

```
library(readxl)
library(afex)
```

```
## Loading required package: lme4
```

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

```
## *****
```

```
## 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_ParticipantPredictions.xlsx", sheet = "Study 3B")
```

```
# 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
```

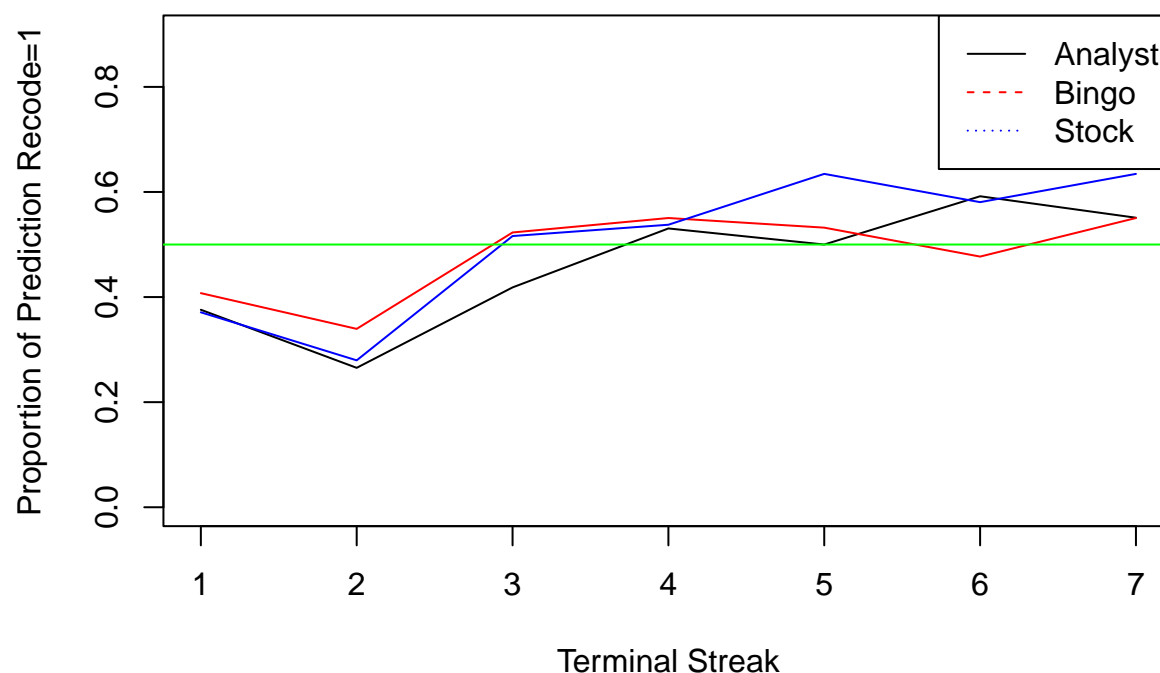
```
lines(prop2$Group.1, prop2$x, col = "red")
```

```
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)
```

## Proportion of Prediction Recode=1 for each Terminal Streak



calculate the effect of condition on participant prediction

```
model1 <- lmer(prediction_recode ~ generator + (1|participant_id), data = data)
summary(model1)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: prediction_recode ~ generator + (1 | participant_id)
## Data: data
##
## REML criterion at convergence: 7666.3
##
## Scaled residuals:
##    Min      1Q  Median      3Q      Max
## -1.2145 -0.8588 -0.6694  1.0838  1.5428
##
## Random effects:
## Groups           Name          Variance Std.Dev.
## participant_id (Intercept) 0.01102  0.105
## Residual              0.23330  0.483
## Number of obs: 5400, groups: participant_id, 300
##
## Fixed effects:
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)   0.40930   0.01564 296.99999  26.167  <2e-16 ***
## generatorbingo  0.02750   0.02156 296.99999   1.276    0.203
```

```
## generatorstock    0.01484    0.02242 296.99999    0.662    0.509
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) gnrtrb
## generatrbnb -0.726
## genertrstck -0.698  0.506
```

```
anova(model1)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## generator 0.37986 0.18993      2   297  0.8141  0.444
```

