

```

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 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

nrow(data1)

## [1] 900

nrow(data2)

## [1] 900

nrow(data3)

## [1] 900

calculate the mean of predicotn_recode for each terminwal streak from 1 to 7
print length of data1 predicton recode and list of terminal_streak of data1

```

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

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

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

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

```
## [1] 0
```

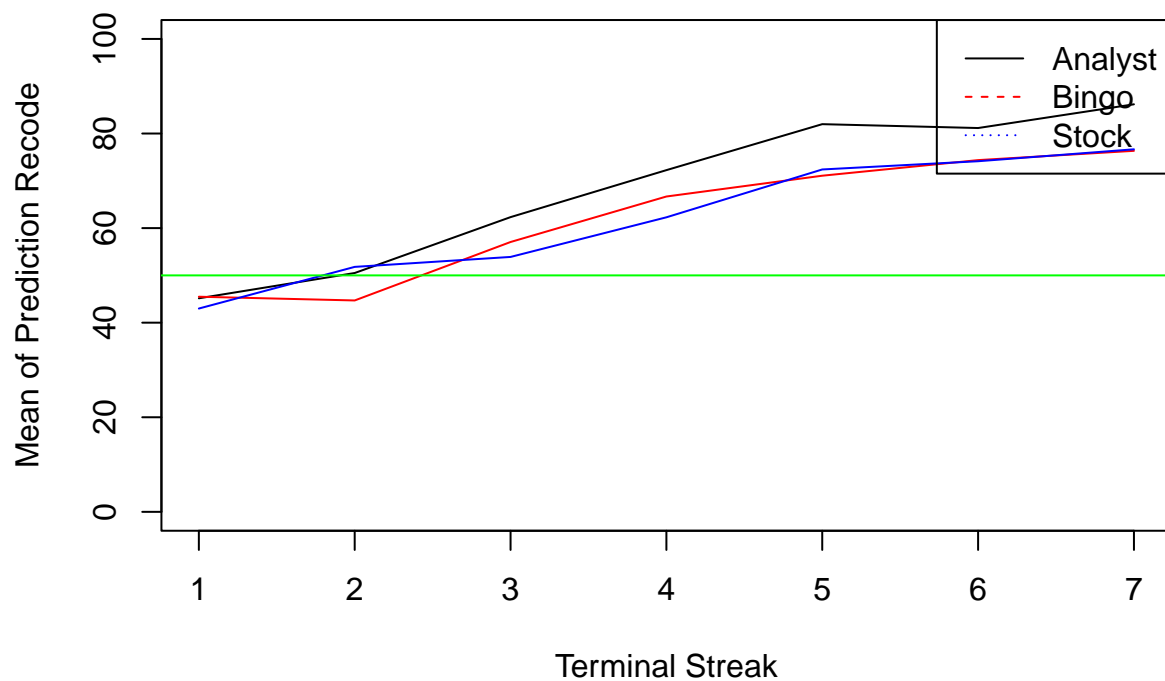
```
mean1 <- aggregate(data1$prediction_recode, by = list(data1$terminal_streak_length), FUN = mean)
```

```
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 Prediction Recode")
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)
```

## Mean of Prediction Recode 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: 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
## generatrbing -0.707
## genertrstck  -0.707  0.500
```

```
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
```

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

