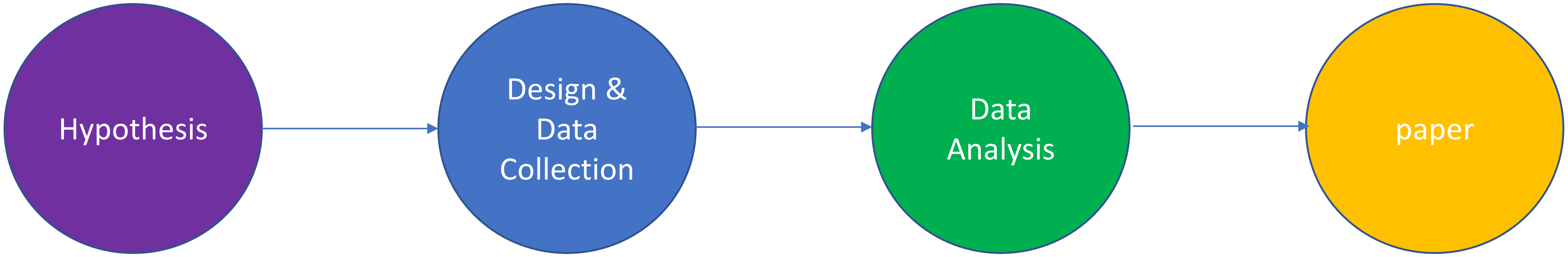
Workflows & Tackling Problems

5 October 2017

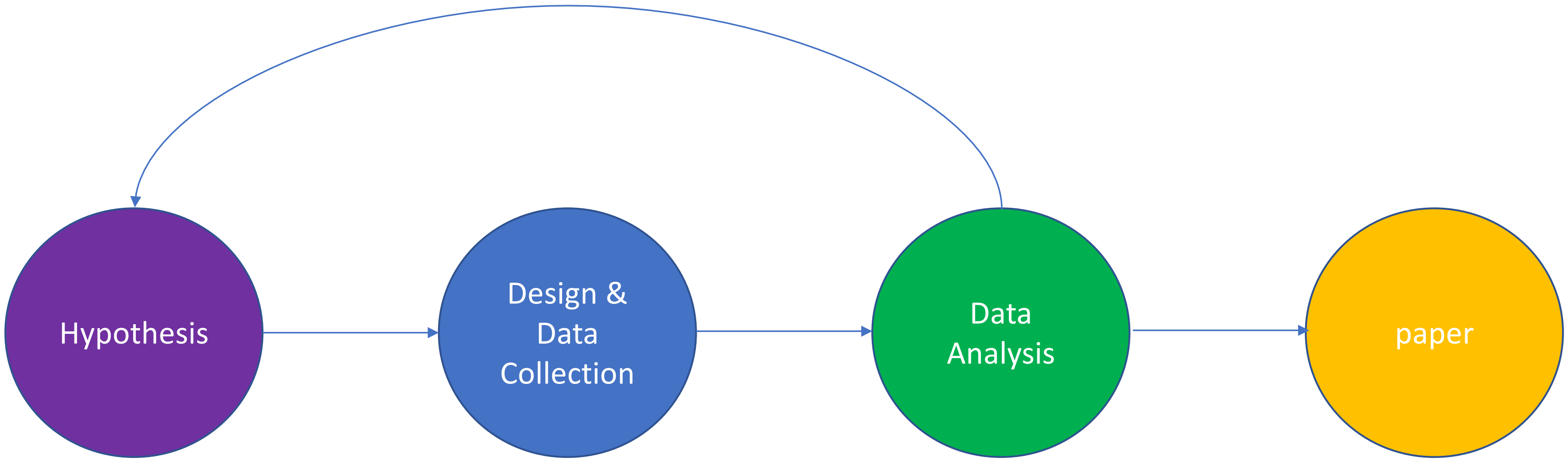
