# Introduction to Causal Inference

# Solutions to Problem Set 2

Professor: Teppei Yamamoto

**Due Saturday, July 16 (at beginning of class)**

Only the required problems are due on the above date. The optional problems will not directly count toward your grade, though you are encouraged to complete them as your time permits. If you choose not to work on the optional problems, use them for your self-study during the summer vacation.

## Required Problems

### Problem 1

This problem continues our introduction to R for those of you who are just getting started with the language. *If you chose to solve Problem 2 in the previous problem set, solve this problem.* If you skipped it, this problem is optional for you; however, you may still want to solve it if you are unfamiliar with the `ggplot2` package.

a) Complete Lesson 2 on this website. Again, you should watch each of the video tutorials (Parts 2.1 to 2.7) and complete all the `swirl` quizzes. You should use the `savehistory()` function to save the log of your work and submit it along with the rest of your problem set.

### Problem 2

Download from the course website the data from "Monitoring Corruption: Evidence from a Field Experiment in Indonesia" by Benjamin A. Olken. The paper evalutes an effort to reduce corruption in road building projects in Indonesia. One of the treatments, which we focus on here, "sought to enhance participation at accountability meetings, the village-level meetings in which project officials account for how they spent project funds." Before construction began, residents in treated villages were encouraged to attend these meetings.

The outcome of interest to us is `pct.missing`, the difference between what officials claimed they spent on road construction and an independent measure of expenditures. Treatment status is given by `treat.invite`, which takes a value of 1 if the village received the intervention and 0 if it did not. There are four pre-treatment covariates in the data: `head.edu`, the education of the village head; `mosques`, mosques per 1,000 residents; `pct.poor`, the percentage of households below the poverty line; and `total.budget`, the budget for each project.

a) Create a balance table. For each pre-treatment covariate, include comparisons for treated and untreated units in terms of the mean and standard deviation. Report a test, for each covariate, of the hypothesis that the difference in means between treatment conditions is zero.

```r
library(xtable)
d = read.csv("olken.csv")
vars = subset(d, select = c("head.edu", "mosques", "pct.poor",
    "total.budget"))

# Get mean and SD by treatment status for each covariate
bal.mean = aggregate(vars, by = list(d$treat.invite), mean)
bal.sd = aggregate(vars, by = list(d$treat.invite), sd)
# Test the difference in means between conditions for each
# covariate; keep the p-values
bal.pv = sapply(1:4, function(x) t.test(vars[d$treat.invite ==
    1, x], vars[d$treat.invite == 0, x])$p.value)

# Stack, rearrange, and label the balance table
bal = rbind(bal.mean, bal.sd, c(NA, bal.pv))
bal = t(bal)
bal = bal[-1, c(1, 3, 2, 4, 5)]
colnames(bal) = c("Control Mean", "Control SD", "Treat Mean",
    "Treat SD", "ttest p-val")
```

```r
print(xtable(bal, caption='Covariate Balance'), caption.placement='top')
```

