Class 4—Political Science 531
Estimating is not Testing. What is a good estimator?

Jake Bowers

February 25, 2013

Estimating versus Testing

Last time we focused on testing the weak null of no effects. But, we did not think much about why the weak null tends to be stated about an "average effect". That is, why would we think that an average would help us address questions about counterfactuals?

1. Recall this table:

i	Z_i	R_i	r_{1i}	r_{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 (r_1, r_0) for Cities (i) in the Newspapers Experiment.

We have seen how Fisher's testing framework allows us to ask and answer questions about partially observed counterfactuals. Yet, the norm in political science is to discuss produce average differences (remember our basic interpretation of a regression slope?) as *estimates* of something and then they test a hypothesis that these averages are zero. Are they ignoring the counterfactuals here?

No. The hypothesis that they are asking about is about averages of potential outcomes. So the focus is still on unobserved quantities.

2. It turns out that Neyman (1923) proposed (1) that we decide to be interested in $\bar{\tau} = (1/n)(\sum_{i=1}^{n} r_{1i}) - (1/n)(\sum_{i=1}^{n} r_{0i}) = \bar{r}_1 - \bar{r}_0$ (i.e. he suggested we choose to care about mean differences of the counterfactuals rather than any individual counterfactuals themselves) and (2) he claimed that $\hat{\bar{\tau}} = (1/m)\sum_{i=1}^{n} Z_i R_i - (1/(n-m))\sum_{i=1}^{n}(1-Z_i)R_i$ was a good **estimator** of $\bar{\tau}$ (where R_i is observed outcomes and relates to potential outcomes via $R_i = Z_i r_{1i} + (1-Z_i) r_{0i}$, m is number of treated observations and n is total number of observations). We are going to prove this is the case, for a specific understanding of "good". So, first let's start by just thinking generally about what would distinguish good estimators from bad ones. What are some different ways to think about an estimator being "good"? Or otherwise useful? [Perhaps useful to first define "estimator" too.]

3. So, Neyman, chose the idea of "unbiasedness" for "good". An unbiased estimator is an estimator (i.e. a formula that when repeatedly applied to the data (across samples, across assignments, etc..) produces a distribution – a sampling distribution, a randomization distribution), where the mean of this distribution is the "true" value in the "population" (or the experimental pool). In Neyman's example, the population is the set of fields to which he wants to assign fertilizer treatments. For us, it is these 8 cities. We can think of experimental pools as "finite populations." We can write this overly wordy definition concisely like this:

$$E_R[\hat{\bar{\tau}}] = \bar{\tau},$$

where E_R (the "expected value") is "the average across all ways to draw assignments, Z, from the urn." You'll mostly just see something like $E[\hat{\beta}] = \beta$ as a statement that whatever formula is giving you $\hat{\beta}$ is, on average across imaginary replications, equal to β — is unbiased. Now, the expectation operator, E has a kind of algebra associated with it such that $E[aX] = aE[X]$ if X is something that can vary and a is some constant, or $E[X+Y] = E[X]+E[Y]$ if X and Y both vary independently (or $E[aX_1 + aX_2 + \ldots] = a(E[X_1] + E[X_2] + \ldots)$).

Here is an incomplete version of the derivation that shows that, in fact, the observed sample difference of means can tell us something about the unobserved population difference of means, and, in fact, will be, on average, equal to the unobserved population difference of means: i.e. that the sample difference of means is an unbiased estimator of the population difference of means.

We want to show that $E_R[\hat{\bar{\tau}}] = \bar{\tau}$. Here only Z_i is random. And, I can tell you that $E_R[Z_i] = m/n$ (what does this mean?). I use E_R at first just to be clear that we are taking expectations over the possible randomizations. Mostly you'll just see people use E without a discussion of the space over which the expectations are to be taken.

Fill in the missing parts:

$$E_R\left[\hat{\tau}\right] = E_R\left[\sum_{i=1}^{n} Z_i \frac{R_i}{m} - \sum_{i=1}^{n} (1 - Z_i)\frac{R_i}{n-m}\right] \tag{1}$$

recall that $R_i = r_{0i}$ when $Z_i = 0$ and $R_i = r_{1i}$ when $Z_i = 1$

$$= E_R\left[\sum_{i=1}^{n} Z_i \frac{?}{m}\right] - E_R\left[\sum_{i=1}^{n} (1 - Z_i)\frac{?}{n-m}\right] \tag{2}$$

$$=? \tag{3}$$

$$= \sum_{i=1}^{n} \frac{r_{1i}}{n} - \sum_{i=1}^{n} \frac{r_{0i}}{n} \tag{4}$$

$$= \bar{r}_{1i} - \bar{r}_{0i} \tag{5}$$

$$= \bar{\tau} \tag{6}$$

$$E_R\left[\hat{\tau}\right] = E_R\left[\sum_{i=1}^{n} Z_i \frac{R_i}{m} - \sum_{i=1}^{n} (1 - Z_i)\frac{R_i}{n-m}\right] \tag{7}$$

recall that $R_i = Z_i r_{1i} + (1 - Z_i) r_{0i}$ or $R_i = r_{0i}$ when $Z_i = 0$, etc.

$$= E\left[\sum_{i=1}^{n} Z_i \frac{r_{1i}}{m}\right] - E\left[\sum_{i=1}^{n} (1 - Z_i)\frac{r_{0i}}{n-m}\right] \tag{8}$$

recall that only Z is random and that $E_R[Z_i] = m/n$ because no pairs for now

$$= \sum_{i=1}^{n} \frac{m}{n}\frac{r_{1i}}{m} - \sum_{i=1}^{n} (1 - \frac{m}{n})\frac{r_{0i}}{n-m} \tag{9}$$

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

$$= \sum_{i=1}^{n} \frac{r_{1i}}{n} - \sum_{i=1}^{n} \frac{r_{0i}}{n} \tag{10}$$

$$= \bar{r}_{1i} - \bar{r}_{0i} = \bar{\tau} \tag{11}$$

4. Now, imagine that you had a new estimator or a new design or some other reason to wonder whether a data summary that you meant to tell you about an unobserved feature of the data really did so. What kind of computer simulation might you dream up to give you a little insight? What kind of key features would such a simulation have?

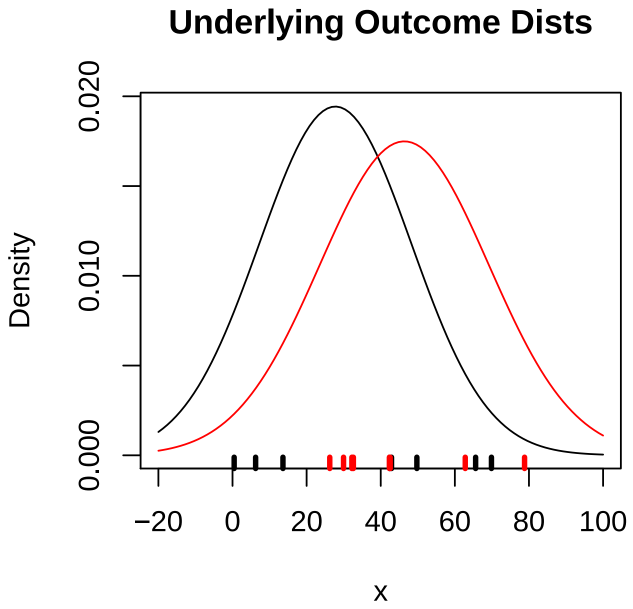
5. Here is my version of such a simulation study. Explain what is happening here. It may help to run each line and print the results.

