COMPAS Data Wrangling and Analysis

Dasha Asienga

2024-02-29

Contents

The thesis body will have more in-depth descriptions of the data analysis as well as select output and results from this file. This file is intended for general preliminary analysis of the COMPAS data set. Note that the results can only be generalized to Broward County, Florida, but there are important findings about the United States judicial system nevertheless.

Reading in the Data

```r
#read in the data
compas_path <- "/home/dasienga24/Statistics-Senior-Honors-Thesis/Data Sets/COMPAS/compas_data.csv"
compasdata <- read.csv(compas_path)
```

The Data Set

The COMPAS data set has 12160 observations of defendants that were evaluated for the risk of recidivism by the COMPAS tool. There are 29 variables of interest as described below:

- `id`: unique person identifier.
- `compas_person_id`: unique COMPAS case identifier.
- `name`: full name.
- `first`: first name.
- `last`: last name.
- `sex`: sex categorized as male or female.
- `race`: race categorized as African-American, Asian, Caucasian, Hispanic, Native American, or Other.
- `age`: numeric age, ranging from 18 to 96.
- `age_cat`: age categorized as Less than 25, 25 - 45, or Greater than 45.
- `marital_status`: marital status categorized as Single, Significant Other, Married, Widowed, Separated, Divorced, or Unknown.
- `custody_status`: custody status categorized as Jail Inmate, Prison Inmate, Pretrial Defendant, Parole, Residential Program, or Probation.
- `juv_fel_count`: number of prior juvenile felonies, ranging from 0 to 20.
- `juv_misd_count`: number of prior juvenile misdemeanors, ranging from 0 to 13.
- `juv_other_count`: number of other prior juvenile offenses, ranging from 0 to 17.
- `priors_count`: number of non-juvenile prior offenses, ranging from 0 to 43.
- `days_b_screening_arrest`: number of days between COMPAS screening and arrest.
- `c_days_from_compas`: the number of days since COMPAS screening.
- `c_charge_degree`: the charge degree according to the appropriate laws.
- `c_charge_desc`: the charge description in words.
- `type_of_assessment`: the type of assessment, in this case, the assessment is 'Risk of Recidivism'.
- `raw_score`: COMPAS tool raw score on risk of recidivism.
- `decile_score`: decile rank on a scale of 1 - 10 based on the COMPAS raw score.
- `score_text`: COMPAS risk of recidivism based on the decile scores and categorized as High, Medium, or Low.
- `is_violent_recid`: categorical variable recording whether a defendant was accused of a violent crime within 2 years (0 = N, 1 = Y).
- `num_vr_cases`: number of times a defendant was accused of a violent crime within 2 years.
- `is_recid`: categorical variable recording whether a defendant was accused of a crime within 2 years (0 = N, 1 = Y).
- `num_r_cases`: number of times a defendant was accused of a crime within 2 years.
- `days_in_jail`: number of days spent in jail before COMPAS screening.
- `days_in_prison`: number of days spent in prison before COMPAS screening.

```r
colnames(compasdata)
```

```
##  [1] "id"                      "compas_person_id"
##  [3] "name"                    "first"
##  [5] "last"                    "sex"
##  [7] "race"                    "age"
##  [9] "age_cat"                 "marital_status"
## [11] "custody_status"          "juv_fel_count"
## [13] "juv_misd_count"          "juv_other_count"
## [15] "priors_count"            "days_b_screening_arrest"
## [17] "c_days_from_compas"      "c_charge_degree"
## [19] "c_charge_desc"           "type_of_assessment"
## [21] "raw_score"               "decile_score"
## [23] "score_text"              "is_violent_recid"
## [25] "num_vr_cases"            "is_recid"
## [27] "num_r_cases"             "days_in_jail"
## [29] "days_in_prison"
```

Data Wrangling

Before proceeding with the data analysis, we first need to handle some data anomalies. We'll also only consider COMPAS cases within 30 days of arrest to improve the data quality. This resulted in 9683 total observations.

```
compasdata <- compasdata %>%
  filter(decile_score > 0 & is_recid != -1 & days_b_screening_arrest >= -30 &
           days_b_screening_arrest <= 30) %>%
  mutate(days_b_screening_arrest = abs(days_b_screening_arrest))

count(compasdata)
```

```
##      n
## 1 9683
```

Next, let's also make sure that there are no duplicate defendants.

```
clean_compasdata <- compasdata[-which(duplicated(compasdata$id)), ]
```

We'll proceed with this data set and 9387 observations total.

Descriptive Statistics

Now that the data is clean, let's generate some descriptive statistics to understand the distribution of the variables in the data set and their relationships with each other.

First, below is a glimpse of the data as described above. Notice that there is a lot of missing data for `num_vr_cases` and `num_r_cases` because that information is only recorded for defendants that recommit a crime in the next 2 years.

```
glimpse(clean_compasdata)
```

```
## Rows: 9,387
## Columns: 29
## $ id                      <int> 1, 3, 4, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, ~
## $ compas_person_id        <int> 56418, 51601, 38864, 59301, 61330, 56890, 6199~
## $ name                    <chr> "miguel hernandez", "kevon dixon", "ed philo",~
## $ first                   <chr> "miguel", "kevon", "ed", "marsha", "edward", "~
## $ last                    <chr> "hernandez", "dixon", "philo", "miles", "riddl~
## $ sex                     <chr> "Male", "Male", "Male", "Male", "Male", "Male"~
## $ race                    <chr> "Other", "African-American", "African-American~
## $ age                     <int> 69, 34, 24, 44, 41, 43, 39, 20, 26, 27, 23, 37~
## $ age_cat                 <chr> "Greater than 45", "25 - 45", "Less than 25", ~
## $ marital_status          <chr> "Single", "Single", "Single", "Separated", "Si~
## $ custody_status          <chr> "Jail Inmate", "Jail Inmate", "Jail Inmate", "~
## $ juv_fel_count           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ juv_misd_count          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ juv_other_count         <int> 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0~
## $ priors_count            <int> 0, 0, 4, 0, 14, 3, 0, 0, 0, 0, 3, 0, 0, 0, 1, ~
## $ days_b_screening_arrest <int> 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 20, ~
## $ c_days_from_compas      <int> 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 490,~
```

```
## $ c_charge_degree        <chr> "(F3)", "(F3)", "(F3)", "(M1)", "(F3)", "(F3)"~
## $ c_charge_desc          <chr> "Aggravated Assault w/Firearm", "Felony Batter~
## $ type_of_assessment     <chr> "Risk of Recidivism", "Risk of Recidivism", "R~
## $ raw_score              <dbl> -2.78, -0.76, -0.66, -1.93, -0.16, -0.72, -1.7~
## $ decile_score           <int> 1, 3, 4, 1, 6, 4, 1, 10, 5, 4, 6, 1, 3, 4, 1, ~
## $ score_text             <chr> "Low", "Low", "Low", "Low", "Medium", "Low", "~
## $ is_violent_recid       <int> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ num_vr_cases           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ is_recid               <int> 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1~
## $ num_r_cases            <int> NA, 3, 1, NA, 3, NA, NA, NA, NA, NA, 1, NA, NA~
## $ days_in_jail           <dbl> 1, 10, 1, NA, 6, 1, 3, 33, 1, 1, NA, NA, NA, 1~
## $ days_in_prison         <dbl> NA, NA, NA, NA, 1065, NA, NA, NA, NA, NA, NA, ~
```

Next, we will perform some univariate analysis for the variables in the data set before proceeding to conduct some bivariate and multivariate analysis.

Univariate Analysis

Univariate analysis will involve looking at some summary statistics and visualizations of the different variables in the data set.

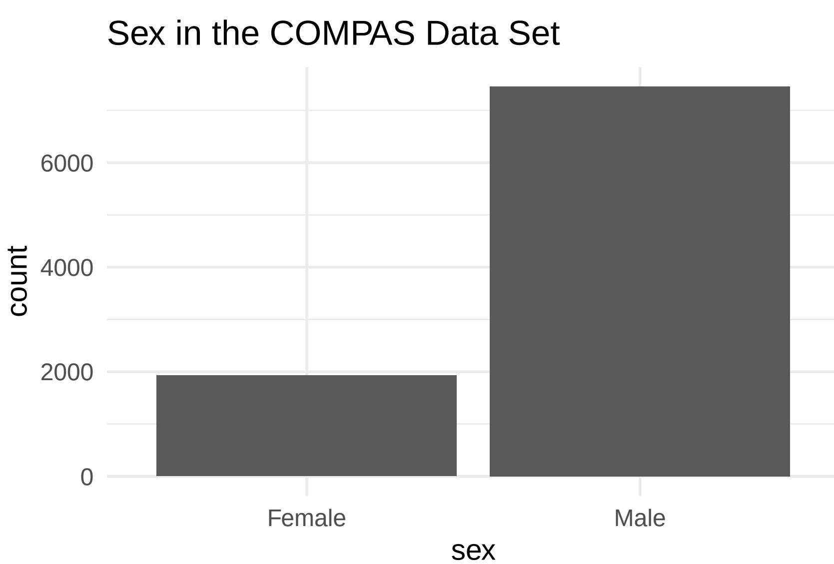
Categorical Variables

There 7457 males and 1930 females in the data set.

```
tally(clean_compasdata$sex)
```

```
## X
## Female   Male
##   1930   7457
```

```
ggplot(data = clean_compasdata, mapping = aes(x = sex)) +
  geom_bar() +
  theme_minimal() +
  labs(title = "Sex in the COMPAS Data Set")
```

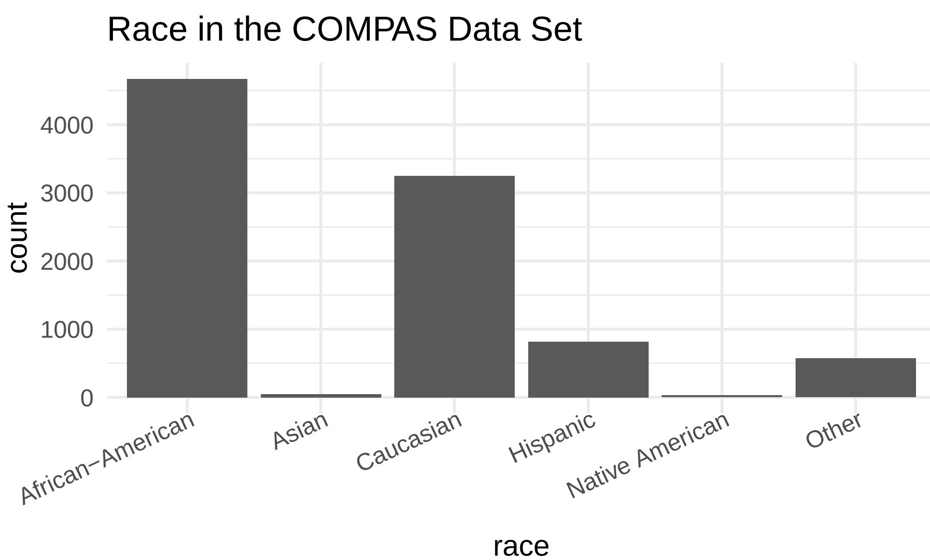


Most of the defendants are African-American and Caucasian, with only 27 Native Americans and 48 Asians.

```
tally(clean_compasdata$race)
```

```
## X
## African-American            Asian        Caucasian         Hispanic
##             4674               48             3250              818
##  Native American            Other
##               27              570
```

```
ggplot(data = clean_compasdata, mapping = aes(x = race)) +
  geom_bar() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1)) +
  labs(title = "Race in the COMPAS Data Set")
```

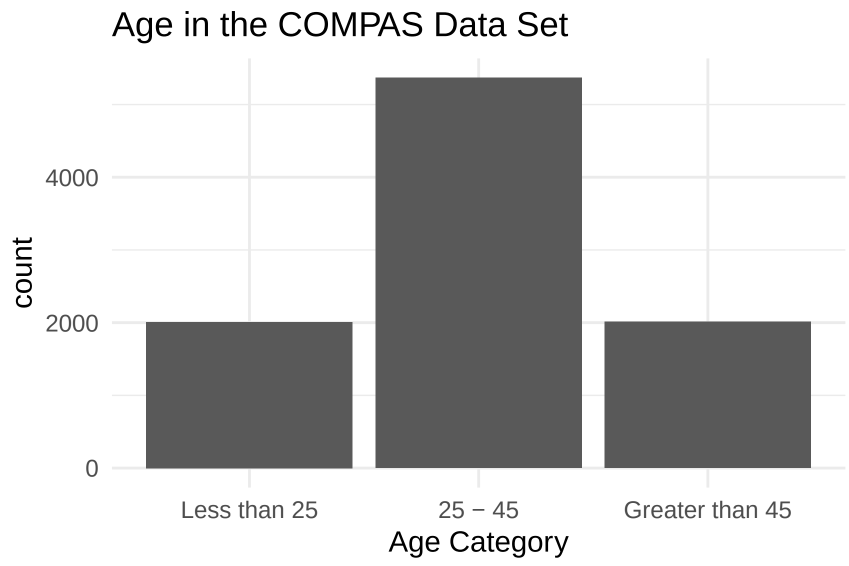


Majority of the defendants are between the age of 25 and 45, with about the same number of defendants less than 25 and greater than 25.

```
tally(clean_compasdata$age_cat)
```

```
## X
##          25 - 45 Greater than 45    Less than 25
##             5366            2012            2009
```

```
order <- c("Less than 25", "25 - 45", "Greater than 45")

ggplot(data = clean_compasdata, mapping = aes(x = age_cat)) +
  geom_bar() +
  theme_minimal() +
  scale_x_discrete(limits = order) +
  labs(x = "Age Category",
       title = "Age in the COMPAS Data Set")
```

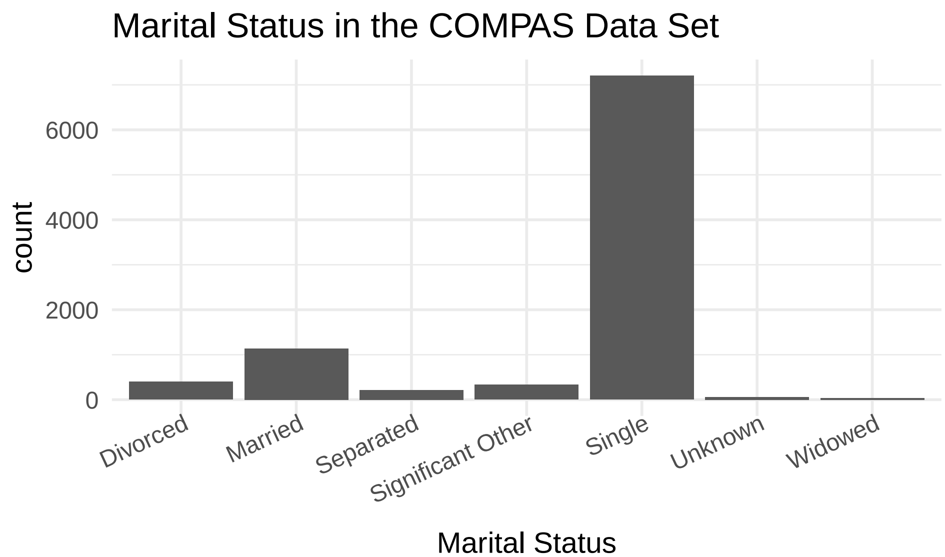


Most of the defendants are single, followed by married.

```r
tally(clean_compasdata$marital_status)
```

```
## X
##            Divorced             Married           Separated Significant Other
##                 398                1138                 219               332
##              Single             Unknown             Widowed
##                7202                  58                  40
```

```r
ggplot(data = clean_compasdata, mapping = aes(x = marital_status)) +
  geom_bar() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1)) +
  labs(x = "Marital Status",
       title = "Marital Status in the COMPAS Data Set")
```

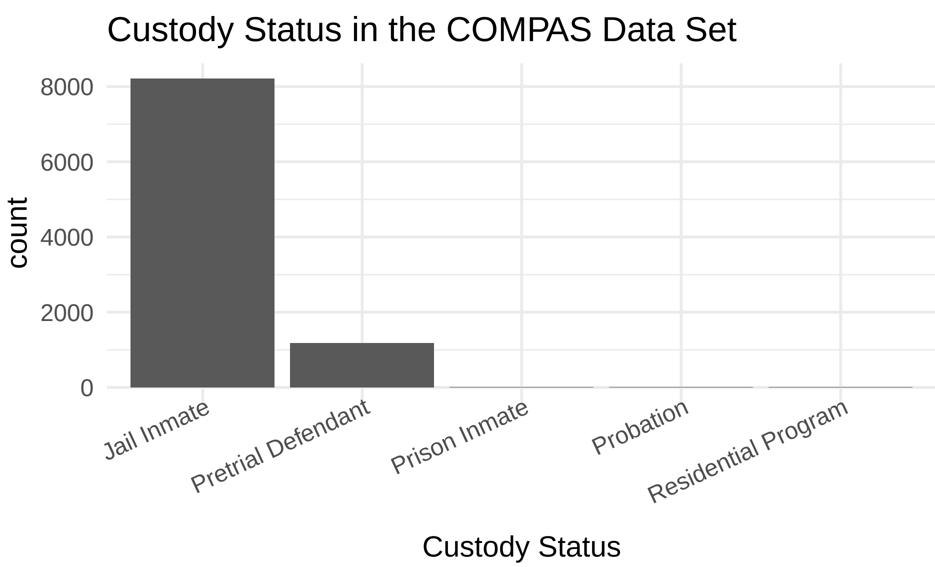


Most of the defendants are jail inmates, with only a handful of prison inmates, probationers, and defendants of the residential program.

```
tally(clean_compasdata$custody_status)
```

```
## X
##         Jail Inmate  Pretrial Defendant       Prison Inmate          Probation
##                8207                1171                   4                  3
## Residential Program
##                   2
```

```
ggplot(data = clean_compasdata, mapping = aes(x = custody_status)) +
  geom_bar() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1)) +
  labs(x = "Custody Status",
       title = "Custody Status in the COMPAS Data Set")
```



As a data check, all the assessments are for risk of recidivism.

```
tally(clean_compasdata$type_of_assessment)
```

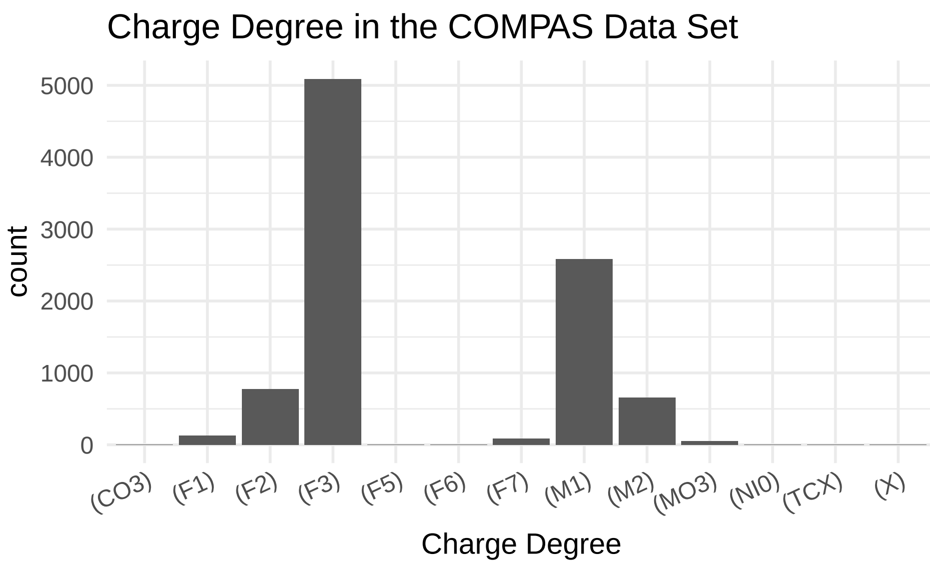
```
## X
## Risk of Recidivism
##               9387
```

There are 13 different charge degrees present in the data set. Most defendants were charged with (F3), which are felonies of the third degree. These are the least serious felonies in Florida and typically include crimes like breaking and entering, collecting and keeping stolen property, fraud, and petty theft. Many other defendants were also charged with (M1), which are a first-degree misdemeanors and can be punished by up to one year in jail. These include simple battery, disorderly conduct, DUI, indecent exposure, marijuana possession, shoplifting, prostitution, and vandalism, among others.

```
tally(clean_compasdata$c_charge_degree)
```

```
## X
## (CO3)  (F1)  (F2)  (F3)  (F5)  (F6)  (F7)  (M1)  (M2) (MO3) (NI0) (TCX)   (X)
##     1   129   774  5091     5     3    85  2584   658    51     4     1     1
```

```
ggplot(data = clean_compasdata, mapping = aes(x = c_charge_degree)) +
  geom_bar() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1)) +
  labs(x = "Charge Degree",
       title = "Charge Degree in the COMPAS Data Set")
```

