Recodes, grouping, etc.

Frank Edwards

Data for today

```
dat<-read_csv(”./data/criminalrecord.csv”)
```

Recoding and conditionals

Let’s make distance categorical, with cuts at the 25th, 50th, and 75th quantile

```
summary(dat$distance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA’s
##    0.00    8.00   12.00   11.96   16.00   25.00       2
```

```
## NA???
```

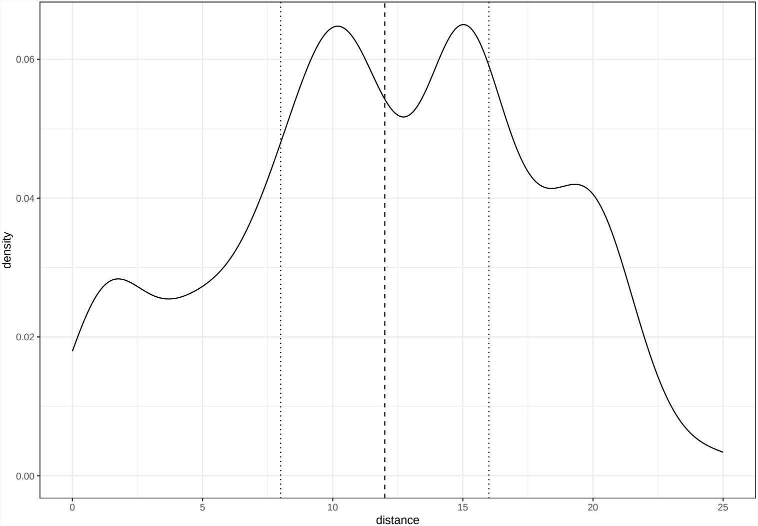
Filter out missing values

```r
## remove pesky NA values
dat_clean<-dat %>%
  filter(!(is.na(distance)))

summary(dat_clean$distance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    8.00   12.00   11.96   16.00   25.00
```

Visualizing quantiles: area under the curve



Making a recode with one condition

Make a new variable for distance, with value T if below the median, and F if above

```
dat_clean<-dat_clean %>%
  mutate(distance_binary = distance < median(distance))
```

Making a recode with one condition: ifelse()

Make a new variable for distance, with value “near” if below the median, and “far” if above

```
dat_clean<-dat_clean %>%
  mutate(distance_binary2 = ifelse(
    distance < median(distance),
    ”near”,
    ”far”
  ))
```

Making a recode with multiple conditions

```r
### define quartile cut points
q1<-quantile(dat_clean$distance, 0.25)
q2<-quantile(dat_clean$distance, 0.5)
q3<-quantile(dat_clean$distance, 0.75)
q1; q2; q3
```

```
## 25%
##   8
```

```
## 50%
##  12
```

```
## 75%
##  16
```

Making a recode with multiple conditions: case_when()

```
### make factor variable
dat_clean <- dat_clean %>%
  mutate(distance_quartile =
           case_when(
             distance < q1 ~ ”1st”,
             distance < q2 ~ ”2nd”,
             distance < q3 ~ ”3rd”,
             distance >= q3 ~ ”4th”
           ))
```

Practice recodes

- Recode crimrec to be equal to “record” if 1; and “no record” if 0
- Recode distance to be ‘far’ if at the 80th percentile; ‘kinda’ if at the 65th percentile, and ‘close’ otherwise
- Create a new variable called ‘race’ and use values from ‘black’ to create sensible values.

Summary operations

Load in the UCR data

```
dat<-read_tsv(”./data/ICPSR_39063/DS0002/39063-0002-Data.tsv”)
```

Aggregating data

We can use `group_by` to aggregate across categories.

Paired with `summarize` we can compute summary statistics for sub-groups of the data.

Practice

- First compute the rate of homicide arrests per 100,000 population for M25_29
- Then compute the average homicide arrest rate at the state-level
- Make a histogram of this average state-level homicide rate

Operations across columns

We can use `rowSums()` to compute the sum of values across multiple columns (by row!)

We pair it with `pick()` to use syntax like `select()` inside of a `mutate` or `summarize`

```r
dat<-dat %>%
  mutate(m_arrest = rowSums(pick(M0_9:M65)))
```

Practice

- Create new variables that contain the total number of arrests for men and women for each agency - offense code pair
- Compute the state-level median arrest rate (per 100,000) by sex for each offense code
- Subset the data to aggravated assault arrests; then visualize the distribution of state-level arrest rates by sex using a histogram (consider how to best visualize the distribution(s))