datafest

Takao

4/30/2022

We are given a large, unstructured data set from an unreleased game. The game can be played on https://user.p2p-games.org/login with the login of USERNAME: DATAFEST-77 PASSWORD: 68uNyNfm It is basically a simulation game which provides awareness to sexually transmitted diseases. The user is given choices and can go along the main character's life. The character's life depends upon the selection the users make. In total, there are 9 levels and on the way, there are many mini games that are also free choice to the users. Our goal as a data scientist is to utilize this unstructured data to draw conclusions and insights. What we base the conclusion on is up to the data scientists and further, we want to specify further investigation methods such as what the game development can do better to best promote their awareness message.

Our team decided to look into the user's priorities or the main selections that they made in the game. We firmly believed that the conclusion that we draw specializing in this section can provide insights to the specific user and on a larger scale, the same age community of the user. We will deeply analyze the specific priorities that they have and what they value the most depending on the age and gender that they have selected.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(tidyr)

mydata <- read_csv("Downloads/logs.csv",guess_max=2156600)
```

```
## Rows: 2106597 Columns: 132
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (69): school, wave, session, event_description, event_category, event_t...
## dbl  (55): row_id, player_id, event_id, event_time_dbl, stack_id, skill_leve...
## lgl   (7): object_locked, object_first_time_unlocked, object_scan_result, st...
## date  (1): date
```

```
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

Make the Data set smaller

```
logs <- mydata[,-c(13,29:32,42,54,55,85,97:104)]
```

The parameter priority_type_that_went_to_zero indicates how much they cared about certain parameters. If the number is greater, they care about it less

```
zero_priority <- data.frame(logs$priority_type_that_went_to_zero)
zero_priority <- zero_priority[!is.na(zero_priority)]
```

The number indicated in the parameter “PriorityData_type” showcases the specific categories such as health, money, school, etc

```
zero_priority[zero_priority=='PriorityData_type:0]'] <- 'Health'
zero_priority[zero_priority=='PriorityData_type:1]'] <- 'Money'
zero_priority[zero_priority=='PriorityData_type:2]'] <- 'School'
zero_priority[zero_priority=='PriorityData_type:3]'] <- 'Friends'
zero_priority[zero_priority=='PriorityData_type:4]'] <- 'Happiness'
zero_priority[zero_priority=='PriorityData_type:5]'] <- 'Family'
```

This function shows the corresponding numbers of each respected parameters (similar to table() function)

```
aggregate(zero_priority, by=list(zero_priority), length)
```

```
##     Group.1   x
## 1    Family 127
## 2   Friends 108
## 3 Happiness   8
## 4    Health  48
## 5     Money   5
## 6    School  79
```

818 indicates when they won, and 800 indicates when they started, 812 indicates when they lost

```
priority_game <- data.frame(logs$player_id[logs$event_id=='818'| logs$event_id=='812'],logs$event_id[lo
```

Changing the name of the columns

```
names(priority_game)[1] <- 'Player ID'
names(priority_game)[2] <- 'Win or Lose'
```

602 Indiciates confirming age and gender

```
test <- data.frame(logs$player_id[logs$event_id=='602'], logs$avatar_age[logs$event_id=='602'],logs$ava
```

Changing the column name again

```
names(test)[1] <- 'Player ID'
names(test)[2] <- 'Age'
names(test)[3] <- 'Gender'
```

Counting how many times they lost

```
lose <- aggregate(cbind(logs$event_id[logs$event_id=='812'])~player_id[logs$event_id=='812'], data=logs
names(lose)[1] <- 'Player ID'
names(lose)[2] <- 'Lose Count'
```

Counting how many times they won

```
win <- aggregate(cbind(logs$event_id[logs$event_id=='818'])~player_id[logs$event_id=='818'], data=logs,
names(win)[1] <- 'Player ID'
names(win)[2] <- 'Win Count'
```

```
test <- merge(x=test,y=lose,by='Player ID', all.x = TRUE)
```

```
test <- merge(x=test,y=win,by='Player ID', all.x=TRUE)
```

Making a data frame of the category that made the player lose, including the Player ID as well

```
lose_type <-data.frame(logs$player_id[logs$event_id=='812'],logs$priority_type_that_went_to_zero[logs$ev
names(lose_type)[1] <- 'Player ID'
names(lose_type)[2] <- 'Priority Type'
lose_type[lose_type=='PriorityData_type:0]'] <- 'Health'
lose_type[lose_type=='PriorityData_type:1]'] <- 'Money'
lose_type[lose_type=='PriorityData_type:2]'] <- 'School'
lose_type[lose_type=='PriorityData_type:3]'] <- 'Friends'
lose_type[lose_type=='PriorityData_type:4]'] <- 'Happiness'
lose_type[lose_type=='PriorityData_type:5]'] <- 'Family'
```

Organizing the data frame to count the times the priority type lost a player

```
#aggregate(lose_type,by=list(lose_type),length)

wer <- as.data.frame(table(lose_type$`Player ID`,lose_type$`Priority Type`))

wer$Family <- wer$Freq[wer$Var2=='Family']
wer$Friends <- wer$Freq[wer$Var2=='Friends']
wer$Health <- wer$Freq[wer$Var2 == 'Health']
wer$Money <- wer$Freq[wer$Var2 == 'Money']
wer$School <- wer$Freq[wer$Var2 == 'School']
wer$Happiness <- wer$Freq[wer$Var2=='Happiness']
wer <- wer[-c(2,3)]
```

