# 431 Class 22

thomaselove.github.io/431

2021-11-11

# Today's Agenda: Some Loose Ends

- Working with the Favorite Movies data
- When is a complete-case analysis reasonable? MCAR!
- ANOVA
  - Assessing Assumptions with Data Visualizations
  - Dealing with Multiple Comparisons via the Holm approach
  - Kruskal-Wallis rank-based alternative
- Wilcoxon rank-based procedures for comparing pseudo-medians
  - in paired samples (Wilcoxon signed rank)
  - in independent samples (Wilcoxon rank sum)

# Today's R Setup, Part 1

```
library(knitr); library(magrittr); library(naniar)
library(broom); library(patchwork)

library(googlesheets4)
source("data/Love-boost.R")
```

```
library(janitor)
```

```
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

# Today's R Setup, Part 2

```
library(tidyverse)
```

```
-- Attaching packages --------------- tidyverse 1.3.1 --

v ggplot2 3.3.5     v purrr   0.3.4
v tibble  3.1.5     v dplyr   1.0.7
v tidyr   1.1.4     v stringr 1.4.0
v readr   2.0.2     v forcats 0.5.1

-- Conflicts ------------------ tidyverse_conflicts() --
x tidyr::extract()   masks magrittr::extract()
x dplyr::filter()    masks stats::filter()
x dplyr::lag()       masks stats::lag()
x purrr::set_names() masks magrittr::set_names()
```

- The 8 core tidyverse packages are listed every time you load the tidyverse (unless you tell R not to show messages.)

```
theme_set(theme_bw())
```

# Favorite Movies

```
url_raw <- "https://docs.google.com/spreadsheets/d/1t4668vGN-9

movie_raw <- read_sheet(url_raw, na = c("", "NA")) %>%
    arrange(film_id) %>%
    clean_names()

dim(movie_raw)
```

```
[1] 115  69
```

```
write_rds(movie_raw, "data/movie_raw.Rds")
```

# Or, more simply, since we have the .Rds file

```
movie_raw <- read_rds("data/movie_raw.Rds")

dim(movie_raw)
```

```
[1] 115  69
```

# Missing Data?

```
miss_var_summary(movie_raw)
```

```
# A tibble: 69 x 3
   variable         n_miss pct_miss
   <chr>             <int>    <dbl>
 1 country2             66    57.4
 2 imdb_cat3            33    28.7
 3 locations            14    12.2
 4 budget_est           13    11.3
 5 prod_co_2            12    10.4
 6 imdb_cat2            10     8.70
 7 domestic_gross        6     5.22
 8 worldwide_gross       3     2.61
 9 rt_critics            2     1.74
10 bom_link              2     1.74
# ... with 59 more rows
```

# Picking Out A Few Variables

```
movie1 <- movie_raw %>%
    select(film_id, film, imdb_stars, mpa)

head(movie1) %>% kable()
```

| film_id | film | imdb_stars | mpa |
|---|---|---|---|
| 1 | 8 1/2 | 8.0 | NR |
| 2 | 2001: A Space Odyssey | 8.3 | G |
| 3 | About Time | 7.8 | R |
| 4 | Avatar | 7.8 | PG-13 |
| 5 | Avengers: Endgame | 8.4 | PG-13 |
| 6 | Avengers: Infinity War | 8.4 | PG-13 |

# How many films in each mpa category?

```
movie1 %>% tabyl(mpa)
```

```
   mpa  n    percent
     G  3 0.02608696
    NR  7 0.06086957
    PG 30 0.26086957
 PG-13 33 0.28695652
     R 42 0.36521739
```

Let's look at the three categories with at least 30 films.

```
movie1 <- movie1 %>%
    filter(mpa %in% c("PG", "PG-13", "R"))

