# Data Visualization with Compas Data

### Author: [Doma]

================================================================

In May 2016, Angwin, et al. published an article in ProPublica (PP) where they claimed that COMPAS, a widely used risk assessment tool, aka recidivism model, is racially baised [1]. According to their analysis, black defendants who do not recidivate were nearly twice as likely to be misclassified by COMPAS as higher risk compared to their white counterparts (45 % vs. 23 %), and white defendants who scored who re-offended within the next two years were mistakenly labeled low risk almost twice as often as black reoffenders (48 % vs. 28 %).

Two months later, Northpointe– the company that sells COMPAS – refuted Angwin, et al. study in the paper, "COMPAS Risk Scales: Demonstrating Accuracy Equity and Predictive Parity" [2]. Brennan et al. write that PP's study didn't classify the defendents properly and suggested the high category be 8-10 and not high category be 1-7. They criticized ProPublica for not using Area Under the Curve, which captures the base rate (the reality of those who did recidivate) for each group. They found that in comparison with whites a slightly lower percentage of blacks were "Labeled Higher Risk, But Didn't Re-Offend" (37% vs. 41%) and only a slightly higher percentage of blacks were "Labeled Lower Risk, Yet Did Re-Offend" (35% vs. 29%).

Angwin et al. received data for 18,610 people who were assessed using COMPAS in the year 2013 and 2014. However, for the initial analysis, they filtered the people who were assessed at parole, probation or other stages in the criminal justice system because Broward County primarily uses the score to determine whether to release or detain a defendant before his or her trial. Brennan, et al. pointed out that failure to appear risk score is primarily focused on pretrial defendants which were not incorporated in PP's study.

I thought about doing my analysis with 18610 people in `compas_scores_raw` table, but I do not have the data on `prisonhistory`, `jailhistory`, etc on those people who are not pretrial defendents (See table `compas_scores`). Other tables will be relevent when testing for Disparate Treatement.

Angwin et al. collected public criminalrecords from the Broward County Clerk's Office website for 11757 people and created two types of datasets `compas_scores_two_years` with 7214 people for General Recidivism, and `compas_scores_two_years_violent` with 4743 people for Violent Recidivism. The datasets were selected from the github repository on Angwin, et al. study. I will recreate Angwin et al. analysis and the studies that followed up with several fairness criteria in the coming section.

```r
library(formatR)
library(dplyr)
library(ggplot2)
library(lubridate)
library(scales)  # for scale_y_continuous(label = percent)
library(AUC)

set.seed(1)
```

## Table `compas_scores_two_years` & `compas_scores_two_years_violent`

Angwin et al. used `compas_scores_two_years` table for their initial analysis and plotted histograms of the general recidivism risk score (GRRS) for black and white defendants and `compas_scores_two_years_violent` table and plotted histograms of the violent recidivism risk score (VRRS) for black and white defendants.

In `compas_scores_two_years_violent`, are `is_recid` and `two_year_recid` supposed to be `is_violent_recid` and `two_year_recid_1` ?? Presumably, `is_violent_recid` is coded with 1 if the person ever recidivated with violence and 0 if not. The number of rows for the dataset, when filtered by is_violent_recid and two_year_recid_1, is 4020 too.

I analyzed only GRRS because of the uncertainty in a variable associated with violent recidivist within two years. For instance, ProPublica mistakenly filtered is_recid instead of is_violent_recid for violent recidivism risk score.

```
# save(compas_scores_two_years, file =
# 'compas_scores_two_years.rda')
load("~/COMPAS_PROPUBLICA/compas_scores_two_years.rda")
nrow(compas_scores_two_years)
```

```
## [1] 7214
```

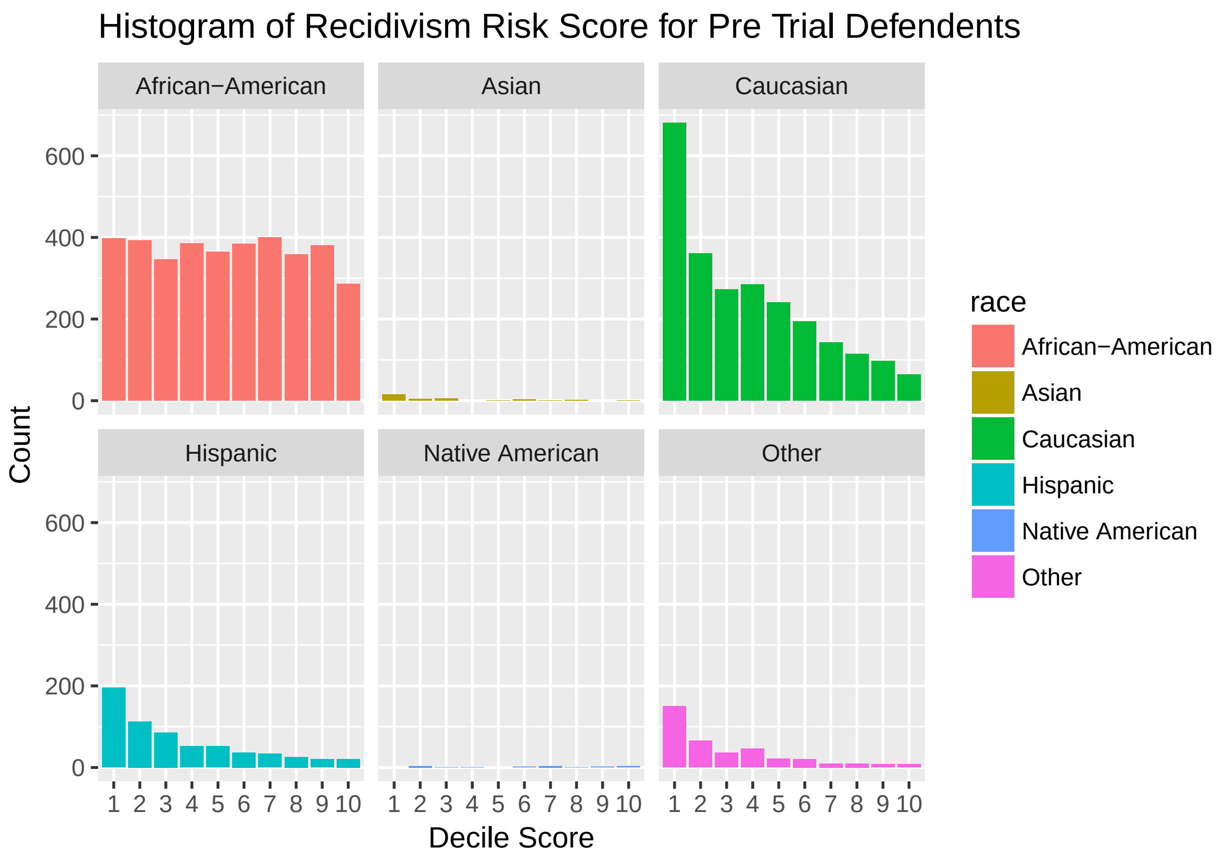
```
count(compas_scores_two_years, race)
```

```
## # A tibble: 6 x 2
##   race                 n
##   <chr>            <int>
## 1 African-American  3696
## 2 Asian               32
## 3 Caucasian         2454
## 4 Hispanic           637
## 5 Native American     18
## 6 Other              377
```

## Decile Scores

The dataset consists of a column with decile scores (1-10) as probability prediction of reoffense and the column with re-offended if yes then 1, if no then 0, and if no corresponding variables then -1.