N.G. Swenson

BIOL709B

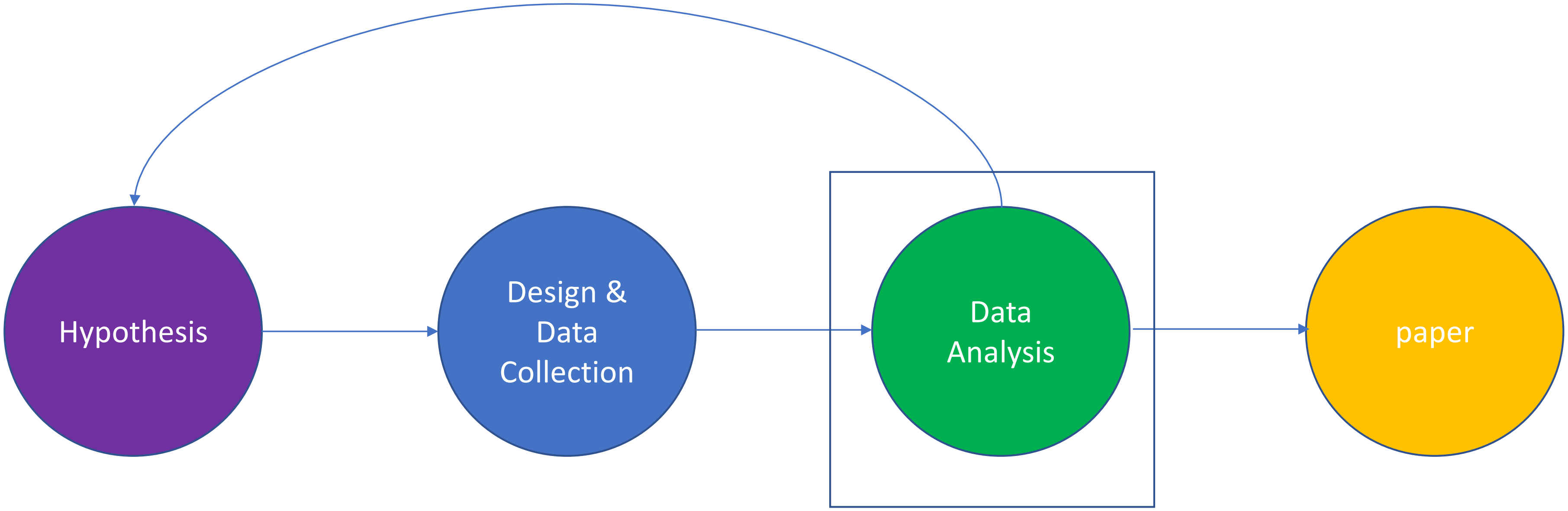
What is it? Is this a workflow?



What is it? Is this a workflow?



