Problem Set #4

Gaurav Khanna

```
# load packages
library(foreign)
```

```
## Warning: package 'foreign' was built under R version 3.4.4
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.4.4
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
library(experiment)
```

```
## Warning: package 'experiment' was built under R version 3.4.4
```

```
## Warning: package 'MASS' was built under R version 3.4.4
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

Generic multiplot function copied from http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

```
# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }
```

```
 if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

---

1. Potential Outcomes

a. Make up a hypothetical schedule of potential outcomes for three Compliers and three Never-Takers where the ATE is positive but the CACE is negative. By ATE, we mean the average treatment effect for the entire population, including both compliers and never-takers. Note that we can never compute this ATE directly in practice, because we never observe both potential outcomes for any individual, especially for never-takers. That's why this question requires you to provide a complete table of hypothetical potential outcomes for all six subjects.

```
# Simulating the experiment
# Hypothetical schedule of all the observations

# Control
dz0 <- c(rep(0, 6))
# Treatment, Compliars and Never Takers
dz1 <- c(rep(0,3), rep(1, 3))

# Hypothetical measurements

# Output variable in control
yd0 <- c(1, 3, 7, 2, 1, 6)
# Output variable in treatment
yd1 <- c(4, 3, 7, 3, 2, 2)

# Converting to a data frame
exp.df <- data.frame(dz0, dz1, yd0, yd1)

# Treatment effect for each
```

```
exp.df$te <- exp.df$yd1 - exp.df$yd0
exp.df
```

```
##   dz0 dz1 yd0 yd1 te
## 1   0   0   1   4  3
## 2   0   0   3   3  0
## 3   0   0   7   7  0
## 4   0   1   2   3  1
## 5   0   1   1   2  1
## 6   0   1   6   2 -4
```

Answer: The hypothetical schedule is created above

```
# Calculations for ATE & CACE

# ATE estimate based on the hypothetical data
exp.ate <- mean(exp.df$yd1) - mean(exp.df$yd0)
print('ATE')
```

```
## [1] "ATE"
```

```
exp.ate
```

```
## [1] 0.1666667
```

```
# Subset DF for compliars
exp.comp <- exp.df[which(dz1==1),]
print('DF of compliars')
```

```
## [1] "DF of compliars"
```

```
exp.comp
```

```
##   dz0 dz1 yd0 yd1 te
## 4   0   1   2   3  1
## 5   0   1   1   2  1
## 6   0   1   6   2 -4
```

```
# Since we have the entire schedule CACE is just the average treatment effect for the compliars
# We'll use the subset of the compliars created above
exp.cace <- mean(exp.comp$yd1 - exp.comp$yd0)
print('CACE')
```

```
## [1] "CACE"
```

```
exp.cace
```

```
## [1] -0.6666667
```

Answer: If we use the entire schedule (hypothetical) we get an ATE of .166 and the CACE of -.667

###Calculating the ATE estimate without the entire schedule (like in an expreriment, by calculating ATE and then CACE by dividing with the ratio of compliars). This will obviously not match the one we got from the hypothetical schedule

```
# The ratio of compliars is 50%
exp.cace1 = exp.ate/.5
exp.cace1
```

```
## [1] 0.3333333
```

b. Suppose that an experiment were conducted on your pool of subjects. In what ways would the estimated CACE be informative or misleading?

Answer: If we conduct an expriment, with random assignment, the CACE could be misleading as would not represent the actual impact on the compliars. It'll be an incorrect guess based on ITT which assumes that the compliers and not takers are alike. A way to fix this issue would be to estimate the compliers in both treatment and control with a placebo treatment

c. Which population is more relevant to study for future decision making: the set of Compliers, or the set of Compliers plus Never-Takers? Why?

Answer:The population of compliers is more important for future decision making, if the decision is about the effectiveness of the treatment. That's where the information on real effect of the treatment comes from. An example would be trying to measure the efficacy of a medicine/pill *There may be some reasons to look at "never takers". It is to get a idea of the ratio of the 2 groups in the overall population especially while designing the expriment. The overall population (compliers + never takers) may also be directly useful in some experiments where we are interested in the "intent to treat" effect instead of the "actual" treatment effect on the treated. Many expriments to influence policy (like rebates for buying health insurance) fall in this category.*

Turnout to Vote

Suppose that a researcher hires a group of canvassers to contact a set of 1,000 voters randomly assigned to a treatment group. When the canvassing effort concludes, the canvassers report that they successfully contacted 500 voters in the treatment group, but the truth is that they only contacted 250. When voter turnout rates are tabulated for the treatment and control groups, it turns out that 400 of the 1,000 subjects in the treatment group voted, as compared to 700 of the 2,000 subjects in the control group (none of whom were contacted).

a. If you believed that 500 subjects were actually contacted, what would your estimate of the CACE be?

```
# Overall turnout rate in the control group
turn.control <- 700/2000
print('Turnout rate in control')
```

```
## [1] "Turnout rate in control"
```

```
turn.control
```

```
## [1] 0.35
```

```
# Overall turnout rate in the treatment group
turn.treat <- 400/1000
print('Turnout rate in treatment')
```

```
## [1] "Turnout rate in treatment"
```

```
turn.treat
```

```
## [1] 0.4
```

```r
# Effect of intent to treat
itt <- turn.treat - turn.control
print('Effect of Intent to treat')
```

```
## [1] "Effect of Intent to treat"
```

```r
itt
```

```
## [1] 0.05
```

```r
# CACE on assumption that 500 subjects were contacted
cace1 <- itt/(500/1000)
print('CACE with assumption that 500 subjects with contacted')
```

```
## [1] "CACE with assumption that 500 subjects with contacted"
```

```r
cace1
```

```
## [1] 0.1
```

Answer: CACE, on assumption that 500 subjects were contacted, would be .1

b. Suppose you learned that only 250 subjects were actually treated. What would your estimate of the CACE be?

```r
# CACE on assumption that 250 subjects were contacted
cace2 <- itt/(250/1000)
print('CACE with assumption that 250 subjects with contacted')
```

```
## [1] "CACE with assumption that 250 subjects with contacted"
```

```r
cace2
```

```
## [1] 0.2
```

Answer: CACE would be .2 in this case

c. Do the canvassers' exaggerated reports make their efforts seem more or less effective? Define effectiveness either in terms of the ITT or CACE. Why does the definition matter?

Answer: The exaggerated reports make the canvassers' efforts seem less effective. We get a lower estimate for CACE with the exaggerated reports Effectiveness can be defined in terms of either ITT or CACE. The definitions could be as follows: 1. An ITT of greater than 0 i.e. we convert some subjects to vote after just knocking on the doors 2. A CACE of greater than 0 and greater than ITT i.e we convert sonme subjects to vote after they answer the door **The definition matters as it forces us to recognize exactly what we want to measure before the expriment. We'll concentrate on ITT if we're trying to measure the effectiveness of the intent, i.e. knocking on doors We'll concentrate on CACE if our concern is limited to measure the impact of actually talking to people who open the door Defining the primary outcome before the experiment would help pick the right metric for measurement**

