431 Class 09

thomaselove.github.io/431

2020-09-22

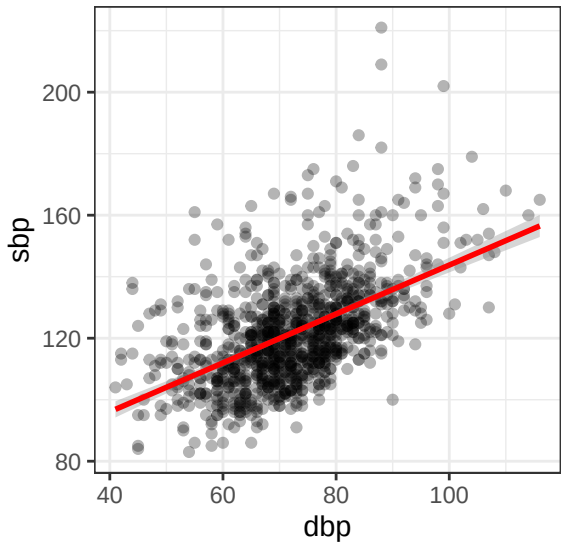
Today's R Packages

```
library(NHANES)
library(janitor)
library(knitr)
library(broom)
library(magrittr)
library(patchwork)
library(rstanarm)
library(tidyverse)

theme_set(theme_bw())
```

Our 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(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)
```

Correlation in our sbp-dbp scatterplot?




```
nh3_new %$% cor(sbp, dbp)
```

```
[1] 0.5299471
```

What does a correlation of +0.53 imply about a linear fit to the data?

What line is being fit?

Least Squares Regression Line (a linear model) to predict sbp using dbp

```
m1 <- lm(sbp ~ dbp, data = nh3_new)
m1
```

```
Call:
lm(formula = sbp ~ dbp, data = nh3_new)

Coefficients:
(Intercept)          dbp
     64.270        0.795
```

Model m1 is **sbp = 64.270 + 0.795 dbp**.

Linear Model `m1`: sbp = 64.27 + 0.795 dbp

64.27 is the intercept = predicted value of `sbp` when `dbp` = 0.

0.795 is the slope = predicted change in `sbp` per 1 unit change in `dbp`

- What are the units?
- What does the fact that this estimated slope is positive mean?
- What would the line look like if the slope was negative?
- What if the slope was zero?

Summarizing the Fit

The summary function when applied to a linear model (lm) produces a lot of output that is not organized in a way that we can plot/manipulate it well.

Here's the start of what it looks like. . . (complete snapshot on next slide)

```
summary(m1)
```

```
Call:
lm(formula = sbp ~ dbp, data = nh3_new)

Residuals:
    Min      1Q  Median      3Q     Max
-35.824  -9.792  -2.103   6.947  86.766

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  64.2698     2.9617   21.70   <2e-16 ***
```

summary(m1) in its entirety

```
> summary(m1)

Call:
lm(formula = sbp ~ dbp, data = nh3_new)

Residuals:
    Min      1Q  Median      3Q     Max
-35.824  -9.792  -2.103   6.947  86.766

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  64.2698     2.9617   21.70   <2e-16 ***
dbp           0.7950     0.0405   19.63   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.52 on 987 degrees of freedom
Multiple R-squared:  0.2808,    Adjusted R-squared:  0.2801
F-statistic: 385.4 on 1 and 987 DF,  p-value: < 2.2e-16
```

Why I like `tidy()` and other `broom` functions

https://github.com/allisonhorst/stats-illustrations

Does R like this linear model?

```
tidy(m1) %>% kable(digits = 2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	64.27	2.96	21.70	0
dbp	0.80	0.04	19.63	0

Yes. Wow. It **really** does. Look at those p values!

How much of the variation in sbp does m1 capture?

The glance function can help us (again from broom.)

```
glance(m1) %>% select(r.squared, p.value, sigma) %>% kable()
```

r.squared	p.value	sigma
0.2808439	0	14.51877

- r.squared = R^2, the proportion of variation in sbp accounted for by the model using dbp.
  - indicates improvement over predicting mean(sbp) for everyone
- p.value = refers to a global F test
  - indicates something about combination of r^2 and sample size
- sigma = residual standard error

glance provides 9 additional summaries for a linear model.

How is the r-squared (r^2)?

R-squared describes the proportion of the variation in `sbp` accounted for by the linear model `m1` using `dbp`.

- R^2 is about 28% (or 0.28) in this case. Is that good?
- Why is this called R-squared? What is the R?