   First load the data.

```
news.df <- read.csv("http://jakebowers.org/PS531Data/news.df.csv")
```

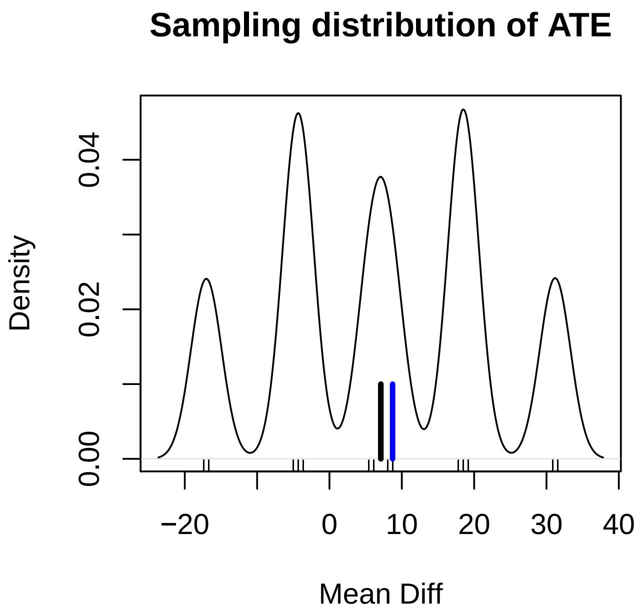
```
##Set up some fake potential and observed outcomes. We are acting as if we could actually observe both
##potential outcomes.
set.seed(20120214)
news.df$faker0<-rnorm(8,mean=mean(news.df$r[news.df$z==0]),sd=sd(news.df$r[news.df$z==0]))
news.df$faker1<-rnorm(8,mean=mean(news.df$faker0)+10,sd=sd(news.df$r[news.df$z==1]))
news.df$obsr<-with(news.df, z*faker1+(1-z)*faker0 )
```

```
curve(dnorm(x, mean(news.df$r[news.df$z == 0]), sd = sd(news.df$r[news.df$z == 0])), from = -20, to = 100, ylab = "Density", main =
curve(dnorm(x, mean(news.df$faker0) + 10, sd = sd(news.df$r[news.df$z == 1])), from = -20, to = 100, add = TRUE, col = "red")
rug(news.df$faker0, lwd = 3)
rug(news.df$faker1, col = "red", lwd = 3)
```



```
## Now calculate the true ATE and the estimated ATE from the sample
trueATE <- with(news.df, mean(faker1) - mean(faker0))
estATE <- with(news.df, mean(obsr[z == 1]) - mean(obsr[z == 0]))

## Now define a function which reveals a difference in observed outcome and calculates a different mean difference given a
## different treatment vector
make.new.R.and.mean.diff <- function(thez) {
    newobsr <- with(news.df, thez * faker1 + (1 - thez) * faker0)
    return(mean(newobsr[thez == 1]) - mean(newobsr[thez == 0]))
}

pair.assignment <- function(pair, z) {
    ## shuffle(z,within=pair) is more elegant but harder to understand as code
    unsplit(lapply(split(z, pair), sample), pair)
}

## Does the pair of functions do what we want them to do?  make.new.R.and.mean.diff(pair.assignment(news.df$s,news.df$z))

## For every way possible to run the experiment, calculate this mean difference
dist.sample.mean.diff <- replicate(10000, make.new.R.and.mean.diff(pair.assignment(news.df$s, news.df$z)))

## Calculate the average of the randomization distribution of the mean difference (i.e. E(\hat{\bar{\tau}}))
E.estATE <- mean(dist.sample.mean.diff)
## And recall that we have simulation error on the order of 1/sqrt(10000)
```

```
plot(density(dist.sample.mean.diff), main = "Sampling distribution of ATE", xlab = "Mean Diff")
rug(dist.sample.mean.diff)
segments(c(E.estATE, estATE, trueATE), rep(0, 3), c(E.estATE, estATE, trueATE), rep(0.01, 3), lwd = 3, col = c("red", "blue", "bla
```



So, does this little simulation suggest that our estimator (the difference of sample means) is unbiased? Is unbiasedness something that we'd like in an estimator?

Assumptions and Evaluation of Frequentist Statistical Inference Techniques

Now, let's think about what we'd like from a testing procedure — what kinds of minimal standards of operation ought we to expect.

4. First, just load data and functions. We'll work with that immediately-post-democratization survey of Chileans again:

```
## this next loads helpful functions into your R session
source("http://jakebowers.org/PS531Data/ps531fns.R")
load(url("http://jakebowers.org/PS531Data/chile90.rda"))  ## Using the Chile 1990 Survey again
library(car)

## Loading required package:  MASS

## Loading required package:  nnet
```

Some recoding here:

```
## milworry=1 if mentioned worry in either variable P202 or P201 about conflicts with the military, 0 otherwise This trick relies
## on the fact that R converts TRUE to 1 and FALSE to 0 when appropriate.
chile90$milworry <- (chile90$P202 == 6) + (chile90$P201 == 6)
table(chile90$milworry, useNA = "ifany")
chile90$age3 <- recode(chile90$P26, "1:2='18-34';3:4='35-54';5:6='55ymas'", as.factor.result = TRUE)
```

5. Last time we assessed the relationship between age and worries about conflict between civil society and the military in Chile, one year after the transition to democracy. And you used the bootstrap to do statistical inference. What did we need to assume in order for the bootstrap to be credible?

We have to believe that we could (re)sample from our sample the same way that our sample was drawn from the population. Any non-iid-ness in the original sampling has to be passed on to the resampling.

In our particular case, we sampled from our "population" such that each person in the population had equal and independent probability of entering our sample.

6. We did not do a permutation based hypothesis test on this data. Please test the sharp null of no relationship between milworry and age against a two-sided alternative. Recall the original model and original data:

```
set.seed(20100205)
## using n=100 here for speed
chile90$ideology <- ifelse(chile90$P22 == 8, NA, chile90$P22)  ##Code DK as missing for now, Codebook excerpt below
chile90s <- resample(chile90[, c("age3", "milworry", "ideology")], size = 100, replace = FALSE)
```

```
lm1 <- lm(milworry ~ age3, data = chile90s)
(lm1.coef <- coef(lm1))
```

And recall that you can generate the distribution of the test statistic (the regression coefficient) under the null hypothesis in a single line of R code (and you've done this before in previous classworks).

