431 Class 15

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

2024-10-15

## Today’s Agenda

* Ingesting the Favorite Movies 2024-10-15 data from our Shared Folder in Google Drive
* Creating an Analytic Data Set

### Three Analyses using the Favorite Movies Data

1. Quantitative Outcome and a Quantitative Predictor
2. Quantitative Outcome and a Categorical Predictor
3. Comparing Means of a Quantity with Matched Samples

## Today’s Packages

library(janitor)  
library(glue)  
library(googlesheets4)  
library(infer)  
library(patchwork)  
library(rstanarm)  
library(xfun)  
library(easystats)  
library(tidyverse)  
  
theme\_set(theme\_light())  
  
source("c15/data/Love-431.R")

## Ingest the movies\_2024-10-15 data

gs4\_deauth()  
  
url <- "https://docs.google.com/spreadsheets/d/16fm1693sFjau9sIM-ORamxLL3ZMrBeRSxXqWpP8IBDs/edit?gid=0#gid=0"  
  
mov\_raw <- read\_sheet(url, na = c("", "NA"))

✔ Reading from "movies\_2024-10-15".

✔ Range '2024-10-15 Data'.

mov\_cleaning <- mov\_raw |>   
 janitor::clean\_names() |>  
 mutate(across(where(is.character), as\_factor)) |>  
 mutate(across(c(mov\_id, movie, director, star\_1, star\_2, star\_3, origin,  
 fc\_link, rt\_link, imdb\_cats, synopsis, imdb\_id, imdb\_link),  
 as.character))  
  
dim(mov\_cleaning)

[1] 228 80

## Select 35 Variables for Analyses

### 15 Quantitative Variables

v\_quant <- c("year", "length", "imdb\_ratings", "imdb\_stars", "imdb\_pct10",  
 "metascore", "awards", "budget", "gross\_world", "fc\_pctwins",   
 "rt\_critic", "rt\_audience", "theaters", "box\_2023", "triggers")

### 10 Binary Categorical Variables

v\_binary <- c("gen\_1", "color", "dr\_love", "drama", "comedy",  
 "st\_apple", "st\_prime", "st\_disney", "st\_max", "st\_paramount")

### 10 Multi-Categorical Variables or Small Counts

v\_multi <- c("mpa", "lang\_1", "list\_24", "bw\_rating", "oscars",   
 "ebert", "stream\_n", "kim\_sn", "kim\_vg", "kim\_lang")

## Create mov\_a with 37 variables

mov\_a <- mov\_cleaning |> select(mov\_id, movie,   
 all\_of(v\_quant), all\_of(v\_binary),   
 all\_of(v\_multi))  
  
dim(mov\_a)

[1] 228 37

## Check Variable Types

glimpse(mov\_a)

