Final Study Data Analysis

April Kim, Jennifer Podracky, Saurav Datta

```r
library(ggplot2)
library(data.table)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(pwr)
library(lsr)
library(cobalt)
library(stringr)
library(AER)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
## Loading required package: sandwich
```

```
## Loading required package: survival
```

Read in data and reformat

```r
assigned_treatment_seq <- data.frame(seq_id = c(1,2,3,4,5,6),
                                     day1 = c(0,0,1,1,2,2),
                                     day2 = c(1,2,0,2,0,1),
                                     day3 = c(2,1,2,0,1,0))
d2 <- fread("241 Participant List - Final Study Results - 20181215.csv", na.strings=c("","NA"))
d2[UserId == 65,]$Q10 <- "In person"
d2[UserId == 13,]$Q6 <- "Through digital means"
# stringsAsFactors = F)
names(d2) <- str_replace_all(names(d2), c(" " = "." , "," = "" ))

# Not applicable = 0
# Through digital means = 1
# In person = 2
# Both in person and through digital means = 3
```

```r
d2 <- d2[, .(userId = UserId,
            treatment_seq = as.integer(Treatment.Seq),
            day1_treatment = as.integer(as.character(factor(Q6, levels = c('Not applicable', 'In person
                                                            'Through digital means'),
                                               labels = c(0, 2, 1)))),
            day2_treatment = as.integer(as.character(factor(Q10, levels = c('Not applicable', 'In perso
                                                            'Through digital means',
                                                            'Both in person and throug
                                               labels = c(0, 2, 1, 3)))),
            day3_treatment = as.integer(as.character(factor(Q14, levels = c('Not applicable', 'In perso
                                                            'Through digital means',
                                                            'Both in person and throug
                                               labels = c(0, 2, 1, 3)))),
            day1_steps = as.numeric(gsub("\\,", "", Q7)),
            day2_steps = as.numeric(gsub("\\,", "", Q11)),
            day3_steps = as.numeric(gsub("\\,", "", Q15)),
            age_range = as.integer(as.character(factor(Age, levels = c('18 - 24',
                                                            "25 - 34",
                                                            "35 - 44",
                                                            "45 - 54",
                                                            "55 - 64",
                                                            "65+"),
                                               labels = c(0, 1, 2, 3, 4, 5)))),
            # gender = factor(Gender),
            gender = as.integer(as.character(factor(Gender, levels = c('Male', 'Female', 'Gender non-co
                                               labels = c(0, 1, 2)))),
            lives_with_others = as.integer(as.character(factor(Living.Situation, levels = c('Alone', '
                                               labels = c(0, 1)))),
            # know_us = factor(Q17),
            know_us = as.integer(as.character(factor(Q17, levels = c('No', 'Yes'),
                                               labels = c(0, 1)))),
            location_lat = as.double(LocationLatitude),
            location_long = as.double(LocationLongitude)
)]
```

```
## Warning in eval(jsub, SDenv, parent.frame()): NAs introduced by coercion
```

```
## Warning in eval(jsub, SDenv, parent.frame()): NAs introduced by coercion
```

```
## Warning in eval(jsub, SDenv, parent.frame()): NAs introduced by coercion
```

```r
d2$gender[is.na(d2$gender)] <- 2
d2$age_range[is.na(d2$age_range)] <- 6
d2$lives_with_others[is.na(d2$lives_with_others)] <- 2
d2$know_us[is.na(d2$know_us)] <- 2

head(d2, 5)
```

```
##     userId treatment_seq day1_treatment day2_treatment day3_treatment
## 1:      82             6              0              1              0
## 2:      57             3              1              0              2
## 3:      89             4             NA             NA             NA
## 4:      69             3              1              0              2
```

```
## 5:        85                  3              1              0              2
##     day1_steps day2_steps day3_steps age_range gender lives_with_others
## 1:          NA       5040       3788         1      0                 1
## 2:       21290      13959      13717         0      0                 1
## 3:          NA         NA         NA         1      0                 1
## 4:        6343       3247      10198         1      0                 1
## 5:       13624       5406       7851         1      1                 1
##     know_us location_lat location_long
## 1:        1     41.89250      -87.7895
## 2:        1     37.75101      -97.8220
## 3:        1     37.97240     -122.3369
## 4:        1     40.37070      -74.0084
## 5:        1     42.41730      -71.1087
```

```r
#Covariate Balance Check
bal.tab(treatment_seq ~ gender + age_range + lives_with_others + know_us + location_lat + location_long
        data = d2)
```

```
## Balance Measures
##                          Type Corr.Un
## gender            Contin. -0.0625
## age_range         Contin. -0.0099
## lives_with_others Contin. -0.0105
## know_us           Contin.  0.0214
## location_lat      Contin.  0.0157
## location_long     Contin. -0.0480
##
## Sample sizes
##     Total
## All    75
```

```r
cov_check <- lm(treatment_seq ~ gender + age_range + lives_with_others + know_us + location_lat + locati
                data = d2)
summary(cov_check)
```

```
##
## Call:
## lm(formula = treatment_seq ~ gender + age_range + lives_with_others +
##     know_us + location_lat + location_long, data = d2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7912 -1.4863  0.1883  1.4598  2.6363
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.1350717  4.1690323   0.272    0.786
## gender            -0.1639576  0.3999806  -0.410    0.683
## age_range         -0.0003737  0.1750913  -0.002    0.998
## lives_with_others -0.1171429  0.7099934  -0.165    0.869
## know_us            0.0212714  0.3840969   0.055    0.956
## location_lat       0.0465410  0.0841290   0.553    0.582
## location_long     -0.0082195  0.0124256  -0.661    0.511
```

