(X^T X)_{jj}^{-1}$ denotes the (j, j) th element of $(X^T X)^{-1}$.

Therefore,

$$\frac{\hat{\beta}_j - \beta_j}{\sqrt{\sigma^2(X^T X)_{jj}^{-1}}} \sim N(0, 1). \quad (9)$$

If we divide (9) by

$$\sqrt{\frac{S^2}{\sigma^2}} \sim \sqrt{\frac{\chi^2_{n-p}}{(n-p)}},$$

we get

$$\frac{\hat{\beta}_j - \beta_j}{\sqrt{S^2(X^T X)_{jj}^{-1}}} \sim t_{n-p}. \quad (10)$$

Therefore, a 95% confidence interval for β_j is

$$\hat{\beta}_j \pm t_{.975, n-p} SE(\hat{\beta}_j),$$

where

$$SE(\hat{\beta}_j) = \sqrt{S^2(X^T X)_{jj}^{-1}}$$

is the standard error for $\hat{\beta}_j$.

The result (10) can also be used to test hypotheses about β_j . In nearly all cases, the null hypothesis of interest is

$H_0 : \beta_j = 0$. This null hypothesis says that the j th predictor, X_j , is unnecessary and can be omitted from the model. Under this null hypothesis, the statistic

$$t = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)} \quad (11)$$

is distributed as t_{n-p} . We would reject H_0 in a .05-level, two-tailed test if $|t|$ exceeds the 97.5th percentile of t_{n-p} . (When $n - p$ is large, this percentile is approximately 2.) More generally, the p-value for this two-tailed test is twice the area to the right of $|t|$ under the t_{n-p} density,

$$p = 2 P(t_{n-p} \geq |t|).$$

Linear regression in R using lm. The most popular way to fit a linear regression model in R is to use the function `lm`, which operates on variables in a data frame. We will demonstrate the use of `lm` by example.

In Assignment 2, we examined body measurements from a sample of $n = 1,835$ adults. We used the file `body.dat`, which contained the following five columns.

HEIGHT	standing height without shoes (inches)
WEIGHT	body weight (pounds)
WAIST	waist circumference (inches)
HIPS	hips circumference (inches)
CHOL	total serum cholesterol (mg/dl)

Let's read in the data and create two more variables: body mass index,

$$\text{BMI} = \frac{\text{weight in kilograms}}{(\text{height in meters})^2},$$

and our measure of body shape,

$$\text{MORPH} = \frac{\text{WAIST}}{\text{HIPS}}.$$

```
> body <- read.table("body.dat", header=T) # read in data
> names(body) # see the variable names
[1] "height" "weight" "waist" "hips" "chol"

> meters <- body$height * 2.54 / 100 # height in meters
> kg <- body$weight * 0.45359237 # weight in pounds
> body$bmi <- kg / meters^2 # put bmi into the data frame
> body$morph <- body$waist / body$hips # put morph into the data frame
```

Now let's fit the regression model that says

$$\text{CHOL}_i \sim N(\mu_i, \sigma^2),$$

where

$$\mu_i = \beta_0 + \beta_1 \text{BMI}_i + \beta_2 \text{MORPH}_i.$$

The basic syntax is

```
lm( formula, data=...)
```

where ... is the name of the data frame, and **formula** is a model formula involving variables in the data frame. Our model would be specified as

```
chol ~ bmi + morph
```

and the constant is included by default. The **summary**

function, when applied to the result of `lm`, prints out a nicely formatted summary of the results.

```
> result <- lm( chol ~ bmi + morph, data=body )
> summary( result )

Call:
lm(formula = chol ~ bmi + morph, data = body)

Residuals:
    Min      1Q  Median      3Q     Max 
-125.825 -30.146 - 3.667  24.883 266.235 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 71.8942   10.5007   6.847 1.03e-11 ***
bmi         0.2226    0.1724   1.292   0.197    
morph       137.9966   12.6014  10.951 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 43.43 on 1832 degrees of freedom
Multiple R-Squared: 0.07813,    Adjusted R-squared: 0.07713 
F-statistic: 77.63 on 2 and 1832 DF,  p-value: < 2.2e-16
```

The most important part of the output is the table of coefficients. Any regression routines in any major statistical package will produce this. This table has one row for each coefficient β_j . The first column is the estimate $\hat{\beta}_j$. The second column is the standard error $SE(\hat{\beta}_j)$. The third column is the t -statistic (11) for testing the null hypothesis $H_0 : \beta_j = 0$, and the fourth column is the p-value for testing $H_0 : \beta_j = 0$ against the two-tailed alternative.

Beside the p-values are symbols indicating which of the null hypotheses can be rejected at various levels

(.01, .05, .01, ...). In this case, the coefficient for MORPH is significantly different from zero, but the coefficient for BMI is not. (The intercept is also significantly different from zero, but that test is rarely of interest.)

Another important part of the output is this line:

```
Residual standard error: 43.43 on 1832 degrees of freedom
```

This line reports the residual standard error S and its degrees of freedom $n - p$.

What does a call to `lm` produce? The result of a call to `lm` is a list. A list is a collection of data objects that may be of any storage mode (numeric, logical, character) and dimensions. To see what objects are part of the list, you can use the `names` function.

```
> names(result)
[1] "coefficients"   "residuals"      "effects"       "rank"
[5] "fitted.values"  "assign"        "qr"           "df.residual"
[9] "xlevels"        "call"          "terms"        "model"
```

The first component of the list, `coefficients`, is the vector $\hat{\beta}$. To access components of the list, use the `$` operator.

```
> result$coefficients
(Intercept)      bmi      morph
71.8942477  0.2226309 137.9966500
```

The component `fitted.values` contains the vector $\hat{y} = X\hat{\beta}$, and `residuals` contains the vector $\hat{\epsilon} = y - \hat{y}$. The length of these vectors is $n = 1,835$, so we will not display them here. The component `df.residual` is the integer

$n - p$, and `rank` is the number of linearly independent columns of X .

```
> result$df.residual  
[1] 1832  
> result$rank  
[1] 3
```

The other components of the list are not so important to us right now. But, as a whole, this list is useful because it serves as input to other functions. R has many functions that extract important information. One such function is `summary`, which we have already applied. In the lectures ahead, we will learn about other functions.

Linear regression in R using `lsfit`. Another R function for linear regression is `lsfit`. It is much older than `lm`, going back to the earliest days of the S language. (S is the language upon which the R package is built.) The function `lsfit` existed before the data frame, so it operates on numeric vectors and matrices rather than data frames. (A matrix is a two-dimensional array whose elements are all the same type of data. A data frame is more general than a matrix, because it may have columns of different types.)

The first required argument to this function, `x`, is a matrix whose columns are the predictor variables. (By default, `lsfit` includes intercept, so the actual X matrix will be a column of ones, followed by the first column of `x`, followed by the second column of `x`, and so on.) The second

required argument, `y`, is the vector of responses. There is an optional argument `intercept`, a logical variable whose default value is `T`. If you don't want `lsfit` to append a column of ones to the predictor matrix, put `intercept=F` in the arguments.

The result of `lsfit` is a list similar to that produced by `lm` but with fewer components. The function `ls.print`, when applied to this list, will print out a brief summary of the results.

```
> y <- body$chol
> x <- cbind( bmi=body$bmi, morph=body$morph ) # bind the columns into a matrix
> result <- lsfit( x, y )

> names(result)
[1] "coefficients" "residuals"      "intercept"       "qr"

> ls.print( result )
Residual Standard Error=43.4341
R-Square=0.0781
F-statistic (df=2, 1832)=77.6348
p-value=0

          Estimate Std.Err t-value Pr(>|t|)
Intercept 71.8942 10.5007 6.8466 0.0000
bmi        0.2226  0.1724 1.2916 0.1966
morph     137.9966 12.6014 10.9509 0.0000
```

The main difference between `lm` and `lsfit` is:

- In `lm`, the model is specified by a formula, and the software creates the X matrix.
- In `lsfit`, the user supplies the X matrix directly (except possibly for the column of ones.)

If all the terms in the model formula are numeric variables, then these two functions do essentially the same thing. But variables in a data frame are not necessarily numeric. In particular, a variable in a data frame may be a **factor**. A factor is a categorical variable whose levels may be ordered (i.e. ordinal) or unordered (i.e. nominal). If one or more of the variables on the right-hand side of the model formula is a factor, then `lm` will automatically convert this variable into a set of dummy codes or contrasts to distinguish among its levels. (In SAS modeling procedures, this would be done by a `CLASS` statement.) The `lsfit` function, on the other hand, does not accept factors as predictors. So if you want to use a categorical variable as a predictor, you will have to create the dummy codes or contrasts yourself. We will learn how to do this in future lectures.

SIMPLE LINEAR REGRESSION, PART I

“Simple linear regression” is a model with one predictor. Many textbooks devote a great deal of space to this model—not because it is often used in practice (it isn’t), but because a thorough understanding of this model gives us excellent insights into the properties of the more general version with multiple predictors. In the next two lectures, we will quickly cover material in KNNL Chapters 1–2.

Simple linear regression, in the notation of the last lecture, is a regression model in which the X matrix has $p = 2$ columns, and the first one is a column of 1’s. The model is often written as

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad (1)$$

where $\epsilon_i \sim N(0, \sigma^2)$, or as

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2).$$

Note that this conflicts with the notation that we have been using. In the previous lectures, x_i represented the $p \times 1$ vector of predictors for unit i . Now we are using x_i to denote a single predictor for unit i . Both types of notation are common in statistics, however, and we will

use either one as the need arises.

Collecting the responses into a vector $y = (y_1, \dots, y_n)^T$, this model asserts that

$$y \sim N(\mu, \sigma^2 I),$$

and that μ lies in the linear space spanned by the two columns of

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} = [1, X_1].$$

Note that we are now using X_1 to denote the single predictor variable, even though it is actually the second column of X .

Interpretation of coefficients. The two coefficients can be interpreted as follows.

The intercept, β_0 , is **the expected value of the response y_i when $x_i = 0$** . It may happen that $x_i = 0$ lies outside the range of the observed values for x_i , and it may even be a physical impossibility. (Think of a regression of weight on height.) In such cases, the intercept will have little or no substantive meaning. If desired, the intercept can be made more meaningful by recoding the predictor. If we replace x_i by $x_i - c$, the intercept becomes the mean

response when $x_i = c$.

The slope, β_1 , is **the increase in the expected value of y_i associated with a one-unit increase in x_i** . If we compared the subpopulation of subjects with $x_i = c + 1$ to the subpopulation of subjects with $x_i = c$, the difference in the average responses between these two groups will be β_1 . Similarly, the difference in average response between subjects with $x_i = d$ and subjects with $x_i = c$ is $(d - c)\beta_1$. For these statements to be correct, of course, the true relationship between the mean of y_i and x_i needs to be linear.

When interpreting β_1 , we must be careful not to use the language of causality unless it is truly warranted. If the data come from a **randomized experiment** (i.e. if the values of x_i have been assigned to subjects in a random fashion), then the subjects receiving any specific value of x_i will, on average, be no different from those receiving any other value of x_i when the experiment begins. In that case, the parameter β_1 may be interpreted as the average causal effect on y_i resulting from a one-unit increase in x_i . However, when the data come from an **observational study** (i.e. when the values of x_i have not been randomly assigned by the investigators), the subjects having different values of x_i may be different in other ways as well. In observational studies, there may be confounders (variables associated with x_i that may also affect y_i) that could greatly distort the relationship between these two

variables. Causal inference from observational data is a complicated issue, and we will discuss it at length near the end of the semester. Until then, please **do not interpret regression coefficients as causal effects unless the data come from a truly randomized experiment.**

One very important case of the simple linear regression model arises when x_i is a binary indicator that separates the subjects into two groups. Suppose, for example, that x_i is a dummy indicator for sex,

$$x_i = \begin{cases} 0 & \text{if subject } i \text{ is male,} \\ 1 & \text{if subject } i \text{ is female.} \end{cases}$$

In this case, β_0 becomes the average response for males, and β_1 becomes the average response for females minus the average response for males. Inferences for β_1 under this model become identical to inferences about the difference between two means from a pooled two-sample t -test.

Deriving the least-squares estimates for simple linear regression. To derive the least-squares estimate of $\beta = (\beta_0, \beta_1)^T$, we could algebraically invert the $X^T X$ matrix and multiply the result by $X^T y$. In this case, $X^T X$ is

$$X^T X = \begin{bmatrix} n & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{bmatrix},$$

and $X^T y$ is

$$X^T y = \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}.$$

But a more elegant way is to transform this to an equivalent model in which the predictor x_i has been centered at its mean.

Suppose we transform the predictor to

$$x_i^* = x_i - \bar{x},$$

where $\bar{x} = \sum_i x_i/n$, and fit the model

$$y_i = \beta_0^* + \beta_1^* x_i^* + \epsilon_i. \quad (2)$$

In vector notation, the centered model is

$$y \sim N(X^* \beta^*, \sigma^2),$$

where $\beta^* = (\beta_0^*, \beta_1^*)^T$ and

$$X^* = \begin{bmatrix} 1 & x_1^* \\ 1 & x_2^* \\ \vdots & \vdots \\ 1 & x_n^* \end{bmatrix} = \begin{bmatrix} 1 & (x_1 - \bar{x}) \\ 1 & (x_2 - \bar{x}) \\ \vdots & \vdots \\ 1 & (x_n - \bar{x}) \end{bmatrix}.$$

Notice that any linear combination of 1 and x_i^* is also a linear combination of 1 and x_i ,

$$a 1 + b x_i^* = (a - b \bar{x}) 1 + b x_i.$$

Therefore, the two columns of X^* span the same linear space as the two columns of X , and the centered model will give the same predicted values \hat{y}_i as the original model. The two models are essentially the same; the only difference is that the coefficients have been transformed. Comparing (2) to (1), we see that the two models are equivalent, with

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0^* - \bar{x}\beta_1^* \\ \beta_1^* \end{bmatrix} = A\beta^*,$$

where

$$A = \begin{bmatrix} 1 & -\bar{x} \\ 0 & 1 \end{bmatrix}.$$

Similarly, $\beta^* = A^{-1}\beta$, where

$$A^{-1} = \begin{bmatrix} 1 & \bar{x} \\ 0 & 1 \end{bmatrix}.$$

The slope β_1^* in the centered model is identical to the slope β_1 in the original model; it is the change in the average value of y_i associated with a one-unit increase in x_i . But the intercept β_0^* is now the average value of y_i when x_i is equal to its sample mean.

The least-squares estimates for the centered model are

$$\hat{\beta}^* = (X^{*T} X^*)^{-1} X^{*T} y.$$

But the matrix $X^{*T} X^*$ is

$$X^{*T} X^* = \begin{bmatrix} \sum_i 1 & \sum_i (x_i - \bar{x}) \\ \sum_i (x_i - \bar{x}) & \sum_i (x_i - \bar{x})^2 \end{bmatrix} = \begin{bmatrix} n & 0 \\ 0 & S_{xx} \end{bmatrix},$$

where $S_{xx} = \sum_i (x_i - \bar{x})^2$. (The fact that this matrix is diagonal indicates that the columns of X^* are orthogonal. And a column is orthogonal to $1 = (1, \dots, 1)^T$ if the variable has a mean of zero.) The inverse of this diagonal matrix is

$$(X^{*T} X^*)^{-1} = \begin{bmatrix} n^{-1} & 0 \\ 0 & S_{xx}^{-1} \end{bmatrix}.$$

Multiplying the latter by

$$X^{*T} y = \begin{bmatrix} \sum_i y_i \\ \sum_i (x_i - \bar{x})y_i \end{bmatrix}$$

gives

$$\hat{\beta}^* = \begin{bmatrix} \hat{\beta}_0^* \\ \hat{\beta}_1^* \end{bmatrix} = \begin{bmatrix} \bar{y} \\ S_{xy}/S_{xx} \end{bmatrix},$$

where $\bar{y} = \sum_i y_i/n$ and $S_{xy} = \sum_i (x_i - \bar{x})y_i$. The corresponding estimates for the original model are

$$\begin{aligned} \hat{\beta} &= A \hat{\beta}^* \\ &= \begin{bmatrix} 1 & -\bar{x} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{y} \\ S_{xy}/S_{xx} \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} \bar{y} - (S_{xy}/S_{xx})\bar{x} \\ S_{xy}/S_{xx} \end{bmatrix}.$$

Therefore, the least-squares coefficients for the simple linear regression model are

$$\begin{aligned}\hat{\beta}_1 &= S_{xy}/S_{xx}, \\ \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x}.\end{aligned}$$

Notice that

$$\begin{aligned}\sum_i (x_i - \bar{x})(y_i - \bar{y}) &= S_{xy} - \bar{y} \sum_i (x_i - \bar{x}) \\ &= S_{xy},\end{aligned}$$

so the estimated slope can also be written as

$$\begin{aligned}\hat{\beta}_1 &= \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} \\ &= \frac{\text{sample covariance of } x_i \text{ and } y_i}{\text{sample variance of } x_i}.\end{aligned}$$

It can also be written as

$$\hat{\beta}_1 = r \times \left(\frac{\text{sample standard deviation of } y_i}{\text{sample standard deviation of } x_i} \right), \quad (3)$$

where r is the sample correlation between x_i and y_i .

This last formula (3) is intuitively very sensible. Recall from Lecture 4 that the population correlation coefficient ρ could be interpreted as “the slope of the regression of y_i on

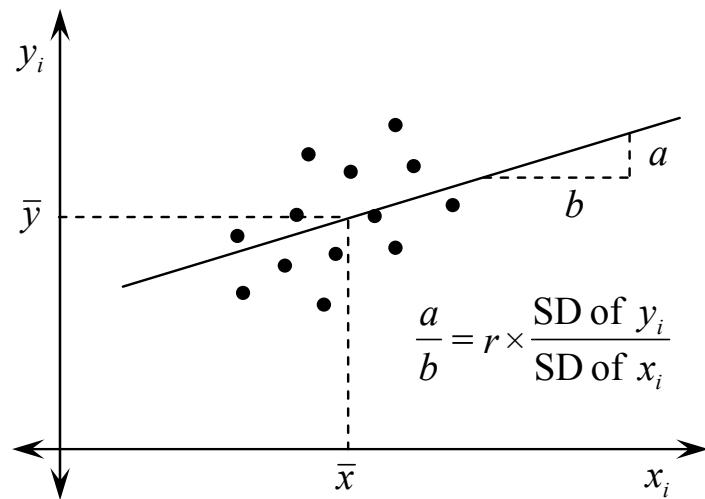
x_i when both variables are expressed in standard units.” “Standard units” refers to the number of standard deviations above or below the mean. If x_i and y_i were both expressed in standard units, then each would have a standard deviation of one, and the slope of the least-squares regression line would just be the dimensionless measure r . If x_i and y_i are expressed on any other scale, the slope β_1 must be expressed in terms of the units of y_i divided by the units of x_i . The factors that convert standard units to the units in which x_i and y_i are actually measured are the standard deviations of the two variables.

The least-squares estimate of the intercept, $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$, ensures that the prediction equation

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

predicts an average response of $\hat{y}_i = \bar{y}$ when $x_i = \bar{x}$. That is, it ensures that the estimated regression line on the (x_i, y_i) plane passes through the point of averages (\bar{x}, \bar{y}) . Thus we have an easy way to understand these formulas.

The least-squares regression line passes through the point of averages, (\bar{x}, \bar{y}) , and has slope equal to the sample correlation coefficient, times the standard deviation of y_i , divided by the standard deviation of x_i .



Therefore, if you are given a **five-number summary** of a bivariate dataset, which consists of

- the two sample means,
- the sample standard deviations (or variances), and
- the sample correlation,

you are able to obtain the least-squares slope and intercept.

The residual variance. The third parameter of the simple linear regression model, the residual variance σ^2 , can be estimated as follows. First, compute the predicted value

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

for each subject $i = 1, \dots, n$. Next, compute the residuals

$$\hat{\epsilon}_i = y_i - \hat{y}_i$$

for $i = 1, \dots, n$. Next, compute the sum of the squared

residuals,

$$\|\hat{\epsilon}\|^2 = \hat{\epsilon}^T \hat{\epsilon} = \sum_i \hat{\epsilon}_i^2,$$

which is also known as the **residual sum of squares** or **error sum of squares**, and is sometimes denoted by

$$SS_{Err} = \sum_i (y_i - \hat{y}_i)^2.$$

Finally, divide the residual sum of squares by the residual degrees of freedom, $n - p$, which in this case is $n - 2$. The resulting unbiased estimate of σ^2 is

$$S^2 = \frac{SS_{Err}}{n - p} = \frac{1}{n - 2} \sum_i (y_i - \hat{y}_i)^2.$$

As we noted last time, this estimate is called the **mean squared error**, and its square root, S , is sometimes called the **residual standard error**.

To compute S^2 by the method above without a computer is tedious. Moreover, it presumes that we have access to all of the data points (x_i, y_i) , $i = 1, \dots, n$. There is an easier way to compute this quantity from the five-number summary, using an orthogonal decomposition of the variation among the y_i 's. Define the **total sum of squares** as the sum of the squared deviations of y_i from their average,

$$SS_{Tot} = \sum_i (y_i - \bar{y})^2.$$

Notice that

$$\text{sample variance of } y_i = \frac{SS_{Tot}}{n - 1},$$

so we can easily get SS_{Tot} if we are given the sample variance or standard deviation of y_i . Then we can apply the formula

$$SS_{Err} = (1 - r^2) \times SS_{Tot}, \quad (4)$$

where r is the sample correlation between x_i and y_i .

A justification for this formula (4) will be given in the next lecture. This formula makes intuitive sense. Recall from Lecture 4 that **the squared population correlation coefficient can be interpreted as the proportion of variance in the response explained by the predictor.** SS_{Tot} measures the total variation among the y_i 's, and SS_{Err} measures the variation among the responses that has not been explained by the x_i 's. Therefore, (4) is saying that the squared sample correlation, r^2 , is the proportion of SS_{Tot} that has been explained by the predictor.

Standard errors for the coefficients. The covariance matrix for the coefficients of the centered model is

$$V(\hat{\beta}^*) = \sigma^2 (X^{*T} X^*)^{-1} = \begin{bmatrix} \sigma^2/n & 0 \\ 0 & \sigma^2/S_{xx} \end{bmatrix}.$$

But because $\hat{\beta}_0 = \hat{\beta}_0^* - \bar{x}\hat{\beta}_1^*$ and $\hat{\beta}_1 = \hat{\beta}_1^*$, it follows that

$$\begin{aligned} V(\hat{\beta}_0) &= \sigma^2/n + \bar{x}^2/\sigma^2 S_{xx}, \\ V(\hat{\beta}_1) &= \sigma^2/S_{xx}, \\ Cov(\hat{\beta}_0, \hat{\beta}_1) &= -\bar{x}\sigma^2/S_{xx}. \end{aligned}$$

Therefore, the covariance matrix for $\hat{\beta}$ is

$$V(\hat{\beta}) = \sigma^2 \begin{bmatrix} 1/n + \bar{x}^2/S_{xx} & -\bar{x}/S_{xx} \\ -\bar{x}/S_{xx} & 1/S_{xx} \end{bmatrix},$$

and its unbiased estimate is

$$\hat{V}(\hat{\beta}) = S^2 \begin{bmatrix} 1/n + \bar{x}^2/S_{xx} & -\bar{x}/S_{xx} \\ -\bar{x}/S_{xx} & 1/S_{xx} \end{bmatrix}.$$

The standard errors are

$$\begin{aligned} SE(\hat{\beta}_0) &= \sqrt{S^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right)}, \\ SE(\hat{\beta}_1) &= \sqrt{S^2 \left(\frac{1}{S_{xx}} \right)}. \end{aligned}$$

We'll have more to say about these standard errors in the next lecture.

Example in R. Let's go back to the blood pressure dataset that we first examined in Lecture 2, and let's use R to fit the simple linear regression

$$BPSYS_i = \beta_0 + \beta_1 BPDIAS_i + \epsilon_i$$

in a couple of different ways. First, let's use `lm`.

```
> bp <- read.table("bp.dat", header=T)
> names(bp)
[1] "BPSYS"   "BPDIAS"
> result <- lm( BPSYS ~ BPDIAS, data=bp)
> summary( result )

Call:
lm(formula = BPSYS ~ BPDIAS, data = bp)

Residuals:
    Min      1Q  Median      3Q     Max 
-34.111 -8.652 -2.196  8.233 45.604 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 50.56202   4.55329   11.11  <2e-16 ***
BPDIAS       0.97149   0.05358   18.13  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.34 on 317 degrees of freedom
Multiple R-Squared: 0.5091,    Adjusted R-squared: 0.5076 
F-statistic: 328.8 on 1 and 317 DF,  p-value: < 2.2e-16
```

From the output, we see that the three parameter estimates are

$$\begin{aligned}\hat{\beta}_0 &= 50.56202, \\ \hat{\beta}_1 &= 0.97149, \\ S^2 &= (13.34)^2 = 178.0.\end{aligned}$$

These estimates have been rounded for display purposes. We can get the full double-precision values by extracting them from the components of the list `result`.

```
> names(result)
[1] "coefficients" "residuals"      "effects"        "rank"
```

```
[5] "fitted.values" "assign"           "qr"          "df.residual"
[9] "xlevels"       "call"            "terms"        "model"

> betahat <- result$coefficients
> betahat
(Intercept)      BPDIAS
50.5620199    0.9714949

> eps.hat <- sum( result$residuals^2 ) / result$df.residual
> eps.hat
[1] 177.8271
```

(In a real analysis, we would rarely need to report parameter estimates beyond three significant digits. Here we are doing so merely to check that our formulas are working properly.)

Now let's obtain these estimates from a five-number summary.

```
> n <- nrow( bp )
> xbar <- mean( bp$BPDIAS )
> ybar <- mean( bp$BPSYS )
> var.x <- var( bp$BPDIAS )
> var.y <- var( bp$BPSYS )
> r <- cor( bp$BPSYS, bp$BPDIAS )

> beta.1 <- r * sqrt( var.y ) / sqrt( var.x )
> beta.0 <- ybar - beta.1 * xbar
> SS.Tot <- (n-1) * var.y
> SS.Err <- (1-r^2) * SS.Tot
> S2 <- SS.Err / (n-2)

> beta.0
[1] 50.56202
> beta.1
[1] 0.9714949
> S2
[1] 177.8271
```

SIMPLE LINEAR REGRESSION, PART II

Recap. In the last lecture, we found the least-squares estimates of the coefficients from the simple linear regression model and derived their standard errors. The model may be written as

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$$

for $i = 1, \dots, n$. In vector notation, the model is

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}),$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \quad \text{and} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}.$$

We found that the least-squares estimates are

$$\hat{\beta}_1 = S_{xy}/S_{xx},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

where

$$S_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}),$$

$$S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2.$$

On the (x_i, y_i) plane, the least-squares regression line

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

passes through the point of averages, (\bar{x}, \bar{y}) , and has slope equal to

$$\hat{\beta}_1 = r \times \left(\frac{\text{standard deviation of } y_i}{\text{standard deviation of } x_i} \right).$$

The covariance matrix for the estimated coefficients is

$$V(\hat{\beta}) = \sigma^2 (X^T X)^{-1} = \sigma^2 \begin{bmatrix} 1/n + \bar{x}^2/S_{xx} & -\bar{x}/S_{xx} \\ -\bar{x}/S_{xx} & 1/S_{xx} \end{bmatrix}.$$

To estimate this covariance matrix, we replace σ^2 by its unbiased estimate

$$S^2 = \frac{SS_{Err}}{n-2},$$

where

$$SS_{Err} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = (1 - r^2) SS_{Tot}$$

and

$$SS_{Tot} = \sum_{i=1}^n (y_i - \bar{y})^2.$$

Precision of the estimated slope. The formula

$$V(\hat{\beta}_1) = \sigma^2 / S_{xx}$$

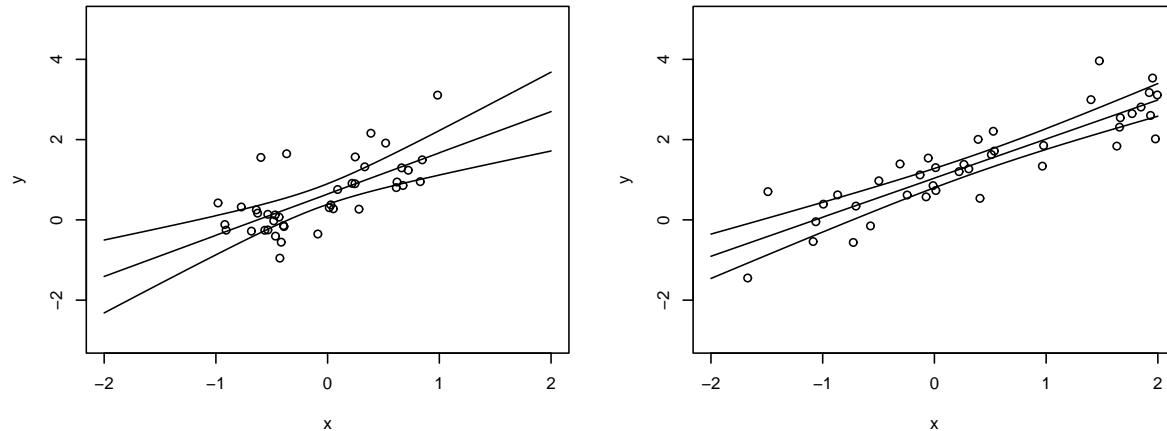
shows that the estimated slope becomes more precise as the variance of the x_i 's increases. In a typical observational study, investigators have little or no control over how the x_i 's are distributed. In a controlled experiment, however, we may have the opportunity to select the values of x_i at which the y_i 's are measured. If the goal is to determine whether and how y_i varies with x_i , then it is advantageous to make the values of x_i widely dispersed. The chosen values of x_i are sometimes called **design points**.

To illustrate, I generated $n = 40$ observations from the linear regression model

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$$

with $\beta_0 = 1$, $\beta_1 = 1$ and $\sigma^2 = 0.49$. But I used two different sets of design points. First, I drew the x_i 's from a uniform distribution on the interval $(-1, 1)$. Second, I used a uniform distribution on the interval $(-2, 2)$. The plots below show the two samples and the estimated regression lines. Each plot also includes a 95% confidence band for the whole regression line. This is a region which, over repeated samples, will cover the true regression line with

probability 0.95. (We will learn how to construct this region later.)



The region for the second sample is smaller than the region for the first sample, because the second estimate of the slope is more precise.

Suppose that the x_i 's are restricted to lie within the interval $a \leq x_i \leq b$. For a fixed value of n , the variance among the x_i 's will be maximized by taking half of the x_i 's equal to a and the other half equal to b . This is the most efficient design for estimating β_1 . This design, however, has one important disadvantage. If we collect data at only two design points, we will have no way to evaluate whether the effect of x_i on y_i is truly linear. In practice, it makes sense to collect at least a few responses at intermediate values of x_i , so that the assumption of linearity can be evaluated.

Standard error for prediction at a specific value of x_i . Recall that $\hat{\beta}_0$ is the estimated mean response when

$x_i = 0$. This quantity is not so meaningful when zero lies outside the range of plausible values for x_i . More generally, the estimated mean response at any particular value $x_i = x$ is

$$\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x.$$

The variance of this estimated mean is

$$\begin{aligned} V(\hat{y}(x)) &= V(\hat{\beta}_0) + x^2 V(\hat{\beta}_1) + 2x \operatorname{Cov}(\hat{\beta}_0, \hat{\beta}_1) \\ &= \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} + \frac{x^2}{S_{xx}} - 2 \frac{x\bar{x}}{S_{xx}} \right] \\ &= \sigma^2 \left[\frac{1}{n} + \frac{(x - \bar{x})^2}{S_{xx}} \right]. \end{aligned}$$

The standard error of the estimate is

$$SE(\hat{y}(x)) = S \sqrt{\frac{1}{n} + \frac{(x - \bar{x})^2}{S_{xx}}}.$$

The variance reaches its minimum at $x = \bar{x}$, at which

$$V(\hat{y}(\bar{x})) = \sigma^2/n.$$

Recall that the regression line passes through the point of averages. It is an algebraic fact that $\hat{y}(\bar{x}) = \bar{y}$, so the variance of $\hat{y}(x)$ at $x = \bar{x}$ must be $V(\bar{y}) = \sigma^2/n$. As x moves away from \bar{x} , the predictions become less precise. This illustrates a general principle about regression: **The regression line is best estimated at the average value of the predictor and poorly estimated for extreme values of the predictor.**

Confidence interval for mean response. A 95% confidence interval for the mean response when $x_i = x$ is

$$\hat{y}(x) \pm t_{.975,n-2} SE(\hat{y}(x)). \quad (1)$$

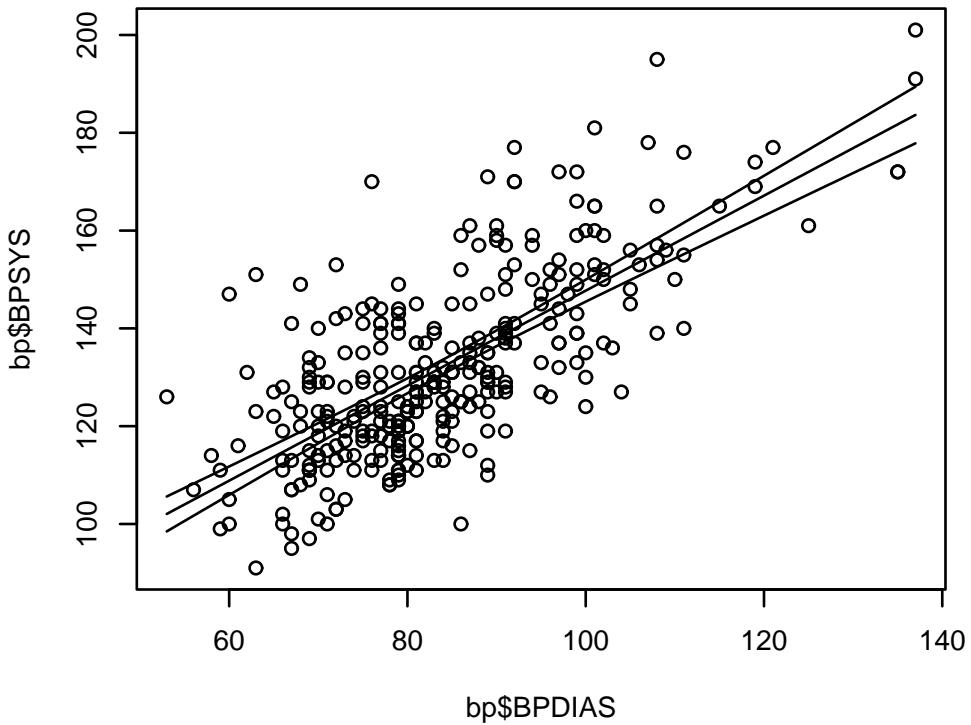
If the simple linear regression model holds, then this interval will capture the true value of $E(y_i | x_i = x)$ with probability .95.

Suppose that we compute $\hat{y}(x)$ and the limits of the 95% confidence interval (1) for a sequence of x values that span the range of the x_i 's. We can then plot the regression line and the confidence intervals as lines on the (x, y) plane. Let's do this with our blood pressure dataset, for the regression of systolic on diastolic.

```
> result <- lm( BPSYS ~ BPDIAS, data=bp )
> beta.0 <- result$coef[1] # estimated intercept
> beta.1 <- result$coef[2] # estimated slope
> S2 <- sum( result$residuals^2 ) / result$df.residual # MSE

> n <- nrow(bp) # sample size
> x <- seq( from=min(bp$BPDIAS), to=max(bp$BPDIAS), length=200) # grid of x's
> xbar <- mean( bp$BPDIAS ) # mean of observed x's
> Sxx <- (n-1) * var( bp$BPDIAS )
> yhat <- beta.0 + beta.1 * x
> se.yhat <- sqrt( S2 * ( 1/n + ( x - xbar )^2 / Sxx ) )
> lower <- yhat - qt(.975,n-2) * se.yhat
> upper <- yhat + qt(.975,n-2) * se.yhat

> plot( bp$BPDIAS, bp$BPSYS ) # scatterplot of data
> lines( x, yhat )
> lines( x, lower )
> lines( x, upper )
```



The upper and lower lines on this plot represent the 95% confidence intervals for the mean response at various values of the predictor. These may be called **pointwise confidence intervals for the mean response**. At any single value x , the interval will, over repeated samples, cover the true value of the mean response with probability .95.

It is important to note, however, that region enclosed by the upper and lower limits does not cover the mean response with probability 95% **at multiple values of x** . Whenever we construct multiple confidence intervals, the probability of making at least one error rapidly

accumulates. For example, if we construct ten independent 95% intervals, the probability that all ten of them will capture their target parameters is only $(.95)^{10} \approx 60\%$. With independent intervals, it would be easy to adjust the overall confidence level of each interval to maintain a given rate of error overall. But in our regression analysis, a single dataset has been used to construct intervals at all the values of x . Constructing a region that encloses the whole regression line with a given probability is a bit tricky. We will discuss that problem later.

Prediction interval for a future response. We have just described how to construct a confidence interval for the mean response at any particular value of the predictor. A different, but related, problem is to construct a prediction interval for a future response.

Suppose we want to predict a future observation y_i^* at a given design point $x_i = x^*$. Assume that y^* is independent of the current dataset y_1, \dots, y_n but follows the same model, so that

$$y^* \sim N(\beta_0 + \beta_1 x^*, \sigma^2).$$

The estimated mean of y^* is $\hat{y}(x^*) = \hat{\beta}_0 + \hat{\beta}_1 x^*$, with variance

$$V(\hat{y}(x^*)) = \sigma^2 \left[\frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{xx}} \right].$$

The error of prediction, $y^* - \hat{y}(x^*)$, is the difference between two independent normal random variables.

Therefore, it is normally distributed with mean

$$\begin{aligned} E(y^* - \hat{y}(x^*)) &= E(y^*) - E(\hat{y}(x^*)) \\ &= (\beta_0 + \beta_1 x^*) - (\beta_0 + \beta_1 x^*) \\ &= 0 \end{aligned}$$

and variance

$$\begin{aligned} V(y^* - \hat{y}(x^*)) &= V(y^*) + V(\hat{y}(x^*)) \\ &= \sigma^2 + \sigma^2 \left[\frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{xx}} \right] \\ &= \sigma^2 \left[1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{xx}} \right]. \end{aligned}$$

The estimated square root of this quantity,

$$SE(y^*) = S \sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{S_{xx}}},$$

is called the **standard error of prediction**.

A 95% prediction interval for the future response is

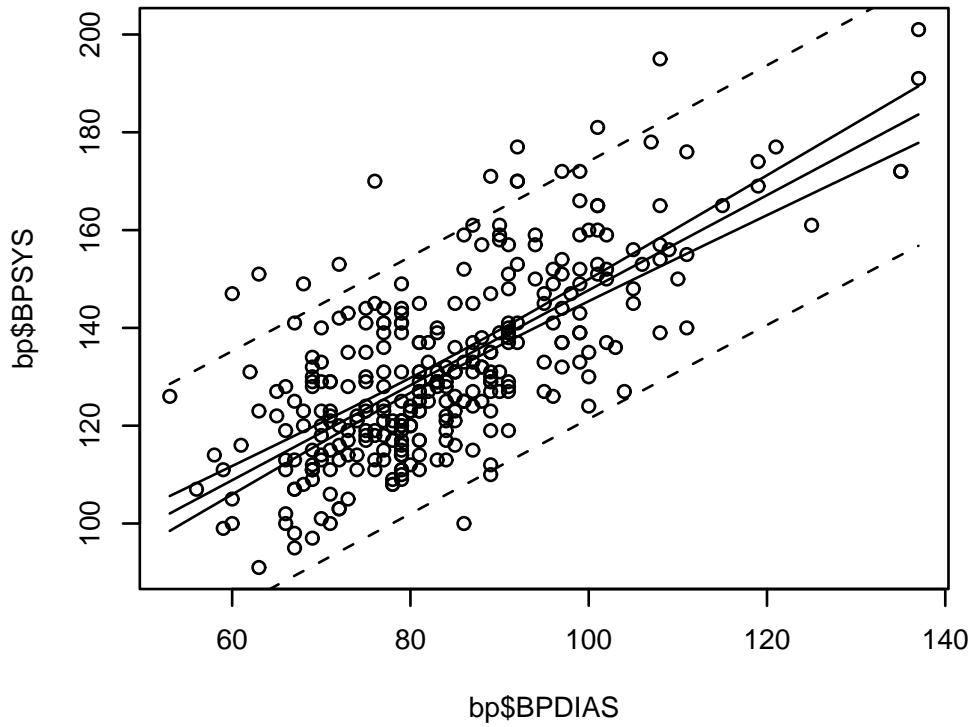
$$\hat{y}(x) \pm t_{.975, n-2} SE(y^*). \quad (2)$$

Be careful not to confuse this new interval (2) with the confidence interval for the mean response (1). The prediction interval is wider. The prediction interval is attempting to capture the future observation, whereas the confidence interval is attempting to capture its mean.

Once again, we can define a grid of x^* values and compute

the prediction interval at each value in the grid. Then we can add the prediction interval to the scatterplot, like this.

```
> se.pred <- sqrt( S2 * ( 1 + 1/n + ( x - xbar )^2 / Sxx ) )
> lower.pred <- yhat - qt(.975,n-2) * se.pred
> upper.pred <- yhat + qt(.975,n-2) * se.pred
> lines( x, lower.pred, lty=2 ) # the lty argument changes the line type
> lines( x, upper.pred, lty=2 ) # the lty argument changes the line type
```



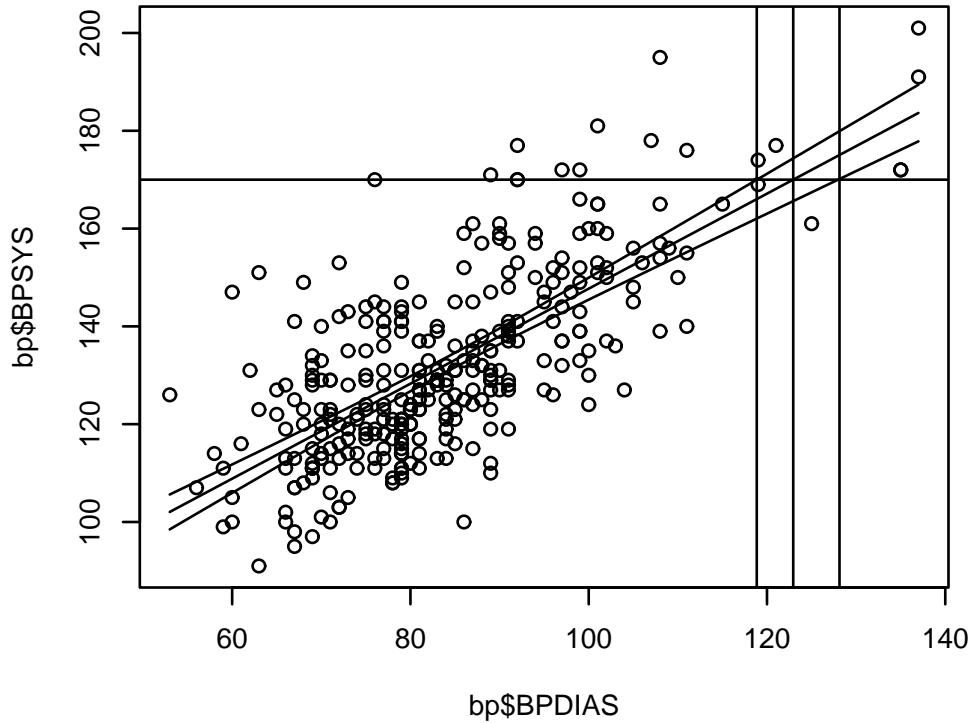
Again, it is important to realize that (2) will cover a single future response at $x = x^*$ with probability .95. But prediction intervals for multiple future responses computed in this fashion will not simultaneously cover their targets with probability .95. Prediction bands for simultaneous

coverage will be discussed in a future lecture.

Inverse prediction. On rare occasions, one may be led to ask: What is the value of the predictor that leads to a given predicted response? That is, we may need to estimate the value x^* for which the predicted response is y^* . This problem is known as “inverse prediction” or “calibration.” Substituting $\hat{y} = y^*$ into the prediction equation and solving for x gives

$$\hat{x}^* = \frac{y^* - \hat{\beta}_0}{\hat{\beta}_1}.$$

Because this estimate involves the ratio of the normal variates $\hat{\beta}_0$ and $\hat{\beta}_1$, its exact distribution is not nice. The mean of \hat{x}^* does not exist. If $\hat{\beta}_1$ is sufficiently far from zero, we can use a first-order Taylor expansion with respect to $\hat{\beta}$ to obtain a normal limiting distribution, and we can use that to construct an approximate confidence interval for the desired value of x . Or we can draw a horizontal line at $y = y^*$ on the scatterplot, find out where this line intersects the 95% confidence limits for the mean response (1), and project down to the x -axis to get an interval for x^* .

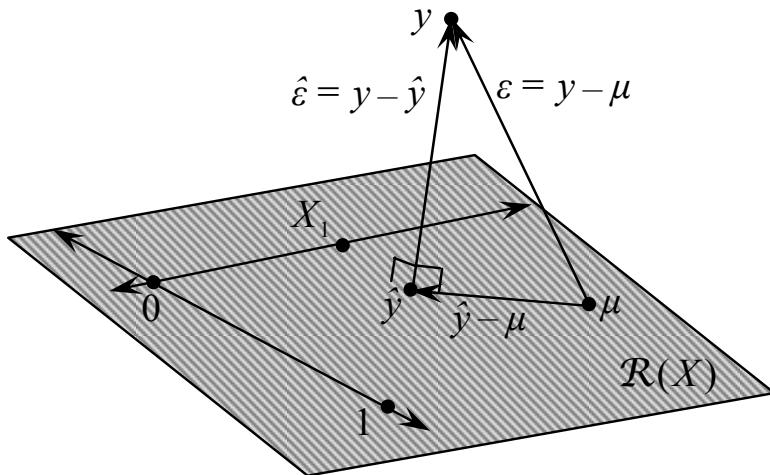


The occasions where this procedure is needed are rare. More often than not, if we need to predict x from y , it's more appropriate to reverse the roles of the two variables and regress x_i on y_i . The data needed for this regression are not different from the data needed for the regression of y_i on x_i .

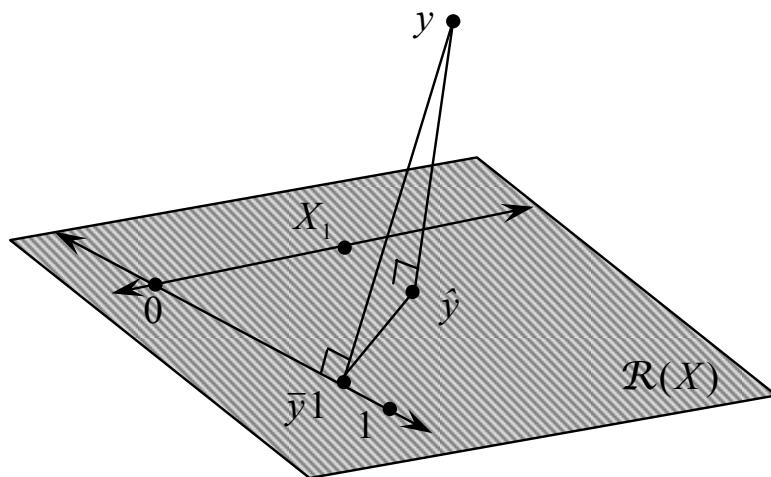
Simple linear regression and correlation. Back in Lecture 5, we presented a geometric interpretation of the sample correlation coefficient. And we showed how to test the null hypothesis $H_0 : \rho = 0$ by converting r to a t-statistic. This is equivalent to the t-test for $H_0 : \beta_1 = 0$

in simple linear regression, because $\beta_1 = 0$ if and only if $\rho = 0$.

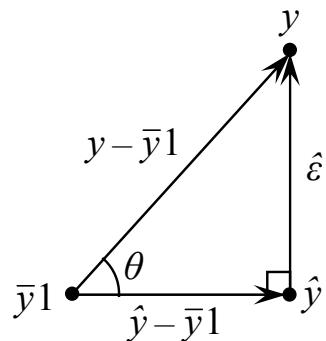
Now let's relate this to simple linear regression. In this model, we are projecting the vector $y = (y_1, \dots, y_n)^T$ onto the linear space (in this case, the plane) spanned by the two vectors, $1 = (1, \dots, 1)^T$ and $X_1 = (x_1, \dots, x_n)^T$. The projection is $\hat{y} = X\hat{\beta}$, and the true mean of y is $\mu = X\beta$. The vector of estimated residuals, $\hat{\epsilon}$, is the difference between y and \hat{y} . The vector of true residuals, ϵ , is the difference between y and μ . The theorem we learned in Lecture 9 concerns the right triangle that connects y , \hat{y} and μ .



But there is another triangle worth considering. If we project y onto $\mathcal{R}(1)$, we obtain $\bar{y}1$, the fitted values from the no-predictors model.



The triangle with vertices y , \hat{y} and \bar{y}_1 is also a right triangle. (We know this because $y - \hat{y}$ is orthogonal to the $\mathcal{R}(X)$ plane, and $\hat{y} - \bar{y}_1$ lies within the plane.)



The correlation coefficient r is the cosine of the angle shown in the picture above. Why is this so?
Previously, we have argued that r is the cosine of the angle between

- the difference between y and \bar{y}_1 , and
- the difference between X_1 and \bar{x}_1 .

But that angle is equal to the angle in the triangle above.

The reason why the two angles are the same is that $X_1 - \bar{x}$ also lies within the $\mathcal{R}(X)$ plane.

Simple linear regression and the analysis of variance. “Analysis of variance” has different meanings. In its most general sense, it is the basic idea of R.A. Fisher that the variation in a set of response measurements $y = (y_1, \dots, y_n)^T$ from a randomized experiment can be partitioned into parts that are explained by the factors in the experiment (i.e. the predictors) and a residual part (error) that is unexplained.

The triangle that connects y , \hat{y} and $\bar{y}1$ produces an analysis of variance for simple linear regression. Define the **total sum of squares** as

$$SS_{Tot} = \|y - \bar{y}1\|^2 = \sum_{i=1}^n (y_i - \bar{y})^2.$$

The **residual or error sum of squares** is

$$SS_{Err} = \|y - \hat{y}\|^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2.$$

Finally, define the **regression sum of squares** or **sum of squares due to x** as

$$SS_{Reg} = \|\hat{y} - \bar{y}1\|^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2.$$

The last quantity is an intuitive measure of how much of

the variation in y can be explained by x . A small value of SS_{Reg} indicates that \hat{y} is close to \bar{y}_1 . This means that the regression model with x gives nearly the same predictions as the no-predictors model, and thus x explains little of the variation in y . Conversely, a large value of SS_{Reg} means that x explains a lot.

The Pythagorean theorem says that

$$\| y - \bar{y}_1 \|^2 = \| \hat{y} - \bar{y}_1 \|^2 + \| y - \hat{y} \|^2,$$

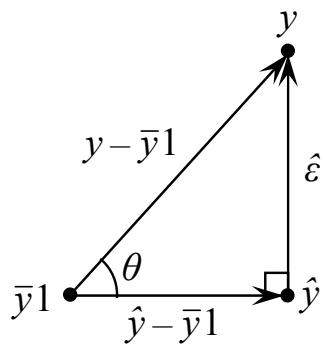
or that

$$SS_{Tot} = SS_{Reg} + SS_{Err}.$$

Next time, we will present a theorem about how the two quantities on the right-hand side are distributed.

SIMPLE LINEAR REGRESSION, PART III

Distributions of sums of squares in the simple linear regression model. Last time, we drew the triangle with vertices at y , \hat{y} and \bar{y}_1 ,



and we established that

$$\| y - \bar{y}_1 \|^2 = \| \hat{y} - \bar{y}_1 \|^2 + \| y - \hat{y} \|^2,$$

which may also be written as

$$SS_{Tot} = SS_{Reg} + SS_{Err}.$$

What can we say about the distributions of these quantities? In particular, what can we say about their expected values?

First, note that $y - \hat{y} = \hat{\epsilon}$ appeared in our “Fundamental

Theorem of Regression," which said that

$$\|\hat{\epsilon}\|^2 \sim \sigma^2 \chi_{n-p}^2.$$

In this case, $p = 2$. Therefore, SS_{Err} is distributed as $\sigma^2 \chi_{n-2}^2$, and its mean is

$$E(SS_{Err}) = \sigma^2(n - 2).$$

What about SS_{Reg} ? We can write it as

$$\sum_{i=1}^n (\hat{y}_i - \bar{y})^2 = \sum_{i=1}^n (\hat{\beta}_0 + \hat{\beta}_1 x_i - \bar{y})^2.$$

But $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$, so

$$SS_{Reg} = \sum_{i=1}^n (\bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 x_i - \bar{y})^2 = \hat{\beta}_1^2 S_{xx}.$$

But because $\hat{\beta}_1 \sim N(\beta_1, \sigma^2/S_{xx})$, we know that

$$E(\hat{\beta}_1^2) = V(\hat{\beta}_1) + (E(\hat{\beta}_1))^2 = \sigma^2/S_{xx} + \beta_1^2,$$

and thus

$$E(SS_{Reg}) = \sigma^2 + \hat{\beta}_1^2 S_{xx}.$$

If the null hypothesis $H_0 : \beta_1 = 0$ happens to be true, then $E(SS_{Reg}) = \sigma^2$. Moreover, in that case,

$$\frac{\hat{\beta}_1}{\sigma/\sqrt{S_{xx}}} \sim N(0, 1),$$

which implies that

$$\frac{S_{xx} \hat{\beta}_1^2}{\sigma^2} \sim \chi_1^2,$$

and thus

$$SS_{Reg} \sim \sigma^2 \chi_1^2.$$

(If the null hypothesis $H_0 : \beta_1 = 0$ is false, then the distribution becomes a noncentral χ^2 .)

Finally, what about SS_{Tot} ? If the null hypothesis $H_0 : \beta_1 = 0$ is true, then the responses y_1, \dots, y_n are independent $N(\beta_0, \sigma^2)^2$, and by the results from the no-predictors model,

$$SS_{Tot} \sim \sigma^2 \chi_{n-1}^2.$$

In summary, in the special case of $\beta_1 = 0$, we have

$$SS_{Reg} \sim \sigma^2 \chi_1^2, \tag{1}$$

$$SS_{Err} \sim \sigma^2 \chi_{n-2}^2, \tag{2}$$

$$SS_{Tot} \sim \sigma^2 \chi_{n-1}^2. \tag{3}$$

It can also be shown that (1) and (2) are independent. If $\beta_1 \neq 0$, then (2) still holds, but (1) and (3) do not.

Mean squares and the F-test. If we divide the sums of squares by their degrees of freedom, we obtain quantities that are commonly known as “mean squares,” which we denote by “ MS .” Dividing SS_{Err} by its degrees of freedom

$(n - 2)$ gives the mean squared error,

$$MS_{Err} = \frac{SS_{Err}}{(n - 2)} \sim \sigma^2 \frac{\chi_{n-2}^2}{(n - 2)}, \quad (4)$$

which we have also called S^2 . Because $E(\chi_{n-2}^2) = n - 2$, we again see that this is an unbiased estimate of σ^2 .

Notice that (4) is true regardless of whether $\beta_1 = 0$. But if $H_0 : \beta_1 = 0$ is true, then we have

$$MS_{Reg} = \frac{SS_{Reg}}{1} \sim \sigma^2 \chi_1^2$$

and

$$MS_{Tot} = \frac{SS_{Tot}}{n - 1} \sim \sigma^2 \frac{\chi_{n-1}^2}{(n - 1)},$$

and thus

$$E(MS_{Reg}) = E(MS_{Tot}) = \sigma^2$$

as well.

Now consider the ratio

$$F = \frac{MS_{Reg}}{MS_{Err}}.$$

Under $H_0 : \beta_1 = 0$, we expect F to be about 1, because then the numerator and denominator would both have expectation σ^2 . If $H_0 : \beta_1 = 0$ is false, then we would expect F to be greater than 1, because in that case

$E(MS_{Reg}) = \sigma^2 + \beta_1^2 S_{xx} > \sigma^2$. Therfore, it makes sense to use F as a statistic for testing H_0 , and we would reject H_0

if the statistic is too large. Under the null hypothesis, this statistic is distributed as $F_{1,n-2}$, so for a .05-level test, we would reject H_0 if F exceeds the 95th percentile of $F_{1,n-2}$.

Recall that we have also tested the null hypothesis $H_0 : \beta_1 = 0$ by comparing

$$t = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)}$$

to a t-distribution with $n - 2$ degrees of freedom. This is equivalent to the test described above, because

$$t^2 = \frac{\hat{\beta}_1^2}{S^2/S_{xx}} = \frac{\hat{\beta}_1^2 S_{xx}}{S^2} = \frac{MS_{Reg}}{MS_{Err}} = F,$$

and the square of a t_{n-2} random variable is distributed as $F_{1,n-2}$.

The ANOVA table. R.A. Fisher placed these results in a convenient analysis-of-variance (ANOVA) table that has become a key part of the output of nearly every major regression program. The table usually has four columns that report the sums of squares (SS), degrees of freedom (df), mean squares (MS) and the F-statistic. The ANOVA table for simple linear regression looks like this.

	SS	df	$MS = SS/df$	F
Regression	SS_{Reg}	1	MS_{Reg}	MS_{Reg}/MS_{Err}
Error	SS_{Err}	$n - 2$	MS_{Err}	
Total	SS_{Tot}	$n - 1$		

Sometimes the regression software will print a p-value corresponding to the F-statistic. If so, it should be exactly the same as the p-value corresponding to the t-statistic for β_1 appearing in the table of coefficients.

For simple linear regression, the ANOVA table does not convey much information beyond that which is already provided in the table of coefficients. For more complicated models, however, the ANOVA table will be extremely valuable.

Sometimes it is helpful to include one more column in the ANOVA table for the expected mean squares ($EMS = E(MS)$).

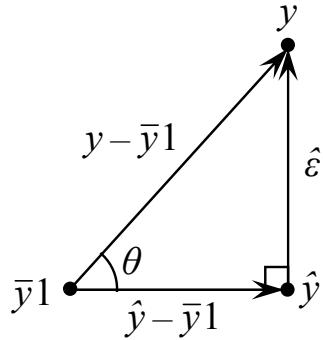
	SS	df	$MS = SS/df$	$E(MS)$
Regression	SS_{Reg}	1	MS_{Reg}	$\sigma^2 + \hat{\beta}_1^2 S_{xx}$
Error	SS_{Err}	$n - 2$	MS_{Err}	σ^2
Total	SS_{Tot}	$n - 1$		

Glancing at $E(MS)$, it becomes obvious why $F = MS_{Reg}/MS_{Err}$ is an appropriate statistic for testing $H_0 : \beta_1 = 0$. If H_0 is true, then this ratio will tend to be about one. If H_0 is false, it will tend to be greater than one. Larger observed values of F indicate greater evidence

against H_0 .

As $n \rightarrow \infty$, $F_{1,n-2} \rightarrow \chi^2_1$, and we would reject H_0 if the F-statistic exceeds $\chi^2_{.95,1} = 3.84 \approx 4$. Therefore, a handy rule-of-thumb is to reject H_0 if $F \geq 4$.

Correlation, r^2 and F . In the last lecture, we argued that the sample correlation r between the response y_i and the predictor x_i is the cosine of the angle θ in the triangle below.



From the well-known identity

$$\sin^2(\theta) + \cos^2(\theta) = 1,$$

we see that

$$\begin{aligned} 1 - r^2 &= \sin^2(\theta) \\ &= \|\hat{\epsilon}\|^2 / \|y - \bar{y}_1\|^2 \\ &= SS_{Err} / SS_{Tot}, \end{aligned}$$

which implies

$$SS_{Err} = (1 - r^2) SS_{Tot}. \quad (5)$$

A couple of lectures ago, we asserted that (5) is true, but

now we have proven it.

From the triangle, we also see that

$$\begin{aligned}\cot^2(\theta) &= 1/\tan^2(\theta) \\ &= \|\hat{y} - \bar{y}_1\|^2 / \|\hat{\epsilon}\|^2 \\ &= SS_{Reg}/SS_{Err}.\end{aligned}$$

But

$$SS_{Reg}/SS_{Err} = F/df,$$

where $df = n - 2$ is the error degrees of freedom, so

$$\cot^2(\theta) = F/df,$$

which we asserted back in Lecture 8. The quantity F/df is an “effect size.” It measures the degree of departure from the null hypothesis. In this case, the null hypothesis is $H_0 : \beta_1 = 0$, so F/df measures the strength of the linear relationship between y and x . So F/df is just another way of expressing the degree to which x and y appear to be correlated. In fact, one of the less familiar trigonometric identities says that

$$\cos^2(\theta) = \frac{\cot^2(\theta)}{\cot^2(\theta) + 1}, \tag{6}$$

or that

$$\cot^2(\theta) = \frac{\cos^2(\theta)}{1 - \cos^2(\theta)}. \tag{7}$$

Identity (6) immediately leads to

$$r^2 = \frac{F/df}{F/df + 1} = \frac{F}{F + df},$$

and identity (6) leads to

$$F = \frac{r^2}{1 - r^2} \times df.$$

Both of these were asserted back in Lecture 5, but now we have proven them.

Exercise. Here's a useful exercise to help you make sure that you thoroughly understand simple linear regression.

If you run a simple linear regression in almost any popular statistical package (e.g. Minitab), the output will look something like the form below, with the blank spaces (-----) replaced by numbers.

THE REGRESSION EQUATION IS

Y = ----- + ----- X

TABLE OF COEFFICIENTS

Predictor	Coef	SE	T	p
Constant	-----	-----	-----	-----
X	-----	-----	-----	-----
<hr/>				
S = -----	R-squared = -----			

ANALYSIS OF VARIANCE

Source	DF	SS	MS	F	p
Regression	-----	-----	-----	-----	-----
Residual Error	-----	-----	-----		
Total	-----	-----			

Suppose you are given the five-number summary

$$\begin{aligned}
 \text{mean of } X &= 9.00 \\
 \text{variance of } X &= 11.00 \\
 \text{mean of } Y &= 7.50 \\
 \text{variance of } Y &= 4.13 \\
 \text{correlation} &= 0.816
 \end{aligned}$$

and you are told that $n = 11$. Can you fill in all of the blanks above *without referring to any notes or sheets of formulas?*

Here is one way do it. First, find the estimated slope,

$$\hat{\beta}_1 = 0.816 \times \sqrt{\frac{4.13}{11.00}} = 0.500,$$

and the intercept,

$$\hat{\beta}_0 = 7.50 - (0.500)(9.00) = 3.00.$$

Next, find the sums of squares,

$$\begin{aligned} SS_{Tot} &= (11 - 1) \times 4.13 \\ &= 41.3, \\ SS_{Reg} &= 0.816^2 \times 41.3 \\ &= 27.5, \\ SS_{Err} &= 41.3 - 27.5 \\ &= 13.8, \end{aligned}$$

the mean-squared error and the residual standard error,

$$\begin{aligned} S^2 &= 13.8/9 = 1.533, \\ S &= \sqrt{1.533} = 1.238. \end{aligned}$$

Now we can find the standard errors for the coefficients,

$$\begin{aligned} S_{xx} &= (11 - 1) \times 11.00 = 110.0, \\ SE(\hat{\beta}_1) &= \sqrt{1.533/110.0} = 0.118, \\ SE(\hat{\beta}_0) &= \sqrt{1.533 \left(\frac{1}{11} + \frac{9.00^2}{110.0} \right)} = 1.125, \end{aligned}$$

and their corresponding t-statistics,

$$\begin{aligned} t \text{ for } \hat{\beta}_0 &= 3.00/1.125 = 2.67, \\ t \text{ for } \hat{\beta}_1 &= 0.500/0.118 = 4.24. \end{aligned}$$

The p-values require the use of a computer,

$$\begin{aligned} 2 \times (1 - P(t_9 \leq 2.67)) &= 0.026, \\ 2 \times (1 - P(t_9 \leq 4.24)) &= 0.002. \end{aligned}$$

Two more necessary numbers are

$$r^2 = 0.816^2 = 0.666,$$

$$F = 27.5/1.533 = 17.9,$$

but we could have also computed F by squaring the t-statistic for $\hat{\beta}_1$,

$$F = 4.24^2 = 18.0,$$

and the discrepancy between the two answers is just due to rounding error. The p-value associated with F is

$$1 - P(F_{1,9} \leq 17.9) = 0.002,$$

which agrees with the p-value from the t-statistic.

Now we can fill in all the blanks.

THE REGRESSION EQUATION IS

$Y = 3.00 + 0.500 X$

TABLE OF COEFFICIENTS

Predictor	Coef	SE	T	p
Constant	3.00	1.125	2.67	.026
X	.500	0.118	4.24	.002

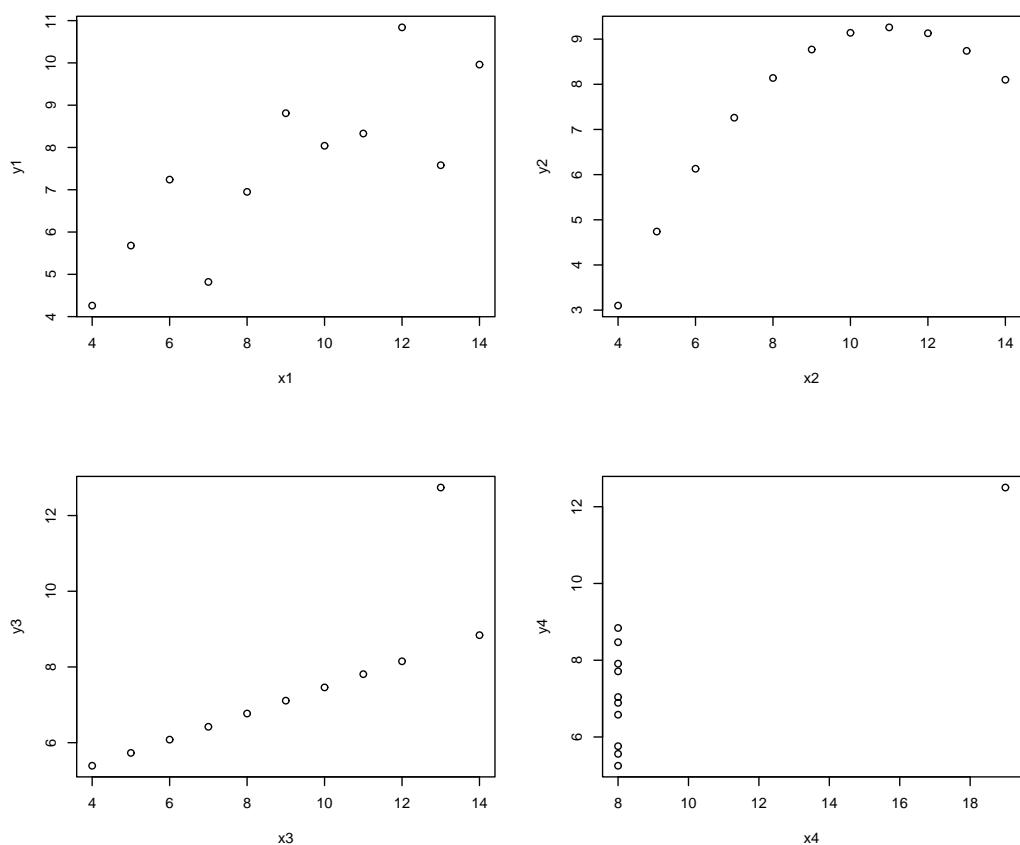
S = 1.238 R-squared = 0.666

ANALYSIS OF VARIANCE

Source	DF	SS	MS	F	p
Regression	1	27.5	27.5	17.9	.002
Residual Error	9	13.8	1.533		
Total	10	41.3			

Don't forget to look at the data! The five-number summary is a set of sufficient statistics that tells you everything you need to know *if the model holds*. But these numbers do not say anything about whether the model itself is appropriate. Graphical displays are a great help in this regard.

To illustrate this point, Anscombe (1973) created four bivariate datasets with $n = 11$ observations each. All four datasets produce the same five-number summary that we just worked with, so linear regression analyses will lead to exactly the same results. But if you simply plot the data, you will immediately see that conventional linear regression is appropriate for only one of the four datasets.



These data are included in the R distribution. You can access them this way.

```
> attach(anscombe)
> anscombe
   x1 x2 x3 x4     y1     y2     y3     y4
1 10 10 10  8 8.04 9.14  7.46  6.58
2  8  8  8  8 6.95 8.14  6.77  5.76
3 13 13 13  8 7.58 8.74 12.74  7.71
4  9  9  9  8 8.81 8.77  7.11  8.84
5 11 11 11  8 8.33 9.26  7.81  8.47
6 14 14 14  8 9.96 8.10  8.84  7.04
7  6  6  6  8 7.24 6.13  6.08  5.25
8  4  4  4 19 4.26 3.10  5.39 12.50
9 12 12 12  8 10.84 9.13  8.15  5.56
10  7  7  7  8 4.82 7.26  6.42  7.91
11  5  5  5  8 5.68 4.74  5.73  6.89
```

Confidence bands for simultaneous inference. In the last lecture, we derived a 95% confidence interval for the mean response at any given value of x ,

$$\hat{y}(x) \pm t_{.975,n-2} SE(\hat{y}(x)),$$

where

$$SE(\hat{y}(x)) = S \sqrt{\frac{1}{n} + \frac{(x - \bar{x})^2}{S_{xx}}} \quad (8)$$

is the standard error for the mean response. And we derived a 95% prediction interval for a future response y^* at a given value of x ,

$$\hat{y}(x) \pm t_{.975,n-2} SE(y^*),$$

where

$$SE(y^*) = S \sqrt{1 + \frac{1}{n} + \frac{(x - \bar{x})^2}{S_{xx}}},$$

is the standard error of prediction. We remarked that these are “pointwise” intervals. That is, they will cover their targets with probability 95% when applied to a single value of x . If we apply these procedures to multiple x values, then the probability that at least one interval fails to cover its target will exceed 5%.

Is there a way to create intervals that will cover their targets for multiple values of x with probability .95? Yes. There are many ways to construct intervals for simultaneous inference. Today we describe two ways.

Method 1: The Bonferroni method (also known as “ α -splitting.”) Suppose we want to construct a set of I confidence intervals that simultaneously cover all of their targets with probability at least $1 - \alpha$ for a given value of α (typically $\alpha = .05$). Let E_i denote the event “interval i misses its target.” In the language of hypothesis testing, E_i corresponds to a Type 1 error, because it implies that the true value of the i th target parameter would be rejected in an α -level test. We want keep the probability of at making at least one error at or below α ,

$$P(E_1 \cup E_2 \cup \dots \cup E_I) \leq \alpha. \quad (9)$$

But the well known Bonferroni inequality says that

$$P(E_1 \cup E_2 \cup \dots \cup E_I) \leq \sum_{i=1}^I P(E_i).$$

Therefore, (9) will be satisfied if the individual error rates sum to α ,

$$\sum_{i=1}^I P(E_i) = \alpha.$$

The overall error rate α can be maintained by splitting it into equal parts, by setting

$$P(E_i) = \alpha/I \quad \text{for } i = 1, \dots, I.$$

(We could have split α into unequal parts, but this is rarely done.)

This is a very general principle that we can apply whenever we need to construct multiple confidence intervals or hypothesis tests. In the regression setting, suppose we need confidence intervals for the mean response at I different values of x , which we denote by $x_1^*, x_2^*, \dots, x_I^*$. And we want to make sure that all of these intervals simultaneously their targets with probability of at least .95. The Bonferroni-adjusted intervals are

$$\hat{y}(x) \pm t_{1-\alpha/(2I), n-2} SE(\hat{y}(x)).$$

We would maintain 95% simultaneous confidence by constructing

- two 97.5% intervals,
- five 99% intervals,
- ten 99.5% intervals,
- fifty 99.9% intervals,

and so on. The same adjustment can also be applied to the prediction intervals.

Method 2: Scheffe's method. The Bonferroni method assumes that we want confidence or prediction intervals at a finite number of design points $x_1^*, x_2^*, \dots, x_I^*$. But what if we want intervals that are valid for all possible values of x simultaneously? Is it possible to compute a region that captures the true regression function for all $x \in (-\infty, \infty)$? Yes, it is possible.

First, let's construct a joint confidence region for the regression coefficients. The vector of estimated coefficients $\hat{\beta}$ is normally distributed about the true coefficients,

$$\hat{\beta} \sim N(\beta, \sigma^2(X^T X)^{-1}).$$

It follows from property of the multivariate normal distribution (Lecture 4) that

$$\frac{1}{\sigma^2} (\hat{\beta} - \beta)^T (X^T X) (\hat{\beta} - \beta) \sim \chi_p^2, \quad (10)$$

where $p = \dim(\beta)$. Our “Fundamental Theorem” says that

$$S^2 / \sigma^2 \sim \chi_{n-p}^2 / (n - p), \quad (11)$$

and dividing (10) by (11) gives

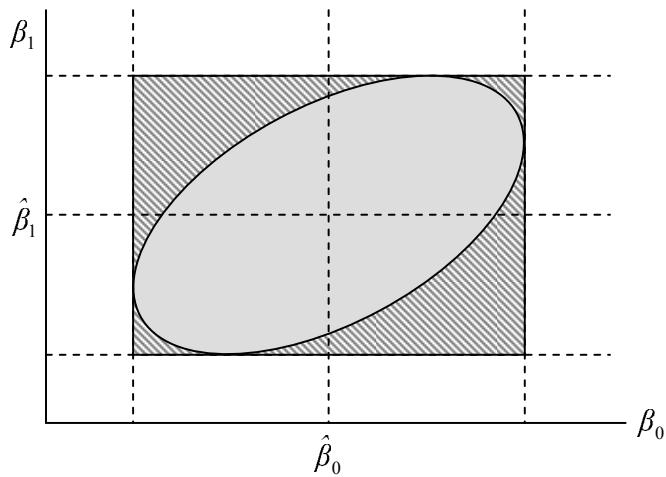
$$\frac{1}{S^2} (\hat{\beta} - \beta)^T (X^T X) (\hat{\beta} - \beta) \sim p F_{p,n-p}.$$

A 95% joint confidence region for β is the set of all β values such that

$$\frac{1}{S^2} (\hat{\beta} - \beta)^T (X^T X) (\hat{\beta} - \beta) \leq p F_{.95,p,n-p}, \quad (12)$$

where $F_{.95,p,n-p}$ is the 95th percentile of $F_{p,n-p}$. For the simple linear regression model ($p = 2$), this region is an ellipse centered at $(\hat{\beta}_0, \hat{\beta}_1)$. More generally, it is a p -dimensional ellipsoid centered at $\hat{\beta}$.

In the two-dimensional case, suppose we project the horizontal and vertical edges of the ellipse onto the β_0 and β_1 axes, producing an interval for β_0 and an interval for β_1 :



The shaded rectangular region enclosed by these tangent lines also encloses the elliptical confidence region.

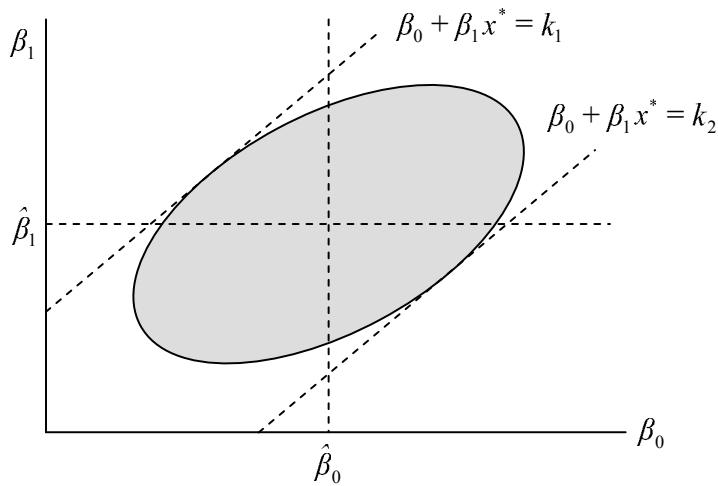
Therefore, the confidence intervals for β_0 and β_1 created by this procedure will simultaneously cover their true values with probability exceeding .95.

Now let's use the same idea to get an interval for the parameter $\beta_0 + \beta_1 x^*$, the mean response when $x = x^*$. That is, we need to find two values, k_1 and k_2 , such that the lines

$$\beta_0 + \beta_1 x^* = k_1,$$

$$\beta_0 + \beta_1 x^* = k_2$$

will be tangent to the ellipse. (In higher dimensions, we would need to find hyperplanes that are tangent to the ellipsoid.)



The solution to this geometric problem is that k_1 and k_2 are given by

$$\hat{y}(x^*) \pm \sqrt{p F_{.95,p,n-p}} SE(\hat{y}(x^*)), \quad (13)$$

where $SE(\hat{y}(x^*))$ is the standard error of the predicted mean of y at $x = x^*$. This result holds for the general linear regression model. In the case of simple linear regression, $p = 2$, and the formula for $SE(\hat{y}(x^*))$ is given in (8). If a β vector falls within the 95% ellipsoid (12), then the implied mean for y at $x = x^*$ falls within the limits shown in (13) *simultaneously for every x^** .

