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

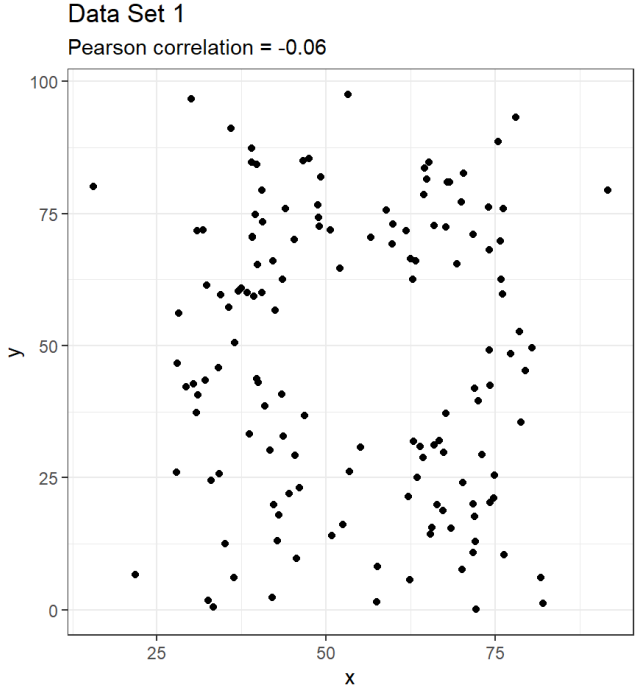
```
library(knitr); library(kableExtra)
library(janitor); library(mosaic)
library(equatiomatic); library(patchwork)
library(broom)
library(tidyverse)

opts_chunk$set(comment=NA)
options(dplyr.summarise.inform = FALSE)
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)
```

