Class 3—Political Science 531
The Next Most Basic Statistical Inference — The Confidence Interval

Jake Bowers

February 17, 2013

Confidence Intervals via Inversion of a Hypothesis Test: Continuing Random Assignment Justified Statistical Inference

Recall our p-value for the sharp null-hypothesis of no effects calculated from the Newspapers field experiment: it was something like $p = .4$ (with some variation depending on whether we used least-squares to calculate a mean-difference, or used median-difference or rank sum). We *interpreted* this p-value as indicating that our observed value of 1.5 percentage points of turnout would be a plausible value if the null of "absolutely no effects", $H_0 : y_{1i} = y_{0i}$, were true.

```
news.df<-read.csv("http://jakebowers.org/PS531Data/news.df.csv")
##this next loads helpful functions into your R session
source("http://jakebowers.org/PS531Data/ps531fns.R")
library(mosaic)

news.df$sF<-factor(news.df$s) ##we want to use s as a set of dummies
##for mean differences as a test statistic (so only need slope not constant)
obs.md<-fit.please(r~z+sF,thedata=news.df)["z",] ## if we had unequal blocks we'd need to fixed effects for b
##approximate \Omega by simulation
nsims<-500
dnull.0<-do(nsims)*fit.please(r~shuffle(z,within=news.df$sF)+sF,thedata=news.df)
dimnames(dnull.0)[[2]]<-c("Intercept","z","s2","s3","s4") ##add names
##The approximated/simulated randomization distribution of the test statistic under the sharp null
table(dnull.0[,"z"])/nsims
```

```
   -5  -3.5    -3    -2  -1.5  -0.5     0   0.5   1.5     2     3   3.5     5
0.072 0.066 0.066 0.064 0.120 0.080 0.132 0.080 0.128 0.052 0.050 0.054 0.036
```

```
##The p-value for sharp null of no effect:
mean(dnull.0[,"z"]>=obs.md)
```

```
[1] 0.32
```

But, we weren't really talking about "estimating" a value for our treatment effect. Rather we were assessing the plausibility of a hypothesis about our treatment effect in a kind of funny way by actually summarizing the plausibility of observing what we observed if the null were true: the p-value of $p = .4$ summarized the how surprising it would be observe 1.5 if our null were true.

1. What are the most basic ingredients that you need to produce a p-value? *Hint:* I came up with six ingredients. You might also consider thinking about what we don't need. (1) A hypothesis stated about something that we do not observe. (2) A link between what we do not observe and what we do observe. (3) A summary of what we observe (a test statistic). (4) A stochastic process, a way in which the test statistic might vary. (5) A way to characterize the variation in the test statistic under the null (where "under the null" means "taking the null seriously" or "imagining that the null were correct"). (6) An observed value for the test statistic.

   Notice what we don't need: a large sample size, a probability model of outcomes (a probability model of assignment or sampling will work too), a linear model, a hypothesis about test statistics (i.e. hypotheses do not have to be about means).

2. Last time we saw that we need to take the design of the study into account when we do statistical inference: here the probability distribution defining our p-values changed a lot when we repeated the actual procedure (i.e. paired design versus a completely or simply randomized design).[1]

[1] And notice that if we assigned treatment with coin flips it would generate a slightly different distribution than if we assigned treatment by drawing a fixed number of balls from an urn [because, recall, that coin flips leave open the possibility of assigning more than or fewer than four objects to treatment in the completely assigned case].

What we'd like, however, most of the time, is not just a p-value rejecting a null of no effects, but a range of plausible values. Notice, $H_0 : y_{1i} - y_{0i} = 0$ or $H_0 : y_{1i} = y_{0i} + 0$ doesn't make our observed data that implausible (at least we decided $p \approx .4$ doesn't indicate a lot of doubt to be cast against this null). Notice that writing our null in this way encourages us to consider other values for the treatment effect (let's call it τ). So, we could write $H_0 : y_{1i} = y_{0i} + \tau_i$ as a statement about a hunch we have about how the unobserved targets of inference relate: so far we tested $H_0 : \tau_i = 0$. What if we had a different hypothesis, like $H_0 : \tau = 5$? Notice no i subscript. What would this hypothesis say about the unobserved "?" for potential responses to the control condition in Table 1?

i	Z_i	Y_i	y_{1i}	y_{0i}
Saginaw	0	16	?	16
Sioux City	1	22	22	?
Battle Creek	0	14	?	14
Midland	1	7	7	?
Oxford	0	23	?	23
Lowell	1	27	27	?
Yakima	0	58	?	58
Richland	1	61	61	?

Table 1: Treatment assignment (Z), observed turnout (R), and unobserved potential outcomes (y_1, y_0) for Cities (i) in the Newspapers Experiment.

All of the missing control responses (i.e. the control responses for the treated cities) would be the treated city response minus 5: i.e.$y_{0i} = Z_i y_{1i} - 5 = [16, 17, 14, , 2, 23, 22, 58, 56]$.

3. Your last answer implied that $H_0 : \tau = 5$ is equivalent to $H_0 : y_{1i} = y_{0i} + 5$. Now, recall that hypothesis testing requires that we take the null seriously for the sake of arguing with it. And recall

$$Y_i = Z_i y_{1i} + (1 - Z_i) y_{0i} \tag{1}$$

What would $H_0 : y_{1i} = y_{0i} + 5$ imply about the relationship between what we observe in Y_i and what we'd like to know about in y_{0i}: *Hint:* Plug in for y_{1i}, solve for y_{0i}.

$$\begin{aligned} Y_i &= Z_i y_{1i} + (1 - Z_i) y_{0i} \\ &= Z_i y_{0i} + Z_i 5 + (1 - Z_i) y_{0i} \\ &= Z_i y_{0i} + Z_i 5 + y_{0i} - Z_i y_{0i} \\ &= Z_i 5 + y_{0i} \\ y_{0i} &= Y_i - Z_i 5 \end{aligned}$$

So, if we want to generate a distribution reflecting what would happen in the absence of treatment if the true treatment effect were 5, we'd subtract off 5 from the outcome for each treated unit.

Notice: A null hypothesis is a statement about some counterfactual. A simple or sharp or strong null hypothesis is a statement that completely specifies the counter-factual — that tells us how each and every unit would act in the absence of treatment.

4. Now test this hypothesis: You'll need to *adjust the observed outcomes* to represent the null of $\tau = 5$ as you simulate. Does our observed value provide more or less doubt against the null of $\tau = 5$ than against $\tau = 0$? *Hint:* Try the following function. Remember to compare the observed value of the test statistic (1.5) to this new randomization distribution to get the p-value. This function (or one of your own creation that does the same thing) can be fed to `do()`.

```
##Setup a data frame for use in the simulations
##This is required because fit.please() is not as smart as lm() and requires a dataframe as input
simdat<-data.frame(r=news.df$r,sF=factor(news.df$s))
adj.fit<-function(null,theobsdata=news.df,thesimdat=simdat,strata="s"){
  ##Shuffle the treatment assignment following the real design
  ##and add it to the new dataset
  thesimdat$z<-shuffle(theobsdata$z,within=theobsdata[,strata])
```

```
  ##Adjust outcomes according to the hypothesis about constant, additive effects
  thesimdat$adjR<-theobsdata$r-(thesimdat$z*null)
  ##est mean diff and just return the slope
  fit.please(adjR~z+sF,thedata=thesimdat)["z",]
}


##I find it useful to only have to type in "500" once, and to only change it once.
nsims<-500
##First check the function for our old null of 0
dnull.0.check<-do(nsims)*adj.fit(null=0)
##The p-value for null of no effect:
mean(dnull.0.check>=obs.md)
###Next test \tau=5
##To Illustrate: Adjust the outcomes to reflect the null.
news.df$obs.adjR<-news.df$r-(news.df$z*5)
print(news.df[,c("city","z","r","obs.adjR")])
dnull.5<-do(nsims)*adj.fit(null=5)
##The approximated/simulated randomization distribution of the
##test statistic under the sharp null
table(dnull.5)/nsims
##The p-value for sharp null of no effect:
sum(dnull.5>=obs.md)/nsims
```

1.5 would cast more doubt on a null of 5 than on a null of 0.

