Facebook Analysis

Pratik Agrawal

21 December 2019

1. How would you describe the structure of the dataset?

using str() function. Dataset is structured, having 15 columns. out of which 13 columns have numerical data while 'Gender' Column is categorical i.e. it contains factors.

```
users = read.csv("pseudo_facebook.csv")
str(users)
```

```
## 'data.frame':    99003 obs. of  15 variables:
##  $ userid               : int  2094382 1192601 2083884 1203168 1733186 1524765 1136133 1680361 13651
##  $ age                  : int  14 14 14 14 14 14 13 13 13 13 ...
##  $ dob_day              : int  19 2 16 25 4 1 14 4 1 2 ...
##  $ dob_year             : int  1999 1999 1999 1999 1999 1999 2000 2000 2000 2000 ...
##  $ dob_month            : int  11 11 11 12 12 12 1 1 1 2 ...
##  $ gender               : Factor w/ 2 levels "female","male": 2 1 2 1 2 2 2 1 2 2 ...
##  $ tenure               : int  266 6 13 93 82 15 12 0 81 171 ...
##  $ friend_count         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ friendships_initiated: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ likes                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ likes_received       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mobile_likes         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ mobile_likes_received: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ www_likes            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ www_likes_received   : int  0 0 0 0 0 0 0 0 0 0 ...
```

2. How many missing values are in the data set?

There are 177 missing values in the dataset. if we simply do the summation of NA values of the dataset, it yields 177 as result. this means there are 177 rows for which one or more column of these rows has missing values. see the command below.

```
sum(is.na(users))
```

```
## [1] 177
```

3.Which variables are the missing values concentrated in? (Hint: apply or colSums function may be helpful) Make a decision on how you treat missing values. Reason your choice.

gender aand tenure variables are missing the respectively 175 and 2 values.

I have created new dataframe which has all rows without missing value rows. I would omit all these 'NA' rows. The reason being ommiting ### 177 rows from dataset of close to 1,00,000 rows would not have substancial effect on values of aggregate functions as well as Visualization.

Other reason being we can not simply substitute the categorical values like gender for the missing values. for numerically missing values it is possible to substitute these missing values by mean or median of that variable.

```
colSums(is.na(users))
```

```
##                userid                   age               dob_day
##                     0                     0                     0
##              dob_year             dob_month                gender
##                     0                     0                   175
##                tenure          friend_count friendships_initiated
##                     2                     0                     0
##                 likes        likes_received          mobile_likes
##                     0                     0                     0
## mobile_likes_received             www_likes    www_likes_received
##                     0                     0                     0
```

```
users_wona <- na.omit(users)
```

4. Examine basic descriptives of the dataset using summary() or describe() function.

```
summary(users)
```

```
##      userid             age            dob_day         dob_year
##  Min.   :1000008   Min.   : 13.00   Min.   : 1.00   Min.   :1900
##  1st Qu.:1298806   1st Qu.: 20.00   1st Qu.: 7.00   1st Qu.:1963
##  Median :1596148   Median : 28.00   Median :14.00   Median :1985
##  Mean   :1597045   Mean   : 37.28   Mean   :14.53   Mean   :1976
##  3rd Qu.:1895744   3rd Qu.: 50.00   3rd Qu.:22.00   3rd Qu.:1993
##  Max.   :2193542   Max.   :113.00   Max.   :31.00   Max.   :2000
##
##    dob_month         gender          tenure        friend_count
##  Min.   : 1.000   female:40254   Min.   :   0.0   Min.   :   0.0
##  1st Qu.: 3.000   male  :58574   1st Qu.: 226.0   1st Qu.:  31.0
##  Median : 6.000   NA's  :  175   Median : 412.0   Median :  82.0
##  Mean   : 6.283                  Mean   : 537.9   Mean   : 196.4
##  3rd Qu.: 9.000                  3rd Qu.: 675.0   3rd Qu.: 206.0
##  Max.   :12.000                  Max.   :3139.0   Max.   :4923.0
```

```
##                                  NA's   :2
##  friendships_initiated     likes          likes_received      mobile_likes
##  Min.   :   0.0        Min.   :    0.0   Min.   :     0.0   Min.   :    0.0
##  1st Qu.:  17.0        1st Qu.:    1.0   1st Qu.:     1.0   1st Qu.:    0.0
##  Median :  46.0        Median :   11.0   Median :     8.0   Median :    4.0
##  Mean   : 107.5        Mean   :  156.1   Mean   :   142.7   Mean   :  106.1
##  3rd Qu.: 117.0        3rd Qu.:   81.0   3rd Qu.:    59.0   3rd Qu.:   46.0
##  Max.   :4144.0        Max.   :25111.0   Max.   :261197.0   Max.   :25111.0
##
##  mobile_likes_received   www_likes         www_likes_received
##  Min.   :     0.00     Min.   :    0.00   Min.   :     0.00
##  1st Qu.:     0.00     1st Qu.:    0.00   1st Qu.:     0.00
##  Median :     4.00     Median :    0.00   Median :     2.00
##  Mean   :    84.12     Mean   :   49.96   Mean   :    58.57
##  3rd Qu.:    33.00     3rd Qu.:    7.00   3rd Qu.:    20.00
##  Max.   :138561.00     Max.   :14865.00   Max.   :129953.00
##
```