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

```
## 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, 297 0.61          0.58 .002    .561
## 2 terminal_streak_length 5.16, 1533.11 0.17 23.22 *** .044   <.001
## 3 generator:terminal_streak_length 10.32, 1533.11 0.17          1.52 .006    .123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
##
## Sphericity correction method: GG
```

```
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 0.27      -
## stock 1.00      1.00
##
## P value adjustment method: bonferroni
```

```

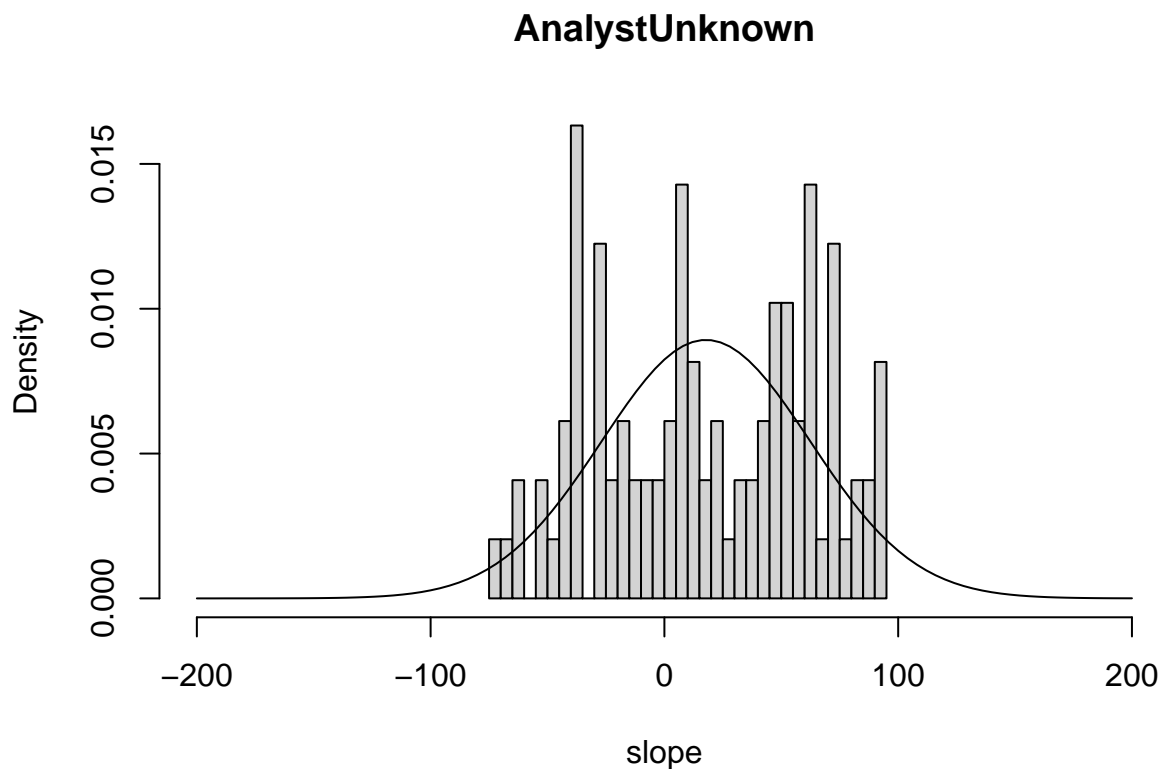
id <- unique(data1$participant_id)

slope <- c()

for (i in id){
  x <- as.character(i)
  datax <- data1[data1$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="AnalystUnknown")
curve(dnorm(x, mean = mean(slope), sd = sd(slope)), add = TRUE, col = "black")

```



```

id <- unique(data2$participant_id)

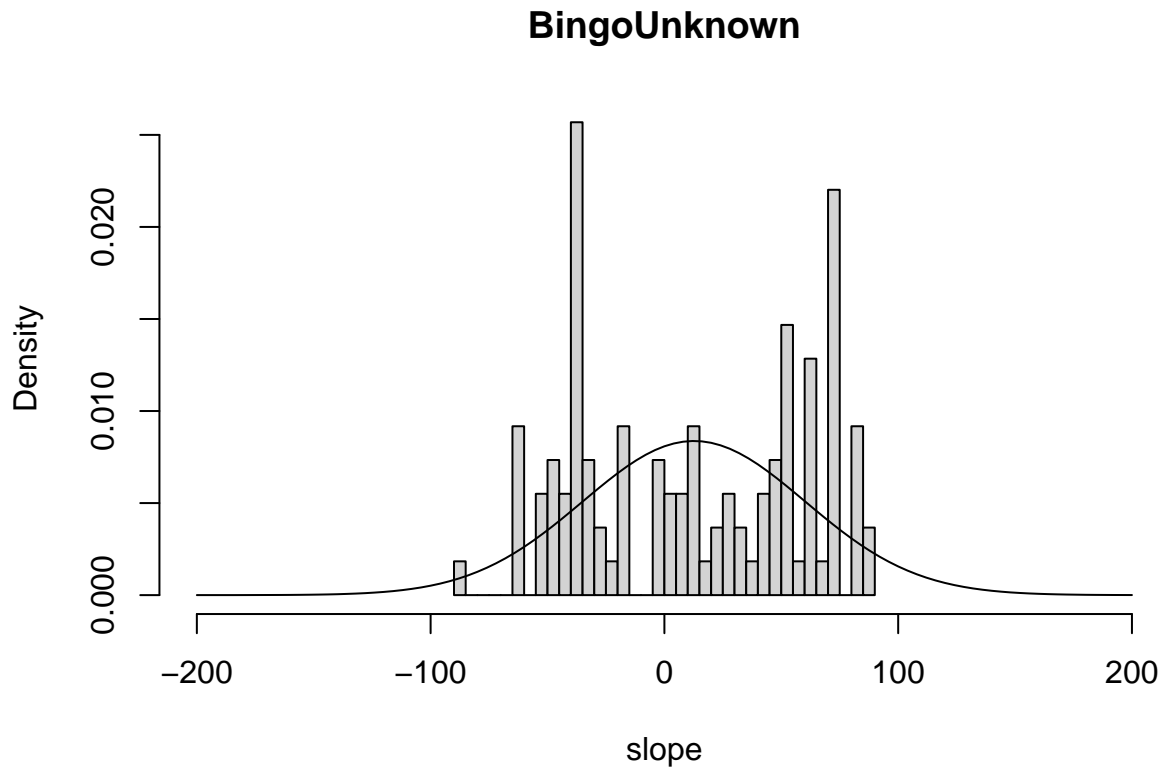
slope <- c()

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")
```

