Lesson 1 - Frequency and Response Latencies

Erin M. Buchanan

01/16/2019

Language Topics Discussed

- English Lexicon Project
- The Subtitle Projects
- Extension to Priming Projects

English Lexicon Project

- http://elexicon.wustl.edu/
- ELP was a large undertaking that helped kick start the recent uptick in standardized behavioral data for language researchers
  - Corpora have been around, so have databases, but in the last 10 years, this field has grown exponentially
- The data contains 40K + words and 40K + nonwords with many characteristics
- Lexical Decision and Naming Tasks included

English Lexicon Project

- https://www.psytoolkit.org/experiment-library/ldt.html
  - However, you would only see one word at a time (in this example you see two)
- In the naming task, you would be asked to read those words aloud, one at a time
- Output we are interested in is how the lexical variables might predict the response latencies

Subtitle Projects

- Starting with Brysbaert and New (several papers), there was a movement to rethink frequency and its relation to predicting language results
- Traditionally, two sources of frequency were used:
  - The Brown Corpus: Kucera and Francis (1967)
  - HAL Corpus: Burgess and Livesay (1998)
- However, we weren't sure these were the best estimators of frequency

Subtitle Projects

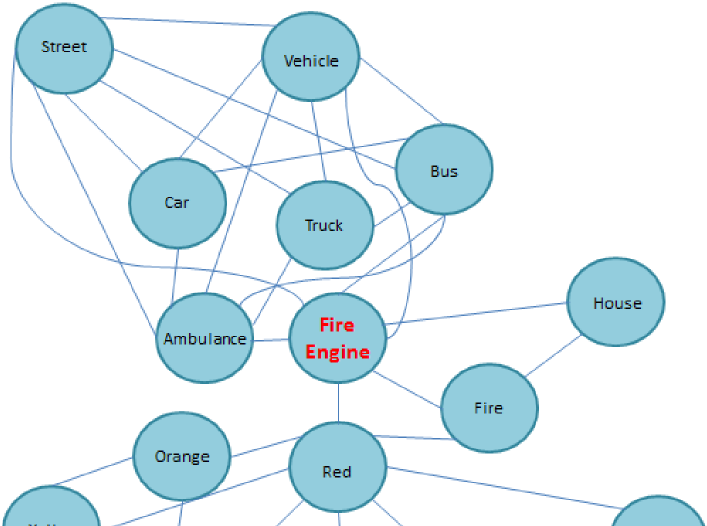
- http://subtlexus.lexique.org/
- Downloaded subtitles from www.opensubtitles.org with 50 million words+
- Provided both estimates for subtitles and music lyrics
- Models estimating lexical decision and naming times indicate these estimations of frequency are better predictors
- Expanded to 15-20 different languages

The Semantic Priming Project

- Priming occurs when cognitive processing is speeded because of a previous event
- Generally, we measure priming using lexical decision and naming tasks
- Let's say you have two trials:
  - DOCTOR –> TREE (unrelated)
  - DOCTOR –> NURSE (related)

The Semantic Priming Project

- Priming is thought to occur by several different mechanisms: spreading activation, deliberate cognitive processes such as expectancy generation, etc.



The Semantic Priming Project

- Contains lexical decision and naming response latencies for related, unrelated, and nonword trials
- Is paired with the ELP and SUBTLEX projects
- Gives us more data to predict either response latencies or priming latencies

Regression

- Simple regression is the relationship between one independent and one dependent variable (also correlation)
- Multiple regression is the relationship between two or more independent variables and one dependent variable
  - Useful because it allows you to examine the predictive ability of each variable adjusting for the other variables
- We can fit parametric (linear) models or nonparametric models, depending on the expectation of linearity, as well as the type of dependent variable

