```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
```

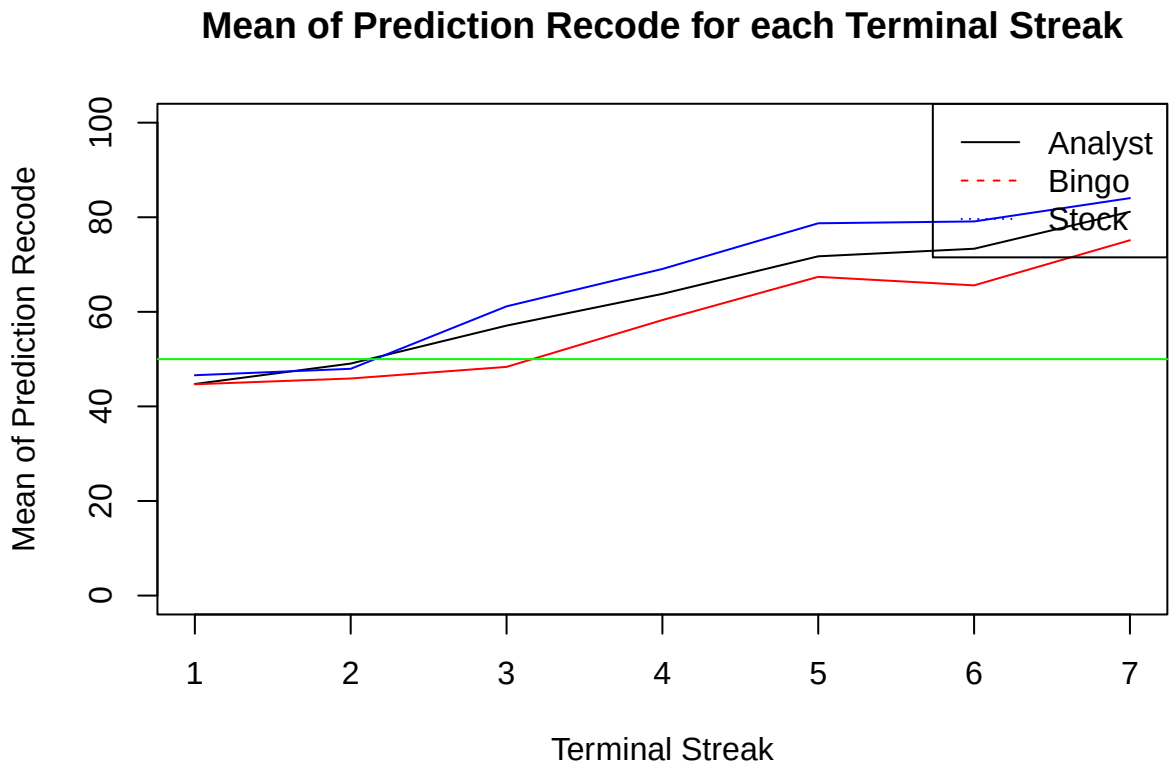
```r
library(MuMIn)
library(afex)
```

```
## ************
## 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 1A")

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

```r
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)
```

```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 mean of prediciotn_recode for each participant_id

```r
mean1_id <- aggregate(data1$prediction_recode, by = list(data1$participant_id), FUN = mean)
mean2_id <- aggregate(data2$prediction_recode, by = list(data2$participant_id), FUN = mean)
mean3_id <- aggregate(data3$prediction_recode, by = list(data3$participant_id), FUN = mean)
```

