

431 Class 06

`thomaseLove.github.io/431`

2020-09-10

Today's Agenda

Working with the NHANES (`nh2`) data we built last time, focusing today on systolic and diastolic blood pressures for our sample of 1000 adults.

- How did we build that tibble again, in R?
- How should we explore these data before modeling?
 - What can we learn about the center, spread, outliers, and shape of quantitative data?
 - Are these blood pressure data well described by a Normal distribution?
- How might we start to look at Associations between our quantities?
 - Scatterplots, Correlation, Linear Models, Smoothing

Loading our R Packages

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

theme_set(theme_bw())
```

We'll also load another package later that I'll keep secret for now.

Creating the nh2 data set (from Tuesday)

This is a lot of code, and I've changed something subtle. Slower?

```
set.seed(20200908) # very important to get same result later
```

```
nh2 <- NHANES %>%  
  filter(SurveyYr == "2011_12") %>%  
  select(ID, SurveyYr, Age, Height, Weight, BMI, Pulse,  
         SleepHrsNight, BPSysAve, BPDiaAve, Gender,  
         PhysActive, SleepTrouble, Smoke100,  
         Race1, HealthGen, Depressed) %>%  
  rename(SleepHours = SleepHrsNight, Sex = Gender,  
         SBP = BPSysAve, DBP = BPDiaAve) %>%  
  filter(Age > 20 & Age < 80) %>%  
  drop_na() %>% # removes all rows with NA  
  slice_sample(., n = 1000) %>% # sample 1000 rows  
  clean_names() # from the janitor package (snake case)
```

Steps in creating the nh2 data set (tibble)

- 1 Set a seed so we can get the same sample when we rerun it later.

```
set.seed(20200908)
```

- 2 Start with the NHANES data contained in the NHANES package we loaded earlier.

```
nh2 <- NHANES %>%
```

At this point, we have 10,000 rows (subjects) and 76 columns (variables)

- 3 Restrict (filter) our subjects (rows) to those from Survey Year 2011-12

```
filter(SurveyYr == "2011_12") %>%
```

This reduces our sample to 5,000 rows and 76 columns

Steps in creating the nh2 data set (tibble)

- 4 Select the variables we will be using in our analyses

```
select(ID, SurveyYr, Age, Height, Weight, BMI, Pulse,  
       SleepHrsNight, BPSysAve, BPDiaAve, Gender,  
       PhysActive, SleepTrouble, Smoke100,  
       Race1, HealthGen, Depressed) %>%
```

Now at 5,000 rows and 17 columns (variables)

- 5 Rename a few to make the names more useful

```
rename(SleepHours = SleepHrsNight, Sex = Gender,  
       SBP = BPSysAve, DBP = BPDiaAve) %>%
```

Still 5,000 rows and 17 columns, just with the new names SleepHours, Sex, SBP and DBP.

Steps in creating the nh2 data set (tibble)

- ⑥ Restrict our data to adults ages 21-79

```
filter(Age > 20 & Age < 80) %>%
```

Now down to 3,347 rows and 17 columns

- ⑦ Drop all observations with any missing values on our selected variables

```
drop_na() %>%
```

Now we have 2,936 rows and 17 columns with no missing data

Steps in creating the nh2 data set (tibble)

- ⑧ Sample a random set of 1,000 rows (without replacement)

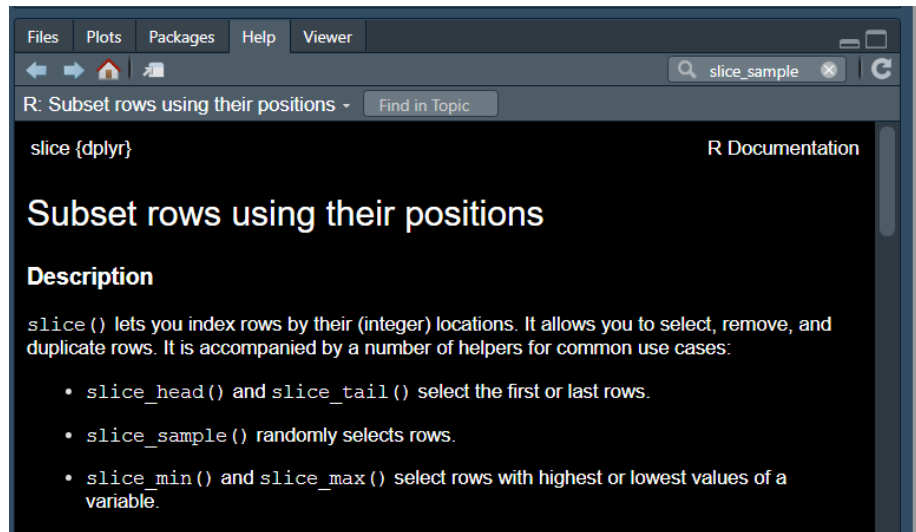
```
slice_sample(., n = 1000)
```

This is a newer approach to sampling than the old one I'd used in our last class. The old method has been superceded by this new approach in the tidyverse, so we'll use it going forward.

- It yields the same sample as we generated on Tuesday, so we'll still call the eventual tibble we build `nh2`.

How do we get help on a new function?

Try the Help window and type in the command, or just type `?(slice_sample)` into the Console...



The screenshot shows the R Help window with the 'Help' tab selected. The search bar contains 'slice_sample'. The main content area displays the documentation for the `slice` function from the `dplyr` package. The title 'Subset rows using their positions' is prominently displayed. Below it, the 'Description' section explains that `slice()` allows indexing rows by their integer locations. A bulleted list provides details on specific helpers: `slice_head()` and `slice_tail()` for first/last rows, `slice_sample()` for random selection, and `slice_min()` and `slice_max()` for selecting rows by value.

Files Plots Packages Help Viewer

slice_sample

R: Subset rows using their positions - Find in Topic

slice {dplyr} R Documentation

Subset rows using their positions

Description

`slice()` lets you index rows by their (integer) locations. It allows you to select, remove, and duplicate rows. It is accompanied by a number of helpers for common use cases:

- `slice_head()` and `slice_tail()` select the first or last rows.
- `slice_sample()` randomly selects rows.
- `slice_min()` and `slice_max()` select rows with highest or lowest values of a variable.

Steps in creating the nh2 data set (tibble)

- 9 Clean up the names to use “snake case”

The first few “old” names were:

- ID, SurveyYr, Age, Height, Weight, BMI, Pulse, SleepHours

The first few “new” names are:

- id, survey_yr, age, height, weight, bmi, pulse, sleep_hours

I like this change, personally. It helps me know what to expect and have to remember fewer details.

- `clean_names` can be used to do other sorts of cleaning, too.

All 9 steps: Creating the nh2 data set

