W241 - Final Project Data Analysis

Ramya Balasubramaniam, Jeffrey Hsu and Roiana Reid

02 May 2017

1. Load Data

1.1 Load Sales Data

The datasets starting with SO represents sales data on sales order level. The datasets starting with SOI represents sales data on sales order item level. These are datasets provided by the online shop owner. We start from loading the dataset, subsetting the variables needed, renaming the columns and merging them into each SO and SOI dataset.

```r
SOColumnName <- c("OrderID", "Date", "BuyerID", "NrItems",
                "OrderValue", "ShippingCost", "OrderTotal",
                "City", "Country")

# Load Sales Order Data for 2017-03
SO1703 <- read.csv("./data/SoldOrders201703.csv", sep=",")
SO1703 <- subset(SO1703,
                 select=c(Order.ID,Sale.Date,Buyer.User.ID,
                          Number.of.Items,Order.Value,Shipping,
                          Order.Total,Ship.City,Ship.Country))
colnames(SO1703) <- SOColumnName
SO1703$Date <- as.Date(SO1703$Date,format = "%m/%d/%y")

# Load Sales Order Data for 2017-04
SO1704 <- read.csv("./data/SoldOrders201704.csv", sep=",")
SO1704 <- subset(SO1704,
                 select=c(Order.ID,Sale.Date,Buyer.User.ID,
                          Number.of.Items,Order.Value,Shipping,
                          Order.Total,Ship.City,Ship.Country))
colnames(SO1704) <- SOColumnName
SO1704$Date <- as.Date(SO1704$Date,format = "%m/%d/%y")

# Load Sales Order Data from 2016-09 to 2017-02
SO1617 <- read.csv("./data/SoldOrders201609-201702.csv",sep=",")
SO1617 <- subset(SO1617,
                 select=c(Order.ID,Sale.Date,Buyer.User.ID,
                          Number.of.Items,Order.Value,Shipping,
                          Order.Total,Ship.City,Ship.Country))
colnames(SO1617) <- SOColumnName
SO1617$Date <- as.Date(SO1617$Date,format = "%m/%d/%y")

# Merge into 1 SO dataset
SO <- rbind(SO1704, SO1703)
SO <- rbind(SO, SO1617)

## Load Sales Order Item data
SOIColumnName <- c("Date","Qty","ItemPrice","Buyer",
                   "ItemPriceDiscounted", "OrderID")
```

```
SOI1703 <- read.csv("./data/SoldOrderItems201703.csv", sep=",")
SOI1703 <- subset(SOI1703,
                 select=c("Sale.Date","Quantity",
                          "Price","Buyer","Item.Total",
                          "Order.ID"))
colnames(SOI1703) <- SOIColumnName
SOI1703$Date <- as.Date(SOI1703$Date,format = "%m/%d/%y")
SOI1704 <- read.csv("./data/SoldOrderItems201704.csv", sep=",")
SOI1704 <- subset(SOI1704,
                 select=c("Sale.Date","Quantity",
                          "Price","Buyer","Item.Total",
                          "Order.ID"))
colnames(SOI1704) <- SOIColumnName
SOI1704$Date <- as.Date(SOI1704$Date,format = "%m/%d/%y")
SOI <- rbind(SOI1704,SOI1703)
```

This part subsets the main SO and SOI table into treatment and control datasets.

```
## Sales Order Data
SO_cont1 <- subset(SO, SO$Date > "2017-03-12" & SO$Date < "2017-03-20")
SO_treat <- subset(SO, SO$Date > "2017-03-19" & SO$Date < "2017-03-27")
SO_cont2 <- subset(SO, SO$Date > "2017-03-26" & SO$Date < "2017-04-03")
SO_cont3 <- subset(SO, SO$Date > "2017-04-02" & SO$Date < "2017-04-10")
SO_base <- subset(SO, SO$Date < "2017-03-13")
## Sales Order Item Data
SOI_cont1 <- subset(SOI, SOI$Date > "2017-03-12" & SOI$Date < "2017-03-20")
SOI_treat <- subset(SOI, SOI$Date > "2017-03-19" & SOI$Date < "2017-03-27")
SOI_cont2 <- subset(SOI, SOI$Date > "2017-03-26" & SOI$Date < "2017-04-03")
SOI_cont3 <- subset(SOI, SOI$Date > "2017-04-02" & SOI$Date < "2017-04-10")
sum(SO_treat$NrItems)
```

```
## [1] 67
```

```
sum(SO_cont1$NrItems)
```

```
## [1] 32
```

```
sum(SOI_treat$Qty)
```

```
## [1] 67
```

```
sum(SOI_cont1$Qty)
```

```
## [1] 32
```

Cross checking the total number of items in both data sources, they should be the same. This is just a check on the trustworthiness of our sales data source.

1.2 Load Websession Data

The web session data is gathered from Google Analytics(GA). Total number of web sessions per day is shown in GA. We have prepared the web session data into pre-treatment, treatment and 2 post-treatment weeks. Within the GA data, unique visitors are identified as well. (assuming user doesn't clean cookies)

```
session_control1 <- read.csv("./data/websession_control1.csv", sep=",")
session_control2 <- read.csv("./data/websession_control2.csv", sep=",")
```

```
session_control3 <- read.csv("./data/websession_control3.csv", sep=",")
session_treatment <- read.csv("./data/websession_treatment.csv", sep=",")
```

1.3 Load Demographic Data

Demographics data are also gathered from GA aggregated on day level. GA provides web session age, gender, country, language, affinity group, market segment and many other factors that we can use for covariate balance checks. These are variables that cannot be changed with the treatment we are offering thus can be seen as pre-experimental covariants.

```
d_age <- read.csv("./data/GA_age.csv", sep=",")
d_gender <- read.csv("./data/GA_gender.csv", sep=",")
d_country_cont1 <- read.csv(file="./data/GA_country_cont1.csv", sep=",")
d_country_treat <- read.csv(file="./data/GA_country_treat.csv", sep=",")
d_country_cont2 <- read.csv(file="./data/GA_country_cont2.csv", sep=",")
d_lang_cont1 <- read.csv(file="./data/GA_lang_cont1.csv", sep=",")
d_lang_treat <- read.csv(file="./data/GA_lang_treat.csv", sep=",")
d_lang_cont2 <- read.csv(file="./data/GA_lang_cont2.csv", sep=",")
```

2. Covariate Balance

Covariate Balance checks that other factors of sales stays in similar level across control and treatment weeks. This assures the randomness of treatment or control given a single website visit.

2.1 Age

```
## -----Contorl week 1 V.s. Treatment Week-----

# Check balance cont1 v.s. treat of age group 18-24
t.test(as.numeric(d_age[1,2:8]),
       as.numeric(d_age[1,9:15]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[1, 2:8]) and as.numeric(d_age[1, 9:15])
## t = 0.85797, df = 6, p-value = 0.4238
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.143321 14.857606
## sample estimates:
## mean of the differences
##                3.857143
```

```
wilcox.test(as.numeric(d_age[1,2:8]),
            as.numeric(d_age[1,9:15]), paired = TRUE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  as.numeric(d_age[1, 2:8]) and as.numeric(d_age[1, 9:15])
## V = 19, p-value = 0.4688
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Check balance cont1 v.s. treat of age group 15-34
t.test(as.numeric(d_age[2,2:8]),
       as.numeric(d_age[2,9:15]), paired = TRUE)
```

```
## 
##  Paired t-test
## 
## data:  as.numeric(d_age[2, 2:8]) and as.numeric(d_age[2, 9:15])
## t = -1.8297, df = 6, p-value = 0.117
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -12.688407   1.831264
## sample estimates:
## mean of the differences 
##               -5.428571
```

```r
wilcox.test(as.numeric(d_age[2,2:8]),
            as.numeric(d_age[2,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[2, 2:8]),
## as.numeric(d_age[2, : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(as.numeric(d_age[2, 2:8]),
## as.numeric(d_age[2, : cannot compute exact p-value with zeroes
```

```
## 
##  Wilcoxon signed rank test with continuity correction
## 
## data:  as.numeric(d_age[2, 2:8]) and as.numeric(d_age[2, 9:15])
## V = 3.5, p-value = 0.1718
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Check balance cont1 v.s. treat of age group 35-44
t.test(as.numeric(d_age[3,2:8]),
       as.numeric(d_age[3,9:15]), paired = TRUE)
```

```
## 
##  Paired t-test
## 
## data:  as.numeric(d_age[3, 2:8]) and as.numeric(d_age[3, 9:15])
## t = -0.6563, df = 6, p-value = 0.536
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -10.807593   6.236164
## sample estimates:
## mean of the differences 
##               -2.285714
```

```r
wilcox.test(as.numeric(d_age[3,2:8]),
            as.numeric(d_age[3,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[3, 2:8]),
## as.numeric(d_age[3, : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(as.numeric(d_age[3, 2:8]),
## as.numeric(d_age[3, : cannot compute exact p-value with zeroes
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[3, 2:8]) and as.numeric(d_age[3, 9:15])
## V = 7, p-value = 0.5271
## alternative hypothesis: true location shift is not equal to 0
```

```
# Check balance cont1 v.s. treat of age group 45-54
t.test(as.numeric(d_age[4,2:8]),
       as.numeric(d_age[4,9:15]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[4, 2:8]) and as.numeric(d_age[4, 9:15])
## t = -1.082, df = 6, p-value = 0.3208
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -13.045855   5.045855
## sample estimates:
## mean of the differences
##                      -4
```

```
wilcox.test(as.numeric(d_age[4,2:8]),
            as.numeric(d_age[4,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[4, 2:8]),
## as.numeric(d_age[4, : cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[4, 2:8]) and as.numeric(d_age[4, 9:15])
## V = 2, p-value = 0.3613
## alternative hypothesis: true location shift is not equal to 0
```

```
# Check balance cont1 v.s. treat of all age groups
t.test(as.numeric(d_age[5,2:8]),
       as.numeric(d_age[5,9:15]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[5, 2:8]) and as.numeric(d_age[5, 9:15])
## t = -0.74001, df = 6, p-value = 0.4872
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -33.83762  18.12333
## sample estimates:
## mean of the differences
##               -7.857143
```

```
wilcox.test(as.numeric(d_age[5,2:8]),
            as.numeric(d_age[5,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[5, 2:8]),
## as.numeric(d_age[5, : cannot compute exact p-value with ties
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[5, 2:8]) and as.numeric(d_age[5, 9:15])
## V = 10.5, p-value = 0.6108
## alternative hypothesis: true location shift is not equal to 0
```

```
## -----Contorl week 2 V.s. Treatment Week-----

# Check balance cont2 v.s. treat of age group 18-24
t.test(as.numeric(d_age[1,16:22]),
       as.numeric(d_age[1,9:15]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[1, 16:22]) and as.numeric(d_age[1, 9:15])
## t = -0.036572, df = 6, p-value = 0.972
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -9.701017  9.415303
## sample estimates:
## mean of the differences
##              -0.1428571
```

```
wilcox.test(as.numeric(d_age[1,16:22]),
            as.numeric(d_age[1,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[1, 16:22]),
## as.numeric(d_age[1, : cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[1, 16:22]) and as.numeric(d_age[1, 9:15])
## V = 10, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
```

```
# Check balance cont2 v.s. treat of age group 25-34
t.test(as.numeric(d_age[2,16:22]),
       as.numeric(d_age[2,9:15]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[2, 16:22]) and as.numeric(d_age[2, 9:15])
## t = -1.2718, df = 6, p-value = 0.2505
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -15.455013   4.883584
## sample estimates:
## mean of the differences
##               -5.285714
```

```
wilcox.test(as.numeric(d_age[2,16:22]),
            as.numeric(d_age[2,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[2, 16:22]),
## as.numeric(d_age[2, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[2, 16:22]) and as.numeric(d_age[2, 9:15])
## V = 7, p-value = 0.2685
## alternative hypothesis: true location shift is not equal to 0
```

```
# Check balance cont2 v.s. treat of age group 35-44
t.test(as.numeric(d_age[3,16:22]),
       as.numeric(d_age[3,9:15]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[3, 16:22]) and as.numeric(d_age[3, 9:15])
## t = -2.2529, df = 6, p-value = 0.06518
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.3909616   0.6766759
## sample estimates:
## mean of the differences
##               -7.857143
```

```
wilcox.test(as.numeric(d_age[3,16:22]),
            as.numeric(d_age[3,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[3, 16:22]),
## as.numeric(d_age[3, : cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[3, 16:22]) and as.numeric(d_age[3, 9:15])
## V = 3, p-value = 0.1422
## alternative hypothesis: true location shift is not equal to 0
```

```
# Check balance cont2 v.s. treat of age group 45-54
t.test(as.numeric(d_age[4,16:22]),
       as.numeric(d_age[4,9:15]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[4, 16:22]) and as.numeric(d_age[4, 9:15])
## t = -0.036515, df = 6, p-value = 0.9721
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -9.715920  9.430206
## sample estimates:
## mean of the differences
##              -0.1428571
```

```
wilcox.test(as.numeric(d_age[4,16:22]),
            as.numeric(d_age[4,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[4, 16:22]),
## as.numeric(d_age[4, : cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[4, 16:22]) and as.numeric(d_age[4, 9:15])
## V = 7, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Check balance cont2 v.s. treat of all age groups
t.test(as.numeric(d_age[5,16:22]),
       as.numeric(d_age[5,9:15]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[5, 16:22]) and as.numeric(d_age[5, 9:15])
## t = -1.0906, df = 6, p-value = 0.3173
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -43.55762  16.70048
## sample estimates:
## mean of the differences
##               -13.42857
```

```r
wilcox.test(as.numeric(d_age[5,16:22]),
            as.numeric(d_age[5,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[5, 16:22]),
## as.numeric(d_age[5, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[5, 16:22]) and as.numeric(d_age[5, 9:15])
## V = 8.5, p-value = 0.3972
## alternative hypothesis: true location shift is not equal to 0
```

```r
## -----Contorl week 2 V.s. Control Week 1-----

# Check balance cont2 v.s. cont1 of age group 18-24
t.test(as.numeric(d_age[1,16:22]),
       as.numeric(d_age[1,2:8]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[1, 16:22]) and as.numeric(d_age[1, 2:8])
## t = -0.96077, df = 6, p-value = 0.3738
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -14.187306   6.187306
## sample estimates:
## mean of the differences
##                      -4
```

```r
wilcox.test(as.numeric(d_age[1,16:22]),
            as.numeric(d_age[1,2:8]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[1, 16:22]),
## as.numeric(d_age[1, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[1, 16:22]) and as.numeric(d_age[1, 2:8])
## V = 8, p-value = 0.3517
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Check balance cont2 v.s. cont1 of age group 25-34
t.test(as.numeric(d_age[2,16:22]),
       as.numeric(d_age[2,2:8]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[2, 16:22]) and as.numeric(d_age[2, 2:8])
## t = 0.036629, df = 6, p-value = 0.972
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -9.400377  9.686091
## sample estimates:
## mean of the differences
##               0.1428571
```

```r
wilcox.test(as.numeric(d_age[2,16:22]),
            as.numeric(d_age[2,2:8]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[2, 16:22]),
## as.numeric(d_age[2, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[2, 16:22]) and as.numeric(d_age[2, 2:8])
## V = 15, p-value = 0.9325
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Check balance cont2 v.s. cont1 of age group 35-44
t.test(as.numeric(d_age[3,16:22]),
       as.numeric(d_age[3,2:8]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[3, 16:22]) and as.numeric(d_age[3, 2:8])
## t = -1.4945, df = 6, p-value = 0.1857
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -14.693508   3.550651
## sample estimates:
## mean of the differences
##               -5.571429
```

```r
wilcox.test(as.numeric(d_age[3,16:22]),
            as.numeric(d_age[3,2:8]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[3, 16:22]),
## as.numeric(d_age[3, : cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[3, 16:22]) and as.numeric(d_age[3, 2:8])
## V = 1, p-value = 0.2012
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Check balance cont2 v.s. cont1 of age group 45-54
t.test(as.numeric(d_age[4,16:22]),
       as.numeric(d_age[4,2:8]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[4, 16:22]) and as.numeric(d_age[4, 2:8])
## t = 1.3955, df = 6, p-value = 0.2123
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.906015 10.620301
## sample estimates:
## mean of the differences
##                3.857143
```

```r
wilcox.test(as.numeric(d_age[4,16:22]),
            as.numeric(d_age[4,2:8]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[4, 16:22]),
## as.numeric(d_age[4, : cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[4, 16:22]) and as.numeric(d_age[4, 2:8])
## V = 5, p-value = 0.4227
## alternative hypothesis: true location shift is not equal to 0
```

