431 Project Study 1 Demonstration

Thomas E. Love

2017-11-26

Contents

1 Setup in R

```
library(pander); library(mice); library(Epi)
library(gridExtra); library(vcd); library(Hmisc)
library(mosaic); library(forcats); library(tidyverse)

source("Love-boost.R")

sur15_a <- read.csv("survey2015raw_a.csv") %>% tbl_df
sur15_b <- read.csv("survey2015raw_b.csv") %>% tbl_df
```

2 What is this?

This document demonstrates the sorts of analyses we are asking you to complete in the context of Task E of your project for Study 1 (using the class survey.) We will use data from a prior class survey, gathered in two data files (called `survey2015raw_a` and `survey2015raw_b`) available on the Projects - Class Survey page of our website.

3 Merging the Data Sets

We'll begin by merging the `sur15_a` and `sur15_b` tibbles (each of which contains the linking variable `S.id`, into a data frame called `sur15_merge` following the approach taken in the Data Management section of the 431 Project Instructions after Proposal document.

```
sur15_merge <- inner_join(sur15_a, sur15_b, by = "S.id")
```

The `sur15_merge` data includes many variables we don't need, so we'll prune down to the variables we'll need in what follows...

```
sur15_m <- select(sur15_merge, S.id, r.pre, r.now, sex.x, height, weight,
                  comfort.431, grades, load.431, prior.r, english,
                  medium, fiction, seat)
sur15_m
```

```
# A tibble: 53 x 14
    S.id r.pre r.now  sex.x height weight comfort.431
   <int> <int> <int> <fctr>  <dbl>  <dbl>       <int>
 1   501     0    70   Male  68.10    162          90
 2   502     0    70   Male  67.00    151          50
 3   503     0    10 Female  62.50    127          70
 4   504    20    80 Female  22.83    140          50
 5   505    80    90   Male  70.00    178          85
 6   506     0    50   Male  73.00    145          80
 7   507    50    50   Male  74.00    320          50
 8   508    60    75   Male  70.00    165          80
 9   509     0    50 Female  64.00    135          75
10   510    30    30 Female  69.00    155          50
# ... with 43 more rows, and 7 more variables: grades <fctr>,
#   load.431 <int>, prior.r <fctr>, english <fctr>, medium <fctr>,
#   fiction <fctr>, seat <fctr>
```

4 Data Management: Building a Tidy Data Set

We're going to build a 15-variable data set, which we'll call `sur15` for this demonstration. The data set will need to contain each of the variables listed in the Codebook, below. For each variable, we'll also check to see that all of the values fall in a reasonable range (with no results that fall outside of the parameters of how we are measuring the results) and we'll identify whether there are any missing values.

4.1 Range and Missingness Checks: The seven "easy" variables

Seven variables come straight from the merged raw data in `sur15_merge`, with no further need for manipulation. All we need to do for these variables is a simple check of the range of values covered and whether or not there are any missing values. Generally, the easiest variables to deal with in cleaning have either quantitative responses or binary (No/Yes) responses, and no missing values. These are the seven variables we need that fall in that group.

1. `S.id` should have 53 unique values, specifically between 501 and 553.
2. `r.pre` should fall within the range 0-100.
3. `r.now` should also fall within the range 0-100.
4. `comfort.431` should be in 0-100.
5. `load.431` should be in 0-100.
6. `prior.r` should be No or Yes.
7. `english` should be No or Yes.

```
sur15_m %>%
  select(S.id, r.pre, r.now, comfort.431, load.431, prior.r, english) %>%
  Hmisc::describe()
```

```
.

 7  Variables      53  Observations
--------------------------------------------------------------------------------
S.id
       n  missing distinct     Info     Mean      Gmd      .05      .10
      53        0       53        1      527       18    503.6    506.2
     .25      .50      .75      .90      .95
   514.0    527.0    540.0    547.8    550.4

lowest : 501 502 503 504 505, highest: 549 550 551 552 553
--------------------------------------------------------------------------------
r.pre
       n  missing distinct     Info     Mean      Gmd      .05      .10
      53        0       12    0.799    21.13    30.36        0        0
     .25      .50      .75      .90      .95
       0        0       50       70       84

Value          0    10    15    20    25    30    40    50    60    70
Frequency     31     2     1     2     1     1     1     3     4     3
Proportion 0.585 0.038 0.019 0.038 0.019 0.019 0.019 0.057 0.075 0.057

Value         80    90
Frequency      1     3
Proportion 0.019 0.057
--------------------------------------------------------------------------------
r.now
```

```
       n  missing distinct     Info     Mean      Gmd      .05      .10
      53        0       18    0.992    56.58    30.25        6       12
     .25      .50      .75      .90      .95
      40       60       76       89       90

Value          0    10    20    25    27    30    40    50    60    61
Frequency      3     3     2     1     1     2     2     8     5     1
Proportion 0.057 0.057 0.038 0.019 0.019 0.038 0.038 0.151 0.094 0.019

Value         65    70    75    76    80    85    90    95
Frequency      1     6     4     1     6     1     4     2
Proportion 0.019 0.113 0.075 0.019 0.113 0.019 0.075 0.038
---------------------------------------------------------------------------
comfort.431
       n  missing distinct     Info     Mean      Gmd      .05      .10
      53        0       17    0.988    75.94     19.5     44.0     50.0
     .25      .50      .75      .90      .95
    70.0     80.0     90.0     94.4     95.0

Value         15    30    35    50    60    70    73    75    78    80
Frequency      1     1     1     6     1     7     1     4     1     7
Proportion 0.019 0.019 0.019 0.113 0.019 0.132 0.019 0.075 0.019 0.132

Value         85    88    89    90    92    95   100
Frequency      5     1     1     9     1     4     2
Proportion 0.094 0.019 0.019 0.170 0.019 0.075 0.038
---------------------------------------------------------------------------
load.431
       n  missing distinct     Info     Mean      Gmd      .05      .10
      53        0       18    0.987    38.23    36.55      0.0      0.0
     .25      .50      .75      .90      .95
     5.0     40.0     70.0     79.0     80.8

Value          0     2     5    10    20    30    40    45    50    60
Frequency     11     1     2     6     4     2     1     1     3     5
Proportion 0.208 0.019 0.038 0.113 0.075 0.038 0.019 0.019 0.057 0.094

Value         62    65    70    75    80    82    95   100
Frequency      1     2     6     2     3     1     1     1
Proportion 0.019 0.038 0.113 0.038 0.057 0.019 0.019 0.019
---------------------------------------------------------------------------
prior.r
       n  missing distinct
      53        0        2

Value         No   Yes
Frequency     28    25
Proportion 0.528 0.472
---------------------------------------------------------------------------
english
       n  missing distinct
      53        0        2

Value        No  Yes
```

```
Frequency    18   35
Proportion 0.34 0.66
-------------------------------------------------------------------------------
```

4.2 Reordering the No/Yes factors as Yes/No instead

You'll note that the `prior.r` and `english` variables have No listed before Yes in this output. We might want to change that, using the `fct_relevel` function from the `forcats` package. If you feed this function a factor, and a list of levels, it will store those levels first, in order, and then retain any prior ordering. So, if we want "Yes" to show up first, we'll do...

```
sur15_m$prior.r <- fct_relevel(sur15_m$prior.r, "Yes")
sur15_m$english <- fct_relevel(sur15_m$english, "Yes")

sur15_m %>%
  select(prior.r, english) %>%
  Hmisc::describe()
```

.

```
 2  Variables      53  Observations
-------------------------------------------------------------------------------
prior.r
       n  missing distinct
      53        0        2

Value        Yes    No
Frequency     25    28
Proportion 0.472 0.528
-------------------------------------------------------------------------------
english
       n  missing distinct
      53        0        2

Value       Yes   No
Frequency    35   18
Proportion 0.66 0.34
-------------------------------------------------------------------------------
```

OK. No missingness, and no values out of the range of our expectations. Good.

4.3 Categorical Variables in need of Management

Five of our remaining variables are categorical (several have more than 2 categories.) In addition to checking for missingness and inappropriate values, we want to collapse some categories, or adjust names or labeling to mirror our desired codebook. Here's the original data for this set of variables.

```
sur15_m %>%
  select(sex.x, grades, seat, medium, fiction) %>%
  Hmisc::describe()
```

.

```
 5  Variables      53  Observations
```

```
---------------------------------------------------------------------------
sex.x
       n  missing distinct
      53        0        2

Value      Female   Male
Frequency      25     28
Proportion  0.472  0.528
---------------------------------------------------------------------------
grades
       n  missing distinct
      52        1        3

A. Individual Assignments (40, 0.769), B. Partner Assignments (you and 1
other student) (6, 0.115), C. Group Assignments (you and 2 or more others)
(6, 0.115)
---------------------------------------------------------------------------
seat
       n  missing distinct
      53        0        3

A. On the left side as you face the screen (20, 0.377), B. In the middle
of the room (14, 0.264), C. On the right side (closest to the podium) (19,
0.358)
---------------------------------------------------------------------------
medium
       n  missing distinct
      53        0        4

A. Movies (17, 0.321), B. Television (22, 0.415), C. Print (including
books, comics, etc.) (9, 0.170), D. Other (5, 0.094)
---------------------------------------------------------------------------
fiction
       n  missing distinct
      53        0        5

A. Comedy (18, 0.340), B. Drama (15, 0.283), C. Action (5, 0.094), D.
Horror / Thriller (1, 0.019), E. Fantasy / Science Fiction (14, 0.264)
---------------------------------------------------------------------------
```

You'll note that we have a missing `grades` value here, and we'll eventually have to decide what we want to do about that. But, for the moment, I'll address other concerns.

4.3.1 Renaming `sex.x` as `sex`

When we merged the data, since `sex` was in both of the original files, the variable appeared in the merged file as both `sex.x` and `sex.y`. We selected `sex.x` only to keep (since the two were identical), but really, we want to recreate the variable as simply `sex`.

```
sur15_m$sex <- sur15_m$sex.x
Hmisc::describe(sur15_m$sex)
```

```
sur15_m$sex
       n  missing distinct
      53        0        2
```

```
Value      Female   Male
Frequency      25     28
Proportion  0.472  0.528
```

4.3.2 Relabeling the levels of grades

For `grades`, we want to wind up with a factor that has shorter level names, specifically: Individual, Partner and Group, in that order. We'll store the old version in `grades.old` so we can later do a little sanity check on our work, and then recode the `grades` information using the `fct_recode` function from the `forcats` package:

```
sur15_m$grades.old <- sur15_m$grades
levels(sur15_m$grades)
```

```
[1] "A. Individual Assignments"
[2] "B. Partner Assignments (you and 1 other student)"
[3] "C. Group Assignments (you and 2 or more others)"
```

```
sur15_m$grades <- fct_recode(sur15_m$grades,
                             "Individual" = "A. Individual Assignments",
                             "Partner" = "B. Partner Assignments (you and 1 other student)",
                             "Group" = "C. Group Assignments (you and 2 or more others)")
table(sur15_m$grades.old, sur15_m$grades) # sanity check
```

```
                                                Individual Partner
  A. Individual Assignments                             40       0
  B. Partner Assignments (you and 1 other student)       0       6
  C. Group Assignments (you and 2 or more others)        0       0

                                                Group
  A. Individual Assignments                         0
  B. Partner Assignments (you and 1 other student)  0
  C. Group Assignments (you and 2 or more others)   6
```

```
Hmisc::describe(sur15_m$grades)
```

```
sur15_m$grades
       n  missing distinct
      52        1        3

Value      Individual    Partner      Group
Frequency          40          6          6
Proportion      0.769      0.115      0.115
```

4.3.3 Relabeling the levels of seat

For `seat`, we want to wind up with a factor that has shorter level names, specifically: Left, Middle and Right, in that order. We'll store the old version in `seat.old` so we can later do a little sanity check on our work, and then recode the `seat` information using the `fct_recode` function from the `forcats` package:

```
sur15_m$seat.old <- sur15_m$seat
levels(sur15_m$seat)
```

```
[1] "A. On the left side as you face the screen"
[2] "B. In the middle of the room"
```

```
[3] "C. On the right side (closest to the podium)"
```

```
sur15_m$seat <- fct_recode(sur15_m$seat,
                           "Left" = "A. On the left side as you face the screen",
                           "Middle" = "B. In the middle of the room",
                           "Right" = "C. On the right side (closest to the podium)")
table(sur15_m$seat.old, sur15_m$seat, useNA = "ifany") # sanity check
```

```
                                                Left Middle Right
  A. On the left side as you face the screen     20      0     0
  B. In the middle of the room                    0     14     0
  C. On the right side (closest to the podium)    0      0    19
```

```
Hmisc::describe(sur15_m$seat)
```

```
sur15_m$seat
       n  missing distinct
      53        0        3

Value        Left Middle  Right
Frequency      20     14     19
Proportion  0.377  0.264  0.358
```

4.3.4 Collapsing and recoding levels of `medium`

For the `medium` variable, we want to collapse the Print and Other levels to form a three category variable (with levels Movies, TV and Other) called `medium.3c`.

```
levels(sur15_m$medium)
```

```
[1] "A. Movies"
[2] "B. Television"
[3] "C. Print (including books, comics, etc.)"
[4] "D. Other"
```

```
sur15_m$medium.3c <- fct_recode(sur15_m$medium,
                                "Movies" = "A. Movies",
                                "TV" = "B. Television",
                                "Other" = "C. Print (including books, comics, etc.)",
                                "Other" = "D. Other")
table(sur15_m$medium, sur15_m$medium.3c, useNA = "ifany") # sanity check
```

```
                                          Movies TV Other
  A. Movies                                   17  0     0
  B. Television                                0 22     0
  C. Print (including books, comics, etc.)     0  0     9
  D. Other                                     0  0     5
```

