432 Class 03 Slides

thomaselove.github.io/432

2022-01-18

Today's Agenda

- Create a data set for week 2 analyses from `smart_ohio`
- Making cleaning / tidying decisions, then saving our work
- Simple imputation
- Splitting the sample with `rsample` tools
- Fitting a model (and then several more models) with `lm`
    - Incorporating an interaction between factors
- Regression Diagnostics via Residual Plots

Creating and Managing the Data for Week 2

Setup

```
knitr::opts_chunk$set(comment = NA)
options(width = 60)

library(here); library(knitr)
library(janitor); library(patchwork)
library(naniar); library(simputation)
library(skimr)          ## for a specific summary
library(equatiomatic)   ## print equations
library(broom)
library(rsample)        ## new today: data splitting
library(yardstick)      ## new today: evaluating fits
library(tidyverse)

theme_set(theme_bw())
options(dplyr.summarise.inform = FALSE)  ## avoid message
```

Similar approach as last time...

```
smart_ohio <- read_csv(here("data/smart_ohio.csv"))

week2 <- smart_ohio %>%
    filter(hx_diabetes == 0,
           mmsa == "Cleveland-Elyria",
           complete.cases(bmi)) %>%
    select(bmi, inc_imp, fruit_day, drinks_wk,
           female, exerany, genhealth, race_eth,
           hx_diabetes, mmsa, SEQNO) %>%
    type.convert(as.is = FALSE) %>%
    mutate(ID = as.character(SEQNO - 2017000000)) %>%
    relocate(ID)
```

```
week2
```

```
# A tibble: 894 x 12
   ID      bmi inc_imp fruit_day drinks_wk female exerany
   <chr> <dbl>   <int>     <dbl>     <dbl>  <int>   <int>
 1 2      23.0   86865      4         0          1       0
 2 3      26.9      NA      3         0          1       1
 3 4      26.5      NA      2         4.67       1       1
 4 5      24.2   58311      0.57      0.93       0       1
 5 7      23.0    2318      2         2          0       1
 6 8      28.4   79667      1         0          0       1
 7 9      30.1   47880      0.23      0          0       1
 8 10     19.8  100136      0.77      0.47       1       1
 9 11     27.2   73145      0.71      0          0       1
10 12     24.6   76917      1.07      0          1       1
# ... with 884 more rows, and 5 more variables:
#   genhealth <fct>, race_eth <fct>, hx_diabetes <int>,
#   mmsa <fct>, SEQNO <int>
```

Codebook for useful `week2` variables

- 894 subjects in Cleveland-Elyria with `bmi` and no history of diabetes

Variable	Description
`bmi`	(outcome) Body-Mass index in kg/m^2.
`inc_imp`	income (imputed from grouped values) in $
`fruit_day`	average fruit servings consumed per day
`drinks_wk`	average alcoholic drinks consumed per week
`female`	sex: 1 = female, 0 = male
`exerany`	any exercise in the past month: 1 = yes, 0 = no
`genhealth`	self-reported overall health (5 levels)
`race_eth`	race and Hispanic/Latinx ethnicity (5 levels)

- plus `ID`, `SEQNO`, `hx_diabetes` (all 0), `MMSA` (all Cleveland-Elyria)
- See Chapter 2 of the Course Notes for details on the variables

Basic Data Summaries

Available approaches include:

- `summary`
- `mosaic` package's `inspect()`
- `skimr` package's `skim_without_charts()`
- `Hmisc` package's `describe`

all of which can work nicely in an HTML presentation, but none of them fit well on one of these slides.

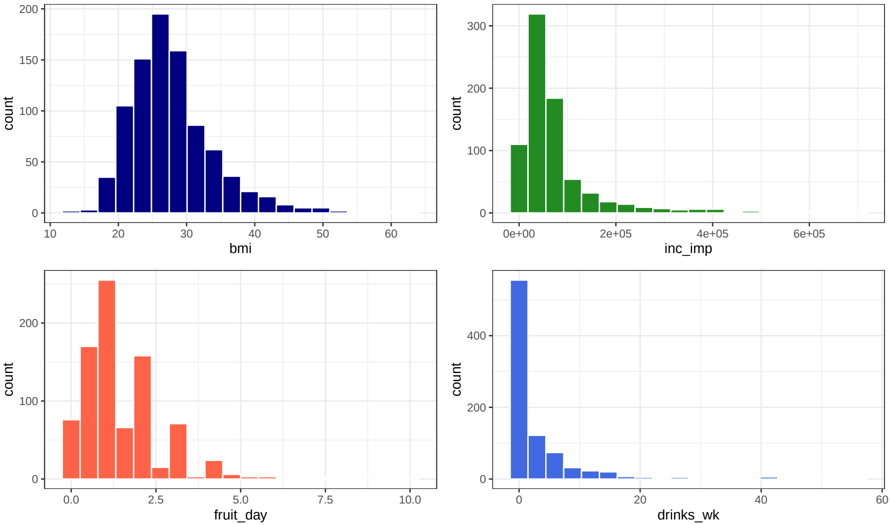
Summarizing the Quantities (Raw `week2`)

```
week2 %>% select(bmi, inc_imp, fruit_day, drinks_wk) %>%
    skim_without_charts() %>%
    yank(., "numeric") %>%
    select(var = skim_variable, n_missing, min = p0,
           median = p50, max = p100, mean, sd) %>%
    kable(digits = 1)
```

var	n_missing	min	median	max	mean	sd
bmi	0	13.3	26.8	63	27.9	6.3
inc_imp	120	216.0	48224.5	700676	75673.5	90695.8
fruit_day	41	0.0	1.1	10	1.4	1.1
drinks_wk	39	0.0	0.5	56	3.0	6.1

- Any signs of trouble? (What are we looking for?)

