431 Class 21

thomaselove.github.io/431

2021-11-09

Today's R Setup

```r
library(janitor); library(knitr)
library(magrittr); library(naniar)
library(simputation) # to permit imputation work
library(broom); library(car)
library(equatiomatic); library(patchwork)
library(GGally) # to build scatterplot matrices
library(tidyverse) # always load tidyverse last

theme_set(theme_bw())

opts_chunk$set(comment=NA) # already loaded knitr
options(width = 55) # for the slides
options(dplyr.summarise.inform = FALSE)
```

Today's Agenda

- Regression Analysis: Today we focus on...
  - Data Management
  - Single Imputation
  - How well will retain our R^2 in new data?
  - Interpreting our Coefficients
  - Comparisons In-Sample via ANOVA, AIC, BIC, σ
  - "Uncertainty" Intervals around a model's coefficients
- A closer look at assumptions, and at collinearity
  - Calibrating yourself

Today's Main Example: 192 adults with diabetes in NE Ohio

Load the dm192.csv Data

```r
dm192_raw <- read_csv("data/dm192.csv") %>%
  clean_names() %>%
  mutate(across(where(is.character), ~ as.factor(.))) %>%
  mutate(pt_id = as.character(pt_id)) %>%
  select(-practice)
```

Anything wrong here?

```
glimpse(dm192_raw)
```

```
Rows: 192
Columns: 13
$ pt_id     <chr> "1", "2", "3", "4", "5", "6", "7", ~
$ sbp       <dbl> 108, 162, 135, 133, 128, 153, 134, ~
$ dbp       <dbl> 71, 92, 84, 87, 72, 71, 69, 70, 71,~
$ a1c       <dbl> 5.8, 11.6, NA, 12.7, 6.8, 5.8, 6.4,~
$ ldl       <dbl> 58, 54, NA, 112, 105, NA, 151, 98, ~
$ age       <dbl> 44, 28, 58, 56, 54, 67, 46, 62, 54,~
$ sex       <fct> male, female, female, male, female,~
$ race      <fct> black, black, black, black, white, ~
$ hisp      <fct> no, no, no, no, no, no, no, no, no,~
$ insurance <fct> medicaid, medicaid, medicare, medic~
$ statin    <dbl> 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,~
$ sbp_old   <dbl> 110, 158, 142, 145, 140, 152, 134, ~
$ a1c_old   <fct> 7.6, 12.1, 8.8, 10.9, 6.4, 6.3, 8.7~
```

Why is a1c_old a factor variable?

```
dm192_raw %>% count(a1c_old)
```

```
# A tibble: 75 x 2
   a1c_old     n
   <fct>   <int>
 1 #VALUE!     2
 2 10          3
 3 10.1        4
 4 10.2        1
 5 10.3        1
 6 10.4        2
 7 10.5        1
 8 10.6        1
 9 10.7        2
10 10.9        3
# ... with 65 more rows
```

Let’s try importing that again. . .

```
dm192_raw <- read_csv("data/dm192.csv") %>%
  clean_names() %>%
  mutate(a1c_old =
           ifelse(a1c_old == "#VALUE!", NA, a1c_old)) %>%
  mutate(a1c_old = as.numeric(a1c_old)) %>%
  mutate(across(where(is.character), ~ as.factor(.))) %>%
  mutate(pt_id = as.character(pt_id)) %>%
  select(-practice)
```

Now, how do things look?

```
head(dm192_raw)
```

```
# A tibble: 6 x 13
  pt_id   sbp   dbp   a1c   ldl   age sex   race  hisp
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <fct> <fct> <fct>
1 1       108    71   5.8    58    44 male  black no
2 2       162    92  11.6    54    28 fema~ black no
3 3       135    84  NA      NA    58 fema~ black no
4 4       133    87  12.7   112    56 male  black no
5 5       128    72   6.8   105    54 fema~ white no
6 6       153    71   5.8    NA    67 male  black no
# ... with 4 more variables: insurance <fct>,
#   statin <dbl>, sbp_old <dbl>, a1c_old <dbl>
```

```
miss_var_summary(dm192_raw)
```

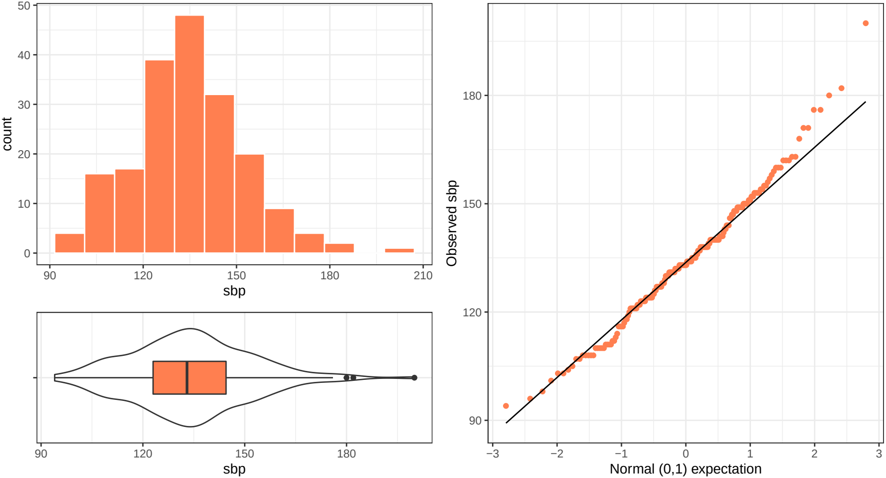
```
# A tibble: 13 x 3
   variable  n_miss pct_miss
   <chr>      <int>    <dbl>
 1 ldl           43    22.4 
 2 a1c            4     2.08
 3 a1c_old        4     2.08
 4 hisp           2     1.04
 5 pt_id          0     0   
 6 sbp            0     0   
 7 dbp            0     0   
 8 age            0     0   
 9 sex            0     0   
10 race           0     0   
11 insurance      0     0   
12 statin         0     0   
13 sbp_old        0     0   
```

Single Imputation into dm192

```
dm192 <- dm192_raw %>%
  impute_cart(hisp ~ race) %>%
  impute_rlm(ldl ~ age + statin) %>%
  impute_pmm(a1c ~ ldl + age) %>%
  impute_rlm(a1c_old ~ a1c)
```

- Why do I not need to set a seed here?

