Why Not Watch streaming service data analysis

Astrid de Geest

2023-06-11

Install packages

```
library(dplyr) # for working with the data
library(ggplot2) # for creating graphs
library(magrittr) # for using pipes
library(tidyr) # for changing data to be tidy
library(readr) # for importing data
library(knitr) # for pdf and data analysis
library(lubridate) # to work with times and dates
library(corrplot) # to do a correlation plot
library(plot3D) # to create a 3d plot
library(stringr) # for output statements
library(car) #for variance test
```

Input the data

```
data <- read_csv("../Data/streaming_data.csv")
head(data)
```

```
## # A tibble: 6 x 8
##   date  gender   age social_metric time_since_signup demographic group
##   <chr> <chr>  <dbl>         <dbl>             <dbl>       <dbl> <chr>
## 1 1-Jul F         28             5              19.3           1 A
## 2 1-Jul F         32             7              11.5           1 A
## 3 1-Jul F         39             4               4.3           3 A
## 4 1-Jul M         52            10               9.5           4 A
## 5 1-Jul M         25             1              19.5           2 A
## 6 1-Jul M         51             0              22.6           4 A
## # i 1 more variable: hours_watched <dbl>
```

Initial analysis of the data

There is no missing data or extreme outliers however the some of the columns are not the right catagory

```
# check columns types data and dimensions
str(data)
```

```
## spc_tbl_ [1,000 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ date             : chr [1:1000] "1-Jul" "1-Jul" "1-Jul" "1-Jul" ...
##  $ gender           : chr [1:1000] "F" "F" "F" "M" ...
##  $ age              : num [1:1000] 28 32 39 52 25 51 53 42 41 20 ...
##  $ social_metric    : num [1:1000] 5 7 4 10 1 0 5 6 8 7 ...
##  $ time_since_signup: num [1:1000] 19.3 11.5 4.3 9.5 19.5 22.6 4.2 8.5 16.9 23 ...
```

```
##  $ demographic      : num [1:1000] 1 1 3 4 2 4 3 4 4 2 ...
##  $ group            : chr [1:1000] "A" "A" "A" "A" ...
##  $ hours_watched    : num [1:1000] 4.08 2.99 5.74 4.13 4.68 3.4 3.07 2.77 2.24 5.39 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   date = col_character(),
##   ..   gender = col_character(),
##   ..   age = col_double(),
##   ..   social_metric = col_double(),
##   ..   time_since_signup = col_double(),
##   ..   demographic = col_double(),
##   ..   group = col_character(),
##   ..   hours_watched = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
# check data for numeric columns
summary(data)
```

```
##      date              gender               age        social_metric
##  Length:1000        Length:1000        Min.   :18.00   Min.   : 0.000
##  Class :character   Class :character   1st Qu.:28.00   1st Qu.: 2.000
##  Mode  :character   Mode  :character   Median :36.00   Median : 5.000
##                                        Mean   :36.49   Mean   : 4.911
##                                        3rd Qu.:46.00   3rd Qu.: 8.000
##                                        Max.   :55.00   Max.   :10.000
##  time_since_signup  demographic       group           hours_watched
##  Min.   : 0.00     Min.   :1.000   Length:1000        Min.   :0.500
##  1st Qu.: 5.70     1st Qu.:2.000   Class :character   1st Qu.:3.530
##  Median :11.80     Median :3.000   Mode  :character   Median :4.415
##  Mean   :11.97     Mean   :2.603                      Mean   :4.393
##  3rd Qu.:18.70     3rd Qu.:4.000                      3rd Qu.:5.322
##  Max.   :24.00     Max.   :4.000                      Max.   :8.300
```

```
# check for missing data
sum(is.na(data))
```

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

```
# check unique variables of catagory data
table(data$gender)
```

```
##
##   F   M
## 429 571
```

```
table(data$group)
```

```
##
##   A   B
## 880 120
```

Clean the data

The columns were changed into the right category. The dataset C was created so correlations could easily be done on the whole dataset with an error occurring.

As time and data data can be tricky to work with datat as a data set was created with those parameters and

data retains the time duration columns as numerical.

```
#Create dataset for correlations prior to changing categories
dataC <- data

# change age to whole numbers only
data$age <- as.integer(data$age)

# change categories of data to factors
data$gender <- as.factor(data$gender)
data$social_metric <- as.factor(data$social_metric)
data$demographic <- as.factor(data$demographic)
data$group <- as.factor(data$group)

# change date
default_year = "2022" # R requires a year so I made an assumption of 2022
data <- data %>%
  mutate(date = as.Date(paste0(date, "-", default_year), format = "%d-%b-%Y"))

# create new dataset to give the option of time in numeric or time category
datat <- data

# change time since last signup
f <- function(x) {
    month <- floor(x)
    day <- round((x - month) * 30.42)
    return(sprintf("%i months, %i days", month, day))
}

datat$time_since_signup <- period(sapply(datat$time_since_signup, f))

# change hours watched
g <- function(x) {
    hours <- floor(x)
    minutes <- round((x - hours) * 60)
    return(sprintf("%i hours, %i minutes", hours, minutes))
}

datat$hours_watched <- period(sapply(datat$hours_watched, g))

#check both datasets to ensure changes are correct
head(data)
```

```
## # A tibble: 6 x 8
##   date       gender   age social_metric time_since_signup demographic group
##   <date>     <fct>  <int> <fct>                     <dbl> <fct>       <fct>
## 1 2022-07-01 F         28 5                          19.3 1           A
## 2 2022-07-01 F         32 7                          11.5 1           A
## 3 2022-07-01 F         39 4                           4.3 3           A
## 4 2022-07-01 M         52 10                          9.5 4           A
## 5 2022-07-01 M         25 1                          19.5 2           A
## 6 2022-07-01 M         51 0                          22.6 4           A
## # i 1 more variable: hours_watched <dbl>
```

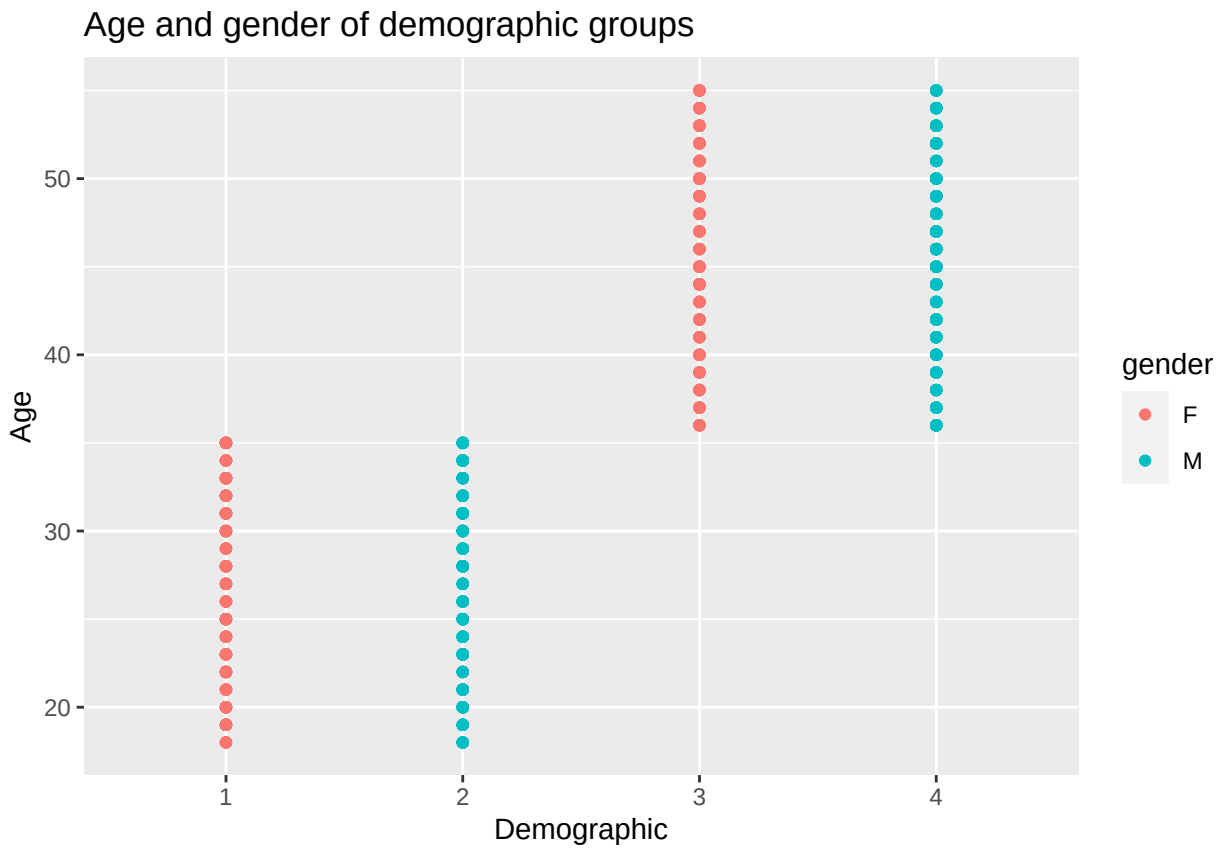
```
head(datat)
```

```
## # A tibble: 6 x 8
##   date       gender   age social_metric time_since_signup demographic group
##   <date>     <fct>  <int> <fct>         <Period>           <fct>       <fct>
## 1 2022-07-01 F         28 5             19m 9d 0H 0M 0S    1           A
## 2 2022-07-01 F         32 7             11m 15d 0H 0M 0S   1           A
## 3 2022-07-01 F         39 4             4m 9d 0H 0M 0S     3           A
## 4 2022-07-01 M         52 10            9m 15d 0H 0M 0S    4           A
## 5 2022-07-01 M         25 1             19m 15d 0H 0M 0S   2           A
## 6 2022-07-01 M         51 0             22m 18d 0H 0M 0S   4           A
## # i 1 more variable: hours_watched <Period>
```