Quick Histogram of each quantitative variable



Code for previous slide

```
p1 <- ggplot(week2, aes(x = bmi)) +
    geom_histogram(fill = "navy", col = "white", bins = 20)
p2 <- ggplot(week2, aes(x = inc_imp)) +
    geom_histogram(fill = "forestgreen", col = "white",
                   bins = 20)
p3 <- ggplot(week2, aes(x = fruit_day)) +
    geom_histogram(fill = "tomato", col = "white", bins = 20)
p4 <- ggplot(week2, aes(x = drinks_wk)) +
    geom_histogram(fill = "royalblue", col = "white",
                   bins = 20)
(p1 + p2) / (p3 + p4)
```

I also used `warning = FALSE` in the plot's code chunk label to avoid warnings about missing values, like this one for `inc_imp`:

```
Warning: Removed 120 rows containing non-finite values
```

Binary variables in raw `week2`

```
week2 %>% tabyl(female, exerany) %>% adorn_title()
```

```
       exerany
 female       0   1 NA_
      0      95 268  20
      1     128 361  22
```

- `female` is based on biological sex (1 = female, 0 = male)
- `exerany` comes from a response to "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, don't know and refused = missing)
- Any signs of trouble here?

Binary variables in raw `week2`

```
week2 %>% tabyl(female, exerany) %>% adorn_title()
```

```
        exerany
 female       0   1 NA_
      0      95 268  20
      1     128 361  22
```

- `female` is based on biological sex (1 = female, 0 = male)
- `exerany` comes from a response to "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, don't know and refused = missing)

- Any signs of trouble here?
- I think the 1/0 values and names are OK choices.

Multicategorical `genhealth` in raw `week2`

```
week2 %>% tabyl(genhealth)
```

```
   genhealth   n     percent valid_percent
 1_Excellent 148 0.165548098    0.16573348
  2_VeryGood 324 0.362416107    0.36282195
      3_Good 274 0.306487696    0.30683091
      4_Fair 112 0.125279642    0.12541993
      5_Poor  35 0.039149888    0.03919373
        <NA>   1 0.001118568            NA
```

- The variable is based on "Would you say that in general your health is …" using the five specified categories (Excellent -> Poor), numbered for convenience after data collection.
- Don't know / not sure / refused were each treated as missing.
- How might we manage this variable?

Changing the levels for genhealth

```
week2 <- week2 %>%
    mutate(health =
              fct_recode(genhealth,
                         E = "1_Excellent",
                         VG = "2_VeryGood",
                         G = "3_Good",
                         F = "4_Fair",
                         P = "5_Poor"))
```

Might want to run a sanity check here, just to be sure...

Checking health vs. genhealth in week2

```
week2 %>% tabyl(genhealth, health) %>% adorn_title()
```

```
             health
   genhealth      E  VG   G   F  P NA_
 1_Excellent    148   0   0   0  0   0
  2_VeryGood      0 324   0   0  0   0
      3_Good      0   0 274   0  0   0
      4_Fair      0   0   0 112  0   0
      5_Poor      0   0   0   0 35   0
        <NA>      0   0   0   0  0   1
```

- OK. We've preserved the order and we have much shorter labels. Sometimes, that's helpful.

Multicategorical `race_eth` in raw `week2`

```
week2 %>% count(race_eth)
```

```
# A tibble: 6 x 2
  race_eth                      n
  <fct>                     <int>
1 Black non-Hispanic          167
2 Hispanic                     27
3 Multiracial non-Hispanic     19
4 Other race non-Hispanic      22
5 White non-Hispanic          646
6 <NA>                         13
```

"Don't know", "Not sure", and "Refused" were treated as missing.

- What is this variable actually about?

Multicategorical race_eth in raw week2

```
week2 %>% count(race_eth)
```

```
# A tibble: 6 x 2
  race_eth                      n
  <fct>                     <int>
1 Black non-Hispanic          167
2 Hispanic                     27
3 Multiracial non-Hispanic     19
4 Other race non-Hispanic      22
5 White non-Hispanic          646
6 <NA>                         13
```

"Don't know", "Not sure", and "Refused" were treated as missing.

- What is this variable actually about?
- What is the most common thing people do here?

What is the question you are asking?

Collapsing `race_eth` levels *might* be rational for *some* questions.

- We have lots of data from two categories, but only two.
- Systemic racism affects people of color in different ways across these categories, but also *within* them.
- Is combining race and Hispanic/Latinx ethnicity helpful?

It's hard to see the justice in collecting this information and not using it in as granular a form as possible, though this leaves some small sample sizes. There is no magic number for "too small a sample size."

- Most people identified themselves in one of the categories.
- These data are not ordered, and (I'd argue) ordering them isn't helpful.
- Regression models are easier to interpret, though, if the "baseline" category is a common one.