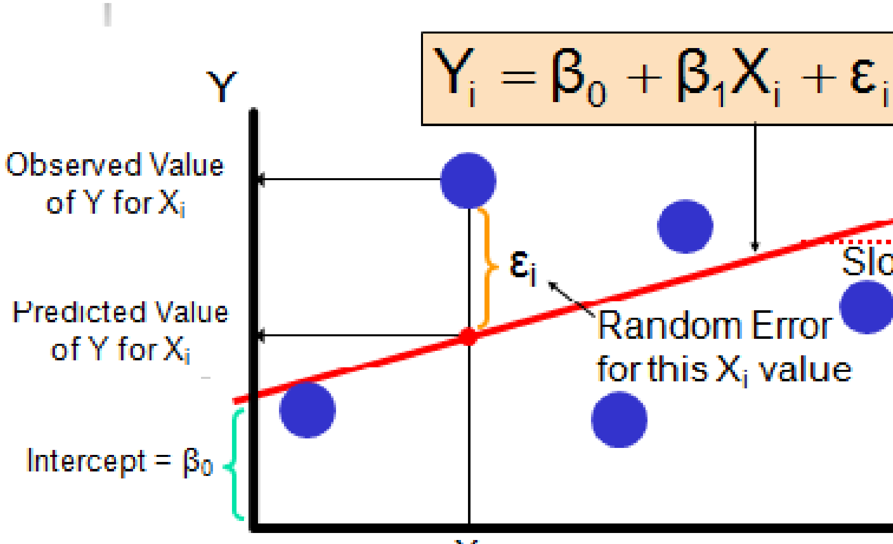
Understand Regression Models

$$\hat{y}_i = b_0 + b_1 x_{1i} + b_2 x_{2i} ... + \epsilon_i$$

- Y-hat is the predicted score for each person (i) on the dependent variable
- B-zero is the y-intercept
- B-1+ are the slope values for each predictor
  - Slopes are interpreted as *for every one unit increase in X, we see B unit increases in Y*
- X is each individual predictor
- Error for each individual person, as we never get their predicted score exactly right

Understand Regression Models

- Least Squares estimation
  - Creates the line of best fit by minimizing the residual error ϵ



Understand Regression Models

- For the overall model including all variables:
  - Determine statistical significance by using p values from an F-test for linear models
  - Determine practical significance by using R^2
- For the individual predictors:
  - Determine statistical significance by using p values from a t-test
  - Determine practical significance by using partial correlation pr^2

Examples Using ELP

- Word is the word presented to the participant
- Length is the number of characters in each word
- SUBTLWF is the subtitle word frequency estimate
- POS is part of speech
- Mean_RT is the mean response latency in milliseconds

```r
library(Rling)
data(ELP)
head(ELP)
```

```
##            Word Length SUBTLWF POS Mean_RT
## 1      rackets      7    0.96  NN  790.87
## 2 stepmother     10    4.24  NN  692.55
## 3 delineated     10    0.04  VB  960.45
## 4   swimmers      8    1.49  NN  771.13
## 5     umpire      6    1.06  NN  882.50
## 6      cobra      5    3.33  NN  645.85
```

Dealing with Categorical Predictors

- How can we interpret and use categorical predictors?
- When X is continuous, the interpretation is that *for every one unit increase in X, we see B unit increases in Y*
- That doesn't work as well for categorical predictors...
- Instead, we have to use Dummy Coding (well, *R* does it for us)

Dummy Coding

- A way to represent categorical data for regression/least squares analyses
- You will get (categories - 1) predictors
- How to interpret these predictors?
  - Each predictor represents the difference in means between the coded group (the one with the 1) and the group coded as all zeroes (the "control" group)

	Dummy V
No Affiliation	
Indie Kid	

Dummy Coding

- POS is a categorical predictor we want to use
- Three categories:
  - JJ: Adjective
  - NN: Noun
  - VB: Verb
- Default is to make the first category the comparison category

```
table(ELP$POS)
```

```
##
##  JJ  NN  VB
## 159 532 189
```

Dummy Coding

- Generally, nouns would be considered the comparison group, so let's rearrange them so they are "first".

```
ELP$POS = factor(ELP$POS, #the column you want to update
                 #the values in the data in the order you w
                 levels = c("NN", "JJ", "VB"),
                 #give them better labels if you want
                 labels = c("Noun", "Adjective", "Verb"))
table(ELP$POS)
```

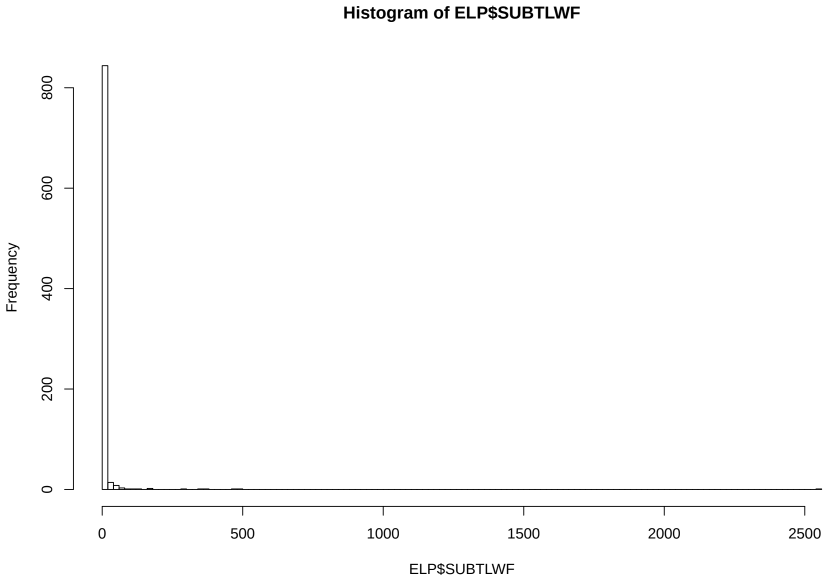
```
##
##      Noun Adjective      Verb
##       532       159       189
```

