```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
```

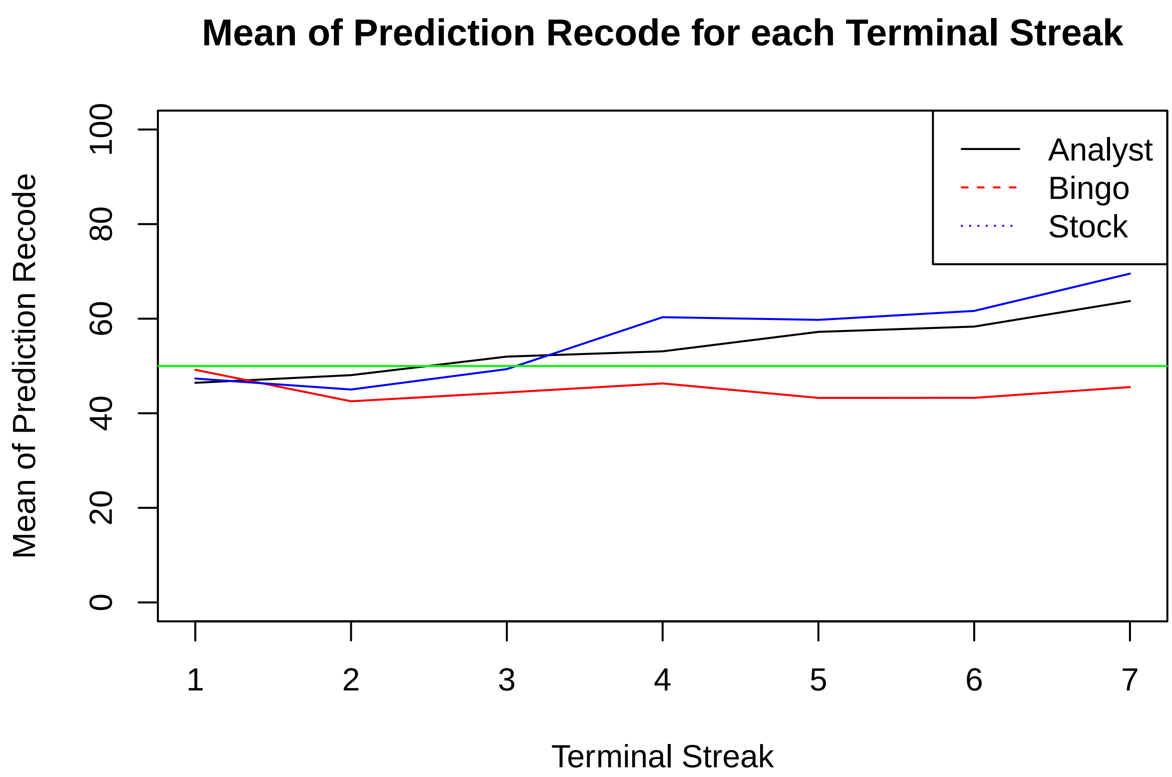
```r
data <- read_excel("../Data/PredictingOutcomes_ParticipantPredictions.xlsx", sheet = "Study 2A")

# divide the data based on the generator
data1 <- data[data$generator == "analyst",]
data2 <- data[data$generator == "bingo",]
data3 <- data[data$generator == "stock",]
```

