

PSET 3

Sarah Van Alsten

2/4/2020

Question 1

```
#a.  
#function to randomly assign treatment (P = 0.5) given 2 vectors of  
#potential outcomes  
  
treatFx <- function(v1, v0){  
  
  #randomly assign treatment, 1 = treated, 0 = control  
  #number in sample will be = to length of v1  
  treat <- rbinom(n = length(v1), size = 1, prob = 0.5)  
  
  #create vector for obs outcomes under treatment: keep  
  #outcomes if treated, otherwise not observed  
  yi1 <- ifelse(treat == 1, v1, NA)  
  
  #create vector for obs outcomes under control: keep  
  #outcomes if control, otherwise not observed  
  yi0 <- ifelse(treat == 0, v0, NA)  
  
  #calculate ATE: mean yi1 - mean yi0  
  ate <- mean(yi1, na.rm = T) - mean(yi0, na.rm = T)  
  
  #t test  
  tres <- t.test(yi1, yi0)  
  
  #return pvalue for t test  
  return(tres$p.value)  
}
```

```
#1 and 2  
set.seed(02139)  
  
#outcomes under control: mean = 1, var/sd = 1  
y0 <- rnorm(n = 1000, mean = 1, sd = 1)  
  
#ate of outcomes under treatment: mean = t = 0.2; var/sd = 1  
y1 <- rnorm(n = 1000, mean = 0.2, sd = 1)  
  
#add to potential outcome under control  
y1 <- y0 + y1  
  
#run function  
treatFx(y1, y0)
```

```
## [1] 0.00369171
```

The p value for the given set up is $\sim 5.28 \times 10^{-33}$, indicating that it is very unlikely that the expected value of observed outcomes in the treatment and control groups are the same. In other words, the average treatment effect is non-zero and statistically significant.

```
#b.  
#repeat process 10,000 times  
reps.10k.02 <- replicate(n = 10000,  
                        expr = treatFx(v1 = y1, v0 = y0),  
                        simplify = "vector")  
  
#get % p values < 0.05  
sum(reps.10k.02 < 0.05) / (length(reps.10k.02)) #85.74% < 0.05
```

```
## [1] 0.8574
```

85.74% of calculated p-values in the simulation were less than 0.05. This number gives us the power of our experiment (how often do we correctly reject the null given that the null is true).

```
#c.  
#function that does repeated simulation, which can be called by map function  
#m1 = treatment effect  
replicateSimFx <- function(times = 1000, size = 1000, m1){  
  
  #generate treatment vector given number observation(size), and mean (m1)  
  v1 <- y0 + rnorm(n = size, mean = m1, sd = 1)  
  
  #do simulation for 'times' times, with previously generated v0 and our new v1  
  res <- replicate(n = times,  
                  expr = treatFx(v0 = y0, v1 = v1),  
                  simplify = "vector")  
  
  #return % of values < 0.05  
  return(sum(res < 0.05) / (length(res)))  
}  
  
#run simulation for different values of treatment group mean (0 - 1, along increases of .1),  
#1000 reps each per args in replicateSimFx  
results <- purrr::map_dbl(seq(0, 1, .1), ~replicateSimFx(m1 = .x))  
  
#plot results  
as.data.frame(cbind(results, seq(0, 1, .1))) %>%  
  ggplot(aes(x = V2, y = results)) +  
  geom_point() +  
  geom_line() +  
  theme_minimal() +  
  labs(x = "Average Treatment Effect", y = "Power") +  
  ggtitle("Power for Various Treatment Effect Sizes")
```

We have 100% or approximately 100% power to detect a significant ATE when the ATE is 0.4 or greater (as compared to the expected value of 1 in the control group; thus, expected value of 1.4 or more in the

