Notes for Monday, February 3

Sampling Distribution

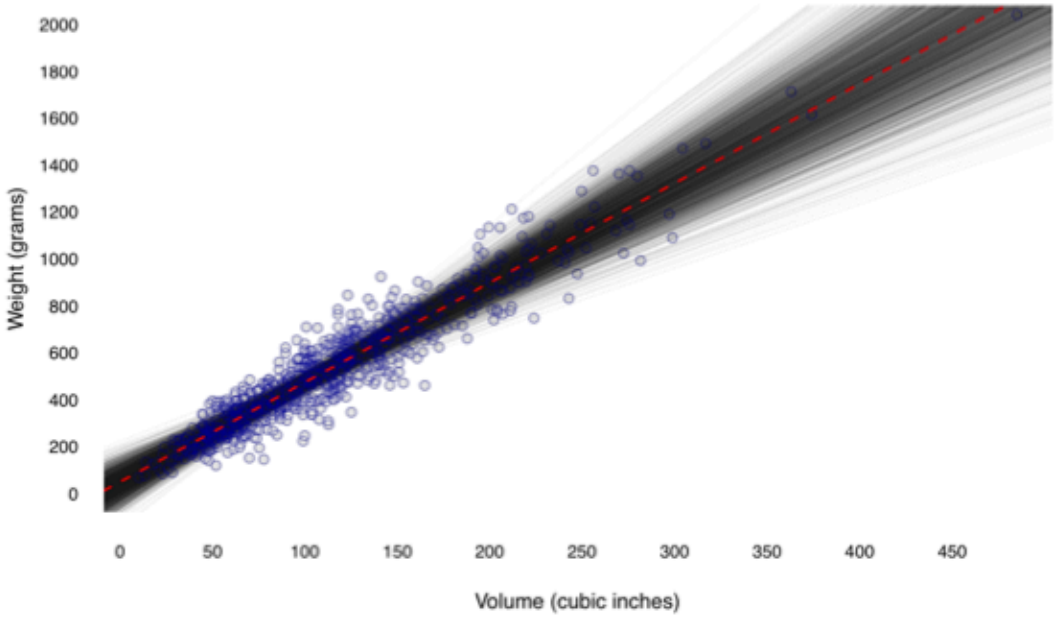
Main question: How sure can we be sure that $\hat{\beta}_0$ is close to β_0 (the true value) and $\hat{\beta}_1$ is close to β_1 (the true slope)?

- Can we quantify our uncertainty associated with these estimates?
- Can we provide more than a point estimate but an interval estimate (aka confidence interval)?

Fish Exercise

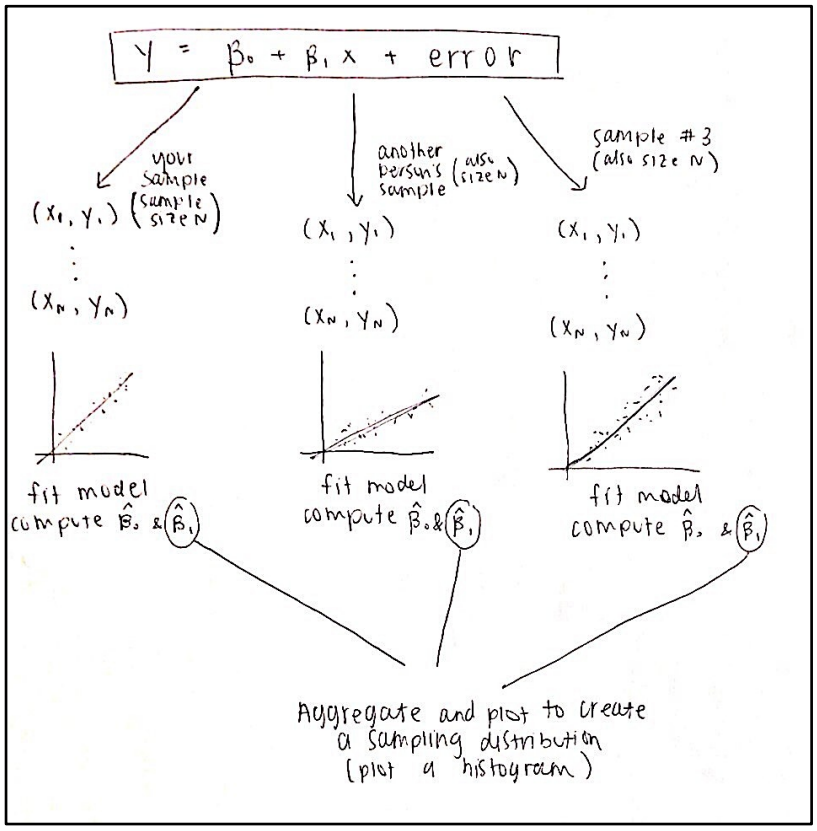
1. Get in pairs of two, and each pair fish 10 "fish" from a bucket.
2. Gather the data printed on the fish – weight, length, height, and width – and record in Excel. Save as a .csv file.
3. In R, find the relationship between volume (predictor) and weight (response).
4. Compare intercept and volume results with the rest of the pairs.