nrow(movie1)
```

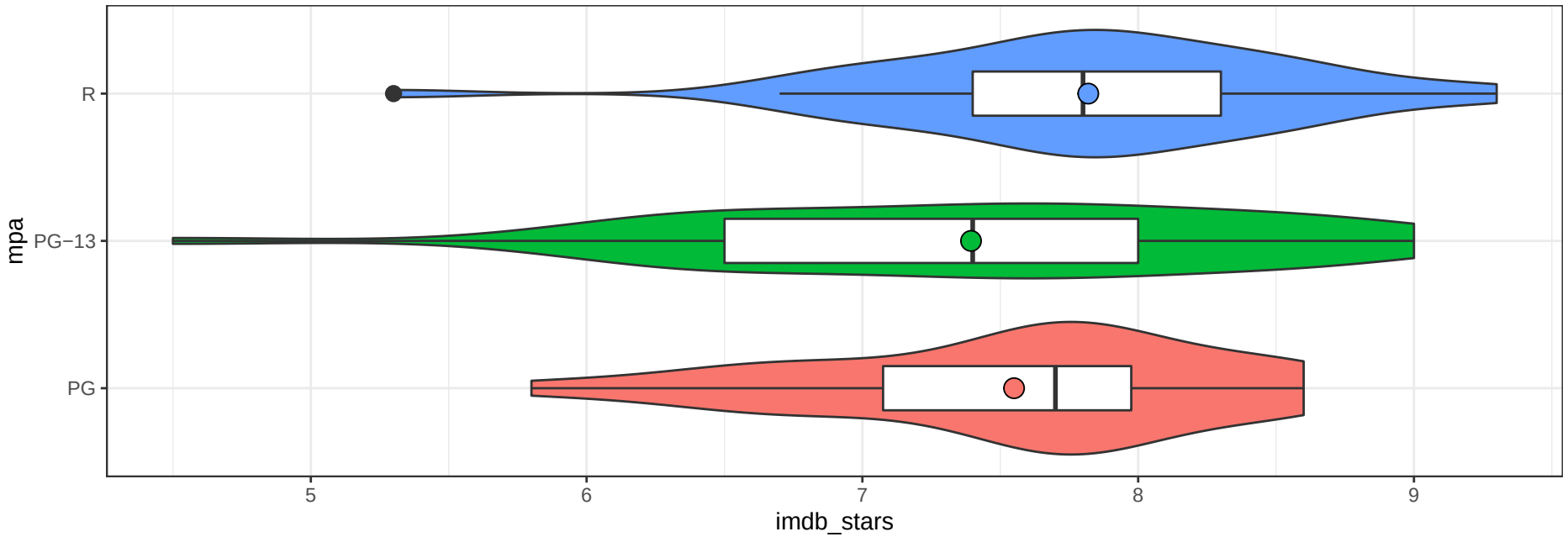
```
[1] 105
```

# ANOVA assumptions

1. Approximately Normal distribution of outcome within each group
   - Can use a rank-based alternative (Kruskal-Wallis) if there is a serious concern, or consider a transformation of the outcome
2. Equal population variance of outcome within each group (extension of pooled t)
3. Independently drawn samples of the outcome within each group

# Plot IMDB Stars by MPA Rating

```
ggplot(movie1, aes(x = mpa, y = imdb_stars)) +
    geom_violin(aes(fill = mpa)) +
    geom_boxplot(width = 0.3, outlier.size = 3) +
    stat_summary(aes(fill = mpa), fun = mean,
                 geom="point", pch = 21, size = 4) +
    guides(fill = "none") + coord_flip()
```



# Numerical Summary of IMDB Stars by MPA Rating

```
mosaic::favstats(imdb_stars ~ mpa, data = movie1) %>%
    kable(digits = 2)
```

| mpa | min | Q1 | median | Q3 | max | mean | sd | n | missing |
|---|---|---|---|---|---|---|---|---|---|
| PG | 5.8 | 7.08 | 7.7 | 7.97 | 8.6 | 7.55 | 0.72 | 30 | 0 |
| PG-13 | 4.5 | 6.50 | 7.4 | 8.00 | 9.0 | 7.39 | 1.01 | 33 | 0 |
| R | 5.3 | 7.40 | 7.8 | 8.30 | 9.3 | 7.82 | 0.74 | 42 | 0 |

# Analysis of Variance of IMDB Stars by MPA

```
mod1 <- lm(imdb_stars ~ mpa, data = movie1)
anova(mod1) %>% kable(digits = 3)
```

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| mpa | 2 | 3.484 | 1.742 | 2.529 | 0.085 |
| Residuals | 102 | 70.259 | 0.689 | NA | NA |

```
tidy(mod1, conf.int = TRUE, conf.level = 0.90) %>%
    select(term, estimate, std.error, conf.low, conf.high) %>%
    kable(digits = 3)
```

| term | estimate | std.error | conf.low | conf.high |
|---|---|---|---|---|
| (Intercept) | 7.550 | 0.152 | 7.298 | 7.802 |
| mpaPG-13 | -0.156 | 0.209 | -0.504 | 0.191 |
| mpaR | 0.269 | 0.198 | -0.060 | 0.598 |

# Bonferroni pairwise comparisons for mpa groups

```
movie1 %$%
    pairwise.t.test(imdb_stars, mpa,
                    p.adjust.method = "bonferroni")
```

```
    Pairwise comparisons using t tests with pooled SD

data:  imdb_stars and mpa

      PG   PG-13
PG-13 1.00 -
R     0.53 0.09

P value adjustment method: bonferroni
```

# Holm pairwise comparisons across mpa groups

- Works well even with an unbalanced design so long as ANOVA assumptions hold.
- Not as conservative as Bonferroni, but uniformly more powerful.