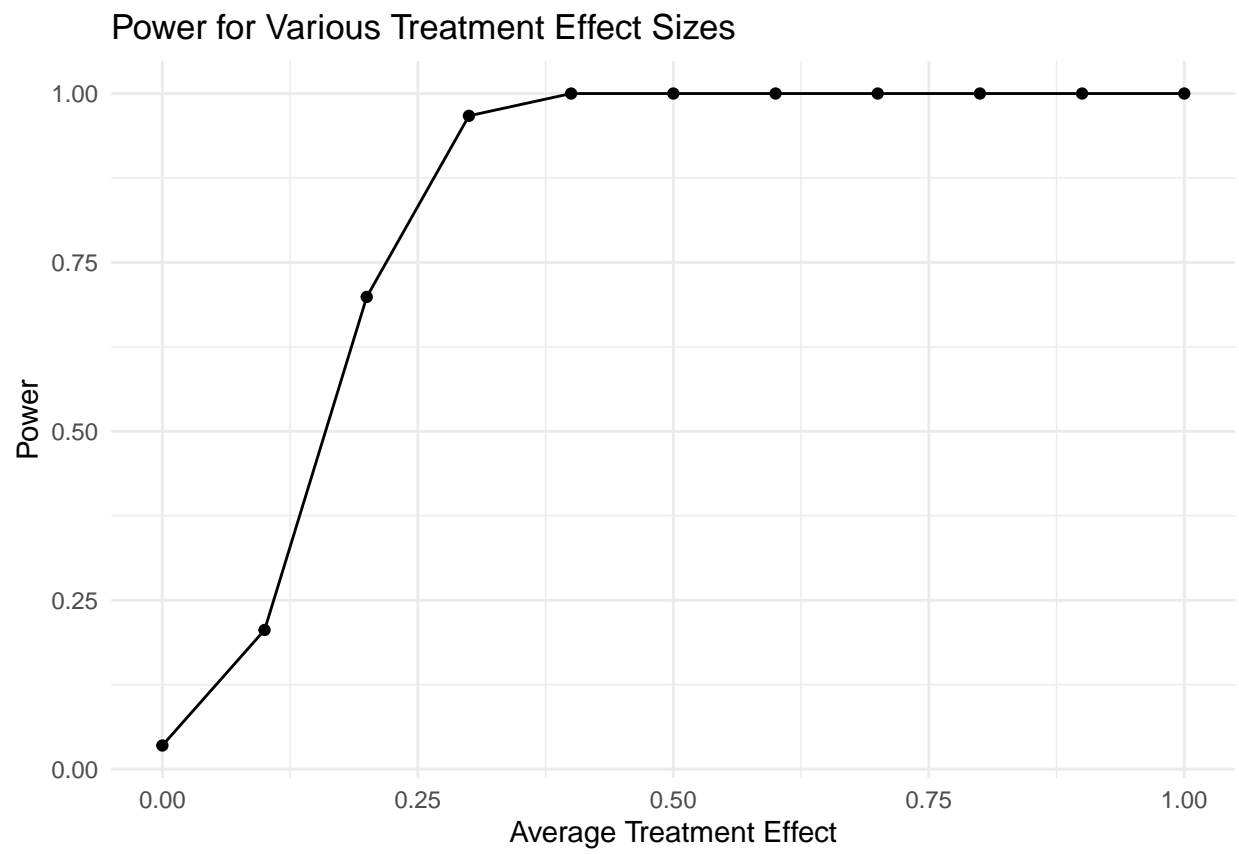


Figure 1: Power Plot

treated). We still have about 70% power to detect a significant ATE when the ATE is 0.3, but for effect sizes smaller than this, power declines rapidly. This tells us that (for a given sample size) we have better ability to detect the ‘true’ ATE when the difference in outcomes between the treatment and control groups is greater (ie when we have a larger effect size).

Draw one random sample of 50 units from your population without replacement, then repeat the process in part c) using only that sample. Do the same for sample sizes 100, 200, and 500. Plot lines for each sample size on the same graph with your result from part c) and clearly label the different lines. What do you notice? What does this tell you about the relationship between sample size and your ability to detect treatment effects for a given “true” ATE? Why might this be the case?

```
#d.
#add in sampling of diff sizes to previous function: sample.n = sample size, size = popl size
replicateSimFx2 <- function(times = 1000, size = 1000, m1, sample.n){

  #generate treatment vector given #obs(size), and mean (m1)
  v1 <- y0 + rnorm(n = size, mean = m1, sd = 1)

  #generate random sample of size sample.n
  v1.samp <- sample(v1, size = sample.n, replace = FALSE)

  #do simulation for 'times' times, with previously generated v0 and our new v1.samp
  #bc # in sample = length of v1, adjusting size of v1 only accounts for smaller
  #sample size
  res <- replicate(n = times,
                   expr = treatFx(v0 = y0, v1 = v1.samp),
                   simplify = "vector")

  #return % of values <0.05
  return(sum(res < 0.05) / (length(res)))
}

#replicate with sample size 50, various effect sizes 0-1
results.50 <- purrr::map_dbl(seq(0, 1, .1), ~replicateSimFx2(m1 = .x, sample.n = 50))

#draw random sample of 100 and repeat
results.100 <- purrr::map_dbl(seq(0, 1, .1), ~replicateSimFx2(m1 = .x, sample.n = 100))

#draw random sample of 200 and repeat
results.200 <- purrr::map_dbl(seq(0, 1, .1), ~replicateSimFx2(m1 = .x, sample.n = 200))

#draw random sample of 500 and repeat
results.500 <- purrr::map_dbl(seq(0, 1, .1), ~replicateSimFx2(m1 = .x, sample.n = 500))

#add these lines to graph
as.data.frame(cbind(results,
                    results.50,
                    results.100,
                    results.200,
                    results.500,
                    seq(0, 1, .1))) %>%
  ggplot(aes(x = V6)) +
```

```

#sample size 50
geom_point(aes(y = results.50, color = "50")) +
geom_line(aes(y = results.50, color = "50")) +
#sample size 100
geom_point(aes(y = results.100, color = "100")) +
geom_line(aes(y = results.100, color = "100")) +
#sample size 200
geom_point(aes(y = results.200, color = "200")) +
geom_line(aes(y = results.200, color = "200")) +
#sample size 500
geom_point(aes(y = results.500, color = "500")) +
geom_line(aes(y = results.500, color = "500")) +
#sample size 1000
geom_point(aes(y = results, color = "1000")) +
geom_line(aes(y = results, color = "1000")) +
theme_minimal() +
labs(x = "Average Treatment Effect", y = "Power") +
ggtitle("Power for Various Treatment Effect Sizes + Sample Sizes") +
#add color legend
scale_color_manual(labels = paste0("n = ", c(100, 1000, 200, 50, 500)),
                  values = c("red", "black", "orange", "blue", "forestgreen"),
                  name = "Sample Size")