---

3. Turnout in Dorms

Guan and Green report the results of a canvassing experiment conduced in Beijing on the eve of a local election. Students on the campus of Peking University were randomly assigned to treatment or control groups. Canvassers attempted to contact students in their dorm rooms and encourage them to vote. No contact with the control group was attempted. Of the 2,688 students assigned to the treatment group, 2,380 were contacted. A total of 2,152 students in the treatment group voted; of the 1,334 students assigned to the control group, 892 voted. One aspect of this experiment threatens to violate the exclusion restriction. At every dorm room they visited, even those where no one answered, canvassers left a leaflet encouraging students to vote.

```
library(foreign)
d3 <- read.dta("./data/Guan_Green_CPS_2006.dta")
head(d3, 20)
```

```
##    turnout contact  dormid treat2
## 1        0       0 1010101      0
## 2        0       0 1010101      0
## 3        0       0 1010101      0
## 4        0       0 1010102      0
## 5        0       0 1010102      0
## 6        0       1 1010103      1
## 7        0       1 1010103      1
## 8        0       1 1010103      1
## 9        1       1 1010103      1
## 10       0       1 1010104      1
## 11       0       1 1010104      1
## 12       0       1 1010104      1
## 13       0       1 1010104      1
## 14       0       0 1010105      0
## 15       0       0 1010105      0
## 16       0       0 1010105      0
## 17       0       0 1010105      0
## 18       0       1 1010106      1
## 19       0       1 1010106      1
## 20       0       1 1010106      1
```

```
# Finding more about the data
nrow(d3)
```

```
## [1] 4024
```

```
summary(d3)
```

```
##     turnout          contact          dormid            treat2
##  Min.   :0.0000   Min.   :0.0000   Min.   : 1010101   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.: 6010146   1st Qu.:0.0000
##  Median :1.0000   Median :1.0000   Median : 9020212   Median :1.0000
##  Mean   :0.7568   Mean   :0.5922   Mean   :10226152   Mean   :0.6685
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:14010136   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :24033068   Max.   :1.0000
##  NA's   :2
```

a. Using the data set from the book's website, estimate the ITT. First, estimate the ITT using the difference in two-group means. Then, estimate the ITT using a linear regression on the appropriate subset of data. *Heads up: There are two NAs in the data frame. Just na.omit to remove these rows.*

```
# Getting key values out of the data

# Number in treated
exp3.treat.n = 2688
# Number that voted from treatment
exp3.treat.v = 2152
# Number that were contacted
exp3.treat.c = 2380

# Number in control
exp3.cont.n = 1334
# Number that voted
exp3.cont.v = 892

# Conversion ratio in treatment
exp3.treat.convert = exp3.treat.v/exp3.treat.n
paste0("Conversion ratio in treatment: ", exp3.treat.convert)
```

```
## [1] "Conversion ratio in treatment: 0.800595238095238"
```

```
# Conversion ratio in control
exp3.cont.convert = exp3.cont.v/exp3.cont.n
paste0("Conversion ratio in control: ", exp3.cont.convert)
```

```
## [1] "Conversion ratio in control: 0.668665667166417"
```

```
# Ratio of actually treated
exp3.treat.ar = exp3.treat.c/exp3.treat.n
paste0("Ratio of actually treated or alpha: ", exp3.treat.ar)
```

```
## [1] "Ratio of actually treated or alpha: 0.885416666666667"
```

ITT using difference in 2 group means.

```
itt.gm = exp3.treat.convert - exp3.cont.convert
paste0("ITT using difference in means: ", itt.gm)
```

```
## [1] "ITT using difference in means: 0.131929570928821"
```

ITT using linear regression

```
# Linear regression
lr3 <- lm(turnout ~ treat2, data=d3, na.action = na.omit)
summary(lr3)
```

```
##
## Call:
## lm(formula = turnout ~ treat2, data = d3, na.action = na.omit)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8006  0.1994  0.1994  0.1994  0.3313
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.66867    0.01162  57.521   <2e-16 ***
## treat2       0.13193    0.01422   9.278   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4246 on 4020 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.02096,    Adjusted R-squared:  0.02072
## F-statistic: 86.08 on 1 and 4020 DF,  p-value: < 2.2e-16
```

```
paste0("ITT with regression: ", lr3$coefficients['treat2'])
```

```
## [1] "ITT with regression: 0.131929570928821"
```

Answer ITT from difference in means: 0.132 ITT from regression: 0.132 (0.01422)

b. Use randomization inference to test the sharp null hypothesis that the ITT is zero for all observations, taking into account the fact that random assignment was clustered by dorm room. Interpret your results.

```
# Pull out the cluster ID's
clusters.all <- unique(d3$dormid)
print('Head-Unique clusters: ')
```

```
## [1] "Head-Unique clusters: "
```

```
head(clusters.all)
```

```
## [1] 1010101 1010102 1010103 1010104 1010105 1010106
```

```
# How many clusters
clusters.n = length(clusters.all)
print('Number of unique clusters/DormID: ')
```

```
## [1] "Number of unique clusters/DormID: "
```

```
clusters.n
```

```
## [1] 1004
```

Generic functions for cluster randomization and ATE

```
# Generic function to pick the cluster ID's for treatment

clusters.in.treatment = clusters.n/2
paste0("Clusters in treatment: ", clusters.in.treatment)
```

```
## [1] "Clusters in treatment: 502"
```

```
randomize.clusters <- function(d) {
  treat.clusters <- sample(x = clusters.all,
                           size = clusters.in.treatment,
                           replace = FALSE)
  # returns 1 if a cluster ID in those that are picked
  return(as.numeric(d$dormid %in% treat.clusters))
}

# Generic function to estimate the ATE
est.ate <- function(result, treat) {
  mean(result[treat==1]) - mean(result[treat==0])
  }
```

```
# Trying out one randomization. Now we have the data prepared for the experiment
# First creating a new DF
d31 <- read.dta("./data/Guan_Green_CPS_2006.dta")
d31$treat2 <- randomize.clusters(d31)
```

```
# Finding out more about the new data set after one randomization
clusters <- group_by(d31, dormid)
clusters.summary <- summarize(clusters, count=n(), mean_y=mean(turnout), mean_treat2=mean(treat2) )
clusters.summary
```

```
## # A tibble: 1,004 x 4
##     dormid count mean_y mean_treat2
##      <int> <int>  <dbl>       <dbl>
##  1 1010101     3   0               0
##  2 1010102     2   0               1
##  3 1010103     4   0.25            1
##  4 1010104     4   0               1
##  5 1010105     4   0               0
##  6 1010106     4   0.25            1
##  7 1010108     4   0               0
##  8 1010109     4   0.25            1
##  9 1010110     4   0.75            1
## 10 1010111     4   1               0
## # ... with 994 more rows
```

###Validating the data after randomization

```
# Validating the data
mean(d31$turnout[d31$treat2 == 1])
```

```
## [1] 0.7621407
```

```
mean(d31$turnout[d31$treat2 == 0])
```

```
## [1] NA
```

```
# Got to remove the Rows with na
nrow(d31)
```

```
## [1] 4024
```

```
d31 <- na.omit(d31)
nrow(d31)
```

```
## [1] 4022
```

```r
print('2 rows with NAS removed')
```

```
## [1] "2 rows with NAS removed"
```

```r
# Now validate
mean(d31$turnout[d31$treat2 == 1])
```

```
## [1] 0.7621407
```

```r
mean(d31$turnout[d31$treat2 == 0])
```

```
## [1] 0.751497
```

```r
# ATE for one randomization
ateR <- est.ate(d31$turnout, d31$treat2)
paste0('ATE with one randomization: ', ateR)
```

```
## [1] "ATE with one randomization: 0.0106437274113814"
```

```r
ateR
```

