Week 4: *Exploring Data*

EMSE 4572/6572: Exploratory Data Analysis

John Paul Helveston

September 18, 2024

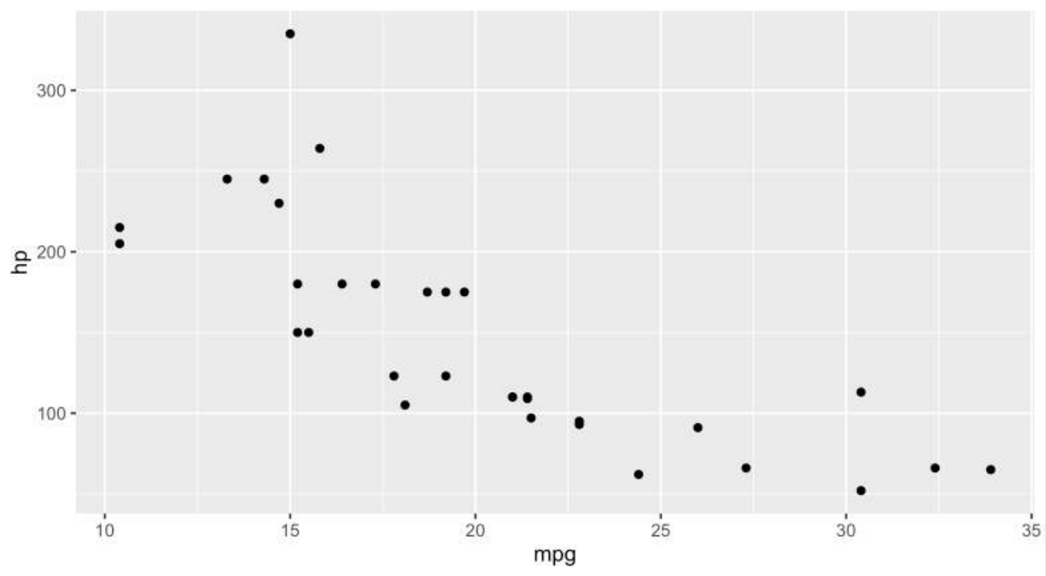
Quiz solution

Tip of the week:

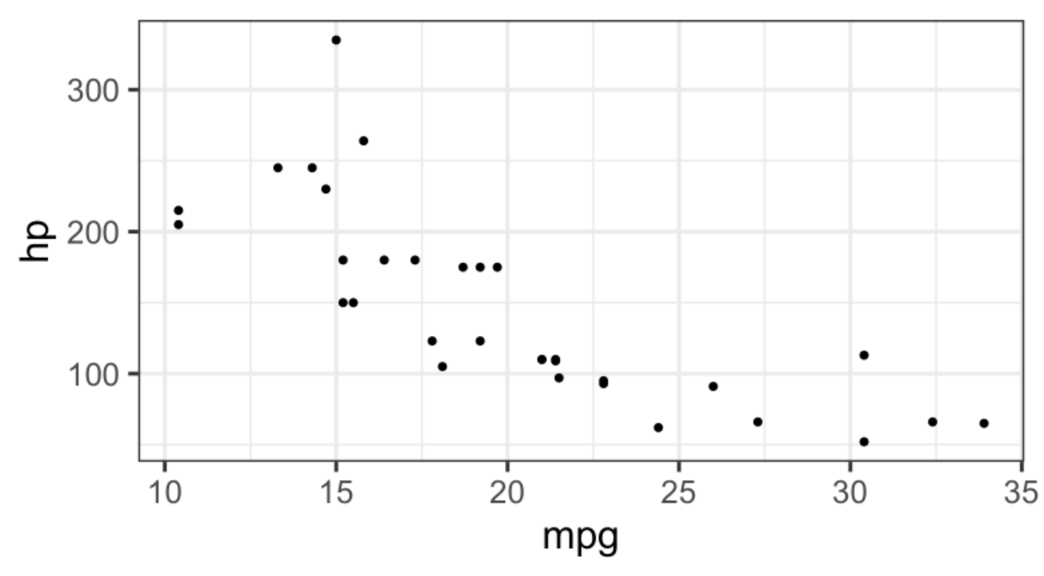
`theme_set()`

```
ggplot(mtcars) +
  geom_point(aes(x = mpg, y = hp))
```

Default theme



`theme_bw(base_size = 20)`



Week 4: *Exploring Data*

1. Exploring Data
2. Data Types
3. Centrality & Variability
4. Visualizing Centrality & Variability

BREAK

5. Correlation
6. Visualizing Correlation
7. Visualizing Relationships

Week 4: *Exploring Data*

1. Exploring Data
2. Data Types
3. Centrality & Variability
4. Visualizing Centrality & Variability

BREAK

5. Correlation
6. Visualizing Correlation
7. Visualizing Relationships

Exploratory Analysis

Goal: **Form** hypotheses.

Improves quality of **questions**.

(what we do in THIS class)

Confirmatory Analysis

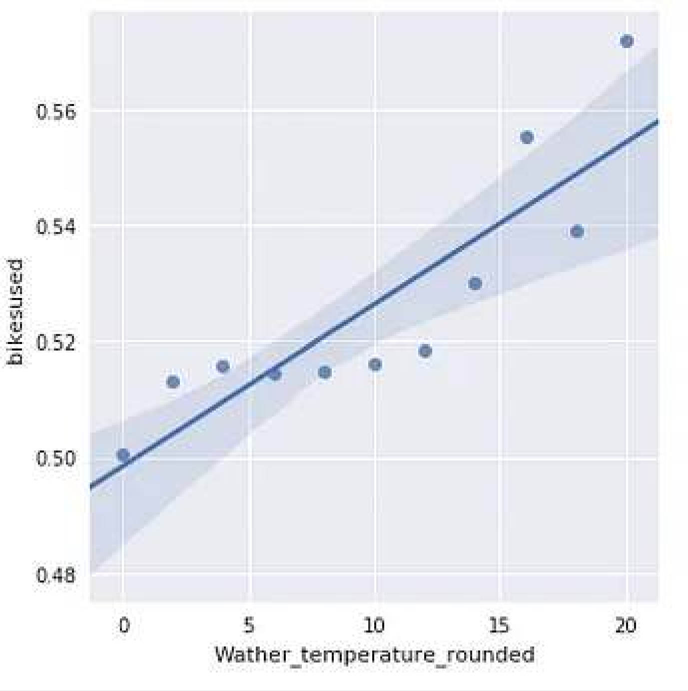
Goal: **Test** hypotheses.

Improves quality of **answers**.

(what you do in a stats class)

Exploratory Analysis

RQ: Do people bike more when the weather is nice?



Confirmatory Analysis

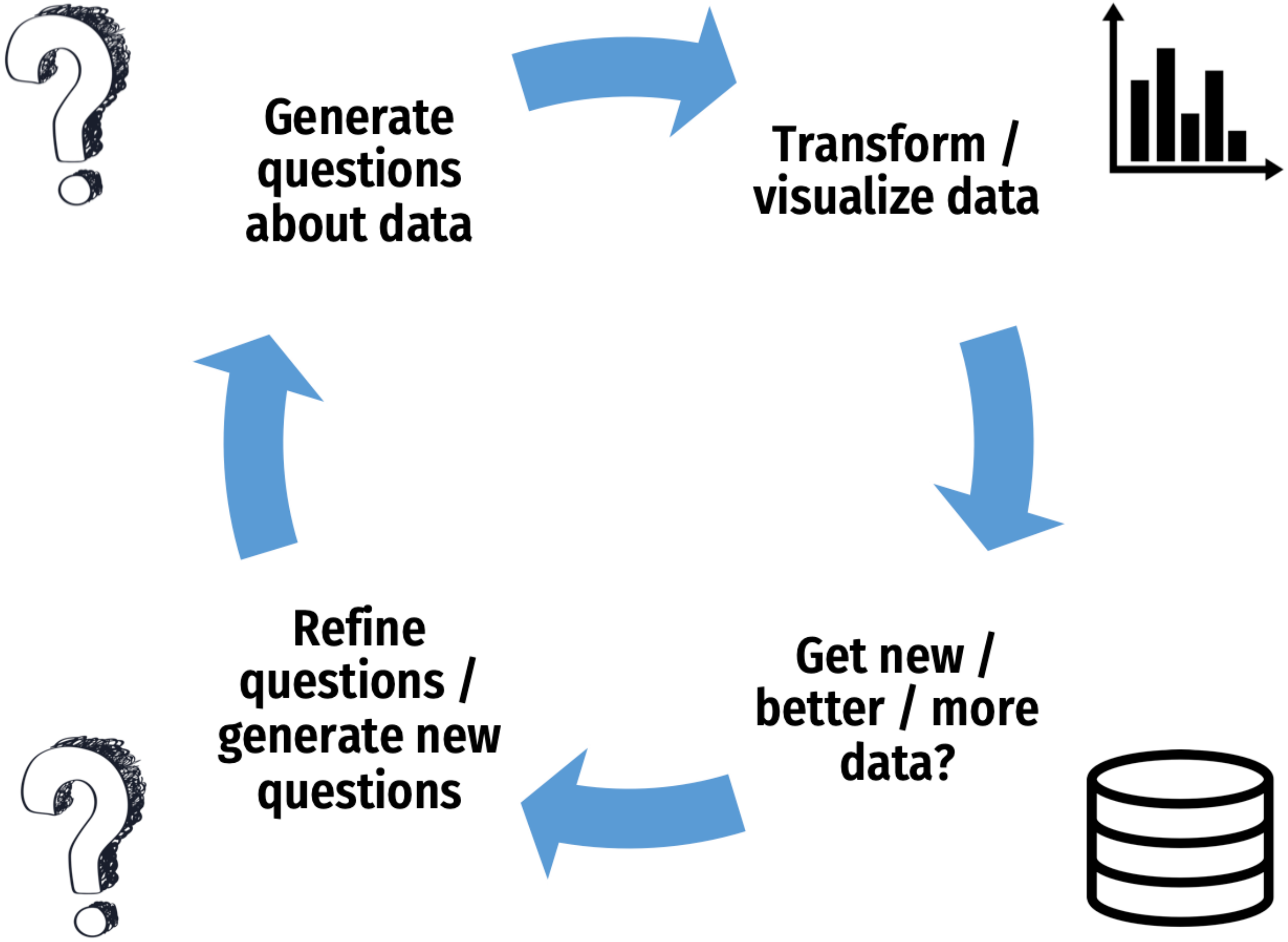
Let's build a model to predict bike usage based on weather.

Don't be Icarus

"An *approximate* answer to the *right* question is better than an *exact* answer to the *wrong* question."

— John Tukey

EDA is an iterative process to help you *understand* your data and ask better questions



Week 4: *Exploring Data*

1. Exploring Data
2. Data Types
3. Centrality & Variability
4. Visualizing Centrality & Variability

BREAK

5. Correlation
6. Visualizing Correlation
7. Visualizing Relationships

24,901

Earth's circumference at the equator:

24,901

Earth's circumference at the equator:
24,901 miles

Types of Data

Categorical

Subdivide things into *groups*

- What type?
- Which category?

Numerical

Measure things with numbers

- How many?
- How much?

Categorical (discrete) variables

Nominal

- Order doesn't matter
- Differ in "name" (nominal) only

e.g. `country` in TB case data:

```
#> # A tibble: 6 × 4
#>   country      year  cases population
#>   <chr>       <dbl>  <dbl>      <dbl>
#> 1 Afghanistan  1999    745   19987071
#> 2 Afghanistan  2000   2666   20595360
#> 3 Brazil       1999  37737  172006362
#> 4 Brazil       2000  80488  174504898
#> 5 China        1999 212258 1272915272
#> 6 China        2000 213766 1280428583
```

Ordinal

- Order matters
- Distance between units not equal

e.g.: `Placement` 2017 Boston marathon:

```
#> # A tibble: 6 × 3
#>   Placement `Official Time` Name
#>       <dbl> <time>          <chr>
#> 1         1 02:09:37        Kirui, Geo
#> 2         2 02:09:58        Rupp, Gale
#> 3         3 02:10:28        Osako, Sug
#> 4         4 02:12:08        Biwott, Sh
#> 5         5 02:12:35        Chebet, Wi
#> 6         6 02:12:45        Abdirahman
```

Numerical data

Interval

- Numerical scale with arbitrary starting point
- No "0" point
- Can't say "x" is double "y"

e.g.: `temp` in Beaver data

```
#>   day time  temp activ
#> 1 346  840 36.33     0
#> 2 346  850 36.34     0
#> 3 346  900 36.35     0
#> 4 346  910 36.42     0
#> 5 346  920 36.55     0
#> 6 346  930 36.69     0
```

Ratio

- Has a "0" point
- Can be described as percentages
- Can say "x" is double "y"

e.g.: `height` & `speed` in wildlife impacts

```
#> # A tibble: 6 × 3
#>   incident_date       height speed
#>   <dttm>               <dbl> <dbl>
#> 1 2018-12-31 00:00:00    700   200
#> 2 2018-12-27 00:00:00    600   145
#> 3 2018-12-23 00:00:00      0   130
#> 4 2018-12-22 00:00:00    500   160
#> 5 2018-12-21 00:00:00    100   150
#> 6 2018-12-18 00:00:00   4500   250
```

Key Questions

Categorical

Does the order matter?

Yes: **Ordinal**

No: **Nominal**

Numerical

Is there a "baseline"?

Yes: **Ratio**

No: **Interval**

Be careful of how variables are encoded!

When numbers are categories

- "Dummy coding": e.g., `passedTest` = `1` or `0`)
- "North", "South", "East", "West" = `1`, `2`, `3`, `4`

When ratio data are discrete (i.e. counts)

- Number of eggs in a carton, heart beats per minute, etc.
- Continuous variables measured discretely (e.g. age)

Time

- As *ordinal* categories: "Jan.", "Feb.", "Mar.", etc.
- As *interval* scale: "Jan. 1", "Jan. 2", "Jan. 3", etc.
- As *ratio* scale: "30 sec", "60 sec", "70 sec", etc.

Week 4: *Exploring Data*

1. Exploring Data
2. Data Types
3. Centrality & Variability
4. Visualizing Centrality & Variability

BREAK

5. Correlation
6. Visualizing Correlation
7. Visualizing Relationships

Summary Measures:

Single variables: Centrality & Variability

Two variables: Correlation

Centrality (a.k.a. The "Average" Value)

A single number representing the *middle* of a set of numbers

Mean: $\frac{\text{Sum of values}}{\text{\# of values}}$

Median: "Middle" value (50% of data above & below)

Mean isn't always the "best" choice

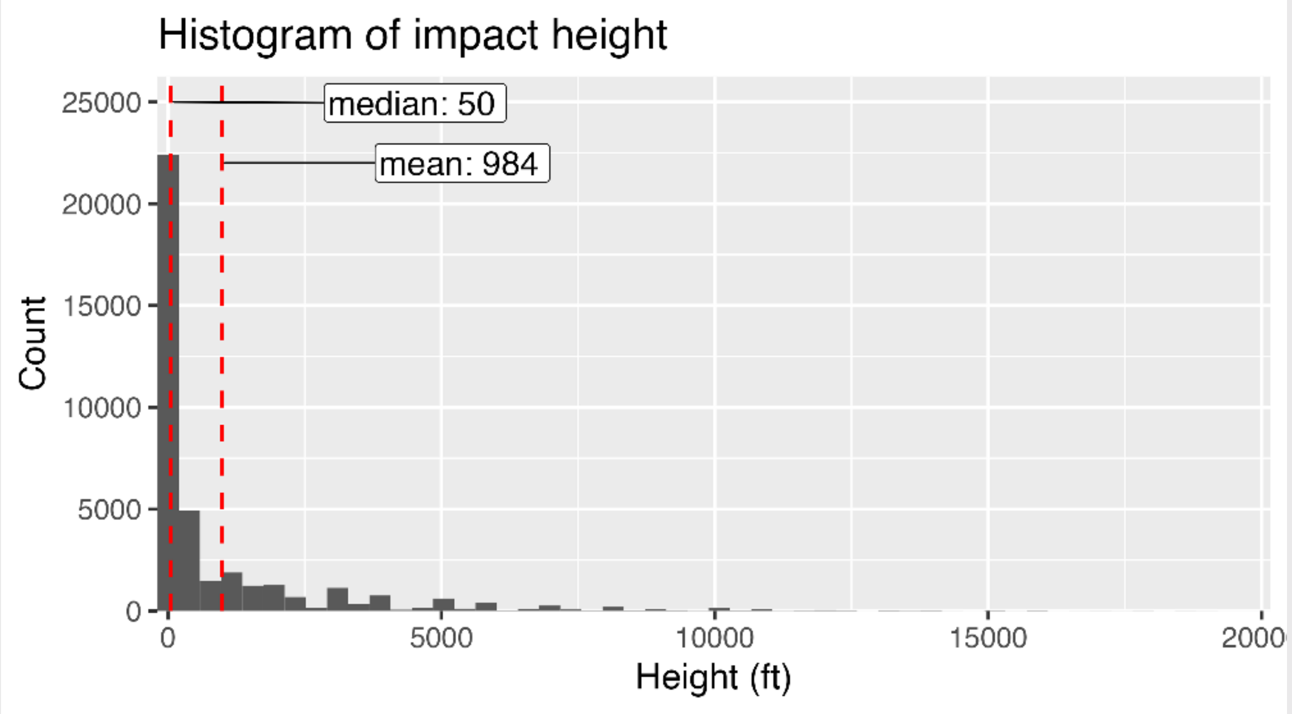
```r
wildlife_impacts %>%
    filter(! is.na(height)) %>%
    summarise(
      mean = mean(height),
      median = median(height)
    )
```

```
#> # A tibble: 1 × 2
#>    mean median
#>   <dbl>  <dbl>
#> 1  984.     50
```

Percent of data below mean:

```
#> [1] "73.9%"
```

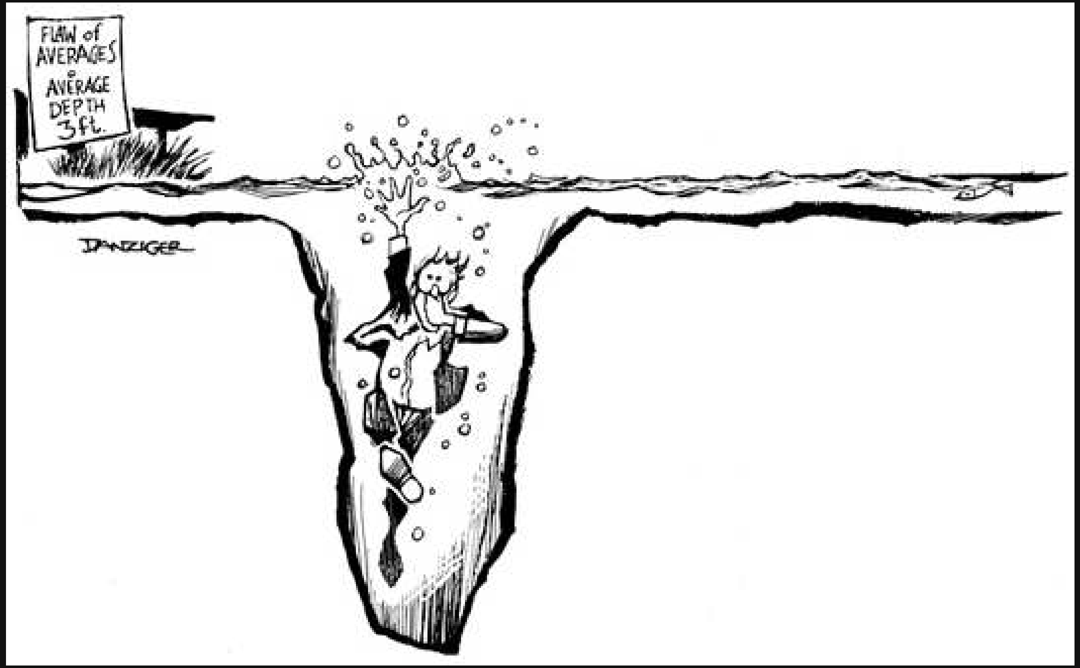
On average, at what height do planes hit birds?



