Social Media and Mental Health Exploratory Data Analysis

Corey Land

2025-11-25

This data analysis looks into exploring the relatonship between social media usage and mental health variables include: User ID, Age, Gender, Daily Screen Time,Sleep Quality, Stress Level, Days without Social Media,Exercise Frequency,Social Media Platform and Happiness Index

Load Libraries

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'tibble' was built under R version 4.4.3
```

```
## Warning: package 'tidyr' was built under R version 4.4.3
```

```
## Warning: package 'readr' was built under R version 4.4.3
```

```
## Warning: package 'purrr' was built under R version 4.4.3
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
## Warning: package 'stringr' was built under R version 4.4.3
```

```
## Warning: package 'forcats' was built under R version 4.4.3
```

```
## Warning: package 'lubridate' was built under R version 4.4.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   4.0.0     ✓ tibble    3.2.1
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.0.4
## ── 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 becom
e errors
```

```
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.4.3
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(corrr)
```

```
## Warning: package 'corrr' was built under R version 4.4.3
```

```
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.4.3
```

```
library(FactoMineR)
```

```
## Warning: package 'FactoMineR' was built under R version 4.4.3
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.4.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.3
```

```
## corrplot 0.95 loaded
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.4.3
```

```
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 4.4.3
```

```
##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(aplpack)
```

```
## Warning: package 'aplpack' was built under R version 4.4.3
```

```
library(scatterplot3d)
library(asbio)
```

```
## Warning: package 'asbio' was built under R version 4.4.3
```

```
## Loading required package: tcltk
##
## Attaching package: 'asbio'
##
## The following object is masked from 'package:psych':
##
##     skew
##
## The following object is masked from 'package:lubridate':
##
##     pm
```

```
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 4.4.3
```

```
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.4.3
```

Load Dataset

```
df <- read.csv("Mental_Health_and_Social_Media_Balance_Dataset.csv", stringsAsFactors = TRUE)
```

Dataset consists of 500 observations and 10 variables

```
# View first 6 rows of data
head(df)
```

```
##   User_ID Age Gender Daily_Screen_Time.hrs. Sleep_Quality.1.10.
## 1    U001  44   Male                   3.1                   7
## 2    U002  30  Other                   5.1                   7
## 3    U003  23  Other                   7.4                   6
## 4    U004  36 Female                   5.7                   7
## 5    U005  34 Female                   7.0                   4
## 6    U006  38   Male                   6.6                   5
##   Stress_Level.1.10. Days_Without_Social_Media Exercise_Frequency.week.
## 1                  6                         2                        5
## 2                  8                         5                        3
## 3                  7                         1                        3
## 4                  8                         1                        1
## 5                  7                         5                        1
## 6                  7                         4                        3
##   Social_Media_Platform Happiness_Index.1.10.
## 1              Facebook                    10
## 2              LinkedIn                    10
## 3               YouTube                     6
## 4                TikTok                     8
## 5          X (Twitter)                     8
## 6              LinkedIn                     8
```

```
# Rename columns for simplicity
df <- df %>%
  rename(
    Screen_Time      = Daily_Screen_Time.hrs.,
    Sleep_Quality    = Sleep_Quality.1.10.,
    Stress_Level     = Stress_Level.1.10.,
    Days_No_Social   = Days_Without_Social_Media,
    Exercise_Freq    = Exercise_Frequency.week.,
    Social_Media     = Social_Media_Platform,
    Happiness_Level  = Happiness_Index.1.10.
  )

# Check contents of data
str(df)
```

```
## 'data.frame':    500 obs. of  10 variables:
##  $ User_ID        : Factor w/ 500 levels "U001","U002",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Age            : int  44 30 23 36 34 38 26 26 39 39 ...
##  $ Gender         : Factor w/ 3 levels "Female","Male",..: 2 3 3 1 1 2 1 1 2 1 ...
##  $ Screen_Time    : num  3.1 5.1 7.4 5.7 7 6.6 7.8 7.4 4.7 6.6 ...
##  $ Sleep_Quality  : num  7 7 6 7 4 5 4 5 7 6 ...
##  $ Stress_Level   : num  6 8 7 8 7 7 8 6 7 8 ...
##  $ Days_No_Social : num  2 5 1 1 5 4 2 1 6 0 ...
##  $ Exercise_Freq  : num  5 3 3 1 1 3 0 4 1 2 ...
##  $ Social_Media   : Factor w/ 6 levels "Facebook","Instagram",..: 1 3 6 4 5 3 4 2 6 1 ...
##  $ Happiness_Level: num  10 10 6 8 8 8 7 7 9 7 ...
```

```
# View Summary Statistics
summary(df)
```

```
##     User_ID         Age           Gender     Screen_Time    Sleep_Quality
##  U001   :  1   Min.   :16.00   Female:229   Min.   : 1.00   Min.   : 2.000
##  U002   :  1   1st Qu.:24.00   Male  :248   1st Qu.: 4.30   1st Qu.: 5.000
##  U003   :  1   Median :34.00   Other : 23   Median : 5.60   Median : 6.000
##  U004   :  1   Mean   :32.99                Mean   : 5.53   Mean   : 6.304
##  U005   :  1   3rd Qu.:41.00                3rd Qu.: 6.70   3rd Qu.: 7.000
##  U006   :  1   Max.   :49.00                Max.   :10.80   Max.   :10.000
##  (Other):494
##   Stress_Level    Days_No_Social  Exercise_Freq        Social_Media
##  Min.   : 2.000   Min.   :0.000   Min.   :0.000   Facebook   :81
##  1st Qu.: 6.000   1st Qu.:2.000   1st Qu.:1.000   Instagram  :74
##  Median : 7.000   Median :3.000   Median :2.000   LinkedIn   :87
##  Mean   : 6.618   Mean   :3.134   Mean   :2.448   TikTok     :95
##  3rd Qu.: 8.000   3rd Qu.:5.000   3rd Qu.:3.000   X (Twitter):88
##  Max.   :10.000   Max.   :9.000   Max.   :7.000   YouTube    :75
##
##  Happiness_Level
##  Min.   : 4.000
##  1st Qu.: 7.000
##  Median : 9.000
##  Mean   : 8.376
##  3rd Qu.:10.000
##  Max.   :10.000
##
```