This approach for creating simultaneous intervals is known as Scheffe's method, but an early solution to this problem was also published by Working and Hotelling (1929). When applied to the linear regression model, the resulting region is often called the Working-Hotelling confidence band. It is a bit wider than the pointwise intervals described in the last lecture, because it is guaranteed to

capture the whole regression line with probability 0.95.

Here's an example in R using the blood pressure data.

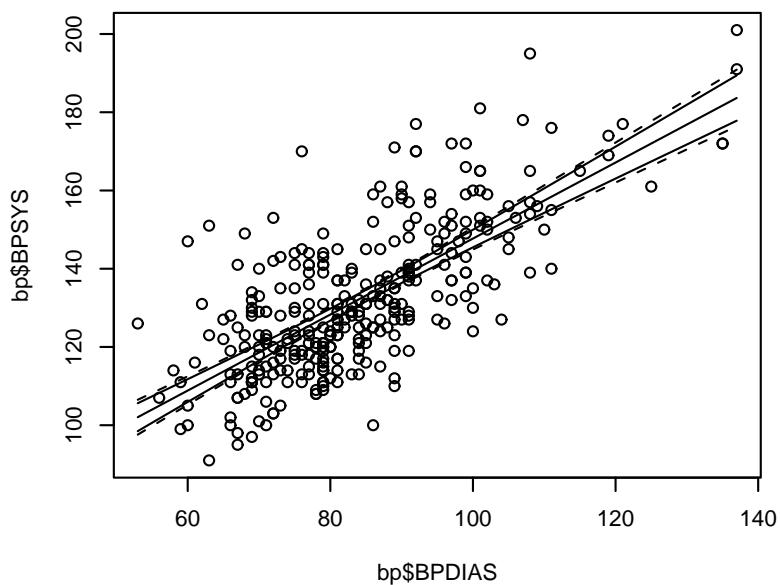
```
> # regress BPSYS on BPDIAS and extract the estimated parameters
> bp <- read.table("bp.dat", header=T)
> result <- lm( BPSYS ~ BPDIAS, data=bp )
> beta.0 <- result$coef[1] # estimated intercept
> beta.1 <- result$coef[2] # estimated slope
> S2 <- sum( result$residuals^2 ) / result$df.residual # MSE

> # compute predictions and their standard errors
> n <- nrow(bp) # sample size
> x <- seq( from=min(bp$BPDIAS), to=max(bp$BPDIAS), length=200)
> xbar <- mean( bp$BPDIAS ) # mean of observed x's
> Sxx <- (n-1) * var( bp$BPDIAS )
> yhat <- beta.0 + beta.1 * x
> se.yhat <- sqrt( S2 * ( 1/n + ( x - xbar )^2 / Sxx ) )

> # scatterplot and regression predictions
> plot( bp$BPDIAS, bp$BPSYS ) # scatterplot of data
> lines( x, yhat )

> # add the 95% pointwise confidence intervals as solid lines
> lower <- yhat - qt(.975,n-2) * se.yhat
> upper <- yhat + qt(.975,n-2) * se.yhat
> lines( x, lower )
> lines( x, upper )

> # add the Working-Hotelling confidence band as dashed lines
> lower <- yhat - sqrt(2*qf(.95,2,n-2)) * se.yhat
> upper <- yhat + sqrt(2*qf(.95,2,n-2)) * se.yhat
> lines( x, lower, lty=2 )
> lines( x, upper, lty=2 )
```



IMPORTANT MATRICES IN REGRESSION

Beyond simple linear regression. In practice, simple linear regression modeling is rare because

- we often have more than one predictor, and
- the relationship between the response and a predictor may be nonlinear.

These two issues are related. Even if you have just one predictor, but its relationship with the response is nonlinear, then you may need to create two or more columns in X to characterize its effect. Nonlinear relationships cannot be fully dealt with until we understand what happens with multiple predictors. So we will now return to the more general linear regression model.

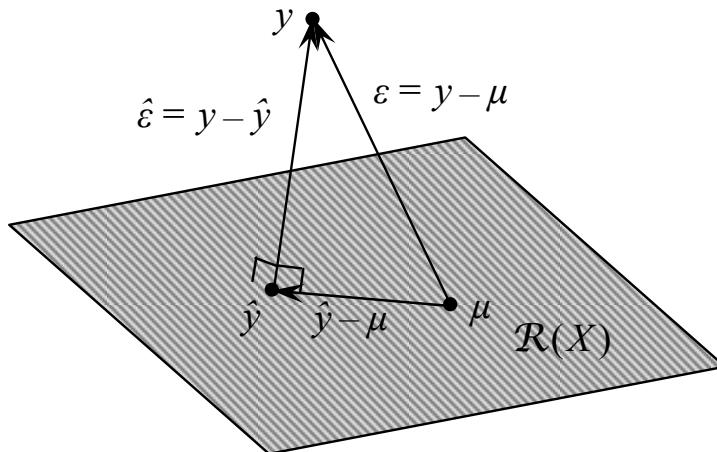
We have expressed the general linear regression model as

$$\begin{aligned} y &\sim N(\mu, \sigma^2 I), \\ \mu &\in \mathcal{R}(X), \end{aligned}$$

which implies that $\mu = X\beta$ for some β . The vector of fitted values, $\hat{y} = X\hat{\beta}$, is the projection of y into $\mathcal{R}(X)$.

The true mean, $\mu = X\beta$, which is unknown, also lies within $\mathcal{R}(X)$, but it is farther away from y than \hat{y} is.

Try to keep this picture in mind as we discuss the properties of the regression model.



Important matrices in regression. Thus far, we have encountered several important vectors in the linear regression model, including

- the **response vector** y ,
- the **estimated coefficients** $\hat{\beta} = (X^T X)^{-1} X^T y$,
- the **fitted values** $\hat{y} = X\hat{\beta}$ (also called $\hat{\mu}$), and
- the **estimated residuals** $\hat{\epsilon} = y - \hat{y}$.

We have also encountered the **matrix of predictors** X , which is also sometimes called the **design matrix**. The latter term comes from the analysis of data from randomized experiments, in which case the variables in X

are related to the experimental design. There are several other matrices derived from X that play an important role in the theory and practice of regression.

The SSCP matrix. $X^T X$ is a $p \times p$ symmetric matrix of the form

$$X^T X = \begin{bmatrix} \sum_i x_{i1}^2 & \sum_i x_{i1}x_{i2} & \cdots & \sum_i x_{i1}x_{ip} \\ \sum_i x_{i2}x_{i1} & \sum_i x_{i2}^2 & \cdots & \sum_i x_{i2}x_{ip} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_i x_{ip}x_{i1} & \sum_i x_{ip}x_{i2} & \cdots & \sum_i x_{ip}^2 \end{bmatrix},$$

where the sums are taken over $i = 1, \dots, n$. The (j, k) th element of this matrix is the inner product of X_j and X_k . $X^T X$ is often called the sums of squares and cross-products (SSCP) matrix. It plays the same role that the sample size $n = 1^T 1$ played in the no-predictors model. In that model, the variance of $\hat{\beta} = \bar{y}$ was σ^2/n , and thus n determined the precision with which β could be estimated. In the model with predictors, the precision of $\hat{\beta}$ is determined by $X^T X$. As n grows, the elements of $X^T X$ become large, and the elements of $(X^T X)^{-1}$ become small.

One can show that $X^T X$ has the same rank as X ,

$$\text{rank}(X^T X) = \text{rank}(X).$$

Therefore, $(X^T X)^{-1}$ is defined if and only if the columns

of X are linearly independent, i.e. if $\text{rank}(X) = p$. If a linear dependency exists—which occurs if one or more columns can be expressed as a linear combination of other columns—then X will be rank-deficient, and a unique inverse of $X^T X$ will not exist.

The inverse of SSCP. The true covariance matrix for $\hat{\beta}$,

$$V(\hat{\beta}) = \sigma^2(X^T X)^{-1},$$

is unknown because σ^2 is unknown. Replacing σ^2 by its unbiased estimate gives

$$\hat{V}(\hat{\beta}) = S^2(X^T X)^{-1}.$$

The square roots of the diagonal elements of $\hat{V}(\hat{\beta})$ are the standard errors for $\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_p$. In R, the matrix $S^2(X^T X)^{-1}$ is often called `cov.scaled`, and $(X^T X)^{-1}$ is called `cov.unscaled`.

Relationships among the columns of X will determine the structure of $(X^T X)^{-1}$. If the j th and k th columns of X are orthogonal,

$$\sum_{i=1}^n x_{ij} x_{ik} = 0,$$

then the (j, k) th element of $X^T X$ will be zero. In that case, the corresponding element of $(X^T X)^{-1}$ is not necessarily zero, because that element is influenced by the other columns of X as well. But there are situations where zeros in $X^T X$ do produce zeros in the corresponding

entries of $(X^T X)^{-1}$.

Suppose we partition X as

$$X = [X_1, X_2],$$

where X_1 is $n \times p_1$, X_2 is $n \times p_2$, and $p_1 + p_2 = p$. (Until now, we have used X_1 and X_2 to denote single columns of X . Now they are matrices.) Then

$$X^T X = \begin{bmatrix} X_1^T \\ X_2^T \end{bmatrix} \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} X_1^T X_1 & X_1^T X_2 \\ X_2^T X_1 & X_2^T X_2 \end{bmatrix}.$$

We can partition β in a similar fashion,

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix},$$

where β_1 and β_2 are vectors of length p_1 and p_2 , so that the regression function splits into two parts,

$$\mu = X\beta = X_1\beta_1 + X_2\beta_2.$$

If X_1 is orthogonal to X_2 , in the sense that every column in X_1 is orthogonal to every column in X_2 , then the off-diagonal blocks of $X^T X$ are zero, and

$$(X^T X)^{-1} = \begin{bmatrix} (X_1^T X_1)^{-1} & 0 \\ 0 & (X_2^T X_2)^{-1} \end{bmatrix}.$$

In this case, the least-squares estimate $\hat{\beta}$ can be computed

by separate regressions of y on X_1 and X_2 ,

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix},$$

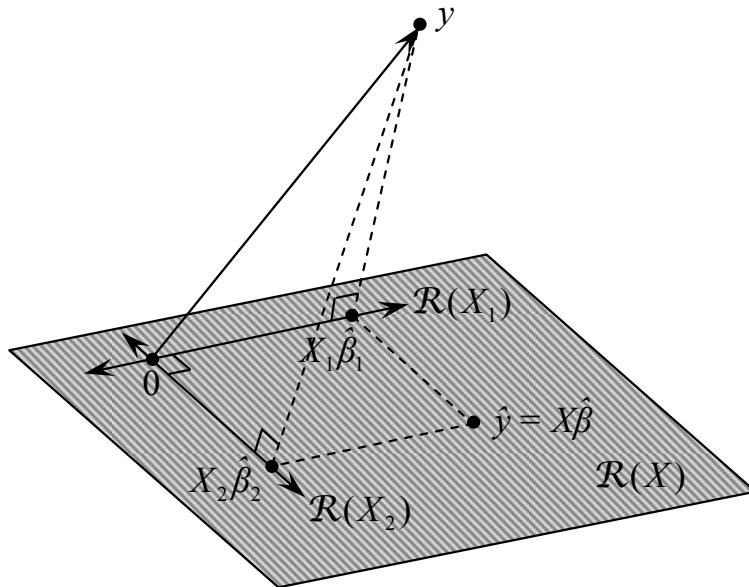
$$\hat{\beta}_1 = (X_1^T X_1)^{-1} X_1^T y,$$

$$\hat{\beta}_2 = (X_2^T X_2)^{-1} X_2^T y,$$

and then

$$\hat{y} = X_1 \hat{\beta}_1 + X_2 \hat{\beta}_2$$

is the projection of y onto $\mathcal{R}(X_1)$, plus the projection of y onto $\mathcal{R}(X_2)$.



Here, $\hat{\beta}_1$ and $\hat{\beta}_2$ are independent.

Now consider what happens when the first column of X is $1 = (1, \dots, 1)^T$. Suppose we partition X as

$$X = [1, X_1, X_2],$$

where X_1 is $n \times p_1$, X_2 is $n \times p_2$, and $1 + p_1 + p_2 = p$. And let's partition β as

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix},$$

where the lengths of β_0 , β_1 and β_2 are 1, p_1 and p_2 . And let's suppose that every column in X_1 is *uncorrelated* with every column in X_2 . (Recall that uncorrelated is not the same thing as orthogonality. Two vectors are uncorrelated if their deviations from their respective means are orthogonal.) In this case, $(X^T X)^{-1}$ has the pattern

$$\begin{aligned} (X^T X)^{-1} &= \begin{bmatrix} n & 1^T X_1 & 1^T X_2 \\ X_1^T 1 & X_1^T X_1 & X_1^T X_2 \\ X_2^T 1 & X_2^T X_1 & X_2^T X_2 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \times & \times & \times \\ \times & \times & 0 \\ \times & 0 & \times \end{bmatrix}, \end{aligned}$$

where “0” denotes a block of zeros, and “ \times ” denotes a block that is non-zero. Here, the estimate of β_1 will be the same whether or not X_2 is included in the model, and the estimate of β_2 will be the same whether or not X_1 is included in the model. If we regress y on $[1, X_1]$ and obtain the least squares estimate for $(\beta_0, \beta_1^T)^T$, the estimate for β_1 will be the same as if we had regressed y on the full X .

And if we regress y on $[1, X_2]$, the estimate for β_2 will be the same as if we had regressed y on the full X . The estimated *intercepts* may change, because the columns of X_1 and X_2 are not necessarily orthogonal to 1. But if X_1 and X_2 are orthogonal to 1—which means that each column of X_1 and X_2 has been centered at its mean—and X_1 and X_2 are also uncorrelated with each other, then we obtain a fully block-diagonal pattern

$$(X^T X)^{-1} = \begin{bmatrix} n^{-1} & 0 & 0 \\ 0 & (X_1^T X_1)^{-1} & 0 \\ 0 & 0 & (X_2^T X_2)^{-1} \end{bmatrix}.$$

We can summarize these results as follows.

- Adding variables to a regression model that are uncorrelated with all of the variables already in the model will not change the estimated slopes for the variables already in the model.
- If these additional variables also have means of zero, then the estimated intercept will not change either.

The projection matrix. Another key matrix is the $n \times n$ projection matrix

$$H = X(X^T X)^{-1}X^T.$$

H is obviously symmetric. It is called a projection matrix

because it projects y into $\mathcal{R}(X)$,

$$Hy = X\hat{\beta} = \hat{y}.$$

It is also called the “hat matrix” because it is the matrix that changes y into \hat{y} . That is, it “puts a hat on y .”

If a is any vector of length n , $\hat{a} = Ha$ is the projection of a into $\mathcal{R}(X)$. If a is orthogonal to $\mathcal{R}(X)$, then its projection onto $\mathcal{R}(X)$ will be zero ($Ha = 0$). If a already lies within $\mathcal{R}(X)$, then projecting it onto $\mathcal{R}(X)$ will leave it unchanged ($Ha = a$). For this reason, $HHa = Ha$ and $HH = H$. This property is called idempotence.

H can be very large, and regression analysts rarely compute this matrix directly. Its columns are not linearly independent; in fact, this matrix has the same rank as X ,

$$\text{rank}(H) = \text{rank}(X) = p.$$

If the columns of two different matrices span the same space,

$$\mathcal{R}(X) = \mathcal{R}(Z)$$

then the hat matrices obtained from X and Z are equal,

$$X(X^T X)^{-1} X^T = Z(Z^T Z)^{-1} Z^T.$$

Finally, the trace of the hat matrix (i.e. the sum of its diagonal elements) is equal to its rank,

$$\text{tr } H = \text{rank } H = p.$$

These properties are useful for deriving many important

results about the general linear regression model. Here is a small example. What is the distribution of \hat{y} ? Because

$$y \sim N(X\beta, \sigma^2 I),$$

and because $\hat{y} = Hy$, it follows that \hat{y} is multivariate normal with mean

$$\begin{aligned} E(\hat{y}) &= H X \beta \\ &= X \beta \end{aligned}$$

(because $X\beta$ already lies within $\mathcal{R}(X)$), and covariance matrix

$$\begin{aligned} V(\hat{y}) &= H \sigma^2 I H \\ &= \sigma^2 H H \\ &= \sigma^2 H. \end{aligned}$$

The other projection matrix. If we subtract H from the $n \times n$ identity matrix,

$$(I - H) = I - X(X^T X)^{-1} X^T,$$

we get the matrix that transforms y into $\hat{\epsilon}$,

$$(I - H)y = y - \hat{y} = \hat{\epsilon}.$$

This is also projection matrix. It projects a vector into the linear space of vectors orthogonal to $\mathcal{R}(X)$. For any n -dimensional vector a , $(I - H)a$ is orthogonal to Ha and

to any other vector in $\mathcal{R}(X)$.

As a projection matrix, $(I - H)$ it is symmetric and idempotent,

$$\begin{aligned}(I - H)(I - H) &= I - IH - HI + HH \\ &= I - 2H + H \\ &= I - H.\end{aligned}$$

Its rank is equal to its trace, which is

$$\text{tr}(I - H) = \text{rank}(I - H) = n - p.$$

Using these properties, we can derive many more important results. For example, we can find the distribution of $\hat{\epsilon}$. Because $\hat{\epsilon} = (I - H)y$, it follows that $\hat{\epsilon}$ is multivariate normal with mean

$$E(\hat{\epsilon}) = (I - H)X\beta = 0$$

and covariance matrix

$$V(\hat{\epsilon}) = (I - H)\sigma^2 I (I - H) = \sigma^2(I - H).$$

Analysis of variance and the omnibus F-test. In the last lecture, when we were examining the simple linear regression model, we divided the total sum of squares as

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^n (y_i - \hat{y}_i)^2,$$

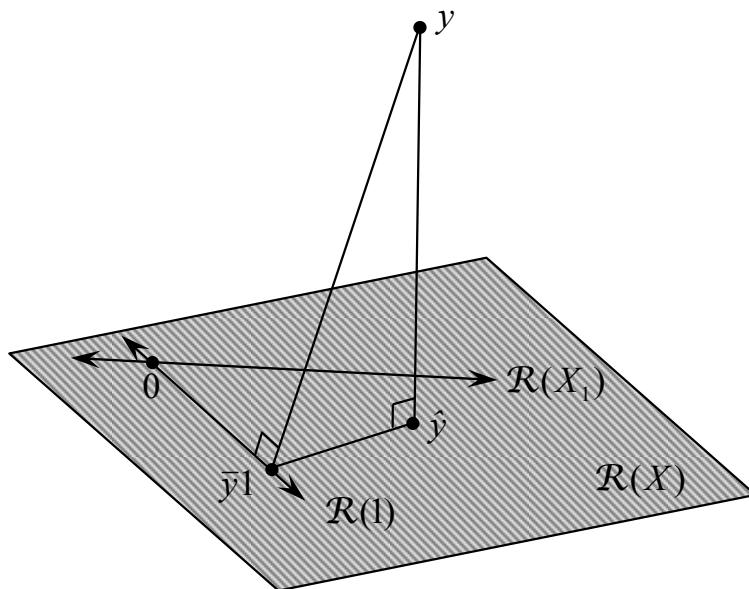
or

$$SS_{Tot} = SS_{Reg} + SS_{Err},$$

and we discussed the distribution of each part. This same decomposition holds for the general linear regression model, as long as the design matrix includes a constant (i.e., a column of ones). Suppose we partition the X -matrix as

$$X = [1, X_1],$$

where 1 is a column of ones, and X_1 is the $n \times (p - 1)$ matrix containing everything else. The decomposition of SS_{Tot} into SS_{Reg} and SS_{Err} corresponds to the Pythagorean Theorem applied to the right triangle shown below.



(In this picture, we have depicted $\mathcal{R}(X)$ as a line, which will be the case if X_1 has a single column. More generally, it will be a $(p - 1)$ -dimensional hyperplane.)

This triangle compares the fit of the model

$$\begin{aligned} E(y) &= X\beta \\ &= 1\beta_0 + X_1\beta_1 \end{aligned}$$

(where β_0 is the intercept, and β_1 is now a vector of length $(p - 1)$) to the no-predictors model

$$E(y) = 1\beta_0.$$

The triangle enables us to test the null hypothesis $H_0 : \beta_1 = 0$, which implies that all predictors in the model beyond the constant 1 are unnecessary. If the vector $\hat{y} - \bar{y}1$ is short, i.e. if SS_{Reg} is small, then the evidence against H_0 is weak. If SS_{Reg} is large, then we can reject H_0 in favor of the alternative that at least one element of β_1 is nonzero. The test is based on the following theorem.

Theorem. Under the general linear regression model

$$y \sim N(X\beta, \sigma^2 I),$$

the error sum of squares is distributed as

$$\|\hat{\epsilon}\|^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \sim \sigma^2 \chi_{n-p}^2. \quad (1)$$

Moreover, if we partition the $n \times p$ design matrix as

$$X = [1, X_1],$$

where 1 is a column of ones, and partition the coefficients as $\beta^T = (\beta_0, \beta_1^T)^T$, where $\dim(\beta_1) = p-1$, then, under the null hypothesis $H_0 : \beta_1 = 0$, the regression sum of squares is distributed as

$$\|\hat{y} - \bar{y}1\|^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 \sim \sigma^2 \chi_{p-1}^2 \quad (2)$$

and is independent of (1).

Using this theorem, the test of the null hypothesis $H_0 : \beta_1 = 0$ is carried out by comparing

$$F = \frac{SS_{Reg}/(p-1)}{SS_{Err}/(n-p)} \quad (3)$$

to an F-distribution with $p-1$ numerator and $n-p$ denominator degrees of freedom. In a .05-level test, we

would reject H_0 if

$$F \geq F_{.95,p-1,n-p},$$

and the p-value is

$$p = P(F_{p-1,n-p} \geq F).$$

If H_0 is rejected, then we conclude that *at least one of the predictors in the model is related to the response*. But we do not necessarily know which one(s). The omnibus F is testing whether or not the predictors, as a group, have any ability to predict the response.

PARTIAL AND SEQUENTIAL F-TESTS

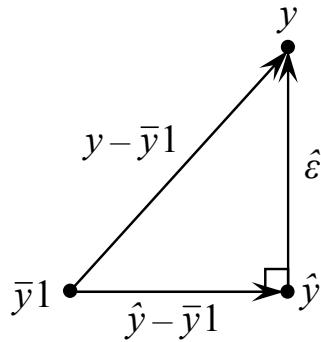
Last time, we discussed the omnibus F-test for the general linear regression model

$$y \sim N(X\beta, \sigma^2 I).$$

We partitioned the $(n \times p)$ design matrix as $X = [1, X_1]$, where X_1 is $n \times (p - 1)$. And we partition the coefficients as $\beta = (\beta_0, \beta_1^T)^T$, where $\dim(\beta_1) = p - 1$, so that

$$y = 1\beta_0 + X_1\beta_1 + \epsilon.$$

The omnibus F-test, which is a test of $H_0 : \beta_1 = 0$, was based on the triangle below.



Our theorem said that

- SS_{Err} , which is the squared length of $\hat{\epsilon} = y - \hat{y}$, is distributed as $\sigma^2 \chi_{n-p}^2$.

- If the null hypothesis is true, then SS_{Reg} , which is the squared length of $\hat{y} - \bar{y}1$, is distributed as $\sigma^2\chi_{p-1}^2$ and is independent of SS_{Err} .

Therefore, under H_0 , the test statistic

$$F = \frac{SS_{Reg}/(p-1)}{SS_{Err}/(n-p)}$$

is distributed as $F_{p-1, n-p}$.

Another way to obtain the omnibus F. Recall that

$$\hat{\beta} \sim N(\beta, \sigma^2(X^T X)^{-1}).$$

Suppose we partition the inverse-SSCP matrix as

$$(X^T X)^{-1} = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix},$$

where A is 1×1 , B is $1 \times (p-1)$, and C is $(p-1) \times (p-1)$. (Notice that, in general, C is different from $(X_1^T X_1)^{-1}$; the two are equal only if each column of X_1 is orthogonal to 1, i.e. if all of the predictors have been de-meaned. Then

$$\hat{\beta}_1 \sim N(\beta_1, \sigma^2 C),$$

and

$$\frac{1}{\sigma^2} (\hat{\beta}_1 - \beta_1)^T C^{-1} (\hat{\beta}_1 - \beta_1) \sim \chi_{p-1}^2.$$

If we divide this quantity by

$$\frac{S^2}{\sigma^2} \sim \chi_{n-p}^2 / (n-p),$$

it follows that

$$\frac{1}{S^2} (\hat{\beta}_1 - \beta_1)^T C^{-1} (\hat{\beta}_1 - \beta_1) \sim (p-1) F_{p-1, n-p}.$$

A joint 95% confidence region for β_1 is the set of all β_1 -vectors such that

$$\frac{1}{S^2} (\hat{\beta}_1 - \beta_1)^T C^{-1} (\hat{\beta}_1 - \beta_1) \leq (p-1) F_{.95, p-1, n-p}.$$

This set is a $p-1$ -dimensional ellipsoid centered at $\hat{\beta}_1$, with shape determined by C . If $\beta_1 = 0$ lies outside of this region, then we can reject $H_0 : \beta_1 = 0$ at the .05 level. That is, we would reject H_0 if

$$\frac{1}{S^2} \hat{\beta}_1^T C^{-1} \hat{\beta}_1 \leq (p-1) F_{.95, p-1, n-p},$$

or if

$$\frac{\hat{\beta}_1^T C^{-1} \hat{\beta}_1 / (p-1)}{S^2} \leq F_{.95, p-1, n-p}. \quad (1)$$

But it is possible to show that

$$\hat{\beta}_1^T C^{-1} \hat{\beta}_1 = SS_{Reg},$$

so (1) is just another way of writing the omnibus F-test.

Example. Let's go back to the body measurements dataset and regress log(CHOL) on log(BMI) and MORPH.

```
> body <- read.table("body.dat", header=T)

> # create new variables for regression
> meters <- body$height * 2.54 / 100 # height in meters
> kg <- body$weight * 0.45359237 # weight in pounds
> body$log.bmi <- log(kg/meters^2)
> body$log.chol <- log( body$chol )
> body$morph <- body$waist / body$hips

> result <- lm( log.chol ~ log.bmi + morph, data=body)
> summary(result)

Call:
lm(formula = log.chol ~ log.bmi + morph, data = body)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.775122 -0.138062  0.002809  0.139035  0.879628 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.51682   0.07753  58.260 <2e-16 ***
log.bmi     0.05559   0.02488   2.234  0.0256 *  
morph       0.65073   0.06226  10.452 <2e-16 *** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2115 on 1832 degrees of freedom
Multiple R-Squared:  0.08088,    Adjusted R-squared: 0.07988 
F-statistic: 80.61 on 2 and 1832 DF,  p-value: < 2.2e-16
```

The F-statistic of 80.61 reported at the bottom is the omnibus F.

Let's see if we can reproduce this from the raw data using vector and matrix operations.

First, let's form the design matrix and compute the least-squares estimates. In R, the function `t()` transposes a matrix, the operator `%*%` performs matrix multiplication, and the function `solve()` computes an inverse.

```
> y <- body$log.chol
> x <- cbind( 1, body$log.bmi, body$morph)
> betahat <- solve( t(x) %*% x ) %*% t(x) %*% y
> betahat
      [,1]
[1,] 4.5168170
[2,] 0.0555872
[3,] 0.6507308
```

Now let's compute the omnibus F-statistic using the sums of squares.

```
> SSTot <- sum( ( y - mean(y) )^2 )
> yhat <- x %*% betahat
> SSReg <- sum( ( yhat - mean(y) )^2 )
> SSErr <- sum( ( y - yhat )^2 )
> n <- length(y)
> p <- ncol(x)
> F <- ( SSReg/(p-1) ) / ( SSErr / (n-p) )
> F
[1] 80.60813
```

Now compute F the other way, by using the 95% confidence region for the coefficients beyond the intercept.

```
> beta1hat <- betahat[2:3]
> xtxinv <- solve( t(x) %*% x )
> C <- xtxinv[2:3, 2:3]
> num <- t(beta1hat) %*% solve(C) %*% beta1hat / (p-1)
> den <- SSErr / (n-p)
> F <- num / den
> F
      [,1]
[1,] 80.60813
```

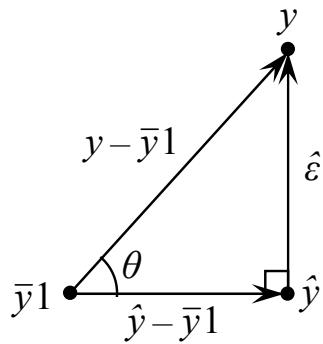
Omnibus F and the R^2 statistic. The omnibus F statistic measures the strength of total evidence against $H_0 : \beta_1 = 0$. Therefore:

- It grows as the predictors in the regression model explain more of the variance in the response.
- It grows as the sample size n increases.

The strength of the relationship between the response and the predictors can be measured by

$$R^2 = \frac{SS_{Reg}}{SS_{Tot}},$$

which does not tend to increase with n . R^2 is the squared correlation between y and \hat{y} , and it is the proportion of variance in y that is collectively explained by all the predictors in X_1 . It can be viewed as the squared cosine of the angle θ shown below.



The squared cotangent of this angle is

$$\cot^2 \theta = \frac{SS_{Reg}}{SS_{Err}} = \frac{df_1}{df_2} F,$$

where $df_1 = p - 1$ and $df_2 = n - p$ are the numerator and denominator degrees of freedom in the omnibus F. Using the identity

$$\cos^2(\theta) = \frac{\cot^2(\theta)}{\cot^2(\theta) + 1},$$

it immediately follows that

$$R^2 = \frac{F}{F + df_2/df_1}.$$

Notice that this is a generalization of our formula

$$R^2 = \frac{F}{F + df}.$$

for tests about single coefficients. Going the other way,

$$\cot^2(\theta) = \frac{\cos^2(\theta)}{1 - \cos^2(\theta)}$$

immediately leads to

$$F = \left(\frac{R^2}{1 - R^2} \right) \times \left(\frac{df_2}{df_1} \right),$$

which generalizes the formula

$$F = \left(\frac{R^2}{1 - R^2} \right) \times df$$

from simple linear regression.

R^2 is an important summary statistic for a regression model, and it is one of the first things that a data analyst will typically look at. Models with higher values of R^2 are

“better” in the sense that they can predict future values of the response more precisely. But the values of R^2 that we can expect will vary greatly from one application to another. For a psychologist who is trying to predict human attitudes or behaviors, a model with R^2 of 30-40% may be regarded as unusually strong, because human attitudes and behaviors are in general very difficult to predict. On the other hand, for an engineer who is trying to predict some aspect of a physical system where the measurements are very precise, a model with $R^2 = .98$ might be unusually weak.

Whenever another predictor is added to the model, the new vector of fitted values \hat{y} is never farther away from y than the old \hat{y} . As more predictors are introduced, R^2 never goes down, and it typically goes up. We can make R^2 larger and larger by simply adding more predictors, even if they have no real value in predicting the response. For this reason, R is not a good measure for comparing the fit of alternative models with different numbers of predictors. But it is easy to construct an adjusted version of R^2 that does not necessarily increase with p . Criteria for comparing models will be discussed in a future lecture.

Partial F-tests. The theorem that leads to the omnibus F-test can be generalized in the following way. Suppose we partition the $n \times p$ design matrix as

$$X = [1, X_1, X_2],$$

where 1 is a column of ones, X_1 is $(n \times p_1)$, and X_2 is $n \times p_2$. And suppose we partition the coefficients as $\beta = (\beta_0, \beta_1^T, \beta_2^T)^T$, where $\dim(\beta_1) = p_1$ and $\dim(\beta_2) = p_2$. And suppose we want to test $H_0 : \beta_2 = 0$. That is, we are comparing the fit of the “full model”

$$y = 1\beta_0 + X_1\beta_1 + X_2\beta_2 + \epsilon$$

to that of the “reduced model”

$$y = 1\beta_0 + X_1\beta_1 + \epsilon.$$

To compare these models, suppose we first regress y on $[1, X_1]$ and obtain the fitted values, which we call $\hat{y}_{Reduced}$. Then suppose we regress y on $X = [1, X_1, X_2]$ and obtain the fitted values, which we call \hat{y}_{Full} . Define the regression sum of squares due to X_2 given X_1 as

$$\begin{aligned} SS_{Reg}(X_2 | X_1) &= \|\hat{y}_{Full} - \hat{y}_{Reduced}\|^2 \\ &= \sum_{i=1}^n (\hat{y}_{i, Full} - \hat{y}_{i, Reduced})^2. \end{aligned}$$

This measures of how much the fitted values change when X_2 is added to the model with X_1 already in. It is also called the “partial sum of squares for X_2 given X_1 .” In regression analysis, the word “partial” indicates the effect of a predictor or group of predictors in the presence of other predictors. The difference vector $\hat{y}_{Full} - \hat{y}_{Reduced}$ lies within $\mathcal{R}(X)$, but it is orthogonal to $\mathcal{R}(1, X_1)$, the space spanned by the columns of the design matrix for the reduced model.

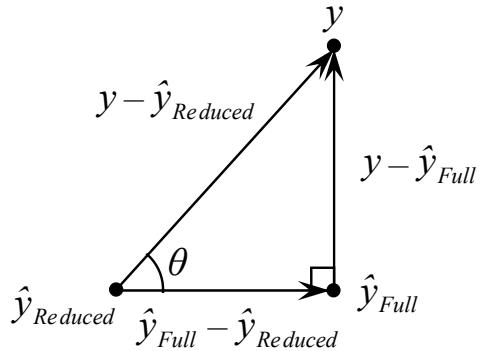
Denote the error sum of squares for the full model by

$$\begin{aligned} SS_{Err}(X_1, X_2) &= \|y - \hat{y}_{Full}\|^2 \\ &= \sum_{i=1}^n (y_i - \hat{y}_{i, Full})^2, \end{aligned}$$

and the error sum of squares for the reduced model by

$$\begin{aligned} SS_{Err}(X_1) &= \|y - \hat{y}_{Reduced}\|^2 \\ &= \sum_{i=1}^n (y_i - \hat{y}_{i, Reduced})^2. \end{aligned}$$

The test of the null hypothesis $H_0 : \beta_2 = 0$ is based on the right triangle shown below,



and on the following theorem.

Theorem. Under the null hypothesis $H_0 : \beta_2 = 0$, the partial sum of squares for X_2 given X_1 is distributed as

$$SS_{Reg}(X_2 | X_1) \sim \sigma^2 \chi_{p_2}^2$$

and is independent of $SS_{Err}(X_1, X_2)$.

Under H_0 , the F-statistic

$$F = \frac{SS_{Reg}(X_2 | X_1)/p_2}{SS_{Err}(X_1, X_2)/(n - p)}$$

is distributed as $F_{p_2, n-p}$.

We have already seen two special cases of this test,

- **The case where $p_1 = 0$.** If $p_1 = 0$, then X_1 is empty. Then $\hat{y}_{Reduced}$ becomes \bar{y}_1 , and the partial F-test for X_2 becomes the omnibus F-test.
- **The case where $p_2 = 1$.** If $p_2 = 1$, then we are testing the significance of the single predictor X_2 in the presence of all the variables in X_1 . In this case, the partial F-test for X_2 given X_1 becomes equivalent to the t-test based on

$$t = \frac{\hat{\beta}_2}{SE(\hat{\beta}_2)},$$

where $SE(\hat{\beta}_2)$ is the square-root of the diagonal element of $S^2(X^T X)^{-1}$ corresponding to $\hat{\beta}_2$, and t^2 is precisely equal to F .

More generally, if $p_2 > 1$, we can construct the .05-level partial F-test by forming the 95% confidence ellipsoid for β_2 , and checking whether $\beta_2 = 0$ lies within the ellipsoid. The argument is basically the same as for the omnibus F,

as we discussed earlier in this lecture.

Partial F and partial correlation. Earlier in this lecture, we showed how to obtain R^2 from the omnibus F. Suppose we apply the same transformation to a partial F,

$$R^2 = \frac{F}{F + df_2/df_1},$$

where $df_1 = p_2$ and $df_2 = n - p$. Then R^2 becomes **the proportion of variance in y explained by X_2 after we have accounted for X_1** . If $p_2 = 1$, then the square root of this R^2 is (plus or minus) the **partial correlation between y and X_2 given all the variables in X_1** .

In a multiple regression model, it is sometimes useful to convert the t-statistic for each predictor to a partial correlation coefficient using the formula

$$R = \pm \sqrt{\frac{F}{F + df_2}}$$

(because $df_1 = 1$). The sign of R is chosen to agree with the sign of the estimated coefficient. The resulting R is an effect size that measures the strength of the association between the response and the predictor given all the other predictors, and it does not tend to increase with n .

For example, consider the output from our linear regression of $\log(\text{CHOL})$ on $\log(\text{BMI})$ and MORPH .

```

> result <- lm(log.chol ~ log.bmi + morph, data=body)
> summary(result)

Call:
lm(formula = log.chol ~ log.bmi + morph, data = body)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.775122 -0.138062  0.002809  0.139035  0.879628 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.51682   0.07753  58.260 <2e-16 ***
log.bmi     0.05559   0.02488   2.234   0.0256 *  
morph       0.65073   0.06226  10.452 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2115 on 1832 degrees of freedom
Multiple R-Squared: 0.08088,    Adjusted R-squared: 0.07988 
F-statistic: 80.61 on 2 and 1832 DF,  p-value: < 2.2e-16

```

The partial correlation between log(CHOL) and log(BMI) is

$$R = \sqrt{\frac{2.234^2}{2.234^2 + 1832}} = 0.052,$$

and the partial correlation between log(CHOL) and MORPH is

$$R = \sqrt{\frac{10.452^2}{10.452^2 + 1832}} = 0.237.$$

Both of these effects are statistically significant at the .05 level, but MORPH is obviously the stronger predictor.

Neither one, however, explains a large part of the variance of log(CHOL). After accounting for MORPH, log(BMI)

explains only

$$0.052^2 = 0.00271 = 0.271\%$$

of the variance in $\log(\text{CHOL})$. After accounting for $\log(\text{BMI})$, MORPH explains only

$$0.237^2 = 0.0562 = 5.62\%$$

of the variance in $\log(\text{CHOL})$.

The sequential decomposition. The F-test for X_1 in a model without X_2 is based on the decomposition

$$SS_{Tot} = SS_{Reg}(X_1) + SS_{Err}(X_1).$$

The F-test for X_2 in the presence of X_1 is based on the decomposition

$$SS_{Err}(X_1) = SS_{Reg}(X_2 | X_1) + SS_{Err}(X_1, X_2).$$

Putting these together, we get

$$SS_{Tot} = SS_{Reg}(X_1) + SS_{Reg}(X_2 | X_1) + SS_{Err}(X_1, X_2).$$

We can arrange the various SS's into an ANOVA table, like this:

Source	SS	df	$MS = SS/df$
1. X_1	$SS_{Reg}(X_1)$	p_1	$MS_{Reg}(X_1)$
2. $X_2 X_1$	$SS_{Reg}(X_2 X_1)$	p_2	$MS_{Reg}(X_2 X_1)$
3. Error	$SS_{Err}(X_1, X_2)$	$n - p$	$MS_{Err}(X_1, X_2)$
Total	SS_{Tot}	$n - 1$	

Notice that

$$SS_{Reg}(X_1, X_2) = SS_{Reg}(X_1) + SS_{Reg}(X_2 | X_1).$$

That is, the regression sum of squares for the model with two (sets of) predictors, X_1 and X_2 , can be broken up into

- the regression sum of squares for introducing X_1 into the model containing no predictors, plus
- the regression sum of squares for introducing X_2 into the model containing X_1 .

So the ANOVA table shown above is just a more detailed version of this table:

Source	SS	df	$MS = SS/df$
X_1, X_2	$SS_{Reg}(X_1, X_2)$	$p_1 + p_2$	$MS_{Reg}(X_1, X_2)$
Error	$SS_{Err}(X_1, X_2)$	$n - p$	$MS_{Err}(X_1, X_2)$
Total	SS_{Tot}	$n - 1$	

The more detailed version is helpful, because it allows us to perform a greater variety of F-tests.

Various F-tests. First, we can perform the omnibus test for X_1 and X_2 . This is a test of the null hypothesis

$H_0 : \beta_1 = \beta_2 = 0$ against the alternative that at least one element of β_1 or β_2 is nonzero. That is, it tests the joint significance of all predictors simultaneously. Putting it yet another way, it tests the null model

$$H_0 : y = 1\beta_0 + \epsilon$$

against the alternative

$$H_1 : y = 1\beta_0 + X_1\beta_1 + X_2\beta_2 + \epsilon.$$

The test statistic is

$$F = \frac{\frac{SS_{Reg}(X_1) + SS_{Reg}(X_2 | X_1)}{(p_1 + p_2)}}{\frac{SS_{Err}(X_1, X_2)}{(n - p)}},$$

which is distributed as $F_{p_1+p_2, n-p}$ under H_0 . An easy way to remember this test is

$$F = \frac{(\text{Line 1}) + (\text{Line 2})}{(\text{Line 3})}.$$

In this shorthand notation, “(Line 3)” is the mean square from line 3 of the ANOVA table, which is the error sum of squares divided by its degrees of freedom (i.e., the mean-squared error or S^2). “(Line 1) + (Line 2)” is the mean square from lines 1 and 2 combined, which is obtained by adding the SS’s from lines 1 and 2, adding the df’s from lines 1 and 2, and dividing the total SS by the total df.

Next is the partial F-test for X_2 given X_1 . This is a test of the null hypothesis $H_0 : \beta_2 = 0$ against the alternative that at least one element of β_2 is nonzero, without assuming anything about β_1 . That is, it tests the null model

$$H_0 : y = 1\beta_0 + X_1\beta_1 + \epsilon$$

against the alternative

$$H_1 : y = 1\beta_0 + X_1\beta_1 + X_2\beta_2 + \epsilon.$$

The test statistic is

$$F = \frac{\frac{SS_{Reg}(X_2 | X_1)}{p_2}}{\frac{SS_{Err}(X_1, X_2)}{(n - p)}},$$

which is distributed as $F_{p_2, n-p}$ under H_0 . An easy way to remember this test is

$$F = \frac{(\text{Line 2})}{(\text{Line 3})}.$$

Testing the effect of X_1 . What does the ANOVA table

Source	SS	df	MS = SS/df
1. X_1	$SS_{Reg}(X_1)$	p_1	$MS_{Reg}(X_1)$
2. $X_2 X_1$	$SS_{Reg}(X_2 X_1)$	p_2	$MS_{Reg}(X_2 X_1)$
3. Error	$SS_{Err}(X_1, X_2)$	$n - p$	$MS_{Err}(X_1, X_2)$
Total	SS_{Tot}	$n - 1$	

tell us about X_1 ? Without further information, it does not allow us to test the null hypothesis $H_0 : \beta_1 = 0$ without making assumptions about β_2 . That is, it does not allow us to test the null model

$$H_0 : y = 1\beta_0 + X_2\beta_2 + \epsilon$$

against the alternative

$$H_1 : y = 1\beta_0 + \beta_1 X_1 + X_2\beta_2 + \epsilon.$$

To perform that test, we would have to decompose the regression sum of squares like this,

$$SS_{Reg}(X_1, X_2) = SS_{Reg}(X_2) + SS_{Reg}(X_1 | X_2),$$

introducing X_2 into the model before X_1 . **The order in which predictors are brought into the model does matter.** In general, the regression sum of squares due to X_1 alone is not the same as the regression sum of squares due to X_1 given X_2 ,

$$SS_{Reg}(X_1) \neq SS_{Reg}(X_1 | X_2).$$

(There is one situation where the two are the same; we will discuss this next time.) So, in order to test the significance of X_1 in the presence of X_2 , we would have to rearrange the ANOVA table like this,

Source	SS	df	$MS = SS/df$
1*. X_2	$SS_{Reg}(X_2)$	p_2	$MS_{Reg}(X_2)$
2*. $X_1 X_2$	$SS_{Reg}(X_1 X_2)$	p_1	$MS_{Reg}(X_1 X_2)$
3. Error	$SS_{Err}(X_1, X_2)$	$n - p$	$MS_{Err}(X_1, X_2)$
Total	SS_{Tot}	$n - 1$	

and then compute

$$F = \frac{(\text{Line } 2^*)}{(\text{Line } 3)}.$$

In certain cases, however, there are some tests that we can apply to X_1 using the original table with X_1 entered first. We will discuss these next time.

MORE ABOUT HYPOTHESIS TESTING

We have been exploring the properties of the general linear regression model

$$y \sim N(X\beta, \sigma^2 I),$$

where the design matrix is partitioned as $X = [1, X_1, X_2]$ and the coefficients are partitioned as $\beta = (\beta_0, \beta_1^T, \beta_2^T)$. We created the ANOVA table with X_1 entered first.

<i>Source</i>	<i>SS</i>	<i>df</i>	<i>MS</i> = <i>SS/df</i>
1. X_1	$SS_{Reg}(X_1)$	p_1	$MS_{Reg}(X_1)$
2. $X_2 X_1$	$SS_{Reg}(X_2 X_1)$	p_2	$MS_{Reg}(X_2 X_1)$
3. Error	$SS_{Err}(X_1, X_2)$	$n - p$	$MS_{Err}(X_1, X_2)$
Total	SS_{Tot}	$n - 1$	

We tested the null model

$$H_0 : y = 1\beta_0 + \epsilon$$

against the alternative

$$H_1 : y = 1\beta_0 + X_1\beta_1 + X_2\beta_2 + \epsilon.$$

using

$$F = \frac{(\text{Line 1}) + (\text{Line 2})}{(\text{Line 3})}. \quad (1)$$

And we tested

$$H_0 : y = \beta_0 + X_1\beta_1 + \epsilon$$

against

$$H_1 : y = \beta_0 + X_1\beta_1 + X_2\beta_2 + \epsilon$$

using

$$F = \frac{(\text{Line 2})}{(\text{Line 3})}. \quad (2)$$

But can this decomposition help us to test hypotheses about the effects due to X_1 ? The answer is, “It depends.” It depends on the relationship between X_1 and X_2 , and on the effect of X_2 on y . There are three different situations where we can perform meaningful tests about β_1 based on the sequential decomposition with X_1 entered first.

Situation 1: When X_1 and X_2 are uncorrelated.

Let’s suppose that X_1 and X_2 are uncorrelated, in the sense that every column in X_1 is uncorrelated with every column in X_2 . In this special case, it is possible to show that the sum of squares due to X_1 is the same whether or not X_2 is present, and vice-versa:

$$SS_{Reg}(X_1 | X_2) = SS_{Reg}(X_1),$$

$$SS_{Reg}(X_2 | X_1) = SS_{Reg}(X_2).$$

Then the regression SS decomposes without ambiguity into

a part due to X_1 and another part due to X_2 :

$$SS_{Reg}(X_1, X_2) = SS_{Reg}(X_1) + SS_{Reg}(X_2).$$

Then we can write the ANOVA table as

Source	SS	df	$MS = SS/df$
1. X_1	$SS_{Reg}(X_1)$	p_1	$MS_{Reg}(X_1)$
2. X_2	$SS_{Reg}(X_2)$	p_2	$MS_{Reg}(X_2)$
3. Error	$SS_{Err}(X_1, X_2)$	$n - p$	$MS_{Err}(X_1, X_2)$
Total	SS_{Tot}	$n - 1$	

and perform separate tests for X_1 ,

$$F = \frac{(\text{Line 1})}{(\text{Line 3})},$$

and for X_2 ,

$$F = \frac{(\text{Line 2})}{(\text{Line 3})}.$$

Inferences about β_1 are unaffected by assumptions about β_2 and vice-versa, because $\hat{\beta}_1$ and $\hat{\beta}_2$ are independent.

A situation where some columns of X are precisely uncorrelated with other columns rarely happens by accident. It tends to arise in designed experiments, where the effects of multiple factors on a response are being assessed simultaneously. We will learn about this in Stat 512. It may also arise with other types of data, if the analyst intentionally creates columns that are uncorrelated with other columns. We will discuss that next week, when

we consider orthogonal polynomials.

Situation 2: If we can assume that $\beta_2 = 0$. If $\beta_2 = 0$ is true, our theorem for partial SS says that

$$SS_{Reg}(X_2 | X_1) \sim \sigma^2 \chi_{p_2}^2$$

independently of $SS_{Err}(X_1, X_2)$. Under the assumption $\beta_2 = 0$, we can combine Lines 2 and 3 from the ANOVA table into

$$\begin{aligned} SS_{Err}(X_1) &= SS_{Reg}(X_2 | X_1) + SS_{Err}(X_1, X_2) \\ &\sim \sigma^2 \chi_{n-p+p_2}^2, \end{aligned} \tag{3}$$

the error term from the regression of y on X_1 alone. Then we can test the null hypothesis $H_0 : \beta_1 = 0$ against the alternative that at least one element of β_1 is nonzero by comparing

$$F = \frac{(\text{Line 1})}{(\text{Line 2}) + (\text{Line 3})} \tag{4}$$

to an F distribution with p_1 and $(n - p + p_2)$ degrees of freedom.

In this case, we could have also compared

$$F = \frac{(\text{Line 1})}{(\text{Line 3})} \tag{5}$$

to an F distribution with p_1 and $(n - p)$ degrees of freedom. Both statistics (4) and (5) are valid for testing the null hypothesis $H_0 : \beta_1 = 0$ under the assumption that

$\beta_2 = 0$. But (4) has a slight advantage over (5), because it has more degrees of freedom in the error term; the denominator of (4) is more stable, producing a test that is slightly more powerful.

When is it safe to assume that $\beta_2 = 0$? It might be tempting to test $H_0 : \beta_2 = 0$ by the partial F-test (2) and, if the result is not significant at the .05 level, assume that $\beta_2 = 0$ and collapse Line 2 into Line 3. But if we fail to reject a null hypothesis, that does not mean it is safe to assume that the null hypothesis is true. If we collapse Line 2 into Line 3 when β_2 is not zero, the distributional result (3) will no longer hold, and the overall MS from Lines 2 and 3 combined will no longer be an unbiased estimate of σ^2 . Some have suggested that it may be safe to collapse Line 2 into Line 3 if the partial F statistic (2) is less than 2.

If $\beta_2 \neq 0$, then the denominator of the statistic (5) is still an unbiased estimate of σ^2 . But the meaning of the numerator is not clear. It usually does not make sense to talk about “the effect of X_1 alone” in a model that contains X_2 as well (unless the two are uncorrelated, as we have discussed).

Situation 3: When X_2 will be unavailable or is not of interest. There is another situation where it makes sense to talk about the effect of X_1 apart from X_2 , even though X_2 may be related to y . It is when we want to assess the marginal relationship between X_1 and y without

regard for X_2 .

Consider the regression model without X_2 ,

$$y = \beta_0 + X_1\beta_1 + \epsilon, \quad (6)$$

where $\epsilon \sim N(0, \sigma^2 I)$. We can interpret this model in two ways.

- As a model for the conditional distribution of y given X_1 and X_2 ,

$$E(y | X_1, X_2) = \beta_0 + X_1\beta_1 + X_2\beta_2,$$

under the assumption that $\beta_2 = 0$.

- As a model for the conditional distribution of y given X_1 alone,

$$E(y | X_1) = \beta_0 + X_1\beta_1,$$

where X_2 is unavailable or not of interest.

For example, if we need to build a model to predict future values of y , and the variables in X_2 will not be available for these predictions, then it could make sense to use a model of the form (6) even though some or all of the variables in X_2 may be related to y .

If we decide to omit X_2 for these reasons, then we may have a model of the form (6) regardless of how y is related to X_2 . We may define σ^2 to be the variance of the response given X_1 , rather than the variance given X_1 and X_2 . An unbiased estimate of this σ^2 will be the MS from

Lines 2 and 3 combined. And

$$F = \frac{(\text{Line 1})}{(\text{Line 2}) + (\text{Line 3})}$$

will become a valid statistic for testing $H_0 : \beta_1 = 0$.

This raises important theoretical questions about what a regression model really means. In nearly every regression analysis, potentially important predictors are omitted from the model, either because these variables are unavailable to us, or because including them would be impractical (e.g., because n might not be large enough to support a model with so many parameters). If these omitted variables are also related to the predictors in our model (and they often are), then the relationships between the response and the predictors in the model may not accurately describe causal relationships in the population. The coefficients β_1 may not accurately describe the true causal effects of X_1 on y . The regression of y on X_1 may still be useful for prediction, but it should not be interpreted as a causal model. We will discuss causal inference later in this semester.

ANOVA tables in R. The function `anova()`, when applied to the result of a call to `lm()`, produces an ANOVA table with sequential sums of squares. The order in which the variables are entered into the model corresponds to the order of the predictors in the `formula` argument to `lm()`.

Let's apply this to our body measurements dataset, for the

regression of log(CHOL) on log(BMI) and MORPH.

```
> body <- read.table("body.dat", header=T)

> # create new variables for regression
> meters <- body$height * 2.54 / 100 # height in meters
> kg <- body$weight * 0.45359237 # weight in pounds
> body$log.bmi <- log(kg/meters^2)
> body$log.chol <- log( body$chol )
> body$morph <- body$waist / body$hips

> result <- lm( log.chol ~ log.bmi + morph, data=body)
> anova(result)
Analysis of Variance Table

Response: log.chol
  Df Sum Sq Mean Sq F value    Pr(>F)
log.bmi     1  2.325   2.325  51.967 8.216e-13 ***
morph       1  4.887   4.887 109.249 < 2.2e-16 ***
Residuals 1832 81.948   0.045
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Examining Line 2, we see that the regression SS for MORPH given log(BMI) is 4.887, and the partial F statistic

$$F = \frac{(\text{Line 2})}{(\text{Line 3})} = \frac{4.887}{0.045} = 109.249$$

is highly significant. If we had printed out the table of coefficients with the `summary()` command, we would have seen that the t-statistic for MORPH is $\sqrt{109.249} = 10.45$. Therefore, MORPH is a significant predictor of log(CHOL) even after accounting for their mutual associations with log(BMI).

Examining Line 1, we see that the regression SS for $\log(\text{BMI})$ is 2.325, and the reported F statistic is 51.967 and it is highly significant. Where did this F come from? The F statistic for comparing Line 1 to Line 3 is

$$F = \frac{2.325}{81.948/1832} = 51.977,$$

which is a little different from the reported F. But the F statistic for comparing Line 1 to Lines 2+3 is

$$F = \frac{2.325}{(81.948 + 4.887)/1833} = 49.078,$$

which is much farther away. So we conclude that the F reported in Line 1 is a comparison of Line 1 to Line 3, and the discrepancy is due to rounding error. What does this test mean? It means that $\log(\text{BMI})$ is a significant predictor of $\log(\text{CHOL})$ under the assumption that the coefficient of MORPH is zero. But the coefficient of MORPH is clearly nonzero, as shown by the partial F-test. So the F-test in Line 1 is not very meaningful.

To get the partial F test for the effect of $\log(\text{BMI})$ given MORPH, we need to enter the predictors in the reverse order.

```
> result <- lm(log.chol ~ morph + log.bmi, data=body)
> anova(result)
Analysis of Variance Table

Response: log.chol
          Df  Sum Sq Mean Sq  F value    Pr(>F)
morph       1   6.988   6.988 156.2243 < 2e-16 ***

```

```

log.bmi      1  0.223   0.223   4.9919  0.02559 *
Residuals 1832 81.948   0.045
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The new F-statistic for MORPH is 4.9919, which corresponds to the t-statistic of $\sqrt{4.9919} = 2.234^2$. In the presence of MORPH, log(BMI) is still a significant predictor of log(CHOL), but it is not nearly as powerful as MORPH.

Correlated predictors and shared significance. The discrepancy between

$$SS_{Reg}(\log(\text{BMI})) = 2.325$$

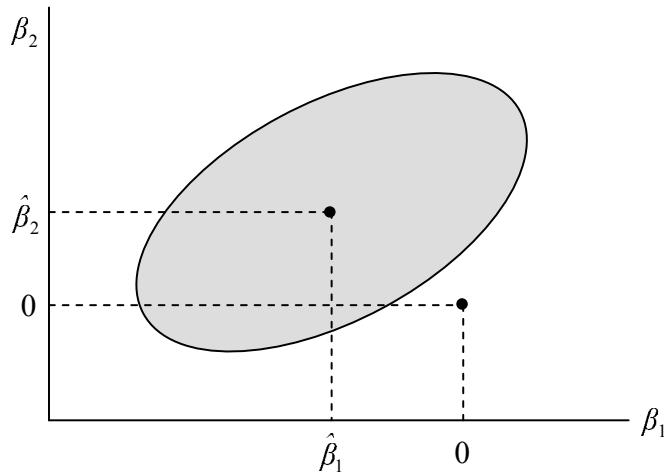
and

$$SS_{Reg}(\log(\text{BMI}) \mid \text{MORPH}) = 0.223$$

suggests that the two predictors are correlated. Having MORPH in the model greatly diminishes the apparent effect of log(BMI), bringing its F statistic down from 51.967 to just 4.99. And having log(BMI) in the model also diminishes the effect of MORPH, bringing its F statistic down from 156.2243 to 109.249 (which is still highly significant). The two variables “share significance” because they are correlated with each other. Part of the correlation between log(CHOL) and log(BMI) can be attributed to MORPH, and part of the correlation between log(CHOL) and MORPH can be attributed to log(BMI). The effects of the two predictors cannot be entirely separated.

Regardless of which predictor is entered first, the SS_{Reg}

for both predictors combined is 7.211. If $\log(\text{BMI})$ is entered first, it accounts for 2.325 or 32% of the SS_{Reg} , and MORPH accounts for the remaining 4.887 or 68%. If MORPH is entered first, it accounts for 6.988 or 97% of the SS_{Reg} , and $\log(\text{BMI})$ accounts for the remaining 0.223 or 3%. In this case, each predictor is significant in the presence of the other. But more extreme situations may arise where X_1 and X_2 are significant jointly (i.e. we can reject $H_0 : \beta_1 = \beta_2 = 0$) but not significant individually (we cannot reject $H_0 : \beta_1 = 0$ or $H_0 : \beta_2 = 0$). If this happens, it indicates that X_1 and X_2 are highly correlated, causing $\hat{\beta}_1$ and $\hat{\beta}_2$ to be highly correlated. In the special case where $p_1 = p_2 = 1$, we can visualize the situation as follows: The individual confidence intervals for β_1 and β_2 both cover zero, but the elliptical 95% joint confidence region for β_1 and β_2 misses the point $(0, 0)$.



Sequential ANOVA table for many predictors.

Consider the regression model $y \sim N(X\beta, \sigma^2 I)$ where

$$X = [1, X_1, X_2, \dots, X_{p-1}]$$

denotes a partitioning of X into its columns, so that

$$E(y) = 1\beta_0 + X_1\beta_1 + X_2\beta_2 + \cdots + X_{p-1}\beta_{p-1}.$$

If we fit this model in R (or just about any other statistical package), then by default the ANOVA table will list the sequential sums of squares corresponding to the order in which the predictors were listed in the regression command. If you fit the model like this,

```
> result <- lm( y ~ x1 + x2 + x3 + x4, data=mydata )
```

then the ANOVA table will divide the regression sum of squares as

$$\begin{aligned} SS_{Reg}(X_1, X_2, X_3, X_4) &= SS_{Reg}(X_1) \\ &\quad + SS_{Reg}(X_2 | X_1) \\ &\quad + SS_{Reg}(X_3 | X_1, X_2) \\ &\quad + SS_{Reg}(X_4 | X_1, X_2, X_3). \end{aligned}$$

The software may provide F-tests and p-values for each of the predictors. But if the predictors are correlated (and they usually are), then the only F test in this ANOVA table that is readily interpretable is the partial F-test for the last predictor (in this case, X_4). The meaning of the sequential tests for the other predictors (X_1 , X_2 and X_3)

may be dubious, because the sequential F-test for β_j assumes that $\beta_{j'} = 0$ for $j' > j$.

Sequential sums of squares are sometimes called “Type 1” sums of squares. The terminology of “Type x ” sums of squares for $x = 1, 2, 3$ or 4 is old-fashioned, but it is still heavily used in the documentation for certain procedures in SAS. Outside of the SAS Institute, few statisticians can tell you what Type 2 or Type 4 sums of squares are. But everyone seems to know that

Type 1 = sequential SS, and

Type 3 = partial SS.

By default, most regression programs will print out the Type 1 or sequential sums of squares in the sequence that the predictors were specified by the user. But, depending on the software, you may be able to request an ANOVA table containing Type 3 or partial sums of squares instead. If you have four regressors, you might receive a table with lines corresponding to

$$\begin{aligned}SS_{Reg}(X_1 | X_2, X_3, X_4), \\SS_{Reg}(X_2 | X_1, X_3, X_4), \\SS_{Reg}(X_3 | X_1, X_2, X_4), \\SS_{Reg}(X_4 | X_1, X_2, X_3).\end{aligned}$$

These would not add up to the overall SS_{Reg} . But the F-statistic on each line would then be a partial F for

testing $H_0 : \beta_j = 0$ without assuming anything about the other coefficients.

Testing general linear hypotheses. The null hypothesis that any individual element of β is zero (without assuming anything about the other elements) may be tested by examining the t-statistic for that coefficient. This t-test is equivalent to the partial F-test in which the variable in question is entered last. The null hypothesis that a group of coefficients within β are simultaneously zero may be tested by the partial F-test in which the group of variables in question is entered last. My point is that any hypothesis of the form

$$H_0 : \text{some elements of } \beta \text{ are zero} \quad (7)$$

can be tested by comparing the fit of

- the full model with all the predictors, and
- the reduced model in which the predictors in question have been removed.

The difference in SS_{Reg} between the full and reduced models (when divided by the appropriate df) gives the numerator of the partial F statistic, and the SS_{Err} from the full model (again, divided by its df) gives the denominator.

But we may occasionally be interested in testing hypotheses that are not of the form (7). Using normal

theory, we can easily test more general linear hypotheses of the form

$$H_0 : A\beta = c,$$

where A is a known $(q \times p)$ matrix, and c is a known $q \times 1$ vector. If we write

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_q^T \end{bmatrix}, \quad c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_q \end{bmatrix},$$

where each a_j is a $p \times 1$ vector and each c_j is a constant, then H_0 can be expressed as

$$a_1^T \beta = c_1,$$

$$a_2^T \beta = c_2,$$

$$\vdots$$

$$a_q^T \beta = c_q.$$

The linear combinations of coefficients $a_j^T \beta$ are often called **contrasts**.

By choosing appropriate forms for A and c , we can formulate and test many kinds of hypotheses. For example, suppose that

$$\beta = (\beta_1, \beta_2, \dots, \beta_5)^T,$$

and we want to test the combined null hypothesis

$$H_0 : \beta_1 = -\beta_2 = \frac{\beta_3 + \beta_4}{2}.$$

We could express this hypothesis as $A\beta = c$, with

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1/2 & 1/2 & 0 \end{bmatrix}$$

and $c = (0, 0)^T$.

We test the general linear hypothesis $H_0 : A\beta = c$ as follows. From the result

$$\hat{\beta} \sim N(\beta, \sigma^2(X^T X)^{-1}),$$

it follows that

$$A\hat{\beta} \sim N(A\beta, \sigma^2 A(X^T X)^{-1} A^T).$$

If H_0 were true, then

$$A\hat{\beta} \sim N(c, \sigma^2 A(X^T X)^{-1} A^T),$$

and then

$$\frac{1}{\sigma^2} (A\hat{\beta} - c)^T \left[A(X^T X)^{-1} A^T \right]^{-1} (A\hat{\beta} - c) \sim \chi_q^2. \quad (8)$$

(Technically speaking, (8) also requires that the rows of A be linearly independent, i.e. that

$$\text{rank}(A^T) = q,$$

which requires $q \leq p$. If A^T is rank-deficient, then the inverse in (8) won't exist. A rank-deficiency in A^T

indicates that at least one contrast is redundant.) If we divide (8) by

$$\frac{S^2}{\sigma^2} \sim \chi_{n-p}^2 / (n - p),$$

then we obtain

$$\frac{1}{S^2} (A\hat{\beta} - c)^T \left[A(X^T X)^{-1} A^T \right]^{-1} (A\hat{\beta} - c) \sim q F_{q, n-p}. \quad (9)$$

If the statistic on the left-hand side of (9) exceeds q times the 95th percentile of $F_{q, n-p}$, then we can reject H_0 at the .05 level. If the test concerns just one contrast ($q = 1$), then the square root of the left-hand side of (9) becomes a t-statistic with $n - p$ degrees of freedom.

We can rewrite (9) as

$$F = \frac{SS/q}{S^2} \sim F_{q, n-p},$$

where

$$SS = (A\hat{\beta} - c)^T \left[A(X^T X)^{-1} A^T \right]^{-1} (A\hat{\beta} - c). \quad (10)$$

We can regard (10) as the portion of the overall regression sum of squares attributable to departures from H_0 . If we were to fit the linear regression model $y \sim N(X\beta, \sigma^2 I)$ subject to the constraint that $A\beta = c$, the regression sum of squares from that constrained model would be equal to SS_{Reg} from the full model, minus the quantity (10). Thus

we can regard this as a more general version of the partial F-test, where the reduced model is the model with the constraint $A\beta = c$, and the full model is the model without the constraint.

If the matrix

$$\left[A(X^T X)^{-1} A^T \right]^{-1}$$

happens to be diagonal, then the contrasts in $A\beta$ are said to be **orthogonal**. In that case, the extra SS in (10) could be partitioned into independent pieces corresponding to departures from the individual hypotheses $a_j^T \beta = c_j$, $j = 1, \dots, q$. That is, we could write

$$SS = SS_1 + SS_2 + \dots + SS_q,$$

where

$$SS_j = \frac{(a_j^T \hat{\beta} - c_j)^2}{a_j^T (X^T X)^{-1} a_j}$$

is the portion of the sum of squares due to contrast j . Orthogonal contrasts are nice for the following reason. If the combined null hypothesis $H_0 : A\beta = c$ is rejected, then we can look at the individual contrasts to see which of the hypotheses $a_j^T \beta = c_j$ are plausible and which are not, and our conclusion about any one hypothesis will be unaffected by conclusions about the other hypotheses. When the constraints are not orthogonal, we can still test them one at

a time. The test for the j th contrast will be based on

$$T = \frac{a_j^T \hat{\beta} - c_j}{S \sqrt{a_j^T (X^T X)^{-1} a_j}},$$

which we compare to a t-distribution with $n - p$ degrees of freedom. But the tests for the individual contrasts will be related, just as the test for the significance of one coefficient will depend on what we assume about the other coefficients.

ORTHOGONALIZATION AND THE QR DECOMPOSITION

Today we are going to consider how a software package such as R actually does the computations when fitting a regression model.

A naive way to compute $\hat{\beta} = (X^T X)^{-1} X^T y$ would be

- compute the matrix $X^T X$ and the vector $X^T y$,
- invert $X^T X$, and
- multiply $(X^T X)^{-1}$ by $X^T y$.

This may not be a bad way to do it, particularly if the inversion technique takes advantage of the fact that $X^T X$ is symmetric and positive definite. In practice, however, a well designed regression program typically does not do this. Rather, the program will transform the columns of the design matrix X to simplify the rank- p regression problem into a sequence of independent rank-one regressions. (A rank-one regression is a regression of y onto a single column, i.e. a projection of one vector onto another.) The advantages of such a transformation are

- computational efficiency,

- greater numerical stability when $X^T X$ is nearly singular, and
- the ability to handle situations where $\text{rank}(X) < p$ (but we won't discuss this now).

Learning about this will give us greater insight into the nature of least-squares regression and what we can do when the columns of X are highly correlated.

Orthogonalization. A matrix X is said to be orthogonal if every column is orthogonal to every other column, i.e. if $X^T X$ is diagonal. The matrix is said to be **orthonormal** if the magnitude (Euclidean length) of each column is one, i.e. if $X^T X = I$. If the design matrix X in a regression problem happened to be orthonormal, then the least-squares estimates would simply become $\hat{\beta} = X^T y$. Design matrices rarely have this property. Given an arbitrary X , however, it is possible to find another matrix Q with the same dimensions as X ($n \times p$), whose columns span the same space as X (so that $\mathcal{R}(Q) = \mathcal{R}(X)$), but which is orthonormal ($Q^T Q = I$).

One way to do this is by **replacing each column of X by the (rescaled) residuals from the regression of that column on all preceding columns.** Let's write the original matrix as

$$X = [X_1, X_2, \dots, X_p],$$

And, using a notation similar to the R language, let

$$X[:, j:k] = [X_j, X_{j+1}, \dots, X_k]$$

denote the submatrix consisting of columns j through k . A single column will be written as

$$X_j = X[:, j:j] = X[:, j].$$

Suppose we transform X in the following way.

1. Replace X_1 by $X_1 / \sqrt{X_1^T X_1}$, so that the new version of X_1 has a magnitude of one.
2. Replace X_2 by the residuals from the regression of X_2 on X_1 , and then divide the new X_2 by $\sqrt{X_2^T X_2}$ so that it has a magnitude of one.
3. Replace X_3 by the residuals from the regression of X_3 on $X[:, 1:2]$, and then divide the new X_3 by $\sqrt{X_3^T X_3}$ so that it has a magnitude of one.
4. Continue the procedure for each of the remaining columns.

After this transformation, the new matrix, which we may call Q , is orthonormal, and its columns span the same space as the original X .

This procedure for constructing an orthogonal basis is called the (modified) Gram-Schmidt method. There are many techniques for doing this, but Gram-Schmidt is the

most familiar and easy to understand. To carry out this procedure efficiently, note that design matrix for each of the regressions is orthonormal, so no matrix inversions are required.

The QR decomposition. Regression software will typically decompose the $n \times p$ design matrix X as

$$X = QR,$$

where Q is an $n \times p$ orthonormal matrix and R is a $p \times p$ upper-triangular matrix. The columns of Q span the same space as the original X , and R contains the information needed to transform Q back to X . The fact that R is upper-triangular indicates that

- the first column of X is just a rescaled version of the first column of Q ,
- the second column of X is a linear combination of the first two columns of Q , and so on.

The QR decomposition can be computed by the Gram-Schmidt method described above, but many regression programs (including `lm` and `lsfit` in R) use another technique called “Householder reflections and Givens rotations.”

Using the QR decomposition, the regression model becomes

$$y \sim N(Q\gamma, \sigma^2 I),$$

where $\gamma = R\beta$. The least-squares estimate for γ is

$$\hat{\gamma} = (Q^T Q)^{-1} Q^T y = Q^T y,$$

which requires no matrix inversion. The corresponding estimate for β is the solution to

$$R\beta = Q^T y, \quad (1)$$

which is

$$\hat{\beta} = R^{-1} Q^T y.$$

Because R is upper-triangular, the linear system (1) can be efficiently solved by a backsolve operation.

R has a function called `qr` that performs the QR decomposition on a matrix. This function is also used by `lm`. To see this, let's regress

$$y = \begin{bmatrix} 2.8 \\ 3.2 \\ 7.1 \\ 6.8 \\ 8.8 \end{bmatrix} \quad \text{on} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \end{bmatrix}$$

and look at the results from `lm`.

```
> X1 <- c(1, 1, 1, 1, 1)
> X2 <- c(1, 2, 3, 4, 5)
> X3 <- c(1, 4, 9, 16, 25)
> X <- cbind(X1, X2, X3)
> y <- c(2.8, 3.2, 7.1, 6.8, 8.8)
```

```

> # the -1 in the formula below tells R not to add an intercept
> result <- lm( y ~ -1 + X1 + X2 + X3 )

> result$coef
      X1          X2          X3
0.56000000 1.98857143 -0.07142857

> names(result)
[1] "coefficients"   "residuals"       "effects"        "rank"
[5] "fitted.values"  "assign"         "qr"             "df.residual"
[9] "xlevels"         "call"          "terms"          "model"

```

One of the components of the list, `qr`, contains the results from the QR decomposition. These results are stored in a compact form that is not so easy to understand.

```

> result$qr
$qr
      X1          X2          X3
1 -2.2360680 -6.7082039 -24.59674775
2  0.4472136  3.1622777  18.97366596
3  0.4472136 -0.1954395  3.74165739
4  0.4472136 -0.5116673  0.62985620
5  0.4472136 -0.8278950 -0.04991623
attr(,"assign")
[1] 1 2 3

$qraux
[1] 1.447214 1.120788 1.775106

$pivot
[1] 1 2 3

$tol
[1] 1e-07

$rank
[1] 3

attr(,"class")

```

```
[1] "qr"
```

You can get this same result by calling the `qr` function directly.

```
> tmp <- qr(X)
> tmp
$qr
      X1          X2          X3
[1,] -2.2360680 -6.7082039 -24.59674775
[2,]  0.4472136  3.1622777 18.97366596
[3,]  0.4472136 -0.1954395  3.74165739
[4,]  0.4472136 -0.5116673  0.62985620
[5,]  0.4472136 -0.8278950 -0.04991623

$rank
[1] 3

$qraux
[1] 1.447214 1.120788 1.775106

$pivot
[1] 1 2 3

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

The functions `qr.Q` and `qr.R` will extract the Q and R matrices from this object.

```
> # look at the R matrix
> R <- qr.R(tmp)
> R
      X1          X2          X3
[1,] -2.236068 -6.708204 -24.596748
[2,]  0.000000  3.162278 18.973666
[3,]  0.000000  0.000000  3.741657
```

```

> # look at the Q matrix and verify that it's orthonormal
> Q <- qr.Q( tmp )
> Q
      [,1]          [,2]          [,3]
[1,] -0.4472136 -6.324555e-01  0.5345225
[2,] -0.4472136 -3.162278e-01 -0.2672612
[3,] -0.4472136  1.179070e-17 -0.5345225
[4,] -0.4472136  3.162278e-01 -0.2672612
[5,] -0.4472136  6.324555e-01  0.5345225

> t(Q) %*% Q
      [,1]          [,2]          [,3]
[1,] 1.000000e+00 -5.233986e-17 5.322077e-17
[2,] -5.233986e-17 1.000000e+00 2.321819e-16
[3,] 5.322077e-17 2.321819e-16 1.000000e+00

> # now multiply them to recover the original X
> Q %*% R
      X1  X2  X3
[1,] 1  1  1
[2,] 1  2  4
[3,] 1  3  9
[4,] 1  4 16
[5,] 1  5 25

```

Another useful function is `qr.solve`, which solves the linear system (1).

```

> qr.solve(tmp, y)
      X1          X2          X3
0.56000000  1.98857143 -0.07142857

```

Notice that this agrees with the coefficients from `lm`.

Collinearity. If the columns of X are linearly dependent, then one or more columns can be expressed as linear combinations of other columns. When this happens, $X^T X$

cannot be inverted in the usual way, because the inverse does not exist. In this case, the fitted values \hat{y} and the residuals $\hat{\epsilon} = y - \hat{y}$ exist. But least-squares coefficients are not unique; there are infinitely many solutions $\hat{\beta}$ for which $\hat{y} = X\hat{\beta}$ is the projection of y onto $\mathcal{R}(X)$. When this happens, the regression software should give a warning.

