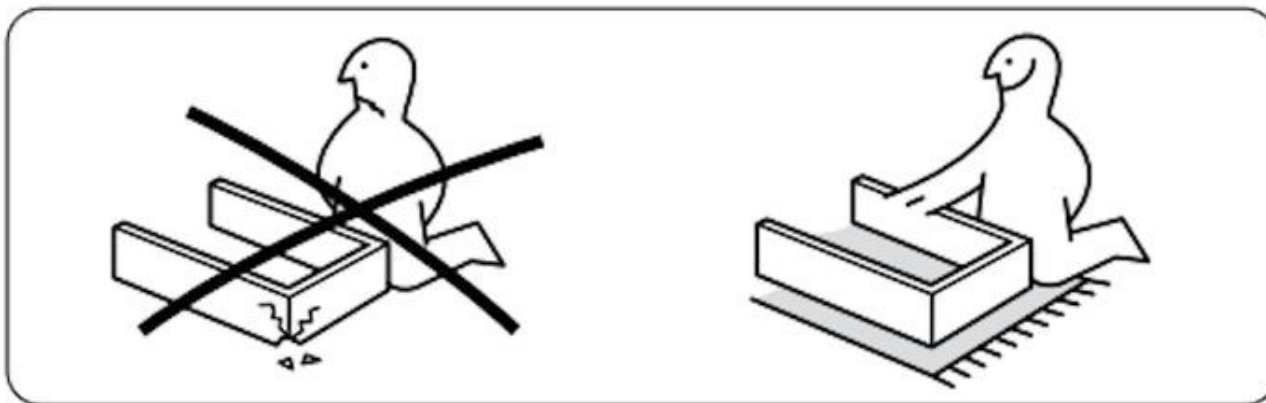
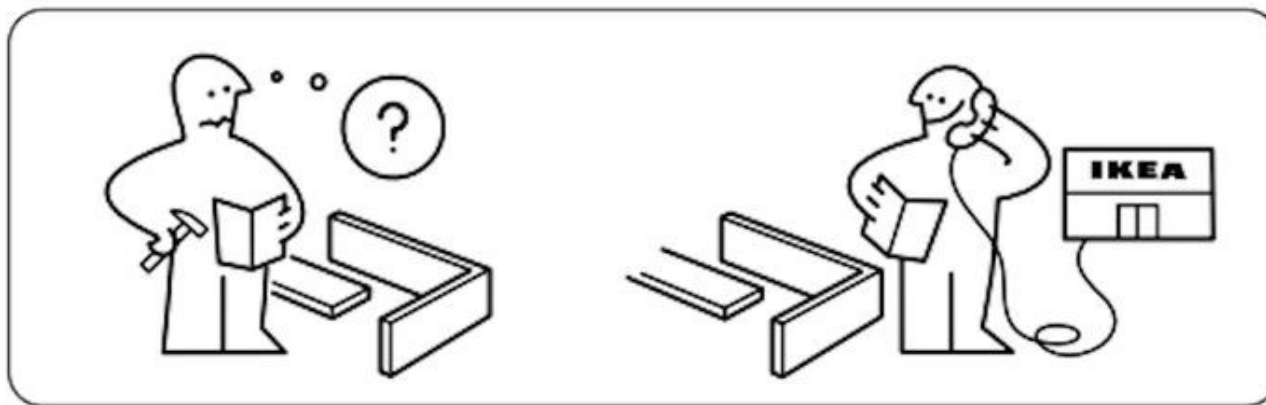
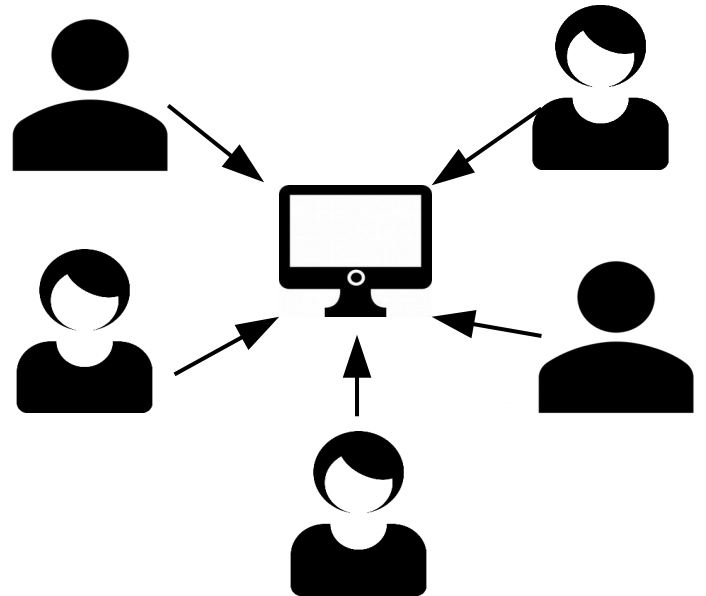
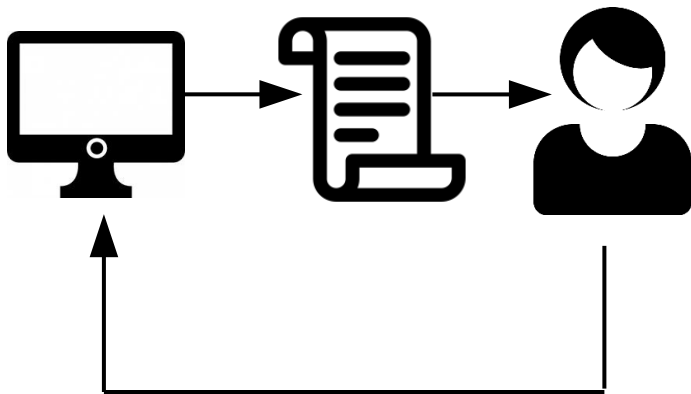


