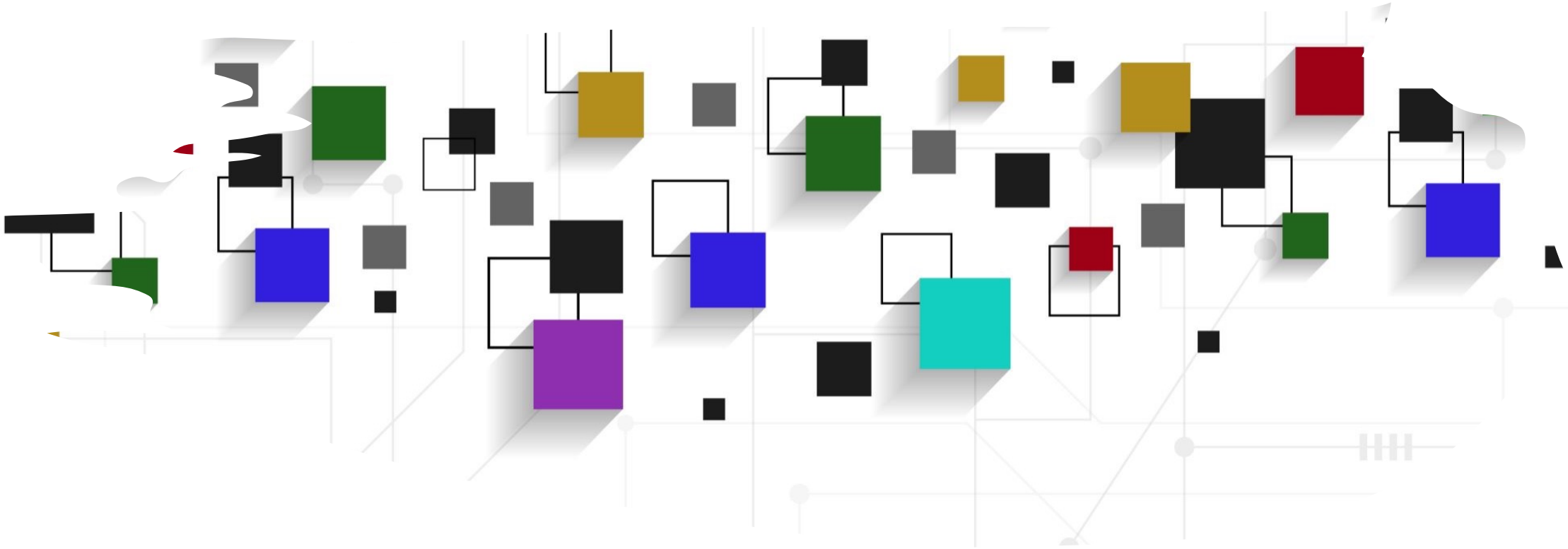
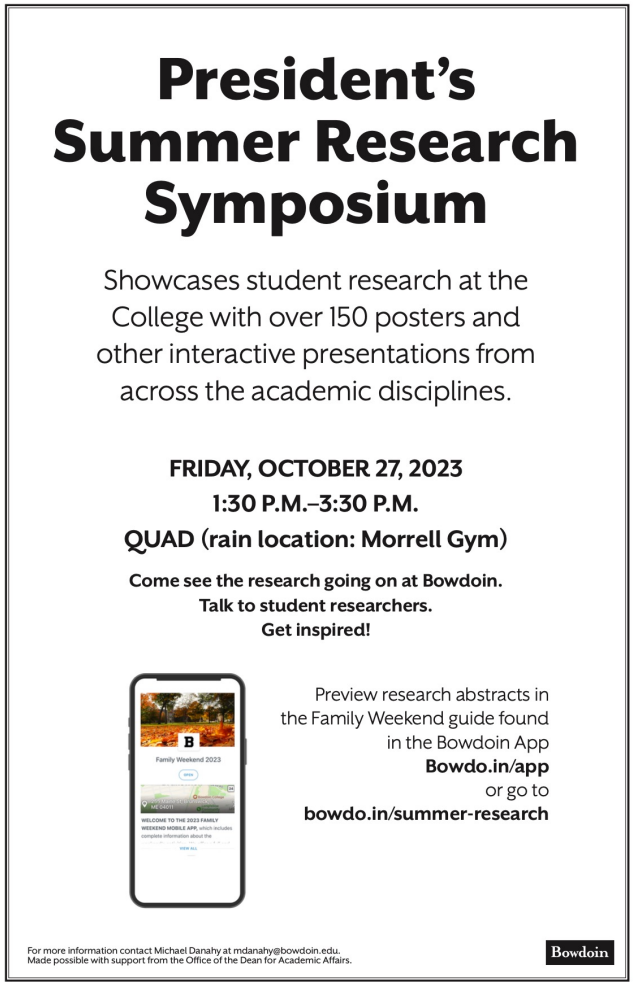


CogLab: Manipulate Data

WEEK 9

Oct 27: research in psychology



President's Summer Research Symposium

Showcases student research at the College with over 150 posters and other interactive presentations from across the academic disciplines.

