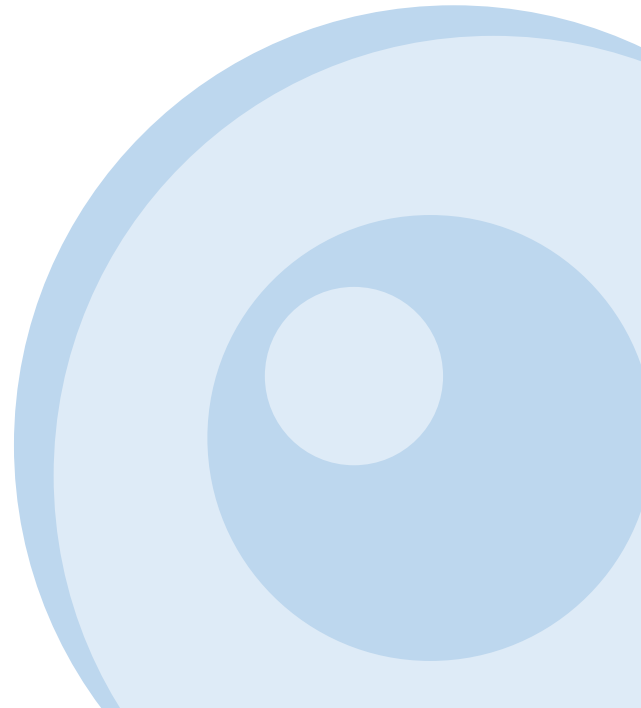


PSYC 193R:

Data Analysis & Visualization with R

Lecture 9: Paired-samples t-test

Jonas S.-H. Lau [silau -et- ucsd.edu]
psyc193r.ucsd.edu



Blind-tasting: The Design

- What was the research question?
- What was the target population?
- What were the IV(s) and DV(s)?
- How were the IV(s) and DV(s) operationalized?
(also, scale of measurement?)
- Was it a between / within-subjects design?
- What are the expected results?
If the manipulation does not work:
If the manipulation works:

Practical Research Tips:

There are multiple ways of doing paired-samples t tests

- Data structure

Import Data

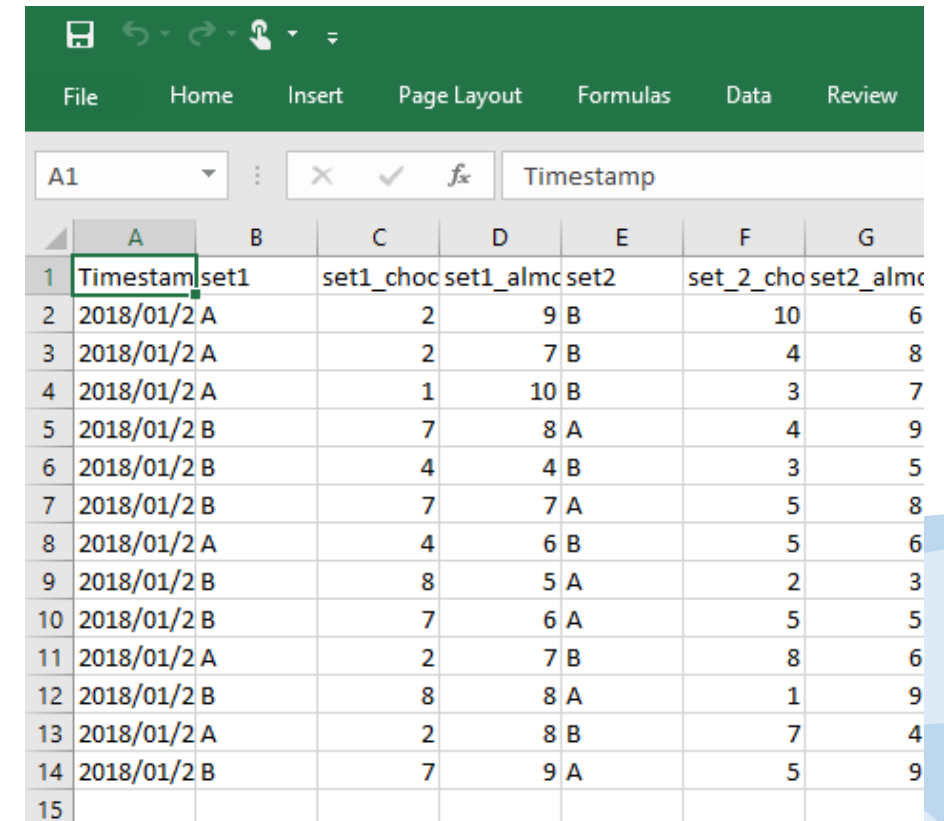
- Download the data file:
https://psyc193r.ucsd.edu/data/psyc193r_blindTaste_data.csv
- Tell R where your data file is:
Session → Set Working Directory → Choose Directory
`setwd()`
- Check the *comma separated values* (csv) file with MS Excel to ensure correct format
- Import the file into R as a data frame, saving it to “blindTasteData” variable
`... = read.csv("file_name.csv")`