```
set.seed(20200908) # very important to get same result later

nh2 <- NHANES %>%
  filter(SurveyYr == "2011_12") %>%
  select(ID, SurveyYr, Age, Height, Weight, BMI, Pulse,
         SleepHrsNight, BPSysAve, BPDiaAve, Gender,
         PhysActive, SleepTrouble, Smoke100,
         Race1, HealthGen, Depressed) %>%
  rename(SleepHours = SleepHrsNight, Sex = Gender,
         SBP = BPSysAve, DBP = BPDiaAve) %>%
  filter(Age > 20 & Age < 80) %>%
  drop_na() %>% # removes all rows with NA
  slice_sample(., n = 1000) %>% # sample 1000 rows
  clean_names() # from the janitor package (snake case)
```

Why do we have to set a seed?

```
set.seed(431431) # change the seed to a new number

nh2_newseed <- NHANES %>%
  filter(SurveyYr == "2011_12") %>%
  select(ID, SurveyYr, Age, Height, Weight, BMI, Pulse,
         SleepHrsNight, BPSysAve, BPDiaAve, Gender,
         PhysActive, SleepTrouble, Smoke100,
         Race1, HealthGen, Depressed) %>%
  rename(SleepHours = SleepHrsNight, Sex = Gender,
         SBP = BPSysAve, DBP = BPDiaAve) %>%
  filter(Age > 20 & Age < 80) %>%
  drop_na() %>% # removes all rows with NA
  slice_sample(., n = 1000) %>% # sample 1000 rows
  clean_names() # from the janitor package (snake case)
```

Look at the data sets?

```
nh2 %>% select(id, age, sbp, dbp) %>% head(3) # show first 3
```

```
# A tibble: 3 x 4
  id    age    sbp    dbp
<int> <int> <int> <int>
1 66817    72   128    47
2 64157    68   131    70
3 69618    64   147    79
```

```
nh2_newseed %>% select(id, age, sbp, dbp) %>% head(3)
```

```
# A tibble: 3 x 4
  id    age    sbp    dbp
<int> <int> <int> <int>
1 63463    61   134    82
2 65462    49   116    77
3 68080    39   107    66
```

Compare summaries from the data sets?

```
nh2 %>% tabyl(smoke100) %>% adorn_pct_formatting()
```

smoke100	n	percent
No	544	54.4%
Yes	456	45.6%

