Northpointe/ProPublica Descriptive Statistics

We start by loading the data

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
raw_data <- read.csv("~/My Drive/tex-documents/working-papers/algo-fairness/algo-fairness-m/rutgers-pres
nrow(raw_data)
```

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

Filtering the data a bit:

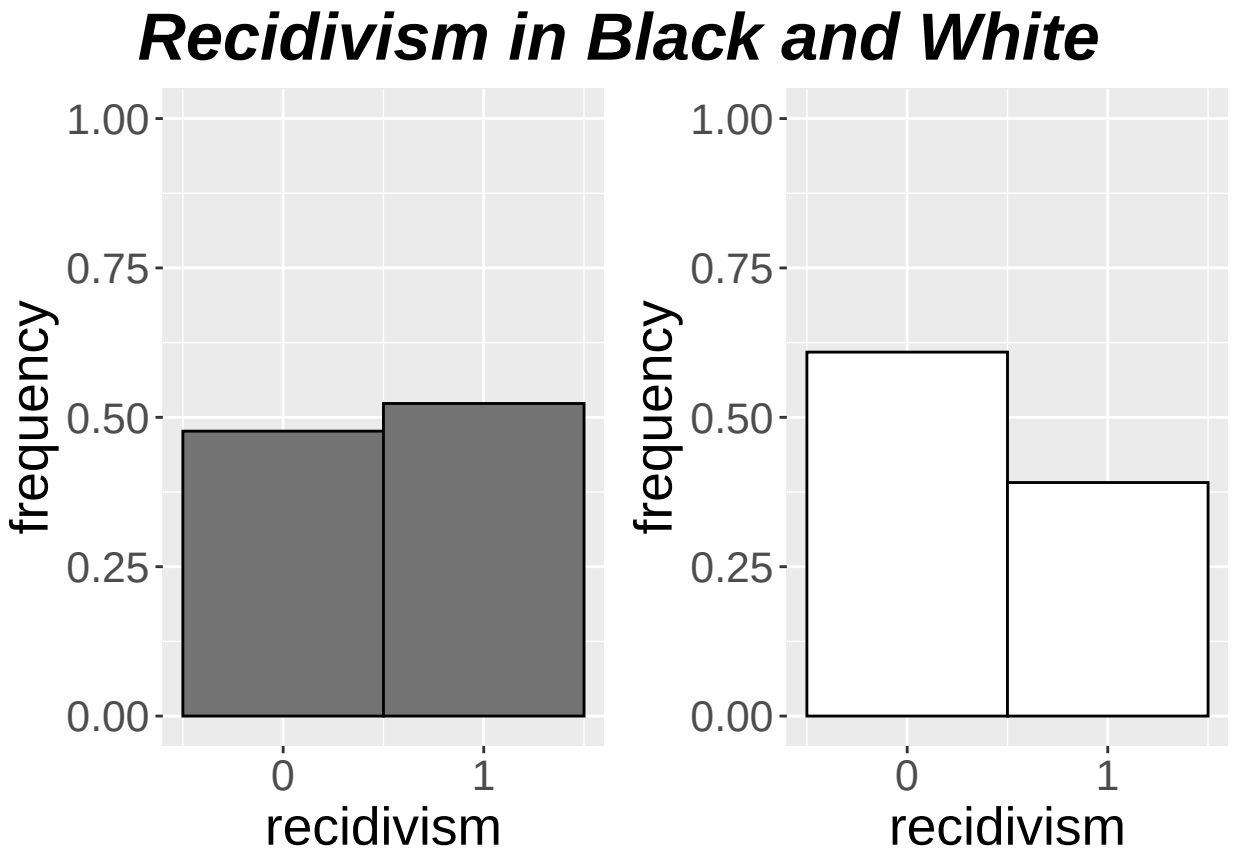
```
df <- dplyr::select(raw_data, age, c_charge_degree, race, age_cat, score_text, sex, priors_count,
                    days_b_screening_arrest, decile_score, is_recid, two_year_recid, c_jail_in, c_jail_
        filter(days_b_screening_arrest <= 30) %>%
        filter(days_b_screening_arrest >= -30) %>%
        filter(is_recid != -1) %>%
        filter(c_charge_degree != "O") %>%
        filter(score_text != 'N/A')
nrow(df)
```