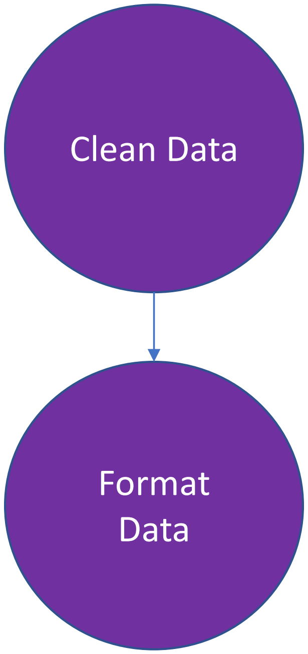
A lot goes into each of these steps



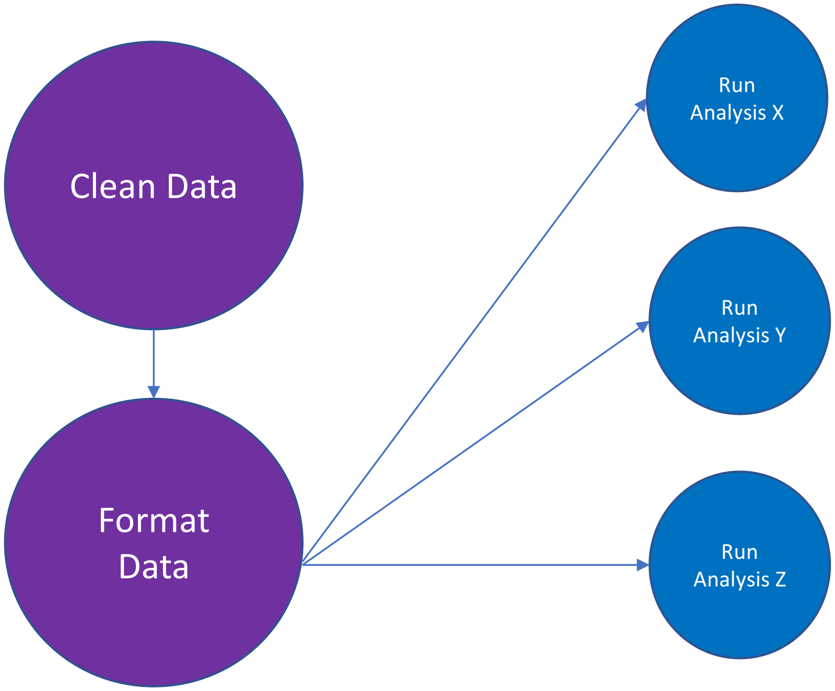
Data analysis workflow?

Clean Data

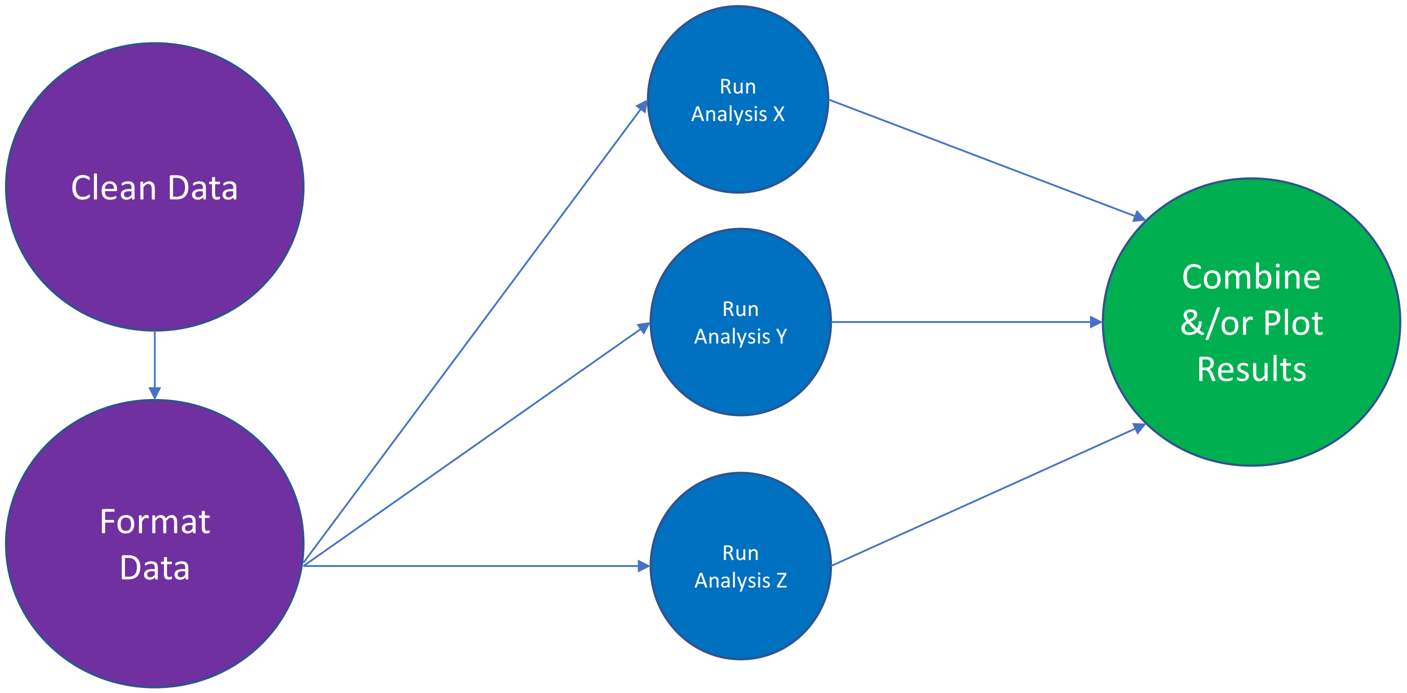
Data analysis workflow?