```
# Check to see if there are missing values in dataset
colSums(is.na(df))
```

```
##         User_ID             Age          Gender     Screen_Time   Sleep_Quality
##               0               0               0               0               0
##    Stress_Level  Days_No_Social   Exercise_Freq    Social_Media Happiness_Level
##               0               0               0               0               0
```

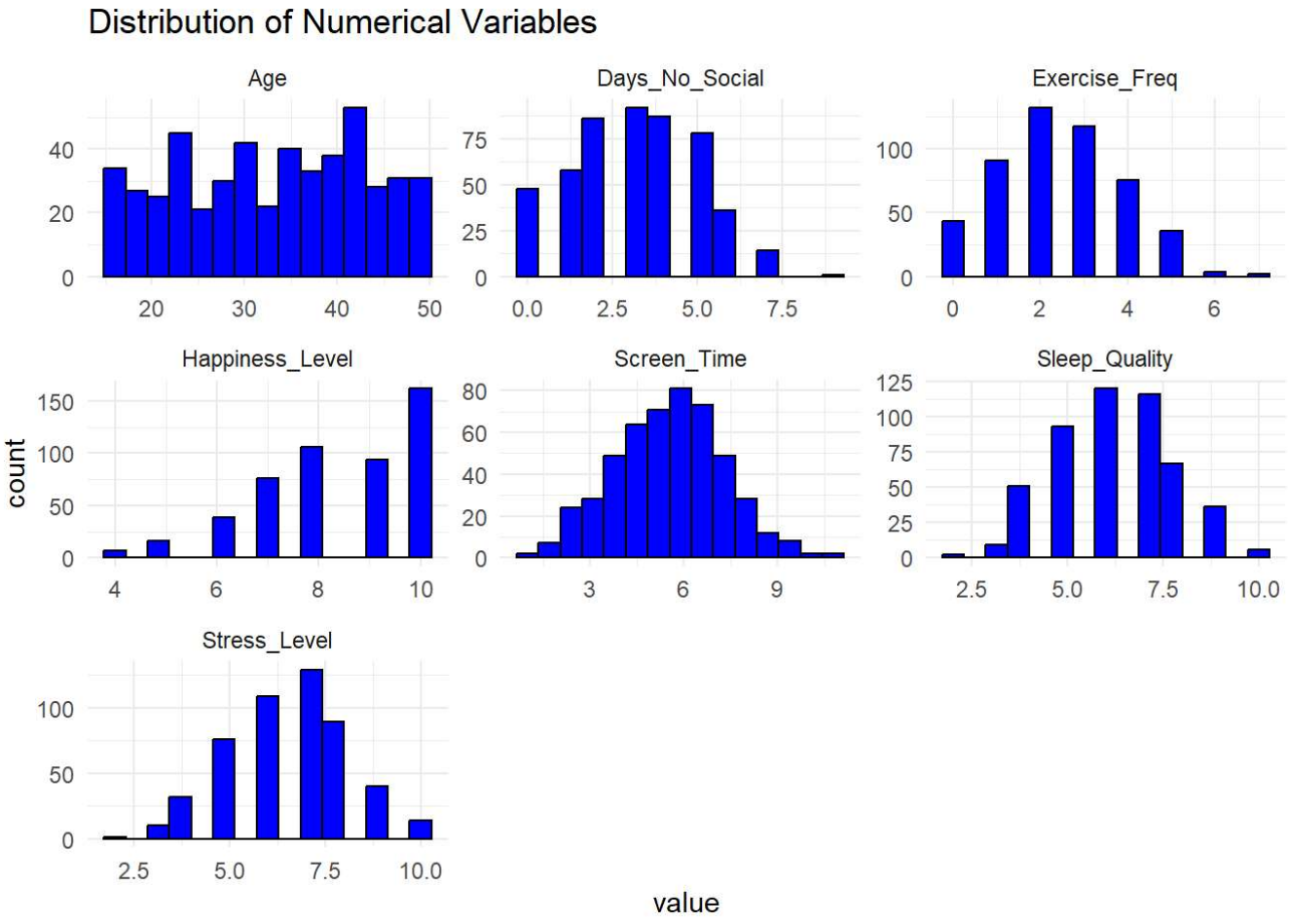
```
# There are no missing values in data

# Check for duplicats in data
sum(duplicated(df))
```

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

```
# No duplicates found in dataset
```

Let's visual explore the numerical variables and the categorical variables

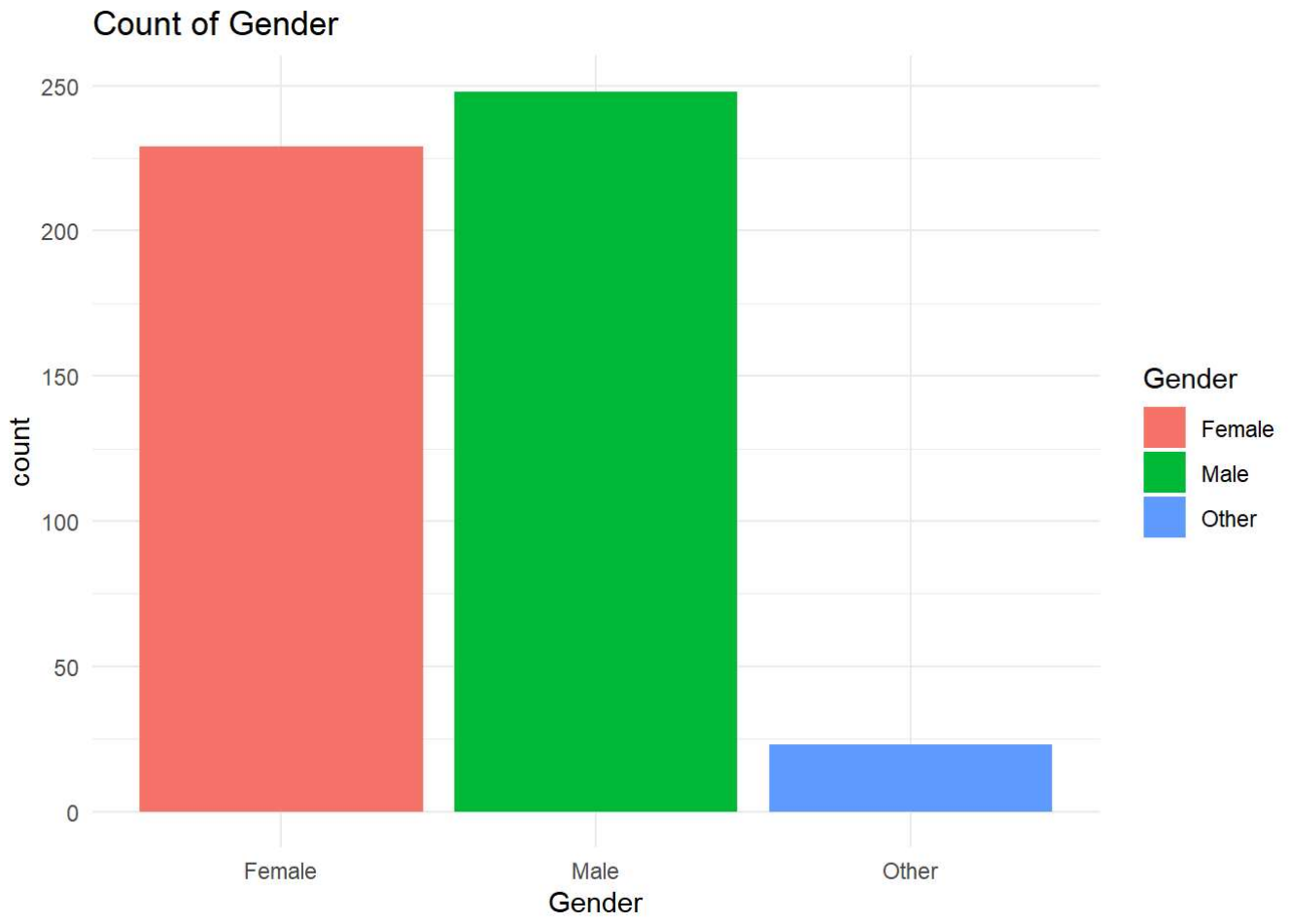
```r
# Numerical variables
num_vars <- df %>% select(where(is.numeric))

