Data Visualization with ggplot2

Session 3: Visualizing Diverse Data

Nathan Barron
nathanbarron@ou.edu

Different data need different visualizations

data | mpg | cyl | am | cyl x am

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

Types of data

Levels of Data Measurement

- Categorical
  - Binary
  - Non-binary
- Continuous

. Cat.: Binary Cat.: Non-binary

Continuous

Click through each header to read more about each type

Storing Data in R

- Boolean
- Character
- Numerical
- Factor

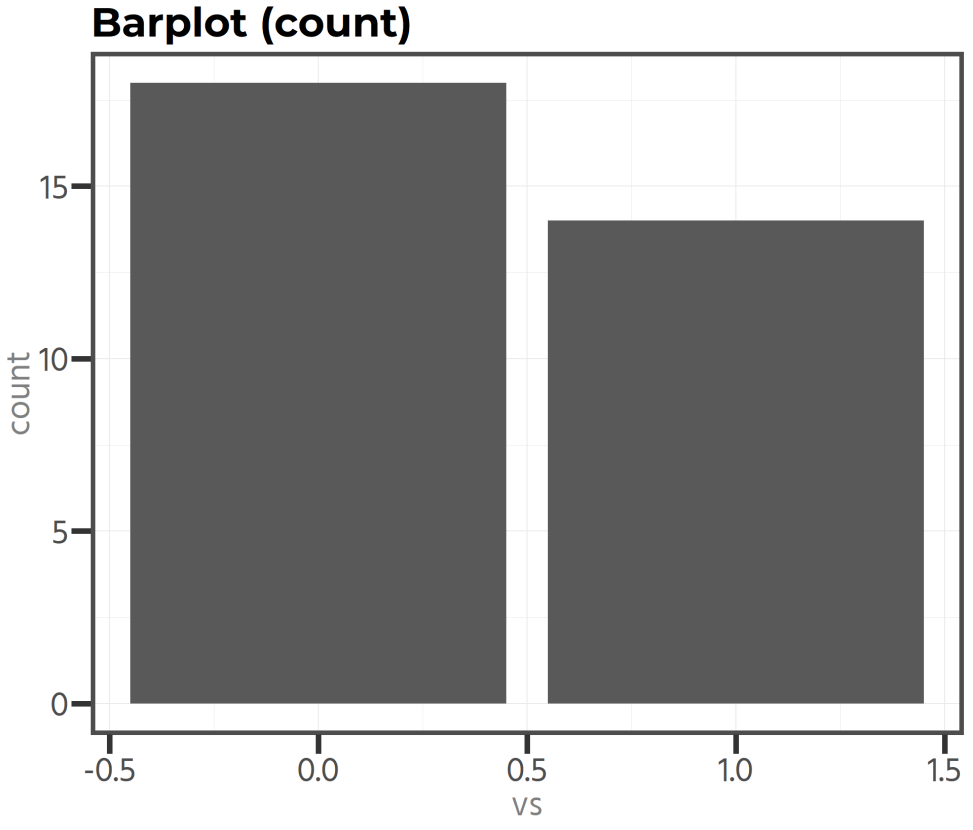
. Boolean Character Numerical

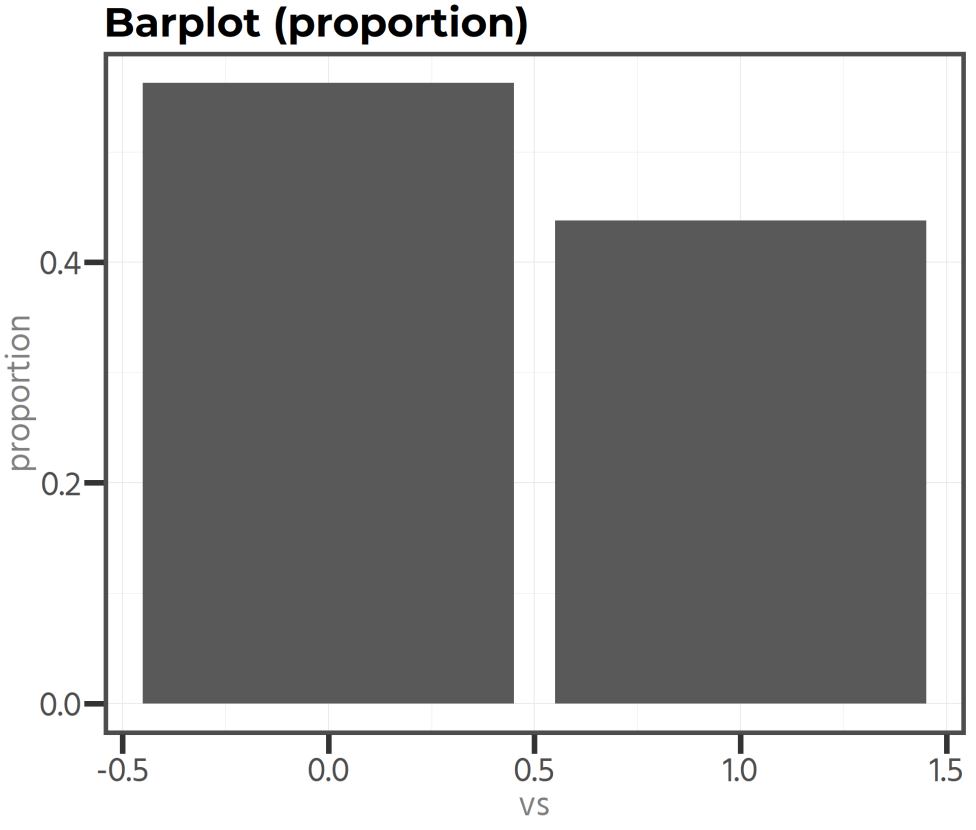
Factor Dates

Click through each header to read more about each type

Visualizing a single variable

Binary



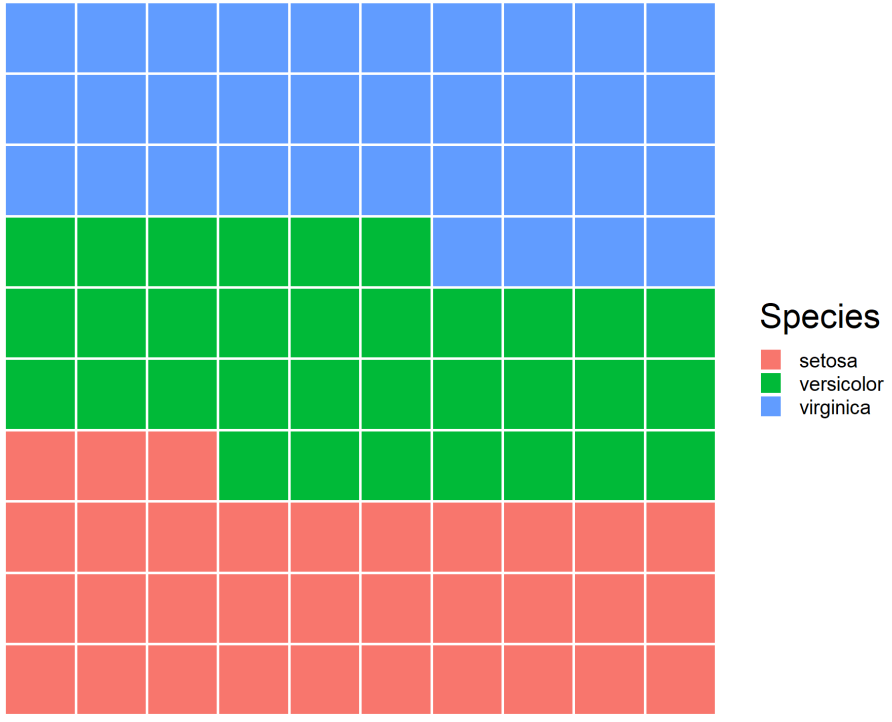



```r
# Barplot with frequency on y-axis
ggplot(mtcars) +
  geom_bar(aes(x=vs))
```

```r
# Barplot with proportion on y-axis
ggplot(mtcars) +
  geom_bar(aes(x=vs, y=..count../sum(..count..)))
```

