hw3

Matthew Zhao

2023-11-13

Part 1.1

```
# creating dataframe of just farmer ids
id <- seq(1,200)

df <- data.frame(
  id
)

# performing randomization
n <- df %>% nrow()
rand <- runif(n, 0, 1)
rank <- rank(rand)
df$treat <- ifelse(rank <= (n / 2), 1, 0)

# simulating plot yields
eps <- rnorm(n = 200, mean = 0, sd = 10)
alpha <- 120
beta <- 2.5
df$plot_yield <- alpha + beta * df$treat + eps

head(df)
```

```
##   id treat plot_yield
## 1  1     0  129.35536
## 2  2     0  111.95547
## 3  3     0   97.11463
## 4  4     0  117.85645
## 5  5     1  115.24454
## 6  6     1  123.84804
```

Part 1.2

```
model <- lm(plot_yield ~ treat, data = df)
summary(model)
```

```
##
## Call:
```

```
## lm(formula = plot_yield ~ treat, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -31.0728  -6.7953   0.5296   6.6526  25.1903
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 118.7486     0.9939 119.479  < 2e-16 ***
## treat         3.6617     1.4056   2.605  0.00988 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.939 on 198 degrees of freedom
## Multiple R-squared:  0.03314,    Adjusted R-squared:  0.02826
## F-statistic: 6.787 on 1 and 198 DF,  p-value: 0.009881
```

The point estimate of the effect of access to a new fertilizer on plot yield is 5.005. The p-value for β is 0.0918. We can only reject the null that $\beta = 0$ at the 10% significance level.

Part 1.3

```
dgp <- function(n) {
  # creating dataframe of just farmer ids
  id <- seq(1,n)

  df <- data.frame(
    id
  )

  # performing randomization
  rand <- runif(n, 0, 1)
  rank <- rank(rand)
  df$treat <- ifelse(rank <= (n / 2), 1, 0)

  # simulating plot yields
  eps <- rnorm(n = n, mean = 0, sd = 10)
  alpha <- 120
  beta <- 2.5
  df$plot_yield <- alpha + beta * df$treat + eps

  df
}

pvals <- c()
for (i in 1:1000) {
  df <- dgp(200)
  model <- lm(plot_yield ~ treat, data = df)
  pval <- summary(model)$coefficients['treat','Pr(>|t|)']
  pvals <- c(pvals,pval)
}
```

```r
reject10pct <- sum(pvals < 0.10)
print(paste0('Number of Rejections at 10% Significance Level: ', reject10pct))
```

```
## [1] "Number of Rejections at 10% Significance Level: 567"
```

```r
print(paste0('Power: ', reject10pct / 1000 * 100, '%'))
```

```
## [1] "Power: 56.7%"
```

