V. Advanced Regression Topics

a. MR Standard Errors
b. Multi-colinearity
c. Standardized Residuals, Leverage, and Outliers
d. Nonlinearity
e. Dummy Variables
f. Heteroskedasticity
g. Bootstrapping Regression Models
h. LASSO example

Appendix. Bootstrap Confidence Intervals Explained

a. MR Standard Errors

In Chapter II, we developed an intuition regarding the computation of standard errors in multiple regression.

We decided that the SLR formula could be modified by inserting a measure of the *independent variation* of the X variable.

Now let's look at this formula for the general case:

$$Var(b_j) = \frac{\sigma^2}{(N-1)s^2_{j,i.v}}$$

Estimate of the fraction of the variance of X_j independent of other X variables

a. MR Standard Errors

We estimate this quantity to form the *standard error* of b_j as follows:

$$se(b_j) = \frac{s}{\sqrt{SSE_{j|others}}}$$

where: $SSE_{j|others}$

is from:

$$X_j = a_0 + a_1 X_1 + \ldots + a_{j-1} X_{j-1} + a_{j+1} X_{j+1} + \ldots + a_k X_k + e_{j|others}$$

a. MR Standard Errors Country Returns Example

Recall in chapter II we looked at the relationship between returns in US and other countries.

```
lm(formula = usa ~ canada + uk + australia + france + germany +
    japan)

Residuals:
      Min        1Q    Median        3Q       Max
-0.056345 -0.017073  0.000807  0.011979  0.074612

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.006136   0.002309   2.657 0.009171 **
canada       0.444362   0.069587   6.386 5.41e-09 ***
uk           0.225690   0.064915   3.477 0.000753 ***
australia   -0.056688   0.050366  -1.126 0.263061
france       0.166742   0.061338   2.718 0.007733 **
germany     -0.064793   0.057239  -1.132 0.260353
japan       -0.051028   0.034615  -1.474 0.143580
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02257 on 100 degrees of freedom
Multiple R-squared: 0.566,  Adjusted R-squared:  0.54
F-statistic: 21.74 on 6 and 100 DF,  p-value: 3.267e-16
```

Let's compute the standard error on the canada coef.

data(countryret)

a. MR Standard Errors

Back to Country Returns data:

We will need to regress canada returns on all other countries in the regression on the previous page.

```
> out = lm(canada~uk + australia + germany + france + japan, data=countryret)
> anova(out)
Analysis of Variance Table

Response: canada
           Df   Sum Sq  Mean Sq F value    Pr(>F)
uk          1 0.032471 0.032471 31.1654 2.002e-07 ***
australia   1 0.016127 0.016127 15.4783 0.0001533 ***
germany     1 0.001032 0.001032  0.9902 0.3220624
france      1 0.000626 0.000626  0.6010 0.4400195
japan       1 0.000219 0.000219  0.2099 0.6478568
Residuals 101 0.105232 0.001042
---
```

$$se\left(b_{canada}\right) = \frac{s}{\sqrt{SSE_{canada\,|\,others}}} = \frac{0.02257}{\sqrt{0.1052}} = 0.0695$$

b. Multi-colinearity

In certain situations, the dependence among the X variables can be so strong that it may be difficult to estimate the regression coefficients.

In this situation, we say that the dataset exhibits **multi-colinearity.**

Perfect colinearity refers to a situation in which one or more independent variables can be expressed as a linear combination of other variables – pure redundancy.

Multi-colinearity is about a lack of information.

b. Multi-colinearity

What is true about multi-colinearity:

i. colinear variables can have coefficients with large standard errors

ii. colinear variables can have insignificant t's but very significant F's

iii. getting a larger sample doesn't necessarily help much

What is <u>not</u> true about multi-colinearity :

i. multicolinearity is not a "disease"

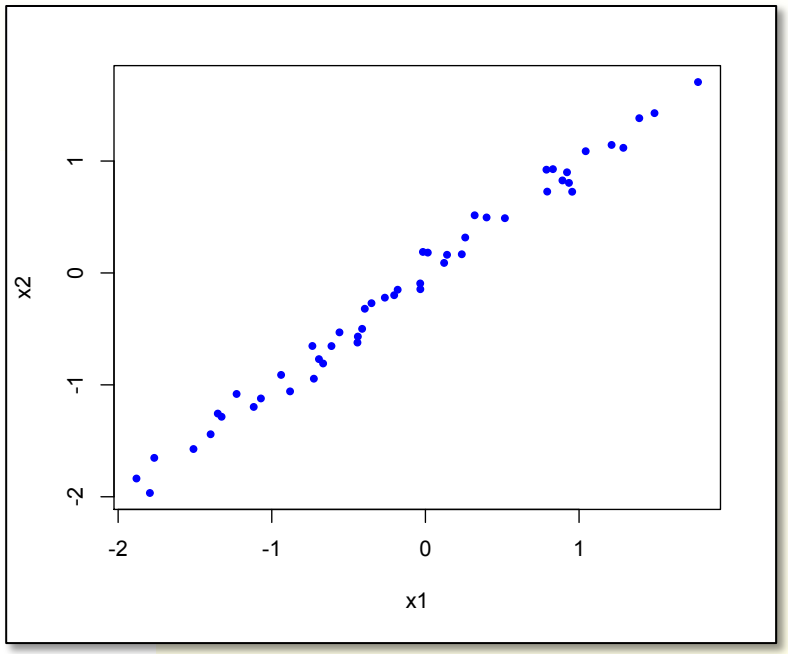
ii. it is not a violation of the model assumptions

iii. Least squares and the least squares standard errors are still OK.

b. Multi-colinearity

Example with simulated data:

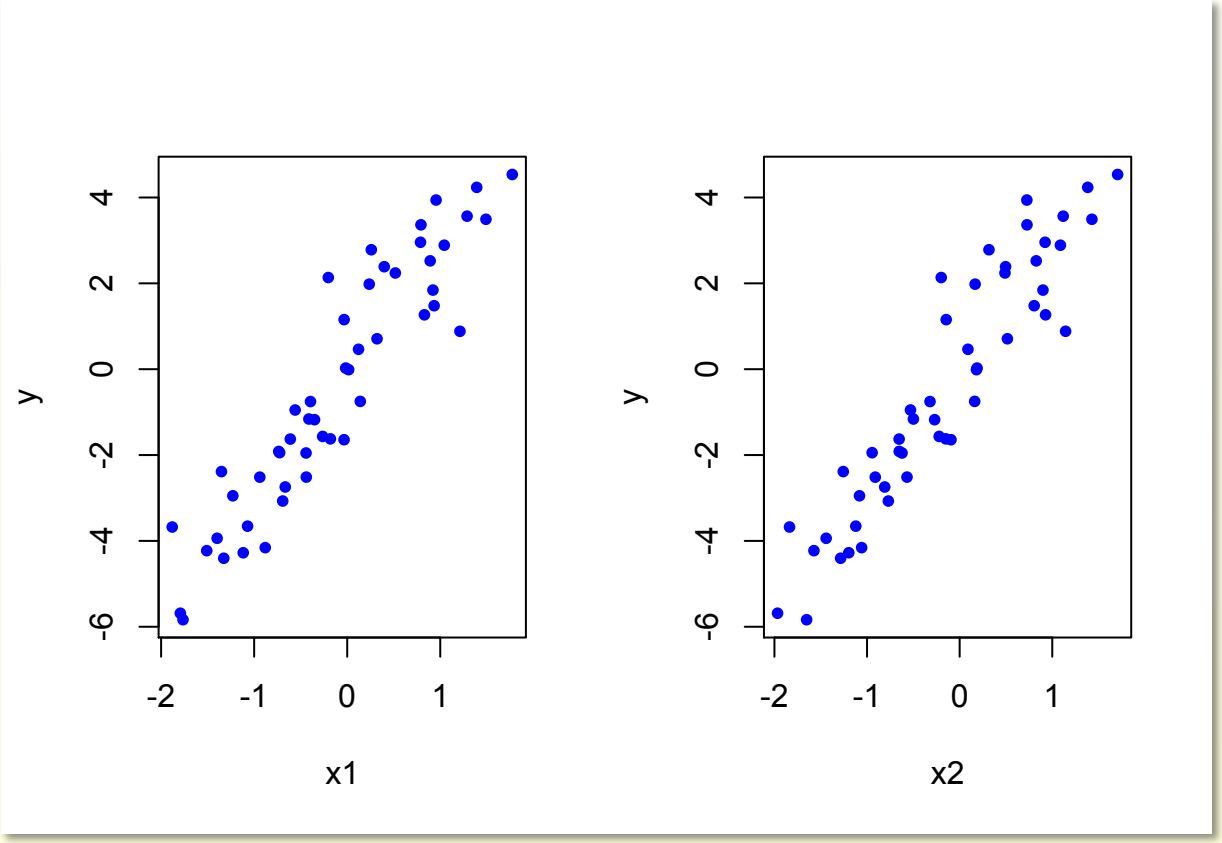
```
> x1=rnorm(50)
> x2=x1+rnorm(50,sd=.1)
> y=1*x1+2*x2+rnorm(50)
> summary(lm(y~x1+x2))

Call:
lm(formula = y ~ x1 + x2)

Residuals:
     Min       1Q   Median       3Q      Max
-2.35068 -0.71101 -0.04493  0.46412  2.73168

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.03393    0.14854  -0.228    0.820
x1           0.73528    1.37255   0.536    0.595
x2           2.07750    1.37438   1.512    0.137

Residual standard error: 1.026 on 47 degrees of freedom
Multiple R-squared: 0.8741, Adjusted R-squared: 0.8687
F-statistic: 163.1 on 2 and 47 DF,  p-value: < 2.2e-16
```



Overall F is very significant, but neither t's are large!!

b. Multi-colinearity

Y is clearly related to each X when considered alone.



b. Multi-colinearity

How Should Multi-colinearity Be Detected?:

- Inter-correlations of X's
- Regress X_j on other X's and get very high R^2

What Can Be Done?

- delete some X's (give up on those partial effects)
- combine the highly correlated indep variables into indices

b. Multi-colinearity

What *Should Not* Be Done:

- delete X variables *just* because they are highly inter-correlated!

dataset(multicolinear) : corr(X1,X2) > .95, yet we can estimate multiple regression!

```
> cor(x1,x2)
[1] 0.9922075
```

```
> summary(lm(y~x1+x2))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.11307    0.09119    1.24    0.218
x1          -1.06905    0.06382  -16.75   <2e-16 ***
x2           1.08695    0.06345   17.13   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.874 on 97 degrees of freedom
Multiple R-squared: 0.754,  Adjusted R-squared: 0.749
```

c. Standardized Residuals, Leverage and Outliers

As in the SLR, the basic diagnostic tools we use plots involving the least squares residuals.

Recall that the basic model assumptions are stated in terms of the true regression lines errors (ε) and we use the least squares residuals as proxies for these true errors.

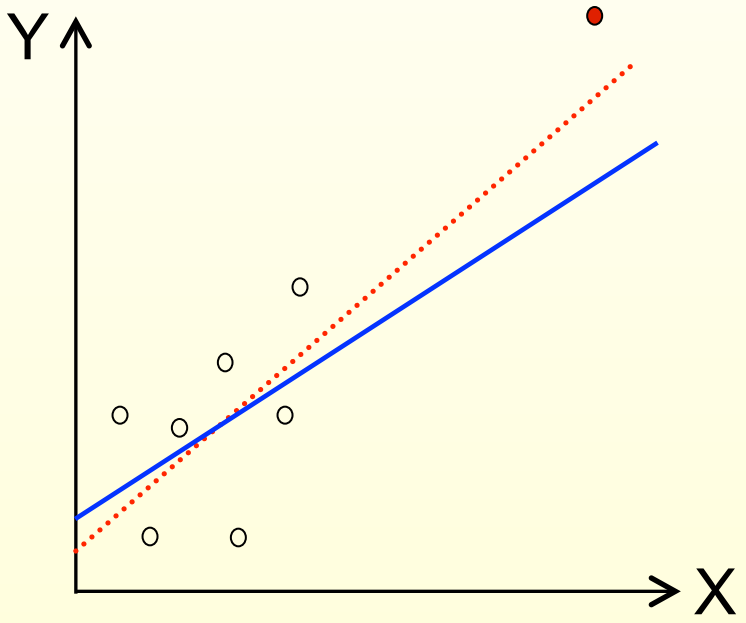
To what extent are e and ε similar and different?

The differences between least squares residuals and true errors show up at extreme x values.