4.3.5 Collapsing and recoding levels of `fiction`

For the `fiction` variable, we want to form a four category variable (with levels Comedy, Drama, Fantasy/SciFi, Other) called `fiction.4c`.

```
levels(sur15_m$fiction)
```

```
[1] "A. Comedy"                    "B. Drama"
[3] "C. Action"                    "D. Horror / Thriller"
[5] "E. Fantasy / Science Fiction"
```

```
sur15_m$fiction.4c <- fct_recode(sur15_m$fiction,
                                 "Comedy" = "A. Comedy",
                                 "Drama" = "B. Drama",
                                 "Fantasy/SciFi" = "E. Fantasy / Science Fiction",
                                 "Other" = "C. Action",
                                 "Other" = "D. Horror / Thriller")
table(sur15_m$fiction, sur15_m$fiction.4c, useNA = "ifany") # sanity check
```

```
                              Comedy Drama Other Fantasy/SciFi
  A. Comedy                       18     0     0             0
  B. Drama                         0    15     0             0
  C. Action                        0     0     5             0
  D. Horror / Thriller             0     0     1             0
  E. Fantasy / Science Fiction     0     0     0            14
```

Actually, I'd like to reorder `fiction.4c` to put Other last.

```
sur15_m$fiction.4c <- fct_relevel(sur15_m$fiction.4c, "Comedy", "Drama", "Fantasy/SciFi", "Other")
```

OK. Let's see what we have now...

```
sur15_m %>%
  select(medium.3c, fiction.4c) %>%
  Hmisc::describe()
```

```
.

 2  Variables      53  Observations
---------------------------------------------------------------------------
medium.3c
       n  missing distinct
      53        0        3

Value      Movies     TV  Other
Frequency      17     22     14
Proportion  0.321  0.415  0.264
---------------------------------------------------------------------------
fiction.4c
       n  missing distinct
      53        0        4

Value             Comedy         Drama Fantasy/SciFi         Other
Frequency             18            15            14             6
Proportion         0.340         0.283         0.264         0.113
---------------------------------------------------------------------------
```

4.4 Combining `height` and `weight` into `bmi` and Specifying `NA` for Implausible Values

The last three variables we need are `height` and `weight`, and calculated `bmi`. Following the approach used in the Data Management materials posted as part of the Project Instructions after the Proposal, we will calculate `bmi` (body-mass index) from the available `height` (inches) and `weight` (pounds) data. Again, the BMI formula for inches and pounds is available at http://www.bmi-calculator.net/bmi-formula.php. A reasonable range for BMI values is probably about 15 to 50.

```
sur15_m$bmi <- 703 * sur15_m$weight / sur15_m$height^2
Hmisc::describe(sur15_m$bmi)
```

```
sur15_m$bmi
       n  missing distinct     Info     Mean      Gmd      .05      .10
      53        0       51        1     26.2    11.64    18.83    19.26
     .25      .50      .75      .90      .95
   21.16    23.17    25.54    29.54    32.05

lowest :   0.1589689   2.4633141  18.6357396  18.9521290  19.0041090
highest:  30.2705347  31.0081916  33.6108876  41.0810811 188.8302368
```

Those two smallest calculated `bmi` values seem impossibly low, and the highest `bmi` seems impossibly high. Let's look at the heights and weights involved. A reasonable guess is that no one in the class was less than 4 feet tall (48 inches) nor were they greater than 7 feet tall (84 inches), and that no one was outside 80 - 400 pounds.

```
sur15_m %>%
  select(height, weight) %>%
  Hmisc::describe()
```

```
.

 2  Variables      53  Observations
---------------------------------------------------------------------------
height
       n  missing distinct     Info     Mean      Gmd      .05      .10
      53        0       25    0.996    69.45    11.04    61.77    62.10
     .25      .50      .75      .90      .95
   65.00    68.00    70.00    72.00    73.40

lowest :  22.8300  61.0000  61.4173  62.0000  62.5000
highest:  71.0000  72.0000  73.0000  74.0000 217.0000
---------------------------------------------------------------------------
weight
       n  missing distinct     Info     Mean      Gmd      .05      .10
      53        0       38    0.999    152.8    38.15    112.0    121.8
     .25      .50      .75      .90      .95
   135.0    150.0    170.0    177.7    205.2

lowest :   1 107 112 120 121, highest: 200 202 210 220 320
---------------------------------------------------------------------------
```

The subjects with heights of 22.83 inches and 217 inches seem implausible, and the subject with weight 1 pound is also not reasonable. If we change those values to missing, we'll better describe the believable results. I'll instruct R to change heights less than 48 inches and greater than 84 inches to NA, and also to change the weights less than 80 pounds to NA and those greater than 400 pounds (which we didn't see here) to NA.

```
# make a copy of the original values
sur15_m$height.original <- sur15_m$height
sur15_m$weight.original <- sur15_m$weight
# mark implausible values as NA
sur15_m$height[sur15_m$height < 48] <- NA
sur15_m$height[sur15_m$height > 84] <- NA
sur15_m$weight[sur15_m$weight < 80] <- NA
sur15_m$weight[sur15_m$weight > 400] <- NA
# recalculate BMI
sur15_m$bmi <- 703 * sur15_m$weight / sur15_m$height^2
# verify that this worked out as expected
sur15_m %>%
  select(height, weight, bmi) %>%
  Hmisc::describe()
```

.

```
 3  Variables      53  Observations
--------------------------------------------------------------------------------
height
       n  missing distinct     Info     Mean      Gmd      .05      .10
      51        2       23    0.996    67.47    4.008     62.0     62.5
     .25      .50      .75      .90      .95
    65.0     68.0     70.0     72.0     72.5

lowest : 61.0000 61.4173 62.0000 62.5000 63.0000
highest: 70.7400 71.0000 72.0000 73.0000 74.0000
--------------------------------------------------------------------------------
weight
       n  missing distinct     Info     Mean      Gmd      .05      .10
      52        1       37    0.999    155.7    33.58    116.4    125.1
     .25      .50      .75      .90      .95
   135.0    150.5    170.0    177.8    205.6

lowest : 107 112 120 121 125, highest: 200 202 210 220 320
--------------------------------------------------------------------------------
bmi
       n  missing distinct     Info     Mean      Gmd      .05      .10
      50        3       48        1    23.94    4.211    19.06    19.95
     .25      .50      .75      .90      .95
   21.20    23.17    25.46    28.50    30.68

lowest : 18.63574 18.95213 19.00411 19.12835 19.76427
highest: 29.83410 30.27053 31.00819 33.61089 41.08108
--------------------------------------------------------------------------------
```

So now, we have 2 missing heights, 1 missing weight, and we have calculated BMI results, with 3 missing values.

4.5 Cleaning Up to get to our final data set

Let's build a data set, called `sur15` that contains only the fifteen variables in our code book.

```
sur15 <- select(sur15_m, S.id, r.pre, r.now, sex,
                height, weight, bmi, comfort.431, grades,
                load.431, prior.r, english, medium.3c, fiction.4c, seat)
summary(sur15)
```

```
      S.id         r.pre           r.now          sex         height
 Min.   :501   Min.   : 0.00   Min.   : 0.00   Female:25   Min.   :61.00
 1st Qu.:514   1st Qu.: 0.00   1st Qu.:40.00   Male  :28   1st Qu.:65.00
 Median :527   Median : 0.00   Median :60.00               Median :68.00
 Mean   :527   Mean   :21.13   Mean   :56.58               Mean   :67.47
 3rd Qu.:540   3rd Qu.:50.00   3rd Qu.:76.00               3rd Qu.:70.00
 Max.   :553   Max.   :90.00   Max.   :95.00               Max.   :74.00
                                                           NA's   :2
     weight           bmi         comfort.431            grades
 Min.   :107.0   Min.   :18.64   Min.   : 15.00   Individual:40
 1st Qu.:135.0   1st Qu.:21.20   1st Qu.: 70.00   Partner   : 6
 Median :150.5   Median :23.17   Median : 80.00   Group     : 6
 Mean   :155.7   Mean   :23.94   Mean   : 75.94   NA's      : 1
 3rd Qu.:170.0   3rd Qu.:25.46   3rd Qu.: 90.00
 Max.   :320.0   Max.   :41.08   Max.   :100.00
 NA's   :1       NA's   :3
    load.431      prior.r  english   medium.3c          fiction.4c
 Min.   :  0.00   Yes:25   Yes:35   Movies:17   Comedy       :18
 1st Qu.:  5.00   No :28   No :18   TV    :22   Drama        :15
 Median : 40.00                     Other :14   Fantasy/SciFi:14
 Mean   : 38.23                                 Other        : 6
 3rd Qu.: 70.00
 Max.   :100.00

     seat
 Left  :20
 Middle:14
 Right :19
```

4.5.1 Identifying and Dealing with Missing Values

```
md.pattern(sur15)
```

```
   S.id r.pre r.now sex comfort.431 load.431 prior.r english medium.3c
49    1     1     1   1           1        1       1       1         1
 1    1     1     1   1           1        1       1       1         1
 2    1     1     1   1           1        1       1       1         1
 1    1     1     1   1           1        1       1       1         1
      0     0     0   0           0        0       0       0         0
   fiction.4c seat weight grades height bmi
49          1    1      1      1      1   1 0
 1          1    1      1      0      1   1 1
 2          1    1      1      1      0   0 2
 1          1    1      0      1      1   0 2
```

```
0    0      1      1      2   3 7
```

We have 49 subjects with no missing values, 1 who is missing `grades`, another 2 who are missing `height` and `bmi` and 1 who is missing `weight` and `bmi`. So we'll have to keep that in mind when we do work with `bmi` or `grades` in the analyses that follow.

5 The Codebook

The 15 variables in our tidy data set for this demonstration are as follows. The Type column indicates the number of levels in each categorical (factor) variable. Recall that we have missing data in `height`, `weight`, `bmi` and `grades`

Variable	Type	Description / Levels
`S.id`	Cat-53	subject code (501-533)
`r.pre`	Quant	0 (SD) - 100 (SA) with Prior to taking EPBI 431, I was totally confident and comfortable with using R.
`r.now`	Quant	0 (SD) - 100 (SA) with Right now, I am totally confident and comfortable with using R.
`sex`	Cat-2	female, male
`height`	Quant	What is your height, in inches [**2 NA**]
`weight`	Quant	What is your weight, in pounds [**1 NA**]
`bmi`	Quant	703 x `weight`/(`height` squared) [**3 NA**]
`comfort.431`	Quant	0 (SD) - 100 (SA) with I am very comfortable with my understanding of the material discussed so far in EPBI 431.
`grades`	Cat-3	Individual, Partner, Group: In your graduate and undergraduate educational experience, which of the following types of assignments have you received the HIGHEST grades for? [**1 NA**]
`load.431`	Quant	0 (SD) - 100 (SA) with So far, EPBI 431 has required me to do more work than a course has ever required of me.
`prior.r`	Cat-2	yes, no: Before taking 431, had you ever used R before?
`english`	Cat-2	yes, no: Is English the language you speak better than any other?
`medium.3c`	Cat-3	Movies, TV, Other: Which medium do you use most to get your fictional stories (containing plot)?
`fiction.4c`	Cat-4	Comedy, Drama, Fantasy/SciFi, Other: Which type of fictional stories do you consume most?
`seat`	Cat-3	Left, Middle, Right: In EPBI 431, do you USUALLY sit on the left side, in the middle or on the right side (closest to the podium)?