Recall that shuffling the values of age3 represents the null hypothesis of no effects and also, we claim, the process by which the values of age3 came to occur in our dataset.

```
shuff.fits <- do(1000) * fit.please(milworry ~ shuffle(age3), thedata = chile90s)
```

Recall that there are other definitions of what a two-tailed test would be after permutation. This definition is my favorite. I can give references to other arguments if you'd like.

```
## one-tailed alternative
mean(shuff.fits[, 3] <= lm1.coef[3])
## two tailed as a function of one-tailed alternatives
2 * min(mean(shuff.fits[, 3] <= lm1.coef[3]), mean(shuff.fits[, 3] >= lm1.coef[3]))
```

Notice that Kaplan likes us to shuffle the outcome — it breaks the relationship between the outcome and *all* the explanatory variables. In this case, the results should be more or less the same (within the simulation error of $\sqrt{\frac{p(1-p)}{1000}} \approx .016$ with $p = .5$).

```
shuff.fits.2 <- do(1000) * fit.please(shuffle(milworry) ~ age3, thedata = chile90s)
```

```
mean(shuff.fits.2[, 3] <= lm1.coef[3])
## two tailed
2 * min(mean(shuff.fits.2[, 3] <= lm1.coef[3]), mean(shuff.fits.2[, 3] >= lm1.coef[3]))
```

7. What assumptions were required for us to believe the permutation based approach to statistical inference?

That our shuffling of the values of our explanatory variable reflects the way in which those values came to be observed. In this case, we are assuming that the distribution of age groups in this dataset was fixed:

```
table(chile90s$age3)/sum(table(chile90s$age3))
```

That is, each re-shuffling ensured that exactly 27 % of the people were coded in the "55ymas" category each reshuffling.

We also assumed, by implication, that it was meaningful to entertain scenarios in which different people got "assigned" different ages. In the shuffling the outcome version of the story people often talk about "exchangeable outcomes" meaning that one person's outcome can be swapped with that of another person to reflect the idea of that the relationship between outcomes and explanatory variables could have been different.

Notice that both the bootstrap and the permutation based approaches require you to know something about a particular piece of the research design: either you need to know how the sample was drawn from the population or you need to know how the values of the explanatory variable of interest were assigned (or how the values of the outcome were assigned). If you know either fact you can repeat it on the computer.

In neither case did we assume homoskedasticity or asymptotic-sized samples. We did assume a kind of independence assumption but that assumption arises from the facts of the design (as we here pretend them to be) rather than from outside the data. So, again, neither approach requires the iid+large-sample assumptions of the canned regression model to make statistical inference meaningful. What they do require is knowledge about how the data arrived.

8. So now we have three p-values: (1) the permutation based p-value you just produced, (2) a bootstrap based p-value:

```
## Bootstrap
bs.fits <- replicate(1000, coef(lm(milworry ~ age3, data = resample(chile90s, replace = TRUE)))["age355ymas"])
```

```
summary(bs.fits)
mean(bs.fits >= 0, na.rm = TRUE)
2 * min(mean(bs.fits >= 0, na.rm = TRUE), mean(bs.fits <= 0, na.rm = TRUE))
```

Notice that, in the observed data, when we break it down into 6 cells in the following table, some of the entries are very sparse/rare. So, it would not be strange to sometimes get a bootstrap sample with no units in the (oldest, worried about the military) cell.

```
table(chile90s$age3, chile90s$milworry)
```

and (3) the large-sample theory, iid based p-value that doesn't require any extensive computation:

```
summary(lm1)$coef["age355ymas", c(1, 2, 4)]
```

Which one (or ones) ought we to feel is most useful and least misleading? As a way to answer this question, let's try a simulation study: under the situation in which there is no relationship between age and worry about the military, how often should we get a p-value less than or equal to α (where, often $\alpha = .05$)?

Try the following code. What do you find? What proportion of the time do you see p-values less than .05? Than .1?

```
lm.sim.p.fn <- function(y = chile90s$milworry, z = chile90s$age3) {
    summary(lm(y ~ shuffle(z)))$coef[, "Pr(>|t|)"]
}

lm.sim.confint <- function(alpha = 0.05) {
    theci <- confint(lm(milworry ~ shuffle(age3), data = chile90s), level = 1 - alpha, parm = "shuffle(age3)55ymas")
    theci[1] <= 0 & 0 <= theci[2]  ##is 0 in the CI?
}

sim.lm.p <- replicate(1000, lm.sim.p.fn())
sim.lm.c <- replicate(1000, lm.sim.confint())
```

So, the confidence interval simulation and the p-value simulation tell the same story. I use both to make the points more clear.

```
mean(sim.lm.c)  ##0 ought to be in 95% of the 95\% confidence intervals

str(sim.lm.p)
apply(sim.lm.p, 1, function(x) {
    quantile(x, c(0.025, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.975))
})
mean(sim.lm.p["shuffle(z)55ymas", ] <= 0.05)
quantile(sim.lm.p["shuffle(z)55ymas", ], p = mean(sim.lm.p["shuffle(z)55ymas", ] <= 0.05))
mean(sim.lm.p["shuffle(z)55ymas", ] <= 0.1)
quantile(sim.lm.p["shuffle(z)55ymas", ], p = mean(sim.lm.p["shuffle(z)55ymas", ] <= 0.1))
```

What does this simulation suggest about whether the large-sample, iid theory linear model p-values are useful or not in this particular dataset?

By shuffling age3, we are breaking the relationship of age3 with the outcome — the coefficient should be zero, on average (within simulation error).

Anyway, we see that the 95% confidence intervals contain 0 a bit more than 95% of the time. This makes them a bit too wide (although they seem to be only slightly too wide). We also see too few p-values of .05 below .05 — we are rejecting at .05 too rarely, leading to a conservative test (but only slightly so). We see, in fact about 3.7 % below .05.

This next just shows how shuffling makes the relationships zero on average without making the intercept zero:

```
sim.coef1 <- do(1000) * coef(lm(milworry ~ shuffle(age3), data = chile90s))
sim.coef2 <- do(1000) * coef(lm(shuffle(milworry) ~ age3, data = chile90s))
```

```
summary(sim.coef1)
summary(sim.coef2)
```

So, leaving this simulation as is, I would say that it doesn't make us that worried about the extent to which the iid+large-sample assumptions are approximating the situation in these data.

