```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
library(ggplot2)
```

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

# 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
```

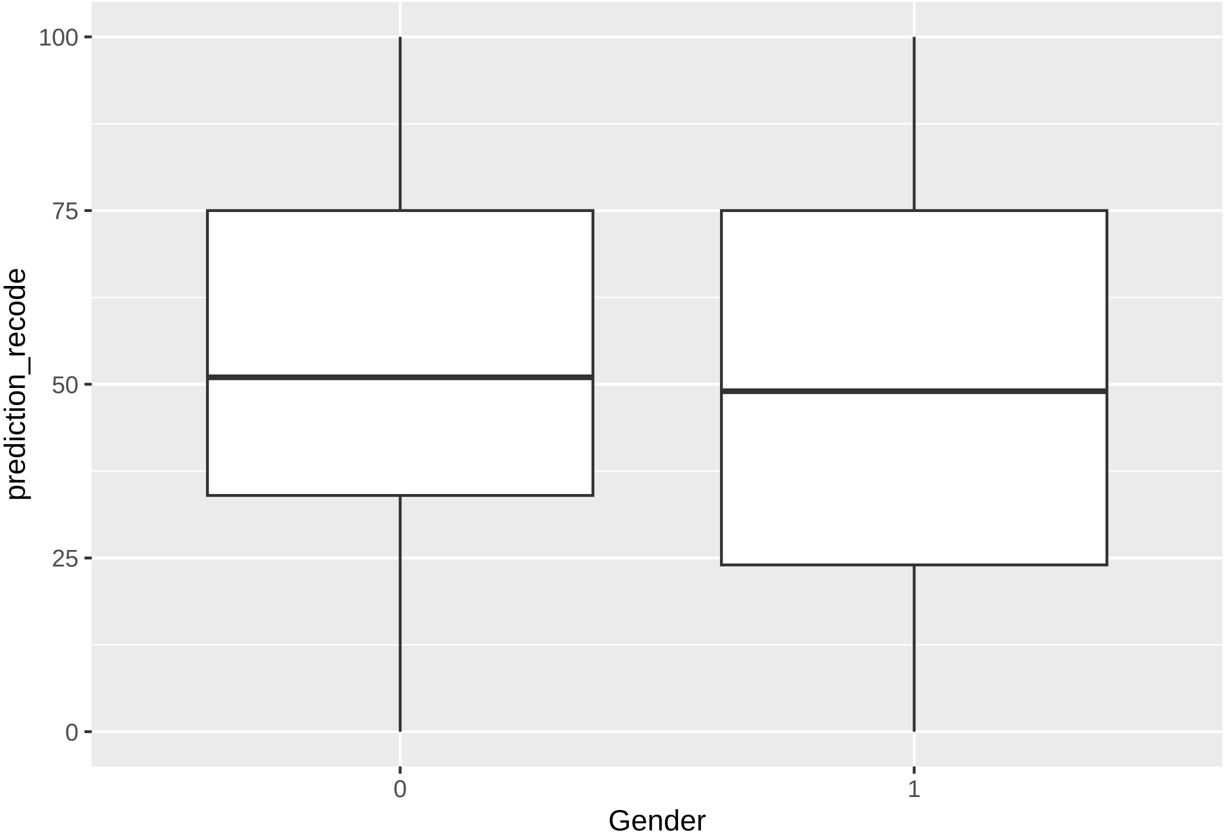
```r
data1 <- read_excel("../Data/PredictingOutcomes_ParticipantPredictions.xlsx", sheet = "Study 1A")
# only tke the data for columns participant_id, prediction_recode, prediction_recode
data1 <- data1[,c(2,3,10)]
# print(data1)
```

```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 = 3.8576, df = 2045.1, p-value = 0.0001181
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  2.101319 6.447286
## sample estimates:
## mean of x mean of y
##  53.64212  49.36782
```

apply scatter gram on male and female

```r
ggplot(df, aes(x = as.factor(gender), y = prediction_recode)) + geom_boxplot() + xlab("Gender")
```



```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 = 2.4113, df = 749.27, p-value = 0.01613
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.8476775 8.2732858
## sample estimates:
## mean of x mean of y
##  53.75096  49.19048
```

```
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 = 1.7288, df = 515.29, p-value = 0.08444
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4939742  7.7390139
## sample estimates:
## mean of x mean of y
##  55.71627  52.09375
```

```
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 = 2.2476, df = 772.82, p-value = 0.02489
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.514269 7.610526
## sample estimates:
## mean of x mean of y
##  51.53065  47.46825
```

```
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

```
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_sto
df_bingo <- aggregate(data_bingo$prediction_recode, by = list(data_bingo$terminal_streak_length,data_bin
```

