Data Science for Biological, Medical and Health Research: Notes for 432

Thomas E. Love, Ph.D.

Built 2018-01-22 06:39:12

Contents

Introduction

These Notes provide a series of examples using R to work through issues that are likely to come up in PQHS/CRSP/MPHP 432.

While these Notes share some of the features of a textbook, they are neither comprehensive nor completely original. The main purpose is to give students in 432 a set of common materials on which to draw during the course. In class, we will sometimes:

- reiterate points made in this document,
- amplify what is here,
- simplify the presentation of things done here,
- use new examples to show some of the same techniques,
- refer to issues not mentioned in this document,

but what we don't (always) do is follow these notes very precisely. We assume instead that you will read the materials and try to learn from them, just as you will attend classes and try to learn from them. We welcome feedback of all kinds on this document or anything else. Just email us at `431-help at case dot edu`, or submit a pull request. Note that we still use `431-help` even though we're now in 432.

What you will mostly find are brief explanations of a key idea or summary, accompanied (most of the time) by R code and a demonstration of the results of applying that code.

Everything you see here is available to you as HTML or PDF. You will also have access to the R Markdown files, which contain the code which generates everything in the document, including all of the R results. We will demonstrate the use of R Markdown (this document is generated with the additional help of an R package called bookdown) and R Studio (the "program" which we use to interface with the R language) in class.

To download the data and R code related to these notes, visit the Data and Code section of the 432 course website.

R Packages used in these notes

Here, we'll load in the packages used in these notes.

```r
library(tableone)
library(skimr)
library(ggridges)
library(simputation)
library(magrittr)
library(modelr)
library(broom)
library(tidyverse)
```

Data used in these notes

Here, we'll load in the data sets used in these notes.

```
fakestroke <- read.csv("data/fakestroke.csv") %>% tbl_df
bloodbrain <- read.csv("data/bloodbrain.csv") %>% tbl_df
smartcle1 <- read.csv("data/smartcle1.csv") %>% tbl_df
bonding <- read.csv("data/bonding.csv") %>% tbl_df
cortisol <- read.csv("data/cortisol.csv") %>% tbl_df
emphysema <- read.csv("data/emphysema.csv") %>% tbl_df
```

Chapter 1

Building Table 1

Many scientific articles involve direct comparison of results from various exposures, perhaps treatments. In 431, we studied numerous methods, including various sorts of hypothesis tests, confidence intervals, and descriptive summaries, which can help us to understand and compare outcomes in such a setting. One common approach is to present what's often called Table 1. Table 1 provides a summary of the characteristics of a sample, or of groups of samples, which is most commonly used to help understand the nature of the data being compared.

1.1 Two examples from the *New England Journal of Medicine*

1.1.1 A simple Table 1

Table 1 is especially common in the context of clinical research. Consider the excerpt below, from a January 2015 article in the *New England Journal of Medicine* (Tolaney et al., 2015).

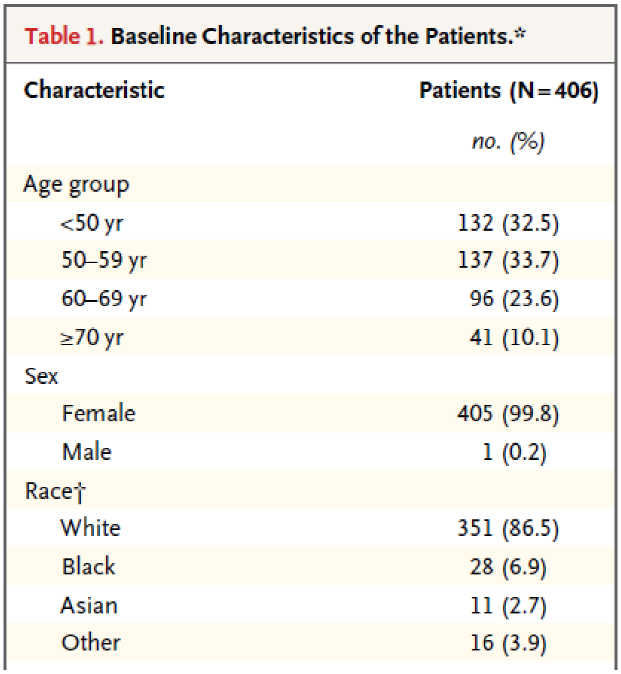


Table 1. Baseline Characteristics of the Patients.*

Characteristic	Patients (N=406)
	no. (%)
Age group	
<50 yr	132 (32.5)
50–59 yr	137 (33.7)
60–69 yr	96 (23.6)
≥70 yr	41 (10.1)
Sex	
Female	405 (99.8)
Male	1 (0.2)
Race†	
White	351 (86.5)
Black	28 (6.9)
Asian	11 (2.7)
Other	16 (3.9)

This (partial) table reports baseline characteristics on age group, sex and race, describing 406 patients with

HER2-positive[1] invasive breast cancer that began the protocol therapy. Age, sex and race (along with severity of illness) are the most commonly identified characteristics in a Table 1.

In addition to the measures shown in this excerpt, the full Table also includes detailed information on the primary tumor for each patient, including its size, nodal status and histologic grade. Footnotes tell us that the percentages shown are subject to rounding, and may not total 100, and that the race information was self-reported.

1.1.2 A group comparison

A more typical Table 1 involves a group comparison, for example in this excerpt from Roy et al. (2008). This Table 1 describes a multi-center randomized clinical trial comparing two different approaches to caring for patients with heart failure and atrial fibrillation[2].

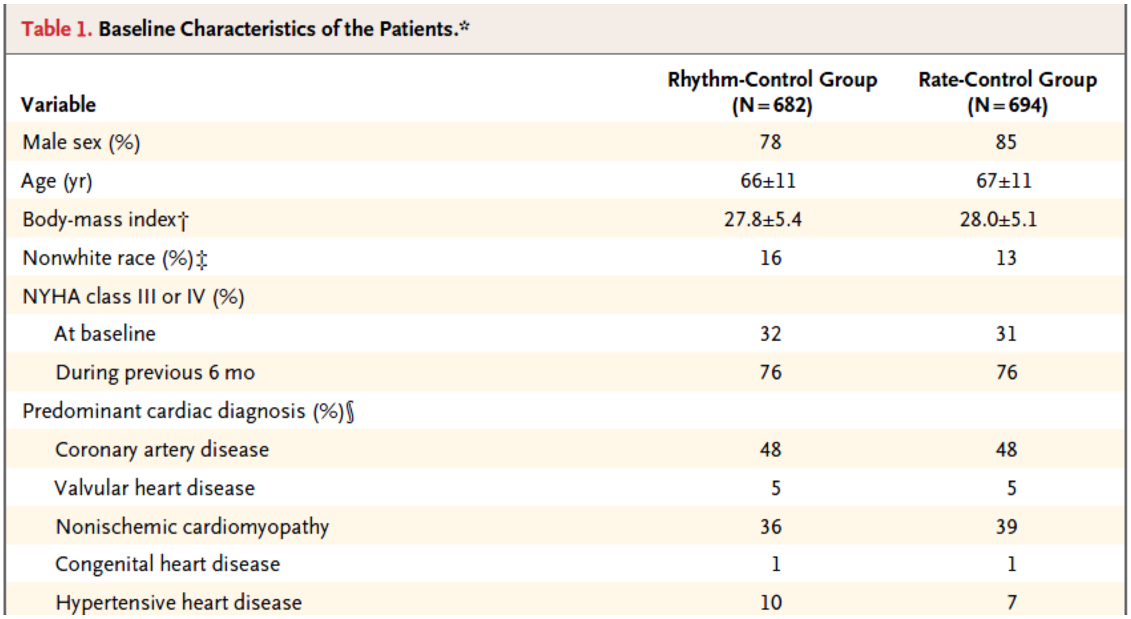


Table 1. Baseline Characteristics of the Patients.*

Variable	Rhythm-Control Group (N = 682)	Rate-Control Group (N = 694)
Male sex (%)	78	85
Age (yr)	66±11	67±11
Body-mass index†	27.8±5.4	28.0±5.1
Nonwhite race (%)‡	16	13
NYHA class III or IV (%)		
At baseline	32	31
During previous 6 mo	76	76
Predominant cardiac diagnosis (%)§		
Coronary artery disease	48	48
Valvular heart disease	5	5
Nonischemic cardiomyopathy	36	39
Congenital heart disease	1	1
Hypertensive heart disease	10	7

The article provides percentages, means and standard deviations across groups, but note that it does not provide p values for the comparison of baseline characteristics. This is a common feature of NEJM reports on randomized clinical trials, where we anticipate that the two groups will be well matched at baseline. Note that the patients in this study were *randomly* assigned to either the rhythm-control group or to the rate-control group, using blocked randomizations stratified by study center.

1.2 The MR CLEAN trial

Berkhemer et al. (2015) reported on the MR CLEAN trial, involving 500 patients with acute ischemic stroke caused by a proximal intracranial arterial occlusion. The trial was conducted at 16 medical centers in the Netherlands, where 233 were randomly assigned to the intervention (intraarterial treatment plus usual care) and 267 to control (usual care alone.) The primary outcome was the modified Rankin scale score at 90 days; this categorical scale measures functional outcome, with scores ranging from 0 (no symptoms) to 6 (death). The fundamental conclusion of Berkhemer et al. (2015) was that in patients with acute ischemic stroke

[1] HER2 = human epidermal growth factor receptor type 2. Over-expression of this occurs in 15-20% of invasive breast cancers, and has been associated with poor outcomes.

[2] The complete Table 1 appears on pages 2668-2669 of Roy et al. (2008), but I have only reproduced the first page and the footnote in this excerpt.

caused by a proximal intracranial occlusion of the anterior circulation, intraarterial treatment administered within 6 hours after stroke onset was effective and safe.

Here's the Table 1 from Berkhemer et al. (2015).

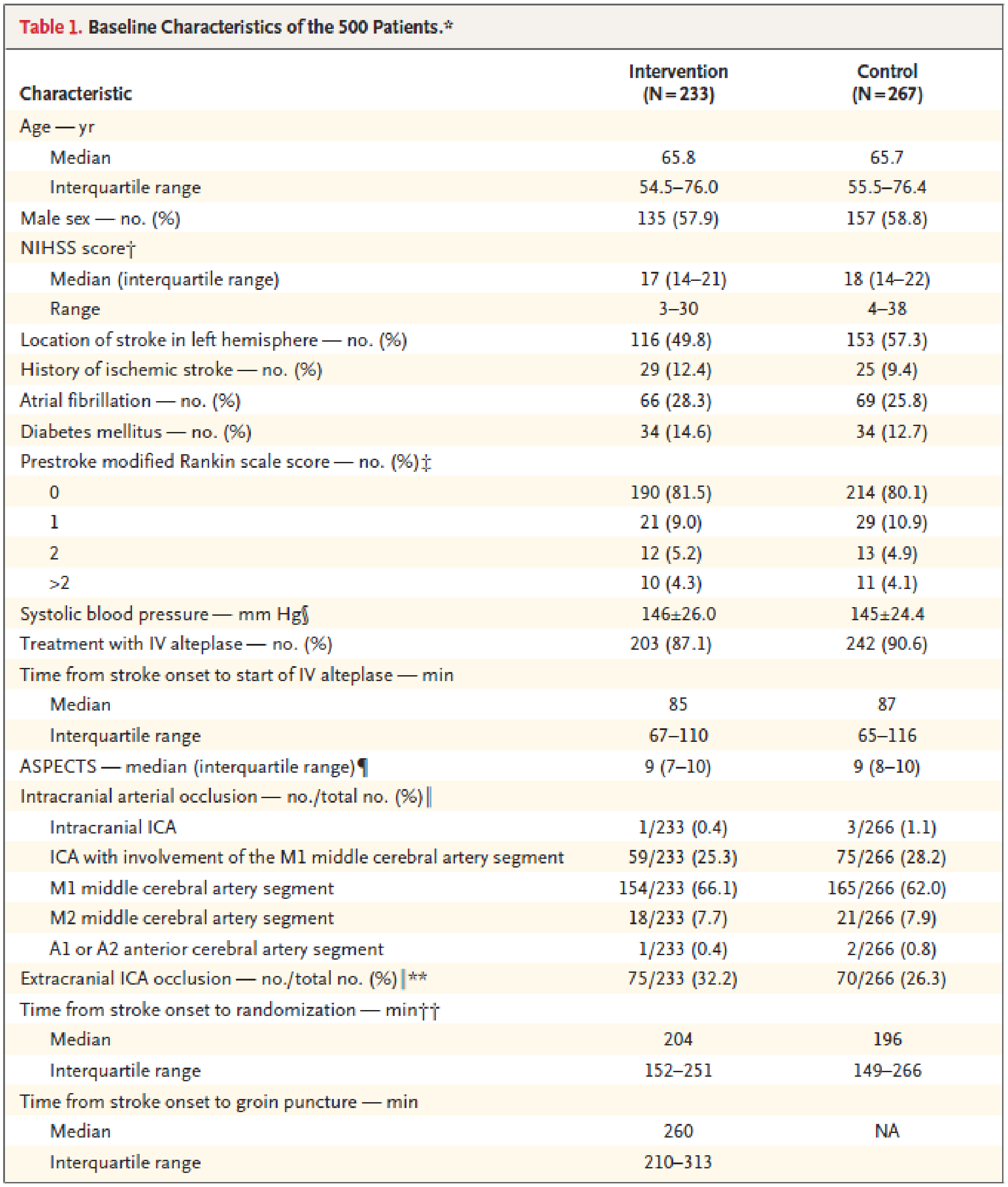
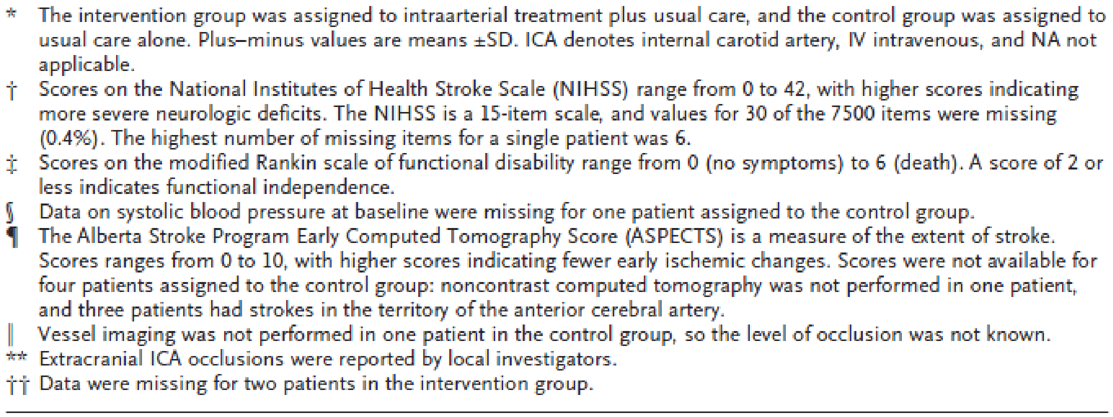


Table 1. Baseline Characteristics of the 500 Patients.*

Characteristic	Intervention (N = 233)	Control (N = 267)
Age — yr		
Median	65.8	65.7
Interquartile range	54.5–76.0	55.5–76.4
Male sex — no. (%)	135 (57.9)	157 (58.8)
NIHSS score†		
Median (interquartile range)	17 (14–21)	18 (14–22)
Range	3–30	4–38
Location of stroke in left hemisphere — no. (%)	116 (49.8)	153 (57.3)
History of ischemic stroke — no. (%)	29 (12.4)	25 (9.4)
Atrial fibrillation — no. (%)	66 (28.3)	69 (25.8)
Diabetes mellitus — no. (%)	34 (14.6)	34 (12.7)
Prestroke modified Rankin scale score — no. (%)‡		
0	190 (81.5)	214 (80.1)
1	21 (9.0)	29 (10.9)
2	12 (5.2)	13 (4.9)
>2	10 (4.3)	11 (4.1)
Systolic blood pressure — mm Hg§	146±26.0	145±24.4
Treatment with IV alteplase — no. (%)	203 (87.1)	242 (90.6)
Time from stroke onset to start of IV alteplase — min		
Median	85	87
Interquartile range	67–110	65–116
ASPECTS — median (interquartile range)¶	9 (7–10)	9 (8–10)
Intracranial arterial occlusion — no./total no. (%)‖		
Intracranial ICA	1/233 (0.4)	3/266 (1.1)
ICA with involvement of the M1 middle cerebral artery segment	59/233 (25.3)	75/266 (28.2)
M1 middle cerebral artery segment	154/233 (66.1)	165/266 (62.0)
M2 middle cerebral artery segment	18/233 (7.7)	21/266 (7.9)
A1 or A2 anterior cerebral artery segment	1/233 (0.4)	2/266 (0.8)
Extracranial ICA occlusion — no./total no. (%)‖**	75/233 (32.2)	70/266 (26.3)
Time from stroke onset to randomization — min††		
Median	204	196
Interquartile range	152–251	149–266
Time from stroke onset to groin puncture — min		
Median	260	NA
Interquartile range	210–313	

The Table was accompanied by the following notes.



* The intervention group was assigned to intraarterial treatment plus usual care, and the control group was assigned to usual care alone. Plus–minus values are means ±SD. ICA denotes internal carotid artery, IV intravenous, and NA not applicable.

† Scores on the National Institutes of Health Stroke Scale (NIHSS) range from 0 to 42, with higher scores indicating more severe neurologic deficits. The NIHSS is a 15-item scale, and values for 30 of the 7500 items were missing (0.4%). The highest number of missing items for a single patient was 6.

‡ Scores on the modified Rankin scale of functional disability range from 0 (no symptoms) to 6 (death). A score of 2 or less indicates functional independence.

§ Data on systolic blood pressure at baseline were missing for one patient assigned to the control group.

¶ The Alberta Stroke Program Early Computed Tomography Score (ASPECTS) is a measure of the extent of stroke. Scores ranges from 0 to 10, with higher scores indicating fewer early ischemic changes. Scores were not available for four patients assigned to the control group: noncontrast computed tomography was not performed in one patient, and three patients had strokes in the territory of the anterior cerebral artery.

‖ Vessel imaging was not performed in one patient in the control group, so the level of occlusion was not known.

** Extracranial ICA occlusions were reported by local investigators.

†† Data were missing for two patients in the intervention group.

1.3 Simulated `fakestroke` data

Consider the simulated data, available on the Data and Code page of our course website in the `fakestroke.csv` file, which I built to let us mirror the Table 1 for MR CLEAN (Berkhemer et al., 2015). The `fakestroke.csv` file contains the following 18 variables for 500 patients.

Variable	Description
`studyid`	Study ID # (z001 through z500)
`trt`	Treatment group (Intervention or Control)
`age`	Age in years
`sex`	Male or Female
`nihss`	NIH Stroke Scale Score (can range from 0-42; higher scores indicate more severe neurological deficits)
`location`	Stroke Location - Left or Right Hemisphere
`hx.isch`	History of Ischemic Stroke (Yes/No)
`afib`	Atrial Fibrillation (1 = Yes, 0 = No)
`dm`	Diabetes Mellitus (1 = Yes, 0 = No)
`mrankin`	Pre-stroke modified Rankin scale score (0, 1, 2 or > 2) indicating functional disability - complete range is 0 (no symptoms) to 6 (death)
`sbp`	Systolic blood pressure, in mm Hg
`iv.altep`	Treatment with IV alteplase (Yes/No)
`time.iv`	Time from stroke onset to start of IV alteplase (minutes) if iv.altep=Yes
`aspects`	Alberta Stroke Program Early Computed Tomography score, which measures extent of stroke from 0 - 10; higher scores indicate fewer early ischemic changes
`ia.occlus`	Intracranial arterial occlusion, based on vessel imaging - five categories[3]
`extra.ica`	Extracranial ICA occlusion (1 = Yes, 0 = No)
`time.rand`	Time from stroke onset to study randomization, in minutes
`time.punc`	Time from stroke onset to groin puncture, in minutes (only if Intervention)

Here's a quick look at the simulated data in `fakestroke`.

[3] The five categories are Intracranial ICA, ICA with involvement of the M1 middle cerebral artery segment, M1 middle cerebral artery segment, M2 middle cerebral artery segment, A1 or A2 anterior cerebral artery segment

```
fakestroke
```

```
# A tibble: 500 x 18
   studyid trt         age sex   nihss location hx.isch  afib    dm mrankin
   <fct>   <fct>     <dbl> <fct> <int> <fct>    <fct>   <int> <int> <fct>
 1 z001    Control    53.0 Male     21 Right    No          0     0 2
 2 z002    Interve~   51.0 Male     23 Left     No          1     0 0
 3 z003    Control    68.0 Fema~    11 Right    No          0     0 0
 4 z004    Control    28.0 Male     22 Left     No          0     0 0
 5 z005    Control    91.0 Male     24 Right    No          0     0 0
 6 z006    Control    34.0 Fema~    18 Left     No          0     0 2
 7 z007    Interve~   75.0 Male     25 Right    No          0     0 0
 8 z008    Control    89.0 Fema~    18 Right    No          0     0 0
 9 z009    Control    75.0 Male     25 Left     No          1     0 2
10 z010    Interve~   26.0 Fema~    27 Right    No          0     0 0
# ... with 490 more rows, and 8 more variables: sbp <int>, iv.altep <fct>,
#   time.iv <int>, aspects <int>, ia.occlus <fct>, extra.ica <int>,
#   time.rand <int>, time.punc <int>
```

1.4 Building Table 1 for `fakestroke`: Attempt 1

Our goal, then, is to take the data in `fakestroke.csv` and use it to generate a Table 1 for the study that compares the 233 patients in the Intervention group to the 267 patients in the Control group, on all of the other variables (except study ID #) available. I'll use the `tableone` package of functions available in R to help me complete this task. We'll make a first attempt, using the `CreateTableOne` function in the `tableone` package. To use the function, we'll need to specify:

- the `vars` or variables we want to place in the rows of our Table 1 (which will include just about everything in the `fakestroke` data except the `studyid` code and the `trt` variable for which we have other plans, and the `time.punc` which applies only to subjects in the Intervention group.)
  - A useful trick here is to use the `dput` function, specifically something like `dput(names(fakestroke))` can be used to generate a list of all of the variables included in the `fakestroke` tibble, and then this can be copied and pasted into the `vars` specification, saving some typing.
- the `strata` which indicates the levels want to use in the columns of our Table 1 (for us, that's `trt`)

```
fs.vars <- c("age", "sex", "nihss", "location",
          "hx.isch", "afib", "dm", "mrankin", "sbp",
          "iv.altep", "time.iv", "aspects",
          "ia.occlus", "extra.ica", "time.rand")

fs.trt <- c("trt")

att1 <- CreateTableOne(data = fakestroke,
                       vars = fs.vars,
                       strata = fs.trt)
print(att1)
```

```
                       Stratified by trt
                        Control        Intervention   p      test
  n                        267            233
  age (mean (sd))        65.38 (16.10)  63.93 (18.09)  0.343
  sex = Male (%)           157 (58.8)     135 (57.9)   0.917
  nihss (mean (sd))      18.08 (4.32)   17.97 (5.04)   0.787
  location = Right (%)     114 (42.7)     117 (50.2)   0.111
```

```
hx.isch = Yes (%)         25 ( 9.4)      29 (12.4)   0.335
afib (mean (sd))        0.26 (0.44)    0.28 (0.45)   0.534
dm (mean (sd))          0.13 (0.33)    0.12 (0.33)   0.923
mrankin (%)                                          0.922
   > 2                    11 ( 4.1)      10 ( 4.3)
   0                     214 (80.1)     190 (81.5)
   1                      29 (10.9)      21 ( 9.0)
   2                      13 ( 4.9)      12 ( 5.2)
sbp (mean (sd))       145.00 (24.40) 146.03 (26.00)  0.647
iv.altep = Yes (%)       242 (90.6)     203 (87.1)   0.267
time.iv (mean (sd))    87.96 (26.01)  98.22 (45.48)  0.003
aspects (mean (sd))     8.65 (1.47)    8.35 (1.64)   0.033
ia.occlus (%)                                        0.795
   A1 or A2                2 ( 0.8)       1 ( 0.4)
   ICA with M1            75 (28.2)      59 (25.3)
   Intracranial ICA        3 ( 1.1)       1 ( 0.4)
   M1                    165 (62.0)     154 (66.1)
   M2                     21 ( 7.9)      18 ( 7.7)
extra.ica (mean (sd))   0.26 (0.44)    0.32 (0.47)   0.150
time.rand (mean (sd)) 213.88 (70.29) 202.51 (57.33)  0.051
```

1.4.1 Some of this is very useful, and other parts need to be fixed.

1. The 1/0 variables (`afib`, `dm`, `extra.ica`) might be better if they were treated as the factors they are, and reported as the Yes/No variables are reported, with counts and percentages rather than with means and standard deviations.
2. In some cases, we may prefer to re-order the levels of the categorical (factor) variables, particularly the `mrankin` variable, but also the `ia.occlus` variable. It would also be more typical to put the Intervention group to the left and the Control group to the right, so we may need to adjust our `trt` variable's levels accordingly.
3. For each of the quantitative variables (`age`, `nihss`, `sbp`, `time.iv`, `aspects`, `extra.ica`, `time.rand` and `time.punc`) we should make a decision whether a summary with mean and standard deviation is appropriate, or whether we should instead summarize with, say, the median and quartiles. A mean and standard deviation really only yields an appropriate summary when the data are least approximately Normally distributed. This will make the *p* values a bit more reasonable, too. The `test` column in the first attempt will soon have something useful to tell us.
4. If we'd left in the `time.punc` variable, we'd get some warnings, having to do with the fact that `time.punc` is only relevant to patients in the Intervention group.

1.4.2 `fakestroke` Cleaning Up Categorical Variables

Let's specify each of the categorical variables as categorical explicitly. This helps the `CreateTableOne` function treat them appropriately, and display them with counts and percentages. This includes all of the 1/0, Yes/No and multi-categorical variables.

```
fs.factorvars <- c("sex", "location", "hx.isch", "afib", "dm",
                   "mrankin", "iv.altep", "ia.occlus", "extra.ica")
```

Then we simply add a `factorVars = fs.factorvars` call to the `CreateTableOne` function.

We also want to re-order some of those categorical variables, so that the levels are more useful to us. Specifically, we want to:

- place Intervention before Control in the `trt` variable,
- reorder the `mrankin` scale as 0, 1, 2, > 2, and

- rearrange the `ia.occlus` variable to the order[4] presented in Berkhemer et al. (2015).

To accomplish this, we'll use the `fct_relevel` function from the `forcats` package (loaded with the rest of the core `tidyverse` packages) to reorder our levels manually.

```
fakestroke <- fakestroke %>%
    mutate(trt = fct_relevel(trt, "Intervention", "Control"),
           mrankin = fct_relevel(mrankin, "0", "1", "2", "> 2"),
           ia.occlus = fct_relevel(ia.occlus, "Intracranial ICA",
                                   "ICA with M1", "M1", "M2",
                                   "A1 or A2")
           )
```

1.5 fakestroke Table 1: Attempt 2

```
att2 <- CreateTableOne(data = fakestroke,
                       vars = fs.vars,
                       factorVars = fs.factorvars,
                       strata = fs.trt)
print(att2)
```

```
                       Stratified by trt
                        Intervention   Control        p      test
  n                        233            267                     
  age (mean (sd))        63.93 (18.09)  65.38 (16.10)  0.343     
  sex = Male (%)           135 (57.9)     157 (58.8)   0.917     
  nihss (mean (sd))      17.97 (5.04)   18.08 (4.32)   0.787     
  location = Right (%)     117 (50.2)     114 (42.7)   0.111     
  hx.isch = Yes (%)         29 (12.4)      25 ( 9.4)   0.335     
  afib = 1 (%)              66 (28.3)      69 (25.8)   0.601     
  dm = 1 (%)                29 (12.4)      34 (12.7)   1.000     
  mrankin (%)                                          0.922     
     0                     190 (81.5)     214 (80.1)             
     1                      21 ( 9.0)      29 (10.9)             
     2                      12 ( 5.2)      13 ( 4.9)             
     > 2                    10 ( 4.3)      11 ( 4.1)             
  sbp (mean (sd))       146.03 (26.00) 145.00 (24.40)  0.647     
  iv.altep = Yes (%)       203 (87.1)     242 (90.6)   0.267     
  time.iv (mean (sd))    98.22 (45.48)  87.96 (26.01)  0.003     
  aspects (mean (sd))     8.35 (1.64)    8.65 (1.47)   0.033     
  ia.occlus (%)                                        0.795     
     Intracranial ICA        1 ( 0.4)       3 ( 1.1)             
     ICA with M1            59 (25.3)      75 (28.2)             
     M1                    154 (66.1)     165 (62.0)             
     M2                     18 ( 7.7)      21 ( 7.9)             
     A1 or A2                1 ( 0.4)       2 ( 0.8)             
  extra.ica = 1 (%)         75 (32.2)      70 (26.3)   0.179     
  time.rand (mean (sd)) 202.51 (57.33) 213.88 (70.29)  0.051     
```

The categorical data presentation looks much improved.

[4] We might also have considered reordering the `ia.occlus` factor by its frequency, using the `fct_infreq` function

1.5.1 What summaries should we show?

Now, we'll move on to the issue of making a decision about what type of summary to show for the quantitative variables. Since the `fakestroke` data are just simulated and only match the summary statistics of the original results, not the details, we'll adopt the decisions made by Berkhemer et al. (2015), which were to use medians and interquartile ranges to summarize the distributions of all of the continuous variables **except** systolic blood pressure.

- Specifying certain quantitative variables as *non-normal* causes R to show them with medians and the 25th and 75th percentiles, rather than means and standard deviations, and also causes those variables to be tested using non-parametric tests, like the Wilcoxon signed rank test, rather than the t test. The `test` column indicates this with the word `nonnorm`.
  – In real data situations, what should we do? The answer is to look at the data. I would not make the decision as to which approach to take without first plotting (perhaps in a histogram or a Normal Q-Q plot) the observed distributions in each of the two samples, so that I could make a sound decision about whether Normality was a reasonable assumption. If the means and medians are meaningfully different from each other, this is especially important.
  – To be honest, though, if the variable in question is a relatively unimportant covariate and the *p* values for the two approaches are nearly the same, I'm not sure that further investigation is especially important,
- Specifying *exact* tests for certain categorical variables (we'll try this for the `location` and `mrankin` variables) can be done, and these changes will be noted in the `test` column, as well.
  – In real data situations, I would rarely be concerned about this issue, and often choose Pearson (approximate) options across the board. This is reasonable so long as the number of subjects falling in each category is reasonably large, say above 10. If not, then an exact test may be an improvement.

To accomplish the Table 1, then, we need to specify which variables should be treated as non-Normal in the `print` statement - notice that we don't need to redo the `CreateTableOne` for this change.

```r
print(att2,
      nonnormal = c("age", "nihss", "time.iv", "aspects", "time.rand"),
      exact = c("location", "mrankin"))
```

```
                          Stratified by trt
                           Intervention            Control
  n                           233                     267
  age (median [IQR])        65.80 [54.50, 76.00]    65.70 [55.75, 76.20]
  sex = Male (%)              135 (57.9)              157 (58.8)
  nihss (median [IQR])      17.00 [14.00, 21.00]    18.00 [14.00, 22.00]
  location = Right (%)        117 (50.2)              114 (42.7)
  hx.isch = Yes (%)            29 (12.4)               25 ( 9.4)
  afib = 1 (%)                 66 (28.3)               69 (25.8)
  dm = 1 (%)                   29 (12.4)               34 (12.7)
  mrankin (%)
     0                        190 (81.5)              214 (80.1)
     1                         21 ( 9.0)               29 (10.9)
     2                         12 ( 5.2)               13 ( 4.9)
     > 2                       10 ( 4.3)               11 ( 4.1)
  sbp (mean (sd))          146.03 (26.00)          145.00 (24.40)
  iv.altep = Yes (%)          203 (87.1)              242 (90.6)
  time.iv (median [IQR])    85.00 [67.00, 110.00]   87.00 [65.00, 116.00]
  aspects (median [IQR])     9.00 [7.00, 10.00]      9.00 [8.00, 10.00]
  ia.occlus (%)
     Intracranial ICA           1 ( 0.4)                3 ( 1.1)
     ICA with M1               59 (25.3)               75 (28.2)
     M1                       154 (66.1)              165 (62.0)        
     M2                        18 ( 7.7)               21 ( 7.9)        
     A1 or A2                   1 ( 0.4)                2 ( 0.8)        
  extra.ica = 1 (%)            75 (32.2)               70 (26.3)        
  time.rand (median [IQR]) 204.00 [152.00, 249.50] 196.00 [149.00, 266.00]
                          Stratified by trt
                           p      test   
  n                                      
  age (median [IQR])        0.579 nonnorm
  sex = Male (%)            0.917        
  nihss (median [IQR])      0.453 nonnorm
  location = Right (%)      0.106 exact  
  hx.isch = Yes (%)         0.335        
  afib = 1 (%)              0.601        
  dm = 1 (%)                1.000        
  mrankin (%)               0.917 exact  
     0                                   
     1                                   
     2                                   
     > 2                                 
  sbp (mean (sd))           0.647        
  iv.altep = Yes (%)        0.267        
  time.iv (median [IQR])    0.596 nonnorm
  aspects (median [IQR])    0.075 nonnorm
  ia.occlus (%)             0.795        
     Intracranial ICA                    
     ICA with M1                         
     M1                                  
     M2                                  
     A1 or A2                            
  extra.ica = 1 (%)         0.179        
  time.rand (median [IQR])  0.251 nonnorm
```

1.6 Obtaining a more detailed Summary

If this was a real data set, we'd want to get a more detailed description of the data to make decisions about things like potentially collapsing categories of a variable, or whether or not a normal distribution was useful for a particular continuous variable, etc. You can do this with the `summary` command applied to a created Table 1, which shows, among other things, the effect of changing from normal to non-normal *p* values for continuous variables, and from approximate to "exact" *p* values for categorical factors.

Again, as noted above, in a real data situation, we'd want to plot the quantitative variables (within each group) to make a smart decision about whether a t test or Wilcoxon approach is more appropriate.