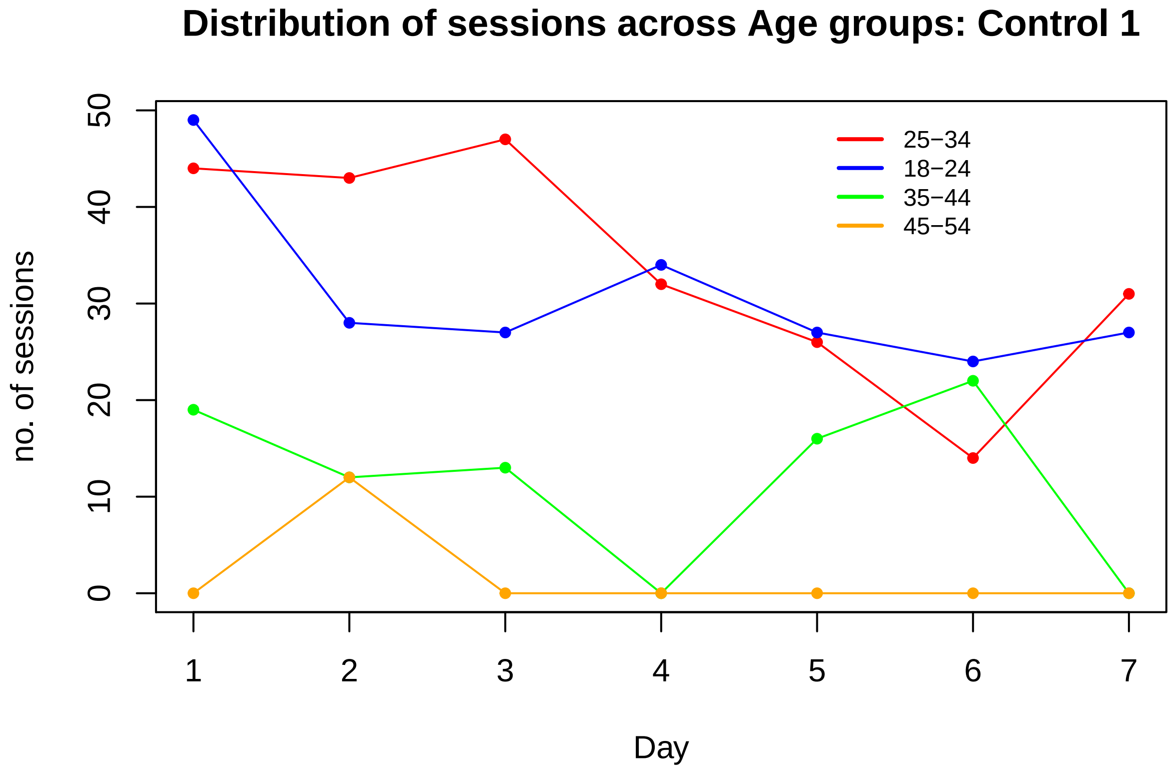
```r
# Check balance cont2 v.s. cont1 of age group 45-54
t.test(as.numeric(d_age[5,16:22]),
       as.numeric(d_age[5,2:8]), paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  as.numeric(d_age[5, 16:22]) and as.numeric(d_age[5, 2:8])
## t = -0.8228, df = 6, p-value = 0.4421
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -22.14017  10.99732
## sample estimates:
## mean of the differences
##               -5.571429
```

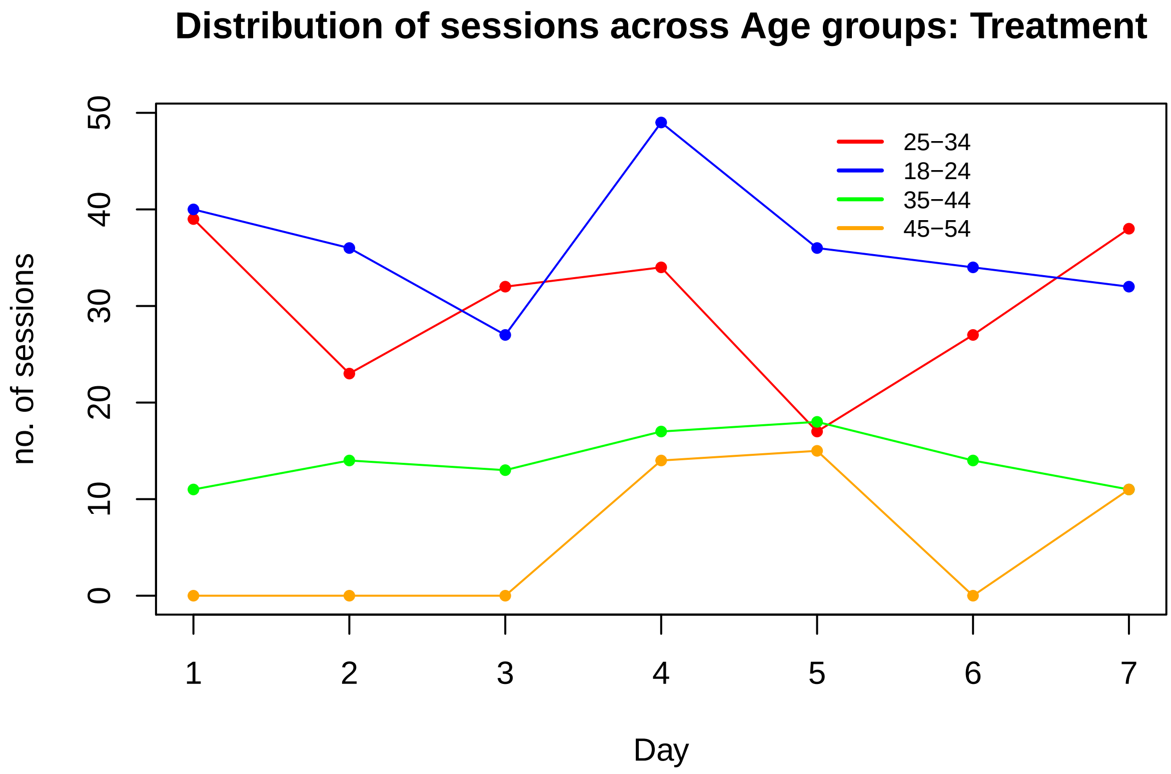
```r
wilcox.test(as.numeric(d_age[5,16:22]),
            as.numeric(d_age[5,2:8]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_age[5, 16:22]),
## as.numeric(d_age[5, : cannot compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_age[5, 16:22]) and as.numeric(d_age[5, 2:8])
## V = 8, p-value = 0.675
## alternative hypothesis: true location shift is not equal to 0
```

```r
# Plot the age group day by day during the Control 1 week
plot(c(1:7),d_age[1,2:8],
     type = "l",
     main = "Distribution of sessions across Age groups: Control 1",
     xlab= "Day", ylab = "no. of sessions",
     ylim = c(min(d_age[1:4,2:8]),
              max(d_age[1:4,2:8])),col = "red")
points(c(1:7),d_age[1,2:8], pch = 20, col = "red")
lines(c(1:7),d_age[2,2:8],type = "l",col = "blue")
points(c(1:7),d_age[2,2:8], pch = 20, col = "blue")
lines(c(1:7),d_age[3,2:8],type = "l", col = "green")
points(c(1:7),d_age[3,2:8], pch = 20, col = "green")
lines(c(1:7),d_age[4,2:8],type = "l", col = "orange")
points(c(1:7),d_age[4,2:8], pch = 20, col = "orange")
legend(x=5, y=50, legend = c("25-34", "18-24", "35-44", "45-54"),
       col = c("red","blue", "green", "orange"),
       lwd = 2, bty = "n", cex = 0.75)
```

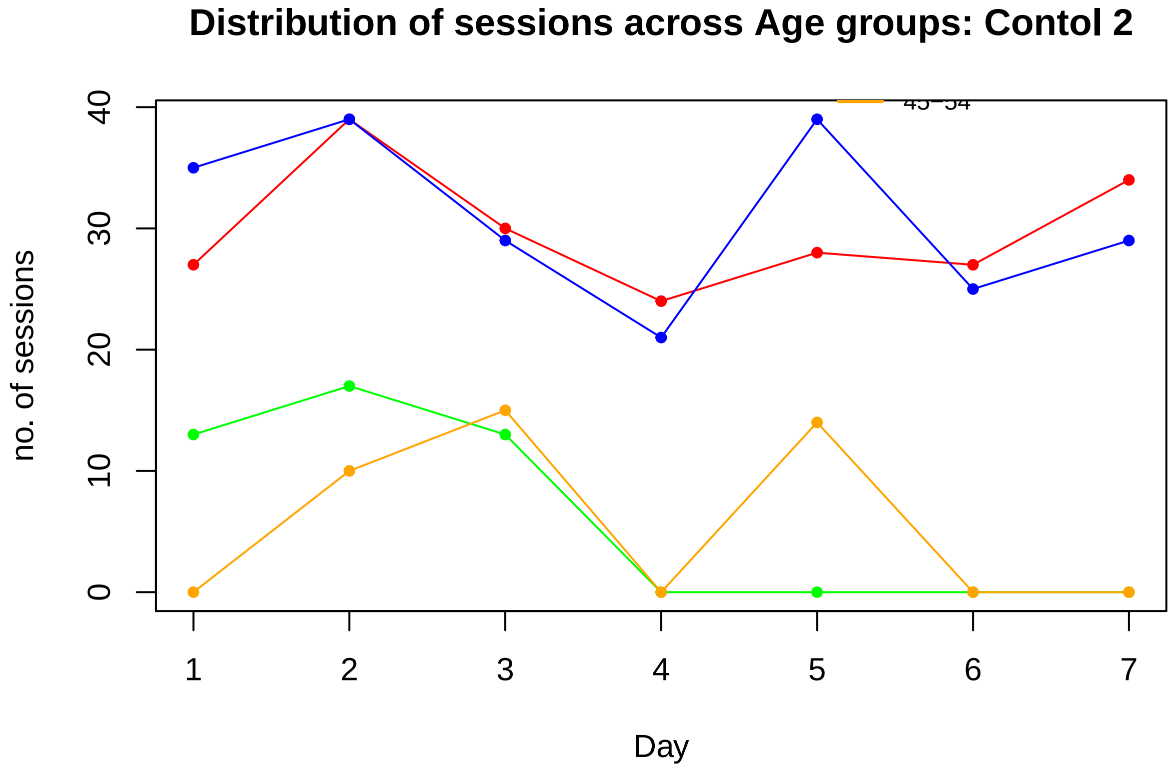


```
# Plot the age group day by day during the Treatment week
plot(c(1:7),d_age[1,9:15],
     type = "l",
     main = "Distribution of sessions across Age groups: Treatment",
     xlab= "Day", ylab = "no. of sessions",
     ylim = c(min(d_age[1:4,9:15]),
              max(d_age[1:4,9:15])),col = "red")
points(c(1:7),d_age[1,9:15], pch = 20, col = "red")
lines(c(1:7),d_age[2,9:15],type = "l",col = "blue")
points(c(1:7),d_age[2,9:15], pch = 20, col = "blue")
lines(c(1:7),d_age[3,9:15],type = "l", col = "green")
points(c(1:7),d_age[3,9:15], pch = 20, col = "green")
lines(c(1:7),d_age[4,9:15],type = "l", col = "orange")
points(c(1:7),d_age[4,9:15], pch = 20, col = "orange")
legend(x=5, y=50, legend = c("25-34", "18-24", "35-44", "45-54"),
       col = c("red","blue", "green", "orange"),
       lwd = 2, bty = "n", cex = 0.75)
```



```
# Plot the age group day by day during the Contol 2 week
plot(c(1:7),d_age[1,16:22],
     type = "l",
     main = "Distribution of sessions across Age groups: Contol 2",
     xlab= "Day", ylab = "no. of sessions",
     ylim = c(min(d_age[1:4,16:22]),
              max(d_age[1:4,16:22])),col = "red")
points(c(1:7),d_age[1,16:22], pch = 20, col = "red")
lines(c(1:7),d_age[2,16:22],type = "l",col = "blue")
points(c(1:7),d_age[2,16:22], pch = 20, col = "blue")
lines(c(1:7),d_age[3,16:22],type = "l", col = "green")
```

```r
points(c(1:7),d_age[3,16:22], pch = 20, col = "green")
lines(c(1:7),d_age[4,16:22],type = "l", col = "orange")
points(c(1:7),d_age[4,16:22], pch = 20, col = "orange")
legend(x=5, y=50, legend = c("25-34", "18-24", "35-44", "45-54"),
       col = c("red","blue", "green", "orange"),
       lwd = 2, bty = "n", cex = 0.75)
```



From the aggregated data on day level, we see that number of web sessions per each age group per day is balanced across treatment, control 1 and control 2 weeks. The fact that both t test and wilcox test failed to reject the null hypothesis gives evidence to the randomization assumption. Ideally, if GA is able to provide demographic data on single web session other than the daily aggregated results, we would be having a much bigger sample size and thus the result will be much more accurate. Also, since we are comparing across all age groups, this is a multiple comparison situation. Thus, the critical p value can be 4-5 times smaller than 0.05, giving stronger base for covariate balance.

