432 Class 01

https://thomaselove.github.io/432-2025/

2025-01-14

## Getting To These Slides

Our web site: <https://thomaselove.github.io/432-2025/>

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

Visit the [Calendar](https://thomaselove.github.io/432-2025/calendar.html) 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 Word version.

## Today’s Agenda

1. Mechanics of the course
2. Why I write dates the way I do
3. Data organization in spreadsheets
4. Naming Things and Getting Organized
5. Switching from R Markdown to Quarto
6. Building and Validating small models for Penguin Bill Length

# Course Mechanics

## Welcome to 432.

Just about everything is linked at <https://thomaselove.github.io/432-2025>

* [Calendar](https://thomaselove.github.io/432-2025/calendar.html)
  + final word on all deadlines, and links to each class and TA office hours.
* [Syllabus](https://thomaselove.github.io/432-syllabus-2025/) (can download as PDF)
* [Course Notes](https://thomaselove.github.io/432-notes/) HTML and PDF

## Also linked on our website

* [Software](https://thomaselove.github.io/432-2025/software.html)
  + Updating / Installing R and RStudio, necessary R Packages
* Get Data (Code, Quarto templates) at [our 432-data page](https://github.com/THOMASELOVE/432-data)
* Assignments (Labs, Projects, Quizzes - see next few slides)
* [Sources](https://github.com/THOMASELOVE/432-sources) (books, articles, videos, etc.)
* Key Links ([Canvas](https://canvas.case.edu/), [Campuswire](https://campuswire.com/), Shared Google Drive, [Minute Papers](https://github.com/THOMASELOVE/432-minute-2025))
* Contact Us ([Campuswire](https://campuswire.com/) + TA office hours + My email)

## Assignments

Every deliverable is listed in the [Calendar](https://thomaselove.github.io/432-2025/calendar.html).

* Welcome to 432 Survey at <https://bit.ly/432-2025-welcome-survey> due tomorrow (Wednesday 2025-01-15) at Noon.
  + Be sure you see the course in [Campuswire](https://campuswire.com/) and on [Canvas](https://canvas.case.edu/), and the Shared Google Drive at your CWRU log-in. Thanks.

Assignments include two projects, seven labs, and two quizzes. Almost everything is due on Wednesdays at noon.

## Two Projects

[Project A](https://thomaselove.github.io/432-2025/projA.html) (publicly available data: linear & logistic models)

1. Plan: 2025-02-05 (data selection, cleaning, exploration)
2. Final Portfolio & (recorded) Presentation due 2025-03-19

[Project B](https://thomaselove.github.io/432-2025/projB.html) (use almost any data and build specific models)

1. Proposal Form 2025-04-02
2. Presentation (in-person or Zoom) in mid-late April
3. Portfolio (prepared using Quarto) due 2025-04-30

## Seven Labs

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

1. [Lab 1](https://thomaselove.github.io/432-2025/lab1.html) is due Wednesday 2025-01-22 at Noon.
2. [Lab 2](https://thomaselove.github.io/432-2025/lab2.html) is due Wednesday 2025-01-29 at Noon.

[Lab 6](https://thomaselove.github.io/432-2025/lab6.html) is about building or augmenting your website, and can be done now (or at any time), although it’s not due until 2025-03-26.

* You can skip one Lab from Labs 1-5 (we’ll take your four highest grades), but **everyone** will do Lab 6 and Lab 7.

## Two Quizzes

* Quiz 1 due 2025-03-05, Quiz 2 due 2025-04-23.
  + Receive Quiz on Friday by 5 PM, due Wednesday at Noon.
  + Mostly multiple choice or short answer, via Google Form.

[Syllabus](https://thomaselove.github.io/432-syllabus-2025/), Lab Instructions provide feedback details.

## Getting Help

* [Campuswire](https://campuswire.com/) is the location for discussion about the class.
* Seven teaching assistants volunteering their time to help you.
* TAs hold Zoom Office Hours (Fridays and Sunday - Tuesday) starting Friday 2025-01-17.
* Dr. Love is also available after every class to chat.
* Email Dr. Love if you have a matter you need to discuss with him, at **Thomas dot Love** at **case dot edu**.

