

# 431 Class 24

Thomas E. Love, Ph.D.

2022-12-08

# Today's Agenda

1. Graphical and Numerical Summaries of Data
2. It's Just a Linear Model
3. 432 preview: A `tidymodels` example with interaction
4. Takeaways from 431

# Today's Packages

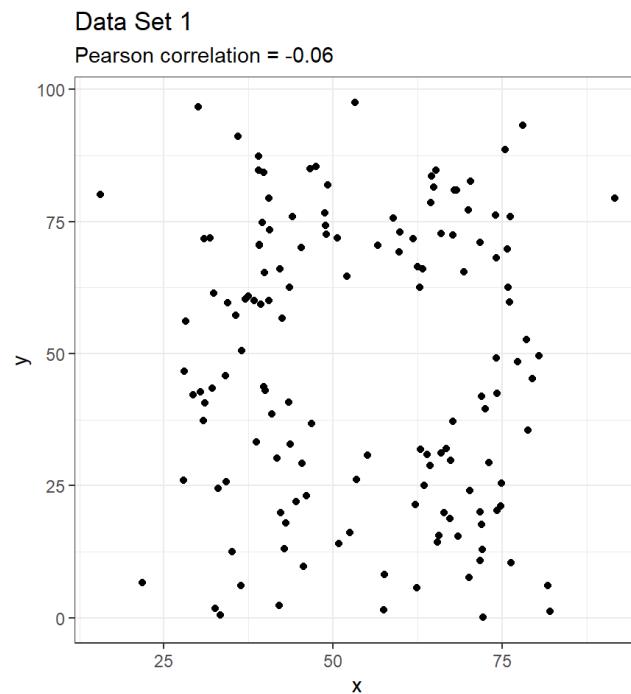
```
1 library(knitr); library(kableExtra)
2 library(janitor); library(mosaic)
3 library(equatiomatic); library(patchwork)
4 library(broom)
5 library(tidyverse)
6
7 opts_chunk$set(comment=NA)
8 options(dplyr.summarise.inform = FALSE)
9 theme_set(theme_bw())
```

and a couple of secrets, hidden for now.

# Visualizing Data

# New Data Set 1

response	n	missing	mean	sd
y	142	0	47.83	26.94
x	142	0	54.27	16.77



# 13 Data Sets in the df tibble:

	set	n	mean_x	sd_x	mean_y	sd_y	corr_xy
1	1	142	54.27	16.77	47.83	26.94	-0.06
2	2	142	54.27	16.77	47.83	26.94	-0.07
3	3	142	54.27	16.76	47.84	26.93	-0.07
4	4	142	54.26	16.77	47.83	26.94	-0.06
5	5	142	54.26	16.77	47.84	26.93	-0.06
6	6	142	54.26	16.77	47.83	26.94	-0.06
7	7	142	54.27	16.77	47.84	26.94	-0.07
8	8	142	54.27	16.77	47.84	26.94	-0.07
9	9	142	54.27	16.77	47.83	26.94	-0.07
10	10	142	54.27	16.77	47.84	26.93	-0.06
11	11	142	54.27	16.77	47.84	26.94	-0.07
12	12	142	54.27	16.77	47.83	26.94	-0.07
13	13	142	54.26	16.77	47.84	26.93	-0.07

# New Data: Model for Set 1

```

1 set_1 <- lm(y ~ x, data = df |> filter(set == 1))
2
3 tidy(set_1, conf.int = T, conf.level = 0.9) |>
4   select(-statistic, -p.value) |> kable(digits = 2)

```

<b>term</b>	<b>estimate</b>	<b>std.error</b>	<b>conf.low</b>	<b>conf.high</b>
(Intercept)	53.43	7.69	40.69	66.16
x	-0.10	0.14	-0.33	0.12

```

1 glance(set_1) |>
2   select(r.squared, adj.r.squared, sigma, BIC, p.value) |>
3   kable(digits = 3)

```

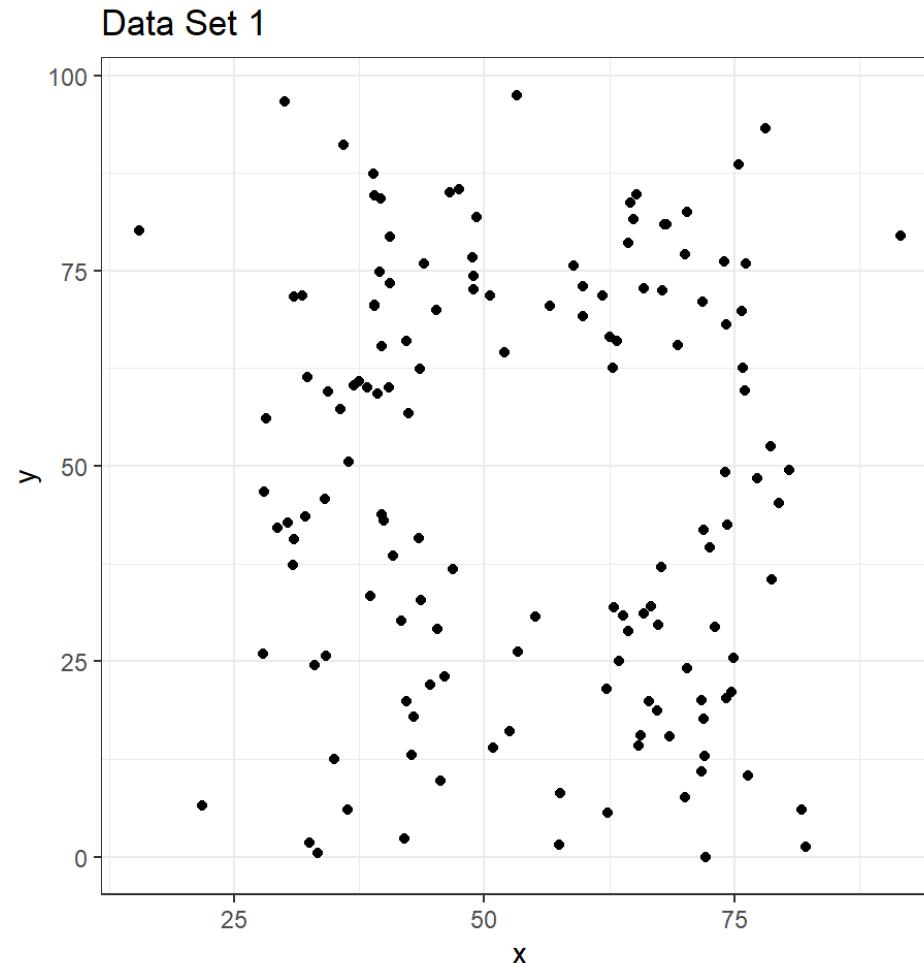
<b>r.squared</b>	<b>adj.r.squared</b>	<b>sigma</b>	<b>BIC</b>	<b>p.value</b>
0.004	-0.003	26.98	1351.64	0.448

# All 13 Models, at a glance

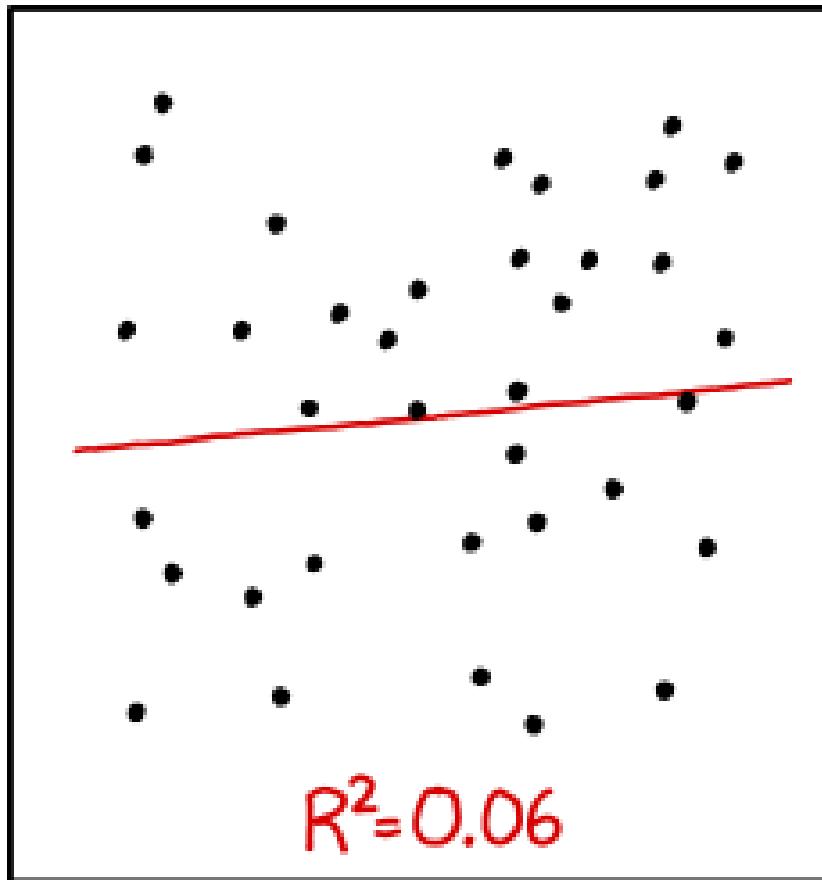
dataset	r.squared	adj.r.squared	sigma	AIC	BIC
1	0.004	-0.003	27	1343	1352
2	0.005	-0.002	27	1343	1352
3	0.005	-0.002	27	1343	1351
4	0.004	-0.003	27	1343	1352
5	0.004	-0.003	27	1343	1352
6	0.004	-0.003	27	1343	1352
7	0.005	-0.002	27	1343	1352
8	0.005	-0.002	27	1343	1352
9	0.005	-0.002	27	1343	1352
10	0.004	-0.003	27	1343	1352
11	0.005	-0.002	27	1343	1352
12	0.004	-0.003	27	1343	1352
13	0.004	-0.003	27	1343	1352

# Plot for Data Set 1

Does a linear model for  $y$  using  $x$  seem appropriate?

