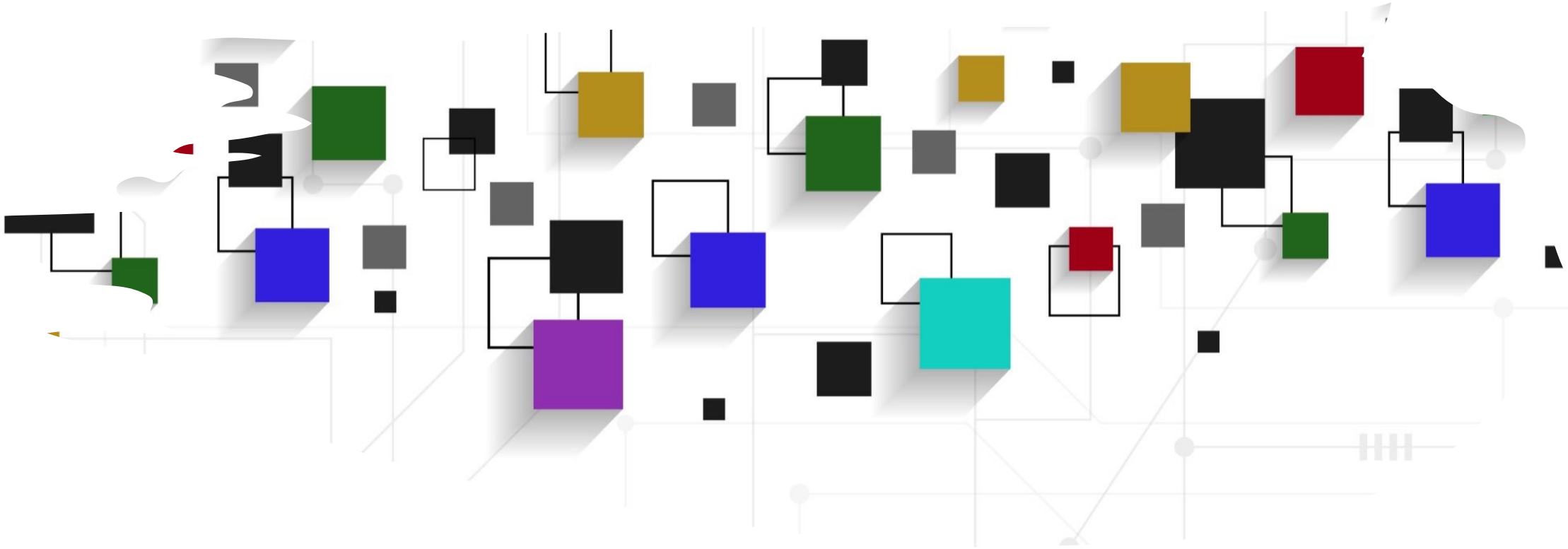


CogLab: Making Inferences

WEEK 10

recap: Oct 31, 2023

- what we covered:
  - manipulating data using tidyverse verbs
- your to-do's were:
  - *apply:* formative assignment #2 (R descriptive)
  - *send:* experiment for piloting

today's agenda

- association data analysis
- statistical inferences

creating an association scoring sheet

- four possible cues were presented
- each cue has six possible valid responses
- each response can be congruent / incongruent for a given cue
- the type of association can be direct / shared / random for a given cue-response

dodish | foobly | geck | mipp

apple | dodish | foobly

geck | horse | mipp

read in scoring sheet

- new heading # association
- read in the scoring sheet and view the dataframe
- what are congruent responses?
- what is a direct association?
- what is a random association?

```
# association

```{r}

scoring = read_csv("association_scoring.csv")%>%
  arrange(cue, response)
```

cue	response	congruence	type_of_association	cue_type
dodish	apple	incongruent	direct	adjective
dodish	dodish	repeat	random	adjective
dodish	foobly	incongruent	random	adjective
dodish	geck	congruent	direct	adjective
dodish	horse	congruent	direct	adjective
dodish	mipp	incongruent	direct	adjective
foobly	apple	congruent	direct	adjective
foobly	dodish	incongruent	random	adjective
foobly	foobly	repeat	random	adjective
foobly	geck	incongruent	direct	adjective
```

merging two dataframes

- we want to merge our association data with this scoring sheet
- first, filter for association trials
- select relevant columns
- compare association trials to scoring data
- to merge, we need at least one shared column between two dataframes
- potential problems?

```
association_trials = savic %>%
  filter(typeoftrial == "association")
```

```
association_trials = savic %>%
  filter(typeoftrial == "association") %>%
  select(ID, revised_response, cue)
```

first-R-notebook.Rmd* | association_trials

Filter

	ID	revised_response	cue
1	5418680	Apple	foobly
2	5418680	Apple	foobly
3	5418680	Dodish	geck
4	5418680	Dodish	geck
5	5418680	Dodish	geck