6 Analysis 1a: Compare 2 Population Means using Paired Samples

We'll compare the `r.now` scores to `r.pre` scores. The scores are paired by subject, as each subject gives us both a `r.pre` and `r.now` score, and computing and assessing within-subject differences in comfort with R makes sense, because we are interested in the change in each person's comfort level. We'll generally use `r.now - r.pre` in our calculations, so that positive numbers indicate improvements in confidence. **Note that we'll use a 90% confidence level throughout this demonstration project for all analyses, and I encourage you to do this in your actual Project Study 1 work, as well.**

6.1 Compute and summarize the paired differences

The natural first step is to compute paired differences between the `r.now` and `r.pre` samples, and then use graphical and numerical summaries to assess whether the sample (of differences) can be assumed to follow a Normal distribution. First, we'll calculate the paired differences.

```
sur15$diffs <- sur15$r.now - sur15$r.pre
Hmisc::describe(sur15$diffs)
```

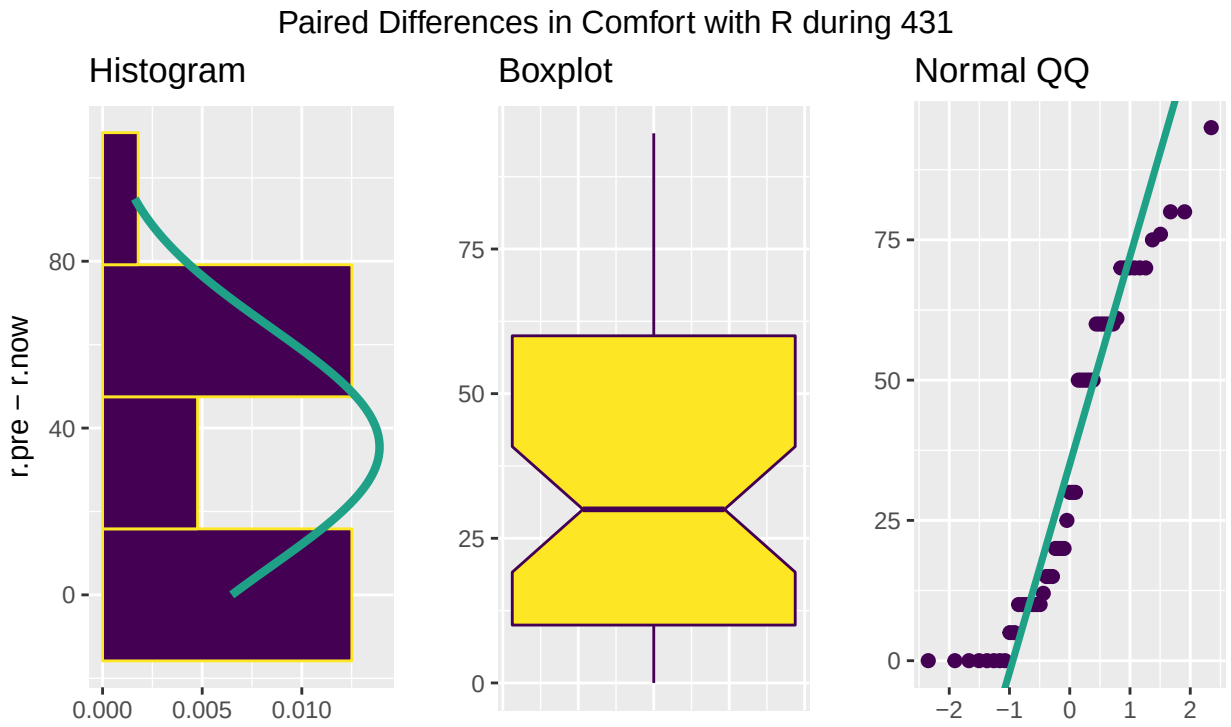
```
sur15$diffs
       n  missing distinct     Info     Mean      Gmd      .05      .10
      53        0       16    0.989    35.45    32.71      0.0      0.0
     .25      .50      .75      .90      .95
    10.0     30.0     60.0     70.0     77.6

Value          0     5    10    12    15    20    25    30    50    60
Frequency      8     2     7     1     3     4     1     3     6     6
Proportion 0.151 0.038 0.132 0.019 0.057 0.075 0.019 0.057 0.113 0.113

Value         61    70    75    76    80    95
Frequency      1     6     1     1     2     1
Proportion 0.019 0.113 0.019 0.019 0.038 0.019
```

OK. It appears that we have successfully subtracted the PRE data from the NOW data, and everyone has a difference of at least zero. But we have a lot of people (8) who have a value of 0. Now, we'll assess whether or not a Normal distribution might be a reasonable model for the data.

6.1.1 Graphical Summaries to Assess Normality

We should start by looking at the Normality of these 53 samples. I'll use the `eda.1sam` tool from the `Love-boost.R` script to view the paired differences. As we've seen, there's a floor effect at zero.

```
eda.1sam(dataframe = sur15, variable = sur15$diffs,
         x.title = "r.pre - r.now",
         ov.title = "Paired Differences in Comfort with R during 431")
```



6.1.2 Numerical Summaries to Assess Normality

In addition to running the usual summary statistics, we could also calculate skew$_1$, to help assess the potential for serious asymmetry, and we could assess whether the Empirical Rule holds well for these differences.

```
mosaic::favstats(sur15$diffs)
```

```
 min Q1 median Q3 max     mean       sd  n missing
   0 10     30 60  95 35.45283 28.69774 53       0
```

```
skew1(sur15$diffs)
```

```
[1] 0.1900091
```

```
Emp_Rule(sur15$diffs)
```

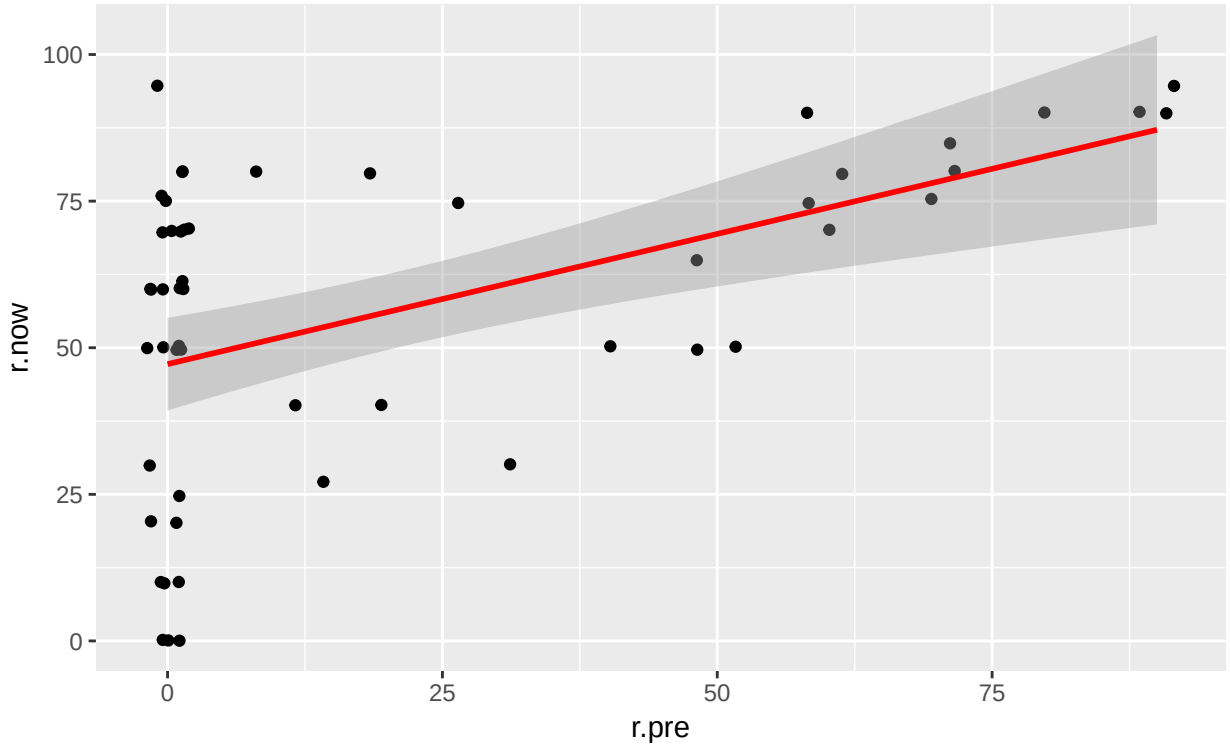
	count	proportion
Mean +/- 1 SD	32	0.6038
Mean +/- 2 SD	52	0.9811
Mean +/- 3 SD	53	1
Entire Data Set	53	1

With just 53 observations, it will be a little difficult to get a clear picture of whether a Normal approximation is reasonable or not. I would likely conclude that a bootstrap approach would be a better choice here than a Normal model for the paired differences, owing to the floor effect (many zeros) in the paired differences. The data are a bit skewed, although they don't quite sneak over the 0.2 cutoff for skew$_1$, and the Empirical Rule is a bit off expectations if the differences truly were sampled from a Normal distribution.

6.2 Did Pairing Help Reduce Nuisance Variation?

We would expect a strong correlation between the `r.pre` and `r.now` scores in this repeated measures analysis where each subject is assessing both their confidence before the class and then again during the class. To assess whether pairing helped reduce nuisance variation, I'll build a scatterplot of the `r.pre` and `r.now` scores, supplemented by a Pearson correlation coefficient. Since we have so many ties in the data, with two or more points in the same place, I'll use `geom_jitter` rather than `geom_point` to plot the points. The larger the correlation, the more that pairing will help reduce the impact of differences between subjects on the `r.pre` score on the comparison we're trying to make.

```
ggplot(sur15, aes(x = r.pre, y = r.now)) +
  geom_jitter() +
  geom_smooth(method = "lm", col = "red")
```




```
cor(sur15$r.pre, sur15$r.now)
```

```
[1] 0.5025712
```

For people with a `r.pre` score greater than zero, we see a pretty strong linear relationship between `r.pre` and `r.now`. The Pearson correlation is quite strong at 0.503 so that a linear model using the `r.pre` score accounts for a reasonably large fraction (25.3%) of the variation in `r.now` scores.

- If the Pearson correlation was small (perhaps less than 0.2), we might conclude that pairing wouldn't be exceptionally helpful, but if the samples are meant to be paired, we should still do a paired samples analysis, but such a small correlation would imply that an independent samples comparison would come to about the same conclusion.

6.3 Building Confidence Intervals

As you'll recall, we have three primary methods for building confidence intervals in a paired samples analysis:

- The Paired t test
- The Wilcoxon Signed Rank test
- The Bootstrap, using `smean.cl.boot`

Let's run each of the three here just so you have the code, even though, as mentioned, I'd be most interested in what the bootstrap approach suggests, owing to the modest non-Normality we see in the sample of differences. In each case, we'll build a 90% confidence interval for the population mean (or pseudo-median, in the case of the Signed Rank test) of the `r.now - r.pre` differences.

6.3.1 The Paired t test approach

Here is a 90% confidence interval for the population mean of the paired `r.now - r.pre` differences.

```
t.test(sur15$diffs, conf.level = .90)
```

```

    One Sample t-test

data:  sur15$diffs
t = 8.9938, df = 52, p-value = 3.549e-12
alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 28.85131 42.05435
sample estimates:
mean of x
 35.45283
```

- The point estimate for the population mean of the differences is 35.45, indicating that the average subject rated agreement with the statement about confidence in R 35 points higher now than when they started the class.
- Our 90% confidence interval for the population mean of the differences is (28.9, 42.1)
- Here, I've assumed a two-sided confidence interval and testing procedure[1]. We conclude, either from the confidence interval (which does not contain zero) or the p value (which is 3.5 x 10^{-12}) that there is a statistically significant difference between the `r.pre` and `r.now` scores.
- The assumptions of the paired t test are
  - that the matched differences are independent of each other,
  - that the matched differences represent a random sample of the population of possible matched differences,
  - and that the matched differences are drawn from a Normally distributed population.
  - The last of these assumptions is hard to justify given these data.

6.3.2 The Wilcoxon signed rank test approach

Here is a 90% confidence interval for the population pseudo-median of the paired `r.now - r.pre` differences, as estimated by the Wilcoxon signed rank approach.

```
wilcox.test(sur15$diffs, conf.level = .90, conf.int = TRUE, exact = FALSE)
```

[1] In this case, a one-sided test might also have been a good choice, since we don't anticipate people will actually admit to being less confident about R after taking the course.

```
    Wilcoxon signed rank test with continuity correction

data:  sur15$diffs
V = 1035, p-value = 5.093e-09
alternative hypothesis: true location is not equal to 0
90 percent confidence interval:
 35.00005 47.50001
sample estimates:
(pseudo)median
      40.00005
```

- The point estimate for the population pseudo-median of the differences is 40, indicating that the average subject rated agreement with the statement about confidence in R 40 points higher now than when they started the class. Note that this is meaningfully different from the sample median difference, which was 30, and that's because there was some skew in the sample data. The interpretation of the Wilcoxon approach is easiest for data that are light-tailed or heavy-tailed, but still generally symmetric.
- Our 90% confidence interval for the population pseudo-median of the differences is (35, 47.5)
- Here, I've assumed a two-sided confidence interval and testing procedure. We conclude, either from the confidence interval (which does not contain zero) or the p value (which is 5.1 x 10^{-9}) that there is a statistically significant difference between `r.pre` and `r.now` scores.
- The assumptions of the Wilcoxon signed rank procedure are
  - that the matched differences are independent of each other,
  - that the matched differences represent a random sample of the population of possible matched differences,
  - and that the matched differences are drawn from a population that is symmetric, but potentially light-tailed, or even outlier-prone
  - The last of these assumptions is hard to justify given these data.

6.3.3 The Bootstrap approach for the mean from paired samples

Here is a 90% confidence interval for the population mean of the paired `r.now - r.pre` differences, as estimated by a bootstrap approach using a random seed of `431`. (*Note*: when you set a seed for this or other analyses in the project, pick something other than `431`.)

```
set.seed(431)
Hmisc::smean.cl.boot(sur15$diffs, conf.int = 0.90)
```

```
    Mean    Lower    Upper
35.45283 29.07453 41.98774
```

- The point estimate for the population mean of the differences is 35.45, indicating that the average subject rated agreement with the statement about confidence in R 35 points higher now than when they started the class.
- Our 90% confidence interval for the population mean of the differences is (29.1, 42.0), which is fairly close to what we got from the paired t test, as it turns out.
- Here, I've assumed a two-sided confidence interval and testing procedure. We conclude, at the 10% significance level, that there is a statistically significant difference between the `r.pre` and `r.now` scores.
- The assumptions of this bootstrap procedure are
  - that the matched differences are independent of each other, and
  - that the matched differences represent a random sample of the population of possible matched differences,

My conclusion would be to use the bootstrap in this case, as it is most justified by my assessment of Normality, and my point estimate and 90% confidence interval for the improvement in R comfort levels is 35.45 (90% CI: 29.1, 42.0) on the 0-100 scale.

