Essential Research Toolkit for the Humanities

Week 5: R workflow

Anna Pryslopska

May 9, 2022

Psycholinguistics and Cognitive Modeling Lab

Questions?

1. What do the following evaluate to and why?

```
FALSE + 0L                                   0
1 - FALSE                                    1
FALSE + 1                                    1
!TRUE                                    FALSE
5 & TRUE                                  TRUE
0 & TRUE                                 FALSE
1 | FALSE                                 TRUE
FALSE | NA                                  NA
```

2. Why do the following functions fail?

```
Summary(moses)
read_cvs(moses.csv)
tail(moses, n==10)
describe(Moses)
filter(moses, Condition == 102)
arragne(moses, ID)
mutate(moses, Items = as.character("Item"))
```

3. Clean up the Moses illusion data and save it to a new data frame.

```
moses_cleaned <- select(moses, c(ID, Item,
Condition, Answer))

moses_cleaned <- na.omit(moses_cleaned)

moses_cleaned <- arrange(moses_cleaned, c(Item,
Condition))

prince = c("prince", "prince (charming)", "a
prince", "the prince")

printing <- c("buchdruck", "invention of printing",
"printed books", "printing", "printing press", "the
letterpress")
```

…

4. Solve the logic exercise.

🐓 🦆
🦕 🐊

bird:
🐓 🦆

can swim:
🦆 🐊

🦕 🐊
🐓 🦕
🦆

🐊
🦕
🐓
🐓 🦆 🐊
🦆 🦕 🐊
🐓 🦆 🦕 🐊
∅

!bird

!can swim

bird & can swim

!bird & can swim

!bird & !can swim

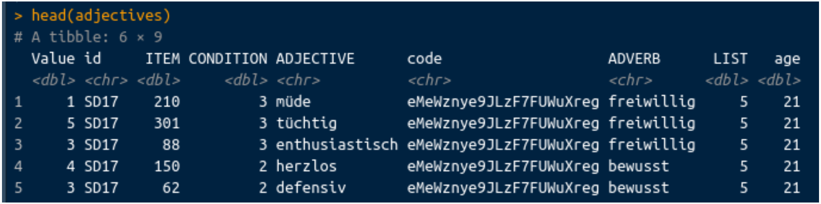
bird & !can swim

bird | can swim

!bird | can swim

bird | !bird
bird & !bird

5. Tidy up the `adjectives.csv` data.



```
> head(adjectives)
# A tibble: 6 × 9
  Value id    ITEM CONDITION ADJECTIVE      code                   ADVERB      LIST   age
  <dbl> <chr> <dbl>     <dbl> <chr>          <chr>                  <chr>      <dbl> <dbl>
1     1 SD17    210         3 müde           eMeWznye9JLzF7FUWuXreg freiwillig     5    21
2     5 SD17    301         3 tüchtig        eMeWznye9JLzF7FUWuXreg freiwillig     5    21
3     3 SD17     88         3 enthusiastisch eMeWznye9JLzF7FUWuXreg freiwillig     5    21
4     4 SD17    150         2 herzlos        eMeWznye9JLzF7FUWuXreg bewusst        5    21
5     3 SD17     62         2 defensiv       eMeWznye9JLzF7FUWuXreg bewusst        5    21
```

	Info you were given	Info to infer	
`Value`	acceptability rating on 1–7 scale	1 number	
`id`	participant ID 1–63	4 characters	
`ITEM`	sentence ID 1–360	number 1:3 digits	=ADJECTIVE
`CONDITION`	sentence group 1–3	1 number	=ADVERB
`ADJECTIVE`	adjective used in the sentence	character	=ITEM
`code`	random letters and numbers	character	
`ADVERB`	adverb used in the sentence	character	=CONDITION
`LIST`	version of experiment 1–6	1 number	
`age`	age of participant in years	number 18–80	

Standing in the way of control

Anna war absichtlich kalt/mager/liebevoll/einarmig/... .