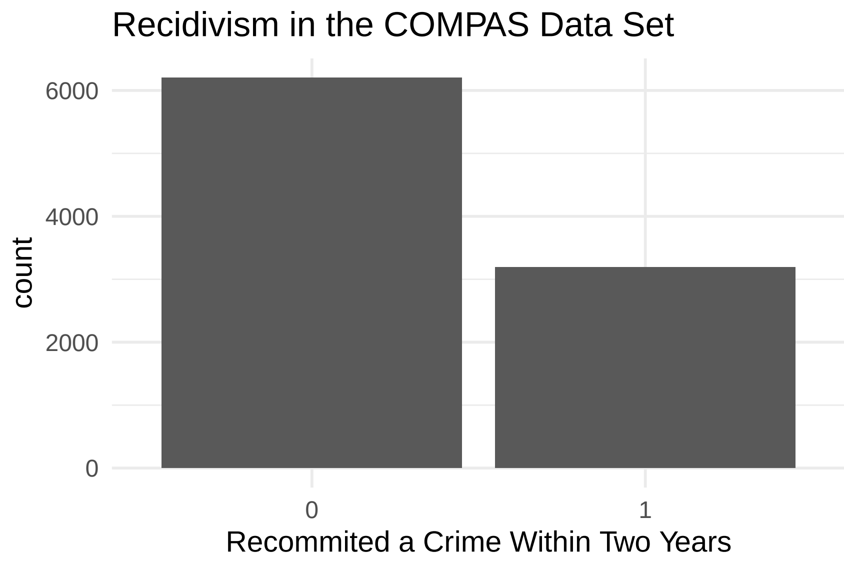


About two-thirds of the defendants did not recommit a crime within two years, while one-thirds did. This is our response variable and is indicative of class imbalance, which can affect the performance of machine learning classification algorithms. This is important to keep in mind when assessing model performance later on.

```
tally(clean_compasdata$is_recid)
```

```
## X
##    0    1
## 6199 3188
```

```
ggplot(data = clean_compasdata, mapping = aes(x = as.factor(is_recid))) +
  geom_bar() +
  theme_minimal() +
  labs(x = "Recommitted a Crime Within Two Years",
       title = "Recidivism in the COMPAS Data Set")
```



Only 745 defendants recommitted a violent crime.

```
tally(clean_compasdata$is_violent_recid)
```

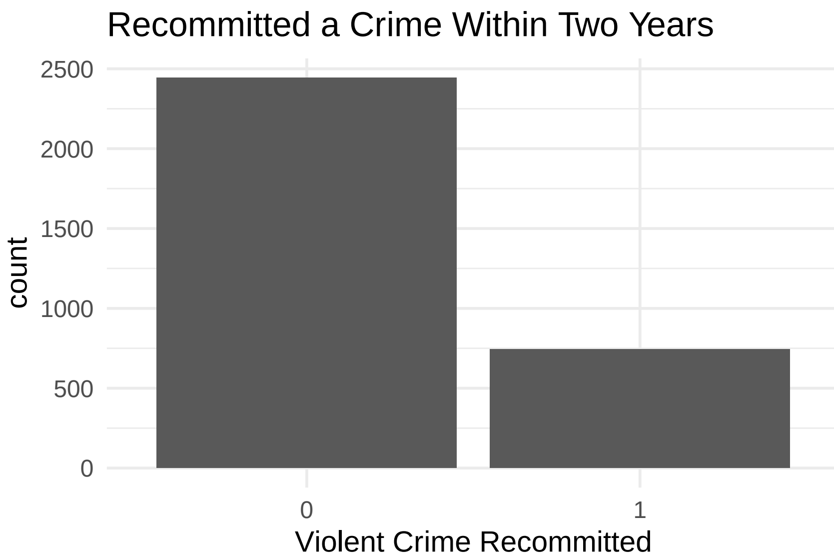
```
## X
##    0    1
## 8642  745
```

Out of the 3188 who recommitted a crime, 2443 re-committed a non-violent crime,

```
tally(clean_compasdata[clean_compasdata$is_recid == 1, ]$is_violent_recid,
      margins = TRUE)
```

```
## X
##     0     1 Total
##  2443   745  3188
```

```
ggplot(data = clean_compasdata[clean_compasdata$is_recid == 1, ],
       mapping = aes(x = as.factor(is_violent_recid))) +
  geom_bar() +
  theme_minimal() +
  labs(x = "Violent Crime Recommitted",
       title = "Recommitted a Crime Within Two Years")
```

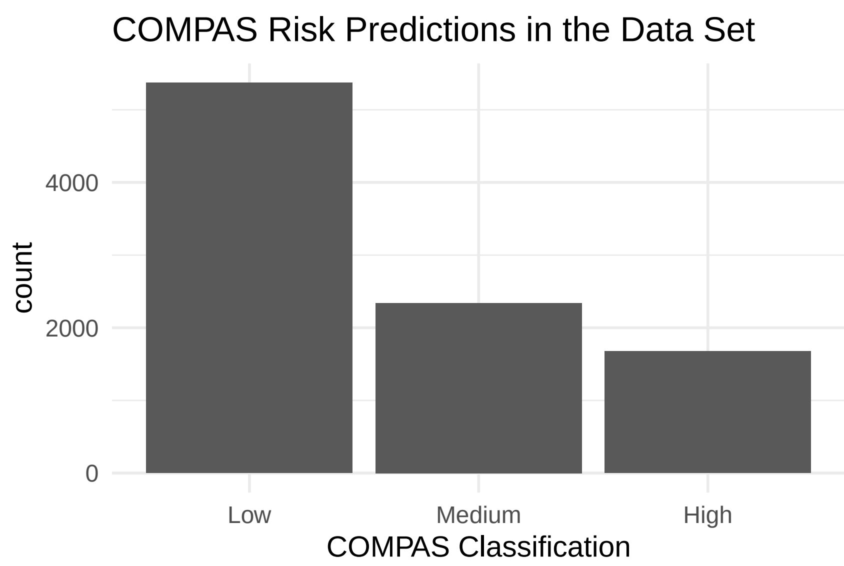


Finally, the COMPAS tool classified more than half of the defendants as low risk. In particular, 5370 were classified as low risk and 1677 as high risk, with the remaining 2340 as medium risk. This is expected since most of the defendants did not recommit a crime within the two year time window.

```
tally(clean_compasdata$score_text)
```

```
## X
##   High    Low Medium
##   1677   5370   2340
```

```
order <- c("Low", "Medium", "High")

ggplot(data = clean_compasdata, mapping = aes(x = score_text)) +
  geom_bar() +
  theme_minimal() +
  scale_x_discrete(limits = order) +
  labs(x = "COMPAS Classification",
       title = "COMPAS Risk Predictions in the Data Set")
```



This wraps up our univariate analysis of the categorical variables. Next, let's examine the univariate distribution of the continuous variables.

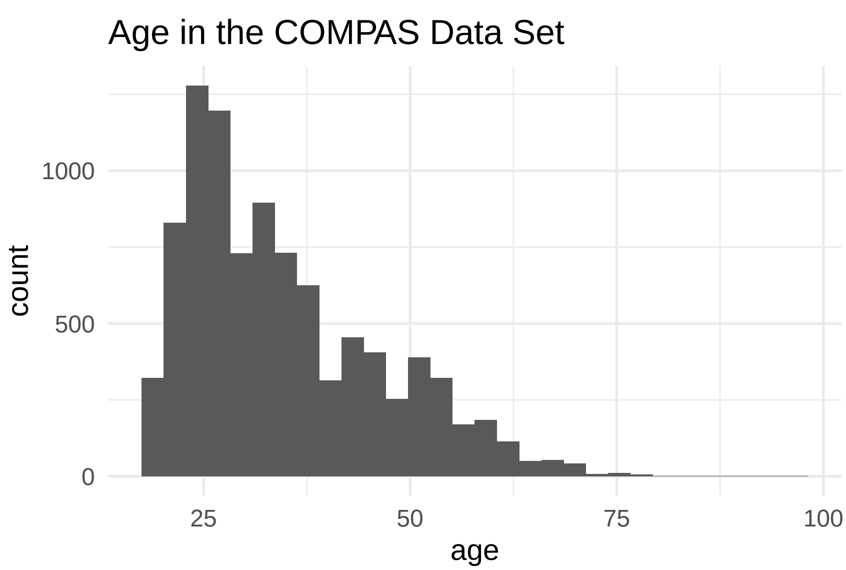
Continuous Variables

The age of the defendants ranges from 18 to 96 with a mean of 34 and a median of 32. There is no missing data. There's a right-skew in the distribution because of the few really old defendants.

```
favstats(clean_compasdata$age)
```

```
##  min Q1 median Q3 max     mean       sd    n missing
##   18 25     32 42  96 34.75413 11.80854 9387       0
```

```
ggplot(data = clean_compasdata, mapping = aes(x = age)) +
  geom_histogram() +
  theme_minimal() +
  labs(title = "Age in the COMPAS Data Set")
```



Most of the defendants had no juvenile felony accounts. The maximum juvenile felony count is 20. There is not enough variation in this variable.

```
favstats(clean_compasdata$juv_fel_count)
```

```
##  min Q1 median Q3 max       mean        sd    n missing
##    0  0      0  0  20 0.05837861 0.4518127 9387       0
```

Similarly, most defendants had no juvenile misdemeanor counts, which are less serious crimes than felonies. The maximum was 13, but there is not enough variation in this variable.

```
favstats(clean_compasdata$juv_misd_count)
```

```
##  min Q1 median Q3 max      mean        sd    n missing
##    0  0      0  0  13 0.0787259 0.4640061 9387       0
```

Similarly, most defendants had no other juvenile counts, excluding misdemeanors and felonies. The maximum was 11, but there is not enough variation in this variable.

```
favstats(clean_compasdata$juv_other_count)
```

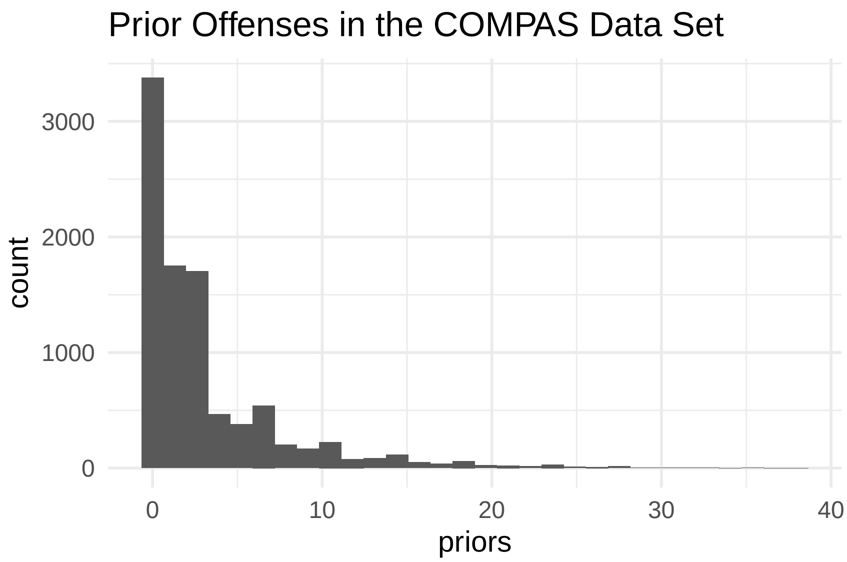
```
##  min Q1 median Q3 max       mean        sd    n missing
##    0  0      0  0  11 0.09917972 0.4683305 9387       0
```

There is slightly more variation in the `priors_count` variable which records the number of non-juvenile prior offenses for each defendant. It ranges from 0 to 38, with a median of 1 and a mean of 3.02, indicating a right skew as visualized in the histogram below. There is no missing data and the standard deviation is 4.586, suggesting that this may be a more informative variable when modeling.

```
favstats(clean_compasdata$priors_count)
```

```
##  min Q1 median Q3 max     mean       sd    n missing
##    0  0      1  4  38 3.023863 4.586441 9387       0
```

```
ggplot(data = clean_compasdata, mapping = aes(x = priors_count)) +
  geom_histogram() +
  theme_minimal() +
  labs(x = "priors",
       title = "Prior Offenses in the COMPAS Data Set")
```



The `days_b_screening_arrest` variable indicates how many days passed between arrest and COMPAS screening. It may not be indicative of recidivism, however. We will evaluate this when performing bivariate analysis.

```
favstats(clean_compasdata$days_b_screening_arrest)
```

```
##  min Q1 median Q3 max     mean      sd    n missing
##    0  1      1  1  30 2.140194 4.89312 9387       0
```

The interpretation of this variable is not clear – it seems to indicate the number of days since COMPAS screening to date. We will not include this in the analysis.

```
favstats(clean_compasdata$c_days_from_compas)
```

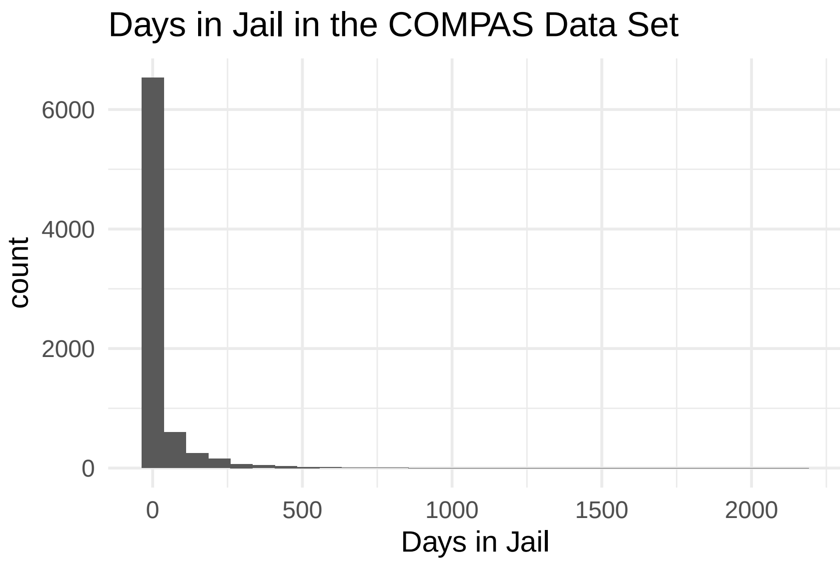
```
##  min Q1 median Q3  max     mean       sd    n missing
##    0  1      1  1 9485 24.92436 263.4065 9387       0
```

Considering only the observations for which we have data, the number of days spent in jail ranges from 0 to 2154 days (5.9 years), with a median of 2 days and a mean of 29 days. This variable is extremely right skewed, as visualized in the histogram. The standard deviation is also 82.9, indicating a lot of variation that may potentially be useful for predicting the risk of recidivism.

```
favstats(clean_compasdata$days_in_jail)
```

```
##  min Q1 median Q3  max     mean      sd    n missing
##    0  1      2 13 2154 28.96933 82.8563 7727    1660
```

```
ggplot(data = clean_compasdata, mapping = aes(x = days_in_jail)) +
  geom_histogram() +
  theme_minimal() +
  labs(x = "Days in Jail",
       title = "Days in Jail in the COMPAS Data Set")
```

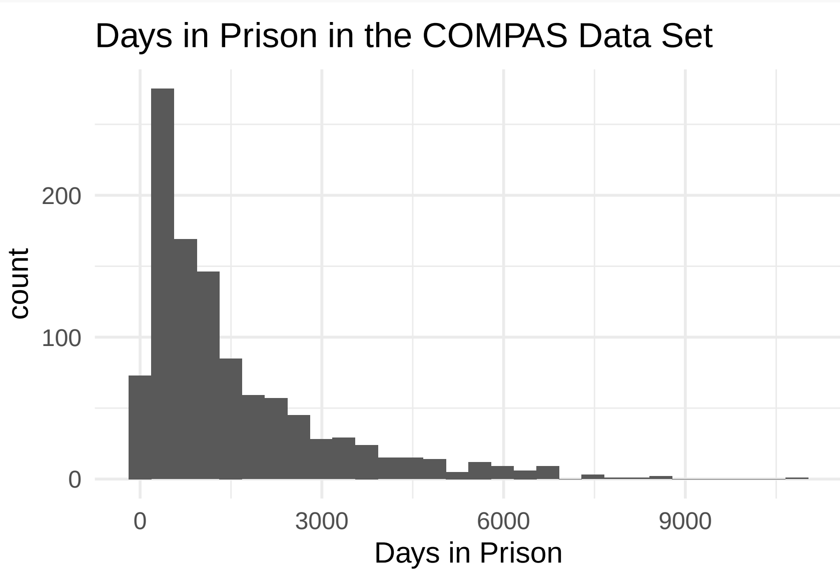


Considering only the observations for which we have data, the number of days spent in prison ranges from 0 to 10840 days (29.6 years), with a median of 1005 days (2.8 years) and a mean of 1534 days (4.2 years). This variable is quite right skewed, as visualized in the histogram. The standard deviation is also 1554.728, indicating a lot of variation that may potentially be useful for predicting the risk of recidivism.

```
favstats(clean_compasdata$days_in_prison)
```

```
##  min     Q1 median     Q3   max     mean       sd    n missing
##    0 445.5   1005 2102.5 10840 1534.246 1554.728 1083    8304
```

```
ggplot(data = clean_compasdata, mapping = aes(x = days_in_prison)) +
  geom_histogram() +
  theme_minimal() +
  labs(x = "Days in Prison",
       title = "Days in Prison in the COMPAS Data Set")
```



Notice, however, that there is a lot of missing data for the days_in_jail and days_in_prison variables, with the former having 1660 missing observations and the latter having 8304 missing observations. Let's sum both columns up and create a new variable, days_in_jail_or_prison, to assess the impact on the amount of missing observations.

```
clean_compasdata <- clean_compasdata %>%
  rowwise() %>%
  mutate(days_in_jail_or_prison = sum(days_in_jail, days_in_prison, na.rm=TRUE)) %>%
  ungroup()
```

There are 1732 defendants with either no jail or prison history, or that spent less than half a day in jail or prison.

```
count(clean_compasdata$days_in_jail_or_prison == 0)
```

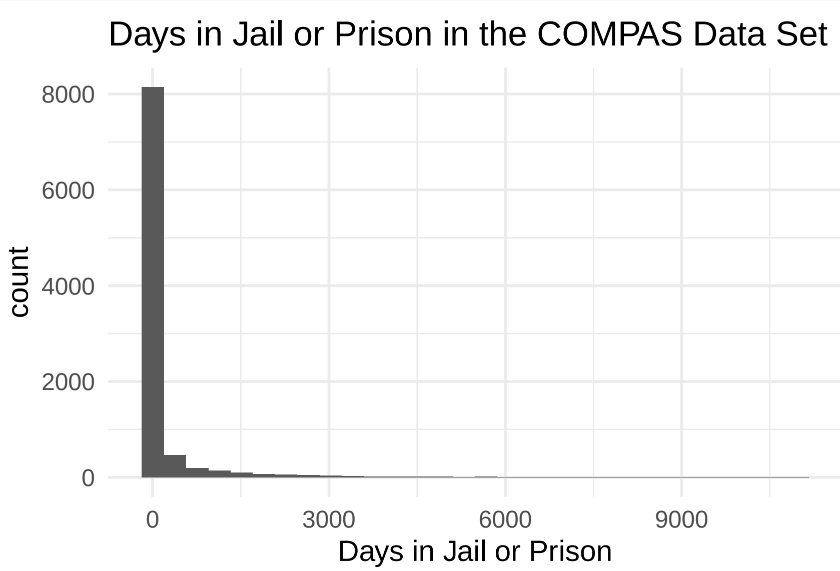
```
## n_TRUE
##   1732
```

The mean of this variable is now 200 days, with a median of 1 day, indicating an extreme right skew. There is no missing data, however.

```
favstats(clean_compasdata$days_in_jail_or_prison)
```

```
##  min Q1 median Q3   max     mean       sd    n missing
##    0  1      1 19 10973 200.8559 739.5814 9387       0
```

```
ggplot(data = clean_compasdata, mapping = aes(x = days_in_jail_or_prison)) +
  geom_histogram() +
  theme_minimal() +
  labs(x = "Days in Jail or Prison",
       title = "Days in Jail or Prison in the COMPAS Data Set")
```



