w241_final_project

```
library(stringr)
library(dplyr)
library(stargazer)
library(lmtest)
library(sandwich)
library(ggplot2)
library(magrittr)
library(lmtest)
# install.packages(c('ri2','tidyr','RItools'))
# library(ri2)
library(data.table)
library(tidyr)
library(RItools)
```

Loading Data

```
d = read.csv('PrelimRESULTS.csv')
#g1 = read.csv('~/Desktop/group1.csv')
#g2_sms = read.csv('~/Desktop/group2_SMS.csv')
#g2_no_sms = read.csv('~/Desktop/group2_NoSMS.csv')
s_prior = read.csv('scripts_prior.csv')
s_during = read.csv('scripts_during.csv')
```

Examine data

Combining the PrelimRESULTS table with the two scripts table using left join.

```
# find all carer patient relationship
carer_patient = rbind(s_prior,s_during) %>%
  mutate(Patient.self = UNIQUE.ID) %>%
  gather(., Patient, Patient_id, Patient.1:Patient.self, factor_key=TRUE) %>%
  filter(!is.na(Patient_id)) %>% data.table()

# create a list of distinct patient id
patient_id = unique(carer_patient[Patient != 'Patient.self',Patient_id])
# create a list of distinct carer id
carer_id = unique(carer_patient$UNIQUE.ID)

# clean scripts_prior file
s_p = s_prior %>%
  transmute(
    id = UNIQUE.ID,
    date = DATE,
    total_meds = Medications.ordered,
    self_meds = Ordered.for.self,
    other_meds = ORDERED.FOR.OTHER) %>%
  filter(as.Date(s_prior$DATE,'%d/%m/%Y') >= as.Date('5/8/2019','%d/%m/%Y')) %>% # filter for prior thr
```

```
  group_by(id) %>%
  summarise(
    total_meds_3m = sum(total_meds),
    self_meds_3m = sum(self_meds),
    other_meds_3m = sum(other_meds)
  )

# clean scripts_during file
s_d = s_during %>%
  transmute(
    id = UNIQUE.ID,
    date = DATE,
    total_meds = Medications.ordered,
    self_meds = Ordered.for.self,
    other_meds = ORDERED.FOR.OTHER) %>%
  group_by(id) %>%
  summarise(
    total_meds_in_exp = sum(total_meds),
    self_meds_in_exp = sum(self_meds),
    other_meds_in_exp = sum(other_meds)
  )

# clean the final data file
df = d %>%
  transmute(id = UniqueID,
            treatment = ifelse(TREATMENT == '', '', ifelse(str_detect(TREATMENT,'CONTROL'), 0, 1)),
            phone = ifelse(TREATMENT == '', '', ifelse(str_detect(TREATMENT,'GROUP 1'), 0, 1)),
            sms = ifelse(TREATMENT == '', '', ifelse(str_detect(TREATMENT,'GROUP 1'), 0, ifelse(str_dete
            active_in_exp = ifelse(is.na(ACTIVE_IN._EXPERIMENT), 0, 1),
            active_3m = ifelse(is.na(X3MONTH_ACTIVE.NEW.), 0, 1), # Use the new column!
            voucher = ifelse(VOUCHER == '', 0, 1),
            drug_count = ifelse(is.na(Drug.Count), 0, 1),
            nov05_fail = ifelse(is.na(Nov05_FAIL), 0, 1),
            nov13_fail = ifelse(is.na(Nov13_FAIL), 0, 1),
            nov23_fail = ifelse(is.na(Nov23_FAIL), 0, 1),
            unsub = ifelse(Unsubscribe_Request == '', 0, 1),
            msg_fail = ifelse(is.na(Other_msg_fail), 0, 1),
            nursing_home = ifelse(is.na(Nursing_HOME), 0, 1),
            male = ifelse(is.na(MALE), 0, 1),
            reminder_type = Reminder.Type,
            address_group = Address.GROUP,
            address_group_i = ifelse(Address.GROUP == 0, 0, 1), # create an indicator variable used for
            carer_group = ifelse(is.na(CARER.GROUP), 0, CARER.GROUP),
            carer_group_i = ifelse(is.na(CARER.GROUP), 0, 1), # create an indicator variable used for c
            carer_status = ifelse(UniqueID %in% patient_id, 0, 1), # find non-carers
            cell_phone = HAS_CELL_PHONE,
            email = HAS_EMAIL,
            cell_dup = ifelse(is.na(CELL_DUP_GP), 0, 1),
            name_dup = ifelse(is.na(NAME_DUP_GP), 0, 1)
            ) %>%
  filter(treatment != '') %>%
  filter(!duplicated(id)) %>% # removing duplicated row, because some users converted reminder type hal
  filter(carer_status == 1) # removing all non-carer rows
```

```
df = df %>%
  left_join(.,s_p) %>%
  left_join(.,s_d) %>% data.table()
```

```
## Joining, by = "id"
## Joining, by = "id"
```

```
# Note that at this point, each row represents a carer group, with the carer's id as the unique id
```

Covariate Balance

```
group_assignment = df %>%
  select(id, treatment, phone, sms) %>%
  group_by(treatment, phone, sms) %>%
  summarise(count = n())
group_assignment
```

```
## # A tibble: 6 x 4
## # Groups:   treatment, phone [4]
##   treatment phone sms   count
##   <chr>     <chr> <chr> <int>
## 1 0         0     0        13
## 2 0         1     0        22
## 3 0         1     1        23
## 4 1         0     0        15
## 5 1         1     0        44
## 6 1         1     1        45
```

