CSCI454_hw1

Jen Johnson

9/15/2017

Part 1 Distribution of Scores

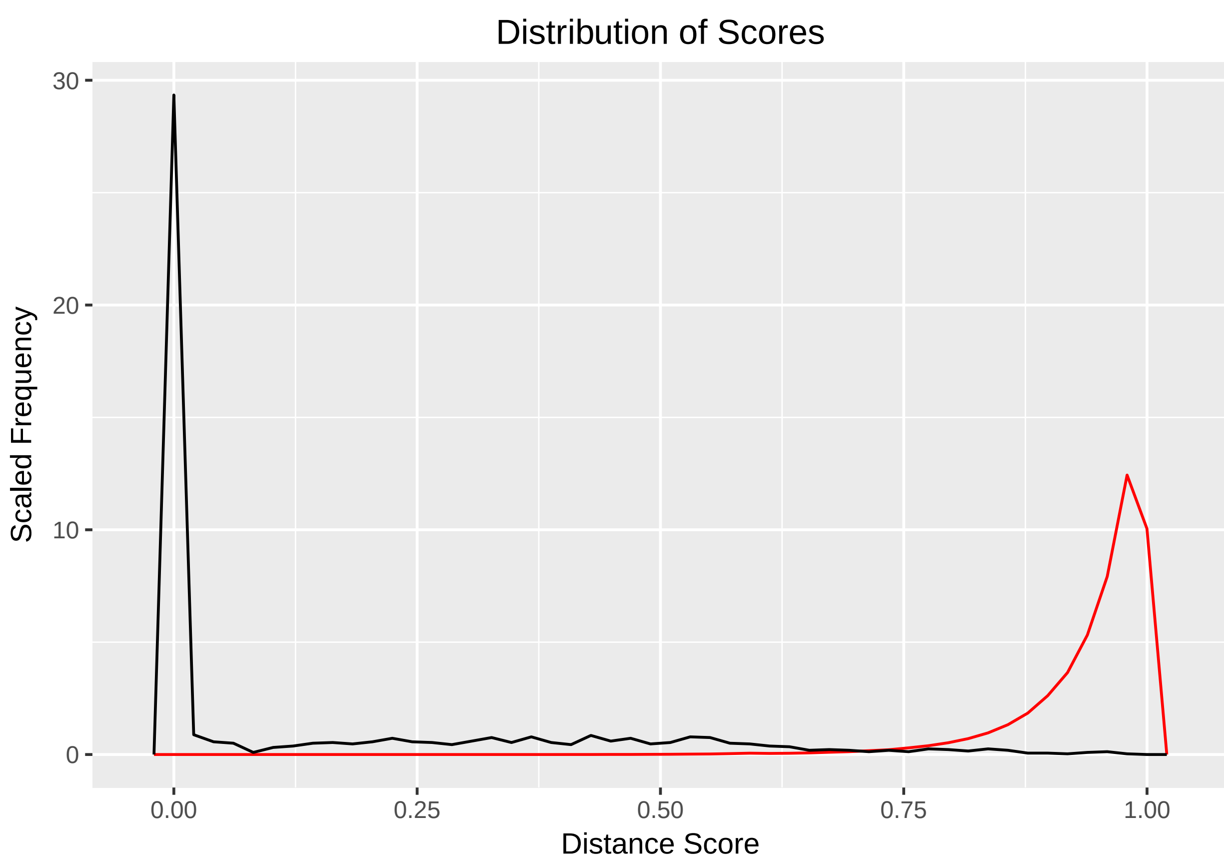
```
data <- read.csv("hw1_list.csv")

#Make new columns with only subject id instead of entire string.
data$seq_1_subject <- substr(data$sequence1, 2, 6)
data$seq_2_subject <- substr(data$sequence2, 2, 6)

#Use new columns to sort data.
imposter_data <- data[!data$seq_1_subject==data$seq_2_subject, ]
genuine_data <- data[data$seq_1_subject==data$seq_2_subject, ]

library(ggplot2)
theme_update(plot.title = element_text(hjust = 0.5))

#Scale using the density function and plot using ggplot's geom_freqpoly.
ggplot() + geom_freqpoly(data = imposter_data, aes(imposter_data$score, ..density..), bins = 50, color =
```



Part 2 d prime

```
numerator <- sqrt(2) * abs(mean(imposter_data$score) - mean(genuine_data$score))
sd_imposter_squared <- sd(imposter_data$score) ^ 2
sd_genuine_squared <- sd(genuine_data$score) ^ 2
denom <- sqrt(sd_imposter_squared + sd_genuine_squared)

dprime <- numerator/denom
print(dprime)
```

```
## [1] 4.489991
```