# View histograms
if(ncol(num_vars) > 0) {
  num_vars %>%
    gather(key="variable", value="value") %>%
    ggplot(aes(x=value)) +
    geom_histogram(fill="blue", color="black", bins=15) +
    theme_minimal() +
    facet_wrap(~variable, scales="free") +
    labs(title="Distribution of Numerical Variables")
}
```



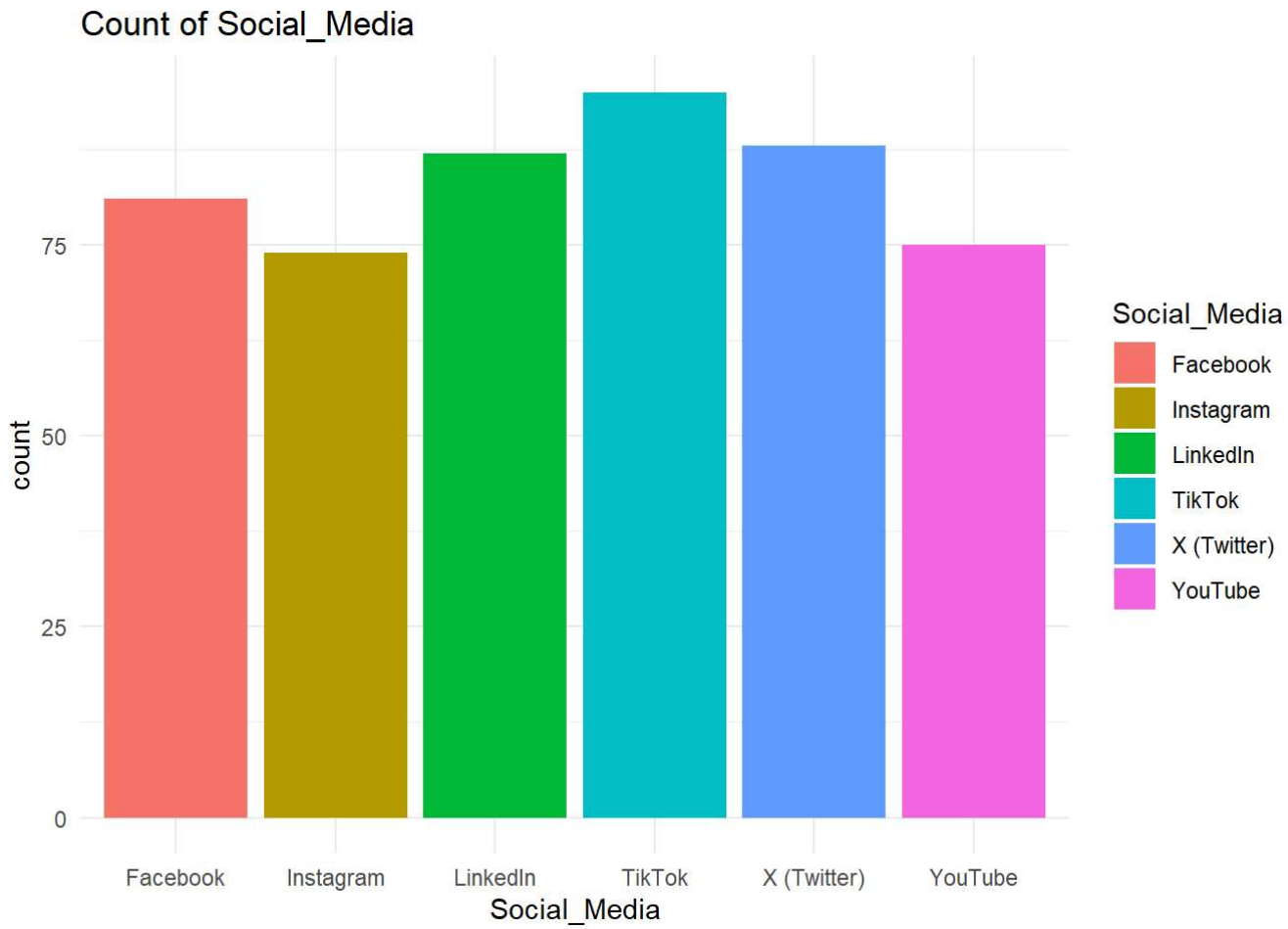
```r
# Categorical variables
cat_vars <- df %>%
  select(where(is.factor)) %>%
  select(-User_ID)

# View count and bar plots
if(ncol(cat_vars) > 0) {
  for(v in names(cat_vars)){
    print(df %>% group_by(.data[[v]]) %>% summarise(count= n()))
    p <- ggplot(df, aes_string(x=v, fill=v)) +
      geom_bar() +
      theme_minimal() +
      labs(title=paste("Count of", v))

    print(p)
  }
}
```

```
## # A tibble: 3 × 2
##   Gender count
##   <fct>  <int>
## 1 Female   229
## 2 Male     248
## 3 Other     23
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## ℹ Please use tidy evaluation idioms with `aes()`.
## ℹ See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

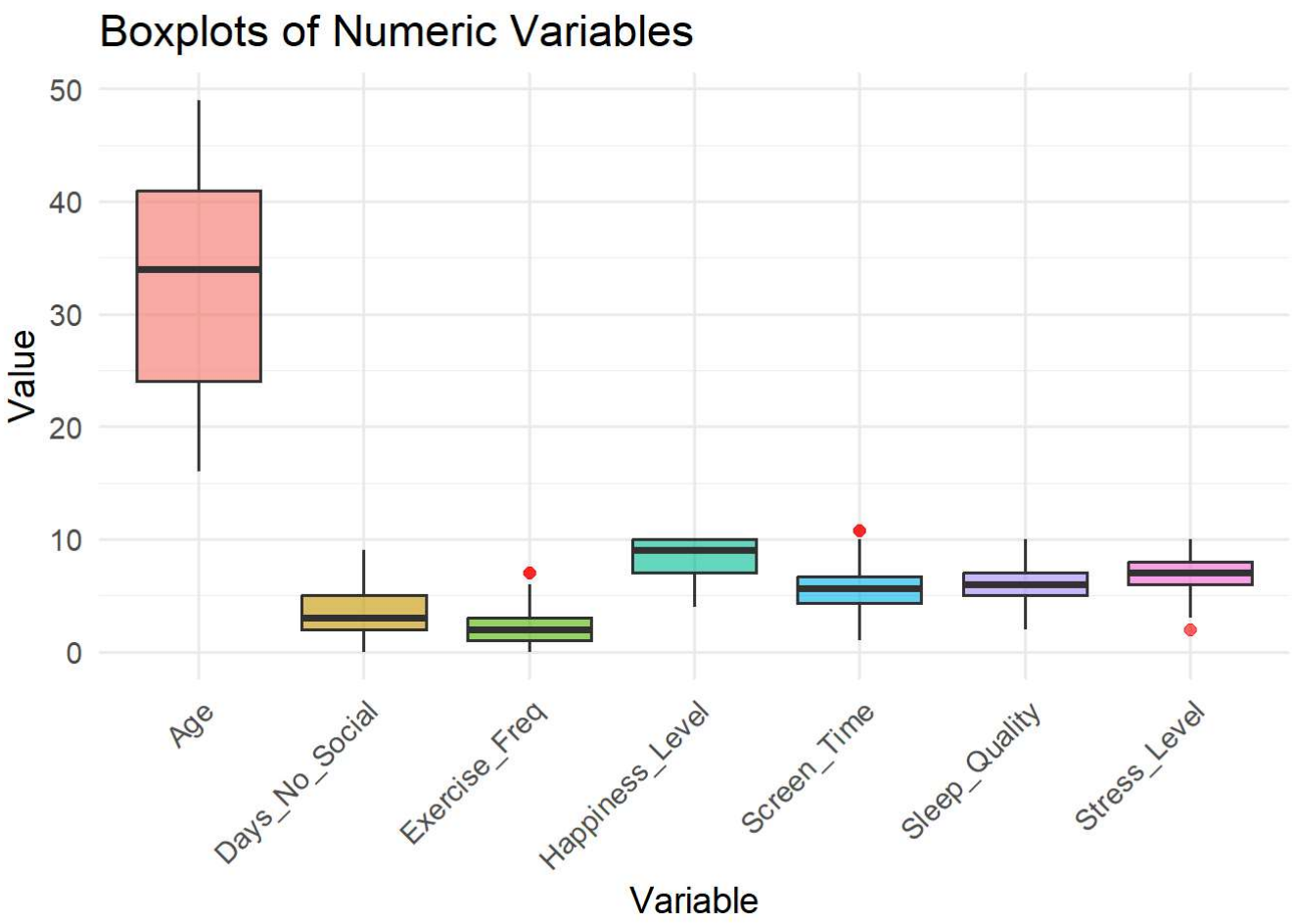



```
## # A tibble: 6 × 2
##   Social_Media count
##   <fct>        <int>
## 1 Facebook        81
## 2 Instagram       74
## 3 LinkedIn        87
## 4 TikTok          95
## 5 X (Twitter)     88
## 6 YouTube         75
```