Beware the "flaw of averages"

What happened to the statistician that crossed a river with an average depth of 3 feet?

...he drowned



Variability ("Spread")

Standard deviation: distribution of values relative to the mean

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}}$$

Interquartile range (IQR): $Q_3 - Q_1$ (middle 50% of data)

Range: max - min

Example: Days to ship

Complaints are coming in about orders shipped from warehouse B, so you collect some data:

```
daysToShip
```

```
#>    order warehouseA warehouseB
#> 1      1          3          1
#> 2      2          3          1
#> 3      3          3          1
#> 4      4          4          3
#> 5      5          4          3
#> 6      6          4          4
#> 7      7          5          5
#> 8      8          5          5
#> 9      9          5          5
#> 10    10          5          6
#> 11    11          5          7
#> 12    12          5         10
```

Here, **averages** are misleading:

```
daysToShip %>%
    gather(warehouse, days, warehouseA:warehouseB) %>
    group_by(warehouse) %>%
    summarise(
        mean   = mean(days),
        median = median(days))
```

```
#> # A tibble: 2 × 3
#>   warehouse   mean median
#>   <chr>      <dbl>  <dbl>
#> 1 warehouseA  4.25    4.5
#> 2 warehouseB  4.25    4.5
```

Example: Days to ship

Complaints are coming in about orders shipped from warehouse B, so you collect some data:

```r
daysToShip
```

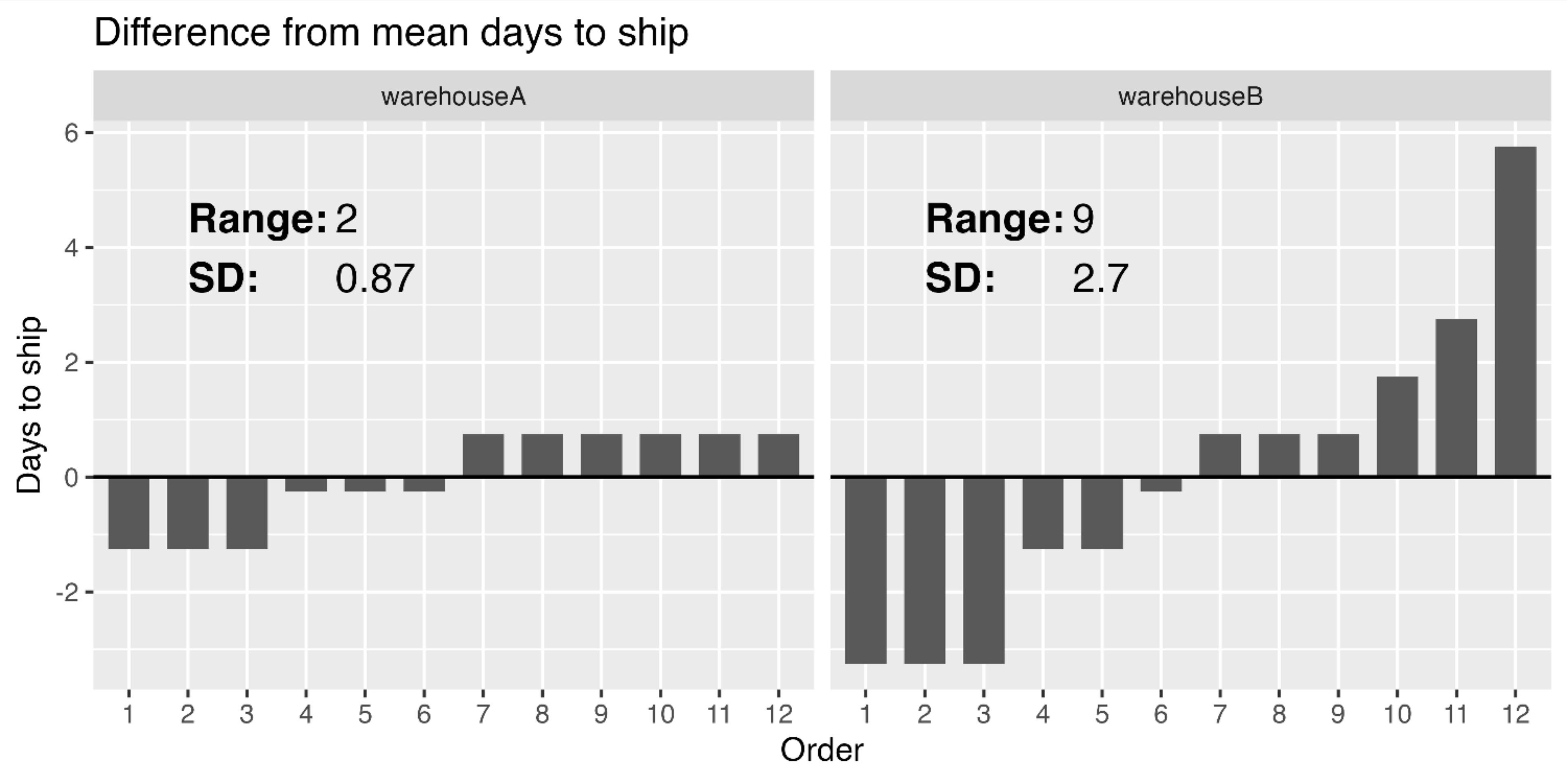
```
#>    order warehouseA warehouseB
#> 1      1          3          1
#> 2      2          3          1
#> 3      3          3          1
#> 4      4          4          3
#> 5      5          4          3
#> 6      6          4          4
#> 7      7          5          5
#> 8      8          5          5
#> 9      9          5          5
#> 10    10          5          6
#> 11    11          5          7
#> 12    12          5         10
```

Variability reveals difference in days to ship:

```r
daysToShip %>%
    gather(warehouse, days, warehouseA:warehouseB) %>
    group_by(warehouse) %>%
    summarise(
        mean   = mean(days),
        median = median(days),
        range = max(days) - min(days),
        sd    = sd(days))
```

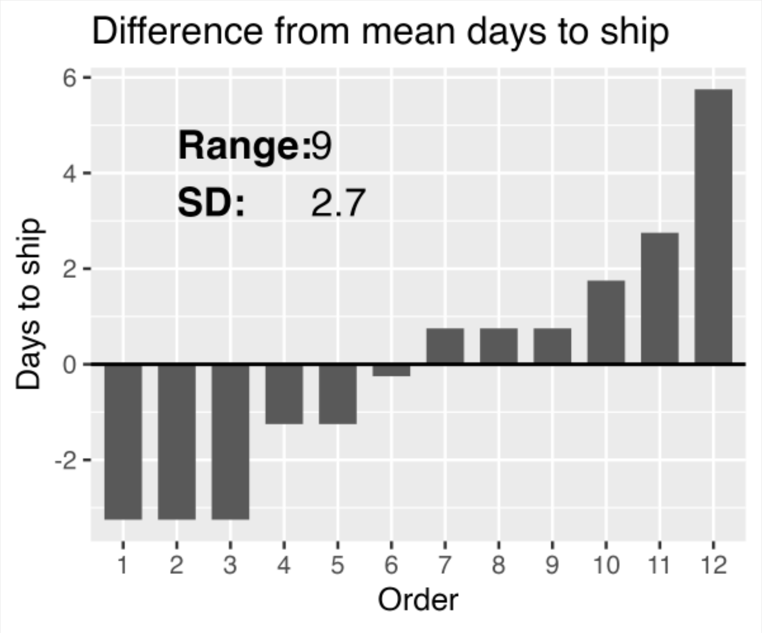
```
#> # A tibble: 2 × 5
#>   warehouse   mean median range    sd
#>   <chr>      <dbl>  <dbl> <dbl> <dbl>
#> 1 warehouseA  4.25    4.5     2 0.866
#> 2 warehouseB  4.25    4.5     9 2.70
```

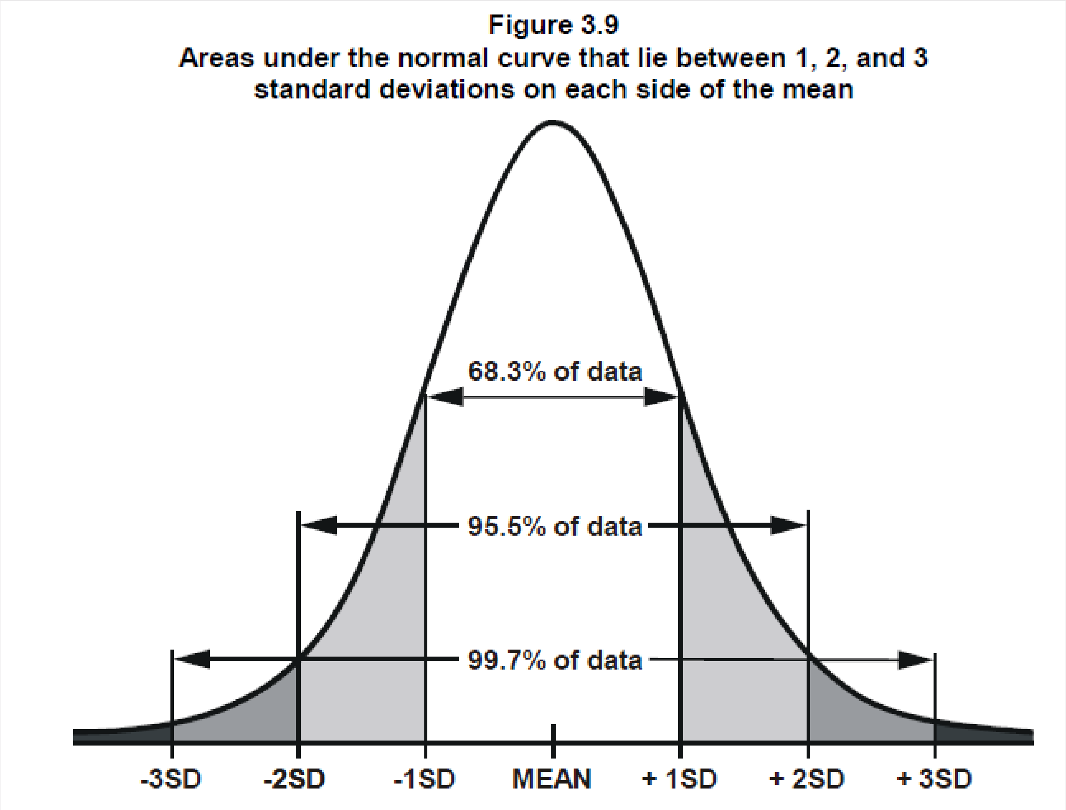
Example: Days to ship



Interpreting the standard deviation

$$s = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}}$$





Outliers

Mean & **Standard Deviation** are sensitive to outliers

Outliers: $Q_1 - 1.5IQR$ or $Q_3 + 1.5IQR$

Extreme values: $Q_1 - 3IQR$ or $Q_3 + 3IQR$

```r
data1 <- c(3,3,4,5,5,6,6,7,8,9)
```

- Mean: 5.6
- Standard Deviation: 2.01
- Median: 5.5
- IQR: 2.5

```r
data2 <- data1
data2[10] <- 20
```

- Mean: 6.7
- Standard Deviation: 4.95
- Median: 5.5
- IQR: 2.5

Robust statistics for continuous data

(less sensitive to outliers)

Centrality: Use *median* rather than *mean*

Variability: Use *IQR* rather than *standard deviation*

Practice with summary measurements

10:00

1) Read in the following data sets:

- `milk_production.csv`
- `lotr_words.csv`

2) For each variable in each data set, if possible, summarize its

1. Centrality

2. Variability

Week 4: *Exploring Data*

1. Exploring Data
2. Data Types
3. Centrality & Variability
4. Visualizing Centrality & Variability

BREAK

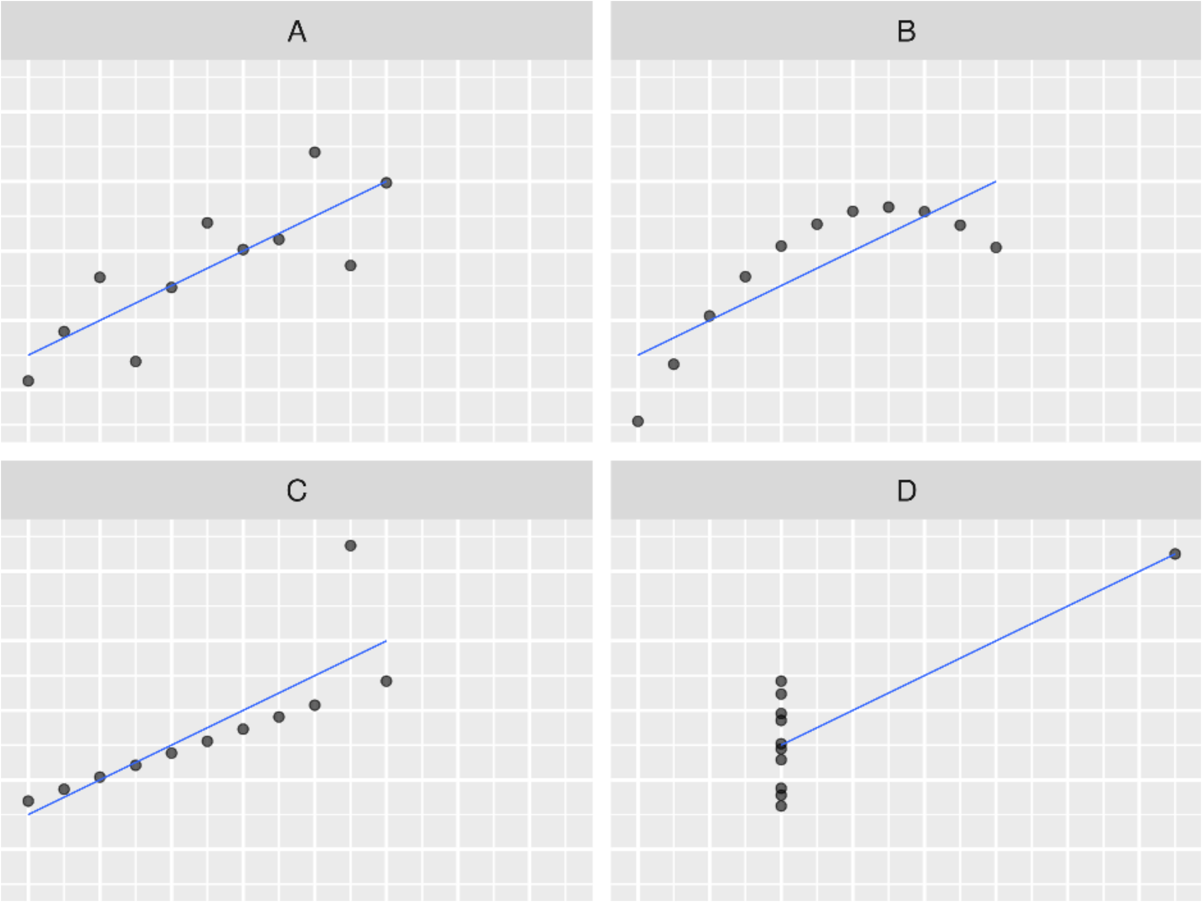
5. Correlation
6. Visualizing Correlation
7. Visualizing Relationships

"Visualizing data helps us think"

	A		B		C		D	
	x	y	x	y	x	y	x	y
	10	8.04	10	9.14	10	7.46	8	6.58
	8	6.95	8	8.14	8	6.77	8	5.76
	13	7.58	13	8.74	13	12.74	8	7.71
	9	8.81	9	8.77	9	7.11	8	8.84
	11	8.33	11	9.26	11	7.81	8	8.47
	14	9.96	14	8.1	14	8.84	8	7.04
	6	7.24	6	6.13	6	6.08	8	5.25
	4	4.26	4	3.1	4	5.39	19	12.5
	12	10.84	12	9.13	12	8.15	8	5.56
	7	4.82	7	7.26	7	6.42	8	7.91
	5	5.68	5	4.74	5	5.73	8	6.89
Sum:	99	82.51	99	82.51	99	82.5	99	82.51
Mean:	9	7.5	9	7.5	9	7.5	9	7.5
St. Dev:	3.3	2	3.3	2	3.3	2	3.3	2

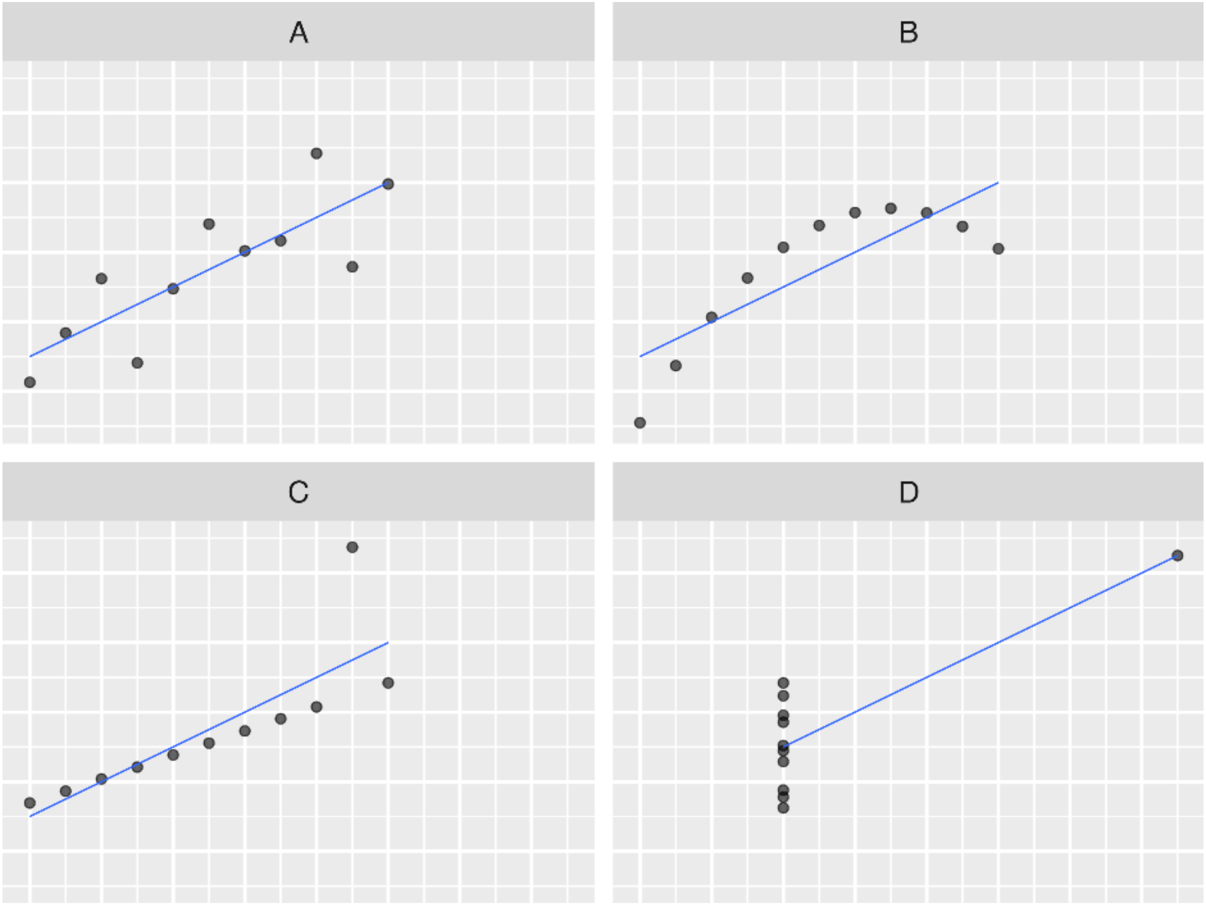
Stephen Few (2009, pg. 6)

