

CogLab: Manipulate More

WEEK 8

logistics: formative assignments

- formative assignment #1: resubmission due Sunday
- formative assignment #2: descriptive statistics and plotting in R
 - due Nov 3

7	Monday, Oct 21, 2024	Project Milestone #4 (Full Experiment) Due
8	Tuesday, October 22, 2024	<u>W8: Manipulate Data</u>
8	Thursday, October 24, 2024	W8 continued...
8	Sunday, October 27, 2024	Formative Assignment (jsPsych) Resubmission Due
9	Tuesday, October 29, 2024	<u>W9: Making Inferences</u>
9	Thursday, October 31, 2024	W9 continued...
9	Sunday, November 3, 2024	Formative Assignment (R Descriptive) Due
10	Tuesday, November 5, 2024	<u>Weeks 10-12: Data Collection</u>
10	Thursday, November 7, 2024	<u>Weeks 10-12: Data Collection</u>
10	Sunday, November 10, 2024	Formative Assignment (R Inferential) Due
11	Tuesday, November 12, 2024	<u>Weeks 10-12: Data Collection</u>
11	Thursday, November 14, 2024	<u>Weeks 10-12: Data Collection</u>
11	Sunday, November 17, 2024	Formative Assignment (R Descriptive) Resubmission Due
11	Monday, November 18, 2024	Project Milestone #5 (Pre-Registration) Due

project checklist

- create a checklist! work on provided feedback
- schedule a meeting with me before Tuesday

pre-registration

- next [milestone #6](#): pre-registration (Nov 18: might move)
- **before** pre-registration:
 - providing accuracy feedback on priming trials
 - recording IP addresses
 - commenting the condition definition inside cognition.run
 - piloting your experiment (Uma + other group + 5 friends, N = 8), [pilot feedback sheet](#)
 - send cognition.run link by Nov 10
 - finalizing analysis plan + sample size

1. **Data Collection:** Have any data been collected for this study already?
2. **Main Question:** What is the main question being asked or hypothesis being tested in this study?
3. **Dependent Variable(s):** Describe the key dependent variable(s) specifying how they will be measured.
4. **Condition(s):** How many and which conditions will participants be assigned to? Please include an example trial of each type of condition you have in your experiment. Please also specify which independent variable will be within-participants or between-participants.
5. **Analyses:** Specify exactly which analyses you will conduct to examine the main question/hypothesis.
6. **Outliers & Exclusions:** Describe exactly how outliers will be defined and handled, and your precise rule(s) for excluding observations.
7. **Predicted Plot:** Please submit a predicted plot for your study based on what you expect the pattern to look like for your main hypothesis.
8. **Sample Size:** How many observations will be collected or what will determine sample size? No need to justify the decision, but be precise about exactly how the number will be determined.
9. **Exploratory details:** Anything else you would like to pre-register? (e.g., secondary analyses, variables collected for exploratory purposes, unusual analyses planned?)

recap

- what we covered:
 - tidyverse verbs
- your to-do's were:
 - *brainstorm*: group project code (accuracy feedback)
 - *complete*: formative assignment #1 resubmission
 - *prep*: Transform Tables recipes

today's agenda

- reviewing **tidyverse verbs** through your data
- learning a few more verbs/functions

wrangling your data

phase	measure	type	exclusion criteria
attention	accuracy	descriptive	< 0.75
priming	RT_{related} vs. $RT_{\text{unrelated}}$ for direct and shared pairs	inferential (mixed effects model / ANOVA)	$RT < 200$ ms and $RT > 1500$ ms correct responses

analyses checklist

- confirm/correct all datatypes
- figure out how to “filter” certain types of trials
- fix all typos in attention responses
- compute mean attention accuracy
- apply exclusions based on accuracy AND RTs
- create RT bar graph
- fit a statistical model
- report statistics

open your RStudio project

- open the project and your .Rmd file
- run all chunks
- create new heading # class data analysis
- change data types for a few columns

```
# class data analysis
```

```
```{r}
savic = read_csv("class_data.csv") %>%
 mutate(rt = as.numeric(rt),
 relatedness = as.factor(relatedness),
 type = as.factor(type))
```
```

basic descriptives

- how many **levels** does the relatedness variable have?
- what if you wanted the count the number of total trials **per participant**?

```
levels(savic$relatedness)
```