cue	response	congruence	type_of_association	cue_type
dodish	apple	incongruent	direct	adjective
dodish	dodish	repeat	random	adjective
dodish	foobly	incongruent	random	adjective
dodish	geck	congruent	direct	adjective
dodish	horse	congruent	direct	adjective
dodish	mipp	incongruent	direct	adjective
foobly	apple	congruent	direct	adjective
foobly	dodish	incongruent	random	adjective
foobly	foobly	repeat	random	adjective
foobly	geck	incongruent	direct	adjective

setting up for merging

- rename() the response column
- convert to lowercase

```
association_trials = savic %>%
  filter(typeoftrial == "association") %>%
  select(ID, revised_response, cue) %>%
  rename(response = "revised_response")%>%
  mutate(response = tolower(response))
```

first-R-notebook.Rmd* | association_trials

Filter

	ID	revised_response	cue
1	5418680	Apple	foobly
2	5418680	Apple	foobly
3	5418680	Dodish	geck
4	5418680	Dodish	geck
5	5418680	Dodish	geck

first-R-notebook.Rmd | association_trials

Filter

	ID	response	cue
1	5418680	apple	foobly
2	5418680	apple	foobly
3	5418680	dodish	geck
4	5418680	dodish	geck
5	5418680	dodish	geck
6	5418680	dodish	geck
7	5418680	dodish	geck

tidyverse: left_join()

- left_join() allows you to merge additional columns from a different dataframe to your dataframe, by matching on common column names and values

```
association_trials = savic %>%
  filter(typeoftrial == "association") %>%
  select(ID, revised_response, cue) %>%
  rename(response = "revised_response")%>%
  mutate(response = tolower(response)) %>%
  left_join(scoring)
```

ID	response	cue	congruence	type_of_association	cue_type
5418680	apple	foobly	congruent	direct	adjective
5418680	apple	foobly	congruent	direct	adjective
5418680	dodish	geck	congruent	direct	noun
5418680	dodish	geck	congruent	direct	noun
5418680	dodish	geck	congruent	direct	noun
5418680	dodish	geck	congruent	direct	noun
5418680	dodish	geck	congruent	direct	noun

computing congruence

- first, we remove NA trials
- keep only congruent/incongruent trials
- keep only direct/shared associations

```
congruence_trials = association_trials %>%
  filter(!is.na(congruence))%>%
  filter(congruence %in% c("congruent", "incongruent")) %>%
  filter(type_of_association %in% c("direct", "shared"))
```

congruence counts

- create new dataframe called congruence_counts
- group by ID, congruent, association type, and cue type and compute a count

```
congruence_counts = congruence_trials %>%
  group_by(ID, cue_type,  congruence,type_of_association) %>%
  count()
```

ID	cue_type	congruence	type_of_association	n
5418680	adjective	congruent	direct	18
5418680	noun	congruent	direct	18
46356924	adjective	congruent	direct	15
46356924	adjective	incongruent	direct	2
46356924	noun	congruent	direct	5
46356924	noun	incongruent	direct	12
46356924	noun	incongruent	shared	1

congruence proportions

ID	cue_type	congruence	type_of_association	n
5418680	adjective	congruent	direct	18
5418680	noun	congruent	direct	18
46356924	adjective	congruent	direct	15
46356924	adjective	incongruent	direct	2
46356924	noun	congruent	direct	5
46356924	noun	incongruent	direct	12
46356924	noun	incongruent	shared	1

- next, group by ID and cue type and compute a proportion

```
congruence_counts = congruence_trials %>%
  group_by(ID, cue_type,  congruence,type_of_association) %>%
  count() %>%
  group_by(ID, cue_type) %>%
  mutate(proportion = n / sum(n))
```

ID	cue_type	congruence	type_of_association	n	proportion
5418680	adjective	congruent	direct	18	1.00000000
5418680	noun	congruent	direct	18	1.00000000
46356924	adjective	congruent	direct	15	0.88235294
46356924	adjective	incongruent	direct	2	0.11764706
46356924	noun	congruent	direct	5	0.27777778
46356924	noun	incongruent	direct	12	0.66666667
46356924	noun	incongruent	shared	1	0.05555556

correcting for guessing

- we could just look at the proportion of trials that were congruent
- but this doesn't account for incongruent trials (or guessing)
- we want to subtract the proportion of incongruent trials from congruent trials

```
congruence_counts %>%
  filter(congruence == "congruent") %>%
  ungroup()%>%
  summarise(mean_prop = mean(proportion))
```

```
A tibble: 1 × 1
  mean_prop
      <dbl>
      0.860
```

long vs. wide data

- data is often in 2 main formats:
  - long
  - wide
- long data has multiple rows indicating each observation
- wide data has multiple columns indicating each observation

long

ID	cue_type	congruence	type_of_association	n	proportion
5418680	adjective	congruent	direct	18	1.00000000
5418680	noun	congruent	direct	18	1.00000000
46356924	adjective	congruent	direct	15	0.88235294
46356924	adjective	incongruent	direct	2	0.11764706
46356924	noun	congruent	direct	5	0.27777778
46356924	noun	incongruent	direct	12	0.66666667
46356924	noun	incongruent	shared	1	0.05555556