```
> X2 <- -5*X1
> result <- lm( y ~ X1 + X2 )
> summary(result)

Call:
lm(formula = y ~ X1 + X2)

Residuals:
    1     2     3     4     5 
-2.94 -2.54  1.36  1.06  3.06 

Coefficients: (2 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.740      1.171   4.901  0.00804 ***
X1          NA         NA       NA       NA      
X2          NA         NA       NA       NA      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.619 on 4 degrees of freedom
```

In this example, the problem is easily solved by removing the redundant column from X . Some regression software will find and remove the redundant columns automatically.

Difficulties may also arise when a column of X is not exactly, but nearly, a linear combination of other columns. In that case, $X^T X$ is nearly singular, and the least-squares method could become numerically unstable, introducing

large rounding errors into $\hat{\beta}$. Modern regression software tends to be very stable, so rounding errors are not the problem that they used to be. Even if the problem is not numerically unstable, it may be “statistically unstable” in the sense that some parameters are very poorly estimated. When $X^T X$ is nearly singular, one or more diagonal elements of $(X^T X)^{-1}$ may become very large, producing large standard errors.

To see why this happens, recall the simple linear regression model

$$y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2),$$

for which the design matrix is

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}.$$

The $X^T X$ matrix becomes nearly singular when the second column of X becomes nearly proportional to the first column, i.e. when x_1, x_2, \dots, x_n are nearly constant. As the variance of the x_i 's decreases, it becomes increasingly difficult to estimate the slope β_1 , as we recall from the formula

$$V(\hat{\beta}_1) = \sigma^2 / S_{xx}$$

(see Lecture 11). This is intuitively sensible, because β_1 is

supposed to measure the average change in y_i when x_i changes. If the values of x_i do not change much in the observed data, then we have little information to estimate how y_i changes with x_i .

Now consider what happens if we have two predictors, X_1 and X_2 . The design matrix becomes

$$X = [1, X_1, X_2],$$

and $X^T X$ becomes singular if

- X_1 is nearly constant,
- X_2 is nearly constant, or
- X_1 and X_2 are highly correlated.

If X_1 and X_2 are highly correlated, then there exist constants a and b such that

$$X_1 \approx a + bX_2,$$

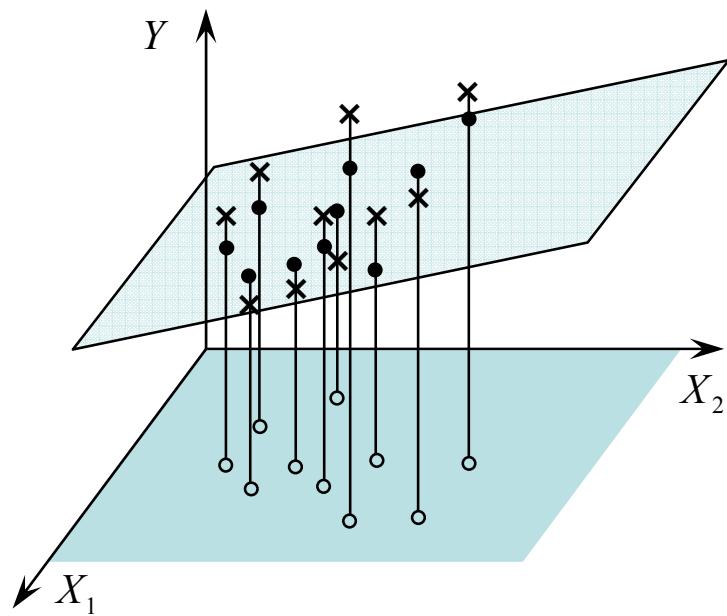
which means that $X_1 - bX_2$ has almost zero variance. In the regression model

$$y \sim N(1\beta_0 + X_1\beta_1 + X_2\beta_2, \sigma^2 I),$$

the coefficient β_1 is the change in the mean response when X_1 increases by one unit, with the value of X_2 held constant. For this parameter to be well estimated, the sample must have variation in the values of X_1 for fixed values of X_2 . But when X_1 and X_2 are highly correlated,

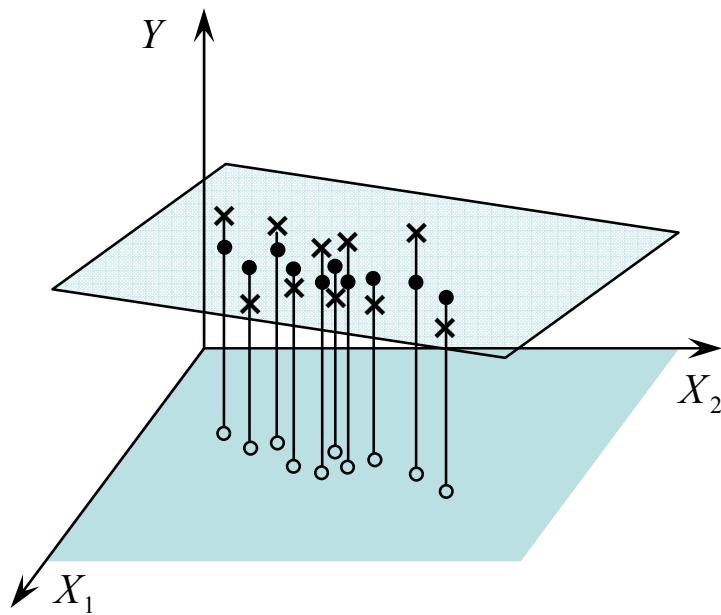
the values of X_1 among units with the same values of X_2 are very similar, so the data contain little information to estimate β_1 . Similarly, the values of X_2 among units with the same values of X_1 are very similar, so the data contain little information to estimate β_2 . The impact of collinearity is that it becomes difficult to separate the effects of X_1 and X_2 on the response and, as a result, the standard errors for the individual coefficients are large.

With two predictors, the regression function is a plane that predicts the value of the response for specific values of X_1 and X_2 . In the picture below, the symbol \times denotes an observation, and a black dot denotes a fitted or predicted value.



In this example, the regression plane is stable, because the predictors X_1 and X_2 are not highly correlated.

When the values of X_1 and X_2 are highly correlated, however, the plane becomes wobbly. It's like trying to balance a sheet of plywood on top of a picket fence.



Although the regression plane wobbles in the direction perpendicular to the fence, it is quite stable in the direction of the fence itself. This is an important point, because certain aspects of the regression relationship can still be estimated well. The height of the regression plane is well estimated in the region of the covariate space **where the observed values of X_1 and X_2 actually occur**. But outside this region—especially as we move away in directions perpendicular to the fence—the estimated height of the regression plane is based on extrapolation, and is not trustworthy.

What should we do? If collinearity is present, and if the purpose of the regression analysis is to interpret the coefficients and understand the effects of X_1 and X_2 on Y , we may have to (a) give up or (b) collect more data. For example, suppose we want to understand the relationships between $X_1 = \text{sex}$ and $X_2 = \text{holding an advanced degree}$ on $Y = \text{income}$. And suppose that none of the women in the sample have advanced degrees. In that case, the effects of sex and advanced education will be impossible to separate. However, if we are able to augment the existing sample with data on some women with advanced degrees, then the problem might be solved.

If the purpose of the analysis is not to ascribe scientific meaning to the coefficients but to predict future values of Y , then collinearity may not be a problem **if the values of X_1 and X_2 for the future predictions lie in the region of the covariate space represented by our sample.** In that region, the height of the regression plane may be well estimated, even though the individual contributions of X_1 and X_2 are not. Outside that region, the predictions will be based on extrapolation.

Extrapolation is not a good idea, because we never know if the estimated relationships hold in that region.

Multicollinearity with many predictors. With only two predictors, the X matrix is

$$X = [1, X_1, X_2],$$

and collinearity can be diagnosed simply by examining the correlation between X_1 and X_2 . With many predictors,

$$X = [1, X_1, X_2, \dots, X_{p-1}],$$

we can compute the correlation matrix for X_1, \dots, X_{p-1} and look for high values. This will reveal situations where one variable is nearly a linear transformation of another. But it may not reveal situations where one variable is nearly a linear combination of two or more other variables, as in

$$X_4 \approx aX_1 + bX_2 + cX_3 + d.$$

This situation is called **multicollinearity**, as opposed to simple collinearity between two variables.

To diagnose multicollinearity, we can regress each of the predictors X_j on all the others ($X_{j'}, j' \neq j$) and compute the R^2 from that regression. A high value for R_j^2 indicates that X_j is nearly redundant.

Sometimes these R^2 values are converted to **variance inflation factors** (VIF's). The j th variance inflation factor is

$$\text{VIF}_j = \frac{1}{1 - R_j^2}.$$

It estimates the factor by which the variance of $\hat{\beta}_j$ is inflated by the correlation between X_j and the other predictors.

LEVERAGE AND INFLUENCE

Example. Over the last two lectures, we have discussed various aspects of hypothesis testing and multicollinearity. Before going on, let's illustrate some of these ideas with a real data example. The data came from

<http://www.statsci.org/data/general/punting.html>

The description is:

Investigators studied physical characteristics and ability in 13 football punters. Each volunteer punted a football ten times. The investigators recorded the average distance for the ten punts, in feet. They also recorded the average hang time (time the ball is in the air before the receiver catches it) for the ten punts, in seconds. In addition, the investigators recorded five measures of strength and flexibility for each punter: right leg strength (pounds), left leg strength (pounds), right hamstring muscle flexibility (degrees), left hamstring muscle flexibility (degrees), and overall leg strength (foot-pounds). From the study “The relationship between selected physical performance variables and football punting ability” by the Department of Health, Physical Education and

*Recreation at the Virginia Polytechnic Institute
and State University, 1983.*

The data file `punting.txt` looks like this.

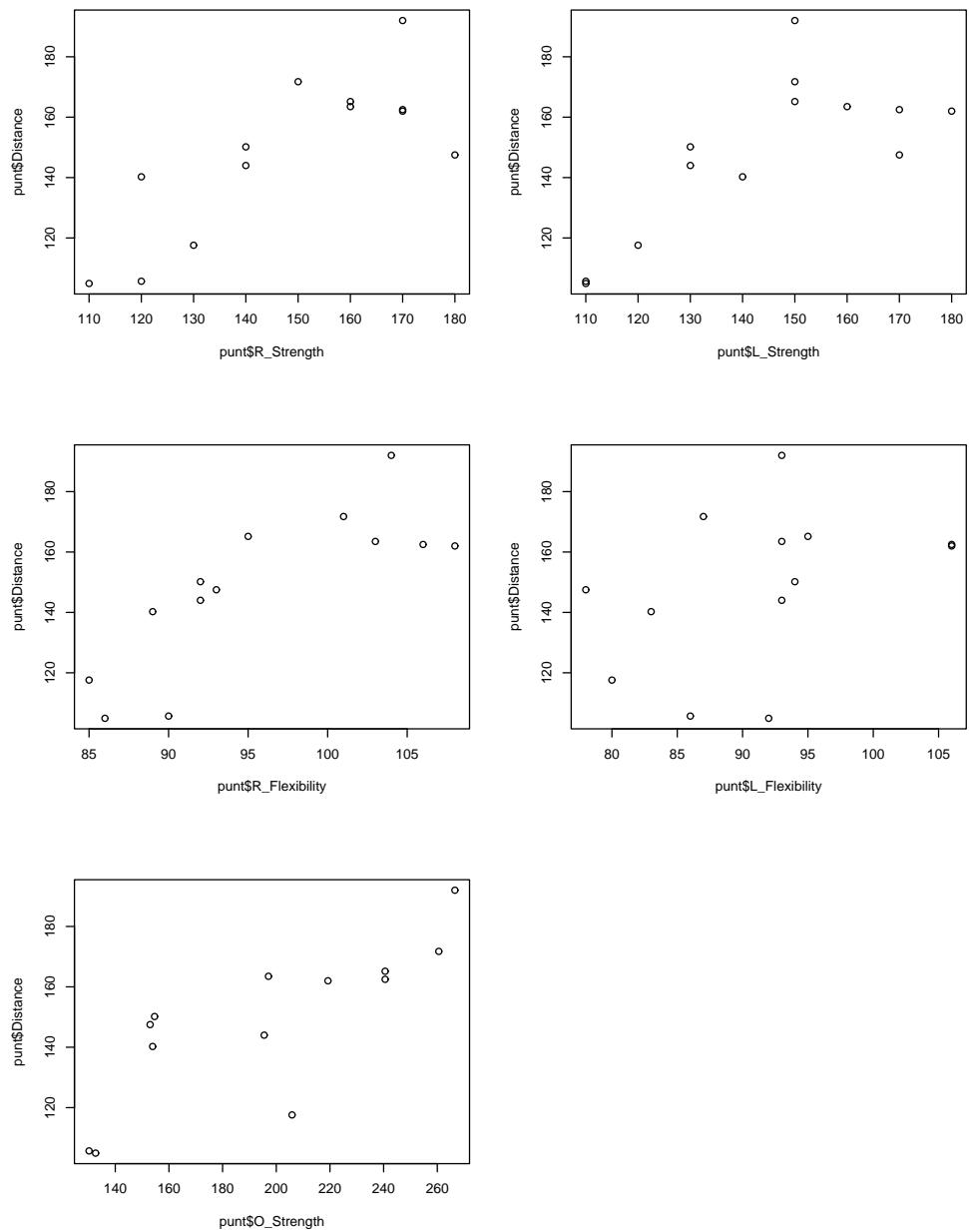
```
Distance Hang R_Strength L_Strength R_Flexibility L_Flexibility O_Strength
162.5 4.75 170 170 106 106 240.57
144 4.07 140 130 92 93 195.49
147.5 4.04 180 170 93 78 152.99
163.5 4.18 160 160 103 93 197.09
192 4.35 170 150 104 93 266.56
171.75 4.16 150 150 101 87 260.56
162 4.43 170 180 108 106 219.25
104.93 3.2 110 110 86 92 132.68
105.67 3.02 120 110 90 86 130.24
117.59 3.64 130 120 85 80 205.88
140.25 3.68 120 140 89 83 153.92
150.17 3.6 140 130 92 94 154.64
165.17 3.85 160 150 95 95 240.57
```

It makes sense to regard the first two variables (`Distance` and `Hang`) as responses and the remaining five variables as predictors. Let's build a regression model to predict `Distance` from the last five variables.

First, let's read the data into R and plot the response variable against each of the five predictors.

```
> punt <- read.table("punting.txt", header=T)

> par(mfrow=c(3,2)) # put six plots on a page, in 3 rows and 2 columns
> plot(punt$R_Strength, punt$Distance)
> plot(punt$L_Strength, punt$Distance)
> plot(punt$R_Flexibility, punt$Distance)
> plot(punt$L_Flexibility, punt$Distance)
> plot(punt$O_Strength, punt$Distance)
```



Now let's regress **Distance** on the five predictors.

```
> result <- lm( Distance ~ R_Strength + L_Strength +
+   R_Flexibility + L_Flexibility + O_Strength, data=punt)
> summary( result )

Call:
lm(formula = Distance ~ R_Strength + L_Strength + R_Flexibility +
```

```
L_Flexibility + O_Strength, data = punt)

Residuals:
    Min      1Q  Median      3Q     Max 
-17.3829 -9.5711 -0.2166  5.4988 20.0188 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -29.58047   65.70042 -0.450   0.666    
R_Strength    0.27877    0.45638   0.611   0.561    
L_Strength    0.06971    0.48388   0.144   0.890    
R_Flexibility 1.24146    1.44927   0.857   0.420    
L_Flexibility -0.39535    0.74472  -0.531   0.612    
O_Strength     0.22369    0.13053   1.714   0.130    

Residual standard error: 14.65 on 7 degrees of freedom
Multiple R-Squared:  0.8144,    Adjusted R-squared:  0.6818 
F-statistic: 6.142 on 5 and 7 DF,  p-value: 0.01694
```

Notice that none of the five slopes are significantly different from zero. However, the omnibus F-statistic is highly significant, and the overall R^2 is high. This suggests that the predictors may be “sharing significance” because they are correlated with each other. Let’s look at the pairwise correlations among the predictors.

```
> # form matrix of predictors
> x <- cbind( punt$R_Strength, punt$L_Strength, punt$R_Flexibility,
+   punt$L_Flexibility, punt$O_Strength)

> cor(x)
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.0000000 0.8957224 0.7746767 0.3569478 0.6065448
[2,] 0.8957224 1.0000000 0.8140684 0.4232394 0.5230756
[3,] 0.7746767 0.8140684 1.0000000 0.6895403 0.6902845
[4,] 0.3569478 0.4232394 0.6895403 1.0000000 0.4081231
[5,] 0.6065448 0.5230756 0.6902845 0.4081231 1.0000000
```

To diagnose the extent of multicollinearity, let's regress each of the predictors on the others and compute R_j^2 and VIF_j . To do this efficiently, let's write a loop that repeatedly calls `lsfit`.

```

> # compute R^2 and VIF's
> R2.table <- matrix( NA, 5, 2 ) # to hold the results
> dimnames(R2.table) <- list(
+   c("R_Strength", "L_Strength", "R_Flexibility", "L_Flexibility",
+     "O_Strength"),
+   c("R2", "VIF") )

> n <- nrow(x)

> for( j in 1:5 ){
+   tmp <- lsfit( x[,-j], x[,j] )
+   SSTot <- (n-1) * var( x[,j] )
+   SSErr <- sum( tmp$res^2 )
+   R2 <- 1 - SSErr / SSTot
+   VIF <- 1 / (1-R2)
+   R2.table[j,1] <- R2
+   R2.table[j,2] <- VIF}

> R2.table
      R2      VIF
R_Strength 0.8346251 6.046867
L_Strength 0.8546821 6.881465
R_Flexibility 0.8597918 7.132251
L_Flexibility 0.5614097 2.280032
O_Strength 0.5406464 2.176972

```

Multicollinearity is clearly playing a role, but none of the R_j^2 's or VIF_j 's are extremely high. What should we do? If our goal is to build a model that predicts Distance as precisely as possible, we don't want to omit important predictors, because doing so could bias the predictions. That is, we don't want to omit a predictor whose true

coefficient is far from zero. At the same time, we don't want to include predictors that are unnecessary, because doing so will make the model needlessly complicated, and estimating the extra parameters will make the predictions less precise. In selecting a good model, we have to consider the bias-variance tradeoff. Criteria for model selection will be discussed at length in a future lecture.

Now let's look at some hypotheses that we cannot test by simply looking at the table of coefficients. First, let's see if the two predictors `R_Strength` and `L_Strength` can be removed from the model. If we label the coefficients as β_0, \dots, β_5 , then we can discard `R_Strength` and `L_Strength` if $\beta_1 = \beta_2 = 0$. We can compute the F statistic for this test in two different ways. First, let's express this as the general linear hypothesis $H_0 : A\beta = c$, with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad c = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The test statistic is

$$F = \frac{SS/q}{S^2} \sim F_{q,n-p},$$

where

$$SS = (A\hat{\beta} - c)^T \left[A(X^T X)^{-1} A^T \right]^{-1} (A\hat{\beta} - c)$$

and $q = 2$. If you apply the `summary` function to the result of `lm`, and save the results from `summary`, the matrix $(X^T X)^{-1}$ is included as `cov.unscaled`.

```

> # test H0: beta.1 = beta.2 = 0
> a1 <- c(0,1,0,0,0,0)
> a2 <- c(0,0,1,0,0,0)
> A <- rbind(a1,a2)
> c <- c(0,0)

> betahat <- result$coef
> s2 <- sum(result$res^2) / result$df.res
> tmp <- summary(result)
> xtxinv <- tmp$cov.unscaled

> SS <- t(A %*% betahat - c) %*% solve(A %*% xtxinv %*% t(A)) %*% (A %*% betahat - c)
> SS
[1,] 225.9443

> q <- 2
> F <- (SS / q) / s2
> F
[1,] 0.5263889

> p <- 1 - pf(F, q, result$df.res)
> p
[1,] 0.6123971

```

We cannot reject H_0 . Another way to compute the F-statistic is to re-run `lm` with these two predictors entered last, print the ANOVA table, and obtain the SS due to these two predictors given all the others.

```

> result2 <- lm(Distance ~ R_Flexibility + L_Flexibility + O_Strength
+     + R_Strength + L_Strength, data=punt)

> anova(result2)
Analysis of Variance Table

Response: Distance

```

```

      Df Sum Sq Mean Sq F value    Pr(>F)
R_Flexibility  1 5262.1 5262.1 24.5184 0.001653 ***
L_Flexibility  1  339.1   339.1  1.5802 0.249047
O_Strength     1  763.8   763.8  3.5591 0.101188
R_Strength     1  221.5   221.5  1.0320 0.343503
L_Strength     1     4.5     4.5  0.0208 0.889513
Residuals      7 1502.3   214.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The SS due to `R_Strength` and `L_Strength` given the other three predictors is

$$SS = 221.5 + 4.5 = 226.0,$$

which agrees with the previous method, and the F-statistic is

$$F = \frac{226.0/2}{214.6} = 0.526561,$$

which agrees with the previous answer except for rounding error.

Leverage. Now we are going to begin discussion of diagnostic quantities that help us to understand how well the individual observations are described by the model, and how the individual observations influence the model fit. This material, which we will cover in the remainder of this lecture and the next one, is addressed in Chapter 10 of KNNL.

The hat matrix, which is defined as

$$H = X(X^T X)^{-1} X^T,$$

is the $n \times n$ matrix that projects y (or any other n -dimensional vector) onto $\mathcal{R}(X)$. This matrix can be very large, and it is rarely stored or computed. But the diagonal elements of this matrix, which are usually denoted by h_i , $i = 1, \dots, n$, play an important role in model diagnosis. These values are called **leverages**.

It is easy to see that

$$h_i = x_i^T (X^T X)^{-1} x_i,$$

where x_i is the i th row of X expressed as a column vector. So we can compute the leverages without forming the entire H matrix.

What does h_i measure? Points that lie far from the center of the covariate space have higher values of h_i . For the simple linear regression model (Lectures 10-11), we showed that

$$(X^T X)^{-1} = \begin{bmatrix} 1/n + \bar{x}^2/S_{xx} & -\bar{x}/S_{xx} \\ -\bar{x}/S_{xx} & 1/S_{xx} \end{bmatrix}$$

In this case, the i th row of the design matrix is $(1, x_i)$, so

$$\begin{aligned} h_i &= \begin{bmatrix} 1 & x_i \end{bmatrix} \begin{bmatrix} 1/n + \bar{x}^2/S_{xx} & -\bar{x}/S_{xx} \\ -\bar{x}/S_{xx} & 1/S_{xx} \end{bmatrix} \begin{bmatrix} 1 \\ x_i \end{bmatrix} \\ &= \frac{1}{n} + \frac{(x_i - \bar{x})^2}{S_{xx}}. \end{aligned}$$

The minimum value of h_i is achieved when $x_i = \bar{x}$, and it

grows as x_i moves away from \bar{x} . So a large value of h_i indicates that subject i 's covariates are unusual relative to the others.

With multiple predictors, we need to take into account the relationships among the columns of X when judging how far an observation lies from the center. Recall the factorization

$$X = QR,$$

where Q is orthonormal and R is upper-triangular.

Because the columns of Q span the same space as X , we can also write the hat matrix as

$$H = Q(Q^T Q)^{-1} Q^T = QQ^T,$$

and thus

$$h_i = \sum_{j=1}^p q_{ij}^2 = q_i^T q_i = \|q_i\|^2,$$

where q_{ij} is the (i, j) th element of Q , and q_i the i th row of Q expressed as a column vector. If the first column of X is constant, then the first column of Q is also constant, and q_{i1}^2 is the same for every subject. The j th column of Q is a scale-free measure of how far x_{ij} lies from $\{x_{i'j}, i' \neq i\}$ in the unique direction contributed by column j that is not explained by the previous columns. So we can regard h_i as something like a scale-free measure of “distance” between

x_i , the vector of predictors for subject i , and

$$\bar{x} = \frac{1}{n} \sum_{i=1}^N x_i,$$

the average value of x_i in the dataset. (Technically speaking, it is not a distance measure because $h_i \neq 0$ when $x_i = \bar{x}$. But it is minimized when $x_i = \bar{x}$.)

Using the fact that

$$\text{tr } H = \text{rank } H = p$$

we see that the average value of h_i in any dataset is p/n . This average, p/n , is a useful benchmark for judging how far an observation lies from the center of the covariate space, relative to the other observations. KNNL says (p. 399) that observations for which h_i exceed $2p/n$ should be considered “high leverage.” Other books say $3p/n$, or even $5p/n$, is a useful cutoff for identifying points with high leverage. All of these rules are arbitrary. Once we identify some observations as having high leverage, it’s not clear what you should do with them.

Leverage and prediction error. In Lectures 11-12, as we discussed the simple linear regression model, we constructed a confidence interval for the mean response, and a prediction intervals for a future response, at a given value of the predictor. The extension to multiple linear regression is straightforward. The multiple linear

regression model says that

$$y_i \sim N(x_i^T \beta, \sigma^2),$$

where x_i is the $p \times 1$ vector of predictors (including a constant, if present) for subject i . Let x_* denote fixed values of the predictors at which we want to predict the response. Under our model, the mean value of the response at x_* is $x_*^T \beta$, the natural estimate of this mean is

$$\hat{y}(x_*) = x_*^T \hat{\beta},$$

and the variance of this estimate is

$$\begin{aligned} V(\hat{y}(x_*)) &= \sigma^2 x_*^T (X^T X)^{-1} x_* \\ &= \sigma^2 h_*, \end{aligned}$$

where

$$h_* = x_*^T (X^T X)^{-1} x_*$$

is the leverage of x_* relative to the sample observations x_1, \dots, x_n . (If x_* happens to be equal to x_i , one of the predictor vectors in the sample, then h_* will equal h_i .) Replacing σ^2 by its unbiased estimate gives us the standard error for the mean response,

$$SE(\hat{y}(x_*)) = S \sqrt{h_*}.$$

A 95% confidence interval for the mean response is

$$\hat{y}(x_*) \pm t_{.975, n-p} SE(\hat{y}(x_*)).$$

If y_* denotes an unseen future response at x_* , a 95%

prediction interval for x_* is

$$\hat{y}(x_*) \pm t_{.975,n-p} S \sqrt{1 + h_*}.$$

Notice that these are pointwise intervals with 95% coverage probability at a single value x_* . A confidence band that encloses the mean response at x_* simultaneously at all possible values of x_* is

$$\hat{y}(x_*) \pm \sqrt{p F_{.95,p,n-p}} SE(\hat{y}(x_*)),$$

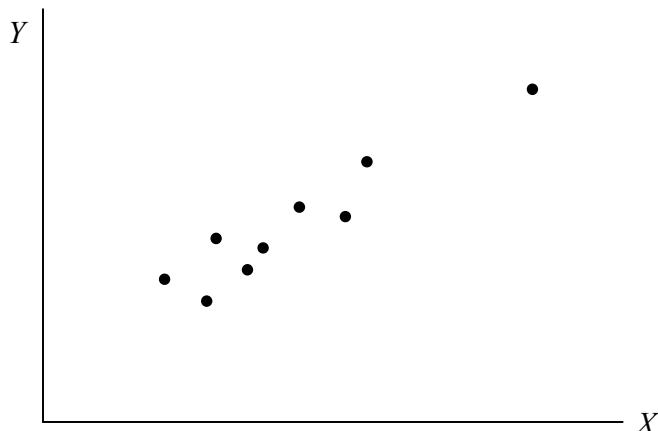
which was given in Lecture 12 as the Working-Hotelling procedure.

Here we see that the variance of estimation and prediction goes up as the leverage h_* increases. Regression-based estimation and prediction is most precise at the center of the covariate space, and becomes less precise in the outlying regions.

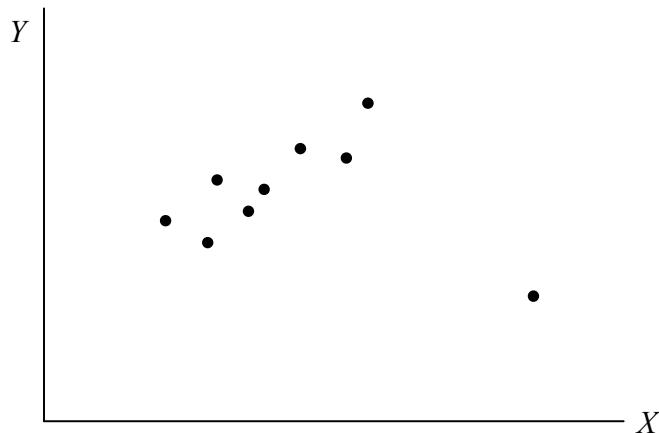
When computing predictions, it is helpful to compare h_* to the distribution of leverages h_1, \dots, h_n in the actual data. If h_* is large relative to h_1, \dots, h_n , it is a sign that we are extrapolating to regions of the covariate space beyond the observed data, which is unwise. Predictions in these outlying regions will have large variance even if the model is true. But the greatest danger is that the model may not be true, and we have no basis to evaluate whether the regression model fits in these outlying regions.

Leverage versus influence. Leverage is related to influence, but the two concepts are not the same. In regression analysis, influence refers to how much an observation affects or changes the fit of the regression model. An observation is considered influential if deleting it from the data set produces a large change in $\hat{\beta}$ and \hat{y} .

Observations with high leverage are potentially but not necessarily influential. To see this, consider the rightmost point in the scatterplot below.

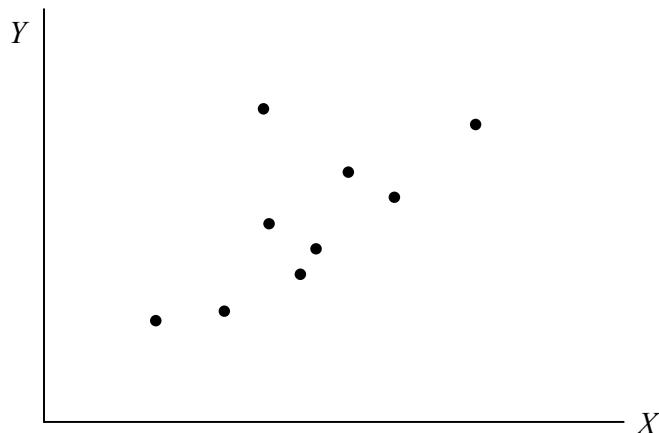


This observation has high leverage, but it is not influential. The estimated regression line is essentially the same whether or not this point is included. Now consider the rightmost point in this next plot.



This point has exactly the same leverage as the point in the previous plot, but it is extremely influential. Including this point will change the estimated slope from positive to negative. Observations with high leverage may be influential.

On the other hand, if a point has small leverage, it cannot exert much influence. Consider the topmost observation in this next plot.



Including that point will raise the estimated regression line by a small amount, slightly increasing the intercept but barely changing the slope. An observation with small leverage is not influential.

Observations with high leverage h_i are potentially influential, but they are not necessarily influential.

Leverage and residuals. Leverage affects the properties of residuals. The residual for observation i is the difference between the observation and the fitted value,

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - x_i^T \hat{\beta}.$$

Recall that the vector of residuals is

$$\hat{\epsilon} = (\hat{\epsilon}_1, \dots, \hat{\epsilon}_n)^T = (I - H)y.$$

Because $y \sim N(X\beta, \sigma^2 I)$, it follows that $\hat{\epsilon}$ is multivariate normal with mean

$$E(\hat{\epsilon}) = (I - H)X\hat{\beta} = 0$$

(because $X\hat{\beta}$ lies in $\mathcal{R}(X)$) and covariance matrix

$$V(\hat{\epsilon}) = \sigma^2(I - H)$$

(because $(I - H)$ is symmetric and idempotent). The variance of a single residual is

$$V(\hat{\epsilon}_i) = \sigma^2(1 - h_i).$$

Therefore, **observations with high leverage tend to have small residuals.**

This makes sense. An observation with high leverage will strongly pull the estimated regression line or plane toward itself, causing the residual to be small. On the other hand, an observation near the center of the covariate space exerts little influence over the regression function, so its residual tends to be larger.

The leave-one-out formula. Measures of influence address the question, “How much would the answer change if observation i were deleted?” To compute these measures, we could imagine refitting the regression model n times, leaving out each of the observations $i = 1, \dots, n$ one at a time. If n were large, this would be very tedious. Fortunately, we don’t have to do it this way. It is not difficult to derive expressions for how the least-squares results change if one observation is deleted.

Let $X_{(i)}$ denote the $(n - 1) \times p$ design matrix that we would get if we left out observation i , and let $y_{(i)}$ denote the corresponding $(n - 1) \times 1$ vector of responses. The least-squares estimate with observation i left out is

$$\hat{\beta}_{(i)} = (X_{(i)}^T X_{(i)})^{-1} X_{(i)}^T y_{(i)}.$$

But

$$X^T y = \sum_{i=1}^n x_i y_i \quad \text{and} \quad X^T X = \sum_{i=1}^n x_i x_i^T,$$

so

$$\hat{\beta}_{(i)} = \left(X^T X - x_i x_i^T \right)^{-1} \left(X^T y - x_i y_i \right).$$

A very useful matrix identity for statisticians is

$$(A + BCB^T)^{-1} = A^{-1} - A^{-1}B(C^{-1} + B^T A^{-1}B)^{-1}B^T A^T,$$

assuming everything is conformable and the necessary inverses exist. By taking $A = X^T X$, $B = x_i$ and $C = -1$, we get

$$\begin{aligned} (X_{(i)}^T X_{(i)})^{-1} &= (X^T X)^{-1} - (X^T X)^{-1} \\ &\quad \times x_i (-1 + x_i^T (X^T X)^{-1} x_i)^{-1} x_i^T (X^T X)^{-1} \\ &= (X^T X)^{-1} + (X^T X)^{-1} \\ &\quad \times x_i (1 - h_i)^{-1} x_i^T (X^T X)^{-1} \end{aligned}$$

and

$$\begin{aligned} \hat{\beta}_{(i)} &= \left[(X^T X)^{-1} + (X^T X)^{-1} \right. \\ &\quad \left. \times x_i (1 - h_i)^{-1} x_i^T (X^T X)^{-1} \right] \left[X^T y - x_i y_i \right] \\ &= \hat{\beta} - (X^T X)^{-1} x_i y_i \\ &\quad + (X^T X)^{-1} x_i (1 - h_i)^{-1} x_i^T \hat{\beta} \\ &\quad - (X^T X)^{-1} x_i (1 - h_i)^{-1} h_i y_i \\ &= \hat{\beta} - (1 - h_i)^{-1} (X^T X)^{-1} x_i y_i \\ &\quad + (1 - h_i)^{-1} (X^T X)^{-1} x_i \hat{y}_i \\ &= \hat{\beta} - (1 - h_i)^{-1} (X^T X)^{-1} x_i \hat{\epsilon}_i. \end{aligned}$$

We will call this the “leave-one-out” formula. It shows how the estimated coefficients change when a single observation is left out.

An even nicer formula arises when we look at how leaving an observation out affects the error of prediction **for that observation**. Let

$$\hat{\epsilon}_{(i)} = y_i - x_i^T \hat{\beta}_{(i)}$$

denote the error for predicting y_i based on all the other observations. This quantity is called the “PRESS residual,” where PRESS is an acronym for “PREdiction Sum of Squares.” Using the leave-one-out formula, we get

$$\begin{aligned}\hat{\epsilon}_{(i)} &= y_i - x_i^T \left[\hat{\beta} - \left(\frac{1}{1 - h_i} \right) (X^T X)^{-1} x_i \hat{\epsilon}_i \right] \\ &= y_i - \hat{y}_i + \left(\frac{h_i}{1 - h_i} \right) \hat{\epsilon}_i \\ &= \left(\frac{1 - h_i}{1 - h_i} \right) \hat{\epsilon}_i + \left(\frac{h_i}{1 - h_i} \right) \hat{\epsilon}_i \\ &= \left(\frac{1}{1 - h_i} \right) \hat{\epsilon}_i.\end{aligned}$$

An observation with a large PRESS residual is poorly predicted by the other observations. The difference between the ordinary residuals and the PRESS residuals vanishes as n becomes large, because the impact of any one observation on the regression fit becomes negligible.

ANALYSIS OF RESIDUALS, PART I

Last time, we discussed the role of the leverages h_1, \dots, h_n , which are the diagonal elements of the hat matrix H .

Before going father, let's examine them in an example.

Example: Punting data. Recall the punting data that we began to analyze in the last lecture.

```
> punt <- read.table("punting.txt", header=T)
> punt
  Distance Hang R_Strength L_Strength R_Flexibility L_Flexibility O_Strength
  1   162.50 4.75      170       170        106        106     240.57
  2   144.00 4.07      140       130        92         93     195.49
  3   147.50 4.04      180       170        93         78     152.99
  4   163.50 4.18      160       160        103        93     197.09
  5   192.00 4.35      170       150        104        93     266.56
  6   171.75 4.16      150       150        101        87     260.56
  7   162.00 4.43      170       180        108        106     219.25
  8   104.93 3.20      110       110        86         92     132.68
  9   105.67 3.02      120       110        90         86     130.24
 10   117.59 3.64      130       120        85         80     205.88
 11   140.25 3.68      120       140        89         83     153.92
 12   150.17 3.60      140       130        92         94     154.64
 13   165.17 3.85      160       150        95         95     240.57
```

We regressed `Distance` on the last five variables and found that, although the five predictors were jointly significant, none was significant individually. Now let's find the leverage values. One way is to form the X matrix and

compute $h_i = x_i^T (X^T X)^{-1} x_i$ for $i = 1, \dots, n$.

```
> x <- cbind( 1, punt$R_Strength, punt$L_Strength, punt$R_Flexibility,
+             punt$L_Flexibility, punt$O_Strength)
> xtxinv <- solve( t(x) %*% x )

> n <- nrow(punt)
> lev <- rep(NA,n)
> for(i in 1:n){
+   lev[i] <- t( x[i,] ) %*% xtxinv %*% x[i,] }
> lev
[1] 0.3532719 0.1849579 0.9114954 0.2779728 0.5005858 0.5277087 0.4792926
[8] 0.3966875 0.5518719 0.4548661 0.6718640 0.2670713 0.4223540
```

Another way is to compute them is $h_i = \sum_{j=1}^p q_{ij}^2$, where the q_{ij} 's are elements of the orthonormal basis matrix Q .

```
> lev <- apply( qr.Q( result$qr )^2, 1, sum)
> lev
[1] 0.3532719 0.1849579 0.9114954 0.2779728 0.5005858 0.5277087 0.4792926
[8] 0.3966875 0.5518719 0.4548661 0.6718640 0.2670713 0.4223540
```

A third way is to let R do it. The function `lm.influence`, when applied to the result of `lm`, produces a list of diagnostics related to influence. The first component of the list, called `hat`, contains the leverage values.

```
> tmp <- lm.influence(result)
> lev <- tmp$hat
> lev
      1         2         3         4         5         6         7         8
0.3532719 0.1849579 0.9114954 0.2779728 0.5005858 0.5277087 0.4792926 0.3966875
      9        10        11        12        13
0.5518719 0.4548661 0.6718640 0.2670713 0.4223540
```

Note that the average value of the leverages is $p/n = 6/13 = .461$.

```
> mean(lev)
[1] 0.4615385
```

Observation #3 has a leverage value nearly twice the average, so by the rule of KNNL we might want to regard it as potentially influential. If we look at the dataset, we immediately see why this observation has high leverage. Last time, we examined the pairwise correlations among the predictors and found that `R_Strength` and `L_Strength` were positively correlated with `R_Flexibility` and with `L_Flexibility` and with `O_Strength`. Subject #3 has high values for `R_Strength` and `L_Strength`, low values for `R_Flexibility` and `L_Flexibility` and a moderate value for `O_Strength`. He has an unusual combination of covariates, even though his individual values for the covariates are not so unusual.

Subject #3 has relatively high leverage, but is he influential? Last time, we showed that if observation i is removed, the least-squares estimates become

$$\hat{\beta}_{(i)} = \hat{\beta} - (1 - h_i)^{-1}(X^T X)^{-1}x_i\hat{\epsilon}_i,$$

so the change in coefficients when observation i is added to the dataset is

$$\hat{\beta} - \hat{\beta}_{(i)} = (1 - h_i)^{-1}(X^T X)^{-1}x_i\hat{\epsilon}_i.$$

Let's compute these changes using our formula.

```

> # create a blank matrix to hold the changes
> p <- 6
> changes <- matrix( NA, n, p )
> dimnames(changes) <- list( NULL, names(result$coef) )

> res <- result$residuals
> for(i in 1:n){
+   changes[i,] <- xtxinv %*% x[i,] * res[i] / (1-lev[i]) }
>
> changes
      (Intercept) R_Strength L_Strength R_Flexibility L_Flexibility
[1,] 19.43460323 0.001453205 -0.060176208 0.17421768 -0.29572266
[2,] 2.70252974 0.023487130 -0.020071085 -0.09946865 0.06265667
[3,] -46.81690170 -0.543782495 -0.110374560 0.63074622 0.56951433
[4,] -4.15863603 -0.005227749 -0.005181880 0.11154164 -0.03744356
[5,] -29.19972214 0.216783591 -0.356852944 0.73249184 -0.26957427
[6,] 0.02585984 0.004828973 -0.001062123 -0.01376778 0.01072945
[7,] 29.04670446 0.115597769 -0.177956499 -0.05263272 -0.22414131
[8,] -3.84585245 0.020794782 0.004418028 0.06464430 -0.08652992
[9,] 23.36615020 -0.103257835 0.431095117 -1.31939191 0.34824754
[10,] -78.10234304 0.013449830 -0.050518358 0.99829967 0.10312152
[11,] 73.22537160 -0.923086850 0.913464292 -0.32120488 -0.45882281
[12,] -5.95363204 0.221443335 -0.200058712 -0.15322939 0.34135494
[13,] 11.39358734 0.048596756 0.032450923 -0.47453742 0.17897862
      O_Strength
[1,] -0.009956499
[2,] 0.004716119
[3,] 0.139657626
[4,] -0.006901812
[5,] 0.024598538
[6,] -0.001442168
[7,] 0.018094695
[8,] 0.006885643
[9,] 0.113013517
[10,] -0.122370314
[11,] 0.039284270
[12,] -0.062961809
[13,] 0.033511718

```

Actually, we didn't have to do it this way. The function `lm.influence` also produces the matrix of changes; it's

called **coefficients**.

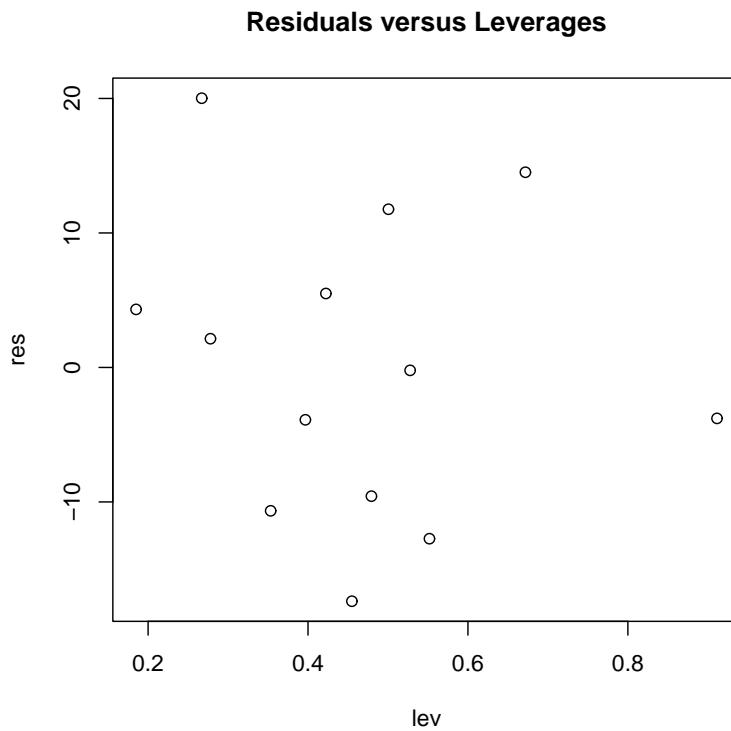
```
> tmp <- lm.influence( result )
> tmp$coefficients
  (Intercept) R_Strength L_Strength R_Flexibility L_Flexibility
1 19.43460323 0.001453205 -0.060176208 0.17421768 -0.29572266
2 2.70252974 0.023487130 -0.020071085 -0.09946865 0.06265667
3 -46.81690170 -0.543782495 -0.110374560 0.63074622 0.56951433
4 -4.15863603 -0.005227749 -0.005181880 0.11154164 -0.03744356
5 -29.19972214 0.216783591 -0.356852944 0.73249184 -0.26957427
6 0.02585984 0.004828973 -0.001062123 -0.01376778 0.01072945
7 29.04670446 0.115597769 -0.177956499 -0.05263272 -0.22414131
8 -3.84585245 0.020794782 0.004418028 0.06464430 -0.08652992
9 23.36615020 -0.103257835 0.431095117 -1.31939191 0.34824754
10 -78.10234304 0.013449830 -0.050518358 0.99829967 0.10312152
11 73.22537160 -0.923086850 0.913464292 -0.32120488 -0.45882281
12 -5.95363204 0.221443335 -0.200058712 -0.15322939 0.34135494
13 11.39358734 0.048596756 0.032450923 -0.47453742 0.17897862
  O_Strength
1 -0.009956499
2 0.004716119
3 0.139657626
4 -0.006901812
5 0.024598538
6 -0.001442168
7 0.018094695
8 0.006885643
9 0.113013517
10 -0.122370314
11 0.039284270
12 -0.062961809
13 0.033511718
```

Examining these changes, we see that Subject #3 accounts for the largest change in the coefficient of **L_Flexibility** (0.5695) and the largest change in the coefficient for **O_Strength** (0.1397).

When n is large, it becomes inconvenient to print out the leverages and examine them directly. Later we will

describe how to summarize these effects more efficiently using other diagnostic measures and plots. For now, let's plot the residuals versus the leverage values.

```
> plot(lev, res, main="Residuals versus Leverages")
```

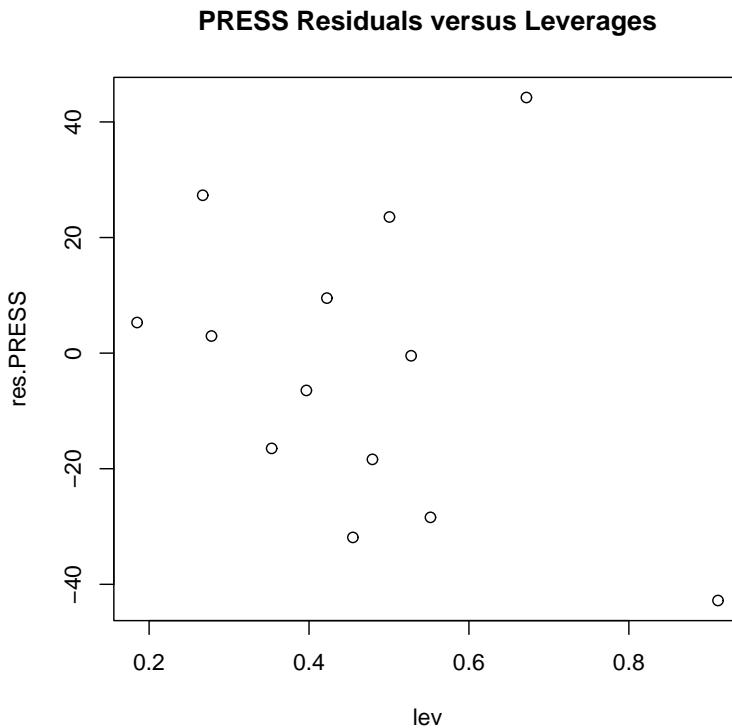


Subject #3 has the highest leverage but a small residual. This agrees with what we discovered last time: an observation with high leverage tends to pull the estimated regression function toward itself, making the residual small. Now let's plot the PRESS residuals

$$\hat{\epsilon}_{(i)} = \frac{\hat{\epsilon}_i}{(1 - h_i)}$$

versus the leverages.

```
> res.PRESS <- res/(1-lev)
> plot(lev, res.PRESS, main="PRESS Residuals versus Leverages")
```



Recall that the PRESS residual measures the error in predicting each observation from all the other observations. Comparing this plot to the previous one, we see that the PRESS residuals are substantially larger than the ordinary residuals. This is because n is small. (As n grows, the differences between the two types of residuals will vanish.) In particular, the PRESS residual for Subject #3 has increased dramatically, showing that this point is not predicted very well by the others. Including this point

shifts the regression plane quite a lot.

What to do about points with high leverage. In one sense, points with high leverage are a good thing, because they help us to estimate the slopes more precisely. (Recall the discussion in Lecture 11 about how, in simple linear regression, the variance of $\hat{\beta}_1$ is inversely proportional to S_{xx} .) High-leverage observations contribute the most information about the regression slopes and are thus very helpful **if the model is true**, i.e. if

$$E(y_i) = x_i^T \beta \tag{1}$$

accurately describes the mean structure over the relevant region of the covariate space. With real data, however, the regression relationships specified by (1) are only an approximation to the true mean structure. When the model is not exactly true, high-leverage observations may distort the regression fit and make the predictions for the other observations worse. In general, it's not a good idea to trust any analysis whose results are largely determined by just one observation or a small group of observations.

When a few points have leverage values much higher than the rest, it's often because the distributions of one or more predictors are highly skewed. If we transform the skewed predictors to make them more nearly normal, then the high-leverage points will be pulled in toward the center of the predictor space, and the problem may be solved.

Technically speaking, the linear regression model does not assume anything about the distribution of the predictors. But if some predictors are highly skewed, the fit of the model may be unduly influenced by a small number of observations, which is not a good idea.

A note on terminology. In previous lectures, we have called

$$\epsilon_i = y_i - x_i^T \beta$$

the “true residual” and

$$\hat{\epsilon}_i = y_i - x_i^T \hat{\beta}$$

the “estimated residual.” The latter is observable, but the former is not. From now on, we will use the term “residual” to refer to $\hat{\epsilon}_i$. We may also call $\hat{\epsilon}_i$ a “raw residual” to distinguish it from the PRESS residual and other kinds of adjusted residuals. If we need to speak of ϵ_i , we will now call it the “true error.”

Checking the normality of residuals. The vector of true errors, which we cannot see, is distributed as

$$\epsilon \sim N(0, \sigma^2 I),$$

so the ϵ_i ’s are independent and identically distributed as $N(0, \sigma^2)$. In contrast, the residuals that we do see are distributed as

$$\hat{\epsilon} \sim N(0, \sigma^2(I - H)).$$

Therefore, they are

- not identically distributed (because they have different variances) and
- not independent (because the off-diagonal elements of $I - H$ are not zero).

The joint distribution of $\hat{\epsilon}_1, \dots, \hat{\epsilon}_n$ is unusual because the $n \times n$ matrix $I - H$ is singular, with

$$\text{rank}(I - H) = n - p.$$

The vector $\hat{\epsilon}$ is not free to lie anywhere in the n -dimensional space. Rather, it is restricted to lie within the $(n - p)$ -dimensional subspace orthogonal to $\mathcal{R}(X)$. If X_j denotes a column of X , then

$$X_j^T \hat{\epsilon} = \sum_{i=1}^n x_{ij} \hat{\epsilon}_i = 0.$$

If the design matrix contains a column of ones, then the residuals must sum to zero,

$$1^T \hat{\epsilon} = \sum_{i=1}^n \hat{\epsilon}_i = 0. \quad (2)$$

More generally, (2) holds if some linear combination of the columns of X is constant, i.e. if $1 \in \mathcal{R}(X)$. Moreover, if $1 \in \mathcal{R}(X)$, then $\hat{\epsilon}$ is also *uncorrelated* with every predictor in the model, and with every linear combination of predictors.

If n is large relative to p , which we will sometimes write as $n \gg p$, then $\hat{\beta} \approx \beta$ and $\hat{\epsilon}_i \approx \epsilon_i$. In that case, the off-diagonal elements of $I - H$ will be small, and the $\hat{\epsilon}_i$'s should be approximately $N(0, \sigma^2)$ if the model is true. So if $n \gg p$, we can check the assumption of normality by creating a normal probability plot of the raw residuals.

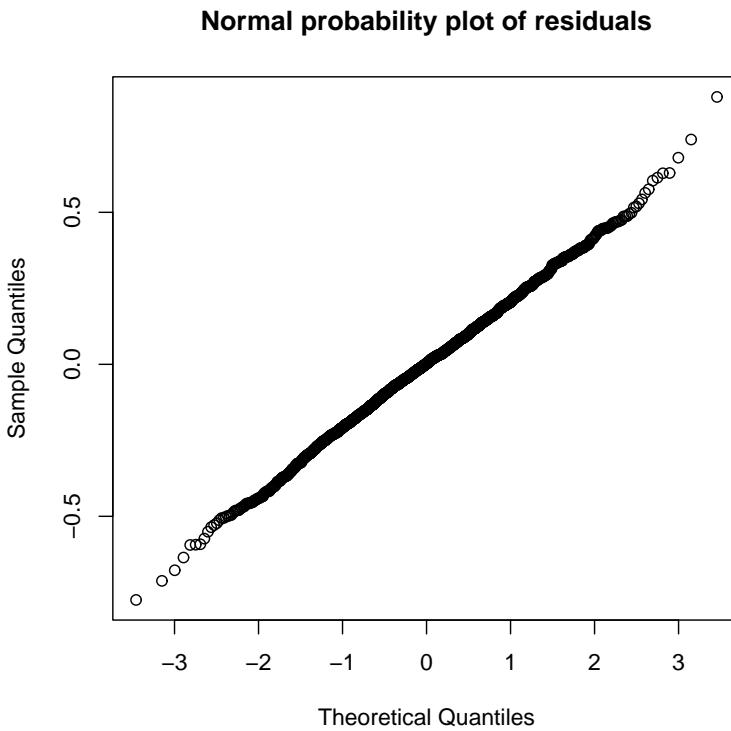
For example, here is a normal probability plot of the $n = 1835$ residuals from the regression of $\log(\text{CHOL})$ on $\log(\text{BMI})$ and MORPH in our body measurements dataset.

```
> body <- read.table("body.dat", header=T)

> meters <- body$height * 2.54 / 100 # height in meters
> kg <- body$weight * 0.45359237      # weight in pounds
> body$log.bmi <- log(kg/meters^2)
> body$log.chol <- log( body$chol )
> body$morph <- body$waist / body$hips

> result <- lm( log.chol ~ log.bmi + morph, data=body)

> qqnorm( result$res, main="Normal probability plot of residuals")
```



The plot is remarkably straight, meaning that the residuals are approximately normal. If we applied the Wilk-Shapiro test to these residuals, we would not reject the hypothesis of normality ($p = .36$). Technically speaking, the Wilk-Shapiro test would be valid if applied to the true errors $\epsilon_1, \dots, \epsilon_n$, not to the observed residuals $\hat{\epsilon}_1, \dots, \hat{\epsilon}_n$. But since $n \gg p$, we can suppose that $\hat{\epsilon}_i \approx \epsilon_i$ and apply the test to the $\hat{\epsilon}_i$'s.

What happens if the residuals are not normal? If the residuals are not normally distributed **but still have constant variance**, then a generalization of the Central Limit Theorem ensures that

$$\hat{\beta} \sim N(\beta, \sigma^2(X^T X)^{-1})$$

will be approximately true if $n \gg p$. In that case, all of the inferences about the coefficients that we have discussed—including t-tests, F-tests and tests of general linear hypotheses—will be reasonably accurate.

Confidence intervals for the mean response at given values of the predictors will be accurate. But prediction intervals may not be accurate, because those intervals pertain to a single future observation.

But non-normality in the residuals—especially skewness—is often accompanied by non-constant variance.

Standardizing the residuals. Because the variance of the raw residual is $V(\hat{\epsilon}_i) = \sigma^2(1 - h_i)$, it is common practice to “standardize” the residual by dividing it by its estimated standard deviation. The standardized residual, defined as

$$\tilde{\epsilon}_i = \frac{\hat{\epsilon}_i}{S\sqrt{1 - h_i}},$$

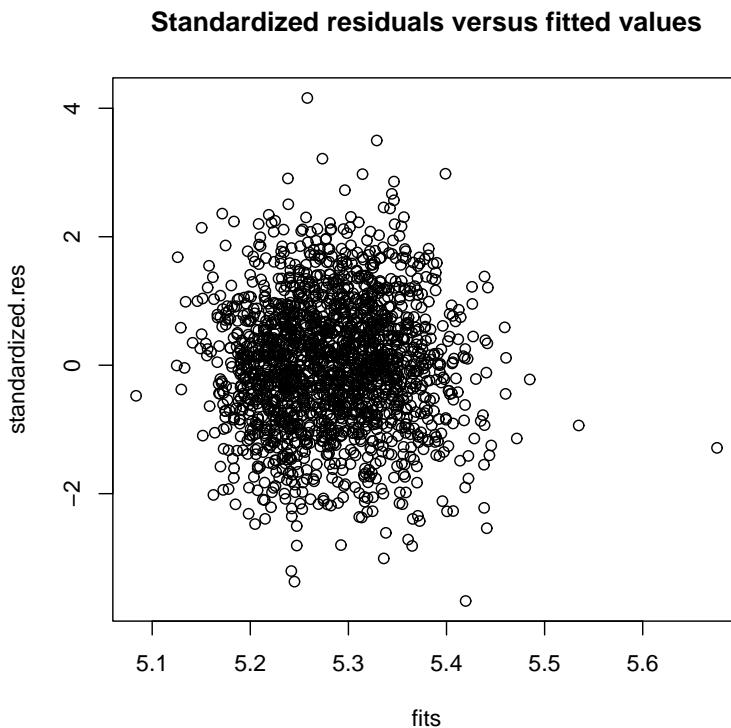
will have constant variance if the model is true. At first glance, one might think that the standardized residual should be distributed as t_{n-p} , because the unknown σ^2 in the denominator has been replaced by S^2 . But it is not exactly t_{n-p} , because the numerator and the denominator are not independent; the squared residual for observation i appears in SS_{Err} and thus affects S^2 . (We could make the numerator and denominator independent by eliminating observation i from the estimate of σ^2 ; more about that next time.) Even though the standardized residuals are

not exactly t_{n-p} , analysts often think of them as being approximately t_{n-p} or, if $n \gg p$, as approximately $N(0, 1)$. Standardized residuals that are unusually large relative to $N(0, 1)$ (e.g. $|\tilde{\epsilon}_i| > 3$ or $|\tilde{\epsilon}_i| > 4$) may be considered “outliers” in the sense that these y_i ’s are much farther away from their \hat{y}_i ’s than they should be under the normal linear regression model. Outliers indicate that the regression relationship $E(y_i) = x_i^T \beta$ does not hold in the regions of the predictor space where these observations are found, or that the true errors are not $N(0, \sigma^2)$, or both.

When $n \gg p$, the leverages are all close to zero, and the $\tilde{\epsilon}_i$ ’s are essentially the $\hat{\epsilon}_i$ ’s divided by S . In large-sample situations, plots based on the $\tilde{\epsilon}_i$ ’s will look essentially the same as plots based on the $\hat{\epsilon}_i$ ’s, except that the scale has changed. In smaller samples, the plots of the two types of residuals may look different, and it is probably better to use the $\tilde{\epsilon}_i$ ’s, especially when looking for heteroscedasticity.

Looking for heteroscedasticity. The most common way to look for heteroscedasticity is to plot the standardized residuals against the fitted values $\hat{y}_i = x_i^T \hat{\beta} = y_i - \hat{\epsilon}_i$, like this.

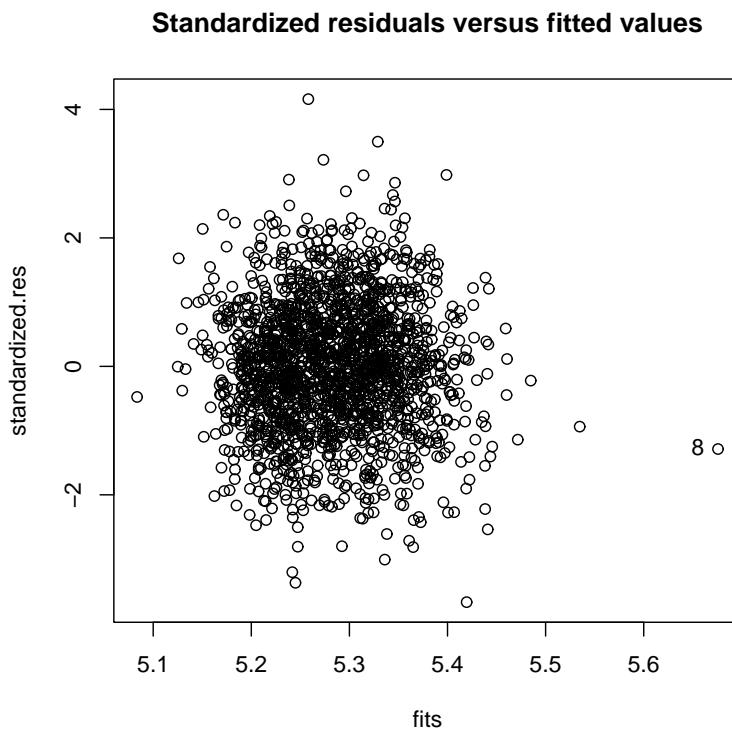
```
> result <- lm(log.chol ~ log.bmi + morph, data=body)
> fits <- body$log.chol - result$res
> s2 <- sum(result$res^2) / result$df.res
> lev <- lm.influence(result)$hat
> standardized.res <- result$res / sqrt(s2 * (1-lev))
>
> plot(fits, standardized.res, main="Standardized residuals versus fitted values")
```



If the variance is constant, then this plot should look like a “horizontal band” centered at zero. There should be no tendency for the standardized residuals to become more or less variable as we move across the plot from left to right. In this example, one observation has a fitted value much larger than the rest ($\hat{y}_i \approx 5.7$). This observation has high leverage. (If an observation has an unusually large or small fitted value, then it must have high leverage. But the converse is not always true; a high-leverage point may not necessarily have an unusual fitted value.) We can identify this point using the `identify` function. If you type

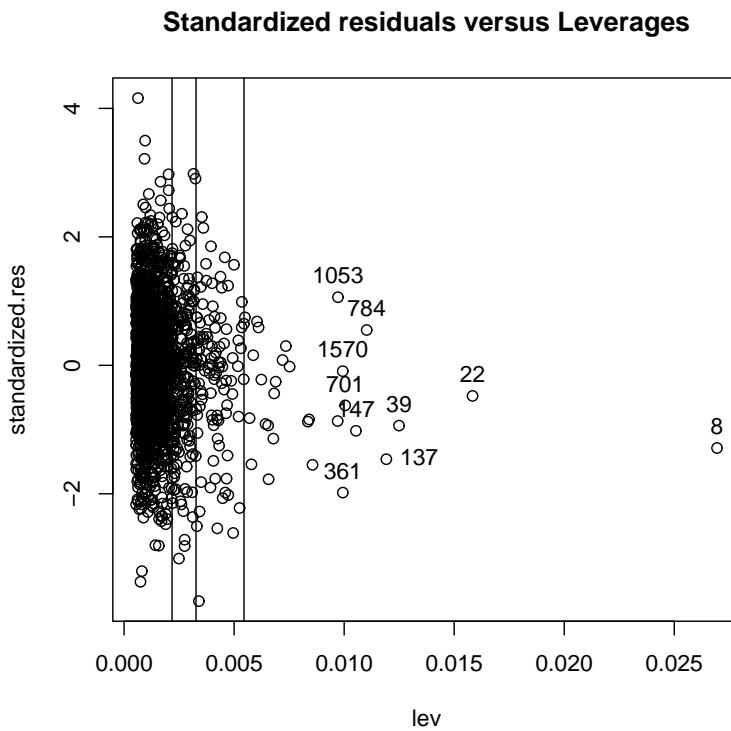
```
> identify( fits, standardized.res )
```

while the plot is displayed, the mouse cursor becomes a crosshair, and as you click on specific points, R identifies them. The unusual observation belongs to Subject #8.



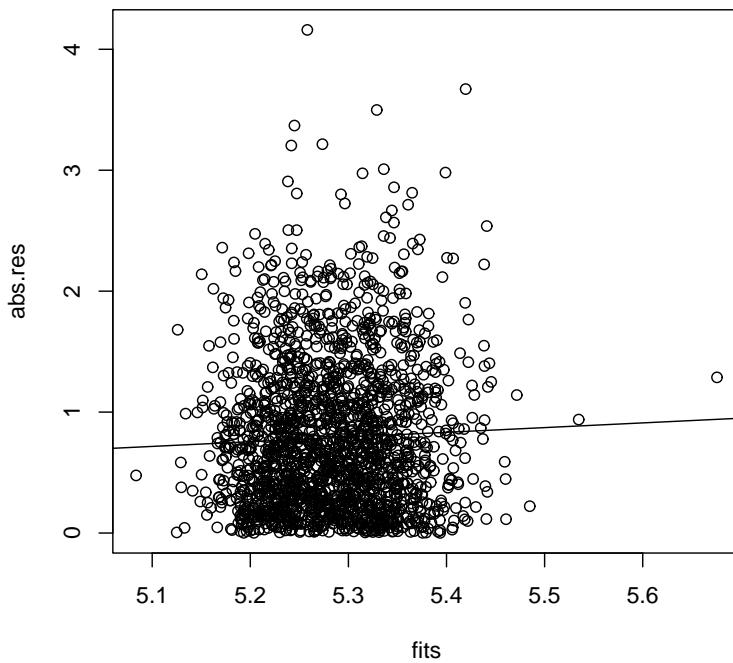
Let's plot the standardized residuals versus the leverage values and identify observations with potentially high influence. In this dataset, there are many observations with leverage exceeding $2p/n$, $3p/n$ and even $5p/n$. The person with the largest leverage is Subject #8.

```
> plot( lev, result$res, main="Residuals versus Leverages")
> p <- 2
> n <- nrow(body)
> abline( v = 2*p/n )
> abline( v = 3*p/n )
> abline( v = 5*p/n )
> identify( lev, result$res)
```

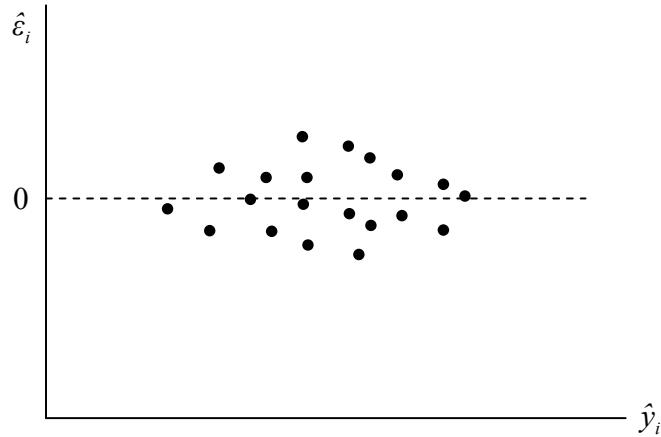


When we look at the plot of standardized residuals versus fitted values, it is not always easy to see whether the variance is constant. It may be useful to plot the absolute values of the standardized residuals, $|\tilde{\epsilon}_i|$, versus the fitted values. If we add a least-squares line to this plot, we can easily see whether there is a tendency for the magnitude of the residuals to increase or decrease with the fitted values.

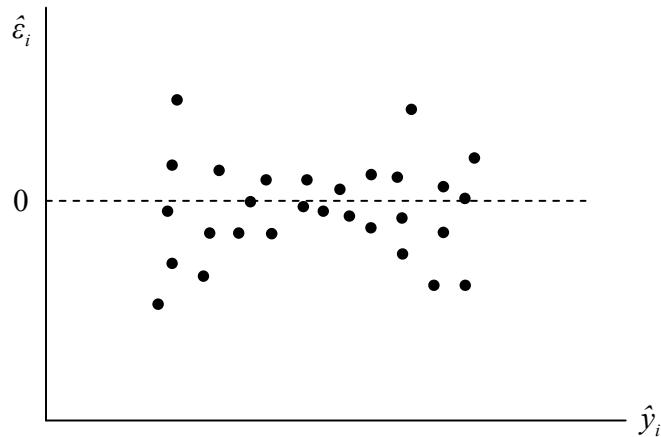
```
> abs.res <- abs(standardized.res)
> plot( fits, abs.res )
> abline( lsfit( fits, abs.res ) )
```



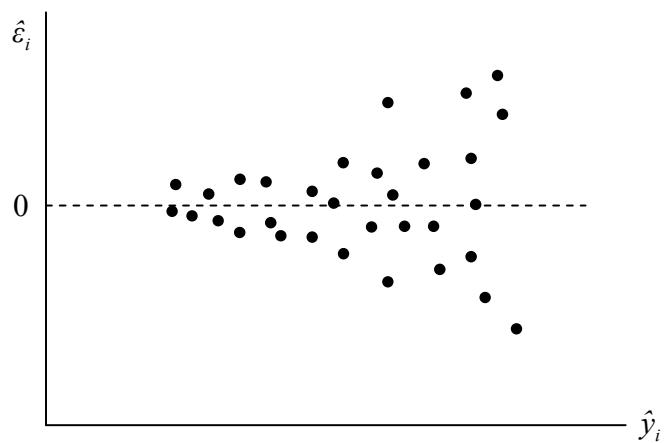
From this plot, it does appear that the variance of the response mildly increases with the mean. This technique is not foolproof, because if the variance is related to the mean in a non-monotonic fashion—e.g. it increases and then decreases, or vice-versa—then the least-squares line might be horizontal. But if that happens, you will probably have realized it already, because the plot of residuals versus fitted values will resemble a football,



or a butterfly:



The most common type of heteroscedasticity, however, is when the variance increases with the mean. In that case, the plot of residuals versus fitted values fans out, like this.



This often happens when the response variable is positively skewed. Transforming the variable to reduce the skewness (e.g. by taking the log) often solves the problem.

ANALYSIS OF RESIDUALS, PART II

Why is heteroscedasticity a problem? Last time, we defined the standardized residuals

$$\tilde{\epsilon}_i = \frac{\hat{\epsilon}_i}{S\sqrt{1-h_i}}$$

and plotted them against the fitted values \hat{y}_i to look for evidence of heteroscedasticity (non-constant variance). Heteroscedasticity, if present, is problematic in some ways but not others. It does not cause the ordinary least-squares (OLS) estimates $\hat{\beta}$ to be biased. But it does make OLS inefficient. Heteroscedasticity means that some observations are less reliable than others and should be down-weighted in the fitting procedure. Heteroscedasticity is a sign that OLS is not optimal, and we can do better by switching to weighted least squares (WLS) (more about that later).

The more serious problem with heteroscedasticity is that it can distort standard errors. The result

$$V(\hat{\beta}) = \sigma^2(X^T X)^{-1}$$

is true when the errors have constant variance. But when the errors are heteroscedastic, $S^2(X^T X)^{-1}$ can be a poor

estimate of the true covariance matrix for $\hat{\beta}$, making tests and confidence intervals inaccurate. Next month, we will discuss another estimate for $V(\hat{\beta})$ which remains accurate in large samples even when the homoscedasticity assumption is violated.

Clarification on the behavior of standardized residuals. Last time, we pointed out that the standardized residual $\tilde{\epsilon}_i$ is not distributed as t_{n-p} because the numerator and denominator are not independent. When $n \gg p$, we can regard them as approximately t_{n-p} on $N(0, 1)$, and we said that any observation for which $|\tilde{\epsilon}_i|$ exceeds 3 or 4 is a potential outlier.

In smaller samples, however, the dependence between the numerator and denominator keeps them from getting too large, because a large raw residual in the numerator also increases the size of S . In fact, it is possible to show that they are bounded by

$$|\tilde{\epsilon}_i| \leq \sqrt{n-p}.$$

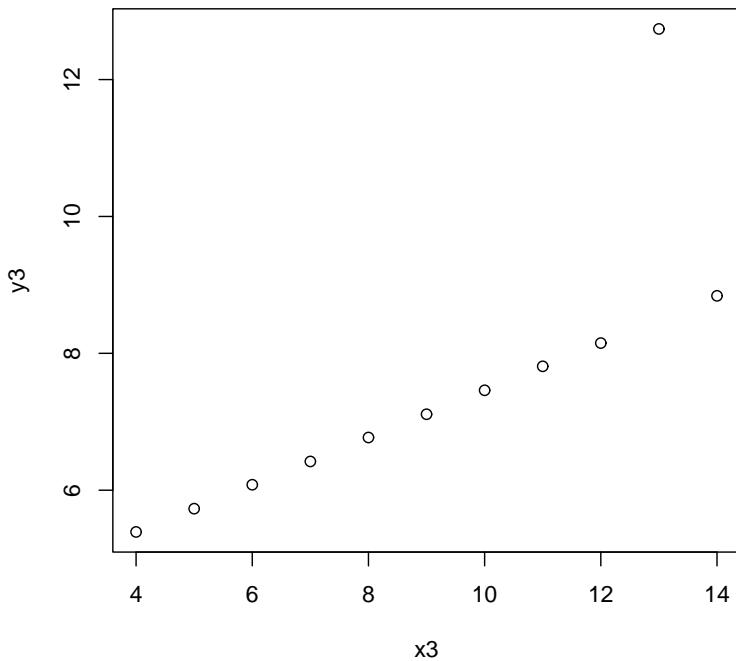
(see Gray and Woodall (1994), *The American Statistician*, 48, 111–113). For example, with punting dataset, we built a model with $p = 6$ coefficients from only $n = 13$ observations. In that case, it is impossible for the magnitude of any $\tilde{\epsilon}_i$ to exceed $\sqrt{7} = 2.65$.

The upper bound of $|\tilde{\epsilon}_i| = \sqrt{n-p}$ will be achieved when deleting observation i produces a model that fits the

remaining data points perfectly ($R^2 = 1$). We have seen such a dataset. Back in Lecture 12, we plotted the famous four bivariate datasets published by Anscombe (1973), each with $n = 11$. In the third dataset, ten of the observations followed a perfect linear relationship.

```
> attach(anscombe)
> anscombe
   x1  x2  x3  x4      y1    y2    y3    y4
 1  10  10  10   8  8.04  9.14  7.46  6.58
 2   8   8   8   8  6.95  8.14  6.77  5.76
 3  13  13  13   8  7.58  8.74 12.74  7.71
 4   9   9   9   8  8.81  8.77  7.11  8.84
 5  11  11  11   8  8.33  9.26  7.81  8.47
 6  14  14  14   8  9.96  8.10  8.84  7.04
 7   6   6   6   8  7.24  6.13  6.08  5.25
 8   4   4   4  19  4.26  3.10  5.39 12.50
 9  12  12  12   8 10.84  9.13  8.15  5.56
10   7   7   7   8  4.82  7.26  6.42  7.91
11   5   5   5   8  5.68  4.74  5.73  6.89

> plot( anscombe$x3, anscombe$y3 )
```



The standardized residuals cannot exceed $\sqrt{11 - 2} = 3.00$.
Let's compute them and see.

```

> tmp <- lm( y3 ~ x3, data=anscombe)
> res <- tmp$res
> s2 <- sum( tmp$res^2 ) / tmp$df
> lev <- lm.influence(tmp)$hat
> r <- res/sqrt(s2*(1-lev))
> r
      1          2          3          4          5          6
-0.46017736 -0.19633304  2.99999172 -0.33085149 -0.59695076 -1.13497164
      7          8          9         10         11
  0.07041631  0.38069861 -0.75517652 -0.06973871  0.21188094
  
```

Externally Studentized residuals. To get rid of the dependence between the numerator and denominator in the standardized residual, we can replace the usual

estimate of σ in the denominator, $S = \sqrt{MSE}$, by the estimate we would get **if that observation were deleted**. If we omitted observation i from the model fitting procedure, then the residuals for all the other observations would become

$$y_j - x_j^T \hat{\beta}_{(i)}$$

for $j \neq i$. Note that this is not the PRESS residual, because we have omitted subject i , not subject j . If we refit the model with subject i deleted, the residual sum of squares would be

$$SS_{Err(i)} = \sum_{j \neq i} (y_j - x_j^T \hat{\beta}_{(i)})^2,$$

and the corresponding estimate of σ^2 becomes

$$S_{(i)}^2 = \frac{1}{n-p-1} SS_{Err(i)}.$$

The externally Studentized residual is

$$\tilde{\epsilon}_{(i)} = \frac{\hat{\epsilon}_i}{S_{(i)} \sqrt{1-h_i}}$$

which is just the standardized residual with S replaced by $S_{(i)}$. The externally Studentized residual is distributed as

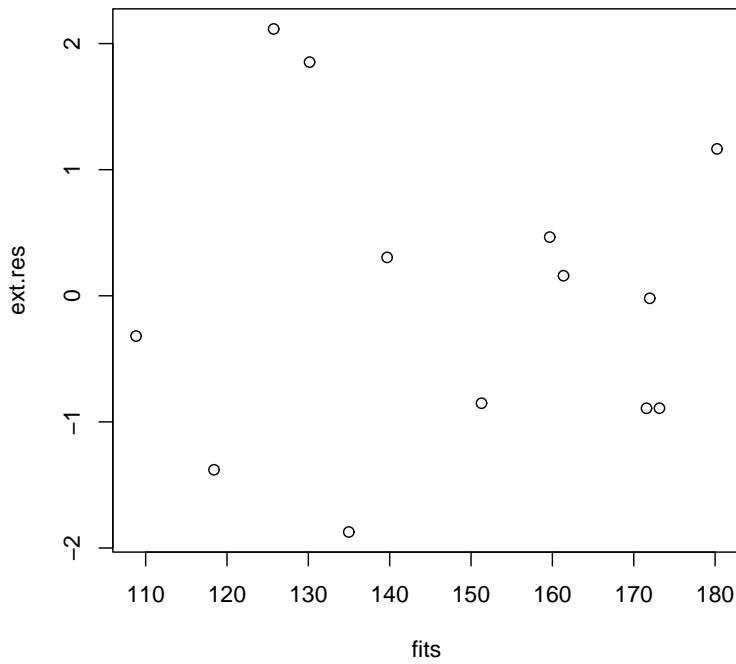
$$\tilde{\epsilon}_{(i)} \sim t_{n-p-1},$$

provided that the model $y_i \sim N(x_i^T \beta, \sigma^2)$ is true. It is important to note, however, that the externally

Studentized residuals **are not independent of each other**, because they make use of the same data.