Anscombe's Quartet



Stephen Few (2009, pg. 6)

Anscombe's Quartet



EDA

EMSE GWU

The data *type* determines how to summarize it

Nominal (Categorical)

Measures:

- Frequency counts / Proportions

Charts:

- Bars

Ordinal (Categorical)

Measures:

- Frequency counts / Proportions
- Centrality: Median, Mode
- Variability: IQR

Charts:

- Bars

Numerical (Continuous)

Measures:

- Centrality: Mean, median
- Variability: Range, standard deviation, IQR

Charts:

- Histogram
- Boxplot

Summarizing **Nominal** data

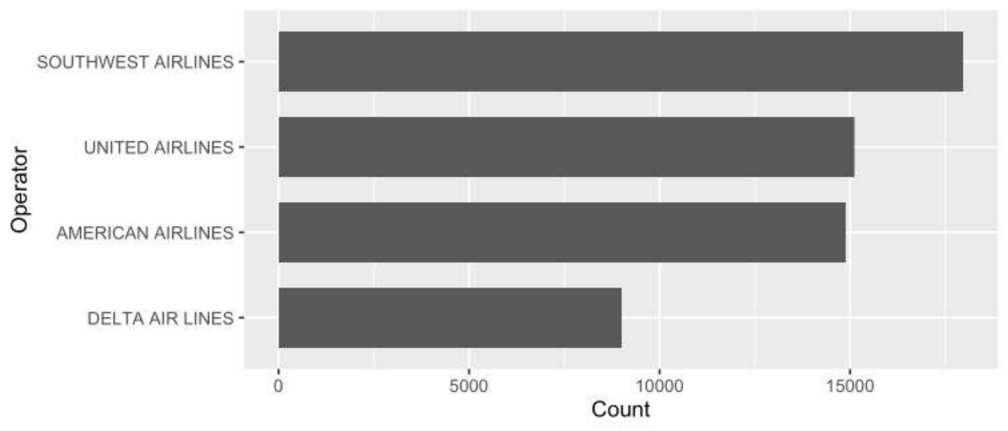
Summarize with counts / percentages

```
wildlife_impacts %>%
    count(operator, sort = TRUE) %>%
    mutate(p = n / sum(n))
```

```
#> # A tibble: 4 × 3
#>   operator               n     p
#>   <chr>              <int> <dbl>
#> 1 SOUTHWEST AIRLINES 17970 0.315
#> 2 UNITED AIRLINES    15116 0.265
#> 3 AMERICAN AIRLINES  14887 0.261
#> 4 DELTA AIR LINES     9005 0.158
```

Visualize with (usually sorted) bars

```
wildlife_impacts %>%
    count(operator, sort = TRUE) %>%
    ggplot() +
    geom_col(aes(x = n, y = reorder(operator, n)),
             width = 0.7) +
    labs(x = "Count", y = "Operator")
```



Summarizing **Ordinal** data

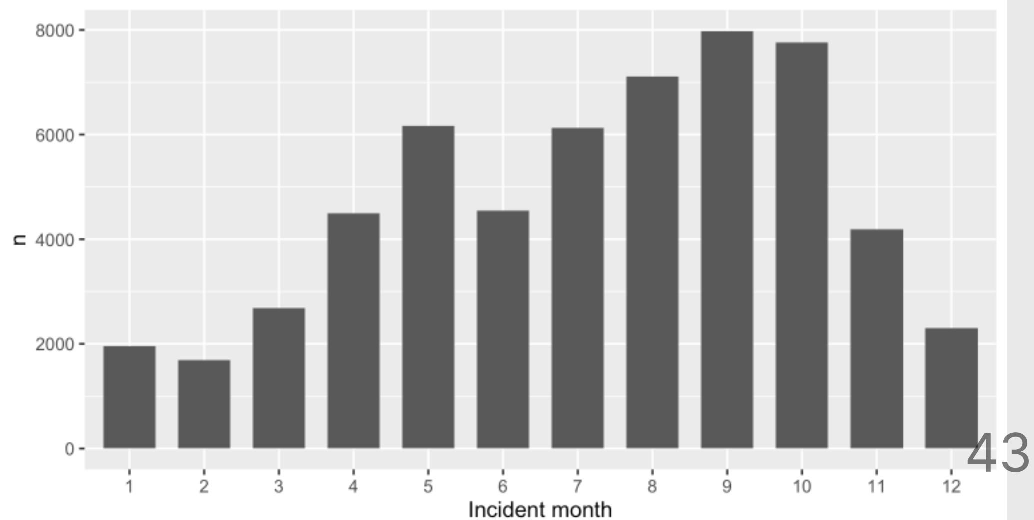
Summarize: Counts / percentages

```
wildlife_impacts %>%
    count(incident_month, sort = TRUE) %>%
    mutate(p = n / sum(n))
```

```
#> # A tibble: 12 × 3
#>    incident_month     n      p
#>             <dbl> <int>  <dbl>
#>  1              9  7980 0.140
#>  2             10  7754 0.136
#>  3              8  7104 0.125
#>  4              5  6161 0.108
#>  5              7  6133 0.108
#>  6              6  4541 0.0797
#>  7              4  4490 0.0788
#>  8             11  4191 0.0736
#>  9              3  2678 0.0470
#> 10             12  2303 0.0404
#> 11              1  1951 0.0342
#> 12              2  1692 0.0297
```

Visualize: Bars

```
wildlife_impacts %>%
    count(incident_month, sort = TRUE) %>%
    ggplot() +
    geom_col(aes(x = as.factor(incident_month),
                 y = n), width = 0.7) +
    labs(x = "Incident month")
```



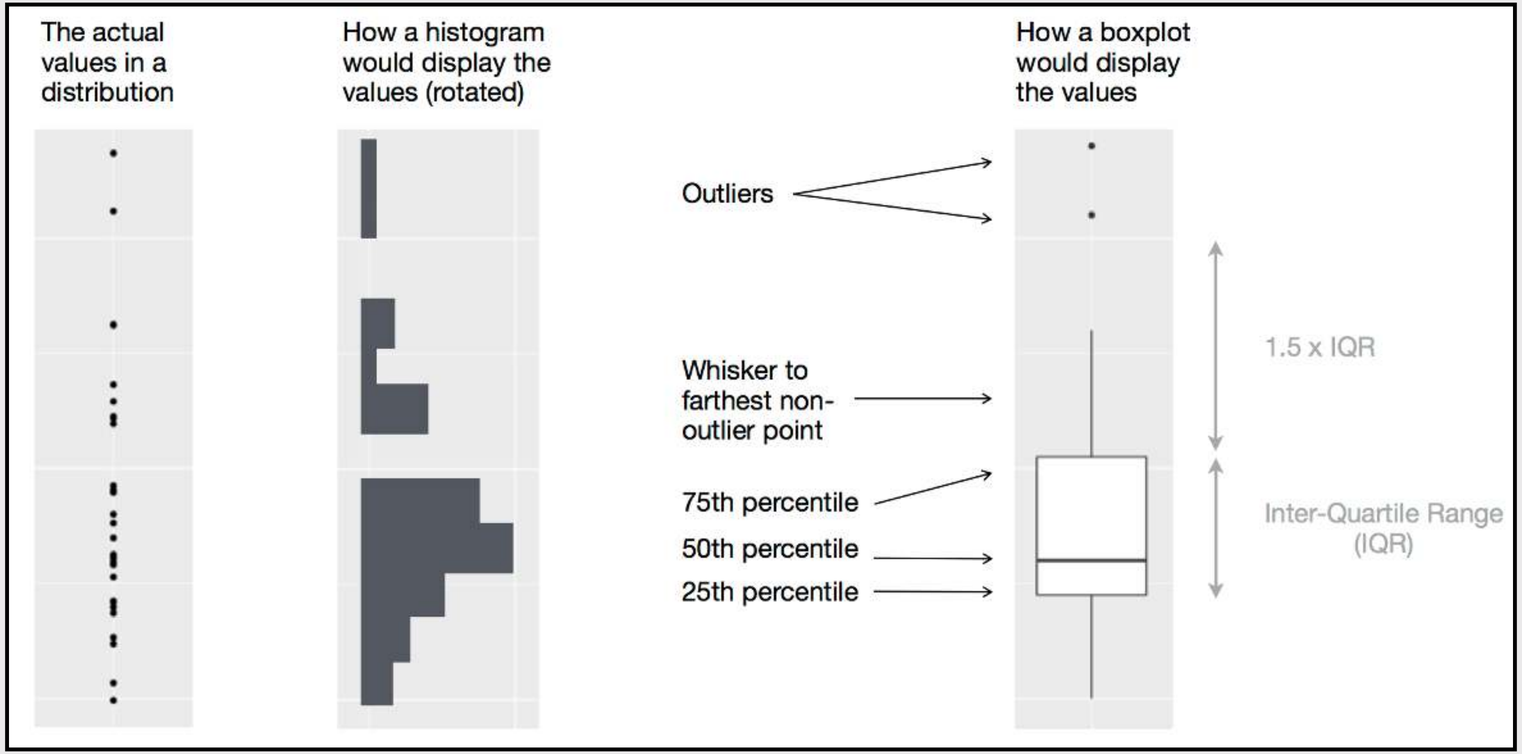
Summarizing **continuous** variables

Histograms:

- Skewness
- Number of modes

Boxplots:

- Outliers
- Comparing variables



Histogram: Identify Skewness & # of Modes

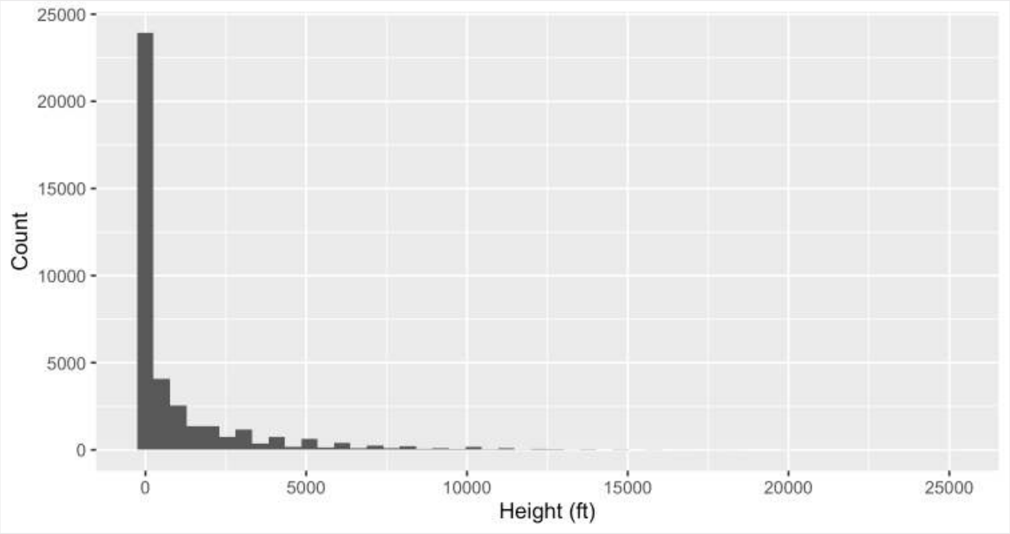
Summarise:
Mean, median, sd, range, & IQR:

```
summary(wildlife_impacts$height)
```

```
#>     Min. 1st Qu.  Median    Mean
#>      0.0     0.0    50.0   983.8
```

Visualize:
Histogram (identify skewness & modes)

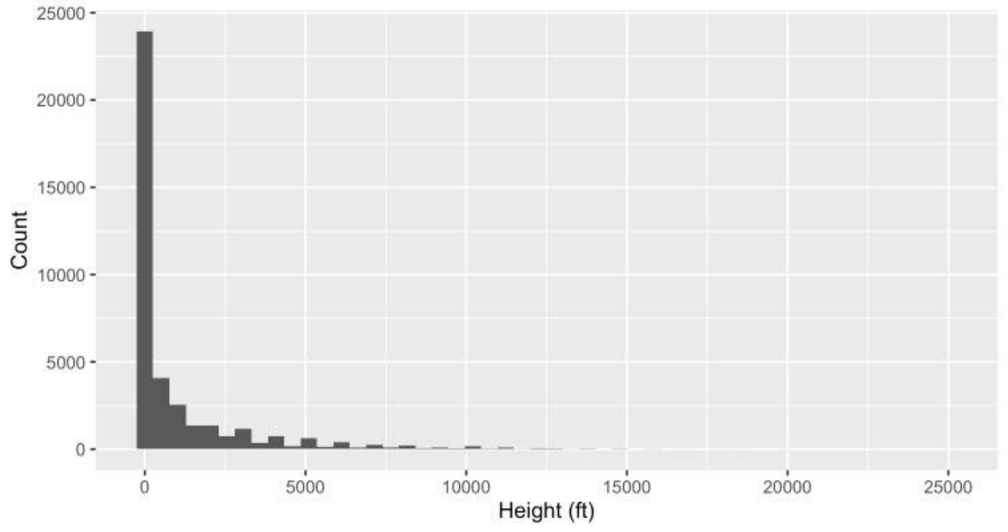
```
ggplot(wildlife_impacts) +
  geom_histogram(aes(x = height), bins = 50) +
  labs(x = 'Height (ft)', y = 'Count')
```



Histogram: Identify Skewness & # of Modes

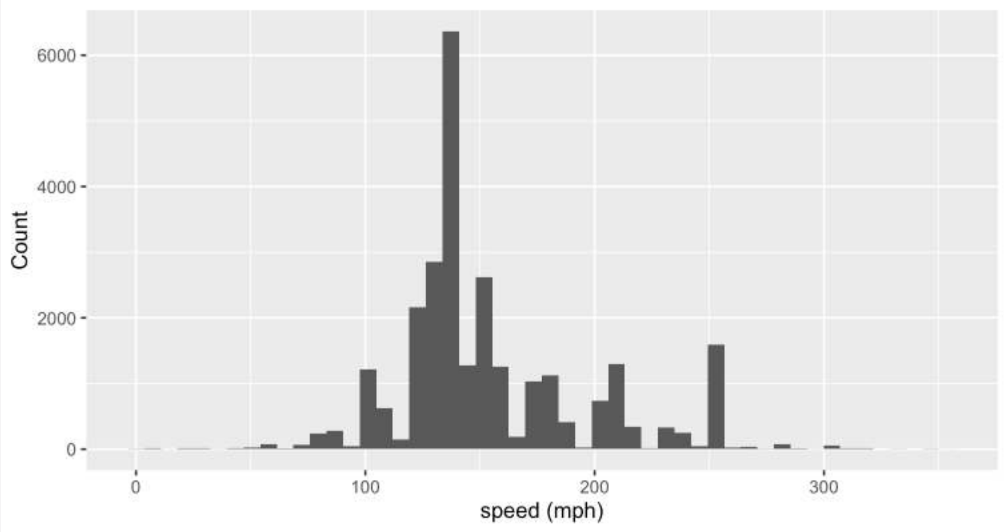
Height

```
ggplot(wildlife_impacts) +
  geom_histogram(aes(x = height), bins = 50)
  labs(x = 'Height (ft)', y = 'Count')
```



Speed

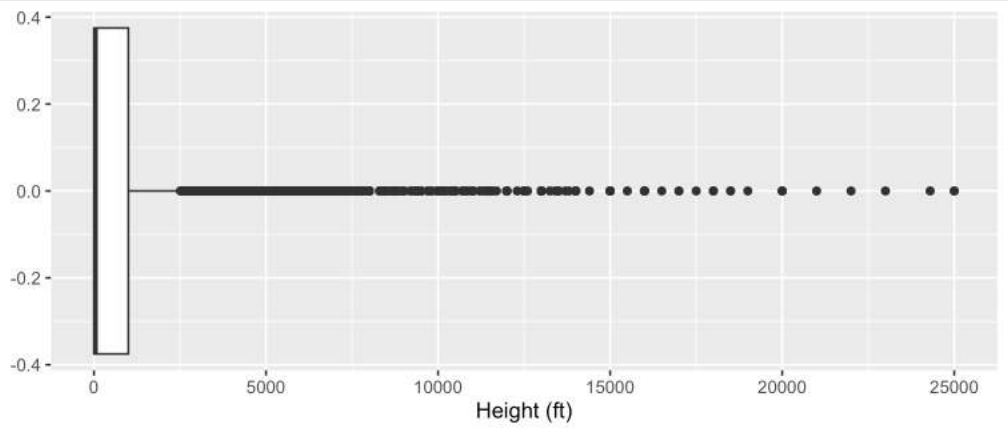
```
ggplot(wildlife_impacts) +
  geom_histogram(aes(x = speed), bins = 50)
  labs(x = 'speed (mph)', y = 'Count')
```



Boxplot: Identify outliers

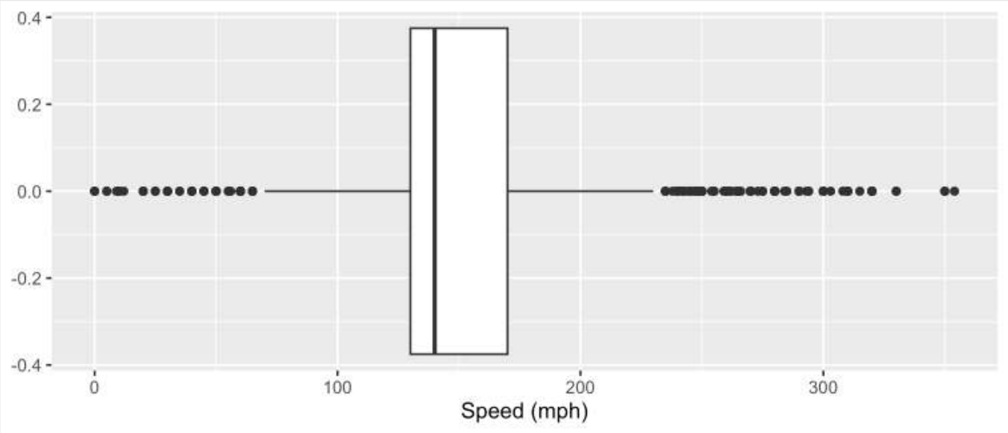
Height

```
ggplot(wildlife_impacts) +
    geom_boxplot(aes(x = height)) +
    labs(x = 'Height (ft)', y = NULL)
```



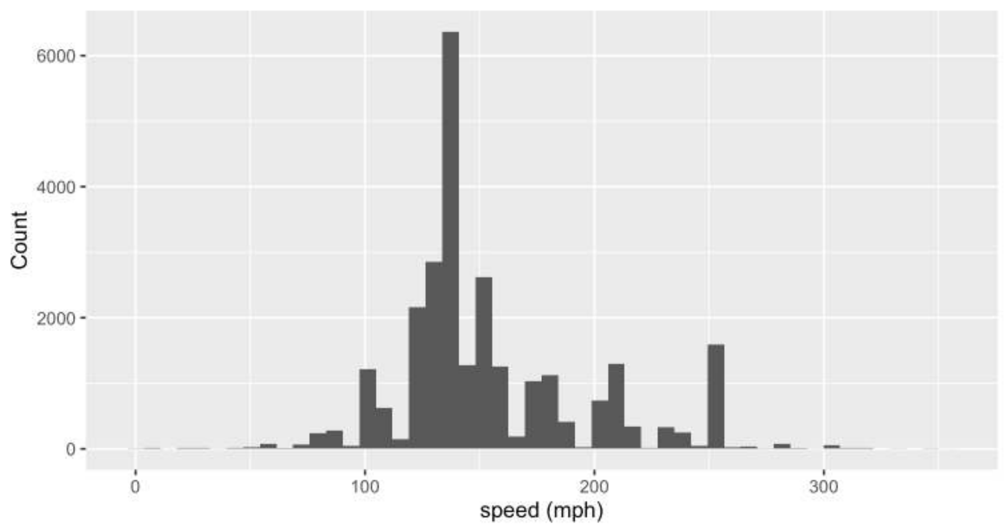
Speed

```
ggplot(wildlife_impacts) +
    geom_boxplot(aes(x = speed)) +
    labs(x = 'Speed (mph)', y = NULL)
```



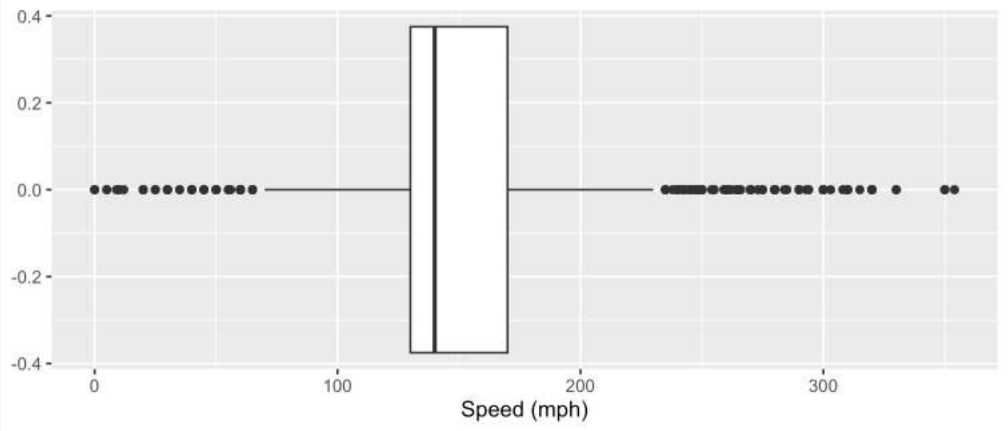
Histogram

- Skewness
- Modes



Boxplot

- Outliers



Practicing visual summaries

15:00

1) Read in the following data sets:

- `faithful.csv`
- `marathon.csv`

2) Summarize the following variables using an appropriate chart (bar chart, histogram, and / or boxplot):

- faithful: `eruptions`
- faithful: `waiting`
- marathon: `Age`
- marathon: `State`
- marathon: `Country`
- marathon: ``` `Official Time` ```

Break!