```
movie1 %$%
    pairwise.t.test(imdb_stars, mpa,
                    p.adjust.method = "holm")
```

```
    Pairwise comparisons using t tests with pooled SD

data:  imdb_stars and mpa

      PG   PG-13
PG-13 0.46 -
R     0.36 0.09

P value adjustment method: holm
```

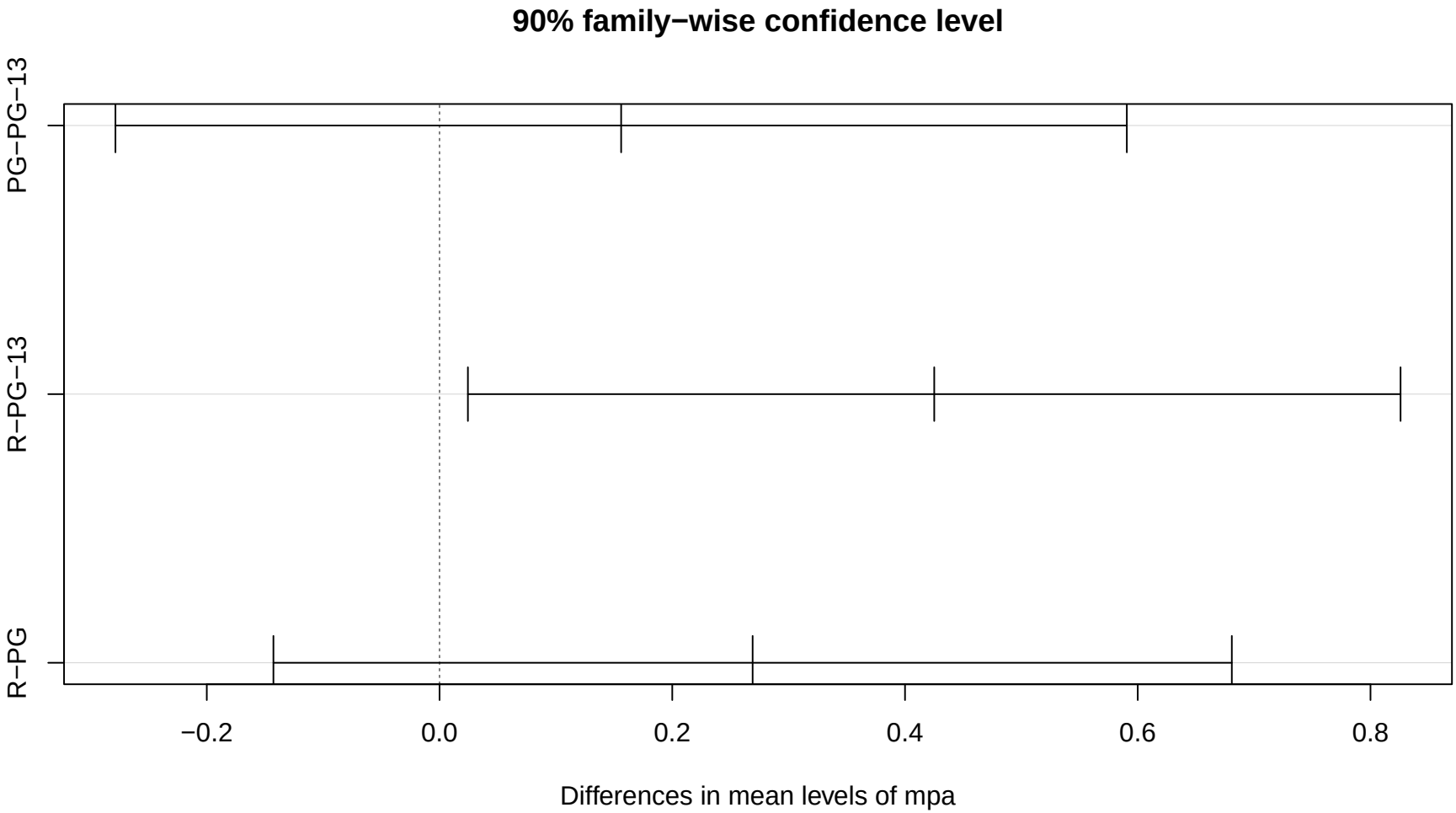
# Tukey HSD set of Pairwise Comparisons