Note in the summary below that we have some missing values here. Often, we'll present this information within the Table 1, as well.

```r
summary(att2)
```

```

     ### Summary of continuous variables ###

trt: Intervention
            n miss p.miss mean sd median p25 p75 min max  skew  kurt
```

```
age       233    0    0.0   64 18     66  54  76  23  96 -0.34 -0.52
nihss     233    0    0.0   18  5     17  14  21  10  28  0.48 -0.74
sbp       233    0    0.0  146 26    146 129 164  78 214 -0.07 -0.22
time.iv   233   30   12.9   98 45     85  67 110  42 218  1.03  0.08
aspects   233    0    0.0    8  2      9   7  10   5  10 -0.56 -0.98
time.rand 233    2    0.9  203 57    204 152 250 100 300  0.01 -1.16
--------------------------------------------------------
trt: Control
            n miss p.miss mean sd median p25 p75 min max   skew  kurt
age       267    0    0.0   65 16     66  56  76  24  94 -0.296 -0.28
nihss     267    0    0.0   18  4     18  14  22  11  25  0.017 -1.24
sbp       267    1    0.4  145 24    145 128 161  82 231  0.156  0.08
time.iv   267   25    9.4   88 26     87  65 116  44 130  0.001 -1.32
aspects   267    4    1.5    9  1      9   8  10   5  10 -1.071  0.36
time.rand 267    0    0.0  214 70    196 149 266 120 360  0.508 -0.93

p-values
             pNormal pNonNormal
age       0.342813660 0.57856976
nihss     0.787487252 0.45311695
sbp       0.647157646 0.51346132
time.iv   0.003073372 0.59641104
aspects   0.032662901 0.07464683
time.rand 0.050803672 0.25134327

Standardize mean differences
              1 vs 2
age       0.08478764
nihss     0.02405390
sbp       0.04100833
time.iv   0.27691223
aspects   0.19210662
time.rand 0.17720957

=======================================================================================

     ### Summary of categorical variables ###

trt: Intervention
       var   n miss p.miss           level freq percent cum.percent
       sex 233    0    0.0          Female   98    42.1        42.1
                                      Male  135    57.9       100.0

  location 233    0    0.0            Left  116    49.8        49.8
                                     Right  117    50.2       100.0

   hx.isch 233    0    0.0              No  204    87.6        87.6
                                       Yes   29    12.4       100.0

      afib 233    0    0.0               0  167    71.7        71.7
                                         1   66    28.3       100.0

        dm 233    0    0.0               0  204    87.6        87.6
                                         1   29    12.4       100.0
   mrankin 233    0    0.0                0  190    81.5        81.5
                                         1   21     9.0        90.6
                                         2   12     5.2        95.7
                                       > 2   10     4.3       100.0

  iv.altep 233    0    0.0               No   30    12.9        12.9
                                       Yes  203    87.1       100.0

ia.occlus 233    0    0.0 Intracranial ICA    1     0.4         0.4
                               ICA with M1   59    25.3        25.8
                                        M1  154    66.1        91.8
                                        M2   18     7.7        99.6
                                  A1 or A2    1     0.4       100.0

extra.ica 233    0    0.0                0  158    67.8        67.8
                                         1   75    32.2       100.0

--------------------------------------------------------
trt: Control
       var   n miss p.miss            level freq percent cum.percent
       sex 267    0    0.0           Female  110    41.2        41.2
                                      Male  157    58.8       100.0

  location 267    0    0.0             Left  153    57.3        57.3
                                     Right  114    42.7       100.0

   hx.isch 267    0    0.0               No  242    90.6        90.6
                                       Yes   25     9.4       100.0

      afib 267    0    0.0                0  198    74.2        74.2
                                         1   69    25.8       100.0

        dm 267    0    0.0                0  233    87.3        87.3
                                         1   34    12.7       100.0

   mrankin 267    0    0.0                0  214    80.1        80.1
                                         1   29    10.9        91.0
                                         2   13     4.9        95.9
                                       > 2   11     4.1       100.0

  iv.altep 267    0    0.0               No   25     9.4         9.4
                                       Yes  242    90.6       100.0

ia.occlus 267    1    0.4 Intracranial ICA    3     1.1         1.1
                               ICA with M1   75    28.2        29.3
                                        M1  165    62.0        91.4
                                        M2   21     7.9        99.2
                                  A1 or A2    2     0.8       100.0

extra.ica 267    1    0.4                0  196    73.7        73.7
                                         1   70    26.3       100.0
```

```
p-values
            pApprox    pExact
sex       0.9171387 0.8561188
location  0.1113553 0.1056020
hx.isch   0.3352617 0.3124683
afib      0.6009691 0.5460206
dm        1.0000000 1.0000000
mrankin   0.9224798 0.9173657
iv.altep  0.2674968 0.2518374
ia.occlus 0.7945580 0.8189090
extra.ica 0.1793385 0.1667574

Standardize mean differences
               1 vs 2
sex       0.017479025
location  0.151168444
hx.isch   0.099032275
afib      0.055906317
dm        0.008673478
mrankin   0.062543164
iv.altep  0.111897009
ia.occlus 0.117394890
extra.ica 0.129370206
```

In this case, I have simulated the data to mirror the results in the published Table 1 for this study. In no way have I captured the full range of the real data, or any of the relationships in that data, so it's more important here to see what's available in the analysis, rather than to interpret it closely in the clinical context.

1.7 Exporting the Completed Table 1 from R to Excel or Word

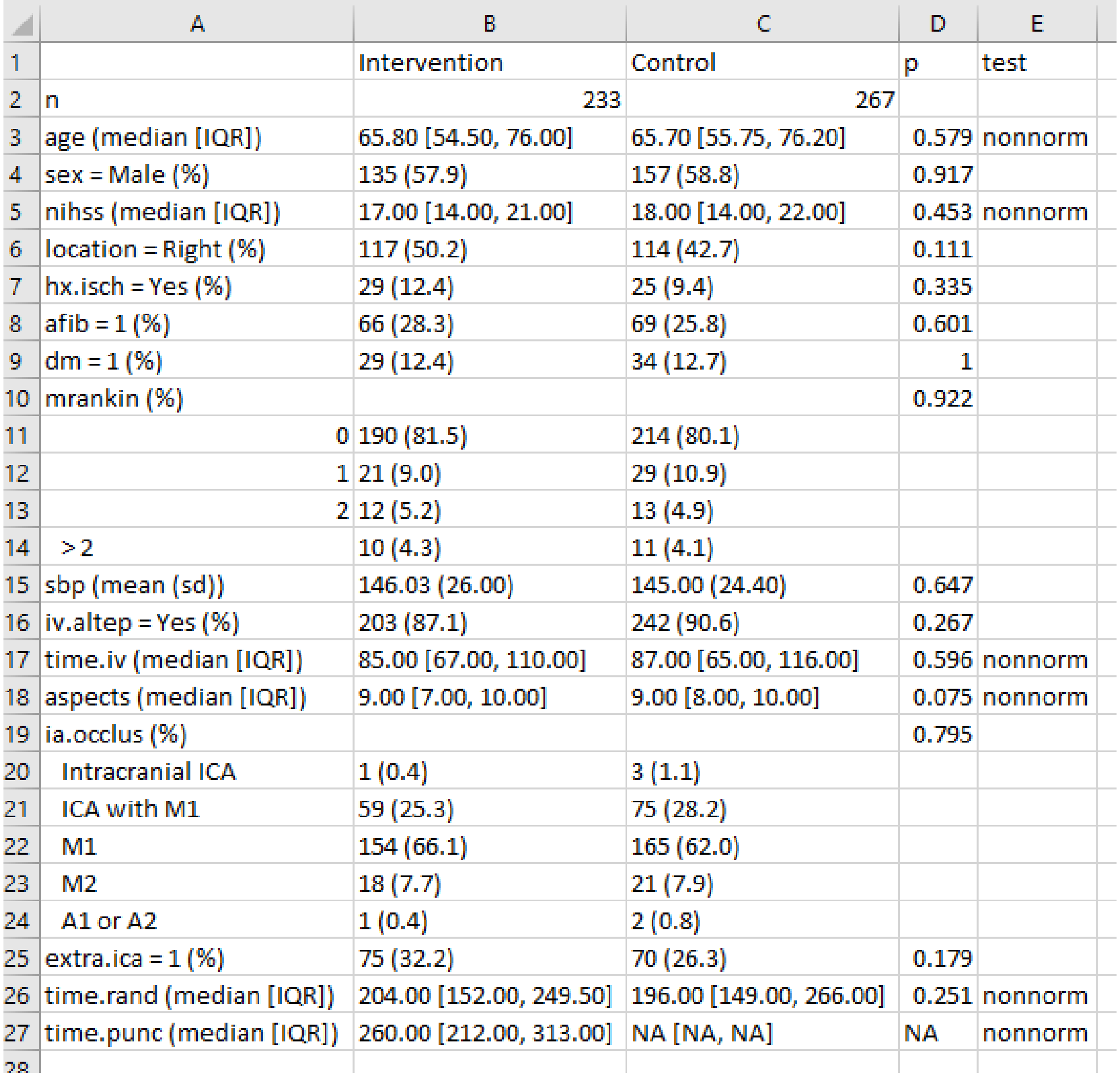
Once you've built the table and are generally satisfied with it, you'll probably want to be able to drop it into Excel or Word for final cleanup.

1.7.1 Approach A: Save and open in Excel

One option is to **save the Table 1** to a `.csv` file within our `data` subfolder (note that the `data` folder must already exist), which you can then open directly in Excel. This is the approach I generally use. Note the addition of some `quote`, `noSpaces` and `printToggle` selections here.

```
fs.table1save <- print(att2,
      nonnormal = c("age", "nihss", "time.iv", "aspects", "time.rand"),
      exact = c("location", "mrankin"),
      quote = FALSE, noSpaces = TRUE, printToggle = FALSE)

write.csv(fs.table1save, file = "data/fs-table1.csv")
```

When I then open the `fs-table1.csv` file in Excel, it looks like this:



	A	B	C	D	E
1		Intervention	Control	p	test
2	n	233	267		
3	age (median [IQR])	65.80 [54.50, 76.00]	65.70 [55.75, 76.20]	0.579	nonnorm
4	sex = Male (%)	135 (57.9)	157 (58.8)	0.917	
5	nihss (median [IQR])	17.00 [14.00, 21.00]	18.00 [14.00, 22.00]	0.453	nonnorm
6	location = Right (%)	117 (50.2)	114 (42.7)	0.111	
7	hx.isch = Yes (%)	29 (12.4)	25 (9.4)	0.335	
8	afib = 1 (%)	66 (28.3)	69 (25.8)	0.601	
9	dm = 1 (%)	29 (12.4)	34 (12.7)	1	
10	mrankin (%)			0.922	
11	0	190 (81.5)	214 (80.1)		
12	1	21 (9.0)	29 (10.9)		
13	2	12 (5.2)	13 (4.9)		
14	> 2	10 (4.3)	11 (4.1)		
15	sbp (mean (sd))	146.03 (26.00)	145.00 (24.40)	0.647	
16	iv.altep = Yes (%)	203 (87.1)	242 (90.6)	0.267	
17	time.iv (median [IQR])	85.00 [67.00, 110.00]	87.00 [65.00, 116.00]	0.596	nonnorm
18	aspects (median [IQR])	9.00 [7.00, 10.00]	9.00 [8.00, 10.00]	0.075	nonnorm
19	ia.occlus (%)			0.795	
20	Intracranial ICA	1 (0.4)	3 (1.1)		
21	ICA with M1	59 (25.3)	75 (28.2)		
22	M1	154 (66.1)	165 (62.0)		
23	M2	18 (7.7)	21 (7.9)		
24	A1 or A2	1 (0.4)	2 (0.8)		
25	extra.ica = 1 (%)	75 (32.2)	70 (26.3)	0.179	
26	time.rand (median [IQR])	204.00 [152.00, 249.50]	196.00 [149.00, 266.00]	0.251	nonnorm
27	time.punc (median [IQR])	260.00 [212.00, 313.00]	NA [NA, NA]	NA	nonnorm
28					

And from here, I can either drop it directly into Word, or present it as is, or start tweaking it to meet formatting needs.

1.7.2 Approach B: Produce the Table so you can cut and paste it

```
print(att2,
      nonnormal = c("age", "nihss", "time.iv", "aspects", "time.rand"),
      exact = c("location", "mrankin"),
      quote = TRUE, noSpaces = TRUE)
```

This will look like a mess by itself, but if you:

1. copy and paste that mess into Excel
2. select Text to Columns from the Data menu
3. select Delimited, then Space and select Treat consecutive delimiters as one

you should get something usable again.

Or, in Word,

1. insert the text

2. select the text with your mouse
3. select Insert … Table … Convert Text to Table
4. place a quotation mark in the "Other" area under Separate text at …

After dropping blank columns, the result looks pretty good.

1.8 A Controlled Biological Experiment - The Blood-Brain Barrier

My source for the data and the following explanatory paragraph is page 307 from Ramsey and Schafer (2002). The original data come from Barnett et al. (1995).

> The human brain (and that of rats, coincidentally) is protected from the bacteria and toxins that course through the bloodstream by something called the blood-brain barrier. After a method of disrupting the barrier was developed, researchers tested this new mechanism, as follows. A series of 34 rats were inoculated with human lung cancer cells to induce brain tumors. After 9-11 days they were infused with either the barrier disruption (BD) solution or, as a control, a normal saline (NS) solution. Fifteen minutes later, the rats received a standard dose of a particular therapeutic antibody (L6-F(ab')2. The key measure of the effectiveness of transmission across the brain-blood barrier is the ratio of the antibody concentration in the brain tumor to the antibody concentration in normal tissue outside the brain. The rats were then sacrificed, and the amounts of antibody in the brain tumor and in normal tissue from the liver were measured. The study's primary objective is to determine whether the antibody concentration in the tumor increased when the blood-barrier disruption infusion was given, and if so, by how much?

1.9 The `bloodbrain.csv` file

Consider the data, available on the Data and Code page of our course website in the `bloodbrain.csv` file, which includes the following variables:

Variable	Description
`case`	identification number for the rat (1 - 34)
`brain`	an outcome: Brain tumor antibody count (per gram)
`liver`	an outcome: Liver antibody count (per gram)
`tlratio`	an outcome: tumor / liver concentration ratio
`solution`	the treatment: BD (barrier disruption) or NS (normal saline)
`sactime`	a design variable: Sacrifice time (hours; either 0.5, 3, 24 or 72)
`postin`	covariate: Days post-inoculation of lung cancer cells (9, 10 or 11)
`sex`	covariate: M or F
`wt.init`	covariate: Initial weight (grams)
`wt.loss`	covariate: Weight loss (grams)
`wt.tumor`	covariate: Tumor weight (10^{-4} grams)

And here's what the data look like in R.

```
bloodbrain
```

```
# A tibble: 34 x 11
    case  brain   liver tlratio solution sactime postin sex   wt.init
   <int>  <int>   <int>   <dbl> <fct>      <dbl>  <int> <fct>   <int>
 1     1  41081 1456164  0.0282 BD         0.500     10 F         239
 2     2  44286 1602171  0.0276 BD         0.500     10 F         225
 3     3 102926 1601936  0.0642 BD         0.500     10 F         224
 4     4  25927 1776411  0.0146 BD         0.500     10 F         184
 5     5  42643 1351184  0.0316 BD         0.500     10 F         250
 6     6  31342 1790863  0.0175 NS         0.500     10 F         196
 7     7  22815 1633386  0.0140 NS         0.500     10 F         200
 8     8  16629 1618757  0.0103 NS         0.500     10 F         273
 9     9  22315 1567602  0.0142 NS         0.500     10 F         216
10    10  77961 1060057  0.0735 BD         3.00      10 F         267
# ... with 24 more rows, and 2 more variables: wt.loss <dbl>, wt.tumor
#   <int>
```

1.10 A Table 1 for bloodbrain

Barnett et al. (1995) did not provide a Table 1 for these data, so let's build one to compare the two `solutions` (`BD` vs. `NS`) on the covariates and outcomes, plus the natural logarithm of the tumor/liver concentration ratio (`tlratio`). We'll opt to treat the sacrifice time (`sactime`) and the days post-inoculation of lung cancer cells (`postin`) as categorical rather than quantitative variables.

```
bloodbrain <- bloodbrain %>%
    mutate(logTL = log(tlratio))

dput(names(bloodbrain))
```

```
c("case", "brain", "liver", "tlratio", "solution", "sactime",
"postin", "sex", "wt.init", "wt.loss", "wt.tumor", "logTL")
```

OK - there's the list of variables we'll need. I'll put the outcomes at the bottom of the table.

```
bb.vars <- c("sactime", "postin", "sex", "wt.init", "wt.loss",
             "wt.tumor", "brain", "liver", "tlratio", "logTL")

bb.factors <- c("sactime", "sex", "postin")

bb.att1 <- CreateTableOne(data = bloodbrain,
                          vars = bb.vars,
                          factorVars = bb.factors,
                          strata = c("solution"))
summary(bb.att1)
```

```
     ### Summary of continuous variables ###

solution: BD
          n miss p.miss    mean    sd median    p25   p75    min   max
wt.init  17    0      0     243 3e+01  2e+02  2e+02 3e+02  2e+02 3e+02
wt.loss  17    0      0       3 5e+00  4e+00  1e+00 6e+00 -5e+00 1e+01
wt.tumor 17    0      0     157 8e+01  2e+02  1e+02 2e+02  2e+01 4e+02
brain    17    0      0   56043 3e+04  5e+04  4e+04 8e+04  6e+03 1e+05
liver    17    0      0 672577 7e+05  6e+05  2e+04 1e+06  2e+03 2e+06
tlratio  17    0      0       2 3e+00  1e-01  6e-02 3e+00  1e-02 9e+00
logTL    17    0      0      -1 2e+00 -2e+00 -3e+00 1e+00 -4e+00 2e+00
          skew kurt
wt.init  -0.39  0.7
wt.loss  -0.10  0.2
wt.tumor  0.53  1.0
brain     0.29 -0.6
liver     0.35 -1.7
tlratio   1.58  1.7
logTL     0.08 -1.7
--------------------------------------------------------
solution: NS
          n miss p.miss   mean    sd median    p25    p75    min   max
wt.init  17    0      0    240 3e+01  2e+02  2e+02  3e+02  2e+02 3e+02
wt.loss  17    0      0      4 4e+00  3e+00  2e+00  7e+00 -4e+00 1e+01
wt.tumor 17    0      0    209 1e+02  2e+02  2e+02  3e+02  3e+01 5e+02
brain    17    0      0  23887 1e+04  2e+04  1e+04  3e+04  1e+03 5e+04
liver    17    0      0 664975 7e+05  7e+05  2e+04  1e+06  9e+02 2e+06
tlratio  17    0      0      1 2e+00  5e-02  3e-02  9e-01  1e-02 7e+00
logTL    17    0      0     -2 2e+00 -3e+00 -3e+00 -7e-02 -5e+00 2e+00
          skew  kurt
wt.init   0.33 -0.48
wt.loss  -0.09  0.08
wt.tumor  0.63  0.77
brain     0.30 -0.35
liver     0.40 -1.56
tlratio   2.27  4.84
logTL     0.27 -1.61

p-values
             pNormal  pNonNormal
wt.init  0.807308940 0.641940278
wt.loss  0.683756156 0.876749808
wt.tumor 0.151510151 0.190482094
brain    0.001027678 0.002579901
liver    0.974853609 0.904045603
tlratio  0.320501715 0.221425879
logTL    0.351633525 0.221425879

Standardize mean differences
             1 vs 2
wt.init  0.08435244
wt.loss  0.14099823
wt.tumor 0.50397184
brain    1.23884159
liver    0.01089667
tlratio  0.34611465
logTL    0.32420504

=======================================================================================

     ### Summary of categorical variables ###

solution: BD
     var  n miss p.miss level freq percent cum.percent
 sactime 17    0    0.0   0.5    5    29.4        29.4
                            3    4    23.5        52.9
                           24    4    23.5        76.5
                           72    4    23.5       100.0
```

```
  postin 17   0    0.0     9    1     5.9         5.9
                          10   14    82.4        88.2
                          11    2    11.8       100.0

     sex 17   0    0.0     F   13    76.5        76.5
                           M    4    23.5       100.0

--------------------------------------------------------
solution: NS
     var  n miss p.miss level freq percent cum.percent
 sactime 17   0    0.0   0.5    4    23.5        23.5
                           3    5    29.4        52.9
                          24    4    23.5        76.5
                          72    4    23.5       100.0

  postin 17   0    0.0     9    2    11.8        11.8
                          10   13    76.5        88.2
                          11    2    11.8       100.0

     sex 17   0    0.0     F   13    76.5        76.5
                           M    4    23.5       100.0


p-values
          pApprox pExact
sactime 0.9739246      1
postin  0.8309504      1
sex     1.0000000      1

Standardize mean differences
           1 vs 2
sactime 0.1622214
postin  0.2098877
sex     0.0000000
```

Note that, in this particular case, the decisions we make about normality vs. non-normality (for quantitative variables) and the decisions we make about approximate vs. exact testing (for categorical variables) won't actually change the implications of the *p* values. Each approach gives similar results for each variable. Of course, that's not always true.

1.10.1 Generate final Table 1 for `bloodbrain`

I'll choose to treat `tlratio` and its logarithm as non-Normal, but otherwise, use t tests, but admittedly, that's an arbitrary decision, really.

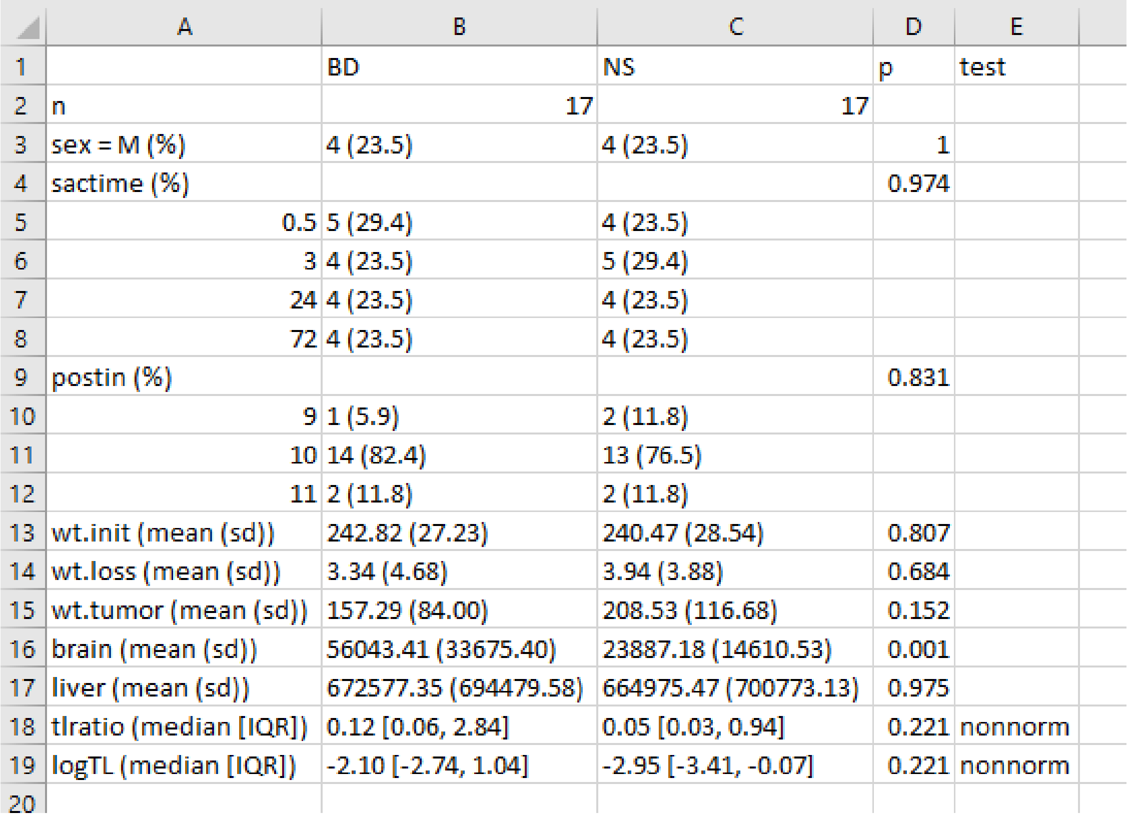
```
print(bb.att1, nonnormal = c("tlratio", "logTL"))
```

```
                        Stratified by solution
                         BD                       NS
  n                             17                       17
  sactime (%)
     0.5                         5 (29.4)                 4 (23.5)
     3                           4 (23.5)                 5 (29.4)
     24                          4 (23.5)                 4 (23.5)
     72                            4 (23.5)                4 (23.5)
  postin (%)
     9                             1 ( 5.9)                2 (11.8)
     10                           14 (82.4)               13 (76.5)
     11                            2 (11.8)                2 (11.8)
  sex = M (%)                      4 (23.5)                4 (23.5)
  wt.init (mean (sd))         242.82 (27.23)          240.47 (28.54)
  wt.loss (mean (sd))           3.34 (4.68)             3.94 (3.88)
  wt.tumor (mean (sd))        157.29 (84.00)          208.53 (116.68)
  brain (mean (sd))         56043.41 (33675.40)     23887.18 (14610.53)
  liver (mean (sd))        672577.35 (694479.58)   664975.47 (700773.13)
  tlratio (median [IQR])        0.12 [0.06, 2.84]       0.05 [0.03, 0.94]
  logTL (median [IQR])         -2.10 [-2.74, 1.04]     -2.95 [-3.41, -0.07]
                        Stratified by solution
                         p      test
  n
  sactime (%)             0.974
     0.5
     3
     24
     72
  postin (%)              0.831
     9
     10
     11
  sex = M (%)             1.000
  wt.init (mean (sd))     0.807
  wt.loss (mean (sd))     0.684
  wt.tumor (mean (sd))    0.152
  brain (mean (sd))       0.001
  liver (mean (sd))       0.975
  tlratio (median [IQR])  0.221 nonnorm
  logTL (median [IQR])    0.221 nonnorm
```

Or, we can get an Excel-readable version placed in a `data` subfolder, using

```
bb.t1 <- print(bb.att1, nonnormal = c("tlratio", "logTL"), quote = FALSE,
               noSpaces = TRUE, printToggle = FALSE)

write.csv(bb.t1, file = "data/bb-table1.csv")
```

which, when dropped into Excel, will look like this:



	A	B	C	D	E
1		BD	NS	p	test
2	n	17	17		
3	sex = M (%)	4 (23.5)	4 (23.5)	1	
4	sactime (%)			0.974	
5	0.5	5 (29.4)	4 (23.5)		
6	3	4 (23.5)	5 (29.4)		
7	24	4 (23.5)	4 (23.5)		
8	72	4 (23.5)	4 (23.5)		
9	postin (%)			0.831	
10	9	1 (5.9)	2 (11.8)		
11	10	14 (82.4)	13 (76.5)		
12	11	2 (11.8)	2 (11.8)		
13	wt.init (mean (sd))	242.82 (27.23)	240.47 (28.54)	0.807	
14	wt.loss (mean (sd))	3.34 (4.68)	3.94 (3.88)	0.684	
15	wt.tumor (mean (sd))	157.29 (84.00)	208.53 (116.68)	0.152	
16	brain (mean (sd))	56043.41 (33675.40)	23887.18 (14610.53)	0.001	
17	liver (mean (sd))	672577.35 (694479.58)	664975.47 (700773.13)	0.975	
18	tlratio (median [IQR])	0.12 [0.06, 2.84]	0.05 [0.03, 0.94]	0.221	nonnorm
19	logTL (median [IQR])	-2.10 [-2.74, 1.04]	-2.95 [-3.41, -0.07]	0.221	nonnorm
20					

One thing I would definitely clean up here, in practice, is to change the presentation of the *p* value for `sex` from 1 to > 0.99, or just omit it altogether. I'd also drop the `computer-ese` where possible, add units for the measures, round **a lot**, identify the outcomes carefully, and use notes to indicate deviations from the main approach.

1.10.2 A More Finished Version (after Cleanup in Word)

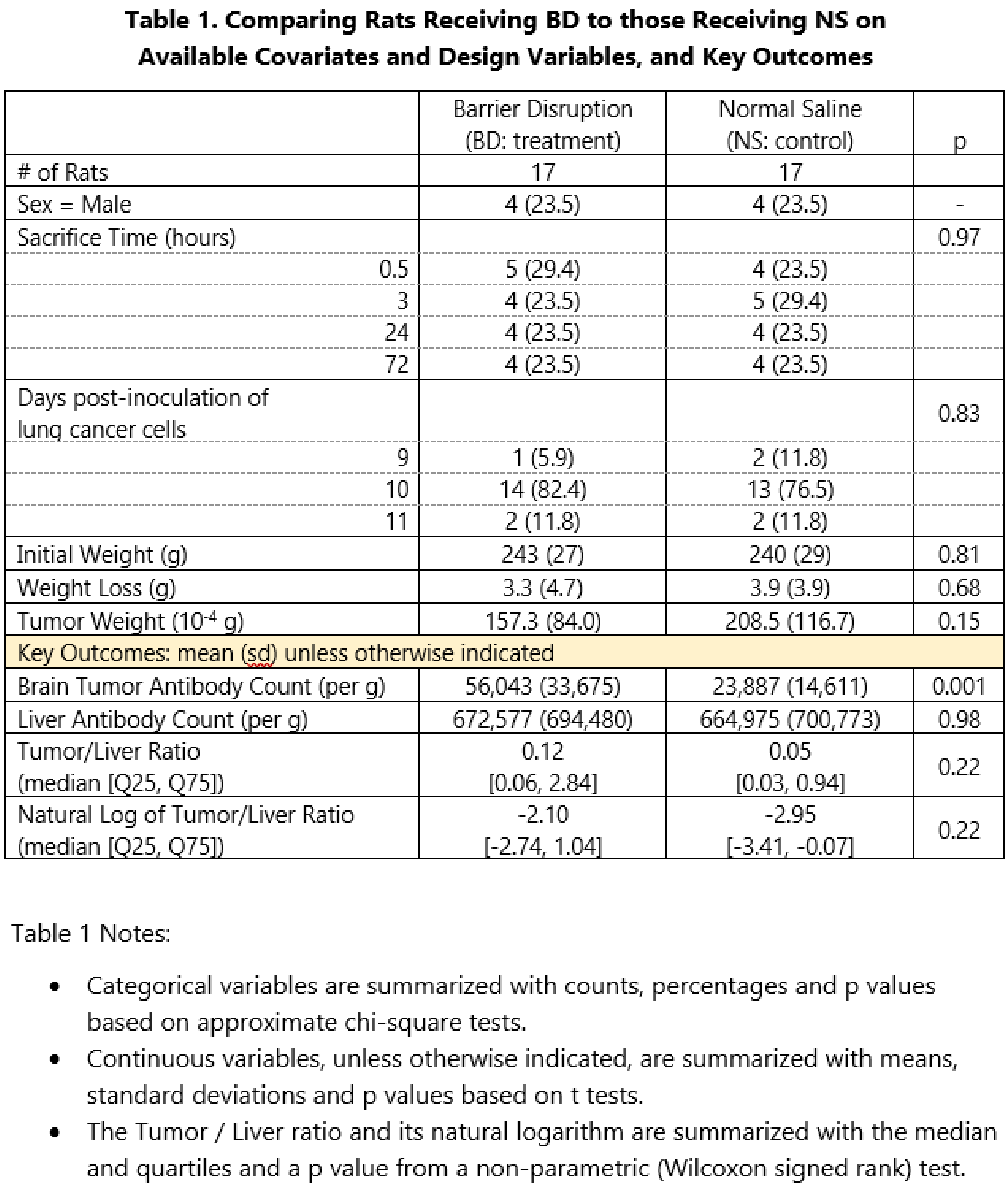


Table 1. Comparing Rats Receiving BD to those Receiving NS on Available Covariates and Design Variables, and Key Outcomes

	Barrier Disruption (BD: treatment)	Normal Saline (NS: control)	p
# of Rats	17	17	
Sex = Male	4 (23.5)	4 (23.5)	-
Sacrifice Time (hours)			0.97
0.5	5 (29.4)	4 (23.5)	
3	4 (23.5)	5 (29.4)	
24	4 (23.5)	4 (23.5)	
72	4 (23.5)	4 (23.5)	
Days post-inoculation of lung cancer cells			0.83
9	1 (5.9)	2 (11.8)	
10	14 (82.4)	13 (76.5)	
11	2 (11.8)	2 (11.8)	
Initial Weight (g)	243 (27)	240 (29)	0.81
Weight Loss (g)	3.3 (4.7)	3.9 (3.9)	0.68
Tumor Weight (10^{-4} g)	157.3 (84.0)	208.5 (116.7)	0.15
Key Outcomes: mean (sd) unless otherwise indicated			
Brain Tumor Antibody Count (per g)	56,043 (33,675)	23,887 (14,611)	0.001
Liver Antibody Count (per g)	672,577 (694,480)	664,975 (700,773)	0.98
Tumor/Liver Ratio (median [Q25, Q75])	0.12 [0.06, 2.84]	0.05 [0.03, 0.94]	0.22
Natural Log of Tumor/Liver Ratio (median [Q25, Q75])	-2.10 [-2.74, 1.04]	-2.95 [-3.41, -0.07]	0.22

Table 1 Notes:

- Categorical variables are summarized with counts, percentages and p values based on approximate chi-square tests.
- Continuous variables, unless otherwise indicated, are summarized with means, standard deviations and p values based on t tests.
- The Tumor / Liver ratio and its natural logarithm are summarized with the median and quartiles and a p value from a non-parametric (Wilcoxon signed rank) test.

Chapter 2

Linear Regression on a small SMART data set

2.1 BRFSS and SMART

The Centers for Disease Control analyzes Behavioral Risk Factor Surveillance System (BRFSS) survey data for specific metropolitan and micropolitan statistical areas (MMSAs) in a program called the Selected Metropolitan/Micropolitan Area Risk Trends of BRFSS (SMART BRFSS.)

In this work, we will focus on data from the 2016 SMART, and in particular on data from the Cleveland-Elyria, OH, Metropolitan Statistical Area. The purpose of this survey is to provide localized health information that can help public health practitioners identify local emerging health problems, plan and evaluate local responses, and efficiently allocate resources to specific needs.