Systolic BP values in `dm192` (our outcome)



Scatterplot Matrix?

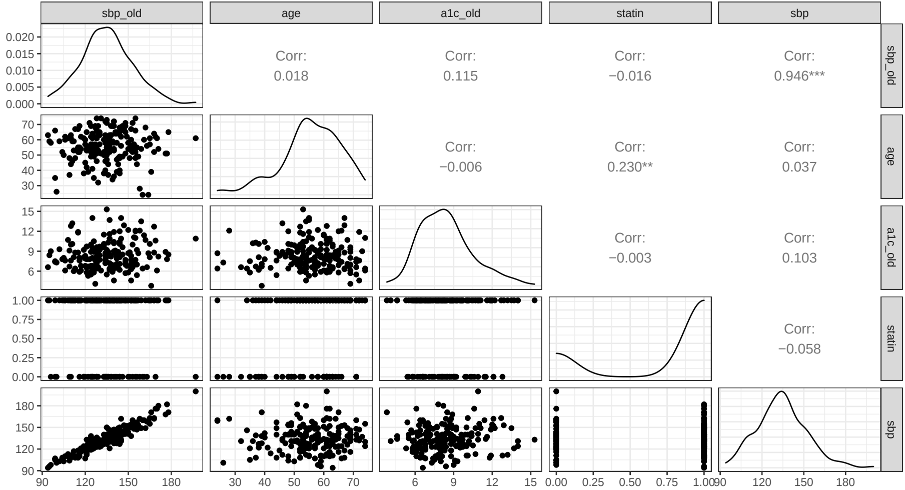
Too many predictors to fit on one slide, so I'll split them...

```
dm192 %>% select(sbp_old, age, a1c_old, statin, sbp) %>%
  ggpairs(.,
          lower=list(combo=wrap("facethist", binwidth=0.8)))
```

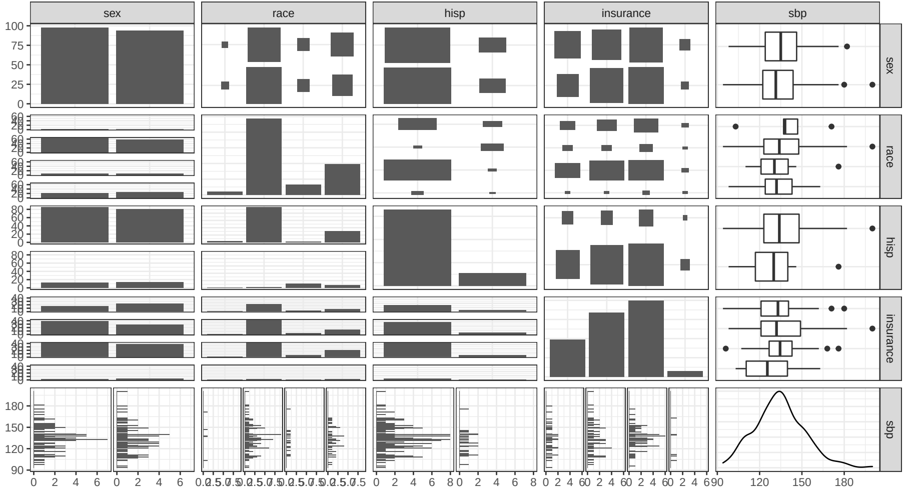
```
dm192 %>% select(sex, race, hisp, insurance, sbp) %>%
  ggpairs(.,
          lower=list(combo=wrap("facethist", binwidth=0.8)))
```

- Adding the `lower = ...` stuff eliminates the binwidth message.
- Note that I list my outcome `sbp` last in each plot. Why?

Scatterplot Matrix 1 (numerical predictors)



Scatterplot Matrix 2 (categorical predictors)



Consider a transformation of our outcome sbp?

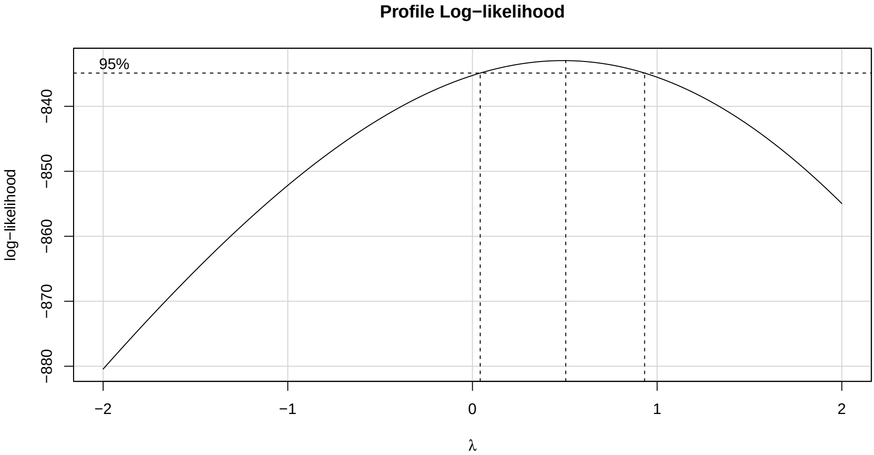
- What are the predictors we'll consider?
  - sbp_old
  - age
  - sex
  - race
  - hisp
  - insurance
  - statin
  - a1c_old

So we could fit that model and look at the Box-Cox results...

```
m0 <- lm(sbp ~ sbp_old + age + sex + race + hisp +
             insurance + statin + a1c_old, data = dm192)
```