Let's Look at Outlier by using boxplots and Mahalanobis Distance

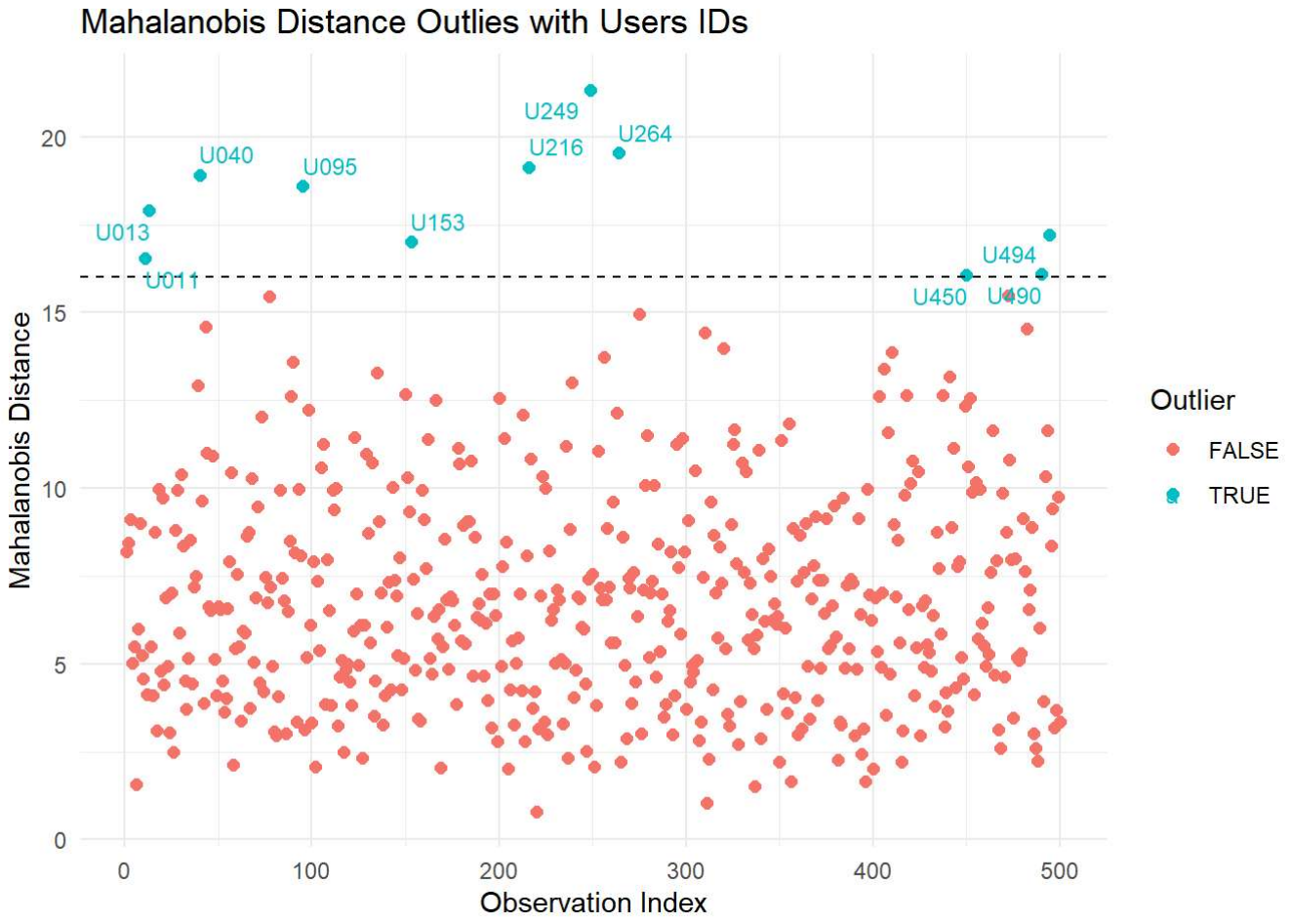
```
# Boxplots are a way to visually see min,1st quartile, median, 3rd quartile, max and any potenti
al outliers
if(ncol(num_vars) > 0) {
  num_vars %>%
    pivot_longer(cols = everything()) %>%
    ggplot(aes(x = name, y = value, fill = name)) +
    geom_boxplot(outlier.colour = "red", alpha = 0.6) +
    labs(title = "Boxplots of Numeric Variables", x = "Variable", y = "Value") +
    theme_minimal(base_size = 14) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    guides(fill = "none")
}
```




```
## Daily Screen Time, Exercise Frequency and Stress Level seems to have outliers. This shows out
liers of for each individual variable of a user.

## We can use Mahalanobis Distance to determine which users based off of all variables are consi
dered outliers

## Let's create new dataframe for Mahlanobis Distance
df_2 <- num_vars

# Standardize data
df_scaled <- scale(df_2)

# Compute Mahlanobis Distance
center <- colMeans(df_scaled)
cov_matrix <- cov(df_scaled)
mahal <- mahalanobis(df_scaled, center, cov_matrix)

# Determine outliers
threshold <- qchisq(0.975, df = ncol(df_scaled))
outlier_flag <- mahal > threshold
table(outlier_flag)
```

```
## outlier_flag
## FALSE  TRUE
##   489    11
```

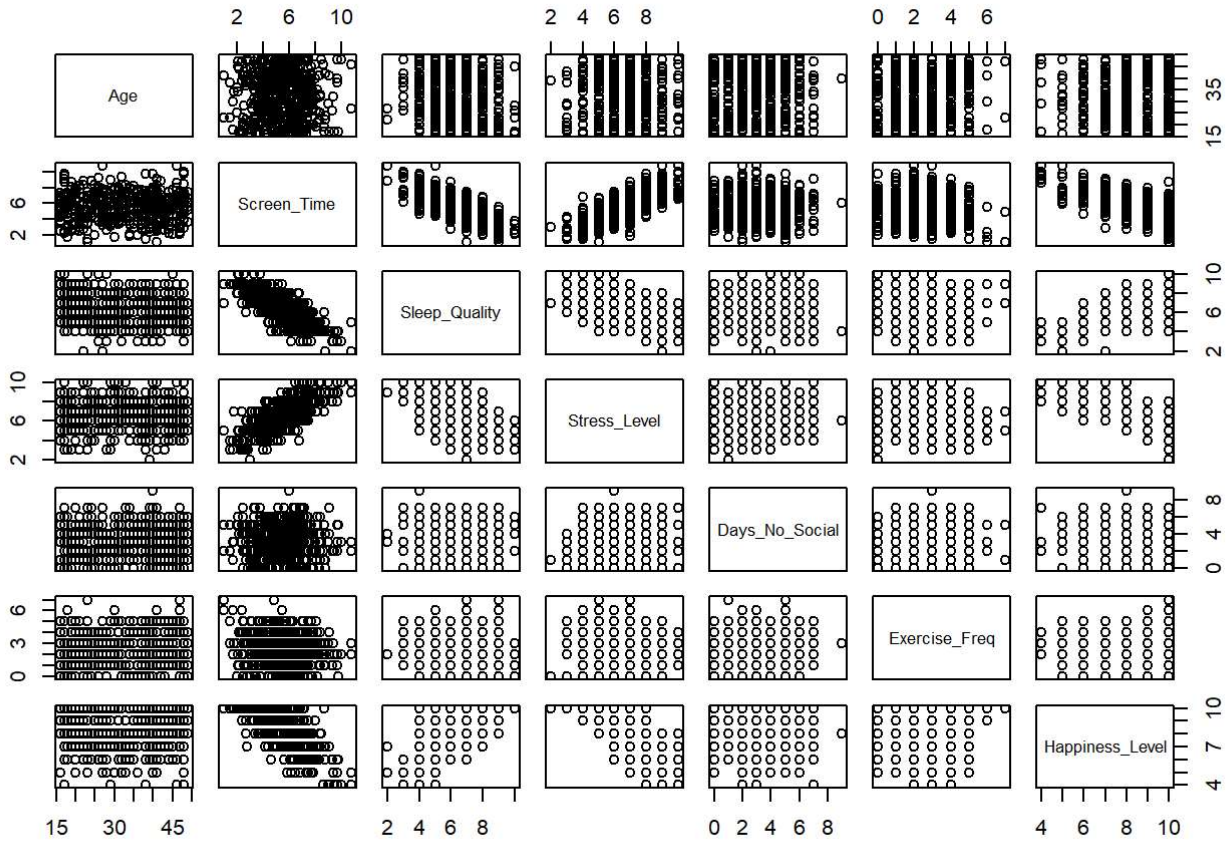
```
# Create dataframe for plotting
df_plot <- data.frame(
  ID = df$User_ID,
  Index = 1:nrow(df_scaled),
  Mahalanobis_Dis = mahal,
  Outlier = outlier_flag
)

# Visualize
outliers <- df_plot[df_plot$Outlier,]

ggplot(df_plot, aes(x = Index, y = Mahalanobis_Dis, color = Outlier)) +
  geom_point(size = 2) +
  geom_hline(yintercept = threshold, linetype="dashed") +
  geom_text_repel(data = outliers, aes(label = ID), size = 3) +
  labs(title="Mahalanobis Distance Outlies with Users IDs", x = "Observation Index", y="Mahalano
bis Distance") +
  theme_minimal()
```



