Math 23C Final Project

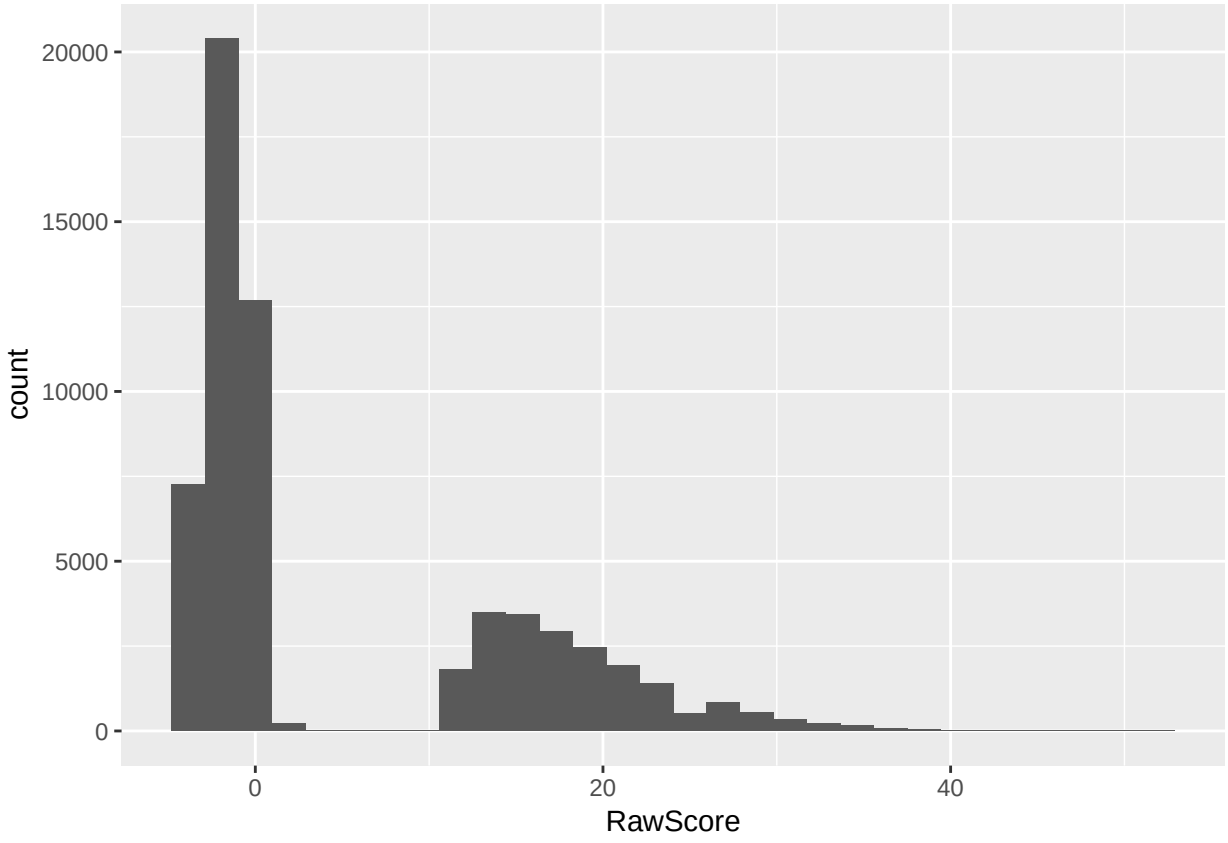
Shuvom Sadhuka and Lavanya Singh

5/14/2019

The purpose of this analysis was to explore racial biases in COMPAS, a criminal justice algorithm used to aid judges in deciding sentencing. A subset of the full dataset was published by ProPublica, and we used this dataset to conduct our analyses. We first load, k-means cluster, and visualize the dataset.

```
compas <- read.csv ("compass/compas-scores-raw.csv"); #View(compas)
library(ggplot2)
library(gmodels)
library(stats4)

#messy but there are two clear clusters we can isolate
ggplot(compas, aes(x=RawScore)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

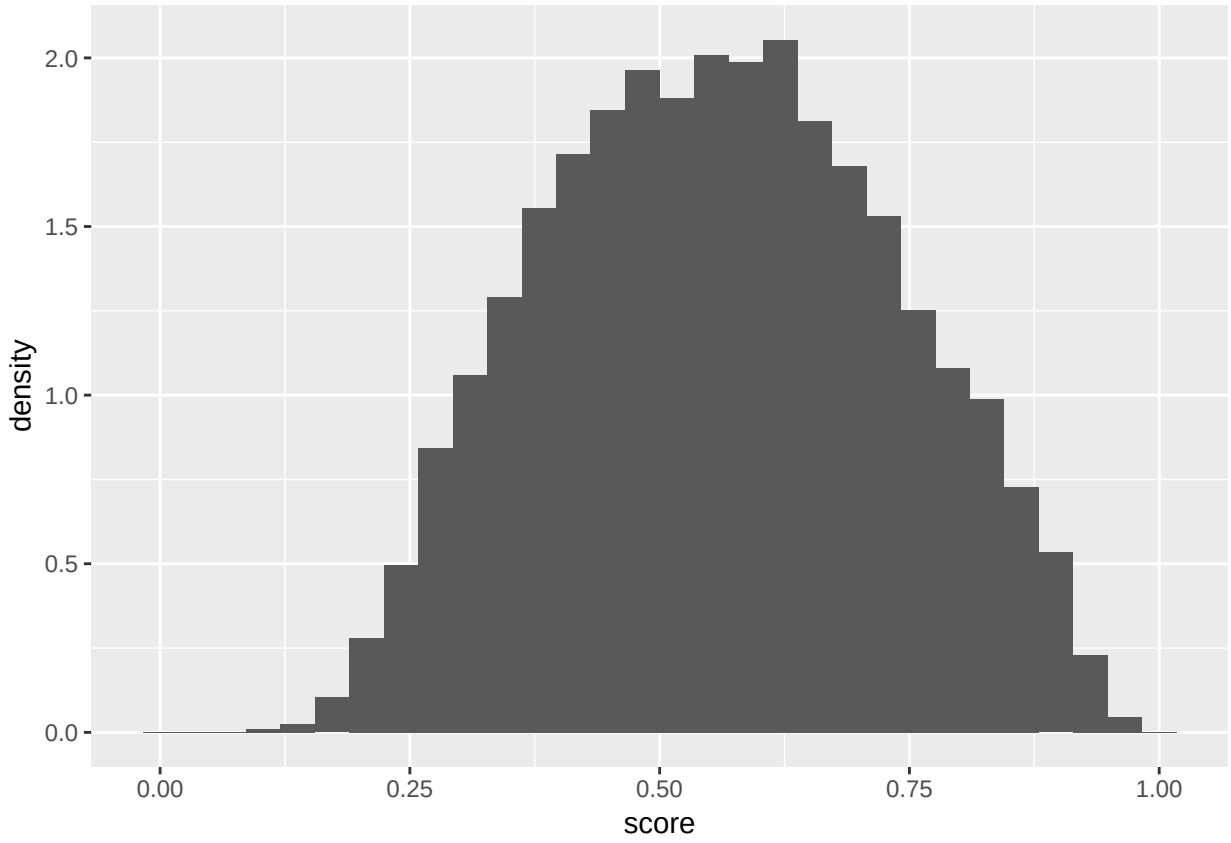



