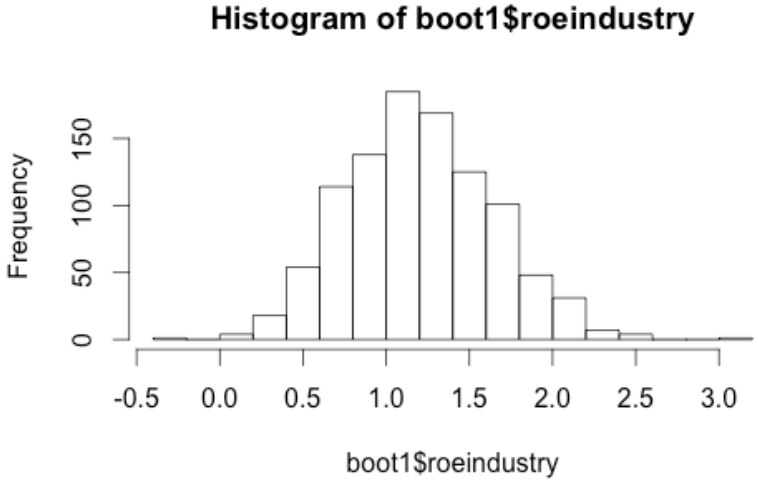
March 19, 2014
STA 371H
Scribing Notes

1. Ad in the Economist

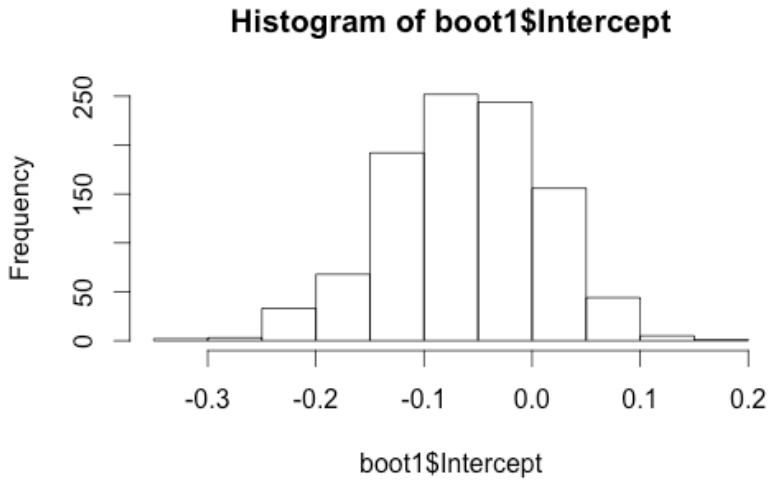
- Oracle Ad said: "SAP Customers are 20% less profitable than their industry peers"
- Oracle would want this to be true about SAP because they are big competitors.

In class, we explored the following three questions:

- Can you replicate the conclusion data?
  - The way this ad came up with the 20% statistic was taking the difference of the mean ROE of the firms (12.6%) and the mean ROE of the industries (15.7%), and dividing it by (15.7%).
  - The ad is basically saying that one average is 20% lower than the other average. While the information is not false, it is very misleading. It would have been more reasonable to say SAP customers are 3.1% (15.7% - 12.6%) less profitable than their industry peers.
  - This doesn't seem like a very straightforward nor correct way to summarize what is happening here.
- Do you think that it is a reasonable conclusion?
  - Thinking in an absolute sense, not relatively, we say that on average companies listed on SAP, were 3.1% less profitable for that one set of data. To see if the conclusion seemed reasonable, we performed a bootstrap test, to test if this 3.1% difference could have arisen due to chance.
  - Taking our null hypothesis into account (that there is no difference between SAP firms and their industry peers), if we were to plot SAP firm ROE (y-axis) against the industry ROE (x-axis), we would expect to see a positive-linear correlation with a slope of +1 and an intercept of 0. We expect to see this because if the SAP firms were random (no one is purposefully better than another) we would expect them to perform the same as the industry on average.
  - Our bootstrapped test looked like:
    - `boot1= do(1000)*lm(roefirm~roeindustry, data=resample(sap))`
  - To get a histogram we:
    - `hist(boot1$roeindustry, 15)`
    - `hist(boot1$Intercept,15)`
  - From the bootstrap test, we concluded that it is reasonable to say that there is a positive-linear relationship between the ROE of the industry and the company, and that a slope of 1 is very plausible. The histogram looked like this:

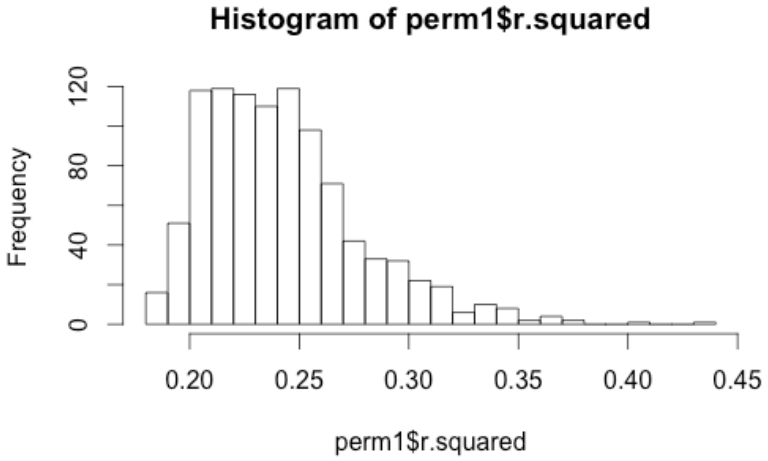


- The histogram for the Intercept looked like:




  - We also found strong evidence that the Intercept lied anywhere between -0.2 and 0.1. Again, we conclude that an intercept of 0 is probable for these samples (you can't reject the null).
  - Therefore, from bootstrapping, we conclude that the Oracle ad claim cannot be proven definitively from this data, and the difference could have arisen due to chance.

