

```

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

library(ggplot2)
data <- read_excel("../Data/PredictingOutcomes_ParticipantPredictions.xlsx", sheet = "Study 2B")

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

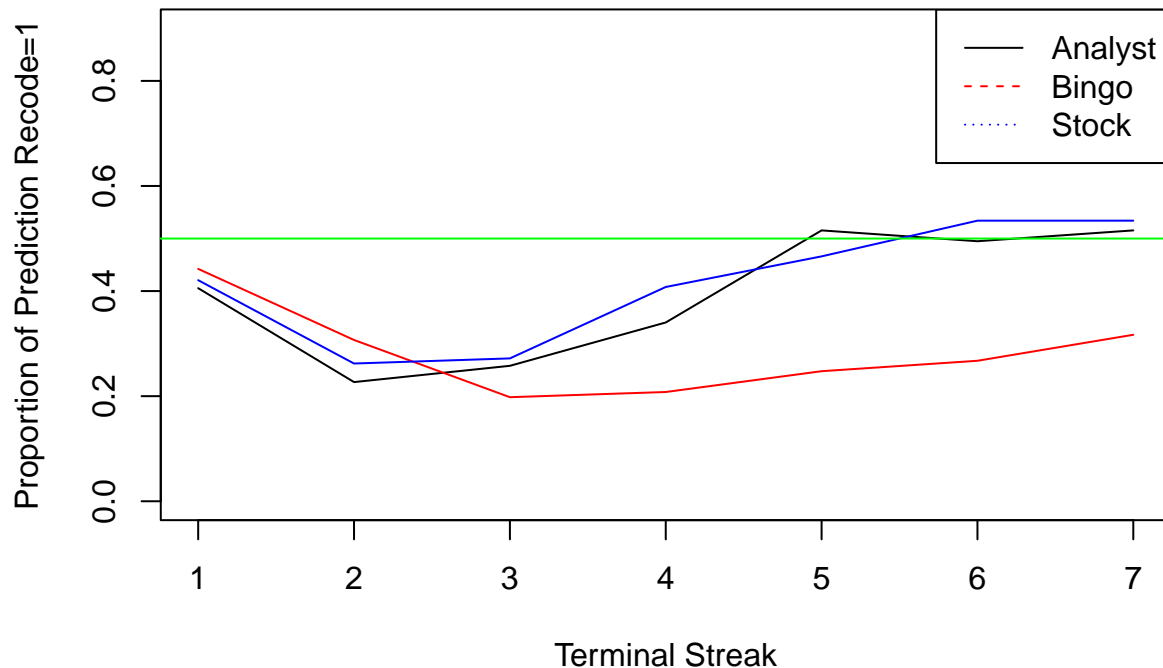
calculate the proportion of participants who predicted the prediction_recode=1 for each terminal_streak_length from 1 to 7

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

```
modell1 <- lmer(prediction_recode ~ generator + (1|participant_id), data = data)
summary(modell1)
```

```
## 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: 7625.4
##
## Scaled residuals:
## Min 1Q Median 3Q Max
## -1.1610 -0.8025 -0.6954 1.1671 1.4900
##
## Random effects:
## Groups Name Variance Std.Dev.
## participant_id (Intercept) 0.007499 0.08659
## Residual 0.232460 0.48214
## Number of obs: 5418, groups: participant_id, 301
##
## Fixed effects:
## Estimate Std. Error df t value Pr(>|t|)
## (Intercept) 0.40092 0.01451 298.00004 27.637 <2e-16 ***
## generatorbingo -0.02028 0.02031 298.00004 -0.998 0.319
## generatorstock 0.01710 0.02021 298.00004 0.846 0.398
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```

##          (Intr) gnrtrb
## generatrbnb -0.714
## genertrstck -0.718  0.513
anova(model1)

## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## generator 0.81278 0.40639      2   298  1.7482 0.1759
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, 298 0.52      6.68 ** .017   .001
## 2 terminal_streak_length 5.17, 1541.14 0.16 16.90 *** .033  <.001
## 3 generator:terminal_streak_length 10.34, 1541.14 0.16  4.40 *** .018  <.001
## ---
## 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")