FRIDAY, OCTOBER 27, 2023
1:30 P.M.–3:30 P.M.
QUAD (rain location: Morrell Gym)

Come see the research going on at Bowdoin.
Talk to student researchers.
Get inspired!

Preview research abstracts in the Family Weekend guide found in the Bowdoin App
Bowdo.in/app
or go to
bowdo.in/summer-research

For more information contact Michael Danahy at mdanahy@bowdoin.edu.
Made possible with support from the Office of the Dean for Academic Affairs.

Bowdoin

logistics: project

- next milestone #6: pre-registration (Nov 5)
- **before** pre-registration:
  - piloting your experiment (Jon + other groups, N = 7), pilot feedback sheet
  - send cognition.run link by Friday
  - finalizing analysis plan + sample size
- each group should come this week to office hours
  - Thursdays, 9-10 am
  - Thursdays, 4-5.30 pm
  - Fridays, 10-11 am
  - extra: Friday, 12 pm – 1.30 pm
  - extra: Friday, 3.20 pm – 5.30 pm

logistics: demographics

```
var language_question1 = {
    type: jsPsychHtmlButtonResponse,
    stimulus: 'Is English your first language?',
    choices: ["Yes", "No"],
    name: 'Language',
    data: {
        typeoftrial: "demo_language",
    },
}
var no_lang_question1 = {
    type: jsPsychSurveyText,
    questions: [{prompt: "What is your first language?", required : true}]
}

var no_lang_question2 = {
    type: jsPsychSurveyText,
    questions: [{prompt: "At what age did you learn English?", required : true}]
}

var no_lang = {
    timeline: [no_lang_question1, no_lang_question2],
    conditional_function: function () {
        var last_trial_data = jsPsych.data.get().filter({typeoftrial: 'demo_language'}).last(1).values()[0];
        console.log("last_trial_data=", last_trial_data);

        if (last_trial_data.response == 1) {
            return true
        }
        else {
           return false
       }
    }
}

var language = {
    timeline: [language_question1,no_lang]
}
```

logistics: demographics

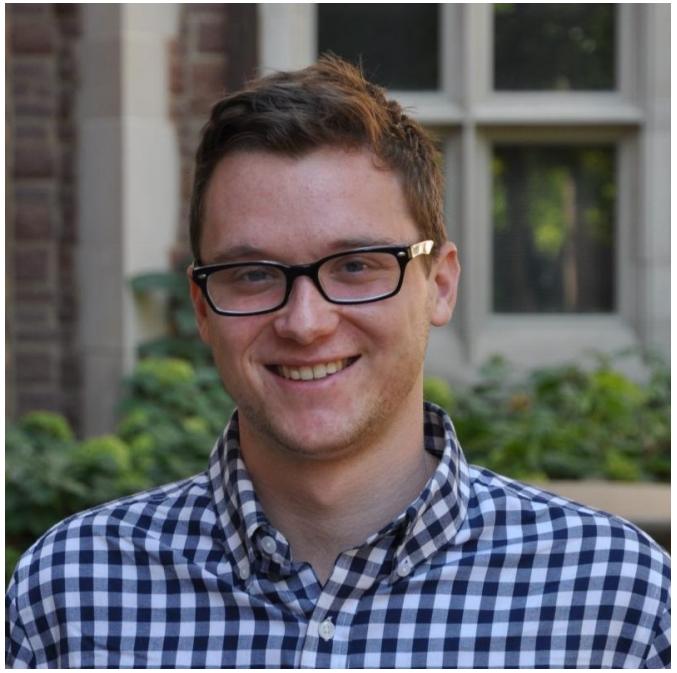
- make sure the data is getting recorded correctly
- especially from survey-text questions
- data.response = data.response.Q0

logistics: formative assignment #2

- descriptive statistics and plotting in R
- due Oct 29 (first draft worth 2%, second worth 8%)

Nov 7: guest speaker

- Dr. Kyle Featherston
- Ph.D., Psychological and Brain Sciences
- Research Program Director, Columbia University School of Nursing
- available for one-on-one career meetings:
  - 9 am – 10 am
  - 1 pm – 3 pm
- sign up here



8	Sunday, October 22, 2023	**Project Milestone #5 (Full Experiment) Due**
9	Tuesday, October 24, 2023	W9: Manipulate Data
9	Thursday, October 26, 2023	W9 continued…
9	Sunday, October 29, 2023	**Formative Assignment (R Descriptive) Due**
10	Tuesday, October 31, 2023	W10: Making Inferences
10	Thursday, November 2, 2023	W10 continued…
10	Sunday, November 5, 2023	**Project Milestone #6 (Pre-Registration) Due**
11	Tuesday, November 7, 2023	**Guest Session: Dr. Kyle Featherston**
11	Thursday, November 9, 2023	Weeks 11-13: Data Collection
11	Sunday, November 12, 2023	**Formative Assignment (R Inferential) Due**
12	Tuesday, November 14, 2023	Data Collection continued…
12	Thursday, November 16, 2023	**Psychonomics Conference: NO CLASS**
12	Sunday, November 19, 2023	**Project Milestone #7 (Analyses) Due**

recap: Oct 17/19, 2023

- what we covered:
  - R101, data analysis plan
  - visualizing data
- your to-do's were:
  - *prep:* Work with Data primer
  - *try:* HW, fix the data!
  - *apply:* Week 8 Quiz
  - *apply:* formative milestone # 1 resubmission
  - *apply:* project milestone 5 (full experiment)

