431 Class 08

https://thomaselove.github.io/431-2024/

2024-09-19

## Today’s Agenda

* One-Factor Analysis of Variance
  + Using Regression to Develop an ANOVA model
  + Methods for pairwise multiple comparisons
  + Checking Model Assumptions
  + Bayesian fitting of an ANOVA model

## Load packages and set theme

library(janitor)  
  
library(readxl) # to read in an .xlsx file  
  
library(car)   
library(rstanarm)  
library(easystats)  
library(tidyverse)  
  
theme\_set(theme\_bw())  
knitr::opts\_chunk$set(comment = NA)  
  
source("c08/data/Love-431.R") # for the lovedist() function

## Importing Data / Creating Factors

ohio20 <- read\_xlsx("c08/data/ohio\_2020.xlsx") |>  
 mutate(behavior = Hmisc::cut2(rk\_behavior, g = 4),  
 clin\_care = Hmisc::cut2(rk\_clin\_care, g = 3)) |>  
 mutate(behavior = fct\_recode(behavior,  
 "Best" = "[ 1,23)", "High" = "[23,45)",  
 "Low" = "[45,67)", "Worst" = "[67,88]")) |>  
 mutate(clin\_care = fct\_recode(clin\_care,  
 "Strong" = "[ 1,31)", "Middle" = "[31,60)",  
 "Weak" = "[60,88]")) |>  
 select(FIPS, state, county, outcomes, behavior, clin\_care,   
 everything())

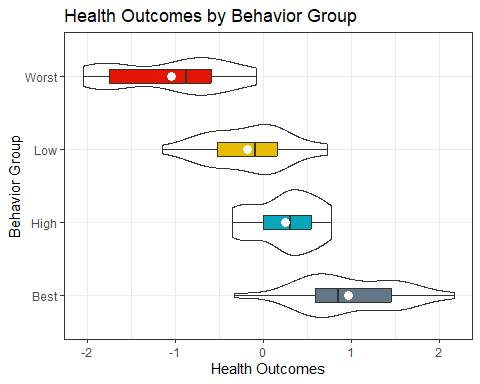
## Any missing values?

data\_codebook(ohio20)

ohio20 (88 rows and 12 variables, 12 shown)  
  
ID | Name | Type | Missings | Values | N  
---+--------------+-------------+----------+-----------------+------------  
1 | FIPS | character | 0 (0.0%) | 39001 | 1 ( 1.1%)  
 | | | | 39003 | 1 ( 1.1%)  
 | | | | 39005 | 1 ( 1.1%)  
 | | | | 39007 | 1 ( 1.1%)  
 | | | | 39009 | 1 ( 1.1%)  
 | | | | 39011 | 1 ( 1.1%)  
 | | | | 39013 | 1 ( 1.1%)  
 | | | | 39015 | 1 ( 1.1%)  
 | | | | 39017 | 1 ( 1.1%)  
 | | | | 39019 | 1 ( 1.1%)  
 | | | | (...) |   
---+--------------+-------------+----------+-----------------+------------  
2 | state | character | 0 (0.0%) | Ohio | 88 (100.0%)  
---+--------------+-------------+----------+-----------------+------------  
3 | county | character | 0 (0.0%) | Adams | 1 ( 1.1%)  
 | | | | Allen | 1 ( 1.1%)  
 | | | | Ashland | 1 ( 1.1%)  
 | | | | Ashtabula | 1 ( 1.1%)  
 | | | | Athens | 1 ( 1.1%)  
 | | | | Auglaize | 1 ( 1.1%)  
 | | | | Belmont | 1 ( 1.1%)  
 | | | | Brown | 1 ( 1.1%)  
 | | | | Butler | 1 ( 1.1%)  
 | | | | Carroll | 1 ( 1.1%)  
 | | | | (...) |   
---+--------------+-------------+----------+-----------------+------------  
4 | outcomes | numeric | 0 (0.0%) | [-2.05, 2.17] | 88  
---+--------------+-------------+----------+-----------------+------------  
5 | behavior | categorical | 0 (0.0%) | Best | 22 ( 25.0%)  
 | | | | High | 22 ( 25.0%)  
 | | | | Low | 22 ( 25.0%)  
 | | | | Worst | 22 ( 25.0%)  
---+--------------+-------------+----------+-----------------+------------  
6 | clin\_care | categorical | 0 (0.0%) | Strong | 30 ( 34.1%)  
 | | | | Middle | 29 ( 33.0%)  
 | | | | Weak | 29 ( 33.0%)  
---+--------------+-------------+----------+-----------------+------------  
7 | rk\_behavior | numeric | 0 (0.0%) | [1, 88] | 88  
---+--------------+-------------+----------+-----------------+------------  
8 | rk\_clin\_care | numeric | 0 (0.0%) | [1, 88] | 88  
---+--------------+-------------+----------+-----------------+------------  
9 | rural | numeric | 0 (0.0%) | [0.58, 100] | 88  
---+--------------+-------------+----------+-----------------+------------  
10 | income | numeric | 0 (0.0%) | [40416, 103536] | 88  
---+--------------+-------------+----------+-----------------+------------  
11 | trump16 | numeric | 0 (0.0%) | [0.31, 0.81] | 88  
---+--------------+-------------+----------+-----------------+------------  
12 | somecollege | numeric | 0 (0.0%) | [20.38, 85.07] | 88  
--------------------------------------------------------------------------