Non-binary Categorical

Waffle plot

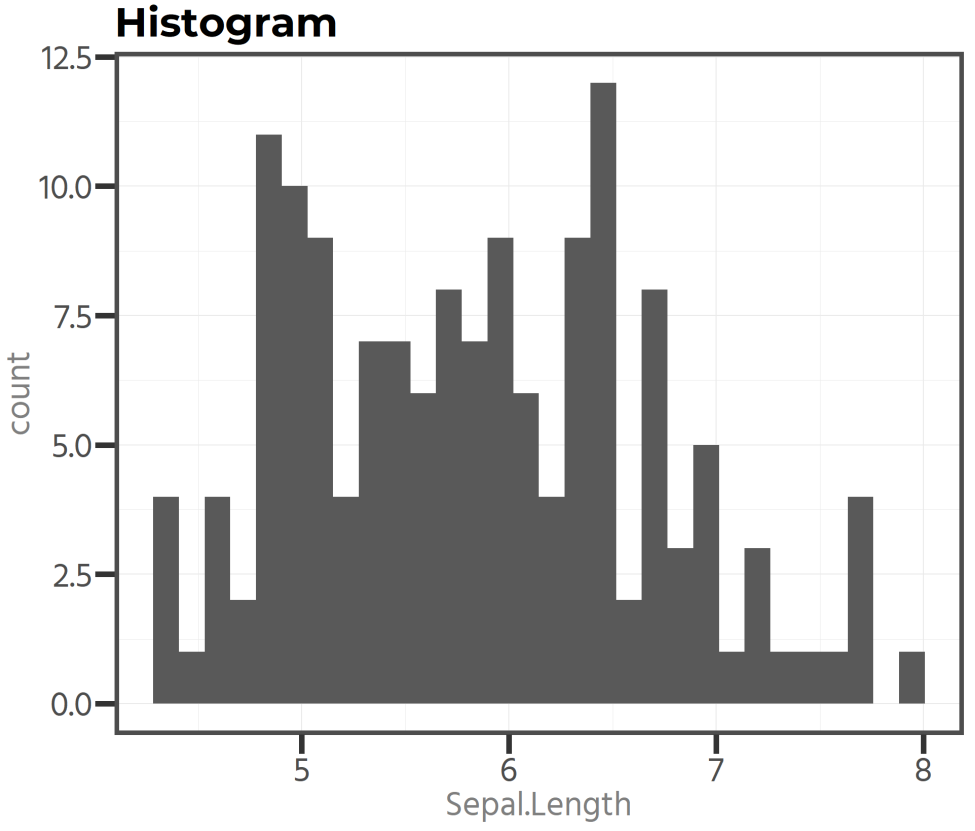


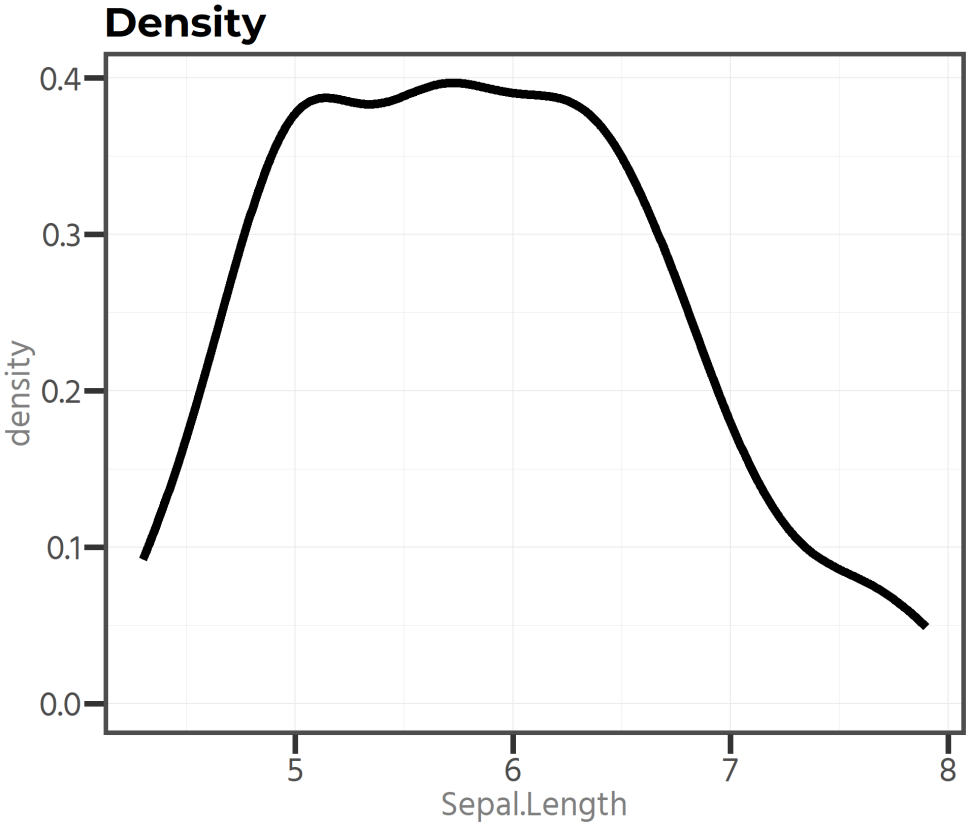

```r
# install.packages("waffle",
#                  repos = "https://cinc.rud.is")

library(waffle)
library(dplyr)

iris %>%
  count(Species) %>%
  ggplot(aes(fill = Species, values = n)) +
  geom_waffle(size = 1,
              colour = "white",
              na.rm=TRUE,
              flip = TRUE,
              make_proportional = TRUE) +
  theme_void() +
  coord_equal()
```