term	estimate	std.error	conf.low	conf.high
(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)
```

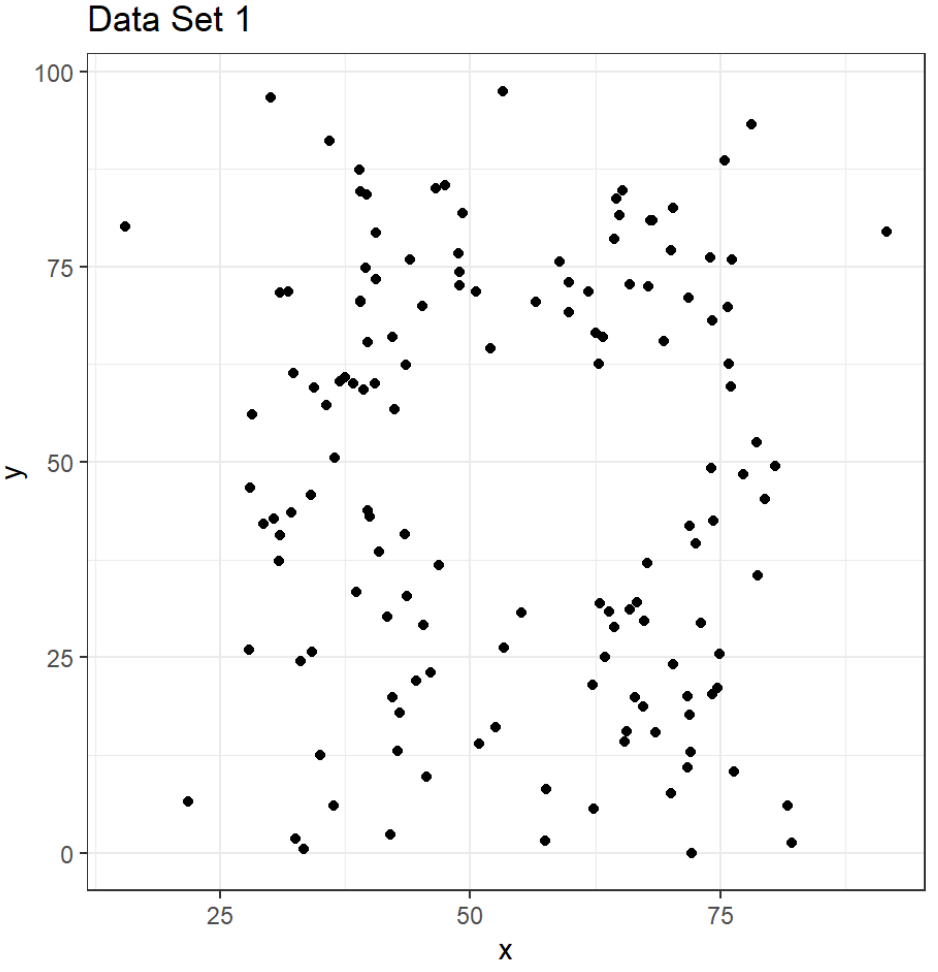
r.squared	adj.r.squared	sigma	BIC	p.value
0.004	-0.003	26.98	1351.64	0.448

All 13 Models, at a glance

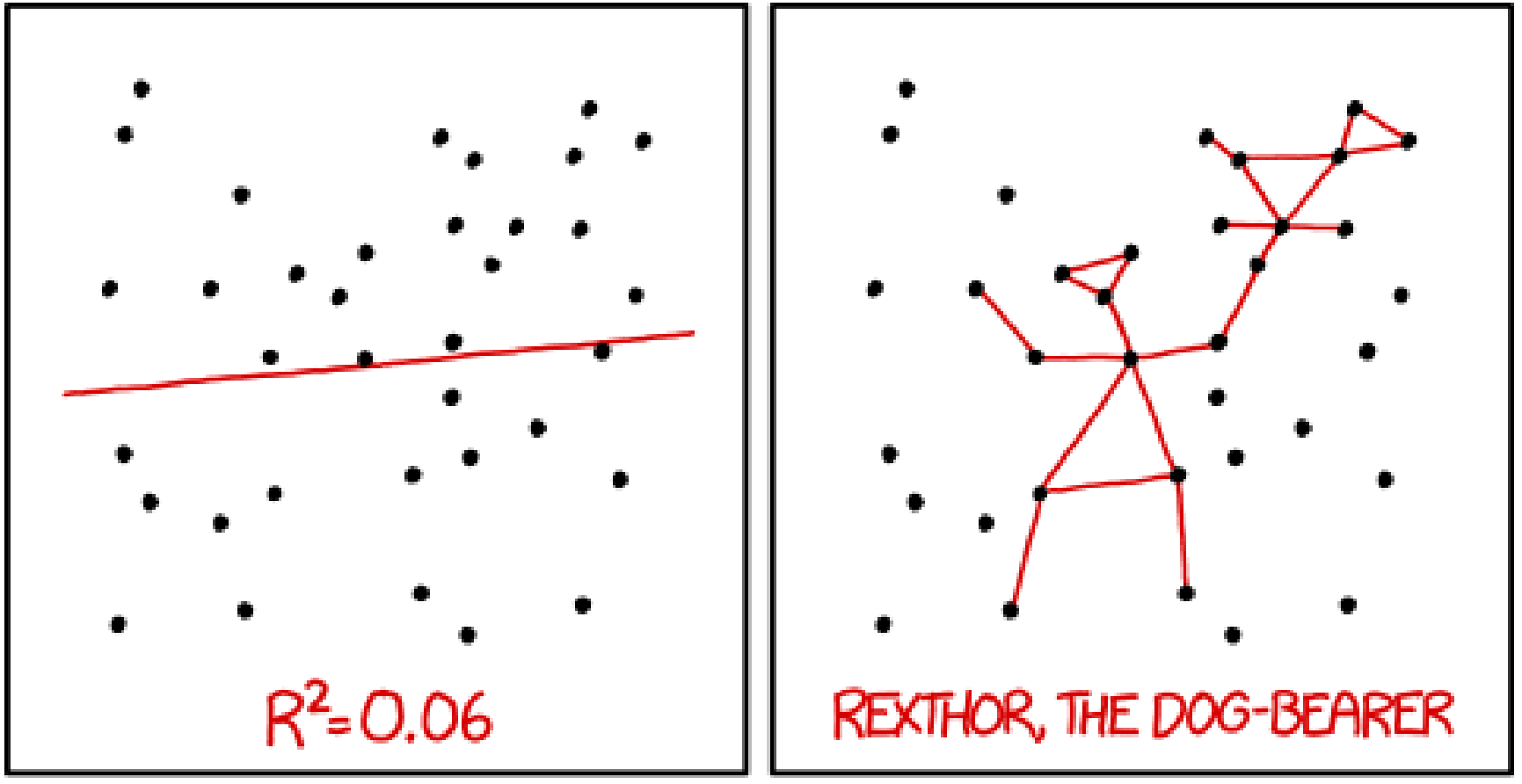
dataset	r.squared	adj.r.squared	sigma	AIC	BIC
1	0.004	-0.003	27	1343	1352
2	0.005	-0.002	27	1343	1352
3	0.005	-0.002	27	1343	1351
4	0.004	-0.003	27	1343	1352
5	0.004	-0.003	27	1343	1352
6	0.004	-0.003	27	1343	1352
7	0.005	-0.002	27	1343	1352
8	0.005	-0.002	27	1343	1352
9	0.005	-0.002	27	1343	1352
10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

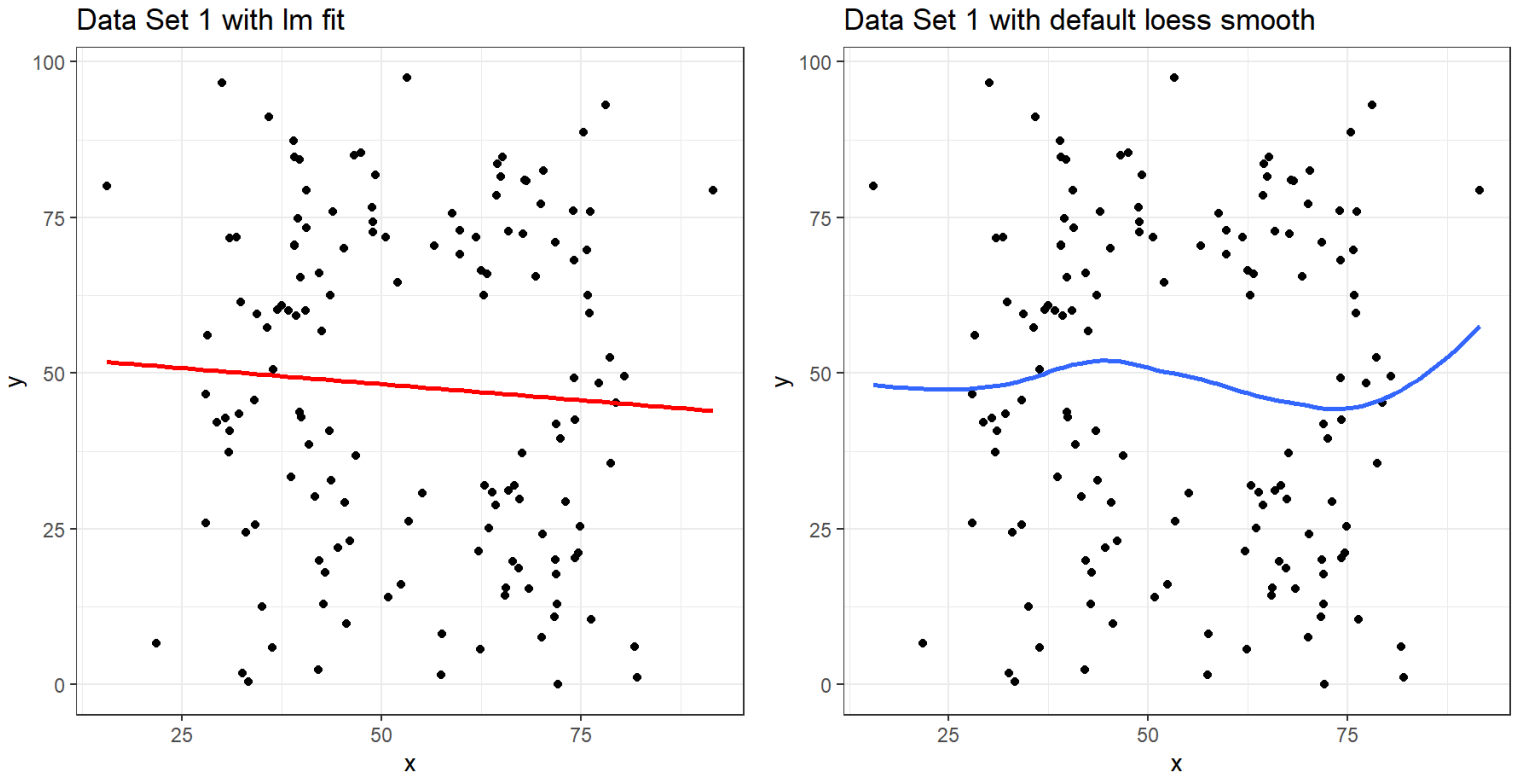