tidyverse: `count()`

- `count()` is another useful descriptive function like `summarize()` that tallies up counts of different things in your dataset while respecting groupings
- group by `ID` and count the trials
- how many `target` trials per ID?

```
savic %>% group_by(ID) %>% count()
```

```
...
```

```
savic %>% filter(typeoftrial == "target") %>%  
  group_by(ID) %>% count()
```

```
...
```

```
> savic %>% filter(typeoftrial == "target") %>%  
+   group_by(ID) %>% count()
```

```
# A tibble: 45 × 2  
# Groups:   ID [45]
```

| | ID | n |
|---|-----------|-------|
| | <dbl> | <int> |
| 1 | 5418680 | 104 |
| 2 | 46356924 | 104 |
| 3 | 52271504 | 104 |
| 4 | 59881077 | 104 |
| 5 | 161705773 | 104 |
| 6 | 223076836 | 104 |
| 7 | 275998227 | 104 |

```
# A tibble: 47 × 2  
# Groups:   ID [47]  
      ID     n  
  <dbl> <int>  
1  5418680  588  
2  46356924  588  
3  52271504  595  
4  59881077  604  
5  161705773  594  
6  223076836  592  
7  275998227  691  
8  276772242  591  
9  291529588  617  
10 317312681  602  
# ... with 37 more rows  
# Use `print(n = .`
```

tidyverse: `pull()`

- `pull()` is a convenient function that allows you to get the values inside one specific column as a vector
- extract the RTs from target trials

```
savic %>%  
  filter(typeoftrial == "target") %>%  
  pull(rt)
```

```
> savic %>%  
+ filter(typeoftrial == "target") %>%  
+ pull(rt)  
[1] NA NA 75 NA NA NA NA 214 NA 34  
[17] 242 NA 136 NA NA 277 NA 294 NA NA  
[33] 176 208 177 NA 254 NA NA NA 150 64  
[49] NA NA 227 223 302 NA 230 NA 88 253  
[65] 75 224 81 NA 98 NA 187 NA NA NA  
[81] NA 133 NA NA NA NA NA NA 246 214
```

tidyverse: `unique()`

- `unique()` lets you see how many unique values are inside a particular column or vector
- `how many subjects` did the experiment?
- the length of a vector can be obtained using `length()`

```
savic %>%  
  pull(ID) %>% unique()  
```\n
```

```
[1] 823806428 275998227 399427091 617108779 438881597 59881077 685459176 789634588 896801386 636502299 500683496
[12] 9793241458 7008638007 7840701730 6636886676 1804967059 1017677598 958013040 223076836 633750828 713138680 560469737
[23] 52271504 983568245 510710057 343557222 46356924 5418680 463680341 276772242 823472278 291529588 969332346
[34] 964433595 648738364 653349550 161705773 858059169 317312681 492031667 768759264 607043191 794294242 393616702
[45] 988749039 366197048 662530864
```

```
savic %>%
 pull(ID) %>% unique() %>% length()
```\n
```

analyses checklist

- confirm/correct all datatypes
- figure out how to “filter” certain types of trials
- fix all typos in attention responses
- compute mean attention accuracy
- apply exclusions based on accuracy AND RTs
- create RT bar graph
- fit a statistical model
- report statistics

attention

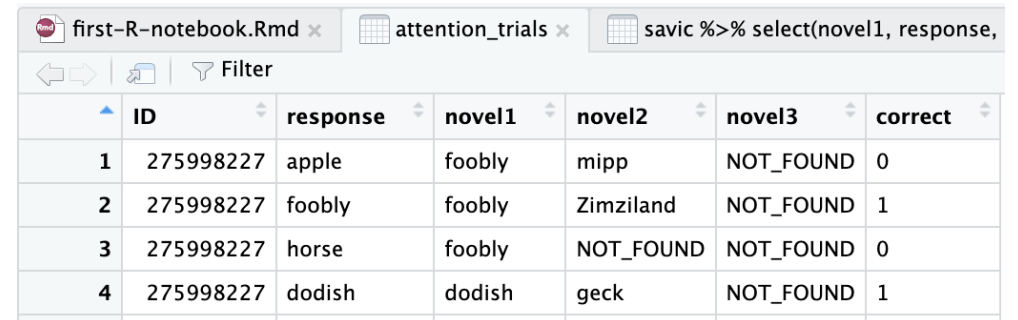
- create new heading # attention
- define a new dataframe `attention_trials` that only consists of the attention check rows
- which columns are most relevant?
- only keep the relevant columns
- view this data

```
# attention
```

```
```{r}
```