```
## [1] 6172
```

```
pblack <- ggplot(data=filter(df, race =="African-American"), aes(x=two_year_recid)) +
          geom_histogram(aes(y=..count../sum(..count..)), binwidth = 1,  colour="black", fill="grey45")
          ylab("frequency") +
          ylim(0, 1) +
          scale_x_continuous(breaks = seq(0, 1, by = 1)) +
          theme(plot.title = element_text(hjust = 0.5)) +
          theme(text=element_text(size = 20))
pwhite <- ggplot(data=filter(df, race =="Caucasian"), aes(x=two_year_recid)) +
```

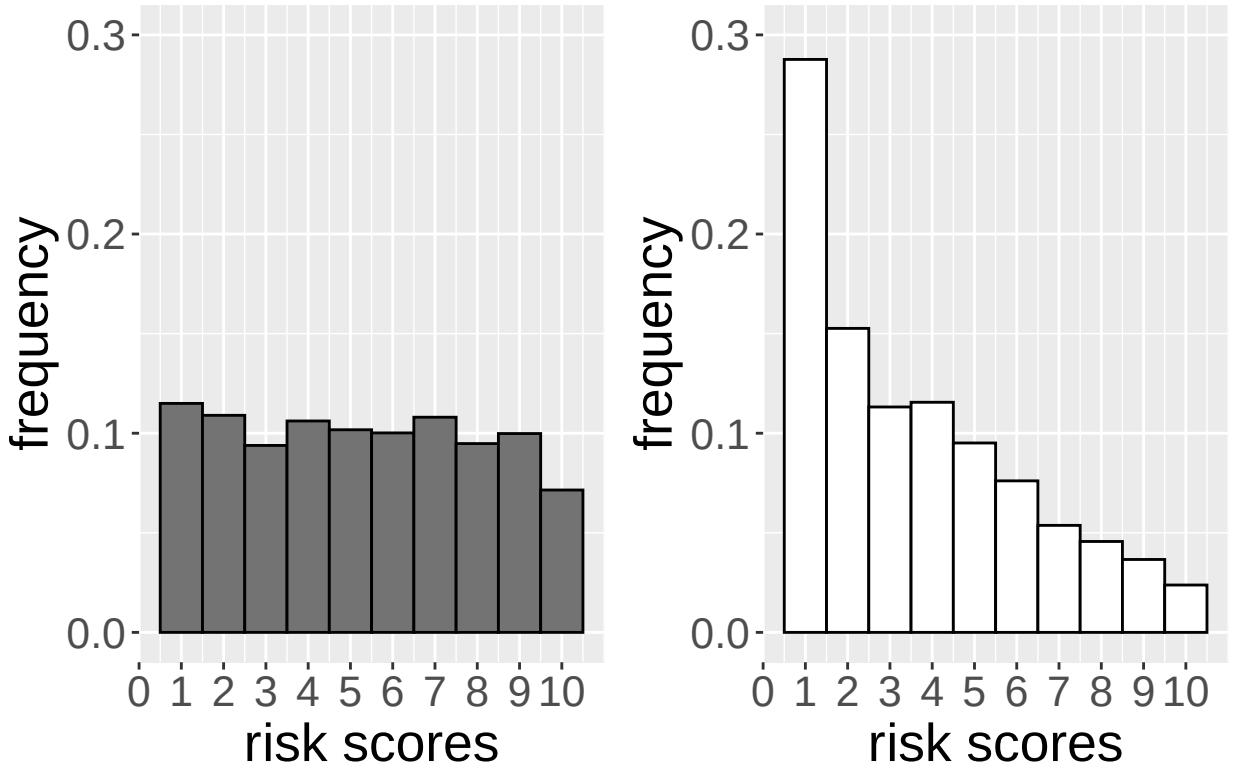
```
          geom_histogram(aes(y=..count../sum(..count..)), binwidth = 1,  colour="black", fill="grey100")
          ylab("frequency") +
          ylim(0, 1) +
          scale_x_continuous(breaks = seq(0, 1, by = 1)) +
          theme(plot.title = element_text(hjust = 0.5)) +
          theme(text=element_text(size = 20))

grid.arrange(pblack, pwhite, ncol = 2, top=textGrob("Recidivism in Black and White", vjust= 0.4, gp=gpar
```




```
pblack_s <- ggplot(data=filter(df, race =="African-American"), aes(x=decile_score)) +
          geom_histogram(aes(y=..count../sum(..count..)), binwidth = 1,  colour="black", fill="grey45")
          ylab("frequency") +
          ylim(0, 0.3) +
          scale_x_continuous(breaks = seq(0, 10, by = 1)) +
          theme(plot.title = element_text(hjust = 0.5)) +
          theme(text=element_text(size = 20))
pwhite_s <- ggplot(data=filter(df, race =="Caucasian"), aes(x=decile_score)) +
          geom_histogram(aes(y=..count../sum(..count..)), binwidth = 1,  colour="black", fill="grey100")
          ylab("frequency") +
          ylim(0, 0.3) +
          scale_x_continuous(breaks = seq(0, 10, by = 1)) +
          theme(plot.title = element_text(hjust = 0.5)) +
          theme(text=element_text(size = 20))

grid.arrange(pblack_s, pwhite_s, ncol = 2, top=textGrob("Scores in Black and White", vjust= 0.4, gp=gpar
```



Distribution of scores for whites:

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.