Reproducible research

Guido España

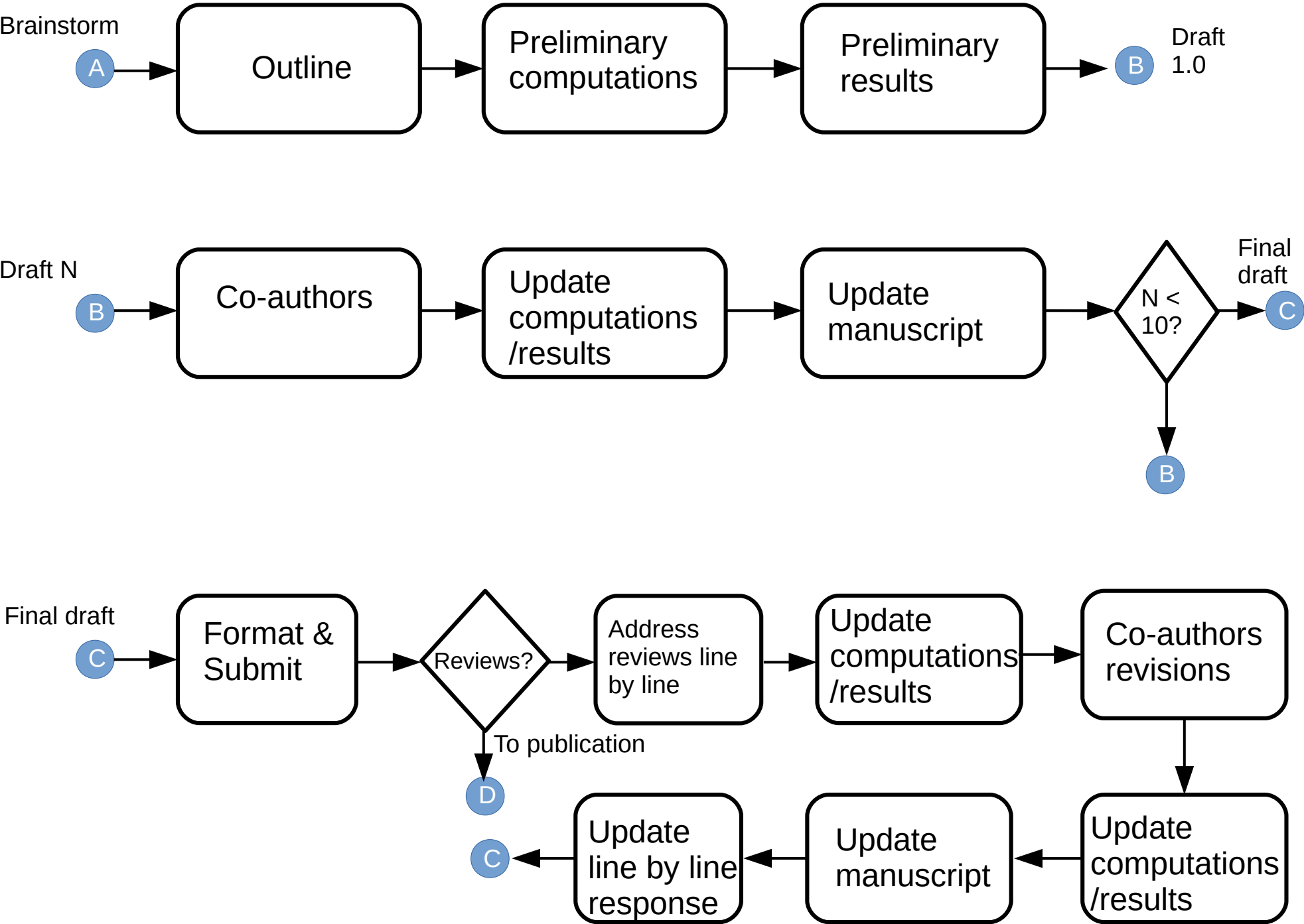


Reproducible research from two perspectives



- For peers:
 - The data and code should be available and sufficient to recreate the findings
- For authors:
 - Authors should replicate their own findings!
 - Better work habits

Steps toward scientific publishing



Many things can go wrong at every step of the process

- Gathering data, analyzing data, presenting results
- Responding to reviewers
- Remember which scripts create which figures N times after first iteration and 2 years later