```
attention_trials = savic %>% filter(typeoftrial == "attention")
```

```
attention_trials = savic %>% filter(typeoftrial == "attention") %>%
 select(ID, response, novel1, novel2, novel3, correct)
```



first-R-notebook.Rmd × attention\_trials × savic %>% select(novel1, response,

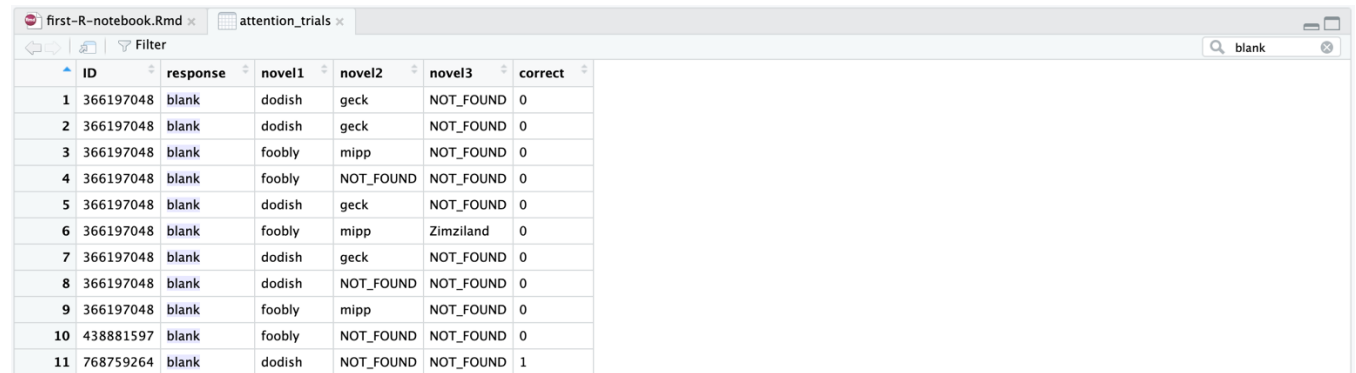
Filter

	ID	response	novel1	novel2	novel3	correct
1	275998227	apple	foobly	mipp	NOT_FOUND	0
2	275998227	foobly	foobly	Zimziland	NOT_FOUND	1
3	275998227	horse	foobly	NOT_FOUND	NOT_FOUND	0
4	275998227	dodish	dodish	geck	NOT_FOUND	1

# fixing accuracy

- first, we make sure every response has some text and novel words do not have blanks

```
attention_trials = savic %>% filter(typeoftrial == "attention")# %>%
 select(ID, response, novel1, novel2, novel3, correct) %>%
 rowwise() %>%
 mutate(response = ifelse(is.na(response), "blank", response)) %>%
 mutate(across(c(novel1, novel2, novel3), ~ replace_na(., "NOT_FOUND")))
```



ID	response	novel1	novel2	novel3	correct
1	blank	dodish	geck	NOT_FOUND	0
2	blank	dodish	geck	NOT_FOUND	0
3	blank	foobly	mipp	NOT_FOUND	0
4	blank	foobly	NOT_FOUND	NOT_FOUND	0
5	blank	dodish	geck	NOT_FOUND	0
6	blank	foobly	mipp	Zimziland	0
7	blank	dodish	geck	NOT_FOUND	0
8	blank	dodish	NOT_FOUND	NOT_FOUND	0
9	blank	foobly	mipp	NOT_FOUND	0
10	blank	foobly	NOT_FOUND	NOT_FOUND	0
11	blank	dodish	NOT_FOUND	NOT_FOUND	1



# fixing accuracy

- now create 3 new columns that measure the **distance** between the response and novel1, novel2, novel3
- what do you think these columns contain?

