

431 Class 01

thomaselove.github.io/431

2021-08-24

This is PQHS 431 / CRSP 431 / MPH 431

I USED TO THINK
CORRELATION IMPLIED
CAUSATION.



THEN I TOOK A
STATISTICS CLASS.
NOW I DON'T.



SOUNDS LIKE THE
CLASS HELPED.

WELL, MAYBE.



First Activity

First Thing: Write down your guess of Dr. Love's age in years in the appropriate spot on the convenient piece of paper we've provided. Hang on to the paper, as you'll need it again later.

Here's a picture of me, in case that's helpful.



Course Details

Instructor: Thomas E. Love, Ph.D.

Email (best way to reach me): [Thomas dot Love at case dot edu](mailto:Thomas.dot.Love.at.case.dot.edu)

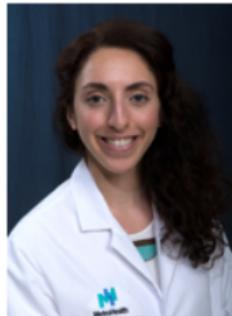
Our web site: <https://thomaselove.github.io/431/>

Links there to:

- Course Syllabus
- Course Calendar, which is the final word for all deadlines, and also links to each day's class page.
- Course Notes (essentially a textbook)
- Software Details (R and R Studio, installation, data and code downloads)
- Assignments: Labs, Minute Papers, Quizzes and Projects

Getting Help: Piazza is your first step (please accept the invitation)

Teaching Assistants (office hours begin 2021-08-29)



Stephanie



Wyatt



Ali



Shiying



Marie



Julia



Monika



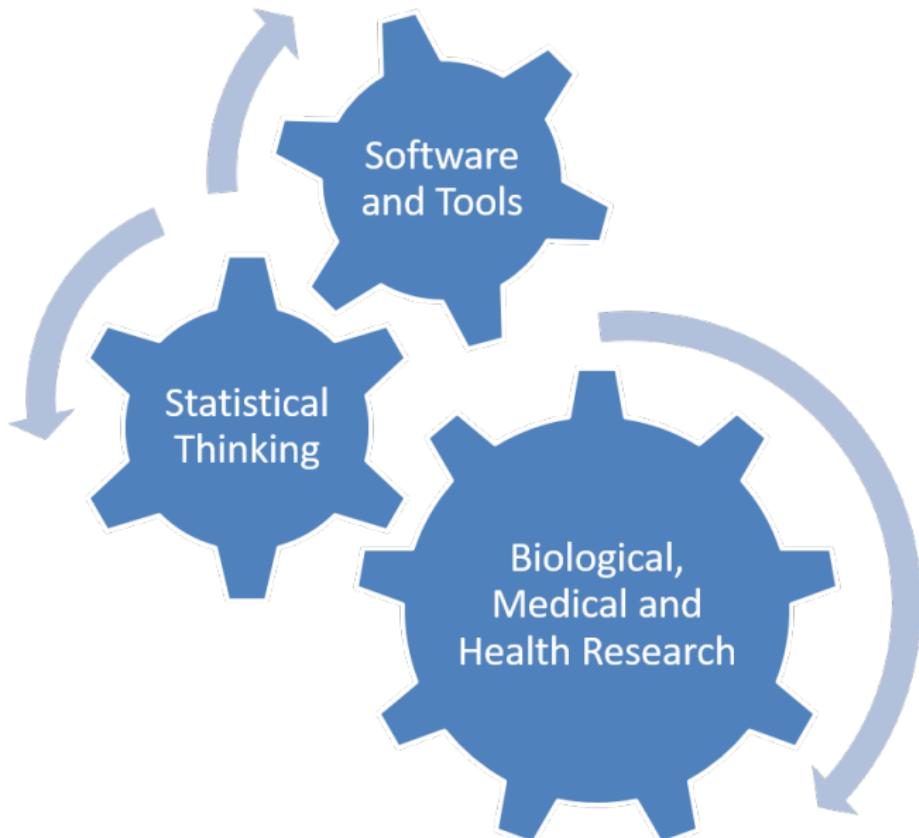
Dipak



Yanning

- TA **office hours** Zoom links posted on our Shared Google Drive.

What is this course about?



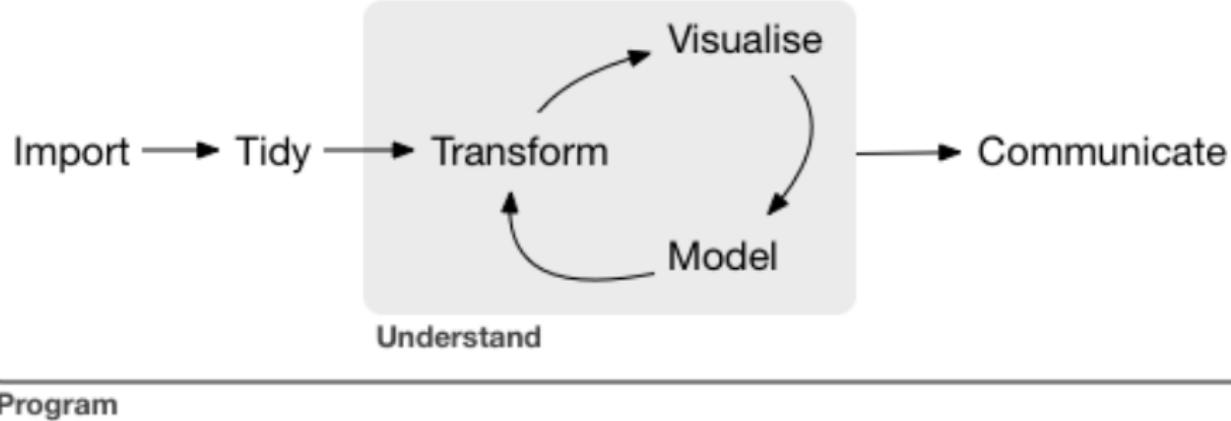
What is this course about?

- Exploratory Data Analysis, Visualization
- Statistical Inference, Making Comparisons
- Linear Regression and related Models

The course is about biostatistics, replicable research, using state-of-the-art tools (R, R Studio, R Markdown), and thinking about how science is most effectively done.

- It is more a course in **how** to do things (highly applied) rather than a theoretical/mathematical justification for **why** we do them. We focus here on practical work.
- It's mostly about getting you doing data science projects for biological, medical and health applications.

What is Data Science about?



Source: <http://r4ds.had.co.nz/introduction.html>

What will we be reading?

Links in the Readings section of [our Calendar](#)

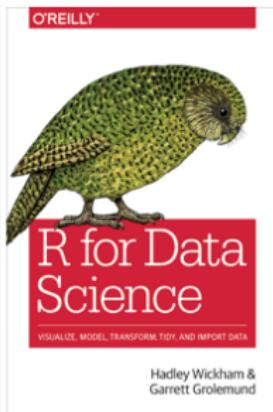
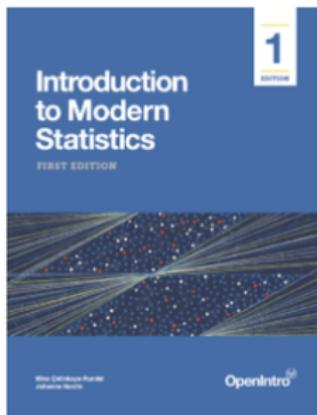
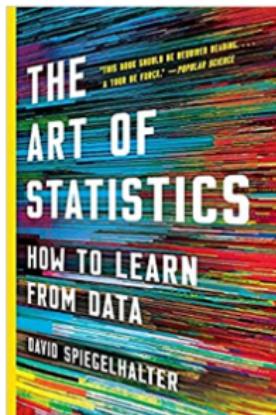
Data Science for
Biological, Medical and
Health Research: Notes
for PQHS/CRSP/MPHP
431

[Search](#)

[Table of contents](#)

[Working with These Notes](#)

[1 Data Science](#)



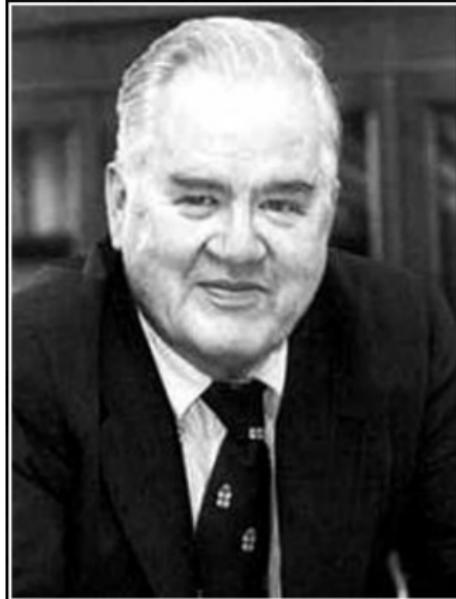
Attendance

Please understand that it is not necessary for you to tell us if you're going to miss any particular class session. Just catch up before the next class. You are responsible for getting everything done, but we certainly understand that things happen, and we will do what we can to be flexible.

When we have assignments, those have deadlines. Please meet them, and please get in touch early if meeting those deadlines will not be possible. The Syllabus has more details, and the deadlines are in the Calendar.

If you will need to miss **more than two classes** in a row, or if you are not able to keep up with watching the recordings because of some external issue, that's when you need to get in touch with Dr. Love.

Great Statisticians in History



The greatest value of a picture is
when it forces us to notice what we
never expected to see.

— *John Tukey* —

AZ QUOTES

Photo Source: http://www.azquotes.com/author/14847-John_Tukey

John Tukey (1915-2000)

Gathering Some Data: Age Guessing Activity

- Shortly, we will be asking you to form into a few groups, each with 3-5 people.
- Pay attention to the number of your breakout group.
- One member of your group will then need to open a Google Form on their laptop. I'll provide the link in a moment.
- After a few brief introductory questions, your group will then see one of a series of 10 photos, each of a person.
- For each photo your group will ...
 - estimate the age of the person in the photo (in years)
 - type your (group) guess into the form (so if you guess age 35, you will just type 35.)
- When you've produced all 10 guesses, submit the form. The submitter will get an email confirmation.
- Later, you will be told the true ages and we'll be able to compute errors.

Seeing the Form

One member of your group needs to visit

<https://bit.ly/431-2021-class01-breakout>

- To see the form, you must log into Google using your CWRU account.
- The rest of you can also visit that form, if you like, but only one of you should fill it out for your group.

If you have a little extra time, make sure everyone in your group knows the name and field of everyone else in the group, and knows your group's name.

Here come the photos

OK. Here come the photos. We'll give you a little more time for the first two than we will for the remaining ones.

Remember to have one member of your group fill out the form at <https://bit.ly/431-2021-class01-breakout> although it may help all of you to keep track on paper, as well.

Photo 1



Photo 2



Making Progress

- Your guesses should be going into the form at <https://bit.ly/431-2021-class01-breakout> and you might also want to keep track on your convenient piece of paper, so that when I tell you the ages later, you'll be in a position to see how your group did.
- In spare time between photos, please make the effort to learn the names of the other people in your group, and perhaps what field they are in.

Photo 3



Photo 4



Photo 5



Photo 6



Photo 7



Photo 8



Photo 9



Photo 10



Guess My Age

- ① You should have an initial guess of my age written down from the start of the session.
- ② Now, if you haven't done so already, make a second guess of my age based on what you know about me now, and write that down next to the initial guess.

So if you guessed 18 initially, but now think I'm 19, you should write 18/19. If you still think I'm 18, write 18/18. Make it easy for us to understand your guess.

Age Guessing Robots?

Well, Microsoft used to have a tool online at how-old.net to do this. You can still play “Guess My Age” at <https://www.guessmyage.net/>.

<https://how-old.net>



The AI's guess was...

7 years too high

6 years too low

Do you think you did that well?

OK. Back to the photos!

Card 1



Card 1

Eric Chong
Master Chef Canada winner
Photo date: April 2014

Age 21

Card 2



Card 2

Katherine Archuleta
Former U.S. OPM Director
Photo date: 2013

Age 64

Card 3



Card 3

Elise Mayfield
Chef, Actor, Baker
Photo date: 2014

Age 28

Card 4



Card 4

Kevin Love
(then) High School Student
Photo date: June 2014

Age 14

No, not THAT Kevin Love



THIS Kevin Love, on the right (January 2019)



Card 5



Card 5

Rosemary McGinn

Photo date: July 2013

Age 54

Card 6



Card 6

John Chaney
Basketball Coach
Photo date: 2006

Age 74

Card 7



Card 7

David Storm

Photo date: August 2014

Age 44

Card 8



Card 8

Margo Glantz
Writer
Photo date: 2013

Age 83

Card 9



Card 9

Quade Ross Honey
Fugitive
Photo date: 2012

Age 24

Card 10



Card 10

Bianca Lawson
Actress
Photo date: 2013

Age 34

How did the AI at <https://how-old.net> do?



#1 Age 21
AI guess 27



#2 Age 64
AI 44



#3 Age 28
AI 22



#4 Age 14
AI 19



#5 Age 54
AI 36



#6 Age 74
AI 63



#7 Age 44
AI 55



#8 Age 83
AI 79



#9 Age 24
AI 35

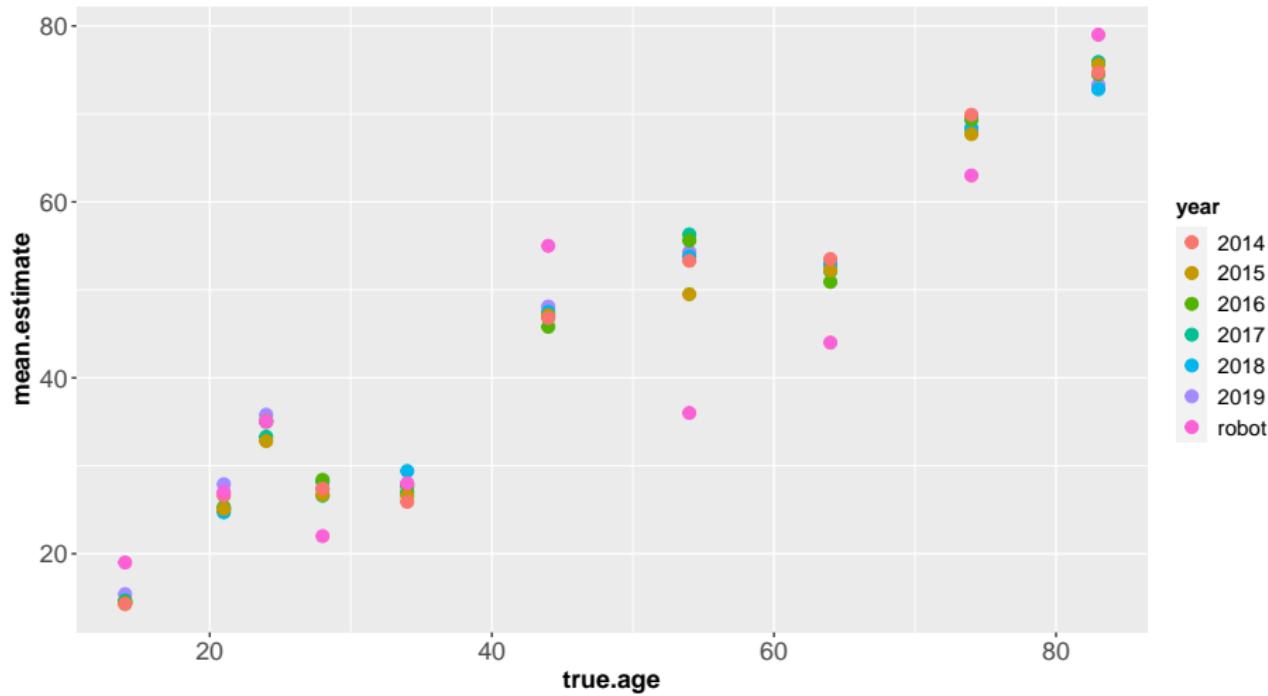


#10 Age 34
AI 28

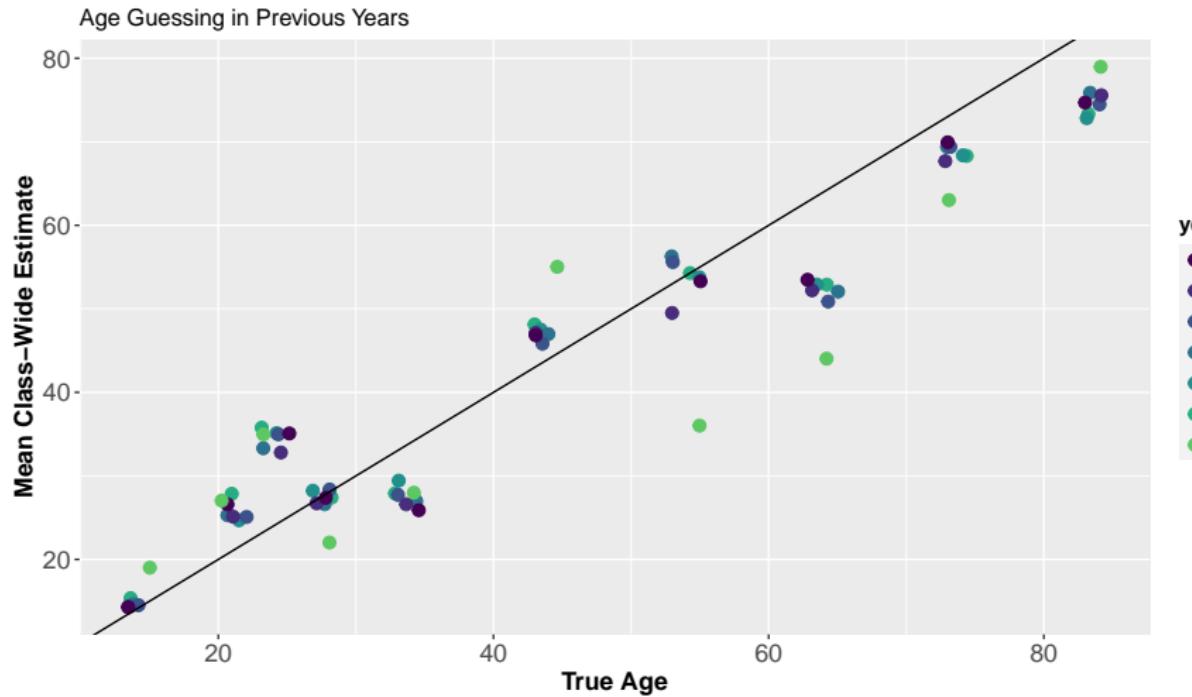
photo-age-history-2019.csv Data Set (excerpt)

card	label	true.age	sex	facing	year	mean.estimate	error
1	Chong	21	M	R	2019	27.9	6.9
2	Archuleta	64	F	L	2019	52.9	-11.1
3	Mayfield	28	F	L	2019	27.4	-0.6
4	Love	14	M	L	2019	15.4	1.4
5	McGinn	54	F	R	2019	54.3	0.3
6	Chaney	74	M	L	2019	68.3	-5.7
7	Storm	44	M	R	2019	48.1	4.1
8	Glantz	83	F	L	2019	73.3	-9.7
9	Honey	24	M	L	2019	35.8	11.8
10	Lawson	34	F	R	2019	27.9	-6.1
1	Chong	21	M	R	2018	24.7	3.7
2	Archuleta	64	F	L	2018	52.9	-11.1

Scatterplot of Prior Results, 1



Scatterplot of Prior Results, 2



Mean Class-Wide Guesses (2014-17 combined)



#1 Age 21 2014-17 Mean Guesses	25.5	#2 Age 64 52.2	46.7	#3 Age 28 27.3	75.2	#4 Age 14 14.4	34.1	#5 Age 54 53.7	26.8
#6 Age 74		#7 Age 44		#8 Age 83		#9 Age 24		#10 Age 34	



Mean Class-Wide Errors (2014-17 combined)



#1 Age 21
2014-17
Errors +4.5
 -4.9

#2 Age 64
 -11.8
 +2.7

#3 Age 28
 -0.7
 -7.8

#4 Age 14
 +0.4
 +10.1
 -7.2

#6 Age 74

#7 Age 44

#8 Age 83

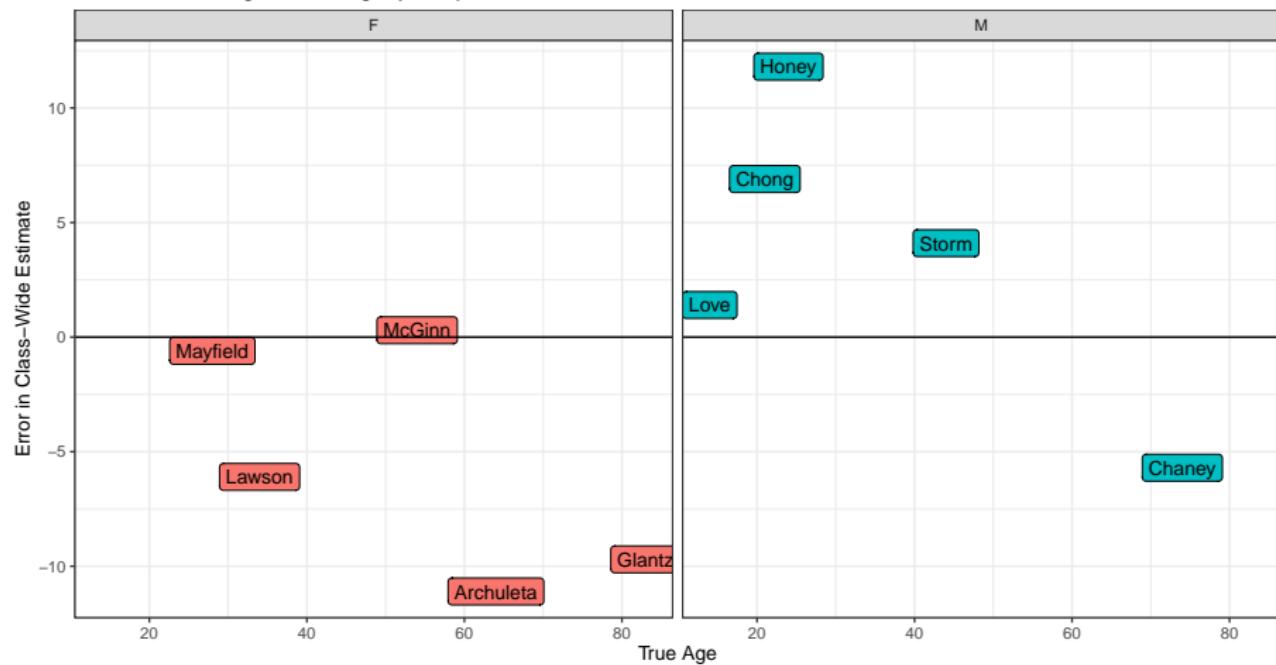
#9 Age 24

#10 Age 34



Scatterplot of 2019 Results with Labels

Errors in 2019 Age Guessing, by Subject's Sex



Hans Rosling and “The Joy of Stats”

200 countries over 200 years using 120,000 numbers, in about 4 minutes.

<http://bit.ly/431-rosling>

And if you liked that ...

- The 20 minute version (from 2007):
<https://www.youtube.com/watch?v=RUwS1uAdUcl>
- The full documentary from the BBC:
<https://www.gapminder.org/videos/the-joy-of-stats/>
- Video playlist from Gapminder: <https://www.gapminder.org/videos/>

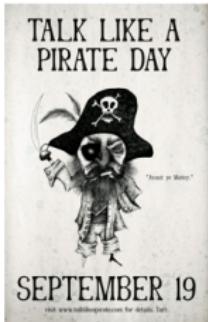
What's next?



RStudio makes R easier to use. It includes a code editor, debugging & visualization tools.



R Packages



R Markdown

from Studio

Analyze. Share. Reproduce.

Your data tells a story. Tell it with R Markdown.
Turn your analyses into high quality documents, reports, presentations and dashboards.

What's next?

- Return the “Convenient Piece of Paper” to us so we can collect data on your guesses of my age.
- See the Ten Things To Do After Class on our Class 01 README.

431 Class 02

thomaselove.github.io/431

2021-08-26

Instructions for the Quick Survey

Please read these instructions carefully before writing anything down.

- ① Introduce yourself to someone that you don't know.
- ② Record the survey answers **for that other person**, while they record your responses.
- ③ Be sure to complete all 15 questions (both sides of the paper).
- ④ When you are finished, thank your partner and raise your hand.
Someone will come to collect your survey.

Regarding Question 4 on the Quick Survey, Professor Love is the large fellow standing in the front of the room.

Today's Agenda

- Data Structures and Variables
- Evaluating some of the Survey variables
- Using R to look at a little data
- Guessing Dr. Love's Age

Class 03 will be a Pre-Recorded Demonstration

We won't meet as a group for Class 03. Instead, I will provide a recording of myself using R and R Studio to:

- create a Project in RStudio
- create an R Markdown document to obtain and describe results
- ingest a data set from the wild into an R data frame
- manage that data frame so that it becomes a tidy tibble
- use that tibble to learn about the data by iterating through
 - visualizing the data
 - transforming the data
 - modeling the data (with a few simple summaries)
- interspersing code and text in my R Markdown document
- “knitting” my R Markdown document into an attractive HTML result

All materials will be posted as soon as possible to the Class 03 page.

Today's Package Loading

```
library(janitor)
library(googlesheets4)
library(patchwork)
library(tidyverse)
```

If you actually run this, you will get some messages which we will suppress and ignore today.

Thinking about The Quick Survey

Chatfield's Six Rules for Data Analysis

- ① Do not attempt to analyze the data until you understand what is being measured and why.
- ② Find out how the data were collected.
- ③ Look at the structure of the data.
- ④ Carefully examine the data in an exploratory way, before attempting a more sophisticated analysis.
- ⑤ Use your common sense at all times.
- ⑥ Report the results in a clear, self-explanatory way.

Chatfield, Chris (1996) *Problem Solving: A Statistician's Guide*, 2nd ed.

Types of Data

Data can be **quantitative (numerical)** or **qualitative (categorical)**

- **Quantitative**

- Variables recorded in numbers that we use as numbers.
- All quantitative variables must have units of measurement.
- Can break into *continuous* (may take any value in a range) or *discrete* (limited set of potential values.)
 - Height is certainly continuous as a concept, but how precise is our ruler?
 - Piano vs. Violin
- (less common) *interval* (equal distances between values, but zero point is arbitrary) as compared to *ratio* variables (a meaningful zero point.)
 - Is *weight* an interval or ratio variable? How about *IQ*?
- Taking a mean or median is a reasonable idea.

Types of Data

Data can be **quantitative (numerical)** or **qualitative (categorical)**

- Qualitative
 - Variables consisting of names of categories.
 - Each possible value is a code for a category (could use numerical or non-numerical codes.)
 - *Binary* categorical variables (two categories, often labeled 1 or 0)
 - *Multi-categorical* variables (usually taken to be 3+ categories)
 - Also, *nominal* (no underlying order) or *ordinal* (categories are ordered.)
 - How is your overall health? (Excellent, Very Good, Good, Fair, Poor)
 - Which candidate would you vote for if the election were held today?
 - Did this patient receive this procedure?

Our Quick Survey

431 Quick Survey for 2021: Class 02 (15 Questions)

Please introduce yourself to someone near you, ask them these 15 questions, and record their answers on this sheet. At the same time, provide your partner with your answers so they can record your responses on their sheet. Do not place any names on this sheet so that the responses will remain anonymous. Thank you!

1. Do you wear corrective lenses (contacts or glasses)? (Yes or No) _____

2. Is English your *most comfortable* language? (Yes or No) _____

3. Fill in the number that best describes your answer to this question:

Has statistical thinking been important in your life so far?						
Not at all important	Slightly important	Somewhat important	Extremely important			
①	②	③	④	⑤	⑥	⑦

4. How tall do you think Dr. Love is? (Please indicate units.) _____

5. Do you smoke? Fill in the appropriate circle:

No	I used to.	Yes.
Non-Smoker	Former Smoker	Smoker

6. Please indicate which hand you use for each of the following activities by putting a + in the appropriate column, or ++ if you would *never* use the other hand for that activity. If, in any case, you really are indifferent, put + in both columns.

Task	Left	Right
Writing		
Drawing		
Throwing		
Scissors		
Toothbrush		
Knife (without fork)		
Spoon		
Broom (upper hand)		
Striking match (hand that holds the match)		
Opening box (hand that holds the lid)		
Total Count of +s:		

$$\text{Right} - \text{Left} = \underline{\hspace{2cm}} \quad \text{Right} + \text{Left} = \underline{\hspace{2cm}} \quad \frac{\text{Right} - \text{Left}}{\text{Right} + \text{Left}} = \underline{\hspace{2cm}}$$

431 Quick Survey for 2021: Class 02 (15 Questions)

7. How important do you think statistics will be in your *future career*?

Not at all important	Slightly important	Somewhat important	Extremely important
①	②	③	④

8. How much did you pay for your most recent haircut? (in \$): _____

Please indicate your agreement with the following statements:

	Strongly Disagree	Agree	Strongly Agree
9. I prefer to learn from lectures than to learn from activities.	1	2	3
10. I prefer to work on projects alone than in a team.	1	2	3

11. What is your height (indicate units of measurement): _____

12. Use the ruler provided on the side of this page to measure the span of your right hand (distance from the thumb to the little finger when your fingers are spread apart): _____ cm.

13. What is your favorite color? _____

14. How many hours did you sleep last night? _____ hours.

15. Record your pulse by counting the beats of your heart for 15 seconds, then quadrupling the result: _____ beats/minute.

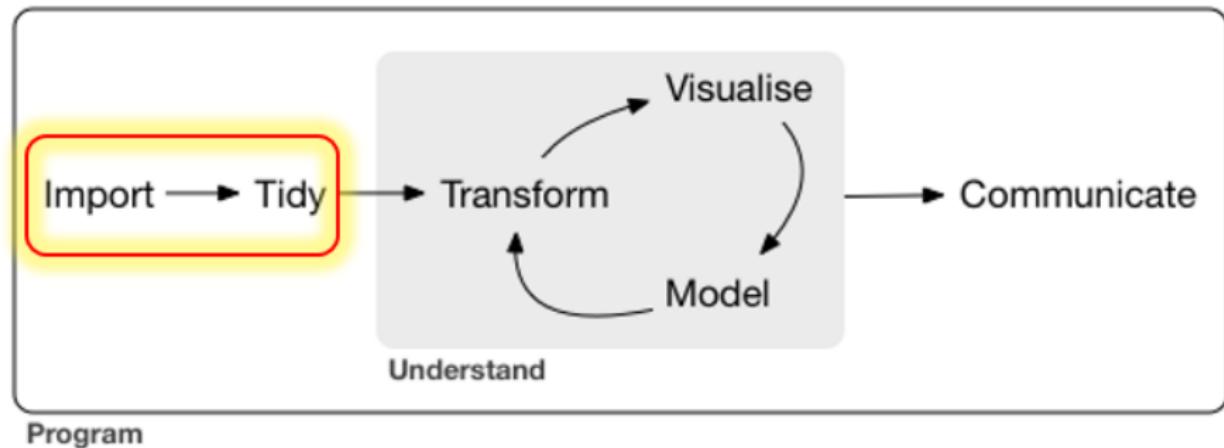
Evaluating Some Quick Survey variables

- ① Do you **smoke**? (1 = Non-Smoker, 2 = Former Smoker, 3 = Smoker)
- ② How much did you pay for your most recent **haircut**? (in \$)
- ③ What is your favorite **color**?
- ④ How many hours did you **sleep** last night?
- ⑤ Statistical thinking in your future **career**? (1 = Not at all important to 7 = Extremely important)

Are these quantitative or qualitative?

- If quantitative, are they *discrete* or *continuous*? Do they have a meaningful *zero point*?
- If qualitative, how many categories? *Nominal* or *ordinal*?

Importing and Tidying Data



Ingesting the Quick Surveys

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	student	sex	glasses	english	statsofar	ageguess	smoke	h.left	h.right	handedness	statfuture	haircut	lecture	alone	height.in	hand.s
2	201901	NA	y	y	6	42	1	1	19							
3	201902	NA	y	y	7	53	1	19	10							
4	201903	NA	y	y	4	45	1	0	10							
5	201904	NA	y	y	7	45	1	16	10							
6	201905	NA	y	y	6	42	1	2	16							
7	201906	NA	y	y	7	50	1	10	0							
8	201907	NA	y	y	5	56	1	1	13							
9	201908	NA	n	n	6	50	1	0	10							
10	201909	NA	n	y	6	52	1	0	17							
11	201910	NA	n	y	4	42	1	18	10							
12	201911	NA	n	n	5	43	1	5	13							
13	201912	NA	y	y	5	52	1	1	13							
14	201913	NA	y	y	7	50	2	1	19							
15	201914	NA	y	y	4	50	1	1	9							
	-----	-----	-----	-----	-	--	-	-	--							

The screenshot shows a Google search results page for "165 cm to inches". The search bar at the top contains the query. Below the search bar, there's a snippet of text: "About 20,900,000 results (0.56 seconds)". On the right side of the search bar, there's a conversion calculator. It has two input fields: one for "Length" containing "165" and another for the unit "Centimeter". To the right of the equals sign, it shows the result "64.9606" in the unit "Inch". Below the calculator, there's a note: "Formats divide the length value by 2.54".

The Quick Survey

315 people before you have taken (essentially) this same survey in the same way.

Fall	2019	2018	2017	2016	2015	2014	Total
<i>n</i>	61	51	48	64	49	42	315

Question

About how many of those 315 surveys caused *no problems* in recording responses?

The 15 Survey Items

#	Topic	#	Topic
Q1	glasses	Q9	lectures_vs_activities
Q2	english	Q10	projects_alone
Q3	stats_so_far	Q11	height
Q4	guess_TL_ht	Q12	hand_span
Q5	smoke	Q13	color
Q6	handedness	Q14	sleep
Q7	stats_future	Q15	pulse_rate
Q8	haircut	-	-

- At one time, I asked about sex rather than glasses.
- In prior years, people guessed my age, rather than height here.
- Sometimes, I've asked for a 30-second pulse check, then doubled.

Response to the Question I asked

About how many of those 315 surveys caused *no problems* in recording responses?

- Guesses?

Response to the Question I asked

About how many of those 315 surveys caused *no problems* in recording responses?

- Guesses?
- $110/315$ (35%) caused no problems.

Guess My Age

4. How old (in years) do you think Professor Love is?

early fifties years

4. How old (in years) do you think Professor Love is?

late 50's years.

4. How old (in years) do you think Professor Love is?

50ish years.

What should we do in these cases?

English best language?

2. Is English your *most comfortable* language? (Yes or No)

English

TEL Decision: Yes

1. What is your *gender*? (Male or Female)

(Male or Female)

2. Is English your *most comfortable* language? (Yes or No)

(Yes or No)

TEL Decision: NA

Is English your *most comfortable* language? (Yes or No)

maybe

TEL decision: NA

Height

ii. What is your height (indicate units of measurement): 5'4 (inches)

ii. What is your height (indicate units of measurement): 6'0

ii. What is your height (indicate units of measurement): 5'2

e units of measurement): 5'7"

ii. What is your height (indicate units of measurement): 5'5

Handedness Scale (2016-21 version)

6. Please indicate which hand you use for each of the following activities by putting a + in the appropriate column, or ++ if you would *never* use the other hand for that activity. If, in any case, you really are indifferent, put + in both columns.

Task	Left	Right
Writing	++	+
Drawing	++	+
Throwing	++	+
Scissors	++	+
Toothbrush	++	+
Knife (without fork)	++	+
Spoon	++	+
Broom (upper hand)	++	++
Striking match (hand that holds the match)	++	+
Opening box (hand that holds the lid)	++	+
Total Count of +s:	20	11

Favorite color

13. What is your favorite color? depends

NA

13. What is your favorite color? orange

orange

13. What is your favorite color? Blue, Brown

13. What is your favorite color? N/A

Following the Rules?

15. Record your pulse by counting the beats of your heart for 30 seconds, then doubling the result:

75 beats/minute.

2019 pulse responses, sorted ($n = 61, 1 \text{ NA}$)

33	46	48	56	60	60	3		3
62	63	65	65	66	66	4		68
68	68	68	69	70	70	5		6
70	70	70	70	70	70	6		002355668889
71	72	72	74	74	74	7		00000000122444445666888
74	74	75	76	76	76	8		000012445668
78	78	78	80	80	80	9		000046
80	81	82	84	84	85	10		44
86	86	88	90	90	90	11		0
90	94	96	104	104	110			

Stem and Leaf: Pulse Rates, 2014-2019

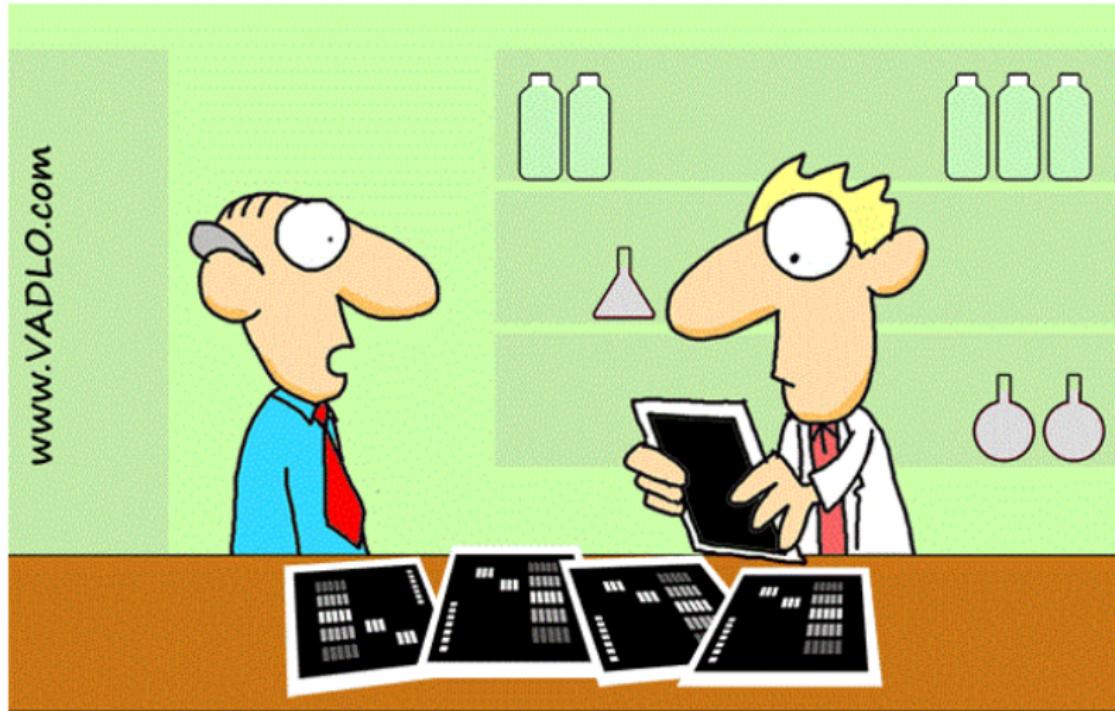
```
> stem(survey1$pulse)
```

The decimal point is 1 digit(s) to the right of the |

3 03
3
4
4 688
5 00022244444
5 566666666667888889
6 00000000000000000000222222222344444444444444
6 555555666666666666666888888888888888888899
7 00000000000000000000000000000000001222222222222444444444444444
7 5555566666666788888888888888
8 000000000000000000000000000000000012222224444444444
8 55666666666688888888
9 000000000001222224444
9 5668888
10 0000444
10 6
11 0

(Thanks, John **Tukey**)

Garbage in, garbage out . . .



“Data don’t make any sense,
we will have to resort to statistics.”

Guessing My Age (from Class 01)

From our Shared Google Drive

I've placed a Google Sheet called `class01_age_guesses_2021-08-24` on our Shared Google Drive. Remember that you have to log into Google via CWRU to see the Drive I've shared with you.

The screenshot shows a Google Sheets interface with the following details:

- Title Bar:** class01_age_guesses_2021-08-24, with icons for star, folder, and cloud.
- Menu Bar:** File, Edit, View, Insert, Format, Data, Tools, Add-ons.
- Toolbar:** Back, Forward, Print, Filter, 100% zoom, Currency (\$), Percentage (%), Decimal (.0), .00, 123, Default (A).
- Table:** A grid of data with columns labeled A, B, C, D and rows numbered 1 through 7. Row 1 is a header row. Rows 2 through 6 contain student names and their two age guesses. Row 7 is a summary row.

	A	B	C	D
1	student	guess1	guess2	
2	S-01		48	52
3	S-02		55	
4	S-03		47	50
5	S-04		53	55
6	S-05		54	54
7	S-06	49	53	

Reading from our Shared Google Drive

We'll use the `read_sheet` function from the `googlesheets4` package to read in data from a Google Sheet. My first step is to copy the URL from the Google Sheet into a temporary object I'll call `temp_url`.

```
temp_url <- "https://docs.google.com/spreadsheets/d/1Mgu_Xj0A8
```

Then I'll ask R to read in the data from the sheet to a new data frame (technically a tibble) called `age_guess`.

```
age_guess <- read_sheet(temp_url)
```

When you do this the first time, R will ask you to verify some things in the browser and allow the browser to pull down the sheet. Let it do so.

What is in the age_guess data frame?

```
age_guess
```

```
# A tibble: 59 x 3
  student guess1 guess2
  <chr>    <dbl>   <dbl>
1 S-01      48     52
2 S-02      55     NA
3 S-03      47     50
4 S-04      53     55
5 S-05      54     54
6 S-06      48     53
7 S-07      53     56
8 S-08      55     NA
9 S-09      52     52
10 S-10     55     54
# ... with 49 more rows
```

What do the guess1 values look like?

```
age_guess %>% select(guess1) %>% arrange(guess1)
```

```
# A tibble: 59 x 1
```

```
  guess1
```

```
  <dbl>
```

```
1     30
```

```
2     43
```

```
3     43
```

```
4     47
```

```
5     47
```

```
6     47
```

```
7     48
```

```
8     48
```

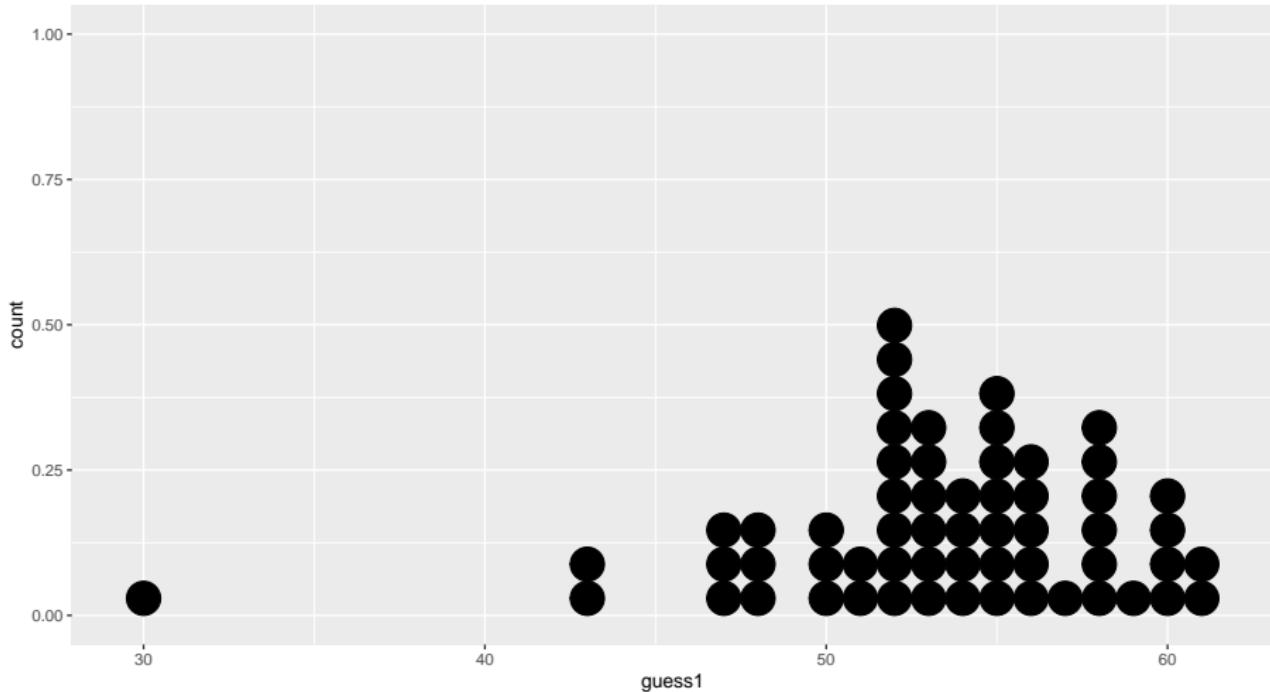
```
9     48
```

```
10    50
```

```
# ... with 49 more rows
```

Plot the guess1 values?

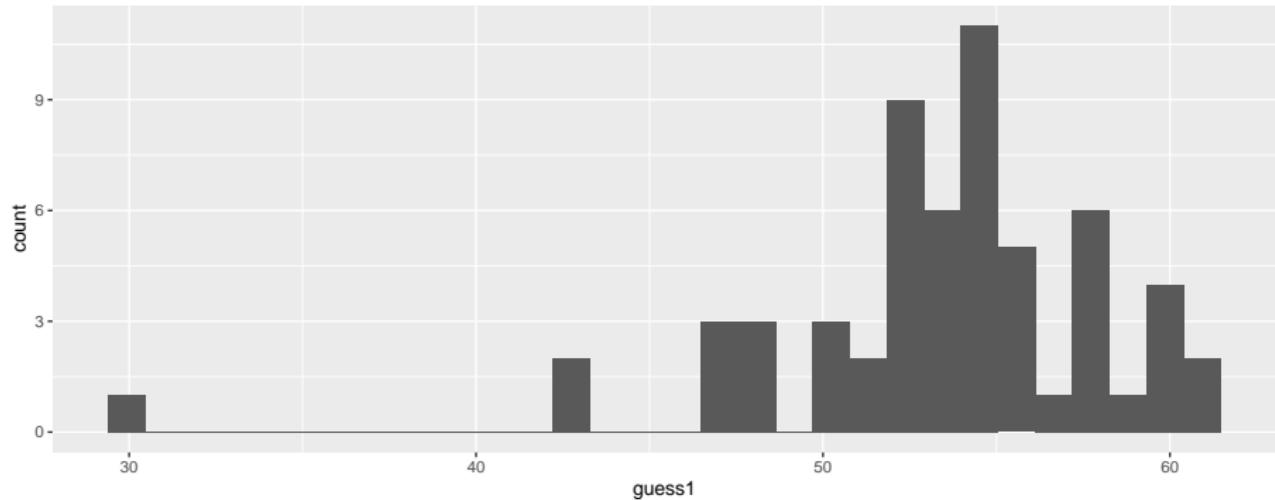
```
ggplot(age_guess, aes(x = guess1)) +  
  geom_dotplot(binwidth = 1)
```



Can we make a histogram?

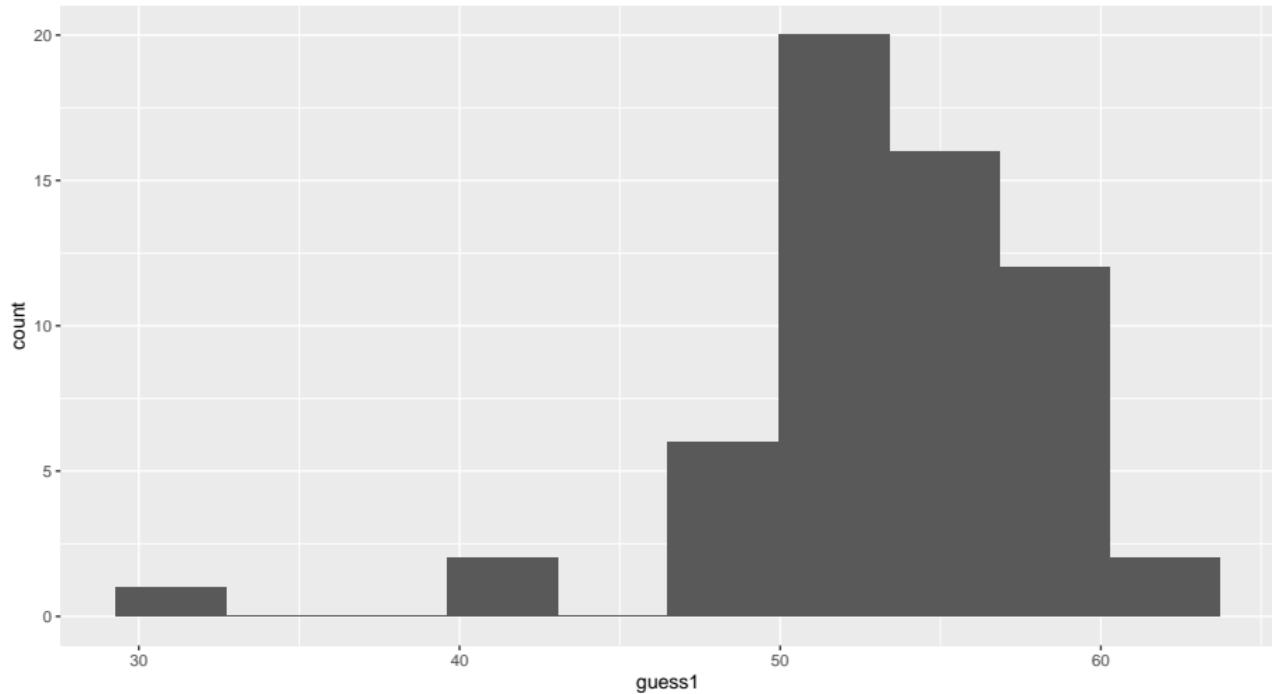
```
ggplot(age_guess, aes(x = guess1)) +  
  geom_histogram()
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



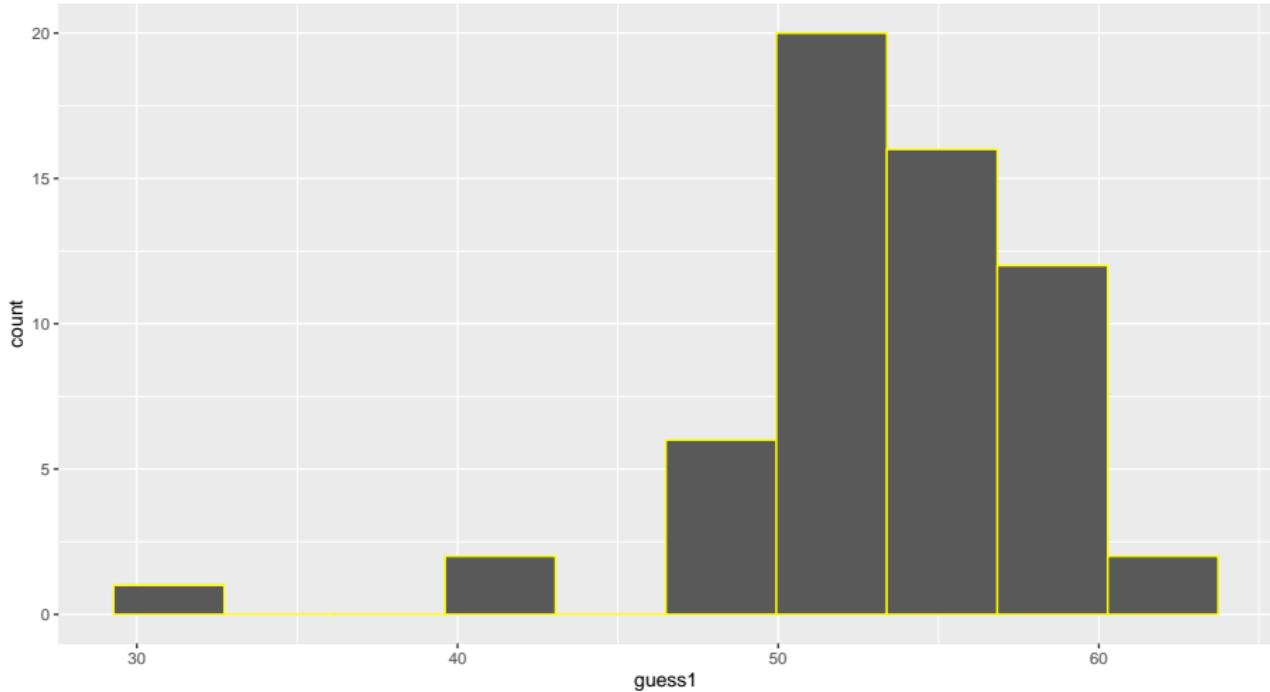
Improving the Histogram, 1

```
ggplot(age_guess, aes(x = guess1)) +  
  geom_histogram(bins = 10)
```



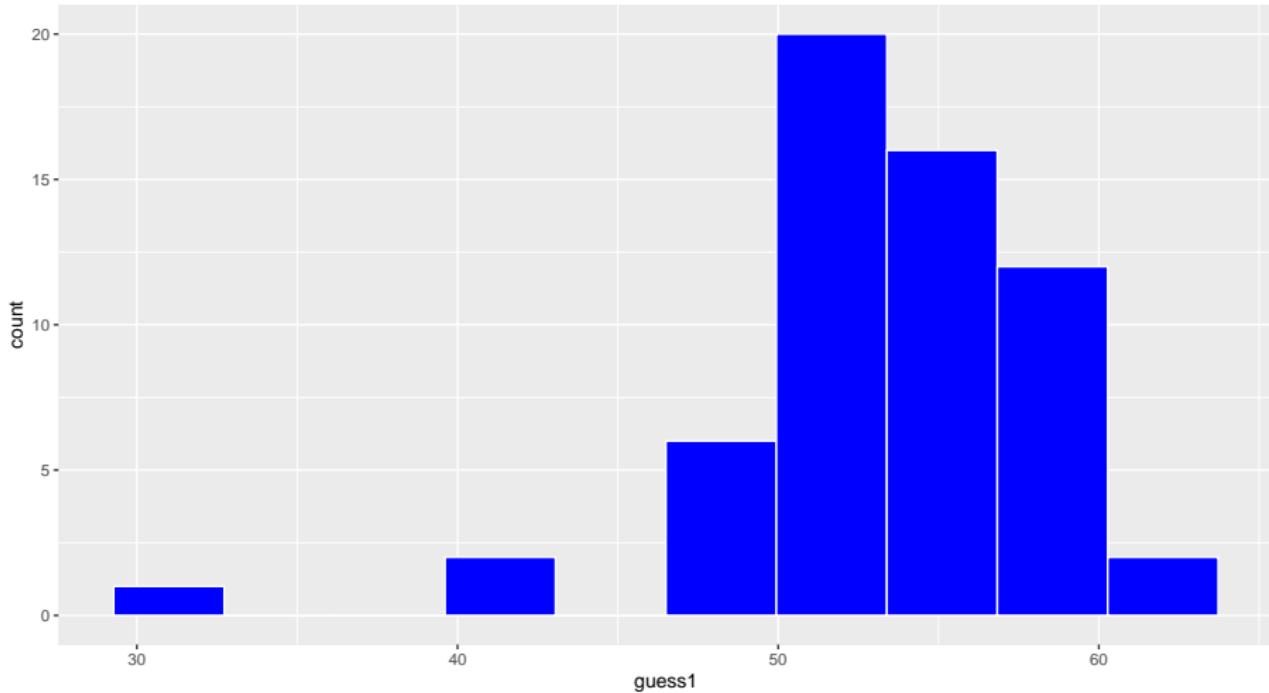
Improving the Histogram, 2

```
ggplot(age_guess, aes(x = guess1)) +  
  geom_histogram(bins = 10, col = "yellow")
```



Improving the Histogram, 3

```
ggplot(age_guess, aes(x = guess1)) +  
  geom_histogram(bins = 10, col = "white", fill = "blue")
```



Improving the Histogram, 4 (code only)

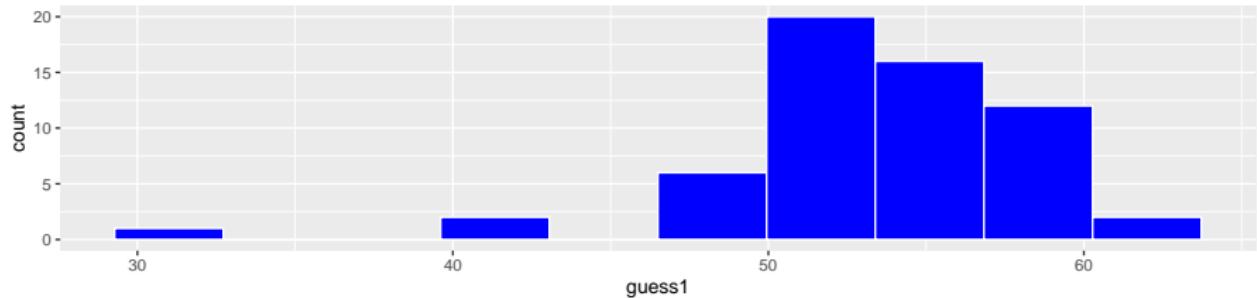
```
ggplot(age_guess, aes(x = guess1)) +  
  geom_histogram(bins = 15, col = "white", fill = "blue") +  
  theme_bw()
```

- We've changed the theme to `theme_bw`
- We've increased the number of bins from 10 to 15.

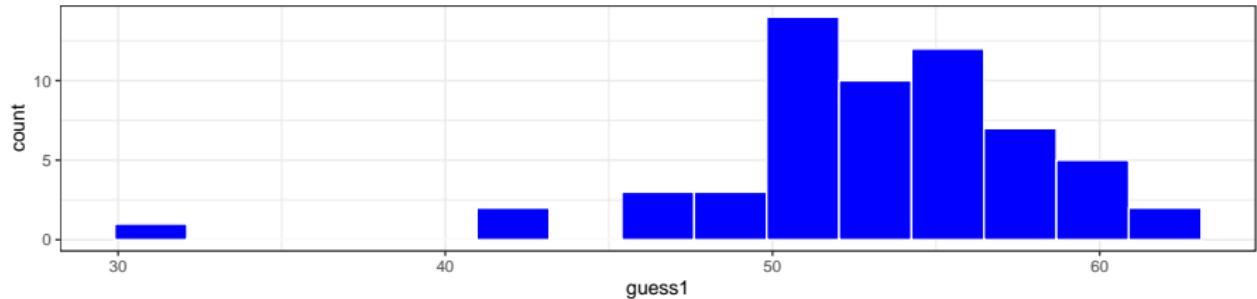
Results of Version 3 and Version 4 shown on the next slide.

Results for Versions 3 and 4

Version 3 (10 bins and default theme)



Version 4 (15 bins, and theme_bw())

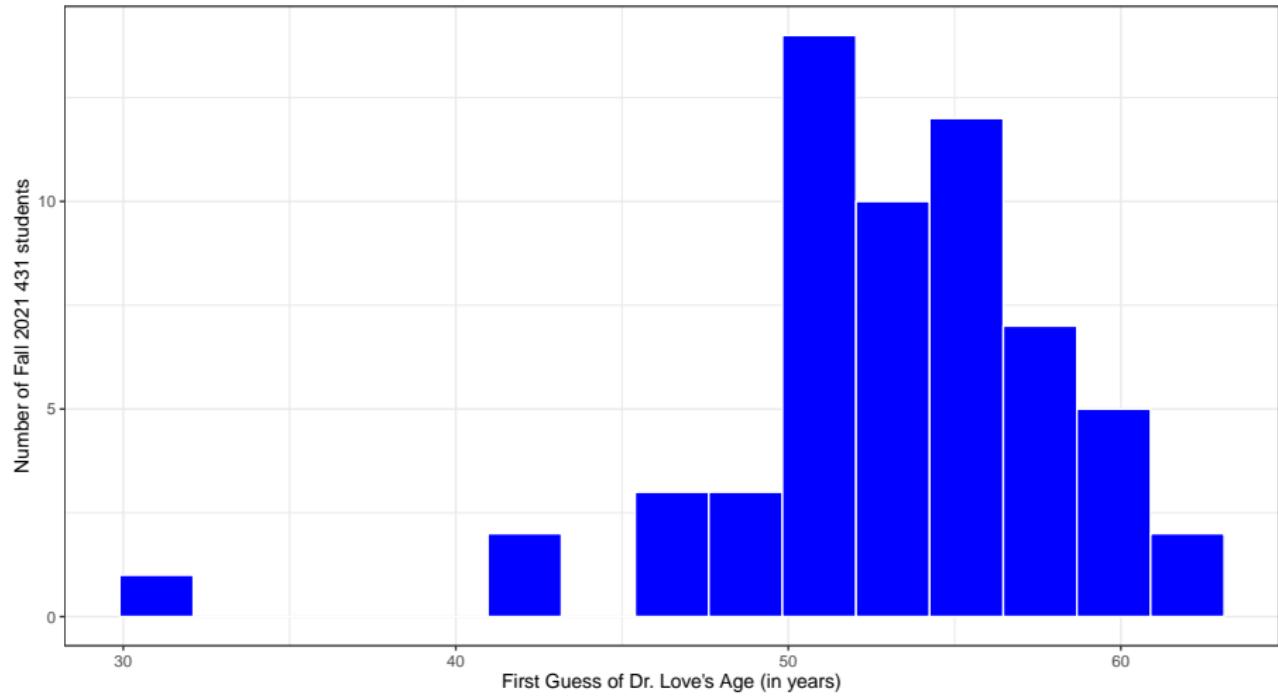


Improving the Histogram, 5 (code only)

```
ggplot(age_guess, aes(x = guess1)) +  
  geom_histogram(bins = 15, col = "white", fill = "blue") +  
  theme_bw() +  
  labs(x = "First Guess of Dr. Love's Age (in years)",  
       y = "Number of Fall 2021 431 students")
```

Here we add axis labels. Result on next slide.

Results of Version 5



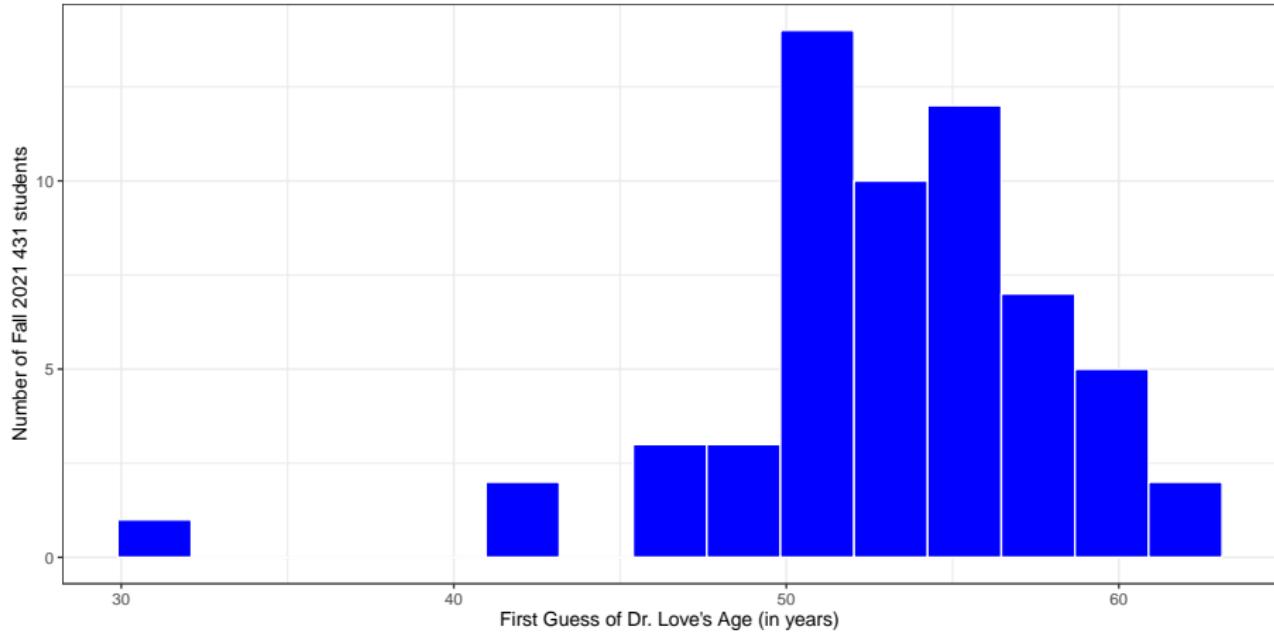
Version 6 adds title and subtitle

```
ggplot(age_guess, aes(x = guess1)) +  
  geom_histogram(bins = 15, col = "white", fill = "blue") +  
  theme_bw() +  
  labs(x = "First Guess of Dr. Love's Age (in years)",  
       y = "Number of Fall 2021 431 students",  
       title = "Most First Guesses were pretty close",  
       subtitle = "Dr. Love's actual age was 54.5")
```

Result of Version 6 code

Most First Guesses were pretty close

Dr. Love's actual age was 54.5



Improving the Histogram, 7

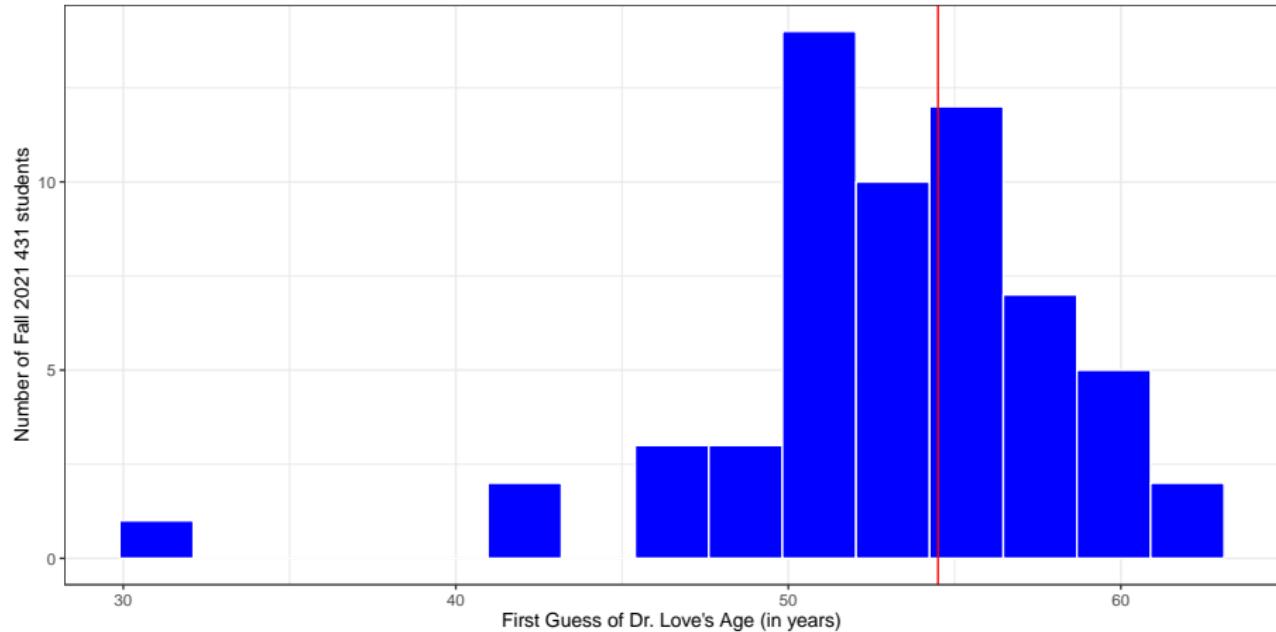
Let's add a vertical line at 54.5 years to show my actual age.