```

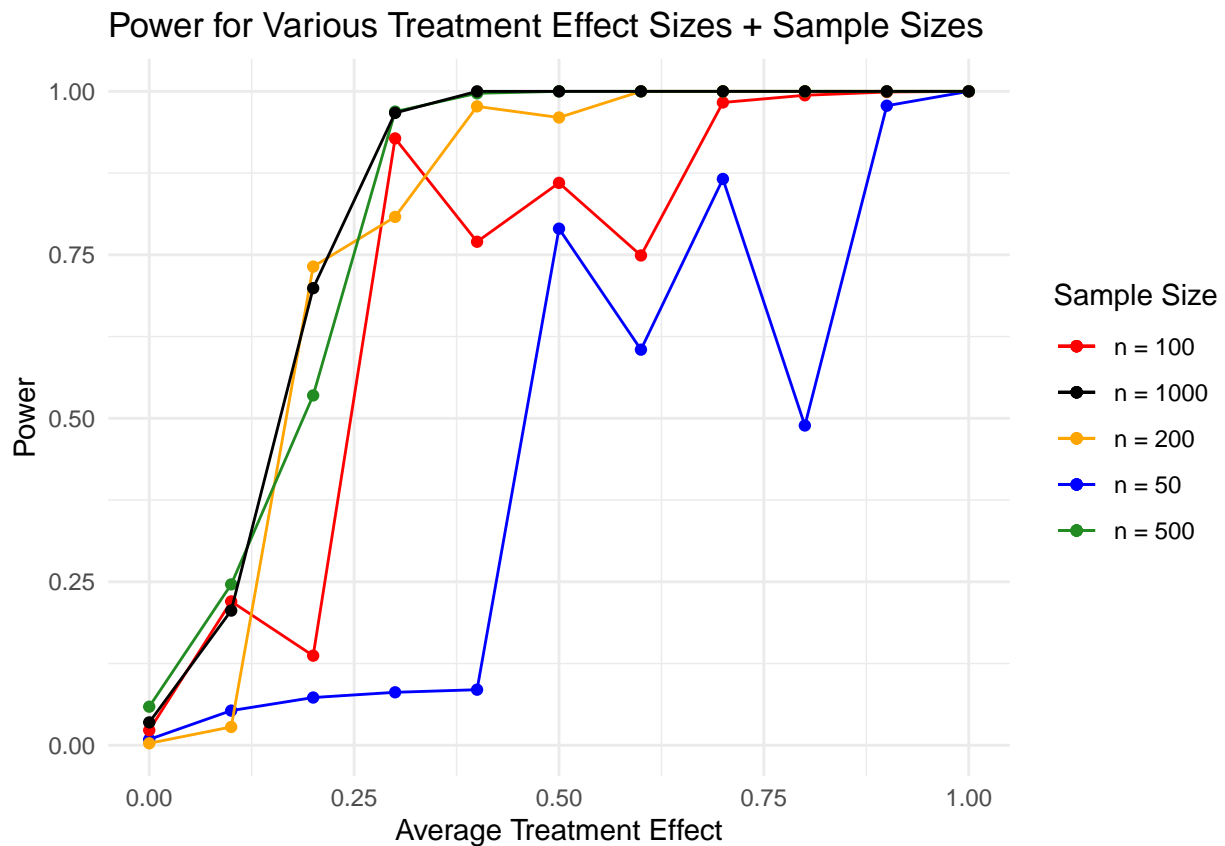


Figure 2: Power For Various Sample Sizes

For a given effect size, power is (generally) greater when the sample size is larger. However, there are diminishing returns to increased sample size. The ability to detect effects of similar magnitude is substantially greater by adding 50 - 150 observations when you start with 50 observations to 200, but you don't get much increase in power by adding observations thereafter. Additionally, there is more variability in curves for smaller samples (they are non-monotonic), indicating how results/estimates drawn from small samples tend to be more variable than results drawn from larger ones.