HW: fixing accuracy

AB	AC	AD	AE	AF	AG	AH	AI	AJ
response	revised_response	sentence	novel1	novel2	novel3	correct	revised_correct	cue
sh it were easier to get a foobly mipp.		Sometimes I wis	foobly	mipp	NOT_FOUND			
cided I'd go looking for a foobly apple.		In the end, I dec	foobly	NOT_FOUND	NOT_FOUND			
et there told me they saw a foobly app		The people I me	foobly	NOT_FOUND	NOT_FOUND			
I would like a dodish horse better.		I am not sure if I	dodish	NOT_FOUND	NOT_FOUND			

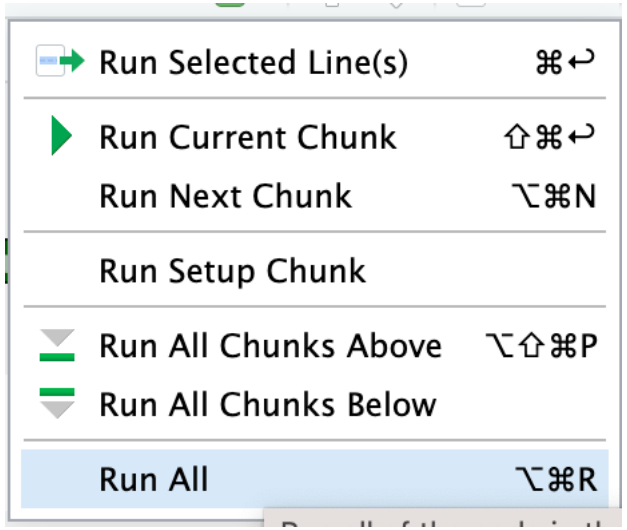
- go to revised_class_data on drive
- group task: fix the data!
  - Semantic Snakes: fix the attention check **responses + accuracy**
  - Berries: fix association responses, IDs 275998227- 276772242
  - Nellaphen: fix association responses, IDs 823472278 – 988749039
- complete before Tuesday (Oct 24)

today's agenda

- tidyverse verbs
  - select()
  - filter()
  - mutate()
  - summarize()
  - group_by()

open your RStudio project

- open the project and your .Rmd file
- re-download the class_data.csv file
- run all chunks

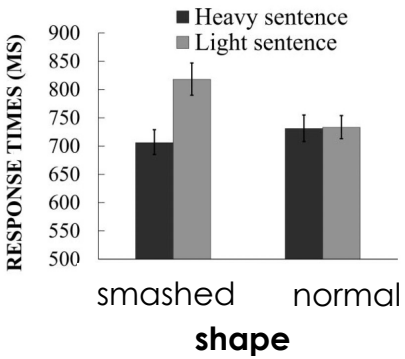


an experiment

- I will show you a sentence
- then I will show you an image
- raise your dominant hand if the object shown was mentioned in the sentence
- raise your non dominant hand otherwise

you drop a bowling ball on a tomato

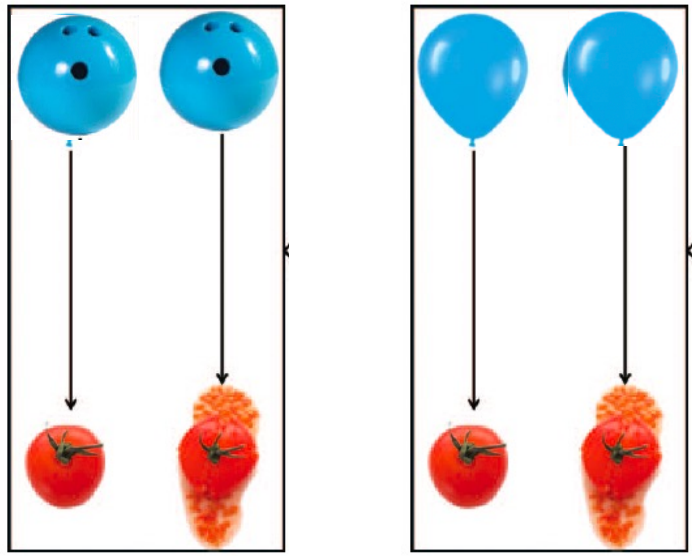
object state changes dataset



- task: object verification from sentences presented to participants
- research questions: do the events mentioned in the sentences influence response time?
- RT (bowling ball + squashed tomato) vs. RT (bowling ball + intact tomato)
- RT (balloon + squashed tomato) vs. RT (balloon + intact tomato)

Dropping Bowling Balls on Tomatoes: Representations of Object State-Changes During Sentence Processing

Oleksandr V. Horchak and Margarida Vaz Garrido
Iscte-Instituto Universitário de Lisboa



review: importing new data

- create a new a # tidyverse verbs heading and code chunk
- download objects.csv
- import this data into your notebook and name it objectdata
- how many rows and columns?