Dealing with Non-Normal Data

- One issue in language research is that often we have non-normal data
- Especially when working with frequency (as it is distributed by Zipf's law)

Dealing with Non-Normal Data

```r
hist(ELP$SUBTLWF, breaks = 100)
```

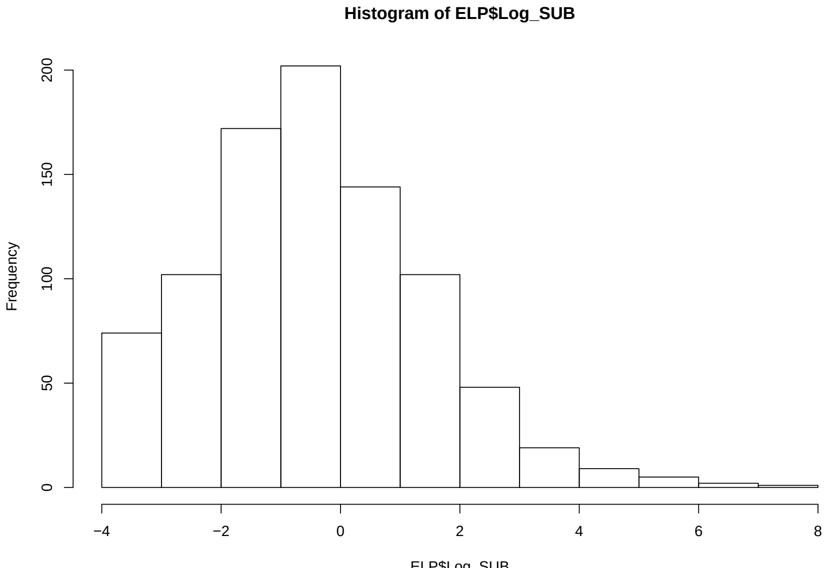


Dealing with Non-Normal Data

- The simplest solution is to take the `log` of the variable.
- Does make interpretation a bit more difficult, but helps with the distribution of the data.

Dealing with Non-Normal Data

```r
ELP$Log_SUB = log(ELP$SUBTLWF)
hist(ELP$Log_SUB)
```



Build the Linear Model

- To be able to use our output for several purposes, we want to save it
  - You can call it whatever you want, I like `model`.
- Format for `lm` function is:
  - Y ~ X + X + ...
  - data = name of data frame

```
model = lm(Mean_RT ~ Length + Log_SUB + POS,
           data = ELP)
```

Summarize the Linear Model

```r
summary(model)
```

```
##
## Call:
## lm(formula = Mean_RT ~ Length + Log_SUB + POS, data = E
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -213.70   -62.55    -9.71    53.87   389.00
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   616.351     12.233  50.385  < 2e-16 ***
## Length         19.555      1.433  13.645  < 2e-16 ***
## Log_SUB       -29.288      1.784 -16.420  < 2e-16 ***
## POSAdjective    6.115      8.506   0.719  0.47238
## POSVerb       -23.069      7.918  -2.913  0.00367 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.
```

Residuals

- A summary of the residuals - remember that residuals are the error terms or how far off we were at predicting the Mean_RT
- We will use this information as part of our assumptions diagnostics for data screening

```
summary(model$residuals)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -213.699  -62.551   -9.714    0.000   53.874  388.998
```

Coefficients

- The coefficients table shows you the individual predictor significance levels
- If you use $p < .05$ as a criterion, we see that:
  - Intercept is the average RL
  - Length is a positive predictor: long words take longer to react to
  - Frequency is a negative predictor: more frequent words are faster (i.e. low freq = high RL)
  - Adjectives and Nouns have the same RL
  - Verbs and Nouns have different RL