We WELCOME questions/comments/corrections/thoughts!

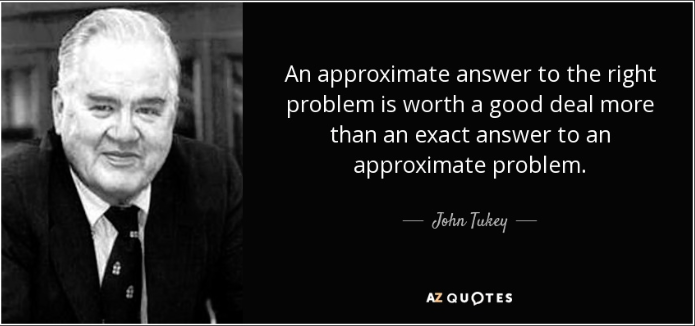
## Tools You Will Use in this Class

* **Course Website** (bottom of every slide) especially the [Calendar](https://thomaselove.github.io/432-2025/calendar.html)
  + Each class has a README plus slides
* **R, RStudio and Quarto** for, well, everything
* [**Canvas**](https://canvas.case.edu/) for access to Zoom meetings and 432 recordings, submission of Labs and Project assignments
* **Google Drive via CWRU** for forms (Surveys/Quizzes) and for feedback on assignments.

## Tools You Will Use in this Class

* [**Campuswire**](https://campuswire.com/) is our discussion board. It’s a moderated place to ask questions, answer questions of your colleagues, and get help fast. Open 24/7.
  + There’s a poll there now (closes tomorrow at noon).
* **Zoom** for class sessions / recordings and TA office hours

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



## This Semester

I broke my left leg and did a fair amount of soft tissue damage 2024-11-19, and this resulted in ankle surgery on 2024-12-11. So this semester will be weird, as I am not allowed to put weight on that foot yet, in a boot and using a walker.

* The 16 classes before Spring Break will be held via Zoom.
* After Spring Break (starting March 18) we plan to return to in-person classes, for the most part.

Zoom link for each class is found in your email, on Canvas, and in our Shared Google Drive.

## Vs. 431 with me in Fall 2024?

* Zoom for the first 16 classes; in person after Spring Break.
* We drop lowest of Labs 1-5; Labs 6 and 7 count for everyone.
* I may or may not use minute papers - less frequent.
* Project B presentations are within the semester, mostly.
* 432 Notes were developed in 2022-24. Updates will come in the form of a new set of examples (SUPPORT study) provided in a different way.
* Shared Google Drive is actually a Drive, not just a folder.

# Why I Write Dates The Way I Do

## How To Write Dates (<https://xkcd.com/1179/>)



# Data Organization in Spreadsheets

## Tidy Data (Wickham)

“A huge amount of effort is spent cleaning data to get it ready for analysis, but there has been little research on how to make data cleaning as easy and effective as possible….

**Tidy datasets are easy to manipulate, model and visualize, and have a specific structure: each variable is a column, each observation is a row, and each type of observational unit is a table.**

## Tidy Data (continued)

This framework makes it easy to tidy messy datasets because only a small set of tools are needed to deal with a wide range of un-tidy datasets. This structure also makes it easier to develop tidy tools for data analysis, tools that both input and output tidy datasets. The advantages of a consistent data structure and matching tools are demonstrated with a case study free from mundane data manipulation chores.”