5. So, now you've tested two hypotheses, and you are starting to get a sense of what the kinds of hypotheses that might be implausible given our data and design. Let us imagine that we don't need to consider negative treatment effects (for now). Test, a few (5 or 10) null hypotheses of the form $H_0 : y_{1i} = y_{0i} + \tau$ where $\tau \geq 0$, saving their p-values (where the p-values are the proportion of simulated experiments with test statistics greater than or equal to our observed value). *Hint:* Here is some code which might help. Feel free to change it to suit you.

```
nsims<-100
my.favorite.hypotheses<-c(0,1,4,40)
dnulls.list<-sapply(my.favorite.hypotheses,function(h){
  do(nsims)*adj.fit(null=h)
})
sapply(dnulls.list,function(d){sum(d>=obs.md)/length(d)})
```

See next answer. I did it for all integer null hypotheses between -30 and 30.

6. Imagine we decided that we'd reject the null whenever the $p \leq .17$ (i.e that $p \leq .17$ indicates "enough doubt" cast against the null from our observed data). Or, we'd say, "plausible" null hypotheses or "not rejected" null hypotheses, are those with $p > .17$. What is the largest null hypothesis you tested which is inside the boundary formed by this rejection criteria (i.e. $p > .17$)?

First, setup the code to test a bunch of hypotheses

```
##Make a vector of hypotheses
somehyps<-seq(from=-10,to=10,by=.5) ##could also have written -20:20
##Increase the number of simulations
nsims<-1000
##Since it took some time to do these simulations (61 hypotheses * 1000 sims each)
#I saved the output and then each time I run the file I try to load the saved
##simulation results. If it fails to load, then the program re-creates the
##simulations. Thus, if I delete the simulation file I can force the
##program to recreate it.

##Define the function
getpval<-function(null.hyp){
  ##print(null) ##just to keep track
```

```
  ##This function calculates three kinds of p-values. For our exercise
  ##we were only interested in a one-sided p-value. For your purposes
  ##you might want other kinds.

  ##Simulate experiment if null were true
  dnull<-do(nsims)*adj.fit(null=null.hyp)

  thepge<-mean(dnull>=obs.md) ## amount of the null ge observed
  theple<-mean(dnull<=obs.md) ## amount of the null le observed
  theptwosided<-2*min(thepge,theple) ##Most common definition of a
  ##two-sided p-value is twice the minimum of the one-sided p-values
  ##thepabs<-sum(abs(dnull)>=obs.md)/nsims ##another definition of a two-sided p-value

  return(c(h=null.hyp,ple=theple,pge=thepge,ptwo=theptwosided)) ##return the nulls and ps
}
  ##Here I use sapply() rather than for() just to collect the results nicely.
theps<-sapply(somehyps,function(null){getpval(null.hyp=null)})
##save(theps,file="handout3-theps.rda")
##}

print(theps[,somehyps %% 1 == 0]) ##a math trick.
##Which hypothesis is just inside the confidence interval?
the.ul<-max(theps["h",theps["pge",]>.17])
```

Notice that the p-values don't strictly increase or decrease. This is mostly because of (1) the fact that we are shuffling/simulating and partially (2) the outliers and our choice of a mean difference (rather than ranks or medians or something else) as a test statistic.

Here I do the same thing using the enumerated randomization distribution:

```
##This is Omega
Om <- data.frame(V1 = c(1, 0, 1, 0, 1, 0, 1, 0),
                 V2 = c(0, 1, 1, 0, 1, 0, 1, 0),
                 V3 = c(1, 0, 0, 1, 1, 0, 1, 0),
                 V4 = c(0, 1, 0, 1, 1, 0, 1, 0),
                 V5 = c(1, 0, 1, 0, 0, 1, 1, 0),
                 V6 = c(0, 1, 1, 0, 0, 1, 1, 0),
                 V7 = c(1, 0, 0, 1, 0, 1, 1, 0),
                 V8 = c(0, 1, 0, 1, 0, 1, 1, 0),
                 V9 = c(1, 0, 1, 0, 1, 0, 0, 1),
                 V10 = c(0, 1, 1, 0, 1, 0, 0, 1),
                 V11 = c(1, 0, 0, 1, 1, 0, 0, 1),
                 V12 = c(0, 1, 0, 1, 1, 0, 0, 1),
                 V13 = c(1, 0, 1, 0, 0, 1, 0, 1),
                 V14 = c(0, 1, 1, 0, 0, 1, 0, 1),
                 V15 = c(1, 0, 0, 1, 0, 1, 0, 1),
                 V16 = c(0, 1, 0, 1, 0, 1, 0, 1))
row.names(Om)<-row.names(news.df)
adj.fit.2<-function(null,newz,theobsdata=news.df,thesimdat=simdat,strata="s"){
  ##Adjust outcomes according to the hypothesis about constant, additive effects
  adjR<-theobsdata$r-(newz*null)
  ##est mean diff and just return the slope
  coef(lm(adjR~newz))["newz"]
}
dnull.5.Om<-sapply(Om,function(Om.z){
  adj.fit.2(null=5,newz=Om.z)
  })
getpval2<-function(null.hyp){
```

```
  ##print(null) ##just to keep track
  ##This function calculates three kinds of p-values. For our exercise
  ##we were only interested in a one-sided p-value. For your purposes
  ##you might want other kinds.

  ##Simulate experiment if null were true
  dnull<-sapply(Om,function(Om.z){ adj.fit.2(null=null.hyp,newz=Om.z) })

  thepge<-mean(dnull>=obs.md)
  theple<-mean(dnull<=obs.md)
  theptwosided<-2*min(thepge,theple) ##Most common definition of a
  ##two-sided p-value is twice the minimum of the one-sided p-values
  ##thepabs<-sum(abs(dnull)>=obs.md)/nsims ##another definition of a two-sided p-value

  return(c(h=null.hyp,ple=theple,pge=thepge,ptwo=theptwosided)) ##return the nulls and ps
}
  ##Here I use sapply() rather than for() just to collect the results nicely.
theps2<-sapply(somehyps,function(null){getpval2(null.hyp=null)})
```

Here, I just plot the p-values from the one-tailed tests to relate to the question asked. The blue line comes from the enumerated distribution — notice that the p-values go strictly down (or are the same) while the p-values from shuffling bounce around a bit more.

```
par(oma=rep(0,4))
plot(theps["h",theps["h",]>=0],theps["pge",theps["h",]>=0],xlab="Null",ylab="P-value")
lines(theps2["h",theps2["h",]>=0],theps2["pge",theps["h",]>=0],type="b",col="blue")
abline(h=.17) ##our Type-I error rate/confidence level
abline(v=the.ul) ##rough upper boundary.
```