2.2 Gender

```r
## -----Control 1 v.s. Treatment-----
wilcox.test(as.numeric(d_gender[1,2:8]),
            as.numeric(d_gender[1,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_gender[1, 2:8]),
## as.numeric(d_gender[1, : cannot compute exact p-value with ties

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_gender[1, 2:8]) and as.numeric(d_gender[1, 9:15])
## V = 10.5, p-value = 0.6115
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(as.numeric(d_gender[2,2:8]),
            as.numeric(d_gender[2,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_gender[2, 2:8]),
## as.numeric(d_gender[2, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_gender[2, 2:8]) and as.numeric(d_gender[2, 9:15])
## V = 20, p-value = 0.3517
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(as.numeric(d_gender[3,2:8]),
            as.numeric(d_gender[3,9:15]), paired = TRUE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  as.numeric(d_gender[3, 2:8]) and as.numeric(d_gender[3, 9:15])
## V = 13, p-value = 0.9375
## alternative hypothesis: true location shift is not equal to 0
```

```r
#cont1_gender_comp <- c(rep(0,sum(d_gender[1,2:8])), rep(1,sum(d_gender[2,2:8])))
#treat_gender_comp <- c(rep(0,sum(d_gender[1,9:15])), rep(1,sum(d_gender[2,9:15])))
#t.test(cont1_gender_comp, treat_gender_comp)

## -----Control 2 v.s. Treatment-----
wilcox.test(as.numeric(d_gender[1,16:22]),
            as.numeric(d_gender[1,9:15]), paired = TRUE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  as.numeric(d_gender[1, 16:22]) and as.numeric(d_gender[1, 9:15])
## V = 12, p-value = 0.8125
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(as.numeric(d_gender[2,16:22]),
            as.numeric(d_gender[2,9:15]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_gender[2, 16:22]),
## as.numeric(d_gender[2, : cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(as.numeric(d_gender[2, 16:22]),
## as.numeric(d_gender[2, : cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_gender[2, 16:22]) and as.numeric(d_gender[2, 9:15])
## V = 5, p-value = 0.5879
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(as.numeric(d_gender[3,16:22]),
            as.numeric(d_gender[3,9:15]), paired = TRUE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  as.numeric(d_gender[3, 16:22]) and as.numeric(d_gender[3, 9:15])
## V = 10, p-value = 0.5781
## alternative hypothesis: true location shift is not equal to 0
```

```
#cont2_gender_comp <- c(rep(0,sum(d_gender[1,16:22])), rep(1,sum(d_gender[2,16:22])))
#t.test(cont2_gender_comp, treat_gender_comp)

## -----Control 2 v.s. Control 1-----
wilcox.test(as.numeric(d_gender[1,16:22]),
            as.numeric(d_gender[1,2:8]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_gender[1, 16:22]),
## as.numeric(d_gender[1, : cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_gender[1, 16:22]) and as.numeric(d_gender[1, 2:8])
## V = 10, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(as.numeric(d_gender[2,16:22]),
            as.numeric(d_gender[2,2:8]), paired = TRUE)
```

```
## Warning in wilcox.test.default(as.numeric(d_gender[2, 16:22]),
## as.numeric(d_gender[2, : cannot compute exact p-value with zeroes
```

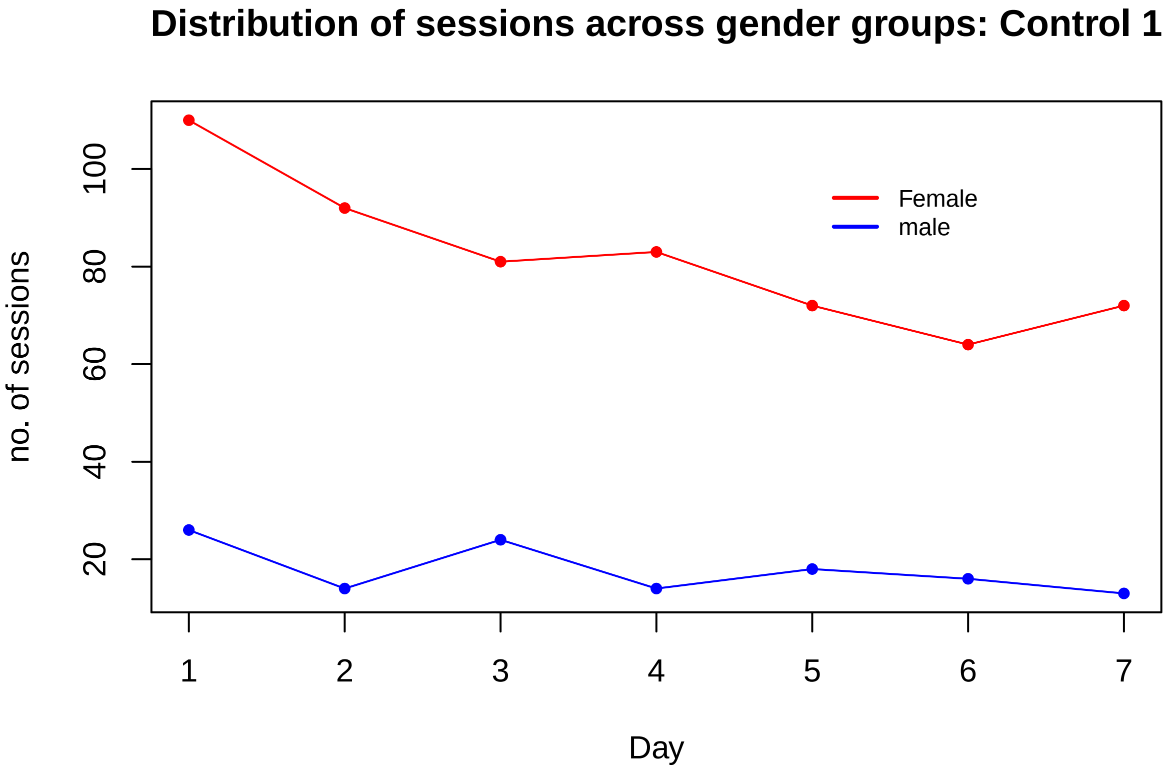
```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_gender[2, 16:22]) and as.numeric(d_gender[2, 2:8])
## V = 1, p-value = 0.1056
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(as.numeric(d_gender[3,16:22]),
            as.numeric(d_gender[3,2:8]), paired = TRUE)
```

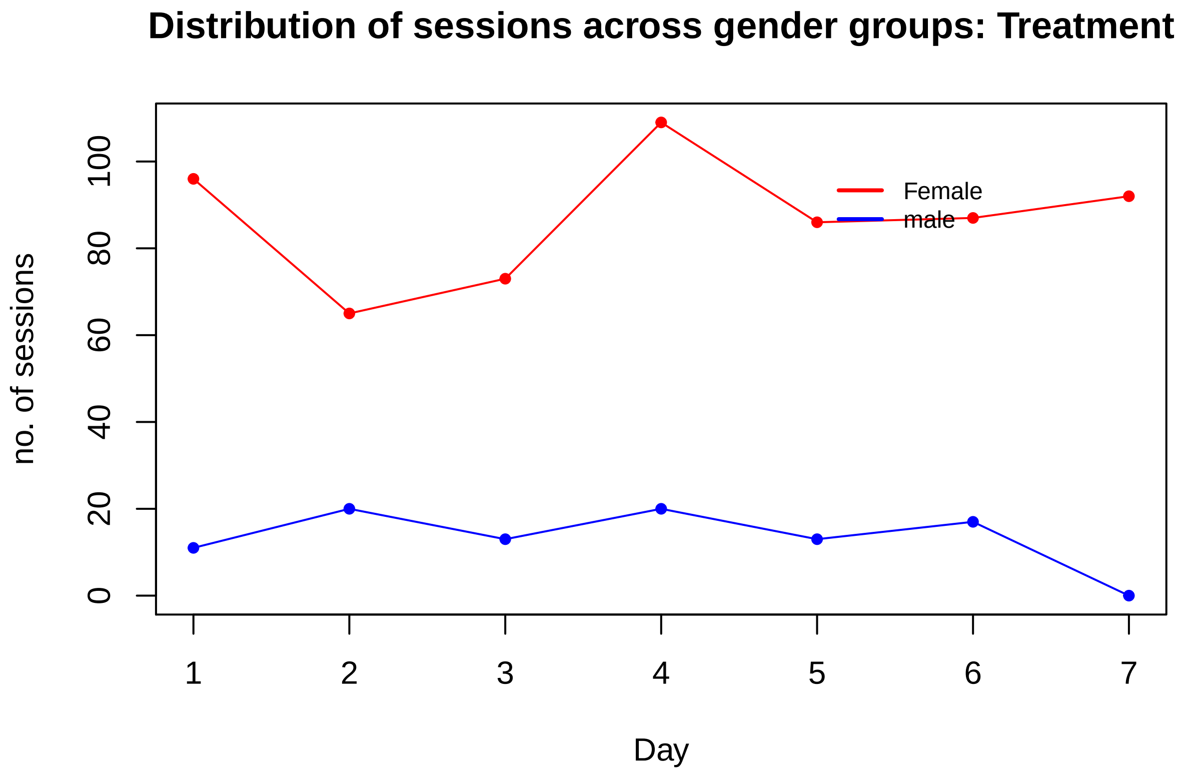
```
## Warning in wilcox.test.default(as.numeric(d_gender[3, 16:22]),
## as.numeric(d_gender[3, : cannot compute exact p-value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  as.numeric(d_gender[3, 16:22]) and as.numeric(d_gender[3, 2:8])
## V = 11.5, p-value = 0.7349
## alternative hypothesis: true location shift is not equal to 0
```

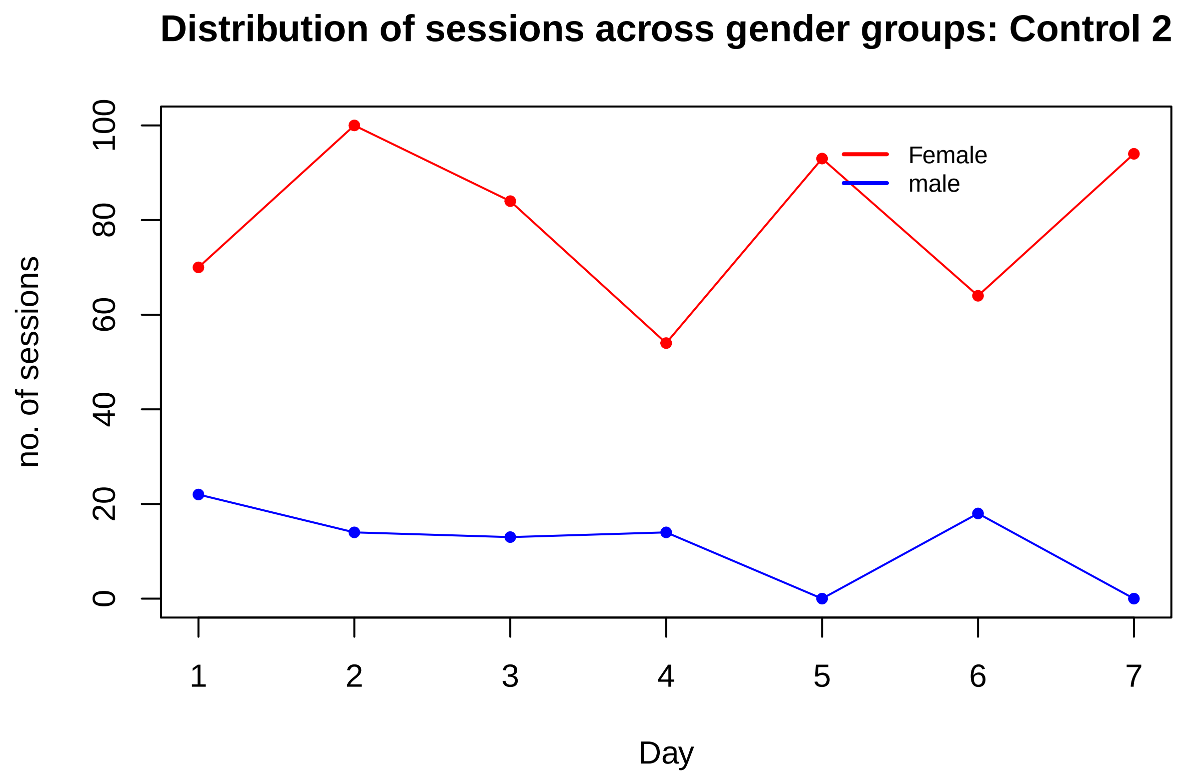
```
#t.test(cont1_gender_comp, cont2_gender_comp)

## Plot the distribution of gender for Control 1 week
plot(c(1:7),d_gender[1,2:8],
     type = "l",
     main = "Distribution of sessions across gender groups: Control 1",
     xlab= "Day", ylab = "no. of sessions",
     ylim = c(min(d_gender[1:2,2:8]),
               max(d_gender[1:2,2:8])),col = "red")
points(c(1:7),d_gender[1,2:8], pch = 20, col = "red")
lines(c(1:7),d_gender[2,2:8],type = "l",col = "blue")
points(c(1:7),d_gender[2,2:8], pch = 20, col = "blue")
legend(x=5, y=100, legend = c("Female", "male"),
       col = c("red","blue"),
       lwd = 2, bty = "n", cex = 0.75)
```



```
## Plot the distribution of gender for Treatment week
plot(c(1:7),d_gender[1,9:15],
     type = "l",
     main = "Distribution of sessions across gender groups: Treatment",
     xlab= "Day", ylab = "no. of sessions",
     ylim = c(min(d_gender[1:2,9:15]),
              max(d_gender[1:2,9:15])),col = "red")
points(c(1:7),d_gender[1,9:15], pch = 20, col = "red")
lines(c(1:7),d_gender[2,9:15],type = "l",col = "blue")
points(c(1:7),d_gender[2,9:15], pch = 20, col = "blue")
legend(x=5, y=100, legend = c("Female", "male"),
       col = c("red","blue"),
       lwd = 2, bty = "n", cex = 0.75)
```



```
## Plot the distribution of gender for Control 2 week
plot(c(1:7),d_gender[1,16:22],
     type = "l",
     main = "Distribution of sessions across gender groups: Control 2",
     xlab= "Day", ylab = "no. of sessions",
     ylim = c(min(d_gender[1:2,16:22]),
              max(d_gender[1:2,16:22])),col = "red")
points(c(1:7),d_gender[1,16:22], pch = 20, col = "red")
lines(c(1:7),d_gender[2,16:22],type = "l",col = "blue")
points(c(1:7),d_gender[2,16:22], pch = 20, col = "blue")
legend(x=5, y=100, legend = c("Female", "male"),
       col = c("red","blue"),
       lwd = 2, bty = "n", cex = 0.75)
```



From gender data we see also that covariate balance is satisfied. The gender GA data comes from the same mechanism as age groups.