9. Now, what about the permutation based test? Does it perform as we would expect?

```
perm.sim.p.fn <- function(n.shuf, y = chile90s$milworry, z = chile90s$age3) {
    newz <- shuffle(z)  ## for each sim, make a new age var that has no relationship with the outcome.
    obs.coef <- coef(lm(y ~ newz))
    perm.dist <- do(n.shuf) * coef(lm(y ~ shuffle(newz)))
    ## two tailed p value
    theps <- sapply(1:length(obs.coef), function(which.coef) {
        upper.p <- mean(perm.dist[, which.coef] >= obs.coef[which.coef])
        lower.p <- mean(perm.dist[, which.coef] <= obs.coef[which.coef])
        return(2 * min(upper.p, lower.p))
    })
    names(theps) <- names(obs.coef)

    return(theps)
}


## Do this in parallel on all 8 cores on my office machine
library(parallel)
## sim.perm.p<-replicate(100,perm.sim.p.fn(500)) ## old way, not parallel
sim.perm.p <- simplify2array(mclapply(1:100, function(i) {
    perm.sim.p.fn(500)
}, mc.cores = 8))
```

Notice here the relationship with the Intercept is not guaranteed to be zero so the below results do not reflect on the performance of the procedure when applied to the intercept.

```
str(sim.perm.p)
apply(sim.perm.p, 1, summary)
apply(sim.perm.p, 1, function(x) {
    mean(x <= 0.05)
})
apply(sim.perm.p, 1, function(x) {
    quantile(x, c(0.025, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.975))
})
```

Now, here is a fake data setup in which all three should work well. Imagining that we know the population (Normal) and sampling method (iid) and the assignment of values to z method (50% 1s and 50% 0s at random with equal probability).

```
set.seed(20110218)
y <- rnorm(100)
z <- rep(c(1, 0), 50)

## Notice these two are the same
t.test(y ~ z)$p.value
summary(lm(y ~ z))$coef["z", 4]

## Coverage/performance of the large sample iid method
lm.sim.p.2 <- replicate(1000, lm.sim.p.fn(y = y, z = z))
apply(lm.sim.p.2, 1, summary)
apply(lm.sim.p.2, 1, function(x) {
    mean(x <= 0.05)
})
apply(lm.sim.p.2, 1, function(x) {
    quantile(x, c(0.025, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.975))
})

## Of the permutation test method
perm.sim.p.2 <- simplify2array(mclapply(1:100, function(i) {
    perm.sim.p.fn(500, y = y, z = z)
}, mc.cores = 8))
apply(perm.sim.p.2, 1, summary)
apply(perm.sim.p.2, 1, function(x) {
    mean(x <= 0.05)
})
apply(perm.sim.p.2, 1, function(x) {
    quantile(x, c(0.025, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.975))
})

## And of the bootstrap method (using CIs rather than hypothesis tests just for fun)
bs.sim.ci.fn <- function(n.bs, y = chile90s$milworry, z = chile90s$age3) {
    newz <- shuffle(z)  ##for each sim, make a new age var that has no relationship with the outcome.
    newdat <- data.frame(y, newz)
    bs.fits <- do(n.bs) * coef(lm(y ~ newz, data = resample(newdat)))
    bs95ci.coverage <- apply(bs.fits, 2, function(thefit) {
        bs95ci <- quantile(thefit, c(0.025, 0.975), na.rm = TRUE)  ## use simple ptile bs ci for now, perhaps 2*se later
        bs95ci[1] <= 0 & 0 <= bs95ci[2]  ## is 0 in the CI?
    })
    return(bs95ci.coverage)
}

bs.sim.ci.2 <- simplify2array(mclapply(1:100, function(i) {
    bs.sim.ci.fn(500, y = y, z = z)
}, mc.cores = 8))
mean(bs.sim.ci.2)

## Now mess up the data so that iid+large sample approach ought not work

y <- c(rep(0, 100), rnorm(100, 10, seq(1:100)))  ##Still mean zero
y <- sample(y, size = 20)
z <- rep(c(1, 0), 10)

## Coverage/performance of the large sample iid method
lm.sim.p.3 <- replicate(1000, lm.sim.p.fn(y = y, z = z))  ## this is very fast, only 1000 sims vs 100*500 sims.
apply(lm.sim.p.3, 1, summary)
apply(lm.sim.p.3, 1, function(x) {
    mean(x <= 0.05)
})
apply(lm.sim.p.3, 1, function(x) {
    quantile(x, c(0.025, 0.05, 0.1, 0.35, 0.5, 0.75, 0.9, 0.95, 0.975))
})

## Of the permutation test method
perm.sim.p.3 <- simplify2array(mclapply(1:100, function(i) {
    perm.sim.p.fn(500, y = y, z = z)
}, mc.cores = 8))
apply(perm.sim.p.3, 1, summary)
apply(perm.sim.p.3, 1, function(x) {
    mean(x <= 0.05)
})
apply(perm.sim.p.3, 1, function(x) {
    quantile(x, c(0.025, 0.05, 0.1, 0.35, 0.5, 0.75, 0.9, 0.95, 0.975))
})

## Bootstrap
bs.sim.ci.3 <- simplify2array(mclapply(1:100, function(i) {
    bs.sim.ci.fn(500, y = y, z = z)
}, mc.cores = 8))
mean(bs.sim.ci.3)
```

So, the iid+large sample based p-values seem more or less ok.

```
## Here trying the shuffle outcomes method
perm.sim.p.2.fn <- function(n.shuf, y = chile90s$milworry, z = chile90s$age3) {
    newy <- shuffle(y)  ## for each sim, make a new age var that has no relationship with the outcome.
    obs.coef <- coef(lm(newy ~ z))
    perm.dist <- do(n.shuf) * coef(lm(shuffle(newy) ~ z))
    ## two tailed p value

    theps <- sapply(1:length(obs.coef), function(which.coef) {
        upper.p <- mean(perm.dist[, which.coef] >= obs.coef[which.coef])
        lower.p <- mean(perm.dist[, which.coef] <= obs.coef[which.coef])
        return(2 * min(upper.p, lower.p))
    })
    names(theps) <- names(obs.coef)

    return(theps)
}


sim.perm.p.2 <- simplify2array(mclapply(1:100, function(i) {
    perm.sim.p.fn(n.shuf = 100, y = y, z = z)
}, mc.cores = 8))
```

Let us look at bootstrap confidence intervals as another easy assessment:

```
bs.sim.ci.fn <- function(n.bs, milworry = chile90s$milworry) {
    newage <- shuffle(chile90s$age3)  ##for each sim, make a new age var that has no relationship with the outcome.
    newdat <- data.frame(milworry = milworry, newage = newage)
    bs.fits <- do(n.bs) * coef(lm(milworry ~ newage, data = resample(newdat)))["newage55ymas"]
    bs95ci <- quantile(bs.fits, c(0.025, 0.975), na.rm = TRUE)  ## use simple ptile bs ci for now
    bs95ci[1] <= 0 & 0 <= bs95ci[2]  ## is 0 in the CI?
}

sim.bs.ci.coverage <- simplify2array(mclapply(1:100, function(i) {
    bs.sim.ci.fn(100)
}, mc.cores = 8))
```

```
(bs.sim.coverage <- mean(sim.bs.ci.coverage, na.rm = TRUE))
```

So, when using the bootstrap, the confidence interval excludes the true value of 0 92 % of the time. This is a too narrow confidence interval.

Covariance Adjustment: Residualization Approximating Stratification

Ok. So, now we have some sense for how to assess the performance of difference ways to do statistical inference (and some sense for three different methods for doing such inference). Let us now turn our attention to the question of “Controlling for”.

1. Now, use the full dataset (`chile90`) to estimate the difference of means between the youngest group and the oldest group in the survey. What story might we tell to make sense of this relationship?