```
ggplot(age_guess, aes(x = guess1)) +  
  geom_histogram(bins = 15, col = "white", fill = "blue") +  
  geom_vline(aes(xintercept = 54.5), col = "red") +  
  theme_bw() +  
  labs(x = "First Guess of Dr. Love's Age (in years)",  
       y = "Number of Fall 2021 431 students",  
       title = "Most First Guesses were pretty close",  
       subtitle = "Dr. Love's actual age was 54.5")
```

Result from Improving the Histogram, 7

Most First Guesses were pretty close

Dr. Love's actual age was 54.5



A Quick Numerical Summary of the Data Frame

```
summary(age_guess)
```

student	guess1	guess2
Length:59	Min. :30.00	Min. :26.00
Class :character	1st Qu.:52.00	1st Qu.:53.00
Mode :character	Median :54.00	Median :55.00
	Mean :53.32	Mean :54.56
	3rd Qu.:56.00	3rd Qu.:57.00
	Max. :61.00	Max. :65.00
	NA's :2	

- Was the average guess closer to my actual age (54.5) on the first or second guess?

A Quick Numerical Summary of the Data Frame

```
summary(age_guess)
```

student	guess1	guess2
Length:59	Min. :30.00	Min. :26.00
Class :character	1st Qu.:52.00	1st Qu.:53.00
Mode :character	Median :54.00	Median :55.00
	Mean :53.32	Mean :54.56
	3rd Qu.:56.00	3rd Qu.:57.00
	Max. :61.00	Max. :65.00
	NA's :2	

- Was the average guess closer to my actual age (54.5) on the first or second guess?
- What was the range of first guesses? Second guesses?

A Quick Numerical Summary of the Data Frame

```
summary(age_guess)
```

student	guess1	guess2
Length:59	Min. :30.00	Min. :26.00
Class :character	1st Qu.:52.00	1st Qu.:53.00
Mode :character	Median :54.00	Median :55.00
	Mean :53.32	Mean :54.56
	3rd Qu.:56.00	3rd Qu.:57.00
	Max. :61.00	Max. :65.00
	NA's :2	

- Was the average guess closer to my actual age (54.5) on the first or second guess?
- What was the range of first guesses? Second guesses?
- What does the NA's : 2 mean in guess2?

A Quick Numerical Summary of the Data Frame

```
summary(age_guess)
```

student	guess1	guess2
Length:59	Min. :30.00	Min. :26.00
Class :character	1st Qu.:52.00	1st Qu.:53.00
Mode :character	Median :54.00	Median :55.00
	Mean :53.32	Mean :54.56
	3rd Qu.:56.00	3rd Qu.:57.00
	Max. :61.00	Max. :65.00
	NA's :2	

- Was the average guess closer to my actual age (54.5) on the first or second guess?
- What was the range of first guesses? Second guesses?
- What does the NA's : 2 mean in guess2?
- Why is student not summarized any further?

Some additional summaries

```
mosaic::favstats(~ guess1, data = age_guess)
```

min	Q1	median	Q3	max	mean	sd	n	missing
30	52	54	56	61	53.32203	5.174342	59	0

```
mosaic::favstats(~ guess2, data = age_guess)
```

min	Q1	median	Q3	max	mean	sd	n	missing
26	53	55	57	65	54.5614	5.067465	57	2

How many first guesses were between 53 and 56?

```
age_guess %>% count(guess1 >= 53 & guess1 <= 56)
```

```
# A tibble: 2 x 2
`guess1 >= 53 & guess1 <= 56`      n
<lgl>                           <int>
1 FALSE                         37
2 TRUE                          22
```

How many second guesses were between 53 and 56?

```
age_guess %>% count(guess2 >= 53 & guess1 <= 56)
```

```
# A tibble: 3 x 2
`guess2 >= 53 & guess1 <= 56`      n
<lgl>                      <int>
1 FALSE                     25
2 TRUE                      32
3 NA                        2
```

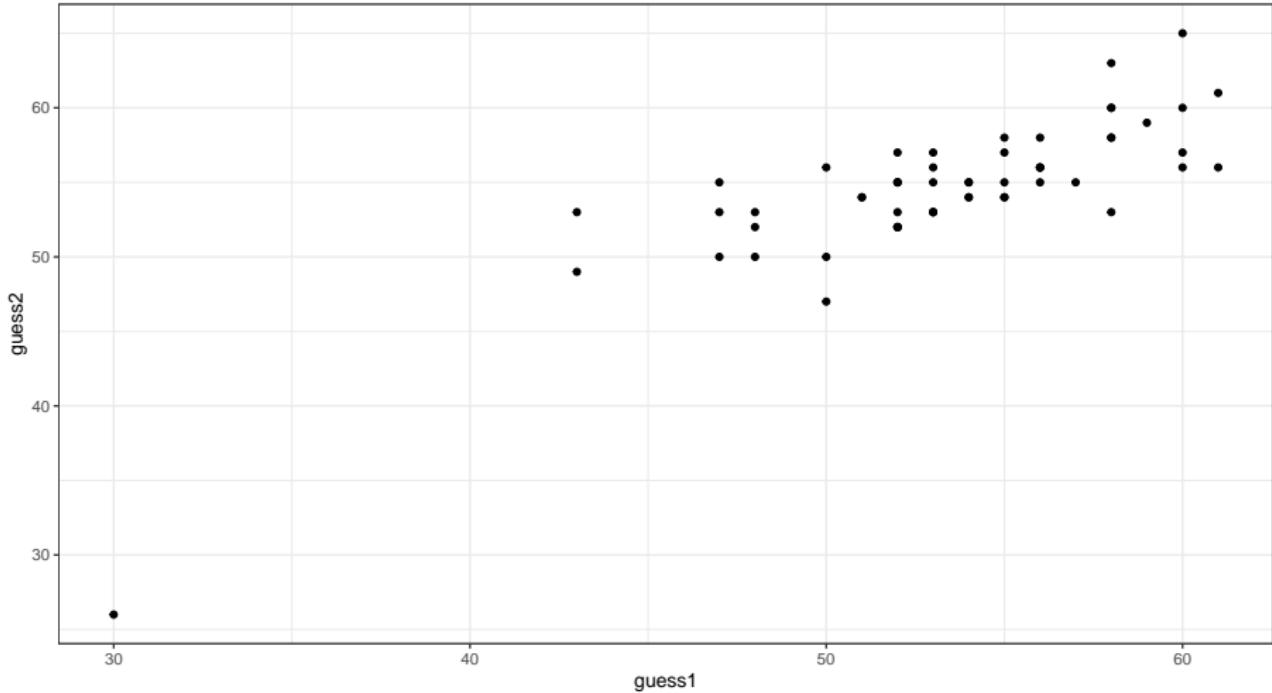
Comparing First Guess to Second Guess

Here's the code. Resulting scatterplot on next slide.

```
ggplot(data = age_guess, aes(x = guess1, y = guess2)) +  
  geom_point() + theme_bw()
```

Comparing First Guess to Second Guess

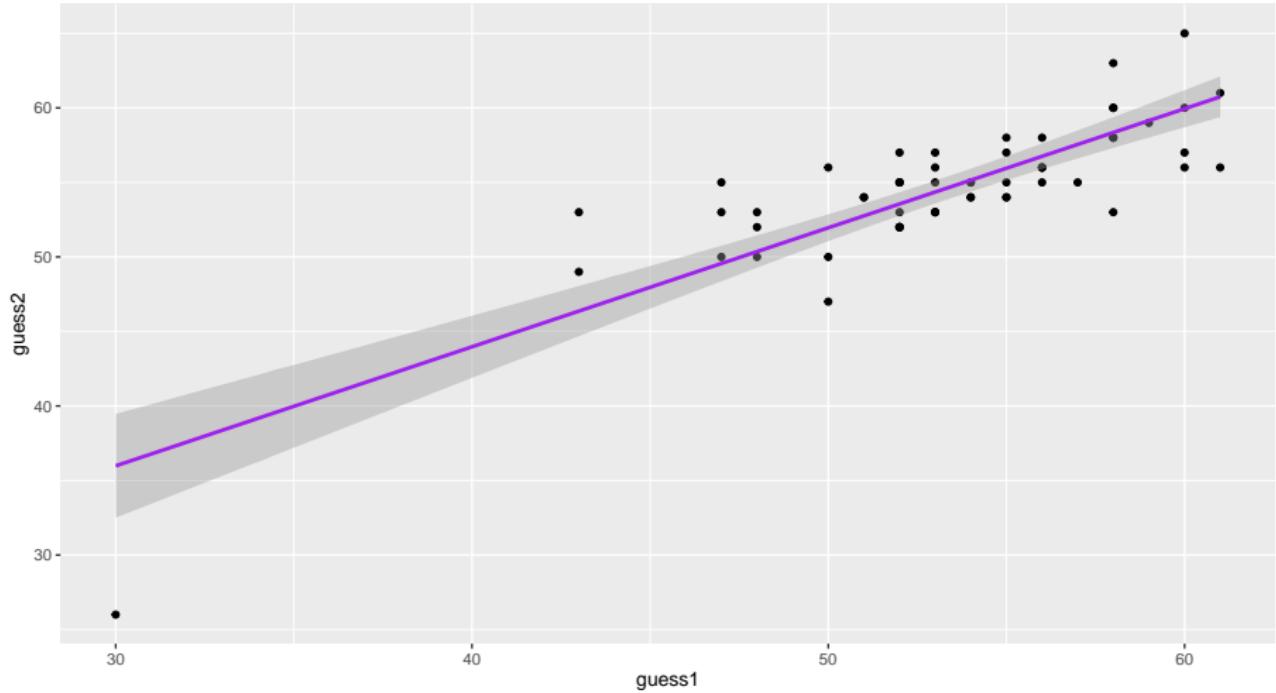
Warning: Removed 2 rows containing missing values
(geom_point).



Filter to complete cases, and add regression line

```
age_guess %>%
  filter(complete.cases(guess1, guess2)) %>%
  ggplot(data = ., aes(x = guess1, y = guess2)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x,
              col = "purple")
```

Resulting Scatterplot



What is that regression line?

```
lm(guess2 ~ guess1, data = age_guess)
```

Call:

```
lm(formula = guess2 ~ guess1, data = age_guess)
```

Coefficients:

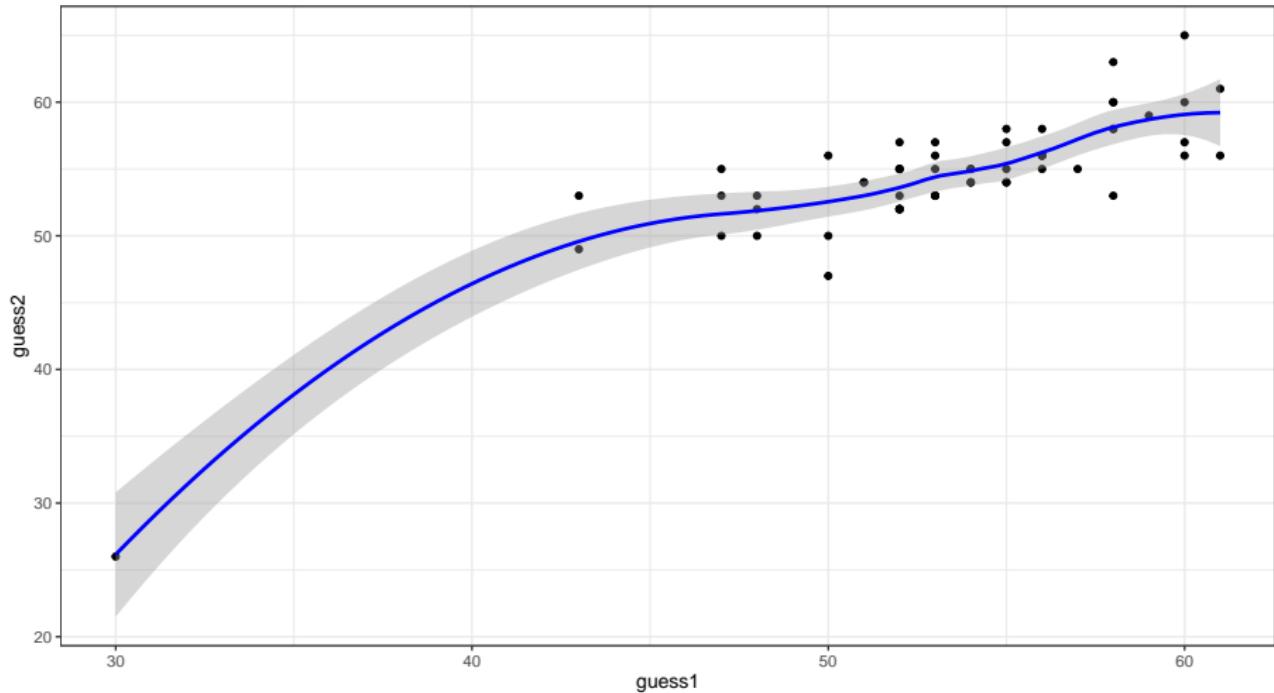
(Intercept)	guess1
12.0219	0.7987

- Note that `lm` filters to complete cases by default.

How about a loess smooth curve, instead?

```
age_guess %>%
  filter(complete.cases(guess1, guess2)) %>%
  ggplot(data = ., aes(x = guess1, y = guess2)) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x,
              col = "blue") +
  theme_bw()
```

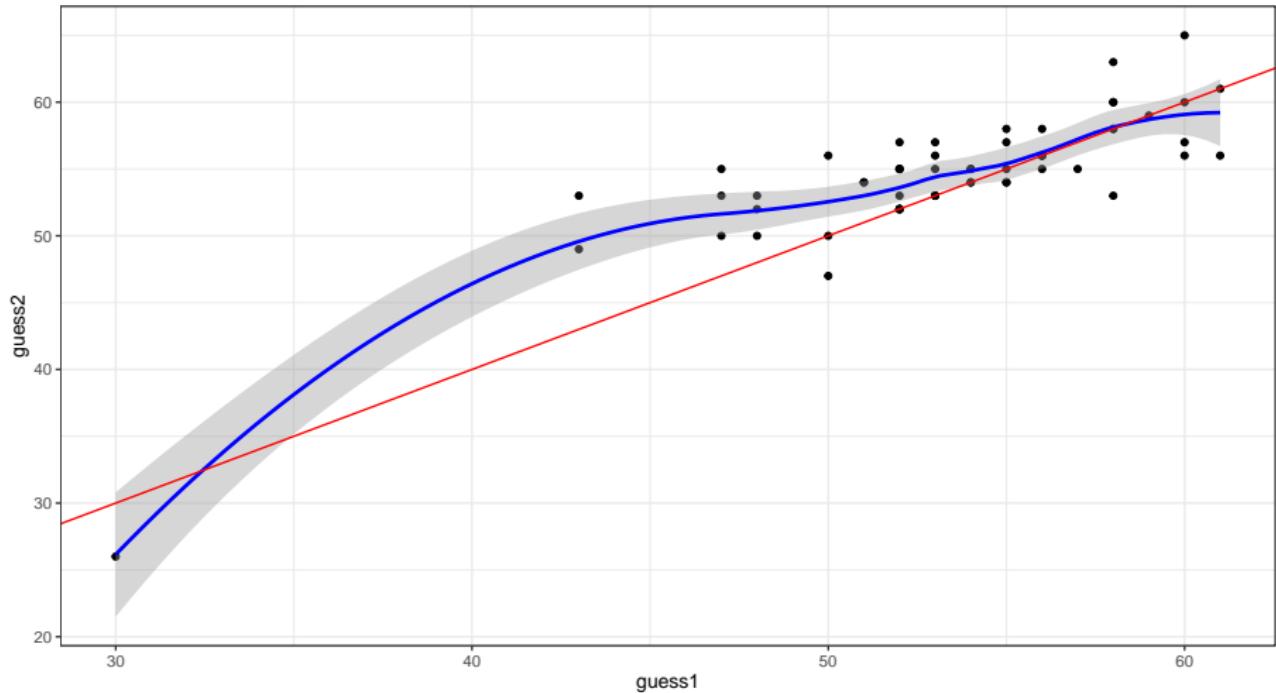
Scatterplot with loess smooth



Add $y = x$ line (no change in guess)?

```
age_guess %>%
  filter(complete.cases(guess1, guess2)) %>%
  ggplot(data = ., aes(x = guess1, y = guess2)) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x,
              col = "blue") +
  geom_abline(intercept = 0, slope = 1, col = "red") +
  theme_bw()
```

Blue smooth and Red line at $y = x$

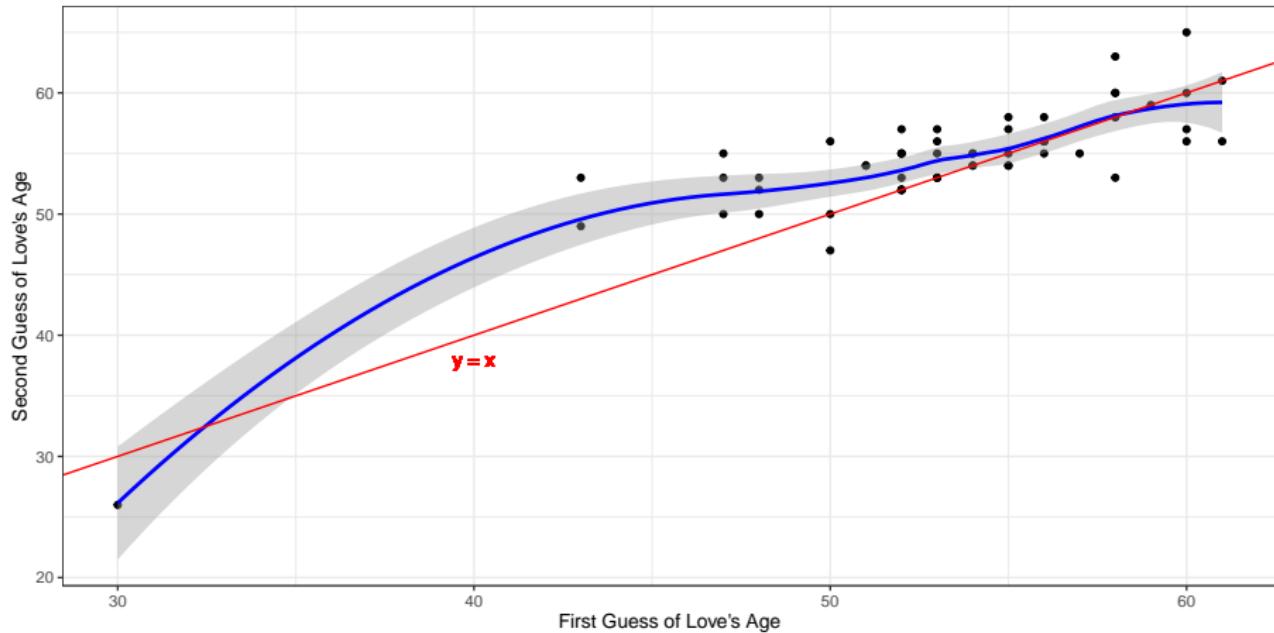


With Better Labels

```
age_guess %>%
  filter(complete.cases(guess1, guess2)) %>%
  ggplot(data = ., aes(x = guess1, y = guess2)) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x,
              col = "blue") +
  geom_abline(intercept = 0, slope = 1, col = "red") +
  geom_text(x = 40, y = 38, label = "y = x", col = "red") +
  labs(x = "First Guess of Love's Age",
       y = "Second Guess of Love's Age",
       title = "Comparing 2021 Age Guesses",
       subtitle = "Love's actual age = 54.5") +
  theme_bw()
```

The Resulting Plot

Comparing 2021 Age Guesses
Love's actual age = 54.5



Decreased / Stayed the Same / Increased

```
age_guess %>% count(sign(guess2 - guess1))
```

```
# A tibble: 4 x 2
`sign(guess2 - guess1)`     n
<dbl> <int>
1                 -1     10
2                  0     19
3                  1     28
4                 NA      2
```

How much did people change their guesses?

```
age_guess <- age_guess %>%
  mutate(change = guess2 - guess1)

summary(age_guess$change)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-5.000	0.000	0.000	1.298	3.000	10.000	
NA's						
2						

Table (via tabyl) of guess changes

```
age_guess %>%
  tabyl(change) %>%
  adorn_pct_formatting()
```

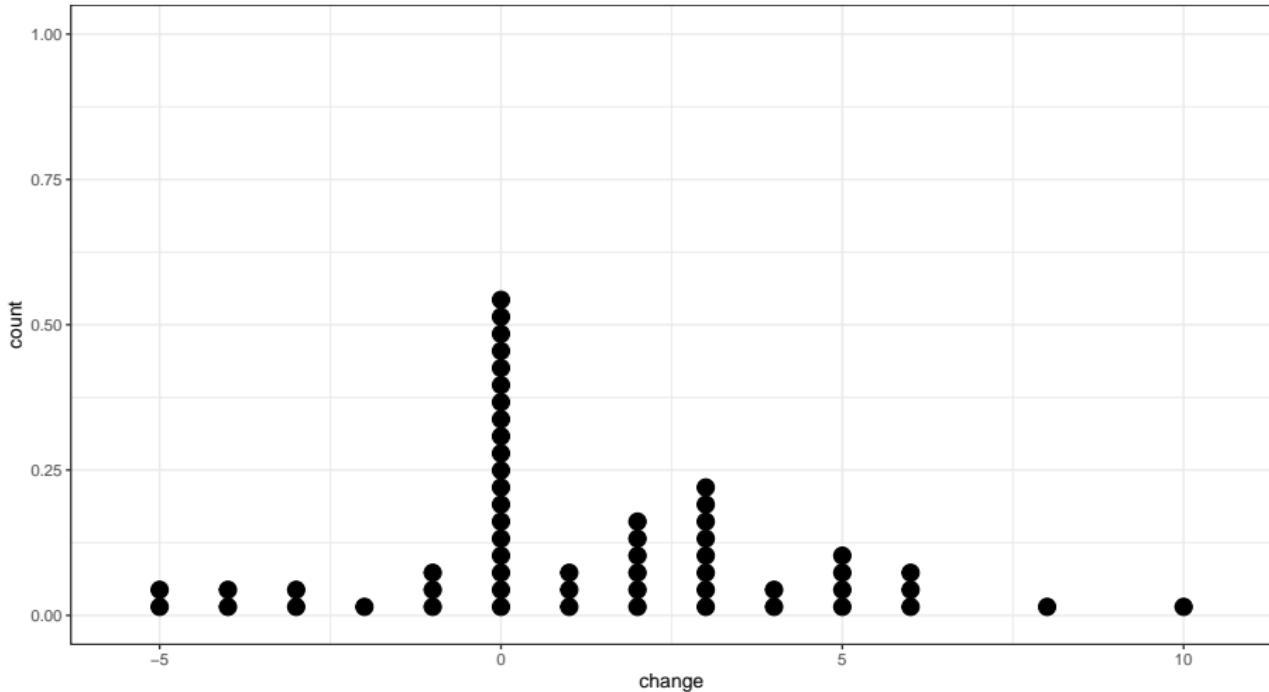
Results on next slide.

Table (via tabyl) of guess changes

change	n	percent	valid_percent
-5	2	3.4%	3.5%
-4	2	3.4%	3.5%
-3	2	3.4%	3.5%
-2	1	1.7%	1.8%
-1	3	5.1%	5.3%
0	19	32.2%	33.3%
1	3	5.1%	5.3%
2	6	10.2%	10.5%
3	8	13.6%	14.0%
4	2	3.4%	3.5%
5	4	6.8%	7.0%
6	3	5.1%	5.3%
8	1	1.7%	1.8%
10	1	1.7%	1.8%
NA	2	3.4%	-

Dotplot of guess changes

```
ggplot(data = age_guess, aes(x = change)) +  
  geom_dotplot(binwidth = 1, dotsizes = 0.25) + theme_bw()
```



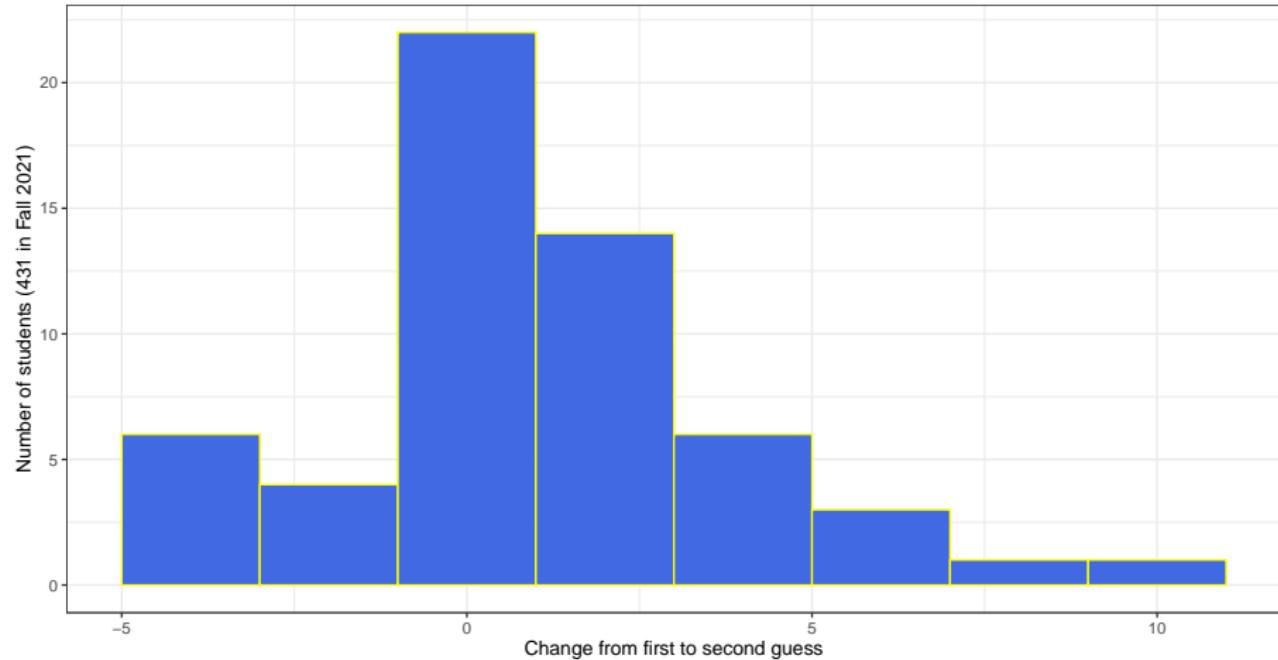
Histogram of guess changes (code)

What will this look like?

```
age_guess %>%
  mutate(change = guess2 - guess1) %>%
  filter(complete.cases(change)) %>%
  ggplot(data = ., aes(x = change)) +
  geom_histogram(binwidth = 2,
                 fill = "royalblue", col = "yellow") +
  theme_bw() +
  labs(x = "Change from first to second guess",
       y = "Number of students (431 in Fall 2021)",
       title = "Most stayed close to their first guess.")
```

Histogram of Changes in Age Guesses

Most stayed close to their first guess.



What Happens Next?

Next Week

Dr. Love will be taking his younger son to college.

- There will be no class session on Tuesday. Instead, Class 03 will involve watching a recording of Dr. Love walking through a series of analyses in R. That recording will be available soon.
- Class 04 (Thursday) will be held at the usual time and place, and the materials for it will be posted soon.

Lab 01 is due on Monday 2021-09-06 at 9 PM, and so you'll need to get started on that as soon as possible.

431 Class 04

Wyatt P. Bensken

2021-09-02

Welcome

Today's Agenda

- Review Box plots, Scatterplots, and Loess smooth curves
 - IMS Chapter 5
- Hands on with R to explore data
- Walk through two figures
- Question(s)

Upcoming Due Dates

- **Lab 01 is due Monday, 2021-09-06, at 9PM**
- As always, see our course website for the most recent course updates
 - thomaselove.github.io/431

A Note

- It's okay if you don't feel completely comfortable with R and building the visualizations we'll work on today!
- The goal of today is just to get hands on with R and some data to start building those foundational coding skills
- Be patient with yourself as you learn and don't be afraid to ask questions

Introduction

Important Visualizations

Box plots

- Summarizes a dataset with 5 statistics, while identifying outliers
 - Median, Interquartile Range (IQR), Range
 - Outliers are generally marked as a point and are generally $1.5 \times$ IQR

Scatterplots

- Used to visualize two numerical variables
- Each point is an observation
- Useful in assessing relationship between variables, and the trend

Loess smooth curves

- We can fit a Loess smooth curve to the data, which can help reveal trends in the data which are not well estimated with a straight line.

The data we'll use

- The ggplot2 package contains the midwest dataset

```
[1] "area"                  "category"
[3] "county"                "inmetro"
[5] "percadultpoverty"      "percamerindan"
[7] "percasiain"             "percbelowpoverty"
[9] "percblack"              "percchildbelowpovert"
[11] "percelderlypoverty"    "perchsd"
[13] "percollege"             "percother"
[15] "percpovertyknown"       "percprof"
[17] "percwhite"              "PID"
[19] "popadults"              "popamerindian"
[21] "popasian"                "popblack"
[23] "popdensity"              "popother"
[25] "poppovertyknown"         "poptotal"
[27] "popwhite"                "state"
```

Working with the data

The variables we are interested in

- In this in-class work we are interested in the following variables:
 - percbelowpoverty, the percent of people below the poverty line
 - percollege, the percent who are college educated
 - county, the county name

The tasks we'll accomplish

- 1 Load and Explore the Data
- 2 Look at Cuyahoga county (where we are now)
- 3 Make a boxplot
- 4 Make a scatterplot
- 5 Add a Loess smooth to our scatterplot

R Markdown (.Rmd) Template

- There is a .Rmd (R Markdown) template available on today's README and and the Data Downloads page
- This is a template which you should download and save somewhere on your computer for today's activity
 - **Please follow the instructions provided, specific to your operating system, to download the template**
- Note: After Lab 01, all of your Labs will be completed using R Markdown. We have provided templates for Lab 02 and Lab 03

Task 1. Load and Explore the Data

Task 1. Load and Explore the Data

Task 1a. Load the data

- You'll first want to load the tidyverse package we'll need, by running the below code

```
library(tidyverse)
```

- Next we'll want to load the midwest data into our environment (this isn't necessary, but makes things a bit more intuitive)

```
midwest <- ggplot2::midwest
```

Task 1b. Learn about the dataset

- By running the below code, we can open up (in our help tab) the documentation for the data

```
?midwest
```

Task 2. Look at Cuyahoga County

Task 2. Look at Cuyahoga County

- To look at Cuyahoga County we'll need to `filter()` our data to just our observation of interest
 - We'll also `select()` only those two variables we'd like to look at
 - This is a good use of the pipe `%>%`
- We know from data documentation that `county` is our variable name, and from looking at the data we can see that the county names are all capitalized.
 - It is important to remember that R is case sensitive!

```
midwest %>%
  filter(county == "CUYAHOGA") %>%
  select(county, percbelowpoverty, percollege)
```

Task 2. Look at Cuyahoga County

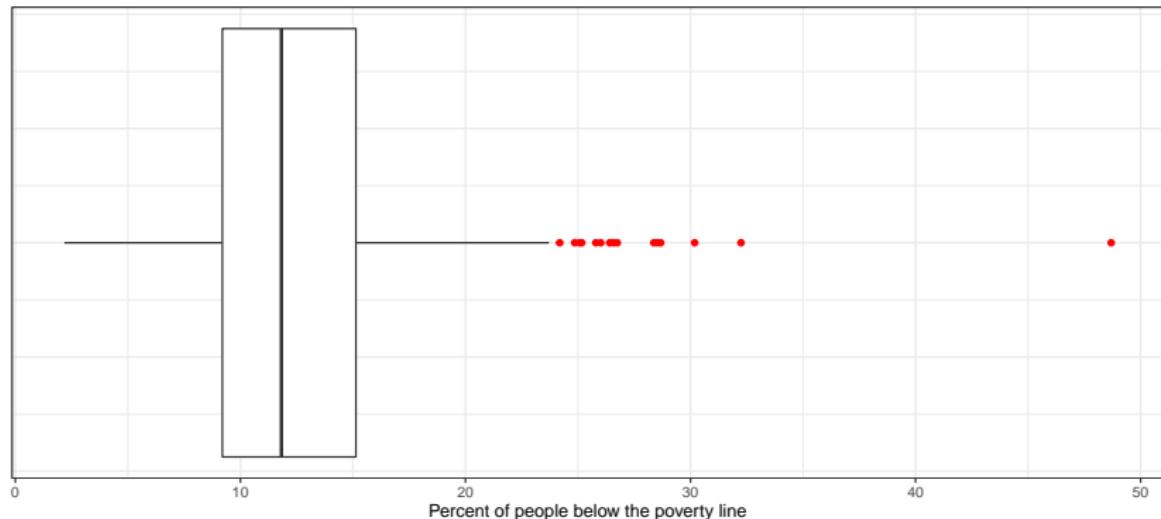
```
# A tibble: 1 x 3
  county    percbelowpoverty percollege
  <chr>          <dbl>        <dbl>
1 CUYAHOGA      13.8        25.1
```

Task 3. Make a boxplot

Task 3. Make a boxplot

- Our goal will be to make a boxplot, of percbelowpoverty which looks like this

Boxplot of poverty in Midwest counties
These data come from the midwest package in ggplot2



Task 3. Make a boxplot

- We'll work through the code step by step, but the complete code looks like this:

```
ggplot(data = midwest, aes(x = percbelowpoverty)) +  
  geom_boxplot(outlier.color = "red") +  
  labs(x = "Percent of people below the poverty line",  
       title = "Boxplot of poverty in Midwest counties",  
       subtitle = "These data come from the  
                  midwest package in ggplot2") +  
  theme_bw() +  
  theme(axis.text.y = element_blank(),  
        axis.ticks.y = element_blank())
```

Task 3. Make a boxplot

Step 1

- First we'll use ggplot to set our dataset and aesthetics (abbreviated "aes")
 - This code won't run anything, until we add our the geom we would like

```
ggplot(data = midwest, aes(x = percbelowpoverty))
```

Task 3. Make a boxplot

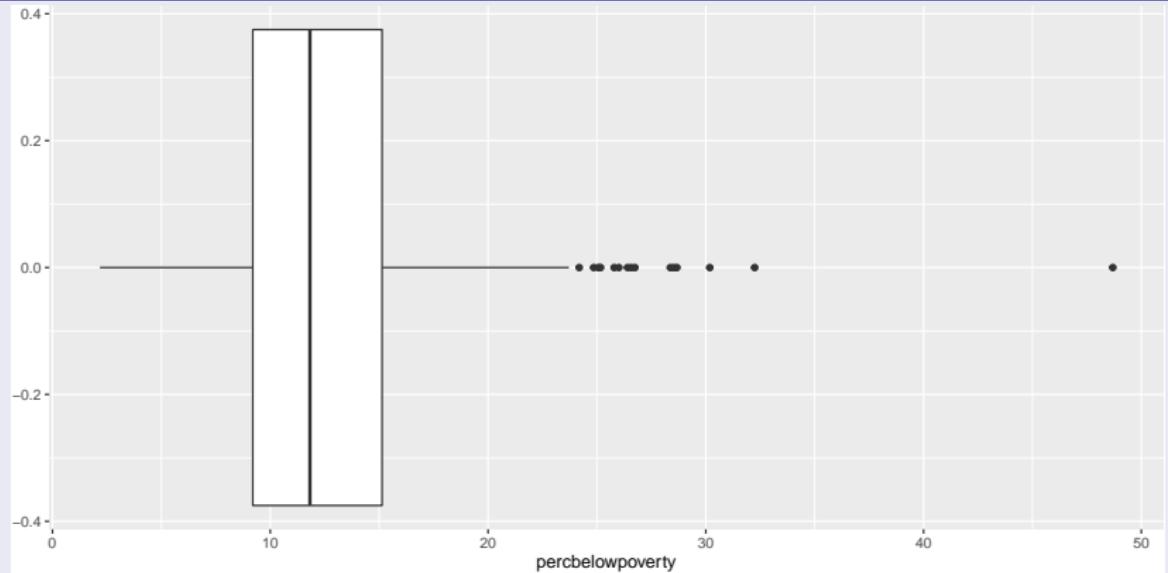
Step 2

- Now we can add (note that we use + here and not the pipe) that we would like the boxplot geom.

```
ggplot(data = midwest, aes(x = percbelowpoverty)) +  
  geom_boxplot()
```

Task 3. Make a boxplot

Step 2



Task 3. Make a boxplot

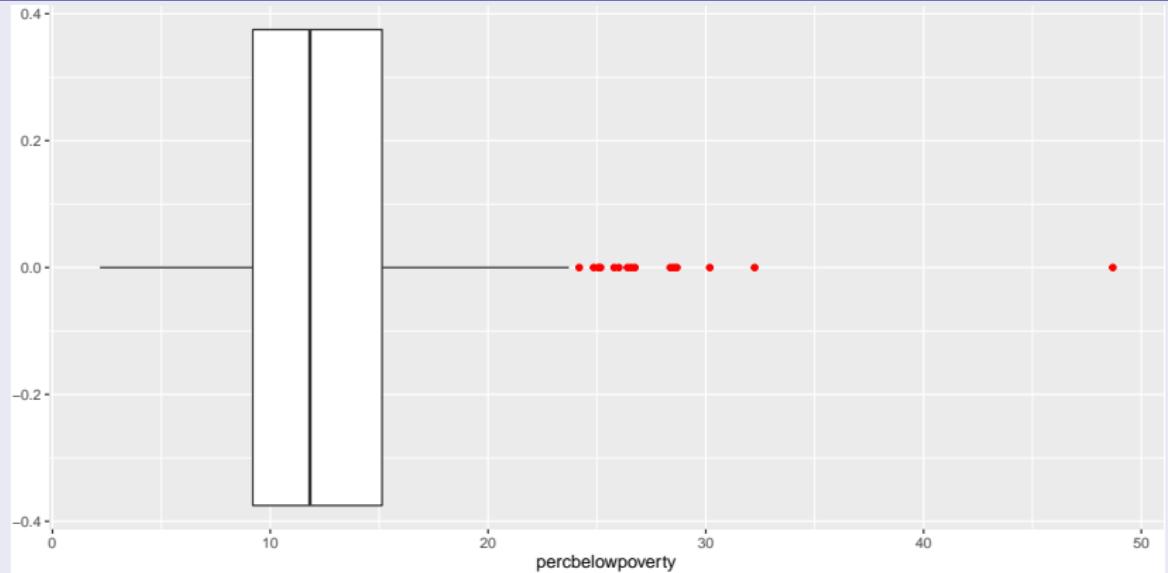
Step 3

- Each geom has a number of options specific to that type of figure, here we'd like to color our outliers red.

```
ggplot(data = midwest, aes(x = percbelowpoverty)) +  
  geom_boxplot(outlier.color = "red")
```

Task 3. Make a boxplot

Step 3



Task 3. Make a boxplot

Step 4

- In this course no figure is complete without appropriate axes labels and titles
- We can add (again use +) these using the labs() statement where we have x, title, and subtitle
 - There are numerous other options such as y, subtitle, and caption, that are available but that we don't use here.

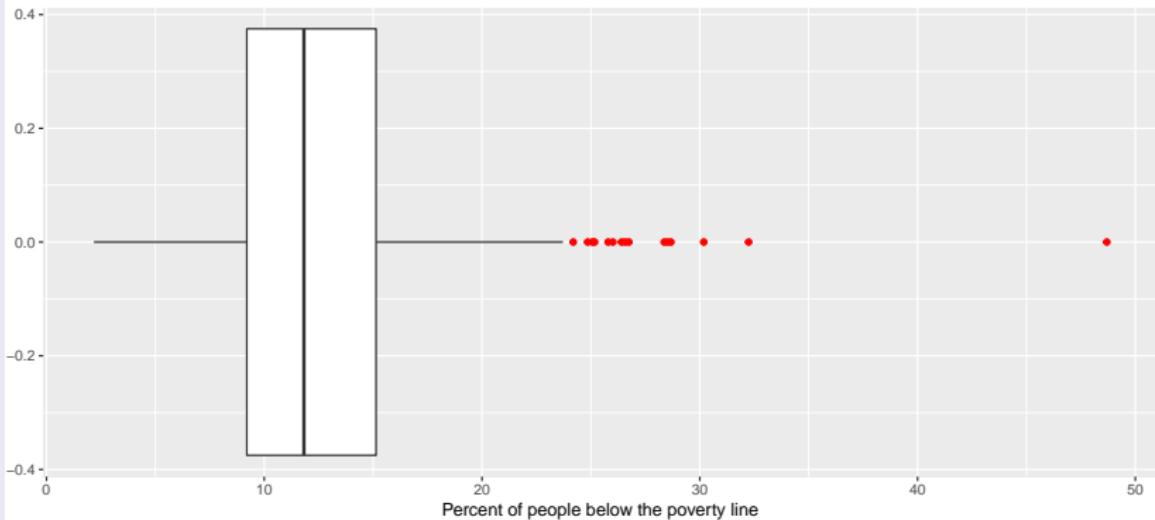
```
ggplot(data = midwest, aes(x = percbelowpoverty)) +
  geom_boxplot(outlier.color = "red") +
  labs(x = "Percent of people below the poverty line",
       title = "Boxplot of poverty in Midwest counties",
       subtitle = "These data come from the
                  midwest package in ggplot2")
```

Task 3. Make a boxplot

Step 4

Boxplot of poverty in Midwest counties

These data come from the midwest package in ggplot2



Task 3. Make a boxplot

Step 5

- I'd like to get rid of that odd gray background which is, somewhat annoyingly, the default
- We can do this using a theme - theme_bw()

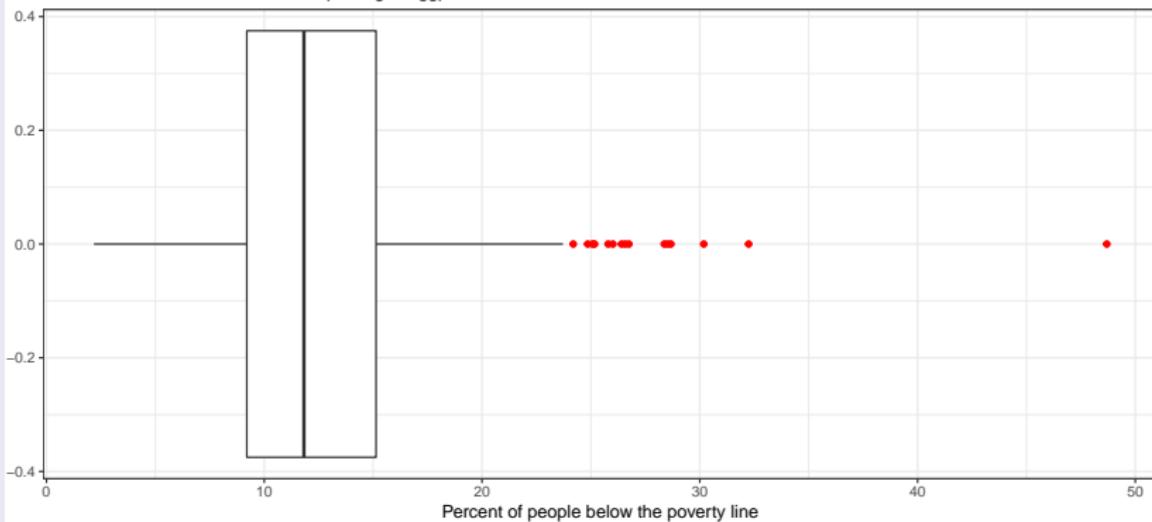
```
ggplot(data = midwest, aes(x = percbelowpoverty)) +  
  geom_boxplot(outlier.color = "red") +  
  labs(x = "Percent of people below the poverty line",  
       title = "Boxplot of poverty in Midwest counties",  
       subtitle = "These data come from the  
                  midwest package in ggplot2") +  
  theme_bw()
```

Task 3. Make a boxplot

Step 5

Boxplot of poverty in Midwest counties

These data come from the midwest package in ggplot2



Task 3. Make a boxplot

Step 6 - the final figure

- Finally, in this boxplot the y-axis text and tick marks are not informative or helpful, so we'd like to remove them
- The theme() command has a whole host of options, but here we'll just use 2.

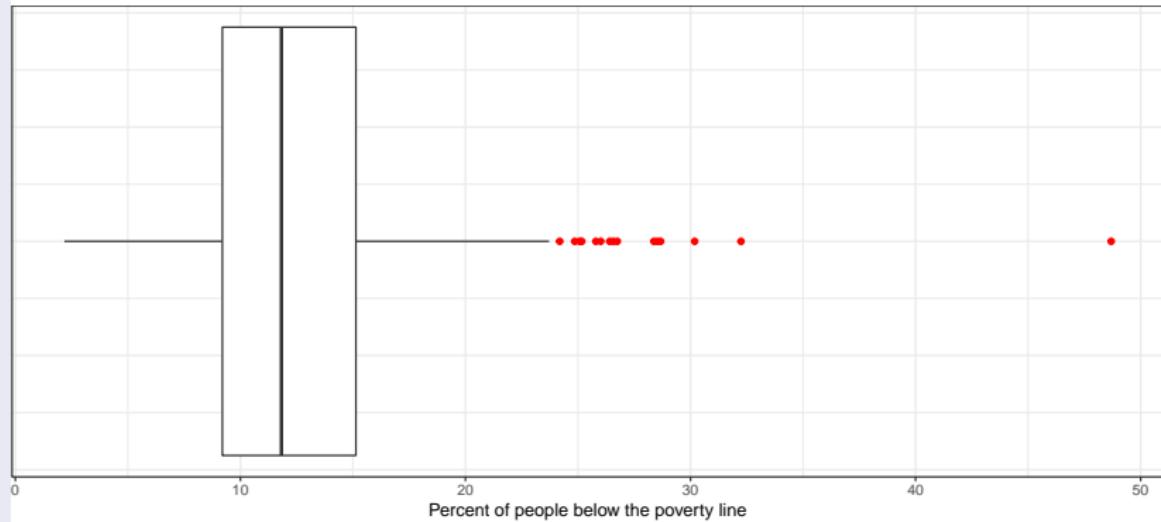
```
ggplot(data = midwest, aes(x = percbelowpoverty)) +  
  geom_boxplot(outlier.color = "red") +  
  labs(x = "Percent of people below the poverty line",  
       title = "Boxplot of poverty in Midwest counties",  
       subtitle = "These data come from the  
                  midwest package in ggplot2") +  
  theme_bw() +  
  theme(axis.text.y = element_blank(),  
        axis.ticks.y = element_blank())
```

Task 3. Make a boxplot

Step 6 - the final figure

Boxplot of poverty in Midwest counties

These data come from the midwest package in ggplot2



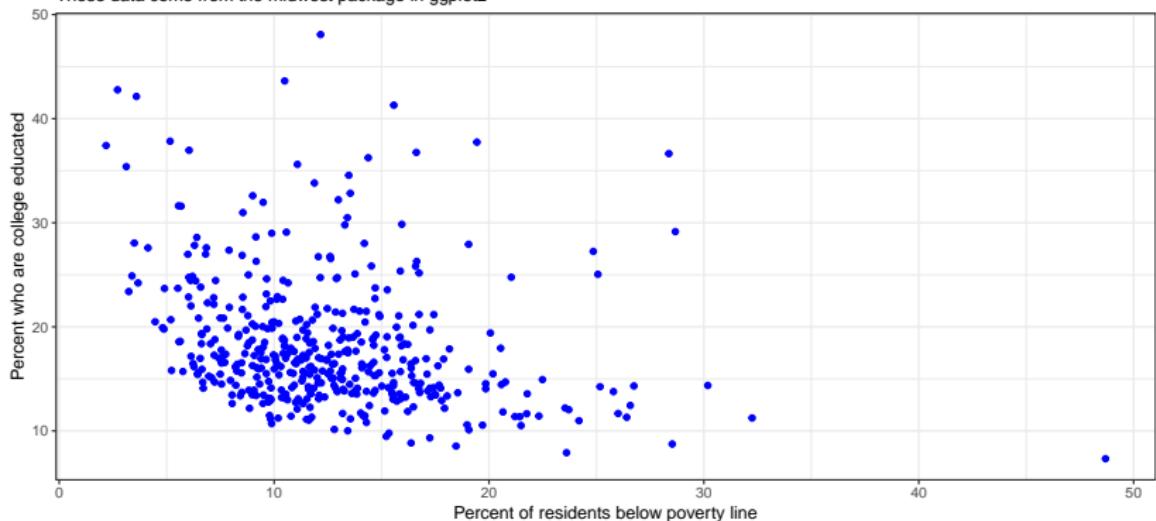
Task 4. Make a scatterplot

Task 4. Make a scatterplot

- Now we'd like to make a scatterplot with `percbelowpoverty` on our x-axis and `percollege` on our y-axis

Relationship between poverty and college education

These data come from the `midwest` package in `ggplot2`



Task 4. Make a scatterplot

Step 1

- Each figure we'll make in R using the `ggplot2` package will share substantial syntax, including the first step
- Here, however we assign not just an x aesthetic but a y aesthetic as well

```
ggplot(data = midwest, aes(x = percbelowpoverty,  
                           y = percollege)) +
```

Task 4. Make a scatterplot

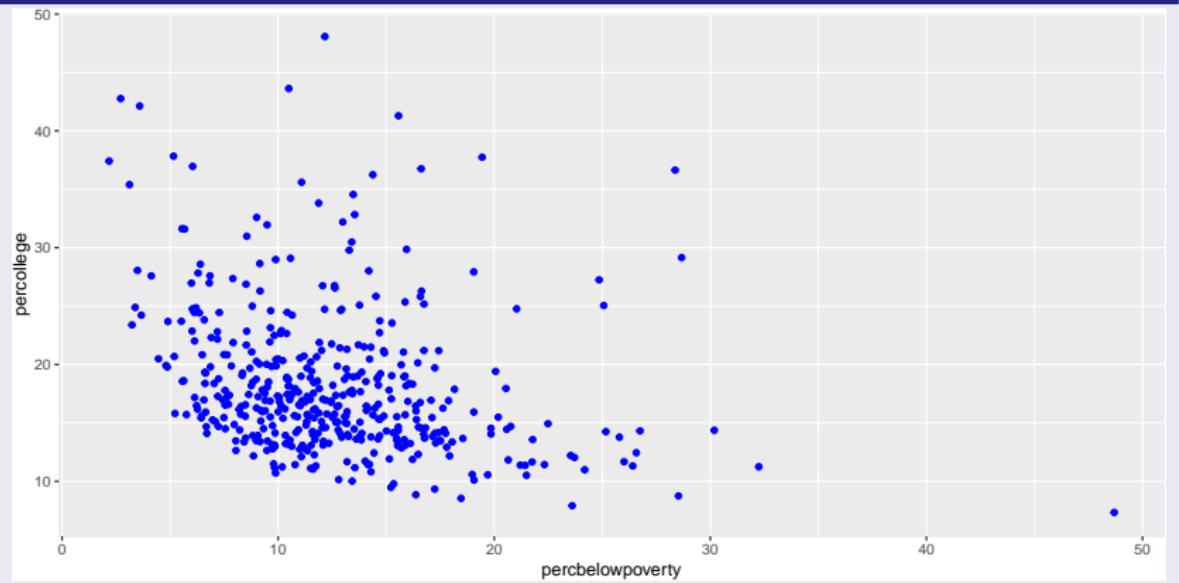
Step 2

- In this example, we want to add a point geom, which makes a scatterplot when we have properly assigned our x and y variables in the aesthetic
- We can again take advantage of the options within our geom to make the points a specific color.

```
ggplot(data = midwest, aes(x = percbelowpoverty,  
                           y = percollege)) +  
  geom_point(color = "blue")
```

Task 4. Make a scatterplot

Step 2



Task 4. Make a scatterplot

Step 3

- As in our boxplot, we **must** add appropriate axis legends

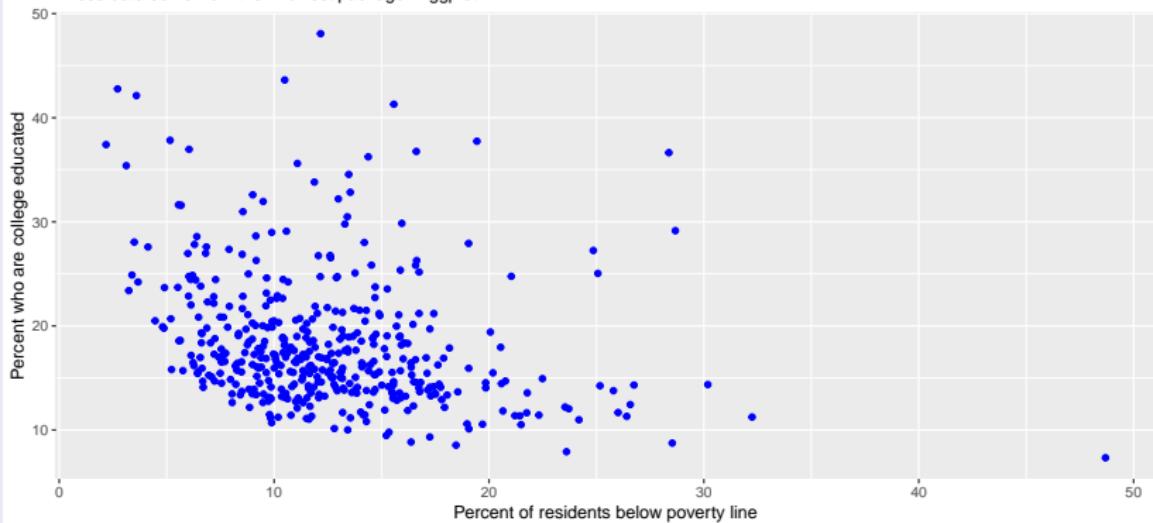
```
ggplot(data = midwest, aes(x = percbelowpoverty,  
                           y = percollege)) +  
  geom_point(color = "blue") +  
  labs(x = "Percent of residents below poverty line",  
       y = "Percent who are college educated",  
       title = "Relationship between poverty  
                 and college education",  
       subtitle = "These data come from the  
                  midwest package in ggplot2")
```

Task 4. Make a scatterplot

Step 3

Relationship between poverty and college education

These data come from the midwest package in ggplot2



Task 4. Make a scatterplot

Step 4 - the final figure

- Finally, we'd like to again use our `theme_bw()` to get rid of that gray background

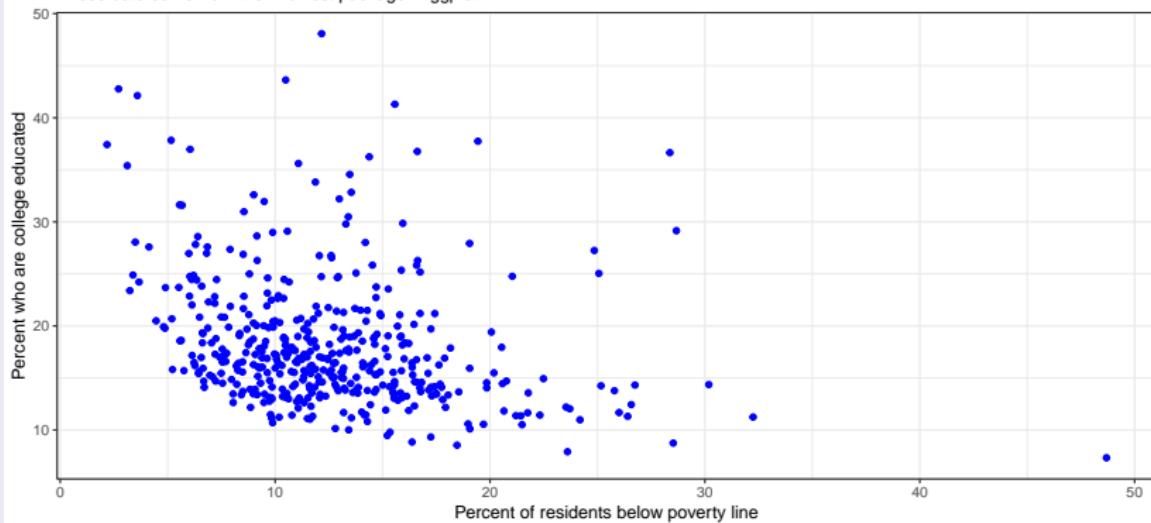
```
ggplot(data = midwest, aes(x = percbelowpoverty,  
                           y = percollege)) +  
  geom_point(color = "blue") +  
  labs(x = "Percent of residents below poverty line",  
       y = "Percent who are college educated",  
       title = "Relationship between poverty  
                 and college education",  
       subtitle = "These data come from the  
                  midwest package in ggplot2") +  
  theme_bw()
```

Task 4. Make a scatterplot

Step 4 - the final figure

Relationship between poverty and college education

These data come from the midwest package in ggplot2



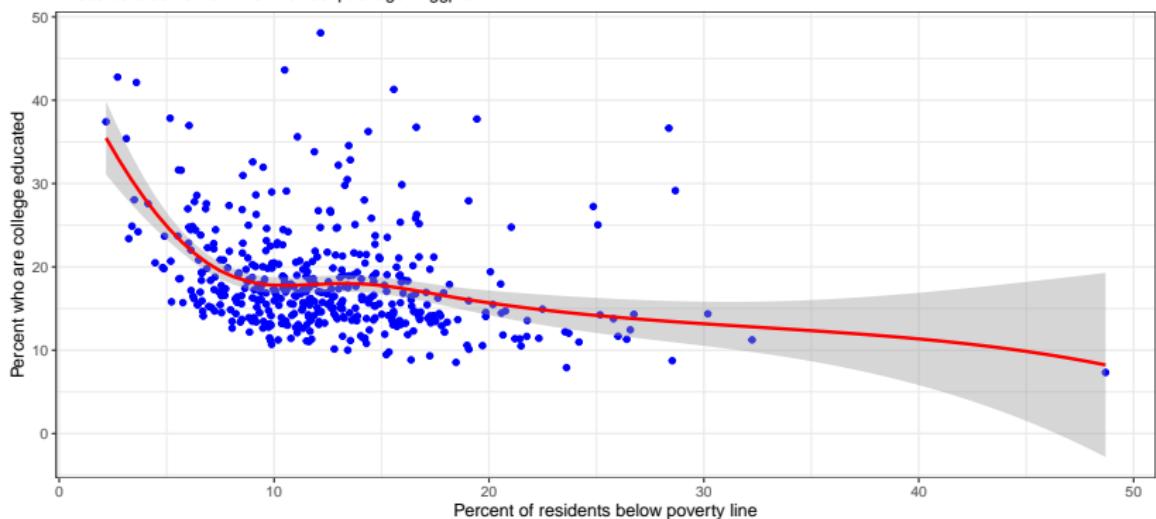
Task 5. Add a Loess smooth

Task 5. Add a Loess smooth

- We'd now like to add a Loess smooth curve to our scatterplot to examine what type of relationship is fit with a smooth line.

Relationship between poverty and college education, with a Loess smooth curve

These data come from the midwest package in ggplot2



Task 5. Add a Loess smooth

Step 1

- One of the most powerful parts of ggplot and R is the ability to layer geoms
- We'll want to make sure to specify that we want a Loess curve in our method

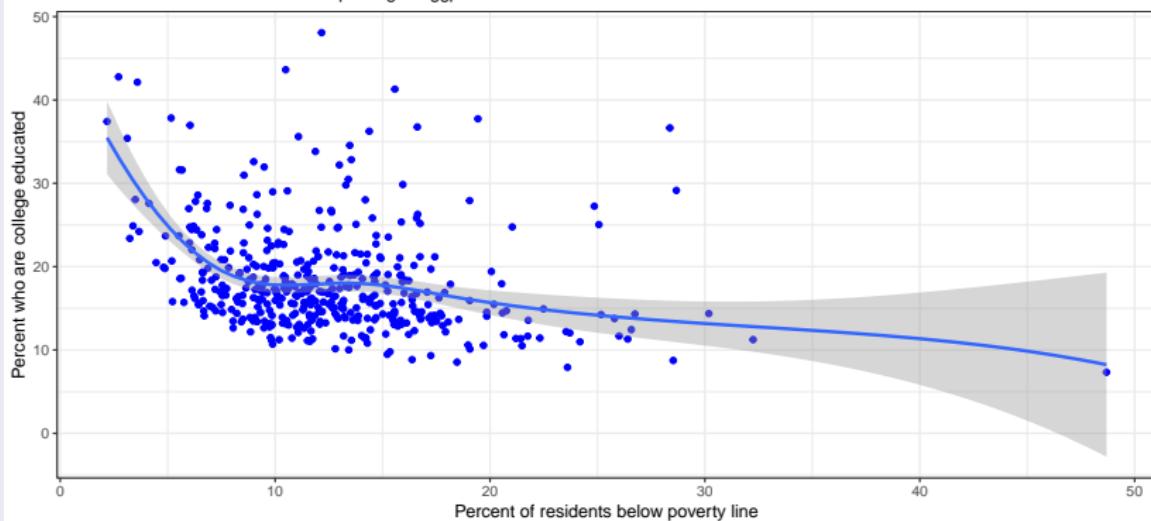
```
ggplot(data = midwest, aes(x = percbelowpoverty,  
                           y = percollege)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "loess", formula = y ~ x) +  
  labs(x = "Percent of residents below poverty line",  
       y = "Percent who are college educated",  
       title = "Relationship between poverty  
               and college education",  
       subtitle = "These data come from the  
               midwest package in ggplot2") +  
  theme_bw()
```

Task 5. Add a Loess smooth

Step 1

Relationship between poverty and college education

These data come from the midwest package in ggplot2



Task 5. Add a Loess smooth

Step 2

- We can easily change the color of our Loess curve, to better differentiate it from the points.