Note that I used carer_group_i and address_group_i to check covariate balance. I couldn't use the original variables, because the id's were interpretated as numeric. So I had to create these two binary dummy variable.

```
df$treatment = as.integer(df$treatment)
df$sms = as.integer(df$treatment)

# check covariant balance in the no phone group, with respect to treatment
xBalance(treatment ~ male + drug_count + active_3m + carer_group_i + address_group_i,
         data = df[phone == 0,],
         report = 'all')
```

```
##                 strata     unstrat
##                 stat   treatment=0 treatment=1  adj.diff adj.diff.null.sd  std.diff      z
## vars
## male                      3.85e-01    4.00e-01  1.54e-02         1.88e-01  3.04e-02  8.16e-02
## drug_count                1.00e+00    1.00e+00  1.12e-16         1.29e-16  Inf       0.00e+00
## active_3m                 1.54e-01    1.33e-01  -2.05e-02        1.35e-01  -5.65e-02 -1.52e-01
## carer_group_i             0.00e+00    6.67e-02  6.67e-02         7.16e-02  3.52e-01  9.31e-01
## address_group_i           0.00e+00    2.00e-01  2.00e-01         1.19e-01  6.58e-01  1.68e+00  .
## ---Overall Test---
##          chisquare df p.value
## unstrat      3.09  4   0.542
## ---
## Signif. codes:  0 '***' 0.001 '** ' 0.01 '*  ' 0.05 '.  ' 0.1 '   ' 1
```

```r
# check covariant balance in the has phone group, with respect to treatment
xBalance(treatment ~ male + drug_count + active_3m + carer_group_i + address_group_i,
         data = df[phone == 1,],
         report = 'all')
```

```
##                 strata     unstrat
##                 stat   treatment=0 treatment=1 adj.diff adj.diff.null.sd std.diff    z
## vars
## male                       0.3333      0.3483   0.0150           0.0872   0.0313  0.1719
## drug_count                 0.9333      0.8989   -0.0345          0.0524   -0.1200 -0.6573
## active_3m                  0.3333      0.4045   0.0712           0.0891   0.1458  0.7983
## carer_group_i              0.0667      0.0899   0.0232           0.0504   0.0840  0.4607
## address_group_i            0.2444      0.2921   0.0477           0.0821   0.1060  0.5810
## ---Overall Test---
##         chisquare df p.value
## unstrat      1.74  5   0.884
## ---
## Signif. codes:  0 '***' 0.001 '** ' 0.01 '*  ' 0.05 '.  ' 0.1 '   ' 1
```

```r
# check covariant balance in the has phone group, with respect to sms
xBalance(sms ~ male + drug_count + active_3m + carer_group_i + address_group_i,
         data = df[phone == 1,],
         report = 'all')
```

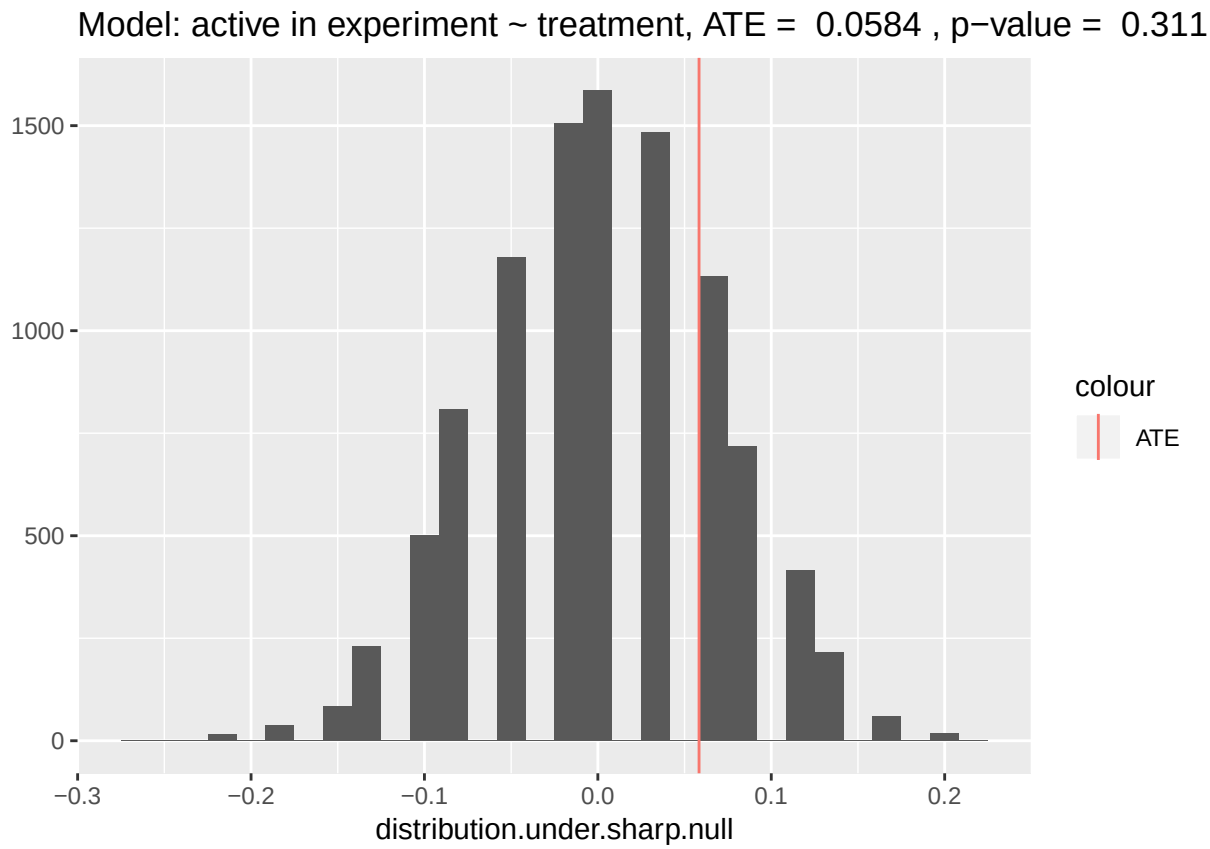
```
##                 strata unstrat
##                 stat     sms=0   sms=1 adj.diff adj.diff.null.sd std.diff    z
## vars
## male                   0.3333  0.3483   0.0150           0.0872   0.0313  0.1719
## drug_count             0.9333  0.8989   -0.0345          0.0524   -0.1200 -0.6573
## active_3m              0.3333  0.4045   0.0712           0.0891   0.1458  0.7983
## carer_group_i          0.0667  0.0899   0.0232           0.0504   0.0840  0.4607
## address_group_i        0.2444  0.2921   0.0477           0.0821   0.1060  0.5810
## ---Overall Test---
##         chisquare df p.value
## unstrat      1.74  5   0.884
## ---
## Signif. codes:  0 '***' 0.001 '** ' 0.01 '*  ' 0.05 '.  ' 0.1 '   ' 1
```