Question 2.

```
#reset the seed
set.seed(02139)

#generate vector of cluster memberships
jij <- rep(c(1:50), each = 20)

#draw group level means for each cluster
jij.mu <- runif(n = 50, min = 0, max = 10)

#data frame of cluster num + cluster mean
clusterInfo <- as.data.frame(cbind(c(1:50), jij.mu))

#observation frame to match to clusterInfo
obs.frame <- as.data.frame(cbind(jij))

#now assign each individual observation its cluster mean
#by merging
matched <- clusterInfo %>% right_join(obs.frame, by = c("V1" = "jij"))

#add individual error term
matched$jij.error <- rnorm(n = 1000, mean = 0, sd = 1)

#generate the correlated error terms as group mean + individual error
matched$group.corr.error <- matched$jij.mu + matched$jij.error

#add to old y0 vector and call y*0
y0star <- y0 + matched$group.corr.error

#also add it to df
matched$y0star <- y0star
```

To simulate the potential outcomes under control for 1000 clustered individuals (50 groups of 20 units each), I first generated a random group-level mean for each cluster using a random uniform distribution ranging from 0 to 10 and assigned the group mean to every individual within that cluster. I then simulated within cluster variability by adding a random error term to each individual, this time drawn from a normal distribution with a mean of 0 and standard deviation of 1. Finally, to account for sampling variability, I added to each individual an additional error term, drawn from a normal distribution with mean and standard deviation of 1.

```
#b.
#modify function from pt 1 to sample by cluster

treatFxClus <- function(v1, v0, cluster){
```

```
#####
# Assign Treatment
#####

#randomly assign treatment by cluster, 1 = treated, 0 = control
#where 1/2 clusters are treated
treat <- sample(x = unique(cluster), size = length(unique(cluster))/2, replace = FALSE)

#make a data frame of outcomes and their respective clusters
df <- as.data.frame(cbind(v1,v0,cluster))

#if cluster was sampled, treated, else control
df <- df %>%
  mutate(treat = ifelse(cluster %in% treat, 1, 0)) %>%
  #create one observed outcome vec: y1 is treated, y0 if not
  mutate(obs = ifelse(treat == 1, v1, v0))

#get cluster group means
df <- df %>%
  group_by(cluster) %>%
  mutate(groupMean = mean(obs, na.rm = T)) %>%
  ungroup()

#####
# Get ATE
#####

#get means of treated/control
treat.mean <- df %>%
  filter(treat == 1) %>%
  summarise(mean = mean(obs, na.rm = T))

control.mean <- df %>%
  filter(treat == 0) %>%
  summarise(mean = mean(obs, na.rm = T))

#get variances of treated/control
treat.var <- df %>%
  filter(treat == 1) %>%
  summarise(var = var(obs, na.rm = T))

#control variance
control.var <- df %>%
  filter(treat == 0) %>%
  summarise(var = var(obs, na.rm = T))

#ate:
ate <- treat.mean$mean - control.mean$mean

#get p value and perform hypothesis test
test.res <-
  Hmisc::t.test.cluster(y = df$obs,
```

```

        cluster = df$cluster,
        group = df$treat)

#return p value
return(test.res[20])
}

#generate potential outcomes under treatment

#create random treatment effect (mean = tau)
treat.error <- rnorm(n = 1000, mean = 0.2, sd = 1)

y1star <- matched$y0star + treat.error

#Run function
treatFxClus(v1 = y1star, v0 = y0star, cluster = matched$V1)

