Data Challenge

Abhinav Pathak

July 01, 2017

```
suppressPackageStartupMessages({
library(readr)
library(sqldf)
library(ggplot2)
library(dplyr)
library(Hmisc)
library(tidyverse)
library(lubridate)
library(gridExtra)
})
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
signups <- read_csv("signups.csv")
visits <- read_csv("visits.csv")
```

```
summary(signups)
```

```
##        X1              uid              signup_dt
##  Min.   :    1   Min.   :21635668   Min.   :2016-06-01
##  1st Qu.:17998   1st Qu.:22197086   1st Qu.:2016-07-11
##  Median :35995   Median :22783414   Median :2016-08-20
##  Mean   :35995   Mean   :22793449   Mean   :2016-08-19
##  3rd Qu.:53992   3rd Qu.:23399901   3rd Qu.:2016-09-28
##  Max.   :71989   Max.   :23951517   Max.   :2016-11-03
##   auth_type             device
##  Length:71989       Min.   :1.000
##  Class :character   1st Qu.:1.000
##  Mode  :character   Median :2.000
##                     Mean   :3.298
##                     3rd Qu.:6.000
##                     Max.   :7.000
```

```
summary(visits)
```

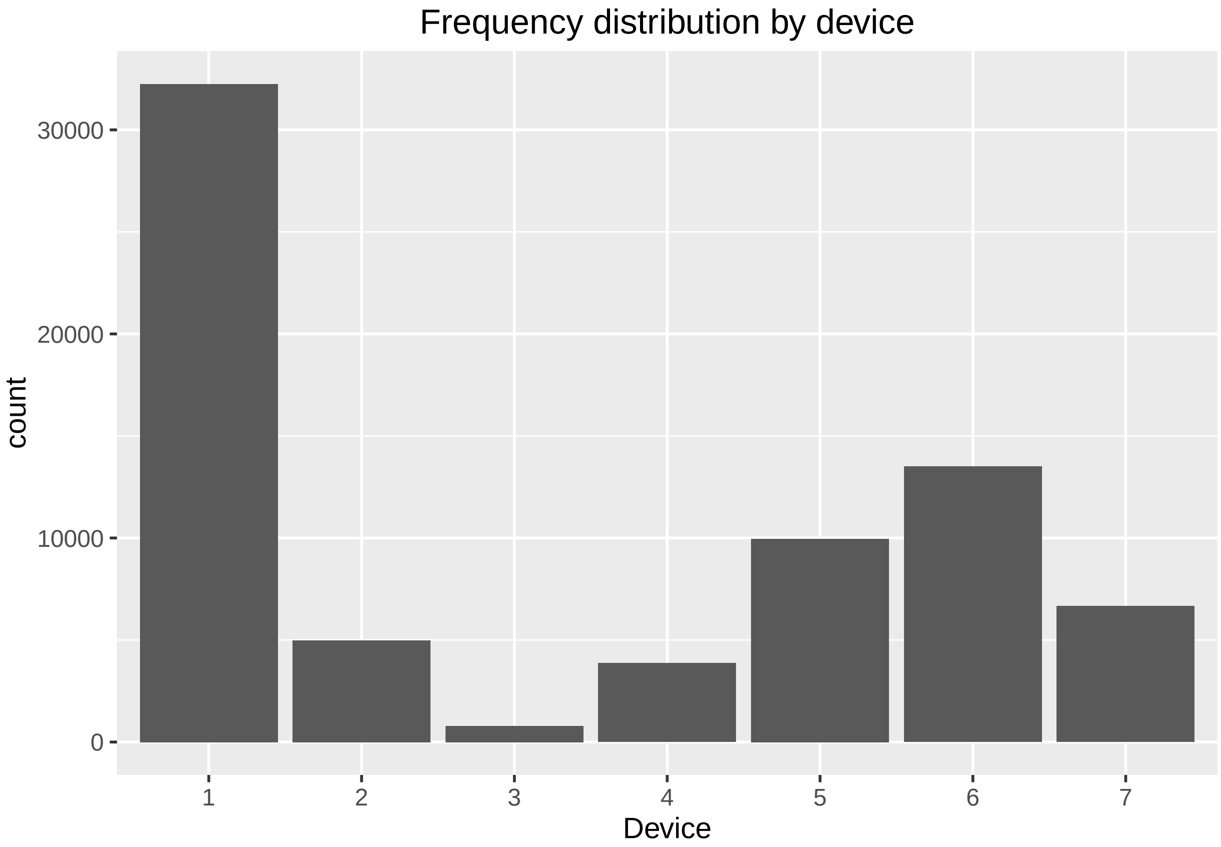
```
##        X1               uid                 dt
##  Min.   :      1   Min.   :21635668   Min.   :2016-06-01
##  1st Qu.: 257204   1st Qu.:22138963   1st Qu.:2016-09-06
##  Median : 514407   Median :22680200   Median :2016-11-02
##  Mean   : 514407   Mean   :22722109   Mean   :2016-11-11
##  3rd Qu.: 771610   3rd Qu.:23306308   3rd Qu.:2017-01-20
##  Max.   :1028813   Max.   :23951517   Max.   :2017-04-27
```