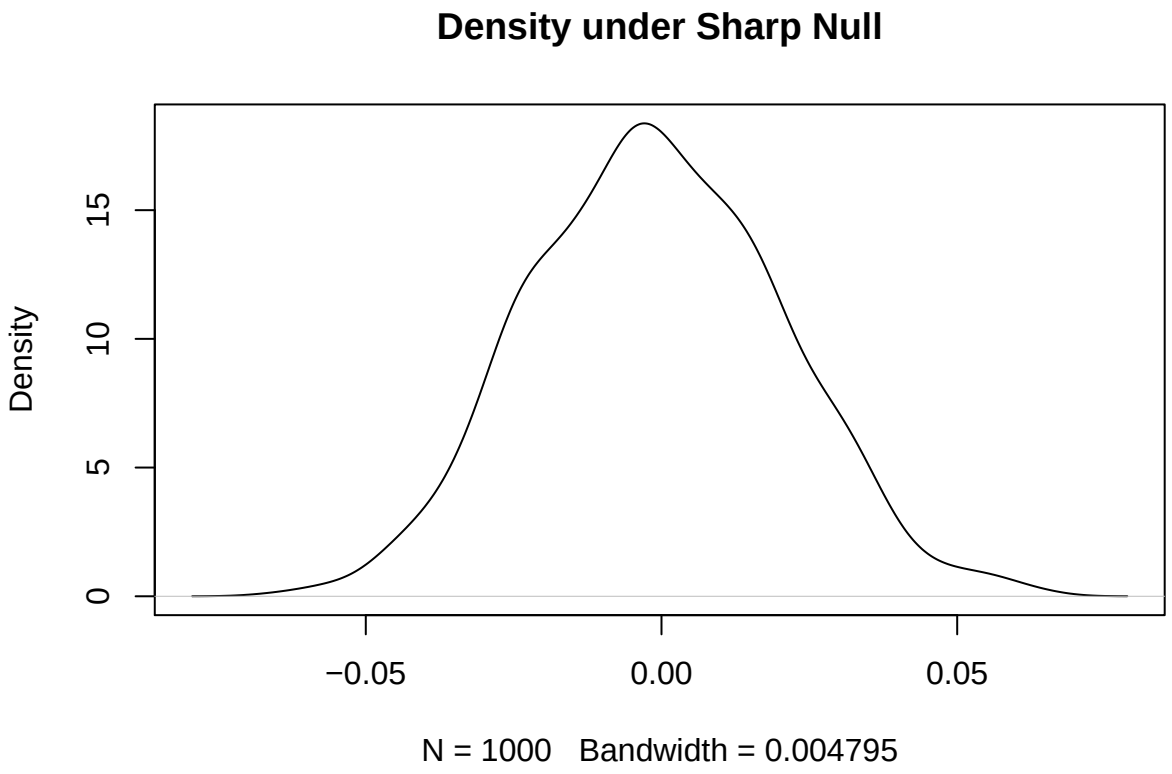
```
## [1] 0.01064373
```

1K Randomizations

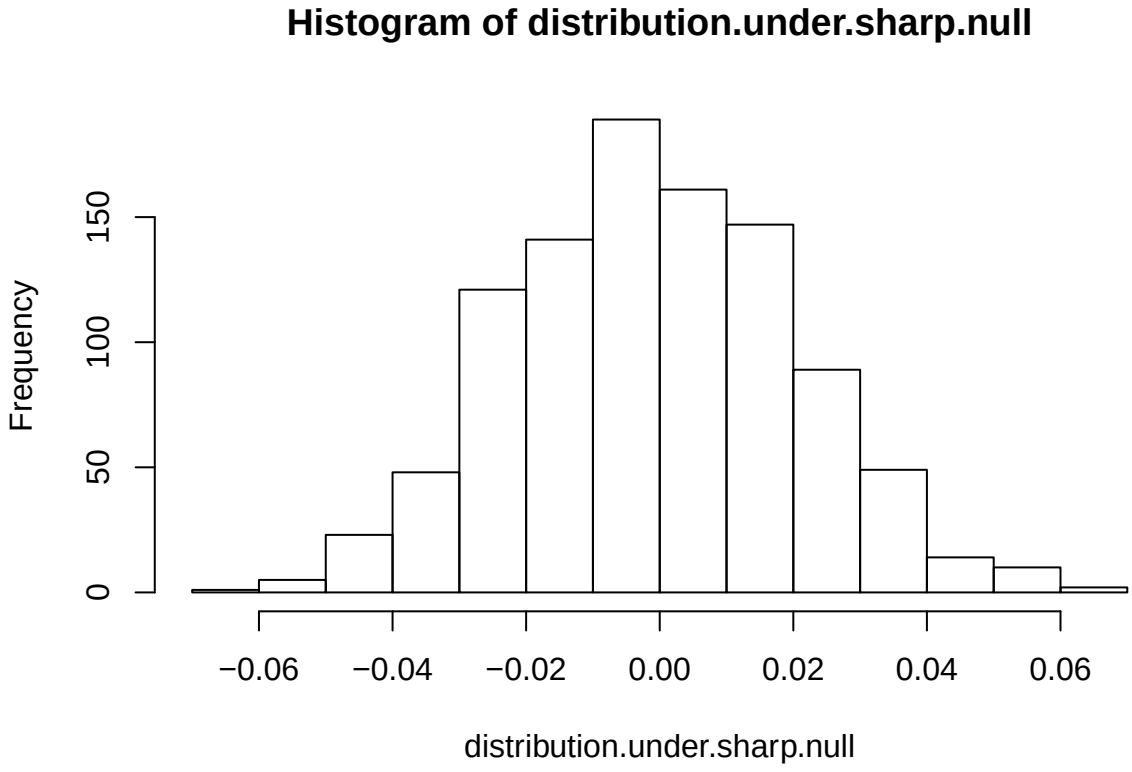
```r
# 1K Randomizations
# Repeating the expriment 10,000 times
distribution.under.sharp.null <- replicate(1000, est.ate(d31$turnout, (randomize.clusters(d31))))
ate.mean <- mean(distribution.under.sharp.null)
paste0("Mean ITT from 1k randomizations: ", ate.mean)
```

```
## [1] "Mean ITT from 1k randomizations: -0.000897012549298192"
```

```r
# Density distribution for the estimates
plot(density(distribution.under.sharp.null),
     main = "Density under Sharp Null")
abline(v = itt.gm, col = "blue")
```




```
# Histogram for the estimates
hist(distribution.under.sharp.null)
```




```
# p value based on the above distribution
p <- mean(distribution.under.sharp.null >= itt.gm )
```

```
p
```

```
## [1] 0
```

Answer: The graphs show that majority of the estimates (under sharp null) lie in range [-.06, .06]. Our observed ATE (#a) was .132 which does not lie on the graph. We should be able to reject the sharp NULL hypothesis. This validates that our observed impact (via the expriments) is highly unlikely to happen by chance. The intervention in dorms does have a positive influence ~13% on turnout

c. Assume that the leaflet had no effect on turnout. Estimate the CACE. Do this in two ways: First, estimate the CACE using means. Second, use some form of linear model to estimate this as well. If you use a 2SLS, then report the standard errors and draw inference about whether the leaflet had any causal effect among compliers.

CACE using means

```
# CACE using the means. ITT/alpha
cace.m <- itt.gm/exp3.treat.ar
paste0('CACE using means: ', cace.m)
```

```
## [1] "CACE using means: 0.14900280951961"
```

CACE using a linear model

We'll split treatment variable into 2 variables for the new linear model:

1. "contact": This already exists in the data. Its when the door is answered
2. "leaflet": Generated independent variable (treat2 * !contact). Its when the door is not answered and a leaflet is dropped **The linear model tries to measure the effect of contact and leaflet separately using the regression below**

```
# Adding a new variable "leaflet"
d3$leaflet <- d3$treat2 * !d3$contact
print('Data with the new variable LEAFLET')
```

```
## [1] "Data with the new variable LEAFLET"
```

```
head(d3, 20)
```

```
##    turnout contact  dormid treat2 leaflet
## 1        0       0 1010101      0       0
## 2        0       0 1010101      0       0
## 3        0       0 1010101      0       0
## 4        0       0 1010102      0       0
## 5        0       0 1010102      0       0
## 6        0       1 1010103      1       0
## 7        0       1 1010103      1       0
## 8        0       1 1010103      1       0
## 9        1       1 1010103      1       0
## 10       0       1 1010104      1       0
## 11       0       1 1010104      1       0
```

```
## 12       0       1 1010104      1       0
## 13       0       1 1010104      1       0
## 14       0       0 1010105      0       0
## 15       0       0 1010105      0       0
## 16       0       0 1010105      0       0
## 17       0       0 1010105      0       0
## 18       0       1 1010106      1       0
## 19       0       1 1010106      1       0
## 20       0       1 1010106      1       0
```

```
summary(d3)
```

```
##     turnout          contact           dormid             treat2
##  Min.   :0.0000   Min.   :0.0000   Min.   : 1010101   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.: 6010146   1st Qu.:0.0000
##  Median :1.0000   Median :1.0000   Median : 9020212   Median :1.0000
##  Mean   :0.7568   Mean   :0.5922   Mean   :10226152   Mean   :0.6685
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:14010136   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :24033068   Max.   :1.0000
##  NA's   :2
##     leaflet
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.07629
##  3rd Qu.:0.00000
##  Max.   :1.00000
##
```

```
lr33 <- lm(turnout ~ contact + leaflet, data=d3, na.action = na.omit)
summary(lr33)
```

```
##
## Call:
## lm(formula = turnout ~ contact + leaflet, data = d3, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8043  0.1957  0.1957  0.2280  0.3313
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.66867    0.01162  57.526  < 2e-16 ***
## contact      0.13562    0.01452   9.341  < 2e-16 ***
## leaflet      0.10332    0.02687   3.845 0.000123 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4245 on 4019 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.02135,    Adjusted R-squared:  0.02086
## F-statistic: 43.83 on 2 and 4019 DF,  p-value: < 2.2e-16
```

```
# paste0("ITT with regression: ", lr33$coefficients['treat2', 'contact'])
```