<https://www.jstatsoft.org/article/view/v059i10>

## “Data Tidying” presentation in *R for Data Science, 2e*

* Defines tidy data
* Demonstrates methods for tidying messy data in R

Read Sections 3 (Data transformation) and 5 (Data tidying)

<https://r4ds.hadley.nz/>

## 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**

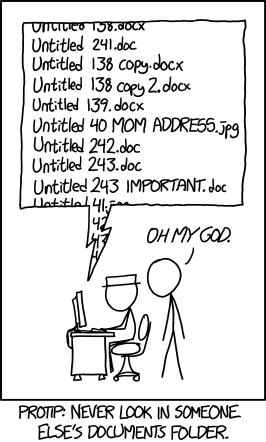
* Consistent subject / record identifiers
  + And if you’re building a .csv in Excel, don’t use ID as the name of that identifier.
* Consistent data layouts across multiple files.

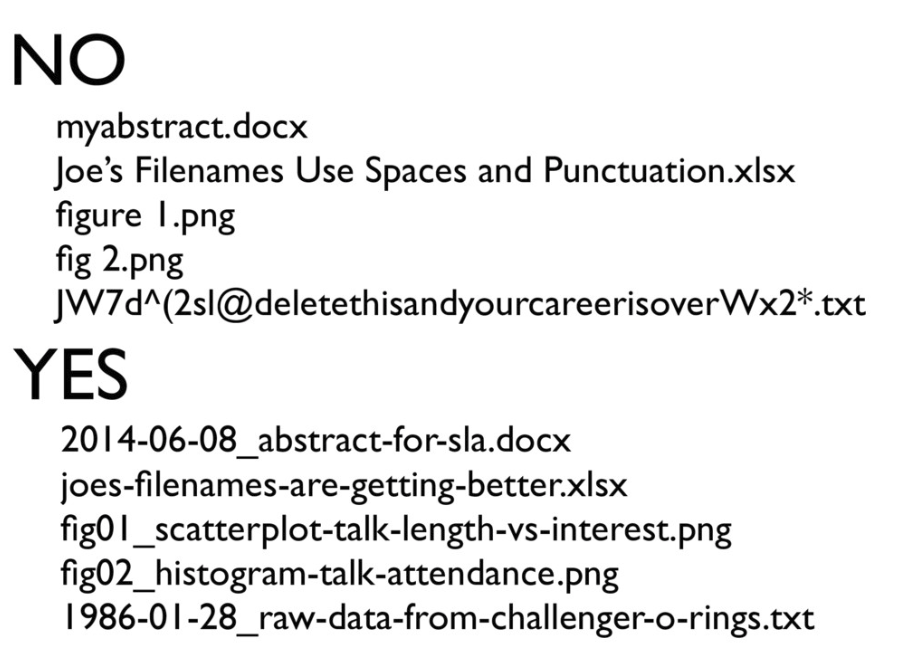
## What Goes in a Cell?

* 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

# Naming Things and Getting Organized

## Naming Files is Hard (<https://xkcd.com/1459/>)





## Data Organization in Spreadsheets: Use consistent, strong file names.

Jenny Bryan’s advice on “Naming Things” hold up well. There’s a full presentation at [SpeakerDeck](https://speakerdeck.com/jennybc/how-to-name-files).

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)

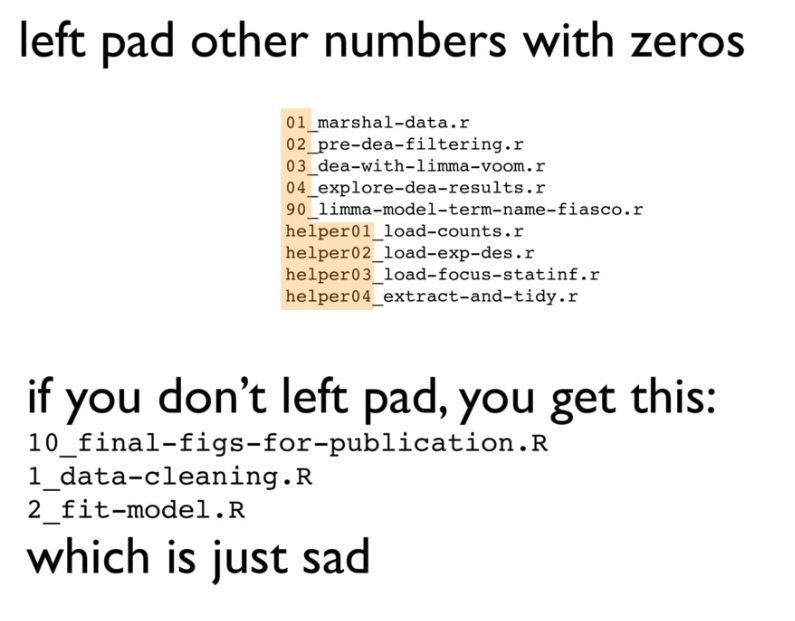
## from Jenny Bryan’s “Naming Things” slides…