The number of crimes recommitted by the defendants who re-committed a crime within two years ranges from 1 to 55, with a median of 1 and a mean of 1.73.

```
favstats(clean_compasdata$num_r_cases)
```

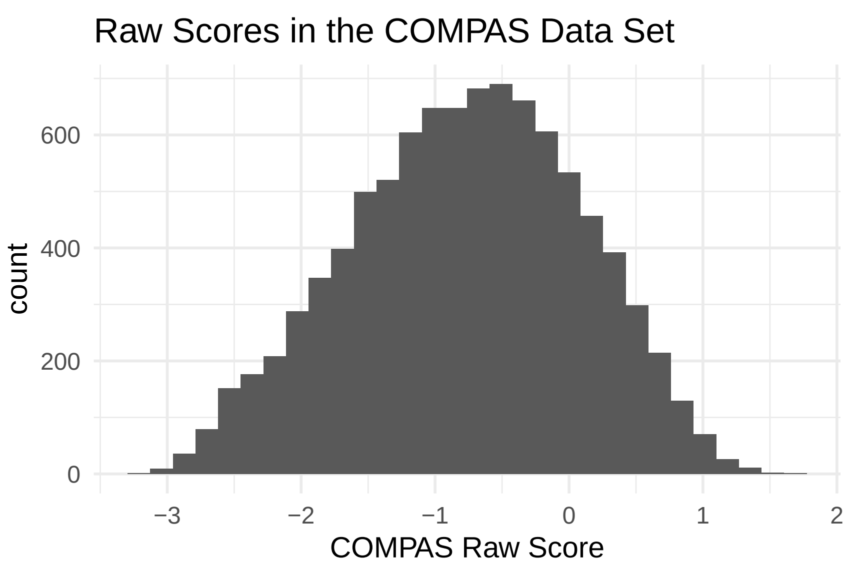
```
##  min Q1 median Q3 max     mean       sd    n missing
##    1  1      1  2  55 1.736512 1.629916 3188    6199
```

Finally, the COMPAS tool outputs a raw score for each defendant. The raw score ranges from -3.21 to 1.69 with a median of -0.74 and a mean of -0.78. The distribution of the raw scores is visualized on the histogram below. The distribution is unimodal and symmetric with a slight left skew.

```
favstats(clean_compasdata$raw_score)
```

```
##    min    Q1 median    Q3  max       mean        sd    n missing
##  -3.21 -1.38  -0.74 -0.15 1.69 -0.7759146 0.8573394 9387       0
```

```
ggplot(data = clean_compasdata, mapping = aes(x = raw_score)) +
  geom_histogram() +
  theme_minimal() +
  labs(x = "COMPAS Raw Score",
       title = "Raw Scores in the COMPAS Data Set")
```



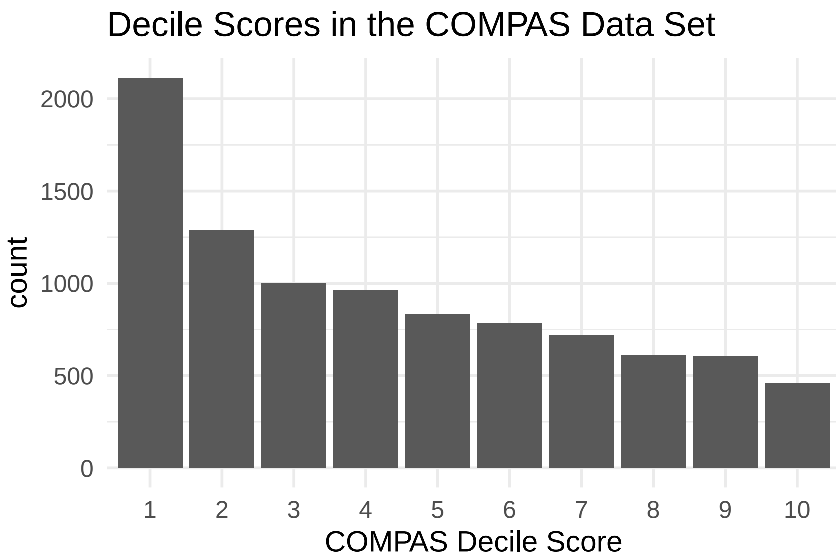
The raw scores are then converted into decile scores that determine the predicted risk of recidivism. The decile scores range from 1 to 10 with a median of 4 and a mean of 4.3. The histogram displays the distribution of the decile scores – it makes me wonder how, or whether, the decile scores are computed from the raw scores.

```
favstats(clean_compasdata$decile_score)
```

```
##  min Q1 median Q3 max     mean       sd    n missing
##    1  2      4  7  10 4.305849 2.849011 9387       0
```

```
ggplot(data = clean_compasdata, mapping = aes(x = as.factor(decile_score))) +
  geom_bar() +
```

```r
  theme_minimal() +
  labs(x = "COMPAS Decile Score",
       title = "Decile Scores in the COMPAS Data Set")
```



There is significant overlap between the decile scores and the raw scores. It's unclear how exactly these scores were mapped. However, because the decile score is what is handed to judges at court and what determines the risk of recidivism, we will proceed with that variable.

```r
favstats(clean_compasdata$raw_score ~ clean_compasdata$decile_score)
```

```
##    clean_compasdata$decile_score   min    Q1 median      Q3   max        mean
## 1                              1 -3.21 -2.21  -1.89 -1.6500 -1.39 -1.95135289
## 2                              2 -1.65 -1.35  -1.23 -1.1100 -1.01 -1.24964286
## 3                              3 -1.30 -1.00  -0.91 -0.8300 -0.75 -0.94058765
## 4                              4 -1.02 -0.74  -0.66 -0.5800 -0.40 -0.69193983
## 5                              5 -0.81 -0.51  -0.44 -0.3800 -0.32 -0.48439521
## 6                              6 -0.58 -0.31  -0.25 -0.1900 -0.02 -0.28100637
## 7                              7 -0.36 -0.11  -0.05  0.0125  0.19 -0.06243056
## 8                              8 -0.13  0.11   0.17  0.2400  0.33  0.16462541
## 9                              9  0.13  0.35   0.42  0.5200  0.65  0.42686469
## 10                            10  0.44  0.68   0.78  0.9200  1.69  0.81579869
##            sd    n missing
## 1  0.38018237 2114       0
## 2  0.16522602 1288       0
## 3  0.14595631 1004       0
## 4  0.13642777  964       0
## 5  0.14094393  835       0
## 6  0.12481877  785       0
## 7  0.10487589  720       0
## 8  0.09722801  614       0
## 9  0.11202009  606       0
## 10 0.19736051  457       0
```

Note that the decile scores are mapped to 'low', 'medium', and 'high' risk as detailed in the table below.

```r
clean_compasdata %>%
  dplyr::select(decile_score, score_text) %>%
  filter(score_text != 'N/A') %>%
  rename("Risk" = score_text) %>%
  group_by(Risk) %>%
  summarise("Min" = min(decile_score),
            "Max" = max(decile_score)) %>%
  arrange(Min) %>%
  kable(booktabs = TRUE)
```

Risk	Min	Max
Low	1	4
Medium	5	7
High	8	10

Adding Some New Variables

Based on the univariate analysis in this section, we will create 2 more columns:

- `juv_offense_count`, which is a continuous variable that records how many total prior juvenile offenses a defendant had.
- `juv_offense`, which is a binary variable that records whether or not a defendant had a juvenile offense (0 = 'No'; 1 = 'Yes').

Combining the information from multiple non-informative variables may create more informative variables.

```r
clean_compasdata <- clean_compasdata %>%
  mutate(juv_offense_count = juv_fel_count + juv_misd_count + juv_other_count,
         juv_offense = ifelse(juv_offense_count == 0, 0, 1))
```

Let's assess the univariate distribution of these new variables.

`juv_offense_count` variable is still highly invariable, with at least $\frac{3}{4}$ of observations have no juvenile offenses.

```r
favstats(clean_compasdata$juv_offense_count)
```

```
##  min Q1 median Q3 max      mean        sd    n missing
##    0  0      0  0  21 0.2362842 0.9082055 9387       0
```

We see the same trends for `juv_offense`, the categorical variable, with only 1133 defendants reporting a juvenile offense.

```r
tally(clean_compasdata$juv_offense)
```

```
## X
##    0    1
## 8254 1133
```

We'll keep the continuous variable, which may have slightly more quantifiable information than the binary variable.

This concludes our univariate analysis of the variables in the COMPAS data set. Next, we will look at some of the bivariate relationships.

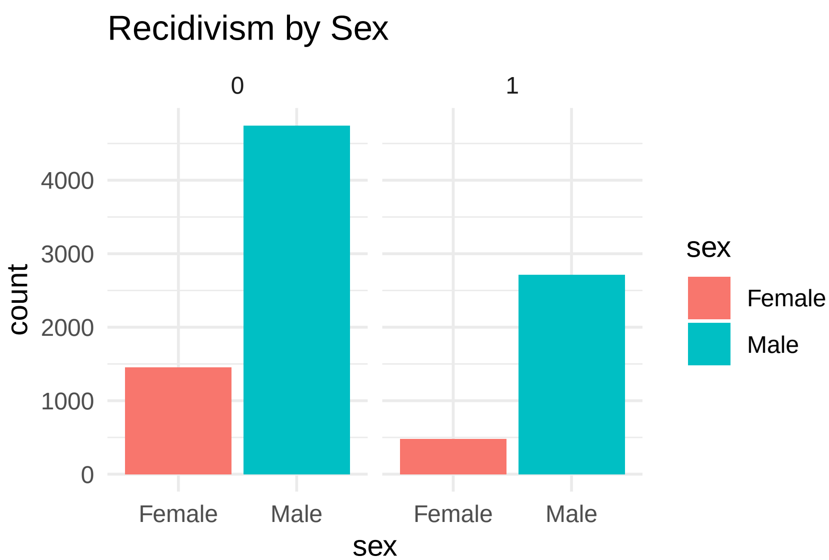
Bivariate Analysis

In this section, we will explore the relationships between our variables and the response variable, `is_recid`, which records whether or not a defendant recommitted a crime within 2 years.

Categorical Variables

It doesn't appear as though there is much evident relationship between sex and recidivism.

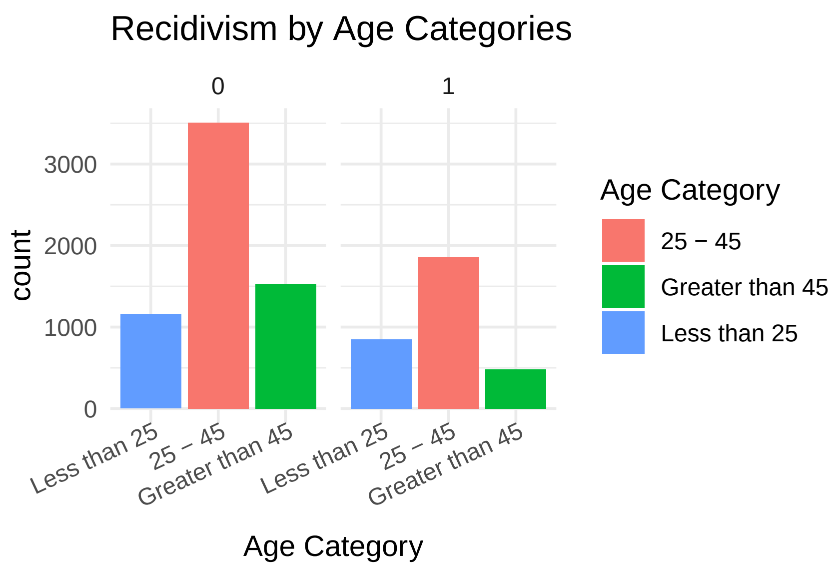
```
ggplot(data = clean_compasdata, mapping = aes(x = sex, fill = sex)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~is_recid) +
  labs(title = "Recidivism by Sex")
```



Among defendants who do not recidivate, there are more defendants that are aged 45 in comparison to those less than 25. However, among those that recidivated, there are more defendants that are less than 25 in comparison to those that are greater than 45. This indicates that age may hold some valuable information regarding a defendant's likelihood of recidivism.

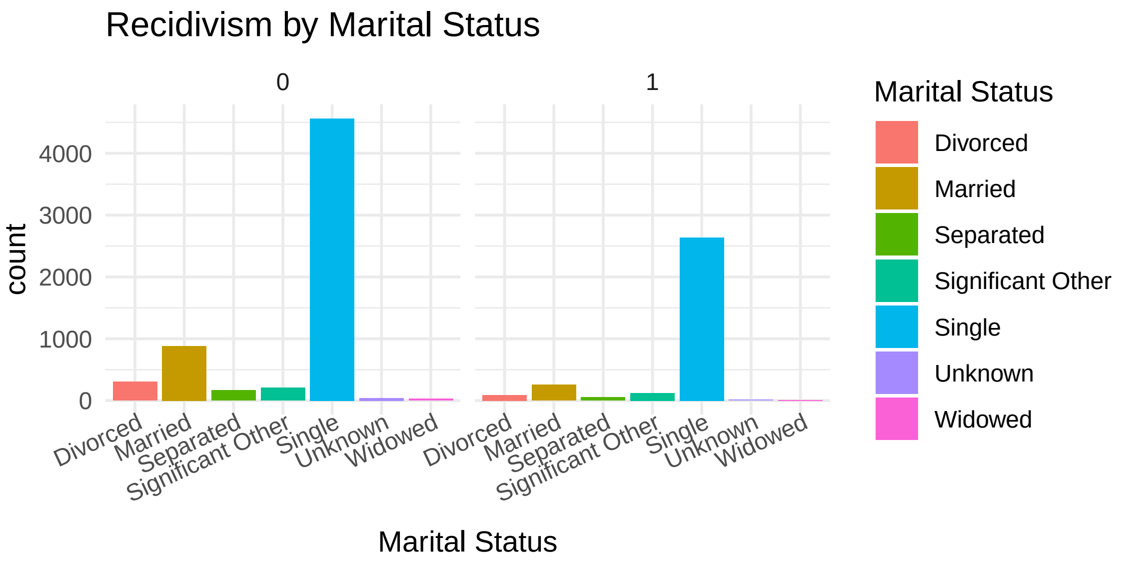
```
order <- c("Less than 25", "25 - 45", "Greater than 45")

ggplot(data = clean_compasdata,
       mapping = aes(x = age_cat, fill = age_cat)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~is_recid) +
```

```r
  labs(title = "Recidivism by Age Categories",
       x = "Age Category",
       fill = "Age Category") +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1)) +
  scale_x_discrete(limits = order)
```



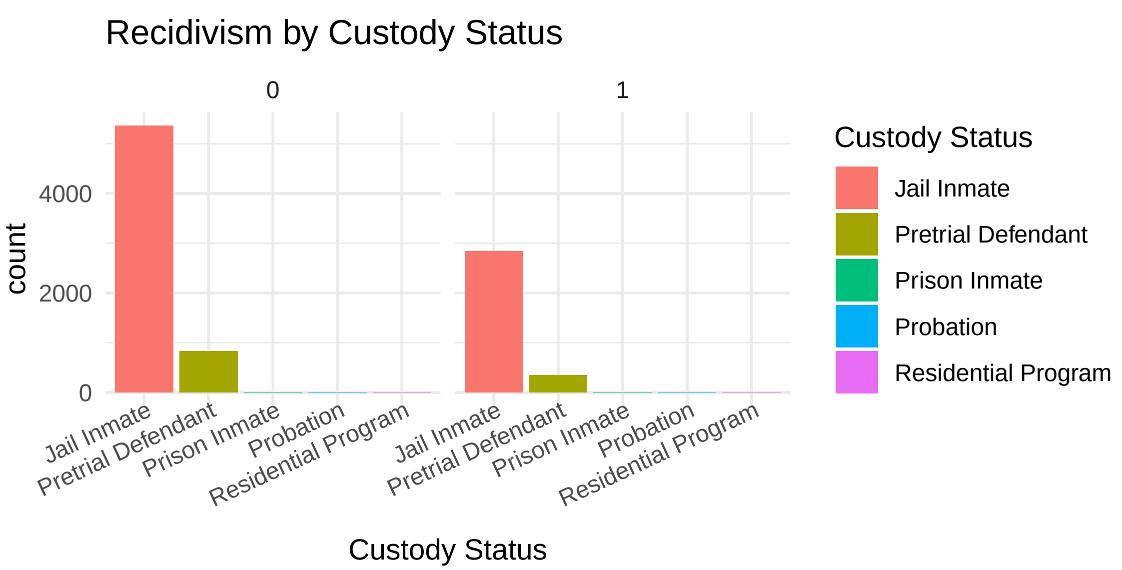
It doesn't appear as though there is much relationship between recidivism and marital status.

```r
ggplot(data = clean_compasdata,
       mapping = aes(x = marital_status, fill = marital_status)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~is_recid) +
  labs(title = "Recidivism by Marital Status",
       x = "Marital Status",
       fill = "Marital Status") +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1))
```



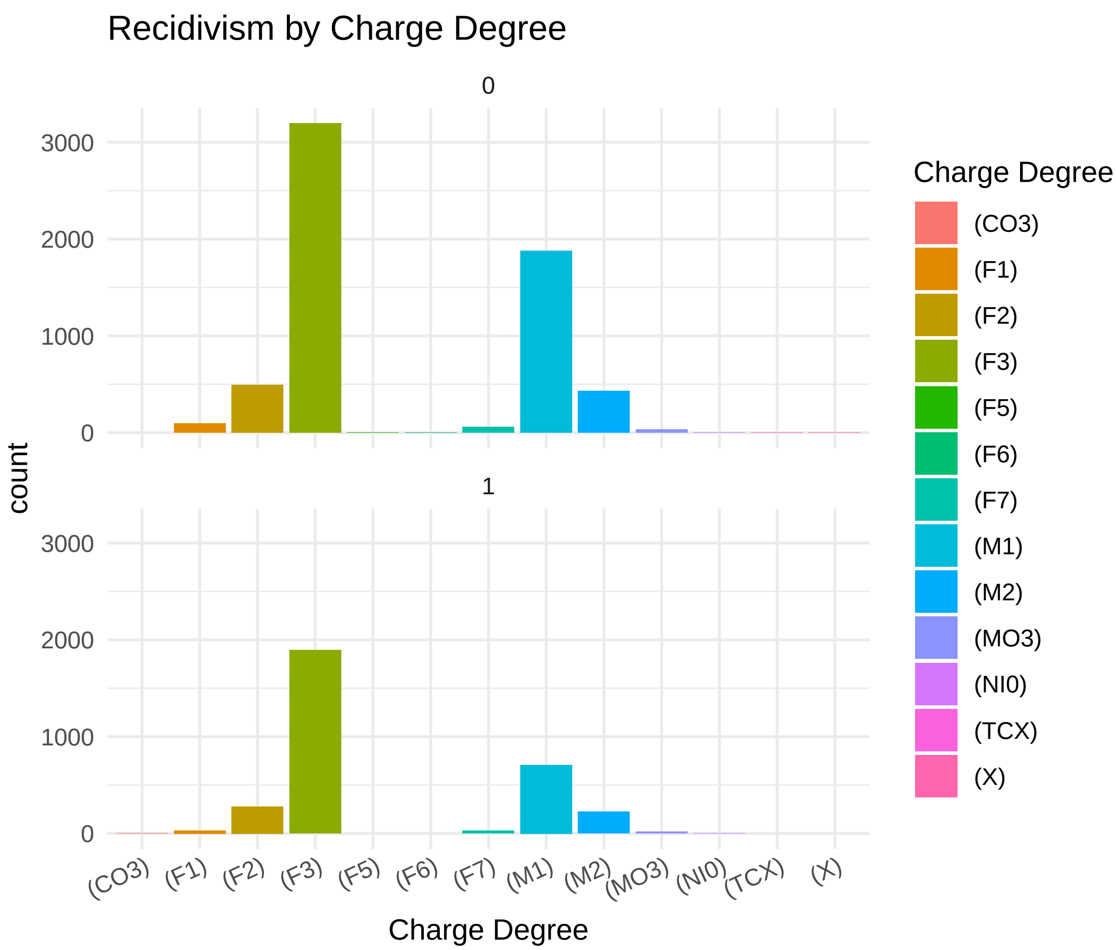
It doesn't appear as though there is much relationship between recidivism and custody status.

```r
ggplot(data = clean_compasdata,
       mapping = aes(x = custody_status, fill = custody_status)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~is_recid) +
  labs(title = "Recidivism by Custody Status",
       x = "Custody Status",
       fill = "Custody Status") +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1))
```



It doesn't appear as though there is much relationship between recidivism and charge degree.