Box-Cox results (requires `car` package)

```
boxCox(m0)
```



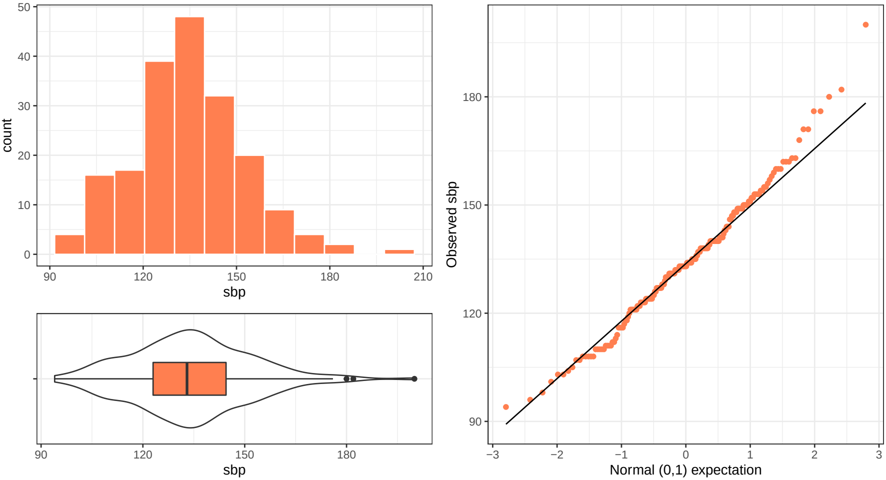
Power Transformation results (for `m0`)

```
summary(powerTransform(m0))
```

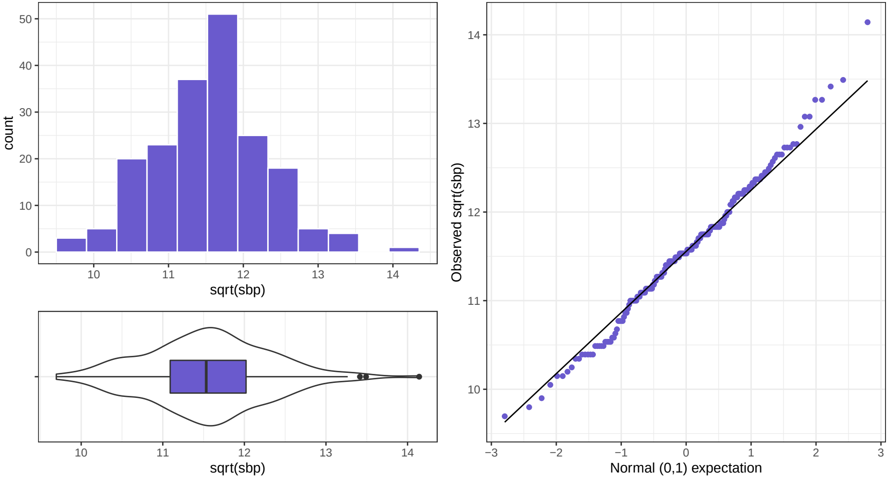
```
bcPower Transformation to Normality
   Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
Y1     0.491         0.5       0.0471        0.935

Likelihood ratio test that transformation parameter is equal t
 (log transformation)
                           LRT df     pval
LR test, lambda = (0) 4.571498  1 0.032508

Likelihood ratio test that no transformation is needed
                           LRT df     pval
LR test, lambda = (1) 5.113835  1 0.023736
```

Note: I suppressed a warning here. What to do?

Systolic BP values in `dm192` (untransformed)



Square Root of Systolic BP values in `dm192`



What if I was going to fit a model to $\sqrt{sbp}$?

Before anything else, I'd create a new variable containing the transformed outcome.

```
dm192 <- dm192 %>%
    mutate(sbp_sqrt = sqrt(sbp))
```

and I'd split the data after this, build a scatterplot matrix, run models, all based on this, I think.

So, for instance, if the choice is to run:

```
modelX <- lm(sqrt(sbp) ~ statin + sbp_old, data = dm192)
```

or

```
modelX <- lm(sbp_sqrt ~ statin + sbp_old, data = dm192)
```

it doesn't matter, actually, but for anything else (like a scatterplot matrix, for instance), I'd use `sbp_sqrt`.

Two models for sbp in the dm192 data

For now, let's hold off on a square root transformation - we'll return to this later.

```
m1 <- lm(sbp ~ sbp_old + statin, data = dm192)
m2 <- lm(sbp ~ sbp_old + age + sex + race + hisp +
             insurance + statin + a1c_old, data = dm192)
```

Stepwise Variable Selection?

I'll just note here that if you start with m2, and run

```
step(m2)
```

you wind up with m1. That's how I came up with them as candidate models.

Model Equations with `equatiomatic`

Remember to include ‘results = ’asis“ in the code chunk.

```
extract_eq(m1, use_coefs = TRUE, coef_digits = 3)
```

$$\widehat{\text{sbp}} = 8.63 + 0.94(\text{sbp\_old}) - 1.775(\text{statin}) \tag{1}$$

```
extract_eq(m2, use_coefs = TRUE,
           wrap = TRUE, terms_per_line = 3)
```

$$\begin{aligned}
\widehat{\text{sbp}} &= 3.01 + 0.94(\text{sbp\_old}) + 0.09(\text{age})\ + \\
&\quad 0.06(\text{sex}_{\text{male}}) + 1.5(\text{race}_{\text{black}}) + 2.78(\text{race}_{\text{other}})\ + \\
&\quad 0.67(\text{race}_{\text{white}}) - 0.13(\text{hisp}_{\text{yes}}) + 0.86(\text{insurance}_{\text{medicaid}})\ - \\
&\quad 0.38(\text{insurance}_{\text{medicare}}) + 1.6(\text{insurance}_{\text{uninsured}}) - 2.27(\text{statin})\ - \\
&\quad 0.1(\text{a1c\_old})
\end{aligned} \tag{2}$$

Which model looks better, by R^2 and Adjusted R^2?

```r
g1 <- glance(m1) %>% mutate(model = "m1")
g2 <- glance(m2) %>% mutate(model = "m2")
comp <- bind_rows(g1, g2)

comp %>% select(model, r.squared, adj.r.squared)
```

```
# A tibble: 2 x 3
  model r.squared adj.r.squared
  <chr>     <dbl>         <dbl>
1 m1        0.897         0.896
2 m2        0.900         0.894
```

- Which of these two models is more likely to **retain its nominal** R^2 **value** in new data?

