C91AR | ADVANCED STATISTICS USING R

Lecture 4: Tidying realistic data

Dr Pete McKenna

2025-01-31

HERIOT WATT UNIVERSITY
SoSS Doctoral Centre

SETUP FOR THIS MODULE

- Create an R Markdown called “Lecture4_tiding_real_data”
- Load the following packages

```
# load packages
pacman::p_load(tidyverse,
               snakecase,
               psych,
               summarytools)
```

WHAT WE COVERED HAVE COVERED THUS FAR

- Setting up an RProject
- Creating sub-directories
- Functions in R
- Operators (e.g., `>`, `!=`, `==`) in R
- Creating an object with that data (e.g., `short_flights`)
- Writing tidy code with the *tidyverse* verbs

HERIOT WATT UNIVERSITY
SoSS Doctoral Centre

How you started the course…

HERIOT WATT UNIVERSITY
SoSS Doctoral Centre

How you might be feeling now…

HERIOT WATT UNIVERSITY
SoSS Doctoral Centre

SOME ADVICE MOVING FORWARD

- If this is your first experience of programming do not worry - it takes time to get into the philosophy of it
- I will be showing you new functions and arguments but, after week 6 this will ease up as we focus on data analysis.
- Do talk to your peers and ask questions in class
- Programming has a steep learning curve, but once you wrap your head around the principles it has be very rewarding (I promise!)

ANY QUESTIONS ABOUT WHAT WE HAVE COVERED SO FAR?

WHAT WE ARE MOVING ON TO

- Reading in real data from a data file
- Tidying vector labels and values
- Dealing with date data
- Summarising data
- **What we don't get through today we will continue with on Thursday**

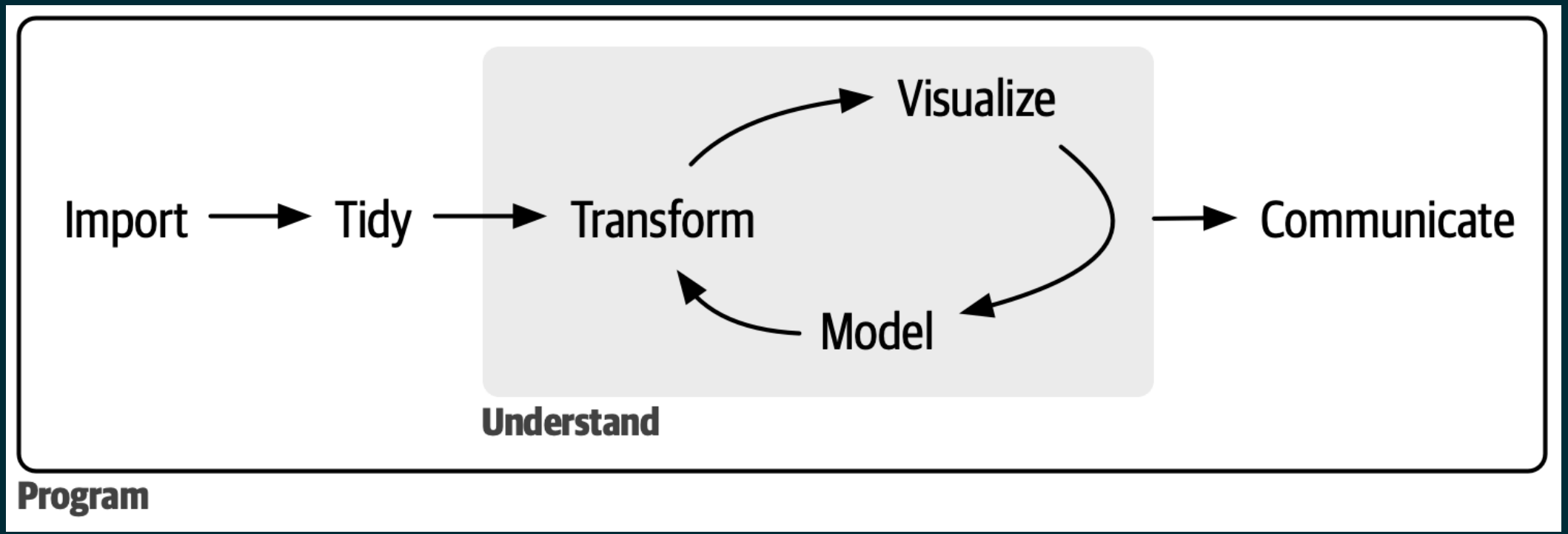
REALISTIC DATA IN RESEARCH (MCALEER ET AL. 2022)

- A core strength to using a programming language for data processing and analysis is that you reduce the risk of human error significantly.
- This is especially important when confronted with realistic datasets that may be untidy for historical or procedural reasons (e.g., a database that is maintained by different people who used their own unique way of storing and labelling things).
- An option would be to simply open the data and update the values as an when required.
- However, if the issue is prevalent across many datasets then an automated solution is more efficient.

READING IN AND TIDYING DATA

SAVING DATA IN YOUR DIRECTORY

- At the beginning of the course we create three subdirectories “data_raw”, “data_tidy”, and “fig_out”
- We are going to real untidy experiment data into the “data_raw” folder.
- The best format for R is to save data as a `.csv` but there are options to read in other file formats (e.g., `.xlsx`).
- Unprocessed data is saved in the “data_raw” folder and processed data in the “data_tidy” folder so that we the raw and processed data separate
- This will save you a lot of headache once you get to grips with processing data and creating new `.csv`s for different purposes.