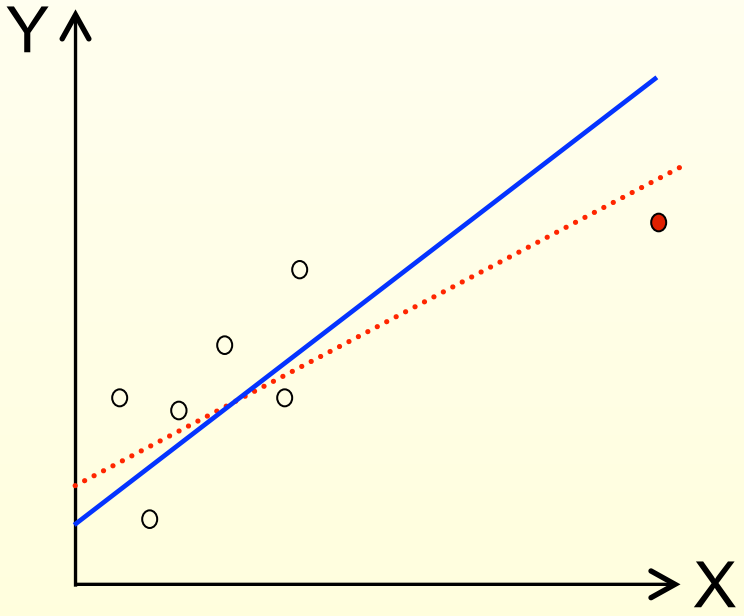
c. Standardized Residuals, Leverage, and Outliers

Now for some plots where ε and e are not at all alike. (blue line is the "true" line)

Large positive ε, e < ε

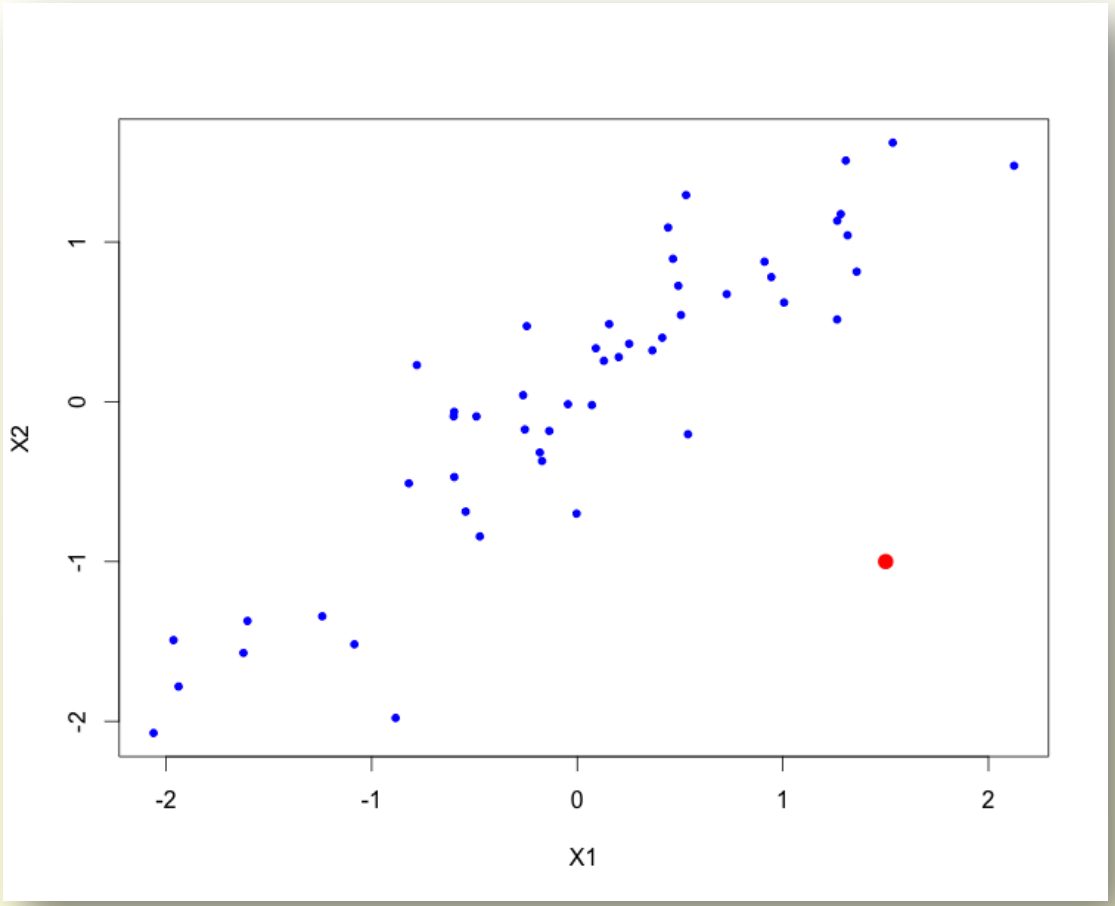


Large negative ε, |e| < |ε|



c. Standardized Residuals and Leverage

What about more than one regressor? Is the red point unusual?

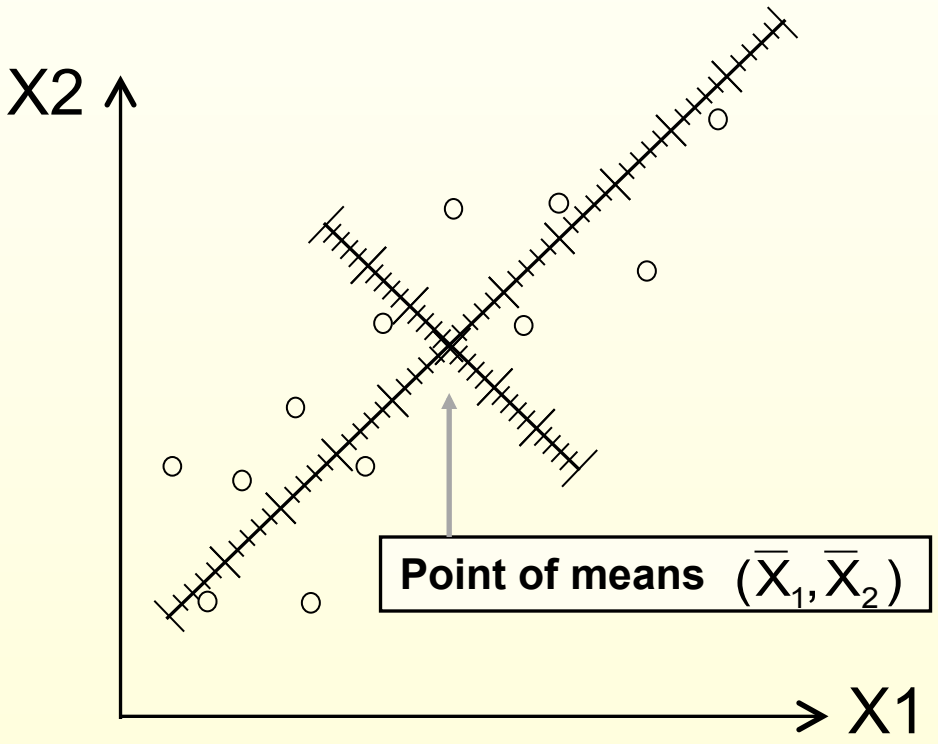


We need a measure of *remoteness* relative to the bulk of the X data.

This measure must work for more than 2 Xs.

c. Standardized Residuals and Leverage

The basic idea behind the measure used most commonly is to measure the distance from point of means.



The idea is that we measure distance from the point of means and adjust for the extent of variability along the principle axes of variation in the X data.

c. Standardized Residuals and Leverage

The distance measure motivated by the above diagram can be computed for each observation.

We can't write down any easy formula for the leverage except in the case of SLR:

$$\text{For SLR}: h_i = \frac{1}{N} + \frac{\left(X_i - \bar{X}\right)^2}{\sum_{j=1}^{N}\left(X_j - \bar{X}\right)^2}$$

$$\text{For MLR: } h_i = x_i'\left(X'X\right)^{-1}x_i$$

c. Standardized Residuals, Leverage, and Outliers

We can relate the variance of the least squares residuals to the h_i.

$$\mathrm{Var}\left(e_i\right)=\sigma^2\left(1-h_i\right)$$

Observations with large leverage (h_i) will have smaller residuals which is exactly the intuition we developed using the plots on the earlier slides.

We use the leverage measures to *standardize* the least squares residuals:

$$r_i=\frac{e_i}{s\sqrt{\left(1-h_i\right)}}\approx\frac{\varepsilon_i}{\sigma}\sim N\left(0,1\right)$$

c. Standardized Residuals and Leverage

Where does this formula for the variance of e come from?

$$e = \left(I - X\left(X'X\right)^{-1}X'\right)y = M_X y = M_X \varepsilon$$

$$\begin{aligned} Var\left(e\right) &= E\left[M_X \varepsilon \left(M_X \varepsilon\right)'\right] = M_X E\left[\varepsilon\varepsilon'\right] M_X \\ &= \sigma^2 M_X \end{aligned}$$

$$Var\left(e_i\right) = \sigma^2 (1 - x_i'\left(X'X\right)^{-1} x_i)$$

What is a large h_i value?

It turns out that $0 < h_i < 1$ and $\sum_{i=1}^{N} h_i = k+1$

Thus, the average h_i value is (k+1)/N

We can use the following useful rule:

$$\text{Flag if : } h_i > 3\frac{(k+1)}{N}$$

i.e. flag observations with h_i values > 3 times the average h_i

c. Standardized Residuals and Leverage

What is true:

- observations with large h_i values are remote in X space
- observations with large h_i values are the potential to influence the fitted line

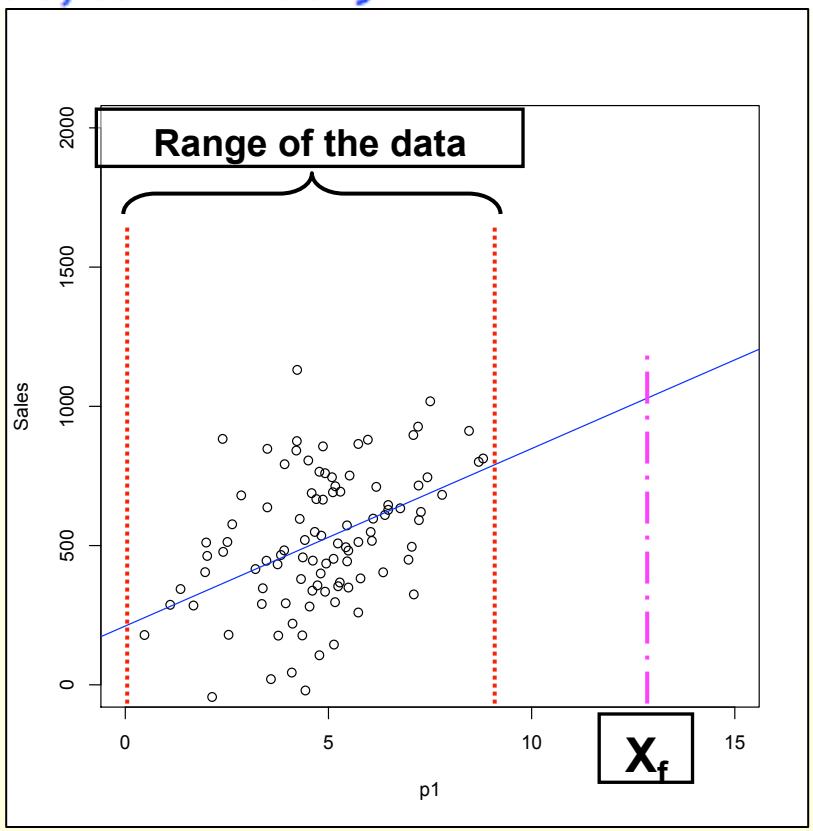
What is *not* true:

- observations with large h_i values have actually exerted influence
- observations with large h_i values should be deleted

c. Hidden Extrapolation and Leverage

Consider the following SLR example:

```
> out=lm(Sales~p1)
> plot(p1,Sales,ylim=c(0,2000),xlim=c(0,15))
> abline(out,col="blue")
```

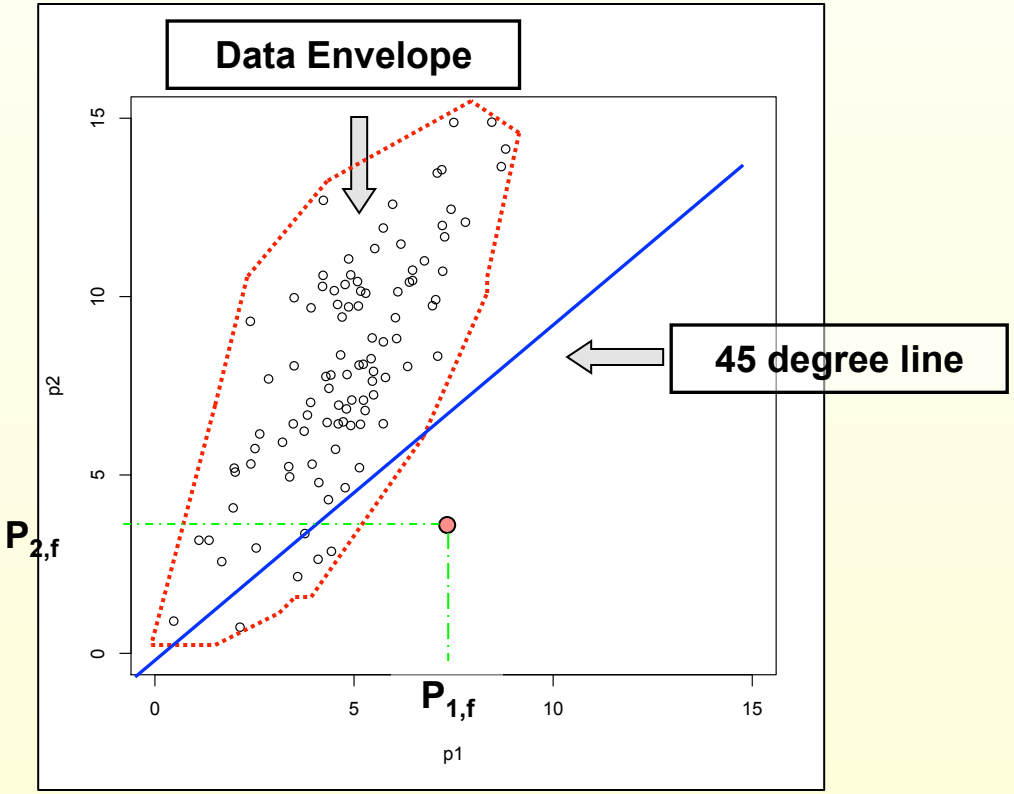


Outside the range of the data

c. Hidden Extrapolation and Leverage

Detecting this situation (prediction outside the range of the data) can be difficult with more than one X.

