```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
```

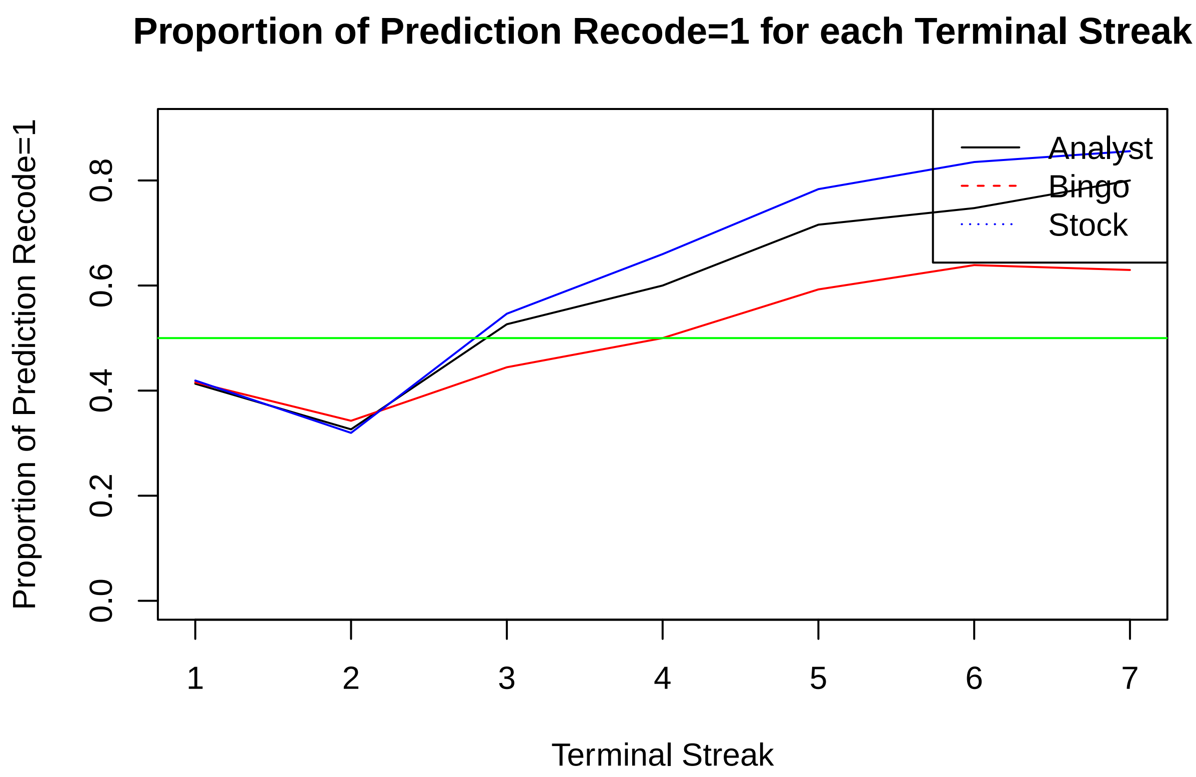
```r
data <- read_excel("../Data/PredictingOutcomes_ParticipantPredictions.xlsx", sheet = "Study 1B")

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

calculate the proportion of participants who predicted the prediction_recode=1 for each terminal_streak_length from 1 to 7

```
prop1 <- aggregate(data1$prediction_recode, by = list(data1$terminal_streak_length), FUN = mean)
prop2 <- aggregate(data2$prediction_recode, by = list(data2$terminal_streak_length), FUN = mean)
prop3 <- aggregate(data3$prediction_recode, by = list(data3$terminal_streak_length), FUN = mean)
```

```
plot(prop1$Group.1,prop1$x, type = "l",ylim=c(0.0,0.9), xlab = "Terminal Streak", ylab = "Proportion of
lines(prop2$Group.1,prop2$x, col = "red")
lines(prop3$Group.1,prop3$x,  col = "blue")
abline(h = 0.5, 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
```