2.3 Country

The following section test the covariate balance for web session origin countries. Since we have 67 different countries/categories, we use “Matched Pair Wilcoxon Test”" to see if the distribution is identical for common countries that are found for both treatment and control weeks.

```
## Control 1 v.s. Treatment
common_country <- intersect(d_country_cont1$Country,d_country_treat$Country)
country_cont1 <- subset(d_country_cont1, d_country_cont1$Country %in% common_country)
country_treat <- subset(d_country_treat, d_country_treat$Country %in% common_country)
country_cont1 <- country_cont1[order(country_cont1$Country),1:2]
country_treat <- country_treat[order(country_treat$Country),1:2]
country <- cbind(country_cont1,country_treat$Sessions)
names(country) <- c("Country", "Sess.control1", "Sess.treat")
wilcox.test(country$Sess.control1,country$Sess.treat, paired = T)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  country$Sess.control1 and country$Sess.treat
## V = 869, p-value = 0.9066
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(country$Sess.control1,country$Sess.treat, paired = T)
```

```
##
##  Paired t-test
```

```
##
## data:  country$Sess.control1 and country$Sess.treat
## t = -1.0864, df = 66, p-value = 0.2812
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.473033  1.025272
## sample estimates:
## mean of the differences
##               -1.223881
```

```
## Treatment v.s. Control 2
common_country <- intersect(d_country_cont2$Country,d_country_treat$Country)
country_cont2 <- subset(d_country_cont2, d_country_cont2$Country %in% common_country)
country_treat <- subset(d_country_treat, d_country_treat$Country %in% common_country)
country_cont2 <- country_cont2[order(country_cont2$Country),1:2]
country_treat <- country_treat[order(country_treat$Country),1:2]
country <- cbind(country_cont2,country_treat$Sessions)
names(country) <- c("Country", "Sess.control2", "Sess.treat")
wilcox.test(country$Sess.control2,country$Sess.treat, paired = T)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  country$Sess.control2 and country$Sess.treat
## V = 660, p-value = 0.2608
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(country$Sess.control2,country$Sess.treat, paired = T)
```

```
##
##  Paired t-test
##
## data:  country$Sess.control2 and country$Sess.treat
## t = -1.5184, df = 62, p-value = 0.134
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.0814870  0.5576774
## sample estimates:
## mean of the differences
##               -1.761905
```

```
## Control 1 v.s. Control 2
common_country <- intersect(d_country_cont1$Country,d_country_cont2$Country)
country_cont1 <- subset(d_country_cont1, d_country_cont1$Country %in% common_country)
country_cont2 <- subset(d_country_cont2, d_country_cont2$Country %in% common_country)
country_cont1 <- country_cont1[order(country_cont1$Country),1:2]
country_cont2 <- country_cont2[order(country_cont2$Country),1:2]
country <- cbind(country_cont1,country_cont2$Sessions)
names(country) <- c("Country", "Sess.control1", "Sess.control2")
wilcox.test(country$Sess.control1,country$Sess.control2, paired = T)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  country$Sess.control1 and country$Sess.control2
## V = 906, p-value = 0.8765
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(country$Sess.control1,country$Sess.control2, paired = T)
```

```
##
##  Paired t-test
##
## data:  country$Sess.control1 and country$Sess.control2
## t = 0.4805, df = 64, p-value = 0.6325
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.505916  2.459762
## sample estimates:
## mean of the differences
##               0.4769231
```

We see that since the number of sessions across countries are very different, the distribution is not close to normal. Thus, a paired Wilcoxon test will yield better results. From that we see the country is balanced across the experimental weeks.

2.4 Language

The following section test the covariate balance for the language of web session visitor. This data is gathered from GA. GA detects the browser language of the online shop visitors. Since we have 58 different languages/categories, we use “Matched Pair Wilcoxon Test”" to see if the distribution is identical for common countries that are found for both treatment and control weeks.

```
## Control 1 v.s. Treatment
common_lang <- intersect(d_lang_cont1$Language,d_lang_treat$Language)
lang_cont1 <- subset(d_lang_cont1, d_lang_cont1$Language %in% common_lang)
lang_treat <- subset(d_lang_treat, d_lang_treat$Language %in% common_lang)
lang_cont1 <- lang_cont1[order(lang_cont1$Language),1:2]
lang_treat <- lang_treat[order(lang_treat$Language),1:2]
lang <- cbind(lang_cont1,lang_treat$Sessions)
names(lang) <- c("Language", "Sess.control1", "Sess.treat")
wilcox.test(lang$Sess.control1,lang$Sess.treat, paired = T)
```

```
## Warning in wilcox.test.default(lang$Sess.control1, lang$Sess.treat, paired
## = T): cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(lang$Sess.control1, lang$Sess.treat, paired
## = T): cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  lang$Sess.control1 and lang$Sess.treat
## V = 602.5, p-value = 0.9242
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(lang$Sess.control1,lang$Sess.treat, paired = T)
```

```
##
##  Paired t-test
##
## data:  lang$Sess.control1 and lang$Sess.treat
## t = -0.90231, df = 57, p-value = 0.3707
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.384866  1.660728
## sample estimates:
## mean of the differences
##               -1.362069
```

```
## Treatment v.s. Control 2
common_lang <- intersect(d_lang_cont2$Language,d_lang_treat$Language)
lang_cont2 <- subset(d_lang_cont2, d_lang_cont2$Language %in% common_lang)
lang_treat <- subset(d_lang_treat, d_lang_treat$Language %in% common_lang)
lang_cont2 <- lang_cont2[order(lang_cont2$Language),1:2]
lang_treat <- lang_treat[order(lang_treat$Language),1:2]
lang <- cbind(lang_cont2,lang_treat$Sessions)
names(lang) <- c("Language", "Sess.control2", "Sess.treat")
wilcox.test(lang$Sess.control2,lang$Sess.treat, paired = T)
```

```
## Warning in wilcox.test.default(lang$Sess.control2, lang$Sess.treat, paired
## = T): cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(lang$Sess.control2, lang$Sess.treat, paired
## = T): cannot compute exact p-value with zeroes
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  lang$Sess.control2 and lang$Sess.treat
## V = 343.5, p-value = 0.04948
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(lang$Sess.control2,lang$Sess.treat, paired = T)
```

```
##
##  Paired t-test
##
## data:  lang$Sess.control2 and lang$Sess.treat
## t = -1.7369, df = 56, p-value = 0.08791
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.3445003  0.3094126
## sample estimates:
## mean of the differences
##               -2.017544
```

```
## Control 1 v.s. Control 2
common_lang <- intersect(d_lang_cont1$Language,d_lang_cont2$Language)
lang_cont1 <- subset(d_lang_cont1, d_lang_cont1$Language %in% common_lang)
lang_cont2 <- subset(d_lang_cont2, d_lang_cont2$Language %in% common_lang)
lang_cont1 <- lang_cont1[order(lang_cont1$Language),1:2]
lang_cont2 <- lang_cont2[order(lang_cont2$Language),1:2]
lang <- cbind(lang_cont1,lang_cont2$Sessions)
names(lang) <- c("Language", "Sess.control1", "Sess.control2")
wilcox.test(lang$Sess.control1,lang$Sess.control2, paired = T)
```

```
## Warning in wilcox.test.default(lang$Sess.control1, lang$Sess.control2,
## paired = T): cannot compute exact p-value with ties
```

```
## Warning in wilcox.test.default(lang$Sess.control1, lang$Sess.control2,
## paired = T): cannot compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  lang$Sess.control1 and lang$Sess.control2
## V = 543, p-value = 0.3996
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(lang$Sess.control1,lang$Sess.control2, paired = T)
```

```
##
##  Paired t-test
##
## data:  lang$Sess.control1 and lang$Sess.control2
## t = 0.55619, df = 54, p-value = 0.5804
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.278661  2.260479
## sample estimates:
## mean of the differences
##               0.4909091
```

The comparison of control 2 and treatment shows a smaller p value (significant under 0.05 for Wilcoxon paired test). However, because again by comparing all the possible combinations, we are performing a multiple comparison, the critical p value should be several times smaller than 0.05. Thus, this level of significance shouldn't be too worring.

2.5 Average number of returning web sessions per week

This compares the average number of web sessions for a unique visitor. GA provides the number of returning to the website on unique visitor level. (assuming visitors do not clean up cookies) And again, we see that across the experimental weeks, the average number of sessions seems for a visitor remains stable.

```
t.test(session_control1$Sessions, session_treatment$Sessions)
```

```
##
##  Welch Two Sample t-test
##
## data:  session_control1$Sessions and session_treatment$Sessions
## t = 0.031802, df = 2019.6, p-value = 0.9746
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.08921820  0.09215946
## sample estimates:
## mean of x mean of y
##  1.187117  1.185646
```

```
t.test(session_control2$Sessions, session_treatment$Sessions)
```

```
##
##  Welch Two Sample t-test
##
## data:  session_control2$Sessions and session_treatment$Sessions
## t = 0.20635, df = 1957.7, p-value = 0.8365
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.07707255  0.09519868
## sample estimates:
## mean of x mean of y
##  1.194709  1.185646
```

```r
t.test(session_control1$Sessions, session_control2$Sessions)
```

```
##
##  Welch Two Sample t-test
##
## data:  session_control1$Sessions and session_control2$Sessions
## t = -0.17888, df = 1902.3, p-value = 0.8581
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.09083619  0.07565133
## sample estimates:
## mean of x mean of y
##  1.187117  1.194709
```

2.6 Affinity Group and Market Segment

As discussed during office hours, since how GA generated the affinity group and market segment data is unknown to us, using that as covariate balance check can be misinterpreting the results. It is to note that for a single web session, it can be labeled as multiple affinity groups and market segments. Thus, just by checking the balance of each affinity group or market segment may not gives us the real results we are expecting. We will not use these covariates in our analysis.

3. Treatment Effect Analysis

Below are estimations for treatement effects on sales performance.

```r
past_customer <- unique(SO[SO$Date < "2017-03-13",]$BuyerID)
new_customer <- setdiff(unique(SO[SO$Date > "2017-03-12" &
                                          SO$Date < "2017-03-27",]$BuyerID),
                                    past_customer)
intersect(unique(SO[SO$Date > "2017-03-12" &
                            SO$Date < "2017-03-20",]$BuyerID),
          unique(SO[SO$Date > "2017-03-19" &
                            SO$Date < "2017-03-27",]$BuyerID))
```

```
## character(0)
```

```r
# there's no customer who placed an order in both control 1 week and treatment week
intersect(unique(SO[SO$Date > "2017-03-26" &
                            SO$Date < "2017-04-09",]$BuyerID),
          unique(SO[SO$Date > "2017-03-19" &
                            SO$Date < "2017-03-27",]$BuyerID))
```

```
## [1] "danielornelas2"
```

```r
# there's only 1 buyer that placed an order post treatment
# that also placed a order during the treatment week

# add new indicator on returning customer orders
```

```
SO$Returning_Customer <- as.numeric(SO$BuyerID %in% past_customer)
SO$treat <- as.numeric(SO$Date > "2017-03-19" & SO$Date < "2017-03-27")
```

3.1 Order Value

Control 1 v.s. Treatment Sales Order Value Comparison

```
## average effect on order value of treatment v.s. control 1
SO_cont1_treat <- SO[SO$Date > "2017-03-12" & SO$Date < "2017-03-27",]
m1_ordervalue <- lm(OrderValue~treat, data=SO_cont1_treat)
summary(m1_ordervalue)
```

```
##
## Call:
## lm(formula = OrderValue ~ treat, data = SO_cont1_treat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.249 -13.849  -7.815  -0.825 113.931
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   14.805      5.381   2.751  0.00896 **
## treat          9.034      7.905   1.143  0.26005
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.24 on 39 degrees of freedom
## Multiple R-squared:  0.03241,    Adjusted R-squared:  0.007597
## F-statistic: 1.306 on 1 and 39 DF,  p-value: 0.2601
```

```
m1_ordervalue$vcov <- vcov(m1_ordervalue)
coeftest(m1_ordervalue, m1_ordervalue$vcov)
```