https://xkcd.com/1725/



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.

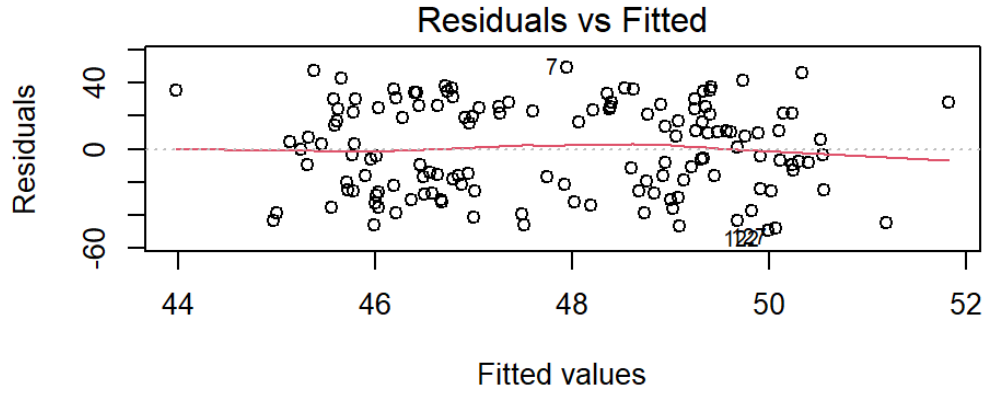
Set 1 Plot (+lm, +loess)

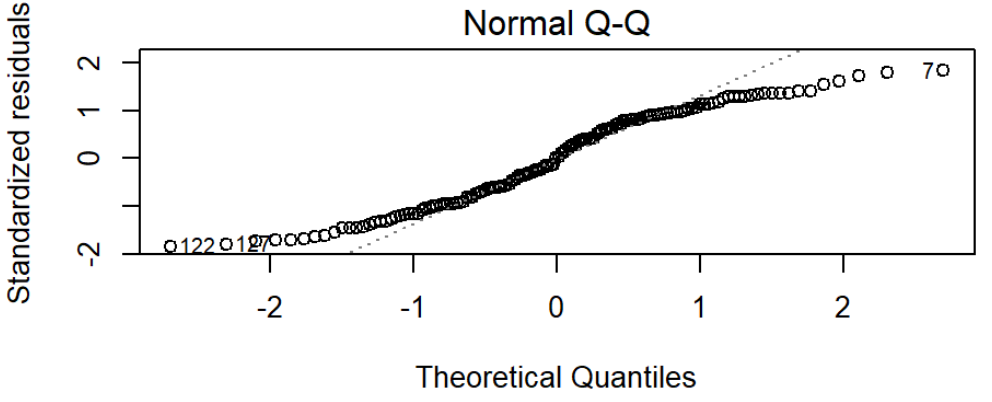


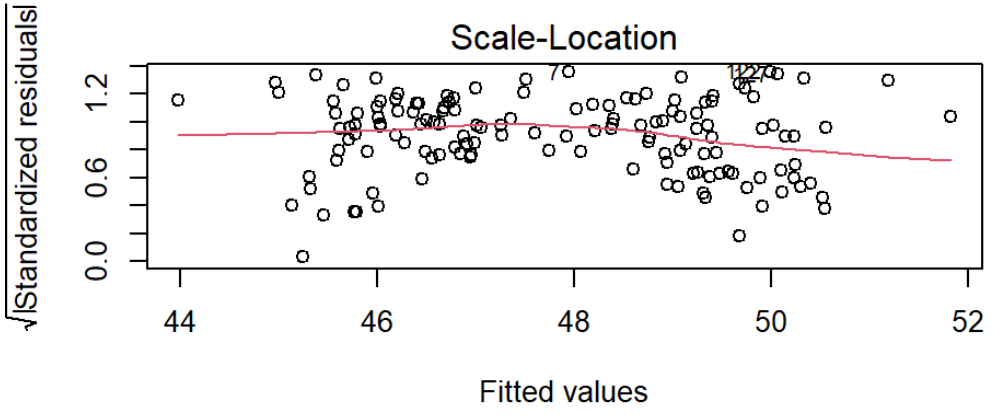
Model 1 (linear model for Set 1)

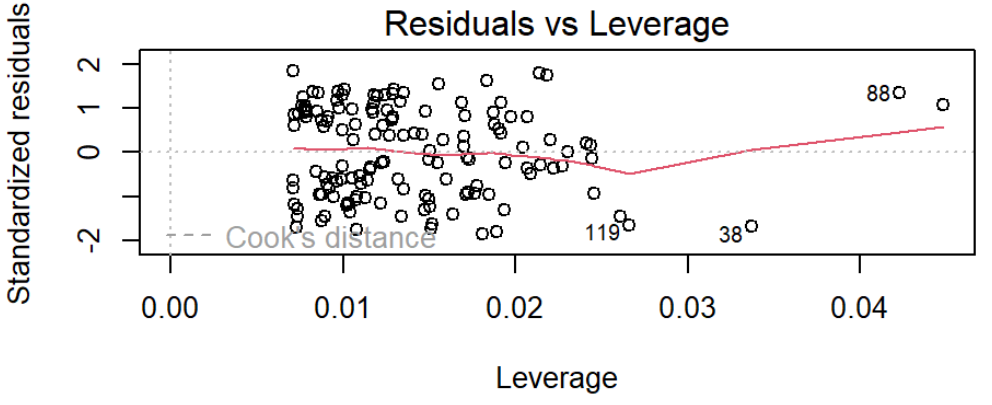
$$\operatorname{\widehat{y}} = 53.43 - 0.1(\operatorname{x})$$

Residual Plots for Set 1 Model







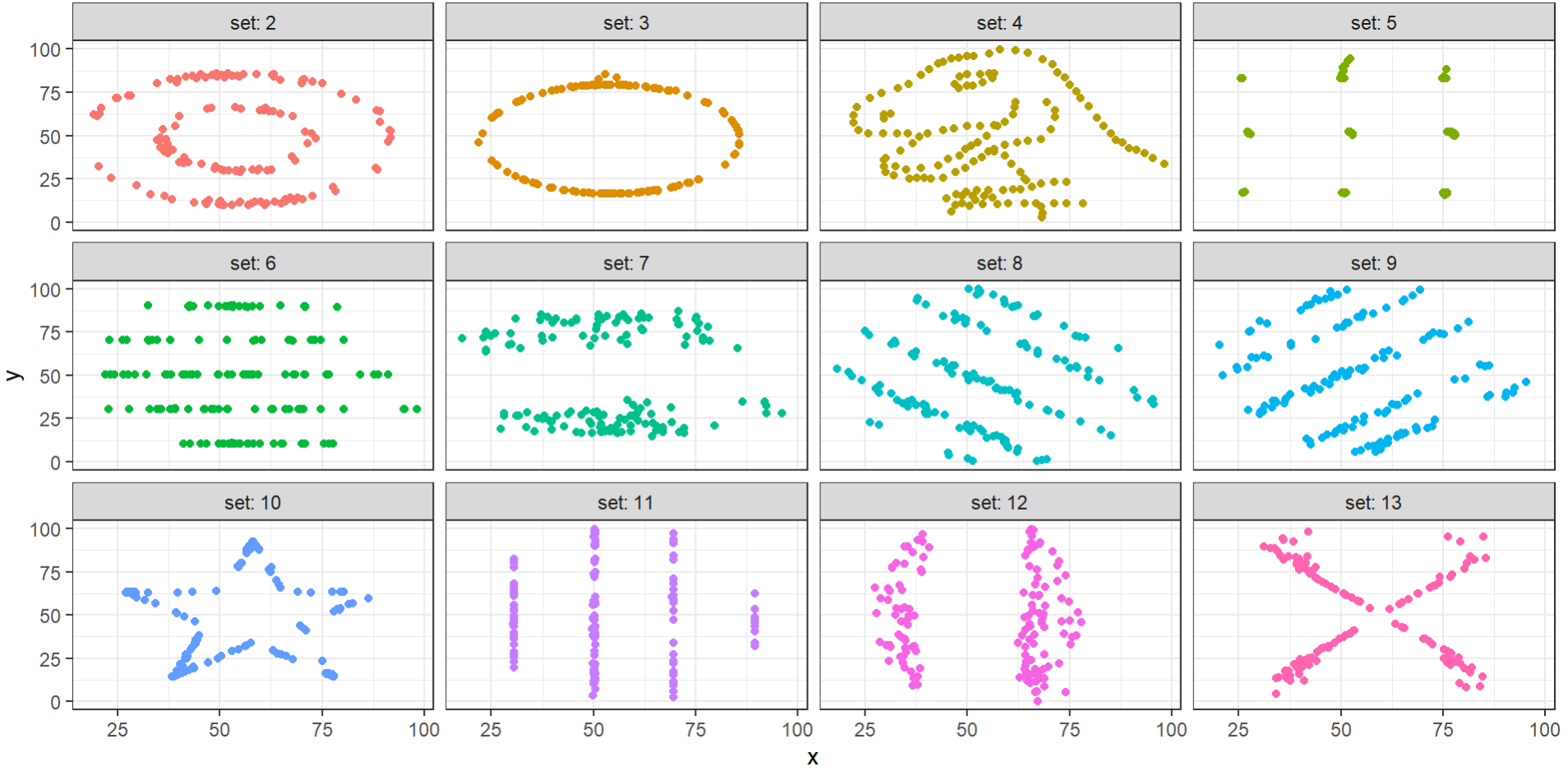


The Other 12 Data Sets

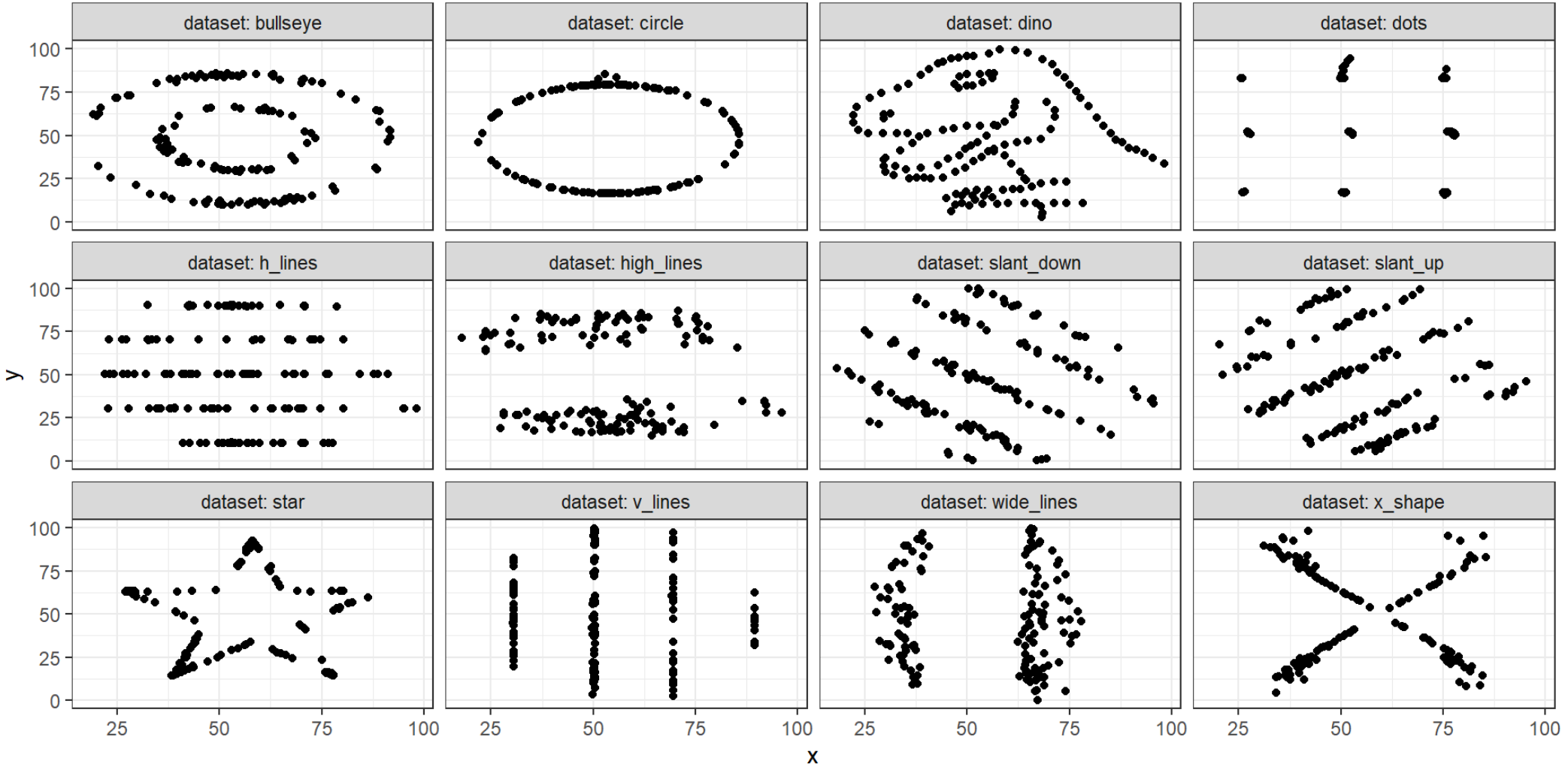
Models 2-13 all look about the same in terms of means, medians, correlations, regression models, but what happens if we plot the data?