Five obstacles towards reproducible research

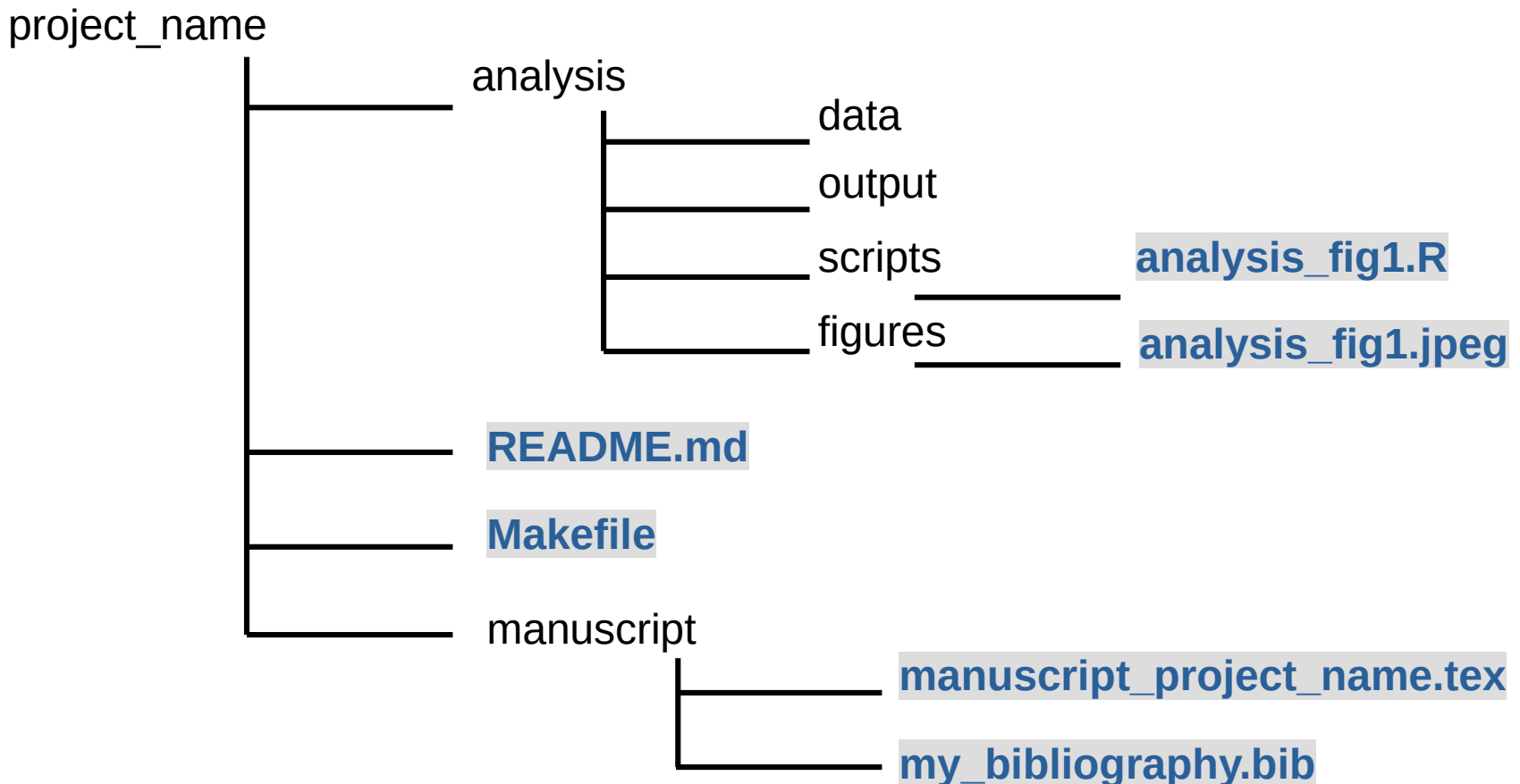
- 1) Keeping results updated in the manuscript
- 2) Previous versions of the document
- 3) Collaborating with co-authors
- 4) Responding to reviewers
- 5) Sharing with other researchers

Some tips¹

- Document everything
- Everything is a (text) file
- All files should be human readable