Continuous



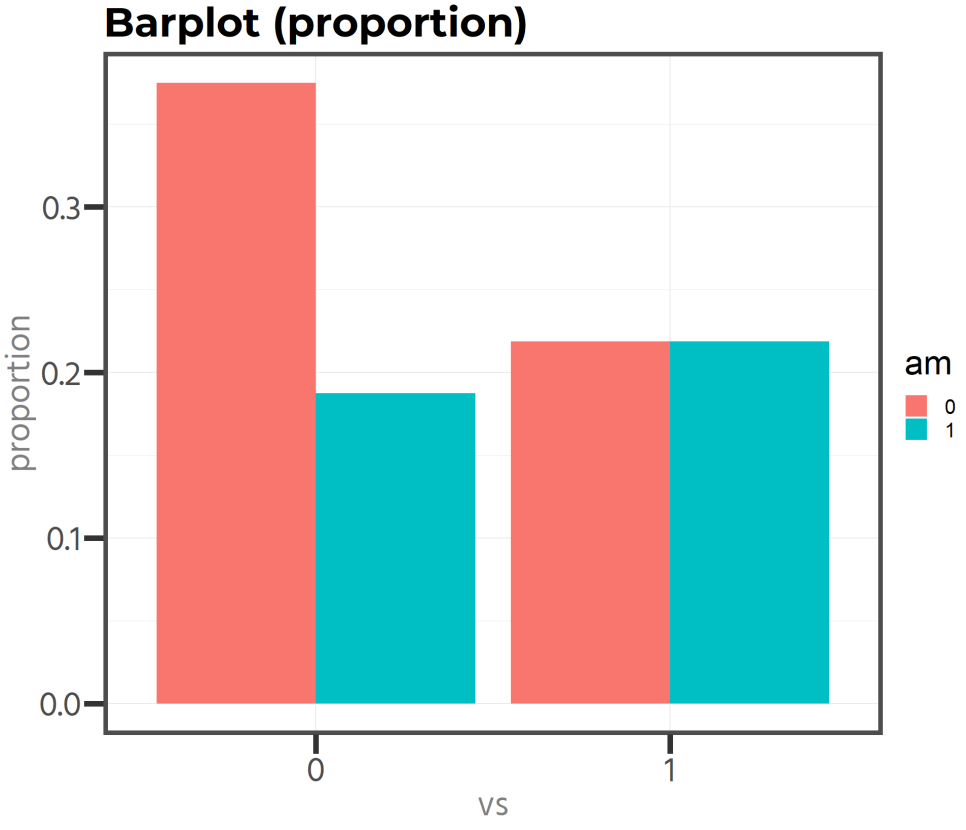



```
ggplot(iris) +
  geom_histogram(aes(x=Sepal.Length))
```

```
ggplot(iris) +
  geom_density(aes(x=Sepal.Length))
```