Which model looks better, by σ, AIC or BIC?

```
comp %>% select(model, sigma, AIC, BIC)
```

```
# A tibble: 2 x 4
  model sigma   AIC   BIC
  <chr> <dbl> <dbl> <dbl>
1 m1     5.72 1220. 1233.
2 m2     5.80 1234. 1280.
```

Is one model detectably better that the other in-sample?

```
anova(m1, m2)
```

```
Analysis of Variance Table

Model 1: sbp ~ sbp_old + statin
Model 2: sbp ~ sbp_old + age + sex + race + hisp + insurance +
    a1c_old
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    189 6192.1
2    179 6022.8 10    169.28 0.5031 0.8863
```

Interpreting the slopes and their CIs

```
tidy(m1, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 2)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	8.63	3.25	3.25	14.01
sbp_old	0.94	0.02	0.90	0.98
statin	-1.78	0.98	-3.39	-0.16

- Averaging over repeated samples, we can be 90% confident that the true slope of `sbp_old` lies between 0.90 and 0.98, assuming that `statin` is unchanged.

Model m1, and 90% uncertainty intervals

Let's describe these as *uncertainty intervals*, since they are meant to help you understand how much uncertainty you have.

```
tidy(m1, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 2)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	8.63	3.25	3.25	14.01
sbp_old	0.94	0.02	0.90	0.98
statin	-1.78	0.98	-3.39	-0.16

"Uncertainty Interval" is kind of nice because it fights the ambiguity between confidence intervals and predictive intervals. Also notice that confidence intervals are smaller when you have more confidence, which can confuse people. (references in a few slides)

Model m2, and 50% uncertainty intervals (Code)

```
tidy(m2, conf.int = TRUE, conf.level = 0.50) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 2)
```

50% intervals have some potential advantages over 95% intervals. . .

- Computational Stability
- More intuitive (half the 50% intervals should contain the true value)
- Sometimes it's best to get a sense of where the parameters will be, not to attempt an unrealistic near-certainty.

Model `m2`, and 50% uncertainty intervals (Results)

term	estimate	std.error	conf.low	conf.high
(Intercept)	3.01	5.77	-0.89	6.91
sbp_old	0.94	0.02	0.93	0.96
age	0.09	0.06	0.05	0.13
sexmale	0.06	0.86	-0.52	0.64
raceblack	1.50	2.75	-0.36	3.36
raceother	2.78	3.23	0.60	4.96
racewhite	0.67	2.84	-1.26	2.59
hispyes	-0.13	1.66	-1.26	0.99
insurancemedicaid	0.86	1.25	0.01	1.70
insurancemedicare	-0.38	1.21	-1.20	0.43
insuranceuninsured	1.60	2.66	-0.19	3.40
statin	-2.27	1.03	-2.96	-1.57
a1c_old	-0.10	0.21	-0.25	0.04

Model m1, and 50% uncertainty intervals

```
tidy(m1, conf.int = TRUE, conf.level = 0.50) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  knitr::kable(digits = 2)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	8.63	3.25	6.43	10.83
sbp_old	0.94	0.02	0.92	0.96
statin	-1.78	0.98	-2.43	-1.12

Under repeated sampling, the true slope of `sbp_old` will be inside (0.92, 0.96) half of the time, assuming `statin` is held constant.

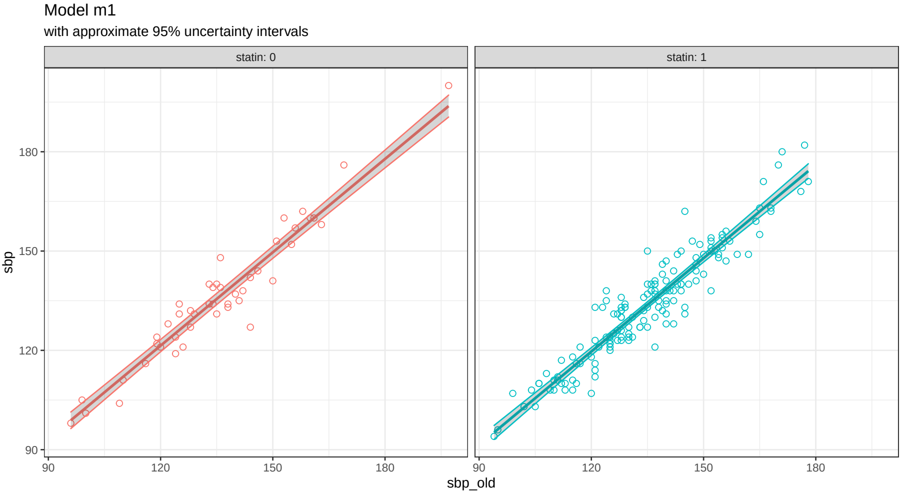
Andrew Gelman Blog Posts Worth a Little Time

- Instead of "confidence interval," let's say "uncertainty interval" at https://andrewgelman.com/2010/12/21/lets_say_uncert/
- "Why I prefer 50% rather than 95% intervals" at https://andrewgelman.com/2016/11/05/why-i-prefer-50-to-95-intervals/
- "Abraham Lincoln and confidence intervals" at https://andrewgelman.com/2016/11/23/abraham-lincoln-confidence-intervals/

Plotting Model m1 (Code)

```
m1_aug <- augment(m1, data = dm192, se_fit = TRUE)

ggplot(m1_aug, aes(x = sbp_old, y = sbp,
                   col = factor(statin))) +
  geom_point(pch = 1, size = 2) +
  geom_line(aes(y = .fitted), size = 1) +
  geom_ribbon(aes(ymin = .fitted - .se.fit*2,
                  ymax = .fitted + .se.fit*2),
              alpha = 0.2) +
  facet_wrap(~ statin, labeller = "label_both") +
  guides(col = "none") +
  labs(title = "Model m1",
       subtitle = "with approximate 95% uncertainty intervals'
```

Plotting Model `m1` (Result)



Residual Plots and Regression Assumptions

Multivariate Regression: Checking Assumptions

Assumptions (see Course Notes, Section 27)

- Linearity
- Normality
- Homoscedasticity
- Independence

Available Residual Plots

```
plot(model, which = c(1:3,5))
```

1. Residuals vs. Fitted Values
2. Normal Q-Q Plot of Standardized Residuals
3. Scale-Location Plot
4. Index Plot of Cook's Distance
5. Residuals, Leverage and Influence

An Idealized Model (by Simulation)