```
####POINT 11: ggplot
#let's use R's built-in k-means cluster function
set.seed(42)
clusters <- kmeans(compas$RawScore, 2); #clusters
#storing the cluster of each point in the dataframe
compas$cluster <- clusters$cluster
```

The first cluster looks pretty bell-shaped, so let's fit a Beta and Normal to it.

```
scores <- compas[which(compas$cluster == 2),];
max <- max(scores$RawScore)
min <- min(scores$RawScore)
scores$score <- (max - scores$RawScore) / (max - min)
plot <- ggplot(scores, aes(score)) + geom_histogram(aes(y = stat(density))); plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
mu <- mean(scores$score); mu
```

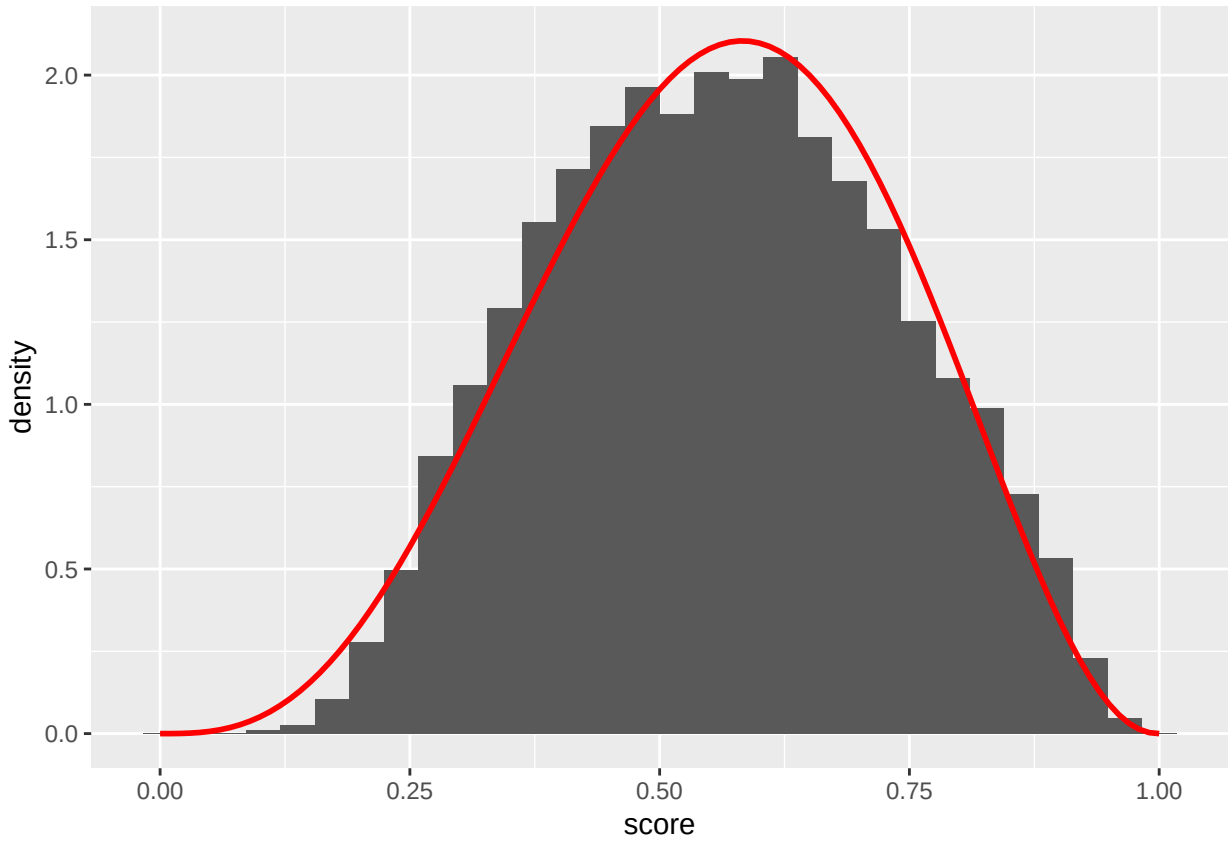
```
## [1] 0.5603551
```