```
tukey1 <- movie1 %$%
    TukeyHSD(aov(imdb_stars ~ mpa),
             ordered = TRUE, conf.level = 0.90)

tidy(tukey1) %>% kable(digits = 3)
```

| term | contrast | null.value | estimate | conf.low | conf.high | adj.p.value |
|---|---|---|---|---|---|---|
| mpa | PG-PG-13 | 0 | 0.156 | -0.279 | 0.591 | 0.737 |
| mpa | R-PG-13 | 0 | 0.425 | 0.024 | 0.826 | 0.076 |
| mpa | R-PG | 0 | 0.269 | -0.143 | 0.681 | 0.368 |

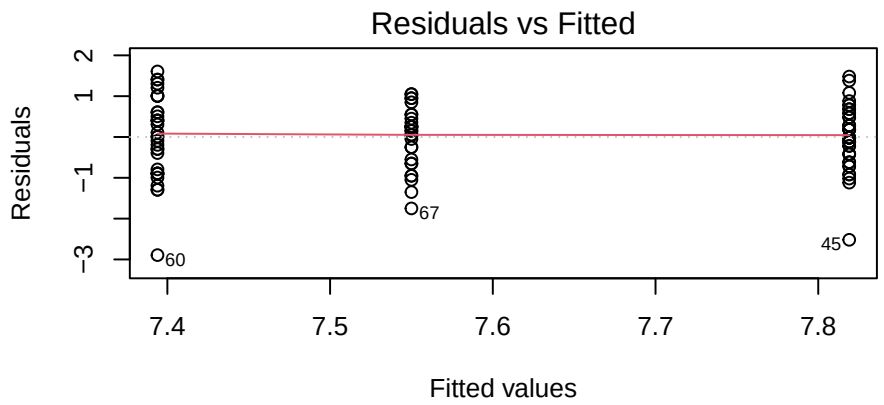
# Tukey HSD set of Pairwise Comparisons

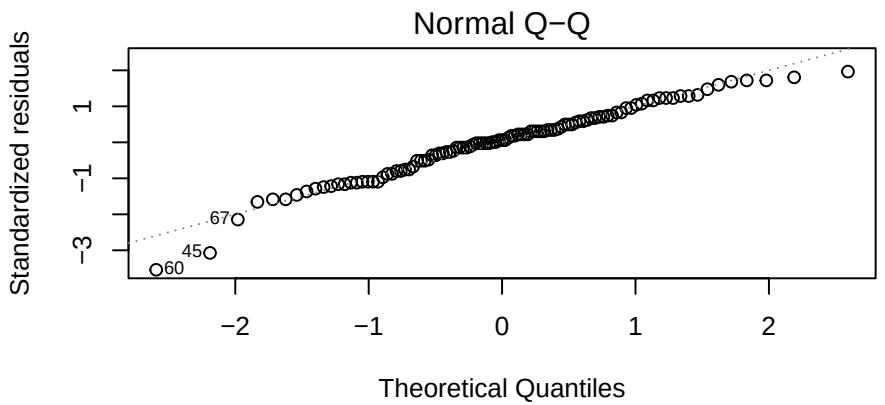
```
plot(tukey1)
```

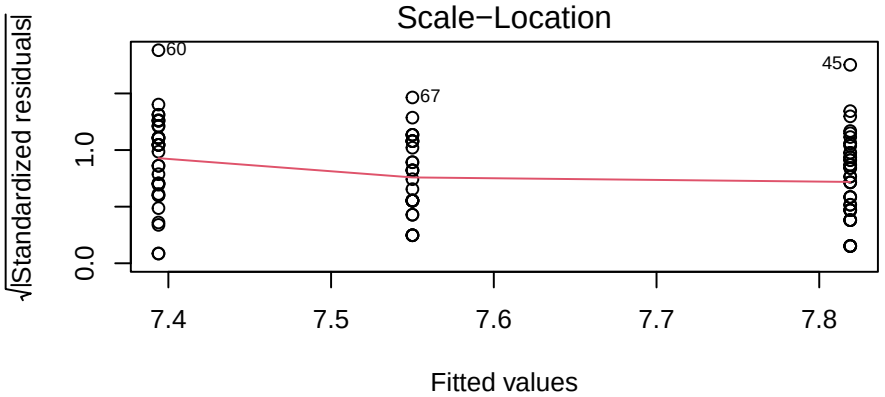


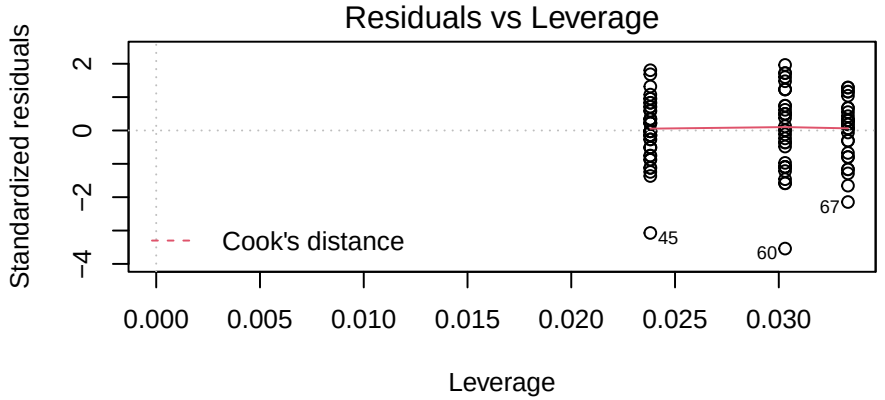
# Residual Analysis for our ANOVA model