```
temp2 <- df |> filter(set != 1)

ggplot(temp2, aes(x = x, y = y, color = dataset)) +
  geom_point(show.legend = FALSE) +
  facet_wrap(~ set, labeller = "label_both")
```

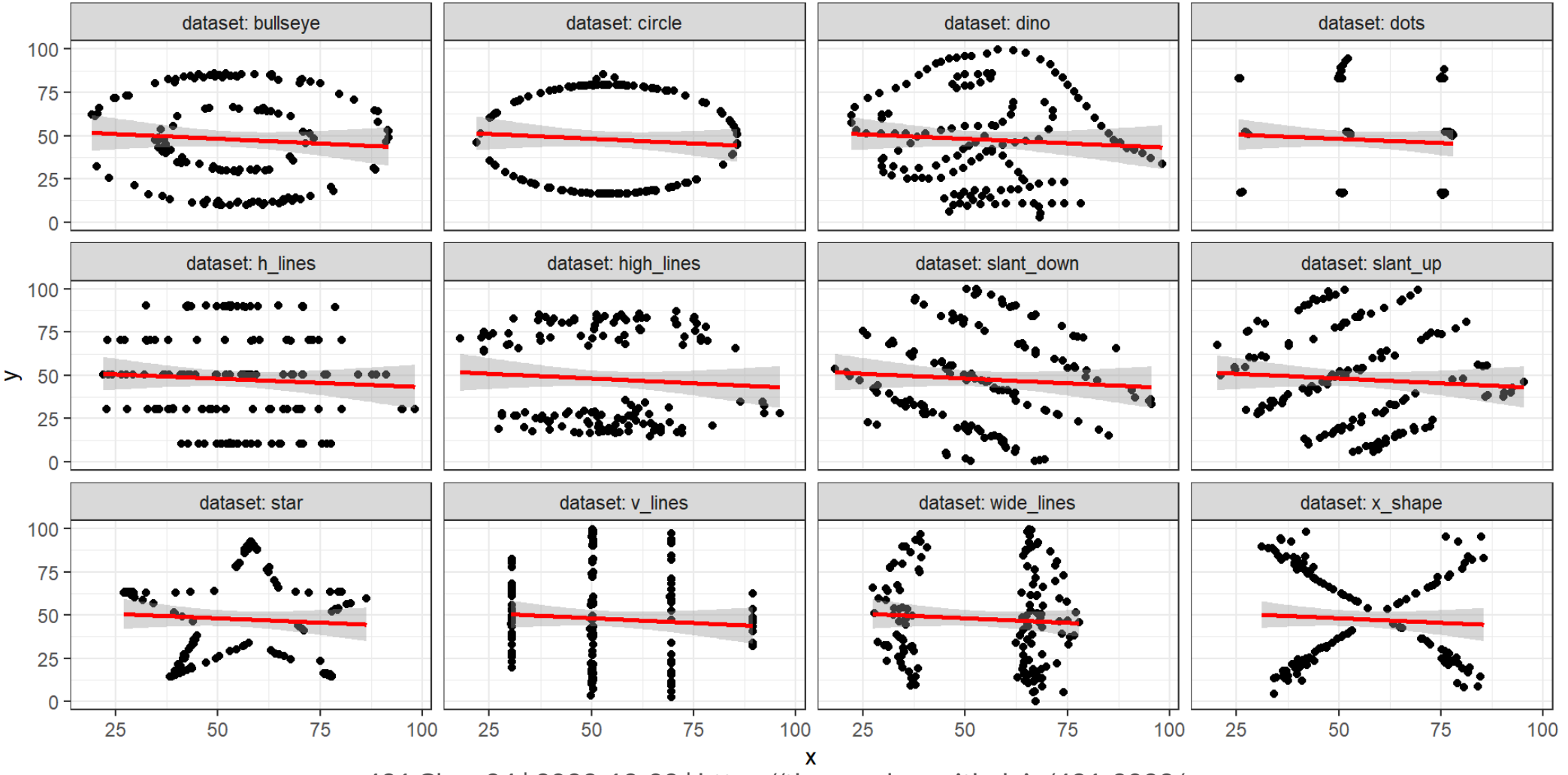
The Other 12 Data Sets



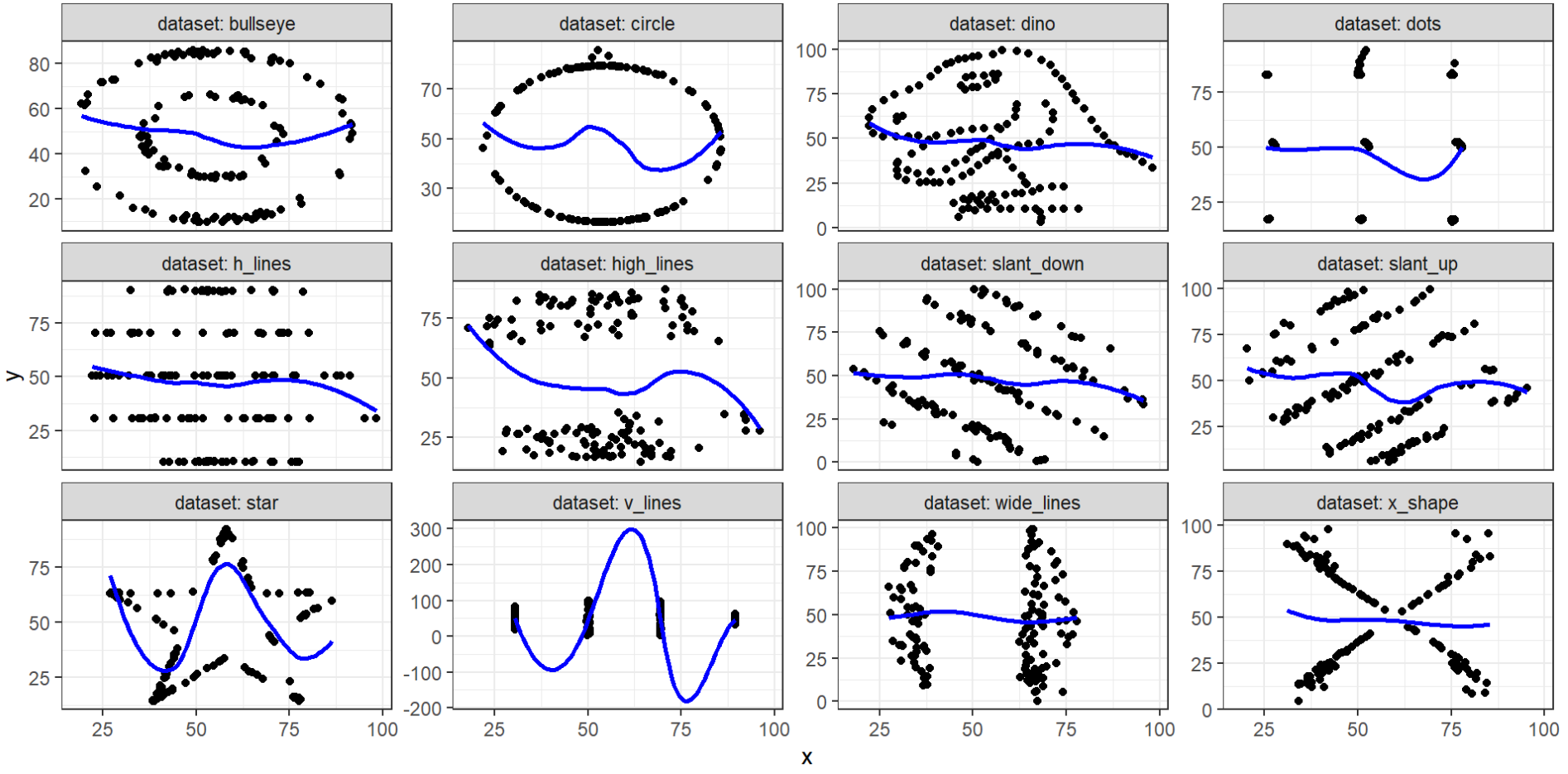
Actually, each set has a name



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/

```
library(datasauRus)
library(ggplot2)
library(gganimate)

ggplot(datasaurus_dozen, aes(x=x, y=y))+
  geom_point()+
  theme_minimal() +
  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.