Rows: 228  
Columns: 37  
$ mov\_id <chr> "M-001", "M-002", "M-003", "M-004", "M-005", "M-006", "M-…  
$ movie <chr> "3 Idiots", "8 1/2", "10 Things I Hate About You", "2001:…  
$ year <dbl> 2009, 1963, 1999, 1968, 2009, 2013, 1979, 1984, 2009, 201…  
$ length <dbl> 170, 138, 97, 149, 119, 123, 117, 160, 162, 149, 181, 116…  
$ imdb\_ratings <dbl> 442000, 127000, 397000, 730000, 58000, 396000, 978000, 43…  
$ imdb\_stars <dbl> 8.4, 8.0, 7.3, 8.3, 7.9, 7.8, 8.5, 8.4, 7.9, 8.4, 8.4, 8.…  
$ imdb\_pct10 <dbl> 37.9, 27.0, 15.1, 29.9, 21.9, 17.1, 22.8, 22.5, 21.0, 31.…  
$ metascore <dbl> 67, 93, 70, 84, 87, 55, 89, 87, 83, 68, 78, 87, 57, 80, 4…  
$ awards <dbl> 64, 19, 2, 18, 23, 3, 18, 43, 89, 48, 70, 25, 9, 204, 2, …  
$ budget <dbl> 6632918, NA, 30000000, 12000000, NA, 12000000, 11000000, …  
$ gross\_world <dbl> 60262836, 196375, 53479734, 66441232, 879422, 87100449, 1…  
$ fc\_pctwins <dbl> 57, 61, 42, 57, 55, 57, 65, 53, 50, 69, 68, 70, 56, 56, 3…  
$ rt\_critic <dbl> 100, 98, 71, 92, 99, 70, 98, 89, 81, 85, 94, 93, 63, 88, …  
$ rt\_audience <dbl> 93, 92, 69, 89, 84, 81, 94, 95, 82, 92, 90, 94, 85, 83, 8…  
$ theaters <dbl> 156, NA, 2311, NA, 13, 1280, 757, 802, 3461, 4474, 4662, …  
$ box\_2023 <dbl> 9050209, NA, 79136604, NA, 587511, 19847586, 255494234, 1…  
$ triggers <dbl> 12, NA, 21, 20, 0, 21, 39, 27, 47, 41, 45, 23, 19, 33, 2,…  
$ gen\_1 <fct> M, M, M, M, F, M, F, M, M, M, M, M, M, F, F, M, M, M, M, …  
$ color <fct> Color, BW, Color, Color, Color, Color, Color, Color, Colo…  
$ dr\_love <fct> No, No, No, Yes, No, No, No, Yes, No, Yes, Yes, Yes, Yes,…  
$ drama <dbl> 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, …  
$ comedy <dbl> 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, …  
$ st\_apple <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, …  
$ st\_prime <dbl> 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, …  
$ st\_disney <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, …  
$ st\_max <dbl> 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, …  
$ st\_paramount <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
$ mpa <fct> PG-13, NR, PG-13, G, TV-PG, R, R, PG, PG-13, PG-13, PG-13…  
$ lang\_1 <fct> Hindi, Italian, English, English, Persian, English, Engli…  
$ list\_24 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
$ bw\_rating <dbl> 1, 3, 3, 0, 3, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 1, 3, 3, 3, …  
$ oscars <dbl> 0, 2, 0, 1, 0, 0, 1, 8, 3, 0, 0, 1, 0, 1, 0, 0, 1, 3, 1, …  
$ ebert <dbl> NA, 4.0, 2.5, 4.0, 4.0, 2.5, 4.0, 4.0, 4.0, 2.5, 3.0, 3.5…  
$ stream\_n <dbl> 2, 4, 5, 4, 2, 3, 4, 3, 4, 4, 4, 4, 0, 4, 3, 4, 4, 2, 4, …  
$ kim\_sn <dbl> NA, NA, 4, NA, NA, 6, 3, NA, 5, 2, 1, NA, NA, 3, NA, 2, N…  
$ kim\_vg <dbl> NA, NA, 3, NA, NA, 4, 8, NA, 7, 7, 6, NA, NA, 3, NA, 2, N…  
$ kim\_lang <dbl> NA, NA, 4, NA, NA, 5, 5, NA, 4, 4, 4, NA, NA, 4, NA, 2, N…

## Quantities

mov\_a |> select(year:triggers) |> summary()

year length imdb\_ratings imdb\_stars   
 Min. :1942 Min. : 70.0 Min. : 282 Min. :3.400   
 1st Qu.:1996 1st Qu.:103.0 1st Qu.: 158000 1st Qu.:7.100   
 Median :2006 Median :117.5 Median : 367000 Median :7.700   
 Mean :2003 Mean :123.2 Mean : 555783 Mean :7.546   
 3rd Qu.:2013 3rd Qu.:138.0 3rd Qu.: 783750 3rd Qu.:8.100   
 Max. :2024 Max. :207.0 Max. :2900000 Max. :9.300   
   