## Outcomes by Behavior Groups

ggplot(ohio20, aes(x = outcomes, y = behavior)) +  
 geom\_violin() +  
 geom\_boxplot(aes(fill = behavior), width = 0.2) +  
 stat\_summary(fun = mean, geom = "point",   
 shape = 16, size = 3, col = "white") +  
 scale\_fill\_metro\_d() +  
 guides(fill = "none") +  
 labs(x = "Health Outcomes", y = "Behavior Group",  
 title = "Health Outcomes by Behavior Group")



## Building the Linear Model

Do we see meaningful differences in population means of outcomes across behavior groups?

model\_one <- lm(outcomes ~ behavior, data = ohio20)  
  
model\_parameters(model\_one, ci = 0.90)

Parameter | Coefficient | SE | 90% CI | t(84) | p  
------------------------------------------------------------------------  
(Intercept) | 0.96 | 0.11 | [ 0.78, 1.15] | 8.74 | < .001  
behavior [High] | -0.71 | 0.16 | [-0.97, -0.45] | -4.55 | < .001  
behavior [Low] | -1.14 | 0.16 | [-1.40, -0.88] | -7.32 | < .001  
behavior [Worst] | -2.01 | 0.16 | [-2.27, -1.75] | -12.85 | < .001

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

## ANOVA Table

anova(model\_one)

Analysis of Variance Table  
  
Response: outcomes  
 Df Sum Sq Mean Sq F value Pr(>F)   
behavior 3 46.421 15.4736 57.718 < 2.2e-16 \*\*\*  
Residuals 84 22.519 0.2681   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Estimate Means with 90% CIs

means\_one <- estimate\_means(model\_one, ci = 0.90)

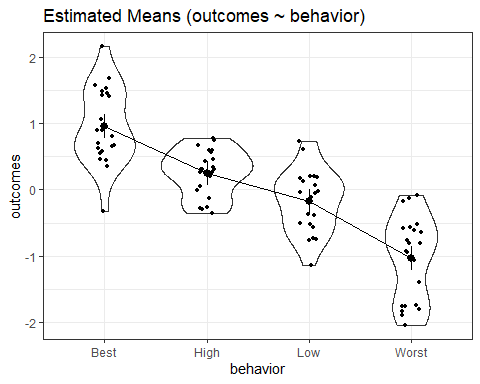
We selected `by = c("behavior")`.

means\_one

Estimated Marginal Means  
  
behavior | Mean | SE | 90% CI  
----------------------------------------  
Best | 0.96 | 0.11 | [ 0.78, 1.15]  
High | 0.25 | 0.11 | [ 0.07, 0.44]  
Low | -0.18 | 0.11 | [-0.36, 0.01]  
Worst | -1.04 | 0.11 | [-1.22, -0.86]  
  
Marginal means estimated at behavior

### Plot the means

plot(visualisation\_recipe(means\_one,   
 show\_data = c("violin", "jitter")))



## What’s Left? (Multiple Comparisons)

* Can we identify pairwise comparisons of means that show meaningful differences?
* Can we build a confidence interval procedure that protects against Type I error expansion due to multiple comparisons?

## Two Methods for Multiple Comparisons

There are two (main) methods we’ll study to estimate the difference between specific pairs of means with uncertainty intervals, while dealing with the problem of multiple comparisons.

* Holm-Bonferroni pairwise comparisons
* Tukey’s HSD (Honestly Significant Differences) approach

Just for fun, we’ll use a 99% confidence level today.

## Compare behavior group means?

ANOVA suggests that the means aren’t likely to all be the same, but we really want to make pairwise comparisons…

estimate\_means(model\_one, ci = 0.99)

We selected `by = c("behavior")`.

Estimated Marginal Means  
  
behavior | Mean | SE | 99% CI  
----------------------------------------  
Best | 0.96 | 0.11 | [ 0.67, 1.26]  
High | 0.25 | 0.11 | [-0.04, 0.54]  
Low | -0.18 | 0.11 | [-0.47, 0.11]  
Worst | -1.04 | 0.11 | [-1.33, -0.75]  
  
Marginal means estimated at behavior

For example, is Best meaningfully different from High?