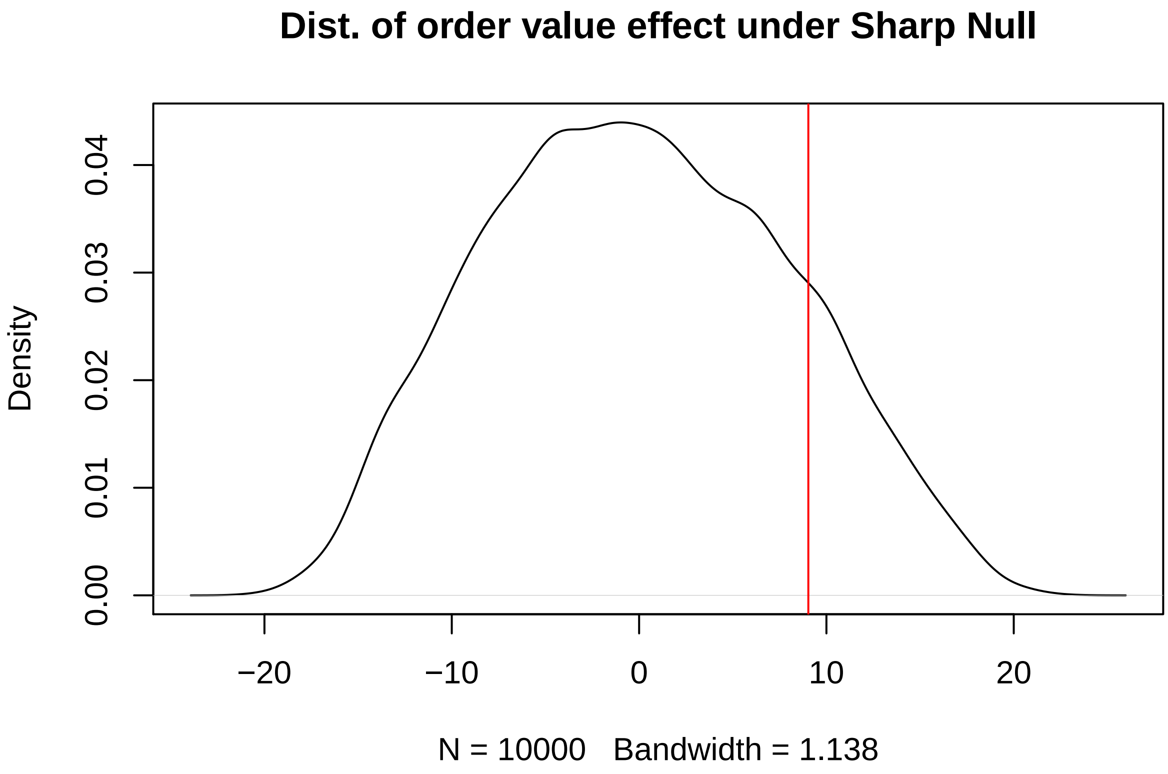
```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.8050     5.3813  2.7512 0.008959 **
## treat         9.0345     7.9050  1.1429 0.260053
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# non-parametric test
wilcox.test(SO_cont1$OrderValue,SO_treat$OrderValue)
```

```
## Warning in wilcox.test.default(SO_cont1$OrderValue, SO_treat$OrderValue):
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  SO_cont1$OrderValue and SO_treat$OrderValue
## W = 228, p-value = 0.6237
## alternative hypothesis: true location shift is not equal to 0
```

```
# ramdomization inference
ate <- mean(SO_treat$OrderValue) - mean(SO_cont1$OrderValue)
dist_sharpnull_ordervalue <- rep(0,10000)
for (i in 1:10000) {
  treat <- sample(SO_cont1_treat$treat)
  dist_sharpnull_ordervalue[i] <- mean(SO_cont1_treat[treat==1,]$OrderValue) -
    mean(SO_cont1_treat[treat==0,]$OrderValue)
}
plot(density(dist_sharpnull_ordervalue),
     main="Dist. of order value effect under Sharp Null")
abline(v=ate, col="red")
```



```
mean(abs(dist_sharpnull_ordervalue)>=abs(ate)) # p value
```

```
## [1] 0.2934
```

```
# Controlling for returning customer
m2_ordervalue <- lm(OrderValue~treat+Returning_Customer,
                    data=SO_cont1_treat)
summary(m2_ordervalue)
```

```
##
## Call:
## lm(formula = OrderValue ~ treat + Returning_Customer, data = SO_cont1_treat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.147 -11.587  -3.499   1.501  88.945
##
## Coefficients:
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           10.489      4.945   2.121  0.04050 *
## treat                  6.687      7.053   0.948  0.34907
## Returning_Customer    31.648      9.348   3.386  0.00166 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.41 on 38 degrees of freedom
## Multiple R-squared:  0.2566, Adjusted R-squared:  0.2175
## F-statistic:  6.56 on 2 and 38 DF,  p-value: 0.003571
```

```r
m2_ordervalue$vcov <- vcov(m2_ordervalue)
coeftest(m2_ordervalue, m2_ordervalue$vcov)
```

```
##
## t test of coefficients:
##
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          10.4894     4.9454  2.1210 0.040503 *
## treat                 6.6874     7.0534  0.9481 0.349068
## Returning_Customer   31.6478     9.3476  3.3857 0.001662 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
m3_ordervalue <- lm(OrderValue~treat+Returning_Customer+treat*Returning_Customer,
                    data=SO_cont1_treat)
summary(m3_ordervalue)
```

```
##
## Call:
## lm(formula = OrderValue ~ treat + Returning_Customer + treat *
##     Returning_Customer, data = SO_cont1_treat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.438  -8.681  -3.681   3.367  64.333
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 15.671      4.179   3.750 0.000604 ***
## treat                       -5.058      6.292  -0.804 0.426612
## Returning_Customer          -6.351     11.317  -0.561 0.578047
## treat:Returning_Customer    69.175     15.269   4.530 5.96e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.22 on 37 degrees of freedom
## Multiple R-squared:  0.5219, Adjusted R-squared:  0.4831
## F-statistic: 13.46 on 3 and 37 DF,  p-value: 4.316e-06
```

```r
## Regression Summary Control 1 v.s. Treatment
stargazer(m1_ordervalue, m2_ordervalue, m3_ordervalue,
          type = "latex")
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, May 03, 2017 - 01:50:36

Table 1:

	Dependent variable:		
	OrderValue		
	(1)	(2)	(3)
treat	9.034	6.687	−5.058
	(7.905)	(7.053)	(6.292)
Returning_Customer		31.648***	−6.351
		(9.348)	(11.317)
treat:Returning_Customer			69.175***
			(15.269)
Constant	14.805***	10.489**	15.671***
	(5.381)	(4.945)	(4.179)
Observations	41	41	41
R^2	0.032	0.257	0.522
Adjusted R^2	0.008	0.218	0.483
Residual Std. Error	25.240 (df = 39)	22.412 (df = 38)	18.216 (df = 37)
F Statistic	1.306 (df = 1; 39)	6.560*** (df = 2; 38)	13.461*** (df = 3; 37)
Note:			*p<0.1; **p<0.05; ***p<0.01

It is interesting to note that there seems to be a **strong heterogeneous treatment effect among returning customers**. We see from model m2_ordervalue that returning customer has a strong correlation with the sales order value outcome. When we add an interaction term of returning customer and our treatment, we found that all the treatment effect seems to be shown on the interaction term, meaning treatment has a strong effect on returning customers. The test statistics is highly significant. When looking deeper into the data, we were caucious to found out that the result is being affected by 2 returning customers who used an coupon on top of the discount offered and purchased an very large order with lots of different items. In total across control 1 and treatment week, there're only 7 returning customers and the result of 2 intense buyer resulted in this effect estimate. We would like further studies which focuses more on returning customers and make sure to control for coupon usage in order to confirm for this result. From blank sight, it seems treatment has an much stroger effect on customers returning to the site. But we should be careful in claiming this given our small sample in returning customers.

Treatment v.s. Control 2 Sales Order Value Comparison

```
## average effect on order value of treatment v.s. control 2
SO_treat_cont2 <- SO[SO$Date > "2017-03-19" & SO$Date < "2017-04-02",]
m4_ordervalue <- lm(OrderValue~treat, data=SO_treat_cont2)
summary(m4_ordervalue)
```

```
##
## Call:
## lm(formula = OrderValue ~ treat, data = SO_treat_cont2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.249 -17.249  -5.905   3.644 113.931
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   12.895      8.268   1.559    0.130
## treat         10.945     10.390   1.053    0.301
##
## Residual standard error: 27.42 on 28 degrees of freedom
## Multiple R-squared:  0.03812,    Adjusted R-squared:  0.003769
## F-statistic:  1.11 on 1 and 28 DF,  p-value: 0.3012
```

```
m4_ordervalue$vcov <- vcov(m4_ordervalue)
coeftest(m4_ordervalue, m4_ordervalue$vcov)
```

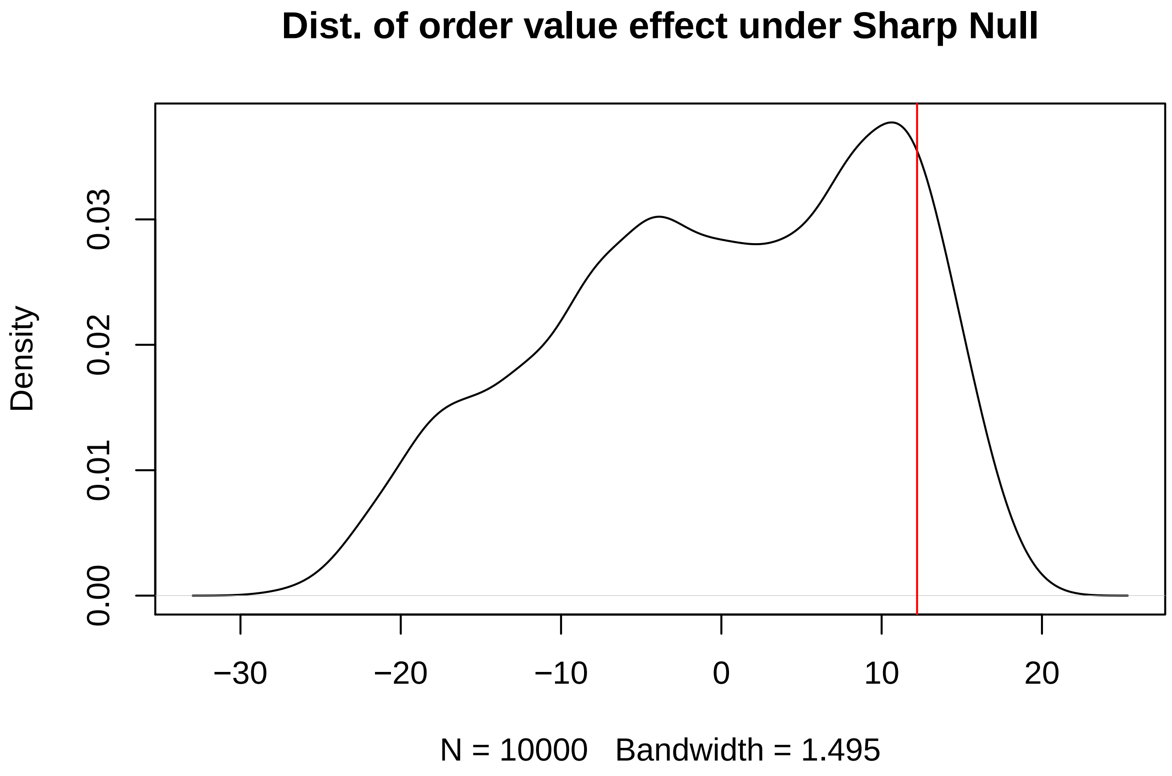
```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.8945     8.2685  1.5595   0.1301
## treat        10.9449    10.3898  1.0534   0.3012
```

```
# non-parametric test
wilcox.test(SO_cont2$OrderValue,SO_treat$OrderValue)
```

```
## Warning in wilcox.test.default(SO_cont2$OrderValue, SO_treat$OrderValue):
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  SO_cont2$OrderValue and SO_treat$OrderValue
## W = 143, p-value = 0.7262
## alternative hypothesis: true location shift is not equal to 0
```

```
# ramdomization inference
ate <- mean(SO_treat$OrderValue) - mean(SO_cont2$OrderValue)
dist_sharpnull_ordervalue <- rep(0,10000)
for (i in 1:10000) {
  treat <- sample(SO_treat_cont2$treat)
  dist_sharpnull_ordervalue[i] <- mean(SO_treat_cont2[treat==1,]$OrderValue) -
    mean(SO_treat_cont2[treat==0,]$OrderValue)
}
plot(density(dist_sharpnull_ordervalue),
     main="Dist. of order value effect under Sharp Null")
abline(v=ate, col="red")
```



```
mean(abs(dist_sharpnull_ordervalue)>=abs(ate)) # p value
```

```
## [1] 0.2807
```

```
# Controlling for returning customer
m5_ordervalue <- lm(OrderValue~treat+Returning_Customer,
                    data=SO_treat_cont2)
summary(m5_ordervalue)
```

```
##
## Call:
## lm(formula = OrderValue ~ treat + Returning_Customer, data = SO_treat_cont2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.438  -5.023  -2.905   4.867  64.332
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          12.895      5.379   2.397   0.0237 *
## treat                -2.281      7.082  -0.322   0.7498
## Returning_Customer   62.824     10.039   6.258 1.08e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.84 on 27 degrees of freedom
## Multiple R-squared:  0.6075, Adjusted R-squared:  0.5784
## F-statistic: 20.89 on 2 and 27 DF,  p-value: 3.292e-06
```

```
m5_ordervalue$vcov <- vcov(m5_ordervalue)
coeftest(m5_ordervalue, m5_ordervalue$vcov)
```

```
##
## t test of coefficients:
##
##                     Estimate Std. Error t value  Pr(>|t|)
## (Intercept)          12.8945     5.3790  2.3972   0.02371 *
## treat                -2.2812     7.0818 -0.3221   0.74984
## Returning_Customer   62.8242    10.0392  6.2579 1.075e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
m6_ordervalue <- lm(OrderValue~treat+Returning_Customer+treat*Returning_Customer,
                    data=SO_treat_cont2)
summary(m6_ordervalue)
```

```
##
## Call:
## lm(formula = OrderValue ~ treat + Returning_Customer + treat *
##     Returning_Customer, data = SO_treat_cont2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.438  -5.023  -2.905   4.867  64.332
##
## Coefficients: (1 not defined because of singularities)
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 12.895      5.379   2.397   0.0237 *
## treat                       -2.281      7.082  -0.322   0.7498
## Returning_Customer          62.824     10.039   6.258 1.08e-06 ***
## treat:Returning_Customer        NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.84 on 27 degrees of freedom
## Multiple R-squared:  0.6075, Adjusted R-squared:  0.5784
## F-statistic: 20.89 on 2 and 27 DF,  p-value: 3.292e-06
```

```
## Regression Summary Control 1 v.s. Treatment
stargazer(m4_ordervalue, m5_ordervalue, m6_ordervalue,
          type = "latex")
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, May 03, 2017 - 01:50:40