5. How can you characterize the sample in terms of gender?

the given sample has both genders i.e. Male and Female distributed close to in ~60% and ~40% ratio.

```
summary(users$gender)
```

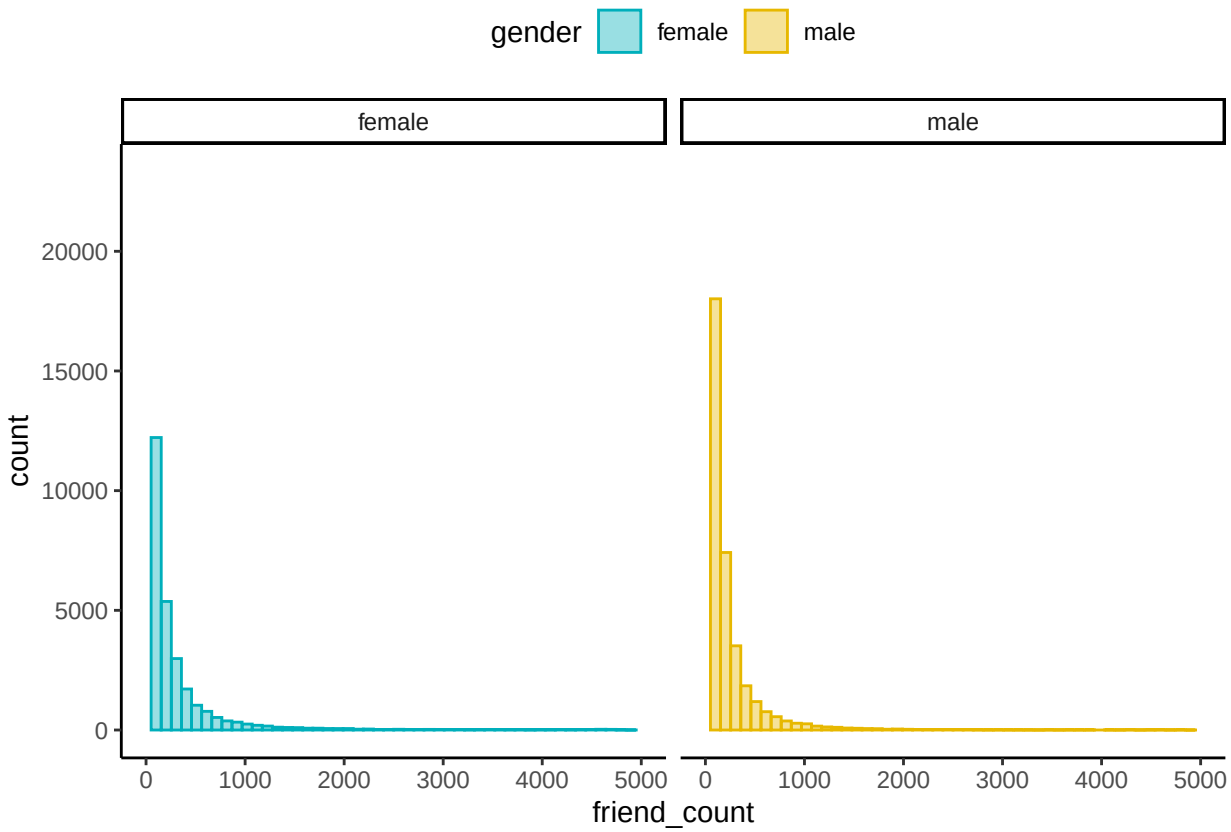
```
## female   male   NA's
##  40254  58574    175
```