## Could we just run a bunch of t tests?

This approach assumes that you need to make no adjustment for the fact that you are doing multiple comparisons, simultaneously.

estimate\_contrasts(model\_one, ci = 0.99, p\_adjust = "none")

No variable was specified for contrast estimation. Selecting `contrast =  
 "  
 behavior  
 "`.

Marginal Contrasts Analysis  
  
Level1 | Level2 | Difference | 99% CI | SE | t(84) | p  
-------------------------------------------------------------------  
Best | High | 0.71 | [0.30, 1.12] | 0.16 | 4.55 | < .001  
Best | Low | 1.14 | [0.73, 1.55] | 0.16 | 7.32 | < .001  
Best | Worst | 2.01 | [1.59, 2.42] | 0.16 | 12.85 | < .001  
High | Low | 0.43 | [0.02, 0.84] | 0.16 | 2.76 | 0.007   
High | Worst | 1.29 | [0.88, 1.71] | 0.16 | 8.29 | < .001  
Low | Worst | 0.86 | [0.45, 1.27] | 0.16 | 5.53 | < .001  
  
Marginal contrasts estimated at behavior  
p-values are uncorrected.

## The problem of Multiple Comparisons

* The more comparisons you do simultaneously, the more likely you are to make an error.

In the worst case scenario, suppose you do two tests - first A vs. B and then A vs. C, each at the 99% confidence level.

* What is the combined error rate across those two t tests?

## The problem of Multiple Comparisons

Run the first test. Make a Type I error 1% of the time.

| A vs B Type I error | Probability |
| --- | --- |
| Yes | 0.01 |
| No | 0.99 |

Now, run the second test. Assume (perhaps wrongly) that comparing A to C is independent of your A-B test result. What is the error rate now?

## The problem of Multiple Comparisons

Assuming there is a 1% chance of making an error in either test, independently …

| – | Error in A vs. C | No Error | Total |
| --- | --- | --- | --- |
| Type I error in A vs. B | 0.0001 | 0.0099 | 0.01 |
| No Type I error in A-B | 0.0099 | 0.9801 | 0.99 |
| Total | 0.01 | 0.99 | 1.00 |

So you will make an error in the A-B or A-C comparison **1.99%** of the time, rather than the nominal error rate.

## But we’re building SIX tests

1. Best vs. High
2. Best vs. Low
3. Best vs. Worst
4. High vs. Low
5. High vs. Worst
6. Low vs. Worst

and if they were independent, and each done at a 5% error rate, we could still wind up with an error rate of

= .265

Or worse, if they’re not independent.

## The Bonferroni Method

If we do 6 tests, we could reduce the necessary to 0.01 / 6 = 0.00167 and that maintains an error rate no higher than across the 6 tests.

## Better Approach: Holm-Bonferroni

Suppose you have comparisons, with p-values sorted from low to high as , , …, .

* Is ? If so, reject and continue, otherwise STOP.
* Is ? If so, reject and continue, else STOP.
* and so on…

## Holm-Bonferroni Approach

This is uniformly more powerful than Bonferroni, while preserving the overall false positive rate at . It’s also the default choice of estimate\_contrasts().

estimate\_contrasts(model\_one, ci = 0.99, p\_adjust = "holm")

No variable was specified for contrast estimation. Selecting `contrast =  
 "  
 behavior  
 "`.

Marginal Contrasts Analysis  
  
Level1 | Level2 | Difference | 99% CI | SE | t(84) | p  
--------------------------------------------------------------------  
Best | High | 0.71 | [ 0.20, 1.22] | 0.16 | 4.55 | < .001  
Best | Low | 1.14 | [ 0.64, 1.65] | 0.16 | 7.32 | < .001  
Best | Worst | 2.01 | [ 1.50, 2.51] | 0.16 | 12.85 | < .001  
High | Low | 0.43 | [-0.08, 0.94] | 0.16 | 2.76 | 0.007   
High | Worst | 1.29 | [ 0.79, 1.80] | 0.16 | 8.29 | < .001  
Low | Worst | 0.86 | [ 0.36, 1.37] | 0.16 | 5.53 | < .001  
  
Marginal contrasts estimated at behavior  
p-value adjustment method: Holm (1979)

## Get the Holm results into a tibble

con\_holm <- estimate\_contrasts(model\_one, ci = 0.99, p\_adjust = "holm")

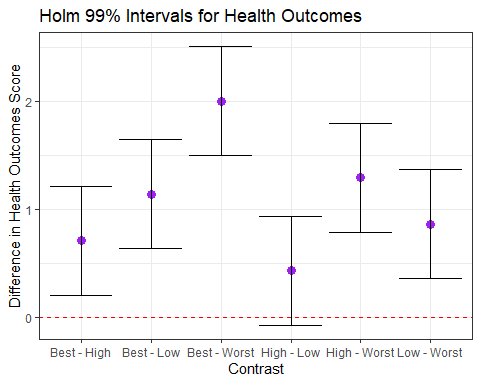
No variable was specified for contrast estimation. Selecting `contrast =  
 "  
 behavior  
 "`.

