432 Class 01

Thomas E. Love, Ph.D.

2026-01-13

Getting To These Slides

Our web site: https://thomaselove.github.io/432-2026/

- Note that this link is posted to the bottom of every slide.

Visit the Calendar at the top of the page, which will take you to the Class 01 README page.

- Slides for Class 01 linked at Class 01 README.
  - We'll look at the **HTML slides** during class.
  - We also provide the Quarto code, and a PDF version. All of the class materials are written in Quarto within RStudio.

Today's Agenda

1. Mechanics of the course
2. Why I write dates the way I do
3. Getting Organized
4. Building and Validating models for Penguin Bill Length
5. Setting Up Lab 1

Appendix in slides works through another linear regression.

Course Mechanics

Welcome to 432.

Just about everything is linked at https://thomaselove.github.io/432-2026

- Calendar
  - final word on all deadlines, and links to each class and TA office hours.
- Syllabus (can download as PDF)
- Course Notes HTML and PDF

Course Notes

- https://thomaselove.github.io/432-notes/ has 34 chapters (> 900 pages) to supplement this course.
- Review of Key Ideas from 431 in Chapters 3-5
  - Chapter 3: Comparing Means
  - Chapter 4: Comparing Rates/Proportions
  - Chapter 5: Fitting Linear Models (where we'll start)
- Walk-through of Two Large Examples (NHANES data in Chapters 1-2, BRFSS SMART data in Chapter 6)

Also linked on our website

- Software
  - Updating / Installing R and RStudio, necessary R Packages
- Get Data (Code, Quarto templates) at our 432-data page
- Assignments (7 Labs, 2 Projects, 2 Quizzes)
- Sources (books, articles, videos, etc.)
- Key Links (Canvas, Shared Google Drive, Minute Papers)
- Contact Us (**431-help at case dot edu** + TA office hours + Me)

Assignments

Every deliverable is listed in the Calendar.

- Welcome to 432 Survey due tomorrow (2026-01-14) at Noon.
- Be sure you see the course in Canvas, and the Shared Google Drive at your CWRU log-in. Thanks.

Assignments include two projects, seven labs, and two quizzes, plus some minute papers. Almost everything due on Wednesdays at noon.

Two Projects

Project A (publicly available data: linear & logistic models)

- You’ll need your data cleaned and in R by 2026-02-01
- Final Portfolio & (recorded) Presentation due 2026-02-25

Project B (use almost any data and build specific models)

1. Proposal is part of Lab 6: 2026-04-01
2. Presentation (in-person or Zoom) in late April
3. Portfolio (prepared using Quarto) due 2026-05-06

Seven Labs

Seven labs, meant to be (generally) shorter than 431 Labs

1. Lab 1 is due Wednesday 2026-01-21 at Noon.
2. Lab 2 is due Wednesday 2026-01-28 at Noon.

Lab 5 is about building or augmenting your website, and can be done now (or at any time), although it’s not due until 2026-03-18.

- Everyone needs to do all seven labs - there is no “skipping” this term.

Two Quizzes

- Quiz 1 due 2026-03-06 (Friday)
- Quiz 2 due 2026-04-24 (Friday)
  - Receive each Quiz one week in advance
  - Mostly multiple choice or short answer, via Google Form.

Syllabus, Lab Instructions provide feedback details and grading approach for the semester.

Getting Help

- 9 teaching assistants volunteering their time to help you.
- TAs hold Zoom Office Hours (every day but Wednesday) starting Friday 2026-01-16.
  - No office hours on MLK Day or during Spring Break.
- Or email us at **431-help** at **case dot edu**.
- I am also available after every class to chat.
- Email me at Thomas dot Love at case dot edu.

We WELCOME questions/comments/corrections/thoughts!

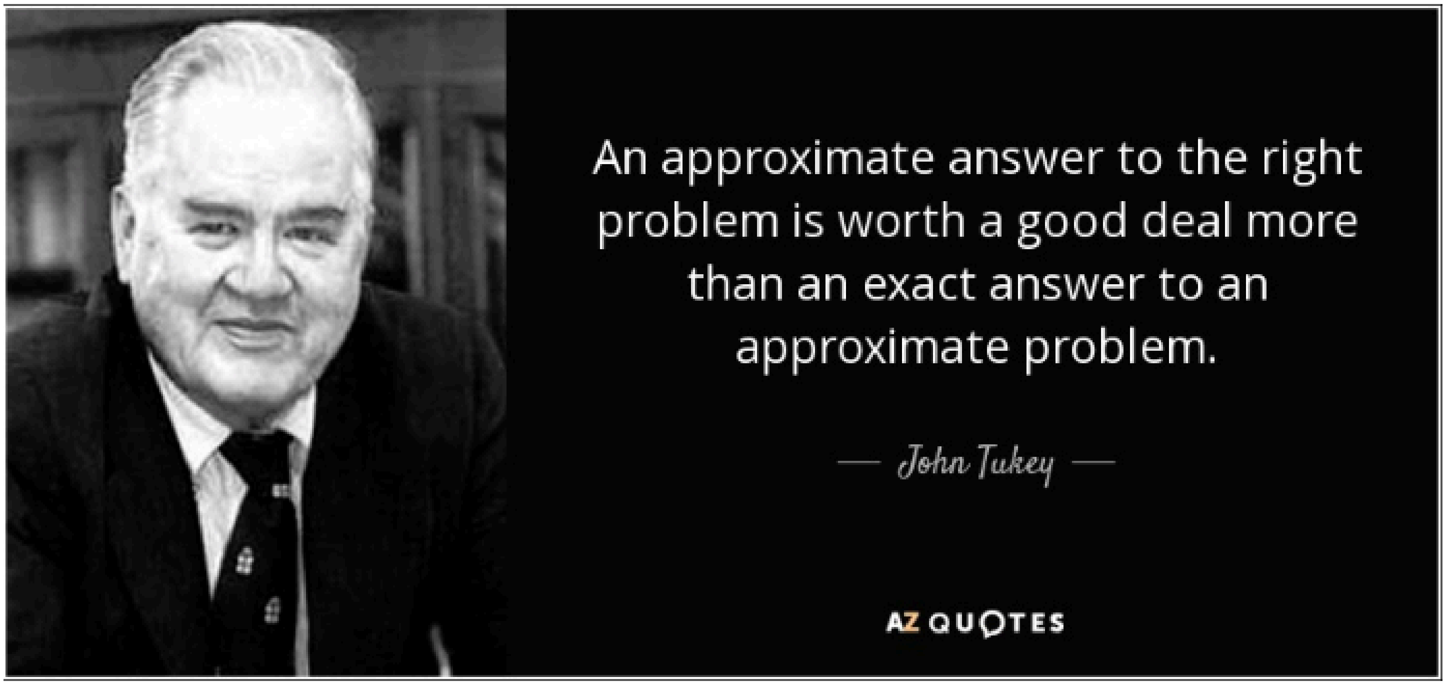
Tools You Will Use in this Class

- **Course Website** (bottom of every slide) especially the Calendar
  - Each class has a README plus slides
- **R, RStudio and Quarto** for, well, everything
- Canvas for access to Zoom meetings and 432 recordings, submission of Labs and Project assignments

Tools You Will Use in this Class

- **Google Drive via CWRU** for forms (Surveys/Quizzes) and for feedback on assignments.
- **Zoom** for class sessions / recordings and TA office hours
- Jeff Leek's 80-page book How To Be A Modern Scientist which you'll finish reading by 2026-02-01.

A few source materials are **password-protected**. What is the password?

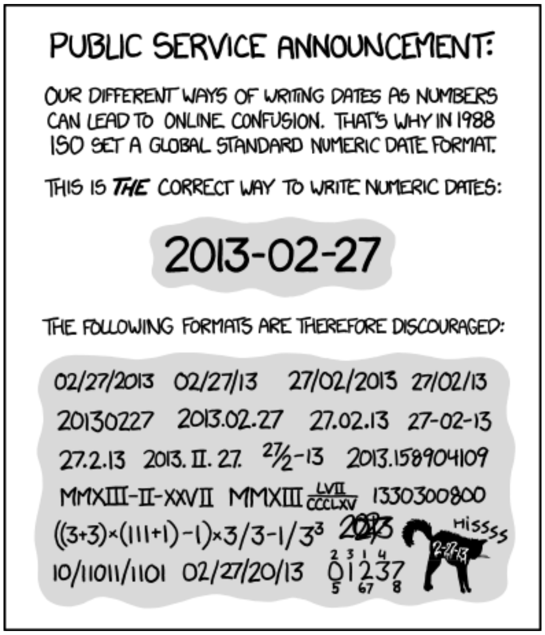


An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.