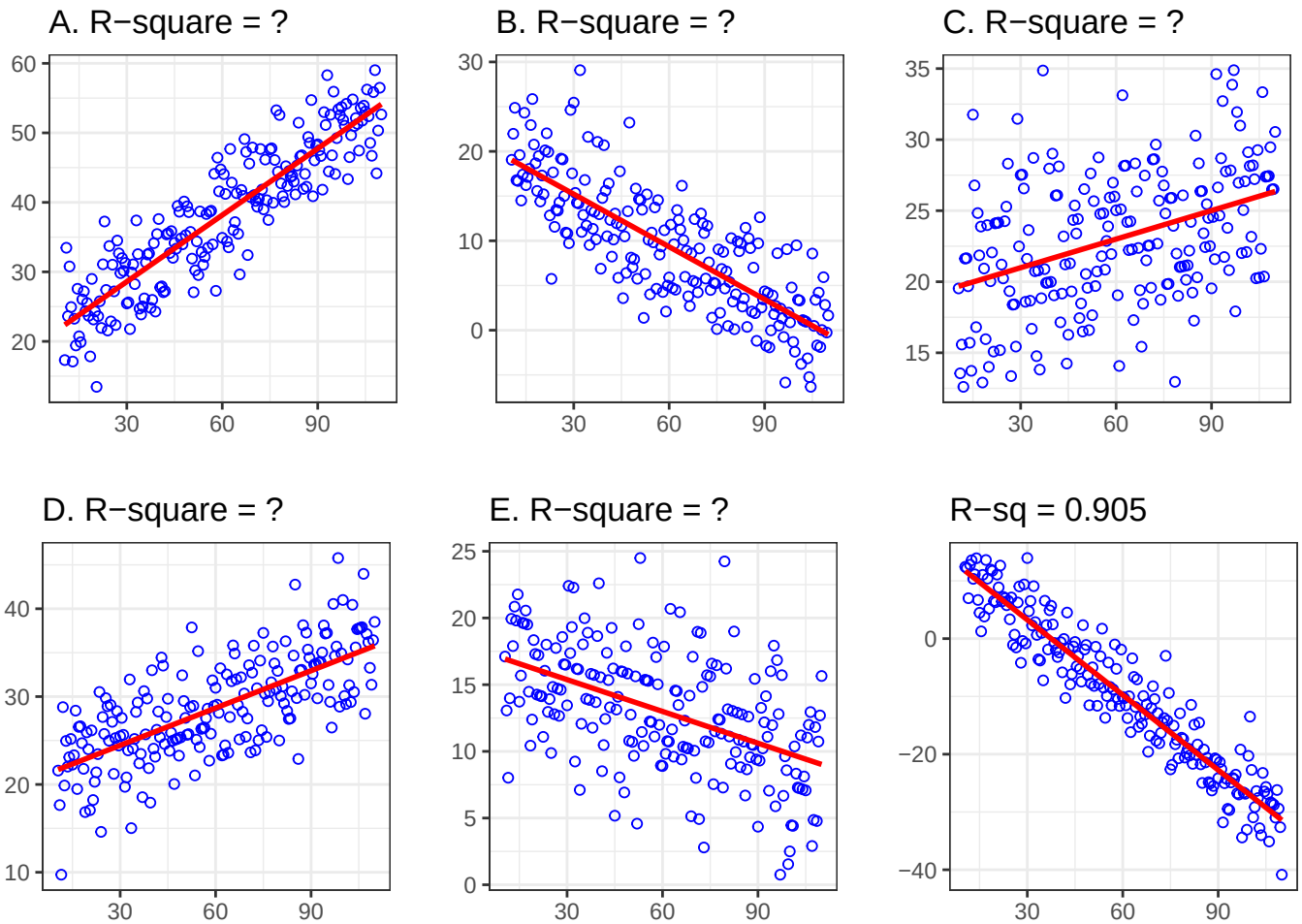
```
nh3_new %$% cor(sbp, dbp)
```

```
[1] 0.5299471
```