7 Analysis 1b: Compare 2 Population Means using Independent Samples

We'll compare `bmi` by `sex` in this analysis using independent samples. We're comparing the mean `bmi` of the population represented by the male respondents to the mean `bmi` of the population represented by the female respondents. There is nothing to suggest that the two samples (male `bmi` and female `bmi` values) are paired or matched in any way. There is no link between, for example, the first male subject's `bmi` and any particular female subject's `bmi`. Plus, as we'll see, there are different numbers of male and female subjects, so there's no way their `bmi` values could be paired. As a result, we're going to be interested in looking at the two samples separately (males and females) to help us understand issues related to hypothesis testing assumptions. **Note that we'll use a 90% confidence level throughout this demonstration project for all analyses, and I encourage you to do this in your actual Project Study 1 work, as well.**

7.1 Summarizing the Distributions for each of the two samples

I'll start by looking at the range of the `bmi` data within each sex.

```
by(sur15$bmi, sur15$sex, Hmisc::describe)
```

```
sur15$sex: Female
dd[x, ]
       n  missing distinct     Info     Mean      Gmd      .05      .10
      24        1       24        1     22.9    3.971    18.96    19.23
     .25      .50      .75      .90      .95
   20.71    21.72    23.69    27.78    29.98

lowest : 18.63574 18.95213 19.00411 19.76427 19.96686
highest: 25.60354 26.44988 28.34677 30.27053 33.61089
---------------------------------------------------------
sur15$sex: Male
dd[x, ]
       n  missing distinct     Info     Mean      Gmd      .05      .10
      26        2       24    0.999     24.9    4.163    21.19    21.49
     .25      .50      .75      .90      .95
   22.21    23.69    26.11    28.96    30.71

lowest : 19.12835 21.15509 21.28460 21.69753 21.97982
highest: 27.89129 28.08127 29.83410 31.00819 41.08108
```

You'll note that there's been a change in the `describe` function in `Hmisc` (with a new release of R), and it now shows `Gmd` which is an abbreviation for Gini's mean difference. It's a robust measure of spread (like the IQR) that shows the mean absolute difference between any pairs of observations.

As we have previously seen, we have three missing BMI values. We could either impute these values, or remove those cases for this analysis. In this case, I'll remove the three missing values, and create a new data set called sur15_1b that contains only the variables I will use in this Analysis, and only the cases where `bmi` is available.

7.1.1 Building a New Data Set, Dropping the subjects with missing `bmi`

```
sur15_1b <- sur15 %>%
  filter(is.na(bmi) != TRUE) %>%
  select(S.id, sex, bmi)
sur15_1b
```

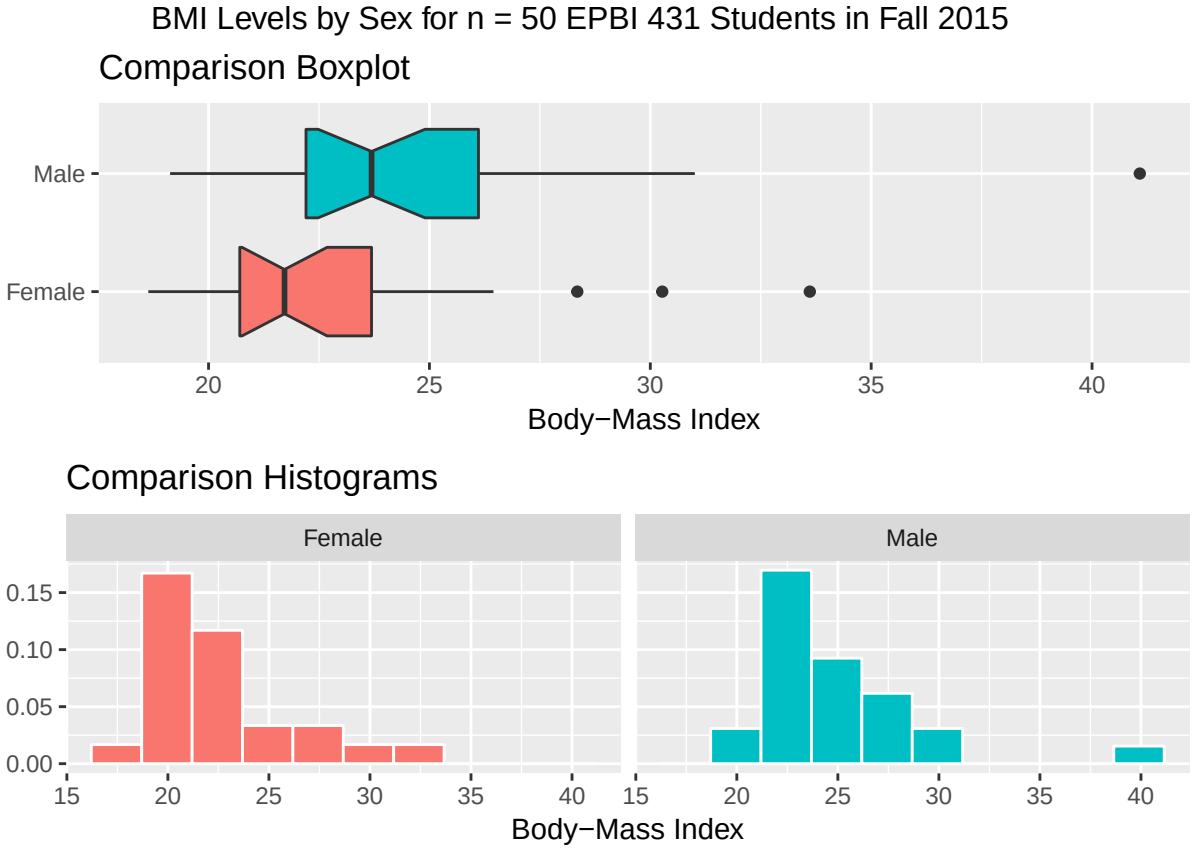
```
# A tibble: 50 x 3
    S.id    sex      bmi
   <int> <fctr>    <dbl>
 1   501   Male 24.55705
 2   502   Male 23.64736
 3   503 Female 22.85594
 4   505   Male 25.53755
 5   506   Male 19.12835
 6   507   Male 41.08108
 7   508   Male 23.67245
 8   509 Female 23.17017
 9   510 Female 22.88700
10   511 Female 33.61089
# ... with 40 more rows
```

Next, we'll use graphical and numerical summaries to assess whether the samples (of males, and of females, separately) can *each* be modeled appropriately by a Normal distribution.

7.1.2 Graphical Summaries

Since we are exploring the distributions of two independent samples, I'll use the `eda.ksam` function from the `Love-boost.R` script.

```
eda.ksam(outcome = sur15_1b$bmi, group = sur15_1b$sex,
         axis.title = "Body-Mass Index",
         main.title = "BMI Levels by Sex for n = 50 EPBI 431 Students in Fall 2015")
```



I see a few candidate outliers in the female data on the high end, which suggest some meaningful skew, and one high candidate outlier and some sign of right skew also among the male subjects. There's room for concern about whether a test that requires Normal distributions in the populations is a good choice here. With these small sample sizes, we'd probably be better off not making too many strong assumptions.

7.1.3 Numerical Summaries

```
pander(by(sur15_1b$bmi, sur15_1b$sex, mosaic::favstats))
```

- **Female**:

min	Q1	median	Q3	max	mean	sd	n	missing
18.64	20.71	21.72	23.69	33.61	22.9	3.726	24	0

- **Male**:

min	Q1	median	Q3	max	mean	sd	n	missing
19.13	22.21	23.69	26.11	41.08	24.9	4.286	26	0

We have 24 female and 26 male subjects with known BMI values. The skew_1 values can be calculated from these summary statistics, or we can ask for them directly with...

```
by(sur15_1b$bmi, sur15_1b$sex, skew1)
```

```
sur15_1b$sex: Female
[1] 0.3179627
------------------------------------------------------------
sur15_1b$sex: Male
[1] 0.2829772
```

It looks like the right skew is large enough in each group to warrant avoiding tests that require Normality. We could also check the Empirical Rule, with...

```
pander(by(sur15_1b$bmi, sur15_1b$sex, Emp_Rule))
```

	count	proportion
Mean +/- 1 SD	18	0.75
Mean +/- 2 SD	23	0.9583
Mean +/- 3 SD	24	1
Entire Data Set	24	1

	count	proportion
Mean +/- 1 SD	22	0.8462
Mean +/- 2 SD	25	0.9615
Mean +/- 3 SD	25	0.9615
Entire Data Set	26	1

- **Female**:
- **Male**:

In any case, it's not reasonable to assume Normality here.

7.2 Building Confidence Intervals

As you'll recall, we have four available methods for building confidence intervals in an independent samples analysis:

- Welch's t test (t test without assuming equal variances)
- The Pooled t test (t test with equal variances assumed)
- The Wilcoxon-Mann-Whitney Rank Sum Test
- The Bootstrap, using `bootdif`

Let's run each of the four here just so you have the code, even though, as mentioned, I'd be most interested in what the bootstrap approach or the rank sum test suggests, owing to the fact that the samples aren't well described by Normal models. In each case, we'll build a 90% confidence interval for the population mean (or another measure of central tendency, in the case of the Rank Sum test) comparing `bmi` for females and males.

7.2.1 The Welch's t test approach

With a nearly balanced design (24 females and 26 males), it is unlikely that the assumption of equal population variances will make much of a difference here, so we might expect the Welch t test and pooled t test to look similar. Neither is a great choice here, due to the samples showing some non-Normality. Regardless, here is a 90% confidence interval for the difference between the female and male population mean `bmi` based on Welch's test.

```
t.test(bmi ~ sex, data = sur15_1b, conf.level = 0.90)
```

```
bmi ~ sex


    Welch Two Sample t-test

data:  bmi by sex
t = -1.7631, df = 47.839, p-value = 0.08428
alternative hypothesis: true difference in means is not equal to 0
90 percent confidence interval:
 -3.90019474 -0.09718156
sample estimates:
mean in group Female   mean in group Male
            22.90420             24.90289
```

- The point estimates for the two population `bmi` means are 22.9 for females and 24.9 for males, so the average male has a BMI estimated to be about 2.0 points higher than the average for females, based on our samples.
- Our 90% confidence interval for the difference (Male - Female) of the population means is (0.1, 3.9).
- Here, I've assumed a two-sided confidence interval and testing procedure. We conclude, either from the confidence interval (which does not quite contain zero) or the p value (which is 0.084) that there is a statistically significant difference between the true means of the male and female `bmi` levels.
- The assumptions of the Welch's t test are
  - that the samples in each group are drawn independently of each other,
  - that the samples in each group represent a random sample of the population of interest,
  - and that the samples in each group are drawn from a Normally distributed population.
  - The last of these assumptions is hard to justify given these data.

7.2.2 The Pooled t test (t test with equal variances)

The pooled t test, of course, actually adds an assumption (that either the sample sizes or the population variances are equal) to the assumptions of the Welch test. With a nearly balanced design (24 females and 26 males), it is unlikely that the assumption of equal population variances will make much of a difference here, so we might expect the Welch t test and pooled t test to look similar. Neither is a great choice here, due to the samples showing some non-Normality. Regardless, here is a 90% confidence interval for the difference between the female and male population mean `bmi` based on the pooled t test.

```
t.test(bmi ~ sex, data = sur15_1b, conf.level = .90, var.equal = TRUE)
```

```
bmi ~ sex


    Two Sample t-test

data:  bmi by sex
t = -1.7531, df = 48, p-value = 0.08597
alternative hypothesis: true difference in means is not equal to 0
90 percent confidence interval:
 -3.9108918 -0.0864845
sample estimates:
mean in group Female   mean in group Male
            22.90420             24.90289
```

- The point estimates for the two population `bmi` means are still 22.9 for females and 24.9 for males, so the average male has a BMI estimated to be about 2.0 points higher than the average for females, based on our samples.
- Our 90% confidence interval for the difference (Male - Female) of the population means is again (0.1, 3.9).
- Here, I've assumed a two-sided confidence interval and testing procedure. We conclude, either from the confidence interval (which does not quite contain zero) or the p value (which is 0.086) that there is a statistically significant difference between the true means of the male and female `bmi` levels.
- The assumptions of the pooled t test are
  - that the samples in each group are drawn independently of each other,
  - that the samples in each group represent a random sample of the population of interest,
  - the samples in each group are drawn from a Normally distributed population,
  - *and* that either the sample sizes or the population variances are equal.
  - The Normality assumption remains hard to justify given these data, so we should look at alternatives.

