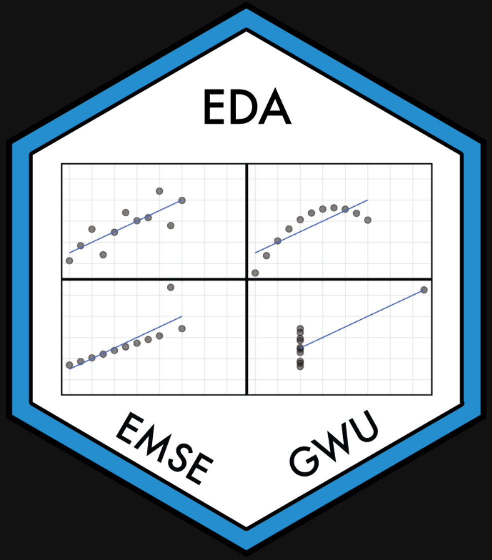


Week 5: *Correlation*

🏛 EMSE 4572: Exploratory Data Analysis

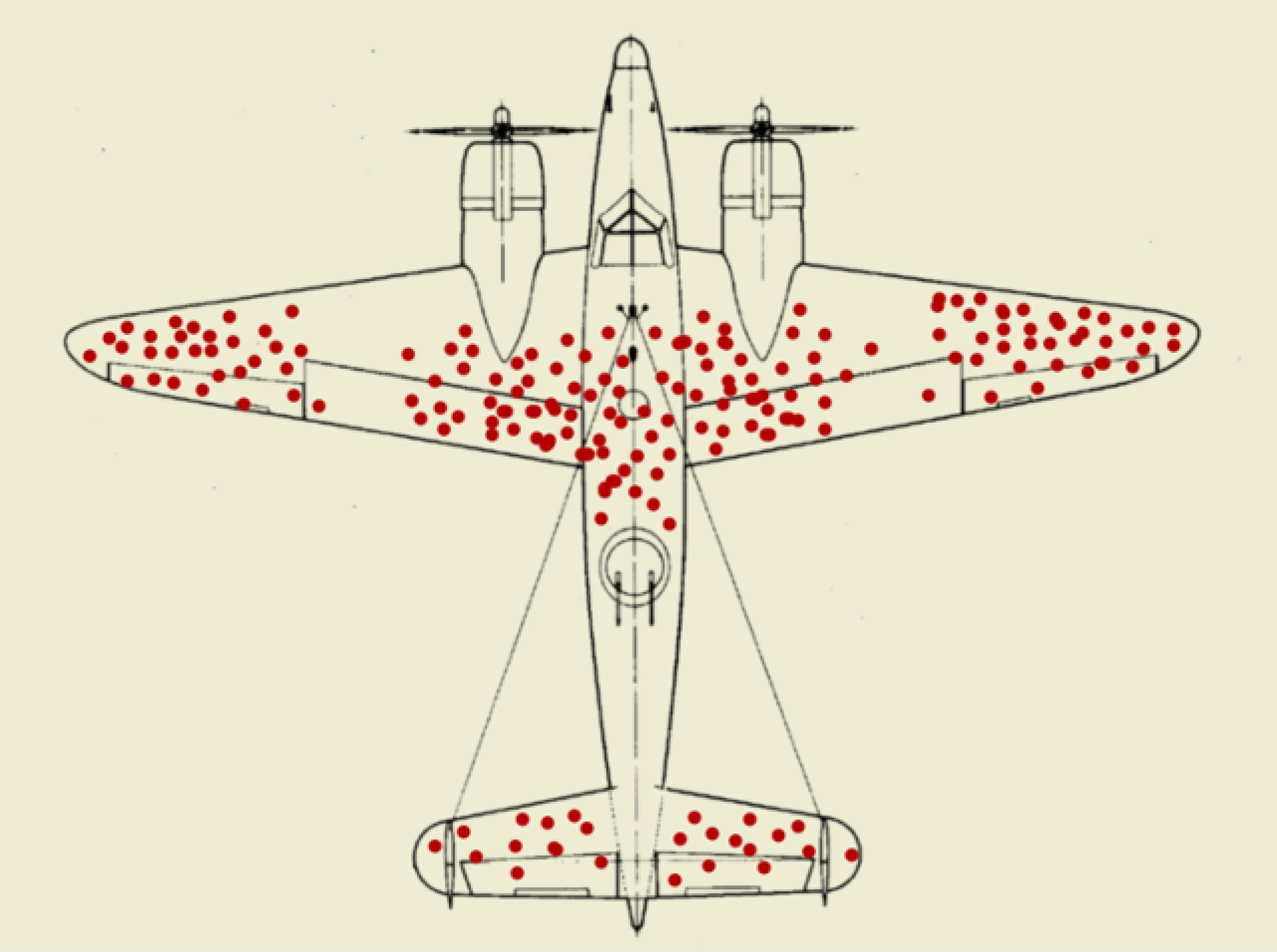
👤 John Paul Helveston

📅 September 28, 2022

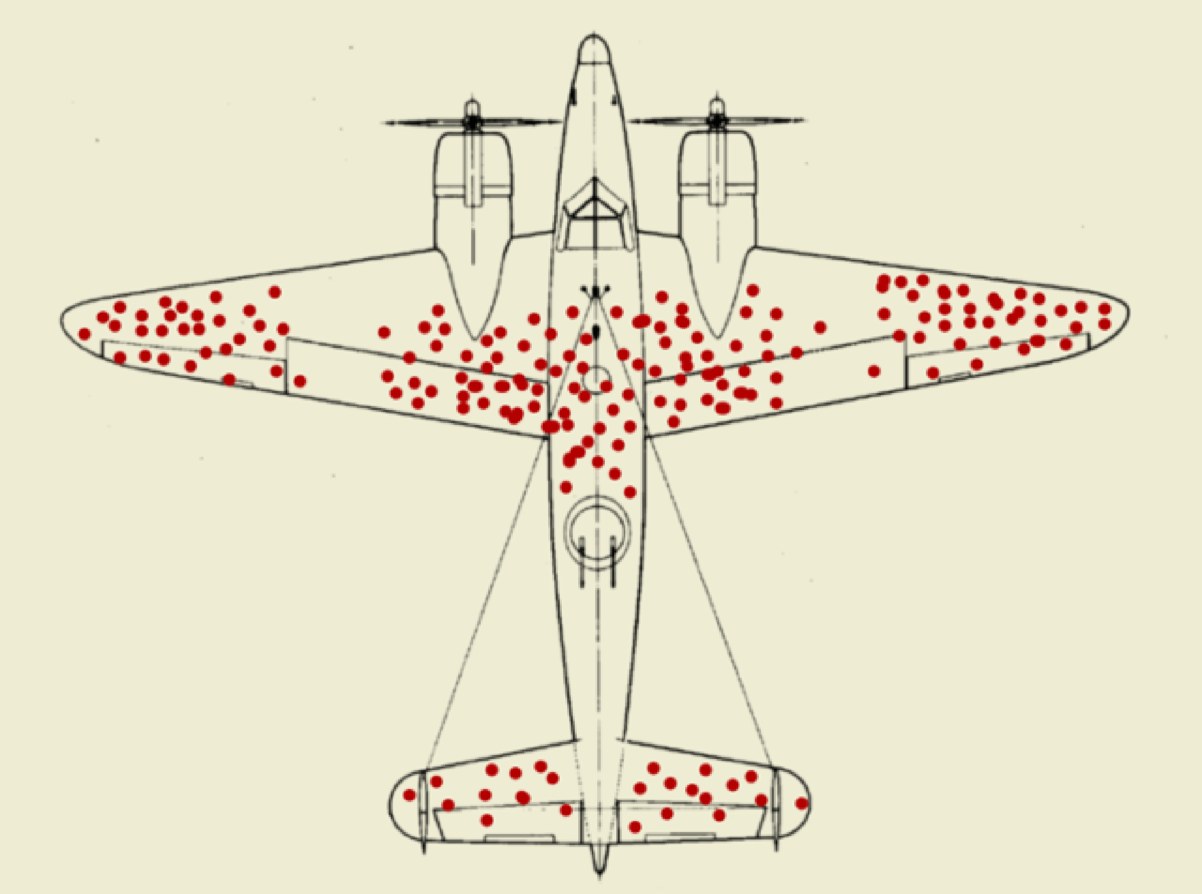
Quick Rmd updates

Tip of the week

All data are biased



Abraham Wald



Today's data

```r
msleep <- read_csv(here::here('data', 'msleep.csv'))
```

New packages:

```r
install.packages('HistData')
install.packages('palmerpenguins')
install.packages('GGally')
```

Week 5: *Correlation*

1. What is correlation?

2. Visualizing correlation

BREAK

3. Linear models

4. Visualizing linear models

Week 5: *Correlation*

1. What is correlation?
2. Visualizing correlation

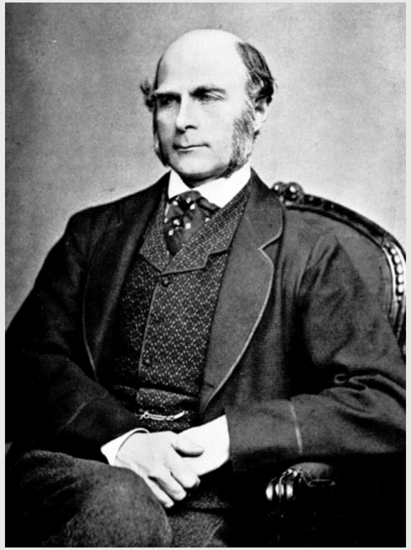
BREAK

3. Linear models
4. Visualizing linear models

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

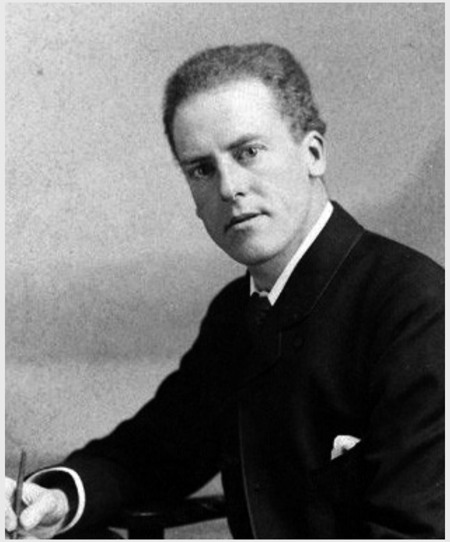
Sir Francis Galton (1822 - 1911)

- 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

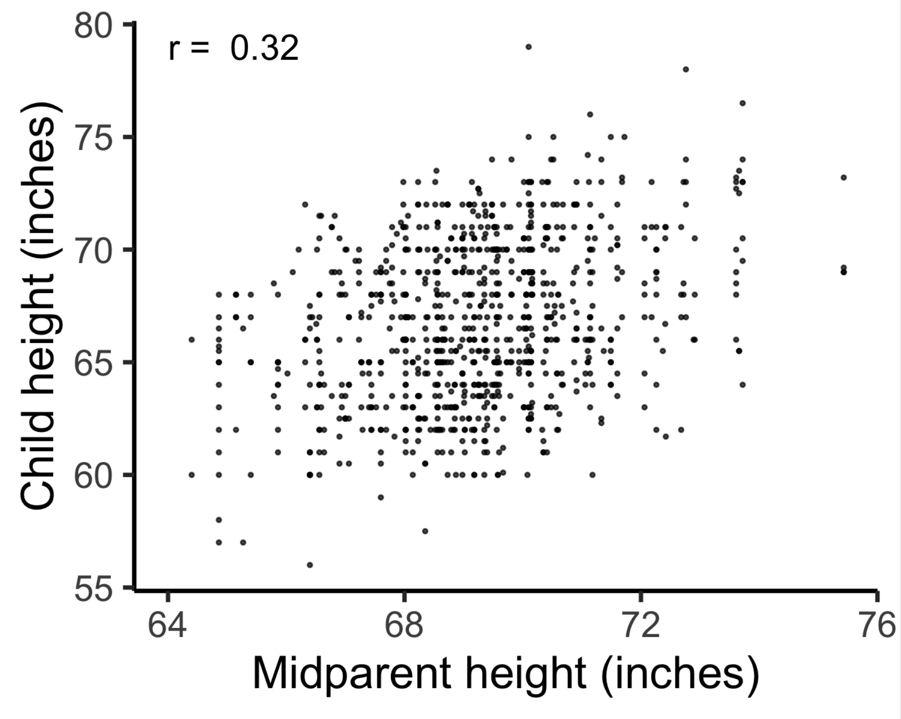
Galton, F. (1886). "Regression towards mediocrity in hereditary stature". *The Journal of the Anthropological Institute of Great Britain and Ireland* 15: 246-263.

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:

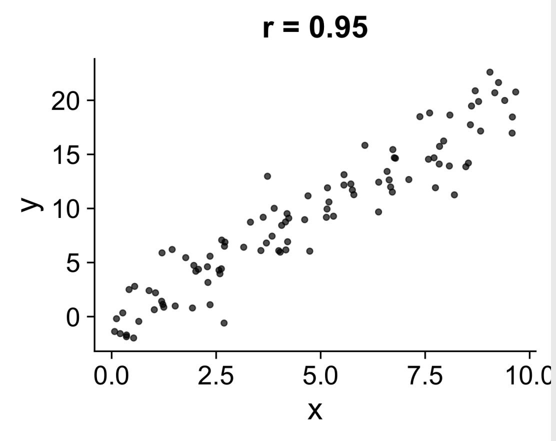
$$r = \frac{\mathrm{Cov}(x,y)}{\mathrm{sd}(x) * \mathrm{sd}(y)}$$

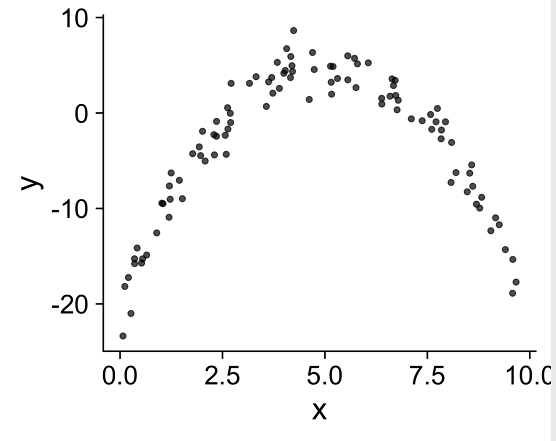
How do you measure correlation?

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

Assumptions:

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





How do you *interpret* r?

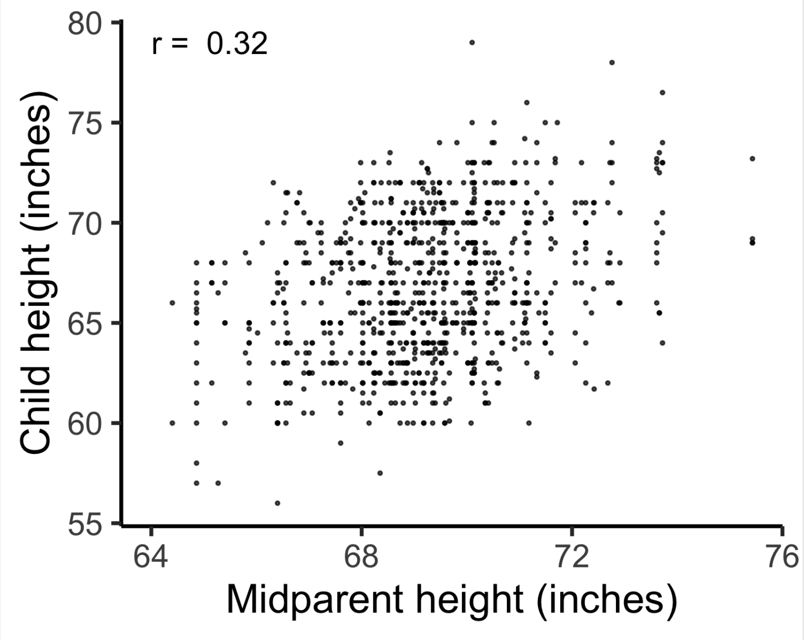
$$r = \frac{\mathrm{Cov}(x,y)}{\mathrm{sd}(x) * \mathrm{sd}(y)}$$

Interpretation:

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

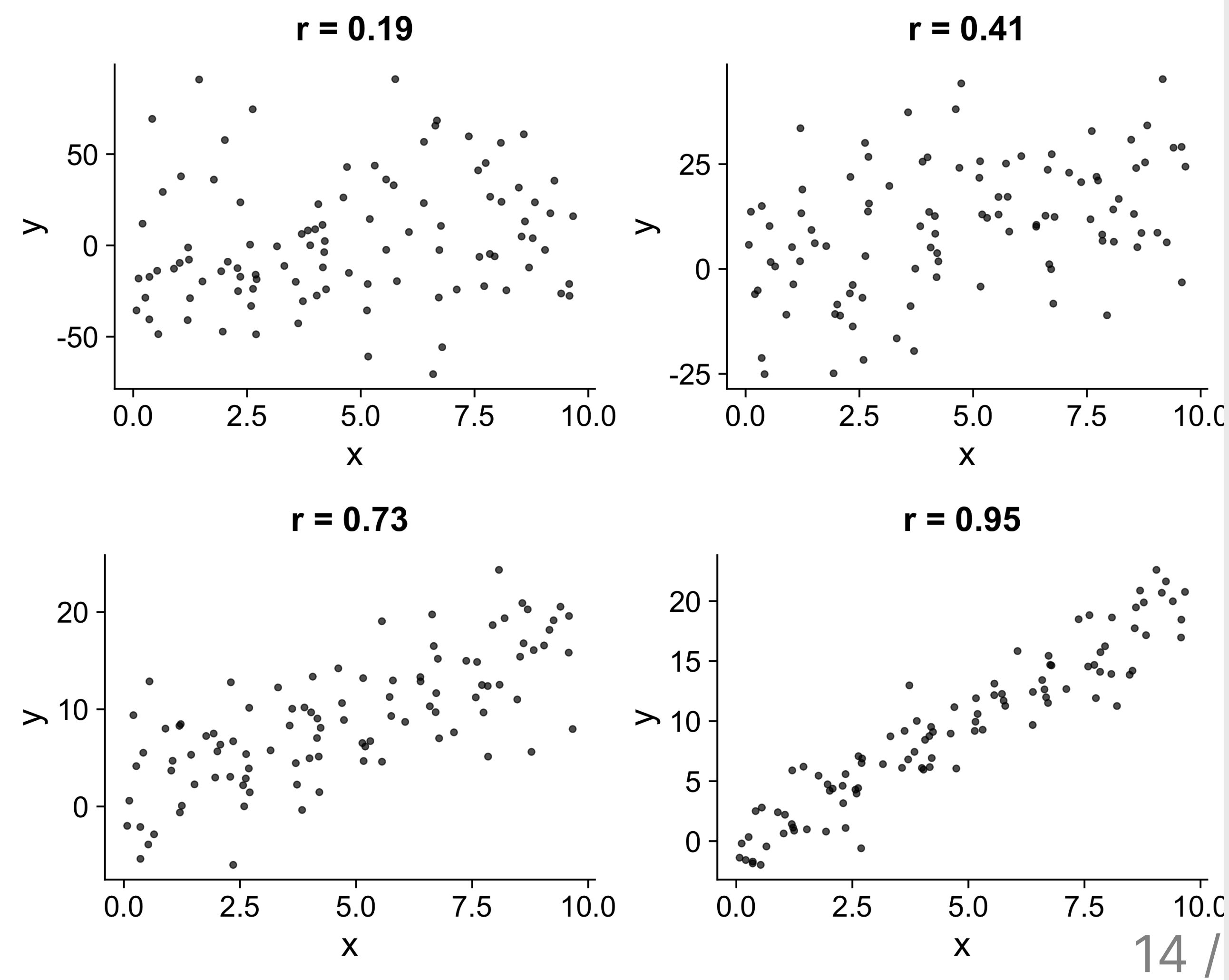
```r
cor(x = GaltonFamilies$midparentHeight,
    y = GaltonFamilies$childHeight,
    method = 'pearson')
```

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



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



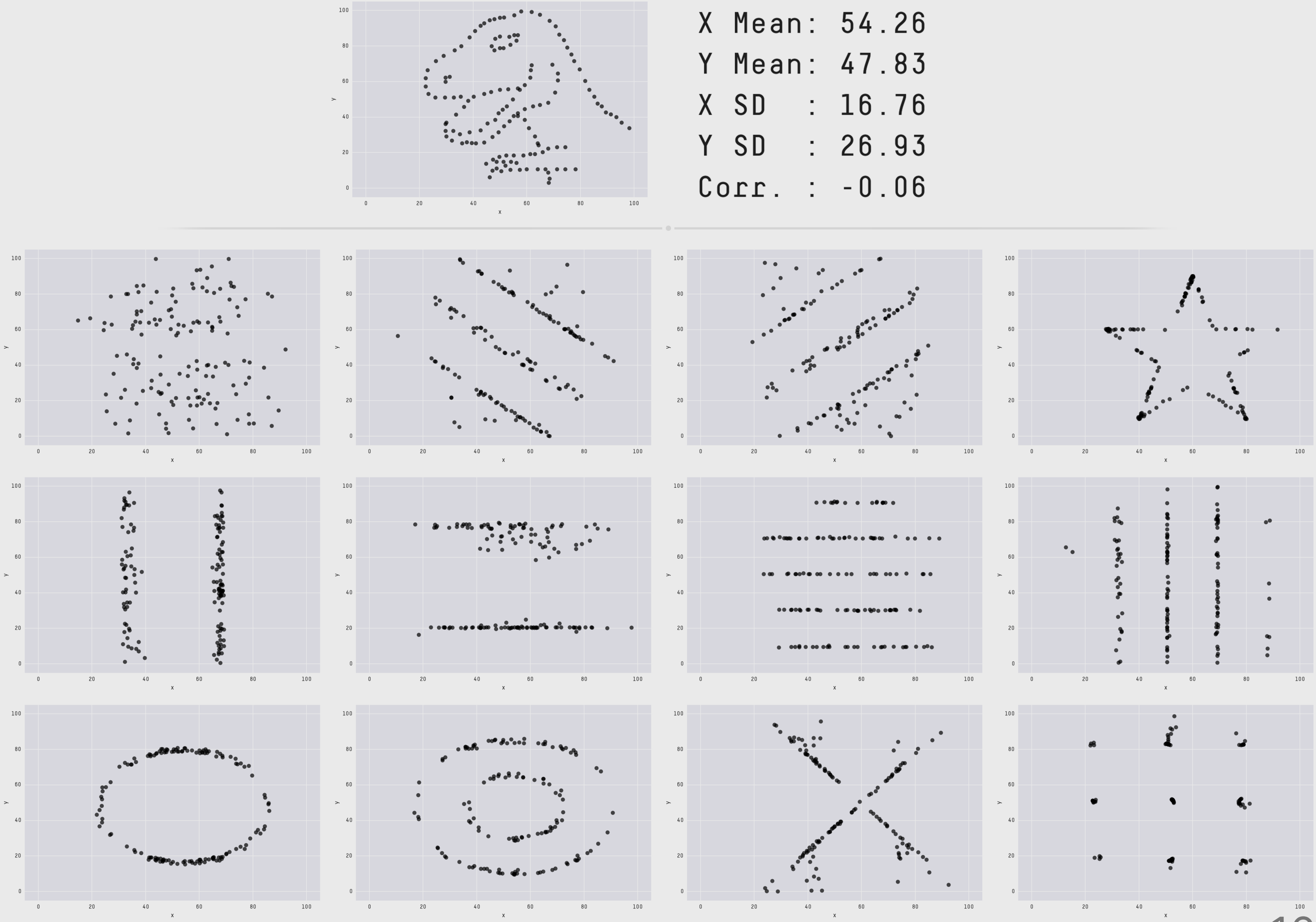
Visualizing correlation is…um…easy, right?

guessthecorrelation.com

Click here to vote!

The datasaurus

(More here)



X Mean: 54.26
Y Mean: 47.83
X SD : 16.76
Y SD : 26.93
Corr. : -0.06

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.

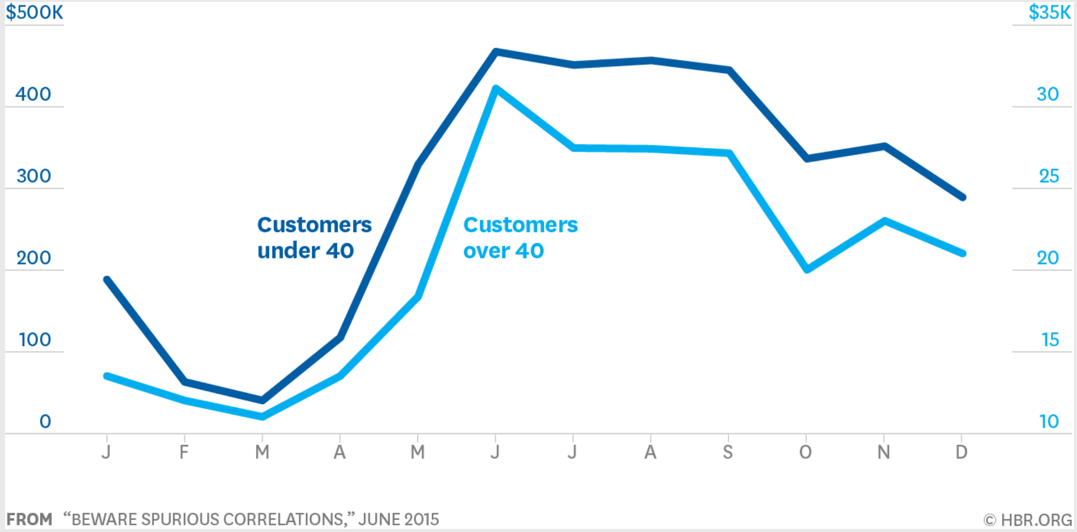
Z causes both X & Y

- Commitment and motivation cause increased training and better performance.

Be weary of dual axes!

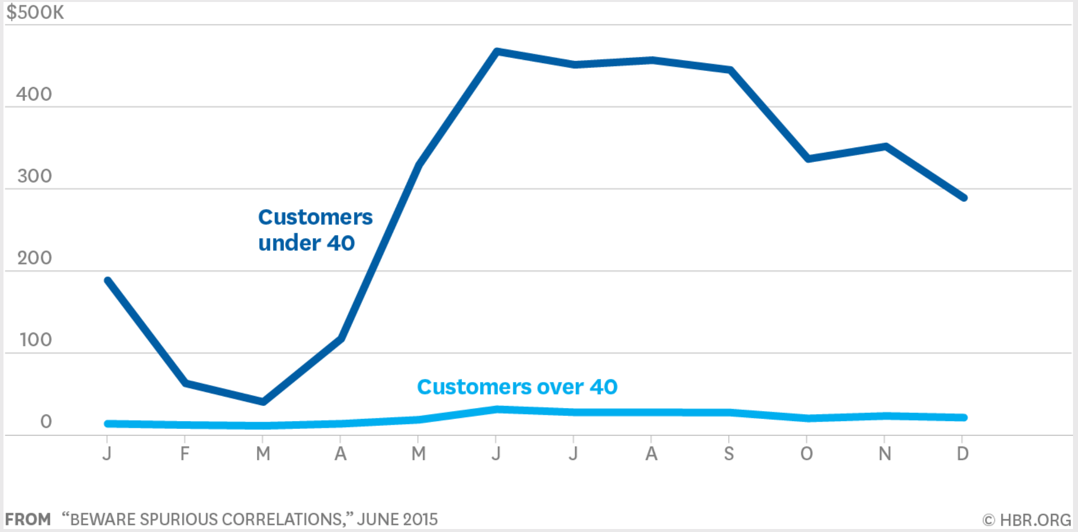
(They can cause spurious correlations)

Dual axes



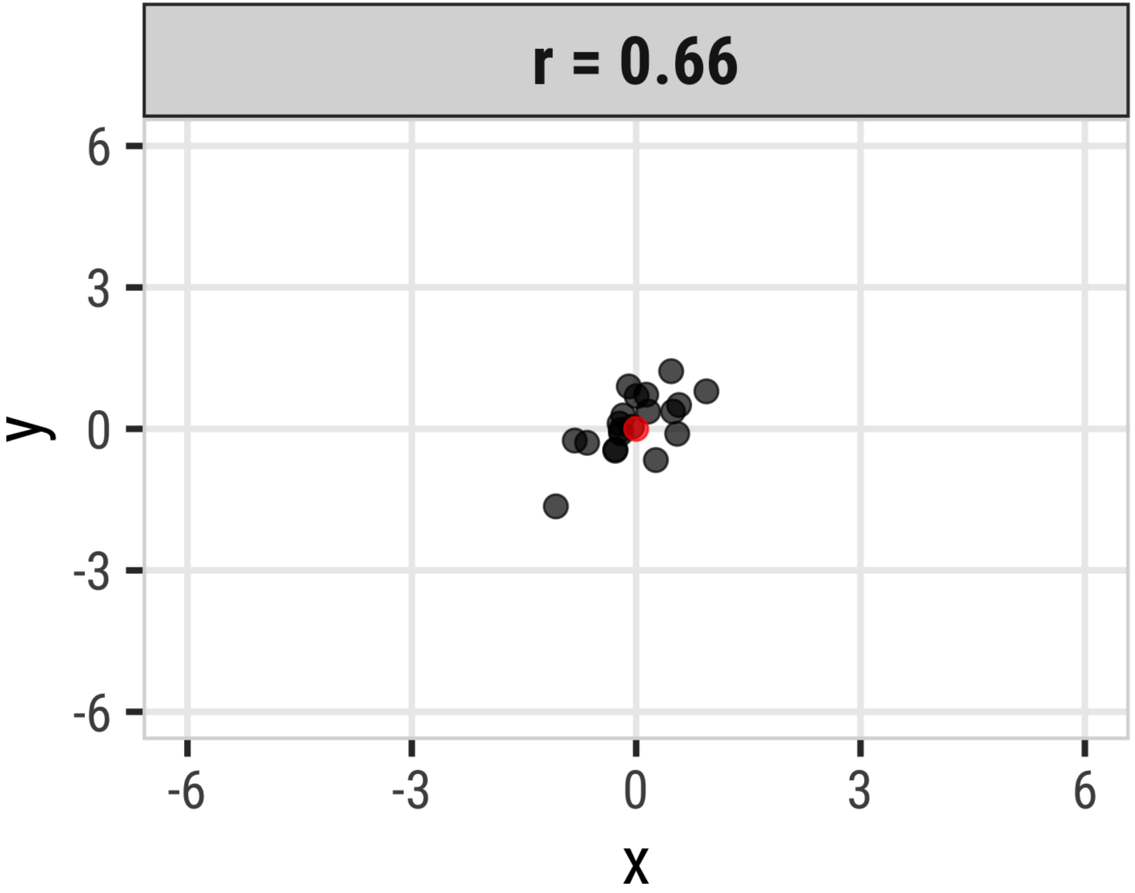
FROM "BEWARE SPURIOUS CORRELATIONS," JUNE 2015 © HBR.ORG