 imdb\_pct10 metascore awards budget   
 Min. : 3.80 Min. : 9.00 Min. : 0.00 Min. :2.00e+05   
 1st Qu.:11.57 1st Qu.: 60.25 1st Qu.: 5.00 1st Qu.:1.30e+07   
 Median :15.55 Median : 71.00 Median : 16.00 Median :3.00e+07   
 Mean :17.57 Mean : 70.86 Mean : 35.62 Mean :5.91e+07   
 3rd Qu.:22.25 3rd Qu.: 83.00 3rd Qu.: 43.25 3rd Qu.:9.00e+07   
 Max. :59.50 Max. :100.00 Max. :405.00 Max. :3.56e+08   
 NA's :14 NA's :22   
 gross\_world fc\_pctwins rt\_critic rt\_audience   
 Min. :1.546e+03 Min. :23.00 Min. : 22.00 Min. :28.00   
 1st Qu.:4.753e+07 1st Qu.:42.00 1st Qu.: 73.00 1st Qu.:76.00   
 Median :1.710e+08 Median :52.00 Median : 86.00 Median :86.00   
 Mean :3.321e+08 Mean :50.73 Mean : 80.38 Mean :82.13   
 3rd Qu.:4.684e+08 3rd Qu.:59.00 3rd Qu.: 93.00 3rd Qu.:92.00   
 Max. :2.924e+09 Max. :79.00 Max. :100.00 Max. :98.00   
 NA's :8 NA's :1 NA's :4 NA's :1   
 theaters box\_2023 triggers   
 Min. : 10 Min. :2.345e+05 Min. : 0.00   
 1st Qu.:1352 1st Qu.:5.685e+07 1st Qu.:10.00   
 Median :2586 Median :1.594e+08 Median :24.00   
 Mean :2437 Mean :2.494e+08 Mean :25.96   
 3rd Qu.:3517 3rd Qu.:3.283e+08 3rd Qu.:41.00   
 Max. :4662 Max. :1.730e+09 Max. :77.00   
 NA's :36 NA's :31 NA's :3

## Binary Variables

mov\_a |> select(gen\_1:st\_paramount) |> summary()

gen\_1 color dr\_love drama comedy   
 M:178 Color:224 No :137 Min. :0.0000 Min. :0.0000   
 F: 50 BW : 4 Yes: 91 1st Qu.:0.0000 1st Qu.:0.0000   
 Median :1.0000 Median :0.0000   
 Mean :0.5789 Mean :0.4254   
 3rd Qu.:1.0000 3rd Qu.:1.0000   
 Max. :1.0000 Max. :1.0000   
 st\_apple st\_prime st\_disney st\_max   
 Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
 1st Qu.:1.0000 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:0.0000   
 Median :1.0000 Median :1.0000 Median :0.0000 Median :0.0000   
 Mean :0.8904 Mean :0.9079 Mean :0.1579 Mean :0.1711   
 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:0.0000   
 Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
 st\_paramount   
 Min. :0.000   
 1st Qu.:0.000   
 Median :0.000   
 Mean :0.136   
 3rd Qu.:0.000   
 Max. :1.000

## Multi-Categorical Variables

mov\_a |> select(mpa:kim\_lang) |> summary()

mpa lang\_1 list\_24 bw\_rating oscars   
 PG-13 :74 English :200 Min. :0.0000 Min. :0.000 Min. : 0.0000   
 R :67 Japanese: 8 1st Qu.:0.0000 1st Qu.:1.000 1st Qu.: 0.0000   
 PG :62 Hindi : 7 Median :0.0000 Median :3.000 Median : 0.0000   
 NR :13 Italian : 2 Mean :0.2456 Mean :2.124 Mean : 0.9123   
 G : 7 Mandarin: 2 3rd Qu.:0.0000 3rd Qu.:3.000 3rd Qu.: 1.0000   
 TV-G : 2 Persian : 1 Max. :3.0000 Max. :3.000 Max. :11.0000   
 (Other): 3 (Other) : 8 NA's :10   
 ebert stream\_n kim\_sn kim\_vg   
 Min. :1.000 Min. :0.000 Min. : 0.000 Min. : 0.000   
 1st Qu.:3.000 1st Qu.:3.000 1st Qu.: 2.000 1st Qu.: 3.000   
 Median :3.500 Median :4.000 Median : 3.000 Median : 5.000   
 Mean :3.198 Mean :3.364 Mean : 3.293 Mean : 4.924   
 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.: 5.000 3rd Qu.: 7.000   
 Max. :4.000 Max. :5.000 Max. :10.000 Max. :10.000   
 NA's :29 NA's :71 NA's :71   
 kim\_lang   
 Min. : 0.000   
 1st Qu.: 3.000   
 Median : 4.000   
 Mean : 4.204   
 3rd Qu.: 5.000   
 Max. :10.000   
 NA's :71