```
set.seed(431122)

x1 <- rnorm(200, 20, 5)
x2 <- rnorm(200, 20, 12)
x3 <- rnorm(200, 20, 10)
er <- rnorm(200, 0, 1)
y <- .3*x1 - .2*x2 + .4*x3 + er

sim0 <- tibble(y, x1, x2, x3)

mod0 <- lm(y ~ x1 + x2 + x3, data = sim0)

summary(mod0) # appears on next slide
```

```
Call:
lm(formula = y ~ x1 + x2 + x3, data = sim0)
```

An Idealized Model (by Simulation)

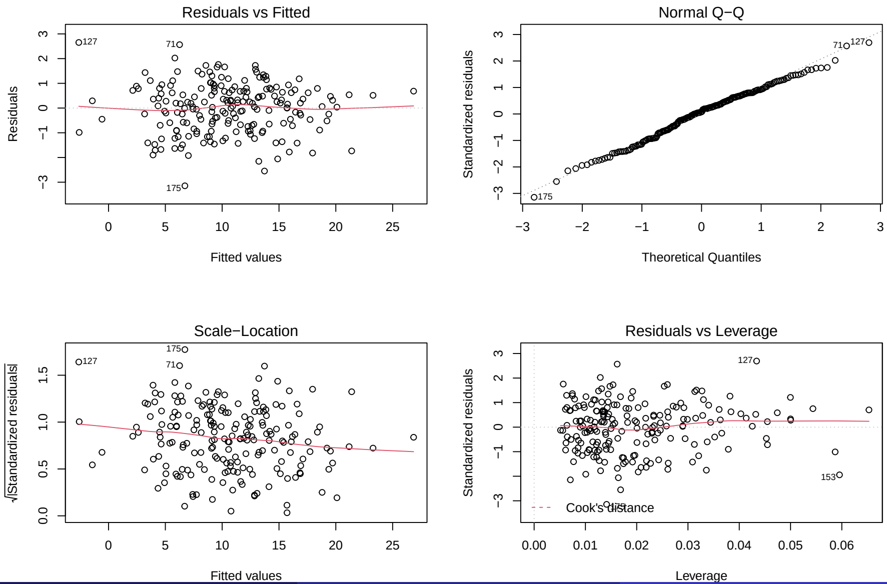
```
Call: lm(formula = y ~ x1 + x2 + x3, data = sim0)

Residuals:     Min        1Q   Median        3Q       Max
         -3.14553 -0.68079  0.08096  0.69216  2.65265

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.122852   0.348584   0.352    0.725
x1            0.285539   0.014211  20.093   <2e-16 ***
x2           -0.204908   0.005828 -35.159   <2e-16 ***
x3            0.413308   0.007172  57.631   <2e-16 ***
---
Signif codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.007 on 196 degrees of freedom
Multiple R-squared:  0.9589,    Adjusted R-squared:  0.9583
F-statistic:  1524 on 3 and 196 DF,  p-value: < 2.2e-16
```

Building Residual Plots for Idealized Model

```
par(mfrow=c(2,2))
plot(mod0)
par(mfrow=c(1,1))
```

- Residuals vs. Fitted values (Top Left)
- Normal Q-Q plot of Standardized Residuals (Top Right)
- Scale-Location plot (Bottom Left)
- Residuals vs. Leverage, Cook's Distance contours (Bottom Right)

Residual Analysis (Idealized Model: n = 200)



Is one of the regression assumptions violated?

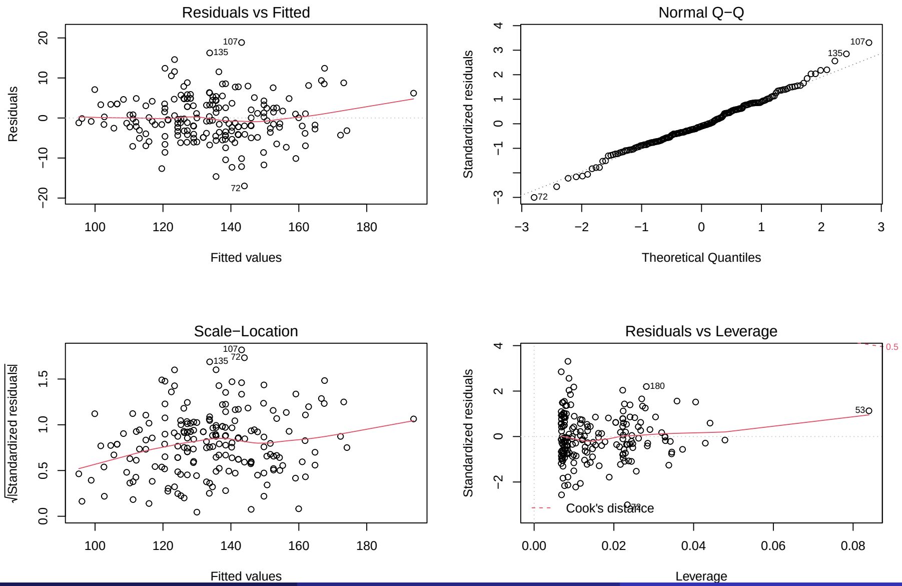
- Non-linearity problems
  - show up as a curve in the Top Left plot (Residuals vs. Fitted)
- Heteroscedasticity problems
  - show up as a fan in the Top Left plot
  - show up as a trend (up or down) in the Scale-Location plot
- Non-Normality problems
  - shows up as individual outliers in all plots
  - Normal Q-Q plot describes skew / many outliers / a few big outliers
  - Bottom Right plot shows each point's residual, leverage and influence

What to Do?

Importance of Assumptions:

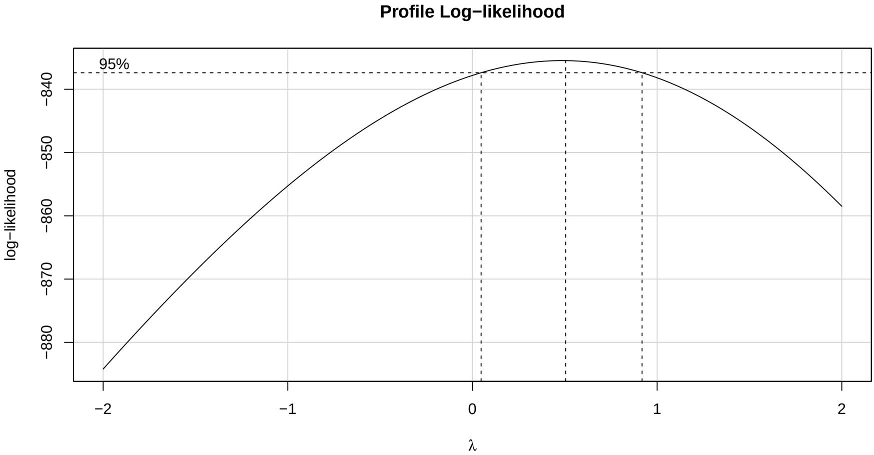
1. Linearity (critical, but amenable to transformations, often)
2. Independence (critical, not relevant if data are a cross-section with no meaningful ordering in space or time, but vitally important if space/time play a meaningful role - longitudinal data analysis required)
3. Homoscedasticity (constant variance: important, sometimes amenable to transformation)
4. Normality due to skew (usually amenable to transformation)
5. Normality due to many more outliers than we would expect (heavy-tailed - inference is problematic unless you account for this, sometimes a transformation can help)
6. Normality due to a severe outlier (or a small number of severely poorly fitted points - can consider setting those points away from modeling, but requires a meaningful external explanation)

Residual Plots for Model `m1` for our `dm192` data?



Recall the Box-Cox plot's suggestion, earlier. . .