Analysis of social metric and demographic

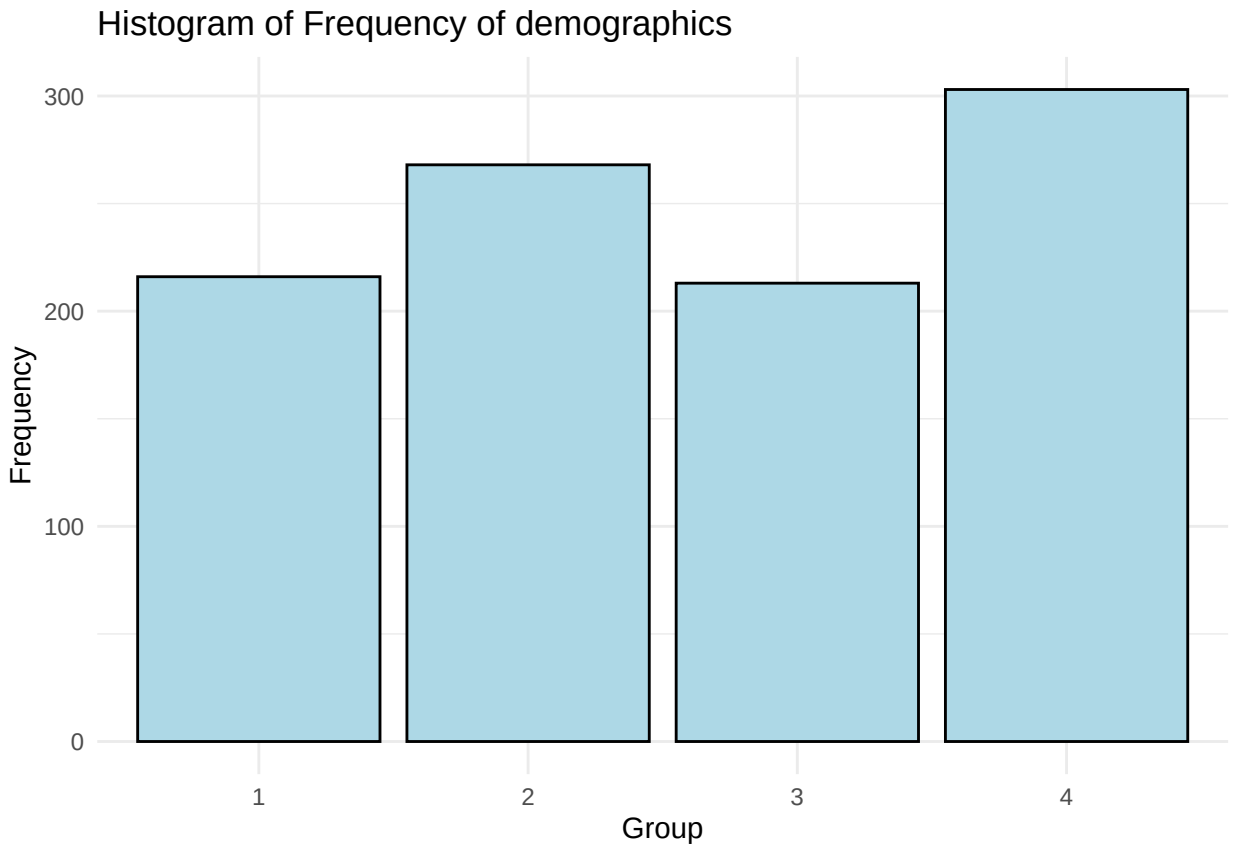
The data for social metric and demographic were examined against other values. The demographic group splits the data into young men and women and old men and women. The social metric has fairly even demographics but the group 0 and 10 are much smaller than the others.

```
ggplot(data, aes(x = demographic, y = age)) +
  geom_point(aes(colour = gender)) +
  labs(title = "Age and gender of demographic groups",
           x = "Demographic",
           y = "Age")
```

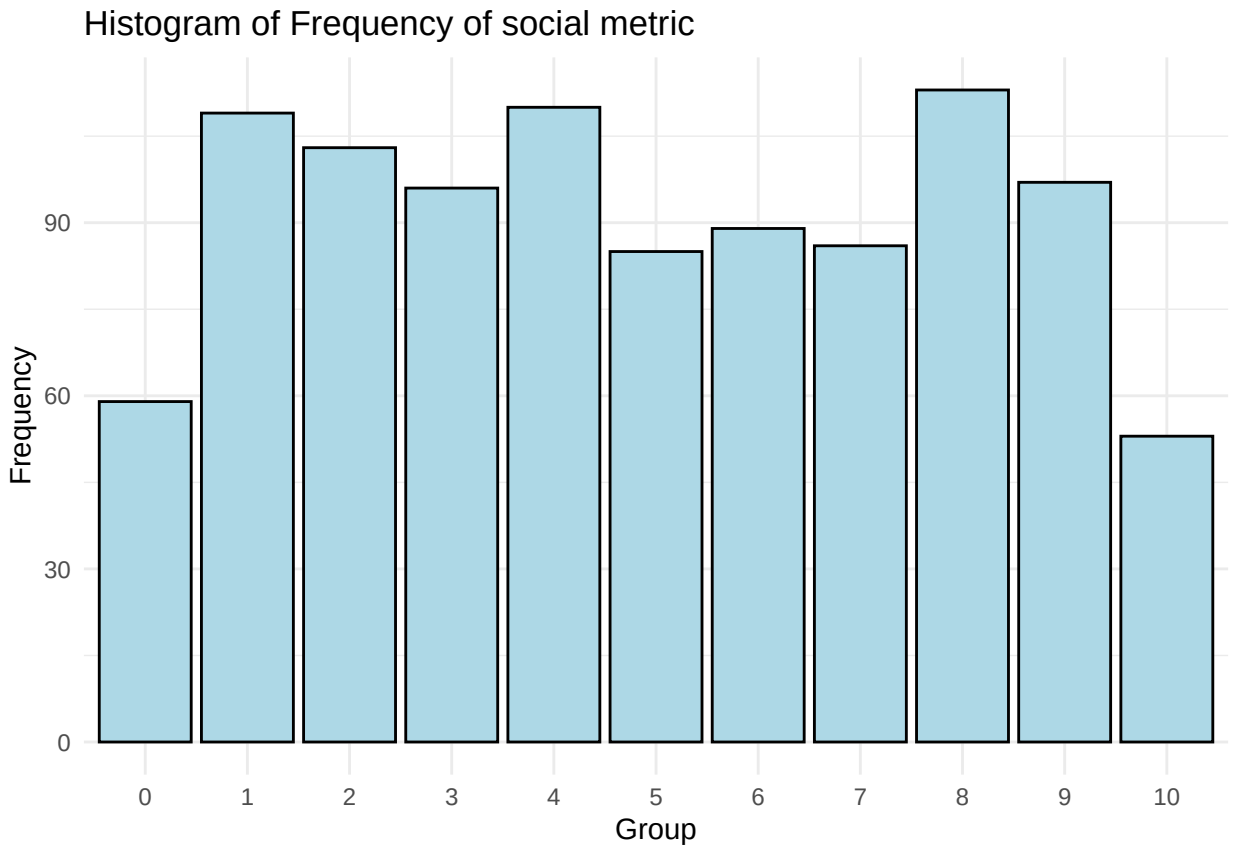



```
ggplot(data, aes(x = demographic)) +
  geom_histogram(stat = "count", fill = "light blue", color = "black") +
  theme_minimal() +
```

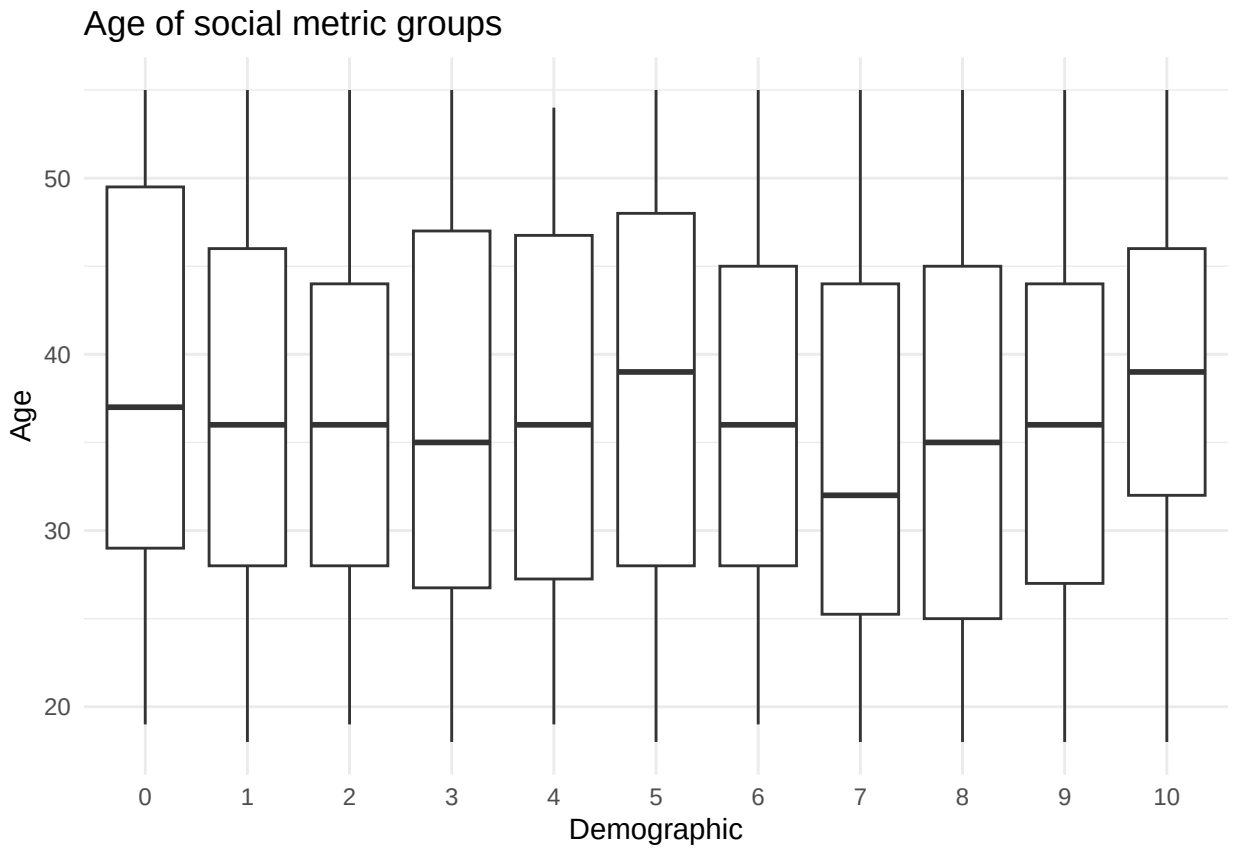
```
  labs(x = "Group", y = "Frequency", title = "Histogram of Frequency of demographics")
```