   ```
   (fit1 <- fit.please(milworry ~ age3, thedata = chile90))

   ## Another way to show that the least squares estimates are just means and differences of means The with() is also a nice way to
   ## get some of the benefits of attach(). See also within() for recoding and such
   with(chile90, tapply(milworry, age3, mean))
   ## or
   tapply(chile90$milworry, chile90$age3, mean)

   diff(c(0, with(chile90, tapply(milworry, age3, mean))))
   ```

   About 32% of the youngest group is worried about near term conflict with the military whereas about 32-14=18% of the oldest group reports such worry.

   Why could this be? How about a hunch about the willingness of people who have just spent about 20 years of their lives under an authoritarian dictatorship to talk about the military government to strangers (i.e. survey interviewers)? The older group lived through and suffered the transition to authoritarianism and related violence most directly. The younger group basically don’t remember it. Thus, the younger group may be less guarded/feel less fear and be more likely to more honestly express themselves. Just a made up hunch here but it would be supported by the work of Timur Kuran, Vaclav Havel, among others on civil society during and after authoritarian/totalitarian regimes.

2. Upon hearing your story, what might a critic say? What alternative explanation might she propose? (I am not asking you to delve into the codebook to assess this explanation here since I don’t assume you all know Spanish! But I wanted you to practice thinking about alternative explanations for relationships that you’ve uncovered.)

   Maybe this relationship is real: since the question was about worry and fear, the old might be telling us that they just worry less about it even if they might judge the probability of another coup as the same as the younger group — i.e. the younger group have more to lose from a regime transition since they have a longer future and thus they ought to worry more.

3. How about this for yet another alternative explanation: People of different ages had very different experiences with the military regime that took power in 1973 and ruled until 1989: the older folks might, in fact, be deeply divided into two groups — one of which supported the military regime (and perhaps directly benefited from it) and the other of which did not (and perhaps directly suffered from it). And what we want here is a comparison among age groups uncontaminated by the ideology of the person — perhaps to reflect something else about age. So, now, let’s “control for” or “adjust” our estimate of the difference between the oldest and the youngest for ideology. Choose either of the two variables recoded below (and make sure to recode them as I did, or some other way that makes sense to you). Now, estimate the mean difference that you did above, but now adjusted for one of the two measures of ideology. Now, what do you find? Interpret the results. *Hint:* Use `fit.please()` or `lm()`.

```
  ##P11L: p11l De 1 a 7 una opinion favorable o desfavorable (A.PINOCHET)
##'1' 1 '2' 2 '3' 3 '4' 4 '5' 5 '6' 6 '7' 7 '9' NO SABE/NO CONTESTA
##11. Las siguientes personas tienen o han tenido diversos grados de
##  notoriedad en la vida publica nacional. Indique con una nota de 1 a
##  7 el grado en que Ud. tiene una opinion favorable segun esta escala
##  de evaluacion (TRATE DE USAR TODOS LOS RANGOS DE LA ESCALA, NO SOLO
##  LOS EXTREMOS) (ENTREVISTADOR: SI CONTESTA NO CONOCE MARCAR
##  9. PARTA ALTERNADAMENTE DE ARRIBA HACIA ABAJO O DE ABAJO HACIA
##  ARRIBA, O DEL CENTRO, COMPLETANDO TODA LA LISTA. INDIQUE POR DONDE
##  PARTIO) PARTIDA

##Note: In Chilean schools a 7=A and 1=F. And a "grade" is a "nota".
##So, the respondents are being asked to give Pinochet a grade.
```

```
chile90$like.pinocho <- as.numeric(chile90$P11L > 3)  ##1 if likes Pinochet, 0 if not.
```

```
##OR

##p22 Con Cual posicion identifica o simpatiza Ud. mas?
##'1' DERECHA
##'2'   CENTRO DERECHA
##'3' CENTRO
##'4'   CENTRO IZQUIERDA
##'5' IZQUIERDA
##'6' INDEPENDIENTE
##'7' NINGUNA
##'8'   NO SABE
```

```
chile90$ideology <- ifelse(chile90$P22 == 8, NA, chile90$P22)  ##Code DK as missing for now
chile90$right.wing <- as.numeric(chile90$ideology <= 2)
```

```
(fit2 <- fit.please(milworry ~ age3 + right.wing, thedata = chile90))
## compare to fit1
fit1
```

"Controlling for" right-wing self-identification does not seem to make much difference in the estimated proportion of people who worry about the military. Right-wingers are less worried about the military than center- and left-wingers.

4. Now, let's simplify the model:

```
chile90$oldest <- ifelse(chile90$age3 == "55ymas", 1, 0)
table(chile90$oldest, chile90$age3, exclude = c())

(fit3 <- lm(milworry ~ oldest + right.wing, data = chile90))
```

5. Now, let's make sure that we understand what is going on here. I'm going to propose an alternative method that I claim will give you the same answer as above [this is tricky, see the hint]: (1) regress `milworry` on your ideology variable; (2) record the residuals from the `milworry~your.ideology` regression — these represent worry with the mean differences in worry by ideology removed (i.e. a form of mean deviation); (3) regress `oldest` on your chosen ideology variable ; (4) record the residuals from the `oldest~your.ideology.var` (these are age with mean differences in age due to ideology); (5) regress the residuals from `milworry~your.ideology` on the residuals from `oldest~your.ideology.var`. How might you explain the residuals from `milworry~your.ideology` regression? What is the scale of those residuals? *Hint:* Use `lm()` rather than `fit.please()` for this. I recommend using `resid()` on the `lm()` objects to get the residuals quickly and easily without having to calculate predicted values and then subtracting them from the observed values. Here are some recommended ways to run these regressions. Make sure you understand what is going on with these lines of code [swap your own ideology measure for "right.wing" if you want].

```
(fit3 <- fit.please(milworry ~ oldest + right.wing, thedata = chile90))

## Now here I use lm() because it has a nice residuals() function already defined for it.

(mwrw.fit <- lm(milworry ~ right.wing, data = chile90))
(agerw.fit <- lm(oldest ~ right.wing, data = chile90))
```

```
mwrw.e <- resid(mwrw.fit)
agerw.e <- resid(agerw.fit)

lm(mwrw.e ~ agerw.e)
```

Notice that we got the same coefficient of roughly .126 from this "residualization" procedure as we did using multiple regression (using either `lm()` or `fit.please()`).

Now, let's simplify this a bit. I think we got wrapped up in the subsetting and such before — all of which is good R practice but which may have obscured the statistical points. So, first I'm just going to look at a smaller version of the chile90 dataset — a version excluding the folks with missing values on right.wing and folks in the 25–64 age range

```
## Make a new dataset only for the youngest and oldest groups, and excluding missing values from the right.wing variable
chile90small <- subset(chile90, subset = age3 != "25-64" & !is.na(right.wing), select = c(milworry, age3, oldest, right.wing), drop
## The following command is equivalent to the subset() command chile90small<-chile90[chile90$age3!='25-64'&
## !is.na(chile90$right.wing), c('milworry','age3','right.wing'), drop=TRUE]

## Check our subsetting:
summary(chile90small)
## Annoying that the middle level of the factor lingers despite efforts to drop=TRUE it!
chile90small$age3 <- factor(chile90small$age3)
summary(chile90small)
```

Now do the two fits: the one looking at the mean difference in worry by age, and the other "controlling for" right.wing identification.