Good file names:

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

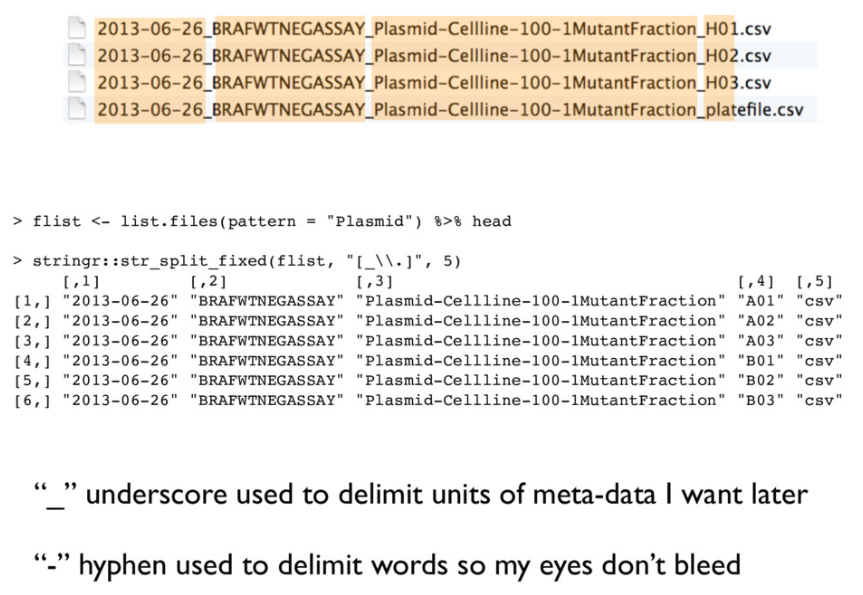
Avoid: spaces, punctuation, accented characters, case sensitivity

## from Jenny Bryan…

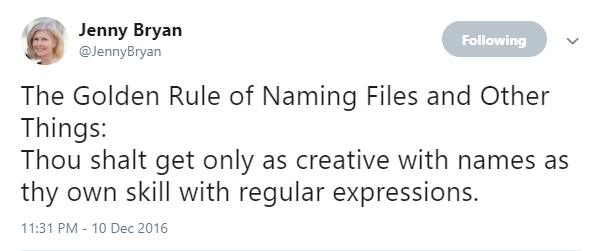


## Jenny Bryan: Deliberate Use of Delimiters

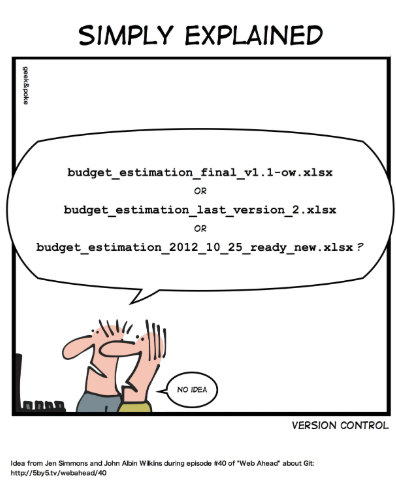
Deliberately use delimiters to make things easy to compute on and make it easy to recover meta-data from the filenames.



