Solution: Pricing Test - A/B Test

Ismat Ara

Description

Company XYZ sells a software for $39. Since revenue has been flat for some time, the VP of Product has decided to run a test increasing the price. She hopes that this would increase revenue. In the experiment, 66% of the users have seen the old price ($39), while a random sample of 33% users a higher price ($59).

The test has been running for some time and the VP of Product is interested in understanding how it went and whether it would make sense to increase the price for all the users.

Especially he asked you the following questions:

- Should the company sell its software for $39 or $59?
- The VP of Product is interested in having a holistic view into user behavior, especially focusing on actionable insights that might increase conversion rate. What are your main findings looking at the data?
- [Bonus] The VP of Product feels that the test has been running for too long and he should have been able to get statistically significant results in a shorter time. Do you agree with her intuition? After how many days you would have stopped the test? Please, explain why.

Check A/B Test Results

First importing the datasets:

```
user <- read.csv('user_table.csv')
test <- read.csv('test_results.csv')
```

The user dataset has 275,616 observations of 5 variables and test dataset has 316,800 observations of 8 variables.

Let's check the datasets first:

```
head(user)
```

```
##   user_id         city country   lat    long
## 1  510335      Peabody     USA 42.53  -70.97
## 2   89568         Reno     USA 39.54 -119.82
## 3  434134       Rialto     USA 34.11 -117.39
## 4  289769 Carson City     USA 39.15 -119.74
## 5  939586      Chicago     USA 41.84  -87.68
## 6  229234     New York     USA 40.67  -73.94
```

```
head(test)
```

```
##   user_id           timestamp          source device operative_system test
## 1  604839 2015-05-08 03:38:34   ads_facebook mobile              iOS    0
## 2  624057 2015-05-10 21:08:46     seo-google mobile          android    0
## 3  317970 2015-04-04 15:01:23       ads-bing mobile          android    0
## 4  685636 2015-05-07 07:26:01 direct_traffic mobile              iOS    1
## 5  820854 2015-05-24 11:04:40   ads_facebook    web              mac    0
## 6  169971 2015-04-13 12:07:08     ads-google mobile              iOS    0
##   price converted
## 1    39         0
## 2    39         0
## 3    39         0
## 4    59         0
## 5    39         0
## 6    39         0
```

Check if the users are unique in both datasets:

```
length(unique(user$user_id)) == length(user$user_id)
```

```
## [1] TRUE
```

```
length(unique(test$user_id)) == length(test$user_id)
```

```
## [1] TRUE
```

```
length(user$user_id)-length(test$user_id)
```

```
## [1] -41184
```

Looks like user table is splitted and 41,184 user ids are missing.

let’s combine the tables:

```
data <- merge(user,test, by='user_id', all=TRUE)
```

Check the structure of the combined data:

```
str(data)
```

```
## 'data.frame':    316800 obs. of  12 variables:
##  $ user_id         : int  3 9 14 16 19 22 23 24 27 30 ...
##  $ city            : Factor w/ 923 levels "Abilene","Akron",..: 587 100 909 23 2 8
57 112 655 NA 393 ...
##  $ country         : Factor w/ 1 level "USA": 1 1 1 1 1 1 1 1 NA 1 ...
##  $ lat             : num  38.9 41.7 39.7 38 41.1 ...
##  $ long            : num  -94.8 -72.9 -75.5 -121.8 -81.5 ...
##  $ timestamp       : Factor w/ 140931 levels "2015-03-02 00:04:12",..: 71600 90540
41454 125216 44891 127025 98925 78499 31249 100803 ...
##  $ source          : Factor w/ 12 levels "ads_facebook",..: 8 10 7 4 4 3 4 6 4 4 .
..
##  $ device          : Factor w/ 2 levels "mobile","web": 2 1 1 1 1 2 2 1 1 2 ...
##  $ operative_system: Factor w/ 6 levels "android","iOS",..: 4 1 2 1 1 6 6 1 2 6 ..
.
##  $ test            : int  1 0 0 0 0 0 1 0 0 1 ...
##  $ price           : int  59 39 39 39 39 39 59 39 39 59 ...
##  $ converted       : int  0 0 0 0 0 0 0 0 0 0 ...
```

Converting the ‘timestamp’ as Date and ‘test’ and ‘price’ as factor.

```
data$timestamp <- as.Date(data$timestamp)
data$test <- as.factor(data$test)
data$price <- as.factor(data$price)
```

Check the summary of the data:

```
summary(data)
```

```
##     user_id                    city          country            lat
##  Min.   :      3   New York    : 25748   USA :275616   Min.   :19.70
##  1st Qu.: 249526   Chicago     :  7153   NA's: 41184   1st Qu.:33.66
##  Median : 499022   Houston     :  6706                 Median :37.74
##  Mean   : 499281   San Antonio:  4633                  Mean   :37.11
##  3rd Qu.: 749026   Los Angeles:  4141                  3rd Qu.:40.70
##  Max.   :1000000   (Other)     :227235                 Max.   :61.18
##                    NA's        : 41184                 NA's   :41184
##       long              timestamp                      source
##  Min.   :-157.80   Min.   :2015-03-02   direct_traffic:60357
##  1st Qu.:-112.20   1st Qu.:2015-03-26   ads-google    :59379
##  Median : -88.93   Median :2015-04-17   ads_facebook  :53396
##  Mean   : -93.98   Mean   :2015-04-16   ads_other     :29876
##  3rd Qu.: -78.91   3rd Qu.:2015-05-09   seo-google    :23175
##  Max.   :  30.31   Max.   :2015-05-31   ads-bing      :22873
##  NA's   :41184                          (Other)       :67744
##     device        operative_system test        price         converted
##  mobile:186471   android: 74935   0:202727   39:202672   Min.   :0.00000
##  web   :130329   iOS    : 95465   1:114073   59:114128   1st Qu.:0.00000
##                  linux  :  4135                          Median :0.00000
##                  mac    : 25085                          Mean   :0.01833
##                  other  : 16204                          3rd Qu.:0.00000
##                  windows:100976                          Max.   :1.00000
##
```

Check the missing data:

```
colSums(is.na(data))
```

```
##          user_id             city          country              lat
##                0            41184            41184            41184
##             long        timestamp           source           device
##            41184                0                0                0
## operative_system             test            price        converted
##                0                0                0                0
```