```
boxCox(m1)
```

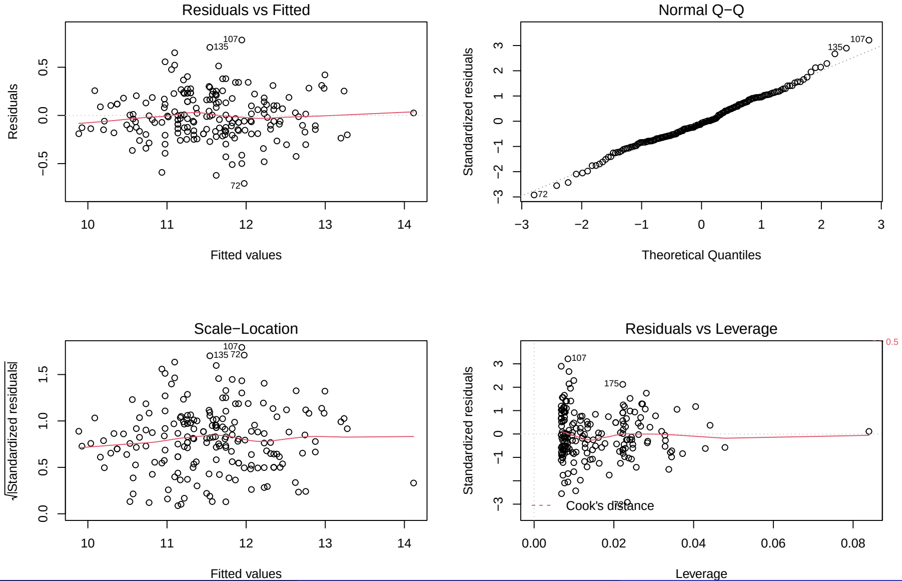


Adjusted model `m1` predicting $\sqrt{sbp}$

```r
m1_adj <- lm(sqrt(sbp) ~ sbp_old + statin, data = dm192)
summary(m1_adj)
```