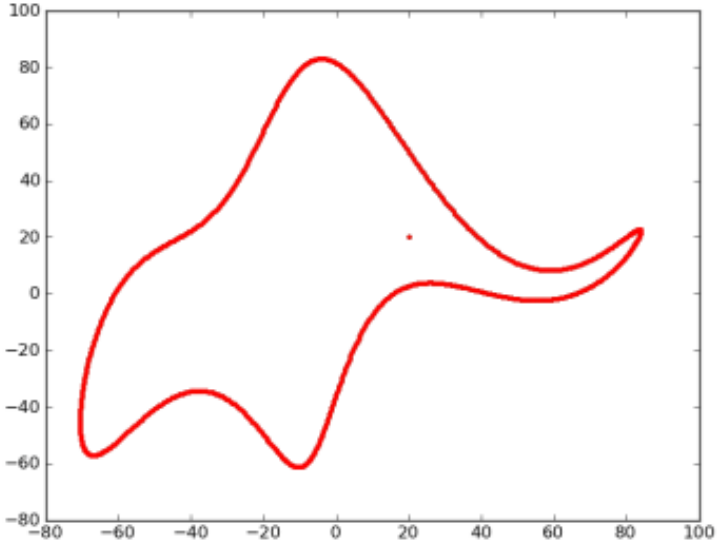
We'll look at a couple of the animated plots they generate there.

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

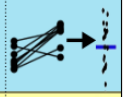
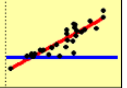
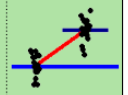
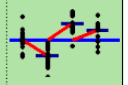
https://lindeloev.github.io/tests-as-linear/

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: lm(y ~ 1 + x)	**y is independent of x** P: One-sample t-test N: Wilcoxon signed-rank	t.test(y) wilcox.test(y)	lm(y ~ 1) lm(signed_rank(y) ~ 1)	✓ for N >14	One number (intercept, i.e., the mean) predicts **y**. - (Same, but it predicts the *signed rank* of **y**.)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	t.test(y_1, y_2, paired=TRUE) wilcox.test(y_1, y_2, paired=TRUE)	lm(y_2 - y_1 ~ 1) lm(signed_rank(y_2 - y_1) ~ 1)	✓ for N >14	One intercept predicts the pairwise y_2-y_1 differences. - (Same, but it predicts the *signed rank* of y_2-y_1.)	
	y ~ continuous x P: Pearson correlation N: Spearman correlation	cor.test(x, y, method='Pearson') cor.test(x, y, method='Spearman')	lm(y ~ 1 + x) lm(rank(y) ~ 1 + rank(x))	✓ for N >10	One intercept plus **x** multiplied by a number (slope) predicts **y**. - (Same, but with *ranked* **x** and **y**)	
	y ~ discrete x P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	t.test(y_1, y_2, var.equal=TRUE) t.test(y_1, y_2, var.equal=FALSE) wilcox.test(y_1, y_2)	lm(y ~ 1 + G_2)[A] gls(y ~ 1 + G_2, weights=…[B])[A] lm(signed_rank(y) ~ 1 + G_2)[A]	✓ ✓ for N >11	An intercept for **group 1** (plus a difference if **group 2**) predicts **y**. - (Same, but with one variance *per group* instead of one common.) - (Same, but it predicts the *signed rank* of **y**.)	
Multiple regression: lm(y ~ 1 + x_1 + x_2 +…)	P: One-way ANOVA N: Kruskal-Wallis	aov(y ~ group) kruskal.test(y ~ group)	lm(y ~ 1 + G_2 + G_3 +…+ G_N)[A] lm(rank(y) ~ 1 + G_2 + G_3 +…+ G_N)[A]	✓ for N >11	An intercept for **group 1** (plus a difference if group ≠ 1) predicts **y**. - (Same, but it predicts the *rank* of **y**.)	
	P: One-way ANCOVA	aov(y ~ group + x)	lm(y ~ 1 + G_2 + G_3 +…+ G_N + x)[A]	✓	- (Same, but plus a slope on **x**.) *Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.*	
	P: Two-way ANOVA	aov(y ~ group * sex)	lm(y ~ 1 + G_2 + G_3 + … + G_N + S_2 + S_3+ … + S_K + G_2*S_2+G_3*S_3+…+G_N*S_K)	✓	Interaction term: changing **sex** changes the **y ~ group** parameters. *Note: $G_{2\ to\ N}$ is an indicator (0 or 1) for each non-intercept levels of the **group** variable. Similarly for $S_{2\ to\ K}$ for sex. The first line (with G_i) is main effect of group, the second (with S_i) for sex and the third is the **group × sex** interaction. For two levels (e.g. male/female), line 2 would just be "S_2" and line 3 would be S_2 multiplied with each G_i.*	*[Coming]*
	Counts ~ discrete x N: Chi-square test	chisq.test(groupXsex_table)	**Equivalent log-linear model** glm(y ~ 1 + G_2 + G_3 + … + G_N + S_2 + S_3+ … + S_K + G_2*S_2+G_3*S_3+…+G_N*S_K, family=…)[A]	✓	Interaction term: (Same as Two-way ANOVA.) *Note: Run glm using the following arguments: `glm(model, family=poisson())`* *As linear-model, the Chi-square test is log(y_i) = log(N) + log(α_i) + log(β_j) + log($\alpha_i\beta_j$) where α_i and β_j are proportions. See more info in the accompanying notebook.*	*Same as Two-way ANOVA*
	N: Goodness of fit	chisq.test(y)	glm(y ~ 1 + G_2 + G_3 +…+ G_N, family=…)[A]	✓	(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 ~ 1 + x is R shorthand for y = 1·b + a·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_i and S_i are "dummy coded" indicator variables (either 0 or 1) exploiting the fact that when Δx = 1 between categories the difference equals the slope. Subscripts (e.g., G_2 or y_1) indicate different columns in data. lm 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.

[A] See the note to the two-way ANOVA for explanation of the notation.
[B] 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 2x2 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 JxK table, where \(2 \leq J \leq 5\) and \(3 \leq K \leq 5\)
  - Linear model, as in the 2x2 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)

```
library(tidymodels)
library(readr)
library(broom.mixed)

urchins <-
  # Data were assembled for a tutorial
  # at https://www.flutterbys.com.au/stats/tut/tut7.5a.html
  read_csv("https://tidymodels.org/start/models/urchins.csv") |>
  # Change the names to be a little more verbose
  setNames(c("food_regime", "initial_volume", "width")) |>
  mutate(food_regime =
           factor(food_regime,
                  levels = c("Initial", "Low", "High")))
```

