## General

Hi Radovan,  
thank you and the CodeRefinery group for a very nice workshop!

---

By the way, I am making full usage of several of the things I learnt in the course. Thanks again for the excellent job you guys did in putting this together.

---

Thank's a lot for an excellent course! I hope it was not too much  
feedback,  
but since you guys asked for it... ;) You have a nice team.

---

Thanks for the very informative CodeRefinery workshop. There was a lot to learn in it and I got introduced with platforms which were unknown to me earlier.

## Git intro

* That we ourselves did the small steps in the lectures
* stickers for help
* Well explained how git works
* Sometimes a bit fast
* I think the Git session was well organized and planned. The hands-on Git approach seem to work well combined with some discussion.
* I have no negative comments on the way lecturing was carried out
* Interesting to go beyond git clone/commit/push/pull
* Clear explanation on branches + merging
* Nothing
* Super nice explanations of all the different features and tricks of git
* None
* Very nicely structured session
* Nice speed
* Like the summary of every sub-theme-thingy
* I hope for a short recap tomorrow morning
* Good pace, easy to follow, systematic
* Nothing!!
* Covers a wide range of topics
* Instructor is using large enough fonts for the demostrations
* Too many things of importance were done during the afternoon session (late afternoon)  
  Possible solution: switch with the jupyter session
* A lot of useful information
* A LOT of useful information, this will change into plus after getting comfortable with git
* Very informative and useful
* Nice layout of the time schedule
* Very useful instructions being provided
* The necessary of using [?] repository is discussed
* A bit complicated in terms of visualizing the overall structure
* Nice to see a number of git commands
* Not much new information for me personally
* Recipe example is good for beginners
* Maybe more visualizations, could make commands easier to follow

## Jupyter

* Thanks for the tutorial. I liked the idea of a short tutorial on data analysis
* Interactive, felt I learned and will remember
* Better to use example exercises more relevant (may be hard since everyone’s research is different)
* Nice overview and good tempo
* More help when doing the exercises
* More motivations
* Nice overview of magic commands
* Use cases could be better motivated
* Good overview
* Understandable
* Too short
* Could be more detailed examples
* Really liked the in-depth introduction (especially “magic”+shortcuts)
* Very happy that it was a “hands-on” course
* Would like a bit more time to experiment
* Accelerating python, interfacing different languages
* Almost nothing, but maybe discuss weaknesses more (e.g. not very good for large development)
* Nice instructor, helpful
* Talked very fast
* Too much information for a new user  
  Possible solution: 1. Skip group work and take more time for the intro, 2. Work on a few examples, but do this thoroughly
* Interesting to know that it exists and what is possible
* Too fast. Limits? (in terms of size of project and data sizes)
* Very good examples, not straightforward and not too complicated
* Too fast! Was sometimes difficult to catch up and absorb everything
* Interesting introduction to a program i wouldn’t have discovered on my own
* Not sure if I will use it myself, as I don’t use that much python, but I might discover uses
* Goo exercises
* More information on usage areas
* I think the session was fine with hands-on programming. I appreciate the way it was presented and organized even though for several reasons the presented tool is not something I plan to use myself
* Very nice introduction
* Seems like a useful tool for testing at small scale
* Not much support with Windows OS
* Very interesting way to work
* Maybe too much information at once
* Good presentation, would love to have more time to play around with Jupyter functionality/widgets
* A bit too fast in the beginning
* I want a list of keyboard shortcuts
* Good examples
* Too fast first
* Talked slower after comment about the speed

## Collaborative Git

* Nice explanation of forking and pull requests, I always wondered about those
* Nothing really!
* Learned a lot of Git functionalities that I had never used before
* Interesting to learn about different setups for collaboration
* Not too fast, not too slow
* Again, great speed and easy to follow session. I really liked it. This is super useful stuff!
* Hope the material/exercises/explanations that are online now, stay there or are somehow downloadable for later reference
* Going through the basics in a systematic manner and adding details I couldn’t be bothered to read from Google
* Probably a bit too basic for my level. Could have gone faster
* Interesting topics
* Maybe more fun to go more in depth on some of the topics
* Very nice, slow enough for me to follow. Very useful
* Very good visual explanation
* Good timing
* Good interactive session
* Mention that git bisect takes you automatically to a new commit
* Squash example would be helpful e.g. how to clean up commits such that they are acceptable for a pull request
* Very thorough and easy to follow

## IDE/automated testing

* Well explained about testing
* A bit confusing what to do with pycharm (visual interface more difficult to follow in lecture)
* Very helpful teacher

Improvement: less lecturing about general testing (>30 min) and more time for the Travis CI implementation + setup

* Very confusing
* Too fast
* Seems super useful, ...
* but the session was a bit confusing

Hope for a workflow repeat/recap/schematic for reference on this topic. Maybe a bulletpoint list on what travis/coveralls is etc.