It can be made even more difficult if the X's are correlated. Consider the pricing example again.



c. Hidden Extrapolation and Leverage

The red point denotes a point in the P1-P2 space for which we would like to predict sales.

P1 > P2 is rarely seen in the data, thus this point is well outside any reasonable data envelope.

To forecast sales here is risky since we have no data nearby.

We would be relying exclusively on extrapolating from the model.

- The point is not unusual relative to the *marginal* distribution of P1 or P2. A P1 value of 7 or 8 is not at all unusual, nor is a P2 value of 6.
- The point is only unusual relative to the joint distribution of P1 and P2.

c. Hidden Extrapolation and Leverage

How do we detect this?

To avoid hidden extrapolation problems, we need a measure of remoteness of the new X values relative to the sample X values.

We can compute a h_i leverage measure for the new prediction point (note: remember h_i only depends on X values not on Y!).

That is we must compute for $x_f = (1,8,6)$

$$h_f = x_f^{'}\left(X'X\right)^{-1}x_f$$

c. Hidden Extrapolation and Leverage

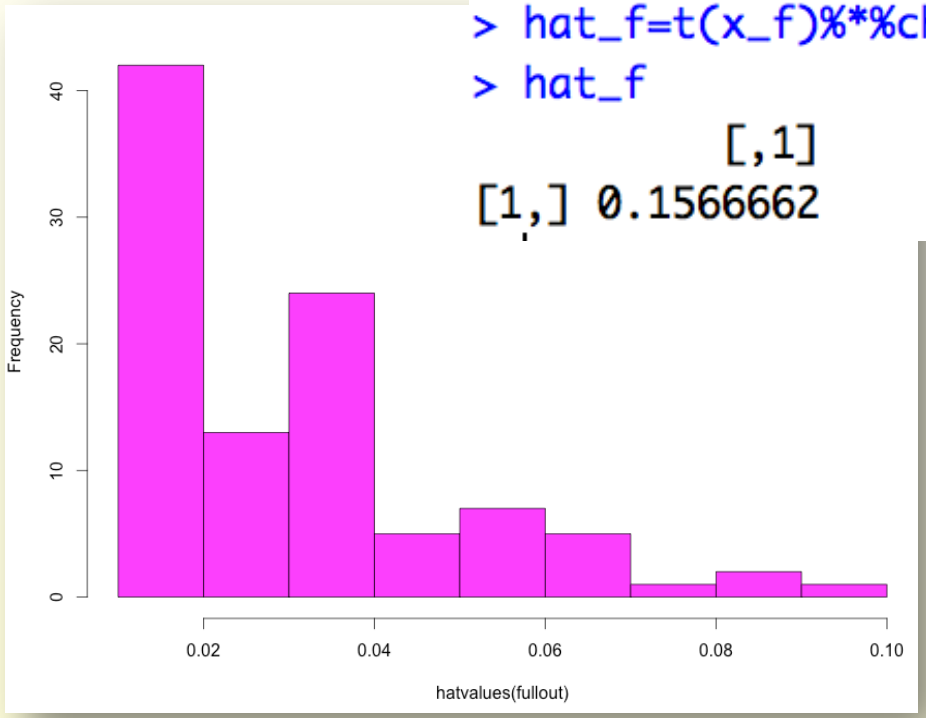
Let's try it in R.

```
> fullout=lm(Sales~p1+p2)
> summary(fullout)

Call:
lm(formula = Sales ~ p1 + p2)

Residuals:
    Min      1Q  Median      3Q     Max
-66.916 -15.663  -0.507  18.907  63.301

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  115.717      8.548   13.54   <2e-16 ***
p1           -97.657      2.669  -36.59   <2e-16 ***
p2           108.800      1.409   77.20   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1

Residual standard error: 28.42 on 97 degrees of freedom
Multiple R-squared: 0.9871, Adjusted R-squared: 0.9869
F-statistic:  3717 on 2 and 97 DF,  p-value: < 2.2e-16
```

c. Hidden Extrapolation and Leverage

Let's try it in R.

```
> data(multi)
> fullout=lm(Sales~p1+p2,data=multi)
> X=cbind(c(rep(1,nrow(multi))),multi$p1,multi$p2)
> H=X%*%solve(crossprod(X))%*%t(X)
> x_f=c(1,8,6)
> x_f=as.matrix(x_f,ncol=1)
> hat_f=t(x_f)%*%chol2inv(chol(crossprod(X)))%*%x_f
> hat_f
           [,1]
[1,] 0.1566662
```



c. Outlying Observations

I prefer the term “unusual” observations. How can we find unusual observations?

- Large h_i values
- Large r_i values

Observations with large r_i values are often called **“outliers”**

We should look for an “assignable cause” for outliers

- error in data entry
- change in accounting methods…

If the outlier reflects some phenomenon that is of no interest to the study and is unlikely to reoccur - *delete the point*

c. Outlying Observations

Sometimes outliers are the most interesting observations.

If y is a performance measure, then observations with large positive r_i are super-performers.

Residuals from the Market Model can be used for "event" studies:

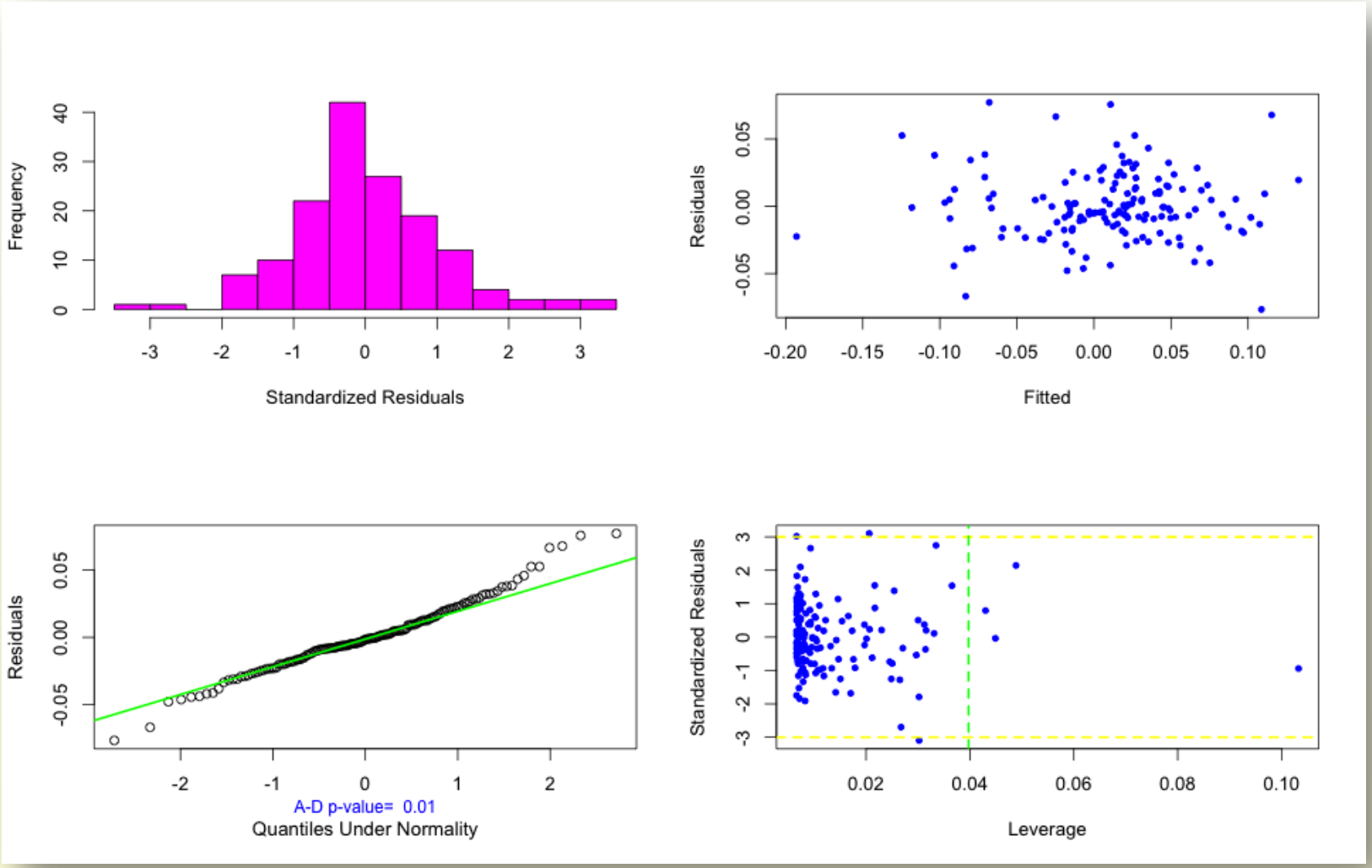
- changes in accounting standards
- regulatory changes
- opening of markets
- effect of tender offers on a firm

Don't routinely delete observations without cause

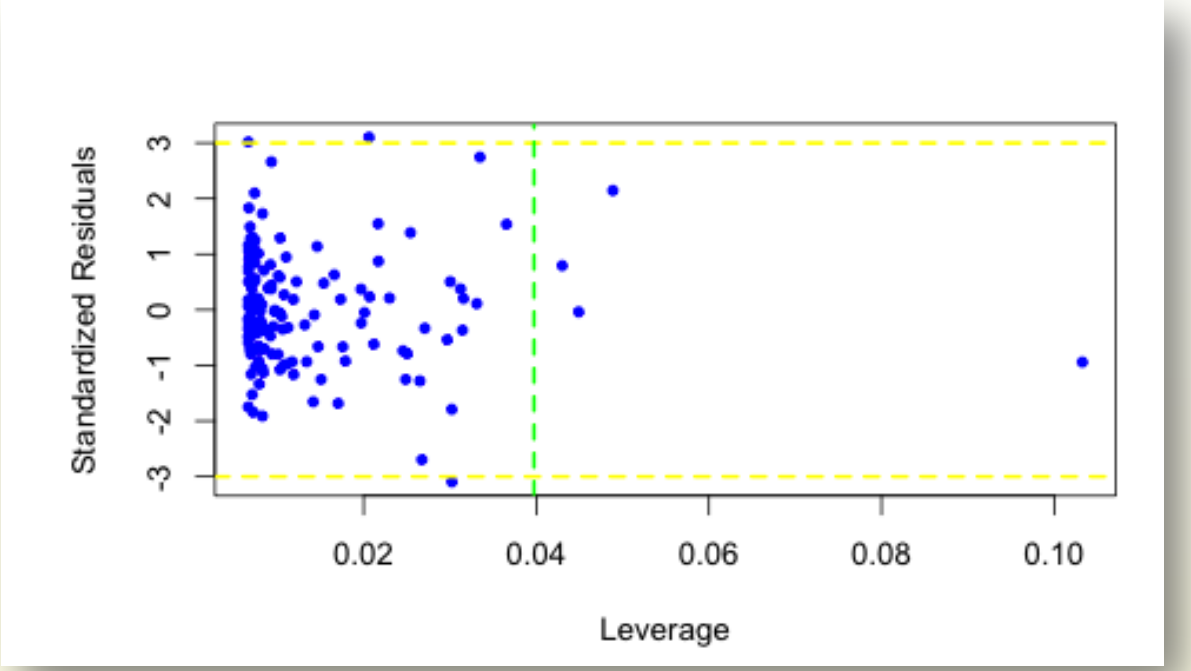
- this will bias the estimate of σ downward
- make you overly confident in your predictions and inferences!

c. Outlying Observations

Let's try it out on the Vanguard data



c. Outlying Observations



Outside the yellow: large std res

Right of green: large leverage

One large leverage value but small residual.

Get worried when both are big!

d. Nonlinearity

We have emphasized that the standard regression model is a linear conditional mean model.