The `urchins` data

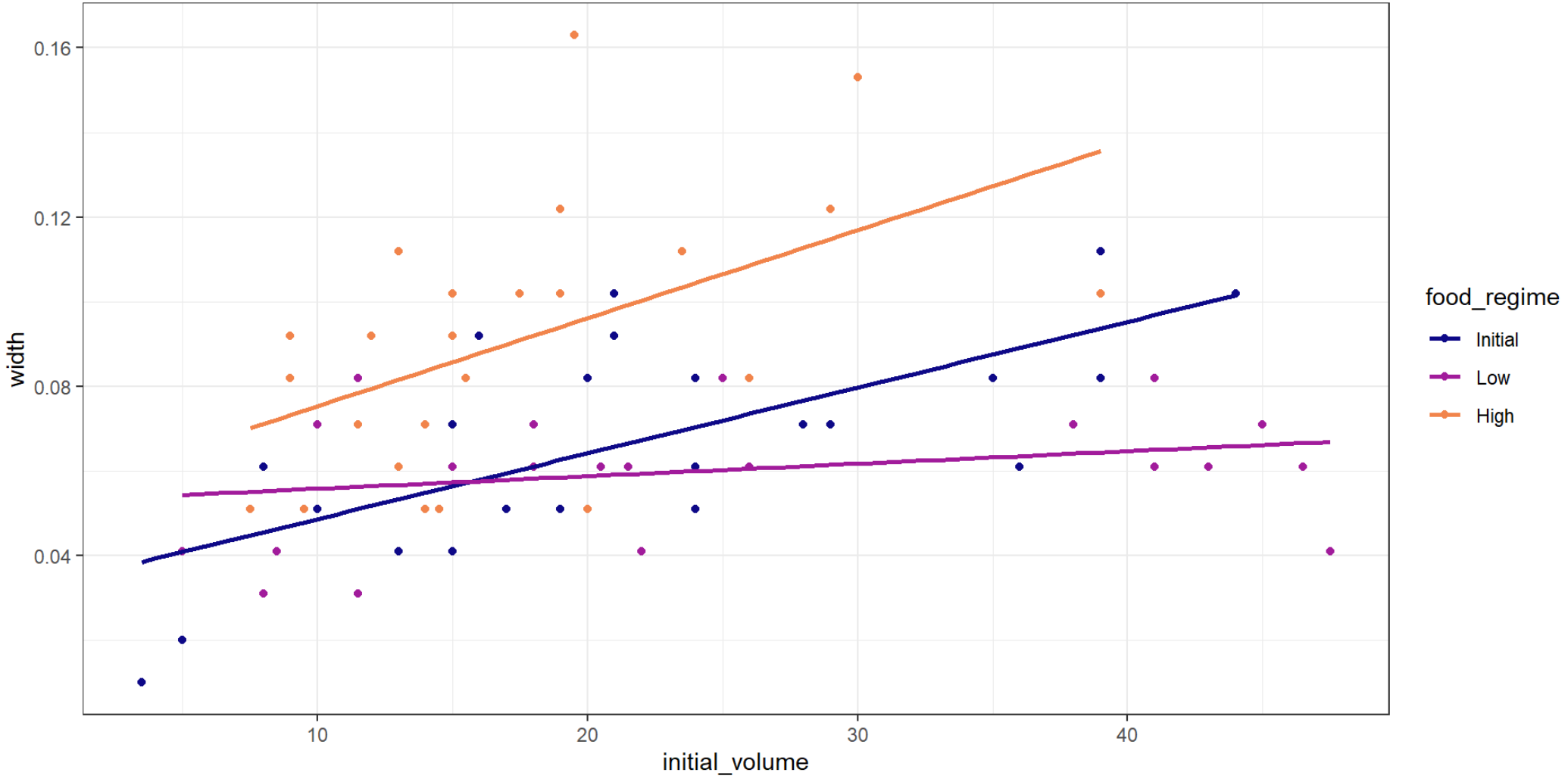
For each of 72 sea urchins, we know their

- 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.

```
lm(width ~ initial_volume * food_regime, data = urchins) |> tidy() |>
  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`

```
lm_mod <-
  linear_reg()   |>
  set_engine("lm")

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()`

```
lm_fit <-
  lm_mod |>
  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`

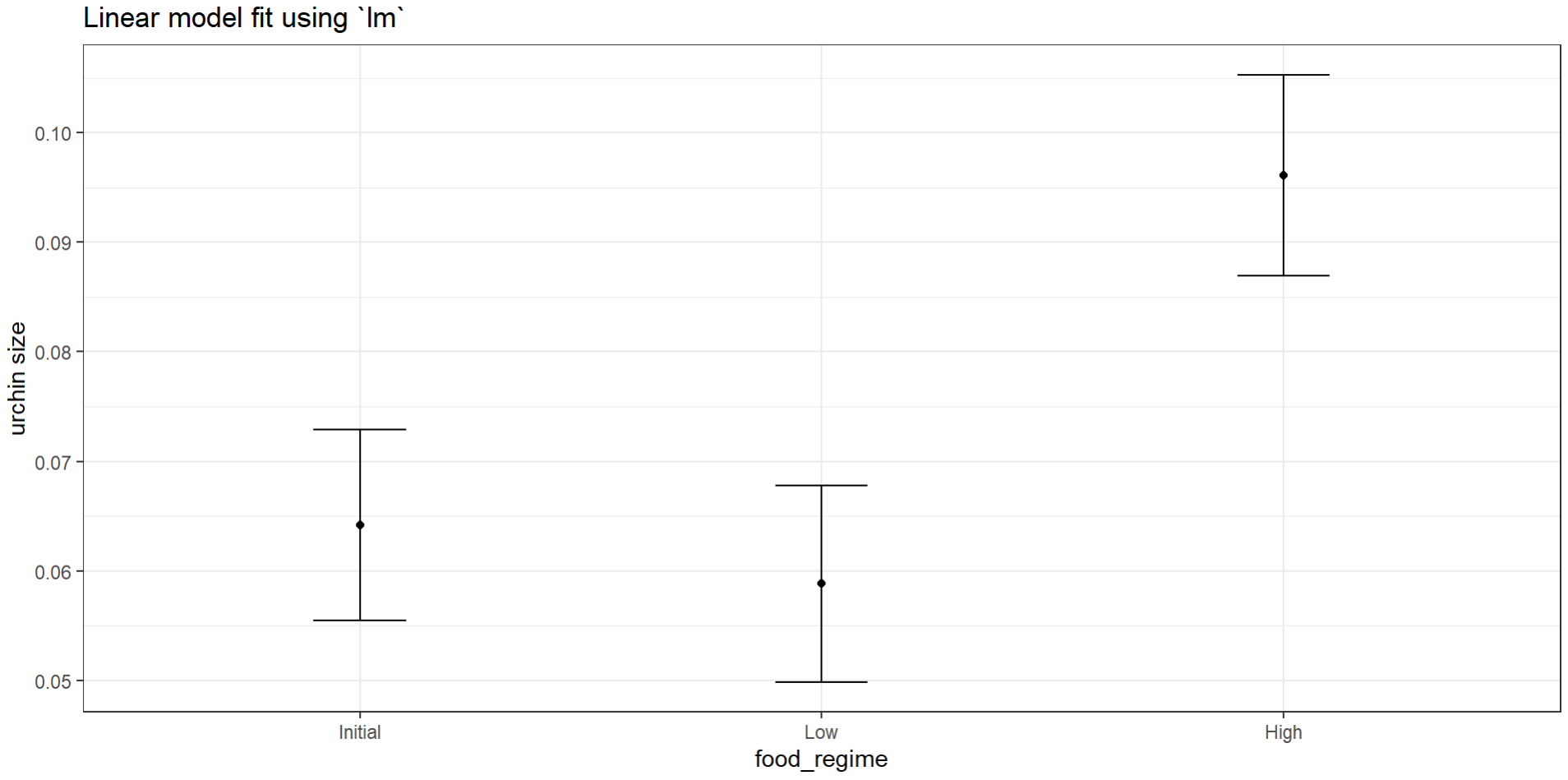
We'll develop mean predictions and uncertainty intervals.