Answer: CACE using means is 0.14900280951961

The linear model for estimating CACE is as follows:
turnout = intercept + ALPHA(Contact) + BETA(leaflet) + errors

We try to find out the impact of contact and leaflet (when there's no contact but a leaflet is dropped) separately
The coefficients from the model are as follows:
Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.66867 0.01162 57.526 < 2e-16 ***contact 0.13562 0.01452 9.341 < 2e-16*** leaflet 0.10332 0.02687 3.845 0.000123 ***

We can interpret the coefficient of contact as the CACE: .136 (.0145) i.e. the intervention (contact) in dorms has a positive (13%) impact on turnout

The impact of leaflet is: 0.10332 (0.02687)
The model shows that dropping the leaflet also has a significant positive effect on the outcome 0.10332 (0.02687) i.e. around 10%

---

4. Why run a placebo?

Nickerson describes a voter mobilization experiment in which subjects were randomly assigned to one of three conditions: a baseline group (no contact was attempted); a treatment group (canvassers attempted to deliver an encouragement to vote); and a placebo group (canvassers attempted to deliver an encouragement to recycle). Based on the results in the table below answer the following questions

Treatment Assignment	Treated ?	N	Turnout
Baseline	No	2572	31.22%
Treatment	Yes	486	39.09%
Treatment	No	2086	32.74%
Placebo	Yes	470	29.79%
Placebo	No	2109	32.15%

First Use the information to make a table that has a full recovery of this data. That is, make a `data.frame` or a `data.table` that will have as many rows a there are observations in this data, and that would fully reproduce the table above. (*Yes, this might seem a little trivial, but this is the sort of "data thinking" that we think is important.*)

Making the data table

```
# Converting the information in above table to variables
n.base = 2572
n.treat.c = 486
n.treat.nt = 2086
n.placebo.c = 470
n.placebo.nt = 2109
n.base.vote = round(n.base * .3122)
n.treat.c.vote = round(n.treat.c * .3909)
n.treat.nt.vote = round(n.treat.nt * .3274)
n.placebo.c.vote = round(n.placebo.c * .2979)
n.placebo.nt.vote = round(n.placebo.nt * .3215)
```

```r
# Total number of observations
n.observations = n.base + n.treat.c + n.treat.nt + n.placebo.nt + n.placebo.c
paste0('Total number of observations: ', n.observations)
```

```
## [1] "Total number of observations: 7723"
```

```r
# Dataframe corresponding to the observations above
df4 <- data.frame(
                  "baseline" = c(rep(1,n.base), rep(0, n.observations - n.base)),
                  "treatment.c" = c(rep(0, n.base), rep(1, n.treat.c ), rep(0, n.observations - n.base
                  "treatment.nt" = c(rep(0, n.base), rep(0, n.treat.c ), rep(1, n.treat.nt), rep(0, n.ob
                  "placebo.c" = c(rep(0, n.base), rep(0, n.treat.c ), rep(0, n.treat.nt), rep(1, n.plac
                  "placebo.nt" = c(rep(0, n.base), rep(0, n.treat.c ), rep(0, n.treat.nt), rep(0, n.pla
                  # "vote" = c(rbinom(n.base, 1, .3122), rbinom(n.treat.c, 1, .3909), rbinom(n.treat.nt
                  "vote" = c(rep(1, n.base.vote), rep(0, n.base - n.base.vote), rep(1, n.treat.c.vote),
                             rep(1, n.placebo.c.vote), rep(0, n.placebo.c - n.placebo.c.vote), rep(1, n
                  )
paste0('Total number of rows in the dataframe: ', nrow(df4))
```

```
## [1] "Total number of rows in the dataframe: 7723"
```

Validating the generated data. Means of 'vote' variable should match 'turn' out from the aggregate data

```r
# validation
head(df4, 10)
```

```
##    baseline treatment.c treatment.nt placebo.c placebo.nt vote
## 1         1           0            0         0          0    1
## 2         1           0            0         0          0    1
## 3         1           0            0         0          0    1
## 4         1           0            0         0          0    1
## 5         1           0            0         0          0    1
## 6         1           0            0         0          0    1
## 7         1           0            0         0          0    1
## 8         1           0            0         0          0    1
## 9         1           0            0         0          0    1
## 10        1           0            0         0          0    1
```

```r
summary(df4)
```

```
##     baseline      treatment.c       treatment.nt      placebo.c
##  Min.   :0.000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.000   Median :0.00000   Median :0.0000   Median :0.00000
##  Mean   :0.333   Mean   :0.06293   Mean   :0.2701   Mean   :0.06086
##  3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
##    placebo.nt          vote
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000
##  Mean   :0.2731   Mean   :0.3229
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

```r
mean(df4$vote[df4$baseline == 1])
```

```
## [1] 0.3122084
```

```r
mean(df4$vote[df4$treatment.c == 1])
```

```
## [1] 0.3909465
```

```r
mean(df4$vote[df4$treatment.nt == 1])
```

```
## [1] 0.3274209
```

```r
mean(df4$vote[df4$placebo.c == 1])
```

```
## [1] 0.2978723
```

```r
mean(df4$vote[df4$placebo.nt == 1])
```

```
## [1] 0.3214794
```

And they match.

a. Estimate the proportion of Compliers by using the data on the Treatment group. Then compute a second estimate of the proportion of Compliers by using the data on the Placebo group. Are these sample proportions statistically significantly different from each other? Explain why you would not expect them to be different, given the experimental design. (Hint: ITT_D means "the average effect of the treatment on the dosage of the treatment." I.E., it's the contact rate α in the async).

```r
# Estimate of compliers from the treatment group
est.compl.treat = n.treat.c/(n.treat.c + n.treat.nt)
# Estimate of compliers from the placebo group
est.compl.placebo = n.placebo.c/(n.placebo.c + n.placebo.nt)

paste0("Estimate of compliers from the treatment group: ", est.compl.treat)
```

```
## [1] "Estimate of compliers from the treatment group: 0.18895800933126"
```

```r
paste0("Estimate of compliers from the placebo group: ", est.compl.placebo)
```

```
## [1] "Estimate of compliers from the placebo group: 0.182241178751454"
```

```r
# Difference in the complier ratio in the 2 groups
paste0("Difference in complier estimate between treatment and placebo group: ", est.compl.treat - est.c
```

```
## [1] "Difference in complier estimate between treatment and placebo group: 0.00671683057980568"
```

Answer:
Estimate of compliers from the treatment group: 0.18895800933126
Estimate of compliers from the placebo group: 0.182241178751454
Difference in complier estimate between treatment and placebo group: 0.00671683057980568

The proportions are not significantly different from each other
We'd not expect them to be different as the groups "treatment", "placebo" are randomly assigned from the same population.

b. Do the data suggest that Never Takers in the treatment and placebo groups have the same rate of turnout? Is this comparison informative?

Treatment | No | 2086 | 32.74% | Placebo | No | 2109 | 32.15% |

Answer: Yes, the data suggests that Never Takers in treatment and placebo groups have the same rate of turnout. Yes, the comparison is informative. It validates the following: 1. The assignmenet procedure (randomization) has preserved the comparability of the groups 2. The Never takers in the 2 groups behave similarly in the absence of an intervention i.e. the assignment has managed to randomize the impacts of any co-variates

c. Estimate the CACE of receiving the placebo. Is this estimate consistent with the substantive assumption that the placebo has no effect on turnout?

Answer: We can get an estimate of the CACE by the ITT/ITT_D method