The difference in coefficient estimates between fit1 and fit1a arises from excluding the missing right.wing folks. The difference in estimates on right.wing between fit2 and fit2a arises from excluding the middle group Notice no difference in the estimate for the age effect between fit2 and fit2a

Calculate mean proportions of worry by the different age groups (i.e. the proportion for the excluded category of age and the differences between the excluded category and the other categories for the other categories)

```
(fit1a <- fit.please(milworry ~ age3, thedata = chile90small))

## Now, 'control for' right.wing.
(fit2a <- fit.please(milworry ~ age3 + right.wing, thedata = chile90small))

## Now calculate differences of means:
(worrybyage.means <- with(chile90small, tapply(milworry, age3 == "55ymas", mean)))

(worrybyrw.means <- with(chile90small, tapply(milworry, right.wing, mean, na.rm = TRUE)))

(agebyrw.means <- with(chile90small, tapply(age3 == "55ymas", right.wing, mean, na.rm = TRUE)))

## Diff of means in worry due to age [youngest vs. oldest] (equiv to fit1a)
(unadjdiff <- worrybyage.means[2] - worrybyage.means[1])
## I'm calling this the 'unadjusted' difference since this is the difference of substantive concern. How much of this difference i
## due to right-wing?

## Diff of means in worry due to right-wing [left/center vs. right] (equiv to lm(milworry~right.wing,data=chile90small))
worrybyrw.means[2] - worrybyrw.means[1]
```

So, we see, yet again, that the regression is giving us differences of means. But the idea of "adjusting" one difference of means for another one (or set of them) is where the real power of the linear model comes from and now I'll try to show a more direct (but more computationally burdensome) route that doesn't require least squares fitting at all.

First, subtract off the the extent to which the means in milworry differ by right.wing. This creates something I'm calling `mymwresid`.

```
## Here, I've got mean worry by right.wing from observed milworry from each person.
mymwresid <- with(chile90small, milworry - worrybyrw.means[as.character(right.wing)])
table(mymwresid)
## Notice that rather than just two values, we now have four:
with(chile90small, table(milworry, right.wing))

## And that this is the same as taking the residuals from the mwrw.fit regression
table(signif(resid(mwrw.fit), 5))
```

For, for folks with no worry but who are right.wing, we subtract off .179, but for folks without worry, but who are not right.wing, we substract off .307.

This, is what "worry without the effect of right.wing" or "worry without right.wing" means in the context of least squares:

Some of this difference of .127 due to right.wing is due to age. So, subtracting off this amount would be too much. We only want to take off the difference due to right.wing that is not related to age. So, how much of the difference in proportion "old" is due to right. wing? [Seems very strange to put it that way, huh? Not like announcing right.wing ideology (or center or left-wing ideology) can change your age. But, rather, we are asking how much ideology appears related to age — and by "related" we mean something very specific — the difference in means of proportion "old" between "right wing" and "not right wing" people.

```
## Diff of means in age due to right-wing. Right-wing folks are older [30% are in the oldest group versus 27% in the youngest
## group] (equiv to lm(I(age3=='55ymas')~right.wing,data=chile90small))
agebyrw.means[2] - agebyrw.means[1]
myageresid <- with(chile90small, (age3 == "55ymas") - agebyrw.means[as.character(right.wing)])

table(myageresid)
## or
table(signif(agerw.e, 5))
## Notice that rather than just two values, we now have four:
with(chile90small, table((age3 == "55ymas"), right.wing))
```

So, a small part of the mean difference in age can be due to differences in right.wingedness [where "due to" is not causal but merely descriptive of the sizes of mean differences].

Now, we can regress those residuals on each other to get the slope that we care about just as we did above:

```
## Using lm() because fit.please() demands a dataset
thelm <- lm(mymwresid ~ myageresid - 1)  ##Excluding intercept because I know it is zero
coef(thelm)
```

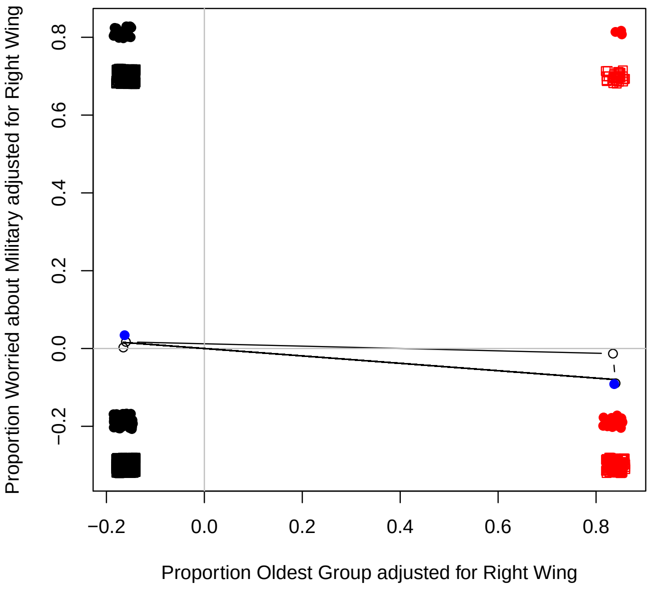
Now, notice that we can do this adjustment without least squares:

```
## Mean of milworry for each value of age [both residualized for right.wing]
(therwadjmeans <- tapply(mymwresid, myageresid, mean))

## Now, the least squares line smooths over those means just calculated. See the next plot: Least squares is like taking a weighte
## mean of the points for the young [the black points] versus the old [the red points]

## The weights are proportionate to the number of people at each value:
(therwadjns <- tapply(mymwresid, myageresid, length))

## Here are the weighted mean calculations. The adjusted worry among the young is a weighted average of the two means among the
## young [where each mean is separately calculated for the right.wing==1 and right.wing==0].

## So the adjusted worry among the young is:
(adj.worry.young <- (-0.0313 * (27/(27 + 203))) + (0.0429 * (203/(27 + 203))))

## And among the old it is:
(adj.worry.old <- (0.0705 * (12/(12 + 74))) + (-0.1177 * (74/(12 + 74))))

adj.worry.old - adj.worry.young
## And recall the least squares result from regressing residuals on residuals:
coef(thelm)
## Or the multiple regression result:
fit2a
```