Part 1.4

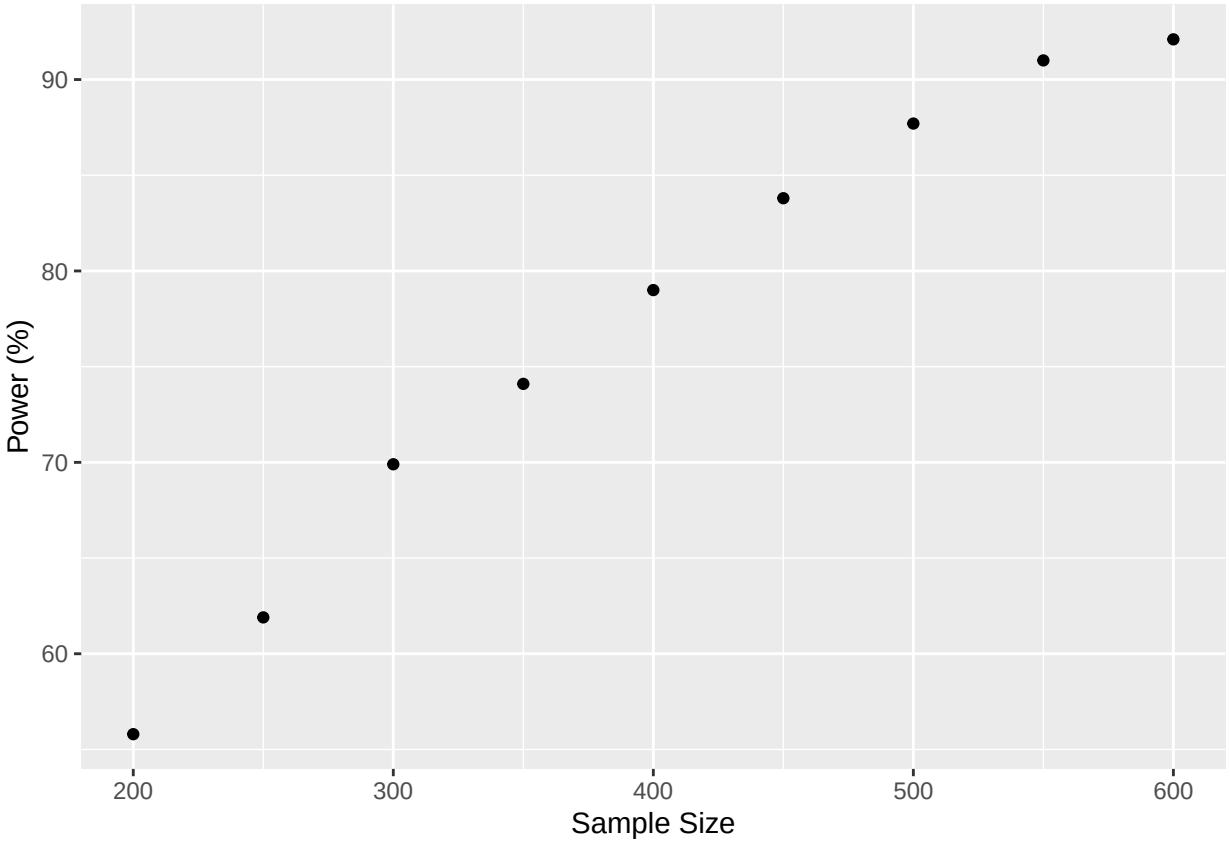
```r
power_vec <- c()

for (n in seq(200,600,50)) {
  pvals <- c()

  for (i in 1:1000) {
    df <- dgp(n)
    model <- lm(plot_yield ~ treat, data = df)
    pval <- summary(model)$coefficients['treat','Pr(>|t|)']
    pvals <- c(pvals,pval)
  }

  power <- sum(pvals < 0.10) / 1000 * 100
  power_vec <- c(power_vec, power)
}

ggplot(mapping = aes(x = seq(200,600,50), y = power_vec)) +
  geom_point() +
  labs(x = 'Sample Size', y = 'Power (%)')
```



The minimum sample size that will give us 80% power is 400.

Part 1.5

```
dgp_cluster <- function(n=200) {
  # creating dataframe of just farmer ids
  id <- seq(1,n)

  df <- data.frame(
    id
  )

  # add village cluster
  rand <- runif(n, 0, 1)
  rank <- rank(rand)
  df$village <- case_when(
    (rank >= quantile(rank, 0)) & (rank < quantile(rank, 0.25)) ~ 1,
    (rank >= quantile(rank, 0.25)) & (rank < quantile(rank, 0.50)) ~ 2,
    (rank >= quantile(rank, 0.50)) & (rank < quantile(rank, 0.75)) ~ 3,
    (rank >= quantile(rank, 0.75)) & (rank <= quantile(rank, 1)) ~ 4
  )


  # performing randomization at cluster level
```

```
  rand <- runif(4, 0, 1)
  rank <- rank(rand)
  cluster_trt <- data.frame(
    village = seq(1,4)
  )
  cluster_trt$treat <- ifelse(rank <= (4 / 2), 1, 0)
  df <- left_join(df, cluster_trt, by = 'village')

  # simulating plot yields
  eps <- rnorm(n = n, mean = 0, sd = 10)
  alpha <- 120
  beta <- 2.5
  df$plot_yield <- alpha + beta * df$treat + eps

  df
}


pvals <- c()
for (i in 1:1000) {
  df <- dgp_cluster()
  model <- feols(plot_yield ~ treat, data = df,
           cluster = c('village'))
  pval <- pvalue(model)[['treat']]
  pvals <- c(pvals,pval)
}

reject10pct <- sum(pvals < 0.10)
print(paste0('Number of Rejections at 10% Significance Level: ', reject10pct))
```

```
## [1] "Number of Rejections at 10% Significance Level: 560"
```

```
print(paste0('Power: ', reject10pct / 1000 * 100, '%'))
```

```
## [1] "Power: 56%"
```

The power is lower in the clustered designed compared to the individual design.

Part 1.6

The power in this case will be lower than either the clustered design from 1.5 or the individual design. This is because the intra-cluster correlation in this case will be higher than in the clustered design from 1.5 since we are grouping individuals by plot yield, thereby artificially increasing the ICC. Higher ICC reduces power by reducing our effective sample size since including more members from the same cluster adds less information/has less variation.