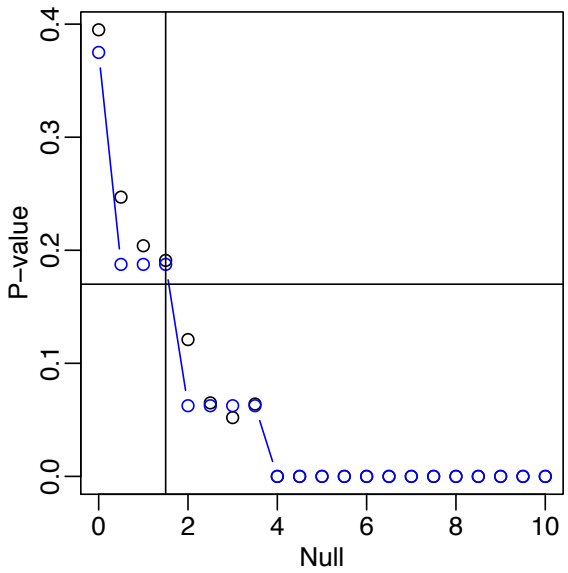


Figure 1: As the null hypotheses become much larger than 0 percentage points difference of turnout, their plausibility diminishes.

Notice: You've just inverted a hypothesis test to create a confidence interval. It is an incomplete, or one-sided interval (such that we don't know about how far it extends below zero). A confidence interval is *defined* as the set of null hypotheses we wouldn't reject: a 66% confidence interval contains those nulls against which the evidence is weaker than $p = .17$ (since (100-66)/2=.17). The upper boundary of our confidence interval is about 1.5 percentage points of turnout. Our observed value of 1.5 and what we know about the design make it pretty implausible that an effect of

more than 1.5 points of turnout would have been caused by this treatment.

The two sided interval would be:

```
range(theps["h",theps["ptwo",]>.17])
```

Now for the weak null

Look back again at Table 1. Let's think about how the weak null hypothesis might be stated and what it might mean.

So, the weak null, in Neyman's formulation (1923) is motivated as follows: we want to know about the causal effect of a treatment, and really, we'd like to know something like $\tau_i = y_{i,1} - y_{i,0}$ — we'd like to know the individual unit level causal effects. But, we don't observe both values [so both Neyman and Fisher start in the same place (and, by the way, my version of Fisher is not Fisher's version of Fisher as of 1935)].

But, Neyman realized that one could think of each set of observed Y_i (one for treated and one for controls) as a sample from a finite population: the population being the experimental pool. So, here, our finite population is of size 8, and we are sampling 1 city to be treated within each of 4 pairs (each pair can be seen as a mini-population of size 2 and sample is of size 1). If we agree with this idea (which we do, since it is a correct description of the procedure), then, by sampling theory we know that our sample difference of means is an unbiased estimator of the population difference means.

What is an unbiased estimator? An unbiased estimator is a procedure which, when repeated over and over on different samples taken the same way from the same population, has, on average, the population value. We will show in a few weeks that this is true, under certain circumstances of the least squares estimator for β, and the short hand for this is $E(\hat{\beta}) = \beta$ [written in words as "the expected value of $\hat{\beta}$ (the estimator of β) is β"]. Now, Neyman proposed that we focus on $\bar{\tau} = (1/n)\sum_{i=1}^{n} \tau_i$ — the average of the individual level counterfactual differences, the average of the individual level treatment effects (unobserved). And he proposed to use the sample means and their differences as a estimator for the "population" (i.e. counterfactual) means: $\hat{\bar{\tau}} = (1/m)\sum_{i=1}^{n} Z_i Y_i - (1/(n-m))\sum_{i=1}^{n}(1-Z_i)Y_i$ as an estimator for $\bar{\tau} = (1/n)\sum_{i=1}^{n} Z_i y_{i,1} - (1/n)\sum_{i=1}^{n}(1-Z_i)y_{i,0} = \bar{y_1} - \bar{y_0}$ where m is number of treated units and n is total sample size.

We can show that $E_Z[\hat{\bar{y_1}}] = \bar{y_1}$ and $E_Z[\hat{\bar{y_0}}] = \bar{y_0}$ and so $E_Z[\hat{\bar{y_1}} - \hat{\bar{y_0}}] = \hat{\bar{y_1}} - \hat{\bar{y_0}}$ (here we just show the second part directly):

$$E_Z[\hat{\bar{\tau}}] = E_Z[\frac{\sum Z_i Y_i}{m} - \frac{\sum(1-Z_i)Y_i}{n-m}] \tag{2}$$

$$= E[\frac{\sum Z_i Y_i}{m}] - E[\frac{\sum(1-Z_i)Y_i}{n-m}] \tag{3}$$

only Z (indicator of which unit is sampled for treatment) is random expectation is only of Z

$$= \frac{\sum E_Z[Z_i]Y_i}{m} - \frac{\sum(1-E_Z[Z_i])Y_i}{n-m} \tag{4}$$

and we know that $E_Z[Z_i] = m/n$ — the average of a 0,1 variable is proportion of 1s

$$= \frac{\sum(m/n)Y_i}{m}] - \frac{\sum(1-(m/n))Y_i}{n-m} \tag{5}$$

$(1-(m/n)) = (n-m)/n$ so

$$= ((m/n)(1/m)\sum Y_i) - ((1/(n-m))((n-m)/n)\sum Y_i) \tag{6}$$

$$= 0 \tag{7}$$

Confidence Intervals using Standard Errors of a Point Estimate: Example of Random Sampling Justified Statistical Inference

So, the first and simplest way to do statistical inference is to repeat the experiment. We repeat the experiment to get a hypothesis test. And we build a confidence interval out of hypothesis tests — after all that is how confidence intervals are defined. We can also get point estimates (called Hodges-Lehmann point estimates) by (very roughly speaking) shrinking the confidence interval [a very tight confidence interval is basically as useful as a point estimate, right?].

Now, there is another way to get confidence intervals — which is nice since manually inverting a hypothesis test is a bit tedious. The downside (or upside) of this method is that the null hypothesis is less well defined: that is, we will be able to say that we cannot reject a null of some average difference of means, but will not be able to say anything about the particular pattern of counterfactuals which led to this difference of means.

This mode of statistical inference is based on how the cases were sampled from some population, not on how some interesting explanatory (or causal) variable was assigned.

A new dataset: About 6 months after the election of the first democratically elected president in Chile since the military coup the Centro de Estudios Publicos did a national in-person survey of of about 1190 people. A question I've had about new democracies is about how much worry about the military lingers, and especially about younger citizens (from whence soldiers/terrorists/resistence fights/social movement activists tend to be recruited). This question is linked to larger concerns about the meaning and shape of democracy given civil societies shaped by authoritarianism and political violence.