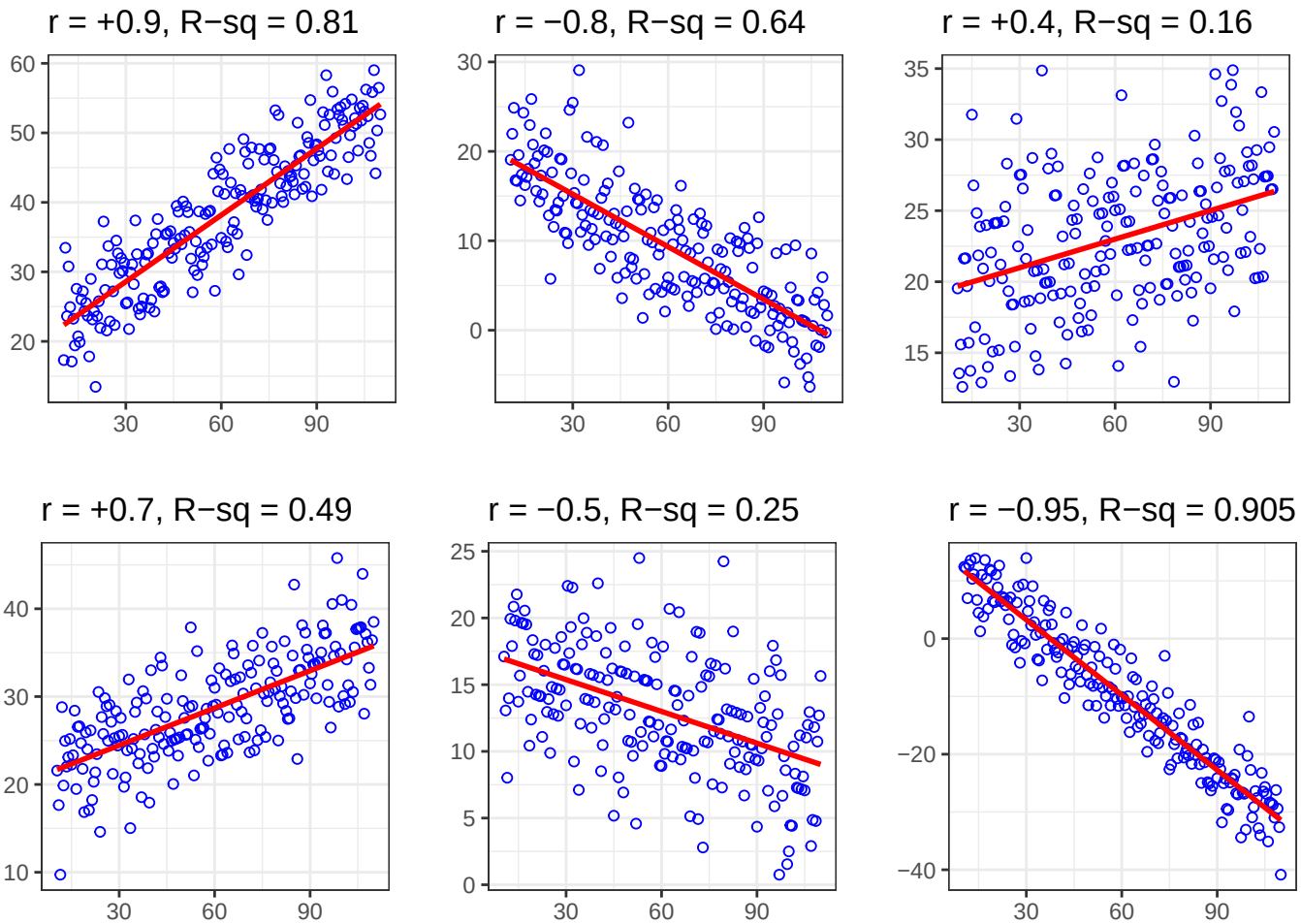
```
nh3_new %$% cor(sbp, dbp)^2
```

```
[1] 0.2808439
```

Can you guess the missing R-squares?



Gaining Insight into what R-square implies



Predict using m1: sbp = 64.27 + 0.795 dbp

Use augment (also from broom) to capture results.

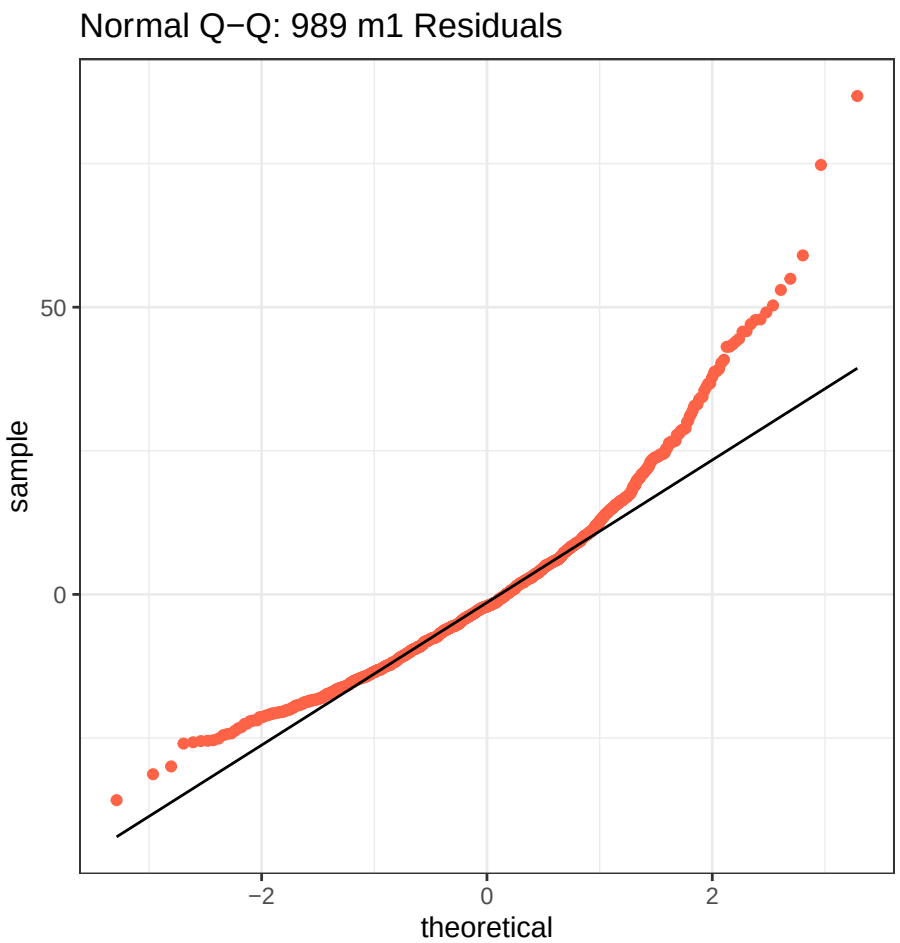
```
m1_insample <- augment(m1, data = nh3_new)

m1_insample %>% select(id, sbp, dbp, .fitted, .resid) %>%
  head(2) %>% kable(digits = 2)
```