We can check this assumption by:

a. plot residuals vs. fitted (overall non-linearity)

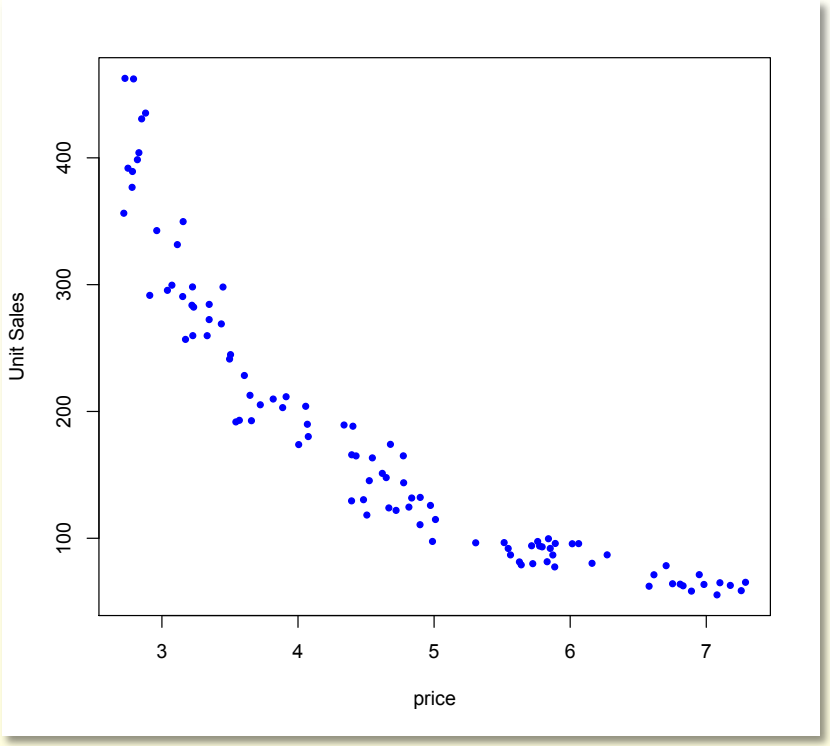
b. plot residuals vs. specific X vars (is non-linearity related to specific variables?)

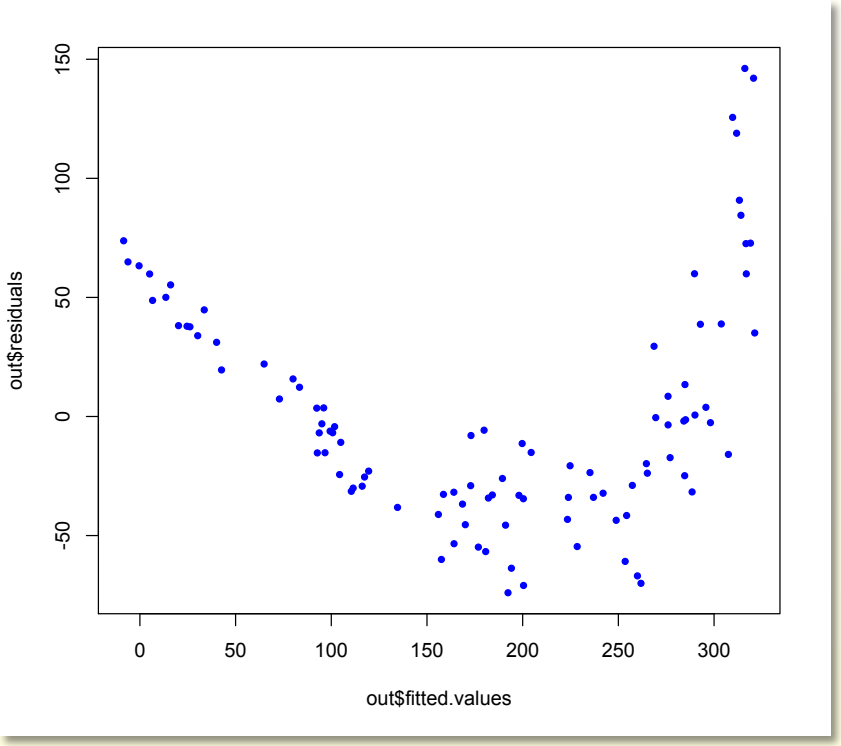
Why should I care?

incorporating non-linearity can improve predictive accuracy!

d. Nonlinear in Residuals Vs. Fitted

What about this dataset?





d. Nonlinearity

In many situations in practice, it is desirable to have some flexibility to specify non-linear regression functions.

The standard linear regression model can be "tricked" into displaying non-linearity by two techniques:

i. Adding additional independent variables

ii. Take transformations of the dependent variable

i. Adding additional independent variables

Trick 1: $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon$

Here we add the additional variable X^2 to the regression.

This allows the regression function to bend in a parabolic or quadratic curve

Remember that:

- If β_2 is negative, it "sheds water";
- if β_2 is positive, it "holds its water."

i. Adding additional independent variables cont.

With a multiple regression, we might consider adding the squares of two or more variables.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2 + \epsilon$$

d. Nonlinearity

Trick 2: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon$

Here we add the **interaction** term which is a new variable that consists of the product of X_1 and X_2.

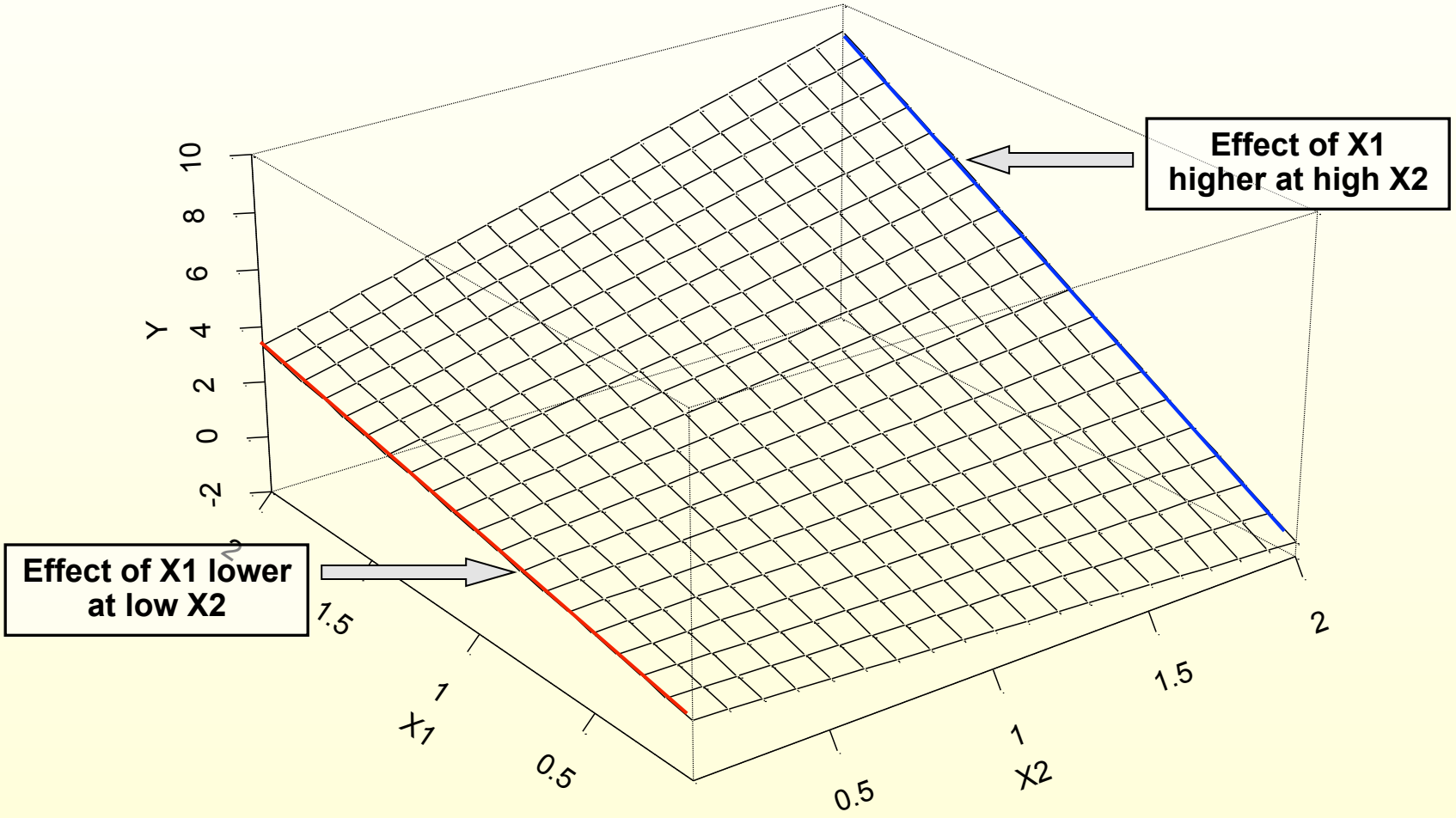
The effect of the interaction term is to make the derivative of the regression with respect to X(1 or 2) depend on the level of the other variable.

$$\partial E\left[Y \mid X_1, X_2 \right] / \partial X_1 = \beta_1 + \beta_3 X_2$$

Note here: the first term is what you get from the standard linear model. Don't include just the interaction term.

d. Nonlinearity

What does this look like?



d. Nonlinearity

We can put both ideas together and include squares and interaction terms for all variables.

What should you be careful about?

- too many coefficients (over-fitting)
- nonlinear terms are dangerous in out-of-sample forecasting.
- possible colinearity problems

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 + \varepsilon$$

ii. Take transformations of the dependent variable

The log transformation is the most common single transformation of the dependent variable.

Statisticians perform this by:

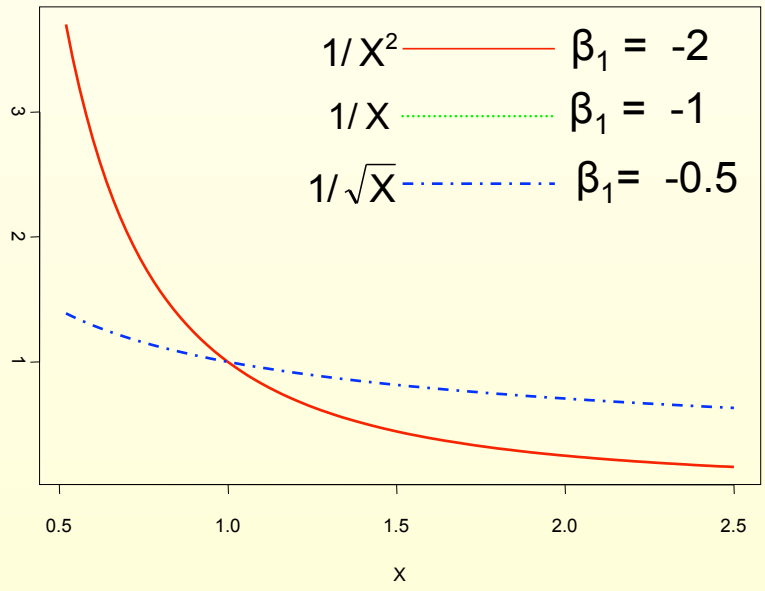
1. taking the log of the dependent variable
2. then entering the log of the dependent variable in the standard regression routine

d. Nonlinearity

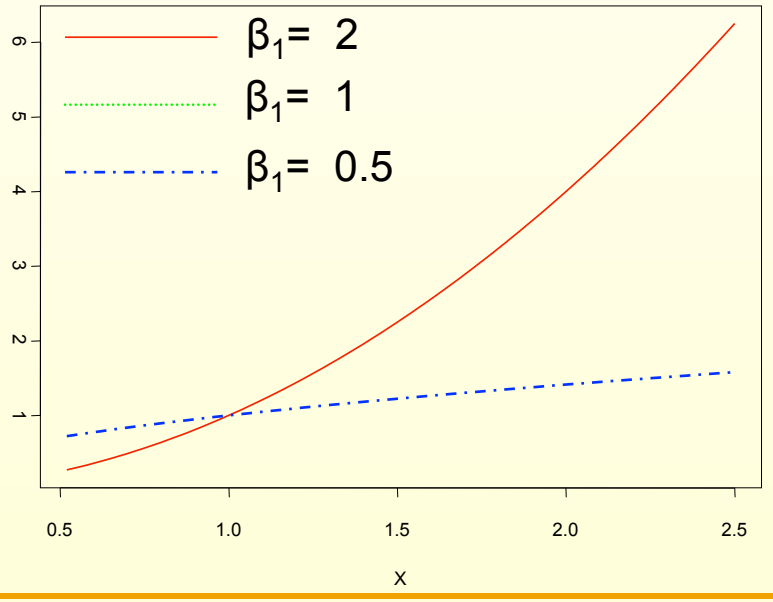
To see the effect of the log transform of Y. Let's first start with a model that is "compatible" with the log transform:

Basic multiplicative model given by: $E[Y|X] = AX^{\beta_1}$

Negative β_1



Positive β_1



d. Nonlinearity

If we take the logs of both sides of the multiplicative model, we get a model that is *linear in the logs.*

$$\log\left(E\left[Y|X\right]\right) = \log(AX^{\beta_1})$$
$$= \log(A) + \beta_1 \log(X)$$

This suggests a simple linear regression model which is linear in the logs:

$$\log(Y) = \beta_0 + \beta_1 \log(X) + \varepsilon$$

That is, we simply regress log(Y) on log(X).

d. Nonlinearity

One important thing to note is that β_1 now has the interpretation of an **elasticity**.

The elasticity is defined as the percentage change in Y for a given percentage change in X.