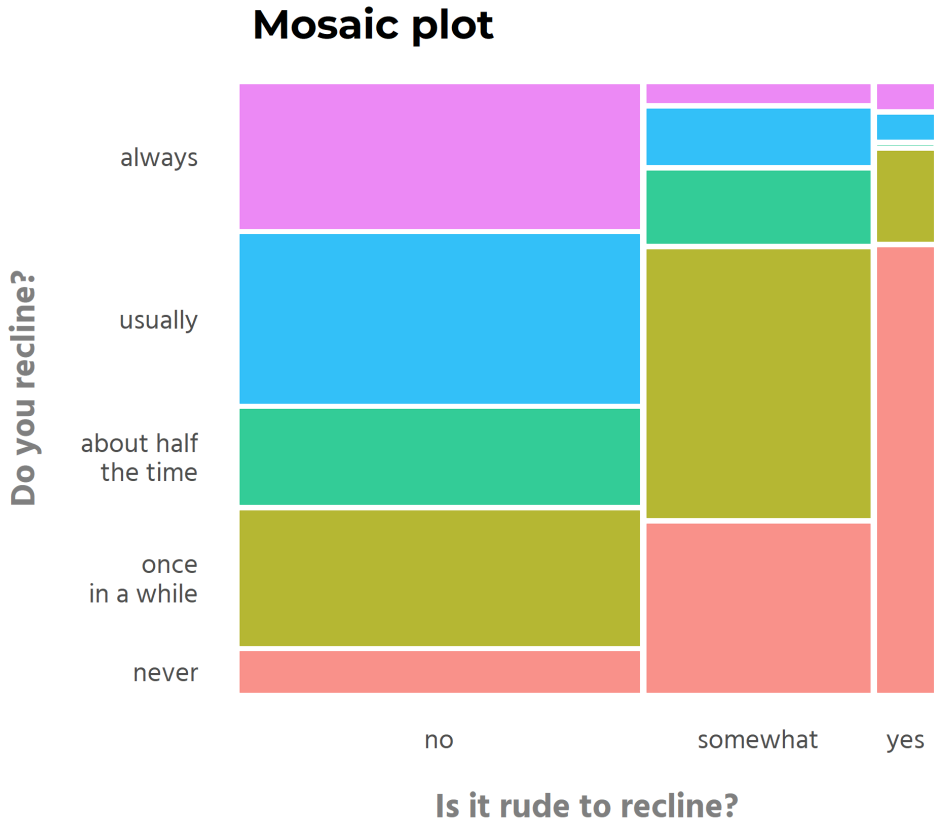
Visualizing multiple variables

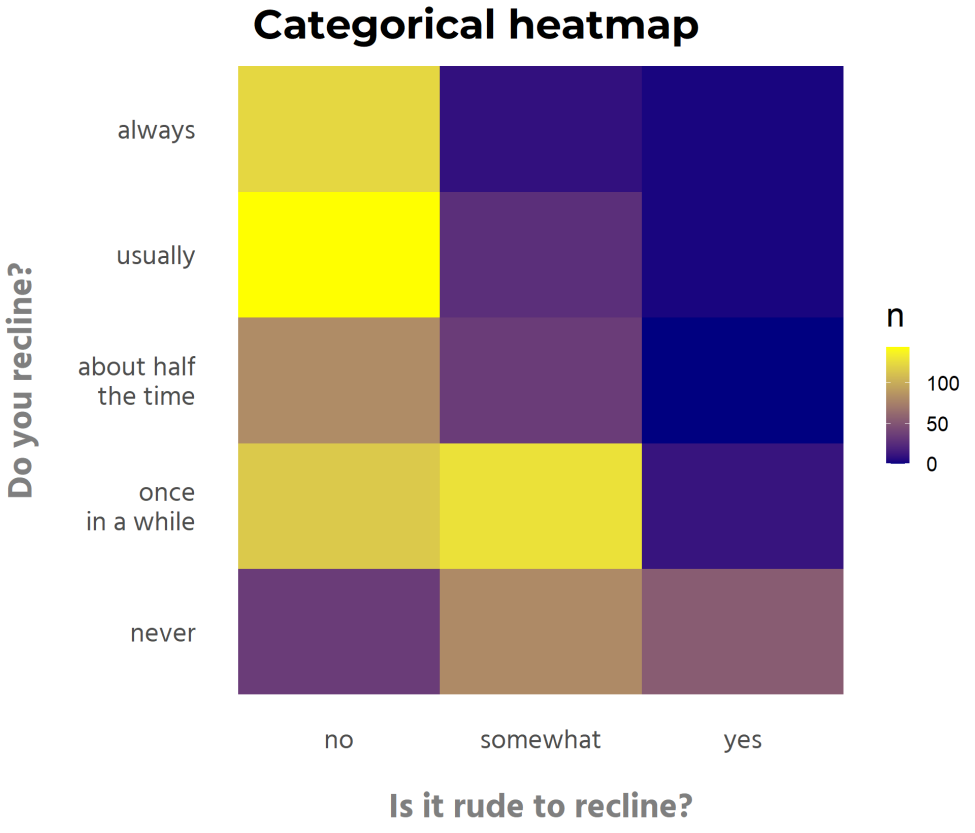
Multiple categorical variables




```
mtcars %>%
  mutate(vs = factor(vs), am = factor(am)) %>%
  ggplot(.) +
  geom_bar( aes(x=vs,
                y=..count../sum(..count..),
                group=am,
                fill=am),
            position = 'dodge')
```

