431 Class 10

thomaselove.github.io/431

2020-09-24

Today's R Packages

```r
library(NHANES)
library(janitor)
library(knitr)
library(broom)
library(magrittr)
library(patchwork)
library(ggrepel)
library(tidyverse)

theme_set(theme_bw())
```

nh3_new data (n = 989, 17 variables)

```
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)

```
set.seed(431)

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

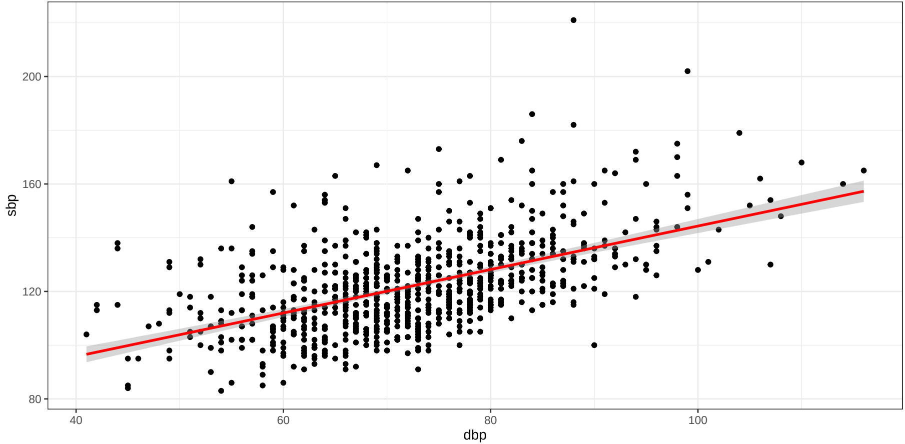
- Outcome (quantitative): `sbp`
- Quantitative predictors: `dbp`, `age`
- Binary predictor: `smoke100` (Yes/No)
- 5-category predictor: `race1` (White, Black, Hispanic, Mexican, Other)
- Identification code: `subject`

```
dim(nh4)
```

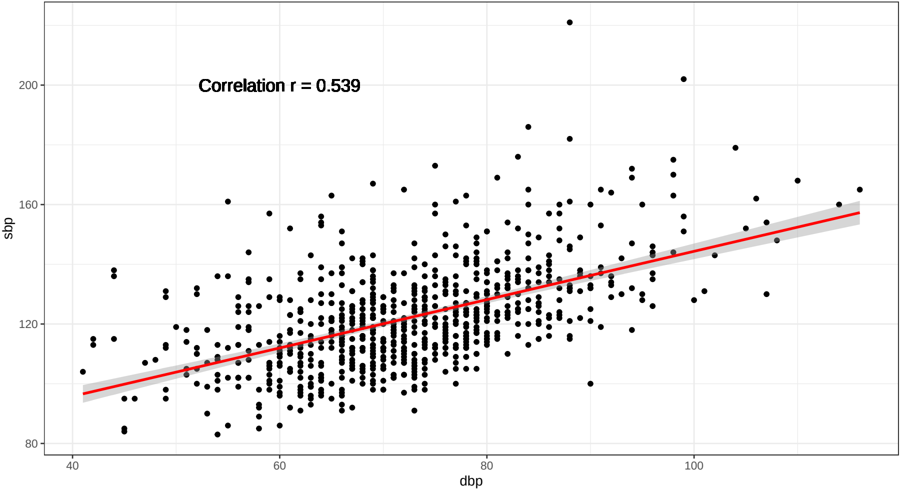
```
[1] 800   6
```

Association of sbp and dbp

```r
ggplot(nh4, aes(x = dbp, y = sbp)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red", formula = y ~ x)
```



Adding text to the plot (Pearson correlation)



Code for the last slide

```r
ggplot(nh4, aes(x = dbp, y = sbp)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red", formula = y ~ x) +
  geom_text(aes(x = 60, y = 200), size = 5,
      label = paste0("Correlation r = ",
                      round_half_up(
                        cor(nh4$sbp, nh4$dbp),3)))
```

Model mod_1 description

We'll use a linear model to predict sbp using dbp:

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

```

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

Coefficients:
(Intercept)          dbp
    63.4338       0.8091
```

Prediction for subject 65867, with sbp $= 115$ and dbp $= 78$?