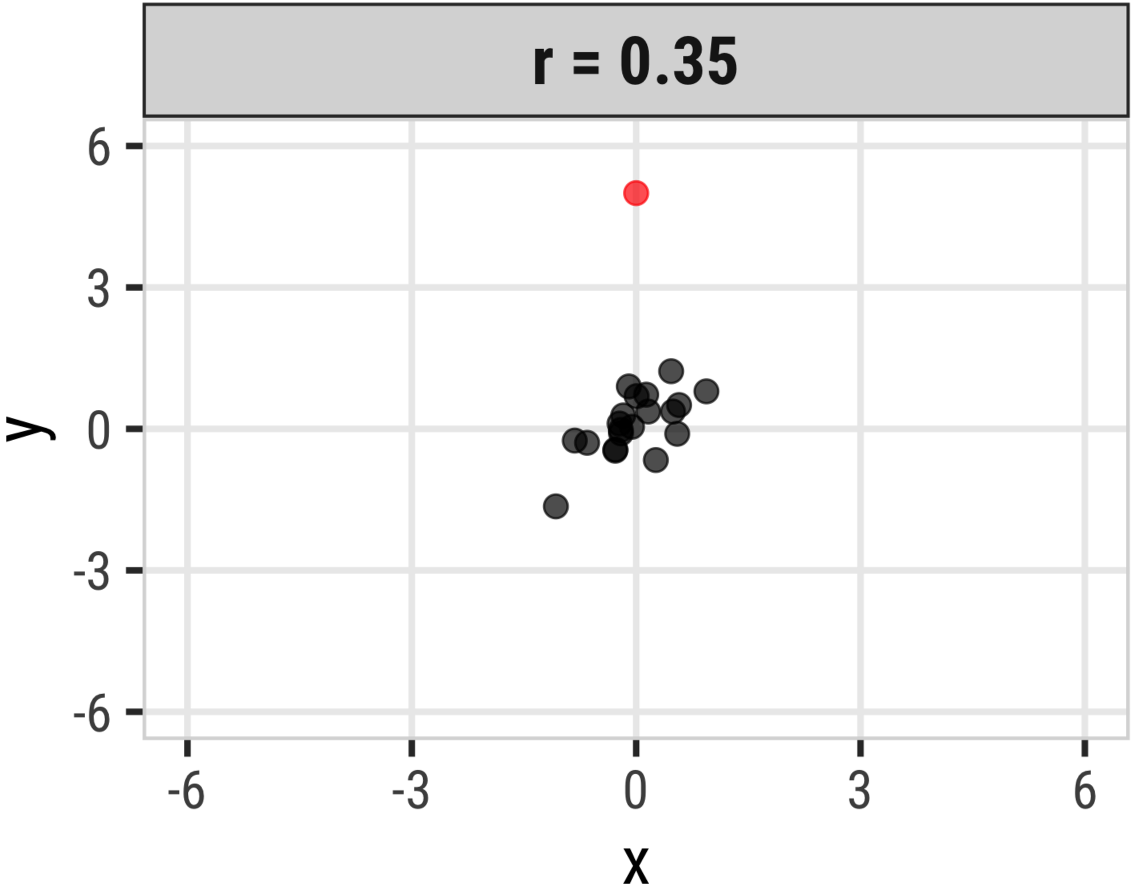
Single axis

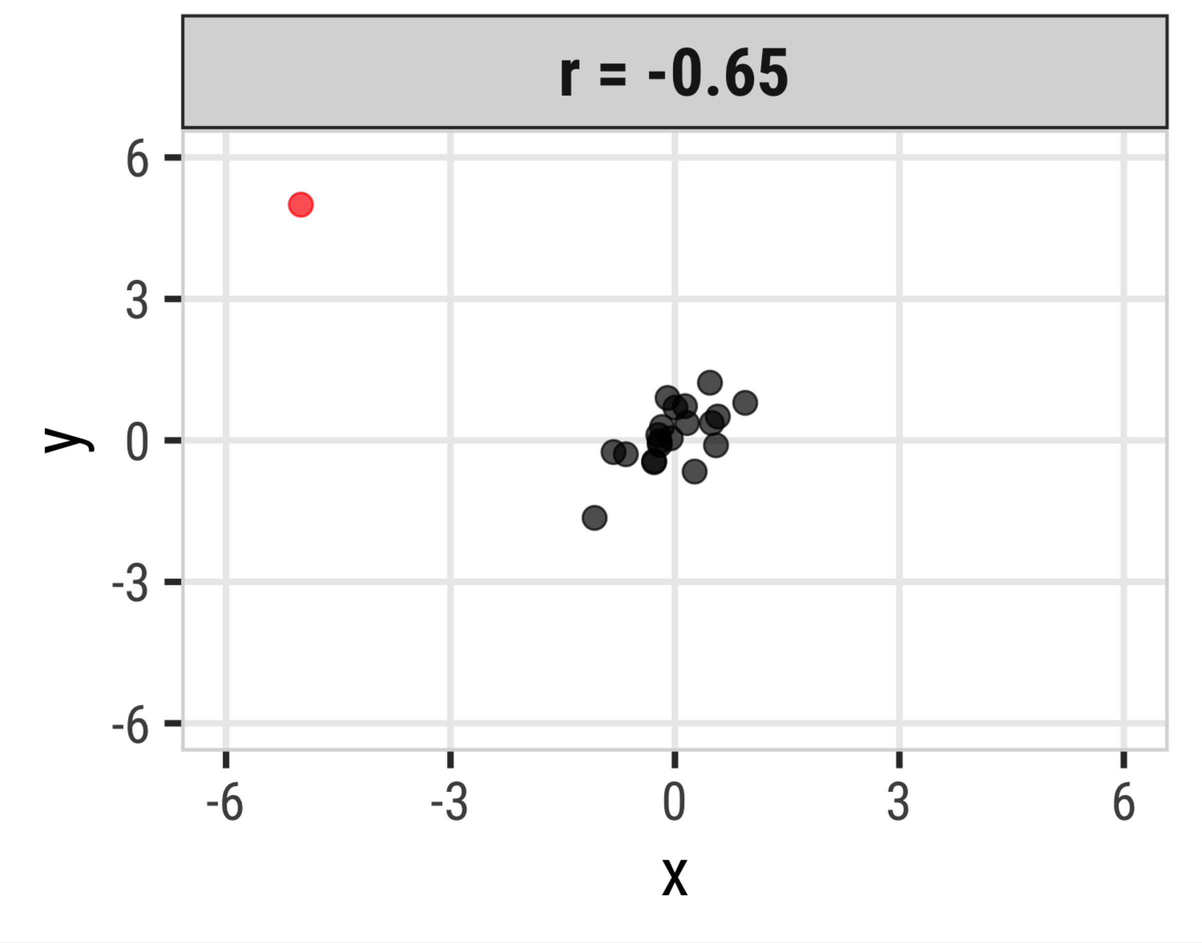


FROM "BEWARE SPURIOUS CORRELATIONS," JUNE 2015 © HBR.ORG

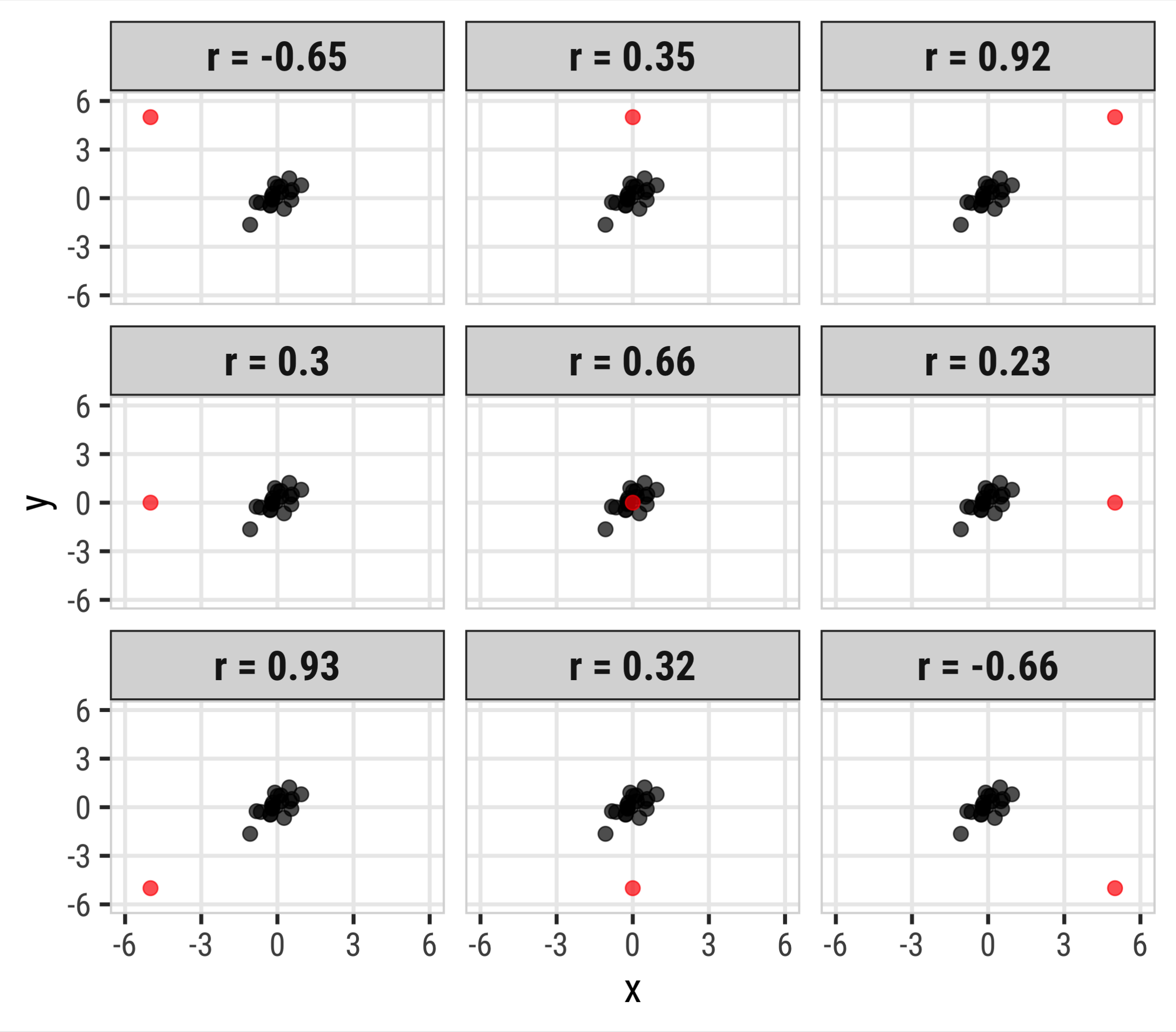
Outliers


r = 0.66
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 = -0.65
6
3
0
-3
-6
y
-6
-3
0
3
6
x

Pearson correlation is highly sensitive to outliers



Spearman's rank-order correlation

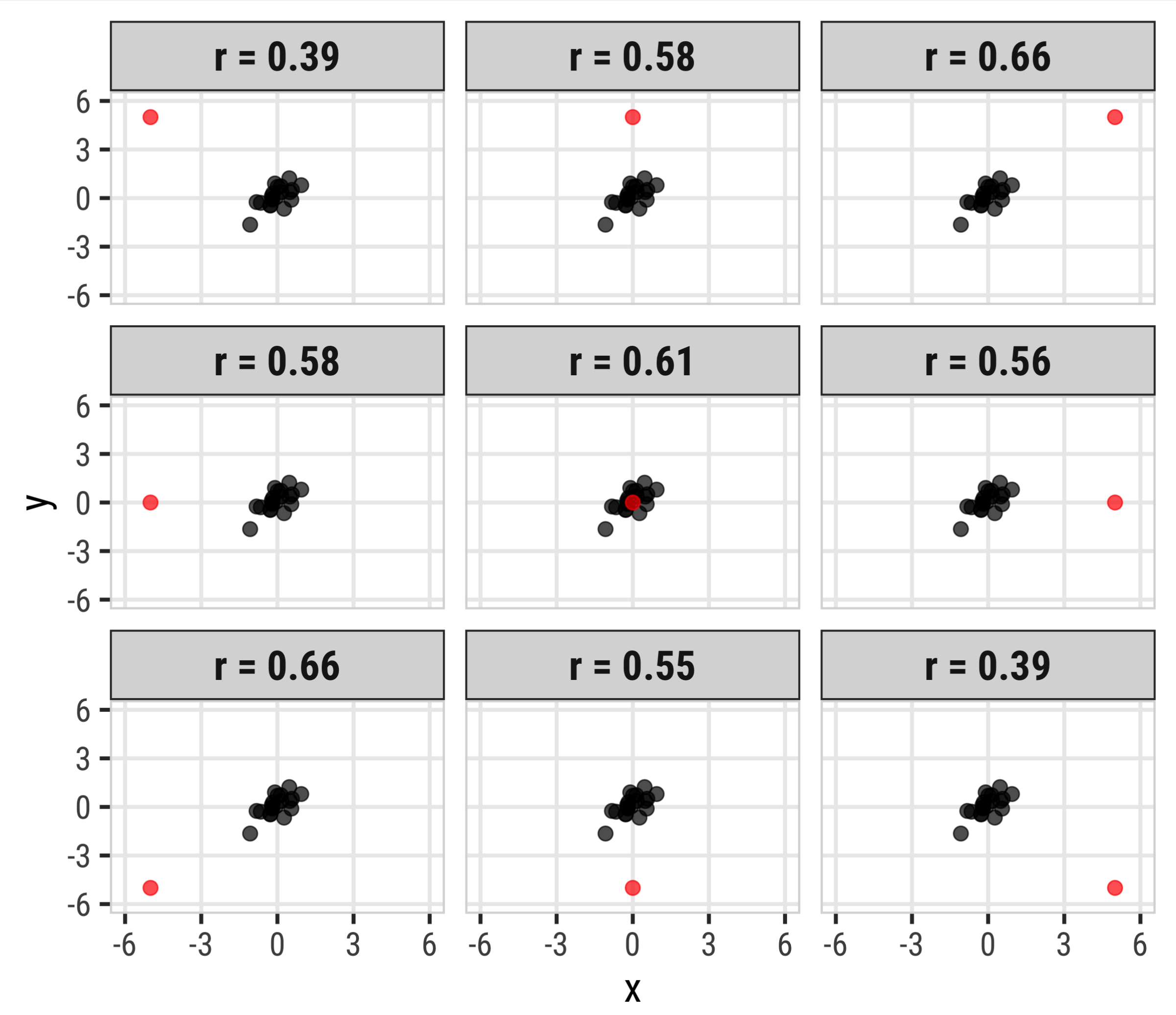
$$r = \frac{\mathrm{Cov}(x,y)}{\mathrm{sd}(x) * \mathrm{sd}(y)}$$

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

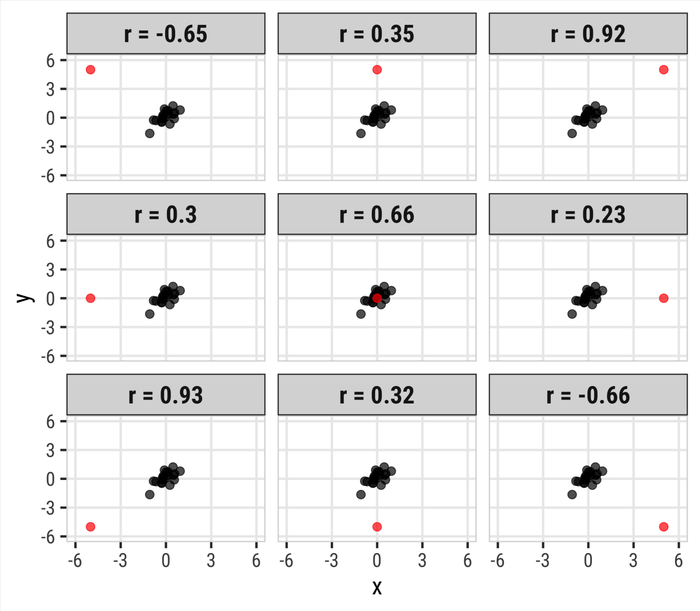
Assumptions:

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

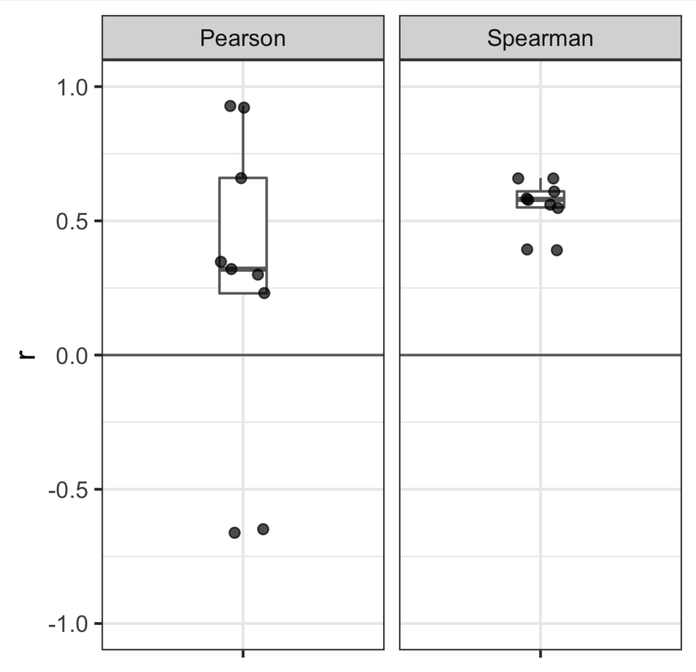
Spearman correlation more robust to outliers



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 5: *Correlation*

1. What is correlation?

2. Visualizing correlation

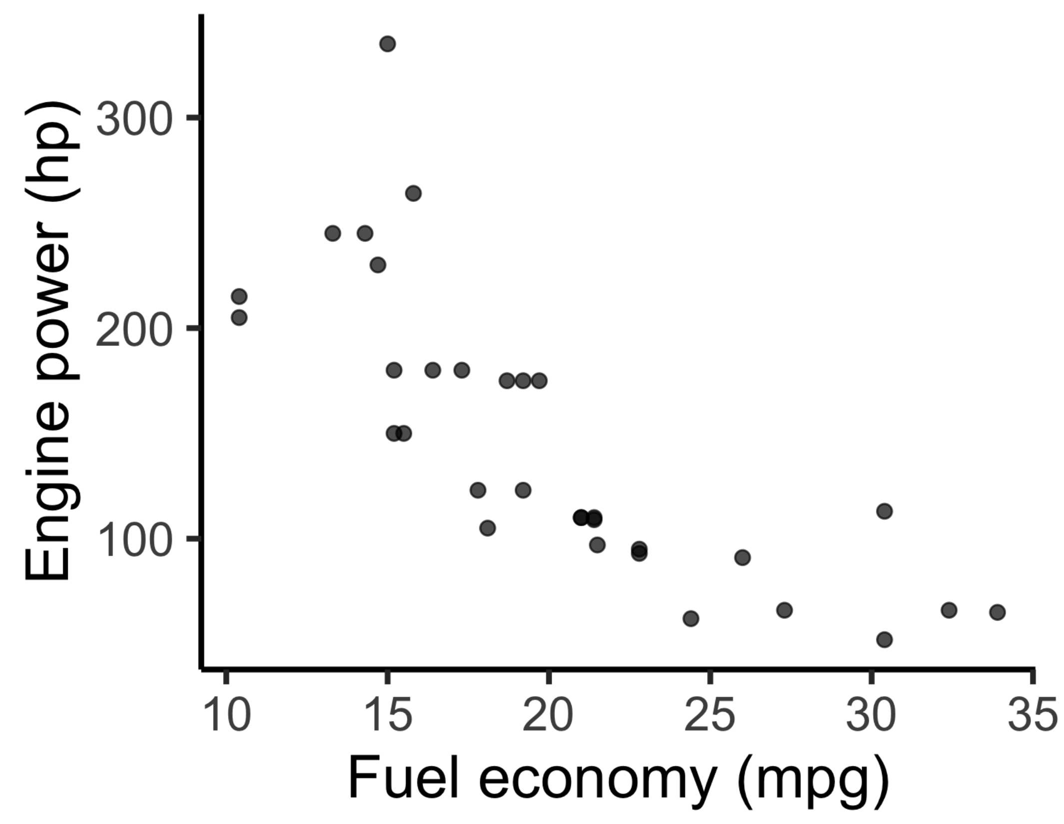
BREAK

3. Linear models

4. Visualizing linear models

Scatterplots: The correlation workhorse

```
scatterplot <- ggplot(mtcars) +
    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
```