- Explicitly tie your files together
- Have a plan to organize, store, and make your files available

¹Gandrud C. Reproducible research with R and R studio. Chapman and Hall/CRC; 2016 Jul 6.

File management



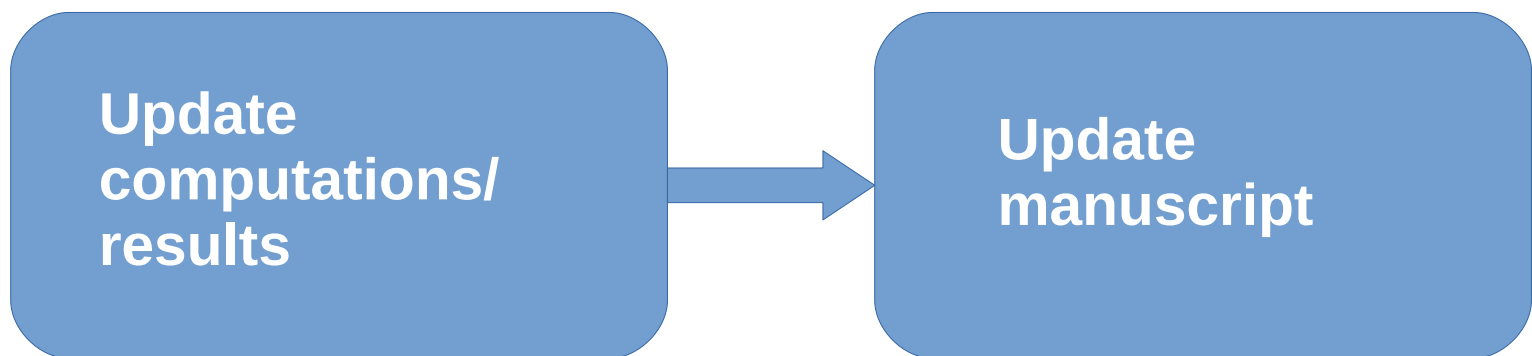
- File conventions
 - Don't use spaces (or strange characters)
 - Use relative paths
- analysis_fig1.R sounds good, but what if that figure # changes?