Multiple categorical variables II



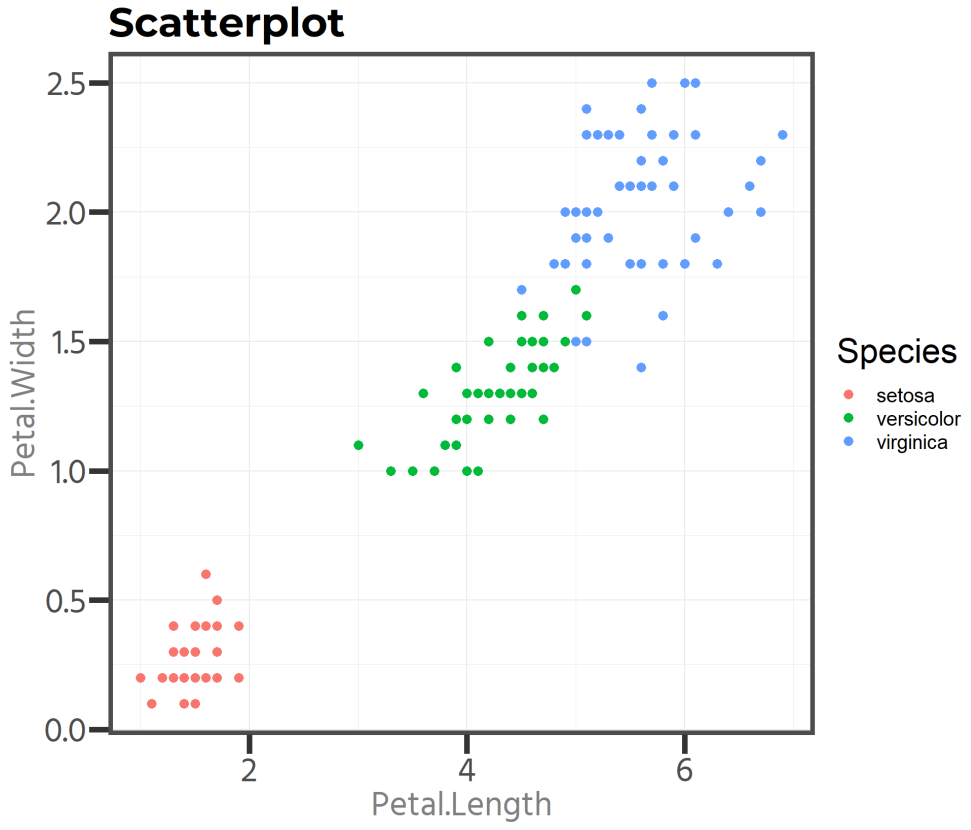


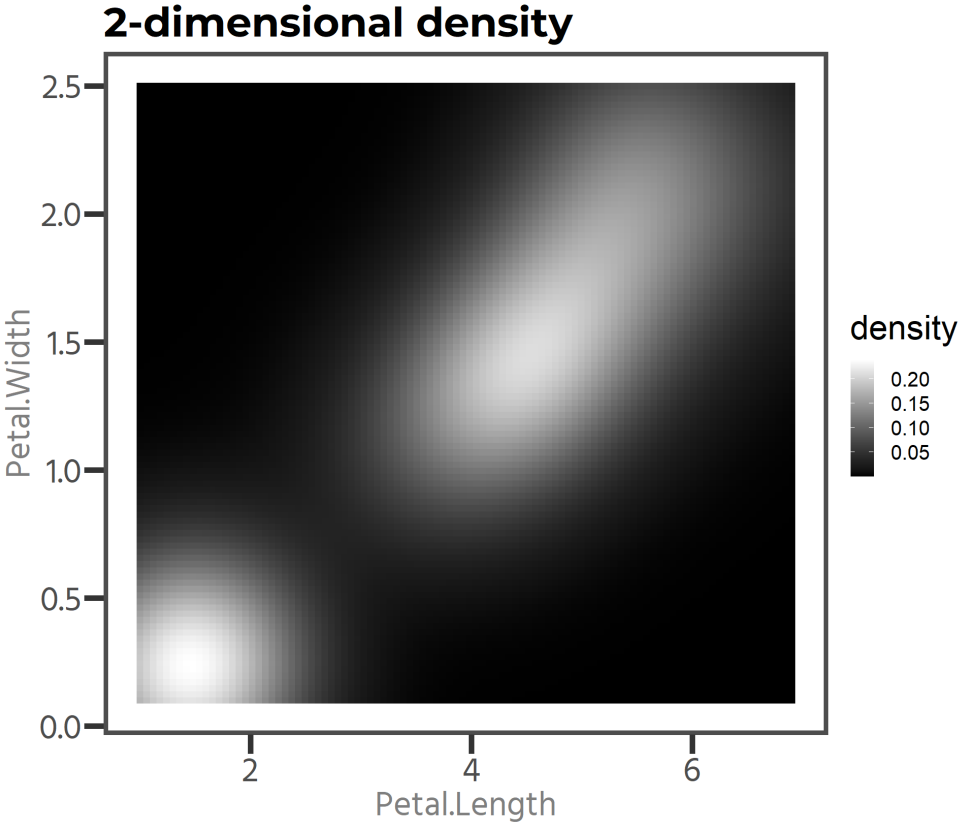

```
# library(ggmosaic)
# library(tidyr)

ggplot(ggmosaic::fly) +
  geom_mosaic(aes(x = product(do_you_recline, rude_to_rec
  theme(panel.grid = element_blank(),
        legend.position = 'none')
```

```
ggmosaic::fly %>%
  count(rude_to_recline, do_you_recline, .drop = FALSE) %
  ggplot(.) +
  geom_tile(aes(x = rude_to_recline,
                y = do_you_recline,