```
attention_trials = savic %>% filter(typeoftrial == "attention") %>%
 select(ID, response, novel1, novel2, novel3, correct) %>%
 rowwise() %>%
 mutate(response = ifelse(is.na(response), "blank", response)) %>%
 mutate(across(c(novel1, novel2, novel3), ~ replace_na(., "NOT_FOUND"))) %>%
 mutate(edit_novel1 = adist(novel1, response),
 edit_novel2 = adist(novel2, response),
 edit_novel3 = adist(novel3, response))
```

	ID	response	novel1	novel2	novel3	correct	edit_novel1[, 1]	edit_novel2[, 1]	edit_novel3[, 1]
1	275998227	apple	foobly	mipp	NOT_FOUND	0	5	4	9
2	275998227	foobly	foobly	Zimziland	NOT_FOUND	1	0	8	9
3	275998227	horse	foobly	NOT_FOUND	NOT_FOUND	0	5	9	9
4	275998227	dodish	dodish	geck	NOT_FOUND	1	0	6	9
5	275998227	dodish	dodish	NOT_FOUND	NOT_FOUND	1	0	9	9

# fixing accuracy

- now we create a `revised_correct` column with an edit distance cut-off of 2 and a `mismatch` column to track where our accuracy has changed

```
attention_trials = savic %>% filter(typeoftrial == "attention") %>%
 select(ID, response, novel1, novel2, novel3, correct) %>%
 rowwise() %>%
 mutate(response = ifelse(is.na(response), "blank", response)) %>%
 mutate(across(c(novel1, novel2, novel3), ~ replace_na(., "NOT_FOUND"))) %>%
 mutate(edit_novel1 = adist(novel1, response),
 edit_novel2 = adist(novel2, response),
 edit_novel3 = adist(novel3, response)) %>%
 mutate(revised_correct = ifelse(edit_novel1 < 2 |
 edit_novel2 < 2 |
 edit_novel3 < 2,
 1, 0),
 mismatch = ifelse(correct == revised_correct, 0, 1)) %>%
 ungroup()
```

	response	novel1	novel2	novel3	correct	edit_novel1[, 1]	edit_novel2[, 1]	edit_novel3[, 1]	revised_correct[, 1]	mismatch[, 1]
0	mipp	foobly	mipp	NOT_FOUND	0	6	0	9	1	1
0	geck	dodish	geck	NOT_FOUND	0	6	0	9	1	1
5	dodish	dodish	NOT_FOUND	NOT_FOUND	0	0	9	9	1	1
5	geck	dodish	geck	NOT_FOUND	0	6	0	9	1	1
8	foobly	foobly	mipp	NOT_FOUND	0	0	6	9	1	1
8	mipp	foobly	mipp	NOT_FOUND	0	6	0	9	1	1
8	mipp	foobly	mipp	NOT_FOUND	0	6	0	9	1	1
8	dodish	dodish	geck	NOT_FOUND	0	0	6	9	1	1
4	foobly	foobly	NOT_FOUND	NOT_FOUND	0	0	9	9	1	1
4	dodish	dodish	NOT_FOUND	NOT_FOUND	0	0	9	9	1	1
0	Heck	dodish	geck	NOT_FOUND	0	6	1	9	1	1
1	Dodish	dodish	NOT_FOUND	NOT_FOUND	0	1	9	9	1	1
8	floobly	foobly	NOT_FOUND	NOT_FOUND	0	1	9	9	1	1
8	dodidh	dodish	NOT_FOUND	NOT_FOUND	0	1	9	9	1	1
5	fooblu	foobly	NOT_FOUND	NOT_FOUND	0	1	9	9	1	1
5	godish	dodish	NOT_FOUND	NOT_FOUND	0	1	9	9	1	1
4	geck	dodish	geck	NOT_FOUND	0	6	0	9	1	1
3	Foobly	foobly	NOT_FOUND	NOT_FOUND	0	1	9	9	1	1

# analyses checklist

- confirm/correct all datatypes
- figure out how to “filter” certain types of trials
- fix all typos in attention responses
- compute mean attention accuracy
- apply exclusions based on accuracy AND RTs
- create RT bar graph
- fit a statistical model
- report statistics

# summarizing accuracy

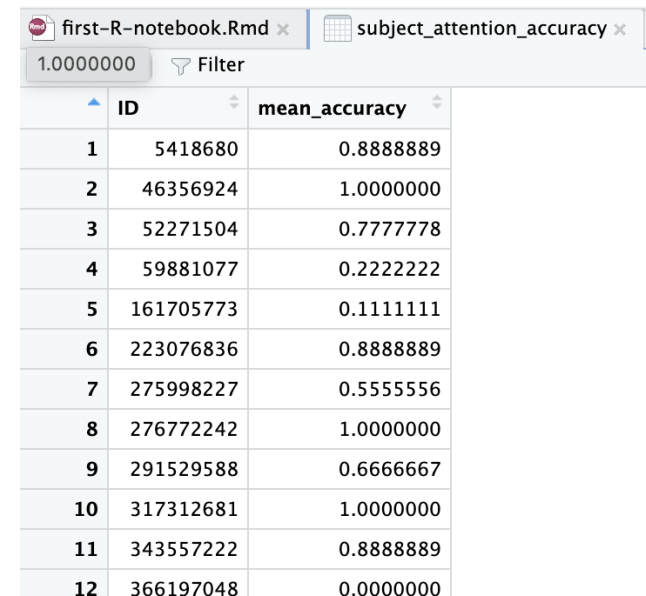
- calculate the mean accuracy and standard deviation across **all** attention trials
- how do we do this for **each** participant?

