The comments have been typed literally. Clarifying comments in [brackets] in black are mine (Radovan). Only obvious spelling mistakes have been fixed. We have asked people to mention one good thing (this is marked as green) and one thing that we should improve (this is marked as red). Comments given by the same person are grouped next to each other.

**General comments**

Quick feedback from 40 minutes on Wed:

* Do a round robin introduction of the participants, either during your session, e.g., after introducing the lecturers and helpers, or at the beginning of first session, here Jyry's git. Usually ~95% participants like it, it helps networking tremendously, in addition it helps you to sense better the level of the participants and their needs.
* Say a bit more about the program than that it is available at the workshop's webpage. Optional: when going through the program you may ask how many participants is especially interested in given topic, e.g., asking those to rise hands.
* In addition to gathering the feedback after each sessions, you may send a feedback form to all participants after the workshops. Sometimes new ideas/thoughts come a day or two after the event.

Once again thank you for the interesting workshop. :)

Thank you for the wonderful sessions that you conducted, particularly the ones on Day 2.

The course could have dived deeper from the angle of research. I felt that the focus was more on software development. (But I guess this was the intention)

Great course, maybe the content could be reduced a bit to not be in a hurry.

Remove DevOps and have more time for CMake! The DevOps lecture was too advanced with a lot of jargon so it was difficult to learn anything from it.

There was quite a jump in the pace and level of the lessons between the beginning (Git) and the end of the workshop.

The workshop was is of extreme use for me.  
The things I extremely liked:  
 - Dynamic, but slow passed without rush explanations of github main concepts  
 - System with stickers  
 - Pair exercises  
 - Assistants  
 - Nice supportive atmosphere  
To improve:  
 - Jupyter session with going through code was slightly slow paced..may be we should not go into details of data frames and go slightly faster:)  
 - To have a limit of questions to be in time and not to rush much to catch up with the material  
 - You can make small quizz in the end of the day via google forms to encourage people to remember the material better.  
 - To have adapters in case people are in trouble with plugs:)  
 - May be spend more time on github and ways to make your own project page..

How to use the gitlab.com - is it different? Seems to be free even for private repos.

[Radovan: Feedback I got via email:] “And yet again thanks for the course! I’ve been getting so much more out of git and also I’ve made good progress in trying to convince my colleagues to use it.”

**IDEs**

Highly motivating!

Break the session in two, perhaps

The session was good and thorough.

None

Good quick intro to PyCharm

Could be more extensive with practical stuff?

Good session and well presented

Nice content.

Session rushed thru, not much hands on.

Very good to see what you should expect from an IDE.

You should split this session in two, one for RStudio, the other for PyCharm.

Very nice introduction on IDE-s and very nice lecture in general, very well presented!

Good class, not too advanced, not too much content (PyCharm) [was marked as positive feedback]

“Big boys” -> “big gender neutral people”

**DevOps and automation**

Good overview of words I heard but previously didn’t understand at all

More explanation of terms and use cases to make it clear why a tool is needed or what it is good for.

Presenter was confident.

Presenter was silent; overview lecture seemed too general, containing too many general phrases like “there are tons of plugins”, “installation can be done easily …” without talking about those things in more detailed way.

Very interesting and clear introduction to the topic. Vagrant example was good!

The overall picture of DevOps is still a bit unclear (all the parts/programs?)

This was a well-organized part, but for me personally [last word was underlined] a bit too “deep”, unlike the previous ones.

Very good overview of tools and techniques out there for DevOps. Useful and practical.

The contents seemed useful and presentation professional but I was a little confused what was going on, what we were doing, in general. I guess I should’ve had more background in automation to understand today’s lecture. There were a lot of terms that I didn’t know. Maybe it’s a good idea to make the presentation more general or make a bit more clear in the beginning what we will do, in more general manner, and make sure everyone understands the terms mentioned in the lecture.

Less hands on. Vagrant didn’t work on my computer. Would have been better to ask the participant to test it beforehand and if any problems then could be resolved.