- predicted sbp $= 63.4338 + 0.8091(78) = 126.54$
- actual sbp for subject 65867 is 115, so residual $=$ -11.54

Model mod_1 coefficients and fit measures

```
tidy(mod_1, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 2)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	63.43	3.28	58.03	68.84
dbp	0.81	0.04	0.74	0.88

```
glance(mod_1) %>%
  select(r.squared, adj.r.squared, sigma, AIC, BIC) %>%
  kable(digits = c(3, 3, 1, 1, 1))
```

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

`augment` yields `.fitted` values & `.resid` (residuals)

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

We include the `data` in the `augment` statement so that all variables from `nh4` are retained here (including those not included in the `mod_1` model.)

```r
names(nh4_aug1)
```

```
 [1] "subject"    "sbp"        "dbp"
 [4] "age"        "smoke100"   "race1"
 [7] ".fitted"    ".resid"     ".std.resid"
[10] ".hat"       ".sigma"     ".cooksd"
```

Here, note that `.resid` = `sbp` - `.fitted`

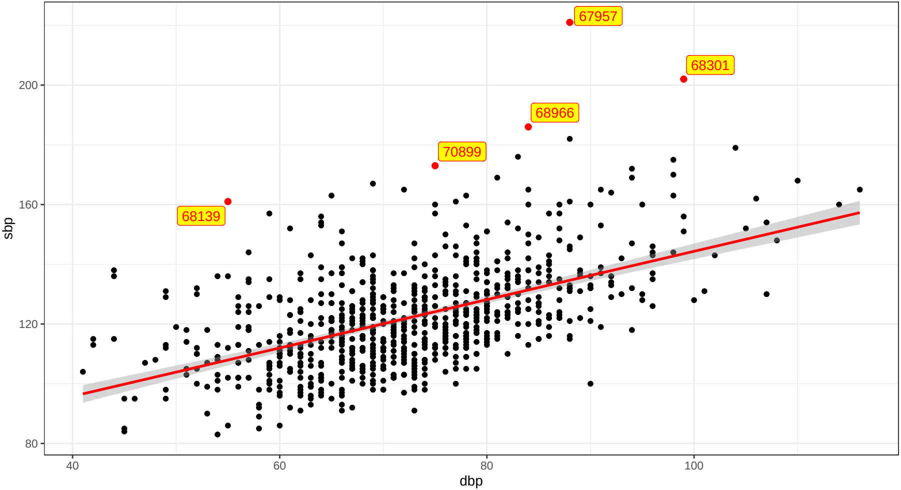
Which five subjects are fit worst by model `mod_1`?

We'll identify those with the five largest residuals (in absolute value).

```
nh4_aug1 %>% select(subject, sbp, dbp, .resid) %>%
  slice_max(abs(.resid), n = 5)
```

```
# A tibble: 5 x 4
  subject   sbp   dbp .resid
  <chr>   <int> <int>  <dbl>
1 67957     221    88   86.4
2 68301     202    99   58.5
3 68966     186    84   54.6
4 68139     161    55   53.1
5 70899     173    75   48.9
```

Label the 5 subjects with the largest |residuals|

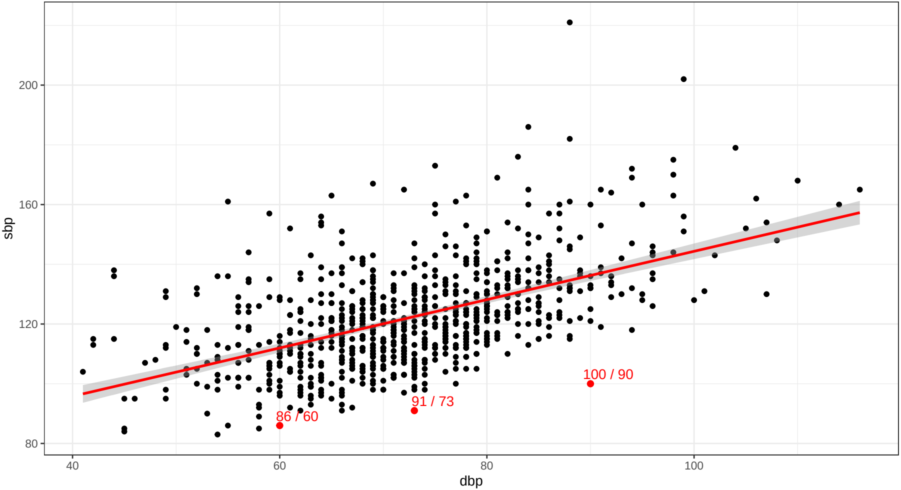


Code for the plot on the previous slide

