431 Class 11

thomaselove.github.io/431

2020-09-29

Today's R Packages

```
library(NHANES)
library(car) # for Box-Cox transformation methods
library(janitor)
library(knitr)
library(broom)
library(magrittr)
library(patchwork)
library(ggrepel)
library(tidyverse)

theme_set(theme_bw())
```

What we're discussing

- The central role of linear regression in understanding associations between quantitative variables.
- The interpretation of a regression model as a prediction model.
- The meaning of key regression summaries, including residuals.
- Using tidy and glance from the broom package to help with summaries.
- Measuring association through correlation coefficients.
- How we might think about "adjusting" for the effect of a categorical predictor on a relationship between two quantitative ones.
- How a transformation might help us "linearize" the relationship shown in a scatterplot.

nh3_new data (n = 989, 17 variables)

```r
set.seed(20200914)

nh3_new <- NHANES %>%
    filter(SurveyYr == "2011_12") %>%
    select(ID, SurveyYr, Age, Height, Weight, BMI, Pulse,
           SleepHrsNight, BPSysAve, BPDiaAve, Gender,
           PhysActive, SleepTrouble, Smoke100,
           Race1, HealthGen, Depressed) %>%
    rename(Subject = ID, SleepHours = SleepHrsNight,
           Sex = Gender, SBP = BPSysAve, DBP = BPDiaAve) %>%
    filter(Age > 20 & Age < 80) %>%
    drop_na() %>%
    distinct() %>%
    slice_sample(n = 1000) %>%
    clean_names() %>%
    filter(dbp > 39) %>%
    mutate(subject = as.character(subject))
```

Today's Data (nh4)

```r
set.seed(431)

nh4 <- nh3_new %>%
  select(subject, sbp, dbp, age, smoke100, race1) %>%
  slice_sample(n = 800, replace = FALSE)
```

- Outcome (quantitative): `sbp`
- Quantitative predictors: `dbp`, `age`
- Binary predictor: `smoke100` (Yes/No)
- 5-category predictor: `race1` (White, Black, Hispanic, Mexican, Other)
- Identification code: `subject`

Models we've seen for sbp

```
mod_1 <- lm(sbp ~ dbp, data = nh4)
nh4_aug1 <- augment(mod_1, data = nh4)

mod_2 <- lm(sbp ~ dbp + age, data = nh4)
nh4_aug2 <- augment(mod_2, data = nh4)
```

- In model mod_2 we're **adjusting** for the effect of age on the sbp - dbp association.
- Next, in model mod_3, we'll also adjust for the effect of smoke100, a categorical (binary) variable.

Model mod_3: add smoke100 as a predictor

```r
mod_3 <- lm(sbp ~ dbp + age + smoke100, data = nh4)
mod_3
```

```

Call:
lm(formula = sbp ~ dbp + age + smoke100, data = nh4)

Coefficients:
(Intercept)          dbp          age  smoke100Yes
    49.1120       0.7497       0.3743       2.3807
```

Interpreting the binary predictor (smoke100) and its slope

- smoke100 was binary: either Yes or No for all subjects, so...
  - smoke100Yes = 1 if smoke100 is Yes, and
  - smoke100Yes = 0 if smoke100 is No.

Prediction for subject 65867?

subject	sbp	dbp	age	smoke100
65867	115	78	60	No

From Model 3, our predicted `sbp` for subject 65867 will be:

49.112 + 0.750 dbp + 0.374 age + 2.381 (indicator of smoke100 = Yes)

So for subject 65867, we'd predict:

$49.112 + 0.750\ (78) + 0.374\ (60) + 2.381\ (0) = 130.05$ mm Hg

augment for mod_3

```
nh4_aug3 <- augment(mod_3, data = nh4)

nh4_aug3 %>% head(4) %>%
  select(subject, sbp, dbp, age, smoke100, .fitted, .resid) %>
  kable()
```

subject	sbp	dbp	age	smoke100	.fitted	.resid
65867	115	78	60	No	130.0450	-15.04496
70046	125	83	55	No	131.9221	-6.92205
64302	98	59	45	No	110.1866	-12.18657
69386	141	68	52	Yes	121.9345	19.06549

Compare `mod_2` coefficients to `mod_3` via tidy?