ID	cue_type	type_of_association	congruent	incongruent
5418680	adjective	direct	1.00000000	NA
5418680	noun	direct	1.00000000	NA
46356924	adjective	direct	0.88235294	0.11764706
46356924	noun	direct	0.27777778	0.66666667

wide

converting to wide format

ID	cue_type	congruence	type_of_association	n	proportion
5418680	adjective	congruent	direct	18	1.00000000
5418680	noun	congruent	direct	18	1.00000000
46356924	adjective	congruent	direct	15	0.88235294
46356924	adjective	incongruent	direct	2	0.11764706
46356924	noun	congruent	direct	5	0.27777778
46356924	noun	incongruent	direct	12	0.66666667
46356924	noun	incongruent	shared	1	0.05555556

- select relevant columns
- pivot_wider()
- specifies which columns to make wide and where to get the values from

```
wide_counts = congruence_counts %>%
  select(ID, cue_type,congruence, type_of_association, proportion) %>%
  pivot_wider(names_from = congruence, values_from = proportion)
```

ID	cue_type	type_of_association	congruent	incongruent
5418680	adjective	direct	1.00000000	NA
5418680	noun	direct	1.00000000	NA
46356924	adjective	direct	0.88235294	0.11764706
46356924	noun	direct	0.27777778	0.66666667

filling empty columns

- use mutate() to fill up NA values with 0s
- create new proportion column that computes difference between congruent and incongruent proportions
- mean of prop column?

ID	cue_type	type_of_association	congruent	incongruent
5418680	adjective	direct	1.00000000	NA
5418680	noun	direct	1.00000000	NA
46356924	adjective	direct	0.88235294	0.11764706
46356924	noun	direct	0.27777778	0.66666667

```
wide_counts = congruence_counts %>%
  select(ID, cue_type,congruence, type_of_association, proportion) %>%
  pivot_wider(names_from = congruence, values_from = proportion)%>%
  mutate(incongruent = ifelse(is.na(incongruent), 0, incongruent),
         congruent = ifelse(is.na(congruent), 0, congruent))
```

	ID	cue_type	type_of_association	congruent	incongruent
1	5418680	adjective	direct	1.00000000	0.00000000
2	5418680	noun	direct	1.00000000	0.00000000
3	46356924	adjective	direct	0.88235294	0.11764706
4	46356924	noun	direct	0.27777778	0.66666667

```
wide_counts = congruence_counts %>%
  select(ID, cue_type,congruence, type_of_association, proportion) %>%
  pivot_wider(names_from = congruence, values_from = proportion)%>%
  mutate(incongruent = ifelse(is.na(incongruent), 0, incongruent),
         congruent = ifelse(is.na(congruent), 0, congruent))%>%
  mutate(prop = congruent - incongruent)
```

ID	cue_type	type_of_association	congruent	incongruent	prop
5418680	adjective	direct	1.00000000	0.00000000	1.00000000
5418680	noun	direct	1.00000000	0.00000000	1.00000000
46356924	adjective	direct	0.88235294	0.11764706	0.76470588

```
mean(wide_counts$prop)
```

going back to the analysis description

- what proportion of trials are congruent?

In the free association task, participants were asked to respond to the prompt word with one of the training triad words. They responded as instructed on an average 96% of the free association trials presented at the end of training. In addition, they tended to respond with training words that had directly co-occurred with the prompt word. Whereas 81% of participants' responses were based on direct co-occurrence, only 2% were based on shared co-occurrence regularities.[3]

```
> mean(wide_counts$prop)
[1] 0.7194747
```

computing association proportions

- write, run, and interpret the code

first-R-notebook.Rmd × | association_type_occurrence ×

Filter

	ID	cue_type	direct	shared
1	5418680	adjective	1.0000000	NA
2	5418680	noun	1.0000000	NA
3	46356924	adjective	0.7647059	NA
4	46356924	noun	-0.3888889	-0.05555556
5	52271504	adjective	1.0000000	NA
6	52271504	noun	1.0000000	NA
7	59881077	adjective	0.6470588	NA
8	59881077	noun	1.0000000	NA
9	161705773	adjective	1.0000000	NA
10	161705773	noun	1.0000000	NA

```r
## counts by type of association

association_type_occurrence = wide_counts %>%
  select(ID, cue_type, type_of_association, prop) %>%
  pivot_wider(names_from = type_of_association, values_from = prop) %>%
  mutate(shared = ifelse(is.na(shared), 0, shared),
         direct = ifelse(is.na(direct), 0, direct))

mean(association_type_occurrence$direct)
mean(association_type_occurrence$shared)
```