```r
## requires library(ggrepel)

ggplot(nh4_aug1, aes(x = dbp, y = sbp)) +
  geom_point() +
  geom_point(data = nh4_aug1 %>%
               slice_max(abs(.resid), n = 5),
             col = "red", size = 2) +
  geom_smooth(method = "lm", col = "red", formula = y ~ x) +
  geom_label_repel(data = nh4_aug1 %>%
                     slice_max(abs(.resid), n = 5),
                   aes(label = subject),
                   fill = "yellow", col = "red")
```

SBP/DBP for the 3 most negative residuals



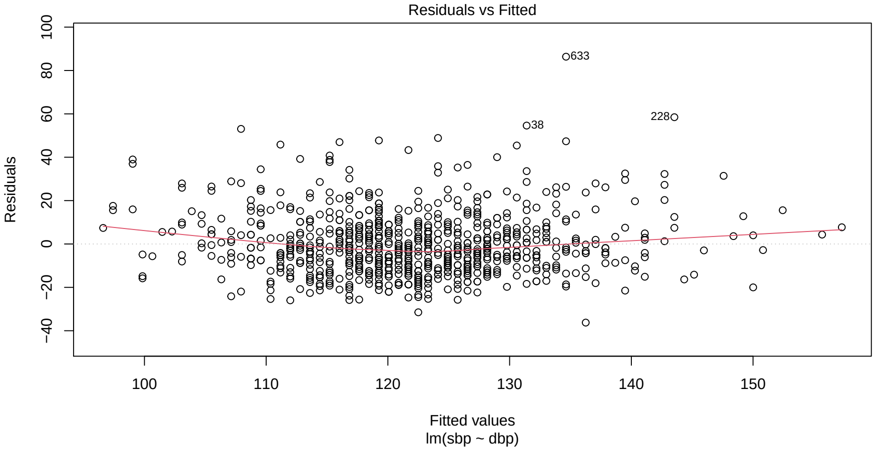
Code for the previous slide

```r
## requires library(ggrepel)

ggplot(nh4_aug1, aes(x = dbp, y = sbp)) +
  geom_point() +
  geom_point(data = nh4_aug1 %>%
               slice_min(.resid, n = 3),
             col = "red", size = 2) +
  geom_smooth(method = "lm", col = "red", formula = y ~ x) +
  geom_text_repel(data = nh4_aug1 %>%
                    slice_min(.resid, n = 3),
                   aes(label = paste0(sbp, " / ", dbp)),
                  col = "red")
```

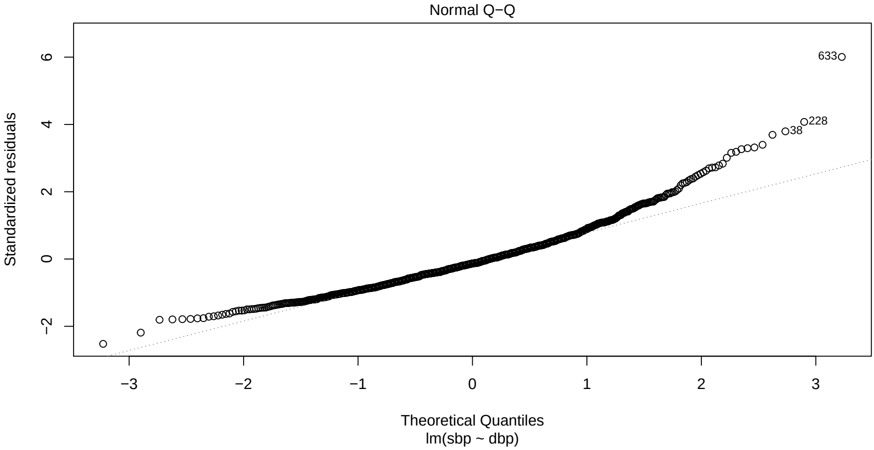
Residuals vs. Fitted Values for `mod_1`

```r
plot(mod_1, which = 1)
```

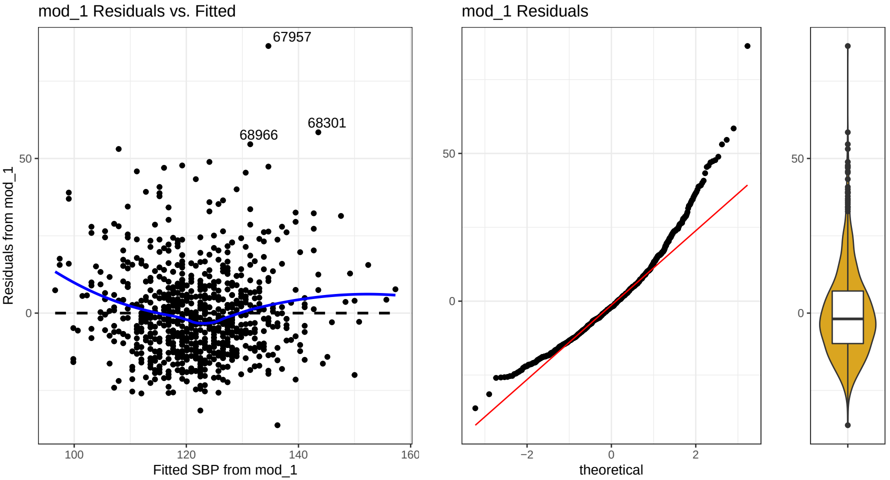


Normal Q-Q plot of Residuals for `mod_1`