```
mean_pred <- predict(lm_fit, new_data = new_points)
conf_int_pred <- predict(lm_fit,
                         new_data = new_points,
                         type = "conf_int")
plot_data <-
  new_points |>
  bind_cols(mean_pred) |>
  bind_cols(conf_int_pred)
```

Plot the `plot_data` results

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

Plot the `plot_data` results



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

```
prior_dist <- rstanarm::student_t(df = 1)

set.seed(123)

bayes_mod <-
  linear_reg() |>
  set_engine("stan",
             prior_intercept = prior_dist,
             prior = prior_dist)
```

Training the Bayes Model

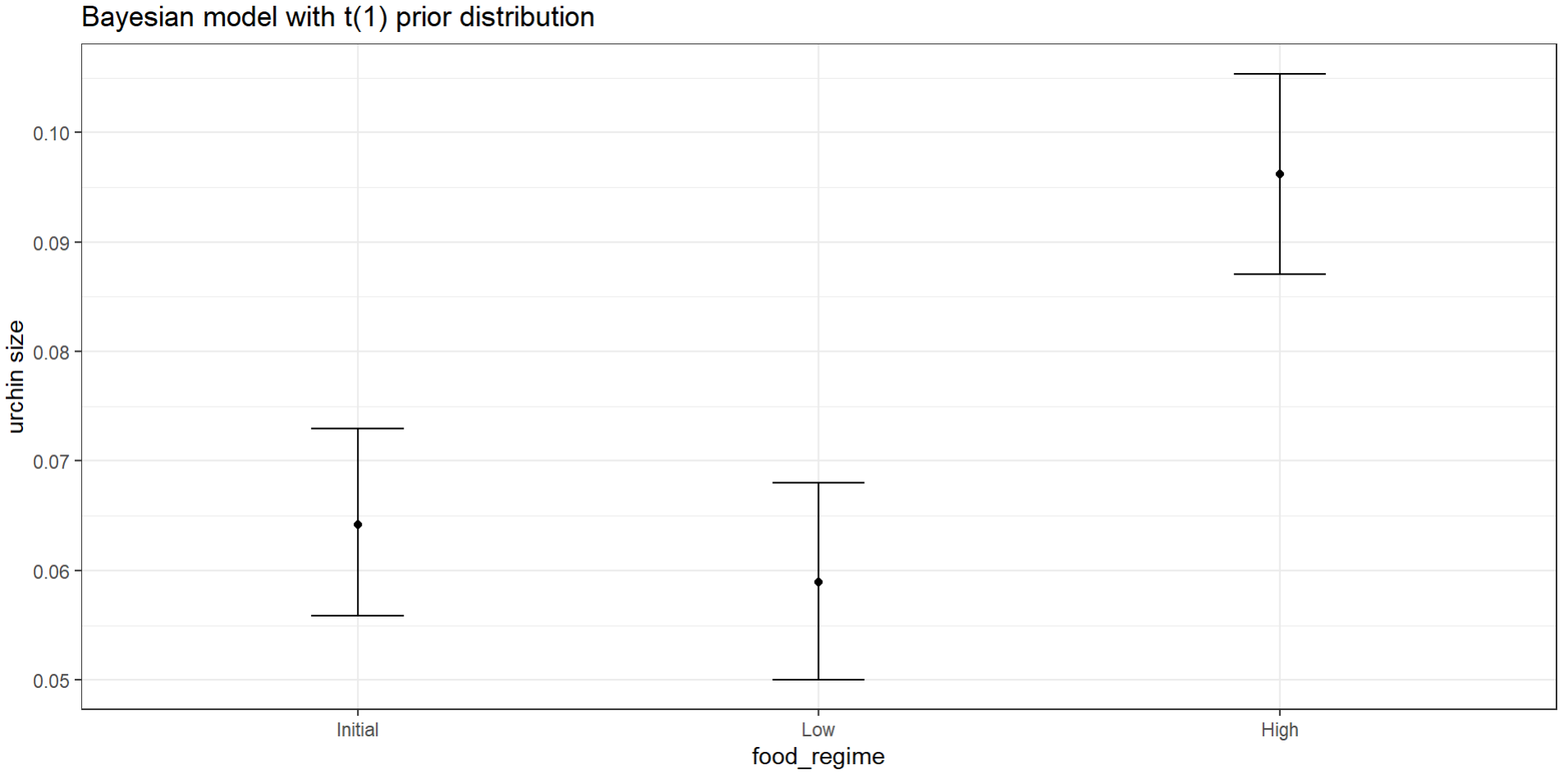
```
bayes_fit <- bayes_mod |>
  fit(width ~ initial_volume * food_regime, data = urchins)

tidy(bayes_fit, conf.int = TRUE) |>
  select(term, estimate, conf.low, conf.high) |>
  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

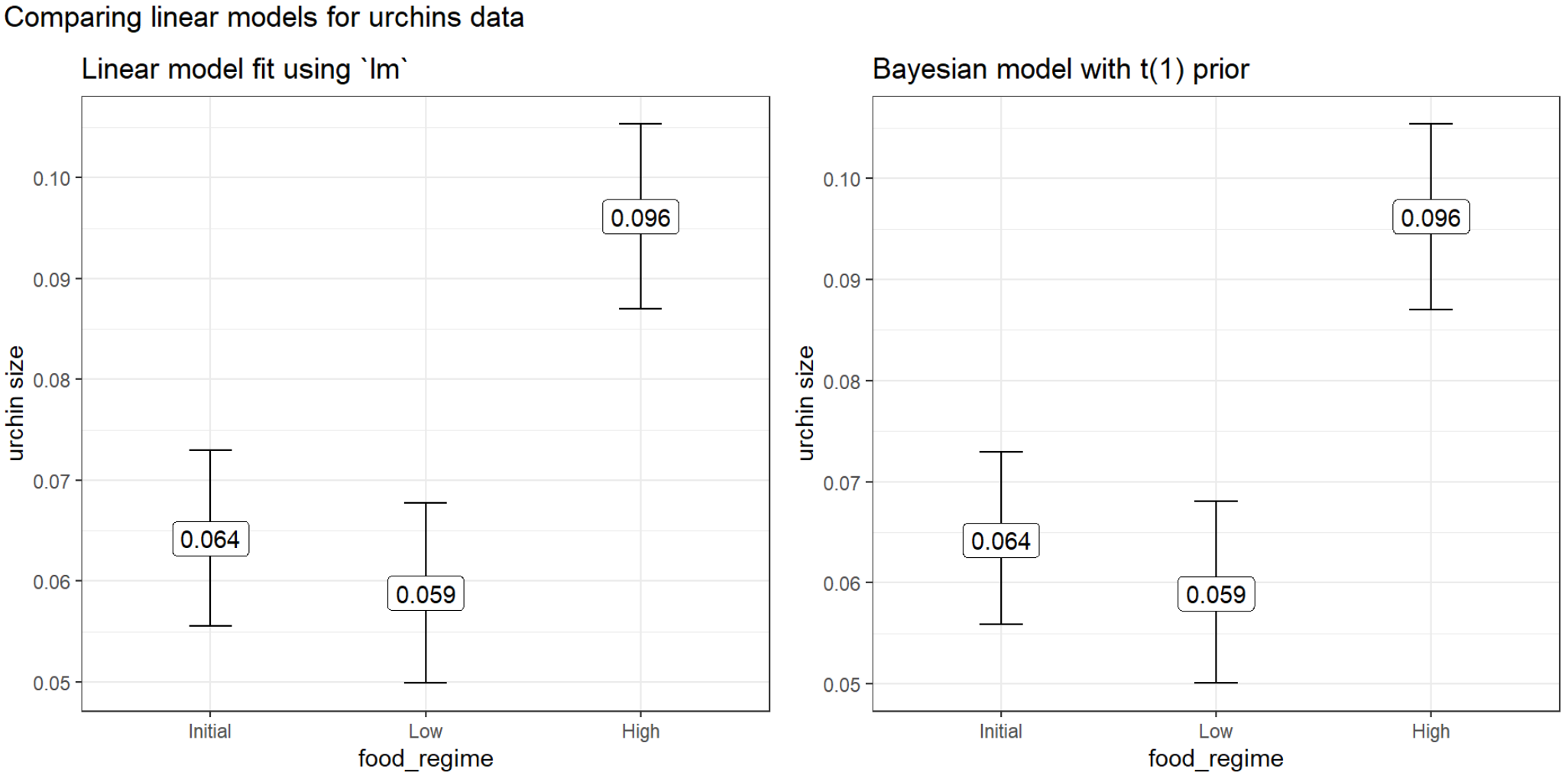
Building the plot for the Bayes model

```
bayes_plot_data <-
  new_points |>
  bind_cols(predict(bayes_fit, new_data = new_points)) |>
  bind_cols(predict(bayes_fit, new_data = new_points,
                    type = "conf_int"))

ggplot(bayes_plot_data, aes(x = food_regime, y = .pred)) +
  geom_point() +
  geom_errorbar(aes(ymin = .pred_lower, ymax = .pred_upper),
                width = .2) +
  labs(y = "urchin size",
       title = "Bayesian model with t(1) prior distribution")
```