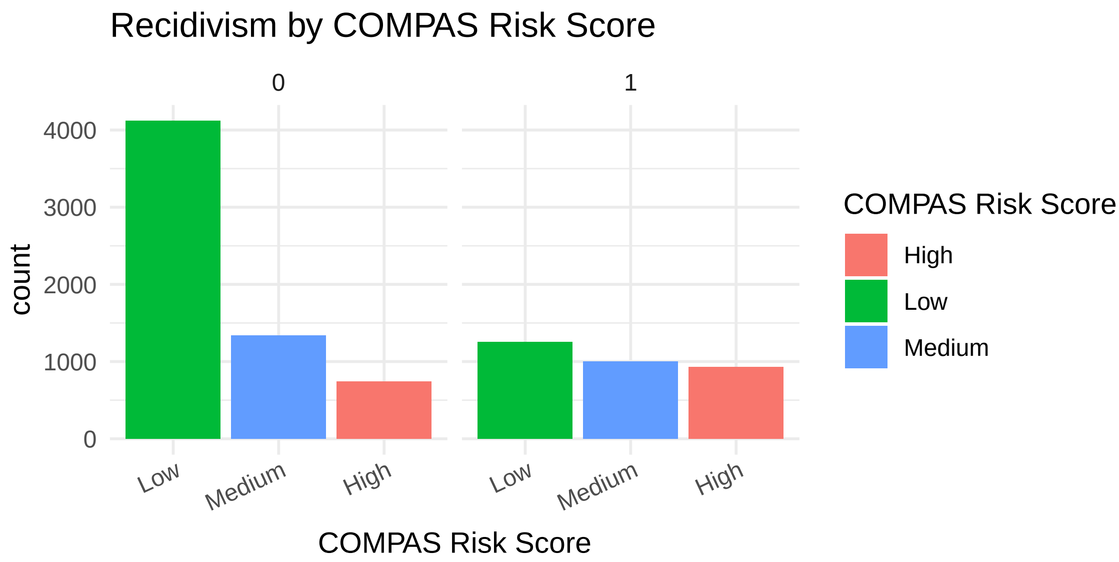
```r
ggplot(data = clean_compasdata,
       mapping = aes(x = c_charge_degree, fill = c_charge_degree)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~is_recid, ncol = 1) +
  labs(title = "Recidivism by Charge Degree",
       x = "Charge Degree",
       fill = "Charge Degree") +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1))
```



However, it appears as though the COMPAS tool classifies defendants who recommit a crime as almost as equally risky of recidivism – there is no significant distinction between 'low', 'medium', and 'high' risk for these defendants. For the defendants that don't recommit a crime, most are predicted as 'low' risk, followed by 'medium', and then 'high' risk. Note, however, that this variable will not be included as a predictor in the model as the purpose of this analysis is to assess COMPAS performance, or more generally, standard ML approaches, in comparison to the Seldonian framework.

```r
order <- c("Low", "Medium", "High")

ggplot(data = clean_compasdata,
       mapping = aes(x = score_text, fill = score_text)) +
  geom_bar() +
```

```r
  theme_minimal() +
  facet_wrap(~is_recid) +
  labs(title = "Recidivism by COMPAS Risk Score",
       x = "COMPAS Risk Score",
       fill = "COMPAS Risk Score") +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1)) +
  scale_x_discrete(limits = order)
```



Next, let's perform a similar analysis for the continuous variables.

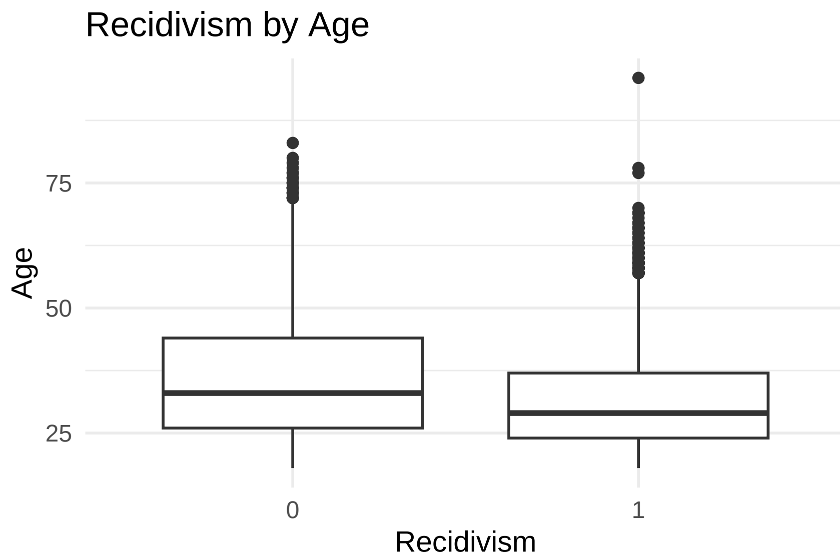
Continous Variables

There is a difference in the mean and median ages for defendants who recommit a crime within two years versus though who don't. Those who recidivate tend to be younger than those who don't, indicating that this will be a useful variable in the model. This is in line with intuition from society.

```r
favstats(data = clean_compasdata, age ~ is_recid)
```

```
##   is_recid min Q1 median Q3 max     mean       sd    n missing
## 1        0  18 26     33 44  83 36.04646 12.17525 6199       0
## 2        1  18 24     29 37  96 32.24122 10.62147 3188       0
```

```r
ggplot(data = clean_compasdata,
       mapping = aes(x = as.factor(is_recid), y = age)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Recidivism by Age",
       x = "Recidivism",
       y = "Age")
```



There is not much distributional difference in juvenile felony counts for defendants who recidivate versus those who don't.

```
favstats(data = clean_compasdata, juv_fel_count ~ is_recid)
```

```
##   is_recid min Q1 median Q3 max       mean        sd    n missing
## 1        0   0  0      0  0  13 0.03645749 0.3297061 6199       0
## 2        1   0  0      0  0  20 0.10100376 0.6221204 3188       0
```

There is not much distributional difference in juvenile misdemeanor counts for defendants who recidivate versus those who don't.

```
favstats(data = clean_compasdata, juv_misd_count ~ is_recid)
```

```
##   is_recid min Q1 median Q3 max       mean        sd    n missing
## 1        0   0  0      0  0  12 0.04387804 0.3356002 6199       0
## 2        1   0  0      0  0  13 0.14648683 0.6388211 3188       0
```

There is not much distributional difference in juvenile offenses for defendants who recidivate versus those who don't.

```
favstats(data = clean_compasdata, juv_other_count ~ is_recid)
```

```
##   is_recid min Q1 median Q3 max       mean        sd    n missing
## 1        0   0  0      0  0  11 0.06355864 0.3966132 6199       0
## 2        1   0  0      0  0   9 0.16844417 0.5768642 3188       0
```

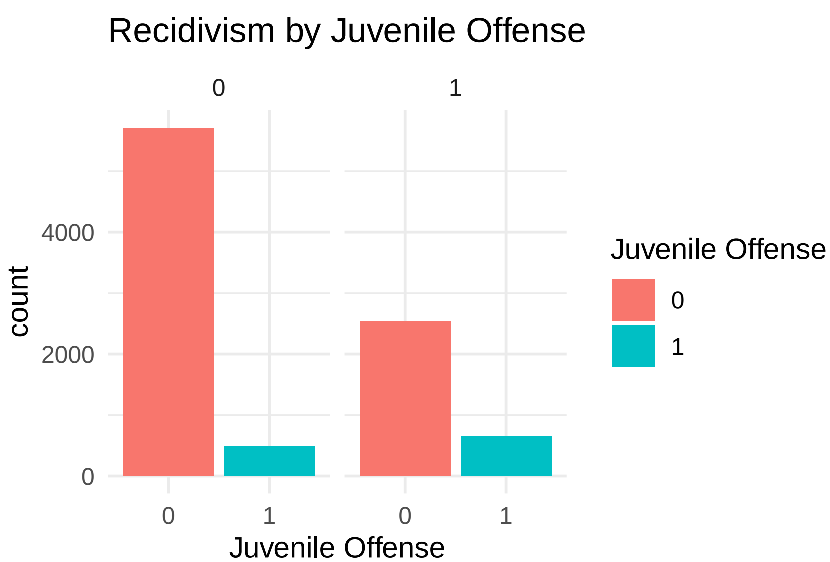
As expected, there is not much distributional difference in the number of juvenile offenses for defendants who do not recidivate versus those who do.

```r
favstats(data = clean_compasdata, juv_offense_count ~ is_recid)
```

```
##   is_recid min Q1 median Q3 max      mean        sd    n missing
## 1        0   0  0      0  0  21 0.1438942 0.7044374 6199       0
## 2        1   0  0      0  0  20 0.4159348 1.1896519 3188       0
```

However, the bar plot below reveals that while most defendants don't have juvenile offenses, the proportion of those who don't to those who do is much smaller for defendants that re-offend in comparison to those that don't. This may be a useful variable to include in the model.

```r
ggplot(data = clean_compasdata,
       mapping = aes(x = as.factor(juv_offense), fill = as.factor(juv_offense))) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~is_recid) +
  labs(title = "Recidivism by Juvenile Offense",
       x = "Juvenile Offense",
       fill = "Juvenile Offense")
```



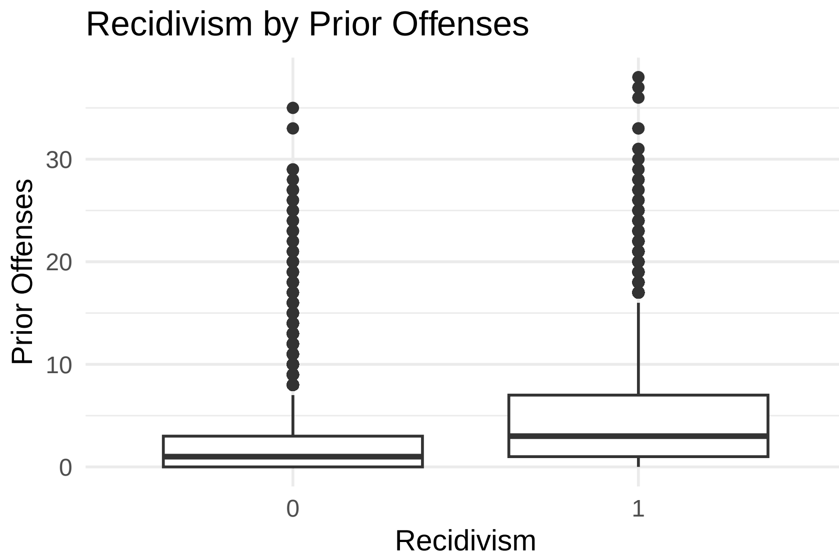
There is some distributional difference in non-juvenile prior offenses for defendants who recidivate versus those who don't, as is indicated by the different means and medians. Those who recommit a crime within two years tend to have more prior offenses. This will be a useful variable to include in the models.

```r
favstats(data = clean_compasdata, priors_count ~ is_recid)
```

```
##   is_recid min Q1 median Q3 max     mean       sd    n missing
## 1        0   0  0      1  3  35 2.157283 3.684641 6199       0
## 2        1   0  1      3  7  38 4.708908 5.589893 3188       0
```

```r
ggplot(data = clean_compasdata,
       mapping = aes(x = as.factor(is_recid), y = priors_count)) +
  geom_boxplot() +
  theme_minimal() +
```

```r
  labs(title = "Recidivism by Prior Offenses",
       x = "Recidivism",
       y = "Prior Offenses")
```



There doesn't appear to be any distributional difference in days between COMPAS screening and arrest for defendants who recidivate versus those who don't. This will not be a useful variable for modeling.

```r
favstats(data = clean_compasdata, days_b_screening_arrest ~ is_recid)
```

```
##   is_recid min Q1 median Q3 max     mean       sd    n missing
## 1        0   0  1      1  1  30 2.149218 4.859912 6199       0
## 2        1   0  1      1  1  30 2.122647 4.957777 3188       0
```

While the means differ because of the right-skew nature of the data, there doesn't appear to be much distributional difference in days since COMPAS screening for defendants who recidivate versus those who don't. This will not be a useful variable for modeling.

```r
favstats(data = clean_compasdata, c_days_from_compas ~ is_recid)
```

```
##   is_recid min Q1 median Q3  max     mean       sd    n missing
## 1        0   0  1      1  1 9485 31.21052 305.1847 6199       0
## 2        1   0  1      1  1 5450 12.70107 151.5946 3188       0
```

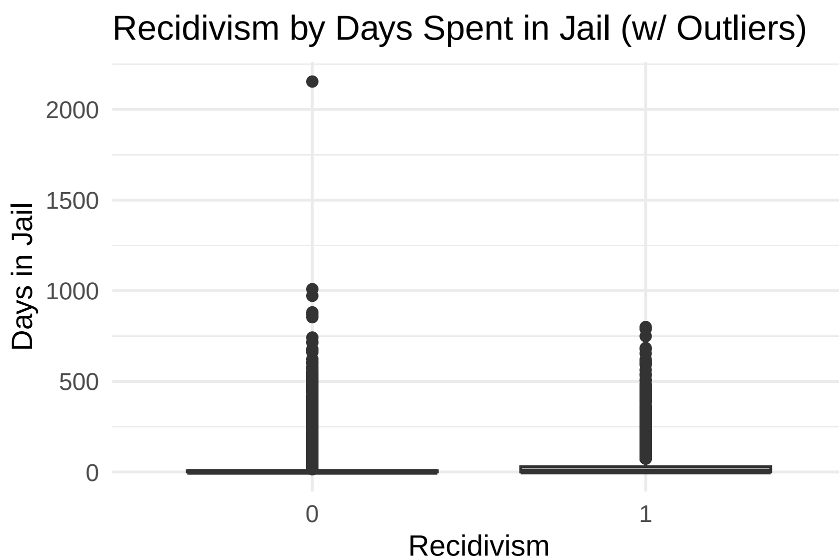
There is an evident difference in the distribution of the number of days spent in jail for participants who recommit a crime within two years versus those who don't. The mean and medians differ, with those who recommit spending more time in jail on average, indicating a distinction that may be useful in modeling. It's hard to visualize the boxplots with all the outliers, so the second boxplot trims the y-axis to better visualize this relationship.

```r
favstats(data = clean_compasdata, days_in_jail ~ is_recid)
```

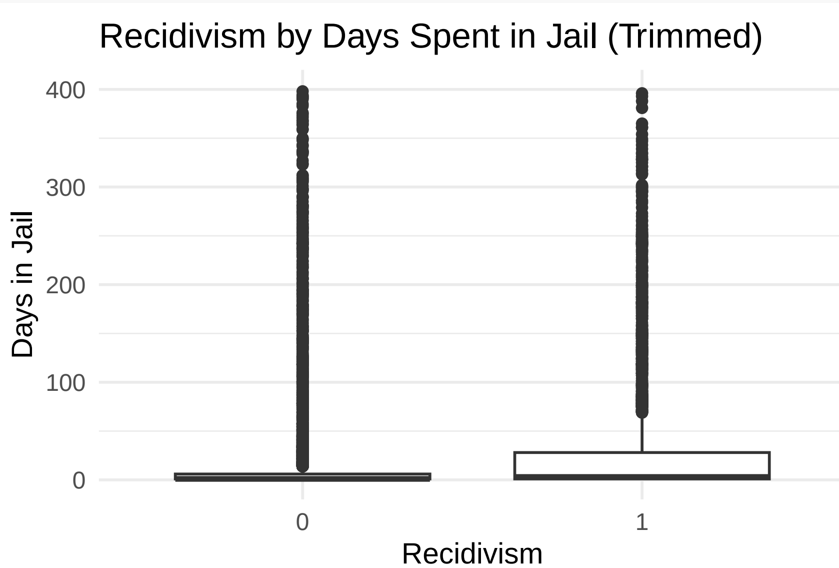
```
##   is_recid min Q1 median Q3  max     mean       sd    n missing
```

```
## 1         0    0   1       1   7 2154 25.52887 83.46318 4988     1211
## 2         1    0   1       3  30   800 35.23476 81.38177 2739      449
```

```r
ggplot(data = clean_compasdata,
       mapping = aes(x = as.factor(is_recid), y = days_in_jail)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Recidivism by Days Spent in Jail (w/ Outliers)",
       x = "Recidivism",
       y = "Days in Jail")
```




```r
ggplot(data = clean_compasdata,
       mapping = aes(x = as.factor(is_recid), y = days_in_jail)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Recidivism by Days Spent in Jail (Trimmed)",
       x = "Recidivism",
       y = "Days in Jail") +
  ylim(0,400)
```



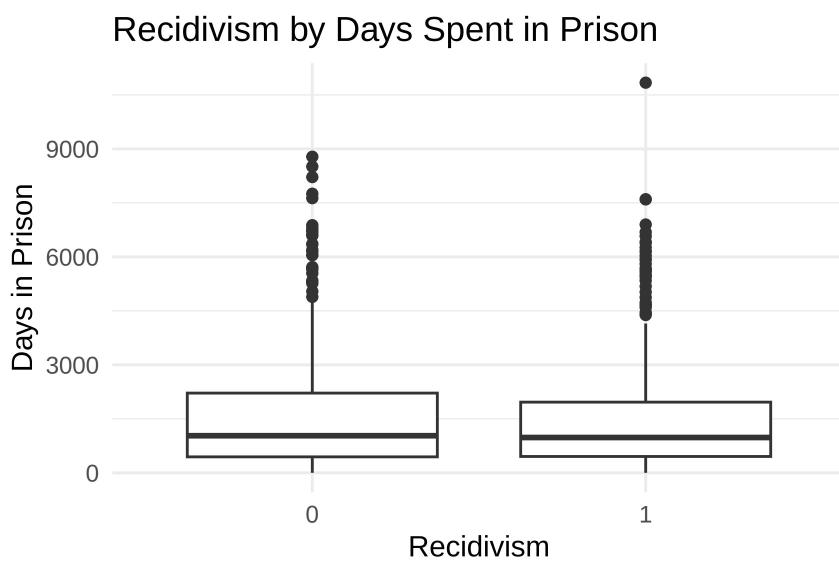
Similarly, there is some distributional difference in the days spent in prison, although in the opposite direction that might be expected and not as extreme as for the days spent in jail. Defendants who don't re-offend spend more days in prison, on average.