tidyverse verbs

```{r}
objectdata = read_csv("objects.csv")
```

first-R-notebook.Rmd × objectdata ×

Filter

	success	timeout	failed_images	failed_audio	failed_video	typeoftrial	subject	trial_type
1	TRUE	FALSE	[]	[]	[]	preload	77491	preload
2	NA	NA	NA	NA	NA	ID	77491	survey-text
3	NA	NA	NA	NA	NA	instructions	77491	html-button-response
4	NA	NA	NA	NA	NA	instructions	77491	html-button-response
5	NA	NA	NA	NA	NA	fixation	77491	html-keyboard-response
6	NA	NA	NA	NA	NA	sentence	77491	html-keyboard-response
7	NA	NA	NA	NA	NA	fixation	77491	html-keyboard-response
8	NA	NA	NA	NA	NA	picture	77491	image-keyboard-response

tidyverse verbs

- often, your experiment data is not in analysis-ready format
- you may need to delete some rows, select some columns, arrange the data, etc.
- tidyverse verbs allow you to manipulate the dataframe and make it analysis and plotting-friendly

first-R-notebook.Rmd | objectdata

Filter

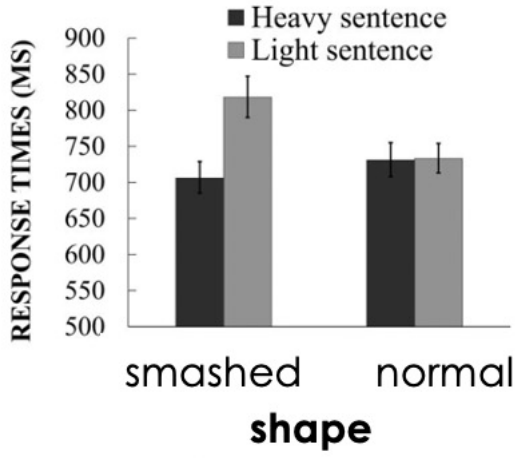
	success	timeout	failed_images	failed_audio	failed_video	typeoftrial	subject	trial_type
1	TRUE	FALSE	[]	[]	[]	preload	77491	preload
2	NA	NA	NA	NA	NA	ID	77491	survey-text
3	NA	NA	NA	NA	NA	instructions	77491	html-button-response
4	NA	NA	NA	NA	NA	instructions	77491	html-button-response
5	NA	NA	NA	NA	NA	fixation	77491	html-keyboard-response
6	NA	NA	NA	NA	NA	sentence	77491	html-keyboard-response
7	NA	NA	NA	NA	NA	fixation	77491	html-keyboard-response
8	NA	NA	NA	NA	NA	picture	77491	image-keyboard-response

tidyverse piping

- piping is a way to define a sequence of operations in R
- this is accomplished using %>%
- the idea is that you use the same data but perform multiple operations on it using the pipe
- we will use piping to combine different operations together

tidyverse: select()

- select() allows you to retain only specific columns from your dataframe
- useful when your data contains too many unnecessary columns that are not relevant for analysis
- what columns might be important in this dataset?
- print the column names and let's make a list!



first-R-notebook.Rmd × objectdata ×

Filter

	success	timeout	failed_images	failed_audio	failed_video	typeoftrial	subject	trial_type
1	TRUE	FALSE	[]	[]	[]	preload	77491	preload
2	NA	NA	NA	NA	NA	ID	77491	survey-text
3	NA	NA	NA	NA	NA	instructions	77491	html-button-response
4	NA	NA	NA	NA	NA	instructions	77491	html-button-response
5	NA	NA	NA	NA	NA	fixation	77491	html-keyboard-response
6	NA	NA	NA	NA	NA	sentence	77491	html-keyboard-response
7	NA	NA	NA	NA	NA	fixation	77491	html-keyboard-response
8	NA	NA	NA	NA	NA	picture	77491	image-keyboard-response

tidyverse: select()

- logic of piping:
  - start with the dataset
  - add a pipe
  - specify an action
- select RT, weight, and shape from objectdata
- run the chunk
- what do you see?
- ALL trials are being included because select only picks the columns, not the rows

```
objectdata %>%
  select(rt, weight, shape)
```

```
# A tibble: 34,057 × 3
   rt    weight   shape
   <chr> <chr>    <chr>
 1 NA    NA       NA
 2 11783 NA       NA
 3 51986 NA       NA
 4 21791 NA       NA
 5 null  NA       NA
 6 4589  practice n
 7 null  NA       NA
 8 6443  practice n
 9 null  NA       NA
10 null  NA       NA
# ℹ 34,047 more rows
# ℹ Use `print(n = ...)` to see more rows
```

tidyverse: filter()

- filter() allows you to retain only specific rows from your dataframe
- if we need only the picture trials, we can use filter to do this *before* we select our columns
- notice how we've used the pipe to continue our code
- run this chunk again!
- what do you notice now?

```
objectdata %>%
  filter(typeoftrial == "picture") %>%
  select(rt, weight, shape)
```

	rt	weight	shape
	<chr>	<chr>	<chr>
1	6443	practice	n
2	6516	practice	s
3	7821	practice	s
4	2096	practice	s
5	2849	filler	NA
6	3256	Heavy	Smashed
7	1698	filler	NA
8	1615	Light	Normal
9	1619	Heavy	Smashed
10	1304	Light	Normal

tidyverse: filter()