Changing the Name again

```
L <- unique(wer)
names(L)[1] <- 'Player ID'
```

```
test <- merge(x=test,y=L,by='Player ID',all.x = TRUE)
```

```
player <- as.data.frame(unique(logs$player_id))
names(player)[1] <- 'Player ID'
# test1 <- merge(x=player,y=test1, by='Player ID',all.x=TRUE)
```

```
test[is.na(test)] <- 0
```

Find the frequencies and seeing which ones are unique and have duplicates

```
n_occur <- data.frame(table(test$`Player ID`))
n_occur[n_occur$Freq > 1,]
```

```
##        Var1 Freq
## 46  6430003    2
## 98  6506005    3
## 114 6546011    2
## 128 6567003    3
## 129 6567004    2
## 138 6606010    3
```

```
duplicate <-test[test$`Player ID` %in% n_occur$Var1[n_occur$Freq > 1],]
unique(duplicate$`Player ID` )
```

```
## [1] 6430003 6506005 6546011 6567003 6567004 6606010
```

```
test$number <- c(1:length(test$`Player ID`))
```

Remove the duplicates

```
remove <- test$number[test$`Player ID`== 6430003 | test$`Player ID` == 6506005 | test$`Player ID` == 65
remove
```

```
##  [1]  46  47  99 100 101 117 118 132 133 134 135 136 145 146 147
```

```
test1 <- test[-remove,]
```

Utilize dplyr and summarize the wincount and lose count

```
wincount <- test1%>%group_by(Age,Gender)%>%summarize(`Win Count` = sum(`Win Count`))
```

```
## `summarise()` has grouped output by 'Age'. You can override using the `.groups` argument.
```

```
losecount <- test1%>%group_by(Age,Gender)%>%summarize(`Lose Count` = sum(`Lose Count`))
```

```
## `summarise()` has grouped output by 'Age'. You can override using the `.groups` argument.
```

```
(wincount$`Win Count`+ losecount$`Lose Count`)/sum(wincount$`Win Count`+ losecount$`Lose Count`)
```

```
## [1] 0.10006807 0.07896528 0.14023145 0.14431586 0.13886998 0.12457454 0.11504425
## [8] 0.15793057
```

```
comparison <- wincount

comparison$`Total` <- (wincount$`Win Count`+ losecount$`Lose Count`)

comparison$`Success Rate` <- comparison$`Win Count`/comparison$Total

comparison
```

```
## # A tibble: 8 x 5
## # Groups:   Age [4]
##     Age Gender ‘Win Count‘ Total ‘Success Rate‘
##   <dbl> <chr>        <dbl> <dbl>          <dbl>
## 1    11 Female         109   147          0.741
## 2    11 Male            82   116          0.707
## 3    12 Female         162   206          0.786
## 4    12 Male           172   212          0.811
## 5    13 Female         145   204          0.711
## 6    13 Male           138   183          0.754
## 7    14 Female         120   169          0.710
## 8    14 Male           177   232          0.763
```

The following is a summary table with the model. There is insignificance, and the multiple R-squared is small

```
summary(lm(`Success Rate`~ `Age`*`Gender`,data = comparison))
```

```
##
## Call:
## lm(formula = ‘Success Rate‘ ~ Age * Gender, data = comparison)
##
## Residuals:
##         1         2         3         4         5         6         7         8
## -0.021181 -0.035283  0.040724  0.058053 -0.017906 -0.010257 -0.001637 -0.012513
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.94961    0.23988   3.959   0.0167 *
## Age             -0.01699    0.01911  -0.889   0.4242
## GenderMale      -0.32940    0.33925  -0.971   0.3865
## Age:GenderMale   0.02808    0.02703   1.039   0.3575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04274 on 4 degrees of freedom
## Multiple R-squared:  0.2906, Adjusted R-squared:  -0.2414
## F-statistic: 0.5463 on 3 and 4 DF,  p-value: 0.6766
```