* Discovered very useful tools
* Good introduction on the “whys” and “hows” of testing
* Pycharm use with git+github
* Would have been cool to write a few tests ourselves

I may have missed that Travis is not a specific python tool so it seemed less relevant at first. Finally I think it should also work with Matlab, right?

* A bit fast, a bit too fast. Would appreciate more practicals on pycharm
* Pycharm was well introduced and will definitely be used in the future
* Good with lengthy discussion in the beginning, with separated presentations and tutorials
* Too little time for tutorial
* Loved Travis and Coveralls
* Too much time on python specifics / pycharm for us non python users
* Lots of great content on testing and IDEs
* Would like these to be covered before advanced git functionalities
* Lots of interesting and important information. Got to try things I probably wouldn’t have tried by myself
* Lots of information to cover, so no time to try for myself and follow the presentation and the same time. I constantly tried to do something and when I looked up the talk was about something else
* Very informative in terms of usage
* A lot more concentration on software development and not on scientific testing of algorithms
* Nice introduction to pycharm + Travis!
* Maybe give an example of another language and how to actually write good test
* IDE not so relevant for me...
* Otherwise nice to know the [unreadable word]
* Nice presentation of tools I did not know existed
* Examples shold be more structured and pedagogic, was challenging to follow

## Mixed martial arts

* Very well prepared session
* I was able to interface stuff
* I have no clue about what I have done and why. I basically copied files and commented things in/out but I do not know why / the meaning of the lines
* Too fast

On pybind11:

* I like Bjørn a lot, he is very helpful
* We had all sort of problems to execute our tasks. It was basically confusing and I felt sorry for the instructor
* I have no clue about what I did and why

Solution: redoce the whole session at home or forget the whole day

* Good presentation
* Maybe too many things in a short time
* Very relevant also for python users
* Maybe a bit much, but ok if intended to be an overview
* Small version problem with python (libraries could only be loaded when using python 3)
* Learned about a lot of tools that I’ve never used
* Fun!
* Jumping too quickly into make/cmake without introduction to these
* Great content once caught up to speed

On windows with only software in coderefinery installation: “cmake -G “MinGW Makefiles” as cmake is trying to use Visual++ by default

* Very useful
* But a bit difficult to follow, should probably use a bit more time on this since it seems to be very useful!

Would be useful to have some Matlab examples otherwise quite packed

* Clear and well prepared contents and nice documentation
* Quite intensive
* Very useful with the pybind exercises
* A little too fast

## Documentation

* Covered a large area
* Not possible to get in-depth but that is probably best to do with google on my own

Default sphinx on my system is 1.2.3 with slightly different syntax

* Interesting to learn
* Good (clear) presentation
* I was tired after a long day
* Good and well prepared lesson and also well documented
* Many highly useful and popular tools/solutions
* Good fun!
* Maybe example how to include docstrings
* Again, discovered many useful things
* Fun!
* None
* Useful information
* Sphinx index.rst - it could be clearer on how to add things. I lost a lot of time with this
* Motivation part could be clearer
* Very helpful instructor

Could have been earlier in the day as it is a bit more relaxed compared to CMake session

* Very clear, and good exercises
* Would be nice to have a cheat sheet for how to write the index.rst files
* Never knew it is all so simple!
* My nonexistent documentation might start taking form from now on

## CMake

* The instructor was patient and speaking slow
* Had compilation problems and was not able to participate
* Just listening did not help, I basically got lost and did not listen any more

Solution add an extra test to check if everybody installed the stuff correctly

* Good - I have some experience with cmake before, but it was very nice to have a more systematic introduction. I got a bit more understanding of how cmake/make actually works
* Content very useful, i think
* Clear presentation
* Well done. Good examples.
* Perhaps a bit more time required to assimilate things but there is enough material provided to do that
* Very clear, good exercises
* A bit too fast
* Good, pedagogical approach starting with “hello world”
* The schedule page was confusing
* I recognized that cmake is something I need and will try to integrate it into my project
* No time to really learn everything
* Even if I often came across cmake/make, it was my first time actually writing my own cmake/make files
* “Make” seems to be very convenient to run pipelines
* Overall, a very good appetizer, motivating to learn more!
* Make for pipelines (data analysis)
* Not particularly relevant for myself… won’t use cmake myself though still good to know

## Unknown session

* Interesting content, well presented
* In future, a course in modular programming would be interesting