— John Tukey —

Why I Write Dates The Way I Do

How To Write Dates


PUBLIC SERVICE ANNOUNCEMENT:
OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.
THIS IS THE CORRECT WAY TO WRITE NUMERIC DATES:
2013-02-27
THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:
02/27/2013 02/27/13 27/02/2013 27/02/13
20130227 2013.02.27 27.02.13 27-02-13
27.2.13 2013. II. 27. 27/2-13 2013.158904109
MMXIII-II-XXVII MMXIII LVII/CCCLXV 1330300800
((3+3)×(111+1)-1)×3/3-1/3³ 2013
10/11011/1101 02/27/20/13 01237
HISSsss
2-27-13

Data Organization in Spreadsheets (Broman & Woo)

- Create a data dictionary.
  - Jeff Leek has good thoughts on this in “How to Share Data with a Statistician” at https://github.com/jtleek/datasharing
  - Shannon Ellis and Jeff Leek’s preprint “How to Share data for Collaboration” touches on many of the same points at https://peerj.com/preprints/3139v5.pdf

Sharing Data with a Statistician

We want:

1. The raw data.
2. A tidy data set.
3. A codebook describing each variable and its values in the tidy data set.
4. An explicit and exact recipe describing how you went from 1 to 2 and 3.

Data Organization in Spreadsheets: Be Consistent

- Consistent codes for categorical variables.
  - Either “M” or “Male” but not both at the same time.
  - Make it clear enough to reduce dependence on a codebook.
  - No spaces or special characters other than _ in category names.

Data Organization in Spreadsheets: Be Consistent

- Consistent fixed codes for missing values.
  - NA is the most convenient R choice.
- Consistent variable names
  - In R, I’ll use `clean_names` from the `janitor` package to turn everything into snake_case.
  - In R, start your variable names with letters. No spaces, no special characters other than _.

Data Organization in Spreadsheets: Be Consistent

- Consistent subject / record identifiers
  - And if you’re building a .csv in Excel, don’t use ID as the name of that identifier.
- Consistent data layouts across multiple files.

What Goes in a Cell?

- Make your data a rectangle.
  - Each row represents a record (sometimes a subject).
  - Each column represents a variable.
  - First column is a unique identifier for each record.
- No empty cells.
- One Thing in each cell.
- No calculations in the raw data
- No font colors and no highlighting

Use consistent, strong file names.

Jenny Bryan’s advice on “Naming Things” hold up well. There’s a full presentation at SpeakerDeck.

Good file names:

- are machine readable (easy to search, easy to extract info from names)
- are human readable (name contains content information, so it’s easy to figure out what something is based on its name)

from Jenny Bryan's "Naming Things" slides…

Good file names:

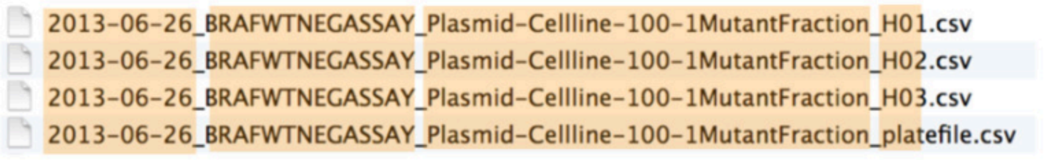
- play well with default ordering (something numeric first, left padded with zeros as needed, use ISO 8601 standard for dates)

Avoid: spaces, punctuation, accented characters, case sensitivity

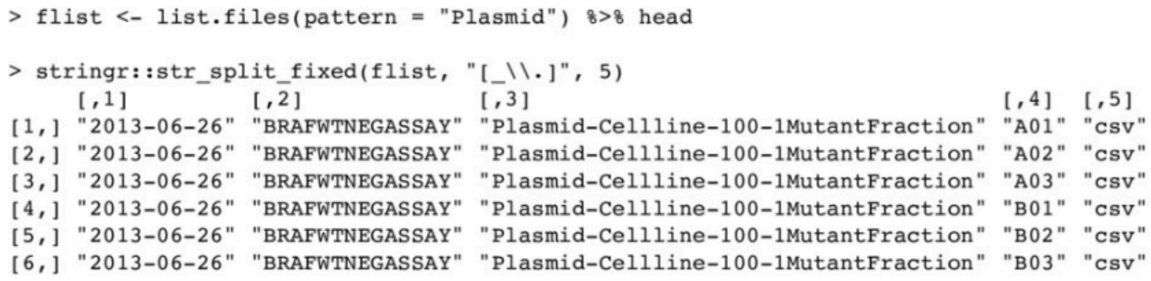
from Jenny Bryan…

Jenny Bryan: Deliberate Use of Delimiters

Deliberately use delimiters to make things easy to compute on and make it easy to recover meta-data from the filenames.



```
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H01.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H02.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H03.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_platefile.csv
```

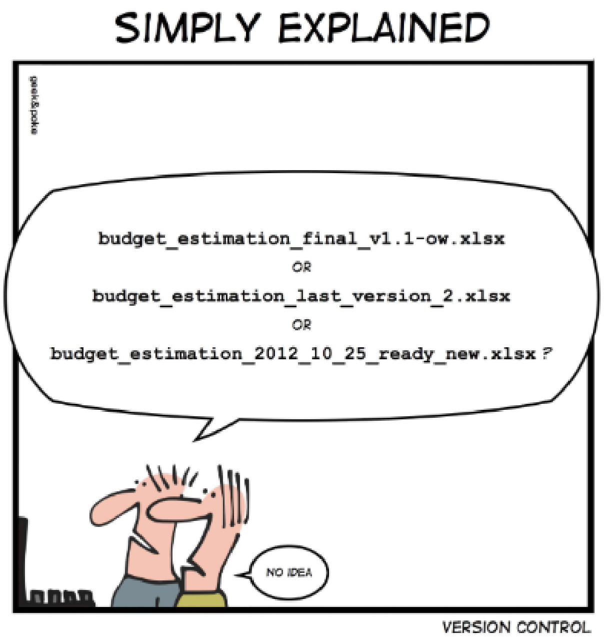


```
> flist <- list.files(pattern = "Plasmid") %>% head

> stringr::str_split_fixed(flist, "[_\\.]", 5)
     [,1]         [,2]            [,3]                                  [,4]  [,5]
[1,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "A01" "csv"
[2,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "A02" "csv"
[3,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "A03" "csv"
[4,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "B01" "csv"
[5,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "B02" "csv"
[6,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "B03" "csv"
```

"_" underscore used to delimit units of meta-data I want later

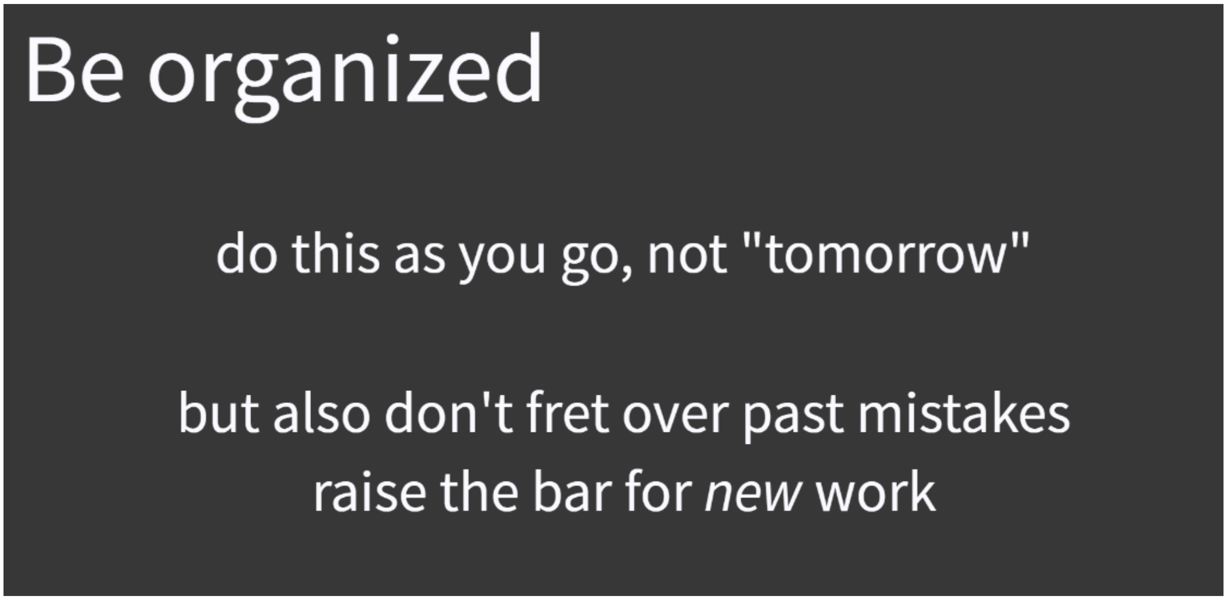
"-" hyphen used to delimit words so my eyes don't bleed

Goal: Avoid this…


SIMPLY EXPLAINED
geek&poke
budget_estimation_final_v1.1-ow.xlsx OR budget_estimation_last_version_2.xlsx OR budget_estimation_2012_10_25_ready_new.xlsx ?
NO IDEA
VERSION CONTROL

Idea from Jen Simmons and John Albin Wilkins during episode #40 of "Web Ahead" about Git: http://5by5.tv/webahead/40

Get organized



Be organized

do this as you go, not "tomorrow"

but also don't fret over past mistakes
raise the bar for *new* work

Don't spend a lot of time bemoaning or cleaning up past ills. Strive to improve this sort of thing going forward.