Coefficients

```r
options(scipen = 999)
round(summary(model)$coefficients, 3)
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     616.351     12.233  50.385    0.000
## Length           19.555      1.433  13.645    0.000
## Log_SUB         -29.288      1.784 -16.420    0.000
## POSAdjective      6.115      8.506   0.719    0.472
## POSVerb         -23.069      7.918  -2.913    0.004
```

Coefficients

- To interpret categorical predictors, it can help to make a means table
- Now I can see that verbs are responded to faster than nouns, interpreting the categorical predictor

```r
tapply(ELP$Mean_RT, #dv
       ELP$POS, #iv group variable
       mean) #function
```

```
##      Noun Adjective      Verb
##  787.5959  822.9145  754.3316
```

Coefficient Confidence Intervals

- We can calculate the confidence intervals for the coefficients, to help understand their precision

```
confint(model)
```

```
##                   2.5 %    97.5 %
## (Intercept)  592.34193 640.36007
## Length        16.74194  22.36757
## Log_SUB      -32.78872 -25.78704
## POSAdjective -10.57915  22.80935
## POSVerb      -38.61021  -7.52737
```

Coefficient Practical Importance

- Calculate pr^2: variance accounted for in the DV by that IV after removing all variance due to other IVs

$$\frac{t_{x_i}}{\sqrt{t_{x_i}^2 + df_{res}}}$$

```r
t = summary(model)$coefficients[-1 , 3]
pr = t / sqrt(t^2 + model$df.residual)
pr^2
```

```
##       Length      Log_SUB POSAdjective      POSVerb
## 0.1754422571 0.2355448602 0.0005903474 0.0096065265
```

Overall Model

- So how much better than a random guess are we at predicting?
  - A good random guess is always the y-intercept or the mean of y.
- The F-statistic represents the difference of the model from zero

```r
summary(model)$fstatistic
```

```
##     value     numdf     dendf
## 183.7024   4.0000 875.0000
```

Overall Model

- R^2 represents the overlap in all IV variance with the DV variance

```
summary(model)$r.squared
```

```
## [1] 0.4564574
```

Diagnostic Tests

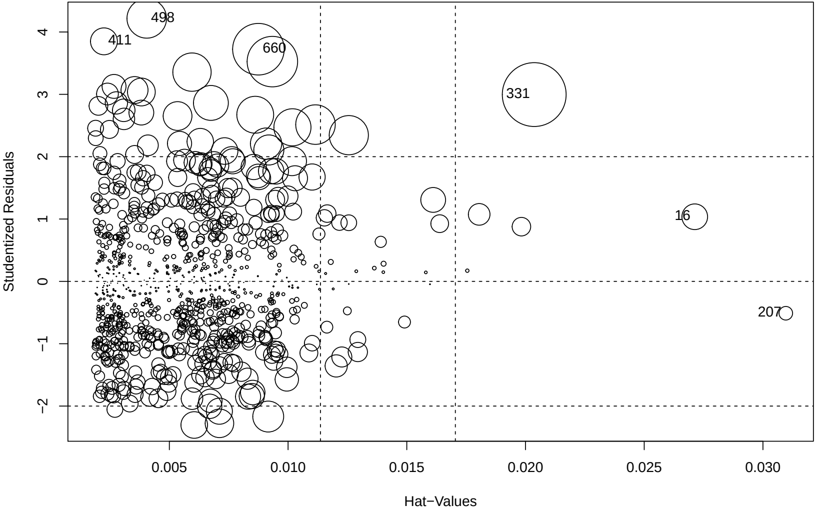
- Outliers and influential observations: data points that have large residuals or are otherwise odd in relation to the rest of the data
- Assumptions of parametric regression:
  - Independence
  - DV is response scale
  - Additivity (no multicollinearity)
  - Linearity
  - Normality
  - Homoscedasticity/Homogeneity

Outliers

- Hat values (or leverage): indicates how much influence on the slope a data point has
- Studentized residuals: the normalized (z-scored) difference between a participant's predicted and actual score
- Cook's values: a measure of influence (both leverage and discrepancy)

Outliers

```
library(car)
influencePlot(model)
```




```
##          StudRes          Hat        CookD
```

Outliers

- What do we do with them?

```
ELP[c(331,660,498,411), ]
```

```
##                     Word Length SUBTLWF       POS Mean_RT
## 331 interdepartmental     17    0.04 Adjective 1324.57
## 660      sacrilegious     12    0.39 Adjective 1228.06
## 498           whippet      7    0.10      Noun 1209.67
## 411         archenemy      9    0.25      Noun 1188.91
```

Assumptions

- Independence - the data is independent from each other (i.e. each data point is from a different "person")
- Interval scale dependent variable: check!