The difference between jail and prison is still not clear, however.

```
favstats(data = clean_compasdata, days_in_prison ~ is_recid)
```

```
##   is_recid min     Q1 median      Q3   max     mean       sd   n missing
## 1        0   0 444.00   1031 2216.00  8783 1583.826 1590.026 535    5664
## 2        1   0 454.75    981 1964.75 10840 1485.841 1519.368 548    2640
```

```
ggplot(data = clean_compasdata,
       mapping = aes(x = as.factor(is_recid), y = days_in_prison)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Recidivism by Days Spent in Prison",
       x = "Recidivism",
       y = "Days in Prison")
```



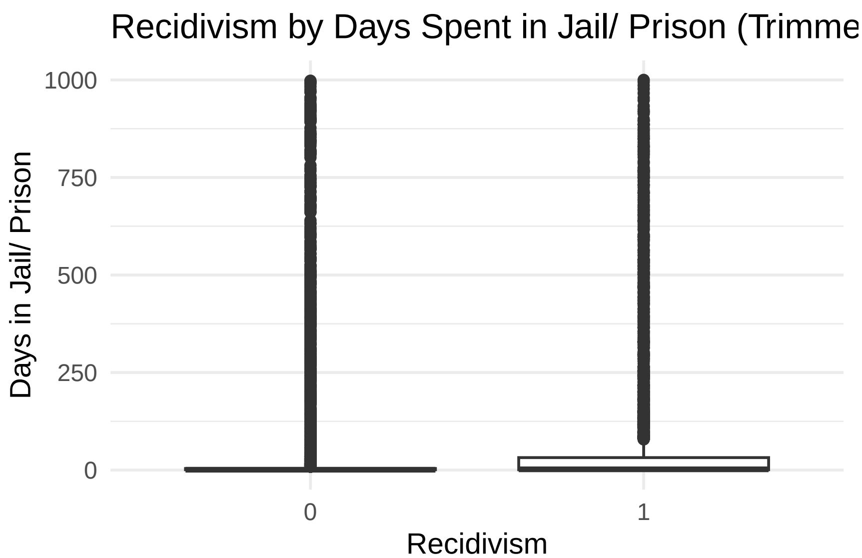
Using the new variable we created, we observe that defendants who reoffend spend, on average, more days in jail or prison. This is a useful variable that may aid our analysis. The boxplot below visualizes the distributional difference, although it is quite difficult to visually assess.

```
favstats(data = clean_compasdata, days_in_jail_or_prison ~ is_recid)
```

```
##   is_recid min Q1 median    Q3   max     mean       sd    n missing
## 1        0   0  1      1  6.00  8956 157.2326 668.0901 6199       0
## 2        1   0  1      3 66.25 10973 285.6804 855.5516 3188       0
```

```
ggplot(data = clean_compasdata,
       mapping = aes(x = as.factor(is_recid), y = days_in_jail_or_prison)) +
  geom_boxplot() +
  theme_minimal() +
```

```r
  labs(title = "Recidivism by Days Spent in Jail/ Prison (Trimmed)",
       x = "Recidivism",
       y = "Days in Jail/ Prison") +
  ylim(0, 1000)
```

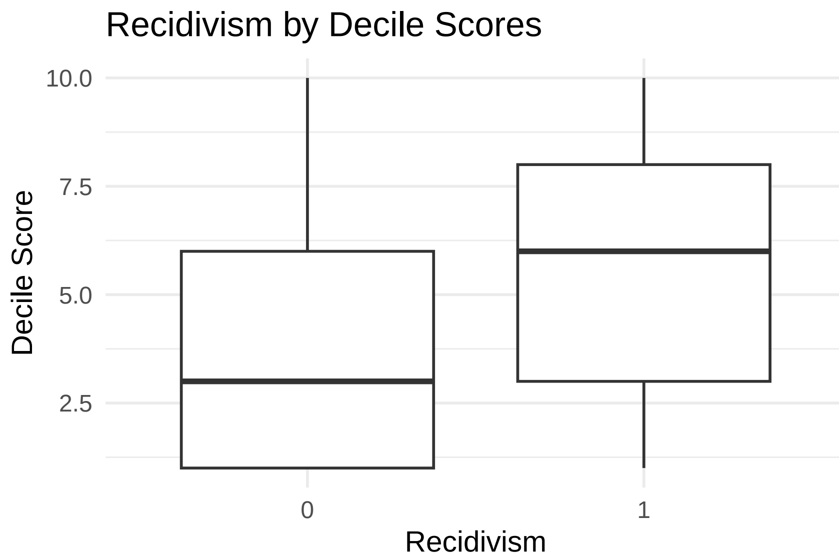


Finally, let's assess the COMPAS decile scores. The median and mean decile scores differ for defendants who recommit a crime within 2 years versus those who don't. The median score for those who don't is 3, which is mapped to low risk. The median score for those who do is 6, which is mapped to medium risk. This indicates that the COMPAS tool has some predictive accuracy. However, the range of scores is the same for both defendants who recidivate versus those who do not, suggesting that the tool is not entirely accurate in its predictions.

```r
favstats(data = clean_compasdata, decile_score ~ is_recid)
```

```
##   is_recid min Q1 median Q3 max     mean       sd    n missing
## 1        0   1  1      3  6  10 3.694467 2.667049 6199       0
## 2        1   1  3      6  8  10 5.494668 2.816137 3188       0
```

```r
ggplot(data = clean_compasdata,
       mapping = aes(x = as.factor(is_recid), y = decile_score)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Recidivism by Decile Scores",
       x = "Recidivism",
       y = "Decile Score")
```



This wraps up our analysis of the bivariate relationships between the continuous variables in the data set and the response variable: `is_recid`.

Multivariate Analysis

Based on the univariate and bivariate analysis, the 9 most informative predictive variables for modeling will be:

- `sex`
- `age`
- `age category`
- `marital status`
- `custody status`
- `juvenile offense` (binary)
- `prior offenses`
- `charge degree`
- `days in jail or prison`

We will also include `race` in the modeling data set as our demographic variable, though it will not be included in the models themselves. Finally, `is_recid`, the response variable, will also be selected in the data set.

For further analysis, we will also include the COMPAS decile scores to assess which of these variables may have been used to model the COMPAS risk assessment tool.

Now, let's create a new data set with these 12 variables. Below is a glimpse of the data set.

```
compas_final <- clean_compasdata %>%
  dplyr::select(c(race, sex, age, age_cat, marital_status,
                  custody_status, juv_offense, priors_count, c_charge_degree,
                  days_in_jail_or_prison, decile_score, is_recid))

glimpse(compas_final)
```

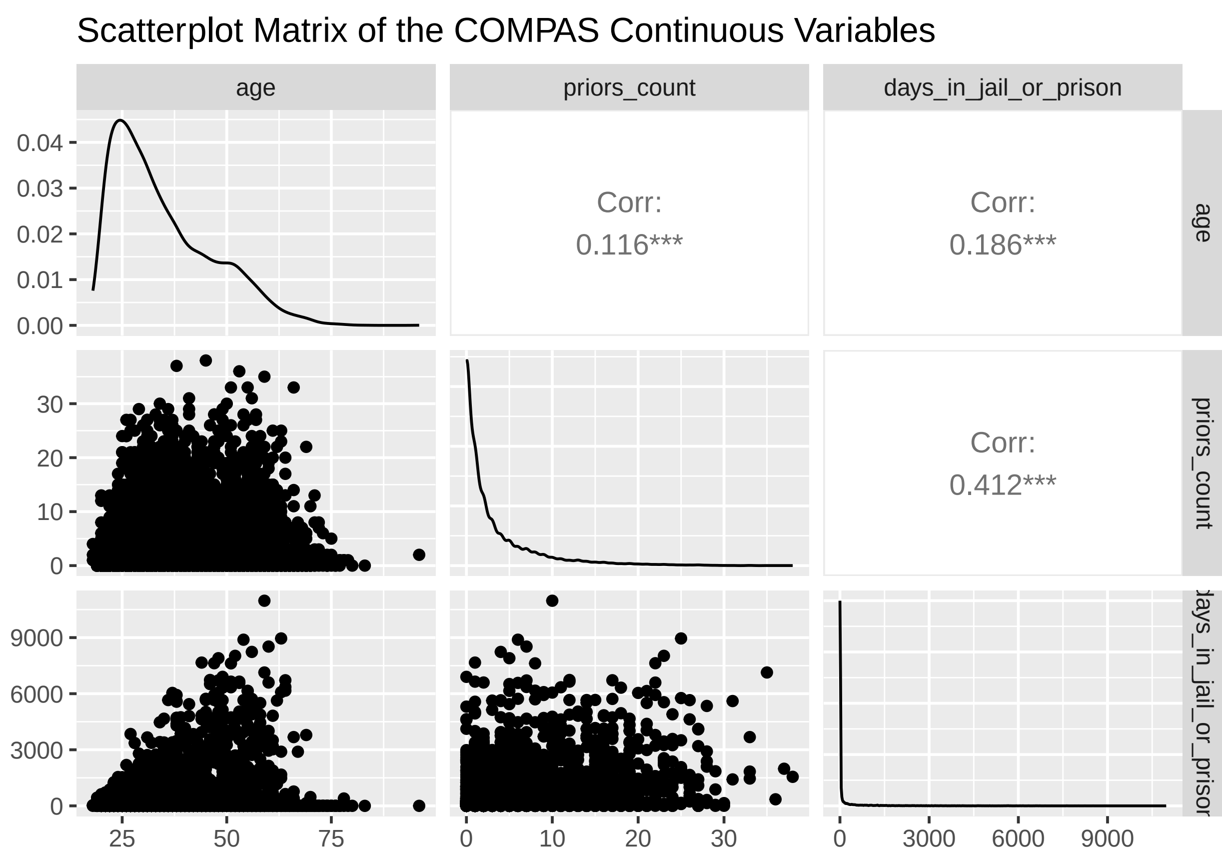
```
## Rows: 9,387
## Columns: 12
## $ race                       <chr> "Other", "African-American", "African-American"~
## $ sex                        <chr> "Male", "Male", "Male", "Male", "Male", "Male",~
```

```
## $ age                     <int> 69, 34, 24, 44, 41, 43, 39, 20, 26, 27, 23, 37,~
## $ age_cat                 <chr> "Greater than 45", "25 - 45", "Less than 25", "~
## $ marital_status          <chr> "Single", "Single", "Single", "Separated", "Sin~
## $ custody_status          <chr> "Jail Inmate", "Jail Inmate", "Jail Inmate", "J~
## $ juv_offense             <dbl> 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ priors_count            <int> 0, 0, 4, 0, 14, 3, 0, 0, 0, 0, 3, 0, 0, 0, 1, 7~
## $ c_charge_degree         <chr> "(F3)", "(F3)", "(F3)", "(M1)", "(F3)", "(F3)",~
## $ days_in_jail_or_prison <dbl> 1, 10, 1, 0, 1071, 1, 3, 33, 1, 1, 0, 0, 0, 1, ~
## $ decile_score            <int> 1, 3, 4, 1, 6, 4, 1, 10, 5, 4, 6, 1, 3, 4, 1, 3~
## $ is_recid                <int> 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,~
```

Scatterplot Matrix

First, a scatterplot matrix with just the 3 continuous predictive variables in the final data set, `age`, `prior offenses`, and `days in jail or prison`, will help to elucidate the covariate relationships between the variables. All the variables have moderate to weak correlations, with the strongest correlation of 0.412 being between the number of prior offenses and the number of days spent in jail. All correlations are significant. There are no major concerns for multicollinearity.

```
ggpairs(data = compas_final,
        columns = c("age", "priors_count", "days_in_jail_or_prison"),
        title = "Scatterplot Matrix of the COMPAS Continuous Variables")
```

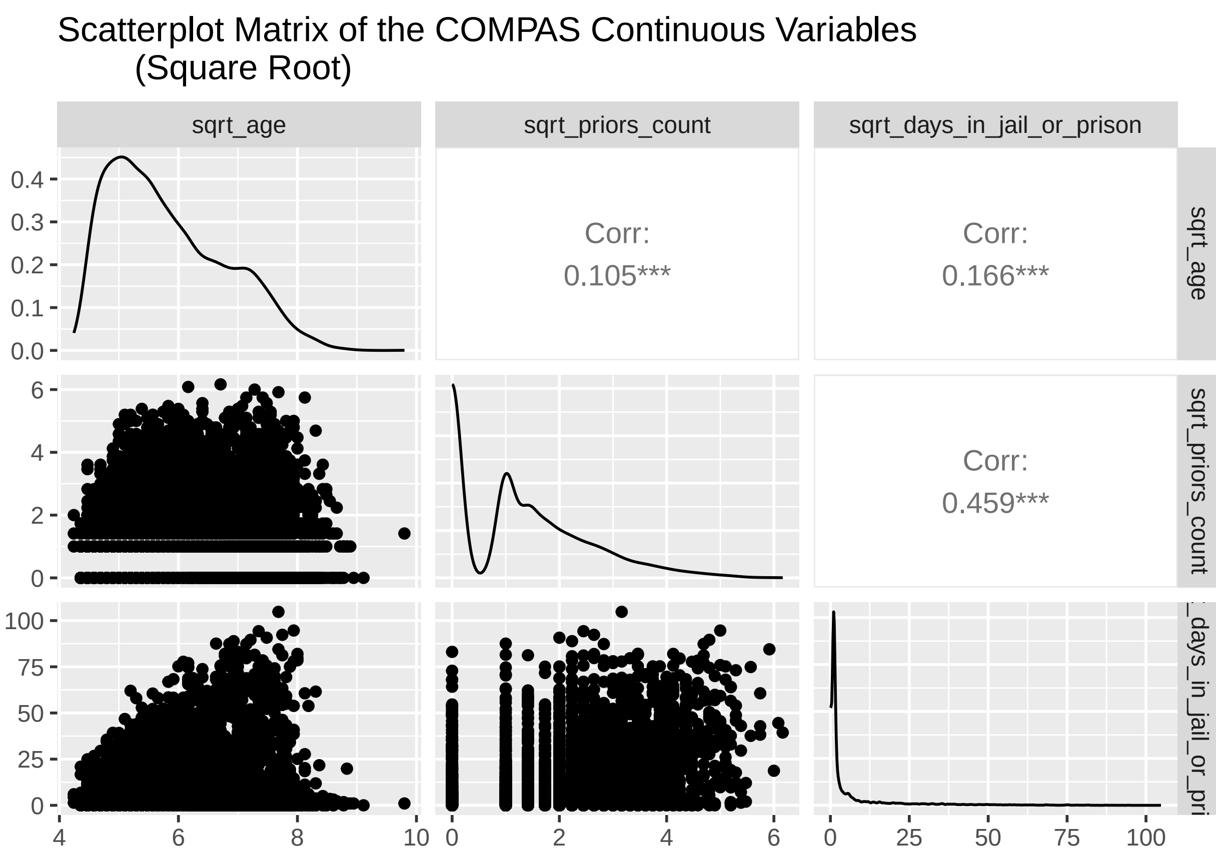


As observed, the variables have a significant right skew and the relationship is non-linear. Let's explore what effect different transformations may have on the covariate relationships. The log transformation resulted in many non-finite values because of the present of zeros, so we will look at a square root transformation instead.

```
compas_final <- compas_final %>%
  mutate(sqrt_age = sqrt(age),
         sqrt_priors_count = sqrt(priors_count),
         sqrt_days_in_jail_or_prison = sqrt(days_in_jail_or_prison))
```

While this transformation strengthened the relationship between the number of prior offenses and the number of days spent in jail or prison ($r = 0.459$), it weakened the other correlations slightly. Next, we'll assess how much this affects the relationship with the COMPAS decile scores.

```
ggpairs(data = compas_final,
        columns = c("sqrt_age", "sqrt_priors_count", "sqrt_days_in_jail_or_prison"),
        title = "Scatterplot Matrix of the COMPAS Continuous Variables
        (Square Root)")
```



Pearson's Correlation Matrix

Decile scores has a moderate relationship with age, prior offenses, and number of days spent in jail or prison – this suggests that these variables may indeed be useful for modeling recidivism.

```
mycordata1 <- compas_final %>%
  dplyr::rename("Age" = age,
         "Priors" = priors_count,
         "Days in Jail or Prison" = days_in_jail_or_prison,
```

```
        "Decile Scores" = decile_score) %>%
  dplyr::select("Age", "Priors", "Days in Jail or Prison", "Decile Scores")

cor(mycordata1) %>%
  kable(digits = 2,
        booktabs = TRUE)
```

	Age	Priors	Days in Jail or Prison	Decile Scores
Age	1.00	0.12	0.19	-0.39
Priors	0.12	1.00	0.41	0.45
Days in Jail or Prison	0.19	0.41	1.00	0.26
Decile Scores	-0.39	0.45	0.26	1.00

The square root transformation of the predictor variables actually slightly strengthens the relationship with decile scores – this suggests that the square root transformation of these variables may be better for modeling recidivism.

```
mycordata2 <- compas_final %>%
  dplyr::rename("Square Root Age" = sqrt_age,
        "Priors" = sqrt_priors_count,
        "Days in Jail or Prison" = sqrt_days_in_jail_or_prison,
        "Decile Scores" = decile_score) %>%
  dplyr::select("Square Root Age", "Priors", "Days in Jail or Prison", "Decile Scores")

cor(mycordata2) %>%
  kable(digits = 2,
        booktabs = TRUE)
```

	Square Root Age	Priors	Days in Jail or Prison	Decile Scores
Square Root Age	1.00	0.11	0.17	-0.39
Priors	0.11	1.00	0.46	0.46
Days in Jail or Prison	0.17	0.46	1.00	0.38
Decile Scores	-0.39	0.46	0.38	1.00

Spearman's Correlation Matrix

Spearman's Correlation is better at capturing non-linear relationships. Using Spearman correlations reveals slightly stronger correlations between the variables and the COMPAS decile scores, especially the days spent in jail or prison.

There is no observable difference when calculating Spearman's correlation with the square-root transformed variables versus the original variables.

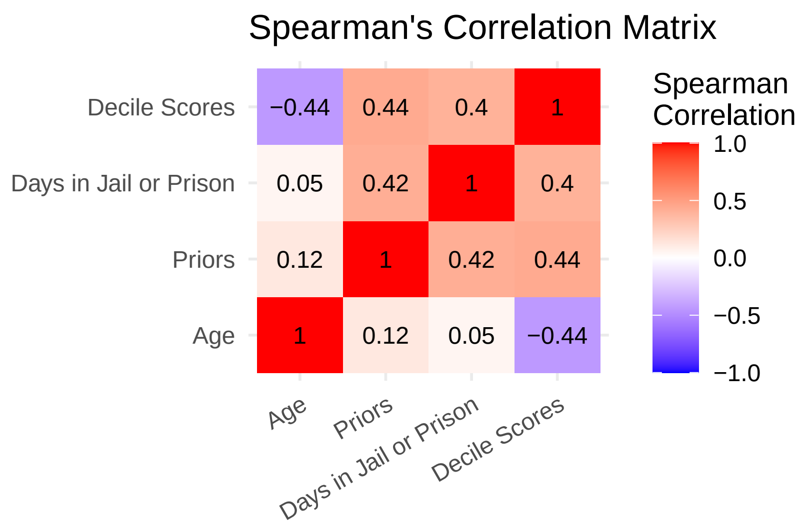
```
mycordata3 <- compas_final %>%
  dplyr::rename("Age" = age,
        "Priors" = priors_count,
        "Days in Jail or Prison" = days_in_jail_or_prison,
        "Decile Scores" = decile_score) %>%
  dplyr::select("Age", "Priors", "Days in Jail or Prison", "Decile Scores")

cor(mycordata3, method = "spearman") %>%
```

```r
kable(digits = 2,
      booktabs = TRUE)
```

	Age	Priors	Days in Jail or Prison	Decile Scores
Age	1.00	0.12	0.05	-0.44
Priors	0.12	1.00	0.42	0.44
Days in Jail or Prison	0.05	0.42	1.00	0.40
Decile Scores	-0.44	0.44	0.40	1.00

Finally, let's visualize these correlations.

```r
mycors <- round(cor(mycordata3, method = "spearman"),2)
mycorplot <- melt(mycors)

ggplot(data = mycorplot, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  labs(x = "",
       y = "",
       title = "Spearman's Correlation Matrix") +
  scale_fill_gradient2(
    low = "blue",
    high = "red",
    mid = "white",
    midpoint = 0,
    limit = c(-1, 1),
    space = "Lab",
    name = "Spearman\nCorrelation"
  ) +
  geom_text(aes(Var2, Var1, label = value),
            color = "black",
            size = 3) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, vjust = 1, hjust = 1))
```



Now that we have a thorough understanding of the make-up of the data set, we will perform a demographic analysis next to get a better understanding of the racial discrepancies that may be present before, finally, proceeding with the recidivism risk modeling.

Demographic Group Analysis

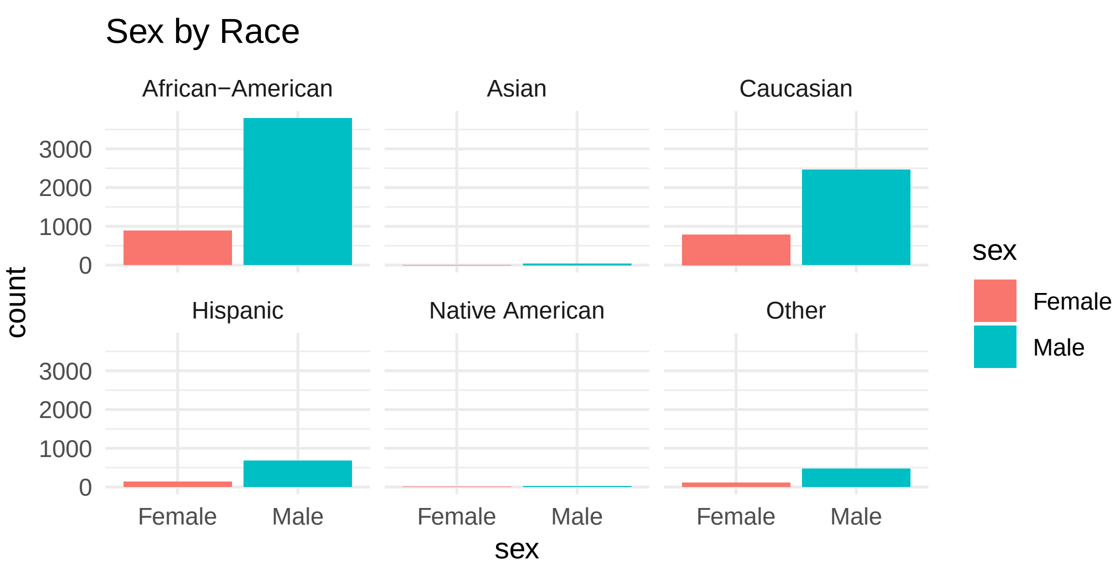
Bivariate Analysis by Race

The demographic variable of interest for this analysis is `race`, and we will employ the Seldonian algorithm in a hope to achieve fairer recidivism risk predictions. With that goal in mind, it's important to perform a demographic group analysis along the defendants' races to better understand any underlying or proxy relationships with the variables.