Data analysis workflow?



Data analysis workflow?



Clean Data

Format Data

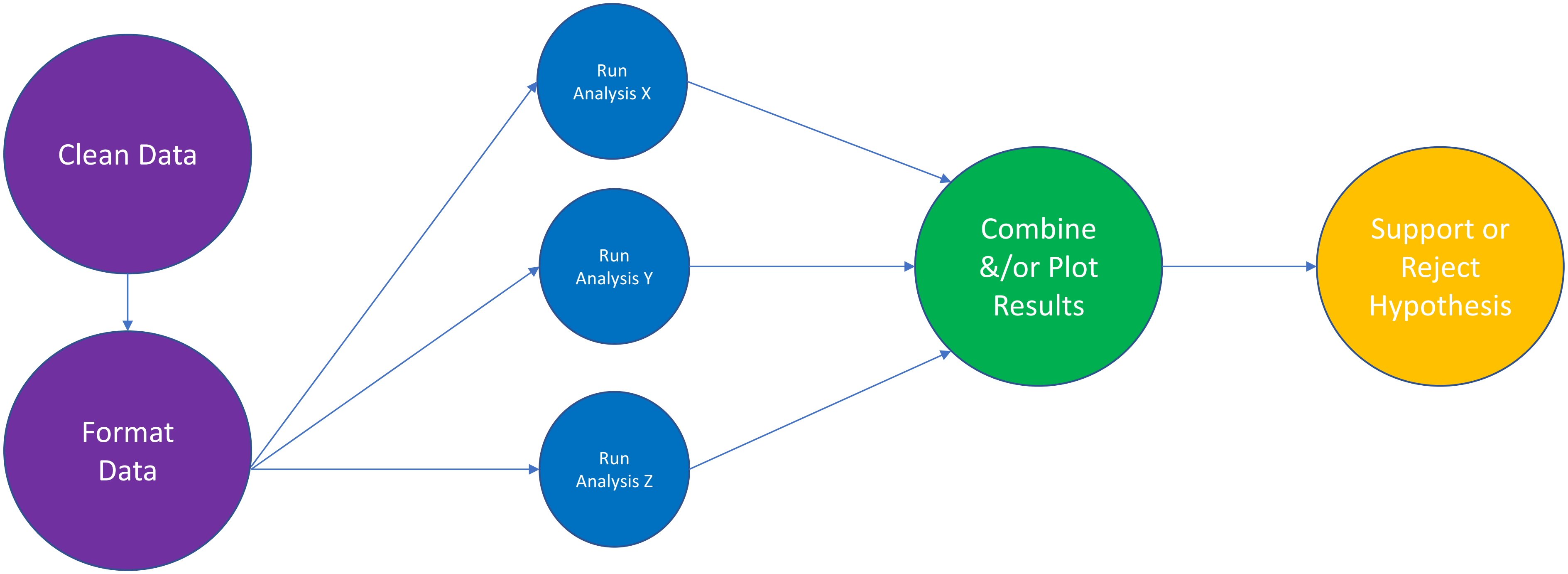
Run Analysis X

Run Analysis Y