We see that the results of each pair varied quite largely. The amount that they vary is the *uncertainty*. Aggregating the different results creates a better estimate of the data, and other groups' results accounts for the uncertainty of our own results.



This graph is the result of aggregating varying estimates of β_0 and β_1 and shows the sampling distribution of a least-squares fit of fish weight based on fish volume.

Consider the picture to the right:



All procedures between the three "boats" and their samples are the same, and each computes his/her own $\hat{\beta}_0$ and $\hat{\beta}_1$. We then take the $\hat{\beta}_1$ of all three boats and plot them in a histogram, thus aggregating the data and creating a *sampling distribution*[1].

The distribution for this data shows that they are different. The tighter the distribution, the more certain you can be about your particular answer.

To determine how trustworthy your estimate is, think about how different your answer would be with 10 different fish, or 10 different data points. With those 10 different points, if you would have gotten very different answers, then you can't trust your own answer. If those 10 different points, and other repeated trials, will give very similar answers, then your first estimate is trustworthy.

Aside from a histogram, you can also summarize data in other ways, such as using the mean or a coverage interval. People also like to use the standard deviation. By computing the standard deviation of a distribution, you get the *standard error*[2].
Sampling distribution and standard error are two key concepts to know and understand

When we plotted the intercept and volume from the different sub-samples of the class' data, we can get a sense of the data's variability. The key insight is that *the amount the data varies is the certainty you can have about your own estimate*.

What we want to know: How does my answer change under repeated estimates?

- If results from repeated trials changes a lot, your answer is not trustworthy
- If results from repeated trials doesn't really change, your answer is trustworthy
- Summarize your answer with coverage intervals (confidence interval) and standard deviations (standard error)
- Caveat: however trustworthy your data may be, you will never know absolutely about the entire population

[1] *Sampling distribution*: how the estimates for β_0 and β_1 change from sample to sample; anything you can compute from a data set (standard deviation, mean, R^2, residual standard deviation) has a sampling distribution
[2] *Standard error*: the standard deviation of a sampling distribution; this value shows the spread of the estimates

Certainty in the confidence of a model parameter can be equated with stability under repeated samples of a population.

- Example: Consider a witness being questioned by police
    - The policeman asks, "Where were you at 6PM on Wednesday night?" and the guy being questioned response with, "I was eating dinner with friends." If the next night the police returns and asks him the same question and he says, "I was hanging out and playing racquetball," this would make him untrustworthy because he provides unstable answers under repeated questioning.
    - If another person had the same story each time, he is more trustworthy.

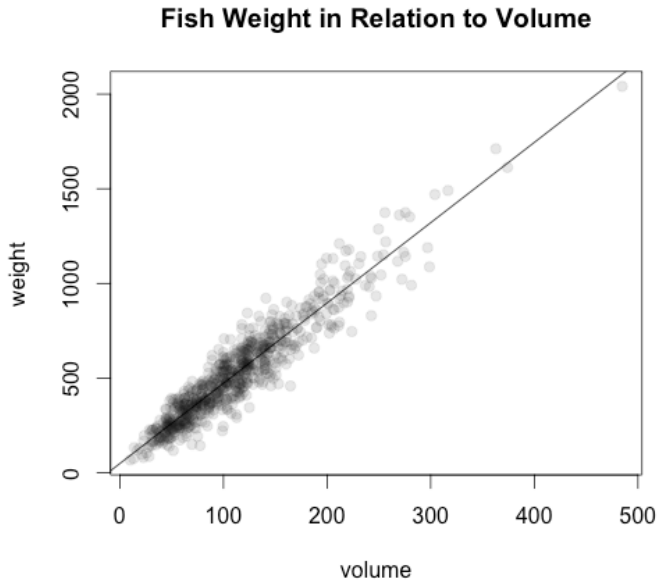
This means that estimates are more trustworthy if they are stable under repeated trials.

If we want to narrow in and look at the sampling distribution for one variable, i.e. $\hat{\beta}_1$, we can look at a histogram.

gonefishing.R and gonefishing.csv scripts

We can replicate the above process of repeating estimation trials more rapidly and efficiently by using software.