```
par(mfrow = c(2,2)); plot(mod1); par(mfrow = c(1,1))
```



# Kruskal-Wallis Rank-Based ANOVA Approach

```
kruskal.test(imdb_stars ~ mpa, data = movie1)
```

```
    Kruskal-Wallis rank sum test

data:  imdb_stars by mpa
Kruskal-Wallis chi-squared = 4.1131, df = 2,
p-value = 0.1279
```

- No longer comparing means, and no confidence intervals here.
- No straightforward decision about what to do about pairwise comparisons, other than Holm-based comparisons based on Wilcoxon rank sum tests.
- Speaking of Wilcoxon rank-based tests. . .

# A New Question...

Are students in 431 more likely to have seen movies that were nominated for Academy Awards?

- How many of the 115 movies received Oscar nominations?

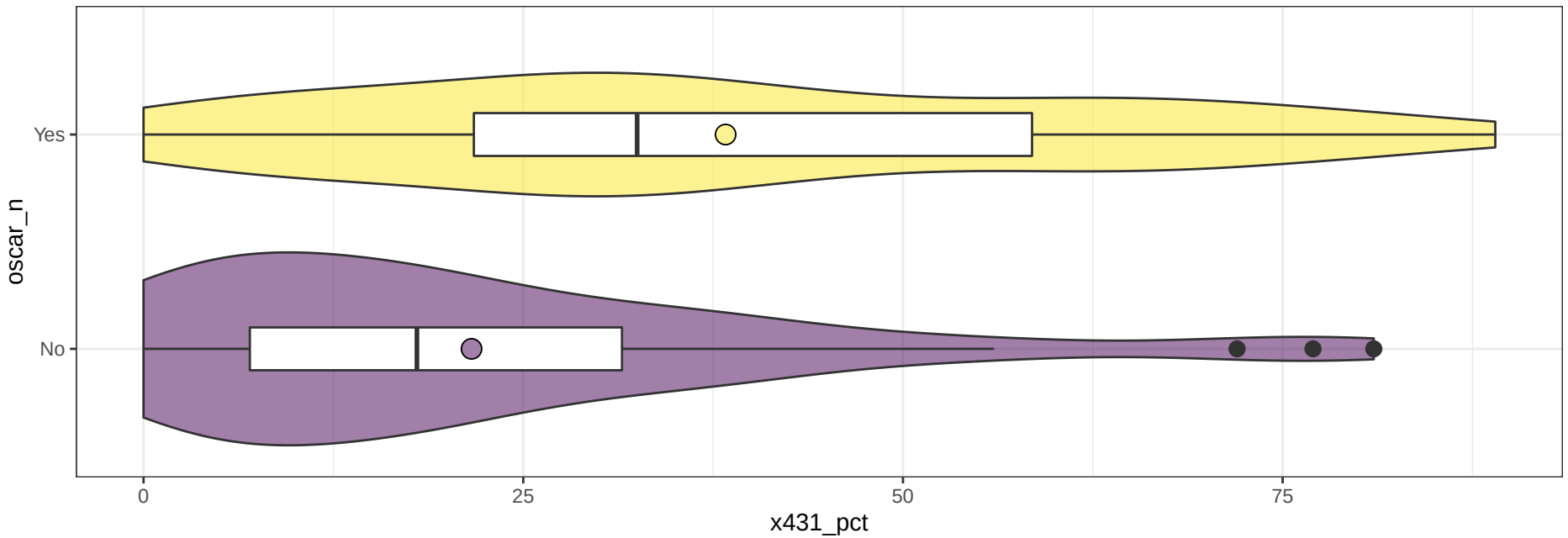
```
movie2 <- movie_raw %>%
    select(film_id, film, x431_pct, oscar_noms) %>%
    mutate(oscar_n = factor(
        ifelse(oscar_noms > 0, "Yes", "No")))
```

```
movie2 %>% tabyl(oscar_n)
```

```
 oscar_n  n   percent
      No 47 0.4086957
     Yes 68 0.5913043
```

# % of students who've seen film, by Oscar Status?

