```
### WEEK 3 ###

#Simulate an experiment with no effect
group <- c(rep("Man",20),rep("Woman",20))
po.control <- c(seq(from = 1, to = 20), seq(from = 51, to = 70))
po.treatment <- po.control #no effect because potential outcomes in
treatment are the same

po.control
po.treatment

#Function to randomly assign units to treatment and control
randomize <- function() sample(c(rep(0,20),rep(1,20)))
randomize()
randomize()

treatment <- randomize() #Conduct randomization for this experiment
treatment

#Realized outcomes - treatment outcome for those randomized to
treatment and control outcome forthose randomized to control
outcomes <- po.treatment * treatment + po.control*(1-treatment)
outcomes

#Function to estimate the average treatment effect
est.ate <- function(outcome, treat) mean(outcome[treat==1]) -
mean(outcome[treat==0])

ate <- est.ate(outcomes, treatment) #Compute the average treatment
effect for this experiment
ate #you see a difference, despite there being no effect!

#How big is that difference likely to be on average?
#We can simulate this a few times to get a sense of how much our
treatment effect estimate would vary by chance
est.ate(outcomes, randomize())
est.ate(outcomes, randomize())
est.ate(outcomes, randomize())

#Do this 5,000 to get a sense of the distribution
distribution.under.sharp.null <- replicate(5000, est.ate(outcomes,
randomize()))
plot(density(distribution.under.sharp.null)) #It's likely we get big
differences by chance. This is a sampling distribution.

#How big was our observed difference?
plot(density(distribution.under.sharp.null))
abline(v=ate) #pretty similar to one we'd get by chance

mean(ate <= distribution.under.sharp.null) #p-value
```

```
#Simulate an experiment with a large effect
#assignment and outcomes
po.treatment <- po.control + 25
po.control
po.treatment
treatment <- randomize()
outcomes <- po.treatment * treatment + po.control*(1-treatment)
outcomes

#estimate ate
ate <- est.ate(outcomes, treatment)
ate

#what's the uncertainty?
distribution.under.sharp.null <- replicate(5000, est.ate(outcomes,
randomize()))
plot(density(distribution.under.sharp.null))
abline(v=ate)
mean(ate < distribution.under.sharp.null) #p-value


#Statistical power
#Function to simulate a study of a given treatment effect and get the
p-value
simulate.study <- function(treatment.effect.size){
        po.control <- c(seq(from = 1, to = 20), seq(from = 51, to =
70))
        po.treatment <- po.control + treatment.effect.size
        treatment <- randomize()
        outcomes <- po.treatment * treatment + po.control * (1-
treatment)
        ate <- est.ate(outcomes, treatment)
        distribution.under.sharp.null <- replicate(1000,
est.ate(outcomes, randomize()))
        return(mean(ate < distribution.under.sharp.null))
}
simulate.study(0) #p-value for no effect
p.values <- replicate(10000, simulate.study(0)) #distribution of p-
values
plot(density(p.values)) #uniform distribution
mean(p.values < 0.05) #how often is p.value under 0.05 when there's no
effect?

p.values <- replicate(1000, simulate.study(10))
plot(density(p.values))
mean(p.values < 0.05) #somewhat likely to detect this effect
```

```
p.values <- replicate(1000, simulate.study(20))
plot(density(p.values))
mean(p.values < 0.05) #very likely to detect this effect


#How does power behave?
simulate.study.lm <- function(baseline, effect.size, sample.size){
        control.units <- rbinom(sample.size, 1, baseline)
        treatment.units <- rbinom(sample.size, 1, baseline +
effect.size)
        all.units <- c(control.units, treatment.units)
        treatment.vector <- c(rep(0,sample.size), rep(1,sample.size))
        p.value <- summary(lm(all.units ~ treatment.vector))
$coefficients[2,4]
        effect.detected <- p.value < 0.05
        return(effect.detected)
}
get.power <- function(baseline, effect.size, sample.size){
        return(mean(replicate(2000, simulate.study.lm(baseline,
effect.size, sample.size))))
}

#Increasing effect size
get.power(.1, .05, 100)
get.power(.1, .1, 100)
get.power(.1, .15, 100)
get.power(.1, .2, 100)
get.power(.1, .25, 100)

#Increasing sample size
get.power(.1, .05, 100)
get.power(.1, .05, 200)
get.power(.1, .05, 300)
get.power(.1, .05, 400)
get.power(.1, .05, 500)
get.power(.1, .05, 1000)
get.power(.1, .05, 5000)


#Confidence interval
summary(lm(outcomes ~ treatment))

estimate.in.confidence.interval <- function(){
        true.effect <- 25
        #Simulate outcomes
        po.control <- c(seq(from = 1, to = 20), seq(from = 51, to =
70))
        po.treatment <- po.control + true.effect
        treatment <- randomize()
        outcomes <- po.treatment * treatment + po.control*(1-
treatment)
        #Run regression
        regression <- summary(lm(outcomes ~ treatment))
        estimate <- regression$coefficients[2,1]
        standard.error <- regression$coefficients[2,2]
        lower.bound <- estimate - standard.error * 1.96
        upper.bound <- estimate + standard.error * 1.96
        #Is estimate in CI?