3

```
## 
## Residual standard error: 1.784 on 68 degrees of freedom
## Multiple R-squared:  0.01086,    Adjusted R-squared:  -0.07641
## F-statistic: 0.1245 on 6 and 68 DF,  p-value: 0.993
```

Checking for ordering/priming effect

Is previous day's treatment highly predictive of how many steps are taken today?

```r
# n = 75
df <- d2

# remove subjects/rows who were non-compliant
# n = 24
df <- df[rowSums(is.na(df[,c(3:8)])) != ncol(df[,c(3:8)]), ]

head(df, 5)
```

```
##    userId treatment_seq day1_treatment day2_treatment day3_treatment
## 1:     82             6              0              1              0
## 2:     57             3              1              0              2
## 3:     69             3              1              0              2
## 4:     85             3              1              0              2
## 5:     66             4              1              2              0
##    day1_steps day2_steps day3_steps age_range gender lives_with_others
## 1:         NA       5040       3788         1      0                 1
## 2:      21290      13959      13717         0      0                 1
## 3:       6343       3247      10198         1      0                 1
## 4:      13624       5406       7851         1      1                 1
## 5:       7016       1211       5717         0      0                 1
##    know_us location_lat location_long
## 1:       1     41.89250      -87.7895
## 2:       1     37.75101      -97.8220
## 3:       1     40.37070      -74.0084
## 4:       1     42.41730      -71.1087
## 5:       1     42.35760      -71.0514
```

```r
# day 3 steps using day 1 and 2 treatment
m1 <- lm(day3_steps ~ day1_treatment + day2_treatment, df)
summary(m1)
```

```
## 
## Call:
## lm(formula = day3_steps ~ day1_treatment + day2_treatment, data = df)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7119.9 -2316.1  -462.4  1377.2  8580.4
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      7505.7       921.5   8.145 2.12e-10 ***
## day1_treatment   -385.7       672.7  -0.573    0.569
## day2_treatment   -365.5       600.8  -0.608    0.546
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3440 on 45 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.0142, Adjusted R-squared:  -0.02961
## F-statistic: 0.3241 on 2 and 45 DF,  p-value: 0.7248
```

```r
# ATE (standard error)
print(paste0("Estimated effect of day1 treatment: ", signif(m1$coefficients[2], 3),
" (", signif(coef(summary(m1))[2,2], 3), ")"))
```

```
## [1] "Estimated effect of day1 treatment: -386 (673)"
```

```r
print(paste0("Estimated effect of day2 treatment: ", signif(m1$coefficients[3], 3),
" (", signif(coef(summary(m1))[3,2], 3), ")"))
```

```
## [1] "Estimated effect of day2 treatment: -366 (601)"
```

```r
# include days1,2 steps as covariates to understand
# subjects' step counts hange as a function of
# treatment against waht they would typically do
m2 <- lm(day3_steps ~ day1_treatment + day2_treatment + day1_steps + day2_steps, df)
summary(m2)
```

```
##
## Call:
## lm(formula = day3_steps ~ day1_treatment + day2_treatment + day1_steps +
##     day2_steps, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8183.6 -1375.5    61.2  1536.6  5057.0
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2602.4071  1128.7513   2.306  0.02614 *
## day1_treatment -484.8396   517.7869  -0.936  0.35444
## day2_treatment -296.7856   460.9784  -0.644  0.52319
## day1_steps        0.2442     0.1230   1.985  0.05373 .
## day2_steps        0.4542     0.1341   3.386  0.00155 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2623 on 42 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.455,  Adjusted R-squared:  0.4031
## F-statistic: 8.766 on 4 and 42 DF,  p-value: 3.076e-05
```

```
print(paste0("Estimated effect of day1 treatment: ", signif(m2$coefficients[2], 3),
             " (", signif(coef(summary(m2))[2,2], 3), ")"))
```