LaTeX example

See the GitHub repository of this demo:

https://github.com/confunguido/latex_demo_rep_res

Issue 1. Keeping results updated in the manuscript



- For tables in R: Xtable, kable, texreg
- For figures: Makefiles

Knit files with R-Makefiles and GNU-Makefiles

```
%% Makefile.R
setwd("analysis/data/")

source(".R")

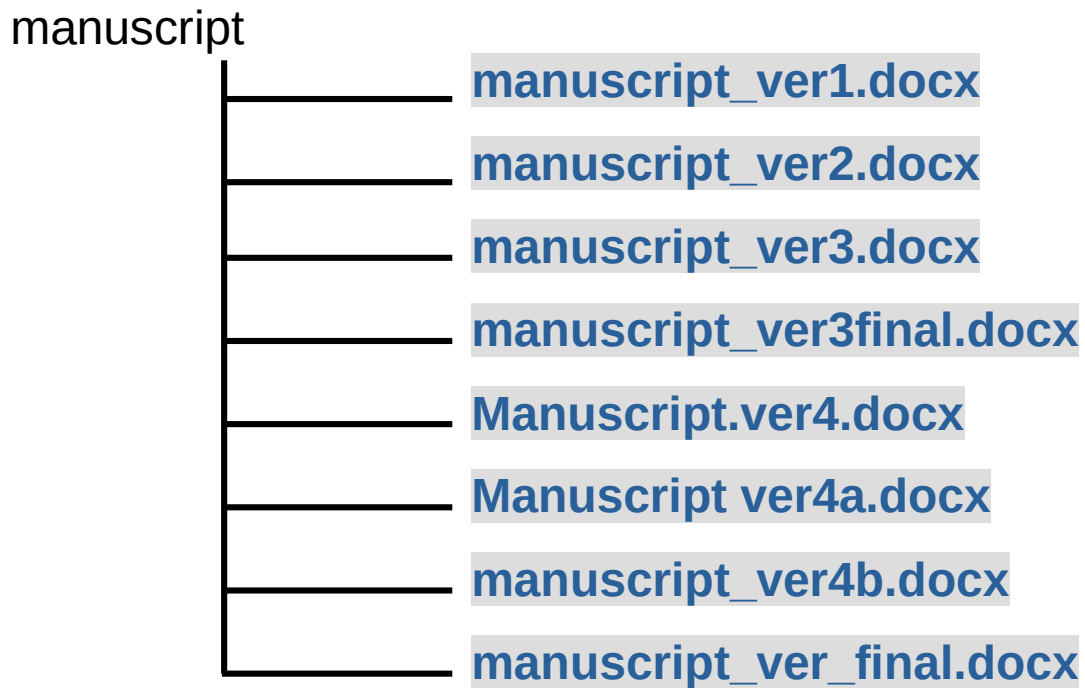
## Latex compiler

## Convert to Docx using pandoc
```

```
%% GNU-Makefile

TARGET: PREREQUISITE...
    RECIPE
    ...
    ...
```

Issue 2. Keeping track of old versions



- Too many files!!
- Difficult to understand different versions
- Version control is a better idea

Version control: Git

```
% mkdir ~/exampleProject  
% git init  
% echo test > README.md  
% git add .  
% git commit -a -m "My very first commit for this great  
research project"
```

```
% git tag  
% git tag -a v0.1 -m "Initial draft"  
% git checkout v0.1  
% git checkout v0.1 -b v0.1Branch
```

