lecture 4

leah brooks

February 9, 2021

Today

A. Heads-up: Bigger Data

B. If-else recap

C. Histograms

D. Results by group: groupings and facets

A. Bigger Data

- You need to work with more data than you can see in a window
- Today's tutorial has techniques to do this
- Look to summary statistics

A. Looking at crashes

```
dim(crash)
```

```
## [1] 59777    44
```

```
table(crash$Light)
```

```
##
## DARK -- UNKNOWN LIGHTING           DARK LIGHTS ON           DARK NO LIGH
##                      660                    13971                      2
##                     DAWN                 DAYLIGHT                     DU
##                     1239                    39305                     13
##                      N/A                    OTHER                   UNKNO
##                      497                      143                      4
```

- look at the total size of the dataset

A. A Legible Version

```
## # A tibble: 9 x 2
##   Light                      light_type
##   <fct>                           <int>
## 1 DARK -- UNKNOWN LIGHTING          660
## 2 DARK LIGHTS ON                  13971
## 3 DARK NO LIGHTS                   2158
## 4 DAWN                             1239
## 5 DAYLIGHT                        39305
## 6 DUSK                             1393
## 7 N/A                               497
## 8 OTHER                             143
## 9 UNKNOWN                           411
```

B. A Key Programming Command: `ifelse()`

```
df$var <- ifelse(test = [condition with ==],
                 yes =  [do if condition true],
                 no =   [do if condition false])
```

B. An Example, 1 of 3

```
ex <- data.frame(building = c("A","B","C"),
                 yb = c("1983","1989","2005"))
ex
```

```
##   building   yb
## 1        A 1983
## 2        B 1989
## 3        C 2005
```

What if I want to know the century in which each building is built?

B. An Example, 2 of 3

```
ex$c <- ifelse(test = ex$yb < 2000,
               yes = "20th",
               no = "21st")
```

```
## Warning in Ops.factor(ex$yb, 2000): '<' not meaningful for factors
```

B. An Example, 3 of 3

```
ex$c <- ifelse(test = as.numeric(as.character(ex$yb)) < 2000,
               yes = "20th",
               no = "21st")
```

B. An Example, 3 of 3

```
ex$c <- ifelse(test = as.numeric(as.character(ex$yb)) < 2000,
               yes = "20th",
               no = "21st")
```

```
table(ex$c)
```

```
##
## 20th 21st
##    2    1
```

B. An Example, 3 of 3

```
ex$c <- ifelse(test = as.numeric(as.character(ex$yb)) < 2000,
               yes = "20th",
               no = "21st")
```

```
table(ex$c)
```

```
##
## 20th 21st
##    2    1
```

What could go wrong with programming like this?

B. Some rules of thumb for `ifelse()`

- check your output!

B. Some rules of thumb for `ifelse()`

- check your output!
- a test can include multiple conditions
- good idea to define all cases – don't let a case be the residual

B. Some rules of thumb for `ifelse()`

- check your output!
- a test can include multiple conditions
- good idea to define all cases – don't let a case be the residual
- you can nest `ifelse()` commands:

```
ex$ybn <- as.numeric(as.character(ex$yb))
summary(ex$ybn)
ex$c <- ifelse(test = ex$ybn >= 1900 & ex$ybn < 2000,
               yes = "20th",
               no = ifelse(test = ex$ybn >= 2000 & ex$ybn < 2100)
                           yes = "21st"
                           no = "trouble"))
```

C. Histograms

We will use three new geoms this lecture

- `geom_histogram()`
- `geom_density()`
- `geom_freqpoly()`

C.1. How to create a histogram

Use

```
geom_histogram(data = [dataframe],
               mapping = aes(x = [variable]))
```

- only need to list one variable
- histograms are univariate graphics
- `geom_histogram()` is best for a distribution with limited values

C.1. How to create a histogram

Use

```
geom_histogram(data = [dataframe],
               mapping = aes(x = [variable]))
```

- only need to list one variable
- histograms are univariate graphics
- `geom_histogram()` is best for a distribution with limited values
- but not a categorical distribution, which should be a bar

