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

```
## Loading required package: Matrix
```

```r
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'
```

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

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

```r
library(car)
```

```
## Loading required package: carData
```

```r
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
```

```r
data <- read_excel("../Data/PredictingOutcomes_ParticipantDemographics.xlsx", sheet = "Study 3A")

# print(data)
```

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

```r
map <- data.frame(unique(data$participant_id), data$gender)
colnames(map) <- c("participant_id","gender")
# map
```

```r
data1 <- read_excel("../Data/PredictingOutcomes_ParticipantPredictions.xlsx", sheet = "Study 3A")
# only tke the data for columns participant_id, prediction_recode, prediction_recode
data1 <- data1[,c(2,3,10)]
# print(data1)
```

```r
df <- merge(data1, map, by = "participant_id")
male <- df[df$gender=='0',]
female <- df[df$gender=='1',]
t.test(male$prediction_recode, female$prediction_recode)
```

```
##
##  Welch Two Sample t-test
##
## data:  male$prediction_recode and female$prediction_recode
## t = -0.66428, df = 2692.9, p-value = 0.5066
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.721755  1.344285
## sample estimates:
## mean of x mean of y
##  51.87208  52.56081
```

```r
df_analyst <- df[df$generator=="analyst",]
male <- df_analyst[df_analyst$gender=='0',]
female <- df_analyst[df_analyst$gender=='1',]
t.test(male$prediction_recode, female$prediction_recode)
```

```
##
##  Welch Two Sample t-test
##
## data:  male$prediction_recode and female$prediction_recode
## t = 0.28766, df = 897.7, p-value = 0.7737
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.053688  4.102577
## sample estimates:
## mean of x mean of y
##  54.50000  53.97556
```

```r
df_stock <- df[df$generator=="stock",]
male <- df_stock[df_stock$gender=='0',]
female <- df_stock[df_stock$gender=='1',]
t.test(male$prediction_recode, female$prediction_recode)
```

```
##
##  Welch Two Sample t-test
##
## data:  male$prediction_recode and female$prediction_recode
## t = -2.7101, df = 888.23, p-value = 0.006857
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.807993 -1.408887
## sample estimates:
## mean of x mean of y
##  47.94017  53.04861
```

```r
df_bingo <- df[df$generator=="bingo",]
male <- df_bingo[df_bingo$gender=='0',]
female <- df_bingo[df_bingo$gender=='1',]
t.test(male$prediction_recode, female$prediction_recode)
```

```
##
##  Welch Two Sample t-test
##
## data:  male$prediction_recode and female$prediction_recode
## t = 1.6014, df = 893.73, p-value = 0.1096
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.599040  5.910151
## sample estimates:
## mean of x mean of y
##  53.33333  50.67778
```

```
data1 <- read_excel("../Data/PredictingOutcomes_ParticipantPredictions.xlsx", sheet = "Study 3A")
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

```
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

```
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 = 0.95624, p-value < 2.2e-16
```

apply spearman correlation on the dataframe

```
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 = 20.087, df = 1048, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4821104 0.5695624
## sample estimates:
##       cor
## 0.5272311
```

```
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 = 16.101, df = 348, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5888901 0.7096101
## sample estimates:
##       cor
## 0.6533844
```