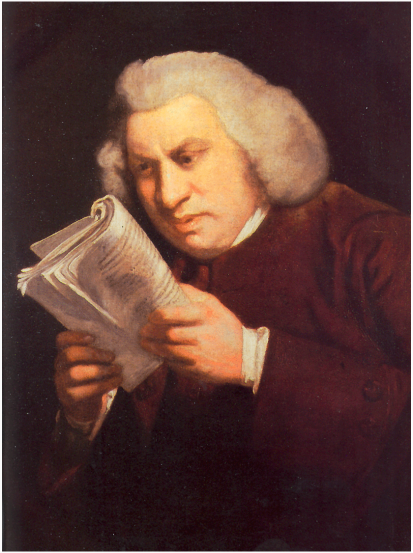
Anna war bewusst kalt/mager/liebevoll/einarmig/... .

Anna war freiwillig kalt/mager/liebevoll/einarmig/... .

Task: How natural is the sentence to you?

1 = unnatural, 7 = perfectly natural

Wait a minute



Something's not right

Carefully inspect the data, because it's usually a hot mess.

Do the values make sense?

→ summary/describe, mean vs. range, sorting, filtering etc.

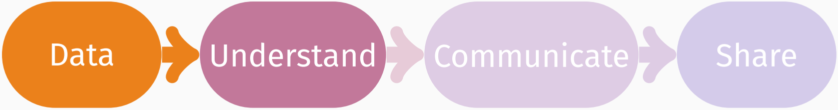
- `Value`: expected 1–7, but have 100! Typo?
- `LIST`: expected 1–6, but have -5! Error?
- `AGE`: expected 18–80, but have 2! Troll?
- `ADVERB`: expected words, but have 123! Error?

Is the data incomplete?	NAs in `Value`, `ITEM`, `ADJECTIVE`
Are there too many columns?	select meaningful variables

Let's get this sorted out!

Remove missing values	`na.omit()`
Select relevant variables	`select()`
Sort the values	`arrange()`
Filter weird values	`filter(adjectives, Value %in% 1:7 &…)`

Recap



R & RStudio, packages, data types, formats, encoding

import from workspace, assign values, operations, clean, filter, arrange, select, merge, group, summarize, export, visualize

document, create clean and beautiful reports

connect, collaborate, backup

Table of contents

1. Tidy code
2. Transform data
3. Getting help
4. Summary and homework assignment

Tidy code

A rose by any other name

Names uniquely identify variables and functions

Capitalization matters `DATA != Data != data`

Everyone has an opinion/preference

- `for_snake_case`
- `somePreferCamelCase`
- `others.use.periods`
- `Some_people.just.WANT_To_Watch.theWorldBURN`

R has no official style guide, but preference for alphanumeric & _

Naming variables and functions

What do I call you?

Use descriptive names and be consistent!

- `data` or `ratings` or `corpus` ✔
- `d` or `aaa` or `foo` or `temp` ✖
- reduce effort
- stay comprehensible and meaningful (good science is reproducible!)
- easy to remember and self-explanatory
- length doesn't matter (much): use `TAB` to autocomplete

Pipes

Pipes

Powerful tool for clearly expressing a sequence of multiple operations.

Created using `%>%` and can be read as “and then”.

The pipe translates `x %>% f(y)` into `f(x, y)`.

Passes the output as the new input.

```
moses_clean1 <- na.omit(moses)
moses_clean2 <- select(moses_clean1, c(Item, Condition, Answer))
moses_clean3 <- filter(moses_clean2, Condition %in% 1:2)
moses_clean4 <- arrange(moses_clean3, Answer)
```

```
moses_clean4 <- moses %>%
                na.omit() %>%
                select(c(Item, Condition, Answer)) %>%
                filter(Condition %in% 1:2) %>%
                arrange(Answer)
```

Why? Simplify code, remove clutter and potential for error, reduce effort, stay reproducible.

Why not? No intermediate steps (need to run the whole code), writing fuctions is more complex.

When not? Very long pipes (>10 lines), multiple inputs or outputs, creating plots.

Transform data

Finish cleaning

In the tidy data, create a new column `Answer_cleaned` using the functions `mutate()` and `fct_collapse()`.