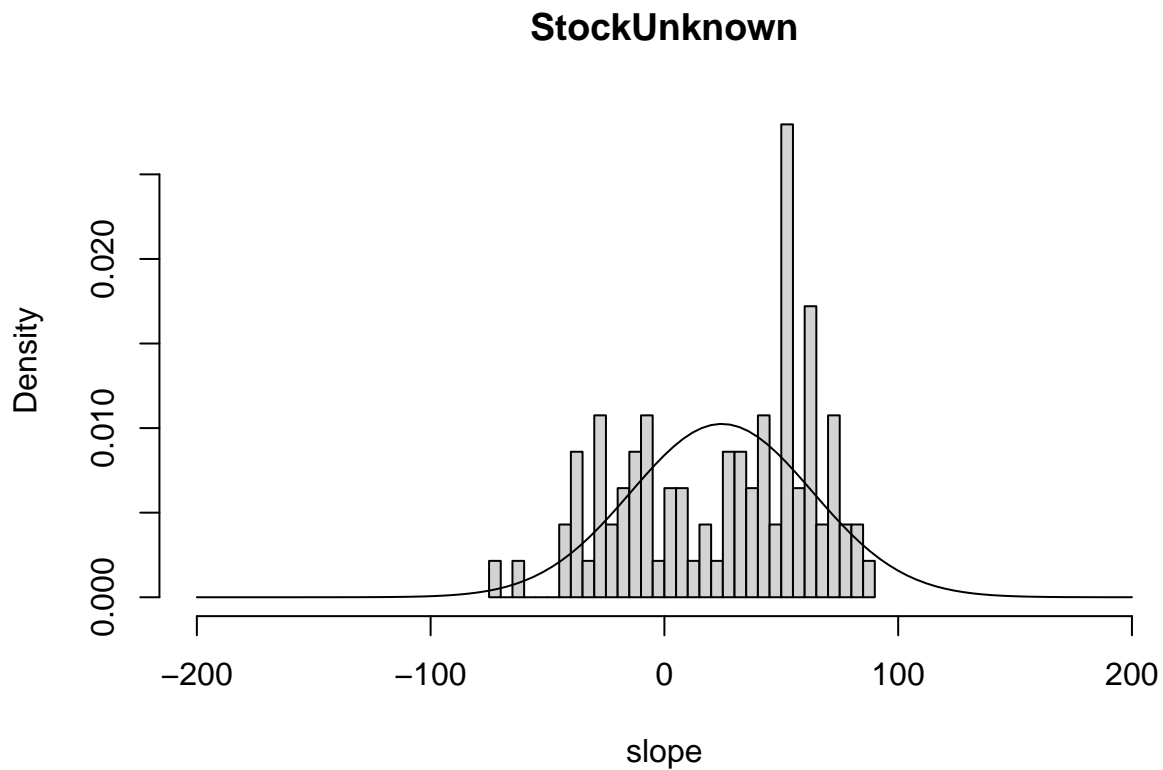


```
id <- unique(data3$participant_id)

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 1860  155  155  155  155  155  155
## 1 1716  143  143  143  143  143  143
```

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

```
##
##      0      1
## 0 1631 1159
## 1 1458 1116
```

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

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 1.733, df = 1, p-value = 0.188
```

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

```
##
##      0      1
## 0 462 330
## 1 580 392
```

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

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 0.26993, df = 1, p-value = 0.6034
```

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

```
##
##      0      1
## 0 630 468
## 1 462 384
```

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

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 1.3759, df = 1, p-value = 0.2408
```

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

```
##  
##      0    1  
## 0 539 361  
## 1 416 340
```

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

```
##  
## Pearson's Chi-squared test with Yates' continuity correction  
##  
## data:  test  
## X-squared = 3.7826, df = 1, p-value = 0.05179
```