7.2.3 The Wilcoxon-Mann-Whitney rank sum test

The first test we'll look that doesn't require Normality is the Wilcoxon-Mann-Whitney rank sum test. The main problem with this approach is that it doesn't estimate the difference in population means, but rather it estimates a location shift for the distribution as a whole. Here is a 90% confidence interval for the difference between the female and male population `bmi` distributions based on the rank sum approach.

```
wilcox.test(sur15_1b$bmi ~ sur15_1b$sex, conf.level = .90, conf.int = TRUE, exact = FALSE)
```

```
    Wilcoxon rank sum test with continuity correction

data:  sur15_1b$bmi by sur15_1b$sex
W = 187, p-value = 0.01562
alternative hypothesis: true location shift is not equal to 0
```

```
90 percent confidence interval:
 -3.2578566 -0.6348994
sample estimates:
difference in location
             -2.149204
```

- The estimated location shift in population `bmi` across the two sexes is 2.15.
- Our 90% confidence interval for the location shift (Male - Female) of the populations is (0.6, 3.3).
- Here, I've assumed a two-sided confidence interval and testing procedure. We conclude, either from the confidence interval (which does not contain zero) or the p value (which is 0.016) that there is a statistically significant difference between the true locations of the male and female `bmi` levels.
- The assumptions of the rank sum test are
  - that the samples in each group are drawn independently of each other,
  - *and* that the samples in each group represent a random sample of the population of interest,

So the Wilcoxon test is a stronger result than either t test, because it doesn't have any serious assumption violations. It is also in this case a more statistically significant result because the confidence interval isn't so close to zero (and thus the p value is smaller.)

7.2.4 The Bootstrap for comparing means from two independent samples

The other approach we have for independent samples comparisons that doesn't require Normality is the bootstrap, and specifically, the `bootdif` function. This approach returns to estimating the difference in population means, but gives a different answer depending on the choice of random number seed. Here is a 90% confidence interval for the difference between the female and male population `bmi` distributions based on the bootstrap using a seed of `431`. (*Note*: when you set a seed for this or other analyses in the project, pick something other than `431`.)

```
set.seed(431); bootdif(sur15_1b$bmi, sur15_1b$sex, conf.level = 0.90)
```

```
Mean Difference            0.05            0.95
      1.9986881       0.2315363       3.9152286
```

- The population mean BMI in Males is estimated to be about 2.0 points higher than the population mean BMI for Females, based on our samples. So the mean differences' point estimate is 2.0
- Our 90% confidence interval for the difference (Male - Female) of the population means is (0.2, 3.9).
- Here, I've assumed a two-sided confidence interval and testing procedure. We conclude, either from the confidence interval (which does not contain zero) that there is a statistically significant difference (at the 10% significance level, since we have a 90% confidence interval) between the true means of the male and female `bmi` levels.
- The assumptions of this bootstrap procedure are:
  - that the samples in each group are drawn independently of each other,
  - *and* that the samples in each group represent a random sample of the population of interest,

So, I think either the bootstrap or rank sum procedure would be appropriate here, due to the non-Normality in the samples. In either case, at the 10% significance level, there is a statistically significant difference between the population mean (or in the rank sum case, location of) BMI for males and the population mean (or location of) BMI for females, based on our sample of 50 respondents.

8 Analysis 2: Comparing 3+ Population Means via ANOVA

We'll compare `comfort.431` by `grades` in this analysis, using the analysis of variance, and related tools. We're comparing the mean `comfort.431` scores of the population represented by the respondents who got their best grades on individual work, to the population represnted by the respondents who got their best grades with a partner, to the population represented by the respondents who got their best grades on group work. There is no link between subjects across the three `grades` groups, so the samples are independent. Plus, as we'll see, there are different numbers of subjects in the three `grades` groups, so there's no way their `comfort.431` values could be matched. As a result, we're going to be interested in looking at the three samples separately to help us understand issues related to hypothesis testing assumptions. **Note that we'll use a 90% confidence level throughout this demonstration project for all analyses, and I encourage you to do this in your actual Project Study 1 work, as well.**

8.1 Summarizing the Distributions for each of the three samples

I'll start by looking at the range of the `comfort.431` data within each `grades` group.

```
by(sur15$comfort.431, sur15$grades, Hmisc::describe)
```

```
sur15$grades: Individual
dd[x, ]
       n  missing distinct     Info     Mean      Gmd      .05      .10
      40        0       13     0.98    78.95    16.55    50.00    50.00
     .25      .50      .75      .90      .95
   73.75    80.00    90.00    95.00    95.25

Value         35    50    60    70    75    78    80    85    88    90
Frequency      1     4     1     4     4     1     7     2     1     9
Proportion 0.025 0.100 0.025 0.100 0.100 0.025 0.175 0.050 0.025 0.225

Value         92    95   100
Frequency      1     3     2
Proportion 0.025 0.075 0.050
------------------------------------------------------------
sur15$grades: Partner
dd[x, ]
       n  missing distinct     Info     Mean      Gmd
       6        0        5    0.971    53.33    32.67

Value         15    30    50    70    85
Frequency      1     1     1     2     1
Proportion 0.167 0.167 0.167 0.333 0.167
------------------------------------------------------------
sur15$grades: Group
dd[x, ]
       n  missing distinct     Info     Mean      Gmd
       6        0        5    0.971     79.5     18.2

Value         50    73    85    89    95
Frequency      1     1     2     1     1
Proportion 0.167 0.167 0.333 0.167 0.167
```

We have only 6 respondents in each of the Partner and Group `grades` categories, so that will make it difficult to say much about the distributions of `comfort.431` in those populations.

8.1.1 Dropping the subject with a missing grades value

```
Hmisc::describe(sur15$grades)
```

```
sur15$grades
       n  missing distinct
      52        1        3

Value      Individual    Partner      Group
Frequency          40          6          6
Proportion      0.769      0.115      0.115
```

As you can see, we have one subject with a missing value for the `grades` variable. We'll drop that subject for the remainder of Analysis 2 (and also for Analysis 3 to come). While I'm at it, I'll also select only those variables that we might use in Analyses 2 and 3. That combined effort will yield the new data frame: `sur15_23`, which I will use for the remainder of Analyses 2 and 3.

```
sur15_23 <- sur15 %>%
  filter(is.na(grades) != TRUE) %>%
  select(S.id, grades, comfort.431, load.431)
sur15_23
```

```
# A tibble: 52 x 4
    S.id     grades comfort.431 load.431
   <int>     <fctr>       <int>    <int>
 1   501 Individual          90       95
 2   502      Group          50       10
 3   503    Partner          70        0
 4   504    Partner          50      100
 5   505      Group          85       20
 6   506 Individual          80       10
 7   507 Individual          50       50
 8   508 Individual          80       10
 9   509 Individual          75       60
10   510 Individual          50       50
# ... with 42 more rows
```

8.1.2 Graphical Summaries

Since we are exploring the distributions of three independent samples, I'll continue to use the `eda.ksam` function from the `Love-boost.R` script, since that function can take any number of independent samples greater than one.

```
eda.ksam(outcome = sur15_23$comfort.431, group = sur15_23$grades,
         axis.title = "Comfort with 431 Materials (0-100)",
         main.title = "Comfort with 431 by Type of Assignment that produces best grades",
         notch = FALSE)
```

Notice that the boxplot notches would have been messy (they extend outside the levels of the boxes) in this case due to the small numbers of subjects in the Partner and Group `grades` groups. So I used the `notch = FALSE` feature in the `eda.ksam` function. The sample sizes are so small that the histograms for those two levels of the `grades` factor tell us nothing of substantial value. In addition, the Individual data look as though they may be either skewed to the left a bit or at least have one potential outlier. With these tiny sample sizes (less than 10 observations) these plots don't really help much. All of the values in each group are within the stated response levels (0-100) but otherwise, there's not a lot to go on. ANOVA is quite robust, so we'll run it, but I expect that a Kruskal-Wallis approach may also be useful here.

8.1.3 Numerical Summaries

With just six observations in the Partner and Group `grades` levels, there's not much to see in numerical summaries, either.

```
pander(by(sur15_23$comfort.431, sur15_23$grades, mosaic::favstats))
```

- **Individual**:

min	Q1	median	Q3	max	mean	sd	n	missing
35	73.75	80	90	100	78.95	15.29	40	0

- **Partner**:

min	Q1	median	Q3	max	mean	sd	n	missing
15	35	60	70	85	53.33	26.77	6	0

- **Group**:

min	Q1	median	Q3	max	mean	sd	n	missing
50	76	85	88	95	79.5	16.15	6	0

We have 40 Individual, 6 Partner and 6 Group subjects with known comfort levels. The skew$_1$ values will not be helpful for the two smaller categories, but can be calculated from these summary statistics, or we can ask for them directly with...

```
by(sur15_23$comfort.431, sur15_23$grades, skew1)
```

```
sur15_23$grades: Individual
[1] -0.06867113
------------------------------------------------------------
sur15_23$grades: Partner
[1] -0.2490291
------------------------------------------------------------
sur15_23$grades: Group
[1] -0.3406373
```

The skew appears modest in the Individual group, so perhaps ANOVA would be OK there. We could also check the Empirical Rule (although, again, this is only appropriate with a moderate sample size - here with the Individual data, only), with...

```
pander(by(sur15_23$comfort.431, sur15_23$grades, Emp_Rule))
```

	count	proportion
Mean +/- 1 SD	29	0.725
Mean +/- 2 SD	39	0.975
Mean +/- 3 SD	40	1
Entire Data Set	40	1

	count	proportion
Mean +/- 1 SD	4	0.6667
Mean +/- 2 SD	6	1
Mean +/- 3 SD	6	1
Entire Data Set	6	1

	count	proportion
Mean +/- 1 SD	5	0.8333
Mean +/- 2 SD	6	1
Mean +/- 3 SD	6	1
Entire Data Set	6	1

- **Individual**:
- **Partner**:
- **Group**:

The conclusion I draw from all of this is that we need to run both ANOVA and Kruskal-Wallis approaches, but that we probably can't trust either of them too much, with such small sample sizes in the non-Individual `grades` levels. Anything below 15 patients is just too small, and, practically, I'd consider collapsing the groups to `Individual` vs. `All Other`. But for this demonstration, I'll press on.

8.2 Building Inferences to Compare the Three Populations

As you'll recall, we have at least two available methods for building statistical inferences when comparing more than two independent samples.

- Analysis of Variance
- The Kruskal-Wallis Test

There is also a bootstrap approach[2], but we'll defer discussion of that until 432.

Let's run both methods here just so you have the code, even though we don't have large enough data samples in the `Partner` and `Group` levels to justify statistical inference at all. In each case, we'll build hypothesis tests, and compare the distributions of `comfort.431` across levels of `grades` using a 90% confidence level.

8.2.1 Kruskal-Wallis Test

I'll start with the Kruskal-Wallis test, which at least doesn't require me to assume Normality in the three populations. The null hypothesis here is that there is no location shift in the distributions of comfort in 431 across the three levels of `grades`. Put another way, the location parameters of the distributions are the same across the three `grades` levels. The Kruskal-Wallis test is the extension of the Wilcoxon-Mann-Whitney rank sum test to studies involving more than two independent samples.

```
kruskal.test(sur15_23$comfort.431 ~ sur15_23$grades)
```

```
    Kruskal-Wallis rank sum test

data:  sur15_23$comfort.431 by sur15_23$grades
Kruskal-Wallis chi-squared = 6.8034, df = 2, p-value = 0.03332
```

[2]The best function I am aware of for Bootstrapped ANOVA (and ANCOVA) in R is by Sam Mancuso - see https://sammancuso.com/2015/05/18/bootstrapped-anova-and-ancova-in-r/

- Here, we'd conclude that there is a statistically significant difference (at least at the 10% significance level we're using, since $p = 0.03 < 0.10$) between the `comfort.431` scores for the three `grades` categories.
- The assumptions of the Kruskal-Wallis test are the same as those of the Wilcoxon-Mann-Whitney rank sum test, specifically that
  - that the samples in each category are drawn independently of each other,
  - *and* that the samples in each category represent a random sample of the population of interest,

The main problem here is that the sample size is so small that we can't tell whether this result is truly more or less reasonable than an ANOVA approach. We really need a minimum of 15 observations (and ideally more like 30) in each group to let our histograms and boxplots have any chance to be informative on these points. So let's look at the ANOVA results.

8.2.2 Analysis of Variance

The Analysis of Variance compares the means of `comfort.431` in the three `grades` populations. We can run the analysis using either of two approaches, each of which we'll show in what follows.

```
anova(lm(sur15_23$comfort.431 ~ sur15_23$grades))
```

```
Analysis of Variance Table

Response: sur15_23$comfort.431
                Df  Sum Sq Mean Sq F value   Pr(>F)
sur15_23$grades  2  3504.1 1752.05  6.1301 0.004207 **
Residuals       49 14004.7  285.81
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Here, we'd conclude that there is a statistically significant difference (at least at the 10% significance level we're using, since $p = 0.004 < 0.10$) between the population mean `comfort.431` scores for the three `grades` categories.
- The `grades` account for $\eta^2 = \frac{3504.1}{3504.1+14004.7} = 0.2$ or 20% of the variation in `comfort.431` scores in our sample.
- The natural next question is to try to identify which pairs of `grades` categories are different, and we'll tackle that in a moment with Bonferroni and Tukey HSD approaches.
- ANOVA is the natural extension of the pooled t test for two independent samples, and so it has the same set of assumptions when we compare population means across multiple categories (here, the three `grades` categories)...
  - that the samples in each category are drawn independently of each other,
  - that the samples in each category represent a random sample of the population of interest,
  - the samples in each category are drawn from a Normally distributed population,
  - *and* that either the sample sizes or the population variances are equal across the categories.

The main problem here is that the sample size is so small that we can't tell whether this result is truly reasonable or not. We really need a minimum of 15 observations (and ideally more like 30) in each group to let our histograms and boxplots have any chance to be informative on these points. We'll move on to looking at the pairwise comparisons, though, in this demonstration.

8.2.3 Bonferroni approach to Pairwise Comparisons of Means

We have two approaches available for dealing with multiple comparisons. If we had not pre-planned the full set of pairwise comparisons of `comfort.431` across the `grades` categories, or if we wanted to use the most conservative approach, we could apply a Bonferroni correction to our comparisons. This works reasonably well even with an unbalanced design, such as we have here.

