CSE5DEV_A2

Nikhil

2023-10-20

Task 1

Step 1: Loading the Dataset

```
library(dplyr)
```

```
##
## 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
```

```
library(tidyr)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.4
## ✓ ggplot2   3.4.1     ✓ stringr   1.5.0
## ✓ lubridate 1.9.2     ✓ tibble    3.2.0
## ✓ purrr     1.0.2
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
#Change the working directory as per the file location
setwd("D:/Latrobe_MDS/Sem 3 (Sem 2 2023)/CSE5DEV/Labs/A2")

dat_T1 <- read.csv("Studentmarks.csv", header = TRUE)
```

Step 2: Inspecting the Dataset

```
head(dat_T1)
```

```
##   StudentID Studentname        dob X2020 X2021 X2022
## 1         1        Anna 12/12/1998    75    78    85
## 2         2       James 10/08/1999    65    74    52
## 3         3        Mary 21/08/1998    56    68    90
## 4         4      Antony 02/05/1999    78    65    85
## 5         5       Jacob 22/07/1998    85    78    65
## 6         6     Angelin 11/05/1998    65    68    70
```

```
str(dat_T1)
```

```
## 'data.frame':    10 obs. of  6 variables:
##  $ StudentID  : int  1 2 3 4 5 6 7 8 9 10
##  $ Studentname: chr  "Anna" "James" "Mary" "Antony" ...
##  $ dob        : chr  "12/12/1998" "10/08/1999" "21/08/1998" "02/05/1999" ...
##  $ X2020      : int  75 65 56 78 85 65 78 77 90 95
##  $ X2021      : int  78 74 68 65 78 68 52 72 87 85
##  $ X2022      : int  85 52 90 85 65 70 88 75 88 75
```

```
dim(dat_T1)
```

```
## [1] 10  6
```

```
#Checking for NAs
any(is.na(dat_T1))
```

```
## [1] FALSE
```

```
#Checking for duplicated rows
any(duplicated(dat_T1))
```

```
## [1] FALSE
```

Task 1: Part a

Step 3: Calculating age1 and binding it as a column to the dataframe. Using floor() as the age is supposed to be in years.

```
#Used floor() as we need to round down to get age in absolute years
age1 <- floor(as.numeric(difftime(Sys.Date(), as.Date(dat_T1$dob, format = "%d/%m/%Y")))/365)
head(age1)
```

```
## [1] 25 24 25 24 25 25
```

```
#Binding age1 as a column to the dataset
dat_T1 <- cbind(dat_T1, age1)
head(dat_T1)
```

```
##   StudentID Studentname        dob X2020 X2021 X2022 age1
## 1         1        Anna 12/12/1998    75    78    85   25
## 2         2       James 10/08/1999    65    74    52   24
## 3         3        Mary 21/08/1998    56    68    90   25
## 4         4      Antony 02/05/1999    78    65    85   24
## 5         5       Jacob 22/07/1998    85    78    65   25
## 6         6     Angelin 11/05/1998    65    68    70   25
```

Task 1: Part b

Step 4: Splitting the “dob” column into month and year columns

```
#Converting the dob column to the date data type
dat_T1$dob <- as.Date(dat_T1$dob, format = "%d/%m/%Y")
str(dat_T1)
```

```
## 'data.frame':    10 obs. of  7 variables:
##  $ StudentID  : int  1 2 3 4 5 6 7 8 9 10
##  $ Studentname: chr  "Anna" "James" "Mary" "Antony" ...
##  $ dob        : Date, format: "1998-12-12" "1999-08-10" ...
##  $ X2020      : int  75 65 56 78 85 65 78 77 90 95
##  $ X2021      : int  78 74 68 65 78 68 52 72 87 85
##  $ X2022      : int  85 52 90 85 65 70 88 75 88 75
##  $ age1       : num  25 24 25 24 25 25 24 25 25 25
```

```
#Extracting month and year column from dob
dat_T1 <- separate(dat_T1, dob, c("Year", "Month"))
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 10 rows [1, 2, 3, 4, 5, 6, 7,
## 8, 9, 10].
```

```
head(dat_T1)
```

```
##   StudentID Studentname Year Month X2020 X2021 X2022 age1
## 1         1        Anna 1998    12    75    78    85   25
## 2         2       James 1999    08    65    74    52   24
## 3         3        Mary 1998    08    56    68    90   25
## 4         4      Antony 1999    05    78    65    85   24
## 5         5       Jacob 1998    07    85    78    65   25
## 6         6     Angelin 1998    05    65    68    70   25
```

Step 5: Calculating age2 and including it as a column in dataframe

```
dat_T1$age2 <- as.numeric(format(Sys.Date(), "%Y"))-as.numeric(dat_T1$Year)
head(dat_T1)
```

```
##   StudentID Studentname Year Month X2020 X2021 X2022 age1 age2
## 1         1        Anna 1998    12    75    78    85   25   26
## 2         2       James 1999    08    65    74    52   24   25
## 3         3        Mary 1998    08    56    68    90   25   26
## 4         4      Antony 1999    05    78    65    85   24   25
## 5         5       Jacob 1998    07    85    78    65   25   26
## 6         6     Angelin 1998    05    65    68    70   25   26
```

Task 1: Part c

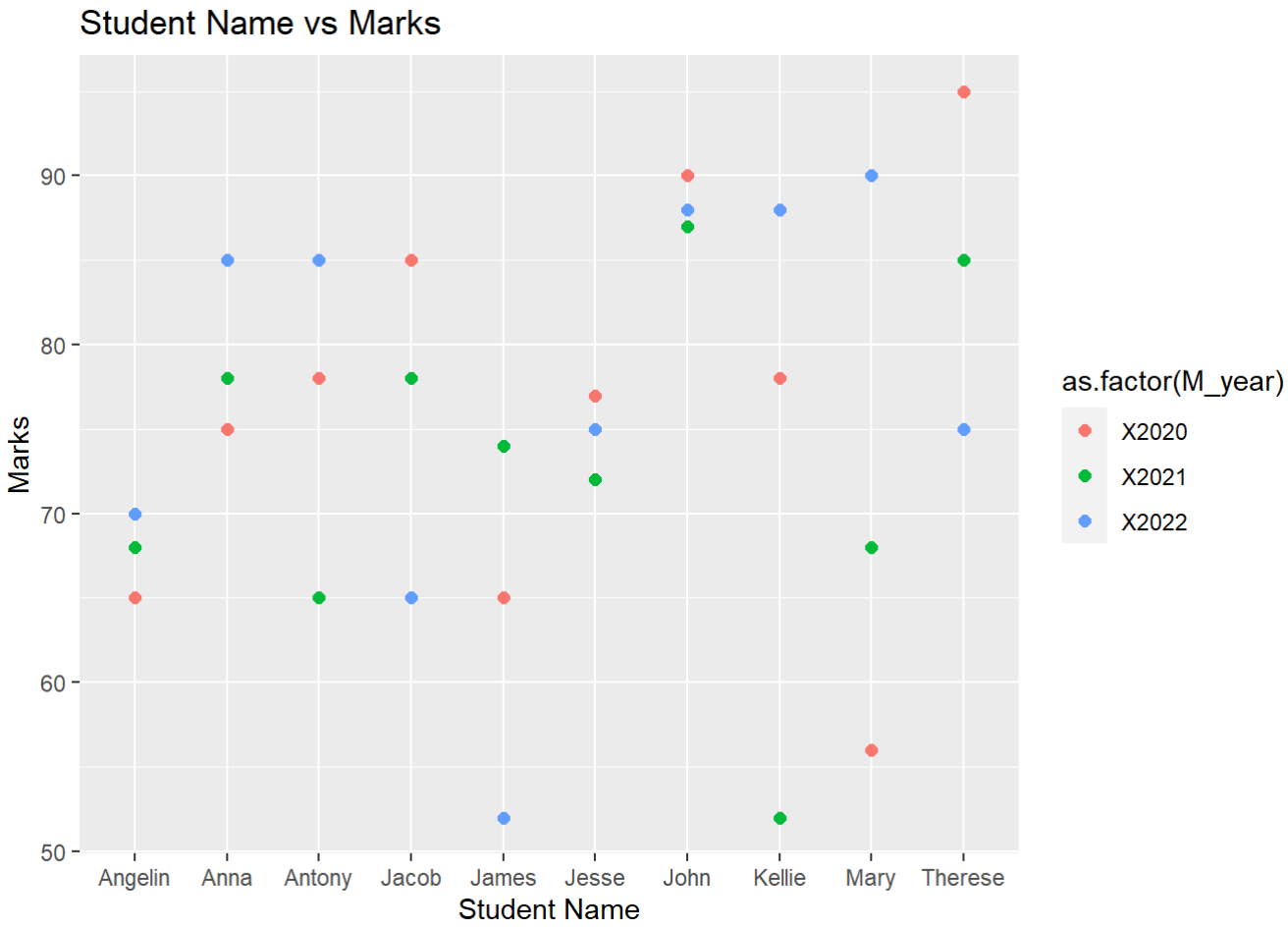
Step 6: Using gather() function to collect marks in a single column

```
dat_T1.2 <- gather(dat_T1, "M_year", "Marks", 5:7)
head(dat_T1.2)
```

```
##   StudentID Studentname Year Month age1 age2 M_year Marks
## 1         1        Anna 1998    12   25   26  X2020    75
## 2         2       James 1999    08   24   25  X2020    65
## 3         3        Mary 1998    08   25   26  X2020    56
## 4         4      Antony 1999    05   24   25  X2020    78
## 5         5       Jacob 1998    07   25   26  X2020    85
## 6         6     Angelin 1998    05   25   26  X2020    65
```