```
nh2_newseed %>% tabyl(smoke100) %>% adorn_pct_formatting()
```

smoke100	n	percent
No	527	52.7%
Yes	473	47.3%

So maintaining the same seed is the way we get the same sample.

Codebook for nh2 (ID and Quantitative Variables)

Name	Description
id	Identifying code for each subject
survey_yr	2011_12 for all, indicates administration date
age	Age in years at screening of subject (must be 21-79)
height	Standing height in cm
weight	Weight in kg
bmi	Body mass index ($\frac{weight}{(height_{meters})^2}$ in $\frac{kg}{m^2}$)
pulse	60 second pulse rate
sleep_hrs	Self-reported hours (usually gets) per night
sbp	Systolic Blood Pressure (mm Hg)
dbp	Diastolic Blood Pressure (mm Hg)

Codebook for nh2 (Categorical Variables)

Binary Variables

Name	Levels	Description
sex	F, M	Sex of study subject
phys_active	No, Yes	Moderate or vigorous sports/recreation?
sleep_trouble	No, Yes	Has told a provider about trouble sleeping?
smoke100	No, Yes	Smoked at least 100 cigarettes in lifetime?

Multi-Categorical Variables

Name	Levels	Description
race1	5	Self-reported Race/Ethnicity
health_gen	5	Self-reported overall general health
depressed	3	How often subject felt depressed in last 30d

Today's Questions

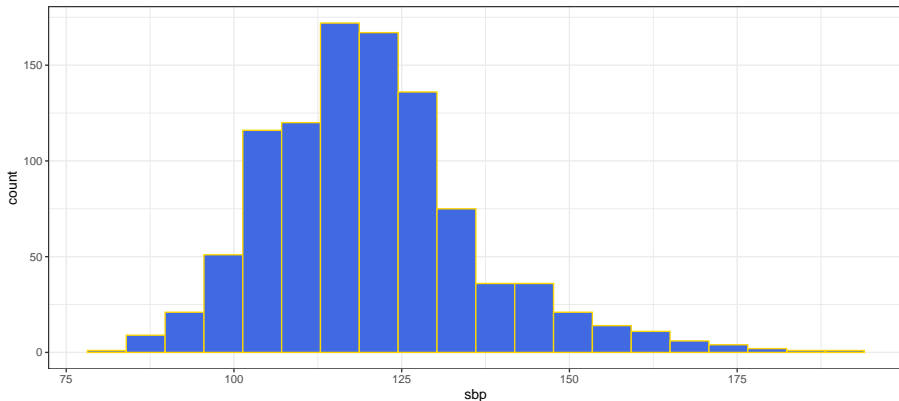
- 1 What is the nature of the association between systolic BP and diastolic BP in these NHANES subjects?
- 2 How might we explore the blood pressure data to understand it better before we address the association in question 1?

Today's Variables

Name	Description
id	Identifying code for each subject
sbp	Systolic Blood Pressure (mm Hg)
dbp	Diastolic Blood Pressure (mm Hg)

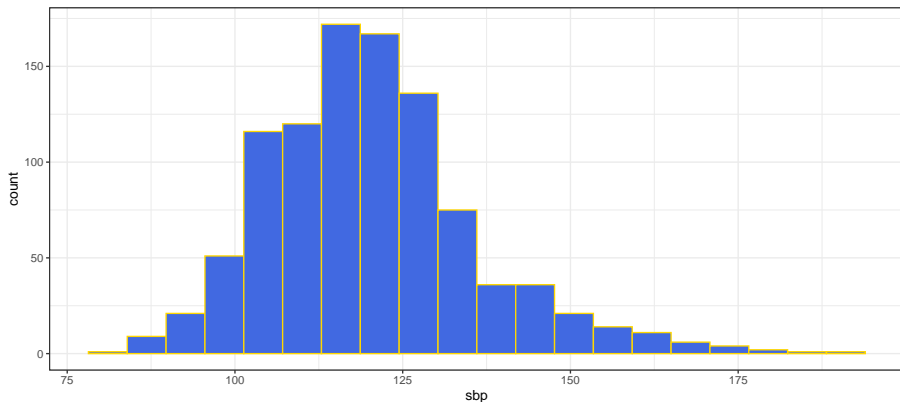
Want to Summarize Systolic BP data? DTDP