```r
aov1
```

```
## Anova Table (Type 3 tests)
##
## Response: prediction_recode
##                               Effect            df  MSE         F  ges p.value
## 1                          generator        2, 297 0.50   5.59 ** .014    .004
## 2             terminal_streak_length  5.28, 1567.01 0.15 61.50 *** .114   <.001
## 3 generator:terminal_streak_length 10.55, 1567.01 0.15    1.96 * .008    .031
## ---
## 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")
```

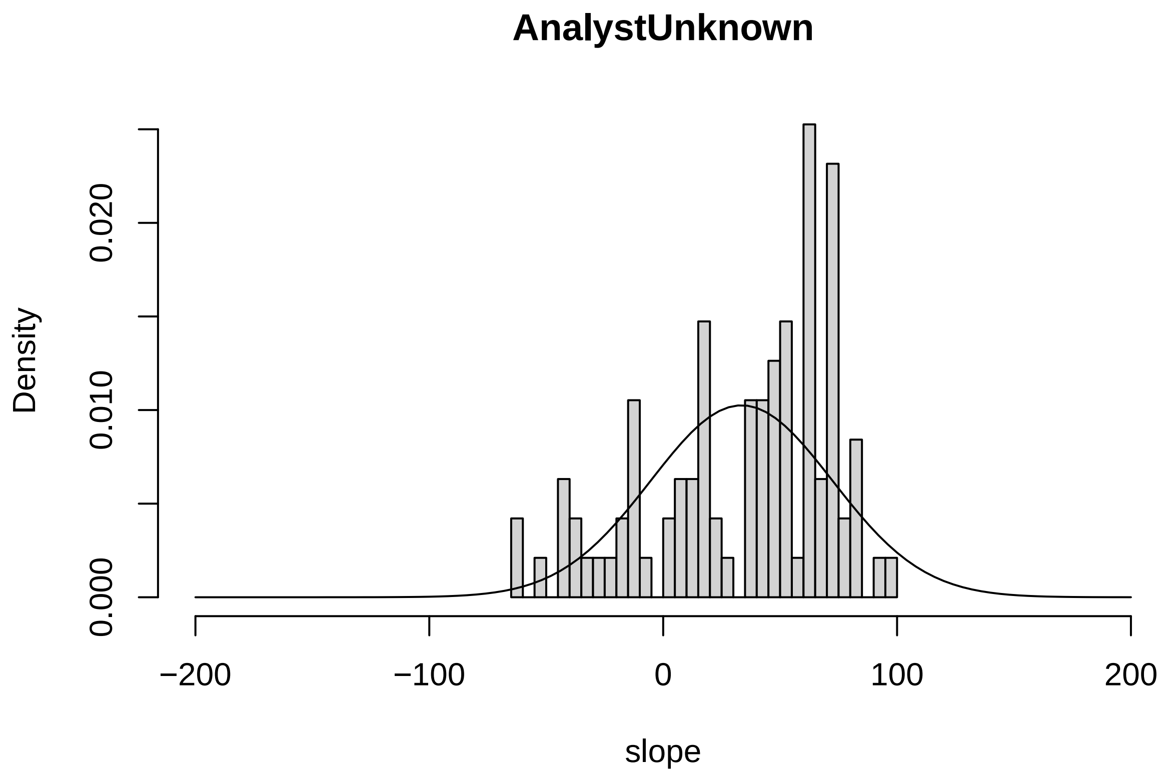
```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  data$prediction_recode and data$generator
##
##       analyst bingo
## bingo 0.2181  -
## stock 0.7279  0.0079
##
## P value adjustment method: bonferroni
```

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

slope <- c()

for (i in id){
x <- as.character(i)
datax <- data1[data1$participant_id == x,]
# datax <- datax[datax$terminal_streak_length != "1",]
model <- glm(prediction_recode ~ terminal_streak_length, data = datax)
beta <- coef(model)[2]
odds <- exp(beta)
slope <- c(slope, beta)
}

slope <- slope*500
hist(slope,breaks=30,xlim=c(-200,200),prob=TRUE,main="AnalystUnknown")
curve(dnorm(x, mean = mean(slope), sd = sd(slope)), add = TRUE, col = "black")
```