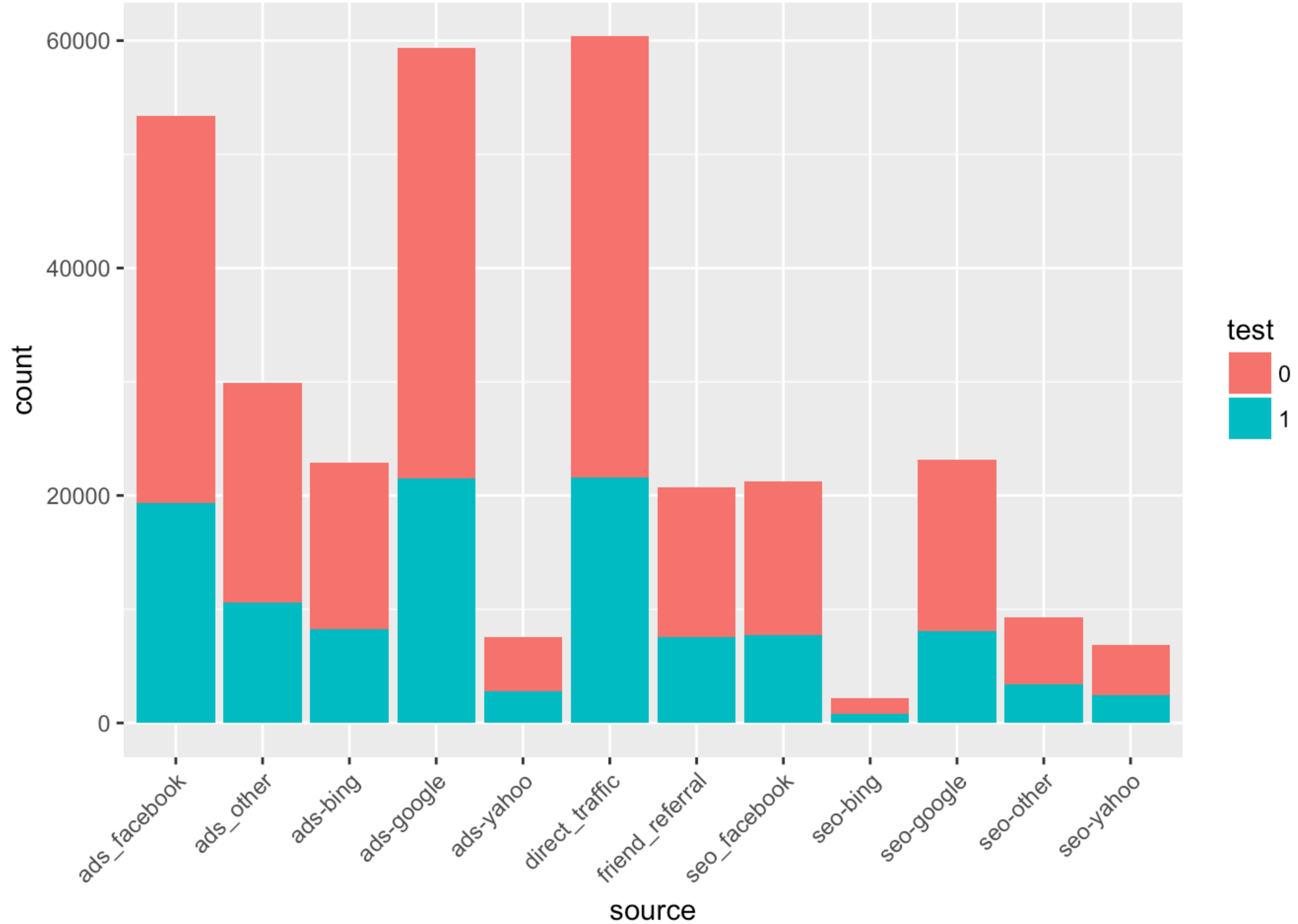
41,184 missing data for city,country,lat and long and it happens for missing user ids. I prefer not to manipulate that huge data and will leave as it is.

Data Visualization

Let’s do some visualization on how the users were distributed over various segments.

Users distribution in source:

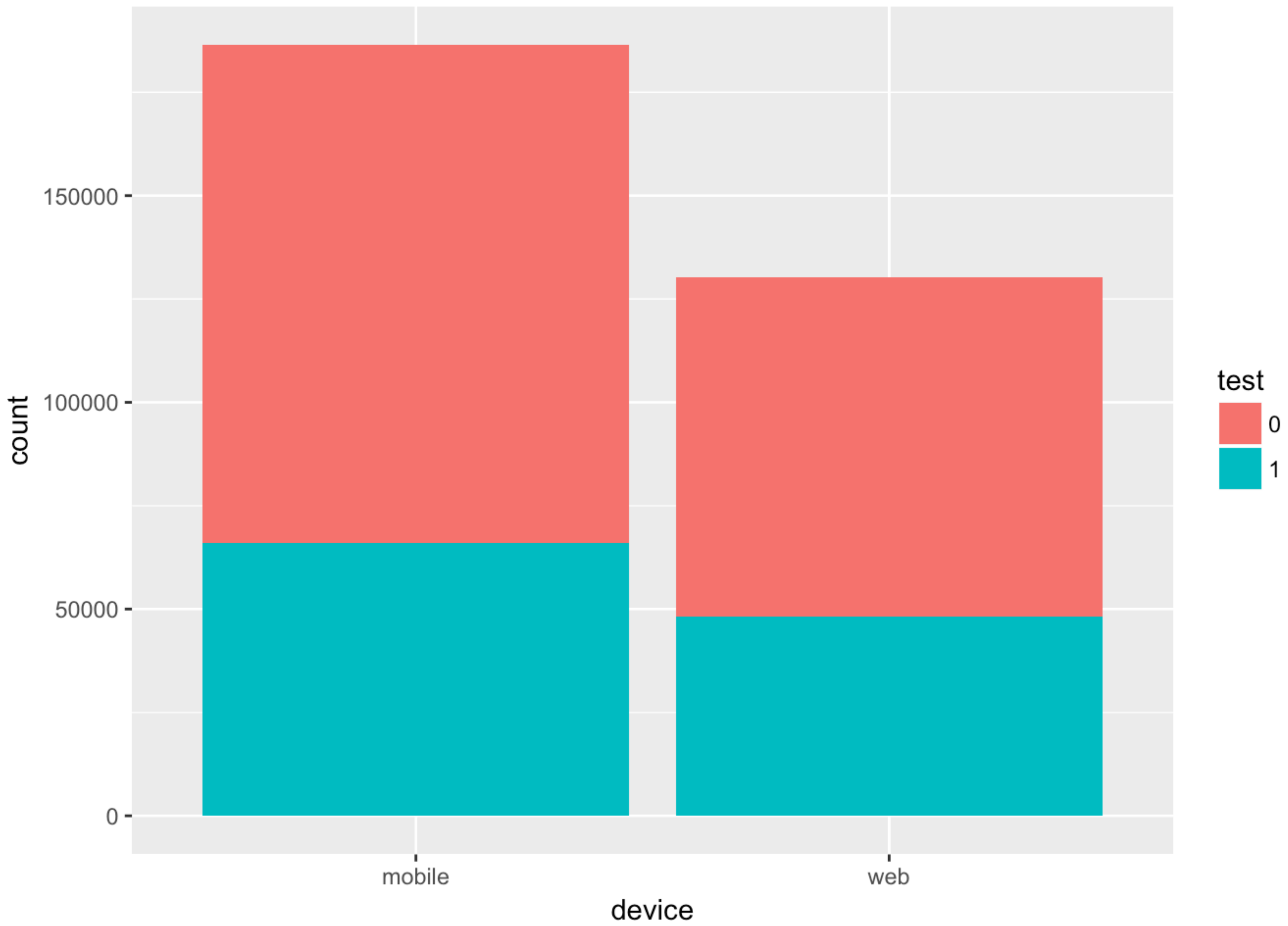
```
library(ggplot2)
ggplot(data, aes(x=source,fill=test)) + geom_bar()+
  theme(axis.text.x = element_text(angle=45, hjust=1))
```