```
ggplot(data = nh2, aes(x = sbp)) +  
  geom_histogram(bins = 20, fill = "royalblue", col = "gold")
```



Describing a Distribution

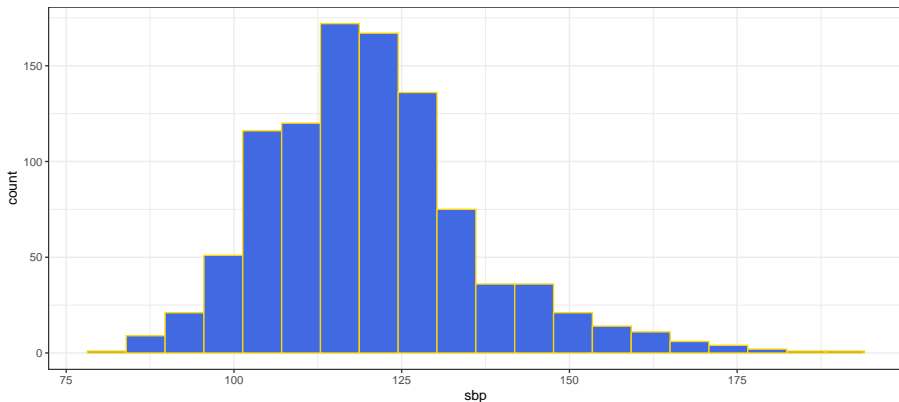
1 Where is the **center** of the distribution?



What else might we look at to improve our response?

Describing a Distribution

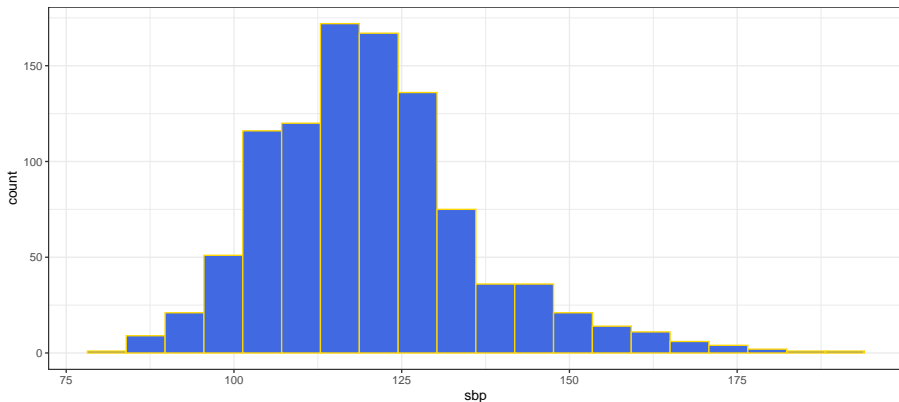
② What is the **spread/dispersion** of the distribution?



What else might we look at to improve our response?

Describing a Distribution

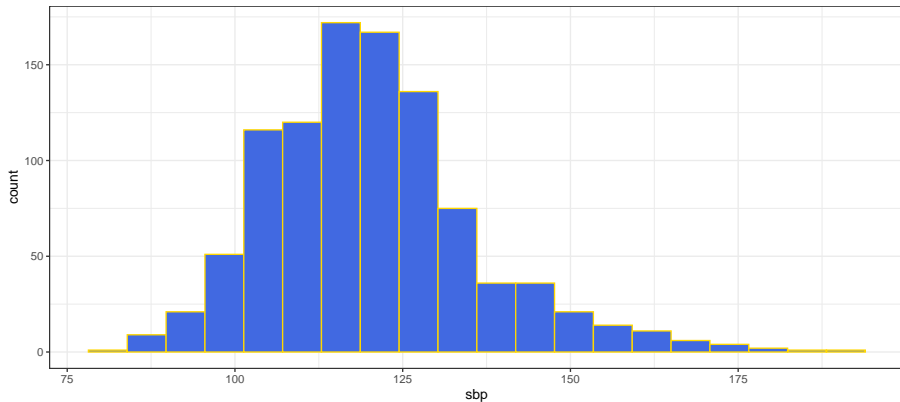
- ③ Are there any **outliers** / **unusual values** we should investigate?



What else might we look at to improve our response?

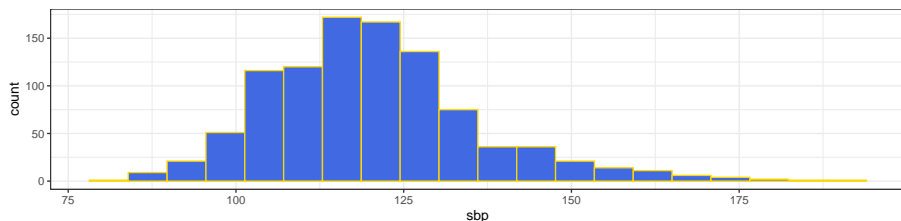
Describing a Distribution

- ④ What is the **shape** of the distribution?



- What kind of shapes should we be thinking about?

Shape Options

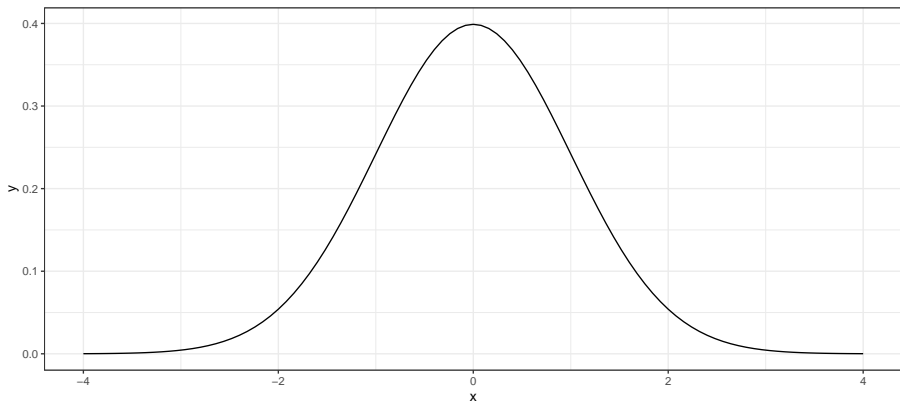


- Is this a unimodal distribution, with one clear peak?
- Are the data symmetric, so that if we placed an imaginary line in the center of the distribution, we'd see mirror images?
- If the data appear skewed, in which direction compared to a “Normal” distribution?
 - Right skew = a longer right tail (more clustered data on the left)

So what does a “Normal” distribution look like?

The “Normal” or “Gaussian” distribution

First, I'll show a Normal distribution with mean 0 and standard deviation 1.