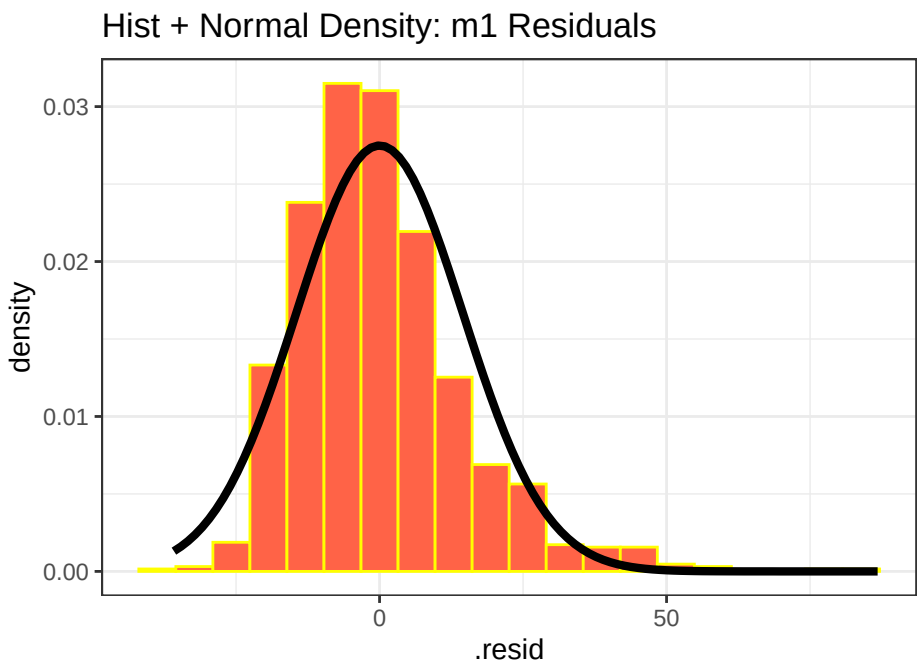
id	sbp	dbp	.fitted	.resid
69036	136	44	99.25	36.75
65956	98	65	115.95	-17.95

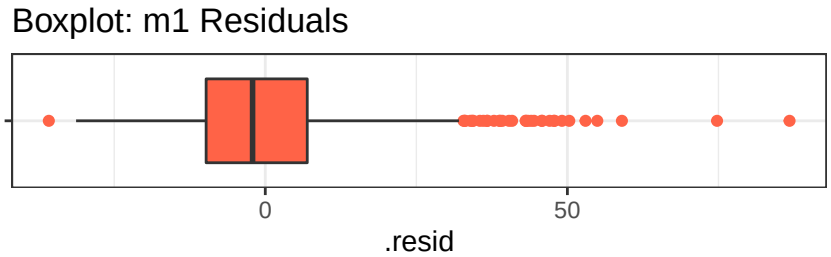
For subject 69036, as an example, we have:

- m1's fitted sbp = 64.27 + 0.795 (44) = 99.25 mm Hg
- **residual** = observed - fitted = 136 - 99.25 = 36.75 mm Hg

Plot residuals from `m1` in our sample (n = 989)