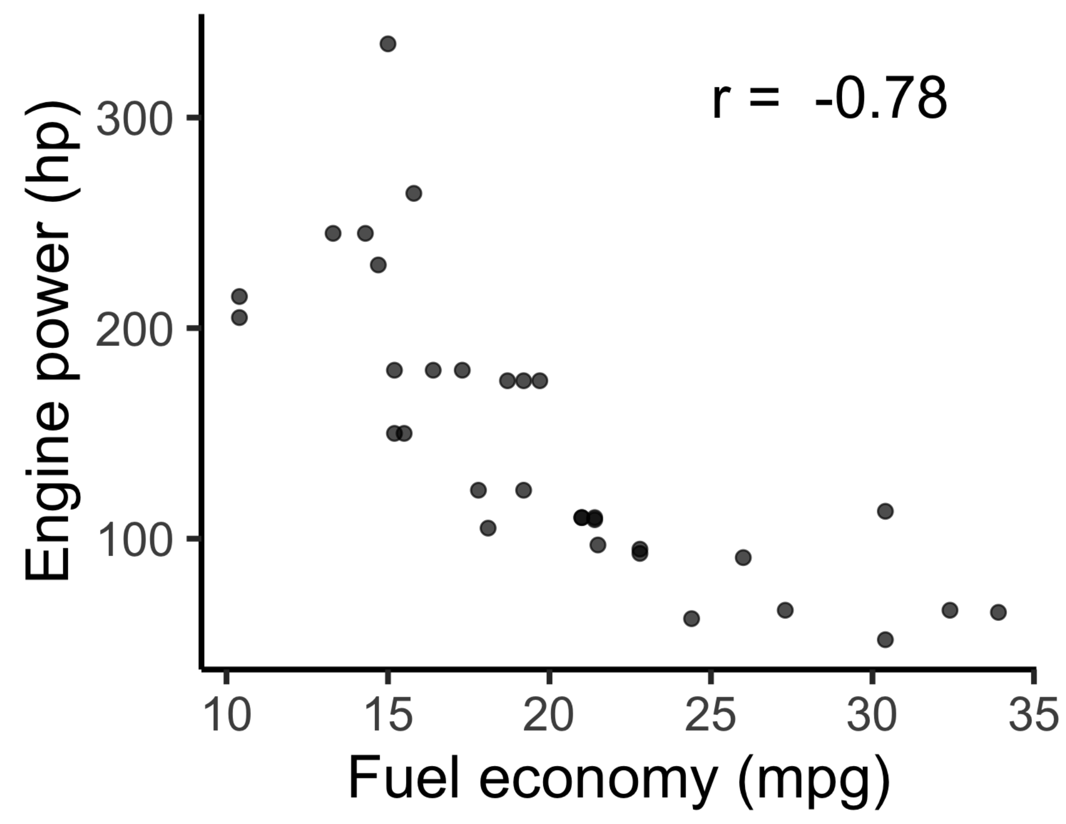
Adding a correlation label to a chart

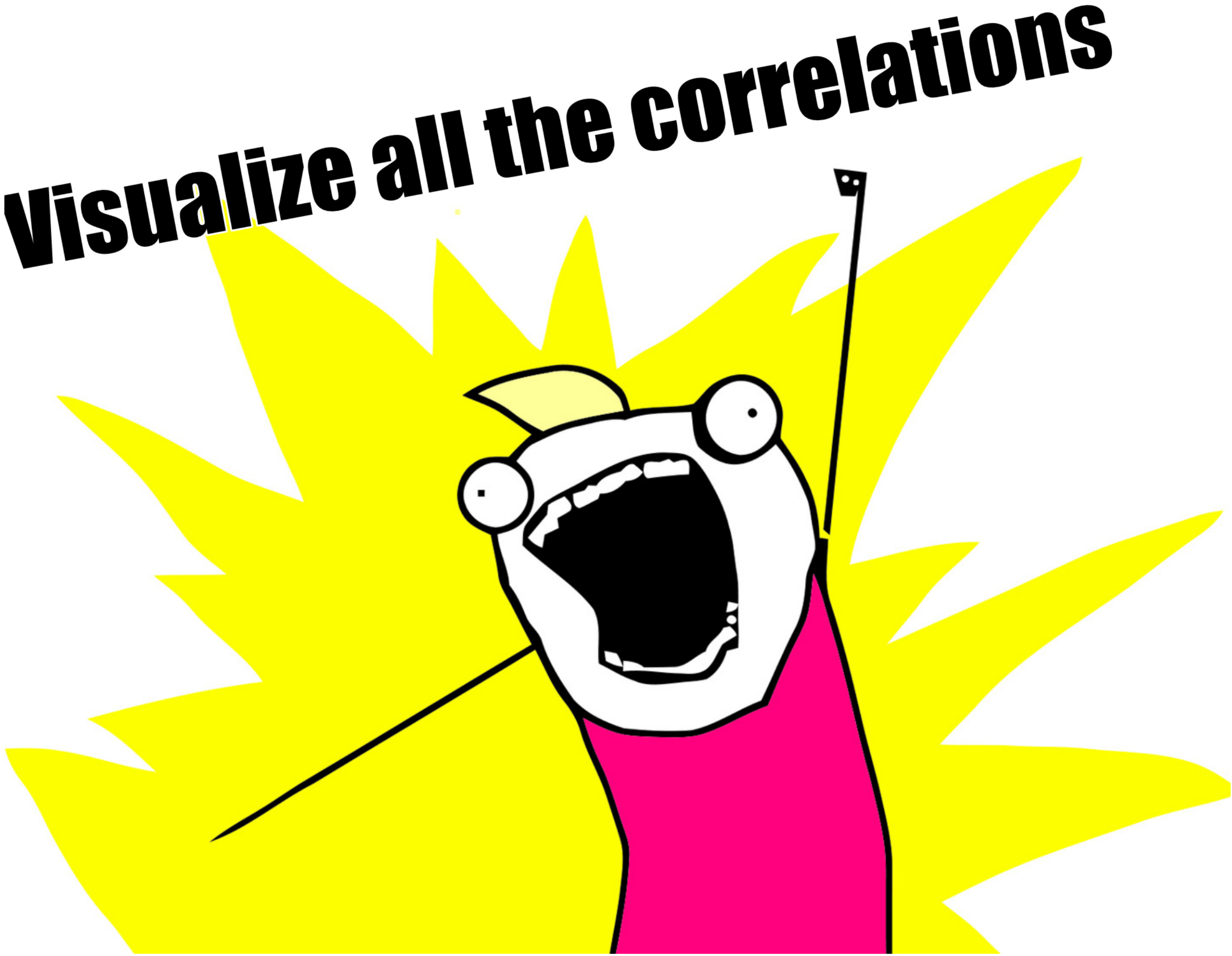
Make the correlation label

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

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

Add label to the chart with `annotate()`

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

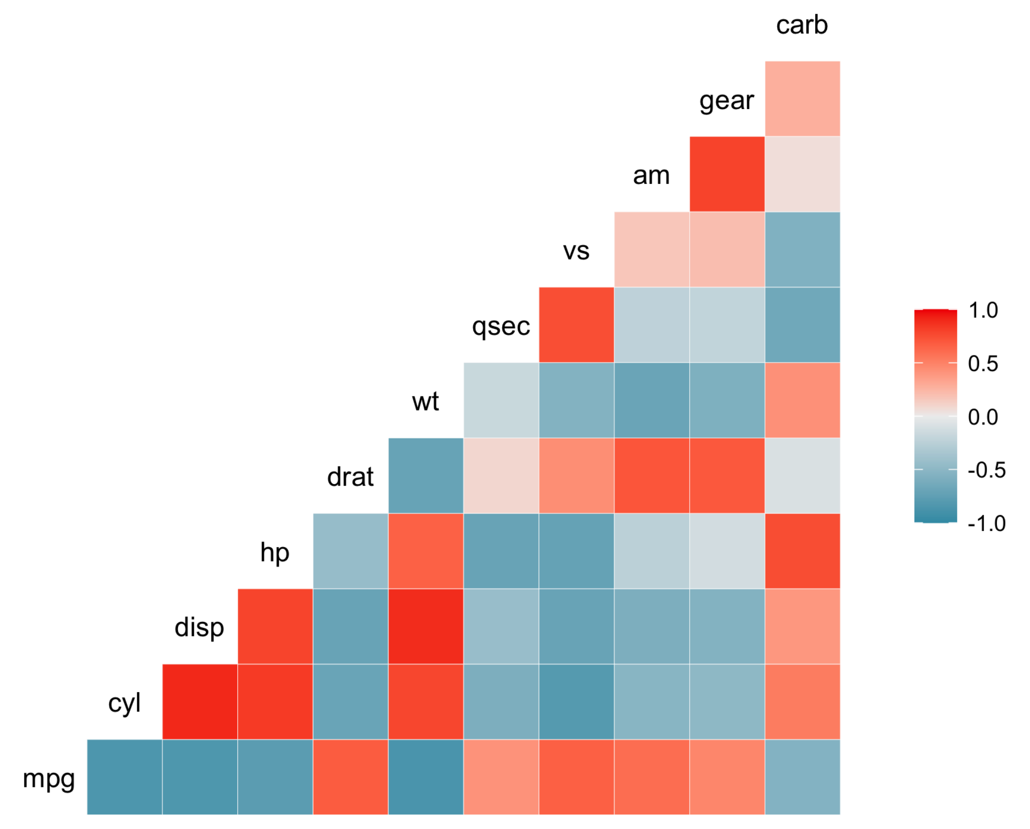



Visualize all the correlations

Visualize all the correlations: `ggcorr()`

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

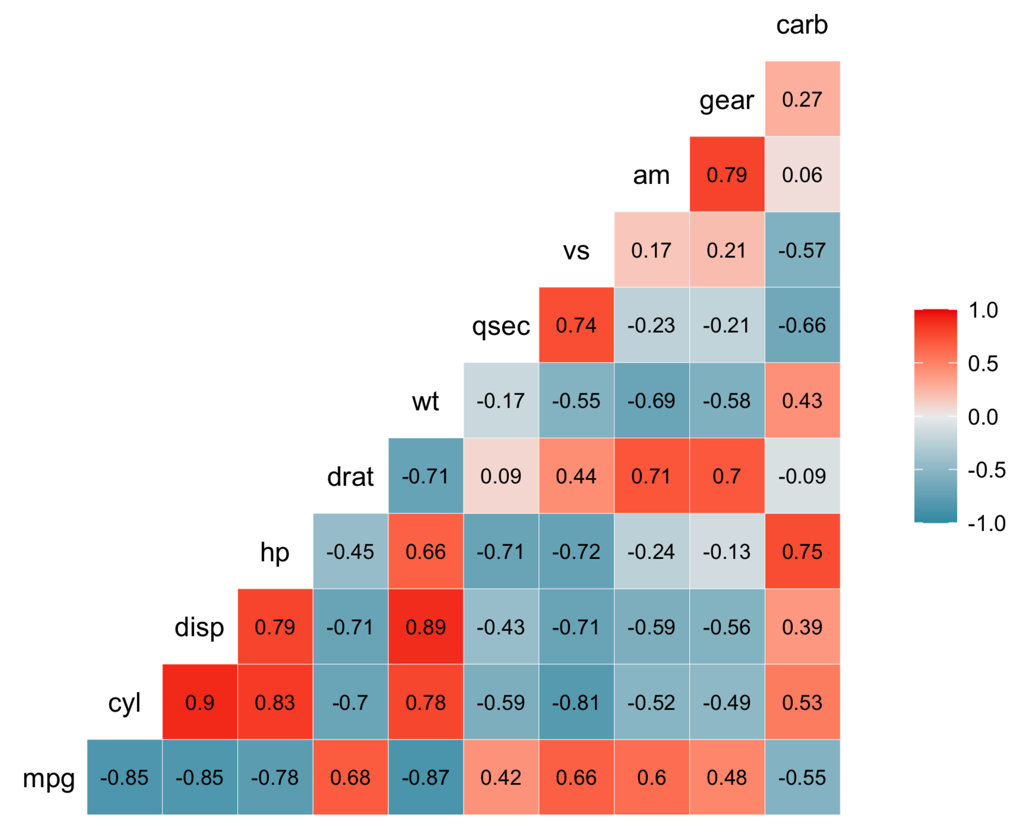
```
mtcars %>%
    ggcorr()
```



Visualizing correlations: `ggcorr()`

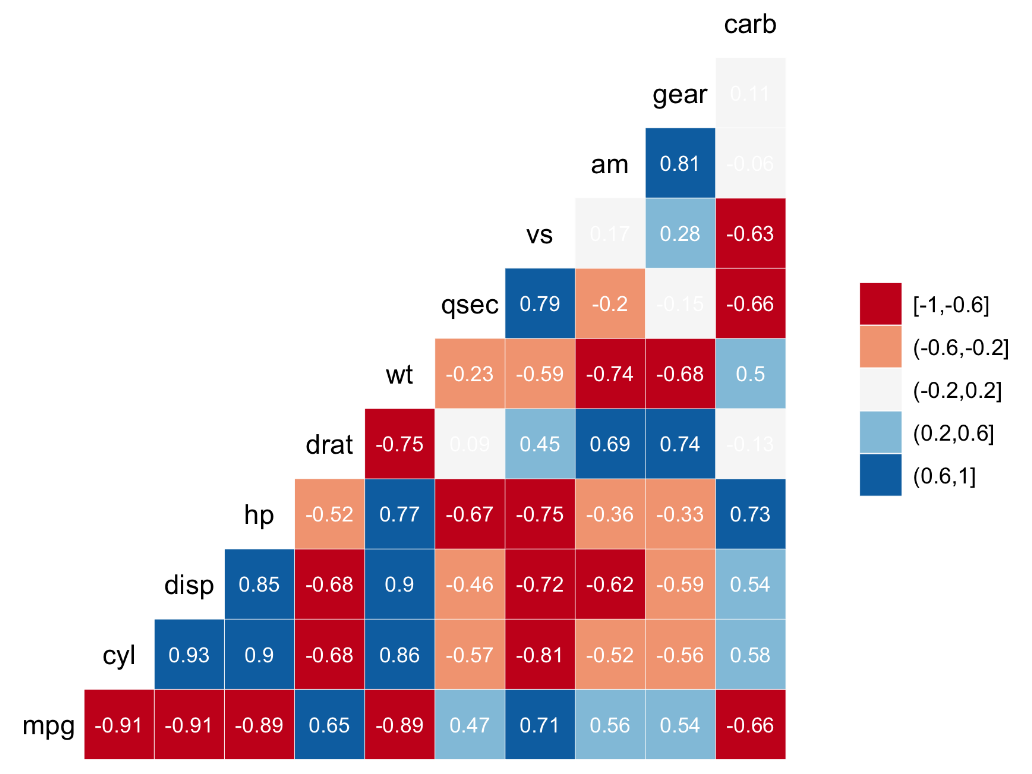
```r
library('GGally')
```

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



Visualizing correlations: `ggcorr()`

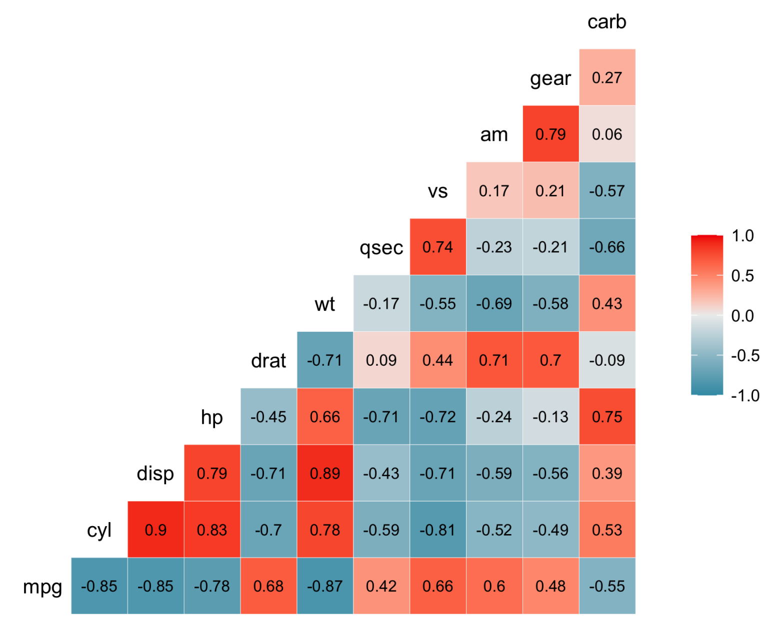
```r
ggcor_mtcars_final <- mtcars %>%
    ggcorr(label = TRUE,
           label_size = 3,
           label_round = 2,
           label_color = 'white',
           nbreaks = 5,
           palette = "RdBu")
```