HERIOT WATT UNIVERSITY
SoSS Doctoral Centre

SAVE THE RAW DATA PROVIDED

- Save the data from Canvas Module **Course datasets** called “robot-expression-data_raw.csv” into the “data_raw” folder

READ IN AND EXPLORE THE ROBOT EXPRESSION DATA

```
1  # Read in raw data
2  df <-
3    read_csv("data_raw/robot-expression-data_raw.csv")
```

EXPRESSION DATASET METADATA

- `ID_` = participant id
- `sex` = participant sex
- `eth` = participant ethnicity
- `eng` = “Is English your native language?”
- `Exp.date` = date of experiment
- `interact.before` = “Have you ever interacted wtih a robot before?
- `Alyx.is` = participants guess of robot’s gender
- `item` = plastic food item offered to robot
- `Expression` = a numeric value for each of the four robot facial expressions
- `resp` = name of the box where participants placed the food item after robot expression
- `Time` = robots onboard clock timer
- `resp_acc` = accuracy of response according to theoretically deigned expressio

DATA TYPES FOR THE `col_types` ARGUMENT

- “f” = factor (i.e., categorical variable)
- “l” = logical
- “i” = integer (no decimals)
- “d” = double numeric (has decimals)
- “c” = character
- “D” = date
- “_” = skip
- “?” = guess

- For more information see the `read_csv` and `col_types` documentation in the help section
- You can pull this up by running `?read_csv` into the console
  - I’d recommend consulting the descriptor info at the top then skipping to the examples at the bottom of the help sheet

SETTING COL TYPES WITHIN THE `read_csv` FUNCTION

```
# Read in data and supply data type argument
df <-
  read_csv("data_raw/robot-expression-data_raw.csv",
           col_types = "ffff?ffffftf")
```

- Now the data from the csv is stored in an object called `df`

DATA EXPLORATION

- Here are a few handy functions to exploring your data
  - `summary`: from base R
  - `headTail`: from *psych*
  - `dfSummary`: from *summarytools*

BASE R'S `summary` FUNCTION

- The `summary` function summarises continuous and categorical vectors
- NOTE: `summary` will not count character vectors, that's why we specified that our categorical vectors were factor type (e.g., "f") at `read_csv`

```
1  # data summary - overall
2  df |>
3    summary()
```

`psych::headTail` FUNCTION

- The `headTail` function from the *psyche* package

```
1  # data summary - head
2  headTail(df)
```

`summarytools` PACKAGE

- We've seen how we can get an overview of the data using `dfSummary`
- This function is particularly useful for spotting anomalies in the data

```
1  dfSummary(df)
```

USING SUMMARY FUNCTIONS WITH THE TIDYVERSE

- You can pipe data into summary functions to specify rows or cols of interest

```
# data summary - sex
df |>
  select(sex) |>
  summary()
```

```
     sex
Length:228
Class :character
Mode  :character
```

RECODING VARIABLE LEVELS

- You may have noticed from the last session that there there are some erroneous values in the sex vector.
- You can check this directly by using the `$` symbol to index the vector of interest in combination with the function `unique`.

```
1  # Check all unique entries in the vector `sex`
2  unique(df$sex)
3
4  # Alternatively you can use tidyverse's `distinct`
5  df |>
6    select(sex) |>
7    distinct()
```

- As you can see, something has gone wrong in the labelling of sex: there should only values, “Female” and “Male”

USING `case_match` TO RECODE DATA

- Previously, the `recode` function was used in combination with `mutate` to recode values in a cell
- But more recently, people have moved to using `case_match`, as it is more intuitive and user friendly
- Info on case_match

case_match SYNTAX

```
# Using case_match to recode cell values
df2 <-
  df |>  # create new and tidy object for illustration
  mutate(sex = case_match(sex,                # mutate to change
                          "male" ~ "Male",    # supply erroneous value and r
                          "ffemale" ~ "Female",
                          "femalew" ~ "Female",
                          .default = sex))    # preserve all other entries i

# Check the values of a vector using unique
unique(df2$sex) # again, we use the $ symbol to isolate a specific vector
```

RECAP ON THE ASSIGN OPERATOR CONSIDERATIONS

- **Irreversibly overwriting dataframe/tibble**

```
df <-
  df |>
  select(id, eth)

# df only contains data about ppt id and eth
```

Create new dataframes/tibbles

```
df1 <-
  df |>
  select(id, eth)

# df1 only contains data about ppt id and eth
# df contains all 12 vectors
```

HERIOT WATT UNIVERSITY
SoSS Doctoral Centre

SAVE YOUR TIDY DATA

- Once you are satisfied with the amendments made to your data you can save it as a new tidy `.csv` file
- Again, use comments and bookmarks to signpost this part of your script

```
# save your new data object into `data_tidy`
write_csv(df, # name of the object
          "data_tidy/robot-expression-data_tidy.csv") # desired directory
```

HERIOT WATT UNIVERSITY
SoSS Doctoral Centre

CLEANUP

```r
# Clear data
rm(list = ls())  # Removes all objects from environment

# Clear packages
p_unload(all)  # Remove all contributed packages

# Clear plots
graphics.off()  # Clears plots, closes all graphics devices

# Clear console
cat("\014")  # Mimics ctrl+L
```

REFERENCES

McAleer, Phil, Niamh Stack, Heather Woods, Lisa Marie DeBruine, Helena Paterson, Emily Nordmann, Carolina Ellen Kuepper-Tetzel, and Dale J. Barr. 2022. “Embedding Data Skills in Research Methods Education: Preparing Students for Reproducible Research,” November. https://doi.org/10.31234/osf.io/hq68s.

HERIOT WATT UNIVERSITY

SoSS Doctoral Centre


HERIOT
WATT
UNIVERSITY
SoSS Doctoral
Centre