```
ggplot(movie2, aes(x = oscar_n, y = x431_pct)) +
    geom_violin(aes(fill = oscar_n)) +
    geom_boxplot(width = 0.2, outlier.size = 3) +
    stat_summary(aes(fill = oscar_n), fun = mean,
                 geom="point", pch = 21, size = 4) +
    guides(fill = "none") + coord_flip() +
    scale_fill_viridis_d(alpha = 0.5)
```



# Comparing Two Population Means

Does this design involve paired samples, or independent samples? Why?

```
mosaic::favstats(x431_pct ~ oscar_n, data = movie2) %>%
    kable(digits = 2)
```

| oscar_n | min | Q1 | median | Q3 | max | mean | sd | n | missing |
|---|---|---|---|---|---|---|---|---|---|
| No | 0 | 7.00 | 18.0 | 31.5 | 81 | 21.60 | 20.38 | 47 | 0 |
| Yes | 0 | 21.75 | 32.5 | 58.5 | 89 | 38.32 | 24.22 | 68 | 0 |

# Comparing 431 % by Oscar Nomination Status

We have four different approaches. Two are based on the t distribution.

```
# pooled t
compA <- t.test(x431_pct ~ oscar_n, data = movie2,
                var.equal = TRUE, conf.level = 0.90)

tidy(compA) %>%
    select(estimate1, estimate2, estimate,
           conf.low, conf.high) %>% kable(dig = 2)
```

| estimate1 | estimate2 | estimate | conf.low | conf.high |
|---|---|---|---|---|
| 21.6 | 38.32 | -16.73 | -23.88 | -9.58 |

# Comparing 431 % by Oscar Nomination Status

Here's the second one based on the t distribution.

```
# Welch's t
compB <- t.test(x431_pct ~ oscar_n, data = movie2,
                conf.level = 0.90)

tidy(compB) %>%
    select(estimate1, estimate2, estimate,
           conf.low, conf.high) %>% kable(dig = 2)
```

| estimate1 | estimate2 | estimate | conf.low | conf.high |
|---|---|---|---|---|
| 21.6 | 38.32 | -16.73 | -23.66 | -9.79 |

# Comparing 431 % by Oscar Nomination Status

This strategy doesn't use the t distribution.

```
# bootstrap
set.seed(431)
compC <- movie2 %$% bootdif(x431_pct, oscar_n,
                            conf.level = 0.90)

compC
```

```
Mean Difference             0.05             0.95
      16.727785         9.971558        23.421621
```

# Comparing 431 % by Oscar Nomination Status

What if we rank the observations from low to high in each group, then compare the results?

```
# Wilcoxon-Mann-Whitney rank sum with continuity correction
compD <- wilcox.test(x431_pct ~ oscar_n, data = movie2,
                     conf.int = TRUE, conf.level = 0.90)

tidy(compD) %>% select(estimate, conf.low, conf.high) %>%
    kable(digits = 2)
```

| estimate | conf.low | conf.high |
|---|---|---|
| -17 | -24 | -10 |

# Interpreting the Rank Sum Test

The Wilcoxon-Mann-Whitney rank sum test (also called the Mann-Whitney U test and lots of other things) tests the null hypothesis that if we randomly select X and Y from the two populations of interest, the probability of X being greater than Y is the same as the probability of Y being greater than X.

- The "Pseudomedian" is sometimes referred to as the Hodges-Lehmann estimate. It is the median of all possible differences between an observation in the first sample and an observation in the second sample.

To write up a Wilcoxon rank sum result, we usually suggest specifying the two medians directly, and then describing the *p* value or (less commonly) a confidence interval.

The Wilcoxon-Mann-Whitney test requires only that the data be ordinal, and this reduces the influence of outliers. It doesn't test the same thing as the t test (or bootstrap) however.

# Comparing 431 % by Oscar Nomination Status

**Four Comparisons for these Independent Samples**

| Method | Yes - No Est. | 90% CI | Statistic |
|---|---|---|---|
| Pooled t | 16.73 | (9.58, 23.88) | Mean |
| Welch's t | 16.73 | (9.79, 23.66) | Mean |
| Bootstrap | 16.73 | (9.97, 23.42) | Mean |
| Rank Sum | 17 | (10, 24) | "Pseudo-median" |

## Numerical Summaries from the Data

| oscar_n | n | mean | median | sd |
|---|---|---|---|---|
| No | 47 | 21.596 | 18.0 | 20.378 |
| Yes | 68 | 38.324 | 32.5 | 24.223 |

# Comparing A New Outcome under 2 Conditions

Suppose we want to compare the percentage of **critics** who recommend a film to the percentage of **the general audience** who recommend the film. Each film has:

- `rt_critics` = from Rotten Tomatoes: percentage of critics who recommend the film (sample mean across 113 films was 80.6)
- `rt_audience` = from Rotten Tomatoes: percentage of audience who recommend the film (sample mean across 114 films was 82.6)

Can we compare the difference between the means (at least for the 113 films with data on each variable) appropriately?

What's the design we have here?

# Creating our Third Data Set