a. Explore the distribution of the friend count for both female and male users. For what gender, the average number of likes is higher?

From the below plots and results, it is clear that the Female users have on average more number of friends.

```
library(ggplot2)
theme_set(
  theme_classic() +
    theme(legend.position = "top")
  )
ggplot(users_wona, aes(x = friend_count)) +
  geom_histogram(aes(color = gender, fill = gender),
                 position = "identity", bins = 50, alpha = 0.4) +
  facet_wrap(~gender)+
  scale_x_continuous(limits = c(0,5000), breaks = seq(0,5000,1000)) +
  scale_color_manual(values = c("#00AFBB", "#E7B800")) +
  scale_fill_manual(values = c("#00AFBB", "#E7B800"))
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```




```
table(users_wona$gender)
```

```
##
## female   male
##  40252  58574
```

```
by(users_wona$friend_count,users_wona$gender,summary)
```

```
## users_wona$gender: female
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0      37      96     242     244    4923
## ------------------------------------------------------------
## users_wona$gender: male
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0      27      74     165     182    4917
```

if we talk about the 'likes_received' variable, For female users Mean and Median is quite high compared to male users. See the results below,

```
by(users_wona$likes_received,users_wona$gender,summary)
```

```
## users_wona$gender: female
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      0.0      3.0     29.0    251.4    153.0 261197.0
## ------------------------------------------------------------
## users_wona$gender: male
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.00     0.00     4.00    67.91    25.00 82623.00
```

b. Which gender has likes by using web platform the most? Base your inference on the average number of likes.

Female users have received more likes from web platform. we can find this out by using aggregate function mean on variable gender.

```
#average likes received by gender
aggregate(users_wona[, "www_likes_received"], list(users_wona$gender), mean)
```

```
##   Group.1         x
## 1  female 104.33827
## 2    male  27.07853
```

```
#by(users_wona$www_likes_received,users_wona$gender,summary)
```

c. Which gender has likes by using the mobile application the most? Base your inference on the average number of likes.

Female users have received more likes from the mobile application. we can find this out by using aggregate function mean on variable gender.

```
#average likes received by gender
aggregate(users_wona[, "mobile_likes_received"], list(users_wona$gender), mean)
```

```
##   Group.1         x
## 1  female 147.10760
## 2    male  40.83301
```

```
#by(users_wona$mobile_likes_received,users_wona$gender,summary)
```

d. On average, who initiated more friendships in our sample: men or women?

Women. See the result below.

```
#average likes received by gender
aggregate(users_wona[, "friendships_initiated"], list(users_wona$gender), mean)
```

