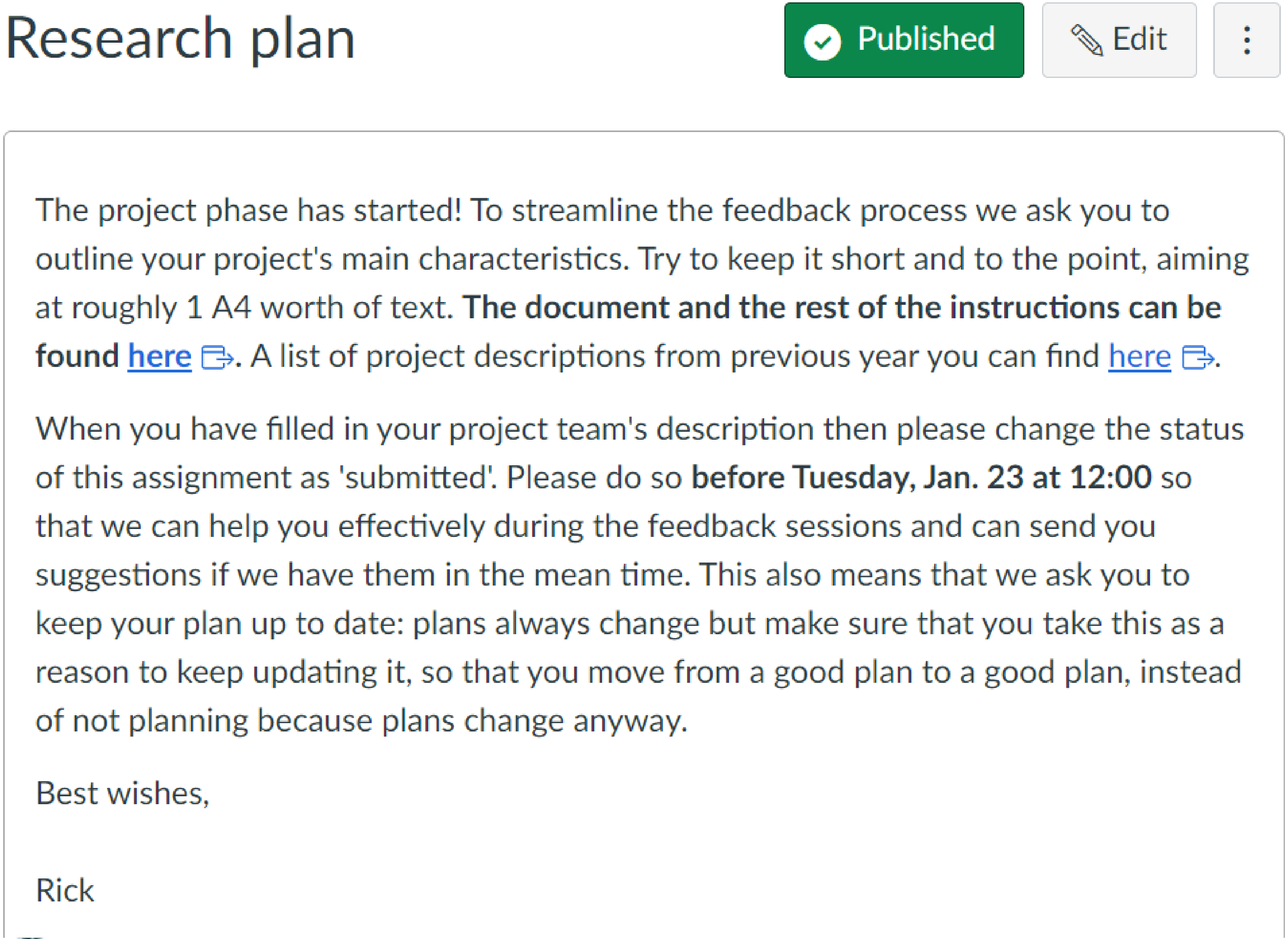
Project phase

Some pointers

Submit project plan



Research plan

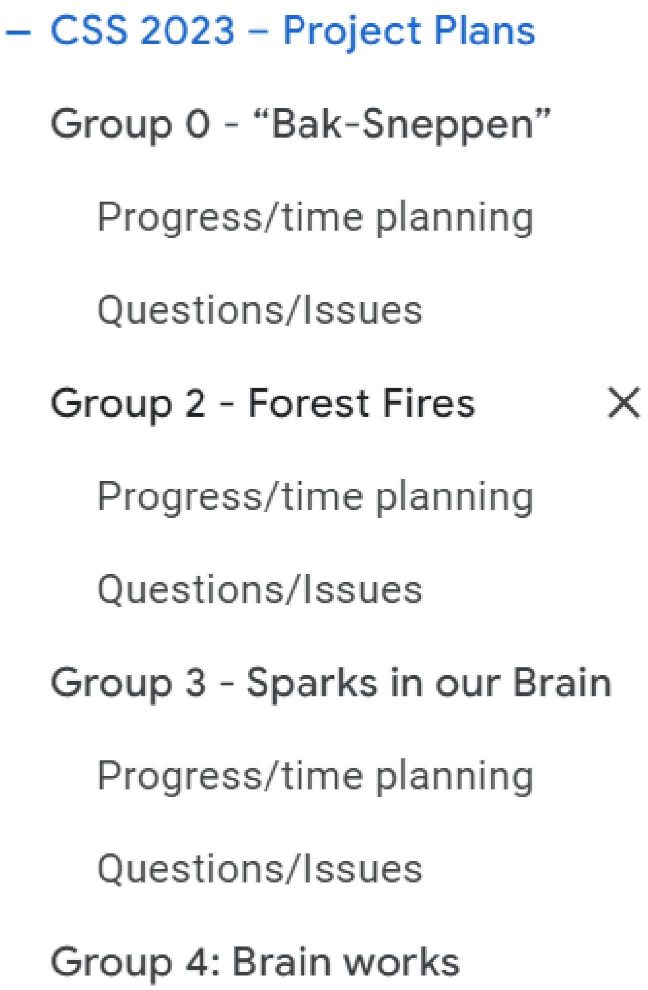
The project phase has started! To streamline the feedback process we ask you to outline your project's main characteristics. Try to keep it short and to the point, aiming at roughly 1 A4 worth of text. **The document and the rest of the instructions can be found here**. A list of project descriptions from previous year you can find here.

When you have filled in your project team's description then please change the status of this assignment as 'submitted'. Please do so **before Tuesday, Jan. 23 at 12:00** so that we can help you effectively during the feedback sessions and can send