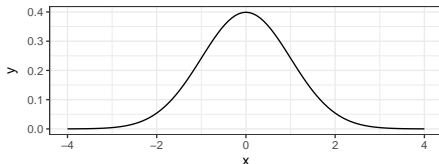


All Normal distributions differ only in their mean (center) and standard deviation (spread), and thus not in terms of their shape.

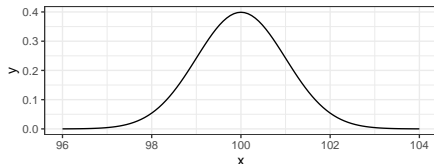
The “Normal” or “Gaussian” distribution

Here we'll show a few examples for illustration.

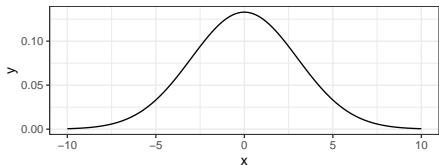
Normal with mean 0, sd 1



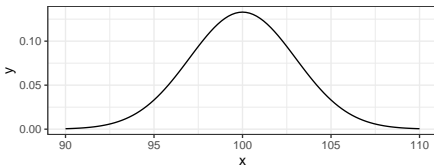
Normal with mean 100, sd 1



Normal with mean 0, sd 3



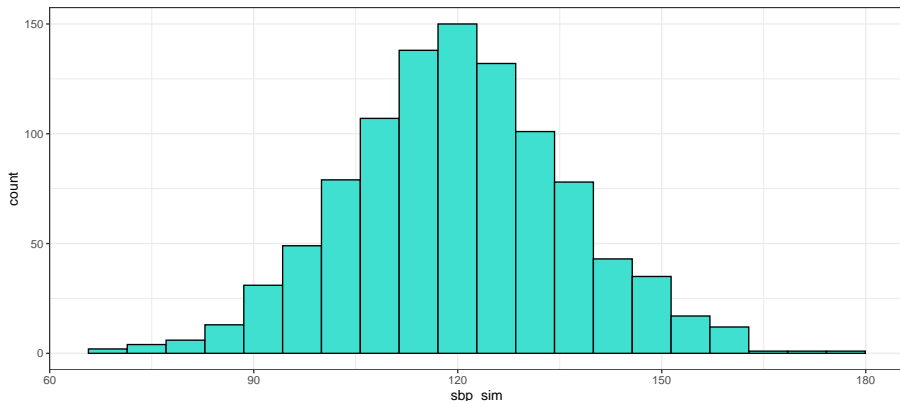
Normal with mean 100, sd 3



- Note the changes in the x axis for each distribution.

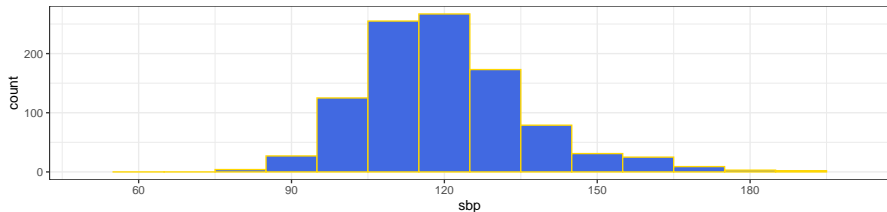
Simulating a “Normal” distribution of our SBP

Here's a histogram of 1000 observations I simulated from a Normal distribution with the same mean (120.5) and standard deviation (15.8) as our `nh2` sample of systolic blood pressures.

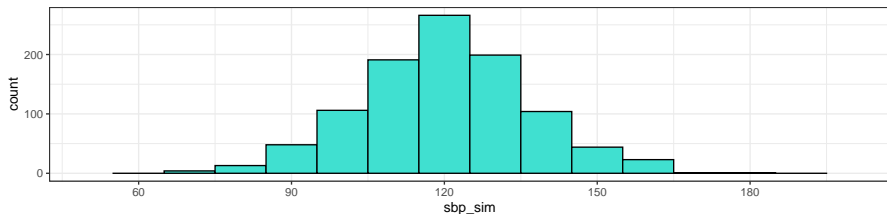


Does the Normal approximate our nh2 SBPs well?

1000 Observed SBP values from nh2 (sample mean = 120.5, sd = 15.8)



1000 Simulated Values from Normal distribution with mean = 120.5, sd = 15.8



Simulated Data from a Normal distribution

Here's some of the code I used to simulate and build a histogram of 1000 observations drawn from a Normal distribution with the same mean (120.5) and standard deviation (15.8) as our `nh2` sample of systolic blood pressures.