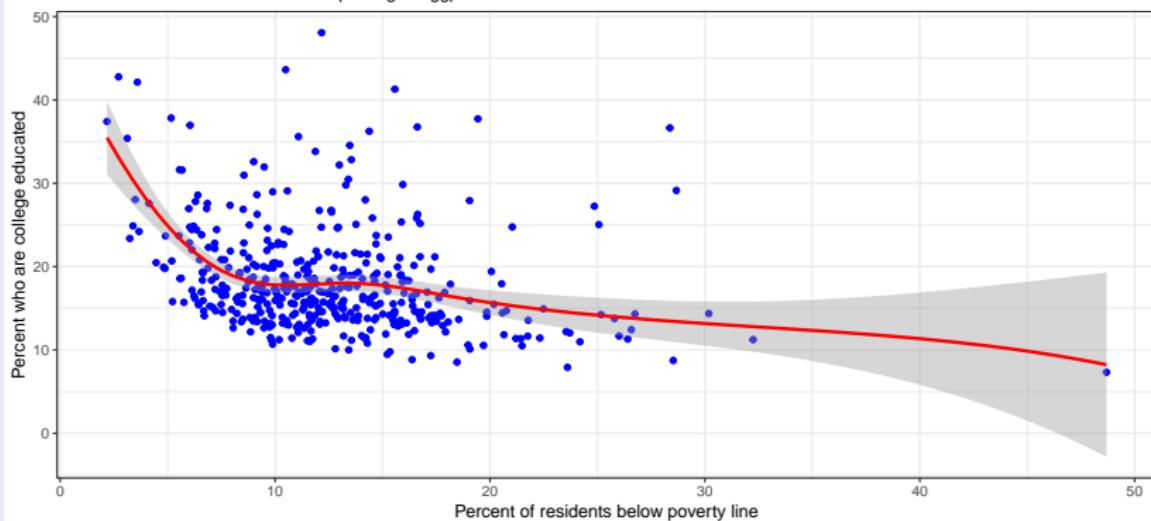
```
ggplot(data = midwest, aes(x = percbelowpoverty,  
                           y = percollege)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "loess", formula = y ~ x,  
              color = "red") +  
  labs(x = "Percent of residents below poverty line",  
       y = "Percent who are college educated",  
       title = "Relationship between poverty  
                and college education",  
       subtitle = "These data come from the  
                  midwest package in ggplot2") +  
  theme_bw()
```

Task 5. Add a Loess smooth

Step 2

Relationship between poverty and college education

These data come from the midwest package in ggplot2



Task 5. Add a Loess smooth

Step 3

- Finally, we'll want to update our title to reflect the new figure

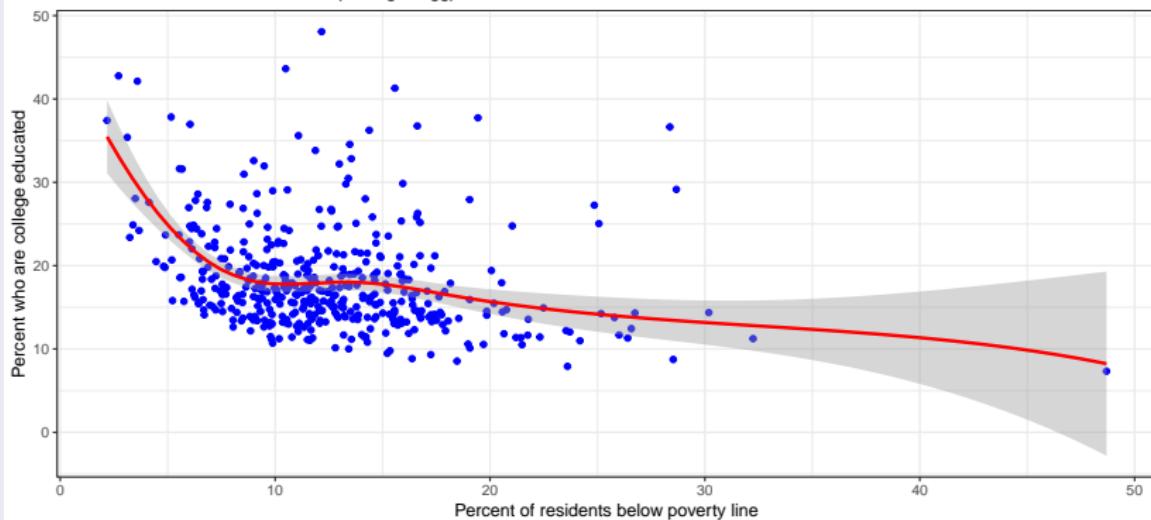
```
ggplot(data = midwest, aes(x = percbelowpoverty,  
                           y = percollege)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "loess", formula = y ~ x,  
              color = "red") +  
  labs(x = "Percent of residents below poverty line",  
       y = "Percent who are college educated",  
       title = "Relationship between poverty  
                and college education,  
                with a Loess smooth curve",  
       subtitle = "These data come from the  
                  midwest package in ggplot2") +  
  theme_bw()
```

Task 5. Add a Loess smooth

Step 3

Relationship between poverty and college education, with a Loess smooth curve

These data come from the midwest package in ggplot2



Knit the file

- At the top of the R Markdown you should see a small button that says “Knit”. Click this.
 - This turns your R Markdown into an HTML file
 - Again, this will be how you will complete Lab 02 - Lab 07

Questions and Discussion

431 Class 05

thomaselove.github.io/431

2021-09-07

You have R Markdown code for all slides

Rmarkdown

TEXT. CODE. OUTPUT.
(GET IT TOGETHER, PEOPLE.)



Today's R Packages

```
library(janitor)
library(knitr)
library(magrittr)
library(naniar) # although today's data are complete
library(patchwork)
library(tidyverse) # always load tidyverse last

theme_set(theme_light()) # other TEL option: theme_bw()
```

- I used {r, message = FALSE} in the code chunk header to suppress some messages about conflicts between R packages.
- By loading tidyverse last, things should work as I expect them to.

Ingesting Today's Data

```
dm431 <- read_csv("data/dm_431.csv",  
                    show_col_types = FALSE)
```

- This is a sample of 431 people from a larger pool of data which we'll study later in the term.
- Note the use of `read_csv` instead of `read.csv` here.
- Updating R to version 4.1.1 and updating your packages may help if you're getting an error
- Can also run this without `show_col_types = FALSE` and you'll just get a message which you can suppress by using `{r, message = FALSE}` as the header of your code chunk.

A First Look

dm431

A tibble: 431 x 17

	CLASS5_ID	AGE	INSURANCE	N_INCOME	HT	WT	SBP
	<chr>	<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	S-001	57	Medicare	22139	1.71	91.2	120
2	S-002	63	Medicaid	39268	1.52	90.6	112
3	S-003	44	Commercial	56837	1.6	89.0	118
4	S-004	56	Uninsured	39962	1.7	88.9	140
5	S-005	38	Medicaid	40228	1.67	116.	156
6	S-006	56	Commercial	43782	1.6	100.	128
7	S-007	50	Medicaid	39574	1.69	80.9	136
8	S-008	49	Medicaid	38676	1.71	106.	120
9	S-009	47	Commercial	71494	1.67	74.2	121
10	S-010	38	Medicaid	11690	1.49	81.8	131

... with 421 more rows, and 10 more variables:

DBP <dbl>, A1C <dbl>, LDL <dbl>, TOBACCO <chr>,

STATIN <dbl>, FVE_EXAM <dbl>

dm431 glimpse at each variable's first few values

```
glimpse(dm431)
```

Rows: 431

Columns: 17

\$ CLASS5_ID	<chr> "S-001", "S-002", "S-003", "S-~
\$ AGE	<dbl> 57, 63, 44, 56, 38, 56, 50, 49~
\$ INSURANCE	<chr> "Medicare", "Medicaid", "Comme~
\$ N_INCOME	<dbl> 22139, 39268, 56837, 39962, 40~
\$ HT	<dbl> 1.71, 1.52, 1.60, 1.70, 1.67, ~
\$ WT	<dbl> 91.22, 90.63, 88.96, 88.91, 11~
\$ SBP	<dbl> 120, 112, 118, 140, 156, 128, ~
\$ DBP	<dbl> 79, 74, 74, 80, 118, 83, 60, 7~
\$ A1C	<dbl> 6.2, 5.9, 8.0, 14.3, 7.8, 6.0,~
\$ LDL	<dbl> 148, 116, 134, 42, 96, 66, 110~
\$ TOBACCO	<chr> "Former", "Never", "Never", "F~
\$ STATIN	<dbl> 1, 0, 1, 1, 1, 1, 1, 1, 1, ~
\$ EYE_EXAM	<dbl> 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, ~

dm431 variable specifications

```
spec(dm431)
```

```
cols(  
  CLASS5_ID = col_character(),  
  AGE = col_double(),  
  INSURANCE = col_character(),  
  N_INCOME = col_double(),  
  HT = col_double(),  
  WT = col_double(),  
  SBP = col_double(),  
  DBP = col_double(),  
  A1C = col_double(),  
  LDL = col_double(),  
  TOBACCO = col_character(),  
  STATIN = col_double(),  
  EYE_EXAM = col_double(),  
  RACE_ETHNICITY = col_character(),
```

What would improve our data ingest?

- Clean up the variable names so that they are lower case (and, if they had any spaces or other problematic characters, replace those with underscores while also de-duplicating)
- Convert the categorical variables like `insurance` we might wind up analyzing from characters to factors
- Keep the `class5_id` variable (subject codes) as a character variable

Re-ingesting Today's Data

```
dm431 <- read_csv("data/dm_431.csv",
                    show_col_types = FALSE) %>%
  clean_names() %>%
  mutate(across(where(is.character), as_factor)) %>%
  mutate(class5_id = as.character(class5_id))
```

The dm431 data: second time around

dm431

```
# A tibble: 431 x 17
```

	class5_id	age	insurance	n_income	ht	wt	sbp
	<chr>	<dbl>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>
1	S-001	57	Medicare	22139	1.71	91.2	120
2	S-002	63	Medicaid	39268	1.52	90.6	112
3	S-003	44	Commercial	56837	1.6	89.0	118
4	S-004	56	Uninsured	39962	1.7	88.9	140
5	S-005	38	Medicaid	40228	1.67	116.	156
6	S-006	56	Commercial	43782	1.6	100.	128
7	S-007	50	Medicaid	39574	1.69	80.9	136
8	S-008	49	Medicaid	38676	1.71	106.	120
9	S-009	47	Commercial	71494	1.67	74.2	121
10	S-010	38	Medicaid	11690	1.49	81.8	131

```
# ... with 421 more rows, and 10 more variables:
```

```
#   dbp <dbl>, a1c <dbl>, ldl <dbl>, tobacco <fct>,
```

```
#   statin <dbl>, euv_exam <dbl>
```

dm431 codebook (part 1)

Simulated data to match Better Health Partnership specs.

Variable	Description
class5_id	subject code (S-001 through S-431)
age	subject's age, in years
insurance	primary insurance, 4 levels
n_income	neighborhood median income, in \$
ht	height, in meters (2 decimal places)
wt	weight, in kilograms (2 decimal places)
sbp	most recent systolic blood pressure (mm Hg)
dbp	most recent diastolic blood pressure (mm Hg)

dm431 codebook (part 2)

Variable	Description
a1c	most recent Hemoglobin A1c (%), with one decimal)
ldl	most recent LDL cholesterol level (mg/dl)
tobacco	most recent tobacco status, 3 levels
statin	1 = prescribed a statin in past 12m, 0 = not
eye_exam	1 = diabetic eye exam in past 12m, 0 = not
race_ethnicity	race/ethnicity category, 3 levels
sex	all subjects turn out to be Female
county	all subjects turn out to be in Cuyahoga County

- This sample includes 431 female adults living with diabetes in Cuyahoga County who are within a certain age range, and who have complete data on all of the variables listed in this codebook.

Checking for missingness

```
miss_case_table(dm431)
```

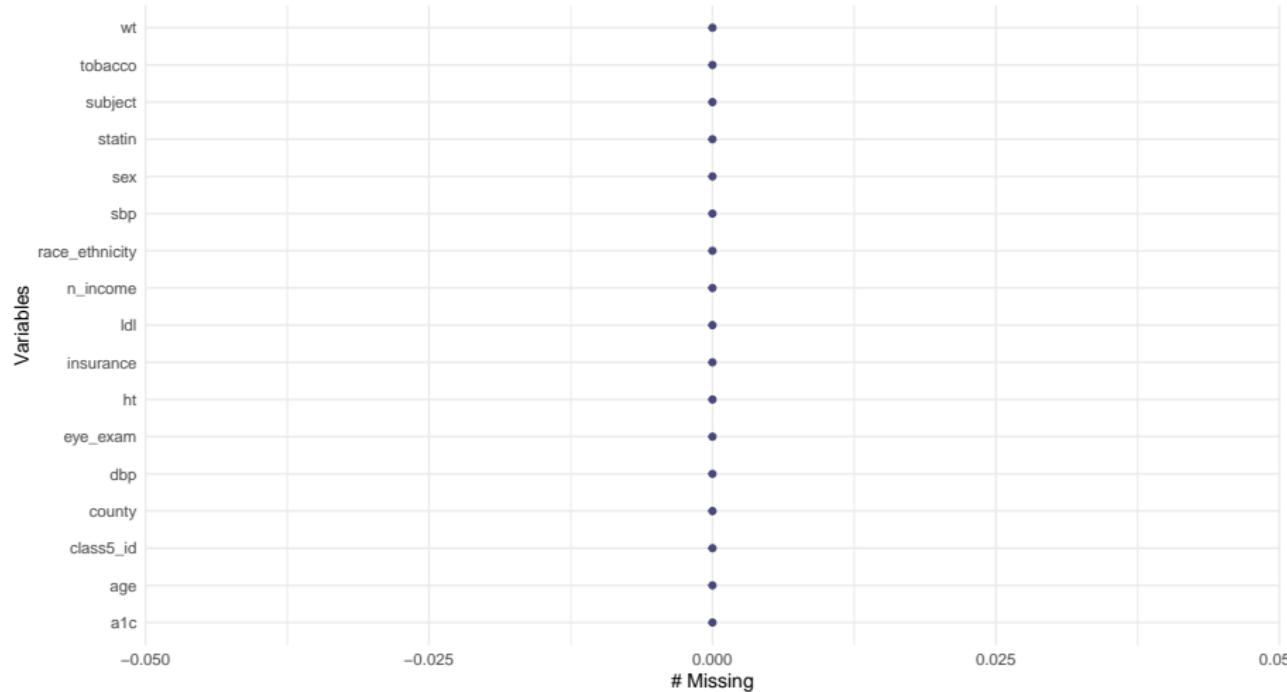
```
# A tibble: 1 x 3
  n_miss_in_case n_cases pct_cases
  <int>     <int>      <dbl>
1          0       431        100
```

Can also use other functions from the `naniar` package to understand and cope with missing values:

- `miss_var_summary()` and `miss_var_table()`
- `gg_miss_var()` as shown on next screen, although that will throw a warning message you can suppress by using `{r, warning = FALSE}` in the chunk header, as I have done.

Plot of missingness in dm431 tibble

```
gg_miss_var(dm431)
```



How old are these women?

- We want to describe the **center**, **spread** (dispersion) and **shape** (symmetry, outliers) of these 431 ages. How do these summaries help?

```
dm431 %$% summary(age)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
30.0	48.0	54.0	52.9	59.0	64.0

```
mosaic::favstats(~ age, data = dm431)
```

min	Q1	median	Q3	max	mean	sd	n	missing
30	48	54	59	64	52.90023	7.993414	431	0

```
mosaic::favstats(~ age, data = dm431) %>% kable(digits = 2)
```

min	Q1	median	Q3	max	mean	sd	n	missing
30	48	54	59	64	52.9	7.99	431	0

Summarizing Age with Hmisc::describe (1/2)

```
dm431 %$% Hmisc::describe(age)
```

age

	n	missing	distinct	Info	Mean	Gmd
431		0	34	0.998	52.9	8.944
.05		.10	.25	.50	.75	.90
38		41	48	54	59	62
.95						
63						

lowest : 30 31 32 33 34, highest: 60 61 62 63 64

- Info is related to how “continuous” the variable is - it’s a relative measure of the available information that is reduced below 1 by having lots of ties or non-distinct values
- Hmisc::describe treats a numeric variable as discrete if it has 10 or fewer distinct values

CONTINUOUS

measured data, can have ∞ values within possible range.



I AM 3.1" TALL
I WEIGH 34.16 grams

DISCRETE

OBSERVATIONS CAN ONLY EXIST AT LIMITED VALUES, OFTEN COUNTS.



I HAVE 8 LEGS
and
4 SPOTS!

@allison_horst

Frank Harrell's Hmisc::describe (2/2)

```
dm431 %$% Hmisc::describe(age)
```

age

	n	missing	distinct	Info	Mean	Gmd
431		0	34	0.998	52.9	8.944
.05		.10	.25	.50	.75	.90
38		41	48	54	59	62
.95						
63						

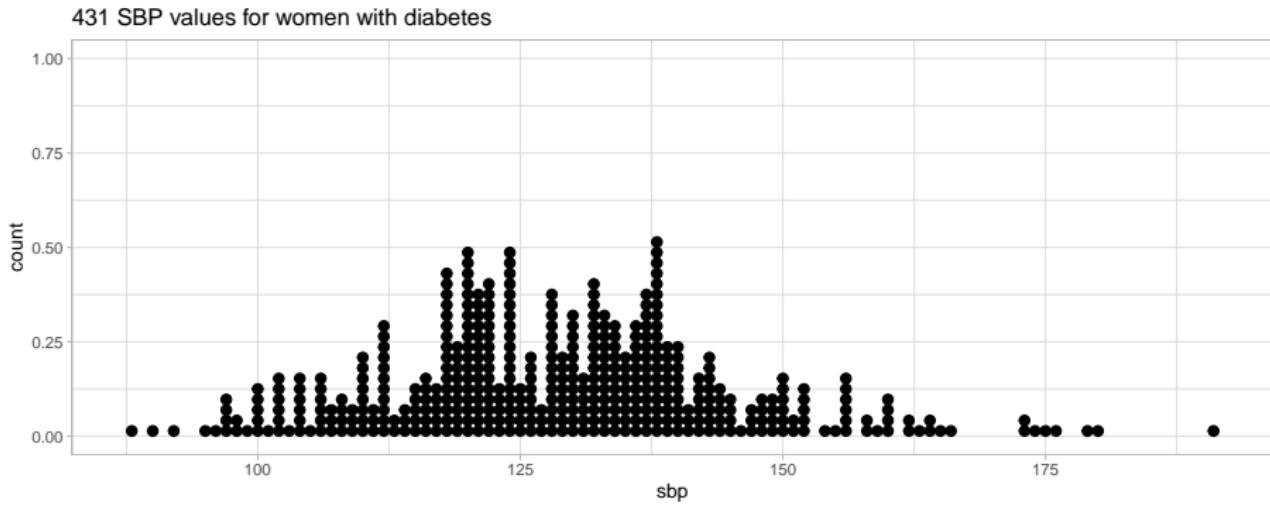
lowest : 30 31 32 33 34, highest: 60 61 62 63 64

- Gmd = Gini's mean difference is a measure of dispersion (spread) that some people like to use rather than a standard deviation in specific settings. It is the mean absolute difference between any pairs of the 431 observations.

**Plotting the sbp data to learn about center,
spread, outliers and shape**

Systolic BP values from dm431 (dotplot)

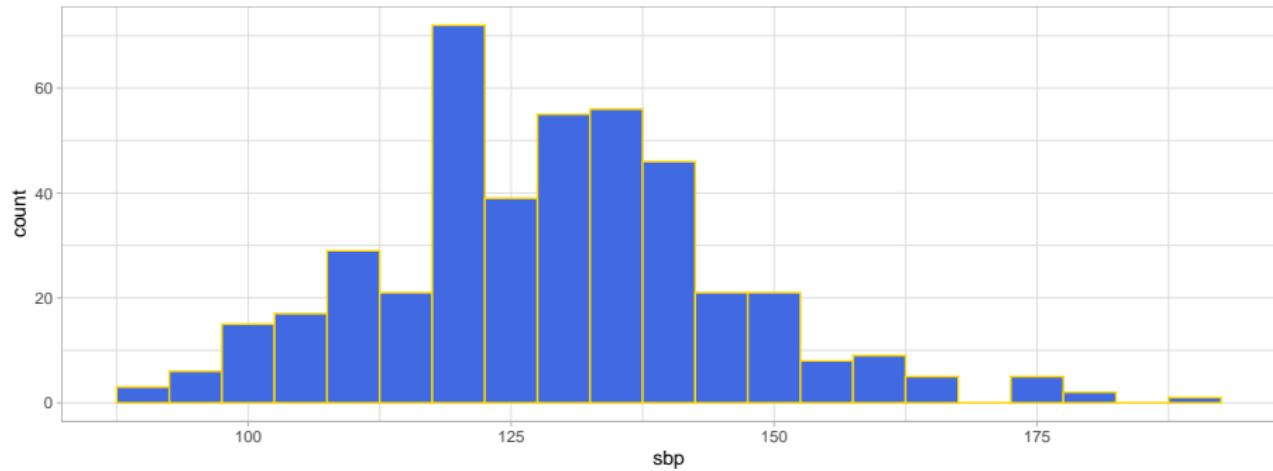
```
ggplot(data = dm431, aes(x = sbp)) +  
  geom_dotplot(binwidth = 1) +  
  labs(title = "431 SBP values for women with diabetes")
```



Systolic BP values from dm431 (histogram)

```
ggplot(data = dm431, aes(x = sbp)) +  
  geom_histogram(binwidth = 5, fill = "royalblue",  
                 col = "gold") +  
  labs(title = "431 SBP values for women with diabetes")
```

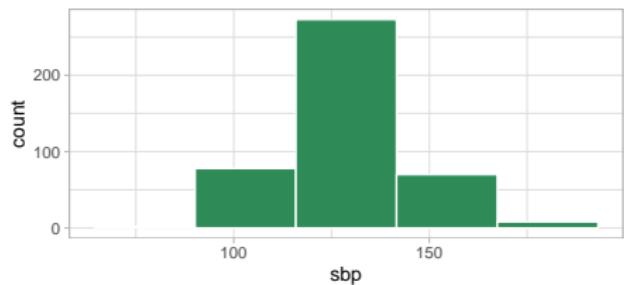
431 SBP values for women with diabetes



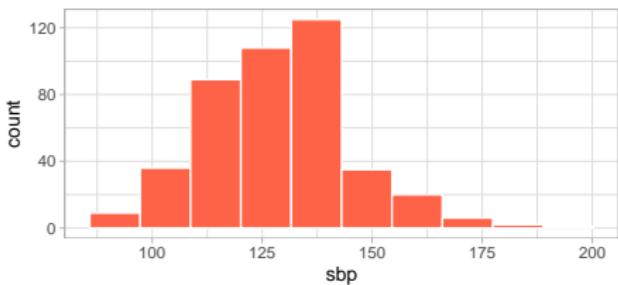
Number of Bins in a Histogram

431 SBP values for women with diabetes

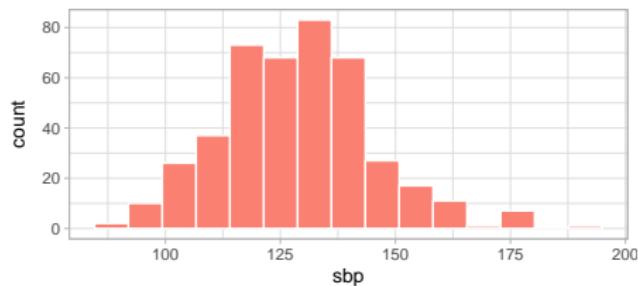
Five bins



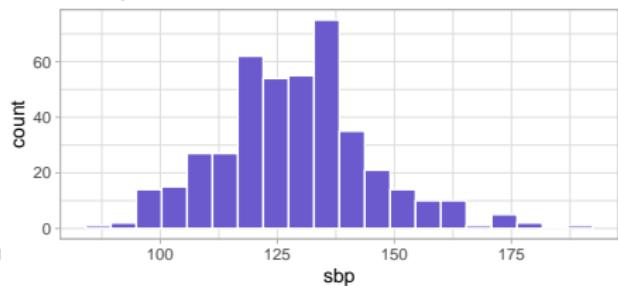
Ten bins



Fifteen bins



Twenty bins



Code for previous slide

```
p1 <- ggplot(data = dm431, aes(x = sbp)) +  
  geom_histogram(bins = 5, fill = "seagreen",  
                 col = "white") +  
  labs(title = "Five bins")  
  
# omitting the code for plots p2-p4 in this slide,  
# use bins = 10, 15 and 20, respectively, and use  
# tomato, salmon and slateblue for fill, respectively  
  
(p1 + p2) / (p3 + p4) +  
  plot_annotation(  
    title = "431 SBP values for women with diabetes")
```

- Remember that you have the R Markdown file for every set of slides.

Can we describe these data as being
well-approximated by a Normal model?

What is a Normal Model?

By a Normal model, we mean that the data are assumed to be the result of selecting at random from a probability distribution called the Normal (or Gaussian) distribution, which is characterized by a bell-shaped curve.

- The Normal model is defined by establishing the values of two parameters: the mean and the standard deviation.

When is it helpful to assume our data follow a Normal model?

- When summarizing the data (especially if we want to interpret the mean and standard deviation)
- When creating inferences about populations from samples (as in a t test, or ANOVA)
- When creating regression models, it will often be important to make distributional assumptions about errors, for instance, that they follow a Normal model.

Does a Normal model fit our data “well enough”?

We evaluate whether a Normal model fits sufficiently well to our data on the basis of (in order of importance):

- ① Graphs (**DTDP**) are the most important tool we have
 - There are several types of graphs available that are designed to (among other things) help us identify clearly several of the potential problems with assuming Normality.
- ② Planned analyses after a Normal model decision is made
 - How serious the problems we see in graphs need to be before we worry about them changes substantially depending on how closely the later analyses we plan to do rely on the assumption of Normality.
- ③ Numerical Summaries are by far the least important even though they seem “easy-to-use” and “objective”.

Simulating Normal data with same Mean and SD

Simulate a sample of 431 observations from a Normal model with mean and standard deviation equal to the mean and standard deviation of our dm431 systolic BPs.

```
set.seed(20210907)
sim_data <- tibble(
  sbp = rnorm(n = 431,
               mean = mean(dm431$sbp),
               sd = sd(dm431$sbp)))
```

Comparing Summary Statistics

```
mosaic::favstats(~ sbp, data = dm431) %>% kable(dig = 1)
```

min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.8	16.3	431	0

```
mosaic::favstats(~ sbp, data = sim_data) %>% kable(dig = 1)
```

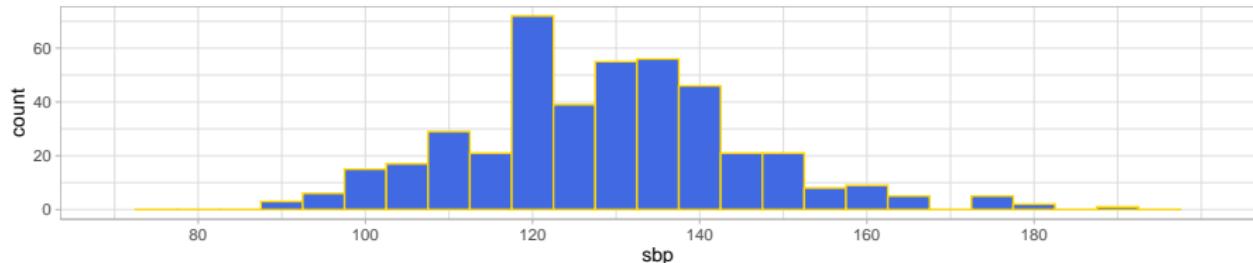
min	Q1	median	Q3	max	mean	sd	n	missing
77.8	117.6	127.9	138.6	175.4	128.2	16.7	431	0

What can we learn from this comparison?

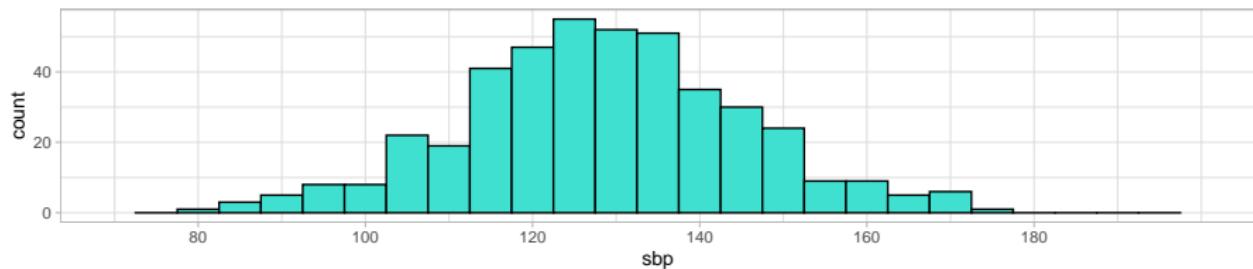
- about the center of the data?
- about the spread of the data?
- about the shape of the data?

Comparing histograms of dm431 and simulated SBP

431 Observed SBP values from dm431 (sample mean = 128.8, sd = 16.3)

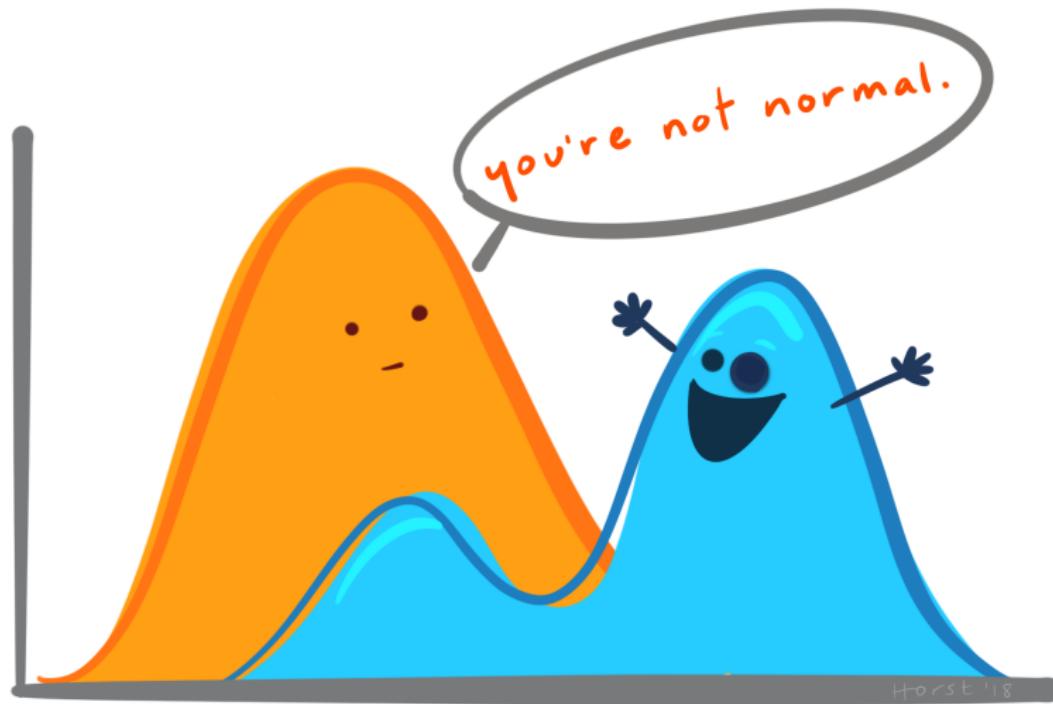


431 Simulated Values from Normal model with mean = 128.8, sd = 16.3



- Does a Normal model look appropriate for describing the SBP in dm431?

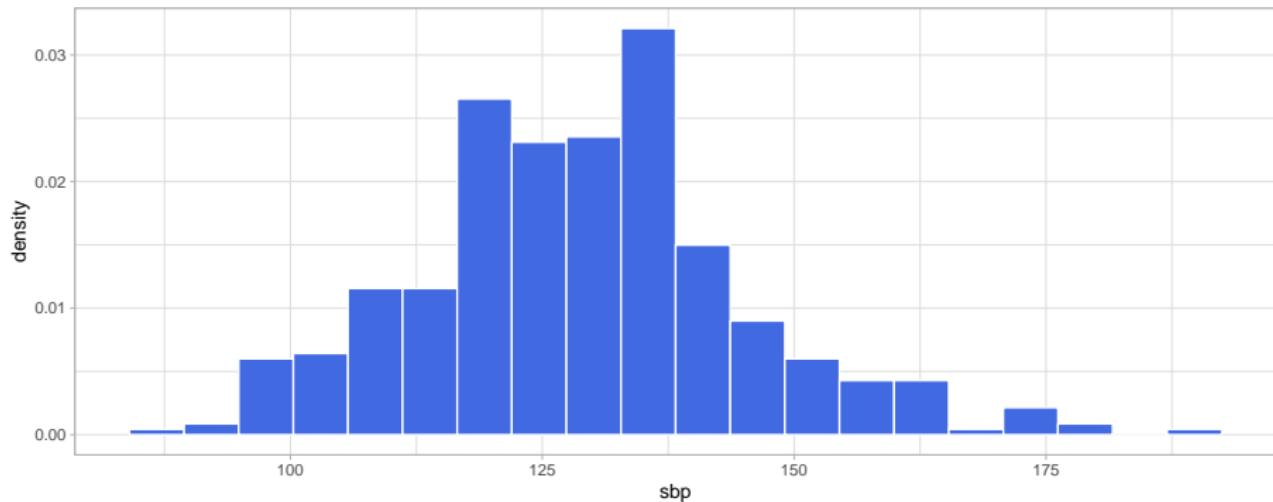
Graphs are our most important tool!



Rescale nh3 SBP histogram as density

Suppose we want to rescale the histogram counts so that the bar areas integrate to 1. This will let us overlay a Normal density onto the results.

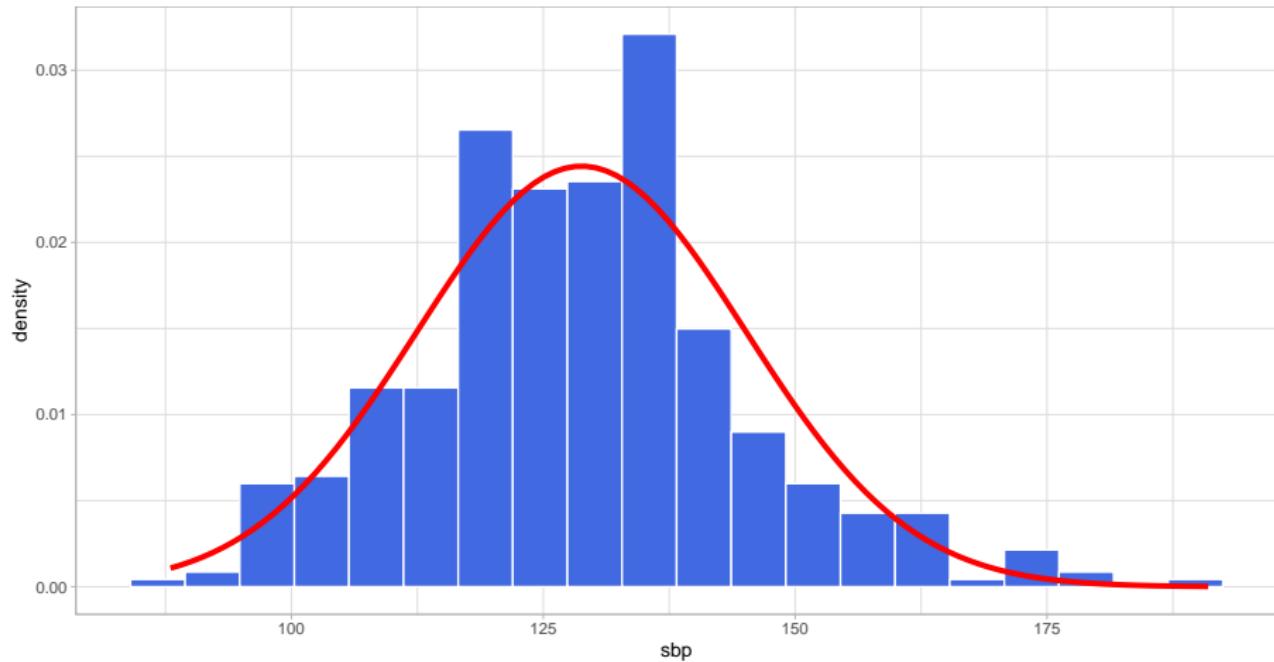
```
ggplot(dm431, aes(x = sbp)) +  
  geom_histogram(aes(y = stat(density)), bins = 20,  
                 fill = "royalblue", col = "white")
```



Density Function, with Normal superimposed

Now we can draw a Normal density curve on top of the rescaled histogram.

SBP density, with Normal model superimposed

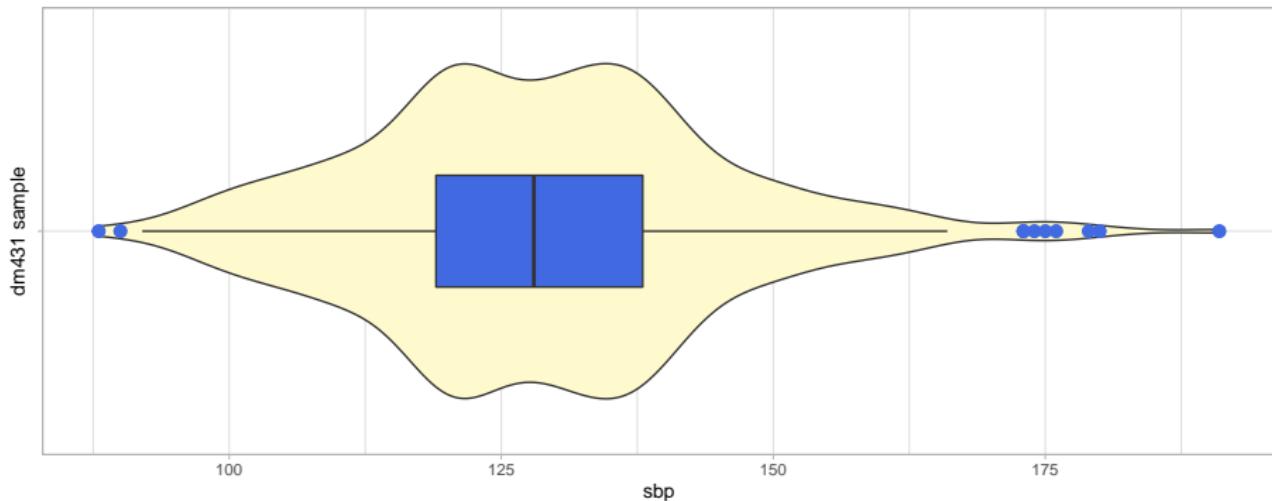


Code for plotting Histogram as Density function

Including the superimposition of a Normal density on top of the histogram.

```
ggplot(dm431, aes(x = sbp)) +  
  geom_histogram(aes(y = stat(density)), bins = 20,  
                 fill = "royalblue", col = "white") +  
  stat_function(fun = dnorm,  
                args = list(mean = mean(dm431$sbp),  
                            sd = sd(dm431$sbp)),  
                col = "red", lwd = 1.5) +  
  labs(title = "SBP density, with Normal model superimposed")
```

Violin and Boxplot for dm431 Systolic BP values

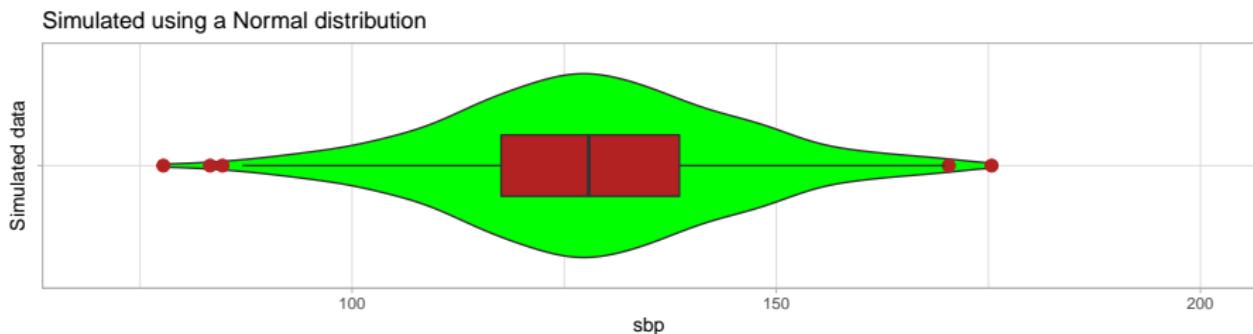
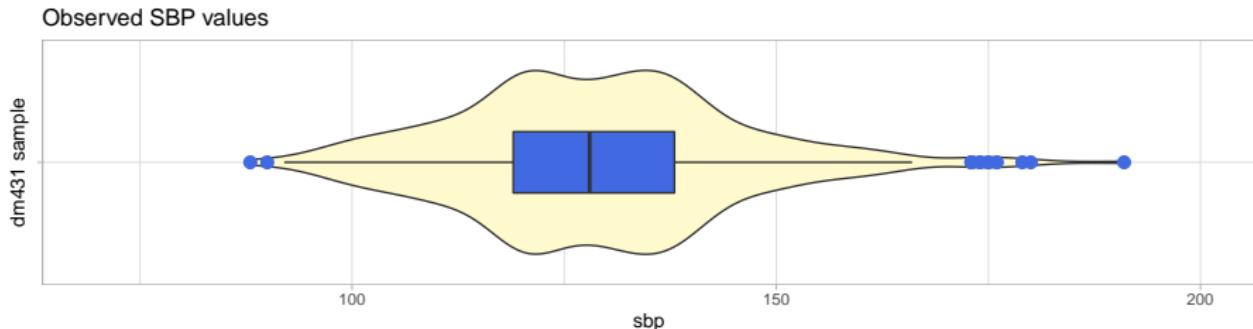


min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.8	16.3	431	0

Code for Previous Slide

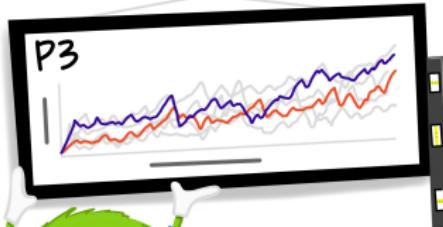
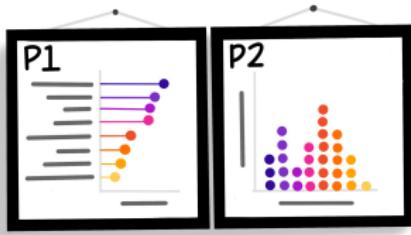
```
ggplot(dm431, aes(x = "", y = sbp)) +  
  geom_violin(fill = "lemonchiffon") +  
  geom_boxplot(width = 0.3, fill = "royalblue",  
               outlier.size = 3,  
               outlier.color = "royalblue") +  
  coord_flip() +  
  labs(x = "dm431 sample")  
  
mosaic::favstats(~ sbp, data = dm431) %>%  
  kable(digits = 1)
```

Observed vs. Simulated Systolic BPs



Does a Normal model look appropriate for describing the SBP in dm431?

Putting the plots together...



Using Numerical Summaries to Assess Normality: A Good Idea?

Does a Normal model fit well for my data?

The least important approach (even though it is seemingly the most objective) is the calculation of various numerical summaries.

Semi-useful summaries help us understand whether they match up well with the expectations of a normal model:

- ① Assessing skewness with $skew_1$ (is the mean close to the median?)
- ② Assessing coverage probabilities (do they match the Normal model?)

Quantifying skew with $skew_1$

$$skew_1 = \frac{mean - median}{standard\ deviation}$$

Interpreting $skew_1$ (for unimodal data)

- $skew_1 = 0$ if the mean and median are the same
- $skew_1 > 0.2$ indicates fairly substantial right skew
- $skew_1 < -0.2$ indicates fairly substantial left skew

Measuring skewness in the SBP values: dm431?

```
mosaic::favstats(~ sbp, data = dm431)
```

min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.7889	16.33058	431	0

```
dm431 %>%
  summarize(skew1 = (mean(sbp) - median(sbp))/sd(sbp))
```

```
# A tibble: 1 x 1
  skew1
  <dbl>
1 0.0483
```

What does this suggest?

Empirical Rule for a Normal Model

If the data followed a Normal distribution, perfectly, then about:

- 68% of the data would fall within 1 standard deviation of the mean
- 95% of the data would fall within 2 standard deviations of the mean
- 99.7% of the data would fall within 3 standard deviations of the mean

Remember that, regardless of the distribution of the data:

- Half of the data will fall below the median, and half above it.
- Half of the data will fall in the Interquartile Range (IQR).

How many SBPs are within 1 SD of the mean?

```
dm431 %>%
  count(sbp > mean(sbp) - sd(sbp),
        sbp < mean(sbp) + sd(sbp)) %>%
  kable()
```

sbp > mean(sbp) - sd(sbp)	sbp < mean(sbp) + sd(sbp)	n
FALSE	TRUE	70
TRUE	FALSE	55
TRUE	TRUE	306

- Note that $306/431 = 0.71$, approximately.
- How does this compare to the expectation under a Normal model?

SBP and the mean \pm 2 standard deviations rule?

The total sample size here is 431

```
dm431 %>%
  count(sbp > mean(sbp) - 2*sd(sbp),
        sbp < mean(sbp) + 2*sd(sbp)) %>%
  kable()
```

sbp > mean(sbp) - 2 * sd(sbp)	sbp < mean(sbp) + 2 * sd(sbp)	n
FALSE	TRUE	5
TRUE	FALSE	15
TRUE	TRUE	411

- Note that $411/431 = 0.95$, approximately.
- How does this compare to the expectation under a Normal model?

Should we use hypothesis tests to assess Normality?

Hypothesis Testing to assess Normality

Don't. Graphical approaches are **far** better than hypothesis tests...

```
dm431 %$% shapiro.test(sbp)
```

Shapiro-Wilk normality test

```
data: sbp  
W = 0.98636, p-value = 0.0004525
```

The very small p value indicates that the test finds some indications **against** adopting a Normal model for these data.

- Exciting, huh? But not actually all that useful, alas.

Why not test for Normality?

There are multiple hypothesis testing schemes (Kolmogorov-Smirnov, etc.) and each looks for one specific violation of a Normality assumption. None can capture the wide range of issues our brains can envision, and none by itself is great at its job.

- With any sort of reasonable sample size, the test is so poor at detecting non-normality compared to our eyes, that it finds problems we don't care about and ignores problems we do care about.
- And without a reasonable sample size, the test is essentially useless.

Whenever you *can* avoid hypothesis testing and instead actually plot the data, you should plot the data. Sometimes you can't plot (especially with really big data) but the test should be your very last resort.

Next Time

That's it for today. Coming Up Next...

- Please complete the **Minute Paper** by noon Wednesday.
- Next time, we'll discuss several things, including...
 - Normal Q-Q plots
 - Building confidence intervals for numerical summaries of a single batch of quantitative data
 - Comparing distributions from two batches of quantitative data

431 Class 06

thomaselove.github.io/431

2021-09-09

Today's R Packages

```
library(janitor)
library(knitr)
library(magrittr)
library(patchwork)
library(tidyverse) # always load tidyverse last

theme_set(theme_light()) # other TEL option: theme_bw()
```

- As mentioned in Class 5, I used {r, message = FALSE} in the code chunk header to suppress messages about conflicts between R packages.

Ingesting Today's Data

```
dm431 <- read_csv("data/dm_431.csv",
                    show_col_types = FALSE) %>%
  clean_names() %>%
  mutate(across(where(is.character), as_factor)) %>%
  mutate(class5_id = as.character(class5_id))
```

- This is the same approach we wound up with in Class 5.

dm431 codebook (part 1)

- This sample includes 431 female adults living with diabetes in Cuyahoga County at ages 30-64, and who have complete data on all of the variables listed in this codebook.

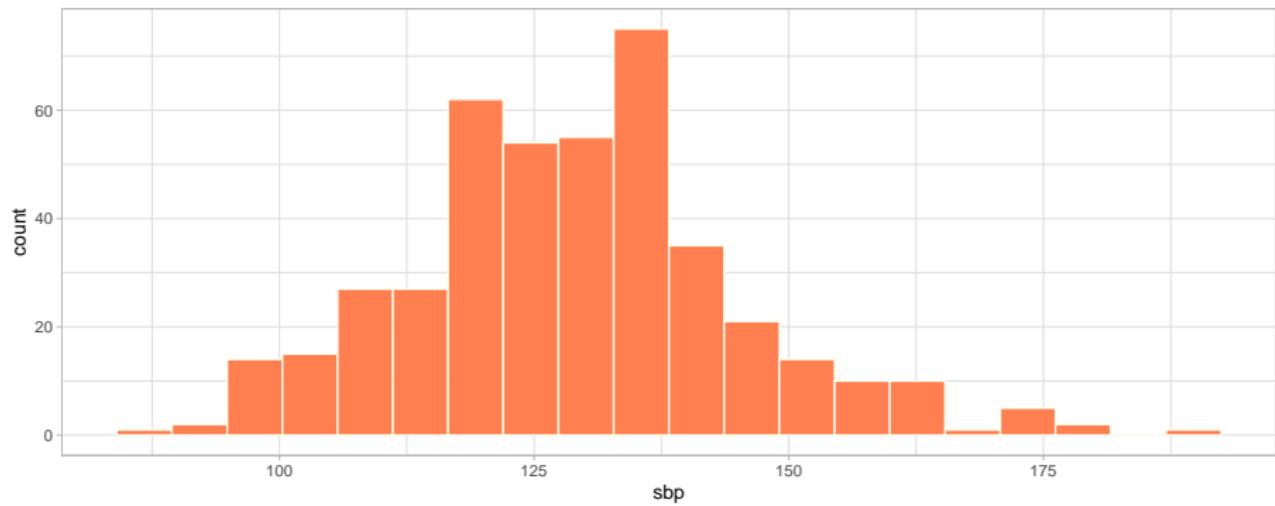
Variable	Description
class5_id	subject code (S-001 through S-431)
age	subject's age, in years
sbp	most recent systolic blood pressure (mm Hg)
dbp	most recent diastolic blood pressure (mm Hg)
n_income	neighborhood median income, in \$
ldl	most recent LDL cholesterol level (mg/dl)
a1c	most recent Hemoglobin A1c (%, with one decimal)
insurance	primary insurance, 4 levels
statin	1 = prescribed a statin in past 12m, 0 = not

Remainder of dm431 codebook (part 2)

Variable	Description
ht	height, in meters (2 decimal places)
wt	weight, in kilograms (2 decimal places)
tobacco	most recent tobacco status, 3 levels
eye_exam	1 = diabetic eye exam in past 12m, 0 = not
race_ethnicity	race/ethnicity category, 3 levels
sex	all subjects turn out to be Female
county	all subjects turn out to be in Cuyahoga County

Histogram of 431 sbp values

```
ggplot(data = dm431, aes(x = sbp)) +  
  geom_histogram(bins = 20, fill = "coral", col = "ivory") +  
  labs("Systolic Blood Pressure for 431 women with diabetes")
```



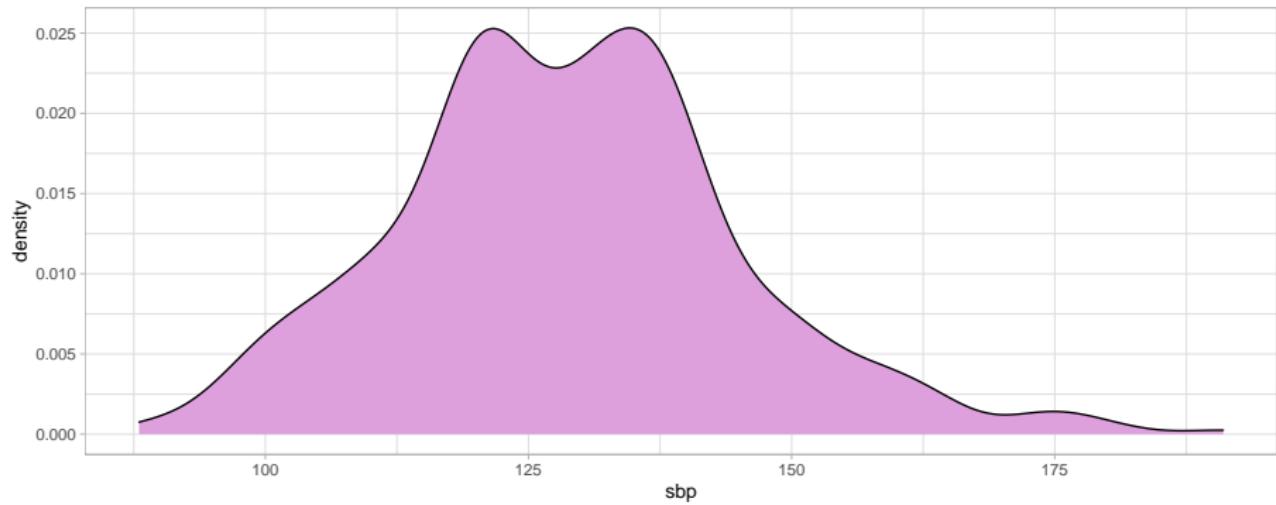
Does a Normal model fit our data “well enough”?

We evaluate whether a Normal model fits sufficiently well to our data on the basis of (in order of importance):

- ① Graphs (**DTDP**) are the most important tool we have
 - There are several types of graphs available that are designed to (among other things) help us identify clearly several of the potential problems with assuming Normality.
- ② Planned analyses after a Normal model decision is made
 - How serious the problems we see in graphs need to be before we worry about them changes substantially depending on how closely the later analyses we plan to do rely on the assumption of Normality.
- ③ Numerical Summaries are by far the least important even though they seem “easy-to-use” and “objective”.

Density Plot of the 431 sbp values

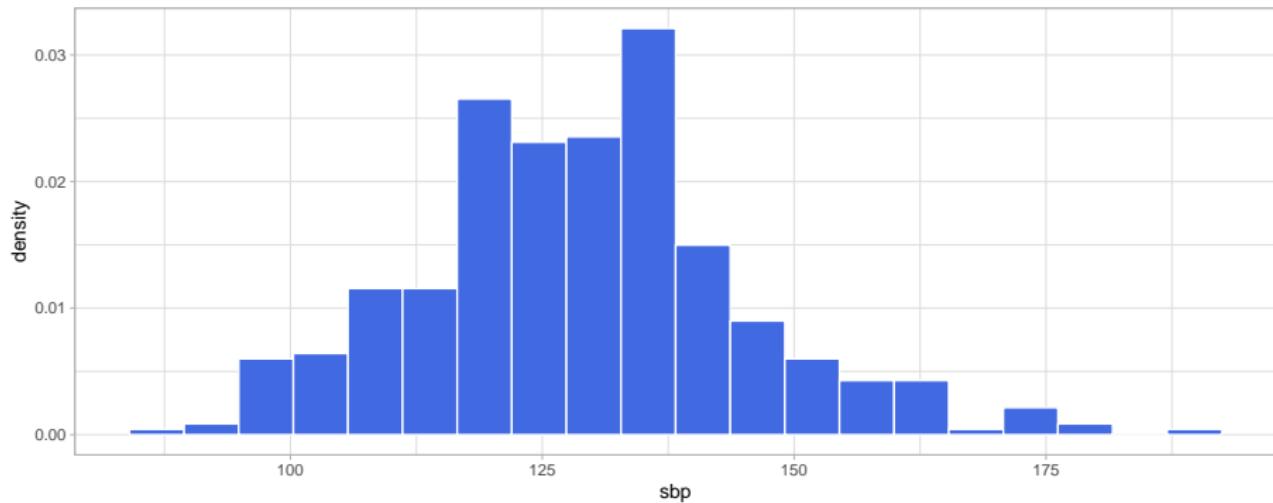
```
ggplot(data = dm431, aes(x = sbp)) +  
  geom_density(fill = "plum") +  
  labs("Systolic Blood Pressure for 431 women with diabetes")
```



Rescale dm431 SBP histogram as density

Suppose we want to rescale the histogram counts so that the bar areas integrate to 1. This will let us overlay a Normal density onto the results.

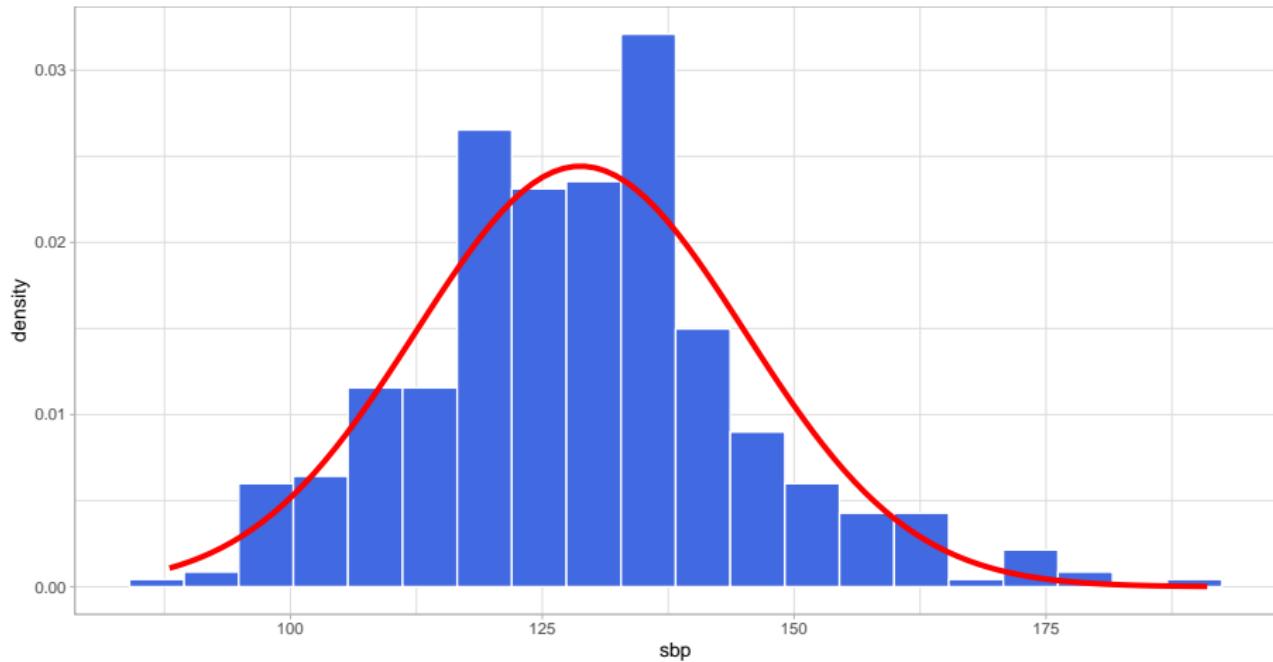
```
ggplot(dm431, aes(x = sbp)) +  
  geom_histogram(aes(y = stat(density)), bins = 20,  
                 fill = "royalblue", col = "white")
```



Density Function, with Normal superimposed

Now we can draw a Normal density curve on top of the rescaled histogram.

SBP density, with Normal model superimposed

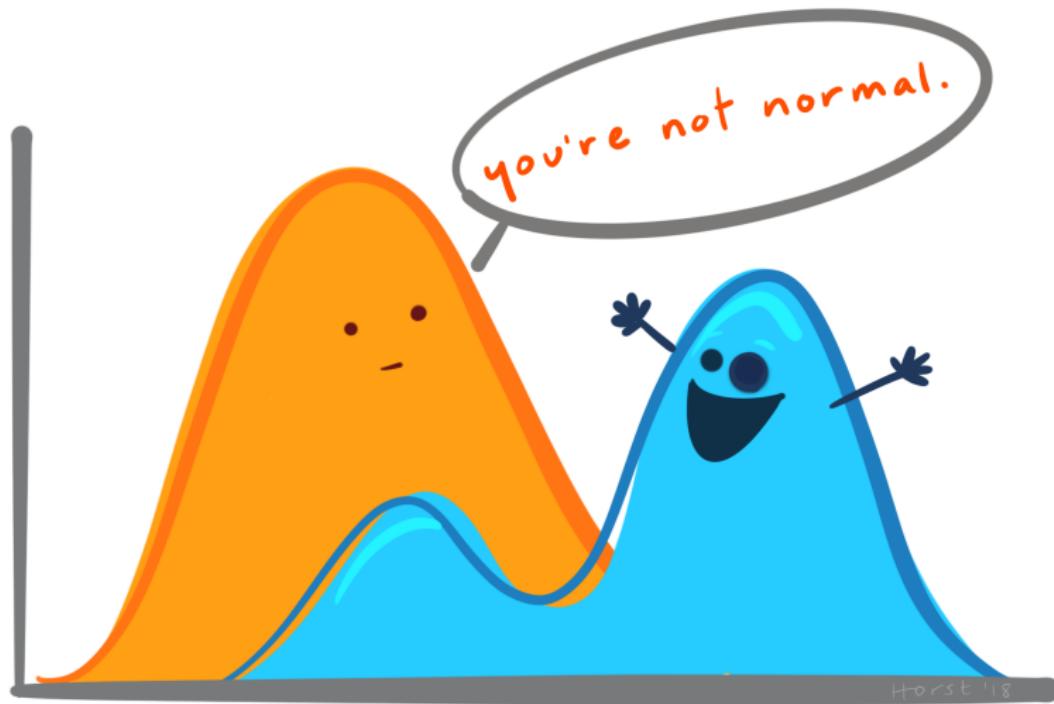


Code for plotting Histogram as Density function

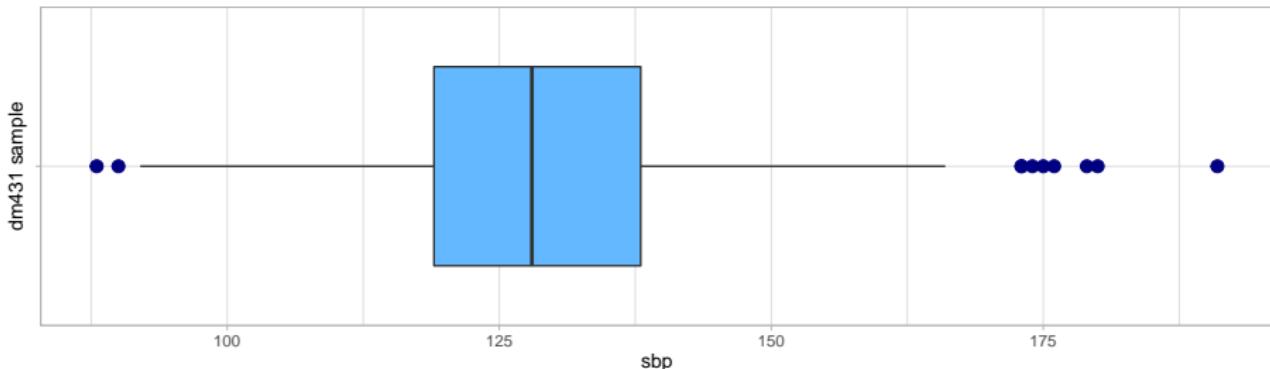
Including the superimposition of a Normal density on top of the histogram.