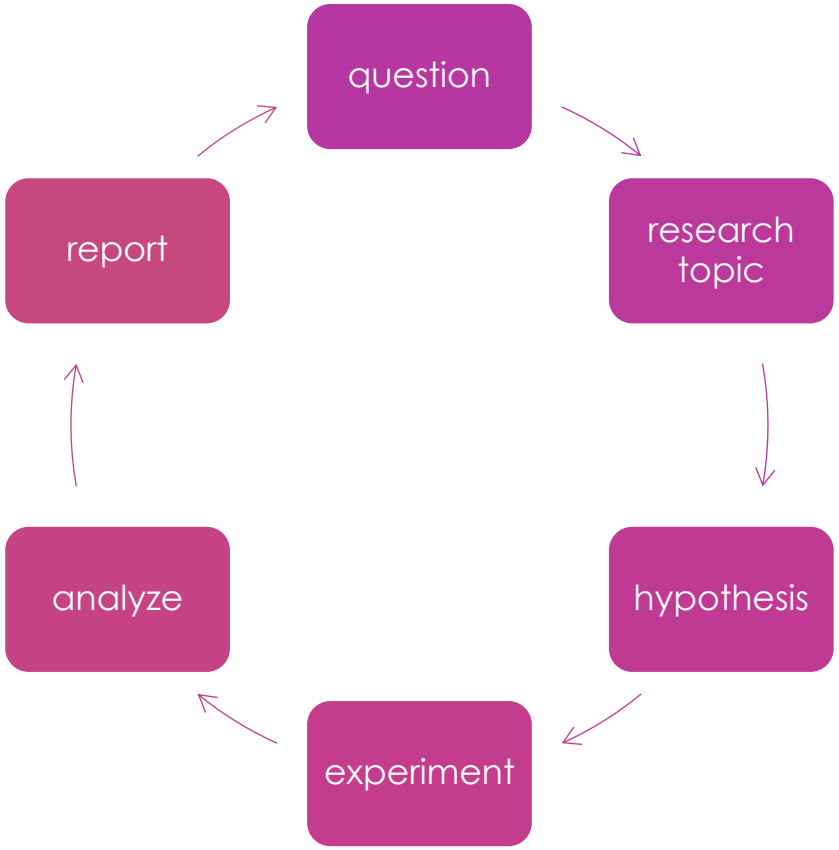
```
> mean(association_type_occurrence$direct)
[1] 0.8387088
> mean(association_type_occurrence$shared)
[1] -0.009946785
```