- the data is a lot better now but still contains filler and practice trials
- we could add an additional conditions in our filter statement that restrict the values of weight and shape
- the & operator combines different constraints we want to apply to the data

```
objectdata %>%
  filter(typeoftrial == "picture" & weight %in% c("Heavy", "Light") &
           shape %in% c("Normal", "Smashed")) %>%
  select(rt, weight, shape)
```

```
# A tibble: 2,376 × 3
   rt    weight shape
   <chr> <chr>  <chr>
 1 3256  Heavy  Smashed
 2 1615  Light  Normal
 3 1619  Heavy  Smashed
 4 1304  Light  Normal
 5 1602  Light  Normal
 6 1713  Heavy  Smashed
 7 1568  Light  Smashed
 8 4007  Light  Smashed
 9 3013  Heavy  Normal
10 1321  Light  Normal
```

tidyverse: %in%

- %in% is a useful tidyverse operator that checks whether an element belongs to a vector
- in your console: check if 3 is inside a vector containing 4, 6, 7, 9, 3
- each part of filter() is a condition being evaluated as TRUE or FALSE

```
> 3 %in% c(4, 6, 7, 9, 3)
[1] TRUE
```

```
objectdata %>%
  filter(typeoftrial == "picture" & weight %in% c("Heavy", "Light") &
           shape %in% c("Normal", "Smashed")) %>%
  select(rt, weight, shape)
```

exercise: more constraints

- we want to evaluate only correct trials, use filter() to do this
- we want to retain the subject/participant identifier in the resulting dataframe: use select() to do this

```r
objectdata %>%
  filter(typeoftrial == "picture" & weight %in% c("Heavy", "Light") &
           shape %in% c("Normal", "Smashed") &
           correct == TRUE) %>%
  select(subject, rt, weight, shape, correct)
```

```
# A tibble: 2,263 × 4
   subject rt    weight shape
     <dbl> <chr> <chr>  <chr>
 1   77491 3256  Heavy  Smashed
 2   77491 1615  Light  Normal
 3   77491 1619  Heavy  Smashed
 4   77491 1304  Light  Normal
 5   77491 1602  Light  Normal
 6   77491 1713  Heavy  Smashed
 7   77491 1568  Light  Smashed
 8   77491 4007  Light  Smashed
 9   77491 3013  Heavy  Normal
10   77491 1321  Light  Normal
```

storing filtered data

- we not only want to subset the data but also store it so that we can do more analyses on the data
- name the filtered data as condition_data
- this should create condition_data in the environment
- click and examine that data

```
condition_data = objectdata %>%
  filter(typeoftrial == "picture" & weight %in% c("Heavy", "Light") &
           shape %in% c("Normal", "Smashed") &
           correct == TRUE) %>%
  select(subject, rt, weight, shape, correct)
```

Filter

	subject	rt	weight	shape	correct
144	69266	864	Light	Smashed	TRUE
145	69266	1285	Heavy	Normal	TRUE
146	69266	942	Heavy	Normal	TRUE
147	69266	1669	Light	Normal	TRUE
148	69266	1745	Light	Smashed	TRUE
149	69266	1106	Light	Smashed	TRUE
150	69266	1077	Heavy	Normal	TRUE
151	69266	932	Heavy	Normal	TRUE
152	69266	908	Heavy	Smashed	TRUE

tidyverse: summarize()

- summarize() calculates descriptive statistics for your data
- we can compute the mean reaction time across all trials and all participants for condition_data
- NAs are produced when the mean cannot be computed

```
condition_data %>%
  summarise(mean_rt = mean(rt))
```

```
> condition_data %>%
+   summarise(mean_rt = mean(rt))
# A tibble: 1 × 1
  mean_rt
    <dbl>
1      NA
Warning message:
There was 1 warning in `summarise()`.
i In argument: `mean_rt = mean(rt)`.
Caused by warning in `mean.default()`:
! argument is not numeric or logical: returning NA
```

tidyverse: mutate()

- mutate() allows you to create new columns in your dataframe or change/replace existing columns
- we can use mutate() to change the data type of important columns <u>when we read in the object data</u>
- re-run your chunk

```
objectdata = read_csv("objects.csv") %>%
  mutate(rt = as.numeric(rt),
         weight = as.factor(weight),
         shape = as.factor(shape))
```

```
  $ rt                : num [1:34057] NA 11783 51986 21791 NA ...
  $ response          : chr [1:34057] NA "{\"ID\":\"60ad7bc194a8625071b
  $ Experiment        : logi [1:34057] NA NA NA NA NA NA ...
  $ stimulus          : chr [1:34057] NA NA "\n  <p style=\"font-size:2
  $ List              : chr [1:34057] NA NA NA NA ...
  $ weight            : Factor w/ 4 levels "filler","Heavy",..: NA NA N
  $ shape             : Factor w/ 4 levels "n","Normal","s",..: NA NA N
```

```
> condition_data %>%
+   summarise(mean_rt = mean(rt))
# A tibble: 1 × 1
  mean_rt
    <dbl>
1   1113.
```

tidyverse: more summarize()