“Good Enough Practices”

1. Save the raw data.
2. Ensure that raw data is backed up more than once.
3. Create the data you wish to see in the world (the data you wish you had received.)
4. Create analysis-friendly, tidy data.
5. Record all of the steps used to process data.
6. Anticipate the need for multiple tables, and use a unique identifier for every record.

Building and Validating Linear Prediction Models

R Setup

```r
knitr::opts_chunk$set(comment = NA)

library(janitor)

library(broom); library(car); library(GGally); library(glue)
library(gt); library(kableExtra); library(knitr); library(mosaic)
library(patchwork); library(rsample); library(palmerpenguins)

library(easystats)
library(tidyverse)

theme_set(theme_bw())
```

Data Load

```r
our_tibble <- penguins |>
  select(species, sex, bill_length_mm) |>
  drop_na()

our_tibble |> summary()
```

```
      species         sex      bill_length_mm
 Adelie   :146   female:165   Min.   :32.10
 Chinstrap: 68   male  :168   1st Qu.:39.50
 Gentoo   :119                Median :44.50
                              Mean   :43.99
                              3rd Qu.:48.60
                              Max.   :59.60
```

- We’re going to try to predict bill length using species and sex as predictors.

Partition our_tibble into training/test samples

We will place 60% of the penguins in our training sample, and require that similar fractions of each species occur in our training and testing samples. We use functions from the **rsample** package here.

```
set.seed(20260113)
our_split <- initial_split(our_tibble, prop = 0.6, strata = species)
our_train <- training(our_split)
our_test <- testing(our_split)
```

We could have used slice_sample() as in the Course Notes, too.

Result of our partitioning

```
our_train |> tabyl(species) |> adorn_totals() |>
  adorn_pct_formatting()
```

```
   species   n percent
    Adelie  87   43.9%
 Chinstrap  40   20.2%
    Gentoo  71   35.9%
     Total 198  100.0%
```

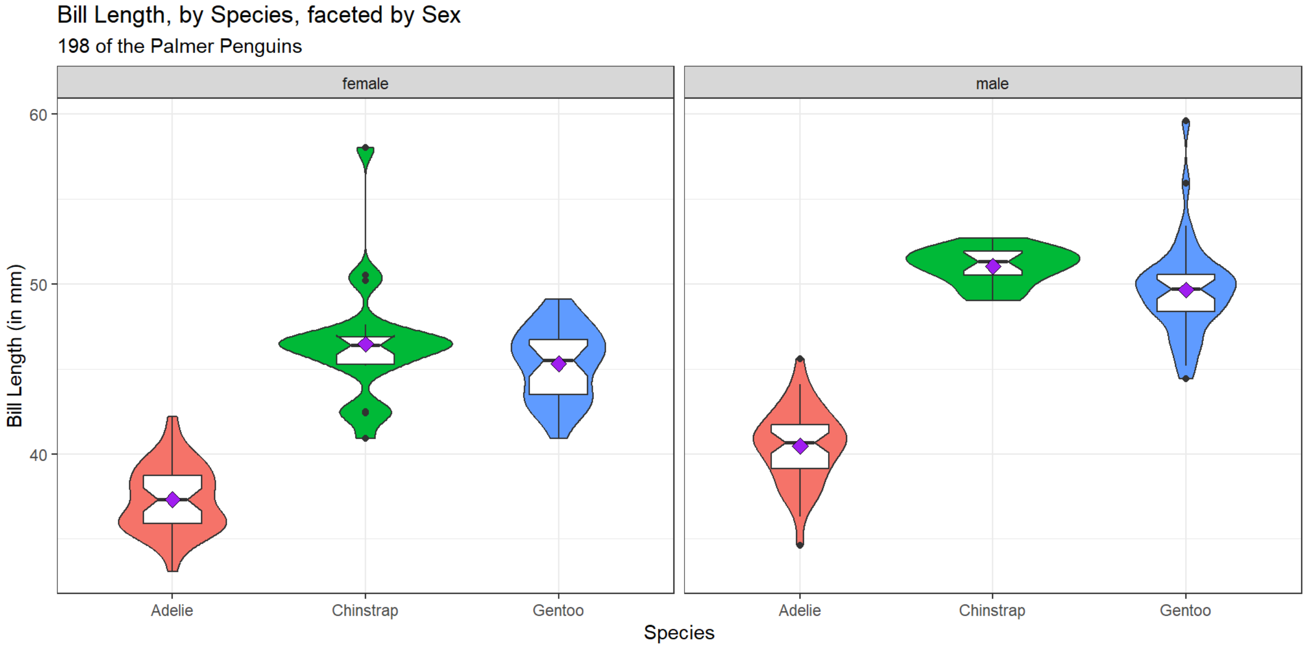
```
our_test |> tabyl(species) |> adorn_totals() |>
  adorn_pct_formatting()
```

```
   species   n percent
    Adelie  59   43.7%
 Chinstrap  28   20.7%
    Gentoo  48   35.6%
     Total 135  100.0%
```

What will this produce?

```r
ggplot(data = our_train, aes(x = species, y = bill_length_mm)) +
  geom_violin(aes(fill = species)) +
  geom_boxplot(width = 0.3, notch = TRUE) +
  stat_summary(fill = "purple", fun = "mean", geom = "point",
               shape = 23, size = 3) +
  facet_wrap(~ sex) +
  guides(fill = "none") +
  labs(title = "Bill Length, by Species, faceted by Sex",
       subtitle = glue(nrow(our_train), " of the Palmer Penguins"),
       x = "Species", y = "Bill Length (in mm)")
```

What will this produce?



Standing Break

Model m1

```r
m1 <- lm(bill_length_mm ~ species + sex, data = our_train)

anova(m1)
```

```
Analysis of Variance Table

Response: bill_length_mm
           Df Sum Sq Mean Sq F value    Pr(>F)    
species     2 4070.3 2035.16  365.42 < 2.2e-16 ***
sex         1  728.6  728.62  130.83 < 2.2e-16 ***
Residuals 194 1080.5    5.57                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 1 coefficients

```
tidy(m1, conf.int = TRUE, conf.level = 0.90) |>
  select(term, estimate, conf.low, conf.high)
```

```
# A tibble: 4 × 4
  term             estimate conf.low conf.high
  <chr>               <dbl>    <dbl>     <dbl>
1 (Intercept)         37.0     36.5      37.5 
2 speciesChinstrap     9.83     9.08     10.6 
3 speciesGentoo        8.62     7.99      9.24
4 sexmale              3.85     3.30      4.41
```

```
model_parameters(m1, ci = 0.90)
```

```
Parameter           | Coefficient |   SE |         90% CI | t(194) |      p
---------------------------------------------------------------------------
(Intercept)         |       36.96 | 0.30 | [36.45, 37.46] | 121.18 | < .001
species [Chinstrap] |        9.83 | 0.45 | [ 9.08, 10.57] |  21.77 | < .001
species [Gentoo]    |        8.62 | 0.38 | [ 7.99,  9.24] |  22.81 | < .001
sex [male]          |        3.85 | 0.34 | [ 3.30,  4.41] |  11.44 | < .001
```

Interlude (Four Ways to Display Tables in Slides)

Model 1 Parameters (version 1)

```
tidy(m1, conf.int = TRUE, conf.level = 0.90) |>
  select(term, estimate, conf.low, conf.high)
```

```
# A tibble: 4 × 4
  term             estimate conf.low conf.high
  <chr>               <dbl>    <dbl>     <dbl>
1 (Intercept)         37.0     36.5      37.5
2 speciesChinstrap     9.83     9.08     10.6
3 speciesGentoo        8.62     7.99      9.24
4 sexmale              3.85     3.30      4.41
```

```
model_parameters(m1, ci = 0.90)
```

```
Parameter            | Coefficient |   SE |         90% CI | t(194) |      p
----------------------------------------------------------------------------
(Intercept)          |       36.96 | 0.30 | [36.45, 37.46] | 121.18 | < .001
species [Chinstrap]  |        9.83 | 0.45 | [ 9.08, 10.57] |  21.77 | < .001
species [Gentoo]     |        8.62 | 0.38 | [ 7.99,  9.24] |  22.81 | < .001
sex [male]           |        3.85 | 0.34 | [ 3.30,  4.41] |  11.44 | < .001
```

Model 1 Parameters (version 2a)