Now the plot comparing the least squares line (the straight sloped black line), the line of means (the broken line with open circles), and the two adjusted means (in blue).

```
##I jitter the points to show how many people are in each group
##This makes the weighting that produces the least squares lines more clear
plot(jitter(myageresid,amount=.02),
     jitter(mymwresid,amount=.02),
     col=c("black","red")[(chile90small$age3=="55ymas")+1], ##red for old
     pch=c(0,19)[(chile90small$right.wing==1)+1], ##filled circle for right.wing
     xlab="Proportion Oldest Group adjusted for Right Wing",
     ylab="Proportion Worried about Military adjusted for Right Wing"
     )
```

```
##Now, plot the line of means --- this is the object of presumed substantive interest,
##the object which will be smoothed.
##Notice that "means" in this case, are "proportions" since
##a mean of a binary variable is a proportion of 1s.
lines(as.numeric(names(therwadjmeans)),
      therwadjmeans,type="b")

##Notice that because these are both residuals (i.e. observed minus
##some summary, here a mean), they both have means of zero:
summary(myageresid)
summary(mymwresid)
abline(v=0,h=0,lty=1,col="gray") ##Mark the means.

##This is the line from the least squares fit.
lines(fitted(thelm)~myageresid)

##Now, add the points that we calculated directly from the adjusted
##means.
points(mean(sort(unique(myageresid))[1:2]),
       adj.worry.young,pch=19,col="blue")
points(mean(sort(unique(myageresid))[3:4]),
       adj.worry.old,pch=19,col="blue")
```



So, we've seen how to "control for" ideology using multiple regression, and what this means: it means "taking away" the "relationship" between right.wing and worry about the military and the relationship between right.wing and age and then assessing the relationship between "worry without right.wing" and "age without right.wing". By "relationship" we've meant "difference of means" (and this has been made particularly clear because we've used binary variables — with continuous variables, "relationship" is the "smoothed difference of means" where the smoothing is done by least squares). And by "taking away" we've meant adjusting the values of the variables (subtracting means from them) such that the new variable would have no difference of means when compared with the adjustor or control variable.

This seems very complicated, and it is, if you do it by hand as we've done it here. What is cool about least squares is that it happens automagically and nearly instantaneously for any number of controls.

6. Now, did this residualization do what we really wanted it to do? Try "holding constant" more directly: run your regression separately for each subgroup on your ideology measure. Interpret the results of the two regressions. *Hint:* You can either use the `subset` argument to `lm()` or directly select the appropriate rows of the the chile90 dataframe for use in `fit.please()` by doing something like `chile90[my.ideology.var==somevalue,]`.

```
(fit.rw1 <- fit.please(milworry ~ age3, thedata = chile90, subset = right.wing == 1))
(fit.rw0 <- fit.please(milworry ~ age3, thedata = chile90, subset = right.wing == 0))
```

Wow! "Controlling for" didn't do what one might have thought. The older folks are *more* worried than the younger folks among the right.wing identifiers but *less* worried than the younger folks among the left/center-identifiers! The hidden assumption that ideology works the same way in the old as in the young [or equivalently, the assumption that age and worry about the military work the same way regardless of ideology] was incorrect.

7. So you've seen that `fit.please()` automagically does this regressing on residuals stuff. It turns out, that it can also handle this stratification version of "controlling for" too. Try estimating your regression using the following formula (adjusted for your own ideology variable): `milworry~oldest*your.ideology.var`. You should be able to recover the very same numbers from this regression as you did when you stratified. Interpret these results: what is the proportion who worry about the military among the old,right-wingers? what is the proportion who worry about the military among the young, right-wingers?

```
(fit4 <- fit.please(milworry ~ age3 * right.wing, thedata = chile90)[, 1])
## Proportion worried among youngest/not-right.wing
fit4["(Intercept)"]
## Proportion worried among youngest/right.wing
fit4["(Intercept)"] + fit4["right.wing"]
## Proportion worried among the oldest/right.wing
fit4["(Intercept)"] + fit4["age355ymas"] + fit4["right.wing"] + fit4["age355ymas:right.wing"]

## Difference in proportion worried between the oldest/right.wing and the youngest/not-right.wing
(fit4["(Intercept)"] + fit4["age355ymas"] + fit4["right.wing"] + fit4["age355ymas:right.wing"]) - (fit4["(Intercept)"])
## same as
fit4["age355ymas"] + fit4["right.wing"] + fit4["age355ymas:right.wing"]

## Difference in proportion worried between the oldest/right.wing and youngest/right.wing
(fit4["(Intercept)"] + fit4["age355ymas"] + fit4["right.wing"] + fit4["age355ymas:right.wing"]) - (fit4["(Intercept)"] + fit4["rig
## same as
fit4["age355ymas"] + fit4["age355ymas:right.wing"]

## Difference in proportion worried between oldest/right.wing and oldest/not-right wing
(fit4["(Intercept)"] + fit4["age355ymas"] + fit4["right.wing"] + fit4["age355ymas:right.wing"]) - (fit4["(Intercept)"] + fit4["age3
## Same as
fit4["right.wing"] + fit4["age355ymas:right.wing"]
```

So, we see the same results that we saw when we estimated the mean differences in worry separately for subgroups defined by right.wing: about 10% more of the older group is worried about the military among the right.wingers than the younger group.

This is often called an "interaction term" regression, or "interactions with dummy variables". It is a way to trick least squares into doing the subgroup-specific or stratified estimation all in one step: it works the same as the subgroup/stratification/subsetting approach when the subgroups are all binary or categorical. They are continuous, then `x*continuous.var` is not the same as `subset=continuous.var==somevalue` since the later breaks the continous variable into parts and the former assumes a smooth and linear interaction.

8. How much more prevalent is worry about the military among the oldest, right-wing group than it is among the youngest, right-wing group? *Hint:* First, you'll need to add coefficients to get the estimated proportion of worry about the military among each group. Second, you'll have to subtract those estimates (try subtracting the youngest group from the oldest group). Is there a simpler way to get this difference? What is it?

See above.

9. Use the bootstrap to produce a simple 95% percentile confidence interval for this difference. Plot your bootstrap estimated/approximated sampling distribution for this difference. *Hint:* Bootstrap the entire regression but now your addition and subtraction will be done on columns of the results matrix.

```
## First see how to get the value of interest:
fit5 <- coef(lm(milworry ~ age3 * right.wing, data = chile90))
fit5["age355ymas"] + fit5["age355ymas:right.wing"]  ##this sum of coefs is what we want
```

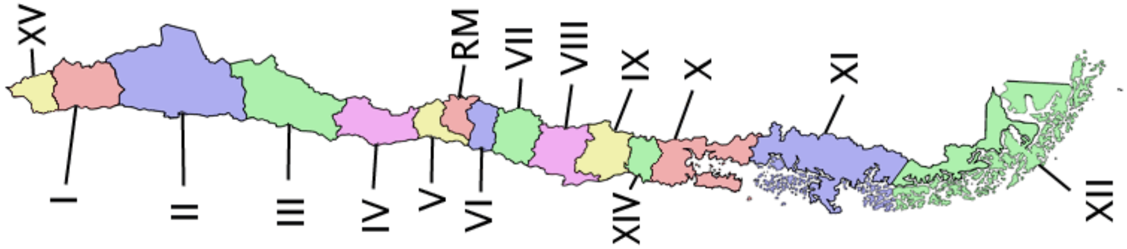
```
## Next bootstrap:
newfit.bs <- do(500) * lm(milworry ~ age3 * right.wing, data = resample(chile90))
## Now work directly with the sum that we care about, only now we are summing columns rather than scalars.
```

```
summary(newfit.bs$age355ymas + newfit.bs$"age355ymas:right.wing")

## The simple percentile bootstrap confidence interval
(bs.ci <- quantile(newfit.bs$age355ymas + newfit.bs$"age355ymas:right.wing", c(0.025, 0.975)))

## The estimated standard error of the sum of coefficients:
(bs.se <- sd(newfit.bs$age355ymas + newfit.bs$"age355ymas:right.wing"))
```

The estimated sampling distribution of the sum of the two coefficients is shown below (after the stratified sampling bootstrap).