First, let's analyze the bivariate relationships of some of the most important variables with race.

For all the races, with the exception of native Americans for who there are not many data points available, there are more male defendants than female defendants.

```
ggplot(data = compas_final, mapping = aes(x = sex, fill = sex)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~race) +
  labs(title = "Sex by Race")
```



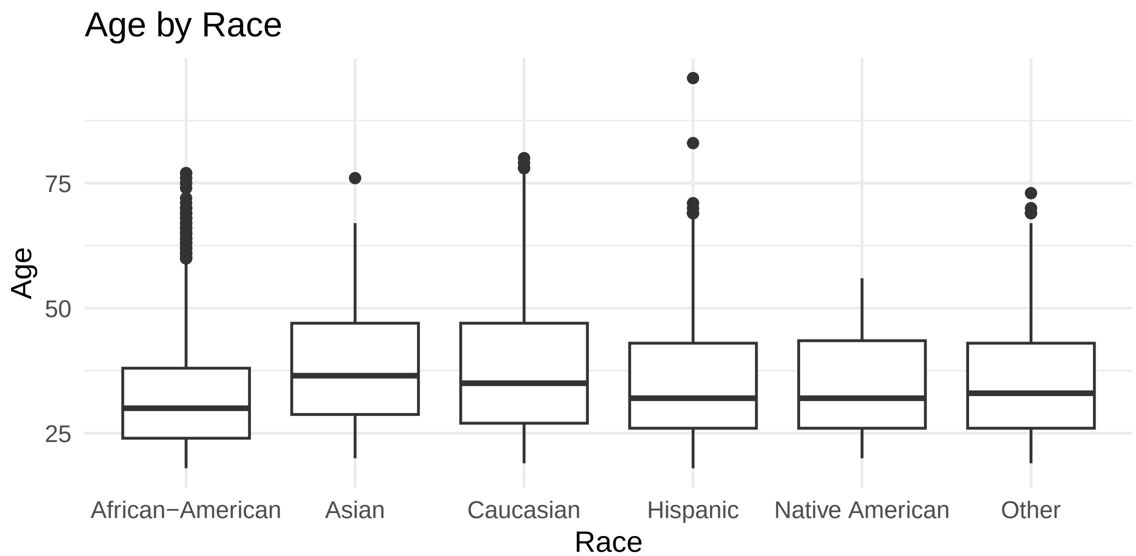
African-American defendants tend to be, on average, the youngest compared to all the other races. Asian defendants, followed by Caucasian defendants, tend to be the oldest. However, there is considerable overlap among all the races, and the relationships is visualized in the boxplot below.

```
favstats(data = compas_final, age ~ race)
```

```
##                 race min    Q1 median   Q3 max     mean       sd    n missing
## 1 African-American  18 24.00   30.0 38.0  77 32.60312 10.77213 4674       0
## 2            Asian  20 28.75   36.5 47.0  76 38.20833 12.21607   48       0
## 3        Caucasian  19 27.00   35.0 47.0  80 37.51969 12.60537 3250       0
## 4         Hispanic  18 26.00   32.0 43.0  96 35.22494 11.86236  818       0
## 5  Native American  20 26.00   32.0 43.5  56 34.29630 10.57870   27       0
```

```
## 6           Other  19 26.00   33.0 43.0  73 35.67895 11.68420  570       0
```

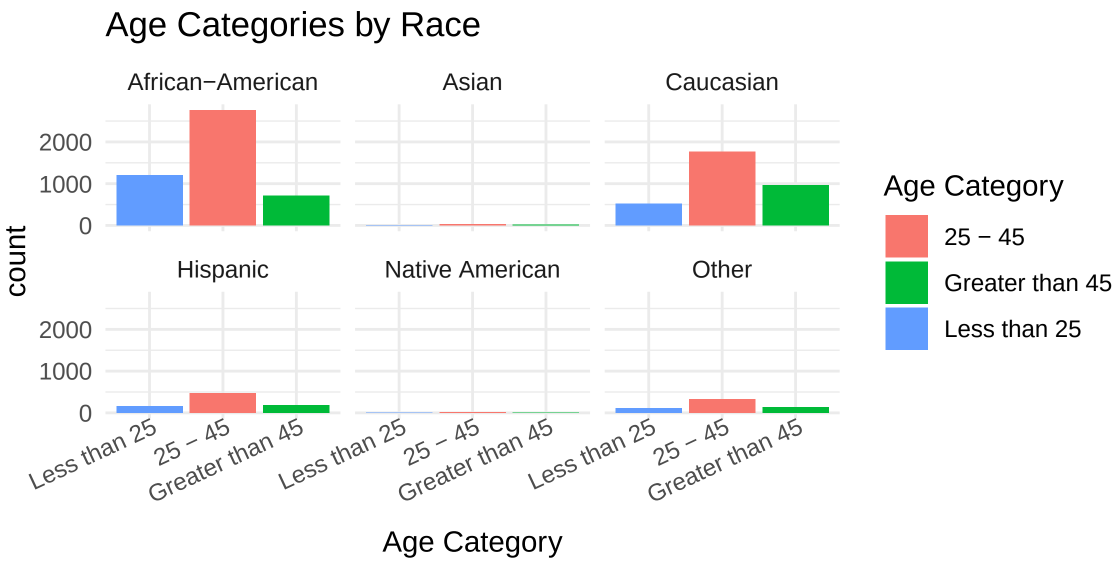
```
ggplot(data = compas_final,
       mapping = aes(x = as.factor(race), y = age)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Age by Race",
       x = "Race",
       y = "Age")
```



The bar plot below illustrates that for all races, most defendants fall between the ages of 25 and 45. Notice, however, that for African-American defendants, there are more defendants that are less than 25 than those that are greater than 45. The converse is true for Caucasians, with more defendants that are greater than 45 in comparison to those less than 25.

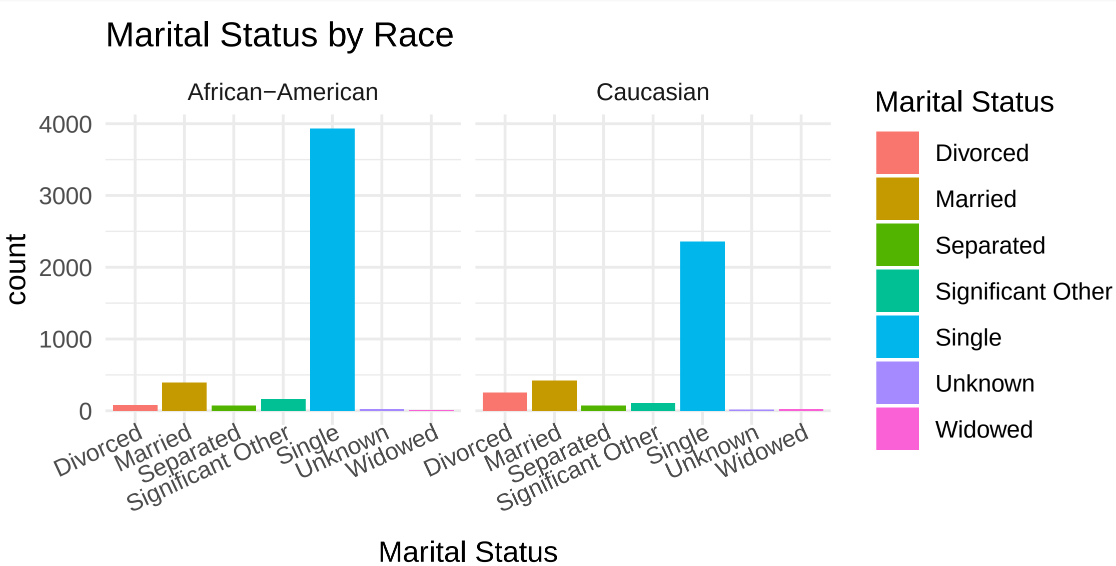
This illustrates that there is some relationship between age and race in this data set, particularly for African-Americans versus Caucasians.

```
order <- c("Less than 25", "25 - 45", "Greater than 45")

ggplot(data = compas_final,
       mapping = aes(x = age_cat, fill = age_cat)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~race) +
  labs(title = "Age Categories by Race",
       x = "Age Category",
       fill = "Age Category") +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1)) +
  scale_x_discrete(limits = order)
```



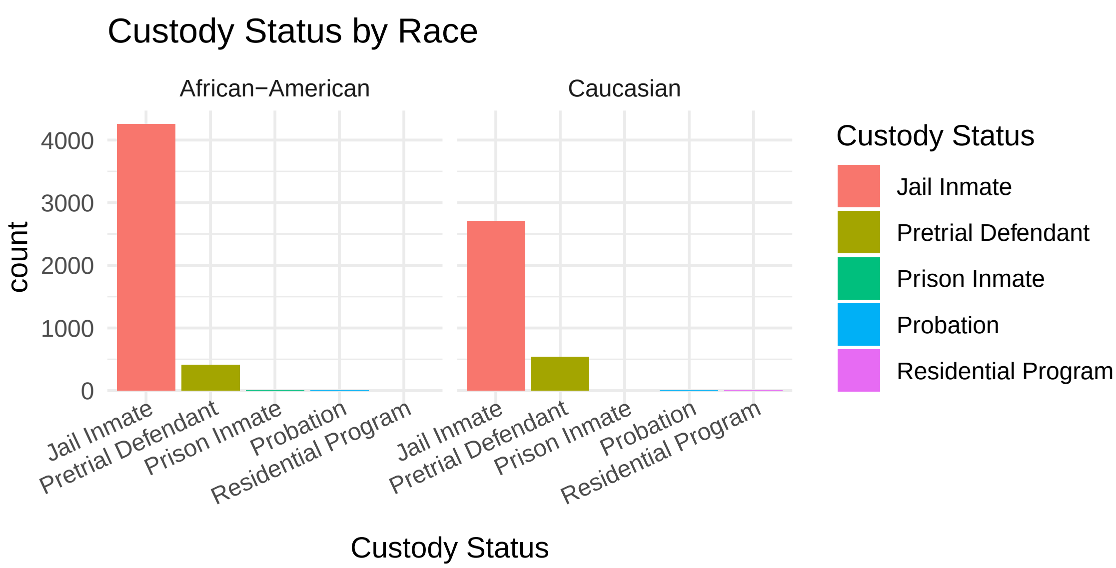
For simpler visualization, let's assess the marital status just for Black and White defendants. For both races, most defendants are single. There are no distinct distributional differences.

```
compas_final_bw <- compas_final %>%
  filter(race %in% c("African-American", "Caucasian"))

ggplot(data = compas_final_bw,
       mapping = aes(x = marital_status, fill = marital_status)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~race) +
  labs(title = "Marital Status by Race",
       x = "Marital Status",
       fill = "Marital Status") +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1))
```



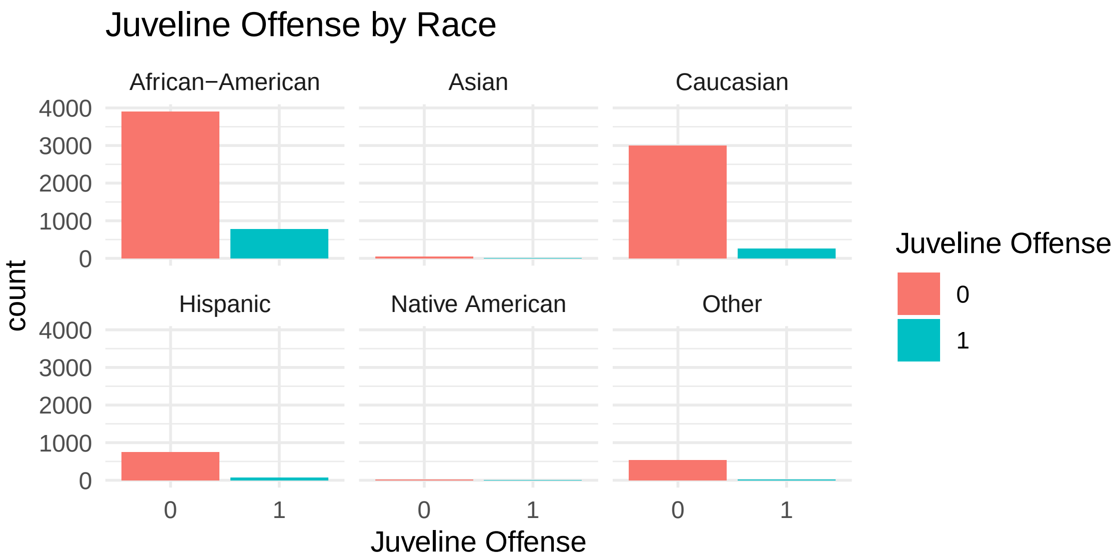
Similarly, there is no observable distribution difference in custody status by race. However, as much as there are less Caucasian defendants overall in comparison to African-American defendants, there are slightly more Caucasian prison defendants.

```r
ggplot(data = compas_final_bw,
       mapping = aes(x = custody_status, fill = custody_status)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~race) +
  labs(title = "Custody Status by Race",
       x = "Custody Status",
       fill = "Custody Status") +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1))
```



There does not appear to be much distributional difference in juvenile offense by race.

```r
ggplot(data = compas_final,
       mapping = aes(x = as.factor(juv_offense), fill = as.factor(juv_offense))) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~race) +
  labs(title = "Juveline Offense by Race",
       x = "Juveline Offense",
       fill = "Juveline Offense")
```

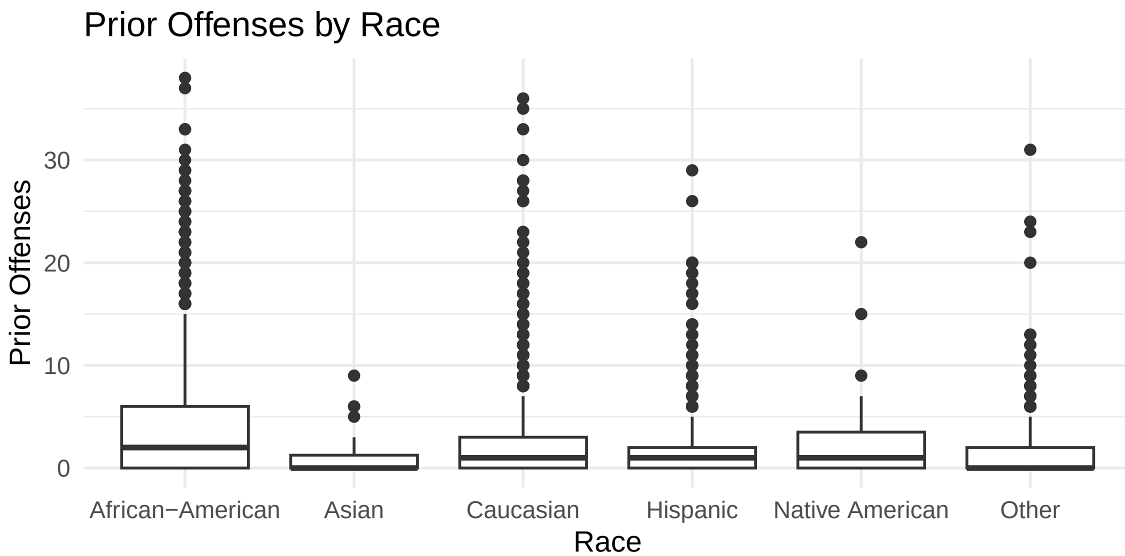


Notably, African-Americans have the most prior offenses, on average, followed by Native Americans. When comparing with Caucasian defendants, African-Americans have almost twice as many prior offenses, suggesting a strong proxy relationship between race and prior offenses. Asian defendants have the least prior offenses. This is an important result and illustrates how a system that pre-disposes certain races to prison can perpetuate that discriminatory trend by using those same variables to predict risk of recommitting another crime. The boxplot helps to visualize this relationship more clearly.

```
favstats(data = compas_final, priors_count ~ race)
```

```
##                 race min Q1 median   Q3 max     mean       sd    n missing
## 1 African-American   0  0      2 6.00  38 4.042576 5.345310 4674       0
## 2            Asian   0  0      0 1.25   9 1.083333 1.888750   48       0
## 3        Caucasian   0  0      1 3.00  36 2.146462 3.472545 3250       0
## 4         Hispanic   0  0      1 2.00  29 1.806846 3.321139  818       0
## 5  Native American   0  0      1 3.50  22 3.185185 5.076621   27       0
## 6            Other   0  0      0 2.00  31 1.575439 2.949861  570       0
```

```
ggplot(data = compas_final,
       mapping = aes(x = as.factor(race), y = priors_count)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Prior Offenses by Race",
       x = "Race",
       y = "Prior Offenses")
```

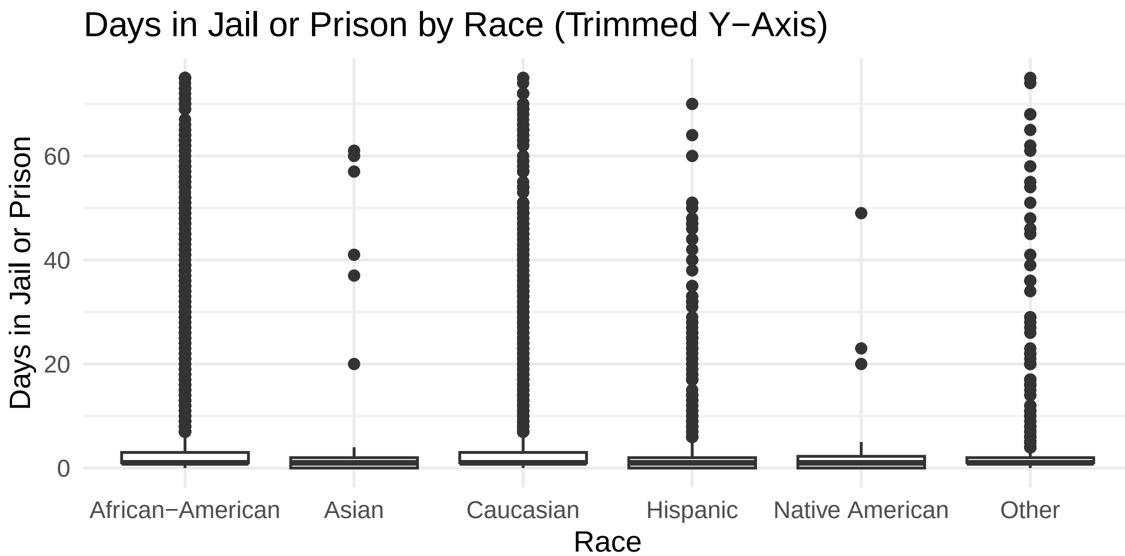


The same trend is observed when observing the difference in the days spent in jail. On average, Native Americans and African-Americans spend the most time in jail or prison. In fact, their average jail/ prison time is more than quadruple and triple, respectively, that of Caucasian defendants. Asian defendants spend the least time on average. When looking at the medians, 50% of the African-American defendants spent 2 days or more days in jail, as compared to only 1 or more days for 50% of the all other defendants. The relationship is hard to visualize on a boxplot because of the presence of many outlying observations.

```
favstats(data = compas_final, days_in_jail_or_prison ~ race)
```