```
# CACE of receiving the placebo

# Total turnout in the placebo group
placebo.turn = .2979 * n.placebo.c + .3215 * n.placebo.nt
paste0('Number that voted from the placebo group: ', placebo.turn)
```

```
## [1] "Number that voted from the placebo group: 818.0565"
```

```
placebo.turn.per = placebo.turn/(n.placebo.c + n.placebo.nt)
paste0('Ratio of voters from the placebo group: ', placebo.turn.per)
```

```
## [1] "Ratio of voters from the placebo group: 0.317199108181466"
```

```
# Comparing with the baseline
placebo.itt = placebo.turn.per - .3122
paste0('ITT for the placebo group: ', placebo.itt)
```

```
## [1] "ITT for the placebo group: 0.00499910818146571"
```

```
# CACE using ITT and alpha
paste0('alpha for the placebo group: ', est.compl.placebo)
```

```
## [1] "alpha for the placebo group: 0.182241178751454"
```

```
placebo.cace = placebo.itt/est.compl.placebo
paste0('CACE for the placebo group: ', placebo.cace)
```

```
## [1] "CACE for the placebo group: 0.0274312765957448"
```

Answer: The CACE estimate from the placebo groupo is .027 (~2.7%). With the baseline of 31%, this change does not look significant and could support our assuption that the placebo has little to no effect on the turnout. We cannot be sure though without investigating the errors or the confidence interval

d. Estimate the CACE of receiving the treatment using two different methods. First, use the conventional method of dividing the ITT by the ITT_{D}. (This should be a treatment vs. control comparison.)

```
# ITT for the treatment group
```

```
# Total turnout in the treatment group
treat.turn = n.treat.c * .3909 + n.treat.nt * .3274
paste0('Number that voted from the treatment group: ', treat.turn)
```

```
## [1] "Number that voted from the treatment group: 872.9338"
```

```
treat.turn.per = treat.turn/(n.treat.c + n.treat.nt)
paste0('Ratio of voters from the treatment group: ', treat.turn.per)
```

```
## [1] "Ratio of voters from the treatment group: 0.339398833592535"
```

```
# We compare with the baseline to find the ITT
treat.itt = treat.turn.per - .3122
paste0('ITT for the treatment group: ', treat.itt)
```

```
## [1] "ITT for the treatment group: 0.027198833592535"
```

```
paste0('Alpha for the treatment group: ', est.compl.treat)
```

```
## [1] "Alpha for the treatment group: 0.18895800933126"
```

```
# CACE using ITT and alpha
treat.cace = treat.itt/est.compl.treat
paste0('CACE for the treatment group: ', treat.cace)
```

```
## [1] "CACE for the treatment group: 0.143941152263375"
```

Answer: The CACE estimate for the treatment group is 0.144 (~14.4%)

e. Then, second, compare the turnout rates among the Compliers in both the treatment and placebo groups. Interpret the results.

```
# Turnout difference among compliers in treatment and placebo
treat.cace1 = .39 - .2979
paste0('CACE for the treatment group by comparing with compliers in placebo group: ', treat.cace1)
```

```
## [1] "CACE for the treatment group by comparing with compliers in placebo group: 0.0921"
```

Answer: CACE for the treatment group by comparing with compliers in placebo group is 0.0921.
A 9.2% uplift in the turnout is significant with respect to the baseline turnout (31.22%). It means that ~9.2% more folks show up for voting from the group that answers the door, and gets the encouragement to vote, when compared to folks that answer the door but do not get the encouragement to vote (instead they get encouragement to recycle)
This estimate is lower than what we get from the ITT/ITT_d method. The ITT/ITT_d method over-estimates the CACE as it ignores the differences among the compliers and never takers.

f. Based on what we talked about in class – that the rate of compliance determines whether one or another design is more efficient – given the compliance rate in this study, which design *should* provide a more efficient estimate of the treatment effect? If you want to review the specific paper that makes this claim, check out this link. Does it?

Answer: The rate of complience in this study is ~18% from both the treatment and the placebo groups, close to the 20% line in Fig 3 [GerberGreenKaplanKern.2010]. At this rate

of complience, its noted in the paper (Fig 3) that assigning 40% of the subjects to Placebo group creates the most efficient design Our calculations above show that commparing with the placebo group gives a different estimate than comparing with the baseline. The take-away is that that there's a difference among compliers and not takers that we cannot ignore i.e. comparing with the baseline would not give us an accurate result. A design that compares compliers is necessary and "placebo" treatment seems like a good choice in this specific case. The cost of delivering the placebo is not much different than delivering the treatment, but the accuracy increases when compared to an experiment where we just distribute the placebo group among treatment and baseline

5. Tetris FTW?

A doctoral student conducted an experiment in which she randomly varied whether she ran or walked 40 minutes each morning. In the middle of the afternoon over a period of 26 days she measured the following outcome variables: (1) her weight; (2) her score in Tetris; (3) her mood on a 0-5 scale; (4) her energy; and (5) whether she got a question right on the math GRE.

```
d5 <- read.dta("./data/Hough_WorkingPaper_2010.dta")
head(d5)
```

```
##   day run weight tetris mood energy appetite gre
## 1   1   1     21  11092    3      3        0   1
## 2   2   1     21  14745    3      1        2   0
## 3   3   0     20  11558    3      3        0   1
## 4   4   0     21  11747    3      1        1   1
## 5   5   0     21  14319    2      3        3   1
## 6   6   1     19   7126    3      2        0   1
```

```
tail(d5)
```

```
##    day run weight tetris mood energy appetite gre
## 21  21   1     20  36665    2      3        0   1
## 22  22   0     21   8094    4      3        1   1
## 23  23   1     19  48769    2      5        0   0
## 24  24   1     20  22601    4      4        1   1
## 25  25   1     19  37950    4      4        0   1
## 26  26   1     20  56047    4      4        0   1
```

Getting a visual sense of the data

```
# Library tidyr to gather data for multiple plots
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.4.3
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
# Getting a visual sense of the data

# Run
```

```
p1 <- ggplot(d5, aes(x=day, y=run)) +
    geom_point() +
    geom_smooth(alpha=.2, size=1) +
    ggtitle("Days with a run")

# Weight
p2 <- ggplot(d5, aes(x=day, y=weight)) +
    geom_point(alpha=.3) +
    geom_smooth(alpha=.2, size=1) +
    ggtitle("Weight by the day")


# Mood, appetite, Energy
p3 <- ggplot(d5, aes(x=day)) +
    geom_line(aes(y = mood), color = "blue") +
    geom_line(aes(y = appetite), color = "red") +
    geom_line(aes(y = energy), color = "grey") +
    ggtitle("Mood, appetite and energy by the day")

# GRE
p4 <- ggplot(d5, aes(x=day, y=gre)) +
    geom_point() +
    geom_smooth(alpha=.2, size=1) +
    ggtitle("GRE by days")

# Tetris
p5 <- ggplot(d5, aes(x=day, y=tetris)) +
    geom_point(alpha=.3) +
    geom_smooth(alpha=.2, size=1) +
    ggtitle("Tetris score by the day")


# Plot using the multiplot function
multiplot(p1, p2, p3, p4, p5, cols=3)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