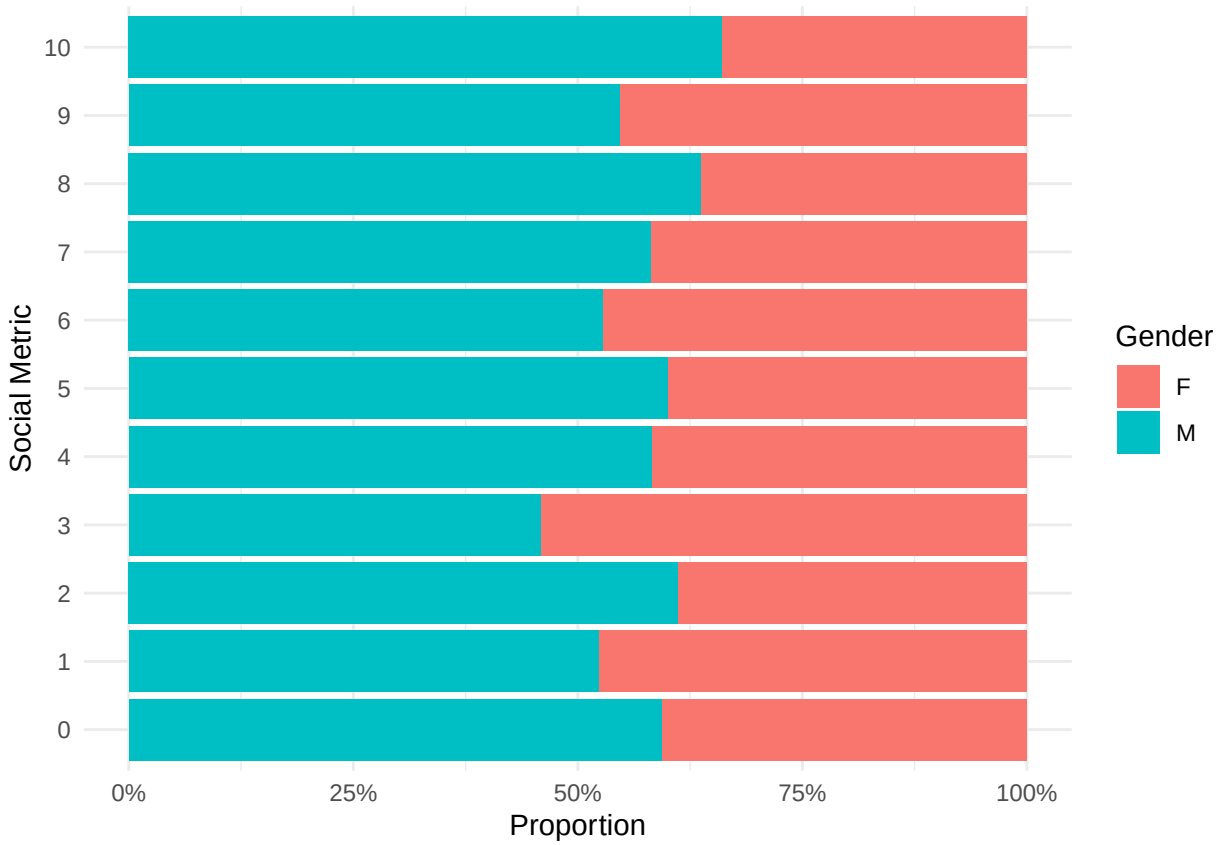

```
ggplot(data, aes(x = social_metric)) +
  geom_histogram(stat = "count", fill = "light blue", color = "black") +
  theme_minimal() +
  labs(x = "Group", y = "Frequency", title = "Histogram of Frequency of social metric")
```




```
ggplot(data, aes(x = social_metric, y = age)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Age of social metric groups",
           x = "Demographic",
           y = "Age")
```




```
ggplot(data, aes(x = social_metric, fill = gender)) +
  geom_bar(position = "fill") +
  coord_flip() +
  labs(x = "Social Metric", y = "Proportion", fill = "Gender") +
  theme_minimal() +
  scale_y_continuous(labels = scales::percent)
```



Analysis of AB groups

```
# Create dataset with A/B in the same timeline
dataD <- data %>%
  filter(date >= as.Date("2022-07-18"))

cat(str_interp("The number of participants in Group A and B when timeline for both are the same\n"))
```

```
## The number of participants in Group A and B when timeline for both are the same
```

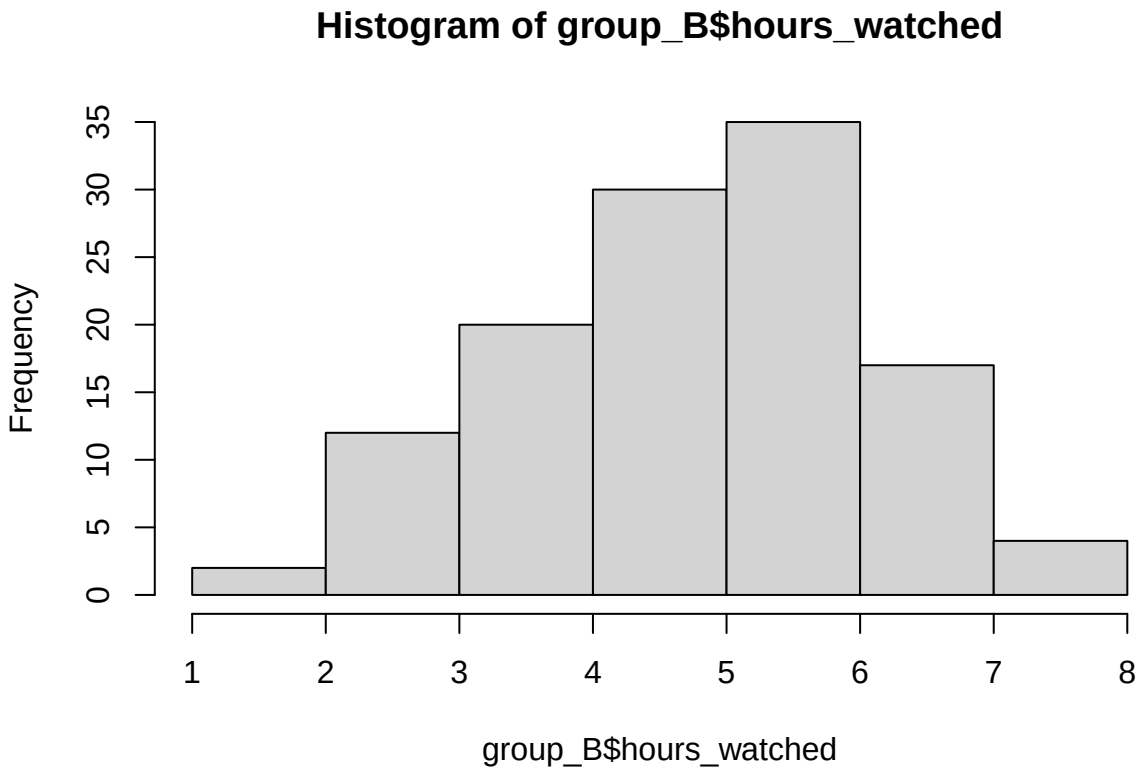
```
#find out new A/B ratio
table(dataD$group)
```

```
##
##   A   B
## 332 120
```

```
#Hours watched by each group
hours_watchedAB <- dataD %>%
  group_by(group) %>%
  summarise(median_time = median(hours_watched))

hours_watchedAB <- data.frame(median_time = period(sapply(hours_watchedAB$median_time, g)))

# check for significance
group_A <- dataD %>%
  filter(group == "A") %>%
  select(hours_watched)
```

```r
group_B <- dataD %>%
  filter(group == "B") %>%
  select(hours_watched)

#check for normal distribution
hist(group_A$hours_watched)
```


```r
hist(group_B$hours_watched)
```




```
# check variance
variance = var.test(group_A$hours_watched, group_B$hours_watched)

#perform t test as not completely normally distributed
t.test(group_A$hours_watched, group_B$hours_watched)
```

```
##
##  Welch Two Sample t-test
##
## data:  group_A$hours_watched and group_B$hours_watched
## t = -2.8602, df = 217.62, p-value = 0.004646
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6907498 -0.1271448
## sample estimates:
## mean of x mean of y
##  4.401928  4.810875
```

```
# compare medians of age between AB
A_Age <-  dataD %>%
  filter(group == "A") %>%
  summarise(median = median(age)) %>%
  pull(median)

B_Age <-  dataD %>%
  filter(group == "B") %>%
  summarise(median = median(age)) %>%
  pull(median)

cat(str_interp("Age of group A and Group B respectively: ${A_Age}, ${B_Age}\n"))
```

```
## Age of group A and Group B respectively: 35, 39.5
```

```
# compare median of time since signup between AB
A_signup <-  dataD %>%
  filter(group == "A") %>%
  summarise(median = median(time_since_signup)) %>%
  pull(median)

B_signup <-  dataD %>%
  filter(group == "B") %>%
  summarise(median = median(time_since_signup)) %>%
  pull(median)

cat(str_interp("The time since last signup of group A and Group B respectively: ${A_signup}, ${B_signup}
```

```
## The time since last signup of group A and Group B respectively: 11.05, 11.35
```

```
# differences in sex demographics between AB
AB_sex <- dataD %>%
  group_by(group, gender) %>%
  summarise(count = n(), .groups = "drop") %>%
  group_by(group) %>%
  mutate(percentage = round(count / sum(count) * 100, 2)) %>%
  select(-count) %>%
  pivot_wider(names_from = group, values_from = percentage)
cat(str_interp("The percentages of each gender in groups A/B \n"))
```

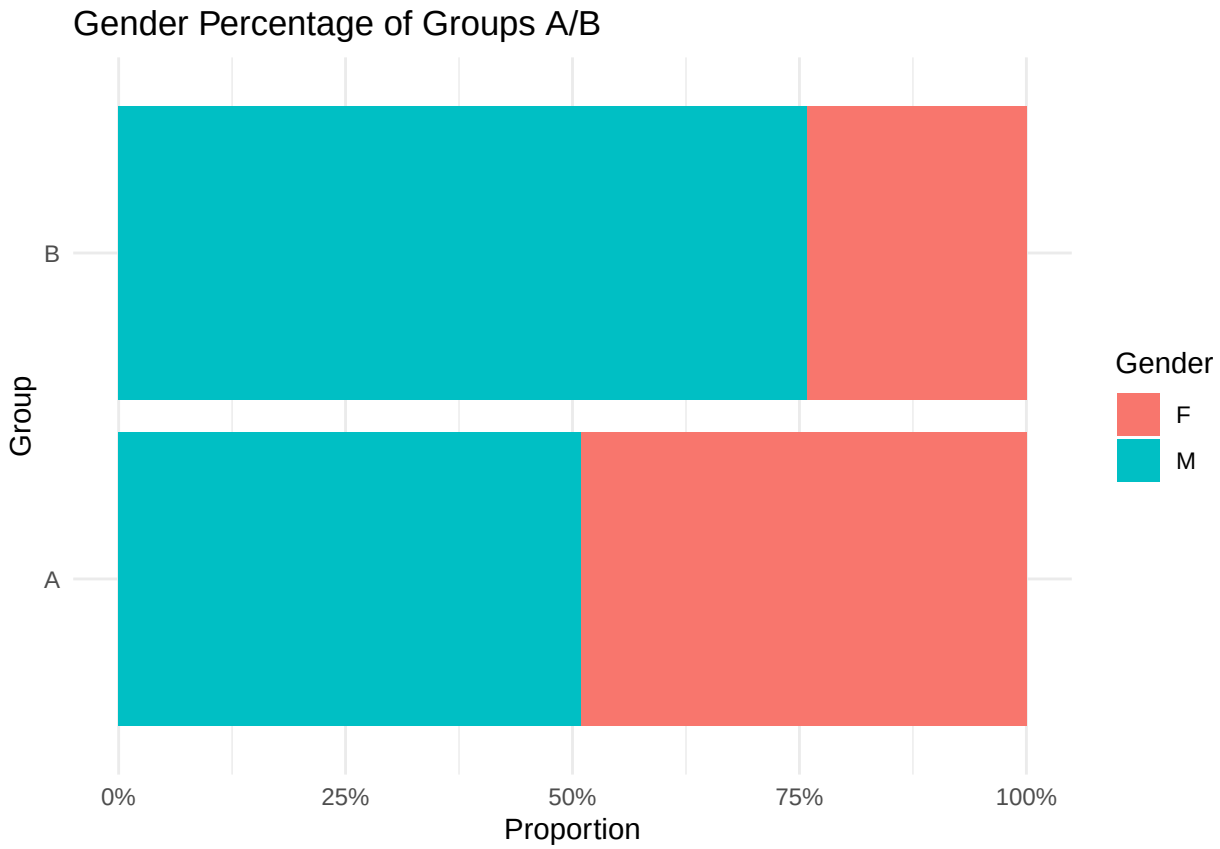
```
## The percentages of each gender in groups A/B
```

```
AB_sex
```