2.1.1 Key resources

- the full data are available in the form of the 2016 SMART BRFSS MMSA Data, found in a zipped SAS Transport Format file. The data were released in August 2017.
- the MMSA Variable Layout PDF which simply lists the variables included in the data file
- the Calculated Variables PDF which describes the risk factors by data variable names - there is also an online summary matrix of these calculated variables, as well.
- the lengthy 2016 Survey Questions PDF which lists all questions asked as part of the BRFSS in 2016
- the enormous Codebook for the 2016 BRFSS Survey PDF which identifies the variables by name for us.

Later this term, we'll use all of those resources to help construct a more complete data set than we'll study today. I'll also demonstrate how I built the `smartcle1` data set that we'll use in this Chapter.

2.2 The `smartcle1` data: Cookbook

The `smartcle1.csv` data file available on the Data and Code page of our website describes information on 11 variables for 1036 respondents to the BRFSS 2016, who live in the Cleveland-Elyria, OH, Metropolitan Statistical Area. The variables in the `smartcle1.csv` file are listed below, along with (in some cases) the BRFSS items that generate these responses.

Variable	Description
`SEQNO`	respondent identification number (all begin with 2016)

Variable	Description
`physhealth`	Now thinking about your physical health, which includes physical illness and injury, for how many days during the past 30 days was your physical health not good?
`menthealth`	Now thinking about your mental health, which includes stress, depression, and problems with emotions, for how many days during the past 30 days was your mental health not good?
`poorhealth`	During the past 30 days, for about how many days did poor physical or mental health keep you from doing your usual activities, such as self-care, work, or recreation?
`genhealth`	Would you say that in general, your health is … (five categories: Excellent, Very Good, Good, Fair or Poor)
`bmi`	Body mass index, in kg/m^2
`female`	Sex, 1 = female, 0 = male
`internet30`	Have you used the internet in the past 30 days? (1 = yes, 0 = no)
`exerany`	During the past month, other than your regular job, did you participate in any physical activities or exercises such as running, calisthenics, golf, gardening, or walking for exercise? (1 = yes, 0 = no)
`sleephrs`	On average, how many hours of sleep do you get in a 24-hour period?
`alcdays`	How many days during the past 30 days did you have at least one drink of any alcoholic beverage such as beer, wine, a malt beverage or liquor?

```r
str(smartcle1)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame':   1036 obs. of  11 variables:
 $ SEQNO     : num  2.02e+09 2.02e+09 2.02e+09 2.02e+09 2.02e+09 ...
 $ physhealth: int  0 0 1 0 5 4 2 2 0 0 ...
 $ menthealth: int  0 0 5 0 0 18 0 3 0 0 ...
 $ poorhealth: int  NA NA 0 NA 0 6 0 0 NA NA ...
 $ genhealth : Factor w/ 5 levels "1_Excellent",..: 2 1 2 3 1 2 3 3 2 3 ...
 $ bmi       : num  26.7 23.7 26.9 21.7 24.1 ...
 $ female    : int  1 0 0 1 0 0 1 1 0 0 ...
 $ internet30: int  1 1 1 1 1 1 1 1 1 1 ...
 $ exerany   : int  1 1 0 1 1 1 1 1 1 0 ...
 $ sleephrs  : int  6 6 8 9 7 5 9 7 7 7 ...
 $ alcdays   : int  1 4 4 3 2 28 4 2 4 25 ...
```

2.3 smartcle2: Omitting Missing Observations: Complete-Case Analyses

For the purpose of fitting our first few models, we will eliminate the missingness problem, and look only at the *complete cases* in our `smartcle1` data. We will discuss methods for imputing missing data later in these Notes.

To inspect the missingness in our data, we might consider using the `skim` function from the `skimr` package. We'll exclude the respondent identifier code (`SEQNO`) from this summary as uninteresting.

```r
skim_with(numeric = list(hist = NULL), integer = list(hist = NULL))
## above line eliminates the sparkline histograms
## it can be commented out when working in the console,
## but I need it to produce the Notes without errors right now
```

```
smartcle1 %>%
    skim(-SEQNO)
```

```
Skim summary statistics
 n obs: 1036
 n variables: 11

Variable type: factor
  variable missing complete    n n_unique
 genhealth       3     1033 1036        5
                             top_counts ordered
 2_V: 350, 3_G: 344, 1_E: 173, 4_F: 122   FALSE

Variable type: integer
   variable missing complete    n mean   sd p0 p25 median p75 p100
    alcdays      46      990 1036 4.65 8.05  0   0      1   4   30
    exerany       3     1033 1036 0.76 0.43  0   1      1   1    1
     female       0     1036 1036 0.6  0.49  0   0      1   1    1
 internet30       6     1030 1036 0.81 0.39  0   1      1   1    1
 menthealth      11     1025 1036 2.72 6.82  0   0      0   2   30
 physhealth      17     1019 1036 3.97 8.67  0   0      0   2   30
 poorhealth     543      493 1036 4.07 8.09  0   0      0   3   30
   sleephrs       8     1028 1036 7.02 1.53  1   6      7   8   20

Variable type: numeric
 variable missing complete    n  mean   sd    p0  p25 median   p75  p100
      bmi      84      952 1036 27.89 6.47 12.71 23.7  26.68 30.53 66.06
```

Now, we'll create a new tibble called `smartcle2` which contains every variable except `poorhealth`, and which includes all respondents with complete data on the variables (other than `poorhealth`). We'll store those observations with complete data in the `smartcle2` tibble.

```
smartcle2 <- smartcle1 %>%
    select(-poorhealth) %>%
    filter(complete.cases(.))

smartcle2
```

```
# A tibble: 896 x 10
     SEQNO physhealth menthealth genhealth   bmi female internet30 exerany
     <dbl>      <int>      <int> <fct>     <dbl>  <int>      <int>   <int>
 1  2.02e9          0          0 2_VeryGo~  26.7      1          1       1
 2  2.02e9          0          0 1_Excell~  23.7      0          1       1
 3  2.02e9          1          5 2_VeryGo~  26.9      0          1       0
 4  2.02e9          0          0 3_Good     21.7      1          1       1
 5  2.02e9          5          0 1_Excell~  24.1      0          1       1
 6  2.02e9          4         18 2_VeryGo~  27.6      0          1       1
 7  2.02e9          2          0 3_Good     25.7      1          1       1
 8  2.02e9          2          3 3_Good     28.5      1          1       1
 9  2.02e9          0          0 2_VeryGo~  28.6      0          1       1
10  2.02e9          0          0 3_Good     23.1      0          1       0
# ... with 886 more rows, and 2 more variables: sleephrs <int>, alcdays
#   <int>
```

Note that there are only 896 respondents with **complete** data on the 10 variables (excluding `poorhealth`) in the `smartcle2` tibble, as compared to our original `smartcle1` data which described 1036 respondents and

11 variables, but with lots of missing data.

2.4 Summarizing the `smartcle2` data numerically

2.4.1 The New Toy: The `skim` function

```
skim(smartcle2, -SEQNO)
```

```
Skim summary statistics
 n obs: 896 
 n variables: 10 

Variable type: factor 
  variable missing complete   n n_unique
 genhealth       0      896 896        5
                             top_counts ordered
 2_V: 306, 3_G: 295, 1_E: 155, 4_F: 102   FALSE

Variable type: integer 
   variable missing complete   n mean   sd p0 p25 median p75 p100
    alcdays       0      896 896 4.83 8.14  0   0      1   5   30
    exerany       0      896 896 0.77 0.42  0   1      1   1    1
     female       0      896 896 0.58 0.49  0   0      1   1    1
 internet30       0      896 896 0.81 0.39  0   1      1   1    1
 menthealth       0      896 896 2.69 6.72  0   0      0   2   30
 physhealth       0      896 896 3.99 8.64  0   0      0   2   30
   sleephrs       0      896 896 7.02 1.48  1   6      7   8   20

Variable type: numeric 
 variable missing complete   n  mean   sd    p0  p25 median   p75  p100
      bmi       0      896 896 27.87 6.33 12.71 23.7   26.8 30.53 66.06
```

2.4.2 The usual summary for a data frame

Of course, we can use the usual `summary` to get some basic information about the data.

```
summary(smartcle2)
```

```
     SEQNO             physhealth      menthealth           genhealth  
 Min.   :2.016e+09   Min.   : 0.00    Min.   : 0.000   1_Excellent:155  
 1st Qu.:2.016e+09   1st Qu.: 0.00    1st Qu.: 0.000   2_VeryGood :306  
 Median :2.016e+09   Median : 0.00    Median : 0.000   3_Good     :295  
 Mean   :2.016e+09   Mean   : 3.99    Mean   : 2.693   4_Fair     :102  
 3rd Qu.:2.016e+09   3rd Qu.: 2.00    3rd Qu.: 2.000   5_Poor     : 38  
 Max.   :2.016e+09   Max.   :30.00    Max.   :30.000                    
      bmi            female         internet30        exerany      
 Min.   :12.71   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:23.70   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:1.0000  
 Median :26.80   Median :1.0000   Median :1.0000   Median :1.0000  
 Mean   :27.87   Mean   :0.5848   Mean   :0.8147   Mean   :0.7667  
 3rd Qu.:30.53   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :66.06   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
```

```
    sleephrs         alcdays
 Min.   : 1.000   Min.   : 0.000
 1st Qu.: 6.000   1st Qu.: 0.000
 Median : 7.000   Median : 1.000
 Mean   : 7.022   Mean   : 4.834
 3rd Qu.: 8.000   3rd Qu.: 5.000
 Max.   :20.000   Max.   :30.000
```

2.4.3 The describe function in Hmisc

Or we can use the describe function from the Hmisc package.

```
Hmisc::describe(select(smartcle2, bmi, genhealth, female))
```

```
select(smartcle2, bmi, genhealth, female)

 3  Variables      896  Observations
---------------------------------------------------------------------------
bmi
       n  missing distinct     Info     Mean      Gmd      .05      .10
     896        0      467        1    27.87    6.572    20.06    21.23
     .25      .50      .75      .90      .95
   23.70    26.80    30.53    35.36    39.30

lowest : 12.71 13.34 14.72 16.22 17.30, highest: 56.89 57.04 60.95 61.84 66.06
---------------------------------------------------------------------------
genhealth
       n  missing distinct
     896        0        5

Value       1_Excellent  2_VeryGood      3_Good      4_Fair      5_Poor
Frequency           155         306         295         102          38
Proportion        0.173       0.342       0.329       0.114       0.042
---------------------------------------------------------------------------
female
       n  missing distinct     Info      Sum     Mean      Gmd
     896        0        2    0.728      524   0.5848   0.4862

---------------------------------------------------------------------------
```

2.5 Counting as exploratory data analysis

Counting things can be amazingly useful.

2.5.1 How many respondents had exercised in the past 30 days? Did this vary by sex?

```
smartcle2 %>% count(female, exerany) %>% mutate(percent = 100*n / sum(n))
```

```
# A tibble: 4 x 4
  female exerany     n percent
   <int>   <int> <int>   <dbl>
```

```
1      0       0    64    7.14
2      0       1   308   34.4
3      1       0   145   16.2
4      1       1   379   42.3
```

so we know now that 42.3% of the subjects in our data were women who exercised. Suppose that instead we want to find the percentage of exercisers within each sex…

```
smartcle2 %>%
    count(female, exerany) %>%
    group_by(female) %>%
    mutate(prob = 100*n / sum(n))
```

```
# A tibble: 4 x 4
# Groups: female [2]
  female exerany     n  prob
   <int>   <int> <int> <dbl>
1      0       0    64  17.2
2      0       1   308  82.8
3      1       0   145  27.7
4      1       1   379  72.3
```

and now we know that 82.8% of the males exercised at least once in the last 30 days, as compared to 72.3% of the females.

2.5.2 What's the distribution of `sleephrs`?

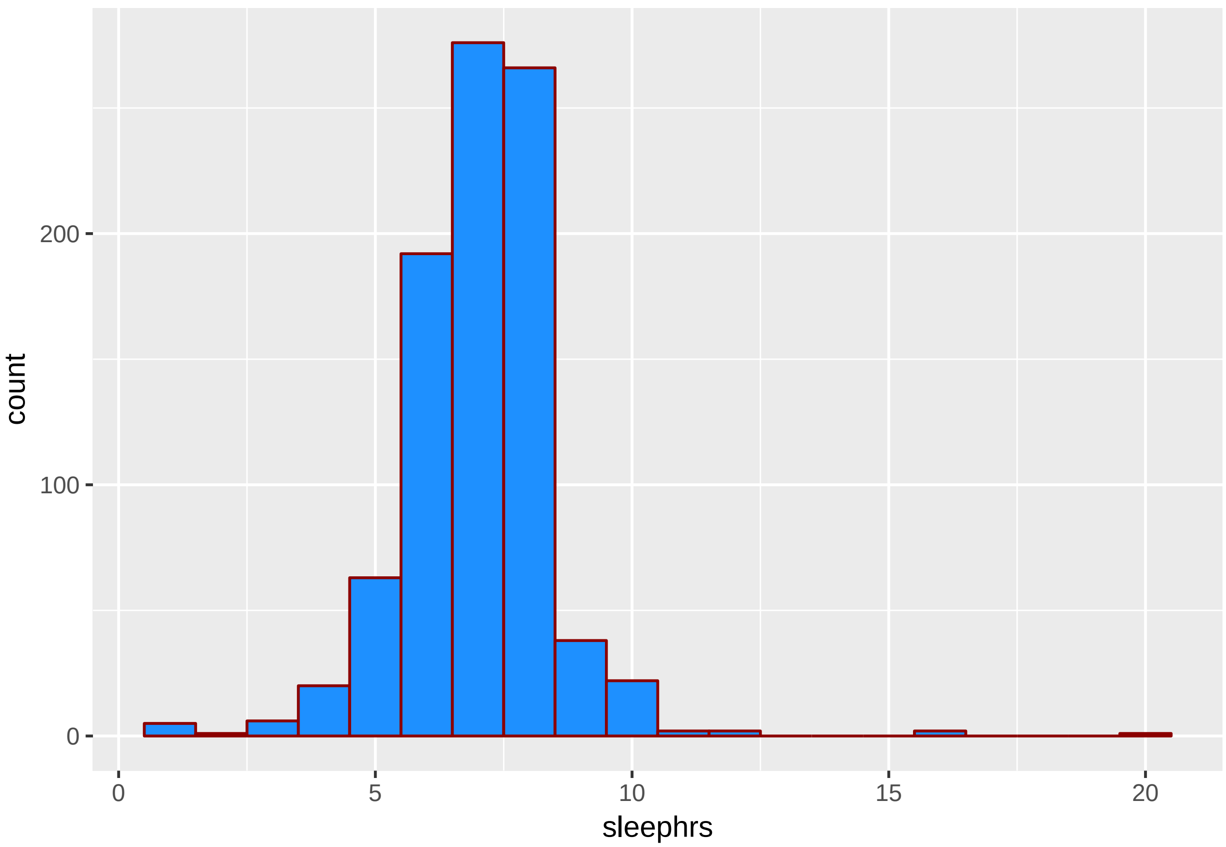
We can count quantitative variables with discrete sets of possible values, like `sleephrs`, which is captured as an integer (that must fall between 0 and 24.)

```
smartcle2 %>% count(sleephrs)
```

```
# A tibble: 14 x 2
   sleephrs     n
      <int> <int>
 1        1     5
 2        2     1
 3        3     6
 4        4    20
 5        5    63
 6        6   192
 7        7   276
 8        8   266
 9        9    38
10       10    22
11       11     2
12       12     2
13       16     2
14       20     1
```

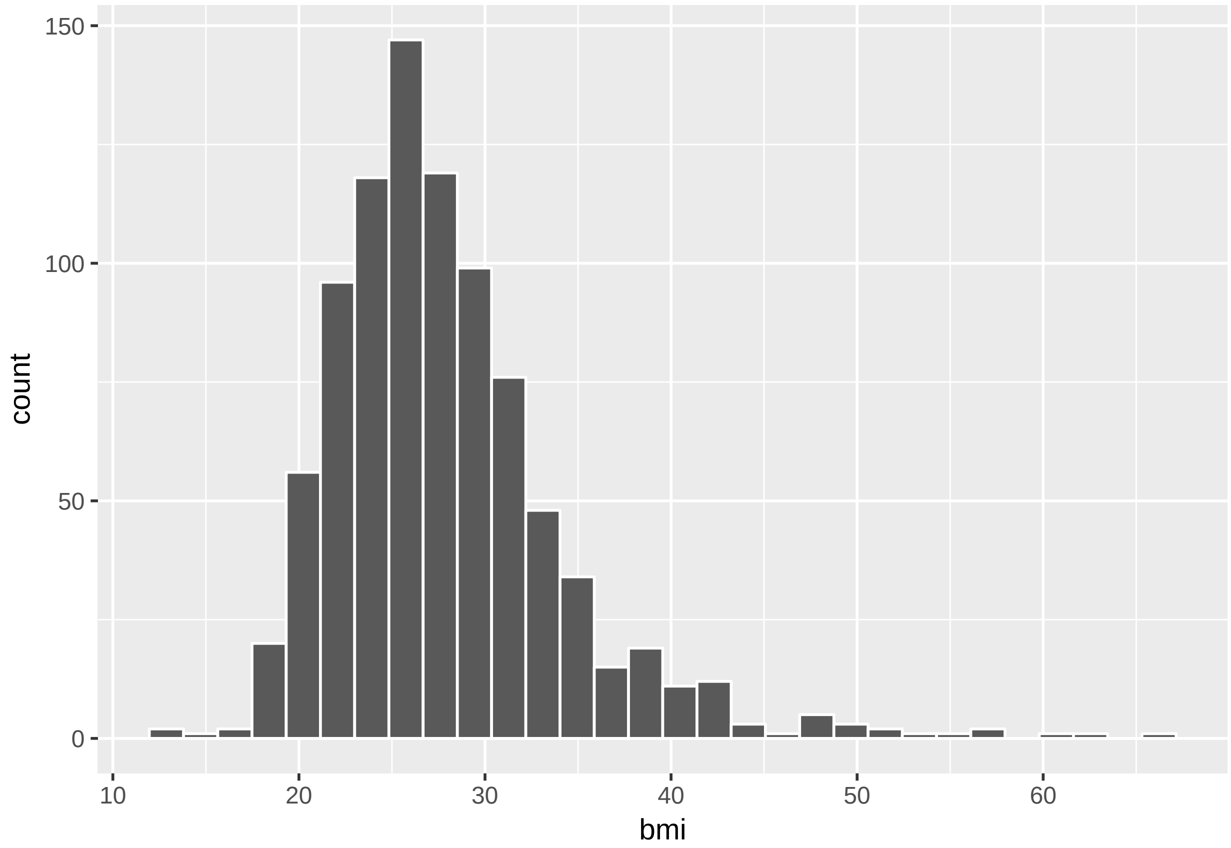
Of course, a natural summary of a quantitative variable like this would be graphical.

```
ggplot(smartcle2, aes(sleephrs)) +
    geom_histogram(binwidth = 1, fill = "dodgerblue", col = "darkred")
```



2.5.3 What's the distribution of `BMI`?

```
ggplot(smartcle2, aes(bmi)) +
    geom_histogram(bins = 30, col = "white")
```



2.5.4 How many of the respondents have a BMI below 30?

```
smartcle2 %>% count(bmi < 30) %>% mutate(proportion = n / sum(n))
```

```
# A tibble: 2 x 3
  `bmi < 30`     n proportion
  <lgl>      <int>      <dbl>
1 F            253      0.282
2 T            643      0.718
```

2.5.5 How many of the respondents who have a BMI < 30 exercised?

```
smartcle2 %>% count(exerany, bmi < 30) %>%
    group_by(exerany) %>%
    mutate(percent = 100*n/sum(n))
```

```
# A tibble: 4 x 4
# Groups: exerany [2]
  exerany `bmi < 30`     n percent
    <int> <lgl>      <int>   <dbl>
1       0 F             88    42.1
2       0 T            121    57.9
3       1 F            165    24.0
4       1 T            522    76.0
```

2.5.6 Is obesity associated with sex, in these data?

```
smartcle2 %>% count(female, bmi < 30) %>%
    group_by(female) %>%
    mutate(percent = 100*n/sum(n))
```

```
# A tibble: 4 x 4
# Groups: female [2]
  female `bmi < 30`     n percent
   <int> <lgl>      <int>   <dbl>
1      0 F            105    28.2
2      0 T            267    71.8
3      1 F            148    28.2
4      1 T            376    71.8
```

2.5.7 Comparing `sleephrs` summaries by obesity status

Can we compare the `sleephrs` means, medians and 75th percentiles for respondents whose BMI is below 30 to the respondents whose BMI is not?

```
smartcle2 %>%
    group_by(bmi < 30) %>%
    summarize(mean(sleephrs), median(sleephrs),
              q75 = quantile(sleephrs, 0.75))
```

```
# A tibble: 2 x 4
  `bmi < 30` `mean(sleephrs)` `median(sleephrs)`   q75
  <lgl>                 <dbl>              <int> <dbl>
1 F                      6.93                  7  8.00
2 T                      7.06                  7  8.00
```

2.5.8 The `skim` function within a pipe

The **skim** function works within pipes and with the other `tidyverse` functions.

```
smartcle2 %>%
    group_by(exerany) %>%
    skim(bmi, sleephrs)
```

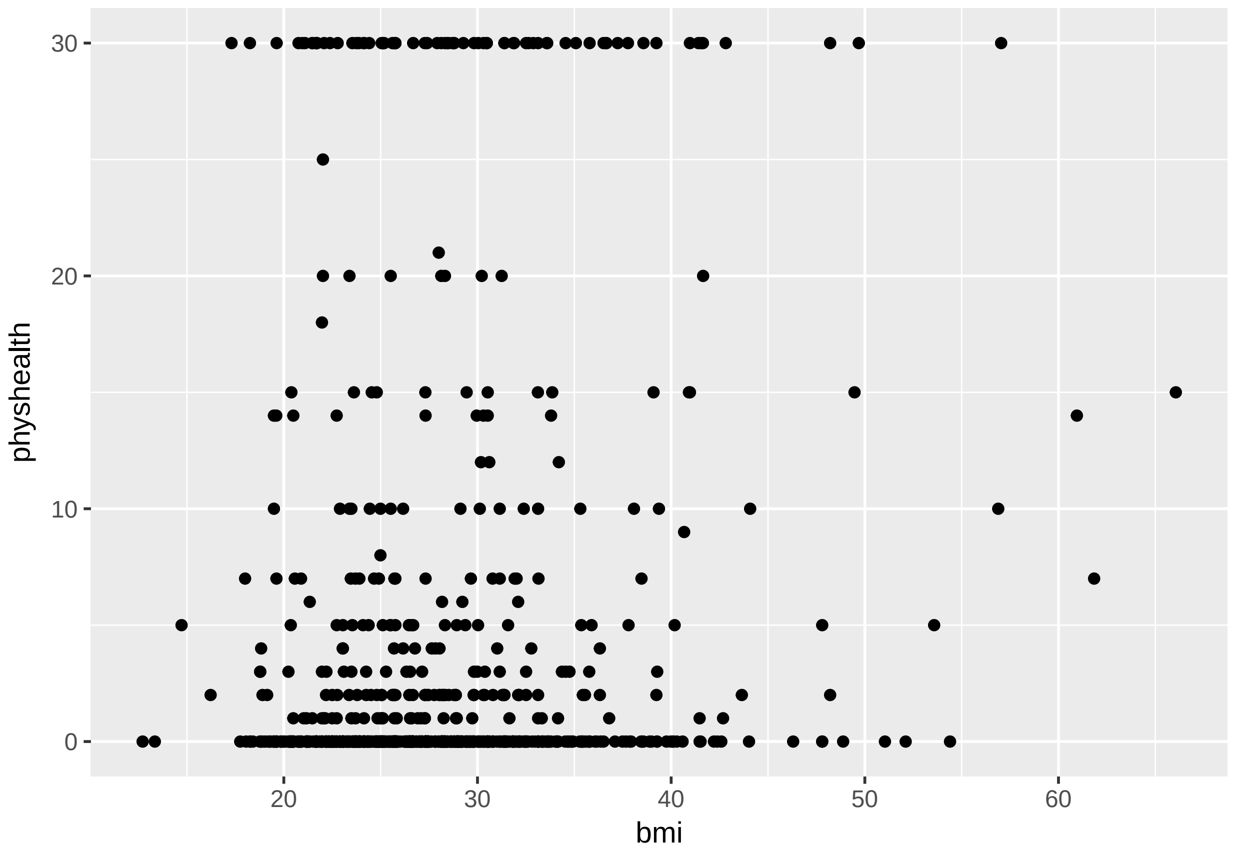
```
Skim summary statistics
 n obs: 896
 n variables: 10
 group variables: exerany

Variable type: integer
 exerany variable missing complete   n mean   sd p0 p25 median p75 p100
       0 sleephrs       0      209 209 7    1.85  1   6      7   8   20
       1 sleephrs       0      687 687 7.03 1.34  1   6      7   8   16

Variable type: numeric
 exerany variable missing complete   n  mean   sd    p0   p25 median   p75
       0      bmi       0      209 209 29.57 7.46 18    24.11  28.49 33.13
       1      bmi       0      687 687 27.35 5.84 12.71 23.7   26.52 29.81
  p100
```

```
66.06
60.95
```

2.6 First Modeling Attempt: Can `bmi` predict `physhealth`?

We'll start with an effort to predict `physhealth` using `bmi`. A natural graph would be a scatterplot.

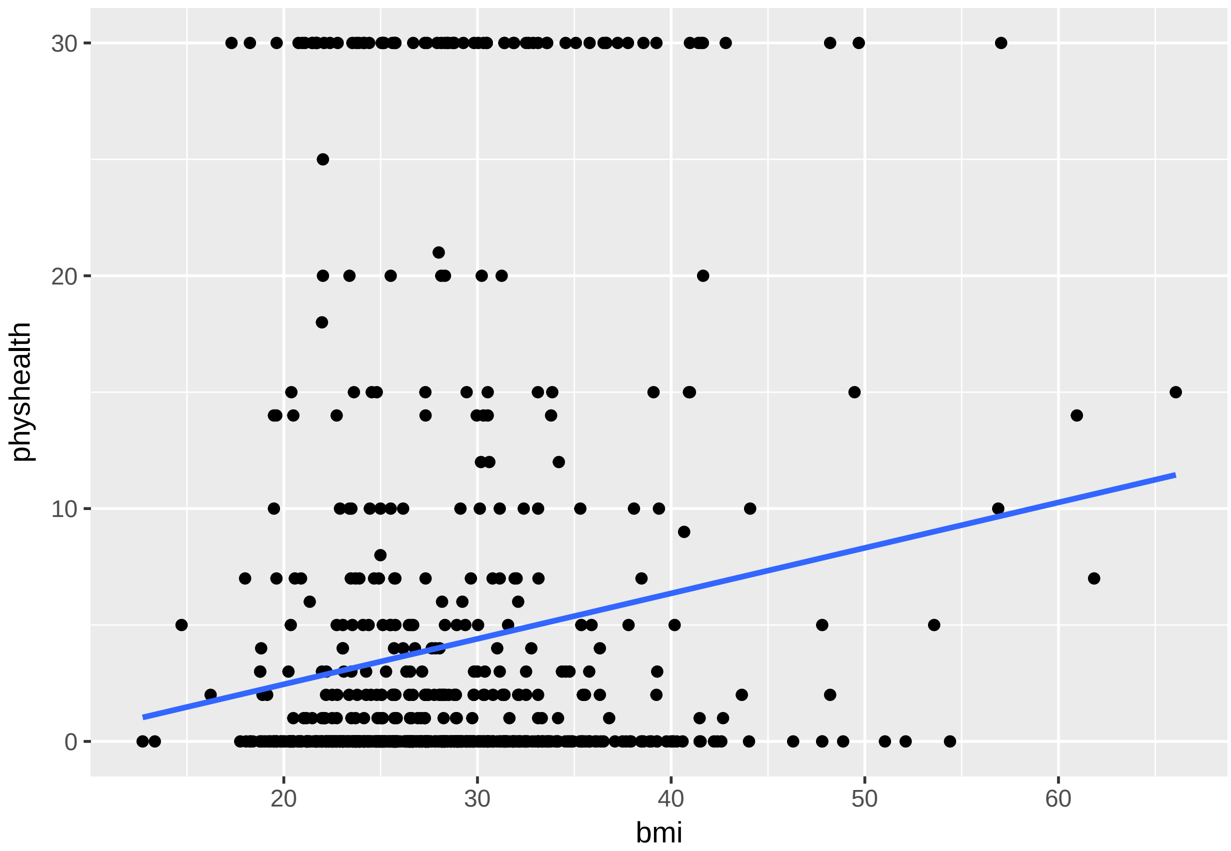
```
ggplot(data = smartcle2, aes(x = bmi, y = physhealth)) +
    geom_point()
```



A good question to ask ourselves here might be: "In what BMI range can we make a reasonable prediction of `physhealth`?"

Now, we might take the plot above and add a simple linear model …

```
ggplot(data = smartcle2, aes(x = bmi, y = physhealth)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE)
```



which shows the same least squares regression model that we can fit with the `lm` command.

2.6.1 Fitting a Simple Regression Model

```
model_A <- lm(physhealth ~ bmi, data = smartcle2)

model_A
```

```

Call:
lm(formula = physhealth ~ bmi, data = smartcle2)

Coefficients:
(Intercept)          bmi
    -1.4514       0.1953
```

```
summary(model_A)
```

```

Call:
lm(formula = physhealth ~ bmi, data = smartcle2)

Residuals:
   Min     1Q Median     3Q    Max
-9.171 -4.057 -3.193 -1.576 28.073
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.45143    1.29185  -1.124    0.262
bmi          0.19527    0.04521   4.319 1.74e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.556 on 894 degrees of freedom
Multiple R-squared:  0.02044,   Adjusted R-squared:  0.01934
F-statistic: 18.65 on 1 and 894 DF,  p-value: 1.742e-05
```

```
confint(model_A, level = 0.95)
```

```
                 2.5 %    97.5 %
(Intercept) -3.9868457 1.0839862
bmi          0.1065409 0.2840068
```

The model coefficients can be obtained by printing the model object, and the `summary` function provides several useful descriptions of the model's residuals, its statistical significance, and quality of fit.

2.6.2 Model Summary for a Simple (One-Predictor) Regression

The fitted model predicts `physhealth` with the equation -1.45 + 0.195*`bmi`, as we can read off from the model coefficients.

Each of the 896 respondents included in the `smartcle2` data makes a contribution to this model.

2.6.2.1 Residuals

Suppose Harry is one of the people in that group, and Harry's data is `bmi` = 20, and `physhealth` = 3.

- Harry's *observed* value of `physhealth` is just the value we have in the data for them, in this case, observed `physhealth` = 3 for Harry.
- Harry's *fitted* or *predicted* `physhealth` value is the result of calculating -1.45 + 0.195*`bmi` for Harry. So, if Harry's BMI was 20, then Harry's predicted `physhealth` value is -1.45 + (0.195)(20) = 2.45.
- The *residual* for Harry is then his *observed* outcome minus his *fitted* outcome, so Harry has a residual of 3 - 2.45 = 0.55.
- Graphically, a residual represents vertical distance between the observed point and the fitted regression line.
- Points above the regression line will have positive residuals, and points below the regression line will have negative residuals. Points on the line have zero residuals.

The residuals are summarized at the top of the `summary` output for linear model.

- The mean residual will always be zero in an ordinary least squares model, but a five number summary of the residuals is provided by the summary, as is an estimated standard deviation of the residuals (called here the Residual standard error.)
- In the `smartcle2` data, the minimum residual was -9.17, so for one subject, the observed value was 9.17 days smaller than the predicted value. This means that the prediction was 9.17 days too large for that subject.
- Similarly, the maximum residual was 28.07 days, so for one subject the prediction was 28.07 days too small. Not a strong performance.
- In a least squares model, the residuals are assumed to follow a Normal distribution, with mean zero, and standard deviation (for the `smartcle2` data) of about 8.6 days. Thus, by the definition of a Normal distribution, we'd expect
- about 68% of the residuals to be between -8.6 and +8.6 days,

- about 95% of the residuals to be between -17.2 and +17.2 days,
- about all (99.7%) of the residuals to be between -25.8 and +25.8 days.

2.6.2.2 Coefficients section

The `summary` for a linear model shows Estimates, Standard Errors, t values and *p* values for each coefficient fit.

- The Estimates are the point estimates of the intercept and slope of `bmi` in our model.
- In this case, our estimated slope is 0.195, which implies that if Harry's BMI is 20 and Sally's BMI is 21, we predict that Sally's `physhealth` will be 0.195 days larger than Harry's.
- The Standard Errors are also provided for each estimate. We can create rough 95% confidence intervals by adding and subtracting two standard errors from each coefficient, or we can get a slightly more accurate answer with the `confint` function.
- Here, the 95% confidence interval for the slope of `bmi` is estimated to be (0.11, 0.28). This is a good measure of the uncertainty in the slope that is captured by our model. We are 95% confident in the process of building this interval, but this doesn't mean we're 95% sure that the true slope is actually in that interval.

Also available are a *t* value (just the Estimate divided by the Standard Error) and the appropriate *p* value for testing the null hypothesis that the true value of the coefficient is 0 against a two-tailed alternative.

- If a slope coefficient is statistically significantly different from 0, this implies that 0 will not be part of the uncertainty interval obtained through `confint`.
- If the slope was zero, it would suggest that `bmi` would add no predictive value to the model. But that's unlikely here.

If the `bmi` slope coefficient is associated with a small *p* value, as in the case of our `model_A`, it suggests that the model including `bmi` is statistically significantly better at predicting `physhealth` than the model without `bmi`.

- Without `bmi` our `model_A` would become an *intercept-only* model, in this case, which would predict the mean `physhealth` for everyone, regardless of any other information.

2.6.2.3 Model Fit Summaries

The `summary` of a linear model also displays:

- The residual standard error and associated degrees of freedom for the residuals.
- For a simple (one-predictor) least regression like this, the residual degrees of freedom will be the sample size minus 2.
- The multiple R-squared (or coefficient of determination)
- This is interpreted as the proportion of variation in the outcome (`physhealth`) accounted for by the model, and will always fall between 0 and 1 as a result.
- Our model_A accounts for a mere 2% of the variation in `physhealth`.
- The Adjusted R-squared value "adjusts" for the size of our model in terms of the number of coefficients included in the model.
- The adjusted R-squared will always be less than the Multiple R-squared.
- We still hope to find models with relatively large adjusted R^2 values.
- In particular, we hope to find models where the adjusted R^2 isn't substantially less than the Multiple R-squared.
- The adjusted R-squared is usually a better estimate of likely performance of our model in new data than is the Multiple R-squared.
- The adjusted R-squared result is no longer interpretable as a proportion of anything - in fact, it can fall below 0.