```

```
## [1] 0.002342347
```

Like in the original simulation, the test under the clustering structure is significant.

```

#c. replicate 1000 times for different values of tau
#create another function that runs results 1000 times each
#(n = pop size, cluster = cluster indicator, v0 = potential outcomes control)
repCluster1000 <- function(tau, times = 1000, n = 1000, dat = matched){

  #create random treatment effect (mean = tau)
  treat.error <- rnorm(n = n, mean = tau, sd = 1)

  # add error to group means calculated previously
  y1star <- dat$y0star + treat.error

  #do simulation for 1000 times
  res <- replicate(n = times,
    expr = treatFxClus(v0 = dat$y0star, v1 = y1star, cluster = dat$V1),
    simplify = "vector")

  #return % of values <0.05

  return(sum(res < 0.05) / (length(res)))
}

#run 1000 times at different values of tau
clus.results <- purrr::map_dbl(seq(0, 1, .1), ~repCluster1000(tau = .x, times = 1000))

#add these to the plot of results of 1c.

#plot results
as.data.frame(cbind(results, clus.results, seq(0, 1, .1))) %>%
  ggplot(aes(x = V3)) +

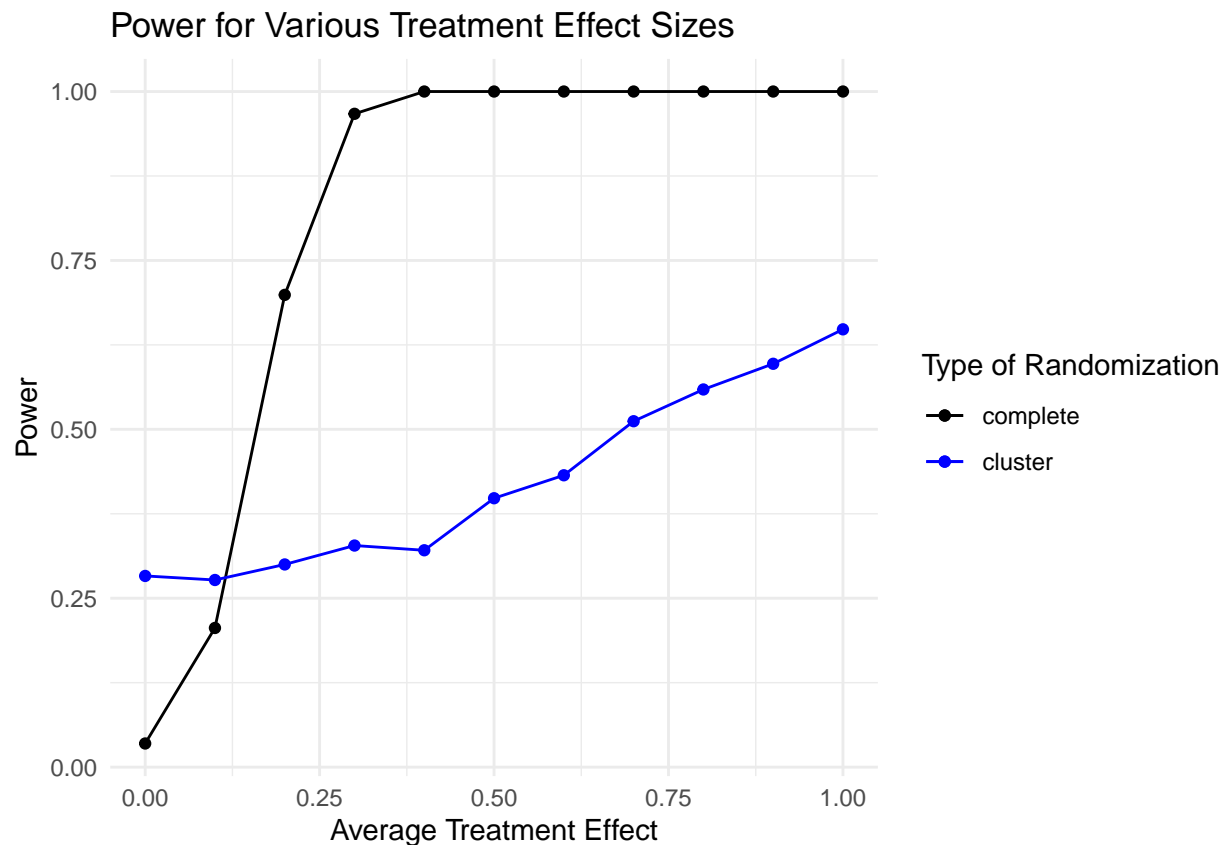
```



```

geom_point(aes(y = results, color = "black")) +
geom_line(aes(y = results, color = "black")) +
geom_point(aes(y = clus.results, color = "blue")) +
geom_line(aes(y = clus.results, color = "blue")) +
theme_minimal() +
labs(x = "Average Treatment Effect", y = "Power") +
ggtitle("Power for Various Treatment Effect Sizes") +
scale_color_manual(values = c("black", "blue"), labels = c("complete", "cluster"),
                    name = "Type of Randomization")

```



Yes, the results differ. The power of the test under cluster randomization is lower for all values of τ (except 0 - 0.10, when there is actually no treatment effect or only a small treatment effect) than the complete randomization. Again, power increases as the effect size increases, but this increase is relatively gradual and constant compared to the sharp increase in power for complete randomization for $\tau > 0.1$. The reason for this is that cluster randomization inflates the sampling variance, and for a given effect size, greater variability will result in less power.

```

#sample 20 clusters (total n = 400) from population
#get 20 cluster indicators = if they've been chosen
clus.20 <- sample(c(1:50), size = 20, replace = F)

#keep original data from only the sampled clusters
clus.20.data <- matched[matched$V1 %in% clus.20,]

#reassign treatment and calculate ATE using this sample only
res.20.clus <- purrr::map_dbl(seq(0, 1, .1), ~repCluster1000(tau = .x,