The revised estimates of σ are available in R from the `lm.influence` function. The result from that function includes a component called `sigma`, which is the vector of $S(i)$ for $i = 1, \dots, n$. Here is an example showing how to compute and plot the externally Studentized residuals versus the fitted values using our punting data.

```
> result <- lm( Distance ~ R_Strength + L_Strength +
+   R_Flexibility + L_Flexibility + O_Strength, data=punt)
>
> fits <- punt$Distance - result$res
> tmp <- lm.influence(result)
> ext.res <- result$res / (tmp$sigma * sqrt(1-tmp$hat) )
> plot( fits, ext.res )
```



Some books indicate that the externally Studentized residuals are useful for outlier detection. A 95% prediction interval for a single $\tilde{\epsilon}_{(i)}$ is $\pm t_{.975, n-p-1}$, but **this assumed that the normal model is true.** There is no Central Limit Theorem that makes residuals approximately normal if the true errors $\epsilon_1, \dots, \epsilon_n$ are not normal. With our punting data, if the population of true errors was normally distributed, then a single externally Studentized residual would lie within $\pm t_{.975, 6} = 2.45$ with probability 95%. Note that this interval would be valid for a single observation chosen in advance. If we were to look at all n externally Studentized residuals to see if any of them lie outside these limits, then we need to adjust the confidence level of each interval to control the overall error rate. We can do this with a Bonferroni correction. To get Bonferroni-corrected intervals for all $n = 13$ observations, note that

$$1 - (.05/13)/2 = 0.9981,$$

so each externally Studentized residual should lie within $\pm t_{.9981, 6} = 4.57$ with simultaneous probability of at least 95%.

Another way to delete an observation. The most efficient way to compute regression quantities from a reduced dataset with observation i removed is to use the “leave-one-out” formula derived last week. But there is another way that is potentially useful conceptually, if not

computationally. We can effectively delete an observation **by adding a variable**. Suppose we augment the design matrix by an extra column E_i , which has 1 in position i and zeroes elsewhere. In other words, E_i is a dummy indicator for subject i . If we regress the response vector y on the $n \times (p + 1)$ design matrix

$$X^* = [X, E_i]$$

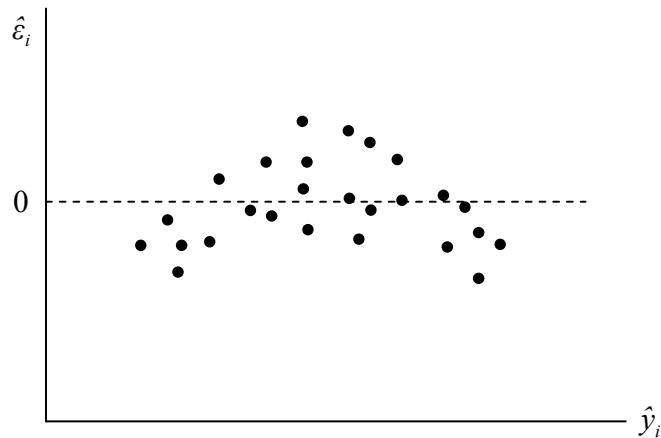
and get the least-squares estimates $\hat{\beta}^*$, then the first p elements of $\hat{\beta}^*$ will be the least-squares estimates $\hat{\beta}_{(i)}$ from regressing y on X with subject i removed. This new model will give a perfect fit to subject i ($\hat{y}_i = y_i$), and the residual for that subject will be $\hat{\epsilon}_i = 0$. The residual sum of squares from this augmented model will be equal to $SS_{Err(i)}$, the residual sum of squares from the regression of y on X with subject i removed. And the mean-squared error for this augmented model will be

$$S_{(i)}^2 = \frac{1}{n - p - 1} SS_{Err(i)}.$$

Adding a dummy indicator for a single observation effectively removes that observation from the model fit.

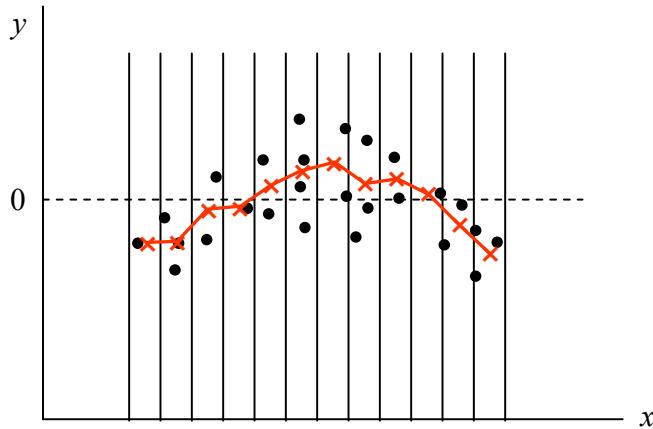
Checking for nonlinearity. Plots of residuals versus fitted values are traditionally used to check for heteroscedasticity, but they may also show evidence of nonlinearity. The residuals have an overall mean of zero. If the model is true, then they should also have a mean of

approximately zero within vertical strips where the fitted value (or any other linear combination of the columns of X) is constant. Curvature in this plot (as shown below) provides evidence that the relationship between the mean of Y and at least some of the predictors is non-linear.



Sometimes evidence of nonlinearity is difficult to detect by simply looking at the plot, and it helps to add a **loess curve**. “Loess” (sometimes called “lowess,” particularly in older sources) is the common name for **local polynomial regression**, a technique for smoothing scatterplots to obtain a nonparametric estimate for the conditional mean of one variable given another.

In a very rough sense, we can think of loess as dividing the scatterplot into vertical strips, computing the averages within each vertical strip, and connecting the averages.



The actual technique, however, is more subtle and more interesting. Suppose we have a set of responses y_1, \dots, y_n and a predictor x_1, \dots, x_n . Given these data, we want to compute an estimate of $E(Y | X)$ assuming only that $E(Y | X)$ is a smooth function of X . Because this function is smooth, we may say that

$$E(Y | X) \approx \beta_0^* + \beta_1^* X$$

for some β_0^* and β_1^* in a neighborhood of $X = x^*$. (This is called “local linear regression,” for obvious reasons.) To estimate β_0^* and β_1^* , we regress y_1, \dots, y_n on x_1, \dots, x_n by weighted least squares, using a weighting function that gives greatest weight to observations for which x_i is near x^* , and downweighting the observations for which x_i is far from x^* . By varying x^* over a range of values and recomputing the regression at each x^* , we get a smooth curve that estimates $E(Y | X)$ with very few assumptions.

Various choices for the weighting function are given in the literature, along with guidelines for choosing among them. A good reference on this and other nonparametric regression techniques is the book *Generalized Additive Models* by Hastie and Tibshirani (1990).

In R, we can easily compute a local linear regression fit and add it to a scatterplot. The function `loess` will computes a loess fit. The required arguments to this function are

- a model formula (e.g., $y \sim x$), similar to the formula used in `lm`, and
- the data frame containing the variables appearing in the model formula.

Optional arguments allow you to specify the weighting function used in the loess fitting procedure, but we will ignore these and let R use its default values.

To plot the loess curve, we need to call the function `predict`. The two necessary arguments to `predict` are

- the result from `loess`, and
- `newdata`, a new data frame containing the grid of predictor values at which the loess predictions are to be generated. The names of the variables in `newdata` must match the names of the predictors in the model formula in the call to `loess`.

The function `predict` also has an optional logical argument `se`. If `se=F` (the default), then `predict` computes

only the loess predictions at the values of the predictors in `newdata`. If you change it to `se=T`, then `predict` will also compute standard errors for these predictions, which are analogous to $\text{SE}(\hat{y}(x^*))$ in linear regression.

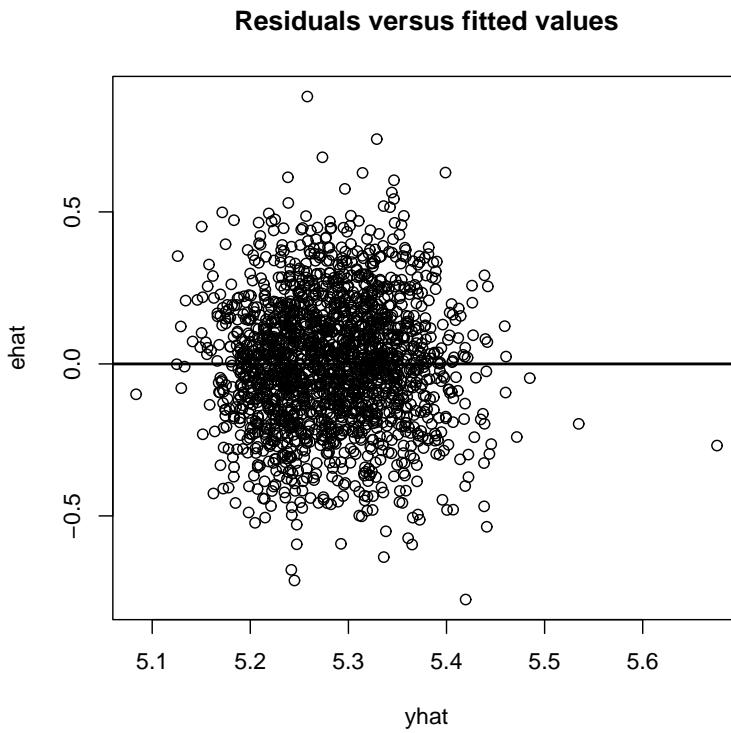
Let's demonstrate this by example using our body measurements data. First, let's regress $\log(\text{CHOL})$ on $\log(\text{BMI})$ and MORPH and plot the raw residuals versus the fitted values.

```
> body <- read.table("body.dat", header=T)

> meters <- body$height * 2.54 / 100 # height in meters
> kg <- body$weight * 0.45359237 # weight in pounds
> body$log.bmi <- log(kg/meters^2)
> body$log.chol <- log( body$chol )
> body$morph <- body$waist / body$hips

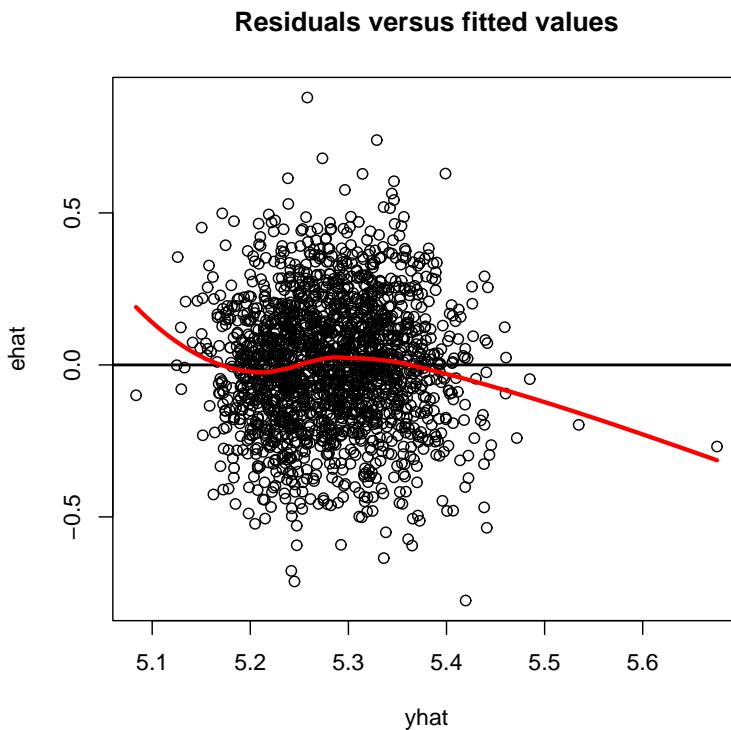
> result <- lm( log.chol ~ log.bmi + morph, data=body)

> ehat <- result$residuals
> yhat <- body$log.chol - ehat
> plot( yhat, ehat, main="Residuals versus fitted values")
> abline( h=0, lwd=2 ) # add a thick horizontal line at zero
```



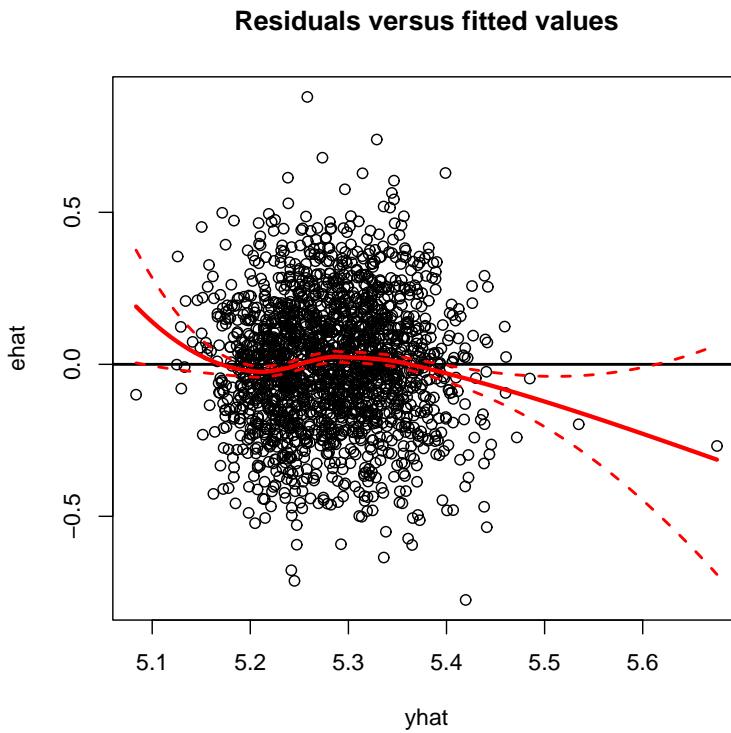
If the model were true, the mean residual within each “vertical slice” of this plot should be approximately zero. For reference, I added a thick horizontal line at zero to the plot. Now let’s compute the loess predictions at a grid of 200 points equally spaced over the range of the \hat{y}_i ’s, and plot the loess predictions as a very thick red line.

```
> loess.fit <- loess( y ~ x, data.frame(x=yhat, y=ehat) )
> yhat.grid <- seq( from=min(yhat), to=max(yhat), length=200)
> tmp <- predict( loess.fit, newdata=data.frame(x=yhat.grid), se=T )
> lines( yhat.grid, tmp$fit, lwd=3, col=2 )
```



The loess predictions are most precise near the center of the \hat{y}_i 's and least precise at the edges of the plot. For reference, let's plot intervals equal to the loess predictions plus or minus two SE's. These can be regarded as approximate 95% (pointwise, not simultaneous) confidence intervals for the mean residual at each value of \hat{y}_i .

```
> lines( yhat.grid, tmp$fit+2*tmp$se.fit, lwd=2, lty=2, col=2)
> lines( yhat.grid, tmp$fit-2*tmp$se.fit, lwd=2, lty=2, col=2)
```



It's now obvious that the linear regression model

$$\log(\text{CHOL}) = \beta_0 + \beta_1 \log(\text{BMI}) + \beta_2 \text{MORPH} + \text{error}$$

systematically underpredicts $\log(\text{CHOL})$ in some regions and overpredicts in others.

When the plot of residuals versus fitted values reveals non-linear trends, the possible remedies include

- transforming the response,
- transforming one or more predictors, and
- introducing additional terms into the model to account for the nonlinear relationships.

Moving across the plot from left to right, we can identify

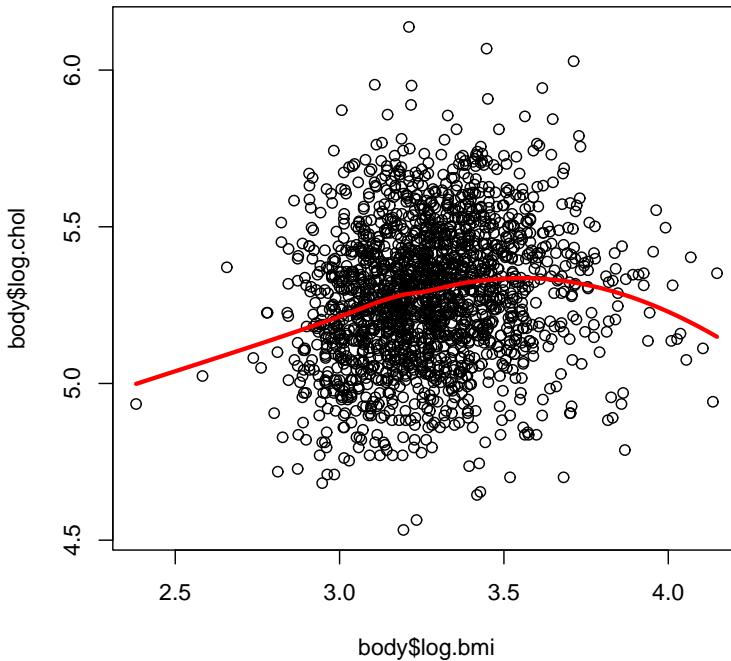
four regions where the residuals tend to be positive, then negative, then positive, then negative. If there were only three such regions (e.g., negative, positive, negative) then a simple monotonic transformation of the response might fix the problem. But the pattern is complicated. Moreover, we have already applied a log transformation to the response, and the log transformation here seems desirable because

- it is easy to interpret, and
- it makes the residuals approximately normally distributed.

So, in this example, we probably don't want to transform the response; it would be better to try to fix up the model by transforming the predictors or adding extra terms.

Plotting the response versus each predictor. Before even fitting a model, it's usually wise to plot the response against each predictor. Adding loess curves to these plots can be extremely informative, especially when the sample is large. Let's plot $\log(\text{CHOL})$ versus $\log(\text{BMI})$.

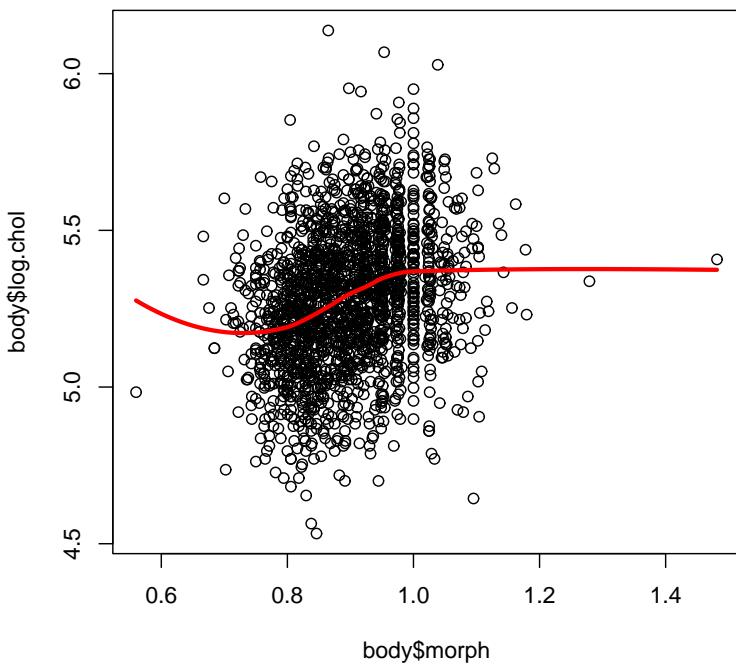
```
> plot( body$log.bmi, body$log.chol )
> loess.fit <- loess( log.chol ~ log.bmi, data=body)
> log.bmi.grid <- seq( from=min(body$log.bmi), to=max(body$log.bmi),
+   length=200)
> tmp <- predict( loess.fit,
+   newdata=data.frame( log.bmi=log.bmi.grid), se=T )
> lines( log.bmi.grid, tmp$fit, lwd=3, col=2)
```



The trend is clearly nonlinear. We could possibly account for this trend by supposing that the relationship is quadratic, i.e. by including $\log(\text{BMI})$ and $(\log(\text{BMI}))^2$ in the model. Quadratic trends are a very simple fix, but they often do not describe the data well in the outlying regions of the predictor space. (In this plot, the trend on the left side looks linear, not quadratic.)

Now let's plot $\log(\text{CHOL})$ versus MORPH.

```
> plot( body$morph, body$log.chol )
> loess.fit <- loess( log.chol ~ morph, data=body)
> morph.grid <- seq( from=min(body$morph), to=max(body$morph),
+   length=200)
> tmp <- predict( loess.fit,
+   newdata=data.frame( morph=morph.grid), se=T )
> lines( morph.grid, tmp$fit, lwd=3, col=2)
```



This trend looks approximately quadratic on the left side and flat on the right. It would be difficult to capture using transformations or polynomials. Next time, we will present strategies to account for these nonlinear relationships.

ACCOUNTING FOR NONLINEAR RELATIONSHIPS

Last time, we discussed how to check assumptions of linearity by looking for curvature in

- the plot of residuals versus the fitted values, and
- plots of the response versus each predictor.

Plotting the response variable against each predictor helps us to understand the relationships between the response and the predictors one at a time. When the predictors are correlated, it is also useful to investigate the relationship between the response and each predictor accounting for the other predictors.

Partial residual plots. Suppose that the columns of the design matrix are

$$X = [1, X_1, X_2, \dots, X_{p-1}].$$

And suppose that we

- regress y on all columns of X except X_j and compute the residuals,
- regress X_j on all columns of X except X_j and compute the residuals, and

- plot the residuals from y against the residuals from X_j .

This is called a **partial residual plot**. (Note: Some statisticians call it an added-variable plot. But that term is also used for another kind of plot. There is a lot of confusion here. We will call it a partial residual plot.)

This plot has some interesting properties.

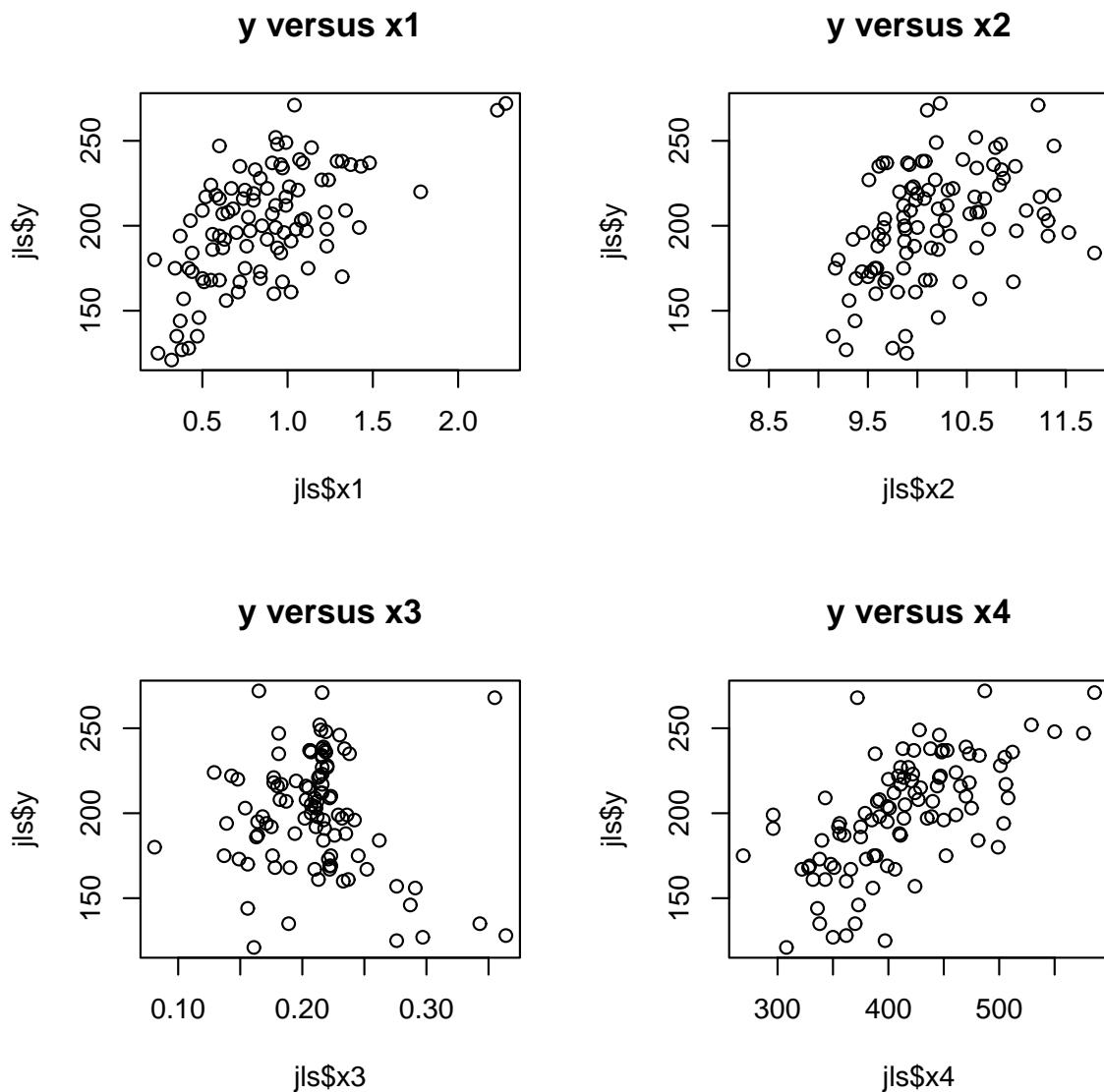
- The correlation coefficient between these residuals is the partial correlation between y and X_j given all the other predictors.
- If you fit a linear regression to the data in this plot, the estimated intercept will be zero, and the estimated slope will be equal to the coefficient of X_j in the full model.

Moreover, if the linear model holds, then the relationships in all of these plots should be linear.

Example. The datafile `jls.dat` contains $n = 100$ observations on five variables, which are labelled `x1`, `x2`, `x3`, `x4` and `y`. The data were artificially generated for a simulation study. Plots of `y` versus `x1`, `x2`, `x3`, and `x4` are shown below.

```
> jls <- read.table("jls.dat",header=T)
> par(mfrow=c(2,2))
> plot( jls$x1, jls$y, main="y versus x1" )
> plot( jls$x2, jls$y, main="y versus x2" )
```

```
> plot( jls$x3, jls$y, main="y versus x3" )
> plot( jls$x4, jls$y, main="y versus x4" )
```



The relationships seem approximately, but not exactly linear. But these four predictors are related to each other, and it is worthwhile to examine the relationship between y and each predictor accounting for the other predictors. Here are the partial residual plots.

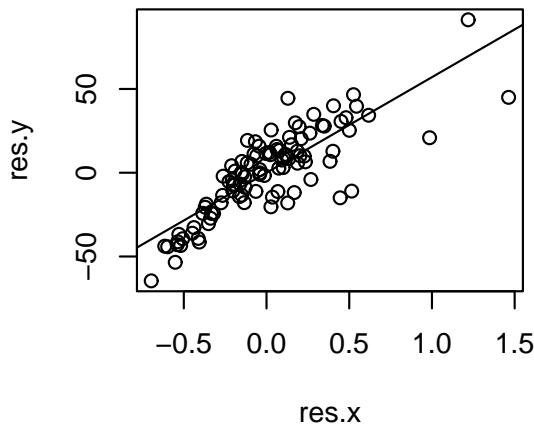
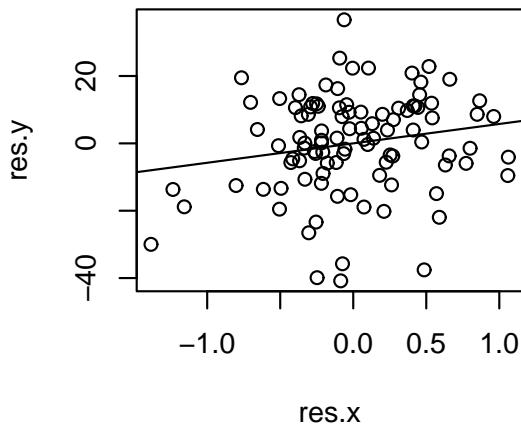
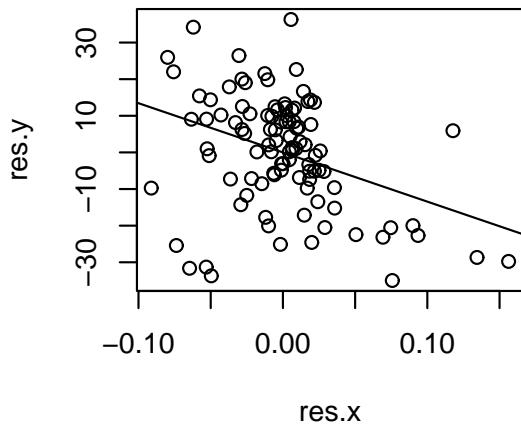
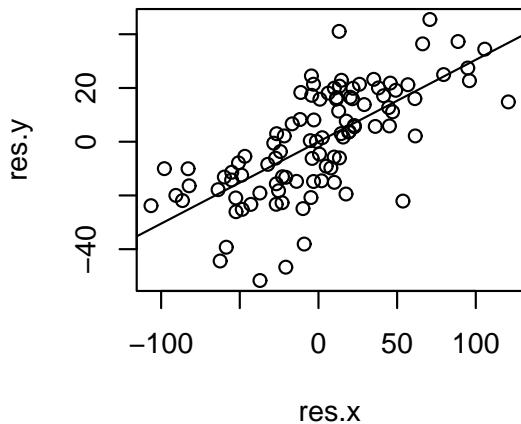
```
> par(mfrow=c(2,2))

> res.y <- lm( y ~ x2 + x3 + x4, data=jls)$res
> res.x <- lm( x1 ~ x2 + x3 + x4, data=jls)$res
> plot(res.x, res.y, main="Partial residual plot for x1")
> abline( lsfit( res.x, res.y) )

> res.y <- lm( y ~ x1 + x3 + x4, data=jls)$res
> res.x <- lm( x2 ~ x1 + x3 + x4, data=jls)$res
> plot(res.x, res.y, main="Partial residual plot for x2")
> abline( lsfit( res.x, res.y) )

> res.y <- lm( y ~ x1 + x2 + x4, data=jls)$res
> res.x <- lm( x3 ~ x1 + x2 + x4, data=jls)$res
> plot(res.x, res.y, main="Partial residual plot for x3")
> abline( lsfit( res.x, res.y) )

> res.y <- lm( y ~ x1 + x2 + x3, data=jls)$res
> res.x <- lm( x4 ~ x1 + x2 + x3, data=jls)$res
> plot(res.x, res.y, main="Partial residual plot for x4")
> abline( lsfit( res.x, res.y) )
```

Partial residual plot for x1**Partial residual plot for x2****Partial residual plot for x3****Partial residual plot for x4**

Notice that the relationship between y and x_1 has gotten stronger after accounting for the other predictors. Our eyes can detect some mild evidence of nonlinearity in the first plot. In a moment, we will add loess curves. Before we do, however, let's print out the estimated coefficients from the simple linear regression for each plot.

```

> res.y <- lm( y ~ x2 + x3 + x4, data=jls)$res
> res.x <- lm( x1 ~ x2 + x3 + x4, data=jls)$res
> lm( res.y ~ res.x)$coef
  (Intercept)      res.x
8.002409e-16 5.702536e+01

> res.y <- lm( y ~ x1 + x3 + x4, data=jls)$res
> res.x <- lm( x2 ~ x1 + x3 + x4, data=jls)$res
> lm( res.y ~ res.x)$coef
  (Intercept)      res.x
1.332580e-16 5.765792e+00

> res.y <- lm( y ~ x1 + x2 + x4, data=jls)$res
> res.x <- lm( x3 ~ x1 + x2 + x4, data=jls)$res
> lm( res.y ~ res.x)$coef
  (Intercept)      res.x
3.716732e-16 -1.344435e+02

> res.y <- lm( y ~ x1 + x2 + x3, data=jls)$res
> res.x <- lm( x4 ~ x1 + x2 + x3, data=jls)$res
> lm( res.y ~ res.x)$coef
  (Intercept)      res.x
2.389654e-16 3.049503e-01

```

Notice that all of the estimated intercepts are zero. The estimated slopes coincide with the slopes from the multiple linear regression of y on all four predictors.

```

> summary( lm( y ~ x1 + x2 + x3 + x4, data=jls) )

Call:
lm(formula = y ~ x1 + x2 + x3 + x4, data = jls)

Residuals:
    Min      1Q  Median      3Q      Max 
-40.367 -8.108   1.794   9.093  37.048 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.87031   26.00819  -0.187 0.851856

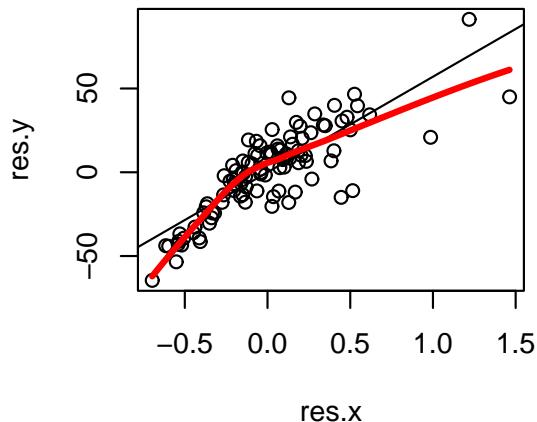
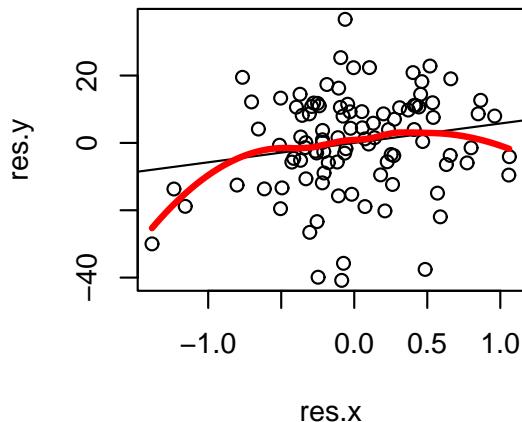
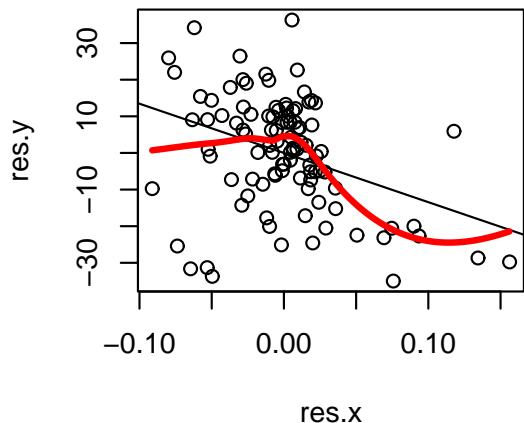
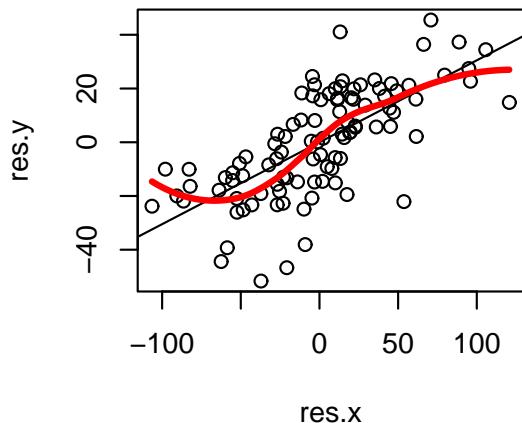
```

```
x1          57.02536   4.02065  14.183 < 2e-16 ***
x2          5.76579   3.13793   1.837  0.069269 .
x3         -134.44354  35.22546  -3.817  0.000241 ***
x4          0.30495   0.03256   9.367  3.72e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

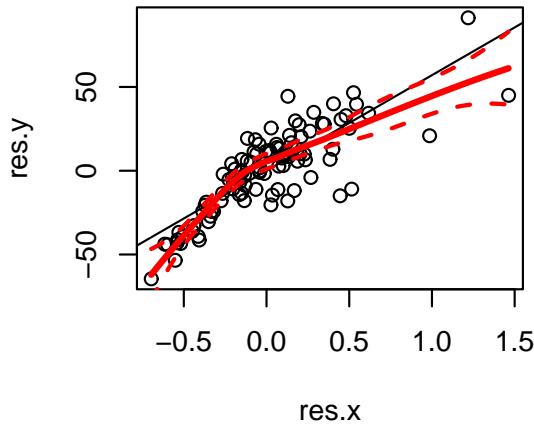
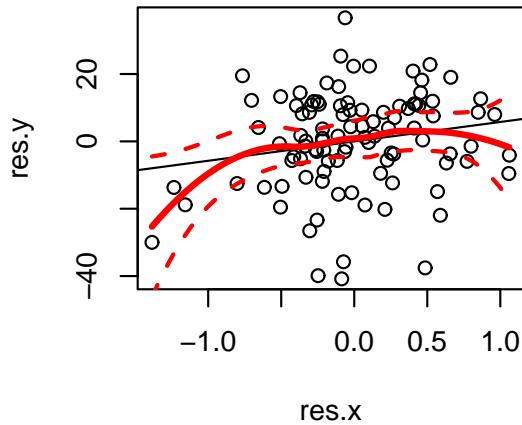
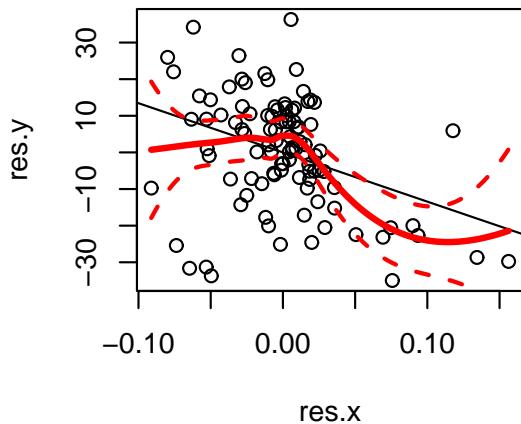
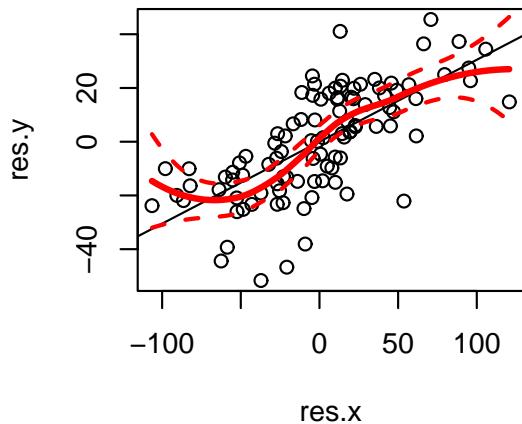
Residual standard error: 14.75 on 95 degrees of freedom
Multiple R-Squared:  0.8125,    Adjusted R-squared:  0.8046
F-statistic: 102.9 on 4 and 95 DF,  p-value: < 2.2e-16
```

Now let's re-display the partial residual plots, adding a loess curve to each one. (R code is shown below only for the first plot.)

```
> res.y <- lm( y ~ x2 + x3 + x4, data=jls)$res
> res.x <- lm( x1 ~ x2 + x3 + x4, data=jls)$res
> plot(res.x, res.y, main="Partial residual plot for x1")
> abline( lsfit( res.x, res.y ) )
> loess.fit <- loess( y ~ x, data=data.frame(x=res.x,y=res.y))
> x.grid <- seq( from=min(res.x), to=max(res.x), length=200)
> tmp <- predict( loess.fit, newdata=data.frame( x=x.grid ), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=2)
```

Partial residual plot for x1**Partial residual plot for x2****Partial residual plot for x3****Partial residual plot for x4**

It is tempting to conclude that all the relationships are nonlinear. But loess curves can be noisy when n is not large. When you plot loess curves, it's wise to add confidence intervals to see if the deviations from linearity are significant.

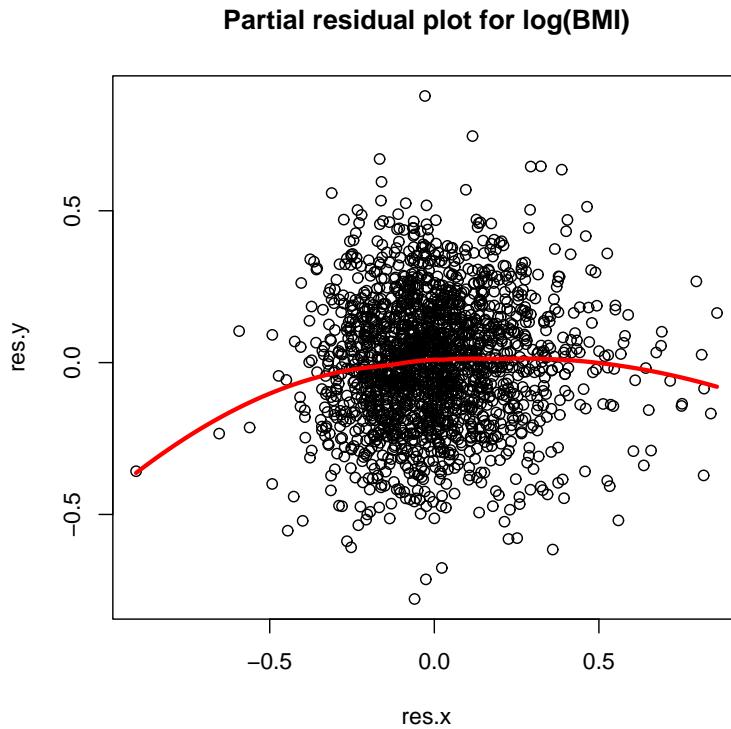
Partial residual plot for x1**Partial residual plot for x2****Partial residual plot for x3****Partial residual plot for x4**

Judging from the intervals, I would say that the plot for x1, and perhaps the plot for x3, show mild evidence of nonlinearity.

Another example. Let's generate the partial residual plots for our body measurements data. This dataset has more than 1,800 observations and only two predictors, so

the loess curves will be very precise. First, the partial residual plot for $\log(\text{CHOL})$ versus $\log(\text{BMI})$ accounting for MORPH.

```
> res.y <- lm( log.chol ~ morph, data=body)$res
> res.x <- lm( log.bmi ~ morph, data=body)$res
> plot(res.x, res.y, main="Partial residual plot for log(BMI)")
> loess.fit <- loess( y ~ x, data=data.frame(x=res.x,y=res.y))
> x.grid <- seq( from=min(res.x), to=max(res.x), length=200)
> tmp <- predict( loess.fit, newdata=data.frame( x=x.grid), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=2)
```



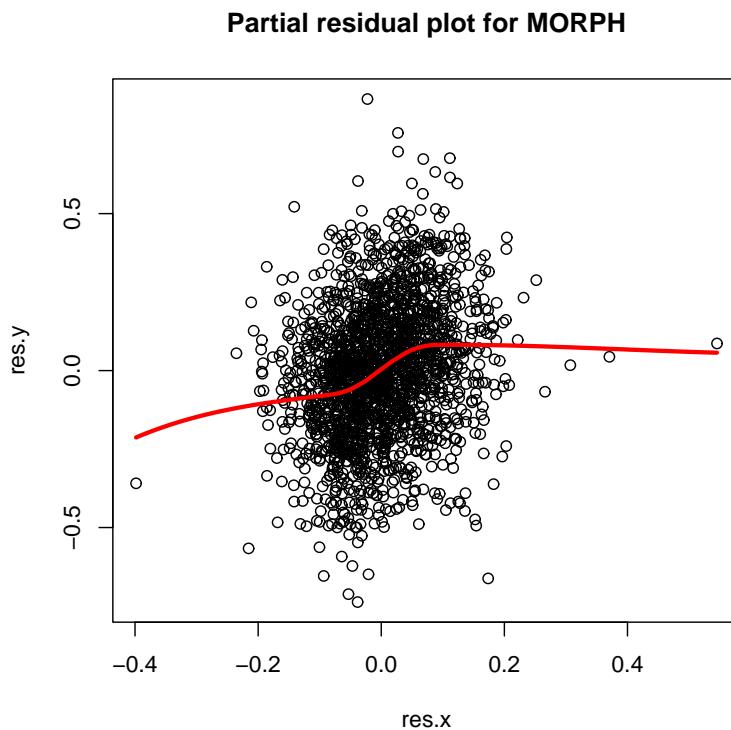
It seems that it may be reasonable to describe the effect of $\log(\text{BMI})$ on $\log(\text{CHOL})$ with a quadratic trend.

Now let's examine the partial residual plot for $\log(\text{CHOL})$ versus MORPH accounting for $\log(\text{BMI})$.

```

> res.y <- lm( log.chol ~ log.bmi, data=body)$res
> res.x <- lm( morph ~ log.bmi, data=body)$res
> plot(res.x, res.y, main="Partial residual plot for MORPH")
> loess.fit <- loess( y ~ x, data=data.frame(x=res.x,y=res.y))
> x.grid <- seq( from=min(res.x), to=max(res.x), length=200)
> tmp <- predict( loess.fit, newdata=data.frame( x=x.grid), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=2)

```



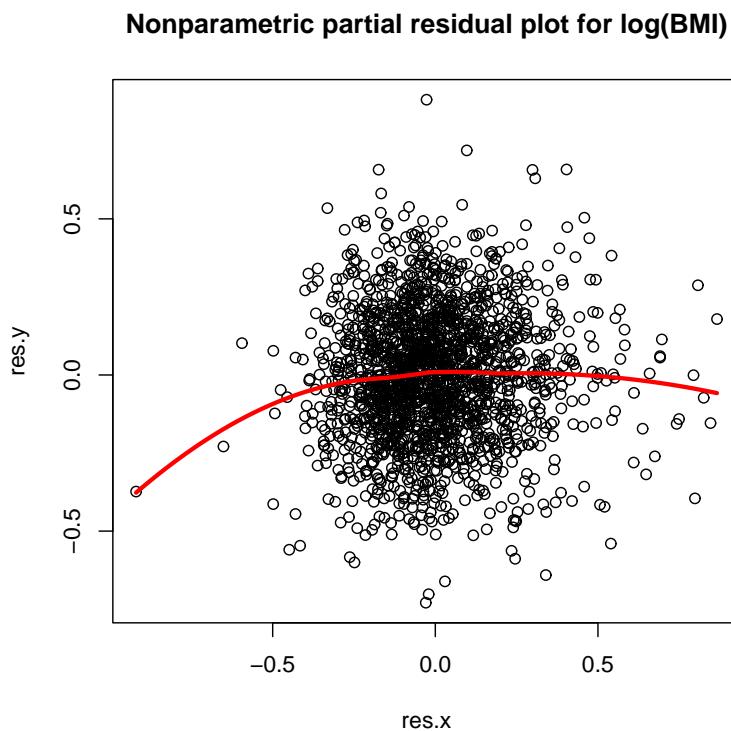
This trend looks something like a piecewise linear function with two abrupt changes in slope.

There's a problem with the partial residual plots. Each of these plots may reveal a nonlinear trend between the response and one predictor, **but it implicitly assumes that the effects of the other predictors (the ones that we are controlling for) are linear**. But what if all the effects are nonlinear? Can we assess the relationship

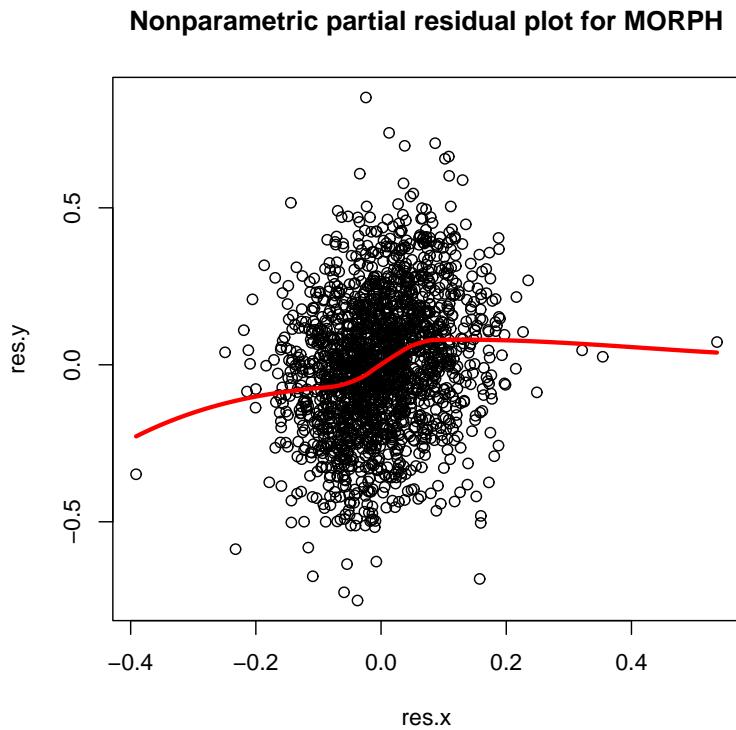
between $\log(\text{CHOL})$ and $\log(\text{BMI})$ controlling for MORPH, but without assuming that the effect of MORPH is linear? And can we assess the relationship between $\log(\text{CHOL})$ and MORPH controlling for $\log(\text{BMI})$, but without assuming that the effect of $\log(\text{BMI})$ is linear?

Nonparametric partial residual plots. If n is large and the number of predictors is small, we can use the following trick to assess the relationship between the response and each predictor without assuming that the effects of the other predictors are linear. The trick is: when calculating the residuals for the partial residual plot, use a loess fit rather than a least-squares fit.

```
> res.y <- loess( log.chol ~ morph, data=body)$res
> res.x <- loess( log.bmi ~ morph, data=body)$res
> plot(res.x, res.y, main="Nonparametric partial residual plot for log(BMI)")
> loess.fit <- loess( y ~ x, data=data.frame(x=res.x,y=res.y))
> x.grid <- seq( from=min(res.x), to=max(res.x), length=200)
> tmp <- predict( loess.fit, newdata=data.frame( x=x.grid), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=2)
```



```
> res.y <- loess( log.chol ~ log.bmi, data=body)$res
> res.x <- loess( morph ~ log.bmi, data=body)$res
> plot(res.x, res.y, main="Nonparametric partial residual plot for MORPH")
> loess.fit <- loess( y ~ x, data=data.frame(x=res.x,y=res.y))
> x.grid <- seq( from=min(res.x), to=max(res.x), length=200)
> tmp <- predict( loess.fit, newdata=data.frame( x=x.grid), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=2)
```



These nonparametric versions of the partial residual plots are most useful when there are many observations but only a few variables. In that case, you have lots of information to discern the relationship between the mean response and each predictor in the presence of the others. When p is large, it becomes difficult to adjust for the other variables without imposing parametric assumptions on those relationships.

Accounting for nonlinear trends through polynomial regression. If we detect strong evidence of a nonlinear relationship between the response and a

predictor, then one way to account for it is to suppose that

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_k X^k + \text{error}. \quad (1)$$

The value k is called the **degree** of the polynomial. When $k = 2$, the relationship is quadratic; when $k = 3$, the trend is cubic; and so on. The model (1) is written as though there is a single predictor X , but it should be understood that other predictors could enter the model as well. We can sometimes choose an appropriate value for k by inspection, but we may also formally compare models with different values of k by partial F-tests.

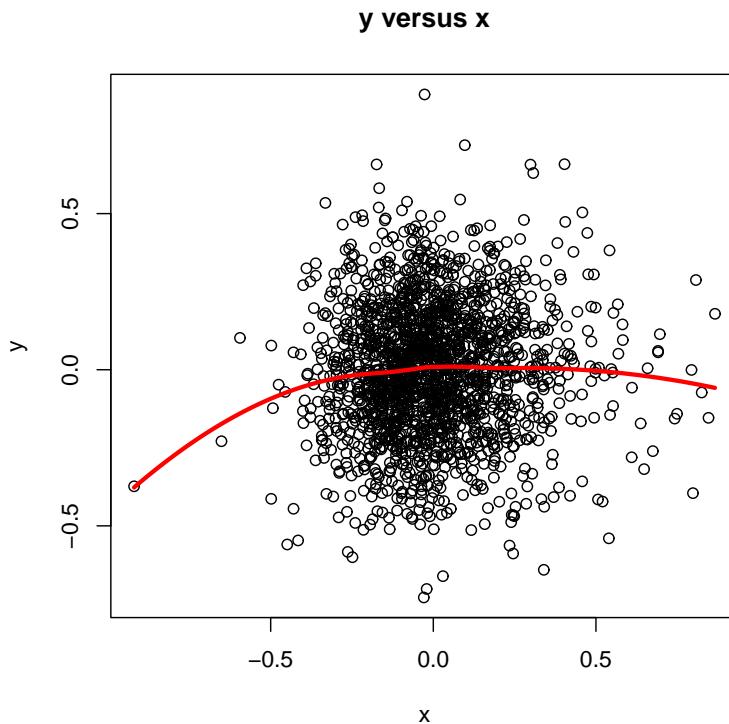
Let's use polynomials with our body measurements dataset to describe the relationship between $\log(\text{CHOL})$ and $\log(\text{BMI})$. Eventually, we want to add MORPH to the model as well, and the relationship between $\log(\text{CHOL})$ and $\log(\text{BMI})$ might be distorted if we do not properly account for the effect of MORPH. For now, let's "partial out" the effects of MORPH with minimal assumptions by working with the residuals from a nonparametric regression on MORPH. That is, we will investigate models of the form (1) where

- Y is the residuals from the loess fit of $\log(\text{CHOL})$ on MORPH, and
- X is the residuals from the loess fit of $\log(\text{BMI})$ on MORPH.

```

> y <- loess( log.chol ~ morph, data=body)$res
> x <- loess( log.bmi ~ morph, data=body)$res
> plot( x, y, main="y versus x")
> loess.fit <- loess( y ~ x, data=data.frame(x=x,y=y))
> x.grid <- seq( from=min(x), to=max(x), length=200)
> tmp <- predict( loess.fit, newdata=data.frame( x=x.grid), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=2)

```



Let's fit the linear regression of Y on X .

```

> summary( lm( y ~ x ) )

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.7270922 -0.1345360 -0.0001757  0.1365269  0.8838222 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.0001757  0.0001757  1.000   0.317    
x            0.1365269  0.0001757  78.000   <2e-16 ***

```

```

      Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0009821  0.0049032 -0.200  0.8413
x            0.0432563  0.0249088  1.737  0.0826 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.21 on 1833 degrees of freedom
Multiple R-Squared: 0.001643,   Adjusted R-squared: 0.001098
F-statistic: 3.016 on 1 and 1833 DF,  p-value: 0.08263

```

The slope is positive and nearly significant ($p = .083$).

Now let's fit the quadratic model:

```

> x2 <- x^2
> summary( lm( y ~ x + x2) )

Call:
lm(formula = y ~ x + x2)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.733407 -0.135415  0.001191  0.136065  0.877427 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.006187  0.005642  1.097  0.2729    
x           0.067105  0.026563  2.526  0.0116 *  
x2          -0.183514  0.071780 -2.557  0.0106 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2097 on 1832 degrees of freedom
Multiple R-Squared: 0.005192,   Adjusted R-squared: 0.004106 
F-statistic: 4.781 on 2 and 1832 DF,  p-value: 0.008496

```

The quadratic term is significant, which agrees with our assessment from the plot. Now the cubic model:

```

> x3 <- x^3
> summary( lm( y ~ x + x2 + x3) )

Call:
lm(formula = y ~ x + x2 + x3)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.734492 -0.135277  0.002067  0.135825  0.876380 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.006771  0.005683   1.191   0.2337  
x           0.048898  0.033960   1.440   0.1501  
x2          -0.215965  0.081087  -2.663   0.0078 **  
x3          0.127397  0.148038   0.861   0.3896  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2097 on 1831 degrees of freedom
Multiple R-Squared: 0.005594,   Adjusted R-squared: 0.003965 
F-statistic: 3.433 on 3 and 1831 DF,   p-value: 0.01637

```

The cubic term seems unnecessary. Now the quartic model:

```

> x4 <- x^4
> summary( lm( y ~ x + x2 + x3 + x4) )

Call:
lm(formula = y ~ x + x2 + x3 + x4)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.7331 -0.1352  0.0022  0.1362  0.8777 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.005304  0.006283   0.844   0.399  
x           0.046536  0.034239   1.359   0.174  
x2          -0.154259  0.138773  -1.112   0.266  
x3          0.145250  0.151608   0.958   0.338  
x4          -0.149580  0.272967  -0.548   0.584  

```

```
Residual standard error: 0.2098 on 1830 degrees of freedom
Multiple R-Squared:  0.005757,   Adjusted R-squared:  0.003584
F-statistic: 2.649 on 4 and 1830 DF,  p-value: 0.03182
```

When we added the quartic term, the quadratic became insignificant. What happened? The columns in the design matrix for X^2 and X^4 are correlated, and they are sharing significance. In fact, all of the columns for these polynomial terms are correlated.

```
> cor( cbind( x, x2, x3, x4 ) )
      x       x2       x3       x4
x  1.0000000 0.3511690 0.6799941 0.3112463
x2 0.3511690 1.0000000 0.5580590 0.8695895
x3 0.6799941 0.5580590 1.0000000 0.5573948
x4 0.3112463 0.8695895 0.5573948 1.0000000
```

This often happens with polynomial models. The variables X, X^2, X^3, \dots tend to be correlated. These correlations can be especially large when X is entirely positive. If we fit a k th degree polynomial, then we can test the null hypothesis that the highest-order term, X^k , is unnecessary by looking at the t-statistic for that term. But in general we cannot tell whether the lower-order terms are needed by looking at the t-statistics because the polynomial terms are intercorrelated.

When investigating polynomial models, it's often helpful to create the columns $X^0 = 1, X, X^2, X^3, \dots$ and then orthogonalize them using the QR decomposition. The orthogonal basis leads to the same fitted values for Y , but

the t-statistics become more meaningful. The coefficient of the orthogonalized version of X^j becomes the “pure effect” of the j th degree polynomial, regardless of the higher-order terms. Let’s fit a 5th-degree polynomial using an orthogonal basis.

```
> xmat <- cbind( 1, x, x^2, x^3, x^4, x^5)
> tmp <- qr(xmat)
> qmat <- qr.Q( tmp )
> ls.print( lsfit( qmat[,-1], y ) )
Residual Standard Error=0.2098
R-Square=0.0058
F-statistic (df=5, 1829)=2.1182
p-value=0.0606

      Estimate Std.Err t-value Pr(>|t|)
Intercept -0.0011  0.0049 -0.2222  0.8242
X1         0.3647  0.2098  1.7383  0.0823
X2        -0.5361  0.2098 -2.5552  0.0107
X3        -0.1805  0.2098 -0.8602  0.3898
X4        -0.1149  0.2098 -0.5478  0.5839
X5        -0.0033  0.2098 -0.0155  0.9876
```

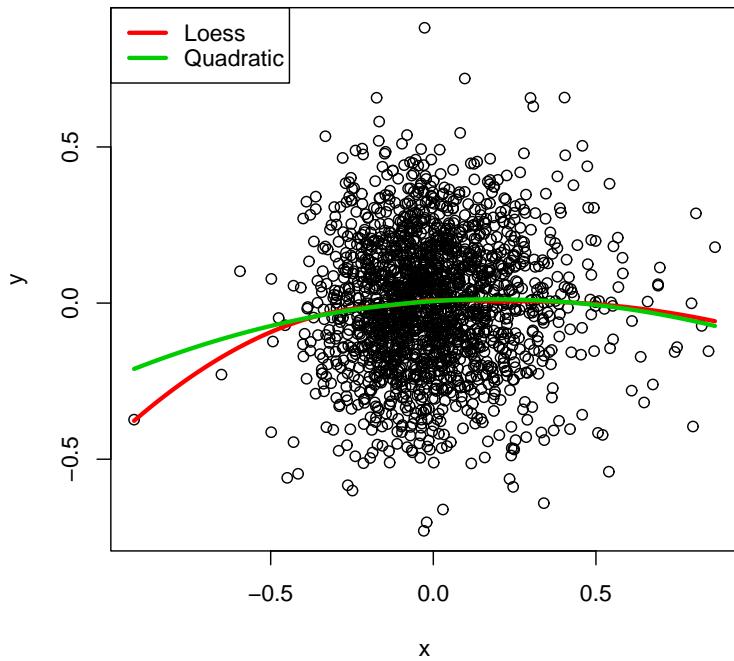
Notice that we explicitly included a column of ones in the original design matrix, but after orthogonalizing it, we removed the first column, allowing `lsfit` to add an intercept term automatically. Looking at the coefficients for the orthogonalized polynomial terms, it is now obvious that a quadratic model fits best.

Now let’s omit the higher-order terms, re-fit the quadratic model, and visually compare the fit of the quadratic model to the loess curve.

```

> plot( x, y )
> loess.fit <- loess( y ~ x, data=data.frame(x=x,y=y) )
> x.grid <- seq( from=min(x), to=max(x), length=200)
> tmp <- predict( loess.fit, newdata=data.frame(x=x.grid), se=T)
> lines( x.grid, tmp$fit, lwd=3, col=2 )
> quadratic.fit <- lm( y ~ x + x2, data=data.frame(x=x, x2=x^2, y=y) )
> tmp <- predict( quadratic.fit, newdata=data.frame(x=x.grid, x2=x.grid^2), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=3)
> legend(x="topleft", c("Loess", "Quadratic"), lwd=c(3,3), col=c(2,3) )

```



Notice that, when fitting the quadratic model, I went back to the original (non-orthogonalized) basis. (I could have done it either way; the fitted values would be the same.) Notice also that I applied the function `predict` to the result of `lm`. The R function `predict` is a “generic function” that can be used in a wide variety of modeling procedures.

Examining the plot, we see that the predictions from loess and the quadratic model are nearly indistinguishable except at the far left side, where there are virtually no observations. The quadratic model fits extremely well.

One word of caution: When fitting a k th-degree polynomial, you may find that the coefficients for some of the lower-order terms $X^j, j < k$ are not significantly different from zero, and it may be tempting to remove them from the model. Do not remove the lower-order terms. If you don't include all the lower-order terms, then the model is no longer invariant to transformations of the columns. If you fit a model in which some of the lower-order terms are missing, then the meaning of the model will change if X is replaced by $a + bX$ for some constants a and b . In general, you should obey the following **hierarchy principle**: If the model includes X^k , then it should also include each X^j for $j < k$, whether or not the coefficients for these lower-order terms are significant. If you obey the hierarchy principle, then your model will be invariant (i.e. give the same fit) under linear transformations of X .

SPLINES AND SINUSOIDS

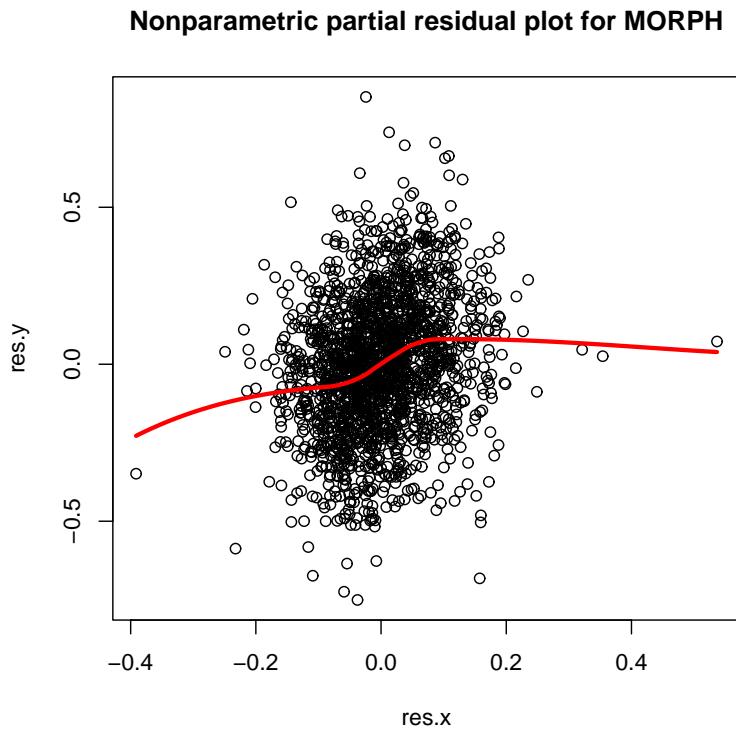
When polynomials do not work. In the last lecture, we showed how to account for a nonlinear trend between a response Y and a single predictor X by fitting a k th degree polynomial model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_k X^k + \text{error}.$$

We applied the method to our body measurements data and found that the relationship between $\log(\text{CHOL})$ and $\log(\text{MORPH})$ was well approximated by a quadratic ($k = 2$) function. But there are many examples for which polynomials do not work well.

For example, let's look at the relationship between $\log(\text{CHOL})$ and MORPH , partialling out $\log(\text{BMI})$ in a nonparametric fashion.

```
> res.y <- loess( log.chol ~ log.bmi, data=body)$res
> res.x <- loess( morph ~ log.bmi, data=body)$res
> plot(res.x, res.y, main="Nonparametric partial residual plot for MORPH")
> loess.fit <- loess( y ~ x, data=data.frame(x=res.x,y=res.y))
> x.grid <- seq( from=min(res.x), to=max(res.x), length=200)
> tmp <- predict( loess.fit, newdata=data.frame( x=x.grid), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=2)
```



Now let's see how well this trend is described by polynomials. On the next page, I show plots of the k th degree polynomial fit for $k = 1, 2, \dots, 6$. R code for the linear fit is shown below.

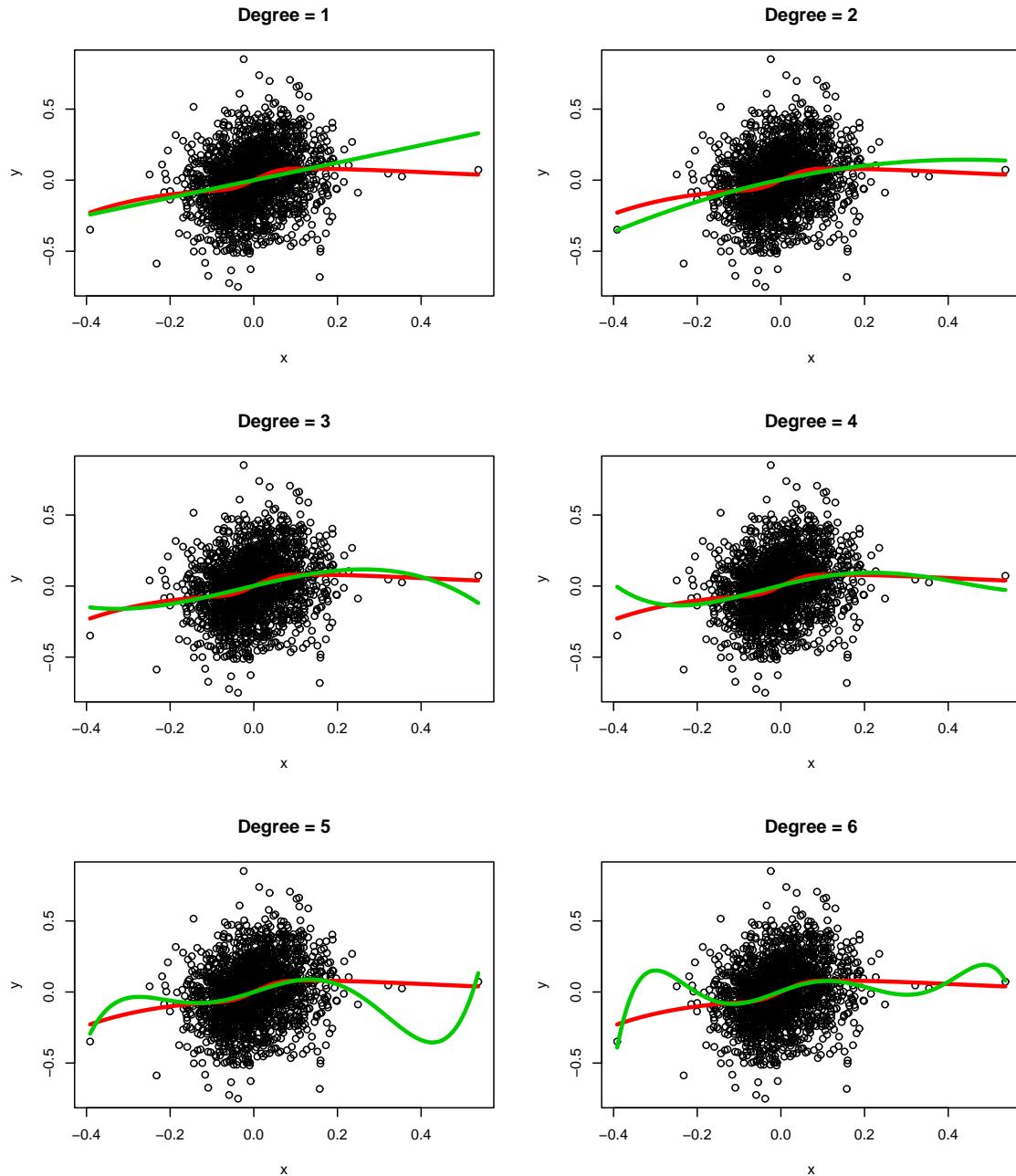
```

> x <- res.x
> y <- res.y
> x.grid <- seq( from=min(x), to=max(x), length=200 )

> par(mfrow=c(3,2))
> plot(x, y, main="Degree = 1")
> tmp <- predict( loess.fit, newdata=data.frame( x=x.grid ), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=2)

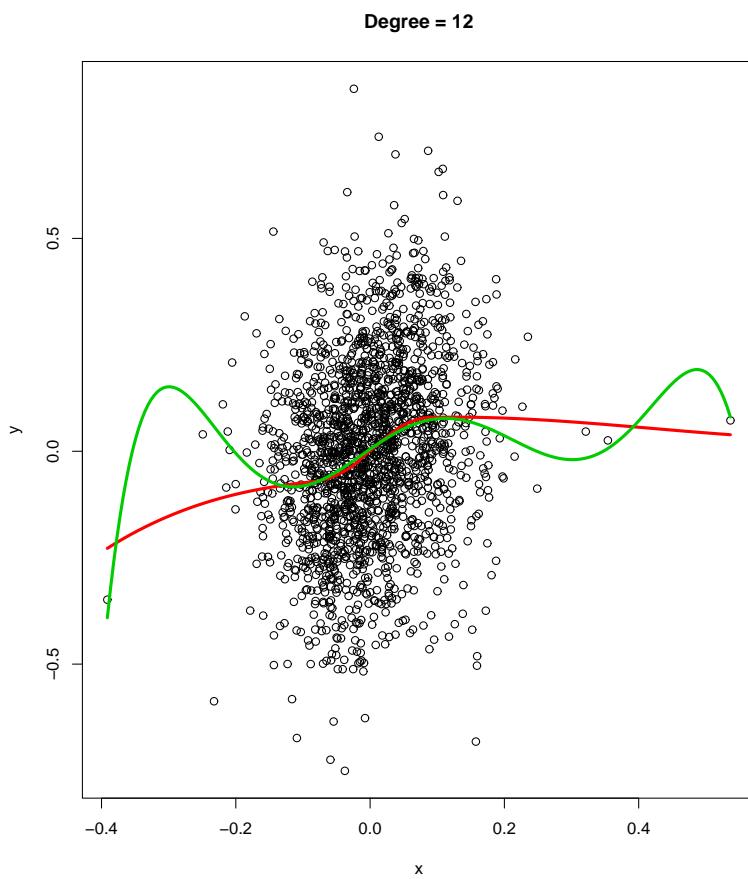
> linear.fit <- lm( y ~ x, data=data.frame( x=x, y=y ) )
> tmp <- predict( linear.fit, newdata=data.frame( x=x.grid ), se=T )
> lines( x.grid, tmp$fit, lwd=3, col=3)

```



The quadratic ($k = 2$) and cubic ($k = 3$) models seem to fit pretty well overall, but neither one captures the S-shaped feature in the middle of the data. Increasing the degree to 4, 5 or 6 begins to capture that feature better but introduces wild fluctuations at the extremes.

This is a common problem with polynomials. By increasing the degree of the polynomial, we can eventually approximate the trends very well in the middle of the data, but the predictions at the ends become increasingly erratic. The next plot shows the fit of a 12th-degree polynomial.



Polynomials sometimes work well, but they are often “too wiggly” to describe erratic trends that we see in real data. When polynomials don’t get the job done, it may be helpful to turn to splines.

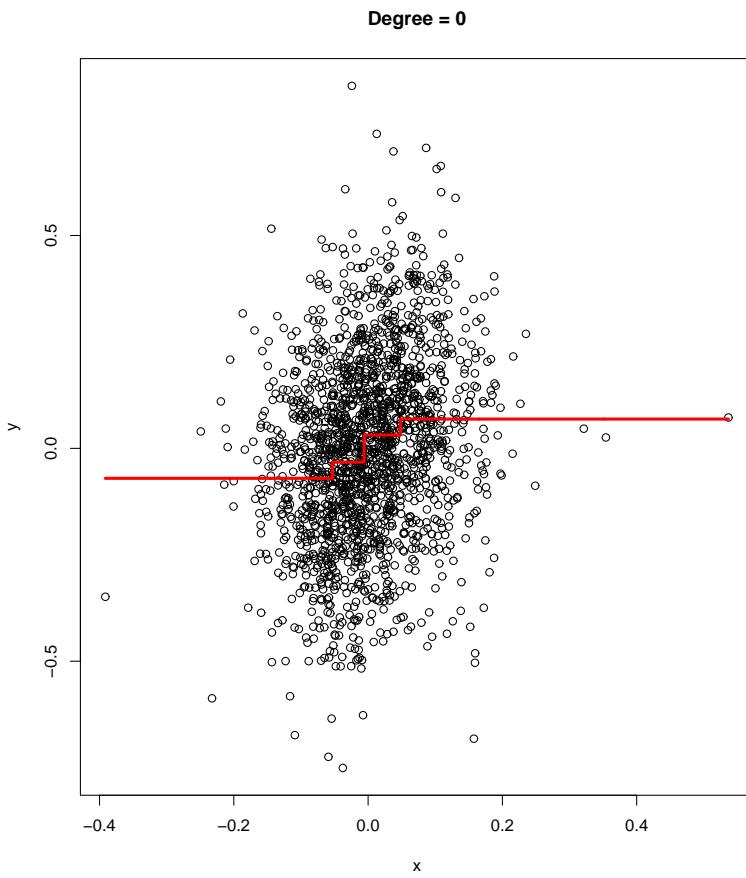
Splines. The basic idea of splines is to approximate a curve by subdividing the range of the data into intervals separated by a set of points (called “knots”), and fitting polynomials in each interval, constraining these polynomials to agree at the knots.

Formally, a k th degree spline with knots $\xi_1, \xi_2, \dots, \xi_Q$ is a function that is a k th degree polynomial within each interval

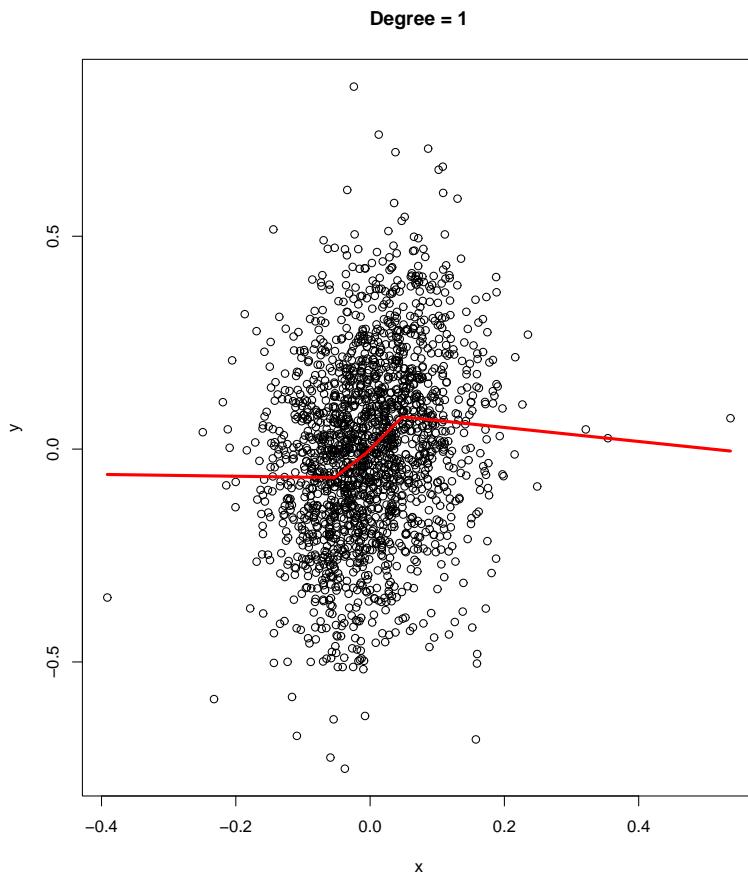
$$(-\infty, \xi_1], (\xi_1, \xi_2], \dots, (\xi_{Q-1}, \xi_Q], (\xi_Q, \infty)$$

with $k - 1$ continuous derivatives. That is, the function itself (if $k > 0$) and its first $k - 1$ derivatives must agree at the knots.