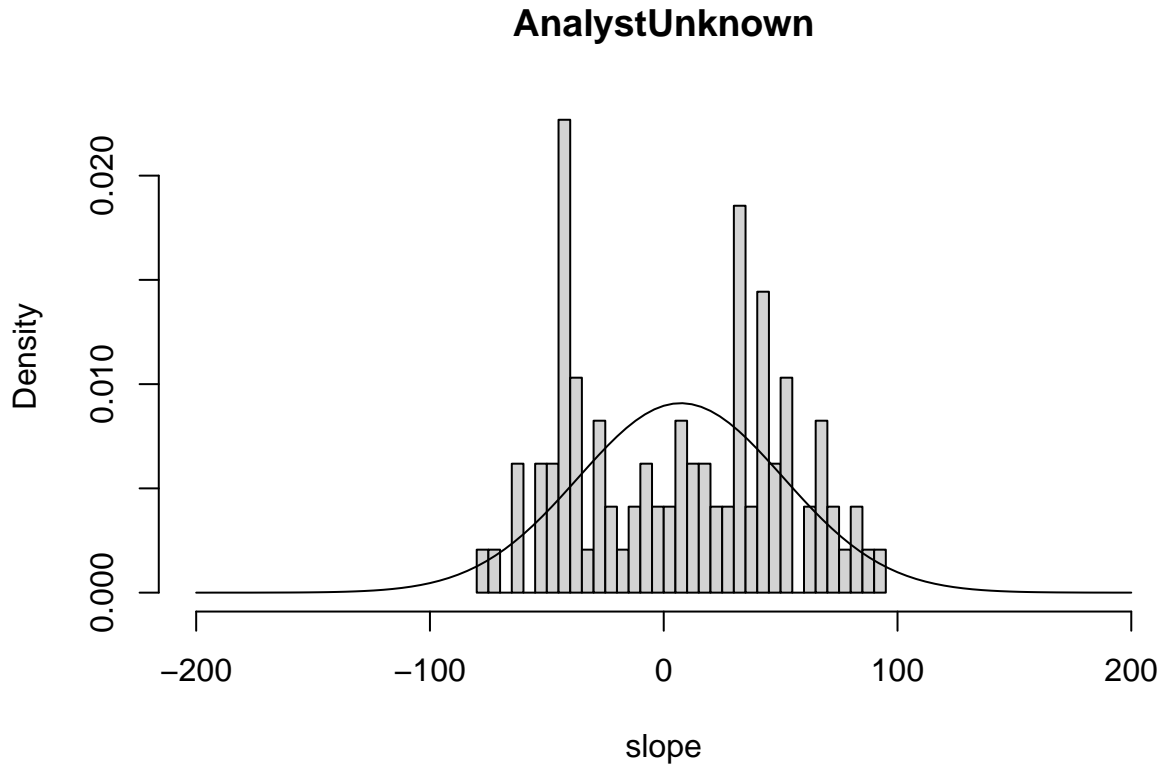
##
## Pairwise comparisons using t tests with pooled SD
##
## data:  data$prediction_recode and data$generator
##
##          analyst bingo
## bingo 0.650      -
## stock 0.886    0.062
##
## P value adjustment method: bonferroni
id <- unique(data1$participant_id)