Resorting the factor for race_eth

Let's sort all five levels, from most observations to least. . .

```
week2 <- week2 %>%
    mutate(race_eth = fct_infreq(race_eth))
```

```
week2 %>% tabyl(race_eth)
```

```
                 race_eth   n    percent valid_percent
       White non-Hispanic 646 0.72259508    0.73325766
       Black non-Hispanic 167 0.18680089    0.18955732
                 Hispanic  27 0.03020134    0.03064699
  Other race non-Hispanic  22 0.02460850    0.02497162
 Multiracial non-Hispanic  19 0.02125280    0.02156640
                     <NA>  13 0.01454139            NA
```

- Not a perfect solution, certainly, but we'll try it out.

“Cleaned” Data and Missing Values

```
week2 <- week2 %>%
    select(ID, bmi, inc_imp, fruit_day, drinks_wk,
           female, exerany, health, race_eth, everything())

miss_var_summary(week2)
```

```
# A tibble: 13 x 3
   variable    n_miss pct_miss
   <chr>        <int>    <dbl>
 1 inc_imp        120   13.4
 2 exerany         42    4.70
 3 fruit_day       41    4.59
 4 drinks_wk       39    4.36
 5 race_eth        13    1.45
 6 health           1    0.112
 7 genhealth        1    0.112
 8 ID               0    0
```

Single Imputation Approach?

```
set.seed(43203)
week2im <- week2 %>%
    select(ID, bmi, inc_imp, fruit_day, drinks_wk,
           female, exerany, health, race_eth) %>%
    data.frame() %>%
    impute_cart(health ~ bmi + female) %>%
    impute_pmm(exerany ~ female + health + bmi) %>%
    impute_rlm(inc_imp + drinks_wk + fruit_day ~
                   bmi + female + health + exerany) %>%
    impute_cart(race_eth ~ health + inc_imp + bmi) %>%
    tibble()

prop_miss_case(week2im)
```

```
[1] 0
```

Saving the tidied data

Let's save both the unimputed and the imputed tidy data as R data sets.

```
saveRDS(week2, here("data", "week2.Rds"))

saveRDS(week2im, here("data", "week2im.Rds"))
```

To reload these files, we'd use `readRDS`.

- The main advantage here is that we've saved the whole R object, including all characteristics that we've added since the original download.

Splitting the Sample

Use `initial_split` from `rsample` to partition the data into:

- Model development (training) sample where we'll build models
- Model evaluation (testing) sample which we'll hold out for a while

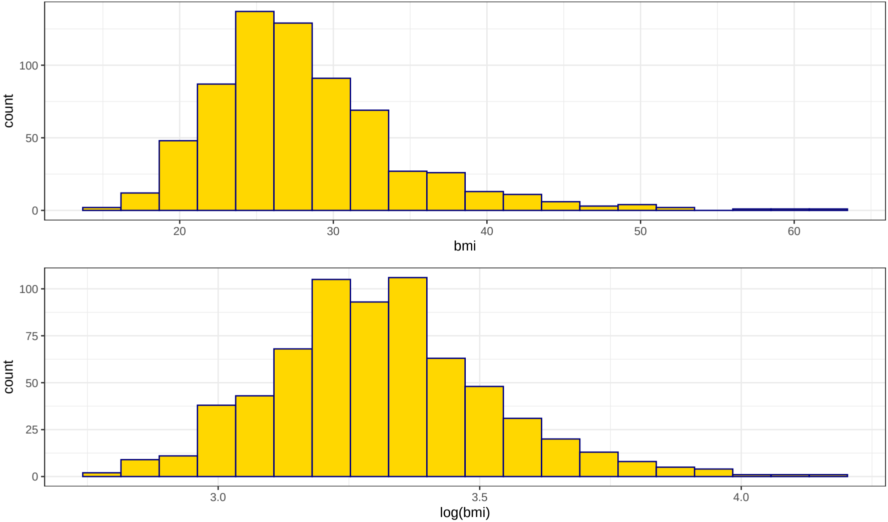
```
set.seed(432)    ## to make the work replicable in the future
week2im_split <- initial_split(week2im, prop = 3/4)

train_w2im <- training(week2im_split)
test_w2im <- testing(week2im_split)
```

```
dim(train_w2im); dim(test_w2im)
```

```
[1] 670   9
```

```
[1] 224   9
```

Should we transform our outcome?



Outcome: `bmi`, with key predictors `exerany` and `health` both categorical (two-way ANOVA!)

bmi means by exerany and health

```
summaries_1 <- train_w2im %>%
    group_by(exerany, health) %>%
    summarise(n = n(), mean = mean(bmi), stdev = sd(bmi))
summaries_1 %>% kable(digits = 2)
```