Table 1: Covariate Balance

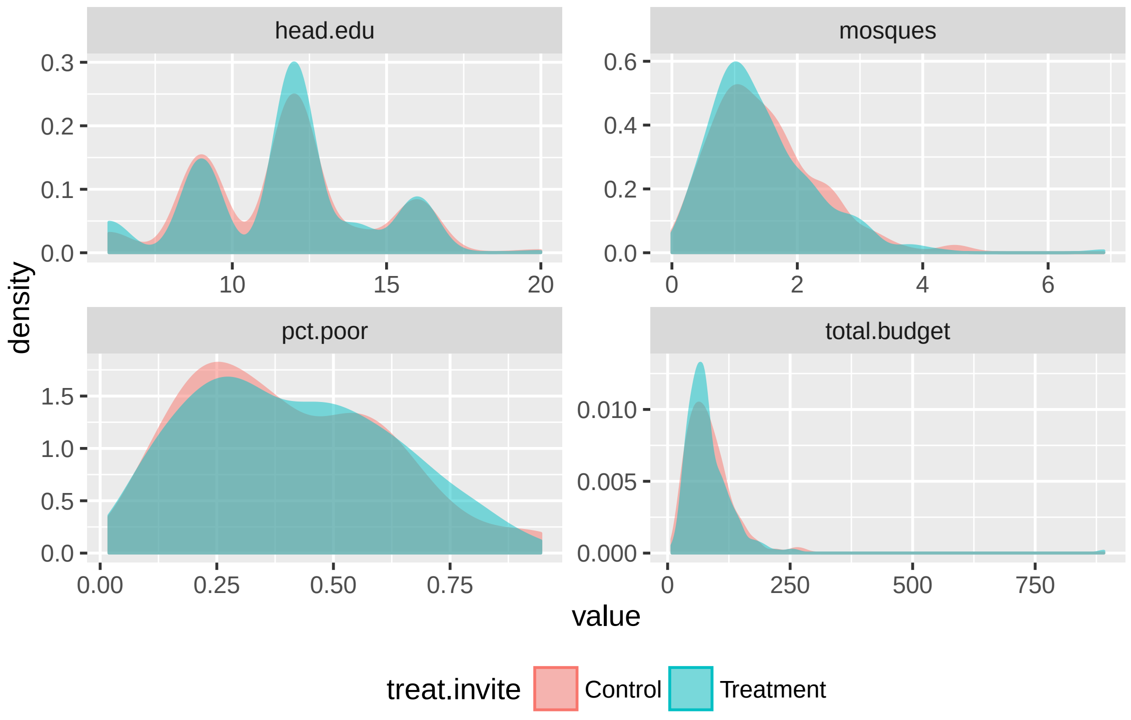
| | Control Mean | Control SD | Treat Mean | Treat SD | ttest p-val |
|---|---|---|---|---|---|
| head.edu | 11.58 | 2.72 | 11.55 | 2.72 | 0.89 |
| mosques | 1.47 | 0.83 | 1.42 | 0.84 | 0.51 |
| pct.poor | 0.40 | 0.21 | 0.41 | 0.21 | 0.62 |
| total.budget | 83.35 | 42.91 | 83.16 | 60.41 | 0.97 |

b) For each covariate, plot its distributions under treatment and control (side by side or overlaying). Include the plots in your writeup.

```r
library(ggplot2)

## Warning:  package 'ggplot2' was built under R version 3.2.3
```

```r
library(reshape2)
d$treat.invite = factor(d$treat.invite, labels = c('Control', 'Treatment'))
# Melt the data for easy plotting with facets in ggplot
d.melt = melt(d, id.vars = c('id', 'treat.invite', 'pct.missing'))
ggplot(d.melt, aes(x=value, fill=treat.invite, color=treat.invite)) +
  geom_density(alpha=.5) +
  facet_wrap(~ variable, scales='free') +
  theme(legend.position='bottom')
```



c) With reference to your table in a) and plots in b), do villages in each condition appear similar in the pre-treatment covariates? Explain the importance of checking balance in a randomized experiment and the result you typically expect to find.

Looking at both our table and our plots, there aren't any obvious imbalances on pre-treatment covariates, with the exception of higher variance in `total.budget` among treated units. Our motivation for checking is that under random assignment, treatment and pre-treatment covariates are independent, so we expect the same distribution of covariates in each condition; imbalance might suggest that random assignment wasn't successful. Of course, we could also be unlucky; we'd want to know about that too, for reasons we explore below.

d) Regress treatment on the pre-treatment covariates and report the results. What do you conclude?

```
library(stargazer)
library(sandwich)
d = read.csv('olken.csv')
mod.propensity = lm(treat.invite ~ head.edu + mosques + pct.poor
                    + total.budget, data = d)
#se.propensity  = sqrt(diag(vcovHC(mod.propensity , type='HC2')))

stargazer(mod.propensity, title='OLS Results: Propensity for Treatment',
          style='apsr', keep.stat=c('n', 'f'))
```