```

Call:
lm(formula = sqrt(sbp) ~ sbp_old + statin, data = dm192)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.70616 -0.15787 -0.01856  0.16667  0.78269 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.1590466  0.1390482  44.294   <2e-16 ***
sbp_old      0.0403926  0.0009897  40.813   <2e-16 ***
statin      -0.0707505  0.0416825  -1.697   0.0913 .  
---
Signif. codes:
```

Residuals for `m1` predicting square root of `sbp`



Resolving Assumption Violations

Options include:

- transform the Y variable, likely with one of our key power transformations (use Box-Cox to help)
- transform one or more of the X variables if it seems particularly problematic, or perhaps combine them (rather than height and weight, perhaps look at BMI, or BMI and height to help reduce collinearity)
- remove a point only if you have a good explanation for the point that can be provided outside of the modeling, and this is especially important if the point is influential
- consider other methods for establishing a non-linear model (432: splines, loess smoothers, non-linear modeling)
- consider other methods for longitudinal data with substantial dependence (432)

Six Simulations To Help You Calibrate Yourself

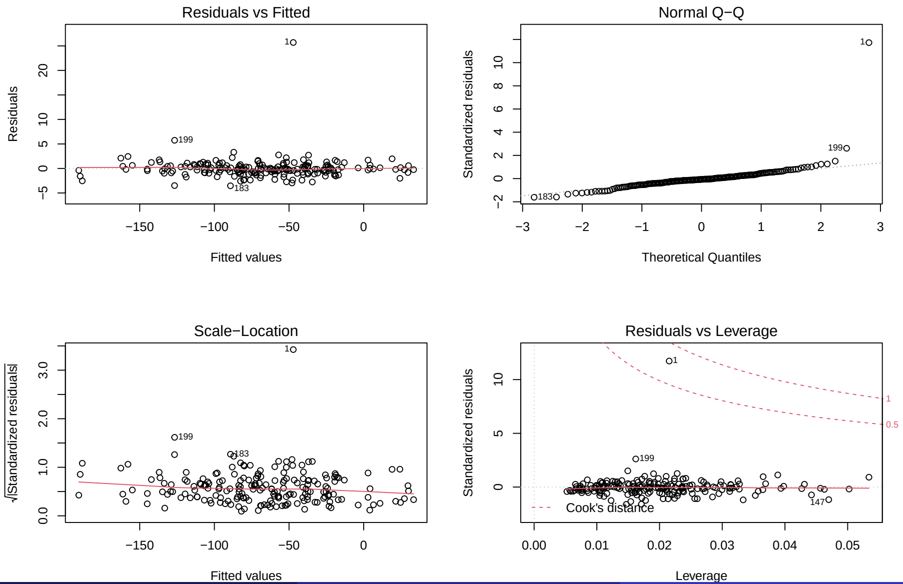
For each simulation, decide on the following:

Is one of the regression assumptions violated?

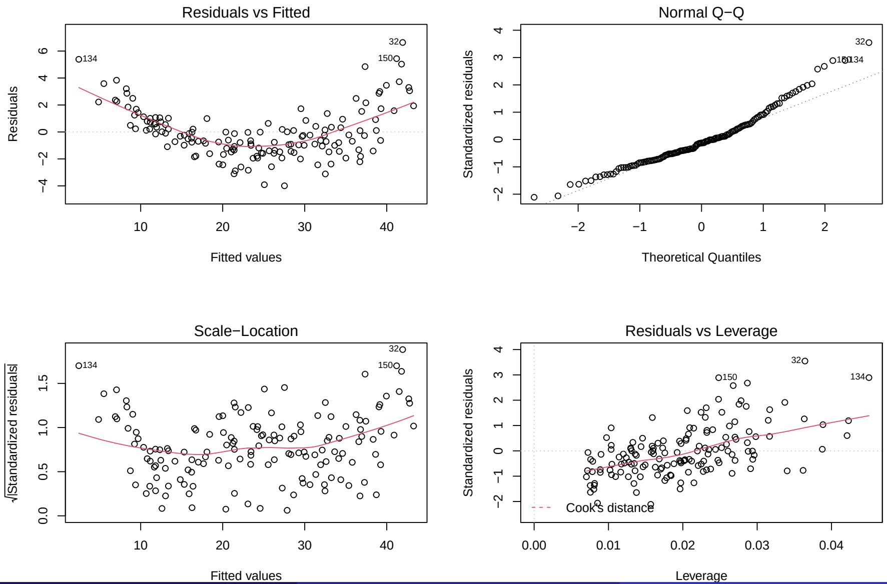
- Linearity, Homoscedasticity, Normality, or multiple problems?
    - All of these simulations describe cross-sectional data, with no importance to the order of the observations, so the assumption of independence isn't a concern.
- In which of the four plot(s) shown do you see the problem?
    - Top Left: Residuals vs. Fitted values (in R: plot 1)
    - Top Right: Normal Q-Q plot of Standardized Residuals (plot 2)
    - Bottom Left: Scale-Location plot (plot 3)
    - Bottom Right: Residuals vs. Leverage, Cook's Distance contours (plot 5)
- If you see a point that is problematic, then:
    - is it poorly fit?
    - is it highly leveraged?
    - is it influential?
- What might you try to do about the assumption problem you see (if you see one), to resolve it?

This **isn't** easy. We'll do three, and then regroup.

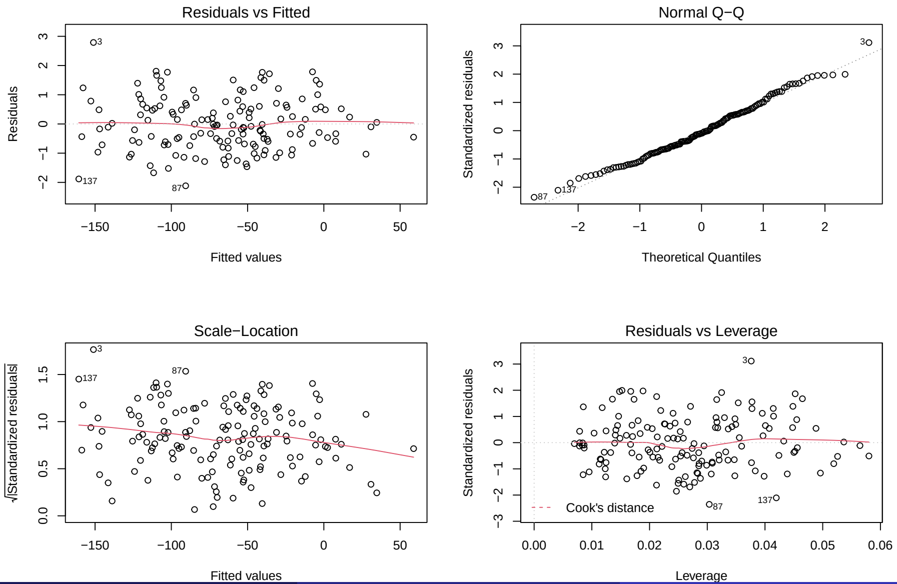
Simulation 1 (n = 200 subjects)



Simulation 2 (n = 150)



Simulation 3 (n = 150)



OK. How are we doing so far?

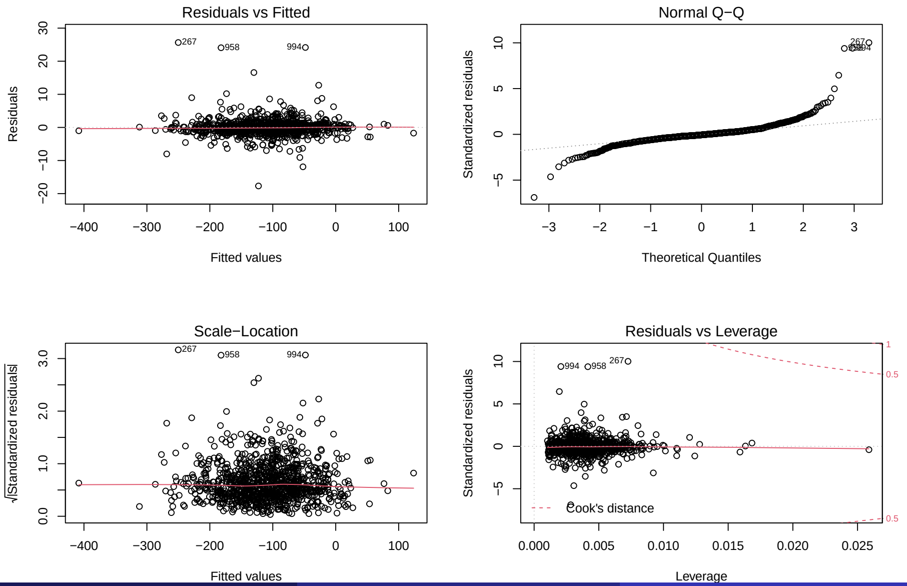
The First Three Simulations

For those of you playing along at home. . .

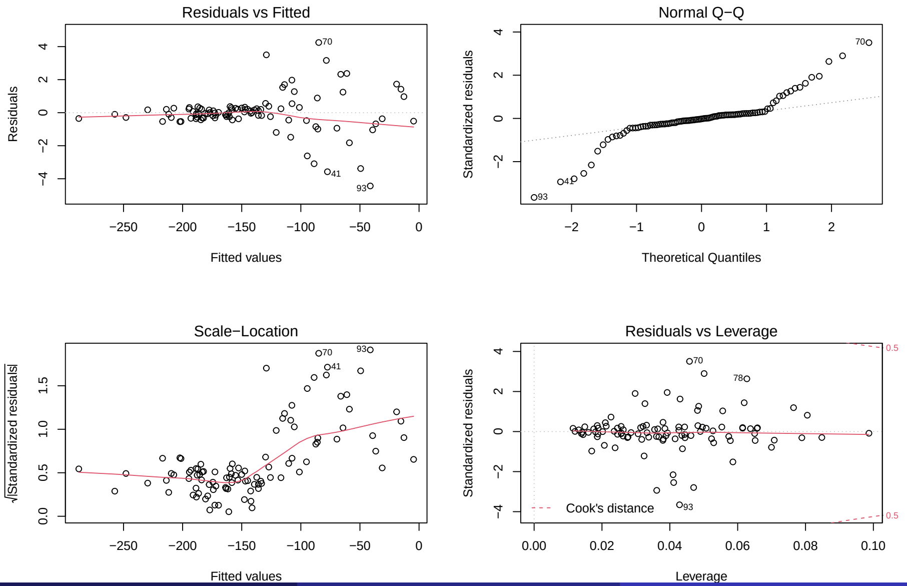
1. Observation 1 has an impossibly large standardized residual (Z score is close to 12), of substantial influence (Cook's distance around 0.7).
    - Probably need to remove the point, and explain it separately.
2. Curve in residuals vs. fitted values plot suggests potential non-linearity.
    - Natural choice would be a transformation of the outcome.
3. No substantial problems, although there's a little bit of heteroscedasticity.
    - I'd probably just go with the model as is.

Let's try three more. . .

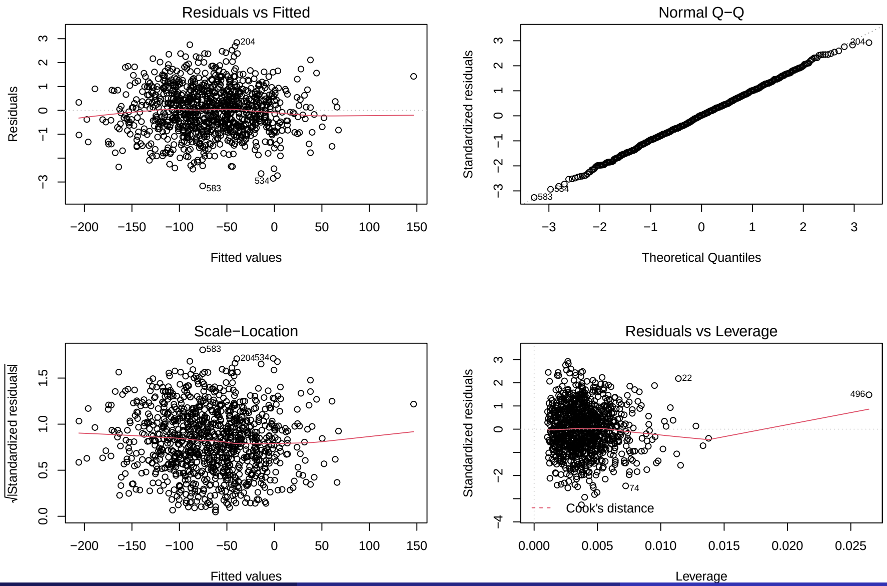
Simulation 4 (n = 1000)



Simulation 5 (n = 100)



Simulation 6 (n = 1000)



OK. How did this go?

The Last Three Simulations

For those of you playing along at home...

4. Normality issues - outlier-prone even with 1000 observations.
   - Transform Y? Consider transforming the Xs?
5. Serious heteroscedasticity - residuals much more varied for larger fitted values.
   - Look at Residuals vs. each individual X to see if this is connected to a specific predictor, which might be skewed or something?
6. No serious violations - point 496 has very substantial leverage, though.
   - I'd probably just go with the model as is, after making sure that point 496's X values aren't incorrect.

What about Collinearity?

Do we have collinearity in our `dm192` models?