The users were distributed 66% to the control group and 33% to the test group in various sources.

Users distribution in device:

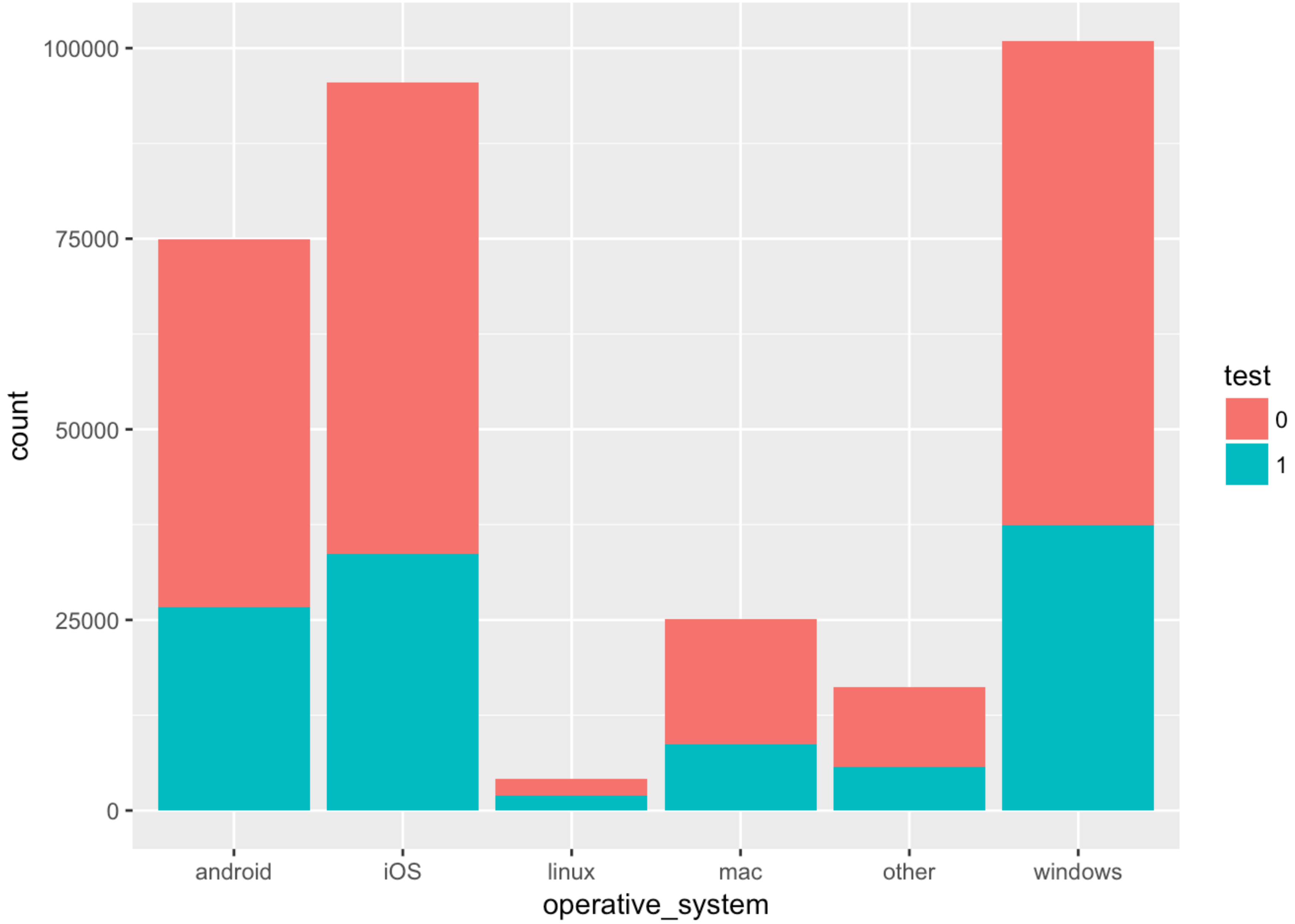
```
ggplot(data, aes(x=device,fill=test)) + geom_bar()
```



The mobile users and web users were not 50/50 split but the user distribution for control and test were 66% and 33% respectively for both mobile and web users.

Users distribution in operative system:

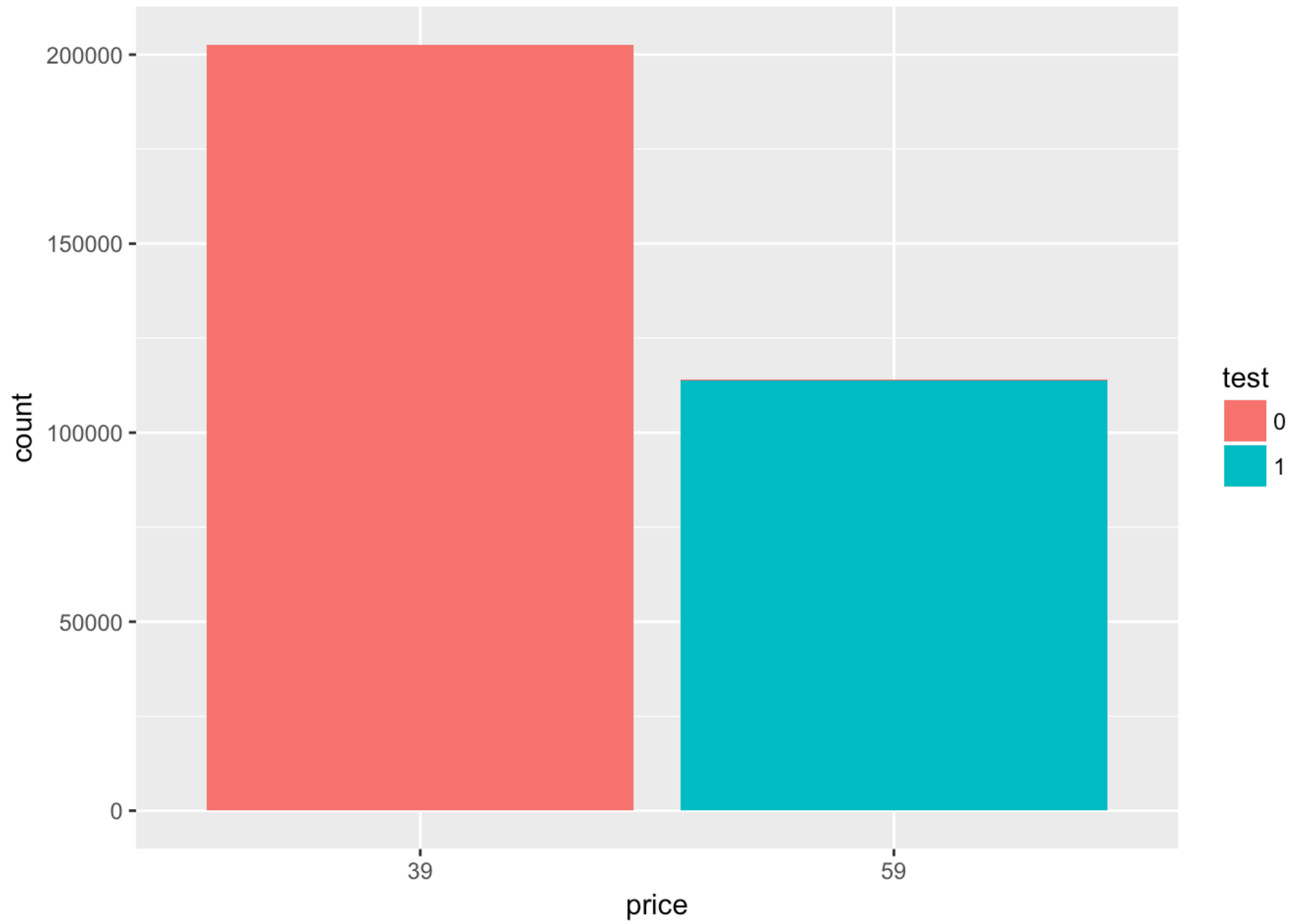
```
ggplot(data, aes(x=operative_system,fill=test)) + geom_bar()
```



Like source and device, users were distributed at 66% for control group and 33% for test group for different operative systems except linux. The user distribution in linux is 50/50 split to both control and test group.

Users distribution in price:

```
ggplot(data, aes(x=price,fill=test)) + geom_bar()
```



Something wrong in this plot. According to test setup, all $39 price should be in control group and all $59 price should be in test group. Looks like some $59 price are in control group.

Let's check how many old price ($39) in test group(test=1) which was supposed to be in control group(0).Same check for new price($59) as well.

```
nrow(data[(data$price==39 & data$test==1),])
```

```
## [1] 155
```

```
nrow(data[(data$price==59 & data$test==0),])
```

```
## [1] 210
```

Looks like 155 old price is in test group and 210 new price in control group. Let's remove them from the data.

```
data_updated <- subset(data, !(price==39 & test==1) & !(price==59 & test==0))
```

Let's double check that no old price data in test group and new price data in control group.

```
nrow(data_updated[(data_updated$price==39 & data_updated$test==1),])
```

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

```
nrow(data_updated[(data_updated$price==59 & data_updated$test==0),])
```

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

So far so good.

t-test

Now let's do the t-test on revenue made per user:

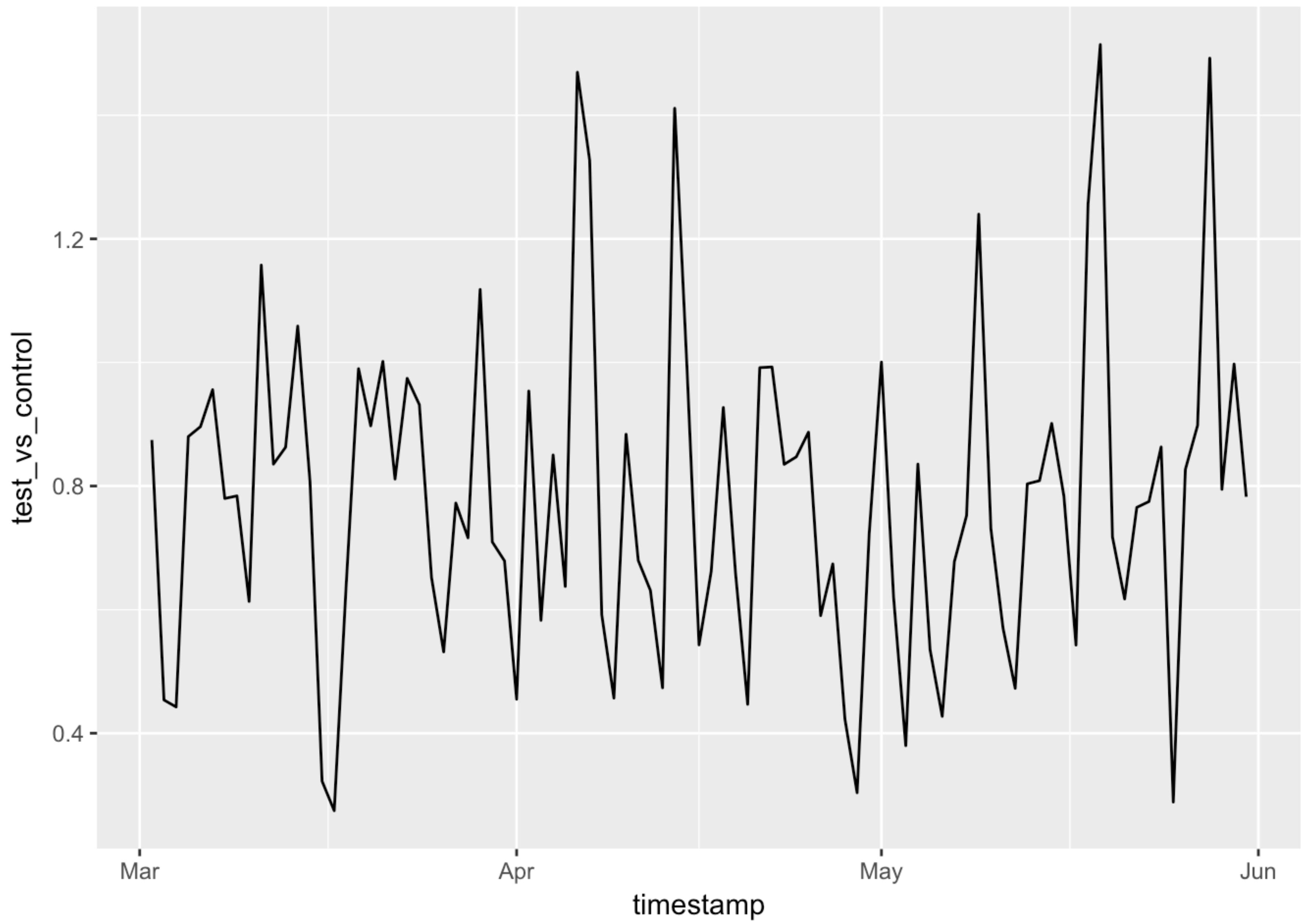
```
t.test(data_updated$converted[data_updated$test==1]*59, data_updated$converted[data_u
pdated$test==0]*39)
```

```
##
##  Welch Two Sample t-test
##
## data:  data_updated$converted[data_updated$test == 1] * 59 and data_updated$conver
ted[data_updated$test == 0] * 39
## t = 5.7152, df = 186140, p-value = 1.097e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.09308237 0.19024734
## sample estimates:
## mean of x mean of y
## 0.9177479 0.7760830
```