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")
```



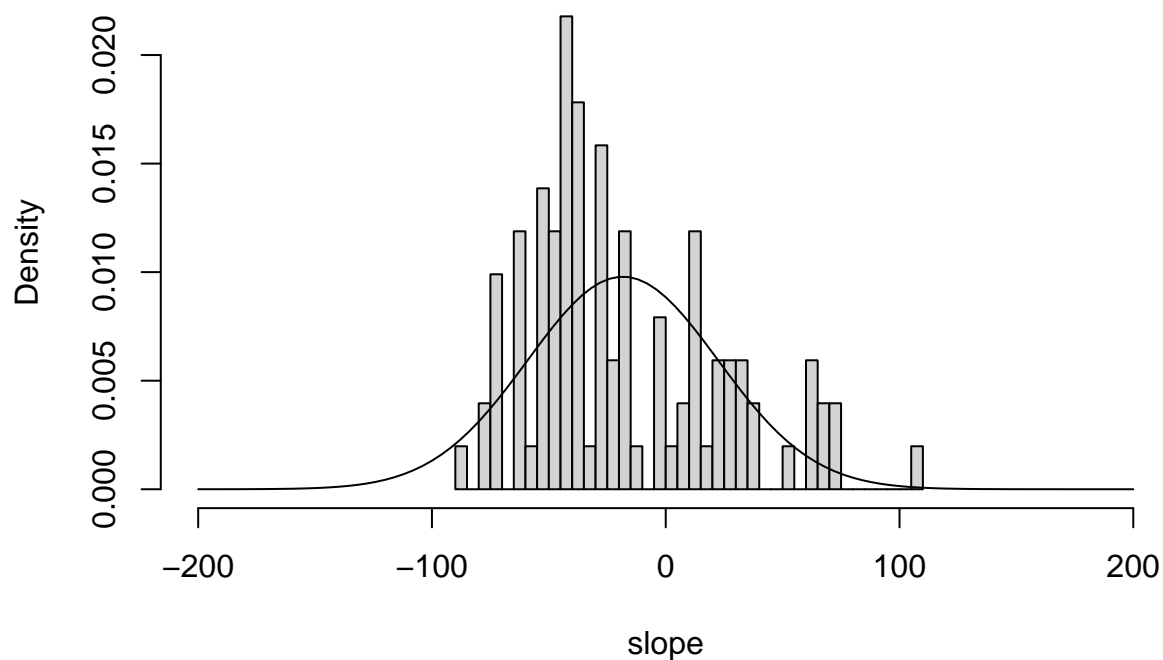
```
id <- unique(data2$participant_id)

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")
```

## BingoUnknown

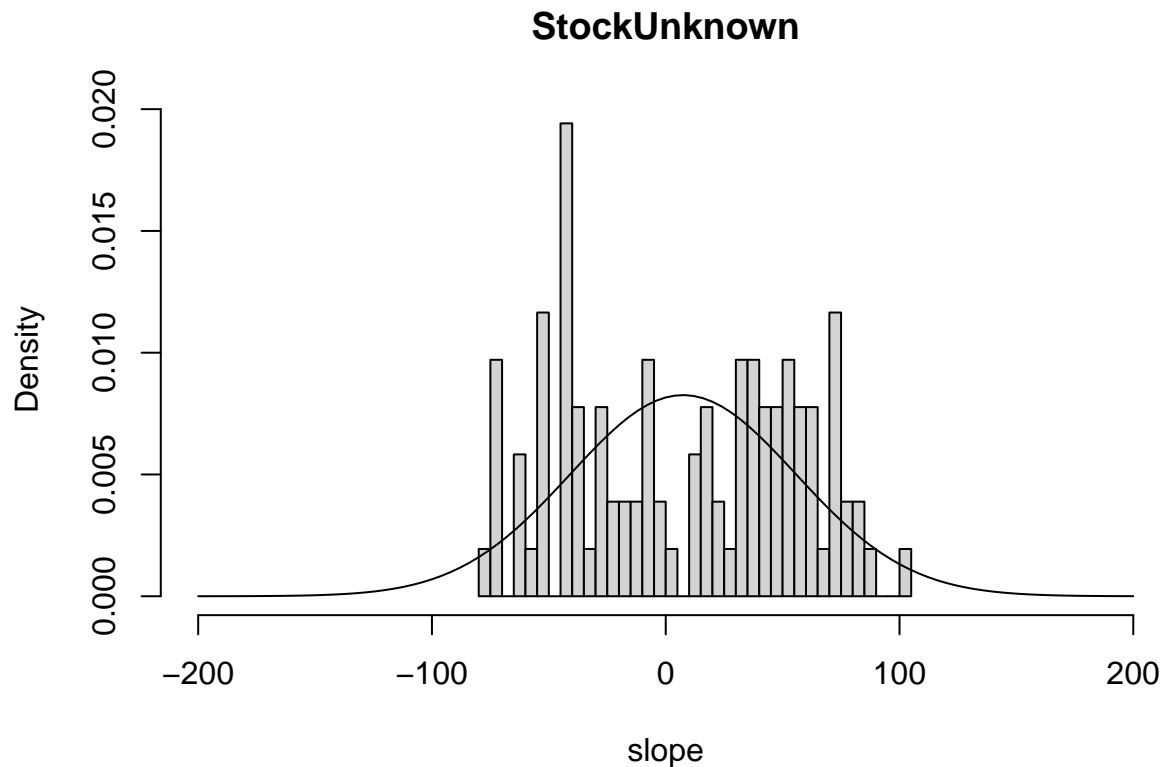