	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
gear										0.11
am									0.81	-0.06
vs								0.17	0.28	-0.63
qsec							0.79	-0.2	-0.15	-0.66
wt						-0.23	-0.59	-0.74	-0.68	0.5
drat					-0.75	0.09	0.45	0.69	0.74	-0.13
hp				-0.52	0.77	-0.67	-0.75	-0.36	-0.33	0.73
disp			0.85	-0.68	0.9	-0.46	-0.72	-0.62	-0.59	0.54
cyl		0.93	0.9	-0.68	0.86	-0.57	-0.81	-0.52	-0.56	0.58
mpg	-0.91	-0.91	-0.89	0.65	-0.89	0.47	0.71	0.56	0.54	-0.66

Pearson

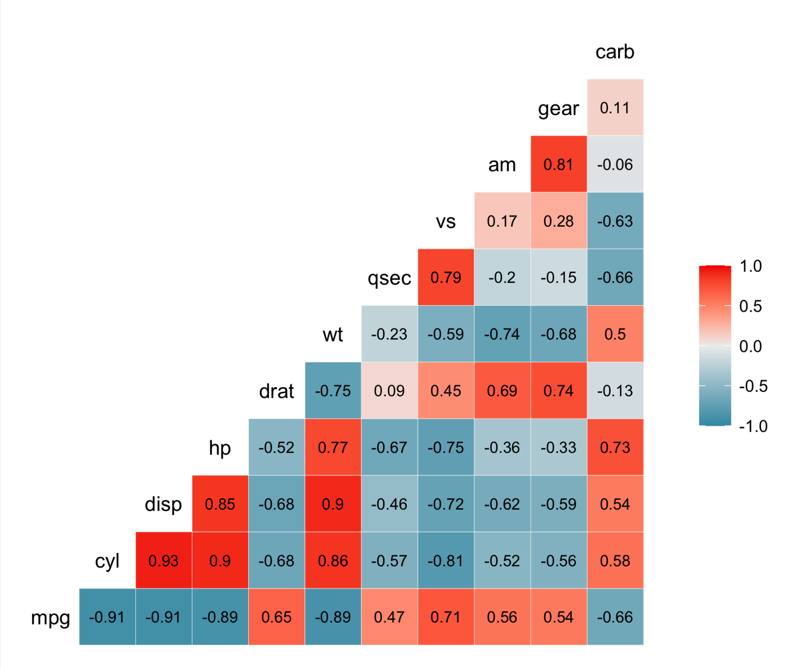
```
mtcars %>%
    ggcorr(label = TRUE,
           label_size = 3,
           label_round = 2,
           method = c("pairwise", "pearson"
```



	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
gear										0.27
am									0.79	0.06
vs								0.17	0.21	-0.57
qsec							0.74	-0.23	-0.21	-0.66
wt						-0.17	-0.55	-0.69	-0.58	0.43
drat					-0.71	0.09	0.44	0.71	0.7	-0.09
hp				-0.45	0.66	-0.71	-0.72	-0.24	-0.13	0.75
disp			0.79	-0.71	0.89	-0.43	-0.71	-0.59	-0.56	0.39
cyl		0.9	0.83	-0.7	0.78	-0.59	-0.81	-0.52	-0.49	0.53
mpg	-0.85	-0.85	-0.78	0.68	-0.87	0.42	0.66	0.6	0.48	-0.55

Spearman

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



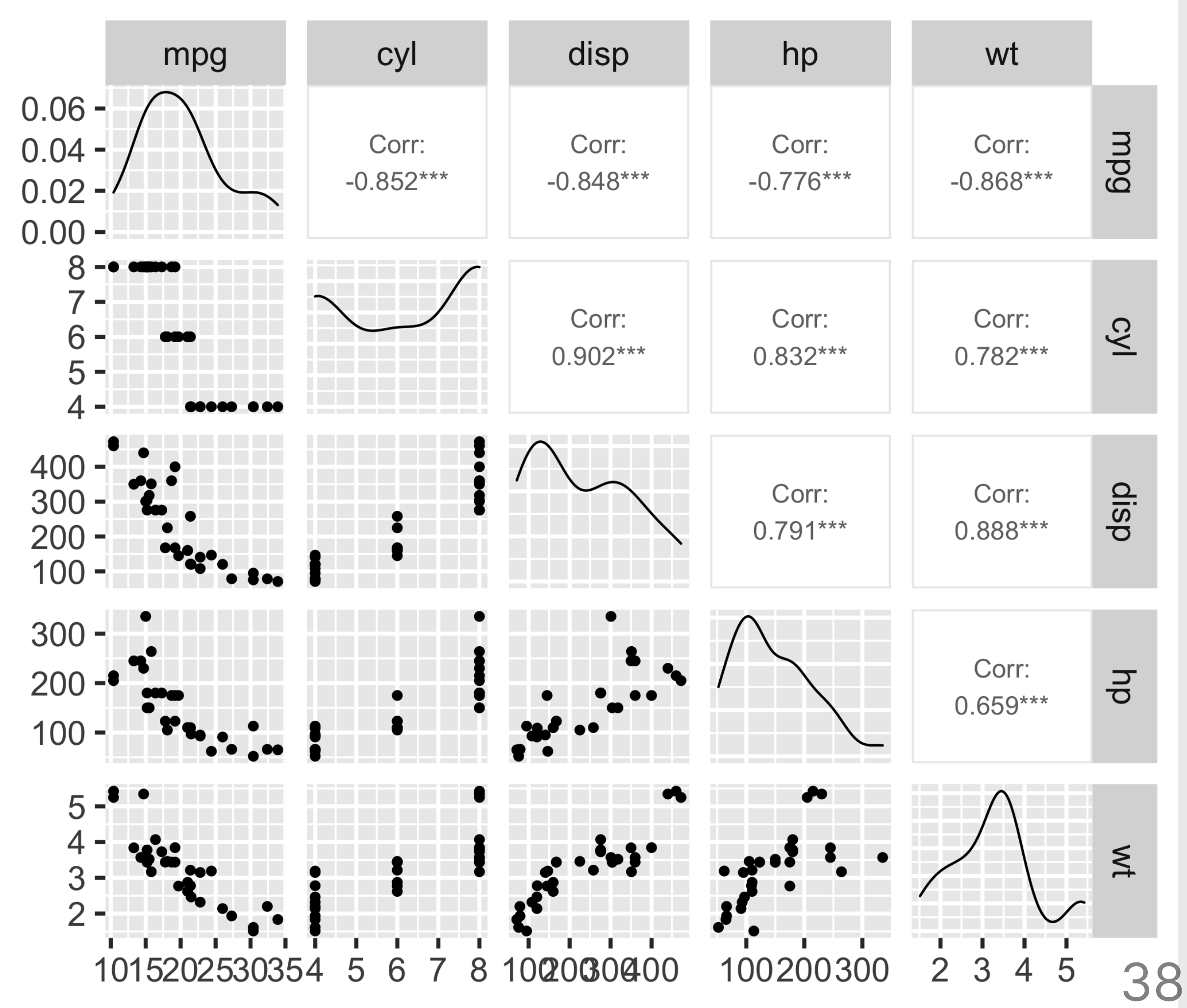
	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
gear										0.11
am									0.81	-0.06
vs								0.17	0.28	-0.63
qsec							0.79	-0.2	-0.15	-0.66
wt						-0.23	-0.59	-0.74	-0.68	0.5
drat					-0.75	0.09	0.45	0.69	0.74	-0.13
hp				-0.52	0.77	-0.67	-0.75	-0.36	-0.33	0.73
disp			0.85	-0.68	0.9	-0.46	-0.72	-0.62	-0.59	0.54
cyl		0.93	0.9	-0.68	0.86	-0.57	-0.81	-0.52	-0.56	0.58
mpg	-0.91	-0.91	-0.89	0.65	-0.89	0.47	0.71	0.56	0.54	-0.66

Correlograms: `ggpairs()`

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

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

- Look for linear relationships
- View distribution of each variable

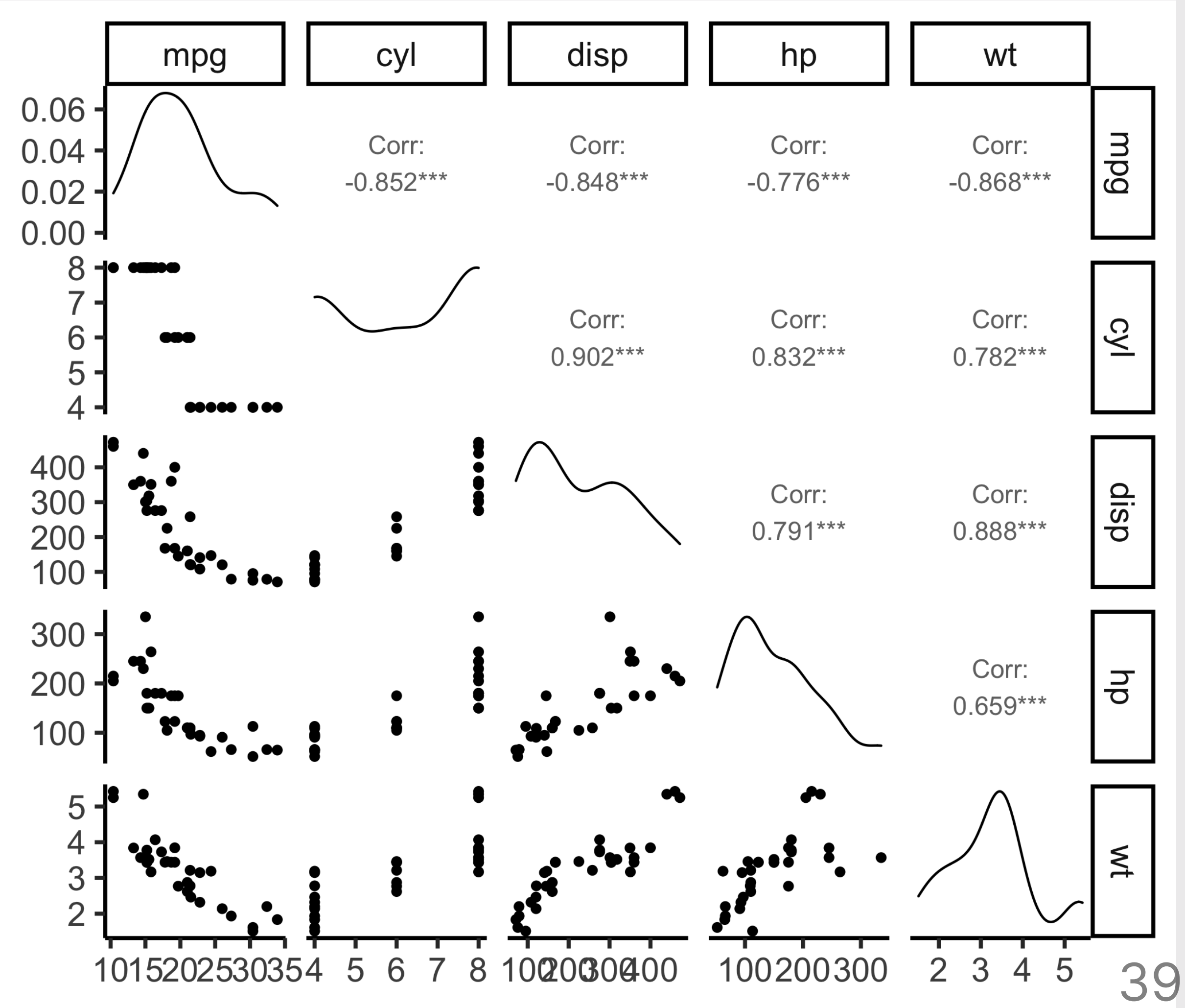


Correlograms: `ggpairs()`

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

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

- Look for linear relationships
- View distribution of each variable



Your turn

15:00

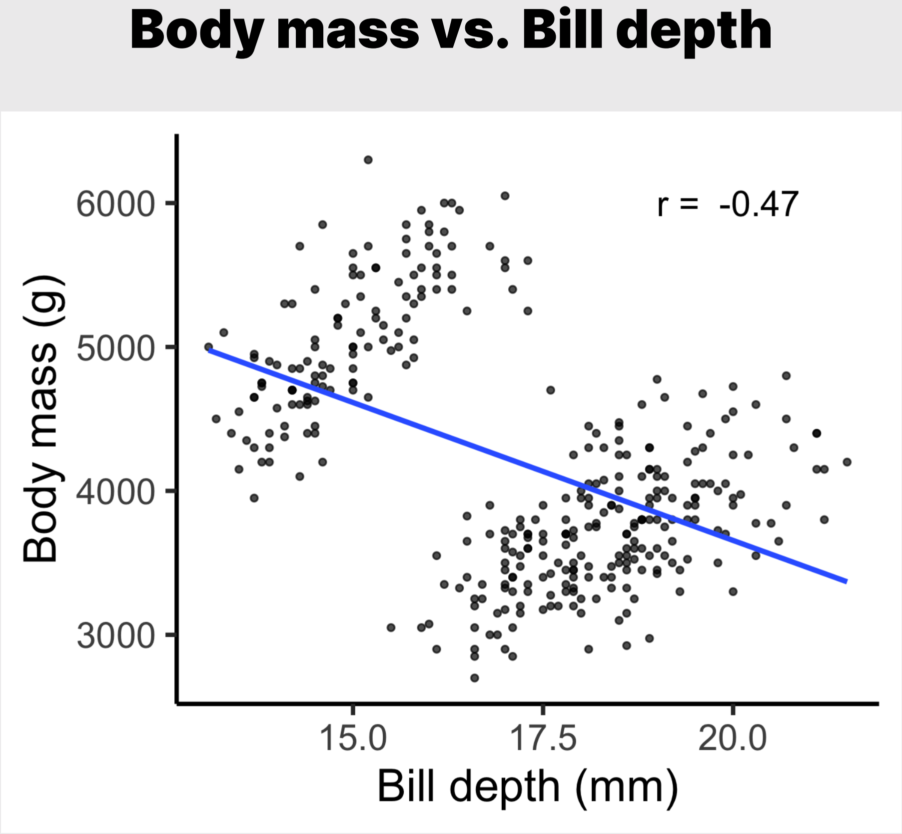
Using the `penguins` data frame:

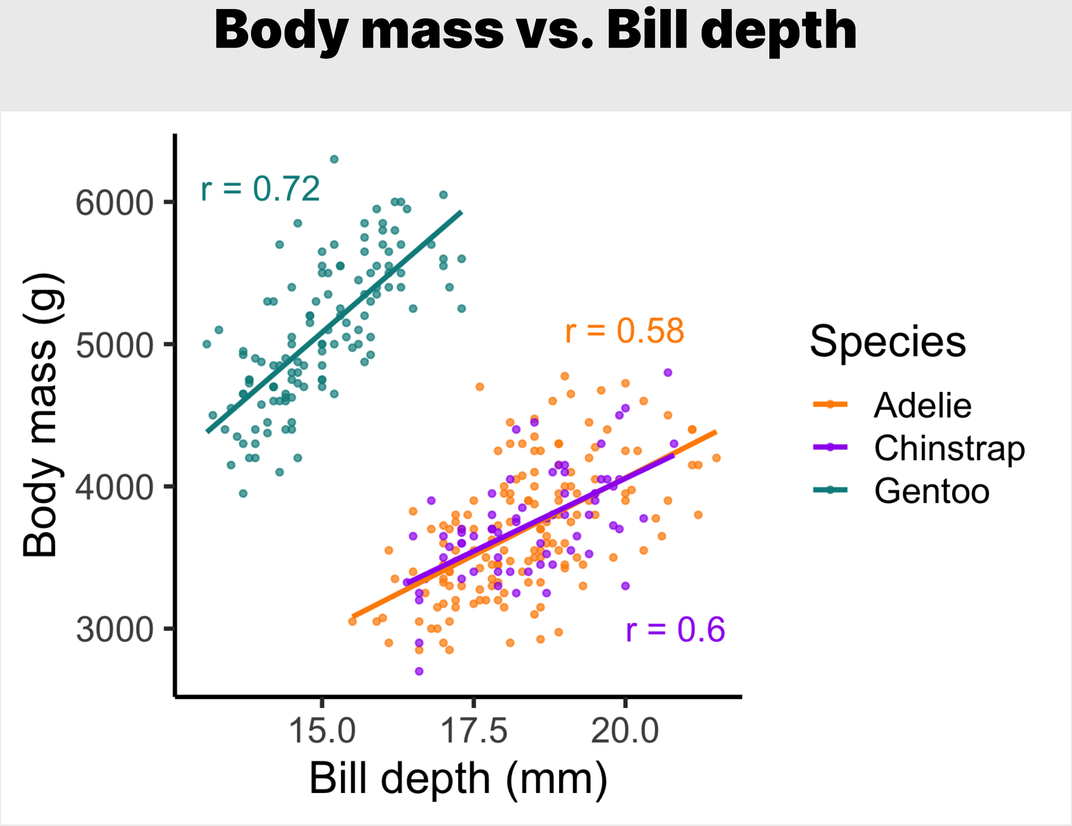
1. Find the two variables with the largest correlation in absolute value (i.e. closest to -1 or 1).
2. Create a scatter plot of those two variables.
3. Add an annotation for the Pearson correlation coefficient.