Stand up, Move around, Stretch!

05:00

Week 4: *Exploring Data*

1. Exploring Data
2. Data Types
3. Centrality & Variability
4. Visualizing Centrality & Variability

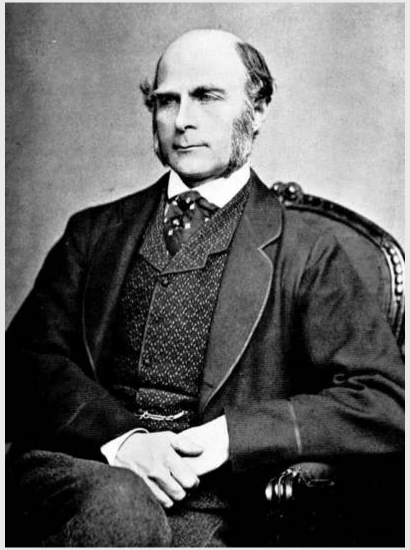
BREAK

5. Correlation
6. Visualizing Correlation
7. Visualizing Relationships

Some pretty racist origins in eugenics ("well born")

Sir Francis Galton (1822 - 1911)

- Charles Darwin's cousin.
- "Father" of eugenics.
- Interested in heredity.



Karl Pearson (1857 - 1936)

- Galton's (hero-worshiping) protégé.
- Defined correlation equation.
- "Father" of mathematical statistics.

Galton's family data

Galton, F. (1886). "Regression towards mediocrity in hereditary stature". *The Journal of the Anthropological Institute of Great Britain and Ireland* 15: 246-263.

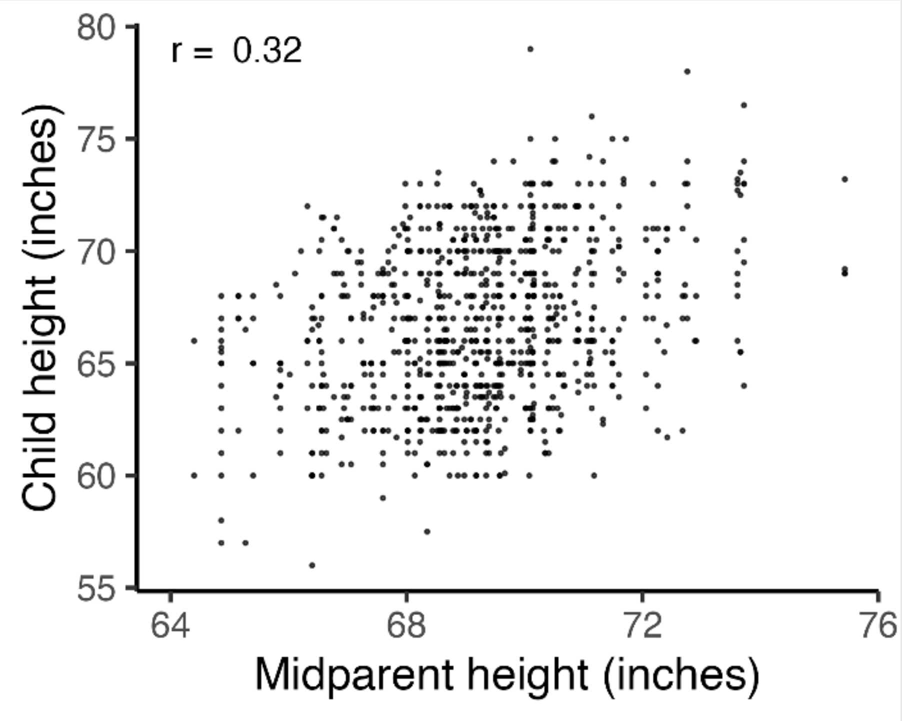
Galton's question: Does marriage selection indicate a relationship between the heights of husbands and wives? (He called this "assortative mating")

"midparent height" is just a scaled mean:

```
midparentHeight =  (father + 1.08*mother)/2
```

```
library(HistData)

galtonScatterplot <- ggplot(GaltonFamilies) +
    geom_point(aes(x = midparentHeight,
                   y = childHeight),
               size = 0.5, alpha = 0.7) +
    theme_classic() +
    labs(x = 'Midparent height (inches)',
         y = 'Child height (inches)')
```



How do you measure correlation?

Pearson came up with this:

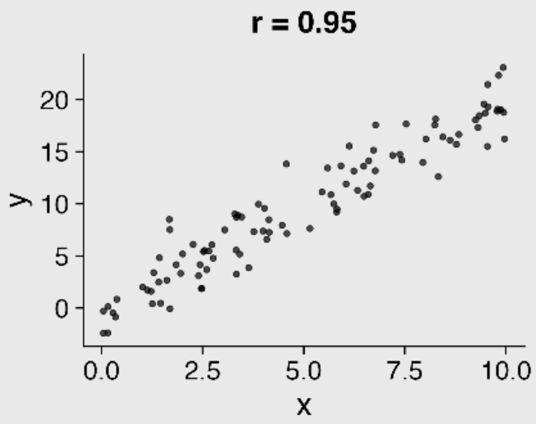
$$r = \frac{\mathrm{Cov}(x,y)}{\mathrm{sd}(x) * \mathrm{sd}(y)}$$

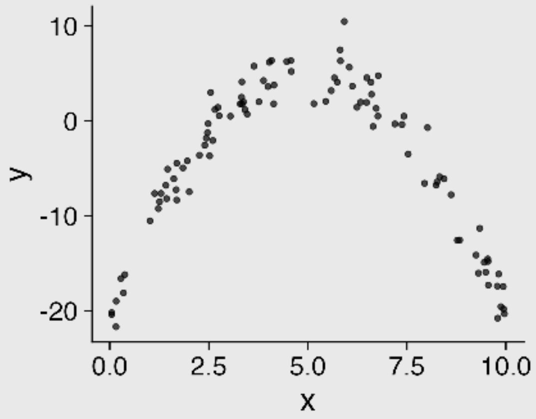
How do you measure correlation?

$$r = \frac{\mathrm{Cov}(x,y)}{\mathrm{sd}(x) * \mathrm{sd}(y)}$$

Assumptions:

1. Variables must be interval or ratio
2. Linear relationship





How do you *interpret* r?

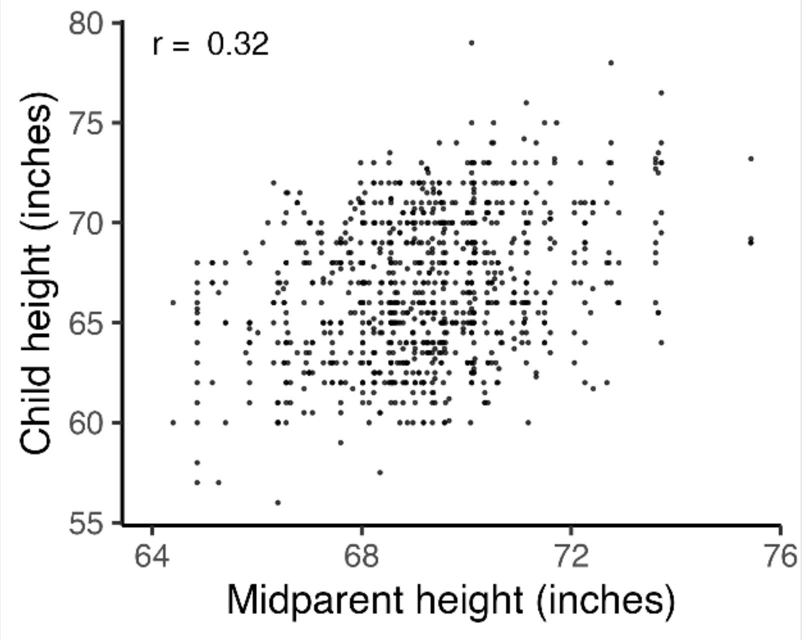
$$r = \frac{\text{Cov}(x,y)}{\text{sd}(x) * \text{sd}(y)}$$

```r
cor(x = GaltonFamilies$midparentHeight,
    y = GaltonFamilies$childHeight,
    method = 'pearson')
```

```
#> [1] 0.3209499
```

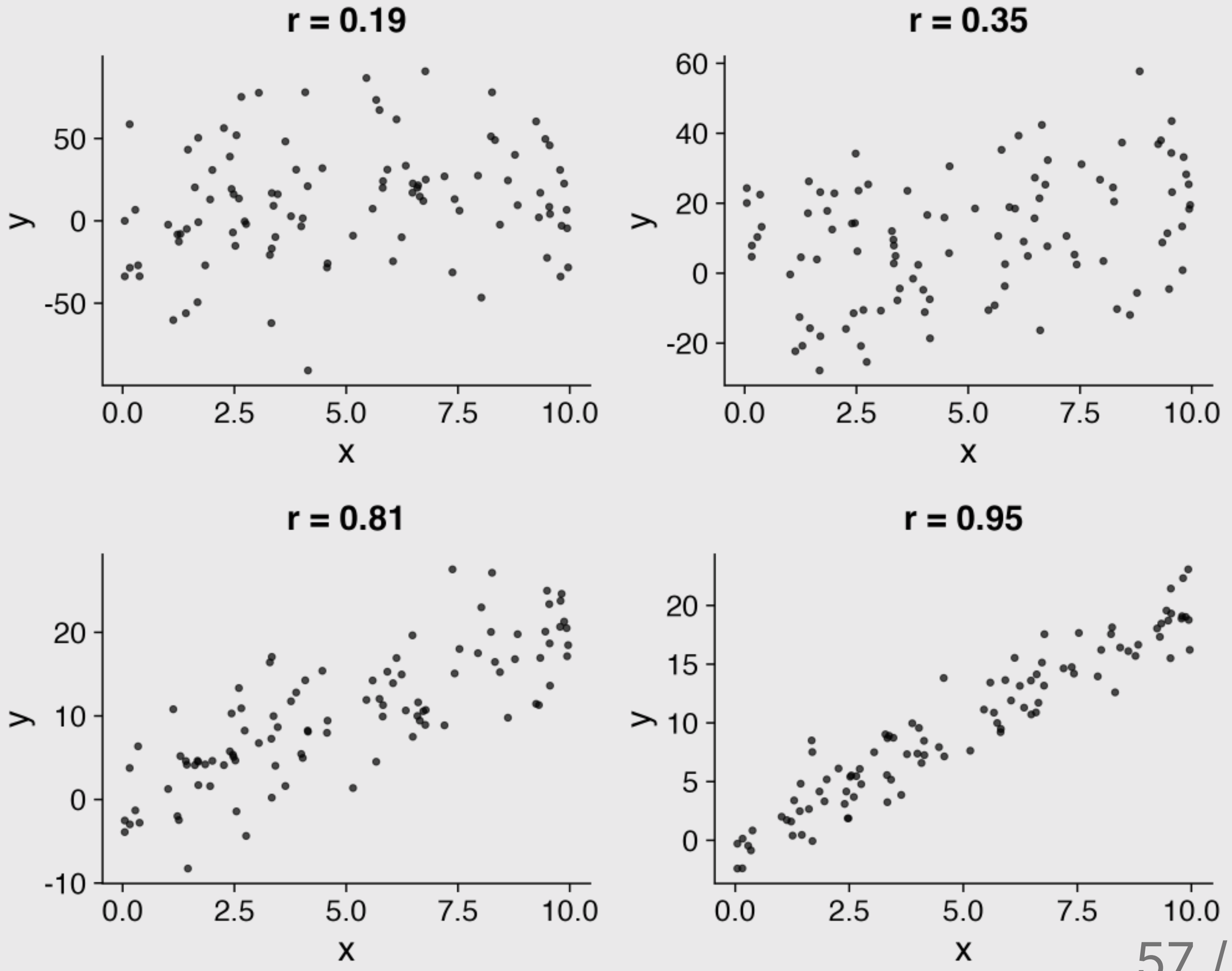
Interpretation:

- $-1 \leq r \leq 1$
- Closer to 1 is stronger correlation
- Closer to 0 is weaker correlation



What does r mean?

- $\pm 0.1 - 0.3$: Weak
- $\pm 0.3 - 0.5$: Moderate
- $\pm 0.5 - 0.8$: Strong
- $\pm 0.8 - 1.0$: Very strong



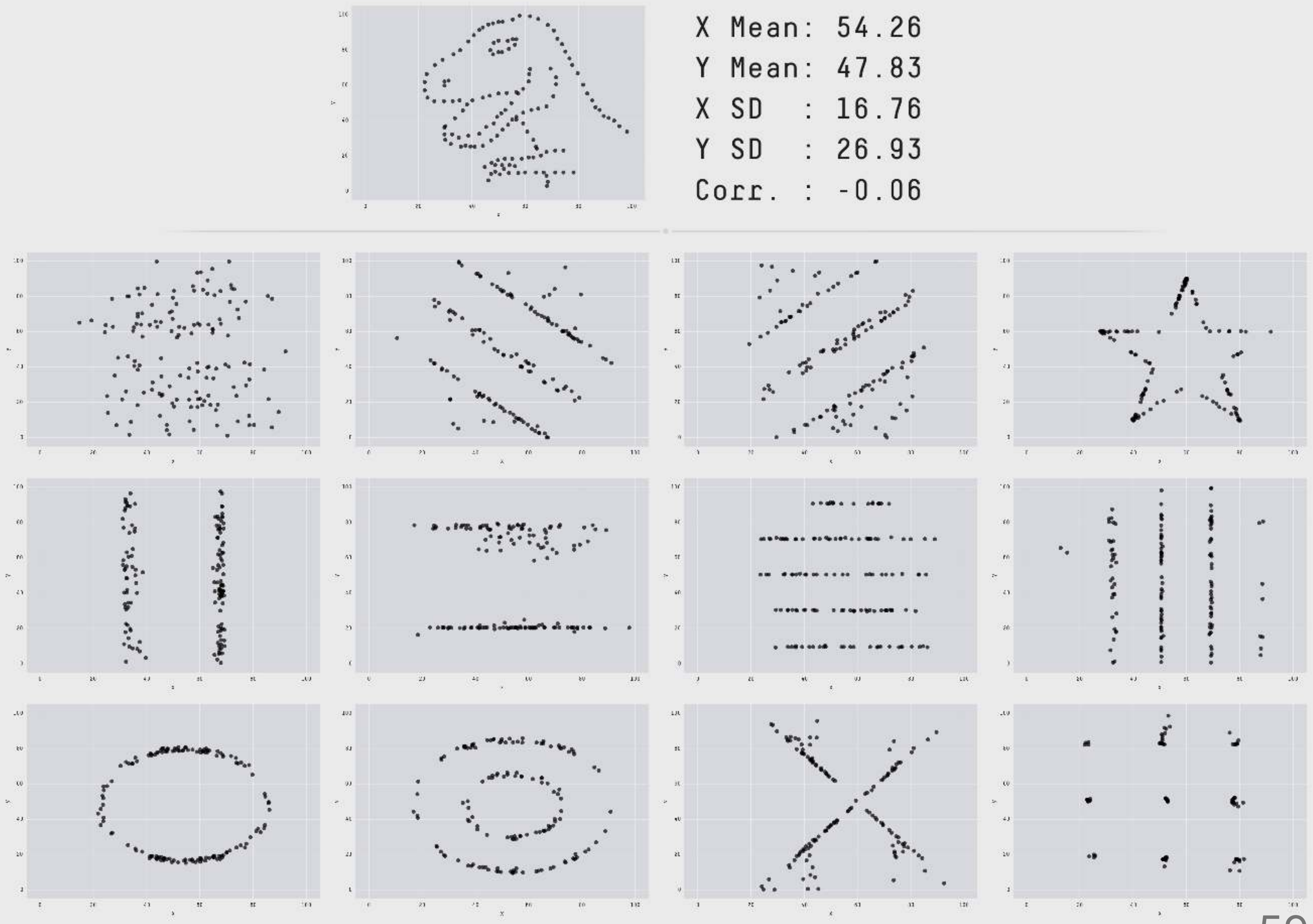
Visualizing correlation is...um...easy, right?

guessthecorrelation.com

Click here to vote!

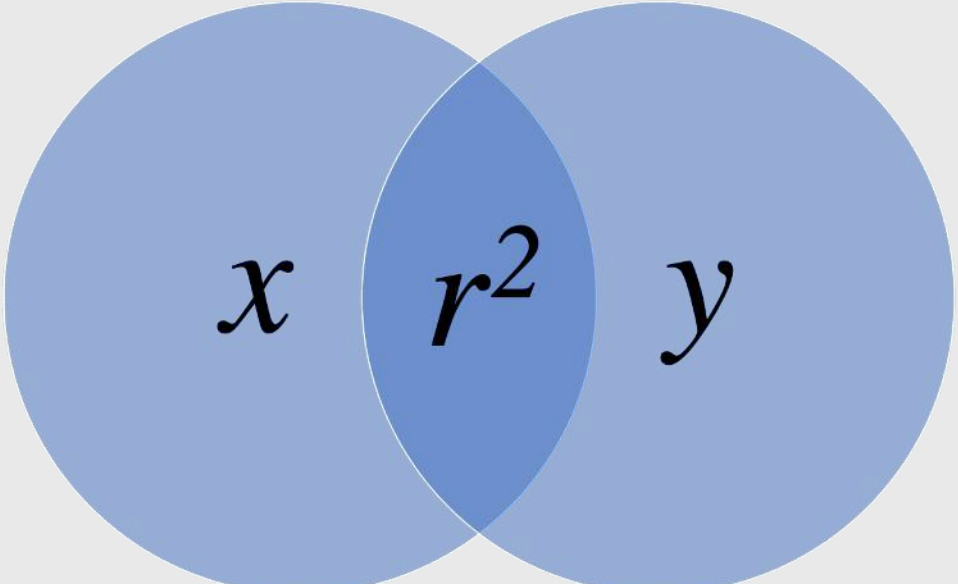
The datasaurus

(More here)



Coefficient of determination: r^2

Percent of variance in one variable that is explained by the other variable



r	r^2
0.1	0.01
0.2	0.04
0.3	0.09
0.4	0.16
0.5	0.25
0.6	0.36
0.7	0.49
0.8	0.64
0.9	0.81
1.0	1.00

You should report both r and r^2

Correlation between parent and child height is 0.32, therefore 10% of the variance in the child height is explained by the parent height.