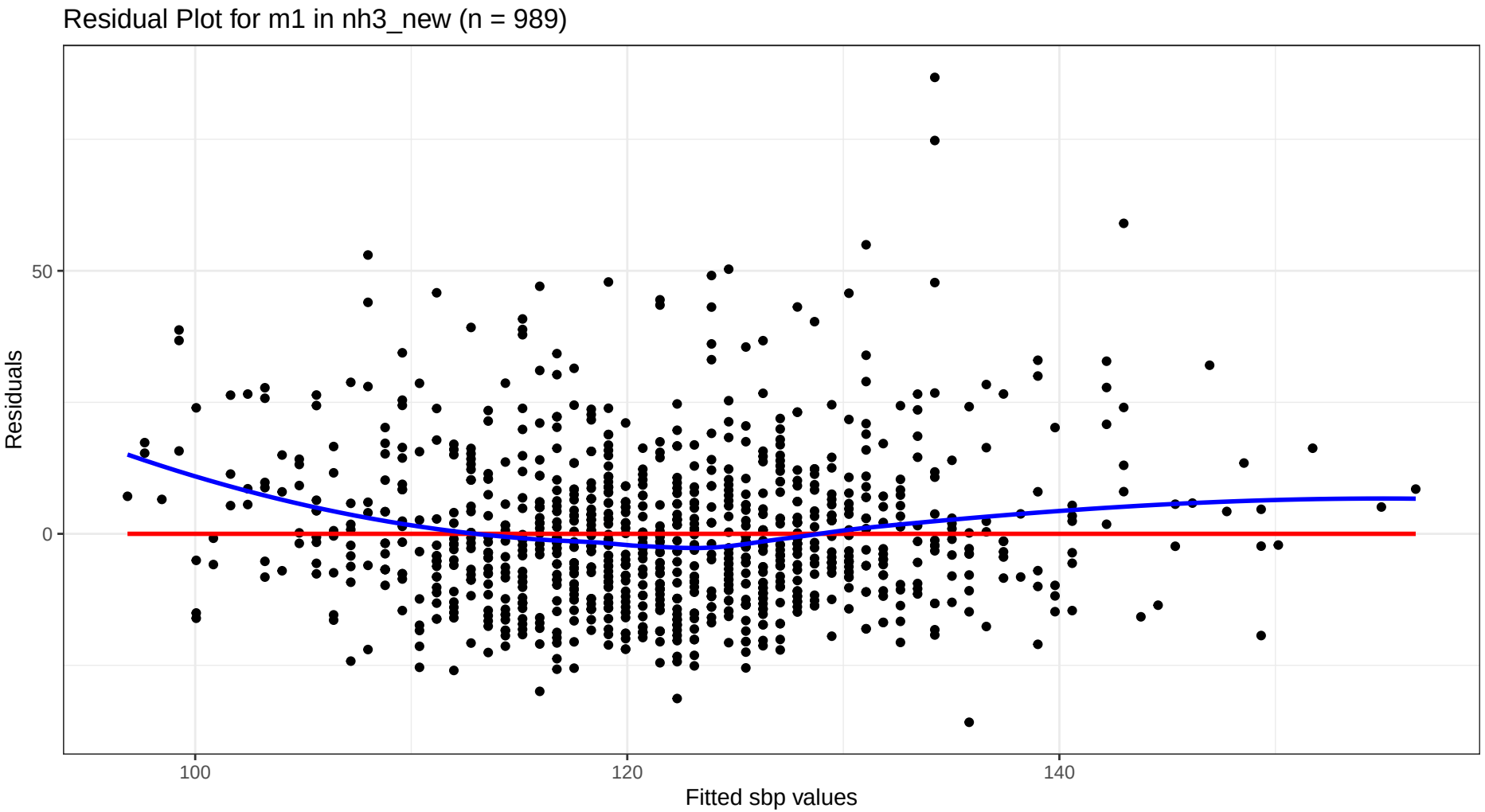






min	Q1	median	Q3	max	mean	sd	n	missing
-35.8	-9.8	-2.1	6.9	86.8	0	14.5	989	0

Plot Residuals vs. Predicted (Fitted) Values



Who else could we make predictions for with m1?

Consider NHANES subjects who we didn't choose for the nh3 sample?

```
nh_deduplicated <- NHANES %>%
    filter(SurveyYr == "2011_12") %>%
    select(ID, SurveyYr, Age, Height, Weight, BMI, Pulse,
           SleepHrsNight, BPSysAve, BPDiaAve, Gender,
           PhysActive, SleepTrouble, Smoke100,
           Race1, HealthGen, Depressed) %>%
    rename(SleepHours = SleepHrsNight, Sex = Gender,
           SBP = BPSysAve, DBP = BPDiaAve) %>%
    filter(Age > 20 & Age < 80) %>%
    drop_na() %>%
    distinct()
```

This nh_deduplicated group is who we sampled from to get nh3.

Identifying those not sampled, but still eligible.