```r
plot(mod_1, which = 2)
```



Using ggplot2 for mod_1 residual plots



Code for ggplot2 residual plots (1/2)

```r
p1 <- ggplot(nh4_aug1, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = F,
              lty = "dashed", col = "black") +
  geom_smooth(method = "loess", formula = y ~ x, se = F,
              col = "blue") +
  geom_text_repel(data = nh4_aug1 %>%
                    slice_max(abs(.resid), n = 3),
                  aes(label = subject)) +
  labs(title = "mod_1 Residuals vs. Fitted",
       x = "Fitted SBP from mod_1",
       y = "Residuals from mod_1")
```

Code for `ggplot2` residual plots (2/2)

```r
p2 <- ggplot(nh4_aug1, aes(sample = .resid)) +
  geom_qq() + geom_qq_line(col = "red") +
  labs(title = "mod_1 Residuals",
       y = "")

p3 <- ggplot(nh4_aug1, aes(y = .resid, x = "")) +
  geom_violin(fill = "goldenrod") +
  geom_boxplot(width = 0.5) +
  labs(y = "", x = "")

p1 + p2 + p3 + plot_layout(widths = c(5, 4, 1))
```

Model mod_2: add age as a predictor

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

```

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

Coefficients:
(Intercept)          dbp          age
    49.5882       0.7528       0.3826
```

Prediction for subject 65867?

subject	sbp	dbp	age
65867	115	78	60

augment for mod_2

```r
nh4_aug2 <- augment(mod_2, data = nh4)

nh4_aug2 %>% head(4) %>%
  select(subject, sbp, dbp, age, .fitted, .resid) %>%
  kable()
```

subject	sbp	dbp	age	.fitted	.resid
65867	115	78	60	131.2639	-16.263902
70046	125	83	55	133.1147	-8.114693
64302	98	59	45	111.2214	-13.221362
69386	141	68	52	120.6749	20.325081

Compare mod_1 to mod_2 with tidy?

```
tidy(mod_1, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 2)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	63.43	3.28	58.03	68.84
dbp	0.81	0.04	0.74	0.88

```
tidy(mod_2, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 2)
```

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

glance for mod_1 and mod_2

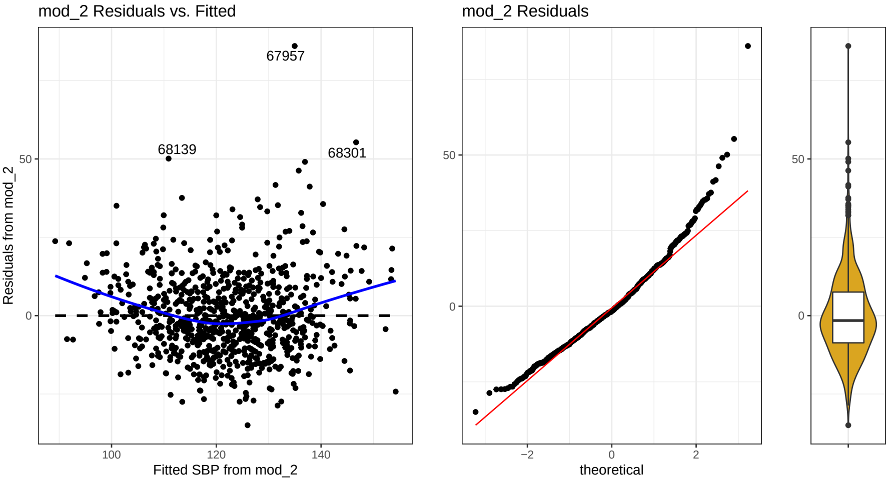
```r
glance(mod_1) %>%
  select(r.squared, adj.r.squared, sigma, AIC, BIC) %>%
  kable(digits = c(3, 3, 1, 1, 1))
```

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

```r
glance(mod_2) %>%
  select(r.squared, adj.r.squared, sigma, AIC, BIC) %>%
  kable(digits = c(3, 3, 1, 1, 1))
```

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

Residual Plots for `mod_2`?