calculate the mean of prediciotn_recode for each terminwal streak from 1 to 7

```
mean1 <- aggregate(data1$prediction_recode, by = list(data1$terminal_streak_length), FUN = mean)
mean2 <- aggregate(data2$prediction_recode, by = list(data2$terminal_streak_length), FUN = mean)
mean3 <- aggregate(data3$prediction_recode, by = list(data3$terminal_streak_length), FUN = mean)
```

```
plot(mean1$Group.1,mean1$x, type = "l",ylim=c(0,100), xlab = "Terminal Streak", ylab = "Mean of Predicti
lines(mean2$Group.1,mean2$x, col = "red")
lines(mean3$Group.1,mean3$x,  col = "blue")
abline(h = 50, col = "green")
legend("topright", legend = c("Analyst", "Bingo", "Stock"), col = c("black", "red", "blue"), lty = 1:3)
```




```
aov1<-aov_ez('participant_id','prediction_recode',data, between=c('generator'),within=c('terminal_streal
```

```
## Converting to factor: generator
```

```
## Warning: More than one observation per design cell, aggregating data using `fun_aggregate = mean`.
## To turn off this warning, pass `fun_aggregate = mean` explicitly.
```

```
## Contrasts set to contr.sum for the following variables: generator
```

```
aov1
```

```
## Anova Table (Type 3 tests)
##
## Response: prediction_recode
##                              Effect           df     MSE        F  ges p.value
## 1                         generator       2, 153 1915.52  6.96 ** .040    .001
## 2            terminal_streak_length 4.08, 624.73  555.43 9.82 *** .034   <.001
## 3 generator:terminal_streak_length 8.17, 624.73  555.43 3.61 *** .025   <.001
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’+’ 0.1 ’ ’ 1
##
## Sphericity correction method: GG
```

```r
pairwise.t.test(data$prediction_recode, data$generator, p.adjust.method = "bonferroni")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  data$prediction_recode and data$generator
##
##       analyst bingo
## bingo 0.249   -
## stock 0.722   0.012
##
## P value adjustment method: bonferroni
```

```r
aov1<-aov_ez('participant_id','prediction_recode',data, between=c('generator'),within=c('terminal_strea
```

```
## Converting to factor: generator
```

```
## Warning: More than one observation per design cell, aggregating data using ‘fun_aggregate = mean‘.
## To turn off this warning, pass ‘fun_aggregate = mean‘ explicitly.
```

```
## Contrasts set to contr.sum for the following variables: generator
```

```r
aov1
```

```
## Anova Table (Type 3 tests)
##
## Response: prediction_recode
##                              Effect           df     MSE        F  ges p.value
## 1                         generator       2, 153 1915.52  6.96 ** .040    .001
## 2            terminal_streak_length 4.08, 624.73  555.43 9.82 *** .034   <.001
## 3 generator:terminal_streak_length 8.17, 624.73  555.43 3.61 *** .025   <.001
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’+’ 0.1 ’ ’ 1
##
## Sphericity correction method: GG
```

```r
pairwise.t.test(data$prediction_recode, data$generator, p.adjust.method = "bonferroni")
```

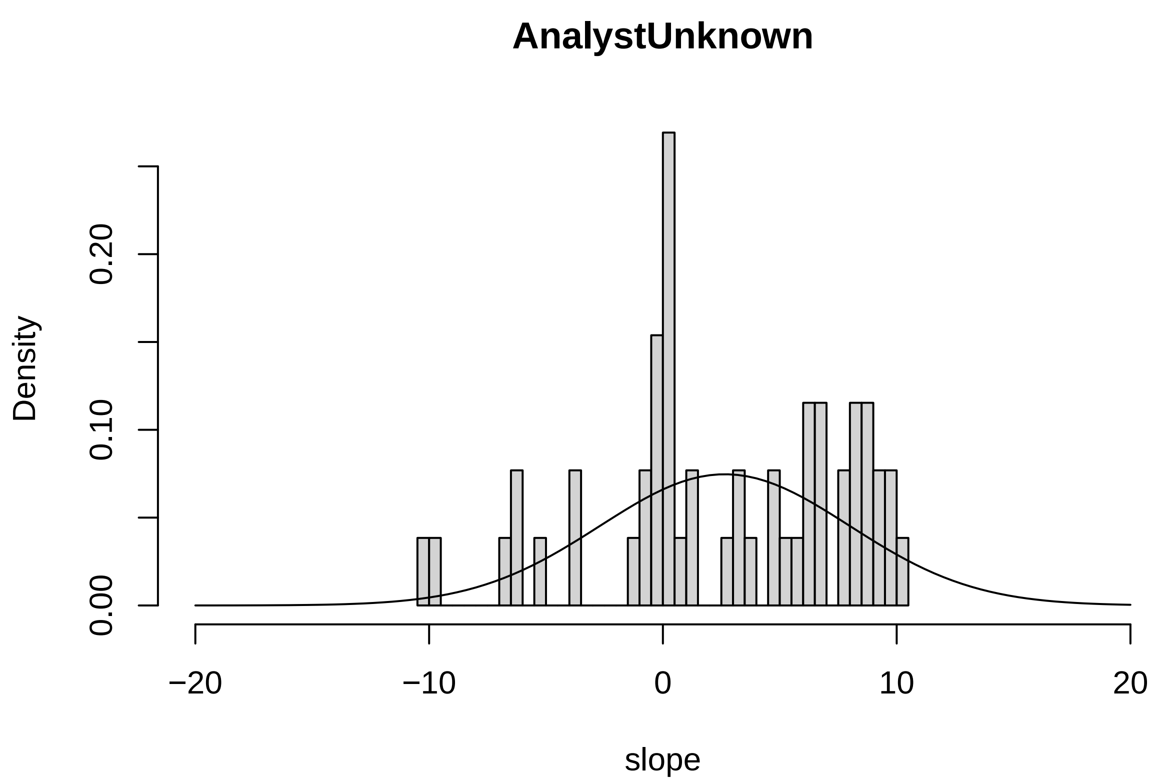
```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  data$prediction_recode and data$generator
##
##       analyst bingo
## bingo 0.249   -
## stock 0.722   0.012
##
## P value adjustment method: bonferroni
```

```r
id <- unique(data1$participant_id)

slope <- c()

for (i in id){
x <- as.character(i)
datax <- data1[data1$participant_id == x,]
model <- lm(prediction_recode ~ terminal_streak_length, data = datax)
slope <- c(slope, coef(model)[2])
}

hist(slope,breaks=30,xlim=c(-20,20),prob=TRUE,main="AnalystUnknown")
curve(dnorm(x, mean = mean(slope), sd = sd(slope)), add = TRUE, col = "black")
```