Step 7: Plotting “Studentname” vs “Marks” using “Year” as color

```
library(ggplot2)
ggplot(dat_T1.2, aes(Studentname, Marks, color = as.factor(M_year))) +
  geom_point(size = 2) +
  labs(x = "Student Name", y = "Marks", title = "Student Name vs Marks")
```



Interpretation:

From the above chart, we can say that Anna, Kellie and Mary has shows significant improvement in their performance, whereas, Jacob and Therese has shown significant decline in their performance.

Task 1: Part d

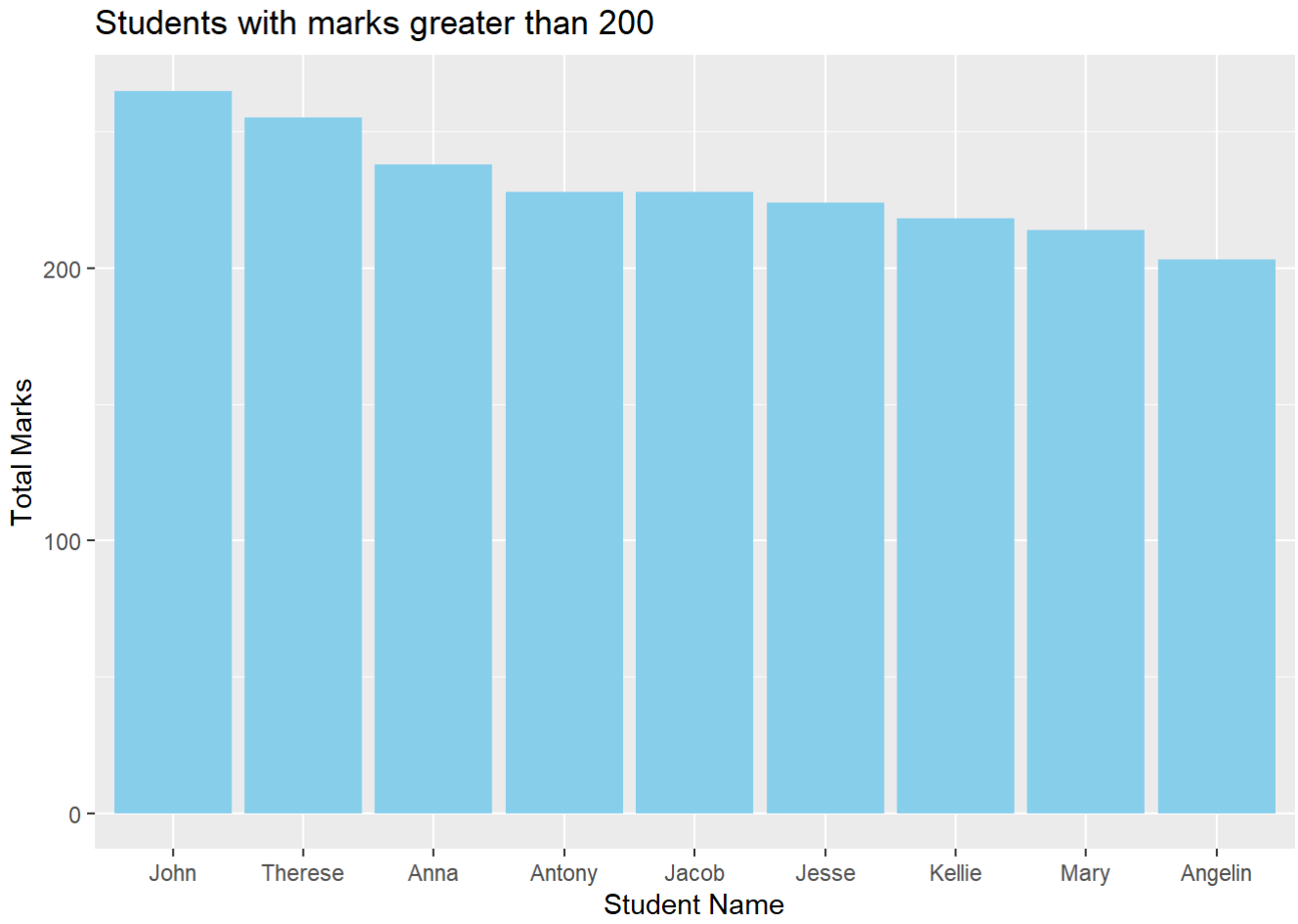
Step 8: Calculating Total Marks and filtering the data

```
dat_T1.3 <- dat_T1 %>%
  mutate(TotalMarks = X2020 + X2021 + X2022) %>%
  filter(TotalMarks >= 200) %>%
  arrange(desc(TotalMarks))

nrow(dat_T1.3) #Only 1 student has been dropped for marks < 200
```

```
## [1] 9
```

Step 9: Plotting Bar Chart is descending order of marks

```
#Load if not done before
#library(ggplot2)

ggplot(dat_T1.3, aes(x = reorder(Studentname, -TotalMarks), y = TotalMarks)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(x = "Student Name", y = "Total Marks",
       title = "Students with marks greater than 200")
```



Task 2

Step 1: Loading the Dataset

```
#Load if not done before
#library(dplyr)
#library(tidyr)
#library(tidyverse)

##Change the working directory as per the file location
setwd("D:/Latrobe_MDS/Sem 3 (Sem 2 2023)/CSE5DEV/Labs/A2")

dat_T2 <- read.csv("Movies.csv", header = TRUE)
```