Elasticity:

$$E[\%\Delta Y|\%\Delta X]=E[\partial \log(Y)/\partial \log(X)]=\beta_1$$

Some also consider a semi-log model in which only Y is transformed by the log.

Semi-log:

$$\log(Y)=\beta_0+\beta_1 X+\varepsilon$$

e. Dummy Variables

Many variables or factors in the world are fundamentally *discrete*. Either you are male or female, have MBA or not,…

To accommodate these sorts of qualitative factors in regression models:

- use **dummy**, **binary** or **indicator** variables to measure these factors

A dummy variable is a variable that takes on the values of only 0 or 1.
Examples:

- Qualitative effects: (1 if on promotion, 0 if not)
- Temporal effects : (1 if Holiday season, 0 if not)
  (1 if Monday, 0 if not)
- Spatial (1 if in West Coast, 0 if not)
- Categorical (1 if female, 0 if not)

e. Dummy Variables

What does a dummy variable do if introduced into regression equation?

Dummy Variables allow the mean to shift. Let's look at example from the detergent data. "promoflag" is a dummy variable which flags those weeks where there was a promotion on the item.

Typically, a promotion is a price discount that is advertised in some what such as by signage in the store or some sort of advertisement.

$$\log\left(\text{q\_tide128}\right) = \beta_0 + \beta_1 \log\left(\text{p\_tide128}\right) + \beta_2 \text{promoflag} + \varepsilon$$

e. Dummy Variables

```
lm(formula = log(q_tide128) ~ log(p_tide128) + promoflag)

Residuals:
      Min        1Q    Median        3Q       Max
-4.588407 -0.459078 -0.009043  0.436672  2.978813

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    13.07862    0.13842  94.488   <2e-16 ***
log(p_tide128) -4.36631    0.06451 -67.687   <2e-16 ***
promoflag       0.14933    0.01572   9.501   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’

Residual standard error: 0.7335 on 14742 degrees of freedom
Multiple R-squared: 0.2454, Adjusted R-squared: 0.2453
F-statistic:  2397 on 2 and 14742 DF,  p-value: < 2.2e-16
```

```
data(detergent)
```

e. Dummy Variables

How do we interpret the coefficient on the promoflag variable?

It's the expected change in the log of quantity for a one unit change in promoflag (as always), holding all other indep variables constant!

What is change in log – it's percentage change.

So coefficient is the precent change in quantity expected from a promotion without a price change.

14.9 percent increase in sales!

More Than Two Levels

We can use dummy variables in situations in which there are more than two categories. One dummy variable is needed for each category (except a designated "base" category).

Example: scores in different sections of the same class by instructor

Y_i = test score

"Instructor" factor takes on three levels:

1 = prof A

2 = prof B

3 = prof C

We introduce two dummy variables:

X1 = 1 if B; 0 otherwise

X2 = 1 if C; 0 otherwise

e. Dummy Variables

The data would look like this

	Y	X1	X2
John	98	1	0
Nancy	87	1	0
Lester	81	0	1
Tom	92	0	1
Lisa	76	0	0
Sue	98	0	0

Instructor A's students fall into the intercept: (0 for X1 and 0 for X2)

What regression would we run?

$$Y_i = \beta_0 + \beta_1 X1_i + \beta_2 X2_i + \varepsilon_i$$

e. Dummy Variables

How do we interpret the β's?

$$E\left[Y \mid X1=1\right]=\beta_0+\beta_1$$

$$E\left[Y \mid X2=1\right]=\beta_0+\beta_2$$

Where is A? In the intercept:

$$E\left[Y \mid X1=X2=0\right]=\beta_0$$

Thus, we should interpret the betas as measure differences against a base case which is in the intercept.

β_1 = B-A

β_2 = C-A

e. Dummy Variables

We can generalize this to q different groups simply by putting in q-1 dummy variables for each of q-1 of the groups and "nominating" one group to be the intercept.

This always gives the regression coefficients an interpretation as the difference in means.

Let's look at an example of this.

Example: Stock Returns and the Weekend Effect

K. R. French (*Journal of Financial Economics* [1980])

Y = daily returns on S&P
X2 = 1 if Tuesday, 0 else
X3 = 1 if Wednesday, 0 else
X4 = 1 if Thursday, 0 else
X5 = 1 if Friday, 0 else

e. Dummy Variables

$$Y_i = \beta_0 + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4I} + \beta_5 X_{5i} + \varepsilon_i$$

Overall F is 25.4 compare with $F_{4,6019}$ distribution – reject.

$$b_0 = -.168, s_{b_0} = .022$$
$$b_2 = .184, s_{b_2} = .03$$
$$b_3 = .265, s_{b_3} = .03$$
$$b_4 = .213, s_{b_4} = .03$$
$$b_5 = .255, s_{b_5} = .03$$

Monday is a downer!

e. Categorical Variables and R

How does R handle variables that are purely categorical. The values of the variable indicate which category the observation is from and have no ordinal meaning.

Example: store variable in detergent. This is just an indicator of which store (out of 86) the observations are from. R stores this as a factor:

```
> str(detergent$store)
 Factor w/ 86 levels "2","5","8","9",..: 1 1 1 1 1 1 1 1 1 1 ...
> table(detergent$store)

  2   5   8   9  12  14  18  21  28  32  33  40  44  45  47  48  49  50  51  52
180 175 182 170 182 182 171 184 182 166 170 175 180 186 173 183 182 172 159 180
 53  54  56  59  62  64  67  68  70  71  72  73  74  75  76  77  78  80  81  83
179 183 184 171 186 173 168 181 170 183 186 183 183 181 185 187 185 185 181 187
 84  86  88  89  90  91  92  93  94  95  97  98 100 101 102 103 104 105 106 107
185 162 181 180 165 184 181 163 182 144 183 180 171 178 179 178 176 178 172 171
```

e. Categorical Variables and R

Put in dummy variables for each store. Do you have to sit there and create dummies? No, R knows what a factor is and creates them automatically.

```
lm(formula = log(q_tide128) ~ log(p_tide128) + store)

Residuals:
      Min        1Q    Median        3Q       Max
-4.246937 -0.361703 -0.009173  0.336968  3.275639

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    13.821333   0.136027 101.607  < 2e-16 ***
log(p_tide128) -4.538344   0.058748 -77.251  < 2e-16 ***
store5          0.021750   0.067081   0.324 0.745766
store8          0.107005   0.066413   1.611 0.107159
store9         -0.272842   0.067521  -4.041 5.35e-05 ***
```

Other coefs are cut off: where is store 2?

Dummy Variables and Handling Nonlinearity

Consider the example of modeling the effect of years of education on Salaries. What's wrong with just putting in Years of Ed in a regression?

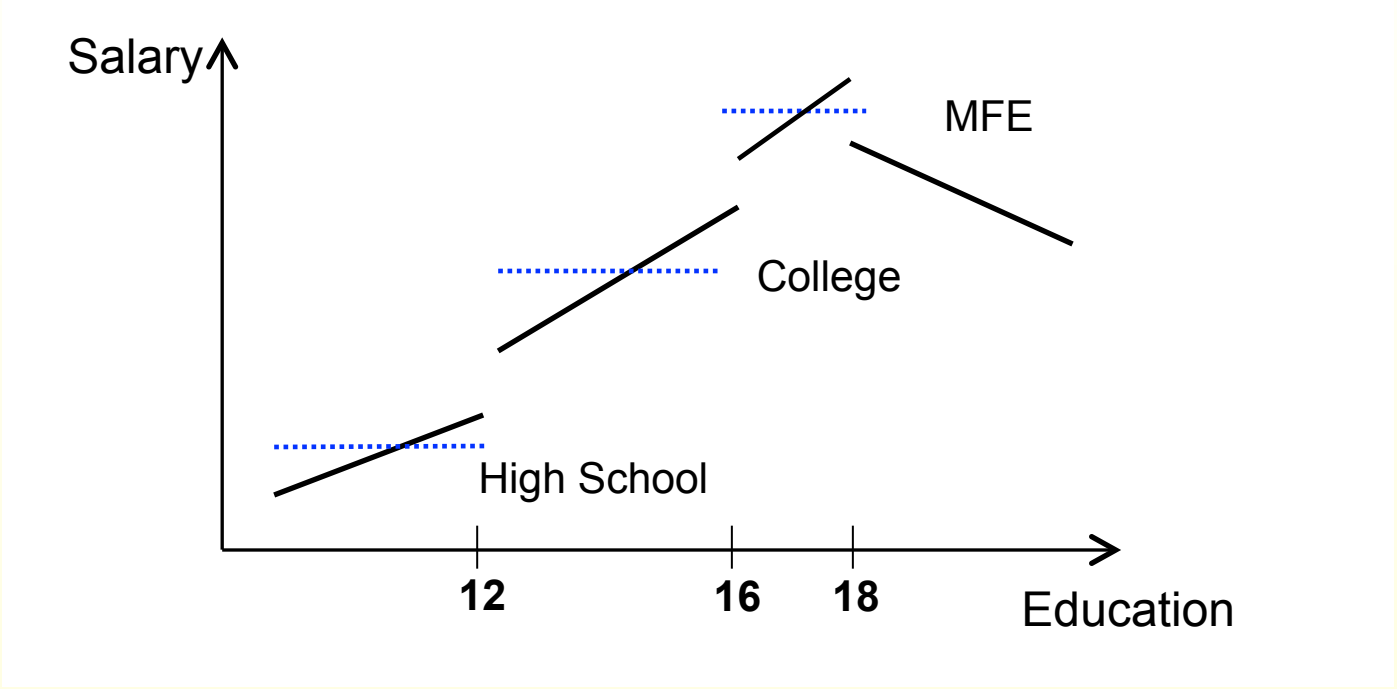
$$\text{Salary}_i = \beta_0 + \beta_1 \text{Yrs}_i + \varepsilon$$

There's a big difference between 11 and 12 years of education or 15 and 16!

Putting in Yrs^2 isn't going to help this!

e. Dummy Variables

The true regression or mean function might look like this:



We could approximate this regression function fairly well by putting in dummy variables for the major "breaks" in the picture.

$$Salary_i = \beta_0 + \beta_1 HS_i + \beta_2 College_i + \beta_3 Masters_i + \varepsilon_i$$

f. Heteroskedasticity

As we have seen heteroskedasticity is a situation in which the error terms do not all have the same variance.

This is usually detected with the residuals vs. fitted plot.

1. Effects of Heteroskedasticity

On Standard Errors:

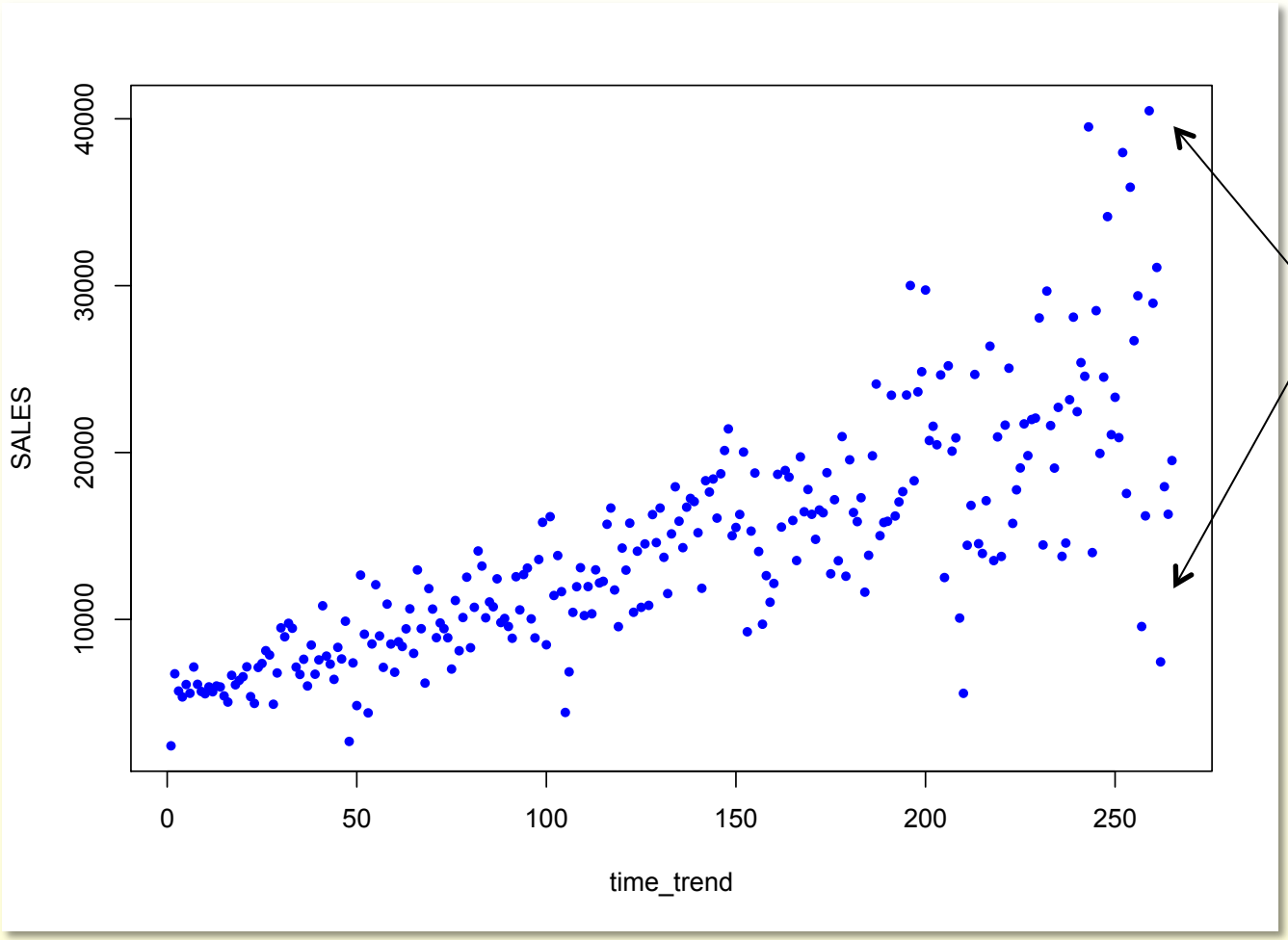
Standard error formulas are derived under the assumption of *homoskedasticity*. They are just plain wrong. Can be too small, i.e. the std errors printed out by lm() are less than the actual!

