```r
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
```

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

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

give count of entries in all three data

```r
nrow(data1)
```

```
## [1] 900
```

```r
nrow(data2)
```

```
## [1] 900
```

```r
nrow(data3)
```

```
## [1] 900
```

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

print length of data1 prediciton recode and list of terminal_streak of data1

```r
length(data1$prediction_recode)
```

```
## [1] 900
```

```r
length(data1$terminal_streak)
```

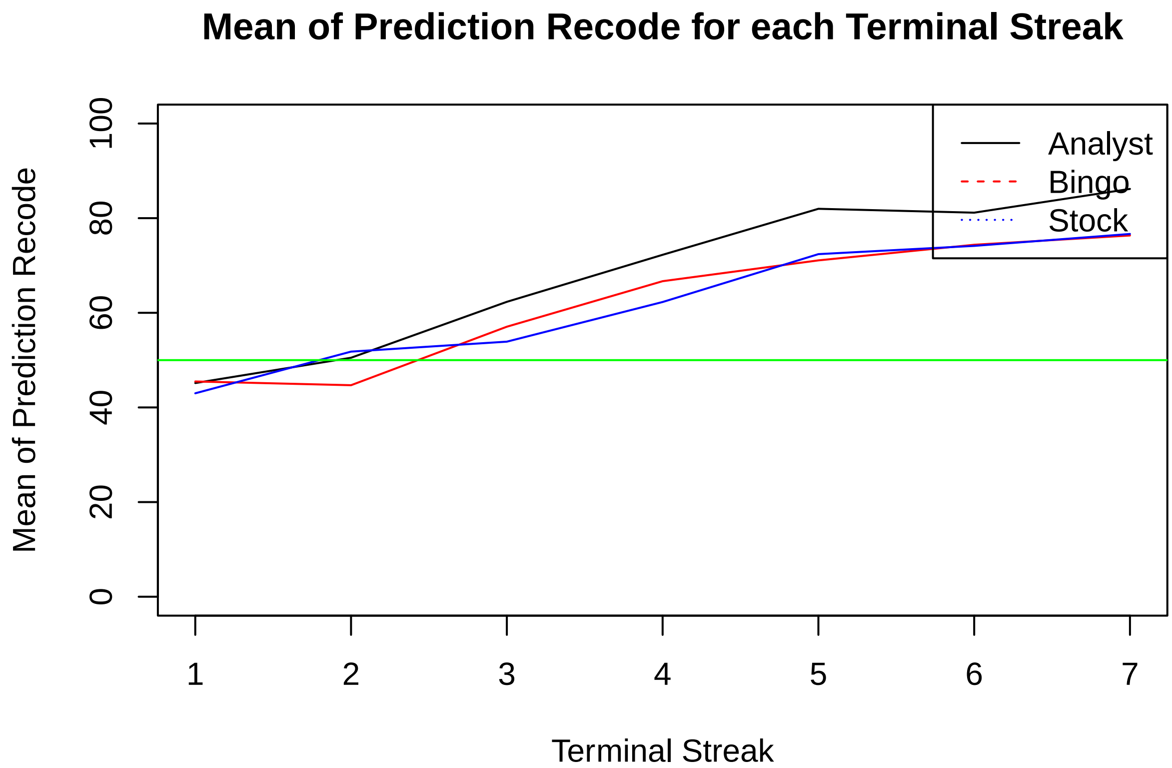
```
## Warning: Unknown or uninitialised column: ‘terminal_streak‘.

## [1] 0
```

```r
mean1 <- aggregate(data1$prediction_recode, by = list(data1$terminal_streak_length), FUN = mean)
```

```r
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)
```

```r
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)
```



calculate the effect of condition on participant predicition

```r
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: 25429.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.03648 -0.84669 -0.04777  0.84530  1.92349
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  participant_id (Intercept)   2.809   1.676
##  Residual                   720.297  26.838
## Number of obs: 2700, groups:  participant_id, 150
##
## Fixed effects:
##                 Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)      54.2378     0.9255 147.0000  58.605  < 2e-16 ***
## generatorbingo  -2.2322     1.3088 147.0000  -1.706  0.09021 .
## generatorstock  -3.8456     1.3088 147.0000  -2.938  0.00383 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) gnrtrb
## generatrbng -0.707
## genertrstck -0.707  0.500
```

```r
anova(model1)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
## generator 6271.9  3135.9     2   147  4.3537 0.01456 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
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
## 1                          generator        2, 147 1241.81     3.82 * .022
## 2              terminal_streak_length  5.09, 748.15  317.00 109.79 *** .297
## 3 generator:terminal_streak_length 10.18, 748.15  317.00       1.45 .011
##   p.value
## 1    .024
## 2   <.001
## 3    .151
## ---
## 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")
```