Hypothesis 0: treatment effect, y = active login, x = treatment

Random Inference: active login in exp ~ treatment, without correcting for clustered standard error

```r
# Define potential outcomes for control and treatment, under the sharp null hypothesis, these two vecto

# Assign treatment and control potential outcomes
po.control = as.integer(df$active_in_exp)
po.treatment = po.control

# Randomly assign units into control and treatment group
randomize = function() {sample(c(rep(0,sum(df$treatment == 0)),rep(1,sum(df$treatment == 1))))}
t = randomize()
```

```r
# Calculate the outcome according to control/treatment assignment
outcomes = po.treatment * t + po.control*(1-t)

# Estimate the Average Treatment Effect
est.ate = function(outcome,success){mean(outcome[success==1]) - mean(outcome[success==0])}
#est.ate(outcomes, randomize())

# Run simulation for 10,000 times
distribution.under.sharp.null <- replicate(10000, est.ate(outcomes, randomize()))

ATE = mean(as.numeric(df[df$treatment == 1]$active_in_exp)) - mean(as.numeric(df[df$treatment == 0]$acti
p_value_two_tailed <- sum(abs(distribution.under.sharp.null) > ATE)/length(distribution.under.sharp.null

qplot(distribution.under.sharp.null) +
 geom_vline(aes(xintercept = ATE,color = 'ATE')) +
  ggtitle(paste('Model: active in experiment ~ treatment, ATE = ', round(ATE,4), ', p-value = ', p_value
```




```r
lm0 = df[,lm(active_in_exp ~ treatment)] # base model
lm0$vcovCL1 = vcovCL(lm0, cluster = df[, c("address_group")])
#coeftest(lm0, vcov. = lm0$vcovCL1)

lm1 = df[,lm(active_in_exp ~ treatment + sms + treatment * sms)] # add sms
lm1$vcovCL1 = vcovCL(lm1, cluster = df[, c("address_group")])
#coeftest(lm1, vcov. = lm1$vcovCL1)

lm2 = df[,lm(active_in_exp ~ treatment + sms + phone + treatment * sms)] # add phone
```

```r
lm2$vcovCL1 = vcovCL(lm2, cluster = df[, c("address_group")])
#coeftest(lm2, vcov. = lm2$vcovCL1)

lm4 = df[,lm(active_in_exp ~ treatment + sms + phone + male)] # add gender
lm4$vcovCL1 = vcovCL(lm4, cluster = df[, c("address_group")])
#coeftest(lm4, vcov. = lm4$vcovCL1)

lm3 = df[,lm(active_in_exp ~ treatment + sms + phone + active_3m + male + treatment * sms + treatment *
lm3$vcovCL1 = vcovCL(lm3, cluster = df[, c("address_group")])
#coeftest(lm3, vcov. = lm3$vcovCL1)

stargazer(lm0, lm1, lm2, lm4, lm3,
          se = list(sqrt(diag(lm0$vcovCL1)),
                    sqrt(diag(lm1$vcovCL1)),
                    sqrt(diag(lm2$vcovCL1)),
                    sqrt(diag(lm4$vcovCL1)),
                    sqrt(diag(lm3$vcovCL1))),
          type = 'text', header = FALSE,
          omit.stat = c('F', 'ser')
)
```

```
##
## ===============================================================
##                                 Dependent variable:
##                     ------------------------------------------
##                                    active_in_exp
##                       (1)      (2)      (3)      (4)      (5)
## ---------------------------------------------------------------
## treatment            0.058    0.058    0.045    0.047    0.002
##                     (0.065)  (0.065)  (0.068)  (0.075)  (0.023)
##
## sms
##
##
## phone1                                0.162*** 0.155***  0.039
##                                       (0.048)  (0.048)  (0.024)
##
## active_3m                                                0.485***
##                                                          (0.076)
##
## treatment:sms
##
##
## treatment:active_3m                                       0.052
##                                                          (0.135)
##
## male                                            -0.130**  -0.060
##                                                 (0.053)  (0.041)
##
## Constant            0.172*** 0.172***  0.047   0.097**   0.021
##                     (0.032)  (0.032)  (0.035)  (0.045)  (0.023)
##
## ---------------------------------------------------------------
## Observations          162      162      162      162      162
```

```
## R2                    0.005    0.005    0.027    0.050    0.395
## Adjusted R2           -0.002   -0.002   0.015    0.032    0.376
## ========================================================================
## Note:                                      *p<0.1; **p<0.05; ***p<0.01
```

- ATE is the same between RI and t-test, however, t-test uses clustered standard error.
- Treatment effect is not significant.
- Having a phone number on file is more important
- Whether the user is active during the test period is highly correlated with whether he/she was active in the app three months before the experiment.
- None of the interaction terms is significant, so no significant differential treatment effect.