Data format

The data set has a "wide" format:
multiple responses in the same row

The variables that are relevant:

- `set1`: the first set to evaluate
- `set1_chocolate`:
quality of the chocolate of the first set
- `set1_almond`:
quality of the almond of the first set
- `set2`: <same> but for 2nd set
- `set2_chocolate`: <same> but for 2nd set
- `set2_almond`: <same> but for 2nd set
- Make sure they are in the correct `class`, i.e., factor / numeric



	A	B	C	D	E	F	G
1	Timestamp	set1	set1_choc	set1_almc	set2	set_2_cho	set2_almc
2	2018/01/2	A	2	9	B	10	6
3	2018/01/2	A	2	7	B	4	8
4	2018/01/2	A	1	10	B	3	7
5	2018/01/2	B	7	8	A	4	9
6	2018/01/2	B	4	4	B	3	5
7	2018/01/2	B	7	7	A	5	8
8	2018/01/2	A	4	6	B	5	6
9	2018/01/2	B	8	5	A	2	3
10	2018/01/2	B	7	6	A	5	5
11	2018/01/2	A	2	7	B	8	6
12	2018/01/2	B	8	8	A	1	9
13	2018/01/2	A	2	8	B	7	4
14	2018/01/2	B	7	9	A	5	9
15							

The Ideal Data Frame: setA / setB columns

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

psyc193r_blindTaste.R x psyc193r_I08_paired-sample-t_jonas.... x blindTasteData x

Filter

chocolate	set1_almond	set2	set2_chocolate	set2_almond	gender	like_chocolate	like_almond	set1_score	set2_score	setA	setB	diff
2	9	B	10	6	Female	10	8	11	16	11	16	5
2	7	B	4	8	Male	9	10	9	12	9	12	3
1	10	B	3	7	Female	10	10	11	10	11	10	-1
7	8	A	4	9	Male	7	8	15	13	13	15	2
4	4	B	3	5	Female	4	4	8	8	8	8	0
7	7	A	5	8	Male	9	7	14	13	13	14	1
4	6	B	5	6	Female	7	7	10	11	10	11	1
8	5	A	2	3	Female	6	5	13	5	5	13	8
7	6	A	5	5	Female	8	7	13	10	10	13	3
2	7	B	8	6	Female	5	5	9	14	9	14	5
8	8	A	1	9	Female	10	8	16	10	10	16	6
2	8	B	7	4	Male	8	8	10	11	10	11	1
7	9	A	6	9	Male	9	9	16	14	14	16	2

Showing 1 to 13 of 13 entries

Data Cleaning / Processing Plan

- `factorize()` / `as.numeric()` the relevant variables

- Remove invalid entries

- Participants evaluating only A or B

```
blindTasteData = blindTasteData[blindTasteData$set1 != __ ,]
```

- Make a composite score for each set

- Combining each participant's set 1 chocolate + almond scores (`set1_score`)
 - Do the same for set 2 (`set2_score`)

```
blindTasteData$set1_score = blindTasteData$set1_chocolate + ...
```



Data Processing Plan

- Make 2 new columns, one for each set
 - setA
 - setB

```
blindTasteData$setA = NA  
blindTasteData$setB = NA
```

- Make a for() loop to go over each row,
if set1 == "A", setA == set1_score, etc

```
for (i in 1:nrow(blindTasteData)){  
  if (blindTasteData$set1[i] == "A"){  
    blindTasteData$setA[i] = NA  
    ...  
  } else {  
    blindTasteData$setA[i] = NA  
    ...  
  }  
}
```