```
tidy(m1, conf.int = TRUE, conf.level = 0.90) |>
  select(term, estimate, conf.low, conf.high) |>
  gt() |> fmt_number(decimals = 2) |>
  tab_options(table.font.size = 24)
```

term	estimate	conf.low	conf.high
(Intercept)	36.96	36.45	37.46
speciesChinstrap	9.83	9.08	10.57
speciesGentoo	8.62	7.99	9.24
sexmale	3.85	3.30	4.41

Model 1 Parameters (version 2b)

```r
model_parameters(m1, ci = 0.90) |>
  gt() |> fmt_number(columns = Coefficient:t, decimals = 2) |>
  tab_options(table.font.size = 24)
```

Parameter	Coefficient	SE	CI	CI_low	CI_high	t	df_error	p
(Intercept)	36.96	0.30	0.90	36.45	37.46	121.18	194	8.692751e-185
speciesChinstrap	9.83	0.45	0.90	9.08	10.57	21.77	194	5.435642e-54
speciesGentoo	8.62	0.38	0.90	7.99	9.24	22.81	194	8.469781e-57
sexmale	3.85	0.34	0.90	3.30	4.41	11.44	194	1.731114e-23

Model 1 Parameters (version 3)

```r
tidy(m1, conf.int = TRUE, conf.level = 0.90) |>
  select(term, estimate, conf.low, conf.high) |>
  kable(digits = 2) |> kable_styling(font_size = 24)
```

term	estimate	conf.low	conf.high
(Intercept)	36.96	36.45	37.46
speciesChinstrap	9.83	9.08	10.57
speciesGentoo	8.62	7.99	9.24
sexmale	3.85	3.30	4.41

```r
model_parameters(m1, ci = 0.90) |>
  kable(digits = 2) |> kable_styling(font_size = 24)
```

Parameter	Coefficient	SE	CI	CI_low	CI_high	t	df_error	p
(Intercept)	36.96	0.30	0.9	36.45	37.46	121.18	194	0
speciesChinstrap	9.83	0.45	0.9	9.08	10.57	21.77	194	0
speciesGentoo	8.62	0.38	0.9	7.99	9.24	22.81	194	0
sexmale	3.85	0.34	0.9	3.30	4.41	11.44	194	0

Model 1 Parameters (version 4)

```r
tidy(m1, conf.int = TRUE, conf.level = 0.90) |>
  select(term, estimate, conf.low, conf.high) |>
  print_html(digits = 2)
```

term	estimate	conf.low	conf.high
(Intercept)	36.96	36.45	37.46
speciesChinstrap	9.83	9.08	10.57
speciesGentoo	8.62	7.99	9.24
sexmale	3.85	3.30	4.41

```r
model_parameters(m1, ci = 0.90) |>
  print_html(digits = 2, font_size = "60%")
```

Parameter	Coefficient	SE	90% CI	t(194)	p
(Intercept)	36.96	0.30	(36.45, 37.46)	121.18	< .001
species (Chinstrap)	9.83	0.45	(9.08, 10.57)	21.77	< .001
species (Gentoo)	8.62	0.38	(7.99, 9.24)	22.81	< .001
sex (male)	3.85	0.34	(3.30, 4.41)	11.44	< .001

Model 1 performance

```r
model_performance(m1) |> gt() |>
  fmt_number(decimals = 3) |> tab_options(table.font.size = 24)
```

AIC	AICc	BIC	R2	R2_adjusted	RMSE	Sigma
907.880	908.192	924.321	0.816	0.813	2.336	2.360

```r
glance(m1) |> gt() |>
  fmt_number(columns = -c("df", "df.residual", "nobs"), decimals = 3) |>
  tab_options(table.font.size = 20)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.816	0.813	2.360	287.223	0.000	3	−448.940	907.880	924.321	1,080.453	194	198

Model m2

```r
m2 <- lm(bill_length_mm ~ species, data = our_train)

## anova(m2) yields p-value < 2.2e-16 (not shown here)

tidy(m2, conf.int = TRUE, conf.level = 0.90) |>
  select(term, estimate, conf.low, conf.high) |>
  kable(digits = 1)
```

term	estimate	conf.low	conf.high
(Intercept)	38.9	38.4	39.4
speciesChinstrap	9.6	8.6	10.6
speciesGentoo	8.8	8.0	9.6

Comparison of Coefficients

```r
compare_models(m1, m2)
```

```
Parameter            |                   m1 |                   m2
----------------------------------------------------------------------
(Intercept)          | 36.96 (36.36, 37.56) | 38.91 (38.26, 39.55)
species [Chinstrap]  |  9.83 ( 8.94, 10.72) |  9.61 ( 8.46, 10.76)
species [Gentoo]     |  8.62 ( 7.87,  9.36) |  8.84 ( 7.88,  9.80)
sex [male]           |  3.85 ( 3.19,  4.52) |
----------------------------------------------------------------------
Observations         |                  198 |                  198
```

In-Sample Comparison

```r
bind_rows(glance(m1), glance(m2)) |>
  mutate(model = c("m1", "m2")) |>
  select(model, r2 = r.squared, adjr2 = adj.r.squared,
         AIC, BIC, sigma, nobs) |>
  kable(digits = c(0, 3, 3, 1, 1, 2, 0))
```

model	r2	adjr2	AIC	BIC	sigma	nobs
m1	0.816	0.813	907.9	924.3	2.36	198
m2	0.692	0.689	1007.9	1021.1	3.05	198

Which model has better in-sample performance?

Comparing `m1` vs. `m2` performance

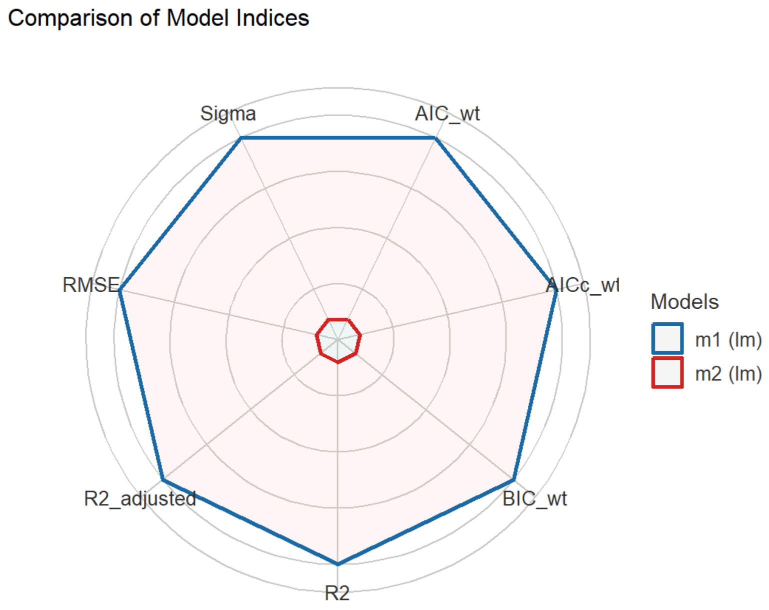
```r
compare_performance(m1, m2)
```

```
# Comparison of Model Performance Indices

Name | Model |   AIC (weights) | AICc (weights) |  BIC (weights) |    R2
-----------------------------------------------------------------------
m1   |    lm |  907.9 (>.999) |  908.2 (>.999) |  924.3 (>.999) | 0.816
m2   |    lm | 1007.9 (<.001) | 1008.1 (<.001) | 1021.1 (<.001) | 0.692

Name | R2 (adj.) |  RMSE | Sigma
--------------------------------
m1   |     0.813 | 2.336 | 2.360
m2   |     0.689 | 3.023 | 3.046
```

Which model has better in-sample performance?