Correlation != Causation

X causes Y

- Training causes improved performance

Y causes X

- Good (bad) performance causes people to train harder (less hard).

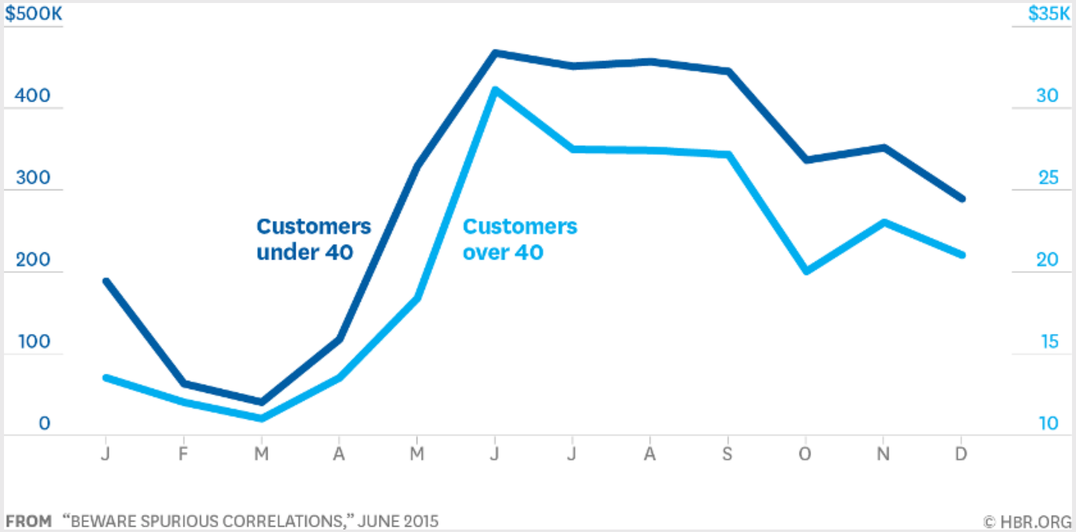
Z causes both X & Y

- Commitment and motivation cause increased training and better performance.

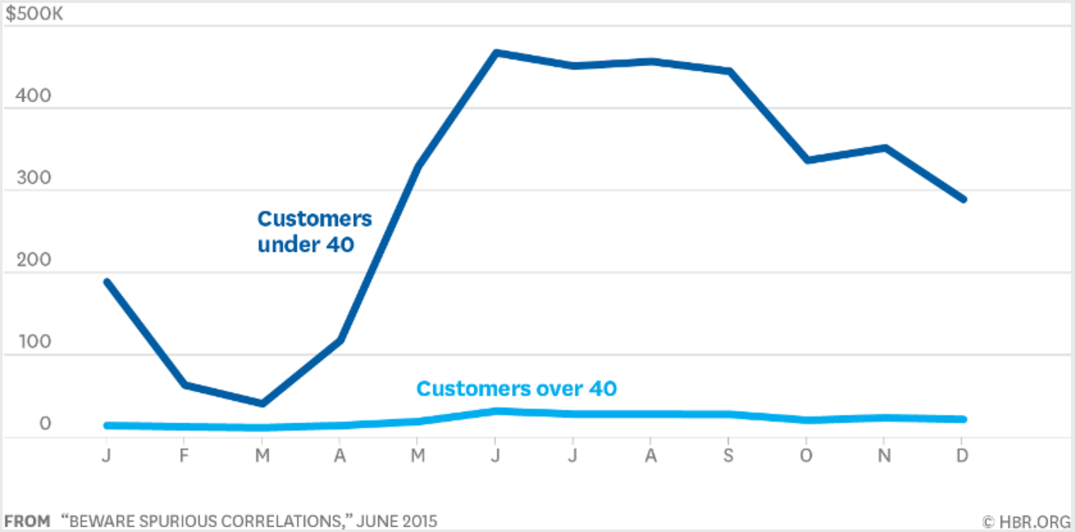
Be weary of dual axes!

(They can cause spurious correlations)