- Plot the data



- Take a sample of size 10 from the population and fit a linear model to that sample
    - First define the sample size
        - nsamp = 10
    - Try taking the sample a few different times
        - sample(gonefishing, nsamp)
        - This gives you a different sample of n each time and replicates different samples of n so you can see your uncertainty
    - Find the coefficients of each data set
        - lmsamp = lm(weight~volume, data=sample(gonefishing,nsamp))
        - coef(lmsamp)
- We can automate the process of taking multiple samples

    - Different number of sets of estimates for regression models of each model parameter
    - do(1000)*lm(weight~volume, data=sample(gonefishing,nsamp))
- Look at the histograms of the sampling distributions of the multiple samples
    - montecarlo = do(1000)*lm(weight~volume, data=sample(gonefishing,30))
    - hist(montecarlo$volume)
    - hist(montecarlo$Intercept)
    - This gives you the *confidence interval*[3]
- Compute the standard error of the slop estimate
    - sd(montecarlo$volume)
- Check that the estimator looks unbiased
    - colMeans(montecarlo)
    - From this, you get the mean estimates of the data.
    - Check with coef(lmfull) and you see that the volume is relatively the same. Tis means that, on average, the sampling distribution is centered around the truth
    - The average answer is right, and this means that the data is unbiased.
- Extract a 95% coverage interval for each model parameter
    - confint(montecarlo, level=0.95)
    - This will give you a confidence interval for each variable

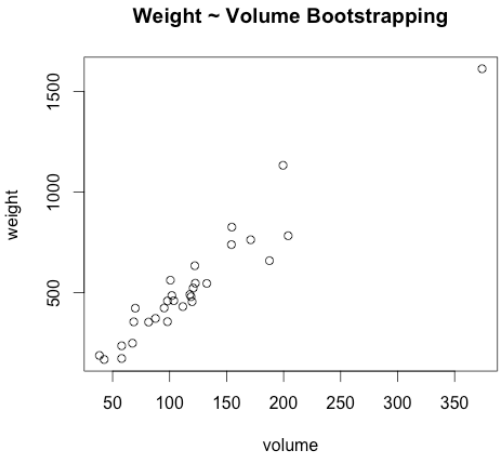
Bootstrapping

Bootstrapping is basically "reaching heights from low circumstances" and getting something from nothing. The fundamental change we make when we talk about bootstrapping: *wherever population is said, insert the phrase "my sample"*

- Example: I want to see how the estimates of the slope vary from one sample to the next when I use samples of size 30 from "my sample."
- No more making references back to the population

Bootstrap in R

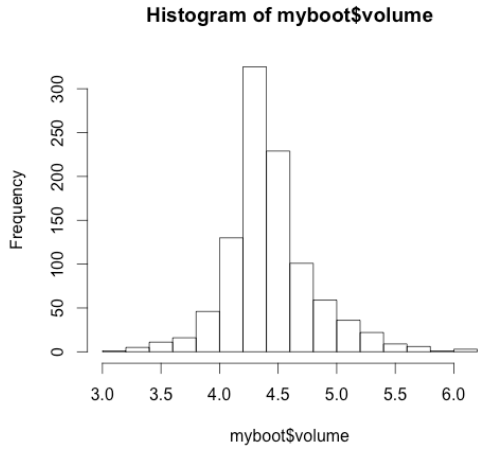
- First, get a sample of size x (x=30)
    - myfishingtrip = sample(gonefishing,30)
    - myfishingtrip is now "my sample"
    - Must sample with replacement[4]
- Create the model, plot, and find equation
    - lmmytrip = lm(weight~volume, data=myfishingtrip)
    - plot(weight~volume, data=myfishingtrip)



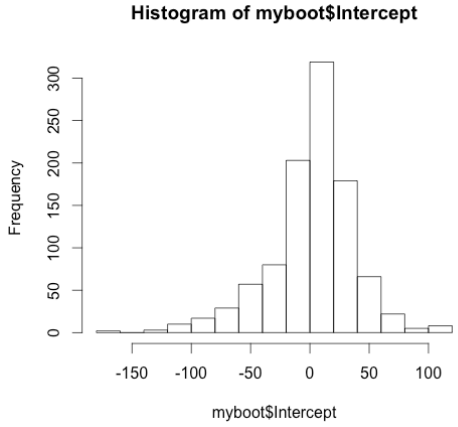
[3] *Confidence intervals* are the coverage intervals of sampling distributions. Coverage interval that covers x% of a set of data points

[4] Sampling with replacement won't give you the same data points each time, and will give you some cases that will never appear at all and some cases that will appear multiple times. Every new resample will have a unique pattern of ties and omissions that will replicate the process of sampling from the population. Resampling in some sense induces variability that is close to the variability that we care about estimating.

    - coef(lmmytrip)
- Try a single bootstrapped sample from your sample
    - lmboot = lm(weight~volume, data=resample(myfishingtrip))
    - coef(lmboot)
- You can also do 10 or even 1,000 bootstrapped samples at once
    - do(10)*lm(weight~volume, data=resample(myfishingtrip))
    - do(1000)*lm(weight~volume, data=resample(myfishingtrip))
    - All have different intercepts and slopes
- Create histograms of the volumes and intercepts
    - hist(myboot$volume)



    - hist(myboot$Intercept)



This is not the true sampling distribution but an estimate. The reason that we do bootstrapping is because we hope that this distribution is about as spread out as the true distribution. If so, then we can estimate the standard error accurately.

- Calculating actual standard deviation of sampling distribution
    - sd(montecarlo$volume)
- Calculating estimate of the standard deviation
    - sd(myboot$volume)