spncrfx@gmail.com

@foxandtheflu

Reproducible Research in R

Spencer Fox

03 November 2017

Acknowledgements

- Heavily drawn from Karl Broman’s website
  - http://kbroman.org/steps2rr

The 5 commandments

The 5 commandments

1. Organize
2. Script
3. Functions
4. Version Control
5. Automate

The 5 commandments

1. Organize
2. Script
3. Functions
4. Version Control
5. Automate

R Packages!

The 5 commandments

1. **Organize**
2. Script
3. Functions
4. Version Control
5. Automate

You need to know where to find everything for you project

- Everything in one folder
- Separate data from code
- Separate data from summary data
- Use relative paths
- Descriptive file names with summary
- Be consistent across projects
- Create a README file

Have no fear, R packages get you half the way there

R package creation demo

The 5 commandments

1. Organize
2. **Script**
3. Functions
4. Version Control
5. Automate

Everything in scripts

- Get as raw data as possible
- No manual changes
- Every analysis step needs to be scripted
    - e.g. reading in data, cleaning data, analyzing data, plotting figures

Setting up scripts to divide tasks

Scripting demo

The 5 commandments

1. Organize
2. Script
3. **Functions**
4. Version Control
5. Automate

Creating functions for repeated tasks

- Anything repeated multiple times can get a function with descriptive name
- Functions can be in their own scripts
- Main script will be easier to read and follow