The Ideal Data Frame: setA / setB columns

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

psyc193r_blindTaste.R x psyc193r_I08_paired-sample-t_jonas.... x blindTasteData x

Filter

chocolate	set1_almond	set2	set2_chocolate	set2_almond	gender	like_chocolate	like_almond	set1_score	set2_score	setA	setB	diff
2	9	B	10	6	Female	10	8	11	16	11	16	5
2	7	B	4	8	Male	9	10	9	12	9	12	3
1	10	B	3	7	Female	10	10	11	10	11	10	-1
7	8	A	4	9	Male	7	8	15	13	13	15	2
4	4	B	3	5	Female	4	4	8	8	8	8	0
7	7	A	5	8	Male	9	7	14	13	13	14	1
4	6	B	5	6	Female	7	7	10	11	10	11	1
8	5	A	2	3	Female	6	5	13	5	5	13	8
7	6	A	5	5	Female	8	7	13	10	10	13	3
2	7	B	8	6	Female	5	5	9	14	9	14	5
8	8	A	1	9	Female	10	8	16	10	10	16	6
2	8	B	7	4	Male	8	8	10	11	10	11	1
7	9	A	6	9	Male	9	9	16	14	14	16	2

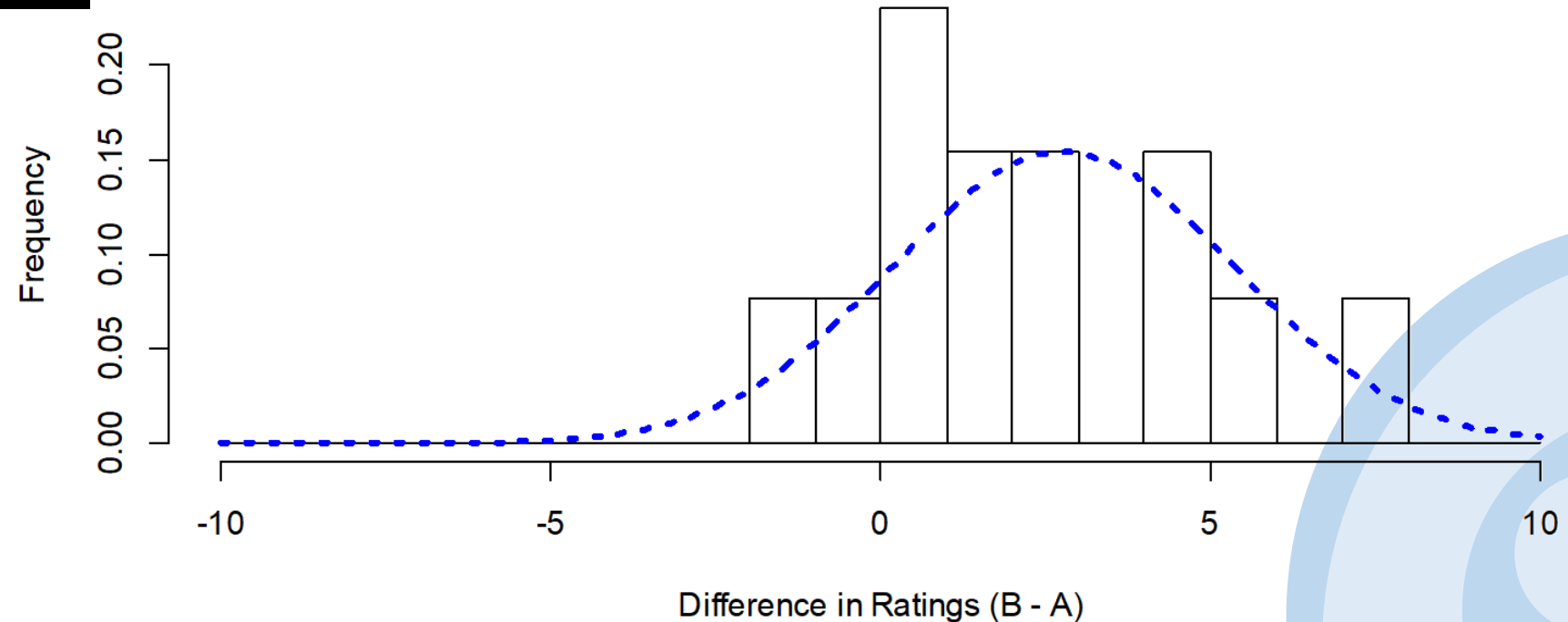
Showing 1 to 13 of 13 entries

Assumptions / Outlier Checking

Checking assumptions on the difference scores

- Make a “difference” score variable: `diff = setB - setA`
- Plot a histogram of the difference scores

```
hist(...)
```



t.test() on wide data frame

Option 1:

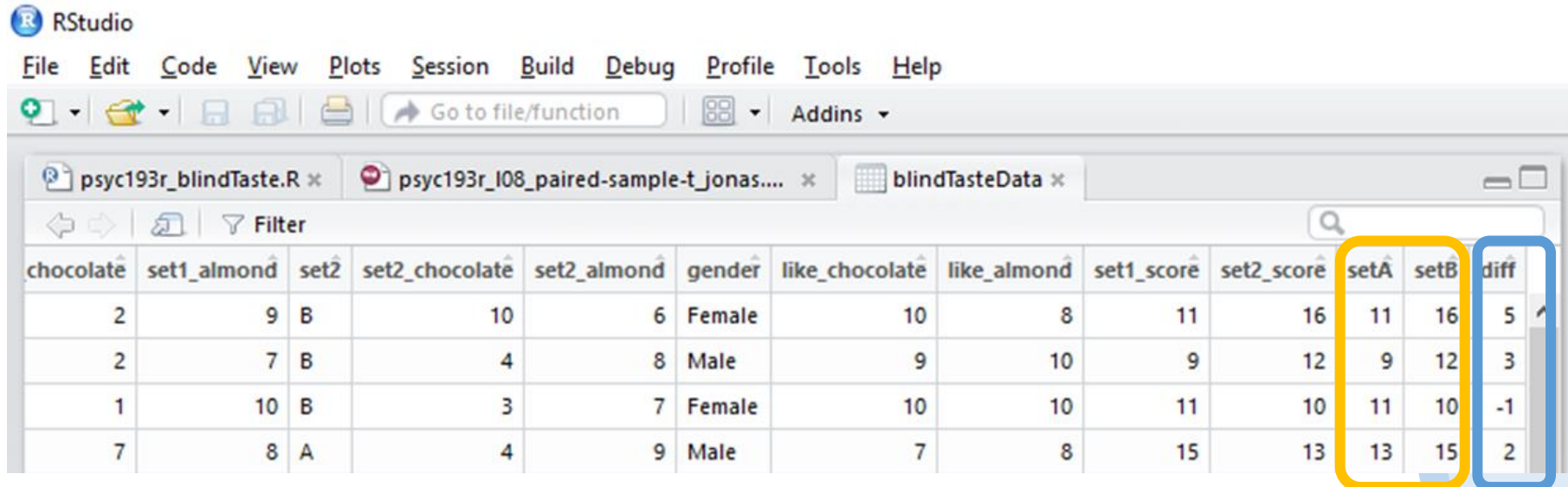
t.test() on the difference scores, a one-sample t-test ☐

```
t.test( __ , mu = 0, paired = FALSE)
```

Option 2:

t.test() using the setA and setB arrays, a paired-samples t-test ☐

```
t.test( __ , __ , mu = 0, paired = TRUE)
```



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

psyc193r_blindTaste.R x psyc193r_I08_paired-sample-t_jonas.... x blindTasteData x

Filter

chocolate	set1_almond	set2	set2_chocolate	set2_almond	gender	like_chocolate	like_almond	set1_score	set2_score	setA	setB	diff
2	9	B	10	6	Female	10	8	11	16	11	16	5
2	7	B	4	8	Male	9	10	9	12	9	12	3
1	10	B	3	7	Female	10	10	11	10	11	10	-1
7	8	A	4	9	Male	7	8	15	13	13	15	2

Hypothesis Testing – Step-by-step

- Step 0: Gather all the information you need

- Check assumptions
- Data cleaning

- Step 1: Set up the hypotheses

- H_0 (null hypothesis)
- H_1 (Research hypothesis)

- Step 2: Calculate the test statistic

- Step 3: Find the probability associated with the test statistic

(i.e., the probability of getting such a score or a more extreme score x_2 , given H_0 is true)

- Step 4: Decision & Conclusion
(compare p against α)

$\mu_{\text{baseline_difference}} = 0$ (theoretical)