Here is `mod_2` with 90% confidence intervals:

term	estimate	std.error	conf.low	conf.high
(Intercept)	49.59	3.17	44.37	54.81
dbp	0.75	0.04	0.69	0.82
age	0.38	0.03	0.33	0.43

And here is `mod_3`, also with 90% confidence intervals:

term	estimate	std.error	conf.low	conf.high
(Intercept)	49.11	3.17	43.90	54.33
dbp	0.75	0.04	0.68	0.82
age	0.37	0.03	0.33	0.42
smoke100Yes	2.38	0.93	0.84	3.92

`glance` for our 3 models so far

Model `mod_1`: `dbp` only

r.squared	adj.r.squared	sigma	AIC	BIC
0.291	0.29	14.4	6542.4	6556.4

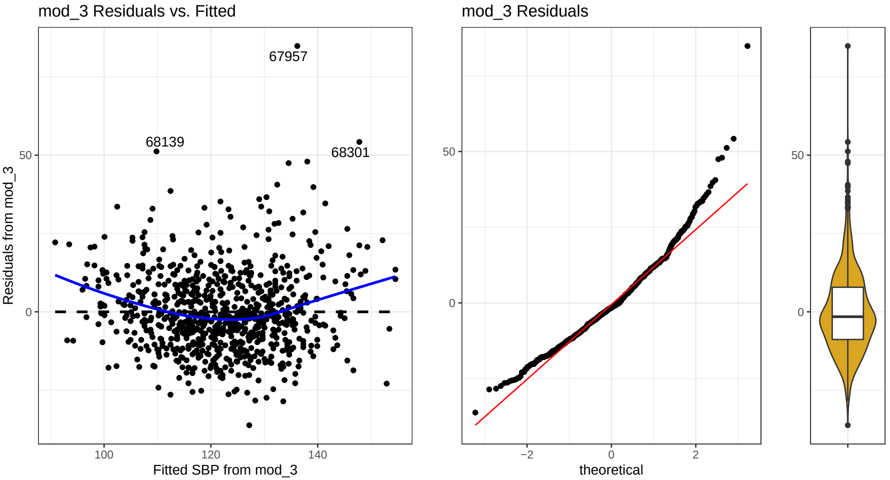
Model `mod_2`: `dbp` and `age`

r.squared	adj.r.squared	sigma	AIC	BIC
0.414	0.413	13.1	6391.8	6410.6

and for model `mod_3`: `dbp` and `age` and `smoke100`

r.squared	adj.r.squared	sigma	AIC	BIC
0.419	0.417	13.1	6387.3	6410.7

Residual Plots for `mod_3`?



Now, we plan to include the race1 data

Generally, what is measured as race/ethnicity here is more about racism and its impact on health disparities than it is about biological distinctions.

```
nh4 %>% tabyl(race1)
```

```
    race1   n percent
    Black 122 0.15250
 Hispanic  63 0.07875
  Mexican  77 0.09625
    White 457 0.57125
    Other  81 0.10125
```

Today, we'll collapse the data to create two factors here, one comparing White to Non-White, and another using three categories (White/Black/all others.)

Creating the Binary Variable race_white

```
nh4 <- nh4 %>%
  mutate(race_white = case_when(race1 == "White" ~ 1,
                                TRUE ~ 0))

nh4 %>% tabyl(race_white, race1) # sanity check
```

```
 race_white Black Hispanic Mexican White Other
          0   122       63      77     0    81
          1     0        0       0   457     0
```

race_white is a 1/0 numeric variable in R, instead of a factor, but that's fine for use as a predictor in our modeling.

Creating the 3-category Variable race_3cat

We want to retain the two largest categories (White and Black) and then put everyone else into a third category. We can use fct_lump_n to help...

```
nh4 <- nh4 %>%
  mutate(race_3cat = fct_lump_n(race1, n = 2))

nh4 %>% tabyl(race_3cat, race1) # sanity check
```

```
 race_3cat Black Hispanic Mexican White Other
     Black   122        0       0     0     0
     White     0        0       0   457     0
     Other     0       63      77     0    81
```

Change the order in the `race_3cat` factor?

I'd like to change the order of the categories in `race_3cat`. There are several ways to do this, for instance, I can sort them by how commonly they occur.

```
nh4 <- nh4 %>%
  mutate(race_3cat = fct_infreq(race_3cat))
```

```
nh4 %>% tabyl(race_3cat)
```

```
 race_3cat   n percent
     White 457 0.57125
     Other 221 0.27625
     Black 122 0.15250
```

That puts White first, then Other, then Black.

What if I want to choose a different order?

I can set the order to anything I like, by hand, with `fct_relevel`:

```
nh4 <- nh4 %>%
  mutate(race_3cat = fct_relevel(race_3cat,
                                 "White", "Black", "Other"))
```

```
nh4 %>% tabyl(race_3cat)
```

```
 race_3cat   n percent
     White 457 0.57125
     Black 122 0.15250
     Other 221 0.27625
```

I'll go with that order for today.