Knowledgeable instructor

Overall not an engaging session

The presentation was a bit confusing. Another example might be useful about the morning topics.

**CMake**

Great, detailed material for future reference.

Demo was too quick to digest.

Great to go through some of CMake features.

The session was conducted by too fast. But considering the time constraint, it’s understandable.

Good general introduction to CMake

Not useful for me at least yet.

Very good knowledge of practical CMake

Didn’t introduce ‘make’ concept. I encountered it for first time. Had hard time following it.

The CMake part was too fast to follow. It would have been nice to do even a very small example to understand the basics.

The CMake part was very useful and I believe it needs to be covered in further detail.

Well explained, clear and useful material.

The exercise would have been nice if we had time to do it ourselves. Without that much explanation, and in case of questions we have you there. For me at least it is easier to learn by doing.

**Jupyter part 1**

Useful intro and documentation.

Maybe you could spend less time on data frames, but rather more on Jupyter. Keep magic [“magic” is underlined] in, as they are useful. Make session more dynamic :-)

Material.

Pace.

Not enough context about Jupyter - did not quite get why should one use it.

Very good overview and detail about Jupyter notebook

It was good session.

Could have planned better to discuss everything.

Interesting

Completely useless for R user who would use RStudio/RNotebooks

Excellent overview, thx

Clear and relevant presentation

Do not add anymore content

Fast and pretty clear introduction to Jupyter

A bit “passive” exercise at this time of the evening!

I have never heard of Jupyter so I would have wanted to hear more about what/how it can be used (I don’t use Python so I maybe didn’t understand all the fancy features)

Came across different scientific packages.

May be a short (bit extended) [sic] session on syntax would be better.

Well-made documentation

Not so clear why to use the topic (Jupyter).

Did not seem very useful for me, I don’t think I will use it in the near future.

A bit too long, the time could have been used for more interesting topics (version control, modular code development, automated testing and CMake)

Good, didn’t notice any rush. Nothing to complain. Good that easier material at the end.

**Jupyter part 2**

The session was good.

It would be nice to have bit more on cpython [sic].

Nicely prepared notebooks with good examples.

Feedback given yesterday not taken into account today.

Very interesting and well explained.

As before, even if there are exercises we can do, it would be better to learn and understand (and keep attention) if at least some of the exercises could be done during the course, between explanations.

Cool and useful, but maybe more real exercises and less lecture? Check ‘Paja’-teaching approach at Uni Hel CS department and Math

High-performance and parallel stuff is really interesting.

Lesson could be a bit more interactive.

Good content.

Would be really nice if we do hands on while teaching is going on. Introduce a concept and implement it.

**Documentation**

Fast and easy.

At some point a bit fast, but I didn’t mind.

Good material.

Not too fast pace.

Would be nice to have an example of inline docs.

Super nice explanation, quick.

Would spend 2 hours on your session, + include deeper info about alternative markdown

The content was useful and got the idea in general what tools exist and how can docs be created.

Though there was not enough time to do try out things myself completely, but I think for this particular topic just a review was enough.

This was pure good, very useful.

None

Useful stuff.

Good instructions.

Could have been integrated with the previous tutorial (commenting the codes written in the previous tutorial)

It was again fast [“fast” underlined].

Good stuff though.

Can slow it down a bit.

Managed to perform in time while in lack of time.

Should have done only the demo, desperately trying to do hand-on didn’t work out (mostly because of lack of time).

Very important stuff

Wouldn’t have a clue of how to use this with R [“R” is underlined]

Pace just right.

Written material.

Speak a little louder :-)

**Testing**

Exercise with partner

Would have been useful to go through the exercise with instructor/group after doing it.

Again the session went great.

No negative feedback!

Example exercise was well prepared: easy and effective.

At some point it became difficult to follow presentation but probably it was due to late time.

Very good overview and exercise. Is fun and practical. Thanksss [sic]

Very nice exercise!

?

Excellent session.

Very important tools for coding.

Better to have more exercises.