Plot for `m1` vs. `m2` (training)

```r
plot(compare_performance(m1, m2))
```



Assessing Performance in Test Sample

```r
m1_aug <- augment(m1, newdata = our_test)

m1_res <- m1_aug |>
  summarize(val_R_sq = cor(bill_length_mm, .fitted)^2,
            MAPE = mean(abs(.resid)),
            RMSPE = sqrt(mean(.resid^2)),
            max_Error = max(abs(.resid)))

m2_aug <- augment(m2, newdata = our_test)

m2_res <- m2_aug |>
  summarize(val_R_sq = cor(bill_length_mm, .fitted)^2,
            MAPE = mean(abs(.resid)),
            RMSPE = sqrt(mean(.resid^2)),
            max_Error = max(abs(.resid)))
```

Test Sample Performance

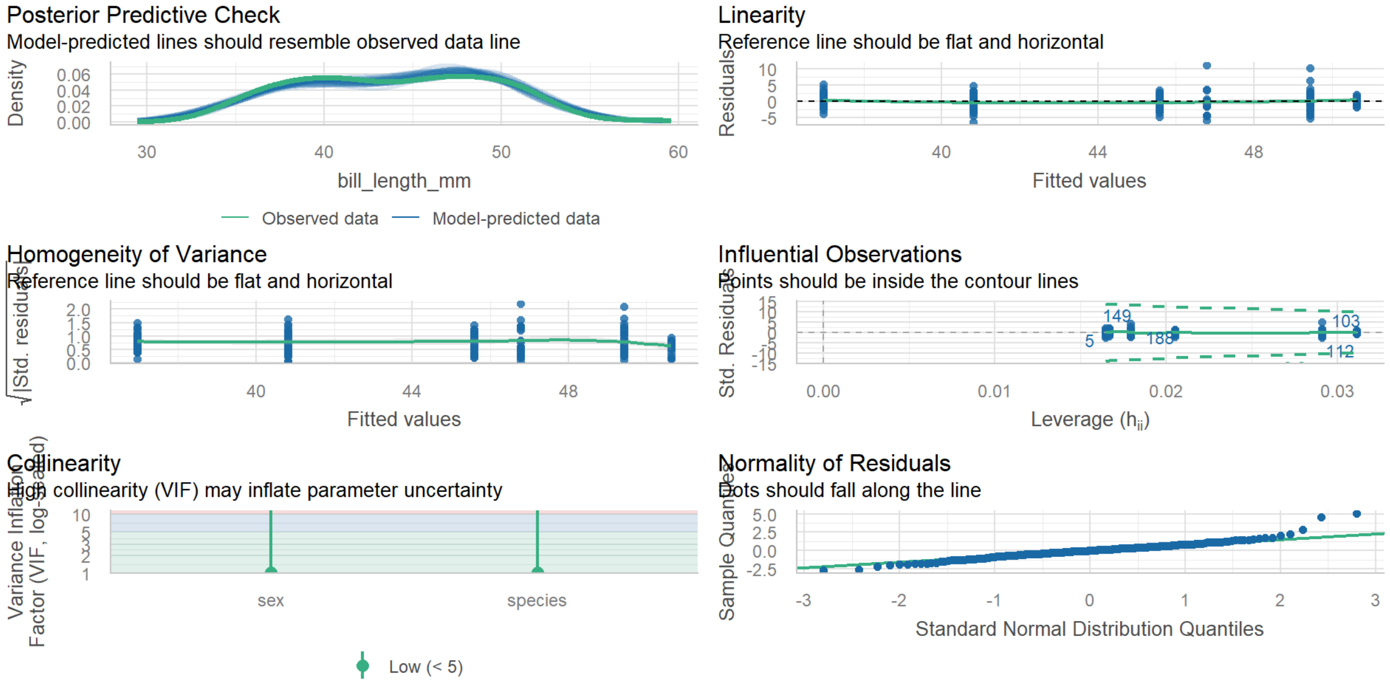
```
bind_rows(m1_res, m2_res) |>
  mutate(model = c("m1", "m2")) |>
  relocate(model) |> kable(digits = c(0, 3, 2, 2, 1))
```

model	val_R_sq	MAPE	RMSPE	max_Error
m1	0.827	1.77	2.28	5.7
m2	0.725	2.34	2.88	7.4

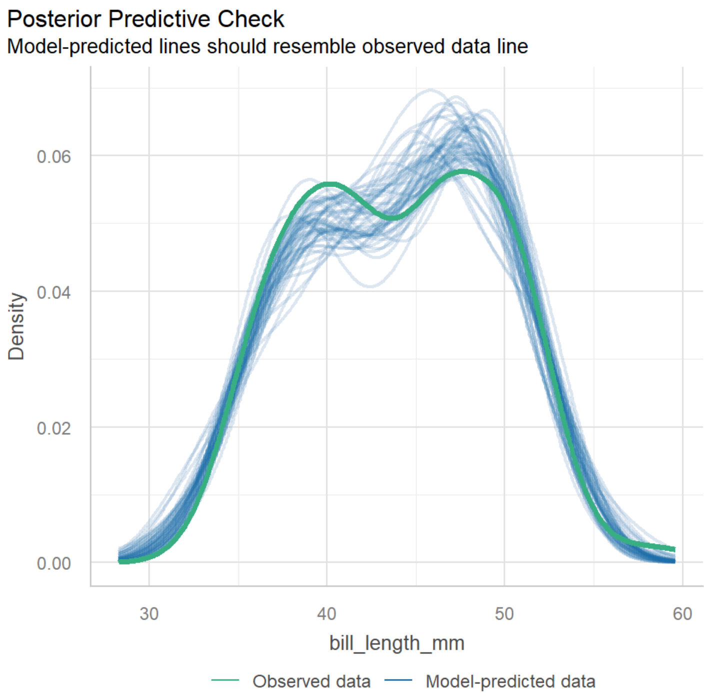
Which model predicts better in the test sample?

Checking m1 (see next 3 slides)

```
check_model(m1, detrend = FALSE)
```

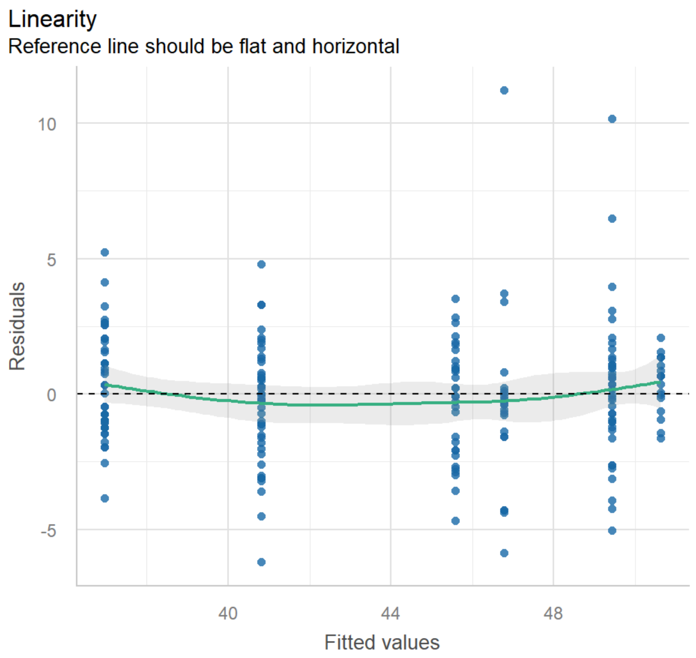


`check_model(m1)`: first 2 plots



Posterior Predictive Check

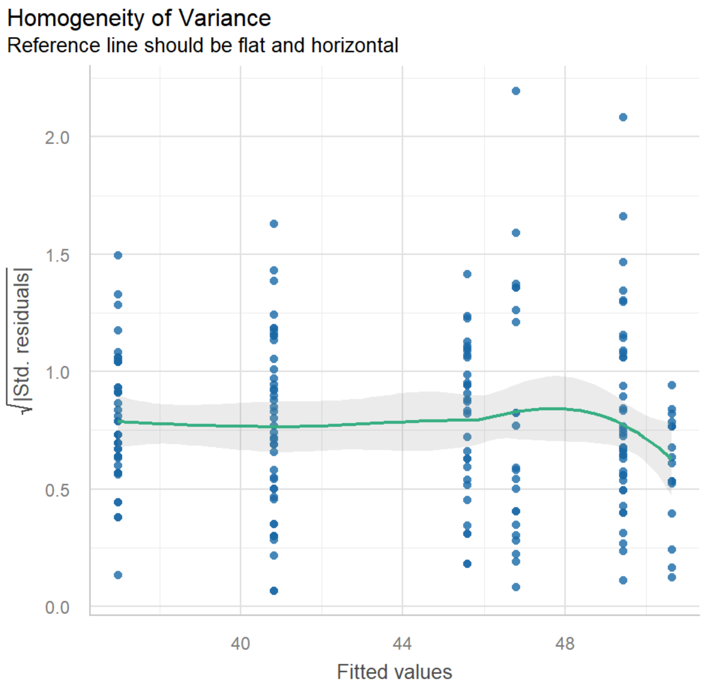
Model-predicted lines should resemble observed data line



Linearity

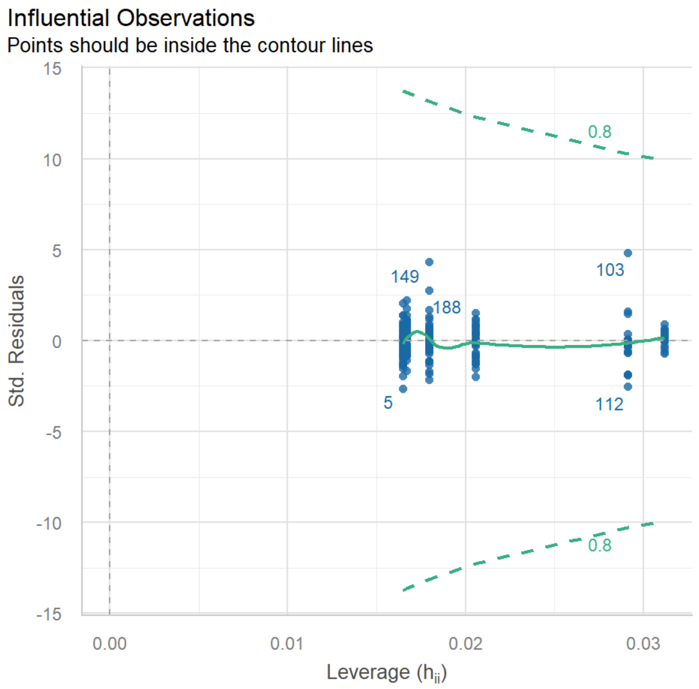
Reference line should be flat and horizontal