- From summary statistics, it is evident that there are no null values or missing values or extreme values

Question 1

Part a)

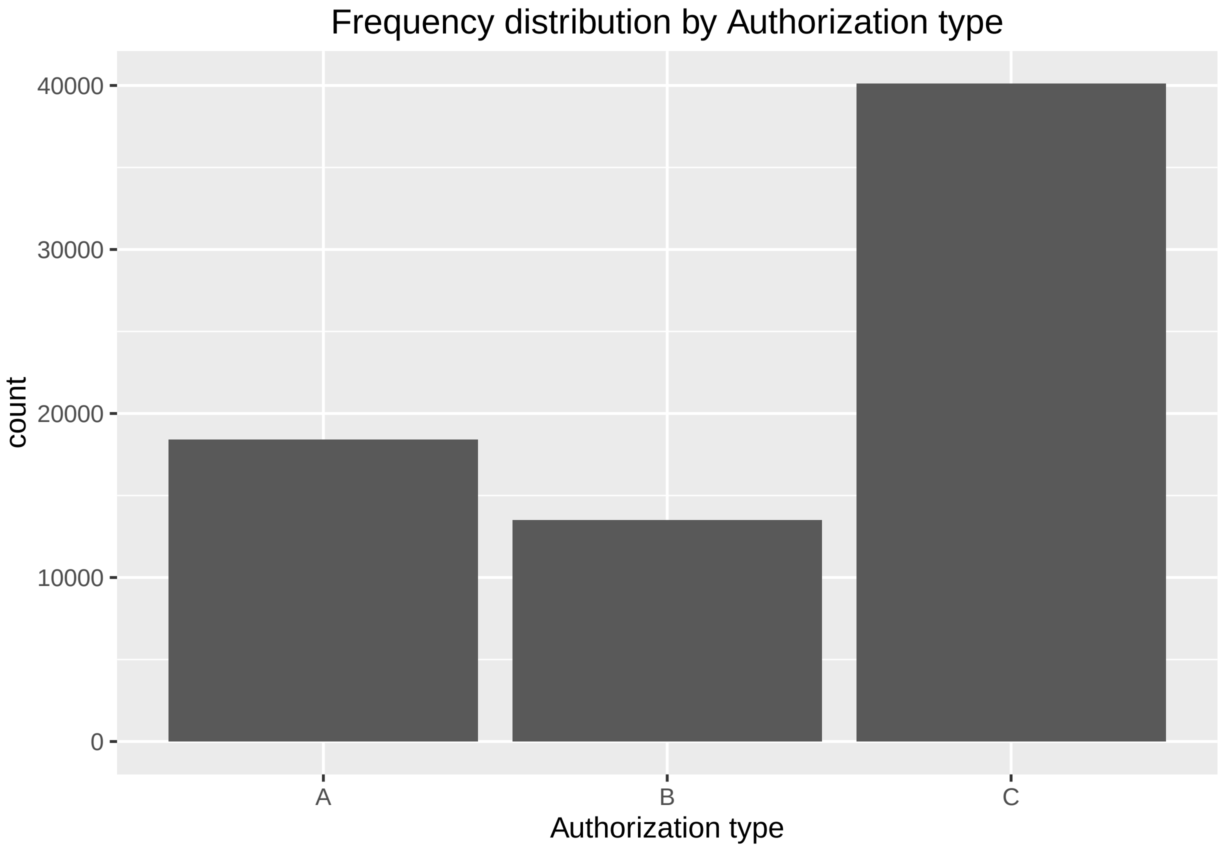
```
ggplot(signups) +
  geom_bar(aes(x = factor(device))) +
  xlab(label = "Device") +
  ggtitle("Frequency distribution by device")
```



- It is evident that Device 1 has the highest number of signups followed by device 6. This is the device which users prefer over other devices

Part b)