[1] "Estimated effect of day1 treatment: -485 (518)"

```
print(paste0("Estimated effect of day2 treatment: ", signif(m2$coefficients[3], 3),
             " (", signif(coef(summary(m2))[3,2], 3), ")"))
```

[1] "Estimated effect of day2 treatment: -297 (461)"

We do not see that the previous days' treatment assignments to predict the last day's step count is highgly predicitive and significant, which is super for us!

Condense treatment sequence to 1 treatment

```
df1 <- df[,-c(4,5,7,8)]
df2 <- df[,-c(3,5,6,8)]
df3 <- df[,-c(3,4,6,7)]
names(df1)[names(df1) == "day1_treatment"] = "treatment"
names(df1)[names(df1) == "day1_steps"] = "steps"
names(df2)[names(df2) == "day2_treatment"] = "treatment"
names(df2)[names(df2) == "day2_steps"] = "steps"
names(df3)[names(df3) == "day3_treatment"] = "treatment"
names(df3)[names(df3) == "day3_steps"] = "steps"
d <- rbind(df1, df2, df3)
# combine digital and in person treatment as one
d$treatment2 <- ifelse(d$treatment == 0, 0, 1)
d$outcome <- ifelse(d$steps > 5000, 1, 0)

head(d, 5)
```

```
##    userId treatment_seq treatment steps age_range gender lives_with_others
## 1:     82             6         0    NA         1      0                 1
## 2:     57             3         1 21290         0      0                 1
## 3:     69             3         1  6343         1      0                 1
## 4:     85             3         1 13624         1      1                 1
## 5:     66             4         1  7016         0      0                 1
##    know_us location_lat location_long treatment2 outcome
## 1:       1     41.89250      -87.7895          0      NA
## 2:       1     37.75101      -97.8220          1       1
## 3:       1     40.37070      -74.0084          1       1
## 4:       1     42.41730      -71.1087          1       1
## 5:       1     42.35760      -71.0514          1       1
```

Check covariates again after transforming and create plots (Maybe we can just check covariates here?)
```

```
#Covariate Balance Check
bal.tab(treatment2 ~ gender + age_range + lives_with_others + know_us + location_lat + location_long,
        data = d)
```

```
## Note: estimand and s.d.denom not specified; assuming ATE and pooled.
```

```
## Balance Measures
##                       Type Diff.Un
## gender             Contin. -0.0582
## age_range          Contin.  0.2103
## lives_with_others  Contin. -0.1690
## know_us             Binary -0.0093
## location_lat       Contin.  0.0444
## location_long      Contin.  0.0661
##
## Sample sizes
##         Control Treated
## All          71      82
```

```
cov_check1 <- lm(treatment_seq ~ gender + age_range + lives_with_others + know_us + location_lat + loca
                 data = d)
summary(cov_check1)
```

```
##
## Call:
## lm(formula = treatment_seq ~ gender + age_range + lives_with_others +
##     know_us + location_lat + location_long, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.45660 -1.45557 -0.07703  1.47990  2.76916
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.949777   3.094677  -0.307    0.759
## gender             0.222693   0.275319   0.809    0.420
## age_range         -0.037168   0.124112  -0.299    0.765
## lives_with_others  0.220037   0.511785   0.430    0.668
## know_us            0.316655   0.508372   0.623    0.534
## location_lat       0.074428   0.056090   1.327    0.187
## location_long     -0.007474   0.009532  -0.784    0.434
##
## Residual standard error: 1.701 on 146 degrees of freedom
## Multiple R-squared:  0.02282,    Adjusted R-squared:  -0.01733
## F-statistic: 0.5683 on 6 and 146 DF,  p-value: 0.755
```