you suggestions if we have them in the mean time. This also means that we ask you to keep your plan up to date: plans always change but make sure that you take this as a reason to keep updating it, so that you move from a good plan to a good plan, instead of not planning because plans change anyway.

Best wishes,

Rick

Previous year's projects



- CSS 2023 – Project Plans
  - Group 0 - “Bak-Sneppen”
    - Progress/time planning
    - Questions/Issues
  - Group 2 - Forest Fires
    - Progress/time planning
    - Questions/Issues
  - Group 3 - Sparks in our Brain
    - Progress/time planning
    - Questions/Issues
  - Group 4: Brain works

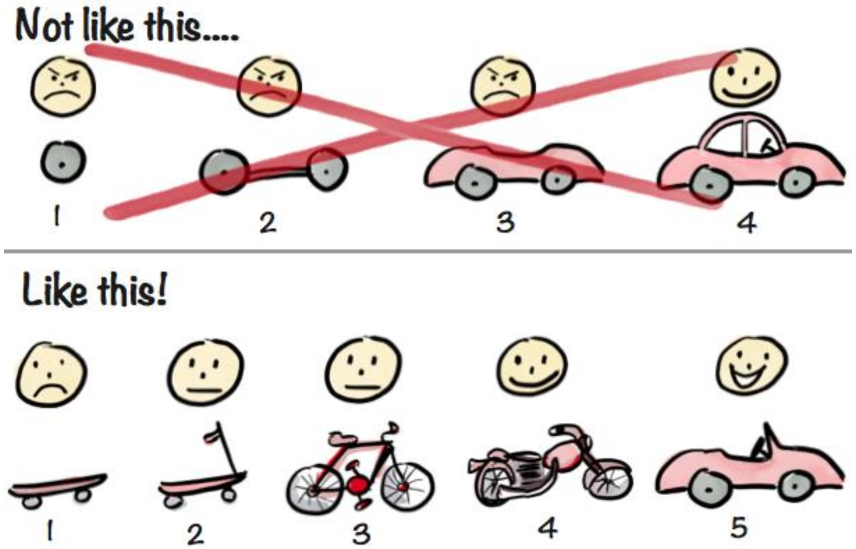
Link can be found in the Research Plan assignment.

