```r
library(readxl)
library(lme4)

## Loading required package: Matrix

library(lmerTest)

##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer

## The following object is masked from 'package:stats':
##
##     step

library(car)

## Loading required package: carData

library(MuMIn)
library(afex)

## ************
## Welcome to afex. For support visit: http://afex.singmann.science/

## - Functions for ANOVAs: aov_car(), aov_ez(), and aov_4()
## - Methods for calculating p-values with mixed(): 'S', 'KR', 'LRT', and 'PB'
## - 'afex_aov' and 'mixed' objects can be passed to emmeans() for follow-up tests
## - NEWS: emmeans() for ANOVA models now uses model = 'multivariate' as default.
## - Get and set global package options with: afex_options()
## - Set orthogonal sum-to-zero contrasts globally: set_sum_contrasts()
## - For example analyses see: browseVignettes("afex")
## ************

##
## Attaching package: 'afex'

## The following object is masked from 'package:lme4':
##
##     lmer

data <- read_excel("../Data/PredictingOutcomes_ParticipantDemographics.xlsx", sheet = "Study 1A")

print(data)

## # A tibble: 144 x 19
##    study partici~1 gener~2 rate  respo~3 score~4 respo~5 score~6 respo~7 score~8
##    <chr>     <dbl> <chr>   <chr> <chr>     <dbl> <chr>     <dbl> <chr>     <dbl>
##  1 1A            1 analyst ukno~ 50%           0 5             1 0.1%          1
##  2 1A            3 analyst ukno~ 12.5%         1 50            0 0.1%          1
##  3 1A            6 analyst ukno~ 25            0 1             0 1             0
##  4 1A            8 analyst ukno~ 12.5%         1 5             1 0.1%          1
##  5 1A           14 analyst ukno~ 15            0 50            0 0.1           1
##  6 1A           17 analyst ukno~ 50            0 100           0 10            0
##  7 1A           20 analyst ukno~ 15            0 5             1 0.1           1
##  8 1A           24 analyst ukno~ 10            0 5             1 0.1           1
```

```
##  9 1A           28 analyst ukno~ 35            0 5              1 .1             1
## 10 1A           32 analyst ukno~ 0.15%         0 1              0 0.1%           1
## # ... with 134 more rows, 9 more variables: response_fin1 <dbl>,
## #   score_fin1 <dbl>, response_fin2 <dbl>, score_fin2 <dbl>, age <dbl>,
## #   gender <dbl>, highest_degree <dbl>, stocks <dbl>, gambling <dbl>, and
## #   abbreviated variable names 1: participant_id, 2: generator,
## #   3: response_prob1, 4: score_prob1, 5: response_prob2, 6: score_prob2,
## #   7: response_prob3, 8: score_prob3
```

create a map like data structure to store the unique participant id with there corresponding gender

```r
data1 <- read_excel("../Data/PredictingOutcomes_ParticipantPredictions.xlsx", sheet = "Study 1A")
data_an <- data1[data1$generator == "analyst",]
data_stock <- data1[data1$generator == "stock",]
data_bingo <- data1[data1$generator == "bingo",]
```

make a dataframe with values aggreagated on the basis of terminal streak length and participant_id

```r
df1 <- aggregate(data1$prediction_recode, by = list(data1$terminal_streak_length, data1$participant_id)
df_an <- aggregate(data_an$prediction_recode, by = list(data_an$terminal_streak_length,data_an$particip
df_stock <- aggregate(data_stock$prediction_recode, by = list(data_stock$terminal_streak_length,data_st
df_bingo <- aggregate(data_bingo$prediction_recode, by = list(data_bingo$terminal_streak_length,data_bi
```

change the column name of the dataframe

```r
colnames(df1) <- c("terminal_streak_length", "participant_id", "prediction_recode")
colnames(df_an) <- c("terminal_streak_length", "participant_id", "prediction_recode")
colnames(df_stock) <- c("terminal_streak_length", "participant_id", "prediction_recode")
colnames(df_bingo) <- c("terminal_streak_length", "participant_id", "prediction_recode")
shapiro.test(df1$prediction_recode)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df1$prediction_recode
## W = 0.95089, p-value < 2.2e-16
```

apply spearman correlation on the dataframe

```r
cor.test(df1$terminal_streak_length, df1$prediction_recode, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  df1$terminal_streak_length and df1$prediction_recode
## t = 16.91, df = 1006, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4209402 0.5171815
## sample estimates:
##       cor
## 0.4704587
```

```r
cor.test(df_an$terminal_streak_length, df_an$prediction_recode, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_an$terminal_streak_length and df_an$prediction_recode
```

```
## t = 10.098, df = 348, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3906956 0.5532504
## sample estimates:
##       cor
## 0.4760289
```

```
cor.test(df_stock$terminal_streak_length, df_stock$prediction_recode, method = "pearson")
```