con\_holm\_tib <- tibble(con\_holm) |>   
 mutate(contr = str\_c(Level1, " - ", Level2))  
  
con\_holm\_tib |> head()

# A tibble: 6 × 10  
 Level1 Level2 Difference CI\_low CI\_high SE df t p contr   
 <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   
1 Best High 0.711 0.204 1.22 0.156 84 4.55 3.52e- 5 Best - Hi…  
2 Best Low 1.14 0.635 1.65 0.156 84 7.32 5.53e-10 Best - Low  
3 Best Worst 2.01 1.50 2.51 0.156 84 12.8 9.54e-21 Best - Wo…  
4 High Low 0.431 -0.0759 0.939 0.156 84 2.76 7.04e- 3 High - Low  
5 High Worst 1.29 0.787 1.80 0.156 84 8.29 7.87e-12 High - Wo…  
6 Low Worst 0.863 0.356 1.37 0.156 84 5.53 1.07e- 6 Low - Wor…

## Plot the Holm results

ggplot(con\_holm\_tib, aes(x = contr, y = Difference)) +  
 geom\_point(size = 3, col = "purple") +  
 geom\_errorbar(aes(ymin = CI\_low, ymax = CI\_high)) +  
 geom\_hline(yintercept = 0, col = "red", lty = "dashed") +  
 labs(title = "Holm 99% Intervals for Health Outcomes",  
 x = "Contrast",   
 y = "Difference in Health Outcomes Score")



## Tukey’s HSD Intervals

Tukey’s HSD approach is a better choice for pre-planned comparisons with a balanced (or nearly balanced) design.

estimate\_contrasts(model\_one, ci = 0.99, p\_adjust = "tukey")

No variable was specified for contrast estimation. Selecting `contrast =  
 "  
 behavior  
 "`.

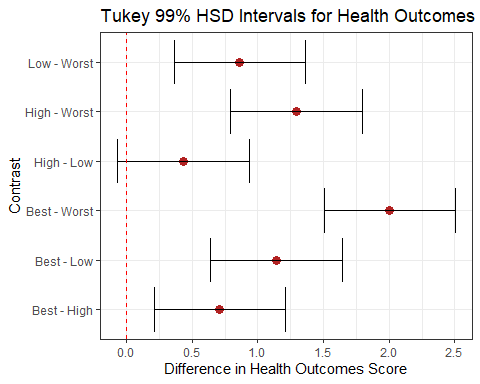
Marginal Contrasts Analysis  
  
Level1 | Level2 | Difference | 99% CI | SE | t(84) | p  
--------------------------------------------------------------------  
Best | High | 0.71 | [ 0.21, 1.21] | 0.16 | 4.55 | < .001  
Best | Low | 1.14 | [ 0.64, 1.64] | 0.16 | 7.32 | < .001  
Best | Worst | 2.01 | [ 1.50, 2.51] | 0.16 | 12.85 | < .001  
High | Low | 0.43 | [-0.07, 0.93] | 0.16 | 2.76 | 0.035   
High | Worst | 1.29 | [ 0.79, 1.80] | 0.16 | 8.29 | < .001  
Low | Worst | 0.86 | [ 0.36, 1.36] | 0.16 | 5.53 | < .001  
  
Marginal contrasts estimated at behavior  
p-value adjustment method: Tukey

## Plot Tukey Intervals

con\_tukey <- estimate\_contrasts(model\_one, ci = 0.99, p\_adjust = "tukey")

No variable was specified for contrast estimation. Selecting `contrast =  
 "  
 behavior  
 "`.

con\_tukey\_tib <- tibble(con\_tukey) |>   
 mutate(contr = str\_c(Level1, " - ", Level2))  
  
ggplot(con\_tukey\_tib, aes(y = contr, x = Difference)) +  
 geom\_point(size = 3, color = "firebrick") +  
 geom\_errorbar(aes(xmin = CI\_low, xmax = CI\_high)) +  
 geom\_vline(xintercept = 0, col = "red", lty = "dashed") +  
 labs(title = "Tukey 99% HSD Intervals for Health Outcomes",  
 y = "Contrast",   
 x = "Difference in Health Outcomes Score")



## ANOVA Assumptions

The assumptions behind analysis of variance are those of a linear model. Of specific interest are:

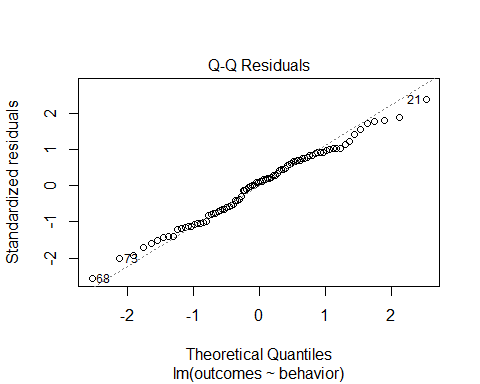
* The samples obtained from each group are independent.
* Ideally, the samples from each group are a random sample from the population described by that group.
* In the population, the variance of the outcome in each group is equal. (This is less of an issue if our study involves a balanced design.)
* In the population, we have Normal distributions of the outcome in each group.