In what follows, you will want to run the commands as I have written them so that you can get a sense for what recoding variables is like.

```
##library(foreign) ##just to show how I converted it from SPSS format
##chile90<-read.spss("Data/Encuesta\ CEP\ 14\ May-Jun\ 1990/Encuesta\ CEP\ 14\ May-Jun\ 1990.por",
##                   use.value.labels = FALSE, to.data.frame = TRUE)
##dim(chile90)
##save(chile90,file="chile90.rda")
load(url("http://jakebowers.org/PS531Data/chile90.rda"))
```

Here is some information from the code book.[2].

```
p26 Cual es su edad por favor
'1' 18-24
'2' 25-34
'3' 35-44
'4' 45-54
'5' 55-64
'6' 65YMAS
'7' NO CONTESTA
```

```
 20.  En esta hoja se presentan una serie de riesgos que el país puede enfrentar en los próximos 4 años.
Por favor, lea la lista, y dígame ?Cuáles son para Ud.  los dos principales riesgos que puede enfrentar
el país?  (PASAR TARJETA No 6)
...
6 Conflicto con las Fuerzas Armadas
...
```

```
##Some recoding
##milworry=1 if mentioned worry about conflicts with the military, 0 otherwise
chile90$milworry<-(chile90$P202==6)+(chile90$P201==6)
##I try to check my recodes against the original variables
table(chile90$milworry,chile90$P202==6,chile90$P201==6,useNA="ifany")
library(car) ## contains the recode() command
##Collapse the age variable into just three categories
chile90$age3<-recode(chile90$P26,"1:2='18-34';3:4='35-54';5:6='55ymas'",as.factor.result=TRUE)
##Check the recode
table(chile90$P26,chile90$age3,useNA="ifany")
```

Today, let us work with a simple random sample of 100 people from the larger dataset:

```
set.seed(20100205) ##This makes the resampling/shuffling/random-number generation the same from run to run.
chile90s<-resample(chile90[,c("age3","milworry")],size=100,replace=FALSE)
##Kaplan's resample plays nice with data frames

##This table provides another summary of the relationship between worry about the military and age
```

[2] http://jakebowers.org/PS531Data/CEP14.pdf

```
table(chile90s$age3,chile90s$milworry,useNA="ifany")
(fit1<-fit.please(milworry~age3,thedata=chile90s))
```

1. Interpret the results of the preceding regression. Especially, I wonder whether the oldest group is very different from the youngest group in their worry about the military.

   About 35% of the youngest group reported fears about the military where as 35-16% of the oldest group reported such worries. The oldest group and the youngest group have different amounts of worry about the military. The oldest group have less worry than the youngest group.

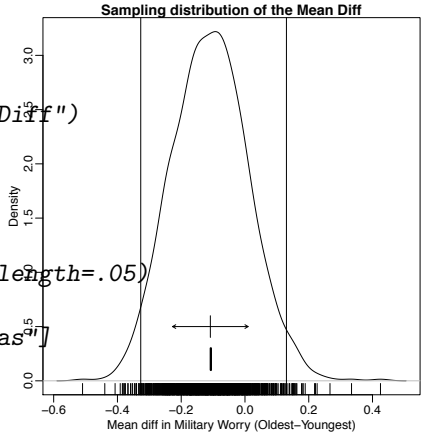
2. Now, we could test the hypothesis that, in our sample, the youngest and the older groups have different amounts of worry about the military [notice, should say, "report to interviewers different amounts of worry" — which is something interesting in and of itself!]. But, let us instead focus on a different target of inference: rather than try to infer from the younger group to the older group, let us try to infer from the sample to the population. In this case, the population is known to us — it is the dataset of 1190 people. So, the question arises, if we drew another sample using exactly the same procedures as gave rise to our observed sample, how far off would the coefficient calculated on that sample be from the coefficient we just calculated? In fact, rather than predict how far off we'd be in another sample, we actually would prefer to know something about the *typical variation* we ought to expect from sample to sample in our coefficient.

   Let us evaluate this directly here, since we actually have the population. Please graph the density of the **sampling distribution** of the statistic describing the difference in worry about the military between the oldest and youngest groups. Notice that I call this a sampling distribution because here, for this class, I am assuming that we *know the population* — here we define the population to be the set of 1190 people in the dataset only to give us some experience with the idea that we sample from a population.*Hint:* The code below provides some hints about how to do this. You'll probably want more than 5 samples from the population (try 100 or 1000). And you can use `plot(density())` as you did last time. I suggest you look at the results of your resampling before plotting to make sure you plot the correct column.

```
n.samps<-100
samp.fits<-do(n.samps)*fit.please(milworry~age3,thedata=resample(chile90[,c("age3","milworry")],size=100
```

```
nsamps<-1000
sampfits<-do(nsamps)*fit.please(milworry~age3,thedata=resample(chile90[,c("age3","milworry")],size=100,re
colnames(sampfits)<-rownames(fit1) ##give the matrix names to make our job easier: this would be automati
str(sampfits) ##summary of the structure of the sampfits objects just to see how it is organized
sampfits[1:10,] ##the first 10 rows --- again just to get a sense of the data
##one could also have done
head(sampfits)
```

```
plot(density(sampfits[,"age355ymas"]),
     xlab="Mean diff in Military Worry (Oldest-Youngest)",
     ylab="Density",main="Sampling distribution of the Mean Diff")
rug(sampfits[,"age355ymas"])
abline(v=quantile(sampfits[,"age355ymas"],c(.025,.975)))
thesd<-sd(sampfits[,"age355ymas"])
the.samp.mean<-mean(sampfits[,"age355ymas"])
arrows(the.samp.mean-thesd,.5,the.samp.mean+thesd,.5,code=3,length=.05)
segments(the.samp.mean,.4,the.samp.mean,.6)
the.pop.mean<-coef(lm(milworry~age3,data=chile90))["age355ymas"]
segments(the.pop.mean,.1,the.pop.mean,.3,lwd=3)
```



Notice that this is the sampling distribution that most of our other techniques would like to approximate: here we *know* the population because we are acting like the big 1100 person survey is the population and we *know* how our sample was drawn because we did it ourselves. So, for example, in other classes you talk about the sampling distribution being Normally- or t-distributed. Here, we don't need to name our sampling distribution — it is what it is (although it turns out that Normal and t distributions may well approximate it.)

We would very rarely actually be able to draw new samples from the population. But, our resampling from our sample (aka the bootstrap procedure) is designed to simulate/approximate what you just did in this exercise.

3. Between what two values do you find 95% slopes after sampling from the population? We'd call this the *95% coverage interval* **not** a confidence interval. (See Kaplan Chap 14 for a nice discussion of this distinction where he says, in §14.4 "A 95% confidence interval is intended to reflect a 95% coverage interval of the sampling distribution, as approximated by the resampling distribution." *Hint:* Try the `quantile()` function on the appropriate column of `samp.fits`

   Plotted as two vertical lines above.

   ```
   quantile(sampfits[,"age355ymas"],c(.025,.975))
   ```

   Why call the target of our approximation a coverage interval? First, recall that a coverage interval is just an interval that "covers" a certain proportion of a variable — so, the middle 95% of the data in a study is about [-0.3269,0.13]. Another summary of a sampling distribution is its standard deviation which is often called its standard error.

   Now, why do we say coverage interval and not confidence interval here? Recall that a confidence interval is defined as a collection of hypothesis tests not-rejected at a certain significance level — that is, the intervals will exclude the true population mean-difference in $100(1-\alpha)$ simulations where $0 < \alpha < 1$, reflecting our toleration for false rejections of the null. As a collection of tests of hypotheses, a confidence interval is a random quantity — it varies from sample to sample. Our population does not vary. By sampling more and more times from the population, we just fill in more and more details about the facts. So, for example, the fact that 0 is inside our confidence interval does not tell us something about a hypothesis test for $H_0 : \beta_{\text{age355ymas}} = 0$, rather it (plus the unimodal nature of the sampling distribution) tells us that the true population value is not far from 0.