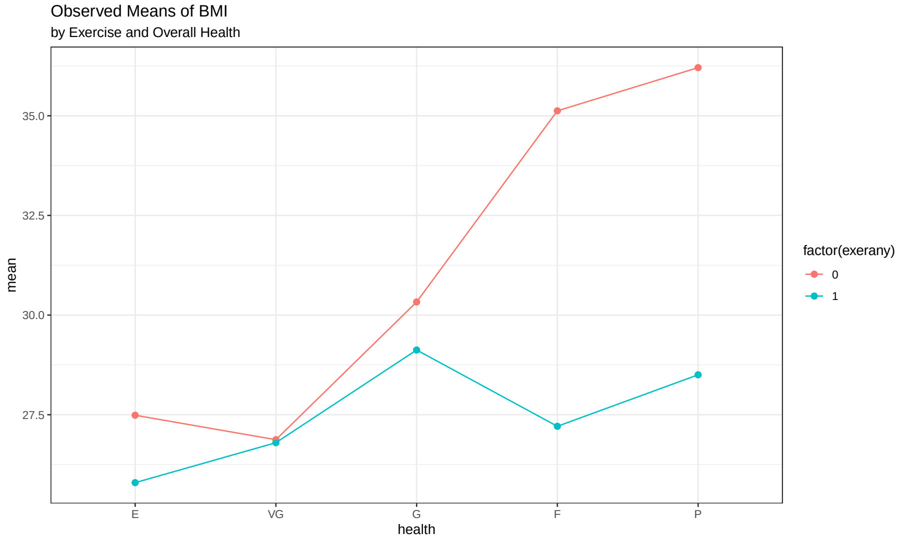
exerany	health	n	mean	stdev
0	E	18	27.49	3.56
0	VG	54	26.87	5.27
0	G	58	30.33	7.45
0	F	31	35.12	9.95
0	P	8	36.21	12.11
1	E	92	25.80	4.49
1	VG	191	26.80	4.89
1	G	152	29.12	6.26
1	F	49	27.21	5.55
1	P	17	28.50	8.61

Code for Interaction Plot

```
ggplot(summaries_1, aes(x = health, y = mean,
                        col = factor(exerany))) +
    geom_point(size = 2) +
    geom_line(aes(group = factor(exerany))) +
    labs(title = "Observed Means of BMI",
         subtitle = "by Exercise and Overall Health")
```

- Note the use of `factor` here since the `exerany` variable is in fact numeric, although it only takes the values 1 and 0.
    - Sometimes it's helpful to treat 1/0 as a factor, and sometimes not.
- Where is the evidence of serious non-parallelism (if any) in the plot on the next slide that results from this code?

Resulting Interaction Plot



Models we'll build today

- `m_1` a linear model without interaction using `exerany` and `health` to predict `bmi`
- `m_1int` add the interaction term for `exerany` and `health` to `m_1`

We'll assess these models carefully (today) in the training sample and (next time) in the test sample.

- We'll also explore adding a covariate `fruit_day` to the models in several different ways.

Fitting ANOVA model `m_1` without interaction

Building a Model (m_1) without interaction

```
m_1 <- lm(bmi ~ exerany + health,
          data = train_w2im)
```

- How well does this model fit the training data?

```
glance(m_1) %>%
    select(r.squared, adj.r.squared, sigma, nobs,
           df, df.residual, AIC, BIC) %>%
    kable(digits = c(3, 3, 2, 0, 0, 0, 1, 1))
```

r.squared	adj.r.squared	sigma	nobs	df	df.residual	AIC	BIC
0.089	0.082	6.12	670	5	664	4335.9	4367.5

ANOVA for the `m_1` model

```
anova(m_1)
```

```
Analysis of Variance Table

Response: bmi
           Df  Sum Sq Mean Sq F value    Pr(>F)    
exerany     1   895.7  895.71  23.948 1.243e-06 ***
health      4  1528.5  382.12  10.217 4.952e-08 ***
Residuals 664 24834.7   37.40                      
---
Signif. codes:  
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tidied ANOVA for the `m_1` model

```
tidy(anova(m_1)) %>%
    kable(dig = c(0, 0, 2, 2, 2, 3))
```

term	df	sumsq	meansq	statistic	p.value
exerany	1	895.71	895.71	23.95	0
health	4	1528.47	382.12	10.22	0
Residuals	664	24834.72	37.40	NA	NA

A summary of `m_1` coefficients

```
summary(m_1)$coeff
```

```
              Estimate Std. Error    t value      Pr(>|t|)
(Intercept) 27.9094987  0.7428015 37.5732944 1.704557e-166
exerany     -2.1966833  0.5501802 -3.9926613  7.262660e-05
healthVG     0.6176707  0.7026030  0.8791177  3.796555e-01
healthG      3.1372434  0.7224634  4.3424255  1.629287e-05
healthF      3.7122198  0.9070315  4.0927131  4.788419e-05
healthP      4.5514459  1.3577495  3.3521985  8.472164e-04
```

Tidied summary of m_1 coefficients

```
tidy(m_1, conf.int = TRUE, conf.level = 0.90) %>%
    kable(digits = c(0,2,2,2,3,2,2))
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	27.91	0.74	37.57	0.000	26.69	29.13
exerany	-2.20	0.55	-3.99	0.000	-3.10	-1.29
healthVG	0.62	0.70	0.88	0.380	-0.54	1.77
healthG	3.14	0.72	4.34	0.000	1.95	4.33
healthF	3.71	0.91	4.09	0.000	2.22	5.21
healthP	4.55	1.36	3.35	0.001	2.32	6.79

Equation for Model without Interaction

From `m1` our equation is ...

```
extract_eq(m_1, use_coefs = TRUE, wrap = TRUE)
```

$$
\begin{aligned}
\widehat{\text{bmi}} &= 27.91 - 2.2(\text{exerany}) + 0.62(\text{health}_{\text{VG}}) + 3.14(\text{health}_{\text{G}}) + \\
&\quad 3.71(\text{health}_{\text{F}}) + 4.55(\text{health}_{\text{P}})
\end{aligned} \tag{1}
$$

- You need to use `results = "asis"` in the code chunk label to get this to work.
- This function `extract_eq` comes from the `equatiomatic` package.