Working with Factors using forcats

The main fct_ functions I use are:

- fct_lump is used to lump together factor levels into "other"
    - fct_lump_min lumps levels that appear less than min times
    - fct_lump_n lumps all levels except the n most frequent
- fct_recode lets you change the factor levels by hand
- fct_relevel lets you rearrange existing factor levels by hand
- fct_reorder lets you sort the levels based on another variable

but there are many others. Read more about forcats tools at the forcats website at https://forcats.tidyverse.org/ which will also link you to the Factors chapter in R for Data Science.

Model mod_4: add race_white as a predictor

```
mod_4 <- lm(sbp ~ dbp + age + smoke100 + race_white, data = nh
mod_4
```

```
Call:
lm(formula = sbp ~ dbp + age + smoke100 + race_white, data = n

Coefficients:
(Intercept)          dbp          age  smoke100Yes
    50.0611       0.7481       0.3842       2.6378
 race_white
    -2.4768
```

Interpreting the binary predictor (race_white) and its slope

- race_white is either 1 or 0 for all subjects . . .
    - if subject's race1 was “White”, then race_white = 1, and
    - if subject's race1 was anything else, race_white = 0

Prediction for subject 65867?

subject	sbp	dbp	age	smoke100	race1	race_white
65867	115	78	60	No	White	1

From Model 4, our predicted `sbp` for subject 65867 will be:

50.061 + 0.748 dbp + 0.384 age + 2.638 (if smoke100 = Yes) - 2.477 (if race = White)

So for subject 65867, we'd predict:

50.061 + 0.748 (78) + 0.384 (60) + 2.638 (0) - 2.477 (1) = 128.97 mm Hg

augment for mod_4

```r
nh4_aug4 <- augment(mod_4, data = nh4)
```

subject	sbp	dbp	age	smoke100	race_white	.fitted	.resid
65867	115	78	60	No	1	128.9862	-13.986234
70046	125	83	55	No	1	130.8060	-5.806014
64302	98	59	45	No	1	109.0098	-11.009802
69386	141	68	52	Yes	0	123.5464	17.453567

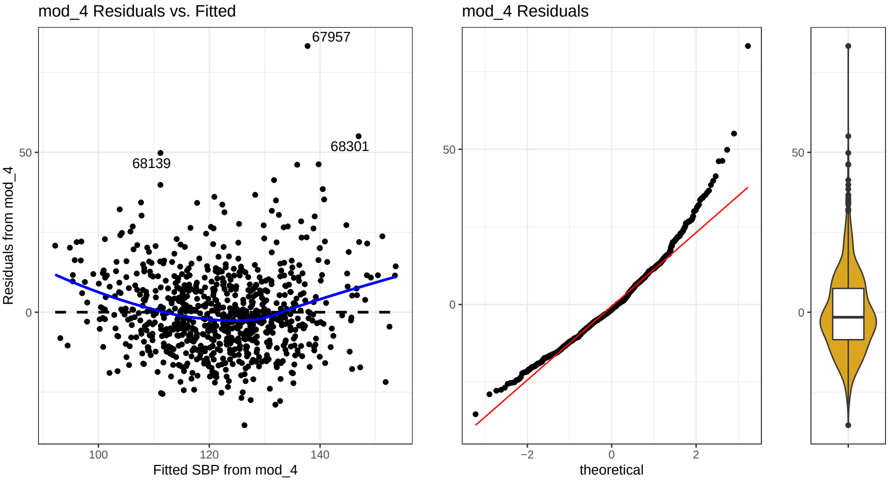
Model `mod_4` results from `tidy` and `glance`

Coefficients for `mod_4` with 90% confidence intervals:

term	estimate	std.error	conf.low	conf.high
(Intercept)	50.06	3.18	44.83	55.29
dbp	0.75	0.04	0.68	0.82
age	0.38	0.03	0.34	0.43
smoke100Yes	2.64	0.93	1.10	4.18
race_white	-2.48	0.94	-4.03	-0.92

r.squared	adj.r.squared	sigma	AIC	BIC
0.424	0.421	13	6382.4	6410.5

Residual Plots for `mod_4`?



mod_5: Using three race/ethnicity categories

```
mod_5 <- lm(sbp ~ dbp + age + smoke100 + race_3cat,
            data = nh4)
mod_5
```

```

Call:
lm(formula = sbp ~ dbp + age + smoke100 + race_3cat, data = nh

Coefficients:
  (Intercept)               dbp               age  
      47.8831            0.7449            0.3835  
  smoke100Yes    race_3catBlack    race_3catOther  
       2.5655            4.7147            1.2232  
```

OK. What's happened here? - What are our three categories for race_3cat? - Why do I only see two of them in the model?