```
## 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
```

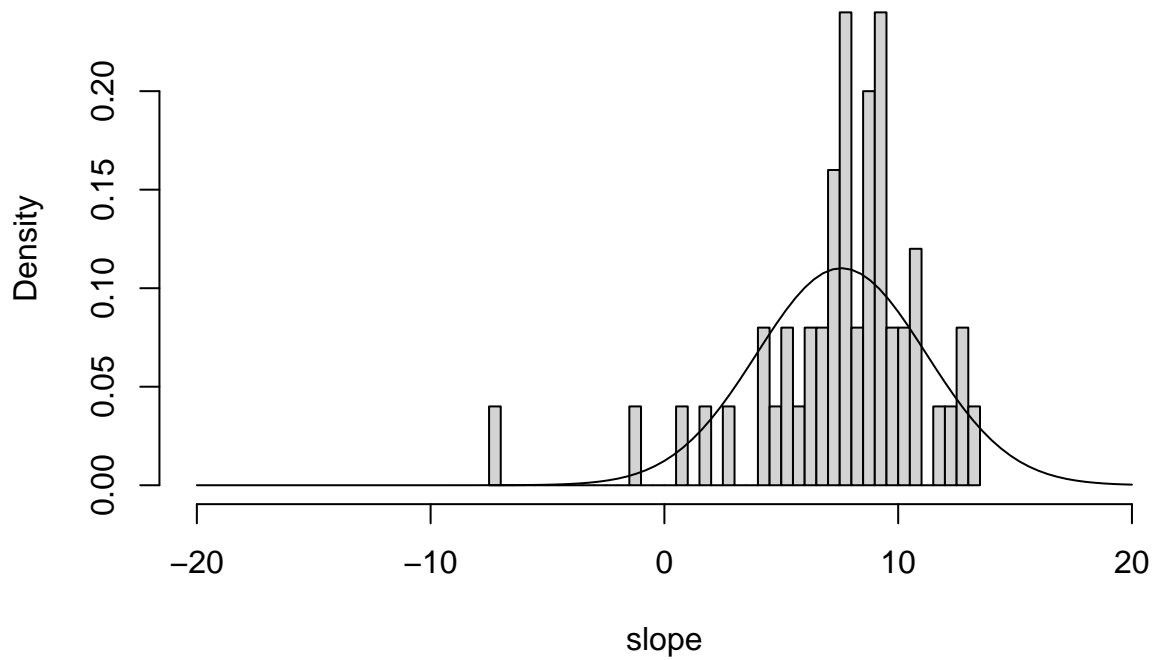
```
id <- unique(data1$participant_id)

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")
```

## AnalystUnknown

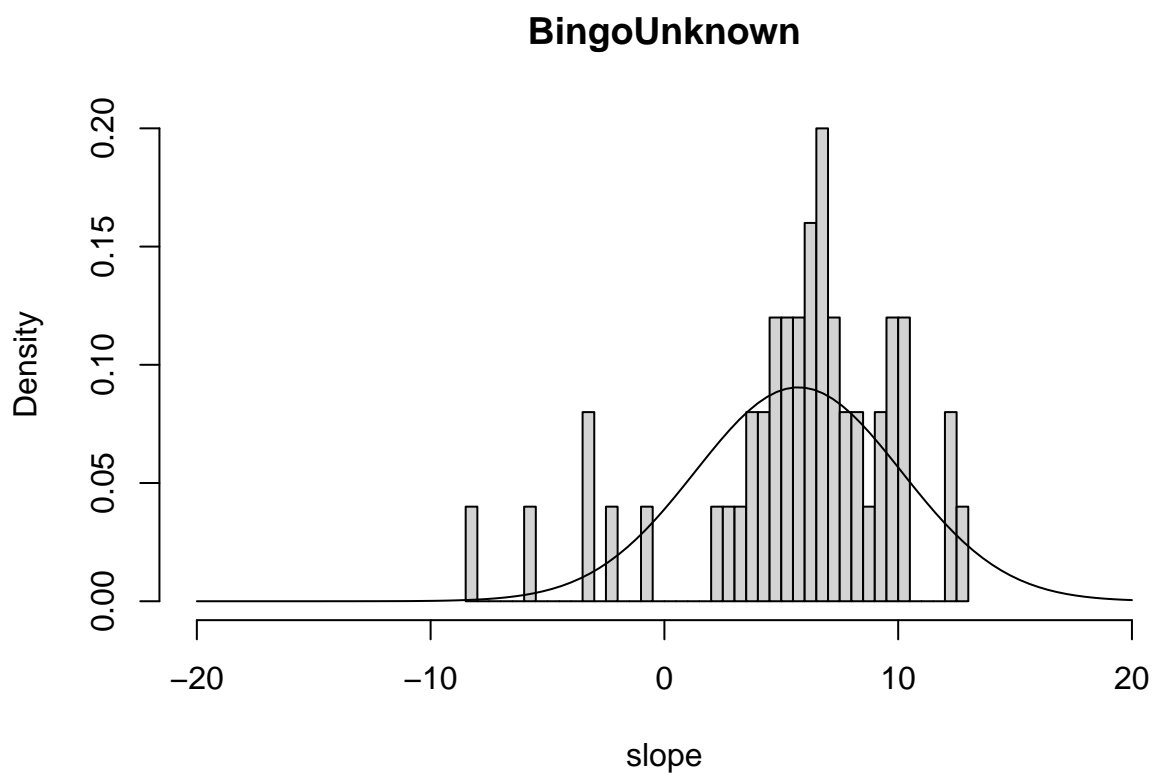


```
id <- unique(data2$participant_id)

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