```

```

times = 1000,
n = 400,
dat = clus.20.data))

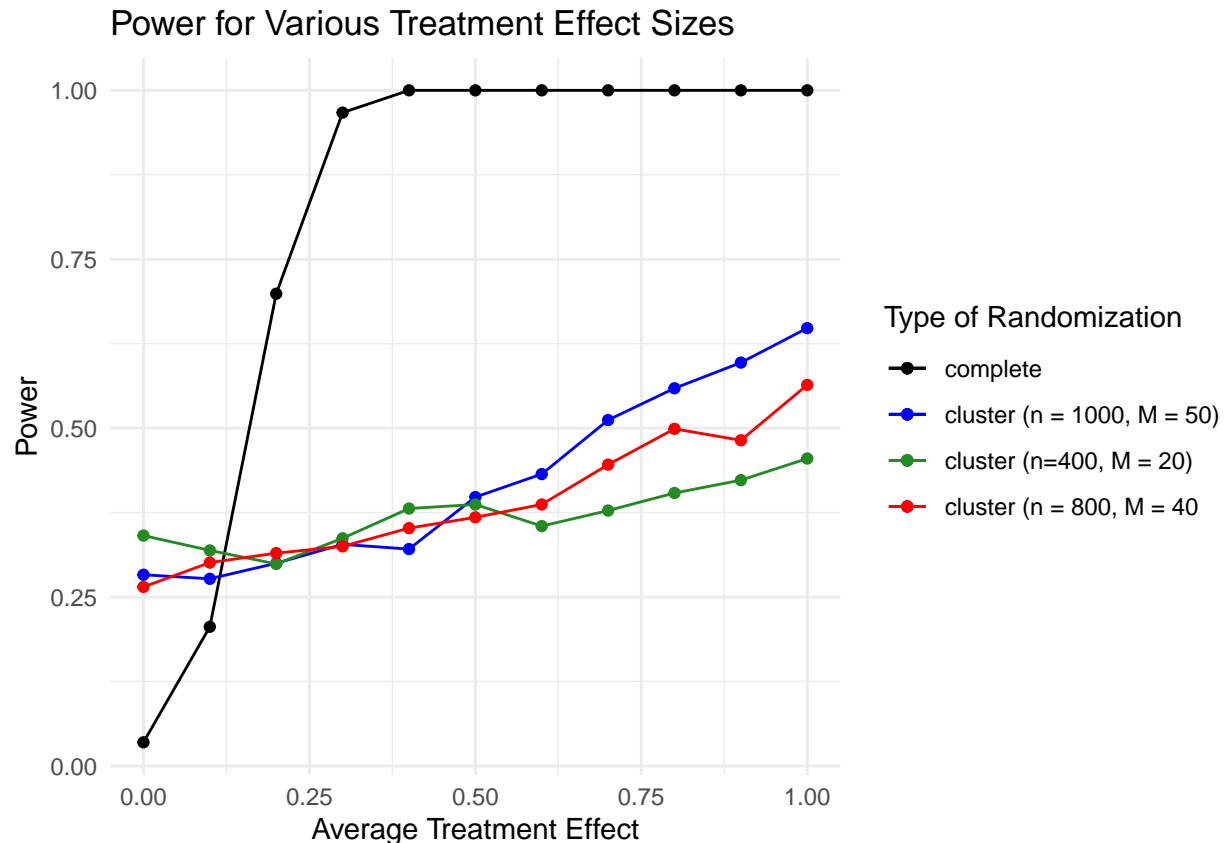
#sample 40 clusters
clus.40 <- sample(c(1:50), size = 40, replace = F)

#keep original data from only the sampled clusters
clus.40.data <- matched[matched$V1 %in% clus.40,]

#reassign treatment and calculate ATE using this sample only
res.40.clus <- purrr::map_dbl(seq(0, 1, .1), ~repCluster1000(tau = .x,
times = 1000,
n = 800,
dat = clus.40.data))

#add results to the plot
as.data.frame(cbind(results, clus.results, seq(0, 1, .1), res.20.clus)) %>%
  ggplot(aes(x = V3)) +
  geom_point(aes(y = results, color = "black")) +
  geom_line(aes(y = results, color = "black")) +
  geom_point(aes(y = clus.results, color = "blue")) +
  geom_line(aes(y = clus.results, color = "blue")) +
  geom_point(aes(y = res.20.clus, color = "forestgreen")) +
  geom_line(aes(y = res.20.clus, color = "forestgreen")) +
  geom_point(aes(y = res.40.clus, color = "red")) +
  geom_line(aes(y = res.40.clus, color = "red")) +
  theme_minimal() +
  labs(x = "Average Treatment Effect", y = "Power") +
  ggtitle("Power for Various Treatment Effect Sizes") +
  scale_color_manual(values = c("black", "blue", "forestgreen", "red"),
labels = c("complete", "cluster (n = 1000, M = 50)", "cluster (n=400, M = 20)",
"cluster (n = 800, M = 40)",
name = "Type of Randomization")

```



You have much lower power in the cluster randomized designs as compared to complete randomization. Keeping cluster sizes the same, there is lower power for smaller sample sizes (and thus a small numbers of clusters) for a given effect size.