Table 2: OLS Results: Propensity for Treatment

| | treat.invite |
|---|---|
| head.edu | −0.001 |
| | (0.008) |
| mosques | −0.019 |
| | (0.026) |
| pct.poor | 0.057 |
| | (0.104) |
| total.budget | −0.00005 |
| | (0.0004) |
| Constant | 0.677*** |
| | (0.117) |
| N | 472 |
| F Statistic | 0.191 (df = 4; 467) |

*p < .1; **p < .05; ***p < .01

This is another balance check: it is possible to predict treatment status based on the set of covariates? We have little reason to think so based on the F test. (Recall that the F test reported here is for the null hypothesis that *all* parameters are zero.) But note that we applied a model of our choice to the data. Treatment could be some non-linear or higher-order function of the covariates.

e) Using the difference-in-means estimator, estimate the average treatment effect and its standard error.

```
# Apply the difference in means estimator
ybar = tapply(d$pct.missing, list('treated'=d$treat.invite), mean)
ybar['1'] - ybar['0']
```

1 -0.02494953

```r
# Estimate the standard error of the difference in means
seDiffMeans = function(y, tx){
  y1 = y[tx == 1]
  y0 = y[tx == 0]
  n1 = length(y1)
  n0 = length(y0)
  sqrt(((var(y1)/n1 + var(y0)/n0)))
}
seDiffMeans(d$pct.missing, d$treat.invite)
```

[1] 0.03310019

f) Now estimate the average treatment effect and its standard error using a simple regression of outcomes on treatment. Are the results different from those in e)? Make the changes necessary for them to match exactly, and explain your method.

```r
mod.bivariate = lm(pct.missing ~ treat.invite, data=d)
#Straight from LM
stargazer(mod.bivariate, title='OLS Estimates: Average Treatment Effect',
          style='apsr', keep.stat=c('n'), digits=5)
```

Table 3: OLS Estimates: Average Treatment Effect

| | pct.missing |
|---|---|
| treat.invite | −0.02495 |
| | (0.03346) |
| Constant | 0.25277*** |
| | (0.02716) |
| N | 472 |

*p < .1; **p < .05; ***p < .01

```r
se.bivariate = sqrt(diag(vcovHC(mod.bivariate, type='HC2')))
#With HC2
stargazer(mod.bivariate, se=list(se.bivariate),
          title='OLS Estimates: Average Treatment Effect, HC2 Robust SEs',
          style='apsr', keep.stat=c('n'), digits=5, notes = "HC2 Robust SEs")
```

The point estimates are the same. The standard error reported by `lm` is close to the estimate in e) but somewhat higher. In e), we allowed for the possibility that the sample

Table 4: OLS Estimates: Average Treatment Effect, HC2 Robust SEs

| | pct.missing |
|---|---|
| treat.invite | −0.02495 |
| | (0.03310) |
| Constant | 0.25277*** |
| | (0.02655) |
| N | 472 |

*p < .1; **p < .05; ***p < .01
HC2 Robust SEs

variance differs between treated and control units, while standard errors computed and reported by default from `lm()` assume homoskedasticity. The result from the HC-2 variant of the homoskedasticity-robust estimator is identical to the result in e).

g) Reestimate the average treatment effect using a regression specification that includes pre-treatment covariates (additively and linearly). Report your estimates of the treatment effect and its standard error. Do you expect them to differ from the difference-in-means estimates, and do they? Explain.

```r
mod.adjusted <- lm(pct.missing ~ treat.invite + head.edu + mosques +
    pct.poor + total.budget , data=d)
se.adjusted = sqrt(diag(vcovHC(mod.adjusted, type='HC2')))

stargazer(mod.adjusted, se=list(se.adjusted), style="apsr", keep.stat = c("n"),
  title="OLS Estimates: Average Treatment Effect with Covariate Adjustment",
  notes = "HC2 Robust SEs", digits=5)
```