[palmerpenguins library]

Artwork by @allison_horst

Simpson's Paradox: when correlation betrays you

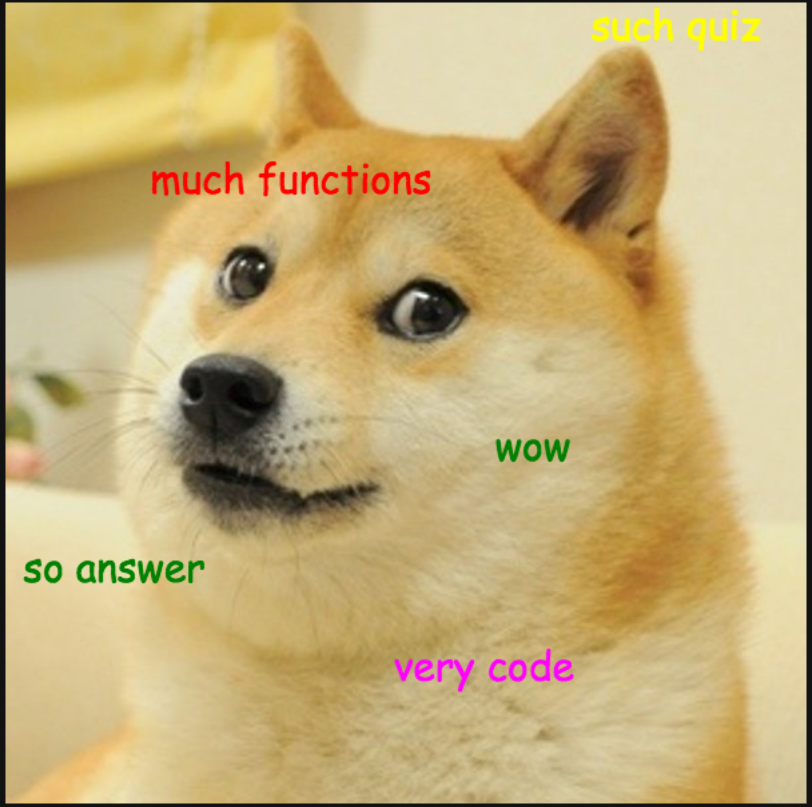




Quiz 2

Link is in the #class channel

10:00



Week 5: *Correlation*

1. What is correlation?

2. Visualizing correlation

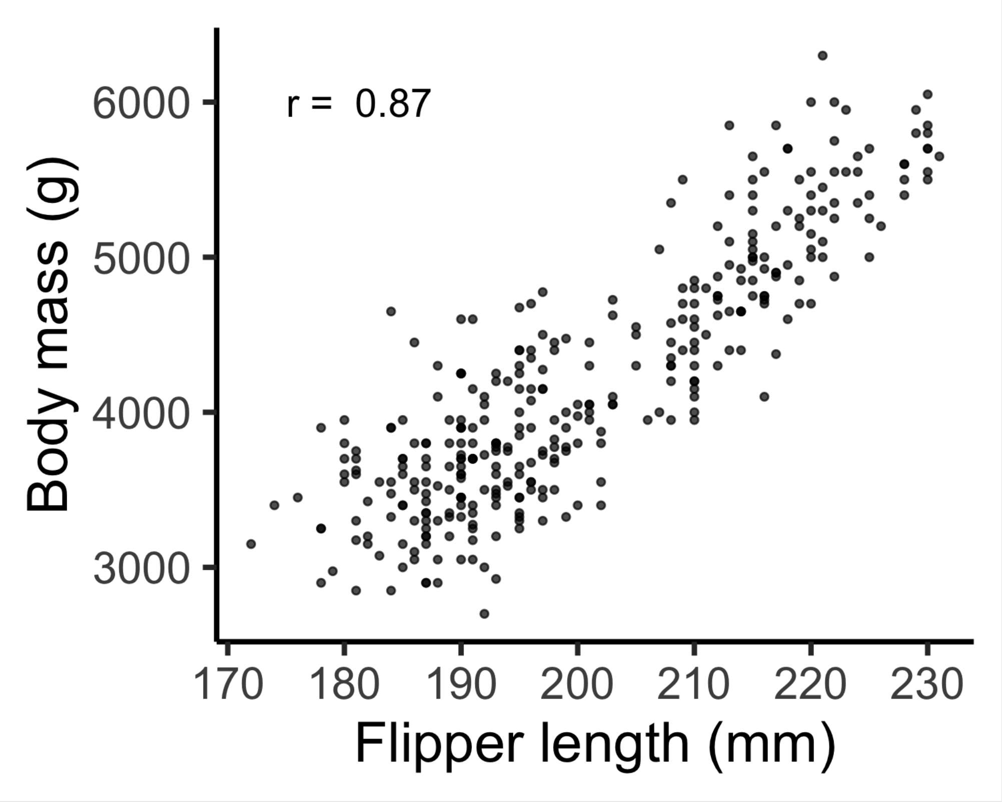
BREAK

3. Linear models

4. Visualizing linear models

Palmer Penguins

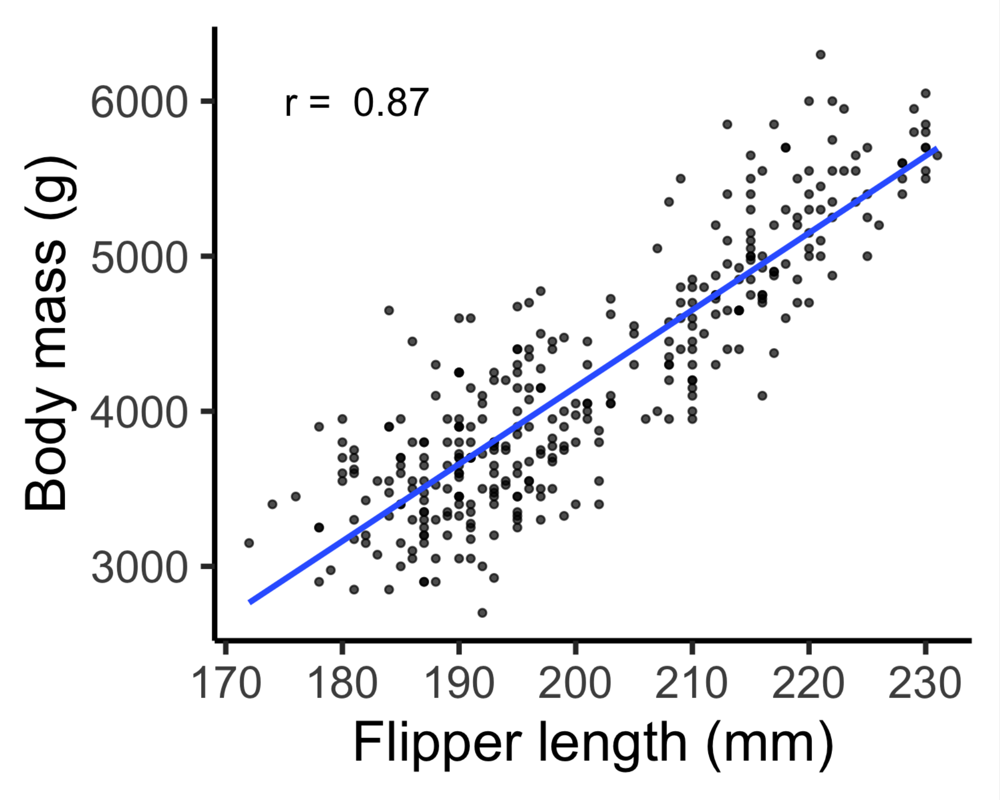
The correlation of 0.87 means that the body mass (g) explains about 75% of the variation in the flipper length (mm).



Palmer Penguins

The correlation of 0.87 means that the body mass (g) explains about 75% of the variation in the flipper length (mm).

Now let's fit a model to these points!

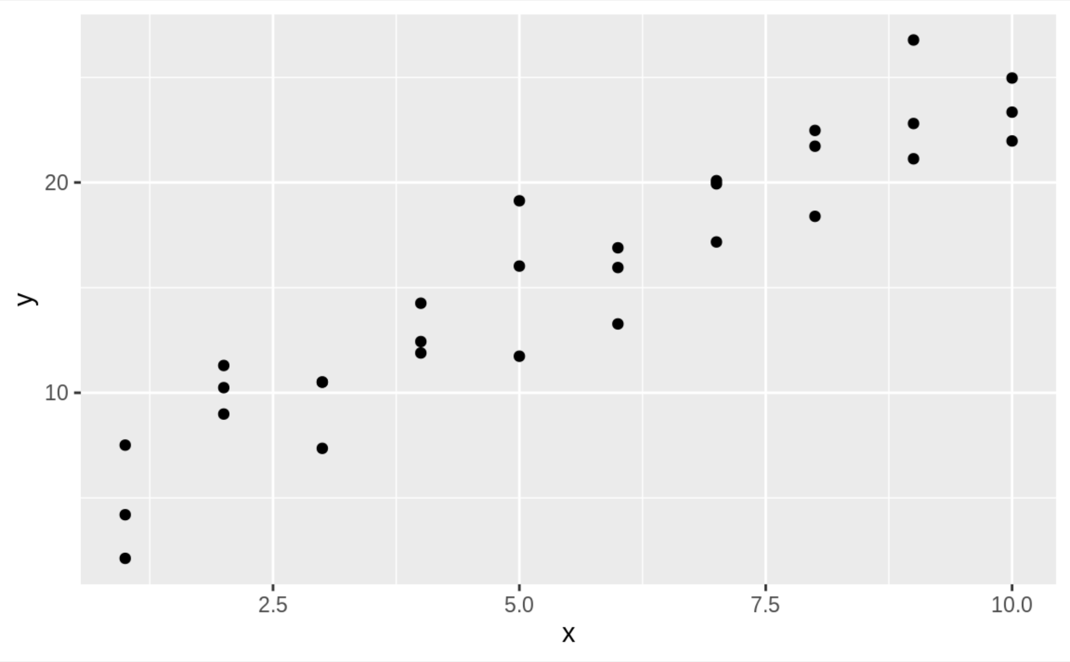


Modeling basics

Two parts to a model:

1. **Model family**: e.g., $y = ax + b$
2. **Fitted model**: e.g., $y = 3x + 7$

Here is some simulated data

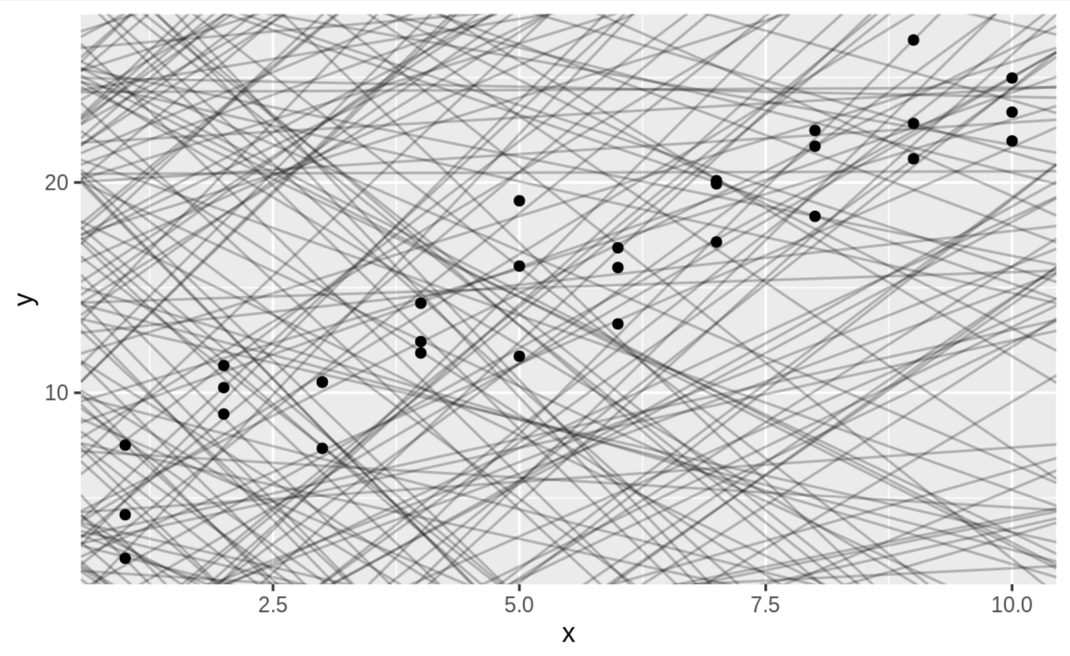


Modeling basics

Two parts to a model:

1. **Model family**: linear model: $y = ax + b$

There are an infinite number of possible models



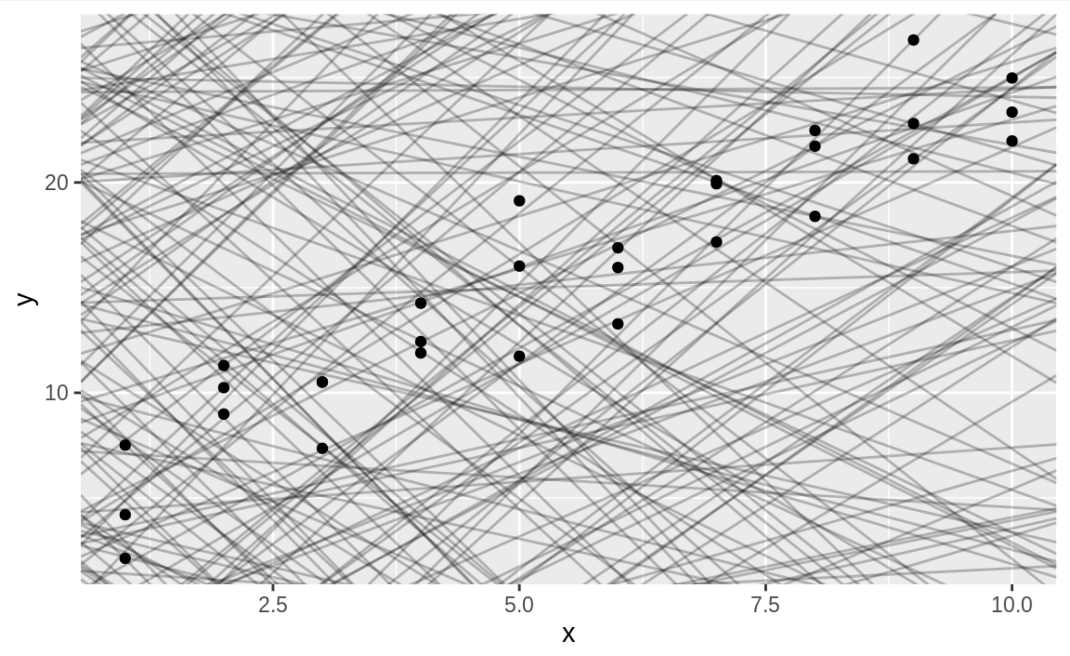
Modeling basics

Two parts to a model:

1. **Model family**: linear model: $y = ax + b$

1. **Fitted model**: How to choose the "best" a and b?