- We can obtain the adjusted R^2 from the raw R^2, the number of observations N and the number of predictors p included in the model, as follows:

$$R^2_{adj} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1},$$

- The F statistic and p value from a global ANOVA test of the model.
  - Obtaining a statistically significant result here is usually pretty straightforward, since the comparison is between our model, and a model which simply predicts the mean value of the outcome for everyone.
  - In a simple (one-predictor) linear regression like this, the t statistic for the slope is just the square root of the F statistic, and the resulting p values for the slope's t test and for the global F test will be identical.
- To see the complete ANOVA F test for this model, we can run `anova(model_A)`.

```
anova(model_A)
```

```
Analysis of Variance Table

Response: physhealth
           Df Sum Sq Mean Sq F value    Pr(>F)
bmi         1   1366  1365.5  18.655 1.742e-05 ***
Residuals 894  65441    73.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.6.3 Using the `broom` package

The `broom` package has three functions of particular use in a linear regression model:

2.6.3.1 The `tidy` function

`tidy` builds a data frame/tibble containing information about the coefficients in the model, their standard errors, t statistics and p values.

```
tidy(model_A)
```

```
         term   estimate  std.error statistic      p.value
1 (Intercept) -1.4514298 1.29185199 -1.123526 2.615156e-01
2         bmi  0.1952739 0.04521145  4.319125 1.741859e-05
```

2.6.3.2 The `glance` function

glance' builds a data frame/tibble containing summary statistics about the model, including

- the (raw) multiple R^2 and adjusted R^2
- `sigma` which is the residual standard error
- the F `statistic`, `p.value` model `df` and `df.residual` associated with the global ANOVA test, plus
- several statistics that will be useful in comparing models down the line:
- the model's log likelihood function value, `logLik`
- the model's Akaike's Information Criterion value, `AIC`
- the model's Bayesian Information Criterion value, `BIC`
- and the model's `deviance` statistic

```
glance(model_A)
```

```
   r.squared adj.r.squared    sigma statistic      p.value df    logLik
1 0.02044019    0.01934449 8.555737  18.65484 1.741859e-05  2 -3193.723
       AIC     BIC deviance df.residual
1 6393.446 6407.84 65441.36         894
```

2.6.3.3 The augment function

`augment` builds a data frame/tibble which adds fitted values, residuals and other diagnostic summaries that describe each observation to the original data used to fit the model, and this includes

- `.fitted` and `.resid`, the fitted and residual values, in addition to
- `.hat`, the leverage value for this observation
- `.cooksd`, the Cook's distance measure of *influence* for this observation
- `.stdresid`, the standardized residual (think of this as a z-score - a measure of the residual divided by its associated standard deviation `.sigma`)
- and `se.fit` which will help us generate prediction intervals for the model downstream

Note that each of the new columns begins with `.` to avoid overwriting any data.

```
head(augment(model_A))
```

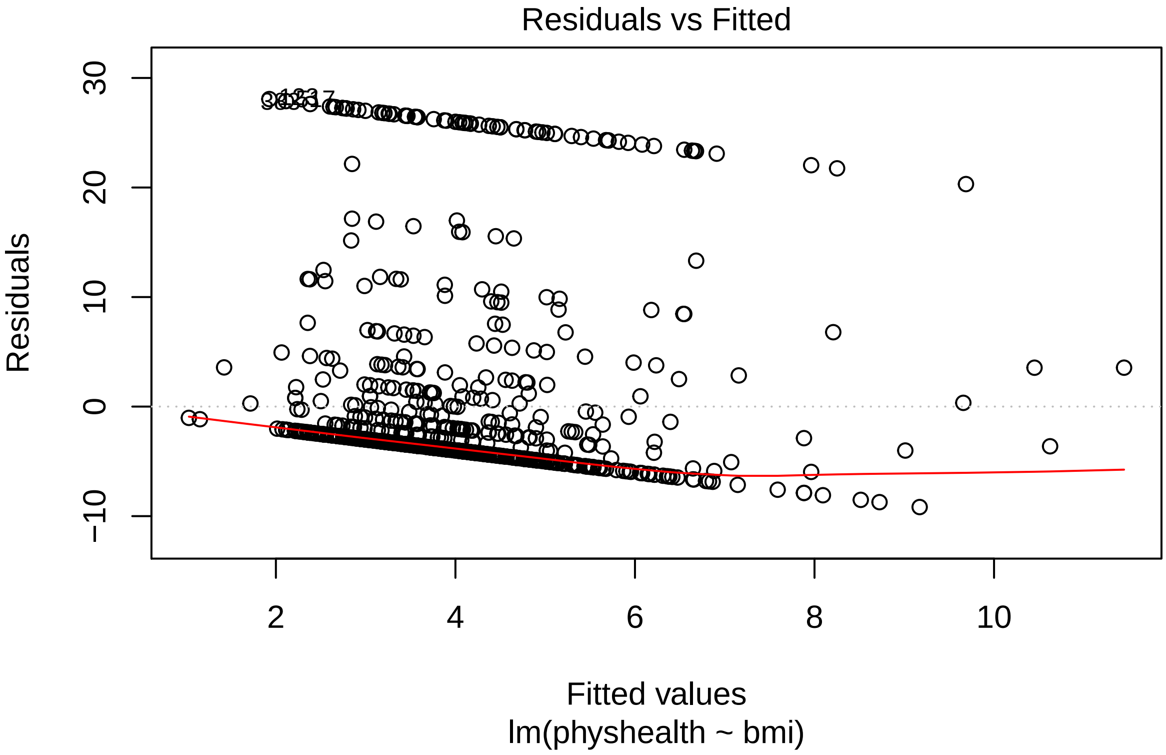
```
  physhealth   bmi  .fitted   .se.fit      .resid         .hat   .sigma
1          0 26.69 3.760430 0.2907252 -3.76043009 0.001154651 8.559600
2          0 23.70 3.176561 0.3422908 -3.17656119 0.001600574 8.559865
3          1 26.92 3.805343 0.2890054 -2.80534308 0.001141030 8.560010
4          0 21.66 2.778202 0.4005101 -2.77820248 0.002191352 8.560020
5          5 24.09 3.252718 0.3329154  1.74728200 0.001514095 8.560326
6          4 27.64 3.945940 0.2860087  0.05405972 0.001117490 8.560526
       .cooksd   .std.resid
1 1.117852e-04 -0.439775451
2 1.106717e-04 -0.371575999
3 6.147744e-05 -0.328077528
4 1.160381e-04 -0.325074461
5 3.167016e-05  0.204378225
6 2.235722e-08  0.006322069
```

For more on the `broom` package, you may want to look at this vignette.

2.6.4 How does the model do? (Residuals vs. Fitted Values)

- Remember that the R^2 value was about 2%.

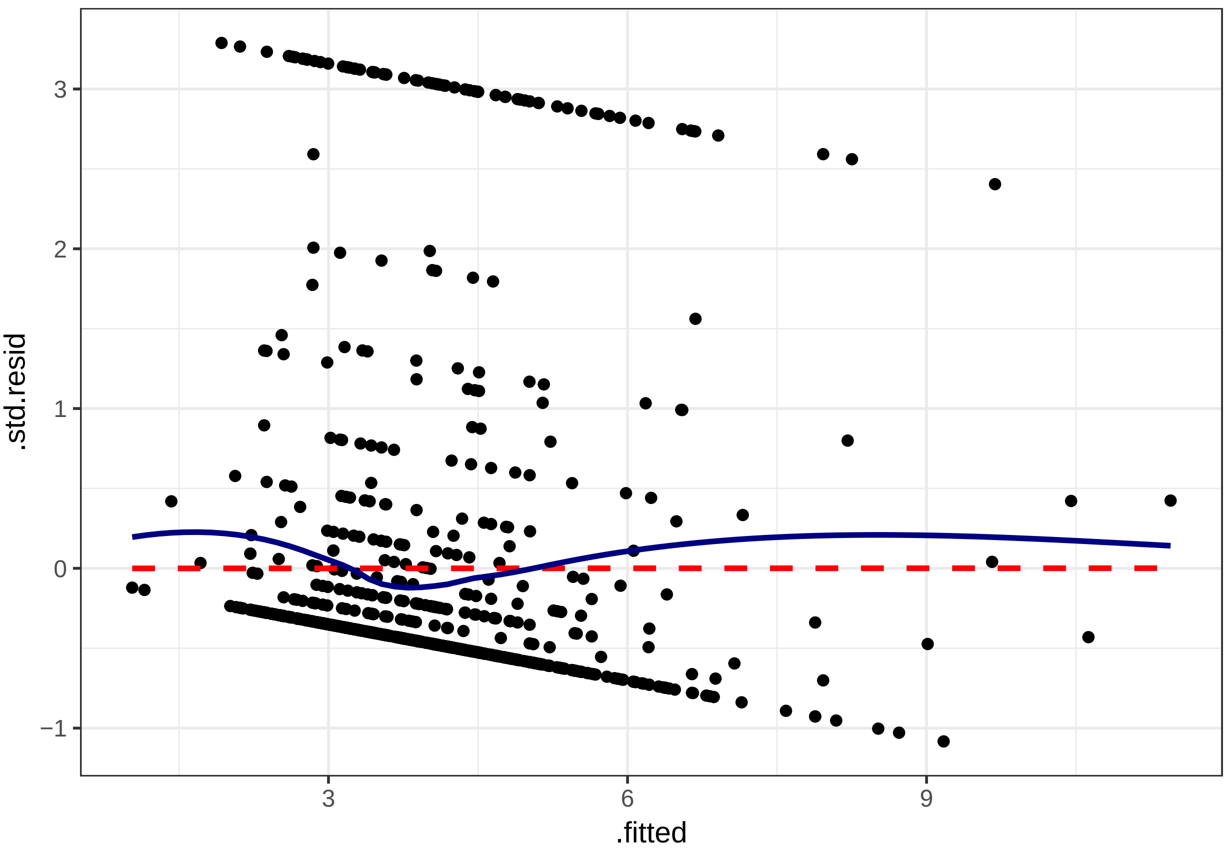
```r
plot(model_A, which = 1)
```



This is a plot of residuals vs. fitted values. The goal here is for this plot to look like a random scatter of points, perhaps like a "fuzzy football", and that's **not** what we have. Why?

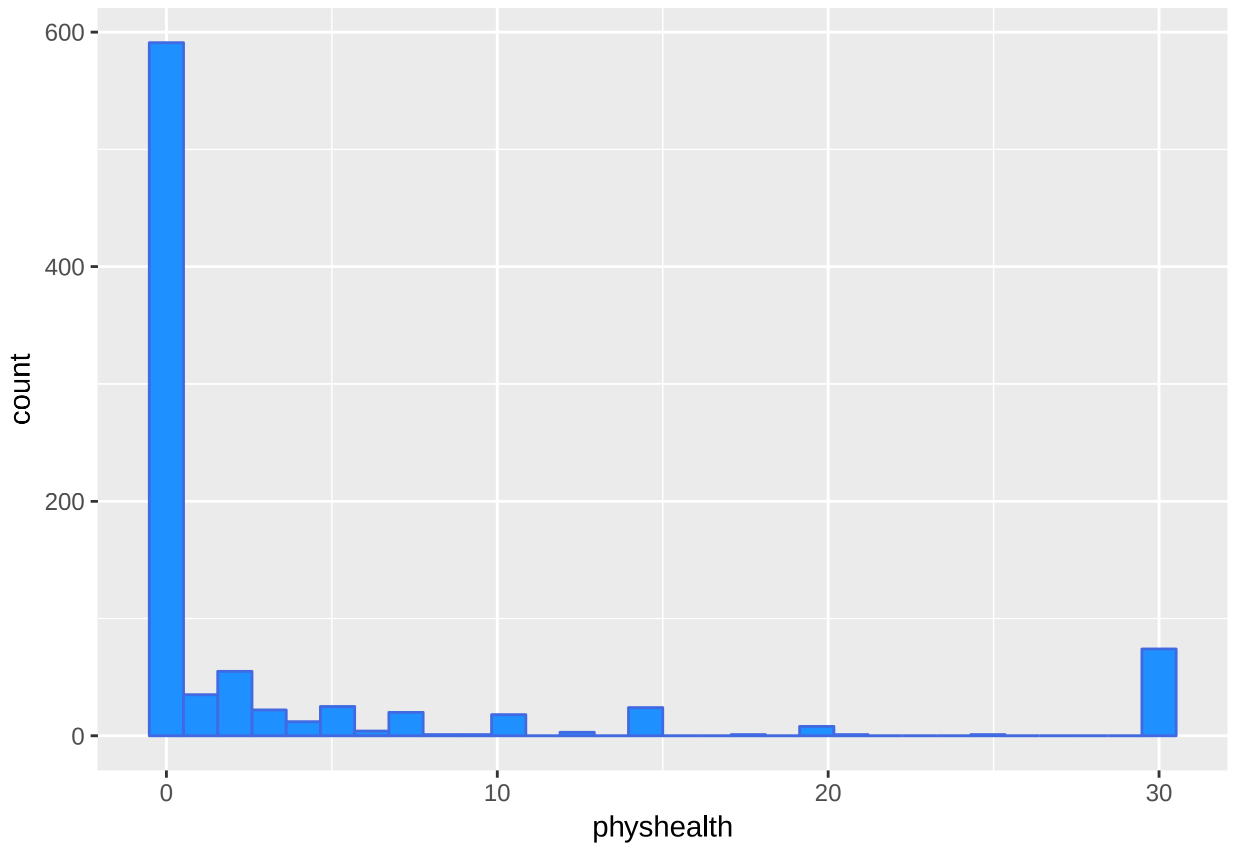
If you prefer, here's a `ggplot2` version of a similar plot, now looking at standardized residuals instead of raw residuals, and adding a loess smooth and a linear fit to the result.

```r
ggplot(augment(model_A), aes(x = .fitted, y = .std.resid)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE, col = "red", linetype = "dashed") +
    geom_smooth(method = "loess", se = FALSE, col = "navy") +
    theme_bw()
```



The problem we're having here becomes, I think, a little more obvious if we look at what we're predicting. Does `physhealth` look like a good candidate for a linear model?

```
ggplot(smartcle2, aes(x = physhealth)) +
geom_histogram(bins = 30, fill = "dodgerblue", color = "royalblue")
```



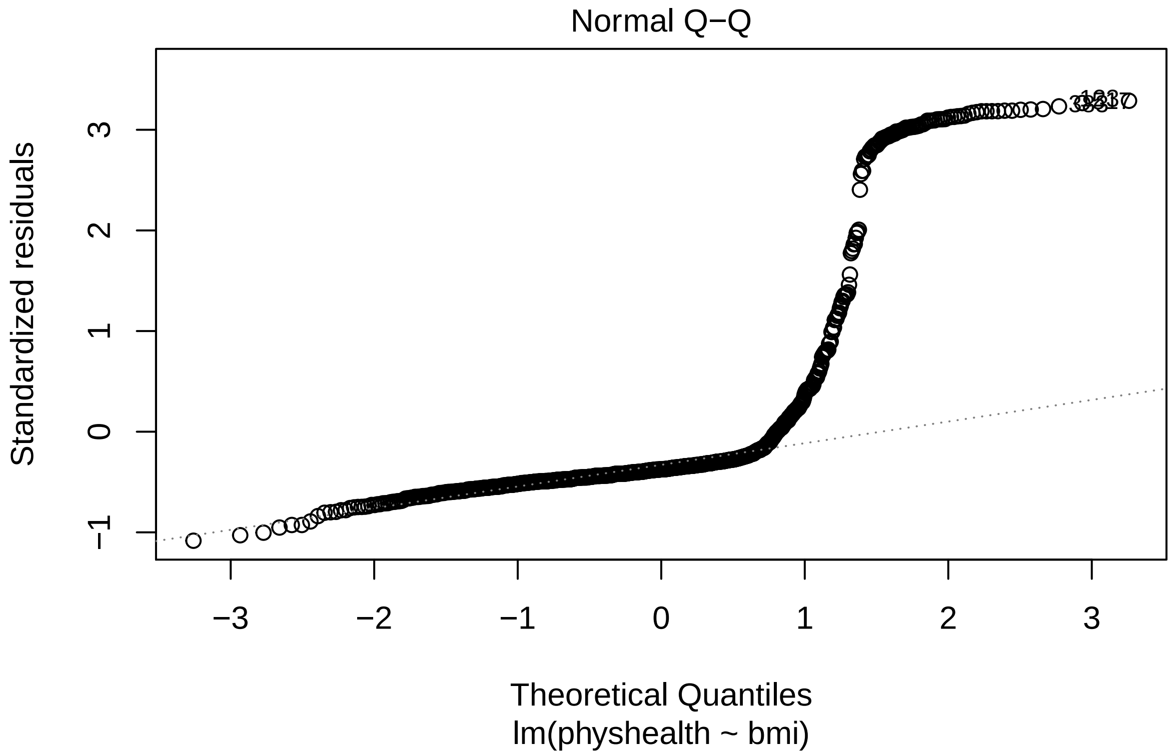
```
smartcle2 %>% count(physhealth == 0, physhealth == 30)
```

```
# A tibble: 3 x 3
  `physhealth == 0` `physhealth == 30`     n
  <lgl>             <lgl>              <int>
1 F                 F                    231
2 F                 T                     74
3 T                 F                    591
```

No matter what model we fit, if we are predicting `physhealth`, and most of the data are values of 0 and 30, we have limited variation in our outcome, and so our linear model will be somewhat questionable just on that basis.

A normal Q-Q plot of the standardized residuals for our `model_A` shows this problem, too.

```
plot(model_A, which = 2)
```



We're going to need a method to deal with this sort of outcome, that has both a floor and a ceiling. We'll get there eventually, but linear regression alone doesn't look promising.

All right, so that didn't go anywhere great. Let's try again, with a new outcome.

2.7 A New Small Study: Predicting BMI

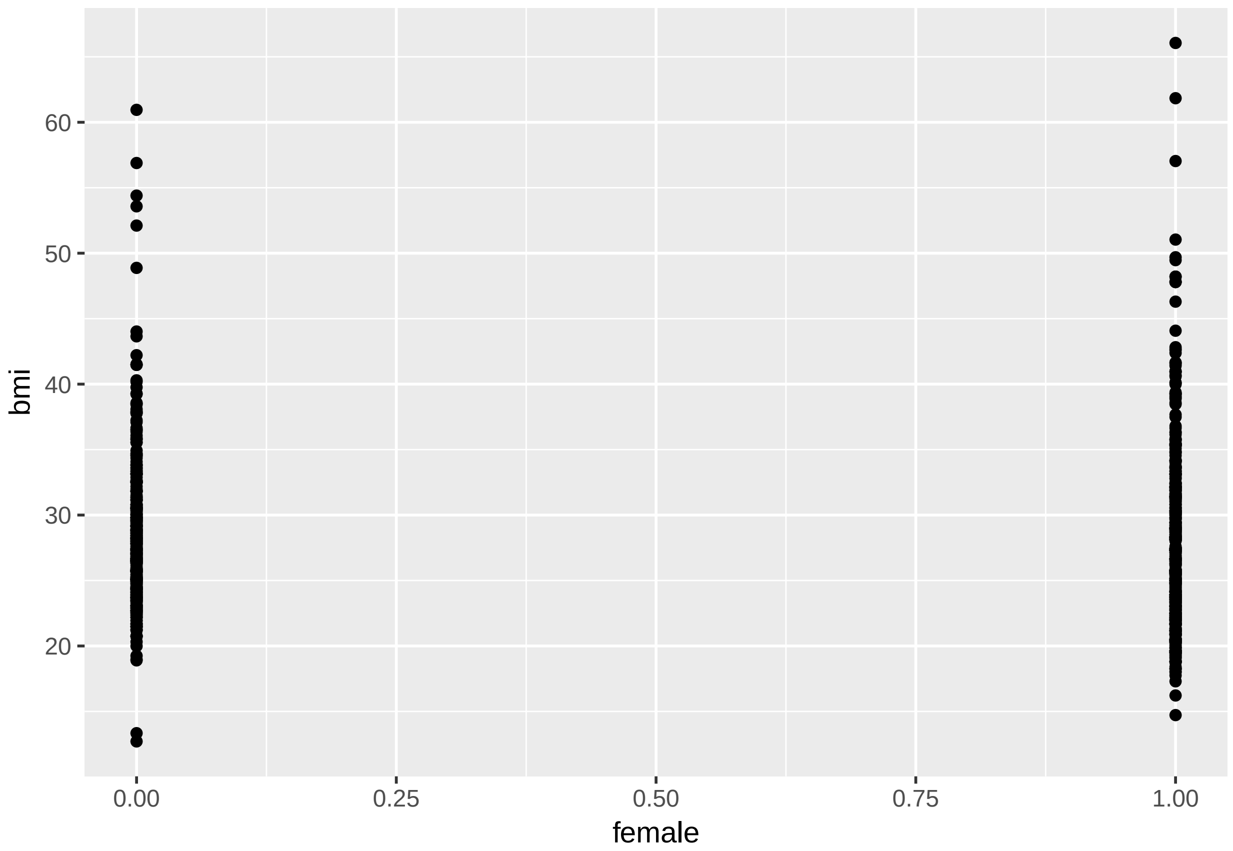
We'll begin by investigating the problem of predicting `bmi`, at first with just three regression inputs: `sex`, `exerany` and `sleephrs`, in our new `smartcle2` data set.

- The outcome of interest is `bmi`.
- Inputs to the regression model are:
  - `female` = 1 if the subject is female, and 0 if they are male
  - `exerany` = 1 if the subject exercised in the past 30 days, and 0 if they didn't
  - `sleephrs` = hours slept in a typical 24-hour period (treated as quantitative)

2.7.1 Does `female` predict `bmi` well?

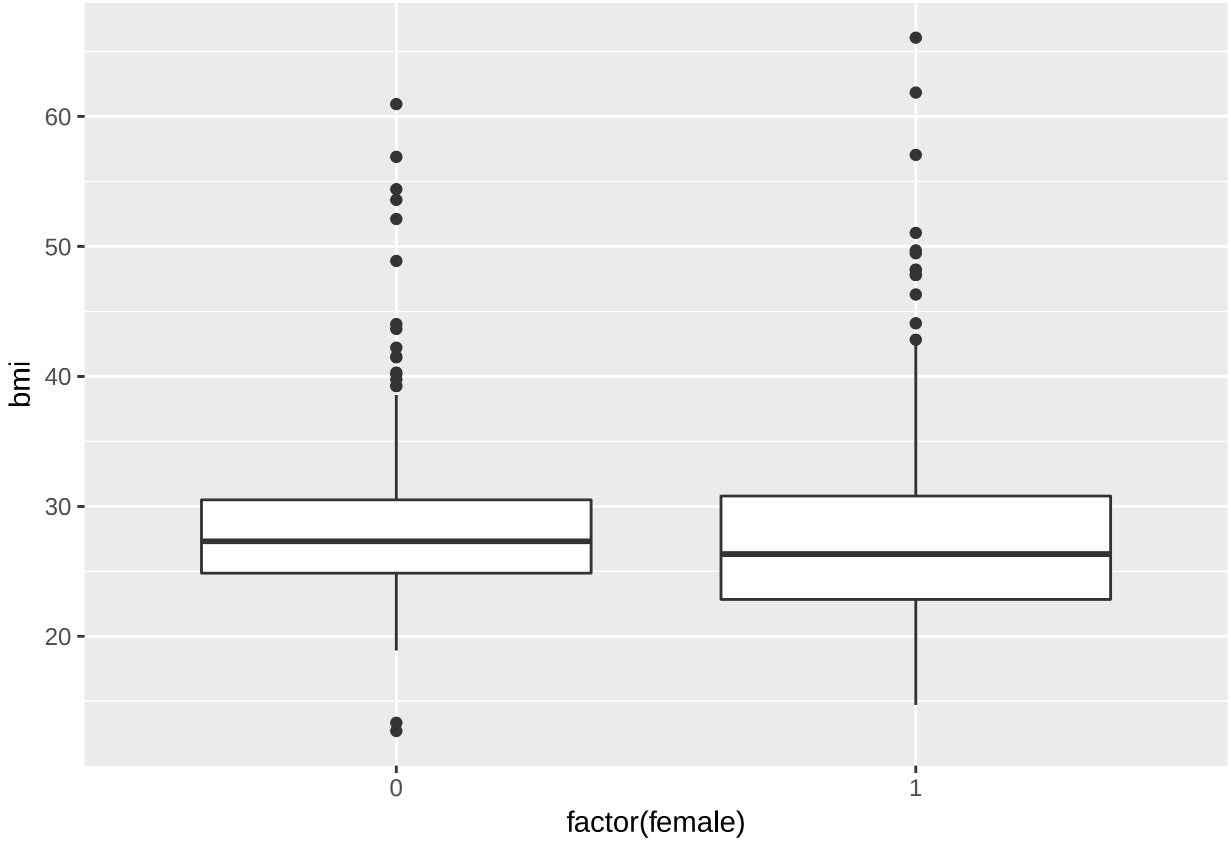
2.7.1.1 Graphical Assessment

```
ggplot(smartcle2, aes(x = female, y = bmi)) +
    geom_point()
```



Not so helpful. We should probably specify that `female` is a factor, and try another plotting approach.

```
ggplot(smartcle2, aes(x = factor(female), y = bmi)) +
    geom_boxplot()
```



The median BMI looks a little higher for males. Let's see if a model reflects that.

2.8 c2_m1: A simple t-test model

```
c2_m1 <- lm(bmi ~ female, data = smartcle2)
c2_m1
```

```

Call:
lm(formula = bmi ~ female, data = smartcle2)

Coefficients:
(Intercept)       female
    28.3600      -0.8457
```

```
summary(c2_m1)
```

```

Call:
lm(formula = bmi ~ female, data = smartcle2)

Residuals:
    Min      1Q  Median      3Q     Max
-15.650  -4.129  -1.080   2.727  38.546

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  28.3600     0.3274  86.613   <2e-16 ***
female       -0.8457     0.4282  -1.975   0.0485 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.315 on 894 degrees of freedom
Multiple R-squared:  0.004345,  Adjusted R-squared:  0.003231
F-statistic: 3.902 on 1 and 894 DF,  p-value: 0.04855
```

```
confint(c2_m1)
```

```
                2.5 %      97.5 %
(Intercept) 27.717372 29.00262801
female      -1.686052 -0.00539878
```

The model suggests, based on these 896 subjects, that

- our best prediction for males is BMI = 28.36 kg/m^2, and
- our best prediction for females is BMI = 28.36 - 0.85 = 27.51 kg/m^2.
- the mean difference between females and males is -0.85 kg/m^2 in BMI
- a 95% confidence (uncertainty) interval for that mean female - male difference in BMI ranges from -1.69 to -0.01
- the model accounts for 0.4% of the variation in BMI, so that knowing the respondent's sex does very little to reduce the size of the prediction errors as compared to an intercept only model that would predict the overall mean (regardless of sex) for all subjects.
- the model makes some enormous errors, with one subject being predicted to have a BMI 38 points lower than his/her actual BMI.

Note that this simple regression model just gives us the t-test.

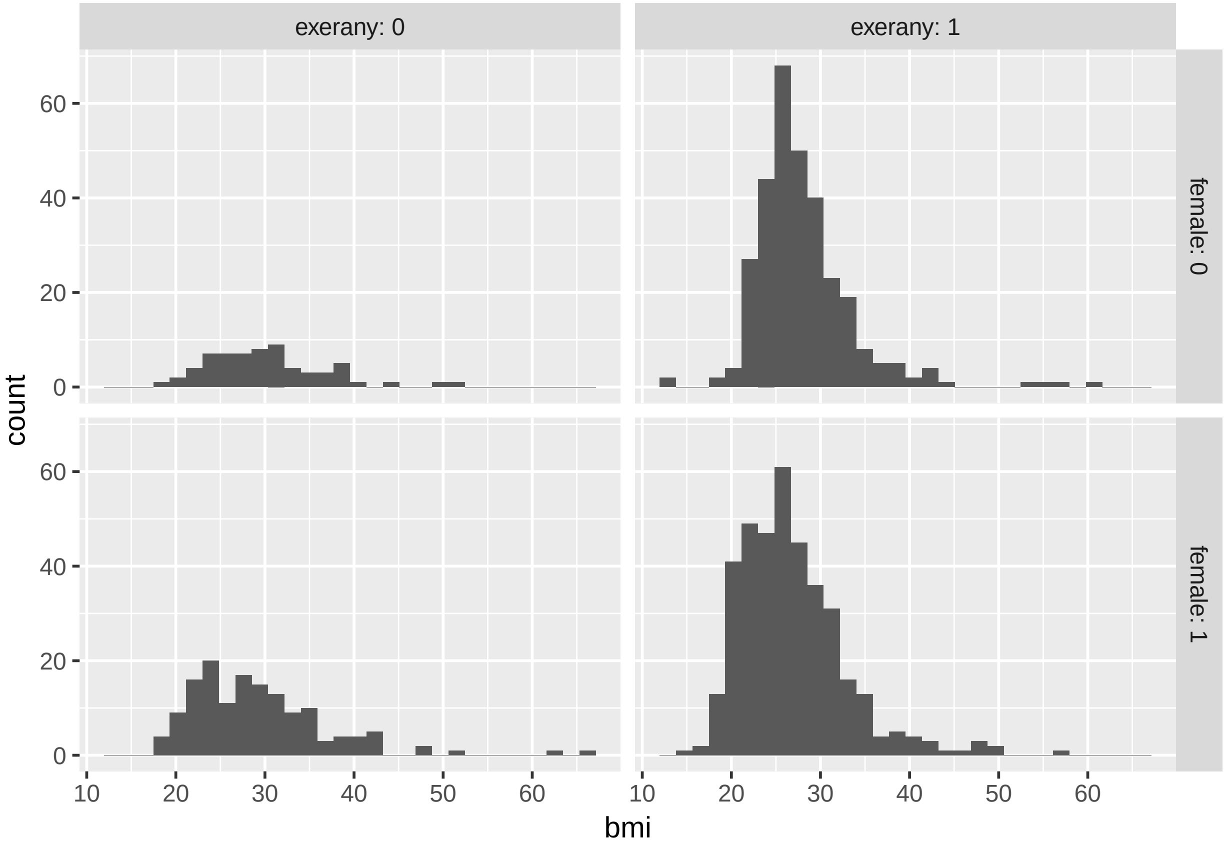
```
t.test(bmi ~ female, var.equal = TRUE, data = smartcle2)
```

```


    Two Sample t-test

data:  bmi by female
t = 1.9752, df = 894, p-value = 0.04855
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.00539878 1.68605160
sample estimates:
mean in group 0 mean in group 1 
       28.36000        27.51427 
```

2.9 c2_m2: Adding another predictor (two-way ANOVA without interaction)

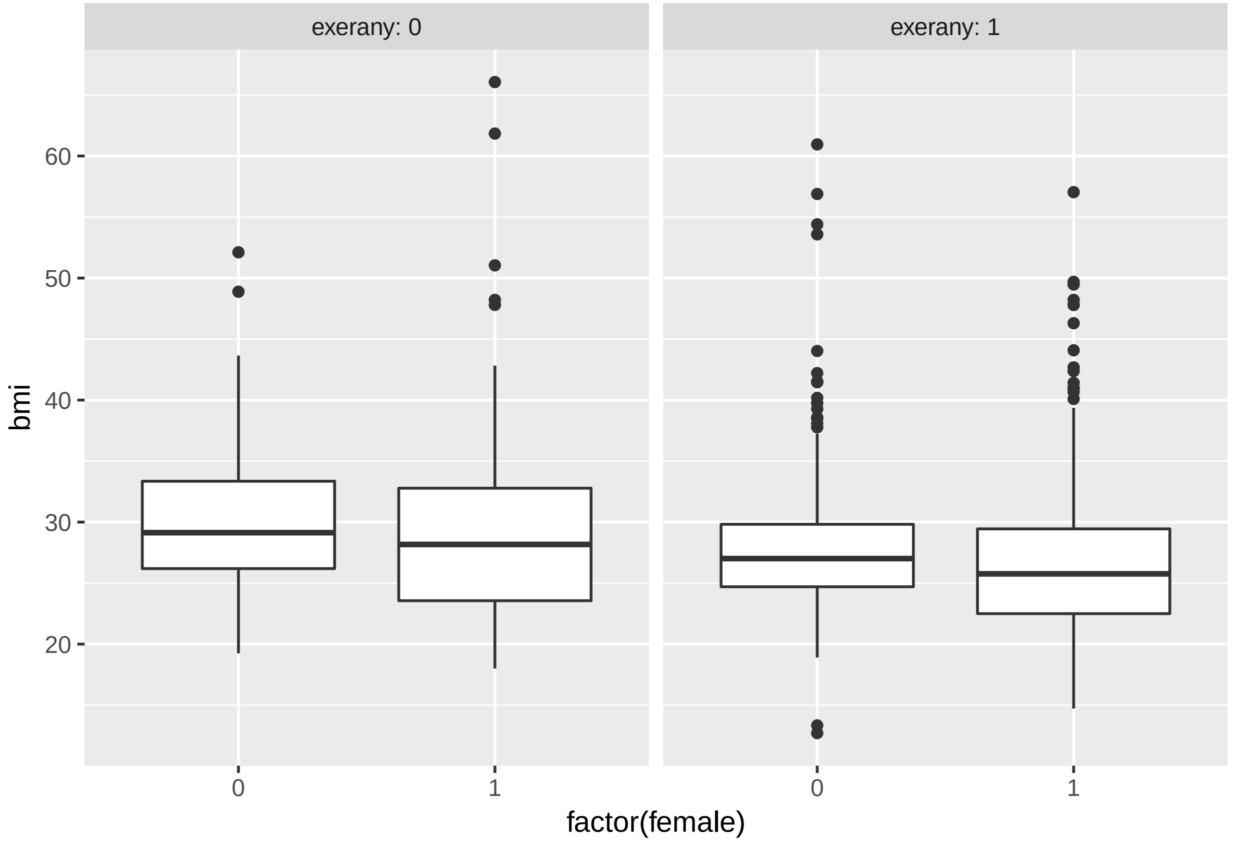
When we add in the information about `exerany` to our original model, we might first picture the data. We could look at separate histograms,

```
ggplot(smartcle2, aes(x = bmi)) +
    geom_histogram(bins = 30) +
    facet_grid(female ~ exerany, labeller = label_both)
```

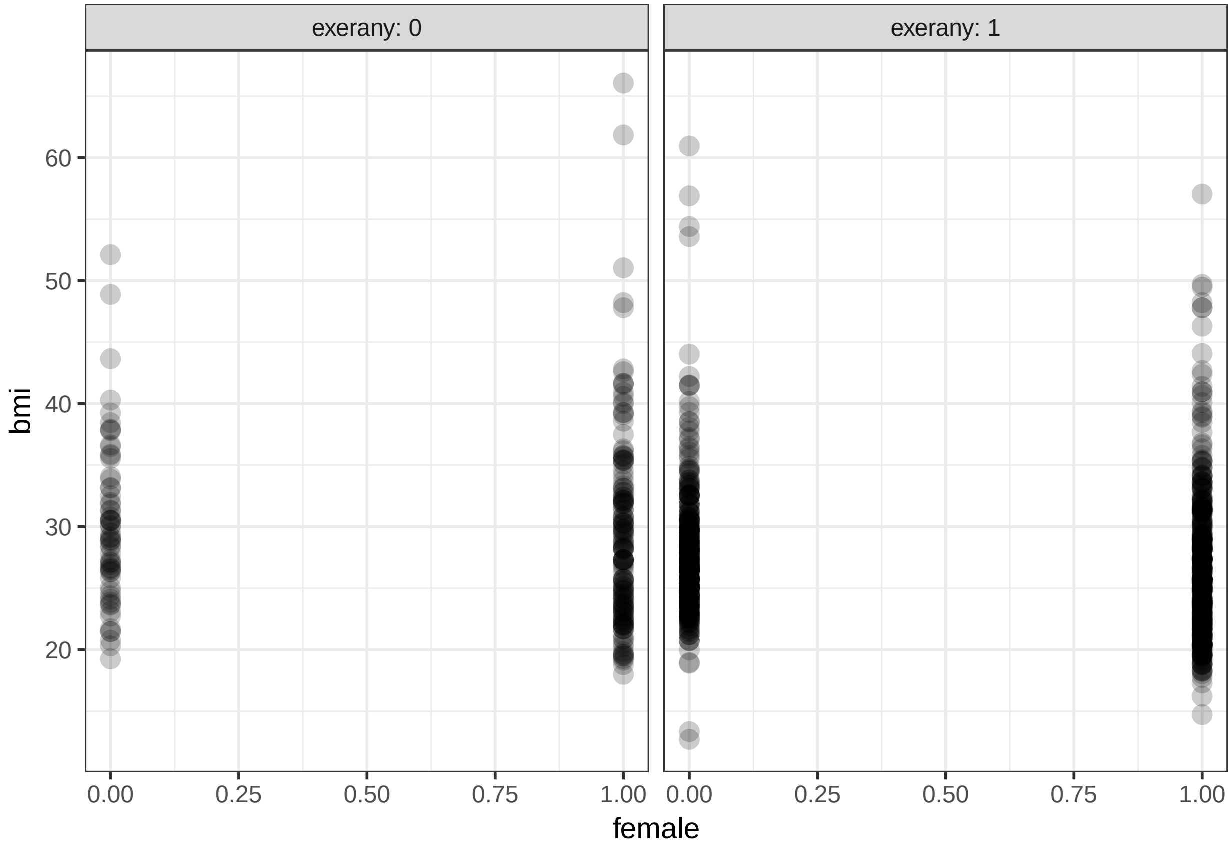


or maybe boxplots?