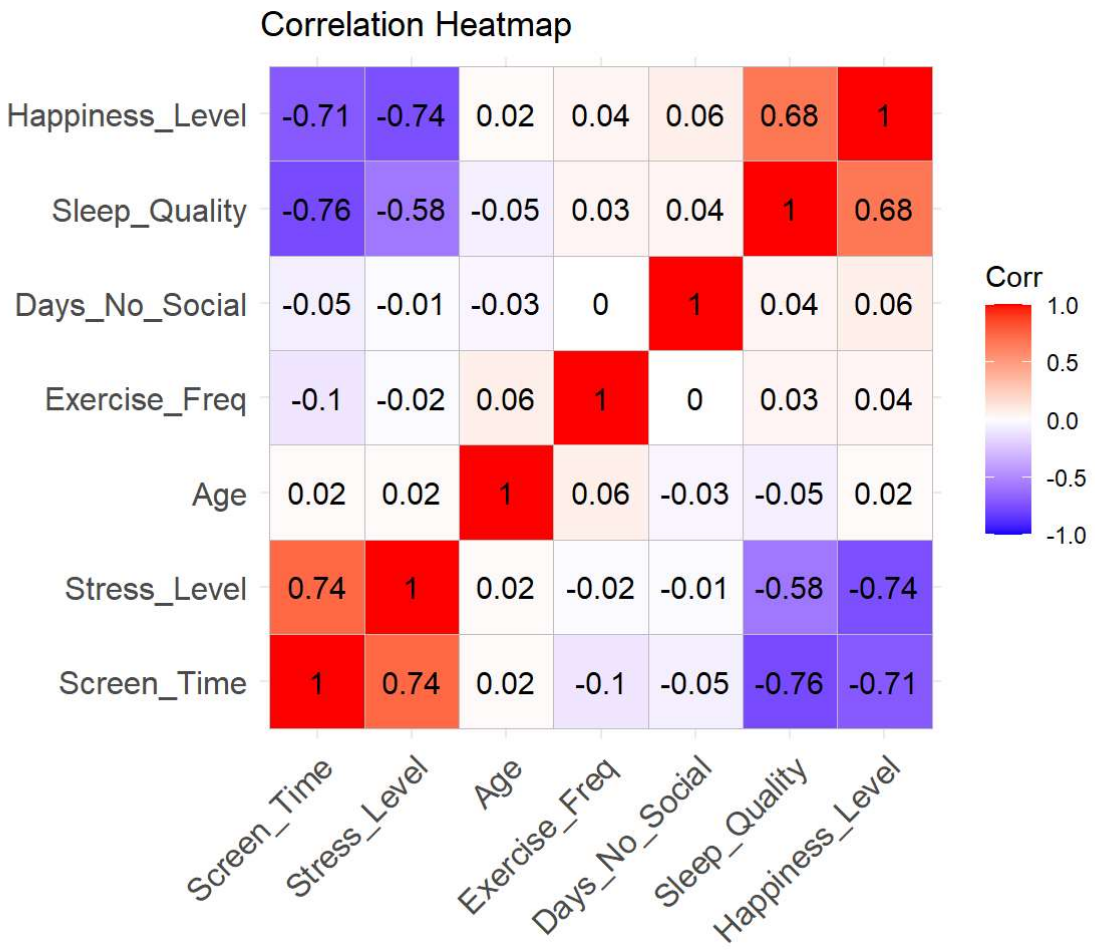
```
# 11 Users are identified as Outliers at the 97.5% threshold: U040,U013, U011, U095, U249, U153, U216, U264, U450, U494, U490
```

Let's now find relationships/associations between numeric variables

```
### Check correlation between numeric variables
# Get correlation matrix
cor_matrix <- cor(num_vars)

# Look at ScatterPlot Matrix
pairs(num_vars)
```




```
# Correlation heatmap
ggcorrplot(cor_matrix, lab = TRUE, hc.order = TRUE, title = "Correlation Heatmap")
```




```
# Happiness Index and Daily Screen Time = -0.71 (strong negative correlation)
# Ex: As Happiness increases daily screen time decreases

# Sleep Quality and Daily Screen Time = -0.76 (Strong negative correlation)
# As sleep quality increases, daily screen time decreases

# Stress Level and Daily Screen Time = 0.74 (Strong positive correlation)
# As stress level increases, daily screen time increases

# Happiness and Sleep Quality = 0.68 (strong positive correlation)
# As happiness increases, sleep quality increases also

# Happiness and Stress Level = -0.74 (strong negative correlation)
# as happiness increases, stress levels decreases

# Sleep Quality and Stress Level = - 0.58 (Moderately strong positive correlation)
# As sleep quality increases, stress levels decreases

# There is correlation among the variables but majority are not correlated
# Sleep, Stress and Screen Time have high amount of impact
# Age does not correlate much with any of the variables

# Dataset has multicollinearity since more than two variables have correlations |r| > 7.
```

Since dataset has multicollinearity let's apply Principal Component Analysis.