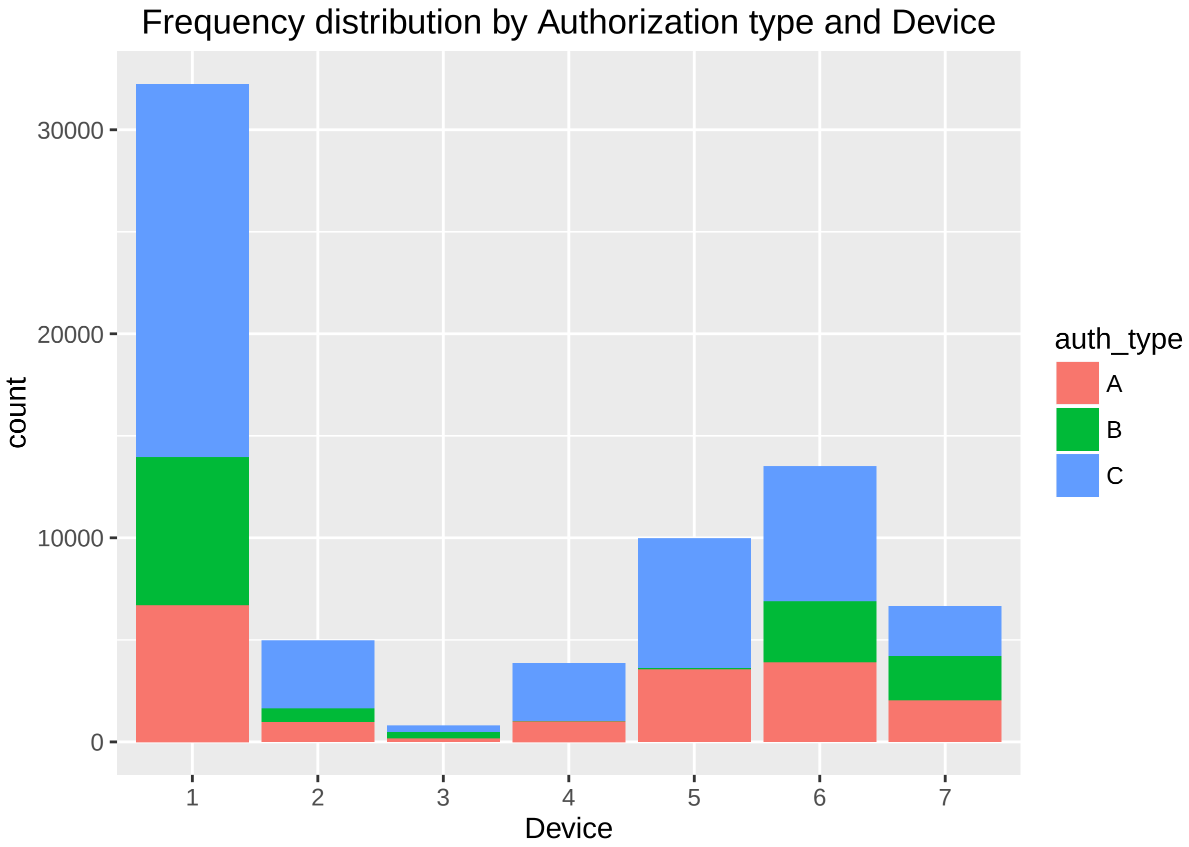
```
ggplot(signups) +
  geom_bar(aes(x = factor(auth_type))) +
  xlab(label = "Authorization type") +
  ggtitle("Frequency distribution by Authorization type")
```



- It is evident that users sign up through Authorization type C

Part c)

```
ggplot(signups) +
  geom_bar(aes(x = factor(device), fill = auth_type)) +
  xlab(label = "Device") +
  ggtitle("Frequency distribution by Authorization type and Device")
```



- It is evident that users sign up through Authorization type C more, irrespective of the device type, because the proportion of C is higher in each of the Device case. It must mean that, users are very comfortable signing up through Authorization type C. Device 7 is an exception, where proportion for each of the 3 Authorization type seems equal.

Question 2

Create a table in long format with count of visitors at visit__week__number X SignUpDate

```
## Create an empty dataframe to store the final table
df_final <- data.frame()

## Run a for loop for each signup date, and keep appending it into the final dataframe
for (date_var in 0:151 ) {
date_week <- data.frame( week = rep(1:24, each= 7),
                   date = seq(as.Date("2016-06-01")+date_var+1, by =1 , len = 24*7))
## create a list user ids for 1 singup date
uid_list <- signups %>%
  filter(signup_dt == as.Date("2016-06-01")+date_var ) %>%
  select(uid)

## a temporary table to store weekly visitor counts for each signup date
visits_2 <- visits %>%
  left_join(date_week, by = c("dt" = "date")) %>%
  filter(!is.na(week)) %>%
  group_by(uid, week) %>%
  summarise(visit_count = n()) %>%
  filter(uid %in% (uid_list$uid)) %>%
  group_by(week) %>%
  summarise(count = n()) %>%
  mutate(signup_dt = date_var)

## Final table where the data from temporary table is appende after each loop
df_final <- rbind(df_final, visits_2)
}
```