```
pairwise.t.test(sur15_23$comfort.431, sur15_23$grades, p.adjust = "bonferroni")
```

```
    Pairwise comparisons using t tests with pooled SD

data:  sur15_23$comfort.431 and sur15_23$grades

        Individual Partner
Partner 0.0034     -
Group   1.0000     0.0299

P value adjustment method: bonferroni
```

- With an overall significance level of 10%, it appears that we can conclude that there are statistically significant differences between the mean of the Partner category and the means of the other two categories, but there is no statistically significant difference between Individual and Group means.
- The assumptions here include the ANOVA assumptions, which are no more or less justified than they were before. We do not, however, require that our pairwise comparisons be pre-planned.

8.2.4 Tukey's Honestly Significant Differences approach to Pairwise Comparisons of Means

The Tukey HSD approach requires us to use the `aov` approach to specifying the ANOVA model, as opposed to the `anova with lm` approach we took above. The results for `aov` are identical, as you can see below.

```
summary(aov(sur15_23$comfort.431 ~ sur15_23$grades))
```

```
                Df Sum Sq Mean Sq F value  Pr(>F)
sur15_23$grades  2   3504  1752.0    6.13 0.00421 **
Residuals       49  14005   285.8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now, we run the Tukey HSD comparisons, both in a plot and table of results. As specified previously, we'll use a 90% confidence level across the set of comparisons.

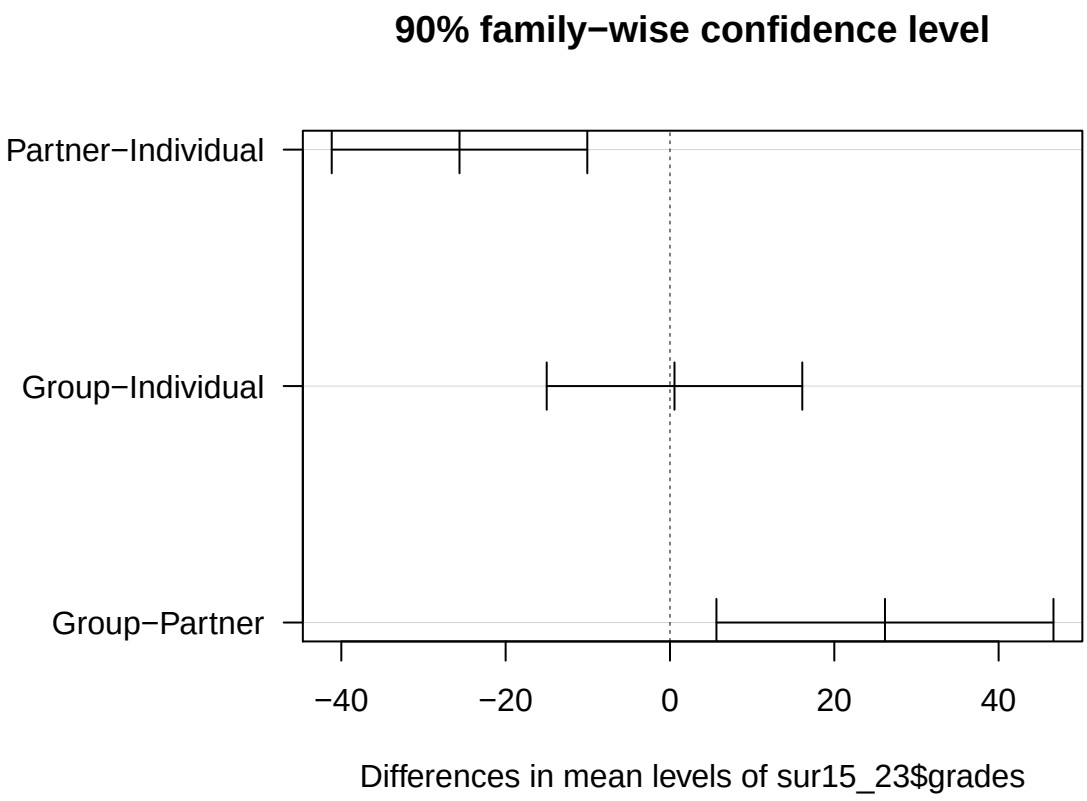
```
TukeyHSD(aov(sur15_23$comfort.431 ~ sur15_23$grades), conf.level = 0.90)
```

```
  Tukey multiple comparisons of means
    90% family-wise confidence level

Fit: aov(formula = sur15_23$comfort.431 ~ sur15_23$grades)

$`sur15_23$grades`
                         diff        lwr       upr     p adj
Partner-Individual -25.61667 -41.169906 -10.06343 0.0031728
Group-Individual     0.55000 -15.003240  16.10324 0.9969604
Group-Partner       26.16667   5.655662  46.67767 0.0265110
```

The confidence intervals suggest that the mean Partner scores are statistically significant different (in fact, lower) than both the mean Individual and Group scores, while the Group and Individual scores are not significantly different.

Note that in the plot below, we see these results a bit more clearly after we adjust the margins of the plot and use the `las = 1` bit at the end of the plotting call to get the x and y axis labels to be horizontal.

```
mar.default <- c(5,6,4,2) + 0.1
par(mar = mar.default + c(0, 4, 0, 0))
plot(TukeyHSD(aov(sur15_23$comfort.431 ~ sur15_23$grades), conf.level = 0.90), las = 1)
```




```
par(mar = mar.default)
```

Our conclusions are:

- that the sample size is just too small in the non-Individual `grades` categories to draw very firm conclusions, but
- despite this, there appears to be evidence of a statistically significant difference in `comfort.431` across the three `grades` categories, according to either an ANOVA or Kruskal-Wallis approach, at the 90% confidence level, and
- specifically, it appears at the 10% significance level that the population means of the Group and Individual comfort levels are comparable and both are higher than the population mean of the Partner comfort levels.

9 Analysis 3: Regression Comparison of Means with Adjustment

In this analysis, we'll again compare `comfort.431` by `grades` but now, after adjusting for `load.431` in a regression model. Here, we've already done the graphical and numerical summaries of primary interest in the context of Analysis 2, so after some checks of range and missingness in the new `load.431` variable within the `sur15_23` data frame, we'll press on to the actual regression model, and then evaluate its assumptions.

```
Hmisc::describe(sur15_23$load.431)
```

```
sur15_23$load.431
       n  missing distinct     Info     Mean      Gmd      .05      .10
      52        0       18    0.988    38.96    36.46     0.00     0.00
     .25      .50      .75      .90      .95
    8.75    42.50    70.00    79.50    80.90

Value          0     2     5    10    20    30    40    45    50    60
Frequency     10     1     2     6     4     2     1     1     3     5
Proportion 0.192 0.019 0.038 0.115 0.077 0.038 0.019 0.019 0.058 0.096

Value         62    65    70    75    80    82    95   100
Frequency      1     2     6     2     3     1     1     1
Proportion 0.019 0.038 0.115 0.038 0.058 0.019 0.019 0.019
```

There are no missing values, and every response is in the pre-specified range (0-100) so we're all set.

9.1 The Regression Model, Adjusting for a Single Quantitative Covariate

```
summary(lm(comfort.431 ~ grades + load.431, data = sur15_23))
```

```

Call:
lm(formula = comfort.431 ~ grades + load.431, data = sur15_23)

Residuals:
    Min      1Q  Median      3Q     Max
-42.299  -4.090   2.507  10.253  34.957

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    82.49309    4.04063  20.416  < 2e-16 ***
gradesPartner -26.38935    7.40409  -3.564 0.000838 ***
gradesGroup    -0.15054    7.39881  -0.020 0.983851
load.431       -0.08658    0.07425  -1.166 0.249373
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.84 on 48 degrees of freedom
Multiple R-squared:  0.2222,    Adjusted R-squared:  0.1735
F-statistic:  4.57 on 3 and 48 DF,  p-value: 0.006787
```

We can certainly go through and interpret the model in some detail, and this will be easier after Part C of the course.

```
anova(lm(comfort.431 ~ grades + load.431, data = sur15_23))
```

```
Analysis of Variance Table

Response: comfort.431
          Df  Sum Sq Mean Sq F value   Pr(>F)
grades     2  3504.1 1752.05  6.1751 0.004107 **
load.431   1   385.7  385.75  1.3596 0.249373
Residuals 48 13619.0  283.73
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- From the ANOVA table, we see a statistically significant `grades` effect at the 10% level (actually the p value [.004] is much below $\alpha = 0.10$) after accounting for `load.431`.
- Combined, the `grades` and `load.431` variables account for 22.22% of the variation in `comfort.431` which is only a modest improvement over the 20% accounted for by `grades` alone in Analysis 2. So the `load.431` variable has only a modest (and, we note, not statistically significant) impact.

9.2 Predicting the outcome at the average level of the covariate for each group

```
mean(sur15_23$load.431)
```

```
[1] 38.96154
```

At the mean level of `load.431`, which turns out to be about 39, we should predict the values of `comfort.431` for subjects in each of the three `grades` categories. Here's how I set up the model and the new subjects we'll predict.

```
# specify the regression model we're using
model3 <- lm(comfort.431 ~ grades + load.431, data = sur15_23)
# specify the new data we'll need to predict for
new3 <- data.frame(grades = c("Individual", "Partner", "Group"),
                   load.431 = rep(mean(sur15_23$load.431), 3)
                   )
new3 # sanity check
```

```
      grades load.431
1 Individual 38.96154
2    Partner 38.96154
3      Group 38.96154
```

We'll use 90% prediction intervals (intervals for predicting individual new subjects) here.

```
# get regression predictions from model for new data
pander(predict(model3, newdata = new3, interval = "prediction", level = 0.90))
```

fit	lwr	upr
79.12	50.52	107.7
52.73	22.2	83.26
78.97	48.44	109.5

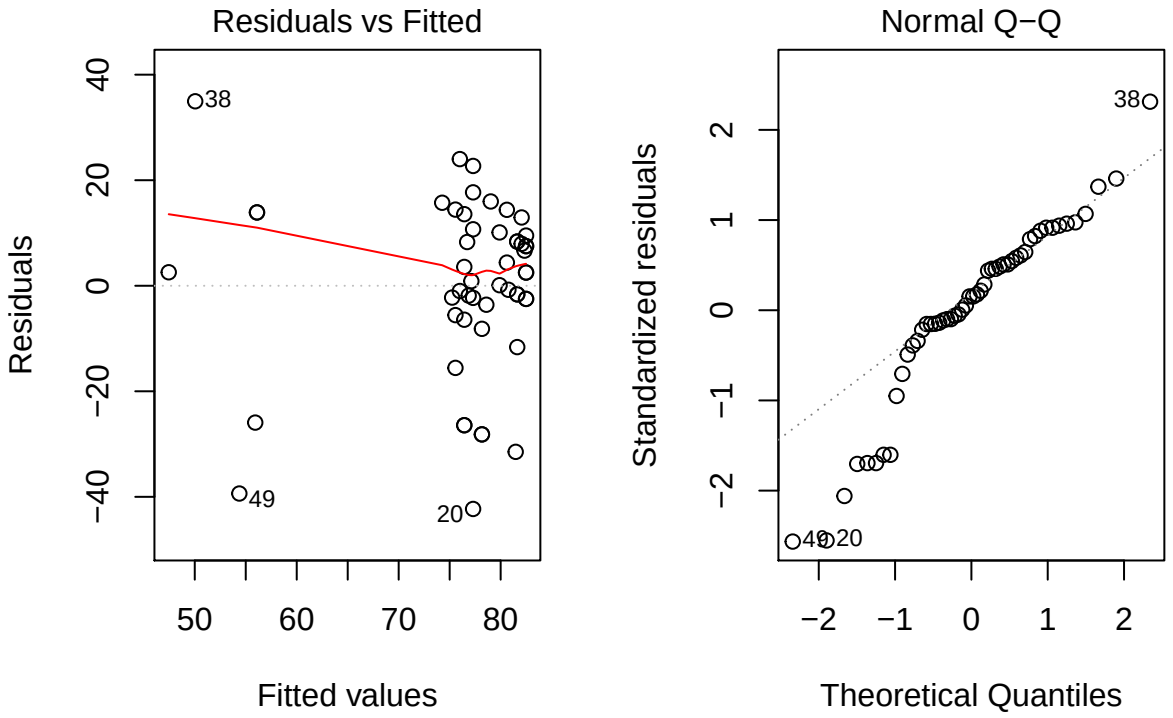
So, we conclude that the model predicts, for example, that a new subject with the mean level of our covariate (`load.431`) who believes they have received their best grade for Individual work will have a `comfort.431` score of 79.1, with a 90% prediction interval of (50.5, 107.7). Since the maximum possible `comfort.431` score is 100, this is a pretty clear indication that our model has some flaws.

9.3 Identifying Assumption Violations

In Part C of the course, we'll focus on several plots for checking regression assumptions, but for now, we'll just look at two of them, and for the project Study 1, I'm just looking for you to identify whether or not there is a problem in each case.

- If the residuals vs. fitted values plot looks like a fuzzy football, with no particular pattern or trend, then we're in good shape.
- If the Normal Q-Q plot of standardized residuals looks like a straight line (so we'd assume a Normal model held for the residuals), then we're in good shape.