Data is not normally distributed

```r
shapiro.test(mean1_id$x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mean1_id$x
## W = 0.92966, p-value = 0.005365
```

```r
shapiro.test(mean2_id$x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mean2_id$x
## W = 0.95628, p-value = 0.06212
```

```r
shapiro.test(mean3_id$x)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mean3_id$x
## W = 0.97847, p-value = 0.5734
```

apply kruskal wais test so that the p value is 0.018

```r
kruskal.test(mean1_id$x, mean2_id$x, mean3_id$x)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  mean1_id$x and mean2_id$x
## Kruskal-Wallis chi-squared = 43.836, df = 44, p-value = 0.4786
```

apply a one way mixed anova to test the effect of condition and one within streak length on the rating of probability that the terminal streak would repeat

```r
model <- lm(prediction_recode ~ generator , data = data)
summary(model)
```

```
##
## Call:
## lm(formula = prediction_recode ~ generator, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.399 -21.824  -0.824  22.164  50.176
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     51.8356     0.9009  57.537   <2e-16 ***
## generatorbingo  -2.0111     1.2741  -1.578   0.1146
## generatorstock   2.5634     1.3168   1.947   0.0517 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.03 on 2589 degrees of freedom
## Multiple R-squared:  0.004645,   Adjusted R-squared:  0.003876
## F-statistic: 6.041 on 2 and 2589 DF,  p-value: 0.002412
```

tell the p value

```
anova(model)
```

```
## Analysis of Variance Table
##
## Response: prediction_recode
##             Df  Sum Sq Mean Sq F value   Pr(>F)
## generator    2    8826  4412.9  6.0412 0.002412 **
## Residuals 2589 1891174   730.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

calculate the effect of condition on participant predicition

```
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: 24430.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.12973 -0.80433 -0.03269  0.81549  2.01101
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  participant_id (Intercept)  15.25    3.905
##  Residual                   715.52   26.749
## Number of obs: 2592, groups:  participant_id, 144
##
## Fixed effects:
##                Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)      51.836      1.049 141.000  49.423   <2e-16 ***
## generatorbingo   -2.011      1.483 141.000  -1.356   0.1773
## generatorstock    2.563      1.533 141.000   1.672   0.0967 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) gnrtrb
## generatrbng -0.707
## genertrstck -0.684  0.484
```

tell the p value using one way mixed anova