On Prediction Intervals:

Can be wrong everywhere! Too large in some places. Too small in others.

f. Heteroskedasticity

```
> data(fiber_conn_sales)
```



Large Var(ε) implies Prediction Interval should be wider!

2. **Correcting for Heteroskedasticity**

One popular approach is known as **variance stabilizing transformations.**

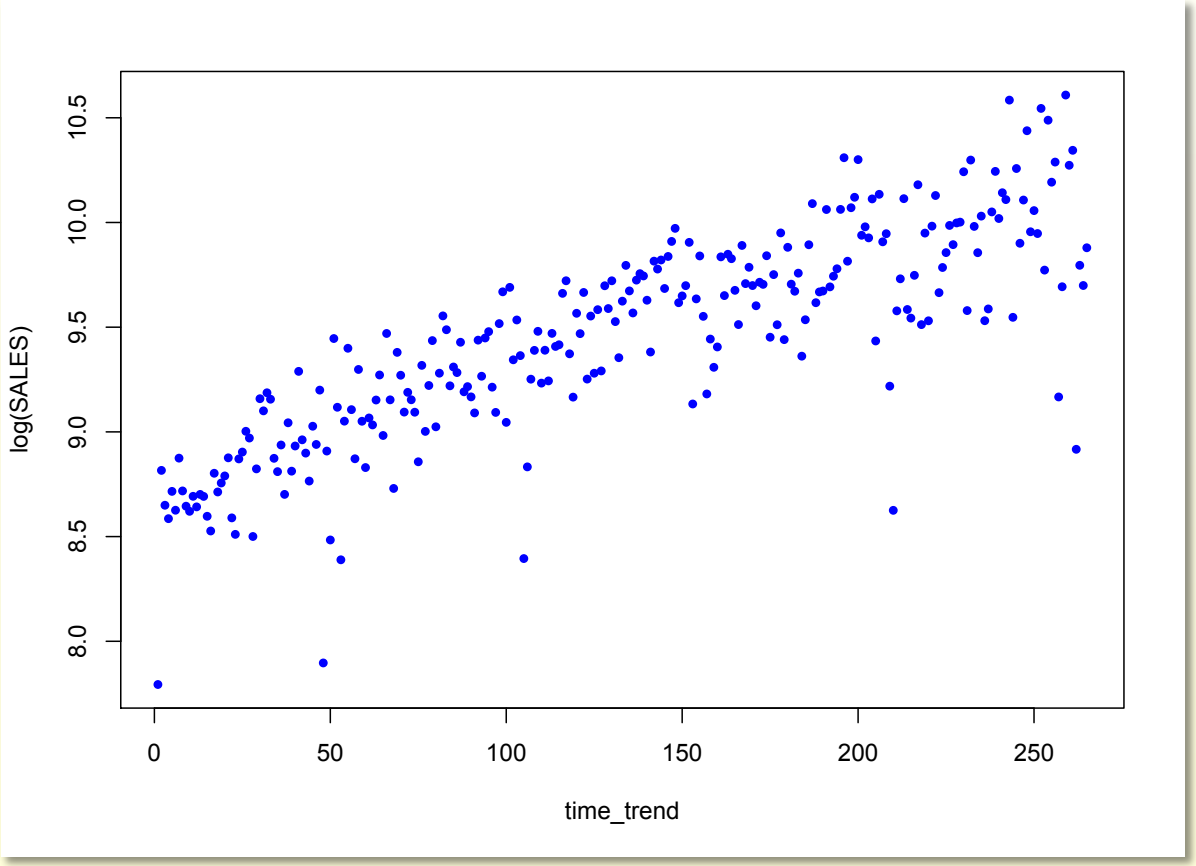
Taking logs of the dependent variable is one type of transformation often used with business oriented data to remove heteroskedasticity.

- use this when the variance increases with the level of the dependent variable

If $\mathrm{Var}\left(Y|X\right) \propto \mathrm{E}\left[Y|X\right]^2$, then the **log transform** is appropriate.

2. **Correcting for Heteroskedasticity**

Let's take the log and re-plot



f. Heteroskedasticity and Least Squares

In regression modeling, we assume that the errors have the same variance. A very general form of heteroskedasticity is that each error term has a different variance.

General Heteroskedasticity: $y_i = x_i'\beta + \varepsilon_i \quad Var(\varepsilon_i) = \sigma_i^2$

Note this includes the case in which the heteroskedasicity is linked to some variable or set of variables, $\sigma_i^2 = \exp(z_i'\gamma)$.

What is the variance of the standard least squares estimator?

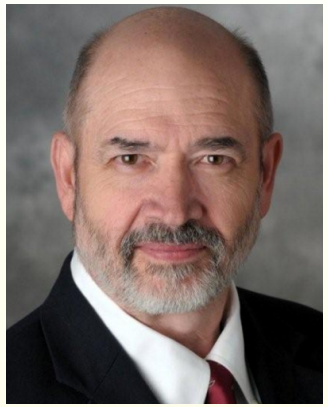
$$Var(b) = E\left[(b-\beta)(b-\beta)'\right] = E\left[(X'X)^{-1}X'\varepsilon\varepsilon'X(X'X)^{-1}\right]$$

$$= (X'X)^{-1}X'E\left[\varepsilon\varepsilon'\right]X(X'X)^{-1} = (X'X)^{-1}X'\Lambda X(X'X)^{-1}$$

f. Heteroskedasticity and Generalized Least Squares

where

$$\Lambda = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n^2 \end{bmatrix}$$



Can we estimate the Var(b)? This looks a bit "hopeless" in the sense that there are N different variance parameters. The insight of Hal White is that we really don't have to estimate all of these different variance parameters, what we have to do is to estimate linear combinations of them. Under some assumptions about the sequence of $\{\sigma_i^2\}$, we can estimate Var(b) for "large" sample sizes.

f. Heteroskedasticity

The White estimator is called a Heteroskedasticity Consistent estimator or a "HAC" estimator. Some call this a "robust" standard error.

$$\widehat{\mathrm{Var}(b)} = (X'X)^{-1} X'\hat{\Lambda}X (X'X)^{-1}$$

$$\hat{\Lambda} = \begin{bmatrix} e_1^2 & 0 & & 0 \\ 0 & e_2^2 & & \\ & & \ddots & 0 \\ 0 & & 0 & e_n^2 \end{bmatrix}$$

This is now implemented in virtually all econometrics software packages (see R package, sandwich). I have also implemented it in the function, `lmSumm(lmfit,HAC=TRUE)`.

f. Heteroskedasticity

Here is an example of using HAC corrections.

```
> lmSumm(lm(SALES~time_trend))
Multiple Regression Analysis:
    2 regressors(including intercept) and 265 observations

lm(formula = SALES ~ time_trend)

Coefficients:
            Estimate Std Error t value p value
(Intercept)  4704.00    512.50    9.18       0
time_trend     72.46      3.34   21.69       0
---
Standard Error of the Regression:  4159
Multiple R-squared:  0.642  Adjusted R-squared:  0.64
Overall F stat: 470.62 on 1 and 263 DF, pvalue= 0
```

```
> lmSumm(lm(SALES~time_trend),HAC=TRUE)
Multiple Regression Analysis:
    2 regressors(including intercept) and 265 observations
    with heteroskedastic|autocorrelation consistent standard errors
    Lag truncation = 0

lm(formula = SALES ~ time_trend)

Coefficients:
            Estimate Std Error t value p value
(Intercept)  4704.00   388.200   12.12       0
time_trend     72.46     4.228   17.14       0
---
Standard Error of the Regression:  4159
Multiple R-squared:  0.642  Adjusted R-squared:  0.64
```

big set

f. Testing for Heteroskedasticity

A popular test for heteroskedasticity is the so-called Breusch-Pagan test. Here the null hypothesis is that there is homoskedasticity and the alternative hypothesis is that there is a form of conditional heteroskedasticity in which the error variance is driven by a set of variables, z_i.

$$H_0 : \sigma_i^2 = \sigma^2$$

$$H_A : \sigma_i^2 = \sigma^2 f\left(\alpha_0 + z_i'\alpha\right)$$

The test is performed in two steps:

1. Perform standard regression.
2. Regress $e_i^* = e_i^2 \Big/ \left(e'e/n\right)$ on z_i
3. Compute the SSR from the regression in 2).

$$\tfrac{1}{2}SSR \sim \chi^2_{\dim(z)} \text{ under } H_0$$

g. Bootstrapping the Regression Model

While least squares estimation of regression coefficients can be justified as a BLUE estimator without specification of the error distribution, most of the test statistics we use (e.g. t and F tests) are based on the assumption of normal error terms.

In addition, we sometimes what to make inferences about non-linear functions of model parameters. For example, consider a semi-log model in which the log of Sales is regressed on an Advertising variable (say GRPs).

$$\ln S_i = \alpha + \beta Ad_i + \varepsilon_i$$

We want to consider the effect on Sales of increasing the Ad variable by 1 unit.

$$\ln S_1 - \ln S_0 = \beta \times 1 \quad \text{or}$$

$$\frac{S_1}{S_0} = \exp(\beta)$$

g. Bootstrapping the Regression Model

Thus, we can interpret, $\exp(\beta)$, as the multiple of sales that is simulated by increasing the Ad variable by one unit. Sometimes this is called "lift factor" for this Ad variable. This is a non-linear function of the regression coefficient. What is the standard error of the lift factor?

$$\hat{\gamma} = \exp(\hat{\beta})$$

The "bootstrap" method provides an answer:

1. A way of considering the distribution of test statistics under non-normal errors.
2. A way of computing the standard error for nonlinear functions of regression parameters.

g. Bootstrapping the Regression Model

The idea of the bootstrap is simple: Let's view our sample of data as the population. Then let's sample from this population by putting mass 1/n on each sample data point. These "samples" are called Bootstrap samples. We then compute the estimator on each of these Bootstrap samples. This was invented by Brad Efron.

Let $\Im_n$ be our sample $\Im_n = \{(y_i, x_i), i = 1, \ldots, n\}$

We draw B bootstrap samples from this sample using a uniform distribution and *replacement*.

Let $B_s, s = 1, \ldots, B$ be the bootstrap samples.

For each bootstrap sample we compute, $b_s = \hat{\beta}_{B_s}$.

The idea of the bootstrap is to use set of bootstrap estimates to approximate the sampling distribution of whatever estimator we are using, $b = \hat{\beta}$.

g. Bootstrapping the Regression Model

In other words, we saying:

Bootstrap idea: Use the distribution of $b_{B_s} - b$ to approximate the sampling distribution of $b - \beta$.

This has a formal justification. The bootstrap can provide a good approximation to the sampling distribution for large samples. Moreover, the Bootstrap can provide a very good approximation to the distribution of so-called Pivotal quantities (this is a statistic whose sampling distribution does not depend on unknown parameters like t or f stat whose null distribution just depends on degrees of freedom parameters).

g. Bootstrapping the Regression Model

Bootstrap confidence intervals are constructed using a slightly different formula that the standard Confidence Intervals (see the appendix for an explanation).

$$\left[2b - b_s^{\left[1-\alpha/2\right]}, 2b - b_s^{\left[\alpha/2\right]}\right]$$

Note that the subscript “bracket” notation means the appropriate quantile of the bootstrap distribution.

g. Bootstrapping the Regression Model

Let's try bootstrapping a regression using the Vanguard data.