Control 1 v.s. Control 2 Sales Order Value Comparison

```
## average effect on order value of control 1 v.s. control 2
SO_cont1_cont2 <- SO[(SO$Date > "2017-03-12" & SO$Date < "2017-03-20") |
     (SO$Date > "2017-03-26" & SO$Date < "2017-04-02"),]
t.test(SO_cont2$OrderValue,SO_cont1$OrderValue)
```

```
##
##  Welch Two Sample t-test
##
## data:  SO_cont2$OrderValue and SO_cont1$OrderValue
## t = -0.90397, df = 31.686, p-value = 0.3728
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

Table 2:

	Dependent variable:		
	OrderValue		
	(1)	(2)	(3)
treat	10.945	−2.281	−2.281
	(10.390)	(7.082)	(7.082)
Returning_Customer		62.824***	62.824***
		(10.039)	(10.039)
treat:Returning_Customer			
Constant	12.895	12.895**	12.895**
	(8.268)	(5.379)	(5.379)
Observations	30	30	30
R^2	0.038	0.607	0.607
Adjusted R^2	0.004	0.578	0.578
Residual Std. Error	27.423 (df = 28)	17.840 (df = 27)	17.840 (df = 27)
F Statistic	1.110 (df = 1; 28)	20.891*** (df = 2; 27)	20.891*** (df = 2; 27)
Note:			*p<0.1; **p<0.05; ***p<0.01

```
##  -10.334373   3.982944
## sample estimates:
## mean of x mean of y
##  11.62929  14.80500
```

```
wilcox.test(SO_cont2$OrderValue,SO_cont1$OrderValue)
```

```
## Warning in wilcox.test.default(SO_cont2$OrderValue, SO_cont1$OrderValue):
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  SO_cont2$OrderValue and SO_cont1$OrderValue
## W = 146.5, p-value = 0.8098
## alternative hypothesis: true location shift is not equal to 0
```

```
m7_ordervalue <- lm(OrderValue~Returning_Customer, data=SO_cont1_cont2)
summary(m7_ordervalue)
```

```
##
## Call:
## lm(formula = OrderValue ~ Returning_Customer, data = SO_cont1_cont2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.663 -7.663 -4.663  4.327 54.267
##
## Coefficients:
```

```
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           14.653      2.248   6.517 2.84e-07 ***
## Returning_Customer    -5.333      7.457  -0.715     0.48    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.31 on 31 degrees of freedom
## Multiple R-squared:  0.01623,    Adjusted R-squared:  -0.0155 
## F-statistic: 0.5115 on 1 and 31 DF,  p-value: 0.4799
```

```
m7_ordervalue$vcov <- vcov(m7_ordervalue)
coeftest(m7_ordervalue, m7_ordervalue$vcov)
```

```
##
## t test of coefficients:
##
##                     Estimate Std. Error t value  Pr(>|t|)    
## (Intercept)          14.6530     2.2484  6.5171 2.843e-07 ***
## Returning_Customer   -5.3330     7.4571 -0.7152    0.4799    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# non-parametric test
wilcox.test(SO_cont2$OrderValue,SO_treat$OrderValue)
```

```
## Warning in wilcox.test.default(SO_cont2$OrderValue, SO_treat$OrderValue):
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  SO_cont2$OrderValue and SO_treat$OrderValue
## W = 143, p-value = 0.7262
## alternative hypothesis: true location shift is not equal to 0
```

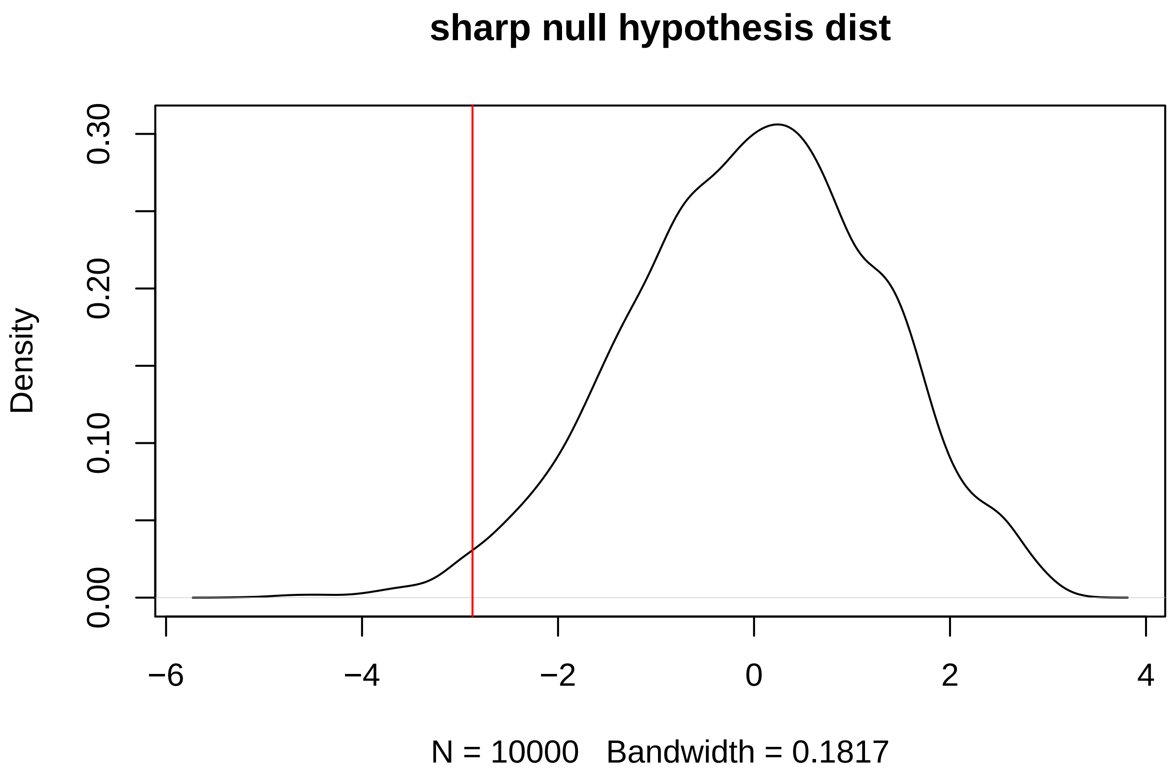
3.2 Price per Unique Item per Order

```
## Control 1 v.s. Treatment
SOI_price <- data.frame(
   price=c(SOI_treat[,"Qty"]*SOI_treat[,"ItemPrice"],
           SOI_cont1[,"Qty"]*SOI_cont1[,"ItemPrice"]),
   treat=c(rep(1,length(SOI_treat[,"Qty"])),
           rep(0,length(SOI_cont1[,"Qty"]))),
   orderid=c(SOI_treat[,"OrderID"],SOI_cont1[,"OrderID"])
    )
t.test(SOI_price[SOI_price$treat==1,]$price,
       SOI_price[SOI_price$treat==0,]$price)
```

```
##
##  Welch Two Sample t-test
##
## data:  SOI_price[SOI_price$treat == 1, ]$price and SOI_price[SOI_price$treat == 0, ]$price
## t = -2.1119, df = 50.443, p-value = 0.03967
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -5.6044098 -0.1411749
## sample estimates:
## mean of x mean of y
##  7.305645 10.178437
```

```
treat <- c(rep(1,length(SOI_treat[,"Qty"])),
           rep(0,length(SOI_cont1[,"Qty"])))
ate <- mean(SOI_price[SOI_price$treat==1,]$price)-
        mean(SOI_price[SOI_price$treat==0,]$price)
dist <- rep(0,10000)
for (i in 1:10000){
  treatment <- sample(treat)
  dist[i] <- mean(SOI_price[treatment==1,]$price)-
          mean(SOI_price[treatment==0,]$price)
}
plot(density(dist), main="sharp null hypothesis dist")
abline(v=ate,col="red")
```



```
pval <- mean(abs(ate)<=abs(dist))
sum(abs(dist)>=abs(ate))/length(dist)
```

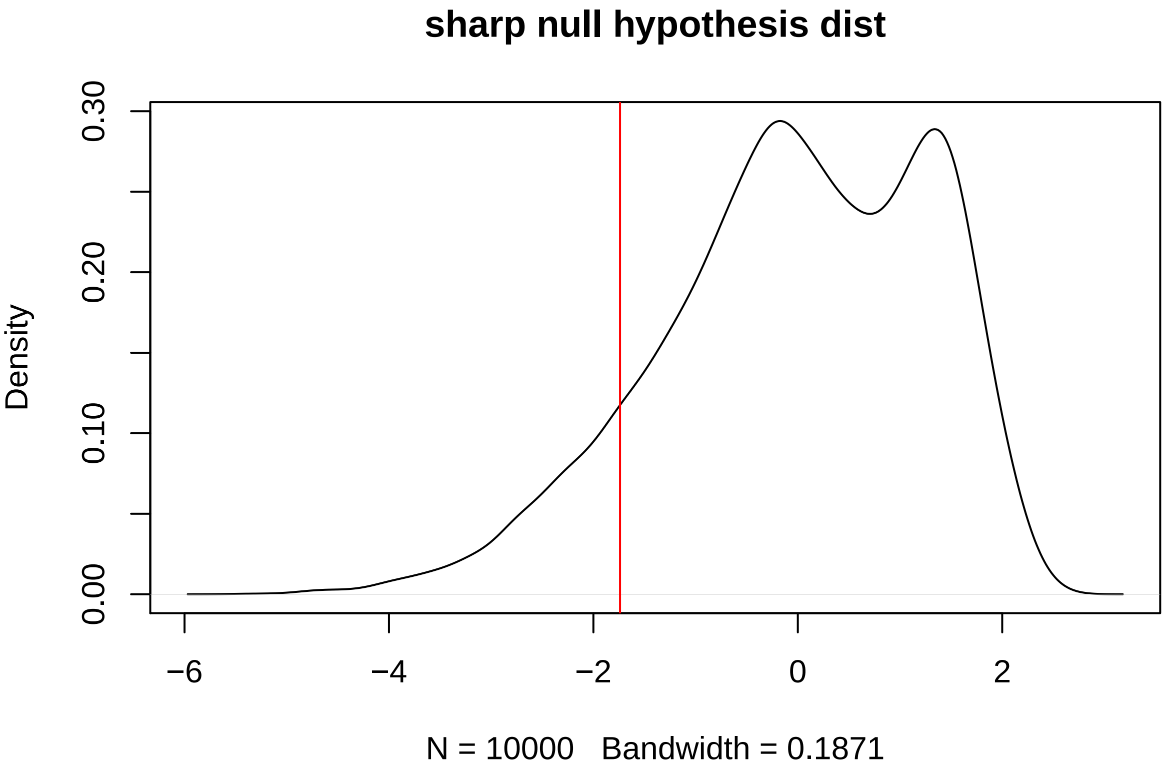
```
## [1] 0.0188
```

```
##==========================================================================

## Control 2 v.s. Treatment
SOI_price <- data.frame(
   price=c(SOI_treat[,"Qty"]*SOI_treat[,"ItemPrice"],
           SOI_cont2[,"Qty"]*SOI_cont2[,"ItemPrice"]),
   treat=c(rep(1,length(SOI_treat[,"Qty"])),
           rep(0,length(SOI_cont2[,"Qty"])))
```

```
 )
t.test(SOI_price[SOI_price$treat==1,]$price,
       SOI_price[SOI_price$treat==0,]$price)
```

```
##
##  Welch Two Sample t-test
##
## data:  SOI_price[SOI_price$treat == 1, ]$price and SOI_price[SOI_price$treat == 0, ]$price
## t = -1.6615, df = 41.526, p-value = 0.1041
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.8527279  0.3740182
## sample estimates:
## mean of x mean of y
##  7.305645  9.045000
```

```
treat <- c(rep(1,length(SOI_treat[,"Qty"])),
           rep(0,length(SOI_cont2[,"Qty"])))
ate <- mean(SOI_price[SOI_price$treat==1,]$price)-
        mean(SOI_price[SOI_price$treat==0,]$price)
dist <- rep(0,10000)
for (i in 1:10000){
  treatment <- sample(treat)
  dist[i] <- mean(SOI_price[treatment==1,]$price)-
          mean(SOI_price[treatment==0,]$price)
}
plot(density(dist), main="sharp null hypothesis dist")
abline(v=ate,col="red")
```



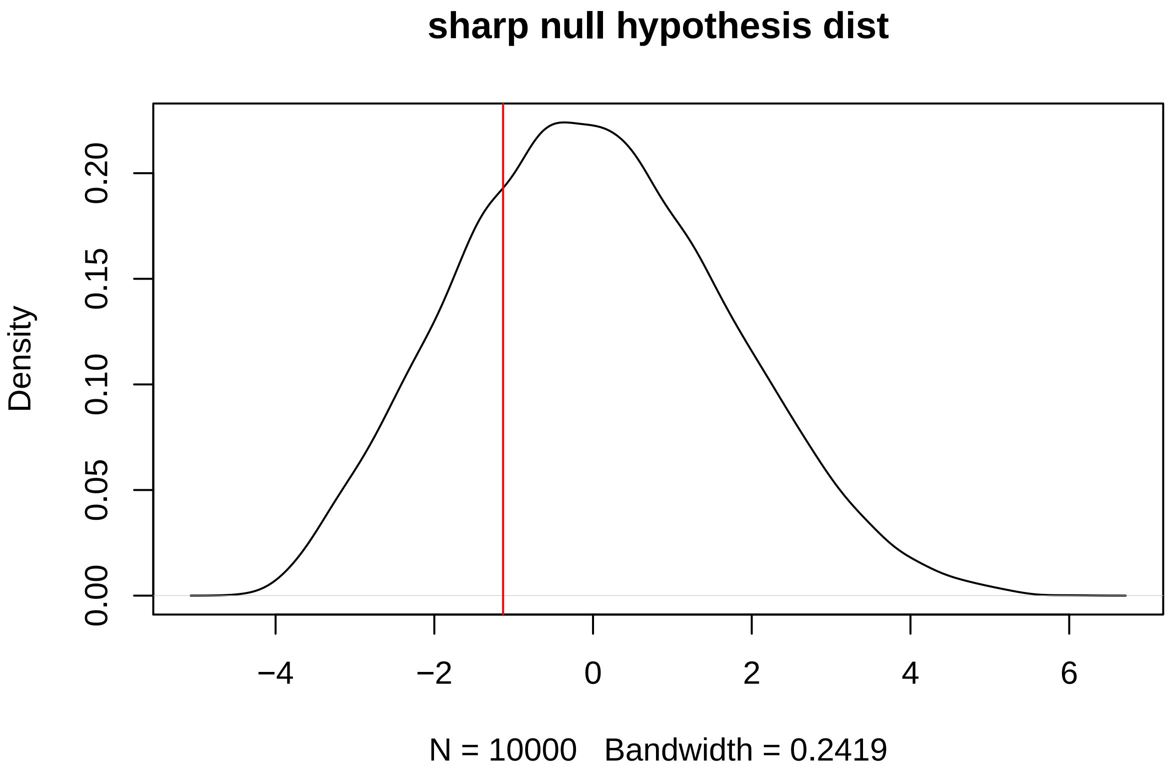
```
pval <- mean(abs(ate)<=abs(dist))
sum(abs(dist)>=abs(ate))/length(dist)
```