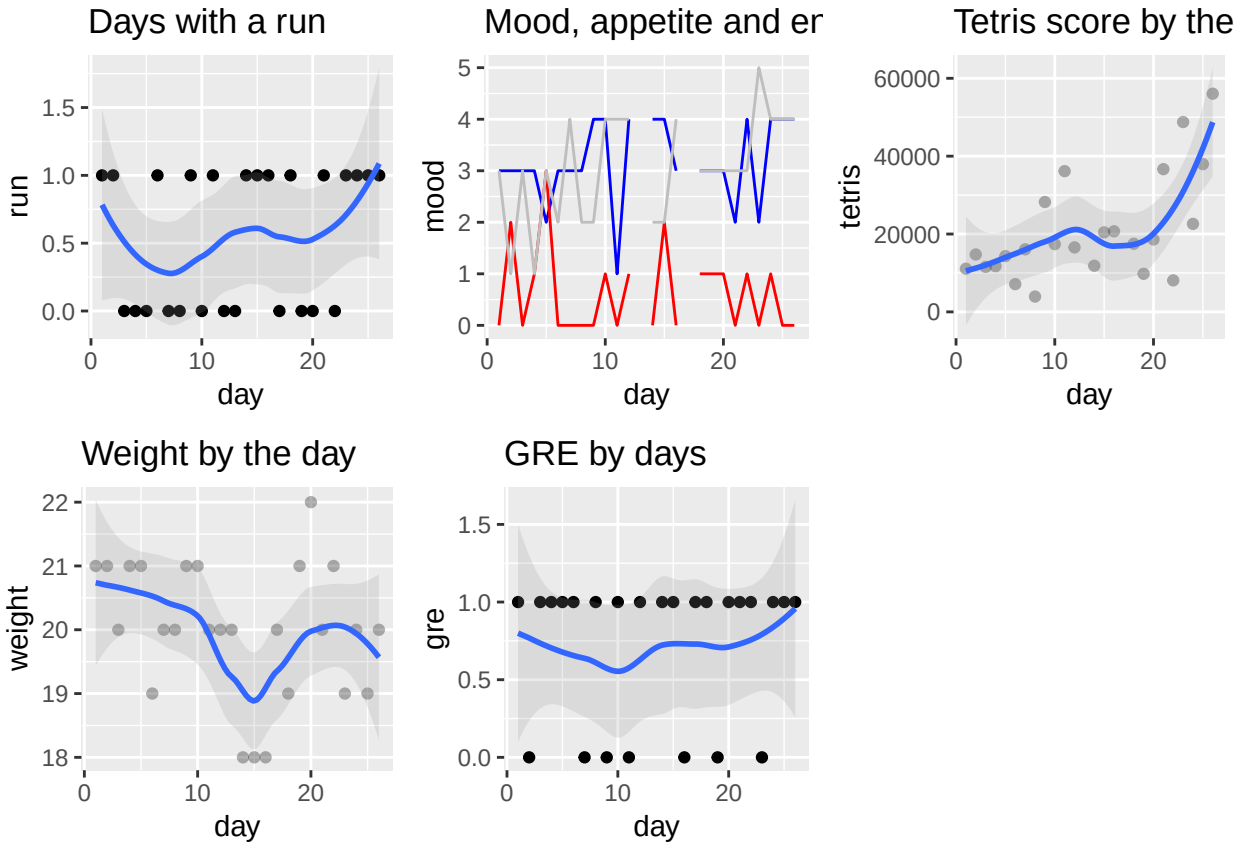
```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



a. Suppose you were seeking to estimate the average effect of running on her Tetris score. Explain the assumptions needed to identify this causal effect based on this within-subjects design. Are these assumptions plausible in this case? What special concerns arise due to the fact that the subject was conducting the study, undergoing the treatments, and measuring her own outcomes?

We'll have to make the following assumptions:
1. **"No Anticipation":** The student's tetris scores and other variables are not influenced by the anticipation of upcoming run. Yes, this assumption is plausibe with self control and mental descipline on part of the student (like not consuming more carbs to pepare for the run or controlling the anxiety/thrill of the upcoming walk/run) 2. **"No Persistence (or no spillover)":** The student's run on one day does not impact the scores and other variables in the future. This does not look very plausible. Science says that the impact of exercise and diet are cumulative. If its really true that the scores are impacted by exercise, then the connection should persist for a while.

There's concerns around the subject conducting the study on herself: 1. **"No Anticipation"** becomes really hard to pull off. The mental boost of anticipating an enjoyable walk or eating more to prepare would be hard to ignore. A better expriment is done with student not being informed of the upcomign treatment(run) 2. The accuracy of account keeping (taking scores, walk times etc) is also suspect with the subject being the experimenter. The bookkeeping rigor may or may not be present

b. Estimate the effect of running today on Tetris score. What is the ATE?

One way to find the ATE would be to compare the means for days when the student does or does not run We can use the earlier define est.ate function

```r
# est.ate(results, treat)
ate.run.tet = est.ate(d5$tetris, d5$run)
paste0('Effect of running on Tetris score: ', ate.run.tet)
```

```
## [1] "Effect of running on Tetris score: NA"
```

```r
# The result is NA. Must be some NA's in the data. Lets find out
summary(d5)
```

```
##       day             run             weight          tetris
##  Min.   : 1.00   Min.   :0.0000   Min.   :18.00   Min.   : 3939
##  1st Qu.: 7.25   1st Qu.:0.0000   1st Qu.:19.25   1st Qu.:11700
##  Median :13.50   Median :1.0000   Median :20.00   Median :16982
##  Mean   :13.50   Mean   :0.5385   Mean   :20.00   Mean   :20747
##  3rd Qu.:19.75   3rd Qu.:1.0000   3rd Qu.:21.00   3rd Qu.:24008
##  Max.   :26.00   Max.   :1.0000   Max.   :22.00   Max.   :56047
##                                                   NA's   :2
##       mood           energy         appetite          gre
##  Min.   :1.000   Min.   :1.000   Min.   :0.000   Min.   :0.00
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:0.000   1st Qu.:0.00
##  Median :3.000   Median :3.000   Median :0.000   Median :1.00
##  Mean   :3.167   Mean   :3.042   Mean   :0.625   Mean   :0.72
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:1.000   3rd Qu.:1.00
##  Max.   :4.000   Max.   :5.000   Max.   :3.000   Max.   :1.00
##  NA's   :2       NA's   :2       NA's   :2       NA's   :1
```

```r
# Removing the NA's
d5 = na.omit(d5)

summary(d5)
```

```
##       day             run             weight       tetris
##  Min.   : 1.00   Min.   :0.0000   Min.   :18   Min.   : 3939
##  1st Qu.: 6.75   1st Qu.:0.0000   1st Qu.:19   1st Qu.:11700
##  Median :13.00   Median :1.0000   Median :20   Median :16982
##  Mean   :13.38   Mean   :0.5833   Mean   :20   Mean   :20747
##  3rd Qu.:20.25   3rd Qu.:1.0000   3rd Qu.:21   3rd Qu.:24008
##  Max.   :26.00   Max.   :1.0000   Max.   :22   Max.   :56047
##       mood           energy         appetite          gre
##  Min.   :1.000   Min.   :1.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :3.000   Median :3.000   Median :0.000   Median :1.0000
##  Mean   :3.167   Mean   :3.042   Mean   :0.625   Mean   :0.7083
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :4.000   Max.   :5.000   Max.   :3.000   Max.   :1.0000
```

```r
# est.ate(results, treat)
ate.run.tet = est.ate(d5$tetris, d5$run)
paste0('Effect of running on Tetris score: ', ate.run.tet)
```

```
## [1] "Effect of running on Tetris score: 13613.1"
```

Another method would be to compare means of differences from the previous day for days that the student runs or not

```r
# Adding a column to compare with the previous day
d5$tetris_diff = d5$tetris - lag(d5$tetris)
```

```
head(d5)
```

```
##   day run weight tetris mood energy appetite gre tetris_diff
## 1   1   1     21  11092    3      3        0   1          NA
## 2   2   1     21  14745    3      1        2   0        3653
## 3   3   0     20  11558    3      3        0   1       -3187
## 4   4   0     21  11747    3      1        1   1         189
## 5   5   0     21  14319    2      3        3   1        2572
## 6   6   1     19   7126    3      2        0   1       -7193
```

```
d51 <- na.omit(d5)
head(d51)
```

```
##   day run weight tetris mood energy appetite gre tetris_diff
## 2   2   1     21  14745    3      1        2   0        3653
## 3   3   0     20  11558    3      3        0   1       -3187
## 4   4   0     21  11747    3      1        1   1         189
## 5   5   0     21  14319    2      3        3   1        2572
## 6   6   1     19   7126    3      2        0   1       -7193
## 7   7   0     20  16067    3      4        0   0        8941
```

```
# est.ate(results, treat)
ate.run.tet1 = est.ate(d51$tetris_diff, d51$run)
paste0('Effect of running on Tetris score: ', ate.run.tet1)
```

```
## [1] "Effect of running on Tetris score: 14341.3230769231"
```

Answer: The ATE from the first method is 13613.1. Seems running has a positive impact of ~13613 points *The 2nd method gives an ATE estimate of 14341*

c. One way to lend credibility to with-subjects results is to verify the no-anticipation assumption. Construct a regression using the variable `run` to predict the `tetris` score *on the preceding day*. Presume that the randomization is fixed. Why is this a test of the no-anticipation assumption? Does a test for no-anticipation confirm this assumption?