Interpreting the `m_1` model

$$\widehat{\text{bmi}} = 27.91 - 2.2(\text{exerany}) + 0.62(\text{health}_{\text{VG}}) + 3.14(\text{health}_{\text{G}}) + 3.71(\text{health}_{\text{F}}) + 4.55(\text{health}_{\text{P}}) \quad (2)$$

Name	`exerany`	`health`	predicted `bmi`
Harry	0	Excellent	27.91
Sally	1	Excellent	27.91 - 2.20 = 25.71
Billy	0	Fair	27.91 + 3.71 = 31.62
Meg	1	Fair	27.91 - 2.20 + 3.71 = 29.42

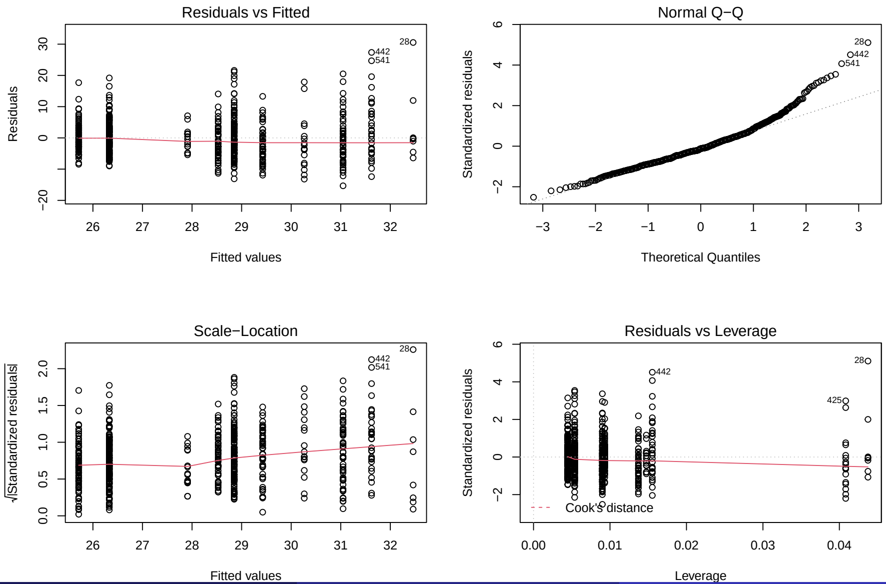
- Effect of `exerany`?
- Effect of `health` = Fair instead of Excellent?

Plot the Residuals from model `m_1`?

```
par(mfrow = c(2,2))
plot(m_1)
par(mfrow = c(1,1))
```

That's the simplest code to get the four key plots to show up in the most familiar pattern, as shown on the next slide...

`m_1` Residual Plots (conclusions?)



Fitting ANOVA model `m_1int` including interaction

Adding the interaction term to m_1

```
m_1int <- lm(bmi ~ exerany * health,
             data = train_w2im)
```

- How does this model compare in terms of fit to the training data?

```
bind_rows(glance(m_1), glance(m_1int)) %>%
    mutate(mod = c("m_1", "m_1int")) %>%
    select(mod, r.sq = r.squared, adj.r.sq = adj.r.squared,
       sigma, nobs, df, df.res = df.residual, AIC, BIC) %>%
    kable(digits = c(0, 3, 3, 2, 0, 0, 0, 1, 1))
```

mod	r.sq	adj.r.sq	sigma	nobs	df	df.res	AIC	BIC
m_1	0.089	0.082	6.12	670	5	664	4335.9	4367.5
m_1int	0.126	0.114	6.01	670	9	660	4315.8	4365.4

ANOVA for the `m_1int` model

```
tidy(anova(m_1int)) %>%
    kable(dig = c(0, 0, 2, 2, 2, 3))
```

term	df	sumsq	meansq	statistic	p.value
exerany	1	895.71	895.71	24.82	0
health	4	1528.47	382.12	10.59	0
exerany:health	4	1020.50	255.13	7.07	0
Residuals	660	23814.22	36.08	NA	NA

ANOVA test comparing `m_1` to `m_1int`

```
anova(m_1, m_1int)
```

```
Analysis of Variance Table

Model 1: bmi ~ exerany + health
Model 2: bmi ~ exerany * health
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    664 24835
2    660 23814  4    1020.5 7.0707 1.411e-05 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A summary of `m_1int` coefficients

```
summary(m_1int)$coeff
```

```
                  Estimate Std. Error    t value
(Intercept)      27.4872222   1.415826 19.4142627
exerany          -1.6917874   1.548148 -1.0927817
healthVG         -0.6140741   1.634855 -0.3756137
healthG           2.8419157   1.620700  1.7535108
healthF           7.6366487   1.780029  4.2901815
healthP           8.7202778   2.552417  3.4164785
exerany:healthVG  1.6167545   1.803846  0.8962818
exerany:healthG   0.4865311   1.804508  0.2696198
exerany:healthF  -6.2226958   2.072938 -3.0018726
exerany:healthP  -6.0145361   3.004914 -2.0015667
                     Pr(>|t|)
(Intercept)      9.227071e-67
exerany          2.748884e-01
healthVG         7.073248e-01
```

Tidied summary of `m_1int` coefficients