```
##   Group.1        x
## 1  female 113.9024
## 2    male 103.0666
```

```
#by(users_wona$friendships_initiated,users_wona$gender,summary)
```

6. Check if the Facebook mobile app is really beneficial for the company. Compare the type of device used for sending and receiving likes. What share of users receives/sends likes on a mobile device? What share of users interacts via the web?

close to 64% of likes are sent/received by mobile app, so mobile app is really important for the company.

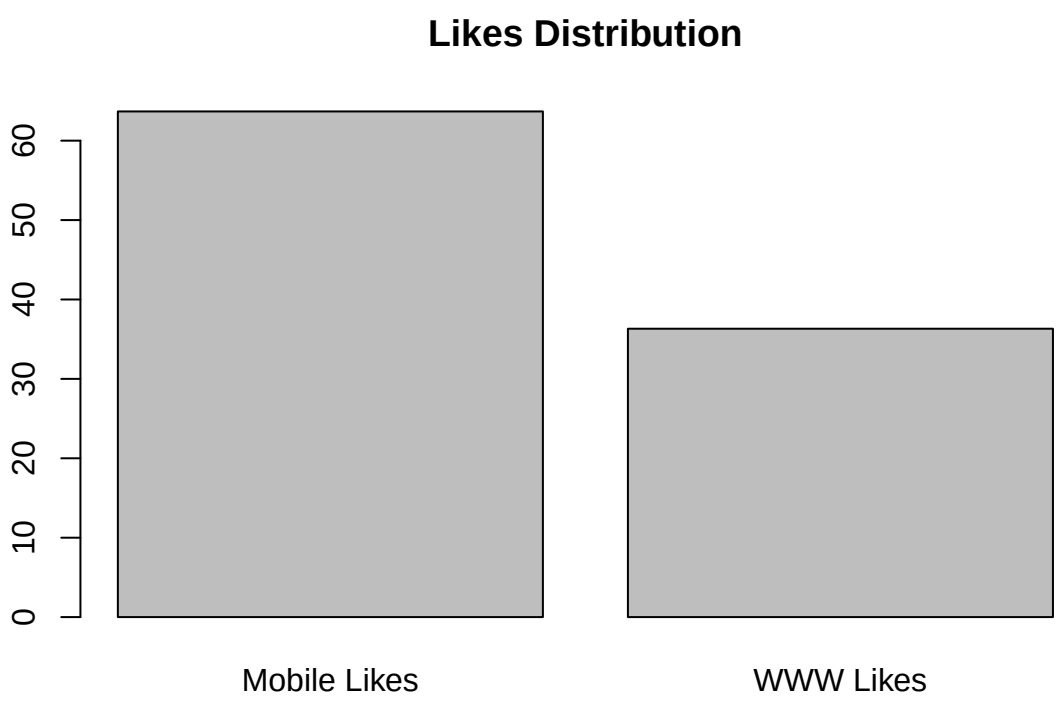
```
total_likes <- sum(users_wona$likes,users_wona$likes_received)
percent_mobile_likes <- (sum(users_wona$mobile_likes,users_wona$mobile_likes_received) / total_likes) *
percent_www_likes <- (sum(users_wona$www_likes,users_wona$www_likes_received)  / total_likes) * 100

print(percent_mobile_likes )
```

```
## [1] 63.6818
```

```
print(percent_www_likes )
```

```
## [1] 36.31817
```

```
barplot(c(percent_mobile_likes,percent_www_likes), main="Likes Distribution",
   xlab="Percentage of Likes",names.arg=c("Mobile Likes", "WWW Likes"))
```



below code creates new categorical column named 'user_type', we would use existing columns 'www_likes' and 'mobile_likes' to identify wheather user has used mobile app, website or both. is user has give 0 likes from both of these mediums, we count that user as 'none' type of user.

```
user_type <- function(mobile_likes,www_likes){
  user_type <- "none"
   if(mobile_likes > 0 && www_likes > 0)
   {
      user_type<- "both"
    }
  else if (www_likes > 0)
  {  user_type<- "www"
  }
  else if (mobile_likes > 0)
  {
    user_type <-"mobile"
  }
  else { user_type <- "none"}
  return(user_type)
}
users_wona$user_type <- mapply(user_type,users_wona$mobile_likes,users_wona$www_likes)
head(users_wona$user_type)
```