`check_model(m1)`: next 2 plots



Homogeneity of Variance

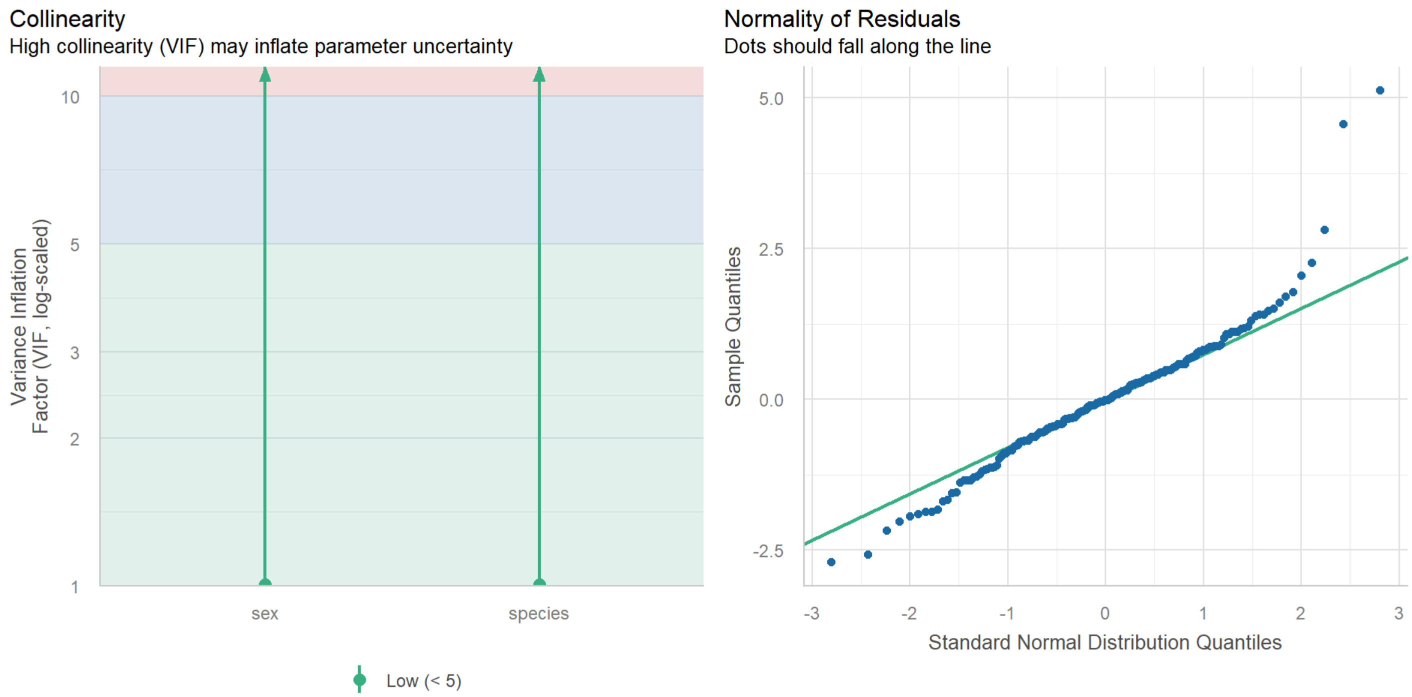
Reference line should be flat and horizontal



Influential Observations

Points should be inside the contour lines

`check_model(m1)`: final 2 plots



What we did in this example…

1. R packages, usual commands, ingest the data.
2. Look at what we have and ensure it makes sense. (DTDP)
3. Partition the data into a training sample and a test sample.
4. Run a two-way ANOVA model (called `m1`) in the training sample; evaluate the quality of fit.
5. Run a one-way ANOVA model (called `m2`) in the training sample; evaluate the quality of fit.

What we did in this example…

6. Use augment to predict from each model into the test sample; summarize and compare predictive quality.
7. Choose between the models and evaluate assumptions for our choice.

Setting Up Lab 1, due 2025-01-22 at Noon

Lab 1 Question 1

I provide some County Health Rankings data for 30 variables and 3054 counties included in the CHR 2024 report. You will filter the data down to the 88 counties in Ohio, and check for missing values.

Then you will create a visualization involving information from three different variables (from a list of 15) using R and Quarto.

There is a Quarto template for Lab 1, in addition to the data set.

Lab 1 Question 2

Create a linear regression model to predict `obesity` as a function of `food_env`, adjusting for `unemployment` (all of these are quantitative variables.)

a. Specify and fit the model, interpret `food_env` coefficient and its confidence interval carefully.

b. Evaluate quality of model in terms of adherence to regression assumptions via `check_model()`.

c. Build a nice table comparing your model to a simple regression for `obesity` using only `food_env`, then reflect on your findings.

For Next Time…

1. If you're not registered with SIS, do so, for PQHS/CRSP/MPHP 432.
2. Check that you see the course on Canvas and that you see the Shared Google Drive when logged into Google via CWRU.
3. Review the website and Calendar, and skim the Syllabus and Course Notes.
4. Welcome to 432 Survey at https://bit.ly/432-2026-welcome-survey by noon tomorrow (2026-01-14.)

For Next Time…

5. Buy Jeff Leek's How to be a Modern Scientist and read it by the end of January.
6. Get started installing or updating the software you need for the course.
7. Get started on Lab 1, due Wednesday 2026-01-21 at Noon.

Appendix: NHANES 1982 Example (Course Notes: Chapters 1-5 provide a very similar example)

Loading the nh1982 R data set

Available at our 432-data page

```
nh1982 <- read_rds("c01/data/nh1982.Rds") |>
  select(SEQN, sbp1, sbp2, sbp3, age, sroh, hospital)

nh1982
```

```
# A tibble: 1,982 × 7
   SEQN    sbp1  sbp2  sbp3   age sroh      hospital
   <chr>  <dbl> <dbl> <dbl> <dbl> <fct>     <fct>   
 1 109266    99    99    99    29 Good      No      
 2 109273   116   110   115    36 Good      No      
 3 109291   107   111   107    42 Fair      Yes     
 4 109297   105   105   102    30 Very Good No      
 5 109315   118   123   125    30 Good      No      
 6 109317   110   110   110    28 Very Good No      
 7 109332   110   105   108    33 Excellent No      
 8 109333   106   107   113    41 Excellent No      
 9 109336   162   148   163    35 Good      No      
10 109373   111   111   113    30 Poor      No      
# ℹ 1,972 more rows
```

2017 - March 2020 NHANES Data

1982 NHANES subjects ages 26-42 with complete data on these variables:

Variable	Source	Description
`SEQN`	P-DEMO	Subject ID: Link (also in BPXO and HUQ)
`age`	P_DEMO	RIDAGEYR (restricted to ages 26-42 here)
`sbp1`	BPXO	BPXOSY1 = 1st Systolic BP reading, in mm Hg
`sbp2`	BPXO	BPXOSY2 = 2nd Systolic BP reading
`sbp3`	BPXO	BPXOSY3 = 3rd Systolic BP reading
`sroh`	HUQ	HUQ010 = five-categories E, VG, G, F, P
`hospital`	HUQ	HUQ071 = Yes or No

Variable Descriptions

Variable	Description (n = 1982)
`SEQN`	Subject identification code from NHANES
`age`	Age in years (range 26-42, mean = 34)
`sbp1`	Systolic Blood Pressure (1st reading)
`sbp2`	Systolic Blood Pressure (2nd reading)
`sbp3`	Systolic Blood Pressure (3rd reading)
`sroh`	Self-reported Overall Health: five categories (see next slide)
`hospital`	Yes if hospitalized in last 12m, else No (8% Yes)

SROH and Hospitalization Status

```
nh1982 |> tabyl(sroh) |> adorn_pct_formatting()
```

```
      sroh   n percent
 Excellent 294   14.8%
 Very Good 598   30.2%
      Good 728   36.7%
      Fair 321   16.2%
      Poor  41    2.1%
```

```
nh1982 |> tabyl(hospital) |> adorn_pct_formatting()
```

```
 hospital    n percent
      Yes  159    8.0%
       No 1823   92.0%
```