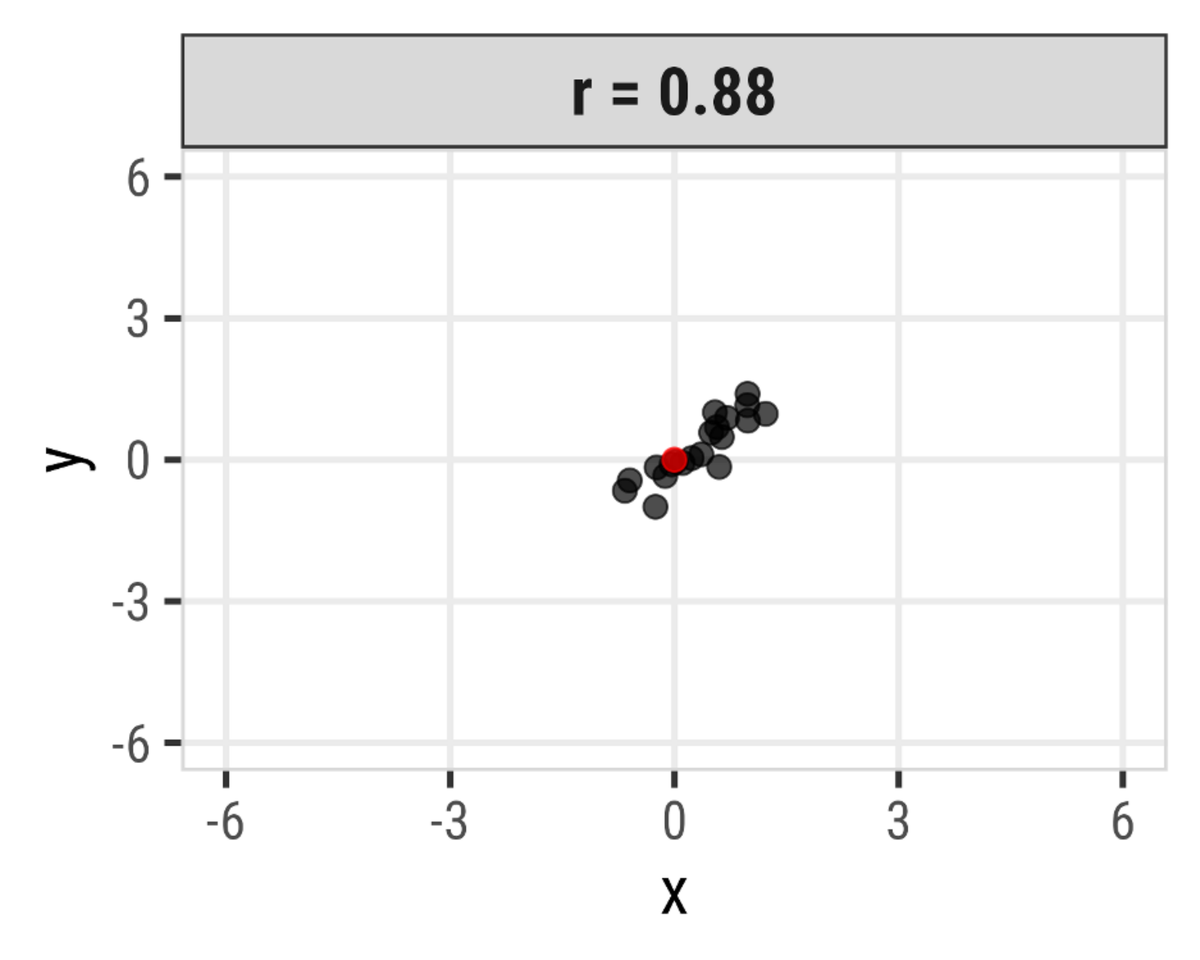
Dual axes

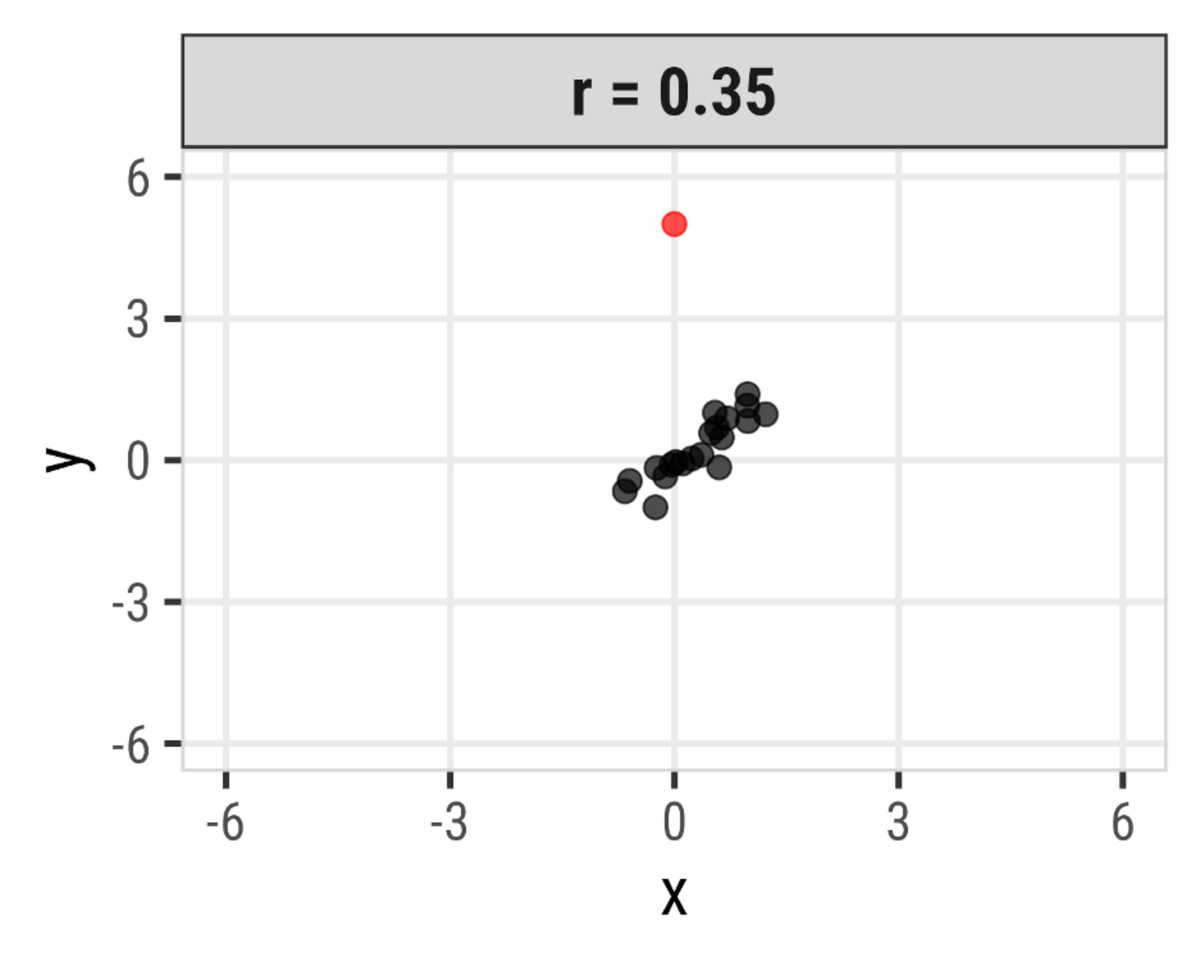


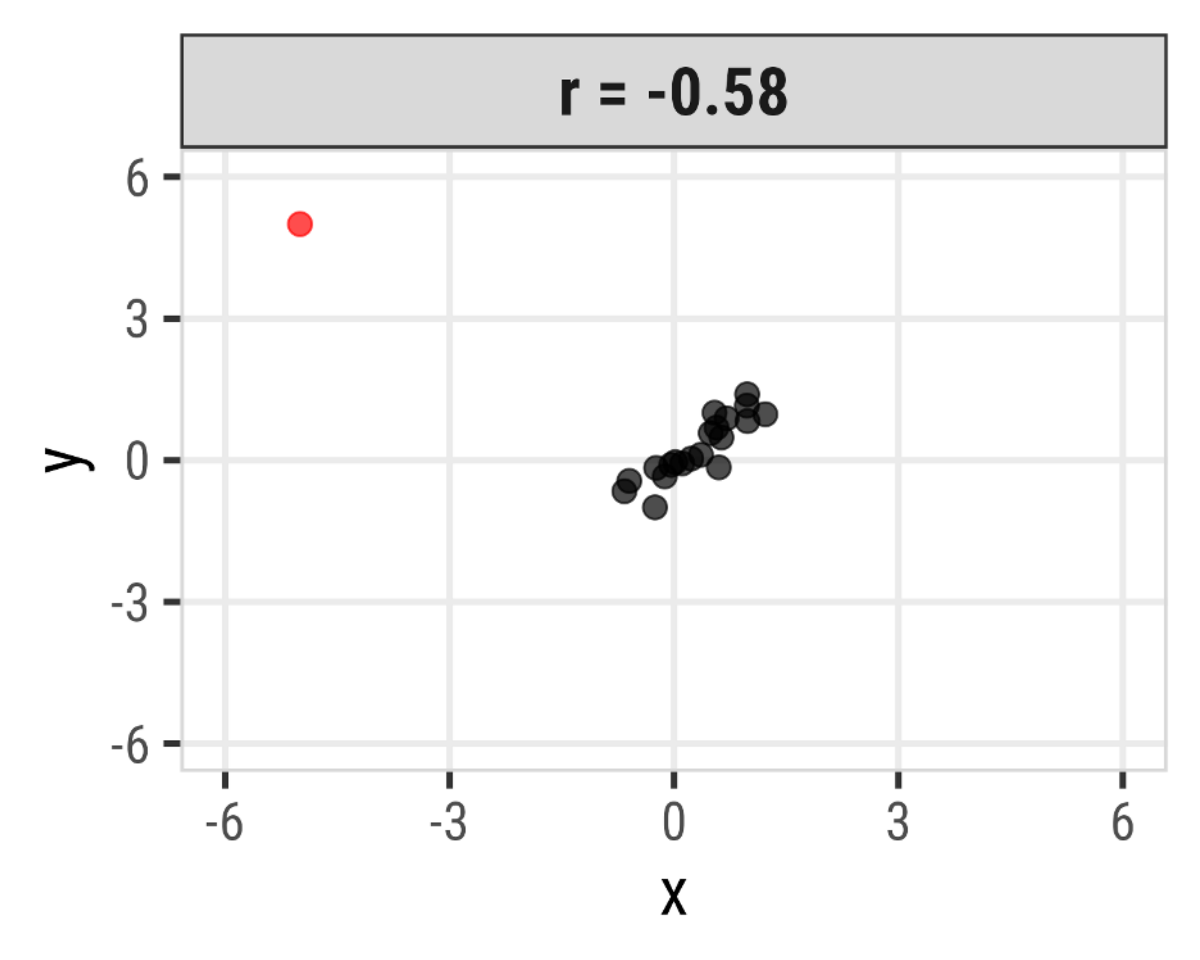
Single axis



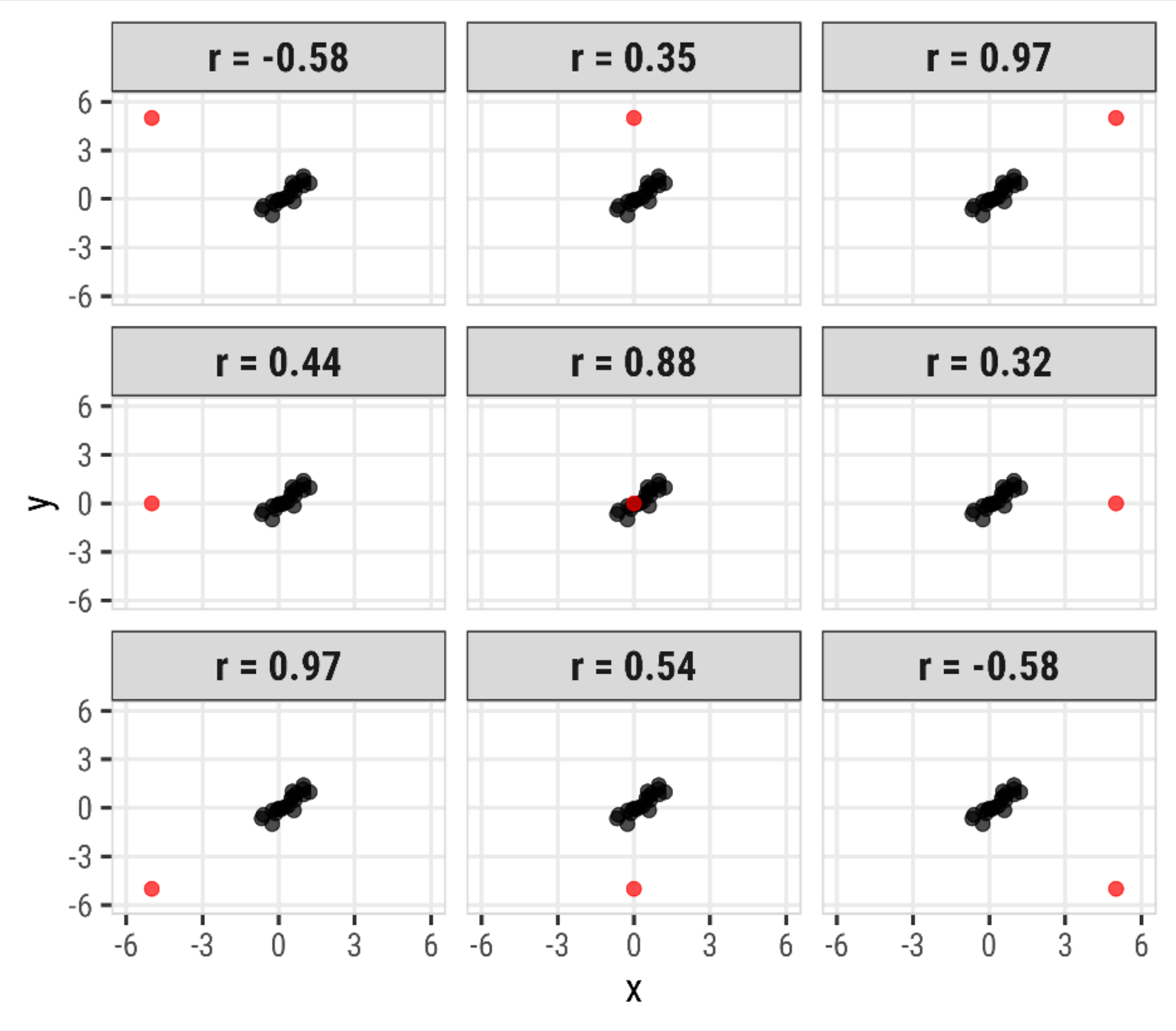
Outliers


r = 0.88
6
3
0
-3
-6
y
-6
-3
0
3
6
x


r = 0.35
6
3
0
-3
-6
y
-6
-3
0
3
6
x


r = -0.58
6
3
0
-3
-6
y
-6
-3
0
3
6
x

Pearson correlation is highly sensitive to outliers



Spearman's rank-order correlation

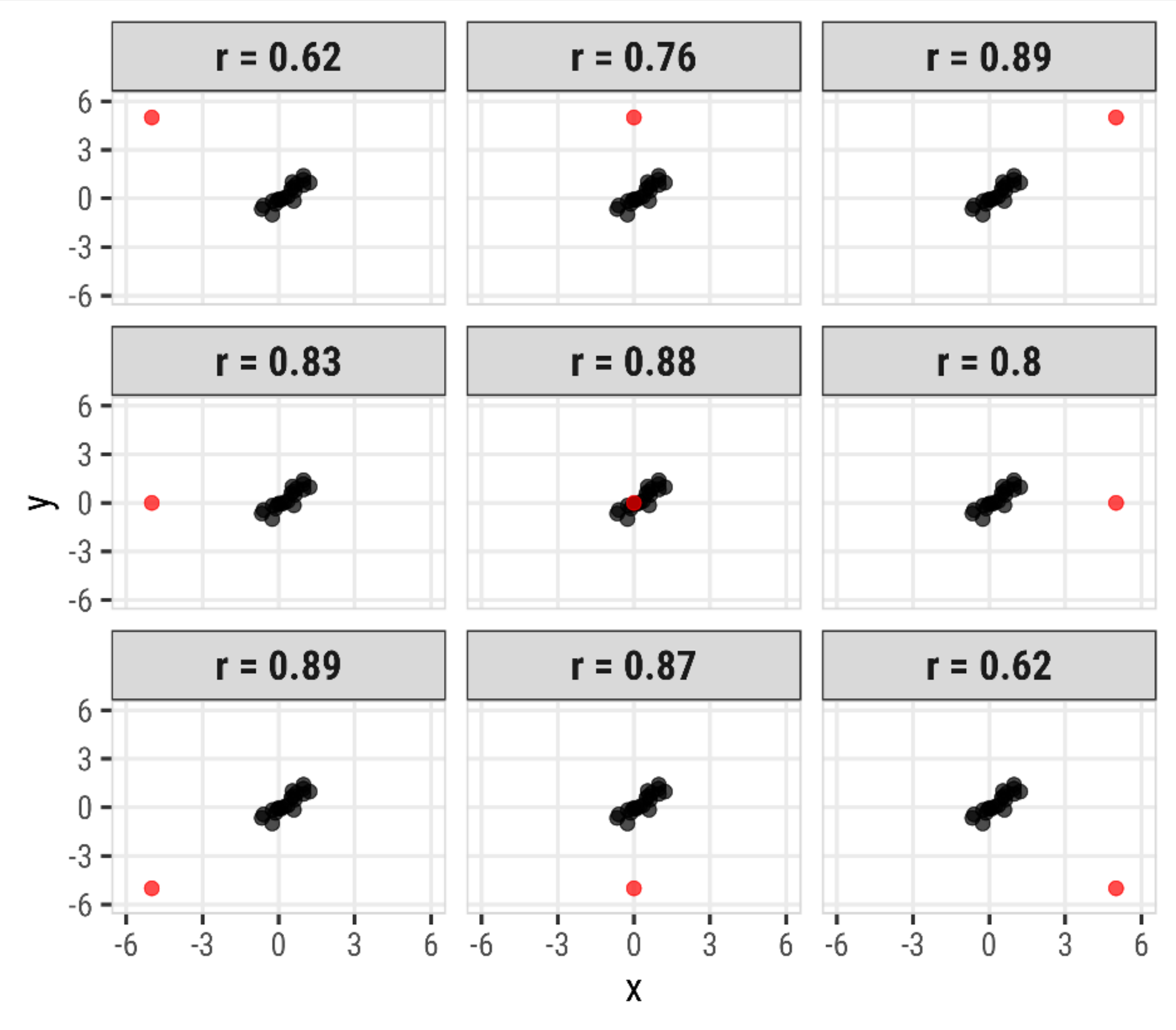
$$r = \frac{\mathrm{Cov}(x,y)}{\mathrm{sd}(x) * \mathrm{sd}(y)}$$

- Separately rank the values of X & Y.
- Use Pearson's correlation on the *ranks* instead of the x & y values.

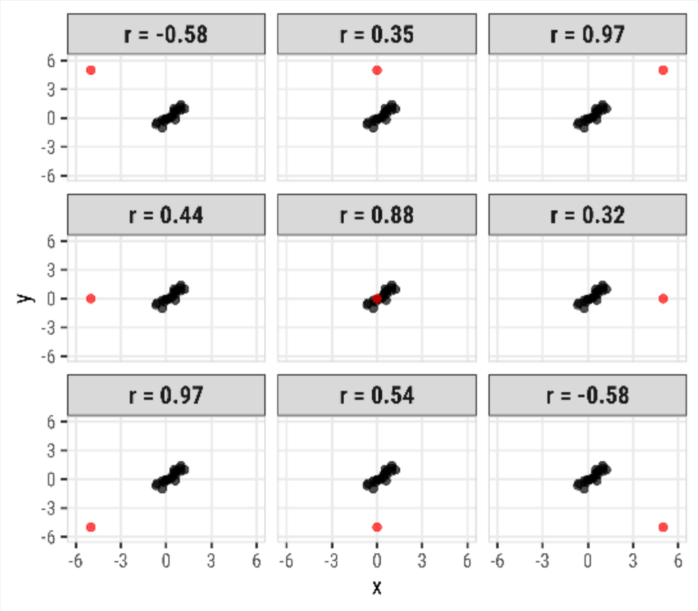
Assumptions:

- Variables can be ordinal, interval or ratio
- Relationship must be monotonic (i.e. does not require linearity)

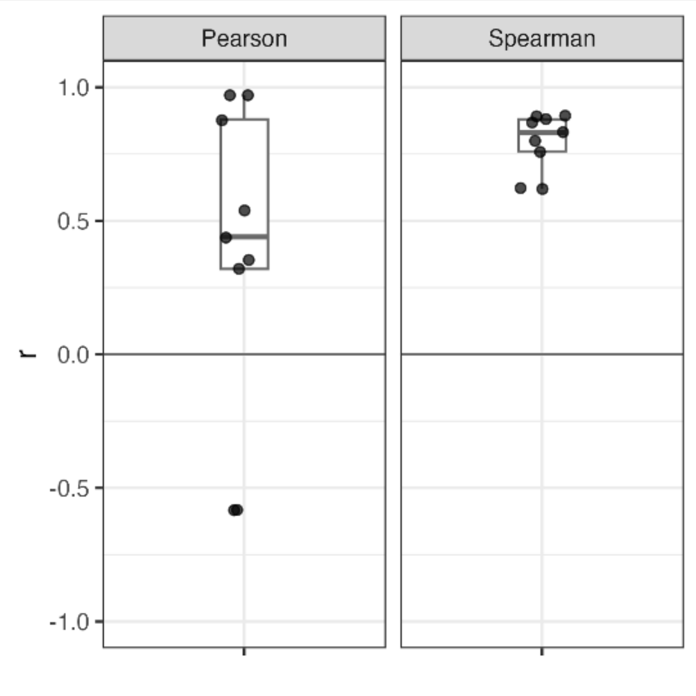
Spearman correlation more robust to outliers



Spearman correlation more robust to outliers



Pearson	Spearman
-0.56	0.53
0.39	0.69
0.94	0.81
0.38	0.76
0.81	0.79
0.31	0.70
0.95	0.81
0.51	0.75
-0.56	0.53



Summary of correlation

- **Pearson's correlation**: Described the strength of a **linear** relationship between two variables that are interval or ratio in nature.
- **Spearman's rank-order correlation**: Describes the strength of a **monotonic** relationship between two variables that are ordinal, interval, or ratio. **It is more robust to outliers**.
- The **coefficient of determination** (r^2) describes the amount of variance in one variable that is explained by the other variable.
- **Correlation != Causation**

R command (hint: add `use = "complete.obs"` to drop NA values)

```
pearson  <- cor(x, y, method = "pearson", use = "complete.obs")
spearman <- cor(x, y, method = "spearman", use = "complete.obs")
```

Week 4: *Exploring Data*

1. Exploring Data
2. Data Types
3. Centrality & Variability
4. Visualizing Centrality & Variability

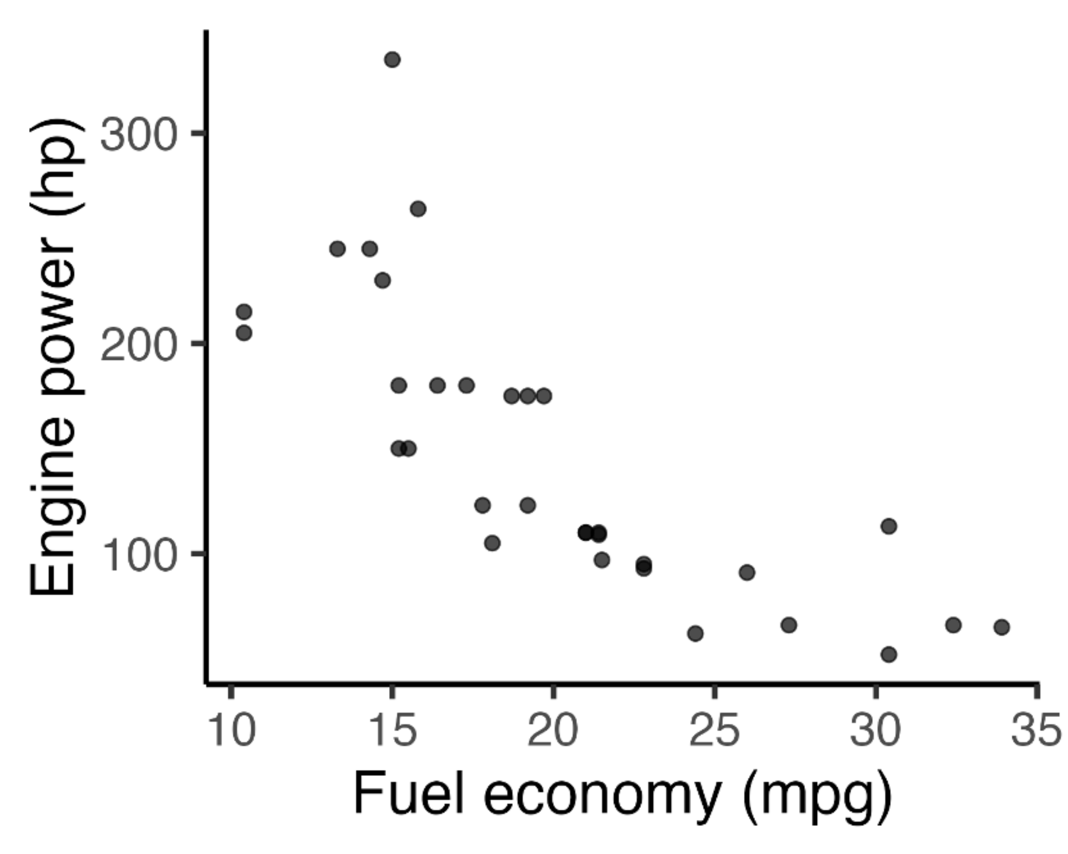
BREAK

5. Correlation
6. Visualizing Correlation
7. Visualizing Relationships

Scatterplots: The correlation workhorse

```r
scatterplot <- mtcars %>%
  ggplot() +
  geom_point(
    aes(x = mpg, y = hp),
    size = 2, alpha = 0.7
  ) +
  theme_classic(base_size = 20) +
  labs(
    x = 'Fuel economy (mpg)',
    y = 'Engine power (hp)'
  )

scatterplot
```



Adding a correlation label to a chart

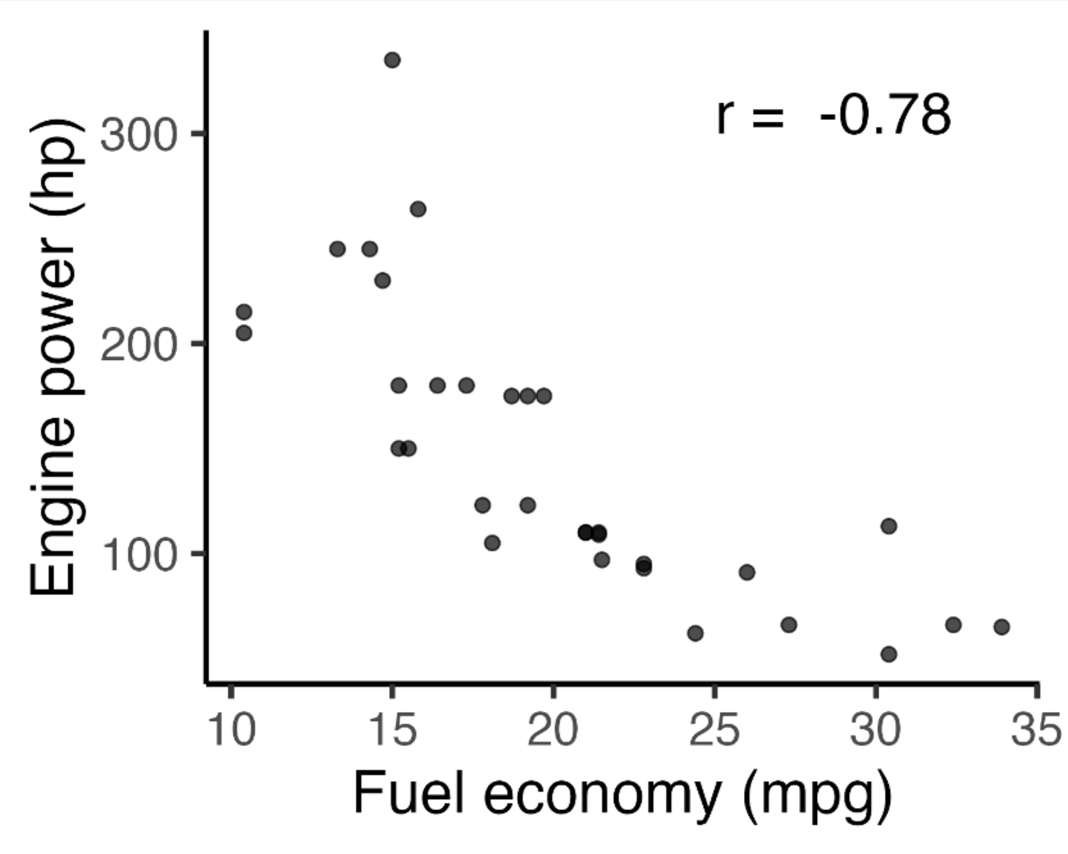
Make the correlation label

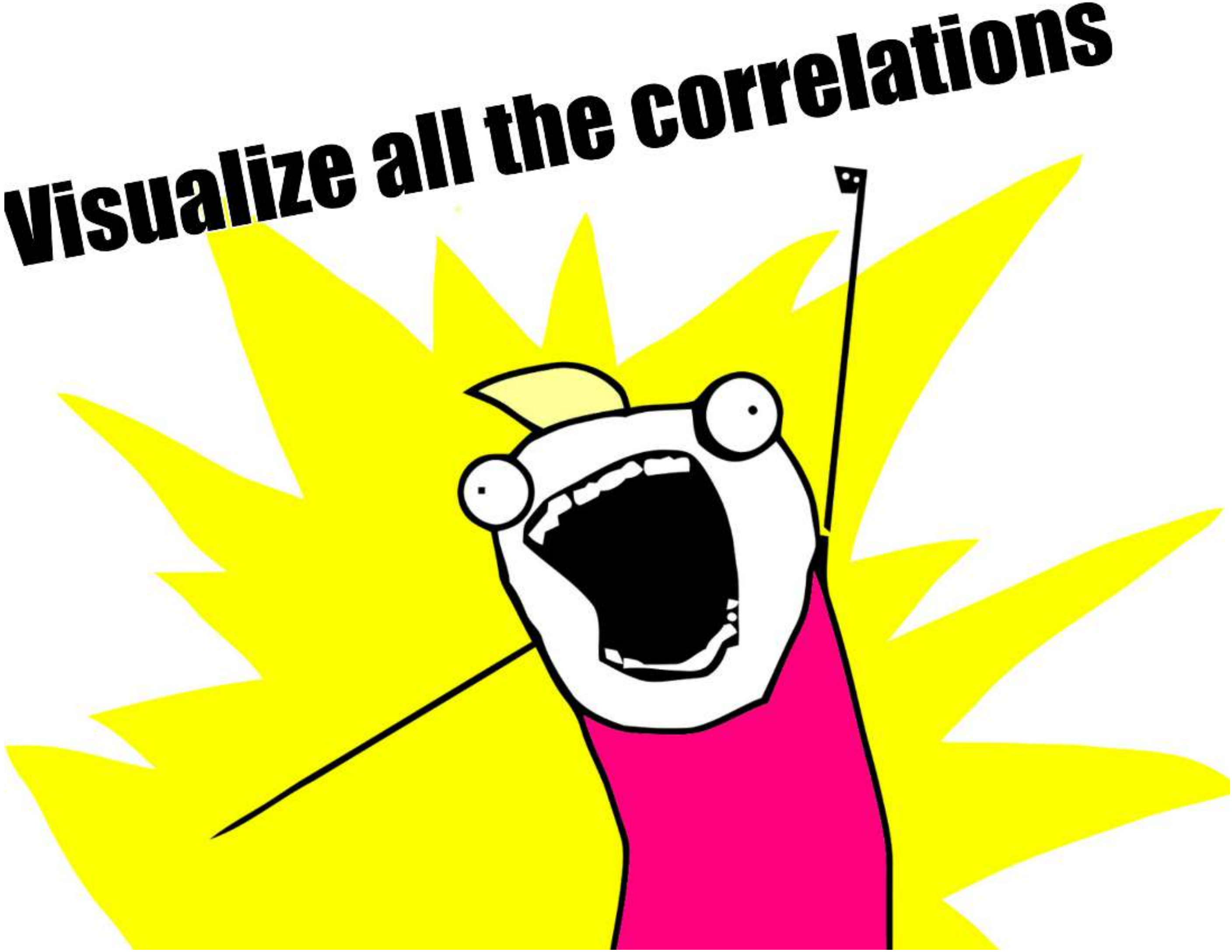
```
corr <- cor(
    mtcars$mpg, mtcars$hp,
    method = 'pearson')

corrLabel <- paste('r = ', round(corr, 2))
```

Add label to the chart with `annotate()`

```
scatterplot +
  annotate(
    geom = 'text',
    x = 25, y = 310,
    label = corrLabel,
    hjust = 0, size = 7
  )
```