```
ggplot(dm431, aes(x = sbp)) +  
  geom_histogram(aes(y = stat(density)), bins = 20,  
                 fill = "royalblue", col = "white") +  
  stat_function(fun = dnorm,  
                args = list(mean = mean(dm431$sbp),  
                            sd = sd(dm431$sbp)),  
                col = "red", lwd = 1.5) +  
  labs(title = "SBP density, with Normal model superimposed")
```

Graphs are our most important tool!



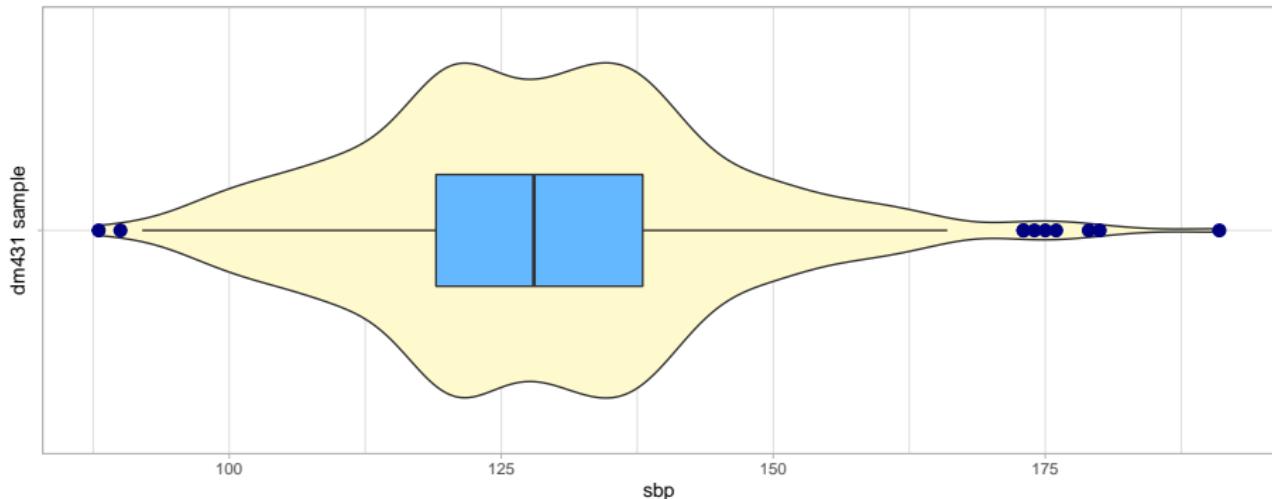
Boxplot for dm431 Systolic BP values



min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.8	16.3	431	0

- Box located at the quartiles (Q1 and Q3), with central line at median
- IQR = interquartile range = $Q3 - Q1$ = width of the box
- Fences identifying outlier candidates at $Q1 - 1.5 \text{ IQR}$ and $Q3 + 1.5 \text{ IQR}$
- Center, Spread, Shape?

Adding a Violin Plot of dm431 Systolic BPs



min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.8	16.3	431	0

- What does the violin plot suggest about the shape?

Code for Previous Slide

```
ggplot(dm431, aes(x = "", y = sbp)) +  
  geom_violin(fill = "lemonchiffon") +  
  geom_boxplot(width = 0.3, fill = "royalblue",  
                outlier.size = 3,  
                outlier.color = "royalblue") +  
  coord_flip() +  
  labs(x = "dm431 sample")  
  
mosaic::favstats(~ sbp, data = dm431) %>%  
  kable(digits = 1)
```

- Remember that you have the R Markdown code for every slide!

What would a sample of 431 systolic blood pressures from a Normal distribution look like?

New simulated sample from Normal distribution

Simulate a sample of 431 observations from a Normal distribution that has mean and standard deviation equal to the mean and standard deviation of our dm431 systolic blood pressures.

```
dm431 %>% summarize(mean(sbp), sd(sbp)) %>% kable(dig = 2)
```

mean(sbp)	sd(sbp)
128.79	16.33

```
set.seed(20210909) # note change from Class 05
sim_data <- tibble(
  sbp = rnorm(n = 431,
               mean = mean(dm431$sbp),
               sd = sd(dm431$sbp)))
```

Observed & Simulated Numerical Summaries

```
mosaic::favstats(~ sbp, data = dm431) %>% kable(dig = 1)
```

min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.8	16.3	431	0

```
mosaic::favstats(~ sbp, data = sim_data) %>% kable(dig = 1)
```

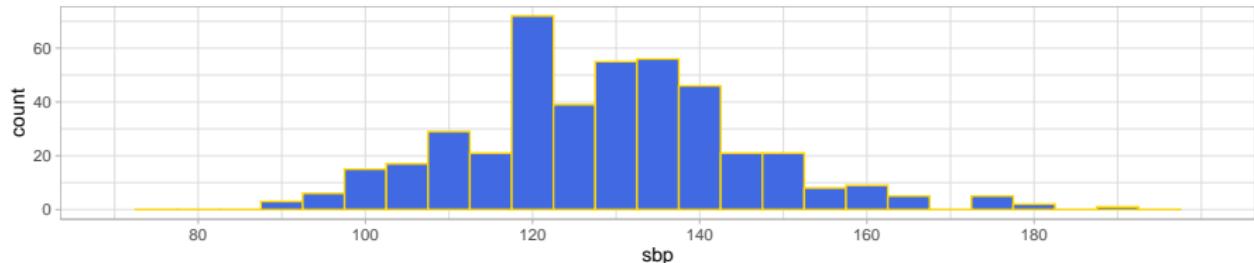
min	Q1	median	Q3	max	mean	sd	n	missing
80.5	118.9	129.6	139.6	178.8	129	16.5	431	0

- The first time you use `mosaic::favstats` in an R Markdown file, R will throw a message about a function that `ggplot2` and `mosaic` share.
- I suppress it with `{r, message = FALSE}` in the code chunk header.

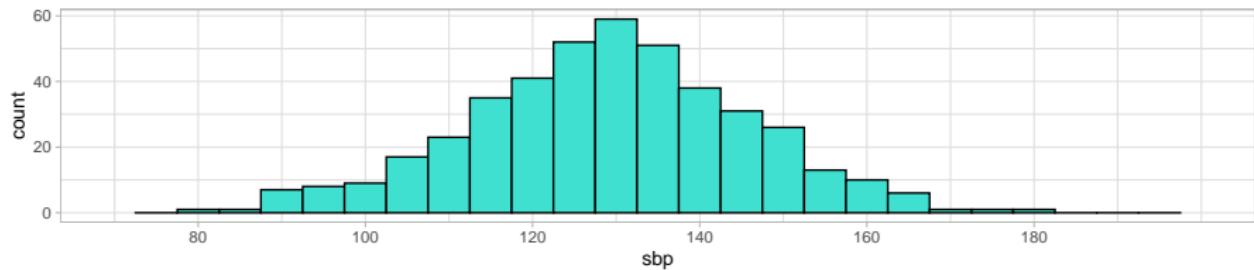
So, do the sbp values we observe look like our simulated sample from a Normal distribution?

Comparing histograms of dm431 and simulated SBP

431 Observed SBP values from dm431 (sample mean = 128.8, sd = 16.3)

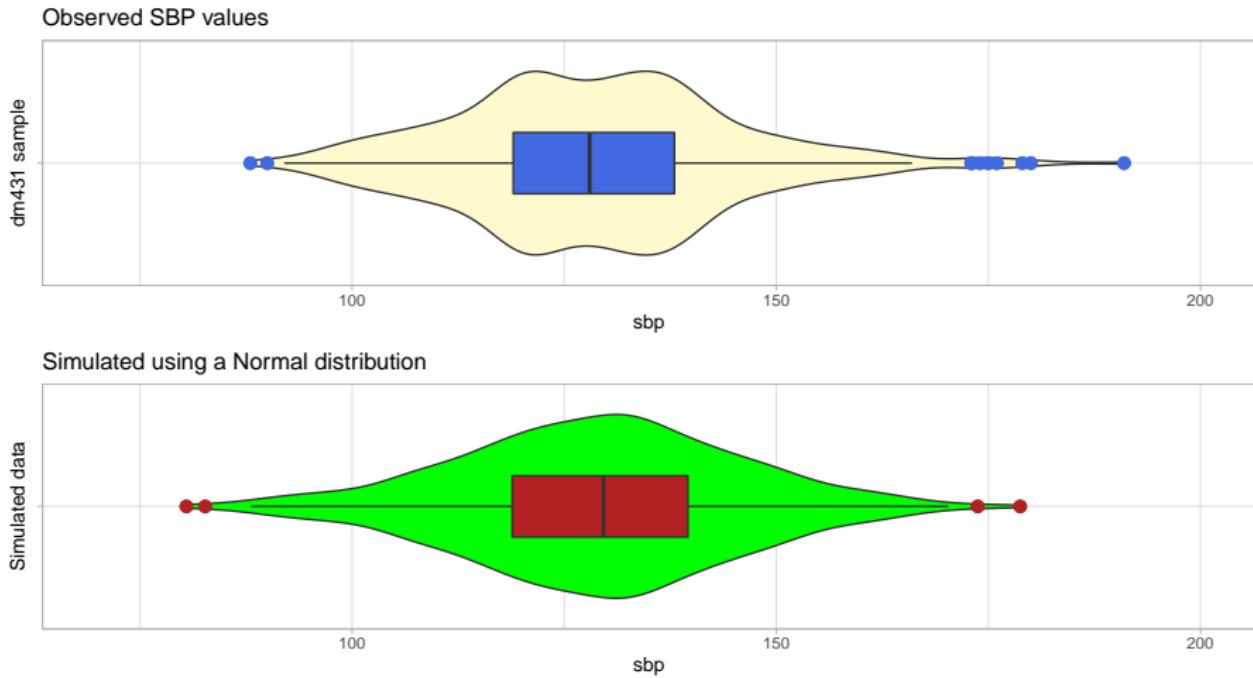


431 Simulated Values from Normal model with mean = 128.8, sd = 16.3



- Does a Normal model look appropriate for describing the SBP in dm431?

Observed vs. Simulated Systolic BPs

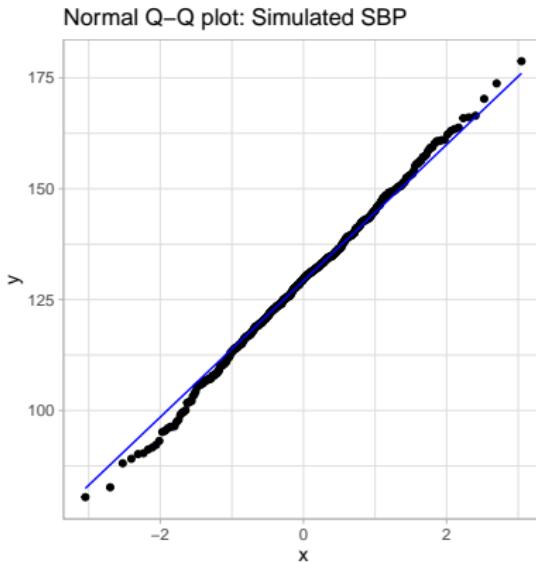


- Does a Normal model look appropriate for describing the SBP in dm431?

Using a Normal Q-Q plot to assess Normality of a batch of data

Normal Q-Q plot of our simulated data

Remember that these are draws from a Normal distribution, so this is what a sample of 431 Normally distributed data points should look like.



The Normal Q-Q Plot

Tool to help assess whether the distribution of a single sample is well-modeled by the Normal.

- Suppose we have N data points in our sample.
- Normal Q-Q plot will plot N points, on a scatterplot.
 - Y value is the data value
 - X value is the expected value for that point in a Normal distribution

Using the Normal distribution with the same mean and SD as our sample, R calculates what the minimum value is expected to be, given a sample of size N , then the next smallest value, and so forth all the way up until the maximum value.

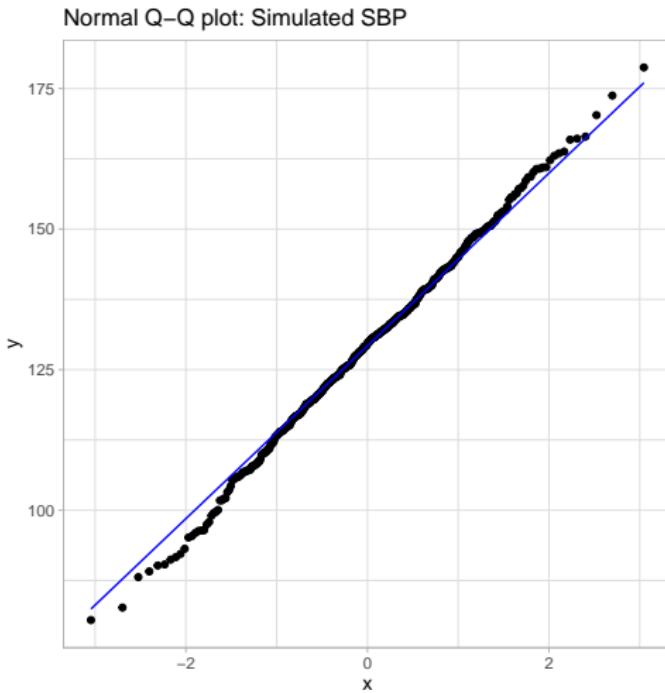
- X value in the Normal Q-Q plot is the value that a Normal distribution would take for that rank in the data set.
- We draw a line through $Y = X$, and points close to the line therefore match what we'd expect from a Normal distribution.

How do we create a Normal Q-Q plot?

For our simulated blood pressure data

```
ggplot(sim_data, aes(sample = sbp)) +  
  geom_qq() + # plot the points  
  geom_qq_line(col = "blue") + # plot the Y = X line  
  theme(aspect.ratio = 1) + # make the plot square  
  labs(title = "Normal Q-Q plot: Simulated SBP")
```

Result, again...



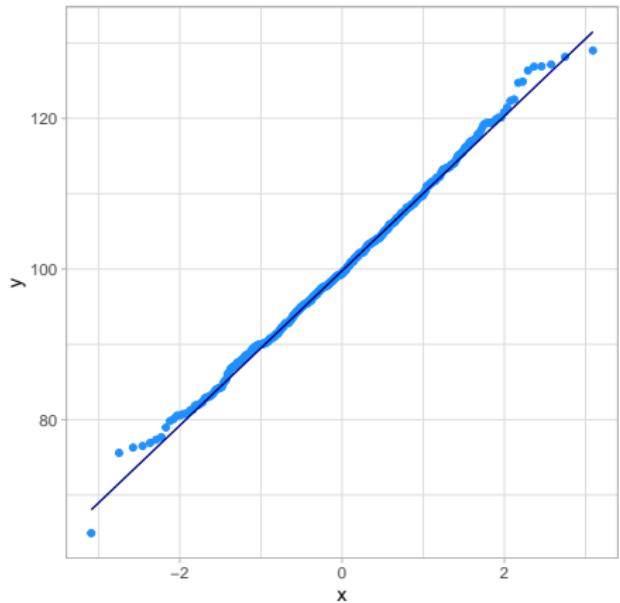
Interpreting the Normal Q-Q plot?

The Normal Q-Q plot can help us identify data as well approximated by a Normal distribution, or not, because of:

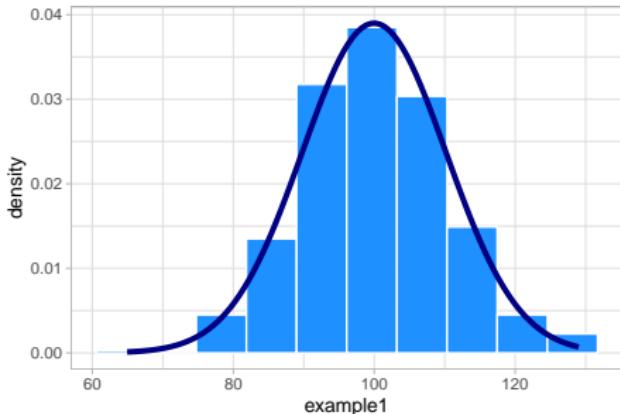
- skew (including distinguishing between right skew and left skew)
 - behavior in the tails (which could be heavy-tailed [more outliers than expected] or light-tailed)
- ① Normally distributed data are indicated by close adherence of the points to the diagonal reference line.
 - ② Skew is indicated by substantial curving (on both ends of the distribution) in the points away from the reference line (if both ends curve up, we have right skew; if both ends curve down, this indicates left skew)
 - ③ An abundance or dearth of outliers (as compared to the expectations of a Normal model) are indicated in the tails of the distribution by an “S” shape or reverse “S” shape in the points.

Example 1: Data from a Normal Distribution

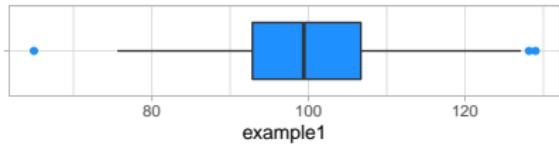
Normal Q-Q plot: Example 1



Density Function: Example 1



Boxplot: Example 1



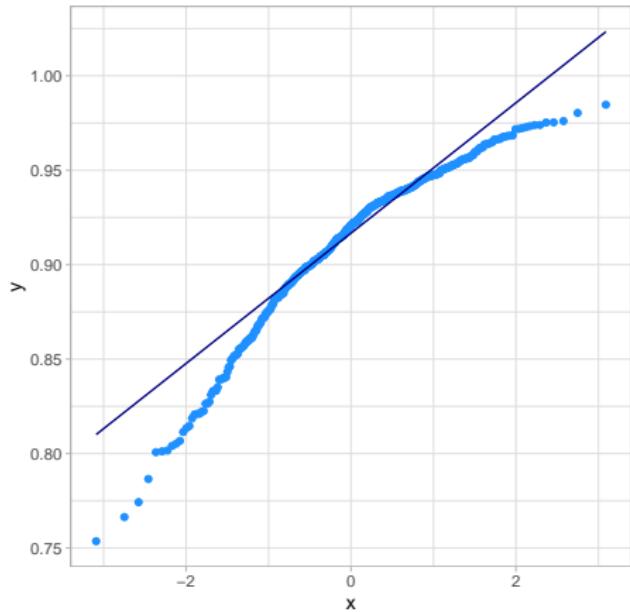
min	Q1	median	Q3	max	mean	sd	n	missing
64.9	92.8	99.4	106.7	129	100	10.2	500	0

Does a Normal model fit well for my data?

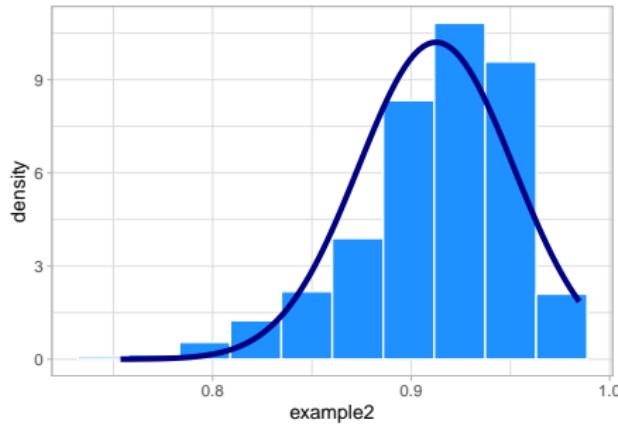
- ① Is a Normal Q-Q plot showing something close to a straight line, without clear signs of skew or indications of lots of outliers (heavy-tailedness)?
- ② Does a boxplot, violin plot and/or histogram also show a symmetric distribution, where both the number of outliers is modest, and the distance of those outliers from the mean is modest?
- ③ Do numerical measures match up with the expectations of a normal model?

Example 2: Data from a Left-Skewed Distribution

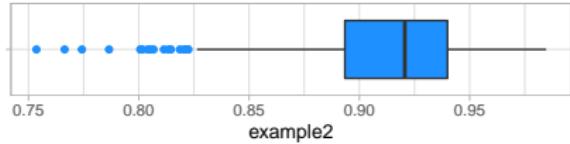
Normal Q-Q plot: Example 2



Density Function: Example 2



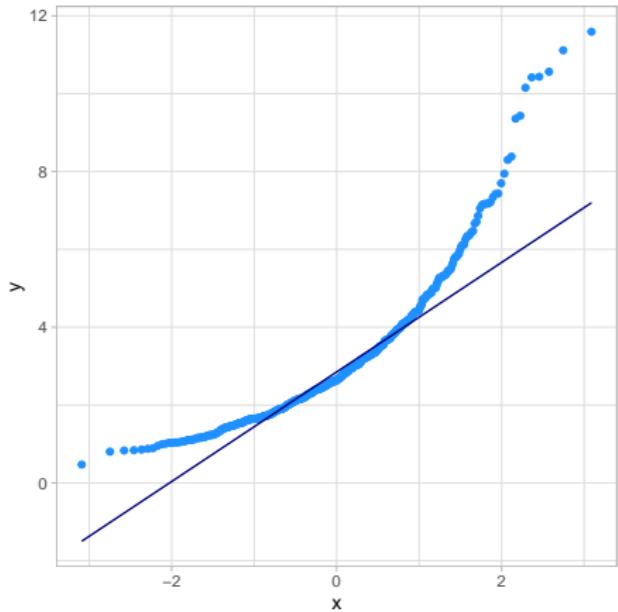
Boxplot: Example 2



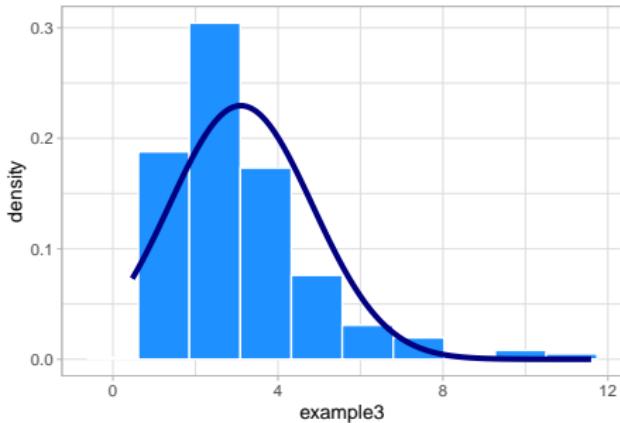
	min	Q1	median	Q3	max	mean	sd	n	missing
	0.8	0.9	0.9	0.9	1	0.9	0	500	0

Example 3: Data from a Right-Skewed Distribution

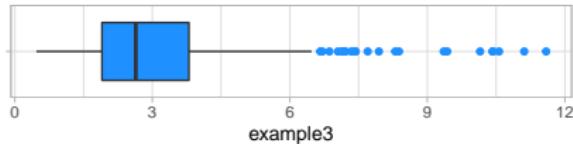
Normal Q-Q plot: Example 3



Density Function: Example 3



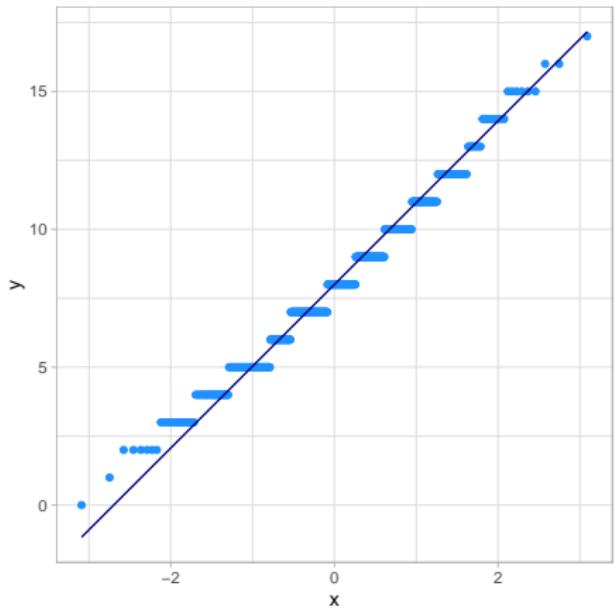
Boxplot: Example 3



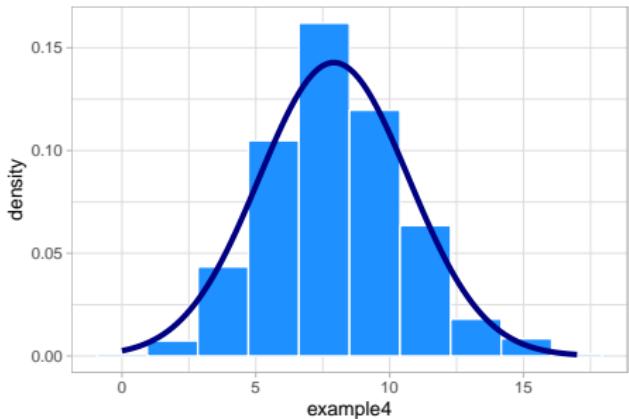
min	Q1	median	Q3	max	mean	sd	n	missing
0.5	1.9	2.6	3.8	11.6	3.1	1.7	500	0

Example 4: Discrete “Symmetric” Distribution

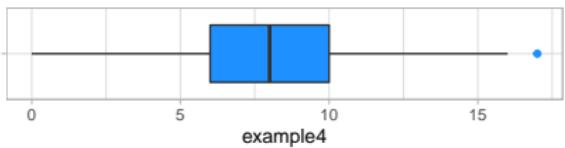
Normal Q-Q plot: Example 4



Density Function: Example 4



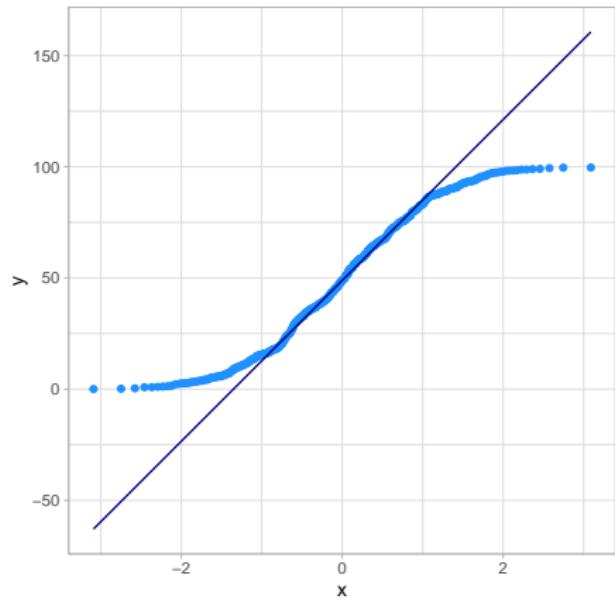
Boxplot: Example 4



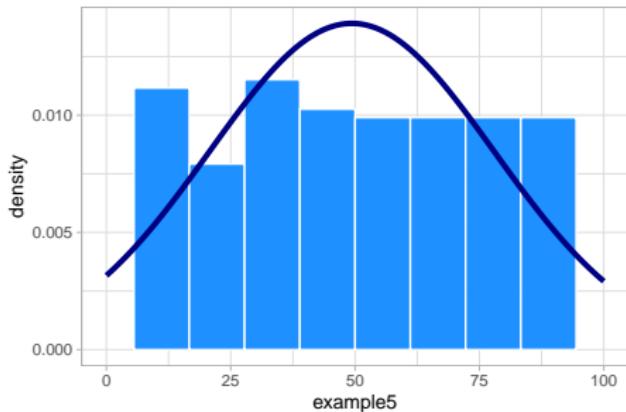
	min	Q1	median	Q3	max	mean	sd	n	missing
	0	6	8	10	17	7.9	2.8	500	0

Example 5: Data from a Uniform Distribution

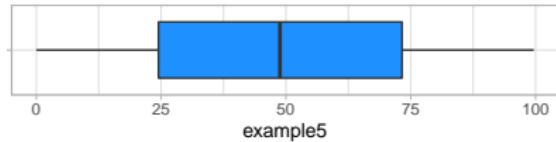
Normal Q-Q plot: Example 5



Density Function: Example 5



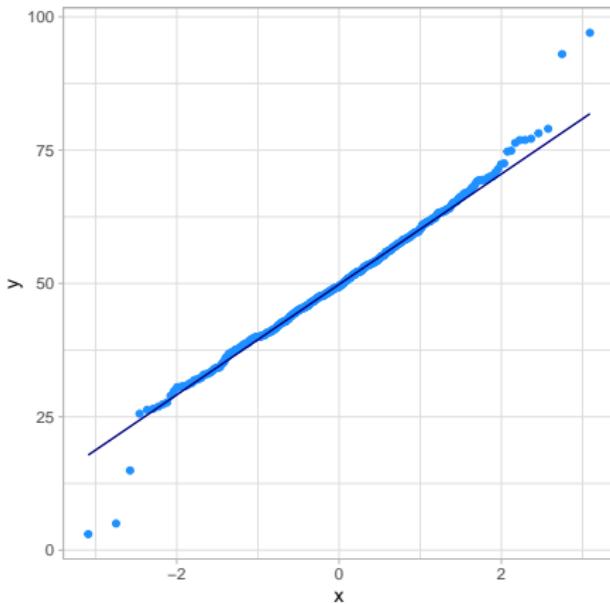
Boxplot: Example 5



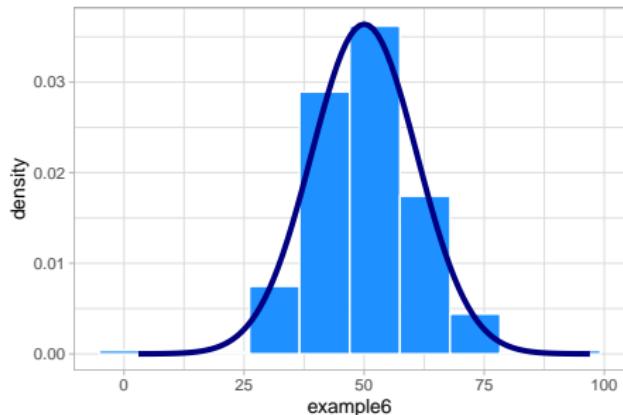
min	Q1	median	Q3	max	mean	sd	n	missing
0	24.5	48.8	73.2	99.6	49.3	28.7	500	0

Example 6: Symmetric data with outliers

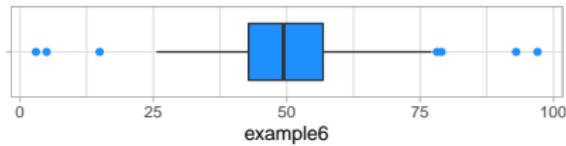
Normal Q-Q plot: Example 6



Density Function: Example 6



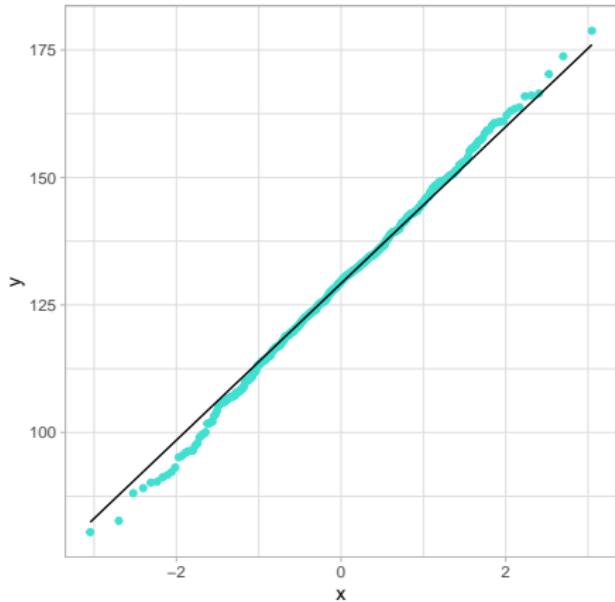
Boxplot: Example 6



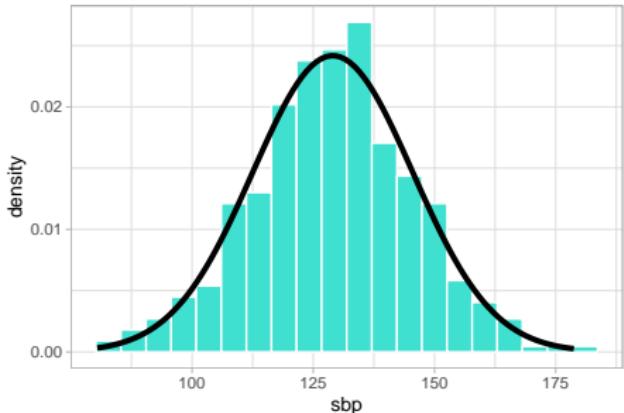
min	Q1	median	Q3	max	mean	sd	n	missing
3	42.8	49.4	56.8	97	50	11	500	0

Our 431 simulated Systolic Blood Pressures

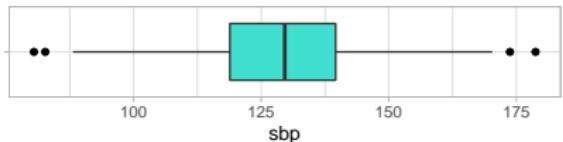
Normal Q-Q plot: sbp



Density Function: sbp

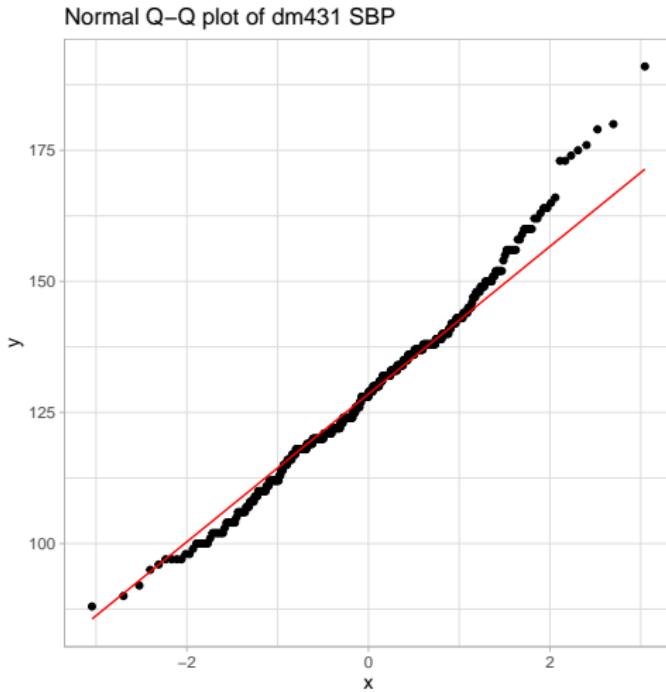


Boxplot: sbp



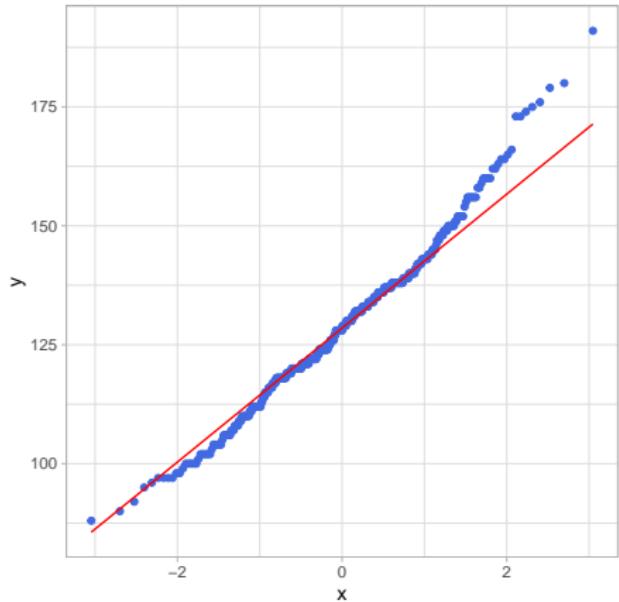
min	Q1	median	Q3	max	mean	sd	n	missing
80.5	118.9	129.6	139.6	178.8	129	16.5	431	0

A Normal Q-Q Plot of the dm431 SBP data

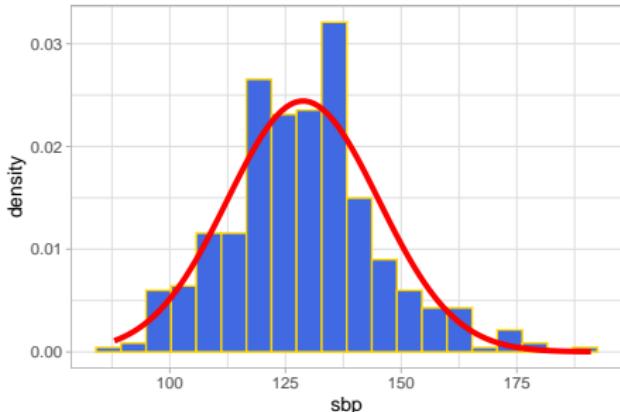


How do we build this slide?

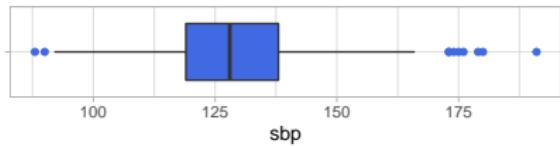
Normal Q-Q plot: dm431 SBP



Density Function: dm431 SBP



Boxplot: dm431 SBP



min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.8	16.3	431	0

Code for sbp in dm431 (First of Three Plots)

```
p1 <- ggplot(dm431, aes(sample = sbp)) +  
  geom_qq(col = "royalblue") +  
  geom_qq_line(col = "red") +  
  theme(aspect.ratio = 1) +  
  labs(title = "Normal Q-Q plot: dm431 SBP")
```

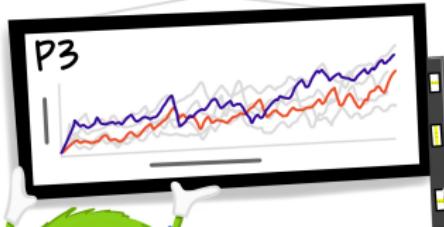
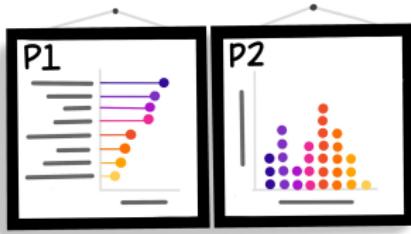
Code for sbp in dm431 (Second of Three Plots)

```
p2 <- ggplot(dm431, aes(x = sbp)) +
  geom_histogram(aes(y = stat(density)),
                 bins = 20,
                 fill = "royalblue", col = "gold") +
  stat_function(fun = dnorm,
                args = list(mean = mean(dm431$sbp),
                            sd = sd(dm431$sbp)),
                col = "red", lwd = 1.5) +
  labs(title = "Density Function: dm431 SBP")
```

Code for sbp in dm431 (Third of Three Plots)

```
p3 <- ggplot(dm431, aes(x = sbp, y = "")) +
  geom_boxplot(fill = "royalblue",
                outlier.color = "royalblue") +
  labs(title = "Boxplot: dm431 SBP", y = "")
```

Putting the plots together...



Using patchwork

```
p1 + (p2 / p3 + plot_layout(heights = c(4,1)))
```

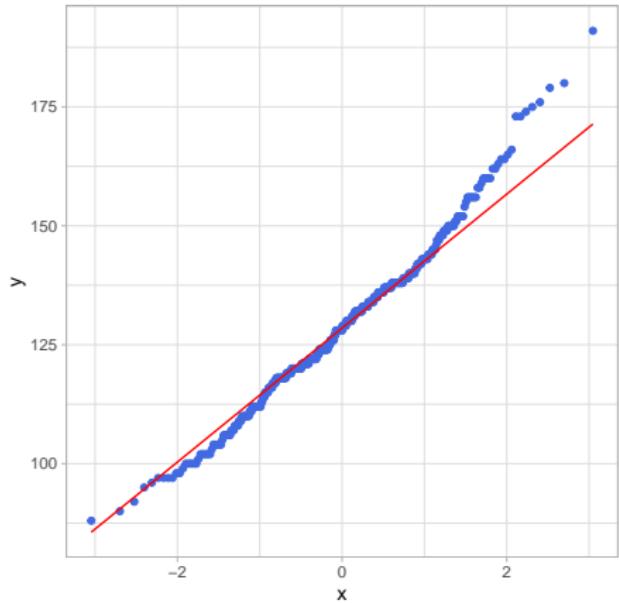
Also added...

```
mosaic::favstats(~ sbp, data = dm431) %>% kable(digits = 1)
```

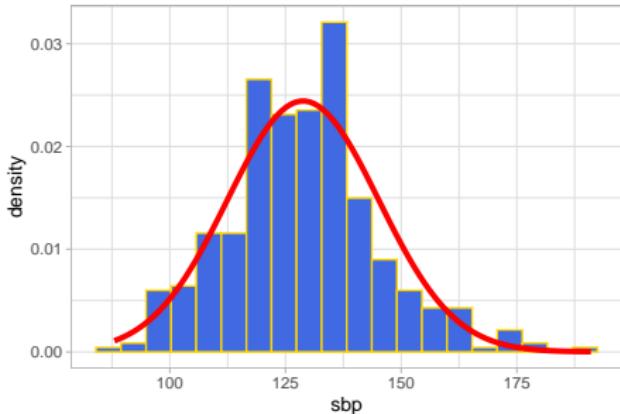
min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.8	16.3	431	0

Result: 431 observed Systolic BP values

Normal Q-Q plot: dm431 SBP



Density Function: dm431 SBP



Boxplot: dm431 SBP



min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.8	16.3	431	0

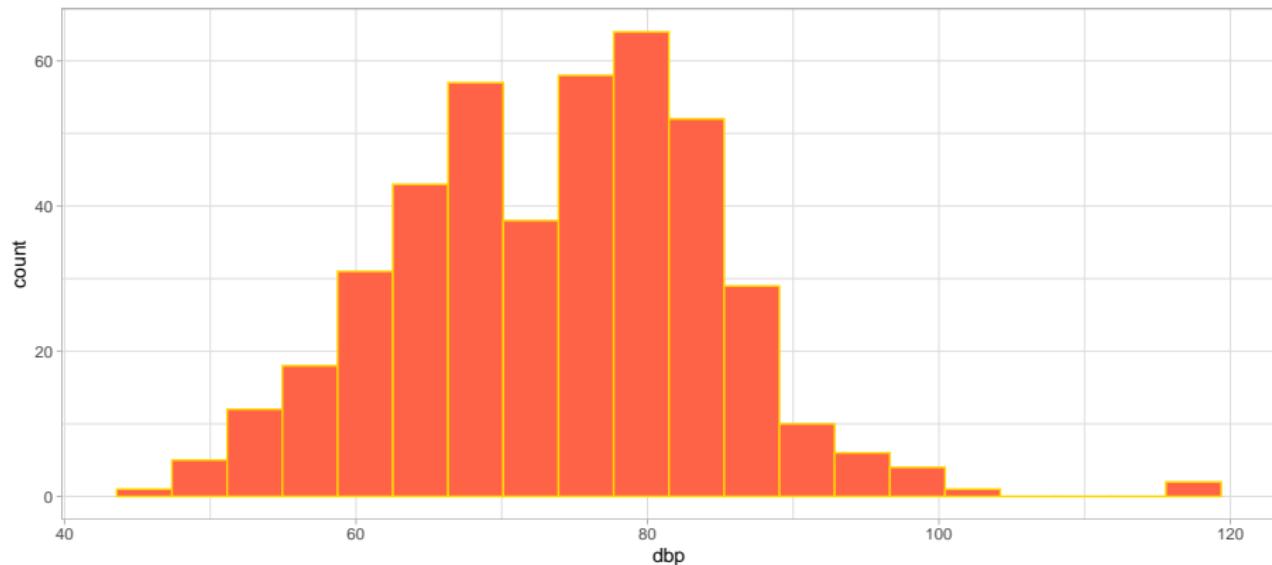
What Summaries to Report

It is usually helpful to focus on the shape, center and spread of a distribution. Bock, Velleman and DeVeaux provide some useful advice:

- If the data are skewed, report the median and IQR (or the three middle quantiles). You may want to include the mean and standard deviation, but you should point out why the mean and median differ. The fact that the mean and median do not agree is a sign that the distribution may be skewed. A histogram will help you make that point.
- If the data are symmetric, report the mean and standard deviation, and possibly the median and IQR as well.
- If there are clear outliers and you are reporting the mean and standard deviation, report them with the outliers present and with the outliers removed. The differences may be revealing. The median and IQR are not likely to be seriously affected by outliers.

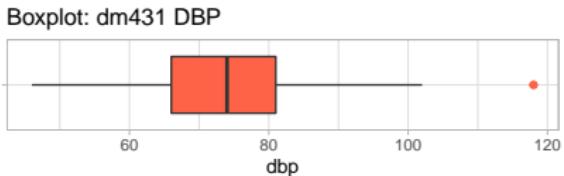
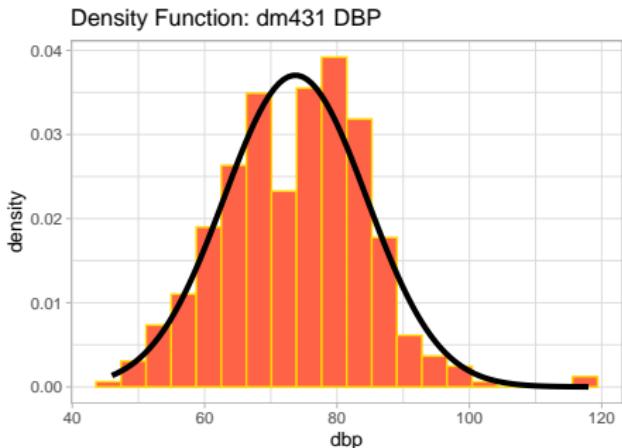
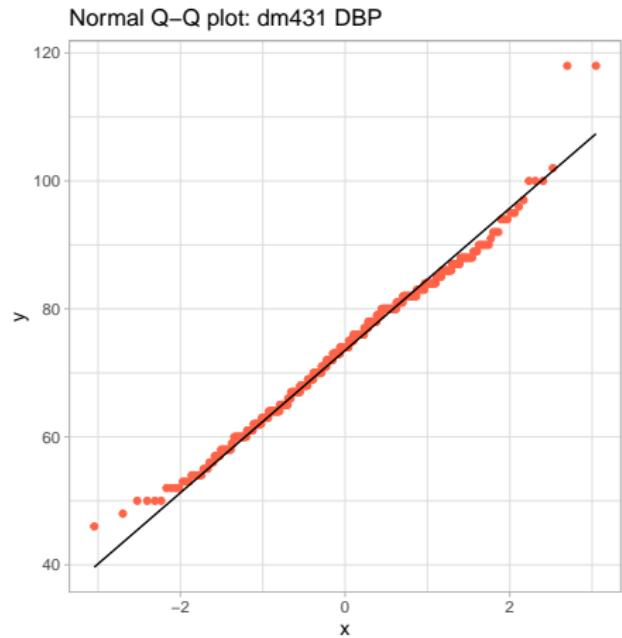
OK, what about Diastolic Blood Pressure?

```
ggplot(data = dm431, aes(x = dbp)) +  
  geom_histogram(bins = 20, fill = "tomato", col = "gold")
```



- We can generate the set of plots we've been using...

DBP in dm431: Center/Spread/Outliers/Shape?



min	Q1	median	Q3	max	mean	sd	n	missing
46	66	74	81	118	73.7	10.8	431	0

Does a Normal model fit well for my data?

- ① Is a Normal Q-Q plot showing something close to a straight line, without clear signs of skew or indications of lots of outliers (heavy-tailedness)?
- ② Does a boxplot, violin plot and/or histogram also show a symmetric distribution, where both the number of outliers is modest, and the distance of those outliers from the mean is modest?
- ③ Do numerical measures match up with the expectations of a normal model?

Hmisc::describe for dbp?

```
dm431 %$% Hmisc::describe(dbp)
```

dbp

	n	missing	distinct	Info	Mean	Gmd
431		0	51	0.999	73.71	12.08
.05		.10	.25	.50	.75	.90
56		60	66	74	81	86
.95						
90						

lowest : 46 48 50 52 53, highest: 96 97 100 102 118

What is a plausible diastolic blood pressure?

Stem-and-Leaf of dbp values?

```
stem(dm431$dbp, scale = 0.6, width = 55)
```

The decimal point is 1 digit(s) to the right of the |

4 68
5 00002222333444445556667777888888899
6 0000000000001111111222222222333333344444+58
7 000000000000000111111111222222222223333+83
8 000000000000000000000000000000000011111111122222+64
9 00000012224445567
10 0002
11 88

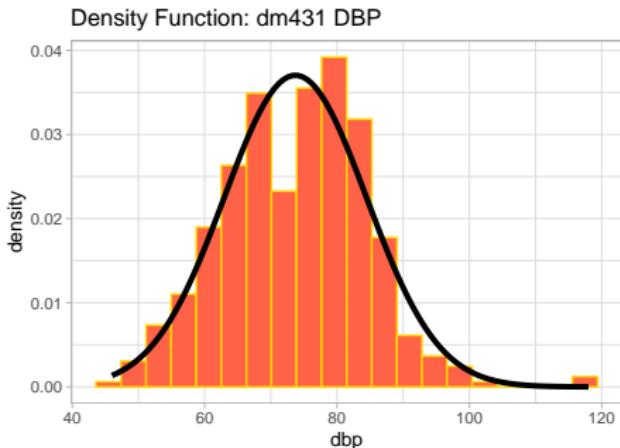
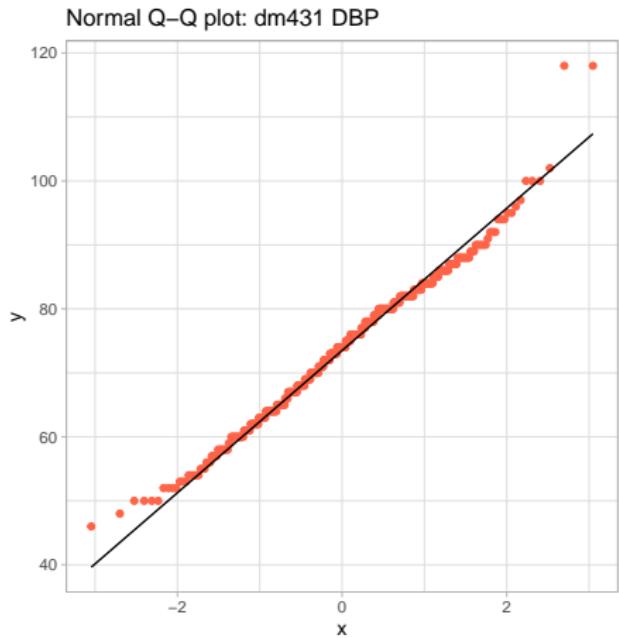
- I've specified scale and width just for this slide.

Who are those people with extreme dbp values?

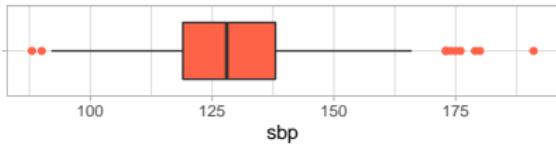
```
dm431 %>%  
  filter(dbp < 50 | dbp > 110) %>%  
  select(class5_id, sbp, dbp)
```

```
# A tibble: 4 x 3  
  class5_id    sbp    dbp  
  <chr>      <dbl>  <dbl>  
1 S-005        156    118  
2 S-202        124     46  
3 S-219        120     48  
4 S-240        158    118
```

dm431: Diastolic Blood Pressure



Boxplot: dm431 DBP



min	Q1	median	Q3	max	mean	sd	n	missing
46	66	74	81	118	73.7	10.8	431	0

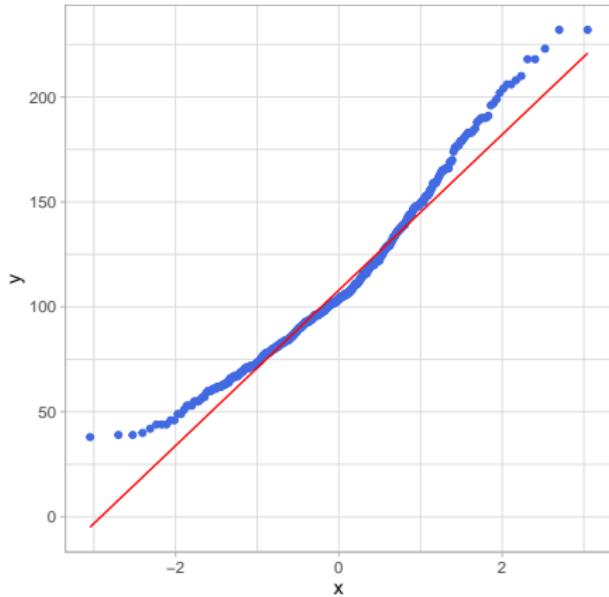
Making Decisions: Does a Normal Model fit well?

If a Normal model fits our data well, then we should see the following graphical indications:

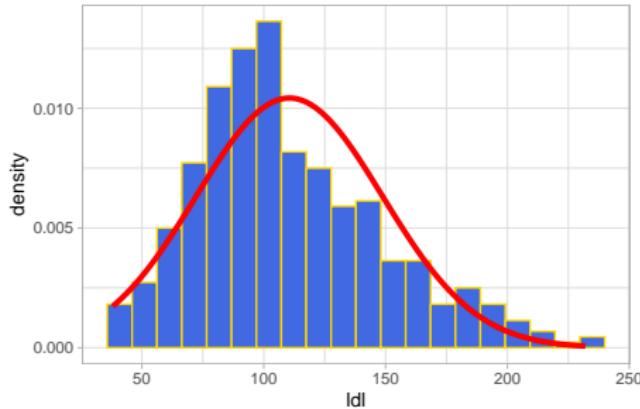
- ① A histogram that is symmetric and bell-shaped.
- ② A boxplot where the box is symmetric around the median, as are the whiskers, without a serious outlier problem.
- ③ A normal Q-Q plot that essentially falls on a straight line.

dm431: LDL Cholesterol

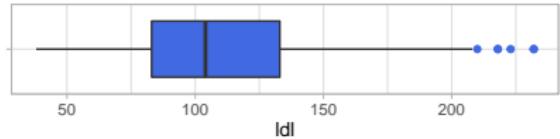
Normal Q-Q plot: dm431 LDL



Density Function: dm431 LDL



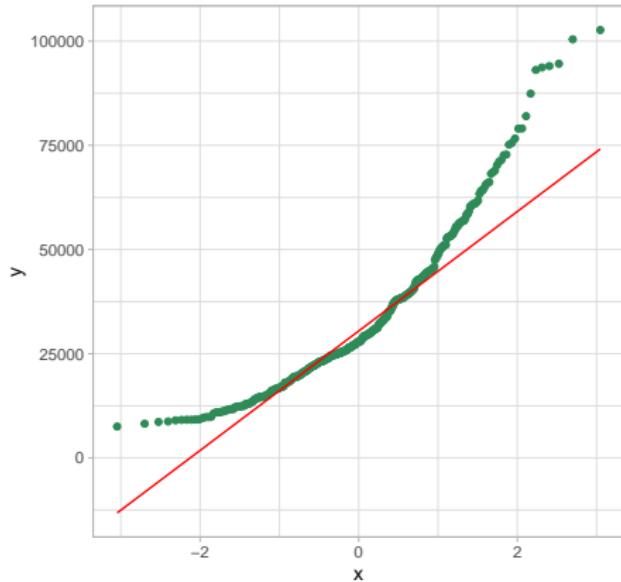
Boxplot: dm431 LDL



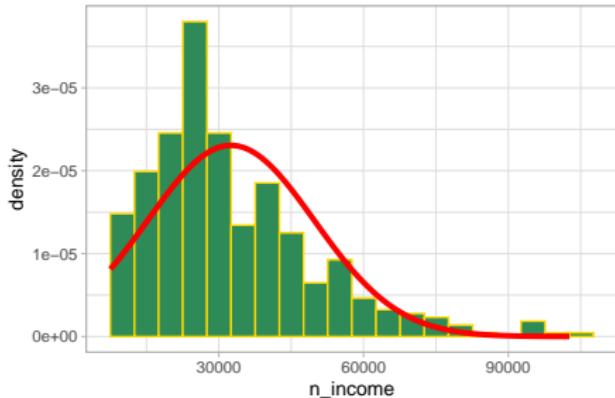
min	Q1	median	Q3	max	mean	sd	n	missing
38	83	104	133	232	110.5	38.3	431	0

dm431: Neighborhood Income

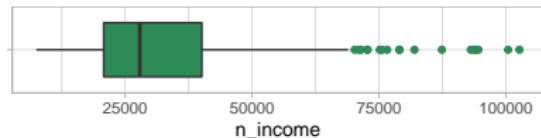
Normal Q-Q plot: dm431 Income



Density Function: dm431 Income



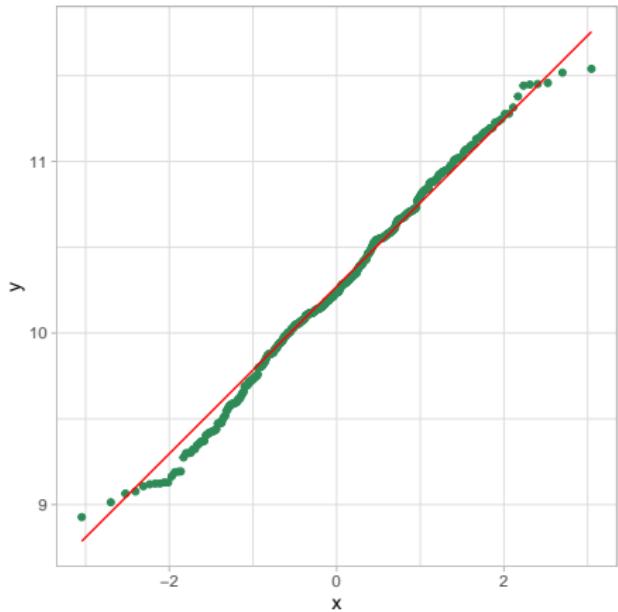
Boxplot: dm431 Income



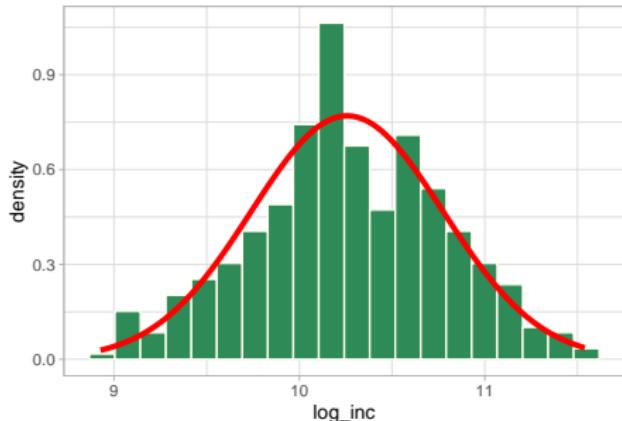
min	Q1	median	Q3	max	mean	sd	n	missing
7534	20794	27903	40128	102672	32514	17295	431	0

dm431: Natural Logarithm of Nbhd. Income

Normal Q-Q plot: log(dm431 Income)



Density Function: log(dm431 Income)



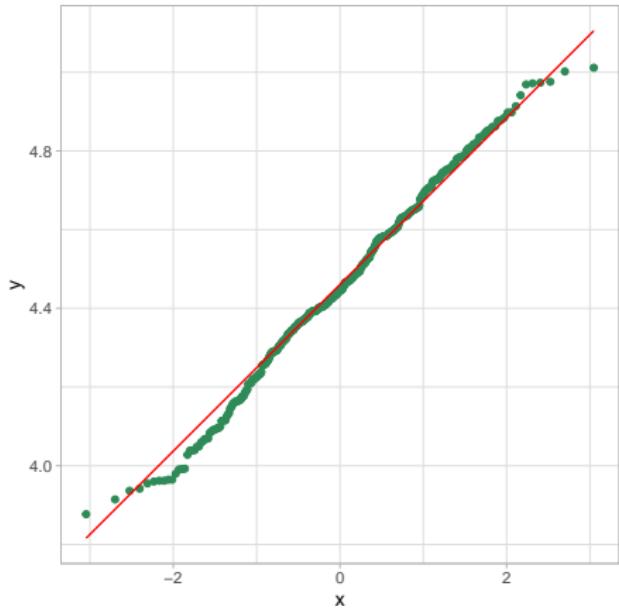
Boxplot: log(dm431 Income)



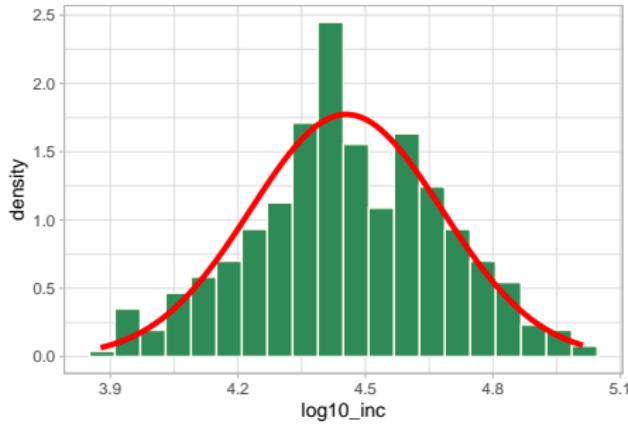
min	Q1	median	Q3	max	mean	sd	n	missing
8.93	9.94	10.24	10.6	11.54	10.26	0.52	431	0

dm431: Base-10 Logarithm of Nbhd. Income

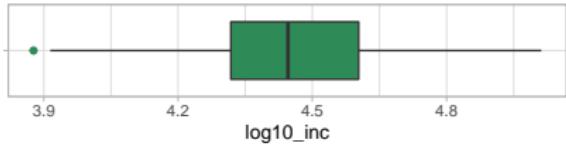
Normal Q-Q plot: $\log_{10}(n_income)$



Density Function: $\log_{10}(n_income)$



Boxplot: $\log_{10}(n_income)$



min	Q1	median	Q3	max	mean	sd	n	missing
3.88	4.32	4.45	4.6	5.01	4.45	0.23	431	0

Using Numerical Summaries to Assess Normality: A Good Idea?

Does a Normal model fit well for my data?

The least important approach (even though it is seemingly the most objective) is the calculation of various numerical summaries.

Semi-useful summaries help us understand whether they match up well with the expectations of a normal model:

- ① Assessing skewness with $skew_1$ (is the mean close to the median?)
- ② Assessing coverage probabilities (do they match the Normal model?)

Quantifying skew with $skew_1$

$$skew_1 = \frac{mean - median}{standard\ deviation}$$

Interpreting $skew_1$ (for unimodal data)

- $skew_1 = 0$ if the mean and median are the same
- $skew_1 > 0.2$ indicates fairly substantial right skew
- $skew_1 < -0.2$ indicates fairly substantial left skew

Measuring skewness in the SBP values: dm431?

```
mosaic::favstats(~ sbp, data = dm431)
```

min	Q1	median	Q3	max	mean	sd	n	missing
88	119	128	138	191	128.7889	16.33058	431	0

```
dm431 %>%
  summarize(skew1 = (mean(sbp) - median(sbp))/sd(sbp))
```

```
# A tibble: 1 x 1
  skew1
  <dbl>
1 0.0483
```

What does this suggest?

How about our other variables?

```
dm431 %>%
  summarize(
    dbp_skew1 = (mean(dbp) - median(dbp))/sd(dbp),
    ldl_skew1 = (mean(ldl) - median(ldl))/sd(ldl),
    ninc_skew1 = (mean(n_income) - median(n_income))/
      sd(n_income))

# A tibble: 1 x 3
  dbp_skew1 ldl_skew1 ninc_skew1
  <dbl>     <dbl>     <dbl>
1 -0.0271    0.170     0.267
```

- How do these results match up with our plots?

Empirical Rule for a Normal Model

If the data followed a Normal distribution, perfectly, then about:

- 68% of the data would fall within 1 standard deviation of the mean
- 95% of the data would fall within 2 standard deviations of the mean
- 99.7% of the data would fall within 3 standard deviations of the mean

Remember that, regardless of the distribution of the data:

- Half of the data will fall below the median, and half above it.
- Half of the data will fall in the Interquartile Range (IQR).

How many SBPs are within 1 SD of the mean?

```
dm431 %>%
  count(sbp > mean(sbp) - sd(sbp),
        sbp < mean(sbp) + sd(sbp)) %>%
  kable()
```

sbp > mean(sbp) - sd(sbp)	sbp < mean(sbp) + sd(sbp)	n
FALSE	TRUE	70
TRUE	FALSE	55
TRUE	TRUE	306

- Note that $306/431 = 0.71$, approximately.
- How does this compare to the expectation under a Normal model?

SBP and the mean \pm 2 standard deviations rule?

The total sample size here is 431

```
dm431 %>%
  count(sbp > mean(sbp) - 2*sd(sbp),
        sbp < mean(sbp) + 2*sd(sbp)) %>%
  kable()
```

sbp > mean(sbp) - 2 * sd(sbp)	sbp < mean(sbp) + 2 * sd(sbp)	n
FALSE	TRUE	5
TRUE	FALSE	15
TRUE	TRUE	411

- Note that $411/431 = 0.95$, approximately.
- How does this compare to the expectation under a Normal model?

Coverage Probabilities for our other variables

- Here are the observed percentages of our data from dm431 that fall in each of the limits specified by the Empirical Rule.

Variable	Mean	SD	Mean ± 1 SD	Within 2 SD	Within 3 SD
sbp	128.8	16.3	71%	95.4%	99.3%
dbp	73.7	10.8	71%	95.8%	99.5%
ldl	110.5	38.3	68.4%	95.4%	99.5%
n_income	32514	17295	72.2%	95.1%	98.4%

- What does this suggest about the effectiveness of a Normal distribution as a model for each of these variables?

Should we use hypothesis tests to assess Normality?

Hypothesis Testing to assess Normality

Don't. Graphical approaches are **far** better than hypothesis tests. . .