today's agenda

- association data analysis
- statistical inferences

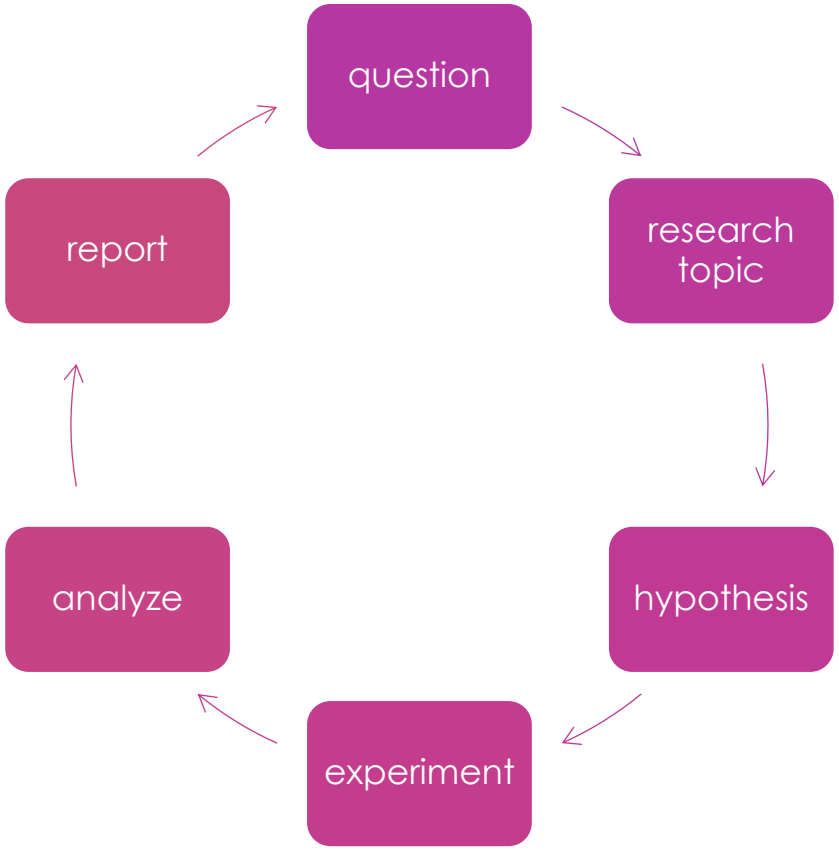
making inferences from data

- the research cycle employs *the scientific method* to answer questions



logic of null hypothesis statistical testing

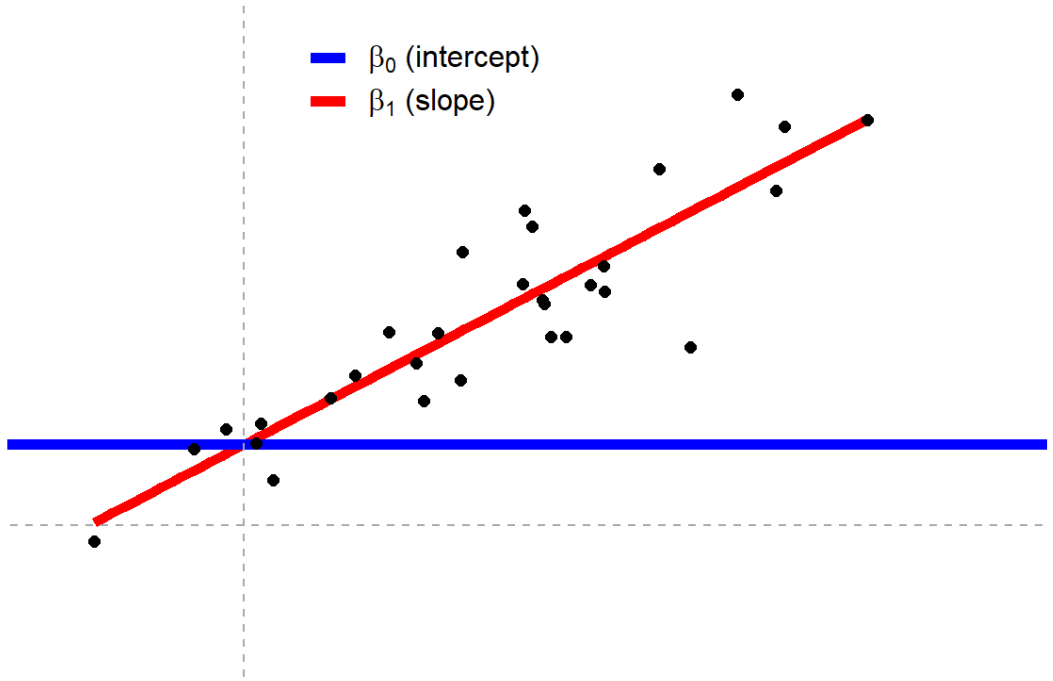
- formulate a hypothesis
- specify null and alternative hypotheses
- collect data relevant to the hypothesis
- fit a model to the data that represents the alternative hypothesis and compute a test statistic
- compute the probability of the observed value of that statistic assuming that the null hypothesis is true
- assess the "statistical significance" of the result



https://statsthinking21.github.io/statsthinking21-core-site/hypothesis-testing.html

linear regression

- a linear regression (or a linear model) is a model that fits a line to a set of data points
  - $Y = aX + b$
  - Y: dependent variable
  - X: independent variable
  - a? b?
- a: slope, b: intercept
- sometimes, we reorder this equation:
  - $y = \beta_0 + \beta_1 x$
  - β_0: intercept (where the line cuts the y-axis)
  - β_1: slope (the change in y due to x)
- in this framework, the null hypothesis (H_0) is that $\beta_1 = 0$, i.e., there is no change in y due to x
  - H_0: $\beta_1 = 0$



https://lindeloev.github.io/tests-as-linear/

exploring the data

- new heading # linear models
- load the dataset women
- make a scatterplot of the data
  - x = weight
  - y = height
- fit a line to the data via geom_smooth()

```
# regression

```{r}

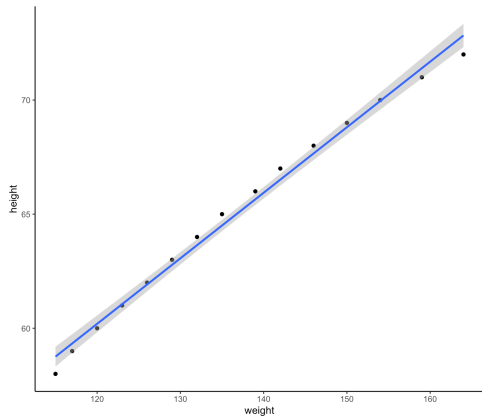
data(women)
```

first-R-notebook.Rmd | women

Filter

	height	weight
1	58	115
2	59	117
3	60	120
4	61	123
5	62	126
6	63	129
7	64	132
8	65	135
9	66	139
10	67	142
11	68	146
12	69	150
13	70	154
14	71	159
15	72	164

```
women %>%
  ggplot(aes(x= weight, y = height))+
  geom_point() +
  geom_smooth(method = "lm")+
  theme_classic()
```


```

linear regression in R

- predict height by weight
- print the summary of the model
- what is the equation of the line?