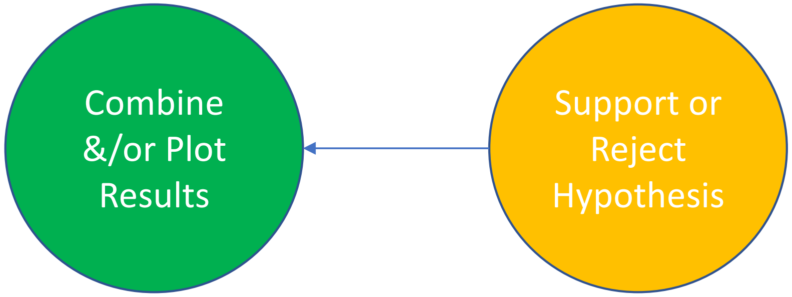
Run Analysis Z

Combine &/or Plot Results

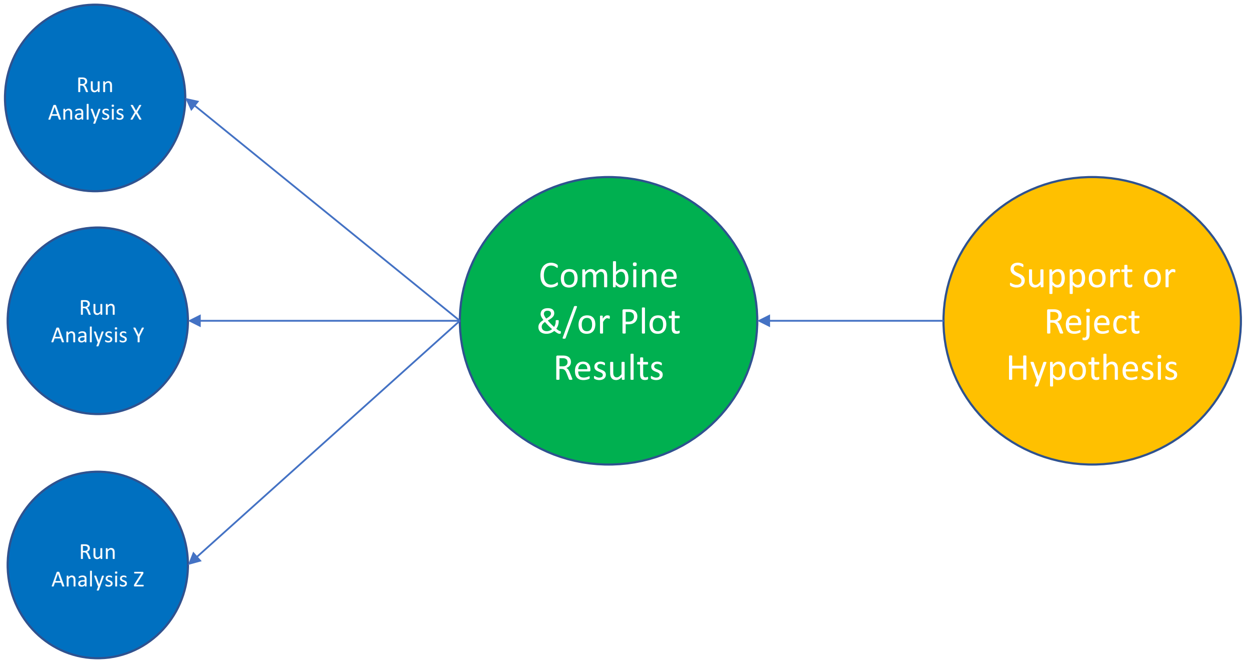
Data analysis workflow?