```
var <- var(scores$score); var
```

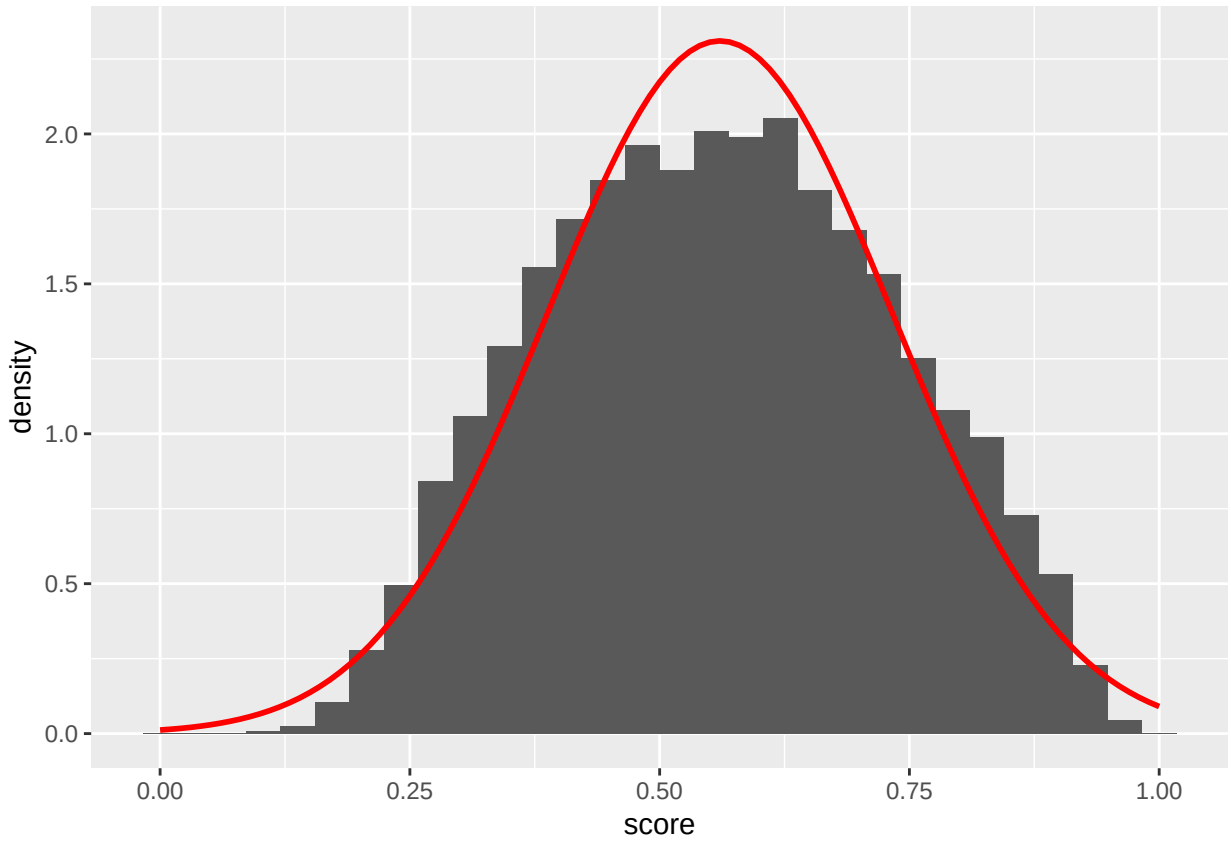
```
## [1] 0.02982039
```