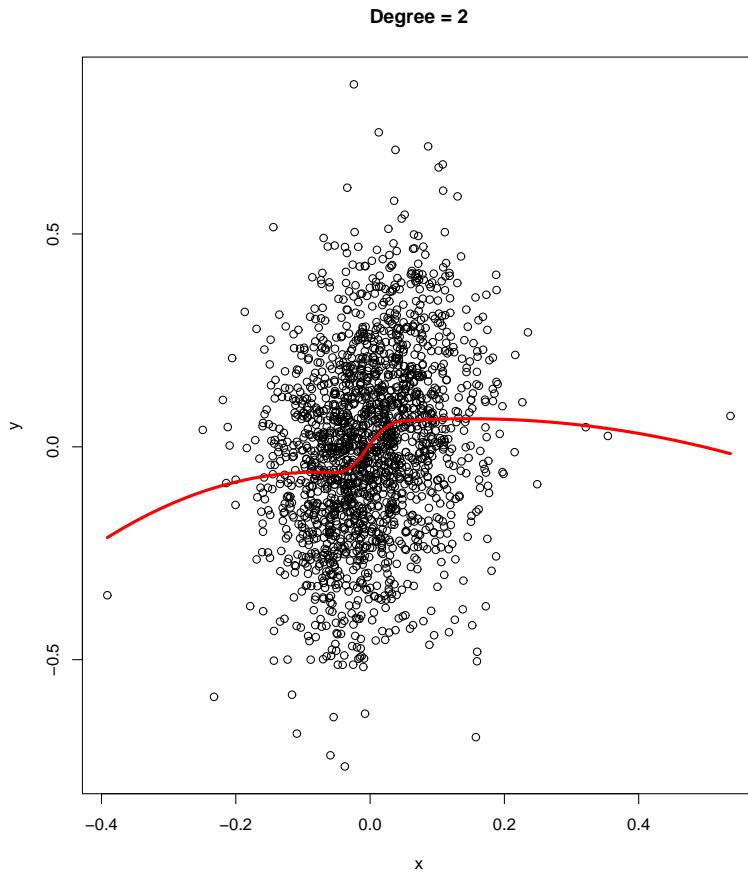
When $k = 0$, the spline is a piecewise constant function with discontinuities at the knots.



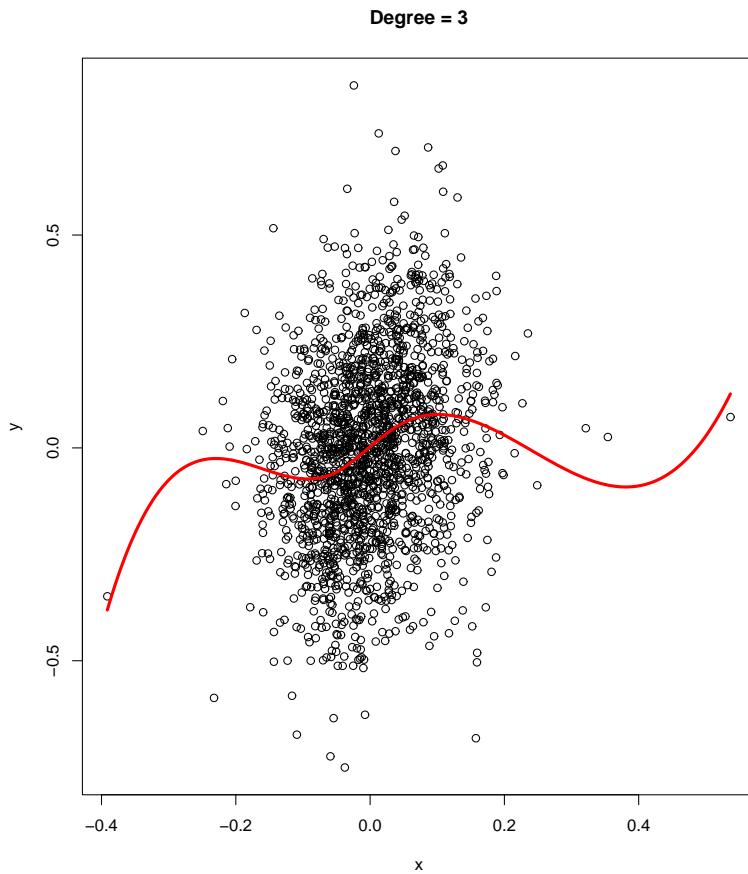
When $k = 1$, the spline is a piecewise linear function. The function is continuous, but the slope may change abruptly at the knots.



When $k = 2$, the spline is a piecewise quadratic function. The slope (first derivative) is continuous at the knots, which gives it a nice smooth appearance.



When $k = 3$, the spline looks even smoother.



The truncated power basis. How do you fit a spline to a set of data to describe the regression of Y on X ? The first thing that you need to do is decide

- the degree of the spline,
- how many knots there will be, and
- where the knots will be placed.

Piecewise constant ($k = 0$) splines are unappealing because they are discontinuous. Piecewise linear ($k = 1$) splines are better, but most users prefer quadratic ($k = 2$) or cubic ($k = 3$) splines because they are more aesthetically

pleasing. There is almost never a good reason to use $k > 3$.

Given the number of knots Q , it is customary to set these at the sample quantiles of X so that an (essentially) equal number of observations fall into each interval. With $Q = 3$ knots, for example, we would set ξ_1 , ξ_2 and ξ_3 equal to the 25th, 50th and 75th percentiles of X .

The next thing you need to do is create a set of basis functions. That is, you need to create a design matrix whose columns span the space of all splines with the desired degree and choice of knots. There are many different ways to do this. The simplest method is to create columns with $X^0 = 1, X, X^2, \dots, X^k$ as in an ordinary k th degree polynomial. Then, for each knot, you would create an additional column equal to

$$[\max(0, (X - \xi_q))]^k.$$

In other words, for each knot ξ_q , create a variable equal $(X - \xi_q)^k$, but then set that variable to zero whenever $X < \xi_q$. Statisticians sometimes write this as

$$(X - \xi_q)_+^k,$$

where the subscript “+” indicates that the value is set to zero whenever the quantity inside the parentheses is negative.

The set of functions

$$1, X, \dots, X^k, (X - \xi_1)_+^k, \dots, (X - \xi_Q)_+^k$$

is called the **truncated power basis**. The set of all functions

$$\begin{aligned} f(X) = & \beta_0 + \beta_1 X + \cdots + \beta_k X^k \\ & + \beta_{k+1} (X - \xi_1)_+^k + \cdots + \beta_{k+Q} (X - \xi_Q)_+^k \end{aligned}$$

for various choices of $\beta_0, \dots, \beta_{k+Q}$ is the set of k th degree splines with knots at ξ_1, \dots, ξ_Q .

Some textbooks recommend that you do not use the truncated power basis, because it may cause the columns of the design matrix to be highly correlated, possibly leading to numerical instability. But numerical instability is not a problem with modern regression software, and most regression programs will automatically orthogonalize the columns anyway. If the columns are highly correlated, then the coefficients for the basis functions will “share significance.” But when we are fitting a spline model, we rarely want to interpret the coefficients. And even if a coefficient is not significantly different from zero, we would not want to remove that term from the model, because then the remaining terms will no longer span the space of splines.

The choice of k and Q is somewhat subjective, and there is always a tradeoff between model fit and complexity. To fit a k th degree spline with Q knots, we need $p = k + 1 + Q$ parameters. We can always get a better fit to the current data (i.e., a higher R^2) by adding more parameters to the model, but higher R^2 does not mean that the model is

better. In the most extreme case, we can fit a model with $p = n$ parameters which will fit the observed data perfectly ($R^2 = 1$), but will not give reasonable predictions for future data. In another lecture, we will explore this tradeoff between model fit and complexity, and we will describe various criteria for choosing among different models. For now, we will simply note that one popular criterion is the PRESS (prediction sum of squares) statistic, defined as the sum of the squared PRESS residuals,

$$\text{PRESS} = \sum_{i=1}^n \hat{\epsilon}_{(i)}^2.$$

Models with lower values of PRESS are better in the sense that they appear to give more precise predictions for unseen future observations.

Example. Let's return to our body measurements data and regress the nonparametric residuals from $\log(\text{CHOL})$ on the nonparametric residuals from MORPH. We will try spline fits with various degrees and number of knots.

To make the programming less tedious, I wrote an R function that creates a truncated power basis. The user supplies three arguments: the variable X , the degree k , and the number of knots Q . The function returns a design matrix whose columns are the basis functions. The knots are positioned at the quantiles of X .

```

spline.basis <- function(x, degree, knots){
  n <- length(x)
  k <- degree
  basis <- matrix( 1, n, 1)
  for( i in 1:k){
    basis <- cbind( basis, x^i )}
  if( knots>0 ){
    xi <- quantile(x, (1:knots)/(knots+1) )
    for( i in 1:knots){
      tmp <- (x-xi[i])
      tmp[tmp<0] <- 0
      basis <- cbind( basis, tmp^k )} }
  basis}

```

Now let's try eighteen different spline models. We will create a table that records, for each model,

- the degree k and number of knots Q ,
- the total number of parameters p ,
- the R^2 , and
- the PRESS statistic.

```

> y <- loess( log.chol ~ log.bmi, data=body)$res
> x <- loess( morph ~ log.bmi, data=body)$res

> # set up vectors to define the various models
> degree <- c( 1, 2, 3, 4, 5, 6, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3)
> knots <- c( 0, 0, 0, 0, 0, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4)

> # set up table to store the results
> nmodel <- length(degree)
> results <- matrix( NA, nmodel, 5)
> dimnames(results) <- list(
+   format(1:nmodel),
+   c("degree","knots","params","r2","PRESS"))

> # fit the models and save results

```

```

> for( i in 1:nmodel){
+   xmat <- spline.basis(x, degree[i], knots[i])
+   tmp <- lsfit( xmat, y, intercept=F)
+   res <- tmp$res
+   fit <- y - res
+   lev <- ls.diag(tmp)$hat
+   results[i,1] <- degree[i]
+   results[i,2] <- knots[i]
+   results[i,3] <- ncol(xmat)
+   results[i,4] <- cor(y,fit)^2
+   results[i,5] <- sum( (res/(1-lev))^2 )}

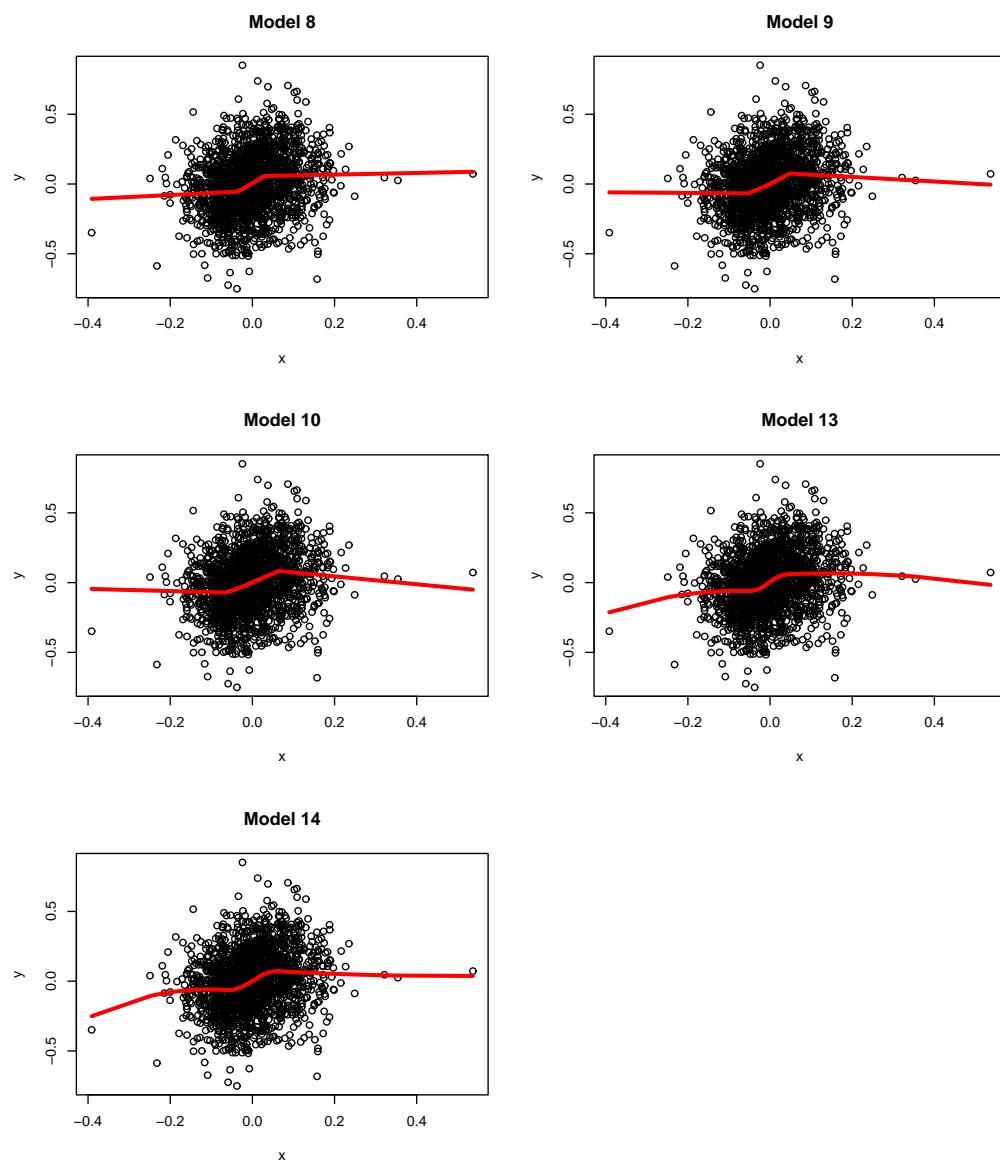
> results
    degree knots params      r2      PRESS
1       1     0      2 0.04928735 81.26950
2       2     0      3 0.05068892 81.19947
3       3     0      4 0.05242907 81.86682
4       4     0      5 0.05303345 87.15888
5       5     0      6 0.05958567 109.63843
6       6     0      7 0.06268712 141.35177
7       1     1      3 0.04950691 81.33769
8       1     2      4 0.06160296 80.37902
9       1     3      5 0.06557909 80.12945
10      1     4      6 0.06529556 80.25869
11      2     1      4 0.05658344 81.34313
12      2     2      5 0.05591352 82.36641
13      2     3      6 0.06326951 80.63424
14      2     4      7 0.06501915 80.40986
15      3     1      5 0.05265172 86.36791
16      3     2      6 0.06523847 81.13966
17      3     3      7 0.06513296 82.73570
18      3     4      8 0.06537238 82.32096

```

The first six models are ordinary polynomials. The quality of the polynomial fit, as measured by PRESS, actually gets worse when we add terms beyond X^2 . Among the spline models, the five best are Models 8 ($k = 1, Q = 2$), 9 ($k = 1, Q = 3$), 10 ($k = 1, Q = 4$), 13 ($k = 2, Q = 3$) and 14 ($k = 1, Q = 4$). To see what these fits look like, here are

plots for the five best.

```
> par(mfrow=c(3,2))
> for( i in c(8,9,10,13,14)){
+   xmat <- spline.basis(x, degree[i], knots[i])
+   tmp <- lsfit( xmat, y, intercept=F)
+   res <- tmp$res
+   fit <- y - res
+   plot(x,y, main=paste("Model", format(i) ))
+   o <- order(x)
+   lines( x[o], fit[o], lwd=3, col=2)}
```



The spline models do a much better job than ordinary polynomials at capturing the trend. For example, Model 8 has the same number of parameters ($p = 4$) as a cubic model, but it gives a better approximation to the data both in the middle and at the extremes.

Now let's turn our attention to the other predictor,

$\log(\text{BMI})$. Last time, we found that a quadratic model seemed to fit well. Let's look at a variety of spline fits and see what happens.

```

> y <- loess( log.chol ~ morph, data=body)$res
> x <- loess( log.bmi ~ morph, data=body)$res
> degree <- c( 1, 2, 3, 4, 5, 6, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3)
> knots <- c( 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4)
> nmodel <- length(degree)
> results <- matrix( NA, nmodel, 5)
> dimnames(results) <- list(
+   format(1:nmodel),
+   c("degree", "knots", "params", "r2", "PRESS"))
> for( i in 1:nmodel){
+   xmat <- spline.basis(x, degree[i], knots[i])
+   tmp <- lsfit( xmat, y, intercept=F)
+   res <- tmp$res
+   fit <- y - res
+   lev <- ls.diag(tmp)$hat
+   results[i,1] <- degree[i]
+   results[i,2] <- knots[i]
+   results[i,3] <- ncol(xmat)
+   results[i,4] <- cor(y,fit)^2
+   results[i,5] <- sum( ( res/(1-lev) )^2 )}
> results
    degree knots params      r2      PRESS
 1       1     0      2 0.001642548 81.03950
 2       2     0      3 0.005191870 80.83557
 3       3     0      4 0.005594075 80.86049
 4       4     0      5 0.005757218 80.92002
 5       5     0      6 0.005757349 81.11920
 6       6     0      7 0.006018007 81.98569
 7       1     1      3 0.004496107 80.90171
 8       1     2      4 0.004532545 80.97962
 9       1     3      5 0.004548333 81.07735
10      1     4      6 0.004865265 81.13359
11      2     1      4 0.005612089 80.86233
12      2     2      5 0.005658747 80.93518
13      2     3      6 0.005776400 80.99697
14      2     4      7 0.006263232 81.07397

```

15	3	1	5	0.005772252	80.91647
16	3	2	6	0.005778859	81.02652
17	3	3	7	0.005839829	81.26583
18	3	4	8	0.005847835	81.32502

In this case, the best model (as judged by PRESS) is Model 2, the ordinary quadratic trend with no knots. This is the model that we chose last time.

Putting it all together. Now let's combine the quadratic terms for $\log(\text{BMI})$ and the spline terms for MORPH into a single model to predict $\log(\text{CHOL})$. We will fit the model, plot the residuals versus the fitted values, and look for any remaining evidence of nonlinearity.

```

> y <- body$log.chol

> # get a basis for morph spline model #9
> basis <- spline.basis( body$morph, degree=1, knots=3)

> # append the linear and quadratic terms for log(bmi)
> xmat <- cbind( basis, body$log.bmi, body$log.bmi^2)

> # give names to the columns
> dimnames(xmat) <- list(NULL,
+   c("Constant", "MORPH.1", "MORPH.2", "MORPH.3", "MORPH.4",
+     "log(BMI)", "log(BMI)^2") )

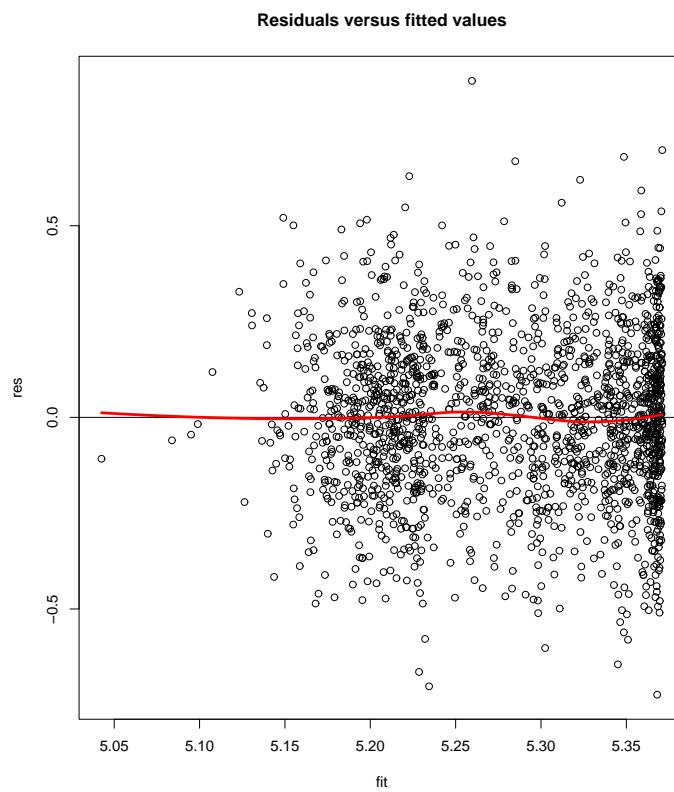
> # fit the model
> result <- lsfit( xmat, y, intercept=F)

> # plot the residuals versus the fitted values
> res <- result$res
> fit <- y - res
> plot(fit, res, main="Residuals versus fitted values")

> # add zero line and loess fit

```

```
> abline(h=0)
> loess.fit <- loess( y ~ x, data=data.frame(x=fit, y=res) )
> x.grid <- seq( from=min(fit), to=max(fit), length=200)
> tmp <- predict( loess.fit, newdata=data.frame(x=x.grid), se=T)
> lines( x.grid, tmp$fit, lwd=3, col=2)
```



Nice!

Periodic trends. Sometimes we encounter data with periodic trends. A function f is said to be periodic with period P if

$$f(x + P) = f(x)$$

for all x . Periodic trends are often found in time-series data. The simplest periodic function is a sinusoidal curve, which can be written as

$$f(x) = \beta_0 + \beta_1 \sin(2\pi x/P).$$

This function takes the value β_0 at $x = 0$, rises to $\beta_0 + \beta_1$ at $x = P/4$, returns to β_0 at $x = P/2$, drops to $\beta_0 - \beta_1$ at $x = 3P/4$, and returns to β_0 at $x = P$.

If we happen to know the position where the phase begins (i.e., if we know x_0 for which $f(x_0) \approx \beta_0$), then we can write the sinusoidal curve as

$$f(x) = \beta_0 + \beta_1 \sin(2\pi(x - x_0)/P). \quad (1)$$

That is, we can regress the response variable on a constant and on the variable $\sin(2\pi(x - x_0)/P)$. If the phase is unknown, we can estimate the shift parameter by including a cosine term in the model,

$$f(x) = \beta_0 + \beta_1 \sin(2\pi x/P) + \beta_2 \cos(2\pi x/P). \quad (2)$$

Because of the trigonometric identity

$$\beta_1 \sin(\theta) + \beta_2 \cos(\theta) = \sqrt{\beta_1^2 + \beta_2^2} \sin\left(\theta + \tan^{-1}\frac{\beta_2}{\beta_1}\right),$$

it follows that (2) is equivalent to (1) with

$$-\frac{2\pi x_0}{P} = \tan^{-1} \frac{\beta_2}{\beta_1},$$

or

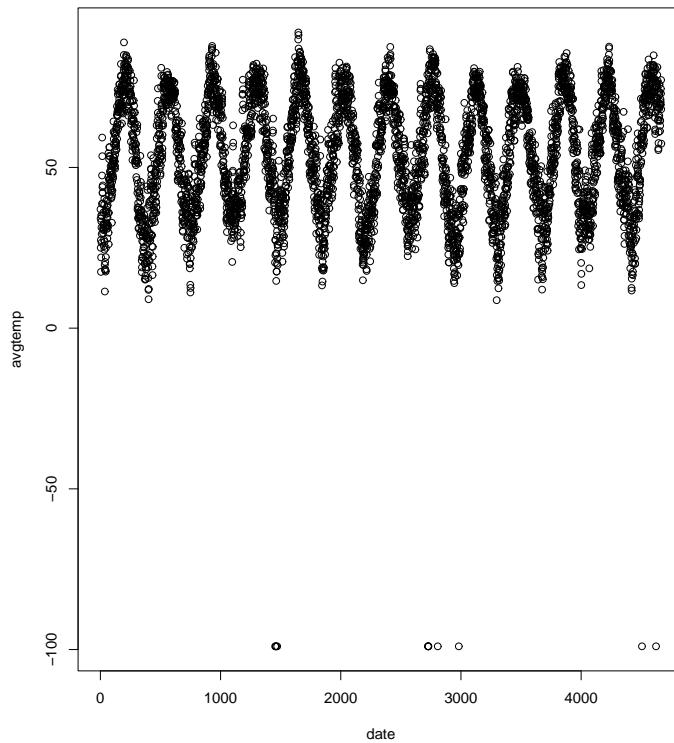
$$x_0 = - \left(\frac{P}{2\pi} \right) \tan^{-1} \frac{\beta_2}{\beta_1}.$$

Example. The University of Dayton maintains an archive of daily temperature readings for various locations. The file PAHARRIS.txt contains the average daily temperatures for Harrisburg, Pennsylvania since January 1, 1995.

1	1	1995	37.0
1	2	1995	34.1
1	3	1995	25.0
1	4	1995	27.4
1	5	1995	17.5
 -- a few thousand lines omitted --			
10	8	2007	73.0
10	9	2007	77.2
10	10	2007	68.1
10	11	2007	57.4

Let's read in the data and plot them.

```
> harrisburg <- read.table("PAHARRIS.txt")
> month <- harrisburg[,1]
> day   <- harrisburg[,2]
> year  <- harrisburg[,3]
> avgtemp <- harrisburg[,4] # fourth column is the temperature
> n <- length(avgtemp)
> date <- 1:n
> plot( date, avgtemp)
```



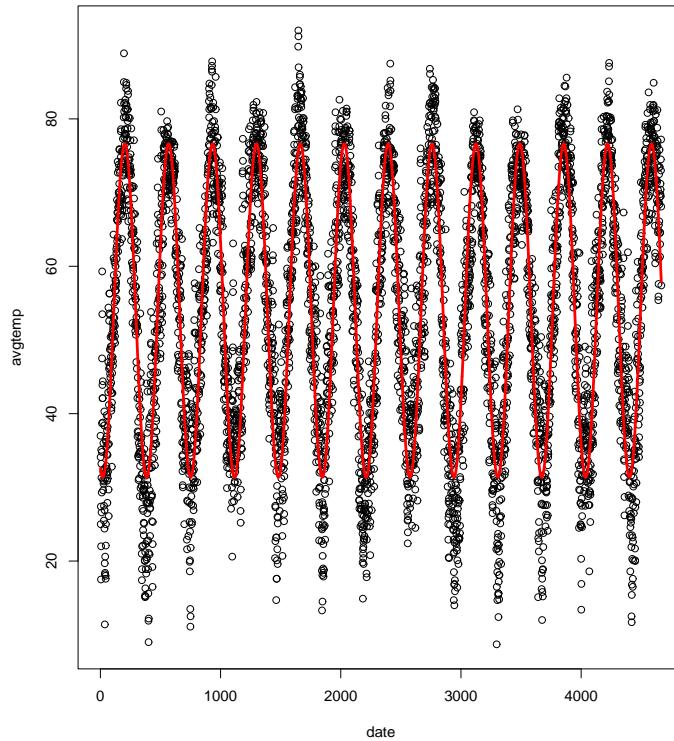
The extreme values at the bottom of the plot are missing values coded as `-99`. Let's remove these non-observations:

```
> w <- (avgtemp == -99)
> date <- date[!w]
> avgtemp <- avgtemp[!w]
```

Let's describe these data by a sinusoid with a period of $P = 365.24$ and estimate the phase shift.

```
> period <- 365.24
> x1 <- sin(2*pi*date/period)
> x2 <- cos(2*pi*date/period)
> result <- lm( avgtemp ~ x1 + x2 )
> summary(result)
```

```
Call:  
lm(formula = avgtemp ~ x1 + x2)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-23.1486 -4.6949 -0.0768  4.4645 31.7899  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) 53.9950     0.1040 519.08 <2e-16 ***  
x1          -7.7467     0.1468 -52.76 <2e-16 ***  
x2         -21.2763     0.1474 -144.38 <2e-16 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 7.095 on 4652 degrees of freedom  
Multiple R-Squared: 0.8366,      Adjusted R-squared: 0.8365  
F-statistic: 1.191e+04 on 2 and 4652 DF, p-value: < 2.2e-16  
  
> plot(date, avgtemp)  
> res <- result$res  
> fit <- avgtemp - res  
> lines( date, fit, lwd=3, col=2)
```



The sinusoid captures the trend well, but the data appear to be heteroscedastic; the error variance seems larger during periods of extreme heat and cold.

The estimate of the phase shift is:

```
> beta.1 <- result$coef[2]
> beta.2 <- result$coef[3]
> shift <- - (period/(2*pi)) * atan(beta.2/beta.1)
> shift
      x2
-71.01223
```

The phase appears to begin about 71 days before Day 0 (December 31), which is the $365 - 71 = 294$ th day of the year (October 21). That is the day on which the predicted

temperature is equal to the yearly average. The temperature is also close to the yearly average on day

$$P + \text{shift} - (P/2) = 111.6 \approx 112,$$

which is April 22. The warmest day of the year is

$$P + \text{shift} - (P/4) = 202.9 \approx 203,$$

which corresponds to July 22, and the coldest day of the year is

$$P + \text{shift} - (3P/4) = 20.3 \approx 20$$

or January 20.

In addition to periodic trends, the data may also exhibit gradual drift. Has there been a gradual rise or fall over this twelve-year period? We can look for a linear trend by simply adding date to the regression.

```
> result <- lm( avgtemp ~ x1 + x2 + date )
> summary(result)

Call:
lm(formula = avgtemp ~ x1 + x2 + date)

Residuals:
    Min      1Q   Median      3Q      Max 
-23.06978 -4.71444 -0.08613  4.48360 31.84439 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 5.388e+01  2.081e-01 258.901  <2e-16 ***
x1          -7.743e+00  1.469e-01 -52.695  <2e-16 ***
x2          -2.127e+01  1.474e-01 -144.293 <2e-16 ***
date        5.064e-05  7.724e-05   0.656    0.512  
---

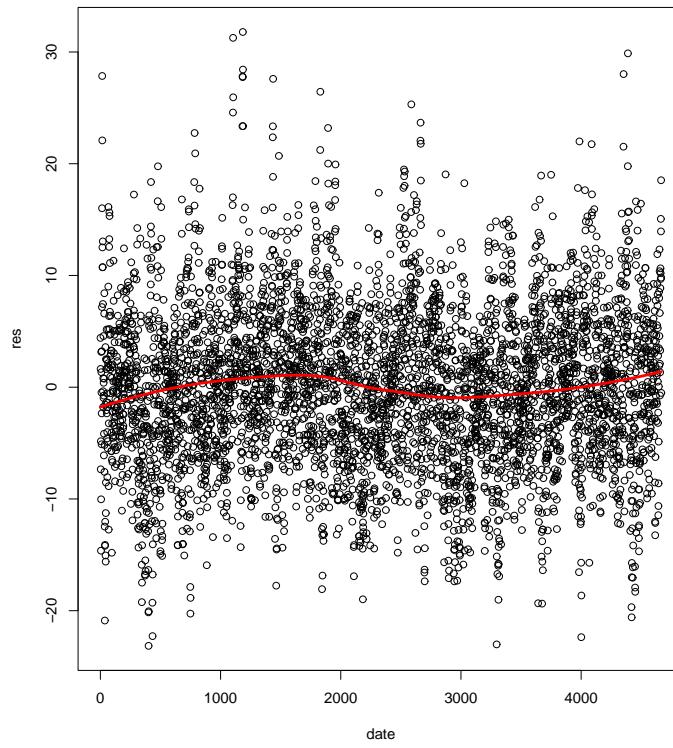
```

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

```
Residual standard error: 7.096 on 4651 degrees of freedom
Multiple R-Squared: 0.8366,      Adjusted R-squared: 0.8365
F-statistic: 7939 on 3 and 4651 DF, p-value: < 2.2e-16
```

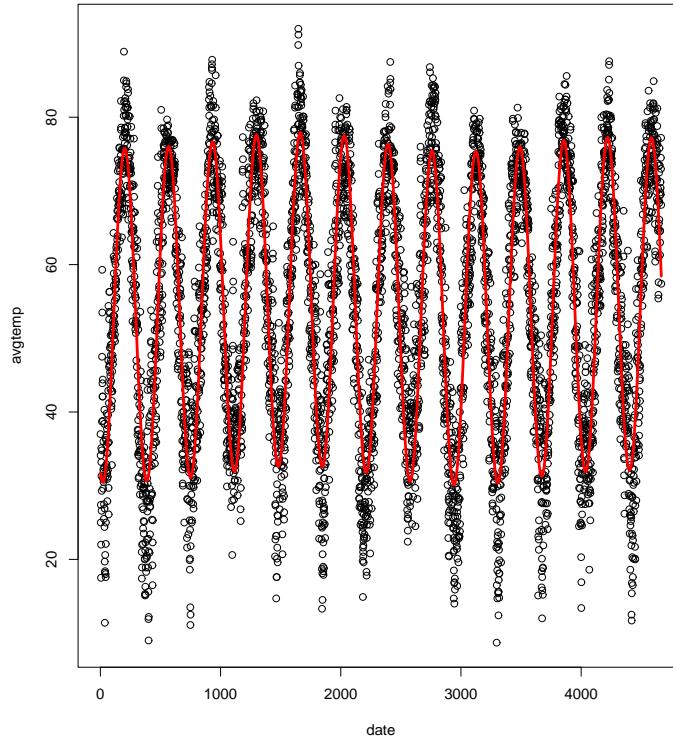
There is no significant increasing or decreasing trend over the twelve years. But there may be other long-term trends. Let's look for trends by plotting the residuals from the sinusoidal fit versus date and adding a loess curve.

```
> result <- lm( avgtemp ~ x1 + x2)
> res <- result$res
> plot( date, res )
> loess.fit <- loess( y ~ x, data=data.frame(x=date, y=res) )
> tmp <- predict( loess.fit, newdata=data.frame(x=date), se=T)
> lines( date, tmp$fit, lwd=3, col=2)
```



There is a noticeable trend, but it's not linear. Let's describe that trend by a spline. I'll use a quadratic spline with 3 knots, which adds five coefficients to the model.

```
> xmat <- spline.basis( date, degree=2, knots=3)
> xmat <- cbind( xmat, x1, x2 )
> dimnames(xmat) <- list( NULL,
+   c("Constant","spline.1","spline.2","spline.3","spline.4",
+     "spline.5","sin","cos"))
> result <- lsfit( xmat, avgtemp, intercept=F)
> plot(date,avgtemp)
> fit <- xmat %*% result$coef
> lines( date, fit, lwd=3, col=2)
```



The fitted curve now shows a combination of yearly cycles and long-term trends. Has the addition of these extra terms changed our estimate of where the cycle begins?

```
> beta.1 <- result$coef["sin"]
> beta.2 <- result$coef["cos"]
> shift <- - (period/(2*pi)) * atan( beta.2/beta.1)
> shift
-71.05171
```

No, the shift parameter has barely changed. Let's finish up this analysis by checking to see whether the long-term trend is significant. To do this, we will perform a partial F-test for the five spline terms.

```
> # fit full model including spline terms
> result <- lsfit( xmat, avgtemp, intercept=F)
> SSErr.full <- sum( result$res^2 )
> df <- nrow(xmat) - ncol(xmat) # error degrees of freedom

> # fit reduced model without spline terms
> result <- lm( avgtemp ~ x1 + x2 )
> SSErr.reduced <- sum( result$res^2 )

> # the SSReg for the spline terms is equal to the change in SSErr
> F <- ( (SSErr.reduced - SSErr.full) / 5 ) / (SSErr.full / df)
> F
[1] 12.54601
> 1 - pf(F, 5, df)
[1] 4.005907e-12
```

Yes, the spline terms are highly significant. We have detected a real trend, but it's not linear.

CODING SCHEMES FOR CATEGORICAL PREDICTORS

Single predictor with two levels. Consider a binary predictor (e.g., sex). Suppose we define two dummy variables

$$X_1 = \begin{cases} 1 & \text{if sex=M,} \\ 0 & \text{if sex=F,} \end{cases} \quad \text{and} \quad X_2 = \begin{cases} 0 & \text{if sex=M,} \\ 1 & \text{if sex=F.} \end{cases}$$

If we try to fit the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \text{error},$$

the design matrix will have a linear dependency, because $X_1 + X_2 = 1$. What can we do?

One possibility is to **omit the intercept**, so that the model becomes

$$Y = \beta_1 X_1 + \beta_2 X_2 + \text{error.} \quad (1)$$

Under this model, the remaining coefficients have a simple interpretation,

$$\begin{aligned} \beta_1 &= \mu_M = E(Y) \text{ for males,} \\ \beta_2 &= \mu_F = E(Y) \text{ for females.} \end{aligned}$$

But analysts rarely do this, because the t-statistics in the table of coefficients will be testing the null hypotheses $H_0 : \beta_1 = 0$ and $H_0 : \beta_2 = 0$, which are rarely of interest. The hypothesis that usually is of interest, $H_0 : \beta_1 = \beta_2$, is not immediately testable from the computer printout.

A more common approach is to **omit one of the dummy indicators**. If we omit X_2 , then the model becomes

$$Y = \beta_0 + \beta_1 X_1 + \text{error.} \quad (2)$$

This model implies that

$$\mu_M = \beta_0 + \beta_1,$$

$$\mu_F = \beta_0,$$

which implies that

$$\beta_1 = \mu_M - \mu_F.$$

Then the null hypothesis of interest, $\mu_M = \mu_F$, becomes $H_0 : \beta_1 = 0$. By omitting the dummy indicator for females, we have made them the “reference group.” The intercept is the mean response for the reference group. What happens if we omit X_1 from the model as well? If we omit X_1 , then the intercept β_0 becomes **the mean response for the overall population of males and females from which the sample was drawn**.

Another common approach for binary predictors uses a

different coding scheme. Suppose we define

$$X = \begin{cases} -1 & \text{for males,} \\ +1 & \text{for females.} \end{cases}$$

In common terminology, this is called an “effect code” to distinguish it from a dummy (0/1) code. Suppose we fit the model

$$Y = \beta_0 + \beta_1 X + \text{error.} \quad (3)$$

Then $\mu_M = \beta_0 - \beta_1$ and $\mu_F = \beta_0 + \beta_1$, which leads to

$$\begin{aligned} \beta_0 &= \frac{\mu_M + \mu_F}{2}, \\ \beta_1 &= \frac{\mu_F - \mu_M}{2}. \end{aligned}$$

In this new coding scheme, β_0 is **the mean response in a population equally split among males and females**, and β_1 is **one-half of the difference between the group means**. The value of β_1 in this model is half the size of β_1 in the previous model, but the t-statistic for testing $\beta_1 = 0$ will have the same value as the t-statistic for β_1 in model (2), because they are testing the same null hypothesis.

Once again, if we omit the predictor from model (3), then the meaning of β_0 changes to the mean response for the overall population of males and females from which the sample was drawn. If the data are “balanced” in the sense that there are equal numbers of males and females, then

the two columns of the design matrix (the constant and X) will be orthogonal, and the estimated intercept $\hat{\beta}_0$ will be the same whether or not X is included in the model. If the data are unbalanced, then $\hat{\beta}_0$ will change if X is included or excluded.

We have discussed three methods for including a binary indicator:

- no intercept and two dummy codes,
- intercept and one dummy code,
- intercept and one effect code.

These three methods will give the same fitted values, because the three design matrices span the same space. They are three different parameterizations of the same linear model. Given the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ from one model, we can transform them to get the estimates for the other models.

Categorical predictor with three or more levels.

Now consider a variable that has three categories. For concreteness, we will call it “political affiliation” and suppose that the levels are “Republican”, “Democrat” and “Independent.” Assuming that the model will contain an intercept, we can account for political affiliation by including two coded variables to distinguish among the three groups.

Dummy codes could be defined as

$$X_1 = \begin{cases} 1 & \text{if Republican,} \\ 0 & \text{otherwise,} \end{cases} \quad X_2 = \begin{cases} 1 & \text{if Democrat,} \\ 0 & \text{otherwise.} \end{cases}$$

Denote the mean responses for Republicans, Democrats and Independents by μ_R , μ_D and μ_I . Under the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \text{error}, \quad (4)$$

the means for the three groups will be

$$\begin{aligned} \mu_R &= \beta_0 + \beta_1, \\ \mu_D &= \beta_0 + \beta_2, \\ \mu_I &= \beta_0. \end{aligned}$$

The intercept, β_0 , is the mean response for the reference group, the group for which no dummy indicator was included (Independent). The other coefficients are

$$\begin{aligned} \beta_1 &= \mu_R - \mu_I, \\ \beta_2 &= \mu_D - \mu_I, \end{aligned}$$

the deviations of the other groups from the reference group. The reference group becomes the baseline against which the means of the other two groups are measured.

Note that

- $\beta_1 = 0$ corresponds to $H_0 : \mu_R = \mu_I$, and
- $\beta_2 = 0$ corresponds to $H_0 : \mu_D = \mu_I$.

The composite null hypothesis $H_0 : \beta_1 = \beta_2 = 0$, which means that political affiliation has no effect whatsoever ($H_0 : \mu_R = \mu_D = \mu_I$), is an F-test with two numerator degrees of freedom.

Under this coding scheme, the dummy indicators X_1 and X_2 are correlated. Removing X_2 from the model will change both the meaning and the estimates of β_0 and β_1 . If we remove X_2 , then β_0 becomes the mean response for non-Republicans (i.e., Democrats and Independents) in the population from which the sample was taken, and β_1 becomes the difference between Republicans and non-Republicans.

Another way is to build this model is with effect codes,

$$X_1 = \begin{cases} 1 & \text{if Republican,} \\ 0 & \text{if Democrat,} \\ -1 & \text{if Independent,} \end{cases} \quad X_2 = \begin{cases} 0 & \text{if Republican,} \\ 1 & \text{if Democrat,} \\ -1 & \text{if Independent.} \end{cases}$$

Then if we fit the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \text{error}, \quad (5)$$

we get

$$\begin{aligned} \mu_R &= \beta_0 + \beta_1, \\ \mu_D &= \beta_0 + \beta_2, \\ \mu_I &= \beta_0 - \beta_1 - \beta_2. \end{aligned}$$

Solving for the coefficients gives

$$\begin{aligned}\beta_0 &= \frac{\mu_R + \mu_D + \mu_I}{3}, \\ \beta_1 &= \mu_R - \frac{\mu_R + \mu_D + \mu_I}{3}, \\ \beta_2 &= \mu_D - \frac{\mu_R + \mu_D + \mu_I}{3}.\end{aligned}$$

With effect coding, β_0 becomes an equally weighted average of the three group averages. (If the three groups happen to be equally represented in the population, then the intercept is also the overall population average, but this is not usually the case.) The coefficient β_1 , which is the “Republican effect,” is the difference between the mean for Republicans and β_0 . The “Democratic effect,” β_2 , is the difference between the mean for Democrats and β_0 . The “Independent effect,” which could be defined as

$$\beta_3 = \mu_I - \frac{\mu_R + \mu_D + \mu_I}{3},$$

does not appear as a coefficient in the model. But it is easy to see that, if we define β_3 in this manner, then

$$\beta_1 + \beta_2 + \beta_3 = 0,$$

because the deviations of μ_R , μ_D and μ_I around their average must add up to zero. Hence we can get β_3 by

$$\beta_3 = -(\beta_1 + \beta_2).$$

In both dummy coding and effect coding, the null

hypothesis of no effects whatsoever,

$$H_0 : \mu_R + \mu_D + \mu_I$$

will be true if $\beta_1 = \beta_2 = 0$. In either scheme, we can test this null hypothesis by computing the partial F statistic for removing X_1 and X_2 . But the meaning of an individual null hypotheses $H_0 : \beta_1 = 0$ (or $H_0 : \beta_2 = 0$) is very different under the two coding schemes. With dummy coding, $\beta_1 = 0$ means that

$$\mu_R = \mu_I,$$

but with effect coding, $\beta_1 = 0$ means that

$$\mu_R = \frac{\mu_R + \mu_D + \mu_I}{3}.$$

Dummy coding and effect coding extend in the obvious way to categorical variables with any number of levels. Dummy coding for a k -level categorical variable looks like this. (The last category is the reference group.)

group	X_1	X_2	\dots	X_{k-1}
1	1	0	\dots	0
2	0	1	\dots	0
3	0	0	\dots	0
\vdots				
$k - 1$	0	0	\dots	1
k	0	0	\dots	0

If we fit the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \text{error},$$

and if μ_1, \dots, μ_k denote the means for the groups, then

$$\begin{aligned}\beta_0 &= \mu_k, \\ \beta_j &= \mu_j - \mu_k, \quad j \neq k.\end{aligned}$$

Effect coding for k levels looks like this.

group	X_1	X_2	\cdots	X_{k-1}
1	1	0	\cdots	0
2	0	1	\cdots	0
3	0	0	\cdots	0
\vdots	\vdots			
$k-1$	0	0	\cdots	1
k	-1	-1	\cdots	-1

Under effect coding,

$$\begin{aligned}\beta_0 &= \frac{\mu_1 + \mu_2 + \cdots + \mu_k}{k}, \\ \beta_j &= \mu_j - \frac{\mu_1 + \mu_2 + \cdots + \mu_k}{k}, \quad j = 1, \dots, k-1,\end{aligned}$$

and the effect of the k th group is

$$\mu_k - \frac{\mu_1 + \mu_2 + \cdots + \mu_k}{k} = -(\beta_1 + \beta_2 + \cdots + \beta_{k-1}).$$

Interpreting other coding schemes. There are many other coding schemes that could be used. For example, suppose that you have a categorical variables with $k = 4$ predictors and you code the columns this way.

group	X_1	X_2	X_3
1	$3/4$	0	0
2	$-1/4$	$2/3$	0
3	$-1/4$	$-1/3$	$1/2$
4	$-1/4$	$-1/3$	$-1/2$

If you fit the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \text{error}$$

with this coding scheme, how would you interpret the coefficients? Imagine taking the entries in the coding table above, and appending a column of 1's on the left side to form a matrix,

$$A = \begin{bmatrix} 1 & 3/4 & 0 & 0 \\ 1 & -1/4 & 2/3 & 0 \\ 1 & -1/4 & -1/3 & 1/2 \\ 1 & -1/4 & -1/3 & -1/2 \end{bmatrix}.$$

The relationship between the group means and the coefficients is given by

$$\mu = A\beta,$$

where $\mu = (\mu_1, \mu_2, \mu_3, \mu_4)^T$ and $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^T$. The

interpretation of the β 's is given by $\beta = A^{-1}\mu$. Let's compute the inverse of A .

```
> A <- matrix(NA, 4, 4)
> A[,1] <- 1
> A[,2] <- c( 3/4, -1/4, -1/4, -1/4 )
> A[,3] <- c( 0, 2/3, -1/3, -1/3 )
> A[,4] <- c( 0, 0, 1/2, -1/2 )

> solve(A)
      [,1]      [,2]      [,3]      [,4]
[1,] 0.25  0.2500000  0.2500000  0.2500000
[2,] 1.00 -0.3333333 -0.3333333 -0.3333333
[3,] 0.00  1.0000000 -0.5000000 -0.5000000
[4,] 0.00  0.0000000  1.0000000 -1.0000000
```

It follows that

$$\begin{aligned}\beta_0 &= \frac{\mu_1 + \mu_2 + \mu_3 + \mu_4}{4}, \\ \beta_1 &= \mu_1 - \frac{\mu_2 + \mu_3 + \mu_4}{3}, \\ \beta_2 &= \mu_2 - \frac{\mu_3 + \mu_4}{2}, \\ \beta_3 &= \mu_3 - \mu_4.\end{aligned}$$

Another coding scheme that is sometimes used is called “Helmert contrasts.” For $k = 4$ groups, Helmert constraints look like this.

group	X_1	X_2	X_3
1	−1	−1	−1
2	1	−1	−1
3	0	2	−1
4	0	0	3

Under this coding scheme, what do the coefficients mean?

If we append a column of 1's,

$$A = \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 0 & 2 & -1 \\ 1 & 0 & 0 & 3 \end{bmatrix},$$

and invert the matrix, we get

$$A^{-1} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ -1/2 & 1/2 & 0 & 0 \\ -1/6 & -1/6 & 1/3 & 0 \\ -1/12 & -1/12 & -1/12 & 1/4 \end{bmatrix}.$$

So the coefficients are

$$\beta_0 = \frac{\mu_1 + \mu_2 + \mu_3 + \mu_4}{4},$$

$$\beta_1 = \frac{1}{2} (\mu_2 - \mu_1),$$

$$\beta_2 = \frac{1}{3} \left(\mu_3 - \frac{\mu_1 + \mu_2}{2} \right),$$

$$\beta_3 = \frac{1}{4} \left(\mu_4 - \frac{\mu_1 + \mu_2 + \mu_3}{3} \right).$$

Categorical predictors in SAS. When fitting regression models with categorical predictors, you can always create the codes yourself. But many software packages will do it for you. For example, when fitting regression models in SAS, you can declare a variable to be categorical by issuing a **CLASS** statement. By default, SAS will include that variable as a set of $k - 1$ dummy codes, excluding the dummy indicator for highest category (category k) and making it the reference group.

Categorical predictors in R. In R, a categorical variable is called a **factor**. You can turn a variable into a factor using the function **factor**. The variable that you turn into a factor can be numeric or character. For example, let's create a vector of character strings and turn it into a factor.

```
> # create the character vector
> tmp <- c("Dem", "Rep", "Ind", "Rep", "Dem")

> # turn it into a factor
> tmp <- factor(tmp)

> # now see what it looks like
> tmp
[1] Dem Rep Ind Rep Dem
Levels: Dem Ind Rep
```

When you print out the factor, you see the character strings. But R does not store the data values as character strings; it stores them as integers.

```
> storage.mode(tmp)
[1] "integer"
```

You can see what the integers are by applying the function `as.integer` to it.

```
> as.integer(tmp)
[1] 1 3 2 3 1
```

When R turned the variable into a factor, this is what it did. First, it scanned the data and found three unique character strings (`Dem`, `Rep`, `Ind`). It sorted these unique values by alphabetical order and assigned them integer values (1 for `Dem`, 2 for `Ind`, 3 for `Rep`). Then it created a vector of integers representing the original data values. So a factor is a vector of integers, taking values $1, 2, \dots, k$, where k is the number of unique values in the original data vector. But it is more than just a vector of integers, because it has some extra “attributes” (characteristics). The most important attribute is `levels`, which is the set of unique values in the original data vector.

```
> # see the levels of the factor
> levels(tmp)
[1] "Dem" "Ind" "Rep"
```

In fact, using the `levels` and the integer codes, you can recover the original data.

```
> levels(tmp)[as.integer(tmp)]
[1] "Dem" "Rep" "Ind" "Rep" "Dem"
```

Numeric variables can also be turned into factors.

```
> tmp <- c( -5.01, -7.65, 1.09, 4.31, 2.54)
> factor(tmp)
[1] -5.01 -7.65 1.09 4.31 2.54
Levels: -7.65 -5.01 1.09 2.54 4.31

> tmp <- c( 1, 4, 3, 2, 5, 2, 3, 1, 4)
> factor(tmp)
[1] 1 4 3 2 5 2 3 1 4
Levels: 1 2 3 4 5
```

If you have a variable that is already coded as $1, 2, \dots, k$, why would you turn it into a factor? Suppose you have a numeric variable X coded as $1, 2, \dots, k$, and you fit a regression model with the R function `lm` using X as a predictor. If you do not turn X into a factor, the model will be

$$Y = \beta_0 + \beta_1 X + \text{error}. \quad (6)$$

But if you turn X into a factor, the model will be

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \text{error}, \quad (7)$$

where X_1, \dots, X_{k-1} are a set of codes created by R to distinguish among the levels of X .

What coding scheme will R use? A factor has another

attribute called **contrasts** which determines the coding scheme when the factor is included as a predictor in a regression model. The **contrasts** attribute is a matrix with k rows and $k - 1$ columns. It is the matrix that we have been calling A , except that it does not have the column of 1's on the left-hand side.

```
> x <- c("Dem", "Rep", "Ind", "Rep", "Dem")
> x <- factor(x)
> contrasts(x)
   Ind Rep
Dem    0   0
Ind    1   0
Rep    0   1
```

By default, R will create dummy codes using the first category as the reference group. But this is easily changed. You can choose among several built-in coding schemes, including **contr.sum** (which we have called “effect coding”), **contr.treatment** (which we have called “dummy coding”) and **contr.helmert** (Helmert contrasts). To see what these are, you can call the functions **contr.sum**, **contr.treatment** and **contr.helmert** with the number of categories k as the argument.

```
> contr.sum(4)
 [,1] [,2] [,3]
 1     1     0     0
 2     0     1     0
 3     0     0     1
 4    -1    -1    -1
```

```
> contr.treatment(4)
  2 3 4
1 0 0 0
2 1 0 0
3 0 1 0
4 0 0 1

> contr.helmert(4)
 [,1] [,2] [,3]
1    -1   -1   -1
2     1   -1   -1
3     0    2   -1
4     0    0    3
```

The function `contr.treatment` has an optional second argument, `base`, which tells R which category should serve as the baseline (i.e. the reference group. The default is `base=1`, but you can change it to something else.

```
> contr.treatment(4, base=3)
  1 2 4
1 1 0 0
2 0 1 0
3 0 0 0
4 0 0 1
```

You can change the coding scheme for a factor like this.

```
> x <- c("Dem", "Rep", "Ind", "Rep", "Dem")
> x <- factor(x)
> contrasts(x)
  Ind Rep
Dem    0   0
Ind    1   0
Rep    0   1
```

```

> # change to effect coding
> contrasts(x) <- "contr.sum"
> contrasts(x)
 [,1] [,2]
Dem    1    0
Ind    0    1
Rep   -1   -1

> # change to Helmert coding
> contrasts(x) <- "contr.helmert"
> contrasts(x)
 [,1] [,2]
Dem   -1   -1
Ind    1   -1
Rep    0    2

```

Or you can provide your own coding matrix, like this.

```

> x <- c("Dem", "Rep", "Ind", "Rep", "Dem")
> x <- factor(x)
> contrasts(x)
      Ind Rep
Dem    0    0
Ind    1    0
Rep    0    1
>
> tmp <- matrix(NA, 3, 2)
> tmp[,1] <- c(1,0,0)
> tmp[,2] <- c(0,1,0)
> contrasts(x) <- tmp
> contrasts(x)
 [,1] [,2]
Dem    1    0
Ind    0    1
Rep    0    0

```

When you create your own coding matrix, it's a good idea to give names to the columns. Then, when you use the factor as a predictor in a regression model, the regression

coefficients will be given those names.

```
> dimnames(tmp) <- list(NULL, c("Dummy.Dem", "Dummy.Ind") )
> contrasts(x) <- tmp
> contrasts(x)
   Dummy.Dem Dummy.Ind
Dem          1          0
Ind          0          1
Rep          0          0
```

Example. The following dataset, which were obtained from

<http://lib.stat.cmu.edu/DASL/Datafiles/teacherpaydat.html>

reports public teacher pay and spending on public schools per pupil for 50 states and the District of Columbia in the year 1985. The data file **teacherpay.dat** looks like this.

	PAY	SPEND	AREA
ME	19583	3346	1
NH	20263	3114	1
VT	20325	3554	1
MA	26800	4642	1
RI	29470	4669	1

- lines omitted -

CA	29132	3608	3
AK	41480	8349	3
HA	25845	3766	3

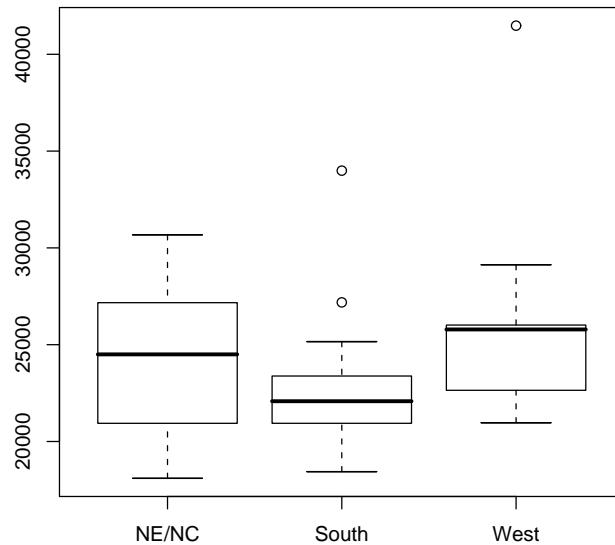
The variables are:

PAY = Average public school teacher annual salary (\$)
 SPEND = Spending on public schools per pupil (\$)
 AREA = Region (1=Northeast and North Central,
 2=South, 3=West)

Let's read in the data and convert AREA to a factor.

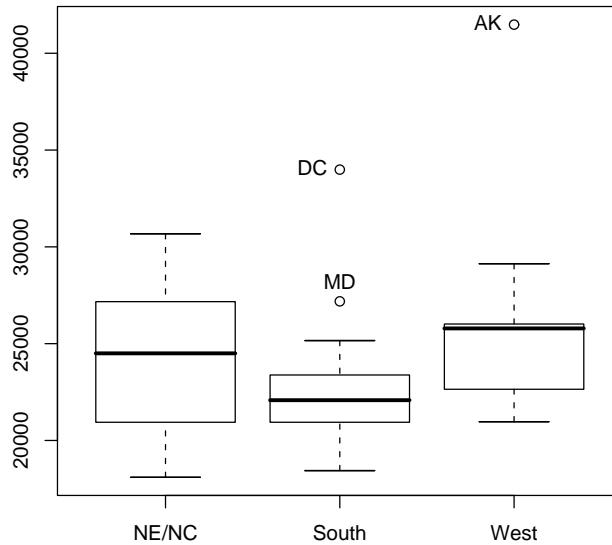
If you try to create a “scatterplot” and the x-axis variable is a factor, then R will set it up as side-by-side boxplots.

```
> plot( pay$AREA, pay$PAY )
```



We can use the `identify` function to tag the unusual observations.

```
> # use the row names of the dataset as labels  
> identify( pay$AREA, pay$PAY, labels=dimnames(pay)[[1]] )
```



Now let's regress PAY on the dummy variables for AREA.

```
> # regress PAY on AREA
> result <- lm( PAY ~ AREA, data=pay)
> summary(result)

Call:
lm(formula = PAY ~ AREA, data = pay)

Residuals:
    Min      1Q  Median      3Q     Max 
-6329.1 -2592.1 - 370.6  2143.4 15321.4 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 24424.1     887.9   27.507 <2e-16 ***
AREASouth   -1530.1    1327.5   -1.153    0.255    
AREAWest     1734.5    1436.0    1.208    0.233    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4069 on 48 degrees of freedom
```

Multiple R-Squared: 0.09008, Adjusted R-squared: 0.05217
F-statistic: 2.376 on 2 and 48 DF, p-value: 0.1038

The t-statistics tell us that

- teacher pay in the South was not significantly different from teacher pay in the Northeast/North Central ($p = .255$), and
- teacher pay in the West was not significantly different from teacher pay in the Northeast/North Central ($p = .233$).

Does this mean that there are no significant regional differences? Not necessarily. Although the South and West are not significantly different from the Northeast/North Central region, it is still possible that the South and West may be different from each other.

But if we look at the F-statistic at the bottom, we see that the fit of this model is not significantly better than that of the intercept-only model at the .05 level ($p = .1038$). We cannot conclude beyond a reasonable doubt that the differences in average pay across the regions represent a regional effect; they could simply be noise.

ORDINAL PREDICTORS AND THE LOF TEST

In the last lecture, we discussed how to include a k -level categorical predictor X in a regression model by creating a set of codes X_1, \dots, X_{k-1} . Today we will continue this discussion, but we will change our notation. We will now denote the categorical variable by C , and the codes that we create to represent C in the regression model will be C_1, \dots, C_{k-1} .

Categorical predictors in the presence of other predictors. Many different coding schemes could be used to create C_1, \dots, C_{k-1} . But if we fit the model

$$Y = \beta_0 + \beta_1 C_1 + \beta_2 C_2 + \cdots + \beta_{k-1} C_{k-1} + \text{error},$$

then, regardless of the coding scheme, the fitted value \hat{y}_i becomes the sample mean of Y within the group to which subject i belongs. That is, if \bar{y}_j is the mean response in the sample among all subjects for whom $C = j$, then the fitted value for each subject in that group is \bar{y}_j .

What happens when we add other predictors to the model,

so that

$$\begin{aligned} Y &= \beta_0 + \beta_1 C_1 + \cdots + \beta_{k-1} C_{k-1} \\ &\quad + \beta_k X_1 + \beta_{k+1} X_2 + \cdots + \text{error?} \end{aligned}$$

Because the subjects within a group $C = j$ will have varying values of X_1, X_2, \dots , their fitted values will no longer be the same. However, **the average value of the fitted values within the group will still be equal to the sample mean \bar{y}_j** . (To see this, note that the dummy indicator for $C = j$ lies within $\mathcal{R}(X)$ and is therefore orthogonal to \hat{y} .)

When the additional predictors X_1, X_2, \dots are included in the model, then the meaning of the coefficients $\beta_0, \beta_1, \dots, \beta_{k-1}$ also change, and their meaning depends on the coding scheme. For example, if C_1, \dots, C_{k-1} are dummy codes, then

- β_0 is the mean response for the reference group when all the additional predictors X_1, X_2, \dots are zero,
- β_1 is the mean response when $C = 1$ minus the mean response for the reference group when X_1, X_2, \dots are held constant,

and so on.

Ordinal predictors. Last time, we considered a variety of coding schemes for C , including dummy codes, effect codes and Helmert contrasts. Now suppose that the levels

of C are ordered, in the send that $C = j$ is “less than” $C = j + 1$ in some meaningful way for $j = 1, \dots, k - 1$. When C is an ordered categorical (i.e., ordinal) variable, is is common to describe the effect of C with **polynomial contrasts**. That is, we create variables from that represent the linear effect of C , the quadratic effect of C , etc. up to the $(k - 1)$ th degree.

With $k = 3$, the coding matrix might look like this.

group	linear	quad
1	-1	1
2	0	-2
3	1	1

With $k = 4$, it might look like this.

group	linear	quad	cubic
1	-3	1	-1
2	-1	-1	3
3	1	-1	-3
4	3	1	1

Codes like these, which are called **orthogonal polynomial contrasts**, are often tabulated in the appendices of regression textbooks.

If the variable C is coded as $C = 1, 2, \dots, k$ (or any linear transformation of $1, 2, \dots, k$), then the the l th degree polynomial regression model

$$Y = \beta_0 + \beta_1 C + \beta_2 C^2 + \cdots + \beta_l C^l + \text{error}$$

will give the same \hat{y}_i 's as

$$Y = \beta_0 + \beta_1 C_1 + \beta_2 C_2 + \cdots + \beta_l C_l + \text{error},$$

where C_1, \dots, C_l are the first l terms from the orthogonal polynomial coding matrix. So why would we use orthogonal polynomials, rather than simply regressing Y on C, C^2, \dots, C^l ?

If the data are balanced in the sense that there are equal numbers of subjects in each category of C , then the columns of the design matrix that we get from an orthogonal polynomial coding scheme will be orthogonal to the first column of the design matrix (the constant) and to each other. The estimated coefficients will be uncorrelated and will not share significance.

When the data are balanced, using an orthogonal polynomial coding scheme is essentially the same as setting up the design matrix with columns containing $1, C, C^2, \dots$ and applying the QR decomposition. I say “essentially the same” because in the QR decomposition, each column of Q is scaled to have a magnitude of one, so the elements of the design matrix are ugly real numbers. In an orthogonal polynomial coding scheme from a textbook table, the entries are scaled to have integer values so that they will look pretty. This prettiness does not change the model in any meaningful way; it simply rescales the coefficients.

If you want to use an orthogonal polynomial coding scheme but don't have access to a table, you can easily create your

own using R. For example, suppose that you need orthogonal polynomials for an ordinal variable with $k = 5$ levels. You could start with the matrix whose columns are $1, 2, \dots, 5$ raised to the power of 1, 2, up to $k - 1 = 4$:

$$A = \begin{bmatrix} 1 & 1^2 & 1^3 & 1^4 \\ 2 & 2^2 & 2^3 & 2^4 \\ 3 & 3^2 & 3^3 & 3^4 \\ 4 & 4^2 & 4^3 & 4^4 \\ 5 & 5^2 & 5^3 & 5^4 \end{bmatrix}$$

Then you replace each column by the residuals from the regression of that column on the columns before it, including a constant in each regression.

```
> a1 <- 1:5
> a2 <- a1^2
> a3 <- a1^3
> a4 <- a1^4
> a <- cbind(a1,a2,a3,a4)
> a
      a1  a2  a3  a4
[1,]  1   1   1   1
[2,]  2   4   8  16
[3,]  3   9  27  81
[4,]  4  16  64 256
[5,]  5  25 125 625

> new.a <- a
> new.a[,1] <- lm( a1 ~ 1 )$res      # intercept only
> new.a[,2] <- lm( a2 ~ a1 )$res
> new.a[,3] <- lm( a3 ~ a1 + a2 )$res
> new.a[,4] <- lm( a4 ~ a1 + a2 + a3 )$res
```

```
> new.a
      a1 a2          a3          a4
[1,] -2.000000e+00  2 -1.200000e+00  0.3428571
[2,] -1.000000e+00 -1  2.400000e+00 -1.3714286
[3,] -1.642566e-17 -2  3.033001e-15  2.0571429
[4,]  1.000000e+00 -1 -2.400000e+00 -1.3714286
[5,]  2.000000e+00  2  1.200000e+00  0.3428571
```

Then, if you like, you can multiply and divide the columns by integers until the values are pretty.

```
> new.a[,3] * 10
[1] -1.200000e+01  2.400000e+01  3.033001e-14 -2.400000e+01  1.200000e+01
> new.a[,3] * 10/12
[1] -1.000000e+00  2.000000e+00  2.527501e-15 -2.000000e+00  1.000000e+00
> new.a[,3] <- new.a[,3] * 10/12

> new.a[,4] * 7
[1]  2.4 -9.6 14.4 -9.6  2.4
> new.a[,4] * 70
[1]  24 -96 144 -96  24
> new.a[,4] * 70/24
[1]  1 -4  6 -4  1
> new.a[,4] <- new.a[,4] * 70/24
> new.a
      a1 a2          a3          a4
[1,] -2.000000e+00  2 -1.000000e+00  1
[2,] -1.000000e+00 -1  2.000000e+00 -4
[3,] -1.642566e-17 -2  2.527501e-15  6
[4,]  1.000000e+00 -1 -2.000000e+00 -4
[5,]  2.000000e+00  2  1.000000e+00  1

> new.a <- round(new.a,12) # get rid of rounding error
> new.a
      a1 a2 a3 a4
[1,] -2   2  -1  1
[2,] -1  -1   2 -4
[3,]  0  -2   0  6
[4,]  1  -1  -2 -4
[5,]  2   2   1  1
```

So the orthogonal polynomial coding matrix for a five-level ordinal variable is:

group	linear	quad	cubic	quartic
1	-2	2	-1	1
2	-1	-1	2	-4
3	0	-2	0	6
4	1	-1	-2	-4
5	2	2	1	1

Remember: If you use one of these orthogonal polynomial coding schemes, then the corresponding columns of the design matrix will be orthogonal if the data are balanced, i.e. if the number of subjects in each category is equal. If the data are not balanced, then variables defined by the orthogonal polynomial coding scheme are not perfectly uncorrelated.

Orthogonal polynomials and ordered factors in R.
R has a coding scheme called `contr.poly`. You can create a factor and set its `contrasts` attribute to `contr.poly`, like this.

```
> # a five-point Likert scale item
> x <- c(1,3,2,5,4,3,4,1,2,5,3)
> x <- factor(x)
> levels(x)
[1] "1" "2" "3" "4" "5"
> levels(x) <- c("strongly disagree", "disagree", "neutral",
+     "agree", "strongly agree")
> contrasts(x)
```

	disagree	neutral	agree	strongly agree
strongly disagree	0	0	0	0
disagree	1	0	0	0
neutral	0	1	0	0
agree	0	0	1	0
strongly agree	0	0	0	1

```
> contrasts(x) <- "contr.poly"
> contrasts(x)
      .L          .Q          .C          ^4
[1,] -6.324555e-01  0.5345225 -3.162278e-01  0.1195229
[2,] -3.162278e-01 -0.2672612  6.324555e-01 -0.4780914
[3,] -3.287978e-17 -0.5345225  1.595204e-16  0.7171372
[4,]  3.162278e-01 -0.2672612 -6.324555e-01 -0.4780914
[5,]  6.324555e-01  0.5345225  3.162278e-01  0.1195229
```

Notice that R's coding scheme does not use pretty integers. If you want pretty integers, you will have to do it yourself.

```
> new.a
      a1 a2 a3 a4
[1,] -2  2 -1  1
[2,] -1 -1  2 -4
[3,]  0 -2  0  6
[4,]  1 -1 -2 -4
[5,]  2  2  1  1

> dimnames(new.a) <- list( NULL, c("lin","quad","cub","^4"))
> contrasts(x) <- new.a
> contrasts(x)
      lin quad cub ^4
strongly disagree -2    2  -1  1
disagree          -1   -1   2 -4
neutral           0   -2   0  6
agree             1   -1  -2 -4
strongly agree    2    2   1  1
```

There's also another way to do it. When you create a factor, the **factor** function has an optional argument

`ordered` whose default value is `F`. If you specify `ordered=T`, then R creates something called an **ordered factor**. An ordered factor is just a factor, except that its default coding scheme is `contr.poly` rather than `contr.treatment`.

```
> # a five-point Likert scale item
> x <- c(1,3,2,5,4,3,4,1,2,5,3)
> x <- factor(x, ordered=T)
> levels(x) <- c("strongly disagree", "disagree", "neutral",
+   "agree", "strongly agree")
> contrasts(x)
      .L          .Q          .C      ^4
strongly disagree -6.324555e-01  0.5345225 -3.162278e-01  0.1195229
disagree           -3.162278e-01 -0.2672612  6.324555e-01 -0.4780914
neutral            -3.287978e-17 -0.5345225  1.595204e-16  0.7171372
agree              3.162278e-01 -0.2672612 -6.324555e-01 -0.4780914
strongly agree     6.324555e-01  0.5345225  3.162278e-01  0.1195229
```

The lack-of-fit test. When you have an ordinal predictor C , the main reason for using an orthogonal polynomial coding scheme (or, equivalently, a regression on C, C^2, \dots) is that you may be able to find a model that captures the relationship between Y and C without using all of the terms.

If you have a k -level ordinal predictor, the $(k - 1)$ th degree polynomial model will give the same fit to the data as a model with $k - 1$ dummy codes, effect codes or Helmert constraints. All of these models are “saturated” in the sense that they impose no constraints on the mean response in each category $C = 1, \dots, k$. The predicted values for Y

will simply be the sample means in each C -group (assuming that there are no other predictors in the model). But if the categories of C are truly ordered, then the relationship between Y and C might be described well enough by a simpler model, perhaps a linear or quadratic one. With an ordinal predictor, we hope that the trend can be described by an l th degree polynomial for some $l < (k - 1)$, so that the higher-order effects can be omitted from the model and considered to be random error.

If you have an l th-degree polynomial model

$$Y = \beta_0 + \beta_1 C + \beta_2 C^2 + \cdots + \beta_l C^l + \text{error}, \quad (1)$$

and if this model is not saturated ($l < k - 1$), then the partial F-test comparing the fit of (1) to the saturated alternative

$$Y = \beta_0 + \beta_1 C + \beta_2 C^2 + \cdots + \beta_{k-1} C^{k-1} + \text{error} \quad (2)$$

is called a **lack-of-fit (LOF) test**. The numerator of the LOF test F statistic is the extra sum of squares for the terms omitted from (1) (the regression sum of squares for C^{l+1}, \dots, C^k given C, \dots, C^l), divided by the number of omitted terms ($k - 1 - l$). You can also get the extra sum of squares by the difference in SS_{Err} for the two models. The denominator for the LOF test F statistic is the MSE for the saturated alternative (2).

When carrying out the LOF test, you can fit the saturated alternative model as in (2), using a $(k - 1)$ th degree

polynomial. Or you can fit the saturated model using a set of $k - 1$ dummy codes, effect codes, Helmert constraints, etc. Any coding scheme will give the same fit and the same SS_{Err} , as long as the model is saturated.

Generalizing the LOF test. The LOF test is a very general concept that can be applied to all kinds of regression models. Suppose you have an arbitrary regression model with p coefficients,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{p-1} X_{p-1} + \text{error}, \quad (3)$$

and you want to test whether this model is true in the sense that the mean response given the predictors X_1, \dots, X_{p-1} really is a linear function of these predictors. That is, you want to test the fit of this model against the fit of an alternative model **that makes no assumptions about how the mean response is related to any of the predictors.**

Can you perform such a test? Perhaps. The key issue is **whether the dataset has enough replication.**

Suppose that we identify all the unique combinations of the values of the covariates X_1, \dots, X_{p-1} appearing in the sample. Each combination of covariates will be called a **covariate pattern**. Let n^* be the number of unique covariate patterns in the sample. If some of the predictors are continuous, then n^* may be approximately or exactly equal to n , and the LOF test is impossible. But if the

predictors are coarse (taking only a small number of different values), then we may have enough replicate observations within the unique covariate patterns to perform the test.

The LOF test is simply a partial F-test, where the null hypothesis is the model of interest (3), and the alternative is a saturated model with an intercept plus $n^* - 1$ dummy codes (or effect codes or whatever you like) to distinguish among the unique covariate patterns. The numerator of the F statistic is the difference in SS_{Err} between the null model (3) and the saturated model, divided by $n^* - p$.

The denominator is the MSE from the saturated model.

If n^* is not too large, you may be able to create the $n^* - 1$ dummy codes and fit the saturated model using regression software. If n^* is very large, that might not be practical. But you can always fit the saturated model by noting that the fitted values are equal the sample means of Y within each covariate pattern. Let

$$\mathcal{I}_j = \{i : \text{subject } i \text{ belongs to pattern } j\}$$

for each pattern $j = 1, \dots, n^*$. Let n_j be the number of subjects in pattern j ,

$$\bar{y}_j = \frac{1}{n_j} \sum_{i \in \mathcal{I}_j} y_i$$

the sample mean response in pattern j , and

$$S_j^2 = \frac{1}{n_j - 1} \sum_{i \in \mathcal{I}_j} (y_i - \bar{y}_j)^2$$

the sample variance of the responses in pattern j . Then the error sum of squares for the saturated model is

$$SS_{Err} = \sum_{j=1}^{n^*} (n_j - 1) S_j^2.$$

Examples of LOF tests will be given later, after we discuss interactions.

INTERACTIONS

Interactions. Consider a model with two predictors,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \text{error}. \quad (1)$$

This model asserts that the effects of X_1 and X_2 on the mean of Y are linear and additive. It says that a one-unit increase in X_1 increases the mean response by β_1 units, regardless of the value of X_2 . And it says that a one-unit increase in X_2 increases the mean response by β_2 units, regardless of the value of X_1 .

But what if the effect of X_1 varies with X_2 and vice-versa? In that case, we need to consider a model with interactions. An interaction means that the effect of a predictor on the response varies with another predictor. One common way to allow an interaction is to include a third predictor equal to the product of X_1 and X_2 ,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \text{error}. \quad (2)$$

If we rewrite the model as

$$Y = \beta_0 + (\beta_1 + \beta_3 X_2) X_1 + \beta_2 X_2 + \text{error},$$

we see that a one-unit increase in X_1 will increase the mean response by $\beta_1 + \beta_3 X_2$. Similarly, a one-unit

increase in X_2 will increase the mean response by $\beta_2 + \beta_3 X_1$. Under this model, the effect of X_1 changes linearly with X_2 , and vice-versa. For this reason, model (2) is said to have a **linear by linear interaction**. In this model, β_1 and β_2 are called the “main effects” of X_1 and X_2 , and β_3 is called the interaction.

When we fit the model (2). How do we interpret the results? Suppose we want to describe the effect of X_1 on Y . The estimate of this effect is a function of X_2 ,

$$\text{estimated effect of } X_1 = \hat{\beta}_1 + \hat{\beta}_3 X_2.$$

The variance of this estimate is

$$V(\hat{\beta}_1) + X_2^2 V(\hat{\beta}_3) + 2 X_2 \text{Cov}(\hat{\beta}_1, \hat{\beta}_3). \quad (3)$$

Estimates of these variances and covariance would come from the elements of $S^2(X^T X)^{-1}$. Substituting these estimates into (3) and taking the square root, we get the standard error for the estimated effect. A 95% confidence interval for the effect of X_1 at a specific value of X_2 is

$$\text{estimated effect} \pm t_{.975, n-p} SE.$$

We can compute the estimate and confidence interval at any given value of X_2 . And we can plot the estimates and intervals as functions of X_2 , in the same way that we plotted confidence intervals for mean response and prediction intervals for future observations in Lecture 11.

We will now discuss three different kinds of hypothesis

tests that can be applied to this model. First is the test for no interaction, $H_0 : \beta_3 = 0$, which is based on the t-statistic for β_3 . This test tells us whether the interaction term is needed. If a linear transformation is applied to X_1 or X_2 , the values of the coefficients may change, but outcome of this test is unaffected. If $\hat{\beta}_3$ is not significantly different from zero, it usually makes sense to describe the data by the simpler model (1) without the interaction.

The second kind of test helps us to judge whether one of the predictors has any effect on Y at all. Under model (2), the effect of X_1 on Y is $\beta_1 + \beta_3 X_2$. The null hypothesis “ X_1 has no effect on Y ” corresponds to $H_0 : \beta_1 = \beta_3 = 0$. Similarly, “ X_2 has no effect on Y ” corresponds to $H_0 : \beta_2 = \beta_3 = 0$. These are carried out as partial F-tests. The outcome of these tests is not affected if linear transformations are applied to X_1 or X_2 .

The third kind of test concerns one of the main effects. Consider the null hypothesis $H_0 : \beta_1 = 0$. What does it mean? It means that X_1 **has no effect on Y when $X_2 = 0$** . Similarly, $H_0 : \beta_2 = 0$ means that X_2 **has no effect on Y when $X_1 = 0$** . Are these null hypotheses meaningful? Sometimes they are, but usually they are not. Moreover, it is important to note that the meaning and outcome of these tests may change if linear transformations are applied to X_1 or X_2 . These tests depend on how the original variables are coded. For this reason, some textbooks say that you should never test the

significance of a main effect when an interaction is present. “Never” is too strong a word, because on occasion the hypothesis $H_0 : \beta_1 = 0$ or $H_0 : \beta_2 = 0$ does have scientific meaning. But in most cases it does not. If you fit the model (2) and find that $\hat{\beta}_3$ is significantly different from zero, you should keep both of the main effects in the model as well even if they are not significant, unless there is a good scientific reason to remove them.