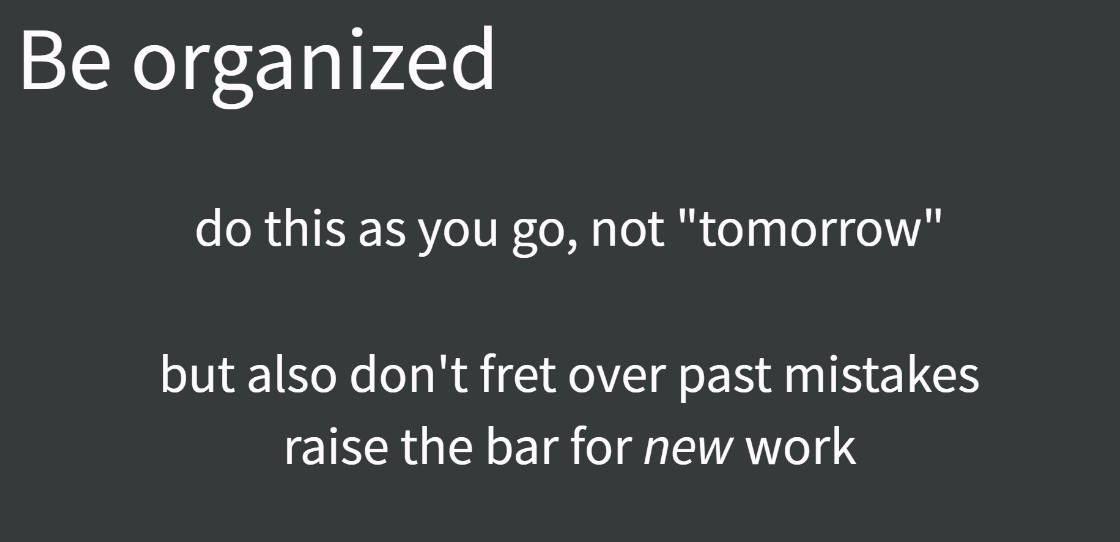
## Don’t get too cute.



## Goal: Avoid this…



## Get organized



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](http://bit.ly/good-enuff)”

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.

# Switching from R Markdown to Quarto

## R Markdown to Quarto

<https://quarto.org/> is the main website for Quarto.

If you can write an R Markdown file, it will also work in Quarto, by switching the extension from .rmd to .qmd.

* Quarto template for Lab 1 (due Wednesday 2025-01-22 at Noon) should ease your transition.
* Read [Chapter 30 (Quarto)](https://r4ds.hadley.nz/quarto.html) in [R for Data Science, 2e](https://r4ds.hadley.nz/)
* Lots of other suggestions in the [Class 01 README](https://github.com/THOMASELOVE/432-classes-2025/tree/main/class01) and [our Sources page](https://github.com/THOMASELOVE/432-classes-2025/tree/main/sources#learning-about-quarto-and-making-the-switch-from-r-markdown).

All course material is written using Quarto.

# Building and Validating Linear Prediction Models

## R Setup

knitr::opts\_chunk$set(comment = NA)  
  
library(janitor)  
  
library(broom); 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

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

## 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(20250117)  
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](https://thomaselove.github.io/432-notes/), too.

## Result of our partitioning

our\_train |> tabyl(species) |> adorn\_totals() |>   
 adorn\_pct\_formatting()

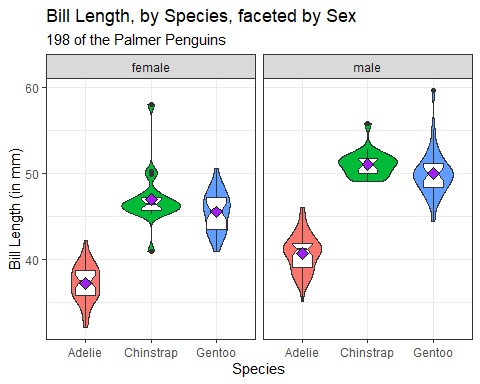
species n percent  
 Adelie 87 43.9%  
 Chinstrap 40 20.2%  
 Gentoo 71 35.9%  
 Total 198 100.0%

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?