- compute the standard deviation of reaction time
- store it all in a dataframe called object_agg

```
object_agg = condition_data %>%
  summarise(mean_rt = mean(rt),
            sd_rt = sd(rt))
```

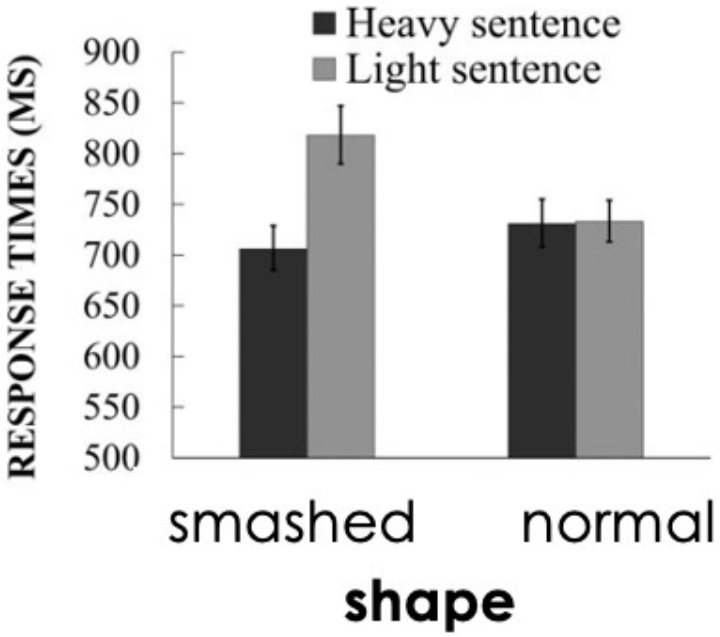
first-R-notebook.Rmd × | object_agg ×

Filter

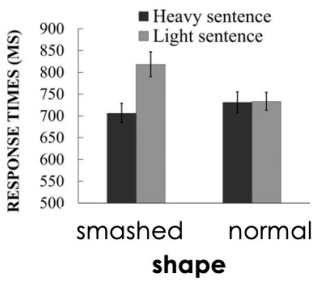
	mean_rt	sd_rt
1	1112.755	826.0641

tidyverse: group_by()

- group_by() allows you to group the data based on specific values within a column
- if we want to obtain reaction times for our conditions, which columns should we use to group the data?



tidyverse: group_by()



- modify object_agg
- group by weight and shape
- compute the mean and sd of reaction time
- are we in business??

```
object_agg = condition_data %>%
  group_by(weight, shape) %>%
  summarise(mean_rt = mean(rt),
            sd_rt = sd(rt))
```

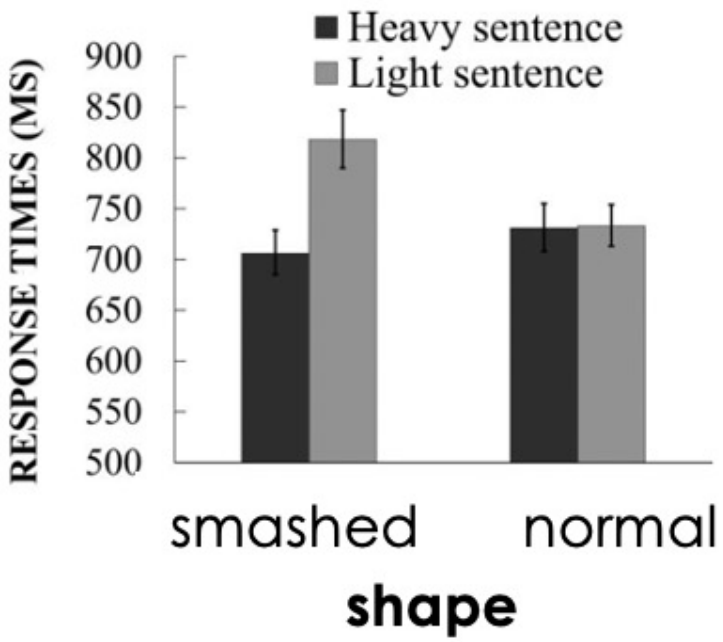
first-R-notebook.Rmd | object_agg | condit

Filter

	weight	shape	mean_rt	sd_rt
1	Heavy	Normal	1134.607	976.1069
2	Heavy	Smashed	1150.814	1007.0356
3	Light	Normal	1048.484	581.6215
4	Light	Smashed	1117.298	644.8903

we're in business!