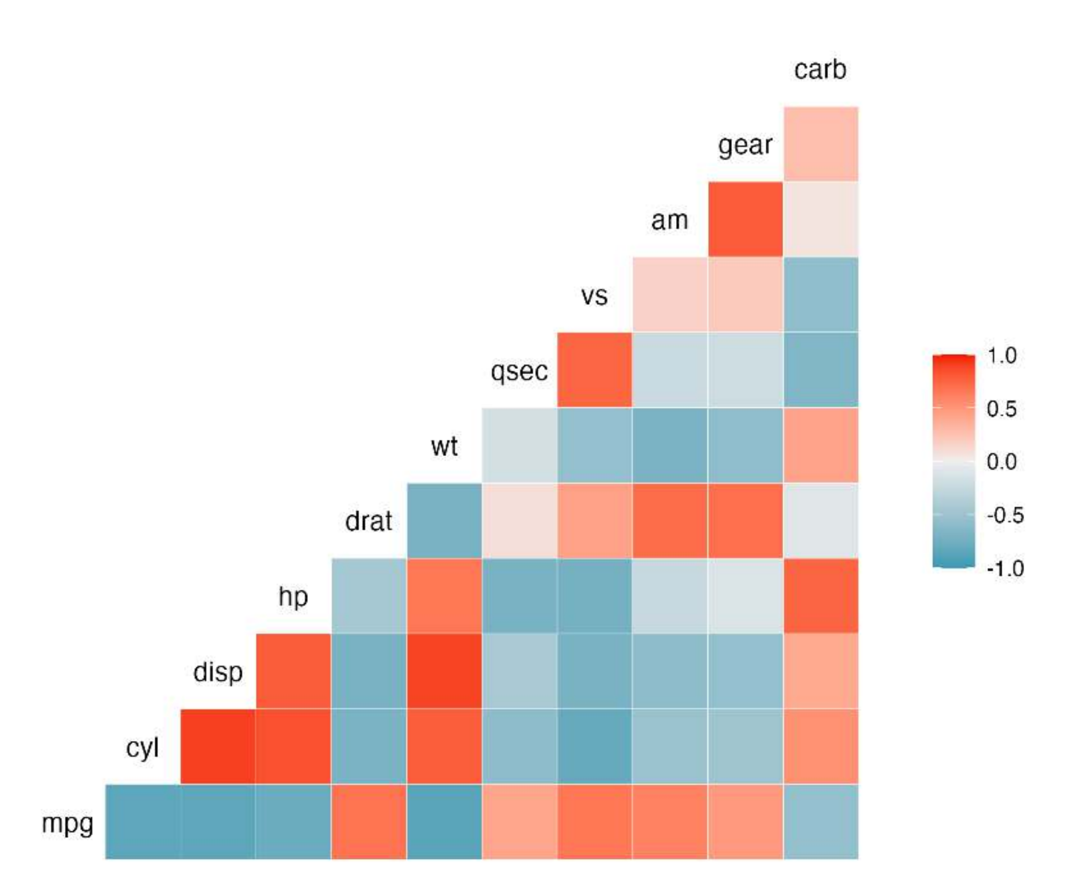



Visualize all the correlations

Visualize all the correlations: `ggcorr()`

```r
library('GGally')
```

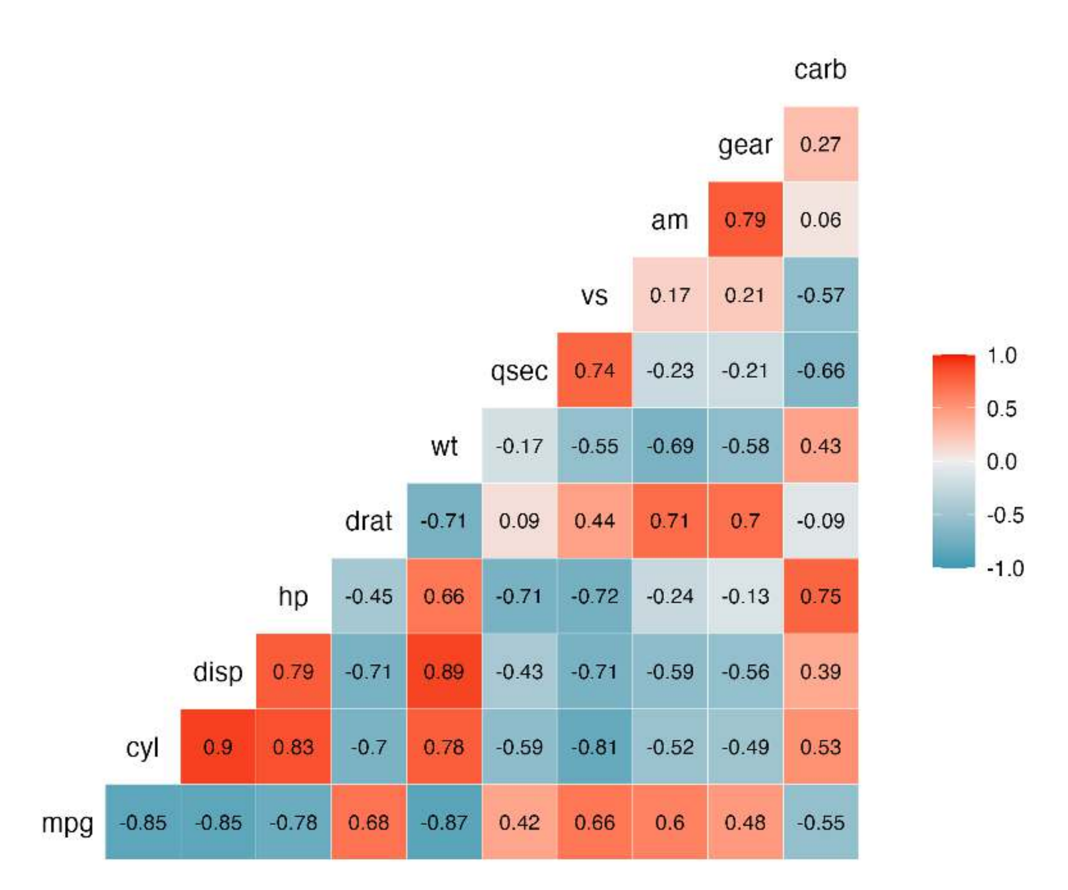
```r
mtcars %>%
    ggcorr()
```



Visualizing correlations: ggcorr()

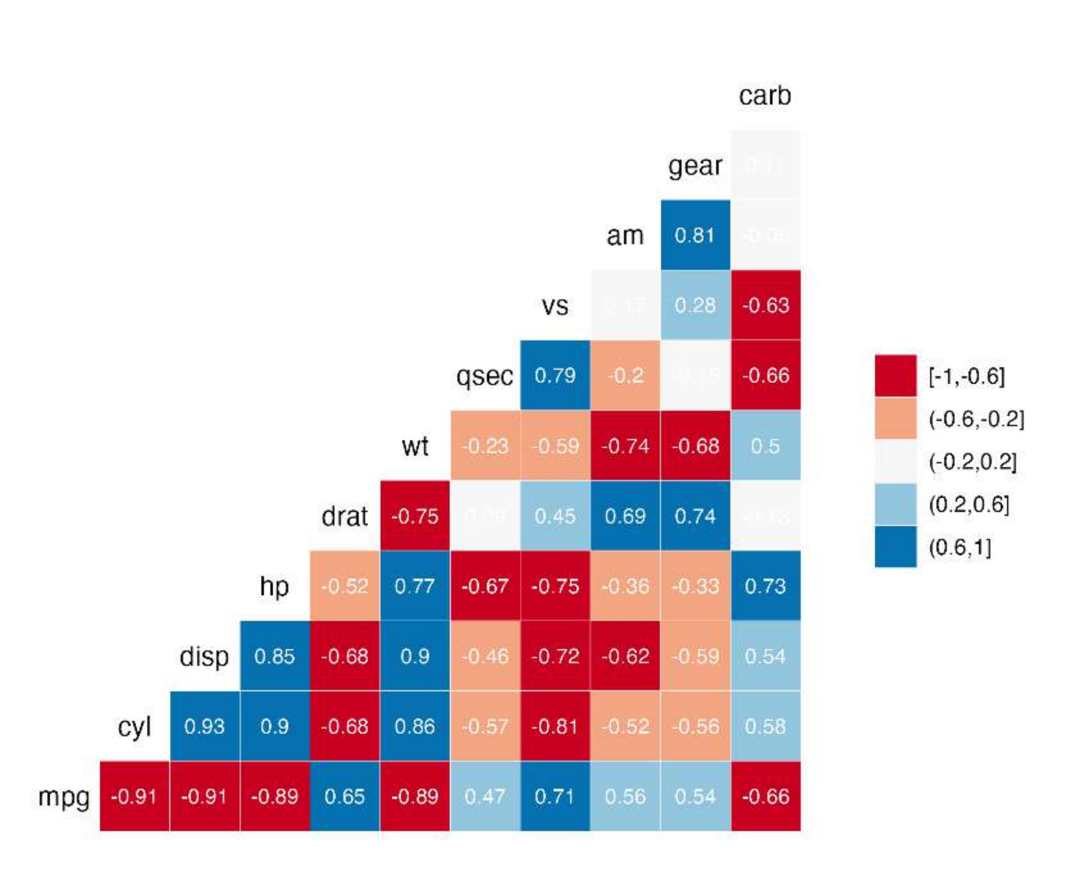
```r
library('GGally')
```

```r
mtcars %>%
    ggcorr(label = TRUE,
           label_size = 3,
           label_round = 2)
```



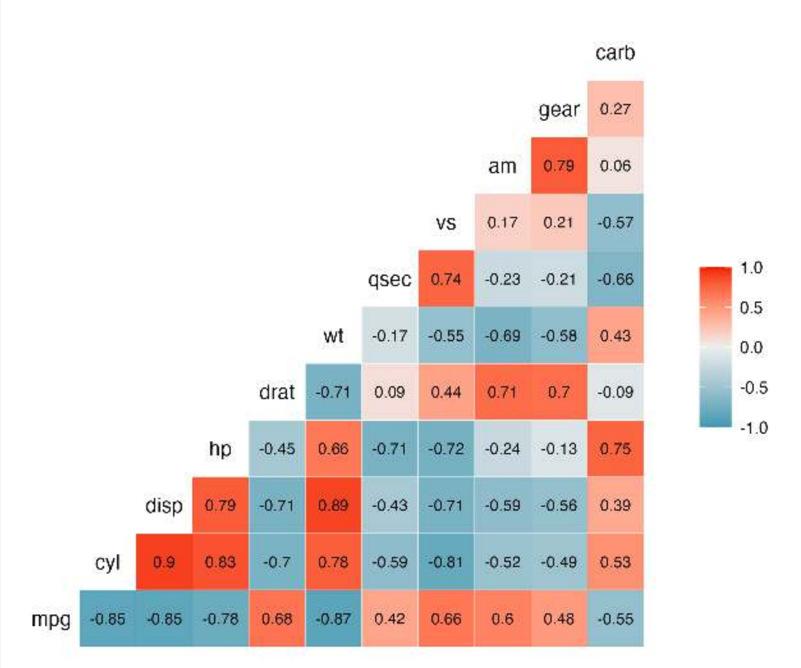
Visualizing correlations: `ggcorr()`

```
ggcor_mtcars_final <- mtcars %>%
    ggcorr(label = TRUE,
           label_size = 3,
           label_round = 2,
           label_color = 'white',
           nbreaks = 5,
           palette = "RdBu")
```



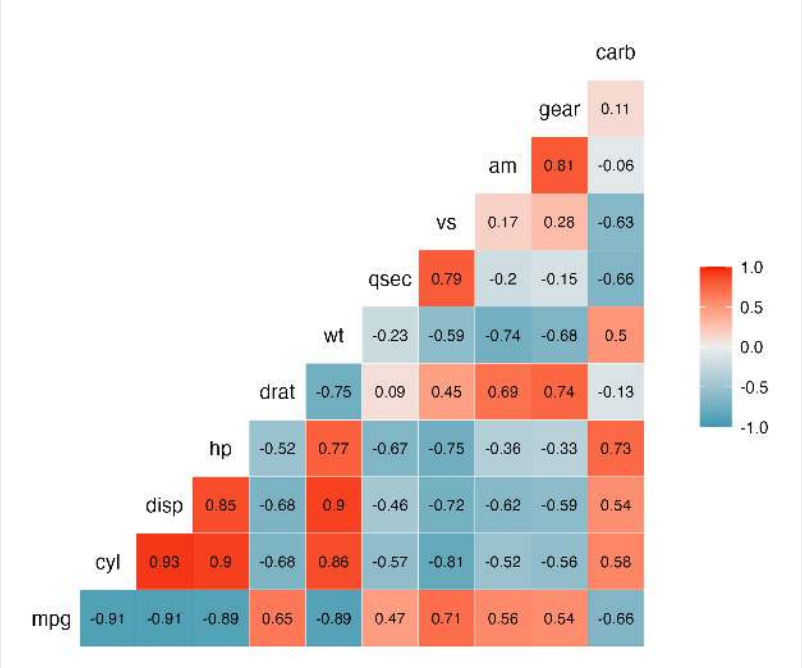
Pearson

```r
mtcars %>%
    ggcorr(label = TRUE,
           label_size = 3,
           label_round = 2,
           method = c("pairwise", "pearson"
```



Spearman

```r
mtcars %>%
    ggcorr(label = TRUE,
           label_size = 3,
           label_round = 2,
           method = c("pairwise", "spearman
```

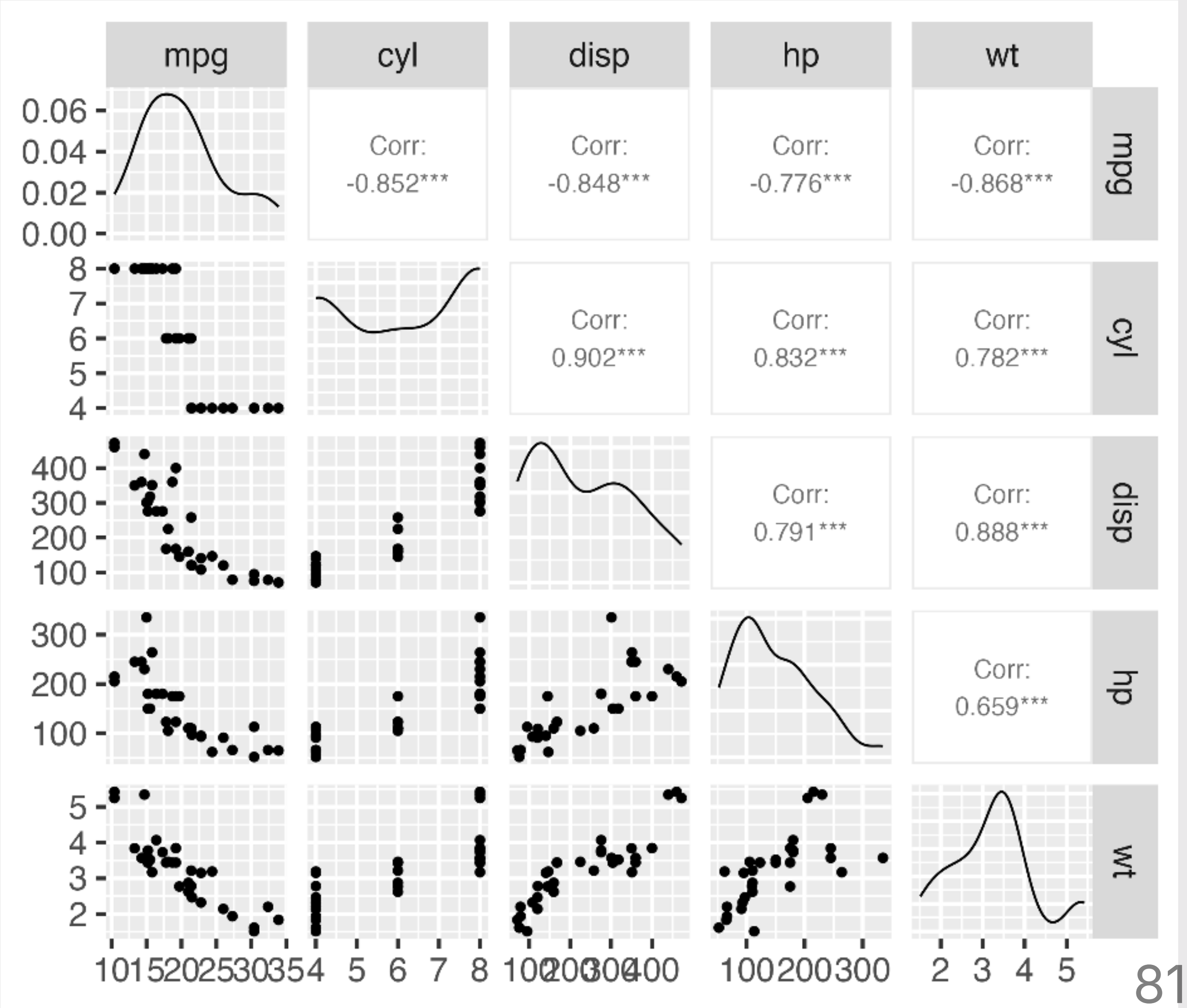


Correlograms: `ggpairs()`

```r
library('GGally')
```

```r
mtcars %>%
    select(mpg, cyl, disp, hp, wt)
    ggpairs()
```

- Look for linear relationships
- View distribution of each variable

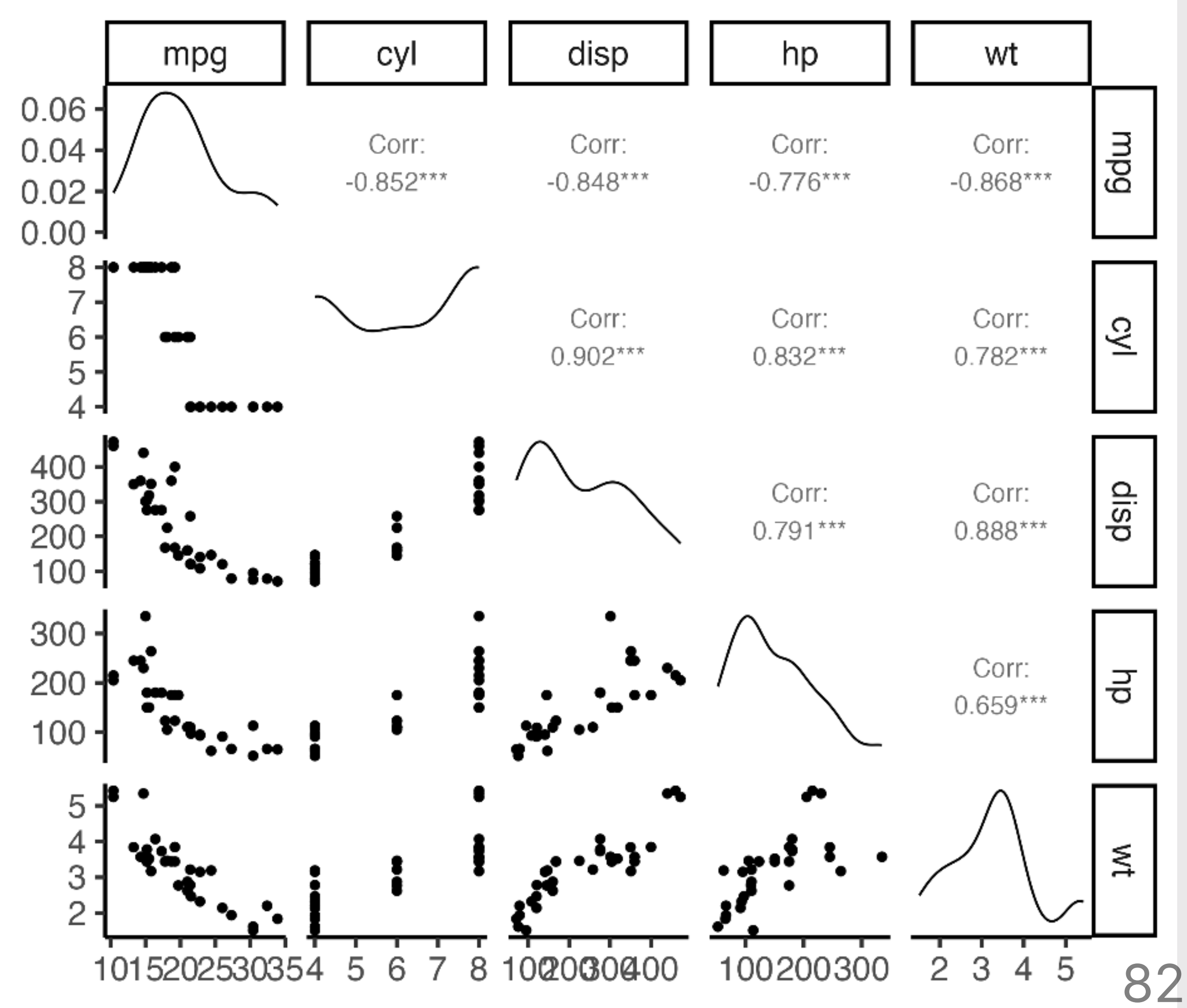


Correlograms: `ggpairs()`

```r
library('GGally')
```

```r
mtcars %>%
    select(mpg, cyl, disp, hp, wt) %>%
    ggpairs() +
    theme_classic()
```