                fill = n)) +
  scale_fill_gradient(low = 'navy', high = "yellow")
```

Multiple continuous variables





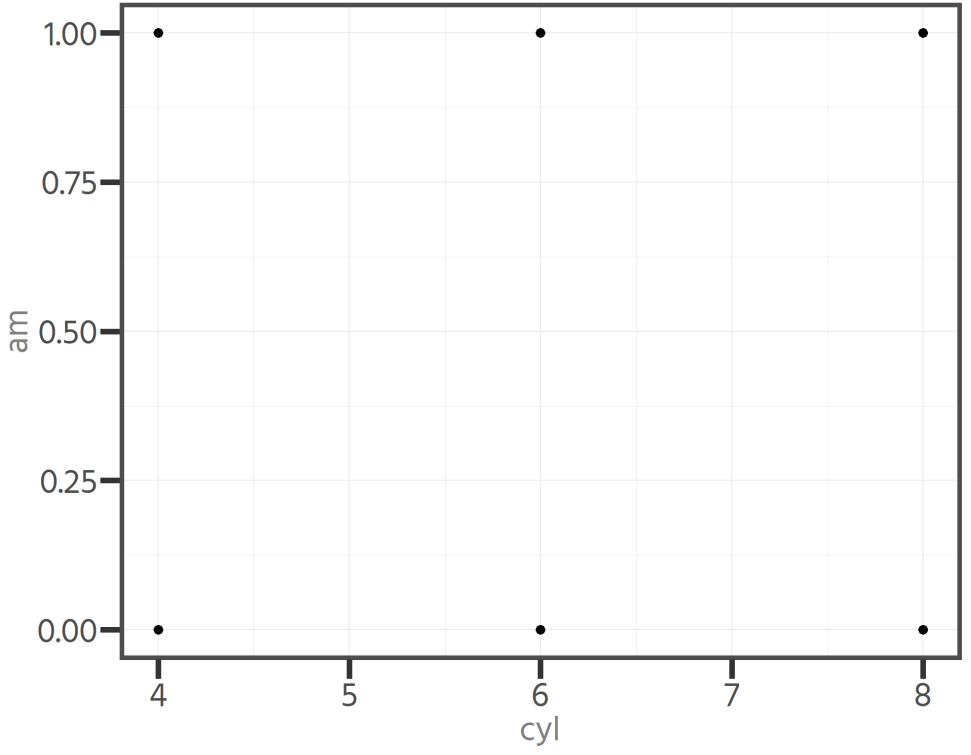

```
ggplot(iris) +
  geom_point(aes(x=Petal.Length,
                 y=Petal.Width,
                 color=Species))
```

```
ggplot(iris, aes(x=Petal.Length,y=Petal.Width)) +
  stat_density2d(aes(fill = ..density..),
                 geom = "raster", contour = FALSE) +
  scale_fill_gradient(low = "black", high = "white")
```