```
mean
```

```
attention_trials %>%
 summarize(mean_accuracy = mean(revised_correct),
 sd_accuracy = sd(revised_correct))
```

```
summarize participant accuracy
```

```
subject_attention_accuracy = attention_trials %>%
 group_by(ID) %>%
 summarize(mean_accuracy = mean(revised_correct))
```



The screenshot shows a data table with two columns: ID and mean\_accuracy. The table contains 12 rows of data. The ID column is sorted in ascending order, and the mean\_accuracy column shows the mean accuracy for each participant.

ID	mean_accuracy
1	0.888889
2	1.000000
3	0.777778
4	0.222222
5	0.111111
6	0.888889
7	0.555556
8	1.000000
9	0.666667
10	1.000000
11	0.888889
12	0.000000

# excluding participants

- what was our **exclusion criteria** for attention check?
- how do we find IDs that have accuracy less than 75%?
- storing these IDs in a vector will be useful later on

```
find IDs that have less than 75% accuracy

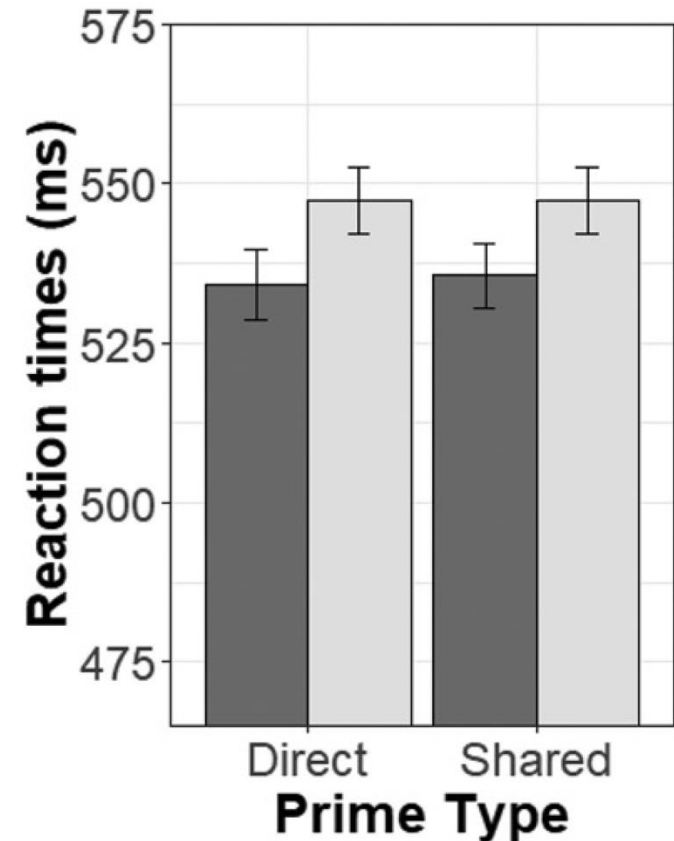
low_acc_IDs = subject_attention_accuracy %>%
 filter(mean_accuracy < 0.75) %>%
 pull(ID)
```

# analyses checklist

- confirm/correct all datatypes
- figure out how to “filter” certain types of trials
- fix all typos in attention responses
- compute mean attention accuracy
- apply exclusions based on accuracy AND RTs
- create RT bar graph
- fit a statistical model
- report statistics

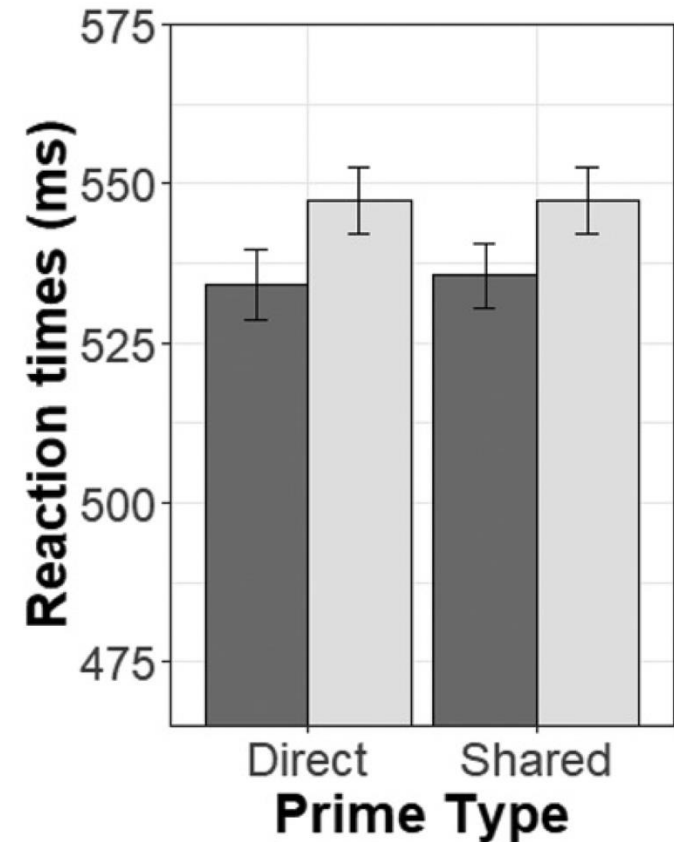
# priming data plan

- list out all the **steps** we will need to take to get to our intended plot from the raw data
- also list the **tidyverse functions** you may need to use for each step