There are an infinite number of possible models



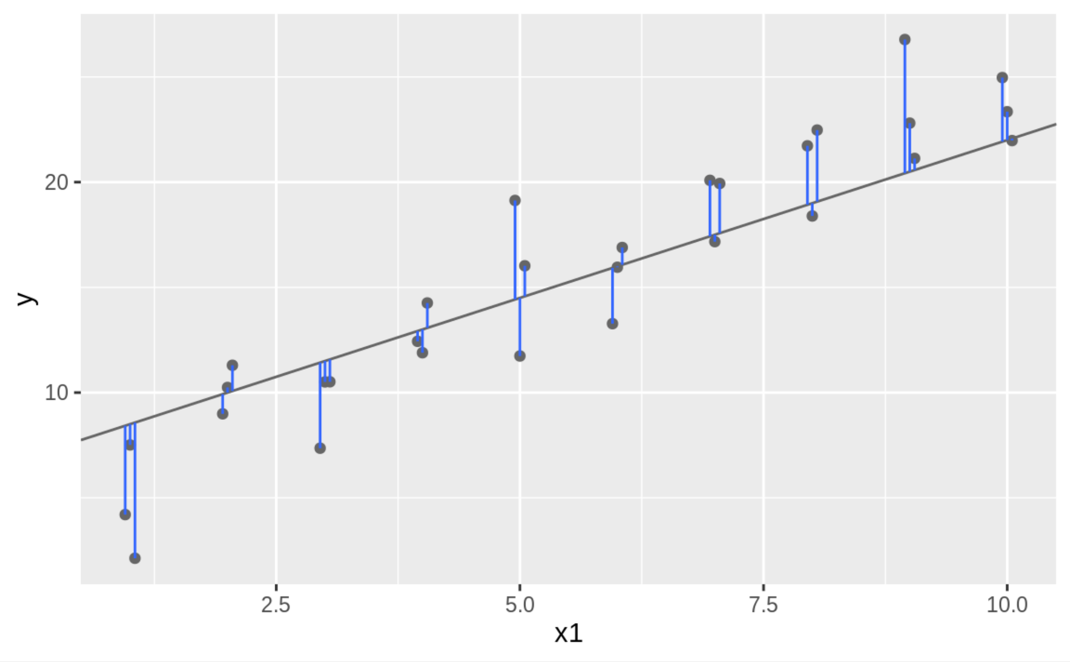
Modeling basics

Two parts to a model:

1. **Model family**: linear model:
   $y = ax + b$
2. **Fitted model**: How to choose the "best" a and b?

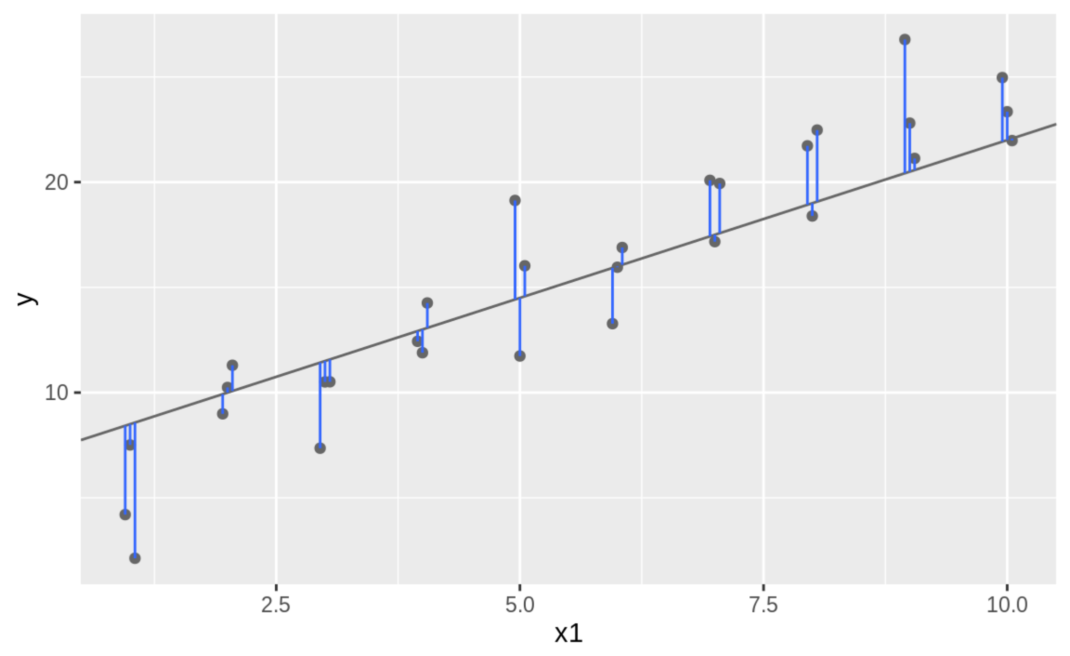
We need to come up with some measure of "distance" from the model to the data

Compute the **"residuals"**:
The distance between the model line and the data



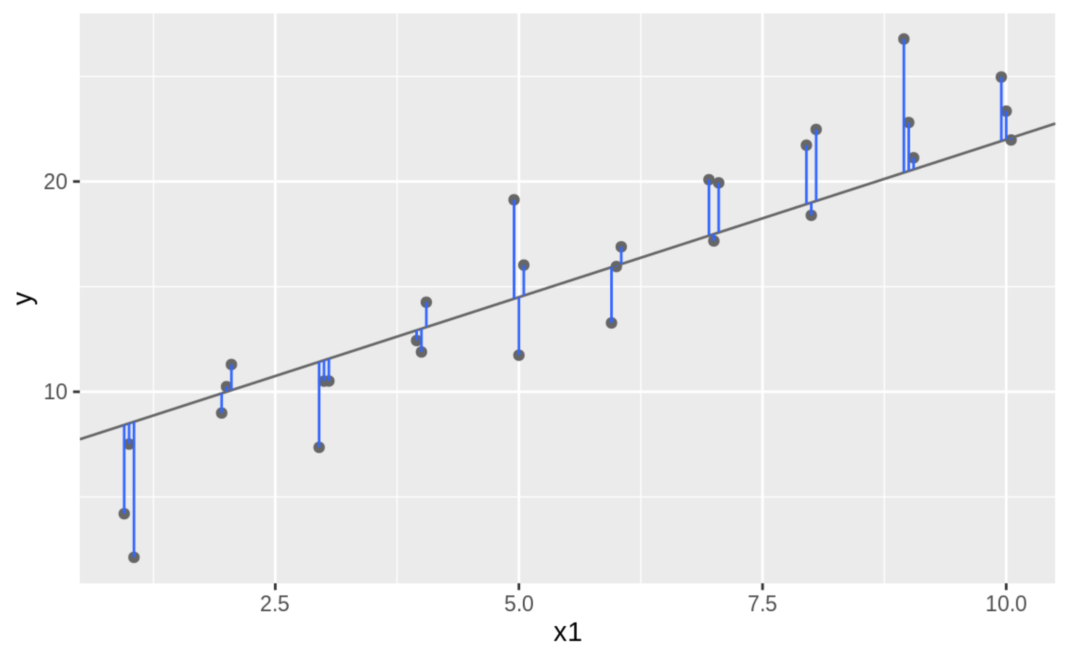
Residual: $y_i - y_i'$

Residual: The distance between the model line and the data



Sum of squared residuals: $\mathrm{SSR} = \sum_{i=1}^{n}(y_i - y_i')^2$

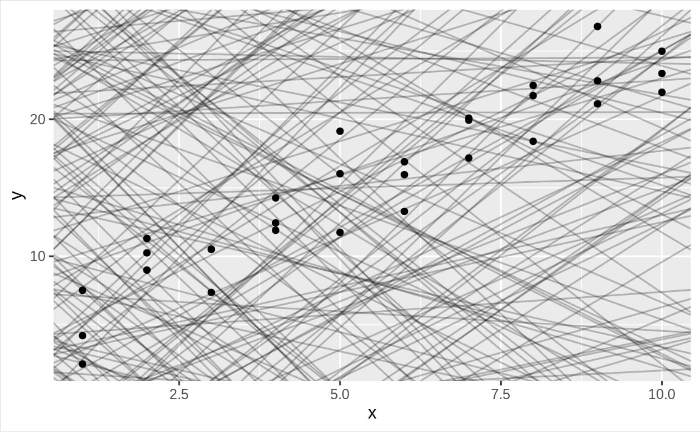
Residual: The distance between the model line and the data



Search algorithm

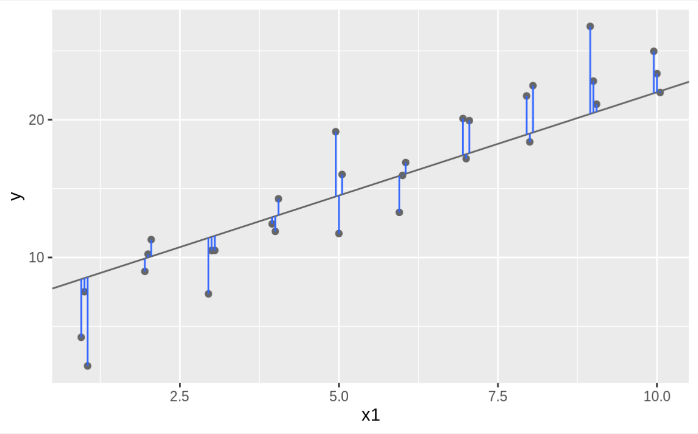
1): Pick a model (a and b):

$$y = ax + b$$

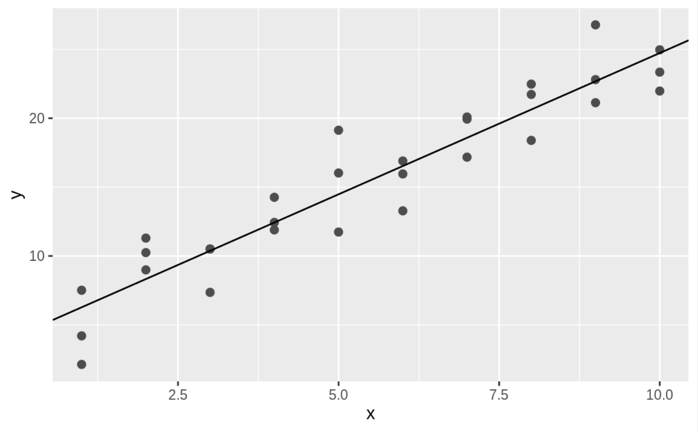


2): Compute the SSR:

$$\text{SSR} = \sum_{i=1}^{n}(y_i - y_i')^2$$



3): Repeat steps 1 & 2 until the smallest SSR is found



Fitting a linear model in R

```
model <- lm(formula = y ~ x, data = data)
```

Penguin data:

```
model <- lm(
  formula = body_mass_g ~ flipper_length_mm,
  data    = penguins)
```

Get coefficients (a & b in $y = ax + b$)

```
coef(model)
```

```
#>       (Intercept) flipper_length_mm
#>       -5780.83136          49.68557
```

Fitting a linear model in R

```
model <- lm(formula = y ~ x,
            data = data)
```

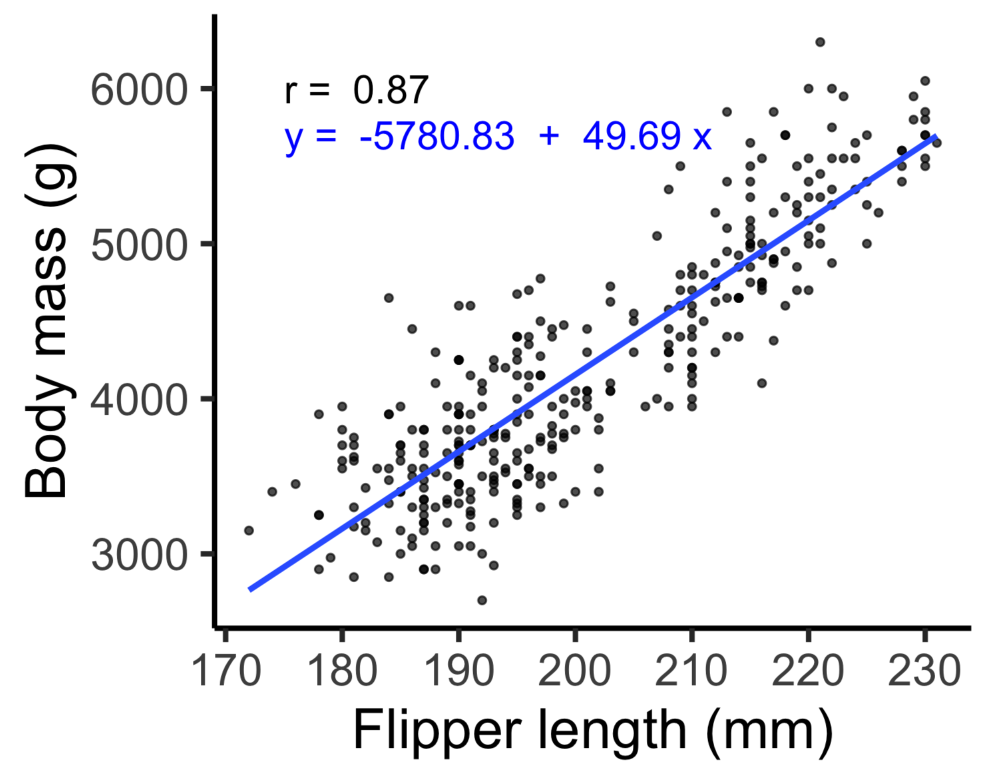
Penguin data:

```
model <- lm(
  formula = body_mass_g ~ flipper_length_mm,
  data    = penguins)
```

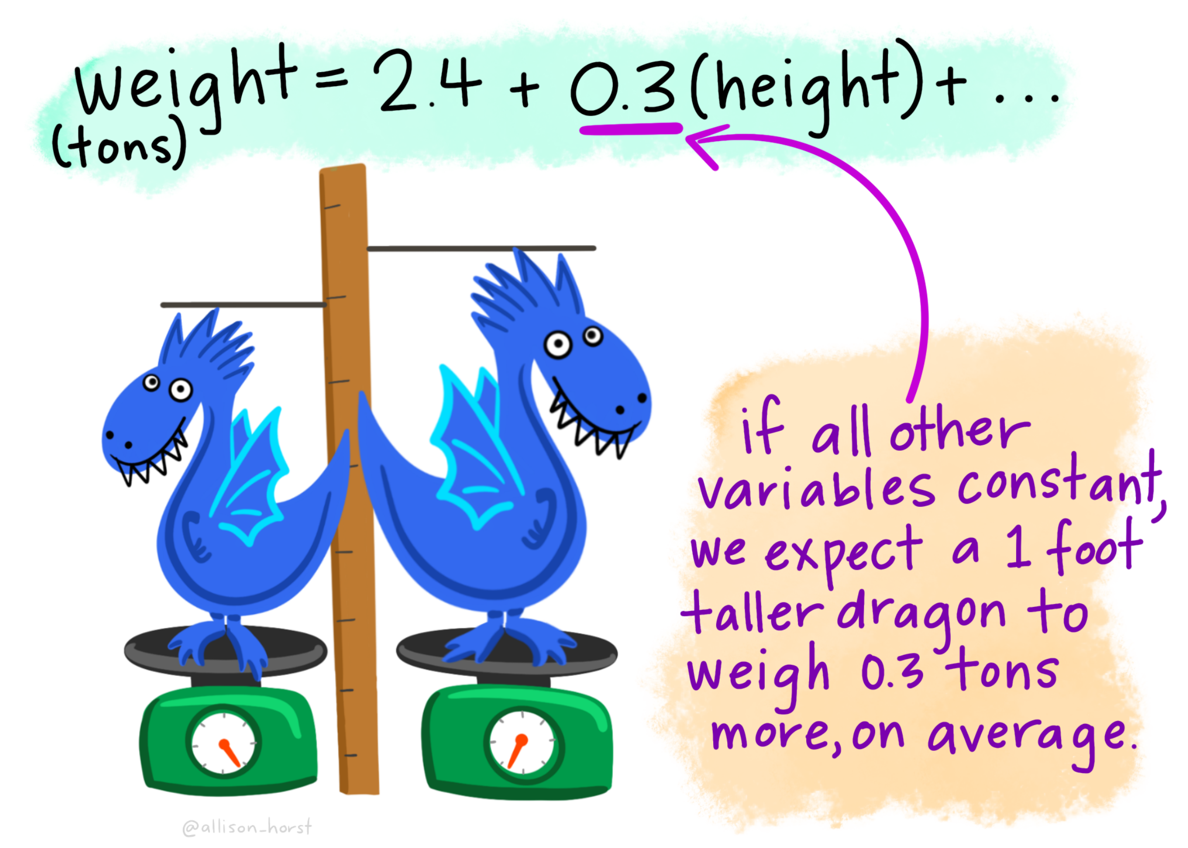
Get coefficients

```
coef(model)
```

```
#>       (Intercept) flipper_length_mm
#>       -5780.83136          49.68557
```