```
ggplot(smartcle2, aes(x = factor(female), y = bmi)) +
    geom_boxplot() +
    facet_wrap(~ exerany, labeller = label_both)
```



```
ggplot(smartcle2, aes(x = female, y = bmi))+
    geom_point(size = 3, alpha = 0.2) +
    theme_bw() +
    facet_wrap(~ exerany, labeller = label_both)
```



OK. Let's try fitting a model.

```
c2_m2 <- lm(bmi ~ female + exerany, data = smartcle2)
c2_m2
```

```

Call:
lm(formula = bmi ~ female + exerany, data = smartcle2)

Coefficients:
(Intercept)       female      exerany
     30.334       -1.095       -2.384
```

This new model predicts only four predicted values:

- `bmi` = 30.334 if the subject is male and did not exercise (so `female` = 0 and `exerany` = 0)
- `bmi` = 30.334 - 1.095 = 29.239 if the subject is female and did not exercise (`female` = 1 and `exerany` = 0)
- `bmi` = 30.334 - 2.384 = 27.950 if the subject is male and exercised (so `female` = 0 and `exerany` = 1), and, finally
- `bmi` = 30.334 - 1.095 - 2.384 = 26.855 if the subject is female and exercised (so both `female` and `exerany` = 1).

For those who did not exercise, the model is:

- `bmi` = 30.334 - 1.095 `female`

and for those who did exercise, the model is:

- `bmi` = 27.95 - 1.095 `female`

Only the intercept of the bmi-female model changes depending on exerany.

```
summary(c2_m2)
```

```

Call:
lm(formula = bmi ~ female + exerany, data = smartcle2)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.240  -4.091  -1.095   2.602  36.822 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  30.3335     0.5231   57.99  < 2e-16 ***
female       -1.0952     0.4262   -2.57   0.0103 *  
exerany      -2.3836     0.4965   -4.80 1.86e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.239 on 893 degrees of freedom
Multiple R-squared:  0.02939,	Adjusted R-squared:  0.02722 
F-statistic: 13.52 on 2 and 893 DF,  p-value: 1.641e-06
```

```
confint(c2_m2)
```

```
                2.5 %     97.5 %
(Intercept) 29.306846 31.3602182
female      -1.931629 -0.2588299
exerany     -3.358156 -1.4090777
```

The slopes of both female and exerany have confidence intervals that are completely below zero, indicating that both female sex and exerany appear to be associated with reductions in bmi.

The R^2 value suggests that just under 3% of the variation in bmi is accounted for by this ANOVA model.

In fact, this regression (on two binary indicator variables) is simply a two-way ANOVA model without an interaction term.

```
anova(c2_m2)
```

```
Analysis of Variance Table

Response: bmi
           Df Sum Sq Mean Sq F value    Pr(>F)    
female      1    156  155.61  3.9977   0.04586 *  
exerany     1    897  896.93 23.0435 1.856e-06 ***
Residuals 893  34759   38.92                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.10 c2_m3: Adding the interaction term (Two-way ANOVA with interaction)

Suppose we want to let the effect of female vary depending on the exerany status. Then we need to incorporate an interaction term in our model.

```
c2_m3 <- lm(bmi ~ female * exerany, data = smartcle2)
c2_m3
```

```

Call:
lm(formula = bmi ~ female * exerany, data = smartcle2)

Coefficients:
   (Intercept)           female          exerany   female:exerany  
       30.1359          -0.8104          -2.1450          -0.3592  
```

So, for example, for a male who exercises, this model predicts

- `bmi` = 30.136 - 0.810 (0) - 2.145 (1) - 0.359 (0)(1) = 30.136 - 2.145 = 27.991

And for a female who exercises, the model predicts

- `bmi` = 30.136 - 0.810 (1) - 2.145 (1) - 0.359 (1)(1) = 30.136 - 0.810 - 2.145 - 0.359 = 26.822

For those who did not exercise, the model is:

- `bmi` = 30.136 - 0.81 `female`

But for those who did exercise, the model is:

- `bmi` = (30.136 - 2.145) + (-0.810 + (-0.359)) `female`, or „,
- `bmi` = 27.991 - 1.169 `female`

Now, both the slope and the intercept of the `bmi-female` model change depending on `exerany`.

```
summary(c2_m3)
```

```

Call:
lm(formula = bmi ~ female * exerany, data = smartcle2)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.281  -4.101  -1.061   2.566  36.734 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     30.1359     0.7802  38.624   <2e-16 ***
female          -0.8104     0.9367  -0.865   0.3872    
exerany         -2.1450     0.8575  -2.501   0.0125 *  
female:exerany  -0.3592     1.0520  -0.341   0.7328    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.242 on 892 degrees of freedom
Multiple R-squared:  0.02952,   Adjusted R-squared:  0.02625 
F-statistic: 9.044 on 3 and 892 DF,  p-value: 6.669e-06
```

```
confint(c2_m3)
```

```
                   2.5 %     97.5 %
(Intercept)    28.604610 31.6672650
female         -2.648893  1.0280526
exerany        -3.827886 -0.4620407
female:exerany -2.423994  1.7055248
```

In fact, this regression (on two binary indicator variables and a product term) is simply a two-way ANOVA model with an interaction term.

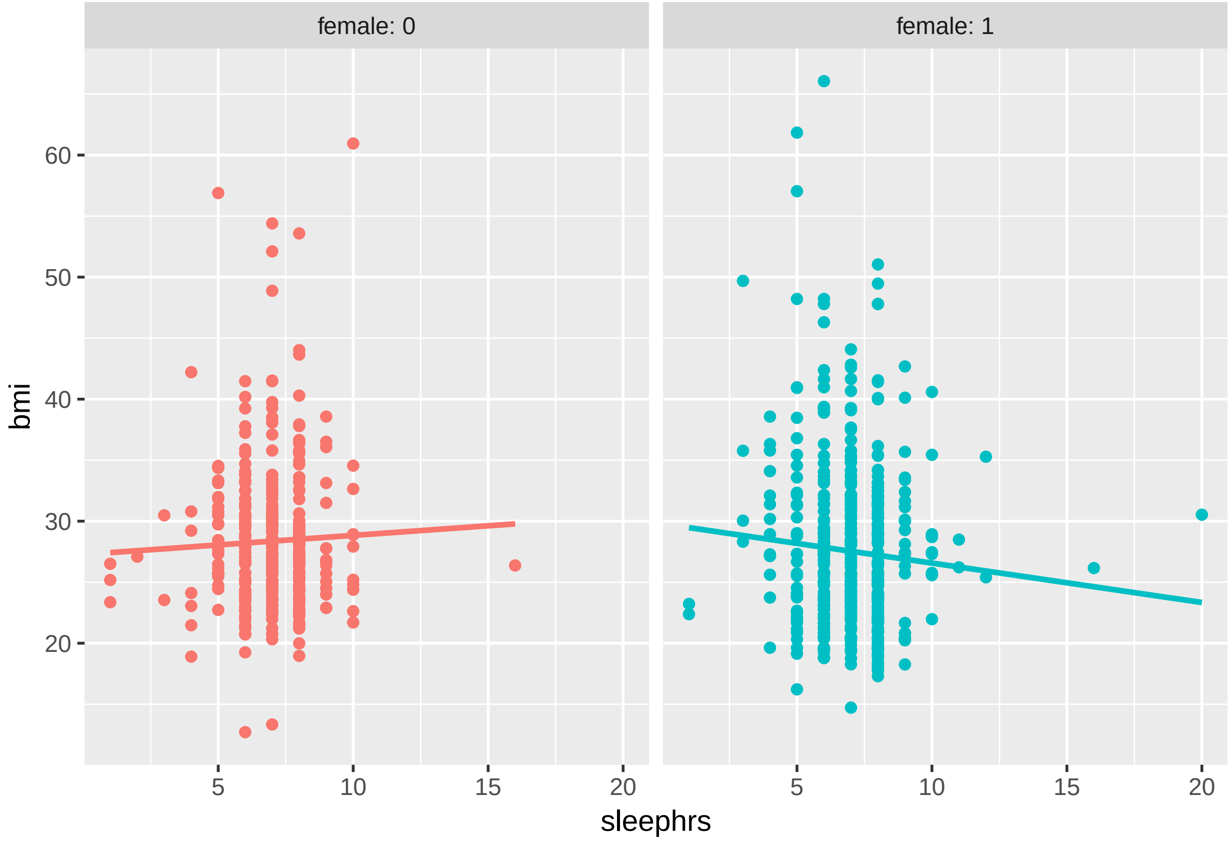
```
anova(c2_m3)
```

```
Analysis of Variance Table

Response: bmi
                 Df Sum Sq Mean Sq F value    Pr(>F)    
female            1    156  155.61  3.9938   0.04597 *  
exerany           1    897  896.93 23.0207 1.878e-06 ***
female:exerany    1      5    4.54  0.1166   0.73283    
Residuals       892  34754   38.96                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction term doesn't change very much here. Its uncertainty interval includes zero, and the overall model still accounts for just under 3% of the variation in bmi.

2.11 c2_m4: Using female and sleephrs in a model for bmi

```
ggplot(smartcle2, aes(x = sleephrs, y = bmi, color = factor(female))) +
    geom_point() +
    guides(col = FALSE) +
    geom_smooth(method = "lm", se = FALSE) +
    facet_wrap(~ female, labeller = label_both)
```



Does the difference in slopes of `bmi` and `sleephrs` for males and females appear to be substantial and important?

```
c2_m4 <- lm(bmi ~ female * sleephrs, data = smartcle2)

summary(c2_m4)
```

```

Call:
lm(formula = bmi ~ female * sleephrs, data = smartcle2)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.498  -4.179  -1.035   2.830  38.204 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)      27.2661     1.6320  16.707   <2e-16 ***
female            2.5263     2.0975   1.204    0.229    
sleephrs          0.1569     0.2294   0.684    0.494    
female:sleephrs  -0.4797     0.2931  -1.636    0.102    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.31 on 892 degrees of freedom
Multiple R-squared:  0.008341,  Adjusted R-squared:  0.005006 
F-statistic: 2.501 on 3 and 892 DF,  p-value: 0.05818
```

Does it seem as though the addition of `sleephrs` has improved our model substantially over a model with `female` alone (which, you recall, was `c2_m1`)?

Since the `c2_m4` model contains the `c2_m1` model's predictors as a subset and the outcome is the same for each model, we consider the models *nested* and have some extra tools available to compare them.

- I might start by looking at the basic summaries for each model.

```
glance(c2_m4)
```

```
    r.squared adj.r.squared    sigma statistic    p.value df    logLik
1 0.008341404   0.005006229 6.309685   2.50104 0.05818038  4 -2919.873
       AIC      BIC deviance df.residual
1 5849.747 5873.736 35512.42         892
```

```
glance(c2_m1)
```

```
    r.squared adj.r.squared   sigma statistic    p.value df    logLik
1 0.004345169   0.003231461 6.31531  3.901534 0.04854928  2 -2921.675
      AIC      BIC deviance df.residual
1 5849.35 5863.744 35655.53         894
```

- The R^2 is twice as large for the model with `sleephrs`, but still very tiny.
- The *p* value for the global ANOVA test is actually less significant in `c2_m4` than in `c2_m1`.
- Smaller AIC and smaller BIC statistics are more desirable. Here, there's little to choose from, but `c2_m1` is a little better on each standard.
- We might also consider a significance test by looking at an ANOVA model comparison. This is only appropriate because `c2_m1` is nested in `c2_m4`.

```
anova(c2_m4, c2_m1)
```

```
Analysis of Variance Table
```

```
Model 1: bmi ~ female * sleephrs
Model 2: bmi ~ female
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1    892 35512
2    894 35656 -2   -143.11 1.7973 0.1663
```

The addition of the `sleephrs` term picked up 143 in the sum of squares column, at a cost of two degrees of freedom, yielding a p value of 0.166, suggesting that this isn't a significant improvement over the model that just did a t-test on `female`.

2.12 `c2_m5`: What if we add more variables?

We can boost our R^2 a bit, to over 5%, by adding in two new variables, related to whether or not the subject (in the past 30 days) used the internet, and on how many days the subject drank alcoholic beverages.

```
c2_m5 <- lm(bmi ~ female + exerany + sleephrs + internet30 + alcdays,
         data = smartcle2)
summary(c2_m5)
```

```

Call:
lm(formula = bmi ~ female + exerany + sleephrs + internet30 + 
    alcdays, data = smartcle2)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.147  -3.997  -0.856   2.487  35.965 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 30.84066    1.18458  26.035  < 2e-16 ***
female      -1.28801    0.42805  -3.009   0.0027 ** 
exerany     -2.42161    0.49853  -4.858 1.40e-06 ***
sleephrs    -0.14118    0.13988  -1.009   0.3131    
internet30   1.38916    0.54252   2.561   0.0106 *  
alcdays     -0.10460    0.02595  -4.030 6.04e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.174 on 890 degrees of freedom
Multiple R-squared:  0.05258,   Adjusted R-squared:  0.04726 
F-statistic: 9.879 on 5 and 890 DF,  p-value: 3.304e-09
```

1. Here's the ANOVA for this model. What can we study with this?

```
anova(c2_m5)
```

```
Analysis of Variance Table

Response: bmi
            Df Sum Sq Mean Sq F value    Pr(>F)    
female       1    156  155.61  4.0818   0.04365 *  
exerany      1    897  896.93 23.5283 1.453e-06 ***
sleephrs     1     33   32.90  0.8631   0.35313    
```

```
internet30   1    178  178.33  4.6779   0.03082 *
alcdays      1    619  619.26 16.2443 6.044e-05 ***
Residuals  890  33928   38.12
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2. Consider the revised output below. Now what can we study?

```
anova(lm(bmi ~ exerany + internet30 + alcdays + female + sleephrs,
         data = smartcle2))
```

```
Analysis of Variance Table

Response: bmi
            Df Sum Sq Mean Sq F value    Pr(>F)
exerany      1    795  795.46 20.8664 5.618e-06 ***
internet30   1    212  211.95  5.5599 0.0185925 *
alcdays      1    486  486.03 12.7496 0.0003752 ***
female       1    351  350.75  9.2010 0.0024891 **
sleephrs     1     39   38.83  1.0186 0.3131176
Residuals  890  33928   38.12
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3. What does the output below let us conclude?

```
anova(lm(bmi ~ exerany + internet30 + alcdays + female + sleephrs,
         data = smartcle2),
      lm(bmi ~ exerany + female + alcdays,
         data = smartcle2))
```

```
Analysis of Variance Table

Model 1: bmi ~ exerany + internet30 + alcdays + female + sleephrs
Model 2: bmi ~ exerany + female + alcdays
  Res.Df   RSS Df Sum of Sq      F  Pr(>F)
1    890 33928
2    892 34221 -2    -293.2 3.8456 0.02173 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4. What does it mean for the models to be "nested"?

2.13 c2_m6: Would adding self-reported health help?

And we can do even a bit better than that by adding in a multi-categorical measure: self-reported general health.

```
c2_m6 <- lm(bmi ~ female + exerany + sleephrs + internet30 + alcdays + genhealth,
         data = smartcle2)
summary(c2_m6)
```

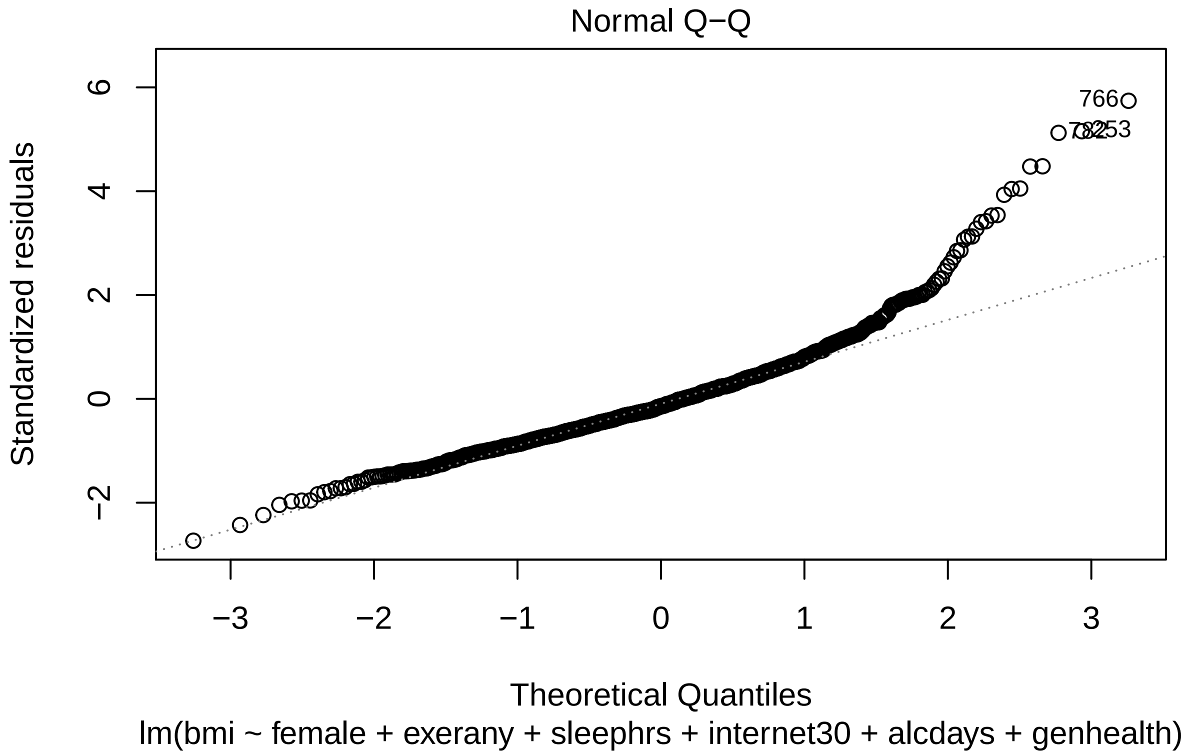
```

Call:
lm(formula = bmi ~ female + exerany + sleephrs + internet30 +
    alcdays + genhealth, data = smartcle2)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-16.331  -3.813  -0.838   2.679  34.166 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)         26.49498    1.31121  20.206  < 2e-16 ***
female              -0.85520    0.41969  -2.038 0.041879 *  
exerany             -1.61968    0.50541  -3.205 0.001400 ** 
sleephrs            -0.12719    0.13613  -0.934 0.350368    
internet30           2.02498    0.53898   3.757 0.000183 ***
alcdays             -0.08431    0.02537  -3.324 0.000925 ***
genhealth2_VeryGood  2.10537    0.59408   3.544 0.000415 ***
genhealth3_Good      4.08245    0.60739   6.721 3.22e-11 ***
genhealth4_Fair      4.99213    0.80178   6.226 7.37e-10 ***
genhealth5_Poor      3.11025    1.12614   2.762 0.005866 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.993 on 886 degrees of freedom
Multiple R-squared:  0.1115,	Adjusted R-squared:  0.1024 
F-statistic: 12.35 on 9 and 886 DF,  p-value: < 2.2e-16
```

1. If Harry and Marty have the same values of `female`, `exerany`, `sleephrs`, `internet30` and `alcdays`, but Harry rates his health as Good, and Marty rates his as Fair, then what is the difference in the predictions? Who is predicted to have a larger BMI, and by how much?

2. What does this normal probability plot of the residuals suggest?

```
plot(c2_m6, which = 2)
```



2.14 c2_m7: What if we added days of work missed?

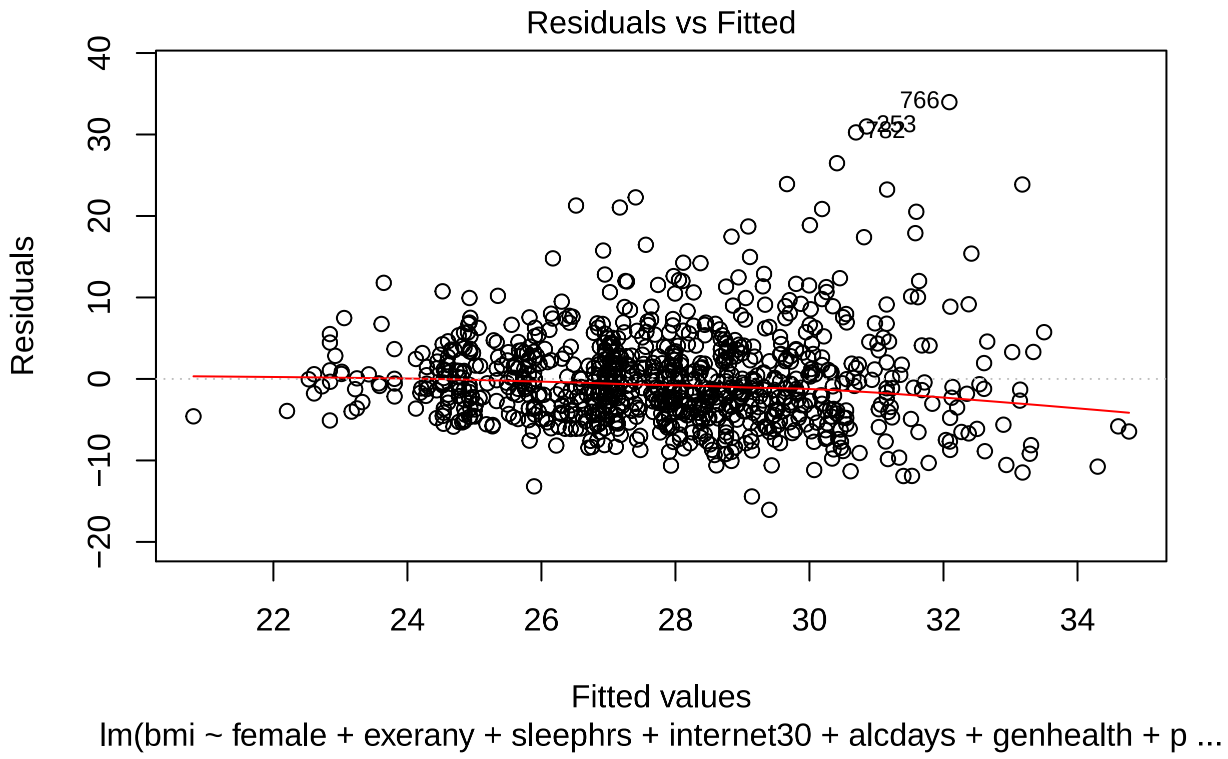
```
c2_m7 <- lm(bmi ~ female + exerany + sleephrs + internet30 + alcdays +
                genhealth + physhealth + menthealth,
         data = smartcle2)
summary(c2_m7)
```

```

Call:
lm(formula = bmi ~ female + exerany + sleephrs + internet30 +
    alcdays + genhealth + physhealth + menthealth, data = smartcle2)

Residuals:
    Min      1Q  Median      3Q     Max
-16.060  -3.804  -0.890   2.794  33.972

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         25.88208    1.31854  19.629  < 2e-16 ***
female              -0.96435    0.41908  -2.301 0.021616 *
exerany             -1.43171    0.50635  -2.828 0.004797 **
sleephrs            -0.08033    0.13624  -0.590 0.555583
internet30           2.00267    0.53759   3.725 0.000207 ***
alcdays             -0.07997    0.02528  -3.163 0.001614 **
```

```
genhealth2_VeryGood  2.09533    0.59238   3.537 0.000425 ***
genhealth3_Good      3.90949    0.60788   6.431 2.07e-10 ***
genhealth4_Fair      4.27152    0.83986   5.086 4.47e-07 ***
genhealth5_Poor      1.26021    1.31556   0.958 0.338361
physhealth           0.06088    0.03005   2.026 0.043064 *
menthealth           0.06636    0.03177   2.089 0.037021 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.964 on 884 degrees of freedom
Multiple R-squared:  0.1219,    Adjusted R-squared:  0.111
F-statistic: 11.16 on 11 and 884 DF,  p-value: < 2.2e-16
```

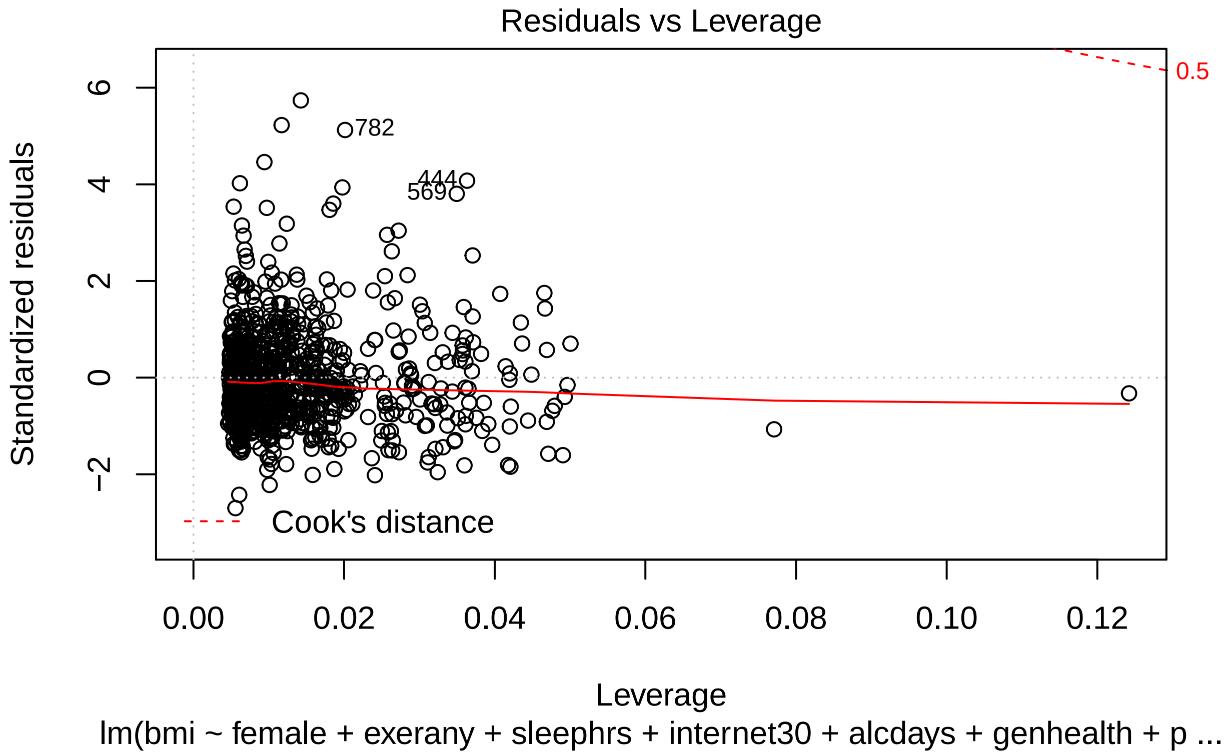
1. How do the assumptions behind this model look?

```
plot(c2_m7, which = 1)
```



2. What can we conclude from the plot below?

```
plot(c2_m7, which = 5)
```



2.15 Key Regression Assumptions for Building Effective Prediction Models

1. Validity - the data you are analyzing should map to the research question you are trying to answer.
    - The outcome should accurately reflect the phenomenon of interest.
    - The model should include all relevant predictors. (It can be difficult to decide which predictors are necessary, and what to do with predictors that have large standard errors.)
    - The model should generalize to all of the cases to which it will be applied.
    - Can the available data answer our question reliably?
2. Additivity and linearity - most important assumption of a regression model is that its deterministic component is a linear function of the predictors. We often think about transformations in this setting.
3. Independence of errors - errors from the prediction line are independent of each other
4. Equal variance of errors - if this is violated, we can more efficiently estimate paramaters using *weighted least squares* approaches, where each point is weighted inversely proportional to its variance, but this doesn't affect the coefficients much, if at all.
5. Normality of errors - not generally important for estimating the regression line

2.16 Making Predictions with a Linear Regression Model

Recall model 4, which yields predictions for body mass index on the basis of the main effects of sex (`female`) and hours of sleep (`sleephrs`) and their interaction.

```
c2_m4
```

```

Call:
lm(formula = bmi ~ female * sleephrs, data = smartcle2)

Coefficients:
    (Intercept)           female         sleephrs  female:sleephrs  
        27.2661           2.5263           0.1569          -0.4797  
```

2.16.1 Fitting an Individual Prediction and 95% Prediction Interval

What do we predict for the `bmi` of a subject who is `female` and gets 8 hours of sleep per night?

```
c2_new1 <- data_frame(female = 1, sleephrs = 8)
predict(c2_m4, newdata = c2_new1, interval = "prediction", level = 0.95)
```

```
       fit     lwr     upr
1 27.21065 14.8107 39.6106
```

The predicted `bmi` for this new subject is 27.61. The prediction interval shows the bounds of a 95% uncertainty interval for a predicted `bmi` for an individual female subject who gets 8 hours of sleep on average per evening. From the `predict` function applied to a linear model, we can get the prediction intervals for any new data points in this manner.

2.16.2 Confidence Interval for an Average Prediction

- What do we predict for the **average body mass index of a population of subjects** who are female and sleep for 8 hours?

```
predict(c2_m4, newdata = c2_new1, interval = "confidence", level = 0.95)
```

```
       fit      lwr      upr
1 27.21065 26.57328 27.84801
```

- How does this result compare to the prediction interval?

2.16.3 Fitting Multiple Individual Predictions to New Data

- How does our prediction change for a respondent if they instead get 7, or 9 hours of sleep? What if they are male, instead of female?

```
c2_new2 <- data_frame(subjectid = 1001:1006, female = c(1, 1, 1, 0, 0, 0), sleephrs = c(7, 8, 9, 7, 8, 9
pred2 <- predict(c2_m4, newdata = c2_new2, interval = "prediction", level = 0.95) %>% tbl_df

result2 <- bind_cols(c2_new2, pred2)
result2
```

```
# A tibble: 6 x 6
  subjectid female sleephrs   fit   lwr   upr
      <int>  <dbl>    <dbl> <dbl> <dbl> <dbl>
1      1001   1.00     7.00  27.5  15.1  39.9
2      1002   1.00     8.00  27.2  14.8  39.6
3      1003   1.00     9.00  26.9  14.5  39.3
4      1004   0        7.00  28.4  16.0  40.8
5      1005   0        8.00  28.5  16.1  40.9
```

```
6      1006   0        9.00  28.7  16.2  41.1
```

The `result2` tibble contains predictions for each scenario.

- Which has a bigger impact on these predictions and prediction intervals? A one category change in `female` or a one hour change in `sleephrs`?