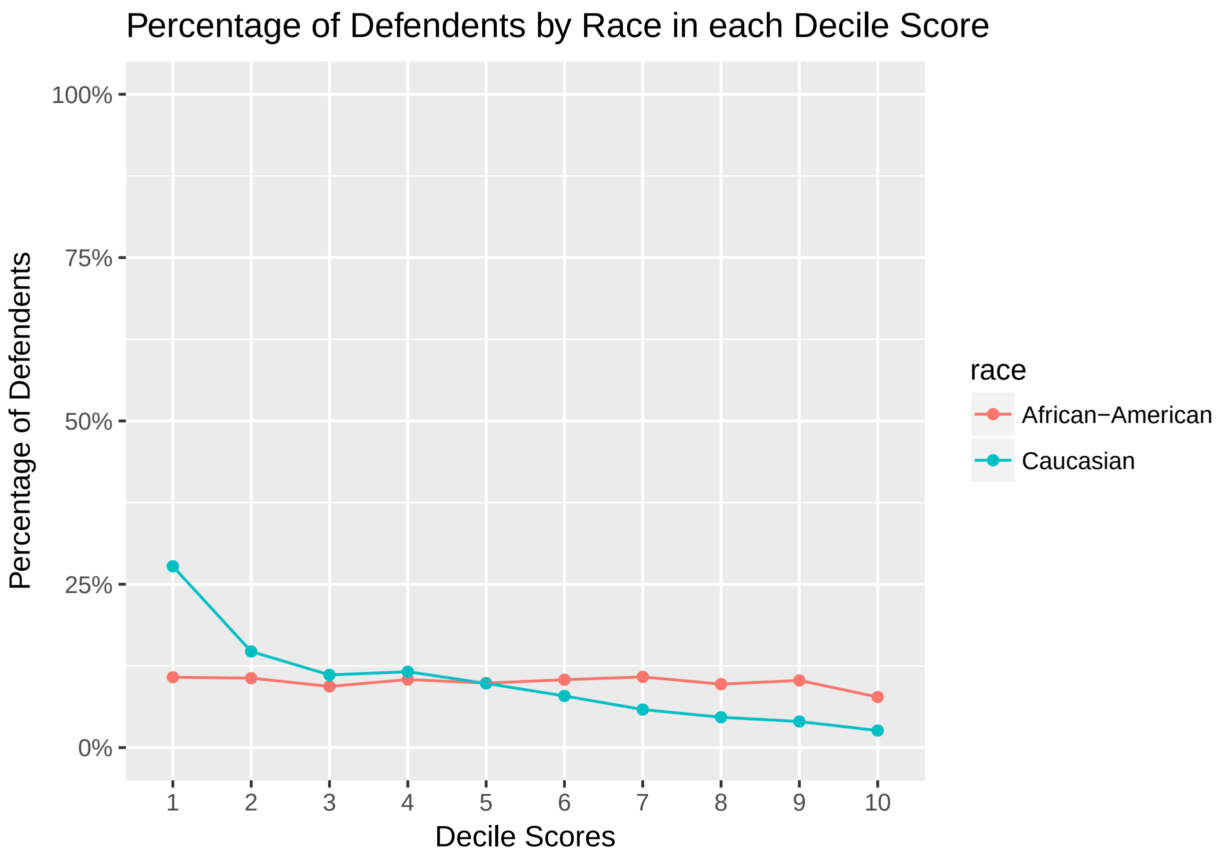
```
# Convert variables into proper type

two_years_df <- compas_scores_two_years
# Arrange Decile score

two_years_df$decile_score <- factor(two_years_df$decile_score,
                                    c("1", "2","3","4","5","6", "7", "8", "9", "10"))

two_years_df$is_recid =  as.factor(two_years_df$is_recid)


ggplot(two_years_df, aes(decile_score, fill= race)) +
  geom_bar(stat = "count") +
  xlab("Decile Score") +
  ylab("Count") +
  ggtitle("Histogram of Recidivism Risk Score for Pre Trial Defendents") +
  facet_wrap(~race)
```



The histogram shows that the distribution of people in each decile score vary immensely between black and white defendents. I will filter the dataset for African-American and Caucasian to match with Angwin et al. study. Instead of count, proportion by race is more informative.

### Plot the proportion defendents in each decile score

```
compas_race <- two_years_df %>%
   filter(race== "African-American" | race == "Caucasian")

# Compute proportion by race
r_prop_race <- compas_race %>%
  group_by(race, decile_score) %>%
  summarise(n = n()) %>%
  mutate(prop = n/sum(n))

ggplot(r_prop_race, aes(decile_score, prop, col = race, group = race)) +
  geom_point() +
  geom_line(stat= "identity") +
  scale_y_continuous( limits = c(0,1), labels = scales::percent) +
  ylab("Percentage of Defendents") +
  xlab("Decile Scores") +
  ggtitle("Percentage of Defendents by Race in each Decile Score")
```

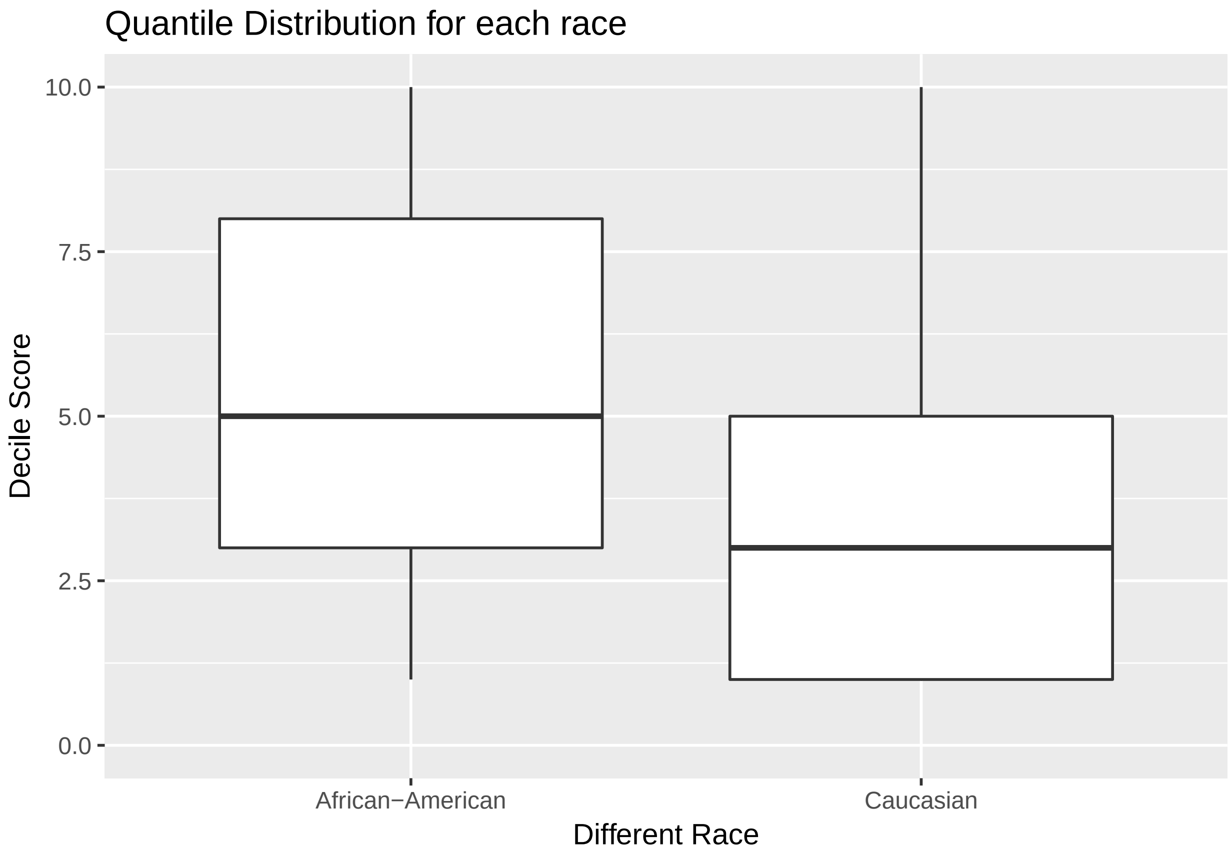


The graph above shows that the proportion of number of people in each decile score is not exactly 10 %. This raises the question whether the complete dataset that was used to obtain the decile scores are equal in proportion.