The revenue made per user in the test group is 91.8% while in the control group is 77.6%. That's a 18% lift.

Let's check the day by day plot of test vs control:

```
library(dplyr)
data_by_day <- data_updated %>%
               group_by(timestamp) %>%
               summarize(test_vs_control = mean(converted[test==1])/
                                            mean(converted[test==0]))
ggplot(data_by_day, aes(x=timestamp, y=test_vs_control))+
  geom_line(stat='identity')
```



Check the conversion in each segment.

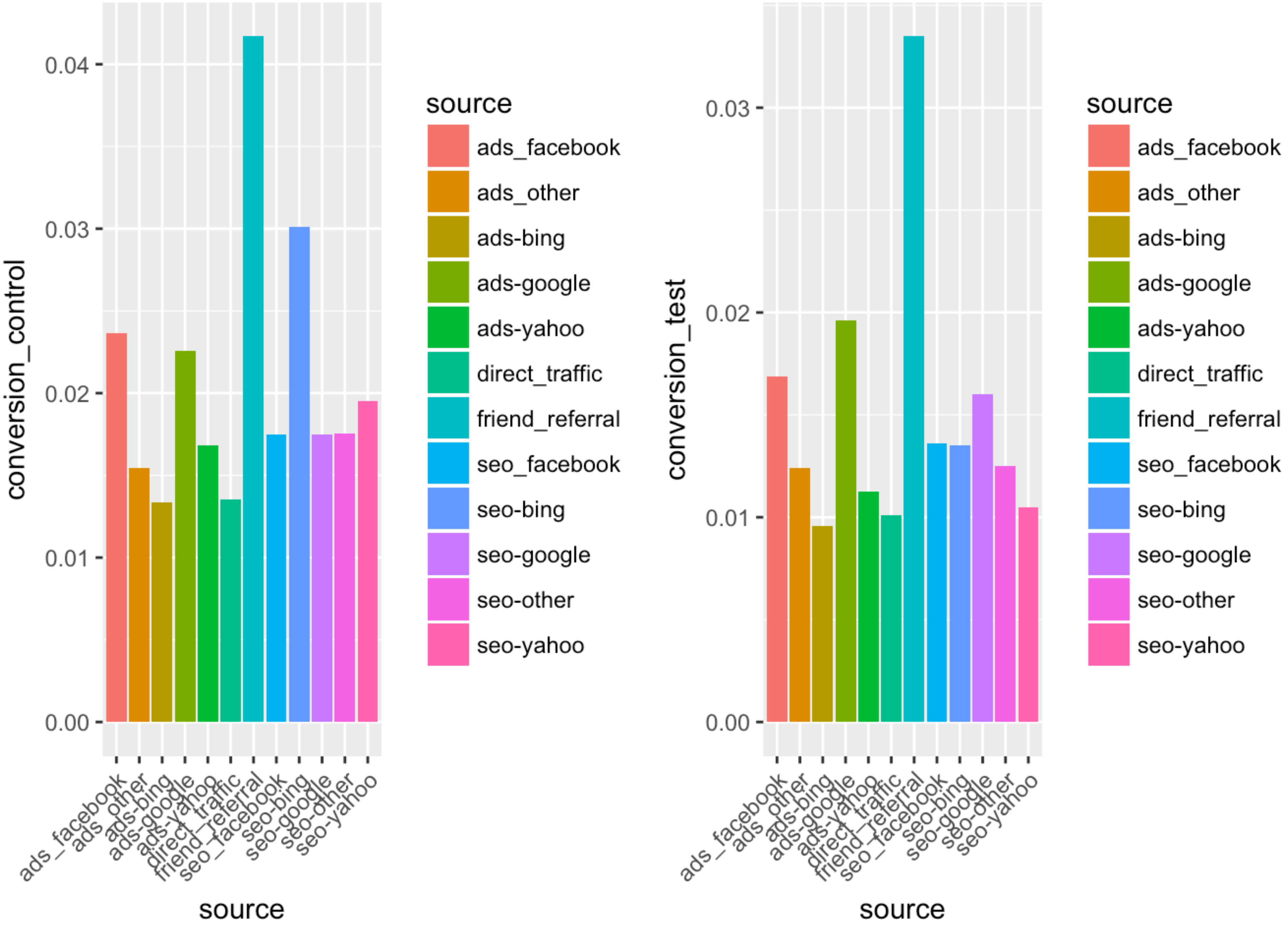
Let's check the user conversion in source:

```
library(gridExtra)
data_by_source <- data_updated %>%
                  group_by(source) %>%
                  summarize(conversion_control = mean(converted[test==0]),
                            conversion_test = mean(converted[test==1]))

p1 <- ggplot(data_by_source, aes(x=source,y=conversion_control)) +
      geom_bar(stat='identity',aes(fill=source))+
      theme(axis.text.x = element_text(angle=45, hjust=1))

p2 <- ggplot(data_by_source, aes(x=source,y=conversion_test)) +
      geom_bar(stat='identity',aes(fill=source))+
      theme(axis.text.x = element_text(angle=45, hjust=1))

grid.arrange(p1,p2, ncol=2)
```



Looks like friend_referral source is doing better in conversion. So, company can focus on it and make it easier for customer so that they can refer more friends.