Step 2: Inspecting the Dataset

```
head(dat_T2)
```

```
##   Color          Director Reviews Duration Director_facebook_likes
## 1 Color     James Cameron     723      178                      0
## 2 Color    Gore Verbinski     302      169                    563
## 3 Color        Sam Mendes     602      148                      0
## 4 Color Christopher Nolan     813      164                  22000
## 5 Color    Andrew Stanton     462      132                    475
## 6 Color         Sam Raimi     392      156                      0
##   Actor_3_facebook_likes     Actor_2_name Actor_1_facebook_likes     Gross
## 1                    855 Joel David Moore                   1000 760505847
## 2                   1000    Orlando Bloom                  40000 309404152
## 3                    161     Rory Kinnear                  11000 200074175
## 4                  23000   Christian Bale                  27000 448130642
## 5                    530  Samantha Morton                    640  73058679
## 6                   4000     James Franco                  24000 336530303
##    Genre    Actor_1_name                                    Title   Votes
## 1 Action     CCH Pounder                                  Avatara  886204
## 2 Action     Johnny Depp Pirates of the Caribbean: At World's Enda  471220
## 3 Action Christoph Waltz                                Spectrea  275868
## 4 Action       Tom Hardy                  The Dark Knight Risesa 1144337
## 5 Action    Daryl Sabara                               John Cartera  212204
## 6 Action    J.K. Simmons                              Spider-Man 3a  383056
##   Cast_total_facebook_likes         Actor_3_name Facenumber_in_poster
## 1                      2791            Wes Studi                    0
## 2                     46563       Jack Davenport                    0
## 3                     11554     Stephanie Sigman                    1
## 4                     95000 Joseph Gordon-Levitt                    0
## 5                      2277         Polly Walker                    1
## 6                     39000        Kirsten Dunst                    0
##                                          Plot_keywords
## 1                       avatar|future|marine|native|paraplegic
## 2    goddess|marriage ceremony|marriage proposal|pirate|singapore
## 3                          bomb|espionage|sequel|spy|terrorist
## 4 deception|imprisonment|lawlessness|police officer|terrorist plot
## 5               alien|american civil war|male nipple|mars|princess
## 6                       sandman|spider man|symbiote|venom|villain
##                                    Movie_imdb_link Language Content_rating
## 1 http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1  English          PG-13
## 2 http://www.imdb.com/title/tt0449088/?ref_=fn_tt_tt_1  English
## 3 http://www.imdb.com/title/tt2379713/?ref_=fn_tt_tt_1  English          PG-13
## 4 http://www.imdb.com/title/tt1345836/?ref_=fn_tt_tt_1  English          PG-13
## 5 http://www.imdb.com/title/tt0401729/?ref_=fn_tt_tt_1  English          PG-13
## 6 http://www.imdb.com/title/tt0413300/?ref_=fn_tt_tt_1  English          PG-13
##      Budget Year Actor_2_facebook_likes Imdb_score Aspect_ratio
## 1 237000000 2009                   936        7.9         1.78
## 2 300000000 2007                  5000        7.1         2.35
## 3 245000000 2015                   393        6.8         2.35
## 4 250000000 2012                 23000        8.5         2.35
## 5 263700000 2012                   632        6.6         2.35
## 6 258000000 2007                 11000        6.2         2.35
##   Movie_facebook_likes
## 1                33000
## 2                    0
## 3                85000
## 4               164000
```

```
## 5                24000
## 6                    0
```

```
str(dat_T2)
```

```
## 'data.frame':    3891 obs. of  26 variables:
##  $ Color                    : chr  "Color" "Color" "Color" "Color" ...
##  $ Director                 : chr  "James Cameron" "Gore Verbinski" "Sam Mendes" "Christop
her Nolan" ...
##  $ Reviews                  : int  723 302 602 813 462 392 324 635 375 673 ...
##  $ Duration                 : int  178 169 148 164 132 156 100 141 153 183 ...
##  $ Director_facebook_likes  : int  0 563 0 22000 475 0 15 0 282 0 ...
##  $ Actor_3_facebook_likes   : int  855 1000 161 23000 530 4000 284 19000 10000 2000 ...
##  $ Actor_2_name             : chr  "Joel David Moore" "Orlando Bloom" "Rory Kinnear" "Chri
stian Bale" ...
##  $ Actor_1_facebook_likes   : int  1000 40000 11000 27000 640 24000 799 26000 25000 15000
...
##  $ Gross                    : int  760505847 309404152 200074175 448130642 73058679 336530
303 200807262 458991599 301956980 330249062 ...
##  $ Genre                    : chr  "Action" "Action" "Action" "Action" ...
##  $ Actor_1_name             : chr  "CCH Pounder" "Johnny Depp" "Christoph Waltz" "Tom Hard
y" ...
##  $ Title                    : chr  "Avatara" "Pirates of the Caribbean: At World's Enda"
"Spectrea" "The Dark Knight Risesa" ...
##  $ Votes                    : int  886204 471220 275868 1144337 212204 383056 294810 46266
9 321795 371639 ...
##  $ Cast_total_facebook_likes: int  2791 46563 11554 95000 2277 39000 1651 66000 46282 2100
0 ...
##  $ Actor_3_name             : chr  "Wes Studi" "Jack Davenport" "Stephanie Sigman" "Joseph
Gordon-Levitt" ...
##  $ Facenumber_in_poster     : int  0 0 1 0 1 0 1 4 3 0 ...
##  $ Plot_keywords            : chr  "avatar|future|marine|native|paraplegic" "goddess|marri
age ceremony|marriage proposal|pirate|singapore" "bomb|espionage|sequel|spy|terrorist" "decep
tion|imprisonment|lawlessness|police officer|terrorist plot" ...
##  $ Movie_imdb_link          : chr  "http://www.imdb.com/title/tt0499549/?ref_=fn_tt_tt_1"
"http://www.imdb.com/title/tt0449088/?ref_=fn_tt_tt_1" "http://www.imdb.com/title/tt2379713/?
ref_=fn_tt_tt_1" "http://www.imdb.com/title/tt1345836/?ref_=fn_tt_tt_1" ...
##  $ Language                 : chr  "English" "English" "English" "English" ...
##  $ Content_rating           : chr  "PG-13" "" "PG-13" "PG-13" ...
##  $ Budget                   : num  2.37e+08 3.00e+08 2.45e+08 2.50e+08 2.64e+08 ...
##  $ Year                     : int  2009 2007 2015 2012 2012 2007 2010 2015 2009 2016 ...
##  $ Actor_2_facebook_likes   : int  936 5000 393 23000 632 11000 553 21000 11000 4000 ...
##  $ Imdb_score               : num  7.9 7.1 6.8 8.5 6.6 6.2 7.8 7.5 7.5 6.9 ...
##  $ Aspect_ratio             : num  1.78 2.35 2.35 2.35 2.35 2.35 1.85 2.35 2.35 2.35 ...
##  $ Movie_facebook_likes     : int  33000 0 85000 164000 24000 0 29000 118000 10000 197000
...
```

```
dim(dat_T2)
```

```
## [1] 3891   26
```

```
#Checking for duplicated records
any(duplicated(dat_T2)) #Returns TRUE
```

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

```
#Removing duplicated records
dat_T2 <- unique(dat_T2)
dim(dat_T2)
```

```
## [1] 3856   26
```

```
#
```

Task 2: Part a

Step 3: Replacing all missing values (““) with NA so that is.na() function works on all the columns irrespective of the data type

```
#Replacing "" with NA to make compatable with is.na() function
dat_T2 <- dat_T2 %>%
  mutate_all(~ifelse(. == "", NA, .))

#Checking NA count for each Column
sapply(dat_T2, function(x) sum(is.na(x)))
```

```
##                     Color                  Director                   Reviews
##                         2                         0                         1
##                  Duration   Director_facebook_likes    Actor_3_facebook_likes
##                         1                         0                        10
##              Actor_2_name    Actor_1_facebook_likes                     Gross
##                         5                         3                         0
##                     Genre              Actor_1_name                     Title
##                         0                         3                         0
##                     Votes Cast_total_facebook_likes             Actor_3_name
##                         0                         0                        10
##      Facenumber_in_poster             Plot_keywords          Movie_imdb_link
##                         6                        31                         0
##                  Language           Content_rating                    Budget
##                         0                        74                         0
##                      Year    Actor_2_facebook_likes               Imdb_score
##                         0                         5                         0
##              Aspect_ratio      Movie_facebook_likes
##                        74                         0
```

Step 4: Removing Columns: Each removal explained with reasoning

Step 4.1: Removing “Movie_imdb_link” and “Plot_keywords” column

–> “Movie_imbd_link” is unique for each movie, so not a good attribute to include.

–> “Plot_keywords” is unique for each Movie, so should be excluded. However, a separate keyword analysis can be done, by separating each keyword for each movie to analyse which keyword attracts most audience. Moreover, it is quite redundant here, as Genre is decided based on the plot.

```
dat_T2 <- subset(dat_T2, select = -c(Movie_imdb_link, Plot_keywords))
dim(dat_T2)
```

```
## [1] 3856   24
```

Step 4.2: Removing “Language” Column

–> The only language we have is “English”, so this column is unnecessary so we can remove it

```
#Checking Language Column
levels(factor(dat_T2$Language))
```

```
## [1] "English"
```

```
#As all the movies are in English, we can remove that column, as it wont be helpful in any an
alysis
dat_T2 <- subset(dat_T2, select = -c(Language))
dim(dat_T2)
```

```
## [1] 3856   23
```

Step 4.3: Dealing with Actor info

–> Actor_1 can be a part of another movie as well, but may be in the column of Actor_2 or Actor_3, so we need to combine them to make them more interpretable. So, extracting them in a separate dataframe to do a separate analysis later on if needed.