Both estimates are nearly unchanged. Under random assignment to treatment, the unconditional estimate of the average treatment effect will be unbiased, but efficiency gains are possible through covariate adjustment, as we see in the very slight reduction in the standard error in this model. However, conditioning on the pre-treatment covariates via regression risks bias due to functional-form misspecification. This is a bias-variance trade-off.

h) (**Optional**) Estimate the ATE for villages with more than half of households below the poverty line, and then do the same for villages with less than half of households below the poverty line. Estimate the standard error of the *difference* in treatment effects and test the null hypothesis that there is no difference between them. What do you conclude?

Estimating the difference in average treatment effects is straightforward.

Table 5: OLS Estimates: Average Treatment Effect with Covariate Adjustment

| | pct.missing |
|---|---|
| treat.invite | −0.02642 |
| | (0.03263) |
| head.edu | −0.00551 |
| | (0.00615) |
| mosques | −0.04819*** |
| | (0.01844) |
| pct.poor | −0.11771 |
| | (0.07337) |
| total.budget | 0.00053* |
| | (0.00032) |
| Constant | 0.39045*** |
| | (0.09374) |
| N | 472 |

*p < .1; **p < .05; ***p < .01
HC2 Robust SEs

```r
# Get mean Y for combinations of treatment and pct.poor > .5
ybar.tab = tapply(d$pct.missing, list(treated = d$treat.invite,
    poor = d$pct.poor > 0.5), mean)
# Estimate the conditional ATEs for poor and not as the
# differences across treatment status
ate.x = ybar.tab["1", ] - ybar.tab["0", ]
ate.x
```

FALSE TRUE 0.0003555733 -0.0738740968

We can estimate the standard error of the difference in average treatment effects as

$$[\mathbb{V}(\hat{\tau}_{rich}) + \mathbb{V}(\hat{\tau}_{poor})]^{\frac{1}{2}} .$$

```r
se.poor1 <- seDiffMeans(d$pct.missing[d$pct.poor > .5],
  d$treat.invite[d$pct.poor > .5])
se.poor0 <- seDiffMeans(d$pct.missing[!d$pct.poor > .5],
  d$treat.invite[!d$pct.poor > .5])
se.diff <- sqrt(se.poor1^2 + se.poor0^2)
se.diff
```

[1] 0.06811946

Calculating a t-statistic is appropriate because the difference between two differences in sample means should converge in distribution to the normal under the central limit theorem.

```r
# Compute a t statistic
unname((ate.x["TRUE"] - ate.x["FALSE"])/se.diff)
```

```
[1] -1.089699
```

```r
# Refer to the null distribution
qt(0.025, ncol(d) - 2, lower.tail = FALSE)
```

```
[1] 2.570582
```

We fail to reject the null of no difference: $|t_{\text{observed}}| < t_{\text{critical}}$. Thus we cannot conclude that there may be heterogeneity in the ATE, conditional on pre-existing poverty in villages.

Using Huber-White heteroskedastic-robust standard errors in the regression framework gives the same result.

```r
d$majority.poor = d$pct.poor > 0.5
mod.subgroup = lm(pct.missing ~ treat.invite * majority.poor,
    data = d)
se.subgroup = sqrt(diag(vcovHC(mod.subgroup, type = "HC2")))
stargazer(mod.subgroup, se = list(se.subgroup), style = "apsr",
    keep.stat = c("n"), title = "OLS Estimates: Conditional ATEs",
    notes = "HC2 Robust SEs", digits = 5)
```