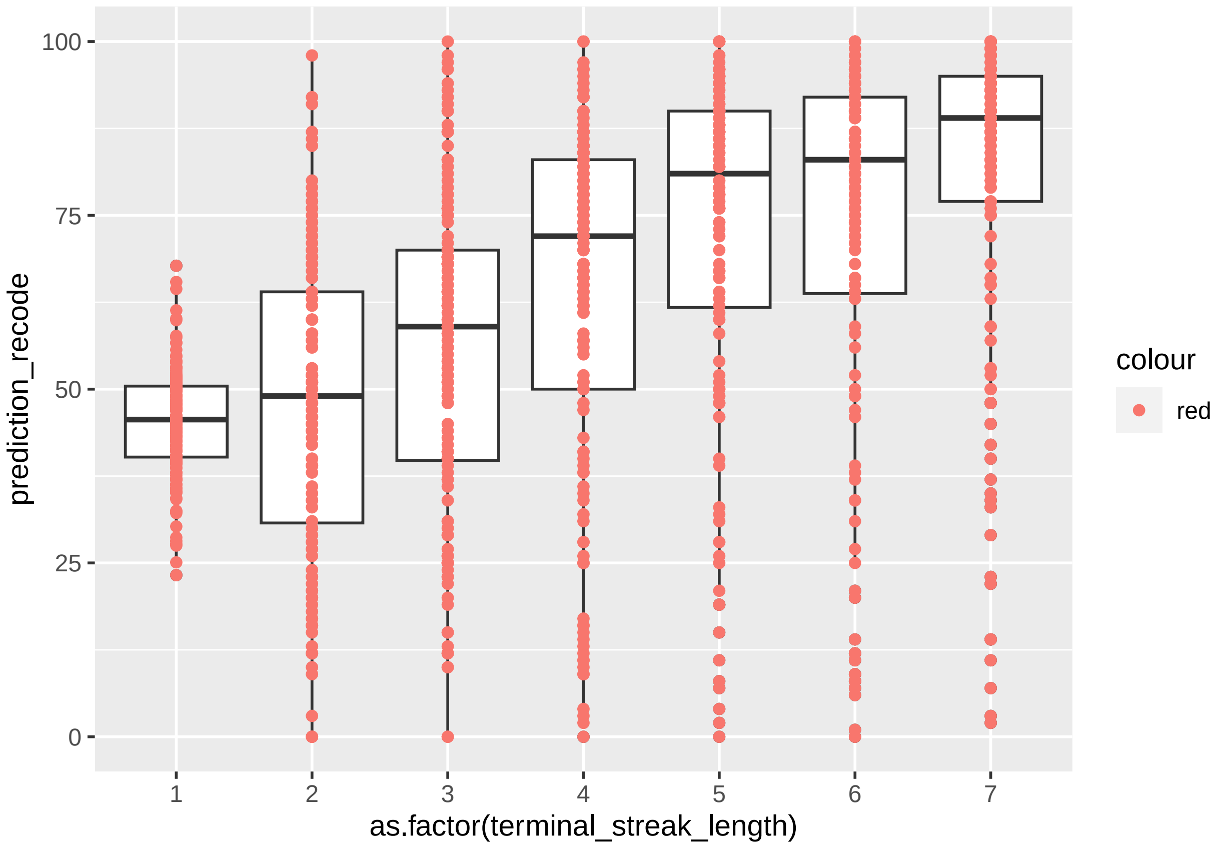
```
##
##  Pearson's product-moment correlation
##
## data:  df_stock$terminal_streak_length and df_stock$prediction_recode
## t = 12.048, df = 306, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4862811 0.6384972
## sample estimates:
##       cor
## 0.5672138
```

```
cor.test(df_bingo$terminal_streak_length, df_bingo$prediction_recode, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_bingo$terminal_streak_length and df_bingo$prediction_recode
## t = 8.1723, df = 348, p-value = 5.682e-15
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3094522 0.4856656
## sample estimates:
##      cor
## 0.401265
```

make boxplot of the data having dots with color blue and boxplot with color yellow

```
library(ggplot2)
ggplot(df1, aes(x = as.factor(terminal_streak_length), y = prediction_recode)) + geom_boxplot() +geom_p
```



correlation heatmap for participant_id vs prediction_recode