–> However, Actor_facebook_likes are included as part of Cast_total_facebook_likes, so it can be removed from our comparative analysis

```
#Extracting actor info in seperate dataframe
dat_actor <- subset(dat_T2, select = c(Title, Gross, Budget,
                                       Actor_1_name, Actor_1_facebook_likes,
                                       Actor_2_name, Actor_2_facebook_likes,
                                       Actor_3_name, Actor_3_facebook_likes))

#Removing it from our current dataframe
dat_T2 <- subset(dat_T2, select = -c(Actor_1_name, Actor_1_facebook_likes,
                                     Actor_2_name, Actor_2_facebook_likes,
                                     Actor_3_name, Actor_3_facebook_likes))
dim(dat_T2)
```

```
## [1] 3856   17
```

```
#Rechecking NAs
sapply(dat_T2, function(x) sum(is.na(x)))
```

```
##                      Color                   Director                    Reviews
##                          2                          0                          1
##                   Duration   Director_facebook_likes                      Gross
##                          1                          0                          0
##                      Genre                      Title                      Votes
##                          0                          0                          0
## Cast_total_facebook_likes      Facenumber_in_poster           Content_rating
##                          0                          6                         74
##                     Budget                       Year                 Imdb_score
##                          0                          0                          0
##               Aspect_ratio      Movie_facebook_likes
##                         74                          0
```

Step 5: Removing the records with NAs in any of its columns

–> All other attributes are important attribute from audience perspective, so we will eliminate all the rows where it has missing values.

```
dat_T2 <- na.omit(dat_T2)
dim(dat_T2)
```

```
## [1] 3719   17
```

```
# We have eliminated a total of 137 additional records with NAs from the dataset
```

Task 2: Part b(i)

Step 6: Calculating Profit and removing Gross and Budget Column

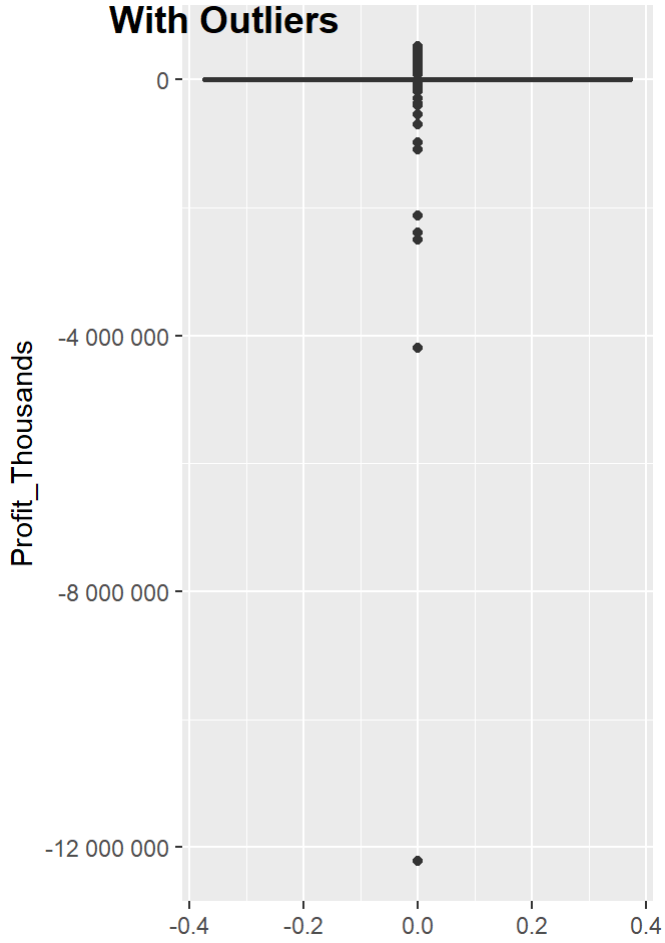
```
#Calculating Profit
dat_T2$Profit <- dat_T2$Gross-dat_T2$Budget

#As now, Gross and Budget Columns are redundant, we can remove them
#Also the profit values are large, so rounding it to thousands to make it more readable
dat_T2 <- dat_T2 %>%
  mutate(Profit_Thousands = round(Profit/1000, 0)) %>%
  select(-c(Gross, Budget, Profit))

head(dat_T2)
```

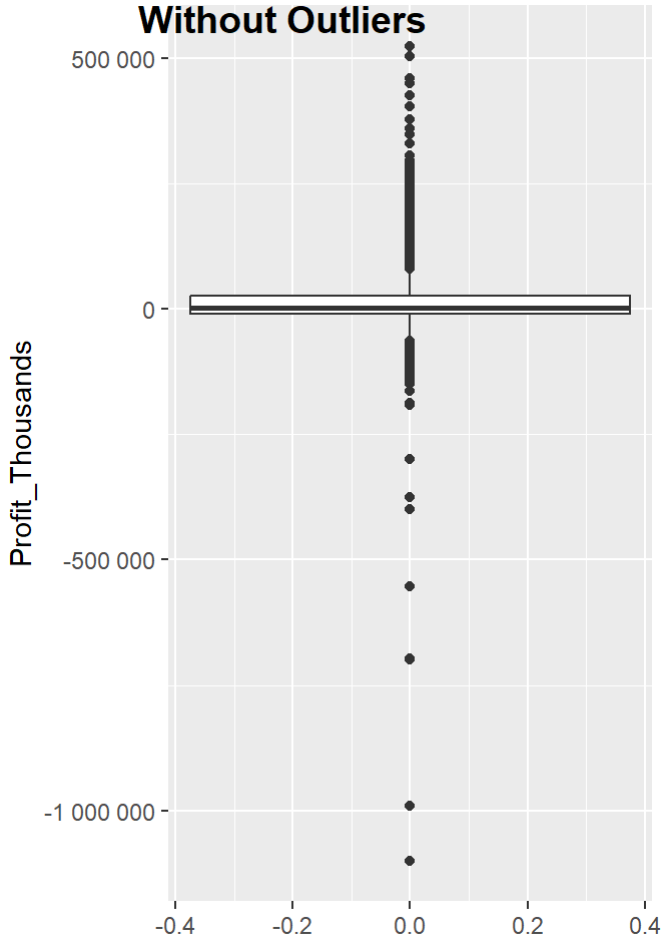
```
##   Color          Director Reviews Duration Director_facebook_likes     Genre
## 1 Color     James Cameron     723      178                       0    Action
## 3 Color        Sam Mendes     602      148                       0    Action
## 4 Color Christopher Nolan     813      164                   22000    Action
## 5 Color    Andrew Stanton     462      132                     475    Action
## 6 Color         Sam Raimi     392      156                       0    Action
## 7 Color      Nathan Greno     324      100                      15 Adventure
##                     Title   Votes Cast_total_facebook_likes Facenumber_in_poster
## 1                Avatara  886204                      2791                    0
## 3               Spectrea  275868                     11554                    1
## 4 The Dark Knight Risesa 1144337                     95000                    0
## 5            John Cartera  212204                      2277                    1
## 6          Spider-Man 3a  383056                     39000                    0
## 7               Tangleda  294810                      1651                    1
##   Content_rating Year Imdb_score Aspect_ratio Movie_facebook_likes
## 1          PG-13 2009        7.9         1.78                33000
## 3          PG-13 2015        6.8         2.35                85000
## 4          PG-13 2012        8.5         2.35               164000
## 5          PG-13 2012        6.6         2.35                24000
## 6          PG-13 2007        6.2         2.35                    0
## 7             PG 2010        7.8         1.85                29000
##   Profit_Thousands
## 1           523506
## 3           -44926
## 4           198131
## 5          -190641
## 6            78530
## 7           -59193
```

Step 7: Checking for Outliers: using boxplot

–> Removed 5 outliers with least profit

```
#Load if not done before
#library(ggplot2)

#Plotting with Outliers
gbox_1 <- ggplot(dat_T2, aes(y=Profit_Thousands))+geom_boxplot() +
  scale_y_continuous(labels = scales::number_format(accuracy = 1))

# Removing last 5 outliers with big losses, to make the charts more readable
dat_T2_w <- dat_T2 %>% arrange(desc(Profit_Thousands))
dat_T2_w <- dat_T2_w[1:(nrow(dat_T2_w)-5),]

#Plotting without Outliers
gbox_2 <- ggplot(dat_T2_w, aes(y=Profit_Thousands))+geom_boxplot() +
  scale_y_continuous(labels = scales::number_format(accuracy = 1))

#Output
library(ggpubr)
ggarrange(gbox_1, gbox_2, labels = c("With Outliers", "Without Outliers"),
          ncol = 2, nrow = 1)
```





Step 8: Relationship analysis between “Profit_Thousands” and other variables

Dividing this Analysis in two parts:

–> Profit vs other Numerical Data-Types: Using Line plot

–> Profit vs other Categorical Data-Types: Using Scatter plot

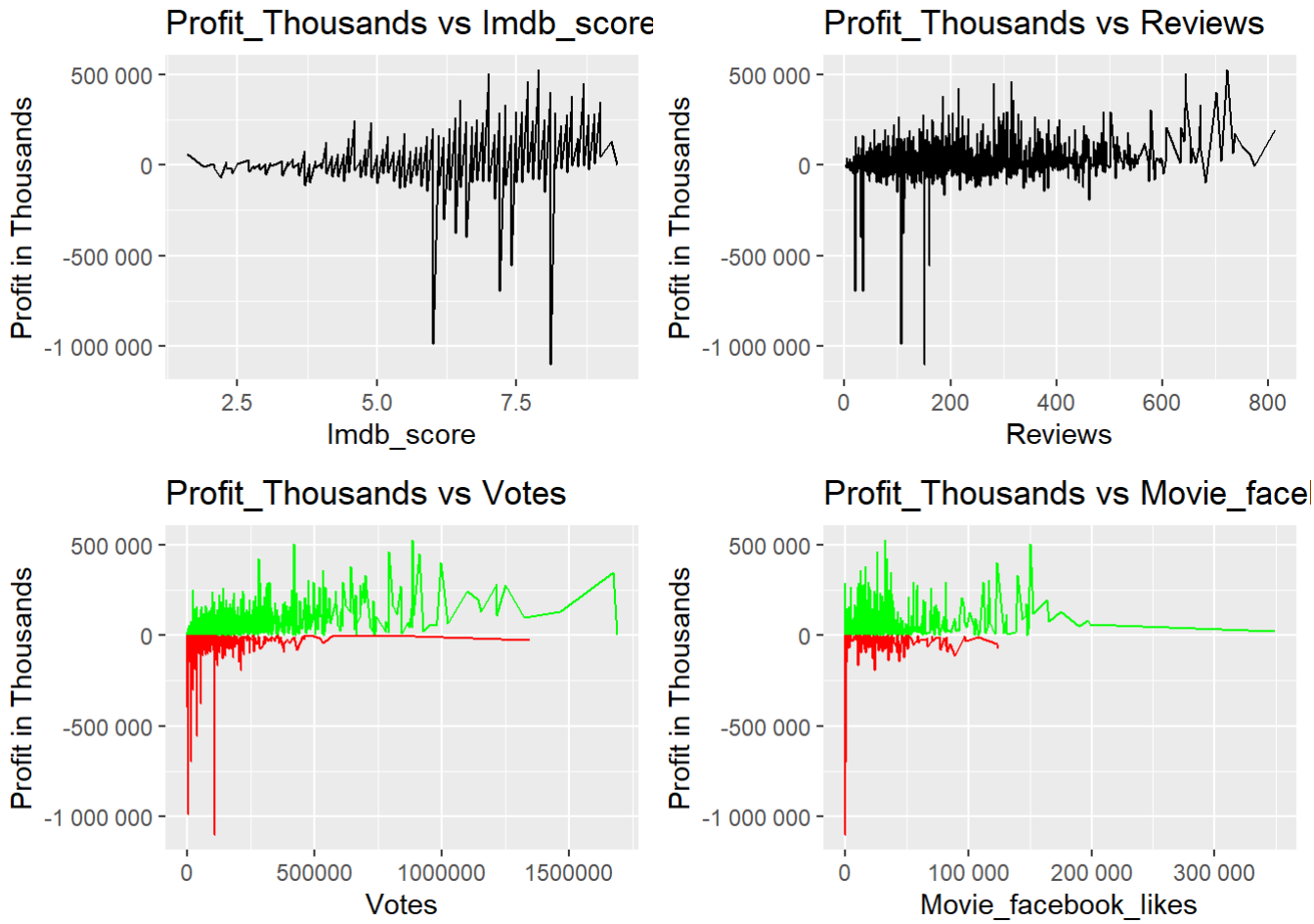
Step 8.1: Profit vs other Numerical Data-Types

We will focus on “Imdb_score”, “Reviews”, “Votes”, “Movie_facebook_likes”

```
#line plot
#We will be using the df without last 5 outliers as it is more visually interpretable

#Load if not done before
#library(ggplot2)

#"Profit_Thousands" vs "Imdb_score"
gline_1 <- ggplot(dat_T2_w, aes(x = Imdb_score, y = Profit_Thousands)) +
  geom_line() +
  scale_y_continuous(labels = scales::number_format(accuracy = 1)) +
  labs(x = "Imdb_score", y = "Profit in Thousands",
       title = "Profit_Thousands vs Imdb_score")

#"Profit_Thousands" vs "Reviews"
gline_2 <- ggplot(dat_T2_w, aes(x = Reviews, y = Profit_Thousands)) +
  geom_line() +
  scale_y_continuous(labels = scales::number_format(accuracy = 1)) +
  labs(x = "Reviews", y = "Profit in Thousands", title = "Profit_Thousands vs Reviews")

#"Profit_Thousands" vs "Votes"
gline_3 <- ggplot(dat_T2_w, aes(x = Votes, y = Profit_Thousands)) +
  geom_line(aes(color = ifelse(Profit_Thousands > 0, "Above 0", "Below 0"))) +
  scale_y_continuous(labels = scales::number_format(accuracy = 1)) +
  labs(x = "Votes", y = "Profit in Thousands", title = "Profit_Thousands vs Votes") +
  scale_color_manual(values = c("Above 0" = "green", "Below 0" = "red")) +
  guides(color = FALSE)
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
#"Profit_Thousands" vs "Movie_facebook_likes"
gline_4 <- ggplot(dat_T2_w, aes(x = Movie_facebook_likes, y = Profit_Thousands)) +
  geom_line(aes(color = ifelse(Profit_Thousands > 0, "Above 0", "Below 0"))) +
  scale_y_continuous(labels = scales::number_format(accuracy = 1)) +
  scale_x_continuous(labels = scales::number_format(accuracy = 1)) +
  labs(x = "Movie_facebook_likes", y = "Profit in Thousands",
       title = "Profit_Thousands vs Movie_facebook_likes") +
  scale_color_manual(values = c("Above 0" = "green", "Below 0" = "red")) +
  guides(color = FALSE)

#Output
#Load if not done before
#library(ggpubr)
ggarrange(gline_1, gline_2, gline_3, gline_4,
          ncol = 2, nrow = 2)
```



Interpretation:

–> It is clear from the line charts that Imdb score is not having much impact on the Profits, as even for the movies with high imdb score, the profits fluctuate a lot.

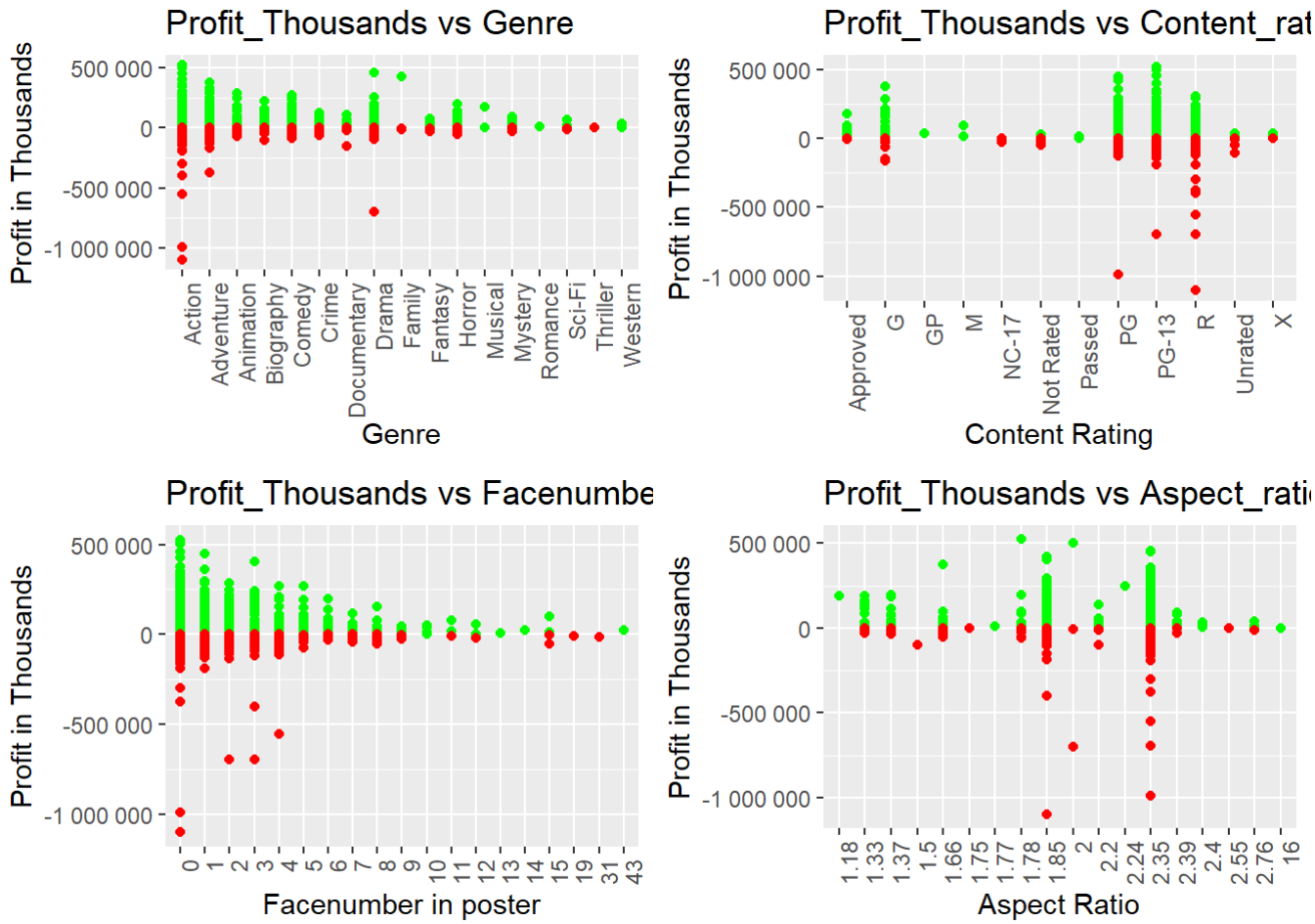
–> For Reviews, there is more fluctuation in profits, where the reviews are below 200. Due to lack of data, it is very hard to say why profits show such a trend for Reviews.