```
dm431 %$% shapiro.test(sbp)
```

Shapiro-Wilk normality test

```
data: sbp  
W = 0.98636, p-value = 0.0004525
```

The very small p value suggests **against** adopting a Normal model.

Variable	p value from Shapiro test
dbp	9.8×10^{-4}
ldl	0
n_income	0

- Exciting, huh? But not actually useful in any real sense.

Why not test for Normality?

There are multiple hypothesis testing schemes (Kolmogorov-Smirnov, etc.) and each looks for one specific violation of a Normality assumption. None can capture the wide range of issues our brains can envision, and none by itself is great at its job.

- With any sort of reasonable sample size, the test is so poor at detecting non-normality compared to our eyes, that it finds problems we don't care about (this is the main problem) and (also, sometimes) ignores problems we do care about.
- And without a reasonable sample size, the test is essentially useless.

Whenever you *can* avoid hypothesis testing and instead actually plot the data, you should plot the data. Sometimes you can't plot (especially with really big data) but the test should be your very last resort.

Summing Up: Does a Normal Model fit well?

If a Normal model fits our data well, then we should see the following graphical indications:

- ① A histogram that is symmetric and bell-shaped.
- ② A boxplot where the box is symmetric around the median, as are the whiskers, without a serious outlier problem.
- ③ A normal Q-Q plot that essentially falls on a straight line.

As for numerical summaries, we'd like to see

- ④ The mean and median within 0.2 standard deviation of each other.
- ⑤ No real evidence of too many outlier candidates (more than 5% starts to get us concerned about a Normal model)
- ⑥ No real evidence of individual outliers outside the reasonable range for the size of our data (we might expect about 3 observations in 1000 to fall more than 3 standard deviations away from the mean.)

Next Time

Building Confidence Intervals and making comparisons

431 Class 07

thomaselove.github.io/431

2021-09-14

Package Setup

```
library(janitor)
library(knitr)
library(magrittr)
library(modelsummary) # new today
library(naniar)
library(patchwork)
library(tidyverse)

theme_set(theme_bw())
```

Loading Some New Data

```
dm1000 <- read_csv("data/dm_1000.csv",
                     show_col_types = FALSE) %>%
  clean_names() %>%
  mutate(across(where(is.character), as_factor)) %>%
  mutate(subject = as.character(subject))
```

- 1000 (simulated) patients with diabetes between the ages of 31 and 75 who live in Cuyahoga County and are in one of four race-ethnicity categories, as well as one of four insurance categories.
- Same variables we saw in dm431 last week, but 1000 new subjects, and one new variable (`residence`)

Listing of dm1000 tibble

dm1000

A tibble: 1,000 x 17

	subject	age	insurance	n_income	ht	wt	sbp
	<chr>	<dbl>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>
1	M-0001	55	Medicaid	29853	1.63	103.	145
2	M-0002	52	Commercial	31248	1.75	112.	151
3	M-0003	69	Medicare	23362	1.65	74.9	127
4	M-0004	57	Medicaid	26033	1.63	81.4	125
5	M-0005	68	Medicare	85374	1.69	92.6	120
6	M-0006	56	Medicaid	31273	1.71	54.6	127
7	M-0007	54	Commercial	25445	1.68	81.6	114
8	M-0008	45	Medicare	67526	1.69	80.6	166
9	M-0009	61	Medicare	15203	1.91	86.7	111
10	M-0010	63	Medicaid	17628	1.86	123.	146

... with 990 more rows, and 10 more variables:

dbp <dbl>, a1c <dbl>, ldl <dbl>, tobacco <fct>,

statin <dbl>, euv_exam <dbl>

Code Book

Variable	Description
subject	subject code (M-0001 through M-1000)
age	subject's age, in years
insurance	primary insurance, 4 levels
n_income	neighborhood median income, in \$
ht	height, in meters (2 decimal places)
wt	weight, in kilograms (2 decimal places)
sbp	most recent systolic blood pressure (mm Hg)
dbp	most recent diastolic blood pressure (mm Hg)
a1c	most recent Hemoglobin A1c (%, with one decimal)
ldl	most recent LDL cholesterol level (mg/dl)

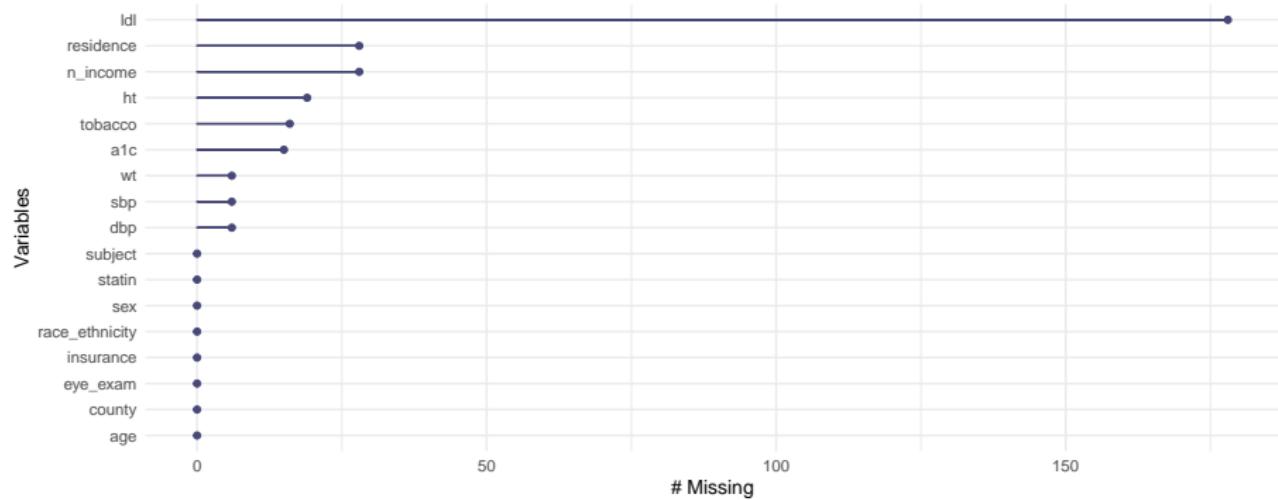
with seven more variables shown on the next slide...

Remainder of dm1000 codebook

Variable	Description
tobacco	most recent tobacco status, 3 levels
statin	1 = prescribed a statin in past 12m, 0 = not
eye_exam	1 = diabetic eye exam in past 12m, 0 = not
race_ethnicity	race/ethnicity category, 3 levels
sex	Female or Male
county	all subjects turn out to be in Cuyahoga County
residence	Cleveland or Suburbs

Any Missing Data?

```
gg_miss_var(dm1000)
```



- For now, `gg_miss_var()` throws a warning, which I've suppressed.

Counts of missingness, by variable

```
miss_var_summary(dm1000)
```

```
# A tibble: 17 x 3
```

	variable	n_miss	pct_miss
	<chr>	<int>	<dbl>
1	ldl	178	17.8
2	n_income	28	2.8
3	residence	28	2.8
4	ht	19	1.9
5	tobacco	16	1.6
6	a1c	15	1.5
7	wt	6	0.6
8	sbp	6	0.6
9	dbp	6	0.6
10	subject	0	0
11	age	0	0
12	insurance	0	0

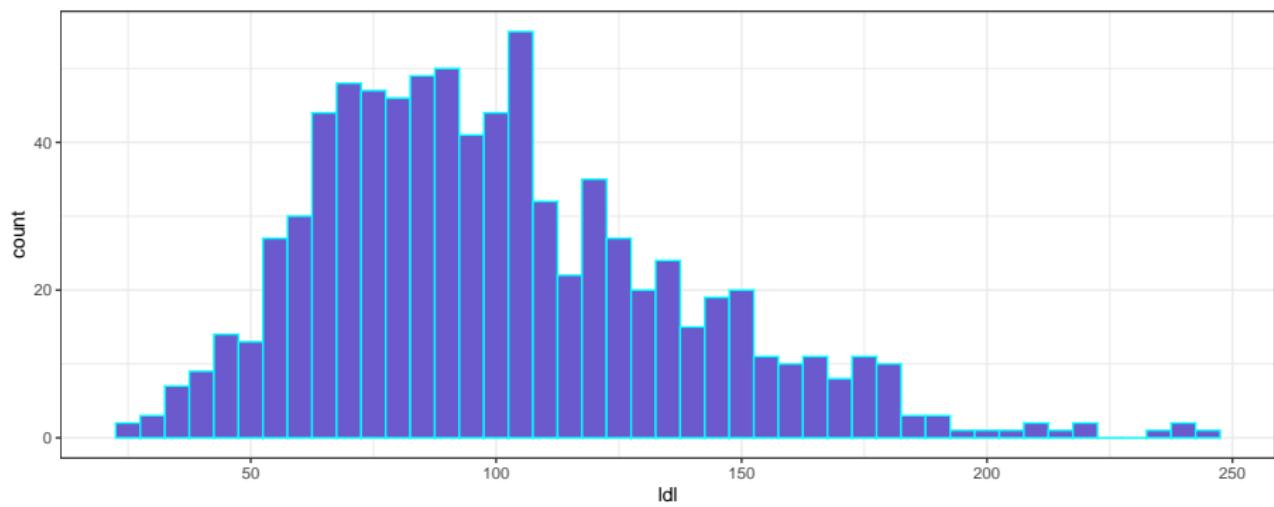
How should we summarize data with missing values?

- If you are providing a data summary, then you should summarize the complete cases, and specify the number of missing values.
- If you are intending to use the sample you've collected to make an inference about a process or population or to build a model, then you may want to consider whether or not a complete-case analysis will introduce bias.

What do graphs do with missing data?

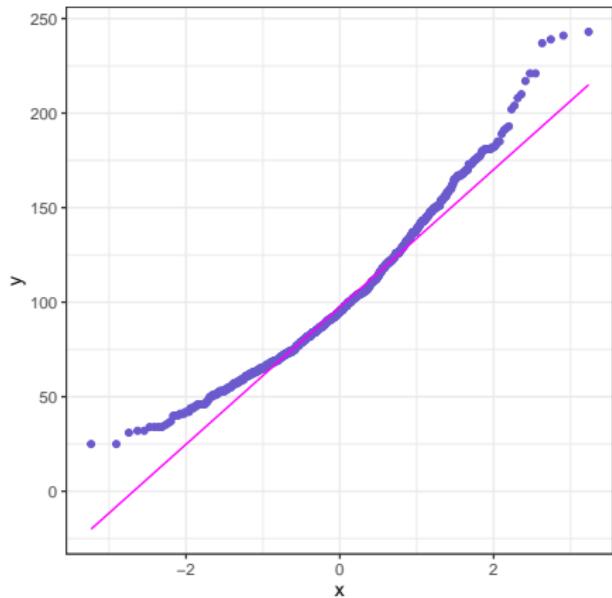
```
ggplot(data = dm1000, aes(x = ldl)) +  
  geom_histogram(binwidth = 5,  
                 fill = "slateblue", col = "cyan")
```

Warning: Removed 178 rows containing non-finite values
(stat_bin).

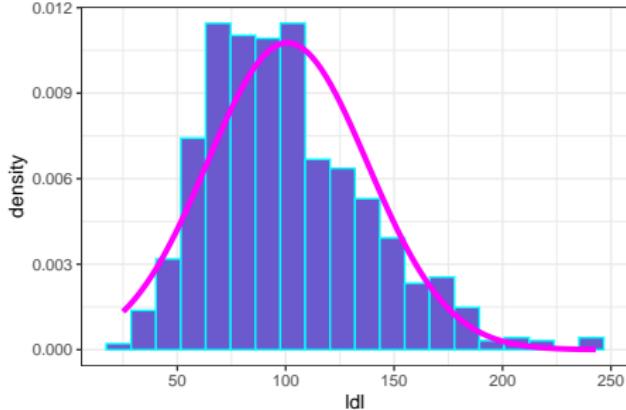


Exploring ldl in dm1000 (warnings suppressed)

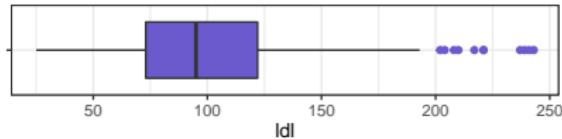
Normal Q-Q plot: dm1000 LDL



Density Function: dm1000 LDL



Boxplot: dm1000 LDL



min	Q1	median	Q3	max	mean	sd	n	missing
25	73	95	122	243	100.7	37.1	822	178

Numerical Summaries with missing values...

```
summary(dm1000$a1c)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	4.100	6.400	7.200	7.853	8.900	16.900
NA's						
	15					

```
mosaic::favstats(~ sbp, data = dm1000)
```

	min	Q1	median	Q3	max	mean	sd	n	missing
	84	122	132	142	209	132.7746	17.95214	994	6

Summarizing dm1000 with tools from modelsummary

Note that `modelsummary` really works well for HTML output but not for these PDF slides, so I am using images from an HTML knitting of the code below to get the results you're seeing.

```
dm1000 %>% select(age, ht, sbp, a1c) %$%
datasummary_skim(.)
```

- I've used `{r, eval = FALSE}` in the code chunk header to get R to print the code but not evaluate it in this slide.
- Results on next slide

```
dm1000 %>% select(age, ht, sbp, a1c) %>%  
datasummary_skim(.)
```

	Unique (#)	Missing (%)	Mean	SD	Min	Median	Max	
age	45	0	57.2	10.1	31.0	58.0	75.0	
ht	56	2	1.7	0.1	1.4	1.7	2.0	
sbp	100	1	132.8	18.0	84.0	132.0	209.0	
a1c	99	2	7.9	2.0	4.1	7.2	16.9	

Summarizing age within insurance type

```
mosaic::favstats(age ~ insurance, data = dm1000) %>%  
  kable(digits = 0)
```

insurance	min	Q1	median	Q3	max	mean	sd	n	missing
Medicaid	31	46	52	58	64	51	8	330	0
Commercial	33	50	56	60	72	55	8	196	0
Medicare	32	58	66	70	75	63	9	432	0
Uninsured	33	51	56	59	64	54	8	42	0

datasummary: ages within insurance types

```
datasummary(insurance * age ~  
            (N + mean + sd + min + P50 + max),  
            data = dm1000, histogram = FALSE)
```

- Results on next slide

```
datasummary(insurance * age ~  
            (N + mean + sd + min + P50 + max) ,  
            data = dm1000)
```

insurance		N	mean	sd	min	P50	max
Medicaid	age	330	51.21	8.39	31.00	52.00	64.00
Commercial	age	196	55.05	7.62	33.00	56.00	72.00
Medicare	age	432	63.05	9.28	32.00	66.00	75.00
Uninsured	age	42	54.10	7.69	33.00	56.50	64.00

Mean of sbp

```
mean(dm1000$sbp)
```

```
[1] NA
```

```
mean(dm1000$sbp, na.rm = TRUE)
```

```
[1] 132.7746
```

Summarizing sbp in dm1000

```
mosaic::favstats(~ sbp, data = dm1000) %>%
  kable(digits = 1)
```

min	Q1	median	Q3	max	mean	sd	n	missing
84	122	132	142	209	132.8	18	994	6

```
Hmisc::describe(dm1000$sbp)
```

dm1000\$sbp

	n	missing	distinct	Info	Mean	Gmd
	994	6	99	1	132.8	19.86
	.05	.10	.25	.50	.75	.90
	105.0	110.0	122.0	132.0	142.0	154.7
	.95					
	165.0					

Summarizing age and sbp with datasummary

```
datasummary((age + sbp) ~  
            (mean + sd + min + P50 + max),  
            data = dm1000)
```

- Results on next slide

```
datasummary((age + sbp) ~  
            (mean + sd + min + P50 + max),  
            data = dm1000)
```

	mean	sd	min	P50	max
age	57.20	10.10	31.00	58.00	75.00
sbp				132.00	

Applying na.rm = TRUE to multiple summaries

```
datasummary((age + sbp + ldl + ht) ~  
            (mean + sd + min + P50 + max) *  
            Arguments(na.rm = TRUE),  
            data = dm1000)
```

- Results on next slide

```
datasummary((age + sbp + ldl + ht) ~  
            (mean + sd + min + P50 + max) *  
            Arguments(na.rm = TRUE),  
            data = dm1000)
```

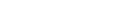
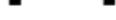
	mean	sd	min	P50	max
age	57.20	10.10	31.00	58.00	75.00
sbp	132.77	17.95	84.00	132.00	209.00
ldl	100.72	37.05	25.00	95.00	243.00
ht	1.69	0.10	1.36	1.69	1.97

datasummary_skim on the whole tibble

```
datasummary_skim(dm1000)
```

- Again, results on next slide
- Which variables are NOT included?

```
datasummary_skim(dm1000)
```

	Unique (#)	Missing (%)	Mean	SD	Min	Median	Max	
age	45	0	57.2	10.1	31.0	58.0	75.0	
n_income	965	3	35177.9	18775.9	2279.0	30586.5	129549.0	
ht	56	2	1.7	0.1	1.4	1.7	2.0	
wt	886	1	95.0	25.9	26.0	92.2	218.2	
sbp	100	1	132.8	18.0	84.0	132.0	209.0	
dbp	72	1	74.5	12.4	41.0	75.0	137.0	
a1c	99	2	7.9	2.0	4.1	7.2	16.9	
ldl	159	18	100.7	37.1	25.0	95.0	243.0	
statin	2	0	0.8	0.4	0.0	1.0	1.0	- 
eye_exam	2	0	0.6	0.5	0.0	1.0	1.0	- 

Using a Sample to Estimate a Population Mean

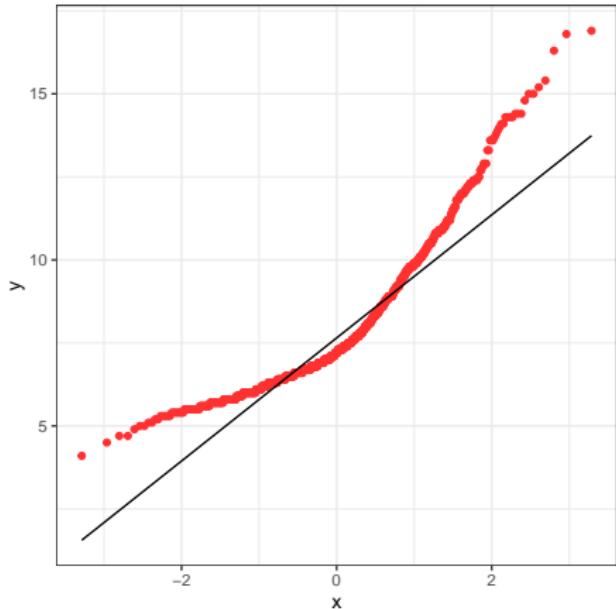
Suppose our sample in `dm1000` is a random sample from the population of all Cuyahoga County residents between the ages of 31-75 receiving care for diabetes.

What's a good estimate for the mean Hemoglobin A1c of the people in that population?

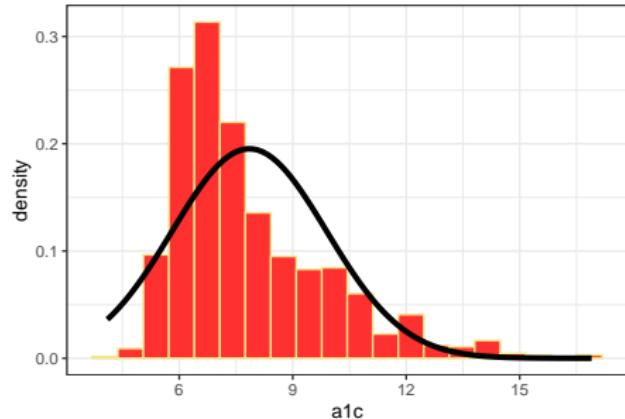
- How would we make this estimate using our data?
- What would we want to know about the data?
- DTDP

Hemoglobin A1c in the dm1000 sample

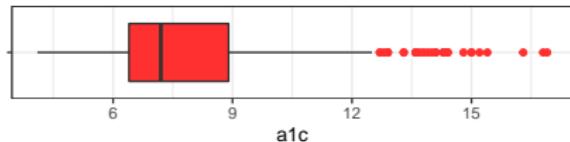
Normal Q–Q plot: dm1000 a1c



Density Function: dm1000 a1c



Boxplot: dm1000 a1c



min	Q1	median	Q3	max	mean	sd	n	missing
4.1	6.4	7.2	8.9	16.9	7.85	2.04	985	15

What if we assume the A1cs are Normally distributed?

Then we could use a linear regression model to obtain a 95% confidence interval for the population mean.

```
m1 <- lm(a1c ~ 1, data = dm1000)
tidy(m1, conf.int = TRUE, conf.level = 0.95) %>%
  select(estimate, conf.low, conf.high) %>%
  kable(digits = 3)
```

estimate	conf.low	conf.high
7.853	7.725	7.981

What is the model we've fit here?

```
m1
```

Call:

```
lm(formula = a1c ~ 1, data = dm1000)
```

Coefficients:

(Intercept)

7.853

- This “intercept only” model simply predicts the mean value of our outcome, a1c.

How do we interpret our 95% confidence interval?

We are estimating the mean of the **population** based on a mean from our *sample* of 1000 observations taken (at random, we assume) from that population.

- Our 95% confidence interval is (7.73, 7.98).
 - Sadly, this **doesn't** mean we're 95% confident that the actual population mean is in that range, even though lots and lots of people (incorrectly) assume it does.
- Essentially, we have 95% confidence in the **process** of fitting confidence intervals this way. If we fit 100 such confidence intervals to a variety of data sets, we have some reason to anticipate that 95 of them will contain the actual unknown value of the population mean.
 - That's oversimplifying a little, and a particular concern in this case is that the data are somewhat skewed (at any rate, not very close to Normally distributed) and this may impact our ability to generate accurate and efficient confidence intervals via a linear model like this.

An Equivalent Approach

We could use a t test to obtain a 95% confidence interval for the population mean.

```
tt <- dm1000 %$% t.test(a1c, conf.level = 0.95)
tidy(tt) %>% select(estimate, conf.low, conf.high) %>%
  kable(digits = 3)
```

estimate	conf.low	conf.high
7.853	7.725	7.981

This is exactly the same result as we obtain from the linear model `m1`.

Another Equivalent Approach

We could use a function called `smean.cl.normal()` from the `Hmisc` package to obtain the same 95% confidence interval for the population mean.

```
dm1000 %$% Hmisc::smean.cl.normal(a1c, conf.int = 0.95)
```

Mean	Lower	Upper
7.852893	7.725167	7.980619

Again, this is the same result as we have seen previously.

What if we weren't willing to assume that the A1c values came from a Normal distribution?

- My first instinct would be to use a bootstrap confidence interval to estimate the population mean with a 95% confidence interval.

```
set.seed(20210914) # why do we set a seed here?  
dm1000 %$% Hmisc::smean.cl.boot(a1c, conf.int = 0.95)
```

Mean	Lower	Upper
7.852893	7.725766	7.977069

```
set.seed(431) # what happens if we change the seed  
dm1000 %$% Hmisc::smean.cl.boot(a1c, conf.int = 0.95)
```

Mean	Lower	Upper
7.852893	7.729234	7.993604

95% Confidence Intervals for Population Mean A1c

Approach	Estimate	95% CI	Assume Normality?
Linear Model (t test)	7.853	(7.725, 7.981)	Yes
Bootstrap	7.853	(7.726, 7.977)	No

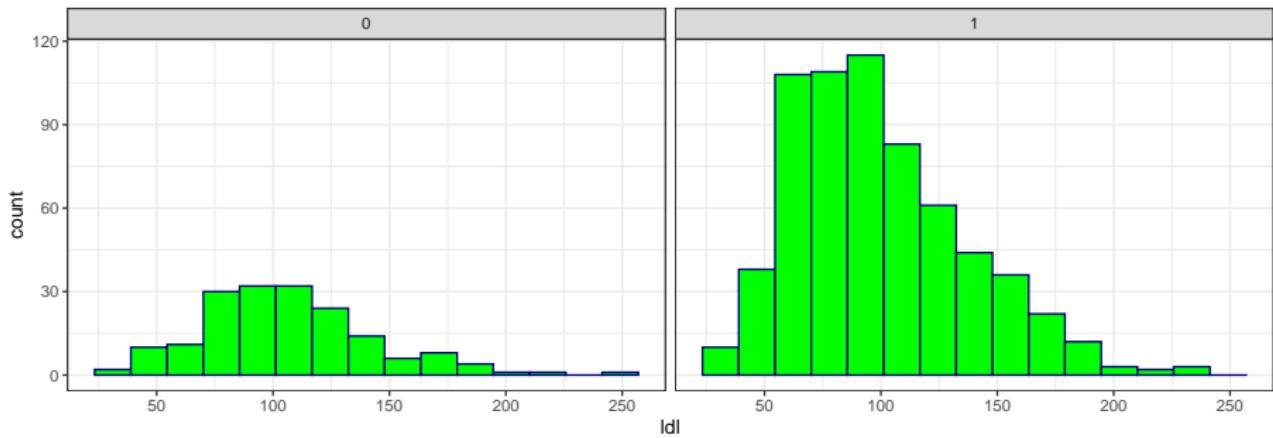
- How does this match up with our understanding of the distribution of the A1c data?
- Might the fairly large sample size ($n = 985$ non-missing) have something to do with these results?
- More on using regression models (and the bootstrap) to compare summaries is coming in Classes 13-15.

Comparing Two Distributions

Is LDL higher or lower among adults with diabetes who have a statin prescription?

```
ggplot(data = dm1000, aes(x = ldl)) +  
  geom_histogram(bins = 15, fill = "green", col = "navy") +  
  facet_wrap(~ statin)
```

Warning: Removed 178 rows containing non-finite values
(stat_bin).

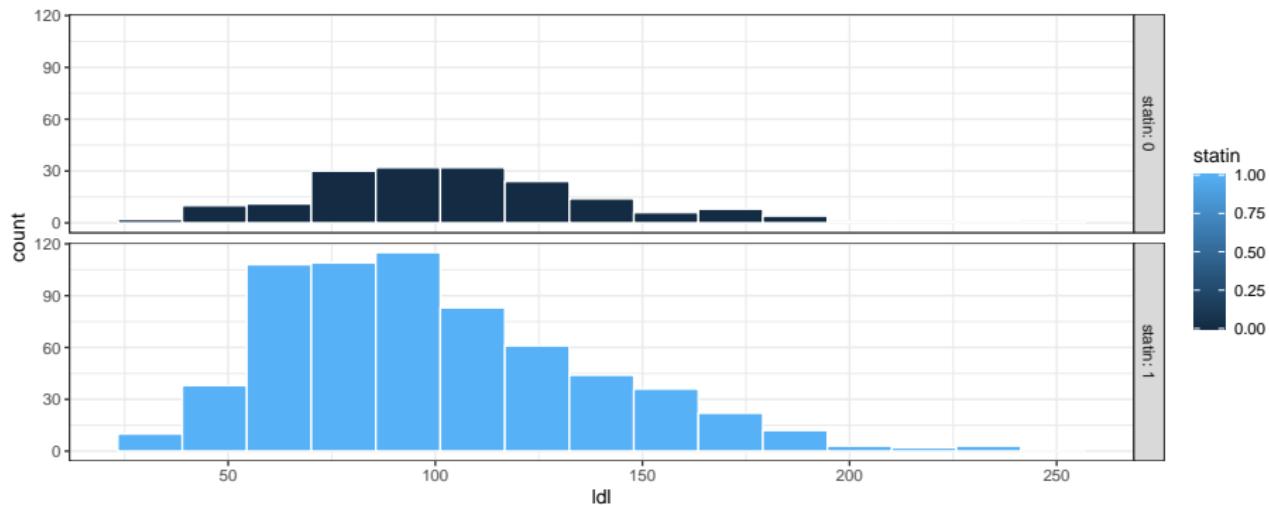


How might we improve this plot?

- ① Remove the warning about non-finite (missing) values.
- ② Place the histograms vertically to ease comparisons.
- ③ Fill the histograms differently for statin and no statin.
- ④ Augment the labels (0 and 1) to show they identify statin use.

LDL stratified by statin status (First Try)

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = ldl, fill = statin)) +
  geom_histogram(bins = 15, col = "white") +
  facet_grid(statin ~ ., labeller = "label_both")
```

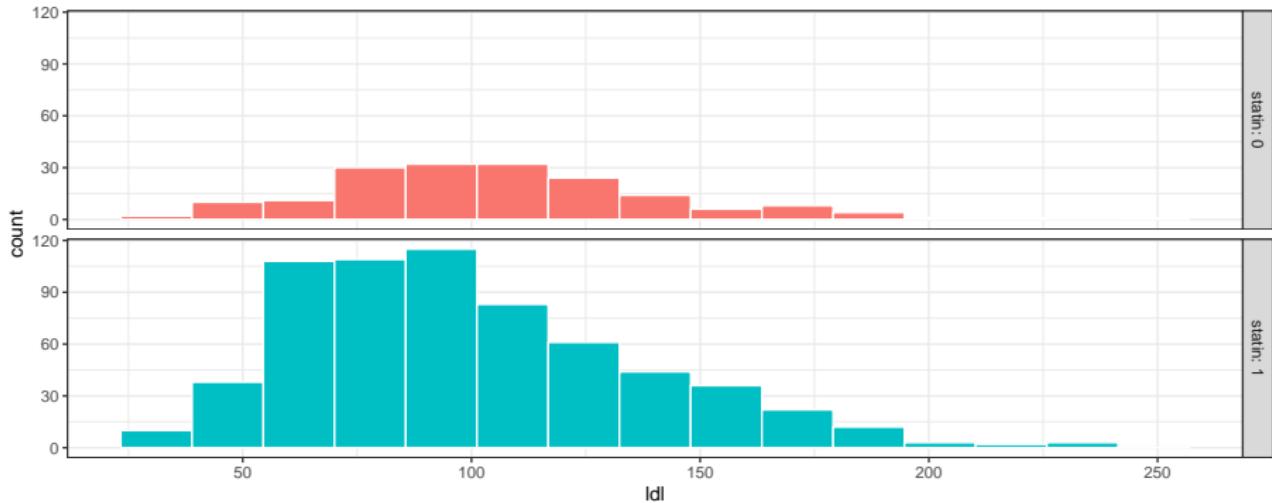


Problems with the previous plot

- ① Statin is actually a two-category variable (1 and 0 are just codes) but the legend is treating it as if it was a numeric variable.
- ② Do we actually need the legend (called a guide in R) or can we remove it?

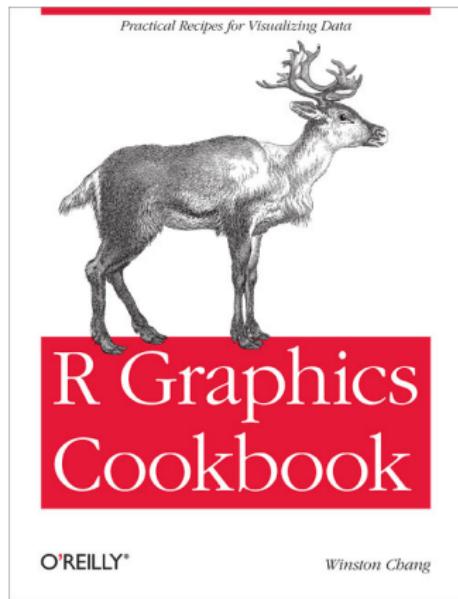
But statin is categorical? (Second Try)

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%  
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +  
  geom_histogram(bins = 15, col = "white") +  
  facet_grid(statin ~ ., labeller = "label_both") +  
  guides(fill = "none")
```



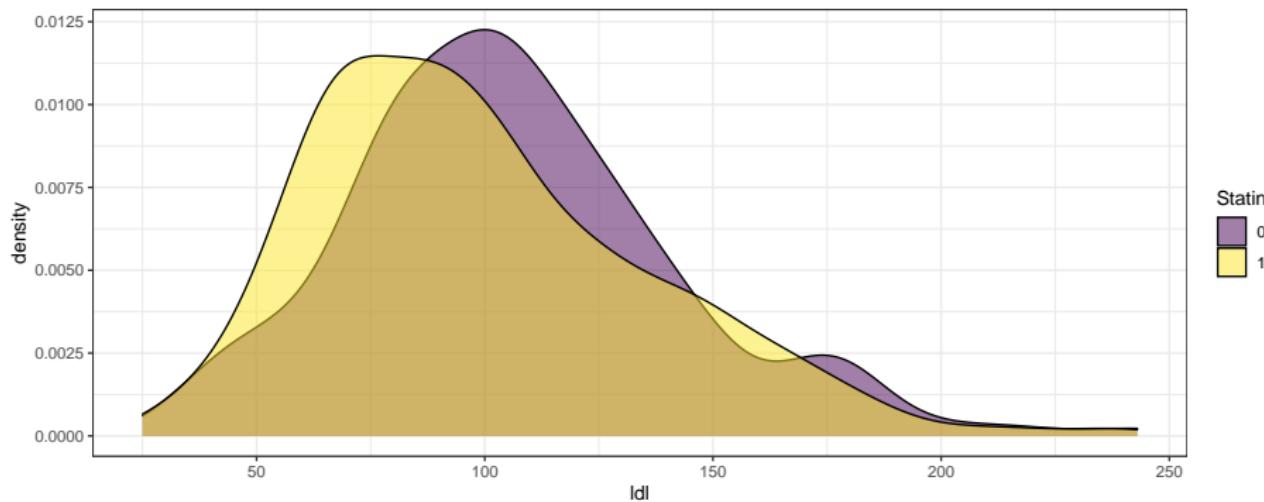
My Main Source for ggplot2 Visualization Recipes

<https://r-graphics.org/>



Comparison of densities (ignores relative frequency)

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +
  geom_density(alpha = 0.5) +
  scale_fill_viridis_d() +
  labs(fill = "Statin")
```



Numerical Summaries comparing Two Groups

```
dm1000 %>% filter(complete.cases(statin, ldl)) %>%
  group_by(statin) %>%
  summarize(n = n(), min = min(ldl), med = median(ldl),
            max = max(ldl), mean = mean(ldl),
            sd = sd(ldl)) %>%
  kable(digits = 2)
```

statin	n	min	med	max	mean	sd
0	176	34	102	243	106.22	36.05
1	646	25	92	241	99.22	37.21

- What is the difference in mean(LDL) between the two samples?

Using favstats to compare LDL by Statin group

```
dm1000 %$%
mosaic::favstats(ldl ~ statin)
```

	statin	min	Q1	median	Q3	max	mean	sd	n	
1		0	34	82	102	126	243	106.22159	36.04619	176
2		1	25	71	92	122	241	99.22136	37.20972	646
	missing									
1			66							
2			112							

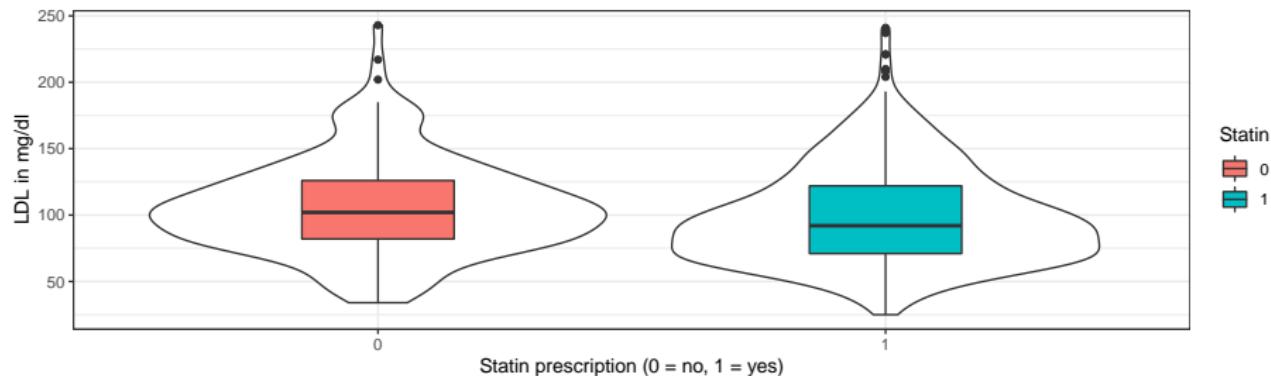
We would have obtained the same result with:

```
mosaic::favstats(ldl ~ statin, data = dm1000)
```

Comparison Boxplot with Violins (LDL and statin)

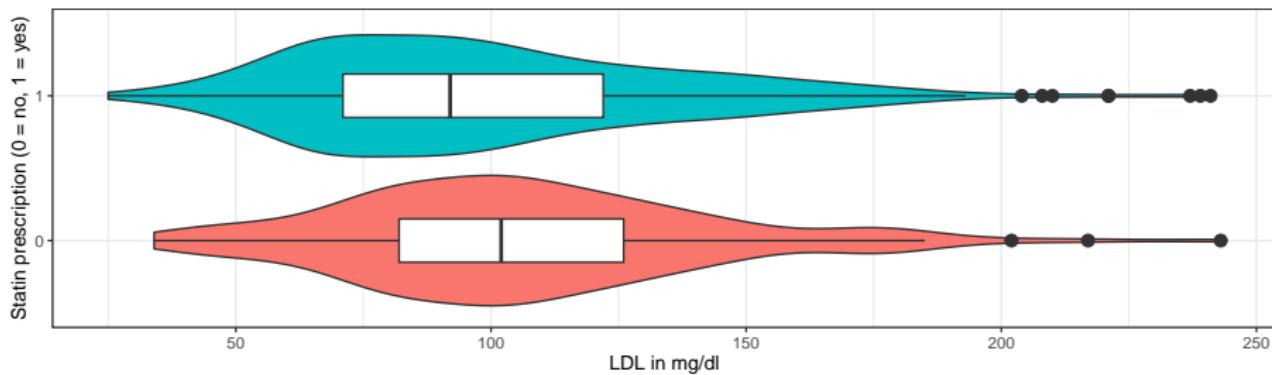
Here's a first attempt...

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = factor(statin), y = ldl)) +
  geom_violin() +
  geom_boxplot(aes(fill = factor(statin)), width = 0.3) +
  labs(x = "Statin prescription (0 = no, 1 = yes)",
       y = "LDL in mg/dl", fill = "Statin")
```



Try 2: Boxplot with Violins for LDL and statin

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = factor(statin), y = ldl)) +
  geom_violin(aes(fill = factor(statin))) +
  geom_boxplot(width = 0.3, outlier.size = 3) +
  coord_flip() +
  guides(fill = "none") +
  labs(x = "Statin prescription (0 = no, 1 = yes)",
       y = "LDL in mg/dl")
```



Setting Up Third Try

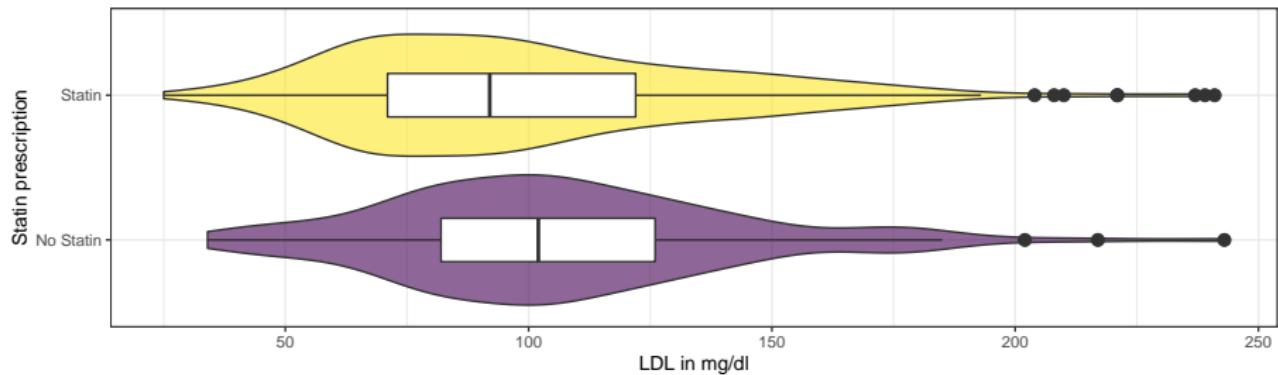
```
dm_for_boxplot <- dm1000 %>%
  filter(complete.cases(statin, ldl)) %>%
  mutate(statin_f = fct_recode(factor(statin),
                               "No Statin" = "0",
                               "Statin" = "1")) %>%
  select(subject, ldl, statin_f, statin)

head(dm_for_boxplot, 3) # print first three rows
```

```
# A tibble: 3 x 4
  subject    ldl statin_f  statin
  <chr>     <dbl> <fct>      <dbl>
1 M-0001     221 Statin       1
2 M-0002     116 No Statin   0
3 M-0003      52 Statin       1
```

Third Try on Boxplot for LDL by Statin Use

```
ggplot(data = dm_for_boxplot, aes(x = statin_f, y = ldl)) +  
  geom_violin(aes(fill = statin_f)) +  
  geom_boxplot(width = 0.3, outlier.size = 3) +  
  coord_flip() + guides(fill = "none") +  
  scale_fill_viridis_d(alpha = 0.6) +  
  labs(x = "Statin prescription",  
       y = "LDL in mg/dl")
```



95% confidence interval for difference between population mean LDL WITH statin and population mean LDL WITHOUT statin

If we are willing to assume that LDL follows a Normal distribution in each statin group, then we can use the following linear model with one predictor.

```
m2 <- lm(ldl ~ statin, data = dm1000)
tidy(m2, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, conf.low, conf.high) %>%
  kable(digits = 3)
```

term	estimate	conf.low	conf.high
(Intercept)	106.222	100.752	111.691
statin	-7.000	-13.170	-0.831

Alternative Approach to get same result

```
tt <- t.test(ldl ~ statin, data = dm1000,  
             var.equal = TRUE, conf.level = 0.95)  
tidy(tt) %>% select(estimate, conf.low, conf.high) %>%  
  kable(digits = 3)
```

estimate	conf.low	conf.high
7	0.831	13.17

95% confidence interval for difference between population mean LDL WITH statin and population mean LDL WITHOUT statin

If we are not willing to assume a Normal distribution for LDL in either the statin or the “no statin” group, then we could use a bootstrap approach.

```
source("data/Love-boost.R")
set.seed(20210914)
dm1000 %$% bootdif(y = ldl,
                      g = factor(statin),
                      conf.level = 0.95)
```

Mean Difference	0.025	0.975
-7.0002287	-12.9555578	-0.9043361

95% confidence intervals for $\mu_{NoStatin} - \mu_{Statin}$

Approach	Estimate	95% CI	Normality Assumption?
linear model	7.00	(0.83, 13.17)	Yes
bootstrap	7.00	(0.90, 12.96)	No

Assumptions these intervals share:

- random samples from the populations of interest
- independent samples (samples aren't paired or matched)

Additional assumptions for linear model:

- Normal distribution in each group (statin and "no statin")
- variance in each group (statin and "no statin") is equal

Next Time

- Visualizing Comparisons across 3+ batches
- Correlation and Scatterplots

431 Class 08

thomaselove.github.io/431

2021-09-16

Today's Agenda

- Building Visualizations to Compare Two Distributions
 - Confidence Intervals for a Difference between Means
- Building Visualizations to Compare > 2 Distributions

Today's R Packages

```
library(broom) # for tidying up output
library(janitor)
library(knitr)
library(magrittr)
library(naniar)
library(patchwork)
library(readxl) # new today, read in .xls or .xlsx files
library(tidyverse)

theme_set(theme_bw())
```

Today's Data

Today, we'll use an Excel file (.xls, rather than .csv) to import the dm1000 data.

```
dm1000 <- read_excel("data/dm_1000.xls") %>%  
  clean_names() %>%  
  mutate(across(where(is.character), as_factor)) %>%  
  mutate(subject = as.character(subject))
```

- There are also functions called `read_xls()` and `read_xlsx()` available in the `readxl` package.

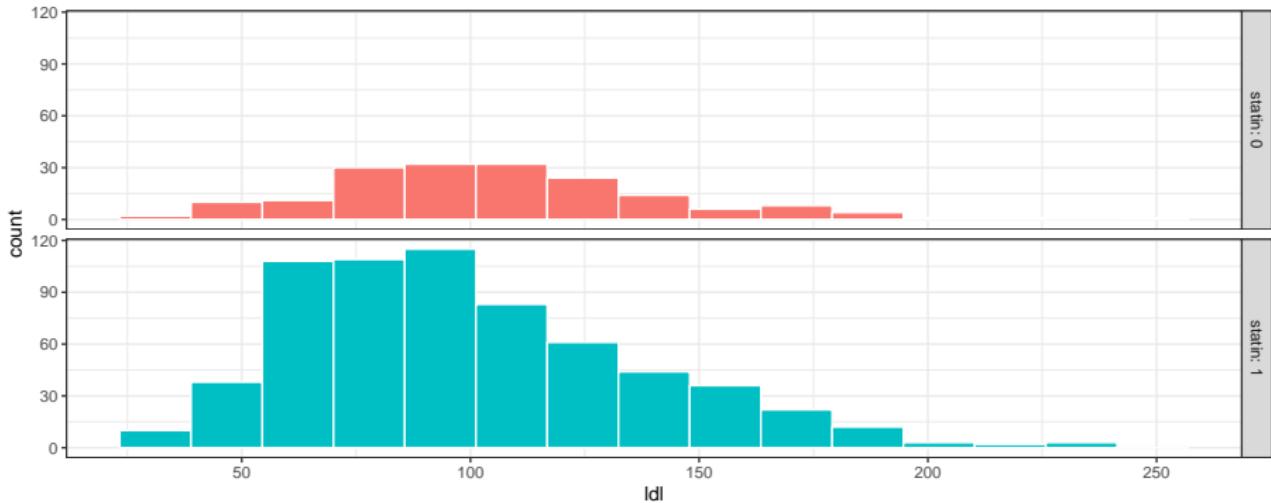
The dm1000 tibble

```
# A tibble: 1,000 x 17
  subject    age insurance n_income      ht      wt     sbp
  <chr>    <dbl> <fct>        <dbl>    <dbl>    <dbl>    <dbl>
1 M-0001      55 Medicaid     29853   1.63   103.    145
2 M-0002      52 Commercial   31248   1.75   112.    151
3 M-0003      69 Medicare     23362   1.65   74.9    127
4 M-0004      57 Medicaid     26033   1.63   81.4    125
5 M-0005      68 Medicare     85374   1.69   92.6    120
6 M-0006      56 Medicaid     31273   1.71   54.6    127
7 M-0007      54 Commercial   25445   1.68   81.6    114
8 M-0008      45 Medicare     67526   1.69   80.6    166
9 M-0009      61 Medicare     15203   1.91   86.7    111
10 M-0010     63 Medicaid     17628   1.86   123.    146
# ... with 990 more rows, and 10 more variables:
#   dbp <dbl>, a1c <dbl>, ldl <dbl>, tobacco <fct>,
#   statin <dbl>, eye_exam <dbl>,
#   race_ethnicity <fct>, sex <fct>, county <fct>,
```

Comparing Two Distributions

LDL cholesterol and statin prescription?

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%  
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +  
  geom_histogram(bins = 15, col = "white") +  
  facet_grid(statin ~ ., labeller = "label_both") +  
  guides(fill = "none")
```



Faceting

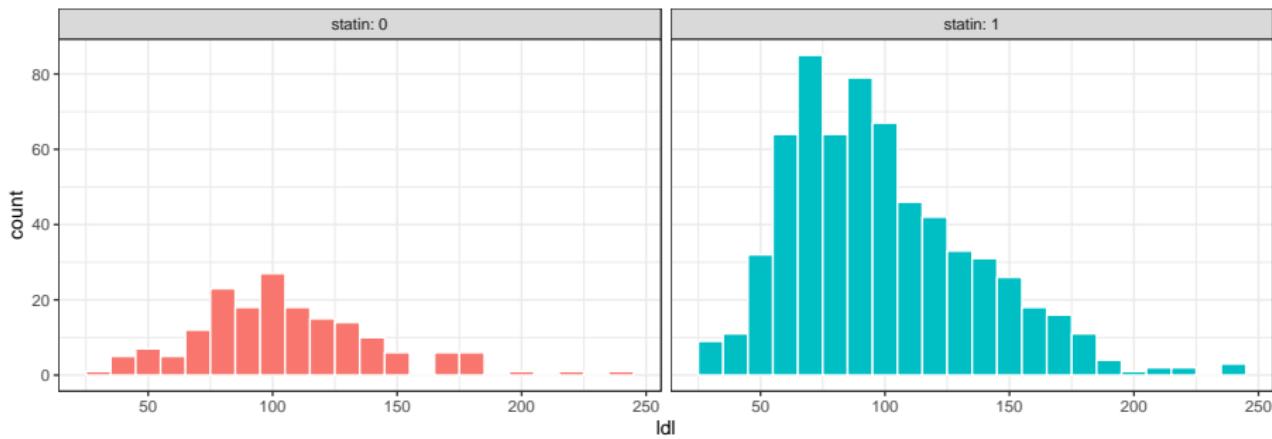
It's very useful to split data into groups and plot each group separately to make comparisons across the groups. We can then draw those subplots side by side.

We have two main tools: `facet_wrap()` and `facet_grid()`

- `facet_wrap(~ grp1)` to obtain plots within each `grp1` arranged into horizontal subpanels and wrapping around, like words on a page.
- `facet_grid(grp1 ~ .)` to obtain plots within each `grp1` arranged vertically (vertical subpanels)
- `facet_grid(grp1 ~ grp2)` to obtain plots within each combination of `grp1` and `grp2` with vertical and horizontal subpanels.

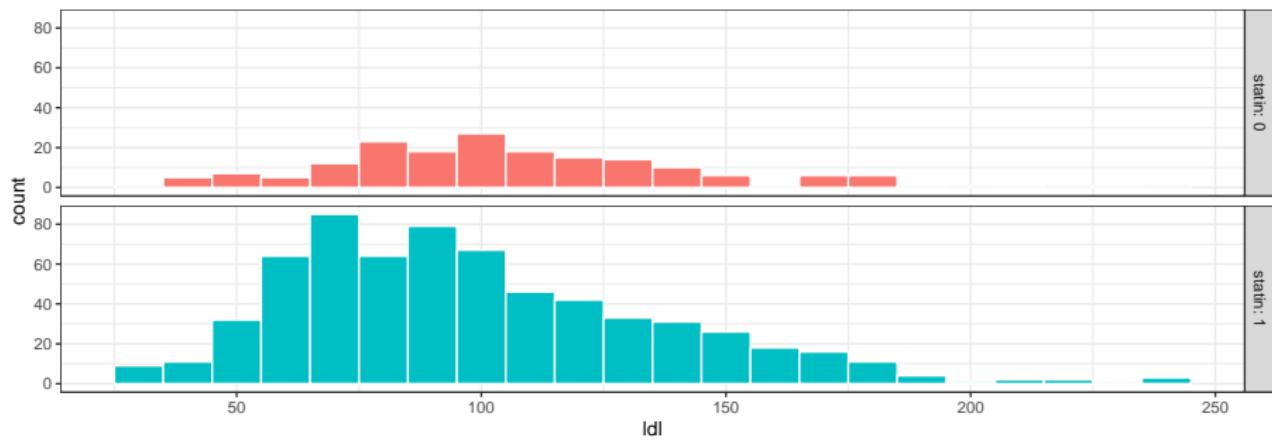
Using facet_wrap()

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%  
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +  
  geom_histogram(binwidth = 10, col = "white") +  
  facet_wrap(~ statin, labeller = "label_both") +  
  guides(fill = "none")
```



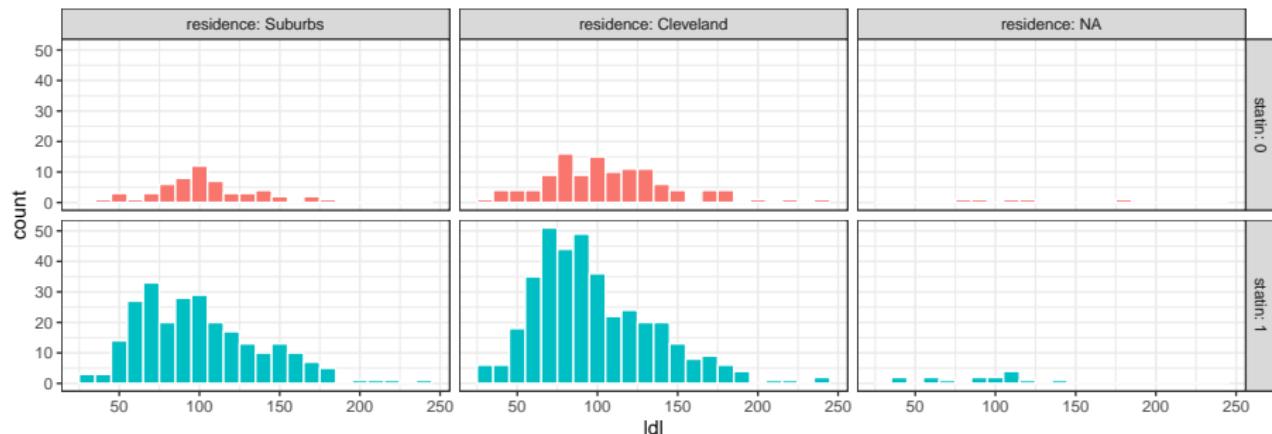
Using facet_grid()

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%  
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +  
  geom_histogram(binwidth = 10, col = "white") +  
  facet_grid(statin ~ ., labeller = "label_both") +  
  guides(fill = "none")
```



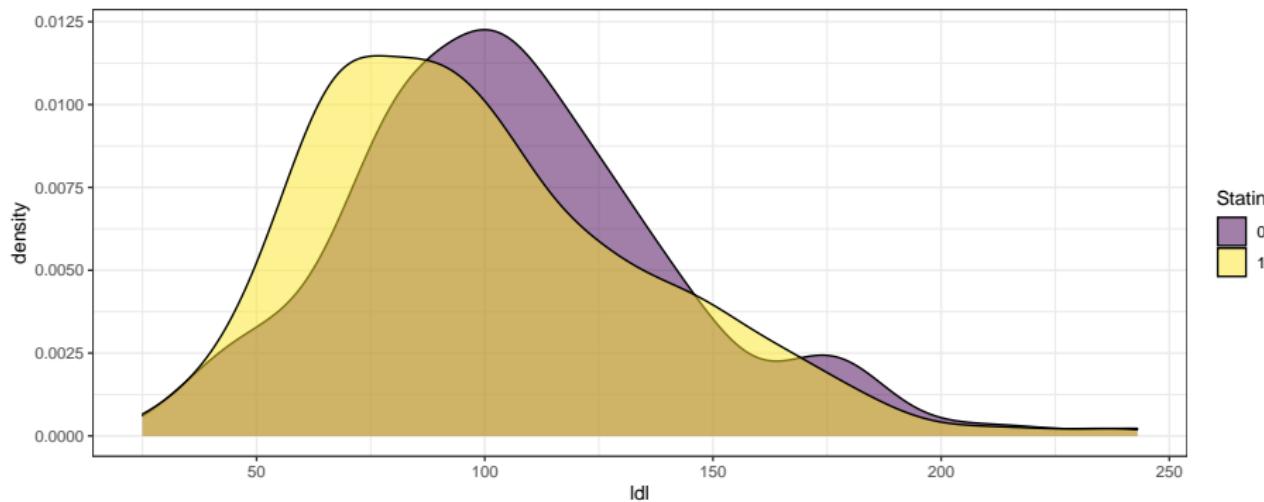
Using facet_grid() with two groupings

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%  
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +  
  geom_histogram(binwidth = 10, col = "white") +  
  facet_grid(residence ~ statin, labeller = "label_both") +  
  guides(fill = "none")
```



Comparison of densities (ignores relative frequency)

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = ldl, fill = factor(statin))) +
  geom_density(alpha = 0.5) +
  scale_fill_viridis_d() +
  labs(fill = "Statin")
```



Numerical Summaries comparing Two Groups

```
dm1000 %>% filter(complete.cases(statin, ldl)) %>%
  group_by(statin) %>%
  summarize(n = n(), min = min(ldl), med = median(ldl),
            max = max(ldl), mean = mean(ldl),
            sd = sd(ldl)) %>%
  kable(digits = 2)
```

statin	n	min	med	max	mean	sd
0	176	34	102	243	106.22	36.05
1	646	25	92	241	99.22	37.21

- What is the difference in mean(LDL) between the two samples?

Using favstats to compare LDL by Statin group

```
dm1000 %$%
mosaic::favstats(ldl ~ statin)
```

	statin	min	Q1	median	Q3	max	mean	sd	n	
1		0	34	82	102	126	243	106.22159	36.04619	176
2		1	25	71	92	122	241	99.22136	37.20972	646
	missing									
1			66							
2			112							

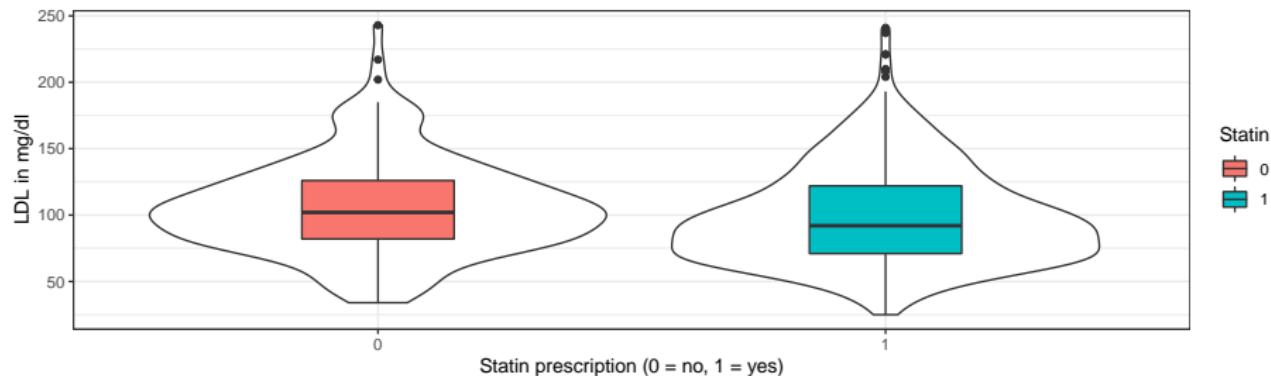
We would have obtained the same result with:

```
mosaic::favstats(ldl ~ statin, data = dm1000)
```

Comparison Boxplot with Violins (LDL and statin)

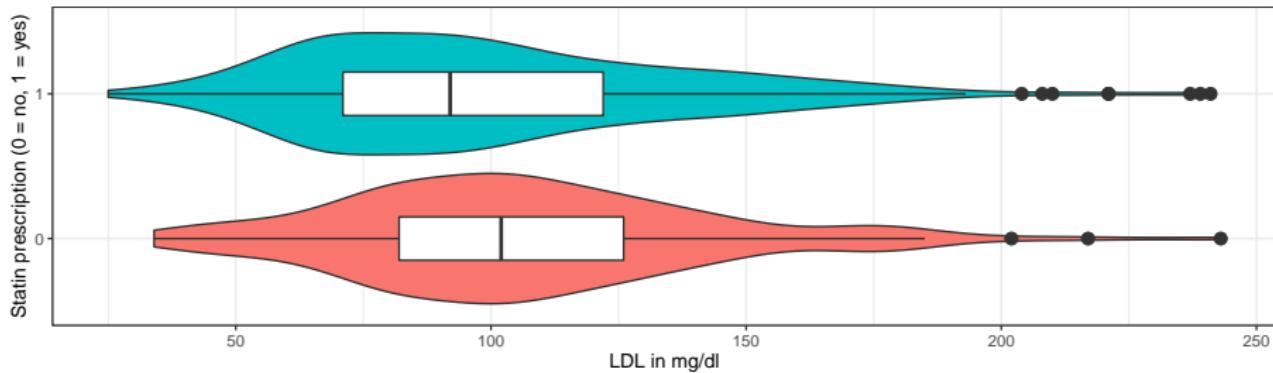
Here's a first attempt...

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = factor(statin), y = ldl)) +
  geom_violin() +
  geom_boxplot(aes(fill = factor(statin)), width = 0.3) +
  labs(x = "Statin prescription (0 = no, 1 = yes)",
       y = "LDL in mg/dl", fill = "Statin")
```



Try 2: Boxplot with Violins for LDL and statin

```
dm1000 %>% filter(complete.cases(ldl, statin)) %>%
  ggplot(data = ., aes(x = factor(statin), y = ldl)) +
  geom_violin(aes(fill = factor(statin))) +
  geom_boxplot(width = 0.3, outlier.size = 3) +
  coord_flip() +
  guides(fill = "none") +
  labs(x = "Statin prescription (0 = no, 1 = yes)",
       y = "LDL in mg/dl")
```



Setting Up Third Try

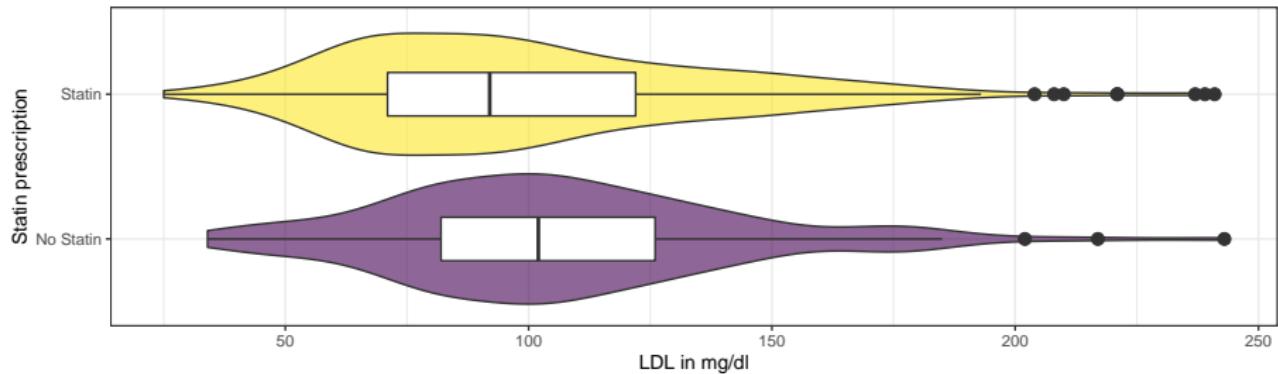
```
dm_for_boxplot <- dm1000 %>%
  filter(complete.cases(statin, ldl)) %>%
  mutate(statin_f = fct_recode(factor(statin),
                               "No Statin" = "0",
                               "Statin" = "1")) %>%
  select(subject, ldl, statin_f, statin)

head(dm_for_boxplot, 3) # print first three rows
```

```
# A tibble: 3 x 4
  subject    ldl statin_f statin
  <chr>     <dbl> <fct>      <dbl>
1 M-0001     221 Statin        1
2 M-0002     116 No Statin    0
3 M-0003      52 Statin        1
```

Third Try on Boxplot for LDL by Statin Use

```
ggplot(data = dm_for_boxplot, aes(x = statin_f, y = ldl)) +  
  geom_violin(aes(fill = statin_f)) +  
  geom_boxplot(width = 0.3, outlier.size = 3) +  
  coord_flip() + guides(fill = "none") +  
  scale_fill_viridis_d(alpha = 0.6) +  
  labs(x = "Statin prescription",  
       y = "LDL in mg/dl")
```



95% confidence interval for difference between population mean LDL WITH statin and population mean LDL WITHOUT statin

If we are willing to assume that LDL follows a Normal distribution in each statin group, then we can use the following linear model with one predictor.

```
m2 <- lm(ldl ~ statin, data = dm1000)
tidy(m2, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, conf.low, conf.high) %>%
  kable(digits = 3)
```

term	estimate	conf.low	conf.high
(Intercept)	106.222	100.752	111.691
statin	-7.000	-13.170	-0.831

Alternative Approach to get same result

```
tt <- t.test(ldl ~ statin, data = dm1000,  
             var.equal = TRUE, conf.level = 0.95)  
tidy(tt) %>% select(estimate, conf.low, conf.high) %>%  
  kable(digits = 3)
```

estimate	conf.low	conf.high
7	0.831	13.17

95% confidence interval for difference between population mean LDL WITH statin and population mean LDL WITHOUT statin

If we are not willing to assume a Normal distribution for LDL in either the statin or the “no statin” group, then we could use a bootstrap approach.