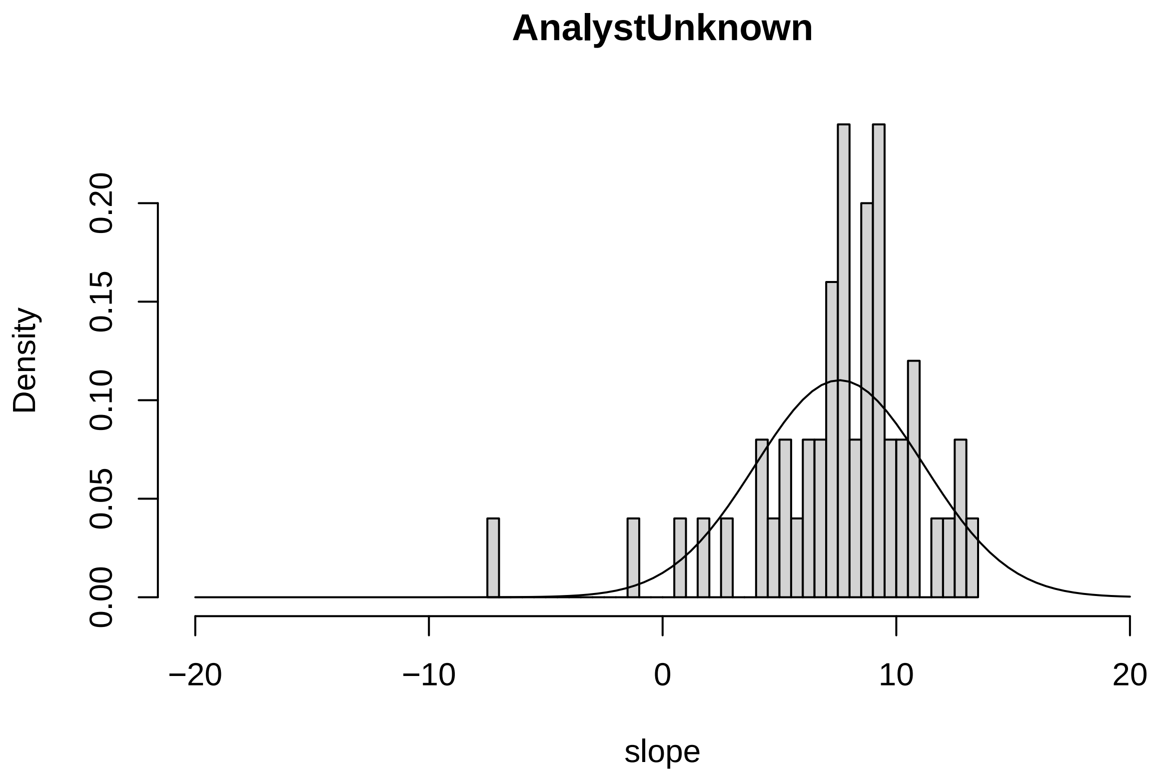
```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  data$prediction_recode and data$generator
##
##       analyst bingo
## bingo 0.2351  -
## stock 0.0073  0.6096
##
## P value adjustment method: bonferroni
```

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

slope <- c()

for (i in id){
x <- as.character(i)
datax <- data1[data1$participant_id == x,]
model <- 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")
```

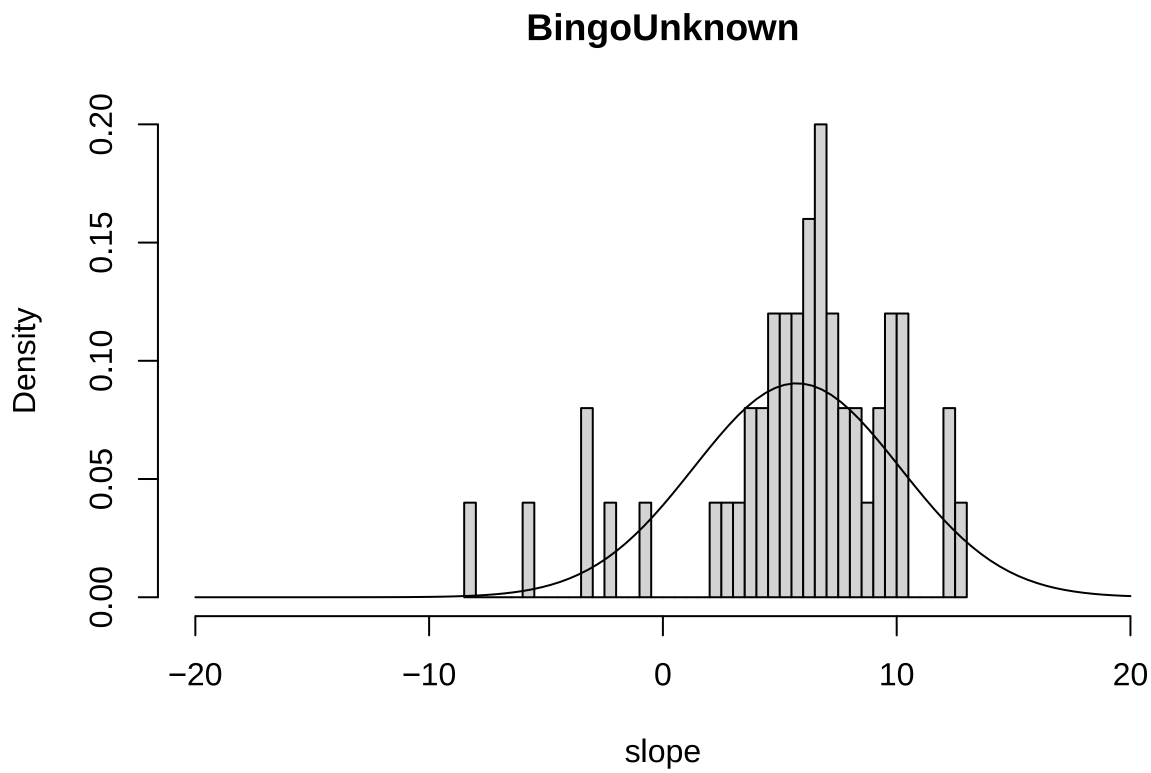