Table 6: OLS Estimates: Conditional ATEs

| | pct.missing |
|---|---|
| treat.invite | 0.00036 |
| | (0.04190) |
| majority.poor | 0.00636 |
| | (0.05324) |
| treat.invite:majority.poor | −0.07423 |
| | (0.06812) |
| Constant | 0.25059*** |
| | (0.03454) |
| N | 472 |

*p < .1; **p < .05; ***p < .01
HC2 Robust SEs

## Optional Problems

## Problem A

Consider a field experiment that first uses covariates to divide units into blocks and then employs complete randomization within each block. Units $1, \ldots, N$ were assigned to a single treatment and control under complete randomization in blocks $j$. $J_i$ is a random variable indicating which of the $M$ blocks unit $i$ belongs to ($J_i \in \{1, \ldots, M\}$). Let the probability of treatment be constant across blocks.

a) Denote the potential outcome under treatment of unit $i$ in block $j$ with $Y_{1ij}$, and the corresponding potential outcome under control $Y_{0ij}$. What is the identification assumption that allows us to identify the ATE in this experimental setup?

   Our identification assumption is: $Y_{1ij}, Y_{0ij} \perp\!\!\!\perp D_i | J_i$, which is guaranteed by random assignment of treatment within blocks.

b) Now, using the identification assumption you wrote down in a), show that the ATE ($\tau_{ATE}$) is identified. (Hint: You'll need to rewrite the ATE to account for the existence of $J_i$.)

$$\begin{aligned}
\tau_{ATE} &= \mathbb{E}[Y_{1ij} - Y_{0ij}] \\
&= \mathbb{E}_J[\mathbb{E}[Y_{1ij}|J_i = j] - \mathbb{E}[Y_{0ij}|J_i = j]] \\
&= \mathbb{E}_J\left[\mathbb{E}[Y_{1ij}|J_i = j, D_i = 1] - \mathbb{E}[Y_{0ij}|J_i = j, D_i = 0]\right] \\
&= \mathbb{E}_J\left[\mathbb{E}[Y_{ij}|J_i = j, D_i = 1] - \mathbb{E}[Y_{ij}|J_i = j, D_i = 0]\right]
\end{aligned}$$

   We apply the law of iterated expectations to write the ATE as the expectation, over blocks, of the expectation of outcomes conditional on block. In the third line, we rely on random assignment within blocks such that $Y_{1ij}, Y_{0ij} \perp\!\!\!\perp D_i | J_i$. This is our identifying assumption. We use it to write each expectation of potential outcomes conditional on block as conditional on treatment status. (Recall that $\mathbb{E}[A] = \mathbb{E}[A|B]$ if $A \perp\!\!\!\perp B$.) All quantities are then observable, so the ATE is identified under blocking.

c) Must the following statements be true in the above scenario? Why?

   i) $J_i \perp\!\!\!\perp D_i$ : Yes. So long as the probability of treatment is constant across all blocks, knowing a unit's treatment status will tell you nothing about their block.

   ii) $Y_i \perp\!\!\!\perp D_i$ : No, not if treatment has any effect at all. There is usually dependence between observed outcomes of interest to us and treatment status!

   iii) $Y_{di} \perp\!\!\!\perp D_i$ : Yes. Random assignment ensures that potential outcomes are independent of treatment status.

iv) $Y_{di} \perp\!\!\!\perp J_i$ : No, it is very likely that block status is associated with potential outcomes. Remember, we block on covariates, and if those covariates are associated with potential outcomes, then block status will be too.

d) Thinking about your own research interests, come up with a scenario where blocking before randomizing treatment might be useful. Say you have a sample of 20 units. Tell us your outcome variable of interest, the treatment you would be randomly assigning, and the variable(s) on which you would like to block. Briefly explain the benefits to blocking in this design. Are there any challenges to blocking?