- What about their claim on different software? Are there systematic differences dependent on the type of software used?
  - Our null hypothesis was that solution/software was unrelated to firm ROE.
  - To investigate further the claim the study makes on the different software used, we performed a permutation test, shuffling the software used by any given company. This way, we could break any sort of association/correlation between the software used and the firm ROE, and know that the solutions are random.
  - The perm test looked like:
    - perm1 = do(1000)*lm(roefirm~roeindustry + shuffle(solution), data=sap)
  - We then looked at a histogram of this data:
    - hist(perm1$r.squared, 25)




  - We took our rejection region to be anything higher than 0.30. So, if we got anything for R-Squared higher than 0.30, we would say that the difference could not have arisen due to chance, but rather due to some systematic difference. However, for our data, we got an R-Squared of 0.20 (see below), so we can say that any difference seen in the data was largely due to chance.

```
Call:
lm(formula = roefirm ~ roeindustry + solution, data = sap)

Residuals:
     Min       1Q   Median       3Q      Max
-0.90087 -0.07188  0.01347  0.09322  0.66271

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.06625    0.06508  -1.018    0.312
roeindustry     1.32734    0.32144   4.129 9.56e-05 ***
solutionERP    -0.01364    0.10256  -0.133    0.895
solutionMobile  0.04794    0.14757   0.325    0.746
solutionPLM     0.06443    0.08980   0.717    0.475
solutionSAM    -0.04865    0.12919  -0.377    0.708
solutionSCM    -0.05764    0.07138  -0.807    0.422
solutionSRM    -0.04891    0.09750  -0.502    0.617
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2396 on 73 degrees of freedom
Multiple R-squared:  0.2042,    Adjusted R-squared:  0.1279
F-statistic: 2.677 on 7 and 73 DF,  p-value: 0.01596
```

- In the end, we can say that in light of the data, the Oracle Ad did not prove its case.

2. Model Choice
   - This is the answer the answer to the question, "Which predictors should I include in the model?"
   - There is no universally applicable way to think about this question. We only have important criteria/principles to think about when we attempt to answer this question. The principles will be subtly different depending on the type of problem. Fore example:
     - Case 1 is a focused question about some particular effect. Case 1 would be similar to our homework, using course instructor surveys to see if attractive teacher are given higher evaluations, holding all else equal. There are all sorts of possible confounders, including if professors are native English speakers, on a tenure/tenure track, their age, etc. The kind of reasoning that you have to deal with is how does accounting for confounders in slightly different ways affect my answer?
     - Case 2 dealt with pure prediction, which can be considered polar opposite from Case 1. We learned case 2 before case 1, because there are some tools and ideas to be learned from pure prediction problems that are useful for case 1.
   - Model Choice: Pure Prediction Problem
     - What to think about
       - y: variable to be predicted (outcome)
       - Possible predictors (p): $x_1$, $x_2$, $x_3$,……,$x_p$
       - The task is to find the "best" model for predicting y vs. some subset of $x_1$,……,$x_p$
       - Why not use them all? You would worry about over fitting
       - **A good model has to have the right balance of fit and simplicity**
       - There is a tradeoff between fit and simplicity
       - Usually, as the fit is better, complexity increases.
       - You want to look for an "optimal" tradeoff
     - Occam's Razor
       - This concept deals with making the model as complex as it <u>needs</u> to be, and no more!
       - The "razor" refers to cutting or shaving off the deadweight parameters, they are extra complexity but don't provide extra explanatory data.
       - Simplicity is optimal, because you don't want to have to evaluate every parameter if it doesn't even contribute to the model.
       - In regression, "Occam's Razor" is any function of the model output that balances fit and simplicity
       - There are 3 main methods to doing this:

- Adjusted R-Squared
  - $R^2 = 1$- UV/TV, where UV = sum of squared residuals, and TV = total number of observations
  - Adjusted $R^2 = 1 – (UV/TV)((n-1)/(n-p-1))$
  - p = number of parameters to estimate
  - Adjusted $R^2$ takes into account fit and simplicity, because it punishes models that have more parameters
    - When the number of parameters increases, UV goes down, but p goes up
- BIC
  - BIC was not discussed in class. Page 166 of the course packet provides some information about it. Consider this important to "statistical literacy." It might be helpful in your future to be able to recognize the term.
- AIC
  - $AIC = 2p + UV/(\hat{O}^2_e)$
  - The 2p, penalizes the model for complexity, and the $UV/(\hat{O}^2_e)$ penalizes the model for poor fit.
  - We would be looking for the model with the **smallest** AIC.

- GoogleFlu
  - A shortcut we learned in this is that by putting a period (.) in R, we can put all the parameters in the model.
  - `lmfull = lm(cdcflu ~ . - week, data=googleflu)`
  - For GoogleFlu, we were worried that by putting all the parameters in the model, we would be over fitting due to some redundancy in the model if some of the parameters were highly correlated.
  - However, it would take too long to manually search and see which predictors would be best to use.
  - By using the Greedy Backwards Selection (the step function), we started with the full model and then pruned the unnecessary parameters. It automatically defaults to using AIC. It's "greedy" because it only includes the best parameters in the model that are helpful to us. It's "backwards" because it starts with all the parameters and dumps the least helpful ones until dropping the next predictor no longer improves AIC.
  - `lmstep = step(lmfull, direction='both')`
  - In this way, we started with 100 parameters and saw which deletions would yield us the best AIC.
  - You should apply Occam's Razor when you can't possibly manually get rid of excess parameters.