Transformation of rows to columns

```
## a temporary dataframe which matches serial number to signup date
date_df <- data.frame(date = seq(as.Date("2016-06-01"), as.Date("2016-10-30"), by=1),
                         sno = seq(0, 151,by=1))
## Adding the date column
df_final <- df_final %>%
  left_join(date_df, by = c("signup_dt" = "sno"))

## selecting only relevant columns for further analysis
df_final <-  df_final %>%
    select(date, week, count)

## Using spread function for transformation of week numbers to columns
df_final_transformed <- df_final %>% spread(key = week, value = count)
signup_count <- signups %>%
  group_by(signup_dt) %>%
  summarise (signup_count = n())

## Adding signup count for each signup date from signup_count table which was created earlier
df_final_transformed_1 <- df_final_transformed %>%
  inner_join(signup_count,by = c("date" = "signup_dt"))
```

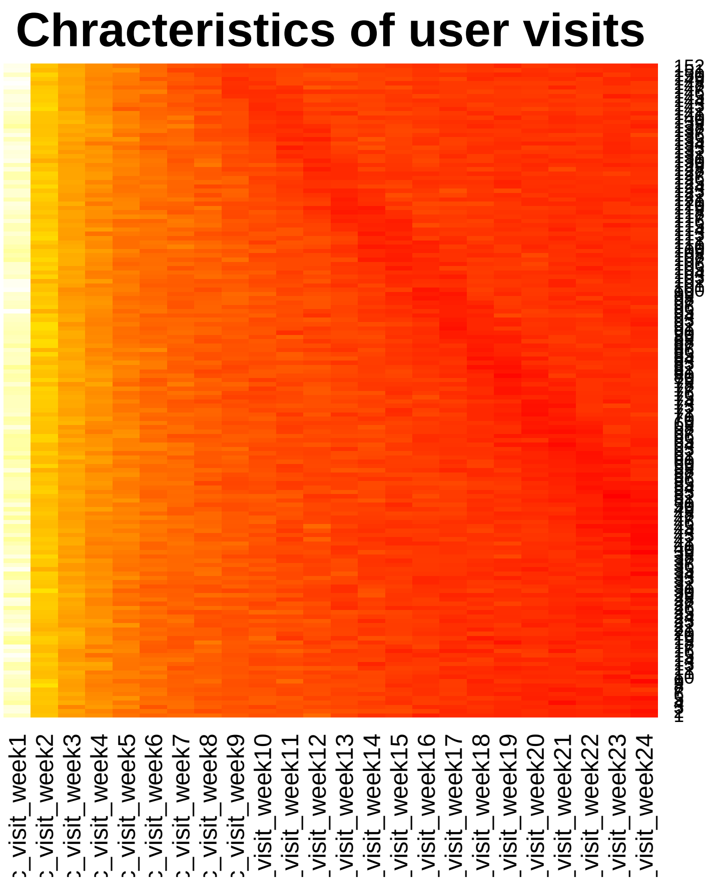
Convert count into %visits

```
## Transmute function creates a new column and drops the old one
df_final_transformed_2 <- df_final_transformed_1 %>%
      transmute( signup_date = date,
                 signup_count = signup_count,
                 perc_visit_week1 =`1`/signup_count,
                 perc_visit_week2= `2`/signup_count,
                 perc_visit_week3 =`3`/signup_count,
                 perc_visit_week4 =`4`/signup_count,
                 perc_visit_week5 =`5`/signup_count,
                 perc_visit_week6 =`6`/signup_count,
                 perc_visit_week7 =`7`/signup_count,
                 perc_visit_week8 =`8`/signup_count,
                 perc_visit_week9 =`9`/signup_count,
                 perc_visit_week10 =`10`/signup_count,
                 perc_visit_week11 =`11`/signup_count,
                 perc_visit_week12 =`12`/signup_count,
                 perc_visit_week13 =`13`/signup_count,
                 perc_visit_week14 =`14`/signup_count,
                 perc_visit_week15 =`15`/signup_count,
                 perc_visit_week16 =`16`/signup_count,
                 perc_visit_week17 =`17`/signup_count,
                 perc_visit_week18 =`18`/signup_count,
                 perc_visit_week19 =`19`/signup_count,
                 perc_visit_week20 =`20`/signup_count,
                 perc_visit_week21 =`21`/signup_count,
                 perc_visit_week22 =`22`/signup_count,
                 perc_visit_week23 =`23`/signup_count,
                 perc_visit_week24 =`24`/signup_count  )

head(df_final_transformed_2)
```

```
## # A tibble: 6 × 26
##   signup_date signup_count perc_visit_week1 perc_visit_week2
##        <date>        <int>            <dbl>            <dbl>
## 1  2016-06-01          400        0.5975000        0.4000000
## 2  2016-06-02          439        0.6674260        0.4282460
## 3  2016-06-03          407        0.6805897        0.4324324
## 4  2016-06-04          436        0.6720183        0.4701835
## 5  2016-06-05          540        0.6240741        0.4259259
## 6  2016-06-06          463        0.6328294        0.4233261
## # ... with 22 more variables: perc_visit_week3 <dbl>,
## #   perc_visit_week4 <dbl>, perc_visit_week5 <dbl>,
## #   perc_visit_week6 <dbl>, perc_visit_week7 <dbl>,
## #   perc_visit_week8 <dbl>, perc_visit_week9 <dbl>,
## #   perc_visit_week10 <dbl>, perc_visit_week11 <dbl>,
## #   perc_visit_week12 <dbl>, perc_visit_week13 <dbl>,
## #   perc_visit_week14 <dbl>, perc_visit_week15 <dbl>,
## #   perc_visit_week16 <dbl>, perc_visit_week17 <dbl>,
## #   perc_visit_week18 <dbl>, perc_visit_week19 <dbl>,
## #   perc_visit_week20 <dbl>, perc_visit_week21 <dbl>,
## #   perc_visit_week22 <dbl>, perc_visit_week23 <dbl>,
## #   perc_visit_week24 <dbl>
```