Hypothesis 1: differential treatment effect by group, y = active login, x = treatment

```
lm1 = df[phone == 0, lm(active_in_exp ~ treatment)] # base model
lm1$vcovCL1 = vcovCL(lm1, cluster = df[phone == 0, c("address_group")])
#coeftest(lm1, vcov. = lm1$vcovCL1)

lm2 = df[phone == 0,lm(active_in_exp ~ treatment + active_3m)] # add active history
lm2$vcovCL1 = vcovCL(lm2, cluster = df[phone == 0, c("address_group")])
#coeftest(lm2, vcov. = lm2$vcovCL1)

lm3 = df[phone == 1, lm(active_in_exp ~ treatment)] # base model
lm3$vcovCL1 = vcovCL(lm3, cluster = df[phone == 1, c("address_group")])
#coeftest(lm3, vcov. = lm3$vcovCL1)

lm4 = df[phone == 1, lm(active_in_exp ~ treatment + active_3m)] # add active history
lm4$vcovCL1 = vcovCL(lm4, cluster = df[phone == 1, c("address_group")])
#coeftest(lm4, vcov. = lm4$vcovCL1)

lm5 = df[phone == 1, lm(active_in_exp ~ treatment + active_3m + sms + treatment * active_3m + treatment
lm5$vcovCL1 = vcovCL(lm5, cluster = df[phone == 1, c("address_group")])
#coeftest(lm5, vcov. = lm5$vcovCL1)

stargazer(lm1, lm2, lm3, lm4, lm5,
         se=list(sqrt(diag(lm1$vcovCL1)),
                 sqrt(diag(lm2$vcovCL1)),
                 sqrt(diag(lm3$vcovCL1)),
                 sqrt(diag(lm4$vcovCL1)),
                 sqrt(diag(lm5$vcovCL1))),
         type = 'text',header=FALSE,
         omit.stat = c('F', 'ser'),
         column.labels = c('No Phone', 'No Phone', 'Has Phone', 'Has Phone', 'Has Phone'))
```

```
##
## ============================================================================
##                                    Dependent variable:
##                      ---------------------------------------------
##                                       active_in_exp
##                      No Phone No Phone  Has Phone Has Phone Has Phone
##                        (1)       (2)       (3)       (4)       (5)
## ----------------------------------------------------------------------------
```

```
## treatment            0.133   0.144***    0.025    -0.013     0.004
##                     (0.084)   (0.046)   (0.078)   (0.070)   (0.026)
##
## active_3m                     0.506***            0.533***  0.567***
##                                (0.188)             (0.070)   (0.092)
##
## sms
##
##
## treatment:active_3m                                          -0.049
##                                                              (0.165)
##
## treatment:sms
##
##
## Constant             0.000   -0.078*** 0.222***    0.044   0.033***
##                     (0.000)   (0.029)   (0.036)   (0.041)   (0.006)
##
## ---------------------------------------------------------------------
## Observations           28        28        134       134       134
## R2                   0.067     0.539     0.001     0.368     0.369
## Adjusted R2          0.031     0.502    -0.007     0.358     0.354
## =====================================================================
## Note:                                    *p<0.1; **p<0.05; ***p<0.01
```