Part 3A DET

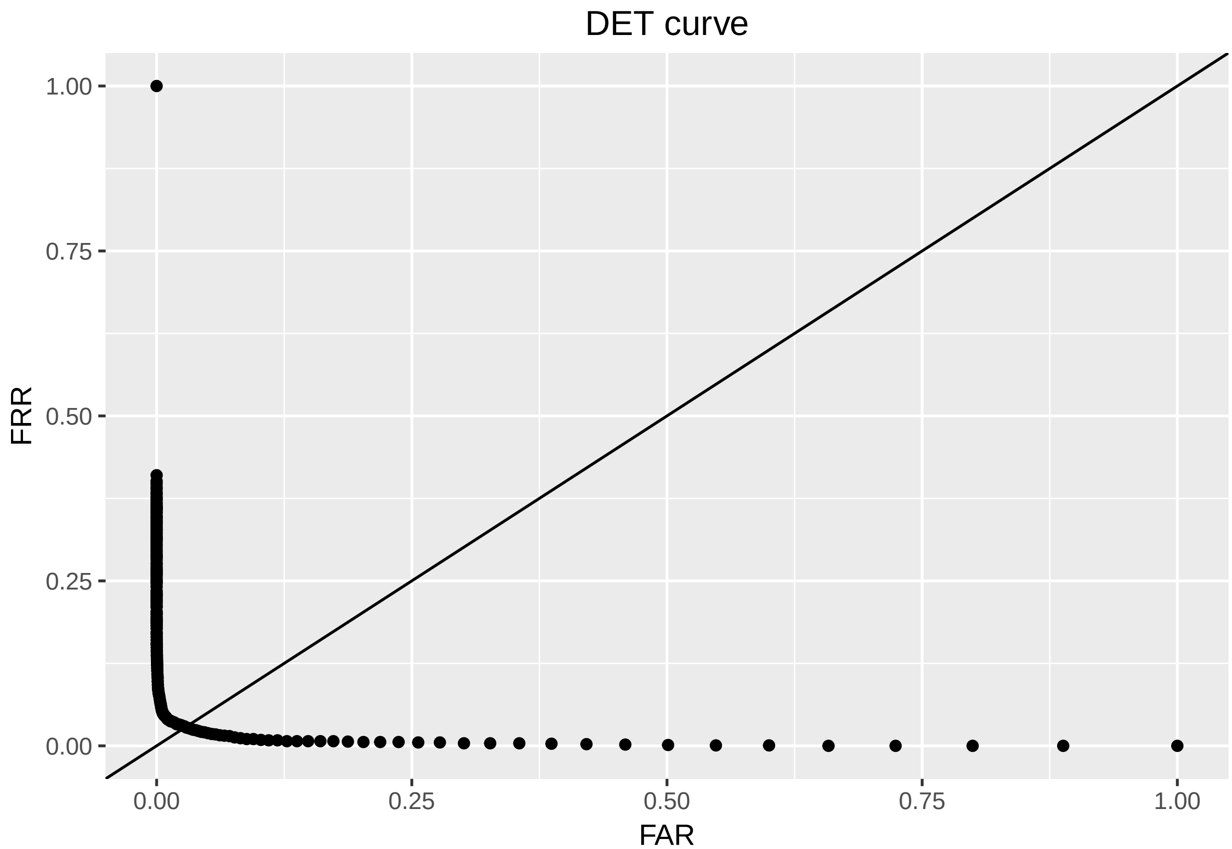
```
FAR_vs_FRR <- NULL

#For each value of t, calculate FAR and FRR and add to dataset.
for (t in seq(from = 0.0, to = 1.0, by = 0.005)){

  # use sizes of initial and subsetted dfs to calculate rate
  false_accept_count <- subset(imposter_data, imposter_data$score < t)
  false_accept_rate <- nrow(false_accept_count)/nrow(imposter_data)
  false_reject_count <- subset(genuine_data, genuine_data$score > t)
  false_reject_rate <- nrow(false_reject_count)/nrow(genuine_data)

  current_row <- c(false_accept_rate, false_reject_rate)
  FAR_vs_FRR<- rbind(FAR_vs_FRR, current_row)
}

#plot
rates_data_frame <- as.data.frame(FAR_vs_FRR)
colnames(rates_data_frame) <- c("FAR", "FRR")
ggplot(rates_data_frame, aes(x=FAR, y = FRR)) + geom_point() + geom_abline(slope = 1, intercept = 0) + l
```



Part 3B EER

```
#Make new column containing boolean FAR > FRR.
rates_data_frame$larger <- rates_data_frame$FAR > rates_data_frame$FRR

#Find where FAR becomes less than FRR. Use these as upper and lower boundaries to estimate the EER.
far_is_smaller <- rates_data_frame[rates_data_frame$larger=="FALSE", ]
lower_bound <- max(far_is_smaller$FAR)
far_is_larger <- rates_data_frame[rates_data_frame$larger=="TRUE", ]
upper_bound <- min(far_is_larger$FAR)

EER <- mean(lower_bound, upper_bound)
print(EER)
```

```
## [1] 0.02735258
```