# Analysis 1 (Quantitative Outcome and a Quantitative Predictor)

## Analysis 1

* How strong is the association between metascore and imdb\_stars?

mov\_a1 <- mov\_a |> filter(complete.cases(metascore, imdb\_stars))  
  
mov\_a1 |> reframe(lovedist(metascore))

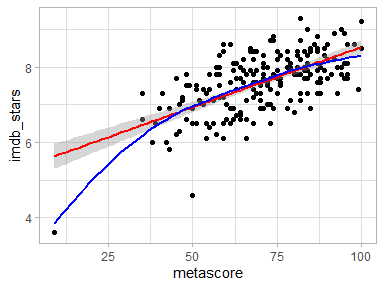
# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 214 0 70.9 15.7 71 17.8 9 60.2 83 100

mov\_a1 |> reframe(lovedist(imdb\_stars))

# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 214 0 7.60 0.789 7.7 0.741 3.6 7.1 8.1 9.3

## Plot our set of 214 movies

ggplot(mov\_a1, aes(x = metascore, y = imdb\_stars)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", formula = y ~ x, se = TRUE, col = "red") +  
 geom\_smooth(method = "loess", formula = y ~ x, se = F, col = "blue")



## Which movies are the outliers?

mov\_a1 |> filter(imdb\_stars < 5) |>   
 select(mov\_id, movie, imdb\_stars, metascore)

# A tibble: 2 × 4  
 mov\_id movie imdb\_stars metascore  
 <chr> <chr> <dbl> <dbl>  
1 M-130 Madea Goes To Jail 4.6 50  
2 M-183 The Room 3.6 9

## What if we ignored these two movies?

mov\_a1\_new <- mov\_a |> filter(imdb\_stars > 5, complete.cases(metascore))  
  
mov\_a1\_new |> reframe(lovedist(metascore))

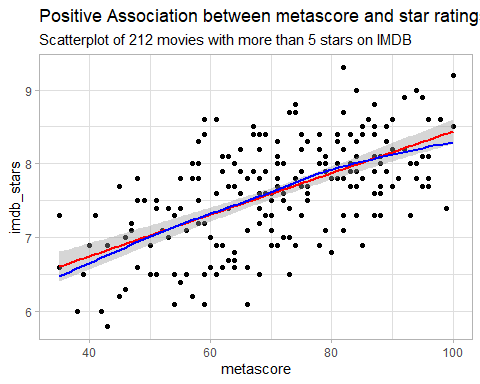
# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 212 0 71.2 15.1 71 17.8 35 61 83.2 100

mov\_a1\_new |> reframe(lovedist(imdb\_stars))

# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 212 0 7.63 0.714 7.7 0.741 5.8 7.1 8.1 9.3

## Plot our new set of 212 movies

ggplot(mov\_a1\_new, aes(x = metascore, y = imdb\_stars)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", formula = y ~ x, se = TRUE, col = "red") +  
 geom\_smooth(method = "loess", formula = y ~ x, se = F, col = "blue") +  
 labs(title = "Positive Association between metascore and star ratings",  
 subtitle = glue("Scatterplot of ", nrow(mov\_a1\_new),   
 " movies with more than 5 stars on IMDB"))



## Fit Model to our 212 Movies

set.seed(20241015)  
  
fit1 <- stan\_glm(imdb\_stars ~ metascore, data = mov\_a1\_new, refresh = 0)  
  
model\_parameters(fit1)

Warning: The `na.rm` argument is deprecated. Use `remove\_na` instead.  
Warning: The `na.rm` argument is deprecated. Use `remove\_na` instead.  
Warning: The `na.rm` argument is deprecated. Use `remove\_na` instead.