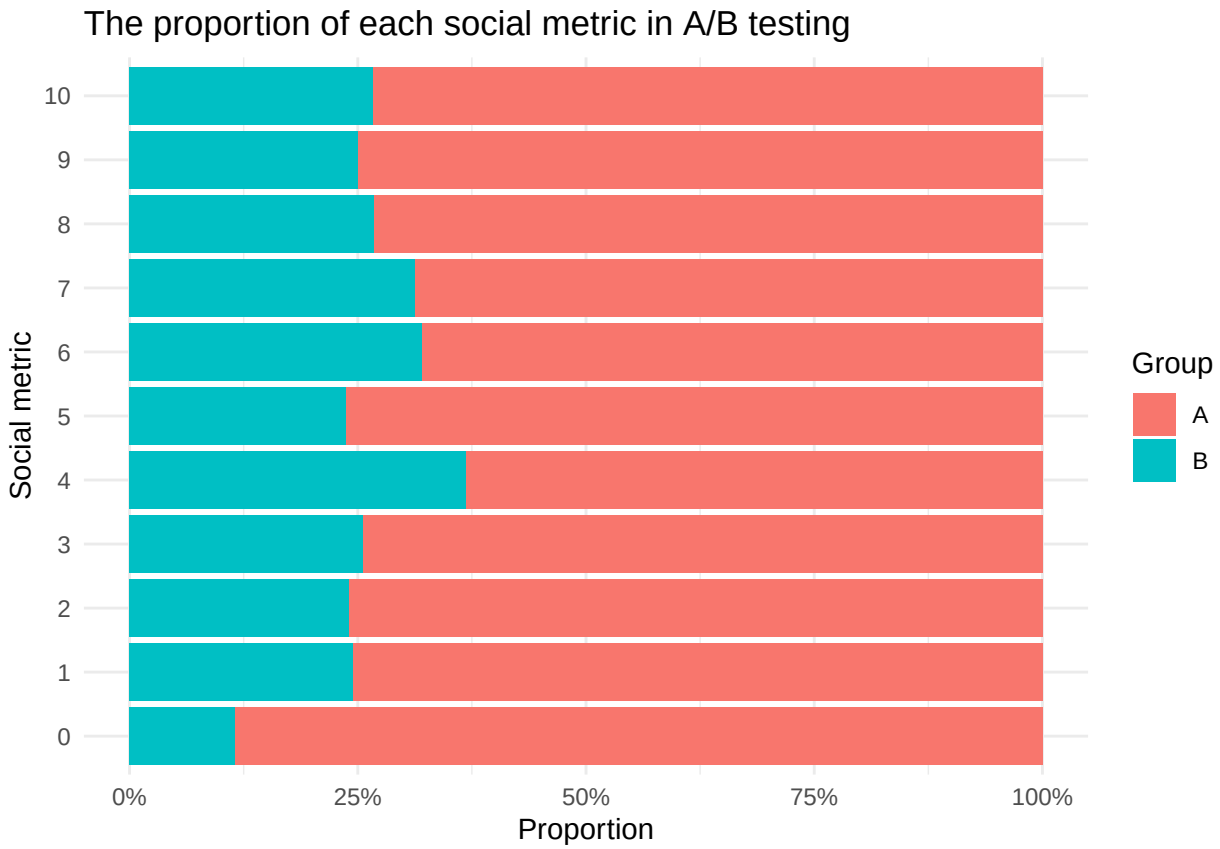
```
## # A tibble: 2 x 3
##   gender     A     B
##   <fct>  <dbl> <dbl>
## 1 F       49.1  24.2
## 2 M       50.9  75.8
```

plots on A/B group characteristics

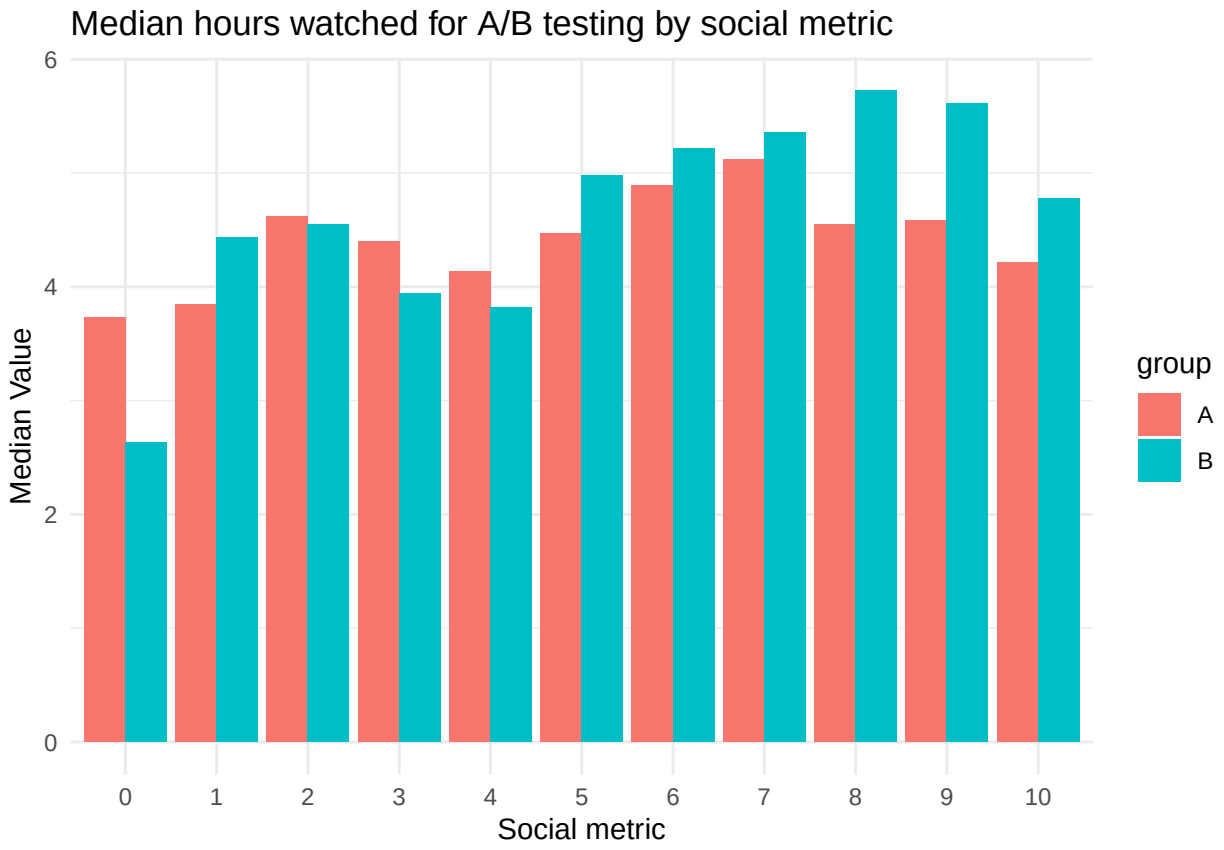
```
#Plot of sex differences
ggplot(dataD, aes(x = group, fill = gender)) +
  geom_bar(position = "fill") +
  coord_flip() +
  labs(x = "Group", y = "Proportion", fill = "Gender",
       title = "Gender Percentage of Groups A/B") +
  theme_minimal() +
  scale_y_continuous(labels = scales::percent)
```




```
#Plot the distribution of each social metric in group A/B testing
ggplot(dataD, aes(x = social_metric, fill = group)) +
  geom_bar(position = "fill") +
  coord_flip() +
  labs(x = "Social metric", y = "Proportion", fill = "Group",
       title = "The proportion of each social metric in A/B testing") +
  theme_minimal() +
  scale_y_continuous(labels = scales::percent)
```




```r
#Plot Median hours by social metric
social_hours <- dataD %>%
  group_by(group, social_metric) %>%
  summarise(median_value1 = median(hours_watched), .groups = "drop")

ggplot(social_hours, aes(x = social_metric, y = median_value1, fill = group)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Median hours watched for A/B testing by social metric",
          x = "Social metric",
          y = "Median Value") +
  theme_minimal()
```



Regression analysis 1: Time of intervention B and potiential effect

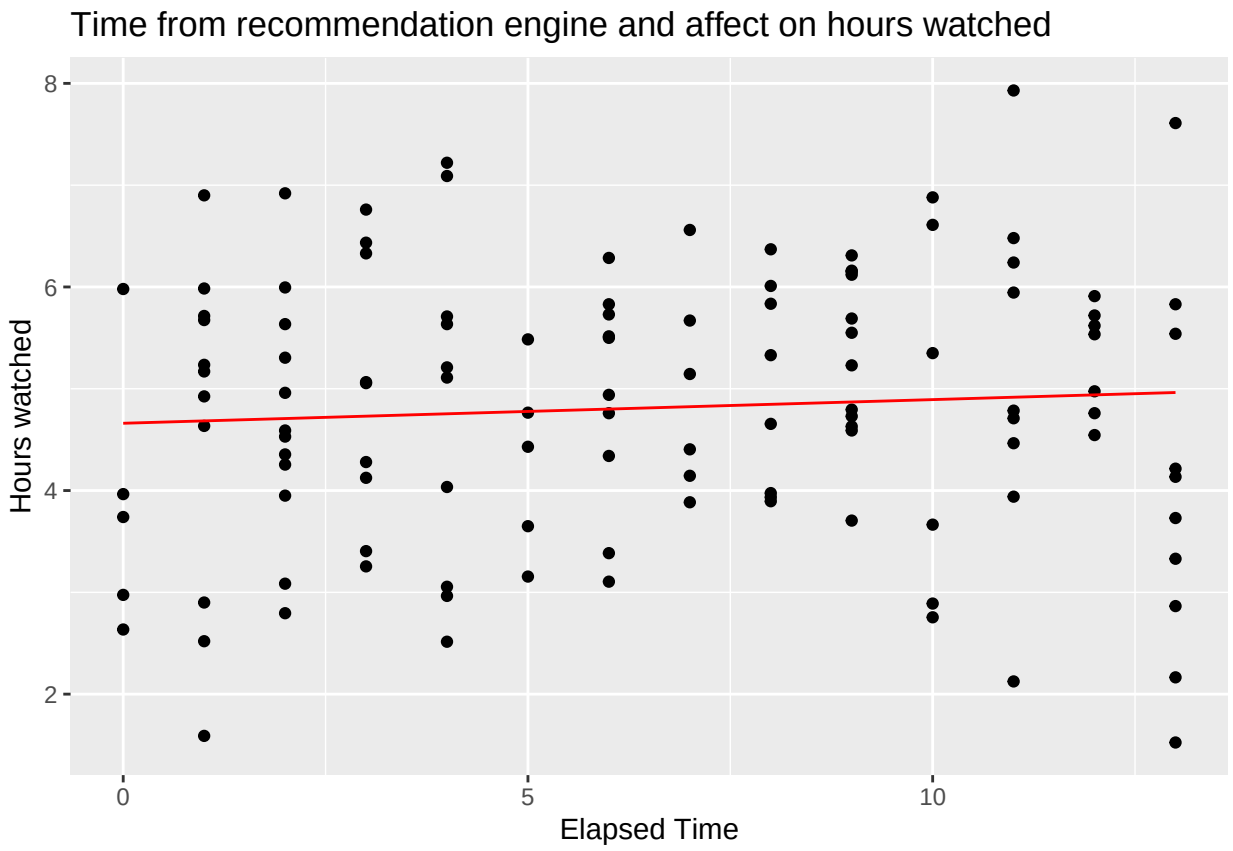
A regression analysis was done on the elapsed time of the B group and impact on watching time. It was not signifcantly signicant and there were no real differences in the data provided for the time period.

```
# create new data set of time elapsed since intervention
dataB <- data %>%
  filter(group == "B") %>%
  mutate(elapsed_time = date - as.Date("2022-07-18"))

dataB$elapsed_time <- as.integer(dataB$elapsed_time)

# create simple linear regression of hours watched and time since intervention
hours_slr = lm(hours_watched ~ elapsed_time, data = dataB)
summary(hours_slr)
```

```
##
## Call:
## lm(formula = hours_watched ~ elapsed_time, data = dataB)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4381 -0.9142  0.0115  0.9596  3.0134
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.66099    0.22901  20.353   <2e-16 ***
## elapsed_time  0.02324    0.03009   0.772    0.442
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.331 on 118 degrees of freedom
## Multiple R-squared:  0.005028,   Adjusted R-squared:  -0.003404
## F-statistic: 0.5963 on 1 and 118 DF,  p-value: 0.4415
```