Solvee for all gender and age combination

```
elevenF <- test1 %>% filter(Age == "11", Gender == "Female")
elevenF_fam <- sum(elevenF$Family) / sum(elevenF$`Lose Count`)
elevenF_Health <- sum(elevenF$Health) / sum(elevenF$`Lose Count`)
elevenF_Money <- sum(elevenF$Money) / sum(elevenF$`Lose Count`)
elevenF_School <- sum(elevenF$School) / sum(elevenF$`Lose Count`)
elevenF_Friends <- sum(elevenF$Friends) / sum(elevenF$`Lose Count`)
elevenF_Happiness <- sum(elevenF$Happiness) / sum(elevenF$`Lose Count`)
elevenF_mat <- cbind("Family" = elevenF_fam, "Friends" = elevenF_Friends,
                     "Happiness" = elevenF_Happiness, "Health" = elevenF_Health,
                      "Money" = elevenF_Money, "School" = elevenF_School)
#barplot(elevenF_mat)

elevenF_mat <- as.data.frame(elevenF_mat)
elevenF_mat$Group <- '11F'
```

```
elevenM <- test1 %>% filter(Age == "11", Gender == "Male")
elevenM_fam <- sum(elevenM$Family) / sum(elevenM$`Lose Count`)
elevenM_Health <- sum(elevenM$Health) / sum(elevenM$`Lose Count`)
elevenM_Money <- sum(elevenM$Money) / sum(elevenM$`Lose Count`)
elevenM_School <- sum(elevenM$School) / sum(elevenM$`Lose Count`)
elevenM_Friends <- sum(elevenM$Friends) / sum(elevenM$`Lose Count`)
elevenM_Happiness <- sum(elevenM$Happiness) / sum(elevenM$`Lose Count`)
elevenM_mat <- cbind("Family" = elevenM_fam, "Friends" = elevenM_Friends,
                     "Happiness" = elevenM_Happiness, "Health" = elevenM_Health,
                      "Money" = elevenM_Money, "School" = elevenM_School)
#barplot(elevenM_mat)

elevenM_mat<-as.data.frame(elevenM_mat)
elevenM_mat$Group <- '11M'
```

```
twelveF <- test1 %>% filter(Age == "12", Gender == "Female")
twelveF_fam <- sum(twelveF$Family) / sum(twelveF$`Lose Count`)
twelveF_Health <- sum(twelveF$Health) / sum(twelveF$`Lose Count`)
twelveF_Money <- sum(twelveF$Money) / sum(twelveF$`Lose Count`)
twelveF_School <- sum(twelveF$School) / sum(twelveF$`Lose Count`)
twelveF_Friends <- sum(twelveF$Friends) / sum(twelveF$`Lose Count`)
twelveF_Happiness <- sum(twelveF$Happiness) / sum(twelveF$`Lose Count`)
twelveF_mat <- cbind("Family" = twelveF_fam, "Friends" = twelveF_Friends,
                     "Happiness" = twelveF_Happiness, "Health" = twelveF_Health,
                      "Money" = twelveF_Money, "School" = twelveF_School)
twelveF_mat <- as.data.frame((twelveF_mat))
twelveF_mat$Group <- '12F'
#barplot(twelveF_mat)
```

```
twelveM <- test1 %>% filter(Age == "12", Gender == "Male")
twelveM_fam <- sum(twelveM$Family) / sum(twelveM$`Lose Count`)
twelveM_Health <- sum(twelveM$Health) / sum(twelveM$`Lose Count`)
twelveM_Money <- sum(twelveM$Money) / sum(twelveM$`Lose Count`)
twelveM_School <- sum(twelveM$School) / sum(twelveM$`Lose Count`)
twelveM_Friends <- sum(twelveM$Friends) / sum(twelveM$`Lose Count`)
twelveM_Happiness <- sum(twelveM$Happiness) / sum(twelveM$`Lose Count`)
```

```
twelveM_mat <- cbind("Family" = twelveM_fam, "Friends" = twelveM_Friends,
                     "Happiness" = twelveM_Happiness, "Health" = twelveM_Health,
                      "Money" = twelveM_Money, "School" = twelveM_School)

twelveM_mat <- as.data.frame(twelveM_mat)
twelveM_mat$Group <- '12M'
#barplot(twelveM_mat)
```

```
thirteenF <- test1 %>% filter(Age == "13", Gender == "Female")
thirteenF_fam <- sum(thirteenF$Family) / sum(thirteenF$`Lose Count`)
thirteenF_Health <- sum(thirteenF$Health) / sum(thirteenF$`Lose Count`)
thirteenF_Money <- sum(thirteenF$Money) / sum(thirteenF$`Lose Count`)
thirteenF_School <- sum(thirteenF$School) / sum(thirteenF$`Lose Count`)
thirteenF_Friends <- sum(thirteenF$Friends) / sum(thirteenF$`Lose Count`)
thirteenF_Happiness <- sum(thirteenF$Happiness) / sum(thirteenF$`Lose Count`)
thirteenF_mat <- cbind("Family" = thirteenF_fam, "Friends" = thirteenF_Friends,
                     "Happiness" = thirteenF_Happiness, "Health" = thirteenF_Health,
                      "Money" = thirteenF_Money, "School" = thirteenF_School)

thirteenF_mat <- as.data.frame(thirteenF_mat)
thirteenF_mat$Group <- '13F'
```

```
thirteenM <- test1 %>% filter(Age == "13", Gender == "Male")
thirteenM_fam <- sum(thirteenM$Family) / sum(thirteenM$`Lose Count`)
thirteenM_Health <- sum(thirteenM$Health) / sum(thirteenM$`Lose Count`)
thirteenM_Money <- sum(thirteenM$Money) / sum(thirteenM$`Lose Count`)
thirteenM_School <- sum(thirteenM$School) / sum(thirteenM$`Lose Count`)
thirteenM_Friends <- sum(thirteenM$Friends) / sum(thirteenM$`Lose Count`)
thirteenM_Happiness <- sum(thirteenM$Happiness) / sum(thirteenM$`Lose Count`)
thirteenM_mat <- cbind("Family" = thirteenM_fam, "Friends" = thirteenM_Friends,
                     "Happiness" = thirteenM_Happiness, "Health" = thirteenM_Health,
                      "Money" = thirteenM_Money, "School" = thirteenM_School)

thirteenM_mat <- as.data.frame(thirteenM_mat)
thirteenM_mat$Group <- '13M'
```

```
fourteenF <- test1 %>% filter(Age == "14", Gender == "Female")
fourteenF_fam <- sum(fourteenF$Family) / sum(fourteenF$`Lose Count`)
fourteenF_Health <- sum(fourteenF$Health) / sum(fourteenF$`Lose Count`)
fourteenF_Money <- sum(fourteenF$Money) / sum(fourteenF$`Lose Count`)
fourteenF_School <- sum(fourteenF$School) / sum(fourteenF$`Lose Count`)
fourteenF_Friends <- sum(fourteenF$Friends) / sum(fourteenF$`Lose Count`)
fourteenF_Happiness <- sum(fourteenF$Happiness) / sum(fourteenF$`Lose Count`)
fourteenF_mat <- cbind("Family" = fourteenF_fam, "Friends" = fourteenF_Friends,
                     "Happiness" = fourteenF_Happiness, "Health" = fourteenF_Health,
                      "Money" = fourteenF_Money, "School" = fourteenF_School)

fourteenF_mat <- as.data.frame(fourteenF_mat)
fourteenF_mat$Group <- '14F'
```