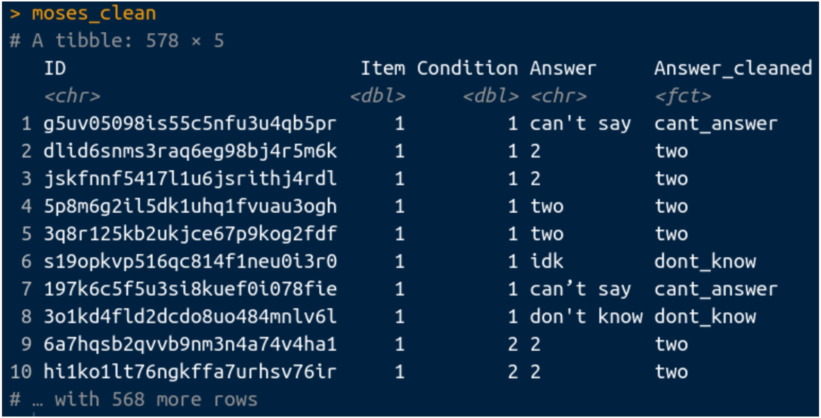
✓ `cant_answer, dont_know, armstrong, everest, madrid, manchester, nobel, olympics, platypus, prince, printing, roman, sagrada, santa, scholz, shakespeare, squirrel, switzerland, ten, two, uk, usa, valentines, whale`

```
prince <- c("prince", "prince (charming)", "a prince",
"the prince")
```

✓ `mutate(WHERE, NEW = FUNCTION(VALUES))`

⚠ `fct_collapse(WHERE, NEW VALUE = OLD VALUES)`

```
moses_clean <-
  moses %>%
    na.omit() %>%
    select(ID, Item, Condition, Answer) %>%
    arrange(Item, Condition) %>%
    mutate(Answer_cleaned = fct_collapse(Answer,
          cant_answer = cant_answer,
          dont_know = dont_know,
          armstrong = armstrong,
          everest = everest,
          madrid = madrid,
          manchester = manchester,
          nobel = nobel,
          olympics = olympics,
          platypus = platypus,
          prince = prince,
          printing = printing,
          …)
```



```
> moses_clean
# A tibble: 578 × 5
   ID                          Item Condition Answer     Answer_cleaned
   <chr>                      <dbl>     <dbl> <chr>      <fct>         
 1 g5uv05098is55c5nfu3u4qb5pr     1         1 can't say  cant_answer   
 2 dlid6snms3raq6eg98bj4r5m6k     1         1 2          two           
 3 jskfnnf5417l1u6jsrithj4rdl     1         1 2          two           
 4 5p8m6g2il5dk1uhq1fvuau3ogh     1         1 two        two           
 5 3q8r125kb2ukjce67p9kog2fdf     1         1 two        two           
 6 s19opkvp516qc814f1neu0i3r0     1         1 idk        dont_know     
 7 197k6c5f5u3si8kuef0i078fie     1         1 can’t say  cant_answer   
 8 3o1kd4fld2dcdo8uo484mnlv6l     1         1 don't know dont_know     
 9 6a7hqsb2qvvb9nm3n4a74v4ha1     1         2 2          two           
10 hi1ko1lt76ngkffa7urhsv76ir     1         2 2          two           
# … with 568 more rows
```

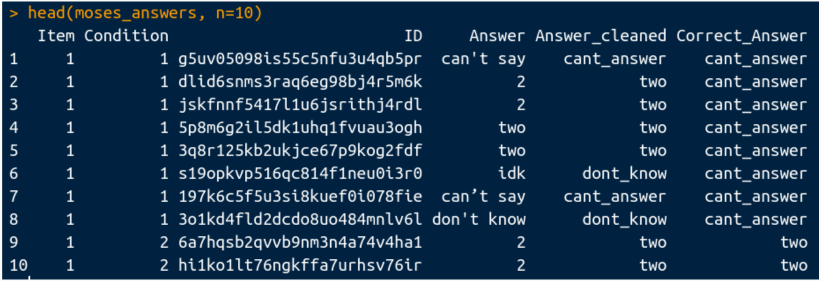
Add correct answers

Download `correct.csv` from ILIAS and read it in as `correct_answer`.