$S_{\text{difference}} = \text{sd}()$

$M_{\text{difference}} ; n ; df = n - 1 \rightarrow \text{t.test}()$

kurtosis of population ? $n < 30$ [Caution]

$H_0: \mu_{\text{difference}} = 0$

$H_1: \mu_{\text{difference}} \neq 0$

See `t.test()` output

See `t.test()` output

Report baseline & sample means, standard deviation, sample size, mean difference, test statistics & p-value, decision/ conclusion/ notes

Effect Size

The effect size tells us how big the average difference between the two conditions is, in standard deviation

$$\text{Cohen's } d = \frac{|\mu_{\text{difference}} - \mu_{\text{baseline_difference}}|}{S_{\text{difference}}}$$



Reliability Measure: Confidence Interval

When you repeat the same experiment (with the same condition and n), for ___% of the time, the sample mean difference will lie between [__ , __]

$$M_{difference} \pm t_{\frac{\alpha}{2}, df} \times S_{M_difference}$$

Or, obtain in `t.test()` output



Power analysis with pwr package

- Load the package each time after restarting R, if you need the package

```
library("pwr")
```

- Usage

```
?pwr.t.test
```

```
pwr.t.test(n = NULL, d = NULL, sig.level = 0.05,  
power = NULL, type = c("two.sample", "one.sample", "paired"),  
alternative = c("two.sided", "less", "greater"))
```

- Leave the argument that you are looking for as NULL

- E.g.,

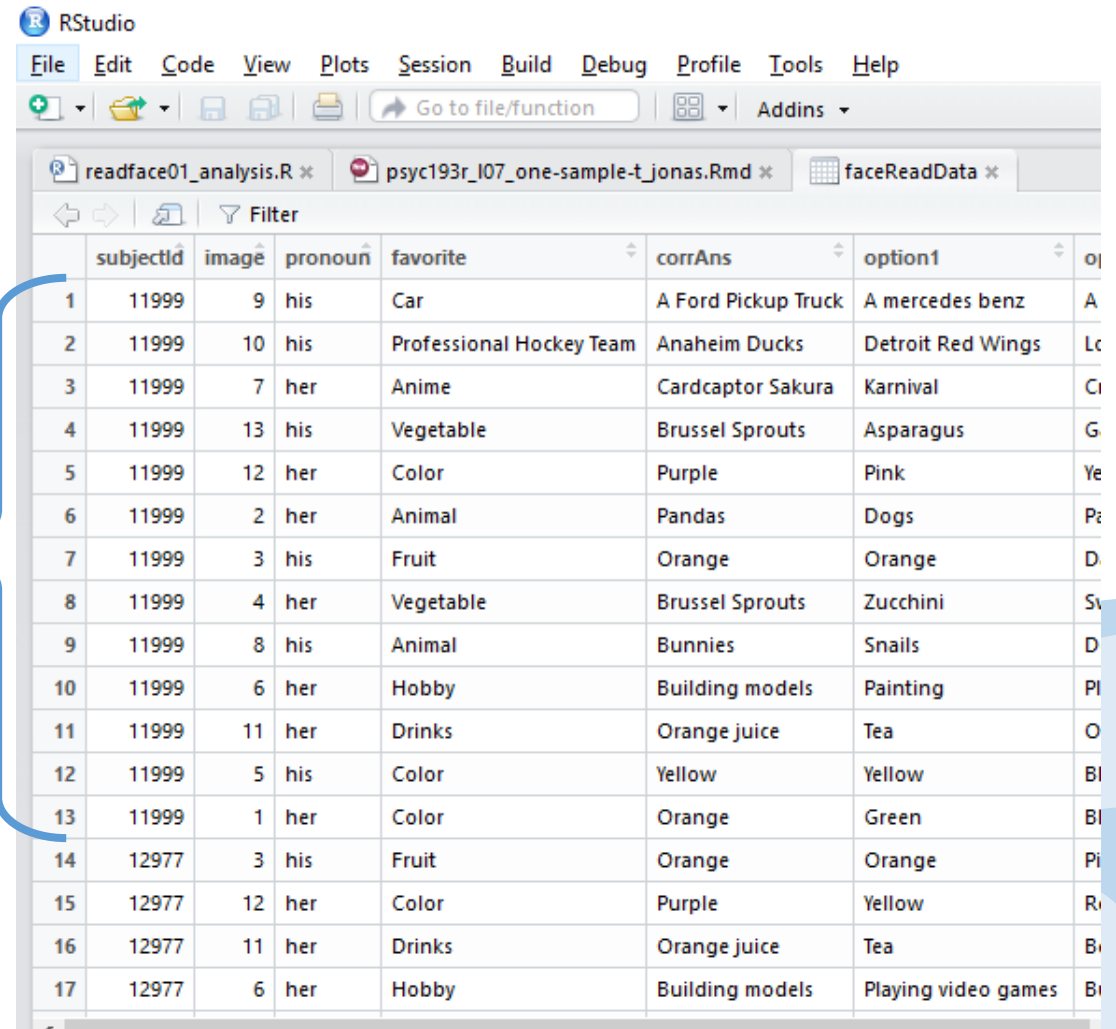
```
pwr.t.test(n = 50, d = 0.5, sig.level = 0.05, power  
= NULL, type = "one.sample")
```

```
pwr.t.test(n = 50, d = 0.5, sig.level = 0.05, power  
= NULL, type = "paired")
```