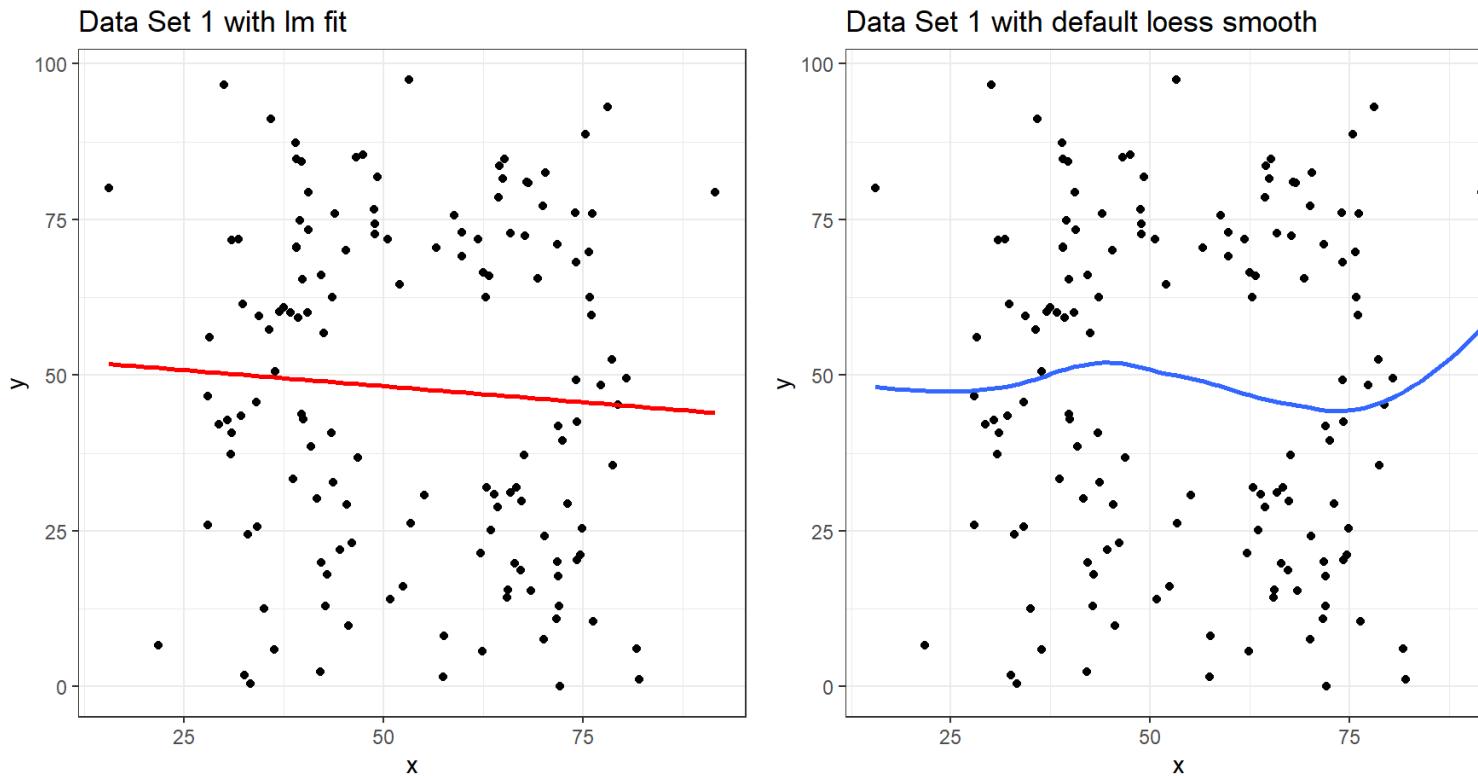


<https://xkcd.com/1725/>



I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER  
TO GUESS THE DIRECTION OF THE CORRELATION FROM THE  
SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

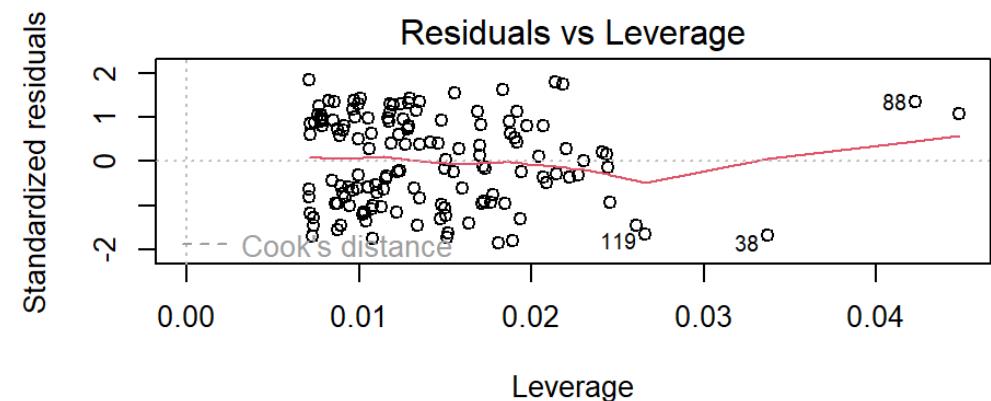
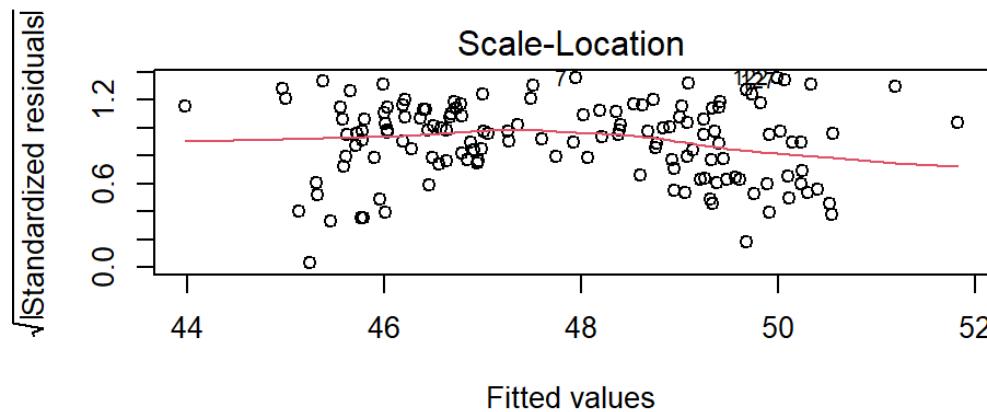
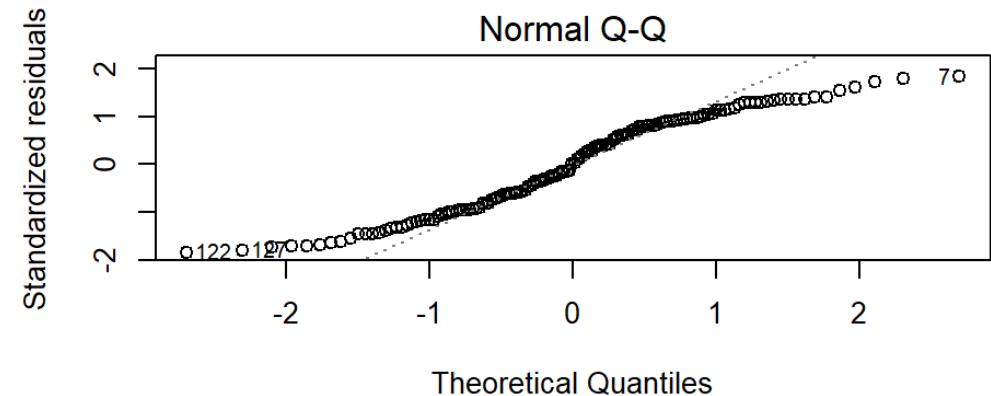
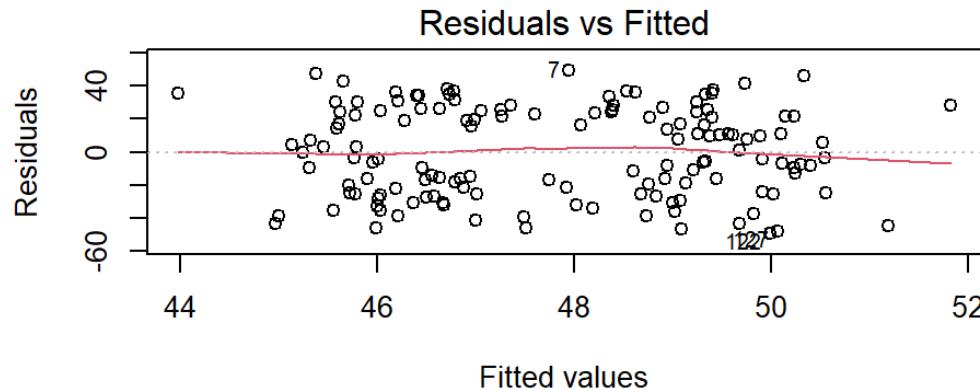
# Set 1 Plot (+lm, +loess)