Adding mean_sbp to the data

```
nh1982 <- nh1982 |>
  mutate(mean_sbp = (sbp1 + sbp2 + sbp3)/3)

nh1982 |> select(mean_sbp) |> summary()
```

```
    mean_sbp     
 Min.   : 76.33  
 1st Qu.:106.33  
 Median :114.67  
 Mean   :116.06  
 3rd Qu.:124.00  
 Max.   :209.33  
```

```
favstats(nh1982$mean_sbp) |>
  kable(digits = 1) |> kable_styling(font_size = 24)
```

min	Q1	median	Q3	max	mean	sd	n	missing
76.3	106.3	114.7	124	209.3	116.1	14.4	1982	0

data_codebook() results

```r
data_codebook(nh1982 |> select(-SEQN))
```

```
select(nh1982, -SEQN) (1982 rows and 7 variables, 7 shown)

ID | Name      | Type        | Missings |         Values |            N
---+-----------+-------------+----------+----------------+-------------
1  | sbp1      | numeric     | 0 (0.0%) |      [76, 205] |         1982
---+-----------+-------------+----------+----------------+-------------
2  | sbp2      | numeric     | 0 (0.0%) |      [69, 219] |         1982
---+-----------+-------------+----------+----------------+-------------
3  | sbp3      | numeric     | 0 (0.0%) |      [60, 204] |         1982
---+-----------+-------------+----------+----------------+-------------
4  | age       | numeric     | 0 (0.0%) |       [26, 42] |         1982
---+-----------+-------------+----------+----------------+-------------
5  | sroh      | categorical | 0 (0.0%) |      Excellent |   294 (14.8%)
   |           |             |          |      Very Good |   598 (30.2%)
```

We'll fit two models today

1. Predict mean SBP using Age alone.
2. Predict mean SBP (across three readings) using Age, Self-Reported Overall Health and Hospitalization Status.

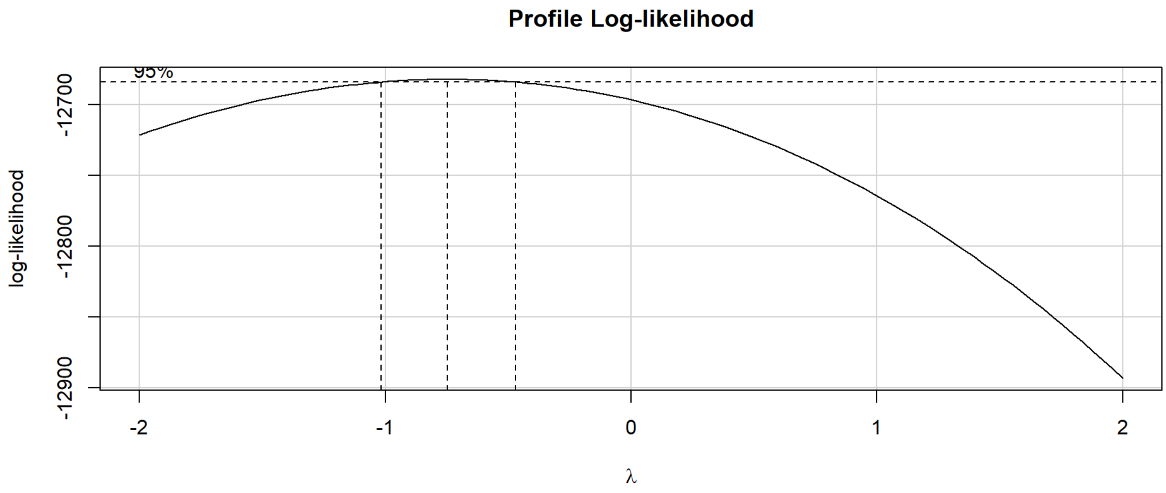
```r
temp_mod1 <- lm(mean_sbp ~ age, data = nh1982)
temp_mod2 <- lm(mean_sbp ~ age + sroh + hospital, data = nh1982)
```

I'm not doing any predictive validation so I won't split the sample.

Box-Cox Plot to suggest potential outcome transformations

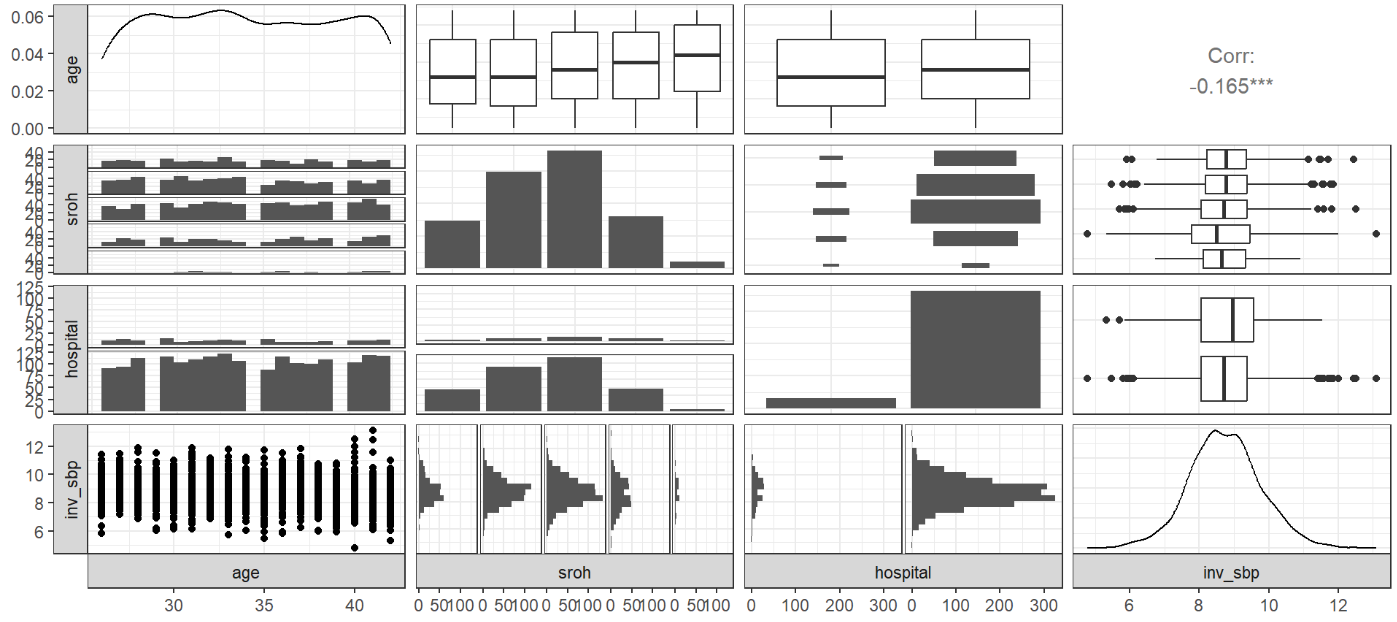
```r
boxCox(temp_mod2)
```

```r
nh1982 <- nh1982 |> mutate(inv_sbp = 1000/mean_sbp)
```



Scatterplot Matrix (from `ggpairs()`)

```r
ggpairs(nh1982, columns = c("age", "sroh", "hospital", "inv_sbp"),
        switch = "both",
        lower=list(combo=wrap("facethist", bins=20)))
```



Variance Inflation Factors

```
car::vif(lm(inv_sbp ~ age + sroh + hospital, data = nh1982))
```

```
             GVIF Df GVIF^(1/(2*Df))
age      1.008723  1        1.004352
sroh     1.020544  4        1.002545
hospital 1.013797  1        1.006875
```

Tidied Coefficients for Model `m1`

```
m1 <- lm(inv_sbp ~ age, data = nh1982)

tidy(m1, conf.int = TRUE, conf.level = 0.9)
```

```
# A tibble: 2 × 7
  term        estimate std.error statistic  p.value conf.low conf.high
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
1 (Intercept)   9.93     0.161       61.5  0           9.66      10.2
2 age          -0.0349   0.00469     -7.44 1.51e-13  -0.0426   -0.0272
```

Model Parameters for `m1`

```
model_parameters(m1, ci = 0.9)
```

```
Parameter   | Coefficient |       SE |          90% CI | t(1980) |      p
-------------------------------------------------------------------------
(Intercept) |        9.93 |     0.16 | [ 9.66, 10.20] |   61.52 | < .001
age         |       -0.03 | 4.69e-03 | [-0.04, -0.03] |   -7.44 | < .001
```