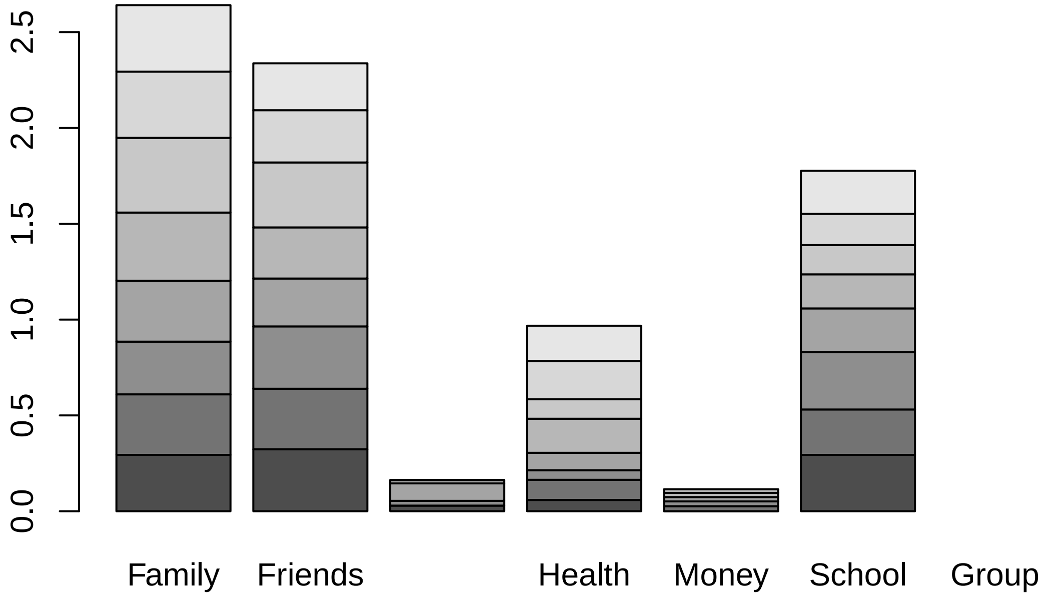
```
fourteenM <- test1 %>% filter(Age == "14", Gender == "Male")
fourteenM_fam <- sum(fourteenM$Family) / sum(fourteenM$`Lose Count`)
```

```
fourteenM_Health <- sum(fourteenM$Health) / sum(fourteenM$`Lose Count`)
fourteenM_Money <- sum(fourteenM$Money) / sum(fourteenM$`Lose Count`)
fourteenM_School <- sum(fourteenM$School) / sum(fourteenM$`Lose Count`)
fourteenM_Friends <- sum(fourteenM$Friends) / sum(fourteenM$`Lose Count`)
fourteenM_Happiness <- sum(fourteenM$Happiness) / sum(fourteenM$`Lose Count`)
fourteenM_mat <- cbind("Family" = fourteenM_fam, "Friends" = fourteenM_Friends,
                     "Happiness" = fourteenM_Happiness, "Health" = fourteenM_Health,
                      "Money" = fourteenM_Money, "School" = fourteenM_School)

fourteenM_mat <- as.data.frame(fourteenM_mat)
fourteenM_mat$Group <- '14M'
```