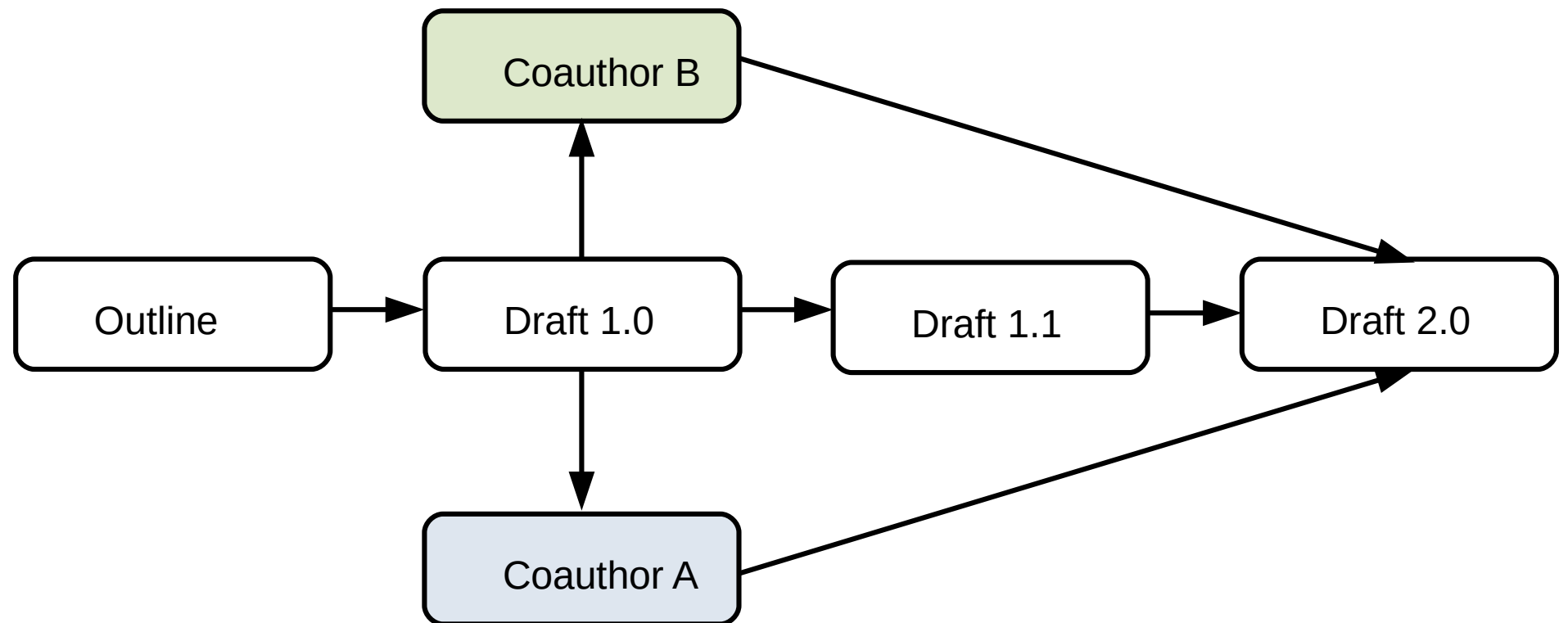
```
% git remote add origin https://  
% git push -u origin master  
% git pull
```