2.16.4 Simulation to represent predictive uncertainty in Model 4

Suppose we want to predict the `bmi` of a female subject who sleeps for eight hours per night. As we have seen, we can do this automatically for a linear model like this one, using the `predict` function applied to the linear model, but a simulation prediction can also be done. Recall the detail of `c2_m4`:

```
c2_m4
```

```

Call:
lm(formula = bmi ~ female * sleephrs, data = smartcle2)

Coefficients:
    (Intercept)           female         sleephrs  female:sleephrs  
        27.2661           2.5263           0.1569          -0.4797  
```

```
glance(c2_m4)
```

```
    r.squared adj.r.squared    sigma statistic    p.value df    logLik
1 0.008341404   0.005006229 6.309685   2.50104 0.05818038  4 -2919.873
       AIC      BIC deviance df.residual
1 5849.747 5873.736 35512.42         892
```

We see that the residual standard error for our `bmi` predictions with this model is 6.31.

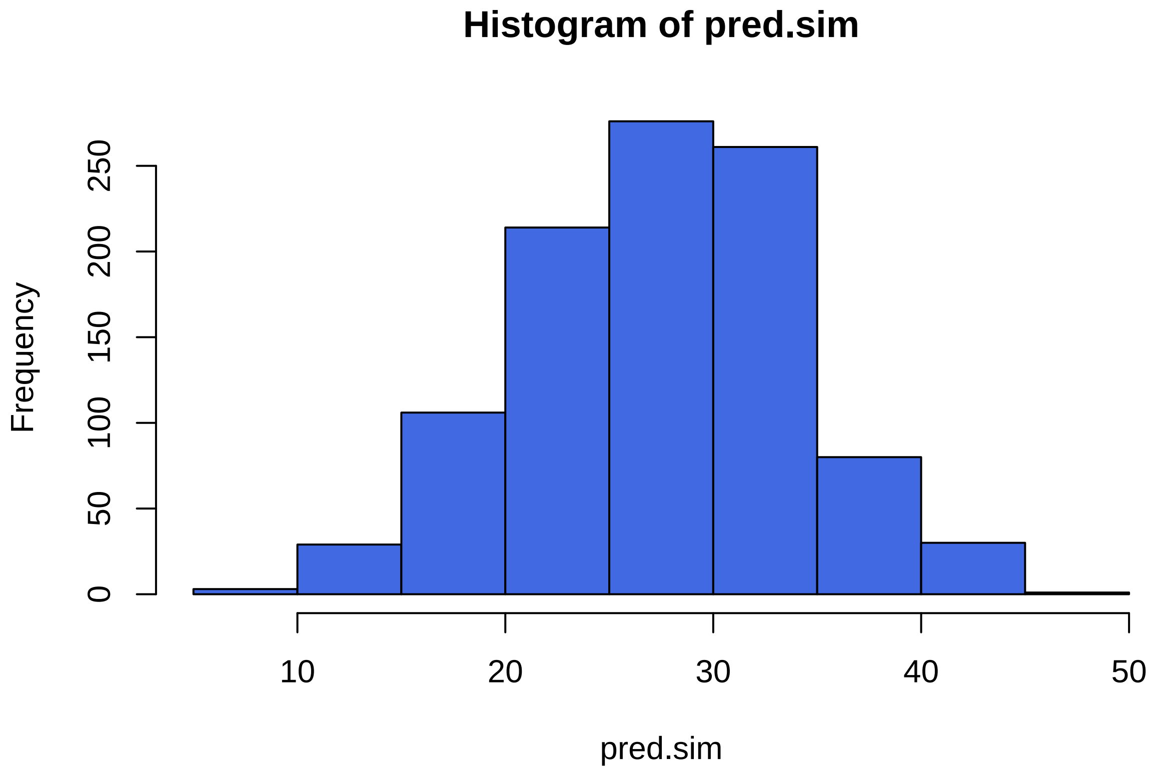
For a female respondent sleeping eight hours, recall that our point estimate (predicted value) of `bmi` is 27.21

```
predict(c2_m4, newdata = c2_new1, interval = "prediction", level = 0.95)
```

```
       fit     lwr     upr
1 27.21065 14.8107 39.6106
```

The standard deviation is 6.31, so we could summarize the predictive distribution with a command that tells R to draw 1000 random numbers from a normal distribution with mean 27.21 and standard deviation 6.31. Let's summarize that and get a quick picture.

```
set.seed(432094)
pred.sim <- rnorm(1000, 27.21, 6.31)
hist(pred.sim, col = "royalblue")
```



```
mean(pred.sim)
```

```
[1] 27.41856
```

```
quantile(pred.sim, c(0.025, 0.975))
```

```
    2.5%    97.5% 
14.48487 40.16778 
```

How do these results compare to the prediction interval of (14.81, 39.61) that we generated earlier?

2.17 Centering the model

Our model `c2_m4` has four predictors (the constant, `sleephrs`, `female` and their interaction) but just two inputs (`female` and `sleephrs`.) If we **center** the quantitative input `sleephrs` before building the model, we get a more interpretable interaction term.

```
smartcle2_c <- smartcle2 %>%
    mutate(sleephrs_c = sleephrs - mean(sleephrs))

c2_m4_c <- lm(bmi ~ female * sleephrs_c, data = smartcle2_c)

summary(c2_m4_c)
```

```

Call:
lm(formula = bmi ~ female * sleephrs_c, data = smartcle2_c)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-15.498  -4.179  -1.035   2.830  38.204 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)        28.3681     0.3274  86.658   <2e-16 ***
female             -0.8420     0.4280  -1.967   0.0495 *  
sleephrs_c          0.1569     0.2294   0.684   0.4940    
female:sleephrs_c  -0.4797     0.2931  -1.636   0.1021    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.31 on 892 degrees of freedom
Multiple R-squared:  0.008341,	Adjusted R-squared:  0.005006 
F-statistic: 2.501 on 3 and 892 DF,  p-value: 0.05818
```

What has changed as compared to the original `c2_m4`?

- Our original model was `bmi` = 27.26 + 2.53 `female` + 0.16 `sleephrs` - 0.48 `female` x `sleephrs`
- Our new model is `bmi` = 28.37 - 0.84 `female` + 0.16 centered `sleephrs` - 0.48 `female` x centered `sleephrs`.

So our new model on centered data is:

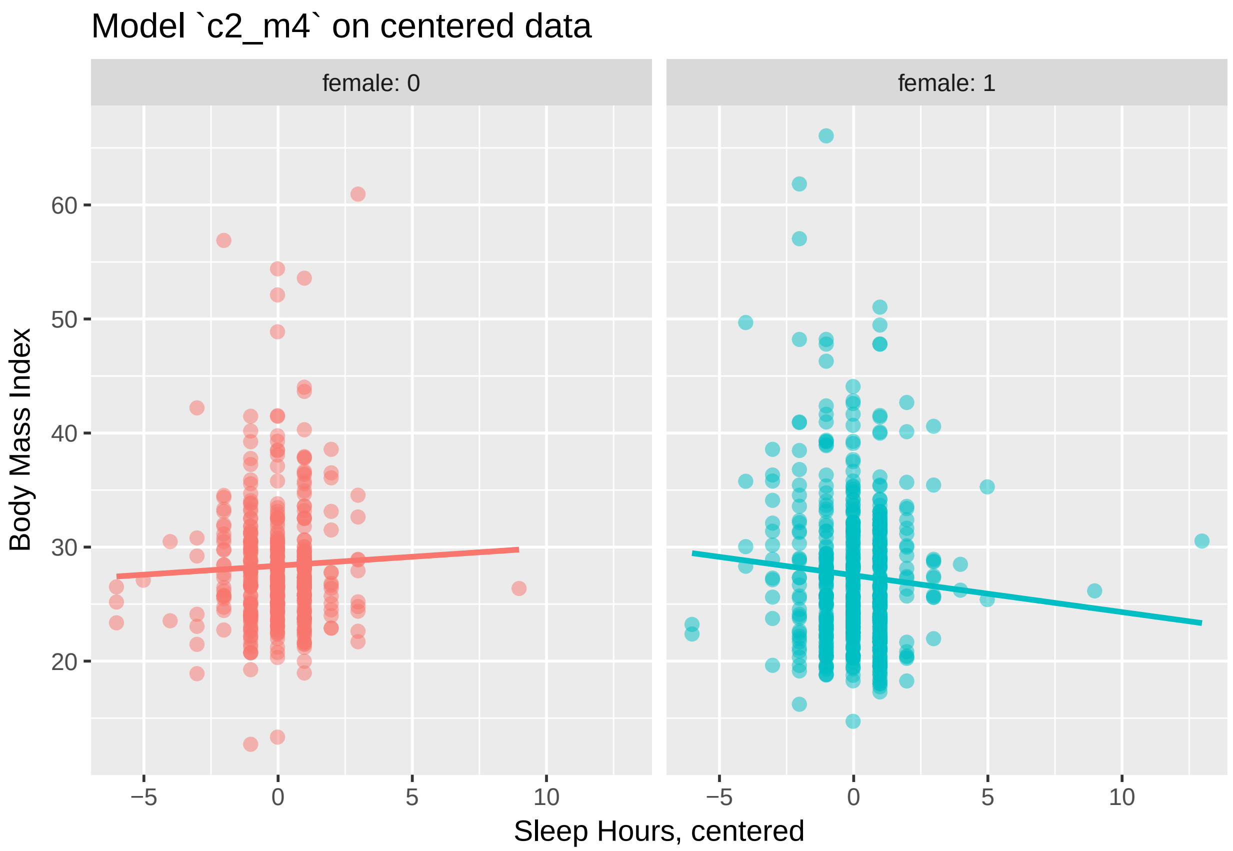
- 28.37 + 0.16 centered `sleephrs_c` for male subjects, and
- (28.37 - 0.84) + (0.16 - 0.48) centered `sleephrs_c`, or 27.53 - 0.32 centered `sleephrs_c` for female subjects.

In our new (centered `sleephrs_c`) model,

- the main effect of `female` now corresponds to a predictive difference (female - male) in `bmi` with `sleephrs` at its mean value, 7.02 hours,
- the intercept term is now the predicted `bmi` for a male respondent who sleeps an average number of hours, and
- the product term corresponds to the change in the slope of centered `sleephrs_c` on `bmi` for a female rather than a male subject, while
- the residual standard deviation and the R-squared values remain unchanged from the model before centering.

2.17.1 Plot of Model 4 on Centered `sleephrs`: `c2_m4_c`

```r
ggplot(smartcle2_c, aes(x = sleephrs_c, y = bmi, group = female, col = factor(female))) +
    geom_point(alpha = 0.5, size = 2) +
    geom_smooth(method = "lm", se = FALSE) +
    guides(color = FALSE) +
    labs(x = "Sleep Hours, centered", y = "Body Mass Index",
         title = "Model `c2_m4` on centered data") +
    facet_wrap(~ female, labeller = label_both)
```



2.18 Rescaling an input by subtracting the mean and dividing by 2 standard deviations

Centering helped us interpret the main effects in the regression, but it still leaves a scaling problem.

- The `female` coefficient estimate is much larger than that of `sleephrs`, but this is misleading, considering that we are comparing the complete change in one variable (sex = female or not) to a 1-hour change in average sleep.
- Gelman and Hill (2007) recommend all continuous predictors be scaled by dividing by 2 standard deviations, so that:
  - a 1-unit change in the rescaled predictor corresponds to a change from 1 standard deviation below the mean, to 1 standard deviation above.
  - an unscaled binary (1/0) predictor with 50% probability of occurring will be exactly comparable to a rescaled continuous predictor done in this way.

```
smartcle2_rescale <- smartcle2 %>%
    mutate(sleephrs_z = (sleephrs - mean(sleephrs))/(2*sd(sleephrs)))
```

2.18.1 Refitting model `c2_m4` to the rescaled data

```
c2_m4_z <- lm(bmi ~ female * sleephrs_z, data = smartcle2_rescale)

summary(c2_m4_z)
```

```
Call:
lm(formula = bmi ~ female * sleephrs_z, data = smartcle2_rescale)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.498  -4.179  -1.035   2.830  38.204 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)        28.3681     0.3274  86.658   <2e-16 ***
female             -0.8420     0.4280  -1.967   0.0495 *  
sleephrs_z          0.4637     0.6778   0.684   0.4940    
female:sleephrs_z  -1.4173     0.8661  -1.636   0.1021    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.31 on 892 degrees of freedom
Multiple R-squared:  0.008341,  Adjusted R-squared:  0.005006 
F-statistic: 2.501 on 3 and 892 DF,  p-value: 0.05818
```

2.18.2 Interpreting the model on rescaled data

What has changed as compared to the original `c2_m4`?

- Our original model was `bmi` = 27.26 + 2.53 `female` + 0.16 `sleephrs` - 0.48 `female` x `sleephrs`
- Our model on centered `sleephrs` was `bmi` = 28.37 - 0.84 `female` + 0.16 centered `sleephrs_c` - 0.48 `female` x centered `sleephrs_c`.
- Our new model on rescaled `sleephrs` is `bmi` = 28.37 - 0.84 `female` + 0.46 rescaled `sleephrs_z` - 1.42 `female` x rescaled `sleephrs_z`.

So our rescaled model is:

- 28.37 + 0.46 rescaled `sleephrs_z` for male subjects, and
- (28.37 - 0.84) + (0.46 - 1.42) rescaled `sleephrs_z`, or 27.53 - 0.96 rescaled `sleephrs_z` for female subjects.

In this new rescaled (`sleephrs_z`) model, then,

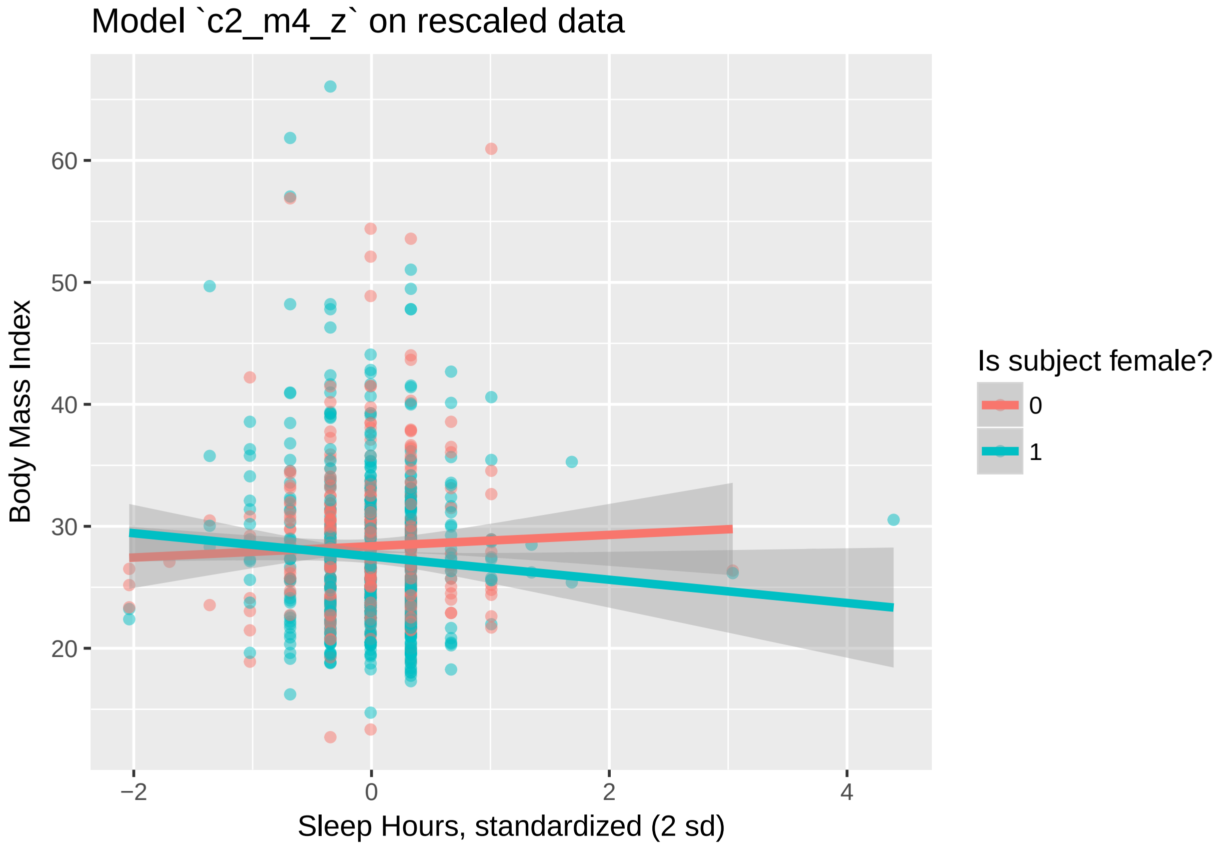
- the main effect of `female`, -0.84, still corresponds to a predictive difference (female - male) in `bmi` with `sleephrs` at its mean value, 7.02 hours,
- the intercept term is still the predicted `bmi` for a male respondent who sleeps an average number of hours, and
- the residual standard deviation and the R-squared values remain unchanged,

as before, but now we also have that:

- the coefficient of `sleephrs_z` indicates the predictive difference in `bmi` associated with a change in `sleephrs` of 2 standard deviations (from one standard deviation below the mean of 7.02 to one standard deviation above 7.02.)
  - Since the standard deviation of `sleephrs` is 1.48, this corresponds to a change from 5.54 hours per night to 8.50 hours per night.
- the coefficient of the product term (-1.42) corresponds to the change in the coefficient of `sleephrs_z` for females as compared to males.

2.18.3 Plot of model on rescaled data

```
ggplot(smartcle2_rescale, aes(x = sleephrs_z, y = bmi,
                              group = female, col = factor(female))) +
    geom_point(alpha = 0.5) +
    geom_smooth(method = "lm", size = 1.5) +
    scale_color_discrete(name = "Is subject female?") +
    labs(x = "Sleep Hours, standardized (2 sd)", y = "Body Mass Index",
         title = "Model `c2_m4_z` on rescaled data")
```



Chapter 3

Analysis of Variance and Analysis of Covariance

3.1 The bonding data: A Designed Dental Experiment

The `bonding` data describe a designed experiment into the properties of four different resin types (`resin` = A, B, C, D) and two different curing light sources (`light` = Halogen, LED) as they relate to the resulting bonding strength (measured in MPa[1]) on the surface of teeth. The source is Kim (2014).

The experiment involved making measurements of bonding strength under a total of 80 experimental setups, or runs, with 10 runs completed at each of the eight combinations of a light source and a resin type. The data are gathered in the `bonding.csv` file.

```
bonding
```

```
# A tibble: 80 x 4
   run_ID light   resin strength
   <fct>  <fct>   <fct>    <dbl>
 1 R101   LED     B         12.8
 2 R102   Halogen B         22.2
 3 R103   Halogen B         24.6
 4 R104   LED     A         17.0
 5 R105   LED     C         32.2
 6 R106   Halogen B         27.1
 7 R107   LED     A         23.4
 8 R108   Halogen A         23.5
 9 R109   Halogen D         37.3
10 R110   Halogen A         19.7
# ... with 70 more rows
```

3.2 A One-Factor Analysis of Variance

Suppose we are interested in the distribution of the `strength` values for the four different types of `resin`.

```
bonding %>% group_by(resin) %>% summarize(n = n(), mean(strength), median(strength))
```

```
# A tibble: 4 x 4
```

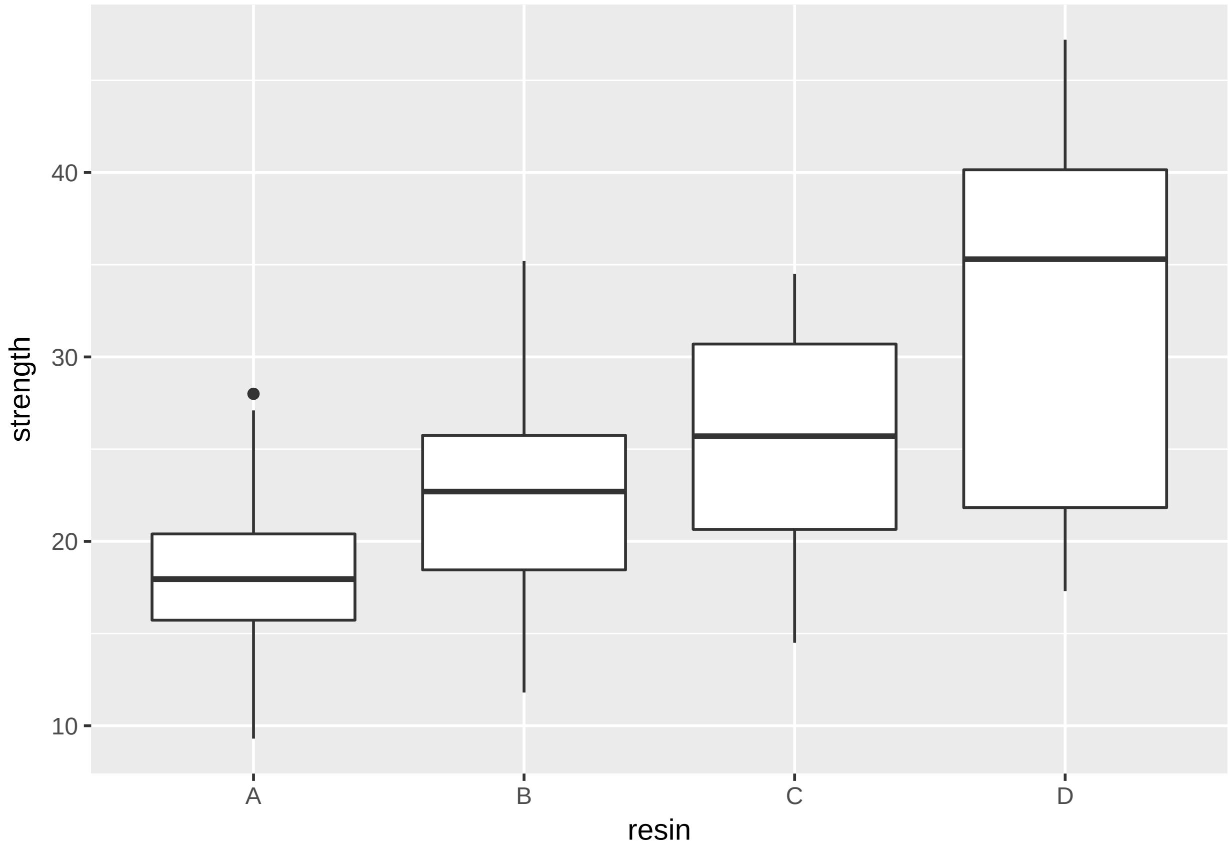
[1] The MPa is defined as the failure load (in Newtons) divided by the entire bonded area, in mm^2.

```
  resin     n `mean(strength)` `median(strength)`
  <fct> <int>            <dbl>              <dbl>
1 A        20             18.4               18.0
2 B        20             22.2               22.7
3 C        20             25.2               25.7
4 D        20             32.1               35.3
```

I'd begin serious work with a plot.

3.2.1 Look at the Data!

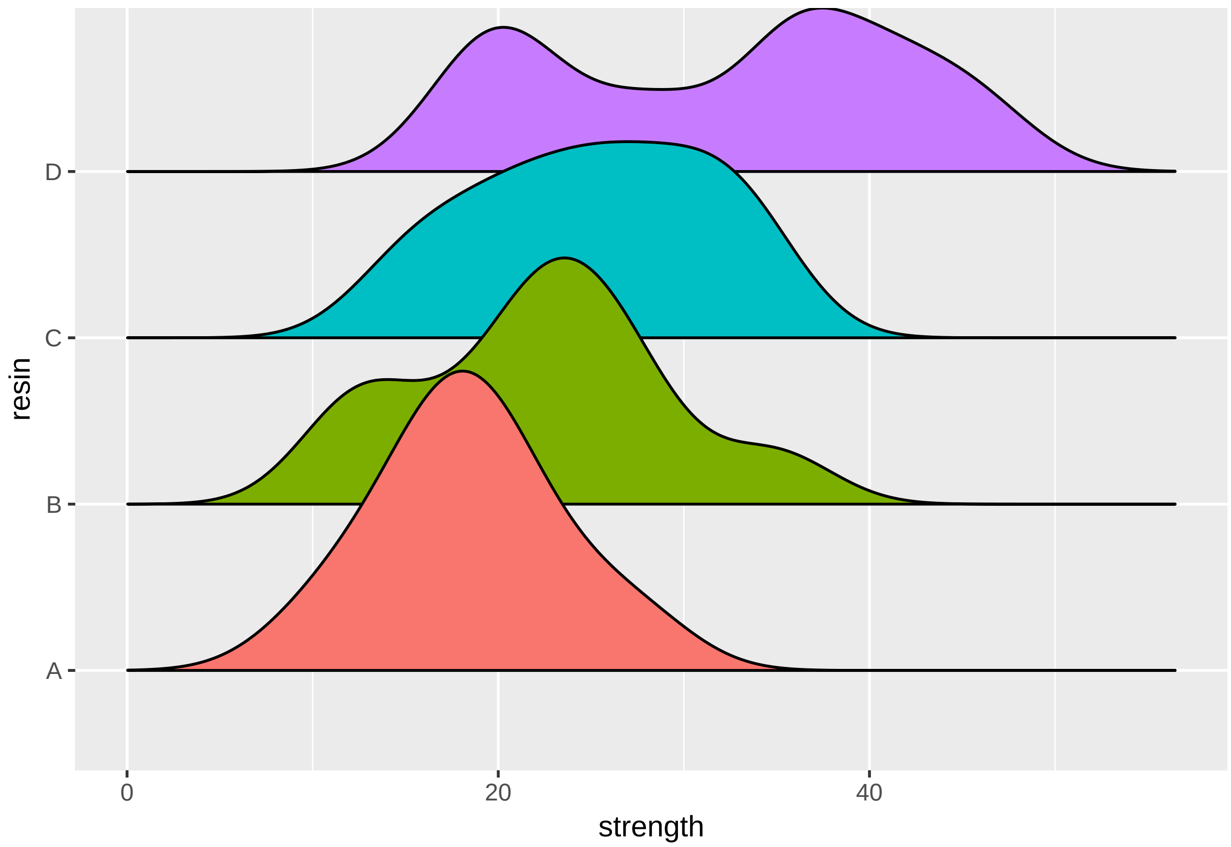
```
ggplot(bonding, aes(x = resin, y = strength)) +
    geom_boxplot()
```



Another good plot for this purpose is a ridgeline plot.

```
ggplot(bonding, aes(x = strength, y = resin, fill = resin)) +
    geom_density_ridges2() +
    guides(fill = FALSE)
```

```
Picking joint bandwidth of 3.09
```



3.2.2 Table of Summary Statistics

With the small size of this experiment ($n = 20$ for each `resin` type), graphical summaries may not perform as well as they often do. We'll also produce a quick table of summary statistics for `strength` within each `resin` type, with the `skim()` function.

```
bonding %>% group_by(resin) %>% skim(strength)
```

```
Skim summary statistics
 n obs: 80
 n variables: 4
 group variables: resin

Variable type: numeric
 resin variable missing complete  n  mean   sd   p0   p25 median   p75
     A strength       0       20 20 18.41 4.81  9.3 15.73  17.95 20.4
     B strength       0       20 20 22.23 6.75 11.8 18.45  22.7  25.75
     C strength       0       20 20 25.16 6.33 14.5 20.65  25.7  30.7
     D strength       0       20 20 32.08 9.74 17.3 21.82  35.3  40.15
 p100
 28
 35.2
 34.5
 47.2
```

Since the means and medians are fairly close, and the distributions (with the possible exception of `resin` D) are reasonably well approximated by the Normal, I'll fit an ANOVA model.

```
anova(lm(strength ~ resin, data = bonding))
```

```
Analysis of Variance Table

Response: strength
          Df Sum Sq Mean Sq F value   Pr(>F)
resin      3 1999.7  666.57  13.107 5.52e-07 ***
Residuals 76 3865.2   50.86
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

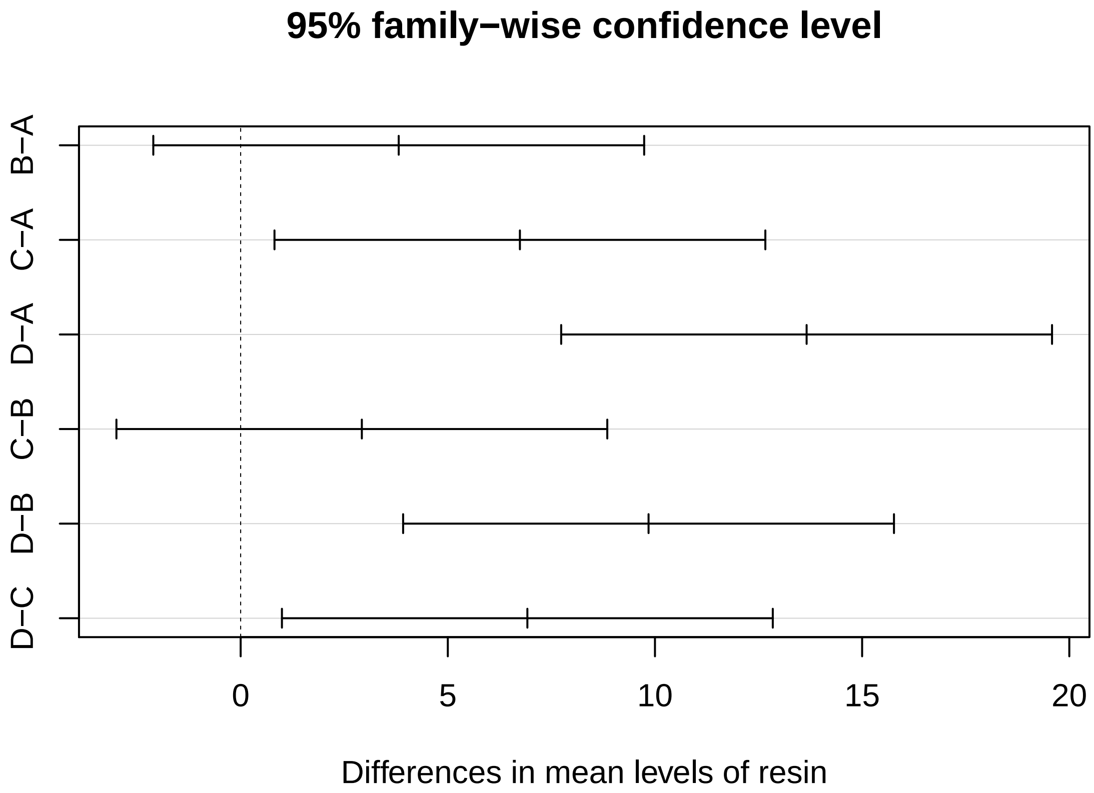
It appears that the `resin` types have a significant association with mean `strength` of the bonds. Can we identify which `resin` types have generally higher or lower `strength`?

```
TukeyHSD(aov(lm(strength ~ resin, data = bonding)))
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = lm(strength ~ resin, data = bonding))

$resin
      diff        lwr       upr     p adj
B-A  3.815 -2.1088676  9.738868 0.3351635
C-A  6.740  0.8161324 12.663868 0.0193344
D-A 13.660  7.7361324 19.583868 0.0000003
C-B  2.925 -2.9988676  8.848868 0.5676635
D-B  9.845  3.9211324 15.768868 0.0002276
D-C  6.920  0.9961324 12.843868 0.0154615
```

Based on these confidence intervals (which have a family-wise 95% confidence level), we see that D is associated with significantly larger mean `strength` than A or B or C, and that C is also associated with significantly larger mean `strength` than A. This may be easier to see in a plot of these confidence intervals.

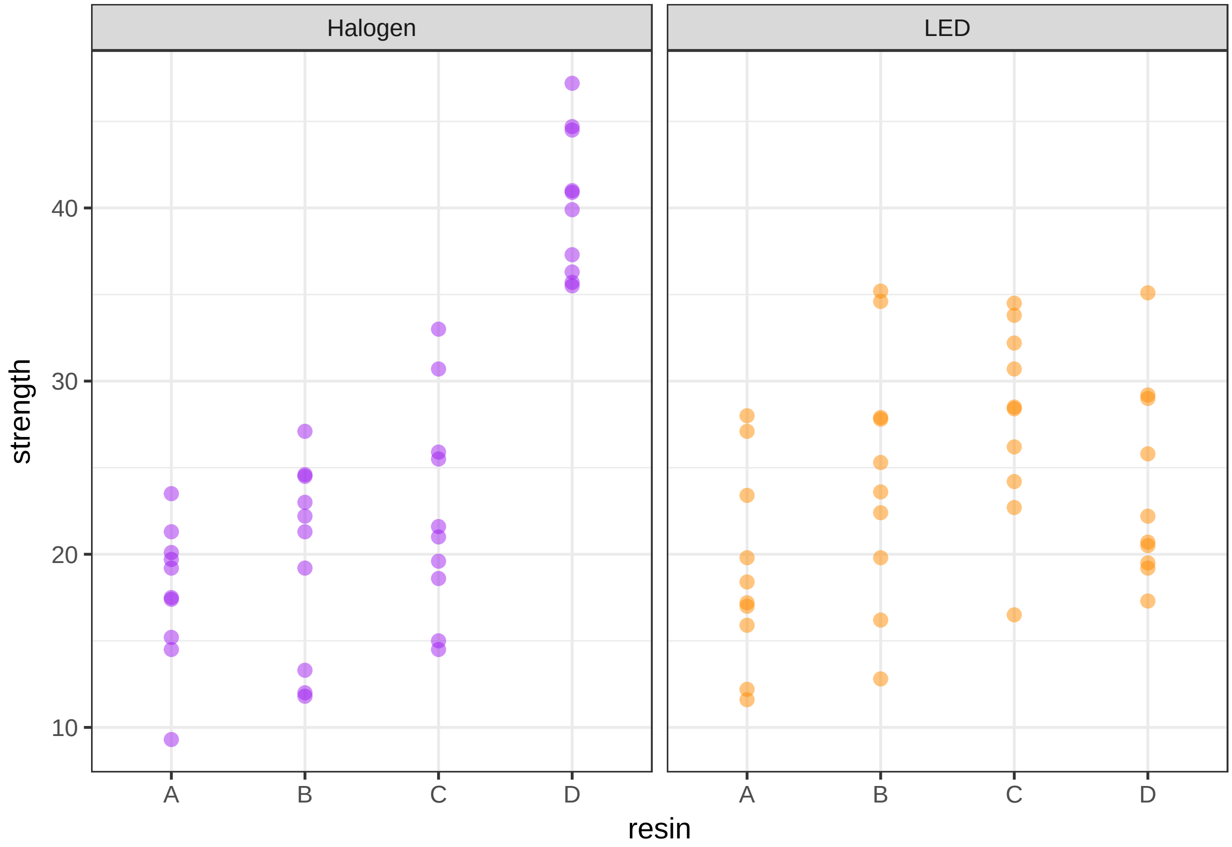
```
plot(TukeyHSD(aov(lm(strength ~ resin, data = bonding))))
```



3.3 A Two-Way ANOVA: Looking at Two Factors

Now, we'll now add consideration of the `light` source into our study. We can look at the distribution of the `strength` values at the combinations of both `light` and `resin`, with a plot like this one...