What do we need to do? We need to draw "samples" from our observations with replacement. Here's the regression we are going to bootstrap.

```
> Reg_data=data.frame(Van_mkt$vwretd,Van_mkt$VGENX)
> Reg_data=na.omit(Reg_data)
> colnames(Reg_data)=c("GENX","VW")
> N=nrow(Reg_data)
> regression=summary(lm(GENX~VW,data=Reg_data))
> regression

Call:
lm(formula = GENX ~ VW, data = Reg_data)

Residuals:
      Min        1Q    Median        3Q       Max
-0.114418 -0.019591 -0.001443  0.020898  0.107473

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.004663   0.001914   2.437   0.0153 *
VW          0.455401   0.031153  14.618   <2e-16 ***
```

g. Bootstrapping the Regression Model

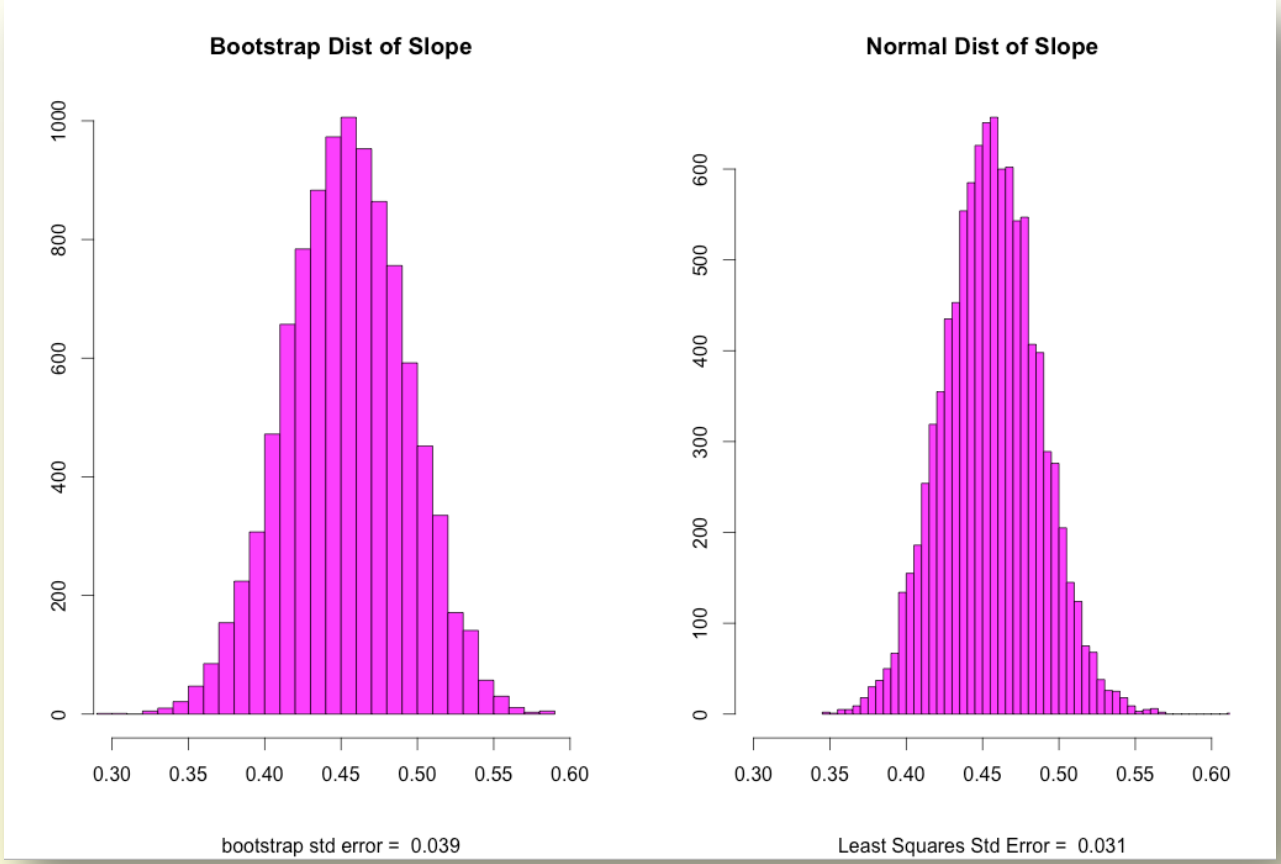
Now let's draw B=10,000 bootstrap samples from this dataset.

```
> B=10000
> BS_coefs=matrix(0,nrow=B,ncol=2)
> for(b in 1:B){
+   BS_sample=Reg_data[sample(1:N,size=N,replace=TRUE),]
+   sum=summary(lm(GENX~VW,data=BS_sample))
+   BS_coefs[b,]=sum$coef[,1]
+   }
```

The key here is that we are sampling the row indices to get our bootstrap samples.

g. Bootstrapping the Regression Model

Under normal errors, we know that the least squares slopes are normally distributed so let's compare the bootstrap distribution to the normal distribution. Slightly skewed with a higher standard deviation.



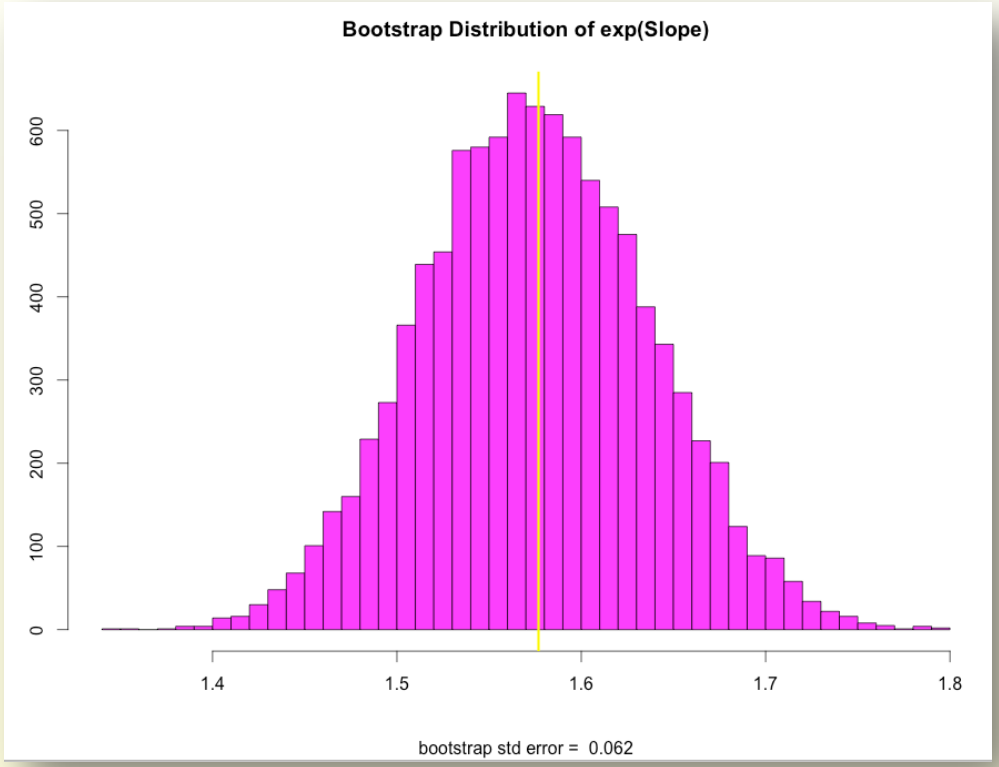
g. Bootstrapping the Regression Model

Now, let's compute the CI for the slope and compare to the standard interval which is based on the assumption of normal errors. The bootstrap interval is wider and no longer symmetric!

```
> int=quantile(BS_coefs[,2],probs=c(.975,.025))
> CI.pivotal.bootstrap = c(2*lsq.slope-int[1],2*lsq.slope-int[2])
> CI.normal.theory=c(lsq.slope+qt(.025,df=347)*lsq.slope.stderr,
+                    lsq.slope+qt(.975,df=347)*lsq.slope.stderr)
> CI.mat=rbind(CI.normal.theory,CI.pivotal.bootstrap)
> colnames(CI.mat)=c("Lower","Upper")
> rownames(CI.mat)=c("Normal Theory","Bootstrap Pivotal")
> print(CI.mat)
                     Lower     Upper
Normal Theory     0.3941290 0.5166730
Bootstrap Pivotal 0.3811393 0.5342607
```

g. Bootstrapping the Regression Model

Now, let's try bootstrapping the distribution of a nonlinear function of the slope (why don't we need to bootstrap linear functions?). Totally trivial – we just transform all of the bootstrap estimates.




```
> hist(exp(BS_coefs[,2]),breaks=40,col="magenta",xlab="",ylab="",
+      main="Bootstrap Dist of exp(Slope)",
+      sub=paste("bootstrap std error = ",round(sd(exp(BS_coefs[,2])),digits=3)))
```

g. Bootstrapping the Regression Model

And Bootstrap C.I.

```
> int=quantile(exp(BS_coefs[,2]),probs=c(.975,.025))
> CI.pivotal.bootstrap = c(2*exp(lsq.slope)-int[1],2*exp(lsq.slope)-int[2])
> names(CI.pivotal.bootstrap)=c("Lower","Upper")
> CI.pivotal.bootstrap
   Lower    Upper
1.455252 1.696375
```

h. LASSO – Introduction

The Least Absolute Shrinkage and Selection Operator is a model selection method.

Instead of fitting a model with a subset of predictors, the LASSO uses all predictors with a technique that *constrains* or "*regularizes*" the coefficient estimates, or equivalently, that *shrinks* the coefficient estimates toward zero.

There is an equivalent Bayesian approach, but the original motive for regularization was just "penalizing complicated functions"

We'll see that instead of simply minimizing loss (as with OLS), the LASSO minimizes loss + a complexity-penalty

h. LASSO – Bias/Variance Tradeoff

Recall the mean squared error

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

Under the assumptions of iid and constant error variance, we can show that

$$MSE = \sigma^2 + Bias^2 + Variance$$

The first term is the "irreducible noise" that cannot be eliminated by modeling. The latter terms show that there is a *bias-variance tradeoff*.

- Simple models tend to have high bias and low variance.
- Complex models that over-fit the data have low bias and high variance.

h. LASSO – Loss Function

OLS comes from minimizing the residual sum of squares (RSS):

$$\hat{\beta}_{OLS} = \text{argmin} \left\{ \sum_{i=1}^{n} (y_i - \hat{y})^2 \right\}$$

And we have shown $\hat{\beta}_{OLS}$ to have the minimum variance out of all unbiased estimators. But might we be willing to accept a little bias for a reduction in variance?

The LASSO minimizes the RSS + a penalty:

$$\hat{\beta}_{L1} = \text{argmin} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \left\| \beta \right\|_1 \right\} \quad \text{where} \quad \lambda \left\| \beta \right\|_1 = \lambda \sum_{j=1}^{J} \left| \beta_j \right|$$

h. LASSO – Standardizing Variables

Because our criteria is to minimize RSS + penalty, we want both a small RSS and a small penalty.

A small penalty is achieved by making the sum of the absolute values of β small. Thus, the coefficient estimates will be shrunk toward (or set equal to) zero.

Notice that you would get different coefficient estimates if you changed the units of the variables. Consider:

- X_1 in millions of dollars and X_2 in acres, vs.
- X_1 in dollars and X_2 in square feet

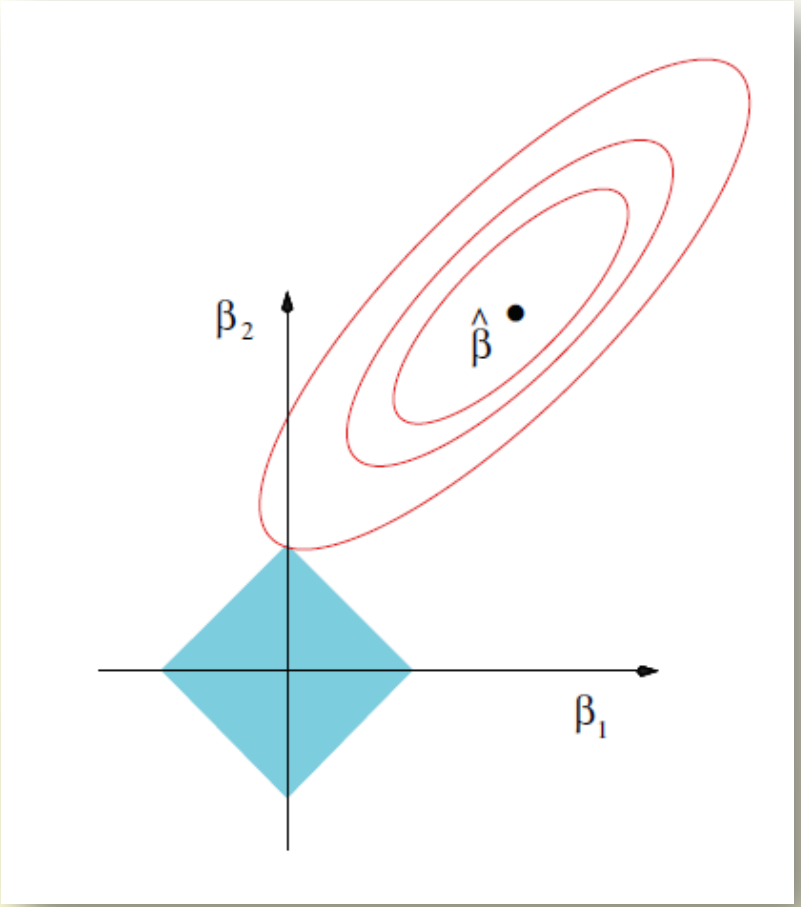
For this reason, variables should be standardized (i.e., have unit variance) before estimating a LASSO model.

h. LASSO – Penalty

Red lines are the contours of the RSS function

Blue area is the contour of the penalty (aka constraint) function

Notice that a solution will often set one of the coefficients equal to zero – here $\hat{\beta}_1 = 0$



h. LASSO – Lambda

The LASSO introduces a new "complexity" parameter λ, which controls the amount of shrinkage:

the larger the value of λ, the greater the amount of shrinkage.