Prediction for subject 65867?

subject	sbp	dbp	age	smoke100	race1	race_3cat
65867	115	78	60	No	White	White

- The **referent** category here is White, because that's the one left out of the set of indicators in the model. (We have coefficients for the other two `race_3cat` categories.)

From Model 5, our predicted `sbp` for subject 65867 will be:

47.883 + 0.745 dbp + 0.384 age + 2.566 (if smoke100 = Yes) + 4.715 (if `race_3cat` = Black) + 1.223 (if `race_3cat` = Other)

So for subject 65867, we'd predict:

47.883 + 0.745 (78) + 0.384 (60) + 2.566 (0) + 4.715 (0) + 1.223 (0) = 129.03 mm Hg

augment for mod_5

```r
nh4_aug5 <- augment(mod_5, data = nh4)
```

subject	sbp	dbp	age	smoke100	race_3cat	.fitted	.resid
65867	115	78	60	No	White	128.9984	-13.998435
70046	125	83	55	No	White	130.8056	-5.805603
64302	98	59	45	No	White	109.0921	-11.092071
69386	141	68	52	Yes	Other	122.2697	18.730255

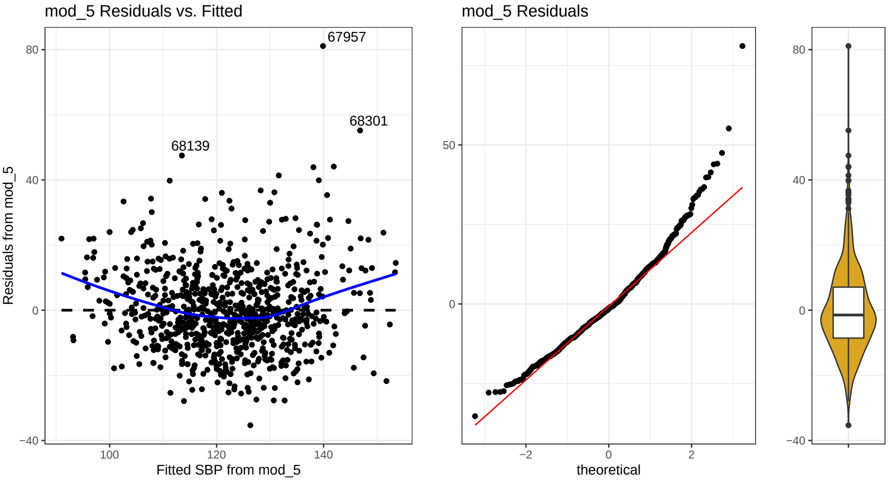
Model mod_5 results from tidy and glance

Coefficients for mod_5 with 90% confidence intervals:

term	estimate	std.error	conf.low	conf.high
(Intercept)	47.88	3.20	42.61	53.15
dbp	0.74	0.04	0.68	0.81
age	0.38	0.03	0.33	0.43
smoke100Yes	2.57	0.93	1.03	4.10
race_3catBlack	4.71	1.33	2.53	6.90
race_3catOther	1.22	1.08	-0.55	3.00

r.squared	adj.r.squared	sigma	AIC	BIC
0.428	0.424	13	6378.7	6411.5

Residual Plots for mod_5?



Glancing at our Five Models

model	preds	r.squared	adj.r.squared	sigma	AIC	BIC
1	dbp	0.291	0.290	14.40	6542.4	6556.4
2	1+age	0.414	0.413	13.10	6391.8	6410.6
3	2+smoke100	0.419	0.417	13.06	6387.3	6410.7
4	3+race_white	0.424	0.421	13.01	6382.4	6410.5
5	3+race_3cat	0.428	0.424	12.97	6378.7	6411.5

Does there appear to be a clear winner here?

Which one does best in our holdout sample?