```
ggplot(bonding, aes(x = resin, y = strength, color = light)) +
    geom_point(size = 2, alpha = 0.5) +
    facet_wrap(~ light) +
    guides(color = FALSE) +
    scale_color_manual(values = c("purple", "darkorange")) +
    theme_bw()
```



3.4 A Means Plot (with standard deviations) to check for interaction

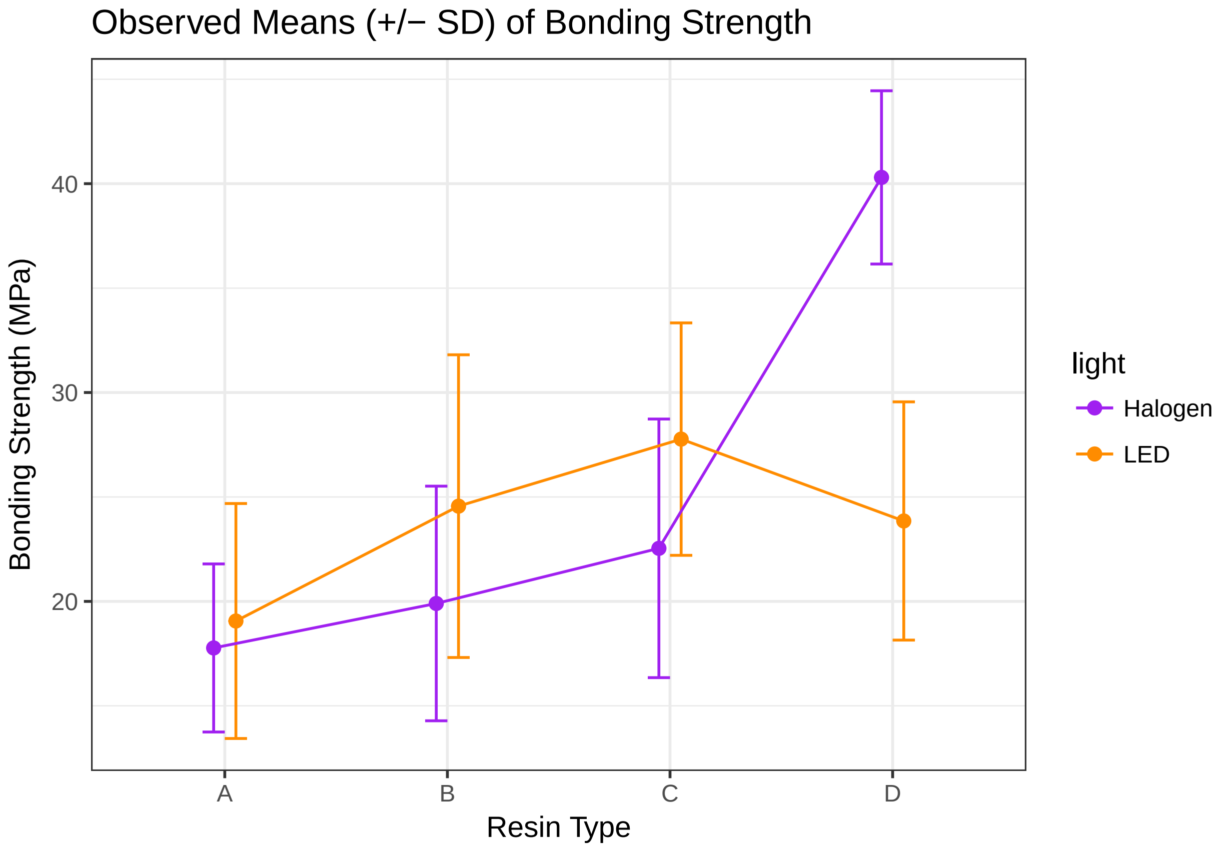
Sometimes, we'll instead look at a plot simply of the means (and, often, the standard deviations) of `strength` at each combination of `light` and `resin`. We'll start by building up a data set with the summaries we want to plot.

```
bond.sum <- bonding %>%
    group_by(resin, light) %>%
    summarize(mean.str = mean(strength), sd.str = sd(strength))

bond.sum
```

```
# A tibble: 8 x 4
# Groups: resin [?]
  resin light   mean.str sd.str
  <fct> <fct>      <dbl>  <dbl>
1 A     Halogen     17.8   4.02
2 A     LED         19.1   5.63
3 B     Halogen     19.9   5.62
4 B     LED         24.6   7.25
5 C     Halogen     22.5   6.19
6 C     LED         27.8   5.56
7 D     Halogen     40.3   4.15
8 D     LED         23.8   5.70
```

Now, we'll use this new data set to plot the means and standard deviations of `strength` at each combination of `resin` and `light`.

```
## The error bars will overlap unless we adjust the position.
pd <- position_dodge(0.2) # move them .1 to the left and right

ggplot(bond.sum, aes(x = resin, y = mean.str, col = light)) +
    geom_errorbar(aes(ymin = mean.str - sd.str,
                      ymax = mean.str + sd.str),
                  width = 0.2, position = pd) +
    geom_point(size = 2, position = pd) +
    geom_line(aes(group = light), position = pd) +
    scale_color_manual(values = c("purple", "darkorange")) +
    theme_bw() +
    labs(y = "Bonding Strength (MPa)", x = "Resin Type",
         title = "Observed Means (+/- SD) of Bonding Strength")
```



Is there evidence of a meaningful interaction between the resin type and the `light` source on the bonding strength in this plot?

- Sure. A meaningful interaction just means that the strength associated with different `resin` types depends on the `light` source.
  - With LED `light`, it appears that `resin` C leads to the strongest bonding strength.
  - With Halogen `light`, though, it seems that `resin` D is substantially stronger.
- Note that the lines we see here connecting the `light` sources aren't in parallel (as they would be if we had zero interaction between `resin` and `light`), but rather, they cross.

3.4.1 Skimming the data after grouping by resin and light

We might want to look at a numerical summary of the strengths within these groups, too.

```
bonding %>%
    group_by(resin, light) %>%
    skim(strength)
```

```
Skim summary statistics
 n obs: 80
 n variables: 4
 group variables: resin, light

Variable type: numeric
 resin   light variable missing complete  n  mean   sd   p0   p25 median
     A Halogen strength       0       10 10 17.77 4.02  9.3 15.75  18.35
     A     LED strength       0       10 10 19.06 5.63 11.6 16.18  17.8
     B Halogen strength       0       10 10 19.9  5.62 11.8 14.78  21.75
     B     LED strength       0       10 10 24.56 7.25 12.8 20.45  24.45
     C Halogen strength       0       10 10 22.54 6.19 14.5 18.85  21.3
     C     LED strength       0       10 10 27.77 5.56 16.5 24.7   28.45
     D Halogen strength       0       10 10 40.3  4.15 35.5 36.55  40.4
     D     LED strength       0       10 10 23.85 5.7  17.3 19.75  21.45
   p75 p100
 20    23.5
 22.5  28
 24.12 27.1
 27.87 35.2
 25.8  33
 31.83 34.5
 43.62 47.2
 28.2  35.1
```

3.5 Fitting the Two-Way ANOVA model with Interaction

```
c3_m1 <- lm(strength ~ resin * light, data = bonding)

summary(c3_m1)
```

```

Call:
lm(formula = strength ~ resin * light, data = bonding)

Residuals:
    Min      1Q  Median      3Q     Max
-11.760  -3.663  -0.320   3.697  11.250

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       17.770      1.771  10.033 2.57e-15 ***
resinB             2.130      2.505   0.850   0.3979
resinC             4.770      2.505   1.904   0.0609 .
resinD            22.530      2.505   8.995 2.13e-13 ***
```

```
lightLED          1.290      2.505   0.515   0.6081
resinB:lightLED    3.370      3.542   0.951   0.3446
resinC:lightLED    3.940      3.542   1.112   0.2697
resinD:lightLED  -17.740      3.542  -5.008 3.78e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.601 on 72 degrees of freedom
Multiple R-squared:  0.6149,    Adjusted R-squared:  0.5775
F-statistic: 16.42 on 7 and 72 DF,  p-value: 9.801e-13
```

3.5.1 The ANOVA table for our model

In a two-way ANOVA model, we begin by assessing the interaction term. If it's important, then our best model is the model including the interaction. If it's not important, we will often move on to consider a new model, fit without an interaction.

The ANOVA table is especially helpful in this case, because it lets us look specifically at the interaction effect.

```
anova(c3_m1)
```

```
Analysis of Variance Table

Response: strength
            Df  Sum Sq Mean Sq F value    Pr(>F)
resin        3 1999.72  666.57 21.2499 5.792e-10 ***
light        1   34.72   34.72  1.1067    0.2963
resin:light  3 1571.96  523.99 16.7043 2.457e-08 ***
Residuals   72 2258.52   31.37
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.5.2 Is the interaction important?

In this case, the interaction:

- is evident in the means plot, and
- is highly statistically significant, and
- accounts for a sizeable fraction (27%) of the overall variation

$$\eta^2_{interaction} = \frac{\text{SS(resin:light)}}{SS(Total)} = \frac{1571.96}{1999.72 + 34.72 + 1571.96 + 2258.52} = 0.268$$

If the interaction were *either* large or significant we would be inclined to keep it in the model. In this case, it's both, so there's no real reason to remove it.

3.5.3 Interpreting the Interaction

Recall the model equation, which is:

```
c3_m1
```

```
Call:
lm(formula = strength ~ resin * light, data = bonding)

Coefficients:
    (Intercept)            resinB            resinC            resinD  
          17.77              2.13              4.77             22.53  
       lightLED  resinB:lightLED  resinC:lightLED  resinD:lightLED  
           1.29              3.37              3.94            -17.74  
```

so we have:

$$strength = 17.77 + 2.13resinB + 4.77resinC + 22.53resinD + 1.29lightLED + 3.37resinB*lightLED + 3.94resinC*lightLED -$$

So, if `light` = Halogen, our equation is:

$$strength = 17.77 + 2.13resinB + 4.77resinC + 22.53resinD$$

And if `light` = LED, our equation is:

$$strength = 19.06 + 5.50resinB + 8.71resinC + 4.79resinD$$

Note that both the intercept and the slopes change as a result of the interaction. The model yields a different prediction for every possible combination of a `resin` type and a `light` source.

3.6 Comparing Individual Combinations of `resin` and `light`

To make comparisons between individual combinations of a `resin` type and a `light` source, using something like Tukey's HSD approach for multiple comparisons, we first refit the model using the `aov` structure, rather than `lm`.

```
c3m1_aov <- aov(strength ~ resin * light, data = bonding)

summary(c3m1_aov)
```

```
             Df Sum Sq Mean Sq F value   Pr(>F)    
resin         3 1999.7   666.6  21.250 5.79e-10 ***
light         1   34.7    34.7   1.107    0.296    
resin:light   3 1572.0   524.0  16.704 2.46e-08 ***
Residuals    72 2258.5    31.4                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And now, we can obtain Tukey HSD comparisons (which will maintain an overall 95% family-wise confidence level) across the `resin` types, the `light` sources, and the combinations, with the TukeyHSD command. This approach is only completely appropriate if these comparisons are pre-planned, and if the design is balanced (as this is, with the same sample size for each combination of a `light` source and `resin` type.)

```
TukeyHSD(c3m1_aov)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level
```

```
Fit: aov(formula = strength ~ resin * light, data = bonding)

$resin
      diff       lwr       upr     p adj
B-A  3.815 -0.843129  8.473129 0.1461960
C-A  6.740  2.081871 11.398129 0.0016436
D-A 13.660  9.001871 18.318129 0.0000000
C-B  2.925 -1.733129  7.583129 0.3568373
D-B  9.845  5.186871 14.503129 0.0000026
D-C  6.920  2.261871 11.578129 0.0011731

$light
               diff       lwr      upr     p adj
LED-Halogen -1.3175 -3.814042 1.179042 0.2963128

$`resin:light`
                      diff          lwr        upr     p adj
B:Halogen-A:Halogen   2.13  -5.68928258   9.949283 0.9893515
C:Halogen-A:Halogen   4.77  -3.04928258  12.589283 0.5525230
D:Halogen-A:Halogen  22.53  14.71071742  30.349283 0.0000000
A:LED-A:Halogen       1.29  -6.52928258   9.109283 0.9995485
B:LED-A:Halogen       6.79  -1.02928258  14.609283 0.1361092
C:LED-A:Halogen      10.00   2.18071742  17.819283 0.0037074
D:LED-A:Halogen       6.08  -1.73928258  13.899283 0.2443200
C:Halogen-B:Halogen   2.64  -5.17928258  10.459283 0.9640100
D:Halogen-B:Halogen  20.40  12.58071742  28.219283 0.0000000
A:LED-B:Halogen      -0.84  -8.65928258   6.979283 0.9999747
B:LED-B:Halogen       4.66  -3.15928258  12.479283 0.5818695
C:LED-B:Halogen       7.87   0.05071742  15.689283 0.0473914
D:LED-B:Halogen       3.95  -3.86928258  11.769283 0.7621860
D:Halogen-C:Halogen  17.76   9.94071742  25.579283 0.0000000
A:LED-C:Halogen      -3.48 -11.29928258   4.339283 0.8591455
B:LED-C:Halogen       2.02  -5.79928258   9.839283 0.9922412
C:LED-C:Halogen       5.23  -2.58928258  13.049283 0.4323859
D:LED-C:Halogen       1.31  -6.50928258   9.129283 0.9995004
A:LED-D:Halogen     -21.24 -29.05928258 -13.420717 0.0000000
B:LED-D:Halogen     -15.74 -23.55928258  -7.920717 0.0000006
C:LED-D:Halogen     -12.53 -20.34928258  -4.710717 0.0001014
D:LED-D:Halogen     -16.45 -24.26928258  -8.630717 0.0000002
B:LED-A:LED           5.50  -2.31928258  13.319283 0.3665620
C:LED-A:LED           8.71   0.89071742  16.529283 0.0185285
D:LED-A:LED           4.79  -3.02928258  12.609283 0.5471915
C:LED-B:LED           3.21  -4.60928258  11.029283 0.9027236
D:LED-B:LED          -0.71  -8.52928258   7.109283 0.9999920
D:LED-C:LED          -3.92 -11.73928258   3.899283 0.7690762
```

One conclusion from this is that the combination of D and Halogen is significantly stronger than each of the other seven combinations.

3.7 The `bonding` model without Interaction

It seems incorrect in this situation to fit a model without the interaction term, but we'll do so just so you can see what's involved.

```
c3_m2 <- lm(strength ~ resin + light, data = bonding)

summary(c3_m2)
```

```

Call:
lm(formula = strength ~ resin + light, data = bonding)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.1163  -4.9531   0.1187   4.4613  14.4663 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   19.074      1.787  10.676  < 2e-16 ***
resinB         3.815      2.260   1.688  0.09555 .  
resinC         6.740      2.260   2.982  0.00386 ** 
resinD        13.660      2.260   6.044 5.39e-08 ***
lightLED      -1.317      1.598  -0.824  0.41229    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.147 on 75 degrees of freedom
Multiple R-squared:  0.3469,	Adjusted R-squared:  0.312 
F-statistic: 9.958 on 4 and 75 DF,  p-value: 1.616e-06
```

In the no-interaction model, if `light` = Halogen, our equation is:

$$strength = 19.07 + 3.82 resinB + 6.74 resinC + 13.66 resinD$$

And if `light` = LED, our equation is:

$$strength = 17.75 + 3.82 resinB + 6.74 resinC + 13.66 resinD$$

So, in the no-interaction model, only the intercept changes.

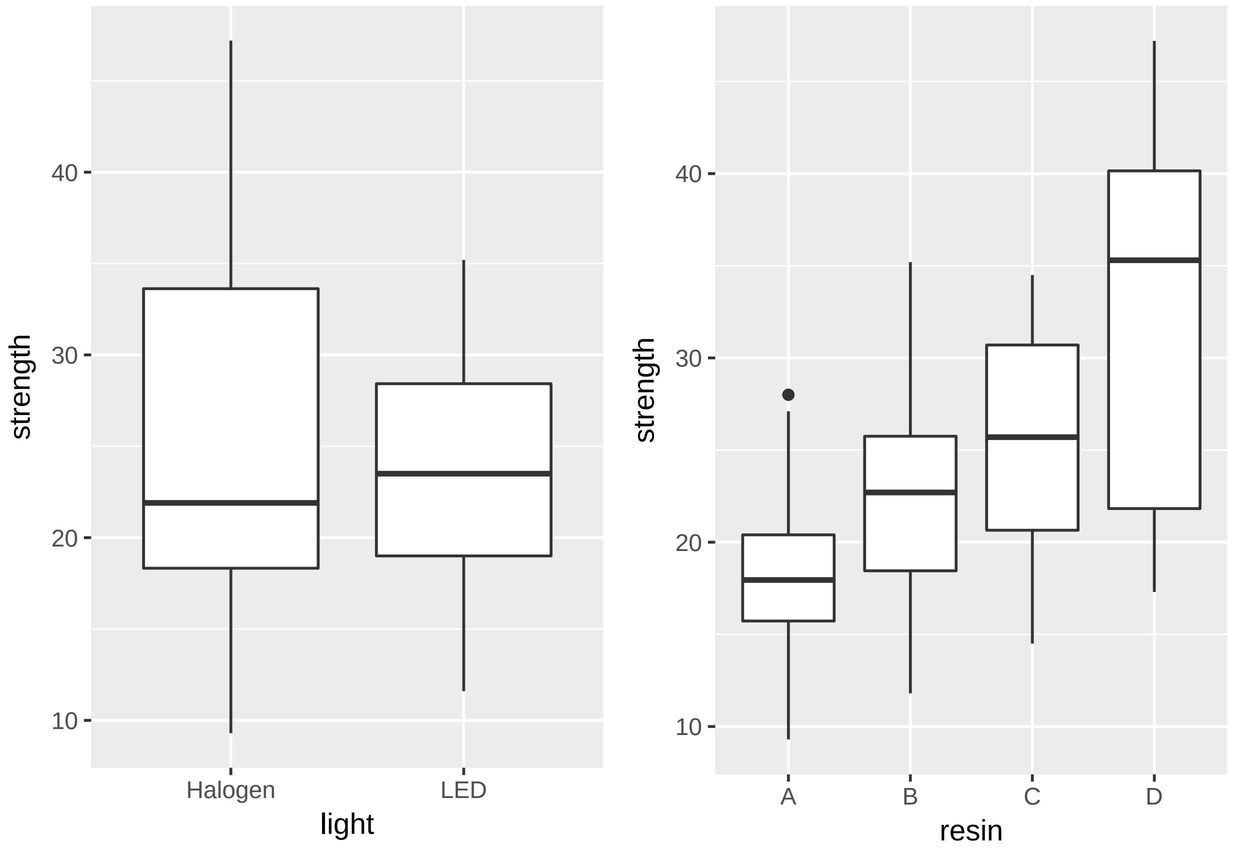
```
anova(c3_m2)
```

```
Analysis of Variance Table

Response: strength
          Df Sum Sq Mean Sq F value    Pr(>F)    
resin      3 1999.7  666.57 13.0514 6.036e-07 ***
light      1   34.7   34.72  0.6797    0.4123    
Residuals 75 3830.5   51.07                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And, it appears, if we ignore the interaction, then `resin` type has a significant impact on `strength` but `light` source doesn't. This is a bit clearer, when we look at boxplots of the separated `light` and `resin` groups.

```
p1 <- ggplot(bonding, aes(x = light, y = strength)) +
    geom_boxplot()
p2 <- ggplot(bonding, aes(x = resin, y = strength)) +
    geom_boxplot()
```

```
gridExtra::grid.arrange(p1, p2, nrow = 1)
```



3.8 cortisol: A Hypothetical Clinical Trial

156 adults who complained of problems with a high-stress lifestyle were enrolled in a hypothetical clinical trial of the effectiveness of a behavioral intervention designed to help reduce stress levels, as measured by salivary cortisol.

The subjects were randomly assigned to one of three intervention groups (usual care, low dose, and high dose.) The "low dose" subjects received a one-week intervention with a follow-up at week 5. The "high dose" subjects received a more intensive three-week intervention, with follow up at week 5.

Since cortisol levels rise and fall with circadian rhythms, the cortisol measurements were taken just after rising for all subjects. These measurements were taken at baseline, and again at five weeks. The difference (baseline - week 5) in cortisol level (in micrograms / l) serves as the primary outcome.

3.8.1 Codebook and Raw Data for cortisol

The data are gathered in the `cortisol` data set. Included are:

Variable	Description
`subject`	subject identification code
`interv`	intervention group (UC = usual care, Low, High)
`waist`	waist circumference at baseline (in inches)

Variable	Description
sex	male or female
cort.1	salivary cortisol level (microg/l) week 1
cort.5	salivary cortisol level (microg/l) week 5

```
cortisol
```

```
# A tibble: 156 x 6
   subject interv waist sex   cort.1 cort.5
     <int> <fct>  <dbl> <fct>  <dbl>  <dbl>
 1    1001 UC      48.3 M      13.4   13.3 
 2    1002 Low     58.3 M      17.8   16.6 
 3    1003 High    43.0 M      14.4   12.7 
 4    1004 Low     44.9 M       9.00   9.80
 5    1005 High    46.1 M      14.2   14.2 
 6    1006 UC      41.3 M      14.8   15.1 
 7    1007 Low     51.0 F      13.7   16.0 
 8    1008 UC      42.0 F      17.3   18.7 
 9    1009 Low     24.7 F      15.3   15.8 
10    1010 Low     59.4 M      12.4   11.7 
# ... with 146 more rows
```

3.9 Creating a factor combining sex and waist

Next, we'll put the **waist** and **sex** data in the **cortisol** example together. We want to build a second categorical variable (called **fat_est**) combining this information, to indicate "healthy" vs. "unhealthy" levels of fat around the waist.

- Male subjects whose waist circumference is 40 inches or more, and
- Female subjects whose waist circumference is 35 inches or more, will fall in the "unhealthy" group.

```
cortisol <- cortisol %>%
    mutate(
        fat_est = factor(case_when(
            sex == "M" & waist >= 40 ~ "unhealthy",
            sex == "F" & waist >= 35 ~ "unhealthy",
            TRUE                     ~ "healthy")),
        cort_diff = cort.1 - cort.5)

summary(cortisol)
```

```
    subject      interv       waist        sex        cort.1      
 Min.   :1001   High:53   Min.   :20.80   F:83   Min.   : 6.000  
 1st Qu.:1040   Low :52   1st Qu.:33.27   M:73   1st Qu.: 9.675  
 Median :1078   UC  :51   Median :40.35          Median :12.400  
 Mean   :1078             Mean   :40.42          Mean   :12.686  
 3rd Qu.:1117             3rd Qu.:47.77          3rd Qu.:16.025  
 Max.   :1156             Max.   :59.90          Max.   :19.000  
     cort.5          fat_est      cort_diff      
 Min.   : 4.2   healthy  : 56   Min.   :-2.3000  
 1st Qu.: 9.6   unhealthy:100   1st Qu.:-0.5000  
 Median :12.6                   Median : 0.2000  
 Mean   :12.4                   Mean   : 0.2821  
 3rd Qu.:15.7                  3rd Qu.: 1.2000
 Max.   :19.7                  Max.   : 2.0000
```

3.10 A Means Plot for the cortisol trial (with standard errors)

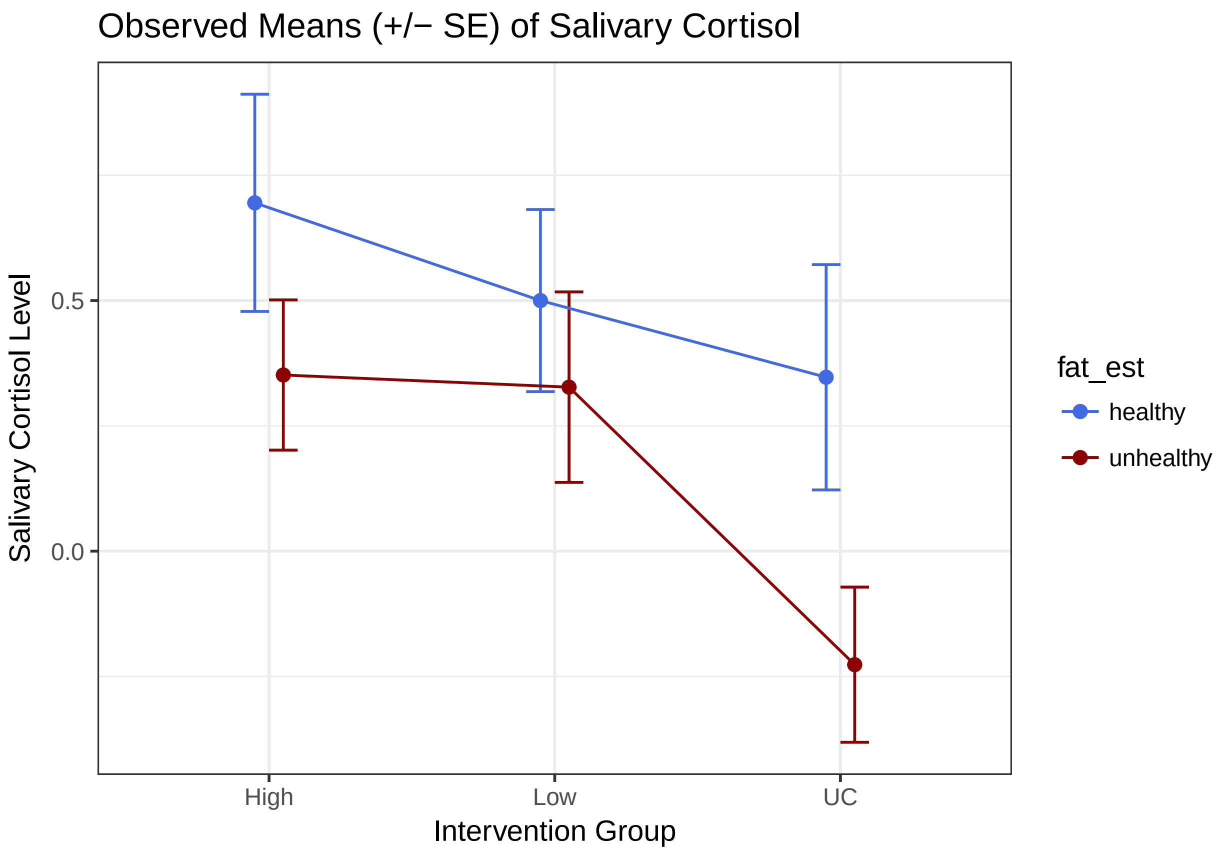
Again, we'll start by building up a data set with the summaries we want to plot.

```
cort.sum <- cortisol %>%
    group_by(interv, fat_est) %>%
    summarize(mean.cort = mean(cort_diff),
              se.cort = sd(cort_diff)/sqrt(n()))

cort.sum
```

```
# A tibble: 6 x 4
# Groups: interv [?]
  interv fat_est   mean.cort se.cort
  <fct>  <fct>         <dbl>   <dbl>
1 High   healthy       0.695   0.217
2 High   unhealthy     0.352   0.150
3 Low    healthy       0.500   0.182
4 Low    unhealthy     0.327   0.190
5 UC     healthy       0.347   0.225
6 UC     unhealthy    -0.226   0.155
```

Now, we'll use this new data set to plot the means and standard errors.

```
## The error bars will overlap unless we adjust the position.
pd <- position_dodge(0.2) # move them .1 to the left and right

ggplot(cort.sum, aes(x = interv, y = mean.cort, col = fat_est)) +
    geom_errorbar(aes(ymin = mean.cort - se.cort,
                      ymax = mean.cort + se.cort),
                  width = 0.2, position = pd) +
    geom_point(size = 2, position = pd) +
    geom_line(aes(group = fat_est), position = pd) +
    scale_color_manual(values = c("royalblue", "darkred")) +
    theme_bw() +
    labs(y = "Salivary Cortisol Level", x = "Intervention Group",
         title = "Observed Means (+/- SE) of Salivary Cortisol")
```



3.11 A Two-Way ANOVA model for cortisol with Interaction

```
c3_m3 <- lm(cort_diff ~ interv * fat_est, data = cortisol)

anova(c3_m3)
```

```
Analysis of Variance Table

Response: cort_diff
                Df  Sum Sq Mean Sq F value  Pr(>F)
interv           2   7.847  3.9235  4.4698 0.01301 *
fat_est          1   4.614  4.6139  5.2564 0.02326 *
interv:fat_est   2   0.943  0.4715  0.5371 0.58554
Residuals      150 131.666  0.8778
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Does it seem like we need the interaction term in this case?

```
summary(c3_m3)
```

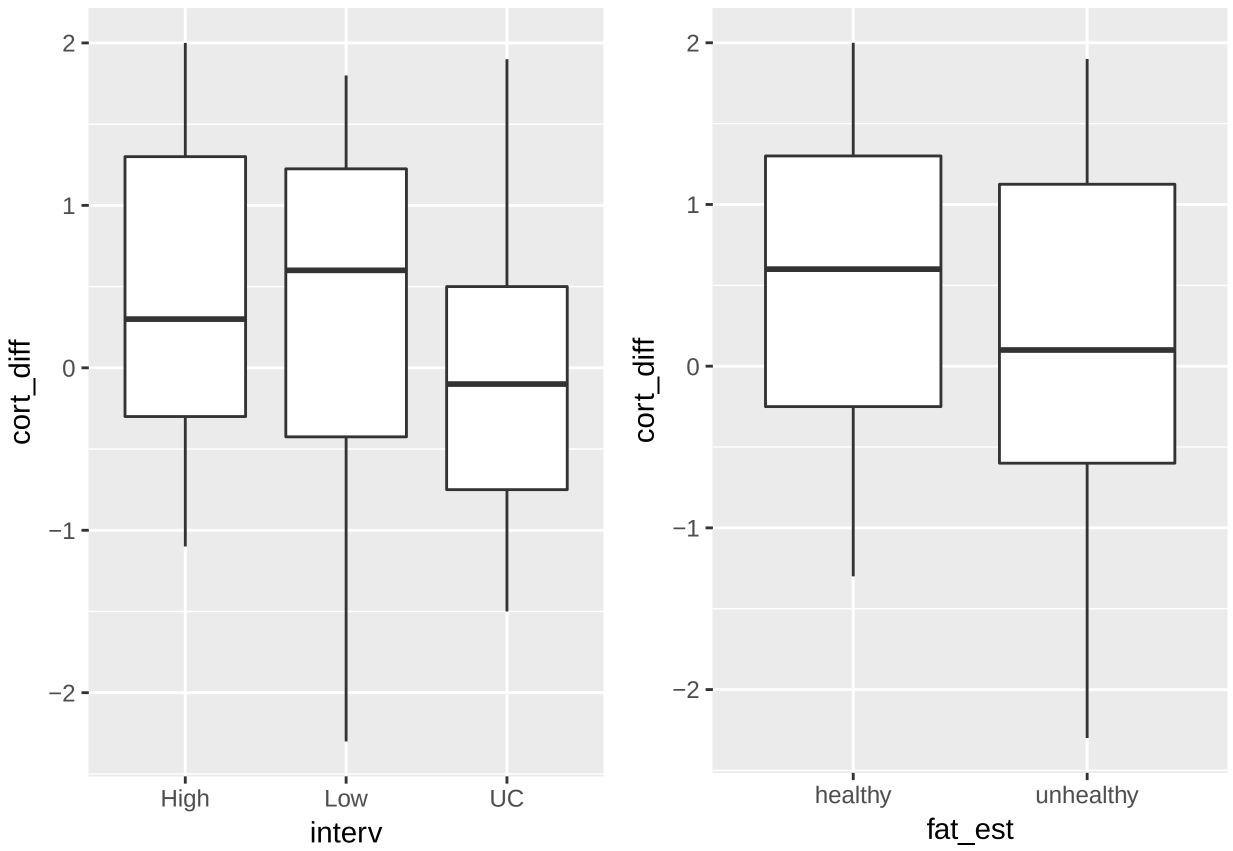
```

Call:
lm(formula = cort_diff ~ interv * fat_est, data = cortisol)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.62727 -0.75702  0.08636  0.84848  2.12647 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)   
(Intercept)                  0.6950     0.2095   3.317  0.00114 **
intervLow                   -0.1950     0.3001  -0.650  0.51689   
intervUC                    -0.3479     0.3091  -1.126  0.26206   
fat_estunhealthy            -0.3435     0.2655  -1.294  0.19774   
intervLow:fat_estunhealthy   0.1708     0.3785   0.451  0.65256   
intervUC:fat_estunhealthy   -0.2300     0.3846  -0.598  0.55068   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9369 on 150 degrees of freedom
Multiple R-squared:  0.0924,    Adjusted R-squared:  0.06214 
F-statistic: 3.054 on 5 and 150 DF,  p-value: 0.01179
```

How do you reconcile the apparent difference in significance levels between this regression summary and the ANOVA table above?

3.12 A Two-Way ANOVA model for cortisol without Interaction

3.12.1 The Graph

```r
p1 <- ggplot(cortisol, aes(x = interv, y = cort_diff)) +
    geom_boxplot()
p2 <- ggplot(cortisol, aes(x = fat_est, y = cort_diff)) +
    geom_boxplot()

gridExtra::grid.arrange(p1, p2, nrow = 1)
```



3.12.2 The ANOVA Model

```
c3_m4 <- lm(cort_diff ~ interv + fat_est, data = cortisol)

anova(c3_m4)
```

```
Analysis of Variance Table

Response: cort_diff
           Df  Sum Sq Mean Sq F value  Pr(>F)
interv      2   7.847  3.9235  4.4972 0.01266 *
fat_est     1   4.614  4.6139  5.2886 0.02283 *
Residuals 152 132.609  0.8724
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How do these results compare to those we saw in the model with interaction?

