When writing your report keep in mind the following points:

1) Your report should be **self-contained**. The reader should be able to understand the whole story without looking into your code, reading instructions, or checking other papers.

2) Your report should be **well structured**. For each task explain what you were supposed to do, what you actually did, what are the results and, most important, draw some conclusions.

3) Your report should be **detailed**. When explaining your experiments, specify the setup (values of parameters you used, algorithms, options, etc.), so the reader could get a fair picture of what you did, without checking your code.

4) Your report should **contain observations and conclusions** that you've made while working on the assignment. In case you see something strange in your results, write something about it! Most important: try to explain possible reasons for the strange things you've observed!

5) Your report should be **technically correct**. Make sure that your report is neatly formatted, spell-checked, tables and figures contain informative captions. DO NOT INCLUDE your name(s) or e-mail address(es)!

When evaluating the reports take into account the following criteria:

1) **Clarity and Readability**: can you easily follow the story and understand what the authors wanted to communicate?

2) **Completeness**: are all tasks that were listed in the assignment fully addressed?

3) **Validity**: do you believe that the results are valid? Are they consistent with your expectations?

4) **Reflection**: are the observations and conclusions made in the paper properly supported by experiments? Are some interesting, unexpected insights presented? Have you learned something new when reading the report?

5) **Technical correctness**: is the report neatly formatted spell-checked, tables and figures contain informative captions, etc.?

6) **Bonus**: if you think that the report deserves an extra bonus, state it explicitly–we will look into it.

7) Quality of code: the course is not about "Programming in Python" so you don't have to look into the code, unless you suspect (based on the report) to find some bugs. The impact of these bugs will be already taken into account in the 5 criteria listed above. **However, you** (the reviewing team), **may receive extra credits (up to 0.5)** for providing constructive feedback to the authors of the code. If the code is perfect, instead of credits you get something more valuable: you learn how a perfect code should look like!