```
set.seed(2020)
temp <- tibble(sbp_sim =
               rnorm(n = 1000, mean = 120.5, sd = 15.8))

ggplot(temp, aes(x = sbp_sim)) +
  geom_histogram(binwidth = 10,
                 fill = "turquoise", col = "black")
```

Of course, I added a title, too, in the version on the previous slide, and did some work to make the limits and tick marks on the x axis match up across the two plots. You have the code in the R Markdown file if you want it.

Summarizing the Systolic BP distribution

```
mosaic::favstats(~ sbp, data = nh2) %>% kable(digits = 1)
```

min	Q1	median	Q3	max	mean	sd	n	missing
81	110	119	128	191	120.5	15.8	1000	0

- **n** = number of non-missing values for this variable
- **missing** = number of missing values for this variable
- **Mean** = arithmetic average of the values (sum / number of values)
- **SD** = standard deviation of the values (larger SD = more spread/dispersion in the data distribution) = also equal to the square root of the variance

If the data follow an (approximately) Normal distribution, then interpreting the Mean and SD becomes much easier than if they don't.

If the data are Normally distributed

- then the Mean and the Median should be in the same place
- about 68% of the data will be within 1 standard deviation of the mean
- about 95% of the data will be within 2 standard deviations of the mean
- about 99.7% of the data will be within 3 standard deviations of the mean

but if the data aren't symmetric, with a bell-shaped curve, then...

- the interpretation of the mean relative to the median changes (in light of skew)
- the interpretation of a standard deviation is not as well connected to the percentages listed above

What else can we learn about Systolic BP?

```
mosaic::favstats(~ sbp, data = nh2) %>% kable(digits = 1)
```

min	Q1	median	Q3	max	mean	sd	n	missing
81	110	119	128	191	120.5	15.8	1000	0

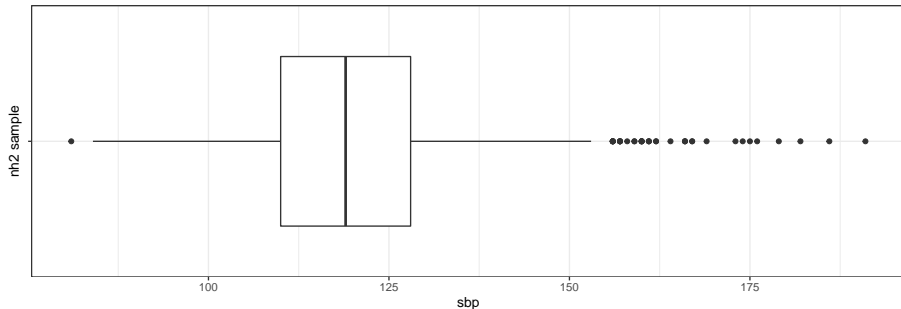
- Five Number Summary (describes five key percentiles of the data)

Minimum	Q1 = 1st Quartile	Median	Q3 = 3rd Quartile	Maximum
P0	P25	P50	P75	P100

- Several measures of spread are derived from these percentiles
 - Range = Maximum - Minimum
 - IQR = Q3 - Q1 (range of the middle half of the distribution)
- The middle three percentiles form the box in a boxplot.

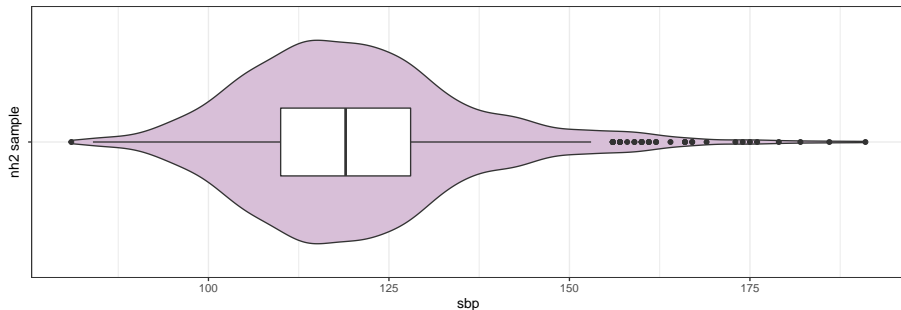
Boxplot of Systolic BP for all subjects

```
ggplot(nh2, aes(x = "", y = sbp)) +  
  geom_boxplot() + coord_flip() +  
  labs(x = "nh2 sample")
```



Add the Violin Plot?

```
ggplot(nh2, aes(x = "", y = sbp)) +  
  geom_violin(fill = "thistle") +  
  geom_boxplot(width = 0.3) +  
  coord_flip() + labs(x = "nh2 sample")
```



More Extensive Numerical Summaries?

We could try

```
nh2 %>% Hmisc::describe(sbp)
```

- but that will throw an error message, specifically `Error in describe.data.frame(., sbp) : object 'sbp' not found.` What is wrong? How can we fix that?

More Extensive Numerical Summaries?

We could try

```
nh2 %>% Hmisc::describe(sbp)
```

- but that will throw an error message, specifically `Error in describe.data.frame(., sbp) : object 'sbp' not found.` What is wrong? How can we fix that?
- We could drop the pipe and use `$` notation, so `Hmisc::describe(nh2$sbp)`

More Extensive Numerical Summaries?

We could try

```
nh2 %>% Hmisc::describe(sbp)
```

- but that will throw an error message, specifically `Error in describe.data.frame(., sbp) : object 'sbp' not found.` What is wrong? How can we fix that?
- We could drop the pipe and use `$` notation, so `Hmisc::describe(nh2$sbp)`
- Another option is to change the pipe (to the `%$%` pipe available in the `magrittr` package): `nh2 %$% Hmisc::describe(sbp)`

What do these summaries tell us?