```
movie3 <- movie_raw %>%
    filter(complete.cases(rt_critics, rt_audience)) %>%
    select(film_id, film, rt_critics, rt_audience)

dim(movie3)
```

```
[1] 113   4
```

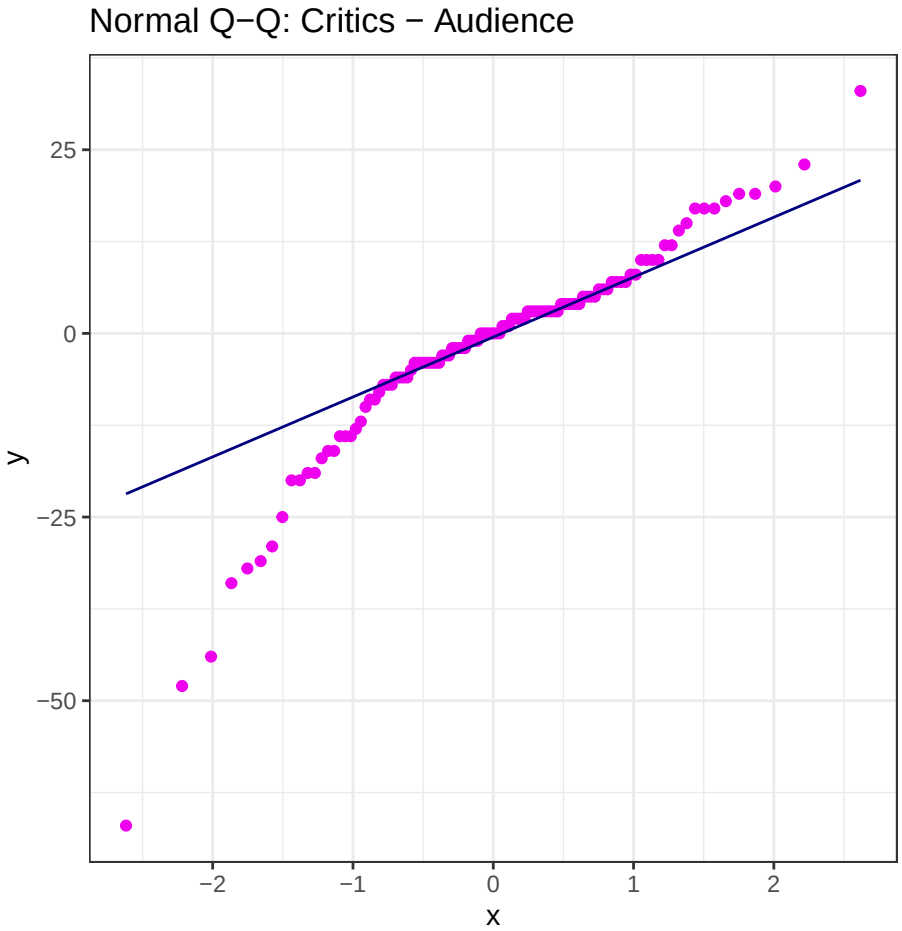
We lost two films (Farewell My Concubine and Jab We Met) which didn't have information on each of our variables.

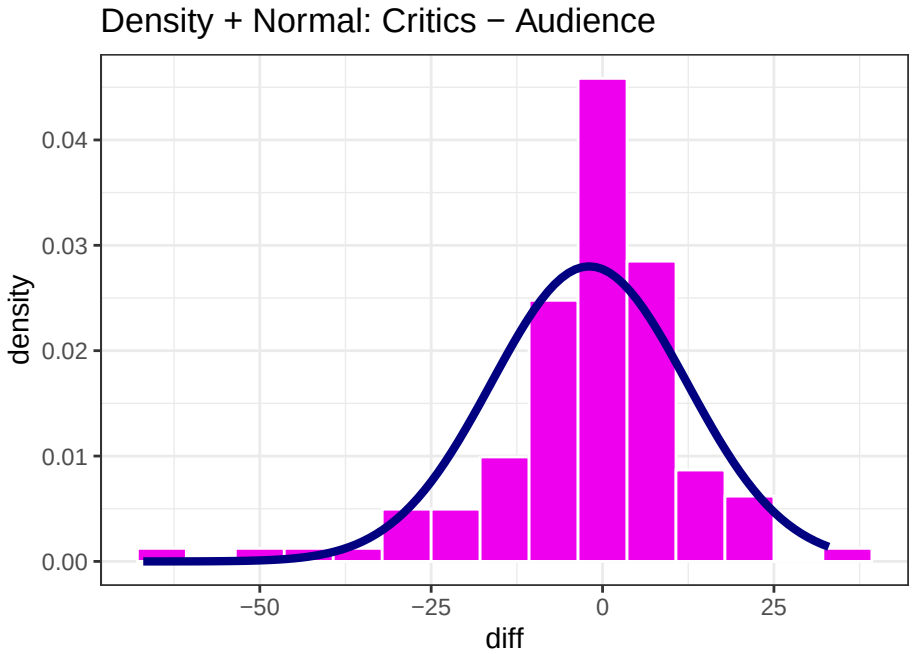
# The movie3 data

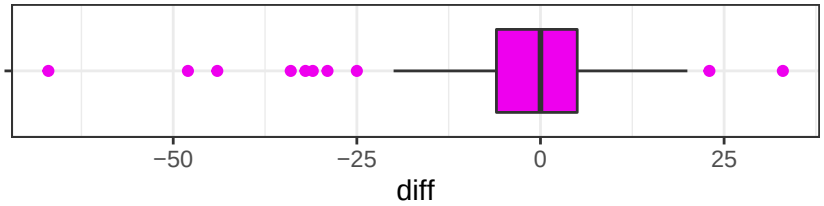
```
movie3 <- movie3 %>%
    mutate(diff = rt_critics - rt_audience)