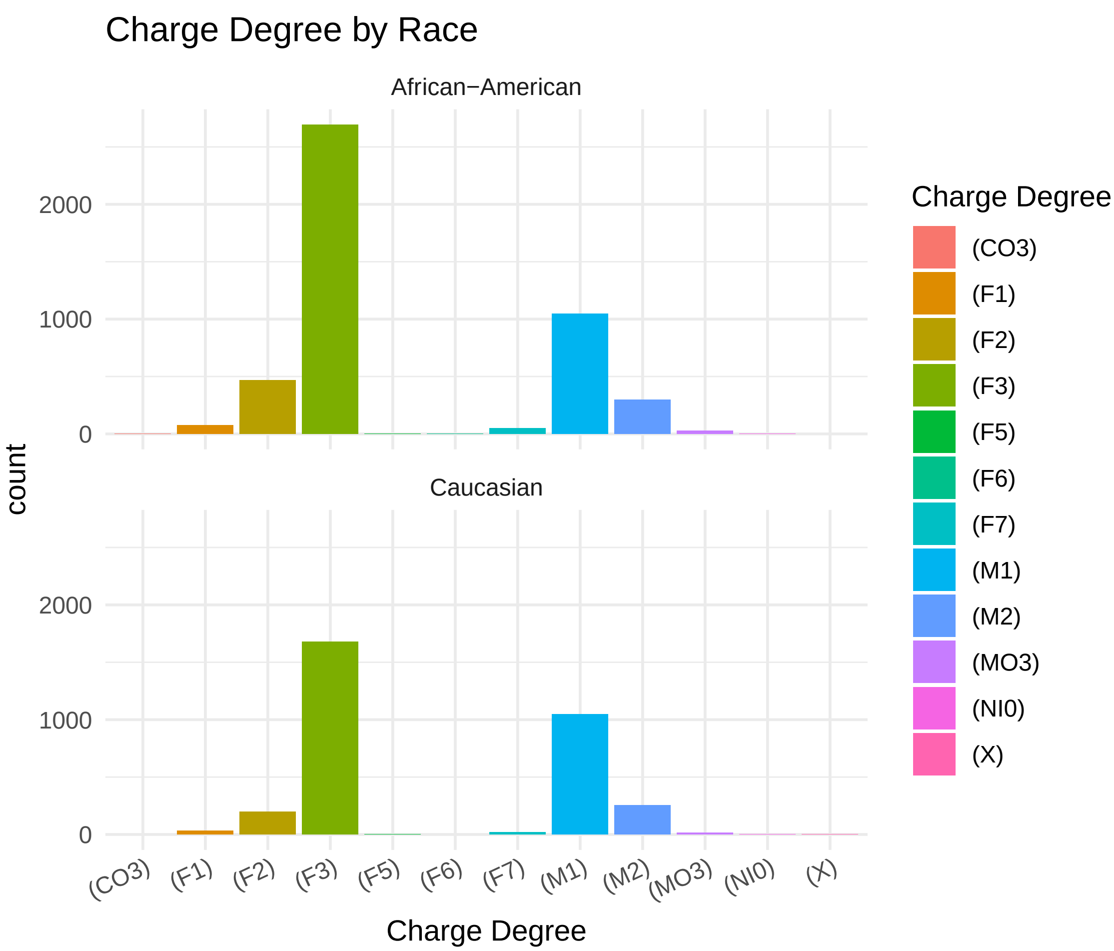
```
##                 race min Q1 median    Q3   max      mean         sd    n missing
## 1 African-American   0  1      2 57.75 10973 307.07018  927.29956 4674       0
## 2            Asian   0  1      1 24.25   388  33.95833   77.45883   48       0
## 3        Caucasian   0  1      1  7.00  7136  98.39908  464.94108 3250       0
## 4         Hispanic   0  0      1  5.00  5608 106.21027  483.40377  818       0
## 5  Native American   0  0      1  4.50  7630 421.25926 1528.72022   27       0
## 6            Other   0  1      1  3.00  3844  53.52105  302.96923  570       0
```

```
ggplot(data = compas_final,
       mapping = aes(x = as.factor(race), y = days_in_jail_or_prison)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Days in Jail or Prison by Race (Trimmed Y-Axis)",
       x = "Race",
       y = "Days in Jail or Prison") +
  ylim(0,75)
```



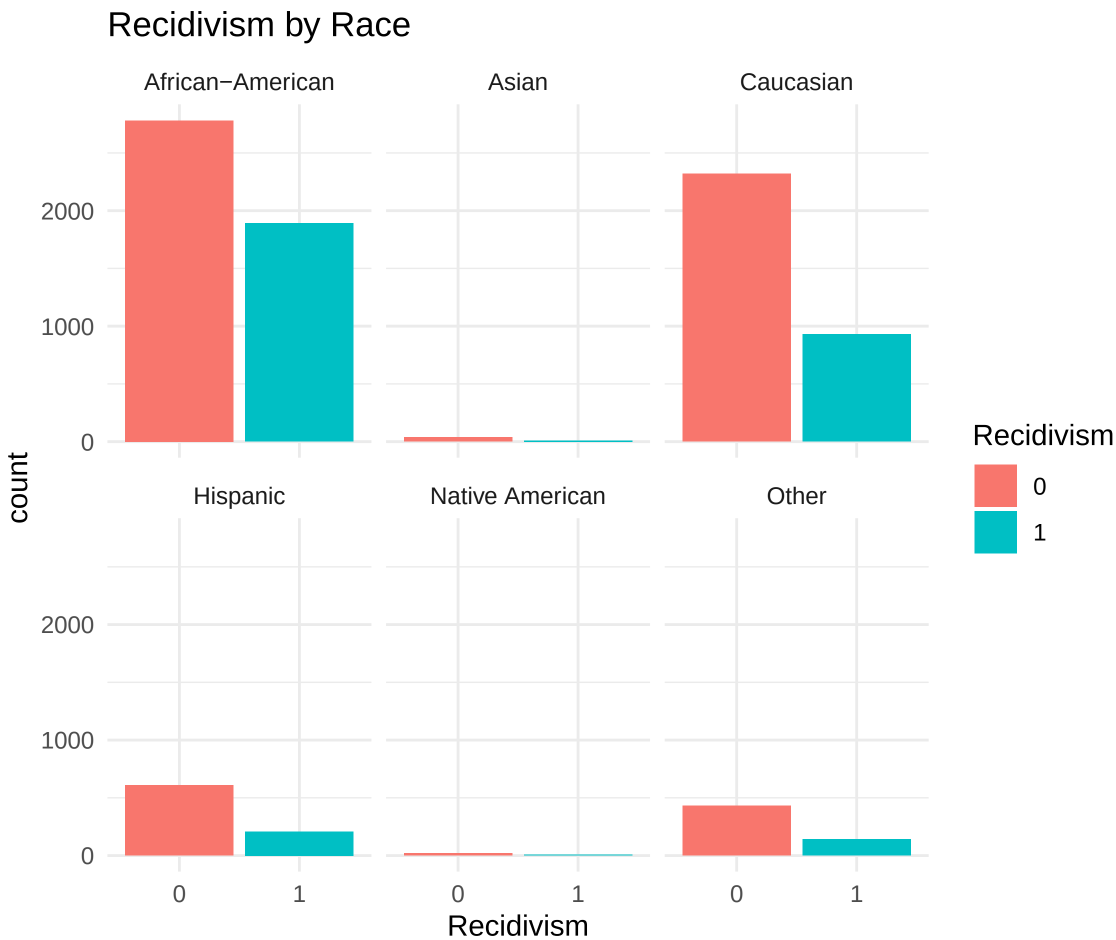
Next, let's look at the distribution of the charge degrees just for Black and White defendants. There are no notable distributional differences.

```
ggplot(data = compas_final_bw,
       mapping = aes(x = c_charge_degree, fill = c_charge_degree)) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~race, ncol = 1) +
  labs(title = "Charge Degree by Race",
       x = "Charge Degree",
       fill = "Charge Degree") +
  theme(axis.text.x = element_text(angle = 25, vjust = 1.2, hjust=1))
```



Finally, let's look at the response variable: `is_recid`. Most defendants do not re-commit a crime, although the ratio of those who don't against those who do appears to be larger for African-American defendants than for Caucasian defendants. Nevertheless, of important note is that the class imbalance is in the same direction.

```
ggplot(data = compas_final,
       mapping = aes(x = as.factor(is_recid), fill = as.factor(is_recid))) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~race) +
  labs(title = "Recidivism by Race",
       x = "Recidivism",
       fill = "Recidivism")
```



In conclusion, our demographic variable, `race` (A), has proxy relationships with some of the predictor variables (X). Even a group-blind classifier will not be entirely blind to race because of the correlations present and the information that can be gained from the proxy variables.

Additionally, we've learned that the distribution of those who recommit a crime within 2 years versus those who don't is not drastically different for Black v White defendants – the class imbalance is in the same direction, so ideally, the model should not misclassify defendants in different directions on the basis of race.

Next, let's see how the COMPAS tool actually performs with regard to race and whether it is in line with these expectations.

COMPAS Analysis

Before we can analyze the COMPAS performance, recall that that the decile scores are mapped to 3 different risk levels – low, medium, or high – yet the response variable only has 2 levels – 0 or 1. Let's recreate this mapping such that there are only 2 risk levels to line up with the 2 levels of the response variable. Instead, decile scores of 1 to 5 will be associated with lower risk of recidivism and decile scores of 6 to 10 will be associated with higher risk as displayed in the table below.

```
compas <- compas_final %>%
  mutate(risk = ifelse(decile_score %in% c(1, 2, 3, 4, 5),'Lower','Higher'))
```

```r
compas %>%
  dplyr::select(decile_score, risk) %>%
  rename("Risk" = risk) %>%
  group_by(Risk) %>%
  summarise("Min" = min(decile_score),
            "Max" = max(decile_score)) %>%
  arrange(Min) %>%
  kable(booktabs = TRUE)
```

Risk	Min	Max
Lower	1	5
Higher	6	10

Now, let's recreate the table from Chapter 1 using this data set to assess the false positive and false negative rates for Black v White defendants.

While the numbers are different, which could likely be due to a host of reasons such as different subsets of the data set, the same trends are evident. 16.34% of White defendants who did not re-offend are labelled as higher risk, compared to more than twice as many Black defendants (37.71%). Similarly, 62.26% of White defendants who do re-offend are labelled as lower risk, compared to 39.01% of Black defendants. This is in line with ProPublica's findings, and is alarming given that race was not included in the model.

```r
compas_table <- compas %>%
  filter(race %in% c("African-American", "Caucasian")) %>%
  dplyr::select(race, risk, is_recid) %>%
  rename("Risk" = risk,
         "Race" = race) %>%
  group_by(Race, is_recid) %>%
  mutate(Total = n()) %>%
  group_by(Risk, Race, Total) %>%
  summarise("Reoffended" = count(is_recid == 1),
            "Did Not Reoffend" = count(is_recid == 0)) %>%
  pivot_longer(cols = c("Reoffended", "Did Not Reoffend"),
               names_to = "Recidivism") %>%
  pivot_wider(
    id_cols = c("Risk", "Recidivism", "Total"),
    names_from = "Race",
    values_from = value
  ) %>%
  rename("Black" = `African-American`,
         "White" = `Caucasian`) %>%
  mutate(Black = round(100 * Black / Total, 2),
         White = round(100 * White / Total, 2)) %>%
  dplyr::select(-Total) %>%
  group_by(Risk, Recidivism) %>%
  summarize(Black = max(Black, na.rm = TRUE),
            White = max(White, na.rm = TRUE)) %>%
  filter((Risk == "Higher" & Recidivism == "Did Not Reoffend") |
           (Risk == "Lower" & Recidivism == "Reoffended")
  )

compas_table %>%
  kable(booktabs = TRUE)
```

Risk	Recidivism	Black	White
Higher	Did Not Reoffend	37.71	16.34
Lower	Reoffended	39.01	62.26

The results in this section also show that the model is more wrong in predicting whether defendants will re-offend versus predicting defendants who do not re-offend. This is expected because of the class imbalance we observed when performing exploratory data analysis – there are more defendants that don't re-offend, so the model maximizes performance for those defendants.

However, it also raises a question of what type of prediction is more important: the risk of recidivism or the risk of non-recidivism. Is wrongly attributing a defendant as higher risk may cause or wrongly attributing a defendant as lower risk worse?

The table below shows the confusion matrix of the COMPAS model as a whole, including all races. As observed, a larger proportion (49.25%) of defendants who re-offended are incorrectly labelled as lower risk in comparison to the proportion (25.23%) that did not re-offend but are incorrectly labelled as higher risk.

Observe that while the overall FPR is 25.23%, it's much higher for Black defendants (37.71%) and much lower for White defendants (16.34%). Similarly, while the overall FNR is 49.25%, it's much higher for White defendants (62.26%) and much lower for Black defendants (39.01%).

```
compas %>%
  dplyr::select(risk, is_recid) %>%
  rename("Risk" = risk) %>%
  group_by(is_recid) %>%
  mutate(Total = n()) %>%
  group_by(Risk, Total) %>%
  summarise("Reoffended" = count(is_recid == 1),
            "Did Not Reoffend" = count(is_recid == 0)) %>%
  mutate(Reoffended = round(100 * Reoffended / Total, 2),
         `Did Not Reoffend` = round(100 * `Did Not Reoffend` / Total, 2)) %>%
  dplyr::select(-Total) %>%
  group_by(Risk) %>%
  summarize(Reoffended = max(Reoffended, na.rm = TRUE),
            `Did Not Reoffend` = max(`Did Not Reoffend`, na.rm = TRUE)) %>%
  kable()
```

Risk	Reoffended	Did Not Reoffend
Higher	50.75	25.23
Lower	49.25	74.77

The table below replicates the above table with raw numbers instead of proportions. This reveals that the model has an overall accuracy of 66.61%, but the analysis above reveals the discrepancies (and unfairness) in the model results for Black v White defendants.

However, if we classified every observation in the majority class (no recidivism), we'd expect an accuracy of 66.04%. The COMPAS model, thus, is not useful in a predictive sense, and on the contrary, introduces more bias into the judicial system.

```
compas %>%
  dplyr::select(risk, is_recid) %>%
  rename("Risk" = risk) %>%
```

```r
  group_by(Risk) %>%
  summarise("Reoffended" = count(is_recid == 1),
            "Did Not Reoffend" = count(is_recid == 0)) %>%
  kable()
```

Risk	Reoffended	Did Not Reoffend
Higher	1618	1564
Lower	1570	4635

```r
(1618 + 4635)/ 9387
```

```
## [1] 0.666134
```

```r
count(compas$is_recid == 0)/ 9387
```

```
##    n_TRUE
## 0.6603814
```

Now that we've recreated the table from Chapter 1, let's examine the false negative and false positive rates for all the races in the data set.

Asian defendants who did not re-offend were the least likely to be labelled as higher risk – Black defendants were the most likely. Conversely, excluding the "Other" group, White defendants who re-offended were the most likely to be labelled as lower risk – Native Americans were the least likely. This, and previous analysis, suggest disparities with favorable outcomes for white and Asian defendants, and unfavorable outcomes for Black and Native American defendants.

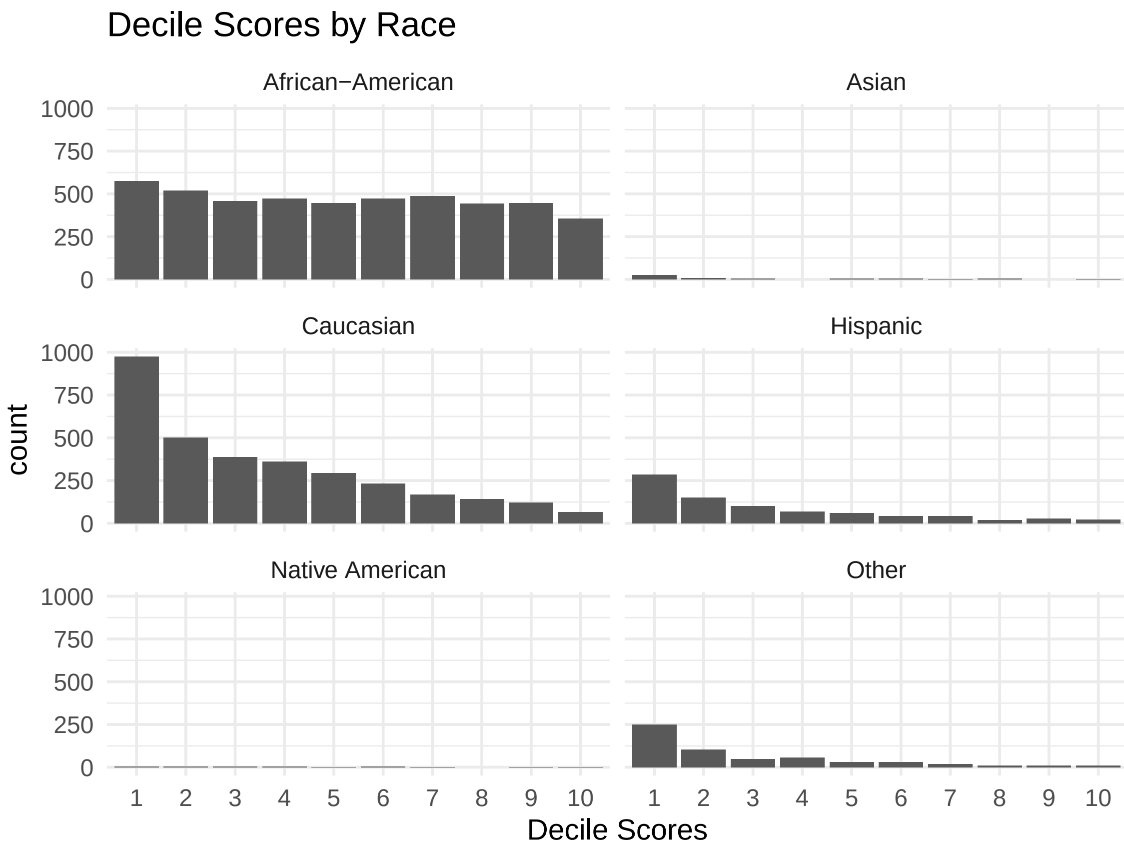
```r
compas %>%
  dplyr::select(race, risk, is_recid) %>%
  rename("Risk" = risk,
         "Race" = race) %>%
  group_by(Race, is_recid) %>%
  mutate(Total = n()) %>%
  group_by(Risk, Race, Total) %>%
  summarise("Reoffended" = count(is_recid == 1),
            "Did Not Reoffend" = count(is_recid == 0)) %>%
  pivot_longer(cols = c("Reoffended", "Did Not Reoffend"),
               names_to = "Recidivism") %>%
  pivot_wider(
    id_cols = c("Risk", "Recidivism", "Total"),
    names_from = "Race",
    values_from = value
  ) %>%
  rename("Black" = `African-American`,
         "White" = `Caucasian`) %>%
  mutate(Black = round(100 * Black / Total, 2),
         White = round(100 * White / Total, 2),
         Hispanic = round(100 * Hispanic / Total, 2),
         Asian = round(100 * Asian / Total, 2),
         `Native American` = round(100 * `Native American` / Total, 2),
         Other = round(100 * Other / Total, 2)) %>%
  dplyr::select(-Total) %>%
  group_by(Risk, Recidivism) %>%
  summarize(Black = max(Black, na.rm = TRUE),
```

```r
            White = max(White, na.rm = TRUE),
            Hispanic = max(Hispanic, na.rm = TRUE),
            Asian = max(Asian, na.rm = TRUE),
            `Native American` = max(`Native American`, na.rm = TRUE),
            Other = max(Other, na.rm = TRUE)) %>%
  filter((Risk == "Higher" & Recidivism == "Did Not Reoffend") |
           (Risk == "Lower" & Recidivism == "Reoffended")
  ) %>%
  kable(booktabs = TRUE)
```

Risk	Recidivism	Black	White	Hispanic	Asian	Native American	Other
Higher	Did Not Reoffend	37.71	16.34	14.26	7.89	21.05	9.77
Lower	Reoffended	39.01	62.26	68.27	50.00	25.00	74.29

Finally, to visualize these results better, let's plot the distribution of the decile scores by race. Observe that while we observed a similar distribution of recidivism for all races, the African-American decile scores are distributed differently from the Caucasian decile scores. The decile scores for the Caucasian defendants is quite right-skewed in comparison to that of the African-Americans which appears to be more evenly distributed, further emphasizing the racial disparity in the risk scores.

```r
ggplot(data = compas,
       mapping = aes(x = as.factor(decile_score))) +
  geom_bar() +
  theme_minimal() +
  facet_wrap(~race, ncol = 2) +
  labs(title = "Decile Scores by Race",
       x = "Decile Scores")
```



We have a thorough understanding of the data set now and the results we would expect from the COMPAS tool.

We're ready to proceed with modeling now. First, we will fit a logistic regression, which uses the standard ML approach discussed in Chapter 2. We expect it to behave similarly to the COMPAS tool. Then, we will define a fairness metric and fit a Seldonian classification algorithm on this data set. We expect to observe less disparities along racial lines, but perhaps with a slight trade-off in overall accuracy.