```
# We'll add a column with the previous days tetris scores
d5$tetris_lag = lag(d5$tetris)
head(d5)
```

```
##   day run weight tetris mood energy appetite gre tetris_diff tetris_lag
## 1   1   1     21  11092    3      3        0   1          NA         NA
## 2   2   1     21  14745    3      1        2   0        3653      11092
## 3   3   0     20  11558    3      3        0   1       -3187      14745
## 4   4   0     21  11747    3      1        1   1         189      11558
## 5   5   0     21  14319    2      3        3   1        2572      11747
## 6   6   1     19   7126    3      2        0   1       -7193      14319
```

```
# Regression to predict tetris score on the preceding day
lr5c <- lm(tetris_lag ~ run, data=d5, na.action = na.omit)
summary(lr5c)
```

```
##
## Call:
## lm(formula = tetris_lag ~ run, data = d5, na.action = na.omit)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -15470  -7478  -2842   2242  29360
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18957.8     3655.0   5.187 3.86e-05 ***
## run            450.8     4861.6   0.093    0.927
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11560 on 21 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.0004093,  Adjusted R-squared:  -0.04719
## F-statistic: 0.008599 on 1 and 21 DF,  p-value: 0.927
```

The coefficient of run is (450.8) 4861.6, which is not significant (practically and statisticaly). We can conclude that data does not show any effect of anticipation The model tests if next day's run influences the tetris scores for the day i.e. if there's an impact of anticipating the run. Yes, it tests the "no-anticipation" assumption. yes, we can confirm that the assumption holds based on the results

d. Now let's use regression to put a standard error on our ATE estimate from part (b). Regress Tetris score on the the variable run, this time using the current rather than the future value of run. Is the impact on Tetris score statistically significant?

```r
# Regression of tetris score on run
lr5d = lm(tetris~run, data=d5, na.action = na.omit)
summary(lr5d)
```

```
##
## Call:
## lm(formula = tetris ~ run, data = d5, na.action = na.omit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -19294  -6707  -1154   4890  29628
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    12806       3708   3.453  0.00226 **
## run            13613       4856   2.804  0.01035 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11730 on 22 degrees of freedom
## Multiple R-squared:  0.2632, Adjusted R-squared:  0.2297
## F-statistic:  7.86 on 1 and 22 DF,  p-value: 0.01035
```

Answer: We see an ATE (coefficient of run) = 13613 (4856). Yes, the impact of run on tetris is statistically significant (ATE is > 2 SE's).

e. If Tetris responds to exercise, one might suppose that energy levels and GRE scores would as well. Are these hypotheses borne out by the data?

```
# Regressing GRE score on the variable run
lr5e = lm(gre~run, data=d5, na.action = na.omit)
summary(lr5e)
```

```
##
## Call:
## lm(formula = gre ~ run, data = d5, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8000 -0.6429  0.2000  0.3571  0.3571
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8000     0.1479   5.408 1.97e-05 ***
## run          -0.1571     0.1937  -0.811    0.426
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4678 on 22 degrees of freedom
## Multiple R-squared:  0.02905,    Adjusted R-squared:  -0.01508
## F-statistic: 0.6583 on 1 and 22 DF,  p-value: 0.4259
```

Answer: The GRE score is not effected by "run" *run -0.1571 (0.1937)*

```
# Regressing "energy" score on the variable "run""
lr5e1 = lm(energy~run, data=d5, na.action = na.omit)
summary(lr5e1)
```

```
##
## Call:
## lm(formula = energy ~ run, data = d5, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0714 -1.0179  0.0000  0.9286  1.9286
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.00000    0.33662   8.912  9.4e-09 ***
## run          0.07143    0.44074   0.162    0.873
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.064 on 22 degrees of freedom
## Multiple R-squared:  0.001192,   Adjusted R-squared:  -0.04421
## F-statistic: 0.02627 on 1 and 22 DF,  p-value: 0.8727
```

Answer: "energy" is not effected by "run" *run 0.07143(0.44074)*

Answer: "GRE score" and "energy" do not seem to respond to exercise. The ATE for both are smaller than one standard deviation. The values are also practically insignificant

f. Suppose the student decides to publish her results on Tetris, since she finds those most interesting. In the paper she writes, she chooses to be concise by ignoring the data she collected on energy levels and GRE scores, since she finds those results less interesting. How might you criticize the student's decision? What trap may she have fallen into?

Answer: One can say that the student cherry picked the outcome variable that showed statistical significance. She designed the experiment for multiple outcome measurements(scores, GRE results, energy etc), so an honest report would present the result for all. Looking at all the outcome measurements, one can conclude that the effect on any one could be by chance or not
The student may be doing a form of "p hacking" or data dredging just by ignorance. With multiple outcomes measurements, a small % (1 in 20) may show statistical significance just by chance. Tetris scores could be that variable. She can avoid this trap by looking at the result of "Tetris scores" in the context of similar measurements like "energy" and "GRE results"

g. After submitting her paper to a journal, the student thinks of another hypothesis. What if running has a relatively long-lasting effect on Tetris scores? Perhaps both today's running and yesterday's running will affect Tetris scores. Run a regression of today's Tetris score on both today's `run` variable and yesterday's `run` variable. How does your coefficient on running today compare with what you found in part (d)? How do you interpret this comparison?

```
# Adding a lag variable to the data set
d5$run_lag = lag(d5$run)
head(d5)
```

```
##   day run weight tetris mood energy appetite gre tetris_diff tetris_lag
## 1   1   1     21  11092    3      3        0   1          NA         NA
## 2   2   1     21  14745    3      1        2   0        3653      11092
## 3   3   0     20  11558    3      3        0   1       -3187      14745
## 4   4   0     21  11747    3      1        1   1         189      11558
## 5   5   0     21  14319    2      3        3   1        2572      11747
## 6   6   1     19   7126    3      2        0   1       -7193      14319
##   run_lag
## 1      NA
## 2       1
## 3       1
## 4       0
## 5       0
## 6       0
```

```
# Regression on run for current and previous day
lr5g = lm(tetris~run + run_lag, data=d5, na.action = na.omit)
summary(lr5g)
```

```
##
## Call:
## lm(formula = tetris ~ run + run_lag, data = d5, na.action = na.omit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -20528  -6984  -1121   5180  28496
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12868.3     4767.3   2.699   0.0138 *
## run          14785.8     4961.9   2.980   0.0074 **
## run_lag       -103.2     4961.9  -0.021   0.9836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11770 on 20 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.3085, Adjusted R-squared:  0.2393
## F-statistic: 4.461 on 2 and 20 DF,  p-value: 0.02501
```

```
confint(lr5g, "run", level=.95)
```

```
##        2.5 %   97.5 %
## run 4435.401 25136.17
```

```
confint(lr5g, "run_lag", level=.95)
```

```
##            2.5 %   97.5 %
## run_lag -10453.6 10247.17
```

Answer: The coefficient of run in this regression is 14785.8(4961.9). The coefficient in "d" was 13613 (4856). Both the coefficient and the SE increases by ~10%. This difference is 1/3rd the standard error and does not look significant.
An interpretation of the regression is that previous days' run does not have a significant impact on Tetris scores. This can be a validation for our "no persistence" assumption.

#h. (optional) Note that the observations in our regression are not necessarily independent of each other. An individual might have serially correlated outcomes, regardless of treatment. For example, I might find that my mood is better on weekends than on weekdays, or I might find that I'm terrible at playing Tetris in the few days before a paper is due, but I get better at the game once my stress level has lowered. In computing standard errors for a regression, OLS assumes that the observations are all independent of each other. If they are positively serially correlated, it's possible that OLS will underestimate the standard errors.

To check this, let's do randomization inference in the regression context. Recall that the idea of randomization inference is that under the sharp null hypothesis, we can re-randomize, recompute the ATE, and get approximately the right answer (zero) for the treatment effect. So, returning to the regression we ran in part (g), please generate 1000 new randomizations of the `run` variable, use those to replace the current and lagged values of `run` in your dataset, then run the regression again. Record the coefficient you get on the contemporaneous value of `run`, and repeat this re-randomization exercise 1000 times. Plot the distribution of beta. What are the 2.5% and 97.5% quantiles? How do they compare with the width of the 95% confidence interval you got for your main `run` coefficient in the regression in part (g)?