# priming data plan

- **filter** for target trials
- **select** only relevant columns
- apply exclusions: **filter**, **%in%**
  - remove NA trials
  - $RT > 200$  and  $RT < 1500$
  - correct responses
  - non-practice trials
  - relatedness is related/unrelated
  - type is direct/shared
  - remove low accuracy IDs
- compute means per condition
  - **summarize**





# priming trials: filtering

- `filter` for target trials

```
priming_data = savic %>% filter(typeoftrial == "target")
```

# priming trials: selecting

- **filter** for target trials
- **select** only relevant columns

```
priming_data = savic %>% filter(typeoftrial == "target") %>%
 select(ID, rt, relatedness, prime, response, type, correct, block_number, target, correct_key)
```

# priming trials: filtering/exclusions

- **filter** for target trials
- **select** only relevant columns
- apply exclusions: **filter**, **%in%**
  - remove NA trials
  - RT > 200 and RT < 1500
  - correct responses
  - non-practice trials
  - relatedness is related/unrelated
  - type is direct/shared
  - remove low accuracy IDs

```
priming_data = savic %>% filter(typeoftrial == "target") %>%
 select(ID, rt, relatedness, prime, response, type, correct, block_number, target, correct_key)%>%
 filter(!is.na(rt))
```

```
filter(!is.na(rt), rt > 200 , rt < 1500)
```

```
filter(!is.na(rt), rt > 200 , rt < 1500, correct == "TRUE")
```

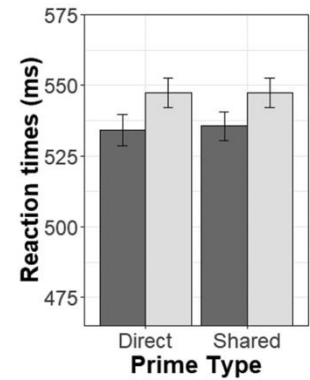
```
filter(!is.na(rt), rt > 200 , rt < 1500, correct == "TRUE", block_number == 1)
```