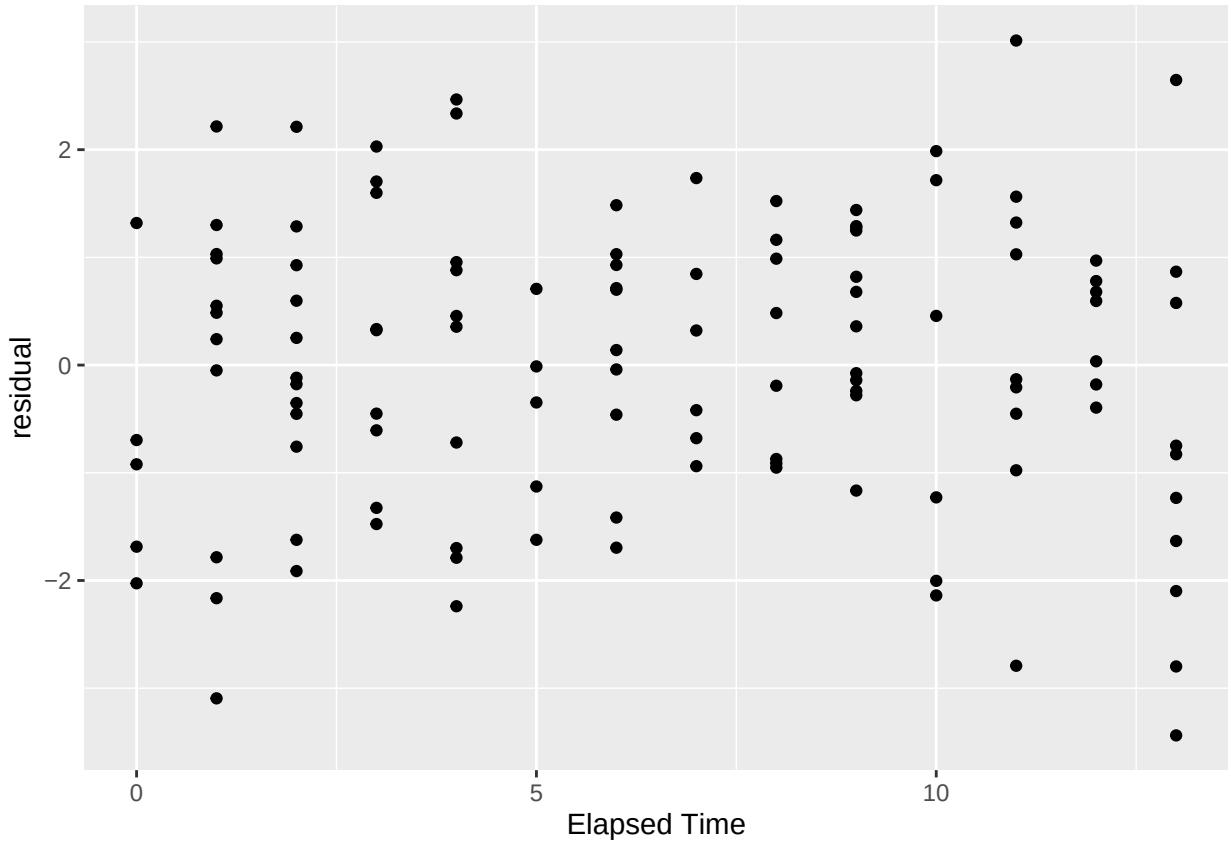
```r
a0 <- coef(hours_slr)[1]
a1 <- coef(hours_slr)[2]

x_slr <- seq(min(dataB$elapsed_time), max(dataB$elapsed_time), 1)
y_slr <- a0 + a1 * x_slr

# plot the regression
ggplot()+
  geom_point(aes(x = dataB$elapsed_time, y = dataB$hours_watched))+
  geom_line(aes(x = x_slr, y = y_slr), colour = "red")+
  labs(x = "Elapsed Time", y ="Hours watched")+
  ggtitle("Time from recommendation engine and affect on hours watched")
```



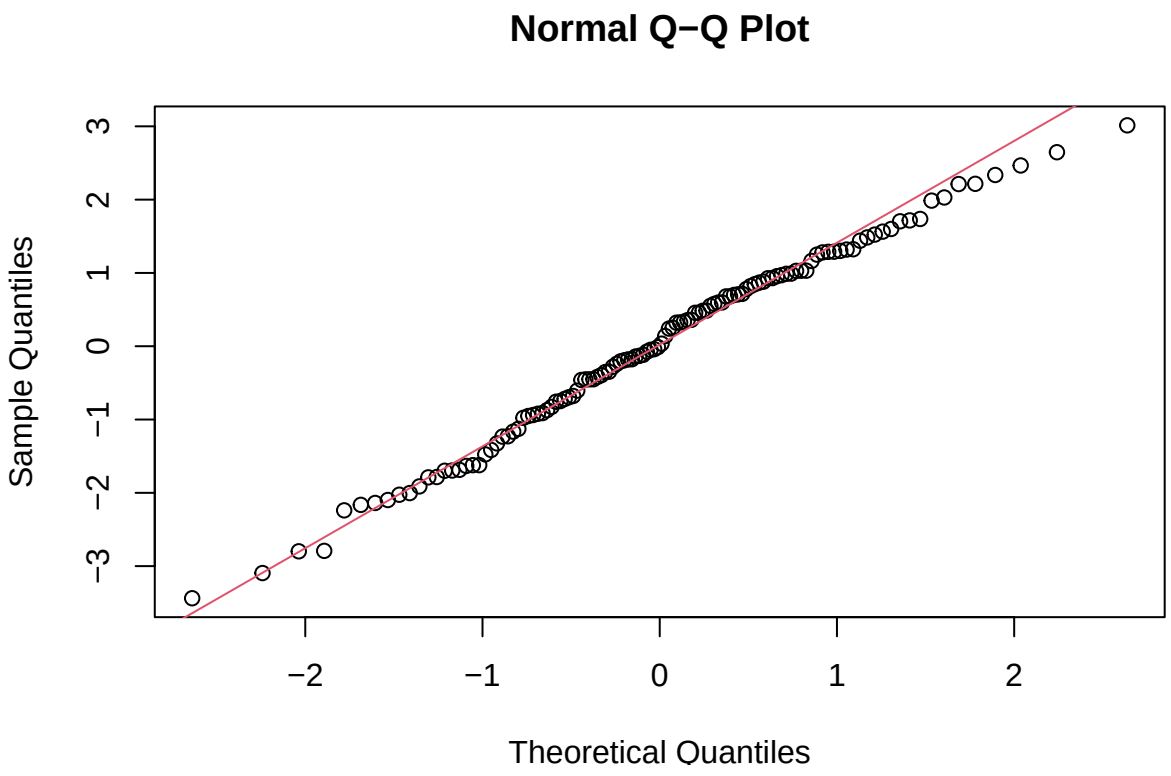
Calculate and plot residuals

```r
dataB$y_hat <- a0 + a1 * dataB$elapsed_time
dataB$error <- dataB$hours_watched - dataB$y_hat

ggplot()+
  geom_point(aes(x = dataB$elapsed_time, y = dataB$error))+
```

```
  labs(x = "Elapsed Time", y = "residual")
```




```
qqnorm(dataB$error)
qqline(dataB$error, col = 2)
```



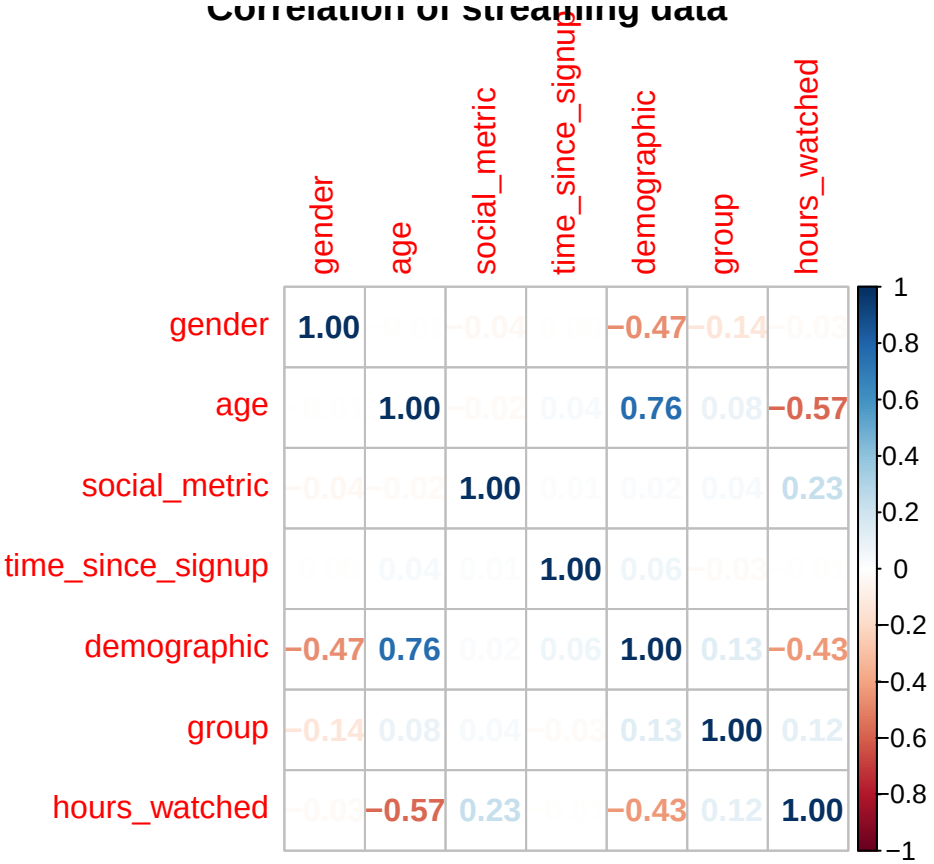
Correlation plot of factors

Notable correlations are age and social metric. I ignored demographic as it is a correlation influenced by the effect of age.

```
#Change dataC to be suitable for multiple correlations
dataC <- dataC %>% select(-date)
dataC <- dataC %>%
  mutate(gender = ifelse(gender == "M", 0, 1),
         gender = as.numeric(gender))


dataC <- dataC %>%
  mutate(group = ifelse(group == "A", 0, 1),
         group = as.numeric(group))

head(dataC)
```

```
## # A tibble: 6 x 7
##   gender   age social_metric time_since_signup demographic group hours_watched
##    <dbl> <dbl>         <dbl>             <dbl>       <dbl> <dbl>         <dbl>
## 1      1    28             5              19.3           1     0          4.08
## 2      1    32             7              11.5           1     0          2.99
## 3      1    39             4               4.3           3     0          5.74
## 4      0    52            10               9.5           4     0          4.13
## 5      0    25             1              19.5           2     0          4.68
## 6      0    51             0              22.6           4     0          3.4
```

```
M = cor(dataC)
corrplot(M, method = "number", title = "Correlation of streaming data")
```