We started with 989 subjects, and sampled 800 of them. How well do these models do when they are asked to predict the other 189 observations?

```r
heldout <- anti_join(nh3_new, nh4, by = "subject") %>%
  select(subject, sbp, dbp, age, smoke100, race1) %>%
  mutate(race_white = case_when(race1 == "White" ~ 1,
                                     TRUE ~ 0)) %>%
  mutate(race_3cat = fct_lump_n(race1, n = 2)) %>%
  mutate(race_3cat =
           fct_relevel(race_3cat,
                       "White", "Black", "Other"))

dim(heldout)
```

```
[1] 189   8
```

Sanity Checks

```
heldout %>% tabyl(race_white, race1)
```

```
 race_white Black Hispanic Mexican White Other
          0    38       18      17     0    15
          1     0        0       0   101     0
```

```
heldout %>% tabyl(race_3cat, race1)
```

```
 race_3cat Black Hispanic Mexican White Other
     White     0        0       0   101     0
     Black    38        0       0     0     0
     Other     0       18      17     0    15
```

How does our mod_1 do out of sample?

```
heldout_mod1 <- augment(mod_1, newdata = heldout)

heldout_mod1 %>% select(subject, sbp, .fitted, .resid) %>%
  head() %>% kable()
```

subject	sbp	.fitted	.resid
65956	98	116.0260	-18.026024
71072	101	121.6898	-20.689797
64134	128	132.2082	-4.208233
66879	123	130.5900	-7.590012
66141	122	119.2625	2.737535
71279	147	150.0087	-3.008663

Out-of-sample crude estimate of R-square

In our new sample, the square of the (Pearson) correlation between the observed `sbp` and the model `mod_1` predicted `sbp` or the `.fitted` values, will be our estimated R-square.

```
heldout_mod1 %$% cor(sbp, .fitted)
```

```
[1] 0.4841063
```

```
heldout_mod1 %$% cor(sbp, .fitted)^2
```

```
[1] 0.2343589
```

OK. So our estimate of the out-of-sample R-square = 0.234 based on this sample.

- How does this compare to our in-sample R-square for `mod_1`, which was 0.291?
- Or maybe our adjusted R-square for `mod_1` which was 0.29?

Create predictions for the other four models

```
heldout_mod2 <- augment(mod_2, newdata = heldout)
heldout_mod3 <- augment(mod_3, newdata = heldout)
heldout_mod4 <- augment(mod_4, newdata = heldout)
heldout_mod5 <- augment(mod_5, newdata = heldout)
```

R^2 Comparisons for Models 1-5

Model	Predictors	In-sample R^2	In-sample R^2_{adj}	Holdout R^2
mod_1	`dbp`	0.291	0.29	0.234
mod_2	1 + `age`	0.414	0.413	0.329
mod_3	2 + `smoke100`	0.419	0.417	0.33
mod_4	3 + `race_white`	0.424	0.421	0.344
mod_5	3 + `race_3cat`	0.428	0.424	0.359

What if we look at the σ values - the residual standard deviations?

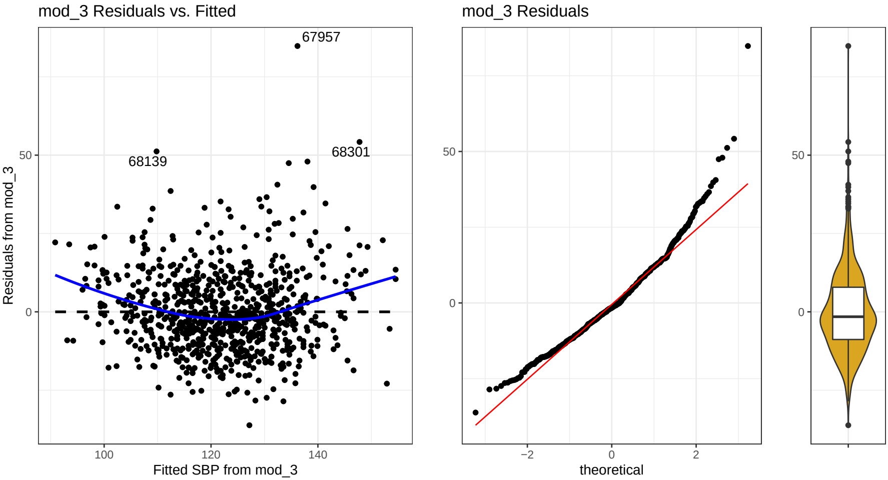
σ Comparisons for Models 1-5

Model	Predictors	In-sample σ	Holdout σ
mod_1	`dbp`	14.4	15.01
mod_2	1 + `age`	13.1	14.06
mod_3	2 + `smoke100`	13.06	14.04
mod_4	3 + `race_white`	13.01	13.9
mod_5	3 + `race_3cat`	12.97	13.73

Looks like our model summaries are just too optimistic?

- What might have tipped us off?