Interactions with categorical predictors. Suppose that we have a k -level categorical predictor C and another predictor X_1 . And suppose that we re-express the categorical variable as C_1, \dots, C_{k-1} , a set of dummy codes, effect codes, etc. The interaction of C with X_1 is expressed by the $k - 1$ product terms $C_1 X_1, C_2 X_1, \dots, C_{k-1} X_1$. If we fit the model

$$\begin{aligned} Y &= \alpha + \beta_1 C_1 + \cdots + \beta_{k-1} C_{k-1} \\ &\quad + \gamma X_1 \\ &\quad + \delta_1 C_1 X_1 + \cdots + \delta_{k-1} C_{k-1} X_1 + \text{error}, \end{aligned}$$

then we are saying that **the mean response is a linear function of X_1 within each category of C , with different slopes and intercepts**. Notice that this model has $2k$ coefficients. It estimates a different intercept and slope for each category $C = 1, \dots, k$. The intercepts and slopes can be obtained from the coefficients, but how to do so depends on the coding scheme used for C .

If C_1, \dots, C_{k-1} are dummy codes, then

- α is the intercept for the reference group (i.e. the mean response when $C = k$ and $X_1 = 0$),
- γ is the slope for the reference group,
- $\alpha + \beta_j$ is the intercept for group $C = j$, and
- $\gamma + \delta_j$ is the slope for group $C = j$.

If C_1, \dots, C_{k-1} are effect codes, then

- α is the unweighted average of the intercepts for groups $C = 1, \dots, k$,
- γ is the unweighted average of the slopes for groups $C = 1, \dots, k$,
- β_j is the intercept for group $C = j$ minus the unweighted average of the intercepts, and
- δ_j is the slope for group $C = j$ minus the unweighted average of the slopes.

The hypothesis of no interactions,

$H_0 : \delta_1 = \delta_2 = \dots = \delta_{k-1} = 0$, means that all slopes are equal. This can be assessed by a partial F-test. The meaning and outcome of this test do not depend on the coding scheme. However, the meaning and outcome of other tests do change with the coding scheme.

For example, consider the null hypothesis $H_0 : \gamma = 0$. With dummy coding, it means that the slope for the reference

group is zero, i.e. that X_1 has no effect on Y when $C = 1$.

With effect coding, however, $H_0 : \gamma = 0$ means that the unweighted average of the slopes across all groups is zero.

As another example, consider $H_0 : \beta_1 = 0$. With dummy coding, it means that the intercept for group $C = 1$ is the same as for the reference group. With effect coding, however, it means that the intercept for group $C = 1$ equals the unweighted average of the intercepts across all groups.

Example. In Lecture 22, we looked at a dataset that recorded the following three variables for each state and the District of Columbia.

PAY = Average public school teacher annual salary (\$)
SPEND = Spending on public schools per pupil (\$)
AREA = Region (1=Northeast and North Central,
2=South, 3=West)

Let's fit a model that assumes that SPEND has a linear effect on PAY with a different slope and intercept for each region.

```
> # read in data  
> pay <- read.table( "teacherpay.dat", header=T)  
  
> # change AREA to a factor
```

```

> pay$AREA <- factor(pay$AREA)

> # give descriptive names to levels
> levels(pay$AREA) <- c("NE/NC", "South", "West")

> # see what the contrasts are
> contrasts(pay$AREA)
  South West
NE/NC      0    0
South      1    0
West       0    1

> # regress PAY on AREA, SPEND and interaction terms
> result <- lm(PAY ~ AREA + SPEND + AREA:SPEND, data=pay)
> summary(result)

Call:
lm(formula = PAY ~ AREA + SPEND + AREA:SPEND, data = pay)

Residuals:
    Min      1Q  Median      3Q     Max 
-3758.6 -1351.8 -237.5 1480.0 6162.0 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10674.7726  2536.7880   4.208 0.000121 ***
AREASouth   -1089.5255  3575.2479  -0.305 0.761968  
AREAWest     3950.5552  3090.2289   1.278 0.207662  
SPEND        3.5249    0.6378   5.527 1.57e-06 ***
AREASouth:SPEND  0.5396    0.9851   0.548 0.586597  
AREAWest:SPEND -0.5821    0.7640  -0.762 0.450062  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2273 on 45 degrees of freedom
Multiple R-Squared: 0.7338,    Adjusted R-squared: 0.7042 
F-statistic: 24.81 on 5 and 45 DF,  p-value: 6.54e-12

```

Neither of the interaction terms is significant, but we don't know whether they are jointly significant. To find out, we can look at the partial F statistic from the ANOVA table.

```
> anova(result)
Analysis of Variance Table

Response: PAY
  Df   Sum Sq  Mean Sq  F value    Pr(>F)
AREA      2 78676547 39338273  7.6139  0.001419 ***
SPEND     1 552484936 552484936 106.9333 1.801e-13 ***
AREA:SPEND 2  9720281   4860141   0.9407  0.397903
Residuals 45 232498501   5166633
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No, it is not significant. The ANOVA table suggests that the main effects of AREA are important, but it is a sequential test that does not take into account the effect of SPEND. Let's test the null hypothesis that AREA has no effect whatsoever. This is a partial F-test of the null hypothesis that the main effects for AREA and the interactions are simultaneously zero.

```
> # re-fit the model with AREA entered after SPEND
> result <- lm(PAY ~ SPEND + AREA + AREA:SPEND, data=pay)

> anova(result)
Analysis of Variance Table

Response: PAY
  Df   Sum Sq  Mean Sq  F value    Pr(>F)
SPEND     1 608555015 608555015 117.7856 3.764e-14 ***
AREA      2 22606468 11303234  2.1877  0.1240
SPEND:AREA 2  9720281   4860141   0.9407  0.3979
Residuals 45 232498501   5166633
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F statistic is

$$F = \frac{(22,606,468 + 9,720,281)/4}{5,166,633} = 1.564,$$

and the p-value is

$$P(F_{4,45} \geq 1.564) = 0.200,$$

so there is little evidence in these data of any regional effects.

Interactions between categorical predictors.

Suppose now that we have two categorical predictors X_1 and X_2 . For simplicity, let's suppose that they are both binary. Let's suppose that X_1 indicates a subject's sex (male, female) and X_2 indicates the treatment received in the experiment (drug, placebo). The variables X_1 and X_2 may be defined using dummy (0, 1) codes or effect (+1, -1) codes. If we fit the model without an interaction,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \text{error},$$

then, regardless of the coding method, the model implies that **the treatment has the same effect for males and females**. If we fit the model with an interaction,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \text{error},$$

then, regardless of the coding method, the model implies that **the treatment has different effects for males and females**. But the exact interpretation of the

coefficients in these models will obviously depend on the coding method.

Suppose that we use dummy coding,

$$X_1 = \begin{cases} 1 & \text{for male,} \\ 0 & \text{for female,} \end{cases} \quad X_2 = \begin{cases} 1 & \text{for drug,} \\ 0 & \text{for placebo.} \end{cases}$$

Then the means for the four groups are as shown below.

$$\begin{aligned} \text{male, drug} &= \mu_{11} = \beta_0 + \beta_1 + \beta_2 + \beta_3, \\ \text{male, placebo} &= \mu_{12} = \beta_0 + \beta_1, \\ \text{female, drug} &= \mu_{21} = \beta_0 + \beta_2, \\ \text{female, placebo} &= \mu_{22} = \beta_0. \end{aligned}$$

It's helpful to arrange them in a 2×2 table.

	<i>Drug</i>	<i>Placebo</i>
<i>Male</i>	$\mu_{11} = \beta_0 + \beta_1 + \beta_2 + \beta_3$	$\mu_{12} = \beta_0 + \beta_1$
<i>Female</i>	$\mu_{21} = \beta_0 + \beta_2$	$\mu_{22} = \beta_0$

How do we interpret the coefficients? The intercept is

$$\beta_0 = \mu_{22},$$

so females taking the placebo are the baseline or reference group. The main effect for X_1 is

$$\beta_1 = \mu_{12} - \mu_{22},$$

which is the effect of sex (male minus female) among those

taking the placebo. The effect of sex (male minus female) among those taking the drug would be

$$\mu_{11} - \mu_{21} = \beta_1 + \beta_3.$$

The main effect for X_2 is

$$\beta_2 = \mu_{21} - \mu_{22},$$

which is the effect of the treatment (drug minus placebo) among females. The effect of the treatment (drug minus placebo) among males is

$$\mu_{11} - \mu_{12} = \beta_2 + \beta_3.$$

The interaction β_3 can be interpreted as

$$\beta_3 = (\beta_1 + \beta_3) - \beta_1 = (\mu_{11} - \mu_{21}) - (\mu_{12} - \mu_{22}),$$

which is the effect of sex (male minus female) among those taking the drug, minus the effect of sex (male minus female) among those taking the placebo. We can also interpret the interaction as

$$\beta_3 = (\beta_2 + \beta_3) - \beta_2 = (\mu_{11} - \mu_{12}) - (\mu_{21} - \mu_{22}),$$

the effect of treatment (drug minus placebo) among males, minus the effect of treatment (drug minus placebo) among females.

Now consider what happens when we switch to effect

coding. Let's define the codes as

$$X_1 = \begin{cases} +1 & \text{for male,} \\ -1 & \text{for female,} \end{cases} \quad X_2 = \begin{cases} +1 & \text{for drug,} \\ -1 & \text{for placebo.} \end{cases}$$

Then the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \text{error}$$

leads to the following table.

	<i>Drug</i>	<i>Placebo</i>
<i>Male</i>	$\mu_{11} = \beta_0 + \beta_1 + \beta_2 + \beta_3$	$\mu_{12} = \beta_0 + \beta_1 - \beta_2 - \beta_3$
<i>Female</i>	$\mu_{21} = \beta_0 - \beta_1 + \beta_2 - \beta_3$	$\mu_{22} = \beta_0 - \beta_1 - \beta_2 + \beta_3$

In this coding scheme, the intercept becomes

$$\beta_0 = \left(\frac{\mu_{11} + \mu_{21} + \mu_{21} + \mu_{22}}{4} \right),$$

an equally weighted average of the four group means. The main effect for sex becomes

$$\beta_1 = \left(\frac{\mu_{11} + \mu_{12}}{2} \right) - \left(\frac{\mu_{11} + \mu_{21} + \mu_{21} + \mu_{22}}{4} \right),$$

the unweighted average of the two male group means, minus the unweighted average of all four group means.

The main effect for treatment becomes

$$\beta_2 = \left(\frac{\mu_{11} + \mu_{21}}{2} \right) - \left(\frac{\mu_{11} + \mu_{21} + \mu_{21} + \mu_{22}}{4} \right),$$

the unweighted average of the two drug group means, minus the unweighted average of all four group means. The interaction can be interpreted as

$$\beta_3 = \frac{1}{4} [(\mu_{11} - \mu_{12}) - (\mu_{21} - \mu_{22})].$$

Therefore, $4\beta_3$ is the treatment effect (drug minus placebo) among males minus the treatment effect (drug minus placebo) among females.

If the data are balanced with equal numbers of subjects in each of the four groups, then effect coding will lead to a design matrix whose columns are orthogonal. Removing the interaction will not change the estimates or interpretation of the main effects. With dummy coding, the columns of the design matrix are correlated, and removing the interaction will change the estimates and the interpretation of the main effects.

MEASURES OF INFLUENCE

Measures of influence. Back in Lecture 17, we began to discuss the concept of influence. Influence refers to how much the results change when a single observation is added or removed. Influence is closely related to the diagonal elements of the hat matrix. The hat matrix is

$$H = X(X^T X)^{-1} X^T,$$

and its diagonal elements h_1, \dots, h_n are called the leverages. They measure how far the observations lie from the center of the covariate space. Because $\text{tr}H = p$, the average value of the h_i 's is p/n , and a value of h_i that is much larger than average indicates that the observation *might be* influential. But a point with high leverage is not necessarily influential. So the leverage is not really a measure of influence.

Change in predicted values (DFFIT). One way to measure actual influence is to see how much the predicted value for an observation changes when that observation is included in the model fit. We know that the vector of ordinary fitted values, $\hat{y} = Hy$, is distributed as

$N(X\beta, \sigma^2 H)$, so an individual fitted value $\hat{y}_i = x_i^T \hat{\beta}$ is normally distributed with mean $x_i^T \beta$ and variance $\sigma^2 h_i$. If we remove observation i from the model fit, then the least-squares estimates become

$$\hat{\beta}_{(i)} = \hat{\beta} - (1 - h_i)^{-1}(X^T X)^{-1} x_i \hat{\epsilon}_i$$

(see end of Lecture 17), and the predicted value for y_i when that observation is removed from the model fit is

$$\begin{aligned}\hat{y}_{(i)} &= x_i^T \hat{\beta}_{(i)} \\ &= y_i - \hat{\epsilon}_{(i)},\end{aligned}$$

where

$$\hat{\epsilon}_{(i)} = \left(\frac{1}{1 - h_i} \right) \hat{\epsilon}_i$$

is the PRESS residual as defined in Lecture 17. The change in the fitted value when observation i is included in the fit is

$$\begin{aligned}\hat{y}_i - \hat{y}_{(i)} &= (y_i - \hat{\epsilon}_i) - \left(y_i - \left(\frac{1}{1 - h_i} \right) \hat{\epsilon}_i \right) \\ &= \left(\frac{h_i}{1 - h_i} \right) \hat{\epsilon}_i\end{aligned}\tag{1}$$

It is common practice to scale this difference, $(\hat{y}_i - \hat{y}_{(i)})$, by the standard error of the fitted value when observation i is removed. That standard error is $S_{(i)} \sqrt{h_i}$, where $S_{(i)}$ is the square root of the MSE that we get when observation i is removed (Lecture 19). Dividing $(\hat{y}_i - \hat{y}_{(i)})$ by this standard error gives the measure of influence that we call

“DFFIT”, defined as

$$\text{DFFIT}_i = \frac{\hat{y}_i - \hat{y}_{(i)}}{S_{(i)}\sqrt{h_i}}.$$

Substituting (1) into the numerator, we get

$$\begin{aligned} \text{DFFIT}_i &= \frac{\left(\frac{h_i}{1-h_i}\right)\hat{\epsilon}_i}{S_{(i)}\sqrt{h_i}} \\ &= \left(\frac{\hat{\epsilon}_i}{S_{(i)}\sqrt{1-h_i}}\right) \sqrt{\frac{h_i}{1-h_i}} \\ &= \tilde{\epsilon}_{(i)} \sqrt{\frac{h_i}{1-h_i}}, \end{aligned}$$

where $\tilde{\epsilon}_{(i)}$ is the externally Studentized residual defined in Lecture 19. So DFFIT is just another kind of residual, inflated by a factor that depends on the leverage.

How large does DFFIT need to be before we consider it to be influential? KNNL say (p. 401) that if the dataset is small, then points with $|\text{DFFIT}_i| > 1$ should be considered influential. Other authors say $|\text{DFFIT}_i| > 2$. If the sample is large, then they suggest the rule should be $|\text{DFFIT}_i| > 2\sqrt{p/n}$.

Change in coefficients (DFBETA, Cook's distance). DFFIT measures the impact of an observation on the fitted value for that observation. Two other popular influence statistics measure the impact of an observation

on the estimated coefficients.

The impact of observation i on coefficient β_j is $\hat{\beta}_j$ minus $\hat{\beta}_{j(i)}$, the j th element of $\hat{\beta}_{(i)}$. It is common practice to scale this difference by a standard error for β_j with the i th observation removed from the fit. This standard error is $S_{(i)}$ times the square root of the (j, j) th element of $(X^T X)^{-1}$, which we can write as $S_{(i)} \sqrt{(X^T X)_{jj}^{-1}}$.

Dividing $(\hat{\beta}_j - \hat{\beta}_{j(i)})$ by this standard error gives a measure called “DFBETA,”

$$\text{DFBETA}_{j(i)} = \frac{(\hat{\beta}_j - \hat{\beta}_{j(i)})}{S_{(i)} \sqrt{(X^T X)_{jj}^{-1}}}.$$

We can interpret DFBETA as the number of SE's by which the estimate of β_j changes when observation i is included. Values of DFBETA greater than 2 or less than -2 are rare. KNNL suggest (p. 405) that we should consider the point influential if $|\text{DFBETA}_{j(i)}| > 1$ in a small dataset or if $|\text{DFBETA}_{j(i)}| > 2/\sqrt{n}$ in a large dataset.

An advantage of DFBETA is that it helps us to pinpoint the effect of each observation on each coefficient. But examining the DFBETAs for each coefficient can be tedious, especially if the model contains lots of predictors. The DFBETA's for all coefficients can be combined into a single measure called Cook's distance. The formula for

Cook's distance is

$$D_i = \frac{(\hat{\beta} - \hat{\beta}_{(i)})^T (X^T X) (\hat{\beta} - \hat{\beta}_{(i)})}{p S^2}.$$

This formula should look familiar to you. The estimated covariance matrix for $\hat{\beta}$ is $S^2(X^T X)^{-1}$, so this is an estimated Mahalanobis distance between $\hat{\beta}$ and $\hat{\beta}_{(i)}$, divided by p . Because we have substituted S^2 for the unknown σ^2 , The distribution of this quantity is something like $F_{p,n-p}$. Some authors recommend that you consider D_i to be large if $D_i > 1$, while others recommend the rule $D_i > F_{.50,p,n-p}$.

Influence diagnostics in R. In Lecture 18, we discussed the use of the R function `lm.influence`, which can be applied to the results of an `lm` fit to obtain h_i , $S_{(i)}$, and $\hat{\beta} - \hat{\beta}_{(i)}$. With a little effort, these quantities could be used to obtain DFFIT's, DFBETA's and Cook's distance. But R also has built-in functions for these influence measures. These five functions,

```
rstandard()
rstudent()
dffits()
dfbetas()
cooks.distance()
```

when applied to the result of an `lm` fit, produce the standardized residuals, externally Studentized residuals,

DFFIT's, DFBETA's and Cooks distances.

Example. Last time, we examined a regression model for predicting the average teacher salary in a state (PAY) given the region of the country (AREA) and the state expenditure per student (SPEND). We found a strong effect for SPEND but no significant main effects or interactions related to AREA. But when we first looked at these data, we noticed that there were a few states with unusually high leverage, and these states could be exerting undue influence on the results.

Here are the results from the model with a main effect for SPEND, main effects for AREA, and the SPEND × AREA interactions.

```
> # read in data
> pay <- read.table( "teacherpay.dat", header=T)
>
> # change AREA to a factor
> pay$AREA <- factor(pay$AREA)
>
> # give descriptive names to levels
> levels(pay$AREA) <- c("NE/NC","South","West")
>
> # regress PAY on AREA, SPEND and interctions
> result <- lm( PAY ~ AREA + SPEND + AREA:SPEND, data=pay)
> summary(result)
```

Call:

```
lm(formula = PAY ~ AREA + SPEND + AREA:SPEND, data = pay)
```

Residuals:

Min	1Q	Median	3Q	Max
-3758.6	-1351.8	-237.5	1480.0	6162.0

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)							
(Intercept)	10674.7726	2536.7880	4.208	0.000121 ***							
AREASouth	-1089.5255	3575.2479	-0.305	0.761968							
AREAWest	3950.5552	3090.2289	1.278	0.207662							
SPEND	3.5249	0.6378	5.527	1.57e-06 ***							
AREASouth:SPEND	0.5396	0.9851	0.548	0.586597							
AREAWest:SPEND	-0.5821	0.7640	-0.762	0.450062							

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

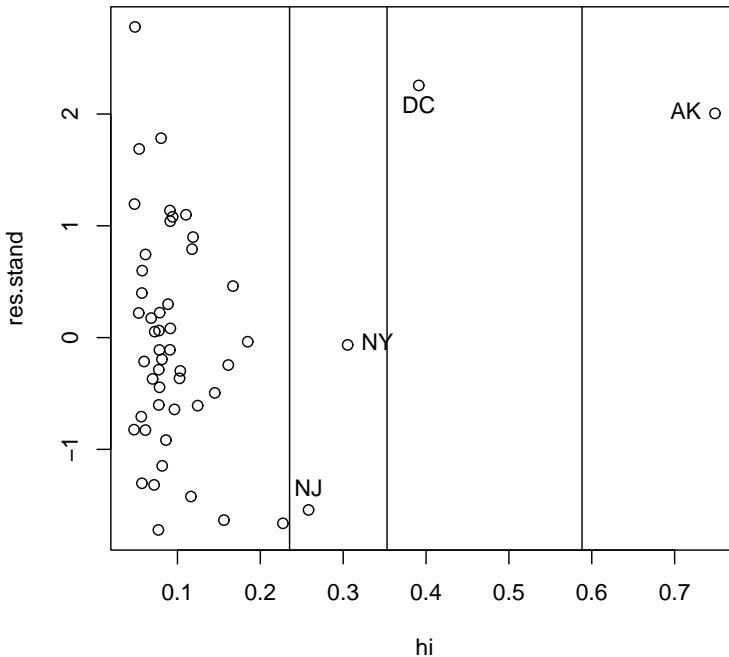
Residual standard error: 2273 on 45 degrees of freedom

Multiple R-Squared: 0.7338, Adjusted R-squared: 0.7042

F-statistic: 24.81 on 5 and 45 DF, p-value: 6.54e-12

First, let's plot the leverages versus the standardized residuals and identify the points with high leverage.

```
> # plot leverages versus standardized residuals
> hi <- lm.influence( result )$hat
> res.stand <- rstandard( result )
> plot( hi, res.stand)
>
> # add vertical lines corresponding to 2p/n 3p/n and 5p/n and
> # identify the high-leverage points
> n <- nrow(pay)
> p <- 6
> abline( v=2*p/n )
> abline( v=3*p/n )
> abline( v=5*p/n )
> identify( hi, res.stand, labels=dimnames(pay)[[1]] )
[1] 7 8 24 50
```



These states have high leverage, but are they influential?
 First, let's compute DFFIT's and identify the states with
 $DFFIT > 1$.

```
> DFFITS <- dffits( result )
> DFFITS[ DFFITS > 1 ]
      DC          AK
1.899019 3.587488
```

Including these states in the model substantially changes the predicted values for those states. Let's compare their values of \hat{y}_i and $\hat{y}_{(i)}$.

```
> # fitted values using all observations
> y <- pay$PAY
> yhat <- y - result$res
```

```

>
> # fitted values leaving out the observation in question
> res.PRESS <- res / (1-hi)
> yhat.i <- y - res.PRESS
>
> # compare them for states with high DFFIT
> cbind( yhat, yhat.i )[ DFFITS > 1, ]
      yhat     yhat.i
DC 29988.89 27417.51
AK 39194.77 32385.49

```

Including DC in the model raises the predicted value for DC by about \$2,500. Including AK raises the predicted value for AK by almost \$7,000.

Now let's look for observations with large DFBETA's.

```

> DFBETAS <- dfbetas( result )
> round( DFBETAS, 3 )
      (Intercept) AREASouth AREAWest   SPEND AREASouth:SPEND AREAWest:SPEND
ME      -0.269     0.191     0.221    0.215      -0.139      -0.179
NH      -0.174     0.123     0.143    0.148      -0.096      -0.124
VT      -0.187     0.133     0.153    0.132      -0.085      -0.110
MA       0.018    -0.013    -0.015   -0.024       0.015      0.020
RI      -0.192     0.136     0.158    0.245      -0.159      -0.205
CT       0.148    -0.105    -0.121   -0.179       0.116      0.149
NY       0.035    -0.025    -0.029   -0.039       0.026      0.033
NJ       0.741    -0.526    -0.608   -0.835       0.540      0.697
PA      -0.007     0.005     0.006    0.017      -0.011      -0.014
OH       0.086    -0.061    -0.070   -0.061       0.039      0.051
IN       0.295    -0.209    -0.242   -0.249       0.161      0.208
IL       0.212    -0.150    -0.174   -0.139       0.090      0.116
MI       0.233    -0.165    -0.191   -0.103       0.067      0.086
WI      -0.021     0.015     0.018    0.040      -0.026      -0.033
MN       0.025    -0.018    -0.020   0.028      -0.018      -0.023
IA      -0.097     0.069     0.080    0.068      -0.044      -0.056
MO       0.021    -0.015    -0.017   -0.018       0.012      0.015
ND      -0.085     0.061     0.070    0.074      -0.048      -0.061
SD      -0.458     0.325     0.376    0.401      -0.260      -0.335
NB      -0.132     0.094     0.108    0.108      -0.070      -0.090

```

KA	-0.033	0.023	0.027	-0.003	0.002	0.003
DE	0.000	0.472	0.000	0.000	-0.604	0.000
MD	0.000	0.008	0.000	0.000	-0.011	0.000
DC	0.000	-1.090	0.000	0.000	1.334	0.000
VA	0.000	0.013	0.000	0.000	-0.031	0.000
WV	0.000	-0.028	0.000	0.000	0.023	0.000
NC	0.000	-0.004	0.000	0.000	-0.005	0.000
SC	0.000	0.006	0.000	0.000	-0.005	0.000
GA	0.000	0.019	0.000	0.000	-0.013	0.000
FL	0.000	0.080	0.000	0.000	-0.138	0.000
KY	0.000	-0.015	0.000	0.000	0.012	0.000
TE	0.000	0.197	0.000	0.000	-0.178	0.000
AL	0.000	0.176	0.000	0.000	-0.150	0.000
MS	0.000	-0.068	0.000	0.000	0.065	0.000
AR	0.000	-0.069	0.000	0.000	0.061	0.000
LA	0.000	-0.061	0.000	0.000	0.032	0.000
OK	0.000	0.048	0.000	0.000	-0.041	0.000
TX	0.000	0.002	0.000	0.000	0.030	0.000
MT	0.000	0.000	-0.099	0.000	0.000	-0.005
ID	0.000	0.000	-0.104	0.000	0.000	0.076
WY	0.000	0.000	0.169	0.000	0.000	-0.281
CO	0.000	0.000	-0.013	0.000	0.000	-0.004
NM	0.000	0.000	-0.103	0.000	0.000	0.050
AZ	0.000	0.000	0.138	0.000	0.000	-0.093
UT	0.000	0.000	0.108	0.000	0.000	-0.083
NV	0.000	0.000	0.180	0.000	0.000	-0.117
WA	0.000	0.000	0.018	0.000	0.000	-0.005
OR	0.000	0.000	-0.017	0.000	0.000	-0.010
CA	0.000	0.000	0.167	0.000	0.000	-0.060
AK	0.000	0.000	-1.578	0.000	0.000	1.871
HA	0.000	0.000	0.005	0.000	0.000	-0.001

Not surprisingly, DC has a large effect on the dummy indicator for AREA=South, and on the interaction between SPEND and AREA=South. AK has a large effect on the dummy indicator for AREA=West and on the interaction between SPEND and AREA=West. Let's look at the Cook's distances.

```

> COOKS <- cooks.distance( result )
> round( COOKS, 3 )
    ME     NH     VT     MA     RI     CT     NY     NJ     PA     OH     IN     IL     MI
  0.022  0.007  0.017  0.000  0.020  0.009  0.000  0.138  0.000  0.004  0.022  0.027  0.066
    WI     MN     IA     MO     ND     SD     NB     KA     DE     MD     DC     VA     WV
  0.002  0.012  0.005  0.000  0.002  0.044  0.005  0.006  0.135  0.000  0.545  0.002  0.001
    NC     SC     GA     FL     KY     TE     AL     MS     AR     LA     OK     TX     MT
  0.000  0.000  0.000  0.019  0.000  0.018  0.018  0.002  0.003  0.007  0.001  0.006  0.041
    ID     WY     CO     NM     AZ     UT     NV     WA     OR     CA     AK     HA
  0.007  0.082  0.001  0.013  0.014  0.007  0.025  0.001  0.003  0.046  1.998  0.000

> # find the 50th percentile of the F distribution for reference
> qf( .5, p, n-p )
[1] 0.904839

```

The value of Cook's distance for DC (0.545) is not so unusual, but the value for Alaska (1.998) is huge.

Now let's re-fit the model without AK, without DC, and without both to see how the results change.

```

> AK <- dimnames(pay)[[1]] == "AK"
> DC <- dimnames(pay)[[1]] == "DC"

> # fit model without AK
> summary( lm( PAY ~ AREA + SPEND + AREA:SPEND, data = pay[ !AK, ] ) )

Call:
lm(formula = PAY ~ AREA + SPEND + AREA:SPEND, data = pay[!AK,
])

Residuals:
    Min      1Q      Median      3Q      Max
-3320.5 -1370.1   -180.8    889.5   6162.0

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 10674.7726  2448.1063   4.360  7.7e-05 ***
AREASouth   -1089.5255  3450.2634  -0.316   0.7537

```

```

AREAWest      8656.3279  3744.3801   2.312   0.0255 *
SPEND         3.5249    0.6155    5.727  8.5e-07 ***
AREASouth:SPEND 0.5396    0.9507    0.568   0.5732
AREAWest:SPEND -1.9613    0.9920   -1.977   0.0543 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 2194 on 44 degrees of freedom
 Multiple R-Squared: 0.6313, Adjusted R-squared: 0.5895
 F-statistic: 15.07 on 5 and 44 DF, p-value: 1.284e-08

```

> # fit model without DC
> summary( lm( PAY ~ AREA + SPEND + AREA:SPEND, data = pay[ !DC, ] ) )

```

Call:

```

lm(formula = PAY ~ AREA + SPEND + AREA:SPEND, data = pay[!DC,
])

```

Residuals:

Min	1Q	Median	3Q	Max
-3758.6	-1368.4	-74.2	1364.2	6162.0

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10674.7726	2416.0153	4.418	6.40e-05 ***
AREASouth	2622.0912	3748.2377	0.700	0.488
AREAWest	3950.5552	2943.1076	1.342	0.186
SPEND	3.5249	0.6074	5.803	6.57e-07 ***
AREASouth:SPEND	-0.7120	1.0767	-0.661	0.512
AREAWest:SPEND	-0.5821	0.7276	-0.800	0.428

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2165 on 44 degrees of freedom
 Multiple R-Squared: 0.7352, Adjusted R-squared: 0.7051
 F-statistic: 24.43 on 5 and 44 DF, p-value: 1.096e-11

```

> # fit model without DC or AK
> summary( lm( PAY ~ AREA + SPEND + AREA:SPEND, data = pay[!(DC|AK), ] ) )

```

Call:

```

lm(formula = PAY ~ AREA + SPEND + AREA:SPEND, data = pay[!(DC |
)

```

```

AK), ])

Residuals:
    Min      1Q   Median      3Q      Max
-3038.21 -1378.63   -24.35   890.81  6161.98

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10674.7726  2317.5136   4.606 3.64e-05 ***
AREASouth    2622.0912  3595.4210   0.729  0.4698
AREAWest     8656.3279  3544.6385   2.442  0.0188 *
SPEND        3.5249    0.5827   6.050 3.10e-07 ***
AREASouth:SPEND -0.7120    1.0328  -0.689  0.4943
AREAWest:SPEND -1.9613    0.9390  -2.089  0.0427 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2077 on 43 degrees of freedom
Multiple R-Squared: 0.6078,    Adjusted R-squared: 0.5622
F-statistic: 13.33 on 5 and 43 DF,  p-value: 7.36e-08

```

If we remove AK, the dummy code for AREA=West becomes significant at the .05 level, and the interaction between AREA=West and SPEND becomes nearly significant. Removing DC, there is not much change. Removing both, the main effect and interaction for AREA=West are both significant.

Finally, let's summarize the effects of AREA and SPEND by plotting the estimated regression line for each region.

```

> # coefficients estimated from the full dataset
> betahat <- lm( PAY ~ AREA + SPEND + AREA:SPEND, data = pay )$coef
> betahat
            (Intercept)      AREASouth      AREAWest       SPEND AREASouth:SPEND
              10674.7726223 -1089.5255410  3950.5552127    3.5249200      0.5395513
          AREAWest:SPEND
              -0.5821196

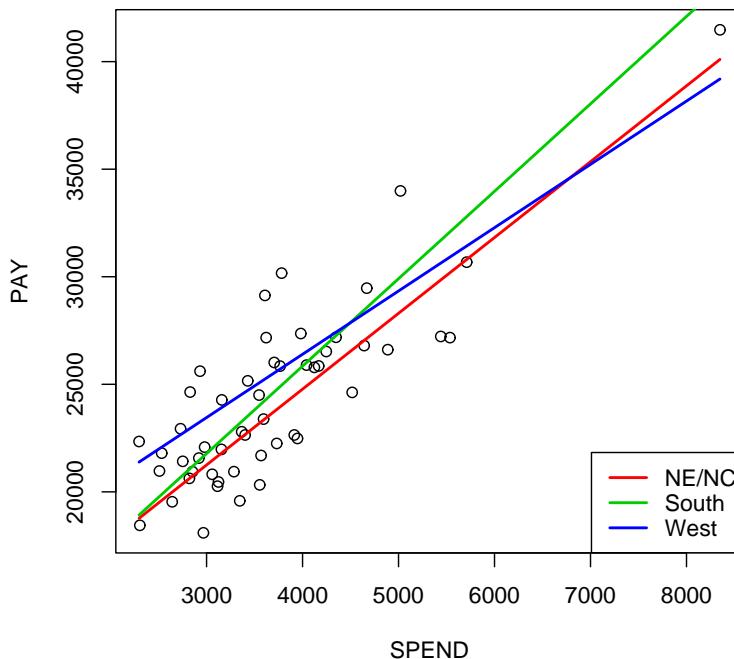
```

```

> spend.grid <- seq( from=min(pay$SPEND), to=max(pay$SPEND), length=200)
> pred.NENC <- betahat[1] + betahat[4] * spend.grid
> pred.South <- (betahat[1] + betahat[2]) +
+   (betahat[4] + betahat[5]) * spend.grid
> pred.West <- (betahat[1] + betahat[3]) +
+   (betahat[4] + betahat[6]) * spend.grid

> plot( PAY ~ SPEND, data=pay)
> lines( spend.grid, pred.NENC, lwd=2, col=2)
> lines( spend.grid, pred.South, lwd=2, col=3)
> lines( spend.grid, pred.West, lwd=2, col=4)
> legend( x="bottomright", legend=c("NE/NC", "South", "West"),
+   col=c(2,3,4), lty=c(1,1,1), lwd=c(2,2,2) )

```



```

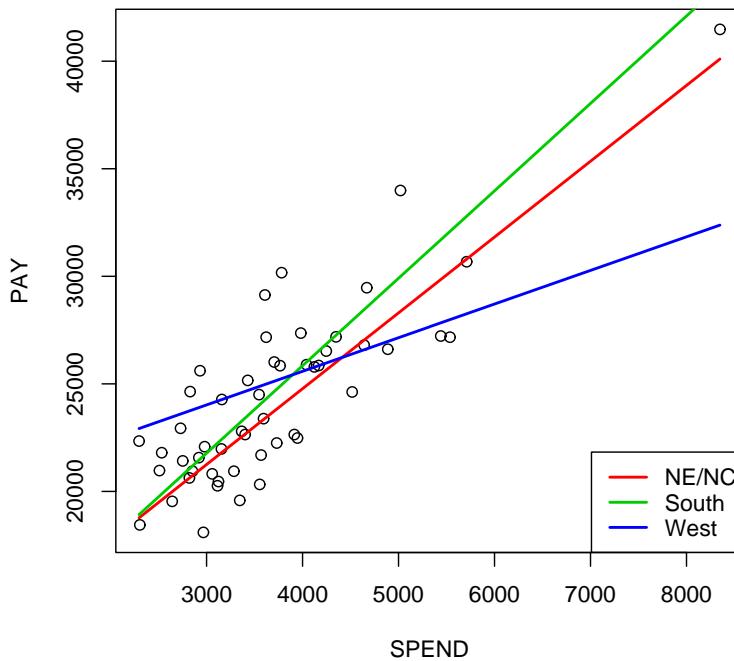
> # repeat with AK removed from the model fit
> betahat <- lm( PAY ~ AREA + SPEND + AREA:SPEND, data = pay[!AK,] )$coef
> betahat
      (Intercept)      AREASouth      AREAWest      SPEND      AREASouth:SPEND
10674.7726223 -1089.5255410     8656.3279379      3.5249200      0.5395513
      AREAWest:SPEND

```

-1.9613333

```
> spend.grid <- seq( from=min(pay$SPEND), to=max(pay$SPEND), length=200)
> pred.NENC <- betahat[1] + betahat[4] * spend.grid
> pred.South <- (betahat[1] + betahat[2]) +
+   (betahat[4] + betahat[5]) * spend.grid
> pred.West <- (betahat[1] + betahat[3]) +
+   (betahat[4] + betahat[6]) * spend.grid

> plot( PAY ~ SPEND, data=pay)
> lines( spend.grid, pred.NENC, lwd=2, col=2)
> lines( spend.grid, pred.South, lwd=2, col=3)
> lines( spend.grid, pred.West, lwd=2, col=4)
> legend( x="bottomright", legend=c("NE/NC", "South", "West"),
+   col=c(2,3,4), lty=c(1,1,1), lwd=c(2,2,2) )
```



Clearly, AK is a special case; it does not follow the model for the rest of the western states. If I had to select one model to publish about these data, this is what I would

do: Fit the model with AREA, SPEND and AREA:SPEND, but also include a dummy indicator for AK. This will effectively remove AK from the estimation of the other parameters.

```
> # create a dummy indicator for AK and include it in the model
> pay$AK <- factor(AK)
> contrasts( pay$AK )
    TRUE
    FALSE      0
    TRUE      1
> result <- lm( PAY ~ AREA + SPEND + AREA:SPEND + AK, data = pay )
> summary( result )

Call:
lm(formula = PAY ~ AREA + SPEND + AREA:SPEND + AK, data = pay)

Residuals:
    Min      1Q  Median      3Q     Max 
-3320.5 -1351.8 -124.1   888.2  6162.0 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10674.7726  2448.1063   4.360  7.7e-05 ***
AREASouth   -1089.5255  3450.2634  -0.316   0.7537    
AREAWest     8656.3279  3744.3801   2.312   0.0255 *  
SPEND        3.5249     0.6155   5.727  8.5e-07 ***
AKTRUE       9094.5141  4375.9723   2.078   0.0435 *  
AREASouth:SPEND  0.5396     0.9507   0.568   0.5732    
AREAWest:SPEND -1.9613     0.9920  -1.977   0.0543 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2194 on 44 degrees of freedom
Multiple R-Squared:  0.7576,    Adjusted R-squared:  0.7245 
F-statistic: 22.92 on 6 and 44 DF,  p-value: 4.698e-12
```

MODEL SELECTION, PART I

Lectures 26–27 cover topics that are found in Chapter 9 of KNNL.

Given a potentially large pool of predictor variables, which ones should we include in our regression model? Many strategies have been proposed. Before describing the methods, however, we will first discuss some of the underlying principles of good modeling.

The principle of parsimony. If we suppose that the “correct” model is a regression of the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \text{error}$$

for some set of predictors X_1, X_2, \dots and some real-valued coefficients β_0, β_1, \dots , then the most general model—the one that includes all of the potential predictors—will be a correct model. If some predictors are unnecessary because their true coefficients are zero, then including those predictors does not invalidate the model, because the model still holds if one or more coefficients are zero.

Including unnecessary predictors does not invalidate the model. But it does make the model needlessly

complicated, which is not a good thing.

Scientists often appeal to the principle known as “Ockham’s razor,” which is named after the 14th-century English philosopher William of Ockham. The principle is this: Given two theories that describe a phenomenon equally well, we should prefer the theory that is simpler. In statistical terms, this means that, if two models give nearly the same fit to the data, then we should prefer the model with fewer parameters. I say “nearly as well” because in practice, one can always improve the fit of a model by making it more complicated. The important question is: Does the improvement in fit that results from adding more parameters justify the extra complexity?

Many data analysts routinely discard predictors whose coefficients are not significantly different from zero at some prespecified level (say, $\alpha = .05$ or $\alpha = .10$). This strategy may be reasonable in some circumstances. But we should realize that Ockham’s razor is not absolute. Ockham’s razor is built on a belief or hope that the true laws of nature are simple. Sometimes they are, but sometimes they aren’t. Moreover, we build regression models for many different reasons. Sometimes we are trying to develop scientific understanding about the variables that influence Y , and we are deeply interested in the interpretation of the coefficients. In those situations, models with fewer predictors can be very appealing. At other times, we may simply be trying to predict unseen

future observations of Y , and the meaning of the β_j 's is irrelevant.

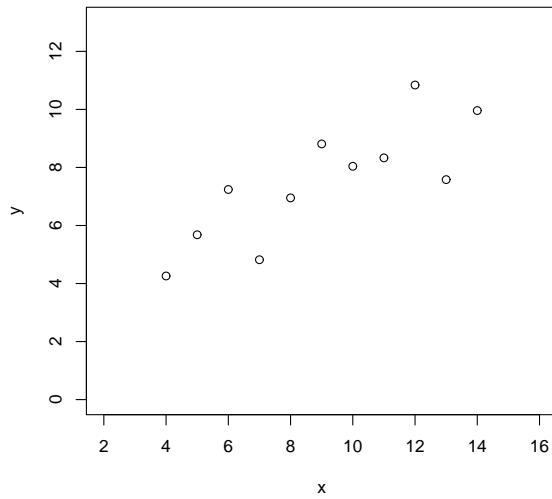
Another issue in trying to build a parsimonious model is that, when the potential predictors are intercorrelated, the significance of any variable may depend on which other variables are in the model. Fitting the full model with all predictors, and then discarding at once all of the ones that do not seem important, may not us to a good model.

Procedures for model selection based on statistical significance need to take into account the interdependence among the predictors.

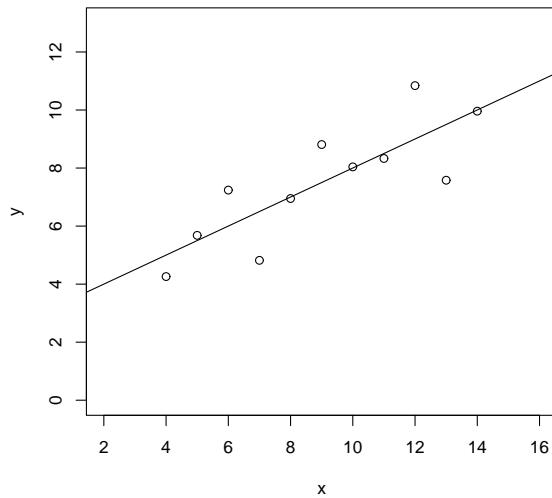
For evaluating regression models, statisticians have proposed a variety of **penalized fit criteria**. These can be viewed as implementations of Ockham's razor. They are measures of fit (functions of R^2) with a penalty for each additional parameter. They include adjusted R^2 , S^2 , AIC (Akaike's information criterion) and BIC (Bayesian information criterion).

The tradeoff between describing the current data and predicting future data. A good regression model should give predictions that are as accurate as possible, not just for the current dataset, but for future observations drawn from the same population. Increasing the complexity of a model may improve its fit to the current data but harm predictions for future data.

For example, consider this bivariate sample of $n = 11$ observations. (This is one of the four famous bivariate datasets published by Anscombe in 1973.)

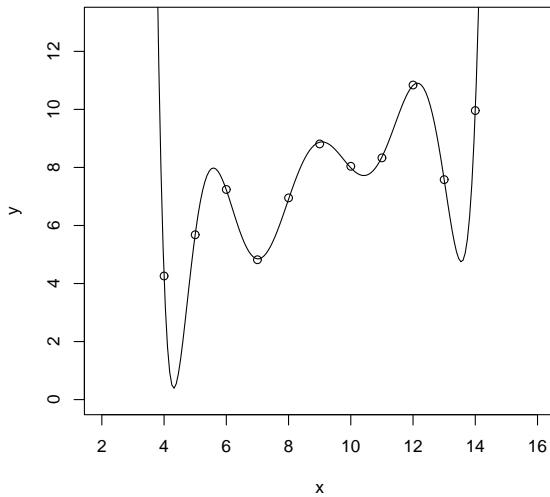


Many of us would describe the relationship by a linear least-squares fit.



The linear fit gives $R^2 = .67$ and thus explains 67% of the

variance in Y . But here's another way to describe the data.



This curve shown above is a 10th degree polynomial that fits the data perfectly ($R^2 = 1$). If you increase the number of predictors to $p = n$, and if the design matrix still has full rank, you will get a perfect fit to the observed data. But this model is overly complicated and the trends it describes are not real. A model that fits the current data so closely that it gives poor predictions for future datasets is said to be **overfitted**. An overfitted model describes idiosyncrasies of the current data that are unlikely to be found in other samples from the population.

Cross-validation and PRESS. We are usually given only one sample from the population, so we never really know how well a model will predict for future observations. But we can get some idea by the technique of **cross validation**. Cross-validation means that we divide the

data into two parts:

- a training sample, which we use to estimate the parameters, and
- a validation sample, which we use to evaluate the model's predictions.

Suppose, for example, that we randomly divide the sample in half, and use the first half to fit various models and the second half to evaluate them. If we fit the models by ordinary least squares, it makes sense to evaluate them by summing $(y_i - \hat{y}_i)^2$ over the validation sample, where \hat{y}_i is the predicted value based on the coefficients estimated from the training sample. When we identify a model that predicts well for the validation sample, we can combine the samples again and re-estimate the model parameters from the full dataset.

Sometimes data analysts actually do this. However, the procedure is somewhat arbitrary and subjective. Why should we reserve 50% of the sample for model fitting and 50% for evaluation? Why shouldn't we use some other proportions (e.g., 60% and 40%)? Moreover, two analysts who follow the same procedure will often be led to different answers, because they will use different random splits. There are

$$\binom{n}{n^*} = \frac{n!}{n^*(n - n^*)!}$$

ways to split a sample of size n into subsamples of size n^* and $n - n^*$. If this number of possible splits is not too large, we may be able to repeat the model fitting and evaluation procedure with all possible splits, which eliminates the randomness from the answer. This is the idea behind the PRESS (Prediction Sum of Squares) criterion. The PRESS statistic, which equals the sum of the squared PRESS residuals, is the total error of prediction over all possible splits with $n^* = n - 1$. We have already used the PRESS statistic for evaluating the performance of various polynomial and spline models (Lecture 21). The PRESS statistic is a very sensible criterion for evaluating the performance of candidate models.

The tradeoff between bias and variance. If a regression model fails to include a predictor whose true coefficient is not zero, then predictions based on that model will be biased. Therefore, for purposes of reducing bias, we want to include as many predictors in the model as possible. On the other hand, including a predictor whose true coefficient is zero reduces precision by increasing the variance of prediction. If a coefficient in a population is truly zero, it is usually better to set it to zero than to estimate it. (There are exceptions to this rule. For example, the analysis of covariance (ANCOVA) for designed experiments is one such exception. We will learn about ANCOVA in Stat 512. But, in most other

circumstances, it is better to set a zero coefficient to zero than to estimate it.)

In populations of real data, coefficients are never exactly zero. But if the coefficient is close to zero, then the gain in precision that results from removing the predictor from the model may outweigh the bias that we incur. Another popular criterion for model selection, called Mallow's C_p statistic, attempts to explicitly evaluate this tradeoff between bias and variance. This method looks for a model with a low estimated value of

$$\text{variance} + \text{bias}^2$$

over all the sample observations.

The concept of “nested models.” Before proceeding, we need to discuss the concept of nested models. Two models are said to be nested if one model is a special case of the other. For example, consider these two models,

$$\text{Model A: } Y = \beta_0 + \beta_1 X_1 + \text{error},$$

$$\text{Model B: } Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \text{error}.$$

Model A is a special case of Model B with $\beta_2 = 0$, so we say that Model A is nested within Model B. But Model A is not nested within

$$\text{Model C: } Y = \beta_0 + \beta_2 X_2 + \text{error},$$

because Model C does not reduce to Model A for any

choice of β_0 and β_2 .

A smaller model is obviously nested within a larger model if the smaller model is obtained by removing some predictors from the larger model. But there are other situations where models are nested even though we cannot get one from the other by omitting predictors. For example, consider a three-level categorical predictor C . Suppose Model A includes an intercept term and just one dummy variable for $C = 1$. And suppose Model B includes an intercept term and two effect codes for C . We cannot reduce Model B to Model A by eliminating one of the effect codes. But Model B is equivalent to a model with an intercept term, a dummy indicator for $C = 1$, and a dummy indicator for $C = 2$, so clearly Model A is nested within Model B. In fact, one can show that Model B reduces to model A if we impose a linear constraint on the coefficients in Model B.

To cover situations like these, we can say that linear regression models are nested if **the linear space spanned by the columns of one model's design matrix is contained in the linear space spanned by the columns of the other model's design matrix**. If X_A and X_B are the design matrices for two models, and if $\mathcal{R}(X_A) \subset \mathcal{R}(X_B)$, then Model A is nested within Model B.

The concept of nesting is important for the following reason. If Model A is nested within Model B, then we can formally test the null hypothesis that Model A is true,

versus the alternative hypothesis that Model B is true, in a standard way. That is, we can summarize the evidence against Model A in favor of Model B by a p-value from an F-test.

If two models are not nested, we cannot compare them by a usual F-test. To compare non-nested models, we need to use some other criterion (e.g., adjusted R^2 , AIC, PRESS or C_p).

Variable selection algorithms. Suppose we have two potential predictors, X_1 and X_2 , and suppose that we limit our attention to models without interactions. We can easily fit the four models

$$\text{Model A: } Y = \beta_0 + \text{error},$$

$$\text{Model B: } Y = \beta_0 + \beta_1 X_1 + \text{error},$$

$$\text{Model C: } Y = \beta_0 + \beta_2 X_2 + \text{error},$$

$$\text{Model D: } Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \text{error}$$

and compare them. Model A is nested within B, C, and D, whereas Models B and C are nested within D, so those comparisons can be made by standard F-tests. Models B and C can be compared in some other way (e.g., by looking at R^2 , which is appropriate for comparing non-nested models of the same size). By fitting all possible models, we can easily find a model that seems best by whatever criterion we choose. The practice of fitting all possible models is called **all-subsets regression**.

All-subsets regression is a good idea if we can do it. With p potential predictors, however, there are 2^p possible models to consider, which can be huge. Fitting all possible models is often impractical. For this reason, analysts often rely on sequential model selection algorithms. One technique is called **forward selection**. In forward selection, we start with a model that has no predictors. Then we search for the predictor that, if added to the model, would give the best improvement in fit. (“Best improvement” can be measured by the largest F statistic, the smallest p-value, or some other criterion.) If that improvement exceeds some pre-specified threshold, then we include that predictor in the model. Then we search among the remaining predictors to find the one that gives the greatest improvement in fit, and include it if the improvement exceeds the threshold. We continue in this fashion until there are no other predictors that would improve the model beyond the threshold.

Another possible method is **backward elimination**. We start with the full model and eliminate the least important predictor provided that the reduction in fit is no worse than some pre-specified threshold. We continue eliminating predictors one at a time until there are no other predictors can be discarded.

Forward selection and backward elimination are both problematic when the predictors are correlated, because the significance of variables in the model changes as

variables are added and removed. As we bring predictors into a model by forward selection, the other variables in the model may become insignificant. As we remove predictors from a model by backward elimination, other variables that have already been omitted may become significant. Forward selection and backward elimination may not eventually lead to a good model.

For this reason, many analysts prefer a combination procedure known as **stepwise regression**. In stepwise regression, we start with any model. We check the significance of each variable that is currently out of the model, and enter the most significant one if its contribution to the fit exceeds a threshold. (If significance is measured by an F-statistic, then the threshold is sometimes called “ F_{in} .”) After entering a variable, we then check the significance of each variable that is currently in the model, and discard the least significant one if its contribution falls below a threshold (e.g., “ F_{out} ”). We repeat until no more variables can be entered or removed. Note that we need to have $F_{in} > F_{out}$, otherwise we get caught in an infinite loop. If we start with the no-predictors model and take $F_{out} = 0$, the procedure becomes forward selection. If we begin with the full model and take $F_{in} = \infty$, it becomes backward elimination.

Stepwise regression has been implemented in many popular software packages, including Minitab, SAS and R. The decisions to include or exclude variables are

sometimes based on F statistics, and sometimes they are based on p -values. (The two are not quite the same, because the critical values of F vary at each step, since the degrees of freedom are varying.) Sometimes the decisions are based on AIC or C_p . Most packages will allow you to select the criteria for entering and removing predictors. They may also allow you to specify the starting model. They may allow you to force some variables to be in the model regardless of their significance. And they may allow you to group together subsets of predictors so that they will be included or excluded as a group. Grouping makes sense if you have a set of dummy codes, effect codes or contrasts that distinguish among the levels of a categorical predictor. Finally, if a stepwise procedure is truly sensible, it may allow you to consider interactions while restricting attention to models that obey the hierarchy principle (always include the main effects if interactions are present).

Even with all these options and safeguards, stepwise regression is not guaranteed to find the “best” model.

When the number of potential predictors is large, the final answer may depend heavily on where you start. Many good models may never be reached by the procedure.

In R, stepwise regression can be carried out automatically by the function `step()`. This function uses a selection criterion based on AIC. So we will discuss `step()` later, after we learn about AIC.

Penalized fit criteria. R^2 measures the proportion of variance in Y explained by the regression model. It can be written as

$$R^2 = 1 - \frac{SS_{Err}}{SS_{Tot}}. \quad (1)$$

Adding more predictors to the model always causes SS_{Err} to go down and R^2 to go up. R^2 eventually reaches 1.00 when $p = n$. R^2 may be appropriate for comparing two non-nested models with the same value of p , but it is not appropriate for comparing models with different p 's.

One popular modification to this statistic replaces the sums of squares in (1) with mean squares. The result is called adjusted R^2 ,

$$\begin{aligned} R_{adj}^2 &= 1 - \frac{MS_{Err}}{MS_{Tot}} \\ &= 1 - \frac{SS_{Err}/(n-p)}{SS_{Tot}/(n-1)}. \end{aligned}$$

Adding a predictor that explains little of the remaining variance in Y may cause R_{adj}^2 to decrease. Given a set of candidate models, choosing the model with the highest value of R_{adj}^2 is equivalent to choosing the model with lowest $MSE = S^2$.

We can think of R_{adj}^2 as a measure of fit with a penalty for additional parameters. This type of statistic is called a penalized fit criterion. Another popular measure is Akaike's Information Criterion (AIC). In general, AIC for

a parametric model is defined as

$$\text{AIC} = -2 \log L + 2p,$$

where L is the value of the likelihood function achieved at the maximum-likelihood estimate, and p is the number of parameters. A model that fits the data well should have a low value of $-2 \log L$. But we can always improve the fit by introducing more parameters, and the term $2p$ can be regarded as a penalty that favors simpler models over more complicated ones. Models with low AIC are desirable.

The penalty term $2p$ in AIC is motivated by work on the theory of information and entropy by the Japanese statistician Akaike in the early 1970's. Schwarz (1978) proposed a different penalty based on Bayesian arguments. His measure, known as the Bayesian information criterion or BIC, has the general form

$$\text{BIC} = -2 \log L + p \log n.$$

For a linear regression model

$$y_i \sim N(x_i^T \beta, \sigma^2),$$

the contribution to the likelihood function of the single observation y_i is

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y_i - x_i^T \beta)^2}{2\sigma^2} \right\},$$

the likelihood function is

$$L = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y_i - x_i^T \beta)^2}{2\sigma^2} \right\},$$

and -2 times the loglikelihood function is

$$-2 \log L = n \log(2\pi) + n \log \sigma^2 + \frac{1}{\sigma^2} \sum_{i=1}^n (y_i - x_i^T \beta)^2.$$

The ML estimates are $\hat{\beta}$ and $\hat{\sigma}^2 = SS_{Err}/n$. Substituting these into the above expression, we get

$$-2 \log L = n \log(2\pi) + n \log \left(\frac{SS_{Err}}{n} \right) + n.$$

It is customary to drop the constant terms involving π and n , because these do not change from one model to another. The loglikelihood becomes

$$-2 \log L = n \log \left(\frac{SS_{Err}}{n} \right),$$

and the measures of fit become

$$\text{AIC} = n \log \left(\frac{SS_{Err}}{n} \right) + 2p,$$

$$\text{BIC} = n \log \left(\frac{SS_{Err}}{n} \right) + p \log n.$$

For both of these measures, smaller is better.

MODEL SELECTION, PART II

Review. In the last lecture, we covered a variety of model selection criteria. Unlike the usual F tests, these measures may all be used to compare non-nested models.

- The ordinary R^2 , defined as

$$R^2 = 1 - \frac{SS_{Err}}{SS_{Tot}}.$$

This is appropriate for comparing models with the same number of coefficients p .

- The adjusted R^2 , defined as

$$R_{adj}^2 = 1 - \frac{SS_{Err}/(n-p)}{SS_{Tot}/(n-1)}.$$

This is a decreasing function of $MSE = SS_{Err}/(n-p)$, so choosing the model with the highest R_{adj}^2 is equivalent to choosing the model with the smallest MSE.

- The PRESS statistic, defined as

$$\text{PRESS} = \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2,$$

which is related to the concept of cross-validation.
Smaller values of PRESS are better.

- Akaike's information criterion,

$$\text{AIC} = n \log \left(\frac{SS_{Err}}{n} \right) + 2p,$$

which is equivalent to $-2 \log L + 2p$. Smaller values of AIC are better.

- The Bayesian information criterion,

$$\text{BIC} = n \log \left(\frac{SS_{Err}}{n} \right) + p \log n,$$

which is equivalent to $-2 \log L + p \log n$. Smaller values of BIC are better.

More about AIC and BIC. Note that if $\log n > 2$, which happens if $n > 7$, then BIC increases with p more quickly than AIC. Thus BIC tends to favor smaller models than AIC. Both of these criteria have a theoretical basis, and each of them has fans who support its use. In my limited experience, however, statisticians who actually use these measures tend to favor AIC over BIC. Many believe that BIC tends penalizes the additional parameters too much, especially when n is large.

Likelihood theory indicates that, if we add an unnecessary predictor to a model, then $2 \log L$ will increase by a random amount distributed approximately as χ_1^2 . The χ_1^2 distribution has a mean of 1, and $P(\chi_1^2 \leq 2) = 0.84$.

Therefore, AIC should go down if the hypothesis test for the significance of the additional predictors has a p-value of 0.16 or less. Therefore, comparing two nested models and choosing the one with the smaller AIC is nearly the same thing as keeping the smaller model unless the hypothesis test is significant at the 0.16 level.

Today we will cover one more criterion and illustrate with an example.

Mallows predictive criterion C_p . This measure attempts to quantify the tradeoff between bias and variance.

Let μ_i denote the true mean of observation y_i . (This should be interpreted as the conditional mean of y_i given all of the predictors in a pool of potential predictors.)

Under the model $y_i \sim N(x_i^T \beta, \sigma^2)$, the true value of μ_i is $x_i^T \beta$, and $\hat{y}_i = x_i^T \hat{\beta}$ is an unbiased estimate of it. But suppose that we don't know what to include in x_i . We need to consider the dangers of underfitting versus overfitting.

First, let's quantify the bias that comes from underfitting. Suppose we partition the design matrix into two parts, $X = [X_1, X_2]$, and partition the coefficients as well,

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix},$$

so that

$$X\beta = X_1\beta_1 + X_2\beta_2.$$

Omitting X_2 from the model is equivalent to estimating the full β by

$$\hat{\beta} = \begin{bmatrix} (X_1^T X_1)^{-1} X_1^T y \\ 0 \end{bmatrix} \quad (1)$$

rather than $(X^T X)^{-1} X^T y$. If one or more elements of β_2 are nonzero, the fitted values under the reduced model,

$$\hat{y} = X\hat{\beta} = X_1(X_1^T X_1)^{-1} X_1^T y,$$

is a biased estimate of $E(y) = X\beta$. The bias is

$$\begin{aligned} \text{bias} &= E(X\hat{\beta}) - X\beta \\ &= E\left[X_1(X_1^T X_1)^{-1} X_1^T y\right] - [X_1\beta_1 + X_2\beta_2] \\ &= X_1(X_1^T X_1)^{-1} X_1^T [X_1\beta_1 + X_2\beta_2] \\ &\quad - [X_1\beta_1 + X_2\beta_2] \\ &= X_1\beta_1 + X_1(X_1^T X_1)^{-1} X_1^T X_2\beta_2 - X_1\beta_1 - X_2\beta_2 \\ &= (H_1 - I)X_2\beta_2, \end{aligned}$$

where $H_1 = X_1(X_1^T X_1)^{-1} X_1^T$.

If one or more elements of β_2 are nonzero but we estimate β by (1), then the usual estimate of σ^2 (the MSE) is also biased. We will not prove this result now, but one can

show that

$$E(S^2) = \sigma^2 + \left(\frac{1}{n - p_1} \right) \|\text{bias}\|^2, \quad (2)$$

where p_1 is the number of columns in X_1 , and “bias” is the bias vector that we just derived.

Now consider the impact of overfitting. Suppose now that the true value of β_2 is zero, but we estimate β by $\hat{\beta} = (X^T X)^{-1} X^T y$ rather than (1). Then

$$E(y) = X_1 \beta_1 + X_2 \beta_2 X \beta$$

is still true, so this $\hat{\beta}$ is an unbiased estimate of β , and $\hat{y} = X \hat{\beta}$ is an unbiased estimate of $E(y) = X \beta$, and S^2 is an unbiased estimate of σ^2 . However, one can show that this \hat{y} is a less precise estimate of $E(y)$ than

$$\hat{y}_1 = X_1 (X_1^T X_1)^{-1} X_1^T y = H_1 y.$$

It is less precise in the sense that the variance of each element of \hat{y} is greater than or equal to the variance of the corresponding element of \hat{y}_1 . In other words, if it is really true that $\beta_2 = 0$, then it is better to simply set β_2 to zero than to estimate it from the data.

The mean squared error of $\hat{y}_i = x_i^T \hat{\beta}$ as an estimate of μ_i is

$$\begin{aligned} MSE(\hat{y}_i) &= E \left[(\hat{y}_i - \mu_i)^2 \right] \\ &= V(\hat{y}_i - \mu_i) + (E(\hat{y}_i - \mu_i))^2 \\ &= V(\hat{y}_i) + (\text{bias of } \hat{y}_i)^2. \end{aligned}$$

(We are now using MSE in a different sense. This MSE is not an estimate of σ^2 , but a statistical measure of the error in a parameter estimate.) The sum of these MSE's for μ_1, \dots, μ_n is

$$\begin{aligned}\sum_{i=1}^n MSE(\hat{y}_i) &= \sum_{i=1}^n V(\hat{y}_i) + \sum_{i=1}^n (\text{bias of } \hat{y}_i)^2 \\ &= \text{tr } V(\hat{y}) + \|\text{bias of } \hat{y}\|^2.\end{aligned}$$

But $\hat{y} = Hy \sim N(X\beta, \sigma^2 H)$, so

$$\text{tr } V(\hat{y}) = \sigma^2 \text{tr } H = \sigma^2 p.$$

We have already asserted in (2) that

$$E(S^2) = \sigma^2 + \left(\frac{1}{n-p} \right) \|\text{bias of } \hat{y}\|^2,$$

where p is the number of coefficients in the current model, so

$$\|\text{bias of } \hat{y}\|^2 = (n-p) [E(S^2) - \sigma^2].$$

Putting the two pieces together, we get

$$\sum_{i=1}^n MSE(\hat{y}_i) = \sigma^2 p + (n-p) [E(S^2) - \sigma^2].$$

Finally, let's divide by σ^2 to get a scale-free measure,

$$\frac{\sum_{i=1}^n MSE(\hat{y}_i)}{p} = p + (n-p) \left[\frac{E(S^2) - \sigma^2}{\sigma^2} \right]. \quad (3)$$

Mallow's C_p is an estimate of (3). We can estimate $E(S^2)$ by S^2 from the current model, which we will call S_{curr}^2 . But if the model has been underfitted, S_{curr}^2 will not be a good estimate of σ^2 . The least biased estimate of σ^2 will be the S^2 from the “maximal” model, the model that includes all of the predictors you want to consider. If we estimate σ^2 by S_{max}^2 , the estimate from the maximal model, we get Mallows C_p statistic,

$$C_p = p + (n - p) \left[\frac{S_{cur}^2 - S_{max}^2}{S_{max}^2} \right].$$

- Models with small C_p are better than models with large C_p , in the sense that they appear to give more precise predictions of the response variable Y *at the sampled values of the predictors*.
- The C_p statistic for the maximal model is always p .
- If the variables not in the current model are unimportant, then S_{max}^2 and S_{cur}^2 are both unbiased estimates of σ^2 and should be close to each other, making $C_p \approx p$. A value of C_p that is close to p gives evidence that *the variables in the maximal model but not in the current model are unimportant*.
- Many authors suggest that you should select the smallest (most parsimonious) model for which Mallows C_p is fairly close to p . Others say that you should just select the model with the smallest C_p .

Example. Let's look at a real data example to see how our model selection criteria work in practice. The file `uscrime.txt` contains aggregate crime statistics for 47 states in 1960, along with some variables that may be related to crime rates. The variables are:

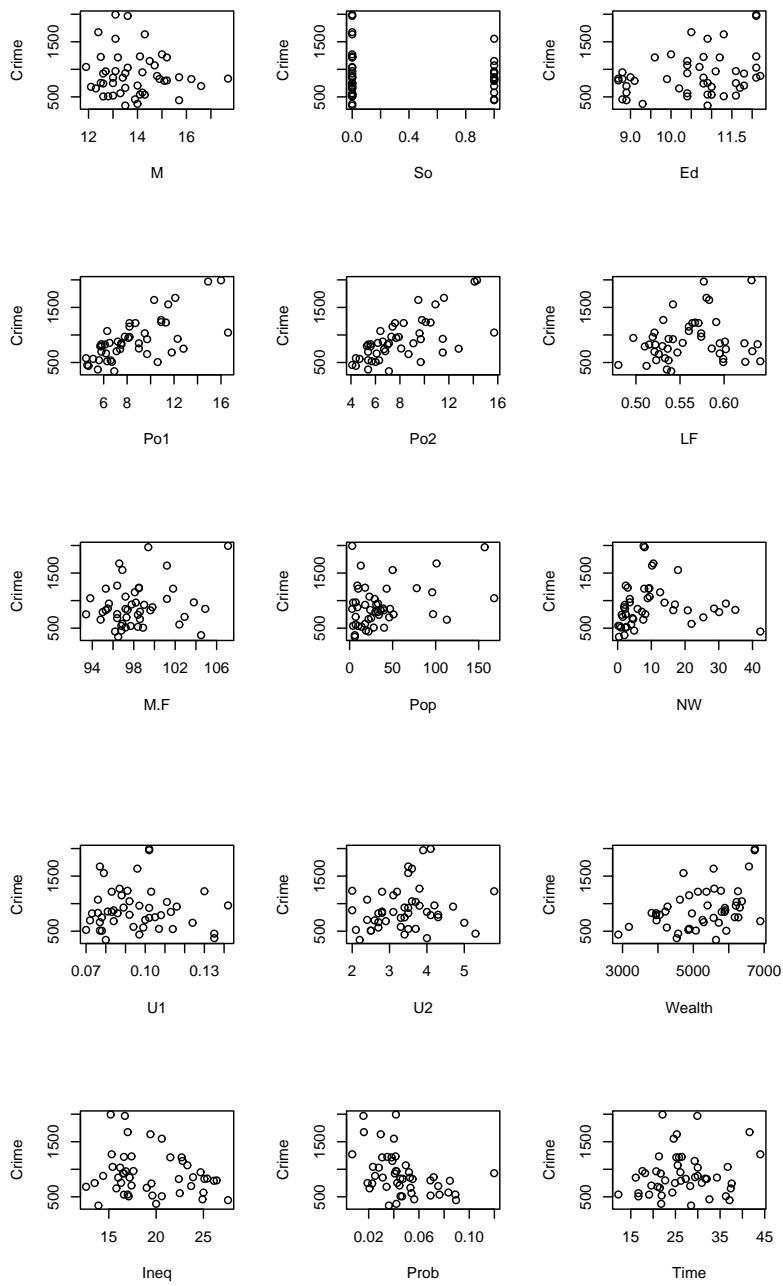
M	=	percentage of males aged 14–24
So	=	1=southern state, 0 otherwise
Ed	=	avg. years schooling among residents age 25+
Po1	=	per capita expenditure on police, 1960
Po2	=	per capita expenditure on police, 1959
LF	=	labor force participation rate of urban males 14–24
M.F	=	number of males per 100 females
Pop	=	state population in 1960 divided by 100,000
NW	=	percentage of nonwhites in the population
U1	=	unemployment rate of urban males 14–24
U2	=	unemployment rate of urban males 35–39
Wealth	=	median of transferable assets or family income
Ineq	=	income inequality (percentage of families earning below half the median income)
Prob	=	probability of imprisonment (number of commitments divided by number of offenses)
Time	=	average months served by offenders in state prisons before their first release
Crime	=	number of offenses per 100,000 population in 1960

The last variable is the response, and the remaining 15 variables are potential predictors. The first thing that we should do is to plot the response against each potential predictor.

```
> crime <- read.table("uscrime.txt", header=T)

> par(mfrow=c(3,3))
> plot( Crime ~ M, data=crime)
> plot( Crime ~ So, data=crime)
> plot( Crime ~ Ed, data=crime)
> plot( Crime ~ Po1, data=crime)
> plot( Crime ~ Po2, data=crime)
> plot( Crime ~ LF, data=crime)
> plot( Crime ~ M.F, data=crime)
> plot( Crime ~ Pop, data=crime)
> plot( Crime ~ NW, data=crime)

> plot( Crime ~ U1, data=crime)
> plot( Crime ~ U2, data=crime)
> plot( Crime ~ Wealth, data=crime)
> plot( Crime ~ Ineq, data=crime)
> plot( Crime ~ Prob, data=crime)
> plot( Crime ~ Time, data=crime)
```



With 15 potential predictors, there are $2^{15} = 32,768$ possible regression models. Let's fit all possible regressions (all subsets) and find the best candidate models using our various criteria. First, create the design matrix for the maximal model.

```
> x <- cbind( Const=1,
+   M=crime$M,
+   So=crime$So,
+   Ed=crime$Ed,
+   Po1=crime$Po1,
+   Po2=crime$Po2,
+   LF=crime$LF,
+   M.F=crime$M.F,
+   Pop=crime$Pop,
+   NW=crime$NW,
+   U1=crime$U1,
+   U2=crime$U2,
+   Wealth=crime$Wealth,
+   Ineq=crime$Ineq,
+   Prob=crime$Prob,
+   Time=crime$Time)

> y <- crime$Crime
```

Next, let's create a logical matrix with 2^{15} rows and 15 columns to indicate, for each of the possible models, which predictors are in or out.

```
> # logical matrix to record which variables are in/out of each model
> models <- matrix( F, 2^15, 15)
> dimnames(models) <- list( NULL,
+   c("M", "So", "Ed", "Po1", "Po2", "LF", "M.F", "Pop", "NW",
+   "U1", "U2", "Wealth", "Ineq", "Prob", "Time") )

> row <- 0
> for( a in c(F,T) ){
```

```

+   for( b in c(F,T) ){
+     for( c in c(F,T) ){
+       for( d in c(F,T) ){
+         for( e in c(F,T) ){
+           for( f in c(F,T) ){
+             for( g in c(F,T) ){
+               for( h in c(F,T) ){
+                 for( i in c(F,T) ){
+                   for( j in c(F,T) ){
+                     for( k in c(F,T) ){
+                       for( l in c(F,T) ){
+                         for( m in c(F,T) ){
+                           for( n in c(F,T) ){
+                             for( o in c(F,T) ){
+
+     row <- row + 1
+     models[row,] <- c(a, b, c, d, e, f, g, h, i, j, k, l, m, n, o)
+   }
+
> # look at first few rows
> models[1:4,]
      M   So   Ed   Po1   Po2    LF   M.F   Pop    NW    U1    U2 Wealth
[1,] FALSE FALSE
[2,] FALSE FALSE
[3,] FALSE FALSE
[4,] FALSE FALSE
      Ineq  Prob  Time
[1,] FALSE FALSE FALSE
[2,] FALSE FALSE TRUE
[3,] FALSE TRUE FALSE
[4,] FALSE TRUE TRUE

```

Now set up a matrix to hold the results, and compute the estimate of σ^2 for the maximal model (needed for C_p).

```

> # matrix to hold fit statistics
> results <- matrix( NA, 2^15, 8)
> dimnames(results) <- list( NULL,
+   c("p", "R2", "R2.adj", "MSE", "PRESS", "AIC", "BIC", "Cp") )
+
> # MSE for maximal model
> tmp <- lsfit( x, y, intercept=F)

```

```
> N <- nrow(crime)
> p <- ncol(x)
> MSE.max <- sum( tmp$res^2 ) / (N-p)
```

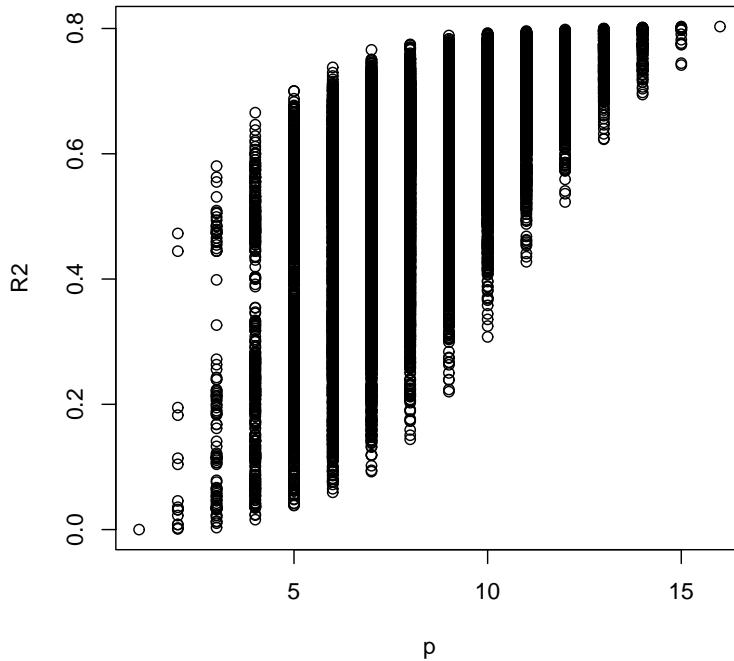
Now fit all possible models. On my computer, it took less than two minutes.

```
> for( i in 1:(2^15) ){
+   # pull out the ith row of models and append T for the intercept
+   which <- c( T, models[i,] )
+   # fit the model and compute the fit measures
+   tmp <- lsfit( x[,which], y, intercept=F )
+   p <- sum(which)
+   SSTot <- (N-1)*var(y)
+   MSTot <- var(y)
+   SSErr <- sum( tmp$res^2 )
+   MSE <- SSErr / (N-p)
+   R2 <- 1 - SSErr / SSTot
+   R2.adj <- 1 - MSE / MSTot
+   hi <- ls.diag(tmp)$hat # leverages
+   res.PRESS <- tmp$res / (1-hi)
+   PRESS <- sum( res.PRESS^2 )
+   AIC <- N*log(SSErr/N) + 2*p
+   BIC <- N*log(SSErr/N) + p*log(N)
+   Cp <- (SSErr/MSE.max) - N + 2*p
+   # save the results
+   results[i,1] <- p
+   results[i,2] <- R2
+   results[i,3] <- R2.adj
+   results[i,4] <- MSE
+   results[i,5] <- PRESS
+   results[i,6] <- AIC
+   results[i,7] <- BIC
+   results[i,8] <- Cp}
```

Summarizing these results can be challenging. KNNL suggest that you plot some of the measures against p , to identify the best model for any given number of predictors. For example, here's the plot of R^2 versus p .

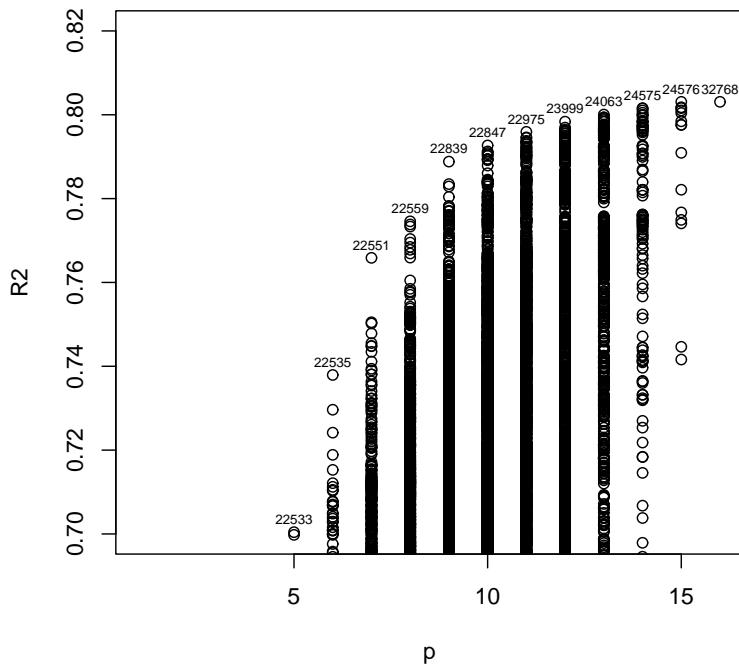
```
> p      <- results[,1]
> R2     <- results[,2]
> R2.adj <- results[,3]
> MSE    <- results[,4]
> PRESS   <- results[,5]
> AIC    <- results[,6]
> BIC    <- results[,7]
> Cp     <- results[,8]

> plot(p,R2)
```



Let's re-plot with different y-limits and identify the model with the highest R^2 for each p .