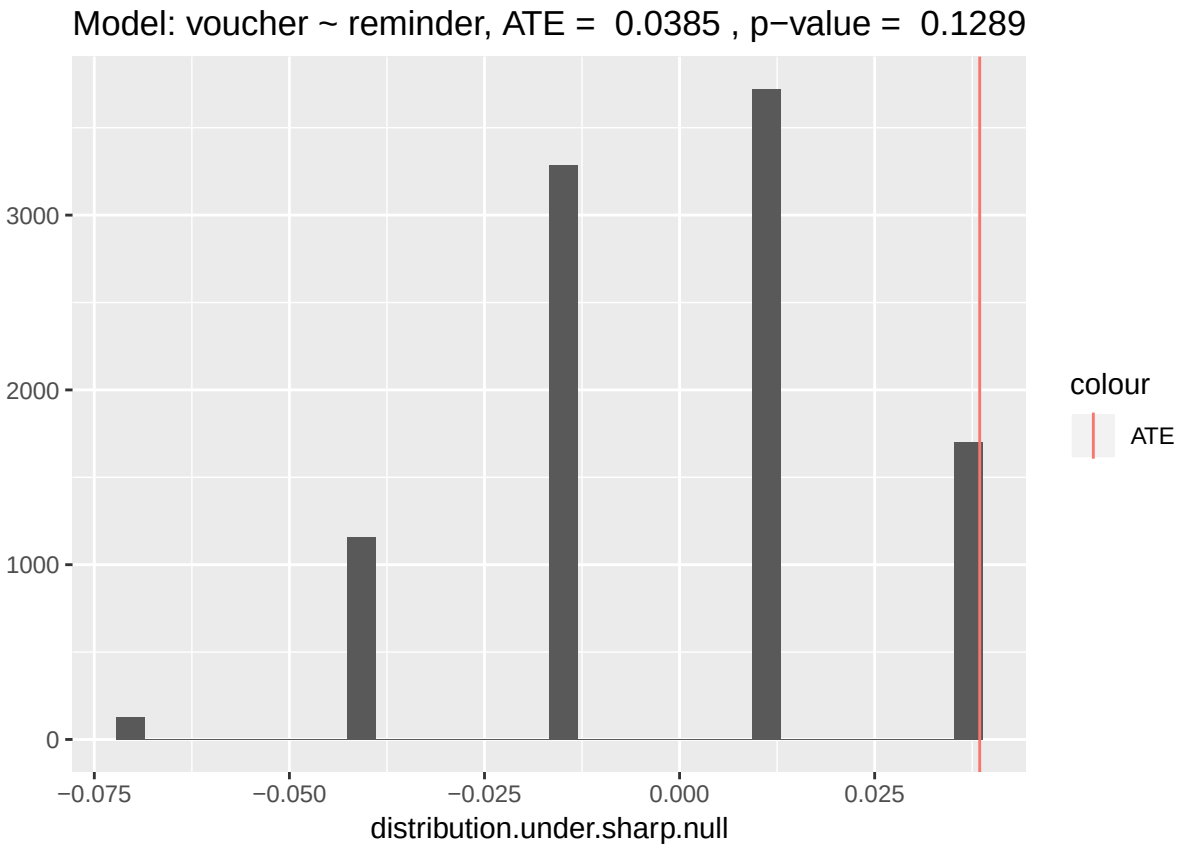
- Treatment effect is significant in the 'No Phone' on record group!
- Otherwise, only the past active login history is significant in the 'Has Phone' group.

Hypothesis 2: reminder effect on voucher uptake, y = voucher, x = sms

Random Inference: voucher ~ sms, without correcting for clustered standard error

```r
# Define potential outcomes for control and treatment, under the sharp null hypothesis, these two vecto
# Assign treatment and control potential outcomes
po.control = as.integer(df$voucher)
po.treatment = po.control

# Randomly assign units into control and treatment group
randomize = function() {sample(c(rep(0,sum(df$sms == 0)),rep(1,sum(df$sms == 1))))}
t = randomize()

# Calculate the outcome according to control/treatment assignment
outcomes = po.treatment * t + po.control*(1-t)

# Estimate the Average Treatment Effect
est.ate = function(outcome,success){mean(outcome[success==1]) - mean(outcome[success==0])}
#est.ate(outcomes, randomize())

# Run simulation for 10,000 times
distribution.under.sharp.null <- replicate(10000, est.ate(outcomes, randomize()))

ATE = mean(as.numeric(df[df$sms == 1]$voucher)) - mean(as.numeric(df[df$sms == 0]$voucher))
p_value_two_tailed <- sum(abs(distribution.under.sharp.null) > ATE)/length(distribution.under.sharp.nul
```

```
qplot(distribution.under.sharp.null) +
 geom_vline(aes(xintercept = ATE,color = 'ATE')) +
 ggtitle(paste('Model: voucher ~ reminder, ATE = ', round(ATE,4), ', p-value = ', p_value_two_tailed)) +
 ylab('')
```




```
table(df$sms, df$voucher)
```

```
##
##       0   1
##   0  58   0
##   1 100   4
```

```
lm2 = df[phone == 1, lm(voucher ~ sms)] # sms
lm2$vcovCL1 = vcovCL(lm2, cluster = df[phone == 1, c("address_group")])
#coeftest(lm2, vcov. = lm2$vcovCL1)

lm1 = df[phone == 1, lm(voucher ~ sms + drug_count)] # add drug count
lm1$vcovCL1 = vcovCL(lm1, cluster = df[phone == 1, c("address_group")])
#coeftest(lm1, vcov. = lm1$vcovCL1)

lm3 = df[phone == 1, lm(voucher ~ sms + drug_count + male)] # add male
lm3$vcovCL1 = vcovCL(lm3, cluster = df[phone == 1, c("address_group")])
#coeftest(lm3, vcov. = lm3$vcovCL1)

lm4 = df[phone == 1, lm(voucher ~ sms + drug_count + male + active_3m)] # add active_3m
lm4$vcovCL1 = vcovCL(lm4, cluster = df[phone == 1, c("address_group")])
```

```
#coeftest(lm4, vcov. = lm4$vcovCL1)

stargazer(lm2, lm1, lm3, lm4,
          type = 'text', header = FALSE,
          se=list(sqrt(diag(lm2$vcovCL1)),
                  sqrt(diag(lm1$vcovCL1)),
                  sqrt(diag(lm3$vcovCL1)),
                  sqrt(diag(lm4$vcovCL1))),
          omit.stat = c('F', 'ser')
          )
```