Multiple regression 1:

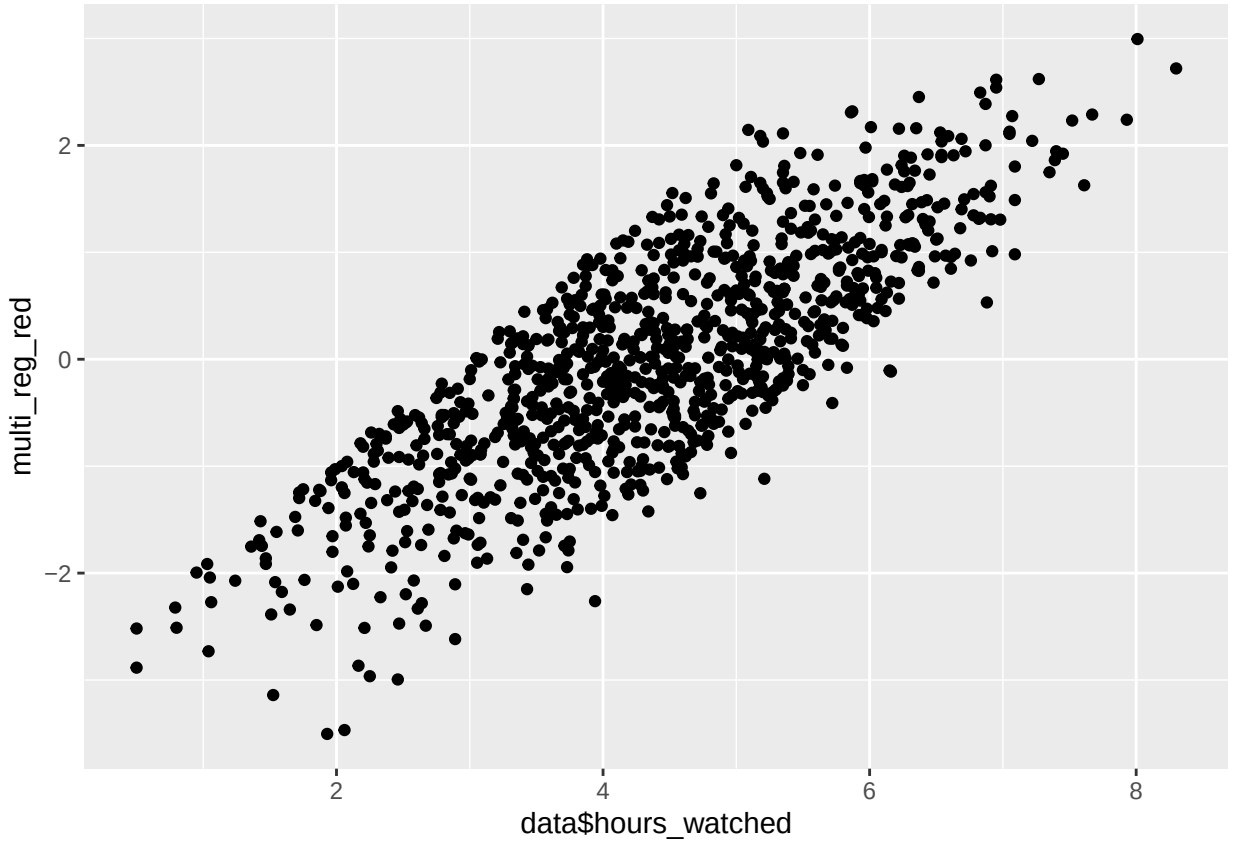
Effect of age, gender and intervention on hours watched

This regression has an adjusted correlation of 0.35 and is statistically significant.

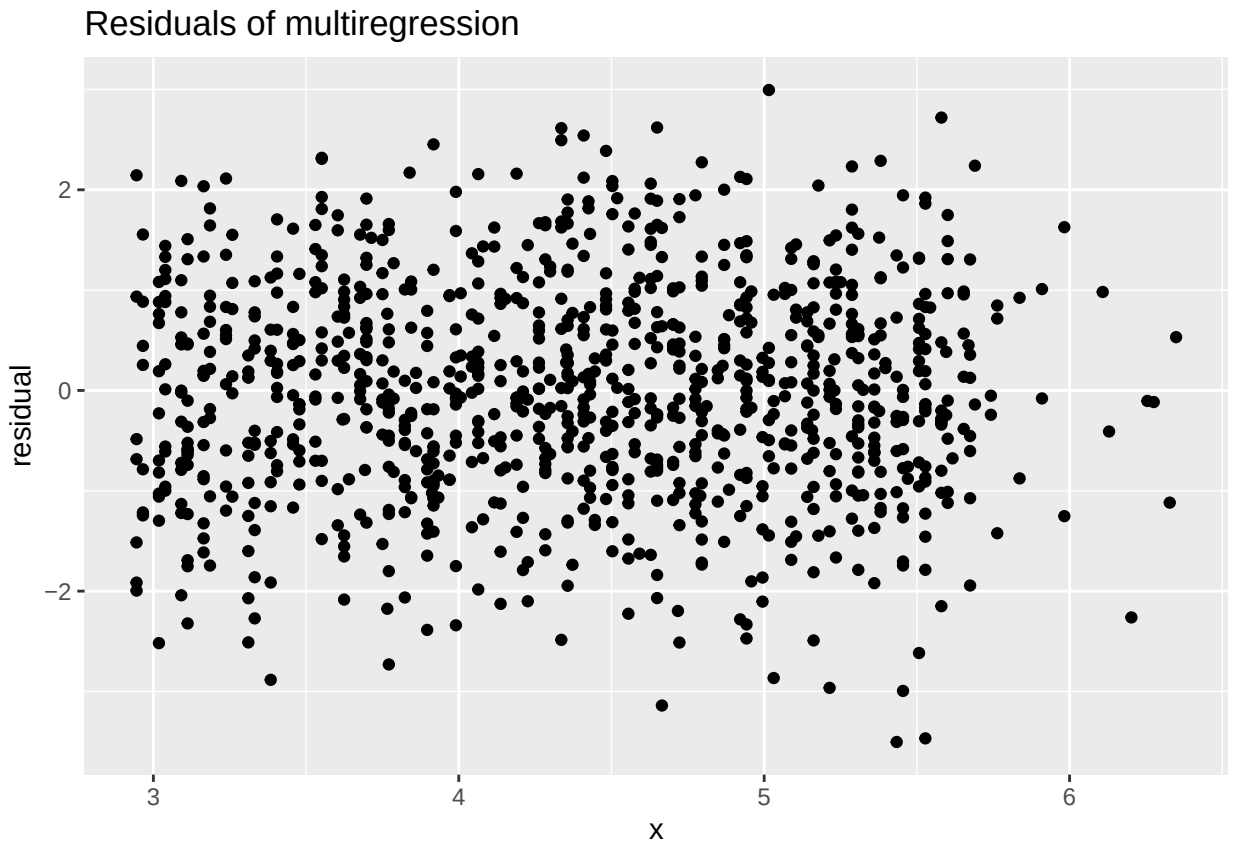
```
# do multiple regression on age, gender and hours
multi_reg <- lm(hours_watched ~ age + gender + group, data = data)
summary(multi_reg)
```

```
##
## Call:
## lm(formula = hours_watched ~ age + gender + group, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5039 -0.7289 -0.0007  0.7620  2.9946
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.971117   0.126592  55.067  < 2e-16 ***
## age         -0.073202   0.003185 -22.986  < 2e-16 ***
## genderM      0.020768   0.069167   0.300    0.764
## groupB       0.674600   0.105723   6.381 2.69e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 996 degrees of freedom
## Multiple R-squared:  0.3554, Adjusted R-squared:  0.3535
## F-statistic: 183.1 on 3 and 996 DF,  p-value: < 2.2e-16
```

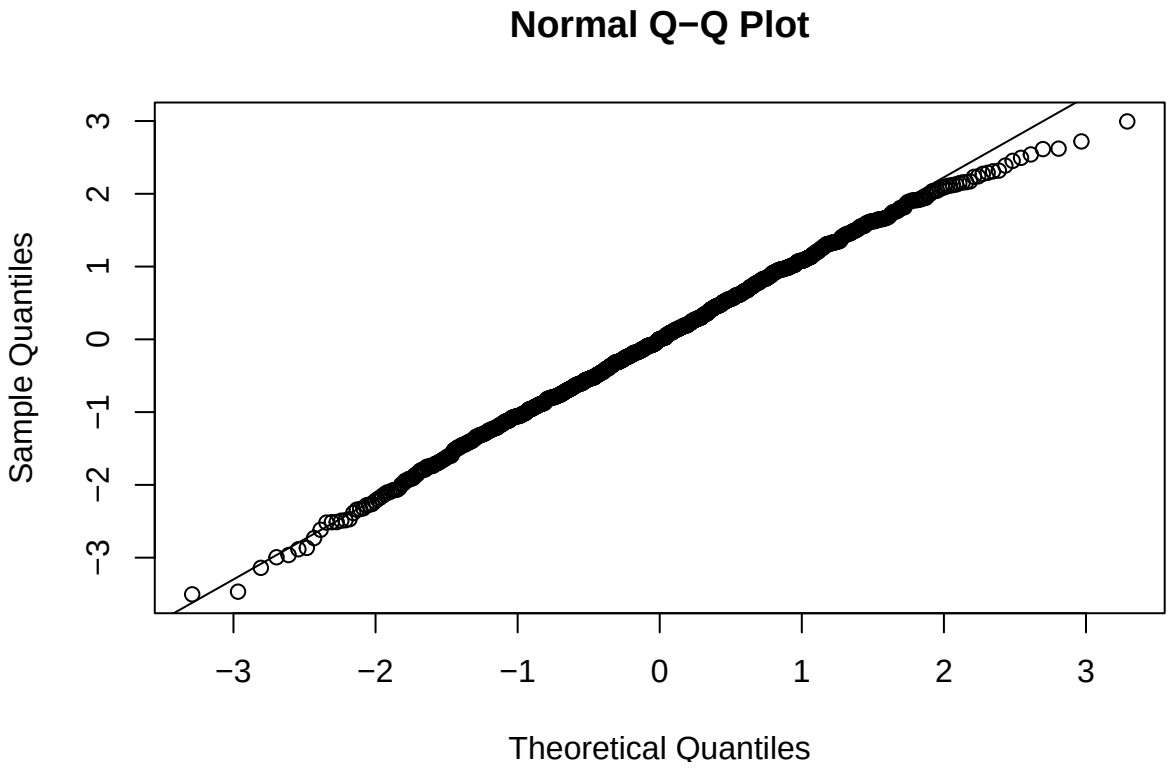
```r
multi_reg_red <- resid(multi_reg)
multi_reg_fit <- fitted(multi_reg)

# plot the regression
qplot(data$hours_watched, multi_reg_red)
```




```r
# plot the residuals
ggplot() +
    geom_point(aes(x = multi_reg_fit, y = multi_reg_red))+
    labs(title = "Residuals of multiregression",
             x = "x",
             y = "residual")
```




```r
# plot qq plot of residuals
qqnorm(multi_reg_red)
qqline(multi_reg_red)
```