Additivity

- No correlation between predictors above .9 (but .7 is actually not good either)

```
summary(model, correlation = T)$correlation[ , -1]
```

```
##                  Length      Log_SUB POSAdjective
## (Intercept)  -0.94238493 -0.272940500  -0.09904041 -0.23
## Length        1.00000000  0.340416664  -0.05527619  0.06
## Log_SUB       0.34041666  1.000000000   0.09363100  0.00
## POSAdjective -0.05527619  0.093631001   1.00000000  0.23
## POSVerb       0.06668845  0.006852748   0.23717736  1.00
```

Additivity

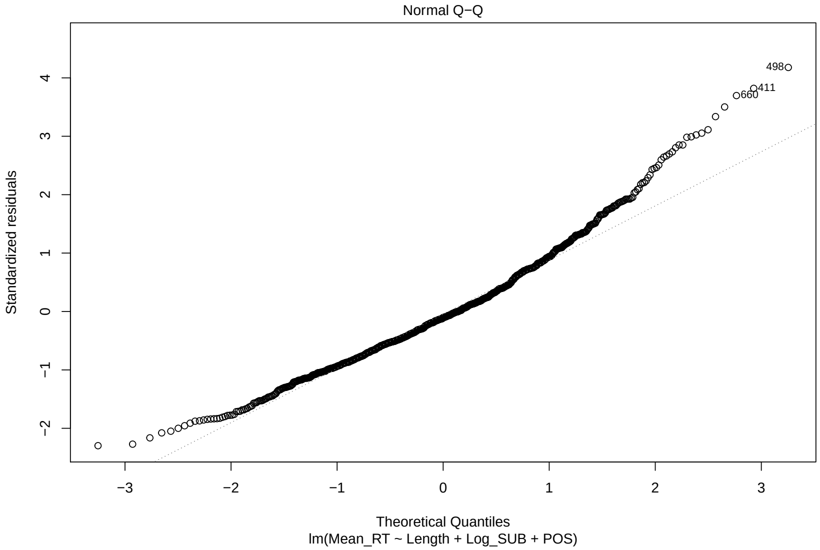
- Small Variance Inflation Scores (VIF) values (less than 5 to 10)

```
vif(model)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## Length  1.151054  1        1.072872
## Log_SUB 1.150140  1        1.072446
## POS     1.026925  2        1.006664
```

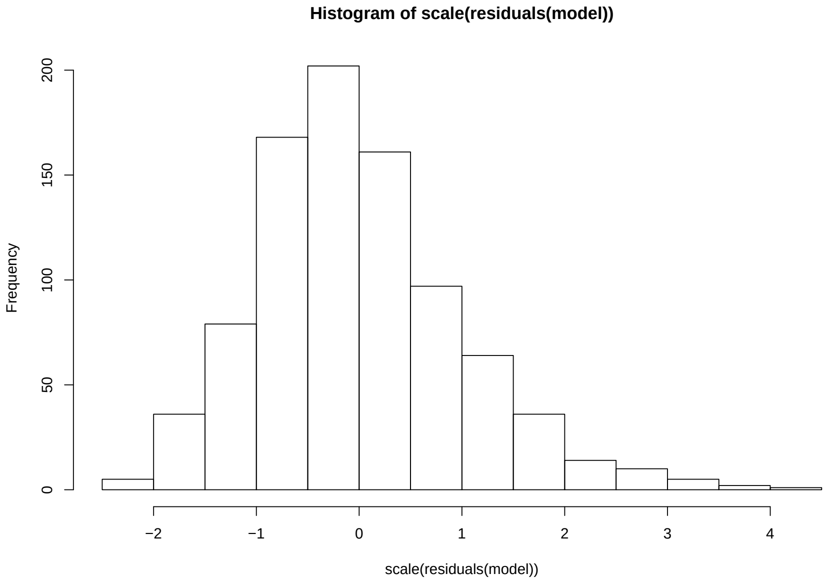
Linearity

```r
plot(model, which = 2)
```



