Section 6: Computer Assisted Portion (Practice)

Shikhar Singh

1 March 2018

Practice Set I

In this section you will analyze a hypothetical canvassing experiment. Professor Smedley, who designed this project, conducted three mini-experiments, one in each locality. In each mini-experiment, he randomly assigned m of N households to the treatment group. Households in the treatment group were visited by one of Professor Smedley's canvassers. No household in the control group was visited by any canvasser, but there was failure to establish contact with some households in the treatment group. After the intervention, Smedley acquired turnout data from the voter file. The quantity of interest is the effect of being *assigned* to the canvassing treatment on turnout in an election.

1. Download the dataset from Canvas or http://github.com/shikhar46/experiments. How many blocks are there? How many clusters are nested in each block? Is the probability of assignment constant across blocks?
2. Use declare_ra to specify the design features of this study to randomizr; then use obtain_condition_probabilities to compute weights.
3. Use lm_robust from the estimatr package to compute the average treatment effect using an IPW estimator, also report the cluster-robust standard error (uncertainty around that ATE estimate).
4. Conduct randomization inference using the ri2 package, and report the p value from a one-tailed test under the sharp null $\tau_i = 0 \forall i$.
5. Make a 95% confidence interval for the ATE estimate under the constant effects assumption. Use either the ri2 package, or a loop for this purpose.
6. **Bonus**: Consider two estimators:

Blocked ATE: $\sum_{j=1}^{J} \frac{N_j}{N} ATE_j$

IPW regression with Block FE: $lm\_robust(Y_i \sim Z_i + Block, weights = weights)$

Are the average treatment effect estimates from these two procedures the same? Can you verify this is the case in Professor Smedley's dataset? Remember to "cluster" observations by household while estimating block-level ATEs or the IPW regression with block fixed effects. (In order to do this, compare your answer in 3 to a blocked ATE estimate obtained in one of two ways: using difference_in_means, or by calculating block-level ATEs, then weighting those estimates by block size). What did you learn about the two estimators in the presence of clustering?

Practice Set II

In this section you will work with Gerber and Green (2000)'s New Haven voter mobilization experiment dataset. Consider a simplified version of the study: exactly 1445 of 7090 voters were randomly assigned to a treatment condition (contact with a canvasser), and the remaining to the control group (no contact was attempted). The outcome of interest is whether voter i voted in the 1998 election; and in particular, whether personal contact increased turnout.

1. Download the dataset from Canvas or http://github.com/shikhar46/experiments. What proportion of subjects voted in the 1998 election? What proportion of subjects that voted in 1996 also voted in 1998?
2. Solely based on the data, can you tell whether there was one-sided or two-sided noncompliance in this study? If so, how? Produce a table or summary statistic to corroborate your claim. What types of respondents are there in the study?
3. Estimate the $\hat{ITT}$, $\hat{ITT}_D$, and $\hat{CACE}$. Confirm that ivreg gives the same complier average causal effect estimate. Interpret the CACE.
4. Consider a sharp null hypothesis $H_0 : \kappa_i = 0 \forall i$ where κ_i is unit-level causal effect for a complier, and i is the set of compliers. Use randomization inference to evaluate this sharp null. Can you reject the null hypothesis? (Use ri2 for this purpose)
5. It is widely claimed that random assignment, in expectation, achieves balance on observed and unobserved covariates. Conduct a randomization check to see if three covariates (Age, $Party$, and $Vote96$) predict treatment assignment. Report the p value for the relevant test-statistic using randomization inference.[1]
6. **Bonus**: Say it was brought to your notice that the random assignment procedure employed in this study was of the following kind:

Obtain an assignment vector (call it a "proposal"). Accept the proposal if the covariates do not jointly predict assignment status (i.e. the F-statistic from $Z_i \sim Age_i + Party_i + Vote96_i$ has $p \geq 0.10$); else reject the proposal and draw a new assignment vector.

Can you retrieve the complier average causal effect, knowing this assignment protocol? Use randomizr and ivreg to estimate the CACE. Use ri2 to test the sharp null $H_0 : ITT_i = 0 \forall i$.