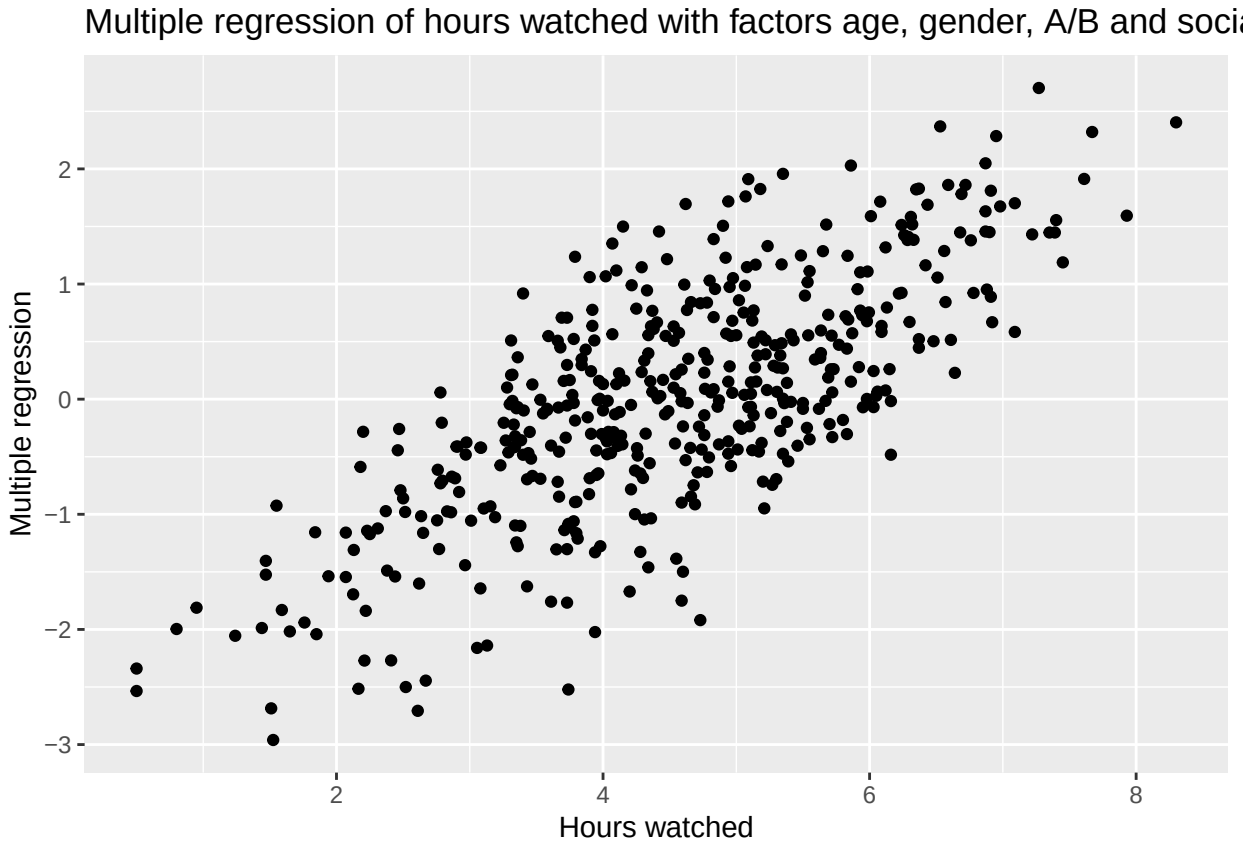
Mutiple regression 2: social metric, age, group effect on hours watched with gender accounted for

This multiple regression has gender accounted for as an interaction term and the data from group A is only in the same timeline as the group B intervention to minimise bias.

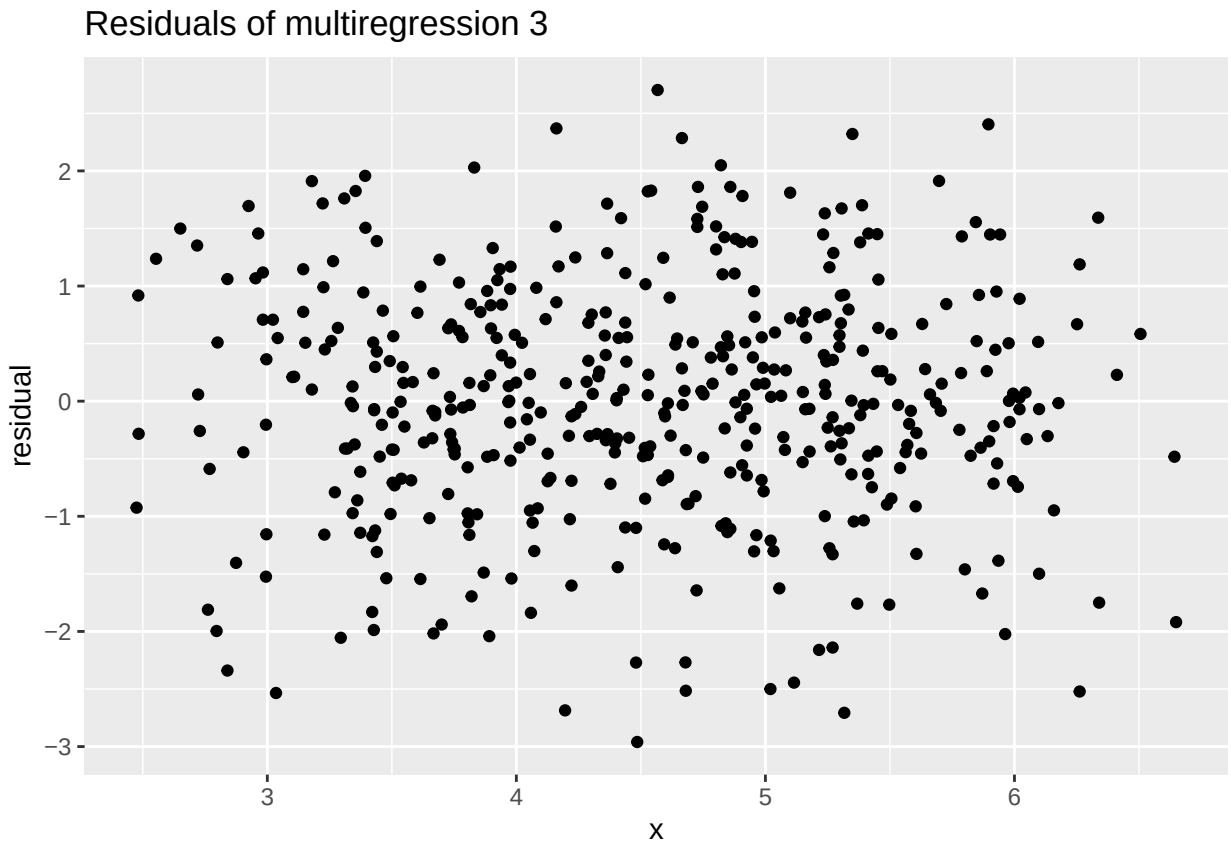
```
# plot regression 3 with interaction term
multi_reg3 <- lm(hours_watched ~ social_metric +age + group * gender, data = dataD)
summary(multi_reg3)
```

```
##
## Call:
## lm(formula = hours_watched ~ social_metric + age + group * gender,
##     data = dataD)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.96029 -0.58268 -0.00652  0.66742  2.70270
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.542056   0.266499  24.548  < 2e-16 ***
## social_metric1    0.164887   0.254820   0.647 0.517925
## social_metric2    0.521192   0.247560   2.105 0.035834 *
## social_metric3    0.356248   0.257586   1.383 0.167363
## social_metric4    0.325268   0.264463   1.230 0.219390
## social_metric5    0.663608   0.263122   2.522 0.012021 *
## social_metric6    0.938526   0.250642   3.744 0.000205 ***
## social_metric7    0.839576   0.275007   3.053 0.002405 **
## social_metric8    0.957825   0.246354   3.888 0.000117 ***
## social_metric9    1.277229   0.260522   4.903 1.34e-06 ***
## social_metric10   0.945829   0.278974   3.390 0.000761 ***
## age              -0.078213   0.004546 -17.206  < 2e-16 ***
## groupB            0.668147   0.210852   3.169 0.001638 **
## genderM           0.006865   0.115748   0.059 0.952731
## groupB:genderM   -0.046236   0.251230  -0.184 0.854069
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.032 on 437 degrees of freedom
## Multiple R-squared:  0.4545, Adjusted R-squared:  0.437
## F-statistic: 26.01 on 14 and 437 DF,  p-value: < 2.2e-16
```