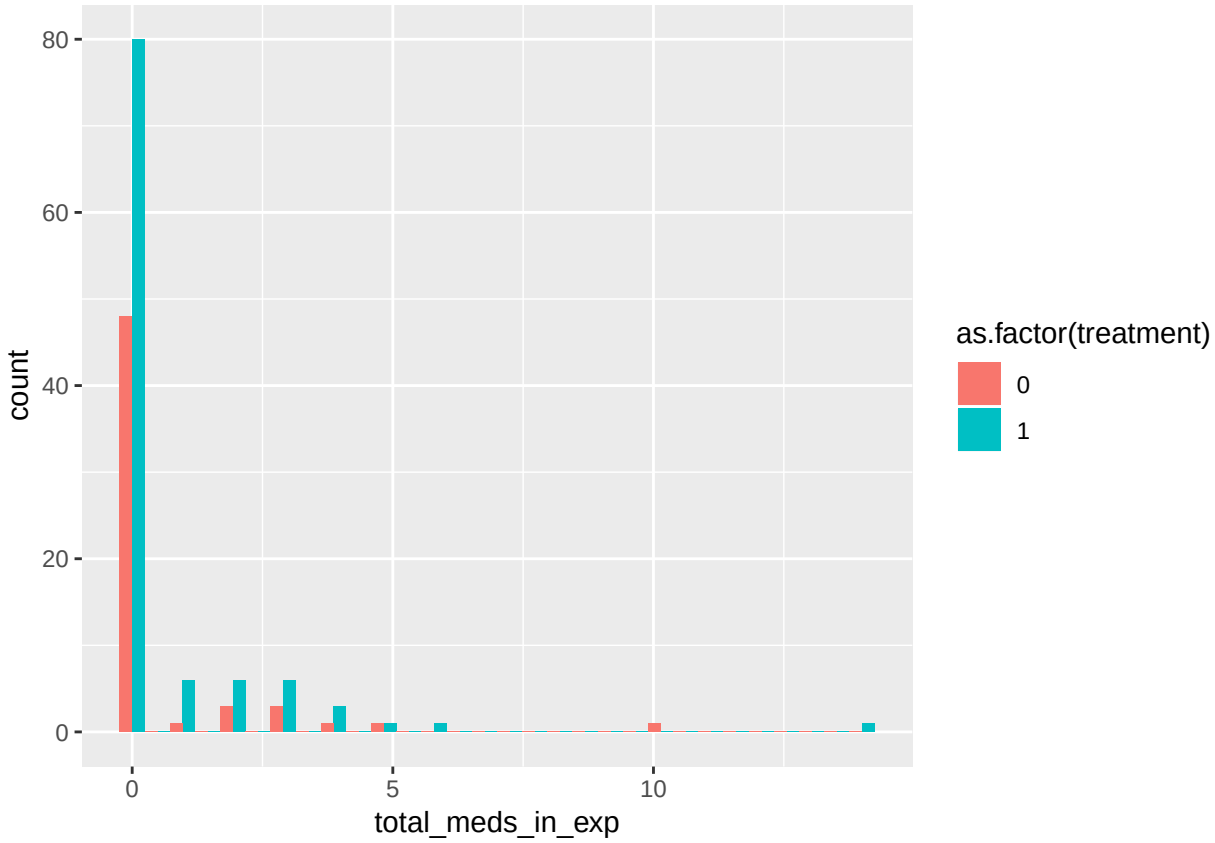
```
##
## ================================================
##                         Dependent variable:
##              ----------------------------------
##                               voucher
##                (1)       (2)       (3)      (4)
## ------------------------------------------------
## sms          0.034*** 0.035***  0.035***  0.030***
##              (0.011)   (0.012)   (0.012)  (0.011)
##
## drug_count            0.028***  0.025***   0.003*
##                        (0.009)   (0.008)  (0.002)
##
## male                            -0.034*** -0.028**
##                                  (0.012)  (0.011)
##
## active_3m                                 0.053***
##                                           (0.019)
##
## Constant      0.000   -0.026*** -0.012*** -0.011**
##              (0.000)   (0.009)   (0.004)  (0.005)
##
## ------------------------------------------------
## Observations   134       134       134      134
## R2            0.012     0.014     0.026    0.054
## Adjusted R2   0.004    -0.001     0.004    0.025
## ================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

- impact of sms is not significant
- but the impact of drug count and male is significant

Hypothesis 3: treatment effect on scripts ordered, y = scripts ordered, x = treatment

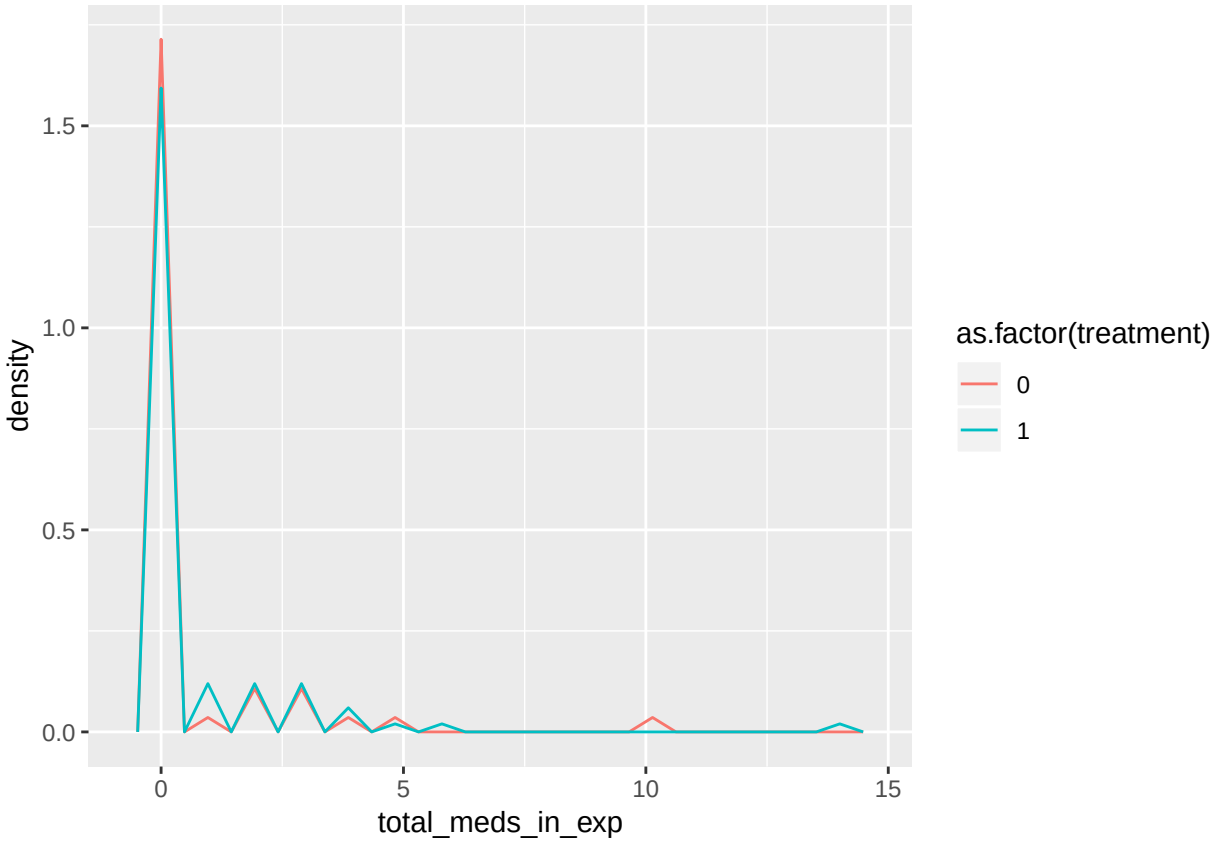
```
df$total_meds_in_exp = as.integer(ifelse(is.na(df$total_meds_in_exp),0,df$total_meds_in_exp)) # need to