So, the difference could plausibly range between -.2 and 4. OR, we wouldn't reject the null of no mean difference at the $\alpha = .05$ significance level (i.e. that null would have a p-value of greater than or equal to .05, making that null a relatively plausible story for our data).

10. Now, close inspection of the codebook tells us that, in fact, the survey sampling was done separately in four places in the country. First, the researchers chose 4 metro areas — not a sample of metro areas, just the 4 largest ones. Then, they sampled within those areas. What should we do to change our resampling procedure given this information? Now, let's create a simple 95% confidence interval for the proportion of the left-wing, youngest group worried about the military using this new, more appropriate sampling plan. Any difference? Why might there be a difference between the sampling distribution for the intercept taking into account the actual design and not? *Hint:* Use `resample(chile90,within=chile90$REGION)`. To think about why there might be differences across the regions look at `prop.table(table(chile90$milworry,chile90$REGION),margin=2)`. In Chile, by the way, 8th Region (Concepcion and Talcahuano) has long been a hot-bed of leftism and labor activism where as the North (the 2nd region) has tended to be more right-wing (despite plenty of copper and nitrate-mining labor activism there, too). (RM=Region 13).



```
chile90$regionF <- factor(chile90$REGION)

newfit.stratA <- do(500) * lm(milworry ~ age3 * right.wing, data = resample(chile90, within = chile90$REGION))
```

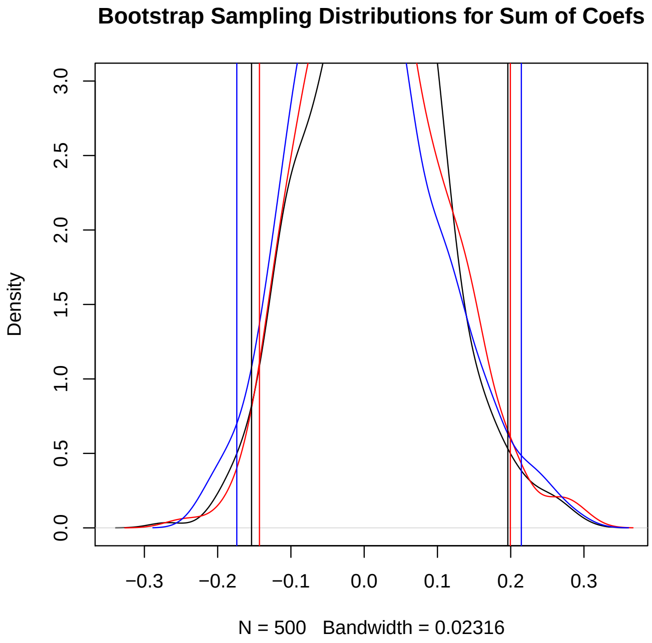
```
(bs.stratA.ci <- quantile(newfit.stratA$age355ymas + newfit.stratA$"age355ymas:right.wing", c(0.025, 0.975)))

## The estimated standard error of the sum of coefficients:
(bs.stratA.se <- sd(newfit.stratA$age355ymas + newfit.stratA$"age355ymas:right.wing"))

bs.ci
bs.se
```

Notice that including the information about the research design has some effect (a small one here) on the statistical inference: namely producing a sampling distribution that is narrower. Now, if we thought that worry about the military varied much across the regions, we'd also want to adjust for region in the regression by allowing different intercepts for each region:

```
## Now what about adding a 'fixed effect' aka indicator variabels, for region?
(fit.region <- lm(milworry ~ age3 * right.wing + regionF, data = chile90))

newfit.stratB <- do(500) * lm(milworry ~ age3 * right.wing + regionF, data = resample(chile90, within = chile90$REGION))
```

```
#### The simple percentile bootstrap confidence interval
(bs.stratB.ci <- quantile(newfit.stratB$age355ymas + newfit.stratB$"age355ymas:right.wing", c(0.025, 0.975), na.rm = TRUE))
## The estimated standard error of the sum of coefficients:
(bs.stratB.se <- sd(newfit.stratB$age355ymas + newfit.stratB$"age355ymas:right.wing", na.rm = TRUE))
```

```
plot(density(newfit.bs$age355ymas + newfit.bs$"age355ymas:right.wing"), ylim = c(0, 3), main = "Bootstrap Sampling Distributions f
lines(density(newfit.stratA$age355ymas + newfit.stratA$"age355ymas:right.wing"), col = "red")
lines(density(newfit.stratB$age355ymas + newfit.stratB$"age355ymas:right.wing"), col = "blue")

abline(v = bs.ci)
abline(v = bs.stratA.ci, col = "red")
abline(v = bs.stratB.ci, col = "blue")
```



So, taking the actual sampling design into account improved our statistical inference slightly, but adjusting the estimates for region did not appreciably make our confidence intervals shorter (or the standard error of our estimates smaller).

11. What do you conclude about the interaction of ideology and age as they matter for fears about the military in newly post-transition Chile?

    The older right wingers are more afraid than the young right wingers. The older left wingers are less afraid than the young left wingers. But, the left wingers are more afraid than the right wingers in general.

12. Finally, did we really succeed in ”controlling for” when we used the ”covariance adjustment” method? Produce a plot or table assessing the extent to which the results depend on the linear functional form (and thus interpolation and/or extrapolation as explained in the Gelman and Hill reading) versus are supported by data. *Hint:* I think that here a table would be fine and more informative than a plot since we are not ”controlling for” a continuous covariate as they did in the Gelman and Hill reading.

    When we use the full dataset, we do have data support for all of the relevant combinations. When we use the 100 person dataset, we cannot claim to control for ”right.wing” for the 35-54 year old group at all (none of them are right.wing), yet the regression still provides us numbers (because we assumed a global straight line fit).

```
chile90s$right.wing <- as.numeric(chile90s$ideology <= 2)

lm(milworry ~ age3 + right.wing, data = chile90s)
## But!
with(chile90s, ftable(milworry, age3, right.wing, col.vars = "milworry", row.vars = c("right.wing", "age3")))

lm(milworry ~ age3 + right.wing, data = chile90)
## But!
with(chile90, ftable(milworry, age3, right.wing, col.vars = "milworry", row.vars = c("right.wing", "age3")))
```

13. Would this problem get worse or better as we add controls? Or if we had smaller samples?

    Worse in each case.