`t.test()` with "long" format data

Face-Reading Data Structure

- The data have a "long format":
 - Each column represents a variable
 - Each row represents an entry (each trial)
- The variable we care the most are "subjectId", "pronoun", & "responseTime"
 - subjectId: a number that represents a participant (each should have 13 rows)
 - pronoun: denotes the trial showed a male or female face
 - responseTime: denotes the time taken for a participant to respond



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

readface01_analysis.R * psyc193r_I07_one-sample-t_jonas.Rmd * faceReadData *

Filter

	subjectId	image	pronoun	favorite	corrAns	option1	responseTime
1	11999	9	his	Car	A Ford Pickup Truck	A mercedes benz	A
2	11999	10	his	Professional Hockey Team	Anaheim Ducks	Detroit Red Wings	Lc
3	11999	7	her	Anime	Cardcaptor Sakura	Karnival	Cr
4	11999	13	his	Vegetable	Brussel Sprouts	Asparagus	G
5	11999	12	her	Color	Purple	Pink	Ye
6	11999	2	her	Animal	Pandas	Dogs	Pa
7	11999	3	his	Fruit	Orange	Orange	D
8	11999	4	her	Vegetable	Brussel Sprouts	Zucchini	Sw
9	11999	8	his	Animal	Bunnies	Snails	D
10	11999	6	her	Hobby	Building models	Painting	Pl
11	11999	11	her	Drinks	Orange juice	Tea	O
12	11999	5	his	Color	Yellow	Yellow	Bl
13	11999	1	her	Color	Orange	Green	Bl
14	12977	3	his	Fruit	Orange	Orange	Pi
15	12977	12	her	Color	Purple	Yellow	Ri
16	12977	11	her	Drinks	Orange juice	Tea	Bi
17	12977	6	her	Hobby	Building models	Playing video games	Br

Face-Reading: New Research Question

In the face reading experiment, do people spend more time looking at the pictures when it is a male / female?