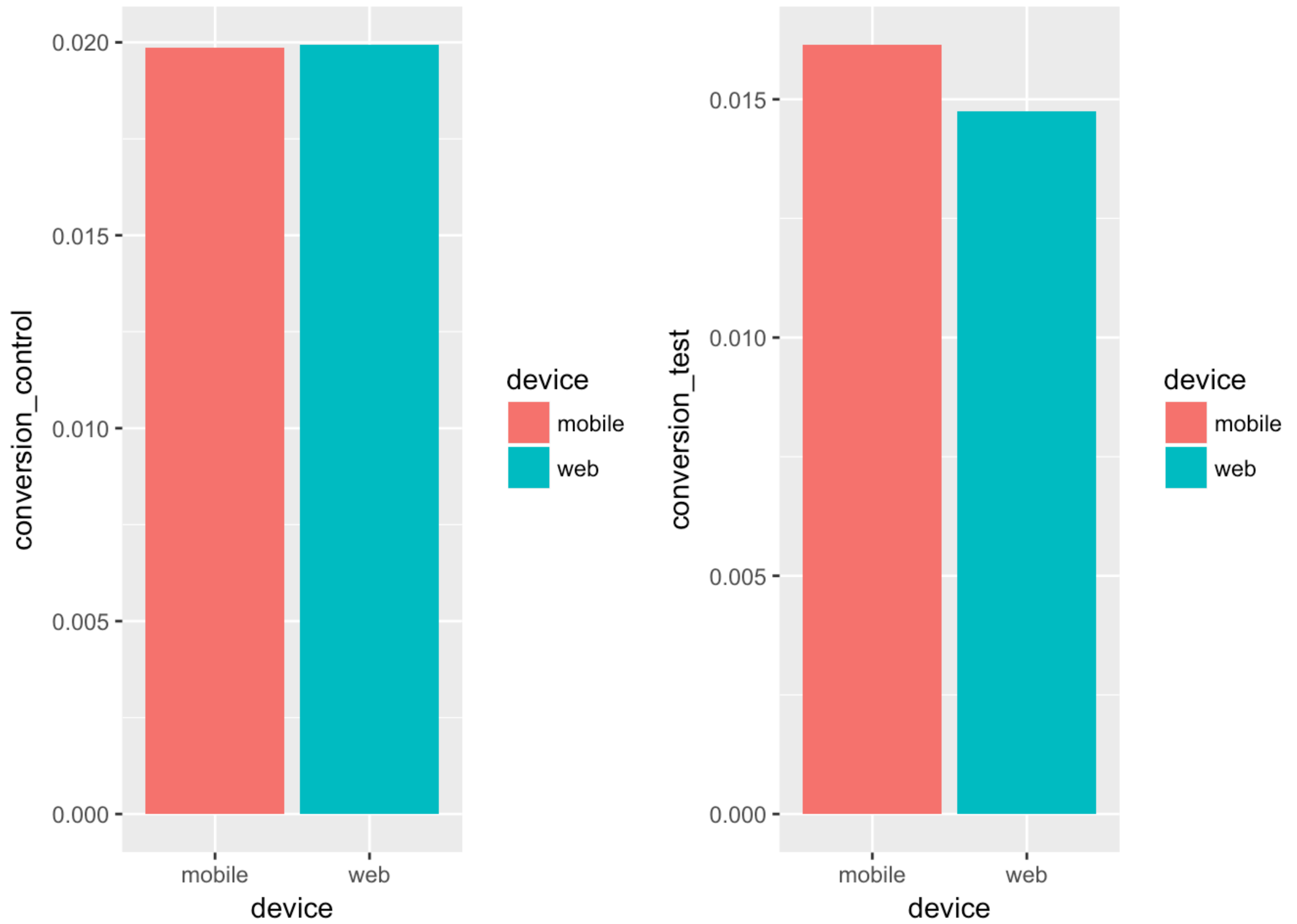
Check the user conversion in device:

```
data_by_device <- data_updated %>%
                  group_by(device) %>%
                  summarize(conversion_control = mean(converted[test==0]),
                            conversion_test = mean(converted[test==1]))

p3 <- ggplot(data_by_device, aes(x=device,y=conversion_control)) +
      geom_bar(stat='identity',aes(fill=device))

p4 <- ggplot(data_by_device, aes(x=device,y=conversion_test)) +
      geom_bar(stat='identity',aes(fill=device))

grid.arrange(p3,p4, ncol=2)
```



Mobile users are better than web users for conversion, therefore company can focus more on mobile users. Beside that they can spend some money on web ads to attract more web users as well.

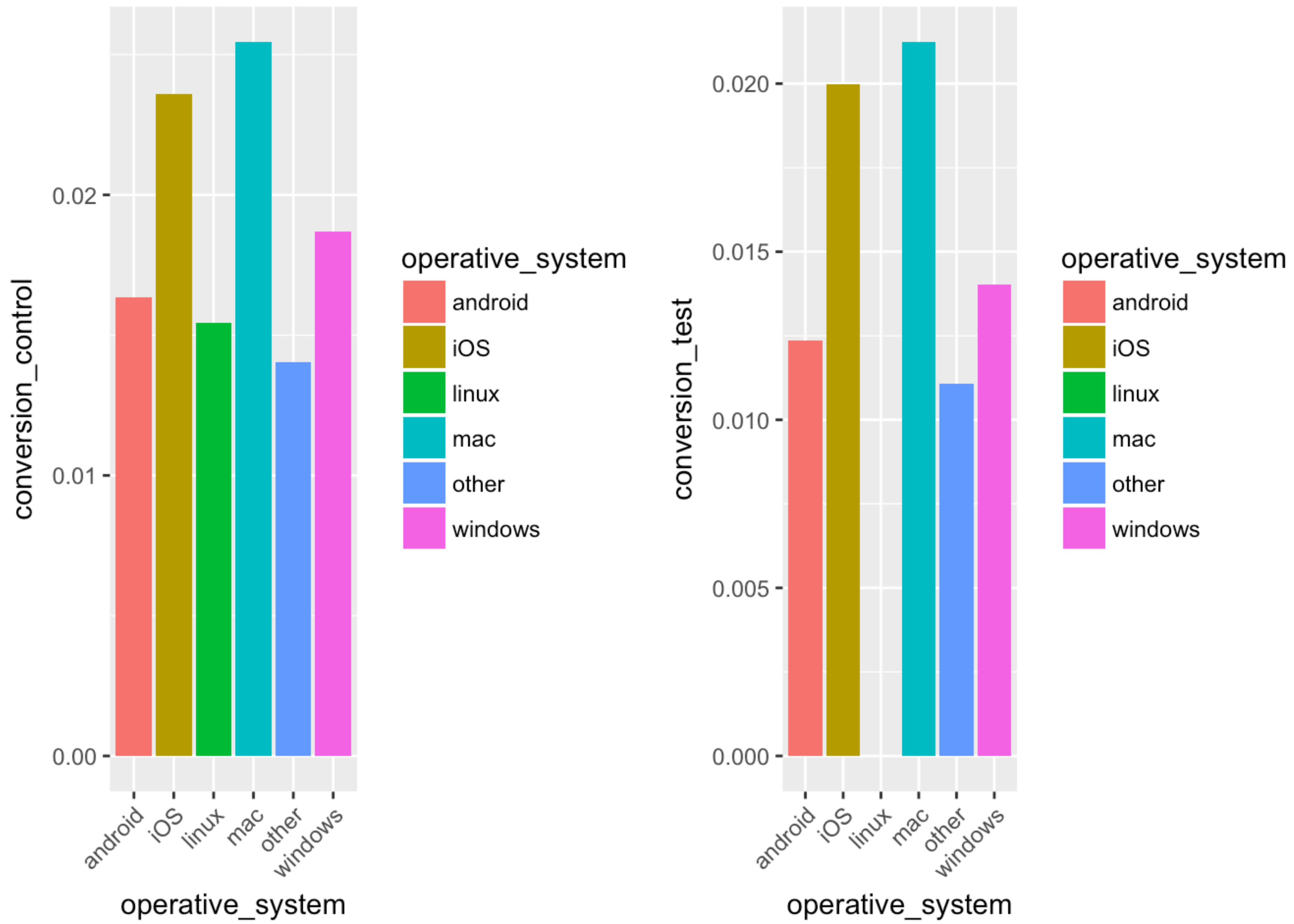
Check the user conversion in operative system:

```
data_by_os <- data_updated %>%
              group_by(operative_system) %>%
              summarize(conversion_control = mean(converted[test==0]),
                        conversion_test = mean(converted[test==1]))

p5 <- ggplot(data_by_os, aes(x=operative_system,y=conversion_control)) +
      geom_bar(stat='identity',aes(fill=operative_system))+
      theme(axis.text.x = element_text(angle=45, hjust=1))

p6 <- ggplot(data_by_os, aes(x=operative_system,y=conversion_test)) +
      geom_bar(stat='identity',aes(fill=operative_system))+
      theme(axis.text.x = element_text(angle=45, hjust=1))

grid.arrange(p5,p6, ncol=2)
```