Building a barplot, we have that

```
library(ggplot2)
bar <- as.data.frame(rbind(elevenM_mat, elevenF_mat, twelveM_mat,twelveF_mat,thirteenM_mat, thirteenF_ma
barplot(as.matrix(bar))
```

```
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
```



```
elevenF <- test1 %>% filter(Age == "11", Gender == "Female")
elevenF_fam <- sum(elevenF$Family)
elevenF_Health <- sum(elevenF$Health)
elevenF_Money <- sum(elevenF$Money)
```

```r
elevenF_School <- sum(elevenF$School)
elevenF_Friends <- sum(elevenF$Friends)
elevenF_Happiness <- sum(elevenF$Happiness)
elevenF_mat <- cbind("Family" = elevenF_fam, "Friends" = elevenF_Friends,
                     "Happiness" = elevenF_Happiness, "Health" = elevenF_Health,
                      "Money" = elevenF_Money, "School" = elevenF_School)
#barplot(elevenF_mat)

#elevenF_mat <- as.data.frame(elevenF_mat)
#elevenF_mat$Group <- '11F'
```

```r
elevenM <- test1 %>% filter(Age == "11", Gender == "Male")
elevenM_fam <- sum(elevenM$Family)
elevenM_Health <- sum(elevenM$Health)
elevenM_Money <- sum(elevenM$Money)
elevenM_School <- sum(elevenM$School)
elevenM_Friends <- sum(elevenM$Friends)
elevenM_Happiness <- sum(elevenM$Happiness)
elevenM_mat <- cbind("Family" = elevenM_fam, "Friends" = elevenM_Friends,
                     "Happiness" = elevenM_Happiness, "Health" = elevenM_Health,
                      "Money" = elevenM_Money, "School" = elevenM_School)
#barplot(elevenM_mat)

#elevenM_mat<-as.data.frame(elevenM_mat)
#elevenM_mat$Group <- '11M'
```

```r
twelveF <- test1 %>% filter(Age == "12", Gender == "Female")
twelveF_fam <- sum(twelveF$Family)
twelveF_Health <- sum(twelveF$Health)
twelveF_Money <- sum(twelveF$Money)
twelveF_School <- sum(twelveF$School)
twelveF_Friends <- sum(twelveF$Friends)
twelveF_Happiness <- sum(twelveF$Happiness)
twelveF_mat <- cbind("Family" = twelveF_fam, "Friends" = twelveF_Friends,
                     "Happiness" = twelveF_Happiness, "Health" = twelveF_Health,
                      "Money" = twelveF_Money, "School" = twelveF_School)
#twelveF_mat <- as.data.frame((twelveF_mat))
#twelveF_mat$Group <- '12F'
#barplot(twelveF_mat)
```

```r
twelveM <- test1 %>% filter(Age == "12", Gender == "Male")
twelveM_fam <- sum(twelveM$Family)
twelveM_Health <- sum(twelveM$Health)
twelveM_Money <- sum(twelveM$Money)
twelveM_School <- sum(twelveM$School)
twelveM_Friends <- sum(twelveM$Friends)
twelveM_Happiness <- sum(twelveM$Happiness)
twelveM_mat <- cbind("Family" = twelveM_fam, "Friends" = twelveM_Friends,
                     "Happiness" = twelveM_Happiness, "Health" = twelveM_Health,
                      "Money" = twelveM_Money, "School" = twelveM_School)

#twelveM_mat <- as.data.frame(twelveM_mat)
```

```
#twelveM_mat$Group <- '12M'
#barplot(twelveM_mat)
```

```
thirteenF <- test1 %>% filter(Age == "13", Gender == "Female")
thirteenF_fam <- sum(thirteenF$Family)
thirteenF_Health <- sum(thirteenF$Health)
thirteenF_Money <- sum(thirteenF$Money)
thirteenF_School <- sum(thirteenF$School)
thirteenF_Friends <- sum(thirteenF$Friends)
thirteenF_Happiness <- sum(thirteenF$Happiness)
thirteenF_mat <- cbind("Family" = thirteenF_fam, "Friends" = thirteenF_Friends,
                     "Happiness" = thirteenF_Happiness, "Health" = thirteenF_Health,
                      "Money" = thirteenF_Money, "School" = thirteenF_School)

#thirteenF_mat <- as.data.frame(thirteenF_mat)
#thirteenF_mat$Group <- '13F'
```

```
thirteenM <- test1 %>% filter(Age == "13", Gender == "Male")
thirteenM_fam <- sum(thirteenM$Family)
thirteenM_Health <- sum(thirteenM$Health)
thirteenM_Money <- sum(thirteenM$Money)
thirteenM_School <- sum(thirteenM$School)
thirteenM_Friends <- sum(thirteenM$Friends)
thirteenM_Happiness <- sum(thirteenM$Happiness)
thirteenM_mat <- cbind("Family" = thirteenM_fam, "Friends" = thirteenM_Friends,
                     "Happiness" = thirteenM_Happiness, "Health" = thirteenM_Health,
                      "Money" = thirteenM_Money, "School" = thirteenM_School)

#thirteenM_mat <- as.data.frame(thirteenM_mat)
#thirteenM_mat$Group <- '13M'
```

```
fourteenF <- test1 %>% filter(Age == "14", Gender == "Female")
fourteenF_fam <- sum(fourteenF$Family)
fourteenF_Health <- sum(fourteenF$Health)
fourteenF_Money <- sum(fourteenF$Money)
fourteenF_School <- sum(fourteenF$School)
fourteenF_Friends <- sum(fourteenF$Friends)
fourteenF_Happiness <- sum(fourteenF$Happiness)
fourteenF_mat <- cbind("Family" = fourteenF_fam, "Friends" = fourteenF_Friends,
                     "Happiness" = fourteenF_Happiness, "Health" = fourteenF_Health,
                      "Money" = fourteenF_Money, "School" = fourteenF_School)

#fourteenF_mat <- as.data.frame(fourteenF_mat)
#fourteenF_mat$Group <- '14F'
```