Blocking balances covariates, instead of leaving it up to chance to balance these covariates across treatment and control groups. This is especially critical in small samples. Blocking also tends to reduce sampling variability when we have covariates that we think might predict outcomes and improves precision of the ATE estimate. Blocking provides the biggest gains in precision when the variables used to form blocks strongly predict outcomes. *Block what you can, randomize what you cannot. (Box)*

However, sometimes it is not possible to block on covariates, for instance if we do not have access to our sample before the point of randomization. Furthermore, without access to similar data or studies it is often difficult to anticipate what variables might strongly predict outcomes. Blocking may also be more complicated for implementation partners to carry out, but is generally good practice!

e) You explain your research design to your friend who says "You don't need to block on those variables. You can just control for those covariates in a regression after you randomize, run the experiment and collect your data. Covariate adjustment is just as good as blocking!" Is he right or wrong? What concerns might you have with your friend's approach?

He is wrong. Pre-randomization blocking is always better than post-randomization stratification (controlling for covariates in the regression) with respect to unbiasedness and precision. With respect to unbiasedness, blocking will perfectly balance the covariates that are used to form blocks and you don't need to know the functional form. With respect to precision blocking will make $X$ (the covariates on which we block) and $D$ completely independent so there will be less variability in estimated causal effects over repeated randomization.

Like blocking, covariate adjustment in regression balances pre-treatment variables across treatment and control groups. But we have reason to be concerned with using this approach. As was explained in lecture, including covariates in our regression to estimate average treatment effects may improve precision but can also induce bias in our estimates. This bias may be due to model misspecification because post-randomization covariate adjustment with regression requires making assumptions of how the covariates relate to the outcome variable.

Covariate adjustment can also lead down the dangerous road of "p-hacking". If post-randomization covariate adjustment must be used it may be a good idea to produce a pre-analysis plan to make your intentions completely transparent. It is always a good idea to report estimates of your ATE with and without covariate adjustments. Blocking on covariates removes the need to interpret results that differ between estimating the average treatment effect with and without covariate adjustment.

## Problem B

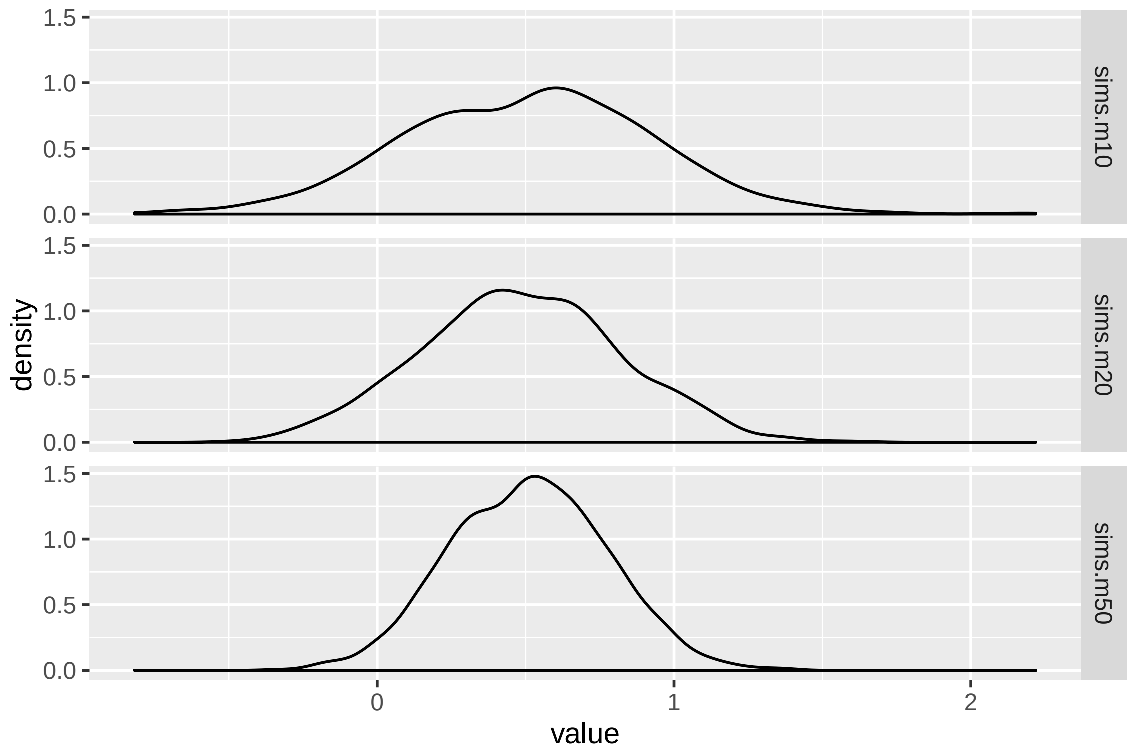
In some research contexts, assigning treatment at the unit level is impossible. Researchers may instead assign clusters of units to treatment. Here, we use simulation to examine the precision of estimates under this design. Consider the following scenario. We have units $i$ in $1, \ldots, N$ grouped in clusters $j$ in $1, \ldots, M$, where each cluster $j$ has size $N/M$. Treatment $D_i \in \{0, 1\}$ is assigned via complete randomization at the level of clusters: in $M/2$ clusters, all units are treated while in the remaining clusters, no units take treatment.