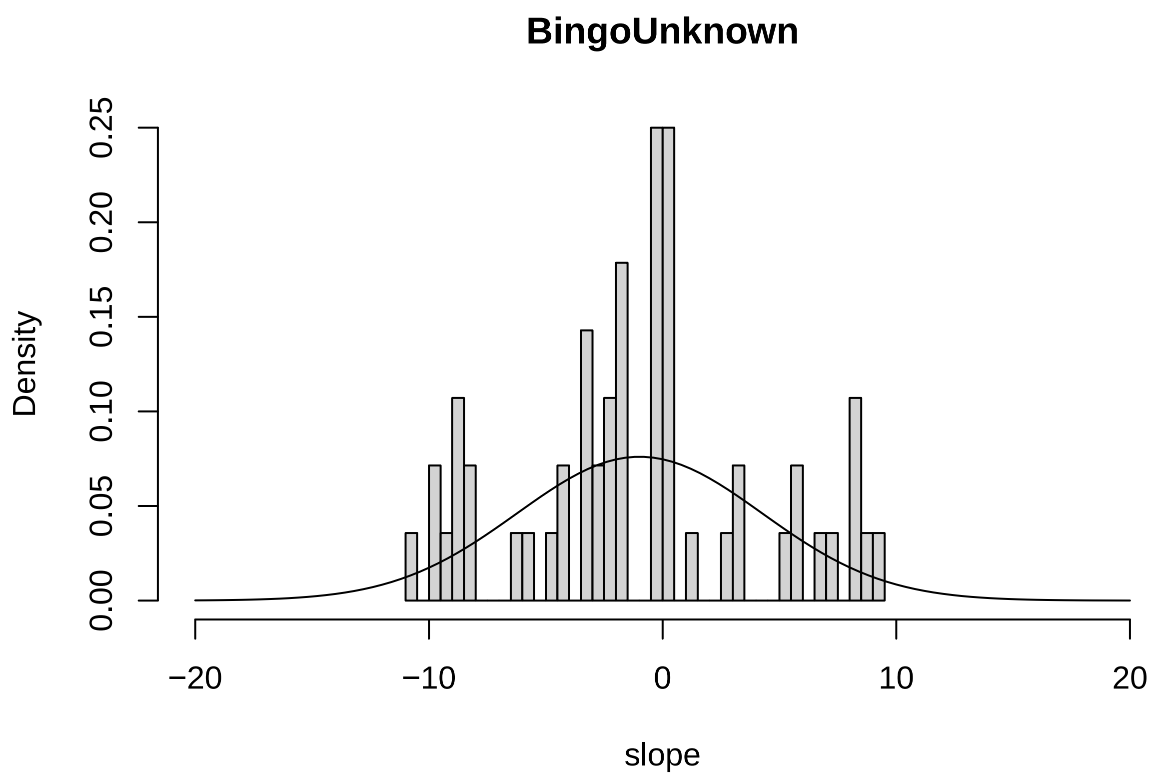


```r
id <- unique(data2$participant_id)

slope <- c()

for (i in id){
x <- as.character(i)
datax <- data2[data2$participant_id == x,]
model <- lm(prediction_recode ~ terminal_streak_length, data = datax)
slope <- c(slope, coef(model)[2])
}

hist(slope,breaks=30,xlim=c(-20,20),prob=TRUE,main="BingoUnknown")
curve(dnorm(x, mean = mean(slope), sd = sd(slope)), add = TRUE, col = "black")
```