   If we calculated an interval using resampling with replacement for each of the samples used to make `sampfits`, then each of those intervals would be a bootstrap confidence interval and have the character of relating directly to tests of hypotheses and thus have a random character. The interval calculated on the true sampling distribution shown above will just get more and more accurate as we sample more and more.

   Here is another way to show it: Assume that the true population is produced by a machine using a Normal distribution with mean 0 and sd 1 (often written N(0,1)). So, the mean is 0. There is no distribution here, just a single number.

   Now, say we ask the question, "How would estimates of the population mean bounce around if could only afford samples of size 100?" No single sample of size 100 will do justice to the true population, but a well-taken sample of size 100 will tell us something about the true population. In our real applied work we *imagine* we could take lots of samples of size 100. So, here, in this exercise, we actually do it:

   ```
   samps.from.pop<-replicate(1000,rnorm(100)) ##could also have used do(1000)*rnorm(100)
   str(samps.from.pop)
   ##100 rows and 1000 columns
   ```

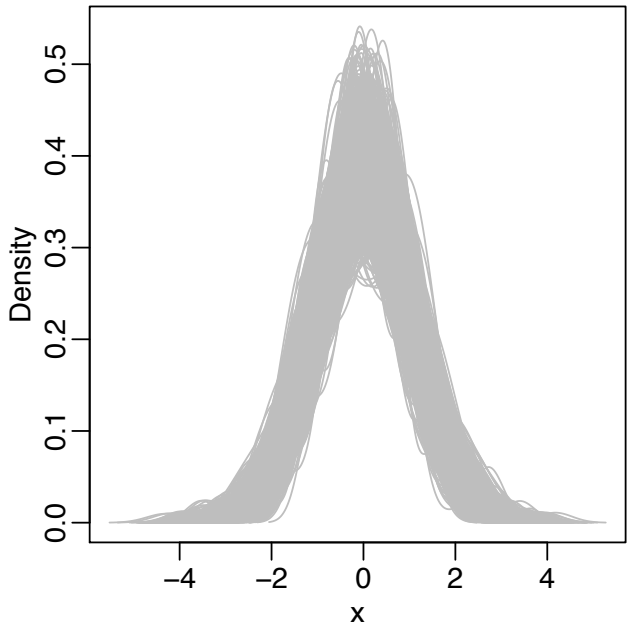
   Notice that the means we get from the sample vary:

   ```
   means.from.samps<-colMeans(samps.from.pop)
   summary(means.from.samps)
   sd(means.from.samps)
   ```

   So, the mean of the means is around 0 — and if we didn't know the population information, we'd say, that we guessed that the population mean is around 0. However, we might also say that our guess could be off by around $\pm.1$ [the sd of the means]. That 50% of the means are in [-0.0674,0.0676] also tells us that the true population mean is near zero — but it doesn't tell us something about the probability that it is near zero.

   Interestingly, the fact that the standard error of the sampling distribution is about .1, combined with a hunch that the actual population might be normalish [from, say, looking at the following plot in which the density of the actual observations from each sample are overlaid on one another] and knowing that standard errors of well-behaved sampling processes tend to be proportional to variance of the population$/\sqrt{n}$ leads me to guess that the sd of the underlying population is 1.

   ```
   the.dens<-apply(samps.from.pop,2,density)
   the.ylim<-range(sapply(the.dens,function(obj){range(obj$y)}))
   the.xlim<-range(sapply(the.dens,function(obj){range(obj$x)}))
   plot(the.xlim,the.ylim,type="n",xlab="x",ylab="Density")
   for(i in 1:length(the.dens)){
     lines(the.dens[[i]],col="gray")
   }
   ```



This next just shows how the standard error of the sampling distribution of the mean can relate to the standard deviation of the underlying population.

```
blah<-replicate(1000,mean(rnorm(100,mean=0,sd=2)))
sd(blah)
##solve x/sqrt(100)=sd(blah) for x to get the sd of the population
##generating these means (in this case of a Normal population and
##Simple random sampling from it)
sd(blah)/sqrt(1/100)
```

4. What is the standard deviation of this distribution? Is there another name for this standard deviation that you've seen? [i.e. the standard deviation of a sampling distribution] What is it?

   The standard deviation of this distribution is the standard error. Most of the time we only have an estimated standard error which approximates what you have here. This measure of the width of the sampling distribution summarizes how much the estimate varies from sample to sample. Of course, what we have here is an approximation because the thought experiment involves taking samples infinitely many times.

   The width of $\pm$ 1 sd is plotted as a double-headed arrow with the mean of the distribution as a small vertical arrow.

   The actual mean difference in the population is plotted with thick vertical line.

5. What is the mean of this distribution? How far off was our sample estimate from this number? Why would our sample estimate be off in any given sample?

   ```
   ##The mean of the sampling distribution is about -.11
   (themean<-mean(sampfits[,3]))
   ##Our estimate is  fit1["age355ymas",]
   fit1["age355ymas",]
   ```

   Although our simple random sampling (as instantiated in `resample()` which is just a fancy wrapper for `sample()`) guarantees that our sample will be representative of the population, what this really means is that *across repeated sampling in the same way, from the same population* the average of the many sample estimates will be the same as the average of the population. You might have seen this written as $E(\hat{\beta}) = \beta$ — estimators like mean differences (aka least squares) tend to have this property which is often called "unbiasedness". So, it doesn't guarantee that any given sample will be a good sample, but it will guarantee that the bad samples ought to be fairly rare.

6. Ok. Now, imagine that we don't have the population, but that, in fact, we only have the sample of 100 people. How can we approximate this sampling distribution (which will be the source of "probability" in our statistical inference)? Well, our

sample was drawn using a simple random sample. That is, each unit in the population had an equal chance of falling into our sample. Is there some way to repeat this using only the information in our sample? Some way to simulate the re-running of this survey? [The answer is yes!].

Ok. So, now resample from our sample in the same way that we sampled from the population BUT, *with replacement*. Calculate a basic/simple bootstrap 95% confidence interval using the results of your simulation. *Hint:* The code you just used to re-sample from the population only requires a couple of changes to enable you to use it to draw **bootstrap** samples from your observed sample. In particular, using the bootstrap procedure to approximate the true sampling distribution requires that we sample **with replacement** and that we draw samples the same size as our dataset.

This interval is often called the "Percentile Interval" (see, for example, Fox § 21.2.2).

```
nsamps<-1000
bootfits<-do(nsamps)*fit.please(milworry~age3,thedata=resample(chile90s,replace=TRUE))
colnames(bootfits)<-rownames(fit1)
quantile(bootfits[,"age355ymas"],c(.025,.975))
```