**Git intro**

Good speed and useful examples/exercises.

Towards the end, when time was “running out” things got way too speedy and therefore confusing. This made it so that the following part on documentation was not too useful, since I was trying to do the previous part. Advice: even if time is running out, don’t speed up, better to understand more, even if less is covered.

Maybe less “let’s do step by step together” style education?

Good content, good hands on.

Teaching is too too fast, sometime I get stuck.

Well made material.

Too much info.

Understood how wonderful branching is.

Too fast. Jyry should “shut up” for 30 s each time he writes a sequence of commands on the screen.

Git, learning more about Git in more detail.

Good helper.

Need to be slower in presentation.

Introduce concept give more time to implement.

Liked ready code, stories available

Fast speed!

Lecturers supervised each other

Problem with time managing

Very nice exercises, shows (I guess) the basic features quite well

The post-it system is good

The last part was a bit fast

Maybe add creating a GitHub repository as prerequisite

Basic stuff was taught well (it was good to do some steps several times so now I might remember them)

Sometimes it was a bit too fast

The system with helpers works well

Schedule management

The course content was built well, spall chunks of info with example are best way to understand new material.

Even though there was nothing complicated and material was really easy, pace was too fast. It was hard to catch up and had to hurry typing, no time to think a bit.

Content was good

The instructor was bit faster, particularly when he was running the commands to explain the topic.

Very detailed!

All the material online.

This was way too fast. I cannot listen, do (type, execute) and understand new stuff at the same time.

Pace is good!

Nothing at the moment?

Story telling

Well prepared written material

For newbies to version control/ Git this way way too fast! (as hinted by several other participants during the session)

Difference Linux and Windows

Delete branch on Linux: git branch -d

Windows: git branch -D [Radovan: the person got an error on Windows and did not understand the behavior of -d]

HEAD^1 (Linux) vs. HEAD~1 (Windows)

Better use git bash than cmd

**Collaborative Git**

It was a wonderful session. Thorough, practical and effective. Thanks!

No negative side! :-)

Just a suggestion though: The instructor can take the intro session on Git as well in the future workshops.

Good speed and explanations quite clear.

I think that generally more practice and less explaining is more useful, so more exercises and less talking, even if the exercise is simple/trivial.

So much so good stuff! But I especially liked the archaeology part.

Large project management is not so relevant for me.

Very useful stuff

Interesting and good examples

There could have been even more exercises on the collaborative version control (or maybe that could be in a workshop just on Git)

Clear explanations

Good exercises

Few examples from programming working life experience

Like previous topics, content was well-chosen and well-explained. Information received is useful and worthy. Just the optimal presentation, nothing unnecessary or redundant. Overall, short and clear big picture on Git.

Distributed version control exercise.

More cherry-sized hands-ons.

Very good lesson for Git

Speed is fine

Good exercise and example.

Thank You!

A lot of new information

Maybe a bit inactive. Could include more exercises?

This pace was much more “human”, thank you

I liked the exercises

The “git bisect” thing isn’t explained in the online material (but I googled it in 20 seconds, so no hard)

Teaching and theory was good.

Exercise was done properly. Not done squashing, cherry picking.

Really helpful info on some advanced features of Git

**Complexity**

It’s great that Radovan decided to share his point of view/ philosophy.

Not negative side.

Nice and smooth presentation

Expected more specific tips, for example what should good program include, how to do it, common mistakes etc. For sure it’s hard to give general tips but it’s fine to do it on the basis of some language.

Very good idea on complexity for research project.

Good overview of modular development.

Dealing with the state section could be expanded. Also you could include some typical elements of bad code (code “smells”)

Important topic.

Very good content

I did not quite understand all the context (I am not from computer science background)

Good clear advices.

Nice speed.

Nothing to complain.

General yet useful

Recommendations for R too would be nice!

Good session. Good content

Very good tips

Personal experience and opinion

Nothing

Good principles, especially the emphasis on functions.

The problems of typical data analysis projects differ from software development projects