## Model 1 (linear model for Set 1)

$$\hat{y} = 53.43 - 0.1(x)$$

# Residual Plots for Set 1 Model

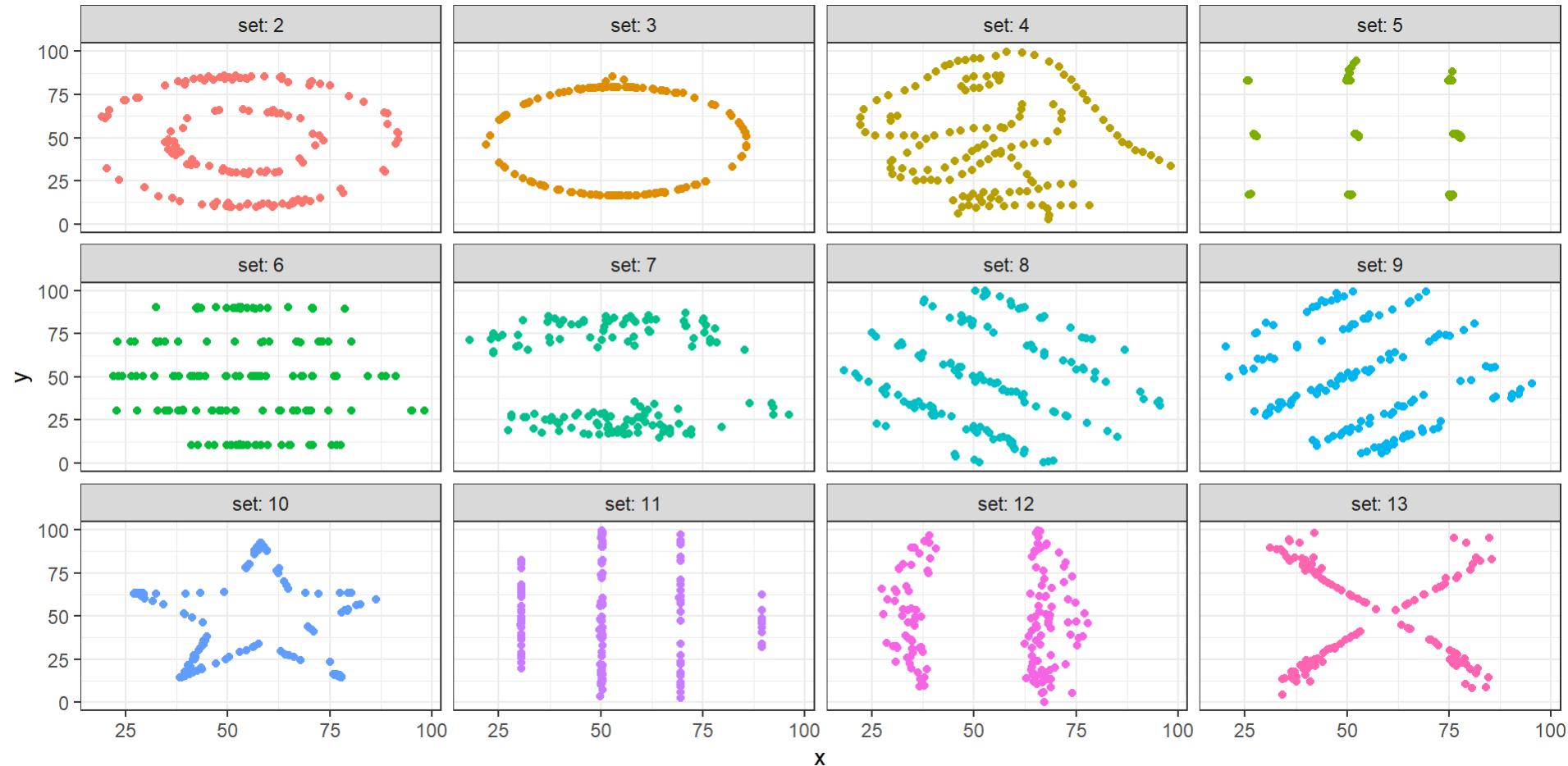


# The Other 12 Data Sets

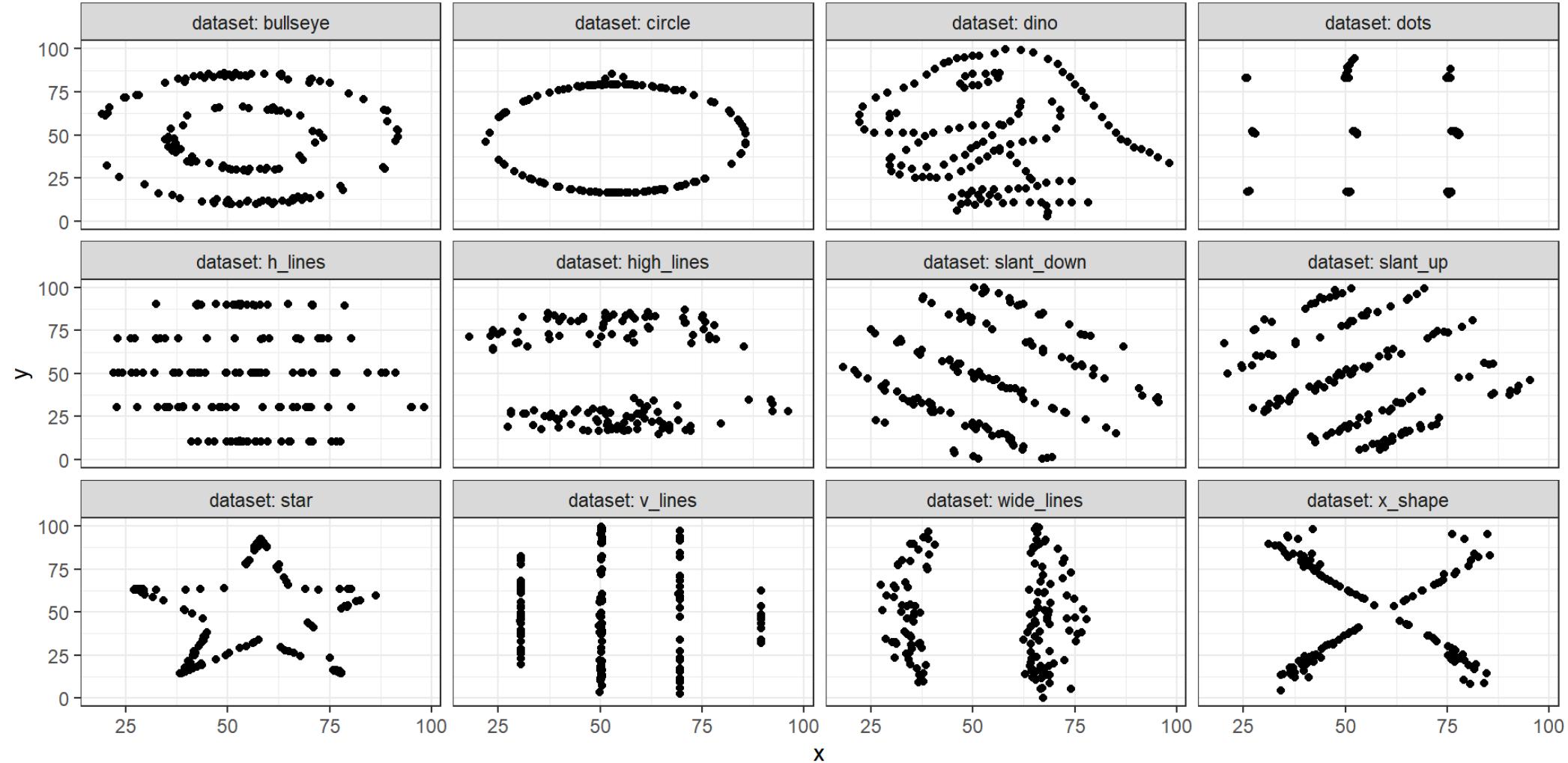
Models 2-13 all look about the same in terms of means, medians, correlations, regression models, but what happens if we plot the data?

```
1 temp2 <- df |> filter(set != 1)
2
3 ggplot(temp2, aes(x = x, y = y, color = dataset)) +
4   geom_point(show.legend = FALSE) +
5   facet_wrap(~ set, labeller = "label_both")
```