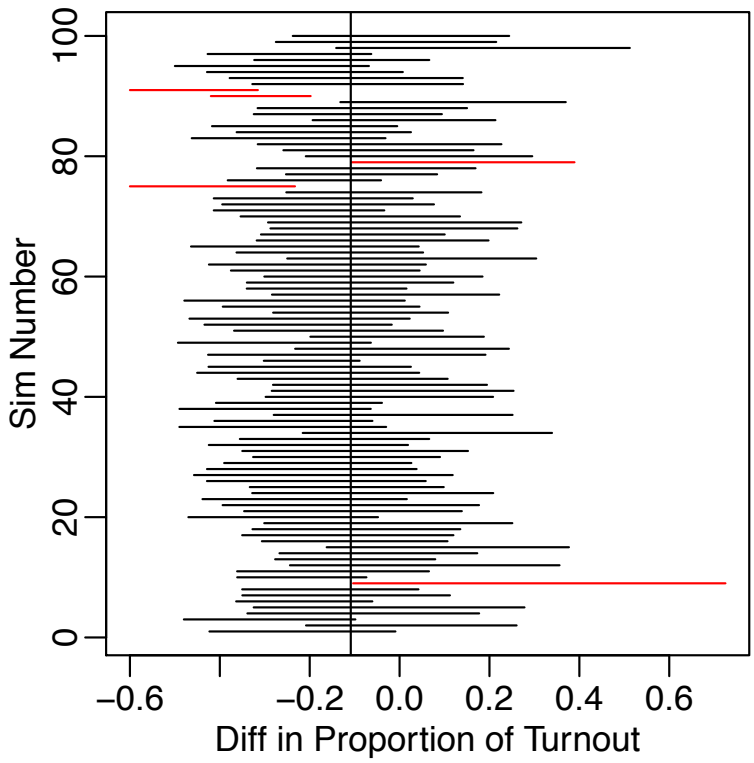
Now, how does this interval, which I call a confidence interval relate to the "real" sampling distribution of the estimate in the population that we calculated above. Here I do the same thing as we did above — sampling repeatedly from the actual population without replacement. But, then I also do 500 bootstrap samples (i.e. samples with replacement) from the sample-data. I record the 95% percentile CI from each sample.

```
ncis<-100
sampsize<-100
n.bsamps<-500
if (inherits(try(load("handout3-somecis.rda")),'try-error')){
##set.seed(20100214)
ci.please<-function(nsamps=n.bsamps){
  thesamp<-resample(chile90,size=sampsize,replace=FALSE) ##sample from the pop.
##Next, do n.bsamps bootstrap resamples with replacement to make the conf. interval
  theboots<-do(nsamps)*lm(milworry~age3,data=resample(thesamp,replace=TRUE))
  theci<-quantile(theboots[,"age355ymas"],c(.025,.975))
  theestse<-sd(theboots[,"age355ymas"])
  return(c(ll=theci[[1]],ul=theci[[2]],se=theestse))
  }

somecis<-do(ncis)*ci.please()

save(somecis,file="handout3-somecis.rda")
}
```

```
overlap<- themean>=somecis[,"ll"]&themean<=somecis[,"ul"]
plot(range(somecis[,c("ll","ul")]),c(1,ncis),type="n",xlab="Diff in Proportion of Turnout",ylab="Sim Numb
segments(somecis[,"ll"],1:ncis,somecis[,"ul"],1:ncis,col=c("red","black")[overlap+1])
abline(v=themean)
```



Notice that some of the intervals [colored in red] do not overlap the mean of the population (in this case the mean is the average of the mean-differences between the oldest and youngest groups — which is the difference in the proportion reporting worry about the military between those two age groups). In fact, in this case only 5 out of 100 confidence intervals do not contain the population value — and this is exactly what we'd expect from a 95% confidence interval. That is, by excluding null hypotheses that we'd reject with p-values lower than .05, we are saying, in essence, that we are willing to reject the null incorrectly about 5% of the time. And, we never know if we are rejecting in error unless we repeat the study — recall that we would only see *one* of these intervals.

```
##number of intervals that contain the population mean
table(overlap)
##lengths of the intervals: Most are about the same length.
summary( apply(somecis,1,function(x){diff(x[c("ll","ul")])}))
##The standard errors calculated using the different samples:
summary( somecis[,"se"])
##The "true" standard error:
sd(sampfits[,"age355ymas"])
##The "true" width of 95% of the estimates across replications in the population:
diff(quantile(sampfits[,"age355ymas"],c(.025,.975)))
```

Effect of sample size on width of interval/sensitivity of hypothesis test.

Now, these results, a sampling distribution for an estimate of a difference of means with a standard error of about .12, are specific to samples of size 100. As our sample size increases, the width of our sampling distribution will shrink, and thus the width of our our estimated sampling distributions [via the bootstrap] will shrink, and thus our confidence intervals will be more narrow.

For example, now let's just calculate confidence intervals based on different sized samples from our "population".

```
sampsizes<-seq(100,1000,by=100)
ci.and.sampsize<-function(samplesize,n.bsamps=500){
  #Sample from the population of size samplesize
  thesamp<-resample(chile90,size=samplesize,replace=FALSE)
  ##Next, do n.bsamps bootstrap resamples with replacement to make the conf. interval
```

```
  theboots<-do(n.bsamps)*lm(milworry~age3,data=resample(thesamp,replace=TRUE))
  theci<-quantile(theboots[,"age355ymas"],c(.025,.975))
  theestse<-sd(theboots[,"age355ymas"])
  return(c(n=samplesize,ll=theci[1],ul=theci[2],
           widthCI=(theci[2]-theci[1]),se=theestse))
}
morecis<-sapply(sampsizes,function(n){
  ci.and.sampsize(samplesize=n)
})
print(morecis)
```

```
                 [,1]     [,2]     [,3]     [,4]      [,5]     [,6]     [,7]
n             100.0000 200.0000 300.0000 400.0000 500.00000 600.0000 700.0000
ll.2.5%        -0.4756  -0.2930  -0.3043  -0.1565  -0.19770  -0.2210  -0.1984
ul.97.5%       -0.0388   0.0078  -0.0633   0.0734   0.00358  -0.0321  -0.0216
widthCI.97.5%   0.4368   0.3008   0.2410   0.2299   0.20127   0.1888   0.1768
se              0.1075   0.0807   0.0604   0.0573   0.05035   0.0479   0.0432
                 [,8]     [,9]     [,10]
n             800.0000 900.0000 1000.0000
ll.2.5%        -0.1914  -0.1744   -0.2148
ul.97.5%       -0.0298  -0.0178   -0.0731
widthCI.97.5%   0.1616   0.1566    0.1417
se              0.0432   0.0402    0.0350
```

So you can see that the standard error and the width of the CIs decrease as sample size rises [later we'll learn a formula to calculate these widths under certain large sample assumptions as an aid to designing studies].

Why sample with replacement? So, the larger the sample size the narrower the CI — the smaller the range of plausible values. So, in this case, if our dataset is actually about 1100 large, we'd prefer to re-sample from it with a sample size as large as possible. It turns out, of course, that the largest possible sample from it is exactly the same as the size of the dataset. Thus, when we say `resample(chile90,replace=TRUE)` what is happening is that we get a new dataset with the same number of rows as chile90. Now, why with replacement? Recall the simple idea behind the bootstrap: we sample from the sample the same way that we sample from the population; we use the process of sampling from the sample to approximate the process of sampling from the population. Any given sample from the population may have more or fewer or each type of row/unit/person. Thus, with replacement most closely mimics the kinds of outcomes we could see if we were to actually draw new samples from the population.

7. How does this confidence interval for your approximate sampling distribution relate to the coverage interval calculated on the actual sampling distribution? *Hint:* Try looking at not only the end points, but also at the width of the intervals on the scale of the outcome [i.e. the coefficient is indicating a difference of means].

   The bootstrapped sampling distribution or estimated sampling distribution is pretty close to the actual one: the standard errors from the bootstrap approximated distributions above are centered on the standard error of the population distribution itself. The range of the 95% coverage interval for the population is about the same as the width of the 95% confidence intervals calculated on the samples.

   Thus Kaplan says:

   "Of all the studies that have computed 95% confidence intervals properly, 95% of them will have captured the population parameter relevant to their study within their confidence interval." (page 257)

   This is clunky, so we often correctly talk about the CI as an indication of precision — rather than reporting a single coefficient, we report a possible range. This range pertains more to our procedure (running a linear model multiple times on different samples (or resamples)) than it does to the data, but it reflects on the amount of information available within a given dataset.