Residual Plots (`mod_3`)



Why Transform the Outcome?

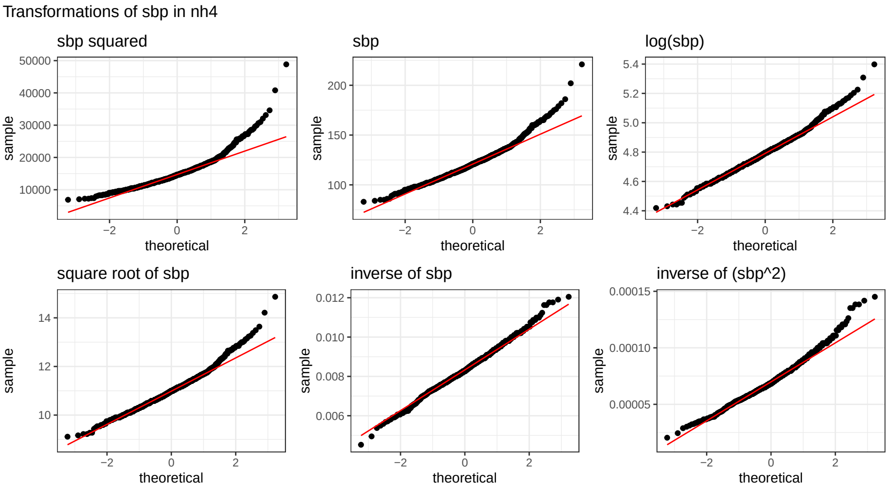
We want to try to identify a good transformation for the conditional distribution of the outcome, given the predictors, in an attempt to make the linear regression assumptions of linearity, Normality and constant variance more appropriate.

Tukey's Ladder of Power Transformations

Transformation	y^2	y	$\sqrt{y}$	log(y)	$1/\sqrt{y}$	$1/y$	$1/y^2$
λ	2	1	0.5	0	-0.5	-1	-2

- The most essential transformations (easy to understand) are the square, square root, logarithm and inverse.

sbp distribution with various transformations



Build model `mod_3` with log(`sbp`)

- Let's try a log transformation. We'll use the natural logarithm (`log`) as opposed to a base 10 logarithm (in R, `log10`) but that choice won't affect the residual plots.

```
mod_3_log <- lm(log(sbp) ~ dbp + age + smoke100, data = nh4)

mod_3_log
```

```

Call:
lm(formula = log(sbp) ~ dbp + age + smoke100, data = nh4)

Coefficients:
(Intercept)          dbp          age  smoke100Yes  
   4.211704     0.006000     0.002989     0.019000  
```

Prediction for subject 65867?

subject	sbp	dbp	age	smoke100
65867	115	78	60	No

```
Call:
lm(formula = log(sbp) ~ dbp + age + smoke100, data = nh4)

Coefficients:
(Intercept)          dbp          age  smoke100Yes
   4.211704     0.006000     0.002989     0.019000
```

- Fitted **log(sbp)** = `4.21 + 0.006(78) + 0.003(60) + 0.019(0)` = 4.859
- Observed **log(sbp)** = 4.745, so residual on the log scale is -0.104
- Predicted **sbp** = exp(4.859) = 128.9, while Observed **sbp** was 115.

Tidied coefficients of our log model

```r
tidy(mod_3_log, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 3)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	4.212	0.025	4.171	4.253
dbp	0.006	0.000	0.005	0.007
age	0.003	0.000	0.003	0.003
smoke100Yes	0.019	0.007	0.007	0.031

- Are these results comparable to our previous models?

Fit summaries for our log model

```
glance(mod_3_log) %>%
  select(r.squared, adj.r.squared, sigma, AIC, BIC, nobs) %>%
  kable(digits = c(3,3,2,1,1,0))
```

r.squared	adj.r.squared	sigma	AIC	BIC	nobs
0.428	0.425	0.1	-1367.6	-1344.1	800

- Are these results comparable to our previous models?

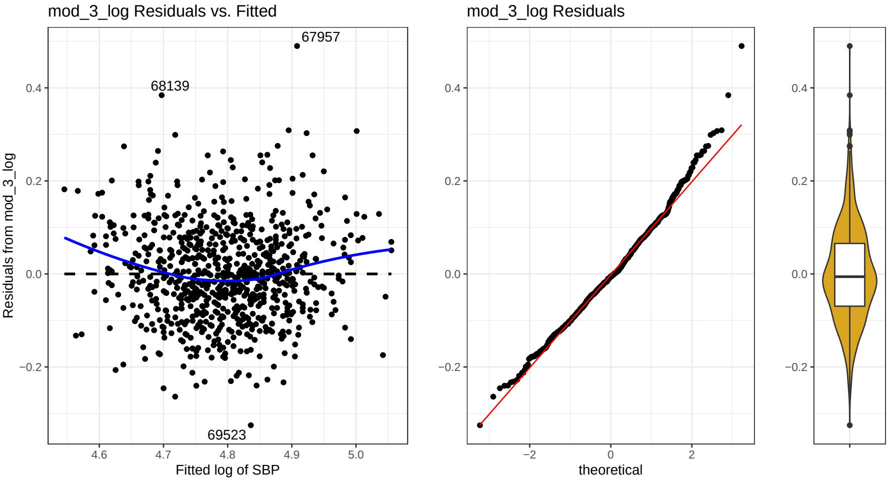
Using augment with our logged model