[1] There are two ways to do this: (1) use a custom function in ri2 to conduct a randomization check; (2) use a loop to get the sampling distribution of the test statistic, then estimate the probability of observing the given value or more extreme in that distribution.

Answers to Practice Set I

```
# Load the dataset

data1 <- read.csv("W6_Smedley.csv")

# Q1
stats <- data1 %>% group_by(Locality) %>% summarise(Cluster.Count = length(unique(Household)),
    Prob.Trt = mean(Assignment == 1), Cluster.Treated = length(unique(Household[Assignment ==
        1])))
stats
```

```
## # A tibble: 3 x 4
##    Locality Cluster.Count  Prob.Trt Cluster.Treated
##      <fctr>         <int>     <dbl>           <int>
## 1  Brooklyn             5 0.6363636               3
## 2  Downtown            10 0.7600000               8
## 3 East Rock            10 0.4166667               4
```

```
# Q2

declaration <- declare_ra(N = nrow(data1), blocks = data1$Locality, clusters = data1$Household,
    block_m = c(3, 8, 4))

probs <- obtain_condition_probabilities(declaration, data1$Assignment)
data1$weights <- 1/probs

# Q3
model <- lm_robust(Voted ~ Assignment + Locality, clusters = Household, weights = weights,
    data = data1)
summary(model)
```

```
##
## Call:
## lm_robust(formula = Voted ~ Assignment + Locality, data = data1,
##     weights = weights, clusters = Household)
##
## Weighted, Standard error type =  CR2
##
## Coefficients:
##                   Estimate Std. Error Pr(>|t|) CI Lower CI Upper    DF
## (Intercept)        0.40730     0.2405   0.1713  -0.2822   1.0968 3.703
## Assignment         0.17215     0.2754   0.5532  -0.4887   0.8330 6.534
## LocalityDowntown   0.09803     0.3528   0.7918  -0.7977   0.9937 5.218
## LocalityEast Rock  0.08986     0.2408   0.7207  -0.4873   0.6670 6.560
##
## Multiple R-squared:  0.03413 ,   Adjusted R-squared:  -0.01761
## F-statistic: 0.6596 on 3 and 56 DF,  p-value: 0.5803
```