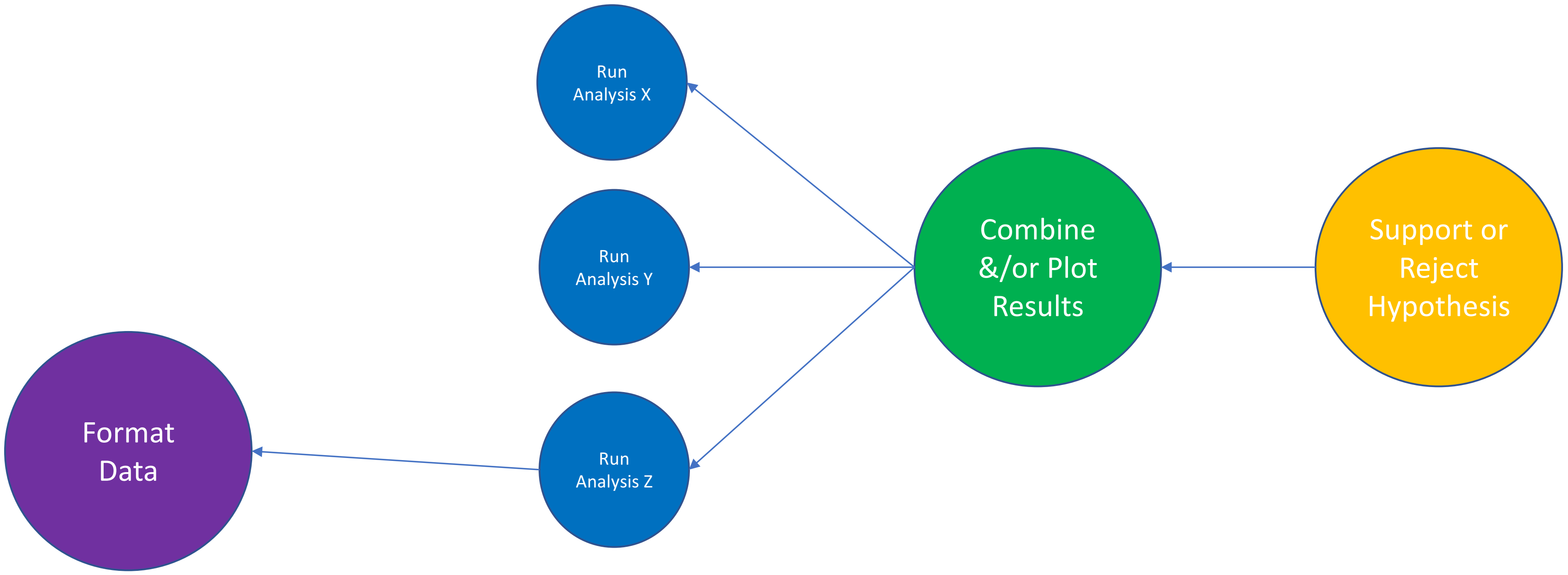
Data analysis workflow?



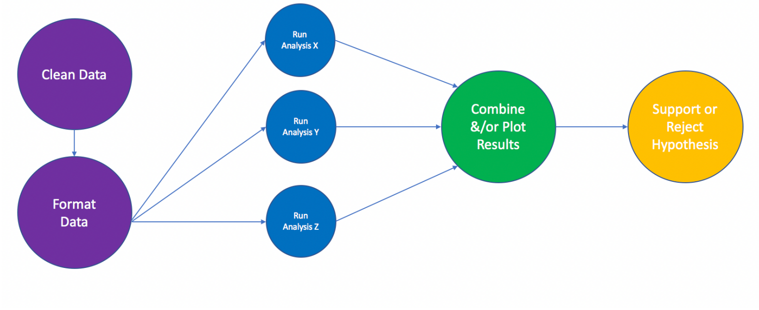
Data analysis workflow?