```
# Trying a randomization of the run variable, calculating lag and then repeating the regression

# Creating new variables for randomizing run and run_lag
d5$run_rnd = sample(d5$run)
d5$run_lag_rnd = lag(d5$run_rnd)
head(d5)
```

```
##   day run weight tetris mood energy appetite gre tetris_diff tetris_lag
## 1   1   1     21  11092    3      3        0   1          NA         NA
## 2   2   1     21  14745    3      1        2   0        3653      11092
## 3   3   0     20  11558    3      3        0   1       -3187      14745
```

```
## 4   4   0     21  11747    3      1        1   1         189      11558
## 5   5   0     21  14319    2      3        3   1        2572      11747
## 6   6   1     19   7126    3      2        0   1       -7193      14319
##   run_lag run_rnd run_lag_rnd
## 1      NA       0          NA
## 2       1       0           0
## 3       1       1           0
## 4       0       1           1
## 5       0       1           1
## 6       0       1           1
```

```
tail(d5)
```

```
##    day run weight tetris mood energy appetite gre tetris_diff tetris_lag
## 21  21   1     20  36665    2      3        0   1       18067      18598
## 22  22   0     21   8094    4      3        1   1      -28571      36665
## 23  23   1     19  48769    2      5        0   0       40675       8094
## 24  24   1     20  22601    4      4        1   1      -26168      48769
## 25  25   1     19  37950    4      4        0   1       15349      22601
## 26  26   1     20  56047    4      4        0   1       18097      37950
##    run_lag run_rnd run_lag_rnd
## 21       0       0           1
## 22       1       1           0
## 23       0       0           1
## 24       1       1           0
## 25       1       1           1
## 26       1       1           1
```

```
# Running the regression with the new randomization
# Regression on run for current and previous day
lr5h = lm(tetris~run_rnd + run_lag_rnd, data=d5, na.action = na.omit)
summary(lr5h)
```

```
##
## Call:
## lm(formula = tetris ~ run_rnd + run_lag_rnd, data = d5, na.action = na.omit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -17007 -10259   1444   6721  31928
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    17597       6286   2.799   0.0111 *
## run_rnd        -2667       5932  -0.450   0.6578
## run_lag_rnd     9189       5840   1.574   0.1313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13040 on 20 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1521, Adjusted R-squared:  0.06735
## F-statistic: 1.794 on 2 and 20 DF,  p-value: 0.192
```

The effect of running is not significant for this randomization

```r
# Generic function to randomly(pseudo) pick the cluster ID's
coeff_run <- function(d) {
  d$run_rnd = sample(d$run)
  d$run_lag_rnd = lag(d$run_rnd)
  lrh = lm(tetris~run_rnd + run_lag_rnd, data=d, na.action = na.omit)
  return(lrh$coefficients['run_rnd'])
}

# Trying another randomization
coeff_run(d5)
```

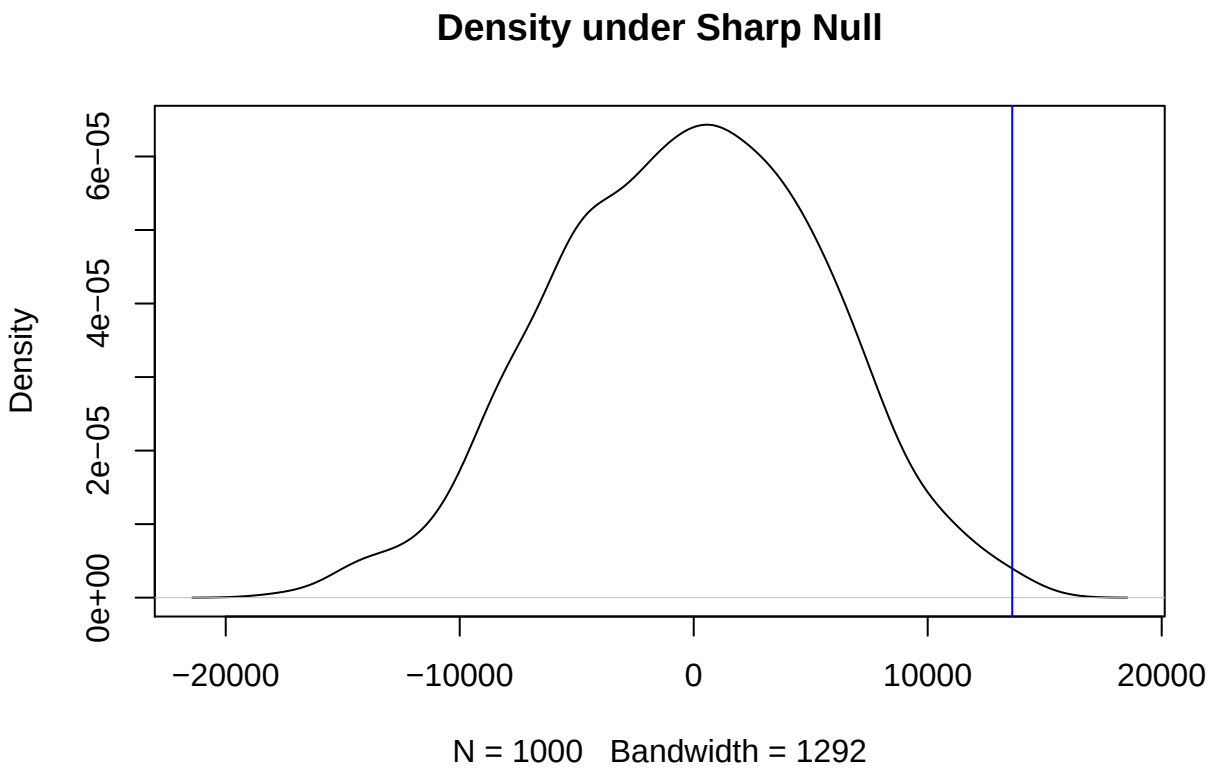
```
##  run_rnd
## 9417.699
```

```r
# 1K Randomizations
h.distribution.under.sharp.null <- replicate(1000, coeff_run(d5))
h.ate.mean <- mean(h.distribution.under.sharp.null)
paste0("Mean ITT fromm 1k randomizations: ", h.ate.mean)
```

```
## [1] "Mean ITT fromm 1k randomizations: -260.737774906895"
```

```r
# Graph for the estimates
plot(density(h.distribution.under.sharp.null),
     main = "Density under Sharp Null")

# Adding our original ATE to the plot
abline(v = ate.run.tet, col = "blue")
```




```r
# p value
pv = mean(h.distribution.under.sharp.null >= ate.run.tet)
paste0('p value: ', pv)
```

```
## [1] "p value: 0.003"
```

```
# num of assignments that generate an estimated ATE at least as large as the actual
n <- sum(h.distribution.under.sharp.null >= ate.run.tet)
paste0("Number of assignments that generate an ATE at least as large as what we got from the experiment
```

```
## [1] "Number of assignments that generate an ATE at least as large as what we got from the experiment
```

This does show that our results from the experiment are not easily reproducible by chance

```
# Standard error
se = sd(h.distribution.under.sharp.null)/sqrt(length(h.distribution.under.sharp.null))
paste0('Standard error: ', se)
```

```
## [1] "Standard error: 180.791720257274"
```

```
# 2.5 and 97.5 quantiles
cint = c(h.ate.mean - 2*se, h.ate.mean + 2*se)
cint
```

```
## [1] -622.3212  100.8457
```

This is a much tighter confidence interval(almost 1/10 of the one we get from the experiment) and does not include the ATE for run from the experiment. Our takeaway could be that the results from the experiment cannot happen by chance