Generate Heat map

```
## Convert to matrix
matrix_final_transformed_2 <- df_final_transformed_2 %>%
  select(-signup_date,-signup_count) %>%
  as.matrix()

heatmap(matrix_final_transformed_2,
        scale = "row",
        col = heat.colors(256),
        main = "Chracteristics of user visits",
        Rowv = NA,
        Colv = NA)
```



- Approximately Week 17 onwards, it reaches a steady state (because the pallet of the whole column is red). This conclusion is from Heat map, where the whole column has turned red, and there is not much difference with other signup dates. This points to a very import chractareistic of user behavior in general. After a certain period of time, the users who really like the service become the regular users. Other users, either did not find the product/service useful or easy to use or cost efficient. The company should note that the users who are still using the service are the loyal ones and they are most likely going to be permanent ones and they awill have high lifetime value and they should be taken care of. On the other hand, the company should also focus on the user retention and find out why are customers stopping to use the product/service, what are the pain points and what can be done to make life easier for them, which leads to customer satisfaction and generates more revenue for the company in long term

Question 3

Create a table in long format with count of visitors at visit_week_number X SignUpDate X Auth_type

```
## Two dataframes to store week number information for each of the two dates in question
date_week_1 <- data.frame( week = rep(1:24, each= 7),
                          date = seq(as.Date("2016-07-24")+1, by =1 , len = 24*7))

date_week_2 <- data.frame( week = rep(1:24, each= 7),
                          date = seq(as.Date("2016-08-18")+1, by =1 , len = 24*7))

## create user column with July 24th as signup date
```

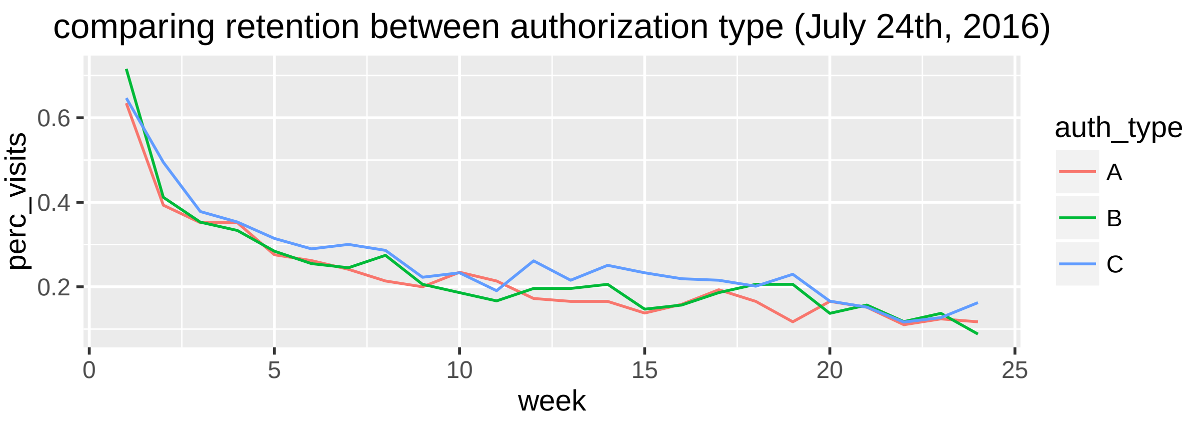
```
uid_list_1 <- signups %>%
  filter(signup_dt == as.Date("2016-07-24"))%>%
  select(uid, auth_type)

## create user column with August 18th as signup date
uid_list_2 <- signups %>%
  filter(signup_dt == as.Date("2016-08-18"))%>%
  select(uid, auth_type)

visits__table_1 <- visits %>%
  left_join(date_week_1, by = c("dt" = "date")) %>%
  filter(!is.na(week)) %>%
  group_by(uid, week) %>%
  summarise(visit_count = n()) %>%
  filter(uid %in% (uid_list_1$uid)) %>%
  left_join(signups, by= c("uid" = "uid")) %>%
  group_by(week,auth_type) %>%
  summarise(count = n()) %>%
  mutate(signup_dt = as.Date("2016-07-24"))

visits__table_2 <- visits %>%
  left_join(date_week_2, by = c("dt" = "date")) %>%
  filter(!is.na(week)) %>%
  group_by(uid, week) %>%
  summarise(visit_count = n()) %>%
  filter(uid %in% (uid_list_2$uid)) %>%
  left_join(signups, by= c("uid" = "uid")) %>%
  group_by(week,auth_type) %>%
  summarise(count = n()) %>%
  mutate(signup_dt = as.Date("2016-08-18"))

## Final table where the data from 2 temporary tables is combined
df_final_q3 <- data.frame()
df_final_q3 <- rbind(visits__table_1, visits__table_2)

## Data transformation

df_final_q3_transformed_1 <- df_final_q3 %>%
  spread(key = week, value = count)  %>%
  arrange(signup_dt)

signup_authtype_count <- signups %>%
  group_by(signup_dt, auth_type) %>%
  summarise (signup_count = n())

df_final_q3_transformed_2 <- df_final_q3_transformed_1 %>%
  inner_join(signup_authtype_count,by = c("signup_dt" = "signup_dt", "auth_type" ="auth_type"))
```