8. When might we worry about the bootstrap not working? Let's explore this by using the newspapers dataset. Try producing a bootstrap confidence interval for the treatment effect. Then ask yourself (1) how wide is this interval compared to the one you did above by inverting hypothesis tests [realizing that we only did a one-sided interval above, but thinking about the distance from 0 to the upper-bound] and (2) the conceptual problem of the target of inference.*Hint:* Try using `coef(lm())` if you have "system is exactly singular" messages. Also try `quantile()` with the `na.rm=TRUE` argument set. Notice that these kinds of

errors are giving you a hint about the problems associated with sampling *with replacement* from small datasets.

```
newsbs<-do(1000)*coef(lm(r~z,data=resample(news.df,replace=TRUE)))
quantile(newsbs$z,c(.17,.83),na.rm=TRUE) ##for a 66\% interval
```

```
  17%   83%
-12.2  15.3
```

```
summary(newsbs$z)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 -45.0    -8.0     2.0     1.8    11.7    44.3      10
```

```
##The two-sided CI calculated by inverting the hypothesis test above
##was about -6 to 3.5
theps[c("h","ptwo"),]
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]   [,8]   [,9] [,10]  [,11]  [,12] [,13]
h     -10 -9.5   -9 -8.5   -8 -7.5   -7 -6.500 -6.000 -5.50 -5.000 -4.500 -4.00
ptwo    0  0.0    0  0.0    0  0.0    0  0.106  0.144  0.12  0.216  0.372  0.38
      [,14]  [,15] [,16]  [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
h    -3.500 -3.000 -2.50 -2.000 -1.50 -1.00 -0.50  0.00 0.500 1.000 1.500 2.000
ptwo  0.492  0.748  0.73  0.878  1.13  0.85  0.78  0.79 0.494 0.408 0.382 0.242
     [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
h     2.50 3.000 3.500     4   4.5     5   5.5     6   6.5     7   7.5     8
ptwo  0.13 0.104 0.128     0   0.0     0   0.0     0   0.0     0   0.0     0
     [,38] [,39] [,40] [,41]
h      8.5     9   9.5    10
ptwo   0.0     0   0.0     0
```

The key assumption of the bootstrap is that you are sampling from the sample the same way that you sampled from the population: as the sample to the population, just so the bootstrap sample to the sample. In this case, we don't know how the cities were sampled from some other set of cities, so we have no guarantee that our bootstrap interval is at all meaningful. In fact, resampling in this way ignores the design — not every sample with have exactly 4 treated and 4 controls.

Another problem with the bootstrap: When resampling with replacement from a very small dataset we are apt to get samples in which there is no variation on either or both of the dependent and/or independent variables. This would lead to nearly infinite coefficient estimates (you can see, for example, that some of the bootstrap samples had estimates like 44 percentage points of turnout!). This issue of overly-wide/inflated bootstrap confidence intervals calculated from small samples is more evident when you are using multiple regression because multicollinearity problems arise more quickly in that context: here, the only problem of identifying a unique number comes when z is constant [which, again, is apt to happen sometimes when resampling a small dataset — notice that there were 6 NAs in the bootstrap samples].

The comparison also highlights a distinction in the meaning of the null hypotheses contained in the confidence interval: the bootstrap-based interval as calculated here contains *weak null hypotheses* but the other confidence interval contains sharp null hypotheses about constant, additive effects. The weak null merely would say that the mean turnout in the treatment group would be the same as the mean turnout in the control group (or would be a constant added onto these means) without specifying whether the constant would be added to all of the units, or just some of them. For example, the weak null of 4 is contained in the bootstrap interval and the strict null of 4 is also contained in the randomization-based interval. Yet, the meaning of these two nulls differs: the strict null claims that the treatment effect added 4 points of turnout to all of the units. The weak null merely says that the difference in proportions is 4 — which could be consistent with negative effects on some cities and effects much stronger than 4 on other cities.

9. If we can build a confidence interval out of hypothesis tests, can we start with a confidence interval and get a hypothesis test? [yes] Let's do a bootstrap-based hypothesis test. So, we can invert the interval to get a hypothesis test just as we can invert the test to get an interval. Here, the p-value is the quantile of the boostrap distribution at which the test statistic takes on the value of the null hypothesis. This is, by necessity, a weak null. So, say the null is that the oldest group were no different from the youngest group (on average). This amounts to $H_0 : \beta_{\text{age355ymas}} = 0$.

   Hints: What proportion of bootstrapped coefficients for age355ymas are greater than or equal to 0? [this is the upper-tailed p-value]. What proportion of bootstrapped coefficients for age355ymas are greater than or equal to 0? [this is the upper-tailed

p-value]. And recall the definition of a two-tailed p-value: `2*min(lower.tail.p,upper.tail.p)`.

```
##For comparison: here is what the canned regression package using the classical
##linear regression assumptions tells us
lm1<-lm(milworry~age3,data=chile90s)
summary(lm1)$coef["age355ymas",c("Estimate","Pr(>|t|)")]
confint(lm1,parm="age355ymas")
```

What is important here is that we are inverting the interval (just as we inverted the test for the permutation based test). Here, the p-value is the quantile of the boostrap distribution at which the test statistic takes on the value of the null hypothesis. This is, by necessity, a weak null.[3]

```
##recall that bootfits is the 1000 x 3 matrix containing the results of the 1000 bootstrap samples for t

##First just look at the object --- str() gives a rough summary of the structure
str(bootfits)
class(bootfits) ##tells us the kind of object
##The standard large sample theory method for getting p-values
##summary(lm(milworry~age3,data=chile90s))$coef

##Recall the fit:
fit.please(milworry~age3,thedata=chile90s)
##Test of whether diff. between youngest and oldest groups = 0
##H0: meandiff=0 or regcoef=0 [same thing here with 2 indicator variables]

##Putting () around commands automatically prints out the result.

(low.tail.p.value<-mean(bootfits[,3]<=0))
(up.tail.p.value<-mean(bootfits[,3]>=0))
##Notice: this is not TRUE here because we have 2 bootstrap samples with
##the estimate=0
(all.equal(up.tail.p.value,(1-low.tail.p.value)))
table(bootfits[,3]==0)
##Equal/Symmetric sided Two-tailed test
(two.tail.p.value<-2*min(low.tail.p.value,up.tail.p.value))
##Compare to the large-sample theory p-value:
summary(lm(milworry~age3,data=chile90s))$coef["age355ymas",c("Estimate","Pr(>|t|)")]
##Estimated standard error of the estimate (aka sd of the estimated/approximated sampling distribution):

(est.se<-sd(bootfits[,3]))
##The observed value of the test statistic:
(the.obs.diff<-fit.please(milworry~age3,thedata=chile90s)["age355ymas",])
##A bootstrap based t-statistic (for what it is worth, not used in our pvalue calc here)
(boot.t.stat<-the.obs.diff/est.se)
##Our boot strap based row of ye olde regression table
c(Estimate=the.obs.diff,SE=est.se,t=boot.t.stat,p=two.tail.p.value)
##The standard large sample theory based row of the regression table.
summary(lm(milworry~age3,data=chile90s))$coef["age355ymas",]
confint(lm(milworry~age3,data=chile90s))["age355ymas",]
quantile(bootfits[,3],c(.025,.975))
```

So, our p-value from inverting the bootstrap CI is basically the same as the large-sample theory p-value that comes from the standard regression table. And our confidence interval is also basically the same as the large-sample theory CI.

