


Macros



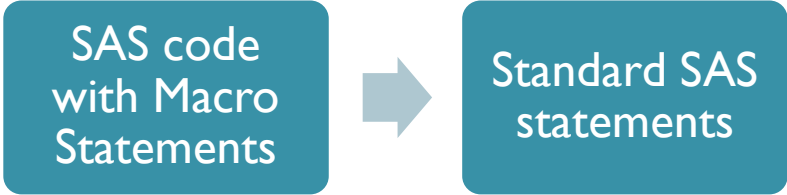
Programming ...
 Read.
 Watch.
 Do.
 Repeat doing until
 you get the hang of it.

AIM POPULATION INFORMATICS
 CC BY-NC-SA

1

SAS Macro (%)

Macro Preprocessor

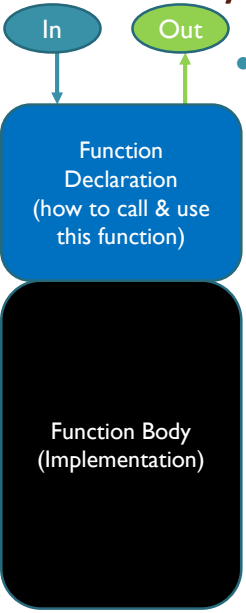


- Macro variables
- Macro functions (macros) : not normally called functions

AIM POPULATION INFORMATICS
 CC BY-NC-SA

2

Why use Functions?



- **Encapsulation**
 - Black box programming
 - Hides internal details of algorithm from users
 - Users typically only care about using the function to get results.
- Isolates computations, protects variables
 - Interaction through arguments
- Separates interface and implementation
 - Interface: what a function does
 - Implementation: how a function does it

AIM POPULATION INFORMATICS
CC BY-NC-SA

3

Double dot

```

%let ver=2;
%let lname=src;
%let lnameassign_=wrong;

set &lnameassign_&ver;
set wrong2;

set &lname.assign_&ver;
set srcassign_2; * in work folder;

set &lname..assign_&ver;
set src.assign_2;
  
```



AIM POPULATION INFORMATICS
CC BY-NC-SA

4

Debugging Macros

- MPRINT
- SYMBOLGEN
- MLOGIC
- %put
- %include
 - config.sas



```
Options MPRINT MLOGIC SYMBOLGEN;  
* Look at log;
```

 POPULATION INFORMATICS


5

Agenda: 3 weeks, 6 lectures

- What would we cover?
 - Data Science?
 - ods
 - File I/O
 - Privacy ?
 - More programming in front of you
 - You program in class, ask me questions
 - Review midterm

 POPULATION INFORMATICS


6