–> Votes and Movie_facebook_likes, show a similar trend, as the count increases for both, the movie is profitable as the profit numbers are more than 0.

Step 8.2: Profit vs other Numerical Data-Types

We will focus on “Genre”, “Content_rating”, “Facenumber_in_poster” and “Aspect_ratio”

```r
#scatter plot
#We will be using the df without last 5 outliers as it is more visually interpretable

#Load if not done before
#library(ggplot2)

#"Profit_Thousands" vs "Genre"
gsp_1 <- ggplot(dat_T2_w, aes(x = factor(Genre), y = Profit_Thousands)) +
  geom_point(aes(color = ifelse(Profit_Thousands > 0, "Above 0", "Below 0"))) +
  scale_y_continuous(labels = scales::number_format(accuracy = 1)) +
  labs(x = "Genre", y = "Profit in Thousands",
       title = "Profit_Thousands vs Genre") +
  scale_color_manual(values = c("Above 0" = "green", "Below 0" = "red")) +
  guides(color = FALSE) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#"Profit_Thousands" vs "Content_rating"
gsp_2 <- ggplot(dat_T2_w, aes(x = factor(Content_rating), y = Profit_Thousands)) +
  geom_point(aes(color = ifelse(Profit_Thousands > 0, "Above 0", "Below 0"))) +
  scale_y_continuous(labels = scales::number_format(accuracy = 1)) +
  labs(x = "Content Rating", y = "Profit in Thousands",
       title = "Profit_Thousands vs Content_rating") +
  scale_color_manual(values = c("Above 0" = "green", "Below 0" = "red")) +
  guides(color = FALSE) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#"Profit_Thousands" vs "Facenumber_in_poster"
gsp_3 <- ggplot(dat_T2_w, aes(x = factor(Facenumber_in_poster), y = Profit_Thousands)) +
  geom_point(aes(color = ifelse(Profit_Thousands > 0, "Above 0", "Below 0"))) +
  scale_y_continuous(labels = scales::number_format(accuracy = 1)) +
  labs(x = "Facenumber in poster", y = "Profit in Thousands",
       title = "Profit_Thousands vs Facenumber_in_poster") +
  scale_color_manual(values = c("Above 0" = "green", "Below 0" = "red")) +
  guides(color = FALSE) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#"Profit_Thousands" vs "Aspect_ratio"
gsp_4 <- ggplot(dat_T2_w, aes(x = factor(Aspect_ratio), y = Profit_Thousands)) +
  geom_point(aes(color = ifelse(Profit_Thousands > 0, "Above 0", "Below 0"))) +
  scale_y_continuous(labels = scales::number_format(accuracy = 1)) +
  labs(x = "Aspect Ratio", y = "Profit in Thousands",
       title = "Profit_Thousands vs Aspect_ratio") +
  scale_color_manual(values = c("Above 0" = "green", "Below 0" = "red")) +
  guides(color = FALSE) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#Output
#Load if not done before
#library(ggpubr)
ggarrange(gsp_1, gsp_2, gsp_3, gsp_4, ncol = 2, nrow = 2)
```



Interpretation:

–> For Genre, Profits are more fluctating for Action, Adventure and Drama in comparison to other Genres

–> For Content Rating, “R” rated movies show the most fluction in Profits. Moreover, “GP”, “M” and “Passed” rated movies has shown profits only so far and “MC-17” rated movies has shown losses only so far.

–> For Facenumber_in_Poster and Aspect_ratio, the visualization is not very meaningful. The only thing, which is visible for facenumber is that, the profits fluctuate less as we have more facenumbers in the poster.

Task 2: Part b(ii)

Step 9: Calculating Correlation between variables: Using Spearman’s rank correlation as it can deal with outliers and skewed distributions

```
#Using the dataset without excluding outliers
#As Spearman's Correlation can only deal with ordinal, interval or ratio, so removing all the
columns not required
dat_T2_c <- dat_T2 %>%
  select(-c(Color, Director, Genre, Title, Content_rating, Aspect_ratio))

#Using Spearman's rank correlation
cor_T2 <- cor(dat_T2_c,  use = "everything", method = c("spearman"))
#Rounding to 4 decimals to make it more readable
round(cor_T2, 4)
```

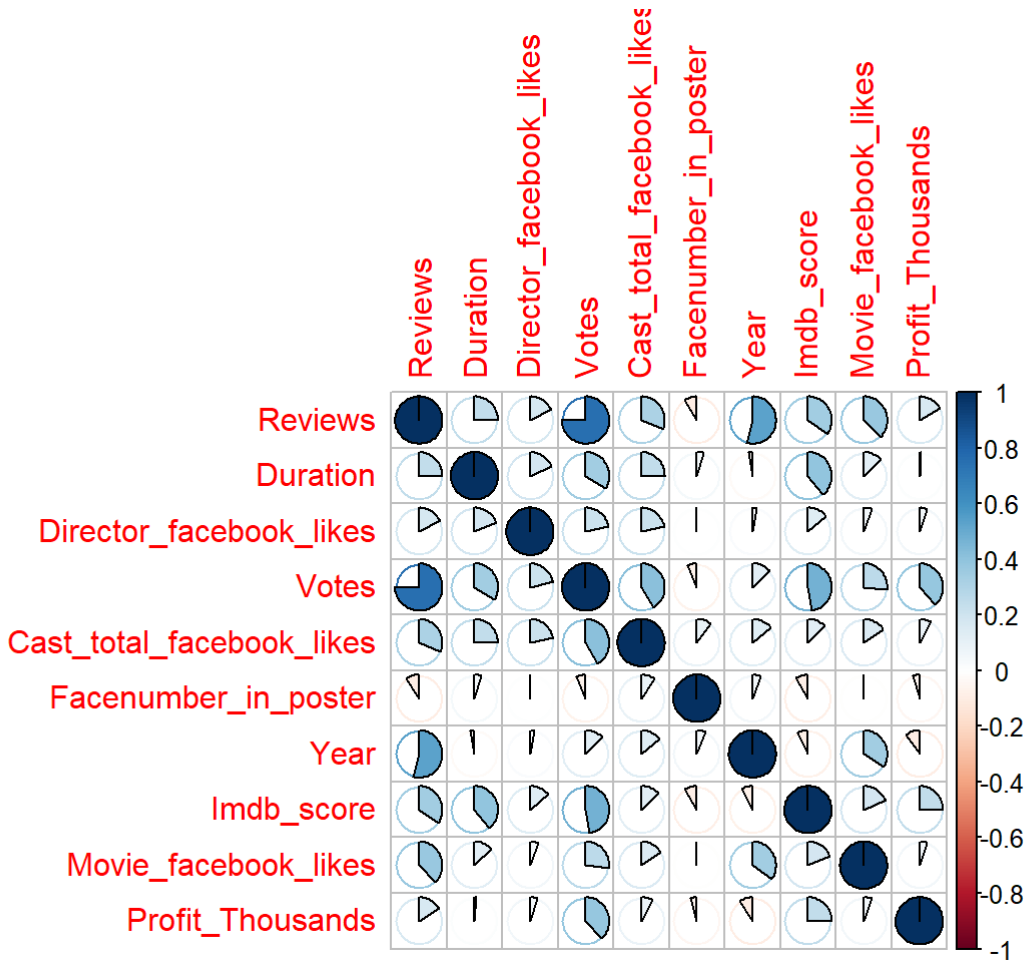
```
##                           Reviews Duration Director_facebook_likes   Votes
## Reviews                    1.0000   0.2456                  0.1738  0.7536
## Duration                   0.2456   1.0000                  0.1843  0.3407
## Director_facebook_likes    0.1738   0.1843                  1.0000  0.2128
## Votes                      0.7536   0.3407                  0.2128  1.0000
## Cast_total_facebook_likes  0.3140   0.2500                  0.2142  0.4172
## Facenumber_in_poster      -0.0918   0.0464                  0.0002 -0.0632
## Year                       0.5397  -0.0277                  0.0287  0.1242
## Imdb_score                 0.3474   0.3912                  0.1364  0.4747
## Movie_facebook_likes       0.3767   0.1290                  0.0562  0.2691
## Profit_Thousands           0.1617   0.0104                  0.0501  0.3849
##                           Cast_total_facebook_likes Facenumber_in_poster
## Reviews                                      0.3140              -0.0918
## Duration                                     0.2500               0.0464
## Director_facebook_likes                      0.2142               0.0002
## Votes                                        0.4172              -0.0632
## Cast_total_facebook_likes                    1.0000               0.0946
## Facenumber_in_poster                         0.0946               1.0000
## Year                                         0.1358               0.0581
## Imdb_score                                   0.1256              -0.0863
## Movie_facebook_likes                         0.1542              -0.0054
## Profit_Thousands                             0.0800              -0.0450
##                              Year Imdb_score Movie_facebook_likes
## Reviews                    0.5397     0.3474               0.3767
## Duration                  -0.0277     0.3912               0.1290
## Director_facebook_likes    0.0287     0.1364               0.0562
## Votes                      0.1242     0.4747               0.2691
## Cast_total_facebook_likes  0.1358     0.1256               0.1542
## Facenumber_in_poster       0.0581    -0.0863              -0.0054
## Year                       1.0000    -0.0750               0.3453
## Imdb_score                -0.0750     1.0000               0.1827
## Movie_facebook_likes       0.3453     0.1827               1.0000
## Profit_Thousands          -0.0943     0.2465               0.0515
##                           Profit_Thousands
## Reviews                             0.1617
## Duration                            0.0104
## Director_facebook_likes             0.0501
## Votes                               0.3849
## Cast_total_facebook_likes           0.0800
## Facenumber_in_poster               -0.0450
## Year                               -0.0943
## Imdb_score                          0.2465
## Movie_facebook_likes                0.0515
## Profit_Thousands                    1.0000
```