```
id <- unique(data3$participant_id)

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 1716  143  143  143  143  143  143
```

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

```
##
##      0      1
## 0 1682 1126
```

```
##      1 1546 1028
chisq.test(test)

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 0.0086918, df = 1, p-value = 0.9257
df <- df_total[df_total$generator=='analyst',]
test <- table(df$gender, df$prediction_recode)
print(test)

##
##      0      1
## 0 537 327
## 1 509 373
chisq.test(test)

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 3.4046, df = 1, p-value = 0.06502
df <- df_total[df_total$generator=='bingo',]
test <- table(df$gender, df$prediction_recode)
print(test)

##
##      0      1
## 0 571 365
## 1 544 320
chisq.test(test)

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
## X-squared = 0.65046, df = 1, p-value = 0.4199
df <- df_total[df_total$generator=='stock',]
test <- table(df$gender, df$prediction_recode)
print(test)

##
##      0      1
## 0 574 434
## 1 493 335
chisq.test(test)

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  test
```

```
## X-squared = 1.1548, df = 1, p-value = 0.2826
```

**Correlation testing and box-plot** form a table with mean of prediction recode on the basis of participant\_id and terminal\_streak\_length

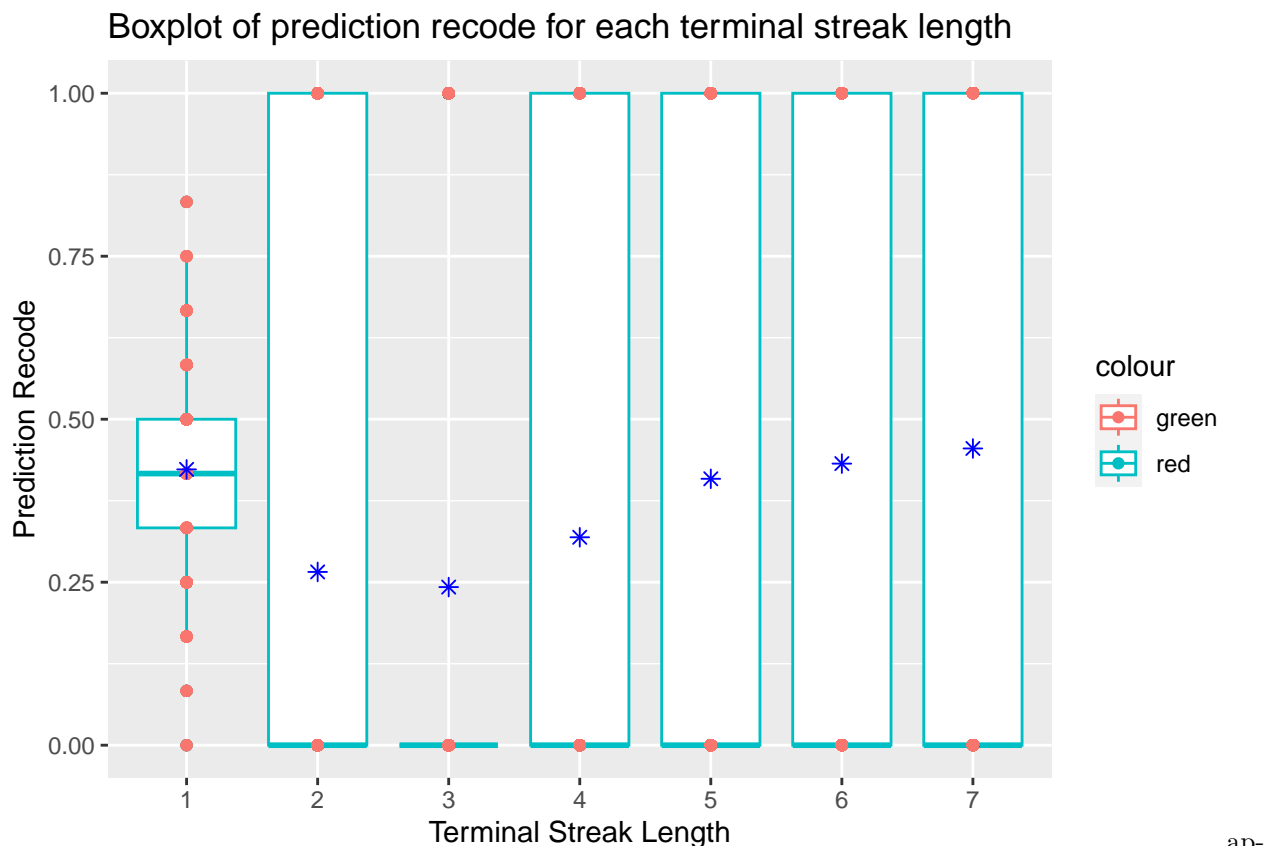
```
cdata <- aggregate(prediction_recode~participant_id+terminal_streak_length,data ,FUN = mean)
cdata1 <- aggregate(data1$prediction_recode, by = list(data1$participant_id, data1$terminal_streak_length), FUN = mean)
cdata2 <- aggregate(data2$prediction_recode, by = list(data2$participant_id, data2$terminal_streak_length), FUN = mean)
cdata3 <- aggregate(data3$prediction_recode, by = list(data3$participant_id, data3$terminal_streak_length), FUN = mean)
colnames(cdata) <- c( "participant_id","terminal_streak_length","prediction_recode")
colnames(cdata1) <- c( "participant_id","terminal_streak_length","prediction_recode")
colnames(cdata2) <- c( "participant_id","terminal_streak_length","prediction_recode")
colnames(cdata3) <- c( "participant_id","terminal_streak_length","prediction_recode")
nrow(cdata)
```