```
param1 <- 15162/3725
param2 <- 11913/3725
stat1 <- stat_function(fun = dbeta, args = list(param1, param2), lwd = 1, col = "red")
stat2 <- stat_function(fun = dnorm, args = list(mu, sqrt(var)), lwd = 1, col = "red")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The plots seem to fit pretty well. In this project we will compare the COMPAS scores of white to non-white people. To do so, we divide the dataset into two parts, white and non-white people. We define a function to compute a permutation test, z-test, t-test, and confidence interval for the differences in risk scores between white and non-white people. Our dataset is structured to show the risk scores for each individual by risk type; that is, the scale of risk scores of violence is different from the scale of risk scores of, for example, failure to appear. We want to subset the dataset for each type of risk and calculate the presence, or lack thereof, of racial bias in each subset.

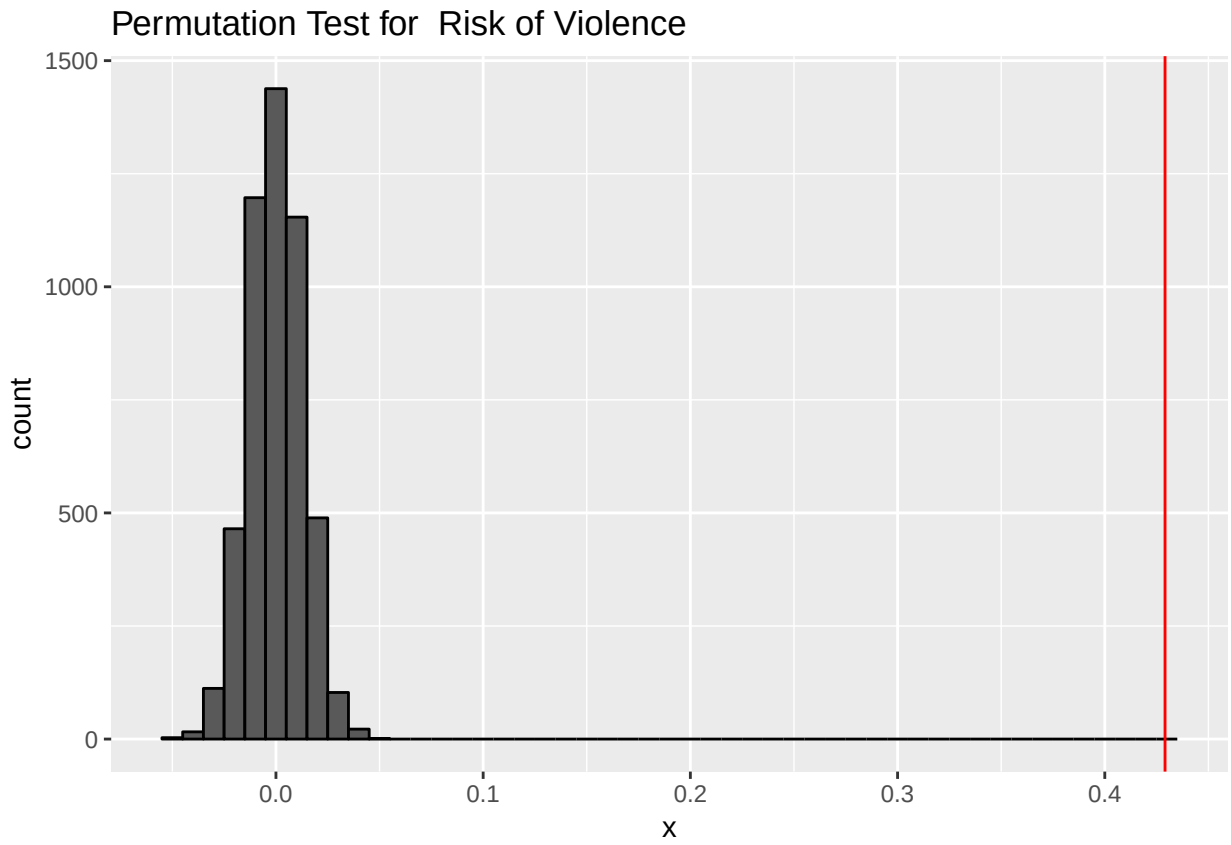
```
type_of_risk <- function(x, y){
  library(ggplot2)
  stopifnot(y < 2)

  if (y == 0) #y == 0 represents where we have subdivided the population
  {
    type <- compas[which(compas$DisplayText == x),]; #View(type)
    type_white <- type[which(type$Ethnic_Code_Text == 'Caucasian'),]
    type_nonwhite <- type[which(!type$Ethnic_Code_Text == 'Caucasian'),]
  }

  if (y == 1) #y == 1 represents the pooled dataset
  {
    type <- compas
    type_white <- type[which(type$Ethnic_Code_Text == 'Caucasian'),]
    type_nonwhite <- type[which(!type$Ethnic_Code_Text == 'Caucasian'),]
  }


  #get the lengths of the non-white and white data
```

```r
nw = length(type_white$RawScore); nw
nn = length(type_nonwhite$RawScore); nn
total = nw+nn
length(type$RawScore)

#here's our first statistical test: a permutation test!
####REQUIRED: permutation test
#we want to do a 2-sample permutation test
#first we compute the actual difference in means
actual = mean(type_nonwhite$RawScore) - mean(type_white$RawScore)
diffs <- vector()

for (i in 1:5000){
  all_scores <- c(type_white$RawScore, type_nonwhite$RawScore); length(all_scores)
  #sample nw incidies for the white mean
  sampled_indicies <- sample(1:total, nw, replace = FALSE); sampled_indicies

  sampled_white <- all_scores[sampled_indicies]
  mean(sampled_white)

  #the rest of the individuals will be in the nonwhite sample
  sampled_nonwhite<- all_scores[-sampled_indicies];
  mean(sampled_nonwhite)
  diff <- mean(sampled_white) - mean(sampled_nonwhite)

  diffs <- c(diffs, diff)
}

diffs_df <- data.frame(x = diffs)


print(ggplot(diffs_df, aes(x=x)) +
  geom_histogram(binwidth=0.01, colour = "black")
  + geom_vline(xintercept = actual, colour = "red")