The difference in median is shown better with a boxplot. The boxplot represents percentiles (25th, 50th, 75th) such that the outliers are shown as dots. The first quartile (25th percentile) is shown as the lower border, the third quartile (75th percentile) is the upper border, and the median (50th percentile) is the line that lies in between those borders.

```
compas_race$decile_score <- as.numeric(compas_race$decile_score)

ggplot(compas_race, aes(race, decile_score)) +
  geom_boxplot(notch = FALSE) +
  xlab("Different Race") +
  ylab("Decile Score") +
  ylim(0,10) +
```

```r
  ggtitle("Quantile Distribution for each race")
```

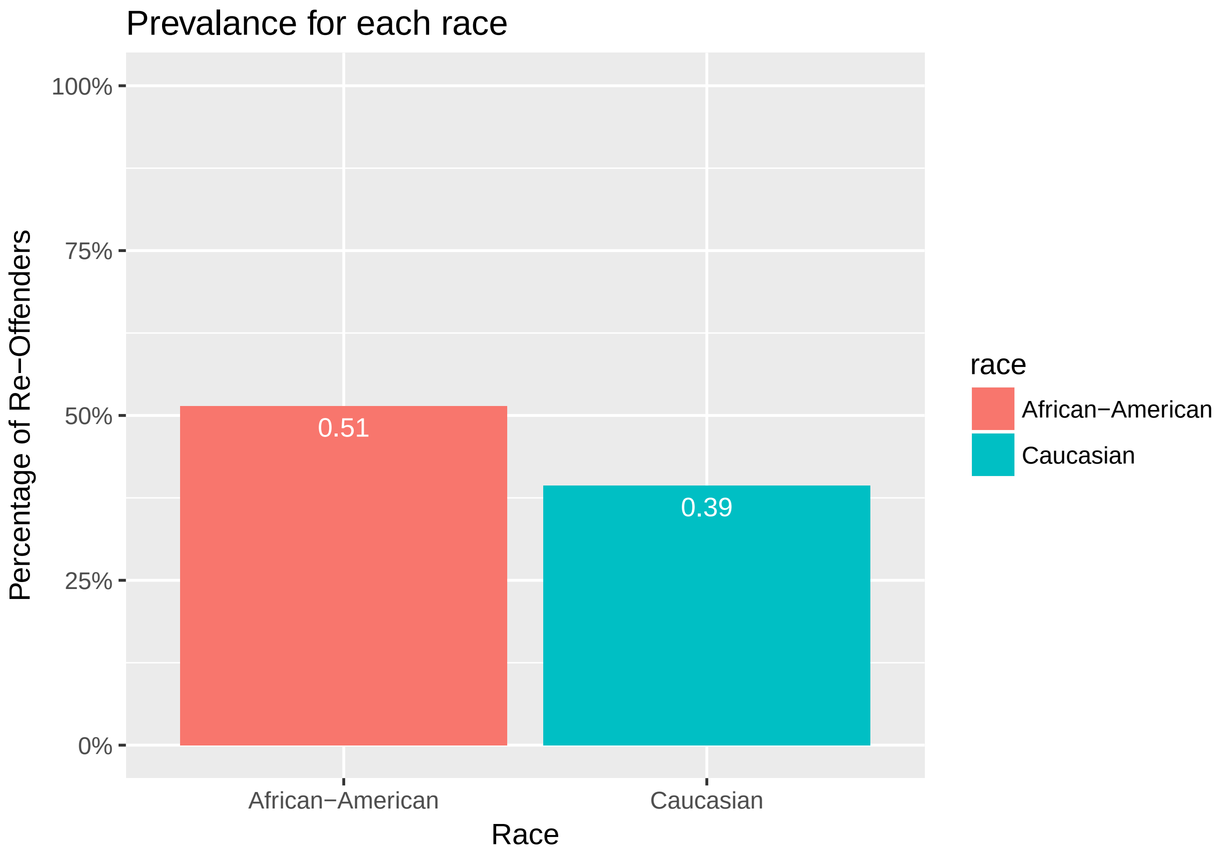


The median score for African-American is 5, whereas for Caucasian is 3.

## Prevalance (Base Rate)

Calculate the proportion of defendents who reoffended within two years of release by race

```r
base_df <- compas_race %>%
  group_by(race, two_year_recid) %>%
  summarize(n = n())

base_df2 <- base_df  %>%
  group_by(race) %>%
  mutate(total = sum(n), prop = round(n/total, 4))

prev <- base_df2 %>%
  filter(two_year_recid == "1" )

ggplot(prev, aes(race, prop, fill = race)) +
  geom_bar(stat = "identity", position = "dodge")  +
  scale_y_continuous( limits = c(0,1), labels = scales::percent) +
  geom_text(aes(label= round(prop, 2)), vjust= 1.6, color= "white", size=3.5) +
  ylab("Percentage of Re-Offenders") +
  xlab("Race") +
   ggtitle("Prevalance for each race")
```

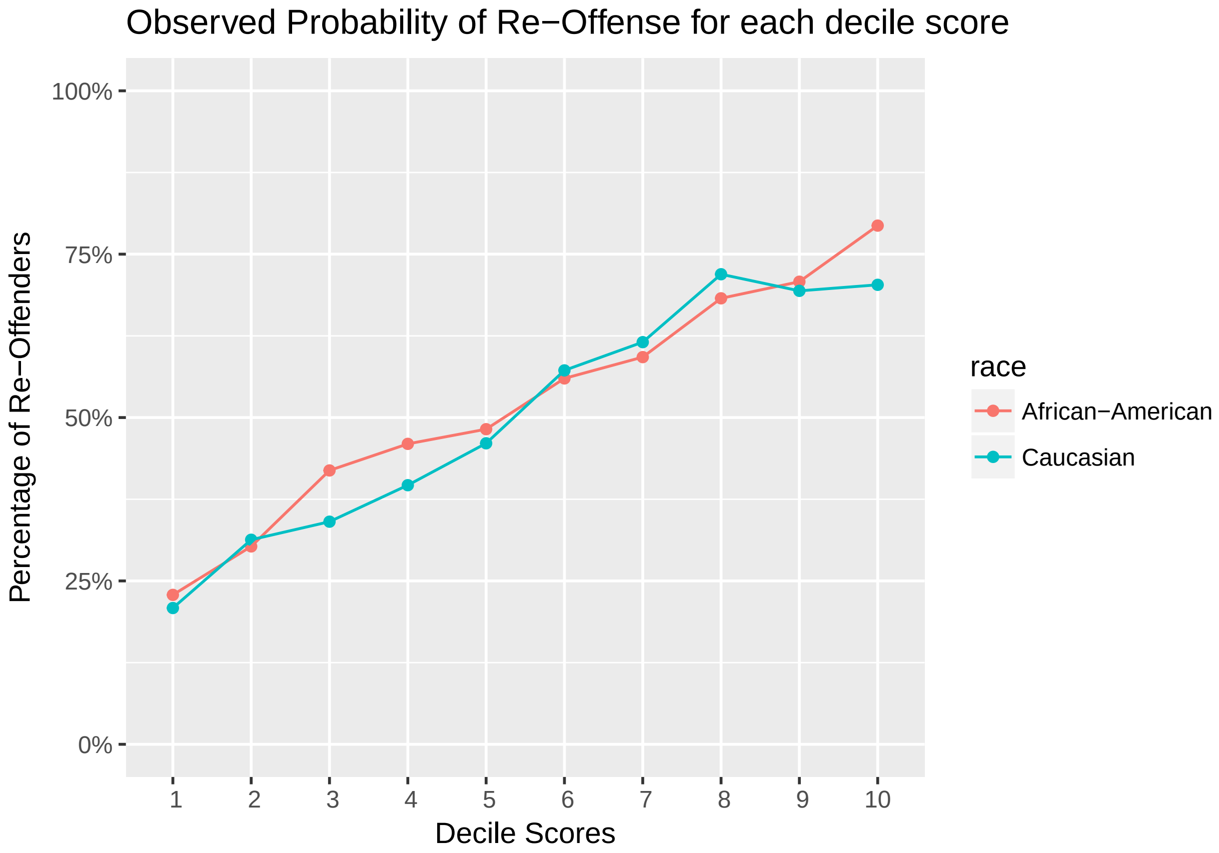


## Calibration

For every decile score, calculate the proportion of defendents who reoffended within two years of release by race

```
base_decile <- compas_race %>%
  group_by(race, two_year_recid, decile_score) %>%
  summarise(n = n())

base_decile2 <- base_decile %>%
  group_by(race, decile_score) %>%
  mutate(total = sum(n), prop = n/total)
```

```
cali <- base_decile2 %>%
  filter(two_year_recid == 1)

ggplot(cali, aes(decile_score, prop, col = race, group = race)) +
  geom_point() +
  geom_line() +
  scale_x_discrete(limits = c(1:10)) +
  scale_y_continuous( limits = c(0,1), labels = scales::percent) +
  ylab("Percentage of Re-Offenders") +
  xlab("Decile Scores")  +
  ggtitle("Observed Probability of Re-Offense for each decile score")
```



## Threshold

Since the outcome is binary (either recidivist or not recidivist), I need to group the decile scores into two categories so that it can be tested for accuracy and bias.