```
tidy(m_1int, conf.int = TRUE, conf.level = 0.90) %>%
    rename(se = std.error, t = statistic, p = p.value) %>%
    kable(digits = c(0,2,2,2,3,2,2))
```

term	estimate	se	t	p	conf.low	conf.high
(Intercept)	27.49	1.42	19.41	0.000	25.16	29.82
exerany	-1.69	1.55	-1.09	0.275	-4.24	0.86
healthVG	-0.61	1.63	-0.38	0.707	-3.31	2.08
healthG	2.84	1.62	1.75	0.080	0.17	5.51
healthF	7.64	1.78	4.29	0.000	4.70	10.57
healthP	8.72	2.55	3.42	0.001	4.52	12.92
exerany:healthVG	1.62	1.80	0.90	0.370	-1.35	4.59
exerany:healthG	0.49	1.80	0.27	0.788	-2.49	3.46
exerany:healthF	-6.22	2.07	-3.00	0.003	-9.64	-2.81
exerany:healthP	-6.01	3.00	-2.00	0.046	-10.96	-1.06

Equation for Interaction Model

From `m1_int` our equation is ...

```
extract_eq(m_1int, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{\text{bmi}} &= 27.49 - 1.69(\text{exerany}) - \\
&\quad 0.61(\text{health}_{\text{VG}}) + 2.84(\text{health}_{\text{G}}) + \\
&\quad 7.64(\text{health}_{\text{F}}) + 8.72(\text{health}_{\text{P}}) + \\
&\quad 1.62(\text{exerany} \times \text{health}_{\text{VG}}) + 0.49(\text{exerany} \times \text{health}_{\text{G}}) - \\
&\quad 6.22(\text{exerany} \times \text{health}_{\text{F}}) - 6.01(\text{exerany} \times \text{health}_{\text{P}})
\end{aligned} \tag{3}
$$

Don't forget to use `results = "asis"` in the code chunk label.

Interpreting the `m_1int` model

$$\begin{aligned}\widehat{\text{bmi}} = 27.49 - 1.69(\text{exerany}) - \\ 0.61(\text{health}_{VG}) + 2.84(\text{health}_{G}) + \\ 7.64(\text{health}_{F}) + 8.72(\text{health}_{P}) + \\ 1.62(\text{exerany} \times \text{health}_{VG}) + 0.49(\text{exerany} \times \text{health}_{G}) - \\ 6.22(\text{exerany} \times \text{health}_{F}) - 6.01(\text{exerany} \times \text{health}_{P})\end{aligned} \tag{4}$$

Name	`exerany`	`health`	predicted `bmi`
Harry	0	Excellent	27.49
Sally	1	Excellent	27.49 - 1.69 = 25.80
Billy	0	Fair	27.49 + 7.64 = 35.13
Meg	1	Fair	27.49 - 1.69 + 7.64 - 6.22 = 27.22

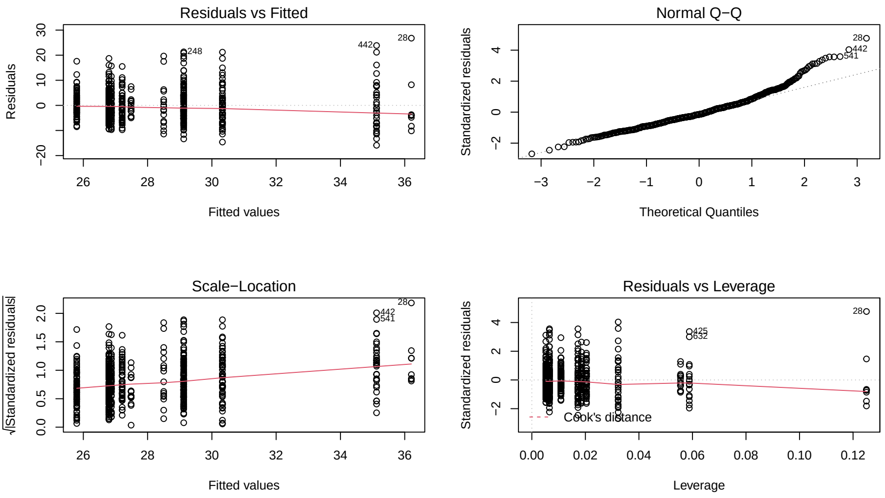
- How do we interpret effect sizes here?

Interpreting the `m_1int` model

Name	`exerany`	`health`	predicted `bmi`
Harry	0	Excellent	27.49
Sally	1	Excellent	27.49 - 1.69 = 25.80
Billy	0	Fair	27.49 + 7.64 = 35.13
Meg	1	Fair	27.49 - 1.69 + 7.64 - 6.22 = 27.22

- How do we interpret effect sizes here? **It depends**.
- Effect of `exerany`?
  - If `health` = Excellent, effect is -1.69
  - If `health` = Fair, effect is (-1.69 - 6.22) = -7.91
- Effect of `health` = Fair instead of Excellent?
  - If `exerany` = 0 (no), effect is 7.64
  - If `exerany` = 1 (yes), effect is (7.64 - 6.22) = 1.42

Plot the Residuals from model `m_1int`?