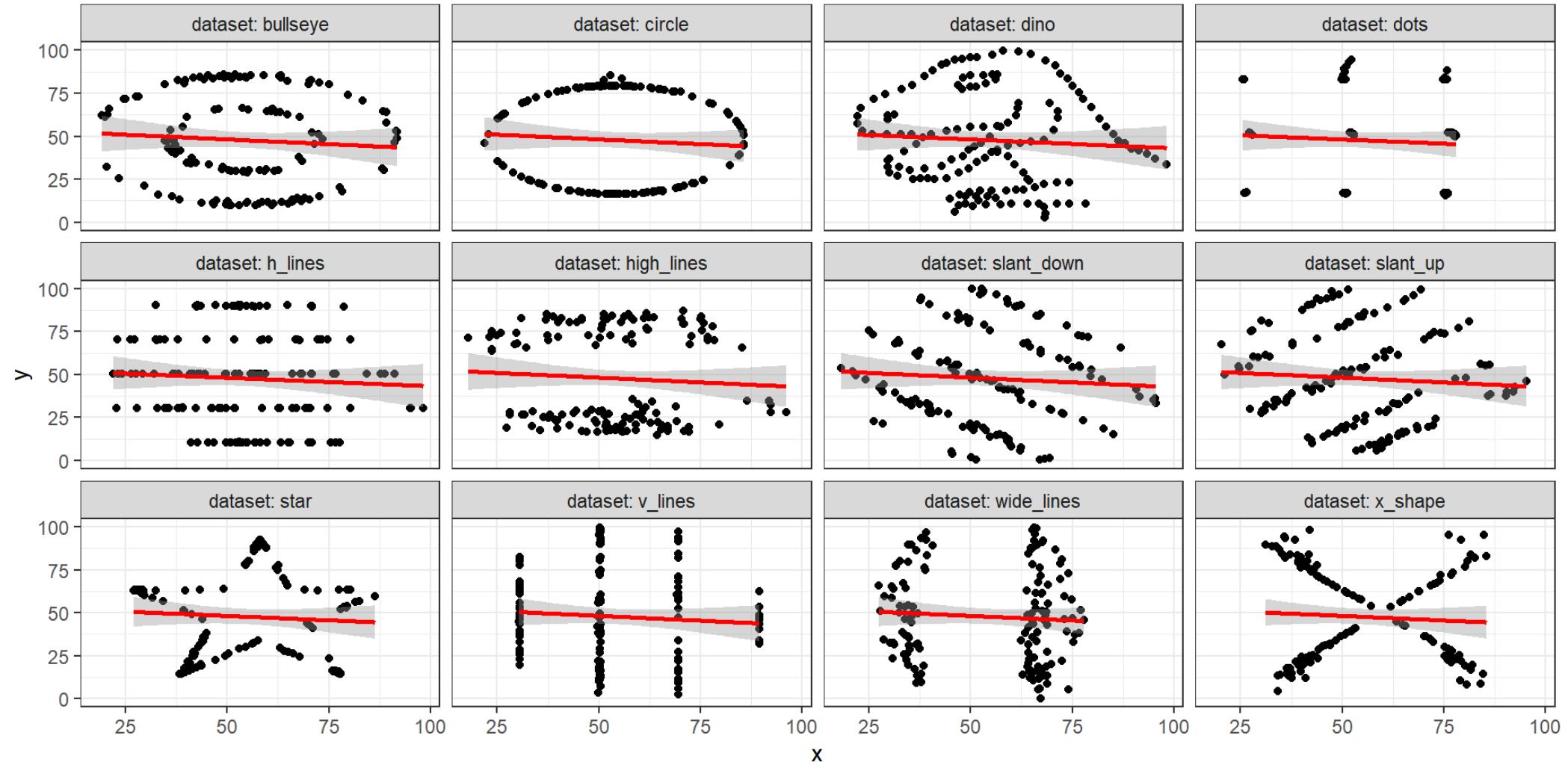
# The Other 12 Data Sets



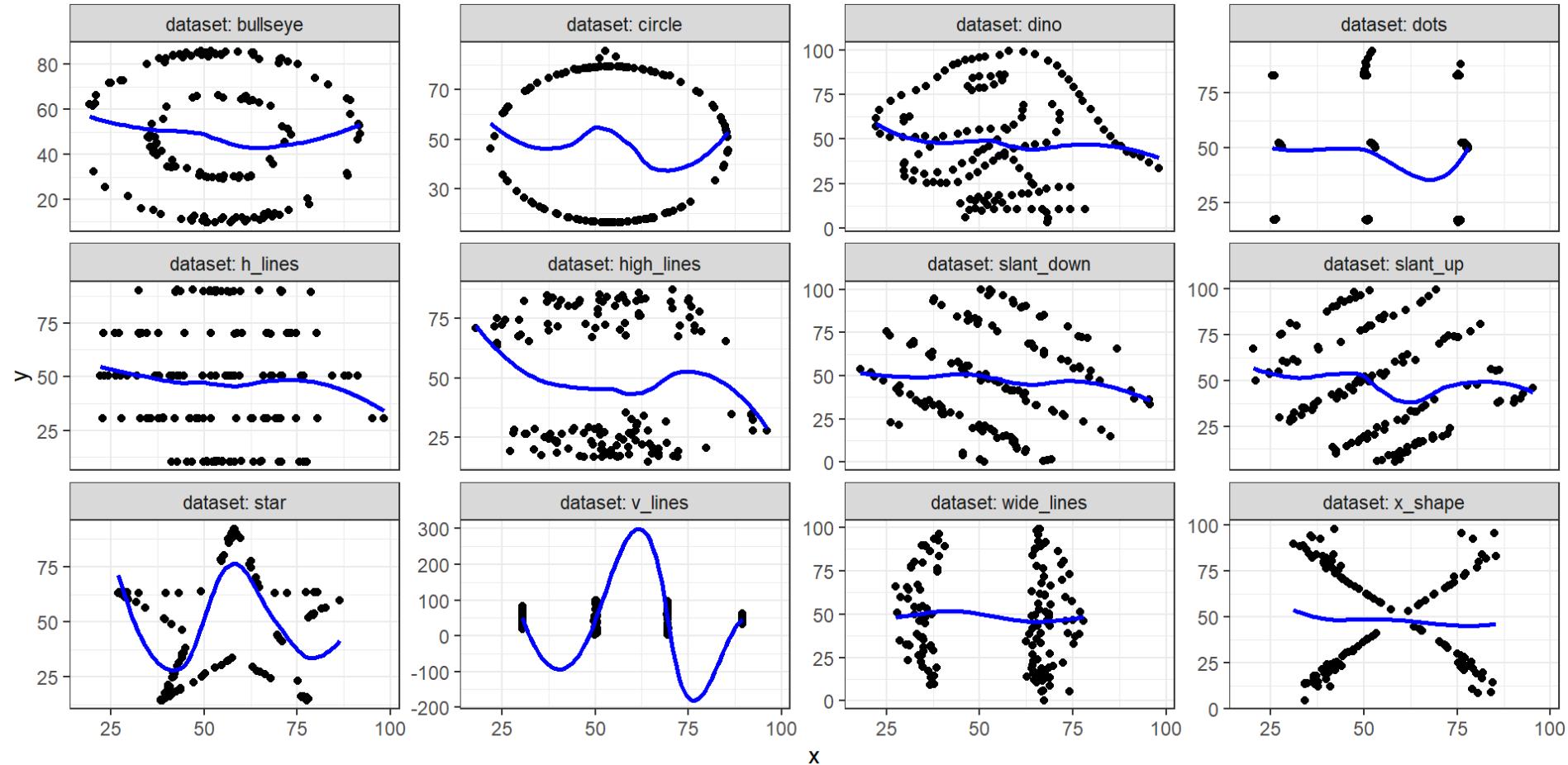
# Actually, each set has a name



# And a linear model yields the same fit for each



# How about a loess smooth with default span?



# And the data come from

These are, of course, the `datasauRus` dozen data sets described in Spiegelhalter, available in the `datasauRus` package, which you can install from CRAN, thanks to the work of Steph Locke.

```
library(datasauRus)
df <- datasaurus_dozen
```

- These were created by Alberto Cairo, who has some great books like *How Charts Lie*

The moral of the story: **Never trust summary statistics alone, always visualize your data**

# Two Cool Things, available online...

1. We'll visit Tomas Westlake's work at [https://r-mageddon.netlify.app/post/reanimating-the-datasaurus/](https://rmageddon.netlify.app/post/reanimating-the-datasaurus/)

```
1 library(datasauRus)
2 library(ggplot2)
3 library(gganimate)
4
5 ggplot(datasaurus_dozen, aes(x=x, y=y)) +
6   geom_point() +
7   theme_minimal() +
8   transition_states(dataset, 3, 1)
```

# Two Cool Things, available online...

2. Next, we'll visit

<https://www.autodesk.com/research/publications/same-stats-different-graphs>

This is Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing by Justin Matejka and George Fitzmaurice.

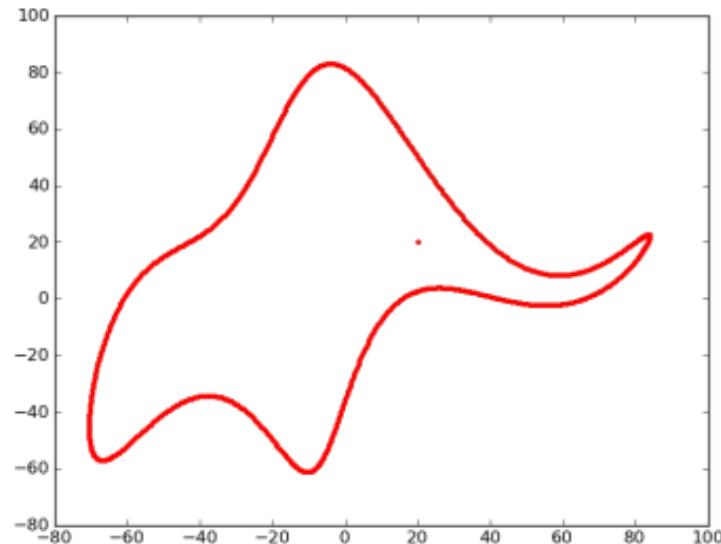
We'll look at a couple of the animated plots they generate there.

John von Neumann famously said

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

By this he meant that one should not be impressed when a complex model fits a data set well. With enough parameters, you can fit any data set.

It turns out you can literally fit an elephant with four parameters if you allow the parameters to be complex numbers.



# It's Just a Linear Model

# Common Statistical Tests are Linear Models

Jonas Kristoffer Lindelov has built up a terrific resources to explain this at

<https://lindeloev.github.io/tests-as-linear/>

What's the point?

# Common statistical tests are linear models

Last updated: 02 April, 2019

See worked examples and more details at the accompanying notebook: <https://lindeloev.github.io/tests-as-linear>

Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon	
Simple regression: $\text{Im}(y \sim 1 + x)$	<b>y is independent of x</b> P: One-sample t-test N: Wilcoxon signed-rank	<code>t.test(y)</code> <code>wilcox.test(y)</code>	<code>Im(y ~ 1)</code> <code>Im(signed_rank(y) ~ 1)</code>	✓ for N > 14	One number (intercept, i.e., the mean) predicts <b>y</b> . - (Same, but it predicts the <i>signed rank</i> of <b>y</b> .)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	<code>t.test(y1, y2, paired=TRUE)</code> <code>wilcox.test(y1, y2, paired=TRUE)</code>	<code>Im(y2 - y1 ~ 1)</code> <code>Im(signed_rank(y2 - y1) ~ 1)</code>	✓ for N > 14	One intercept predicts the pairwise <b>y<sub>2</sub>-y<sub>1</sub></b> differences. - (Same, but it predicts the <i>signed rank</i> of <b>y<sub>2</sub>-y<sub>1</sub></b> .)	
	<b>y ~ continuous x</b> P: Pearson correlation N: Spearman correlation	<code>cor.test(x, y, method='Pearson')</code> <code>cor.test(x, y, method='Spearman')</code>	<code>Im(y ~ 1 + x)</code> <code>Im(rank(y) ~ 1 + rank(x))</code>	✓ for N > 10	One intercept plus <b>x</b> multiplied by a number (slope) predicts <b>y</b> . - (Same, but with <i>ranked x</i> and <b>y</b> )	
	<b>y ~ discrete x</b> P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	<code>t.test(y1, y2, var.equal=TRUE)</code> <code>t.test(y1, y2, var.equal=FALSE)</code> <code>wilcox.test(y1, y2)</code>	<code>Im(y ~ 1 + G<sub>2</sub>)<sup>A</sup></code> <code>gls(y ~ 1 + G<sub>2</sub>, weights=...)<sup>B</sup></code> <code>Im(signed_rank(y) ~ 1 + G<sub>2</sub>)<sup>A</sup></code>	✓ ✓ for N > 11	An intercept for <b>group 1</b> (plus a difference if <b>group 2</b> ) predicts <b>y</b> . - (Same, but with one variance <i>per group</i> instead of one common.) - (Same, but it predicts the <i>signed rank</i> of <b>y</b> .)	
Multiple regression: $\text{Im}(y \sim 1 + x_1 + x_2 + \dots)$	P: One-way ANOVA N: Kruskal-Wallis	<code>aov(y ~ group)</code> <code>kruskal.test(y ~ group)</code>	<code>Im(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub>)<sup>A</sup></code> <code>Im(rank(y) ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub>)<sup>A</sup></code>	✓ for N > 11	An intercept for <b>group 1</b> (plus a difference if group ≠ 1) predicts <b>y</b> . - (Same, but it predicts the <i>rank</i> of <b>y</b> .)	
	P: One-way ANCOVA	<code>aov(y ~ group + x)</code>	<code>Im(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub> + x)<sup>A</sup></code>	✓	- (Same, but plus a slope on <b>x</b> .) Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.	
	P: Two-way ANOVA	<code>aov(y ~ group * sex)</code>	<code>Im(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub> + S<sub>2</sub> + S<sub>3</sub> + ... + S<sub>K</sub> + G<sub>2</sub>*S<sub>2</sub>+G<sub>3</sub>*S<sub>3</sub>+...+G<sub>N</sub>*S<sub>K</sub>)</code>	✓	Interaction term: changing <b>sex</b> changes the <b>y ~ group</b> parameters. Note: G <sub>2 to N</sub> is an <i>indicator (0 or 1)</i> for each non-intercept levels of the <b>group</b> variable. Similarly for S <sub>2 to K</sub> for sex. The first line (with G <sub>i</sub> ) is main effect of group, the second (with S <sub>j</sub> ) for sex and the third is the <b>group × sex</b> interaction. For two levels (e.g. male/female), line 2 would just be "S <sub>2</sub> " and line 3 would be S <sub>2</sub> multiplied with each G <sub>i</sub> .	[Coming]
	Counts ~ discrete x N: Chi-square test	<code>chisq.test(groupXsex_table)</code>	<b>Equivalent log-linear model</b> <code>glm(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub> + S<sub>2</sub> + S<sub>3</sub> + ... + S<sub>K</sub> + G<sub>2</sub>*S<sub>2</sub>+G<sub>3</sub>*S<sub>3</sub>+...+G<sub>N</sub>*S<sub>K</sub>, family=...)<sup>A</sup></code>	✓	Interaction term: (Same as Two-way ANOVA.) Note: Run <code>glm</code> using the following arguments: <code>glm(model, family=poisson())</code> As linear-model, the Chi-square test is $\log(y) = \log(N) + \log(\alpha) + \log(\beta) + \log(\alpha\beta)$ where $\alpha$ and $\beta$ are proportions. See more info in <a href="https://lindeloev.github.io/tests-as-linear">the accompanying notebook</a> .	Same as Two-way ANOVA
	N: Goodness of fit	<code>chisq.test(y)</code>	<code>glm(y ~ 1 + G<sub>2</sub> + G<sub>3</sub> + ... + G<sub>N</sub>, family=...)<sup>A</sup></code>	✓	(Same as One-way ANOVA and see Chi-Square note.)	1W-ANOVA