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)")



## Model m1

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 4032.7 2016.37 360.78 < 2.2e-16 \*\*\*  
sex 1 770.8 770.83 137.92 < 2.2e-16 \*\*\*  
Residuals 194 1084.3 5.59   
---  
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 10.1 9.32 10.8   
3 speciesGentoo 8.77 8.15 9.40  
4 sexmale 3.96 3.40 4.51

model\_parameters(m1, ci = 0.90)

Parameter | Coefficient | SE | 90% CI | t(194) | p  
---------------------------------------------------------------------------  
(Intercept) | 36.99 | 0.31 | [36.47, 37.50] | 118.54 | < .001  
species [Chinstrap] | 10.06 | 0.45 | [ 9.32, 10.81] | 22.28 | < .001  
species [Gentoo] | 8.77 | 0.38 | [ 8.15, 9.40] | 23.15 | < .001  
sex [male] | 3.96 | 0.34 | [ 3.40, 4.51] | 11.74 | < .001

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

# 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 10.1 9.32 10.8   
3 speciesGentoo 8.77 8.15 9.40  
4 sexmale 3.96 3.40 4.51

model\_parameters(m1, ci = 0.90)

Parameter | Coefficient | SE | 90% CI | t(194) | p  
---------------------------------------------------------------------------  
(Intercept) | 36.99 | 0.31 | [36.47, 37.50] | 118.54 | < .001  
species [Chinstrap] | 10.06 | 0.45 | [ 9.32, 10.81] | 22.28 | < .001  
species [Gentoo] | 8.77 | 0.38 | [ 8.15, 9.40] | 23.15 | < .001  
sex [male] | 3.96 | 0.34 | [ 3.40, 4.51] | 11.74 | < .001

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

## 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) | 36.99 | 36.47 | 37.50 |
| speciesChinstrap | 10.06 | 9.32 | 10.81 |
| speciesGentoo | 8.77 | 8.15 | 9.40 |
| sexmale | 3.96 | 3.40 | 4.51 |

## Model 1 Parameters (version 2b)

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) | 36.99 | 0.31 | 0.90 | 36.47 | 37.50 | 118.54 | 194 | 5.923496e-183 |
| speciesChinstrap | 10.06 | 0.45 | 0.90 | 9.32 | 10.81 | 22.28 | 194 | 2.226881e-55 |
| speciesGentoo | 8.77 | 0.38 | 0.90 | 8.15 | 9.40 | 23.15 | 194 | 1.003133e-57 |
| sexmale | 3.96 | 0.34 | 0.90 | 3.40 | 4.51 | 11.74 | 194 | 2.096987e-24 |

## Model 1 Parameters (version 3)

|  |
| --- |
| Note |
| Note that kable() doesn’t produce results that work properly in Word, so I won’t use kable() approaches in the Word documents for slides. |

tidy(m1, conf.int = TRUE, conf.level = 0.90) |>  
 select(term, estimate, conf.low, conf.high) |>   
 kable(digits = 2) |> kable\_styling(font\_size = 24)  
  
model\_parameters(m1, ci = 0.90) |>   
 kable(digits = 2) |> kable\_styling(font\_size = 24)

## Model 1 Parameters (version 4)

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) | 36.99 | 36.47 | 37.50 |
| speciesChinstrap | 10.06 | 9.32 | 10.81 |
| speciesGentoo | 8.77 | 8.15 | 9.40 |
| sexmale | 3.96 | 3.40 | 4.51 |
|  | | | |

model\_parameters(m1, ci = 0.90) |>   
 print\_html(digits = 2, font\_size = "60%")

| Parameter | Coefficient | SE | 90% CI | t(194) | p |
| --- | --- | --- | --- | --- | --- |
| (Intercept) | 36.99 | 0.31 | (36.47, 37.50) | 118.54 | < .001 |
| species (Chinstrap) | 10.06 | 0.45 | (9.32, 10.81) | 22.28 | < .001 |
| species (Gentoo) | 8.77 | 0.38 | (8.15, 9.40) | 23.15 | < .001 |
| sex (male) | 3.96 | 0.34 | (3.40, 4.51) | 11.74 | < .001 |
|  | | | | | |