head(movie3)
```

```
# A tibble: 6 x 5
  film_id film                 rt_critics rt_audience  diff
    <dbl> <chr>                     <dbl>       <dbl> <dbl>
1       1 8 1/2                        98          92     6
2       2 2001: A Space O~             92          89     3
3       3 About Time                   69          81   -12
4       4 Avatar                       81          82    -1
5       5 Avengers: Endga~             94          90     4
6       6 Avengers: Infin~             85          91    -6
```

We can develop paired differences in this paired samples setting. What do those differences look like?




| min | Q1 | median | Q3 | max | mean | sd | n | missing |
|---|---|---|---|---|---|---|---|---|
| -67 | -6 | 0 | 5 | 33 | -1.9 | 14.2 | 113 | 0 |

# Paired T approach

```
compX <- movie3 %$% t.test(diff, conf.level = 0.90)

tidy(compX) %>%
    select(estimate, conf.low, conf.high, p.value) %>%
    kable(dig = 2)
```

| estimate | conf.low | conf.high | p.value |
|---|---|---|---|
| -1.95 | -4.17 | 0.28 | 0.15 |

# Bootstrap approach

```
set.seed(431431)
compY <- movie3 %$%
    Hmisc::smean.cl.boot(diff, conf.int = 0.90)

compY
```

```
      Mean       Lower       Upper
-1.9469027 -4.2491150  0.1712389
```

# Wilcoxon Signed Rank approach

```
compZ <- movie3 %$%
    wilcox.test(diff, conf.int = 0.90)

tidy(compZ) %>%
    select(estimate, conf.low, conf.high, p.value) %>%
    kable(dig = 2)
```

| estimate | conf.low | conf.high | p.value |
|---|---|---|---|
| -0.5 | -3 | 1.5 | 0.62 |

# Comparing % Recommend by Critics vs. Audiences

**Three Comparisons for these Paired Samples**

| Method | Crit - Aud Est. | 90% CI | Statistic |
|---|---|---|---|
| t | -1.95 | (-4.17, 0.28) | Mean difference |
| Bootstrap | -1.95 | (-4.25, 0.17) | Mean difference |
| Signed Rank | -0.5 | (-3, 1.5) | "Pseudo-median" difference |

## Summarizing the Data

| | group | mean | sd | median | n | min | max |
|---|---|---|---|---|---|---|---|
| ...1 | critics | 80.58 | 17.61 | 86 | 113 | 21 | 100 |
| ...2 | audience | 82.53 | 12.24 | 86 | 113 | 42 | 98 |
| ...3 | c-a diffs | -1.95 | 14.25 | 0 | 113 | -67 | 33 |