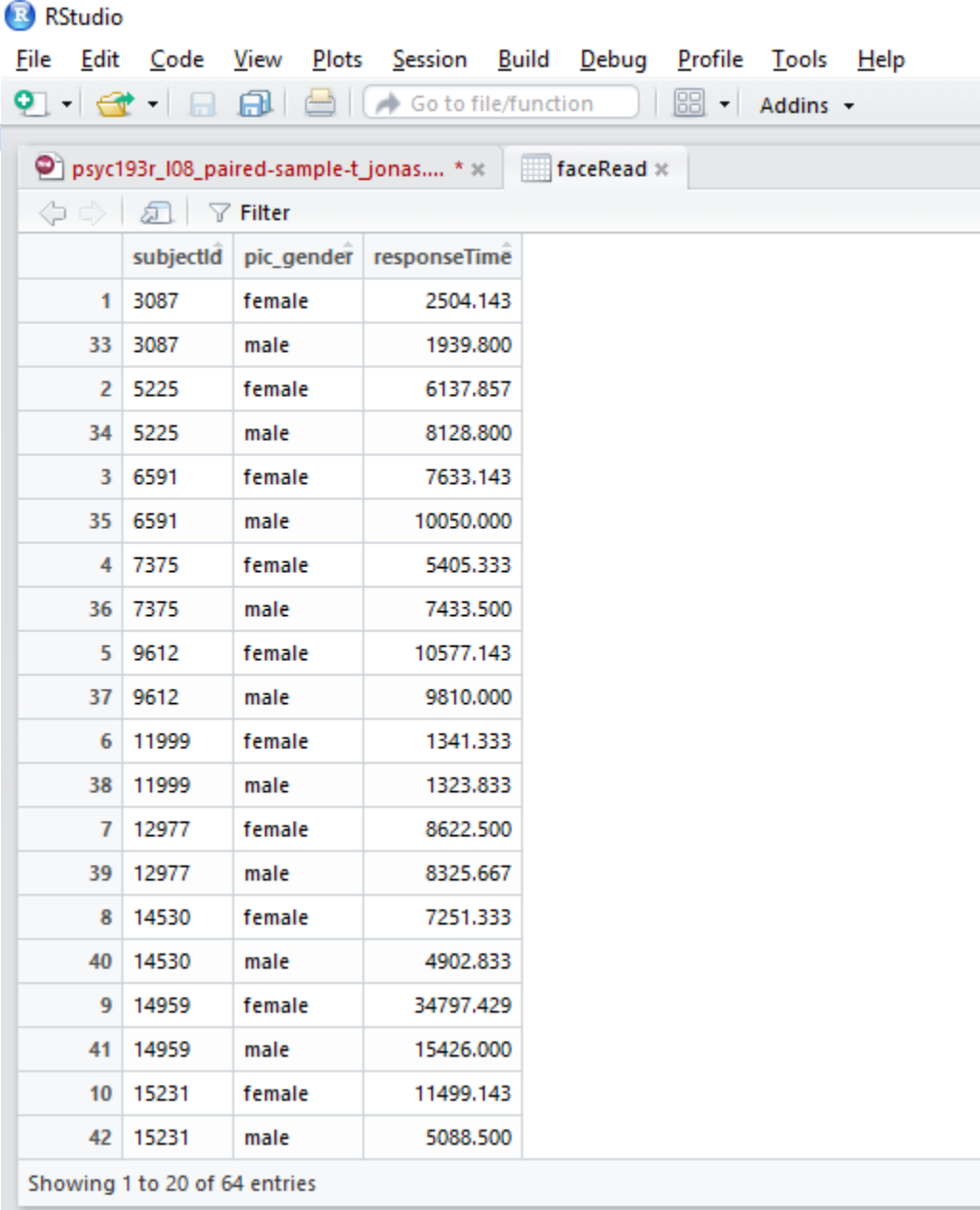
- Every participant perform both male-face and female-face trials
- We will aggregate each participant's response time when they looked at male and female faces



Ideal Data Structure

Three columns (variables)

- `subjectId` : two rows for each participant
- `pic_gender` : one row for female, another for male picture
- `responseTime` : aggregate response times for that condition (seeing female or male picture)



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

psyc193r_l08_paired-sample-t-jonas.... * x faceRead x

Filter

	subjectId	pic_gender	responseTime
1	3087	female	2504.143
33	3087	male	1939.800
2	5225	female	6137.857
34	5225	male	8128.800
3	6591	female	7633.143
35	6591	male	10050.000
4	7375	female	5405.333
36	7375	male	7433.500
5	9612	female	10577.143
37	9612	male	9810.000
6	11999	female	1341.333
38	11999	male	1323.833
7	12977	female	8622.500
39	12977	male	8325.667
8	14530	female	7251.333
40	14530	male	4902.833
9	14959	female	34797.429
41	14959	male	15426.000
10	15231	female	11499.143
42	15231	male	5088.500

Showing 1 to 20 of 64 entries

Data Formatting / Processing / Cleaning

- Factorize subjectId and pronoun

```
faceReadData$subjectId = factor(faceReadData$subjectId)
```

- Coerce responseTime to a numeric variable

```
faceReadData$responseTime = as.numeric(faceReadData$responseTime)
```

- Create a new pic_gender variable from pronoun for clarity

```
faceReadData[faceReadData$pronoun == "her",]$pic_gender = ...
```

- Remove entries with "this is my friend" responses

```
faceReadData = faceReadData[faceReadData$chosen != __ ,]
```

Data Formatting / Processing / Cleaning

- Aggregate responseTime for male and female pictures for each participant

```
faceRead = aggregate(responseTime ~ _____ * _____ ,  
data = faceReadData, FUN = _____ )
```