There are two types of error that prediction can make: false positive and false negative. In this case, false positive means a person classified as high-risk did not recidivate, while false negative means a low-risk person did recidivate. If we are willing to set more non-innocents free then we must avoid high false positives, and if we willing to lock up innocent people then we must avoid high false negatives. Deciding the threshold depends on what the policy makers are aiming to achieve.

### What is the best tool available to check for bias in a prediction model?

In predictive analytics, a two-by-two contingency table (sometimes also called a confusion matrix), is a table that reports the number of false positives, false negatives, true positives, and true negatives. This allows more detailed analysis than mere proportion of correct classifications as a whole (accuracy).

### Confusion matrix could be used to calculate accuracy and identify bias in predictions with binary outcomes.

```r
confusion_matrix <- matrix(c("True Positive", "False Positive", "False Negative", "True Negative"),
                           ncol=2)
colnames(confusion_matrix) <- c('Predicted Positive', 'Predicted Negative')
rownames(confusion_matrix) <- c('Actual Positive', 'Actual Negative')
as.table(confusion_matrix)
```

```
##                 Predicted Positive Predicted Negative
## Actual Positive True Positive      False Negative
## Actual Negative False Positive     True Negative
```

### How does accuracy and bias in prediction change when different threshold for decile scores are selected?

I will set decile score 1 as predicted No and 2-10 as predicted Yes. Then, I will compute confusion matrix for that predictions and repeat the process for every other thresholds to check if some types of categorization are better than other.

Categorize data into threshold 0-10. Then calculate the number of True Postive, True Negative, False Positive and False Negative, and use the results to find the accuracy, and other statistical tests (refer to the "Formulae" section of my thesis paper for details on how I computed the values).

```r
make_matrix <- function( decile ) {

  compas_DT <- compas_race %>%
  mutate(decile_cat = ifelse(decile_score <= decile , "low", "high"))

df1 <- compas_DT %>%
  filter(decile_cat == "high", two_year_recid == 1) %>%
  group_by(race) %>%
  summarize(true_positive = n())

df2 <- compas_DT %>%
  filter(decile_cat == "low", two_year_recid == 1) %>%
  group_by(race) %>%
  summarize(false_negative = n())
```

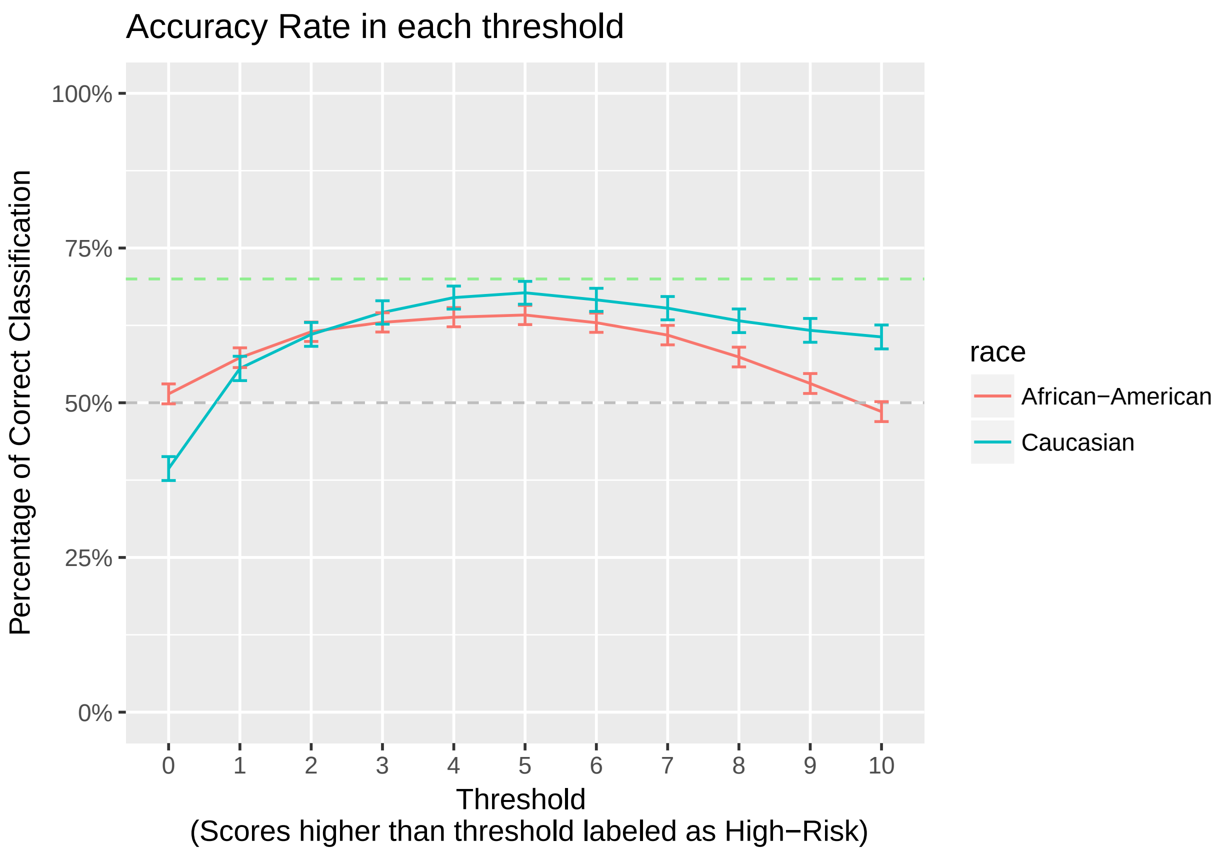
```
df3 <- compas_DT %>%
  filter(decile_cat == "high", two_year_recid == 0) %>%
  group_by(race) %>%
  summarize(false_positive = n())

df4 <- compas_DT %>%
  filter(decile_cat == "low", two_year_recid == 0) %>%
  group_by(race) %>%
  summarize(true_negative = n())

return( df1 %>%
  full_join(df2, by = c("race", "race")) %>%
  full_join(df3, by = c("race", "race")) %>%
  full_join(df4, by = c("race", "race")) %>%
  mutate(threshold = as.character( decile ) ) )
}

full_matrix <- bind_rows(make_matrix( 0 ), make_matrix( 1 ), make_matrix( 2 ),
                         make_matrix( 3 ), make_matrix( 4 ), make_matrix( 5 ),
                         make_matrix( 6 ), make_matrix( 7 ), make_matrix( 8 ),
                         make_matrix( 9 ), make_matrix( 10 ))

 full_matrix[is.na(full_matrix)] <- 0

full_matrix2 <- full_matrix %>%
  group_by(race, threshold) %>%
  mutate(total = sum(true_positive, true_negative, false_positive, false_negative)) %>%
  mutate(accuracy = (true_positive + true_negative)/total,
         overall_misclass_rate =  (false_positive + false_negative)/total,
         prevalance = (true_positive + false_negative)/total,
             not_prevalance = (true_negative + false_positive)/total,
         predicted_probability = (true_positive + false_positive)/total,
         positive_predicted_value = true_positive/(true_positive + false_positive),
         negative_predicted_value = true_negative/(true_negative + false_negative),
         false_discovery_rate = false_positive/(false_positive + true_positive),
         false_omission_rate = false_negative/(false_negative + true_negative),
         true_positive_rate = true_positive/(true_positive + false_negative),
         true_negative_rate = true_negative/(true_negative + false_positive),
         false_positive_rate = false_positive/(false_positive + true_negative),
         false_negative_rate = false_negative/(false_negative + true_positive))


se <- function(n, p) {

    1.96 * sqrt((p*(1-p))/n)

}

se_data2 <- full_matrix2 %>%
  mutate(se_accuracy = se(total, accuracy),
         se_misclass = se(total, overall_misclass_rate),
         se_prevalance = se(total, prevalance),
         se_pred_prob = se(total, predicted_probability),
         se_ppv = se(total, positive_predicted_value),
         se_npv = se(total, negative_predicted_value),
         se_fdr = se(total, false_discovery_rate),
         se_for = se(total, false_omission_rate),
         se_tpr = se(total, true_positive_rate),
         se_tnr = se(total, true_negative_rate),
         se_fnr = se(total, false_negative_rate),
         se_fpr = se(total, false_positive_rate))

# save(se_data2, file = "~/Documents/COMPAS_PROPUBLICA/se_data2.rda")
```