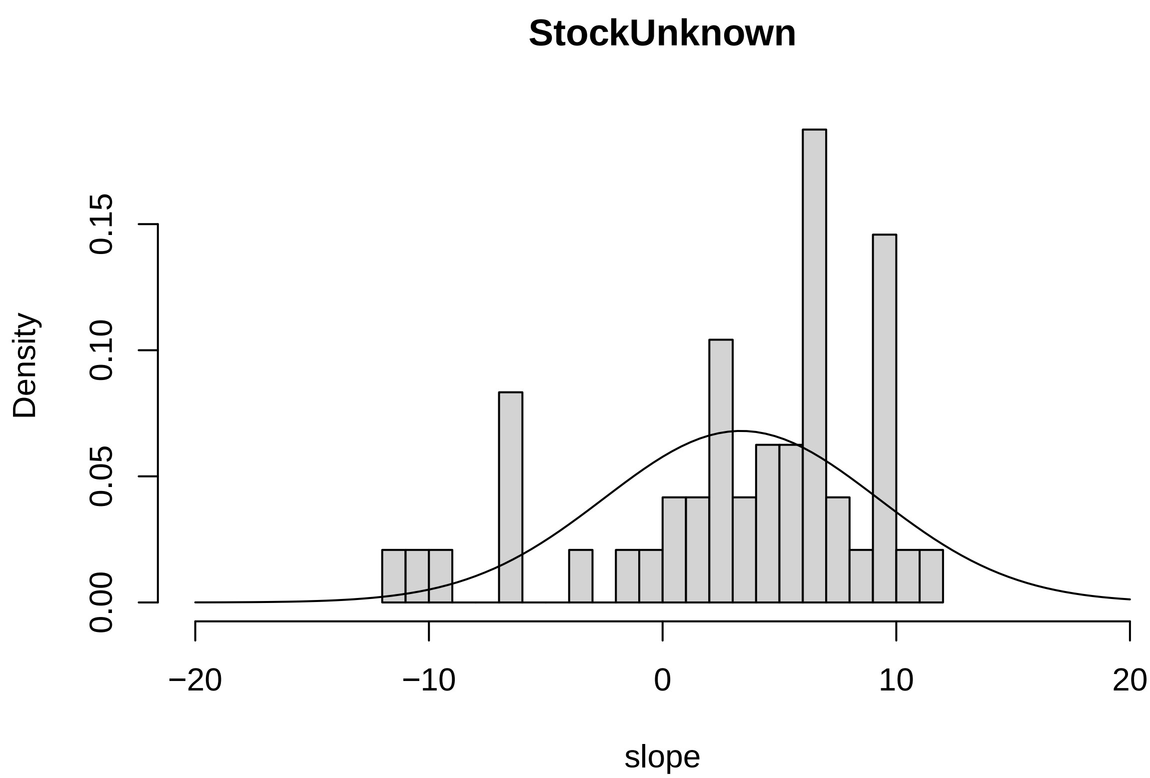



```r
id <- unique(data3$participant_id)

slope <- c()

for (i in id){
x <- as.character(i)
datax <- data3[data3$participant_id == x,]
model <- lm(prediction_recode ~ terminal_streak_length, data = datax)
slope <- c(slope, coef(model)[2])
}
```

```r
hist(slope,breaks=30,xlim=c(-20,20),prob=TRUE,main="StockUnknown")
curve(dnorm(x, mean = mean(slope), sd = sd(slope)), add = TRUE, col = "black")
```



Hypothesis testing

```r
data_dem<- read_excel("../Data/PredictingOutcomes_ParticipantDemographics.xlsx", sheet = "Study 2A")

# print(data)
```

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

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

```r
dataf <- data[,c(2,3,10)]
# print(data1)
```

```r
df <- merge(dataf, 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 = 0.34336, df = 2458.7, p-value = 0.7314
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.507399  2.147346
## sample estimates:
## mean of x mean of y
##  49.33527  49.01530
```

```
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 = 1.7831, df = 827.61, p-value = 0.07493
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.272363  5.676994
## sample estimates:
## mean of x mean of y
##  50.62261  47.92029
```

```
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 = -2.5621, df = 846.03, p-value = 0.01058
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.001251 -1.059744
## sample estimates:
## mean of x mean of y
##  48.68162  53.21212
```

```
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 = 1.5742, df = 745.94, p-value = 0.1159
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.6081508  5.5313886
## sample estimates:
## mean of x mean of y
##  48.67921  46.21759
```