## Model 1 performance

model\_performance(m1) |> gt() |>   
 fmt\_number(decimals = 3) |> tab\_options(table.font.size = 24)

| AIC | AICc | BIC | R2 | R2\_adjusted | RMSE | Sigma |
| --- | --- | --- | --- | --- | --- | --- |
| 908.576 | 908.888 | 925.017 | 0.816 | 0.813 | 2.340 | 2.364 |

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 | 2.364 | 286.492 | 0.000 | 3 | -449.288 | 908.576 | 925.017 | 1,084.258 | 194 | 198 |

## Model m2

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) | 39.1 | 38.6 | 39.7 |
| speciesChinstrap | 10.0 | 9.0 | 11.0 |
| speciesGentoo | 8.5 | 7.7 | 9.3 |

## Comparison of Coefficients

compare\_models(m1, m2)

Parameter | m1 | m2  
-----------------------------------------------------------------  
(Intercept) | 36.99 (36.37, 37.60) | 39.12 (38.47, 39.78)  
species [Chinstrap] | 10.06 ( 9.17, 10.95) | 10.00 ( 8.84, 11.17)  
species [Gentoo] | 8.77 ( 8.03, 9.52) | 8.47 ( 7.50, 9.45)  
sex [male] | 3.96 ( 3.29, 4.62) |   
-----------------------------------------------------------------  
Observations | 198 | 198

## In-Sample Comparison

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 | 908.6 | 925.0 | 2.36 | 198 |
| m2 | 0.685 | 0.682 | 1012.9 | 1026.1 | 3.08 | 198 |

Which model has better in-sample performance?

## Comparing m1 vs. m2 performance

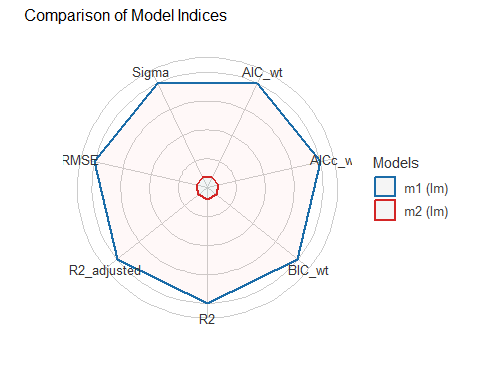
compare\_performance(m1, m2)

# Comparison of Model Performance Indices  
  
Name | Model | AIC (weights) | AICc (weights) | BIC (weights) | R2  
-----------------------------------------------------------------------  
m1 | lm | 908.6 (>.999) | 908.9 (>.999) | 925.0 (>.999) | 0.816  
m2 | lm | 1012.9 (<.001) | 1013.1 (<.001) | 1026.1 (<.001) | 0.685  
  
Name | R2 (adj.) | RMSE | Sigma  
--------------------------------  
m1 | 0.813 | 2.340 | 2.364  
m2 | 0.682 | 3.061 | 3.084

Which model has better in-sample performance?

## Plot for m1 vs. m2 (training)

plot(compare\_performance(m1, m2))



## Assessing Performance in Test Sample

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

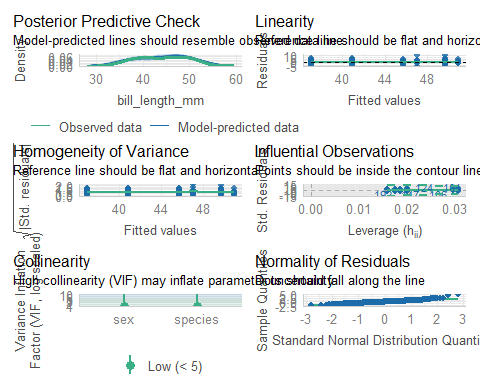
bind\_rows(m1\_res, m2\_res) |>  
 mutate(model = c("m1", "m2")) |>  
 relocate(model) |> kable(digits = c(0, 3, 2, 2, 1))

| model | val\_R\_sq | MAPE | RMSPE | max\_Error |
| --- | --- | --- | --- | --- |
| m1 | 0.831 | 1.77 | 2.29 | 6.3 |
| m2 | 0.742 | 2.31 | 2.83 | 8.3 |

Which model predicts better in the test sample?