e.

```
#divide population into 25 clusters of 40 individuals each

#generate vector of cluster memberships
jij2 <- rep(c(1:25), each = 40)

#draw group level means for each cluster
jij.mu2 <- runif(n = 25, min = 0, max = 10)

#data frame of cluster num + cluster mean
clusterInfo2 <- as.data.frame(cbind(c(1:25), jij.mu2))

#observation frame to match to clusterInfo
obs.frame2 <- as.data.frame(cbind(jij2)) #1:1000 is individual obs number

#now assign each individual observation its cluster mean
matched2 <- clusterInfo2 %>% right_join(obs.frame2, by = c("V1" = "jij2"))

#add individual error term
matched2$jij.error <- rnorm(n = 1000, mean = 0, sd = 1)
```

```

#generate the correlated error terms as group mean + individual error
matched2$group.corr.error <- matched2$jij.mu2 + matched2$jij.error

#add to old y0 vector and call y*0 2
y0star2 <- y0 + matched2$group.corr.error

#also add it to df
matched2$y0star <- y0star2

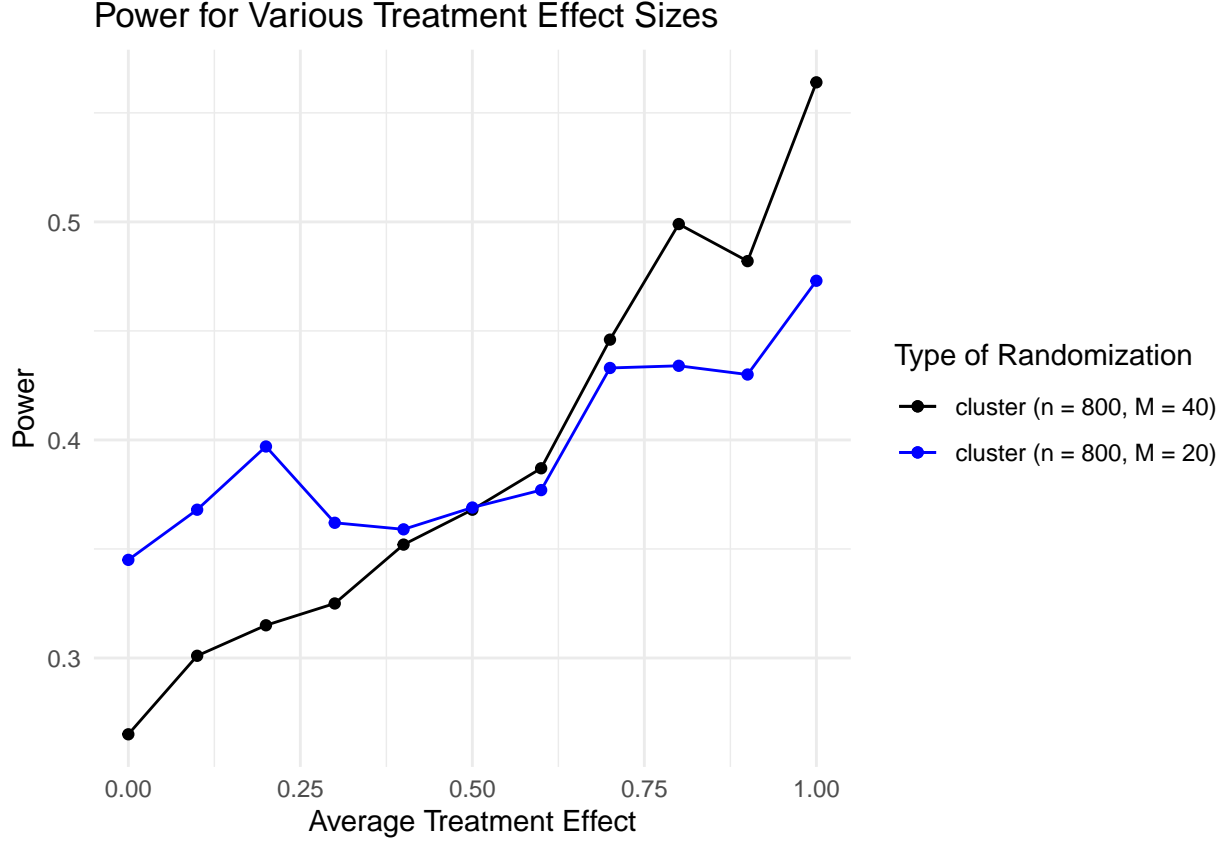
#sample 20 clusters and repeat part d.
clus.20.2 <- sample(c(1:25), size = 20, replace = F)

#keep original data from only the sampled clusters
clus.20.data2 <- matched2[matched2$V1 %in% clus.20.2,]

#reassign treatment and calculate ATE using this sample only
res.20.clus2 <- purrr::map_dbl(seq(0, 1, .1), ~repCluster1000(tau = .x,
                                                                times = 1000,
                                                                n = 800,
                                                                dat = clus.20.data2))

#plot results
as.data.frame(cbind(seq(0, 1, .1), res.40.clus, res.20.clus2)) %>%
  ggplot(aes(x = V1)) +
  geom_point(aes(y = res.40.clus, color = "black")) +
  geom_line(aes(y = res.40.clus, color = "black")) +
  geom_point(aes(y = res.20.clus2, color = "blue")) +
  geom_line(aes(y = res.20.clus2, color = "blue")) +
  theme_minimal() +
  labs(x = "Average Treatment Effect", y = "Power") +
  ggtitle("Power for Various Treatment Effect Sizes") +
  scale_color_manual(values = c("black", "blue"),
                     labels = c("cluster (n = 800, M = 40)", "cluster (n = 800, M = 20)"),
                     name = "Type of Randomization")