```
par(mfrow=c(1,2))
plot(model3, which = 1:2)
```




```
par(mfrow=c(1,1))
```

We certainly have some assumption violations to identify here.

- The plot on the left suggests that we have a much better understanding of what's going on for the subjects in the Individual and Group categories (which have fitted (predicted) values near 80) than we do the subjects in the Partner categories (which have fitted values near 55).
- The plot on the right suggests some problems with assuming Normality for the residuals in this regression model.

For Study 1 in the project, as mentioned, you need only to produce the plots and interpret whether or not a serious concern exists, not propose and test detailed solutions.

10 Analysis 4: Two-Way (2 x 2) Contingency Table

We'll look at the association of `prior.r` with `english` in this analysis. The `prior.r` variable and the `english` variable each have two levels, and suppose we are interested in whether `english` has an impact on `prior.r`, so we'll build a contingency table with `english` in the rows and `prior.r` in the columns. **Note that we'll use a 90% confidence level and the add 2 successes and 2 failures Bayesian augmentation, and I encourage you to do this in your actual Project Study 1 work, as well.**

10.1 Building the 2x2 Table from data

Let's look at the 2x2 table we get. Of course, in our earlier work, we ensured that the `english` levels and the `prior.r` levels would each be posted in the order "Yes" then "No". So the top left corner of our 2x2 table gives the count of subjects for whom English is the language they speak better than any other, and who had used R prior to taking 431.

10.1.1 The Simplest Table

```
table(sur15$english, sur15$prior.r)
```

```
     Yes No
  Yes  17 18
  No    8 10
```

Those names could use some work, I think.

- The row names, in order, should be something like "English" (where "Yes" is now) and "Not English"
- The column names, respectively, should be "Prior R user" and "No Prior R"

10.1.2 Improving the Table

We could also add margins and improve the row and column names for the table, and `pander` it into a more attractive format, as follows.

```
t1 <- table(sur15$english, sur15$prior.r)
colnames(t1) <- c("Prior R user", "No Prior R")
rownames(t1) <- c("English", "Not English")
pander(addmargins(t1))
```

	Prior R user	No Prior R	Sum
English	17	18	35
Not English	8	10	18
Sum	25	28	53

10.1.3 Even Fancier Formatting for the 2x2 Table

And here's an even fancier version of a 2x2 contingency table, where we have eliminated the automatic formatting of the row titles in bold, and used italics for the marginal totals, plus added a caption.

```
t1 <- table(sur15$english, sur15$prior.r)
colnames(t1) <- c("Prior R user", "No Prior R")
```

```
rownames(t1) <- c("English", "Not English")
emphasize.italics.rows(3)
emphasize.italics.cols(3)
panderOptions('table.emphasize.rownames', FALSE)
pander(addmargins(t1),
       caption = "Best Language vs. R use Prior to 431")
```

Table 15: Best Language vs. R use Prior to 431

	Prior R user	No Prior R	Sum
English	17	18	*35*
Not English	8	10	*18*
Sum	*25*	*28*	*53*

10.2 Default 2x2 Analysis: Not Completely Appropriate for the Project

Now, we can call the `twoby2` function from the `Epi` package to do the analysis. Remember that if we have the data in place, and want to use it to specify the table, we use `twoby2`. That's the situation we're in here. If, on the other hand, we had the table of counts, but not the data, so that we wanted to type in the individual cell values as part of our call, then we have to use the script I provided to get the `twobytwo` function.

We could use the default `twoby2` approach shown below, but that's not perfect for our purposes in the project, because:

- it doesn't use a Bayesian augmentation, and
- it uses 95% confidence levels.

```
twoby2(t1) # without Bayesian augmentation, 95% confidence level
```

```
2 by 2 table analysis:
------------------------------------------------------
Outcome   : Prior R user
Comparing : English vs. Not English

            Prior R user No Prior R    P(Prior R user) 95% conf. interval
English               17         18             0.4857    0.3274   0.6470
Not English            8         10             0.4444    0.2400   0.6696

                                 95% conf. interval
            Relative Risk: 1.0929    0.5886   2.0292
        Sample Odds Ratio: 1.1806    0.3769   3.6980
Conditional MLE Odds Ratio: 1.1769    0.3264   4.3469
    Probability difference: 0.0413   -0.2270   0.2956

             Exact P-value: 1
        Asymptotic P-value: 0.7757
------------------------------------------------------
```

Next, we'll develop the 2x2 analysis I want to see in your project.

10.3 A More Appropriate 2x2 Analysis for the Project

Instead, I **strongly encourage** you to use the Bayesian augmentation where we add two successes and add two failures, as recommended in Agresti and Coull[3], and to use 90% confidence levels.

```
twoby2(t1 + 2, conf.level = 0.90) # uses Bayesian augmentation, 90% confidence level
```

```
2 by 2 table analysis:
------------------------------------------------------
Outcome   : Prior R user
Comparing : English vs. Not English

            Prior R user No Prior R    P(Prior R user) 90% conf. interval
English               19         20             0.4872    0.3593   0.6167
Not English           10         12             0.4545    0.2918   0.6276

                                  90% conf. interval
             Relative Risk: 1.0718    0.6701   1.7143
         Sample Odds Ratio: 1.1400    0.4730   2.7473
Conditional MLE Odds Ratio: 1.1376    0.4174   3.1320
    Probability difference: 0.0326   -0.1791   0.2375

             Exact P-value: 1
        Asymptotic P-value: 0.8064
------------------------------------------------------
```

Note what I did to add two observations to each cell of the table. We can draw conclusions now about:

- The individual probabilities of being a prior R user in the English and non-English groups, and 90% confidence intervals for each at the top of the output, so that, for instance, we estimate the probability of prior R usage among subjects for whom English is not their best language at 0.45, with 90% confidence interval (0.29, 0.63).
- The relative risk of Prior R use given English vs. Prior R use given non-English, which is estimated to be 1.07, and based on its 90% confidence interval is clearly not statistically significantly different from 1 at $\alpha = 0.10$.
- The odds ratio describing the odds of Prior R use given English vs. Non-English, which is estimated to be 1.14, and is clearly not statistically significantly different from 1 at $\alpha = 0.10$.
- The difference in probability of Prior R use for English vs. non-English subjects, which is estimated to be 0.033, with a 90% confidence interval of (-0.18, 0.24) and is clearly not statistically significantly different from 0 at $\alpha = 0.10$.
- The chi-square test of independence, which assesses the null hypothesis of no association between language preference and prior R usage, using either Fisher's exact test[4] or the Pearson chi-square test (labeled asymptotic here.) Clearly, with a p value much greater than 0.1, we must retain the null hypothesis in this case, and we see no significant association between the rows and the columns at a 10% significance level.

10.3.1 Checking Assumptions

Since each cell in our (non-augmented) 2x2 table is at least 5, R throws no warning messages. We should be reasonably comfortable with the chi-square test of independence here. If every cell was 10 or more, we'd be even more comfortable.

[3] Agresti A Coull BA 1988 Approximate is Better than "Exact" for Interval Estimation of Binomial Proportions. The Amer Statistician 52(2), 119-126. http://www.jstor.org/stable/2685469

[4] I use Fisher's exact test with small-ish 2x2 tables where R will run it, if you have to choose between the methods.

10.4 What If We Wanted to Type in the Table Ourselves?

With the new version of `twobytwo` available in the "Love-boost.R" script, we can directly obtain 90% confidence intervals. For example, suppose we had the following data, pulled from our 2016 survey:

2016 Survey	Drank Tea Recently	Didn't Drink Tea
Not Born in US	21	10
US Born	20	18

Suppose we wanted to use `twobytwo` and the +2/+4 Bayesian augmentation (adding 2 to the count in each cell of our 2x2 table) and a 90% confidence interval for this comparison, to see whether the population proportions who drank tea recently differ between those born in and out of the US.

```
twobytwo(21+2, 10+2, 20+2, 18+2,
         "Not US Born", "US Born", "Drank Tea", "No Tea",
         conf.level = 0.90)
```

```
2 by 2 table analysis:
------------------------------------------------------
Outcome   : Drank Tea
Comparing : Not US Born vs. US Born

            Drank Tea No Tea    P(Drank Tea) 90% conf. interval
Not US Born        23     12          0.6571    0.5162   0.7749
US Born            22     20          0.5238    0.3982   0.6465

                                  90% conf. interval
            Relative Risk: 1.2545    0.9160   1.7181
        Sample Odds Ratio: 1.7424    0.8024   3.7839
Conditional MLE Odds Ratio: 1.7299    0.7276   4.1948
    Probability difference: 0.1333   -0.0512   0.3036

             Exact P-value: 0.2561
        Asymptotic P-value: 0.2389
------------------------------------------------------
```

Note that this approach will NOT work with the old version of `twobytwo`. You need the new version created by the `Love-boost.R` script.

11 Analysis 5: Two-Way (3 x 4) Contingency Table

We'll look at the association of two categorical factors we created earlier: `medium.3c` and `fiction.4c` in this analysis. We're interested in whether there is an association between the ways in which subjects consumed their fiction, and the type of fiction they most enjoy. The `medium.3c` data have three levels, and the `fiction.4c` data have four levels. **Note that we'll use a 90% confidence level and I encourage you to do this in your actual Project Study 1 work, as well.**

11.1 Building the 3x4 Table from data

11.1.1 The Simplest Table

```
table(sur15$medium.3c, sur15$fiction.4c)
```

```
        Comedy Drama Fantasy/SciFi Other
  Movies      4     5             6     2
  TV         11     5             2     4
  Other       3     5             6     0
```

Note that we still have a 0 cell here, and that might motivate us to consider collapsing or removing the "Other" category from the `fiction.4c` variable. I'll leave it alone for now, and see what happens. The question is whether which medium (Movies, TV or other) you like is associated with which genre (Comedy, Drama, Fantasy/SciFi) you like.

11.1.2 A Fancy Version

Here's a fancier version of the same information, adding some flourishes.

```
t2 <- table(sur15$medium.3c, sur15$fiction.4c)
emphasize.strong.rows(4)
emphasize.strong.cols(5)
panderOptions('table.emphasize.rownames', FALSE)
pander(addmargins(t2),
       caption = "Consumption Method vs. Genre for 53 students in 431 (Fall 2015)")
```

Table 17: Consumption Method vs. Genre for 53 students in 431 (Fall 2015)

	Comedy	Drama	Fantasy/SciFi	Other	Sum
Movies	4	5	6	2	**17**
TV	11	5	2	4	**22**
Other	3	5	6	0	**14**
Sum	**18**	**15**	**14**	**6**	**53**

11.2 Testing Association between Rows and Columns of a Contingency Table

11.2.1 Running the Pearson χ^2 Test

Now that we've stored the table of interest in `t2`, we just run the Pearson χ^2 test using:

```
chisq.test(t2)
```

```
Warning in chisq.test(t2): Chi-squared approximation may be incorrect


    Pearson's Chi-squared test

data:  t2
X-squared = 10.322, df = 6, p-value = 0.1117
```

11.2.2 Running Fisher's Exact Test

Given a small overall sample size, the `fisher.test` command will also produce a Fisher's exact test, which may be a bit more appropriate here, given the presence of cells with small counts.