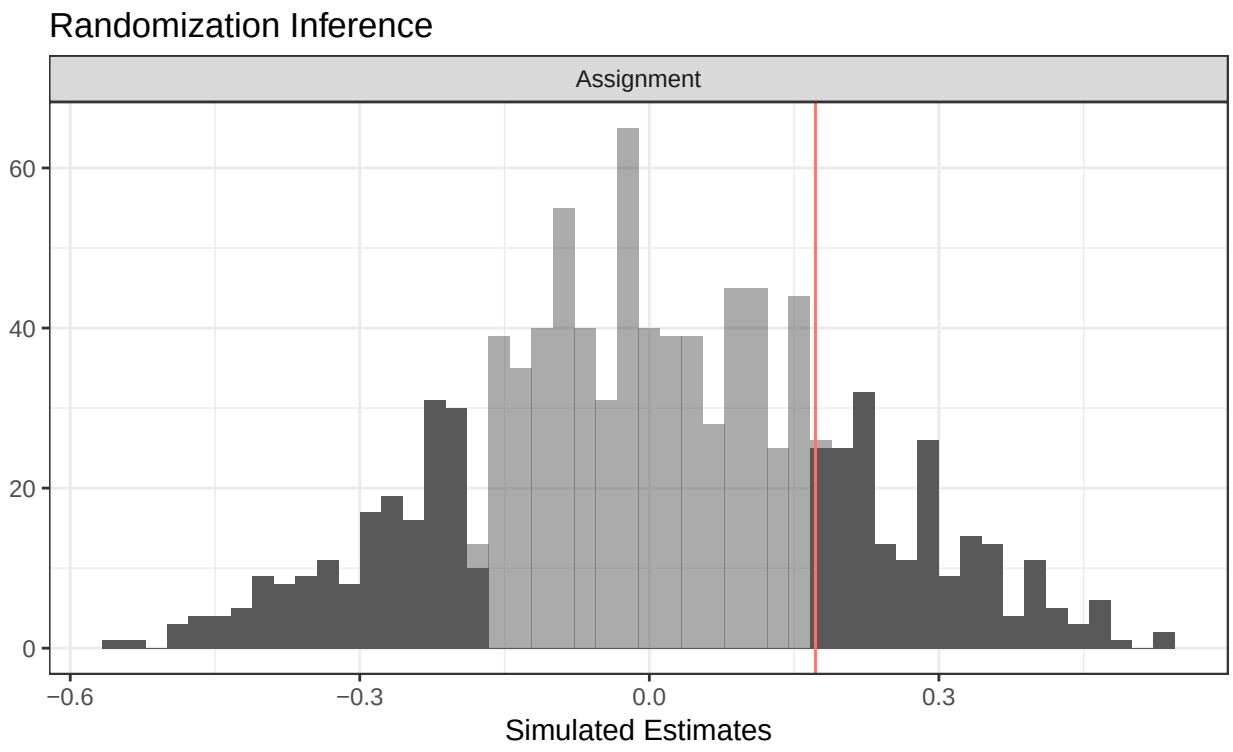
```
# Q4

ri_test <- ri2::conduct_ri(Voted ~ Assignment + Locality, declaration = declaration,
    assignment = "Assignment", sharp_hypothesis = 0, IPW_weights = weights,
    data = data1, sims = 1000)
```

```r
summary(ri_test, "upper")
```

```
##   coefficient  estimate upper_p_value null_ci_lower null_ci_upper
## 1  Assignment 0.1721521           0.2    -0.3942153     0.3934847
```

```r
plot(ri_test)
```




```r
# Q5 (CI using ri2)
ate_hat <- summary(ri_test)$estimate

ci <- ri2::conduct_ri(Voted ~ Assignment + Locality, declaration = declaration,
    assignment = "Assignment", sharp_hypothesis = ate_hat, IPW_weights = weights,
    data = data1, sims = 1000)

summary(ci)
```

```
##   coefficient  estimate two_tailed_p_value null_ci_lower null_ci_upper
## 1  Assignment 0.1721521              0.568     -0.199586     0.5242192
```

```r
# Q6

# Define outputs

block.estimates <- rep(NA, 3)
block.size <- rep(NA, 3)


block.name <- c("Brooklyn", "Downtown", "East Rock")
```

```r
# Compute block level ATE estimates and block size

for (i in 1:3) {
    fit <- data1 %>% filter(Locality == block.name[i]) %>% with(lm_robust(Voted ~
        Assignment, weights = weights, clusters = Household))
    block.estimates[i] <- fit$coefficients[2]
    size <- data1 %>% filter(Locality == block.name[i]) %>% summarise(N = n())
    block.size[i] <- size$N
}

block.estimates
```

```
## [1]  0.4642857 -0.1929825  0.4428571
```

```r
block.size
```

```
## [1] 11 25 24
```

```r
# Aggregating these estimates (weighting by block size)

blocked_ate_hat <- sum(block.estimates * (block.size/sum(block.size)))
blocked_ate_hat
```

```
## [1] 0.1818525
```

```r
# Q6 (Alternative)
blocked_ate_2 <- difference_in_means(Voted ~ Assignment, blocks = Locality,
    clusters = Household, data = data1)
blocked_ate_2
```

```
## Design:  Block-clustered
##             Estimate Std. Error  Pr(>|t|)   CI Lower  CI Upper DF
## Assignment 0.1818525  0.2380733 0.4543378 -0.3164407 0.6801458 19
```

