Now that the workshop has had some time to sink in I have some more practical feedback:  
- overall really good! And verbal feedback from other participants so for has been very positive  
  
Git part:  
- Good flow from basic to advanced, and really good setup so that advanced users can still get something out of it, maybe should be mentioned more that parts that are skipped can be done by advanced user if they get bored during exercises   
- Maybe a bit too much focus on rebase, commit squasing and cherry picking: these are good commands when you already know your way around commit "etiquette" (often enough and coherent goal), I think that taking time out of these and adding more practical tasks focused on add --patch and maybe even a git stash; git difftool -y stash; git commit -a; git stash pop; sequence since too large commits is super common with beginners of git  
- Collaborative distributed version control was great, this is what is needed today!  
- Automated testing was good, but maybe there is also room for just showing how to use pre-commit hooks to do automated testing for those who do not want to sign up to a service?  
  
Documentation:  
- the basics of sphinx and markdown was great!  
- I was really missing the (at least for me) most important part about sphinx: automatic documentation  
I use sphinx because i \_dont\_ have to write the contents of the .rst files myself, they are automatically compiled with things like:  
.. autosummary::  
and  
 .. automodule::   
 :members:  
 :undoc-members:  
 :show-inheritance:  
and so on! Then it is also vital to write the documentation inside the source files instead.  
You could even setup python scripts that generates auto-module .rst files automatically or use sphinx-autogen.  
  
Jupyter:  
- good introduction! I will definitely use it for tutorials of things and examples of how to use my packages that i develop