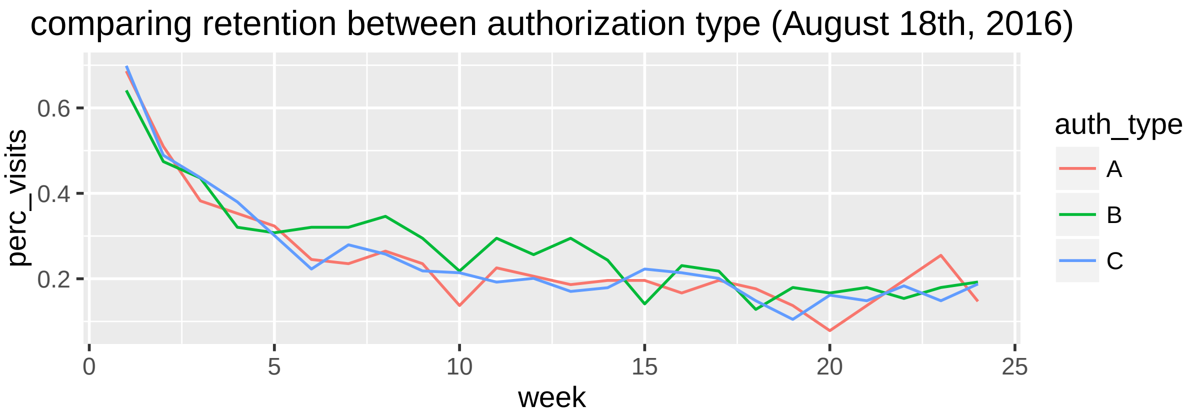
Transformation of rows to columns

```
## Transmute function creates a new column and drops the old one
df_final_q3_transformed_3 <- df_final_q3_transformed_2 %>%
      transmute( signup_date = signup_dt,
                 auth_type = auth_type,
                 signup_count = signup_count,
                 perc_visit_week1 =`1`/signup_count,
                 perc_visit_week2= `2`/signup_count,
                 perc_visit_week3 =`3`/signup_count,
                 perc_visit_week4 =`4`/signup_count,
                 perc_visit_week5 =`5`/signup_count,
                 perc_visit_week6 =`6`/signup_count,
                 perc_visit_week7 =`7`/signup_count,
                 perc_visit_week8 =`8`/signup_count,
                 perc_visit_week9 =`9`/signup_count,
                 perc_visit_week10 =`10`/signup_count,
                 perc_visit_week11 =`11`/signup_count,
                 perc_visit_week12 =`12`/signup_count,
                 perc_visit_week13 =`13`/signup_count,
                 perc_visit_week14 =`14`/signup_count,
                 perc_visit_week15 =`15`/signup_count,
                 perc_visit_week16 =`16`/signup_count,
                 perc_visit_week17 =`17`/signup_count,
                 perc_visit_week18 =`18`/signup_count,
                 perc_visit_week19 =`19`/signup_count,
                 perc_visit_week20 =`20`/signup_count,
                 perc_visit_week21 =`21`/signup_count,
                 perc_visit_week22 =`22`/signup_count,
                 perc_visit_week23 =`23`/signup_count,
                 perc_visit_week24 =`24`/signup_count  )
```