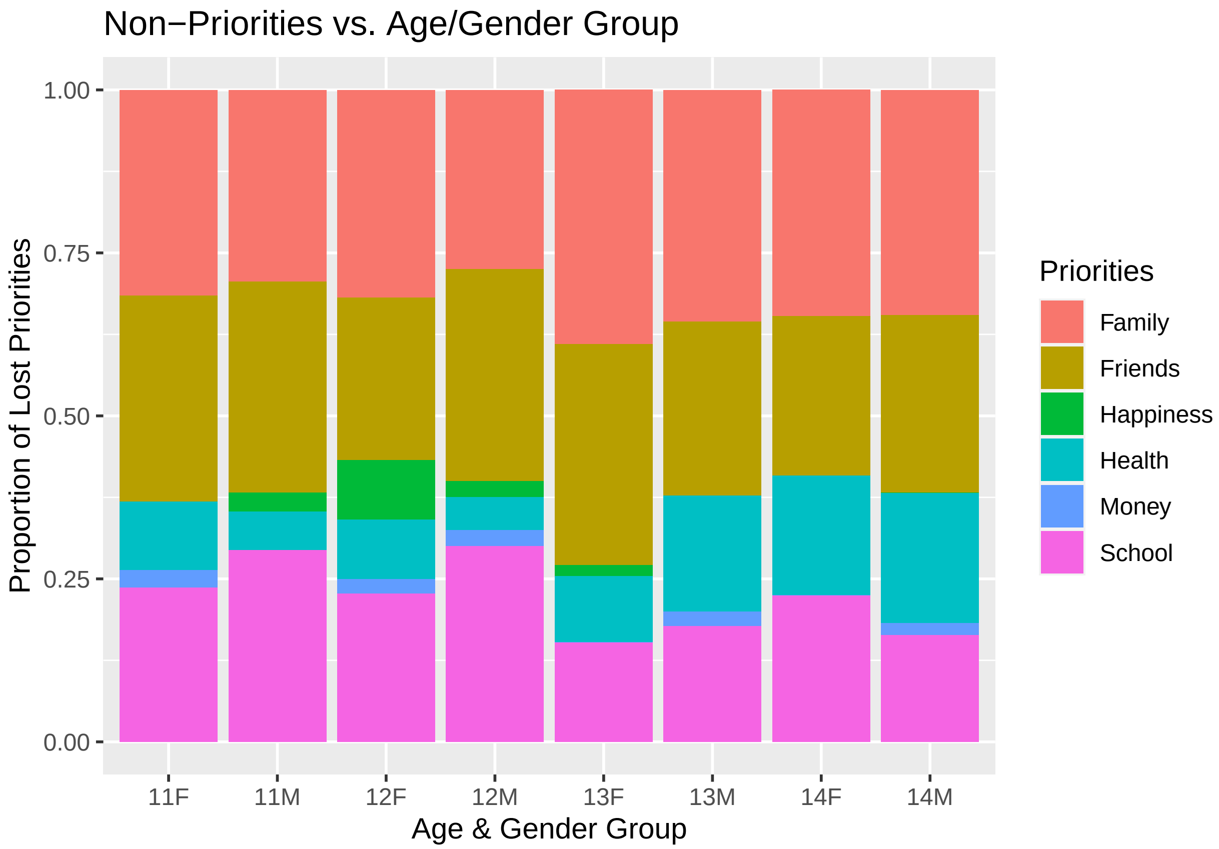
```
fourteenM <- test1 %>% filter(Age == "14", Gender == "Male")
fourteenM_fam <- sum(fourteenM$Family)
fourteenM_Health <- sum(fourteenM$Health)
fourteenM_Money <- sum(fourteenM$Money)
fourteenM_School <- sum(fourteenM$School)
fourteenM_Friends <- sum(fourteenM$Friends)
fourteenM_Happiness <- sum(fourteenM$Happiness)
```

```
fourteenM_mat <- cbind("Family" = fourteenM_fam, "Friends" = fourteenM_Friends,
                       "Happiness" = fourteenM_Happiness, "Health" = fourteenM_Health,
                        "Money" = fourteenM_Money, "School" = fourteenM_School)

#fourteenM_mat <- as.data.frame(fourteenM_mat)
#fourteenM_mat$Group <- '14M'
```

```
specie <- c(rep("11M",6),rep("11F",6),rep("12M",6),rep("12F",6),rep("13M",6),rep("13F",6),rep("14M",6),
condition <- rep(c("Family","Friends","Happiness","Health","Money","School"),4)
value <- c(elevenM_mat,elevenF_mat,twelveM_mat,twelveF_mat,thirteenM_mat,thirteenF_mat, fourteenM_mat,
data <- data.frame(specie,condition,value)
```

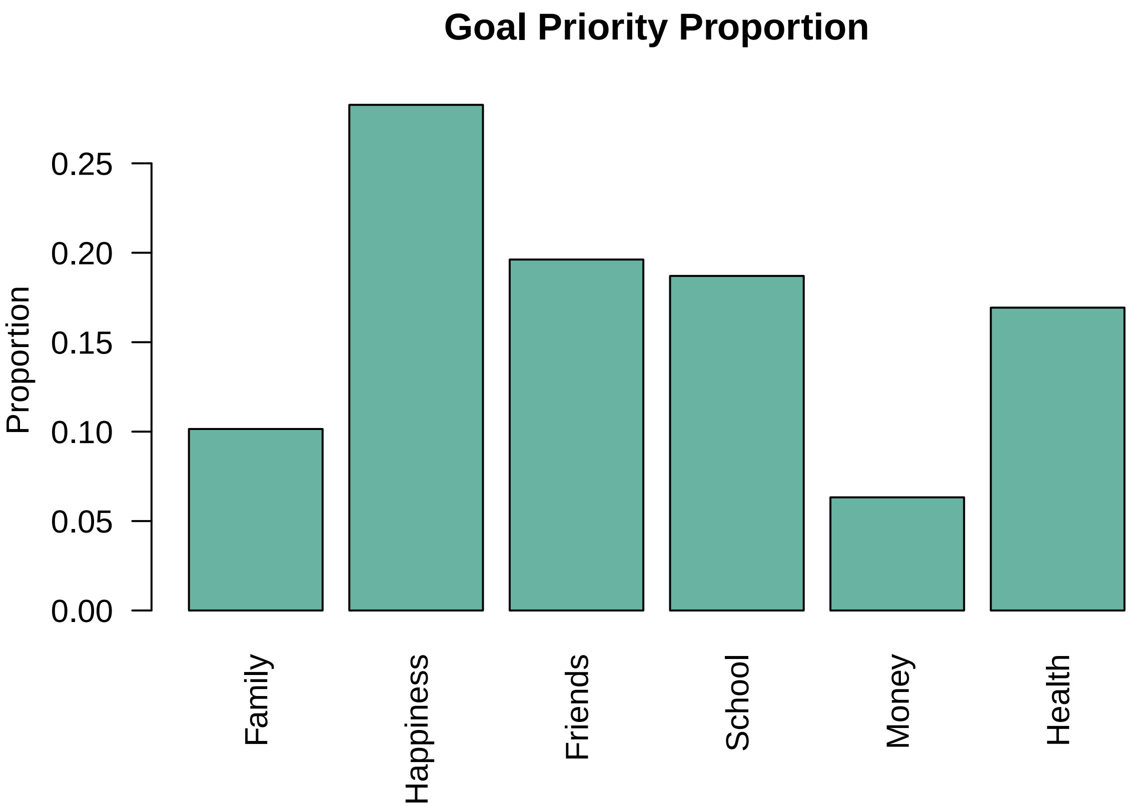
This gives the segmented bar graph with the Age and gender groups on the x-axis and the corresponding proportions on the y-axis

```
ggplot(data, aes(fill=condition, y=value, x=specie)) +
    geom_bar(position="fill", stat="identity") + xlab("Age & Gender Group") + ylab("Proportion of Lost
```