a) Create a dataset that matches this description in `R`, as follows. Create a vector of $j_i$ for $N = 100$ so that there are 10 clusters, and randomly assign five to treatment and the rest to control. Induce intracluster correlation by drawing group means $\mu_{dj}$ for each cluster: $\mu_{0j} \sim \mathcal{U}(-1, 1)$ and $\mu_{1j} \sim \mathcal{U}(-0.5, 1.5)$. Then draw values for $y_{0i}$ from $\mathcal{N}(\mu_{0j}, 1)$ and for $y_{1i}$ from $\mathcal{N}(\mu_{1j}, 1)$. Using $D_i$, create a vector of realized outcomes $y_i$. Now, wrap your work in a function and for each of 1,000 iterations, estimate the average treatment effect, forming a sampling distribution for the ATE. Repeat all of these steps for $M = 20$ and $M = 50$, and plot the three resulting sampling distributions.

```r
library(ggplot2)
library(reshape2)
simCluster = function(m, n = 100) {
    # create a 100-vector that indexes j in 1:M
    j = cut(1:n, m, labels = FALSE)
    # for 1:M, assign half the clusters to treatment; create a treatment
    # indicator by comparison of j to the result of the assignment
    d = j %in% sample(1:m, m/2)
    # draw group means from the uniform
    mu.0j = runif(m, -1, 1)
    mu.1j = runif(m, -0.5, 1.5)
    # draw values of y_di with mean \mu_dj
    y0 = rnorm(n, mu.0j[j])
    y1 = rnorm(n, mu.1j[j])
    # realize potential outcomes by treatment status
    y = y1 * d + (1 - d) * y0
    # with clusters of equal size, the difference in cluster means is the
    # mean difference over units; just take that
    mean(y[d]) - mean(y[!d])
}

# Simulate for 10, 20, and 50 clusters
sims.m10 = replicate(1000, simCluster(10))
sims.m20 = replicate(1000, simCluster(20))
sims.m50 = replicate(1000, simCluster(50))

# Plot
d = data.frame(sims.m10, sims.m20, sims.m50)
d.melt = melt(d)
ggplot(d.melt, aes(x = value)) + geom_density() + facet_grid(variable ~
    .)
```