Let us plot and see if there is a difference between authorization type

```
temp <- df_final_q3_transformed_3 %>%
  gather(perc_visit_week1:perc_visit_week24, key = "week", value = "perc_visits") %>%
  separate(week, into = c("a","week"), sep = "k", convert = FALSE) %>%
  mutate(week = as.numeric(week)) %>%
  select(-a) %>%
  arrange(week)

gg1 <- ggplot(temp %>% filter(signup_date == "2016-07-24")) +
  geom_line(aes(x= week, y = perc_visits, group = auth_type, color = auth_type)) +
  ggtitle("comparing retention between authorization type (July 24th, 2016)")


gg2 <- ggplot(temp %>% filter(signup_date == "2016-08-18")) +
  geom_line(aes(x= week, y = perc_visits, group = auth_type, color = auth_type)) +
  ggtitle("comparing retention between authorization type (August 18th, 2016)")

grid.arrange(gg1, gg2)
```





- Retention does vary by authorization type. It is clear from the 1st plot (users who signed up on July 24th) that retention of authorization type "c" is higher in most of the weeks compared to Authorization type A and B. This means that, customers with Authoriation type A and B are facing some issues. For eg, it could be taking time to authorize everytime they login.
- Whereas, (users who signed up on August 18th), there retention is higher with authorization type B, between week 5 to 14 particularly.

Question 4

```
date_seq <- seq(as.Date("2016-06-01"), as.Date("2016-10-30"), by=1)
df_final_q4 <- data.frame()

for (date_var in 0:151 ) {
date_week <- data.frame( week = rep(1:24, each= 7),
                   date = seq(as.Date("2016-06-01")+date_var+1, by =1 , len = 24*7))

## Creating a column of user ids for a particular date
uid_list <- signups %>%
  filter(signup_dt == as.Date("2016-06-01") + date_var) %>%
  select(uid)

## a temporary table to store cumulative first time visitor counts for a signup date
temp_table_4 <- visits %>%
  left_join(date_week, by = c("dt" = "date")) %>%
  filter(!is.na(week)) %>%
  group_by(uid, week) %>%
  summarise(visit_count = n()) %>%
  ungroup() %>%
  filter(uid %in% (uid_list$uid)) %>%
  arrange(week) %>%
  group_by(uid) %>%
  mutate(var_temp = ifelse(row_number()==1,1,0)) %>%
  ungroup() %>%
  group_by(week) %>%
  summarise(total_count = sum(var_temp)) %>%
  ungroup() %>%
  mutate(signup_dt_no = date_var) %>%
  group_by(signup_dt_no) %>%
  mutate(cum_week_count = cumsum(total_count)) %>%
  ungroup()


df_final_q4 <- rbind(df_final_q4, data.frame(temp_table_4))
}
```

Transformation from rows to columns

```
date_df <- data.frame(date = seq(as.Date("2016-06-01"), as.Date("2016-10-30"), by=1),
                         sno = seq(0, 151,by=1))

## Adding date columns using join operation
df_final_q4_1 <- df_final_q4 %>%
  left_join(date_df, by = c("signup_dt_no" = "sno")) %>%
    select(date, week, cum_week_count)

## Using spread function to transform week to columns
df_final_q4_transformed <- df_final_q4_1 %>% spread(key = week, value = cum_week_count)

## Signup counts for all signup dates
signup_count <- signups %>%
  group_by(signup_dt) %>%
  summarise (signup_count = n())

df_final_q4_transformed_1 <- df_final_q4_transformed %>%
  inner_join(signup_count,by = c("date" = "signup_dt"))
```