“Positive Predictive Value (PV+) is the probability that a person with a positive test result (”Not Low" risk score) will recidivate. The Negative Predictive Value (PV-) is the probability that a person with a negative test result (“Low” risk score) will not recidivate."

In “Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment” Zafar, et al. proposes disparate mistreatment as well-suited for scenarios where ground truth is available for historical decisions used during the training phase [3]. They write that a decision making process suffers from disparate mistreatment with respect to a given sensitive attribute (e.g., race) if the misclassiffcation rates differ for groups of people having different values of that sensitive attribute.

**Plot the accuracy for each threshold by race.**

```
# Arrange threshold

se_data2$threshold <- factor(se_data2$threshold,
                             c("0","1", "2","3","4","5","6", "7", "8", "9", "10"))

ggplot(se_data2, aes(threshold, accuracy, color = race, group = race)) +
  geom_line() +
  geom_hline(yintercept=.7, linetype=2, color = "lightgreen")  +
  geom_hline(yintercept=.5, linetype=2, color = "gray")  +
  scale_y_continuous( limits = c(0,1), labels = scales::percent) +
  geom_errorbar(aes(ymin = accuracy - se_accuracy,
                    ymax = accuracy + se_accuracy),
                    width = 0.2)  +
  ylab("Percentage of Correct Classification") +
  xlab("Threshold \n (Scores higher than threshold labeled as High-Risk)") +
  ggtitle("Accuracy Rate in each threshold")
```



The x-axis represents categorical variable (Thresholds) and the y-axis represents the acurracy in percentage. The graph tells us that the accuracy for African-American is better than Caucasian up to threshold 2, whereas from threshold 3 onwards accuracy for Caucasian increases and we can see substantiate difference in accuracy at threshold 9 and 10.

While ROC is plotted with True Positive Rate as a function of False Positive Rate, it displays the trade-off True Positive Rate (sensitivity) and True Negative Rate(specificity).

```
df <- two_years_df

df[["decile_score"]] <- as.numeric(df[["decile_score"]])
df[["two_year_recid"]] <- as.factor(df[["two_year_recid"]])

white <- df %>%
  filter( race== "Caucasian") %>%
  mutate(predicted = decile_score/10)
nrow(white)
```

```
## [1] 2454
```

```
white[["predicted"]] <- as.factor(white[["predicted"]])

white_roc <- roc(white$predicted,  white$two_year_recid )
```

```r
auc(white_roc)
```

```
## [1] 0.6931463
```

```r
black  <- df %>%
  filter( race== "African-American") %>%
  mutate(predicted = decile_score/10)
nrow(black)
```

```
## [1] 3696
```

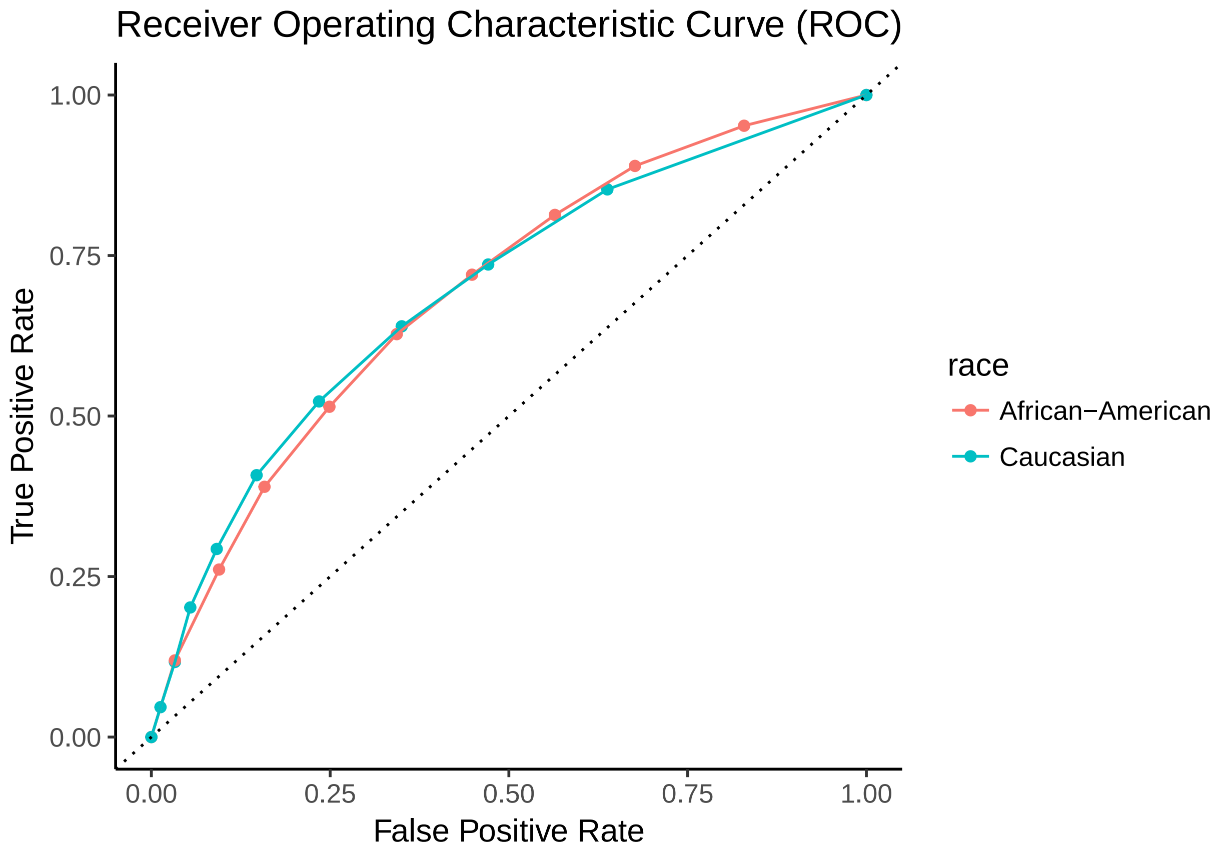
```r
black[["predicted"]] <- as.factor(black[["predicted"]])

black_roc <- roc(black$predicted, black$two_year_recid )
auc(black_roc)
```

```
## [1] 0.6918344
```