```
nh4_aug3_log <- augment(mod_3_log, data = nh4)

nh4_aug3_log %>%
  mutate(log_sbp = log(sbp)) %>%
  select(subject, sbp, log_sbp, .fitted, .resid,
         dbp, age, smoke100) %>%
  head(4) %>% kable(digits = c(0,0,3,3,3,0,0,0))
```

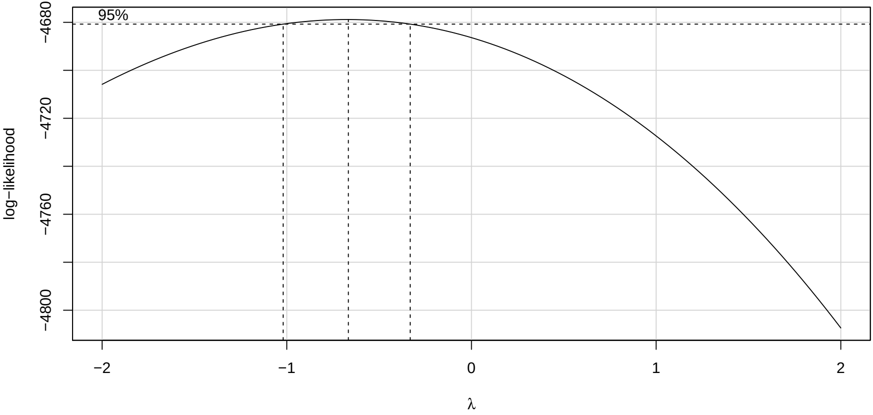
subject	sbp	log_sbp	.fitted	.resid	dbp	age	smoke100
65867	115	4.745	4.859	-0.114	78	60	No
70046	125	4.828	4.874	-0.046	83	55	No
64302	98	4.585	4.700	-0.115	59	45	No
69386	141	4.949	4.794	0.155	68	52	Yes

Residual Plots for `mod_3_log`



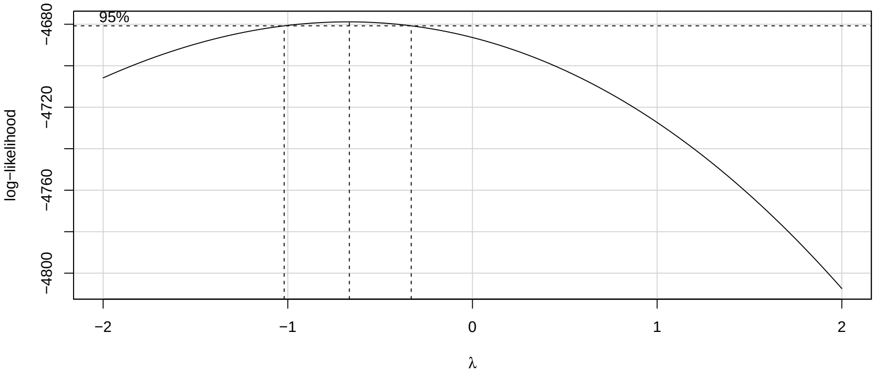
Using Box-Cox to help select a transformation?