Submit project plan

- Background/Motivation
- Research question(s)
  - And/or hypotheses
  - Make them testable/falsifiable!
- Which model?
- Which emergent phenomenon?
- If applicable: which data set?
- 1-3 references

Some tips

1. Organize yourself. Make a tasks list and categorize them as 'must have' or 'nice to have'
   1. Consider Trello, ClickUp, MS Planner, …
2. Follow the MVP approach (Minimum Viable Product)



Source figure: https://blog.crisp.se/2016/01/25/henrikkniberg/making-sense-of-mvp

Development

- **You MUST use Git!** (or SVN, …)
  - GitHub, Bitbucket, …
  - Please make a public repo whenever possible
- I will check if every team member made *at least* one commit
  - I will look at number of surviving lines etc.
- Each member must be able to verbally explain each design choice, each function
- Do not specialize the roles in the teams *too much*
- You must work in Python

Development

- If you use Notebooks…
  - Git merge conflict:
    - Work on separate notebooks, OR:
    - Work in the same notebook but in a different section
    - Before you commit: remove all outputs
  - Use Markdown to clearly delineate the blocks of code
    - re-order blocks to make the storyline clear
  - Use modules (.py) to store functions
    - Keep the notebook for the logic and showing outputs, but not as a storehouse for function definitions

Development

- If not using notebooks, then use ‘argparse’ or similar
- Use inline comments for code blocks
- Give each function a ‘docstring’ (at least 1 line)
- You must have a **README.md** and **requirements.txt**
- Put data in a ‘data’ folder; results in a ‘results’ folder, ...
  - Data too large? Consider, e.g., ‘wget’ it if missing (‘os.path.exists’).
  - Simulation takes longer than half an hour? Indicate the parameters/command to run a smaller simulation.
- Small bonuses (say so in the README if you have it):
  - You have roughly 5-10% ‘assert’ statements inline
  - “pytest .” works and succeeds
  - Structured as a module (“__init__.py”; e.g. “pip install .” works)
  - Documentation generated from docstrings

Rough timeline

- Today: planning, project definition
- Tomorrow morning
  - Define tasks and task dependencies
  - Initial distribution of tasks
  - Set up Git; decide on style, conventions, tools
- Thursday: a first (minimal) model is running
- Friday: some plots are generated based on simulation outputs (may still be bogus)

Rough timeline

- Next week
  - Tuesday
    - A few plots should make sense now / almost production ready
    - Start making a presentation and inserting already some background/questions etc.
    - You could still decide that someone works a few days on a stretch goal (nice to have).
  - Wednesday
    - Make decisions: which nice-to-have are we still going to pursue?
  - Thursday
    - Focus on presentation; sometimes this leads to additional results that should be generated!

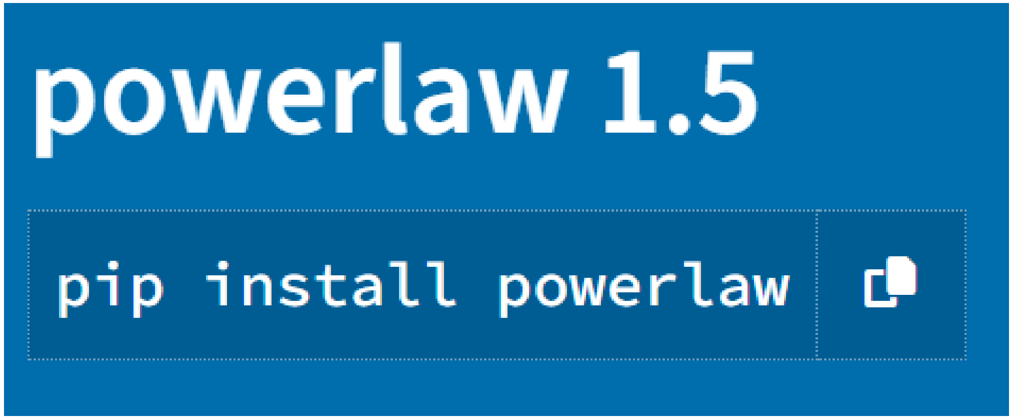
How to use these lab sessions

- Sit together anywhere
  - Can also be outside this classroom but let me know
  - Work on your project; see this as a project **work meeting** with your group
- By default I will go around and ask questions
- Or, you let me know you'd like to discuss something
- Once one of us talked to you and everything is clear to you, your group may also leave and sit somewhere else

In between

- Reach out through Canvas Discussions or through e-mail (address to me and the TA both) if very specific.
- Even better: a comment addressing one of us in the Google Doc. We should get a notification and we can remind ourselves of your project.