Q. How do we choose or find a value for λ?

A. One method is *k*-fold cross validation:

- For each of J values of the tuning parameter: $\lambda_1, \ldots, \lambda_J$
  - Divide your dataset into *k* subsets
  - "Leave out" the i^{th} subset
  - Fit your model on the other *k*-1 subsets
  - Use your model to calculate the prediction error on the i^{th} subset
  - Do this for all λ_j's and subsets
- Select the λ_j for which the average prediction error is smallest

h. LASSO – Example

Let's get the returns for the S&P 500 Index as well as the returns for all stocks U.S. common stocks from the `SP500` dataset in the `DataAnalytics` package

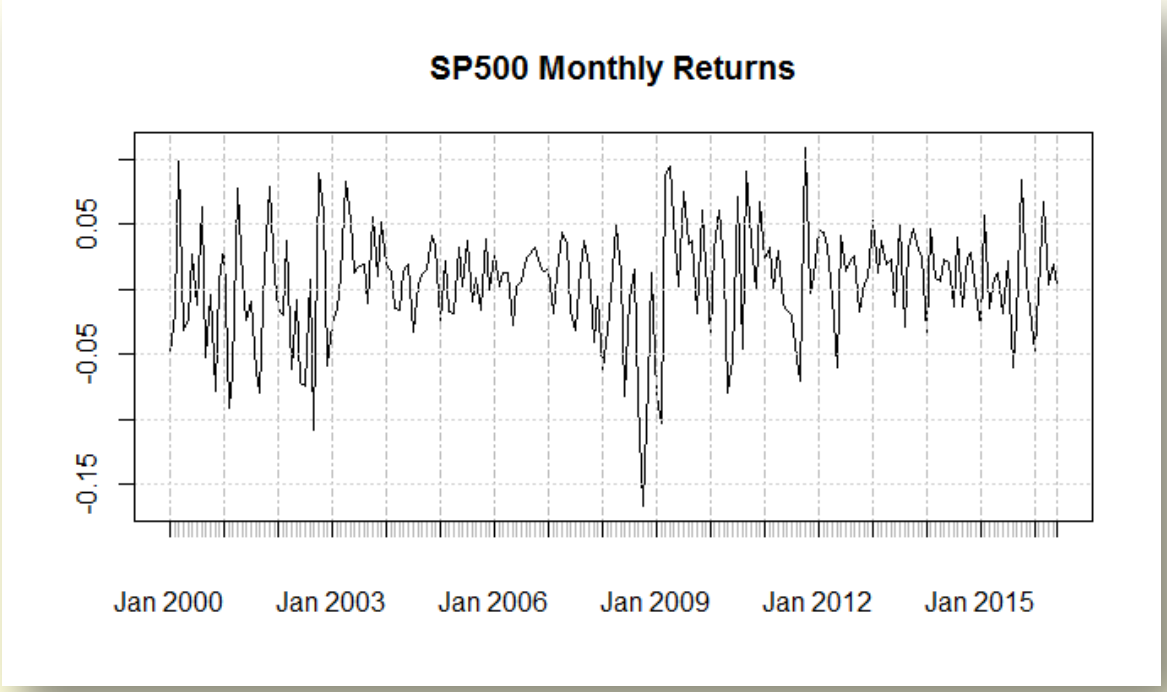
We'll use the LASSO method to pick a subset of stocks that best "explain" the Index return.

```
#data are time series - need zoo package
library("zoo")

#analysis is implemented via the glmnet package
library("glmnet")

#data are in DataAnalytics package on BitBucket
devtools::install_bitbucket("perossichi/DataAnalytics")
library("DataAnalytics")
data("SP500")

#extract "x" and "y"
index <- as.matrix(SP500$sp)
stocks <- as.matrix(SP500$stocks)
```

h. LASSO – Example

The data are from CRSP and are comprised of 198 monthly returns from January 2000 through June 2016.

In addition to the Index returns, the data include returns for 1,670 U.S. common stocks that had a complete time series of data



h. LASSO – Example

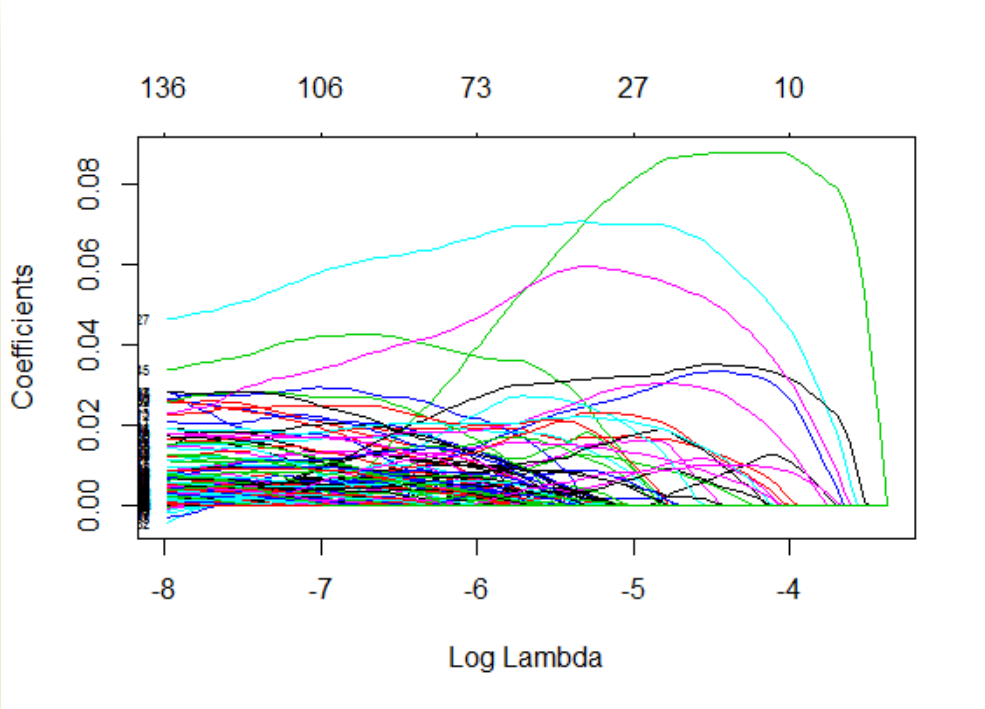
We fit the LASSO and find λ by 10-fold cross-validation.

`glmnet` standardizes the data to have unit variance and fits the model for a sequence of λ values.

Plotting the output of `glmnet` produces a plot where each line is the value of a coefficient across various values of λ.

Larger values of λ sets more coefficient estimates to zero.

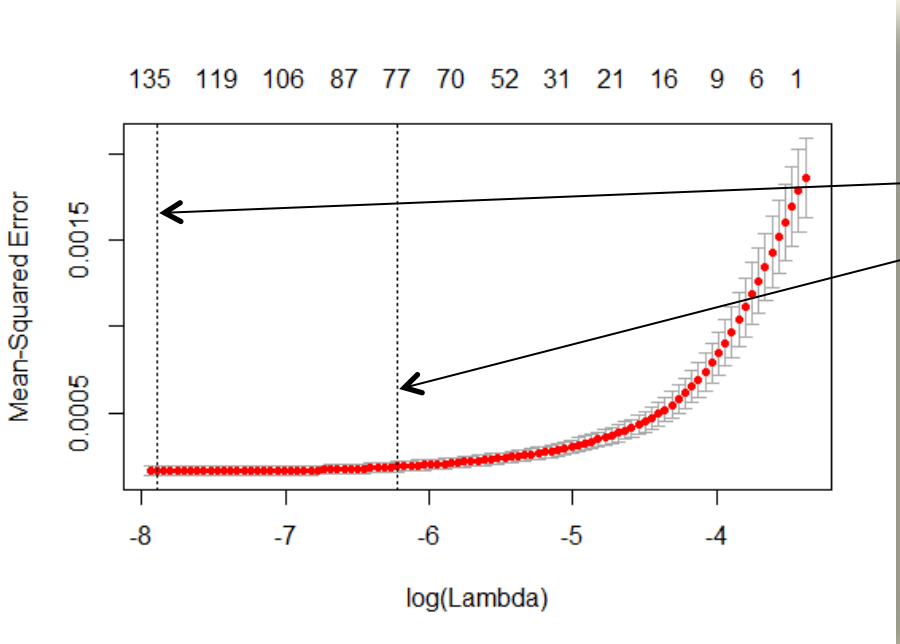
```
#fit lasso model
out <- glmnet(y = index, x = stocks, alpha = 1)
plot(out, xvar = "lambda", label = TRUE)
```



h. LASSO – Example

To see how the MSE changes as we change λ, use `cv.glmnet`

```
#10-fold cross validation
crossval <- cv.glmnet(y = index, x = stocks)
plot(crossval)
```



The vertical lines indicate the values for λ that

- Minimize the MSE
- Are within 1 standard error of the minimized MSE

125 coefs nonzero at log(λ) = -7.8

84 coefs nonzero at log(λ) = -6.3

h. LASSO – Example

Small values of -log(λ), produce models with a small number of non-zero coefficients. For example, if we set log(λ)=-4, then there are 11 non-zero coefficients.

```
> coefs <- coef(crossval, s = exp(-4))
> cbind(c("(Intercept)", coefs@Dimnames[[1]][coefs@i]), round(coefs@x,4))
      [,1]          [,2]
 [1,] "(Intercept)" "0.0025"
 [2,] "MSFT"        "0.087"
 [3,] "IMKTA"       "0.0014"
 [4,] "GD"          "0.0437"
 [5,] "PFE"         "0.0264"
 [6,] "F"           "0.0109"
 [7,] "REX"         "0.0137"
 [8,] "INT"         "0.0316"
 [9,] "PII"         "1e-04"
[10,] "NBTB"        "0.0306"
[11,] "ARKR"        "0.0083"
```

Note: PFE is Pfizer, MSFT is Microsoft, F is Ford, …

Appendix. Bootstrapping CIs Explained

Bootstrap confidence intervals are constructed using a slightly different formula than standard Confidence Intervals

Recall the where the CI comes from:

$$\frac{b-\beta}{s_b} \sim t_\nu \Rightarrow \Pr\left[b - t^*_{1-\alpha/2} s_b < \beta < b + t^*_{1-\alpha/2} s_b \right] = 1-\alpha$$

$$\Pr\left[b - t^*_{1-\alpha/2} s_b < \beta < b - t^*_{\alpha/2} s_b \right] =$$

$$\Pr\left[b - \left(b-\beta\right)^{\left[1-\alpha/2\right]} < \beta < b - \left(b-\beta\right)^{\left[\alpha/2\right]} \right] = 1-\alpha$$

Appendix. Bootstrap Confidence Intervals Explained

Here $\left(b-\beta\right)^{\left[1-\alpha/2\right]}$ means the $100\times\left[1-\alpha/2\right]$ percentile from the sampling distribution of $\left(b-\beta\right)$

The bootstrap idea is to use the quantiles of $\left(b_s-b\right)$ to approximate the quantiles of $\left(b-\beta\right)$

$$\left[b-\left(b-\beta\right)^{\left[1-\alpha/2\right]},b-\left(b-\beta\right)^{\left[\alpha/2\right]}\right]\approx\left[b-\left(b_s-b\right)^{\left[1-\alpha/2\right]},b-\left(b_s-b\right)^{\left[\alpha/2\right]}\right]$$

Hence the Bootstrap Pivotal Interval is:

$$\boxed{\left[2b-b_s^{\left[1-\alpha/2\right]},2b-b_s^{\left[\alpha/2\right]}\right]}$$

Glossary of Symbols

h_i - Leverage measure

Important Equations

$$se\left(b_j\right) = \frac{s}{\sqrt{SSE_{j \mid others}}}$$

std error of multiple regression coef

$$\text{For SLR:} \quad h_i = \frac{1}{N} + \frac{\left(X_i - \bar{X}\right)^2}{\sum_{j=1}^{N}\left(X_j - \bar{X}\right)^2}$$

leverage value for Simple LR Model

$$r_i = \frac{e_i}{s\sqrt{\left(1-h_i\right)}} \approx \frac{\varepsilon_i}{\sigma} \sim N\left(0,1\right)$$

definition of standardized residual

Important Equations

$$\text{Flag if : } h_i > 3\frac{(k+1)}{N}$$

Rule for flagging obs with high potential influence

LASSO Loss Function

$$\hat{\beta}_{L1} = \arg\min\left\{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{J}\left|\beta_j\right|\right\}$$

- `hatvalues(model)`: Return the h_i values of a model
- `Name[-2,]`: the second row is deleted from a matrix or data frame.
- `lmSumm(lmfit,HAC=TRUE)`: compute heteroskedasticity-corrected standard errors for regression output stored in lmfit.
- `glmnet(y,x,alpha,…)`: LASSO model fit
- `cv.glmnet(y,x,…)`: LASSO cross-validation for λ