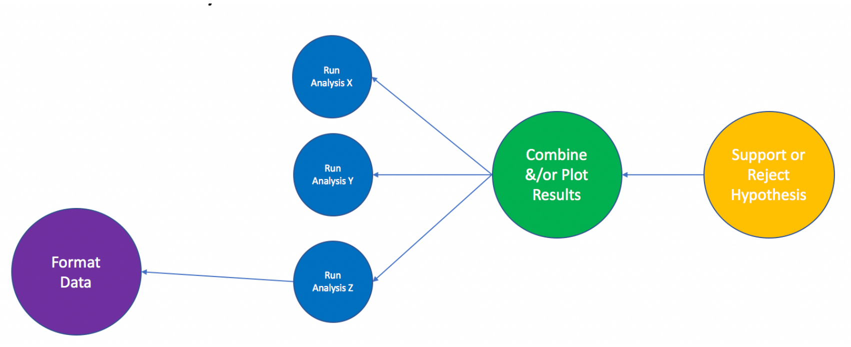


Data analysis workflow?

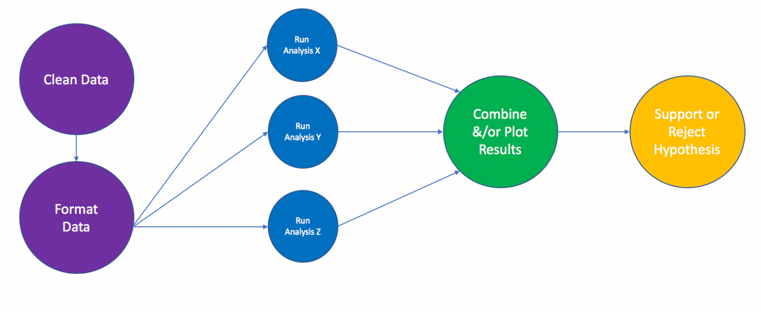


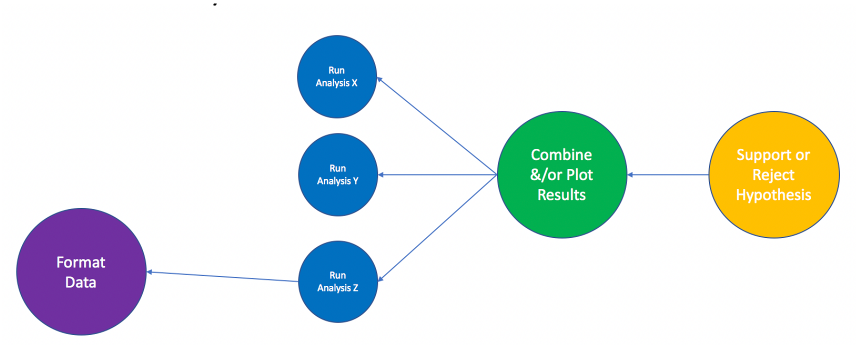
Which workflow is operationally how you would (or do) do things?





What is missing here? Anything important?





What is missing here? Anything important?

- Documentation?
- Data Storage (including backups)?
- Detail (what is actually described)?

Why workflows?

- Replication of your work
  - Often for future self
  - Replication is a fundamental component of good science
  - Time savings
    - Finding and fixing errors
    - Adjustments to approach are easier
- Collaboration
- Dissemination of knowledge

Obvious and hidden work

- What obvious/explicit steps have you taken in your work?
  - May be easily explained, described and documented

- What is not obvious or is implicit or all together hidden in your work?
  - May be steps that you think are obvious that are actually not clear to anyone else in the world (including your future self)
  - How does this limit reproduction of your work?

How many decisions were made to get to your endpoint?

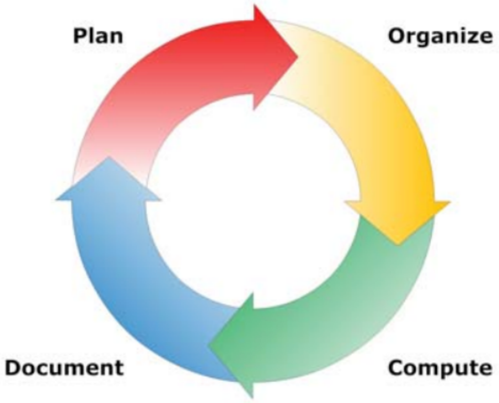
- 10 binary decisions leads to 1,024 possible outcomes. These decisions may seem minor/non-important or obvious. Often they are not.