Parameter | Median | 95% CI | pd | Rhat | ESS | Prior  
------------------------------------------------------------------------------------  
(Intercept) | 5.62 | [5.25, 5.99] | 100% | 1.000 | 4032.00 | Normal (7.63 +- 1.78)  
metascore | 0.03 | [0.02, 0.03] | 100% | 1.000 | 4114.00 | Normal (0.00 +- 0.12)

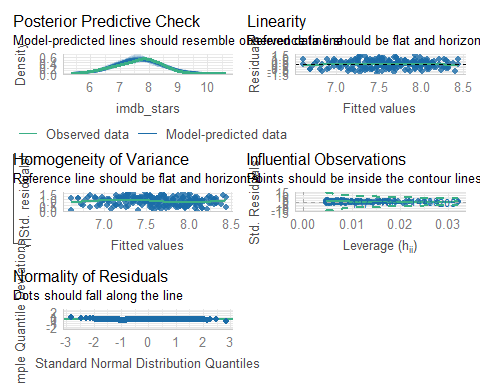
Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a MCMC distribution approximation.

model\_performance(fit1)

# Indices of model performance  
  
ELPD | ELPD\_SE | LOOIC | LOOIC\_SE | WAIC | R2 | R2 (adj.) | RMSE | Sigma  
-------------------------------------------------------------------------------------  
-185.060 | 8.606 | 370.120 | 17.211 | 370.112 | 0.356 | 0.346 | 0.571 | 0.575

## Check the model

check\_model(fit1)



# Analysis 2 (Quantitative Outcome and a Categorical Predictor)

## Analysis 2

* Are higher fc\_pctwins scores associated with higher levels of bw\_score?

| Variable | Description |
| --- | --- |
| fc\_pctwins | % of matchups won on flickchart |
| bw\_rating | Bechdel-Wallace Test Criteria Met (0-3) |

* bw\_rating counts these standards: (1) The movie has to have at least two named women in it. (2) Who talk to each other (3) About something besides a man.

## Create ordered factor bw\_score

Here, we’ll treat the bw\_rating as a factor.

mov\_a2 <- mov\_a |>   
 filter(complete.cases(fc\_pctwins, bw\_rating)) |>  
 mutate(bw\_score = factor(bw\_rating))  
  
mov\_a2 |> count(bw\_score)

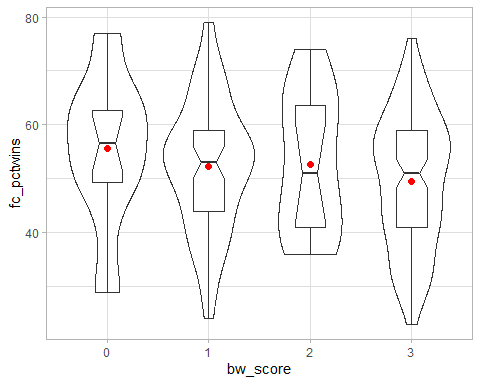
# A tibble: 4 × 2  
 bw\_score n  
 <fct> <int>  
1 0 18  
2 1 61  
3 2 15  
4 3 124

mov\_a2 |> reframe(lovedist(fc\_pctwins))

# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 218 0 51.1 12.3 52 13.3 23 42 59.8 79

## Plot of these 218 movies

ggplot(mov\_a2, aes(x = bw\_score, y = fc\_pctwins)) +  
 geom\_violin() +  
 geom\_boxplot(width = 0.3, notch = TRUE) +  
 stat\_summary(fun = "mean", geom = "point", size = 2, col = "red")



## Fit ANOVA model

fit2 <- lm(fc\_pctwins ~ bw\_score, data = mov\_a2)  
anova(fit2)

Analysis of Variance Table  
  
Response: fc\_pctwins  
 Df Sum Sq Mean Sq F value Pr(>F)  
bw\_score 3 778 259.48 1.7186 0.1642  
Residuals 214 32311 150.99

eta\_squared(fit2)

For one-way between subjects designs, partial eta squared is equivalent  
 to eta squared. Returning eta squared.

# Effect Size for ANOVA  
  
Parameter | Eta2 | 95% CI  
-------------------------------  
bw\_score | 0.02 | [0.00, 1.00]  
  
- One-sided CIs: upper bound fixed at [1.00].