- Look for linear relationships
- View distribution of each variable



Your turn

15:00

Using the `penguins` data frame:

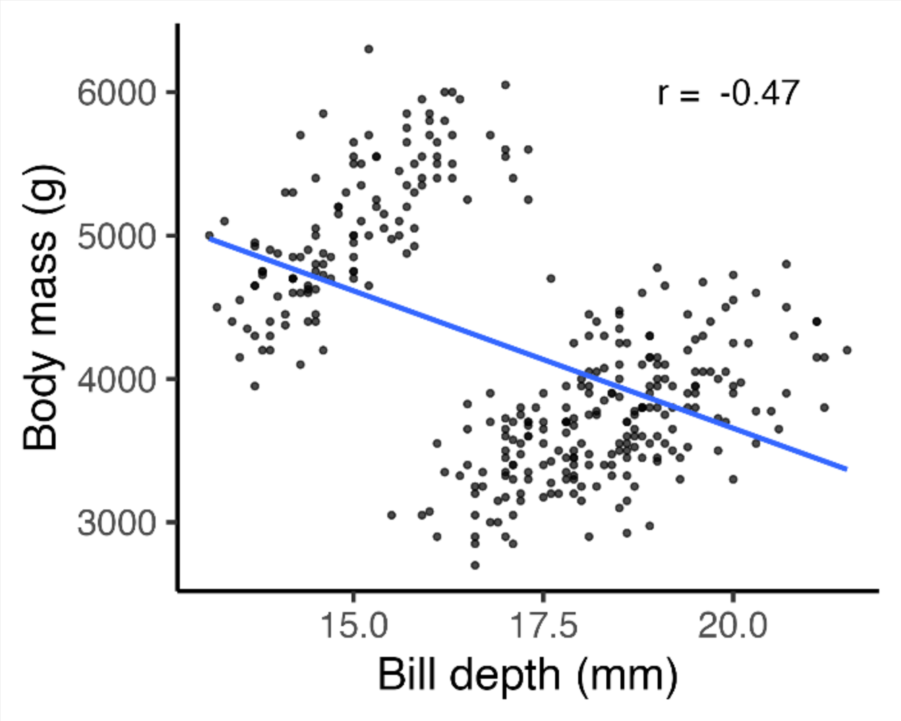
1. Find the two variables with the largest correlation in absolute value (i.e. closest to -1 or 1).
2. Create a scatter plot of those two variables.
3. Add an annotation for the Pearson correlation coefficient.

[palmerpenguins library]

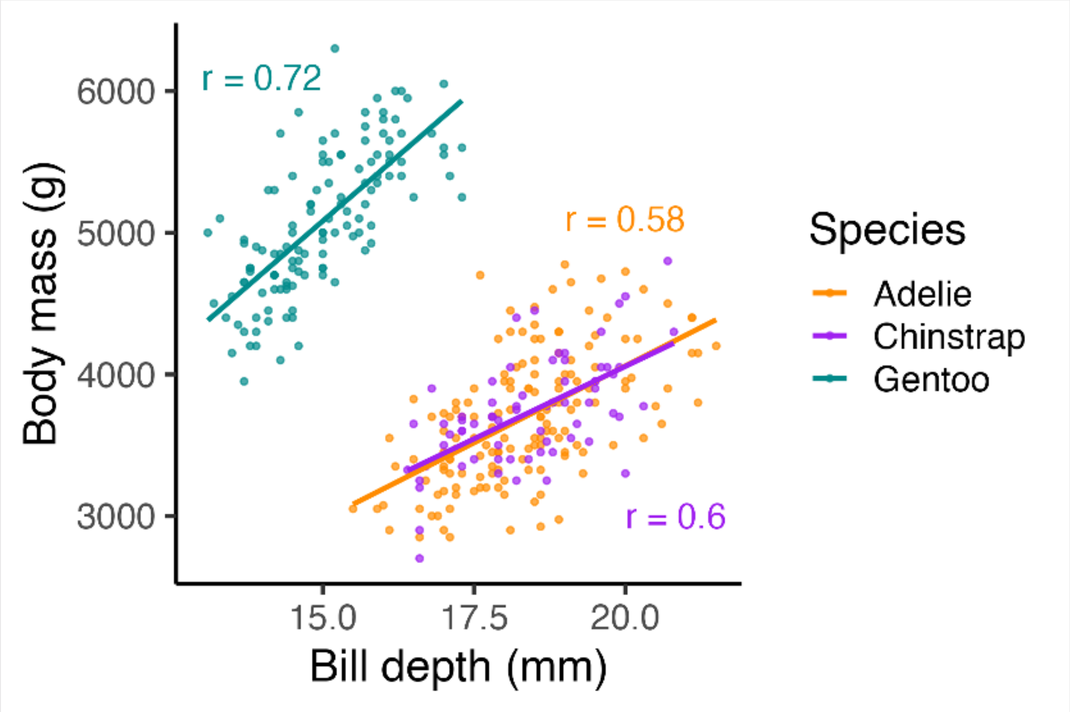
Artwork by @allison_horst

Simpson's Paradox: when correlation betrays you

Body mass vs. Bill depth



Body mass vs. Bill depth



Week 4: *Exploring Data*

1. Exploring Data
2. Data Types
3. Centrality & Variability
4. Visualizing Centrality & Variability

BREAK

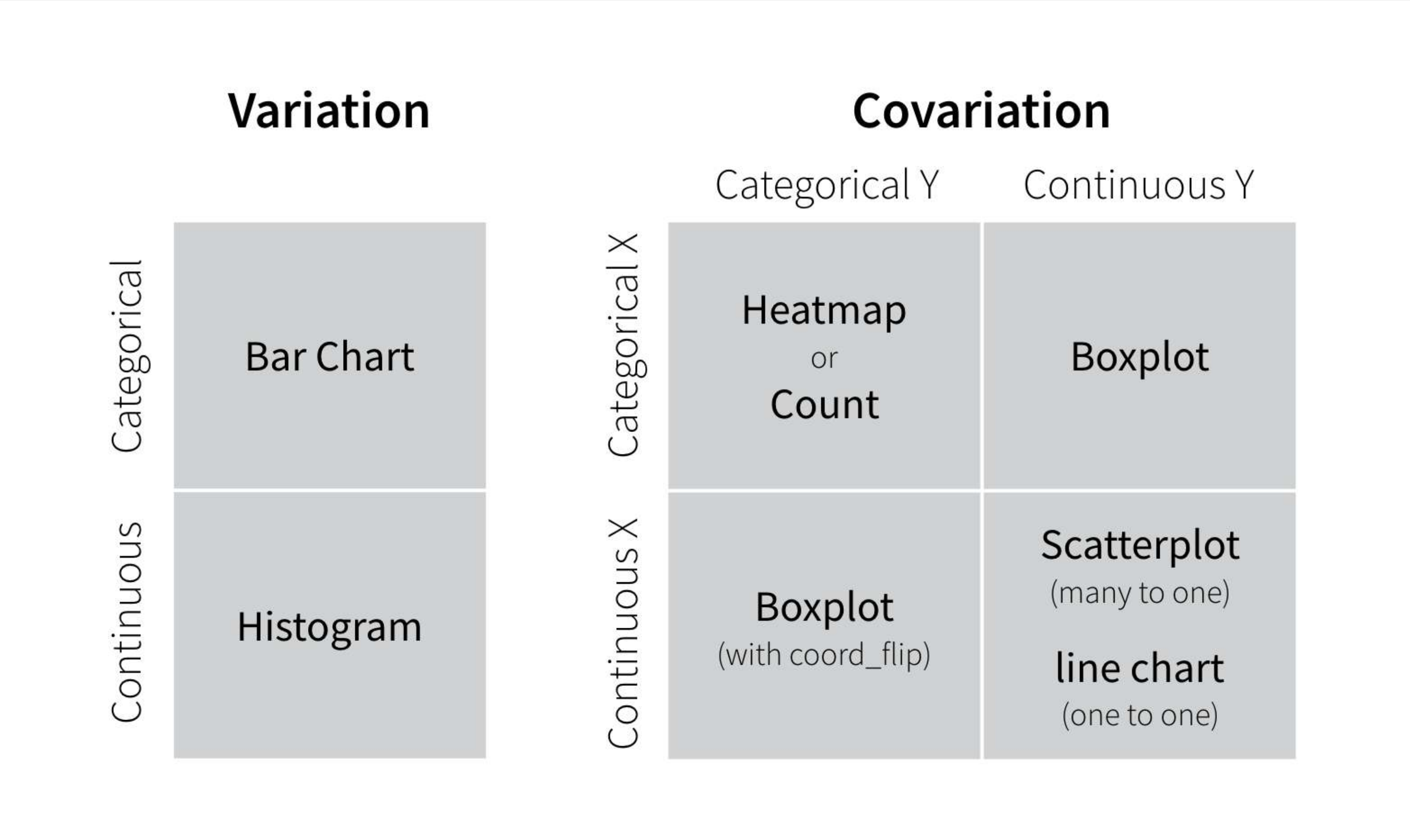
5. Correlation
6. Visualizing Correlation
7. Visualizing Relationships

Visualizing variation

Ask yourself:

- What type of **variation** occurs within my variables?
- What type of **covariation** occurs between my variables?

Check out these guides



Two **Categorical** Variables

Summarize with a table of counts

```r
wildlife_impacts %>%
    count(operator, time_of_day)
```

```
#> # A tibble: 20 × 3
#>    operator           time_of_day     n
#>    <chr>              <chr>       <int>
#>  1 AMERICAN AIRLINES  Dawn          458
#>  2 AMERICAN AIRLINES  Day          7809
#>  3 AMERICAN AIRLINES  Dusk          584
#>  4 AMERICAN AIRLINES  Night        3710
#>  5 AMERICAN AIRLINES  <NA>         2326
#>  6 DELTA AIR LINES    Dawn          267
#>  7 DELTA AIR LINES    Day          4846
#>  8 DELTA AIR LINES    Dusk          353
#>  9 DELTA AIR LINES    Night        2090
#> 10 DELTA AIR LINES    <NA>         1449
#> 11 SOUTHWEST AIRLINES Dawn          394
#> 12 SOUTHWEST AIRLINES Day          9109
```

Two **Categorical** Variables

Convert to "wide" format with `pivot_wider()` to make it easier to compare values

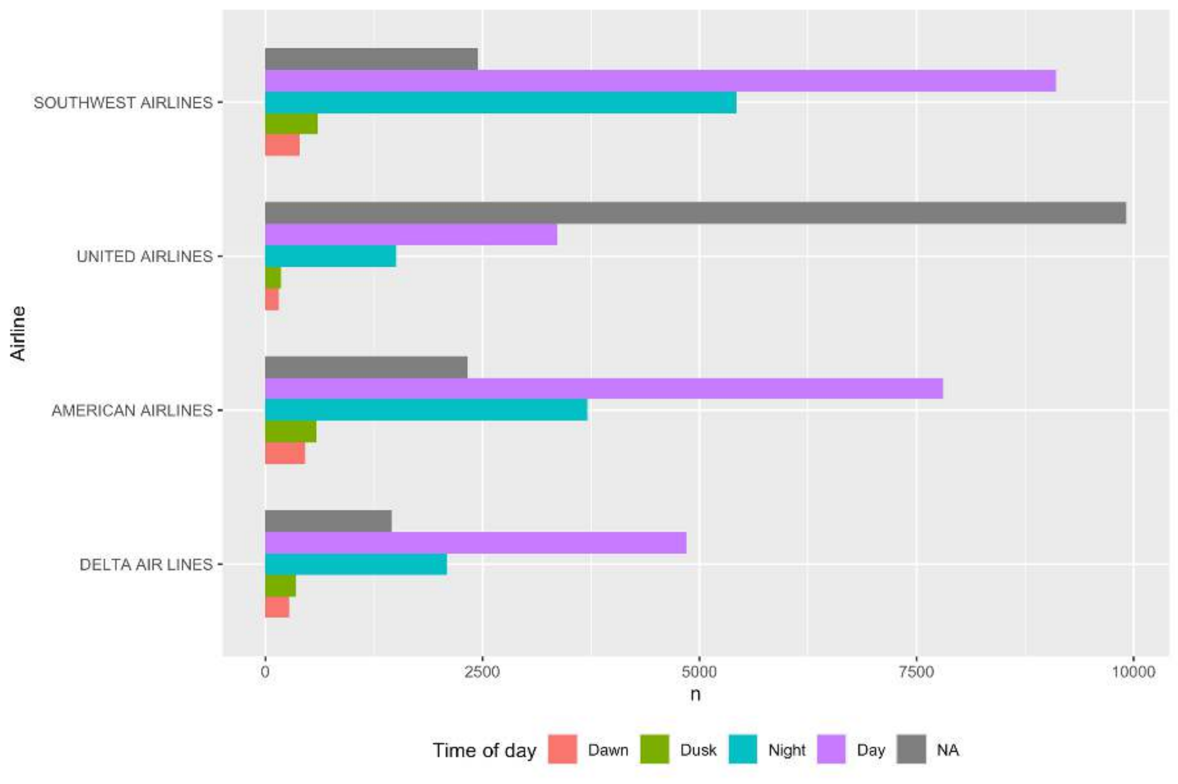
```
wildlife_impacts %>%
    count(operator, time_of_day) %>%
    pivot_wider(names_from = time_of_day, values_from = n)
```

```
#> # A tibble: 4 × 6
#>   operator            Dawn   Day  Dusk Night  `NA`
#>   <chr>              <int> <int> <int> <int> <int>
#> 1 AMERICAN AIRLINES    458  7809   584  3710  2326
#> 2 DELTA AIR LINES      267  4846   353  2090  1449
#> 3 SOUTHWEST AIRLINES   394  9109   599  5425  2443
#> 4 UNITED AIRLINES      151  3359   181  1510  9915
```

Two **Categorical** Variables

Visualize with bars:
map **fill** to denote 2nd categorical var

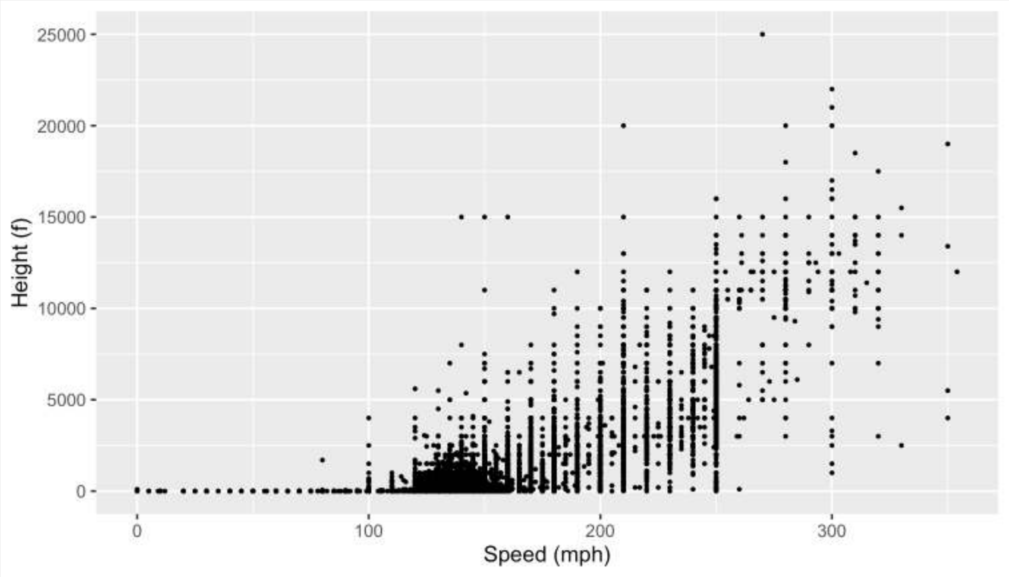
```
wildlife_impacts %>%
  count(operator, time_of_day) %>%
  ggplot() +
  geom_col(
    aes(
      x = n,
      y = reorder(operator, n),
      fill = reorder(time_of_day, n)
    ),
    width = 0.7,
    position = 'dodge') +
  theme(legend.position = "bottom") +
  labs(
    fill = "Time of day",
    y = "Airline"
  )
```



Two **Continuous** Variables

Visualize with scatterplot - looking for *clustering* and/or *correlational* relationship

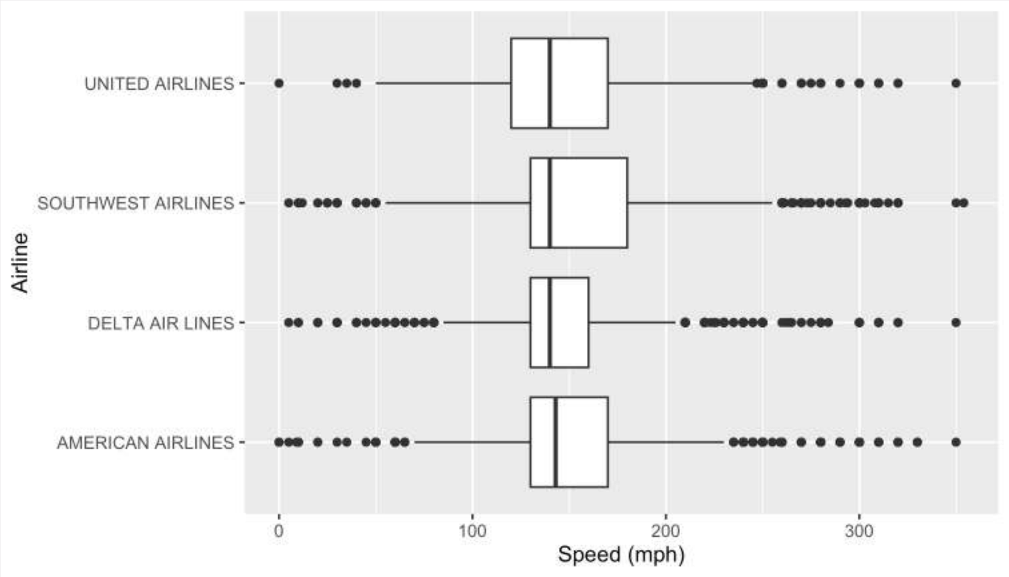
```
ggplot(wildlife_impacts) +
  geom_point(
    aes(
      x = speed,
      y = height
    ),
    size = 0.5) +
  labs(
    x = 'Speed (mph)',
    y = 'Height (f)'
  )
```



One **Continuous**, One **Categorical**

Visualize with **boxplot**

```
ggplot(wildlife_impacts) +
  geom_boxplot(
    aes(
      x = speed,
      y = operator)
    ) +
  labs(
    x = 'Speed (mph)',
    y = 'Airline'
  )
```



Practice doing EDA

15:00

1) Read in the `candy_rankings.csv` data sets

2) Preview the data, note the data types and what each variable is.

3) Visualize (at least) three *relationships* between two variables (guided by a question) using an appropriate chart:

- Bar chart
- Scatterplot
- Boxplot

Reminders:

You have **4** days until your Project Proposal is due

You have **6** days until your Mini Project 1 is due.

Sign up for meeting slot next week