```
source("data/Love-boost.R")
set.seed(20210914)
dm1000 %$% bootdif(y = ldl,
                      g = factor(statin),
                      conf.level = 0.95)
```

Mean Difference	0.025	0.975
-7.0002287	-12.9555578	-0.9043361

The bootdif function (from Love-boost.R)

```
`bootdif` <-
function(y, g, conf.level=0.95, B.reps = 2000) {
  lowq = (1 - conf.level)/2
  g <- as.factor(g)
  a <- attr(Hmisc::smean.cl.boot(y[g==levels(g)[1]],
                                  B=B.reps, reps=TRUE), 'reps')
  b <- attr(Hmisc::smean.cl.boot(y[g==levels(g)[2]],
                                  B=B.reps, reps=TRUE), 'reps')
  meandif <- diff(tapply(y, g, mean, na.rm=TRUE))
  a.b <- quantile(b-a, c(lowq, 1-lowq))
  res <- c(meandif, a.b)
  names(res) <- c('Mean Difference', lowq, 1-lowq)
  res
}
```

95% confidence intervals for $\mu_{NoStatin} - \mu_{Statin}$

Approach	Estimate	95% CI	Normality Assumption?
linear model	7.00	(0.83, 13.17)	Yes
bootstrap	7.00	(0.90, 12.96)	No

Assumptions these intervals share:

- random samples from the populations of interest
- independent samples (samples aren't paired or matched)

Additional assumptions for linear model:

- Normal distribution in each group (statin and "no statin")
- variance in each group (statin and "no statin") is equal

Comparing Multiple (more than 2) Batches of Data

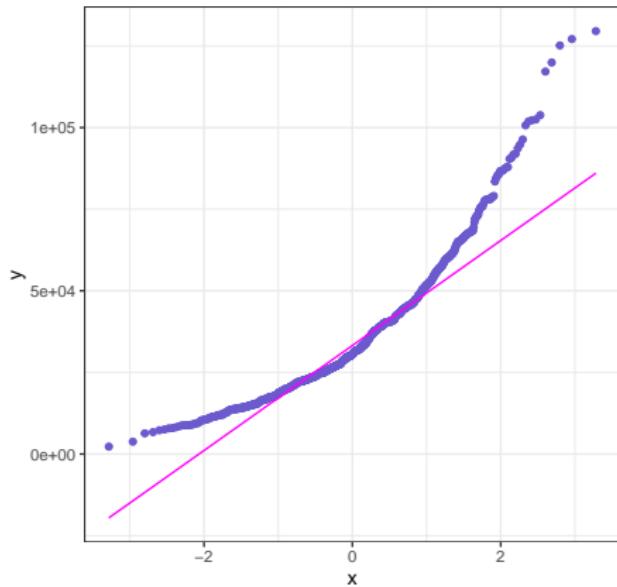
Stratify the subjects by primary insurance?

```
dm1000 %>% count(insurance) %>%  
  mutate(pct = 100*n/sum(n)) %>%  
  kable(digits = 1)
```

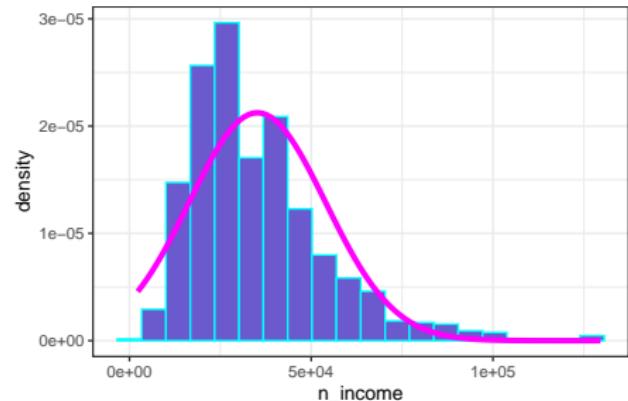
insurance	n	pct
Medicaid	330	33.0
Commercial	196	19.6
Medicare	432	43.2
Uninsured	42	4.2

Let's look at n_income

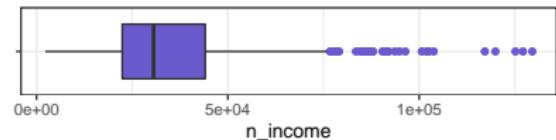
Normal Q-Q plot: dm1000 n_income



Density Function: dm1000 n_income



Boxplot: dm1000 n_income

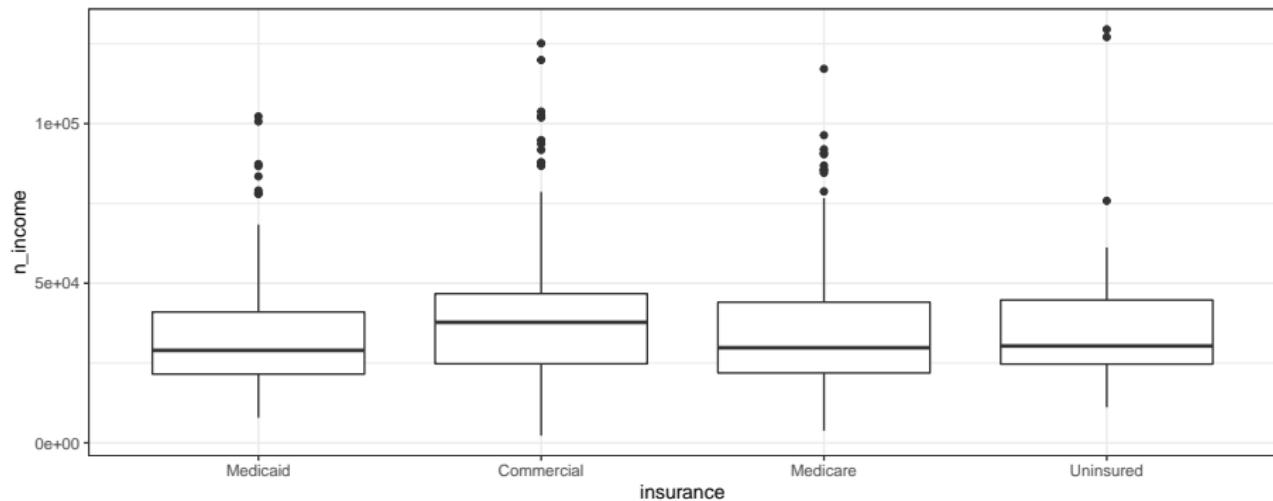


min	Q1	median	Q3	max	mean	sd	n	missing
2279	22393	30586	44085	129549	35178	18776	972	28

Compare n_income by insurance: Boxplot?

```
ggplot(dm1000, aes(x = insurance, y = n_income)) +  
  geom_boxplot()
```

Warning: Removed 28 rows containing non-finite values
(stat_boxplot).



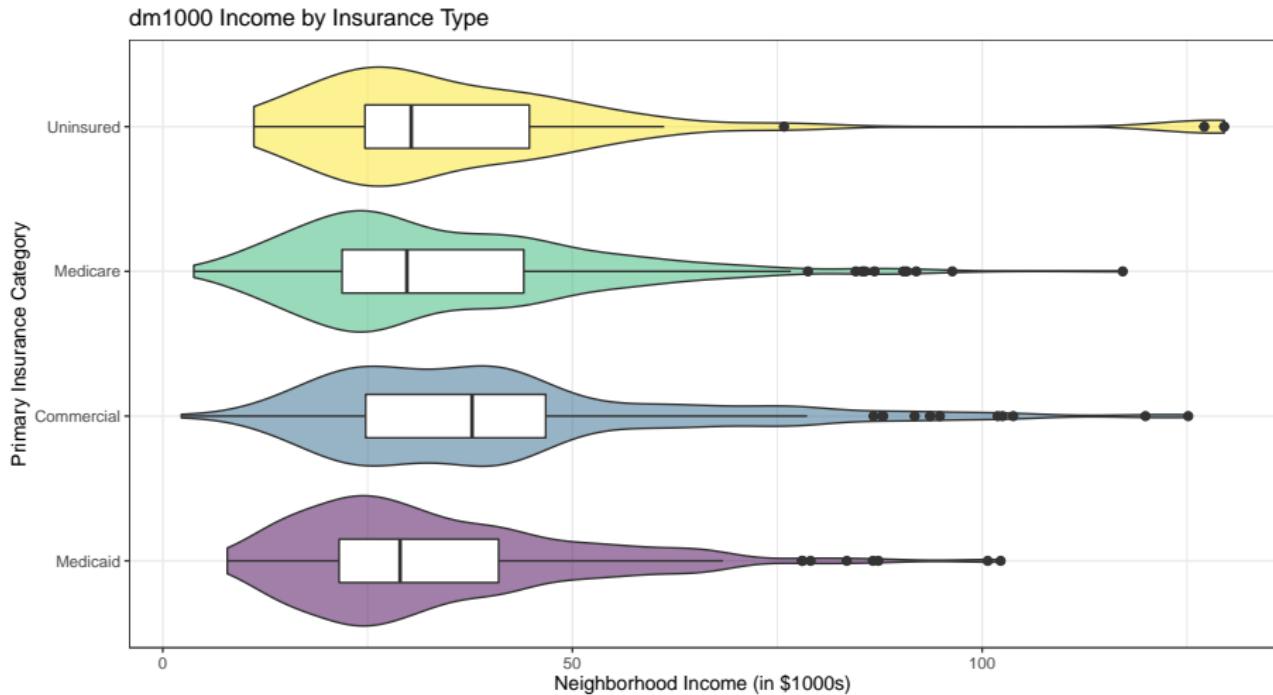
Build a better boxplot?

What am I doing in each line of this code?

```
dm1000 %>% filter(complete.cases(insurance, n_income)) %>%
  ggplot(data = ., aes(x = insurance, y = n_income/1000)) +
  geom_violin(aes(fill = insurance)) +
  geom_boxplot(width = 0.3, outlier.size = 2) +
  guides(fill = "none") +
  coord_flip() +
  scale_fill_viridis_d(alpha = 0.5) +
  labs(y = "Neighborhood Income (in $1000s)",
       x = "Primary Insurance Category",
       title = "dm1000 Income by Insurance Type")
```

- Result on next slide...

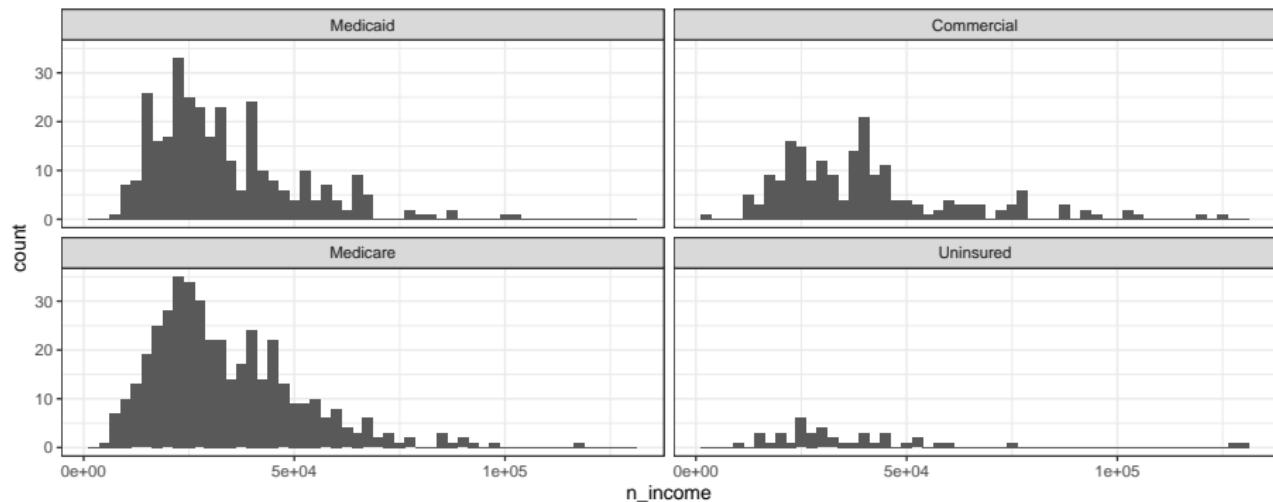
How does n_income vary by insurance in dm1000?



Faceted Histograms of n_income by insurance

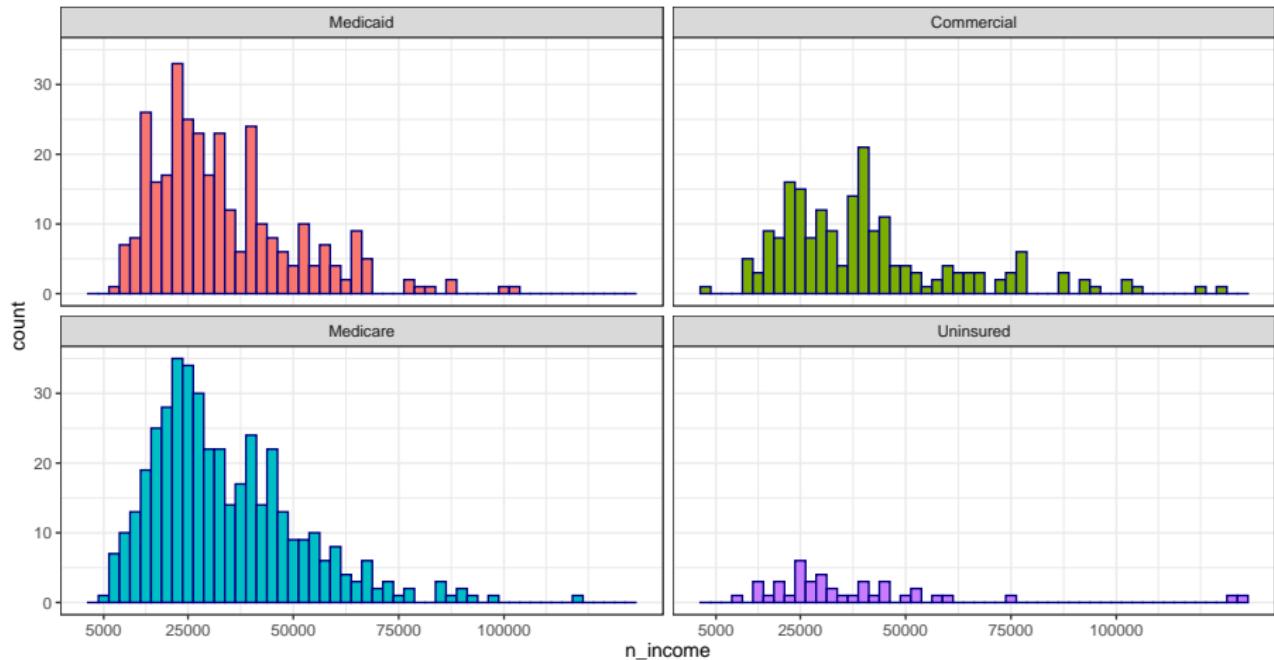
```
ggplot(data = dm1000, aes(x = n_income)) +  
  geom_histogram(binwidth = 2500) +  
  facet_wrap(~ insurance)
```

Warning: Removed 28 rows containing non-finite values
(stat_bin).



Improving the Histograms (result)

Neighborhood Income, in \$

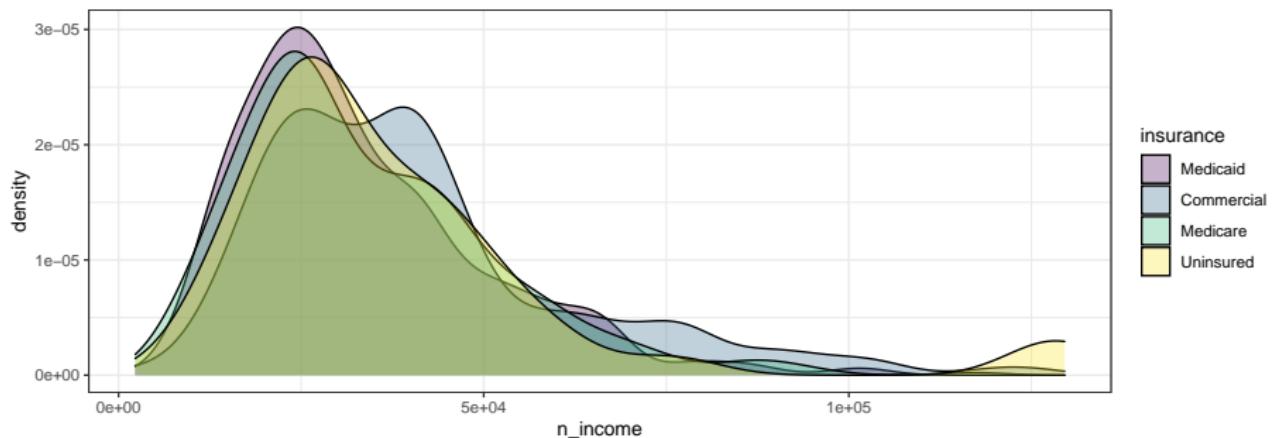


Improving the Histograms (code)

```
dm1000 %>% filter(complete.cases(n_income, insurance)) %>%  
ggplot(data = ., aes(x = n_income, fill = insurance)) +  
  geom_histogram(binwidth = 2500, col = "navy") +  
  scale_x_continuous(  
    breaks = c(5000, 25000, 50000, 75000, 100000)) +  
  guides(fill = "none") +  
  facet_wrap(~ insurance) +  
  labs(title = "Neighborhood Income, in $")
```

Comparing Densities of n_income by insurance

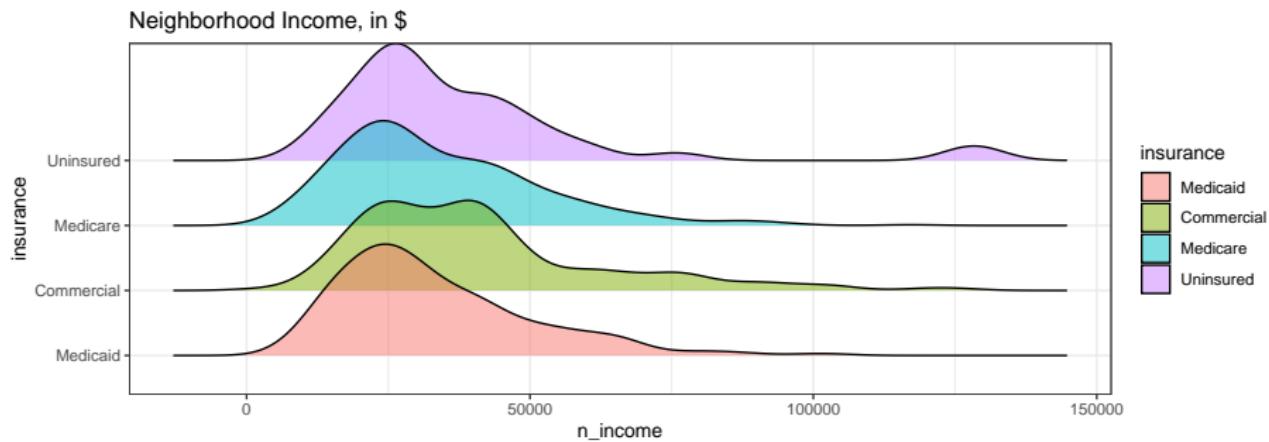
```
dm1000 %>% filter(complete.cases(n_income, insurance)) %>%
  ggplot(data = ., aes(x = n_income, fill = insurance)) +
  geom_density() +
  scale_fill_viridis_d(alpha = 0.3)
```



Using a Ridgeline Plot to Compare Densities

```
dm1000 %>% filter(complete.cases(n_income, insurance)) %>%  
  ggplot(data = ., aes(x = n_income, y = insurance,  
                      fill = insurance)) +  
  geom_density_ridges(alpha = 0.5) +  
  labs(title = "Neighborhood Income, in $")
```

Picking joint bandwidth of 5030



Sample Summaries of n_income

```
dm1000 %$% mosaic::favstats(n_income ~ insurance)
```

	insurance	min	Q1	median	Q3	max
1	Medicaid	7876	21528.5	28965.0	40986.00	102258
2	Commercial	2279	24767.0	37748.0	46736.00	125150
3	Medicare	3787	21883.5	29797.5	44062.75	117161
4	Uninsured	11121	24678.0	30328.0	44743.00	129549
		mean	sd	n	missing	
1	33150.74	16814.04	315		15	
2	40715.02	21713.97	194		2	
3	33883.50	17529.58	422		10	
4	37874.73	24962.68	41		1	

Comparing the Means of n_income

```
m1 <- lm(n_income ~ insurance, data = dm1000)

tidy(m1) %>% select(term, estimate) %>% kable(digits = 2)
```

term	estimate
(Intercept)	33150.74
insuranceCommercial	7564.28
insuranceMedicare	732.76
insuranceUninsured	4724.00

- What is m1's estimated n_income for each of the insurance groups?

m1: Estimated n_income for each insurance

$$\begin{aligned}n_income = & 33150.74 + 7564.28 \text{ Commercial} \\& + 732.76 \text{ Medicare} + 4724.00 \text{ Uninsured}\end{aligned}$$

Insurance	Estimated n_income
Commercial	$33150.74 + 7564.28 = 40715.02$
Medicare	$33150.74 + 732.76 = 33883.50$
Uninsured	$33150.74 + 4724.00 = 37874.74$
Medicaid	33150.74

m1: Estimated n_income for each insurance

n_income = 33150.74 + 7564.28 Commercial
+ 732.76 Medicare + 4724.00 Uninsured

Insurance	m1 est.	Sample mean(n_income)
Commercial	40715.0	40715.0
Medicare	33883.5	33883.5
Uninsured	33874.7	33874.7
Medicaid	33150.7	33150.7

Model m1 coefficients

```
tidy(m1, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, conf.low, conf.high) %>%
  kable(digits = 2)
```

term	estimate	conf.low	conf.high
(Intercept)	33150.74	31096.68	35204.80
insuranceCommercial	7564.28	4237.15	10891.42
insuranceMedicare	732.76	-1981.74	3447.27
insuranceUninsured	4724.00	-1328.66	10776.65

- 95% CI for pop. mean n_income among adults with Medicaid is (31097, 35205)

Model m1 coefficients

```
tidy(m1, conf.int = TRUE, conf.level = 0.95) %>%
  select(term, estimate, conf.low, conf.high) %>%
  kable(digits = 2)
```

term	estimate	conf.low	conf.high
(Intercept)	33150.74	31096.68	35204.80
insuranceCommercial	7564.28	4237.15	10891.42
insuranceMedicare	732.76	-1981.74	3447.27
insuranceUninsured	4724.00	-1328.66	10776.65

- 95% CI for Commercial - Medicaid is (4237, 10891)
- What about Medicare - Medicaid or Uninsured - Medicaid?

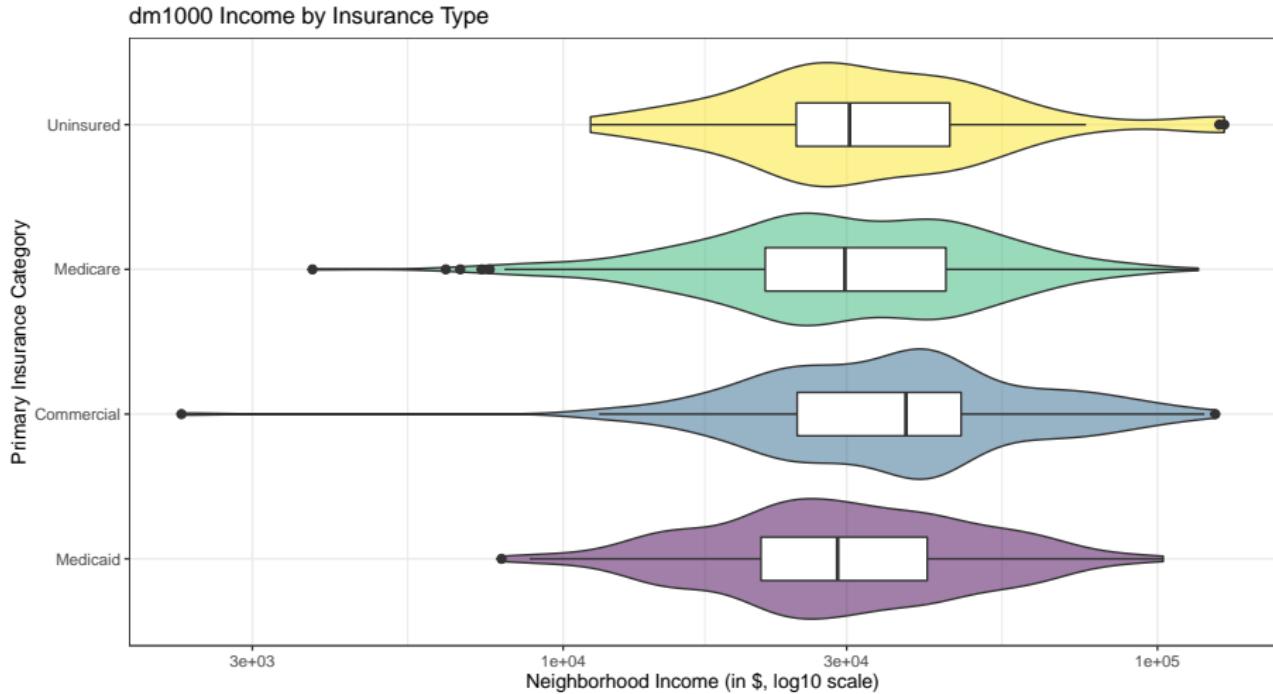
Comparing n_income across insurance groups

```
mosaic::favstats(n_income ~ insurance, data = dm1000)
```

	insurance	min	Q1	median	Q3	max
1	Medicaid	7876	21528.5	28965.0	40986.00	102258
2	Commercial	2279	24767.0	37748.0	46736.00	125150
3	Medicare	3787	21883.5	29797.5	44062.75	117161
4	Uninsured	11121	24678.0	30328.0	44743.00	129549
		mean	sd	n	missing	
1	Medicaid	33150.74	16814.04	315	15	
2	Commercial	40715.02	21713.97	194	2	
3	Medicare	33883.50	17529.58	422	10	
4	Uninsured	37874.73	24962.68	41	1	

- Does a comparison of means make sense here?
- Would it give us the same conclusions as comparing medians?

Replot on logarithmic (base 10) scale?

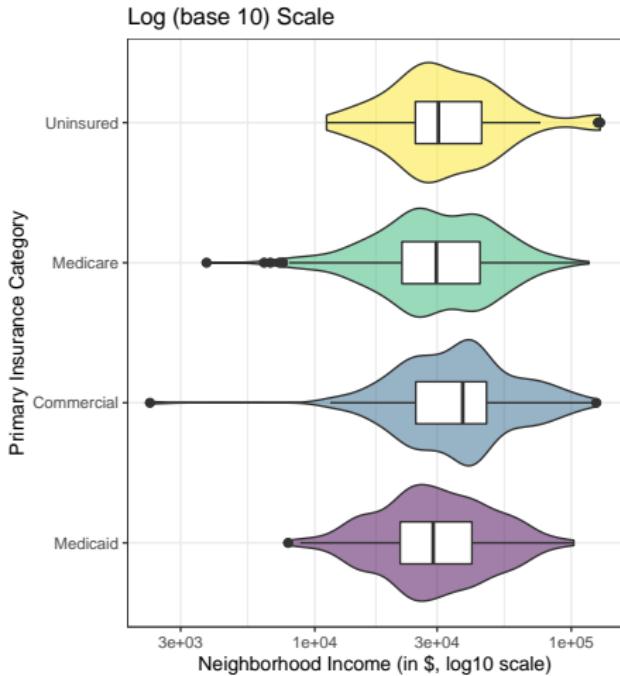
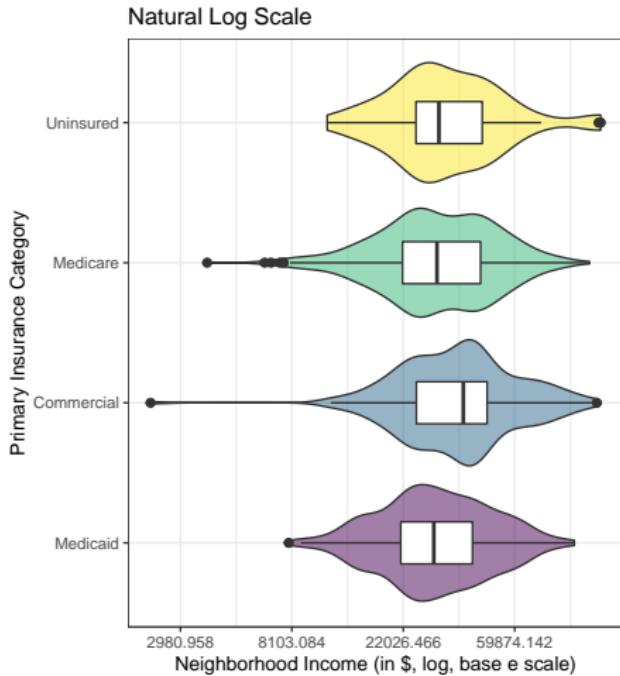


Code for plot on previous slide

```
dm1000 %>% filter(complete.cases(insurance, n_income)) %>%  
  ggplot(data = ., aes(x = insurance, y = n_income)) +  
  geom_violin(aes(fill = insurance)) +  
  geom_boxplot(width = 0.3, outlier.size = 2) +  
  guides(fill = "none") +  
  coord_flip() +  
  scale_fill_viridis_d(alpha = 0.5) +  
  scale_y_continuous(trans = "log10") +  
  labs(y = "Neighborhood Income (in $, log10 scale)",  
       x = "Primary Insurance Category",  
       title = "dm1000 Income by Insurance Type")
```

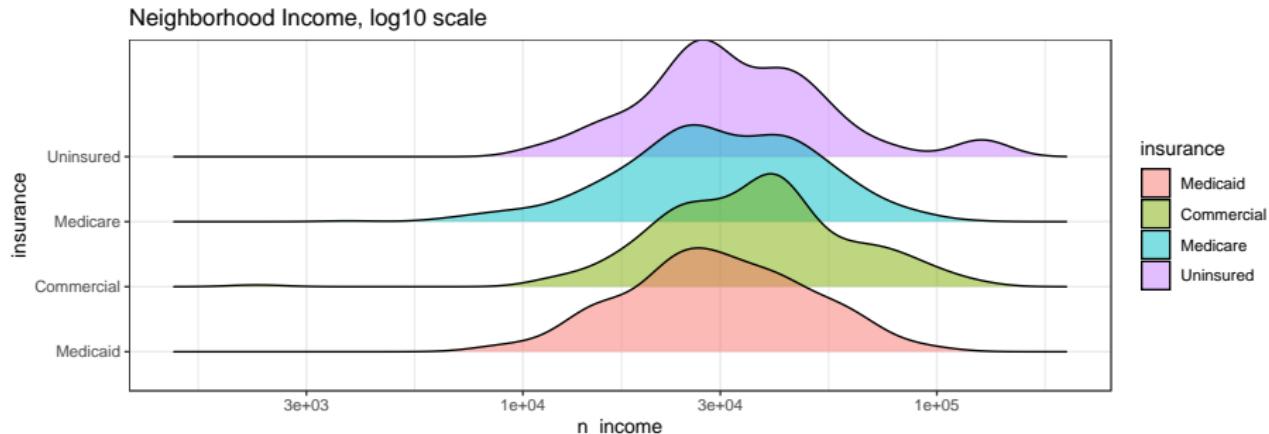
- Could also use `scale_y_log10()`

Does which logarithmic scale you pick matter?



Ridgeline Plot of Densities on Log Scale

```
dm1000 %>% filter(complete.cases(n_income, insurance)) %>%  
  ggplot(data = ., aes(x = n_income, y = insurance,  
                      fill = insurance)) +  
  geom_density_ridges(alpha = 0.5) +  
  scale_x_continuous(trans = "log10") +  
  labs(title = "Neighborhood Income, log10 scale")
```



Next Up

- Favorite Movies activity
- Correlation, Association and Scatterplots

431 Class 09

thomaselove.github.io/431

2021-09-21

Today's R Packages

```
library(broom) # for tidying up output
library(haven) # new today, importing files from SPSS
library(janitor)
library(knitr)
library(magrittr)
library(naniar)
library(patchwork)
library(readxl)
library(tidyverse)

theme_set(theme_bw())
```

Today's Data

Today, we'll use an SPSS file (.sav) to import the dm1000 data.

```
dm1000 <- read_sav("data/dm_1000.sav") %>%  
  clean_names() %>%  
  mutate(across(where(is.character), as_factor)) %>%  
  mutate(across(where(is.labelled), as_factor)) %>%  
  mutate(subject = as.character(subject))
```

- Note the next-to-last line in the code above, which is used to turn “labelled” variables (from SPSS) into factors in R.
- There are also functions called `read_sas()` and `read_xpt()` to read in SAS files, and `read_dta()` to read in Stata .dta files, available in the `haven` package.

The dm1000 tibble

```
# A tibble: 1,000 x 17
  subject    sbp    dbp insurance      age n_income     ht
  <chr>   <dbl>  <dbl>   <fct>      <dbl>    <dbl>  <dbl>
1 M-0001     145     70 Medicaid      55    29853  1.63
2 M-0002     151     77 Commercial    52    31248  1.75
3 M-0003     127     73 Medicare      69    23362  1.65
4 M-0004     125     74 Medicaid      57    26033  1.63
5 M-0005     120     73 Medicare      68    85374  1.69
6 M-0006     127     75 Medicaid      56    31273  1.71
7 M-0007     114     81 Commercial    54    25445  1.68
8 M-0008     166     110 Medicare     45    67526  1.69
9 M-0009     111     77 Medicare      61    15203  1.91
10 M-0010    146     102 Medicaid     63    17628  1.86
# ... with 990 more rows, and 10 more variables:
#   wt <dbl>, a1c <dbl>, ldl <dbl>, tobacco <fct>,
#   statin <dbl>, eye_exam <dbl>,
#   race_ethnicity <fct>, sex <fct>, county <fct>,
```

Describing the association of sbp and dbp

Numerical Summaries of sbp and dbp

```
mosaic::favstats(~ sbp, data = dm1000)
```

min	Q1	median	Q3	max	mean	sd	n	missing
84	122	132	142	209	132.7746	17.95214	994	6

```
dm1000 %$% mosaic::favstats(~ dbp)
```

min	Q1	median	Q3	max	mean	sd	n	missing
41	66	75	82	137	74.46378	12.42027	994	6

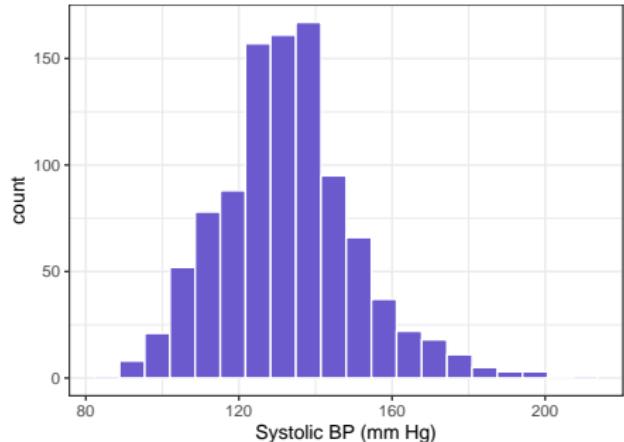
Are the same people missing sbp and dbp?

```
dm1000 %>% select(sbp, dbp) %>%  
  miss_case_summary()
```

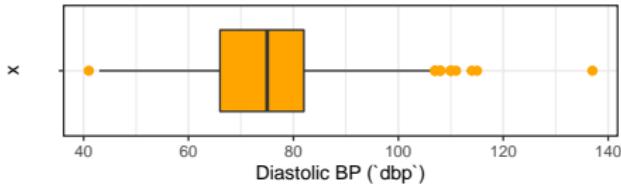
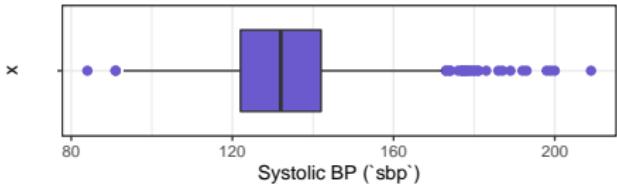
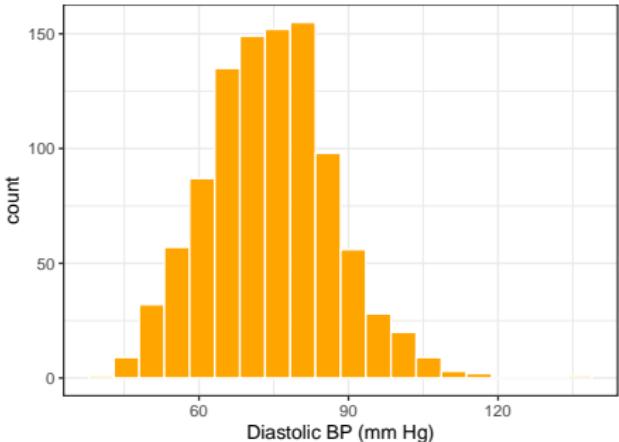
```
# A tibble: 1,000 x 3  
  case n_miss pct_miss  
  <int>   <int>    <dbl>  
1     107      2     100  
2     230      2     100  
3     284      2     100  
4     385      2     100  
5     440      2     100  
6     970      2     100  
7       1      0      0  
8       2      0      0  
9       3      0      0  
10      4      0      0  
# ... with 990 more rows
```

Distributions of sbp and dbp

Systolic BP in `dm1000`

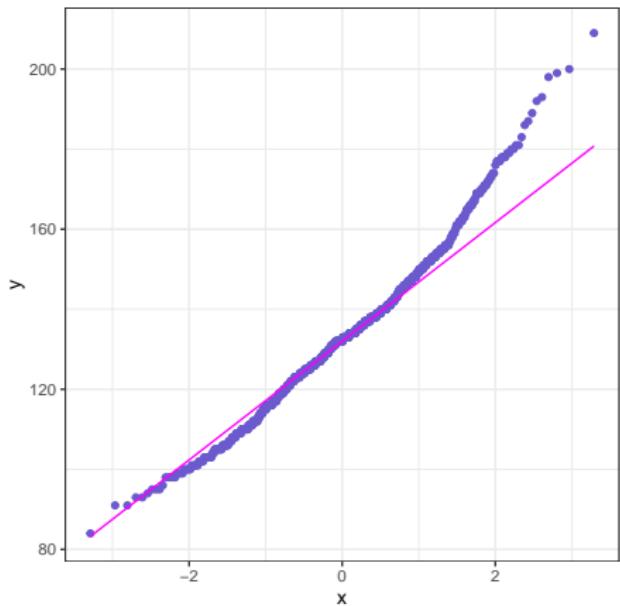


Diastolic BP in `dm1000`

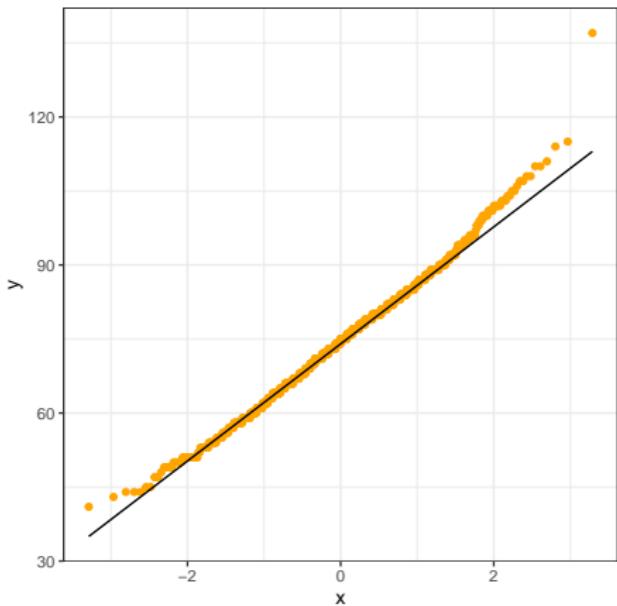


Normal model for sbp and dbp?

Normal Q–Q: dm1000 sbp



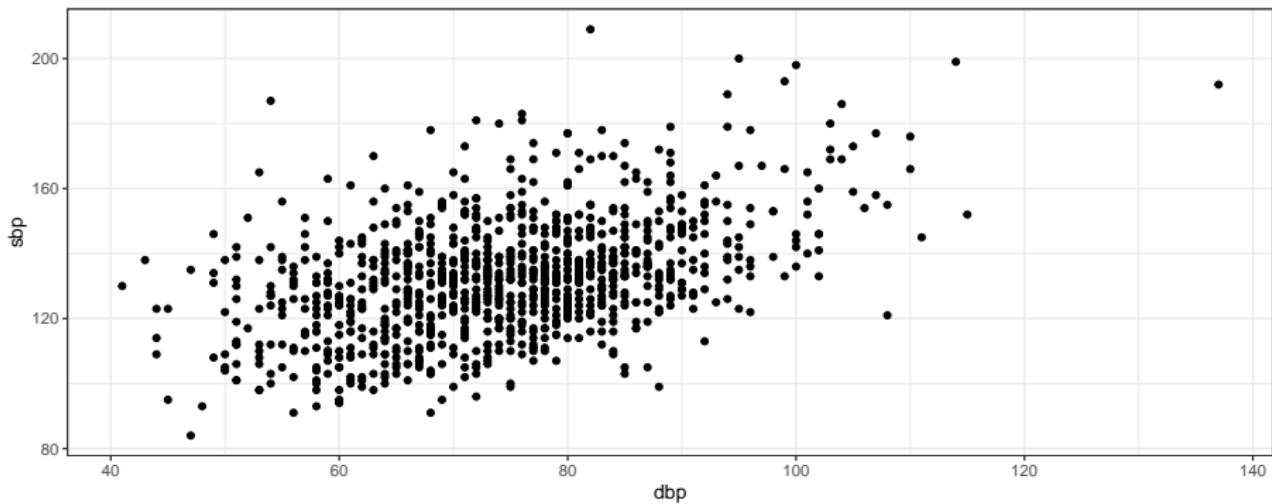
Normal Q–Q: dm1000 dbp



How closely associated are sbp and dbp?

```
ggplot(data = dm1000, aes(x = dbp, y = sbp)) +  
  geom_point()
```

Warning: Removed 6 rows containing missing values
(geom_point).

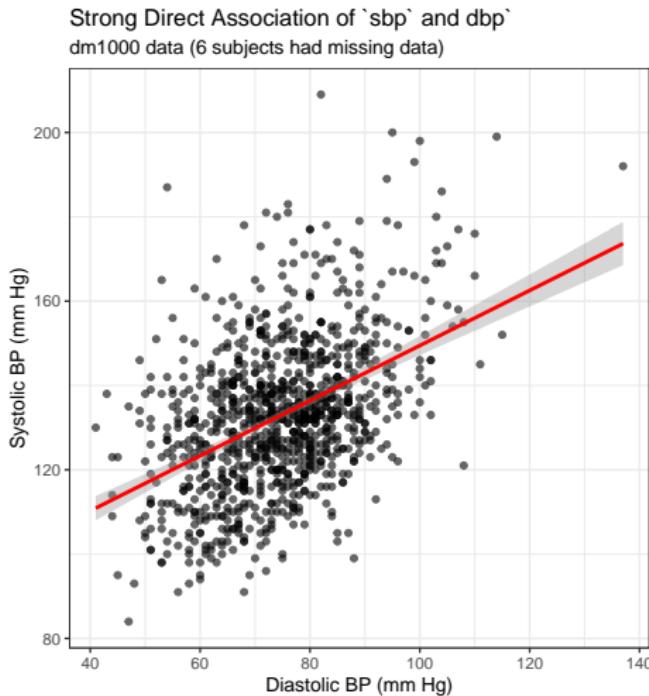


Improving the scatterplot (code)

```
dm1000 %>% filter(complete.cases(sbp, dbp)) %>%
ggplot(data = ., aes(x = dbp, y = sbp)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", col = "red",
              formula = y ~ x, se = TRUE) +
  theme(aspect.ratio = 1) +
  labs(x = "Diastolic BP (mm Hg)",
       y = "Systolic BP (mm Hg)",
       title = "Strong Direct Association of `sbp` and `dbp`",
       subtitle = "dm1000 data (6 subjects had missing data)")
```

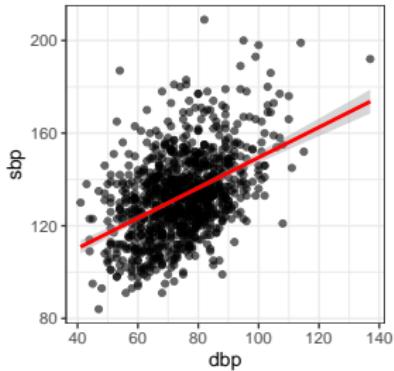
- What am I doing in these lines of code?

Higher DBP is associated with Higher SBP



- One point for each of the 994 subjects with known SBP and DBP...

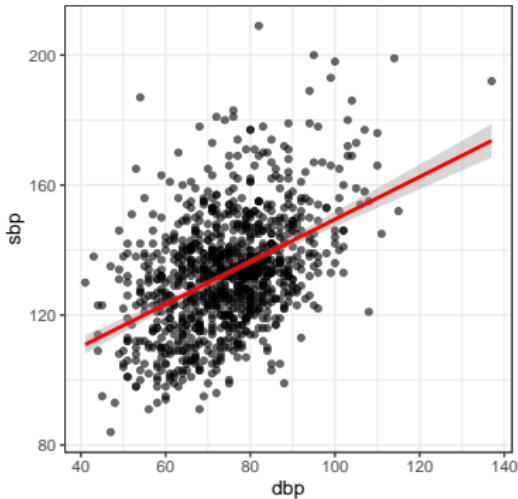
What are we looking for in this plot?



Is the association...

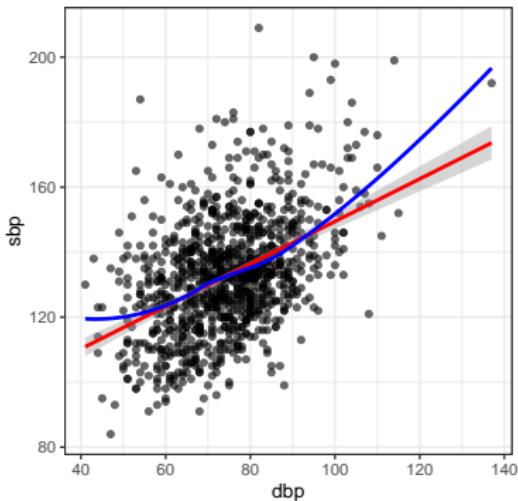
- ① **Linear or Non-Linear?** (is there a curve here?)
- ② **Direction?** (as X increases, what happens to Y?)
- ③ **Outliers?** (far away on X, or Y, or the combination?)
- ④ **Strength?** (points closely clustered together around a line?)

What might we conclude here?



- ➊ **Linear?:** The points roughly follow the straight line's path.
 - Do you see any clear signs of a curve?
 - Would adding a loess smooth help us?

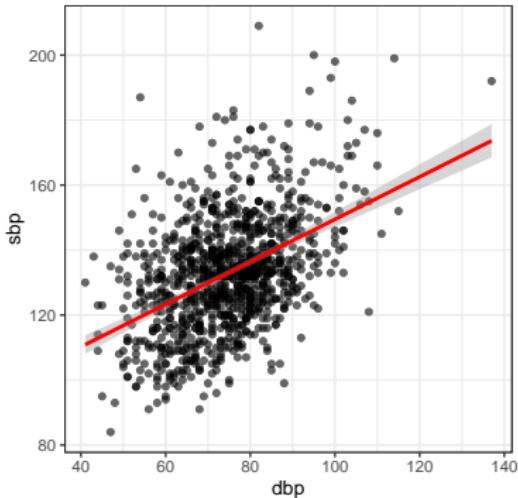
What might we conclude here?



① Linear?

- The loess smooth (in blue) suggests a potential curve
- Is it overreacting to the highly leveraged point ($\text{dbp} = 140$)?

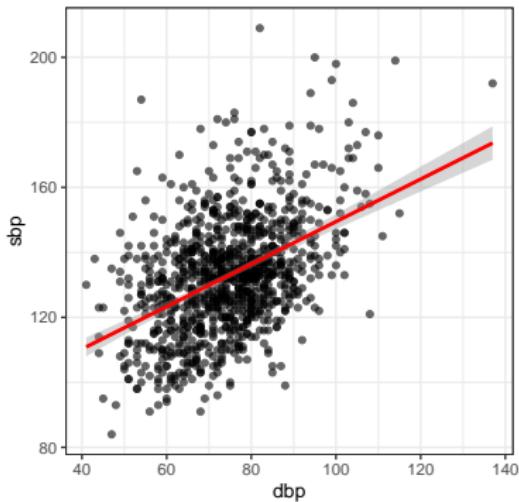
What might we conclude here?



② Direction?

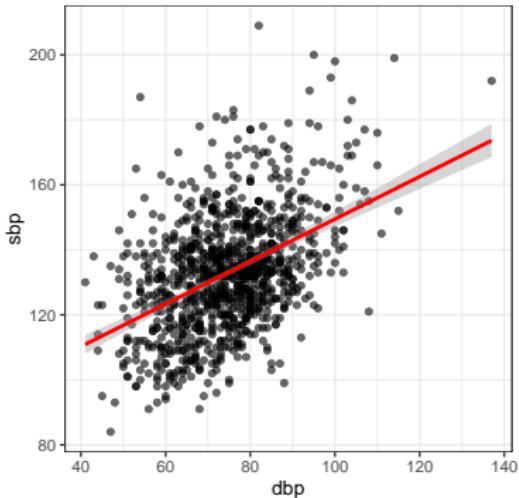
- As dbp increases, so does sbp, generally.
- Slope of the regression line is positive.

What might we conclude here?



- ① **Linear?**: No strong evidence of a meaningful curve.
- ② **Direction?**: As dbp increases, so does sbp, generally.
- ③ **Outliers?**: A few (out of 1000) worth another look, probably.

What might we conclude here?



④ **Strength?**: Does this association seem very strong?

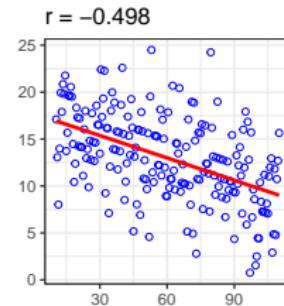
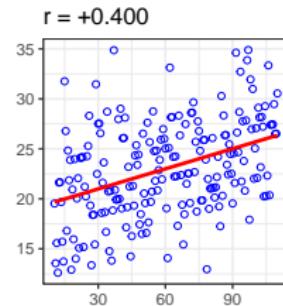
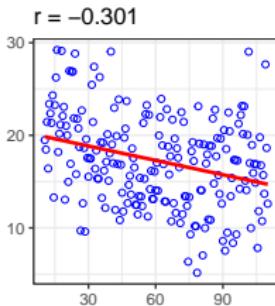
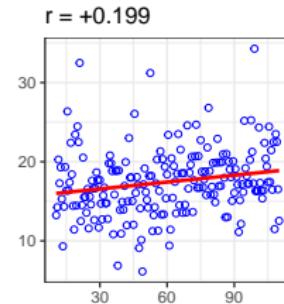
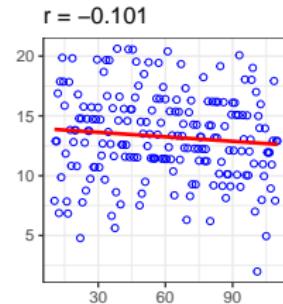
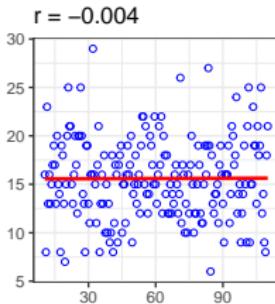
- sbp values associated with any particular dbp value range widely.
- If we know the dbp, that should help us make better predictions of sbp, but how much better than if we didn't know dbp?
- What might the **correlation** of sbp and dbp might be?

Summarizing Strength with the Pearson Correlation

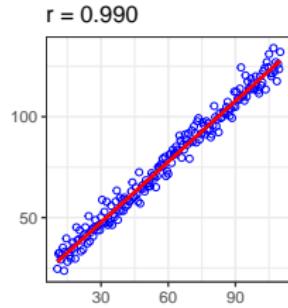
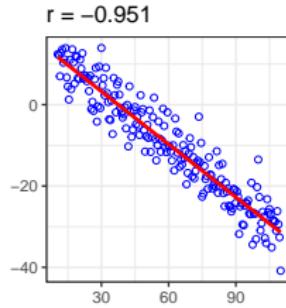
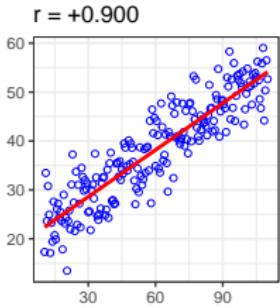
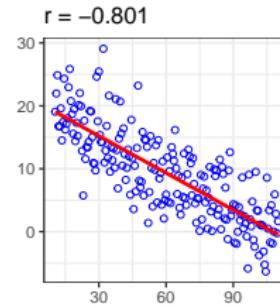
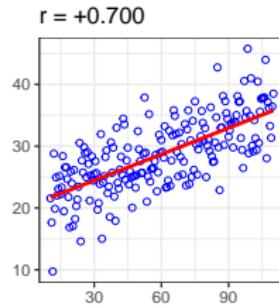
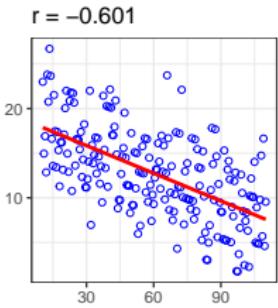
The Pearson correlation (abbreviated r) ranges from -1 to +1.

- The closer the absolute value of the correlation is to 1, the stronger a linear fit will be to the data, (in a limited sense).
- A strong positive correlation (near +1) will indicate a strong model with a positive slope.
- A strong negative correlation (near -1) will indicate a strong linear model with a negative slope.
- A weak correlation (near 0) will indicate a poor fit for a linear model, although a non-linear model may still fit the data quite well.

Gaining Some Insight into Correlation



Some Stronger Correlations



(Pearson) Correlation Coefficients for sbp and dbp

```
dm1000 %$% cor(sbp, dbp)
```

```
[1] NA
```

```
dm1000 %>%
  filter(complete.cases(sbp, dbp)) %$%
  cor(sbp, dbp)
```

```
[1] 0.4521072
```

```
dm1000 %$% cor(sbp, dbp, use = "complete.obs")
```

```
[1] 0.4521072
```

- What does this correlation imply about a linear fit to the data?

What line is being fit in our model m1?

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

```
m1 <- lm(sbp ~ dbp, data = dm1000)  
m1
```

Call:

```
lm(formula = sbp ~ dbp, data = dm1000)
```

Coefficients:

(Intercept)	dbp
84.1147	0.6535

Model m1 is **sbp = 84.11 + 0.65 dbp.**

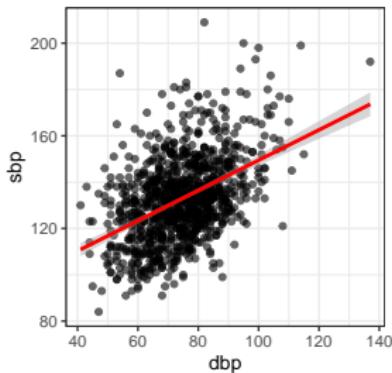
What does the slope mean?

$$\text{Weight} = 2.4 + \underline{0.3}(\text{height}) + \dots$$



if all other variables constant, we expect a 1 foot taller dragon to weigh 0.3 tons more, on average.

Linear Model m1: $sbp = 84.11 + 0.65 dbp$

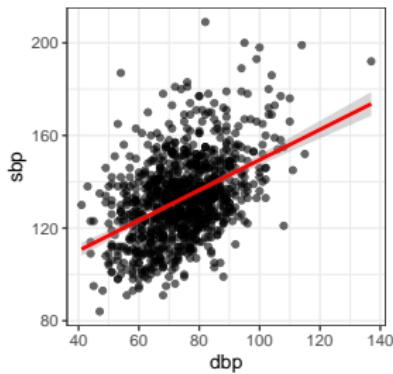


84.11 is the intercept = predicted value of sbp when dbp = 0.

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

- What does the model predict for sbp for a subject with dbp = 100?
- What if the subject had dbp = 99? 101? 110?

Linear Model m1: $sbp = 84.11 + 0.65 dbp$



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

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

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

Confidence Intervals for Regression Coefficients

We'll use the `tidy()` function from the `broom` package.

```
tidy(m1, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  kable(digits = 4)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	84.1147	3.0901	79.0271	89.2022
dbp	0.6535	0.0409	0.5861	0.7209

- How might we interpret the confidence interval for the slope of `dbp`?
 - Remember that the slope is the change in `sbp` per 1 unit change in `dbp` according to our model `m1`.
- How might we interpret the intercept term in model `m1`?

Obtaining R^2 and some Regression Fit Summaries

We'll use the `glance()` function, also from the `broom` package.

```
glance(m1) %>%
  select(nobs, r.squared, adj.r.squared, AIC, BIC) %>%
  kable(digits = c(0, 4, 4, 1, 1))
```

nobs	r.squared	adj.r.squared	AIC	BIC
994	0.2044	0.2036	8339.3	8354

- $n_{\text{obs}} = \#$ of observations actually used to fit the model
- $R^2 = \text{"r-squared"}$ is the square of the Pearson correlation r .
 - Recall we had $r = 0.4521$ for the association of `sbp` and `dbp`.
 - Squaring r , we get 0.2044.
- R^2 can be interpreted as the percentage of variation in `sbp` that `m1` accounts for with `dbp`

Interpreting R^2 and other Regression Summaries

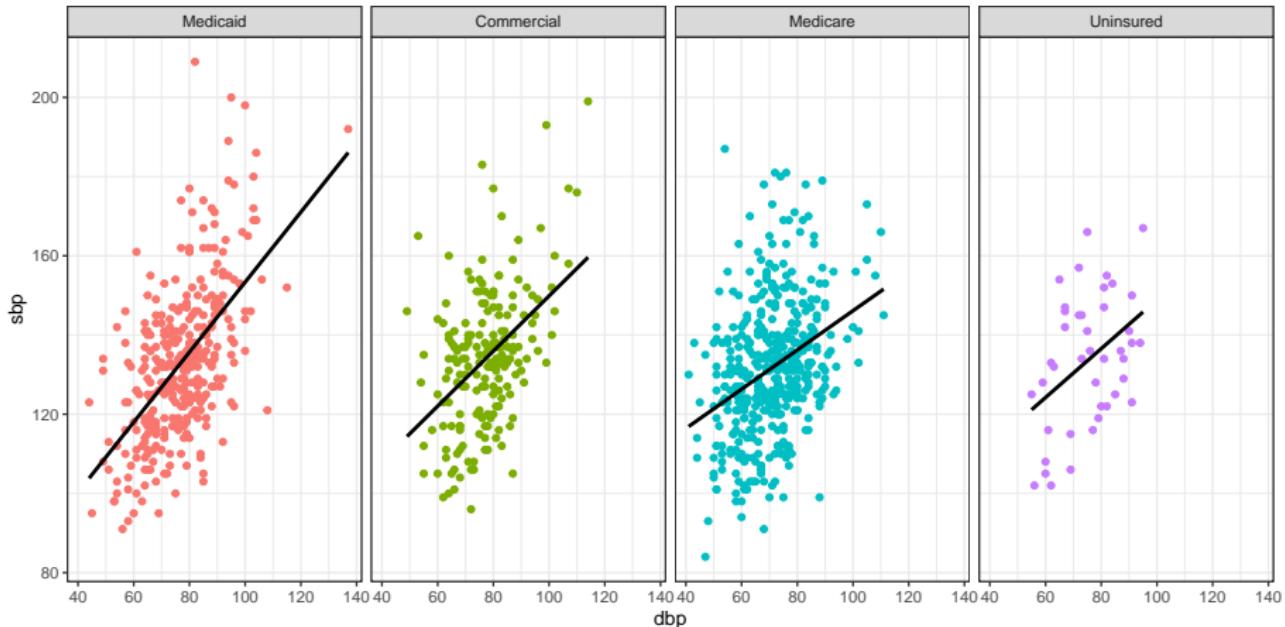
```
glance(m1) %>%
  select(nobs, r.squared, adj.r.squared, AIC, BIC) %>%
  kable(digits = c(0, 4, 4, 1, 1))
```

nobs	r.squared	adj.r.squared	AIC	BIC
994	0.2044	0.2036	8339.3	8354

- R^2 is also the proportionate reduction in error (as measured by sum of squared errors) in our predictions made using `m1` as compared to an “intercept only” regression model where we simply predict the mean of `sbp` for any subject, regardless of their `dbp`.
- Adjusted R^2 , AIC and BIC will become relevant as we compare multiple models for the same outcome.

sbp-dbp association in insurance subgroups?

- Different linear model for sbp using dbp in each insurance category.



Code for previous slide

```
dm1000 %>% filter(complete.cases(sbp, dbp, insurance)) %>%  
  ggplot(data = ., aes(x = dbp, y = sbp, col = insurance)) +  
  geom_point() +  
  geom_smooth(method = "lm", col = "black",  
              formula = y ~ x, se = FALSE) +  
  guides(col = "none") +  
  facet_wrap(~ insurance, nrow = 1)
```

Does sbp-dbp correlation vary by insurance?

```
dm1000 %>%
  filter(complete.cases(insurance, sbp, dbp)) %>%
  group_by(insurance) %>%
  summarize(n = n(), pearson_r = cor(sbp, dbp), r.squared = pe
kable(digits = 3)
```

insurance	n	pearson_r	r.squared
Medicaid	330	0.577	0.332
Commercial	193	0.452	0.204
Medicare	429	0.346	0.120
Uninsured	42	0.413	0.171

- How might we fit a linear model within each insurance type?
- Which of those models would have the largest R^2 ?

Model for subjects with Medicare insurance?

```
m2_medicare <- dm1000 %>%
  filter(insurance == "Medicare") %>%
  filter(complete.cases(sbp, dbp)) %$%
  lm(sbp ~ dbp)

tidy(m2_medicare, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, conf.low, conf.high) %>%
  kable(digits = 3)
```

term	estimate	conf.low	conf.high
(Intercept)	96.589	88.889	104.290
dbp	0.495	0.388	0.603

Glancing at the Medicare-Only Model

```
glance(m2_medicare) %>%  
  select(r.squared, nobs)
```

```
# A tibble: 1 x 2  
  r.squared nobs  
    <dbl> <int>  
1     0.120     429
```

Model including both dbp and insurance?

```
m3 <-  
dm1000 %>%  
filter(complete.cases(sbp, dbp, insurance)) %$%  
lm(sbp ~ dbp * insurance)  
  
glance(m3) %>% select(nobs, r.squared, adj.r.squared) %>%  
kable(digits = c(0, 3, 3))
```

nobs	r.squared	adj.r.squared
994	0.222	0.217

Coefficients of Model m3

```
tidy(m3) %>% select(term, estimate, std.error) %>%
  kable(digits = 3)
```

term	estimate	std.error
(Intercept)	64.948	5.395
dbp	0.884	0.069
insuranceCommercial	15.337	9.613
insuranceMedicare	31.641	7.107
insuranceUninsured	22.247	17.604
dbp:insuranceCommercial	-0.188	0.123
dbp:insuranceMedicare	-0.389	0.094
dbp:insuranceUninsured	-0.267	0.231

- What does this model imply for Medicare subjects?

Understanding the m3 model

Model m3 predicts sbp using

$$\begin{aligned} & 64.948 + 0.884 \text{ `dbp'} \\ & + 31.641 \text{ Medicare} - 0.389 \text{ `dbp'} * \text{Medicare} \\ & + 15.337 \text{ Commer.} - 0.188 \text{ `dbp'} * \text{Commer.} \\ & + 22.247 \text{ Medicaid} - 0.267 \text{ `dbp'} * \text{Medicaid} \end{aligned}$$

- ① What is the resulting equation for a Medicare subject?

Understanding the m3 model

Model m3 predicts sbp using

$$\begin{aligned} & 64.948 + 0.884 \text{ `dbp'} \\ & + 31.641 \text{ Medicare} - 0.389 \text{ `dbp'} * \text{Medicare} \\ & + 15.337 \text{ Commer.} - 0.188 \text{ `dbp'} * \text{Commer.} \\ & + 22.247 \text{ Medicaid} - 0.267 \text{ `dbp'} * \text{Medicaid} \end{aligned}$$

What is the resulting equation for a Medicare subject?

$$\begin{aligned} \text{sbp} &= (64.948 + 31.641) + (0.884 - 0.389) * \text{dbp} \\ \text{sbp} &= 96.589 + 0.495 \text{ dbp} \end{aligned}$$

- This matches the result we obtained running the sbp on dbp regression for the Medicare subjects alone in model m2_medicare.