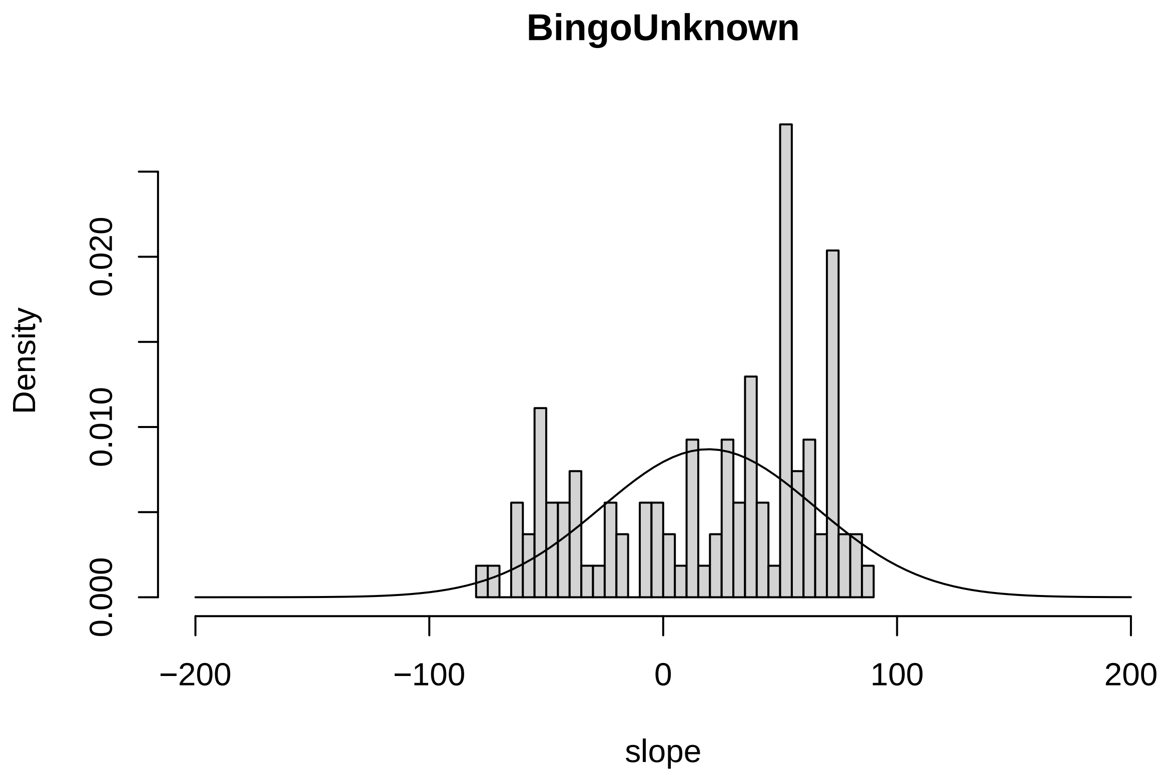
logistic regression

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

slope <- c()

for (i in id){
x <- as.character(i)
datax <- data2[data2$participant_id == x,]
model <- glm(prediction_recode ~ terminal_streak_length, data = datax)
slope <- c(slope, coef(model)[2])
}
slope <- slope*500