Principal Component Analysis is used to reduce the data to fewer dimensions that represent majority of the variation in the data.

```
### PRINCIPAL COMPONENT ANALYSIS
df_pc <- num_vars

# Since we have variables that are highly correlated let's use prcomp for PCA
df_pc_result <- prcomp(df_pc, scale. = TRUE)

# View PCA
summary(df_pc_result)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     1.7640 1.0354 0.9989 0.9662 0.65201 0.53460 0.41693
## Proportion of Variance 0.4446 0.1532 0.1425 0.1334 0.06073 0.04083 0.02483
## Cumulative Proportion  0.4446 0.5977 0.7402 0.8736 0.93434 0.97517 1.00000
```

```
# PC1 explains 44.46% of variance, PC2 15.32%, PC3 14.25%, PC4 13.34% => 87.36% of total varianc
e in dataset
# 4 Principal Components seem to work best here.

# Let's look a the eigenvalues and a scree plot to determine the necessary principal components

# access loadings
loadings <- df_pc_result$rotation
print(loadings)
```

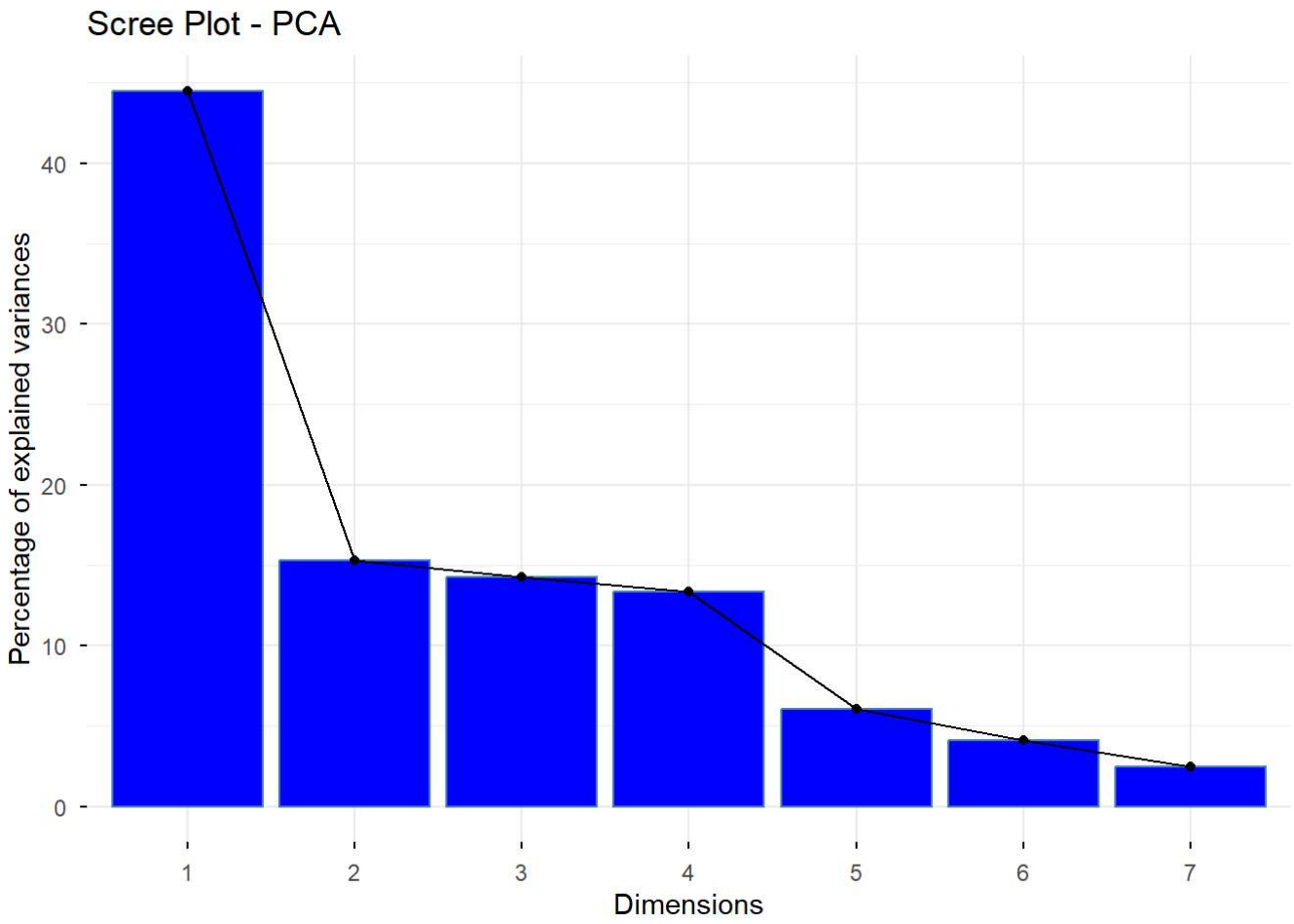
```
##                         PC1          PC2           PC3         PC4         PC5
## Age             -0.01667955  0.709390723 -0.0212906714 -0.69942313 -0.06612579
## Screen_Time     -0.51720155 -0.026988932 -0.0003214193 -0.03862377  0.15799117
## Sleep_Quality    0.48506831 -0.052856268  0.0227142737  0.01407172 -0.72118701
## Stress_Level    -0.49145902  0.008159706 -0.0789549139  0.04329300 -0.62535240
## Days_No_Social   0.03758522 -0.253374143 -0.9376811261 -0.22827028  0.02519155
## Exercise_Freq    0.04416963  0.654763807 -0.3369728607  0.67032832  0.03153247
## Happiness_Level  0.50203337  0.016184401 -0.0004292889 -0.07613022  0.24054299
##                         PC6         PC7
## Age             -0.04243344 -0.02583344
## Screen_Time      0.53620594 -0.64637778
## Sleep_Quality    0.09899905 -0.48092565
## Stress_Level     0.32260939  0.50512458
## Days_No_Social  -0.03808861 -0.03082731
## Exercise_Freq    0.03825926 -0.06312343
## Happiness_Level  0.77064288  0.30025520
```

```
# PC1 describes 44.46% of total variance. It is dominated by Sleep Quality (0.49) and Happiness (0.50) with negative loadings on Screen Time (-0.52) and Stress Level (-0.49). This suggest users tend to have good sleep quality and happiness along with have less amount of screen time and stress levels.

# PC2 describes 15.32% of total variance. It is dominated heavily by Age (0.71) and Exercise Frequency (0.65) suggesting that users who are older also exercise more frequently.

# PC3 describes 14.25% of total variance. It is dominated by Days without Social Media (-0.94) suggesting users which higher PC3 scores spend more time on social media and users with lower scores go more days without social media.

# PC4 describes 13.34% of total variance. It is dominated by exercise frequency (0.67) and Age (-0.70). This suggests that younger users tend to exercise more versus older users who exercise less.


# Extract standard deviation
sdev <- df_pc_result$sdev
# calculate eigenvalues
eigenvalues <- sdev^2
# print eigenvalues
print(eigenvalues)
```