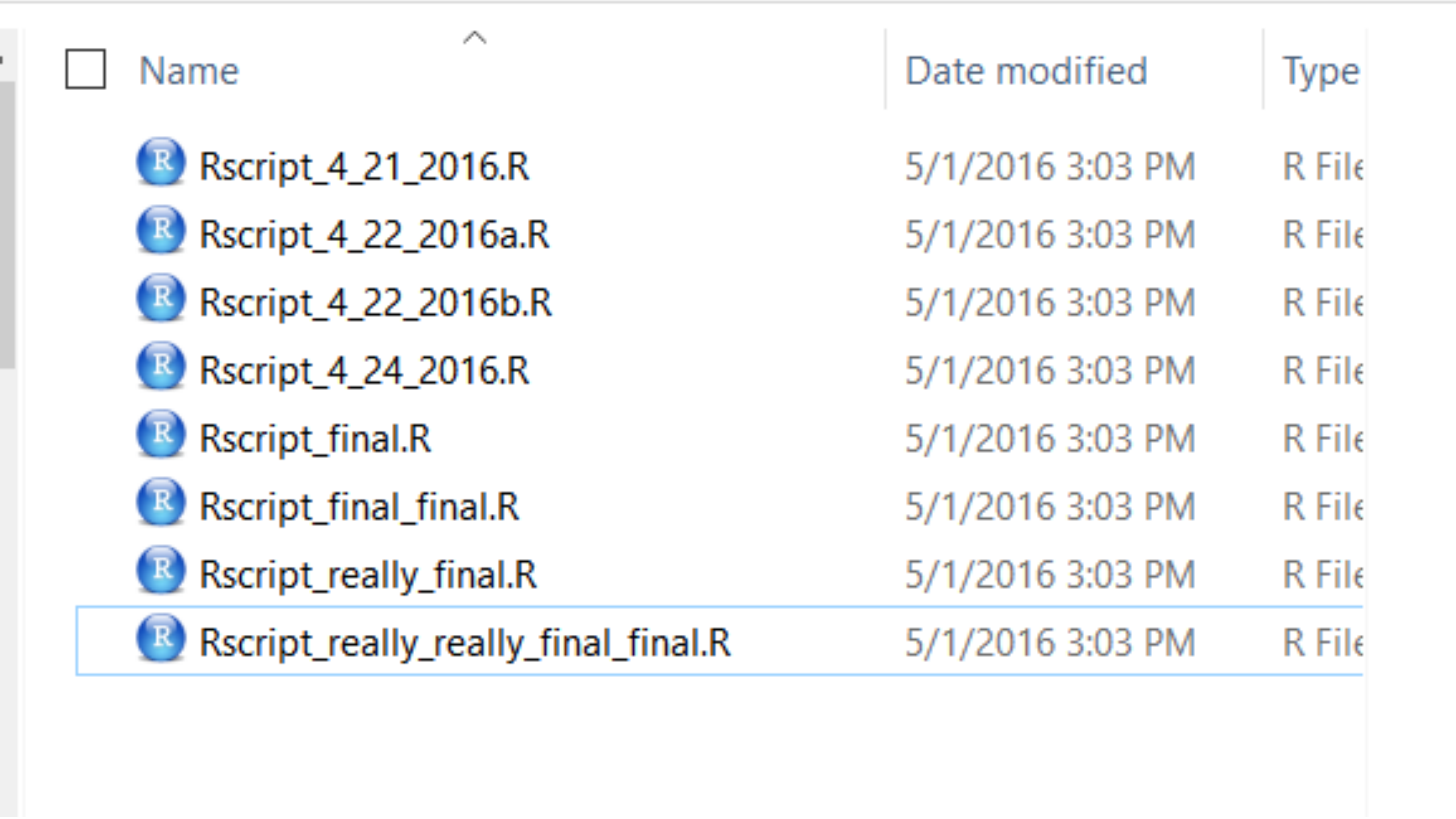
Setting up functions for clean scripts

The 5 commandments

1. Organize
2. Script
3. Functions
4. **Version Control**
5. Automate

Github or Die Tryin’

Traditional version control issues



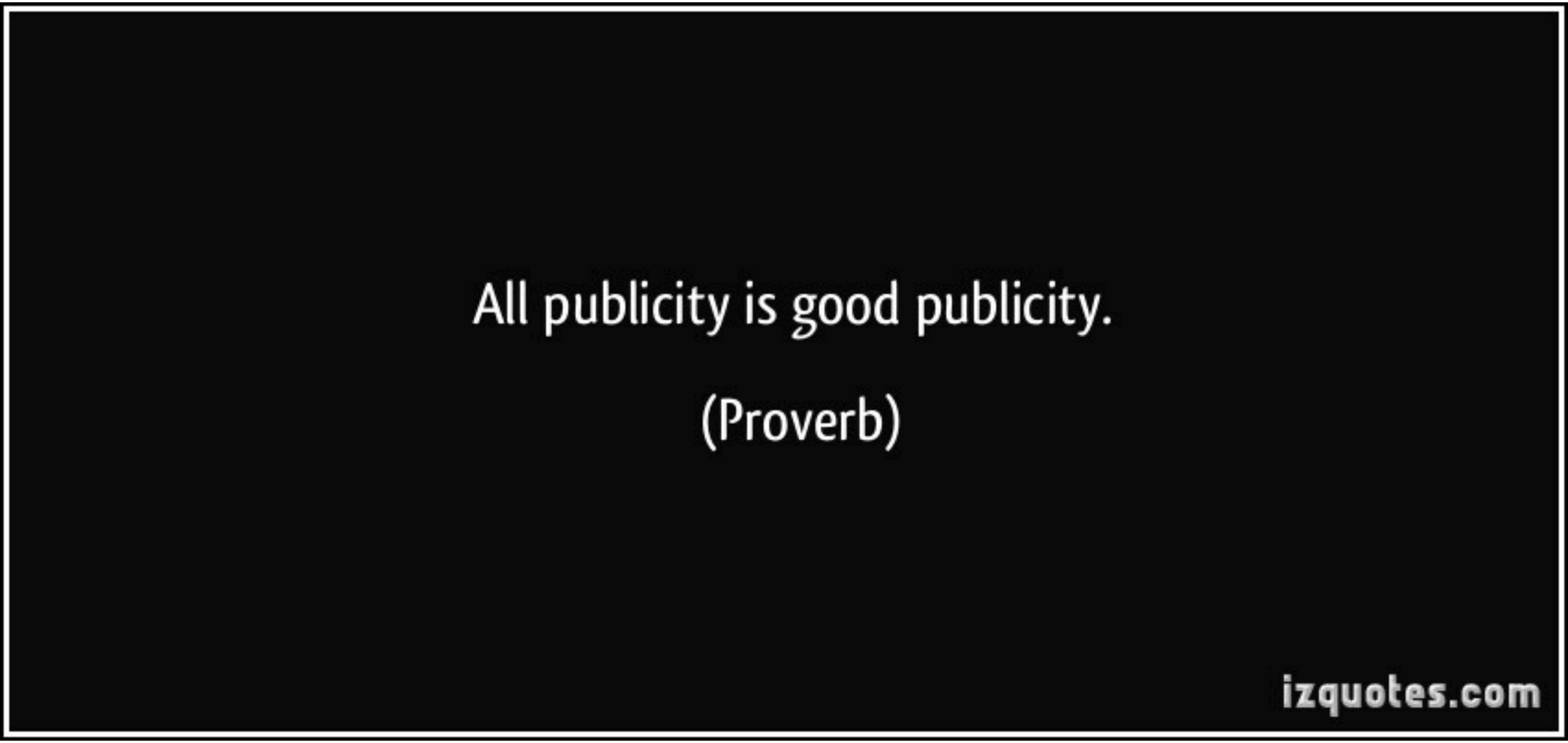
Name	Date modified	Type
Rscript_4_21_2016.R	5/1/2016 3:03 PM	R File
Rscript_4_22_2016a.R	5/1/2016 3:03 PM	R File
Rscript_4_22_2016b.R	5/1/2016 3:03 PM	R File
Rscript_4_24_2016.R	5/1/2016 3:03 PM	R File
Rscript_final.R	5/1/2016 3:03 PM	R File
Rscript_final_final.R	5/1/2016 3:03 PM	R File
Rscript_really_final.R	5/1/2016 3:03 PM	R File
Rscript_really_really_final_final.R	5/1/2016 3:03 PM	R File