```
## [1] 0.1728
```

```
##=================================================================================

## Control 2 v.s. Control 1
SOI_price <- data.frame(
   price=c(SOI_cont2[,"Qty"]*SOI_cont2[,"ItemPrice"],
           SOI_cont1[,"Qty"]*SOI_cont1[,"ItemPrice"]),
   treat=c(rep(1,length(SOI_cont2[,"Qty"])),
           rep(0,length(SOI_cont1[,"Qty"])))
 )
t.test(SOI_price[SOI_price$treat==1,]$price,
       SOI_price[SOI_price$treat==0,]$price)
```

```
##
##  Welch Two Sample t-test
##
## data:  SOI_price[SOI_price$treat == 1, ]$price and SOI_price[SOI_price$treat == 0, ]$price
## t = -0.78585, df = 47.737, p-value = 0.4358
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.033794  1.766919
## sample estimates:
## mean of x mean of y
##   9.04500  10.17844
```

```
treat <- c(rep(1,length(SOI_cont2[,"Qty"])),
           rep(0,length(SOI_cont1[,"Qty"])))
ate <- mean(SOI_price[SOI_price$treat==1,]$price)-
        mean(SOI_price[SOI_price$treat==0,]$price)
dist <- rep(0,10000)
for (i in 1:10000){
  treatment <- sample(treat)
  dist[i] <- mean(SOI_price[treatment==1,]$price)-
          mean(SOI_price[treatment==0,]$price)
}
plot(density(dist), main="sharp null hypothesis dist")
abline(v=ate,col="red")
```



```
pval <- mean(abs(ate)<=abs(dist))
sum(abs(dist)>=abs(ate))/length(dist)
```

```
## [1] 0.5109
```

The following section performs the same test as Sales Order Value comparison, but using SOI data and aggregate by the order ID. This is just a double check to make sure both approaches yields same result.

```
SOI_price <- data.frame(
   price=c(SOI_treat[,"Qty"]*SOI_treat[,"ItemPrice"], SOI_cont1[,"Qty"]*SOI_cont1[,"ItemPrice"]),
   treat=c(rep(1,length(SOI_treat[,"Qty"])),rep(0,length(SOI_cont1[,"Qty"]))),
   orderid=c(SOI_treat[,"OrderID"],SOI_cont1[,"OrderID"])
    )
SO_Value <- aggregate(x=SOI_price$price, by=list(SOI_price$orderid), FUN=sum)
treat <- aggregate(x=SOI_price$treat, by=list(SOI_price$orderid), FUN=mean)
SO_Value <- cbind(SO_Value, treat)
colnames(SO_Value) <- c("Group1", "OrderValue", "Group2" ,"treat")
SO_Value$valid <- SO_Value$Group1 == SO_Value$Group2
SO_Value[SO_Value$valid != TRUE,]
```

```
## [1] Group1     OrderValue Group2     treat      valid
## <0 rows> (or 0-length row.names)
```

```
t.test(SO_Value[SO_Value$treat==1,]$OrderValue,SO_Value[SO_Value$treat==0,]$OrderValue)
```

```
##
##  Welch Two Sample t-test
##
## data:  SO_Value[SO_Value$treat == 1, ]$OrderValue and SO_Value[SO_Value$treat == 0, ]$OrderValue
## t = 1.0844, df = 23.481, p-value = 0.2892
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

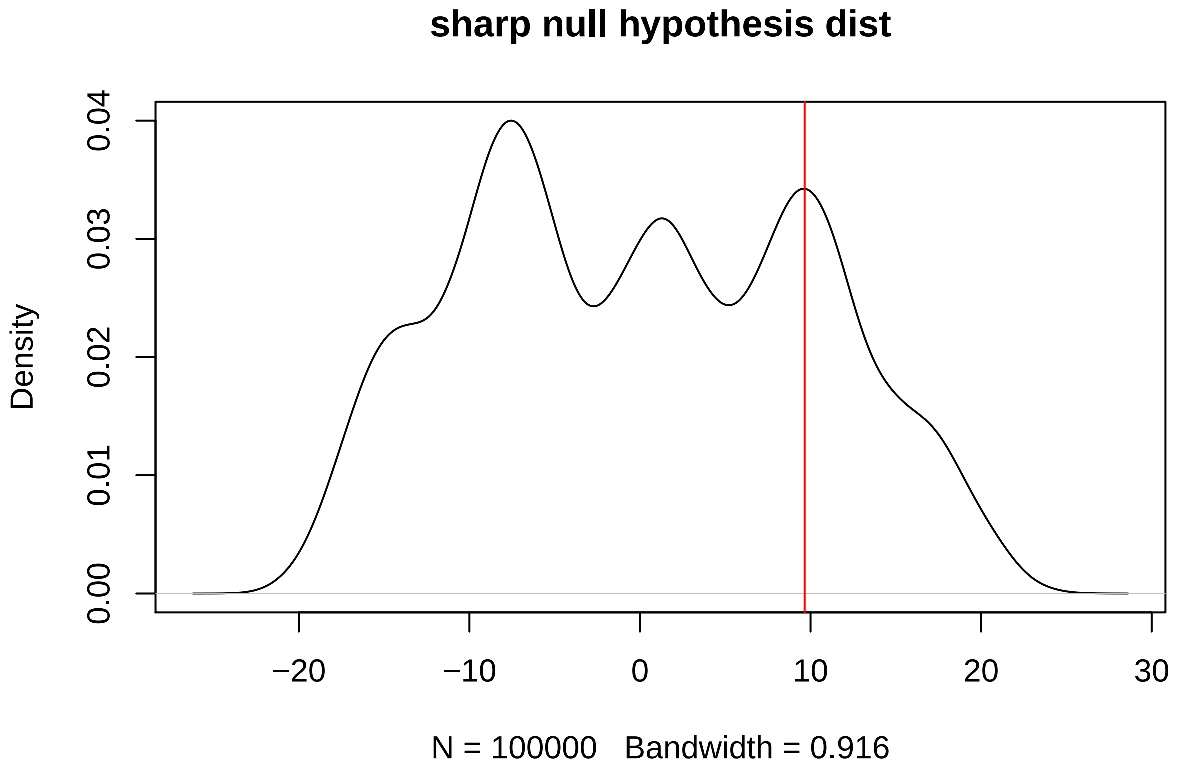
```
##  -8.18003 26.24898
## sample estimates:
## mean of x mean of y
##  23.83947  14.80500
```

3.3 Sales value per Buyer

```
SVB_treat <- aggregate(x=SO_treat$OrderValue,by=list(SO_treat$BuyerID), FUN=sum)
SVB_cont1 <- aggregate(x=SO_cont1$OrderValue,by=list(SO_cont1$BuyerID), FUN=sum)
SVB <- rbind(SVB_treat,SVB_cont1)
t.test(SVB_treat$x,SVB_cont1$x)
```

```
##
##  Welch Two Sample t-test
##
## data:  SVB_treat$x and SVB_cont1$x
## t = 0.89244, df = 21.197, p-value = 0.3822
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -12.82942  32.13720
## sample estimates:
## mean of x mean of y
##  25.16389  15.51000
```

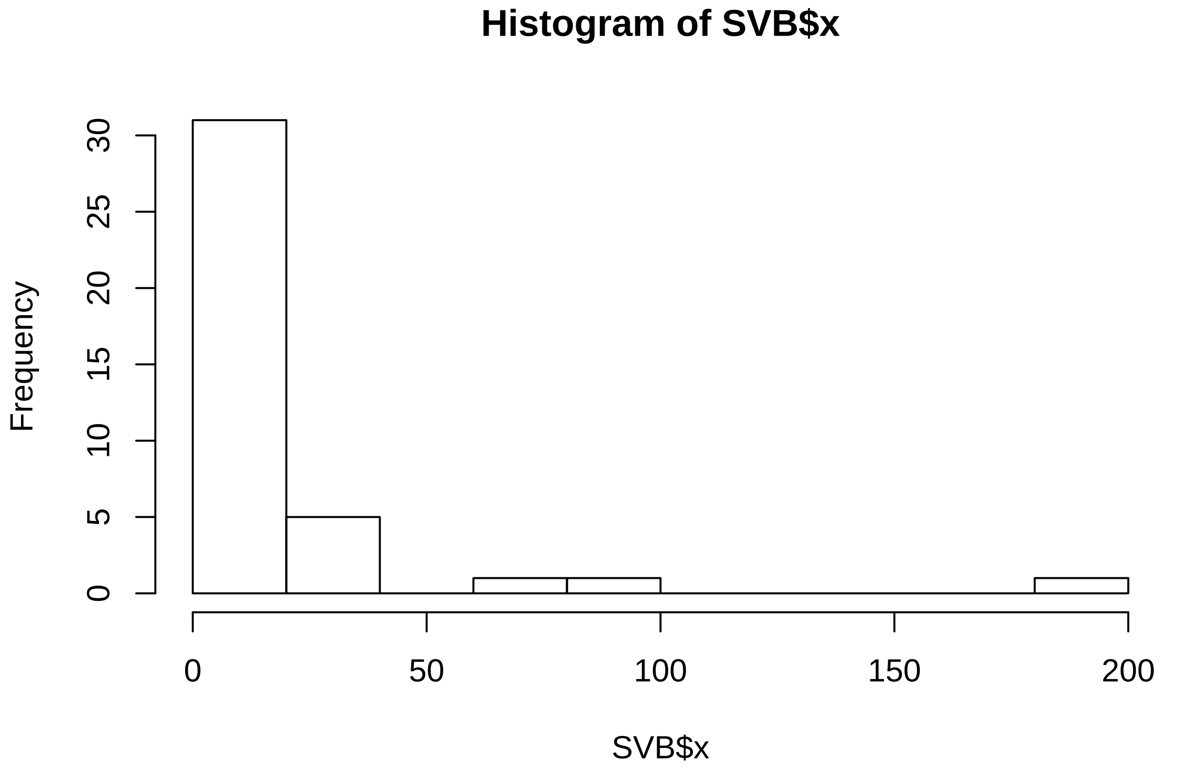
```
ate <- mean(SVB_treat$x) - mean(SVB_cont1$x)
treat <- c(rep(1,length(SVB_treat$x)),rep(0,length(SVB_cont1$x)))
dist <- rep(0,100000)
for (i in 1:100000){
  treatment <- sample(treat)
  dist[i] <- mean(SVB$x[treatment==1])-mean(SVB$x[treatment==0])
}
plot(density(dist), main="sharp null hypothesis dist")
abline(v=ate, col="red")
```



```
sum(abs(dist)>=abs(ate))/length(dist)
```

```
## [1] 0.41258
```

```
hist(SVB$x)
```

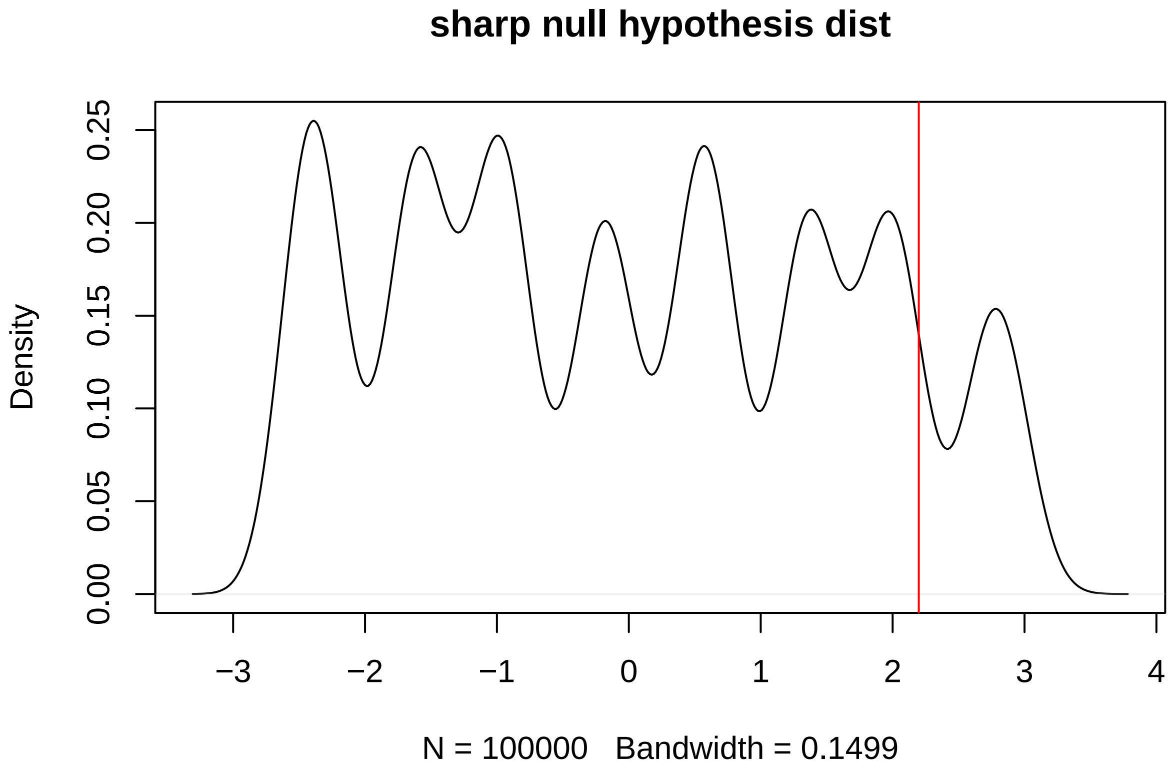


3.4 Sales Quantity per Buyer

```
SQB_treat <- aggregate(x=SO_treat$NrItems,by=list(SO_treat$BuyerID), FUN=sum)
SQB_cont1 <- aggregate(x=SO_cont1$NrItems,by=list(SO_cont1$BuyerID), FUN=sum)
SQB <- rbind(SQB_treat,SQB_cont1)
t.test(SQB_treat$x,SQB_cont1$x)
```

```
##
##  Welch Two Sample t-test
##
## data:  SQB_treat$x and SQB_cont1$x
## t = 1.2423, df = 18.663, p-value = 0.2295
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.509920  5.906745
## sample estimates:
## mean of x mean of y
##  3.722222  1.523810
```