```
## [1] "none" "none" "none" "none" "none" "none"
```

```
head(users_wona)
```

```
##    userid age dob_day dob_year dob_month gender tenure friend_count
## 1 2094382  14      19     1999        11   male    266            0
## 2 1192601  14       2     1999        11 female      6            0
## 3 2083884  14      16     1999        11   male     13            0
## 4 1203168  14      25     1999        12 female     93            0
## 5 1733186  14       4     1999        12   male     82            0
## 6 1524765  14       1     1999        12   male     15            0
##   friendships_initiated likes likes_received mobile_likes mobile_likes_received
## 1                     0     0              0            0                      0
## 2                     0     0              0            0                      0
## 3                     0     0              0            0                      0
## 4                     0     0              0            0                      0
## 5                     0     0              0            0                      0
## 6                     0     0              0            0                      0
##   www_likes www_likes_received user_type
## 1         0                  0      none
## 2         0                  0      none
## 3         0                  0      none
## 4         0                  0      none
## 5         0                  0      none
## 6         0                  0      none
```

count the number of users by newly created user_type

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

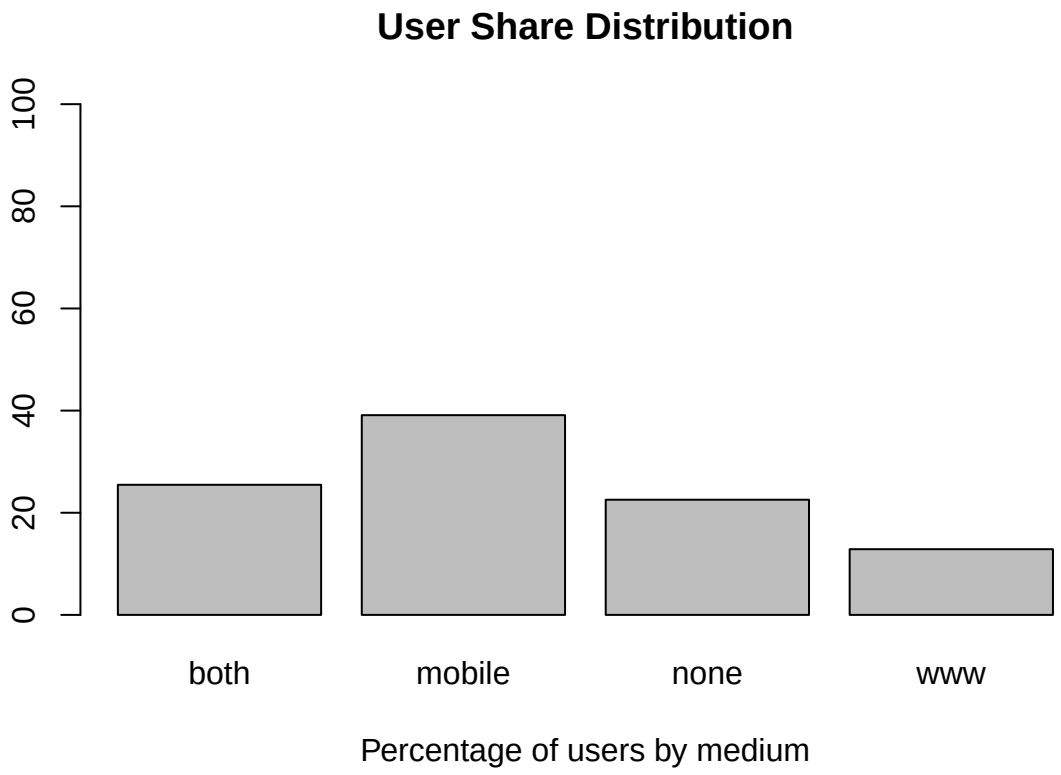
```
pf.fc_by_user_type <-  users_wona %>% group_by(user_type) %>%summarise( Count= n())
head(pf.fc_by_user_type)
```

```
## # A tibble: 4 x 2
##   user_type Count
##   <chr>     <int>
## 1 both      25174
## 2 mobile    38650
## 3 none      22285
## 4 www       12717
```

above calculation tells us that, almost 25% of users use both mobile and www for liking stuff on facebook.

while approx. 39% users are Mobile only users. ~13% users use www only.