Testing for powerlaw?



powerlaw 1.5

```
pip install powerlaw
```

Compare with real data?

- Proceed with care!
- “Nice-to-have”

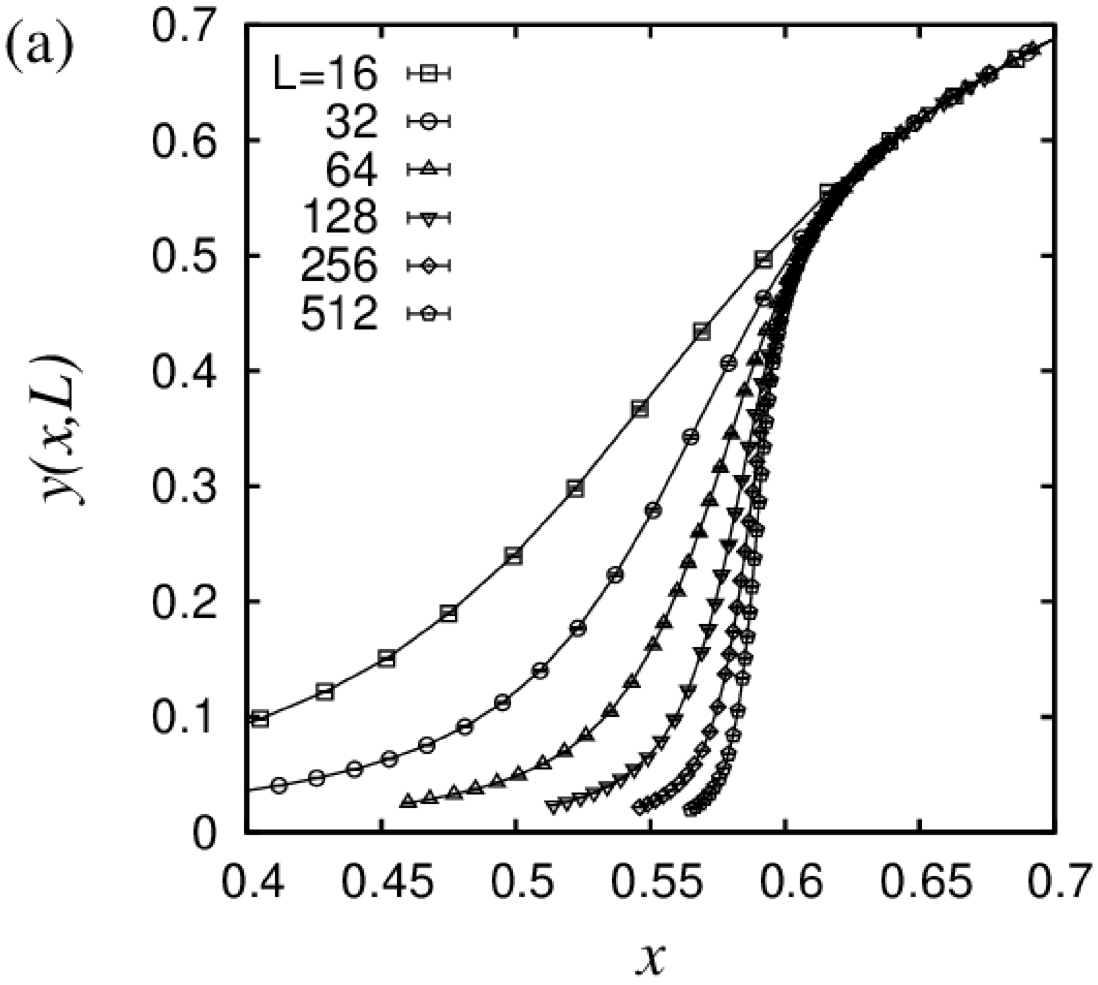
Be focused

- "…will be capable of exhibiting all kinds of complex behavior"
- "…does [the model] show emergence?"