Git + LaTeX



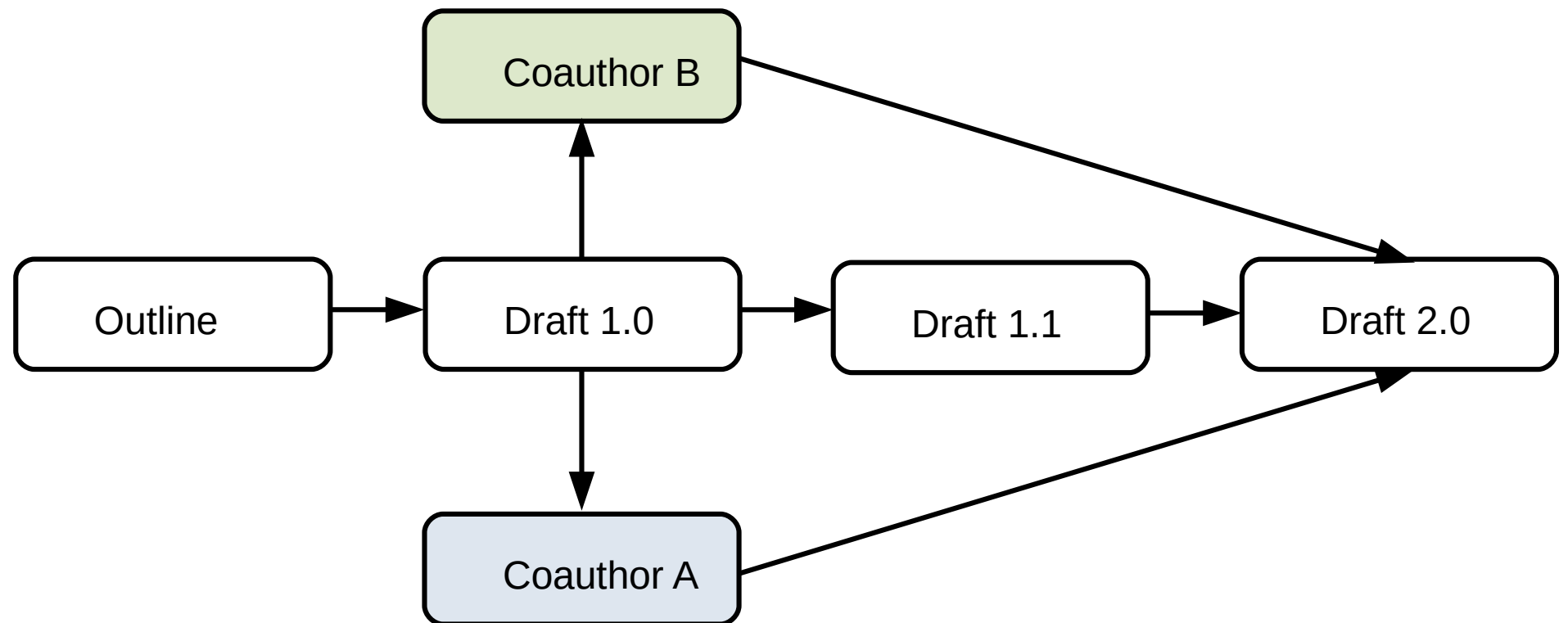
- Keep track of manuscripts, reports, etc
- Keep track of code
- Collaborate & see changes (like word track changes)

Issue 3: Dealing with co-authors revisions



- Works better if the co-authors uses latex, otherwise → pandoc
- `Git tag, git checkout -b bname tname`
- `Git merge --no-commit`
- Use `latexdiff` to see the PDF with differences

Issue 3: Dealing with co-authors revisions



- Drawbacks:
 - Git uses lines and no paragraphs
 - Not as nice GUI as MS Word

Issue 4: Dealing with journal reviews

- Example of a comment from reviewer:

“This was a beautiful work and I could only come up with a couple of comments”

- Figure 2. The authors could extend the cost of the vaccine up to 300 USD
 - Line 18. Who are the others?
 - Please add information of how the PE9 was calculated
- LaTeX:
 - `\usepackage{lineno} \linenumbers`
 - `\lineref` and `\linelabel` are useful commands
 - Run Makefile before submission!!

RStudio + knitr

Sweave:

```
<< >>=
```

```
  insert your code here
```

```
@
```

Rmarkdown:

```
``R
```

```
  Insert your code here :)
```

```
```
```

See the GitHub repository of this demo:

[https://github.com/confunguido/knitr\\_demo\\_rep\\_res](https://github.com/confunguido/knitr_demo_rep_res)

# Emacs + org-mode