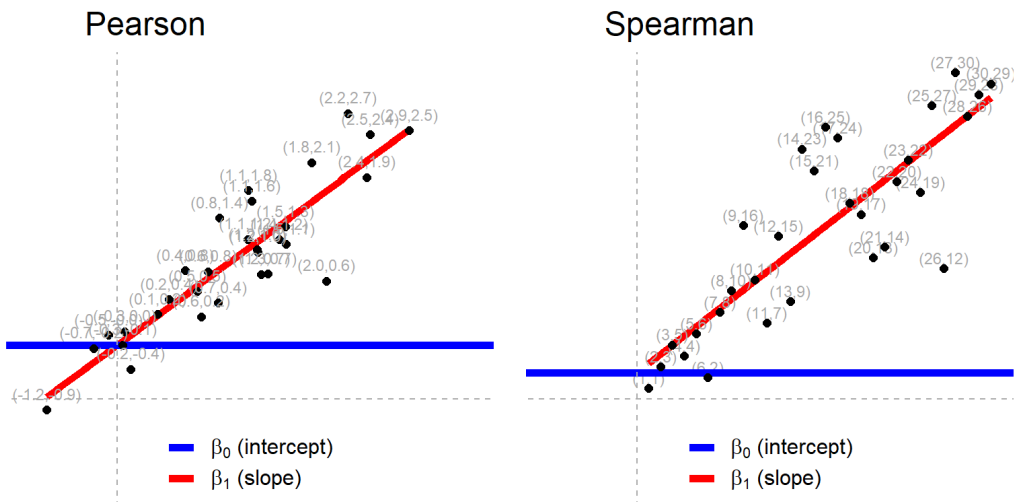
```r
women_model = lm(data = women, height ~ weight)
```

```r
summary(women_model)
```

```
Call:
lm(formula = height ~ weight, data = women)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.83233 -0.26249  0.08314  0.34353  0.49790 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 25.723456   1.043746   24.64 2.68e-12 ***
weight       0.287249   0.007588   37.85 1.09e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.44 on 13 degrees of freedom
Multiple R-squared:  0.991,	Adjusted R-squared:  0.9903 
F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-14
```

linear regression and correlation

- correlations also describe the relationship between Y and X, so what's the difference?
- mathematically, correlations are equivalent to a linear model where a line is being fit to a set of data points
- two common correlation
  - Pearson's *r*: *r* = slope if x and y have the same standard deviation
  - Spearman's *rho* = same linear model but with ranks of x and Y
    - rank(y) = $\beta_0 + \beta_1$ rank(x)



https://lindeloev.github.io/tests-as-linear/

linear regression and correlation

- compute the standard deviation of the height and weight columns
- create two new columns that contain the z-scored height and weight
- compute the standard deviation of the z-scored height and weight columns

```
sd(women$height)
sd(women$weight)
```

```
women = women %>%
  mutate(z_height = scale(height),
         z_weight = scale(weight))
```

```
sd(women$height)
sd(women$weight)
```

linear regression and correlation

- predict the z-scored height with the z-scored weight using linear regression
- now compute the correlation between the two columns using summarize() and cor()

```
women_model_2 = lm(data = women, z_height ~ z_weight)
summary(women_model_2)
```