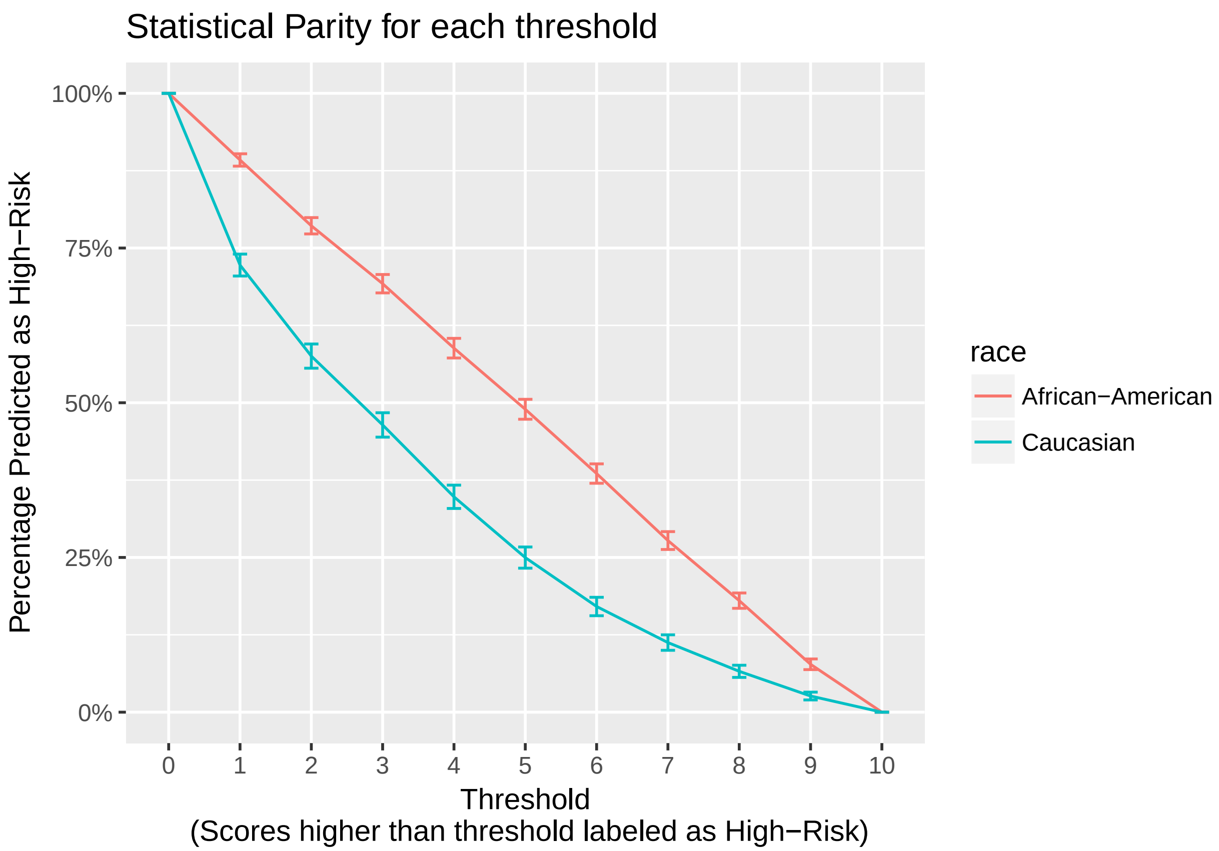
```r
all <- df %>%
  mutate(predicted = decile_score/10)

all_roc <- roc(all$predicted, all$two_year_recid )
auc(all_roc)
```

```
## [1] 0.7021663
```

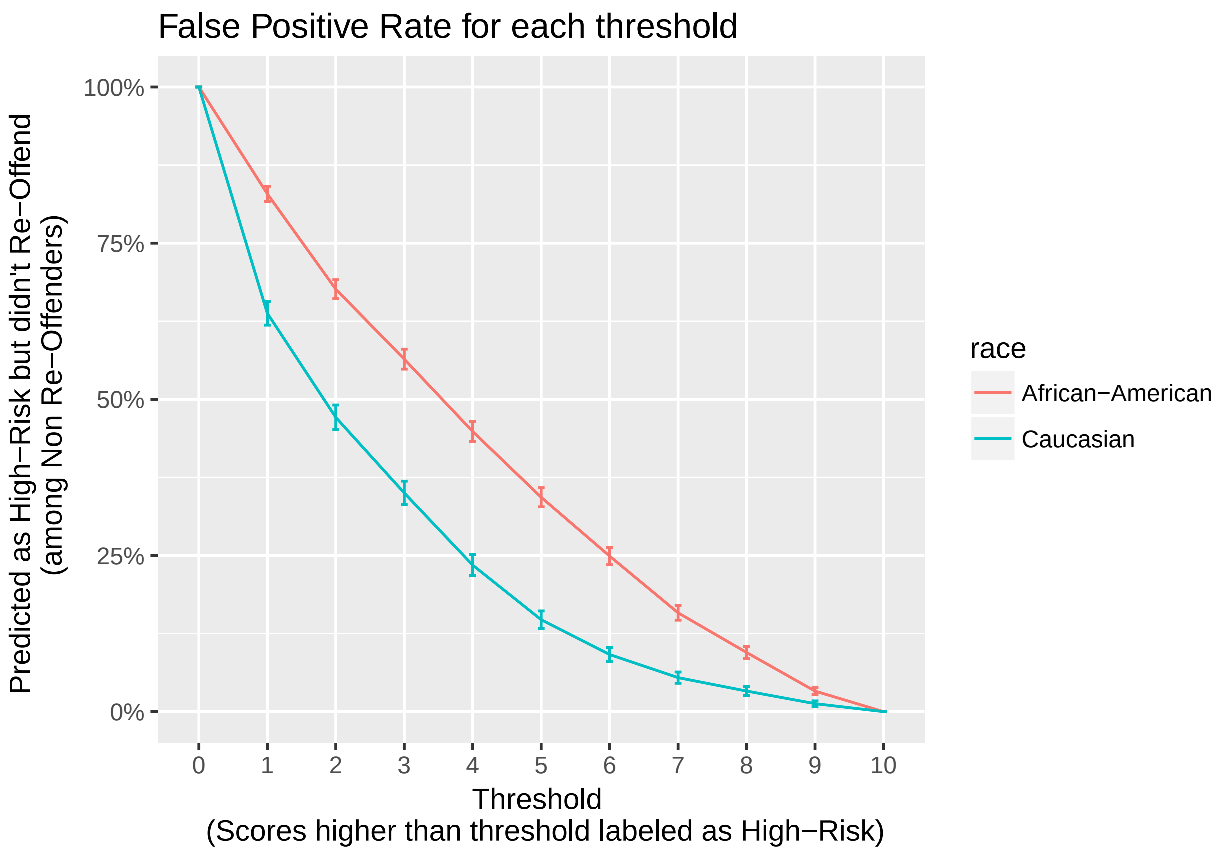
```r
ggplot(se_data2, aes(false_positive_rate, true_positive_rate,
                     col = race,
                     label = threshold,
                     group = race))  +
  geom_point() +
  geom_line() +
  theme_classic(base_size = 12) +
  xlab("False Positive Rate") +
  ylab("True Positive Rate") +
  geom_abline(intercept=0, slope=1, lty=3) +
  xlim(0,1) +
  ylim(0,1) +
  ggtitle("Receiver Operating Characteristic Curve (ROC)")
```