Optimal Criteria for a Workflow (via Scott Long - IU)

- Accuracy
- Efficiency (careful vs quick)
- Standardization (techniques/code/steps)
- Automation (reduce typing or re-writing code)
- Simplicity (less complicated = fewer mistakes?)
- Usability (how you work)
- Scalability (useful for different projects?)

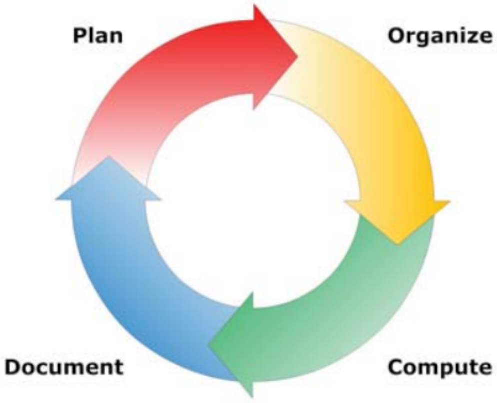
Workflow Steps (via Scott Long)

- Project Idea
- Prepare data
  - Can take up to 90% of your time in workflow
- Carry out analyses
- Presentation of results
- Protection and archiving of files

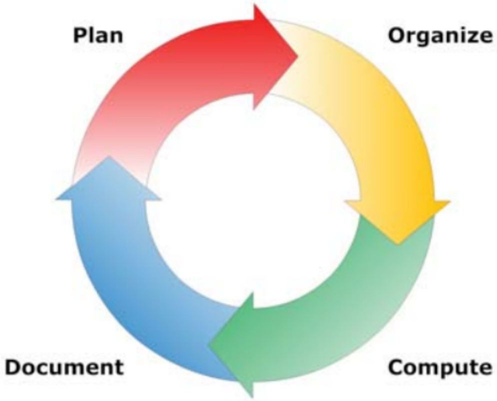


Planning and Organizing

- General goals and firm deadlines
  - Any division of labor?
  - Anticipated output (not the actual result, but what you want to look into) and analyses (how)
- Organizing the project
  - Reduce duplication
  - Standardize
    - tmp.csv
    - Data.csv
    - Wisconsin.trait.data.2015.3.10.csv
    - Swenson.trait.MS.2017.10.5.docx
    - Swenson.trait.MS.2017.10.5.KP.docx

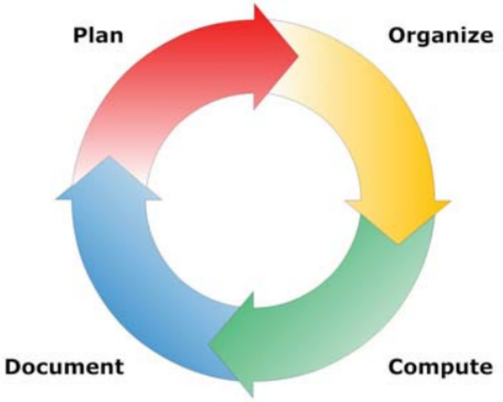


Compute



- (Reusable) modular computation
- Saved version of code
- No saving of “duplicate” versions of data. Conduct your workflow/computation in such a way that you can start from ‘scratch’ each time
  - Save derived products (of course)
- Look at your data and output from many angles along the way

Documenting



- Annoying to almost everyone
  - Can we automate it somehow?
  - Way easier (and better) in the moment than later
- Future self and others will find it extremely useful
  - No documentation on how you did it = much harder (or impossible) to do it again
  - Lab notebook