  + labs(title = paste("Permutation Test for ", x)))

#to calculate the p-value, we use the empirical cdf
percentile <- ecdf(diffs)
print(paste('The permutation test p-value is', (1 - percentile(0.2689452))*2))

type_white$id <- 'white'
type_nonwhite$id <- 'non-white'

Lengths <- data.frame(rbind(type_nonwhite, type_white))
print(ggplot(Lengths, aes(RawScore, fill = id)) +
  geom_histogram(alpha = 0.5, aes(y = ..density..), position = 'identity')
  + labs(title = paste("Raw Scores for ", x)))


##t-test
print(t.test(type_white$RawScore, type_nonwhite$RawScore))

#lets recreate the t-test manually and then check against the R package (above)
```

```
  nonwhite_mean <- mean(type_nonwhite$RawScore)
  white_mean <- mean(type_white$RawScore)
  nonwhite_sd <- sqrt(var(type_nonwhite$RawScore))
  white_sd <- sqrt(var(type_white$RawScore))

  #this is the standard deviation for a two-sample t-test with unequal populations and variances
  std <- sqrt(((nw - 1)*white_sd^2 +
                 (nn - 1)*nonwhite_sd^2)/(nw + nn - 2))*(sqrt(1/nw + 1/nn)); std

  #this is the t-stat
  t_stat <- (nonwhite_mean - white_mean)/std; print(t_stat)

  #the degrees of freedom
  deg.freedom = nw + nn - 2
  print(paste("The two-sided t-test p-value is",
              (pt(t_stat, deg.freedom, lower.tail = FALSE, log.p = FALSE))))

  diff_in_means = nonwhite_mean - white_mean; diff_in_means
  std_norm = sqrt(nonwhite_sd^2/nn + white_sd^2/nw)
  z_stat = diff_in_means/std_norm; z_stat
  print(paste("The p-value for the z-test is",
              pnorm(z_stat, 0, 1, lower.tail = FALSE)))

  theta = diff_in_means
  sigma_sq = var(all_scores); sigma_sq
  conf_lower = theta - 1.96 * (sqrt(sigma_sq))/(sqrt(total));
  conf_upper = theta + 1.96 * (sqrt(sigma_sq))/(sqrt(total));
  print(paste("The confidence interval for the difference in means of the risk scores is",
              conf_lower, conf_upper))

  t.est <- t.test(type_nonwhite$RawScore, type_white$RawScore, var.equal=FALSE)$stat
  print(paste("The t-stat is", t.est))
  #now let's make a bootstrapped t-test
  means_nonwhite <- vector()
  means_white <- vector()

  #the bootstrap can also cross-check against sensitivity to outliers since
  #outliers have a small chance of being included in any given bootstrapped dataset
  #this counts as another simulation method
  b <- function(){
    A <- sample(type_nonwhite$RawScore, nn, replace=T)
    B <- sample(type_white$RawScore, nw, replace=T)
    stud_test <- t.test(A, B, var.equal=FALSE)
    stud_test

    return(stud_test$stat)
  }
  t.stat.vect = vector(length=10000)
  t.vect <- replicate(10000, b())
  print(paste("The percentile for our t-stat relative to bootstrapped t-stats is",
              1 - mean(t.est>t.vect)))
}
```

