

Effective Programming Practices for Economists

# Debugging

(Armchair) Psychology of debugging

Janoś Gabler and Hans-Martin von Gaudecker

# The urge to skip reading

Tracebacks are long and not always helpful, so most people have the urge to skip them. Similarly, it takes a lot of effort to thoroughly read code line by line.

- Read tracebacks until you
  - Know in which line (of your code) the error occurred
  - Have found the type of error that occurred
  - Have read the message at the end of the traceback
- When you located a bug to less than 30 lines of code
  - Read every line and explain to someone what it does
  - Someone can be a rubber duck

# The urge to just run it again

When code does not run, there is a tendency to just execute it again.

- Computers are deterministic
- If you run the same code twice, it produces the same result
- If it does not run the first time but it runs the second time, you have a way bigger problem than you thought!

# The urge to tell yourself it should work

There is tendency during debugging to explain to yourself that and why the code should work.

- Computers do exactly what you tell them to do in a deterministic fashion
- If you are debugging, something did not work
- Don't explain why it should work, when it clearly does not

# The urge to try out things

Errors are stressful, we want them to go away. So there is a tendency to make any changes that could fix errors.

- Take a deep breath and relax
- Make sure to commit before you try out things
- Change one thing at a time, for a reason!

# The urge to blame libraries

We thought our code was correct, so there is a tendency to blame problems on Python itself, libraries, or library versions

- Most libraries we use are
  - well tested
  - written by far by better programmers than us
- Many used so much that any bug would have already been found by someone else
- Until you have a lot of evidence to the contrary, assume libraries are correct
- If you find a bug in a library, report it to the authors!