```
anova(model1)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
## generator 6378.7  3189.4     2   141  4.4574 0.01327 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
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
```

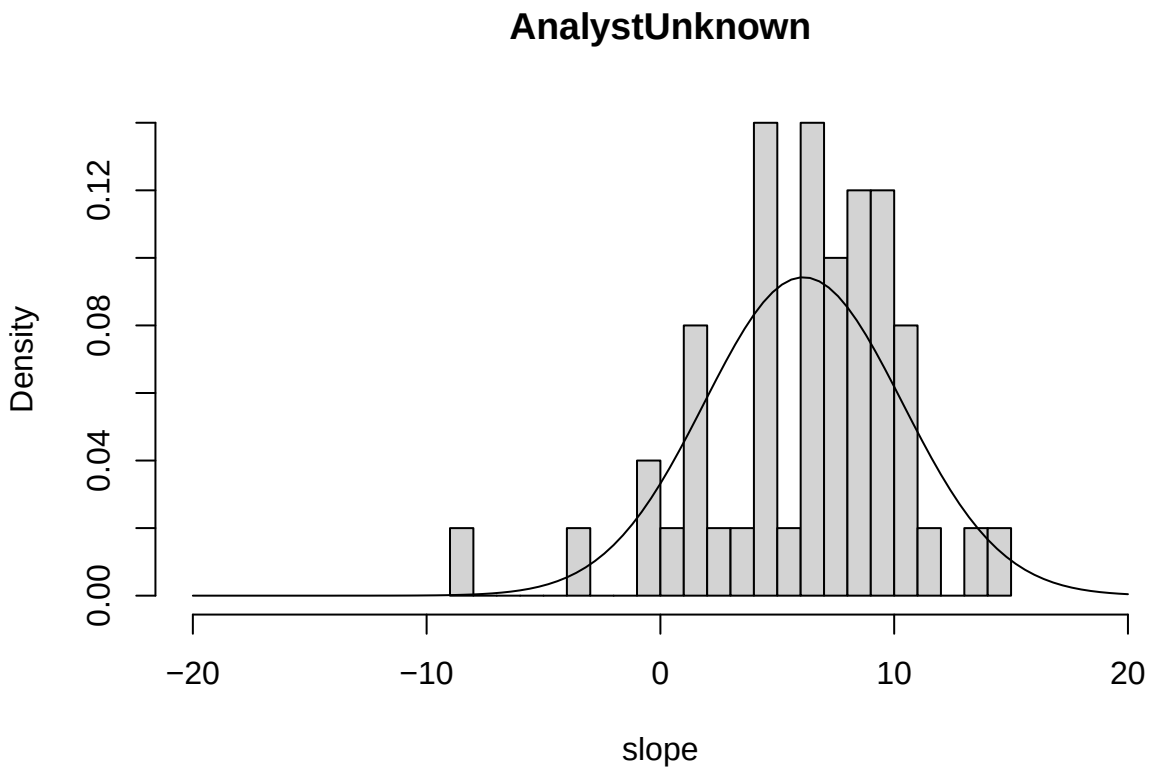
```
## Anova Table (Type 3 tests)
##
## Response: prediction_recode
##                              Effect            df     MSE         F  ges p.value
## 1                         generator        2, 141 1460.31    4.38 * .025    .014
## 2           terminal_streak_length  5.09, 718.00  417.95 72.94 *** .235   <.001
## 3 generator:terminal_streak_length 10.18, 718.00  417.95      0.88 .007    .556
## ---
## 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")
```

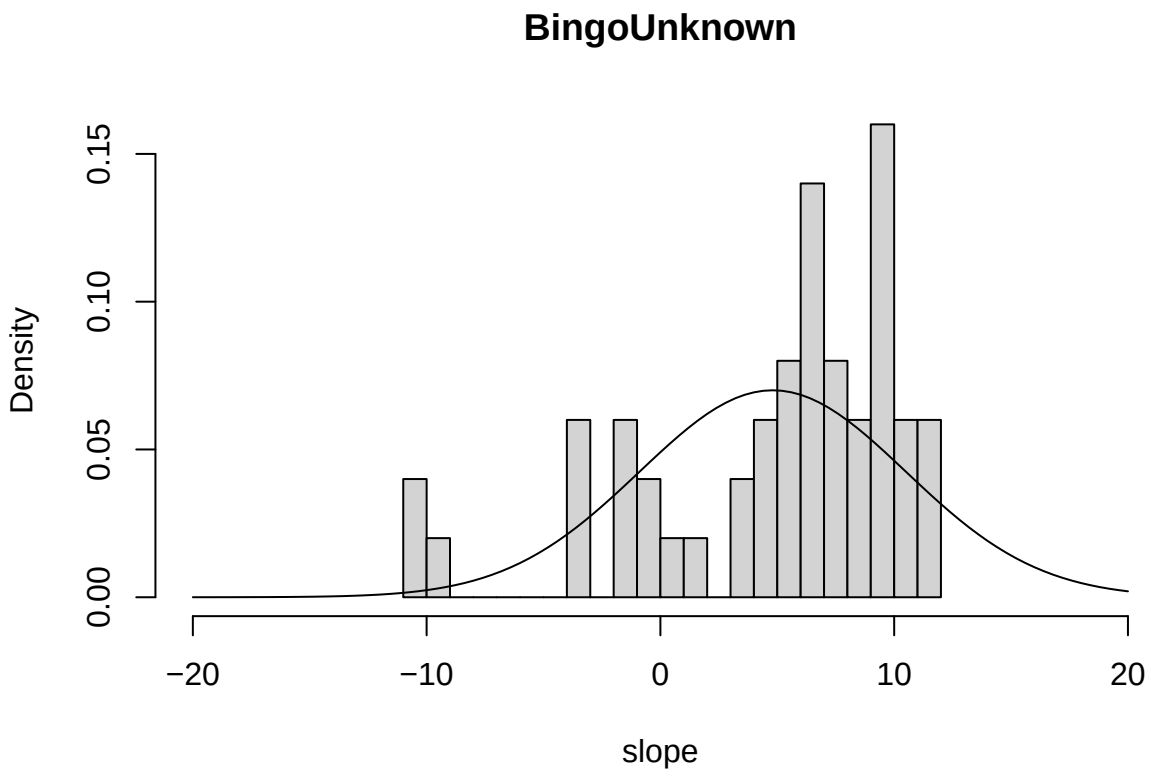
```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  data$prediction_recode and data$generator
##
##       analyst bingo
## bingo 0.3437  -
## stock 0.1550  0.0016
##
## P value adjustment method: bonferroni
```