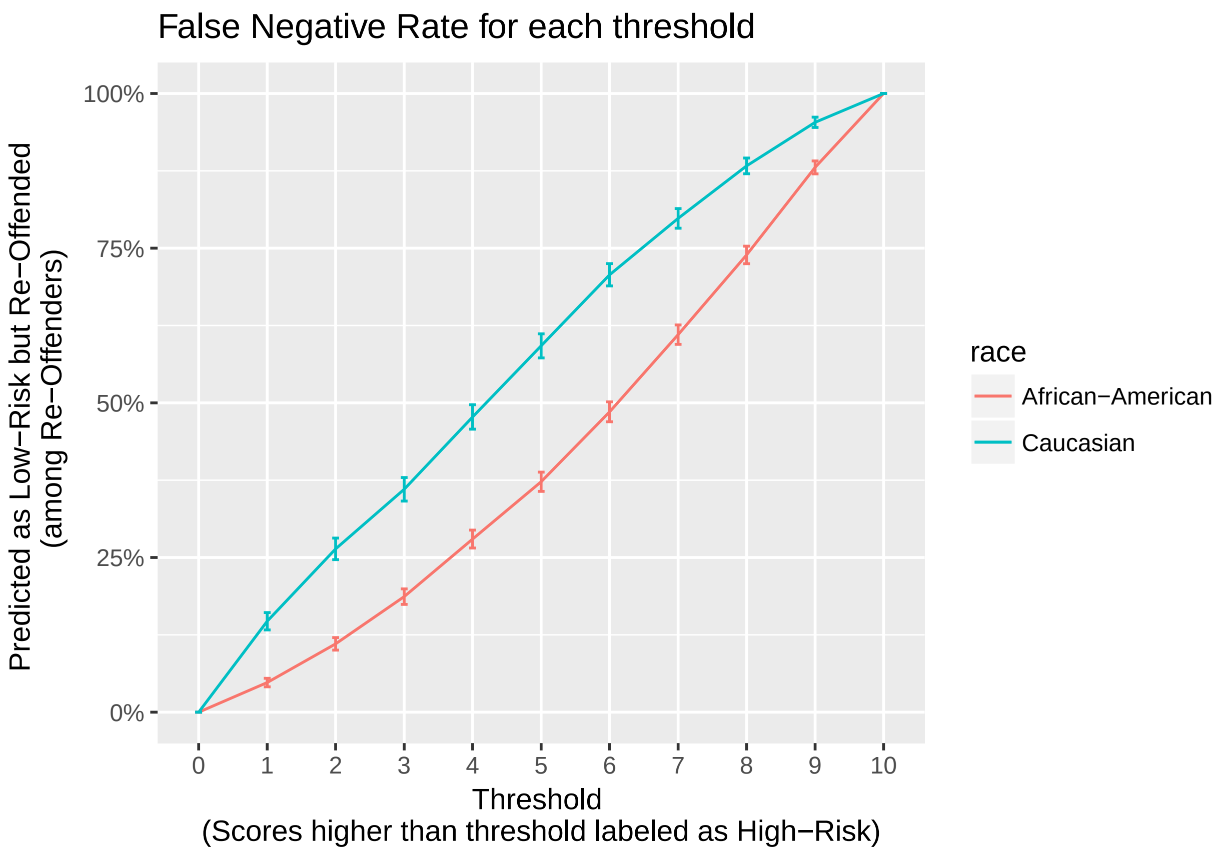
```
ggplot(se_data2, aes(threshold, predicted_probability, color = race, group = race)) +
  geom_line() +
  scale_y_continuous( limits = c(0,1), labels = scales::percent) +
  geom_errorbar(aes(ymin = predicted_probability - se_pred_prob,
                    ymax = predicted_probability + se_pred_prob),
                    width = 0.2)  +
  ylab("Percentage Predicted as High-Risk") +
  xlab("Threshold\n (Scores higher than threshold labeled as High-Risk)") +
  ggtitle("Statistical Parity for each threshold")
```



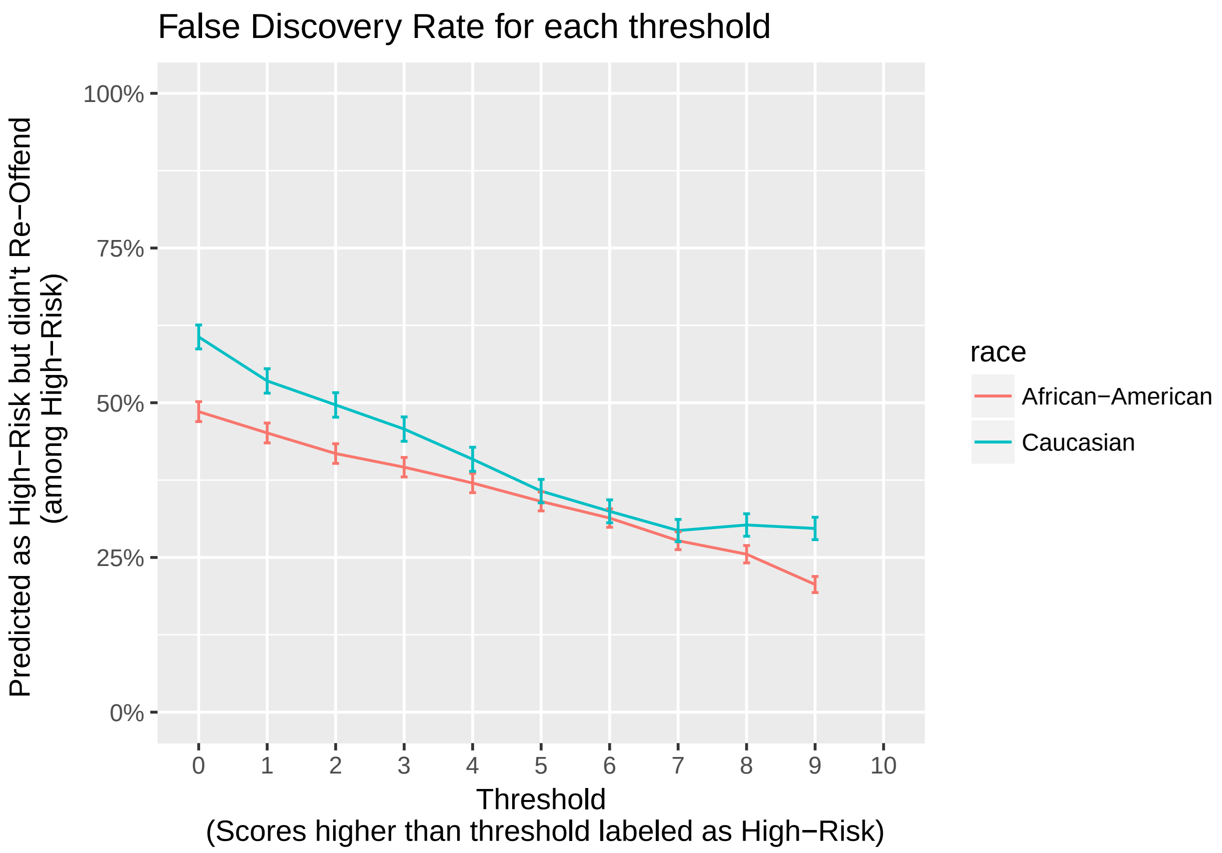
```
ggplot(se_data2, aes(threshold, false_positive_rate, color = race, group = race)) +
  geom_line()+
  scale_y_continuous( limits = c(0,1), labels = scales::percent) +
  geom_errorbar(aes(ymin = false_positive_rate - se_fpr,
                    ymax = false_positive_rate + se_fpr),
                    width = 0.1)  +
  ylab("Predicted as High-Risk but didn't Re-Offend \n (among Non Re-Offenders)") +
  xlab("Threshold \n (Scores higher than threshold labeled as High-Risk)") +
  ggtitle("False Positive Rate for each threshold")
```