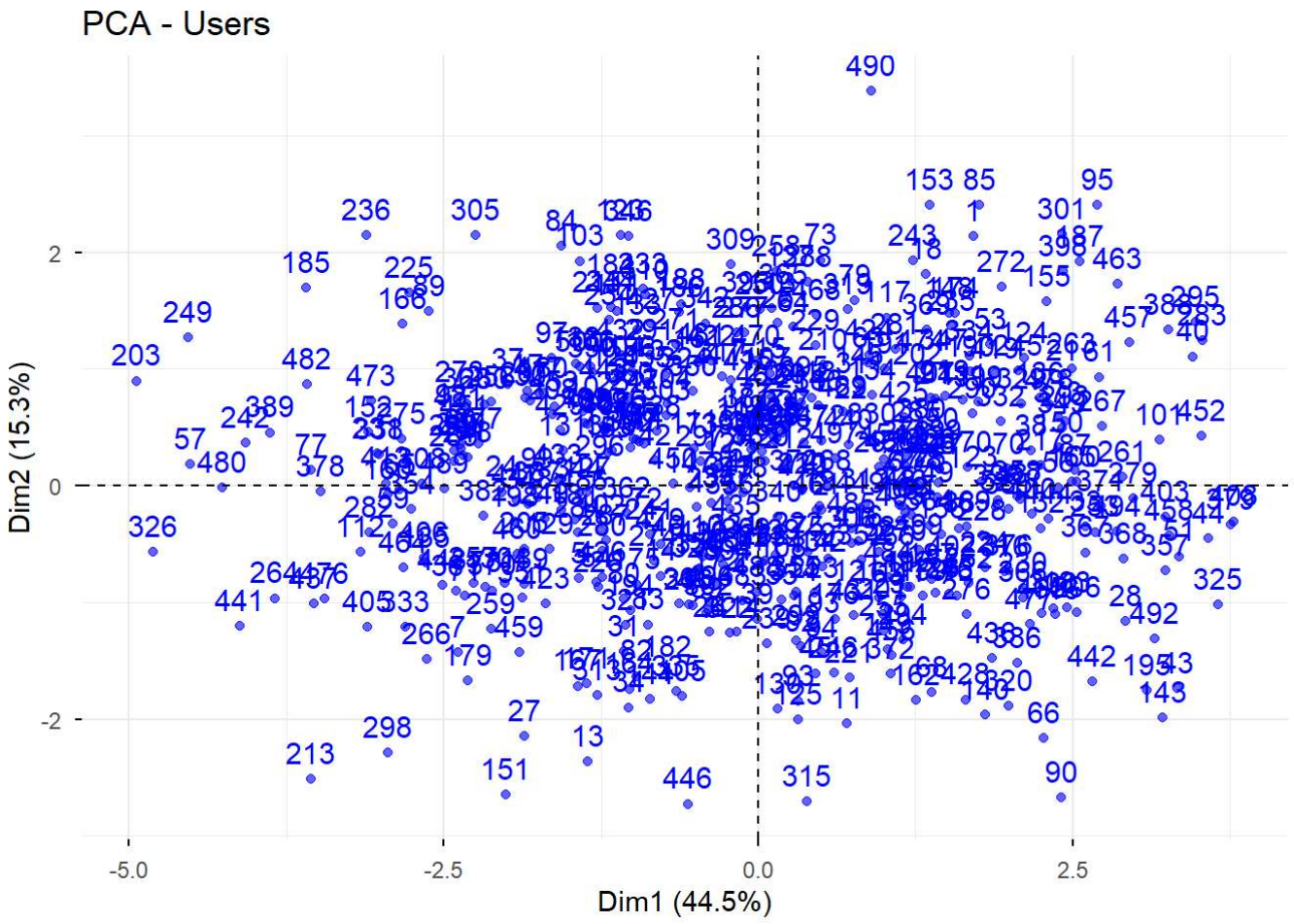
```
## [1] 3.1118553 1.0720623 0.9977688 0.9335646 0.4251126 0.2858016 0.1738347
```

```
# Based on Kisen Criterion: eigenvalue > 1
# PC1 = 3.11, PC2 = 1.07, PC3 = 0.99, PC4 = 0.93

# Lets' look at scree plot
fviz_eig(df_pc_result, addlabes = TRUE, barfill = "blue", main = "Scree Plot - PCA")
```

```
## Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
## Ignoring empty aesthetic: `width`.
```

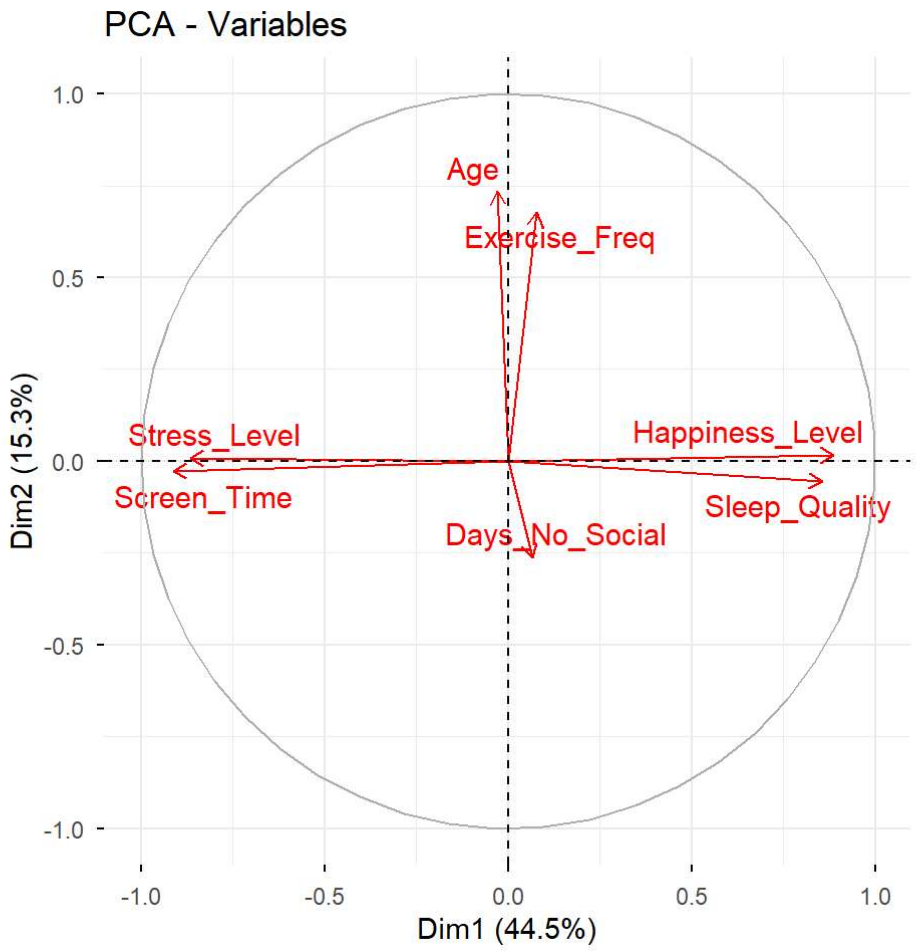



```
# Scree plot looks as if 4 Principal Components work

# Let's look at seperate pca plots for users and variables

# users
fviz_pca_ind(df_pc_result,
             col.ind = "blue",
             alpha.ind = 0.6,
             title = "PCA - Users")
```