        estimate.in.ci <- lower.bound < true.effect & upper.bound >
true.effect
        return(estimate.in.ci)
}
estimate.in.confidence.interval()
mean(replicate(10000, estimate.in.confidence.interval()))


### WEEK 4 ###
#Benefits of blocking
group #recall the group variable
randomize <- function() sample(c(rep(0,20),rep(1,20)))
#on average groups are split evenly, but for any partiuclar
randomization they often are not
table(group, randomize())
table(group, randomize())
table(group, randomize())

est.ate <- function(outcome, treat) mean(outcome[treat==1]) -
mean(outcome[treat==0])
sim.normal.study <- function(){
        po.control <- c(seq(from = 1, to = 20), seq(from = 51, to =
70))
        po.treatment <- po.control
        treatment <- randomize()
        outcomes <- po.treatment * treatment + po.control * (1-
treatment)
        ate <- est.ate(outcomes, treatment)
        n.women.treatment <- table(group, treatment)[2,2]
        return(list(ate = ate, n.women.treatment = n.women.treatment))
}
results <- t(replicate(1000, sim.normal.study()))
plot(results)

randomize.blocked <- function(){
        c(
                sample(c(rep(0,10),rep(1,10))), #group A
                sample(c(rep(0,10),rep(1,10))) #group B
        )
}

#now groups are always balanced
```

```
table(group, randomize.blocked())
table(group, randomize.blocked())
table(group, randomize.blocked())

#results of the experiment with blocking
po.control <- c(seq(from = 1, to = 20), seq(from = 51, to = 70))
po.treatment <- po.control + 10 #simulate effect of 10

treatment.blocked <- randomize.blocked()
outcomes.blocked <- po.treatment * treatment.blocked + po.control *
(1-treatment.blocked)

ate <- est.ate(outcomes.blocked, treatment.blocked)
ate

distribution.under.sharp.null.blocked <- replicate(5000,
est.ate(outcomes.blocked, randomize.blocked()))
plot(density(distribution.under.sharp.null), col="red", ylim=c(0,.17))
#distribution without blocking
abline(v=ate)
mean(ate < distribution.under.sharp.null)
lines(density(distribution.under.sharp.null.blocked), col="blue")
#distribution with blocking
mean(ate < distribution.under.sharp.null.blocked)

#Similar gains when using regression
po.control <- c(seq(from = 1, to = 20), seq(from = 51, to = 70))
po.treatment <- po.control + 10 #simulate effect of 10
treatment <- randomize()
outcomes <- po.treatment * treatment + po.control * (1-treatment)
summary(lm(outcomes ~ treatment)) #without block indicator
summary(lm(outcomes ~ treatment + factor(group))) #with block
indicator


#Clustering can decrease power
n.classrooms <- 8
n.students <- 16

classroom.ids <- unlist(lapply(1:n.classrooms, function(x)
rep(x,times=n.students)))
classroom.ids
all.classrooms <- unique(classroom.ids)
all.classrooms

classroom.level.noise <- rnorm(length(all.classrooms))
classroom.level.noise

student.outcomes.control <- rnorm(length(classroom.ids)) +
classroom.level.noise[classroom.ids]
```

```
student.outcomes.control

student.outcomes.treat <- student.outcomes.control + 0.75

randomize.clustered <- function(){
        treat.classroom.ids <- sample(all.classrooms, n.classrooms/2)
        return(
                as.numeric(classroom.ids %in% treat.classroom.ids)
        )
}

randomize.clustered()
randomize.clustered()
randomize.clustered()

#Clustered
treat <- randomize.clustered()
outcomes <- treat * student.outcomes.treat + (1-treat) *
student.outcomes.control
ate <- est.ate(outcomes, treat)
ate

distribution.under.sharp.null <- replicate(5000, est.ate(outcomes,
randomize.clustered()))
plot(density(distribution.under.sharp.null))
abline(v=ate)
mean(ate < distribution.under.sharp.null) #p-value

#What if we ignore clustering?
randomize.ignorning.clustering <- function()
sample(c(rep(0,n.classrooms*n.students/
2),rep(1,n.classrooms*n.students/2)))
randomize.ignorning.clustering()

distribution.under.sharp.null.wrong <- replicate(5000,
est.ate(outcomes, randomize.ignorning.clustering()))
plot(density(distribution.under.sharp.null), ylim=c(0,1.5))
lines(density(distribution.under.sharp.null.wrong))
abline(v=ate)
mean(ate < distribution.under.sharp.null.wrong) #p-value


#No cluster level noise
plot(density(replicate(5000, est.ate(rnorm(length(classroom.ids)),
randomize.clustered()))), xlim=c(-2,2)) #no classroom level noise
added
lines(density(replicate(5000, est.ate(rnorm(length(classroom.ids)) +
classroom.level.noise[classroom.ids], randomize.clustered()))))
#classroom level noise added
lines(density(replicate(5000, est.ate(rnorm(length(classroom.ids)) +
rnorm(length(classroom.ids)), randomize.clustered()))))) #not clustered
noise of same size
```