- Sort the columns to make sure each participant has 2 rows

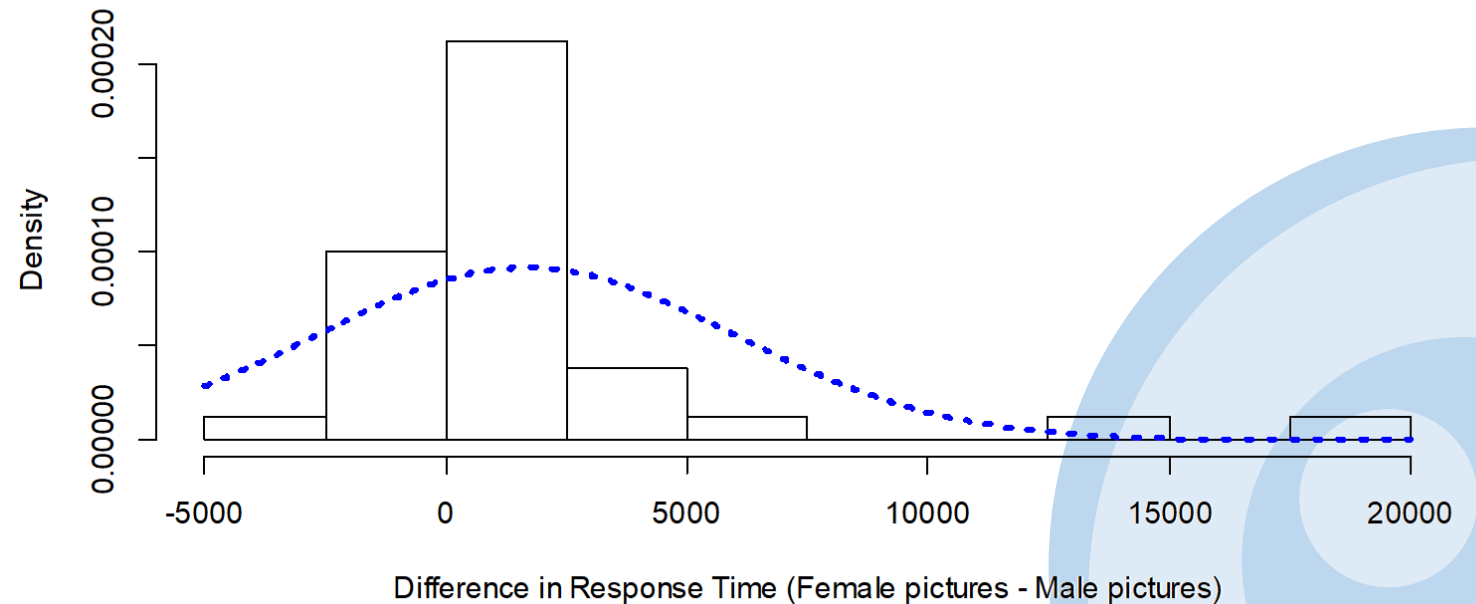
```
... = faceRead[order(faceRead$subjectId, faceRead$pic_gender),]
```



Checking Assumptions

Checking assumptions on the difference scores

- Normality assumption: sample size > 30
- Outliers: There were clear outliers in the data, so results should be interpreted with caution



Performing `t.test()`

```
t.test(responseTime ~ pic_gender, mu = 0, data = faceRead,  
paired = TRUE)
```

and `pic_gender` as IV (technically no manipulation)

Perform a `t.test()` with `responseTime` as DV

and this is a paired-samples t-test

Paired t-test

data: responseTime by pic_gender

t = 2.1546, df = 31, p-value = 0.03908

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

88.58827 3228.98955

sample estimates:

mean of the differences

1658.789

[Data contain outliers. Interpret with caution]

Summing up – paired-samples `t.test()`

- Depending on the data structure:

- "wide" data format:

```
t.test(x = diff_array, mu = 0 , paired = FALSE)
```

```
t.test(x = measure_1, y = measure_2, mu = 0 , paired = FALSE)
```

- "long" data format:

```
... = data[order(data$subjectId, data$iv),]
```

```
t.test(dv ~ iv, mu = 0, data = ..., paired = TRUE)
```

- Effect size: code it yourself
- 95% Confidence Interval: output from `t.test()`
- Power: `pwr.t.test()`