```
barplot((pf.fc_by_user_type$Count/sum(pf.fc_by_user_type$Count))*100, main="User Share Distribution",
   xlab="Percentage of users by medium",names.arg=pf.fc_by_user_type$user_type,ylim = c(0,100))
```

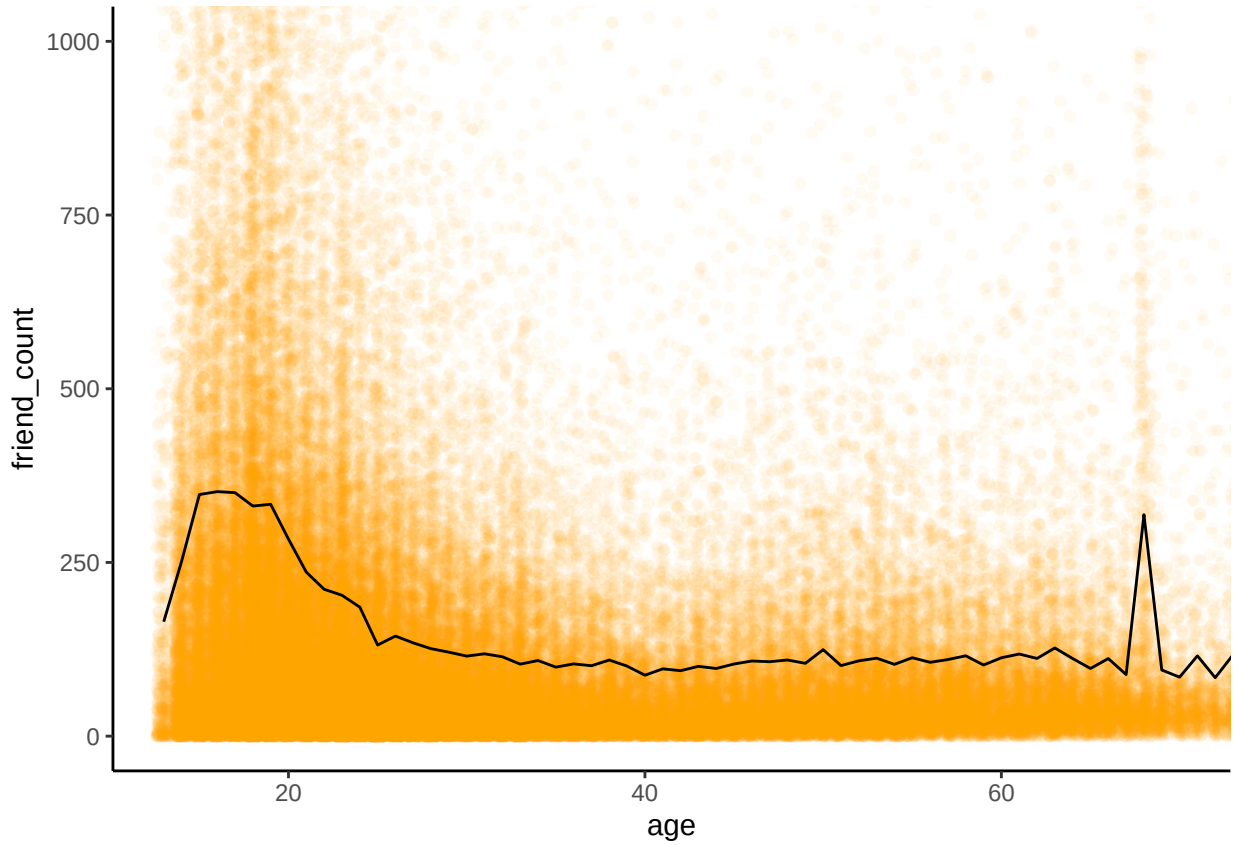


7. Explore the number of friends by gender and age. What inferences can you make?

first let's plot mean of friend_count by age, you can see in below plot,

- for the age group 13 to 20, friends lies between 150 to 375 on average.
- There is a unusual splike in mean friends between age 65 and 70.
- for the age group 25 to 60, mean friends lie near 125 mostly.