Step 10: Plotting Correlation Matrix

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
#Plotting Correlation Matrix using "pie" method
corrplot(cor_T2, method = "pie")
```



Task 2: Part b(iii)

Step 11: Calculating Strong and Weak Correlations

–> Used pie method to visualize correlation matrix, to easily detect the strong and weak correlations.

–> Any value of correlation above 0.5 or below -0.5 are considered as strong and weak otherwise

–> From the above plot, it is clear that we have only two strong correlated variables, i.e. “Reviews and Votes” & “Reviews and Year”

–> For Weak Correlation, we need to look at the values closest to zero, visually the weakest two are “Facenumber_in_poster and Director_Facebook_likes” & “Facenumber_in_poster and Movie_facebook_likes”

```
# Reconfirming strong correlations
cor_st_1 <- cor_T2["Reviews", "Votes"]
cor_st_1
```

```
## [1] 0.753583
```

```
cor_st_2 <- cor_T2["Reviews", "Year"]
cor_st_2
```

```
## [1] 0.5397092
```

```
#both have value above 0.5, so Strong Correlation

# Reconfirming weak correlations
cor_wk_1 <- cor_T2["Facenumber_in_poster", "Director_facebook_likes"]
cor_wk_1
```

```
## [1] 0.0002258789
```

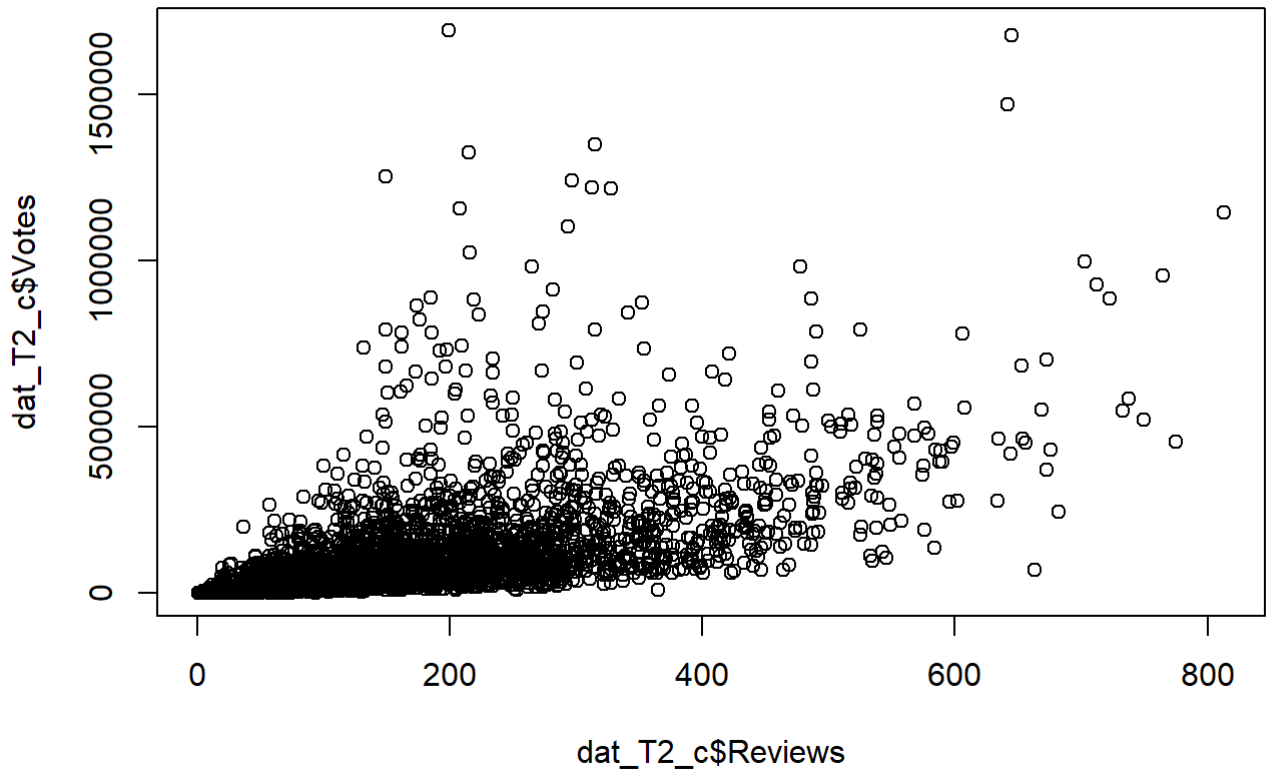
```
cor_wk_2 <- cor_T2["Facenumber_in_poster", "Movie_facebook_likes"]
cor_wk_2
```