```
> plot(p,R2, ylim=c(.7,.82))
> identify(p,R2, cex=.6)
```



Now let's find the model with the highest adjusted R^2 .

```
> # model with best R2.adj
> i <- R2.adj == max( R2.adj )
> (1:2^15)[ i ]
[1] 22839
> round( results[i,], 3 )
      p        R2      R2.adj        MSE      PRESS      AIC
 9.000    0.789    0.744  38238.625 2287076.064  503.935
      BIC      Cp
 520.586   4.245
```

The model with the highest adjusted R^2 is the model # 22.839, which has $p = 9$. The predictors included in this model are:

```
> models[i,]
      M      So      Ed      Po1      Po2      LF      M.F      Pop      NW      U1      U2
TRUE FALSE TRUE TRUE FALSE FALSE TRUE FALSE FALSE TRUE TRUE
```

Wealth	Ineq	Prob	Time
FALSE	TRUE	TRUE	FALSE

Now identify the models with the lowest MSE, PRESS, AIC, BIC and C_p .

```

> # model with best MSE
> i <- MSE == min( MSE )
> (1:2^15)[ i ]
[1] 22839

> # model with best PRESS
> i <- PRESS == min( PRESS )
> (1:2^15)[ i ]
[1] 22839

> # model with best AIC
> i <- AIC == min( AIC )
> (1:2^15)[ i ]
[1] 22839

> # model with best BIC
> i <- BIC == min( BIC )
> (1:2^15)[ i ]
[1] 22551

> # model with best Cp
> i <- Cp == min( Cp )
> (1:2^15)[ i ]
[1] 22551

> round( results[i,], 3)
      P          R2       R2.adj        MSE        PRESS        AIC
    7.000     0.766     0.731  40276.421  2297102.447   504.786
      BIC        Cp
    517.737    3.860

> models[i,]
      M      So      Ed      Po1      Po2      LF      M.F      Pop      NW      U1      U2
    TRUE  FALSE   TRUE   TRUE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  TRUE
      Wealth   Ineq   Prob   Time
  
```

```
FALSE  TRUE  TRUE  FALSE
```

MSE, PRESS, and AIC all identify model #22,839 as the best model. BIC and C_p favor model #22,551, which has $p = 7$.

Both of these models have M, Ed, Po1, U2, Ineq, and Prob, but model #22,839 has M.F and U1, whereas #22,551 does not. These two models are nested, so let's test the smaller model against the larger one.

```
> null <- 22551
> alt  <- 22839

> df.null <- N - p>null]
> df.alt   <- N - p[alt]
> SSRes.null <- MSE>null] * df.null
> SSRes.alt  <- MSE[alt]  * df.alt

> Fnum <- (SSRes.null - SSRes.alt) / 2
> Fden <- MSE[alt]
> F <- Fnum / Fden
> F
[1] 2.065831

> 1 - pf(F, 2, df.alt)
[1] 0.1407100
```

The p-value is 0.14, which is just below the approximate cutoff of 0.16 used by AIC. Notice that with $N = 47$, $\log N = 3.85$ and $P(\chi_1^2 \leq \log N) = 0.95$. So in a dataset of $N = 47$, adding more predictors will drop BIC if they are significant at about the .05 level, which is more stringent.

Finally, let's see what happens if we use the stepwise

regression function **step**. This function adds and removes predictors one at a time based on AIC. At every cycle, the variable that reduces AIC the most is brought into the model, and the variable that raises AIC the most is kicked out. The iterations stop when adding or deleting one variable no longer drops the AIC.

```
> result <- step(
+   lm( Crime ~
+     M + So + Ed + Po1 + Po2 + LF + M.F + Pop +
+     NW + U1 + U2 + Wealth + Ineq + Prob + Time, data=crime) )
Start: AIC= 514.65
Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +
U2 + Wealth + Ineq + Prob + Time
```

	Df	Sum of Sq	RSS	AIC
- So	1	29	1354974	513
- LF	1	8917	1363862	513
- Time	1	10304	1365250	513
- Pop	1	14122	1369068	513
- NW	1	18395	1373341	513
- M.F	1	31967	1386913	514
- Wealth	1	37613	1392558	514
- Po2	1	37919	1392865	514
<none>		1354946		515
- U1	1	83722	1438668	515
- Po1	1	144306	1499252	517
- U2	1	181536	1536482	519
- M	1	193770	1548716	519
- Prob	1	199538	1554484	519
- Ed	1	402117	1757063	525
- Ineq	1	423031	1777977	525

```
Step: AIC= 512.65
Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
Wealth + Ineq + Prob + Time
```

	Df	Sum of Sq	RSS	AIC
- Time	1	10341	1365315	511
- LF	1	10878	1365852	511

- Pop	1	14127	1369101	511
- NW	1	21626	1376600	511
- M.F	1	32449	1387423	512
- Po2	1	37954	1392929	512
- Wealth	1	39223	1394197	512
<none>		1354974		513
- U1	1	96420	1451395	514
- Po1	1	144302	1499277	515
- U2	1	189859	1544834	517
- M	1	195084	1550059	517
- Prob	1	204463	1559437	517
- Ed	1	403140	1758114	523
- Ineq	1	488834	1843808	525

Step: AIC= 511.01

Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
Wealth + Ineq + Prob

	Df	Sum of Sq	RSS	AIC
- LF	1	10533	1375848	509
- NW	1	15482	1380797	510
- Pop	1	21846	1387161	510
- Po2	1	28932	1394247	510
- Wealth	1	36070	1401385	510
- M.F	1	41784	1407099	510
<none>		1365315		511
- U1	1	91420	1456735	512
- Po1	1	134137	1499452	513
- U2	1	184143	1549458	515
- M	1	186110	1551425	515
- Prob	1	237493	1602808	517
- Ed	1	409448	1774763	521
- Ineq	1	502909	1868224	524

Step: AIC= 509.37

Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + NW + U1 + U2 + Wealth +
Ineq + Prob

	Df	Sum of Sq	RSS	AIC
- NW	1	11675	1387523	508
- Po2	1	21418	1397266	508
- Pop	1	27803	1403651	508
- M.F	1	31252	1407100	508
- Wealth	1	35035	1410883	509

<none>		1375848	509
- U1	1	80954	1456802
- Po1	1	123896	1499744
- U2	1	190746	1566594
- M	1	217716	1593564
- Prob	1	226971	1602819
- Ed	1	413254	1789103
- Ineq	1	500944	1876792

Step: AIC= 507.77

Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + U1 + U2 + Wealth + Ineq +
Prob

	Df	Sum of Sq	RSS	AIC
- Po2	1	16706	1404229	506
- Pop	1	25793	1413315	507
- M.F	1	26785	1414308	507
- Wealth	1	31551	1419073	507
<none>		1387523		508
- U1	1	83881	1471404	509
- Po1	1	118348	1505871	510
- U2	1	201453	1588976	512
- Prob	1	216760	1604282	513
- M	1	309214	1696737	515
- Ed	1	402754	1790276	518
- Ineq	1	589736	1977259	522

Step: AIC= 506.33

Crime ~ M + Ed + Po1 + M.F + Pop + U1 + U2 + Wealth + Ineq +
Prob

	Df	Sum of Sq	RSS	AIC
- Pop	1	22345	1426575	505
- Wealth	1	32142	1436371	505
- M.F	1	36808	1441037	506
<none>		1404229		506
- U1	1	86373	1490602	507
- U2	1	205814	1610043	511
- Prob	1	218607	1622836	511
- M	1	307001	1711230	514
- Ed	1	389502	1793731	516
- Ineq	1	608627	2012856	521
- Po1	1	1050202	2454432	531

Step: AIC= 505.07

Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Wealth + Ineq + Prob

	Df	Sum of Sq	RSS	AIC
- Wealth	1	26493	1453068	504
<none>			1426575	505
- M.F	1	84491	1511065	506
- U1	1	99463	1526037	506
- Prob	1	198571	1625145	509
- U2	1	208880	1635455	509
- M	1	320926	1747501	513
- Ed	1	386773	1813348	514
- Ineq	1	594779	2021354	519
- Po1	1	1127277	2553852	530

Step: AIC= 503.93

Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob

	Df	Sum of Sq	RSS	AIC
<none>			1453068	504
- M.F	1	103159	1556227	505
- U1	1	127044	1580112	506
- Prob	1	247978	1701046	509
- U2	1	255443	1708511	510
- M	1	296790	1749858	511
- Ed	1	445788	1898855	515
- Ineq	1	738244	2191312	521
- Po1	1	1672038	3125105	538

> summary(result)

Call:

```
lm(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob,
   data = crime)
```

Residuals:

Min	1Q	Median	3Q	Max
-444.702	-111.070	3.031	122.145	483.298

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-6426.10	1194.61	-5.379	4.04e-06 ***
M	93.32	33.50	2.786	0.00828 **
Ed	180.12	52.75	3.414	0.00153 **

```
Po1           102.65      15.52   6.613 8.26e-08 ***
M.F            22.34      13.60   1.642  0.10874
U1          -6086.63    3339.27  -1.823  0.07622 .
U2            187.35      72.48   2.585  0.01371 *
Ineq           61.33      13.96   4.394 8.63e-05 ***
Prob          -3796.03    1490.65  -2.547  0.01505 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 195.5 on 38 degrees of freedom
Multiple R-Squared: 0.7888,     Adjusted R-squared: 0.7444
F-statistic: 17.74 on 8 and 38 DF,  p-value: 1.159e-10
```

Stepwise regression ended up with model #22,839, which happens to be the model with the lowest AIC among all 2^{15} possible models. That does not always happen. Sometimes there are better models that stepwise cannot reach.

One final comment about model selection. Model selection criteria can help you identify one or more candidate models that are the “best” among the pool of models being considered. But if a model is the best one among a pool of models, that does not mean that the model is correct, nor does it mean that the model is good. You should still check the model for heteroscedasticity, nonlinearity, outliers and highly influential observations.

GENERALIZED LINEAR MODELS: A LIKELIHOOD VIEW (PART I)

Motivating example: Dose-response experiment.

Five groups of animals were exposed to a dangerous substance in varying concentrations. Let n_i be the number of animals, y_i the number that died and $p_i = y_i/n_i$ the proportion that died in group i , $i = 1, \dots, 5$.

Concentration	\log_{10} conc	n_i	y_i	p_i
1×10^{-5}	-5	6	0	0.000
1×10^{-4}	-4	6	1	0.167
1×10^{-3}	-3	6	4	0.667
1×10^{-2}	-2	6	6	1.000
1×10^{-1}	-1	6	6	1.000

We will try to model π_i = probability of death as a function of \log_{10} conc. One obvious method is to regress the p_i 's on \log_{10} conc using OLS. This is a bad idea for two reasons.

- **Non-linearity:** a linear model may give predicted values of π_i outside $(0, 1)$
- **Heteroscedasticity:** the variance of p_i is $\pi_i(1 - \pi_i)/n_i$ which is not constant. We cannot fix this by weighted least squares because π_i is unknown.

Older textbooks may suggest the transformation $\sin^{-1} \sqrt{p_i}$, which makes the variance approximately proportional to $1/n_i$. This solves the second problem but not the first, and the resulting model is hard to interpret.

In modern statistical practice, we would use a maximum-likelihood approach based on a binomial model. Regression models in which the outcome has a binomial distribution are one of the most common types of generalized linear models.

Generalized linear models. Generalized linear models (GLIM's) were introduced by Nelder and Wedderburn (1972) and popularized by McCullagh and Nelder (1989). GLIM's extend normal linear regression model in two ways. First, they allow the response variable to come from a regular exponential family distribution. (Regular exponential families include some of the most commonly used statistical models, including the normal, binomial and Poisson.) Second, they allow the mean response to vary linearly with covariates through a monotonic transformation.

In a GLIM, we assume that the response variable y_i comes from an known exponential family with mean μ_i , and that

$$g(\mu_i) = x_i^T \beta$$

for a known monotonic function g called the **link function**. Therefore, to fit a GLIM, we need to specify

- the distributional family,
- the link function, and
- the covariates to be used as predictors.

The three most common GLIM's are:

- the normal linear regression model, which combines the normal distributional family with the identity link,

$$\begin{aligned} y_i &\sim N(\mu_i, \sigma^2), \\ \mu_i &= x_i^T \beta, \end{aligned}$$

- logistic regression, which combines the binomial family with a logistic link,

$$\begin{aligned} y_i &\sim \text{Bin}(n_i, \pi_i), \\ \log \left(\frac{\pi_i}{1 - \pi_i} \right) &= x_i^T \beta \end{aligned}$$

(for this model we must also specify n_i), and

- the loglinear model, which combines the Poisson family with a log link,

$$\begin{aligned} y_i &\sim \text{Poisson}(\mu_i), \\ \log \mu_i &= x_i^T \beta. \end{aligned}$$

Notice that the first model can be written as

$$y_i = x_i^T \beta + \epsilon_i,$$

where e_i is an error distributed as $N(0, \sigma^2)$. But the second and third models cannot be written as a prediction plus a binomial or Poisson error. A common tendency among novices who are accustomed to normal linear regression is to write the logistic model as

$$\log\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_{p-1} X_{p-1} + \text{error},$$

but this is a mistake. The error in the logistic model is not an additive error on the logit scale, but the randomness in the binomial distribution itself.

In GLIM's, the unknown coefficients β are estimated by the method of maximum likelihood (ML). The details of ML are a little different for each of these models. Today we will describe the ML method for logistic regression.

ML for logistic regression. Suppose that we observe $y_i \sim \text{Bin}(n_i, \pi_i)$ for $i = 1, \dots, N$, where n_i is known and x_i is a vector of covariates

$$x_i = (x_{i1}, \dots, x_{ip})^T.$$

(In many cases, $x_{i1} = 1$.) The relationship is

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right) = x_i^T \beta = \beta_1 x_{i1} + \cdots + \beta_p x_{ip},$$

where $\beta = (\beta_1, \dots, \beta_p)^T$ is to be estimated.

Back-transforming to the probability scale gives

$$\pi_i = \text{expit}(x_i^T \beta) = \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} = \frac{1}{1 + e^{-x_i^T \beta}}.$$

The logit of π_i is log odds, and the coefficients are log-odds ratios. That is, β_j is the additive increase in log-odds resulting from a one-unit increase in x_{ij} . Except in some simple cases, there is no closed-form expression for the ML estimate of β . The estimate must be computed by an iterative procedure such as Newton-Raphson.

Suppose we want to maximize a loglikelihood $l(\theta)$ with respect to a parameter $\theta = (\theta_1, \dots, \theta_p)^T$. At each step of NR, the current estimate $\theta^{(t)}$ is updated as

$$\theta^{(t+1)} = \theta^{(t)} + [-l''(\theta^{(t)})]^{-1} l'(\theta^{(t)}),$$

where $l'(\theta)$ is the vector of first derivatives

$$l'(\theta) = (\partial l / \partial \theta_1, \dots, \partial l / \partial \theta_p)^T$$

(also called the score vector), and $l''(\theta)$ is the matrix of second derivatives (also called the Hessian). That is, $l''(\theta)$ is the $p \times p$ matrix with (j, k) th element equal to $\partial^2 l / \partial \theta_j \partial \theta_k$. We repeat this step until convergence, when $\theta^{(t+1)} \approx \theta^{(t)}$. If $l(\theta)$ were exactly quadratic, then $l'(\theta)$ would be exactly linear and NR would converge in a single step from any starting value. In “regular” problems, likelihood theory says that the loglikelihood tends to become approximately quadratic as the sample size grows.

Upon convergence, the inverse of the Hessian provides an estimated covariance matrix,

$$\hat{V}(\hat{\theta}) = \left[-l''(\hat{\theta}) \right]^{-1}$$

called the inverse of the “observed information.” In GLIM’s, a slightly different estimated covariance matrix comes from inverting the matrix of (minus one times) the expected second derivatives, which is called the inverse of the “expected information.” In the special cases of linear regression, logistic regression and loglinear models, the observed and expected information are identical, so for our purposes the distinction is not important.

For logistic regression, the loglikelihood is

$$\begin{aligned} l(\beta) &= \sum_{i=1}^N \{y_i \log \pi_i + (n_i - y_i) \log(1 - \pi_i)\} \\ &= \sum_{i=1}^N y_i \log \left(\frac{\pi_i}{1 - \pi_i} \right) - \sum_{i=1}^N n_i \log \left(\frac{1}{1 - \pi_i} \right) \\ &= \sum_{i=1}^N x_i^T \beta y_i - \sum_{i=1}^N n_i \log \left(1 + e^{x_i^T \beta} \right). \end{aligned}$$

We can express one step of NR using matrix notation

similar to that used linear regression. Let

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{bmatrix}.$$

where $\mu_i = n_i\pi_i$. Notice that y and μ are $N \times 1$, X is $N \times p$, and that calculating μ requires an assumed value for β . Also, define

$$W = \text{Diag}(n_i\pi_i(1 - \pi_i)),$$

an $N \times N$ matrix with elements $n_i\pi_i(1 - \pi_i)$ on the diagonal and 0's elsewhere; this is also based on an assumed value of β . One can show that the vector of first derivatives and the Hessian matrix are

$$\begin{aligned} l'(\beta) &= X^T(y - \mu), \\ l''(\beta) &= -X^T W X. \end{aligned}$$

For one step of Newton Raphson, we use $\beta^{(t)}$, the current estimate of β , to calculate $\mu^{(t)}$ and $W^{(t)}$. The new estimate of β is then

$$\beta^{(t+1)} = \beta^{(t)} + (X^T W^{(t)} X)^{-1} X^T (y - \mu^{(t)}). \quad (1)$$

We repeat this process until the estimates stop changing, i.e. until $\beta^{(t+1)}$ is sufficiently close to $\beta^{(t)}$. Upon

convergence, $(X^T W X)^{-1}$ is the estimated covariance matrix for $\hat{\beta}$. The fitted value for y_i is

$$\hat{\mu}_i = n_i \hat{\pi}_i = n_i \text{expit}(x_i^T \hat{\beta}),$$

$$i = 1, \dots, N.$$

NR as IRWLS. We can rearrange (1) to resemble weighted least squares; the resulting algorithm is called iteratively reweighted least squares (IRWLS). One iteration is

$$\hat{\beta}^{(t+1)} = (X^T W^{(t)} X)^{-1} X^T W z^{(t)},$$

where $z = (z_1, \dots, z_N)^T$ is the vector with elements

$$\begin{aligned} z_i &= x_i^T \beta + \frac{(y_i/n_i) - \pi_i}{\pi_i(1 - \pi_i)} \\ &= x_i^T \beta + \frac{(y_i/n_i) - \text{expit}(x_i^T \beta)}{\text{expit}(x_i^T \beta)(1 - \text{expit}(x_i^T \beta))}. \end{aligned}$$

In the literature of generalized linear models, z is often called the **adjusted dependent variate** or **working variate**. It takes the place of the Y -variable in ordinary regression. But it depends on β , so it changes at each iteration along with the weights in W .

In summary, this is how you can obtain the ML estimate for β in a logistic regression model. First, start with an

initial guess $\beta^{(0)}$. A good initial guess is

$$\beta^{(0)} = (0, 0, \dots, 0)^T.$$

Then, repeat the following steps:

- Using the current parameter estimate $\beta^{(t)}$, calculate the estimated probabilities

$$\hat{\pi}_i^{(t)} = \text{expit}(x_i^T \beta) = \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}},$$

the fitted values $\hat{\mu}_i^{(t)} = n_i \hat{\pi}_i^{(t)}$, the working variate

$$z_i^{(t)} = x_i^T \beta^{(t)} + \frac{(y_i/n_i) - \hat{\pi}_i^{(t)}}{\hat{\pi}_i^{(t)}(1 - \hat{\pi}_i^{(t)})},$$

and the weights

$$w_i^{(t)} = n_i \hat{\pi}_i^{(t)} (1 - \hat{\pi}_i^{(t)})$$

for $i = 1, \dots, N$.

- Regress the $z_i^{(t)}$'s on the X -variables using the weights $w_i^{(t)}$; the estimated coefficients from this regression become $\beta^{(t+1)}$

Repeat these two steps until convergence ($\beta^{(t+1)} \approx \beta^{(t)}$).

Example: Let's apply this procedure to the dose-response data:

Concentration	\log_{10} conc	n_i	y_i	p_i
1×10^{-5}	-5	6	0	0.000
1×10^{-4}	-4	6	1	0.167
1×10^{-3}	-3	6	4	0.667
1×10^{-2}	-2	6	6	1.000
1×10^{-1}	-1	6	6	1.000

Let's fit the model

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 \log_{10}\text{conc}_i.$$

Here is a simple implementation in R that begins with a starting value of $\beta^{(0)} = (0, 0)^T$ and performs nine steps of NR, saving the estimates at each step.

```
> # enter data for y, X and n's
> y <- c(0,1,4,6,6)
> X <- cbind( 1, c(-5,-4,-3,-2,-1) )
> n <- c(6,6,6,6,6)

> # starting value for beta
> beta <- c(0,0)

> # matrix to hold the estimates at each iteration
> beta.matrix <- matrix(NA, 10, 2)
> beta.matrix[1,] <- beta

> # perform nine steps of NR
> for( iter in 2:10){
+   beta.old <- beta
+   Xbeta <- X %*% beta.old
+   pihat <- exp( Xbeta ) / ( 1 + exp( Xbeta ) )
+   z <- Xbeta + ( y/n - pihat ) / ( pihat * (1-pihat) )
```

```
+     W <- diag( as.vector( n * pihat * (1-pihat) ) )
+     beta <- solve( t(X)%*%W%*%X ) %*% t(X)%*%W%*%z
+     beta.matrix[iter,] <- beta}

> # print the results
> beta.matrix
      [,1]      [,2]
[1,] 0.000000 0.000000
[2,] 3.666667 1.133333
[3,] 6.045018 1.836995
[4,] 8.106183 2.441095
[5,] 9.303587 2.794395
[6,] 9.575939 2.875864
[7,] 9.586791 2.879160
[8,] 9.586808 2.879165
[9,] 9.586808 2.879165
[10,] 9.586808 2.879165

> # compute the estimated covariance matrix
> covmat <- solve( t(X) %*% W %*% X )
> covmat
      [,1]      [,2]
[1,] 13.739425 4.012557
[2,]  4.012557 1.214957

> # print the standard errors
> sqrt( diag( covmat ) )
[1] 3.706673 1.102251
```

GENERALIZED LINEAR MODELS: A LIKELIHOOD VIEW (PART II)

Last time, we introduced generalized linear models (GLIM's) as an extension of normal linear regression. In a GLIM, we assume that the response variable y_i comes from a regular exponential family distribution with mean μ_i , and that

$$g(\mu_i) = x_i^T \beta$$

where g called the link function.

As our first example of a GLIM, we discussed logistic regression, which combines the binomial family with a logistic link,

$$\begin{aligned} y_i &\sim \text{Bin}(n_i, \pi_i), \\ \log\left(\frac{\pi_i}{1 - \pi_i}\right) &= x_i^T \beta. \end{aligned}$$

The n_i 's are assumed to be known and must be supplied by the user. Logistic regression is often used to model a binary response, where $y_i = 0$ or 1 . In that case, all of the n_i 's are taken to be 1.

Counts versus proportions. Notice that the binomial distribution has mean $\mu_i = n_i \pi_i$, but we are applying the

logistic transformation to π_i rather than μ_i . To make this model fit into the GLIM framework, we actually have to define the “response” as the proportion of successes rather than number of successes in n_i trials, so that the success probability π_i becomes the mean. This is a small technical detail that comes up in the GLIM literature. It also comes up when we are using the built-in function for generalized linear modeling in R, which is called `glm`. When using `glm`, we will actually have to supply y_i/n_i rather than y_i as the response variable. In other programs (e.g., PROC LOGISTIC in SAS) you may supply either the counts or the proportions as the response. The distinction becomes irrelevant when y_i is binary. In these lecture notes, we will continue to use y_i to denote the number of successes, take $\mu_i = n_i\pi_i$, and apply the logistic transformation to π_i .

Interpreting the coefficients of the logistic model.

In the last lecture, we described a Newton-Raphson (NR) procedure for computing the maximum-likelihood (ML) estimate of β , and we implemented it in R for the small dose-response dataset. Recall that n_i is the number of animals exposed to the viral solution in group i , and y_i is the number of animals who died.

Concentration	\log_{10} conc	n_i	y_i	p_i
1×10^{-5}	-5	6	0	0.000
1×10^{-4}	-4	6	1	0.167
1×10^{-3}	-3	6	4	0.667
1×10^{-2}	-2	6	6	1.000
1×10^{-1}	-1	6	6	1.000

We fit the model

$$\log\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1 \log_{10} \text{conc}_i$$

and found $\hat{\beta}_0 = 9.586808$ and $\hat{\beta}_1 = 2.879165$. We can interpret the estimated coefficients more easily by exponentiating them. The exponentiated intercept

$$\exp(\hat{\beta}_0) = \exp(9.587) \approx 14570$$

estimates the odds of death when $\log_{10} \text{conc} = 0$. This value of $\log_{10} \text{conc}$ is outside the range of the observed data, so we are extrapolating; when we look at a confidence interval, we'll see that the interval is very wide on the odds scale. But on the probability scale, the interval will be very narrowly concentrated near one. The estimated probability of death when $\log_{10} \text{conc} = 0$ is

$$\frac{\exp(\hat{\beta}_0)}{1 + \exp(\hat{\beta}_0)} = \frac{14570}{14570 + 1} = .9999,$$

which means that death is virtually certain at such high concentrations.

The exponentiated slope is

$$\exp(\hat{\beta}_1) = \exp(2.879) = 17.80.$$

The interpretation is: every one-unit increase in \log_{10} conc *multiplies* the odds of death by 17.8. This is an estimated odds ratio.

The logistic model stipulates that the effect of a covariate on the chance of “success” is linear on the log-odds scale, or multiplicative on the odds scale. Under the model

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots,$$

increasing the value of X_j by one unit causes the odds $\pi/(1 - \pi)$ to be multiplied by $\exp(\beta_j)$, if the other X 's are held constant.

If $\beta_j > 0$, then $\exp(\beta_j) > 1$ and the odds increase as X goes up; if $\beta_j < 0$, then $\exp(\beta_j) < 1$ and the odds go down.

The effect of X_j on the probability scale depends on where you start. For example, suppose that $\beta_j = 0.693$, so that $\exp(\beta_j) = 2.00$; then every one-unit increase in X_j causes the odds to double. The table below shows, for various baseline values of π , the new value π^* that results when the odds are doubled.

π	π^*
.0001	.0002
.0010	.0020
.0100	.0198
.0500	.0952
.1000	.1818
.2500	.4000
.5000	.6667
.7500	.8571
.9000	.9474
.9500	.9744
.9900	.9950
.9990	.9995
.9999	.9999

Notice that when π and π^* are both close to zero, doubling the odds is essentially doubling the probability. This is because

$$\frac{\pi^*/(1 - \pi^*)}{\pi/(1 - \pi)} = \frac{\pi^*}{\pi} \frac{1 - \pi}{1 - \pi^*} \approx \frac{\pi^*}{\pi}$$

when $\pi \approx 0$ and $\pi^* \approx 0$. At the low end of the scale, an odds ratio is approximately the same as a relative risk. This means that, for predicting rare events, an exponentiated logistic regression coefficient can be interpreted as a relative risk.

Confidence intervals. Upon convergence of the IRWLS algorithm,

$$\hat{\beta}^{(t+1)} = \left(X^T W^{(t)} X \right)^{-1} X^T W z^{(t)},$$

the final value of $(X^T W X)^{-1}$ is an estimated covariance matrix for $\hat{\beta}$. Fixing the number of parameters and letting $N \rightarrow \infty$, it can be shown that $\hat{\beta}$ becomes approximately multivariate normally distributed around the true β , and $(X^T W X)^{-1}$ converges to the true covariance matrix $\text{Var}(\hat{\beta})$ if the model is true.

Therefore, an approximate 95% interval for a single coefficient β_j is

$$\hat{\beta}_j \pm 1.96 \sqrt{(X^T \hat{W} X)_{jj}^{-1}},$$

where $(X^T \hat{W} X)_{jj}^{-1}$ denotes the j th diagonal element of $(X^T W X)^{-1}$, and \hat{W} denotes the final estimate of the weight matrix W upon convergence. In practice, we may want an interval for $\exp(\beta_j)$ rather than β_j . In that case, we would calculate endpoints of the interval for β_j and exponentiate them.

In our dose-response example, we found that the estimated covariance matrix and standard errors were:

```
> # compute the estimated covariance matrix
> covmat <- solve( t(X) %*% W %*% X )
> covmat
     [,1]      [,2]
```

```
[1,] 13.739425 4.012557
[2,] 4.012557 1.214957

> # print the standard errors
> sqrt( diag( covmat ) )
[1] 3.706673 1.102251
```

The 95% interval for our exponentiated intercept is from

$$\exp(9.587 - 1.96(3.707)) \approx 10.19$$

to

$$\exp(9.587 + 1.96(3.707)) \approx 20,847,000.$$

The interval for the odds of death when $\log_{10} \text{conc} = 0$ is very wide. But on the probability scale, the interval ranges from .910 to 1.000. The 95% interval for the odds ratio associated with a one-unit increase in $\log_{10} \text{conc}$ ranges from

$$\exp(2.8792 - 1.96(1.102)) = 2.052$$

to

$$\exp(2.8792 + 1.96(1.102)) = 154.4.$$

Tests. It is often of interest to test the null hypothesis $\beta_j = 0$ versus the alternative $\beta_j \neq 0$. That is, we are testing whether X_j is a significant predictor in the presence of the other X 's (if any). One simple way to do this is to compare the ratio of the estimated coefficient to its standard error,

$$z = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)},$$

to a standard normal distribution. The p-value is twice the area to the right of $|z|$ under the normal curve. Basically, if $z > 2$ or $z < -2$ we would call it significant at the .05 level.

Tests for coefficients based on the estimated covariance matrix $(X^T W X)^{-1}$ (the inverse of the information matrix) are called Wald tests. In large samples—when the overall number of binomial trials $\sum_{i=1}^N n_i$ is large—the Wald tests tend to be accurate. Another way to test significance is to use a likelihood ratio test. To do a likelihood-ratio test of $\beta_j = 0$ versus the alternative $\beta_j \neq 0$, we need to first fit the model that allows β_j to be freely estimated (which we have done), and calculate the value of the loglikelihood function

$$\begin{aligned} l(\beta) &= \sum_{i=1}^N \{y_i \log \pi_i + (n_i - y_i) \log(1 - \pi_i)\} \\ &= \sum_{i=1}^N y_i \log \left(\frac{\pi_i}{1 - \pi_i} \right) - \sum_{i=1}^N n_i \log \left(\frac{1}{1 - \pi_i} \right) \\ &= \sum_{i=1}^N x_i^T \beta y_i - \sum_{i=1}^N n_i \log \left(1 + e^{x_i^T \beta} \right) \end{aligned}$$

at $\beta = \hat{\beta}$. Then we need to fit a restricted model that sets $\beta_j = 0$ —in other words, the reduced model that omits X_j but retains all the other covariates—and calculate the value of the loglikelihood function at the ML estimate for that reduced model. The loglikelihood for the full model must be greater than the loglikelihood for the reduced

model. Twice the difference in loglikelihood is approximately distributed as χ^2_1 , if the null model is true.

The Wald test for an individual coefficient of a GLIM is analogous to a t-test for a coefficient in a linear regression model. But there is a slight difference. In normal linear regression, the distribution of $\hat{\beta}_j / SE(\hat{\beta}_j)$ under the null hypothesis that $\beta_j = 0$ is exactly Student's t. In GLIM's, we have no exact distributional theory, only large-sample approximations based on the asymptotic normality of ML estimates. Similarly, a comparison of nested GLIM's based on the difference in $2 \times \log\text{likelihood}$ is analogous to an F-test in normal linear regression. The F distribution for normal models is an exact result, whereas the χ^2 distribution for GLIM's is only an approximation. When the two nested normal models differ by only one coefficient, the F test is identical to the t-test. With GLIM's, however, the likelihood ratio statistic and the squared Wald z -statistic are only approximately equal.

Logistic regression in R. The `glm` function in R is for fitting generalized linear models. It is very similar to `lm` in that you need to supply a model formula and a data frame. But you also need to supply a distributional family and a link function.

When using `glm` to perform logistic regression, the response variable (the left-hand side of the model formula) should be the proportion of successes in n_i trials (or the

binary response if $n_i = 1$). The counts n_i , if present, are supplied through the argument **weights**. (If no weights are supplied, it is assumed that $n_i = 1$.) Let's apply **glm** to the dose-response data.

```
> # enter the data and create a data frame
> y <- c(0,1,4,6,6)
> x <- c(-5,-4,-3,-2,-1)
> n <- c(6,6,6,6,6)
> p <- y/n
> dose <- data.frame( y=y, x=x, p=p, n=n)

> # fit the logistic model
> result <- glm( p ~ x, weights=n,
+   family=binomial(link="logit"), data=dose)

> summary(result)

Call:
glm(formula = p ~ x, family = binomial(link = "logit"), data = dose,
     weights = n)

Deviance Residuals:
      1       2       3       4       5 
-0.3122  0.2821 -0.2913  0.5081  0.1210 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  9.587     3.707   2.586   0.0097 **  
x           2.879     1.102   2.612   0.0090 **  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 28.0090  on 4  degrees of freedom
Residual deviance: 0.5347  on 3  degrees of freedom
AIC: 8.58

Number of Fisher Scoring iterations: 6
```

The result of `glm` is a list of many different quantities. We won't have time to explain all of them. But you can probably figure out what some of them are.

```
> names(result)
[1] "coefficients"      "residuals"          "fitted.values"
[4] "effects"           "R"                  "rank"
[7] "qr"                "family"            "linear.predictors"
[10] "deviance"          "aic"                "null.deviance"
[13] "iter"               "weights"            "prior.weights"
[16] "df.residual"        "df.null"            "y"
[19] "converged"          "boundary"           "model"
[22] "call"               "formula"            "terms"
[25] "data"               "offset"             "control"
[28] "method"             "contrasts"         "xlevels"
```

One important quantity that is not found in this list is $(X^T W X)^{-1}$, the estimated covariance matrix for $\hat{\beta}$. If you need to get this matrix, apply the `summary` function and save the result; the matrix $(X^T W X)^{-1}$ will be called `cov.unscaled`.

```
> tmp <- summary(result)

> names(tmp)
[1] "call"              "terms"             "family"            "deviance"
[5] "aic"               "contrasts"         "df.residual"       "null.deviance"
[9] "df.null"           "iter"              "deviance.resid"   "coefficients"
[13] "aliased"           "dispersion"        "df"                "cov.unscaled"
[17] "cov.scaled"

> tmp$cov.unscaled
(Intercept)      x
(Intercept)  13.739420 4.012555
x            4.012555 1.214956
```

Interpreting the `glm` summary. After running `glm`, the function `summary` produces printout similar to that for a normal linear regression model. But there are a few differences. Notice that in the table of coefficients, the ratio of the estimate to the standard error is now called “z value,” because we compare it to a normal distribution rather than a t.

Notice also the note “Dispersion parameter for binomial family taken to be 1.” Generalized linear models have something called a dispersion parameter which corresponds to the residual variance in normal linear regression. For the binomial family, this parameter is set to 1. We will talk more about this later, when we discuss quasilikelihood.

Near the end of the `summary` output, there is something called a deviance table which is like an ANOVA table. Instead of reporting sums of squares, it reports the deviance, which is basically -2 times the value of the loglikelihood function.

The “residual deviance” is analogous to $SS_{E_{rr}}$. It measures the lack of fit of the current model relative to a saturated model that fits the data perfectly. The residual degrees of freedom is $N - p$, the number of observations in the dataset minus the number of coefficients in the model. In the dose-response example, we have $N = 5$ and $p = 2$, so the residual df is 3.

The “null deviance” is analogous to the residual sum of squares from a model that contains only an intercept (i.e., the total sum of squares). It measures the lack of fit of the intercept-only model,

$$\log \left(\frac{\pi_i}{1 - \pi_i} \right) = \beta_0$$

for $i = 1, \dots, 5$. For this model, N is still 5, but $p = 1$, so the degrees of freedom for the null deviance is $N - 1 = 4$.

If we compare the null deviance to the residual deviance, we get a likelihood ratio test that is analogous to the omnibus F-test in normal linear regression, and it tests the null hypothesis that the slopes for all predictors (except the constant) are simultaneously zero. In this example, there is only a single predictor, so it is testing whether $\beta_1 = 0$. The statistic is called ΔG^2 rather than F , and it is equal to

$$\begin{aligned} \Delta G^2 &= \text{Null deviance} - \text{Residual deviance} \\ &= 28.0090 - 0.5347 \\ &= 27.4743. \end{aligned}$$

We compare this statistic to a χ^2 distribution with degrees of freedom equal to

$$\begin{aligned} \Delta \text{df} &= \text{Null df} - \text{Residual df} \\ &= 4 - 3 \\ &= 1. \end{aligned}$$

The p-value is

$$P(\chi_1^2 \geq 27.4743) \approx 0.$$

Because this is testing the significance of only a single coefficient β_1 , it gives essentially (but not exactly) the same result as the test for $\beta_1 = 0$ based on the z-statistic from the table of coefficients. In this example, the statistics seem far apart ($\Delta G^2 = 27.4743$ versus $z^2 = (2.612)^2 = 6.823$) but the p-values are essentially identical ($p \approx 0$).

How can we test the joint significance of a group of predictors? If Models A and B are nested, we can test the null hypothesis

$$H_0 : \text{Model A is true}$$

against the alternative

$$H_1 : \text{Model B is true}$$

by fitting both models and looking at their residual deviances. The ΔG^2 test statistic, which is analogous to a partial F, would be the residual deviance for Model A minus the residual deviance for Model B. We would compare this to a chisquare distribution with degrees of freedom equal to the number of parameters in Model B minus the number of parameters in Model A (i.e., the residual df for Model A minus the residual df for Model B).

Grouped versus ungrouped data. In the dose-response example, we expressed the data as $N = 5$ groups of $n_i = 6$ animals each. But we can also disaggregate the data into $N = 30$ lines and one animal per line. The data file `dose.txt` looks like this.

```
x y
-5 0
-5 0
-5 0
-5 0
-5 0
-5 0
-4 0
-4 0
-4 0
-4 0
-4 0
-4 1
-3 0
-3 0
-3 1
-3 1
-3 1
-3 1
-2 1
-2 1
-2 1
-2 1
-2 1
-2 1
-1 1
-1 1
-1 1
-1 1
-1 1
-1 1
```

We can fit the same logistic model to the disaggregated

data. The response y_i is the binary variable equal to one if the animal died and zero otherwise. The weight n_i for each observation is assumed to be 1.

```
> dose <- read.table("dose.txt", header=T)
> result <- glm( y ~ x, family=binomial(link="logit"), data=dose)
> summary(result)

Call:
glm(formula = y ~ x, family = binomial(link = "logit"), data = dose)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-1.59779 -0.12746  0.04941  0.20741  2.03244 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  9.587     3.707   2.586   0.0097 **  
x            2.879     1.102   2.612   0.0090 **  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 41.054  on 29  degrees of freedom
Residual deviance: 13.580  on 28  degrees of freedom
AIC: 17.580

Number of Fisher Scoring iterations: 7
```

The estimated coefficients and standard errors are exactly the same as when the observations were grouped, because the loglikelihood functions are the same. The loglikelihood from a binomial experiment is (except for an irrelevant constant) equal to the sum of the loglikelihoods from n_i Bernoulli trials. So we get the same ML estimates and SE's whether the data are grouped or ungrouped.

Notice, however, that the deviance table is not the same. The residual deviance measures the lack of fit of the current model relative to the saturated model. The current model has not changed. But now that the dataset has $N = 30$ lines, the saturated model estimates 30 separate π_i 's, one for each line in the dataset. So the residual df is now $N - p = 30 - 2 = 28$. Similarly, the null deviance measures the lack of fit of the null (intercept-only) model relative to the saturated model, so its degrees of freedom are $30 - 1 = 29$. But if we test the null model against the current model, we get

$$\begin{aligned}\Delta G^2 &= \text{Null deviance} - \text{Residual deviance} \\ &= 41.054 - 13.580 \\ &= 27.474\end{aligned}$$

with $29 - 28 = 1$ degree of freedom, which is exactly the same result as we obtained with grouped data.

GENERALIZED LINEAR MODELS: A LIKELIHOOD VIEW (PART III)

Over the last two lectures, we introduced the logistic regression model, which assumes that

$$y_i \sim \text{Bin}(n_i, \pi_i),$$

$$\log\left(\frac{\pi_i}{1 - \pi_i}\right) = x_i^T \beta$$

for $i = 1, \dots, N$. Equivalently, it assumes that

$$\pi_i = \text{expit}(x_i^T \beta) = \frac{\exp(x_i^T \beta)}{1 + \exp(x_i^T \beta)}.$$

We fit this model by a Newton-Raphson procedure which can be expressed as iteratively reweighted least squares (IRWLS). One iteration of IRWLS is

$$\hat{\beta}^{(t+1)} = \left(X^T W^{(t)} X\right)^{-1} X^T W z^{(t)},$$

where

- W is the $N \times N$ weight matrix with elements $w_i = n_i \pi_i (1 - \pi_i)$ on the diagonal and 0's elsewhere, and
- $z = (z_1, \dots, z_N)^T$ is the working variate, the vector

with elements

$$\begin{aligned} z_i &= x_i^T \beta + \frac{(y_i/n_i) - \pi_i}{\pi_i(1 - \pi_i)} \\ &= x_i^T \beta + \frac{(y_i/n_i) - \text{expit}(x_i^T \beta)}{\text{expit}(x_i^T \beta)(1 - \text{expit}(x_i^T \beta))}. \end{aligned}$$

Upon convergence, the estimated covariance matrix for $\hat{\beta}$ is $(X^T W X)^{-1}$. Today we'll look at one more example of logistic regression and discuss some diagnostics.

Example. The SAS on-line help documentation provides the following quantal assay dataset. In this table, x_i refers to the log-dose, n_i is the number of subjects exposed, and y_i is the number who responded.

x_i	y_i	n_i
2.68	10	31
2.76	17	30
2.82	12	31
2.90	7	27
3.02	23	26
3.04	22	30
3.13	29	31
3.20	29	30
3.21	23	30

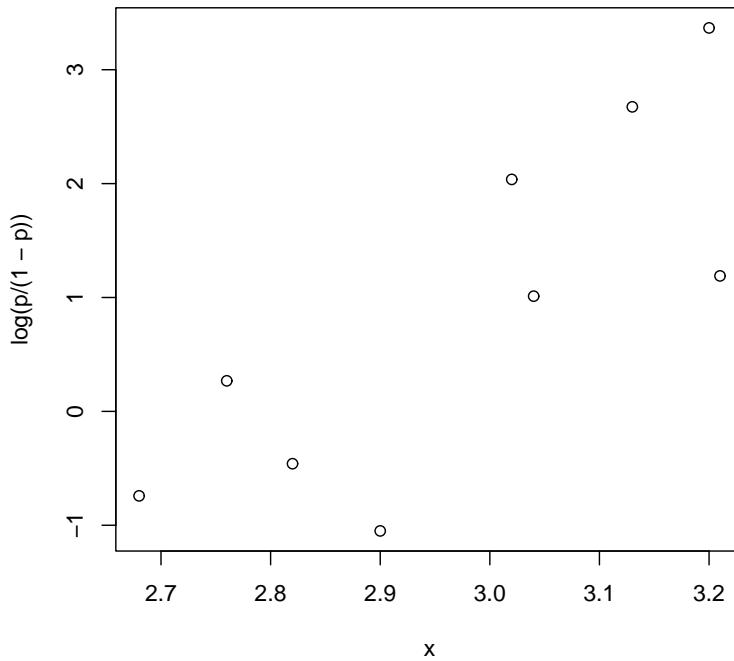
The most obvious model to try is a logistic regression with

$$\log\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1 x_i.$$

But is it reasonable to assume that the log-odds is a linear function of x_i ? An easy way to find out is to compute the sample proportions $p_i = y_i/n_i$, then plot $\log(p_i/(1 - p_i))$ versus x_i .

```
> x <- c(2.68, 2.76, 2.82, 2.90, 3.02, 3.04, 3.13, 3.20, 3.21)
> y <- c(10, 17, 12, 7, 23, 22, 29, 29, 23)
> n <- c(31, 30, 31, 27, 26, 30, 31, 30, 30)

> p <- y/n
> plot( x, log(p/(1-p)) )
```



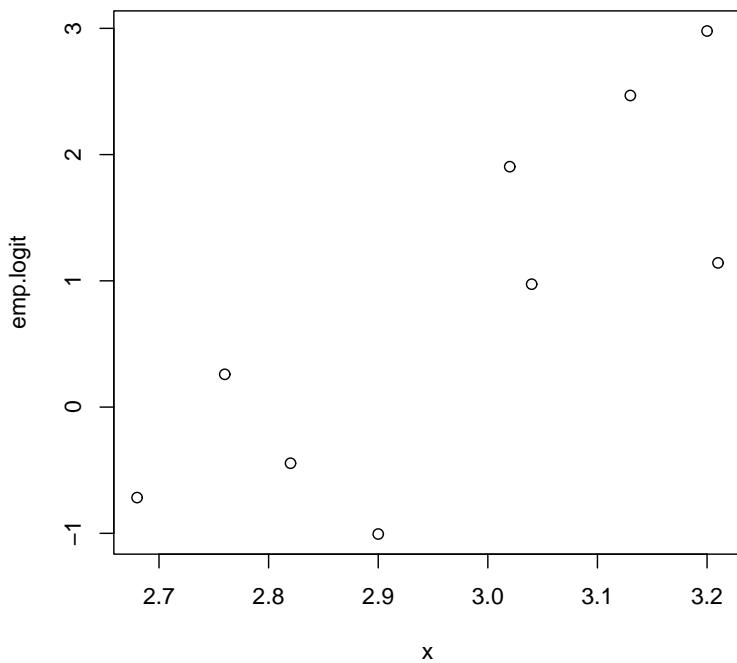
Yes, it does appear that log-odds may increase linearly with x_i .

If any of the groups happens to have a 0% success rate ($y_i = 0$) or a 100% success rate ($y_i = n_i$), then we would have $p_i = 0$ or $p_i = 1$, and $\log(p_i/(1 - p_i))$ would be undefined. If that happens, it is customary to plot

$$\log\left(\frac{y_i + 0.5}{n_i - y_i + 0.5}\right)$$

instead. This is sometimes called the “empirical logit.” In effect, it smooths the data by adding 1/2 of a success and 1/2 of a failure to each group.

```
> emp.logit <- log( (y+.5) / (n-y+.5) )
> plot( x, emp.logit)
```



When multiple predictors are available, it's a good idea to

begin an analysis by examining scatterplots of the empirical logits versus each predictor, just as you would examine plots of Y versus the X 's in ordinary linear regression. When dealing with ungrouped data ($n_i = 1$), however, these plots are not very informative, as all of the points will fall onto two horizontal lines.

Going back to our data example, let's fit the logistic model in R.

```
> p <- y/n
> result <- glm( p ~ x, family=binomial(link="logit"), weights=n)
> summary(result)

Call:
glm(formula = p ~ x, family = binomial(link = "logit"), weights = n)

Deviance Residuals:
    Min      1Q  Median      3Q      Max 
-3.4286 -0.9736  0.3699  1.6777  1.9302 

Coefficients:
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -15.8339    2.4371 -6.497 8.20e-11 ***
x             5.5778    0.8319  6.705 2.02e-11 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 83.631  on 8  degrees of freedom
Residual deviance: 29.346  on 7  degrees of freedom
AIC: 62.886

Number of Fisher Scoring iterations: 4
```

Linear predictors, fitted values and residuals. In

logistic regression, the estimated value for the log-odds is

$$\log \left(\frac{\hat{\pi}_i}{1 - \hat{\pi}_i} \right) = x_i^T \hat{\beta}.$$

In GLIM terminology, this is called the **linear predictor**.

The estimated value of π_i is

$$\hat{\pi}_i = \frac{\exp(x_i^T \hat{\beta})}{1 + \exp(x_i^T \hat{\beta})},$$

and the estimated value of $\mu_i = E(y_i)$ is $\hat{\mu}_i = n_i \hat{\pi}_i$. The difference between y_i and $\hat{\mu}_i$ is the raw residual. In logistic regression and other GLIM's, we usually do not examine the raw residuals because (unlike their counterparts from ordinary linear regression) their variance is not even approximately constsnt. The binomial model stipulates that $E(y_i) = n_i \pi_i$ and $V(y_i) = n_i \pi_i (1 - \pi_i)$, so we can approximately standardize the residual by taking

$$r_i = \frac{y_i - n_i \hat{\pi}_i}{\sqrt{n_i \hat{\pi}_i (1 - \hat{\pi}_i)}}.$$

This is called the Pearson residual. If the model is true, then the Pearson residuals should have mean ≈ 0 and variance ≈ 1 . But they are not normally distributed, because y_i is discrete. If the n_i 's are large, then the r_i 's become approximately normal (recall the normal approximation to the binomial distribution). In that case, a plot of the Pearson residuals versus the linear predictors (or any of the individual predictors) will resemble a

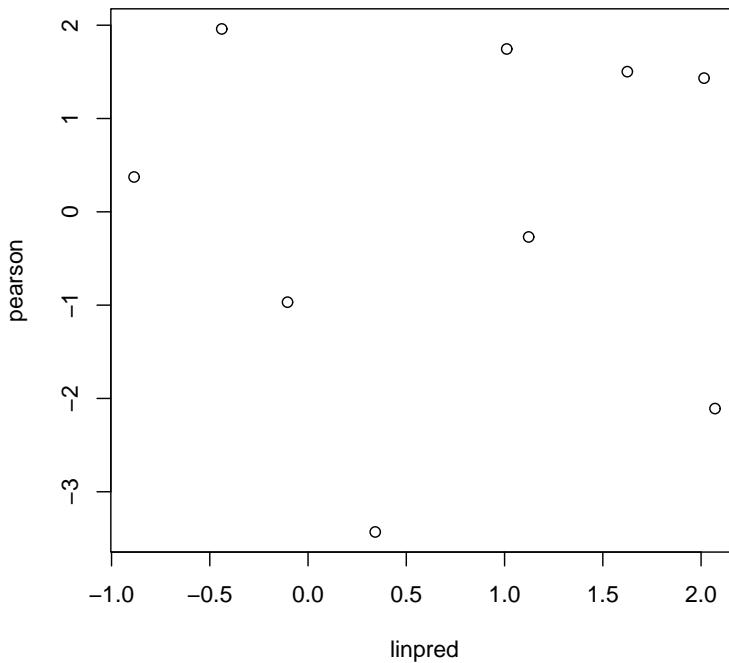
horizontal band, assuming that the model is true. But if the n_i 's are small, residual plots may have an unusual appearance, because the responses are highly discrete. In the most extreme case, when the responses are binary (all n_i 's equal to 1), points in a residual plot may appear to “line up” on two curves corresponding to cases with $y_i = 0$ and $y_i = 1$.

In R, the $\hat{\pi}_i$'s are available as the component **fitted.values** in the result of a call to **glm**. The linear predictors are available as the component **linear.predictors**. Here is a plot of the Pearson residuals versus the linear predictors for our current example.

```
> linpred <- result$linear.predictors  
> pihat <- result$fitted.values  
> pearson <- (y-n*pihat) / sqrt( n*pihat*(1-pihat) )  
> plot( linpred, pearson)
```

The Pearson residuals can also be obtained with the **residuals** function, like this.

```
> pearson <- residuals(result, type="pearson")
```



In this example, we don't see any obvious tendency for the variance of these residuals to increase or decrease with the linear predictor. But one of the Pearson residuals is unusually large ($r_i = -3.43$). In fact, the distribution of all the r_i 's seems to be more dispersed than it should be. Looking at some descriptive statistics, we see that their mean is close to zero, but their standard deviation is nearly twice as large as what we would expect.

```
> mean(pearson)
[1] 0.02639009
> sqrt(var(pearson))
[1] 1.889363
```

This gives us some evidence that our model is wrong. In

fact, we can use the Pearson residuals to construct a formal goodness-of-fit test.

Goodness-of-fit testing. The Pearson goodness-of-fit test statistic for logistic regression is defined as

$$X^2 = \sum_{i=1}^N r_i^2 = \sum_{i=1}^N \frac{(y_i - n_i \hat{\pi}_i)^2}{n_i \hat{\pi}_i (1 - \hat{\pi}_i)}.$$

If the model is true and the within-group sample sizes n_i are sufficiently large, then X^2 will be approximately distributed as χ^2 with degrees of freedom equal to $N - p$. For this reason, it is sometimes called “Pearson’s chisquare.” If the n_i ’s are large enough, we can use X^2 to test the null hypothesis that our current model is true, versus the alternative of a saturated model. (The saturated model estimates π_i by $p_i = y_i/n_i$ independently for $i = 1, \dots, N$. It is equivalent to a logistic regression model with an intercept and $N - 1$ dummy indicators to distinguish among the N lines of the dataset.) If the n_i ’s are large enough, then we can reliably compare X^2 to a chisquare distribution. The p-value would be

$$p = P(X^2 \geq \chi^2_{N-p}),$$

and a small p-value (say, $p \leq .05$) would indicate that the model does not fit. This is analogous to the general lack-of-fit (LOF) test for normal linear regression.

How large do the n_i ’s have to be for this test to be

reliable? Many textbooks give the following rule of thumb: We need to have $n_i\hat{\pi}_i \geq 5$ and $n_i(1 - \hat{\pi}_i) \geq 5$ for 80% or more of the cases in our dataset. In addition, none of the values of $n_i\hat{\pi}_i$ or $n_i(1 - \hat{\pi}_i)$ should be less than 1.0. Because of this rule, **there is no reliable goodness-of-fit test for ungrouped binary data** (i.e., when all n_i 's are equal to 1).

Let's examine the values of $n_i\hat{\pi}_i$ and $n_i(1 - \hat{\pi}_i)$ in our example.

```
> cbind( n*pihat, n*(1-pihat) )
   [,1]      [,2]
1 9.054397 21.945603
2 11.758729 18.241271
3 14.691467 16.308533
4 15.784973 11.215027
5 19.064334  6.935666
6 22.634668  7.365332
7 25.898721  5.101279
8 26.471356  3.528644
9 26.641357  3.358643
```

Here, $7/9 \approx 78\%$ of the observations have $n_i\hat{\pi}_i \geq 5$ and $n_i(1 - \hat{\pi}_i) \geq 5$, and none are below 1.0, so the X^2 test is probably reliable. Here is the X^2 test.

```
> X2 <- sum(pearson^2)
> X2
[1] 28.56380

> # compute the p-value
> N <- 9
> p <- 2
> 1-pchisq( X2, N-p )
```

```
[1] 0.0001737254
```

It appears that this model does not fit.

Another way to assess the overall fit of the model is by the deviance. The deviance is the likelihood-ratio test statistic for comparing the current model to the saturated model.

For logistic regression, the deviance statistic is

$$G^2 = 2 \sum_{i=1}^N \left\{ y_i \log \left(\frac{y_i}{\hat{\mu}_i} \right) + (n_i - y_i) \log \left(\frac{n_i - y_i}{n_i - \hat{\mu}_i} \right) \right\},$$

which we would again compare to χ^2_{N-p} . The deviance test also needs the n_i 's to be large enough so that most cases have $n_i \hat{\pi}_i \geq 5$ and $n_i(1 - \hat{\pi}_i) \geq 5$. Goodness-of-fit tests based on X^2 and G^2 are asymptotically equivalent and should yield similar results when the n_i 's are large. In R, the deviance statistic G^2 is printed out by the **summary** command. It is called “residual deviance.”

```
Null deviance: 83.631  on 8  degrees of freedom
Residual deviance: 29.346  on 7  degrees of freedom
AIC: 62.886
```

```
Number of Fisher Scoring iterations: 4
```

In this example, G^2 is close to $X^2 = 28.56$ and leads to the same conclusion.

The contribution of an individual case to the G^2 statistic is

$$2 \left\{ y_i \log \left(\frac{y_i}{\hat{\mu}_i} \right) + (n_i - y_i) \log \left(\frac{n_i - y_i}{n_i - \hat{\mu}_i} \right) \right\}.$$

If we multiply the square root of this contribution by the sign of $y_i - n_i \hat{\pi}_i$, we get a quantity known as the **deviance residual**. The deviance residual behaves like a Pearson residual, with mean ≈ 0 and variance ≈ 1 . When the n_i 's are large, the deviance residuals and the Pearson residuals will be similar to each other, and data analysts often examine them along with the Pearson residuals. Here are both sets of residuals for our example.

```
> deviance <- residuals( result, type="deviance")
> cbind( pearson, deviance )
   pearson    deviance
1  0.3734963  0.3699037
2  1.9601509  1.9302395
3 -0.9681214 -0.9735884
4 -3.4308382 -3.4285613
5  1.7452194  1.8985941
6 -0.2692301 -0.2667341
7  1.5022528  1.6776633
8  1.4330334  1.6642304
9 -2.1084523 -1.8763202
```

Notice that the squared deviance residuals add up to the “residual deviance.”

```
> sum( deviance^2)
[1] 29.34616
```

What if the model does not fit? If the logistic model does not fit, the problem may be due to **omitted covariates**. For example, if the true relationship between the log-odds and a covariate happens to be quadratic, but

we have only a linear term, then we may get a high X^2 or G^2 . In that case, the problem may be solved by bringing additional terms into the model. In our current example, this does not seem to be the problem, because the plot of the empirical logits versus the predictor appeared to be linear.

Lack of fit may also be due to a phenomenon known as **overdispersion**. Overdispersion means that the actual variance of the response variable y_i is larger than what the model says that it should be. If y_i were binomially distributed, then $V(y_i) = n_i \pi_i(1 - \pi_i)$, and the Pearson residuals should be approximately $N(0, 1)$. In our example, the variance of the Pearson residuals was substantially larger than 1.0, so it appears that we have overdispersion.

What can we do about overdispersion? It's possible to fit an expanded model that assumes

$$V(y_i) = \phi n_i \pi_i (1 - \pi_i)$$

for some $\phi > 0$. This additional factor ϕ is called a **scale parameter** or **dispersion parameter**. The binomial model sets $\phi = 1$. Estimating so setting ϕ to any other value takes us outside of the binomial family and into the realm of **quasilikelihood**, which we will discuss very soon.

INTRODUCTION TO CAUSAL INFERENCE (PART I)

Introduction. Textbooks on elementary statistics never fail to warn us that “correlation does not imply causation.” This statement has become an axiom of scientific research and data analysis. What is the difference between correlation and causation? Correlation simply means that two variables are related. Causation means that changing one variable actually **produces a change** in the other variable. When we speak of causation, we are implicitly considering different outcomes corresponding to different values of the causal variable.

Suppose I enter a room, flip a switch on the wall, and observe that the lights in the room come on. Is it proper for me to conclude that flipping the switch on the wall *caused* the lights in the room to go on? Not necessarily. When we say that one event A caused another event B , we are not merely saying that the two events occurred together (first A happened, then B happened). We are also asserting that **if A had not happened, then B would not have happened**. That is, we are saying, “If I had not flipped the switch, the lights would not have come on.”

The statement “ A causes B ” actually means two things:

$$\begin{array}{ccc} A & \rightarrow & B \\ A^C & \rightarrow & B^C \end{array}$$

This creates an observational dilemma, however, because we cannot observe the outcomes under both A and A^C . If we flip the switch and the lights come on, we do not know for certain that the lights would not have come on anyway by themselves at that precise moment. Similarly, if we do not flip the switch and the lights do not come on, then we do not know what would have happened at that moment if we had flipped the switch. If A happens, we see the outcome (either B or B^C) under A , but we do not see the counterfactual outcome under A^C .

Even though we cannot simultaneously observe the outcome under both conditions, it is possible for us to draw rigorous causal conclusions through **replication** and **randomization**. Suppose we

- designate ten moments in time at which we will observe the state of the lights in the room (off or on). And suppose we
- randomly select five of these time points, flip the switch on those five occasions, and do not flip the switch on the other five occasions.

If we observe that the lights came on at the five occasions when we flipped the switch, and they did not come on at the other five occasions, then we have very strong evidence of causation. If the lights were going to come on by themselves anyway at five occasions, the probability that we would have flipped the switch by chance at all five of those occasions is just

$$\frac{1}{\frac{10!}{5!(10-5)!}} = \frac{1}{252} = 0.004.$$

In this case, the null hypothesis “flipping the switch has no effect on the lights” can be rejected with a p-value of 0.004.

This kind of reasoning allows us to draw causal inferences for events where the relationship is deterministic (A always leads to B and A^C always leads to B^C). But what if the relationship is probabilistic (e.g., A increases the probability of B)? And what if the outcome of interest is not necessarily a binary event, but a discrete or continuous measurement? We need a formal notation that allows us to express causal inference in terms of random variables.

Potential outcomes and average causal effects.

Suppose that, for a sample of n subjects, we observe the following two variables

$t_i =$ treatment given to subject i

(1=treated, 0=untreated)

$y_i =$ numeric outcome

Our question is, does t_i cause a change in the mean value of y_i ?

- Not sufficient to show a significant difference between

$$\bar{y}_1 = \frac{\sum_i t_i y_i}{\sum_i t_i} \quad \text{and} \quad \bar{y}_0 = \frac{\sum_i (1 - t_i) y_i}{\sum_i (1 - t_i)}$$

- Need to rule out alternative explanations (systematic differences between the groups other than t_i)
- Easy for randomized experiments, but challenging and controversial for observational studies, because in the latter we may have **confounders**
- Confounders are pre-treatment variables

$$x_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T$$

that may jointly influence t_i and y_i

Students of regression are taught to regress y_i on x_i and t_i and interpret the coefficient for t_i as the effect of t_i on y_i . This coefficient is

$$E(y_i | t_i = 1, x_i) - E(y_i | t_i = 0, x_i),$$

the difference in mean response between treated and untreated subjects when x_i is held constant

Is this a causal effect? Not really. In the regression approach, we are comparing **two different groups of individuals**. But causality is about changes in response when different treatments are applied **to the same individuals**.

Rubin (1974) introduced a notation of **potential outcomes**

$$\begin{aligned} t_i &= \text{treatment received by subject } i \text{ (0 or 1)} \\ y_{i0} &= \text{outcome for subject } i \text{ if } t_i = 0 \\ y_{i1} &= \text{outcome for subject } i \text{ if } t_i = 1 \\ d_i &= y_{i1} - y_{i0} \\ &= \text{causal effect for subject } i \end{aligned}$$

- Commonly known as Rubin's causal model (RCM) (Holland, 1986)
- Also attributed to Neyman (1923), Haavelmo (1944) and others

“Fundamental problem of causal inference” is that d_i can never be observed; one of the two potential outcomes is missing (Holland, 1986).

By making certain assumptions, however, it becomes possible to estimate the **average causal effect** for the population,

$$ACE = E(d_i) = E(y_{i1}) - E(y_{i0}).$$

If all potential outcomes were seen, we would estimate the ACE by

$$\hat{ACE} = \frac{1}{n} \sum_{i=1}^n d_i = \frac{1}{n} \sum_{i=1}^n y_{i1} - \frac{1}{n} \sum_{i=1}^n y_{i0}.$$

This estimate would be unbiased and highly efficient, but we cannot compute it in any real study.

Randomized experiments and observational studies. In a randomized experiment, t_i is **independent** of y_{i0} and y_{i1} . In that case, the average of the observed values of y_{i1} ,

$$\bar{y}_1 = \frac{\sum_i t_i y_{i1}}{\sum_i t_i}$$

is an unbiased estimate of $E(y_{i1})$. And the average of the observed values of y_{i0} ,

$$\bar{y}_0 = \frac{\sum_i (1 - t_i) y_{i0}}{\sum_i (1 - t_i)}$$

is an unbiased estimate of $E(y_{i0})$. So the difference $\bar{y}_1 - \bar{y}_0$

is an unbiased estimate of the ACE.

In an observational study, however, it is unlikely that t_i will be independent of y_{i0} and y_{i1} .

- Subjects may select their own treatments, for reasons that are possibly related to the outcomes.
- Treated ($t_i = 1$) and untreated ($t_i = 0$) groups are systematically (not just randomly) different at baseline.
- The characteristics on which they differ are potential confounders.

In observational studies, the difference $\bar{y}_1 - \bar{y}_0$ is generally a biased estimate of the ACE. Economists call this “selection bias.”

In an observational study, it is crucial to have good pre-treatment covariates to help us to understand and adjust for baseline differences between the groups.

ACE for the treated. An alternative to the ACE for the whole population is the ACE for the treated (Hirano & Imbens, 2001; Winship & Sobel, 2004),

$$ACE_1 = E(y_{i1} | t_i = 1) - E(y_{i0} | t_i = 1).$$

- Measures how much the treatment helped or hurt the individuals who actually received it
- May be more relevant than ACE_1 for discussing

implications of policy

- If $t_i = 1$ is a harmful behavior to be discouraged (e.g. substance use), then a highly effective intervention would switch all the values of $t_i = 1$ to $t_i = 0$, producing an average change of ACE_1 for those people and having no impact on the rest
- Sometimes data do not contain enough information to reliably estimate ACE , but they can be used to estimate ACE_1

Key assumptions needed to estimate ACE's. Any method for estimating ACE's will require lots of assumptions. The first assumption is that there is **no interference** between the subjects, in the sense that the treatment applied to one subject does not affect the outcome for any other subject. This is called the “stable unit treatment value assumption” (SUTVA) (Rubin, 1980). It may be violated if subjects interact in close proximity and the treatment given to one subject impacts others.

The second important assumption that we need to make is **unconfoundedness**. A treatment mechanism is said to be unconfounded given a set of covariates x_i if the treatment is conditionally independent of the potential outcomes given the covariates,

$$t_i \perp (y_{i0}, y_{i1}) \mid x_i.$$

- In a randomized experiment, t_i would be unconditionally independent of the potential outcomes; unconfoundedness is weaker.
- It means that all confounders are measured and available in x_i .
- This assumption becomes more plausible as the set of covariates in x_i grows.
- In real applications, we want to measure and adjust for a rich set of possible confounders, so that this assumption will be more believable.
- Unconfoundedness cannot be verified or contradicted based on the observed data. It can only be evaluated by expert knowledge of how t_i happened.
- Unconfoundedness may be violated if any variable in x_i is realized after the treatment and causally influenced by the treatment.
- We should not adjust for post-treatment variables as if they were confounders; doing so may introduce post-treatment selection bias (Rosenbaum, 1984).

SUTVA and unconfoundedness are helpful, but they are not enough to give us a method for estimating an ACE. To get an estimate, we will have to propose **at least one model**. That is, we will have to model

- the treatment mechanism (the distribution of t_i given x_i),

- the potential outcomes (the distributions of y_{i1} and y_{i0} given x_i),
- or both.

Different kinds of modeling assumptions will lead to different strategies for estimating ACE's.

Finally, we will have to assume that **the probabilities of receiving each treatment are bounded away from zero and one**. We must assume that each subject in the population could have been exposed to either treatment.

- If $P(t_i = 1) = 0$ or $P(t_i = 0) = 0$, then it's not meaningful to speak of a causal effect for that individual.
- y_{i1} and y_{i0} should both exist, at least in principle.
- If some subjects really had no chance to receive one of the treatments, then perhaps it's better to remove them from the population.

Simulated case study: the effect of dieting on emotional distress

- Dieting has not been shown to be an effective strategy for weight control (Hill, 2004; Katz, 2005)
- Associated with many negative outcomes: depression, anxiety, binge eating, anorexia, bulimia, etc.
- Cross-sectional studies consistently show that girls

who diet have higher levels of negative affect and psychological distress

- Causal link has not been supported by regression modeling of longitudinal data (Johnson & Wardle, 2005)

We will use a simulated observational study to examine causal effects of dieting on emotional distress from a standpoint of potential outcomes. This simulated example is based on the National Longitudinal Study of Adolescent Health (Add Health) (Udry, 2003).

- Nationally representative sample of middle and high-school students measuring health-related characteristics and behaviors
- Many features that make analysis challenging: students clustered within schools, unequal probabilities of selection at each stage, missing data and dropout
- We will use simulated data that mimics Add Health's variables, but not its sampling design, nonresponse or dropout

Using distributions estimated from Add Health Waves I (1994–95) and II (1995–96), we created an artificial population of one million adolescent girls. Then we

- Drew clean simple random samples of $N = 6,000$ girls each
- Applied procedures to estimate ACE's
- Repeated 5,000 times
- Observed how well each method reproduced known effects in the population

One sample of $N = 6,000$ and the code (R language) for performing all analyses have been placed at

<http://www.stat.psu.edu/~jls/causal/index.html>

In this study, the treatment variable is a question from Wave I indicating whether the girl had dieted in the last seven days to lose or maintain weight (20% said yes)

$$t_i = \begin{cases} 1 & \text{if dieted} \\ 0 & \text{if not} \end{cases}$$

The response variable is a composite measure of emotional distress based on 19-item feelings scale (each item scored 0–3)

y_{i1} = distress score if she dieted

y_{i0} = distress score if she did not

In this example, there are also thirteen confounders. These are variables measured at Wave I that are plausibly related to girls' decisions to diet:

- emotional distress score at baseline
- age, race, ethnicity
- self-perceived weight and fitness relative to peers, peer acceptance, self-esteem, etc.

Name	Description
DISTR.1	Emotional distress at Wave I
BLACK	1=Black, 0=otherwise
NBHISP	1=non-Black Hispanic, 0=otherwise
GRADE	Grade in school at Wave I (7, . . . , 11)
SLFHHLTH	Self-rating of overall health (1=excellent, 2=very good, 3=good, 4=fair, 5=poor)
SLFWGHT	Self-rating of weight (1=very underweight, 2=slightly under, 3=about right, 4=slightly over, 5=very over)
WORKHARD	“When you get what you want, it’s usually because you worked hard for it” (1=strongly agree, . . . , 5=strongly disagree)
GOODQUAL	“You have lots of good qualities” (1=strongly agree, . . . , 5=strongly disagree)
PHYSFIT	“You are physically fit” (1=strongly agree, . . . , 5=strongly disagree)
PROUD	“You have a lot to be proud of” (1=strongly agree, . . . , 5=strongly disagree)
LIKESLF	“You like yourself just the way you are” (1=strongly agree, . . . , 5=strongly disagree)
ACCEPTED	“You feel socially accepted” (1=strongly agree, . . . , 5=strongly disagree)
FEELLOVD	“You feel loved and wanted” (1=strongly agree, . . . , 5=strongly disagree)

- Distributions of all variables were estimated from 6,503 girls in Add Health grand sample who yielded usable diet-related data at Waves I and II
- We avoided simple parametric models (e.g. normal distributions and linear relationships) that are commonly assumed in data analyses
- We made special efforts to capture nonnormal shapes, nonlinearities and interactions in the simulated data
- All analyses, however, will be simple linear and logistic models that are only approximately true

If this were a real causal analysis of data from Add Health, it would have several important limitations

- Wording of dieting item is not optimal (“last seven days”)
- Confounders were measured at the same time as the treatment, and some could have been influenced by the treatment
- Many known correlates of dieting e.g., family and peer criticism of weight; thin-ideal internalization) were not measured

The first five girls in the population look like this.

Individual i	y_{i0}	y_{i1}	t_i	y_i	d_i
1	1.27	1.29	1	1.29	0.02
2	0.27	0.28	1	0.28	0.01
3	0.17	0.07	1	0.07	-0.10
4	2.18	2.44	0	2.18	0.26
5	1.05	0.97	0	1.05	-0.08
:	:	:	:	:	:

In this synthetic population, the true average causal effects are

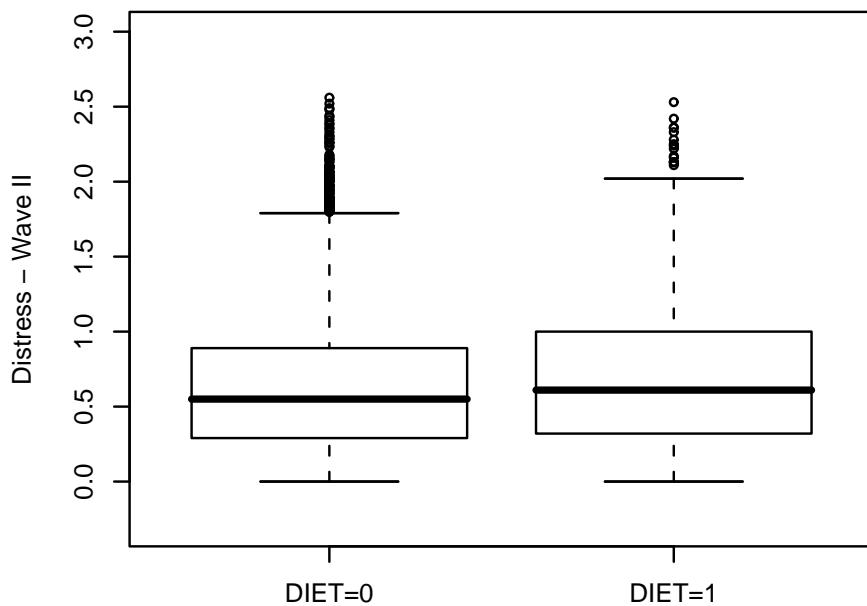
$$\text{ACE} = .003 \text{ essentially zero}$$

$$\text{ACE}_1 = -.022 \text{ slightly beneficial}$$

Over the next few lectures, we will review a large number of methods that have been proposed to estimate the ACE. We will also compare the methods and see how well they perform in this simulated example. The first method is a simple mean comparison. From there, we will move on to regression and analysis of covariance (ANCOVA) and methods based in propensity scores.

Method #1: Difference in means. Here are boxplots of the observed outcome (emotional distress at Wave II) for dieters and nondieters in the first sample of $N = 6,000$ girls.

(a) Distress at Wave II for non-dieters and dieters



The ordinary pooled two-sample t-test gives

$$\bar{y}_1 - \bar{y}_0 = .060 \quad \text{pooled SE} = .015 \quad p = .00$$

Girls who dieted exhibit higher distress at Wave II. But a similar significant difference was found in distress at Wave I (baseline). To get a fair comparison, we need to adjust for systematic differences in the baseline variables. If we don't, the estimate of the ACE will be extremely biased.

Over all 5,000 samples, the difference in means is a badly biased estimate of $ACE = .003$. Here are some summary measures of performance.

$$\begin{aligned} \text{bias} &= .052 \\ \text{SD} &= .016 \\ \text{RMSE} &= .054 \\ \text{coverage} &= 8\% \end{aligned}$$

- Bias is the average of the 5,000 estimates minus the true population value (closer to zero is better).
- SD is the standard deviation of the 5,000 estimates (smaller is better).
- RMSE is the square root of the average squared difference between the 5,000 estimates and the true value (smaller is better).
- Coverage is the percentage of nominal 95% intervals (estimate plus or minus 1.96 SD's) that covered the true value (should be close to 95%).

The difference in means is a disaster. The estimates are so badly biased that virtually all of the confidence intervals miss the true ACE.

INTRODUCTION TO CAUSAL INFERENCE (PART II)

Last time, we introduced the notation of potential outcomes:

- t_i = treatment received by subject i (0 or 1)
- y_{i0} = outcome for subject i if $t_i = 0$
- y_{i1} = outcome for subject i if $t_i = 1$
- d_i = $y_{i1} - y_{i0}$
= causal effect for subject i
- x_i = vector of possible confounders

The average causal effect for the population is

$$ACE = E(d_i) = E(y_{i1}) - E(y_{i0}),$$

and the average causal effect for the treated is

$$ACE_1 = E(y_{i1} \mid t_i = 1) - E(y_{i0} \mid t_i = 1).$$

Inferences about these quantities are not straightforward, because

- y_{i1} is observed if $t_i = 1$ and missing if $t_i = 0$, and
- y_{i0} is observed if $t_i = 0$ and missing if $t_i = 1$.

Any estimate of ACE or ACE_1 must be based only on the observed data,

$$(x_i, t_i, y_i), \quad i = 1, \dots, n,$$

where

$$y_i = t_i y_{i1} + (1 - t_i) y_{i0}$$

is the outcome actually observed for subject i .

In order to proceed, we had to make a number of assumptions. One key assumption is **unconfoundedness**, which means that t_i is conditionally independent of the potential outcomes (y_{i0}, y_{i1}) given x_i . Unconfoundedness means that all confounders have been measured and are available in x_i . In many observational studies, this assumption is unlikely to hold. But it might be a reasonable approximation if x_i contains a rich set of covariates related to t_i .

A simple and naive estimate of ACE is

$$\bar{y}_1 - \bar{y}_0 = \frac{\sum_i t_i y_i}{\sum_i t_i} - \frac{\sum_i (1 - t_i) y_i}{\sum_i (1 - t_i)}.$$

In our simulated example, this estimate did not perform well:

bias	=	.052
SD	=	.016
RMSE	=	.054
coverage	=	8%

How do we interpret these results?

- bias is the average value of the estimates minus the true value
- SD is the standard deviation of the estimates (SE is an estimate of this)
- RMSE is root mean squared error, the average distance between the estimate and the truth (combines bias and variance)

$$\text{RMSE} = \sqrt{\frac{1}{5,000} \sum_{j=1}^{5,000} (A\hat{C}E^{(j)} - ACE)^2}$$

- “coverage” is the actual coverage rate of a nominal 95% confidence interval (estimate \pm 1.96 SE’s)

Here is a useful rule-of-thumb: If the bias in an estimate exceeds 40–50% of its SE, then the bias adversely affects tests and intervals. In this case, the bias is about $.052/.016 = 3.25$ SE’s, so it’s a disaster.

Note that as N grows, SE’s shrink. So the impact of bias becomes worse in large samples.

For the rest of today’s lecture, we will discuss traditional regression modeling to understand when a regression coefficient estimates an ACE. And we will discuss an alternative to regression coefficients, which I call “regression estimation.”

Method #2: ANCOVA. Analysis of covariance (ANCOVA) is a traditional name for comparing two groups by fitting a regression model that includes a dummy indicator to distinguish among the groups, plus additional covariates to reflect the fact that the groups are different in order to adjust for those differences.

The simple linear version of ANCOVA assumes that

$$y_i = \alpha + \theta t_i + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \epsilon_i.$$

Collecting the x 's and β 's into vectors,

$$E(y_i | t_i, x_i) = \alpha + t_i \theta + x_i^T \beta.$$

The coefficients α , θ and β are typically estimated by ordinary least squares (OLS). This is the most common version of ANCOVA, but we may also consider a more general class of regression models that may include nonlinear relationships, interactions and heteroscedastic errors.

In our dieting example, we have thirteen potential confounders that we can include in x_i . The most important one is the measure of emotional distress at baseline. If we take x_i to be equal to this single variable, then, in our first sample of $n = 6,000$ girls, the estimate is

$$\hat{\theta} = .003 \quad (\text{SE} = .012).$$

Using all thirteen covariates, the estimate becomes

$$\hat{\theta} = -.014 \quad (\text{SE} = .013).$$

ANCOVA was developed by R.A. Fisher in the 1930's for comparing groups in randomized trials. The original purpose of ANCOVA was to increase the precision (i.e. reduce the variance) of estimated treatment effects by including pre-test measures of variables that would be highly correlated with the outcome. In a randomized experiment, we don't have to use ANCOVA, because the simple difference in means is already an unbiased estimate of the ACE. But including covariates can be beneficial, because it makes the estimate more precise. ANCOVA for randomized experiments will be covered in detail in Stat 512.

After ANCOVA was invented, however, researchers also began to apply it to data from observational studies. In those contexts, the main purpose is not to increase precision but to reduce bias(Cook & Campbell, 1979; Cochran, 1983). But in situations where the treatment groups are very different at baseline, ANCOVA may not work very well.

The main problem with using ANCOVA for causal inference in observational studies is that the regression coefficient θ is not necessarily an average causal effect. The crucial difference is that

- θ conditions on the covariates, but
- the ACE averages over the covariates.

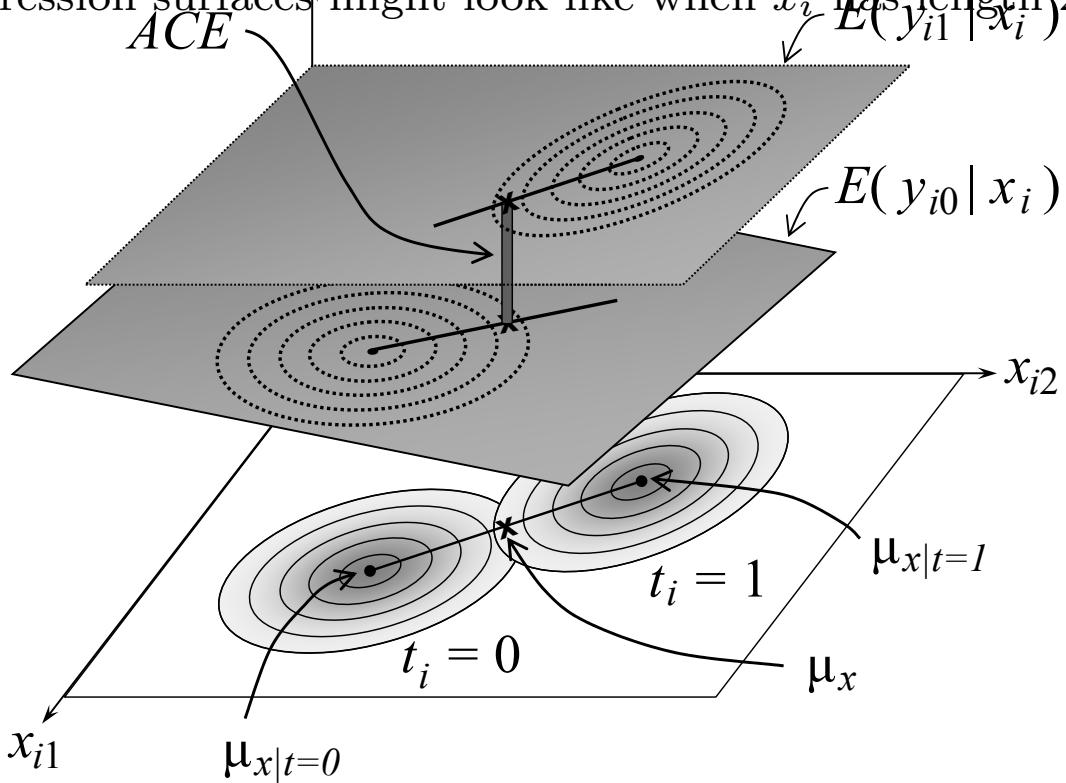
Suppose for a moment that the potential outcomes are

linearly related to the covariates,

$$E(y_{i0} | x_i) = \alpha_0 + x_i^T \beta_0,$$

$$E(y_{i1} | x_i) = \alpha_1 + x_i^T \beta_1,$$

where β_0 and β_1 are vectors of slopes. This what the regression surfaces might look like when x_i has length 2.



In this picture, the ellipses represent the bivariate distributions of the covariates in each group. If this were a randomized experiment, the distributions would coincide, because each group would be a random sample from the

same population. In observational studies, however, these groups may be far apart. Let

$$\begin{aligned}\mu_{x|t=1} &= \text{average value of covariates among treated,} \\ \mu_{x|t=0} &= \text{average value of covariates among untreated,} \\ \mu_x &= \text{average value of covariates in population.}\end{aligned}$$

From the identity $E(a) = E(E(a | b))$, it follows that

$$\begin{aligned}ACE &= E(y_{i1} | x_i = \mu_x) - E(y_{i0} | x_i = \mu_x) \\ &= [\alpha_1 - \alpha_0] + [\beta_1 - \beta_0]^T \mu_x\end{aligned}$$

- ACE is the vertical distance between the planes at $x_i = \mu_x$
- ACE_1 is the vertical distance between the planes at $x_i = \mu_{x|t=1}$

If the planes are exactly parallel (all slopes in β_0 equal the slopes in β_1) then $ACE = ACE_1$.

To obtain a stable estimate of ACE , we need the distributions of x_i when $t_i = 1$ and $t_i = 0$ to **sufficiently overlap**

- $E(y_{i1} | x_i)$ can be estimated best in the vicinity of $\mu_{x|t=1}$, and $E(y_{i0} | x_i)$ can be estimated best in the vicinity of $\mu_{x|t=0}$
- As we move away from the group means, the estimated surfaces become less stable and more sensitive to model misspecification (extrapolation)

- To estimate the population ACE well, we need both regression surfaces to be well estimated in the vicinity of μ_x .

The crucial question that we want to answer is: **When does ANCOVA estimate the ACE?**

- The treatment effect in ANCOVA is the coefficient θ in the model

$$E(y_i | t_i, x_i) = \alpha + t_i\theta + x_i^T \beta,$$

but

- the ACE is the population average difference between y_{i1} and y_{i0} .

The two coincide only in special cases. Recall that the ANCOVA response variable is

$$y_i = t_i y_{i1} + (1 - t_i) y_{i0}.$$

So the ANCOVA model claims that

$$E(t_i y_{i1} + (1 - t_i) y_{i0} | t_i, x_i) = \alpha + t_i\theta + x_i^T \beta \quad (1)$$

Now suppose that the treatment is unconfounded given x_i and that the potential outcomes are linearly related to the covariates,

$$\begin{aligned} E(y_{i0} | x_i) &= E(y_{i0} | t_i, x_i) = \alpha_0 + x_i^T \beta_0, \\ E(y_{i1} | x_i) &= E(y_{i1} | t_i, x_i) = \alpha_1 + x_i^T \beta_1, \end{aligned}$$

so that the average causal effect is

$$ACE = [\alpha_1 - \alpha_0] + [\beta_1 - \beta_0]^T \mu_x$$

Under these same assumptions,

$$\begin{aligned} E(t_i y_{i1} + (1 - t_i) y_{i0} \mid t_i, x_i) &= \alpha_0 + t_i [\alpha_1 - \alpha_0] \\ &\quad + x_i^T \beta_0 + t_i x_i^T [\beta_1 - \beta_0]. \end{aligned}$$

Comparing these last two formulas to (1), we see that the ANCOVA coefficient θ becomes the ACE when **the slopes for all covariates are equal across groups** ($\beta_0 = \beta_1$), i.e. the regression planes are parallel.

If the regression slopes are not equal across groups, we can make the ANCOVA parameter θ correspond to the ACE if we

- **center the covariates** at their mean values in the population (replace x_i by $x_i - \mu_x$), and
- expand the ANCOVA model to include **all baseline-by-treatment interactions**,

$$E(y_i \mid t_i, x_i) = \alpha + t_i \theta + (x_i - \mu_x)^T \beta + t_i (x_i - \mu_x)^T \eta.$$

To make θ correspond to ACE_1 , we must center the covariates at their mean values in the **treated** population,

$$E(y_i \mid t_i, x_i) = \alpha + t_i \theta + (x_i - \mu_{x|t=1})^T \beta + t_i (x_i - \mu_{x|t=1})^T \eta.$$

This discussion helps us to understand some of the pitfalls of using ANCOVA for causal inference in observational studies. If baseline-by-treatment interactions exist but are not included in the model, then $\hat{\theta}$ will be a biased estimate of the ACE.

- This is true even if we only want the average treatment effect and don't care about interactions.
- The bias gets worse if one of the groups is much larger than the other.
- The bias gets worse if the means $\mu_{x|t=1}$ and $\mu_{x|t=0}$ are "far apart" (later we'll give a rule).

It's tempting to include interactions only if they are statistically significant, but that strategy is dangerous, especially if one of the groups is small (because the power to detect interactions will be low).

And we must not forget the assumption of unconfoundedness.

- ANCOVA is based on the assumption that there are no unmeasured confounders.
- Without this assumption, θ will not correspond to an ACE, even if in every other respect the model is correct.
- Omitting variables that are related to t_i , even if they are not statistically significant predictors of y_i , may violate this assumption.

- Covariates may appear insignificant precisely because they are related to t_i (sharing significance). But that does not mean we should omit them. Rather, we should be including them, precisely because they are related to t_i .
- If you try ANCOVA for causal inference in observational studies, don't use statistical significance for deciding whether to include a covariate (Winship & Morgan, 1999).
- Standard variable-selection criteria (Mallows C_p , PRESS, etc.) are designed to find a model that gives good predictions for y_i ; but for causal inference, we need unbiased prediction of the missing values of y_{i1} and y_{i0} , which is a very different objective.

How well does ANCOVA perform in our dieting example?
In our initial simulated sample of $N = 6,000$,

- ANCOVA with main effects for all covariates gave $\hat{ACE} = -.014$ (SE=.013).
- Including all baseline-by-treatment interactions and centering the covariates at their sample means, we get $\hat{ACE} = -.006$ (SE=.016).
- Including all baseline-by-treatment interactions and centering the covariates at their sample means among the treated, we get $\hat{ACE}_1 = -.015$ (SE=.013).

Performance over 5,000 repeated samples:

<i>Method</i>	$ACE = .003$			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	.052	.016	.054	8.7
ANCOVA (main effects)	−.016	.014	.021	76.7
ANCOVA (interactions)	−.004	.015	.016	94.2

<i>Method</i>	$ACE_1 = -.022$			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	—	—	—	—
ANCOVA (main effects)	.009	.014	.016	89.0
ANCOVA (interactions)	.006	.014	.015	91.6

In this example, ANCOVA with main effects is a disaster for ACE . ANCOVA with interactions works well for ACE and ACE_1 . This illustrates the importance of including baseline-by-treatment interactions when performing causal inference for observational studies.

Some final comments about ANCOVA for observational studies:

- In this example, we did not carefully check our model for nonlinear relationships or heteroscedasticity.
- Heteroscedasticity won't introduce bias, but it makes OLS inefficient and may distort SE's.
- SE's can be corrected using "robust" or "empirical" method (sandwich formula) which we will learn about later.
- The most dangerous aspect of ANCOVA is that analysts are often unaware how different the treated and untreated groups are, and don't realize when they are extrapolating.
- When many covariates are available, we need a good strategy for variable reduction.
- Best variable-reduction tool is the propensity score, which we will learn about next time.

Method #3: Regression estimation. As an alternative to ANCOVA, which estimates an ACE by a regression coefficient, we can compute estimates of ACE's using **regression predictions**.

Suppose we

- Regress y_{i1} on x_i among **treated** persons, and
- use model to compute predictions \hat{y}_{i1} 's for **everyone**.

Then the average of the \hat{y}_{i1} 's is an estimate of $E(y_{i1})$.

Similarly, we can

- Regress y_{i0} on x_i among **untreated** persons, and
- use the model to compute \hat{y}_{i0} 's for **everyone**.

Then the average of the \hat{y}_{i0} 's is an estimate of $E(y_{i0})$.

The difference between these averages is an estimate of *ACE*.

This approach is conceptually very different from ANCOVA.

- A regression coefficient is a slope in a regression model for the conditional mean of a response given covariates
- A regression estimate is an average of regression predictions (covariate-assisted estimate of a population mean)

The simplest way to perform regression estimation is to use linear models and OLS. We would split the sample and regress y_i on x_i separately for treated ($t_i = 1$) and untreated ($t_i = 0$) persons.

$$\begin{aligned}\hat{\beta}_1 &= \left(\sum_i t_i x_i x_i^T \right)^{-1} \left(\sum_i t_i x_i y_i \right) \\ \hat{\beta}_0 &= \left(\sum_i (1 - t_i) x_i x_i^T \right)^{-1} \left(\sum_i (1 - t_i) x_i y_i \right).\end{aligned}$$

The regression estimate of ACE is

$$\hat{ACE} = \frac{1}{N} \sum_i (\hat{y}_{i1} - \hat{y}_{i0}),$$

where $\hat{y}_{i1} = x_i^T \hat{\beta}_1$ is a prediction based on the first model, and $\hat{y}_{i0} = x_i^T \hat{\beta}_0$ is a prediction based the second model.

Actually, we don't have to substitute regression predictions for the potential outcomes that are known. If we use predictions only for the missing values, we get

$$\hat{ACE} = \frac{1}{N} \sum_i \{ t_i (y_i - \hat{y}_{i0}) + (1 - t_i) (\hat{y}_{i1} - y_i) \},$$

which is an example of conditional mean imputation (Little & Rubin, 2002; Schafer & Schenker, 2000). With linear models fit by OLS, this equation and the previous one are equivalent (a consequence of the fact that OLS residuals sum to zero).

If our goal is to estimate an ACE among the treated, then the regression model for y_{i1} becomes unnecessary. A regression estimate for ACE_1 is

$$\hat{ACE}_1 = \frac{\sum_i t_i(y_i - \hat{y}_{i0})}{\sum_i t_i},$$

where $\hat{y}_{i0} = x_i^T \hat{\beta}_0$ is a prediction based on a model fit to the untreated.

How does regression estimation compare with ANCOVA?
If

- the same baseline measures appear in the regression models for y_{i1} and y_{i0} , and
- both models are linear,

then regression estimation is equivalent to ANCOVA with all baseline-by-treatment interactions.

But I prefer regression estimation because:

- It forces the analyst to acknowledge that y_{i1} and y_{i0} are two different variables.
- By splitting the sample, all baseline-by-treatment interactions are included automatically.
- SE formulas for regression estimates (see Appendix) are robust to departures from homoscedasticity.

How does regression estimation perform in the simulated example?

$ACE = .003$				
<i>Method</i>	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	.052	.016	.054	8.7
ANCOVA (main effects)	−.016	.014	.021	76.7
ANCOVA (interactions)	−.004	.015	.016	94.2
Regression estimation	−.004	.015	.016	94.3

$ACE_1 = -.022$				
<i>Method</i>	Bias	SD	RMSE	Cvg.
Difference in means (t-test) —	—	—	—	—
ANCOVA (main effects)	.009	.014	.016	89.0
ANCOVA (interactions)	.006	.014	.015	91.6
Regression estimation	.006	.014	.015	93.2

- Estimates are the same as ANCOVA with interactions, but SE's are a little different.
- Coverage of confidence intervals is a little better with regression estimation.

Next time: Introduction to propensity scores

INTRODUCTION TO CAUSAL INFERENCE (PART III)

Last time, we described the difference between a standard regression (ANCOVA) coefficient and an average causal effect. Standard regression fits a linear model to the observed response

$$y_i = t_i y_{i1} + (1 - t_i) y_{i0},$$

with t_i and x_i (and possibly interactions among them) as predictors, and the coefficient of t_i is interpreted as the effect of the treatment. The simplest version of the model is

$$y_i = \alpha + t_i \theta + x_i^T \beta + \text{error}. \quad (1)$$

We have argued that the parameter θ in model (1) is the average causal effect in the population only under the special conditions that

- the treatment mechanism is unconfounded given x_i ,
- the potential outcomes y_{i1} and y_{i0} are linearly related to x_i , and
- the slopes of those two regressions (y_{i1} on x_i and y_{i0} on x_i) are equal.

If the slopes of the two regression models are not all equal, then we can still use ANCOVA to estimate the ACE, but we must

- center each covariate in x_i at the mean of the population being considered, and
- expand the regression model (1) to include the interactions between each x_i and t_i .

The expanded model is

$$y_i = \alpha + t_i \theta + (x_i - \mu_x)^T \beta + t_i (x_i - \mu_x)^T \eta + \text{error}.$$

In practice, μ_x is unknown and must be replaced by the sample estimate

$$\mu_x = \frac{1}{n} \sum_{i=1}^n x_i.$$

Let's illustrate this with our dieting example. Let's read in the first sample of $n = 6,000$ subjects from the artificial population, which is posted at

<http://www.stat.psu.edu/~jls/causal/index.html>