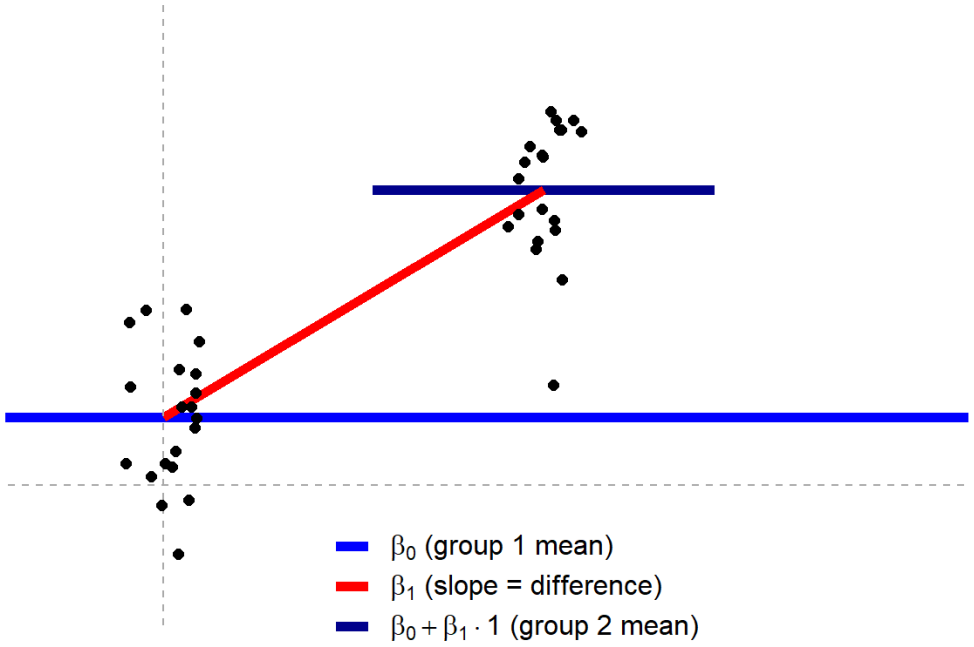
```
Call:
lm(formula = z_height ~ z_weight, data = women)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.18611 -0.05869  0.01859  0.07682  0.11133 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -8.268e-16  2.541e-02    0.00        1    
z_weight     9.955e-01  2.630e-02   37.85 1.09e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0984 on 13 degrees of freedom
Multiple R-squared:  0.991,	Adjusted R-squared:  0.9903 
F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-14
```

```
women %>%
  summarise(r = cor(z_height, z_weight))
```

```
  r
1 0.9954948
```

linear regression and t-tests

- unpaired/independent samples t-test
  - $y = \beta_0 + \beta_1 x$
  - $x = 0$ or 1 (which group)
  - H_0: $\beta_1 = 0$
  - comparing paired differences and testing whether the difference is significantly different from 0
  - note that "x" here contains information about group membership for each y



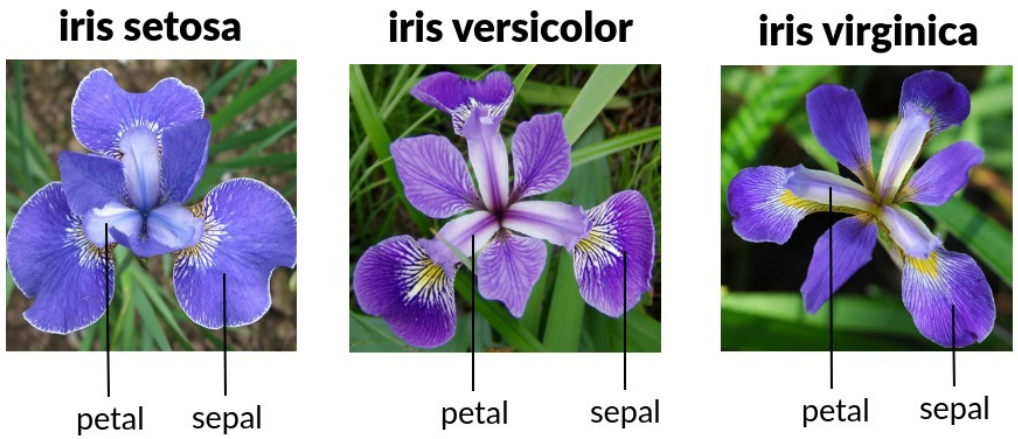
https://lindeloev.github.io/tests-as-linear/