```
id <- unique(data2$participant_id)

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")
```

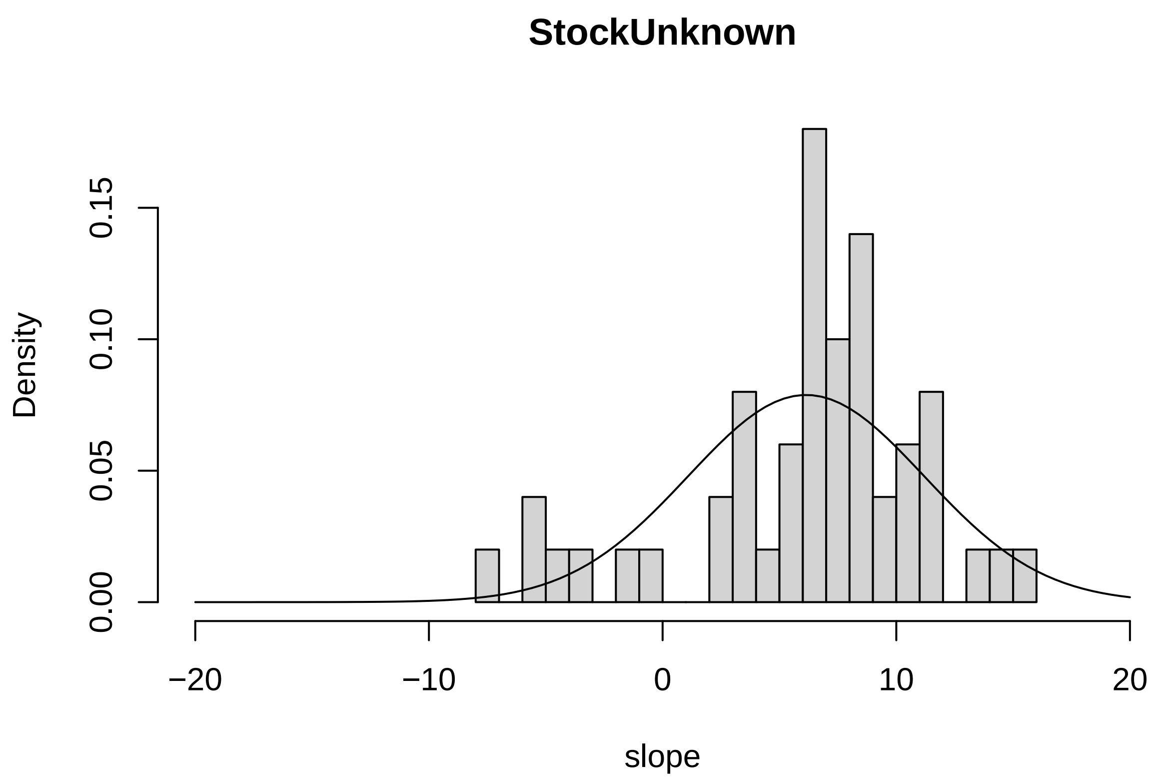



```
id <- unique(data3$participant_id)

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])
}

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

```
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

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

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

```
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 = 0.39691, df = 2387.3, p-value = 0.6915
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.663982  2.508511
## sample estimates:
## mean of x mean of y
##  52.35408  51.93182
```

```
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 = -0.60166, df = 868.74, p-value = 0.5476
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.762569  2.527755
## sample estimates:
## mean of x mean of y
##  53.59556  54.71296
```

```
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 = 0.5483, df = 857.11, p-value = 0.5836
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.692955  4.780806
## sample estimates:
## mean of x mean of y
##  50.87243  49.82850
```

```
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 = 0.92871, df = 598.53, p-value = 0.3534
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.870979  5.227893
## sample estimates:
## mean of x mean of y
##  52.64337  50.96491
```