Happily, the ANOVA F test is fairly robust to violations of the Normality assumption.

## Checking the ANOVA Model

We check assumptions of linear models like ANOVA with residual plots, for instance with a Normal Q-Q plot.

plot(model\_one, which = 2)



## More Serious Checking to come

plot(check\_model(model\_one))

For confidence bands, please install `qqplotr`.

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: pseudoinverse used at -1.0509

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: neighborhood radius 1.3046

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: reciprocal condition number 9.9851e-16

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: There are other near singularities as well. 1.3281

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at  
-1.0509

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius  
1.3046

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition  
number 9.9851e-16

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : There are other near  
singularities as well. 1.3281

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: pseudoinverse used at -1.0509

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: neighborhood radius 1.3046

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: reciprocal condition number 9.9851e-16

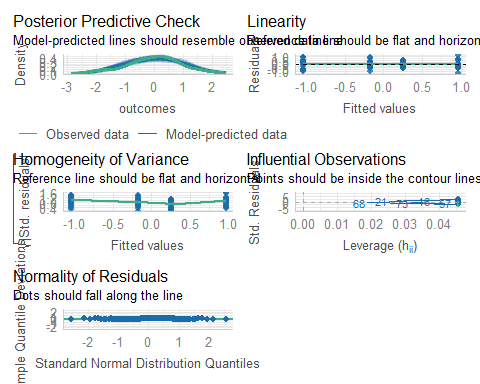
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: There are other near singularities as well. 1.3281

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at  
-1.0509

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius  
1.3046

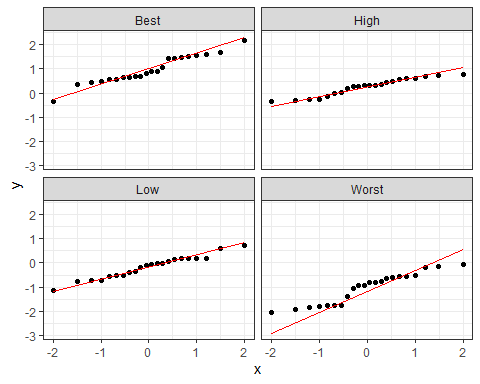
Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition  
number 9.9851e-16

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : There are other near  
singularities as well. 1.3281



## Do we need to transform?

ggplot(ohio20, aes(sample = outcomes)) +  
 geom\_qq() + geom\_qq\_line(col = "red") +  
 facet\_wrap(~ behavior)



## Transformation?

Would a transformation have been helpful in this setting?

Our model was: lm(outcomes ~ behavior, data = ohio20)

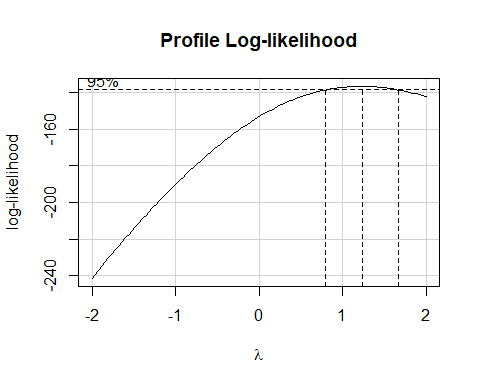
ohio20 |> group\_by(behavior) |> summarize(min(outcomes))

# A tibble: 4 × 2  
 behavior `min(outcomes)`  
 <fct> <dbl>  
1 Best -0.332  
2 High -0.350  
3 Low -1.15   
4 Worst -2.05

* Remember that the Box-Cox method requires our response variable (here outcomes) to be strictly positive.

## Does Box-Cox make a suggestion?

ohio20 <- ohio20 |> mutate(out = outcomes + 3)  
model2 <- lm(out ~ behavior, data = ohio20)  
boxCox(model2)



## Can we avoid assuming equal population variances?

Yes, but this isn’t exciting if we have a balanced design.

oneway.test(outcomes ~ behavior, data = ohio20)

One-way analysis of means (not assuming equal variances)  
  
data: outcomes and behavior  
F = 43.145, num df = 3.000, denom df = 45.494, p-value = 2.349e-13

* Note that this approach uses a fractional degrees of freedom calculation in the denominator.