Understanding the m3 model

Again, model m3 predicts sbp using

$$\begin{aligned} & 64.948 + 0.884 \text{ `dbp'} \\ & + 31.641 \text{ Medicare} - 0.389 \text{ `dbp'} * \text{Medicare} \\ & + 15.337 \text{ Commer.} - 0.188 \text{ `dbp'} * \text{Commer.} \\ & + 22.247 \text{ Medicaid} - 0.267 \text{ `dbp'} * \text{Medicaid} \end{aligned}$$

Insurance	Predicted sbp
Medicare	$96.589 + 0.495 \text{ dbp}$
Commercial	$(64.948 + 15.337) + (0.884 - 0.188) \text{ dbp}$
Commercial	or, $80.285 + 0.696 \text{ dbp}$
Medicaid	$87.195 + 0.617 \text{ dbp}$
Uninsured	$64.948 + 0.884 \text{ dbp}$

Which model shows better fit to the data?

```
g1 <- glance(m1) %>%
  mutate(m_name = "m1 (dbp only)")
g3 <- glance(m3) %>%
  mutate(m_name = "m3 (dbp * insurance)")

bind_rows(g1, g3) %>%
  select(m_name, nobs, r.squared, adj.r.squared, AIC, BIC) %>%
  kable(digits = c(0, 0, 3, 3, 0, 0))
```

m_name	nobs	r.squared	adj.r.squared	AIC	BIC
m1 (dbp only)	994	0.204	0.204	8339	8354
m3 (dbp * insurance)	994	0.222	0.217	8329	8373

- Model m3 has better R^2 , and adjusted R^2 ; better AIC, but worse BIC.
- IGNORING: regression assumptions, and predictions in new data...

Coming Up

- More with your favorite movies
- Associations between categorical variables

431 Class 10

thomaselove.github.io/431

2021-09-23

Today's Agenda

- ① Ingesting dm1000 data using R data set format (.Rds)
- ② Partitioning data into model training/test samples.
- ③ Augmenting a Scatterplot (labeling, size, color) and fitting a simple OLS (linear) model m1
- ④ Using summary() and extract_eq() on a regression model.
- ⑤ The broom package and tidy(), glance() and augment()
- ⑥ Calibrating your understanding of R-square a bit
- ⑦ Assessing Regression Assumptions with Residual Plots
- ⑧ Making Predictions into the Test Sample
- ⑨ Assessing Quality of Fit using the Test Sample with mean and maximum absolute prediction error and with RMSPE
- ⑩ Fitting a Bayesian Linear Model with default priors (m2)
- ⑪ Including Insurance without (m3) and with (m4) interaction with dbp in linear models

Today's Packages

```
library(broom)
library(equatiomatic) # new today
library(ggrepel) # sort of new today
library(glue) # sort of new today
library(janitor)
library(knitr)
library(magrittr)
library(patchwork)
library(rstanarm) # special today
library(tidyverse)

theme_set(theme_bw())
```

Data Ingest and Partitioning

Today's Data

Today, we'll use an R data set (.Rds) to import the dm1000 data.

```
dm1000 <- read_rds("data/dm_1000.Rds")
```

- This allows us to read in the data just as they were last saved in R, including “factoring” and handling of missing data, etc. The function `readRDS()` also works but is a little slower.
- To write an R data set, we'll use `write_rds(datasetname, "locationoncomputer")`. The function `saveRDS()` would also work, in a similar way, but be a little slower.

The dm1000 data

dm1000

A tibble: 1,000 x 17

	subject	sbp	dbp	insurance	age	n_income	ht
	<chr>	<dbl>	<dbl>	<fct>	<dbl>	<dbl>	<dbl>
1	M-0001	145	70	Medicaid	55	29853	1.63
2	M-0002	151	77	Commercial	52	31248	1.75
3	M-0003	127	73	Medicare	69	23362	1.65
4	M-0004	125	74	Medicaid	57	26033	1.63
5	M-0005	120	73	Medicare	68	85374	1.69
6	M-0006	127	75	Medicaid	56	31273	1.71
7	M-0007	114	81	Commercial	54	25445	1.68
8	M-0008	166	110	Medicare	45	67526	1.69
9	M-0009	111	77	Medicare	61	15203	1.91
10	M-0010	146	102	Medicaid	63	17628	1.86

... with 990 more rows, and 10 more variables:

wt <dbl>, a1c <dbl>, ldl <dbl>, tobacco <fct>,

statin <dbl>, cigs烟 <dbl>

Partitioning dm1000 into two groups

Before we do anything else today, let's split the data in dm1000 who have complete data on sbp and dbp into two groups:

- a model **development** or **training** sample (70% of rows)
- a model **evaluation** or **test** sample (the other 30%)

There are many ways to do this in R. Let's start by filtering out the observations with missing values of blood pressure.

```
dm994 <- dm1000 %>% filter(complete.cases(sbp, dbp)) %>%  
  select(subject, sbp, dbp, insurance)
```

```
dm994 %>% nrow()
```

```
[1] 994
```

```
dm994 %$% n_distinct(subject)
```

```
[1] 994
```

Now, let's build the partition.

Again, we want 70% of the sample in our training set, and the remaining 30% in our test set.

```
set.seed(4312021) # for replicating the sampling later  
  
dm_train <- dm994 %>% sample_frac(0.7)  
dm_test <- dm994 %>% anti_join(dm_train, by = "subject")  
  
nrow(dm_train); nrow(dm_test)
```

```
[1] 696
```

```
[1] 298
```

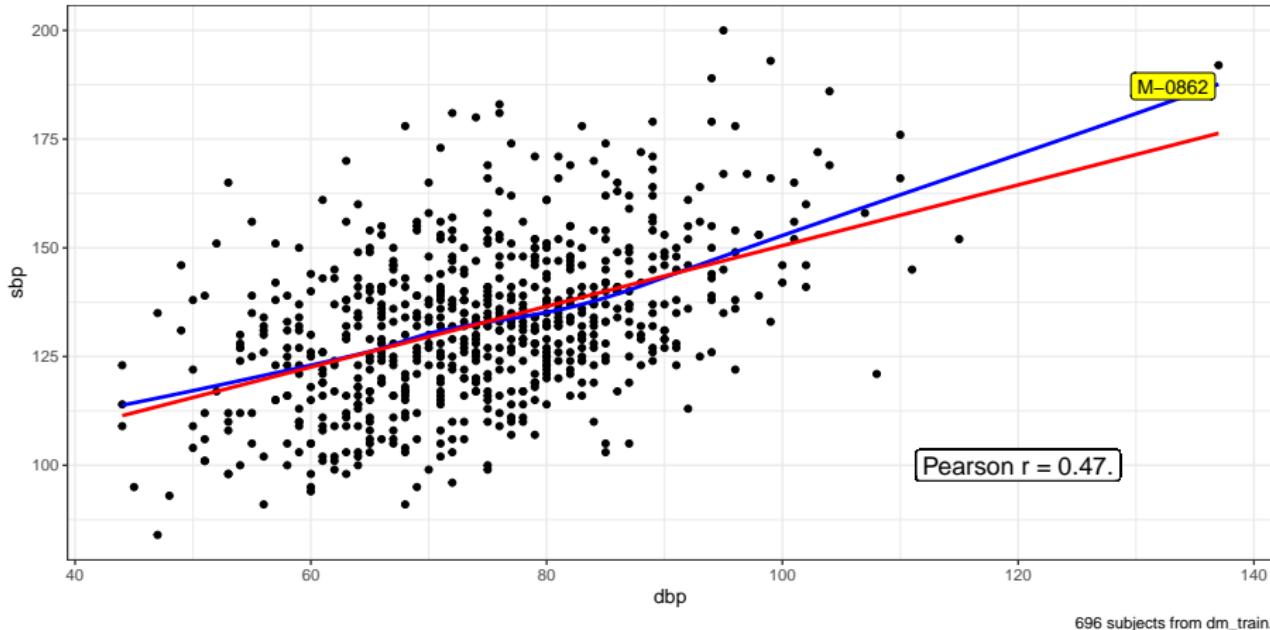
OK. Looks good!

Can dbp predict sbp?

Plotting sbp vs. dbp (training set)

Positive Association of SBP and DBP

loess smooth in blue, OLS model in red



- Note caption, labels, increased text size for Pearson r .

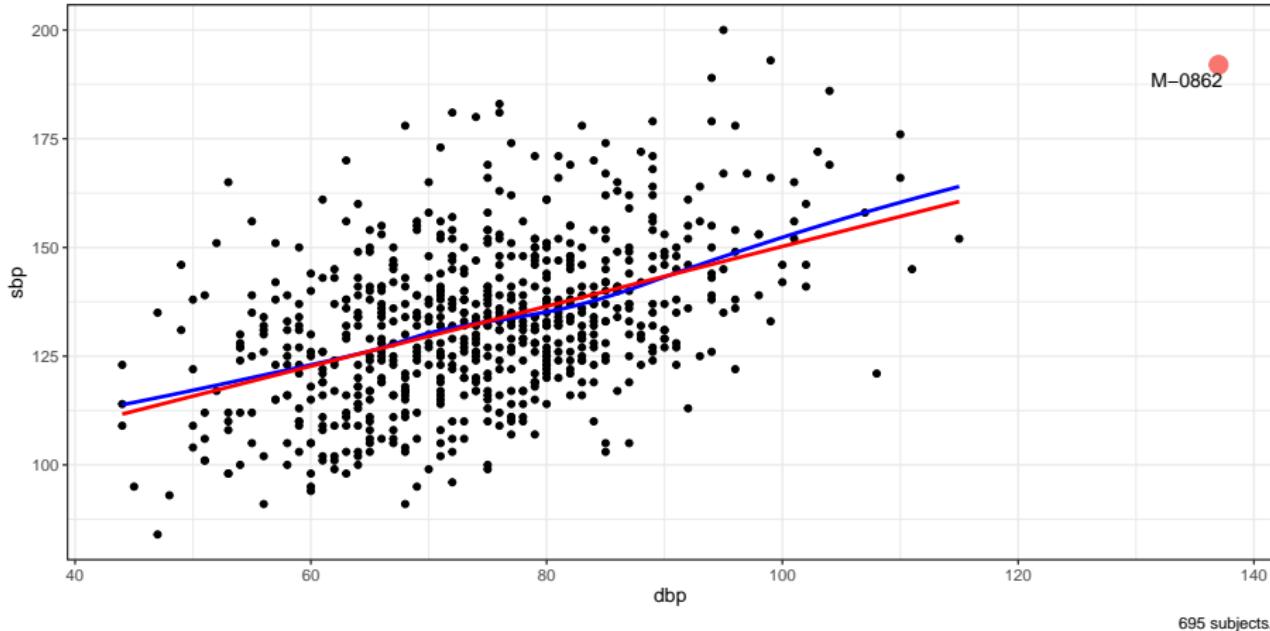
Code from Previous Slide

```
ggplot(data = dm_train, aes(x = dbp, y = sbp)) +  
  geom_point() +  
  geom_smooth(method = "loess", col = "blue",  
              se = FALSE, formula = y ~ x) +  
  geom_smooth(method = "lm", col = "red",  
              se = FALSE, formula = y ~ x) +  
  geom_label(x = 120, y = 100, size = 5,  
             label = glue('Pearson r = {round_half_up(  
               cor(dm_train$sbp, dm_train$dbp),2)}.')) +  
  geom_label_repel(data = dm_train %>% filter(dbp > 120),  
                    aes(label = subject), fill = "yellow") +  
  labs(title = "Positive Association of SBP and DBP",  
        subtitle = "loess smooth in blue, OLS model in red",  
        caption =  
          glue('{nrow(dm_train)} subjects from dm_train.'))
```

Redo plot without point M-0862

What happens if we drop M-0862?

Newly fit loess in blue, new OLS in red



- Note increased size and new color of point M-0862, use of `geom_text_repel` instead of `geom_label_repel`, adjusted caption.

Code from Previous Slide

```
ggplot(data = dm_train, aes(x = dbp, y = sbp)) +  
  geom_point() +  
  geom_smooth(data = dm_train %>% filter(dbp <= 120),  
              method = "loess", col = "blue",  
              se = FALSE, formula = y ~ x) +  
  geom_smooth(data = dm_train %>% filter(dbp <= 120),  
              method = "lm", col = "red",  
              se = FALSE, formula = y ~ x) +  
  geom_point(data = dm_train %>% filter(dbp > 120),  
             aes(col = "purple", size = 3)) +  
  geom_text_repel(data = dm_train %>% filter(dbp > 120),  
                  aes(label = subject)) +  
  guides(color = "none", size = "none") +  
  labs(title = "What happens if we drop M-0862?",  
        subtitle = "Newly fit loess in blue, new OLS in red",  
        caption = glue('{nrow(dm_train)-1} subjects.'))
```

Modeling sbp using dbp (training set)

```
m1_train <- lm(sbp ~ dbp, data = dm_train)

tidy(m1_train, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, conf.low, conf.high) %>% kable()
```

term	estimate	conf.low	conf.high
(Intercept)	80.6798905	74.4662421	86.893539
dbp	0.6982168	0.6160396	0.780394

```
glance(m1_train) %>% select(nobs, r.squared) %>% kable()
```

nobs	r.squared
696	0.2200811

Summarizing the Training Fit

- ① We can use `extract_eq()` from the `equatiomatic` package to present the equation from our model in a fairly attractive way, but we must use the code chunk header `{r, results = 'asis'}`.

```
extract_eq(m1_train, use_coefs = TRUE, coef_digits = 3)
```

$$\widehat{\text{sbp}} = 80.68 + 0.698(\text{dbp})$$

- ② The `summary` function when applied to a linear model (`lm`) produces output that isn't organized in a way that allows up to plot or present it effectively outside of an R session.

```
summary(m1_train)
```

A screenshot follows on the next page.

```
> summary(m1_train)

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

Residuals:
    Min      1Q  Median      3Q     Max 
-37.159 -11.537 -1.348  10.066  52.990 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 80.67989   3.77259   21.39 <2e-16 ***
dbp         0.69822   0.04989   13.99 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16.12 on 694 degrees of freedom
Multiple R-squared:  0.2201,    Adjusted R-squared:  0.219 
F-statistic: 195.8 on 1 and 694 DF,  p-value: < 2.2e-16
```

Why I like tidy() and other broom functions



@allison_horst

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

Does R like this linear model?

```
tidy(m1_train) %>% kable(digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	80.680	3.773	21.386	0
dbp	0.698	0.050	13.994	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_train) %>%  
  select(nobs, r.squared, p.value, sigma) %>% kable()
```

nobs	r.squared	p.value	sigma
696	0.2200811	0	16.11605

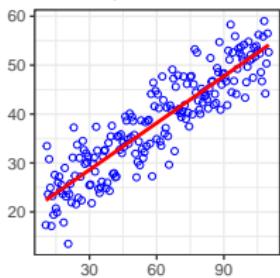
- $r.squared = R^2$, the proportion of variation in `sbp` accounted for by the model using `dbp`.
 - indicates improvement over predicting `mean(sbp)` for everyone
- `p.value` = refers to a global F test
 - indicates something about combination of r^2 and sample size
- `sigma` = residual standard error

`glance` provides 9 additional summaries for a linear model.

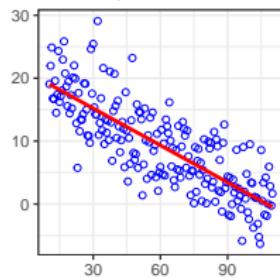
Calibrating Yourself on R-square

Can you match each plot to its R-square?

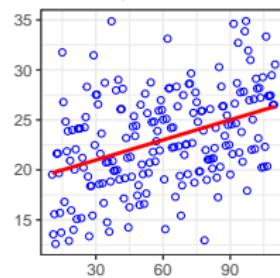
A. R-square = ?



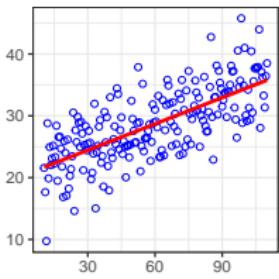
B. R-square = ?



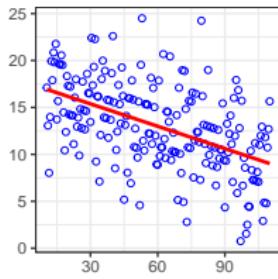
C. R-square = ?



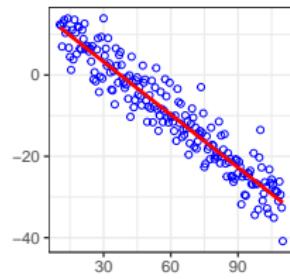
D. R-square = ?



E. R-square = ?



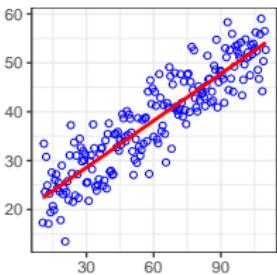
F. R-sq = ?



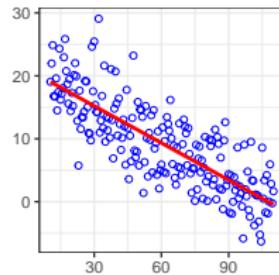
- R^2 values shown include 0.16, 0.25, 0.49, 0.64, 0.81 and 0.91

Gaining Insight into what R-square implies

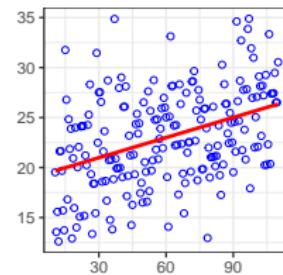
$r = +0.9, R-\text{sq} = 0.81$



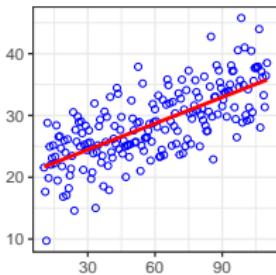
$r = -0.8, R-\text{sq} = 0.64$



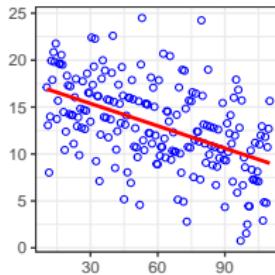
$r = +0.4, R-\text{sq} = 0.16$



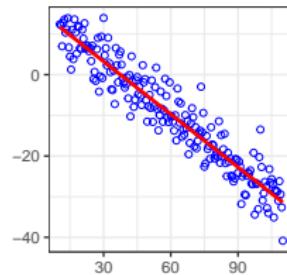
$r = +0.7, R-\text{sq} = 0.49$



$r = -0.5, R-\text{sq} = 0.25$



$r = -0.95, R-\text{sq} = 0.905$



Obtaining Residuals and Fitted Values in the Training Sample

Predict using m1_train: $sbp = 80.68 + 0.70 \text{ dbp}$

Use augment (from broom) to capture fitted values and residuals for all of the data in the training sample.

```
augment(m1_train, data = dm_train) %>%
  select(subject, sbp, dbp, .fitted, .resid) %>%
  slice_min(., order_by = subject, n = 2) %>% kable(dig = 2)
```

subject	sbp	dbp	.fitted	.resid
M-0002	151	77	134.44	16.56
M-0003	127	73	131.65	-4.65

- Subject M-0002 has an observed sbp of 151, and dbp of 77.
- Our m1_train model **fits** (predicts) M-0002's sbp to be 134.44, so that's a **residual** of $151 - 134.44 = 16.56 \text{ mm Hg}$.
- Note that **residual = observed - fitted**.

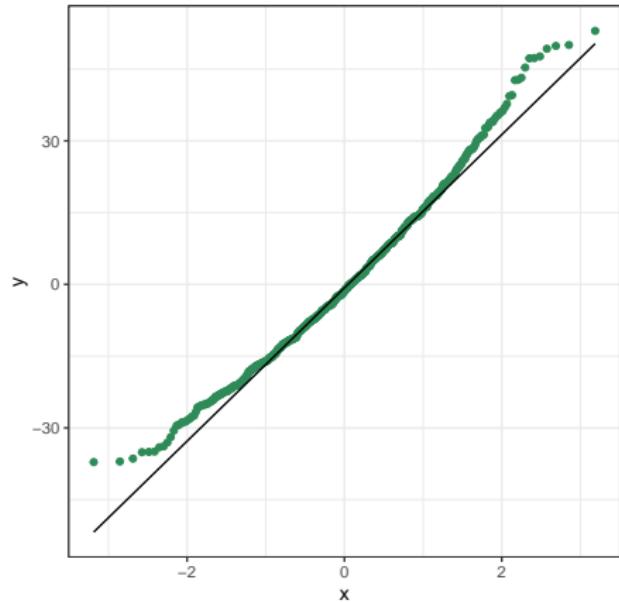
What must we assume for a regression model?

Briefly (for now), we assume that:

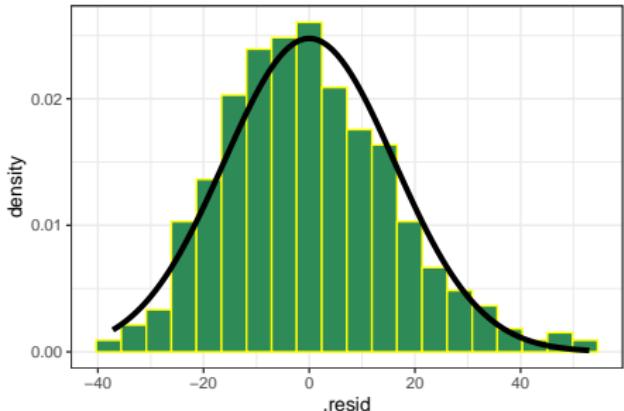
- the regression relationship is linear, rather than curved, and we can assess this by plotting the regression residuals (prediction errors) against the fitted values and looking to see if a curve emerges
- the regression residuals show similar variance across levels of the fitted values, and again we can get insight into this by plotting residuals vs. predicted values
- the regression residuals (prediction errors) are well described by a Normal model, and we can assess this with all of our usual visualizations to help decide on whether a Normal model is reasonable for a batch of data.
- We assess all of these issues (and others) with plots of the residuals.
Let's start with the **Normality** assumption...

Plot residuals from m1_train

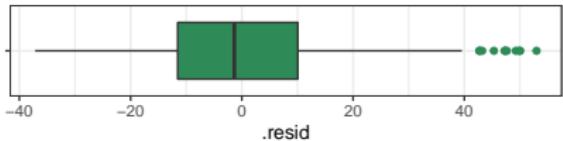
Normal Q-Q: 994 m1 Residuals



Hist + Normal Density: m1 Residuals



Boxplot: m1 Residuals



min	Q1	median	Q3	max	mean	sd	n	missing
-37.2	-11.5	-1.3	10.1	53	0	16.1	696	0

Plot Residuals vs. Predicted (Fitted) Values (code)

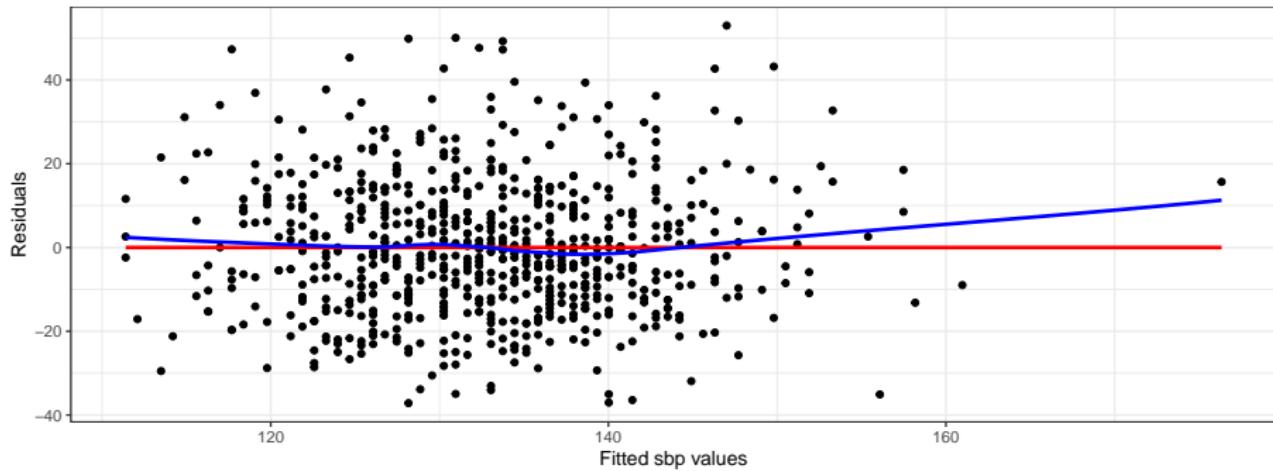
```
m1_train_aug <- augment(m1_train, data = dm_train)

ggplot(m1_train_aug, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red",
              formula = y ~ x, se = FALSE) +
  geom_smooth(method = "loess", col = "blue",
              formula = y ~ x, se = FALSE) +
  labs(title = "m1_train: Residuals vs. Fitted Values",
       x = "Fitted sbp values", y = "Residuals")
```

m1_train: Residuals vs. Predicted (Fitted) Values

- We're looking to see if there is a substantial curve in the plot, or if the variability changes materially from left to right.
- What we want to see is a "fuzzy football" actually.

m1_train: Residuals vs. Fitted Values



This sort of fuzzy football...



Making Predictions Out of Sample (into the Test Sample)

Use model m1_train to predict SBP in dm_test

```
m1_test_aug <- augment(m1_train, newdata = dm_test)
```

```
m1_test_aug %>% nrow()
```

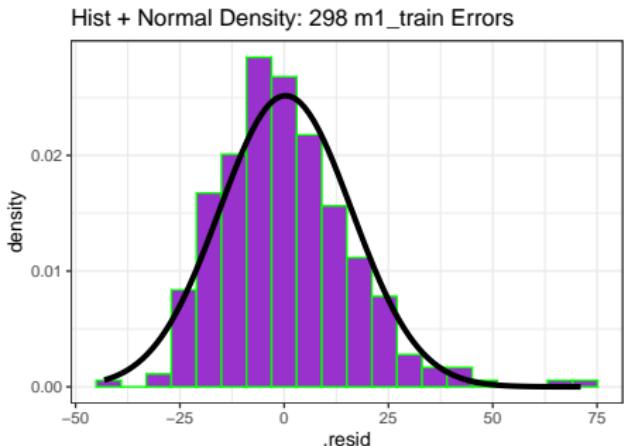
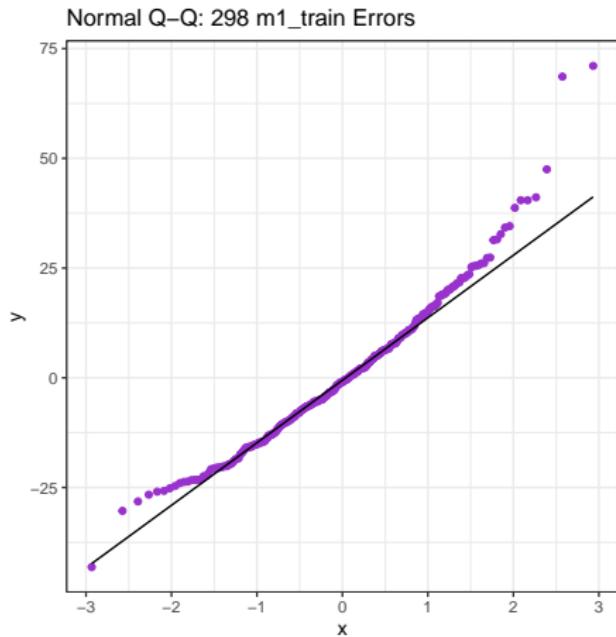
```
[1] 298
```

- We have predictions from m1_train for the 298 subjects in dm_test.
- Remember we didn't use the dm_test data to build m1_train.

```
m1_test_aug %>%
  select(subject, sbp, dbp, .fitted, .resid) %>%
  slice_min(., order_by = subject, n = 2) %>% kable(dig = 2)
```

subject	sbp	dbp	.fitted	.resid
M-0001	145	70	129.56	15.44
M-0007	114	81	137.24	-23.24

dm_test (n = 298): m1_train Prediction Errors



min	Q1	median	Q3	max	mean	sd	n	missing
-43.1	-10.2	-1	9	71.1	0.3	15.9	298	0

Out-of-Sample (Test Set) Error Summaries (m1)

- Mean Absolute Prediction Error = 12.14
- Maximum Absolute Prediction Error = 71.07
- (square Root of) Mean Squared Prediction Error (RMSPE) = 15.83

```
mosaic::favstats(~ abs(.resid), data = m1_test_aug) %>%
  select(n, min, median, max, mean, sd) %>%
  kable(digits = 2)
```

	n	min	median	max	mean	sd
	298	0.03	9.82	71.07	12.14	10.18

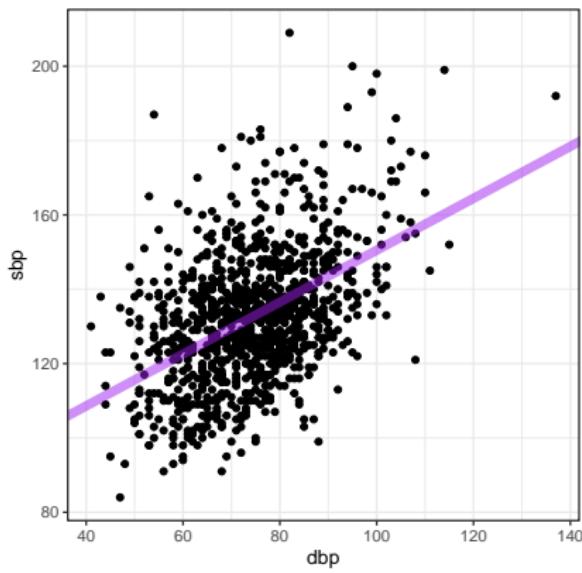
```
sqrt(mean(m1_test_aug$.resid^2))
```

```
[1] 15.83017
```

These statistics are most useful when we're comparing two models.

Back to all 994 values. Does m1_train work well?

```
ggplot(dm994, aes(x = dbp, y = sbp)) +  
  geom_point() + theme(aspect.ratio = 1) +  
  geom_abline(intercept = 80.6799, slope = 0.6982,  
              col = "purple", lwd = 2.5, alpha = 0.5)
```



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

Nope.

```
library(rstanarm)
```

```
m2_train <- stan_glm(sbp ~ dbp, data = dm_train)
```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 0 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition!

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_train)
```

stan_glm

family: gaussian [identity]

formula: sbp ~ dbp

observations: 696

predictors: 2

Median MAD_SD

(Intercept) 80.8 3.7

dbp 0.7 0.0

Auxiliary parameter(s):

Median MAD_SD

sigma 16.1 0.4

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

```
broom::tidy(m1_train) # fit with lm
```

```
# A tibble: 2 x 5
  term      estimate std.error statistic p.value
  <chr>      <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  80.7      3.77     21.4  2.47e-78
2 dbp         0.698     0.0499    14.0  2.23e-39
```



```
broom.mixed::tidy(m2_train) # stan_glm with default priors
```

```
# A tibble: 2 x 3
  term      estimate std.error
  <chr>      <dbl>     <dbl>
1 (Intercept)  80.8      3.74
2 dbp         0.697     0.0490
```

Test Sample fits and residuals from Bayesian model

```
m2_test_aug <- dm_test %>% select(subject, sbp, dbp) %>%
  mutate(.fitted = predict(m2_train, newdata = dm_test),
        .resid = sbp - .fitted)

m2_test_aug %>%
  select(subject, sbp, dbp, .fitted, .resid) %>%
  slice_min(., order_by = subject, n = 2) %>% kable(dig = 2)
```

subject	sbp	dbp	.fitted	.resid
M-0001	145	70	129.56	15.44
M-0007	114	81	137.23	-23.23

Out-of-Sample (Test Set) Error Summaries (m2)

```
mosaic::favstats(~ abs(.resid), data = m2_test_aug) %>%
  select(n, min, median, max, mean, sd) %>%
  kable(digits = 3)
```

	n	min	median	max	mean	sd
	298	0.021	9.807	71.071	12.137	10.178

```
sqrt(mean(m2_test_aug$resid^2))
```

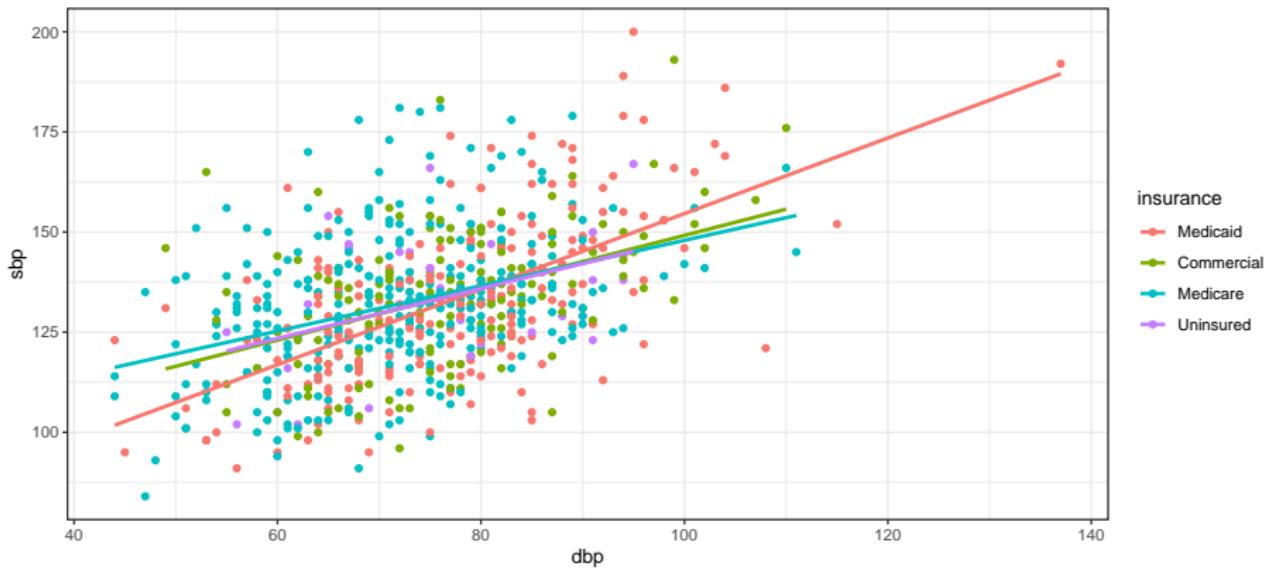
```
[1] 15.82868
```

Test Set Error Summary	OLS model m1	Bayes model m2
Mean Absolute Prediction Error	12.139	12.137
Maximum Absolute Prediction Error	71.066	71.071
Root Mean Squared Prediction Error	15.83	15.829

What if we add another predictor? (Insurance)

Plotting sbp vs. dbp and insurance

```
ggplot(data = dm_train, aes(x = dbp, y = sbp,  
                           col = insurance, group = insurance)) +  
  geom_point() +  
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE)
```



Two possible models

```
m3_train <- lm(sbp ~ dbp + insurance, data = dm_train)  
m4_train <- lm(sbp ~ dbp * insurance, data = dm_train)
```

- What is the difference between m3 and m4?
 - Model m3 will allow the intercept term of the sbp-dbp relationship to vary depending on insurance.
 - Model m4 will allow both the slope and intercept of the sbp-dbp relationship to vary depending on insurance.

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72 * \text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72*\text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$
- $\text{sbp} = 78.69 + 0.72*\text{dbp}$

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72*\text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$
- $\text{sbp} = 78.69 + 0.72*\text{dbp}$
- For a Medicaid subject, m3 predicts $\text{sbp} = 77.58 + 0.72 \text{ dbp}$

Equation for m3 ($sbp \sim dbp + insurance$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{sbp} = 77.58 + 0.72(dbp) +
1.11(insurance_{Commercial}) + 2.73(insurance_{Medicare}) +
1.16(insurance_{Uninsured})$$

- Predicted sbp by m3 for a Commercial subject?
- $sbp = 77.58 + 0.72*dbp + 1.11(1) + 2.73(0) + 1.16(0)$
- $sbp = 78.69 + 0.72*dbp$
- For a Medicaid subject, m3 predicts $sbp = 77.58 + 0.72 dbp$
- For a Medicare subject, m3 predicts $sbp = 80.31 + 0.72 dbp$

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72*\text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$
- $\text{sbp} = 78.69 + 0.72*\text{dbp}$
- For a Medicaid subject, m3 predicts $\text{sbp} = 77.58 + 0.72 \text{ dbp}$
- For a Medicare subject, m3 predicts $\text{sbp} = 80.31 + 0.72 \text{ dbp}$
- For an uninsured subject, m3 predicts $\text{sbp} = 78.74 + 0.72 \text{ dbp}$

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72*\text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$
- $\text{sbp} = 78.69 + 0.72*\text{dbp}$
- For a Medicaid subject, m3 predicts $\text{sbp} = 77.58 + 0.72 \text{ dbp}$
- For a Medicare subject, m3 predicts $\text{sbp} = 80.31 + 0.72 \text{ dbp}$
- For an uninsured subject, m3 predicts $\text{sbp} = 78.74 + 0.72 \text{ dbp}$
- Note: only the intercept term varies by insurance in m3.

Equation for m4 ($\text{sbp} \sim \text{dbp} * \text{insurance}$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 60.26 + 0.94(\text{dbp}) + \\ 23.54(\text{insurance}_{\text{Commercial}}) + 31.04(\text{insurance}_{\text{Medicare}}) + \\ 25.78(\text{insurance}_{\text{Uninsured}}) - 0.29(\text{dbp} \times \text{insurance}_{\text{Commercial}}) - \\ 0.38(\text{dbp} \times \text{insurance}_{\text{Medicare}}) - 0.32(\text{dbp} \times \text{insurance}_{\text{Uninsured}})$$

- m4 predicts, for a Commercial subject...

Equation for m4 ($\text{sbp} \sim \text{dbp} * \text{insurance}$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 60.26 + 0.94(\text{dbp}) +
23.54(\text{insurance}_{\text{Commercial}}) + 31.04(\text{insurance}_{\text{Medicare}}) +
25.78(\text{insurance}_{\text{Uninsured}}) - 0.29(\text{dbp} \times \text{insurance}_{\text{Commercial}}) -
0.38(\text{dbp} \times \text{insurance}_{\text{Medicare}}) - 0.32(\text{dbp} \times \text{insurance}_{\text{Uninsured}})$$

- m4 predicts, for a Commercial subject...
- $\text{sbp} = 60.26 + 0.94 * \text{dbp} + 23.54 (1) + 31.04 (0) + 25.78 (0) - 0.29 (\text{dbp} * 1) - 0.38 (\text{dbp} * 0) - 0.32 (\text{dbp} * 0)$

Equation for m4 ($sbp \sim dbp * insurance$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{sbp} = 60.26 + 0.94(dbp) +
23.54(insurance_{Commercial}) + 31.04(insurance_{Medicare}) +
25.78(insurance_{Uninsured}) - 0.29(dbp \times insurance_{Commercial}) -
0.38(dbp \times insurance_{Medicare}) - 0.32(dbp \times insurance_{Uninsured})$$

- m4 predicts, for a Commercial subject...
- $sbp = 60.26 + 0.94 * dbp + 23.54 (1) + 31.04 (0) + 25.78 (0) - 0.29 (dbp * 1) - 0.38 (dbp * 0) - 0.32 (dbp * 0)$
- $sbp = (60.26 + 23.54) + (0.94 - 0.29) * dbp$

Equation for m4 ($sbp \sim dbp * insurance$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{sbp} = 60.26 + 0.94(dbp) +
23.54(insurance_{Commercial}) + 31.04(insurance_{Medicare}) +
25.78(insurance_{Uninsured}) - 0.29(dbp \times insurance_{Commercial}) -
0.38(dbp \times insurance_{Medicare}) - 0.32(dbp \times insurance_{Uninsured})$$

- m4 predicts, for a Commercial subject...
- $sbp = 60.26 + 0.94 * dbp + 23.54 (1) + 31.04 (0) + 25.78 (0) - 0.29 (dbp * 1) - 0.38 (dbp * 0) - 0.32 (dbp * 0)$
- $sbp = (60.26 + 23.54) + (0.94 - 0.29) * dbp$
- $sbp = 83.80 - 0.65 dbp$ for Commercial subjects

Equation for m4 ($sbp \sim dbp * insurance$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{sbp} = 60.26 + 0.94(dbp) +
23.54(insurance_{Commercial}) + 31.04(insurance_{Medicare}) +
25.78(insurance_{Uninsured}) - 0.29(dbp \times insurance_{Commercial}) -
0.38(dbp \times insurance_{Medicare}) - 0.32(dbp \times insurance_{Uninsured})$$

- For Medicaid subjects, $sbp = 60.26 + 0.94 * dbp$
- For Medicare subjects, $sbp = 91.30 + 0.56 * dbp$
- For the uninsured, $sbp = 86.04 + 0.62 * dbp$
- So both the slope and the intercept are changing in m4

How do these models do in the training sample?

- Model m3

```
glance(m3_train) %>%
  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
0.224	0.22	16.1	5851.1	5878.4

- Model m4

```
glance(m4_train) %>%
  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
0.236	0.229	16	5846	5886.9

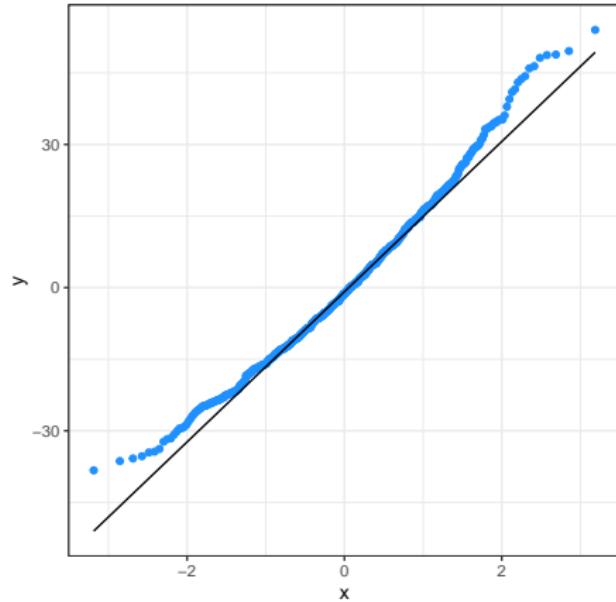
Augmenting and Testing Models m3 and m4

```
m3_train_aug <- augment(m3_train, data = dm_train)
m3_test_aug <- augment(m3_train, newdata = dm_test)

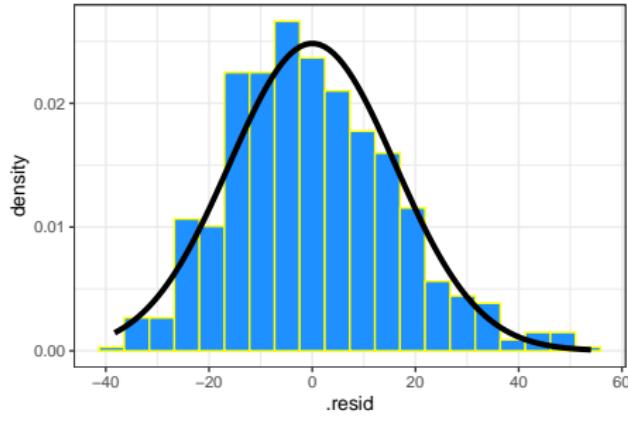
m4_train_aug <- augment(m4_train, data = dm_train)
m4_test_aug <- augment(m4_train, newdata = dm_test)
```

Residuals (training sample) for m3_train

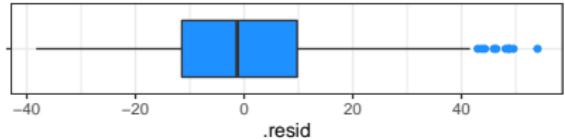
Normal Q-Q: 698 m3 Residuals



Hist + Normal Density: m3 Residuals



Boxplot: m3 Residuals

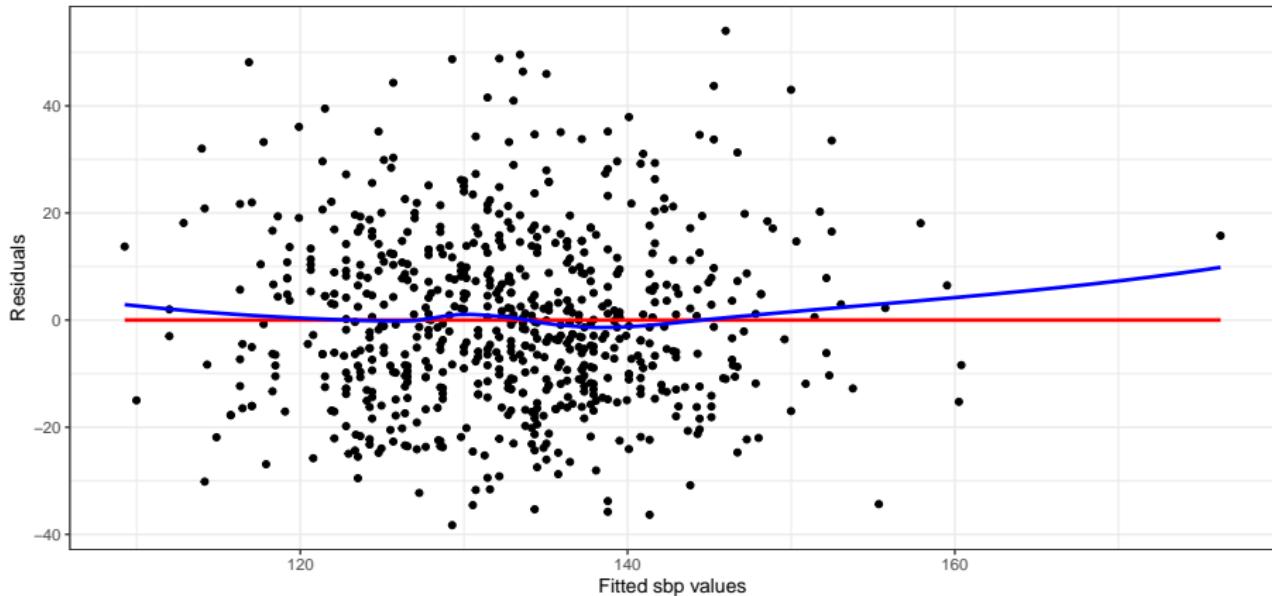


min	Q1	median	Q3	max	mean	sd	n	missing
-38.3	-11.5	-1.3	9.8	54	0	16.1	696	0

m3_train: Residuals vs. Predicted (Fitted) Values

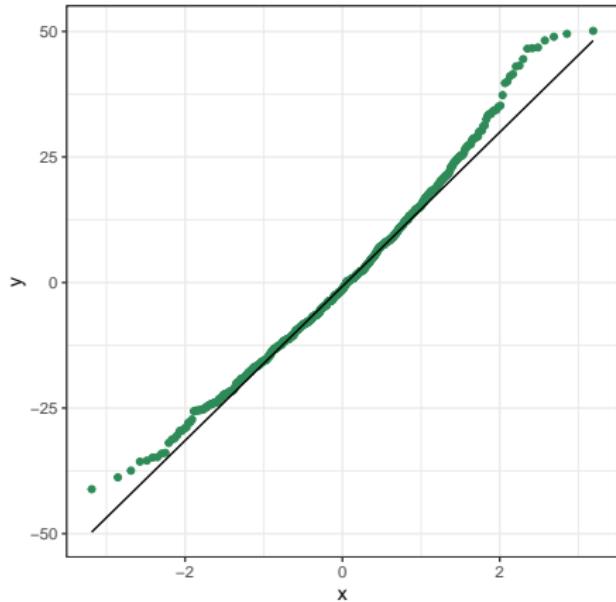
- We're looking for a "fuzzy football"...

m3_train: Residuals vs. Fitted Values

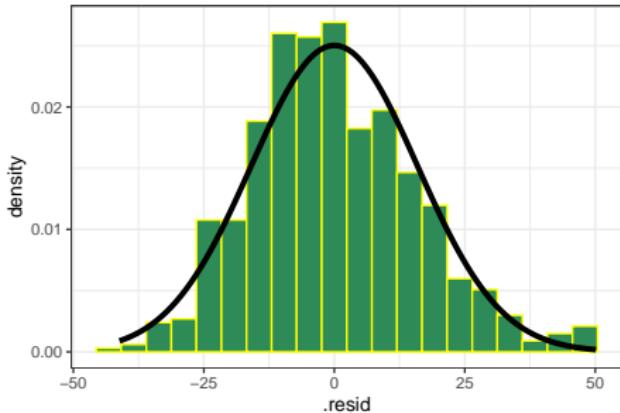


Residuals (training sample) for m4_train

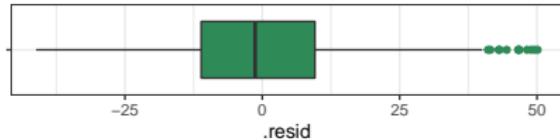
Normal Q-Q: 698 m4 Residuals



Hist + Normal Density: m4 Residuals



Boxplot: m4 Residuals

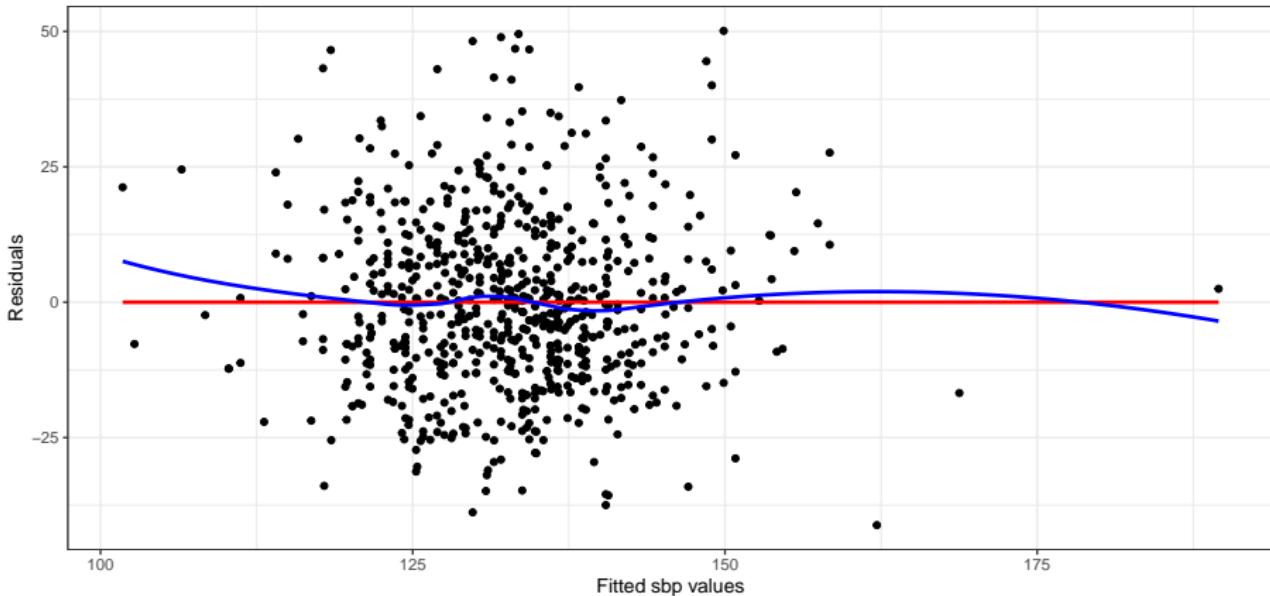


min	Q1	median	Q3	max	mean	sd	n	missing
-41.2	-11.1	-1.3	9.6	50.1	0	15.9	696	0

m4_train: Residuals vs. Predicted (Fitted) Values

- We're looking for a "fuzzy football"...

m4_train: Residuals vs. Fitted Values



Comparing performance on the training data

```
bind_rows(glance(m1_train), glance(m2_train),
          glance(m3_train), glance(m4_train)) %>%
  mutate(modname = c("m1", "m2", "m3", "m4")) %>%
  select(modname, r2 = r.squared, adj_r2 = adj.r.squared,
         sigma, AIC, BIC) %>%
  kable(digits = c(0, 3, 3, 2, 1, 1))
```

modname	r2	adj_r2	sigma	AIC	BIC
m1	0.220	0.219	16.12	5848.7	5862.3
m2	NA	NA	16.12	NA	NA
m3	0.224	0.220	16.11	5851.1	5878.4
m4	0.236	0.229	16.02	5846.0	5886.9

- The `glance()` function produces different results for a Bayesian `stan_glm()` model like `m2`, so we'll ignore that for now.

Comparing performance on the test data

Here are some fundamental summaries of absolute prediction error (APE) along with the root mean squared prediction error (RMSPE) for each of our models, in the **testing** sample.

Summary	Mean APE	Max APE	RMSPE
m1_train: lm	12.139	71.066	15.83
m2_train: stan_glm	12.137	71.071	15.829
m3_train: dbp+insurance	12.04	72.367	15.778
m4_train: dbp*insurance	11.947	71.37	15.647

- Which of these models displays the strongest predictive performance in our test sample?

Reminder of Today's Agenda

- ① Ingesting dm1000 data using R data set format (.Rds)
- ② Partitioning data into model training/test samples.
- ③ Augmenting a Scatterplot (labeling, size, color) and fitting a simple OLS (linear) model m1
- ④ Using summary() and extract_eq() on a regression model.
- ⑤ The broom package and tidy(), glance() and augment()
- ⑥ Calibrating your understanding of R-square a bit
- ⑦ Assessing Regression Assumptions with Residual Plots
- ⑧ Making Predictions into the Test Sample
- ⑨ Assessing Quality of Fit using the Test Sample with mean and maximum absolute prediction error and with RMSPE
- ⑩ Fitting a Bayesian Linear Model with default priors (m2)
- ⑪ Including Insurance without (m3) and with (m4) interaction with dbp in linear models

431 Class 11

thomaselove.github.io/431

2021-09-28

Today's Agenda

- ① Developing Four Models for `sbp` using `dbp` (and `insurance`)
- ② Fitting a Bayesian Linear Model with default priors (`m2`)
- ③ Including Insurance without (`m3`) and with (`m4`) interaction with `dbp` in linear models
- ④ Visualizing Categorical Data
- ⑤ Assessing Association in Cross-Tabulations

Today's Packages

```
library(broom)
library(equatiomatic) # new today
library(ggrepel) # sort of new today
library(glue) # sort of new today
library(janitor)
library(knitr)
library(magrittr)
library(patchwork)
library(rstanarm) # special today
library(tidyverse)

theme_set(theme_bw())
```

Today's Data

Again, we'll use an R data set (.Rds) to import the dm1000 data.

```
dm1000 <- read_rds("data/dm_1000.Rds")
```

Then, we'll again partition the dm1000 cases with complete BP data into training and test samples.

```
dm994 <- dm1000 %>% filter(complete.cases(sbp, dbp)) %>%
  select(subject, sbp, dbp, insurance)

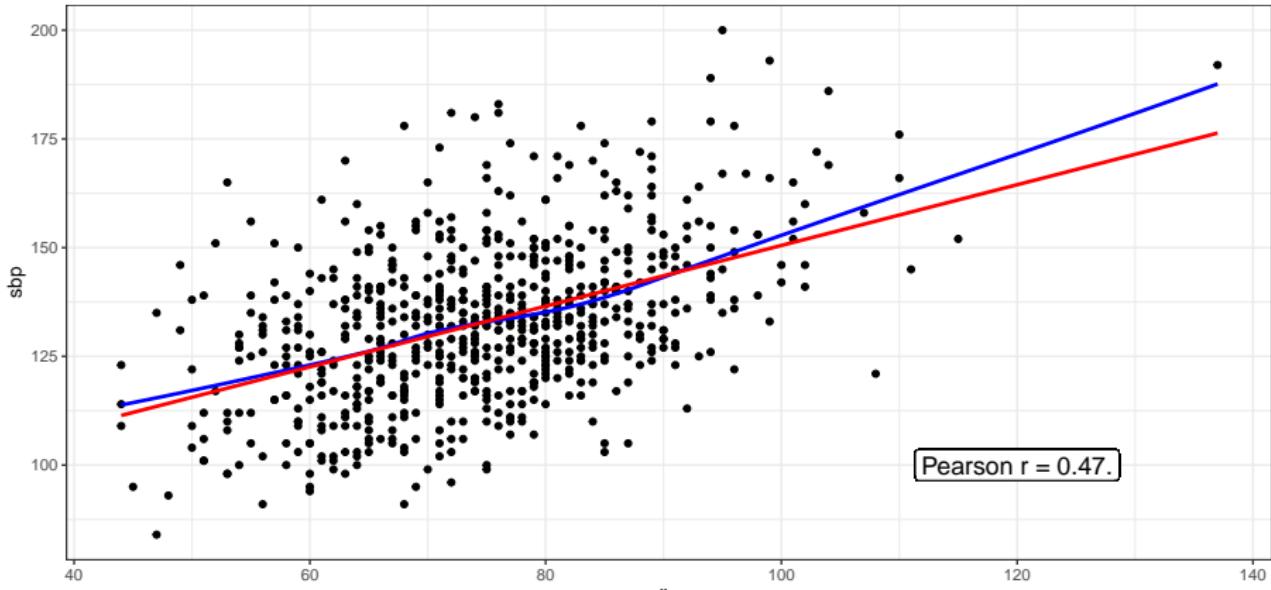
set.seed(4312021) # for replicating the sampling later
dm_train <- dm994 %>% sample_frac(0.7)
dm_test <- dm994 %>% anti_join(dm_train, by = "subject")
```

Back to Regression: Can dbp predict sbp?

Plotting sbp vs. dbp (training set)

Positive Association of SBP and DBP

loess smooth in blue, OLS model in red



696 subjects from dm_train.

Model m1 for sbp using dbp (training set)

```
m1_train <- lm(sbp ~ dbp, data = dm_train)

tidy(m1_train, conf.int = TRUE, conf.level = 0.90) %>%
  select(term, estimate, conf.low, conf.high) %>% kable()
```

term	estimate	conf.low	conf.high
(Intercept)	80.6798905	74.4662421	86.893539
dbp	0.6982168	0.6160396	0.780394

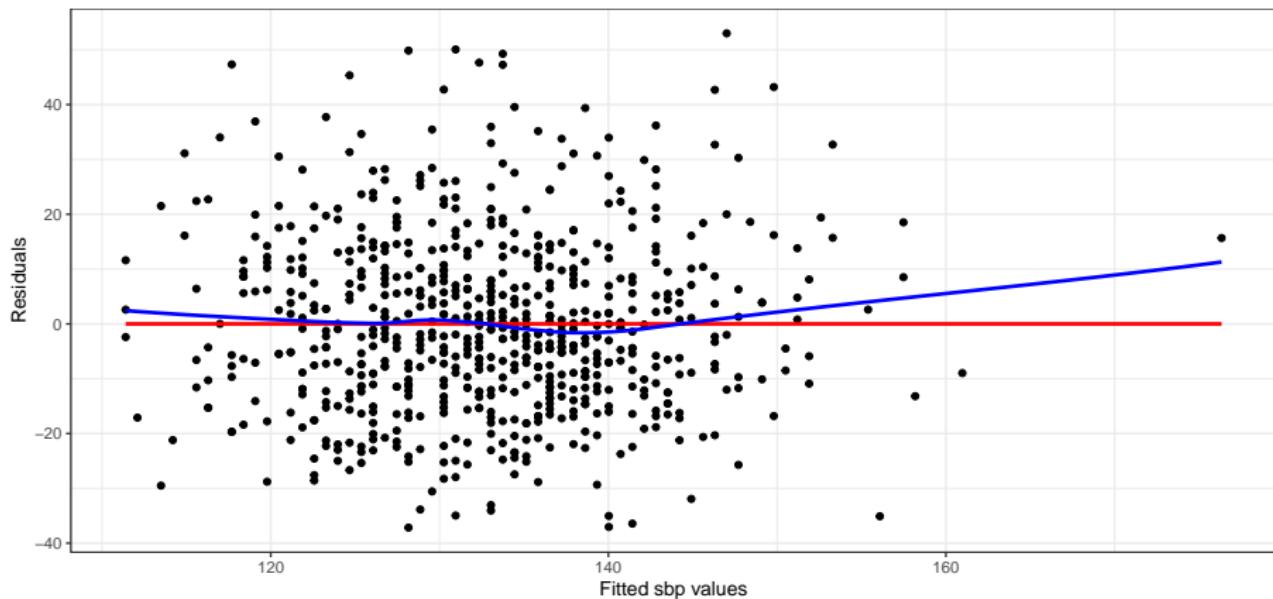
```
glance(m1_train) %>% select(nobs, r.squared, adj.r.squared,
                               sigma, AIC, BIC) %>% kable()
```

nobs	r.squared	adj.r.squared	sigma	AIC	BIC
696	0.2200811	0.2189573	16.11605	5848.663	5862.299

m1_train: Residuals vs. Predicted (Fitted) Values

```
m1_train_aug <- augment(m1_train, data = dm_train)
```

m1_train: Residuals vs. Fitted Values



Use model m1_train to predict SBP in dm_test

```
m1_test_aug <- augment(m1_train, newdata = dm_test)

mosaic::favstats(~ abs(.resid), data = m1_test_aug) %>%
  select(n, min, median, max, mean, sd) %>% kable(digits = 3)
```

n	min	median	max	mean	sd
298	0.028	9.822	71.066	12.139	10.177

```
sqrt(mean(m1_test_aug$.resid^2))
```

```
[1] 15.83017
```

	Summary	Model m1
Mean Absolute Prediction Error		12.139
Maximum Absolute Prediction Error		71.066
Root Mean Squared Prediction Error (RMSPE)		15.83

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

Nope.