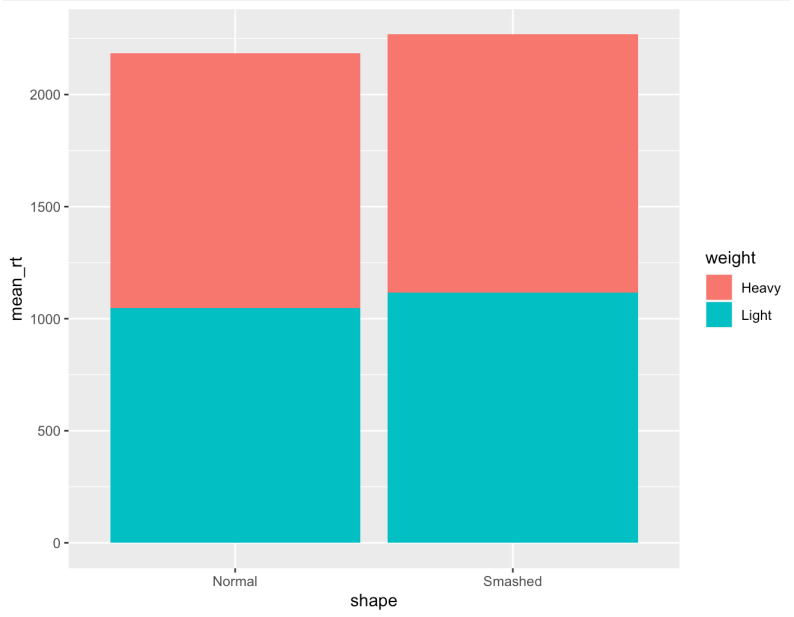
- we can now plot the means using our favorite plotting function
- recall the grammar of graphics…what 3 things do we need?
- data?
- geom?
- mapping/aes?



plotting the means

- use ggplot() to plot the data
- notice the + sign, not %>% for plotting
- notice the fill is inside the aes() because it is a column from the data
- close…?

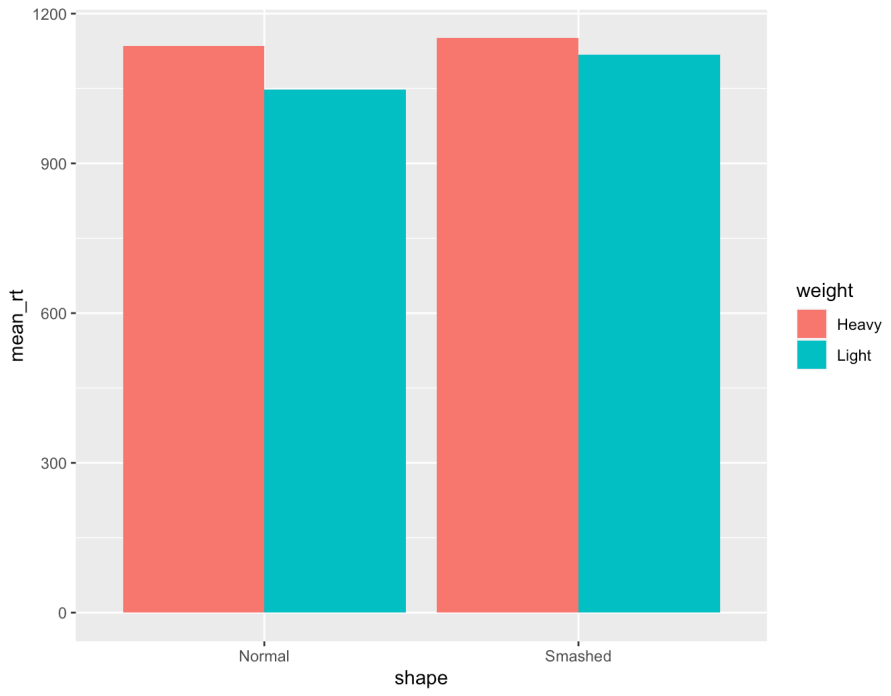
```
ggplot(data = object_agg) +
  geom_col(mapping = aes(x = shape, y = mean_rt, fill = weight))
```



stacked vs. unstacked plots

- stacked bar charts display the grouped data on top of each other
- unstacked bar charts separate the bars
- use position = "dodge" inside geom_col(), after the mapping

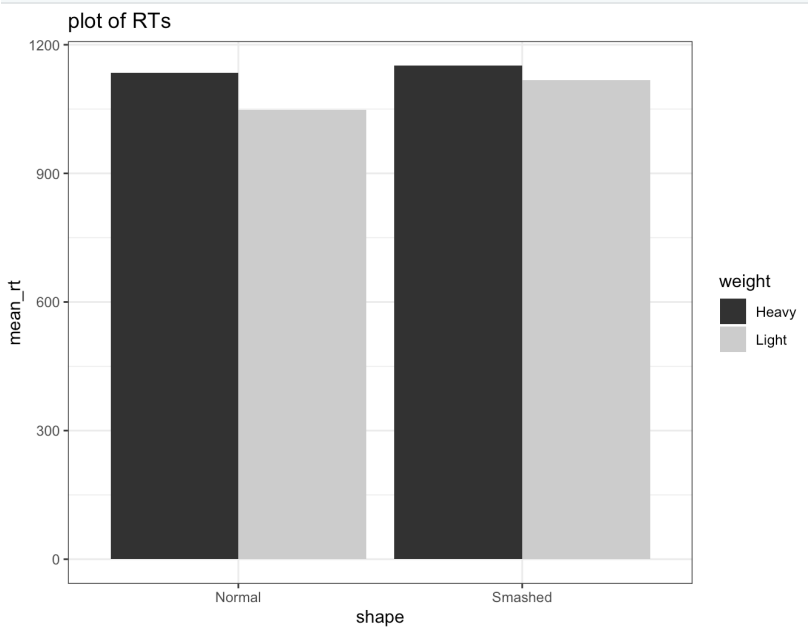
```
ggplot(data = object_agg) +
  geom_col(mapping = aes(x = shape, y = mean_rt, fill = weight),
           position = "dodge")
```