List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation  $y \sim 1 + x$  is R shorthand for  $y = 1 \cdot b + a \cdot x$  which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they *all* are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see "Exact" column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `signed_rank = function(x) sign(x) * rank(abs(x))`. The variables G<sub>i</sub> and S<sub>j</sub> are "dummy coded" *indicator variables* (either 0 or 1) exploiting the fact that when  $\Delta x = 1$  between categories the difference equals the slope. Subscripts (e.g., G<sub>2</sub> or y<sub>1</sub>) indicate different columns in data. Im requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at <https://lindeloev.github.io/tests-as-linear>.

<sup>A</sup> See the note to the two-way ANOVA for explanation of the notation.

<sup>B</sup> Same model, but with one variance per group: `gls(value ~ 1 + G2, weights = varIdent(form = ~1|group), method="ML")`.



Jonas Kristoffer Lindeløv  
<https://lindeloev.net>

# Consider Study 1 from Project B.

- Analysis A. Compare two means/medians using paired samples
  - This is a linear model. See Section 4.2 of [Lindelov](#)

## 4.2.2 R code: Paired sample t-test

```
a = t.test(y, y2, paired = TRUE) # Built-in paired t-test  
b = lm(y - y2 ~ 1) # Equivalent linear model
```

Results:

model	mean	p.value	df	t	conf.low	conf.high
t.test	-0.5952	0.0934	49	-1.7108	-1.2944	0.104
lm	-0.5952	0.0934	49	-1.7108	-1.2944	0.104

# Project B Study 1?

- Analysis B. Compare two means/medians using independent samples
  - This is a linear model, and not just for the t test. See Section 5 of Lindelov
- Analysis C. Compare 3-6 means/medians using independent samples
  - ANOVA is obviously a linear model, but actually we can generate (essentially) the Kruskal-Wallis this way, too. See Section 6.1 of Lindelov

# Project B Study 1?

- Analysis D. Create and analyze a  $2 \times 2$  table
  - Yes, the chi-square test of independence can emerge from a linear model. See Section 7.2 of [Lindelov](#)
- Analysis E. Create and analyze a  $J \times K$  table, where  $2 \leq J \leq 5$  and  $3 \leq K \leq 5$ 
  - Linear model, as in the  $2 \times 2$  case. See Section 7.2 of [Lindelov](#)

Analyses D-E are more commonly thought about in the context of generalized linear models, as we'll see in 432.

# A Taste of 432: Sea Urchins and Tidy Modeling



A recent count found 350 million purple sea urchins on one Oregon reef alone. (Shutterstock)

# Sea Urchins and Tidy Modeling

Constable (1993) compared the inter-radial suture widths of urchins maintained on one of three food regimes

- Initial: no additional food supplied above what was in the initial sample
- Low: food supplied periodically
- High: food supplied ad libitum (as often as desired)

In an attempt to control for substantial variability in urchin sizes, the initial body volume of each urchin was measured as a covariate.

- This example comes from <https://www.tidymodels.org/start/models/>
- Another key source is <https://www.flutterbys.com.au/stats/tut/tut7.5a.html>
- Data from Constable, A.J. The role of sutures in shrinking of the test in *Heliocidaris erythrogramma* (Echinoidea: Echinometridae). *Marine Biology* 117, 423-430 (1993). <https://doi.org/10.1007/BF00349318>

# Package Load / Data ingest (Sea Urchins)

```
1 library(tidymodels)
2 library(readr)
3 library(broom.mixed)
4
5 urchins <-
6   # Data were assembled for a tutorial
7   # at https://www.flutterbys.com.au/stats/tut/tut7.5a.html
8   read_csv("https://tidymodels.org/start/models/urchins.csv") |>
9   # Change the names to be a little more verbose
10  setNames(c("food_regime", "initial_volume", "width")) |>
11  mutate(food_regime =
12    factor(food_regime,
13      levels = c("Initial", "Low", "High")))
```

# The `urchins` data

For each of 72 sea urchins, we know their

- experimental feeding regime group (`food_regime`: either Initial, Low, or High),
- size in milliliters at the start of the experiment (`initial_volume`), and
- suture width at the end of the experiment (`width`).

```
1 glimpse(urchins)
```

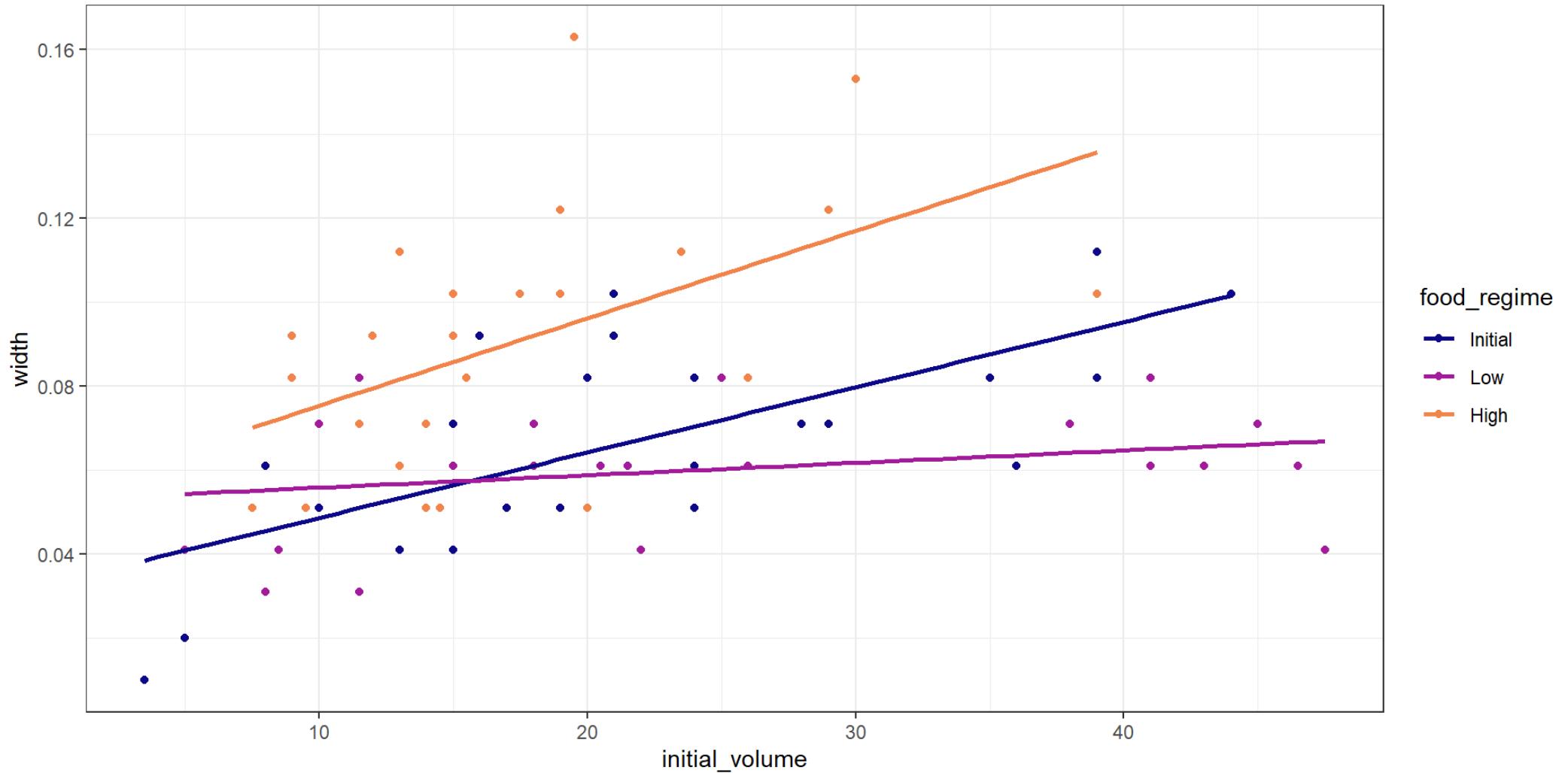
Rows: 72

Columns: 3

\$ food\_regime <fct> Initial, Initial, Initial, Initial, Initial, Initial,  
I...  
\$ initial\_volume <dbl> 3.5, 5.0, 8.0, 10.0, 13.0, 13.0, 15.0, 15.0, 16.0,

17.0...