Hybrid (some categorical, some continuous)

Overplotting

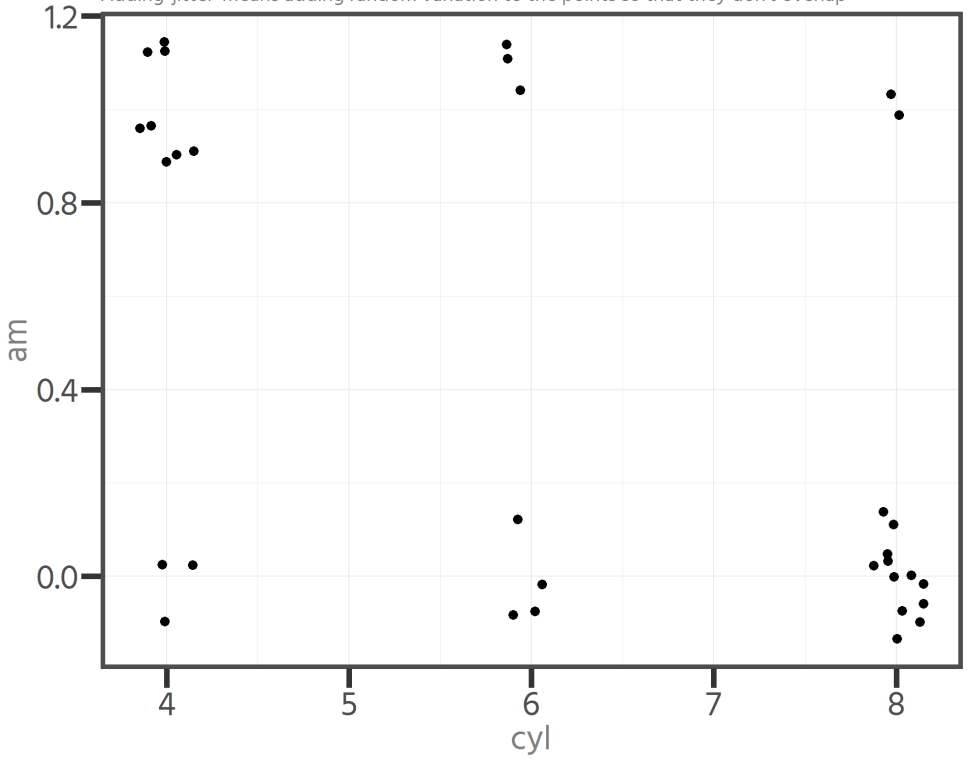
Overplotting is when data or labels overlap, meaning that the viewer loses important information




```
ggplot(mtcars) +
  geom_point(aes(x=cyl,y=am))
```

Jitterplot

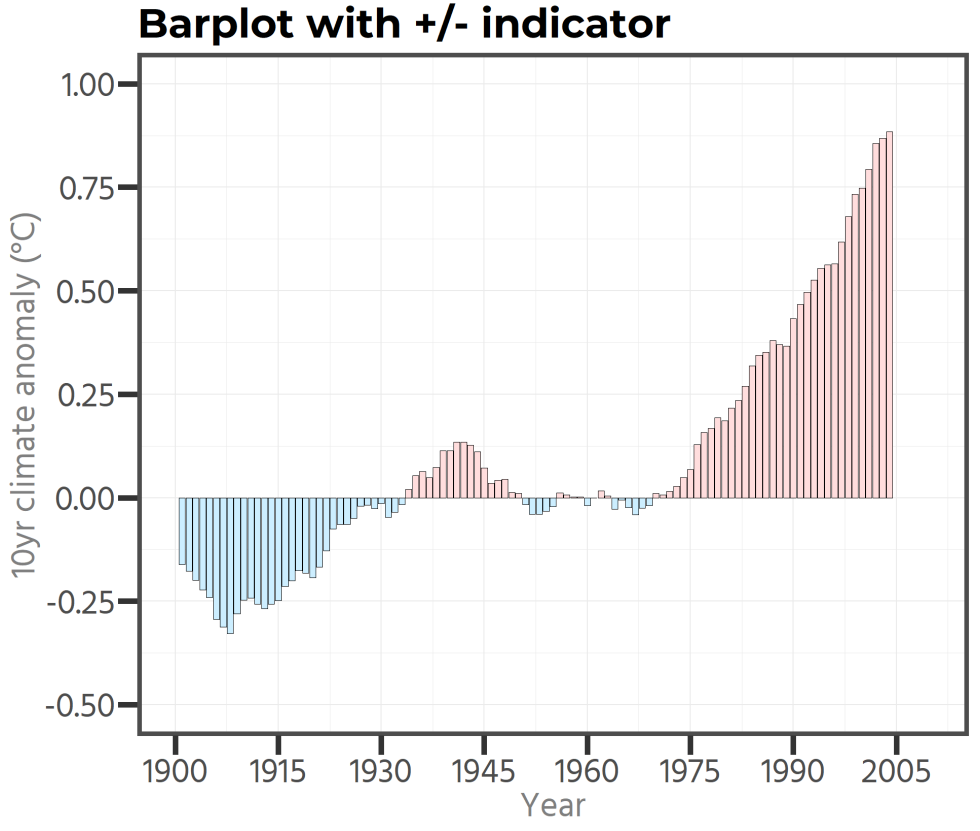
Adding 'jitter' means adding random variation to the points so that they don't overlap