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

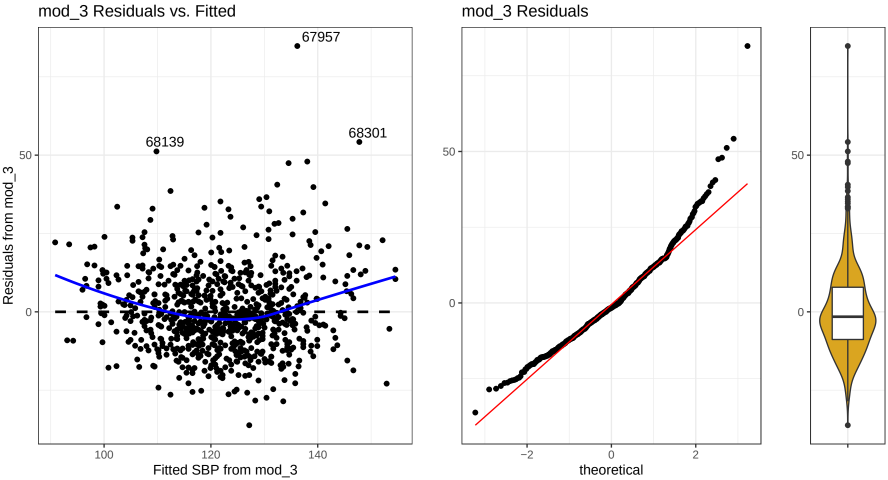
Model `mod_2`: `dbp` and `age`

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`:

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

```r
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

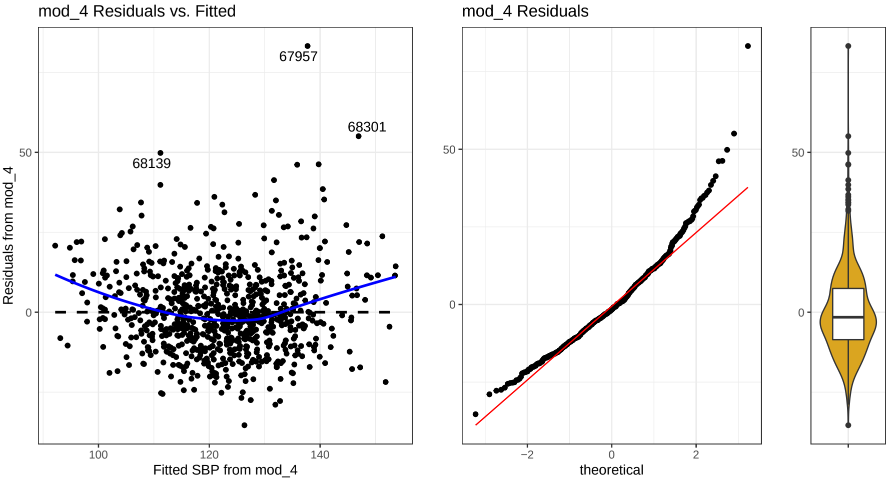
Model mod_4 results from tidy and glance

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

```r
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

```
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

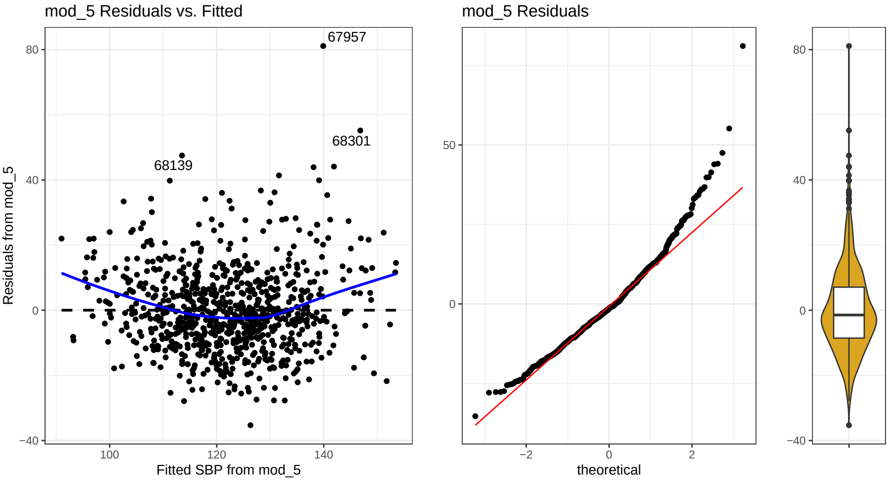
Model mod_5 results from tidy and glance

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?

```r
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?