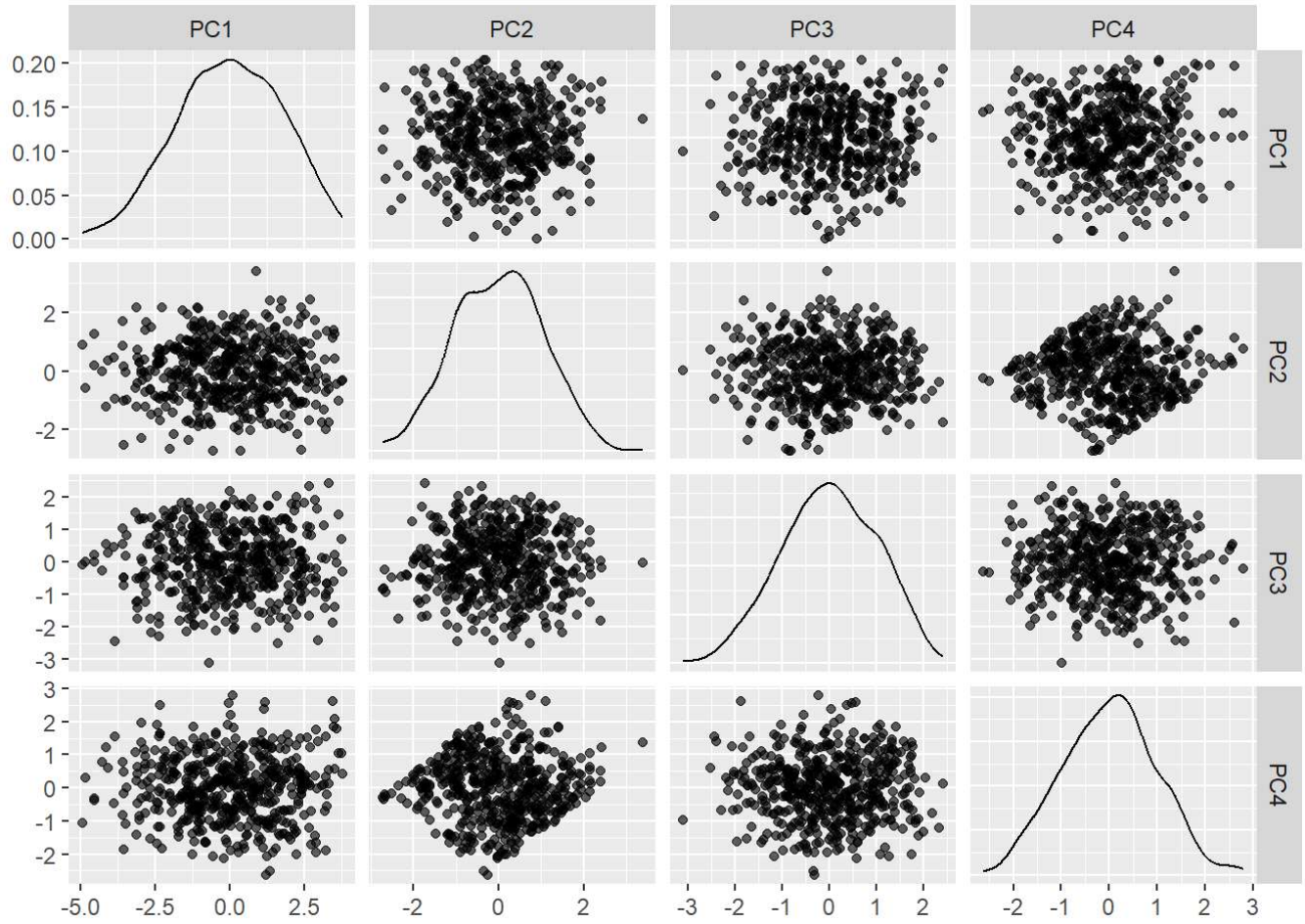



```
# Plot for variables
fviz_pca_var(df_pc_result,
             repel = TRUE,
             col.var = "red",
             title = "PCA - Variables"
)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## ℹ The deprecated feature was likely used in the ggpubr package.
##   Please report the issue at <https://github.com/kassambara/ggpubr/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
# Create DataFrame of PC's for plots
pc_df <- as.data.frame(df_pc_result$x[, 1:4]) %>%
  mutate(ID = df$User_ID)

# Let's look at Scatterplot for each PC
# Only include the first 4 PCs
ggpairs(pc_df[, 1:4],
        upper = list(continuous = wrap("points", alpha = 0.6, size = 1.5)),
        lower = list(continuous = wrap("points", alpha = 0.6, size = 1.5)),
        diag = list(continuous = "densityDiag"))
```



Lastly let's check to see if there are any unobservable factors that are influencing our variables

Perform Factor Analysis

```
# Now let's see if there are any unobservables factors that influence variables
df_fa <- num_vars

# Determine the number of factors
fa1 <- factanal(df_fa, factors = 1, rotation = "varimax")
print(fa1)
```

```
##
## Call:
## factanal(x = df_fa, factors = 1, rotation = "varimax")
##
## Uniquenesses:
##             Age     Screen_Time   Sleep_Quality    Stress_Level  Days_No_Social
##           1.000           0.196           0.349           0.329           0.998
##   Exercise_Freq Happiness_Level
##           0.996           0.314
##
## Loadings:
##                 Factor1
## Age
## Screen_Time     -0.897
## Sleep_Quality    0.807
## Stress_Level    -0.819
## Days_No_Social
## Exercise_Freq
## Happiness_Level  0.828
##
##                Factor1
## SS loadings      2.819
## Proportion Var   0.403
##
## Test of the hypothesis that 1 factor is sufficient.
## The chi square statistic is 89.39 on 14 degrees of freedom.
## The p-value is 4.95e-13
```

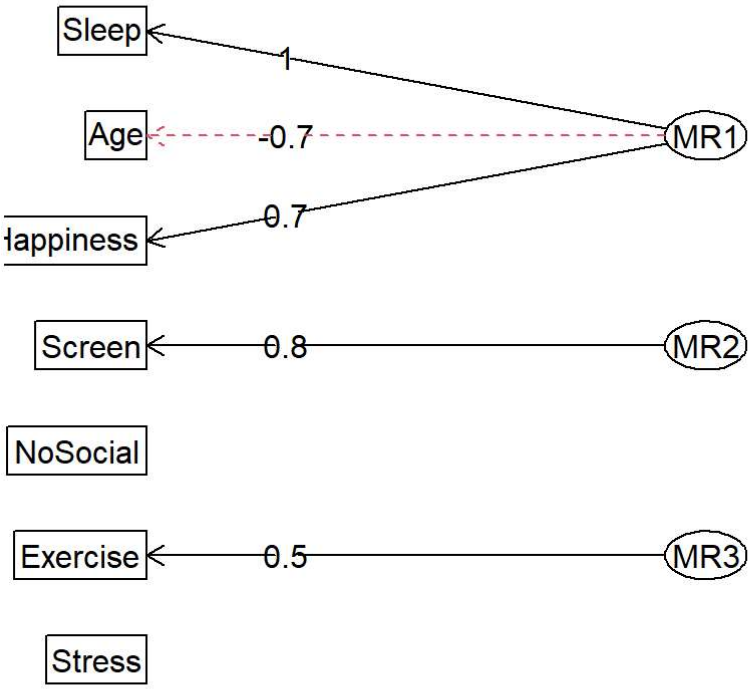
```
# p-value is extremely small try 2

fa2 <- factanal(df_fa, factors = 2, rotation = "varimax")
print(fa2)
```

```
##
## Call:
## factanal(x = df_fa, factors = 2, rotation = "varimax")
##
## Uniquenesses:
##             Age     Screen_Time   Sleep_Quality    Stress_Level  Days_No_Social
##           0.998           0.236           0.165           0.005           0.995
##   Exercise_Freq Happiness_Level
##           0.996           0.333
##
## Loadings:
##                 Factor1 Factor2
## Age
## Screen_Time      0.733  -0.476
## Sleep_Quality   -0.573   0.711
## Stress_Level     0.997
## Days_No_Social
## Exercise_Freq
## Happiness_Level -0.733   0.362
##
##                Factor1 Factor2
## SS loadings      2.398   0.874
## Proportion Var   0.343   0.125
## Cumulative Var   0.343   0.467
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 17.73 on 8 degrees of freedom.
## The p-value is 0.0233
```

```
# p-value is 0.0233, still small so let's try 3

fa3 <- factanal(df_fa, factors = 3, rotation = "varimax")
print(fa3)
```

```
##
## Call:
## factanal(x = df_fa, factors = 3, rotation = "varimax")
##
## Uniquenesses:
##             Age     Screen_Time   Sleep_Quality    Stress_Level  Days_No_Social
##           0.974           0.229           0.098           0.005           0.995
##   Exercise_Freq Happiness_Level
##           0.844           0.339
##
## Loadings:
##                 Factor1 Factor2 Factor3
## Age                              0.161
## Screen_Time      0.771  -0.418
## Sleep_Quality   -0.631   0.674  -0.222
## Stress_Level     0.995
## Days_No_Social
## Exercise_Freq            0.134   0.370
## Happiness_Level -0.760   0.290
##
##                Factor1 Factor2 Factor3
## SS loadings      2.562   0.740   0.215
## Proportion Var   0.366   0.106   0.031
## Cumulative Var   0.366   0.472   0.503
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 5.68 on 3 degrees of freedom.
## The p-value is 0.128
```

```
# p-value is 0.128, seems significant enough, let's use 3 factors

# Choose 3 factor model
names(df_fa) <- c("Stress", "Happiness", "Screen", "Sleep", "NoSocial", "Exercise", "Age")
fa_results <- psych::fa(r = df_fa, nfactors = 3, rotate = "varimax")
fa.diagram(fa_results)
```




```
# Add factors scores to dataframe
fa.scores <- fa_results$scores
df_full_fa <- bind_cols(df_fa, as.data.frame(fa.scores))
```