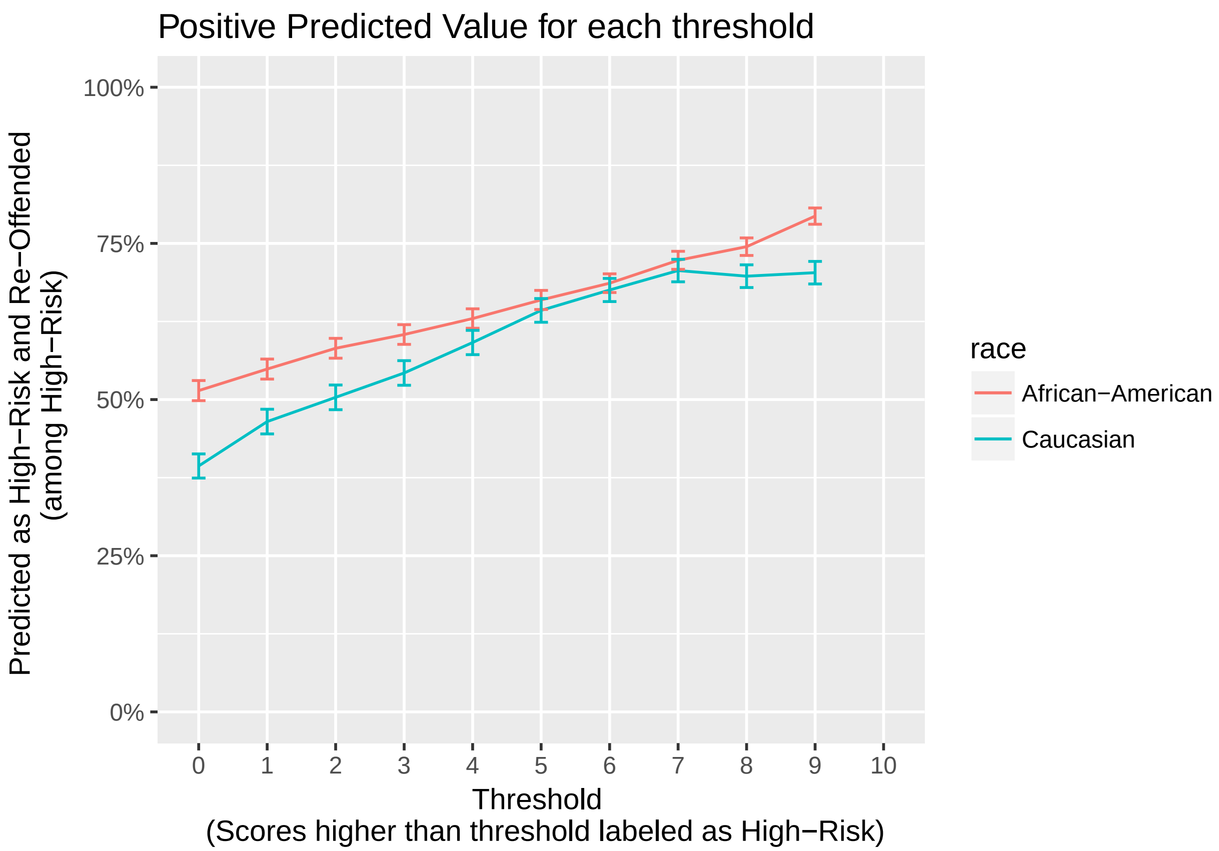
```
ggplot(se_data2, aes(threshold, false_negative_rate, color = race, group = race)) +
  geom_line() +
  scale_y_continuous( limits = c(0,1), labels = scales::percent) +
  geom_errorbar(aes(ymin = false_negative_rate - se_fnr,
                    ymax = false_negative_rate + se_fnr),
                    width = 0.1)  +
  ylab("Predicted as Low-Risk but Re-Offended \n (among Re-Offenders)") +
  xlab("Threshold \n(Scores higher than threshold labeled as High-Risk)") +
  ggtitle("False Negative Rate for each threshold")
```



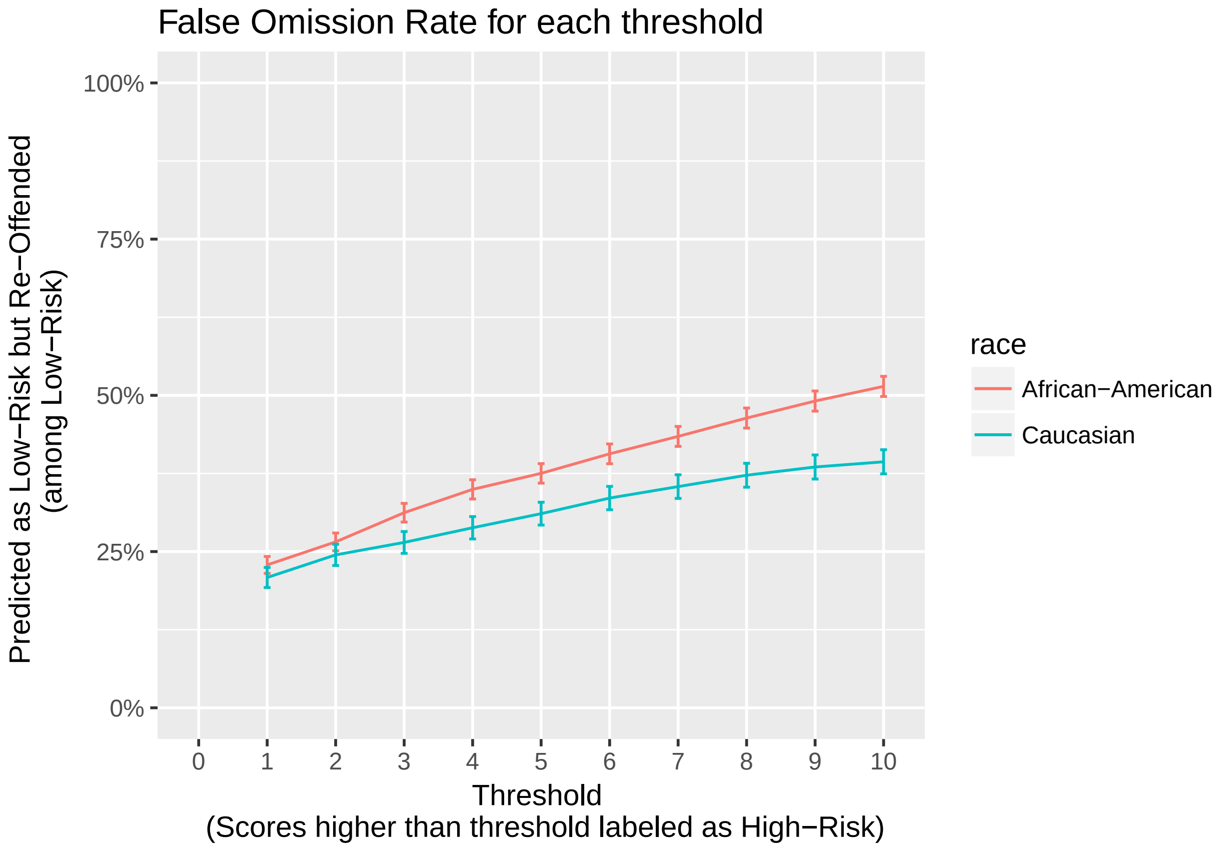
```r
ggplot(se_data2, aes(threshold, false_discovery_rate, color = race, group = race)) +
 geom_line() +
 scale_y_continuous( limits = c(0,1), labels = scales::percent) +
 geom_errorbar(aes(ymin = false_discovery_rate - se_fdr,
                   ymax = false_discovery_rate + se_fdr),
                   width = 0.1)  +
 ylab("Predicted as High-Risk but didn't Re-Offend \n (among High-Risk)") +
 xlab("Threshold\n (Scores higher than threshold labeled as High-Risk)") +
 ggtitle("False Discovery Rate for each threshold")
```



```
ggplot(se_data2, aes(threshold, positive_predicted_value , color = race, group = race)) +
  geom_line() +
  scale_y_continuous( limits = c(0,1), labels = scales::percent) +
  geom_errorbar(aes(ymin = positive_predicted_value - se_ppv,
                    ymax = positive_predicted_value + se_ppv),
                    width = 0.2)  +
  ylab("Predicted as High-Risk and Re-Offended \n (among High-Risk)") +
  xlab("Threshold \n (Scores higher than threshold labeled as High-Risk)") +
  ggtitle("Positive Predicted Value for each threshold")
```



```r
ggplot(se_data2, aes(threshold, false_omission_rate, color = race, group = race)) +
 geom_line() +
 scale_y_continuous( limits = c(0,1), labels = scales::percent) +
 geom_errorbar(aes(ymin = false_omission_rate - se_for,
                   ymax = false_omission_rate + se_for),
                   width = 0.1)  +
 ylab("Predicted as Low-Risk but Re-Offended \n (among Low-Risk)") +
 xlab("Threshold \n (Scores higher than threshold labeled as High-Risk)") +
 ggtitle("False Omission Rate for each threshold")
```



**References**

[1] Angwin, Julia. “Machine Bias: There’s Software Used across the Country to Predict Future Criminals. And It’s Biased against Blacks.” ProPublica, 23 May 2016, www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[2] Dieterich, William, et al. “COMPAS Risk Scales: Demonstrating Accuracy Equity and Predictive Parity.” Equivant, NorthPointe, 8 July 2016,

go.volarisgroup.com/rs/430-MBX-989images/ProPublica_Commentary_Final_070616.pdf.

[3] Zafar, Muhammad Bilal, et al. “Learning Classification without Disparate Mistreatment.” Fairness Beyond Disparate Treatment & Disparate Impact, International World Wide Web Conferences Steering Committee, 8 Mar. 2017, doi.acm.org/10.1145/3038912.3052660.