```



No, your friend is not correct. For the same sample size, power is (generally) higher when you have a larger number of clusters rather than a larger number of observations within each cluster, especially at larger effect sizes. Also somewhat concerning, 35% of tests run in the condition with 20 clusters found a significant effect of treatment when there, in fact, was no effect, representing a substantial type 1 error rate. The reason for this is the design-effect/Moulton factor. The sampling variance associated with cluster randomization increases by a factor of $1 + (\bar{N} - 1)\rho$ where ρ is the ICC and \bar{N} is the average number of observations in a cluster, as defined below.

$$\bar{N} = \frac{1}{G} \sum_{j=1}^G N_j$$

For constant values of N_j , \bar{N} will be smaller with larger values of G (# of clusters). It then follows that the Moulton factor ($1 + (\bar{N} - 1)\rho$) will also be smaller, as will the sampling variability. With less variability, power is increased, thus: with more groups, more power.

f. The sampling variance of the ATE is defined as

$$Var(\hat{\tau}) = \frac{1}{M-1} \left(\frac{n_0}{n_1} Var(\bar{y}_{i1}) + \frac{n_1}{n_0} Var(\bar{y}_{i0}) \right)$$

This shows that the findings in part e are not simply an artifact of the simulation design. The denominator $M - 1$, of number of clusters minus 1. As we increase the number of clusters for fixed sample size and variances in the potential outcomes, the denominator becomes larger and the variance in the estimate of ATE smaller.

Two ways to reduce sampling variance in a clustered design, then, are: 1) Increasing the number of clusters. This makes the

$$M - 1$$

, the denominator of the multiplier, larger for fixed variances in potential outcomes y_{i1} and y_{i0} . With a larger denominator, the end result is smaller, and variance lower.

- 2) Make the ratio of treated:control clusters 1 (i.e. assign the same number of clusters to treatment as to control) Given the variance formula for the ATE:

$$Var(\hat{\tau}) = \frac{1}{M-1} \left(\frac{n_0}{n_1} Var(\bar{y}_{i1}) + \frac{n_1}{n_0} Var(\bar{y}_{i0}) \right)$$

If we let p be the fraction of clusters assigned to control (n_1/N) then the expression becomes:

$$Var(\hat{\tau}) = \frac{1}{M-1} \left(\frac{Var(\bar{y}_{i1})}{pN} + \frac{Var(\bar{y}_{i0})}{(1-p)N} \right)$$

To find the value of p^* that minimizes the variances, we take the derivative of the right hand side with respect to p and setting it to 0.

$$0 = -\frac{Var(\bar{y}_{i1})}{p^2 N} + \frac{Var(\bar{y}_{i0})}{(1-p)^2 N}$$

Then, we solve for p^*

$$\frac{Var(\bar{y}_{i1})}{p^2 N} = \frac{Var(\bar{y}_{i0})}{(1-p)^2 N}$$

$$\frac{sd(\bar{y}_{i1})}{p N} = \frac{sd(\bar{y}_{i0})}{(1-p)N}$$

$$\frac{sd(\bar{y}_{i1})}{sd(\bar{y}_{i0})} = \frac{p}{(1-p)}$$

For cases in which the standard deviation of treatment and control is the same, $sd(\bar{y}_{i1}) = sd(\bar{y}_{i0})$, we get:

$$1 = \frac{p}{(1-p)}$$

$$p = (1-p)$$

$$p = 0.5$$

Thus, the fraction of clusters assigned to 0.5 and the fraction of clusters assigned to control is $1-0.5 = 0.5$. Altogether, this means that equal allocation of clusters to control and treatment will also minimize variance.

Question 3.

- a. The ATE parameter is defined as the expectation of the difference between \bar{X} and \bar{Y} : $\mathbb{E}(\bar{X} - \bar{Y})$. It represents the difference in the potential outcomes given treatment A versus treatment B . We estimate this parameter from our sample means as follows:

$$\begin{aligned} & \mathbb{E}[X - Y] \\ & \mathbb{E}[X] - \mathbb{E}[Y] \\ & \bar{X} - \bar{Y} \\ & \frac{1}{n} \sum_{i=1}^i x_i - \frac{1}{m} \sum_{i=1}^i y_i \end{aligned}$$

- b. The variance of the ATE estimator is defined as:

$$\frac{1}{N} \left(\frac{m}{n} \frac{N-n}{N-1} \frac{\sigma^2}{n} + \frac{n}{m} \frac{N-m}{N-1} \frac{\tau^2}{m} - 2 \left(\frac{1}{N-1} \text{cov}(x, y) \right) \right)$$

Where

$$\frac{N-n}{N-1} \frac{\sigma^2}{n}$$

is

$$\text{Var}(\bar{X})$$

,

$$\frac{N-m}{N-1} \frac{\tau^2}{m}$$

is

$$\text{Var}(\bar{Y})$$

and

$$- \left(\frac{1}{N-1} \text{cov}(x, y) \right)$$

is

$$\text{cov}(\bar{X}, \bar{Y})$$

- c. The difference between this variance and the “true” variance in the population are that this variance treats the covariance of x and y as 0. We cannot ever observe this with the data we have because we can only observe one of the potential outcomes at a time.