```
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```r
d.gender <- d[, c("gender", "treatment2")]
p_gender <- ggplot(d.gender, aes(x=gender, fill = factor(treatment2))) +
  geom_bar(stat="count", position=position_dodge()) +
  theme_minimal() + theme(legend.position="top") +
  xlab("") + ylab("") + ggtitle("Gender") +
  guides(fill = guide_legend(title = "Assignment")) +
  scale_fill_discrete(labels = c("Control", "Treatment")) +
  scale_x_continuous(breaks = c(0, 1, 2),
                     labels = c('Male', 'Female', 'Gender non-conforming'))

d.age <- d[, c("age_range", "treatment2")]
p_age <- ggplot(d.age, aes(x=age_range, fill = factor(treatment2))) +
  geom_bar(stat="count", position=position_dodge()) +
  theme_minimal() + theme(legend.position="none") +
  xlab("") + ylab("") + ggtitle("Age range") +
  # guides(fill = guide_legend(title = "Assignment")) +
  # scale_fill_discrete(labels = c("Control", "Treatment")) +
  scale_x_continuous(breaks = c(0, 1, 2, 3, 4, 5),
                     labels = c('18 - 24',
                                "25 - 34",
                                "35 - 44",
                                "45 - 54",
                                "55 - 64",
                                "65+"))

d.others <- d[, c("lives_with_others", "treatment2")]
p_others <- ggplot(d.others, aes(x=lives_with_others, fill = factor(treatment2))) +
  geom_bar(stat="count", position=position_dodge()) +
  theme_minimal() + theme(legend.position="none") +
  xlab("") + ylab("") + ggtitle("Lives with others") +
  # guides(fill = guide_legend(title = "Assignment")) +
  # scale_fill_discrete(labels = c("Control", "Treatment")) +
  scale_x_continuous(breaks = c(0, 1),
                     labels = c('Alone', 'With others'))

d.know_us <- d[, c("know_us", "treatment2")]
p_know_us <- ggplot(d.know_us, aes(x=know_us, fill = factor(treatment2))) +
  geom_bar(stat="count", position=position_dodge()) +
  theme_minimal() + theme(legend.position="none") +
  xlab("") + ylab("") + ggtitle("Know us") +
  # guides(fill = guide_legend(title = "Assignment")) +
  # scale_fill_discrete(labels = c("Control", "Treatment")) +
  scale_x_continuous(breaks = c(0, 1),
                     labels = c('No', 'Yes'))

grid.arrange(p_gender, p_age, p_others, p_know_us,
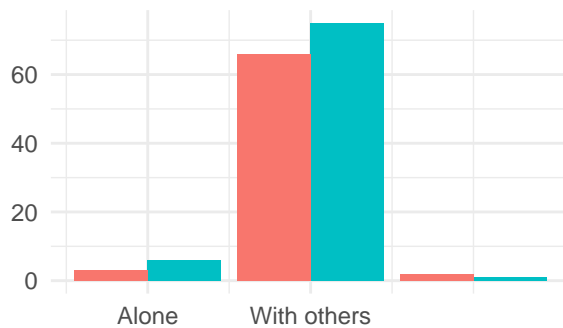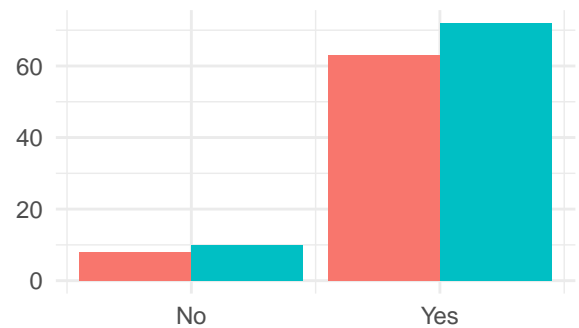             ncol = 2)
```

Gender
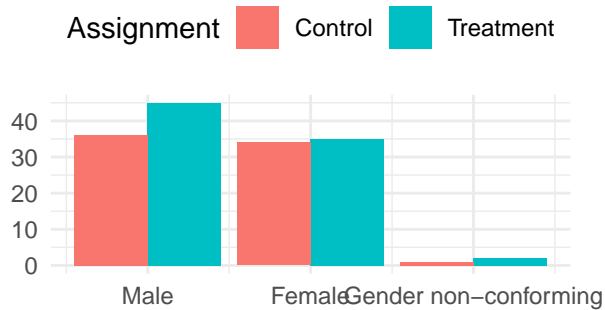
Age range
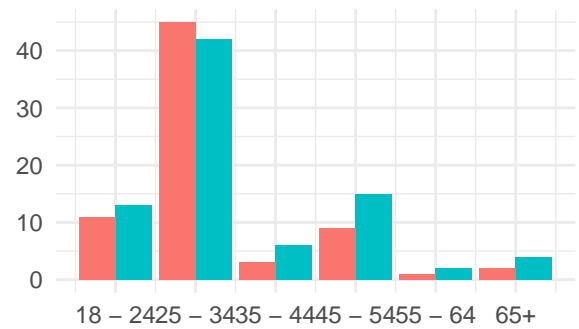
Lives with others

Know us

T-test and power calculations