Phase transitions (a.o.)

- Find sharp changes in a curve.
  - E.g.: transition point = peak in second-derivative (finite differencing)
- “Finite-size scaling”
  - E.g. for estimating exponents
  - Transition should get sharper and sharper
  - Extrapolate the transition point as function of system size

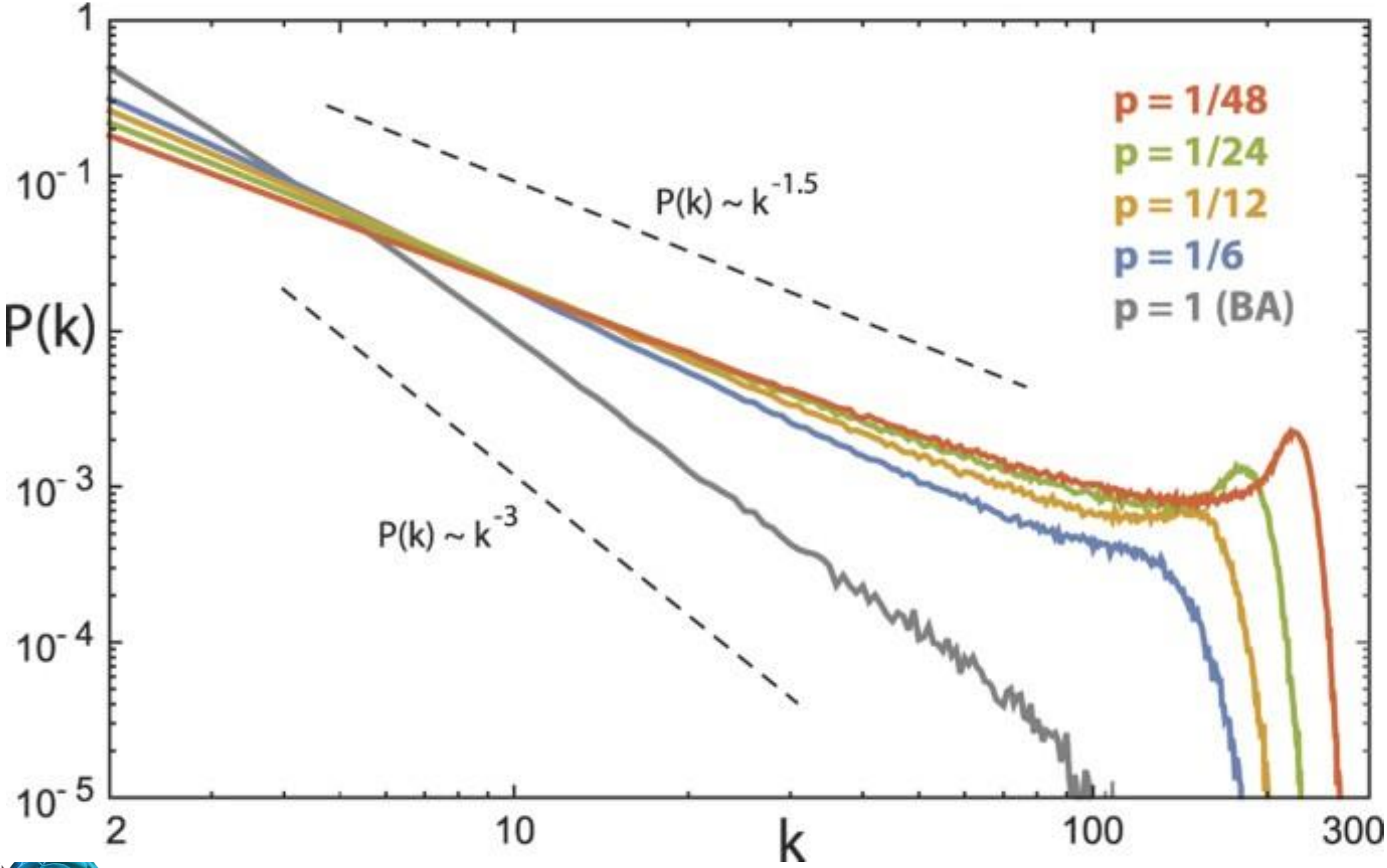
Phase transitions (a.o.)