C.2. Histogram options

- fill overall: outside `aes`, `fill = [color]`
- fill by group: inside `aes`, `fill = [variable]`
- bin width: `bin_width = [unit span]`,
- by groups: inside `aes`, `color = [grouping variable]`

C.3. Approximating Continuous Distributions

For almost-continuous bins, use

```
geom_freqpoly()
```

For much more smoothing, use

```
geom_density()
```

C.4. Example

- take crash-level data from last class
- use `group_by()` and `summarize()` to make daily data
- count number of crashes by day

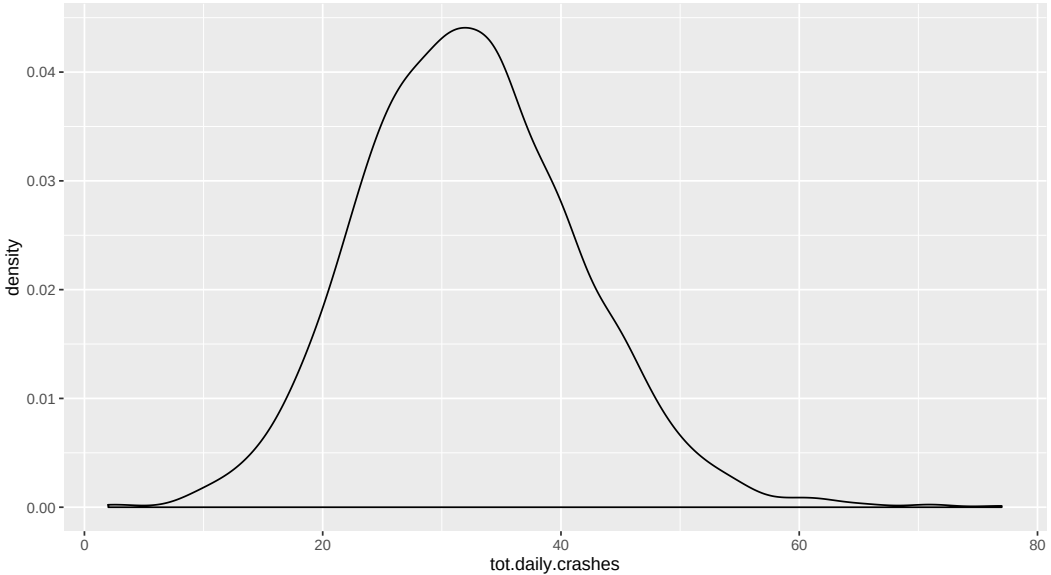
```
# add up total number of crashes by date
crash2 <- group_by(.data = crash, date2)
crash2 <- summarize(.data = crash2, tot.daily.crashes = n())
table(crash2$tot.daily.crashes)
```

```
##
##  2  3  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
##  1  1  1  2  4  5  3  8  8 11 12 20 19 33 26 44 43 57 61 65 76 73 76 74
## 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
## 80 76 85 89 66 59 61 55 59 48 38 27 37 39 18 25 18 11 14  8  8  6  6  5
## 59 60 61 62 63 65 66 71 77
##  1  2  2  2  1  1  1  2  1
```

Plot these data

```
alld <- ggplot() +
  geom_density(data = crash2,
               mapping = aes(x = tot.daily.crashes))
```

Plot these data



D. Results by Group

```
# find the day of the week
crash2$day.of.week <- weekdays(x = crash2$date2)

# check
table(crash2$day.of.week)
```

```
##
##    Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
##       264       264       264       264       265       264       264
```