```r
# requires library(car)
vif(m1)
```

```
 sbp_old   statin
1.000271 1.000271
```

```r
car::vif(m2)
```

```
              GVIF Df GVIF^(1/(2*Df))
sbp_old   1.053618  1        1.026459
age       1.790969  1        1.338271
sex       1.047137  1        1.023297
race      2.155455  3        1.136553
hisp      1.909531  1        1.381858
insurance 1.921500  3        1.114996
statin    1.084574  1        1.041429
a1c_old   1.094353  1        1.046113
```

What about collinearity?

"No collinearity" is not a regression assumption, but if we see substantial collinearity, we are inclined to consider dropping some of the variables, or combining them (height and weight may be highly correlated, height and BMI may be less so).

The variance inflation factor (or VIF), if it exceeds 5, is a clear indication of collinearity. We'd like to see the variances inflated only slightly (that is, VIF not much larger than 1) by correlation between the predictors, to facilitate interpretation.

The best way to tell if you've improved the situation by fitting an alternative model is to actually compare and fit the two models, looking in particular at:

- the standard errors of their coefficients, and
- their VIFs.

What's the Goal Here?

Develop an effective model. (?) (!)

- Models can do many different things. What you're using the model for matters, a lot.
- Don't fall into the trap of making binary decisions (this model isn't perfect, no matter what you do, and so your assessment of residuals will also have shades of gray).
- The tools we have provided (scatterplots, mostly) are well designed for rather modest sample sizes. When you have truly large samples, they don't scale very well.
- Just because R chooses four plots for you to study doesn't mean they provide the only relevant information.
- Embrace the uncertainty. Look at it as an opportunity to study your data more effectively.