We expect SOC

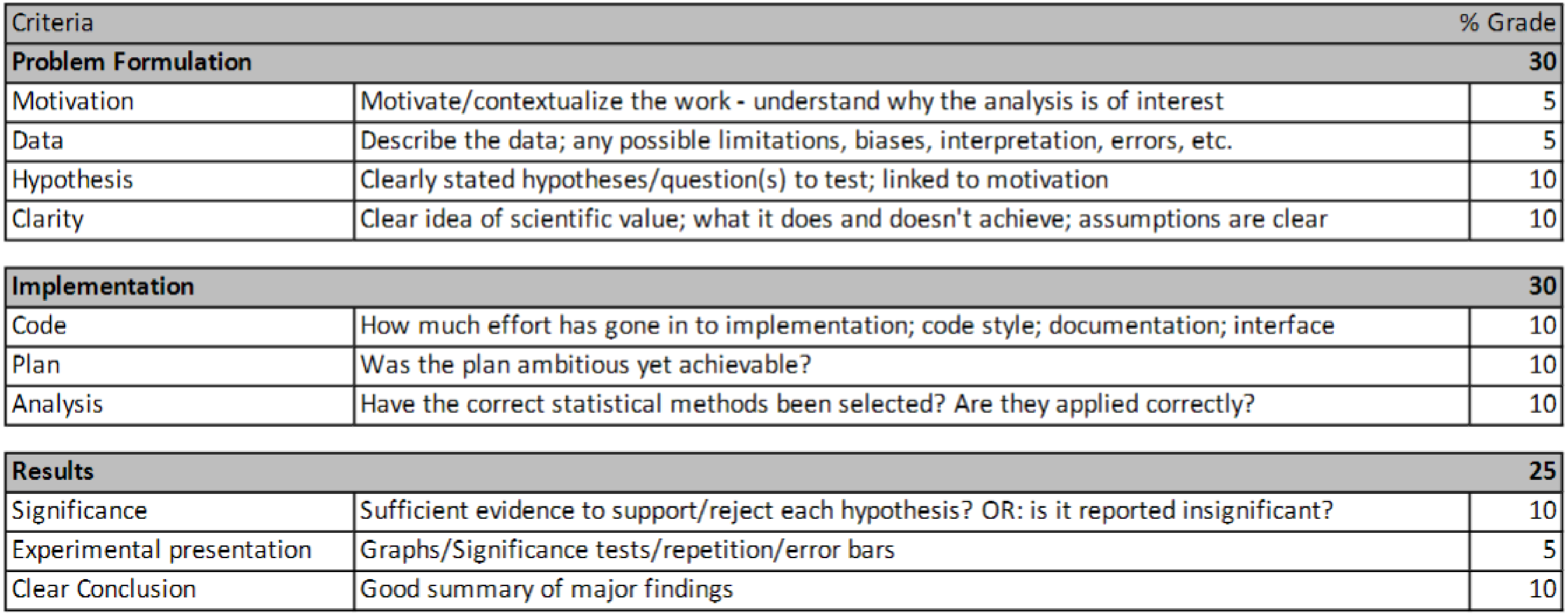
- What are the four ingredients from the lecture in your model?
- A powerlaw ‘avalanche’ distribution is not a proof but another piece of evidence that makes your claim more plausible
  - No powerlaw? Then no SOC.
- SOC is very hard to *prove* (no null-hypothesis test), but you can make it plausible.

‘Saturation’



https://www.nature.com/articles/srep12353

For the presentation



Criteria		% Grade
Problem Formulation		**30**
Motivation	Motivate/contextualize the work - understand why the analysis is of interest	5
Data	Describe the data; any possible limitations, biases, interpretation, errors, etc.	5
Hypothesis	Clearly stated hypotheses/question(s) to test; linked to motivation	10
Clarity	Clear idea of scientific value; what it does and doesn't achieve; assumptions are clear	10
Implementation		**30**
Code	How much effort has gone in to implementation; code style; documentation; interface	10
Plan	Was the plan ambitious yet achievable?	10
Analysis	Have the correct statistical methods been selected? Are they applied correctly?	10
Results		**25**
Significance	Sufficient evidence to support/reject each hypothesis? OR: is it reported insignificant?	10
Experimental presentation	Graphs/Significance tests/repetition/error bars	5
Clear Conclusion	Good summary of major findings	10