```
#+BEGIN_SRC R
 plot(x,y)
#+END_SRC
```

```
#+BEGIN_SRC python
 plt.plot(x,y)
#+END_SRC
```

```
* Section header 1
** Section header 2
*** Section header 3
```

See the GitHub repository of this demo:

[https://github.com/confunguido/orgmode\\_demo](https://github.com/confunguido/orgmode_demo)

# Some other options for literate programming

- Jupyter notebooks
- Google colab (based on jupyter notebooks)
- e-life approach

# Issue 5: Sharing with other researchers

- GitHub, GitLab, BitBucket
- Git pull, git merge, git push

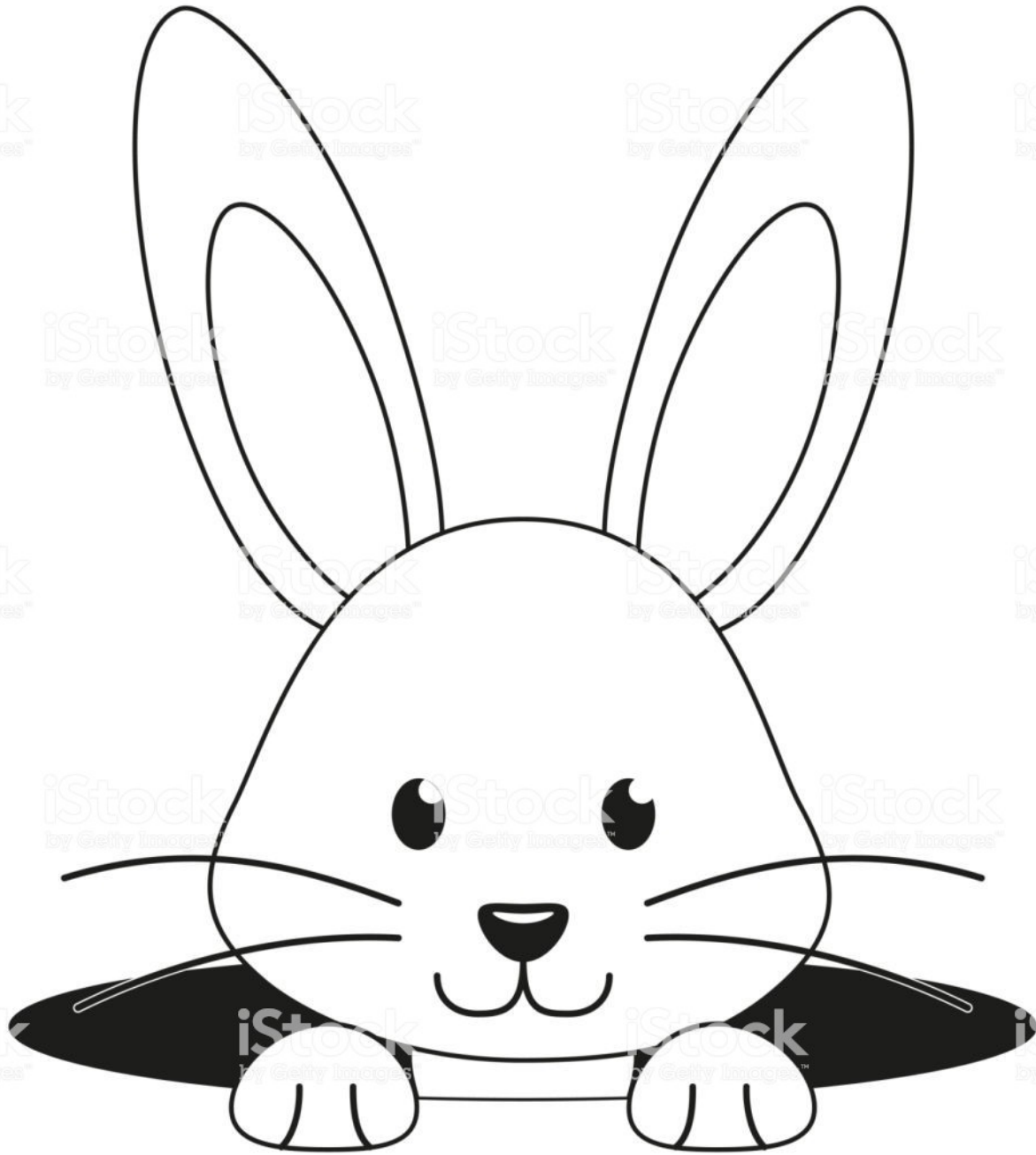


- If we have good work habits, it should be straightforward

# Summary

- Use good name files and folder structure
- Use version control
- Connect all your files together

# Warning!



Don't go down the rabbit hole