```
nh2 %$% Hmisc::describe(sbp)
```

sbp

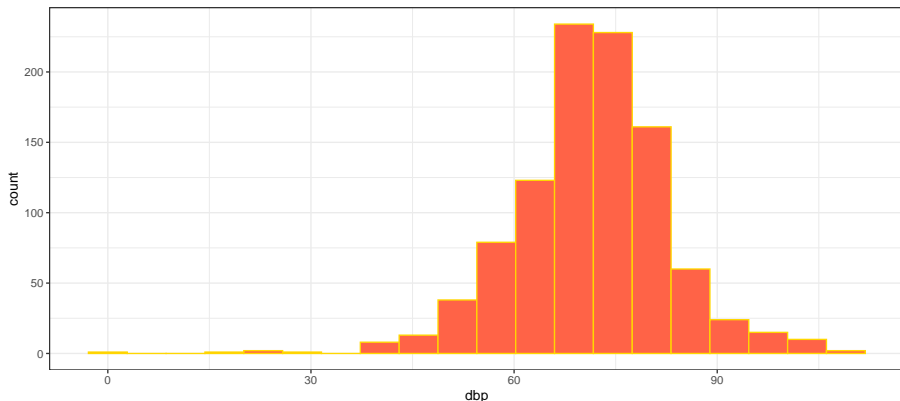
n	missing	distinct	Info	Mean	Gmd
1000	0	87	0.999	120.5	17.27
.05	.10	.25	.50	.75	.90
98.0	102.9	110.0	119.0	128.0	141.0
.95					
151.0					

lowest : 81 84 85 86 89, highest: 176 179 182 186 191

- Gmd = Gini's mean difference (a robust measure of spread) = mean absolute difference between any pairs of observations. Larger Gmd indicates more spread.
- Info = a measure of relative information describing how “continuous” the data are. Higher Info indicates fewer ties.

OK, what about Diastolic Blood Pressure?

```
ggplot(data = nh2, aes(x = dbp)) +  
  geom_histogram(bins = 20, fill = "tomato", col = "gold")
```



- Center? Spread? Outliers? Shape?

Numerical Summary of dbp?

```
mosaic::favstats(~ dbp, data = nh2) %>% kable(digits = 1)
```

min	Q1	median	Q3	max	mean	sd	n	missing
0	65	71.5	78	109	71.3	11.5	1000	0

Hmisc::describe for dbp?

```
nh2 %$% Hmisc::describe(dbp)
```

dbp

n	missing	distinct	Info	Mean	Gmd
1000	0	71	0.999	71.27	12.39
.05	.10	.25	.50	.75	.90
52.95	58.00	65.00	71.50	78.00	84.00
.95					
89.00					

lowest : 0 15 23 24 30, highest: 104 105 106 108 109

What is a plausible diastolic blood pressure?

Stem-and-Leaf of dbp values?

```
stem(nh2$dbp)
```

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

```
0 | 0
1 | 5
2 | 34
3 | 09
4 | 111122344455577778889999999
5 | 00111122222222223333344444444445555555555566666666666
6 | 0000000000000000000000001111111111111111111122222222222
7 | 000000000000000000000000000000000000011111111111111111
8 | 000000000000000000000000000000000000111111111111111112222222
9 | 000000000112223334455556666688999
10 | 0223333456689
```

Who are those people with tiny dbp values?

```
nh2 %>%  
  filter(dbp < 40) %>%  
  select(id, sbp, dbp)
```

```
# A tibble: 6 x 3  
   id    sbp    dbp  
  <int> <int> <int>  
1 68528   133    39  
2 71575   121    15  
3 64507   130    24  
4 62649   122    23  
5 71598    86    30  
6 70664   152     0
```

Let's reset.

```
nh2_new <- nh2 %>%  
  filter(dbp > 39)
```

```
nrow(nh2)
```

```
[1] 1000
```

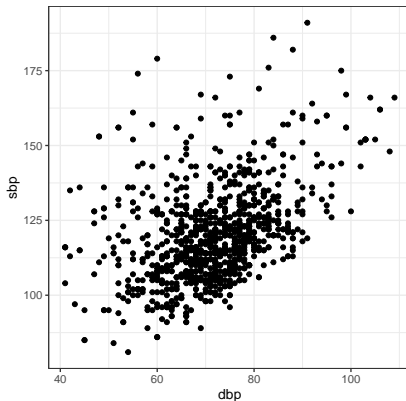
```
nrow(nh2_new)
```

```
[1] 994
```

We'll work with `nh2_new` for the rest of today. Now, finally, let's address the issue of the relationship between `sbp` and `dbp` in these subjects

Scatterplot is the place to start

```
ggplot(nh2_new, aes(x = dbp, y = sbp)) +  
  geom_point() +  
  theme(aspect.ratio = 1) # make the plot square for slide
```



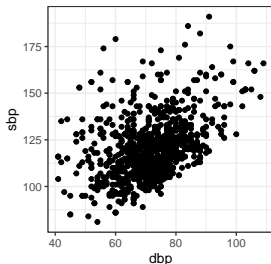
Numerical Summary: Pearson Correlation

The Pearson correlation 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.

Correlation in our sbp-dbp scatterplot?