10. Now, Kaplan encourages us to use 2 $\times$ the standard deviation of the bootstrap distribution rather than merely the quantiles. Why is that? How variable from bootstrap-sampling run to bootstrap-sampling run is 2 $\times$ the sd versus the width of the quantiles [i.e. the 95 pctile-2.5 pctile]? Why would we use "2" times the sd rather than some other number anyway? *Hint:* We

[3] An illustration of the bootstrap hypothesis testing procedure following Fox § 21.4. See also http://www.stat.umn.edu/geyer/old/5601/examp/tests.html

want to do a bunch of bootstrap simulations recording the sd of the simulated sampling distribution and the distance between the quantiles each time. We'd prefer a procedure that bounces around less from simulation to simulation. Let's do this using the 100 person chilean sample and the regression we did above. (Try perhaps 100 bootstrap studies in which each bootstrap study uses 500 samples. Or change depending on your computer, etc..)

Hint 2: To assess this, we do a simulation: we create many different bootstrap sampling distributions and we record 2 $\times$ the standard deviation of each those distributions as well as the simple width of the percentile 95% CI. Does `diff(quantile(thebs.dist,c(.025` vary more or less, in general, than `2*sd(thebs.dist)`?

To assess this, we do a simulation: we create many different bootstrap sampling distributions and we record 2 $\times$ the standard deviation of those distributions as well as the simple width of the percentile 95% CI.

```
nsims<-100
n.bsamps<-500
myfn<-function(){
  thebsdist<-do(n.bsamps)*fit.please(milworry~age3,thedata=resample(chile90s,replace=TRUE))
  thesd<-2*sd(thebsdist[,3])
  theci<-quantile(thebsdist[,3],c(.025,.975))
  thequant.diff<-diff(theci)/2
  return(c(sd=thesd,qdiff=thequant.diff))
}
results<-matrix(ncol=2,nrow=nsims,dimnames=list(1:nsims,c("sd","quantdiff")))
for(i in 1:nsims){
  results[i,]<-myfn()
  }
summary(results)
```

```
      sd          quantdiff
Min.   :0.190   Min.   :0.182
1st Qu.:0.206   1st Qu.:0.198
Median :0.212   Median :0.204
Mean   :0.211   Mean   :0.204
3rd Qu.:0.215   3rd Qu.:0.210
Max.   :0.228   Max.   :0.225
```

```
apply(results,2,sd)
```

```
      sd quantdiff
 0.00696   0.00921
```

```
apply(results,2,IQR)
```

```
      sd quantdiff
 0.00919   0.01179
```

I use the standard deviation to summarize the variability across the simulations in the SEs versus the percentile intervals. We can see that, while the end-points of the sampling distributions may bounce around a fair amount (and thus cause more variability in the percentile intervals), the standard deviation of the sampling distributions varies less.

Thus, if you were wondering why we always see standard errors, one reason is that standard errors are a nice summary of variability that, themselves, have good operating characteristics. Now, where "2" comes from in using the SEs for confidence intervals is something that we'll have to wait to discuss until after we've encountered the central limit theorem and the t-distribution. Notice, however, from the summaries of the two widths (2*se and difference in percentile end-points), that the difference in percentile end-points is not dissimilar from 2*se.

Summing up hypothesis testing and confidence interval generation

We've seen that, at the most fundamental level, statistical inference is the act of *inferring*: of using what we see or know to guess about what we don't see or know, and in particular, *statistical inference* is about making probability statements about the unknown [notice that is not the same as making descriptive or causal statements]. We've also gained some experience with the two dominant ways to make such probability statements: hypothesis tests (i.e. summarizing the evidence against a posited guess — the null hypothesis) and confidence intervals (i.e. delineating a plausible range of entertained guesses). And

we've seen the relationship between hypothesis tests and confidence intervals: in fact we've inverted a hypothesis test under a model of constant and additive effects to produce a confidence interval.

We've also recalled that probability statements require probability distributions and our practice here has reflected the frequentist understanding of where probability distributions come from: *from repetition of some physical process*. So, since we knew how the Newspapers experiment was run, we repeated it by shuffling or permutation to generate the *randomization-based distribution* of the least squares test statistic (which is identical to a difference of means).[4] And, since we knew how our small 100 person sample of the 1200 person survey was drawn, we were able to repeatedly sample from the sample using the *bootstrap* procedure to generate a *sampling-based distribution* (or a "re-sampling" based distribution).

We've seen that there are two different ways to specify the hypothesis to be tested although we didn't spend much in-class time on the most common method: which is to merely say $H_0 : \bar{r}_{1i} = \bar{r}_{0i}$ that is, to talk in terms of means and to avoid talking in terms of the individual units. This null, often called the "weak null" does not require that you say anything in particular about the pattern of effects at the level of the individual unit (for example, we could easily have a situation in which a large p-value encouraged us to no-reject the null but, in fact, the treatment had equal numbers of *both* really large and positive effect *and* really large and negative effects, all of which cancelled out to create a mean difference of zero). So, the "weak null" might be useful sometimes and the "sharp", "strict", or "strong null" might be useful sometimes. In general, one "weak null" can encompass many "strong nulls" — and can be thought of as summarizing a collection of "strong nulls".

We engaged a bit with the limitations of the two approaches: the bootstrap on small datasets can produce ultra-wide sampling distributions because with-replacement sampling can easily give you bootstrap samples with all the same unit or only two kinds of units and thus undefined or infinite or zero answers [in our case we'd be asking for a difference of means in a case where there are no controls, for example]. So, the bootstrap is not a panacea for small samples.

The permutation based approach using direct enumeration (i.e. listing all of Ω) quickly is infeasible as sample sizes grow, although the simulation (i.e. shuffling) based approach can still easily be used. It can be particularly time consuming to invert the hypothesis test — each null stated requires it's own large set of simulations, and finely defined confidence intervals may require testing a lot of hypotheses. We didn't engage much with the question about evaluating the model of effects that is an assumption required to assess the hypotheses about effects (in contrast to the null hypothesis of no effects whatsoever — which, it turns out, is compatible with a wide range of models of effects). We only assessed hypotheses based on the model of constant additive effects: this is the model assumed by standard linear model based test statistics (i.e. if we use the coefficient from the model as the test statistic). However, your reading highlighted other test statistics and other models of effects.

At base, and running throughout this section of the course has been the assumption that *you know the design of the study*. If we didn't know how the sample was drawn from the population (including knowing something about the population) we couldn't really drawn bootstrap samples. If we didn't know how treatment was assigned to the experimental pool, we couldn't shuffle in the right way to repeat the experiment. The validity of these approaches (and the mathematical simplifications and elaborations instantiated in the standard regression table), rests on the ability of the analyst to justify knowledge of design. Later, we'll engage with some of the ways that one can justify statistical inference without being able to, in principle, list the population or Ω — we can claim that we can write down a model which produces said population or set of possible treatment assignments. Of course, at that point, justification then moves to the model of the design rather than the design itself.

[4] And we discovered that we could also just list all of the possible ways to run the experiment in Ω and use that list OR we could simulate Ω by shuffling, and that either method worked although simulation carried with it, its own margin of error [as shown by the fact that even each person in the class got slightly different answers, answers which differed less and less the more we shuffled].