```
filter(!is.na(rt), rt > 200 , rt < 1500, correct == "TRUE", block_number == 1) %>%
 filter(relatedness %in% c("related", "unrelated") & type %in% c("direct", "shared"))
```

```
priming_data = savic %>% filter(typeoftrial == "target") %>%
 select(ID, rt, relatedness, prime, response, type, correct,
 block_number, target, correct_key)%>%
 filter(!is.na(rt), rt > 200 , rt < 1500, correct == "TRUE", block_number == 1) %>%
 filter(relatedness %in% c("related", "unrelated") & type %in% c("direct", "shared")) %>%
 filter(!ID %in% low_acc_IDs)
```

# priming trials: compute means

- use the priming data to compute **means** for each condition



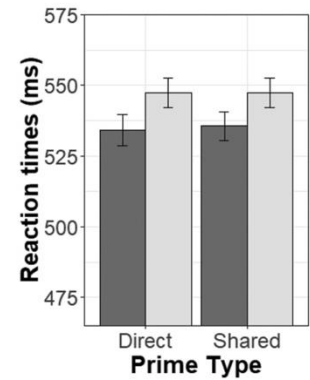
```
plot
```

```
```{r}
priming_data %>%
  group_by(type, relatedness) %>%
  summarise(mean_rt = mean(rt))
```

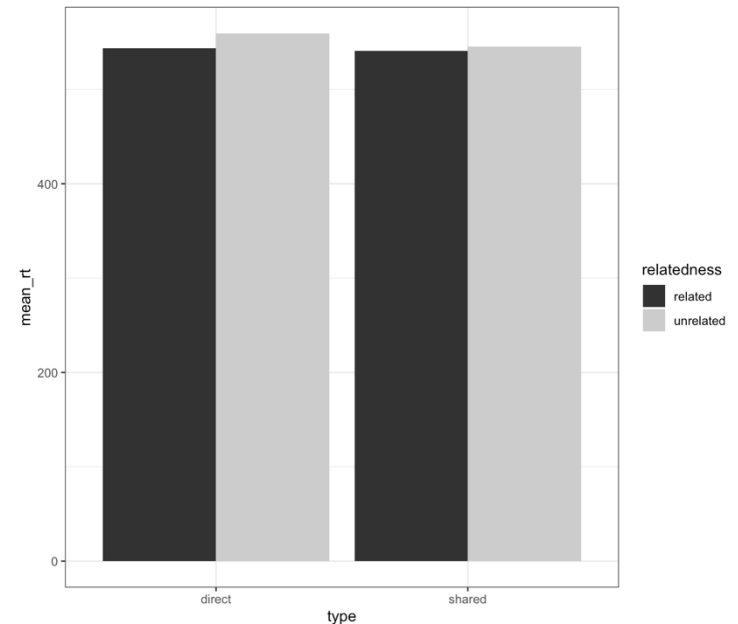
```
  type    relatedness mean_rt
<fct> <fct>         <dbl>
1 direct related      544.
2 direct unrelated    560.
3 shared related      541.
4 shared unrelated    545.
```

plot priming data

- directly pass the data from the means into `ggplot()`
- interpretation?



```
priming_data %>%  
  group_by(type, relatedness) %>%  
  summarise(mean_rt = mean(rt)) %>%  
  ggplot() +  
  geom_col(mapping = aes(x = type, y = mean_rt,  
                        group = relatedness, fill = relatedness),  
          position = "dodge") +  
  theme_bw() +  
  scale_fill_grey()
```



analyses checklist

- confirm/correct all datatypes
- figure out how to “filter” certain types of trials
- fix all typos in attention responses
- compute mean attention accuracy
- apply exclusions based on accuracy AND RTs
- create RT bar graph
- fit a statistical model
- report statistics

next class

- **before** class
 - *work on*: project checklist
 - *schedule*: a meeting with me before Tuesday next week
 - *apply*: formative assignment #2 (R descriptive)
- **during** class
 - making statistical inferences from data!