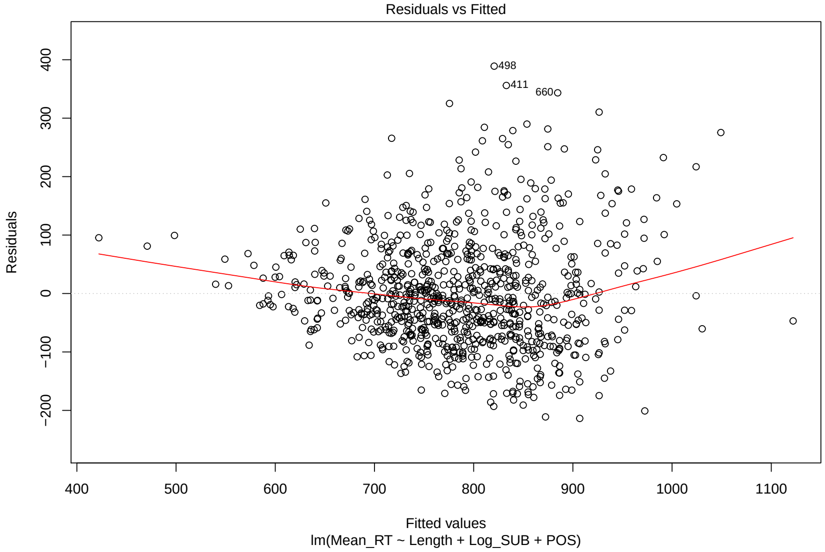
Normality

```r
hist(scale(residuals(model)))
```



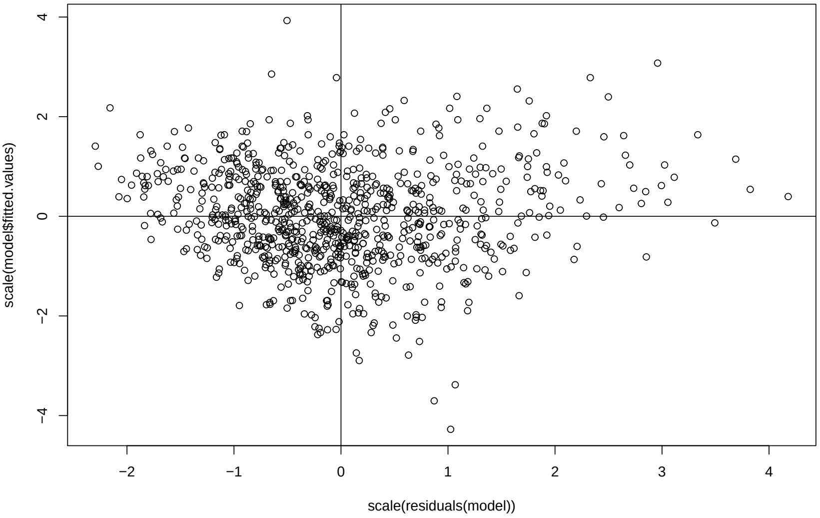
Homoscedasticity/Homogeneity

```
plot(model, which = 1)
```



Homoscedasticity/Homogeneity

```
{plot(scale(residuals(model)), scale(model$fitted.values))
  abline(v = 0, h = 0)}
```



One Solution to Bad Assumptions

- First, build a function that saves the numbers you want:

```
bootcoef = function(formula, data, indices){
  d = data[indices, ] #randomize the data by row
  model = lm(formula, data = d) #run our model
  return(coef(model)) #give back coefficients
}
```

Bootstrapping

- Next, use the boot library to run the bootstraps (lots of runs on randomly sampled data)

```
library(boot)
model.boot = boot(formula = Mean_RT ~ Length + Log_SUB + P(
                  data = ELP,
                  statistic = bootcoef,
                  R = 1000)
```

Bootstrapping

```
model.boot
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = ELP, statistic = bootcoef, R = 1000, formula
##     Length + Log_SUB + POS)
##
##
## Bootstrap Statistics :
##       original        bias    std. error
## t1* 616.350998  0.048541832   12.029762
## t2*  19.554757 -0.002752349    1.455837
## t3* -29.287879 -0.024054074    1.706049
## t4*   6.115101 -0.110164071    9.098897
## t5* -23.068792 -0.285394053    7.237738
```

CIs for Bootstrapped Estimates

```r
boot.ci(model.boot, index = 2)
```

```
## Warning in boot.ci(model.boot, index = 2): bootstrap var
## studentized intervals

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = model.boot, index = 2)
##
## Intervals :
## Level      Normal              Basic
## 95%   (16.70, 22.41 )   (16.63, 22.35 )
##
## Level     Percentile            BCa
## 95%   (16.76, 22.48 )   (16.80, 22.60 )
## Calculations and Intervals on Original Scale
```