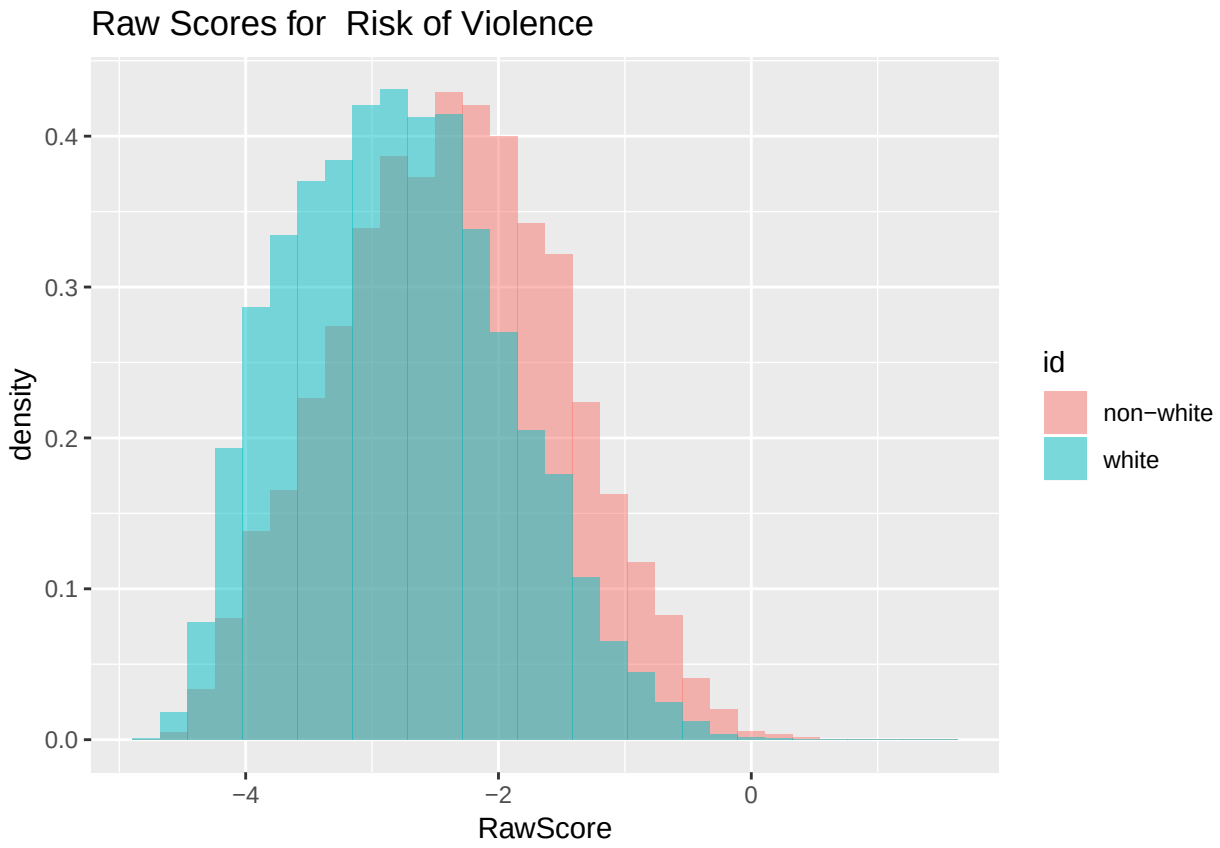
Then, we can look at the results for the risk of violence, for example:

```
type_of_risk('Risk of Violence', 0)
```




```
## [1] "The permutation test p-value is 0"
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
##
##  Welch Two Sample t-test
##
## data:  type_white$RawScore and type_nonwhite$RawScore
## t = -34.24, df = 15658, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4536341 -0.4045092
## sample estimates:
## mean of x mean of y
## -2.797941 -2.368869
##
## [1] 33.75323
## [1] "The two-sided t-test p-value is 2.46598945538721e-243"
## [1] "The p-value for the z-test is 3.02503538613405e-257"
## [1] "The confidence interval for the difference in means of the risk scores is 0.416796008872252 0.4
## [1] "The t-stat is 34.2404511861265"
## [1] "The percentile for our t-stat relative to bootstrapped t-stats is 0.4963"
```