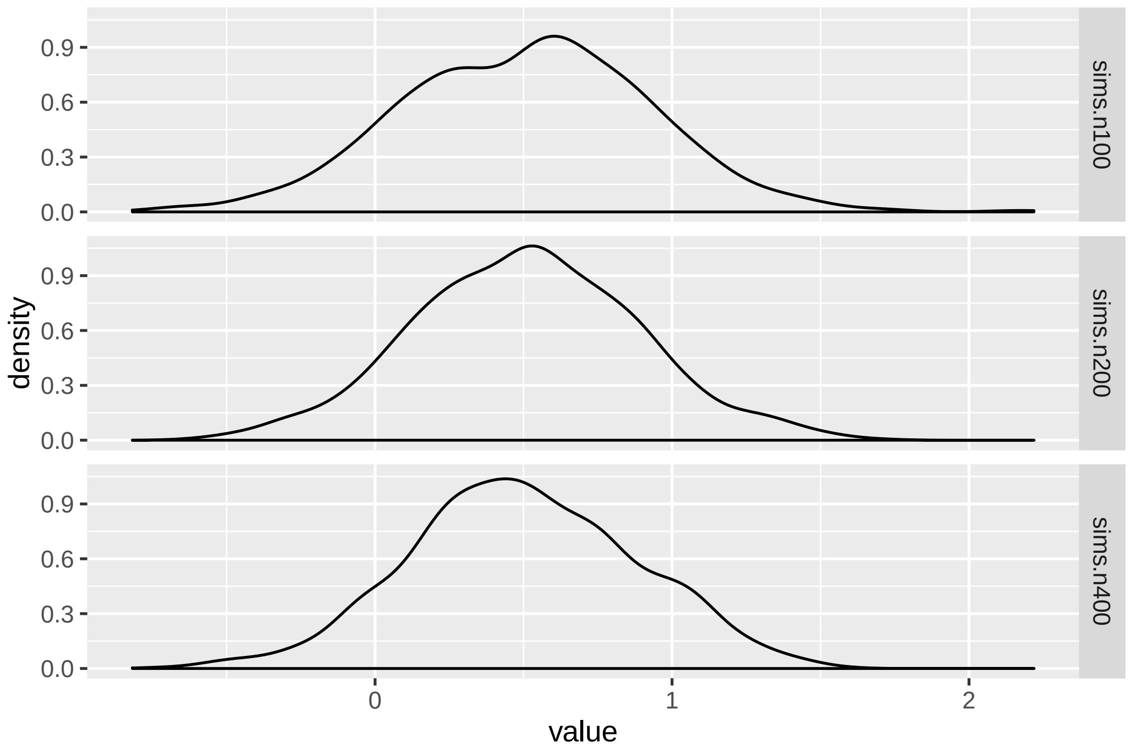


b) Explain what happens to the distribution of estimates as $M$ increases, holding constant $N$.

The sampling variance shrinks with each increment of $M$. $N$ is fixed; we observe this result because units within clusters are similar to each other in terms of $y_{di}$.

c) Repeat your simulation for $N = 200$ and $N = 400$ with $M = 10$ and plot the results. Given 100 units in 10 clusters, are there larger gains to precision from doubling the sample size or the number of clusters?

```
# repeat for m=10 and n=200, 400
sims.n100 = sims.m10
sims.n200 = replicate(1000, simCluster(m = 10, n = 200))
sims.n400 = replicate(1000, simCluster(m = 10, n = 400))
d.n = data.frame(sims.n100, sims.n200, sims.n400)
d.n.melt = melt(d.n)
ggplot(d.n.melt, aes(x = value)) + geom_density() + facet_grid(variable ~
    .)
```



```
# compare the standard deviations of the sampling distributions in the
# question
sd(sims.n200)
```

[1] 0.378297

```
sd(sims.m20)
```

[1] 0.3350719

Based on the simulation results, we can expect better precision in our scenario with 100 units in 20 clusters than with 200 units in 10 clusters.

d) Is this result an artifact of our scenario, or does it generalize? Explain with reference to Equations 3.4 and 3.22 from Gerber and Green (2012) for the standard error of the ATE under complete random assignment without clustering and under cluster random assignment.

The definitions show that the result is general. Without clustering, the variance decreases in $N$, while under cluster random assignment, it decreases in $k$.