We can see how the goal prioroty proportion differs. We will make a barplot with the priorities and their corresponding proportion.

```
goal1 <- as.data.frame(logs %>% filter(event_id == 811))
health <- sum(goal1$goal_1_priority_type == "PriorityData_type:0") + sum(goal1$goal_2_priority_type ==

money <- sum(goal1$goal_1_priority_type == "PriorityData_type:1")+ sum(goal1$goal_2_priority_type == "P
school <- sum(goal1$goal_1_priority_type == "PriorityData_type:2")+ sum(goal1$goal_2_priority_type == "

friends <- sum(goal1$goal_1_priority_type == "PriorityData_type:3")+ sum(goal1$goal_2_priority_type ==

happiness <- sum(goal1$goal_1_priority_type == "PriorityData_type:4")+ sum(goal1$goal_2_priority_type ==

family <- sum(goal1$goal_1_priority_type == "PriorityData_type:5")+ sum(goal1$goal_2_priority_type == "
total <- sum(family, happiness, friends, school, money, health)
prioritymat <- cbind("family" = family / total, "happiness" = happiness / total, "friends" = friends / t

barplot(prioritymat, col = c("#69b3a2"), names.arg = c("Family","Happiness","Friends","School","Money",
```



Calculating based on the levels,

LV 0 Average

```
# tally(c(1, 2, 3))
logsmod <- logs[,c(2, 7,11)]
logs[logs$player_id == 6427001 & (logs$event_id==818 | logs$event_id==800 |logs$event_id==812) & logs$m
```

```
## # A tibble: 3 x 115
##   row_id player_id school wave    session date       event_id event_description
##    <dbl>     <dbl> <chr>  <chr>   <chr>   <date>        <dbl> <chr>
## 1    472   6427001 3561   3561 (~ Sessio~ 2013-03-07      800 Player closes int~
## 2    632   6427001 3561   3561 (~ Sessio~ 2013-03-07      800 Player closes int~
## 3    744   6427001 3561   3561 (~ Sessio~ 2013-03-07      818 Player sees win p~
## # ... with 107 more variables: event_category <chr>, event_time <chr>,
## #   event_time_dbl <dbl>, stack_id <dbl>, data_headers <chr>,
## #   data_values <chr>, skill_level_know <dbl>, skill_level_priority <dbl>,
## #   skill_level_people <dbl>, skill_level_refusal <dbl>, skill_level_me <dbl>,
## #   minigame_id <dbl>, minigame_level <dbl>, skill_id <chr>,
## #   old_skill_point <dbl>, new_skill_point <dbl>, object_id <chr>,
## #   object_locked <lgl>, object_first_time_unlocked <lgl>, sense_id <dbl>, ...
```

```
logsmod0 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==0,]
logsmod0 <- logsmod0[,c(2,7,11)]
logsmod0$number <- c(1:length(logsmod0$player_id))
rows <- which(logsmod0$event_id == 818)
level0_800 <- rows - 1

lv0_avg<-sum((logsmod0$event_time_dbl[rows]-logsmod0$event_time_dbl[level0_800]))/length(level0_800)
lv0_avg
```

```
## [1] 178.4333
```

LV 1 Average

```
logsmod1 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==1,]
logsmod1 <- logsmod1[,c(2,7,11)]
logsmod1$number <- c(1:length(logsmod1$player_id))
rows <- which(logsmod1$event_id == 818)
level0_800 <- rows - 1

lv1_avg<-sum((logsmod1$event_time_dbl[rows]-logsmod1$event_time_dbl[level0_800]))/length(level0_800)
lv1_avg
```

```
## [1] 163.9329
```

LV 2 Average

```
logsmod2 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==2,]
logsmod2 <- logsmod2[,c(2,7,11)]
logsmod2$number <- c(1:length(logsmod2$player_id))
rows <- which(logsmod2$event_id == 818)
level0_800 <- rows - 1

lv2_avg<-sum((logsmod2$event_time_dbl[rows]-logsmod2$event_time_dbl[level0_800]))/length(level0_800)
lv2_avg
```

```
## [1] 211.2835
```

LV 3 Average

```
logsmod3 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==3,]
logsmod3 <- logsmod3[,c(2,7,11)]
logsmod3$number <- c(1:length(logsmod3$player_id))
rows <- which(logsmod3$event_id == 818)
level0_800 <- rows - 1

lv3_avg<-sum((logsmod3$event_time_dbl[rows]-logsmod3$event_time_dbl[level0_800]))/length(level0_800)
lv3_avg
```

```
## [1] 214.2613
```

LV 4 Average

```
logsmod4 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==4,]
logsmod4 <- logsmod4[,c(2,7,11)]
logsmod4$number <- c(1:length(logsmod4$player_id))
rows <- which(logsmod4$event_id == 818)
level0_800 <- rows - 1

lv4_avg<-sum((logsmod4$event_time_dbl[rows]-logsmod4$event_time_dbl[level0_800]))/length(level0_800)
lv4_avg
```

```
## [1] 191.1029
```