revisiting iris

- recall that iris contains flower petal and sepal information for three species

```
data("iris")
View(iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa



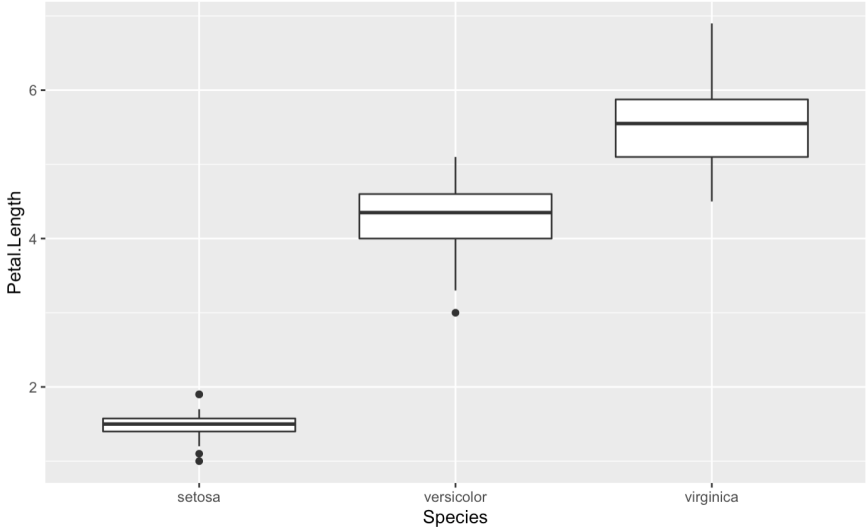
subset of iris

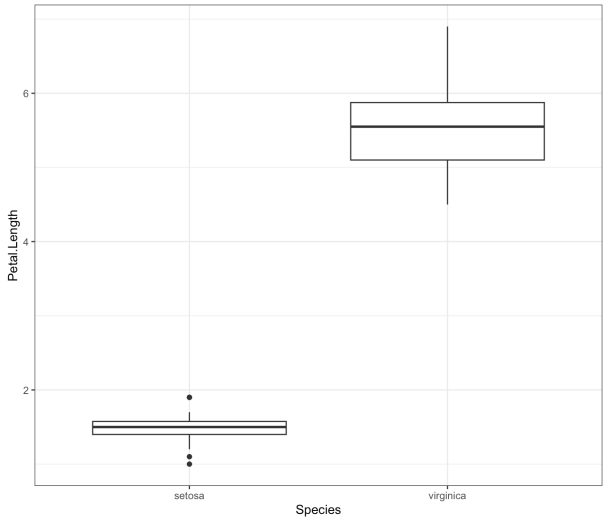
- create a subset of iris that only contains setosa and virginica
- plot the petal lengths by species in a boxplot

```
## t -test
```

```{r}
iris_subset = iris %>%
  filter(Species %in% c("setosa", "virginica"))
```

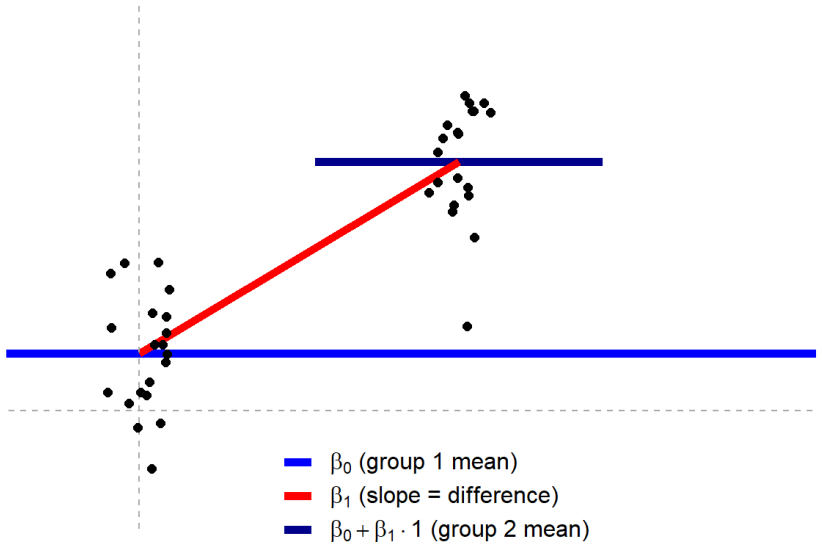
```
iris_subset %>%
  ggplot(aes(x = Species, y = Petal.Length))+
  geom_col()
```





comparing

- create linear model
- conduct t-test




```
iris_subset_lm = lm(data = iris_subset, Petal.Length ~ Species)
summary(iris_subset_lm)
```

```
Call:
lm(formula = Petal.Length ~ Species, data = iris_subset)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.0520 -0.1620  0.0380  0.1405  1.3480 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1.46200    0.05786   25.27   <2e-16 ***
Speciesvirginica  4.09000    0.08182   49.99   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4091 on 98 degrees of freedom
Multiple R-squared:  0.9623,	Adjusted R-squared:  0.9619 
F-statistic:  2499 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
t.test(Petal.Length ~ Species, data = iris_subset)
```

```
	Welch Two Sample t-test

data:  Petal.Length by Species
t = -49.986, df = 58.609, p-value < 2.2e-16
alternative hypothesis: true difference in means between group setosa and group virginica is not equal to 0
95 percent confidence interval:
 -4.253749 -3.926251
sample estimates:
   mean in group setosa mean in group virginica 
                  1.462                  5.552 
```

testing more than two groups

- a t-test is a special case of linear models
- it is *also* a special case of only comparing two groups
- example of comparing more than two groups?

next class

- **before** class
  - *submit:* class survey (October)
  - *try:* W10 quiz
  - *complete:* piloting + feedback (Friday)
  - *apply:* formative assignment #2
  - *apply:* pre-registration draft (milestone #6)
  - prep: complete all primers
- **during** class
  - Nov 7: guest lecture: Dr. Kyle Featherston!
  - Nov 9: ANOVAs and linear models