```
boxCox(mod_3) # requires library(car)
```



Remember the ladder!

Power λ	-2	-1	-0.5	0	0.5	1
transformation	$1/y^2$	$1/y$	$1/\sqrt{y}$	$log(y)$	$\sqrt{y}$	y



Try the transformation suggested by Box-Cox?

Looks like $1/\sqrt{sbp}$ (where $\lambda = -0.5$) is the suggested transformation, although $1/sbp$ (where $\lambda = -1$) is also within the reach of the provided 95% interval for λ.

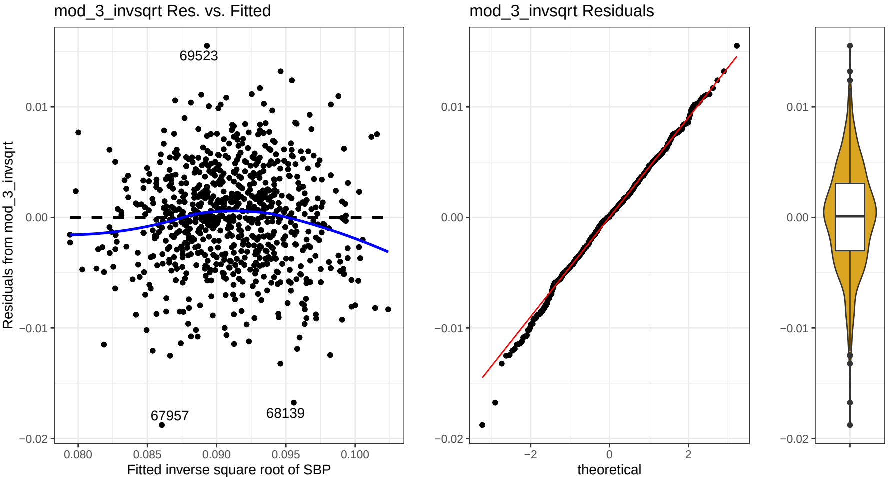
- As compared to the inverse square root, the inverse has the enormous advantage in many studies of being far easier to interpret.

Let's build each model and then we'll look at their residuals to see how well regression assumptions hold.

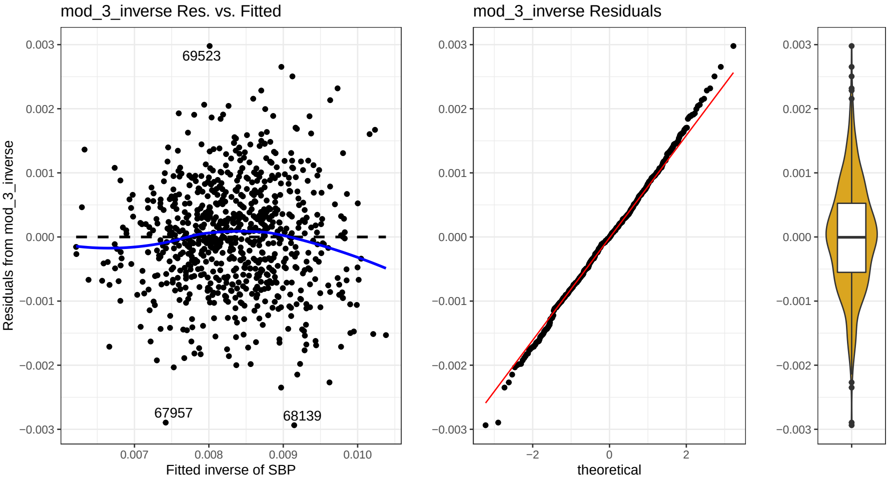
```r
mod_3_invsqrt <-
  lm((1/sqrt(sbp)) ~ dbp + age + smoke100, data = nh4)

mod_3_inverse <-
  lm((1/sbp) ~ dbp + age + smoke100, data = nh4)
```

Residual Plots for `mod_3_invsqrt`



Residual Plots for `mod_3_inverse`



Early Notes on Transformations

The Box-Cox plot (and indeed the ladder of power transformations) is designed to help identify transformations when:

1. all of the values in the data are strictly positive, so that the log and square root, for instance, are defined
    - If your outcome includes zeros, you could just add 1 to each value before transforming at the risk of making interpretation harder.
2. the problems with assuming a linear relationship and/or Normality of residuals are indicated by more than just an outlier, or a few outliers.
    - In that case, I would focus on the *influence* of those outliers (what happens to the model when you remove them?)

In any case, back-transforming predictions will be necessary at some stage if you apply a re-expression, which isn't too bad, but it can be challenging to write down the regression equation in terms of the original outcome.

What have we discussed?

- The central role of linear regression in understanding associations between quantitative variables.
- The interpretation of a regression model as a prediction model.
- The meaning of key regression summaries, including residuals.
- Using tidy and glance from the broom package to help with summaries.
- Measuring association through correlation coefficients.
- How we might think about "adjusting" for the effect of a categorical predictor on a relationship between two quantitative ones.
- How a transformation might help us "linearize" the relationship shown in a scatterplot.