# histogram
ggplot(data=df, aes(x = total_meds_in_exp, fill = as.factor(treatment))) +
  geom_histogram(position="dodge")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# density plot
ggplot(df) + geom_freqpoly(aes(x = total_meds_in_exp,
     y = ..density.., colour = as.factor(treatment)))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

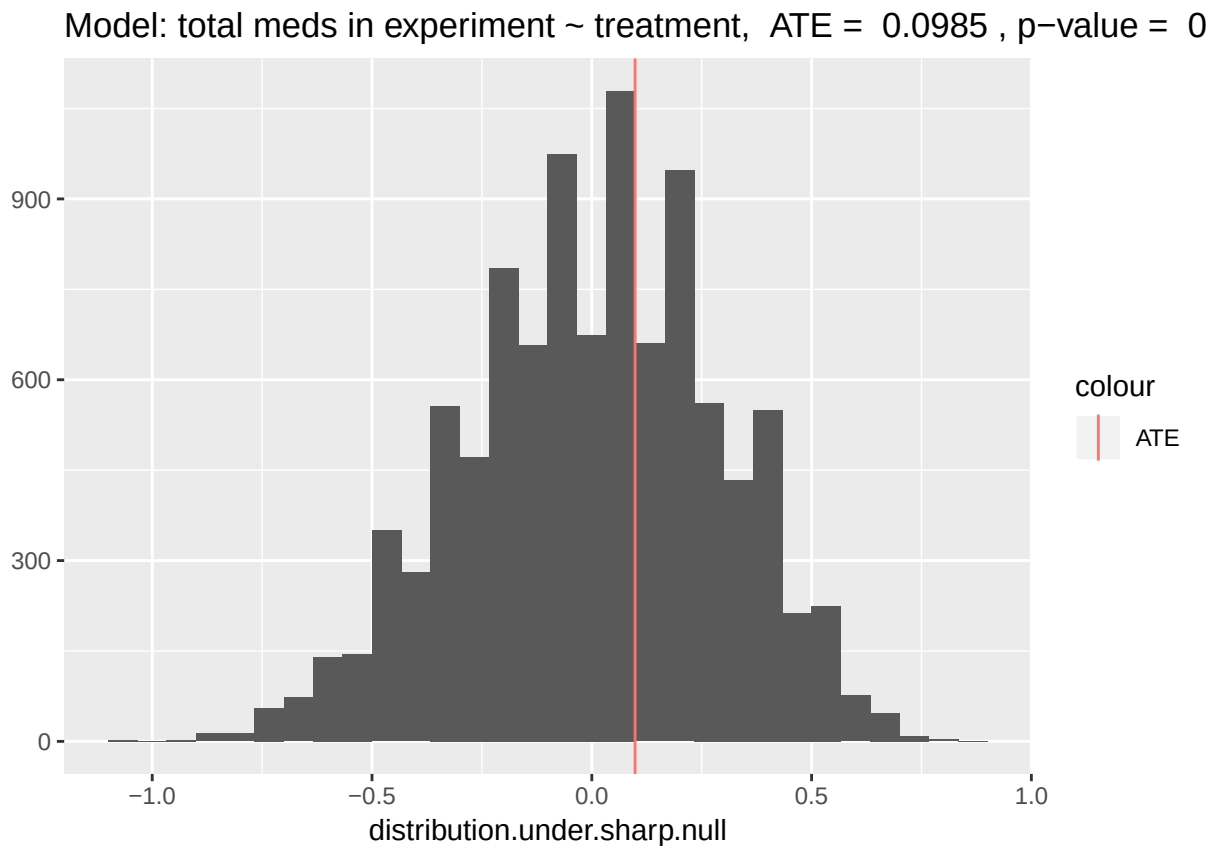


- Highly skeweed distribution, normality assumption probably won't hold given the sample size