We sampled 1000 observations from a group, and then dropped those with dbp below 40, leaving n = 989. How many people in total would be eligible?

```
nh3_new_eligible <- nh_deduplicated %>%
  clean_names() %>%
  filter(dbp > 39)

dim(nh3_new_eligible)
```

```
[1] 1709   17
```

```
dim(nh3_new)
```

```
[1] 989  17
```

Identify the rest: 1709-989 = 720 not sampled

```
nh3_therest <-
  anti_join(nh3_new_eligible, nh3_new, by = "id")

dim(nh3_therest)
```

```
[1] 720  17
```

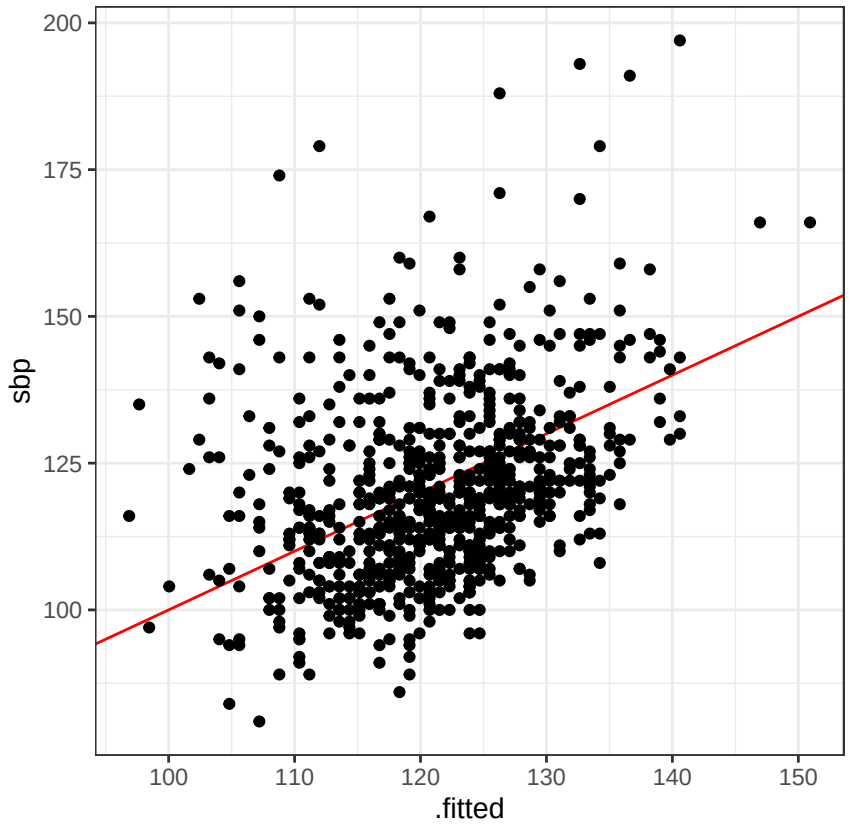
Use model m1 to predict SBP in nh3_therest?

```
new720_nh3 <- augment(m1, newdata = nh3_therest)

new720_nh3 %>% select(id, sbp, dbp, .fitted, .resid) %>%
  head() %>% kable(digits = 2)
```

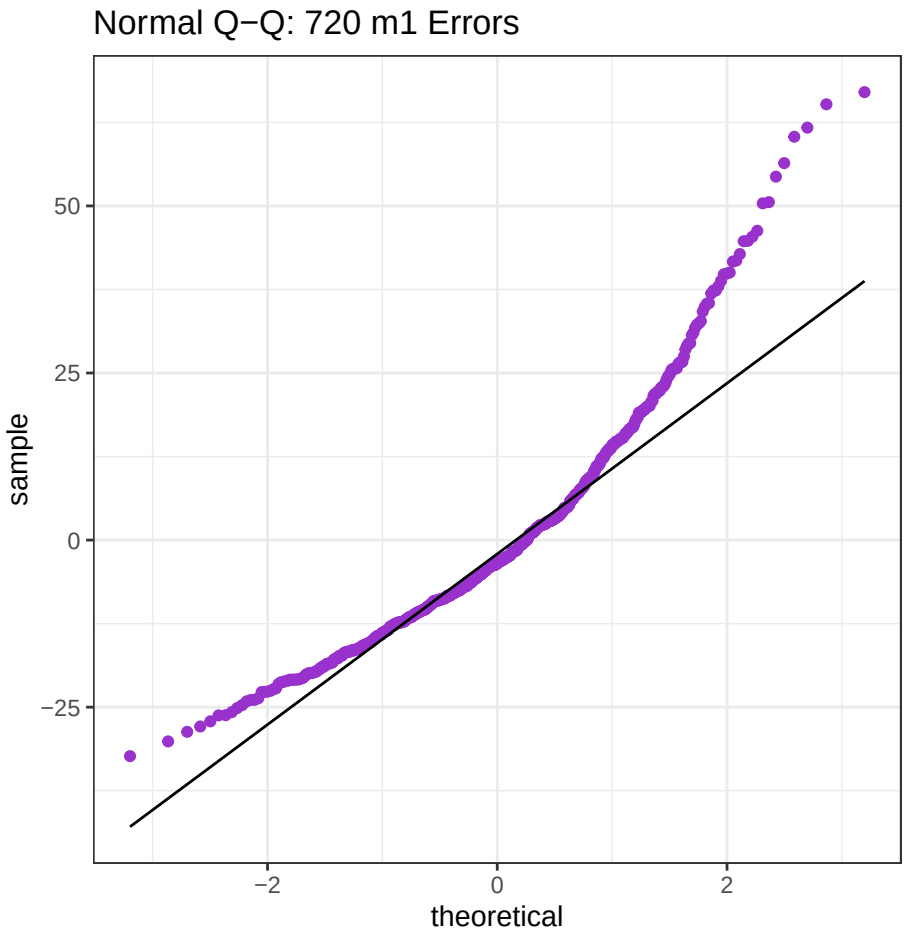
id	sbp	dbp	.fitted	.resid
62172	103	72	121.51	-18.51
62180	107	66	116.74	-9.74
62199	110	65	115.95	-5.95
62205	122	87	133.44	-11.44
62223	105	69	119.13	-14.13
62228	114	74	123.10	-9.10

