THE DATA ANALYSIS CHECKLIST

Make sure you have addressed each of the following before embarking on a new analysis of your data.

PURPOSE

What is this analysis supposed to achieve? What specific question am I trying to answer?

A concrete objective is the anchor point of any analysis. It provides a foundation for your inquiry and provides a contextualization for evaluating potential extensions. Further, a well defined goal allows you to specifically gauge your progress and assess when your analysis is complete.

Most analysis goals fall into one of the following categories:

Characterization
Answers the question: What are the basic scales, correlations and dimensionalities in my data?
Is appropriate when: First analyzing a dataset. However, simple characterization rarely leads to unique insights.

Exploration
Answers the question: Are there any obvious interesting structural features in my data?
Is appropriate when: Ensuring your data has been collected properly and when familiarizing yourself with the data beyond basic characterization. If nothing noteworthy emerges, however, you must decide whether to pursue more focused directions.

Hypothesis Test
Answers the question: Does my data support/refute a particular hypothesis?
Is appropriate when: A specific hypothesis has been suggested by either data exploration or known properties of related systems in the literature.

Hypothesis Comparison
Answers the question: Which of two or more hypotheses best explains my data?
Is appropriate when: Multiple reasonable hypotheses have been suggested by data exploration or the field at large.

Prediction
Answers the question: How well can one set of features in my data be predicted from another?
Is appropriate when: You only desire better approximations of relationships in your data, regardless of their internal meaning (typically in clinical or engineering applications).

EXPECTATION

What do I expect my results to look like? Can I imagine at least two reasonably likely outcomes?

This doesn't mean your expectations will be met, of course, and much good science emerges from unexpected results. However, if you can't think of anything that would be both meaningful and reasonably likely to observe you might want to rethink your analysis strategy.

VISUALIZATION

What plots or tables will I use to depict my results?

Imagining the end product of your analysis as precisely as possible provides a useful target for directing your methods. Good mock visualizations are also very useful for ensuring you will know how to interpret your eventual results.

MEANING

Will my results meaningfully expand my understanding of the data? Will they reflect the structure of the data more than the structure of my analysis?

Translating a plot into a statement that reveals subtleties of the underlying system's composition can be immensely challenging. In complex analyses especially, it is rarely straightforward to determine how strongly the results have been influenced by the analysis details vs. the internal data structures. E.g. in a clustering analysis the resulting clusters can easily reflect your clustering criteria rather than natural groupings in the data. Think carefully about how interpretable your expected results will be before you start a complicated analysis.

EXPLAINABILITY

Can I clearly and concisely explain my analysis so my audience understands and trusts it? If necessary, can I explain and justify every detail?

In the end you have limited time to explain your analysis and convince your audience of its reasonability. A simple analysis that sacrifices a small amount of predictive power can therefore be more scientifically impactful than a more predictive but highly convoluted one, even if the latter is sound and logical. Further, being able to explain your analysis concisely is a good way to ensure you fully understand it yourself.

PARAMETERS

How will I choose the parameters of my analysis and quantify my results' dependence on them?

Almost all analyses require parameter choices, e.g. normalization factors, bin sizes, or smoothing scales, and these quantitatively affect the results. While it's usually intractable to re-run your analysis for every possible parameter combination, you should at least be able to reasonably speak to the effect they have on the qualitative features of your results.

CONFOUNDS

Could factors I don't care about cause my expected results?

Naively interesting structures in a dataset often result from properties of the data irrelevant to your scientific question. For instance, a strong correlation between two variables might be caused by shared but fundamentally irrelevant external input. It can take a lot of thought to ensure your analysis controls for all important confounds.

STATISTICS

How will I quantitatively argue my results didn't arise by chance?

For extreme results this is often only a formality, but for subtler results statistical tests are key. These don't have to generate p-values, but they should be statistically sound and mathematically interpretable.

CODE STRUCTURE

What code will I write? How will I organize it?

The way your code is written can be the difference between a set of crisp, reproducible results, and an unending string of headaches. Thinking through the organization ahead of time so your codebase doesn't end up a random sequence of haphazard logic statements is highly worth it. One useful technique is to start with high-level function specifications and then fill in the details from the top down.

CODE VALIDITY

What tests will I run to ensure my code works properly?

Just because your code doesn't crash doesn't mean it's working. It's crucial to verify functionality with mock datasets or contexts where the ground truth is known. These tests should be included in your codebase itself (as opposed to being run in an interpreter), so that they can be quickly re-run if the analysis code changes. Note that it's common for tests to take up more than 50% of your codebase.

CODE INTEGRATION

Once I write my basic code, how will I integrate it into my existing codebase?

This is especially important in the highly iterative research process, in which one might run several variations of each analysis. While individual changes are often small, they can quickly tangle into a morass of complexity requiring large-scale refactoring to ensure intelligibility. Importantly, while the mathematical heart of your code might be only a couple lines, large infrastructures are often required for file I/O, array manipulation, data formatting, etc. While these tasks are often conceptually straightforward, they can be complicated and time-consuming to implement and are frequently bug-ridden.

RESULTS INTEGRATION

How will I store my results so I can quickly access them later, recalling every detail if need be?

The iterative research process can make for a tangle not just of code, but of figures and tables too, and an analysis has no value unless you can efficiently recall the methodology and output, even if it didn't yield anything noteworthy. As with code, when your set of results from different analyses reaches a critical mass, it can be worth the time to reorganize it.

RISK

What parts of my analysis still contain uncertainty? How is my analysis most likely to fail?

You don't need to flesh out every detail before starting, but you should estimate how much risk each unknown involves. These range from not knowing how to implement a particular computation, to not knowing how a certain milestone will turn out, to not knowing how long your funding will persist. Isolate all uncertainties and failure points and either convince yourself of your ability to work through them or estimate the risk of each to the success of your analysis as a whole before proceeding.

TIME

How long will it take to complete the analysis from start to finish? To write the code? To run it? To test it? To organize and present the methods and results?

All analyses must be completed in a finite amount of time, and if you can judge how long something will take it will be much easier to decide whether to pursue it. This isn't easy, but as you gain experience you will improve your ability to imagine everything involved from start to finish and to quantitatively estimate the time and resources required.

COMPLEXITY

Have I simplified everything as much as possible?

The simpler your analysis is, the easier everything above will be to address, the fewer headaches you'll encounter, and the faster you'll reach your goal. Complexity should only be added when it is absolutely necessary for your final objective.