```
library(rstanarm)
```

```
m2_train <- stan_glm(sbp ~ dbp, data = dm_train)
```

SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 0 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition!

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)

Default Prior Details

```
prior_summary(m2_train)
```

```
> prior_summary(m2_train)
Priors for model 'm2_train'
-----
Intercept (after predictors centered)
  Specified prior:
    ~ normal(location = 133, scale = 2.5)
  Adjusted prior:
    ~ normal(location = 133, scale = 46)

Coefficients
  Specified prior:
    ~ normal(location = 0, scale = 2.5)
  Adjusted prior:
    ~ normal(location = 0, scale = 3.7)

Auxiliary (sigma)
  Specified prior:
    ~ exponential(rate = 1)
  Adjusted prior:
    ~ exponential(rate = 0.055)
-----
See help('prior_summary.stanreg') for more details
```

Bayesian fitted linear model for our sbp data

```
print(m2_train)
```

stan_glm

```
family: gaussian [identity]
formula: sbp ~ dbp
observations: 696
predictors: 2
```

Median MAD_SD

(Intercept)	80.6	4.0
dbp	0.7	0.1

Auxiliary parameter(s):

Median MAD_SD

sigma	16.2	0.4
-------	------	-----

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

```
broom::tidy(m1_train) # fit with lm
```

```
# A tibble: 2 x 5
  term      estimate std.error statistic p.value
  <chr>      <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept) 80.7      3.77      21.4 2.47e-78
2 dbp         0.698     0.0499     14.0 2.23e-39
```



```
broom.mixed::tidy(m2_train) # stan_glm with default priors
```

```
# A tibble: 2 x 3
  term      estimate std.error
  <chr>      <dbl>     <dbl>
1 (Intercept) 80.6      3.97
2 dbp         0.699     0.0522
```

Obtaining fits and residuals from Model m2

In the model training sample

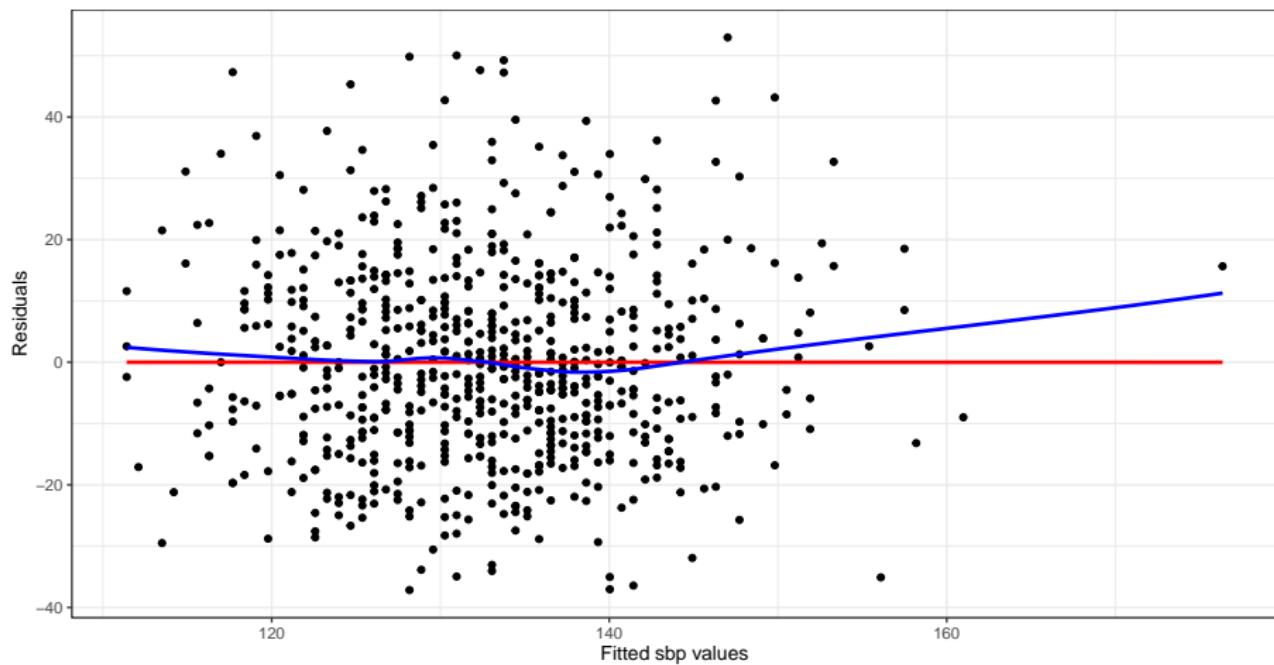
```
m2_train_aug <- dm_train %>% select(subject, sbp, dbp) %>%  
  mutate(.fitted = predict(m2_train, newdata = dm_train),  
        .resid = sbp - .fitted)
```

In the model test sample

```
m2_test_aug <- dm_test %>% select(subject, sbp, dbp) %>%  
  mutate(.fitted = predict(m2_train, newdata = dm_test),  
        .resid = sbp - .fitted)
```

Residuals vs. Fitted Values from Model m2 (training)

m2_train: Residuals vs. Fitted Values



Out-of-Sample (Test Set) Error Summaries (m2)

```
mosaic::favstats(~ abs(.resid), data = m2_test_aug) %>%  
  select(n, min, median, max, mean, sd) %>% kable(digits = 3)
```

n	min	median	max	mean	sd
298	0.036	9.824	71.059	12.14	10.177

```
sqrt(mean(m2_test_aug$.resid^2))
```

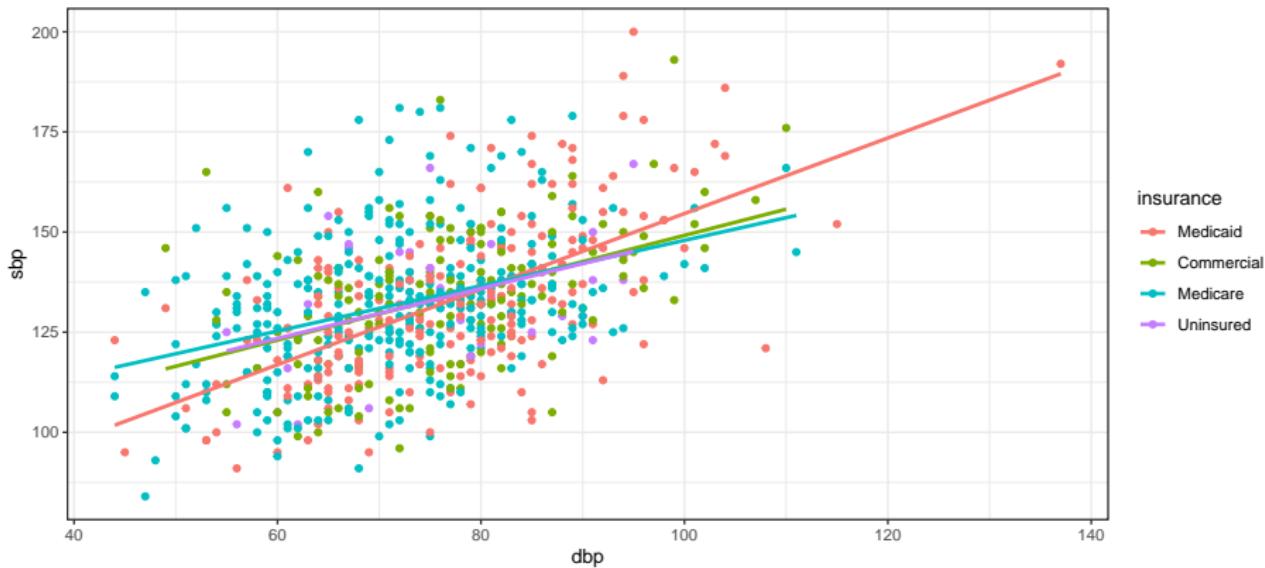
```
[1] 15.83019
```

Test Set Error Summary	OLS model m1	Bayes model m2
Mean Absolute Prediction Error	12.139	12.14
Maximum Absolute Prediction Error	71.066	71.059
Root Mean Squared Prediction Error	15.83	15.83

What if we add another predictor? (Insurance)

Plotting sbp vs. dbp and insurance

```
ggplot(data = dm_train, aes(x = dbp, y = sbp,  
                           col = insurance, group = insurance)) +  
  geom_point() +  
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE)
```



Two possible models

```
m3_train <- lm(sbp ~ dbp + insurance, data = dm_train)  
m4_train <- lm(sbp ~ dbp * insurance, data = dm_train)
```

- What is the difference between m3 and m4?
 - Model m3 will allow the intercept term of the sbp-dbp relationship to vary depending on insurance.
 - Model m4 will allow both the slope and intercept of the sbp-dbp relationship to vary depending on insurance.

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72 * \text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72*\text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$
- $\text{sbp} = 78.69 + 0.72*\text{dbp}$

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72*\text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$
- $\text{sbp} = 78.69 + 0.72*\text{dbp}$
- For a Medicaid subject, m3 predicts $\text{sbp} = 77.58 + 0.72 \text{ dbp}$

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72*\text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$
- $\text{sbp} = 78.69 + 0.72*\text{dbp}$
- For a Medicaid subject, m3 predicts $\text{sbp} = 77.58 + 0.72 \text{ dbp}$
- For a Medicare subject, m3 predicts $\text{sbp} = 80.31 + 0.72 \text{ dbp}$

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72*\text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$
- $\text{sbp} = 78.69 + 0.72*\text{dbp}$
- For a Medicaid subject, m3 predicts $\text{sbp} = 77.58 + 0.72 \text{ dbp}$
- For a Medicare subject, m3 predicts $\text{sbp} = 80.31 + 0.72 \text{ dbp}$
- For an uninsured subject, m3 predicts $\text{sbp} = 78.74 + 0.72 \text{ dbp}$

Equation for m3 ($\text{sbp} \sim \text{dbp} + \text{insurance}$)

```
extract_eq(m3_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 77.58 + 0.72(\text{dbp}) +
1.11(\text{insurance}_{\text{Commercial}}) + 2.73(\text{insurance}_{\text{Medicare}}) +
1.16(\text{insurance}_{\text{Uninsured}})$$

- Predicted sbp by m3 for a Commercial subject?
- $\text{sbp} = 77.58 + 0.72*\text{dbp} + 1.11(1) + 2.73(0) + 1.16(0)$
- $\text{sbp} = 78.69 + 0.72*\text{dbp}$
- For a Medicaid subject, m3 predicts $\text{sbp} = 77.58 + 0.72 \text{ dbp}$
- For a Medicare subject, m3 predicts $\text{sbp} = 80.31 + 0.72 \text{ dbp}$
- For an uninsured subject, m3 predicts $\text{sbp} = 78.74 + 0.72 \text{ dbp}$
- Note: only the intercept term varies by insurance in m3.

Equation for m4 ($\text{sbp} \sim \text{dbp} * \text{insurance}$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 60.26 + 0.94(\text{dbp}) + \\ 23.54(\text{insurance}_{\text{Commercial}}) + 31.04(\text{insurance}_{\text{Medicare}}) + \\ 25.78(\text{insurance}_{\text{Uninsured}}) - 0.29(\text{dbp} \times \text{insurance}_{\text{Commercial}}) - \\ 0.38(\text{dbp} \times \text{insurance}_{\text{Medicare}}) - 0.32(\text{dbp} \times \text{insurance}_{\text{Uninsured}})$$

- m4 predicts, for a Commercial subject...

Equation for m4 ($\text{sbp} \sim \text{dbp} * \text{insurance}$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{\text{sbp}} = 60.26 + 0.94(\text{dbp}) +
23.54(\text{insurance}_{\text{Commercial}}) + 31.04(\text{insurance}_{\text{Medicare}}) +
25.78(\text{insurance}_{\text{Uninsured}}) - 0.29(\text{dbp} \times \text{insurance}_{\text{Commercial}}) -
0.38(\text{dbp} \times \text{insurance}_{\text{Medicare}}) - 0.32(\text{dbp} \times \text{insurance}_{\text{Uninsured}})$$

- m4 predicts, for a Commercial subject...
- $\text{sbp} = 60.26 + 0.94 * \text{dbp} + 23.54 (1) + 31.04 (0) + 25.78 (0) - 0.29 (\text{dbp} * 1) - 0.38 (\text{dbp} * 0) - 0.32 (\text{dbp} * 0)$

Equation for m4 ($sbp \sim dbp * insurance$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{sbp} = 60.26 + 0.94(dbp) +
23.54(insurance_{Commercial}) + 31.04(insurance_{Medicare}) +
25.78(insurance_{Uninsured}) - 0.29(dbp \times insurance_{Commercial}) -
0.38(dbp \times insurance_{Medicare}) - 0.32(dbp \times insurance_{Uninsured})$$

- m4 predicts, for a Commercial subject...
- $sbp = 60.26 + 0.94 * dbp + 23.54 (1) + 31.04 (0) + 25.78 (0) - 0.29 (dbp * 1) - 0.38 (dbp * 0) - 0.32 (dbp * 0)$
- $sbp = (60.26 + 23.54) + (0.94 - 0.29) * dbp$

Equation for m4 ($sbp \sim dbp * insurance$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{sbp} = 60.26 + 0.94(dbp) +
23.54(insurance_{Commercial}) + 31.04(insurance_{Medicare}) +
25.78(insurance_{Uninsured}) - 0.29(dbp \times insurance_{Commercial}) -
0.38(dbp \times insurance_{Medicare}) - 0.32(dbp \times insurance_{Uninsured})$$

- m4 predicts, for a Commercial subject...
- $sbp = 60.26 + 0.94 * dbp + 23.54 (1) + 31.04 (0) + 25.78 (0) - 0.29 (dbp * 1) - 0.38 (dbp * 0) - 0.32 (dbp * 0)$
- $sbp = (60.26 + 23.54) + (0.94 - 0.29) * dbp$
- $sbp = 83.80 - 0.65 dbp$ for Commercial subjects

Equation for m4 ($sbp \sim dbp * insurance$)

```
extract_eq(m4_train, use_coefs = TRUE,  
          wrap = TRUE, terms_per_line = 2)
```

$$\widehat{sbp} = 60.26 + 0.94(dbp) + \\ 23.54(insurance_{Commercial}) + 31.04(insurance_{Medicare}) + \\ 25.78(insurance_{Uninsured}) - 0.29(dbp \times insurance_{Commercial}) - \\ 0.38(dbp \times insurance_{Medicare}) - 0.32(dbp \times insurance_{Uninsured})$$

- For Medicaid subjects, $sbp = 60.26 + 0.94 * dbp$
- For Medicare subjects, $sbp = 91.30 + 0.56 * dbp$
- For the uninsured, $sbp = 86.04 + 0.62 * dbp$
- So both the slope and the intercept are changing in m4

Training Sample Fit Quality

Model m3 (no interaction)

```
glance(m3_train) %>%
  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
0.224	0.22	16.1	5851.1	5878.4

Model m4 (with dbp-insurance interaction)

r.squared	adj.r.squared	sigma	AIC	BIC
0.236	0.229	16	5846	5886.9

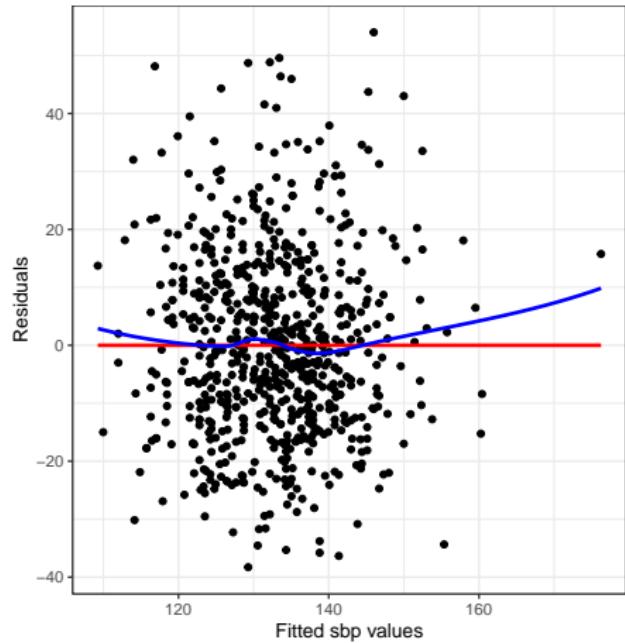
Augmenting and Testing Models m3 and m4

```
m3_train_aug <- augment(m3_train, data = dm_train)
m3_test_aug <- augment(m3_train, newdata = dm_test)

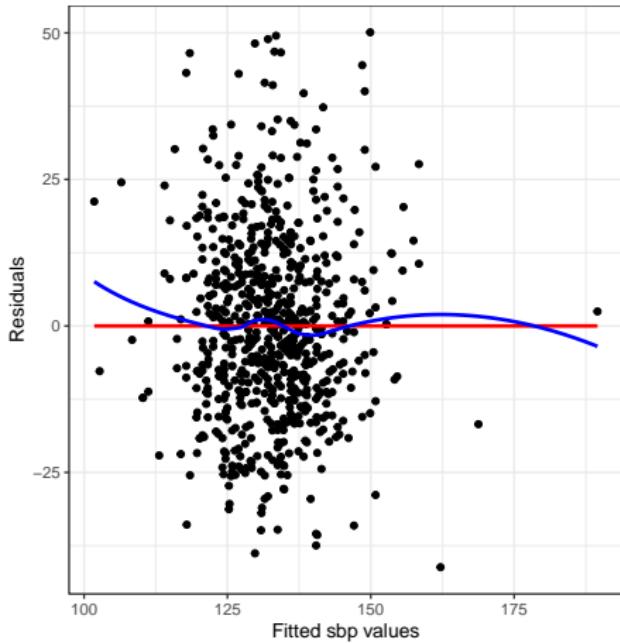
m4_train_aug <- augment(m4_train, data = dm_train)
m4_test_aug <- augment(m4_train, newdata = dm_test)
```

Residuals vs. Fitted Values Plots

Model m3_train



Model m4_train



Comparing performance on the training data

```
bind_rows(glance(m1_train), glance(m2_train),
          glance(m3_train), glance(m4_train)) %>%
  mutate(modname = c("m1", "m2", "m3", "m4")) %>%
  select(modname, r2 = r.squared, adj_r2 = adj.r.squared,
         sigma, AIC, BIC) %>%
  kable(digits = c(0, 3, 3, 2, 1, 1))
```

modname	r2	adj_r2	sigma	AIC	BIC
m1	0.220	0.219	16.12	5848.7	5862.3
m2	NA	NA	16.15	NA	NA
m3	0.224	0.220	16.11	5851.1	5878.4
m4	0.236	0.229	16.02	5846.0	5886.9

- The `glance()` function produces different results for a Bayesian `stan_glm()` model like `m2`, so we'll ignore that for now.

Comparing performance on the test data

Here are some fundamental summaries of absolute prediction error (APE) along with the root mean squared prediction error (RMSPE) for each of our models, in the **testing** sample.

Summary	Mean APE	Max APE	RMSPE
m1_train: lm	12.14	71.07	15.83
m2_train: stan_glm	12.14	71.06	15.83
m3_train: dbp+insurance	12.04	72.37	15.78
m4_train: dbp*insurance	11.95	71.37	15.65

- Which of these models displays the strongest predictive performance in our test sample?

Visualizing Categorical Data in dm1000

8 Categorical Variables from dm1000

```
dm_cat <- dm1000 %>%
  select(subject, sex, residence, insurance,
         tobacco, race_ethnicity, statin, eye_exam)
```

Codebook

- **subject** = ID value (treat as character)
- **sex** = Female or Male (no missing data)
- **insurance** = Medicare, Commercial, Medicaid, Uninsured
- **eye_exam** = 1 for eye examination in past year, else 0
- **statin** = 1 statin prescription in past year, else 0
- **race_ethnicity** = 4 levels (Hispanic or Latinx, Non-Hispanic White, Non-Hispanic Black, Non-Hispanic Asian)
- **residence** = 2 levels (Suburbs, Cleveland), some NA
- **tobacco** = 3 levels (Current, Former, Never), some NA

Using summary()

```
summary(dm_cat)
```

```
> summary(dm_cat)
  subject           insurance      tobacco      statin
Length:1000    Medicaid :330  Current:274  Min.   :0.000
Class :character Commercial:196 Never   :343  1st Qu.:1.000
Mode  :character Medicare :432  Former  :367  Median  :1.000
                  Uninsured : 42   NA's    : 16   Mean    :0.758
                                         NA's    : 16   3rd Qu.:1.000
                                         NA's    : 16   Max.    :1.000
  eye_exam        sex          race_ethnicity      residence
Min.   :0.000  Female:550  Non-Hispanic Black:533  Suburbs  :371
1st Qu.:0.000  Male  :450   Hispanic or Latinx: 91  Cleveland:601
Median  :1.000                    Non-Hispanic White:356  NA's     : 28
Mean    :0.562                    Non-Hispanic Asian: 20
```

Using tabyl to tabulate a categorical variable

```
dm_cat %>% tabyl(tobacco) %>%
  adorn_pct_formatting() %>%
  adorn_totals()
```

	n	percent	valid_percent
Current	274	27.4%	27.8%
Never	343	34.3%	34.9%
Former	367	36.7%	37.3%
<NA>	16	1.6%	-
Total	1000	-	-

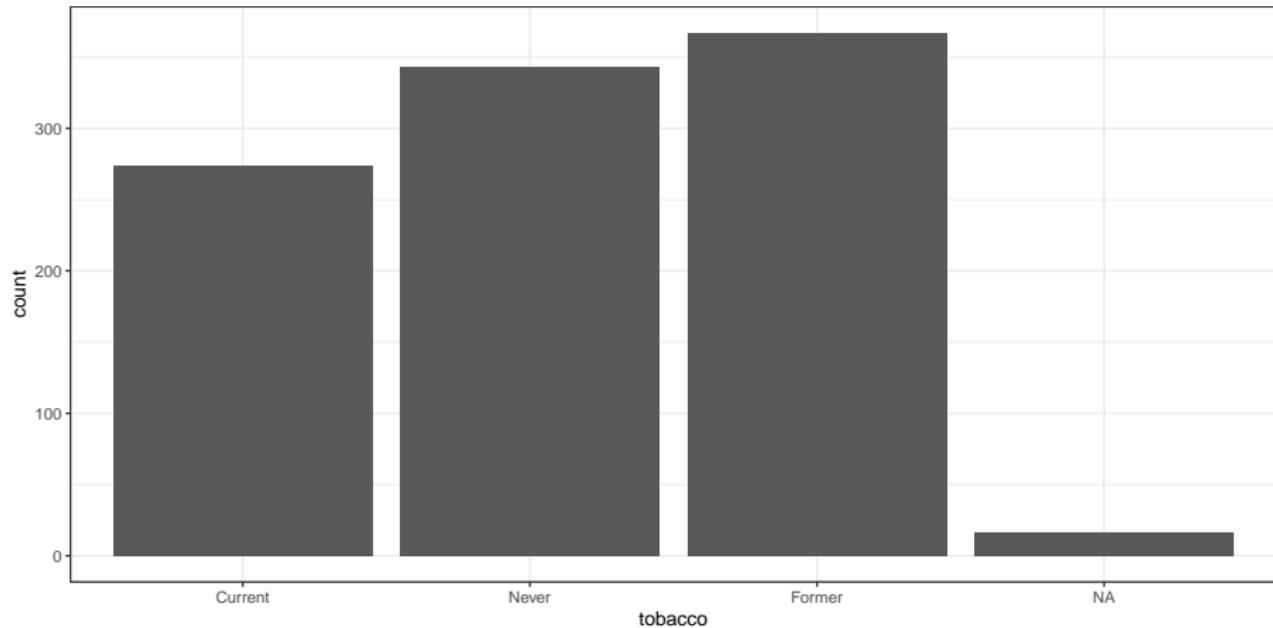
Using count to create a tibble of counts

```
dm_cat %>% count(tobacco)
```

```
# A tibble: 4 x 2
  tobacco     n
  <fct>    <int>
1 Current    274
2 Never      343
3 Former     367
4 <NA>        16
```

Using geom_bar to show a distribution

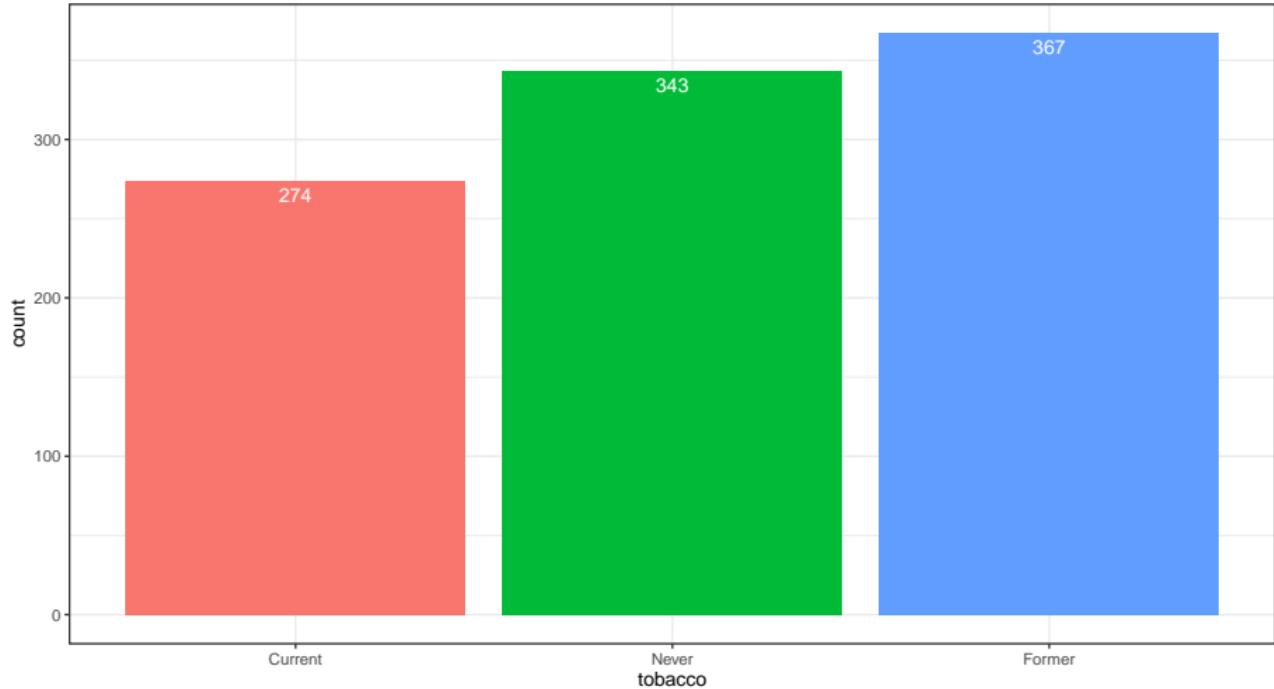
```
ggplot(dm_cat, aes(x = tobacco)) +  
  geom_bar()
```



Augmenting the geom_bar result (code)

```
dm_cat %>% filter(complete.cases(tobacco)) %>%
  ggplot(data = ., aes(x = tobacco, fill = tobacco)) +
  geom_bar() +
  geom_text(aes(label = ..count..), stat = "count",
            vjust = 1.5, col = "white") +
  guides(fill = "none")
```

Augmenting the geom_bar result



Using tabyl to cross-tabulate two variables

```
dm_cat %>% tabyl(insurance, residence) %>%  
  adorn_totals(where = c("row", "col"))
```

	insurance	Suburbs	Cleveland	NA_	Total
Medicaid	114		201	15	330
Commercial	75		119	2	196
Medicare	163		259	10	432
Uninsured	19		22	1	42
Total	371		601	28	1000

Using count to create a tibble of counts

```
dm_cat %>% count(statin, residence)
```

```
# A tibble: 6 x 3
  statin residence     n
  <dbl> <fct>      <int>
1     0 Suburbs      77
2     0 Cleveland    157
3     0 <NA>          8
4     1 Suburbs     294
5     1 Cleveland   444
6     1 <NA>         20
```

Were suburban residents more likely to have a statin prescription?

```
dm_cat %>%
  filter(complete.cases(statin, residence)) %>%
  tabyl(residence, statin)
```

residence	0	1
Suburbs	77	294
Cleveland	157	444

Revise the order of the statin levels, add percentages

```
dm_cat %>% filter(complete.cases(statin, residence)) %>%  
  mutate(statin = fct_relevel(factor(statin), "1", "0")) %>%  
  tabyl(residence, statin)
```

residence	1	0
Suburbs	294	77
Cleveland	444	157

```
dm_cat %>% filter(complete.cases(statin, residence)) %>%  
  mutate(statin = fct_relevel(factor(statin), "1", "0")) %>%  
  tabyl(residence, statin) %>%  
  adorn_percentages(denom = "row") %>%  
  adorn_pct_formatting()
```

residence	1	0
Suburbs	79.2%	20.8%
Cleveland	73.9%	26.1%

Create using table instead

```
tab1 <- dm_cat %>%
  filter(complete.cases(statin, residence)) %>%
  mutate(statin = fct_relevel(factor(statin), "1", "0")) %$%
  table(residence, statin)
```

Assess 2x2 table (results on next slide)

```
Epi::twoby2(tab1)
```

twoby2 results

```
> Epi::twoby2(tab1)
2 by 2 table analysis:
-----
Outcome   : 1
Comparing : Suburbs vs. Cleveland

          1    0    P(1) 95% conf. interval
Suburbs  294  77  0.7925    0.7482    0.8307
Cleveland 444 157  0.7388    0.7022    0.7723

                                95% conf. interval
Relative Risk: 1.0727    0.9996    1.1510
Sample Odds Ratio: 1.3501    0.9903    1.8407
Conditional MLE Odds Ratio: 1.3497    0.9805    1.8679
Probability difference: 0.0537    -0.0018    0.1065

          Exact P-value: 0.0638
          Asymptotic P-value: 0.0577
-----
```

1

A three-by-four two-way table

```
dm_cat %>% filter(complete.cases(tobacco, insurance)) %>%  
  tabyl(tobacco, insurance) %>%  
  adorn_totals(where = c("row", "col"))
```

	tobacco	Medicaid	Commercial	Medicare	Uninsured	Total
Current	118	44	99	13	274	
Never	105	80	140	18	343	
Former	103	70	183	11	367	
Total	326	194	422	42	984	

- 3 rows, 4 columns: hence, this is a 3×4 table
- It's a two-way table, because we are studying the association of two variables (tobacco and insurance)
- Can we compare the insurance percentages by tobacco group?

Compare insurance rates by tobacco group

```
dm_cat %>% filter(complete.cases(tobacco, insurance)) %>%  
  tabyl(tobacco, insurance) %>%  
  adorn_percentages(denominator = "row") %>%  
  adorn_totals(where = "col") %>% kable(digits = 3)
```

tobacco	Medicaid	Commercial	Medicare	Uninsured	Total
Current	0.431	0.161	0.361	0.047	1
Never	0.306	0.233	0.408	0.052	1
Former	0.281	0.191	0.499	0.030	1

- Note that these are actually **proportions** and not percentages.
- Proportions fall between 0 and 1: multiply by 100 for percentages.

Insurance rates by tobacco group?

```
tab2 <- dm_cat %>%
  filter(complete.cases(tobacco, insurance)) %$%
  table(tobacco, insurance)
```

```
tab2
```

		insurance			
tobacco		Medicaid	Commercial	Medicare	Uninsured
Current		118	44	99	13
Never		105	80	140	18
Former		103	70	183	11

```
chisq.test(tab2)
```

Pearson's Chi-squared test

```
data: tab2
```

```
X-squared = 25.592, df = 6, p-value = 0.0002651
```

Using count for three variables

```
dm_cat %>% count(sex, statin, eye_exam)
```

```
# A tibble: 8 x 4
  sex      statin eye_exam     n
  <fct>    <dbl>    <dbl> <int>
1 Female      0        0     68
2 Female      0        1     65
3 Female      1        0    176
4 Female      1        1    241
5 Male        0        0     61
6 Male        0        1     48
7 Male        1        0    133
8 Male        1        1    208
```

A three-way table

```
dm_cat %>% tabyl(statin, residence, sex) %>%
  adorn_title()
```

\$Female

residence

statin	Suburbs	Cleveland	NA_
0	42	87	4
1	160	245	12

\$Male

residence

statin	Suburbs	Cleveland	NA_
0	35	70	4
1	134	199	8

Flattening a three-way table

```
dm_cat %$%  
ftable(sex, residence, statin)
```

		statin	
sex	residence	0	1
Female	Suburbs	42	160
	Cleveland	87	245
Male	Suburbs	35	134
	Cleveland	70	199

- Note that `ftable()` excludes the missing `residence` values by default.

Reminder of Today's Agenda

- ① Developing Four Models for `sbp` using `dbp` (and `insurance`)
- ② Fitting a Bayesian Linear Model with default priors (`m2`)
- ③ Including Insurance without (`m3`) and with (`m4`) interaction with `dbp` in linear models
- ④ Visualizing Categorical Data
- ⑤ Assessing Association in Cross-Tabulations

431 Class 12

thomaselove.github.io/431

2021-09-30

Today's Agenda

- ① Using data from NHANES
- ② A complex data management challenge
- ③ Using dbp to predict sbp again
- ④ Considering a transformation of our outcome

Today's Packages

```
library(NHANES) # for access to NHANES data
library(ggpubr) # to add equation to scatterplot easily
library(equatiomatic)
library(glue)
library(janitor)
library(knitr)
library(broom)
library(magrittr)
library(patchwork)
library(tidyverse)

theme_set(theme_bw())
```

Ingesting and Managing Today's Data

Today's Data

The NHANES data file located in the NHANES package is our source.

NHANES stands for National Health and Nutrition Examination surveys. The NHANES target population is “the non-institutionalized civilian resident population of the United States”. Since 1999, approximately 5,000 individuals of all ages are interviewed in their homes every year and complete the health examination component, in a mobile examination centre.

NHANES and NHANESraw each include 75 variables for the 2009-2010 and 2011-2012 sample years, with complex sampling weights included in NHANESraw. NHANES contains 10,000 rows of data resampled from NHANESraw to undo oversampling effects. NHANES can be treated, for educational purposes, as if it were a simple random sample from the American population.

- For today, we'll do the one thing you should never do with NHANES data, which is to ignore the sampling weights.

Today's Data Ingest and Management

- We'll walk through these steps in the next few slides.

```
set.seed(20210930)
nh12 <- NHANES %>%
  filter(SurveyYr == "2011_12") %>%
  select(ID, BPSysAve, BPDiaAve, Age, Smoke100,
         Race1, HealthGen, SurveyYr) %>%
  rename(Subject = ID, SBP = BPSysAve, DBP = BPDiaAve,
         SROH = HealthGen) %>%
  clean_names() %>%
  mutate(across(where(is.character), as_factor)) %>%
  mutate(subject = as.character(subject)) %>%
  filter(age > 20 & age < 80) %>%
  filter(dbp > 39) %>%
  distinct() %>%
  slice_sample(n = 700) %>%
  droplevels()
```

Today's Data Management: Step 1

- Select the eight variables of interest from NHANES.

```
temp1 <- NHANES %>%
  select(ID, SurveyYr, BPSysAve, BPDiaAve, Age, Smoke100,
         Race1, HealthGen)
# filter(SurveyYr == "2011_12") %>%
# rename(Subject = ID, SBP = BPSysAve, DBP = BPDiaAve,
#        SROH = HealthGen) %>%
# clean_names() %>%
# mutate(across(where(is.character), as_factor)) %>%
# mutate(subject = as.character(subject)) %>%
# filter(age > 20 & age < 80) %>%
# filter(dbp > 39) %>%
# distinct() %>%
# slice_sample(n = 700) %>%
# droplevels()
```

temp1 is a tibble.

temp1

```
# A tibble: 10,000 x 8
  ID SurveyYr BPSysAve BPDiaAve Age Smoke100
  <int> <fct>     <int>     <int> <int> <fct>
1 51624 2009_10      113       85    34 Yes
2 51624 2009_10      113       85    34 Yes
3 51624 2009_10      113       85    34 Yes
4 51625 2009_10      NA        NA     4 <NA>
5 51630 2009_10      112       75    49 Yes
6 51638 2009_10      86        47     9 <NA>
7 51646 2009_10      107       37     8 <NA>
8 51647 2009_10      118       64    45 No
9 51647 2009_10      118       64    45 No
10 51647 2009_10      118       64   45 No
# ... with 9,990 more rows, and 2 more variables:
#   Race1 <fct>, HealthGen <fct>
```

Summarizing temp1

```
> summary(temp1)
      ID       SurveyYr      BPSysAve      BPDiaAve      Age
Min. :51624  2009_10:5000  Min.   : 76.0  Min.   : 0.00  Min.   : 0.00
1st Qu.:56905 2011_12:5000  1st Qu.:106.0  1st Qu.: 61.00  1st Qu.:17.00
Median :62160                           Median :116.0  Median : 69.00  Median :36.00
Mean   :61945                           Mean   :118.2  Mean   : 67.48  Mean   :36.74
3rd Qu.:67039                           3rd Qu.:127.0  3rd Qu.: 76.00  3rd Qu.:54.00
Max.   :71915                           Max.   :226.0  Max.   :116.00  Max.   :80.00
                                         NA's   :1449   NA's   :1449

Smoke100      Race1      HealthGen
No  :4024  Black   :1197  Excellent: 878
Yes :3211  Hispanic: 610  Vgood    :2508
NA's:2765  Mexican :1015  Good     :2956
           White    :6372  Fair     :1010
           Other    : 806  Poor     : 187
                           NA's    :2461
```

Today's Data: Step 2

- Restrict to 2011-12 data, and rename some variables.

```
temp2 <- NHANES %>%
  filter(SurveyYr == "2011_12") %>%
  select(ID, BPSysAve, BPDiaAve, Age,
         Smoke100, Race1, HealthGen, SurveyYr) %>%
  rename(Subject = ID, SBP = BPSysAve,
         DBP = BPDiaAve, SROH = HealthGen)
```

The temp2 tibble

temp2

```
# A tibble: 5,000 x 8
```

	Subject	SBP	DBP	Age	Smoke100	Race1	SROH
	<int>	<int>	<int>	<int>	<fct>	<fct>	<fct>
1	62163	107	37	14	<NA>	Other	Good
2	62172	103	72	43	Yes	Black	Good
3	62174	97	39	80	No	White	Fair
4	62174	97	39	80	No	White	Fair
5	62175	NA	NA	5	<NA>	White	<NA>
6	62176	107	69	34	No	White	Vgood
7	62178	121	72	80	No	White	Fair
8	62180	107	66	35	No	White	Good
9	62186	108	64	17	<NA>	Black	Vgood
10	62190	113	27	15	<NA>	Mexican	Excellent
# ... with 4,990 more rows, and 1 more variable:							
# SurveyYr <fct>							

Summary of temp2

```
> summary(temp2)
  Subject           SBP           DBP           Age       Smoke100
  Min.   :62163   Min.   : 79.0   Min.   : 0.0   Min.   : 0.00   No   :2027
  1st Qu.:64544   1st Qu.:107.0   1st Qu.: 62.0   1st Qu.:17.00   Yes  :1560
  Median :67039   Median :116.0   Median : 69.0   Median :36.00   NA's :1413
  Mean   :67028   Mean   :118.7   Mean   : 68.3   Mean   :36.71
  3rd Qu.:69509   3rd Qu.:128.0   3rd Qu.: 77.0   3rd Qu.:54.00
  Max.   :71915   Max.   :221.0   Max.   :116.0   Max.   :80.00
                  NA's   :719     NA's   :719
  Race1            SROH          SurveyYr
  Black   : 589   Excellent: 486   2009_10:   0
  Hispanic: 350   Vgood    :1278   2011_12:5000
  Mexican : 480   Good     :1485
  White   :3135    Fair     : 472
  Other    : 446   Poor     :   77
                  NA's     :1202
```

Today's Data: Step 3

- Drop unused level (2009-10) from SurveyYr summary with `droplevels()`.
- Clean up the names to lower case with underscores using `clean_names()`.
- Use only distinct observations with `distinct()`.

```
temp3 <- NHANES %>%
  filter(SurveyYr == "2011_12") %>%
  select(ID, BPSysAve, BPDiaAve, Age,
         Smoke100, Race1, HealthGen, SurveyYr) %>%
  rename(Subject = ID, SBP = BPSysAve,
         DBP = BPDiaAve, SROH = HealthGen) %>%
  clean_names() %>%
  distinct() %>%
  droplevels()
```

The temp3 tibble

temp3

```
# A tibble: 3,211 x 8
```

	subject	sbp	dbp	age	smoke100	race1	sroh
	<int>	<int>	<int>	<int>	<fct>	<fct>	<fct>
1	62163	107	37	14	<NA>	Other	Good
2	62172	103	72	43	Yes	Black	Good
3	62174	97	39	80	No	White	Fair
4	62175	NA	NA	5	<NA>	White	<NA>
5	62176	107	69	34	No	White	Vgood
6	62178	121	72	80	No	White	Fair
7	62180	107	66	35	No	White	Good
8	62186	108	64	17	<NA>	Black	Vgood
9	62190	113	27	15	<NA>	Mexican	Excellent
10	62199	110	65	57	Yes	White	Vgood
# ... with 3,201 more rows, and 1 more variable:							
# survey_yr <fct>							

Summary of temp3

```
> summary(temp3)
    subject      sbp       dbp       age     smoke100
  Min. :62163  Min.   : 79.0  Min.   : 0.00  Min.   : 0.00  No   :1244
  1st Qu.:64541 1st Qu.:106.0  1st Qu.: 61.00  1st Qu.:14.00  Yes  : 929
  Median :67027  Median :116.0  Median : 69.00  Median :33.00  NA's:1038
  Mean   :67013  Mean   :118.5  Mean   : 67.35  Mean   :35.07
  3rd Qu.:69457 3rd Qu.:128.0  3rd Qu.: 77.00  3rd Qu.:54.00
  Max.   :71915  Max.   :221.0  Max.   :116.00  Max.   :80.00
                  NA's   :547    NA's   :547
    race1        sroh      survey_yr
  Black   : 514  Excellent:283  2011_12:3211
  Hispanic: 274  Vgood    :732
  Mexican : 390  Good     :911
  White   :1667   Fair     :338
  Other   : 366   Poor    : 60
                  NA's   :887
```

Today's Data: Step 4

- Make Race1 and HealthGen into factors, leave ID as character.
- Restrict Age to 21-79, and require DBP ≥ 40 mm Hg.

```
temp4 <- NHANES %>%
  filter(SurveyYr == "2011_12") %>%
  select(ID, BPSysAve, BPDiaAve, Age, Smoke100,
         Race1, HealthGen, SurveyYr) %>%
  rename(Subject = ID, SBP = BPSysAve, DBP = BPDiaAve,
         SROH = HealthGen) %>%
  clean_names() %>%
  mutate(across(where(is.character), as_factor)) %>%
  mutate(subject = as.character(subject)) %>%
  filter(age > 20 & age < 80) %>%
  filter(dbp > 39) %>%
  distinct() %>%
  droplevels()
```

The temp4 tibble

temp4

```
# A tibble: 1,906 x 8
  subject    sbp    dbp    age smoke100 race1    sroh
  <chr>    <int> <int> <int> <fct>    <fct>    <fct>
1 62172      103     72     43 Yes     Black    Good
2 62176      107     69     34 No      White   Vgood
3 62180      107     66     35 No      White   Good
4 62199      110     65     57 Yes     White   Vgood
5 62205      122     87     28 No      White   Good
6 62206      106     50     35 No      White   <NA>
7 62208      105     59     38 No      Hispanic Good
8 62209      108     57     62 No      Mexican Fair
9 62220      120     71     31 No      Black   Good
10 62222     104     73     32 No      White   Good
# ... with 1,896 more rows, and 1 more variable:
#   survey_yr <fct>
```

Summarizing the temp4 tibble

```
> summary(temp4)
   subject           sbp          dbp          age      smoke100
Length:1906    Min.   : 81.0   Min.   : 41.00  Min.   :21.00  No :1082
Class :character 1st Qu.:109.0  1st Qu.: 65.00  1st Qu.:32.00  Yes: 824
Mode  :character Median :119.0  Median : 72.00  Median :45.00
                  Mean   :120.9  Mean   : 71.88  Mean   :45.93
                  3rd Qu.:129.0  3rd Qu.: 79.00  3rd Qu.:58.00
                  Max.   :221.0  Max.   :116.00  Max.   :79.00
   race1            sroh        survey_yr
Black   : 306  Excellent:199  2011_12:1906
Hispanic: 153  Vgood     :525
Mexican : 191  Good      :684
white   :1038   Fair      :272
Other   : 218   Poor      : 53
                  NA's      :173
```

Today's Data: Select random sample of 700

```
set.seed(20210930)
nh12 <- NHANES %>%
  filter(SurveyYr == "2011_12") %>%
  select(ID, BPSysAve, BPDiaAve, Age, Smoke100,
         Race1, HealthGen, SurveyYr) %>%
  rename(Subject = ID, SBP = BPSysAve, DBP = BPDiaAve,
         SROH = HealthGen) %>%
  clean_names() %>%
  mutate(across(where(is.character), as_factor)) %>%
  mutate(subject = as.character(subject)) %>%
  filter(age > 20 & age < 80) %>%
  filter(dbp > 39) %>%
  distinct() %>%
  slice_sample(n = 700) %>%
  droplevels()
```

The nh12 tibble

nh12

```
# A tibble: 700 x 8
```

	subject	sbp	dbp	age	smoke100	race1	sroh
	<chr>	<int>	<int>	<int>	<fct>	<fct>	<fct>
1	71420	126	69	54	No	Mexican	Good
2	64368	136	74	70	Yes	Black	Vgood
3	62546	150	84	64	Yes	Mexican	Good
4	70531	110	73	49	No	Black	Excelle~
5	62974	98	74	30	No	White	Good
6	66294	143	77	76	Yes	White	Good
7	68762	104	76	34	Yes	White	<NA>
8	70758	125	64	21	Yes	Hispanic	<NA>
9	71315	132	84	44	Yes	White	Good
10	66600	137	72	64	No	Black	Vgood
# ... with 690 more rows, and 1 more variable:							
# survey_yr <fct>							

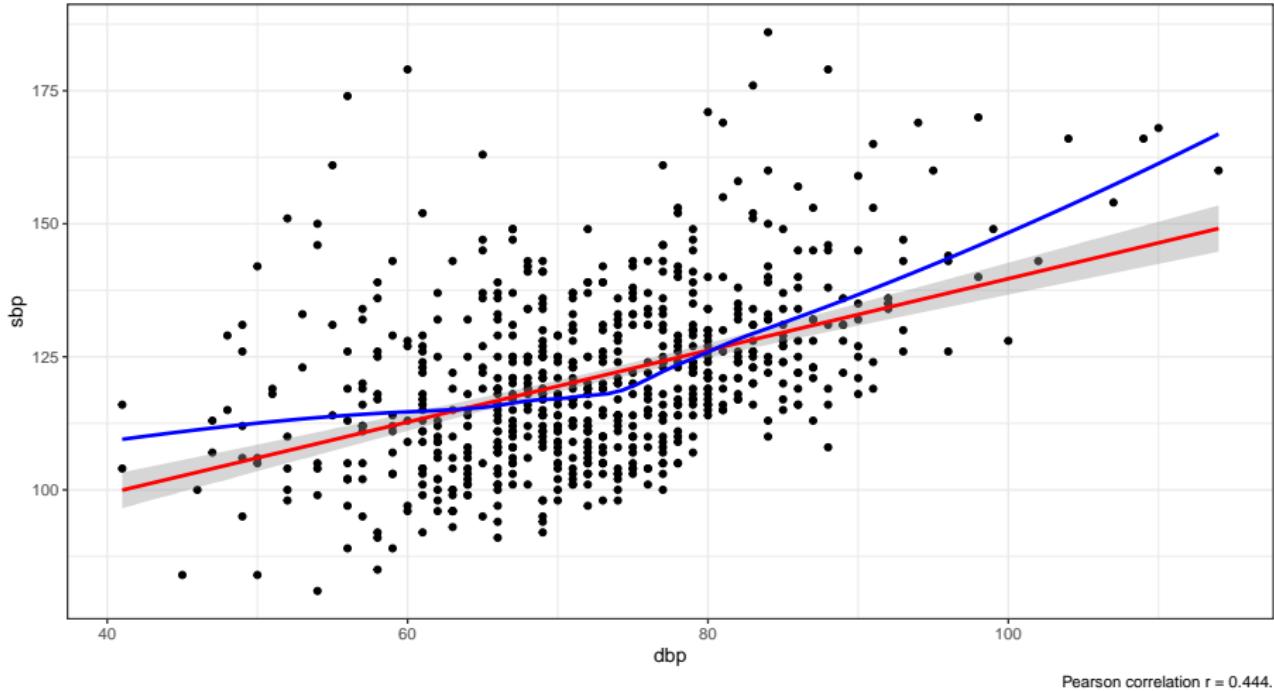
Summary of the nh12 tibble

```
> summary(nh12)
  subject           sbp          dbp         age      smoke100
  Length:700    Min.   : 81   Min.   :41.00  Min.   :21.00  No :376
  Class :character 1st Qu.:110   1st Qu.:66.00  1st Qu.:32.00  Yes:324
  Mode  :character Median :119   Median :72.00   Median :45.00
                    Mean   :121   Mean   :72.31   Mean   :45.62
                    3rd Qu.:130   3rd Qu.:79.00   3rd Qu.:58.00
                    Max.   :186   Max.   :114.00  Max.   :78.00
  race1            sroh        survey_yr
  Black   :115  Excellent: 83  2011_12:700
  Hispanic: 61  Vgood    :196
  Mexican : 73  Good     :241
  White   :367  Fair     : 92
  Other    : 84  Poor     : 21
                  NA's     : 67
```

- Outcome (quantitative): sbp
- Quantitative predictors: dbp, age
- Binary predictor: smoke100 (Yes/No)
- 5-category predictor: race1 (White, Black, Hispanic, Mexican, Other)
- 5-category predictor with missing data: sroh (E, VG, G, F,)
- Identification code: subject

Building Regression Model m1 for sbp

Visualizing sbp against dbp



Model m1

```
m1 <- lm(sbp ~ dbp, data = nh12)

tidy(m1, 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)	72.33	3.76	66.13	78.52
dbp	0.67	0.05	0.59	0.76

```
glance(m1) %>%
  select(r.squared, adj.r.squared, sigma, AIC, BIC, nobs) %>%
  kable(digits = c(3,3,1,1,1,0))
```

r.squared	adj.r.squared	sigma	AIC	BIC	nobs
0.197	0.196	14.3	5717.6	5731.2	700

Model m1

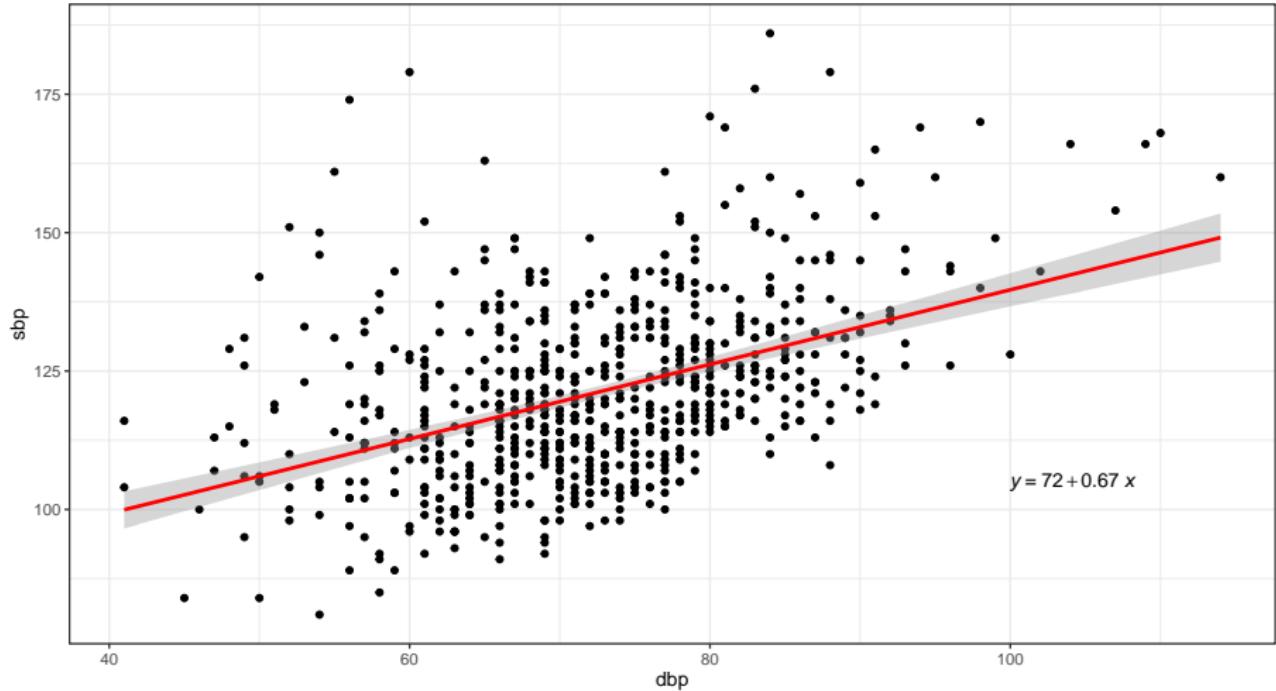
```
extract_eq(m1, use_coefs = TRUE, coef_digits = 3)
```

$$\widehat{sbp} = 72.326 + 0.673(dbp)$$

To include the equation in the scatterplot, I might use
stat_regrline_equation() from the ggpubr package.

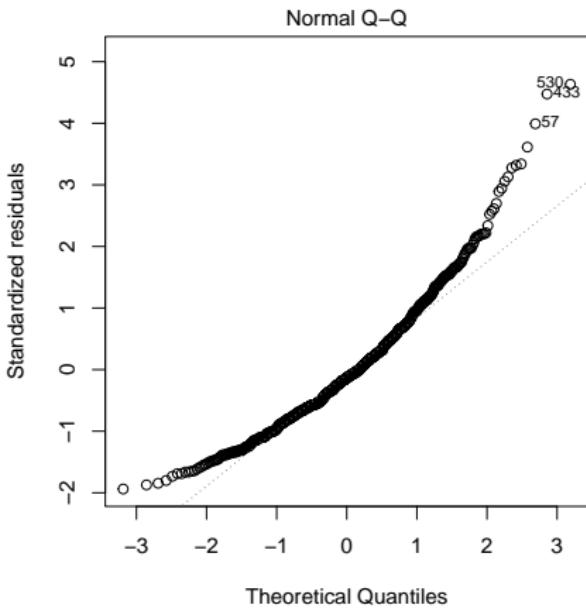
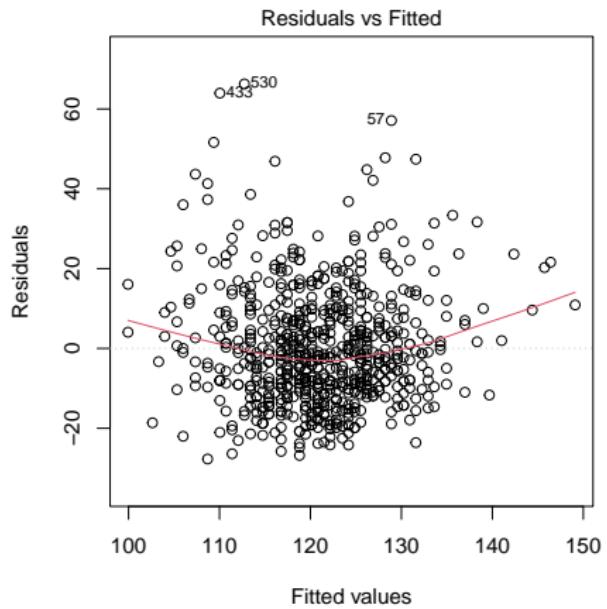
```
ggplot(nh12, aes(x = dbp, y = sbp)) +  
  geom_point() +  
  geom_smooth(method = "lm", col = "red", formula = y ~ x) +  
  stat_regrline_equation(label.x = 100, label.y = 105)
```

Including the Equation in the Scatterplot



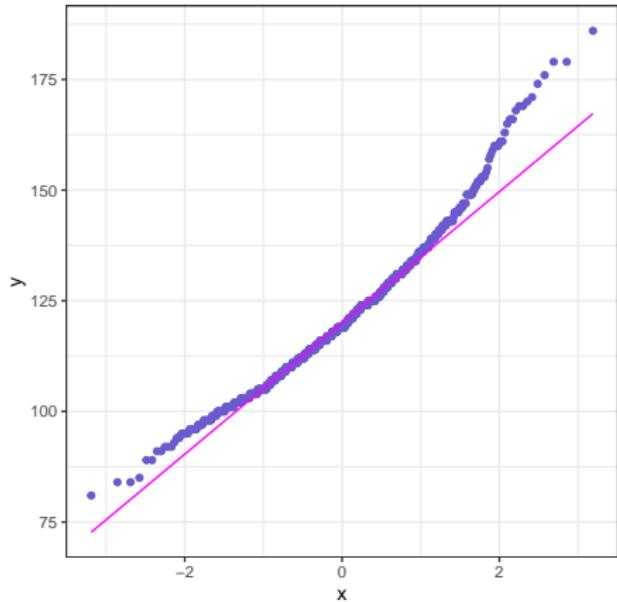
Quick Residual Plots for Model m1

```
par(mfrow = c(1,2))
plot(m1, which = c(1:2)); par(mfrow = c(1,1))
```

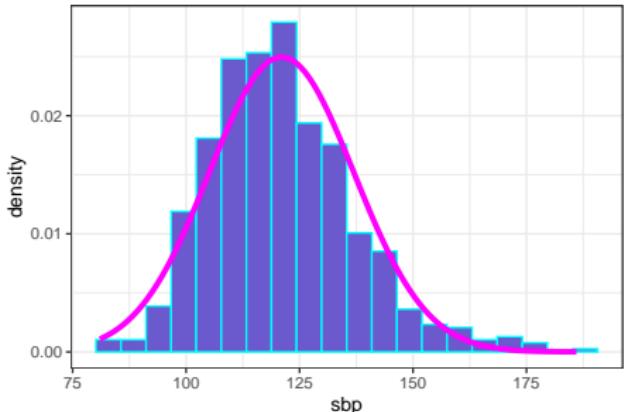


Should we think about transforming sbp here?

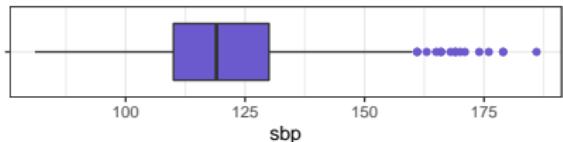
Normal Q-Q plot: nh12 sbp



Density Function: nh12 sbp



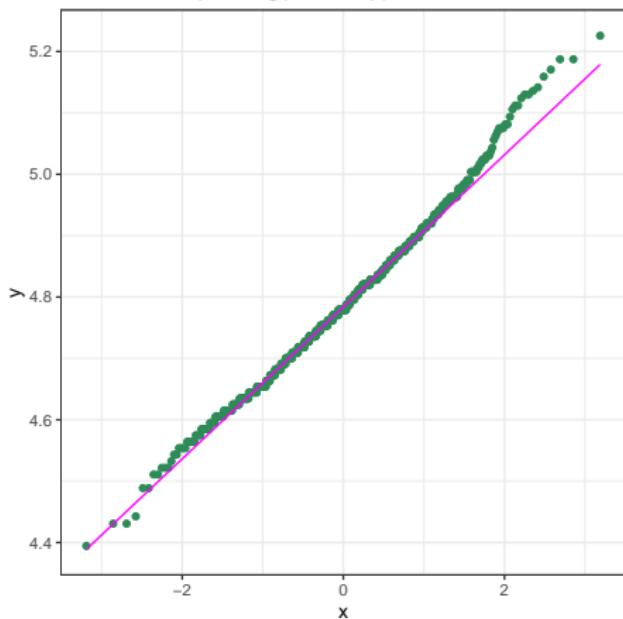
Boxplot: nh12 sbp



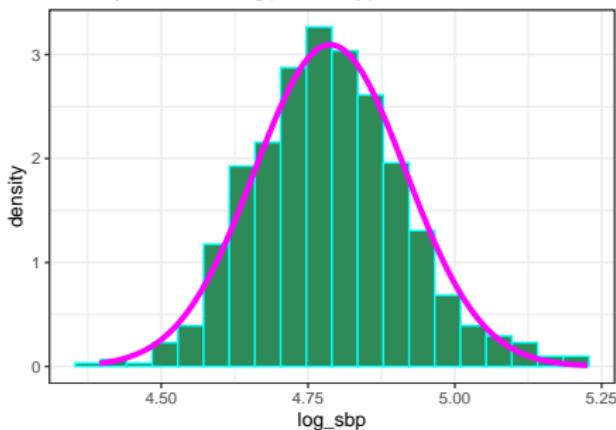
	min	Q1	median	Q3	max	mean	sd	n	missing
	81	110	119	130	186	121	16	700	0

Logarithm of sbp?

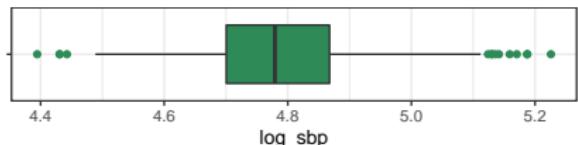
Normal Q-Q plot: $\log(\text{nh12 sbp})$



Density Function: $\log(\text{nh12 sbp})$

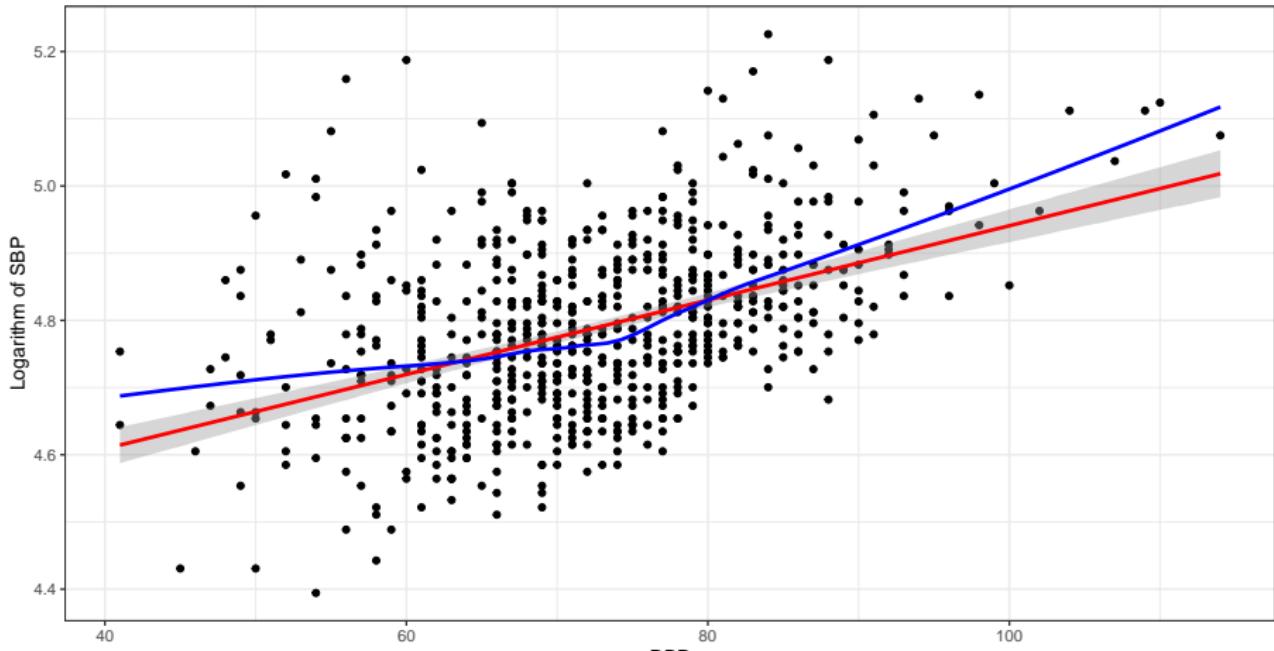


Boxplot: $\log(\text{nh12 sbp})$



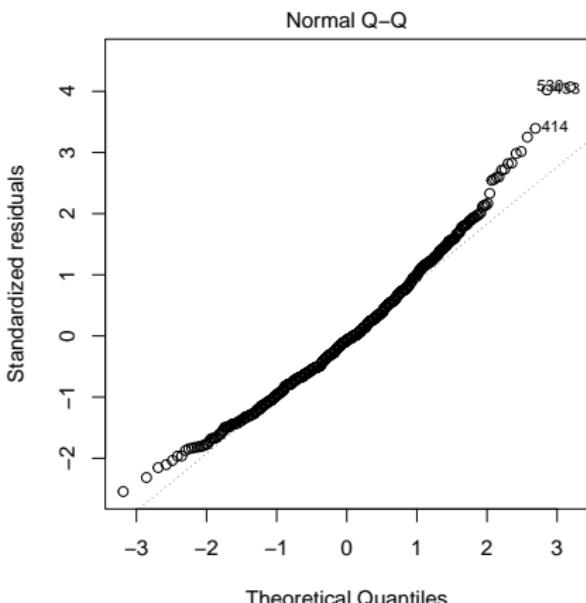
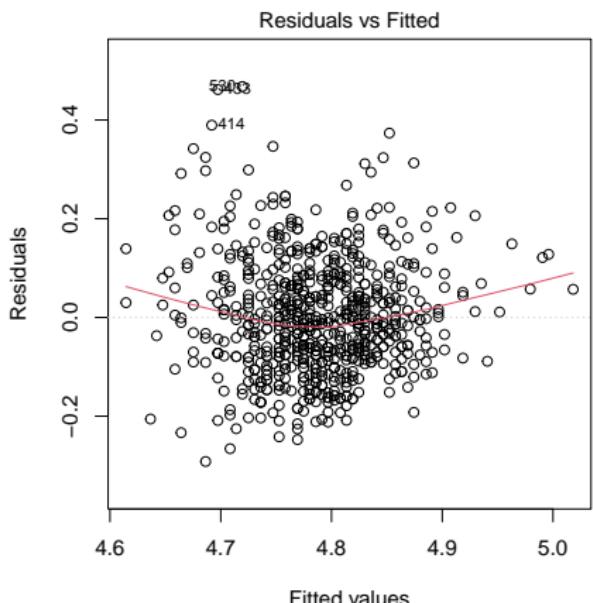
	min	Q1	median	Q3	max	mean	sd	n	missing
	4.4	4.7	4.8	4.9	5.2	4.8	0.1	700	0

Scatterplot of log(sbp) vs. dbp



Quick Residual Plots for Model m2

```
m2 <- lm(log(sbp) ~ dbp, data = nh12)
par(mfrow = c(1,2))
plot(m2, which = c(1:2)); par(mfrow = c(1,1))
```



Today's Agenda

- ① Using data from NHANES
- ② A complex data management challenge
- ③ Using dbp to predict sbp again
- ④ Considering a transformation of our outcome