```
ggplot(data=users_wona,aes(age, friend_count)) +
  coord_cartesian(xlim = c(13,70), ylim = c(0,1000)) +
  geom_point(alpha = 0.05, position=position_jitter(h = 0), color = 'orange') +
  geom_line(stat = 'summary', fun.y = mean)
```



using dplyr package, we can create dataframes by applying aggregate functions like mean, median etc. on variables against other variables of the dataset.

e.g. below we have applied aggregate functions like mean, median on 'friend_count' against the variable 'age'.

these means, we get mean and median for each and every age in our dataset.

```
#install.packages("dplyr")
library(dplyr)
pf.fc_by_age <- users_wona %>% group_by(age) %>% summarise(friend_count_mean = mean(friend_count), frie
head(pf.fc_by_age)
```

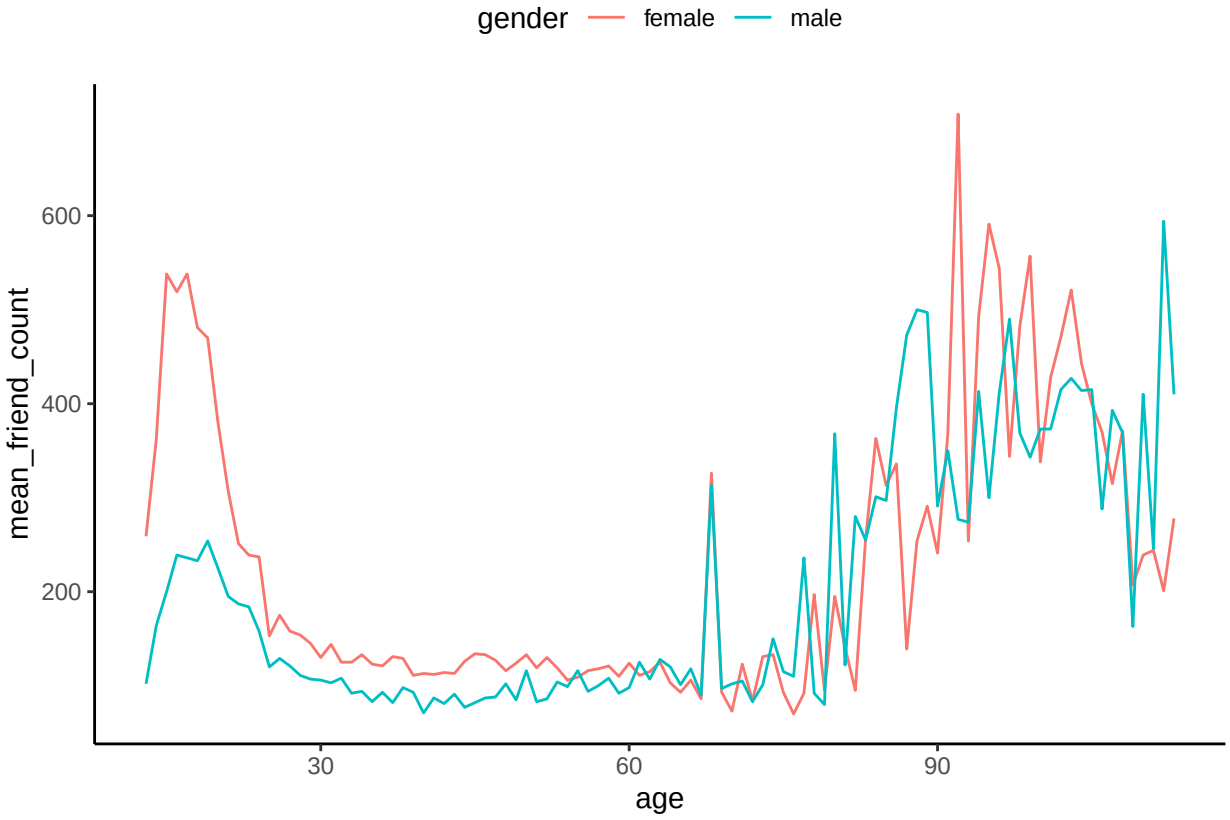
```
## # A tibble: 6 x 4
##     age friend_count_mean friend_count_median     N
##   <int>             <dbl>               <dbl> <int>
## 1    13              165.                  74   484
## 2    14              251.                 132  1925
## 3    15              348.                 161  2617
## 4    16              352.                 172. 3086
## 5    17              350.                 156  3281
## 6    18              331.                 162  5196
```