```
fisher.test(t2)
```

```

    Fisher's Exact Test for Count Data

data:  t2
p-value = 0.0983
alternative hypothesis: two.sided
```

Based on the Fisher test, we would just barely declare the association statistically significant at the 90% confidence level, since the p value is just below 0.10. If we used the Pearson test, we'd have to declare the association not to be statistically significant. But...

a. The difference between statistically significant and not statistically significant is small here.
b. Neither test is really appropriate, since we have very small cell counts, including a zero.

11.2.3 Checking Assumptions - The Cochran Conditions

There is something called the "Cochran conditions", which require that we have:

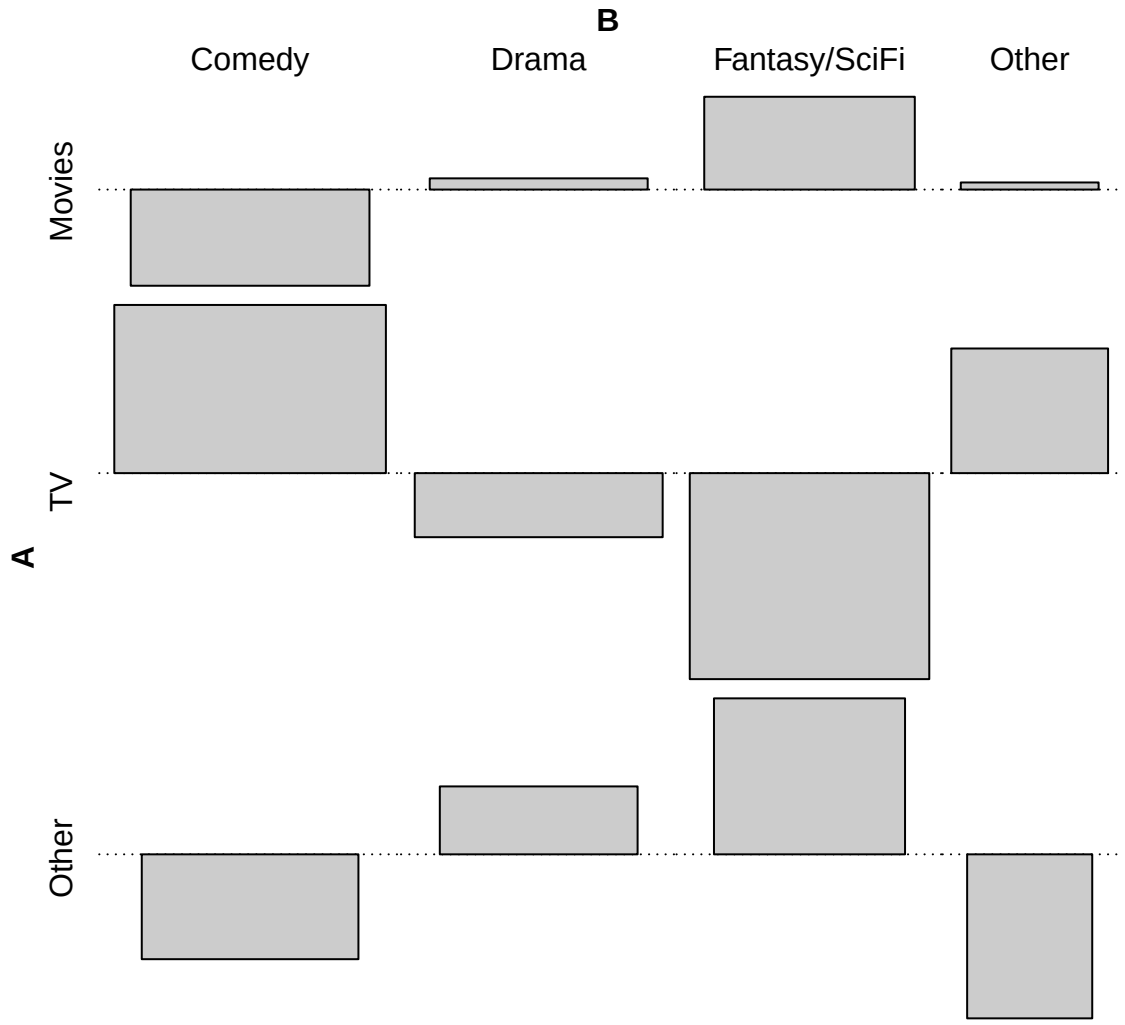
- no cells with 0 counts
- at least 80% of the cells in our table with counts of 5 or higher

We don't meet those Cochran conditions here. In addition, since each cell in our 3x4 table is NOT at least 5, R throws a warning message when we run the Pearson χ^2 test, and since we don't meet the Cochran conditions, the `fisher.test` results are a bit questionable, as well. We should consider whether collapsing or deleting some of the rows or columns might be more reasonable. And we'll do this next.

11.3 An Association Plot for the 3x4 Table

The command `assoc` in the `vcd` library in R produces a plot that indicates deviations from the assumption of independence of rows and columns in a two-way table. For instance, using our original table, we have:

```
assoc(t2)
```



We can see that the independence model really doesn't work well for the cells with larger shapes here, which we note especially in the Fantasy/SciFi column, and to some extent in the Comedy column.

11.4 A 2x3 Table, After Collapsing (Lumping) Some Small Rows and Columns

Suppose we decided to drop down to a study of TV vs. Other media (combining Movies and Other) and also collapsed the Fantasy/SciFi and Other columns (so the remaining subjects form a 2x3 table), in an effort to remove zero cells, and reduce the incidence of cells with counts below 5.

11.4.1 Lumping Together Categories

First, we'll combine the Movies and Other groups to create `medium.2c` from `medium.3c` using `fct_recode`.

```
sur15$medium.2c <- fct_recode(sur15$medium.3c,
                              "Not TV" = "Movies",
                              "TV" = "TV",
                              "Not TV" = "Other")
table(sur15$medium.2c, sur15$medium.3c) # sanity check
```

```
        Movies TV Other
  Not TV     17  0    14
  TV          0 22     0
```

Or, we can use the `fct_lump` function to lump together the two categories with the smallest overall counts directly, in creating `fiction.3c` from `fiction.4c`.

```
sur15$fiction.3c <- fct_lump(sur15$fiction.4c, 2)
table(sur15$fiction.3c, sur15$fiction.4c) # sanity check
```

```
        Comedy Drama Fantasy/SciFi Other
  Comedy     18     0             0     0
  Drama       0    15             0     0
  Other       0     0            14     6
```

11.4.2 The Collapsed 2x3 Contingency Table

OK. Here's the new table.

```
t3 <- table(sur15$medium.2c, sur15$fiction.3c)
emphasize.strong.rows(3)
emphasize.strong.cols(4)
panderOptions('table.emphasize.rownames', FALSE)
pander(addmargins(t3),
       caption = "Collapsed Consumption Method vs. Collapsed Genre")
```

Table 18: Collapsed Consumption Method vs. Collapsed Genre

	Comedy	Drama	Other	Sum
Not TV	7	10	14	**31**
TV	11	5	6	**22**
Sum	**18**	**15**	**20**	**53**

This new 2x3 table loses some fidelity, but gains in that each cell now contains at least 5 subjects.

11.4.3 Chi-Square Testing for the 2x3 Table

And here are the results from chi-square testing...

```
chisq.test(t3)
```

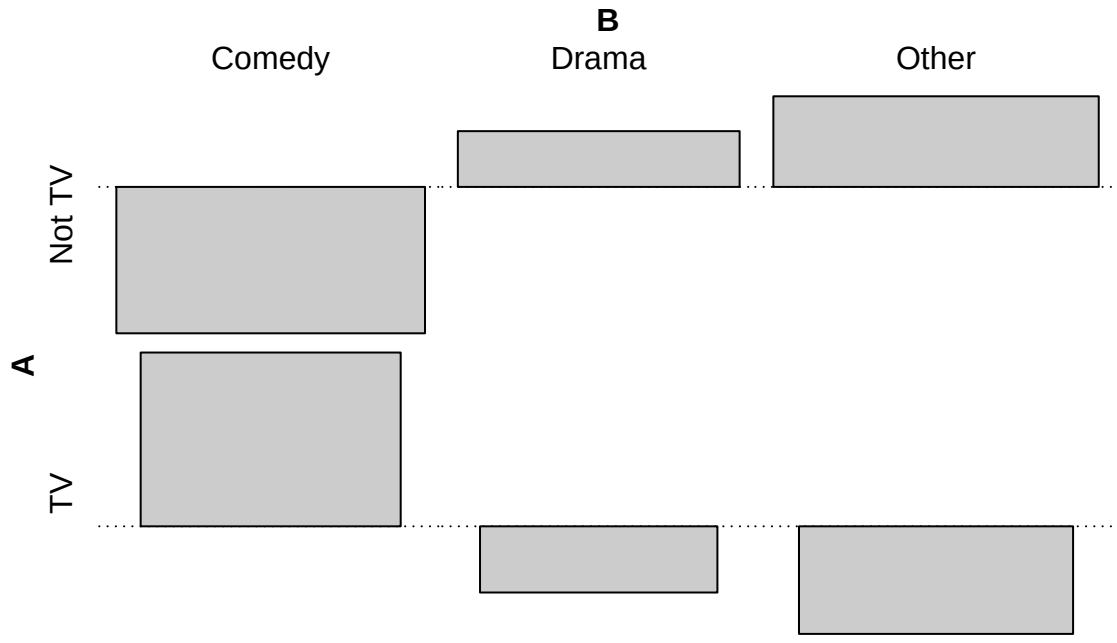
```
    Pearson's Chi-squared test

data:  t3
X-squared = 4.3528, df = 2, p-value = 0.1135
```

```
fisher.test(t3)
```

```
    Fisher's Exact Test for Count Data

data:  t3
p-value = 0.1483
alternative hypothesis: two.sided
```

For the project, once all of the cells have at least 5 observations, I recommend the use of the Pearson approach, unless the table is square (# of rows = # of columns), in which case the Fisher test is also a reasonable choice. Generally, the Fisher test is more appropriate when the sample sizes are small. In this case, of course, it doesn't matter much after collapsing cells and forming this 2x3 table. We'll close with the association plot for this smaller table, which suggests that the independence model inverts its errors for Comedy as compared to the other two categories.

```
assoc(t3)
```



We conclude, using either the Pearson or Fisher test (at the 10% significance level), that there is no statistically significant association between the favorite consumption method and favorite genre.

12 Analysis 6: Three-Way Contingency Table

We'll look at the association of `prior.r` and `english` stratified by `seat` in this analysis. Each of the three variables is categorical, and `prior.r` and `english` have two levels, while `seat` has three. We're interested in whether the rows (`english`) and columns (`prior.r`) from our previous two-by-two table show an association that changes depending on a stratifying variable, `seat`. **Note that we'll use a 90% confidence level and I encourage you to do this in your actual Project Study 1 work, as well.**

12.1 Compiling the Three-Way Contingency Table

12.1.1 The Simplest Version of a Three-Way Table

We start by asking for a table with `english` in the rows, `prior.r` in the columns, and each subtable stratified by `seat`, as follows...

```
t4 <- table(sur15$english, sur15$prior.r, sur15$seat)
t4
```

```
, ,  = Left


      Yes No
  Yes   7  7
  No    2  4

, ,  = Middle


      Yes No
  Yes   5  6
  No    2  1

, ,  = Right


      Yes No
  Yes   5  5
  No    4  5
```

Those "Yes" and "No" results are very confusing.

12.1.2 Adjusting Names of Columns and Rows

```
rownames(t4) <- c("English", "Not English")
colnames(t4) <- c("Prior R user", "No Prior R")
addmargins(t4)
```

```
, ,  = Left


             Prior R user No Prior R Sum
  English               7          7  14
  Not English           2          4   6
```

```
  Sum                    9         11  20

, ,  = Middle


              Prior R user No Prior R Sum
  English                5          6  11
  Not English            2          1   3
  Sum                    7          7  14

, ,  = Right


              Prior R user No Prior R Sum
  English                5          5  10
  Not English            4          5   9
  Sum                    9         10  19

, ,  = Sum


              Prior R user No Prior R Sum
  English               17         18  35
  Not English            8         10  18
  Sum                   25         28  53
```

12.1.3 Flattening the Table

The most streamlined way to present a three-way table is through flattening it. But a flattened table is more helpful when you change the order of the variables a bit to put the stratum first, then the rows, then the columns.

```
t4.flat <- table(sur15$seat, sur15$english, sur15$prior.r)
dimnames(t4.flat)[[1]] <- c("Left", "Middle", "Right") # specify the names in t5's first variable
dimnames(t4.flat)[[2]] <- c("English", "Not English") # specify names in t5's second variable
dimnames(t4.flat)[[3]] <- c("Prior R user", "No Prior R") # specify names in t5's third variable
ftable(t4.flat)
```

```
                  Prior R user No Prior R

Left   English                7          7
       Not English            2          4
Middle English                5          6
       Not English            2          1
Right  English                5          5
       Not English            4          5
```

12.2 Checking Assumptions with the Woolf Test

We'll begin by checking the assumptions of the Cochran-Mantel-Haenszel test for our original table, `t4`. To start, we'll be using the Woolf test for interaction to see if the assumption of equal population odds ratios within each stratum is clearly violated. The approach I use is provided in the `vcd` package.

```
woolf_test(t4)
```

```
    Woolf-test on Homogeneity of Odds Ratios (no 3-Way assoc.)

data:  t4
X-squared = 0.85309, df = 2, p-value = 0.6528
```

The null hypothesis for the Woolf test is that the odds ratios are homogenous across the three strata. Here, we cannot reject that hypothesis. If we could, then the CMH test would definitely be inappropriate.

However, given the small sample size in several of the cells, the CMH test is not likely to be especially accurate - we'd certainly prefer if all of the cells had at least 5 cases.

12.3 The Cochran-Mantel-Haenzel Test

The Cochran-Mantel-Haenszel (CMH) test works with our original table `t4`, and it assesses whether the odds ratio describing the association of `english` and `prior.r` is significantly different from 1, after accounting for differences between the three groups of `seat` responses. The CMH test assumes that the odds ratio is (in the population) identical in each `seat` group, or stratum. We checked that assumption (to the extent possible) with the Woolf test. The CMH test is, by default in R, run with a continuity correction using the `mantelhaen.test` function, as follows...

```
mantelhaen.test(t4, conf.level = .90)
```

```
    Mantel-Haenszel chi-squared test without continuity correction

data:  t4
Mantel-Haenszel X-squared = 0.072097, df = 1, p-value = 0.7883
alternative hypothesis: true common odds ratio is not equal to 1
90 percent confidence interval:
 0.4427956 3.1310986
sample estimates:
common odds ratio
          1.17747
```

Our point estimate for the common population odds ratio comparing rates of `prior.r` use among those who prefer `english` and those who don't is 1.18, with 90% confidence interval (0.44, 3.13). So we'd conclude that there is no statistically significant association between `prior.r` and `english` after accounting for `seat` preference. We can also see this through the p value of 0.79, which is much larger than $\alpha = 0.10$.