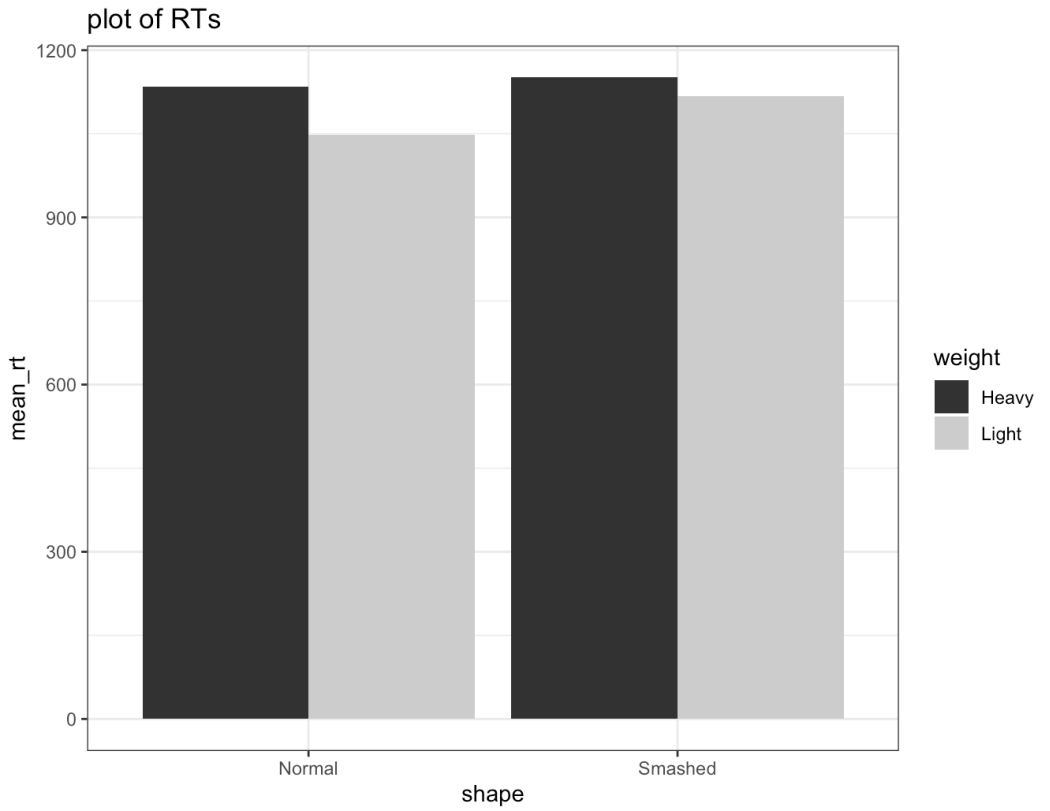
prettify your plot!

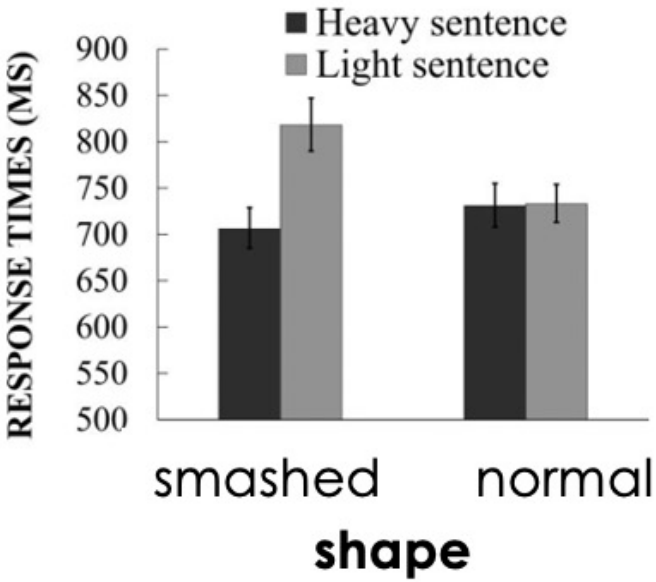
- add a theme
- add a title
- change color palette
- if aesthetics focus on filling, then use scale_fill_ otherwise use scale_color_

```
ggplot(data = object_agg) +
  geom_col(mapping = aes(x = shape, y = mean_rt, fill = weight),
           position = "dodge") +
  theme_bw()+
  labs(title = "plot of RTs")+
  scale_fill_grey()
```



interpreting the plot





HW: exercises

- what if I wanted RTs for each condition for each participant?
- before I analyzed the RTs, what if I wanted to first filter out participants who failed an attention check?

next class

- **before** class
  - *schedule:* group meeting
  - *complete*: data cleaning
  - *prep:* complete the Tidy your Data primer
- **during** class
  - more data wrangling (for your experiments)