Incorporating a Covariate into our two-way ANOVA models

Taking Stock

So far, we've fit two models to predict `bmi`, using `exerany` and `health`, one with an interaction term and one without.

```
m_1 <- lm(bmi ~ exerany + health, data = train_w2im)
m_1int <- lm(bmi ~ exerany * health, data = train_w2im)
```

Next, we'll fit models incorporating a covariate, specifically, `fruit_day`, a quantity (servings/day).

- `m_2` and `m_2int` will add a linear term for `fruit_day`
- Later models (we'll fit next time) will add various non-linear terms in `fruit_day`
- We'll assess these models in our testing sample (next time) as well as our training sample.

Giving away the ending: We'll see that none of these augmented models will clearly improve the fit in our test sample over the performance of `m_1` and `m_1int`.

Adding in the covariate `fruit_day` to `m_1`

```
m_2 <- lm(bmi ~ fruit_day + exerany + health,
          data = train_w2im)
```

- How well does this model fit the training data?

```
bind_rows(glance(m_1), glance(m_2)) %>%
    mutate(mod = c("m_1", "m_2")) %>%
    select(mod, r.sq = r.squared, adj.r.sq = adj.r.squared,
        sigma, df, df.res = df.residual, AIC, BIC) %>%
    kable(digits = c(0, 3, 3, 2, 0, 0, 1, 1))
```

mod	r.sq	adj.r.sq	sigma	df	df.res	AIC	BIC
m_1	0.089	0.082	6.12	5	664	4335.9	4367.5
m_2	0.098	0.090	6.09	6	663	4331.2	4367.3

- Also available in `glance` for a model fit with `lm` are `statistic`, `p.value`, `logLik`, and `deviance`.

ANOVA for the m_2 model

```
tidy(anova(m_2)) %>%
    kable(dig = c(0, 0, 2, 2, 2, 3))
```

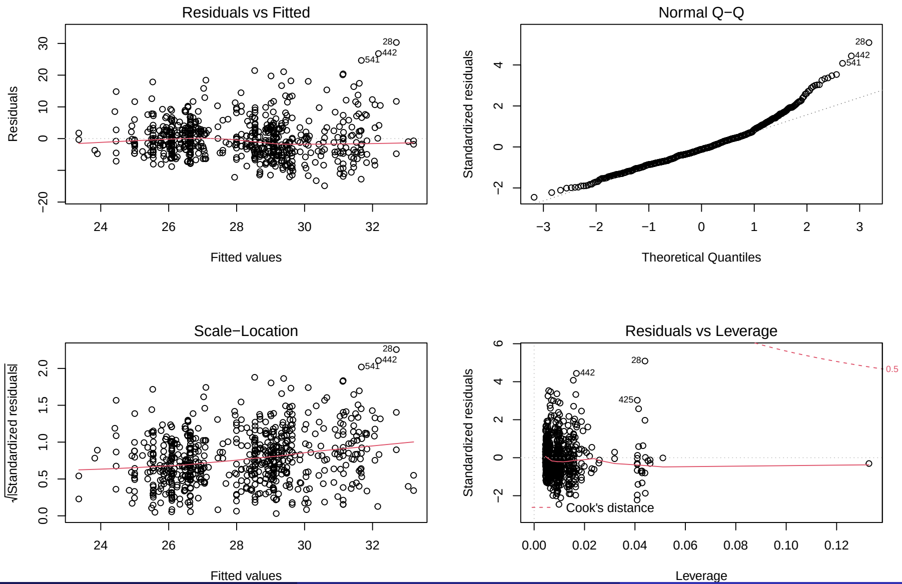
term	df	sumsq	meansq	statistic	p.value
fruit_day	1	468.10	468.10	12.62	0
exerany	1	760.50	760.50	20.51	0
health	4	1441.63	360.41	9.72	0
Residuals	663	24588.68	37.09	NA	NA

Tidied summary of `m_2` coefficients

```
tidy(m_2, conf.int = TRUE, conf.level = 0.90) %>%
    kable(digits = c(0,2,2,2,3,2,2))
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	28.68	0.80	35.94	0.000	27.37	30.00
fruit_day	-0.55	0.21	-2.58	0.010	-0.90	-0.20
exerany	-2.05	0.55	-3.71	0.000	-2.95	-1.14
healthVG	0.55	0.70	0.79	0.430	-0.60	1.71
healthG	3.00	0.72	4.16	0.000	1.81	4.19
healthF	3.55	0.91	3.92	0.000	2.06	5.04
healthP	4.57	1.35	3.38	0.001	2.34	6.79

`m_2` Residual Plots (non-constant variance?)



Who is that poorest fit case?

Plot suggests we look at row 28

```
train_w2im %>% slice(28) %>%
    select(ID, bmi, fruit_day, exerany, health) %>% kable()
```

ID	bmi	fruit_day	exerany	health
320	63	1	0	P

What is unusual about this subject?

```
train_w2im %$% sort(bmi) %>% tail()
```

```
[1] 50.46 51.22 51.54 56.31 58.98 63.00
```

What if we included the interaction term?