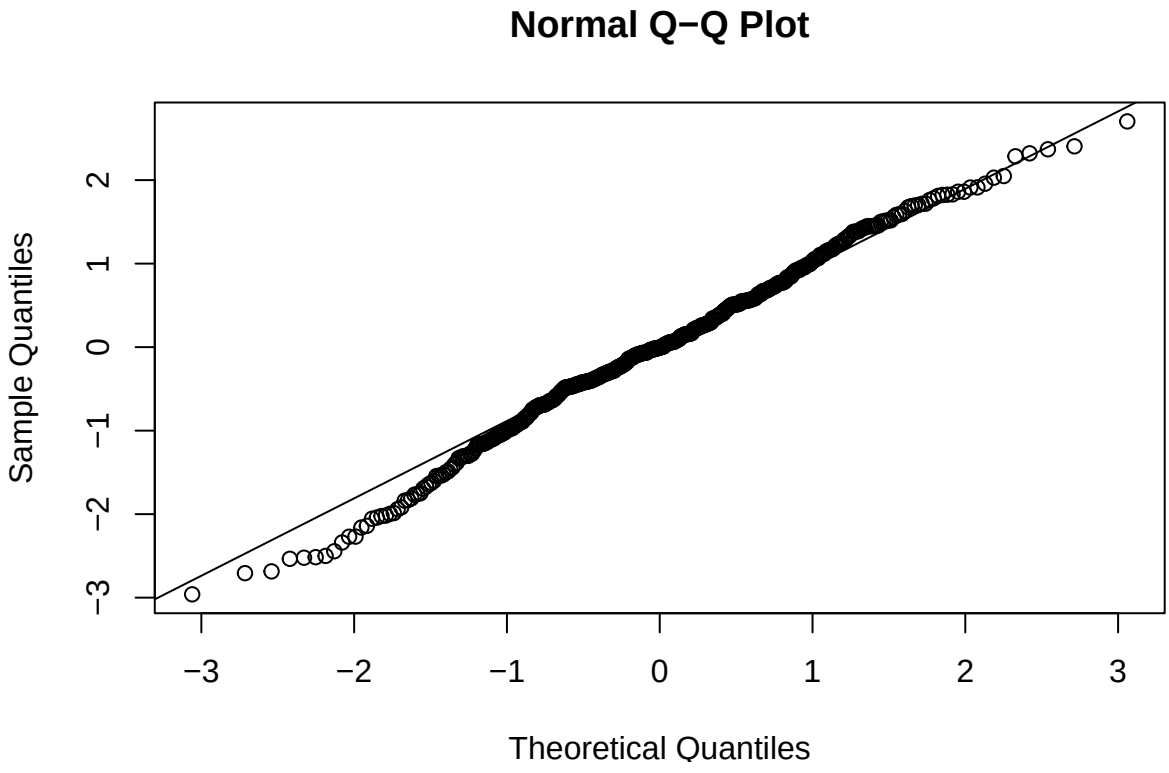
```
multi_reg_red3 <- resid(multi_reg3)
multi_reg_fit3 <- fitted(multi_reg3)

# plot the regression
qplot(dataD$hours_watched, multi_reg_red3,
      main = "Multiple regression of hours watched with factors age, gender, A/B and social metric",
      xlab = "Hours watched",
      ylab = "Multiple regression")
```




```
# plot the residuals
ggplot() +
    geom_point(aes(x = multi_reg_fit3, y = multi_reg_red3))+
    labs(title = "Residuals of multiregression 3",
             x = "x",
             y = "residual")
```



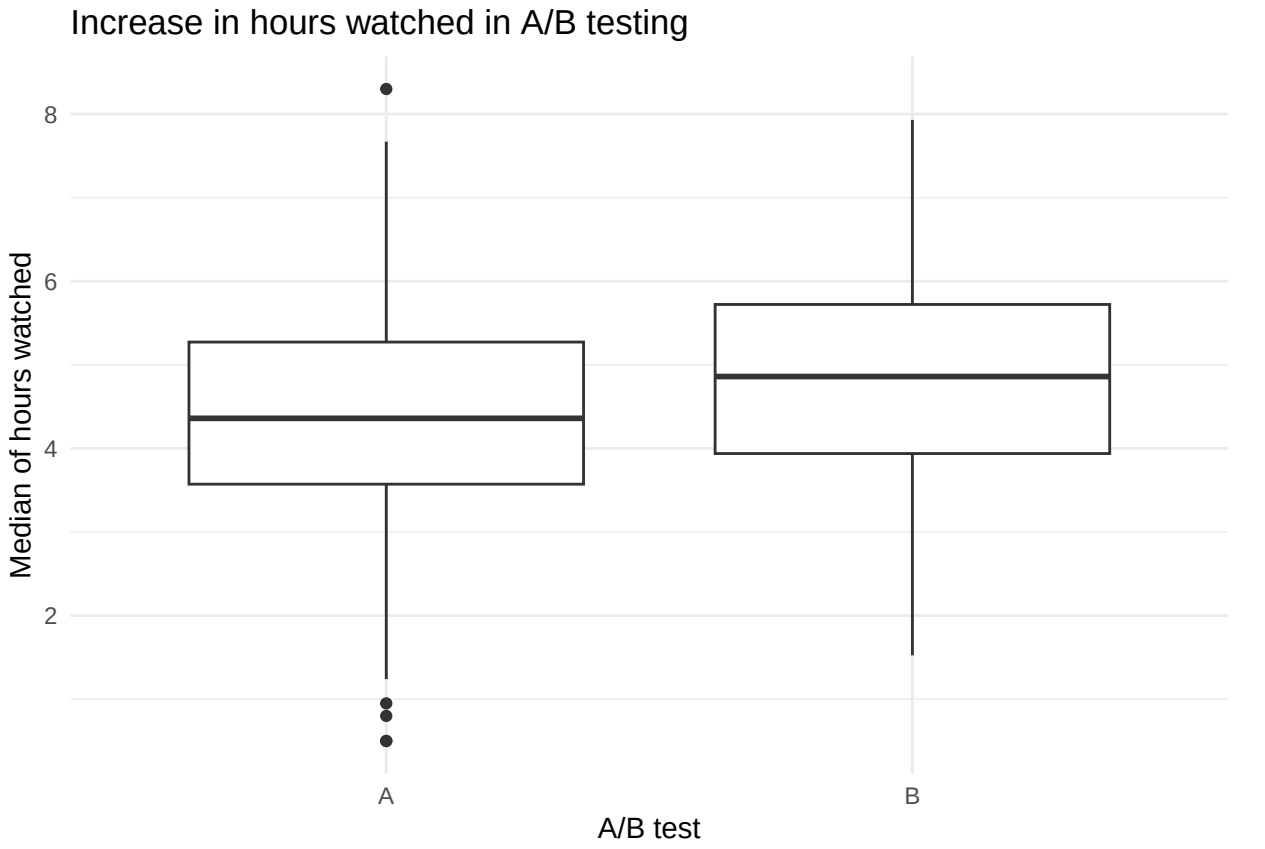

```
# plot the qq plot to assess the regression
qqnorm(multi_reg_red3)
qqline(multi_reg_red3)
```



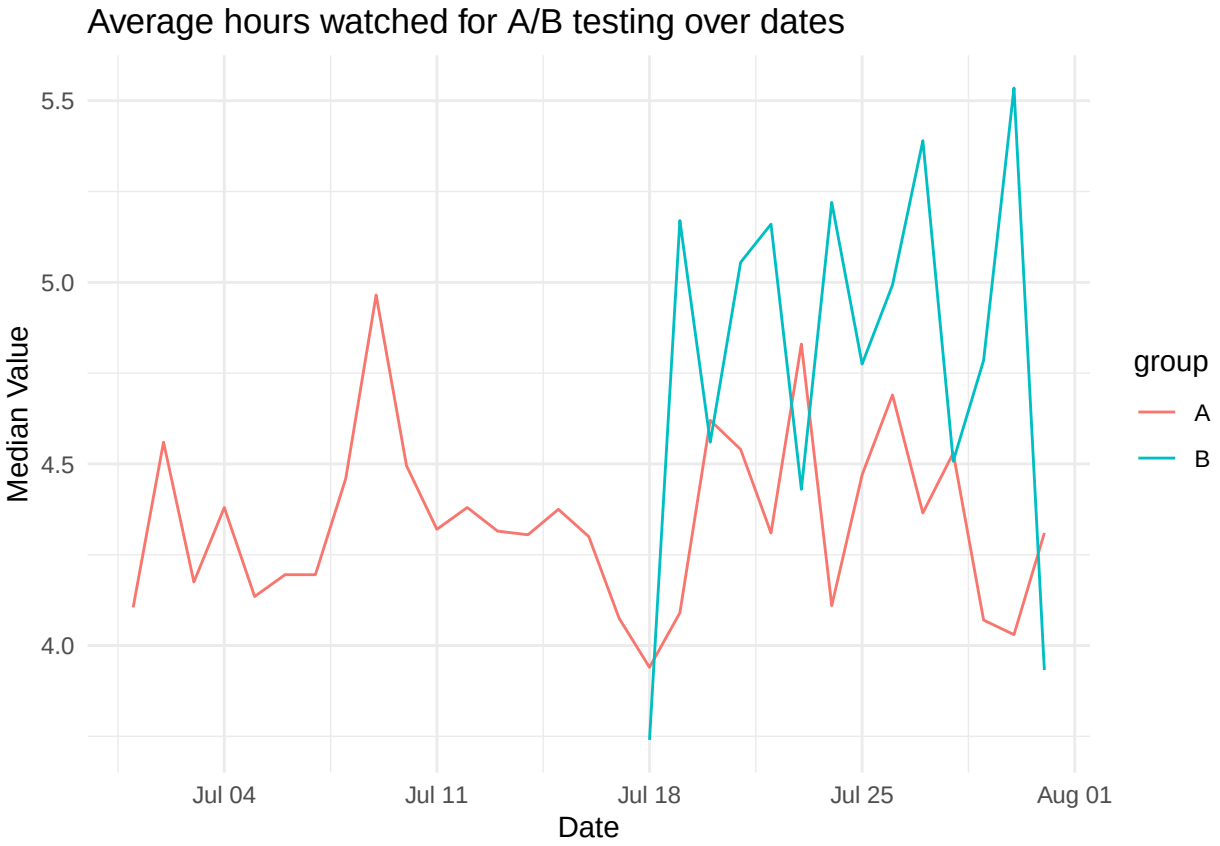
##

Plots of the results

```
# Plot differences in hours watched based on A/B
ggplot(dataD, aes(x = group, y = hours_watched)) +
  geom_boxplot() +
  ylab("Median of hours watched") +
  xlab("A/B test") +
  ggtitle("Increase in hours watched in A/B testing") +
  theme_minimal()
```




```
# plot hours watched over dates for A/B
date_hours <- data %>%
  group_by(group, date) %>%
  summarise(median_value = median(hours_watched), .groups = "drop")

ggplot(date_hours, aes(x = date, y = median_value, color = group)) +
  geom_line() +
  labs(title = "Average hours watched for A/B testing over dates",
          x = "Date",
          y = "Median Value") +
  theme_minimal()
```



References

Wickham H, François R, Henry L, Müller K, Vaughan D (2023). *dplyr: A Grammar of Data Manipulation*. R package version 1.1.2, https://CRAN.R-project.org/package=dplyr.

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Bache S, Wickham H (2022). *magrittr: A Forward-Pipe Operator for R*. R package version 2.0.3, https://CRAN.R-project.org/package=magrittr.

Wickham H, Vaughan D, Girlich M (2023). *tidyr: Tidy Messy Data*. R package version 1.3.0, https://CRAN.R-project.org/package=tidyr.

Wickham H, Hester J, Bryan J (2023). *readr: Read Rectangular Text Data*. R package version 2.1.4, https://CRAN.R-project.org/package=readr.

Xie Y (2023). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.43, https://yihui.org/knitr/.

Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963

Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL https://www.jstatsoft.org/v40/i03/.

Taiyun Wei and Viliam Simko (2021). R package 'corrplot': Visualization of a Correlation Matrix (Version 0.92). Available from https://github.com/taiyun/corrplot

Soetaert K (2021). *plot3D: Plotting Multi-Dimensional Data*. R package version 1.4, https://CRAN.R-project.org/package=plot3D.

Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. https://CRAN.R-project.org/package=stringr

John Fox and Sanford Weisberg (2019). An R Companion to Applied Regression, Third Edition. Sage. R package version 3.0-10. https://CRAN.R-project.org/package=car

Stack overflow (2018) lubridate convert decimals into months, Stack overflow, accessed 15/06/23. https://stackoverflow.com/questions/49510404/lubridate-convert-decimals-into-months