```
ggplot(nh2_new, aes(x = dbp, y = sbp)) +  
  geom_point() + theme(aspect.ratio = 1)
```



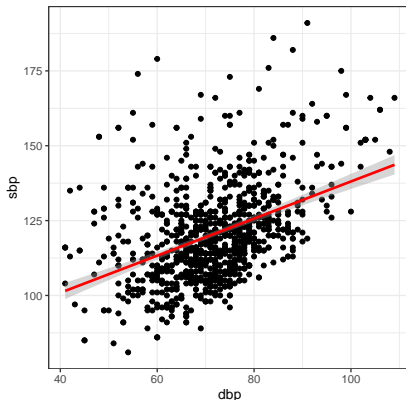
```
nh2_new %>% cor(sbp, dbp)
```

```
[1] 0.4227224
```

What does a correlation of +0.42 imply about a linear fit to the data?

Try to predict sbp from dbp?

```
ggplot(nh2_new, aes(x = dbp, y = sbp)) +  
  geom_point() + theme(aspect.ratio = 1) +  
  geom_smooth(method = "lm", formula = y ~ x,  
             col = "red", se = TRUE)
```



What line is being fit there?

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

```
m1 <- lm(sbp ~ dbp, data = nh2_new)
m1
```

Call:

```
lm(formula = sbp ~ dbp, data = nh2_new)
```

Coefficients:

(Intercept)	dbp
76.1260	0.6193

Model is **sbp = 76.13 + 0.62 dbp**. What do the slope and intercept mean?

Linear Model m_1 : $\text{sbp} = 76.13 + 0.62 \text{ dbp}$

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

- Is that reasonable in this setting?

0.62 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 if the slope was zero?

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

Nope.

```
library(rstanarm)
```

Loading required package: Rcpp

This is rstanarm version 2.21.1

- See <https://mc-stan.org/rstanarm/articles/priors> for changes
- Default priors may change, so it's safest to specify priors,
- For execution on a local, multicore CPU with excess RAM we r
options(mc.cores = parallel::detectCores())

Fit linear model using stan_glm?

```
m2 <- stan_glm(sbp ~ dbp, data = nh2_new)
```

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)

Chain 1: Iteration: 1000 / 2000 [50%] (Warmup)

Chain 1: Iteration: 1001 / 2000 [50%] (Sampling)

Bayesian fitted linear model for our sbp data

```
print(m2)
```

```
stan_glm
family:      gaussian [identity]
formula:     sbp ~ dbp
observations: 994
predictors:  2
```

	Median	MAD_SD
(Intercept)	76.1	3.0
dbp	0.6	0.0

Auxiliary parameter(s):

	Median	MAD_SD
sigma	14.3	0.3

Is the Bayesian model very different from our `lm` in this situation?

```
coef(m1)
```

(Intercept)	dbp
76.1259634	0.6193499

```
coef(m2)
```

(Intercept)	dbp
76.1260226	0.6193974

Does R like this linear model?

```
summary(m1)$coefficients
```

	Estimate	Std. Error	t value
(Intercept)	76.1259634	3.05128313	24.94884
dbp	0.6193499	0.04215776	14.69124

	Pr(> t)
(Intercept)	5.013872e-107
dbp	2.344029e-44

Yes. Wow. It **really** does. Look at those p values (listed as $\text{Pr}(>|t|)$)!

$2.34\text{e-}44$ is just scientific notation: 2.34×10^{-44} .

- I'll note that this is many orders of magnitude smaller than what we can usually deal with without rounding issues.
- R often reports (effectively) zero values with $2.2\text{e-}16$.

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

```
summary(m1)$r.squared
```

```
[1] 0.1786942
```

- This R-squared value says something about the proportion of the variation in our sbp that can be accounted for by the linear model we've built using dbp.
- About 18% in this case. Is that good?
- Why is this called R-squared? What is the R?

```
nh2_new %$% cor(sbp, dbp)
```

```
[1] 0.4227224
```

```
nh2_new %$% cor(sbp, dbp)^2
```

```
[1] 0.1786942
```

But is a linear model really the right choice?

```
ggplot(nh2_new, aes(x = dbp, y = sbp)) +  
  geom_point() + theme(aspect.ratio = 1) +  
  geom_smooth(method = "loess", formula = y ~ x,  
             col = "blue", se = TRUE)
```