Facilitates collaboration and prevents waiting

vs.

A public GitHub account gets you exposure that can lead to jobs and other opportunities

A public GitHub account gets you exposure that can lead to jobs and other opportunities

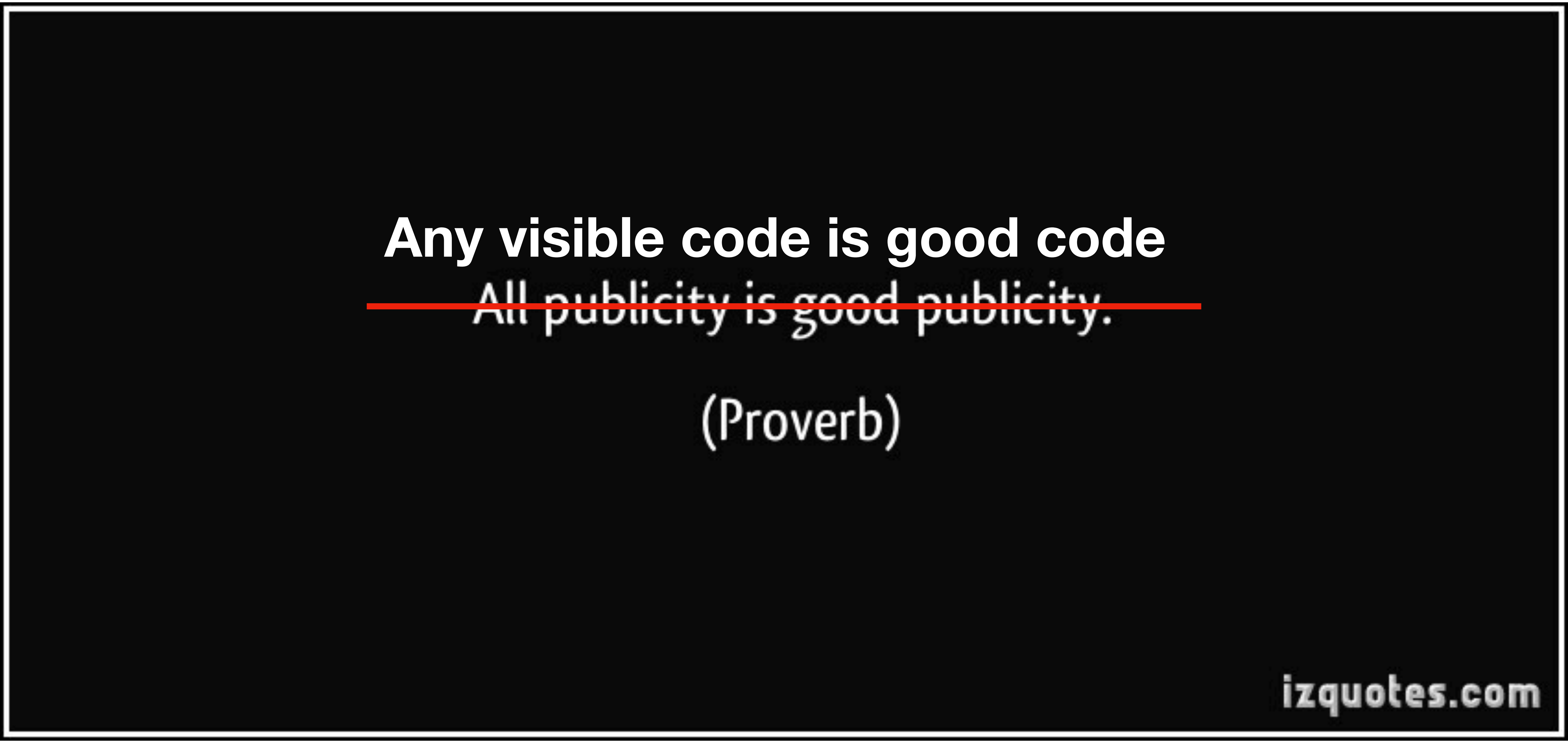


All publicity is good publicity.