hist(slope,breaks=30,xlim=c(-200,200),prob=TRUE,main="BingoUnknown")
curve(dnorm(x, mean = mean(slope), sd = sd(slope)), add = TRUE, col = "black")
```

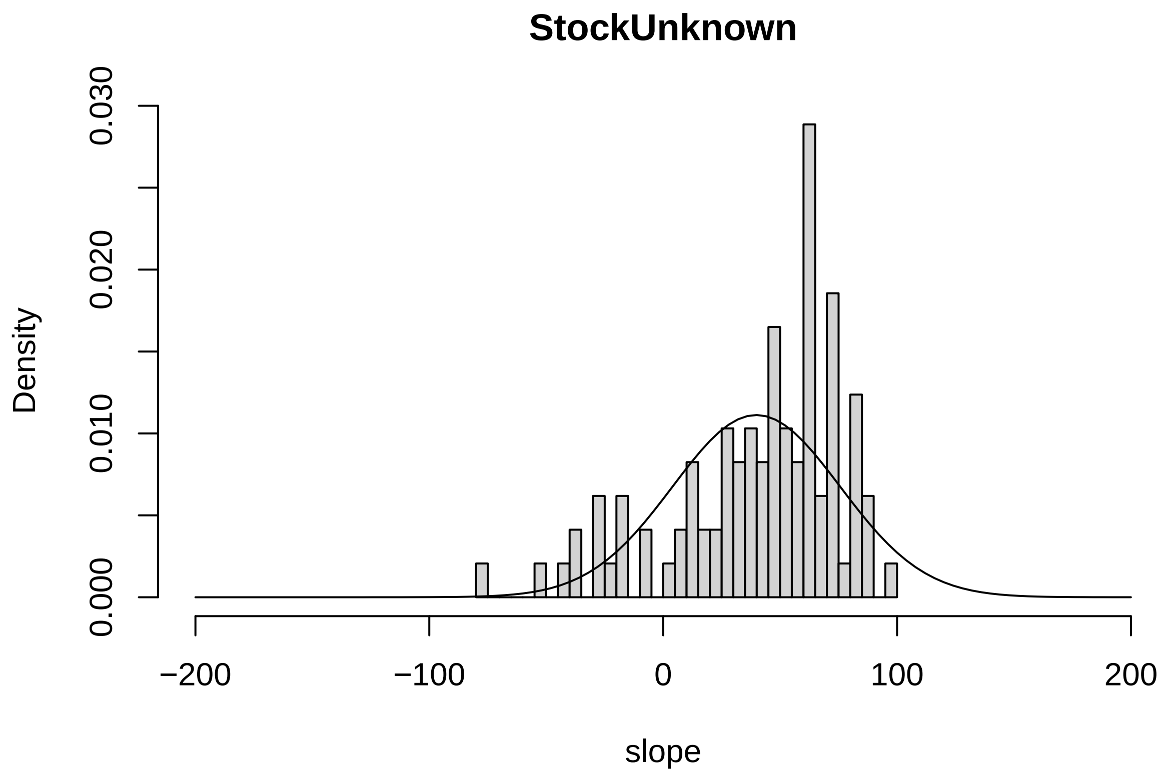



```
id <- unique(data3$participant_id)

slope <- c()

for (i in id){
x <- as.character(i)
datax <- data3[data3$participant_id == x,]
model <- glm(prediction_recode ~ terminal_streak_length, data = datax)
slope <- c(slope, coef(model)[2])
}
slope <- slope*500

hist(slope,breaks=30,xlim=c(-200,200),prob=TRUE,main="StockUnknown")
curve(dnorm(x, mean = mean(slope), sd = sd(slope)), add = TRUE, col = "black")
```




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

# print(data)
```

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

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

```
dataf <- data[,c(2,3,8,10)]
# print(data1)
```

```
df <- merge(dataf, map, by = "participant_id")
df_total <- df[df$gender=='0' | df$gender=='1',]
# male <- df[df$gender=='0',]
# female <- df[df$gender=='1',]
# chisq.test(male$prediction_recode, female$prediction_recode,correct=FALSE)
check <- table(df_total$gender, df_total$terminal_streak_length)
print(check)
```

```
##
##        1    2    3    4    5    6    7
##   0 1872  156  156  156  156  156  156
##   1 1704  142  142  142  142  142  142
```

```r
test <- table(df_total$gender, df_total$prediction_recode)
print(test)
```

```
##
##         0    1
##   0 1467 1341
##   1 1333 1223
```

```r
chisq.test(test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 0.0015883, df = 1, p-value = 0.9682
```

```r
df <- df_total[df_total$generator=='analyst',]
test <- table(df$gender, df$prediction_recode)
print(test)
```

```
##
##       0   1
##   0 410 400
##   1 476 424
```

```r
chisq.test(test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 0.79247, df = 1, p-value = 0.3734
```

```r
df <- df_total[df_total$generator=='bingo',]
test <- table(df$gender, df$prediction_recode)
print(test)
```

```
##
##       0   1
##   0 559 449
##   1 485 415
```

```r
chisq.test(test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 0.41034, df = 1, p-value = 0.5218
```

```r
df <- df_total[df_total$generator=='stock',]
test <- table(df$gender, df$prediction_recode)
print(test)
```

```
##
##       0   1
##   0 498 492
##   1 372 384
```

```r
chisq.test(test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 0.16469, df = 1, p-value = 0.6849
```