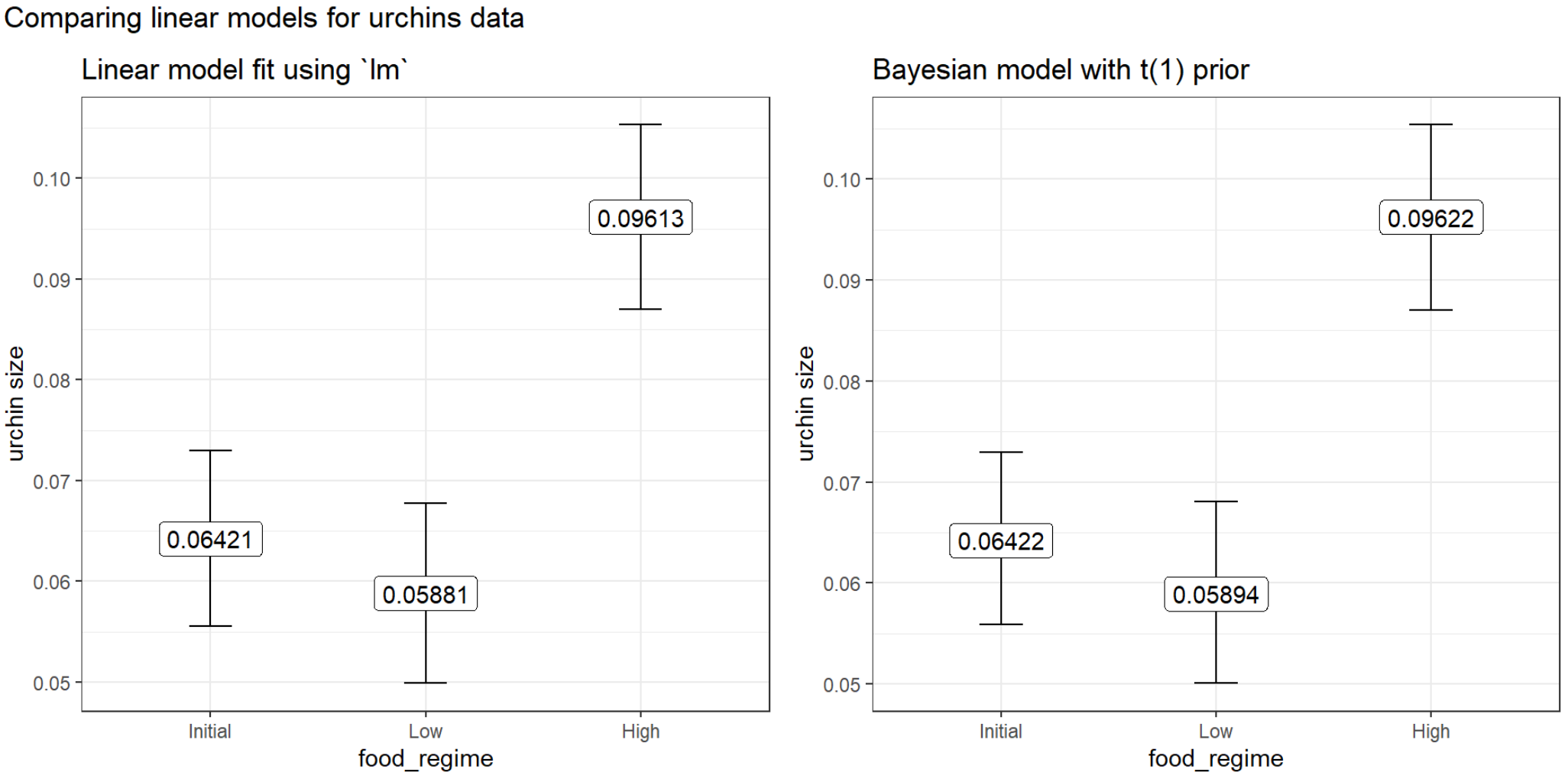
Building the plot for the Bayes model



Comparing the Models



The models aren't actually the same



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.

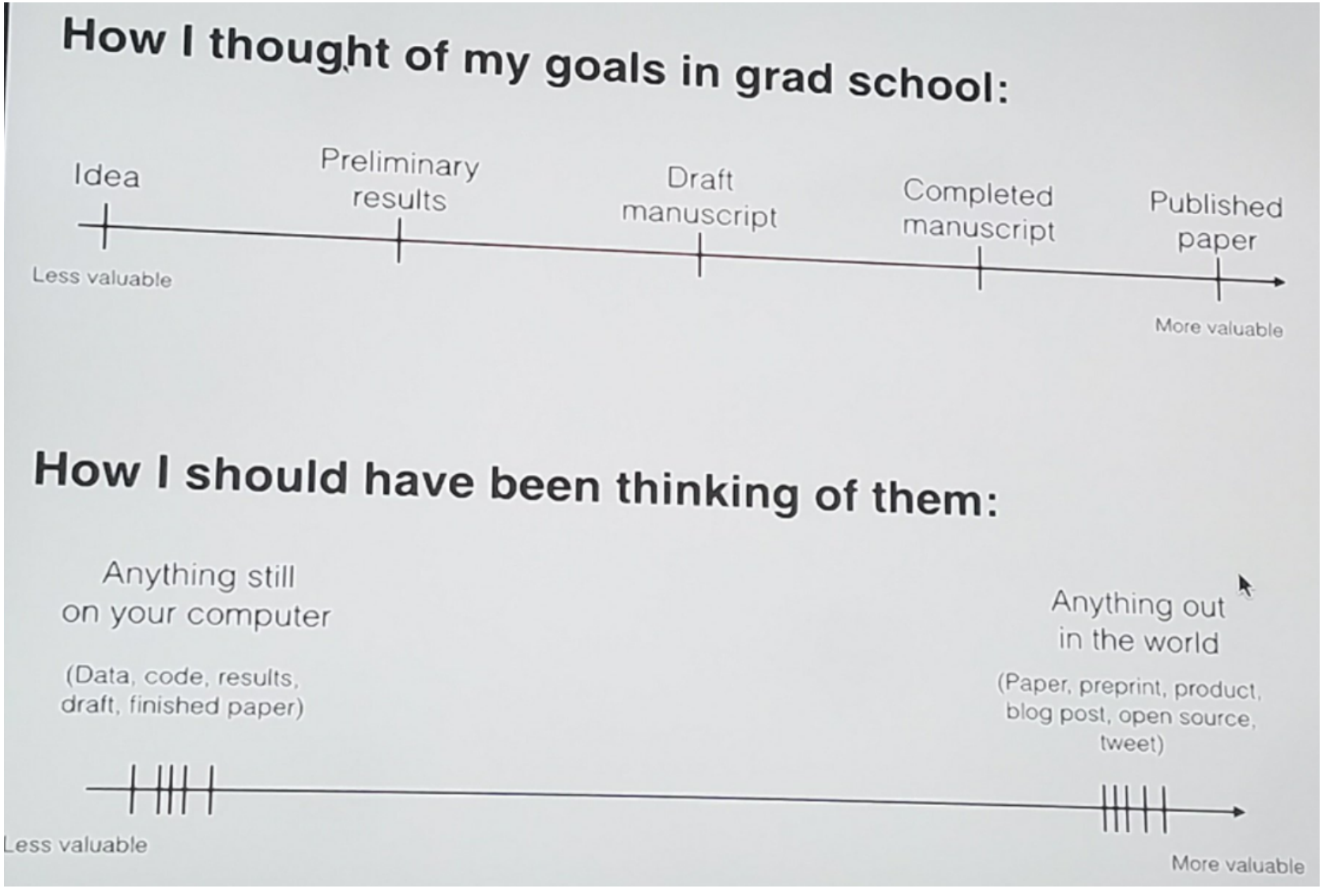
None of this is particularly smart.

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



Ten of the Most Important 431 Ideas

1. You have to visualize and count data to understand it.
2. 90% of statistical work could be described as data management.
3. R Markdown and the tidyverse make it easier to do the right thing.
4. Statistical significance is not a helpful concept.
5. Point estimates and confidence intervals are useful ideas.

Ten of the Most Important 431 Ideas

6. Most statistical procedures are in fact regression models.
7. All statistical methods involve assumptions worth checking.
8. The bootstrap is a very useful, and somewhat underused tool.
9. Prediction models need to predict well in new situations.
10. Statistical thinking is far too important to be left to statisticians.


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
```