(Proverb)

izquotes.com

A public GitHub account gets you exposure that can lead to jobs and other opportunities

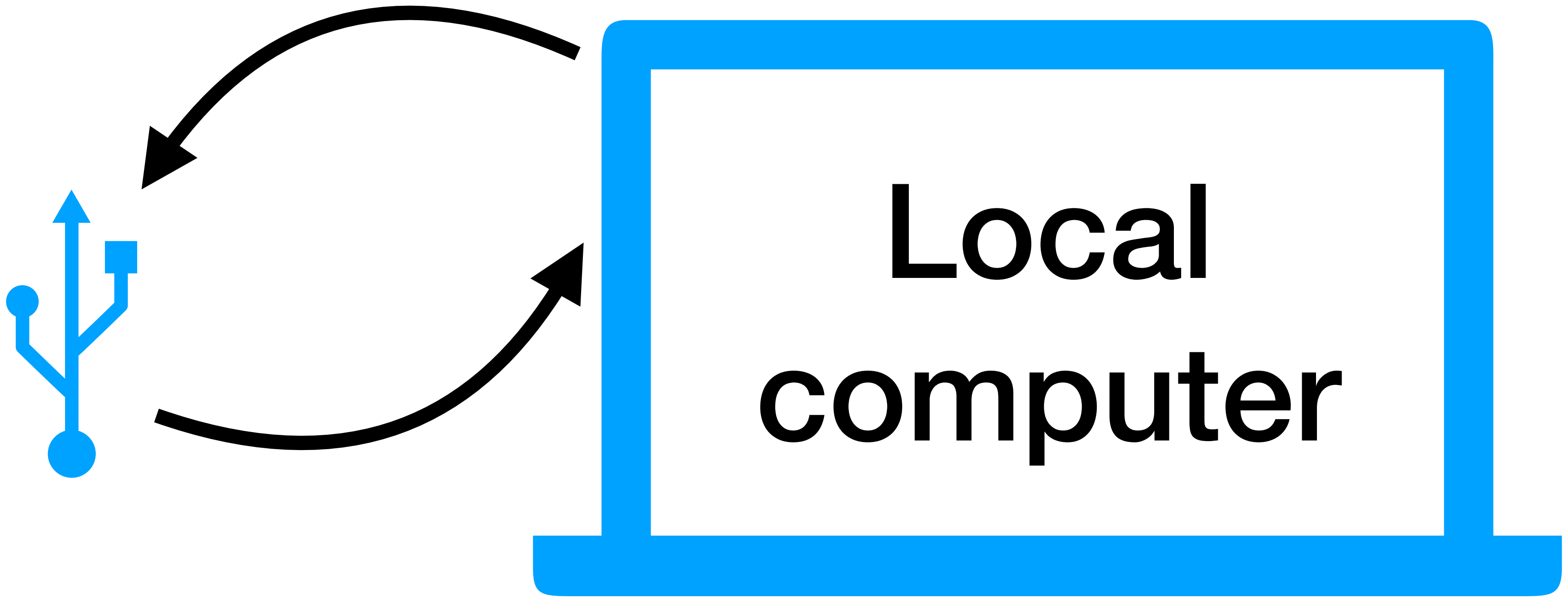


Any visible code is good code

~~All publicity is good publicity.~~

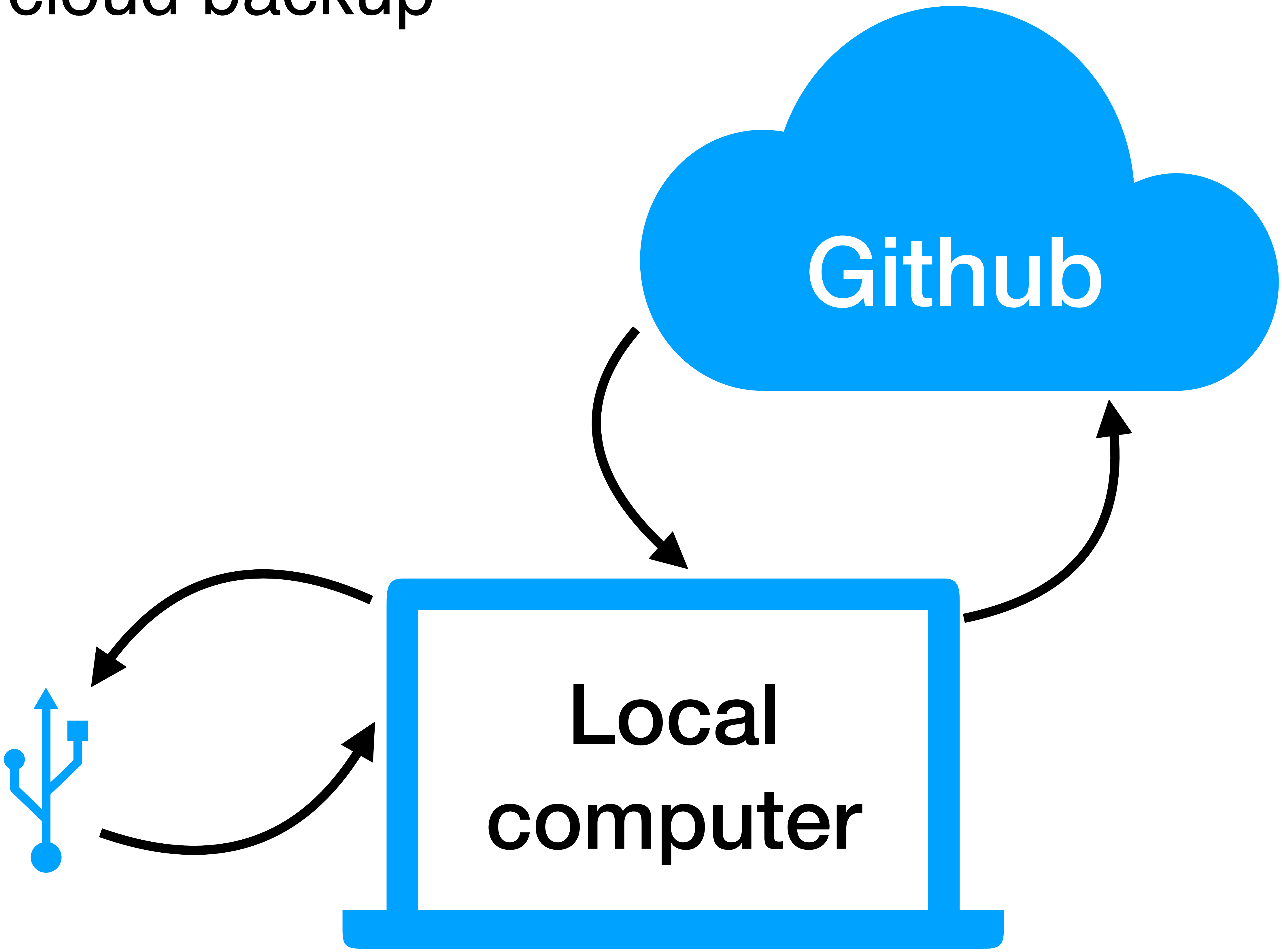
(Proverb)

Traditional programming occurs on your own machine and (hopefully) a backup



Local computer

At the most basic level, git/hub is just a cloud backup



Github

Local computer

Getting started on GitHub alone

http://happygitwithr.com/

Getting started on GitHub alone

1. Install Git on your machine

http://happygitwithr.com/

Getting started on GitHub alone

1. Install Git on your machine
2. Sign up for GitHub

git + GitHub

http://happygitwithr.com/

Getting started on GitHub alone

1. Install Git on your machine
2. Sign up for GitHub
3. Create a repository for existing or new project

git + GitHub

http://happygitwithr.com/

Getting started on GitHub alone

1. Install Git on your machine
2. Sign up for GitHub
3. Create a repository for existing or new project
4. Integrate GitHub backups regularly in your workflow

http://happygitwithr.com/

Github and RStudio

SEE: http://happygitwithr.com/

The 5 commandments

1. Organize
2. Script
3. Functions
4. Version Control
5. **Automate**

Running analysis should be easy (ish)

- Karl advocates for a makefile, that with one command should run the whole analysis
- Complex projects can make this difficult, so I would say a step-by-step guide in the README is sufficient.

Step-by-step README

Next Steps

- Publish you R package alongside the paper
- Package version documentation (packrat)
- Have friends try and run your code! Open Coding Hour :D