

# SAS macros

Hadley Wickham

# Outline

- Variables
- Titles and footnotes
- Macros
- Debugging macros

# Variables

- To create: `%let var = blah;`
- To use: `&var`
- Or `&var.` if next to something else
  
- Just like copying and pasting
- Don't need quotes, inserted as is

# Titles and footnotes

- Can annotate output with titles and footnotes. Useful to add subsetting info etc.
- `title | "Useful information";`
- `footnote | "Footnote goes here":`
- `title | "";` /\* to turn off \*/

# Your turn

- Load the server survey data (again)
- Use proc means to print a summary of a subset of the data, and include the subset in the title
- Set up variables that contain lists of different types variables (demographic, behaviours, personality, etc)

# Macros

- Kind of like functions in R
  - Except user and developers no different in R.
  - In SAS you create macros, developers create procs
- Just do text substitution

# Syntax

- `%macro mymacro(var1, var2=default)`
- ...
- `%mend`
  
- `%mymacro(elephant)`

# Procedure

- Create macros the same way we created function in R:
  - Create the variables we'll use, and write the procedures using those variables
  - Test
  - Then wrap up in a macro

# Your turn

- Create a macro, that for a given set of variables:
  - Summarises them using `proc freq` and `proc means`
  - Tries to predict `pcttip` with them, and display least square means
- Remember to create an informative title

# Debugging macros

- options mprint mlogic;
- Then look at the log to see what's going on

# Advanced features

- Conditional statements:
  - `%if condition %then %do;`
  - `%else %do;`
  - `%end;`

# Advanced features

- Loops
- Useful in conjunction with %scan

# Your turn

- Read about one of the more advanced features on the additional documentation of the website and try it out