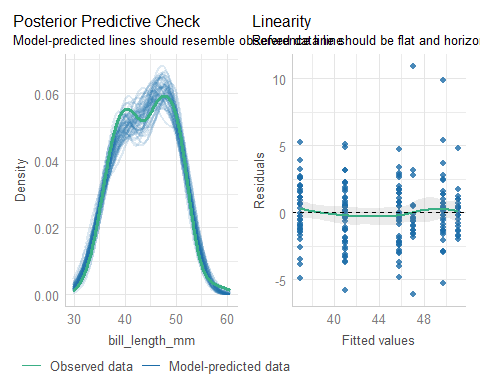
## Checking m1 (see next 3 slides)

check\_model(m1, detrend = FALSE)



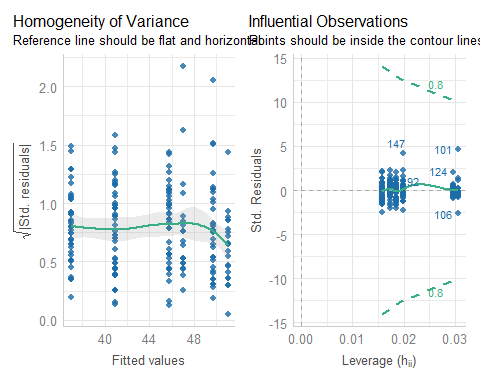
## check\_model(m1): first 2 plots

check\_model(m1, detrend = FALSE, check = c("pp\_check", "linearity"))



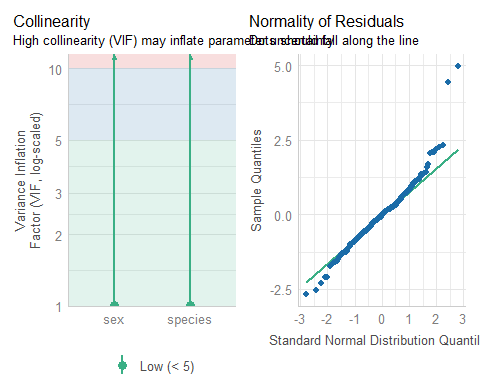
## check\_model(m1): next 2 plots

check\_model(m1, detrend = FALSE, check = c("homogeneity", "outliers"))



## check\_model(m1): final 2 plots

check\_model(m1, detrend = FALSE, check = c("vif", "qq"))



## 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…

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

## For Next Time…

1. If you’re not registered with SIS, do so, for PQHS/CRSP/MPHP 432.
2. Review the website and [Calendar](https://thomaselove.github.io/432-2025/calendar.html), and skim the [Syllabus](https://thomaselove.github.io/432-syllabus-2025/) and [Course Notes](https://thomaselove.github.io/432-notes/).
3. Welcome to 432 Survey at <https://bit.ly/432-2025-welcome-survey> by noon Wednesday 2025-01-15.

## For Next Time…

1. Accept the invitation to join the [Campuswire](https://campuswire.com/) Discussion Forum for 432 and answer the poll question.
2. Buy Jeff Leek’s [How to be a Modern Scientist](https://leanpub.com/modernscientist) and read it by the end of January.
3. Get started installing or updating the [software](https://thomaselove.github.io/432-2025/software.html) you need for the course.
4. Get started on [Lab 1](https://thomaselove.github.io/432-2025/lab1.html), due Wednesday 2025-01-22 at Noon.