LV 5 Average

```
logsmod5 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==5,]
logsmod5 <- logsmod5[,c(2,7,11)]
logsmod5$number <- c(1:length(logsmod5$player_id))
rows <- which(logsmod5$event_id == 818)
level0_800 <- rows - 1

lv5_avg<-sum((logsmod5$event_time_dbl[rows]-logsmod5$event_time_dbl[level0_800]))/length(level0_800)
lv5_avg
```

```
## [1] 163.3028
```

LV 6 Average

```
logsmod6 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==6,]
logsmod6 <- logsmod6[,c(2,7,11)]
logsmod6$number <- c(1:length(logsmod6$player_id))
rows <- which(logsmod6$event_id == 818)
level0_800 <- rows - 1

lv6_avg<-sum((logsmod6$event_time_dbl[rows]-logsmod6$event_time_dbl[level0_800]))/length(level0_800)
lv6_avg
```

```
## [1] 171.4268
```

LV 7 Average

```
logsmod7 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==7,]
logsmod7 <- logsmod7[,c(2,7,11)]
logsmod7$number <- c(1:length(logsmod7$player_id))
rows <- which(logsmod7$event_id == 818)
level0_800 <- rows - 1

lv7_avg<-sum((logsmod7$event_time_dbl[rows]-logsmod7$event_time_dbl[level0_800]))/length(level0_800)
lv7_avg
```

```
## [1] 165.253
```

LV 8 Average

```
logsmod8 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==8,]
logsmod8 <- logsmod8[,c(2,7,11)]
logsmod8$number <- c(1:length(logsmod8$player_id))
rows <- which(logsmod8$event_id == 818)
level0_800 <- rows - 1

lv8_avg<-sum((logsmod8$event_time_dbl[rows]-logsmod8$event_time_dbl[level0_800]))/length(level0_800)
lv8_avg
```

```
## [1] 170.0097
```

LV 9 Average

```
logsmod9 <- logs[(logs$event_id==818 | logs$event_id==800) & logs$minigame_level==9,]
logsmod9 <- logsmod9[,c(2,7,11)]
logsmod9$number <- c(1:length(logsmod9$player_id))
rows <- which(logsmod9$event_id == 818)
level0_800 <- rows - 1
```

```
lv9_avg<-sum((logsmod9$event_time_dbl[rows]-logsmod9$event_time_dbl[level0_800]))/length(level0_800)
lv9_avg
```

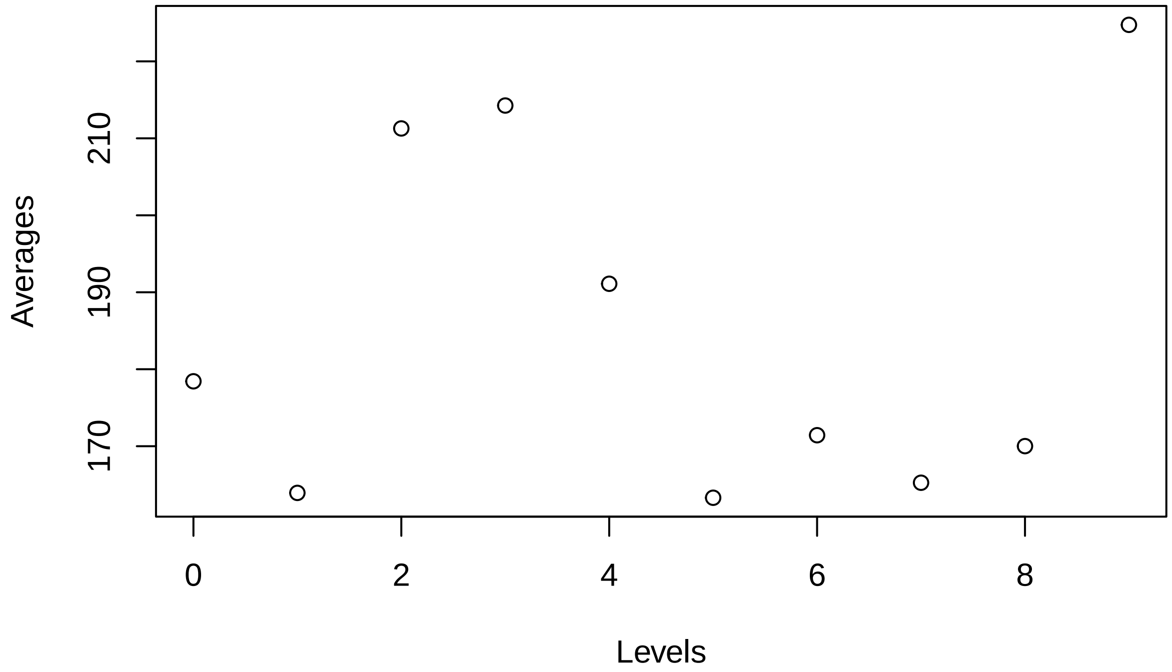
```
## [1] 224.7426
```

Matrix

```
averages <- c(lv0_avg,lv1_avg,lv2_avg,lv3_avg,lv4_avg,lv5_avg,lv6_avg,lv7_avg,lv8_avg,lv9_avg)
df1 <- data.frame("Levels" = 0:9, "Averages" = averages)
df1
```

```
##    Levels Averages
## 1       0 178.4333
## 2       1 163.9329
## 3       2 211.2835
## 4       3 214.2613
## 5       4 191.1029
## 6       5 163.3028
## 7       6 171.4268
## 8       7 165.2530
## 9       8 170.0097
## 10      9 224.7426
```

```
plot(Averages ~ Levels, data = df1)
```



Above is the graph that indicates the average finish time by the various levels