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
```

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

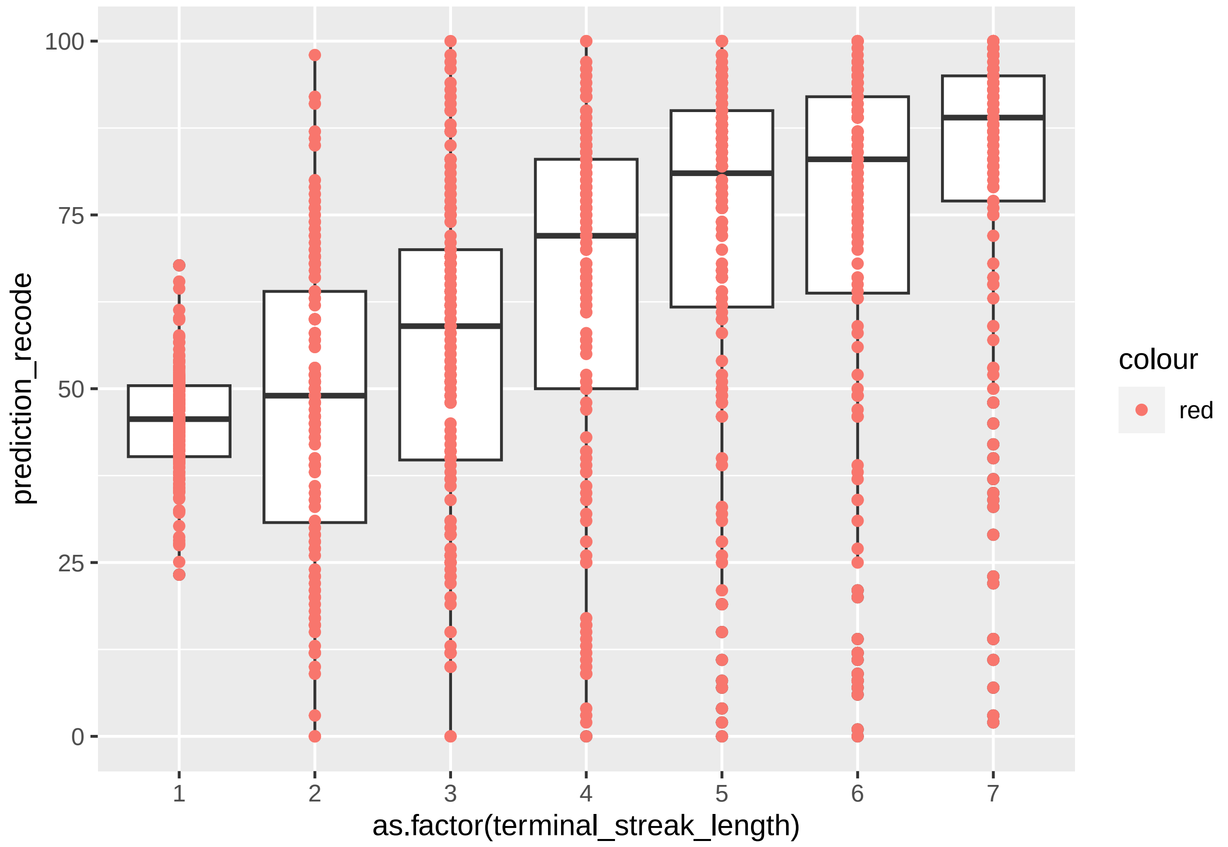
```
##
##  Pearson's product-moment correlation
##
## data:  df_stock$terminal_streak_length and df_stock$prediction_recode
## t = 12.048, df = 306, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4862811 0.6384972
## sample estimates: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



```
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 1A")
# only tke the data for columns participant_id, prediction_recode, prediction_recode
data1 <- data1[,c(2,3,8,10)]
# print(data1)
```

```r
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) = 2592 > 1' in coercion
## to 'logical(1)'

## Warning in df$age >= "35" && df$age <= "60": 'length(x) = 2592 > 1' in coercion
## to 'logical(1)'
```

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

```
##
##  Welch Two Sample t-test
##
## data:  young$prediction_recode and adult$prediction_recode
## t = -0.028997, df = 3117.8, p-value = 0.9769
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.762325  1.710958
## sample estimates:
## mean of x mean of y
##  51.89484  51.92052
```

```r
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) = 2592 > 1' in coercion
## to 'logical(1)'

## Warning in df$age >= "35" && df$age <= "60": 'length(x) = 2592 > 1' in coercion
## to 'logical(1)'
```

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

```
##
##  Welch Two Sample t-test
##
## data:  young$prediction_recode and adult$prediction_recode
## t = -0.028997, df = 3117.8, p-value = 0.9769
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.762325  1.710958
## sample estimates:
## mean of x mean of y
##  51.89484  51.92052
```