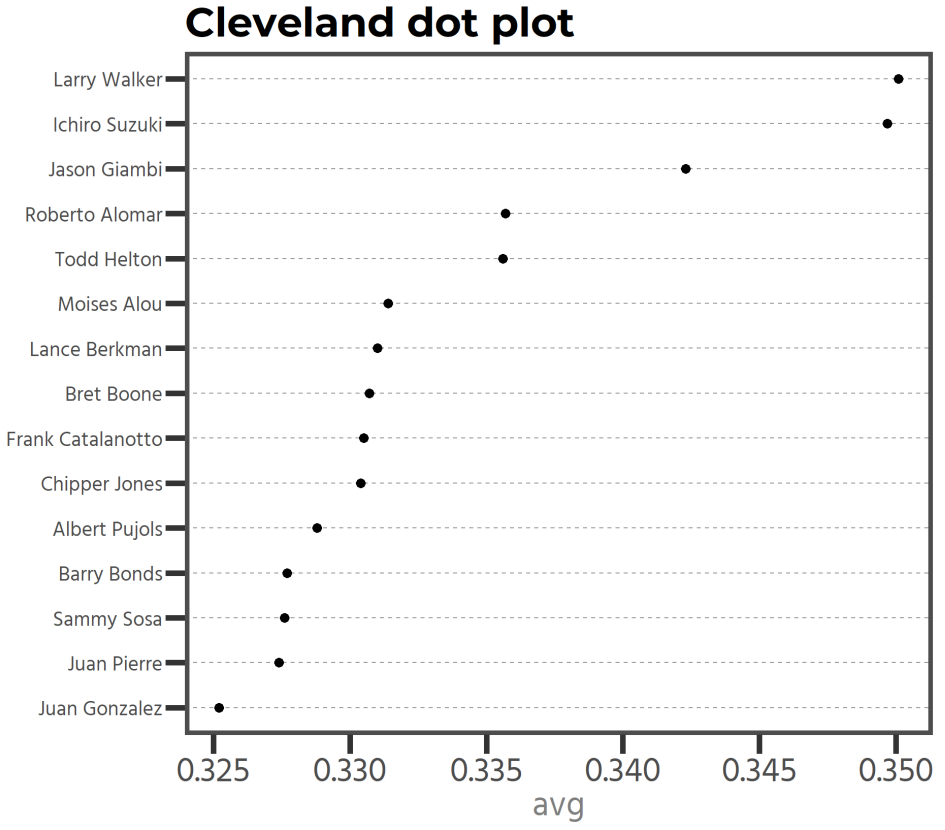



```
ggplot(mtcars) +
  geom_jitter(aes(x=cyl,y=am))

# You can also specify 'width' and 'height'
# as additional arguments to refine the
# jitter. Remember, jitter is random:
# jitter plots will change whenever you
# re-run the code.
```

Hybrid II






```r
library(gcookbook)
climate_sub <- climate %>%
  filter(Source == "Berkeley" & Year >= 1900) %>%
  mutate(pos = Anomaly10y >= 0)

ggplot(climate_sub, aes(x = Year, y = Anomaly10y, fill =
  geom_col(position = "identity", colour = "black", size
  scale_fill_manual(values = c("#CCEEFF", "#FFDDDD"), gui
```

```r
library(gcookbook)
tophit <- tophitters2001[1:15, ]

ggplot(tophit) +
  geom_point(aes(x = avg, y = reorder(name, avg)))

# The `reorder` function is helpful
# when making plots of a non-binary categorical
# variable and a corresponding continuous variable.
```

There is no “one size fits all”

Data viz is creative ...

and creativity requires inspiration.

Check out these cool resources:

- R graphics gallery
- Nightingale: The Magazine of the Data Visualization Society
- Daydreaming Numbers Blog

Data viz is also programming ...

and programming requires code.

Check out these cool resources:

- R Graphics Cookbook
- R Graph Gallery