Answers to Practice Set II

```
# Download the dataset
data2 <- read.csv("W6_GreenStudy.csv")

### Q1 Proportion of subjects who voted in 1998
mean(data2$Vote98)
```

```
## [1] 0.3832158
```

```
# Proporton of subjects who voted in 1996 and 1998
mean(data2$Vote96 == 1 & data2$Vote98 == 1)/mean(data2$Vote96 == 1)
```

```
## [1] 0.6160962
```

```
### Q2
xtabs(~data2$Contact + data2$Contacted)
```

```
##              data2$Contacted
## data2$Contact    0    1
##             0 5645    0
##             1 1050  395
```

```
# There is one-sided noncompliance: no one in the control group was
# contacted; but there was failure to contact in the treatment group ('Never
# Takers' = 1050)

### Q3
model_itt <- lm_robust(Vote98 ~ Contact, data = data2)
itt <- model_itt$coefficients[2]
itt
```

```
##    Contact
## 0.03846439
```

```
model.ittd <- lm_robust(Contacted ~ Contact, data = data2)
itt.d <- model.ittd$coefficients[2]
itt.d
```

```
##   Contact
## 0.2733564
```

```
itt/itt.d
```

```
##   Contact
## 0.1407115
```

```
# Alternatively using ivreg
model_cace <- AER::ivreg(Vote98 ~ Contacted | Contact, data = data2)
summary(model_cace)
```

```
##
## Call:
## AER::ivreg(formula = Vote98 ~ Contacted | Contact, data = data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5161 -0.3754 -0.3754  0.6246  0.6246
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.375376   0.006451  58.186  < 2e-16 ***
## Contacted   0.140712   0.052277   2.692  0.00713 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4847 on 7088 degrees of freedom
## Multiple R-Squared: 0.006279,    Adjusted R-squared: 0.006138
## Wald test: 7.245 on 1 and 7088 DF,  p-value: 0.007126
```

```
### Q4 (RI on ITT)
declaration <- declare_ra(N = nrow(data2), m = sum(data2$Contact))

ri_out <- conduct_ri(Vote98 ~ Contact, declaration = declaration, sharp_hypothesis = 0,
    assignment = "Contact", data = data2)
summary(ri_out)
```

```
##   coefficient   estimate two_tailed_p_value null_ci_lower null_ci_upper
## 1     Contact 0.03846439              0.013   -0.02848493    0.02542655
```

```
### Q5

model_balance <- lm_robust(Contact ~ Age + Party + Vote96, data = data2)
f.stat <- model_balance$fstatistic[1]
f.stat
```

```
##     value
## 0.3718727
```