# Plot the Data



# How should we model the data?

Since the slopes appear to be different for at least two of the feeding regimes, let's build a model that allows for two-way interactions. We'll use a linear model for width which allows each food regime to generate a different slope and intercept for the effect of initial volume.

```
1 lm(width ~ initial_volume * food_regime, data = urchins) |> tidy() |>  
2   select(term, estimate) |> kable(dig = 4) |> kable_styling(font_size = 24)
```

term	estimate
(Intercept)	0.0331
initial_volume	0.0016
food_regimeLow	0.0198
food_regimeHigh	0.0214
initial_volume:food_regimeLow	-0.0013
initial_volume:food_regimeHigh	0.0005

# Setting up a linear regression with tidymodels

```
1 lm_mod <-
2   linear_reg() |>
3   set_engine("lm")
4
5 lm_mod
```

Linear Regression Model Specification (regression)

Computational engine: lm

It turns out that we'll have several options for engines here.

# We can estimate or train the model with `fit()`

```
1 lm_fit <-  
2   lm_mod |>  
3   fit(width ~ initial_volume * food_regime, data = urchins)
```

We'll look at the results on the next slide.

# What's in lm\_fit?

```
1 tidy(lm_fit, conf.int = TRUE) |> select(term, estimate, conf.low, conf.high  
2 kable(dig = 4) |> kable_styling(font_size = 28)
```

term	estimate	conf.low	conf.high
(Intercept)	0.0331	0.0139	0.0523
initial_volume	0.0016	0.0008	0.0023
food_regimeLow	0.0198	-0.0061	0.0457
food_regimeHigh	0.0214	-0.0076	0.0504
initial_volume:food_regimeLow	-0.0013	-0.0023	-0.0002
initial_volume:food_regimeHigh	0.0005	-0.0009	0.0019

# Make Predictions

Suppose that, for a publication, it would be particularly interesting to make a plot of the mean body size for urchins that started the experiment with an initial volume of 20ml. To create such a graph, we start with some new example data that we will make predictions for.

```
1 new_points <- expand.grid(initial_volume = 20,  
2                               food_regime = c("Initial", "Low", "High"))  
3  
4 new_points
```

	initial_volume	food_regime
1	20	Initial
2	20	Low
3	20	High

# Obtain Predicted Results for these new\_points

We'll develop mean predictions and uncertainty intervals.

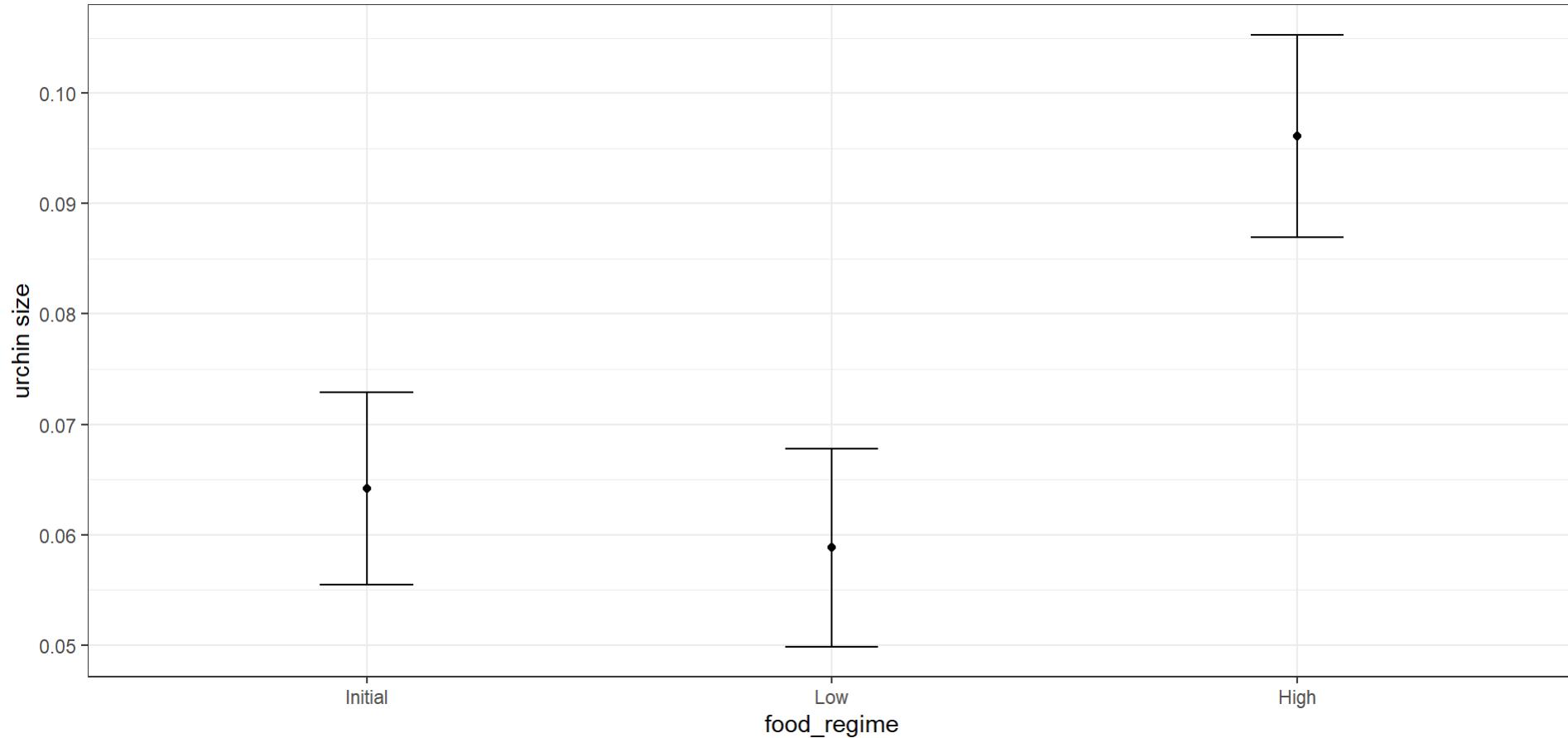
```
1 mean_pred <- predict(lm_fit, new_data = new_points)
2 conf_int_pred <- predict(lm_fit,
3                           new_data = new_points,
4                           type = "conf_int")
5 plot_data <-
6   new_points |>
7   bind_cols(mean_pred) |>
8   bind_cols(conf_int_pred)
```

# Plot the `plot_data` results

```
1 ggplot(plot_data, aes(x = food_regime, y = .pred)) +
2   geom_point() +
3   geom_errorbar(aes(ymin = .pred_lower,
4                     ymax = .pred_upper),
5                 width = .2) +
6   labs(y = "urchin size",
7        title = "Linear model fit using `lm`")
```

# Plot the `plot_data` results

Linear model fit using `lm`



# Could we use Bayesian methods?

Would the results be different if we used a Bayesian approach?

- Need to select a prior.
- Let's use bell-shaped priors on the intercepts and slopes, using a Cauchy distribution (works out to be the same as a t distribution with one degree of freedom)
- The `stan_glm()` function can be used, and this is available as an engine in `tidymodels`, where we need to specify `prior` and `prior_intercept` to fit a linear model.

# Setting up a Bayesian Model

```
1 prior_dist <- rstanarm::student_t(df = 1)
2
3 set.seed(123)
4
5 bayes_mod <-
6   linear_reg() |>
7   set_engine("stan",
8             prior_intercept = prior_dist,
9             prior = prior_dist)
```

# Training the Bayes Model

```
1 bayes_fit <- bayes_mod |>
2   fit(width ~ initial_volume * food_regime, data = urchins)
3
4 tidy(bayes_fit, conf.int = TRUE) |>
5   select(term, estimate, conf.low, conf.high) |>
6   kable(dig = 4) |> kable_styling(font_size = 24)
```

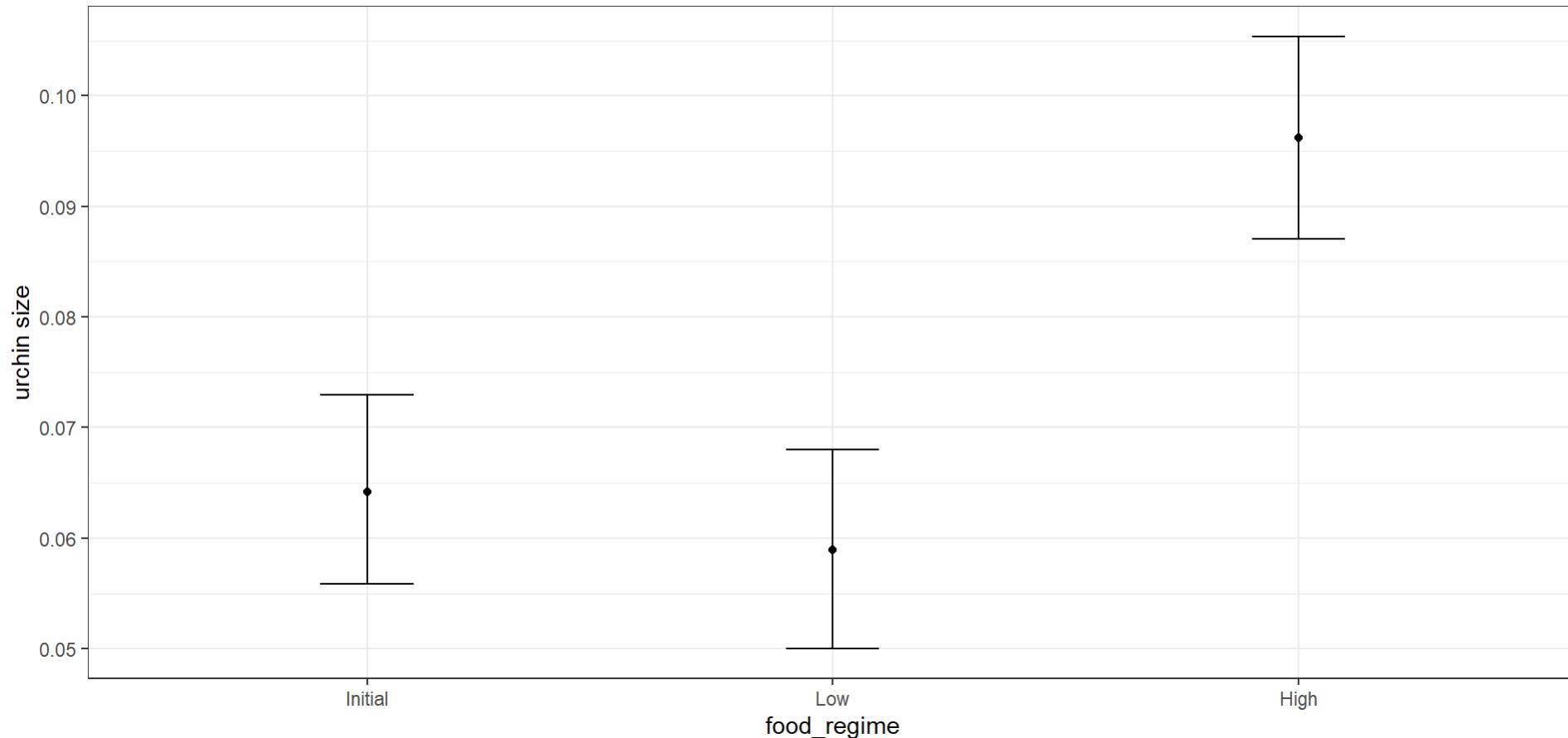
term	estimate	conf.low	conf.high
(Intercept)	0.0330	0.0173	0.0487
initial_volume	0.0016	0.0009	0.0022
food_regimeLow	0.0204	-0.0016	0.0415
food_regimeHigh	0.0214	-0.0020	0.0456
initial_volume:food_regimeLow	-0.0013	-0.0021	-0.0004
initial_volume:food_regimeHigh	0.0005	-0.0006	0.0016