Something wrong here. Looks like there is some bug issue in linux operating system. We will remove linux from the data.

However, iOS is doing better than android for mobile users and mac is in better position than windows for web users. Therefore company can focus on mac and iOS users. Beside that, Android app might need improvement.

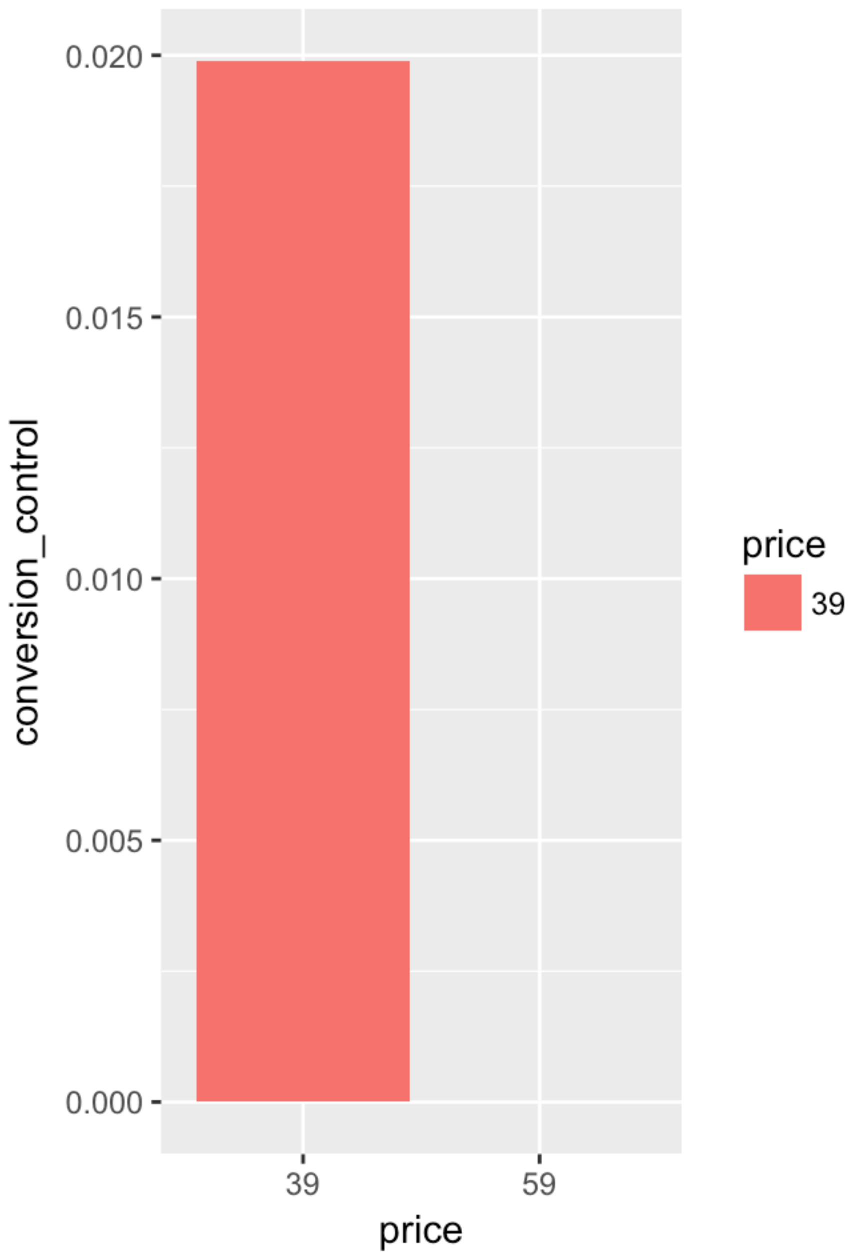
Check the user conversion in price:

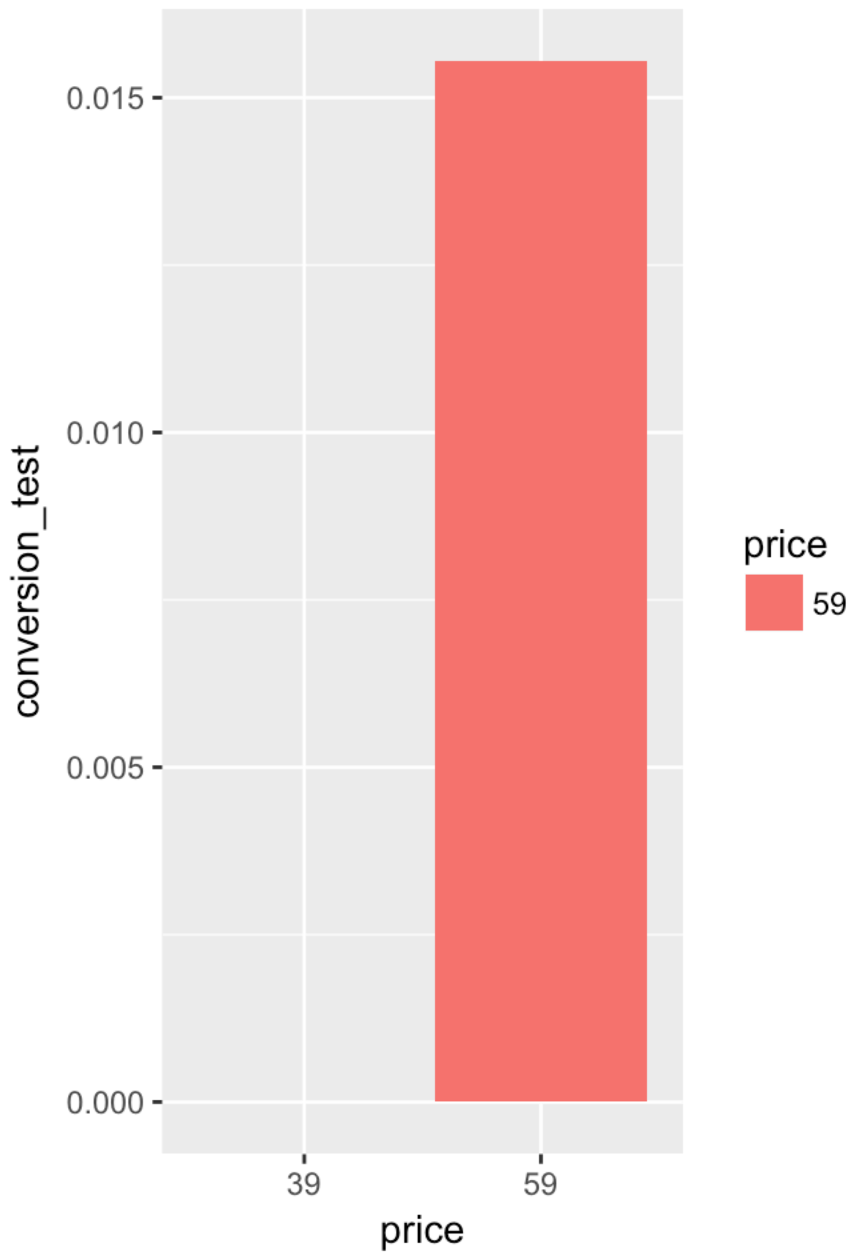
```
data_by_price <- data_updated %>%
                 group_by(price) %>%
                 summarize(conversion_control = mean(converted[test==0]),
                           conversion_test = mean(converted[test==1]))

p7 <- ggplot(data_by_price, aes(x=price,y=conversion_control)) +
      geom_bar(stat='identity',aes(fill=price))

p8 <- ggplot(data_by_price, aes(x=price,y=conversion_test)) +
      geom_bar(stat='identity',aes(fill=price))

grid.arrange(p7,p8, ncol=2)
```





Now remove the linux operating system from the data.

```
data_rev <- subset(data_updated, operative_system !='linux')
```

Check the summary of the revised data:

```
summary(data_rev)
```

```
##     user_id                    city          country          lat
##  Min.   :      3   New York    : 25401   USA :271664   Min.   :19.70
##  1st Qu.: 249612   Chicago     :  7071   NA's: 40641   1st Qu.:33.66
##  Median : 498990   Houston     :  6608                 Median :37.74
##  Mean   : 499258   San Antonio:  4554                  Mean   :37.11
##  3rd Qu.: 748893   Los Angeles:  4089                  3rd Qu.:40.70
##  Max.   :1000000   (Other)    :223941                  Max.   :61.18
##                    NA's       : 40641                  NA's   :40641
##       long             timestamp                  source
##  Min.   :-157.80   Min.   :2015-03-02   direct_traffic:59551
##  1st Qu.:-112.20   1st Qu.:2015-03-26   ads-google    :58464
##  Median : -88.93   Median :2015-04-17   ads_facebook  :52715
##  Mean   : -93.98   Mean   :2015-04-16   ads_other     :29436
##  3rd Qu.: -78.91   3rd Qu.:2015-05-09   seo-google    :22881
##  Max.   :  30.31   Max.   :2015-05-31   ads-bing      :22584
##  NA's   :40641                          (Other)       :66674
##     device       operative_system test       price         converted
##  mobile:186267   android: 74870   0:200313   39:200313   Min.   :0.00000
##  web   :126038   iOS    : 95353   1:111992   59:111992   1st Qu.:0.00000
##                  linux  :     0                          Median :0.00000
##                  mac    : 25055                          Mean   :0.01847
##                  other  : 16177                          3rd Qu.:0.00000
##                  windows:100850                          Max.   :1.00000
##
```

Let's do the t-test again:

```
t.test(data_rev$converted[data_rev$test==1]*59, data_rev$converted[data_rev$test==0]*
39)
```

```
##
##  Welch Two Sample t-test
##
## data:  data_rev$converted[data_rev$test == 1] * 59 and data_rev$converted[data_rev
$test == 0] * 39
## t = 6.1842, df = 181670, p-value = 6.253e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1062367 0.2048203
## sample estimates:
## mean of x mean of y
## 0.9335310 0.7780024
```

The revenue made per user in test group is 93.4% and in control group is 77.8%. It's a 20% lift. Therefore, the test is winning and the company should sell the software for $59.

For the last part of the question, we have to do the sample size analysis. Assume the following:

- significance level = 0.05

- power = 0.8
- expected standard deviation of the change = 1
- minimum effect size = 2 % = 0.02

Let’s do the power.t.test to calculate the sample size:

```
power.t.test (delta=0.02, power=0.8,type='two.sample',alternative='two.sided')
```

```
##
##      Two-sample t test power calculation
##
##               n = 39245.36
##           delta = 0.02
##              sd = 1
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

The sample size is 39,246 for each group. Therefore the total sample size = 2*39246=78,492.

```
visitors_per_day <- length(data$user_id)/as.numeric(difftime(max(data$timestamp), min
(data$timestamp), units='days'))
visitors_per_day
```

```
## [1] 3520
```

The visitors per day is 3,520.

Expected experimet duration = sample size/number of visitors to the tested pages

```
sample_size <- 78492
expected_experiment_duration <- sample_size/visitors_per_day
expected_experiment_duration
```

```
## [1] 22.29886
```

Therefore, the expected experiment duration is 23 days which is in the range of standard A/B testing run time.