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

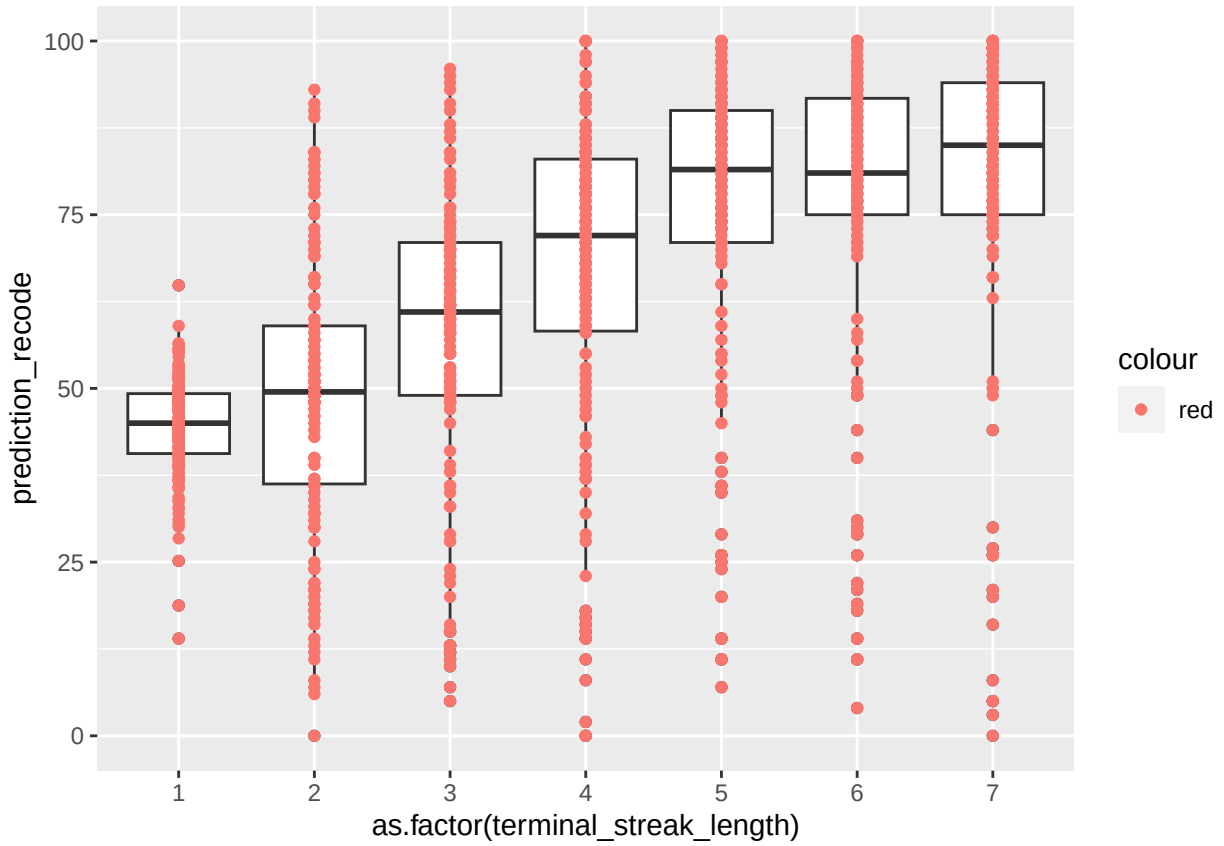
```
##
##  Pearson's product-moment correlation
##
## data:  df_stock$terminal_streak_length and df_stock$prediction_recode
## t = 9.5028, df = 348, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3665139 0.5333554
## sample estimates:
##      cor
## 0.453904
```

```
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 = 10.88, df = 348, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4212133 0.5780964
## sample estimates:
##       cor
## 0.5037976
```

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
```




```
map <- data.frame(unique(data$participant_id), data$age)
colnames(map) <- c("participant_id","age")
# map
```

```
data1 <- read_excel("../Data/PredictingOutcomes_ParticipantPredictions.xlsx", sheet = "Study 3A")
# only tke the data for columns participant_id, prediction_recode, prediction_recode
data1 <- data1[,c(2,3,8,10)]
# print(data1)
```

```
df <- merge(data1, map, by = "participant_id")
young <- df[df$age<'35',]
adult <- df[df$age>='35'& df$age<='60',]
```

```
## Warning in df$age >= "35" && df$age <= "60": 'length(x) = 2700 > 1' in coercion
## to 'logical(1)'

## Warning in df$age >= "35" && df$age <= "60": 'length(x) = 2700 > 1' in coercion
## to 'logical(1)'
```

```
t.test(young$prediction_recode, adult$prediction_recode)
```

```
##
##  Welch Two Sample t-test
##
## data:  young$prediction_recode and adult$prediction_recode
## t = 0.52902, df = 3918.6, p-value = 0.5968
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -1.171791  2.037847
## sample estimates:
## mean of x mean of y
##  52.64488  52.21185
```

```
df1 <- aggregate(prediction_recode~terminal_streak_length+age,df,FUN=mean)
colnames(df1) <- c("terminal_streak_length", "age", "prediction_recode")
young <- df[df$age<'35',]
adult <- df[df$age>='35'&& df$age<='60',]
```

```
## Warning in df$age >= "35" && df$age <= "60": 'length(x) = 2700 > 1' in coercion
## to 'logical(1)'

## Warning in df$age >= "35" && df$age <= "60": 'length(x) = 2700 > 1' in coercion
## to 'logical(1)'
```

```
t.test(young$prediction_recode, adult$prediction_recode)
```

```
##
##  Welch Two Sample t-test
##
## data:  young$prediction_recode and adult$prediction_recode
## t = 0.52902, df = 3918.6, p-value = 0.5968
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.171791  2.037847
## sample estimates:
## mean of x mean of y
##  52.64488  52.21185
```