```
## [1] 2107
```

```
# cdata
```

draw box plot using ggplot2

```
ggplot(cdata, aes(x = as.factor(terminal_streak_length), y = prediction_recode)) +geom_boxplot(aes(colour = terminal_streak_length, size = 2, color = "blue")) + ggtitle("Boxplot of prediction recode for each terminal streak length")
```



ply pearson correlation test on the data

```
cor.test(cdata$terminal_streak_length, cdata$prediction_recode, method = "pearson")
```

```
##
## Pearson's product-moment correlation
##
```



```

## data:  cdata$terminal_streak_length and cdata$prediction_recode
## t = 4.3788, df = 2105, p-value = 1.252e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.05251698 0.13715398
## sample estimates:
##      cor
## 0.09500717

cor.test(cdata1$terminal_streak_length, cdata1$prediction_recode, method = "pearson")

##
## Pearson's product-moment correlation
##
## data:  cdata1$terminal_streak_length and cdata1$prediction_recode
## t = 4.6584, df = 677, p-value = 3.837e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1023517 0.2481851
## sample estimates:
##      cor
## 0.1762353

cor.test(cdata2$terminal_streak_length, cdata2$prediction_recode, method = "pearson")

##
## Pearson's product-moment correlation
##
## data:  cdata2$terminal_streak_length and cdata2$prediction_recode
## t = -1.8634, df = 705, p-value = 0.06282
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.143004617 0.003746241
## sample estimates:
##      cor
## -0.07000795

cor.test(cdata3$terminal_streak_length, cdata3$prediction_recode, method = "pearson")

##
## Pearson's product-moment correlation
##
## data:  cdata3$terminal_streak_length and cdata3$prediction_recode
## t = 4.558, df = 719, p-value = 6.068e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.09573699 0.23768732
## sample estimates:
##      cor
## 0.1675806

```