Logistic Regression

Logistic regression is a statistical generalized linear model (GLM) that is particularly designed for predicting dichotomous/ binary outcomes such as ours. We will use logistic regression to model the probability of a defendant re-committing a crime within two years in Broward County, Florida. We'll then set cutoffs to divide the probabilities into 2 bins: 0 for 'no' and 1 for 'yes' based on the probability predictions. This will also allow us to analyze which features may be most important in predicting recidivism.

Recall that if we classified every observation in the majority class, we'd expect an accuracy 66%. This serves as a benchmark for the race-blind logistic regression model we will implement. Note that logistic regression follows the standard ML process [will describe in more detail in the thesis body] and is one of the most widely used classification algorithms – this will allow us to assess how we might expect state-of-the-art traditional algorithms that do not take fairness guarantees into account to perform.

Check for Missing Data

Linear models do not handle missing observations well. Let's ensure that there are no missing observations. There were no missing observations!

```
comopas <- tidyr::drop_na(compas)
count(compas)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  9387
```

Train and Test Split

Before we proceed to fitting the models, we need to perform a train/ test split. The reason for this is to be able to test how our model would perform on unseen data and to avoid over-fitting. Therefore, we will train our models using the `train` data, and then assess performance on "new" data, that is, the `test` set. Let's partition 70% of the data into the train set and 30% into the test set. We will use randomization without replacement for this.

```
set.seed(123)
n <- nrow(compas)
train_index <- sample(1:n, 0.70 * n)
test_index <- setdiff(1:n, train_index)
train <- compas[train_index, ]
test <- compas[test_index, ]
```

There are now 6570 observations in the train set and 2817 observations in the test set. Let's ensure that the distribution of the response variable is preserved in each set. There split appears to be stratified so there is no concern.

```
tally(train$is_recid)
```

```
## X
##    0    1
## 4333 2237
```

```
tally(test$is_recid)
```

```
## X
##    0    1
## 1866  951
```

Finally, let's also make sure that there are enough observations in each race category for both the train and test splits. There is concern for the Native American and Asian race categories, which have very few observations. We may fit a separate model with just the Caucasian and African-American offenders if this poses a challenge.

```
tally(train$race)
```

```
## X
## African-American            Asian        Caucasian         Hispanic
##             3270               34             2280              573
```

```
##  Native American            Other 
##               18              395 
```

```r
tally(test$race)
```

```
## X
## African-American            Asian        Caucasian         Hispanic 
##             1404               14              970              245 
##  Native American            Other 
##                9              175 
```

We're ready for model fitting!

Modeling

First, let's fit a kitchen sink model without transformations. This means that we'll include all the variables in the initial model. Notice that `race` will not be used as a predictor in this model.

```r
glm1 <- glm(is_recid ~ sex + age + age_cat + marital_status + custody_status +
              juv_offense + priors_count + c_charge_degree +
              days_in_jail_or_prison,
            data = train,
            family = binomial(logit))

msummary(glm1)
```

```
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                       1.487e+01  5.354e+02   0.028 0.977839    
## sexMale                           3.790e-01  7.428e-02   5.103 3.35e-07 ***
## age                              -4.954e-02  6.150e-03  -8.054 8.00e-16 ***
## age_catGreater than 45            4.943e-01  1.446e-01   3.417 0.000633 ***
## age_catLess than 25               8.701e-02  8.949e-02   0.972 0.330914    
## marital_statusMarried            -1.077e-01  1.846e-01  -0.584 0.559505    
## marital_statusSeparated          -3.515e-02  2.575e-01  -0.136 0.891454    
## marital_statusSignificant Other   4.999e-01  2.193e-01   2.279 0.022676 *  
## marital_statusSingle              1.398e-01  1.670e-01   0.837 0.402593    
## marital_statusUnknown             2.335e-01  3.981e-01   0.586 0.557575    
## marital_statusWidowed             1.115e+00  4.781e-01   2.331 0.019734 *  
## custody_statusPretrial Defendant -8.774e-02  8.779e-02  -0.999 0.317580    
## custody_statusPrison Inmate      -8.414e-01  1.238e+00  -0.680 0.496762    
## custody_statusProbation          -1.261e+01  5.354e+02  -0.024 0.981216    
## custody_statusResidential Program 4.276e-01  1.455e+00   0.294 0.768862    
## juv_offense                       3.881e-01  8.830e-02   4.396 1.10e-05 ***
## priors_count                      1.349e-01  7.740e-03  17.430  < 2e-16 ***
## c_charge_degree(F1)              -1.520e+01  5.354e+02  -0.028 0.977355    
## c_charge_degree(F2)              -1.497e+01  5.354e+02  -0.028 0.977698    
## c_charge_degree(F3)              -1.480e+01  5.354e+02  -0.028 0.977941    
## c_charge_degree(F5)              -2.780e+01  5.851e+02  -0.048 0.962099    
## c_charge_degree(F6)              -2.859e+01  6.512e+02  -0.044 0.964987    
## c_charge_degree(F7)              -1.501e+01  5.354e+02  -0.028 0.977633    
## c_charge_degree(M1)              -1.501e+01  5.354e+02  -0.028 0.977636    
## c_charge_degree(M2)              -1.478e+01  5.354e+02  -0.028 0.977980    
## c_charge_degree(MO3)             -1.480e+01  5.354e+02  -0.028 0.977944    
```

```
## c_charge_degree(NI0)              -1.379e+01  5.354e+02  -0.026 0.979450
## c_charge_degree(X)                -2.755e+01  7.572e+02  -0.036 0.970977
## days_in_jail_or_prison            -9.274e-07  4.427e-05  -0.021 0.983287
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8427.4  on 6569  degrees of freedom
## Residual deviance: 7549.8  on 6541  degrees of freedom
## AIC: 7607.8
##
## Number of Fisher Scoring iterations: 12
```

Let's calculate the overall accuracy of this model on the train set, which tends to be overconfident.

The values in the diagonal are correct predictions while those in the off-diagonal are misclassifications using a probability cutoff of 0.5.

```
glm1augment <- glm1 %>%
  broom::augment(type.predict = "response")
glm1augment <- mutate(glm1augment, binprediction = round(.fitted, 0))
with(glm1augment, table(is_recid, binprediction))
```

```
##         binprediction
## is_recid    0    1
##        0 3938  395
##        1 1564  673
```

This kitchen-sink model has an accuracy of 70.2%, slightly better than the COMPAS tool accuracy of 66%.

```
(3938 + 673)/ count(train)
```

```
##           n
## 1 0.7018265
```

Using a probability cutoff of 0.34, which is the probability of being in the positive class given the class imbalance in the COMPAS data set, the accuracy is the same as the COMPAS tool: 66.7%.

```
glm1augment_2 <- glm1 %>%
  broom::augment(type.predict = "response")
glm1augment_2 <- mutate(glm1augment_2, binprediction = ifelse(.fitted >= 0.34, 1, 0))
with(glm1augment_2, table(is_recid, binprediction))
```

```
##         binprediction
## is_recid    0    1
##        0 2905 1428
##        1  761 1476
```

```
(2905+1476)/count(train)
```

```
##           n
## 1 0.6668189
```

Using a stricter probability results in an accuracy of 68.2%

```
glm1augment_2 <- glm1 %>%
  broom::augment(type.predict = "response")
glm1augment_2 <- mutate(glm1augment_2, binprediction = ifelse(.fitted >= 0.66, 1, 0))
with(glm1augment_2, table(is_recid, binprediction))
```

```
##         binprediction
## is_recid    0    1
##        0 4229  104
##        1 1988  249
```

```
(4229+249)/count(train)
```

```
##          n
## 1 0.681583
```

The probability of 0.5 is an appropriate cutoff. Notice, however, that the only significant variables in the kitchen-sink model are `sex`, `age`, `age_category`, `marital_status`, `juv_offense`, and `priors_count`. Let's fit another model with just these predictors.

```
glm2 <- glm(is_recid ~ sex + age + age_cat + marital_status +
              juv_offense + priors_count,
            data = train,
            family = binomial(logit))

msummary(glm2)
```

```
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -0.009828   0.273046  -0.036 0.971287
## sexMale                            0.390681   0.073758   5.297 1.18e-07 ***
## age                               -0.050222   0.006087  -8.251  < 2e-16 ***
## age_catGreater than 45             0.508380   0.144297   3.523 0.000426 ***
## age_catLess than 25                0.089960   0.089052   1.010 0.312403
## marital_statusMarried             -0.136133   0.184174  -0.739 0.459813
## marital_statusSeparated           -0.023786   0.257143  -0.093 0.926301
## marital_statusSignificant Other    0.493876   0.219082   2.254 0.024178 *
## marital_statusSingle               0.141861   0.166802   0.850 0.395062
## marital_statusUnknown              0.255089   0.396522   0.643 0.520019
## marital_statusWidowed              1.139997   0.475660   2.397 0.016545 *
## juv_offense                        0.382606   0.087645   4.365 1.27e-05 ***
## priors_count                       0.137055   0.007249  18.908  < 2e-16 ***
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8427.4  on 6569  degrees of freedom
## Residual deviance: 7576.6  on 6557  degrees of freedom
## AIC: 7602.6
##
## Number of Fisher Scoring iterations: 4
```

```r
glm2augment <- glm2 %>%
  broom::augment(type.predict = "response")
glm2augment <- mutate(glm2augment, binprediction = round(.fitted, 0))
with(glm2augment, table(is_recid, binprediction))
```

```
##          binprediction
## is_recid    0    1
##         0 3956  377
##         1 1581  656
```

There is no effect on predictive accuracy by reducing the number of variables: 70.2%.

```r
(3956 + 656)/ count(train)
```

```
##           n
## 1 0.7019787
```

Finally, let's assess performance when we use the square-root transformed variables. The same variables are significant.

```r
glm3 <- glm(is_recid ~ sex + sqrt_age + age_cat + marital_status + custody_status +
              juv_offense + sqrt_priors_count + c_charge_degree +
              sqrt_days_in_jail_or_prison,
            data = train,
            family = binomial(logit))

msummary(glm3)
```

```
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       16.313306 535.411403   0.030 0.975693
## sexMale                            0.339417   0.075246   4.511 6.46e-06 ***
## sqrt_age                          -0.632307   0.074644  -8.471  < 2e-16 ***
## age_catGreater than 45             0.487163   0.139432   3.494 0.000476 ***
## age_catLess than 25                0.112351   0.096475   1.165 0.244195
## marital_statusMarried             -0.067026   0.185160  -0.362 0.717361
## marital_statusSeparated           -0.032919   0.258771  -0.127 0.898772
## marital_statusSignificant Other    0.475337   0.220975   2.151 0.031469 *
## marital_statusSingle               0.097231   0.167383   0.581 0.561313
## marital_statusUnknown              0.301756   0.404084   0.747 0.455205
## marital_statusWidowed              0.997625   0.486060   2.052 0.040124 *
## custody_statusPretrial Defendant  -0.138916   0.088326  -1.573 0.115775
## custody_statusPrison Inmate       -1.018251   1.237448  -0.823 0.410586
## custody_statusProbation          -12.834477 535.411175  -0.024 0.980876
## custody_statusResidential Program  0.585953   1.505249   0.389 0.697074
## juv_offense                        0.261143   0.088817   2.940 0.003280 **
## sqrt_priors_count                  0.566413   0.028799  19.668  < 2e-16 ***
## c_charge_degree(F1)              -15.010178 535.411237  -0.028 0.977634
## c_charge_degree(F2)              -14.767817 535.411181  -0.028 0.977995
## c_charge_degree(F3)              -14.584335 535.411173  -0.027 0.978269
## c_charge_degree(F5)              -27.778103 584.539153  -0.048 0.962098
## c_charge_degree(F6)              -28.630853 652.924899  -0.044 0.965024
## c_charge_degree(F7)              -14.846965 535.411251  -0.028 0.977877
## c_charge_degree(M1)              -14.720471 535.411175  -0.027 0.978066
```

```
## c_charge_degree(M2)              -14.540801 535.411183  -0.027 0.978334
## c_charge_degree(MO3)             -14.582420 535.411278  -0.027 0.978272
## c_charge_degree(NI0)             -13.486319 535.412727  -0.025 0.979904
## c_charge_degree(X)               -27.202623 757.185745  -0.036 0.971341
## sqrt_days_in_jail_or_prison        0.003900   0.002495   1.563 0.118072
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8427.4  on 6569  degrees of freedom
## Residual deviance: 7434.0  on 6541  degrees of freedom
## AIC: 7492
##
## Number of Fisher Scoring iterations: 12
```

A reduced model yields the same overall accuracy observed in the other models: 70.2%.

```
glm4 <- glm(is_recid ~ sex + age + age_cat + marital_status +
              juv_offense + priors_count,
            data = train,
            family = binomial(logit))

glm4augment <- glm4 %>%
  broom::augment(type.predict = "response")
glm4augment <- mutate(glm4augment, binprediction = round(.fitted, 0))
with(glm4augment, table(is_recid, binprediction))
```

```
##         binprediction
## is_recid    0    1
##        0 3956  377
##        1 1581  656
```

```
(3956 + 656)/ count(train)
```

```
##           n
## 1 0.7019787
```

Based on the modeling results, we will proceed with `glm2`, which fits the most important non-transformed variables.

Evaluating Test Set Performance

It's important to assess performance on unseen data as that serves as a better indicator for model generalization performance and rules out over-fitting.

The performance on the test set is comparable with the training set at 70.1% accuracy.

```
preds <- predict(glm2, newdata=test, type="response")

test2 <- test %>%
  mutate(preds = preds,
         prediction = round(preds, 0))

with(test2, table(is_recid, prediction))
```

```
##          prediction
## is_recid    0    1
##        0 1699  167
##        1  675  276
```

```r
(1699 + 276)/ count(test)
```

```
##           n
## 1 0.7011005
```

The ROC curve is a graph that also helps us assess the performance of a classification model. The diagonal line shows how the model would perform with random predictions. We prefer curves that budge out closer to the top left because those types of curves maximize the area under the curve (AUC) and the sensitivity and specificity of the model is closer to 100%.

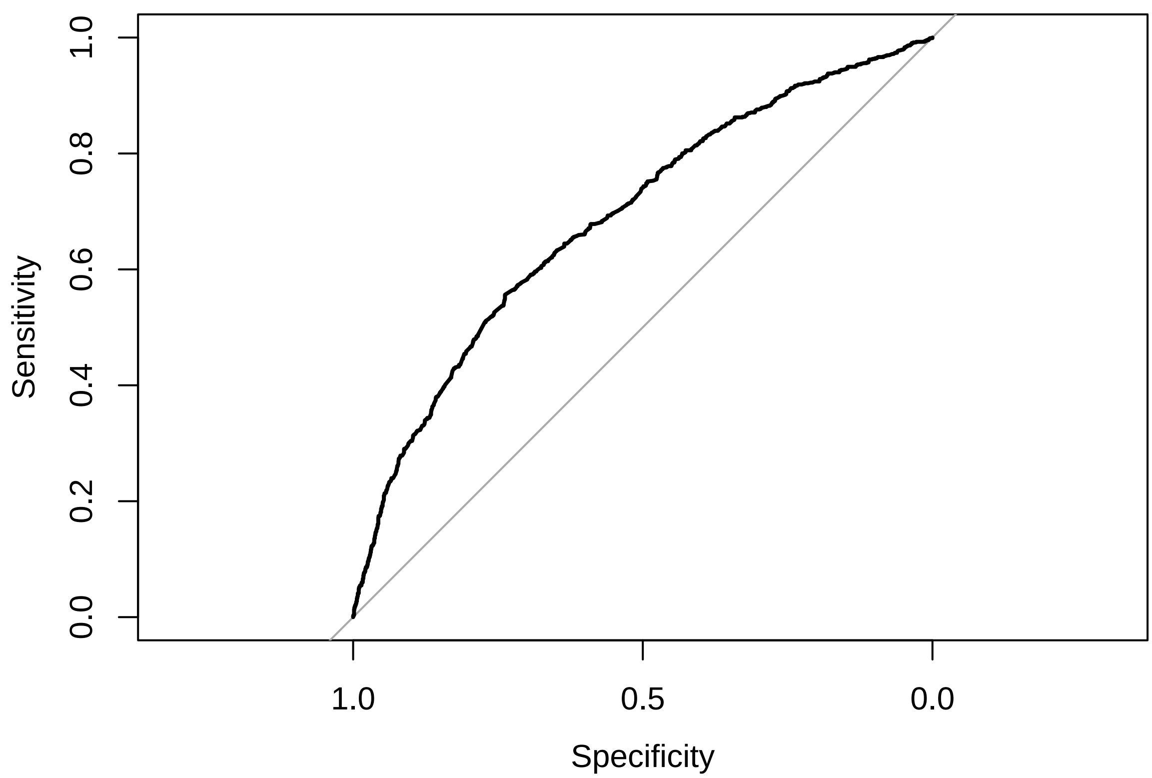
This curve has an AUC of 0.68 and it doesn't exhibit the best shape. This raises overall concerns about the model's performance.

```r
predlm <- predict(glm2, type=c("response"), newdata = test)
roccurve1 <- with(test, roc(is_recid ~ predlm))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
plot(roccurve1)
```




```r
print(roccurve1)
```

```
##
## Call:
## roc.formula(formula = is_recid ~ predlm)
##
## Data: predlm in 1866 controls (is_recid 0) < 951 cases (is_recid 1).
## Area under the curve: 0.688
```

In conclusion, the final logistic regression model has the following 6 variables:

- age
- sex
- age category
- marital status
- whether a defendant committed a binary offense or not
- number of prior offenses

The accuracy on the test set is 70.1%, comparable to the COMPAS test set. This means that ~30% of defendants are misclassified in the incorrect recidivism prediction.

However, the ROC curve indicated some problems with the performance of this model, particularly with regard to sensitivity (1 - FNR) and specificity (1- FPR). Next, let's assess the direction of these misclassifications and perform a demographic analysis by race in an attempt to assess the model's 'fairness' along racial lines.

Demographic and 'Fairness' Analysis

First, let's re-examine the confusion matrix using the test set to determine whether the model makes more false negative or more false positive results.

The first table depicts the raw numbers and the second table interprets that in terms of proportions in order to better understand the magnitude.

The model correctly classifies 276 defendants as reoffenders and 1699 defendants as non-reoffenders. 675 defendants who reoffend are classified as low risk, whereas 167 defendants who did not reoffend are classified as high risk. The model does a better job at classifying those who do not reoffend, as is expected because there is more data for that group. In general, this is consistent with what we observed in the COMPAS data set.

```r
test2 <- test2 %>%
  mutate(pred_risk = ifelse(prediction == 0, 'Low', 'High'))
```

```r
test2 %>%
  dplyr::select(pred_risk, is_recid) %>%
  rename("Risk" = pred_risk) %>%
  group_by(Risk) %>%
  summarise("Reoffended" = count(is_recid == 1),
            "Did Not Reoffend" = count(is_recid == 0)) %>%
  kable()
```

Risk	Reoffended	Did Not Reoffend
High	276	167
Low	675	1699

When looking at the proportions, 91% of participants who did not reoffend are correctly classified as low risk. This means only 9% of participants who do not reoffend are classified as high risk.

On the flip side, only 29% of defendants who reoffended are correctly classified as high risk. This means 71 percent of defendants who reoffended are classified as low risk.

The model, overall, has worse false negative rates than false positive rates. While this is the same general trend observed in the COMPAS data set, the discrepancy is much larger.

```
test2 %>%
  dplyr::select(pred_risk, is_recid) %>%
  rename("Risk" = pred_risk) %>%
  group_by(is_recid) %>%
  mutate(Total = n()) %>%
  group_by(Risk, Total) %>%
  summarise("Reoffended" = count(is_recid == 1),
            "Did Not Reoffend" = count(is_recid == 0)) %>%
  mutate(Reoffended = round(100 * Reoffended / Total, 2),
         `Did Not Reoffend` = round(100 * `Did Not Reoffend` / Total, 2)) %>%
  dplyr::select(-Total) %>%
  group_by(Risk) %>%
  summarize(Reoffended = max(Reoffended, na.rm = TRUE),
            `Did Not Reoffend` = max(`Did Not Reoffend`, na.rm = TRUE)) %>%
  kable()
```

Risk	Reoffended	Did Not Reoffend
High	29.02	8.95
Low	70.98	91.05

Now, let's examine these false positive and false negative rates along racial lines. First. we'll compare Black v White defendants.

Now, let's compare all the defendants.

Finally, ...