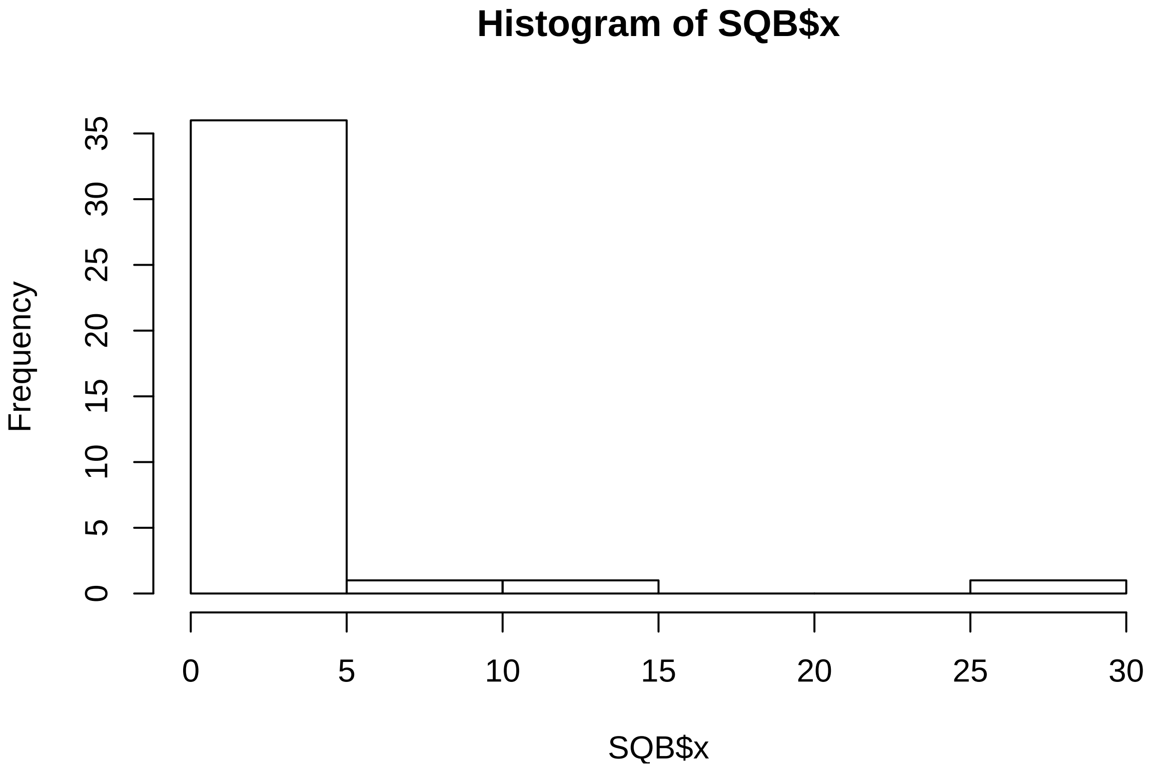
```
ate <- mean(SQB_treat$x) - mean(SQB_cont1$x)
treat <- c(rep(1,length(SQB_treat$x)),rep(0,length(SQB_cont1$x)))
dist <- rep(0,100000)
for (i in 1:100000){
  treatment <- sample(treat)
  dist[i] <- mean(SQB$x[treatment==1])-mean(SQB$x[treatment==0])
}
plot(density(dist), main="sharp null hypothesis dist")
abline(v=ate, col="red")
```



```
sum(abs(dist)>=abs(ate))/length(dist)
```

```
## [1] 0.24067
```

```
hist(SQB$x)
```



3.5 Conversion Rate

```
nrsession_cont1 <- sum(session_control1$Sessions)
nrorder_cont1 <- length(SO_cont1$OrderID)
nrsession_treat <- sum(session_treatment$Sessions)
nrorder_treat <- length(SO_treat$OrderID)
nrsession_cont2 <- sum(session_control2$Sessions)
nrorder_cont2 <- length(SO_cont2$OrderID)
conversion_cont1 <- c(rep(0, nrsession_cont1-nrorder_cont1) ,
                      rep(1,nrorder_cont1) )
conversion_treat <- c(rep(0, nrsession_treat-nrorder_treat) ,
                      rep(1,nrorder_treat) )
conversion_cont2 <- c(rep(0, nrsession_cont2-nrorder_cont2) ,
                      rep(1,nrorder_cont2) )

t.test(conversion_cont1,conversion_treat)
```

```
##
##  Welch Two Sample t-test
##
## data:  conversion_cont1 and conversion_treat
## t = 0.68032, df = 2332, p-value = 0.4964
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.006803522  0.014031991
## sample estimates:
##  mean of x  mean of y
## 0.01894918 0.01533495
```

```
wilcox.test(conversion_cont1,conversion_treat)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conversion_cont1 and conversion_treat
## W = 721840, p-value = 0.4949
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(conversion_cont2,conversion_treat)
```

```
##
##  Welch Two Sample t-test
##
## data:  conversion_cont2 and conversion_treat
## t = -0.61119, df = 2365.7, p-value = 0.5411
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.012350059  0.006480873
## sample estimates:
##  mean of x  mean of y
## 0.01240035 0.01533495
```

```
wilcox.test(conversion_cont2,conversion_treat)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conversion_cont2 and conversion_treat
## W = 697360, p-value = 0.5431
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(conversion_cont1,conversion_cont2)
```

```
##
##  Welch Two Sample t-test
##
## data:  conversion_cont1 and conversion_cont2
## t = 1.2631, df = 2217.5, p-value = 0.2067
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.003618877  0.016716532
## sample estimates:
##  mean of x  mean of y
## 0.01894918 0.01240035
```

```
wilcox.test(conversion_cont1,conversion_cont2)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  conversion_cont1 and conversion_cont2
## W = 659680, p-value = 0.208
## alternative hypothesis: true location shift is not equal to 0
```

3.6 Overall Revenue Compared to Past Half Year

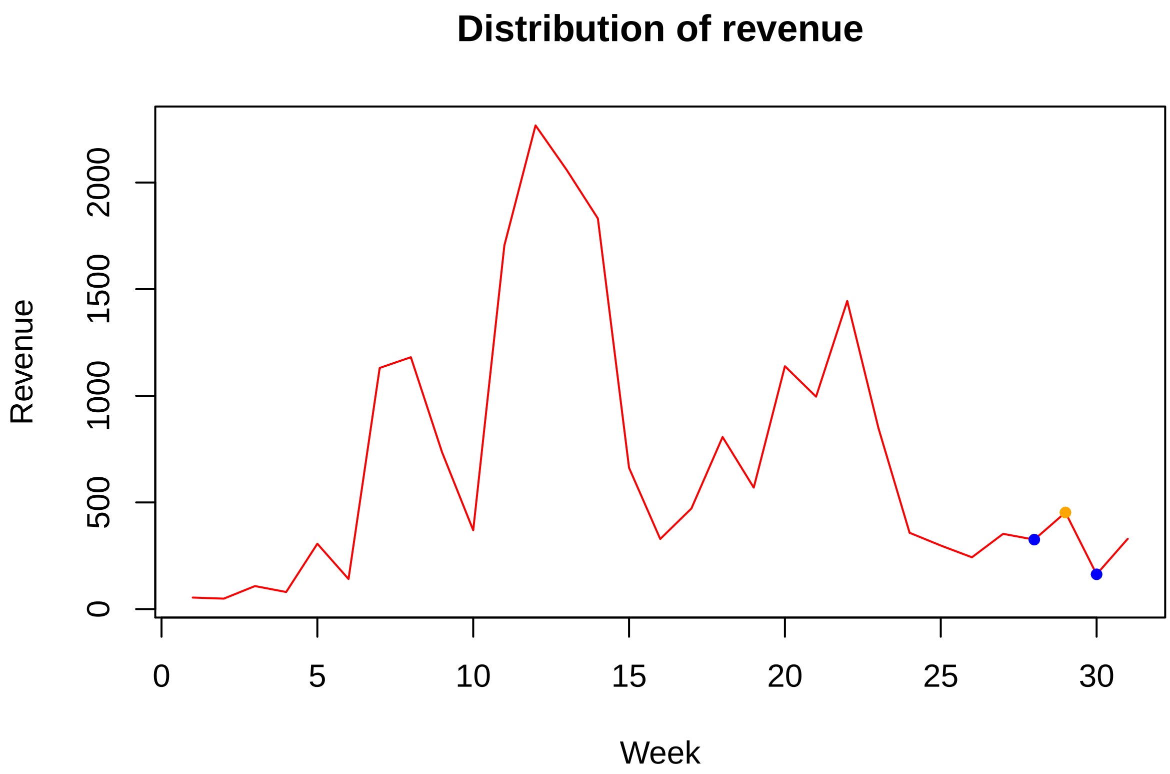
```
max(SO$Date)-min(SO$Date)
```

```
## Time difference of 218 days
```

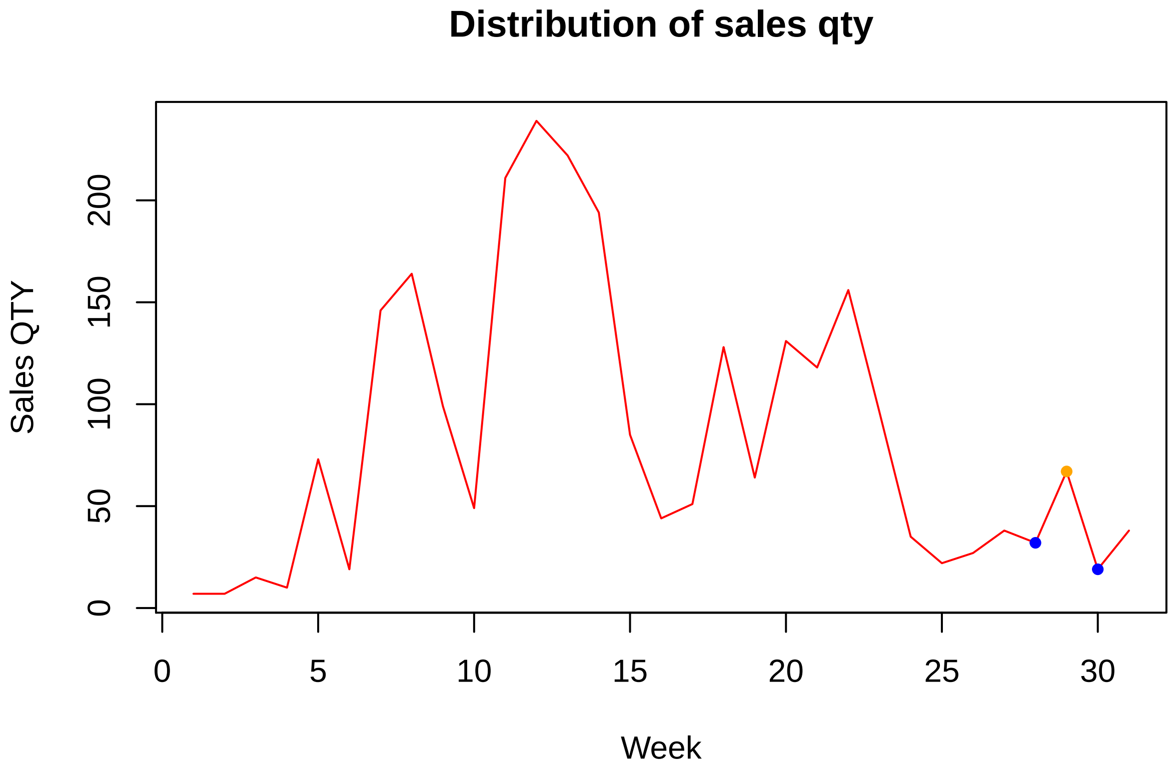
```
218/7
```

```
## [1] 31.14286
```

```
# 31 weeks sales data available
week_sales <- rep(0,31)
week_qty <- rep(0,31)
for (i in 1:31){
        start <- max(SO$Date)-7*i
        end <- max(SO$Date)+8-7*i
        week_sales[i] <- sum(SO[SO$Date>start & SO$Date<end,]$OrderValue)
        week_qty[i] <- sum(SO[SO$Date>start & SO$Date<end,]$NrItems)
}

## Plot results
plot(c(31:1),week_sales,
     type = "l",
     main = "Distribution of revenue",
     xlab= "Week", ylab = "Revenue",
     ylim = c(min(week_sales),
              max(week_sales)),
     col = "red")
points(29,week_sales[3], pch = 20, col = "orange")
points(c(30,28),week_sales[c(2,4)], pch = 20, col = "blue")
```



```
plot(c(31:1),week_qty,
     type = "l",
     main = "Distribution of sales qty",
     xlab= "Week", ylab = "Sales QTY",
     ylim = c(min(week_qty),
              max(week_qty)),
     col = "red")
points(29,week_qty[3], pch = 20, col = "orange")
points(c(30,28),week_qty[c(2,4)], pch = 20, col = "blue")
```



Summary

- Average Order Value is NOT significant across treatment and control weeks.
- Sales Value per Buyer and Sales QTY per Buyer are both NOT significant.
- Sales per Unique Item per Order IS significant. The lower value in treatment week marks that customers tends to buy lower value items during the treatment is offered.
- Conversion Rate is NOT significant. The average across the weeks are similar, marking the no effect of treatment on the likelihood of transforming web sessions into purchases.
- Regression shows a Heterogeneous Treatment Effect among returning customers comparing control 1 to treatment week. Although test is highly significant, the results were mostly driven by small number of returning customers. This effect needs further study focusing on gathering more returning customer subject base to confirm.
- Overall sales and qty sold shows higher in treatment week compared to pre-treatment and post-treatment weeks. The obvious drop in control 2 week indicates a intertemporal substitution effect after treatment is ended. Comparing to revenue during the past 6 months, treatment week is lower than average. However, there seems to be a sales fluctuation pattern across the months, so we cannot just compare treatment week against past 6 months to draw causal estimate. More trustworthy, comparing treatment week to the past month span shows that treatment week is truely higher while controlling for the monthly fluctuations.