## Model fit2 summaries

model\_parameters(fit2)

Parameter | Coefficient | SE | 95% CI | t(214) | p  
---------------------------------------------------------------------  
(Intercept) | 55.61 | 2.90 | [ 49.90, 61.32] | 19.20 | < .001  
bw score [1] | -3.30 | 3.30 | [ -9.80, 3.20] | -1.00 | 0.318   
bw score [2] | -2.94 | 4.30 | [-11.41, 5.52] | -0.69 | 0.494   
bw score [3] | -6.03 | 3.10 | [-12.14, 0.08] | -1.95 | 0.053

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

model\_performance(fit2)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
--------------------------------------------------------------------  
1718.366 | 1718.649 | 1735.289 | 0.024 | 0.010 | 12.174 | 12.288

## Pairwise comparisons (Holm method)

estimate\_contrasts(fit2, contrast = "bw\_score", p\_adjust = "Holm")

Marginal Contrasts Analysis  
  
Level1 | Level2 | Difference | 95% CI | SE | t(214) | p  
----------------------------------------------------------------------------  
bw\_score0 | bw\_score1 | 3.30 | [-5.48, 12.08] | 3.30 | 1.00 | > .999  
bw\_score0 | bw\_score2 | 2.94 | [-8.50, 14.38] | 4.30 | 0.69 | > .999  
bw\_score0 | bw\_score3 | 6.03 | [-2.22, 14.28] | 3.10 | 1.95 | 0.318   
bw\_score1 | bw\_score2 | -0.36 | [-9.79, 9.08] | 3.54 | -0.10 | > .999  
bw\_score1 | bw\_score3 | 2.73 | [-2.39, 7.85] | 1.92 | 1.42 | 0.784   
bw\_score2 | bw\_score3 | 3.09 | [-5.86, 12.03] | 3.36 | 0.92 | > .999  
  
Marginal contrasts estimated at bw\_score  
p-value adjustment method: Holm (1979)

# Analysis 3 (Comparing Means of a Quantity with Matched Samples)

## Audience Score vs. Critic Score

From Rotten Tomatoes, we have, for almost every movie…

* rt\_audience: Popcornmeter (audience verified ratings on scale 0-100)
* rt\_critic: Tomatometer (critic ratings on scale 0-100)

mov\_a3 <- filter(mov\_a, complete.cases(rt\_audience, rt\_critic))  
  
mov\_a3 |> select(mov\_id, movie, rt\_audience, rt\_critic) |>   
 arrange(rt\_audience) |> tail(3)

# A tibble: 3 × 4  
 mov\_id movie rt\_audience rt\_critic  
 <chr> <chr> <dbl> <dbl>  
1 M-203 Star Wars: Episode V: The Empire Strikes Back 97 95  
2 M-072 The Godfather 98 97  
3 M-191 The Shawshank Redemption 98 91

## Some Summaries for Analysis 3

mov\_a3 |> reframe(lovedist(rt\_audience))

# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 224 0 82.5 12.3 86 10.4 40 76 92 98

mov\_a3 |> reframe(lovedist(rt\_critic))

# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 224 0 80.4 17.6 86 11.9 22 73 93 100

correlation(mov\_a3 |> select(rt\_audience, rt\_critic))

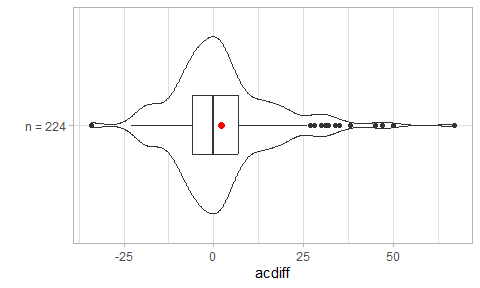
# Correlation Matrix (pearson-method)  
  
Parameter1 | Parameter2 | r | 95% CI | t(222) | p  
-------------------------------------------------------------------  
rt\_audience | rt\_critic | 0.60 | [0.51, 0.68] | 11.26 | < .001\*\*\*  
  
p-value adjustment method: Holm (1979)  
Observations: 224

## Paired Differences in Analysis 3

mov\_a3 <- mov\_a3 |> mutate(acdiff = rt\_audience - rt\_critic)  
  
mov\_a3 |> reframe(lovedist(acdiff))

# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 224 0 2.09 14.1 0 9.64 -34 -6 7 67

ggplot(mov\_a3, aes(x = acdiff, y = "n = 224")) +  
 geom\_violin() + geom\_boxplot(width = 0.3) +  
 stat\_summary(fun = "mean", geom = "point", col = "red", size = 2) +  
 labs(y = "")