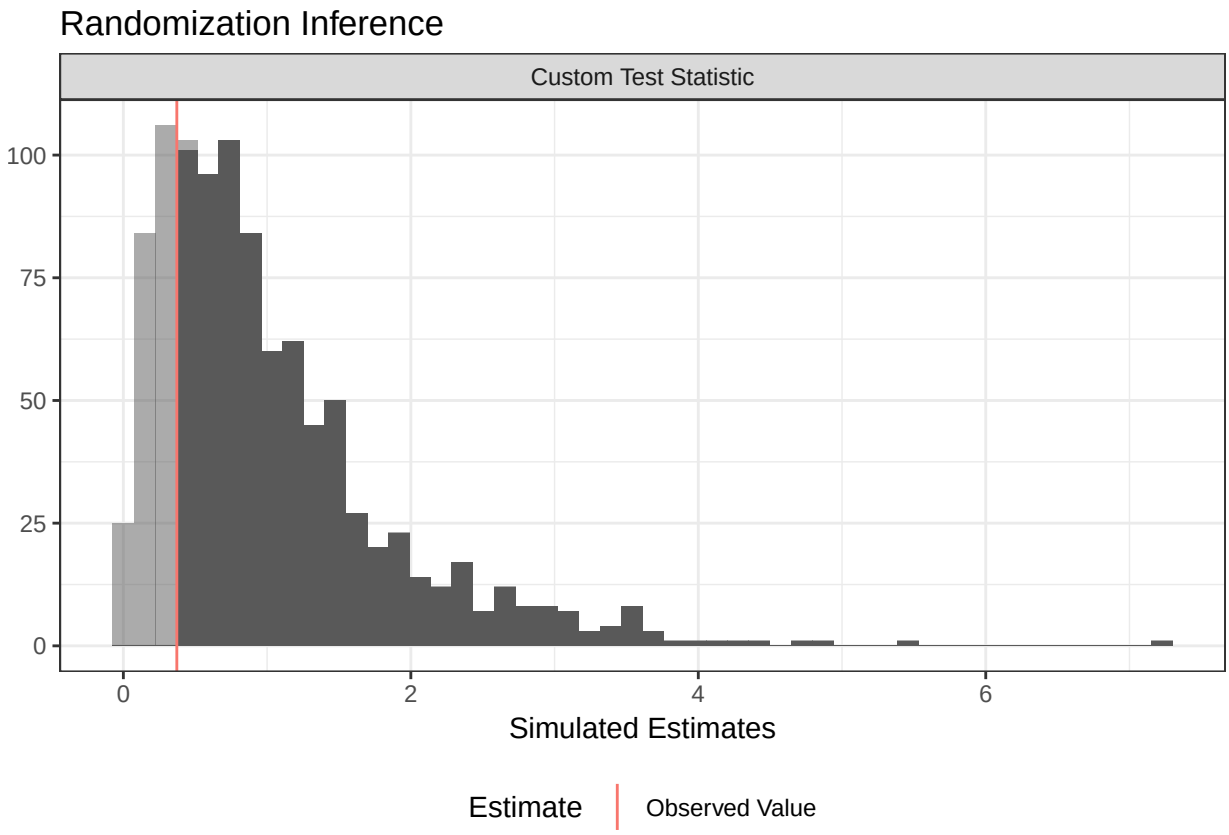
```
# Write a custom function to compute f-statistic

balance.test <- function(data) {
    model <- lm_robust(Contact ~ Age + Party + Vote96, data = data)
    f_stat <- summary(model)$fstatistic[1]
    names(f_stat) <- NULL  # removes the column name 'value'
    return(f_stat)
}

# Conduct RI on the f-statistic
ri_balance <- conduct_ri(test_function = balance.test, declaration = declaration,
    assignment = "Contact", sharp_hypothesis = 0, data = data2, sims = 1000)
summary(ri_balance, "upper")
```

```
##             coefficient  estimate upper_p_value null_ci_lower
## 1 Custom Test Statistic 0.3718727         0.783    0.07473702
##   null_ci_upper
## 1      3.268503
```

```
plot(ri_balance)
```




```
### Q6

# Step 1: Obtain a large number of assignment vectors ('proposals') that
# satisfy the condition

restricted_ra <- function() {
    continue <- TRUE
    while (continue) {
        Z <- complete_ra(N = nrow(data2), m = sum(data2$Contact))
        model <- lm_robust(Contact ~ Age + Party + Vote96, data = data2)
        # Extract F statistic
        f_stat <- summary(model)$fstatistic
        # Calculate p-value of F statistic
        p.val <- 1 - pf(q = f_stat[1], df1 = f_stat[2], df2 = f_stat[3])
        if (p.val >= 0.1) {
            return(Z)
        }
    }
}

# Step 2: Generate restricted random assignments
sims <- 1000
permutation_matrix <- replicate(sims, restricted_ra())

# Step 3: Generate weights using the permutation matrix
restricted_design <- declare_ra(permutation_matrix = permutation_matrix)
```

```r
probs <- obtain_condition_probabilities(restricted_design, data2$Contact)
data2$weights <- 1/probs

# Step 4: IV reg using IPW
model_cace1 <- AER::ivreg(Vote98 ~ Contacted | Contact, data = data2, weights = weights)
summary(model_cace1)
```

```
##
## Call:
## AER::ivreg(formula = Vote98 ~ Contacted | Contact, data = data2,
##     weights = weights)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2457 -0.4229 -0.4182  0.6992  1.5060
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.375417   0.008149  46.068  < 2e-16 ***
## Contacted   0.138215   0.042152   3.279  0.00105 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6862 on 7088 degrees of freedom
## Multiple R-Squared: 0.01438, Adjusted R-squared: 0.01424
## Wald test: 10.75 on 1 and 7088 DF,  p-value: 0.001047
```