```r
m_2int <- lm(bmi ~ fruit_day + exerany * health,
             data = train_w2im)
```

Compare `m_2int` fit to previous models...

mod	r.sq	adj.r.sq	sigma	df	df.res	AIC	BIC
m_1	0.089	0.082	6.12	5	664	4335.9	4367.5
m_2	0.098	0.090	6.09	6	663	4331.2	4367.3
m_1int	0.126	0.114	6.01	9	660	4315.8	4365.4
m_2int	0.138	0.125	5.97	10	659	4309.1	4363.2

- `m_1` = no `fruit_day`, no `exerany*health` interaction
- `m_2` = `fruit_day`, but no interaction
- `m_1int` = no `fruit_day`, with interaction
- `m_2int` = both `fruit_day` and interaction

ANOVA for the m_2int model

```
tidy(anova(m_2int)) %>%
    kable(dig = c(0, 0, 2, 2, 2, 3))
```

term	df	sumsq	meansq	statistic	p.value
fruit_day	1	468.10	468.10	13.12	0
exerany	1	760.50	760.50	21.32	0
health	4	1441.63	360.41	10.10	0
exerany:health	4	1080.39	270.10	7.57	0
Residuals	659	23508.29	35.67	NA	NA

Tidied summary of `m_2int` coefficients

```
tidy(m_2int, conf.int = TRUE, conf.level = 0.90) %>%
    rename(se = std.error, t = statistic, p = p.value) %>%
    kable(digits = c(0,2,2,2,3,2,2))
```

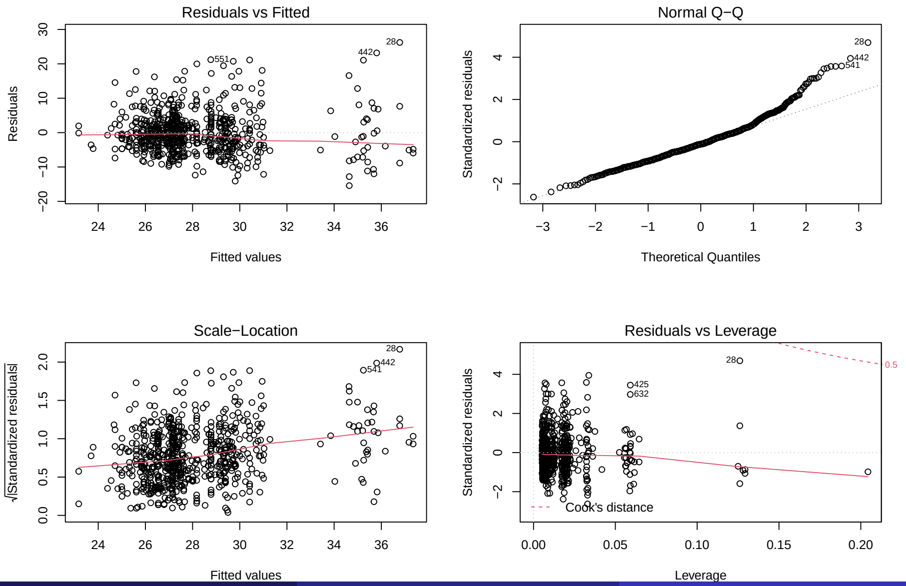
term	estimate	se	t	p	conf.low	conf.high
(Intercept)	28.28	1.43	19.73	0.000	25.91	30.64
fruit_day	-0.61	0.21	-2.93	0.004	-0.96	-0.27
exerany	-1.43	1.54	-0.93	0.353	-3.97	1.11
healthVG	-0.66	1.63	-0.40	0.686	-3.34	2.02
healthG	2.75	1.61	1.71	0.088	0.10	5.41
healthF	7.59	1.77	4.29	0.000	4.67	10.50
healthP	9.12	2.54	3.59	0.000	4.93	13.30
exerany:healthVG	1.59	1.79	0.88	0.377	-1.37	4.54
exerany:healthG	0.41	1.79	0.23	0.819	-2.54	3.37
exerany:healthF	-6.41	2.06	-3.11	0.002	-9.81	-3.02
exerany:healthP	-6.55	2.99	-2.19	0.029	-11.48	-1.62

ANOVA comparison of m_2 and m_2int

```
anova(m_2, m_2int)
```

```
Analysis of Variance Table

Model 1: bmi ~ fruit_day + exerany + health
Model 2: bmi ~ fruit_day + exerany * health
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    663 24589
2    659 23508  4    1080.4 7.5716 5.751e-06 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual plots for model `m_2int`?



Which of the four models fits best?

In the **training** sample, we have...

mod	r.sq	adj.r.sq	sigma	df	df.res	AIC	BIC
m_1	0.089	0.082	6.12	5	664	4335.9	4367.5
m_2	0.098	0.090	6.09	6	663	4331.2	4367.3
m_1int	0.126	0.114	6.01	9	660	4315.8	4365.4
m_2int	0.138	0.125	5.97	10	659	4309.1	4363.2

- Adjusted R^2, σ, AIC and BIC all improve as we move down from `m1` towards `m2_int`.
- BUT the testing sample cannot judge between models accurately. Our models have already *seen* that data.
- For fairer comparisons, we'll need to also consider the (held out) testing sample.

Next Time

- Feedback from the Minute Paper after Class 03, due tomorrow at Noon, please.
- Assessing the models we've fit so far in the testing sample
- Incorporating polynomial terms and splines into linear regression (ANCOVA) models