Tidied Coefficients for Model m2

```r
m2 <- lm(inv_sbp ~ age + sroh + hospital, data = nh1982)

tidy(m2, conf.int = TRUE, conf.level = 0.9)
```

```
# A tibble: 7 × 7
  term           estimate std.error statistic  p.value conf.low conf.high
  <chr>             <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
1 (Intercept)     10.0      0.185      54.3    0           9.74    10.3
2 age             -0.0338   0.00470    -7.19   9.27e-13  -0.0415  -0.0260
3 srohVery Good   -0.0552   0.0727     -0.759  4.48e- 1  -0.175    0.0644
4 srohGood        -0.110    0.0705     -1.56   1.20e- 1  -0.226    0.00627
5 srohFair        -0.265    0.0825     -3.21   1.33e- 3  -0.401   -0.129
6 srohPoor        -0.176    0.171      -1.03   3.02e- 1  -0.457    0.105
7 hospitalNo      -0.0464   0.0849     -0.546  5.85e- 1  -0.186    0.0933
```

Model Parameters for m2

```r
model_parameters(m2, ci = 0.9)
```

```
Parameter          | Coefficient |       SE |         90% CI | t(1975) |      p
------------------------------------------------------------------------------
(Intercept)        |       10.04 |     0.18 | [ 9.74, 10.34] |   54.32 | < .001
age                |       -0.03 | 4.70e-03 | [-0.04, -0.03] |   -7.19 | < .001
sroh [Very Good]   |       -0.06 |     0.07 | [-0.17,  0.06] |   -0.76 | 0.448 
sroh [Good]        |       -0.11 |     0.07 | [-0.23,  0.01] |   -1.56 | 0.120 
sroh [Fair]        |       -0.27 |     0.08 | [-0.40, -0.13] |   -3.21 | 0.001 
sroh [Poor]        |       -0.18 |     0.17 | [-0.46,  0.10] |   -1.03 | 0.302 
hospital [No]      |       -0.05 |     0.08 | [-0.19,  0.09] |   -0.55 | 0.585 
```

Compare Coefficients: `m1` and `m2`

```r
compare_models(m1, m2)
```

```
Parameter          |                   m1 |                   m2
----------------------------------------------------------------
(Intercept)        |  9.93 ( 9.61, 10.25) | 10.04 ( 9.68, 10.40)
age                | -0.03 (-0.04, -0.03) | -0.03 (-0.04, -0.02)
sroh [Very Good]   |                      | -0.06 (-0.20,  0.09)
sroh [Good]        |                      | -0.11 (-0.25,  0.03)
sroh [Fair]        |                      | -0.27 (-0.43, -0.10)
sroh [Poor]        |                      | -0.18 (-0.51,  0.16)
hospital [No]      |                      | -0.05 (-0.21,  0.12)
----------------------------------------------------------------
Observations       |                 1982 |                 1982
```

Fit Summaries for Models `m1` and `m2`

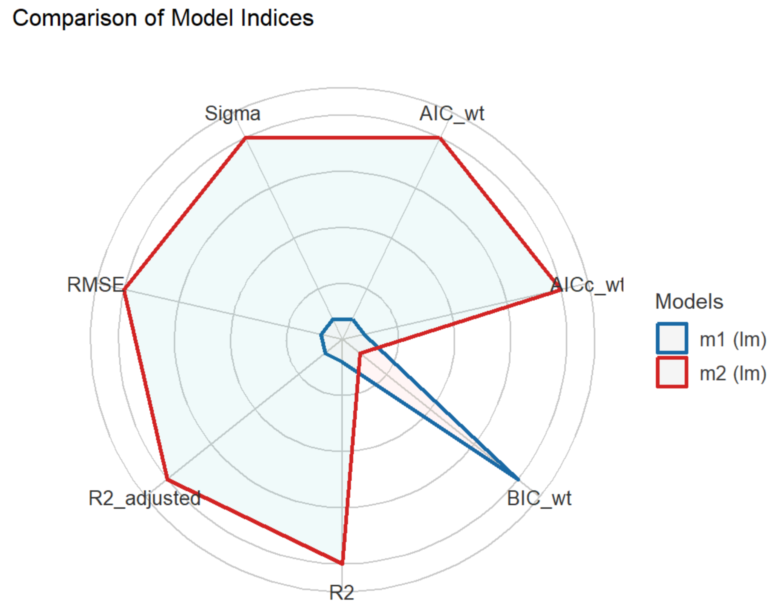
```r
bind_rows(glance(m1), glance(m2)) |>
  mutate(model = c("m1", "m2")) |>
  select(model, r2 = r.squared, adjr2 = adj.r.squared,
         sigma, AIC, BIC, nobs, df, df.residual)
```

```
# A tibble: 2 × 9
  model     r2  adjr2 sigma   AIC   BIC  nobs    df df.residual
  <chr>  <dbl>  <dbl> <dbl> <dbl> <dbl> <int> <dbl>       <int>
1 m1    0.0272 0.0267  1.02 5714. 5731.  1982     1        1980
2 m2    0.0334 0.0304  1.02 5711. 5756.  1982     6        1975
```

Which model appears to fit the data better?

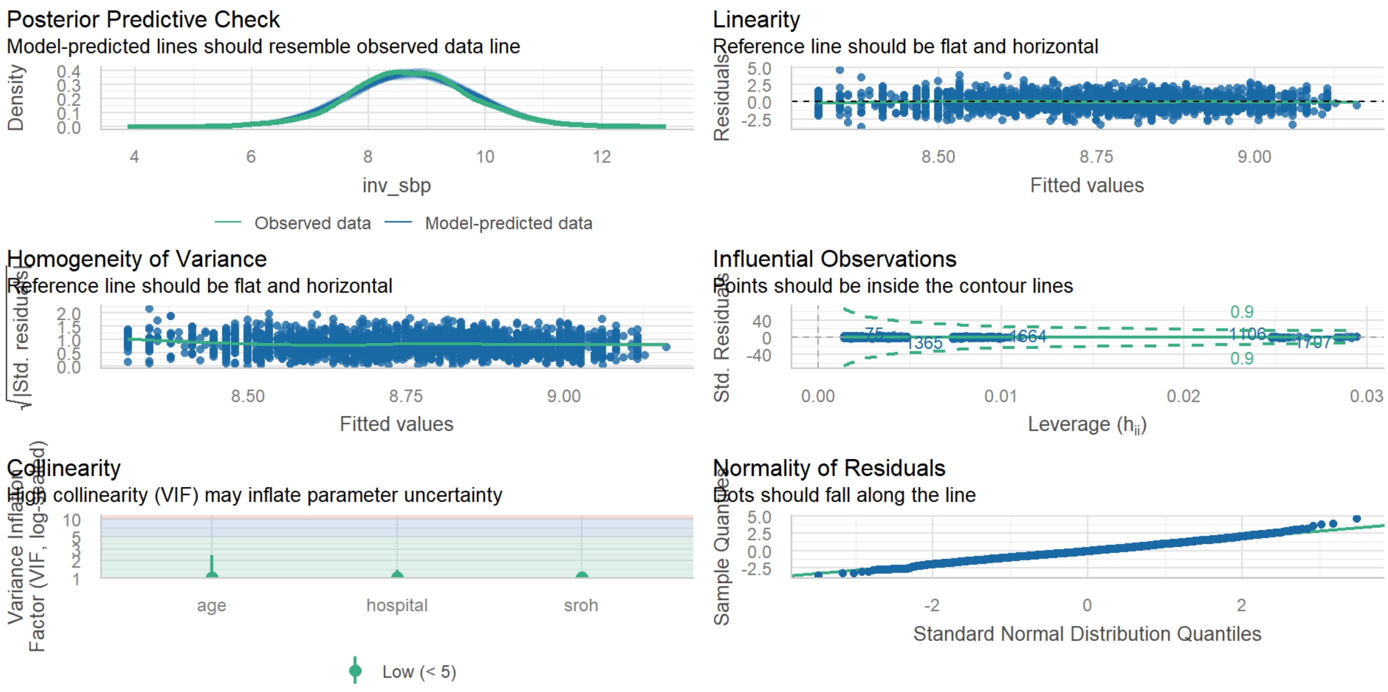
Compare `m1` to `m2`

```r
plot(compare_performance(m1, m2))
```



Residual Plots for Model `m2`

```r
check_model(m2, detrend = FALSE)
```



Making a Prediction in New Data

Suppose a new person is age 29, was not hospitalized, and their SROH is “Good”. What is their predicted mean systolic blood pressure?

- Our models predict 1000/mean_sbp and augment places that prediction into `.fitted`.
- To invert, divide `.fitted` by 1000, then take the reciprocal of that result. That’s just 1000/`.fitted`.

Making a Prediction in New Data

```r
new_person <- tibble(age = 29, sroh = "Good", hospital = "No")
bind_rows(augment(m1, newdata = new_person),
          augment(m2, newdata = new_person)) |>
  mutate(model = c("m1", "m2"), fit_meansbp = 1000/.fitted) |>
  select(model, fit_meansbp, .fitted, age, sroh, hospital)
```

```
# A tibble: 2 × 6
  model fit_meansbp .fitted   age sroh  hospital
  <chr>       <dbl>   <dbl> <dbl> <chr> <chr>
1 m1           112.    8.92    29 Good  No
2 m2           112.    8.90    29 Good  No
```