```
> # read in the data and assign variable names
> samp <- read.table("Diet0001.dat", row.names="ID",
+   col.names=c("ID", "DISTR.1", "BLACK", "NBHISP", "GRADE", "SLFHLTH",
+   "SLFWGHT", "WORKHARD", "GOODQUAL", "PHYSFIT", "PROUD",
+   "LIKESLF", "ACCEPTED", "FEELLOVD", "DIET", "Y.0", "Y.1", "DISTR.2",
+   "PI.TRUE"))
```

In this simulated dataset, the treatment variable t_i is DIET (0=no dieting, 1=dieting) and the observed outcome y_i

(emotional distress at Wave II) is DISTR.2. The variables Y.0 and Y.1 are the potential outcomes, which would not be observable if this were a real study. All of the other variables (except PI.TRUE, which will be explained later) are possible confounders. First, let's fit model (1) by regressing y_i on t_i and all of the possible confounders.

```
> fit <- lm( DISTR.2 ~ DISTR.1 + BLACK + NBHISP + GRADE +
+   SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD +
+   LIKESLF + ACCEPTED + FEELLOVD + DIET, data=samp)
> summary(fit)

Call:
lm(formula = DISTR.2 ~ DISTR.1 + BLACK + NBHISP + GRADE + SLFHLTH +
   SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD + LIKESLF +
   ACCEPTED + FEELLOVD + DIET, data = samp)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.69102 -0.22781 -0.05734  0.18025  1.81213 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.0052087  0.0422214 -0.123  0.901821  
DISTR.1      0.5178247  0.0125785 41.168 < 2e-16 *** 
BLACK        0.0745764  0.0120532  6.187 6.53e-10 *** 
NBHISP       0.0289905  0.0141752  2.045 0.040884 *   
GRADE        0.0019352  0.0035672  0.543 0.587490  
SLFHLTH     0.0196974  0.0059809  3.293 0.000996 *** 
SLFWGHT      0.0038613  0.0069990  0.552 0.581183  
WORKHARD     -0.0121711  0.0056656 -2.148 0.031734 *   
GOODQUAL     0.0209810  0.0098513  2.130 0.033231 *   
PHYSFIT      -0.0005642  0.0069283 -0.081 0.935099  
PROUD        0.0376379  0.0101757  3.699 0.000219 *** 
LIKESLF      0.0242944  0.0064040  3.794 0.000150 *** 
ACCEPTED     0.0167152  0.0068032  2.457 0.014040 *   
FEELLOVD     0.0389178  0.0085254  4.565 5.10e-06 *** 
DIET         -0.0136658  0.0129321 -1.057 0.290675  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.3758 on 5985 degrees of freedom
Multiple R-Squared:  0.3513,    Adjusted R-squared:  0.3498
F-statistic: 231.5 on 14 and 5985 DF,  p-value: < 2.2e-16
```

The estimated effect of dieting is -0.014 ($SE=0.013$). This estimate makes the strong assumption that the regressions of y_{i1} and y_{i0} on x_i are linear and parallel. If that assumption were true, then the coefficient of DIET in this model would be an unbiased estimate of both ACE and ACE_1 . But if that assumption is wrong, then ACE and ACE_1 would not be the same, and the coefficient of DIET would not be an unbiased estimate of either one.

Let's relax the assumption of parallelism by adding the interactions to the model. First, let's center each covariate at its sample mean and re-fit model (1).

```
> sampc <- samp
> sampc$DISTR.1 <- sampc$DISTR.1 - mean(sampc$DISTR.1)
> sampc$BLACK <- sampc$BLACK - mean(sampc$BLACK)
> sampc$NBHISP <- sampc$NBHISP - mean(sampc$NBHISP)
> sampc$GRADE <- sampc$GRADE - mean(sampc$GRADE)
> sampc$SLFHLTH <- sampc$SLFHLTH - mean(sampc$SLFHLTH)
> sampc$SLFWGHT <- sampc$SLFWGHT - mean(sampc$SLFWGHT)
> sampc$WORKHARD <- sampc$WORKHARD - mean(sampc$WORKHARD)
> sampc$GOODQUAL <- sampc$GOODQUAL - mean(sampc$GOODQUAL)
> sampc$PHYSFIT <- sampc$PHYSFIT - mean(sampc$PHYSFIT)
> sampc$PROUD <- sampc$PROUD - mean(sampc$PROUD)
> sampc$LIKESLF <- sampc$LIKESLF - mean(sampc$LIKESLF)
> sampc$ACCEPTED <- sampc$ACCEPTED - mean(sampc$ACCEPTED)
> sampc$FEELLOVD <- sampc$FEELLOVD - mean(sampc$FEELLOVD)
>
>
> fit <- lm( DISTR.2 ~ DISTR.1 + BLACK + NBHISP + GRADE +
+           SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD +
```

```

+ LIKESLF + ACCEPTED + FEELLOVD + DIET, data=sampc)
> summary(fit)

Call:
lm(formula = DISTR.2 ~ DISTR.1 + BLACK + NBHISP + GRADE + SLFHLTH +
    SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD + LIKESLF +
    ACCEPTED + FEELLOVD + DIET, data = sampc)

Residuals:
    Min      1Q   Median      3Q     Max 
-1.69102 -0.22781 -0.05734  0.18025  1.81213 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.6584254  0.0055186 119.311 < 2e-16 ***
DISTR.1     0.5178247  0.0125785  41.168 < 2e-16 ***
BLACK        0.0745764  0.0120532   6.187 6.53e-10 ***
NBHISP       0.0289905  0.0141752   2.045 0.040884 *  
GRADE        0.0019352  0.0035672   0.543 0.587490  
SLFHLTH     0.0196974  0.0059809   3.293 0.000996 *** 
SLFWGHT      0.0038613  0.0069990   0.552 0.581183  
WORKHARD     -0.0121711  0.0056656  -2.148 0.031734 *  
GOODQUAL     0.0209810  0.0098513   2.130 0.033231 *  
PHYSFIT      -0.0005642  0.0069283  -0.081 0.935099  
PROUD        0.0376379  0.0101757   3.699 0.000219 *** 
LIKESLF      0.0242944  0.0064040   3.794 0.000150 *** 
ACCEPTED     0.0167152  0.0068032   2.457 0.014040 *  
FEELLOVD     0.0389178  0.0085254   4.565 5.10e-06 *** 
DIET         -0.0136658  0.0129321  -1.057 0.290675  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3758 on 5985 degrees of freedom
Multiple R-Squared: 0.3513,    Adjusted R-squared: 0.3498 
F-statistic: 231.5 on 14 and 5985 DF,  p-value: < 2.2e-16

```

In this centered version of the model, only the intercept has changed. Now let's add the interactions to the model.

```

> fit <- lm( DISTR.2 ~ DIET +
+   DISTR.1 + BLACK + NBHISP + GRADE +

```

```

+   SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD +
+   LIKESLF + ACCEPTED + FEELLOVD +
+   DISTR.1:DIET + BLACK:DIET + NBHISP:DIET + GRADE:DIET +
+   SLFHLTH:DIET + SLFWGHT:DIET + WORKHARD:DIET + GOODQUAL:DIET +
+   PHYSFIT:DIET + PROUD:DIET +
+   LIKESLF:DIET + ACCEPTED:DIET + FEELLOVD:DIET,
+   data=sampc)

> summary(fit)

Call:
lm(formula = DISTR.2 ~ DIET + DISTR.1 + BLACK + NBHISP + GRADE +
SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD +
LIKESLF + ACCEPTED + FEELLOVD + DISTR.1:DIET + BLACK:DIET +
NBHISP:DIET + GRADE:DIET + SLFHLTH:DIET + SLFWGHT:DIET +
WORKHARD:DIET + GOODQUAL:DIET + PHYSFIT:DIET + PROUD:DIET +
LIKESLF:DIET + ACCEPTED:DIET + FEELLOVD:DIET, data = sampc)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.72288 -0.22801 -0.05723  0.18175  1.82660 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.6587668  0.0055344 119.031 < 2e-16 ***
DIET        -0.0058574  0.0149019  -0.393 0.694286  
DISTR.1     0.5210525  0.0143816  36.230 < 2e-16 ***
BLACK       0.0742012  0.0131869   5.627 1.92e-08 ***
NBHISP      0.0241334  0.0159814   1.510 0.131071  
GRADE       0.0021585  0.0039652   0.544 0.586210  
SLFHLTH    0.0257745  0.0067034   3.845 0.000122 ***
SLFWGHT    0.0026900  0.0077575   0.347 0.728781  
WORKHARD   -0.0194426  0.0062698  -3.101 0.001938 ** 
GOODQUAL   0.0154691  0.0110400   1.401 0.161212  
PHYSFIT    0.0054135  0.0077812   0.696 0.486635  
PROUD      0.0329261  0.0113575   2.899 0.003756 ** 
LIKESLF    0.0225898  0.0072863   3.100 0.001942 ** 
ACCEPTED   0.0159821  0.0077195   2.070 0.038462 *  
FEELLOVD   0.0412613  0.0095958   4.300 1.74e-05 ***
DIET:DISTR.1 -0.0106838  0.0296963  -0.360 0.719033  
DIET:BLACK   -0.0003361  0.0327604  -0.010 0.991814  
DIET:NBHISP   0.0238344  0.0346707   0.687 0.491826  
DIET:GRADE    0.0023927  0.0091386   0.262 0.793465  
DIET:SLFHLTH -0.0297672  0.0150042  -1.984 0.047310 * 

```

```

DIET:SLFWGHT -0.0011506 0.0181310 -0.063 0.949404
DIET:WORKHARD 0.0392602 0.0147076 2.669 0.007620 **
DIET:GOODQUAL 0.0260125 0.0246058 1.057 0.290477
DIET:PHYSFIT -0.0266321 0.0171818 -1.550 0.121190
DIET:PROUD    0.0240841 0.0255893 0.941 0.346650
DIET:LIKESLF  0.0086665 0.0153017 0.566 0.571161
DIET:ACCEPTED -0.0015125 0.0164325 -0.092 0.926664
DIET:FEELLOVD -0.0125126 0.0209234 -0.598 0.549850
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3756 on 5972 degrees of freedom
Multiple R-Squared: 0.3534,      Adjusted R-squared: 0.3505
F-statistic: 120.9 on 27 and 5972 DF, p-value: < 2.2e-16

```

The estimate of *ACE* is now -0.006 ($SE=0.015$). Adding the interactions made the estimate of *ACE* a little closer to zero and the standard error a little larger.

Are these interactions necessary? Looking at the table of coefficients, we see that only two of the thirteen interactions are significantly different from zero at the 0.05 level. We can test the null hypothesis that all interactions are unnecessary by a partial F-test. The error sum of squares for the smaller model is

$$(0.3758)^2 \times 5985 = 845.236.$$

The error sum of squares for the larger model is

$$(0.3756)^2 \times 5972 = 842.502.$$

The partial F-test statistic is

$$F = \frac{(845.236 - 842.502)/13}{0.3756^2} = 1.49,$$

and the P-value is

$$P(F_{13,5972} \geq 1.49) = 0.122.$$

The thirteen interactions are not jointly significant. Yet I would argue that they should be included in the model anyway, because without them the estimate of ACE could be badly biased.

We can use a similar procedure to estimate ACE_1 , the average causal effect of dieting among girls who actually diet. To estimate ACE_1 , we simply re-center the covariates at their mean values among the dieters, and re-fit the model with interactions.

```
> # center the covariates at means among the dieters
> sampt <- samp
> sampt$DISTR.1 <- sampt$DISTR.1 - mean(sampt$DISTR.1[sampt$DIET==1])
> sampt$BLACK <- sampt$BLACK - mean(sampt$BLACK[sampt$DIET==1])
> sampt$NBHISP <- sampt$NBHISP - mean(sampt$NBHISP[sampt$DIET==1])
> sampt$GRADE <- sampt$GRADE - mean(sampt$GRADE[sampt$DIET==1])
> sampt$SLFHLTH <- sampt$SLFHLTH - mean(sampt$SLFHLTH[sampt$DIET==1])
> sampt$SLFWGHT <- sampt$SLFWGHT - mean(sampt$SLFWGHT[sampt$DIET==1])
> sampt$WORKHARD <- sampt$WORKHARD - mean(sampt$WORKHARD[sampt$DIET==1])
> sampt$GOODQUAL <- sampt$GOODQUAL - mean(sampt$GOODQUAL[sampt$DIET==1])
> sampt$PHYSFIT <- sampt$PHYSFIT - mean(sampt$PHYSFIT[sampt$DIET==1])
> sampt$PROUD <- sampt$PROUD - mean(sampt$PROUD[sampt$DIET==1])
> sampt$LIKESLF <- sampt$LIKESLF - mean(sampt$LIKESLF[sampt$DIET==1])
> sampt$ACCEPTED <- sampt$ACCEPTED - mean(sampt$ACCEPTED[sampt$DIET==1])
> sampt$FEELLOVD <- sampt$FEELLOVD - mean(sampt$FEELLOVD[sampt$DIET==1])

> # fit the model to estimate ACE.1
> fit <- lm( DISTR.2 ~ DIET +
+   DISTR.1 + BLACK + NBHISP + GRADE +
+   SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD +
+   LIKESLF + ACCEPTED + FEELLOVD +
+   DISTR.1:DIET + BLACK:DIET + NBHISP:DIET + GRADE:DIET +
```

```

+      SLFHLTH:DIET + SLFWGHT:DIET + WORKHARD:DIET + GOODQUAL:DIET +
+      PHYSFIT:DIET + PROUD:DIET +
+      LIKESLF:DIET + ACCEPTED:DIET + FEELLOVD:DIET,
+      data=sampt)

> summary(fit)

Call:
lm(formula = DISTR.2 ~ DIET + DISTR.1 + BLACK + NBHISP + GRADE +
    SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD +
    LIKESLF + ACCEPTED + FEELLOVD + DISTR.1:DIET + BLACK:DIET +
    NBHISP:DIET + GRADE:DIET + SLFHLTH:DIET + SLFWGHT:DIET +
    WORKHARD:DIET + GOODQUAL:DIET + PHYSFIT:DIET + PROUD:DIET +
    LIKESLF:DIET + ACCEPTED:DIET + FEELLOVD:DIET, data = sampt)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.72288 -0.22801 -0.05723  0.18175  1.82660 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.7184844  0.0075152 95.604 < 2e-16 ***
DIET        -0.0153450  0.0131194 -1.170 0.242192  
DISTR.1     0.5210525  0.0143816 36.230 < 2e-16 ***
BLACK       0.0742012  0.0131869  5.627 1.92e-08 ***
NBHISP      0.0241334  0.0159814  1.510 0.131071  
GRADE       0.0021585  0.0039652  0.544 0.586210  
SLFHLTH    0.0257745  0.0067034  3.845 0.000122 ***
SLFWGHT    0.0026900  0.0077575  0.347 0.728781  
WORKHARD   -0.0194426  0.0062698 -3.101 0.001938 ** 
GOODQUAL   0.0154691  0.0110400  1.401 0.161212  
PHYSFIT    0.0054135  0.0077812  0.696 0.486635  
PROUD      0.0329261  0.0113575  2.899 0.003756 ** 
LIKESLF    0.0225898  0.0072863  3.100 0.001942 ** 
ACCEPTED   0.0159821  0.0077195  2.070 0.038462 *  
FEELLOVD   0.0412613  0.0095958  4.300 1.74e-05 ***
DIET:DISTR.1 -0.0106838  0.0296963 -0.360 0.719033  
DIET:BLACK   -0.0003361  0.0327604 -0.010 0.991814  
DIET:NBHISP   0.0238344  0.0346707  0.687 0.491826  
DIET:GRADE    0.0023927  0.0091386  0.262 0.793465  
DIET:SLFHLTH  -0.0297672  0.0150042 -1.984 0.047310 *  
DIET:SLFWGHT  -0.0011506  0.0181310 -0.063 0.949404  
DIET:WORKHARD  0.0392602  0.0147076  2.669 0.007620 ** 
DIET:GOODQUAL  0.0260125  0.0246058  1.057 0.290477 

```

```

DIET:PHYSFIT -0.0266321  0.0171818  -1.550 0.121190
DIET:PROUD     0.0240841  0.0255893   0.941 0.346650
DIET:LIKESLF   0.0086665  0.0153017   0.566 0.571161
DIET:ACCEPTED  -0.0015125  0.0164325  -0.092 0.926664
DIET:FEELLOVD -0.0125126  0.0209234  -0.598 0.549850
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3756 on 5972 degrees of freedom
Multiple R-Squared: 0.3534,      Adjusted R-squared: 0.3505
F-statistic: 120.9 on 27 and 5972 DF,  p-value: < 2.2e-16

```

The estimate of ACE_1 is -0.015 (SE=0.013).

How well do these methods perform over the 5,000 repeated samples? Here are the summary statistics for the two procedures for estimating ACE , along with the simple t-test which we discussed last time.

<i>Method</i>	$ACE = .003$			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	.052	.016	.054	8.7
ANCOVA (main effects)	-.016	.014	.021	76.7
ANCOVA (interactions)	-.004	.015	.016	94.2

Without interactions, the ANCOVA estimate is badly biased. With interactions, it performs well.

And here are the performance statistics for estimating it ACE_1 .

<i>Method</i>	$ACE_1 = -.022$			
	Bias	SD	RMSE	Cvg.
ANCOVA (main effects)	.009	.014	.016	89.0
ANCOVA (interactions)	.006	.014	.015	91.6

Without interactions, the method isn't terrible, but with interactions it is still better.

This illustrates the importance of including baseline-by-treatment interactions when performing causal inference for observational studies. Even if the baseline-by-treatment interactions are not significant (as in this example), it's still better to include them.

Some final comments about ANCOVA for observational studies:

- In this example, we did not carefully check our model for nonlinear relationships or heteroscedasticity.
- Heteroscedasticity won't introduce bias, but it makes OLS inefficient and may distort SE's.
- SE's can be corrected using "robust" or "empirical" method (sandwich formula) which we will learn about later.
- The most dangerous aspect of ANCOVA is that analysts are often unaware how different the treated and untreated groups are, and don't realize when they are extrapolating.

- When many covariates are available, we need a good strategy for variable reduction.
- Best variable-reduction tool is the propensity score, which we will learn about next time.

Method #3: Regression estimation. As an alternative to ANCOVA, which estimates an ACE by a regression coefficient, we can compute estimates of ACE's using **regression predictions**.

Suppose we

- Regress y_{i1} on x_i among **treated** persons, and
- use model to compute predictions \hat{y}_{i1} 's for **everyone**.

Then the average of the \hat{y}_{i1} 's is an estimate of $E(y_{i1})$.

Similarly, we can

- Regress y_{i0} on x_i among **untreated** persons, and
- use the model to compute \hat{y}_{i0} 's for **everyone**.

Then the average of the \hat{y}_{i0} 's is an estimate of $E(y_{i0})$.

The difference between these averages is an estimate of **ACE**.

This approach is conceptually very different from ANCOVA.

- A regression coefficient is a slope in a regression model for the conditional mean of a response given covariates
- A regression estimate is an average of regression predictions (covariate-assisted estimate of a population mean)

The simplest way to perform regression estimation is to use linear models and OLS. We would split the sample and regress y_i on x_i separately for treated ($t_i = 1$) and untreated ($t_i = 0$) persons.

$$\begin{aligned}\hat{\beta}_1 &= \left(\sum_i t_i x_i x_i^T \right)^{-1} \left(\sum_i t_i x_i y_i \right) \\ \hat{\beta}_0 &= \left(\sum_i (1 - t_i) x_i x_i^T \right)^{-1} \left(\sum_i (1 - t_i) x_i y_i \right).\end{aligned}$$

The regression estimate of ACE is

$$A\hat{CE} = \frac{1}{N} \sum_i (\hat{y}_{i1} - \hat{y}_{i0}),$$

where $\hat{y}_{i1} = x_i^T \hat{\beta}_1$ is a prediction based on the first model, and $\hat{y}_{i0} = x_i^T \hat{\beta}_0$ is a prediction based the second model.

Actually, we don't have to substitute regression predictions for the potential outcomes that are known. If we use

predictions only for the missing values, we get

$$A\hat{CE} = \frac{1}{N} \sum_i \{ t_i (y_i - \hat{y}_{i0}) + (1 - t_i) (\hat{y}_{i1} - y_i) \},$$

which is an example of conditional mean imputation (Little & Rubin, 2002; Schafer & Schenker, 2000). With linear models fit by OLS, this equation and the previous one are equivalent (a consequence of the fact that OLS residuals sum to zero).

If our goal is to estimate an ACE among the treated, then the regression model for y_{i1} becomes unnecessary. A regression estimate for ACE_1 is

$$A\hat{CE}_1 = \frac{\sum_i t_i (y_i - \hat{y}_{i0})}{\sum_i t_i},$$

where $\hat{y}_{i0} = x_i^T \hat{\beta}_0$ is a prediction based on a model fit to the untreated.

How does regression estimation compare with ANCOVA?

If

- the same baseline measures appear in the regression models for y_{i1} and y_{i0} , and
- both models are linear,

then regression estimation is equivalent to ANCOVA with all baseline-by-treatment interactions.

But I prefer regression estimation because:

- It forces the analyst to acknowledge that y_{i1} and y_{i0}

are two different variables.

- By splitting the sample, all baseline-by-treatment interactions are included automatically.
- SE formulas for regression estimates (see Appendix) are robust to departures from homoscedasticity.

How does regression estimation perform in the simulated example?

<i>Method</i>	$ACE = .003$			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	.052	.016	.054	8.7
ANCOVA (main effects)	−.016	.014	.021	76.7
ANCOVA (interactions)	−.004	.015	.016	94.2
Regression estimation	−.004	.015	.016	94.3

<i>Method</i>	$ACE_1 = -.022$			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	—	—	—	—
ANCOVA (main effects)	.009	.014	.016	89.0
ANCOVA (interactions)	.006	.014	.015	91.6
Regression estimation	.006	.014	.015	93.2

- Estimates are the same as ANCOVA with interactions, but SE's are a little different.
- Coverage of confidence intervals is a little better with regression estimation.

INTRODUCTION TO CAUSAL INFERENCE (PART IV)

Introduction to propensity scores. For the last two lectures, we discussed methods for estimating average causal effects (ACE's) based on modeling the potential outcomes y_{i0} and y_{i1} . Traditional ANCOVA models, which regress the observed outcome $y_i = t_i y_{i1} + (1 - t_i) y_{i0}$ on t_i and x_i , make implicit assumptions about how the potential outcomes are related to x_i . But there is another body of methods for estimating ACE's that make few or no assumptions about the distributions of the potential outcomes. Rather, they make assumptions about the distribution of the treatment indicator t_i . These methods are based on propensity scores.

Rosenbaum and Rubin (1983) defined the propensity score as

$$\pi_i = P(t_i = 1 \mid x_i, y_{i0}, y_{i1})$$

When the treatment mechanism is unconfounded, the propensities depend only on x_i ,

$$\pi_i = P(t_i = 1 \mid x_i),$$

and we can model them by regressing t_i on x_i .

The most popular way to estimate propensity scores is by fitting a logistic regression,

$$\log \left(\frac{\pi_i}{1 - \pi_i} \right) = x_i^T \gamma.$$

(We have used γ rather than β to distinguish these coefficients from those of the ANCOVA models we considered last time.) Plugging in ML estimate $\hat{\gamma}$ and solving for π_i gives

$$\hat{\pi}_i = \frac{\exp(z_i^T \hat{\gamma})}{1 + \exp(z_i^T \hat{\gamma})}$$

These are the “fitted values” from a standard logistic regression program.

Logistic regression models are popular because the coefficients are easy to interpret (log odds ratios). In the context of causal inference, however, the goal is not interpretation but prediction. The estimated coefficients in $\hat{\gamma}$ may be interesting to look at, but ultimately they are only a device for obtaining $\hat{\pi}_i$'s.

There are many different ways to use propensity scores to estimate ACE's in observational studies. But even if you don't use them to estimate an ACE, you should model them anyway, because they are an important diagnostic.

Let's read in our sample of $n = 6,000$ subjects and fit a logistic model to predict the dieting variable t_i from the thirteen covariates in x_i .

```

> samp <- read.table("Diet0001.dat", row.names="ID",
+   col.names=c("ID", "DISTR.1", "BLACK", "NBHISP", "GRADE", "SLFHLTH",
+   "SLFWGHT", "WORKHARD", "GOODQUAL", "PHYSFIT", "PROUD",
+   "LIKESLF", "ACCEPTED", "FEELLOVD", "DIET", "Y.0", "Y.1", "DISTR.2",
+   "PI.TRUE"))
>
>
> result <- glm( DIET ~ DISTR.1 + BLACK + NBHISP + GRADE +
+   SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD +
+   LIKESLF + ACCEPTED + FEELLOVD, family=binomial,
+   data=samp)
>
> summary(result)

```

Call:

```
glm(formula = DIET ~ DISTR.1 + BLACK + NBHISP + GRADE + SLFHLTH +
  SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT + PROUD + LIKESLF +
  ACCEPTED + FEELLOVD, family = binomial, data = samp)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.8810	-0.6611	-0.4683	-0.2443	3.0816

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)							
(Intercept)	-6.18454	0.33267	-18.591	< 2e-16 ***							
DISTR.1	0.32328	0.08609	3.755	0.000173 ***							
BLACK	-0.52714	0.09271	-5.686	1.30e-08 ***							
NBHISP	-0.17469	0.09938	-1.758	0.078785 .							
GRADE	0.09175	0.02599	3.530	0.000416 ***							
SLFHLTH	-0.04820	0.04235	-1.138	0.255035							
SLFWGHT	1.16795	0.05446	21.448	< 2e-16 ***							
WORKHARD	-0.13147	0.04133	-3.181	0.001468 **							
GOODQUAL	-0.19374	0.07105	-2.727	0.006395 **							
PHYSFIT	0.04328	0.04813	0.899	0.368585							
PROUD	-0.03753	0.07211	-0.520	0.602797							
LIKESLF	0.28300	0.04366	6.482	9.05e-11 ***							
ACCEPTED	-0.11514	0.04788	-2.405	0.016177 *							
FEELLOVD	0.04735	0.05961	0.794	0.427022							

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 6059.9 on 5999 degrees of freedom
Residual deviance: 5201.7 on 5986 degrees of freedom
AIC: 5229.7
```

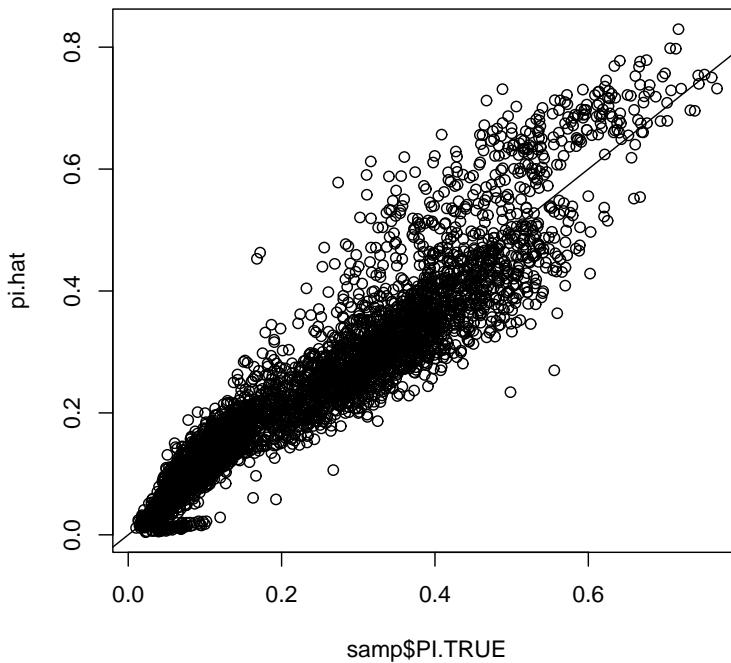
```
Number of Fisher Scoring iterations: 5
```

Many of the covariates are significant predictors of dieting. The most powerful predictor (in terms of the z-statistic) is **SLFWGHT**, the individual's self-perceived weight relative to her peers.

If you fit a propensity model and find that some covariates are not significant predictors of t_i , should you remove them? Not necessarily. The purpose of this model is not parsimonious description, but prediction of the π_i 's. Rich, overfitted propensity models can be a good thing. With propensity modeling, it's more dangerous to omit important covariates than to include unnecessary ones. So we don't have to trim away covariates whose coefficients are insignificant.

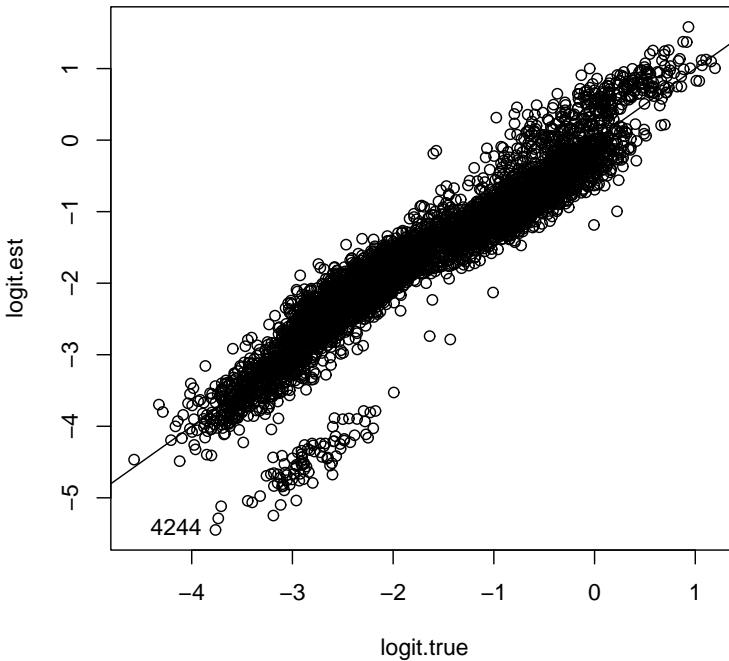
We do, however, need to be concerned about lack of fit. In this population, the logistic propensity model is incorrect. Because the data were simulated, we have access to the true propensity scores, which are included in the dataset as the variable **PI.TRUE**. Let's plot the estimated propensities against the true propensities.

```
> pi.hat <- result$fitted.values
> plot(samp$PI.TRUE, pi.hat)
> abline(0,1) # add the 45-degree line y=x to the plot
```



Because propensities are restricted to lie in the unit interval, this plot has poor resolution near 0. We can get better resolution by re-plotting on the logistic scale.

```
> logit.true <- log( samp$PI.TRUE / (1-samp$PI.TRUE) )
> logit.est  <- log( pi.hat / (1-pi.hat) )
> plot( logit.true, logit.est )
> abline(0,1) # add the 45-degree line y=x to the plot
```



The fit looks pretty good overall, except for the unusual cluster of points for which the estimated values lie below the true values. I have identified one point in this cluster (observation 4244). This observation has a true log-odds of

```
> logit.true[4255]  
[1] -3.079639
```

which corresponds to a probability of

```
> exp(-3.079639) / ( 1 + exp(-3.079639) )  
[1] 0.04395498
```

The estimated log-odds is

```
> logit.est[4255]
-4.527468
```

which corresponds to a probability of

```
> exp(-4.527468) / ( 1 + exp(-4.527468) )
[1] 0.01069244
```

The difference between a propensity of 0.044 and 0.011 may appear to be small. But discrepancies like these can have an enormous impact on ACE's estimated by certain kinds of propensity-score methods.

Propensity scores and balance. The key property of propensity scores is that they balance the distributions of the covariates (Rosenbaum & Rubin, 1983) in the following sense: Treated and untreated persons with identical propensity scores have identical distributions for **all the covariates**. That is,

$$P(x_i \mid \pi_i = c, t_i = 1) = P(x_i \mid \pi_i = c, t_i = 0)$$

for any c between 0 and 1. If we divide the population into groups of constant propensity, then subjects in each group can be treated as if they had participated in a randomized experiment.

This balancing property is a feature of the **true** propensity scores, but if the propensity model is reasonably good, it will approximately hold for the estimated propensities as well.

To illustrate, let's look at our sample and compare the mean of each covariate for dieters and nondieters. In the table below, \bar{x}_0 is the mean among nondieters, \bar{x}_1 is the mean among dieters, d is the standardized difference

$$d = \frac{\bar{x}_1 - \bar{x}_0}{S_p},$$

(S_p is the pooled within-group standard deviation) and T is the t-statistic for the pooled two-sample t-test.

	(a) Full sample			
	\bar{x}_0	\bar{x}_1	d	T
DISTR.1	0.62	0.71	0.22	6.82
BLACK	0.26	0.17	-0.19	-5.90
NBHISP	0.15	0.15	0.02	0.65
GRADE	9.16	9.37	0.15	4.69
SLFHDLTH	2.20	2.35	0.17	5.29
SLFWGHT	3.19	3.84	0.88	27.29
WORKHARD	2.14	2.05	-0.09	-2.85
GOODQUAL	1.80	1.84	0.06	1.75
PHYSFIT	2.24	2.53	0.32	9.99
PROUD	1.76	1.86	0.13	3.96
LIKESLF	2.09	2.52	0.43	13.38
ACCEPTED	2.14	2.35	0.21	6.56
FEELLOVD	1.78	1.93	0.18	5.48

If this were a randomized experiment, then each group would be a random sample from the same population, and most or all of the t-statistics would be non-significant. But

this is obviously not the case; significant differences between the groups exist on virtually all of these covariates.

Now, instead of looking at the full sample, let's restrict our attention only to the girls whose estimated propensities satisfy $.35 \leq \hat{\pi}_i \leq .40$.

	(b) Group with $.35 \leq \hat{\pi}_i \leq .40$			
	\bar{x}_0	\bar{x}_1	d	T
DISTR.1	0.81	0.81	0.01	0.09
BLACK	0.09	0.08	-0.03	-0.23
NBHISP	0.16	0.19	0.07	0.62
GRADE	9.60	9.60	0.01	0.05
SLFHLTH	2.57	2.52	-0.05	-0.41
SLFWGHT	4.01	4.01	0.03	0.24
WORKHARD	1.97	1.96	-0.02	-0.19
GOODQUAL	1.87	1.99	0.18	1.58
PHYSFIT	2.73	2.77	0.05	0.43
PROUD	1.95	2.09	0.16	1.44
LIKESLF	2.80	2.88	0.08	0.70
ACCEPTED	2.43	2.55	0.12	1.08
FEELLOVD	2.05	2.17	0.13	1.18

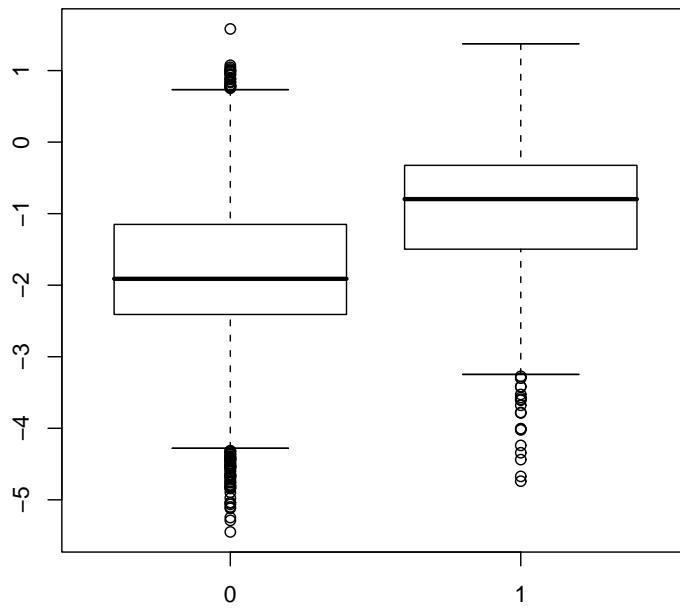
All the differences are now non-significant. This is partly due to the fact that the sample size is now smaller. But the standardized differences, which don't increase with the sample size, are also much smaller.

The propensity score is a covariate. But it is a very

important one, because it captures the essence of how treated and untreated groups differed at the beginning of the study. The propensity score (or any monotonic function of it, such as the logit-propensity score) is the coarsest summary of x_i with this balancing property.

The propensity score (or the logit-propensity score) is the baseline variable that maximally discriminates between the groups. It is the characteristic on which the two groups are most different. For this reason, it's a good idea to compare the distributions of the estimated logit-propensity scores to see how different the groups really are. Let's look at boxplots of the logit-propensities for the two groups.

```
> boxplot( split( logit.est, samp$DIET ) )
```



This plot is probably the single most important diagnostic for causal inference, because it tells us whether causal inference is reliable and when it is not. When examining this plot, we should first consider **the degree of overlap**. To what extent is the distribution for each group covered by the distribution for the other? In this case, the propensities for the dieters ($t_i = 1$) are completely covered by the propensities for the nondieters ($t_i = 0$). This is good news for estimating ACE_1 . For every girl who dieted, we can find a comparable girl who had a similar propensity to diet but did not, who can serve as a proxy to guess what the dieter's value of y_{i0} might be. But the propensities for non-dieters are not covered by those of the

dieters. For the nondieters with very low probabilities, there are few or no comparable dieters, so there is no information to help us guess what those nondieters' missing values of y_{i1} might be. Causal effects in this low-propensity region will be based on extrapolation and will be sensitive to modeling assumptions.

When examining these distributions, we should also consider **the degree of imbalance**. The mean logit-propensity among dieters is

```
> mean.1 <- mean( logit.est[ samp$DIET==1 ] )
> mean.1
[1] -0.892816
```

but the mean among non-dieters is

```
> mean.0 <- mean( logit.est[ samp$DIET==0 ] )
> mean.0
[1] -1.855930
```

In terms of pooled standard deviations, the difference is

```
> n0 <- sum( samp$DIET==0 )
> n1 <- sum( samp$DIET==1 )
> var0 <- var( logit.est[ samp$DIET==0 ] )
> var1 <- var( logit.est[ samp$DIET==1 ] )
> Spooled <- sqrt( ( (n0-1)*var0 + (n1-1)*var1 ) / (n0 + n1 - 2) )
> (mean.1 - mean.0) / Spooled
[1] 0.9639282
```

The means of the two groups differ by almost one standard deviation. This is a big difference. Rubin (2001) claims

that ANCOVA—meaning simple linear ANCOVA without interactions—is unreliable in an observational study if the means of the logit-propensities differ by more than one-half of the pooled SD. If the means differ by more than this, it becomes important to use something other than simple linear ANCOVA. In these situations, we need to consider baseline-by-treatment interactions. And we should strongly consider using the propensity scores for estimating an ACE. There are many different methods for doing this, and we will now examine some of them.

Method #4: Matching. Matching means that we select matched subsamples of treated and untreated persons whose covariate distributions are similar enough that selection bias is not an issue. Matching works best when one group (treated or untreated) is smaller than the other, and the distribution of the propensities in the smaller group is well covered by the distribution in the larger group

- For each person in the smaller group, select one in the larger group that matches it well in terms of propensity values (and perhaps other covariates as well)
- Continue selecting without replacement
- After matches have been found for everyone in the smaller group, throw away the excess in the larger group

An ACE estimated from a matched sample represents the ACE in the population represented by the **smaller group**.

- If the smaller group was $t_i = 1$, then it estimates ACE_1
- If the smaller group was $t_i = 1$, then it estimates ACE_0 (average causal effect among the untreated)

Is it wasteful to throw away the data for the unmatched subjects? Not really. Throwing away the excess cases does not reduce the power very much. The excess cases did not have any comparables in the other group, so they are not so useful for causal inference anyway.

After the matching is completed, the matched samples may be compared by an **unpaired** t-test. Or, better yet, we can perform ANCOVA on the matched sample.

- If the matching was perfect, then the matched samples will resemble a randomized experiment. In that case, ANCOVA is still helpful for increasing precision.
- If the matching was less than perfect, ANCOVA will help to remove the remaining selection bias. (Simple ANCOVA without interactions should be okay, because the differences between the groups should be mild.)

Techniques for propensity-score matching are reviewed by D'Agostino (1998), Rosenbaum (2002) and Rubin and Thomas (1996). One popular procedure is “Mahalanobis-metric matching within calipers defined by the logit-propensity score.”

- Let $\hat{\eta}_i = \log[\hat{\pi}_i/(1 - \hat{\pi}_i)]$ denote the estimated logit-propensity
- For each individual in the smaller group, identify a pool of potential matches in the larger group whose logit-propensities are within $\hat{\eta}_i \pm c$, where c is one-quarter of the within-group SD
- Within this pool, choose the individual who is closest in terms of Mahalanobis distance

This algorithm has been implemented in a SAS macro called GREEDY (Parsons, 2000), a Stata module called PSMATCH2 (Leuven & Sianesi, 2003), and an R package called MatchIt (Ho et al., 2004).

We applied Mahalanobis metric matching within calipers defined by the propensity score. Code for the procedure is on the website.

- In the initial sample, out of 1,220 dieters, 1,194 were successfully matched to non-dieters. The remaining 26 dieters were thrown away. And the remaining $4,780 - 1,194 = 3,586$ non-dieters were thrown away.
- After matching, we performed a simple t-test to

compare the mean of y_i among dieters and nondieters. And we also used an ANCOVA procedure, regressing y_i on t_i and x_i (no interactions).

- Over repeated samples, matching followed by a t-test did not work well. But matching followed by simple ANCOVA worked great.

Method	$ACE_1 = -.022$			
	Bias	SD	RMSE	Cvg.
ANCOVA (main effects)	.009	.014	.016	89.0
ANCOVA (interactions)	.006	.014	.015	91.6
Regression estimation	.006	.014	.015	93.2
Matching + t-test	.035	.017	.039	58.6
Matching + ANCOVA	.004	.016	.017	94.3

Matching + ANCOVA is one of the best methods for estimating ACE_1 . But it cannot be used to estimate ACE .

INTRODUCTION TO CAUSAL INFERENCE (PART V)

Last time, we introduced the propensity score,

$$\pi_i = P(t_i = 1 \mid x_i),$$

which is typically estimated by the fitted values from a logistic regression of t_i on x_i . The key property of propensity scores is that they balance the distributions of the covariates,

$$P(x_i \mid \pi_i = c, t_i = 1) = P(x_i \mid \pi_i = c, t_i = 0)$$

for any c between 0 and 1. If we could divide the population into groups of constant propensity, then subjects in each group can be treated as if they had participated in a randomized experiment.

The logit-propensity score is the covariate that maximally discriminates between the groups. By comparing the distributions of the estimated logit-propensity scores for $t_i = 0$ and $t_i = 1$, we see how different the groups really are. When the means of the logit-propensity scores differ by more than one-half of a pooled SD, then simple linear ANCOVA without interactions is not a reliable method of correcting for selection bias.

The propensity scores are used in many different ways to construct estimates of ACE's. We have already discussed matching, in which subjects from the larger group are selected to have propensity scores similar to those of the smaller group. Today we will examine several other procedures based on the estimated propensities.

Method #5: Inverse-Propensity Weighting (IPW).

Weights are often used to adjust for unequal probabilities of selection in sample surveys (Horvitz & Thompson, 1952; Lohr, 1999). It's based on the principle that

- individuals from oversampled groups should be given less weight, and
- individuals from undersampled groups should be given more weight.

For example, imagine a national survey in which persons of Hispanic origin are sampled at a rate of 1:20,000, and others are sampled at a rate of 1:50,000.

Let $w_i = 20,000$ if person i is Hispanic, and $w_i = 50,000$ otherwise. Then the weighted average

$$\frac{\sum_i w_i y_i}{\sum_i w_i}$$

is an unbiased estimate of $E(y_i)$ in the population. In a survey, the weight w_i can be interpreted as

- the number of population persons represented by

sample person i

- the inverse (or reciprocal) probability with which person i was selected into the sample

This principle can also be applied to estimation of ACE's.

- A weighted average of the observed y_{i1} 's, using weights $1/\hat{\pi}_i$, is an approximately unbiased estimate of $E(y_{i1})$
- A weighted average of the observed y_{i0} 's, using weights $1/(1 - \hat{\pi}_i)$, is an approximately unbiased estimate of $E(y_{i0})$
- The difference between them is an estimate of ACE

The IPW estimate can be written as

$$\hat{ACE} = \frac{\sum_i t_i \hat{\pi}_i^{-1} y_{i1}}{\sum_i t_i \hat{\pi}_i^{-1}} - \frac{\sum_i (1 - t_i)(1 - \hat{\pi}_i)^{-1} y_{i0}}{\sum_i (1 - t_i)(1 - \hat{\pi}_i)^{-1}}$$

where sums are taken over the whole sample

An IPW estimate for the treated is

$$\hat{ACE}_1 = \frac{\sum_i t_i y_i}{\sum_i t_i} - \frac{\sum_i (1 - t_i) \hat{\pi}_i (1 - \hat{\pi}_i)^{-1} y_i}{\sum_i (1 - t_i) \hat{\pi}_i (1 - \hat{\pi}_i)^{-1}}.$$

(Hirano & Imbens, 2002)

SE's for IPW estimates are complicated (see Appendix), because they must account for the fact that the propensities are estimated. Code for computing estimates and standard errors is given on the website.

Here's how the IPW performs over repeated samples in our simulation.

<i>Method</i>	<i>ACE = .003</i>			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	.052	.016	.054	8.7
ANCOVA (main effects)	−.016	.014	.021	76.7
ANCOVA (interactions)	−.004	.015	.016	94.2
Regression estimation	−.004	.015	.016	94.3
Matching + t-test	—	—	—	—
Matching + ANCOVA	—	—	—	—
Inverse-propensity weighting	−.014	.022	.027	88.8

<i>Method</i>	<i>ACE₁ = −.022</i>			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	—	—	—	—
ANCOVA (main effects)	.009	.014	.016	89.0
ANCOVA (interactions)	.006	.014	.015	91.6
Regression estimation	.006	.014	.015	93.2
Matching + t-test	.035	.017	.039	58.6
Matching + ANCOVA	.004	.016	.017	94.3
Inverse-propensity weighting	.002	.016	.016	94.8

In this example, IPW is biased for ACE, and it's not efficient for ACE or ACE₁. The reason why it's biased is because the propensity model is not correct. A better propensity model might fix up the bias. But the inefficiency comes from the fact that IPW estimates are greatly influenced by outliers. Outliers are

- persons with $t_i = 1$ but $\hat{\pi}_i \approx 0$, and
- persons with $t_i = 0$ but $\hat{\pi}_i \approx 1$.

Outliers provide useful information for inferring the missing potential outcomes, but IPW relies on them too much, giving them too much weight in the estimation.

Method #6: Subclassification. This method, first proposed by Rosenbaum and Rubin (1984).

- Divide the sample into S classes (strata) in which the propensity is nearly constant.
- Estimate the treatment effect each class (e.g. by difference in means or ANCOVA); like the results of a randomized experiment.
- \hat{ACE} is a weighted average of these class-specific estimates, weighted by the proportion of the population in each class.
- Similar to IPW, except that the weights have been coarsened into a few categories.

Let

$$\begin{aligned}\hat{\theta}_s &= \text{estimate of the ACE in class } s \\ \hat{V}_s &= \text{estimated variance of } \hat{\theta}_s\end{aligned}$$

The estimate of *ACE* is

$$A\hat{CE} = \sum_s \left(\frac{N_s}{N} \right) \hat{\theta}_s,$$

and its estimated variance is

$$\sum_s \left(\frac{N_s}{N} \right)^2 \hat{V}_s,$$

where N_s is the number of subjects in classs, and
 $N = \sum_s N_s$ is the total number of subjects.

The estimate of ACE_1 is

$$A\hat{CE}_1 = \sum_s \left(\frac{N_s^*}{N^*} \right) \hat{\theta}_s,$$

and its estimated variance is

$$\sum_s \left(\frac{N_s^*}{N^*} \right)^2 \hat{V}_s,$$

where N_s^* is the number of **treated** subjects in class s ,
and $N^* = \sum_s N_s^*$ is the total number of **treated** subjects

How do we create the classes?

- It's customary to create $S = 5$ classes, defined by sample quintiles of the estimated propensity scores.
- But we may want to use more classes if the data allow it.

Results from dividing our sample of $n = 6,000$ from the dieting population:

	range($\hat{\pi}_i$)		N		
	min	max	$t_i = 0$	$t_i = 1$	total
Class 1	.004	.080	1,139	61	1,200
Class 2	.080	.122	1,097	103	1,200
Class 3	.122	.196	1,044	156	1,200
Class 4	.196	.327	856	344	1,200
Class 5	.327	.830	644	556	1,200

- Classes 4 and 5 cover wide ranges of propensities
- Because there are plenty of treated and untreated persons in Class 4 and Class 5, we can afford to split them apart

Let's split Class 4 into two, and Class 5 into four

	range($\hat{\pi}_i$)		N		
	min	max	$t_i = 0$	$t_i = 1$	total
Class 1	.004	.080	1,139	61	1,200
Class 2	.080	.122	1,097	103	1,200
Class 3	.122	.196	1,044	156	1,200
Class 4a	.196	.265	446	154	600
Class 4b	.266	.327	410	190	600
Class 5a	.327	.363	184	116	300
Class 5b	.363	.417	166	134	300
Class 5c	.417	.512	158	142	300
Class 5d	.513	.830	136	164	300

Within each class, let's compute $\hat{\theta}_s$ and $\hat{V}(\hat{\theta})$ in two different ways:

- by an ordinary pooled two-sample t-test, and
- by ANCOVA (main effects only)

Performance over repeated samples:

<i>Method</i>	$ACE = .003$			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	.052	.016	.054	8.7
ANCOVA (main effects)	−.016	.014	.021	76.7
ANCOVA (interactions)	−.004	.015	.016	94.2
Regression estimation	−.004	.015	.016	94.3
Matching + t-test	—	—	—	—
Matching + ANCOVA	—	—	—	—
Inverse-propensity weighting	−.014	.022	.027	88.8
Subclassification + t-test	.005	.018	.018	96.1
Subclassification + ANCOVA	.001	.016	.016	94.4

<i>Method</i>	$ACE_1 = -.022$			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	—	—	—	—
ANCOVA (main effects)	.009	.014	.016	89.0
ANCOVA (interactions)	.006	.014	.015	91.6
Regression estimation	.006	.014	.015	93.2
Matching + t-test	.035	.017	.039	58.6
Matching + ANCOVA	.004	.016	.017	94.3
Inverse-propensity weighting	.002	.016	.016	94.8
Subclassification + t-test	.008	.015	.017	95.8
Subclassification + ANCOVA	.006	.015	.016	93.6

- Subclassification + t-test works well
- Subclassification + ANCOVA works even better

Methods based on dual modeling. Thus far, we have examined two types of estimates for causal effects:

- methods based on modeling the outcomes (e.g., ANCOVA), and
- methods based on estimated propensity scores.

In recent years, some new methods have appeared that combine models for the outcomes with estimates of the propensity scores. These methods have a property known as double robustness (DR), which means that the resulting estimate remains consistent (i.e. converges to the true value as $n \rightarrow \infty$) if either model is true. There are many ways to construct DR estimates. We will mention a few.

Method #7: Weighted residual bias corrections.

Consider the regression estimate

$$\hat{ACE} = \frac{1}{N} \sum_i (\hat{y}_{i1} - \hat{y}_{i0})$$

- If the model for the treated persons does not accurately describe the relationship between y_{i1} and x_i , then the average of the \hat{y}_{i1} 's may be a biased estimate of $E(y_{i1})$
- If the model for the untreated persons does not accurately describe the relationship between y_{i0} and x_i , then the average of the \hat{y}_{i0} 's may be a biased estimate of $E(y_{i0})$

- It's difficult to guess the direction or magnitude of these biases using standard regression diagnostics
- However, we can construct numerical estimates of these biases through a clever combination of **residuals** and **propensity scores**

The residuals from the y_{i1} model,

$$\hat{\epsilon}_{i1} = y_{i1} - \hat{y}_{i1},$$

have a mean of zero in the treated sample. But their average in the whole sample may not be zero if the regression model is wrong.

We cannot see these residuals for the whole sample. But we can estimate their average in the whole sample by a weighted average of the observed $\hat{\epsilon}_{i1}$'s, weighted by $1/\hat{\pi}_i$. Adding this weighted average to $\sum_i \hat{y}_{i1}/N$ gives a bias-corrected estimate of $E(y_{i1})$. Similarly, a weighted average of the observed $\hat{\epsilon}_{i0}$'s, weighted by $1/(1 - \hat{\pi}_i)$, will correct the bias in $\sum_i \hat{y}_{i0}/N$ as an estimate of $E(y_{i0})$.

The bias-corrected estimates are

$$\begin{aligned} A\hat{CE} &= \frac{1}{N} \sum_i (\hat{y}_{i1} - \hat{y}_{i0}) + \frac{\sum_i t_i \hat{\pi}_i^{-1} \hat{\epsilon}_{i1}}{\sum_i t_i \hat{\pi}_i^{-1}} \\ &\quad - \frac{\sum_i (1 - t_i) (1 - \hat{\pi}_i)^{-1} \hat{\epsilon}_{i0}}{\sum_i (1 - t_i) (1 - \hat{\pi}_i)^{-1}} \\ A\hat{CE}_1 &= \frac{\sum_i t_i (y_i - \hat{y}_{i0})}{\sum_i t_i} - \frac{\sum_i (1 - t_i) \hat{\pi}_i (1 - \hat{\pi}_i)^{-1} \hat{\epsilon}_{i0}}{\sum_i (1 - t_i) \hat{\pi}_i (1 - \hat{\pi}_i)^{-1}} \end{aligned}$$

- The bias-corrected estimates of ACE's are consistent if the regression models for y_{i1} and y_{i0} are wrong, provided that the propensity model is correct.
- The estimates are also consistent if the propensity model is wrong, but the regression models for y_{i1} and y_{i0} are correct.

Method #8: Weighted Regression Estimation.

Another way to make a regression estimate DR is to apply inverse-propensity weights during the regression modeling (WLS rather than OLS).

- In ordinary regression modeling, WLS is used to correct for heteroscedasticity, to make estimates more efficient
- Here the purpose is to give consistent estimates of the coefficients that you would get if you fit the regression model to the whole population

As before, assume linear regression models

$E(y_{i1} | x_i) = x_i^T \beta_1$ and $E(y_{i0} | x_i) = x_i^T \beta_0$, but now estimate the coefficients by

$$\begin{aligned}\hat{\beta}_1 &= \left(\sum_i t_i \hat{\pi}_i^{-1} x_i x_i^T \right)^{-1} \left(\sum_i t_i \hat{\pi}_i^{-1} x_i y_i \right) \\ \hat{\beta}_0 &= \left(\sum_i (1 - t_i)(1 - \hat{\pi})^{-1} x_i x_i^T \right)^{-1}\end{aligned}$$

$$\left(\sum_i (1 - t_i)(1 - \hat{\pi}_i)^{-1} x_i y_i \right).$$

The weighted regression estimates are then

$$\begin{aligned} A\hat{CE} &= \frac{1}{N} \sum_i (\hat{y}_{i1} - \hat{y}_{i0}) \\ A\hat{CE}_1 &= \frac{\sum_i t_i (\hat{y}_{i1} - \hat{y}_{i0})}{\sum_i t_i} \end{aligned}$$

where $\hat{y}_{i1} = x_i^T \hat{\beta}_1$ and $\hat{y}_{i0} = x_i^T \hat{\beta}_0$ are predicted values from the regression models

Method #9: Propensity-Related Covariates. Here's a good idea: If the regression estimates are biased because the regression models are wrong, then why not fix up the models?

One simple and effective way to repair the model is to include summaries of the estimated propensity scores as additional covariates in a regression estimation procedure.

Our recommended strategy:

- Divide the subjects into $S \geq 5$ strata based on their $\hat{\pi}_i$'s (in our sample, we use nine)
- Create $S - 1$ dummy variables to distinguish among the strata
- Add these dummies to the regression models for y_{i1}

and y_{i0} , and then compute ordinary regression estimates

Now let's compare the performance of all these methods.

<i>Method</i>	<i>ACE = .003</i>			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	.052	.016	.054	8.7
ANCOVA (main effects)	−.016	.014	.021	76.7
ANCOVA (interactions)	−.004	.015	.016	94.2
Regression estimation	−.004	.015	.016	94.3
Matching + t-test	—	—	—	—
Matching + ANCOVA	—	—	—	—
Inverse-propensity weighting	−.014	.022	.027	88.8
Subclassification + t-test	.005	.018	.018	96.1
Subclassification + ANCOVA	.001	.016	.016	94.4
Residual bias correction	−.013	.018	.023	86.2
Weighted regression estimation	−.009	.016	.019	88.4
Propensity-related covariates	.000	.016	.016	94.6

<i>Method</i>	$ACE_1 = -.022$			
	Bias	SD	RMSE	Cvg.
Difference in means (t-test)	—	—	—	—
ANCOVA (main effects)	.009	.014	.016	89.0
ANCOVA (interactions)	.006	.014	.015	91.6
Regression estimation	.006	.014	.015	93.2
Matching + t-test	.035	.017	.039	58.6
Matching + ANCOVA	.004	.016	.017	94.3
Inverse-propensity weighting	.002	.016	.016	94.8
Subclassification + t-test	.008	.015	.017	95.8
Subclassification + ANCOVA	.006	.015	.016	93.6
Residual bias correction	.005	.015	.016	93.7
Weighted regression estimation	.005	.015	.016	93.7
Propensity-related covariates	.005	.015	.016	93.5

Propensity-related covariates works well both for ACE and ACE_1 . In my opinion, it may be the best of the dual-modeling methods.

Final remarks about causal inference.

- Purists say that causal inference is impossible without a randomized experiment. But in many situations, observational data are all we have.
- Reliable causal inferences from observational data are possible if the confounders have been well measured,

and if the groups are not far apart with respect to the confounders.

Most regression analyses fall far short of causal inferences, because in regression analysis we often fail to make any distinction between the treatment variable and the confounders.

Before interpreting a regression coefficient as a causal effect, make sure that you consider

- the mechanism by which the treatment was assigned
- what confounders may exist, and whether they have been well measured
- how much the treatment groups differ with respect to the confounders
- the possibility of interactions between the treatment and the confounders