Interpreting results

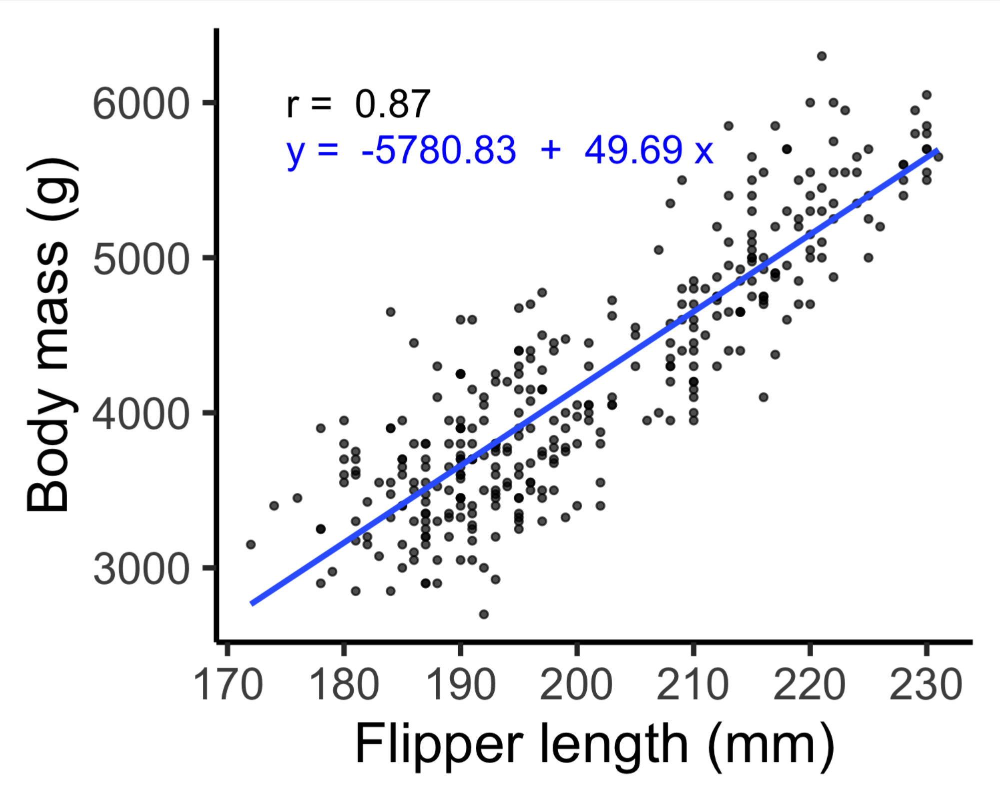


Artwork by @allison_horst

Example write up for Penguin data

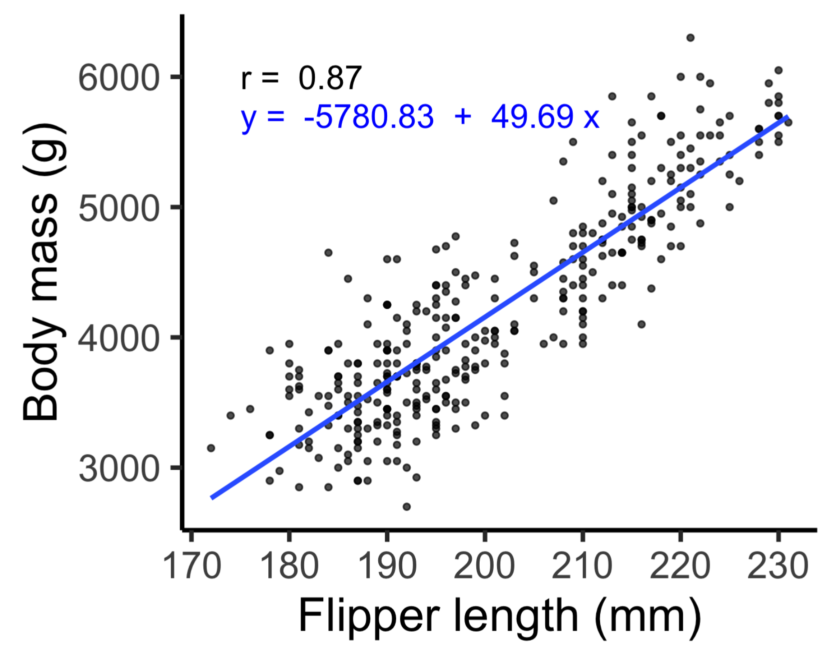
The correlation between flipper length (mm) and body mass (g) is **0.87**. Therefore, **~75%** of the variance in body mass is explained by flipper length.

The slope of the best fitting regression line indicates that body mass increased by **49.7 g** as flipper length increased by one mm.

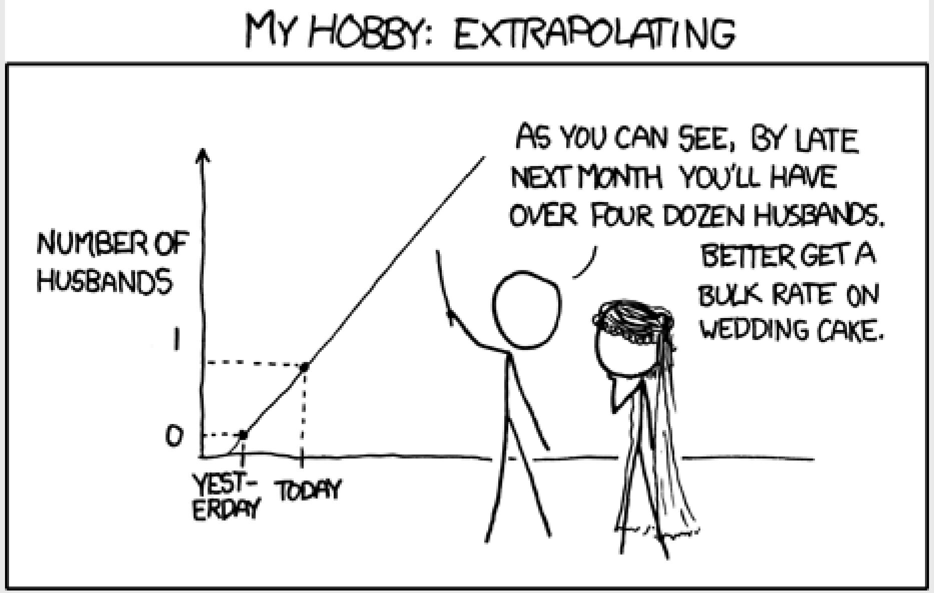


Making predictions

Interpolation is OK: You may predict values of y for values of x that were not observed but are within the range of the observed values of x.

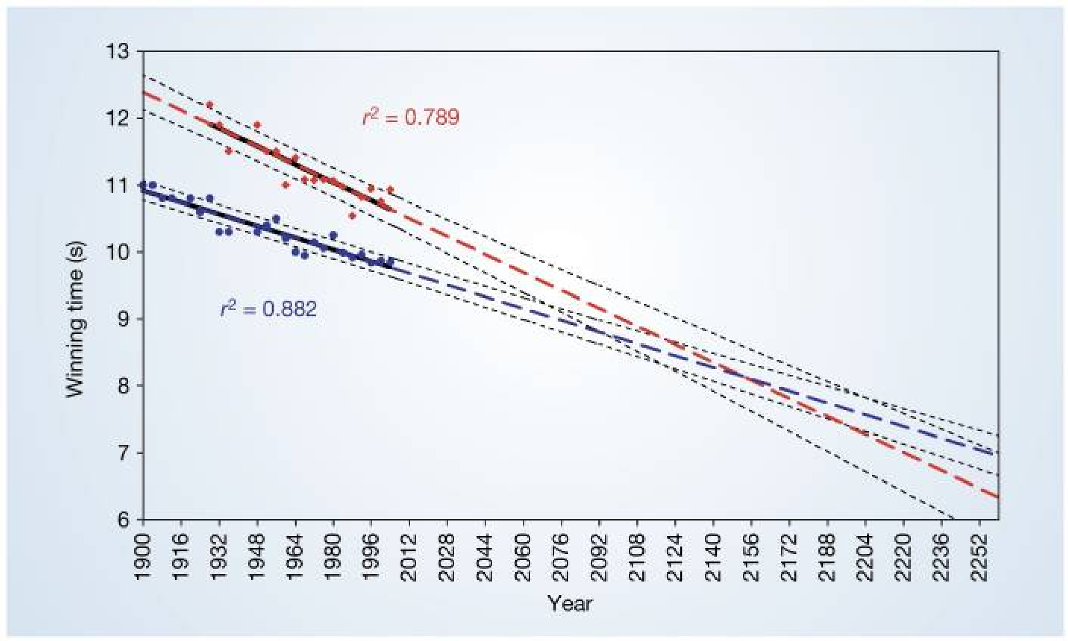


Extrapolation is DANGEROUS: You generally should NOT predict values of y using values of x that are outside the observed range of x.



Repeat: Extrapolation is **DANGEROUS**

"Extrapolation of these trends to the 2008 Olympiad indicates that the women's 100-metre race could be won in a time of 10.57±0.232 seconds and the men's event in 9.73±0.144 seconds. **Should these trends continue, the projections will intersect at the 2156 Olympics, when — for the first time ever — the winning women's 100-metre sprint time of 8.079 seconds will be lower than that of the men's winning time of 8.098 seconds (Fig. 1).**"



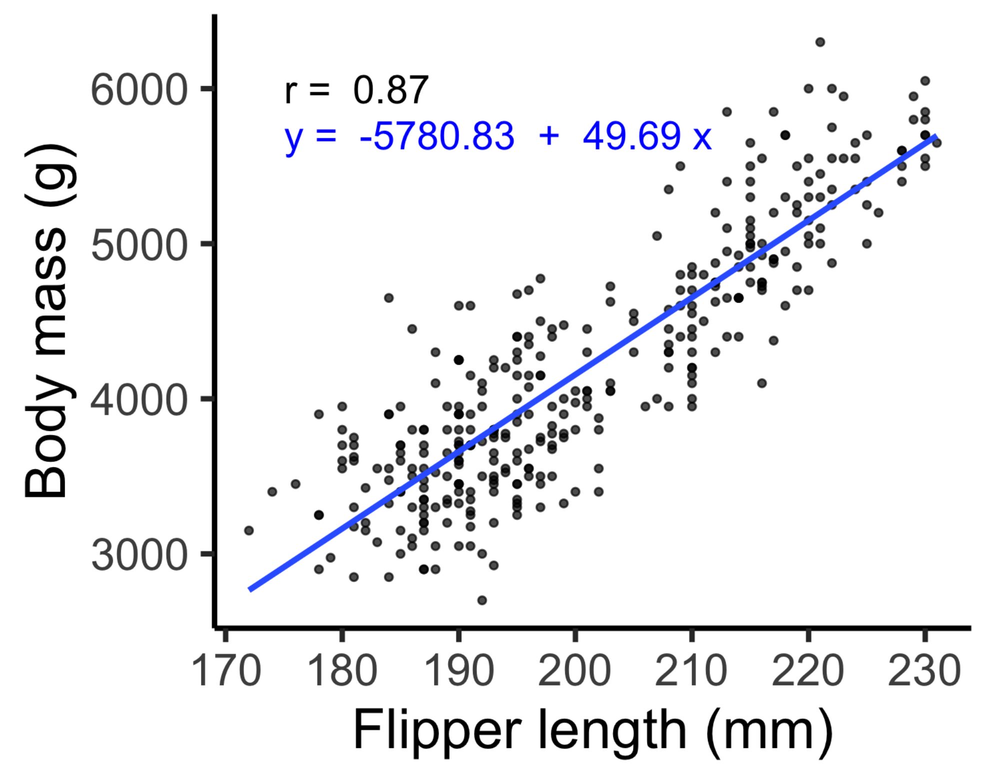
Tatem, A. J., Guerra, C. A., Atkinson, P. M., & Hay, S. I. (2004). Momentous sprint at the 2156 Olympics? *Nature*, 431(7008), 525-525. View online

Symantics

These all mean the same thing:

- "Use X to predict Y"
- "Regress Y *on* X"
- "Regression of Y *on* X"

```
model <- lm(formula = y ~ x,
            data = data)
```



Symantics

```
model <- lm(formula = y ~ x,
            data = data)
```

Y: Dependent variable

- Outcome variable
- Response variable
- Regressand
- Left-hand variable

X: Independent variable

- Predictor variable
- Explanatory variable
- Regressor
- Right-hand variable

Week 5: *Correlation*

1. What is correlation?

2. Visualizing correlation

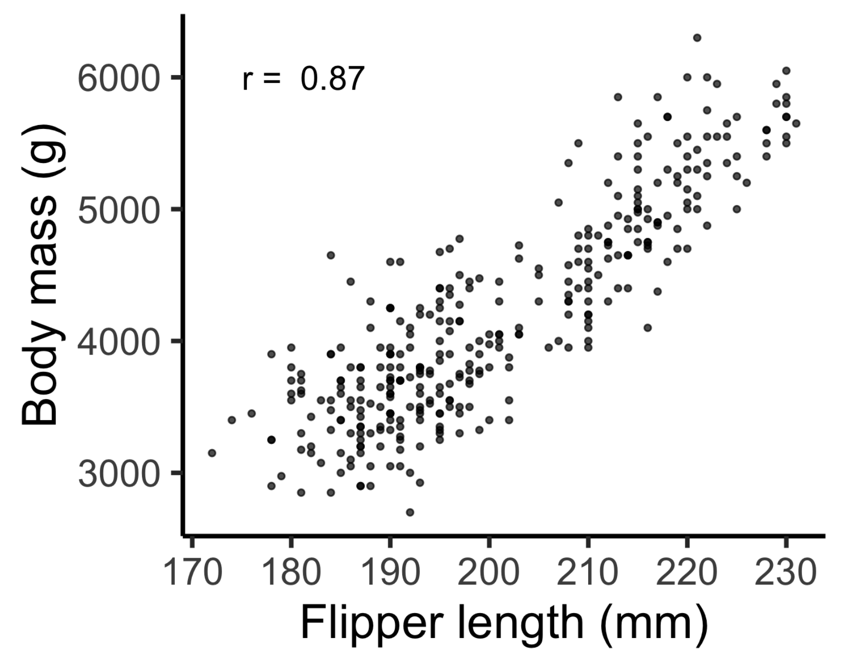
BREAK

3. Linear models

4. Visualizing linear models

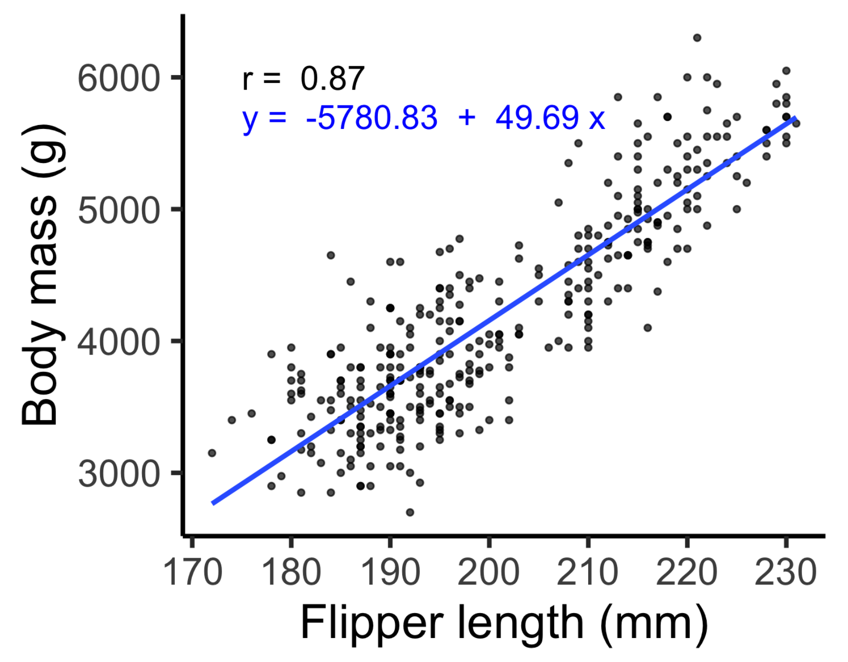
Adding the correlation annotation

```r
# Make the correlation label
corr <- cor(
    penguins$body_mass_g,
    penguins$flipper_length_mm,
    method = 'pearson',
    use = "complete.obs"
)

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

# Make the chart!
penguins %>%
    ggplot(aes(x = flipper_length_mm, y = body_mass_
    geom_point(size = 1, alpha = 0.7) +
    theme_classic(base_size = 20) +
    labs(x = "Flipper length (mm)",
         y = "Body mass (g)") +
    annotate(
        geom = 'text', x = 175, y = 6000,
        label = corrLabel,
        hjust = 0, size = 5)
```



Add model

```
# Make model label
model <- lm(
  formula = body_mass_g ~ flipper_length_mm,
  data    = penguins)
coefs <- round(coef(model), 2)
modelLabel <- paste('y = ', coefs[1], ' + ', coefs[2

# Make the chart!
penguins %>%
    ggplot(aes(x = flipper_length_mm, y = body_mass_
    geom_point(size = 1, alpha = 0.7) +
    geom_smooth(method = 'lm', se = FALSE) +
    theme_classic(base_size = 20) +
    labs(x = "Flipper length (mm)",
         y = "Body mass (g)") +
    annotate(geom = 'text', x = 175, y = 6000,
             label = corrLabel,
             hjust = 0, size = 5) +
    annotate(
        geom = 'text', x = 175, y = 5700,
        label = modelLabel, color = "blue",
        hjust = 0, size = 5)
```



Your turn

15:00

Using the `msleep` data frame:

1. Create a scatter plot of `brainwt` versus `bodywt`.
2. Include an annotation for the Pearson correlation coefficient.
3. Include an annotation for the best fit line.

Bonus: Compare your results to a log-linear relationship by converting the x and y variables to the log of x and y, like this:

```
model <- lm(log(x) ~ log(y), data = msleep)
```

You can also convert your plot to log axes by adding these layers:

```
plot +
    scale_x_log10() +
    scale_y_log10()
```