## The Kruskal-Wallis Test

If you thought the data were severely skewed, you might try:

kruskal.test(outcomes ~ behavior, data = ohio20)

Kruskal-Wallis rank sum test  
  
data: outcomes by behavior  
Kruskal-Wallis chi-squared = 61.596, df = 3, p-value = 2.681e-13

* : The four behavior groups have the same center to their outcomes distributions.
* : At least one group has a shifted distribution, with a different center to its outcomes.

What would be the conclusion here?

## Bayesian Linear Model for ANOVA

set.seed(20240919)  
model\_1b <- stan\_glm(outcomes ~ behavior, data = ohio20, refresh = 0)  
  
model\_parameters(model\_1b, ci = 0.99)

Parameter | Median | 99% CI | pd | Rhat | ESS | Prior  
--------------------------------------------------------------------------------------------  
(Intercept) | 0.97 | [ 0.67, 1.25] | 100% | 1.001 | 1781.00 | Normal (4.09e-11 +- 2.23)  
behaviorHigh | -0.71 | [-1.11, -0.29] | 100% | 1.000 | 1858.00 | Normal (0.00 +- 5.11)  
behaviorLow | -1.14 | [-1.55, -0.74] | 100% | 1.000 | 2230.00 | Normal (0.00 +- 5.11)  
behaviorWorst | -2.00 | [-2.42, -1.59] | 100% | 1.001 | 2090.00 | Normal (0.00 +- 5.11)

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

## Again, more serious checking to come

plot(check\_model(model\_1b))

For confidence bands, please install `qqplotr`.

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: pseudoinverse used at -1.0509

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: neighborhood radius 1.3046

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: reciprocal condition number 9.9851e-16

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: There are other near singularities as well. 1.3281

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at  
-1.0509

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius  
1.3046

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition  
number 9.9851e-16

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : There are other near  
singularities as well. 1.3281

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: pseudoinverse used at -1.0509

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: neighborhood radius 1.3046

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: reciprocal condition number 9.9851e-16

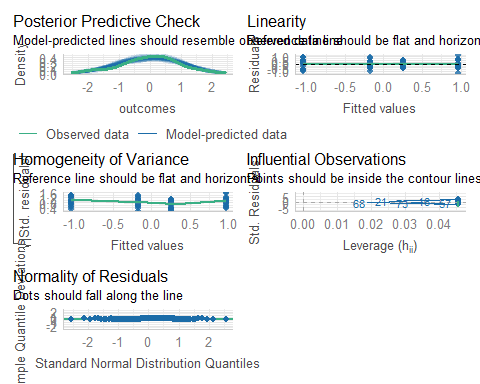
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
: There are other near singularities as well. 1.3281

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at  
-1.0509

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius  
1.3046

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition  
number 9.9851e-16

Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
else if (is.data.frame(newdata))  
as.matrix(model.frame(delete.response(terms(object)), : There are other near  
singularities as well. 1.3281



## Estimate Means after Bayesian Fit

estimate\_means(model\_1b, ci = 0.99)

We selected `by = c("behavior")`.

Estimated Marginal Means  
  
behavior | Mean | 99% CI | pd  
------------------------------------------  
Best | 0.97 | [ 0.67, 1.25] | 100%  
High | 0.25 | [-0.04, 0.54] | 98.83%  
Low | -0.18 | [-0.47, 0.13] | 94.00%  
Worst | -1.04 | [-1.34, -0.76] | 100%  
  
Marginal means estimated at behavior

## Holm 99% CIs after Bayesian fit

estimate\_contrasts(model\_1b, ci = 0.99, p\_adjust = "Holm")

No variable was specified for contrast estimation. Selecting `contrast =  
 "  
 behavior  
 "`.

Marginal Contrasts Analysis  
  
Level1 | Level2 | Difference | 99% CI | pd | % in ROPE  
----------------------------------------------------------------  
Best | High | 0.71 | [0.29, 1.11] | 100% | 0%  
Best | Low | 1.14 | [0.74, 1.55] | 100% | 0%  
Best | Worst | 2.00 | [1.59, 2.42] | 100% | 0%  
High | Low | 0.43 | [0.01, 0.85] | 99.52% | 0%  
High | Worst | 1.29 | [0.90, 1.71] | 100% | 0%  
Low | Worst | 0.86 | [0.44, 1.31] | 100% | 0%  
  