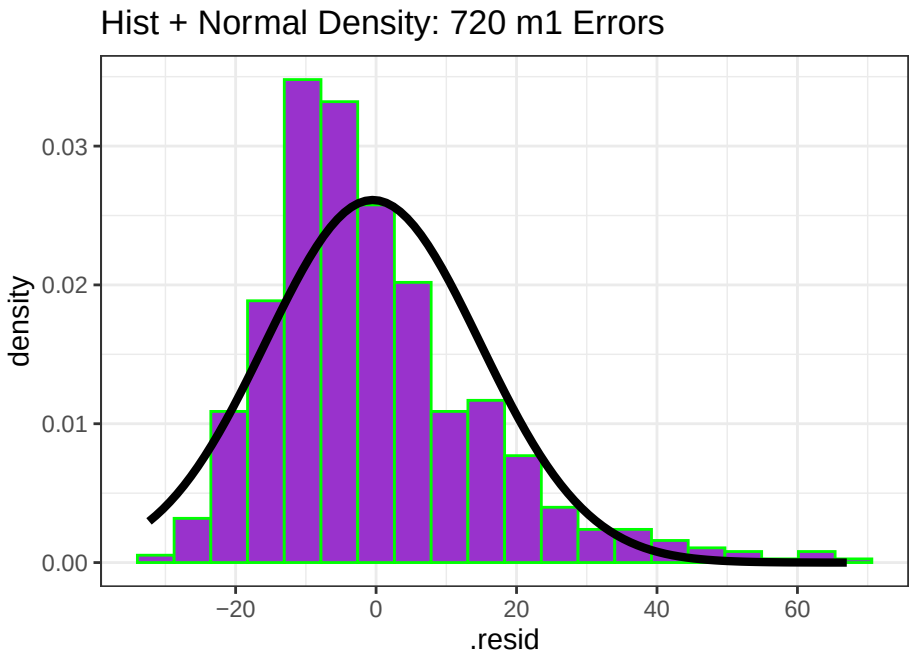
Actual SBP vs. Fitted SBP by `m1` (n = 720)

```
ggplot(new720_nh3, aes(x = .fitted, y = sbp)) +
  geom_abline(slope = 1, intercept = 0, col = "red") +
  geom_point() + theme(aspect.ratio = 1)
```