Convert count into %visits

```
## Transmute function creates a new column and drops the old one
df_final_q4_transformed_2 <- df_final_q4_transformed_1 %>%
      transmute( signup_date = date,
                 signup_count = signup_count,
                 perc_visit_within_week1 =`1`/signup_count,
                 perc_visit_within_week2= `2`/signup_count,
                 perc_visit_within_week3 =`3`/signup_count,
                 perc_visit_within_week4 =`4`/signup_count,
                 perc_visit_within_week5 =`5`/signup_count,
                 perc_visit_within_week6 =`6`/signup_count,
                 perc_visit_within_week7 =`7`/signup_count,
                 perc_visit_within_week8 =`8`/signup_count,
                 perc_visit_within_week9 =`9`/signup_count,
                 perc_visit_within_week10 =`10`/signup_count,
                 perc_visit_within_week11 =`11`/signup_count,
                 perc_visit_within_week12 =`12`/signup_count,
                 perc_visit_within_week13 =`13`/signup_count,
                 perc_visit_within_week14 =`14`/signup_count,
                 perc_visit_within_week15 =`15`/signup_count,
                 perc_visit_within_week16 =`16`/signup_count,
                 perc_visit_within_week17 =`17`/signup_count,
                 perc_visit_within_week18 =`18`/signup_count,
                 perc_visit_within_week19 =`19`/signup_count,
                 perc_visit_within_week20 =`20`/signup_count,
                 perc_visit_within_week21 =`21`/signup_count,
                 perc_visit_within_week22 =`22`/signup_count,
                 perc_visit_within_week23 =`23`/signup_count,
                 perc_visit_within_week24 =`24`/signup_count  )

head(df_final_q4_transformed_2)
```

```
##   signup_date signup_count perc_visit_within_week1 perc_visit_within_week2
## 1  2016-06-01          400               0.5975000               0.6825000
## 2  2016-06-02          439               0.6674260               0.7266515
## 3  2016-06-03          407               0.6805897               0.7690418
## 4  2016-06-04          436               0.6720183               0.7500000
## 5  2016-06-05          540               0.6240741               0.6833333
## 6  2016-06-06          463               0.6328294               0.7105832
##   perc_visit_within_week3 perc_visit_within_week4 perc_visit_within_week5
## 1               0.7150000               0.7475000               0.7700000
## 2               0.7744875               0.7858770               0.7995444
## 3               0.7985258               0.8230958               0.8329238
## 4               0.7775229               0.8096330               0.8233945
## 5               0.7148148               0.7425926               0.7759259
## 6               0.7602592               0.7861771               0.8056156
##   perc_visit_within_week6 perc_visit_within_week7 perc_visit_within_week8
## 1               0.7825000               0.7925000               0.8075000
## 2               0.8063781               0.8154897               0.8177677
## 3               0.8402948               0.8452088               0.8525799
## 4               0.8279817               0.8325688               0.8371560
## 5               0.7907407               0.8037037               0.8037037
## 6               0.8120950               0.8185745               0.8336933
##   perc_visit_within_week9 perc_visit_within_week10
## 1               0.8100000                0.8200000
## 2               0.8246014                0.8268793
## 3               0.8550369                0.8550369
## 4               0.8463303                0.8532110
## 5               0.8166667                0.8185185
## 6               0.8401728                0.8444924
##   perc_visit_within_week11 perc_visit_within_week12
## 1                0.8225000                0.8325000
## 2                0.8337130                0.8382688
## 3                0.8599509                0.8624079
## 4                0.8646789                0.8669725
## 5                0.8277778                0.8333333
## 6                0.8509719                0.8574514
##   perc_visit_within_week13 perc_visit_within_week14
## 1                0.8325000                0.8325000
## 2                0.8405467                0.8405467
## 3                0.8624079                0.8624079
## 4                0.8715596                0.8784404
## 5                0.8388889                0.8444444
## 6                0.8617711                0.8639309
##   perc_visit_within_week15 perc_visit_within_week16
## 1                0.8325000                0.8350000
## 2                0.8451025                0.8542141
## 3                0.8624079                0.8648649
## 4                0.8784404                0.8830275
## 5                0.8462963                0.8500000
## 6                0.8660907                0.8660907
##   perc_visit_within_week17 perc_visit_within_week18
## 1                0.8375000                0.8400000
## 2                0.8542141                0.8587699
## 3                0.8648649                0.8648649
## 4                0.8853211                0.8899083
## 5                0.8574074                0.8592593
## 6                0.8660907                0.8682505
##   perc_visit_within_week19 perc_visit_within_week20
## 1                0.8425000                0.8425000
## 2                0.8610478                0.8633257
## 3                0.8648649                0.8697789
## 4                0.8922018                0.8944954
## 5                0.8611111                0.8629630
## 6                0.8682505                0.8682505
##   perc_visit_within_week21 perc_visit_within_week22
## 1                0.8450000                0.8450000
## 2                0.8633257                0.8656036
## 3                0.8697789                0.8697789
## 4                0.8944954                0.8944954
## 5                0.8629630                0.8629630
## 6                0.8682505                0.8682505
##   perc_visit_within_week23 perc_visit_within_week24
## 1                0.8450000                0.8450000
## 2                0.8656036                0.8656036
## 3                0.8722359                0.8722359
## 4                0.8944954                0.8990826
## 5                0.8666667                0.8666667
## 6                0.8682505                0.8725702
```

Proportion of users on average that do not come back

```
1- mean(df_final_q4_transformed_2$perc_visit_within_week24)
```

```
## [1] 0.1338611
```

- 13.4 % of the users never come back after signing up (within first 24 weeks)