```
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])
```

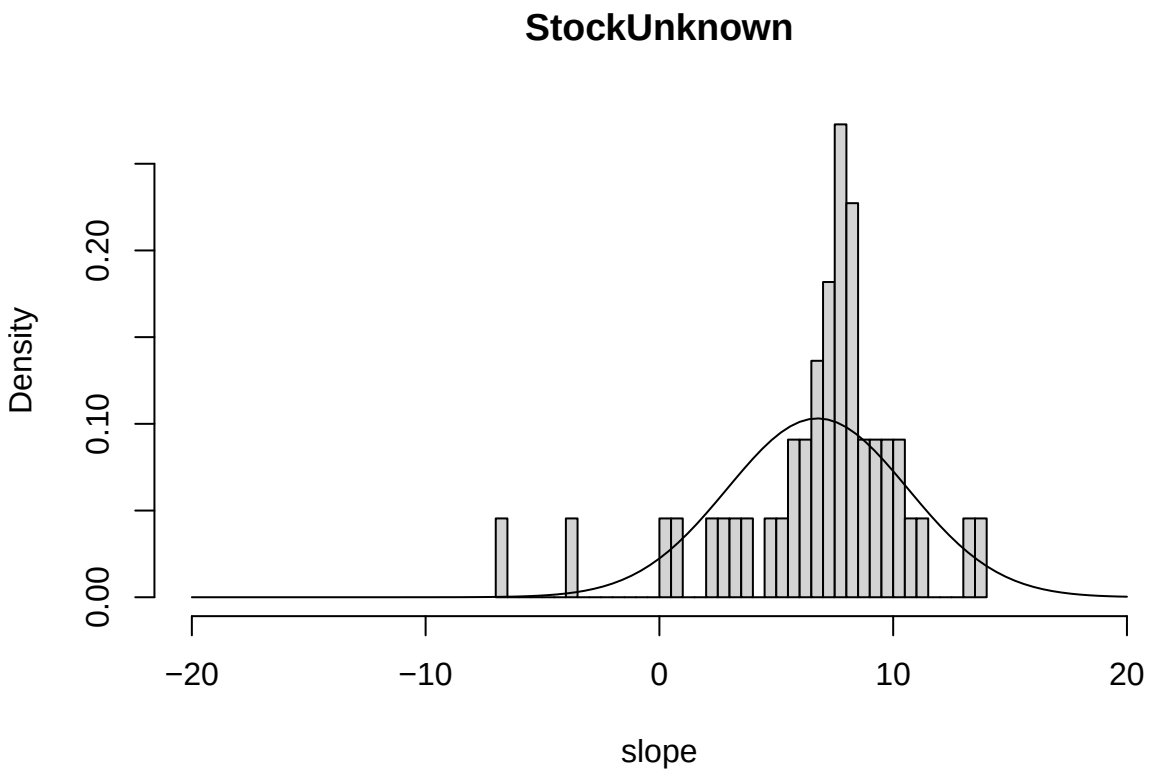
```
}

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")
```


StockUnknown
Density
0.00
0.10
0.20
−20
−10
0
10
20
slope