Guided Exercise 9 - Classification

Daniel Kwik

Setup

Construct the set of people, some who will repay and others who will not.

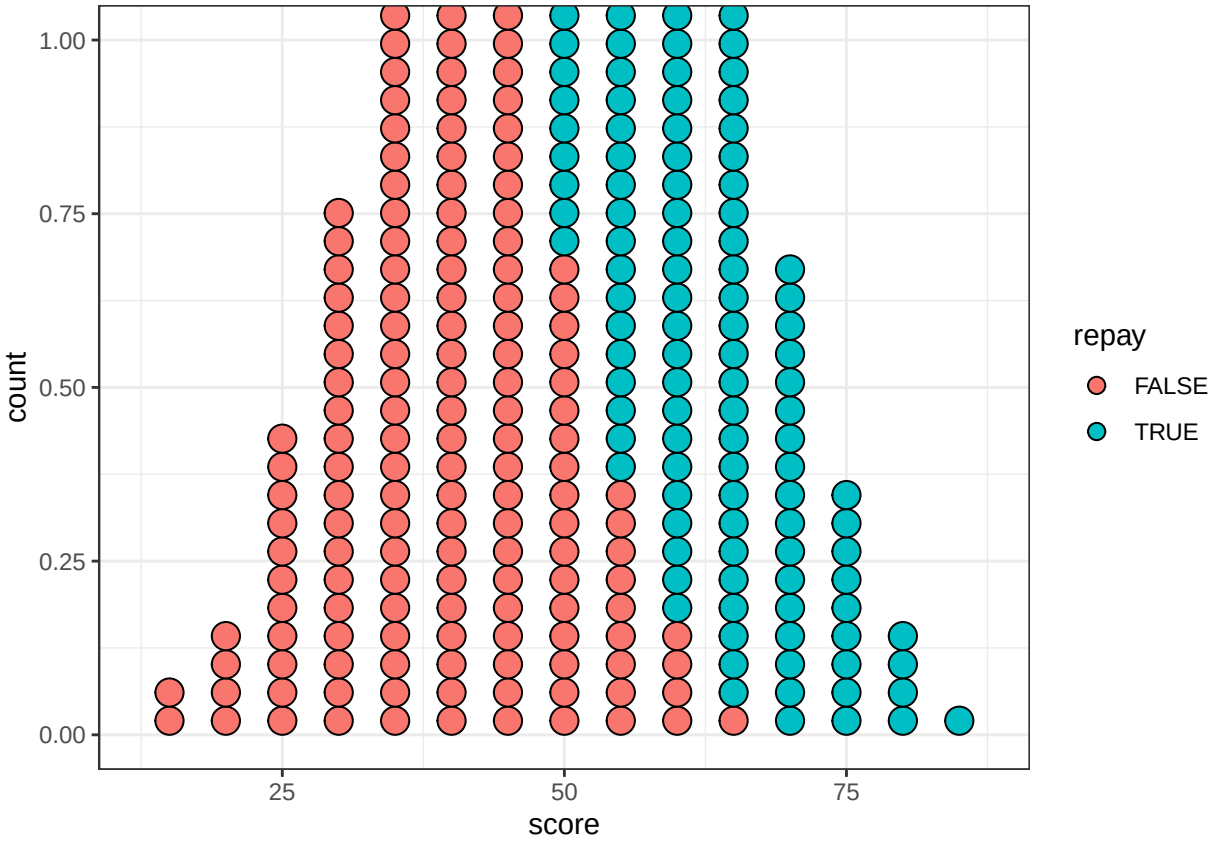
```
# "Deterministic normal": we expect n * pnorm(x, mu, sd) of n points to be below x.
make_group <- function(n, mean, sd, ...) {
  tibble(score = 0:100) %>%
    mutate(n = floor(n * pnorm(score, mean, sd))) %>%
    mutate(delta = c(diff(n), 0)) %>%
    uncount(delta) %>%
    select(score) %>%
    mutate(...)
}

people <- bind_rows(
  make_group(150, 60, 10, repay = TRUE),
  make_group(150, 40, 10, repay = FALSE)
)
people
```

```
# A tibble: 298 x 2
   score repay
   <int> <lgl>
 1    35 TRUE
 2    37 TRUE
 3    39 TRUE
 4    40 TRUE
 5    41 TRUE
 6    42 TRUE
 7    43 TRUE
 8    43 TRUE
 9    44 TRUE
10    44 TRUE
# ... with 288 more rows
```

Make a dotplot of the people.

```
ggplot(people, aes(x = score, fill = repay)) +
  geom_dotplot(binwidth = 5, dotsize = .5,
               binpositions = "all", method = "histodot", stackgroups = TRUE)
```



##Exercise 1

```
threshold <- 64
decisions <-
  people %>%
  mutate(granted = score >= threshold)

decisions
```

```
# A tibble: 298 x 3
   score repay granted
   <int> <lgl> <lgl>
 1    35 TRUE  FALSE
 2    37 TRUE  FALSE
 3    39 TRUE  FALSE
 4    40 TRUE  FALSE
 5    41 TRUE  FALSE
 6    42 TRUE  FALSE
 7    43 TRUE  FALSE
 8    43 TRUE  FALSE
 9    44 TRUE  FALSE
10    44 TRUE  FALSE
# ... with 288 more rows
```

##Exercise 2

```
num_app <- nrow(decisions)
total_granted <- nrow(decisions %>% filter(repay == TRUE))

decisions %>%
  summarize(positive = sum(granted)/num_app)
```

```
# A tibble: 1 x 1
  positive
     <dbl>
1    0.174
```

```
decisions %>%
  filter(repay == TRUE) %>%
  summarize(true_positive = sum(granted)/total_granted)
```

```
# A tibble: 1 x 1
  true_positive
          <dbl>
1         0.342
```

##Exercise 3 true negative - didn't get a loan and didn't pay false positive - got a loan, but didn't pay

```
true_negative <- decisions %>%
  filter(granted == FALSE, repay == FALSE)
false_positive <- decisions %>%
  filter(granted == TRUE, repay == FALSE)
true_positive <- decisions %>%
  filter(granted == TRUE, repay == TRUE)
false_negative <- decisions %>%
  filter(granted == FALSE, repay == FALSE)

specificity = nrow(true_negative)/(nrow(true_negative) + nrow(false_positive))

sensitivity = nrow(true_positive)/(nrow(true_positive) + nrow(false_negative))
```