We can also do further analyses, like chi-square tests, contingency tables, and covariances.

```
violence <- compas[which(compas$DisplayText == 'Risk of Violence'),]
violence_white <- compas[which(compas$Ethnic_Code_Text == 'Caucasian'),]

#are race and a display text of "Risk of Violence" independent?
#extracting logical columns
risk_Log <- compas$DisplayText == "Risk of Violence"; sum(risk_Log)
```

```
## [1] 20281
```

```
race_Log <- compas$Ethnic_Code_Text != "Caucasian"; sum(race_Log)
```

```
## [1] 39060
```

```
dataLog <- data.frame(risk_Log, race_Log)
#a contingency table showing all 4 options
Obs <- table (dataLog$risk_Log, dataLog$race_Log); Obs
```

```
##
##         FALSE  TRUE
##   FALSE 14522 26040
##   TRUE   7261 13020
```

```
#what we would expect if the factors are independent
Expected <- outer(rowSums(Obs), colSums(Obs))/sum(Obs); Expected
```

```
##       FALSE  TRUE
## FALSE 14522 26040
## TRUE   7261 13020
```

```
#REQUIRED: analysis of contingency tables
#WOAH these are the exact same
#chi-sq: p value is 1 meaning that there is 100% chance that race and risk of violence
#as display text are independent
chisq.test(dataLog$risk_Log, dataLog$race_Log)
```

```
##
##  Pearson's Chi-squared test
##
## data:  dataLog$risk_Log and dataLog$race_Log
## X-squared = 0, df = 1, p-value = 1
```

```
#Paul's method of calculating chi-sq value
ChiSq <-function(Obs,Exp){
  sum((Obs-Exp)^2/Exp)
}
#same chi-sq statistic and p value as above
CSq <- ChiSq(Obs, Expected); CSq
```

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

```
pchisq(CSq, df = 3, lower.tail = FALSE)
```

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

```
#are race and a display text of "Risk of Recidivism" independent?
#extracting logical columns
recid_Log <- compas$DisplayText == "Risk of Recidivism"; sum(recid_Log)
```

```
## [1] 20281
```

```
dataLog$recid_Log <- recid_Log
#a contingency table showing all 4 options
Obs <- table (dataLog$recid_Log, dataLog$race_Log); Obs
```

```
##
##         FALSE  TRUE
##   FALSE 14522 26040
```

```
##   TRUE   7261 13020
```

```
#what we would expect if the factors are independent
Expected <- outer(rowSums(Obs), colSums(Obs))/sum(Obs); Expected
```

```
##       FALSE  TRUE
## FALSE 14522 26040
## TRUE   7261 13020
```

```
#WOAH these are the exact same
#chi-sq: p value is 1 meaning that there is 100% chance that race and risk of violence
#as display text are independent
chisq.test(dataLog$recid_Log, dataLog$race_Log)
```

```
##
##  Pearson's Chi-squared test
##
## data:  dataLog$recid_Log and dataLog$race_Log
## X-squared = 0, df = 1, p-value = 1
```

```
#Paul's method of calculating chi-sq value
ChiSq <-function(Obs,Exp){
  sum((Obs-Exp)^2/Exp)
}
#same chi-sq statistic and p value as above
CSq <- ChiSq(Obs, Expected); CSq
```

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

```
pchisq(CSq, df = 3, lower.tail = FALSE)
```

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

```
#are race and a display text of "Risk of Failure to Appear" independent?
#extracting logical columns
appear_Log <- compas$DisplayText == "Risk of Recidivism"; sum(appear_Log)
```

```
## [1] 20281
```

```
dataLog$appear_Log <- appear_Log
#a contingency table showing all 4 options
Obs <- table (dataLog$appear_Log, dataLog$race_Log); Obs
```

```
##
##         FALSE  TRUE
##   FALSE 14522 26040
##   TRUE   7261 13020
```

```
#what we would expect if the factors are independent
Expected <- outer(rowSums(Obs), colSums(Obs))/sum(Obs); Expected
```

```
##       FALSE  TRUE
## FALSE 14522 26040
## TRUE   7261 13020
```

```
#WOAH these are the exact same
#chi-sq: p value is 1 meaning that there is 100% chance that race and risk of violence
#as display text are independent
chisq.test(dataLog$appear_Log, dataLog$race_Log)
```

```
##
##  Pearson's Chi-squared test
##
## data:  dataLog$appear_Log and dataLog$race_Log
## X-squared = 0, df = 1, p-value = 1
```

```
#Paul's method of calculating chi-sq value
ChiSq <-function(Obs,Exp){
  sum((Obs-Exp)^2/Exp)
}
#same chi-sq statistic and p value as above
CSq <- ChiSq(Obs, Expected); CSq
```

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

```
pchisq(CSq, df = 3, lower.tail = FALSE)
```