# Building the plot for the Bayes model

```
1 bayes_plot_data <-
2   new_points |>
3   bind_cols(predict(bayes_fit, new_data = new_points)) |>
4   bind_cols(predict(bayes_fit, new_data = new_points,
5                     type = "conf_int"))
6
7 ggplot(bayes_plot_data, aes(x = food_regime, y = .pred)) +
8   geom_point() +
9   geom_errorbar(aes(ymin = .pred_lower, ymax = .pred_upper),
10                 width = .2) +
11   labs(y = "urchin size",
12        title = "Bayesian model with t(1) prior distribution")
```

# Building the plot for the Bayes model

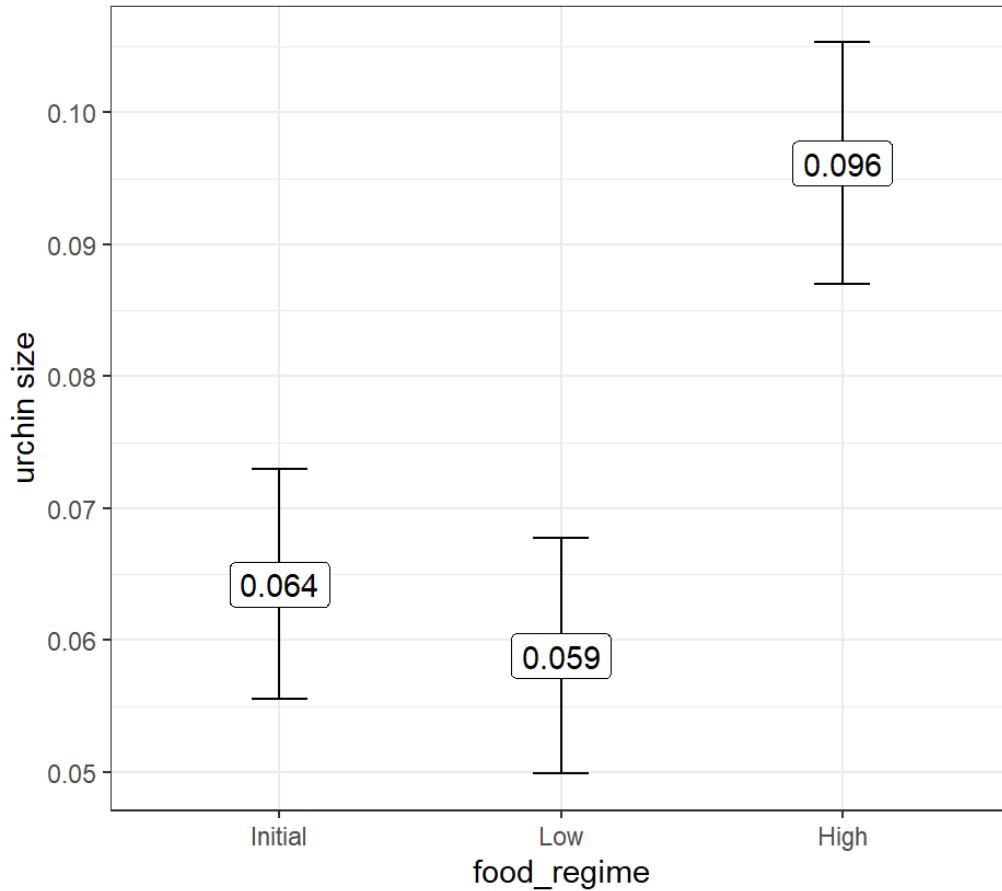
Bayesian model with t(1) prior distribution