Use the `merge()` function to combine `moses_clean` and `correct_answer`.

```
merge(x=DATA1, y=DATA2, by=COLUMNS)
```

```
moses_answers <-
  moses_clean %>%
    merge(correct_answer, by = c("Item", "Condition"))
```



```
> head(moses_answers, n=10)
   Item Condition                       ID     Answer Answer_cleaned Correct_Answer
1     1         1 g5uv05098is55c5nfu3u4qb5pr  can't say    cant_answer    cant_answer
2     1         1 dlid6snms3raq6eg98bj4r5m6k          2            two    cant_answer
3     1         1 jskfnnf5417l1u6jsrithj4rdl          2            two    cant_answer
4     1         1 5p8m6g2il5dk1uhq1fvuau3ogh        two            two    cant_answer
5     1         1 3q8r125kb2ukjce67p9kog2fdf        two            two    cant_answer
6     1         1 s19opkvp516qc814f1neu0i3r0        idk      dont_know    cant_answer
7     1         1 197k6c5f5u3si8kuef0i078fie  can’t say    cant_answer    cant_answer
8     1         1 3o1kd4fld2dcdo8uo484mnlv6l don't know      dont_know    cant_answer
9     1         2 6a7hqsb2qvvb9nm3n4a74v4ha1          2            two            two
10    1         2 hi1ko1lt76ngkffa7urhsv76ir          2            two            two
```

Calculate accuracy: What if?

Was the answer **correct** or **incorrect**?

```
moses_answers %>%
  mutate(Accuracy = Answer_cleaned == Correct_Answer)
```

logical

```
ifelse(TEST, DO WHEN TRUE, DO WHEN FALSE)
```

```
moses_answers %>%
  mutate(Accuracy = ifelse(test=Answer_cleaned==Correct_Answer,
  yes=TRUE, no=FALSE))
```

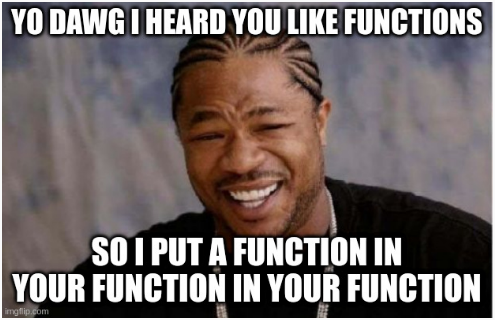
logical

```
moses_answers %>%
  mutate(Accuracy = ifelse(test=Answer_cleaned==Correct_Answer,
  yes="correct", no="incorrect")
```

strings

Calculate accuracy

Was the answer **correct**, **incorrect**, or did you **not know**?


YO DAWG I HEARD YOU LIKE FUNCTIONS
SO I PUT A FUNCTION IN YOUR FUNCTION IN YOUR FUNCTION
imgflip.com

```
moses_accuracy <-
  moses_answers %>%
    mutate(Accuracy = ifelse(Answer_cleaned == Correct_Answer,
        yes = "correct",
        no = ifelse(Answer_cleaned == "dont_know",
        yes = "dont_know", no = "incorrect")))
```

Goal of experiment: Trick, no treat

Groups/conditions in experiment

1 **Moses illusion**
“What is the name of the first man to walk on the *sun*?”

2 **Well-formed question**
“What is the name of the first man to walk on the *moon*?”

100 Well-formed control “In which country is Florida located?”

101 Bad control “Which Nordic country are coconut trees native to?”

Predictions

1 No correct answers but you will try to answer anyway
2 Correct answer is predefined (e.g. Armstrong)
100 Correct answer is predefined (e.g. USA)
101 No correct answers and you will notice this

Grouping and summarizing

Grouping

`group_by(WHERE, BY WHAT)` changes the unit of analysis from the complete dataset to particular groups.

`ungroup(WHERE)` undos grouping.

Useful for summaries: How did the groups compare?