Marginal contrasts estimated at behavior

## Comparison of Holm 99% CIs

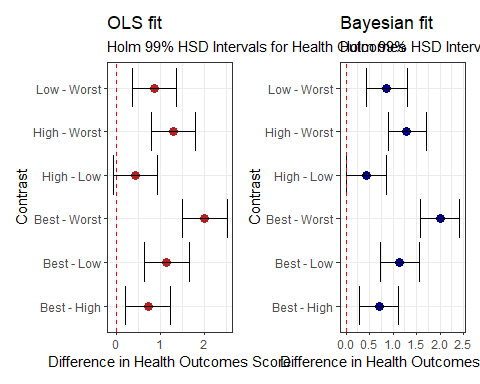
con\_ols <- as\_tibble(estimate\_contrasts(model\_one, ci = 0.99,   
 p\_adjust = "Holm")) |>   
 mutate(contr = str\_c(Level1, " - ", Level2))

No variable was specified for contrast estimation. Selecting `contrast =  
 "  
 behavior  
 "`.

con\_bay <- as\_tibble(estimate\_contrasts(model\_1b, ci = 0.99,   
 p\_adjust = "Holm")) |>   
 mutate(contr = str\_c(Level1, " - ", Level2))

No variable was specified for contrast estimation. Selecting `contrast =  
 "  
 behavior  
 "`.

p1 <- ggplot(con\_ols, aes(y = contr, x = Difference)) +  
 geom\_point(size = 3, color = "firebrick") +  
 geom\_errorbar(aes(xmin = CI\_low, xmax = CI\_high)) +  
 geom\_vline(xintercept = 0, col = "red", lty = "dashed") +  
 labs(title = "OLS fit",  
 subtitle = "Holm 99% HSD Intervals for Health Outcomes",  
 y = "Contrast", x = "Difference in Health Outcomes Score")  
  
p2 <- ggplot(con\_bay, aes(y = contr, x = Difference)) +  
 geom\_point(size = 3, color = "navy") +  
 geom\_errorbar(aes(xmin = CI\_low, xmax = CI\_high)) +  
 geom\_vline(xintercept = 0, col = "red", lty = "dashed") +  
 labs(title = "Bayesian fit",  
 subtitle = "Holm 99% HSD Intervals for Health Outcomes",  
 y = "Contrast", x = "Difference in Health Outcomes Score")  
  
p1 + p2



## K Samples: Comparing Means

1. What is the outcome under study?
2. What are the (in this case, ) treatment/exposure groups?
3. Were the data in fact collected using independent samples?
4. Are the data random samples from the population(s) of interest? Or is there at least a reasonable argument for generalizing from the samples to the population(s)?
5. What is the confidence level we require?
6. Are we doing one-sided or two-sided testing? (usually 2-sided)
7. What does the distribution of each individual sample tell us about which inferential procedure to use?
8. Are there meaningful differences between population means?
9. Can we identify pairwise comparisons of means that show meaningful differences using an appropriate procedure that protects against Type I error expansion due to multiple comparisons?

## Session Information

xfun::session\_info()

R version 4.4.1 (2024-06-14 ucrt)  
Platform: x86\_64-w64-mingw32/x64  
Running under: Windows 11 x64 (build 22631)  
  
Locale:  
 LC\_COLLATE=English\_United States.utf8   
 LC\_CTYPE=English\_United States.utf8   
 LC\_MONETARY=English\_United States.utf8  
 LC\_NUMERIC=C   
 LC\_TIME=English\_United States.utf8   
  