```
###  Control vs treatment (digital+in person)
t.test(d[treatment2 == 0]$outcome, d[treatment2 == 1]$outcome)
```

```
##
##  Welch Two Sample t-test
##
## data:  d[treatment2 == 0]$outcome and d[treatment2 == 1]$outcome
## t = -0.91261, df = 131.56, p-value = 0.3631
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.21320328  0.07858789
## sample estimates:
## mean of x mean of y
## 0.7076923 0.7750000
```

```
# since we fail to reject the null hypothesis,
# let's calculate number of subjects needed for 80% power
effect_size <- cohensD(d[treatment2 == 0]$outcome, d[treatment2 == 1]$outcome)
pwr.t.test(power = 0.8, d = effect_size, sig.level = 0.05, type = "two.sample")
```

```
##
```

```
##      Two-sample t test power calculation
##
##               n = 664.7977
##               d = 0.1537758
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
# if we were to look at digital and in person separately
# digital
t.test(d[treatment == 0]$outcome, d[treatment == 1]$outcome)
```

```
##
##  Welch Two Sample t-test
##
## data:  d[treatment == 0]$outcome and d[treatment == 1]$outcome
## t = -1.5548, df = 87.831, p-value = 0.1236
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.29649330  0.03620224
## sample estimates:
## mean of x mean of y
## 0.7076923 0.8378378
```

```r
effect_size_digital <- cohensD(d[treatment == 0]$outcome, d[treatment == 1]$outcome)
pwr.t.test(power = 0.8, d = effect_size_digital, sig.level = 0.05, type = "two.sample")
```

```
##
##      Two-sample t test power calculation
##
##               n = 172.172
##               d = 0.3028018
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
# in person
t.test(d[treatment == 0]$outcome, d[treatment == 2]$outcome)
```

```
##
##  Welch Two Sample t-test
##
## data:  d[treatment == 0]$outcome and d[treatment == 2]$outcome
## t = -0.1478, df = 90.702, p-value = 0.8828
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1911623  0.1646864
## sample estimates:
## mean of x mean of y
## 0.7076923 0.7209302
```

```r
effect_size_person <- cohensD(d[treatment == 0]$outcome, d[treatment == 2]$outcome)
pwr.t.test(power = 0.8, d = effect_size_person, sig.level = 0.05, type = "two.sample")
```

```
## 
##      Two-sample t test power calculation 
## 
##              n = 18674.66
##              d = 0.02899366
##      sig.level = 0.05
##          power = 0.8
##    alternative = two.sided
## 
## NOTE: n is number in *each* group
```

data manipulation for paired t test!

```r
'%!in%' <- function(x,y)!('%in%'(x,y))
# limit to those who complied and
# get data frame in workable format
d_complied <- rbindlist(list(subset(df, treatment_seq == 1 & df$day1_treatment == assigned_treatment_se
                             & df$day2_treatment == assigned_treatment_seq[1,2]
                             & df$day3_treatment == assigned_treatment_seq[1,3]),
                        subset(df, treatment_seq == 2 & df$day1_treatment == assigned_treatment_se
                             & df$day2_treatment == assigned_treatment_seq[2,2]
                             & df$day3_treatment == assigned_treatment_seq[2,3]),
                        subset(df, treatment_seq == 3 & df$day1_treatment == assigned_treatment_se
                             & df$day2_treatment == assigned_treatment_seq[3,2]
                             & df$day3_treatment == assigned_treatment_seq[3,3]),
                        subset(df, treatment_seq == 4 & df$day1_treatment == assigned_treatment_se
                             & df$day2_treatment == assigned_treatment_seq[4,2]
                             & df$day3_treatment == assigned_treatment_seq[4,3]),
                        subset(df, treatment_seq == 5 & df$day1_treatment == assigned_treatment_se
                             & df$day2_treatment == assigned_treatment_seq[5,2]
                             & df$day3_treatment == assigned_treatment_seq[5,3]),
                        subset(df, treatment_seq == 6 & df$day1_treatment == assigned_treatment_se
                             & df$day2_treatment == assigned_treatment_seq[6,2]
                             & df$day3_treatment == assigned_treatment_seq[6,3])
))
d_not_complied <- subset(df, userId %!in% d_complied$userId)
d_not_complied_ok <- subset(d_not_complied, d_not_complied$day1_treatment != d_not_complied$day2_treatme
                           d_not_complied$day1_treatment != d_not_complied$day3_treatment &
                           d_not_complied$day2_treatment != d_not_complied$day3_treatment)