## Some Extreme Differences (A3)

mov\_a3\_sorted <- mov\_a3 |>  
 select(mov\_id, movie, rt\_audience, rt\_critic, acdiff) |>  
 arrange(desc(acdiff))   
  
mov\_a3\_sorted |> head(4)

# A tibble: 4 × 5  
 mov\_id movie rt\_audience rt\_critic acdiff  
 <chr> <chr> <dbl> <dbl> <dbl>  
1 M-130 Madea Goes To Jail 96 29 67  
2 M-219 A Walk to Remember 78 28 50  
3 M-141 Memoirs of a Geisha 83 36 47  
4 M-015 Beaches 88 43 45

mov\_a3\_sorted |> tail(4)

# A tibble: 4 × 5  
 mov\_id movie rt\_audience rt\_critic acdiff  
 <chr> <chr> <dbl> <dbl> <dbl>  
1 M-124 The Lobster 65 87 -22  
2 M-197 Sorry To Bother You 70 93 -23  
3 M-029 Captain Marvel 45 79 -34  
4 M-220 War of the Worlds 42 76 -34

## Bootstrap CI

Thanks to the outliers, we’ll use a bootstrap to estimate the mean of the paired (audience - critic) rating differences.

set.seed(20241015)  
res3 <- mov\_a3 |>  
 specify(response = acdiff) |>   
 generate(reps = 2000, type = "bootstrap") |>  
 calculate(stat = "mean") |>   
 get\_confidence\_interval(level = 0.95, type = "percentile")  
  
res3 |>  
 mutate(pt\_est = mean(mov\_a3$acdiff)) |>  
 relocate(pt\_est)

# A tibble: 1 × 3  
 pt\_est lower\_ci upper\_ci  
 <dbl> <dbl> <dbl>  
1 2.09 0.335 3.92

# That’s it for the slides. Next we’ll do [Breakout Activity 3](https://github.com/THOMASELOVE/431-classes-2024/blob/main/movies/breakout3.md) for the Favorite Movies

## Session Information

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\_1.2.1 backports\_1.5.0   
 base64enc\_0.1-3 bayesplot\_1.11.1 bayestestR\_0.14.0   
 BH\_1.84.0.0 bit\_4.5.0 bit64\_4.5.2   
 blob\_1.2.4 boot\_1.3-31 broom\_1.0.7   
 bslib\_0.8.0 cachem\_1.1.0 callr\_3.7.6   
 cellranger\_1.1.0 checkmate\_2.3.2 cli\_3.6.3   
 clipr\_0.8.0 coda\_0.19-4.1 codetools\_0.2-20   
 colorspace\_2.1-1 colourpicker\_1.3.0 commonmark\_1.9.2   
 compiler\_4.4.1 conflicted\_1.2.0 correlation\_0.8.5   
 cpp11\_0.5.0 crayon\_1.5.3 crosstalk\_1.2.1   
 curl\_5.2.3 data.table\_1.16.2 datasets\_4.4.1   
 datawizard\_0.13.0 DBI\_1.2.3 dbplyr\_2.5.0   
 desc\_1.4.3 digest\_0.6.37 distributional\_0.5.0  
 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\_1.10.5 estimability\_1.5.1 evaluate\_1.0.1   
 fansi\_1.0.6 farver\_2.1.2 fastmap\_1.2.0   
 fontawesome\_0.5.2 forcats\_1.0.0 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\_1.8.0   
 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 hms\_1.1.3 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 infer\_1.0.7   
 inline\_0.3.19 insight\_0.20.5 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 matrixStats\_1.4.1 memoise\_2.0.1   
 methods\_4.4.1 mgcv\_1.9-1 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   
 numDeriv\_2016.8.1.1 openssl\_2.2.2 parallel\_4.4.1   
 parameters\_0.22.2 patchwork\_1.3.0 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 QuickJSR\_1.4.0   
 R6\_2.5.1 ragg\_1.3.3 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 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 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\_3.4.3 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 viridisLite\_0.4.2 vroom\_1.6.5   
 withr\_3.0.1 xfun\_0.48 xml2\_1.3.6   
 xtable\_1.8-4 xts\_0.14.0 yaml\_2.3.10   
 zoo\_1.8-12