Package version:  
 abind\_1.4-8 askpass\_1.2.0 backports\_1.5.0   
 base64enc\_0.1-3 bayesplot\_1.11.1 bayestestR\_0.14.0   
 BH\_1.84.0.0 bit\_4.0.5 bit64\_4.0.5   
 blob\_1.2.4 boot\_1.3-31 broom\_1.0.6   
 bslib\_0.8.0 cachem\_1.1.0 callr\_3.7.6   
 car\_3.1-2 carData\_3.0-5 cellranger\_1.1.0   
 checkmate\_2.3.2 cli\_3.6.3 clipr\_0.8.0   
 cluster\_2.1.6 coda\_0.19-4.1 codetools\_0.2-20   
 colorspace\_2.1-1 colourpicker\_1.3.0 commonmark\_1.9.1   
 compiler\_4.4.1 conflicted\_1.2.0 correlation\_0.8.5   
 cowplot\_1.1.3 cpp11\_0.5.0 crayon\_1.5.3   
 crosstalk\_1.2.1 curl\_5.2.2 data.table\_1.16.0   
 datasets\_4.4.1 datawizard\_0.12.3 DBI\_1.2.3   
 dbplyr\_2.5.0 Deriv\_4.1.6 desc\_1.4.3   
 digest\_0.6.37 distributional\_0.5.0 doBy\_4.6.22   
 dplyr\_1.1.4 DT\_0.33 dtplyr\_1.3.1   
 dygraphs\_1.1.1.6 easystats\_0.7.3 effectsize\_0.8.9   
 emmeans\_1.10.4 estimability\_1.5.1 evaluate\_1.0.0   
 fansi\_1.0.6 farver\_2.1.2 fastmap\_1.2.0   
 fontawesome\_0.5.2 forcats\_1.0.0 foreign\_0.8-87   
 Formula\_1.2-5 fs\_1.6.4 gargle\_1.5.2   
 generics\_0.1.3 ggplot2\_3.5.1 ggrepel\_0.9.6   
 ggridges\_0.5.6 glue\_1.7.0 googledrive\_2.1.1   
 googlesheets4\_1.1.1 graphics\_4.4.1 grDevices\_4.4.1   
 grid\_4.4.1 gridExtra\_2.3 gtable\_0.3.5   
 gtools\_3.9.5 haven\_2.5.4 highr\_0.11   
 Hmisc\_5.1-3 hms\_1.1.3 htmlTable\_2.4.3   
 htmltools\_0.5.8.1 htmlwidgets\_1.6.4 httpuv\_1.6.15   
 httr\_1.4.7 ids\_1.0.1 igraph\_2.0.3   
 inline\_0.3.19 insight\_0.20.4 isoband\_0.2.7   
 janitor\_2.2.0 jquerylib\_0.1.4 jsonlite\_1.8.9   
 knitr\_1.48 labeling\_0.4.3 later\_1.3.2   
 lattice\_0.22-6 lazyeval\_0.2.2 lifecycle\_1.0.4   
 lme4\_1.1-35.5 loo\_2.8.0 lubridate\_1.9.3   
 magrittr\_2.0.3 markdown\_1.13 MASS\_7.3-61   
 Matrix\_1.7-0 MatrixModels\_0.5.3 matrixStats\_1.4.1   
 memoise\_2.0.1 methods\_4.4.1 mgcv\_1.9-1   
 microbenchmark\_1.5.0 mime\_0.12 miniUI\_0.1.1.1   
 minqa\_1.2.8 modelbased\_0.8.8 modelr\_0.1.11   
 multcomp\_1.4-26 munsell\_0.5.1 mvtnorm\_1.3-1   
 nlme\_3.1-164 nloptr\_2.1.1 nnet\_7.3-19   
 numDeriv\_2016.8.1.1 openssl\_2.2.1 parallel\_4.4.1   
 parameters\_0.22.2 patchwork\_1.3.0 pbkrtest\_0.5.3   
 performance\_0.12.3 pillar\_1.9.0 pkgbuild\_1.4.4   
 pkgconfig\_2.0.3 plyr\_1.8.9 posterior\_1.6.0   
 prettyunits\_1.2.0 processx\_3.8.4 progress\_1.2.3   
 promises\_1.3.0 ps\_1.8.0 purrr\_1.0.2   
 quantreg\_5.98 QuickJSR\_1.3.1 R6\_2.5.1   
 ragg\_1.3.2 rappdirs\_0.3.3 RColorBrewer\_1.1.3   
 Rcpp\_1.0.13 RcppEigen\_0.3.4.0.2 RcppParallel\_5.1.9   
 readr\_2.1.5 readxl\_1.4.3 rematch\_2.0.0   
 rematch2\_2.1.2 report\_0.5.9 reprex\_2.1.1   
 reshape2\_1.4.4 rlang\_1.1.4 rmarkdown\_2.28   
 rpart\_4.1.23 rstan\_2.32.6 rstanarm\_2.32.1   
 rstantools\_2.4.0 rstudioapi\_0.16.0 rvest\_1.0.4   
 sandwich\_3.1-1 sass\_0.4.9 scales\_1.3.0   
 see\_0.9.0 selectr\_0.4.2 shiny\_1.9.1   
 shinyjs\_2.1.0 shinystan\_2.6.0 shinythemes\_1.2.0   
 snakecase\_0.11.1 sourcetools\_0.1.7.1 SparseM\_1.84.2   
 splines\_4.4.1 StanHeaders\_2.32.10 stats\_4.4.1   
 stats4\_4.4.1 stringi\_1.8.4 stringr\_1.5.1   
 survival\_3.7-0 sys\_3.4.2 systemfonts\_1.1.0   
 tensorA\_0.36.2.1 textshaping\_0.4.0 TH.data\_1.1-2   
 threejs\_0.3.3 tibble\_3.2.1 tidyr\_1.3.1   
 tidyselect\_1.2.1 tidyverse\_2.0.0 timechange\_0.3.0   
 tinytex\_0.53 tools\_4.4.1 tzdb\_0.4.0   
 utf8\_1.2.4 utils\_4.4.1 uuid\_1.2.1   
 V8\_5.0.1 vctrs\_0.6.5 viridis\_0.6.5   
 viridisLite\_0.4.2 vroom\_1.6.5 withr\_3.0.1   
 xfun\_0.47 xml2\_1.3.6 xtable\_1.8-4   
 xts\_0.14.0 yaml\_2.3.10 zoo\_1.8-12