d_usable <- rbind(d_complied, d_not_complied_ok)
d <- rbindlist(list(d_usable[1:nrow(d_usable),c("day1_treatment","day1_steps")],
                   d_usable[1:nrow(d_usable),c("day2_treatment","day2_steps")],
                   d_usable[1:nrow(d_usable),c("day3_treatment","day3_steps")]))
names(d) <- c("treatment","steps")
# interpolate using median steps
d$steps[is.na(d$steps)] <- median(d[which(!is.na(d$steps))]$steps)
d$outcome <- ifelse(d$steps > 5000, 1, 0)
```

```r
# CACE
ITT_lm <- lm(outcome ~ treatment, data = d)
summary(ITT_lm)$coefficients[2]
```

```
## [1] -0.04411765
```

```r
# COMMENTED LINES BELOW HAS TO BE FIXED!
# SKIP DOWN TO T TEST IN LINE 290
# combine digital and in person as one treatment as
# "Both in person and through digital means"
# d1 <- d_usable
# d1a <- subset(d1, d1$day1_treatment == 1 & d1$day2_treatment == 2)
# d1a[, treatment_total_steps := day1_steps + day2_steps]
# d1b <- subset(d1, d1$day2_treatment == 1 & d1$day3_treatment == 2)
# d1b[, treatment_total_steps := day2_steps + day3_steps]
# d1c <- subset(d1, d1$day1_treatment == 1 & d1$day3_treatment == 2)
# d1c[, treatment_total_steps := day1_steps + day2_steps]
# d1d <- subset(d1, d1$day2_treatment == 2 & d1$day3_treatment == 1)
# d1d[, treatment_total_steps := day2_steps + day3_steps]
# d1e <- subset(d1, d1$day1_treatment == 2 & d1$day3_treatment == 1)
# d1e[, treatment_total_steps := day1_steps + day2_steps]
# d1f <- subset(d1, d1$day1_treatment == 2 & d1$day2_treatment == 1)
# d1f[, treatment_total_steps := day2_steps + day3_steps]
#
# d1_comb <- rbind(d1a, d1b, d1c, d1d, d1e, d1f)
#
# d1 <- rbindlist(list(d1_comb[1:nrow(d1_comb),c("day1_treatment","day1_steps")],
#                      d1_comb[1:nrow(d1_comb),c("day2_treatment","day2_steps")],
#                      d1_comb[1:nrow(d1_comb),c("day3_treatment","day3_steps")]))
# names(d1) <- c("treatment","steps")
# d1[d1$treatment == 1 | d1$treatment == 2, ]$treatment <- 3
# # linear interpolation
# # d1 <- na.approx(d1, maxgap=5)
# d1 <- data.table(d1)
# d1$steps[is.na(d1$steps)] <- median(d1[which(!is.na(d1$steps))]$steps)
# d1$outcome <- ifelse(d1$steps > 5000, 1, 0)

t.test(d[treatment == 0]$outcome, d[treatment == 1]$outcome, paired = T)
```

```
##
##  Paired t-test
##
## data:  d[treatment == 0]$outcome and d[treatment == 1]$outcome
## t = 0, df = 33, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1920722  0.1920722
## sample estimates:
## mean of the differences
##                       0
```

```r
cohensD(d[treatment == 0]$outcome, d[treatment == 1]$outcome, method = "paired")
```

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

Control vs in person
```r
t.test(d[treatment == 0]$outcome, d[treatment == 2]$outcome, paired = T)
```

```
##
##  Paired t-test
##
## data:  d[treatment == 0]$outcome and d[treatment == 2]$outcome
## t = 0.90205, df = 33, p-value = 0.3736
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1107732  0.2872437
## sample estimates:
## mean of the differences
##             0.08823529
```

```r
cohensD(d[treatment == 0]$outcome, d[treatment == 2]$outcome, method = "paired")
```

```
## [1] 0.1547007
```

Control vs in person + digital
```r
# t.test(d1[treatment == 0]$outcome, d1[treatment == 3]$outcome)
```