```
library(dplyr)
pf.fc_by_age_gender <-  users_wona %>% filter(!is.na(gender)) %>% group_by(age,gender) %>% summarise(mea
head(pf.fc_by_age_gender)
```

```
## # A tibble: 6 x 5
## # Groups:   age [3]
##     age gender mean_friend_count median_friend_count     n
##   <int> <fct>              <int>               <int> <int>
## 1    13 female               259                 148   193
## 2    13 male                 102                  55   291
## 3    14 female               362                 224   847
## 4    14 male                 164                  92  1078
## 5    15 female               538                 276  1139
## 6    15 male                 200                 106  1478
```

```
ggplot(data = filter(pf.fc_by_age_gender, !is.na(gender)), mapping = aes(x = age, y = mean_friend_count)
geom_line(aes(color = gender)) +
stat_summary(fun.y = mean)
```

```
## Warning: Removed 101 rows containing missing values (geom_pointrange).
```



few inferences, * on average, female users have more friends till the age of 60. * after the age 60, both male and female users have more or less same number of average friends. * from age 15 to 20, Female users have as many as twice friends on average compared to male users.

8. What users have higher friend count: people in older join (tenure) or those who joined later? Interpret the variable "tenure" as the number of days a user is registered on Facebook. Assume that those who use Facebook at least 500 days belong to an "older join" category and those who use Facebook less than 500 days belong to the "later join" category.

People who joined early, i.e. whose tenure is greater than 500 days have higher friend count on average and by median. 275 compared to 144 Mean friend_count.

```
create_join_type <- function(tenure){
   if(tenure < 500)
   {
     lj <- "later_join"
     return(lj) }
  else
  {  oj <- "older_join"
  return(oj)
  }
}
users_wona$Tenure_type <- mapply(create_join_type,users_wona$tenure)
```

```
head(users_wona$Tenure_type)
```

```
## [1] "later_join" "later_join" "later_join" "later_join" "later_join"
## [6] "later_join"
```

```
library(dplyr)
pf.fc_by_tenure_count <-  users_wona  %>% group_by(Tenure_type) %>% summarise(mean_friend_count = as.in
head(pf.fc_by_tenure_count)
```

```
## # A tibble: 2 x 4
##   Tenure_type mean_friend_count median_friend_count     n
##   <chr>                   <int>               <int> <int>
## 1 later_join                144                  64 59632
## 2 older_join                275                 122 39194
```

this code is just to visualize the tenure and friend_count

```
library(dplyr)
pf.fc_by_tenure <-  users_wona  %>% group_by(Tenure_type,tenure) %>% summarise(mean_friend_count = as.i
head(pf.fc_by_tenure)
```

```
## # A tibble: 6 x 5
## # Groups:   Tenure_type [1]
##   Tenure_type tenure mean_friend_count median_friend_count     n
##   <chr>        <int>             <int>               <int> <int>
## 1 later_join       0                 2                   0    70
## 2 later_join       1                 5                   1    60
```

```
## 3 later_join       2                 35                   5    72
## 4 later_join       3                 16                   5    79
## 5 later_join       4                 16                   7    86
## 6 later_join       5                 24                   8    92
```

```
ggplot(data = pf.fc_by_tenure, mapping = aes(x = tenure, y = mean_friend_count)) +
  coord_cartesian(xlim = c(10,3300), ylim = c(0,5000)) +
geom_line(aes(color = Tenure_type)) +
stat_summary(fun.y = mean)
```

```
## Warning: Removed 2418 rows containing missing values (geom_pointrange).
```