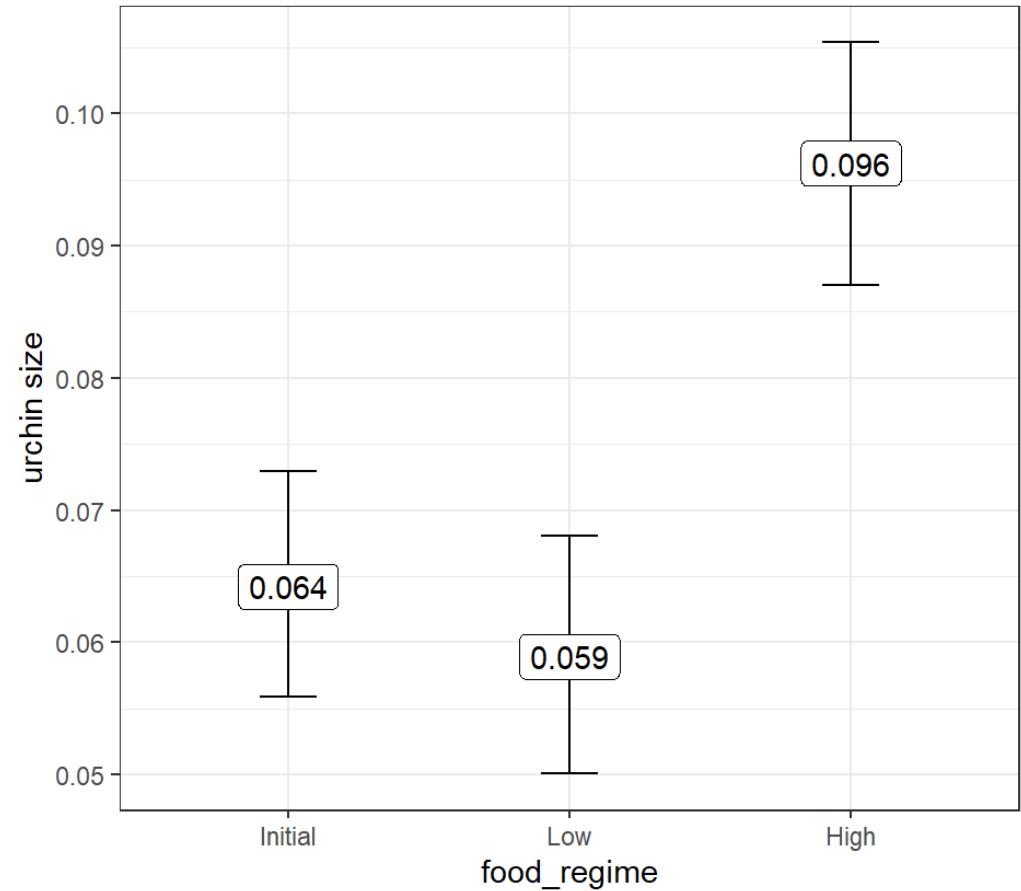
# Comparing the Models

Comparing linear models for urchins data

Linear model fit using `lm`

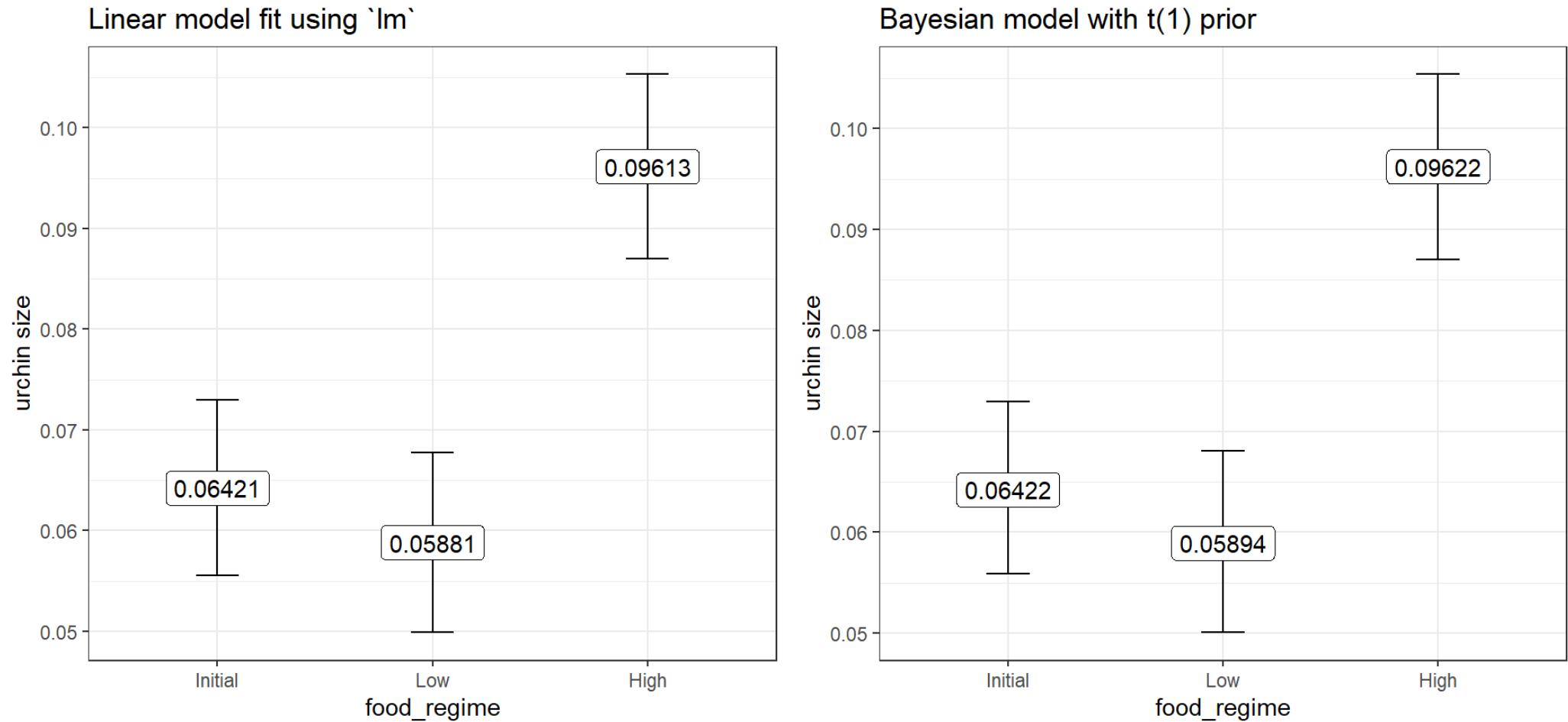


Bayesian model with  $t(1)$  prior



# The models aren't actually the same

Comparing linear models for urchins data



# What are we plotting, actually?

```
1 plot_data |> kable(dig = 4) |> kable_styling(font_size = 28)
```

initial_volume	food_regime	.pred	.pred_lower	.pred_upper
20	Initial	0.0642	0.0555	0.0729
20	Low	0.0588	0.0499	0.0678
20	High	0.0961	0.0870	0.1053

```
1 bayes_plot_data |> kable(dig = 4) |> kable_styling(font_size = 28)
```

initial_volume	food_regime	.pred	.pred_lower	.pred_upper
20	Initial	0.0642	0.0559	0.0729
20	Low	0.0589	0.0501	0.0680
20	High	0.0962	0.0870	0.1054

What do we take away  
from 431 at the end of the  
day?

# Ten Simple Rules for Effective Statistical Practice

From PLoS Computational Biology

1. Statistical Methods Should Enable Data to Answer Scientific Questions
2. Signals Always Come with Noise
3. Plan Ahead, Really Ahead
4. Worry About Data Quality
5. Statistical Analysis Is More Than a Set of Computations

# Ten Simple Rules for Effective Statistical Practice

From PLoS Computational Biology

6. Keep it Simple

7. Provide Assessments of Variability

8. Check Your Assumptions

9. When Possible, Replicate!

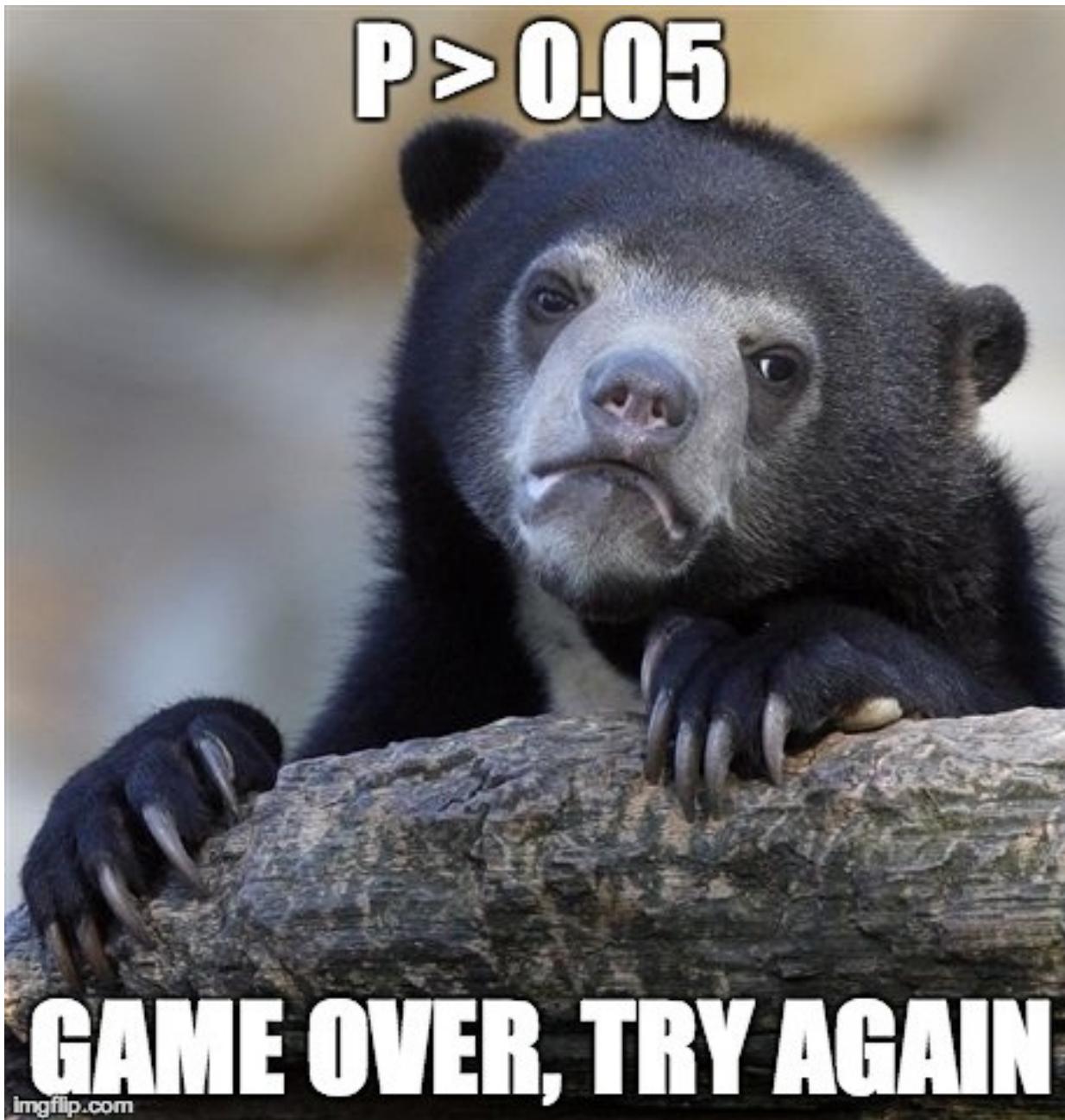
10. Make Your Analysis Reproducible

# The Impact of Study Design (Gelman)

Applied statistics is hard.

- Doing a statistical analysis is like playing basketball, or knitting a sweater. You can get better with practice.
- Incompetent statistics does not necessarily doom a research paper: some findings are solid enough that they show up even when there are mistakes in the data collection and data analyses. But we've also seen many examples where incompetent statistics led to conclusions that made no sense but still received publication and publicity.
- We should be thinking not just about data analysis, but also data quality.

To consult the statistician after an experiment is finished is often merely to ask him to conduct a post mortem examination. He can perhaps say what the experiment died of. (R. A. Fisher)



P > 0.05

GAME OVER, TRY AGAIN

imgflip.com

# What does sad p-value bear lead to?

So you collected data and analyzed the results. Now you want to do an after data gathering (post hoc) power analysis.

## 1. What will you use as your “true” effect size?

- Often, point estimate from data - yuck - results very misleading - power is generally seriously overestimated when computed on the basis of statistically significant results.
- Much better (but rarer) to identify plausible effect sizes based on external information rather than on your sparkling new result.

## 2. What are you trying to do? (too often)

- get researcher off the hook (I didn't get  $p < 0.05$  because I had low power - an alibi to explain away non-significant findings) or
- encourage overconfidence in the finding.

None of this is particularly smart.

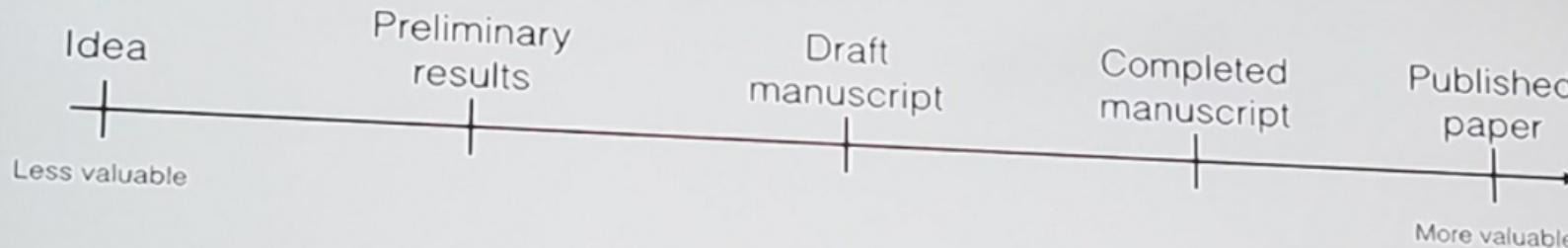
# Build Tidy Data Sets

- Each variable you measure should be in one column.
- Each different observation of that variable should be in a different row.
- There should be one table for each “kind” of variable.
- If you have multiple tables, they should include a column in the table that allows them to be linked.
- Include a row at the top of each data table that contains real row names.  
*Age\_at\_Diagnosis* is a much much better name than *ADx*.
- Build useful codebooks.

Jeff Leek: “[How to share data with a statistician](#)”

# A Tip from David Robinson

**How I thought of my goals in grad school:**



**How I should have been thinking of them:**



# The Course So Far

1. Statistics is too important to be left to statisticians.
2. Models and visualization are the big takeaways, but don't forget about methods for making statistical inferences.
3. Reproducible research is the current wave.
4. Things are changing quickly. We live in interesting times.



*That's all Folks!*

# Session Information

```
1 sessionInfo()
```

```
R version 4.2.2 (2022-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22000)
```

```
Matrix products: default
```

```
locale:
```

```
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8
```

```
attached base packages:
```

```
[1] stats      graphics   grDevices utils      datasets   methods    base
```