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

Our chi-square test is very fishy; we'd not expect complete indpendence between race and risk text, but perhaps this is a consequence of the fact that our dataset was manually constructed by ProPublica. Let's look at the contingency tables for each of the risk texts:

```
#our three contingency tables displayed nicely
#Risk of Violence versus Race (Caucasian or not)
CrossTable(dataLog$risk_Log, dataLog$race_Log, dnn= c("Caucasian", "Risk of Violence"), prop.t=FALSE, pr
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  60843
##
##
##              | Risk of Violence
##    Caucasian |     FALSE |      TRUE | Row Total |
## -------------|-----------|-----------|-----------|
##        FALSE |     14522 |     26040 |     40562 |
## -------------|-----------|-----------|-----------|
##         TRUE |      7261 |     13020 |     20281 |
## -------------|-----------|-----------|-----------|
## Column Total |     21783 |     39060 |     60843 |
## -------------|-----------|-----------|-----------|
##
##
```

```
#Risk of Recidivism versus Race (Caucasian or not)
CrossTable(dataLog$recid_Log, dataLog$race_Log, dnn= c("Caucasian", "Risk of Recidivism"), prop.t=FALSE
```

```
##
##
##    Cell Contents
## |-------------------------|
```

```
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  60843
##
##
##              | Risk of Recidivism
##    Caucasian |     FALSE |      TRUE | Row Total |
## -------------|-----------|-----------|-----------|
##        FALSE |     14522 |     26040 |     40562 |
## -------------|-----------|-----------|-----------|
##         TRUE |      7261 |     13020 |     20281 |
## -------------|-----------|-----------|-----------|
## Column Total |     21783 |     39060 |     60843 |
## -------------|-----------|-----------|-----------|
##
##
```

```
#Risk of Failure to Appear versus Race (Caucasian or not)
CrossTable(dataLog$appear_Log, dataLog$race_Log, dnn= c("Caucasian", "Risk of Failure to Appear"), prop
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |-------------------------|
##
##
## Total Observations in Table:  60843
##
##
##              | Risk of Failure to Appear
##    Caucasian |     FALSE |      TRUE | Row Total |
## -------------|-----------|-----------|-----------|
##        FALSE |     14522 |     26040 |     40562 |
## -------------|-----------|-----------|-----------|
##         TRUE |      7261 |     13020 |     20281 |
## -------------|-----------|-----------|-----------|
## Column Total |     21783 |     39060 |     60843 |
## -------------|-----------|-----------|-----------|
##
##
```

We could have also computed the covariances to show independence (note that correlation/covariance are necessary but insufficient to show independence):

```
#another way of showing race and display text are independent -
#covariance and correlation are close to 0
#race and risk of violence
cov(dataLog$race_Log, dataLog$risk_Log)
```

```
## [1] -2.029248e-21
```

```r
cor(dataLog$race_Log, dataLog$risk_Log)
```

```
## [1] -8.978834e-21
```

```r
#race and risk of recidivism
cov(dataLog$race_Log, dataLog$recid_Log)
```

```
## [1] -2.029248e-21
```

```r
cor(dataLog$race_Log, dataLog$recid_Log)
```

```
## [1] -8.978834e-21
```

```r
#race and risk of failure to appear
cov(dataLog$race_Log, dataLog$appear_Log)
```

```
## [1] -2.029248e-21
```

```r
cor(dataLog$race_Log, dataLog$appear_Log)
```

```
## [1] -8.978834e-21
```

```r
#all are close to 0
```

Lastly, let's look at the probabilities for each group getting each display text; if we can satisfy $P(A \cap B) = P(A)P(B)$:

```r
#probability of a white person getting a display text of "Risk of Failure to Appear"
p1<- length(which(compas$Ethnic_Code_Text == "Caucasian" & compas$DisplayText == "Risk of Failure to Ap
p1
```

```
## [1] 0.1193399
```

```r
#probability of a non-white person getting a display text of "Risk of Failure to Appear"
p2 <- length(which(compas$Ethnic_Code_Text != "Caucasian" & compas$DisplayText == "Risk of Failure to A
p2
```

```
## [1] 0.2139934
```

```r
#probability of a white person getting a display text of "Risk of Violence"
p3 <- length(which(compas$Ethnic_Code_Text == "Caucasian" & compas$DisplayText == "Risk of Violence"))/
p3
```

```
## [1] 0.1193399
```

```r
#probability of a non-white person getting a display text of "Risk of Violence"
p4 <- length(which(compas$Ethnic_Code_Text != "Caucasian" & compas$DisplayText == "Risk of Violence"))/
p4
```

```
## [1] 0.2139934
```

```r
#probability of a white person getting a display text of "Risk of Recidivism"
p5 <- length(which(compas$Ethnic_Code_Text == "Caucasian" & compas$DisplayText == "Risk of Recidivism"))
p5
```

```
## [1] 0.1193399
```

```r
#probability of a non-white person getting a display text of "Risk of Recidivism"
p6 <- length(which(compas$Ethnic_Code_Text != "Caucasian" & compas$DisplayText == "Risk of Recidivism"))
p6
```

```
## [1] 0.2139934
```

```
#in each case, probability of a person of color getting the corresponding display text is higher
#should equal 1
p1+p2+p3+p4+p5+p6
```

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

```
#it does
```