Random Inference: total meds ordered in exp ~ treatment, without correcting for clustered standard error

```
# Define potential outcomes for control and treatment, under the sharp null hypothesis, these two vecto
df = na.fill(df,0) %>% data.table()

# Assign treatment and control potential outcomes
po.control = as.integer(df$total_meds_in_exp)
po.treatment = po.control

# Randomly assign units into control and treatment group
randomize = function() {sample(c(rep(0,sum(df$treatment == 0)),rep(1,sum(df$treatment == 1))))}
t = randomize()

# Calculate the outcome according to control/treatment assignment
outcomes = po.treatment * t + po.control*(1-t)

# Estimate the Average Treatment Effect
est.ate = function(outcome,success){mean(outcome[success==1]) - mean(outcome[success==0])}
#est.ate(outcomes, randomize())

# Run simulation for 10,000 times
distribution.under.sharp.null <- replicate(10000, est.ate(outcomes, randomize()))
```

```
ATE = mean(as.numeric(df[df$treatment == 1]$total_meds_in_exp)) - mean(as.numeric(df[df$treatment == 0]$
p_value_two_tailed <- sum(abs(distribution.under.sharp.null) > ATE)/length(distribution.under.sharp.null

qplot(distribution.under.sharp.null) +
 geom_vline(aes(xintercept = ATE,color = 'ATE')) +
  ggtitle(paste('Model: total meds in experiment ~ treatment, ','ATE = ', round(ATE,4), ', p-value = ',
```



- ATE is the same between RI and t-test, however, t-test uses clustered standard error.

```
table(df$treatment, df$total_meds_in_exp)
```

```
##
##      0  1  2  3  4  5  6 10 14
##   0 48  1  3  3  1  1  0  1  0
##   1 80  6  6  6  3  1  1  0  1
```

```
df$total_meds_in_exp = as.integer(df$total_meds_in_exp)
df$total_meds_3m = as.integer(df$total_meds_3m)

## No Phone Group
lm0 = df[phone == 0, lm(total_meds_in_exp ~ treatment)] # no phone
lm0$vcovCL1 = vcovCL(lm0, cluster = df[phone == 0, c("address_group")])
#coeftest(lm0, vcov. = lm0$vcovCL1)

lm1 = df[phone == 0, lm(total_meds_in_exp ~ treatment + total_meds_3m + treatment * total_meds_3m)] # a
lm1$vcovCL1 = vcovCL(lm1, cluster = df[phone == 0, c("address_group")])
#coeftest(lm1, vcov. = lm1$vcovCL1)
```

```r
## Has Phone Group
lm6 = df[phone == 1, lm(total_meds_in_exp ~ treatment)] # has phone
lm6$vcovCL1 = vcovCL(lm6, cluster = df[phone == 1, c("address_group")])
#coeftest(lm6, vcov. = lm6$vcovCL1)

lm2 = df[phone == 1, lm(total_meds_in_exp ~ treatment + sms + treatment * sms)] # add sms
lm2$vcovCL1 = vcovCL(lm2, cluster = df[phone == 1, c("address_group")])
#coeftest(lm2, vcov. = lm2$vcovCL1)

lm4 = df[phone == 1, lm(total_meds_in_exp ~ treatment + sms + total_meds_3m + treatment * sms + treatmer
lm4$vcovCL1 = vcovCL(lm4, cluster = df[phone == 1, c("address_group")])
#coeftest(lm4, vcov. = lm4$vcovCL1)

stargazer(lm0, lm1, lm6, lm2, lm4,
          se = list(sqrt(diag(lm0$vcovCL1)),
                    sqrt(diag(lm1$vcovCL1)),
                    sqrt(diag(lm6$vcovCL1)),
                    sqrt(diag(lm2$vcovCL1)),
                    sqrt(diag(lm4$vcovCL1))),
          type = 'text', header = FALSE,
          omit.stat = c('F', 'ser'),
          column.labels = c('No Phone', 'No Phone', 'Has Phone', 'Has Phone', 'Has Phone')
          )
```

```
##
## ================================================================================
##                                        Dependent variable:
##                            ---------------------------------------------------
##                                          total_meds_in_exp
##                            No Phone No Phone  Has Phone Has Phone Has Phone
##                              (1)       (2)       (3)       (4)       (5)
## --------------------------------------------------------------------------------
## treatment1                 0.400***  0.047**   -0.025    -0.025     0.132
##                            (0.147)   (0.021)   (0.299)   (0.299)   (0.239)
##
## total_meds_3m                        -0.000                       0.351***
##                                     (0.00000)                      (0.100)
##
## treatment1:total_meds_3m             0.265**                      -0.037
##                                      (0.124)                      (0.137)
##
## sms1
##
##
## treatment1:sms1
##
##
## Constant                    0.000     0.000    0.778**   0.778**    0.131
##                            (0.000)             (0.314)   (0.314)   (0.123)
##
## --------------------------------------------------------------------------------
## Observations                  28        28        134       134       134
## R2                          0.060     0.738    0.00004   0.00004    0.449
```

```
## Adjusted R2                0.023     0.706    -0.008    -0.008     0.436
## ============================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```

- model (1): for the 'No Phone' group, being in the treatment group is predictive of total meds ordered during the experiment
- model (2): power user effect - total meds ordered in the past 3 months is highly predictive of total meds ordered during the experiment
- dont know how to interpret model (3) and (4)... maybe we should take them out
- model (5) receiving a sms reminder has a negative impact on total meds ordered?