3.12.3 The Regression Summary

```
summary(c3_m4)
```

```

Call:
lm(formula = cort_diff ~ interv + fat_est, data = cortisol)
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-2.55929 -0.74527  0.05457  0.86456  2.05489 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       0.70452    0.16093   4.378 2.22e-05 ***
intervLow        -0.08645    0.18232  -0.474  0.63606    
intervUC         -0.50063    0.18334  -2.731  0.00707 ** 
fat_estunhealthy -0.35878    0.15601  -2.300  0.02283 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.934 on 152 degrees of freedom
Multiple R-squared:  0.0859,	Adjusted R-squared:  0.06785 
F-statistic: 4.761 on 3 and 152 DF,  p-value: 0.00335
```

3.12.4 Tukey HSD Comparisons

Without the interaction term, we can make direct comparisons between levels of the intervention, and between levels of the fat_est variable. This is probably best done here in a Tukey HSD comparison.

```
TukeyHSD(aov(cort_diff ~ interv + fat_est, data = cortisol))
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = cort_diff ~ interv + fat_est, data = cortisol)

$interv
                diff        lwr         upr     p adj
Low-High -0.09074746 -0.5222655  0.34077063 0.8724916
UC-High  -0.51642619 -0.9500745 -0.08277793 0.0150150
UC-Low   -0.42567873 -0.8613670  0.01000948 0.0570728

$fat_est
                        diff        lwr         upr     p adj
unhealthy-healthy -0.3582443 -0.6662455 -0.05024305 0.0229266
```

What conclusions can we draw, at a 5% significance level?

3.13 An Emphysema Study: Analysis of Covariance

My source for this example is Riffenburgh (2006), section 18.4. Serum theophylline levels (in mg/dl) were measured in 16 patients with emphysema at baseline, then 5 days later (at the end of a course of antibiotics) and then at 10 days after baseline. Clinicians anticipate that the antibiotic will increase the theophylline level. The data are stored in the emphysema.csv data file, and note that the age for patient 5 is not available.

3.13.1 Codebook

Variable	Description
`patient`	ID code
`age`	patient's age in years
`sex`	patient's sex (F or M)
`st_base`	patient's serum theophylline at baseline (mg/dl)
`st_day5`	patient's serum theophylline at day 5 (mg/dl)
`st_day10`	patient's serum theophylline at day 10 (mg/dl)

We're going to look at the change from baseline to day 5 as our outcome of interest, since the clinical expectation is that the antibiotic (azithromycin) will increase theophylline levels.

```
emphysema <- emphysema %>%
    mutate(st_delta = st_day5 - st_base)

emphysema
```

```
# A tibble: 16 x 7
   patient   age sex   st_base st_day5 st_day10 st_delta
     <int> <int> <fct>   <dbl>   <dbl>    <dbl>    <dbl>
 1       1    61 F       14.1     2.30    10.3   -11.8
 2       2    70 F        7.20    5.40     7.30  - 1.80
 3       3    65 M       14.2    11.9     11.3   - 2.30
 4       4    65 M       10.3    10.7     13.8     0.400
 5       5    NA M        9.90   10.7     11.7     0.800
 6       6    76 M        5.20    6.80     4.20    1.60
 7       7    72 M       10.4    14.6     14.1     4.20
 8       8    69 F       10.5     7.20     5.40  - 3.30
 9       9    66 M        5.00    5.00     5.10    0
10      10    62 M        8.60    8.10     7.40  - 0.500
11      11    65 F       16.6    14.9     13.0   - 1.70
12      12    71 M       16.4    18.6     17.1     2.20
13      13    51 F       12.2    11.0     12.3   - 1.20
14      14    71 M        6.60    3.70     4.50  - 2.90
15      15    64 F       15.4    15.2     13.6   - 0.200
16      16    50 M       10.2    10.8     11.2     0.600
```

3.13.2 Does sex affect the mean change in theophylline?

```
emphysema %>% skim(st_delta)
```

```
Skim summary statistics
 n obs: 16
 n variables: 7

Variable type: numeric
 variable missing complete  n  mean   sd    p0   p25 median  p75 p100
 st_delta       0       16 16 -0.99 3.48 -11.8 -1.92  -0.35 0.65  4.2
```

```
emphysema %>% group_by(sex) %>% skim(st_delta)
```

```
Skim summary statistics
 n obs: 16
 n variables: 7
 group variables: sex

Variable type: numeric
 sex variable missing complete  n  mean   sd    p0   p25 median   p75 p100
   F st_delta       0        6  6 -3.33 4.27 -11.8 -2.92  -1.75 -1.32 -0.2
   M st_delta       0       10 10  0.41 2.07  -2.9 -0.38   0.5   1.4   4.2
```

Overall, the mean change in theophylline during the course of the antibiotic is -0.99, but this is -3.33 for female patients and 0.41 for male patients.

A one-way ANOVA model looks like this:

```
anova(lm(st_delta ~ sex, data = emphysema))
```

```
Analysis of Variance Table

Response: st_delta
          Df  Sum Sq Mean Sq F value  Pr(>F)
sex        1  52.547  52.547  5.6789 0.03189 *
Residuals 14 129.542   9.253
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA F test finds a statistically significant difference between the mean `st_delta` among males and the mean `st_delta` among females. But is there more to the story?

3.13.3 Is there an association between age and sex in this study?

```
emphysema %>% group_by(sex) %>% skim(age)
```

```
Skim summary statistics
 n obs: 16
 n variables: 7
 group variables: sex

Variable type: integer
 sex variable missing complete  n  mean   sd p0   p25 median p75 p100
   F      age       0        6  6 63.33 6.89 51 61.75   64.5  68   70
   M      age       1        9 10 66.44 7.57 50 65      66    71   76
```

But we note that the male patients are also older than the female patients, on average (mean age for males is 66.4, for females 63.3)

- Does the fact that male patients are older affect change in theophylline level?
- And how should we deal with the one missing `age` value (in a male patient)?

3.13.4 Adding a quantitative covariate, age, to the model

We could fit an ANOVA model to predict `st_delta` using `sex` and `age` directly, but only if we categorized `age` into two or more groups. Because `age` is not categorical, we cannot include it in an ANOVA. But if age is an influence, and we don't adjust for it, it may well bias the outcome of our initial ANOVA. With a quantitative variable like `age`, we will need a method called ANCOVA, for **analysis of covariance**.

3.13.4.1 The ANCOVA model

ANCOVA in this case is just an ANOVA model with our outcome (`st_delta`) adjusted for a continuous covariate, called `age`. For the moment, we'll ignore the one subject with missing `age` and simply fit the regression model with `sex` and `age`.

```
summary(lm(st_delta ~ sex + age, data = emphysema))
```

```

Call:
lm(formula = st_delta ~ sex + age, data = emphysema)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.3352 -0.4789  0.6948  1.5580  3.5202 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -6.90266    7.92948  -0.871   0.4011  
sexM         3.52466    1.75815   2.005   0.0681 .
age          0.05636    0.12343   0.457   0.6561  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.255 on 12 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.2882,    Adjusted R-squared:  0.1696 
F-statistic:  2.43 on 2 and 12 DF,  p-value: 0.13
```

This model assumes that the slope of the regression line between `st_delta` and `age` is the same for both sexes.

Note that the model yields `st_delta` = -6.9 + 3.52 (`sex` = male) + 0.056 `age`, or

- `st_delta` = -3.38 + 0.056 `age` for female patients, and
- `st_delta` = -6.9 + 0.056 `age` for male patients.

Note that we can test this assumption of equal slopes by fitting an alternative model (with a product term between `sex` and `age`) that doesn't require the assumption, and we'll do that later.

3.13.4.2 The ANCOVA Table

First, though, we'll look at the ANCOVA table.

```
anova(lm(st_delta ~ sex + age, data = emphysema))
```

```
Analysis of Variance Table

Response: st_delta
          Df  Sum Sq Mean Sq F value  Pr(>F)  
sex        1  49.284  49.284  4.6507 0.05203 .
age        1   2.209   2.209  0.2085 0.65612  
Residuals 12 127.164  10.597                  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When we tested `sex` without accounting for `age`, we found a p value of 0.032, which is less than our usual cutpoint of 0.05. But when we adjusted for `age`, we find that `sex` loses significance, even though `age` is not a significant influence on `st_delta` by itself, according to the ANCOVA table.

3.13.5 Rerunning the ANCOVA model after simple imputation

We could have *imputed* the missing `age` value for patient 5, rather than just deleting that patient. Suppose we do the simplest potentially reasonable thing to do: insert the mean `age` in where the NA value currently exists.

```
emph_imp <- replace_na(emphysema, list(age = mean(emphysema$age, na.rm = TRUE)))

emph_imp
```

```
# A tibble: 16 x 7
   patient   age sex   st_base st_day5 st_day10 st_delta
     <int> <dbl> <fct>   <dbl>   <dbl>    <dbl>    <dbl>
 1       1  61.0 F       14.1     2.30    10.3   -11.8
 2       2  70.0 F        7.20    5.40     7.30  - 1.80
 3       3  65.0 M       14.2    11.9     11.3   - 2.30
 4       4  65.0 M       10.3    10.7     13.8     0.400
 5       5  65.2 M        9.90   10.7     11.7     0.800
 6       6  76.0 M        5.20    6.80     4.20    1.60
 7       7  72.0 M       10.4    14.6     14.1     4.20
 8       8  69.0 F       10.5     7.20     5.40  - 3.30
 9       9  66.0 M        5.00    5.00     5.10    0
10      10  62.0 M        8.60    8.10     7.40  - 0.500
11      11  65.0 F       16.6    14.9     13.0   - 1.70
12      12  71.0 M       16.4    18.6     17.1     2.20
13      13  51.0 F       12.2    11.0     12.3   - 1.20
14      14  71.0 M        6.60    3.70     4.50  - 2.90
15      15  64.0 F       15.4    15.2     13.6   - 0.200
16      16  50.0 M       10.2    10.8     11.2     0.600
```

More on simple imputation and missing data is coming soon.

For now, we can rerun the ANCOVA model on this new data set, after imputation…

```
anova(lm(st_delta ~ sex + age, data = emph_imp))
```

```
Analysis of Variance Table

Response: st_delta
          Df  Sum Sq Mean Sq F value  Pr(>F)
sex        1  52.547  52.547  5.3623 0.03755 *
age        1   2.151   2.151  0.2195 0.64721
Residuals 13 127.392   9.799
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When we do this, we see that now the `sex` variable returns to a p value below 0.05. Our complete case analysis (which omitted patient 5) gives us a different result than the ANCOVA based on the data after mean imputation.

3.13.6 Looking at a factor-covariate interaction

Let's run a model including the interaction (product) term between `age` and `sex`, which implies that the slope of `age` on our outcome (`st_delta`) depends on the patient's sex. We'll use the imputed data again.

Here is the new ANCOVA table, which suggests that the interaction of `age` and `sex` is small (because it accounts for only a small amount of the total Sum of Squares) and not significant (p = 0.91).

```
anova(lm(st_delta ~ sex * age, data = emph_imp))
```

```
Analysis of Variance Table

Response: st_delta
          Df  Sum Sq Mean Sq F value  Pr(>F)  
sex        1  52.547  52.547  4.9549 0.04594 *
age        1   2.151   2.151  0.2028 0.66051  
sex:age    1   0.130   0.130  0.0123 0.91355  
Residuals 12 127.261  10.605                  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the interaction term is neither substantial nor significant, we probably don't need it here. But let's look at its interpretation anyway, just to fix ideas. To do that, we'll need the coefficients from the underlying regression model.

```
tidy(lm(st_delta ~ sex * age, data = emph_imp))
```

```
         term    estimate  std.error  statistic   p.value
1 (Intercept) -5.64606742 13.4536974 -0.4196666 0.6821446
2        sexM  1.72031026 16.8389209  0.1021627 0.9203148
3         age  0.03651685  0.2113871  0.1727488 0.8657284
4    sexM:age  0.02885946  0.2603044  0.1108681 0.9135536
```

Our ANCOVA model for `st_delta` incorporating the `age` x `sex` product term is -5.65 + 1.72 (sex = M) + 0.037 age + 0.029 (sex = M)(age). So that means:

- our model for females is `st_delta` = -5.65 + 0.037 `age`
- our model for males is `st_delta` = (-5.65 + 1.72) + (0.037 + 0.029) `age`, or -3.93 + 0.066 `age`

but, again, our conclusion from the ANCOVA table is that this increase in complexity (letting both the slope and intercept vary by `sex`) doesn't add much in the way of predictive value for our `st_delta` outcome.

Chapter 4

Missing Data Mechanisms and Single Imputation

Almost all serious statistical analyses have to deal with missing data. Data values that are missing are indicated in R, and to R, by the symbol `NA`.

4.1 A Toy Example

In the following tiny data set called `sbp_example`, we have four variables for a set of 15 subjects. In addition to a subject id, we have:

- the treatment this subject received (A, B or C are the treatments),
- an indicator (1 = yes, 0 = no) of whether the subject has diabetes,
- the subject's systolic blood pressure at baseline
- the subject's systolic blood pressure after the application of the treatment

```
# create some temporary variables

subject <- 101:115
x1 <- c("A", "B", "C", "A", "C", "A", "A", NA, "B", "C", "A", "B", "C", "A", "B")
x2 <- c(1, 0, 0, 1, NA, 1, 0, 1, NA, 1, 0, 0, 1, 1, NA)
x3 <- c(120, 145, 150, NA, 155, NA, 135, NA, 115, 170, 150, 145, 140, 160, 135)
x4 <- c(105, 135, 150, 120, 135, 115, 160, 150, 130, 155, 140, 140, 150, 135, 120)

sbp_example <-
  data.frame(subject, treat = x1, diabetes = x2,
             sbp.before = x3, sbp.after = x4) %>%
  tbl_df

rm(subject, x1, x2, x3, x4) # just cleaning up

sbp_example
```

```
# A tibble: 15 x 5
   subject treat diabetes sbp.before sbp.after
     <int> <fct>    <dbl>      <dbl>     <dbl>
 1     101 A         1.00        120       105
 2     102 B         0           145       135
 3     103 C         0           150       150
 4     104 A         1.00         NA       120
 5     105 C        NA           155       135
 6     106 A         1.00         NA       115
 7     107 A         0           135       160
 8     108 <NA>      1.00         NA       150
 9     109 B        NA           115       130
10     110 C         1.00        170       155
11     111 A         0           150       140
12     112 B         0           145       140
13     113 C         1.00        140       150
14     114 A         1.00        160       135
15     115 B        NA           135       120
```

4.1.1 How many missing values do we have in each column?

```
colSums(is.na(sbp_example))
```

```
   subject      treat   diabetes sbp.before  sbp.after 
         0          1          3          3          0 
```

We are missing one `treat`, 3 `diabetes` and 3 `sbp.before` values.

4.1.2 What is the pattern of missing data?

```
mice::md.pattern(sbp_example)
```

```
  subject sbp.after treat diabetes sbp.before  
9       1         1     1        1          1 0
3       1         1     1        0          1 1
2       1         1     1        1          0 1
1       1         1     0        1          0 2
        0         0     1        3          3 7
```

We have nine subjects with complete data, three subjects with missing `diabetes` (only), two subjects with missing `sbp.before` (only), and 1 subject with missing `treat` and `sbp.before`.

4.1.3 How can we identify the subjects with missing data?

```
sbp_example %>% filter(!complete.cases(.))
```

```
# A tibble: 6 x 5
  subject treat diabetes sbp.before sbp.after
    <int> <fct>    <dbl>      <dbl>     <dbl>
1     104 A         1.00         NA       120
2     105 C        NA           155       135
3     106 A         1.00         NA       115
4     108 <NA>      1.00         NA       150
5     109 B        NA           115       130
6     115 B        NA           135       120
```

4.2 Missing-data mechanisms

My source for this description of mechanisms is Chapter 25 of Gelman and Hill (2007), and that chapter is available at this link.

1. **MCAR = Missingness completely at random**. A variable is missing completely at random if the probability of missingness is the same for all units, for example, if for each subject, we decide whether to collect the `diabetes` status by rolling a die and refusing to answer if a "6" shows up. If data are missing completely at random, then throwing out cases with missing data does not bias your inferences.
2. **Missingness that depends only on observed predictors**. A more general assumption, called **missing at random** or **MAR**, is that the probability a variable is missing depends only on available information. Here, we would have to be willing to assume that the probability of nonresponse to `diabetes` depends only on the other, fully recorded variables in the data. It is often reasonable to model this process as a logistic regression, where the outcome variable equals 1 for observed cases and 0 for missing. When an outcome variable is missing at random, it is acceptable to exclude the missing cases (that is, to treat them as NA), as long as the regression controls for all the variables that affect the probability of missingness.
3. **Missingness that depends on unobserved predictors**. Missingness is no longer "at random" if it depends on information that has not been recorded and this information also predicts the missing values. If a particular treatment causes discomfort, a patient is more likely to drop out of the study. This missingness is not at random (unless "discomfort" is measured and observed for all patients). If missingness is not at random, it must be explicitly modeled, or else you must accept some bias in your inferences.
4. **Missingness that depends on the missing value itself.** Finally, a particularly difficult situation arises when the probability of missingness depends on the (potentially missing) variable itself. For example, suppose that people with higher earnings are less likely to reveal them.

Essentially, situations 3 and 4 are referred to collectively as **non-random missingness**, and cause more trouble for us than 1 and 2.

4.3 Options for Dealing with Missingness

There are several available methods for dealing with missing data that are MCAR or MAR, but they basically boil down to:

- Complete Case (or Available Case) analyses
- Single Imputation
- Multiple Imputation

4.4 Complete Case (and Available Case) analyses

In **Complete Case** analyses, rows containing NA values are omitted from the data before analyses commence. This is the default approach for many statistical software packages, and may introduce unpredictable bias and fail to include some useful, often hard-won information.

- A complete case analysis can be appropriate when the number of missing observations is not large, and the missing pattern is either MCAR (missing completely at random) or MAR (missing at random.)
- Two problems arise with complete-case analysis:
    1. If the units with missing values differ systematically from the completely observed cases, this could bias the complete-case analysis.
    2. If many variables are included in a model, there may be very few complete cases, so that most of the data would be discarded for the sake of a straightforward analysis.

- A related approach is *available-case* analysis where different aspects of a problem are studied with different subsets of the data, perhaps identified on the basis of what is missing in them.

4.5 Single Imputation

In **single imputation** analyses, NA values are estimated/replaced *one time* with *one particular data value* for the purpose of obtaining more complete samples, at the expense of creating some potential bias in the eventual conclusions or obtaining slightly *less* accurate estimates than would be available if there were no missing values in the data.

- A single imputation can be just a replacement with the mean or median (for a quantity) or the mode (for a categorical variable.) However, such an approach, though easy to understand, underestimates variance and ignores the relationship of missing values to other variables.
- Single imputation can also be done using a variety of models to try to capture information about the NA values that are available in other variables within the data set.
- The `simputation` package can help us execute single imputations using a wide variety of techniques, within the pipe approach used by the `tidyverse`. Another approach I have used in the past is the `mice` package, which can also perform single imputations.

4.6 Multiple Imputation

Multiple imputation, where NA values are repeatedly estimated/replaced with multiple data values, for the purpose of obtaining mode complete samples *and* capturing details of the variation inherent in the fact that the data have missingness, so as to obtain *more* accurate estimates than are possible with single imputation.

- We'll postpone the discussion of multiple imputation for a while.

4.7 Building a Complete Case Analysis

We can drop all of the missing values from a data set with `drop_na` or with `na.omit` or by filtering for `complete.cases`. Any of these approaches produces the same result - a new data set with 9 rows (after dropping the six subjects with any NA values) and 5 columns.

```
cc.1 <- na.omit(sbp_example)
cc.2 <- sbp_example %>% drop_na
cc.3 <- sbp_example %>% filter(complete.cases(.))
```

4.8 Single Imputation with the Mean or Mode

The most straightforward approach to single imputation is to impute a single summary of the variable, such as the mean, median or mode.

```
skim(sbp_example)
```

```
Skim summary statistics
 n obs: 15
 n variables: 5

Variable type: factor
 variable missing complete  n n_unique              top_counts ordered
```

```
    treat       1       14 15        3 A: 6, B: 4, C: 4, NA: 1   FALSE

Variable type: integer
 variable missing complete  n mean   sd  p0   p25 median   p75 p100
  subject       0       15 15  108 4.47 101 104.5    108 111.5  115

Variable type: numeric
   variable missing complete  n   mean    sd  p0 p25 median    p75 p100
   diabetes       3       12 15   0.58  0.51   0   0      1   1       1
  sbp.after       0       15 15 136     15.83 105 125    135 150     160
 sbp.before       3       12 15 143.33 15.72 115 135    145 151.25   170
```

Here, suppose we decide to impute

- `sbp.before` with the mean (143.33) among non-missing values,
- `diabetes` with its median (1) among non-missing values, and
- `treat` with its most common value, or mode (A)

```
si.1 <- sbp_example %>%
    replace_na(list(sbp.before = 143.33,
                    diabetes = 1,
                    treat = "A"))
si.1
```

```
# A tibble: 15 x 5
   subject treat diabetes sbp.before sbp.after
     <int> <fct>    <dbl>      <dbl>     <dbl>
 1     101 A         1.00        120       105
 2     102 B         0           145       135
 3     103 C         0           150       150
 4     104 A         1.00        143       120
 5     105 C         1.00        155       135
 6     106 A         1.00        143       115
 7     107 A         0           135       160
 8     108 A         1.00        143       150
 9     109 B         1.00        115       130
10     110 C         1.00        170       155
11     111 A         0           150       140
12     112 B         0           145       140
13     113 C         1.00        140       150
14     114 A         1.00        160       135
15     115 B         1.00        135       120
```

We could accomplish the same thing with, for example:

```
si.2 <- sbp_example %>%
    replace_na(list(sbp.before = mean(sbp_example$sbp.before, na.rm = TRUE),
                    diabetes = median(sbp_example$diabetes, na.rm = TRUE),
                    treat = "A"))
```

4.9 Doing Single Imputation with `simputation`

Single imputation is a potentially appropriate method when missingness can be assumed to be either completely at random (MCAR) or dependent only on observed predictors (MAR). We'll use the `simputation` package to accomplish it.

- The `simputation` vignette is available at https://cran.r-project.org/web/packages/simputation/vignettes/intro.html
- The `simputation` reference manual is available at https://cran.r-project.org/web/packages/simputation/simputation.pdf

4.9.1 Mirroring Our Prior Approach (imputing means/medians/modes)

Suppose we want to mirror what we did above, simply impute the mean for `sbp.before` and the median for `diabetes` again.

```
si.3 <- sbp_example %>%
    impute_lm(sbp.before ~ 1) %>%
    impute_median(diabetes ~ 1) %>%
    replace_na(list(treat = "A"))

si.3
```

```
# A tibble: 15 x 5
   subject treat diabetes sbp.before sbp.after
 *   <int> <fct>    <dbl>      <dbl>     <dbl>
 1     101 A         1.00        120       105
 2     102 B         0           145       135
 3     103 C         0           150       150
 4     104 A         1.00        143       120
 5     105 C         1.00        155       135
 6     106 A         1.00        143       115
 7     107 A         0           135       160
 8     108 A         1.00        143       150
 9     109 B         1.00        115       130
10     110 C         1.00        170       155
11     111 A         0           150       140
12     112 B         0           145       140
13     113 C         1.00        140       150
14     114 A         1.00        160       135
15     115 B         1.00        135       120
```

4.9.2 Using a model to impute `sbp.before` and `diabetes`

Suppose we wanted to use:

- a robust linear model to predict `sbp.before` missing values, on the basis of `sbp.after` and `diabetes` status, and
- a predictive mean matching approach to predict `diabetes` status, on the basis of `sbp.after`, and
- a decision tree approach to predict `treat` status, using all other variables in the data

```
set.seed(50001)

imp.4 <- sbp_example %>%
    impute_rlm(sbp.before ~ sbp.after + diabetes) %>%
    impute_pmm(diabetes ~ sbp.after) %>%
    impute_cart(treat ~ .)

imp.4
```

```
# A tibble: 15 x 5
```

```
   subject treat diabetes sbp.before sbp.after
 *   <int> <fct>    <dbl>      <dbl>     <dbl>
 1     101 A         1.00        120       105
 2     102 B         0           145       135
 3     103 C         0           150       150
 4     104 A         1.00        139       120
 5     105 C         1.00        155       135
 6     106 A         1.00        136       115
 7     107 A         0           135       160
 8     108 A         1.00        155       150
 9     109 B         1.00        115       130
10     110 C         1.00        170       155
11     111 A         0           150       140
12     112 B         0           145       140
13     113 C         1.00        140       150
14     114 A         1.00        160       135
15     115 B         1.00        135       120
```

Details on the many available methods in `simputation` are provided in its manual. These include:

- `impute_cart` uses a Classification and Regression Tree approach for numerical or categorical data. There is also an `impute_rf` command which uses Random Forests for imputation.
- `impute_pmm` is one of several "hot deck" options for imputation, this one is predictive mean matching, which can be used with numeric data (only). Missing values are first imputed using a predictive model. Next, these predictions are replaced with the observed values which are nearest to the prediction. Other imputation options in this group include random hot deck, sequential hot deck and k-nearest neighbor imputation.
- `impute_rlm` is one of several regression imputation methods, including linear models, robust linear models (which use what is called M-estimation to impute numerical variables) and lasso/elastic net/ridge regression models.

`simputation` can also do EM-based multivariate imputation, and multivariate random forest imputation, as well as many other sorts of approaches.

4.10 (DRAFT material) How might we validate this model?

Here's some early code for that issue, which is built on some material by David Robinson at https://rpubs.com/dgrtwo/cv-modelr

This bit of code performs what is called *10-crossfold separation.* In words, this approach splits the 896 observations in our data into 10 exclusive partitions of about 90% into a training sample, and the remaining 10% in a test sample. The next part of the code maps a modeling step to the training data, and then fits the resulting model on the test data using the `broom` package's `augment` function.

I've selected the variables in this case so that the model we'll fit is the `m2_c7` model we've been looking at, although there are several ways to accomplish this.

```r
set.seed(4320118)

models <- smartcle2 %>%
    select(bmi, female, exerany, sleephrs,
           internet30, alcdays, genhealth) %>%
    crossv_kfold(k = 10) %>%
    mutate(model = map(train, ~ lm(bmi ~ ., data = .)))

predictions <- models %>%
    unnest(map2(model, test, ~ augment(.x, newdata = .y)))

predictions
```

```
# A tibble: 896 x 10
   .id     bmi female exerany sleephrs internet30 alcdays genhealth
   <chr> <dbl>  <int>   <int>    <int>      <int>   <int> <fct>
 1 01     24.1      0       1        7          1       2 1_Excellent
 2 01     36.4      0       1        8          1       0 4_Fair
 3 01     32.1      1       0        4          1       5 2_VeryGood
 4 01     27.3      0       1        8          1       0 1_Excellent
 5 01     28.0      0       1        7          1       4 2_VeryGood
 6 01     22.5      1       1        7          1       3 2_VeryGood
 7 01     26.3      0       1        7          1       1 1_Excellent
 8 01     22.4      0       1        8          1       4 1_Excellent
 9 01     19.3      1       0        6          1       0 3_Good
10 01     24.2      1       0        6          0       0 3_Good
# ... with 886 more rows, and 2 more variables: .fitted <dbl>, .se.fit
#   <dbl>
```

The results are a set of predictions based on the splits into training and test groups (remember there are 10 of them, indexed by `.id`) that describe the complete set of 896 respondents again.

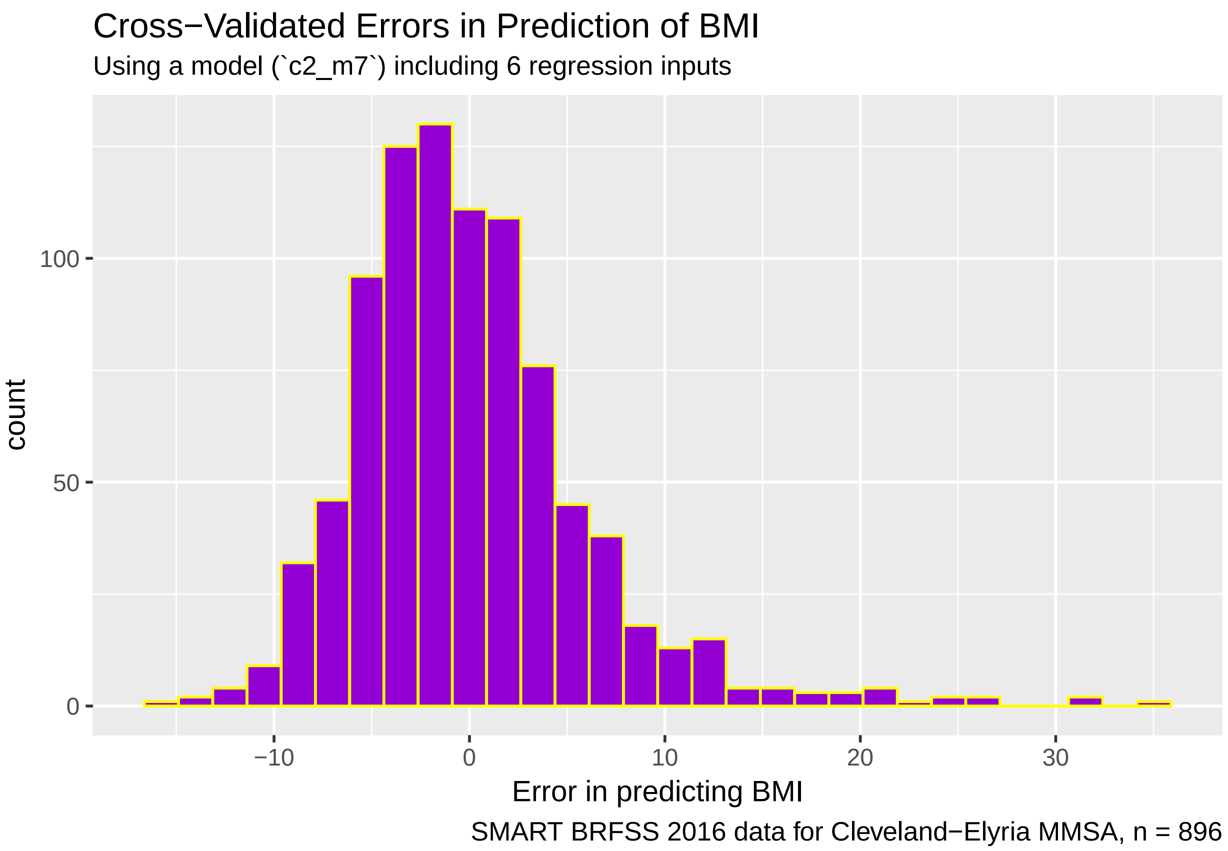
What this lets us now do is calculate the root Mean Squared Prediction Error (RMSE) and Mean Absolute Prediction Error (MAE) for this model (the `c2_m7` model) across these observations, and also to compare that error to a model that simply predicts the mean `bmi` across all patients (the `intercept only` model.) In practice, we could consider two distinct models in doing this work.

```
predictions %>%
    summarize(RMSE_c2_m7 = sqrt(mean((bmi - .fitted) ^2)),
              MAE_c2_m7 = mean(abs(bmi - .fitted)),
              RMSE_interceptonly = sqrt(mean((bmi - mean(bmi))^2)),
              MAE_interceptonly = mean(abs(bmi - mean(bmi))))
```

```
# A tibble: 1 x 4
  RMSE_c2_m7 MAE_c2_m7 RMSE_interceptonly MAE_interceptonly
       <dbl>     <dbl>              <dbl>             <dbl>
1       6.03      4.40               6.32              4.59
```

Another thing we could do with this tibble of predictions we have created is to graph the size of the prediction errors (observed `bmi` minus predicted values in `.fitted`) that our modeling approach makes.

```
predictions %>%
    mutate(errors = bmi - .fitted) %>%
    ggplot(., aes(x = errors)) +
    geom_histogram(bins = 30, fill = "darkviolet", col = "yellow") +
    labs(title = "Cross-Validated Errors in Prediction of BMI",
         subtitle = "Using a model (`c2_m7`) including 6 regression inputs",
         caption = "SMART BRFSS 2016 data for Cleveland-Elyria MMSA, n = 896",
         x = "Error in predicting BMI")
```



4.11 Coming Soon ...

1. Would stepwise regression help us build a better model for `bmi`?
    - Is there a better approach for variable selection? What's this I hear about "best subsets", for example?
2. How should we think about potential transformations of these predictors?
    - What's a Spearman rho-squared plot, and how might it help us decide how to spend degrees of freedom on non-linear terms better?
3. How do we deal with missing data in fitting and evaluating a linear regression model if we don't actually want to drop all of the incomplete cases?
4. How can we use the `ols` tool in the `rms` package to fit regression models?
5. How can we use the tools in the `arm` package to fit and evaluate regression models?

Bibliography

Barnett, P. A., Roman-Golstein, S., Ramsey, F., et al. (1995). Differential permeability and quantitative mr imaging of a human lung carcinoma brain xenograft in the nude rat. *American Journal of Pathology*, 146(2):436–449.

Berkhemer, O. A., Fransen, P. S. S., Buemer, D., et al. (2015). A randomized trial of intraarterial treatment for acute ischemic stroke. *New England Journal of Medicine*, 372:11–20.

Gelman, A. and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge University Press, New York.

Kim, H.-Y. (2014). Statistical notes for clinical researchers: Two-way analysis of variance (anova) - exploring possible interaction between factors. *Restorative Dentistry & Endodontics*, 39(2):143–147.

Ramsey, F. L. and Schafer, D. W. (2002). *The Statistical Sleuth: A Course in Methods of Data Analysis.* Duxbury, Pacific Grove, CA, second edition.

Riffenburgh, R. H. (2006). *Statistics in Medicine.* Elsevier Academic Press, Burlington, MA, second edition.

Roy, D., Talajic, M., Nattel, S., et al. (2008). Rhythm control versus rate control for atrial fibrillation and heart failure. *New England Journal of Medicine*, 358:2667–2677.

Tolaney, S. M., Barry, W. T., Chau, T. D., et al. (2015). Adjuvant paclitaxel and trastuzumab for node-negative, her2-positive breast cancer. *New England Journal of Medicine*, 372:134–141.