New Sample (n = 720): `m1` Prediction Errors

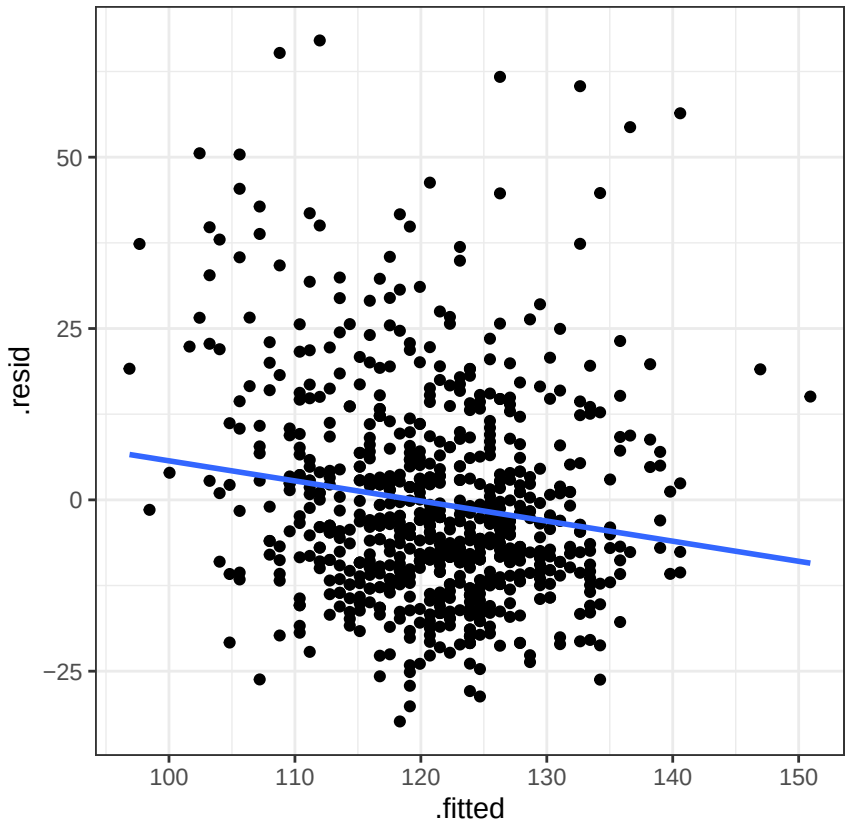




min	Q1	median	Q3	max	mean	sd	n	missing
-32.3	-10.7	-3.5	6.5	67	-0.5	15.3	720	0

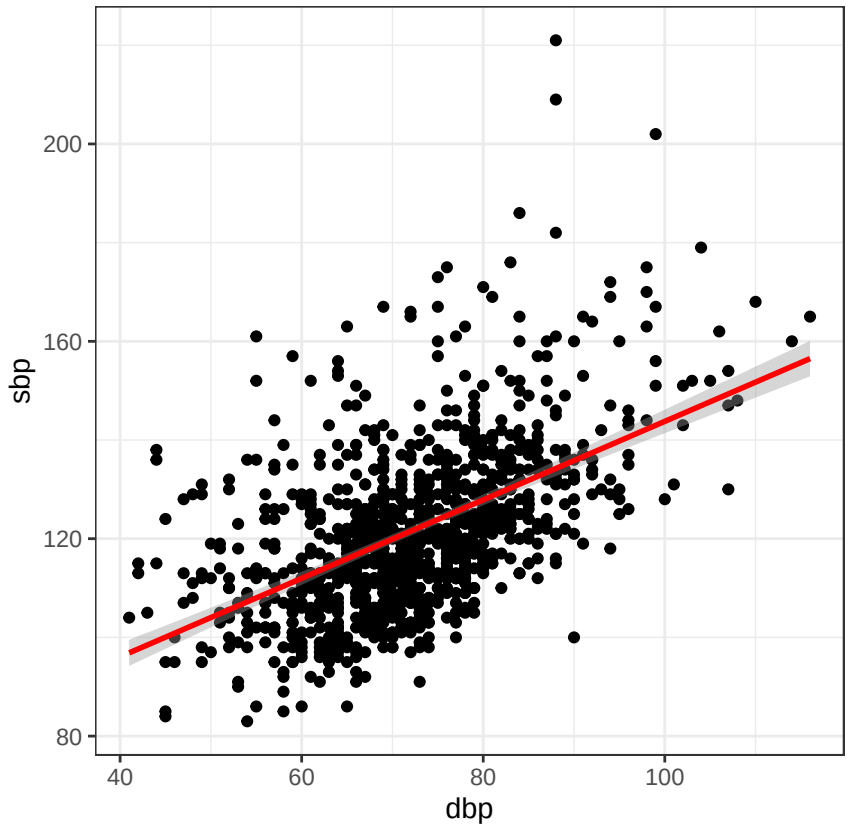
Prediction Errors vs. Fitted SBP (n = 720)

```
ggplot(new720_nh3, aes(x = .fitted, y = .resid)) +
  geom_point() + theme(aspect.ratio = 1) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE)
```



Back to sbp and dbp. Does m1 work well here?

```
ggplot(nh3_new, aes(x = dbp, y = sbp)) +
  geom_point() + theme(aspect.ratio = 1) +
  geom_smooth(method = "lm", formula = y ~ x,
              col = "red", se = TRUE)
```



Is this the only linear model R can fit to these data?

Nope.

Fit linear model using stan_glm?

```
## this is why we ran library(rstanarm)

m2 <- stan_glm(sbp ~ dbp, data = nh3_new)
```

```
SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.001 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transiti
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

Bayesian fitted linear model for our sbp data

```
print(m2)
```

```
stan_glm
 family:       gaussian [identity]
 formula:      sbp ~ dbp
 observations: 989
 predictors:   2
------
            Median MAD_SD
(Intercept) 64.4    3.0
dbp          0.8    0.0

Auxiliary parameter(s):
      Median MAD_SD
sigma 14.5    0.3

------
```

Is the Bayesian model (with default prior) very different from our `lm` in this situation?

```
coef(m1) # fit with lm
```

```
(Intercept)         dbp
 64.2697456   0.7950429
```

```
coef(m2) # stan_glm with default priors
```

```
(Intercept)         dbp
 64.3647450   0.7937923
```

Note that we could use `tidy` and other `broom` functions for the `lm` model but not (yet) for the `stan_glm` model.

Again, consider sbp and dbp. Does m1 work well?

```
ggplot(nh3_new, aes(x = dbp, y = sbp)) +
  geom_point() + theme(aspect.ratio = 1) +
  geom_smooth(method = "loess", formula = y ~ x,
              col = "blue", se = TRUE)
```