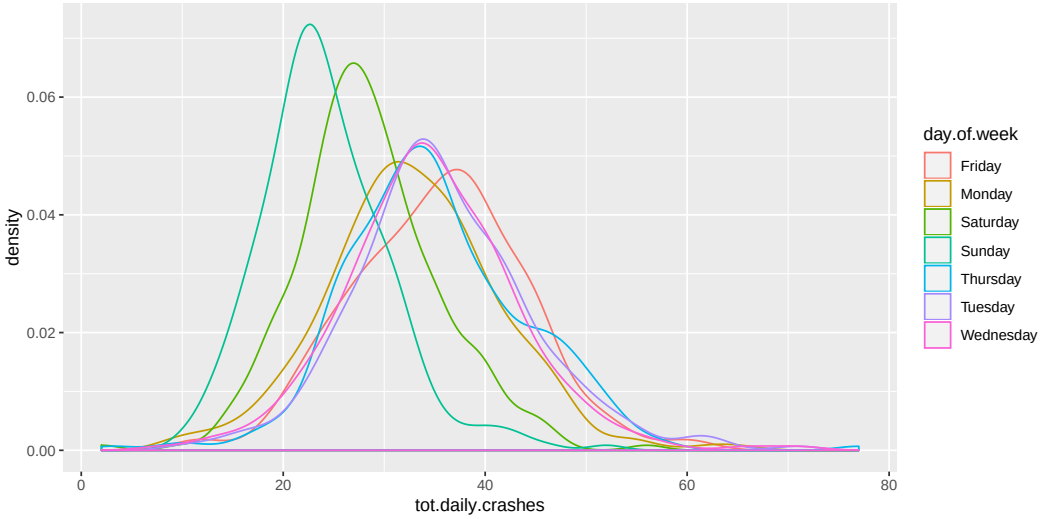
- you need a variable that indicates a group
- then plot distribution by group
- we'll use distribution of traffic accidents (x variable)
- by weekday (grouping variable)

By day of the week

```
wd <- ggplot() +
  geom_density(data = crash2,
               mapping = aes(x = tot.daily.crashes,
                             color = day.of.week))
```

By day of the week

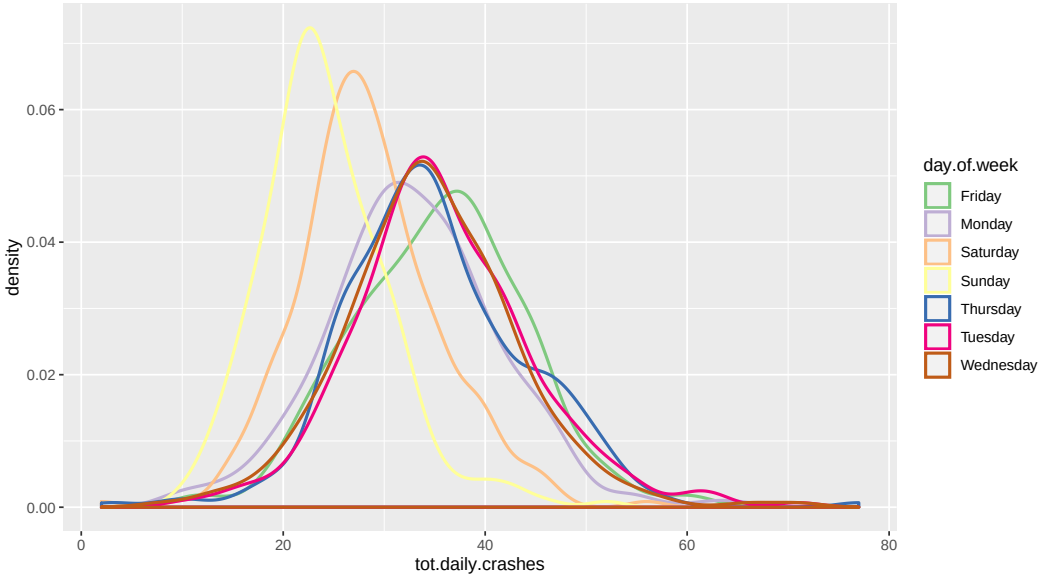
```
wd
```



By day of the week, better colors and thicker lines

```
day.colors <- c('#7fc97f','#beaed4','#fdc086','#ffff99',
                '#386cb0','#f0027f','#bf5b17')
wd <- ggplot() +
  geom_density(data = crash2,
               mapping = aes(x = tot.daily.crashes,
                             color = day.of.week),
               size = 0.9) +
  scale_color_manual(values = day.colors)
```

By day of the week, better colors and thicker lines



By day of the week, facets

```
wd <- ggplot() +
  geom_density(data = crash2,
               mapping = aes(x = tot.daily.crashes,
               group = day.of.week)) +
  facet_wrap(~day.of.week)
```

By day of the week, facets

```
wd
```