```
## [1] -0.005418264
```

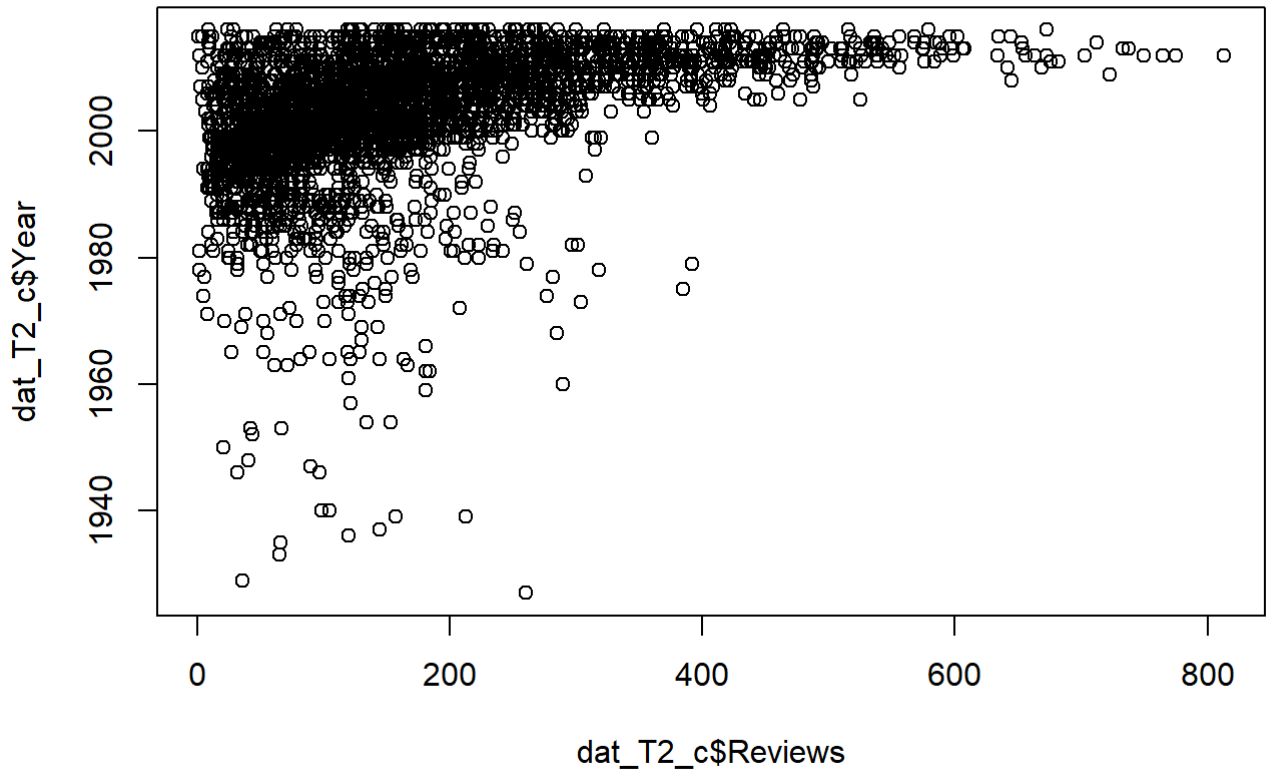
```
#both have value close to 0, so Weak Correlation
```

Step 12: Visualising Strong and Weak Correlations

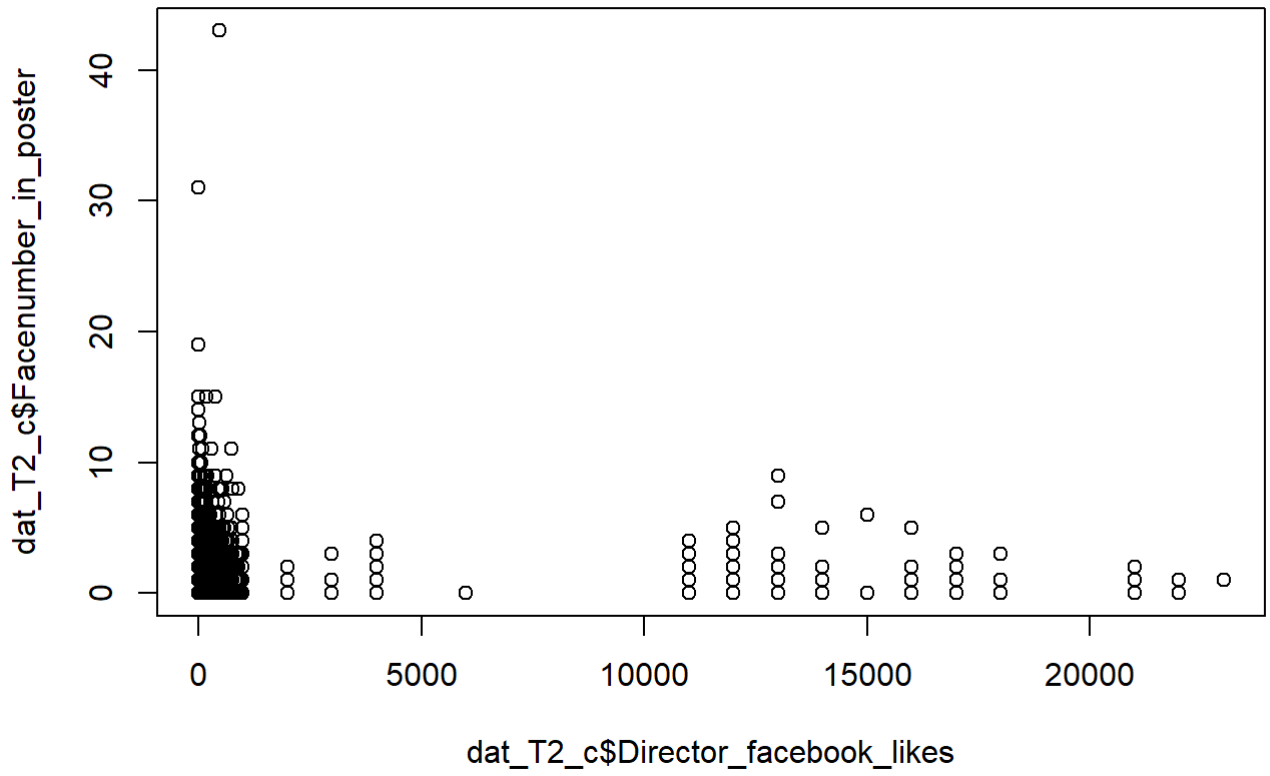
```
#Strong Correlation Reviews and Votes
plot(x = dat_T2_c$Reviews, y = dat_T2_c$Votes)
```



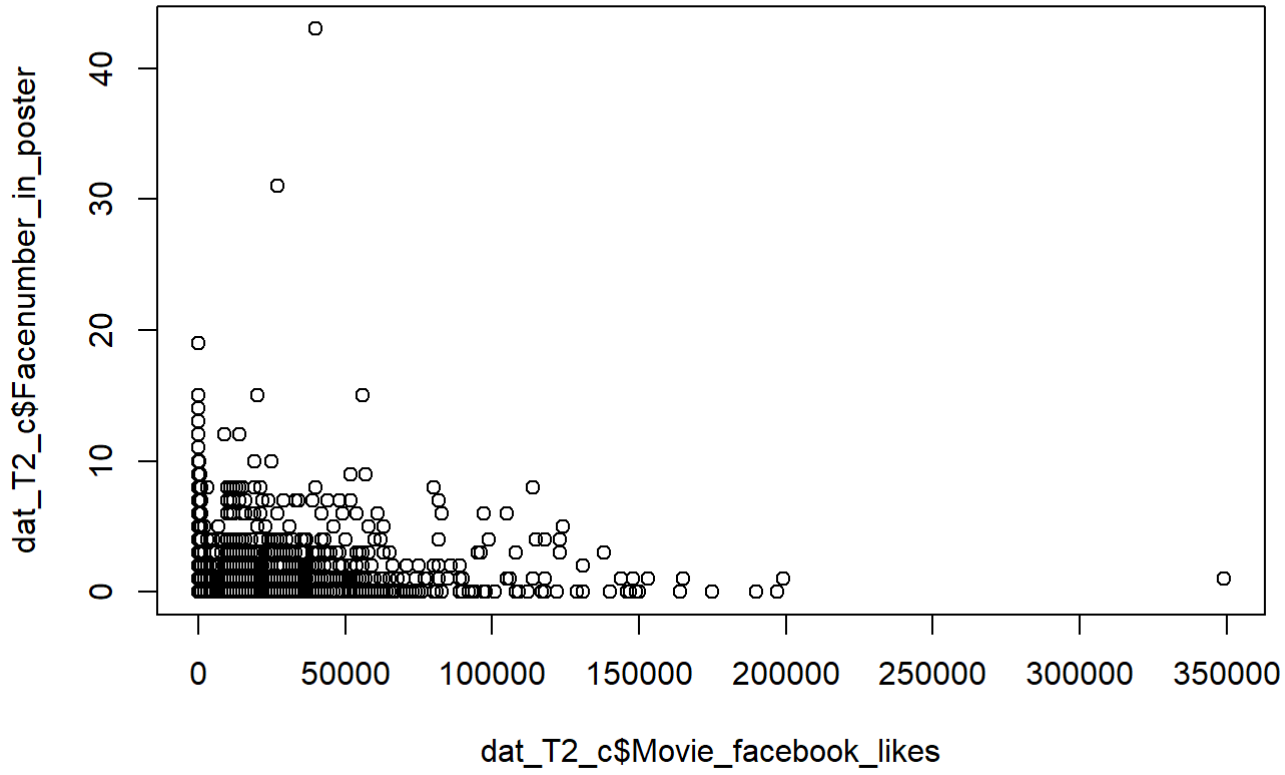

```
#Strong Correlation Reviews and Year
plot(x = dat_T2_c$Reviews, y = dat_T2_c$Year)
```




```
#Weak Correlation Facenumber_in_poster and Director_Facebook_likes
plot(x = dat_T2_c$Director_facebook_likes, y = dat_T2_c$Facenumber_in_poster)
```



```
#Weak Correlation Facenumber_in_poster and Movie_Facebook_likes
plot(x = dat_T2_c$Movie_facebook_likes, y = dat_T2_c$Facenumber_in_poster)
```



The above plots for strong correlation show some directional plot, with outliers, whereas, no such pattern is visible for weak correlations