Summarizing

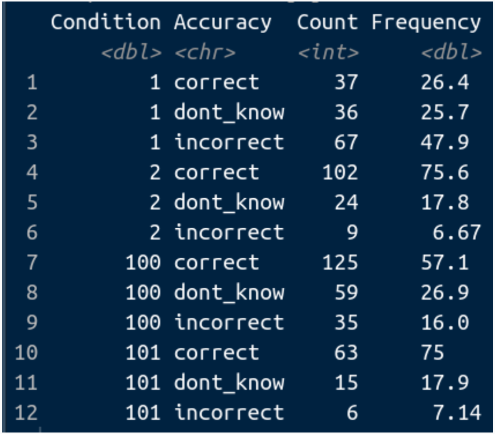
`summarise(WHERE, NEW=FUNCTION(VALUE))` calculates values.

```
summarise(my_awesome_data,
  Count = n(),                                  count cases
  Mean = mean(RT),                     average reading time
  SD = sd(RT),                     how spread out is the data
  Min = min(Rating))                            minimal value
```

`mutate()` changes an existing column or adds a new one.
`summarise()` calculates a single value (per group).

Did you get got?

```
moses_accuracy %>%
  group_by(Condition, Accuracy) %>%
  summarise(Count = n()) %>%
  mutate(Frequency = 100*Count / sum(Count))
```



```
   Condition Accuracy  Count Frequency
       <dbl> <chr>     <int>     <dbl>
 1         1 correct      37     26.4 
 2         1 dont_know    36     25.7 
 3         1 incorrect    67     47.9 
 4         2 correct     102     75.6 
 5         2 dont_know    24     17.8 
 6         2 incorrect     9      6.67
 7       100 correct     125     57.1 
 8       100 dont_know    59     26.9 
 9       100 incorrect    35     16.0 
10       101 correct      63     75   
11       101 dont_know    15     17.9 
12       101 incorrect     6      7.14
```

Getting help

A little stuck

In R

🤔 `help("NAME")` 😟 `?NAME` 😰 `??NAME`

Online

Google the exact error message

RDocumentation: `www.rdocumentation.org`

Cheatsheets: `www.rstudio.com/resources/cheatsheets`

Discord server: `discord.gg/CxFrknxzYV`

Tidyverse: `www.tidyverse.org`

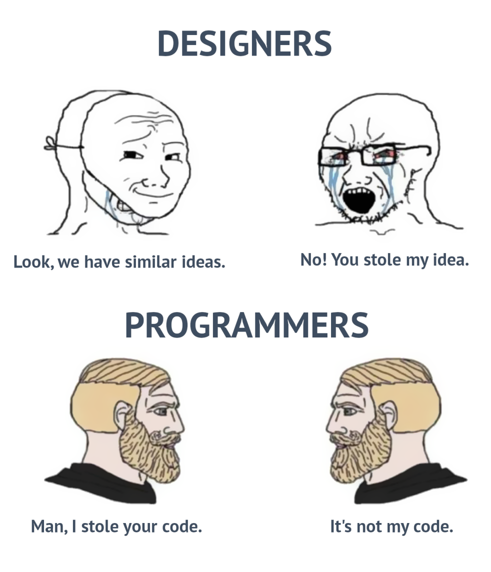
Stack Overflow: `stackoverflow.com`

Reddit: `reddit.com/r/rstats`

Create a *minimal reproducible example* `gist.github.com/hadley/270442`

Recommended reading: `www.r4wrds.com/intro/m_troubleshooting.html`

Everyone does it



Questions?

Summary and homework assignment

Summary: By now, you know how to...

· install and load packages	`install.package()`, `library()`
· read in data	`read_csv()`
· assign values	`<-`, `=`, `<<-`
· inspect data	`head()`, `summary()`, `describe()`, etc.
· operate on data	`&`, `|`, `!`, `+`, `-`, `/`, etc.
· filter data	`filter()`
· recognize missing values	`is.na()`
· remove missing values	`na.omit()`
· select variables	`select()`
· arrange data	`arrange()`
· create new variables	`mutate()`
· merge data frames	`merge()`
· divide data into group	`group_by()`
· summarize data	`summarize()`
· get help	`help()`, `?`, `??`

Homework assignment due May 16

- Complete assignment 3
- Read chapter 3 of R for Data Science `r4ds.had.co.nz/`
- Go through parts 1–6 of “Data visualisation using R, for researchers who don’t use R” (if you come across functions you don’t know, don’t worry, just run the code you’re provided): `psyteachr.github.io/introdataviz/`