How to Test Data Analytics

(without knowing anything about Data Analytics)

By Daniel Hunt

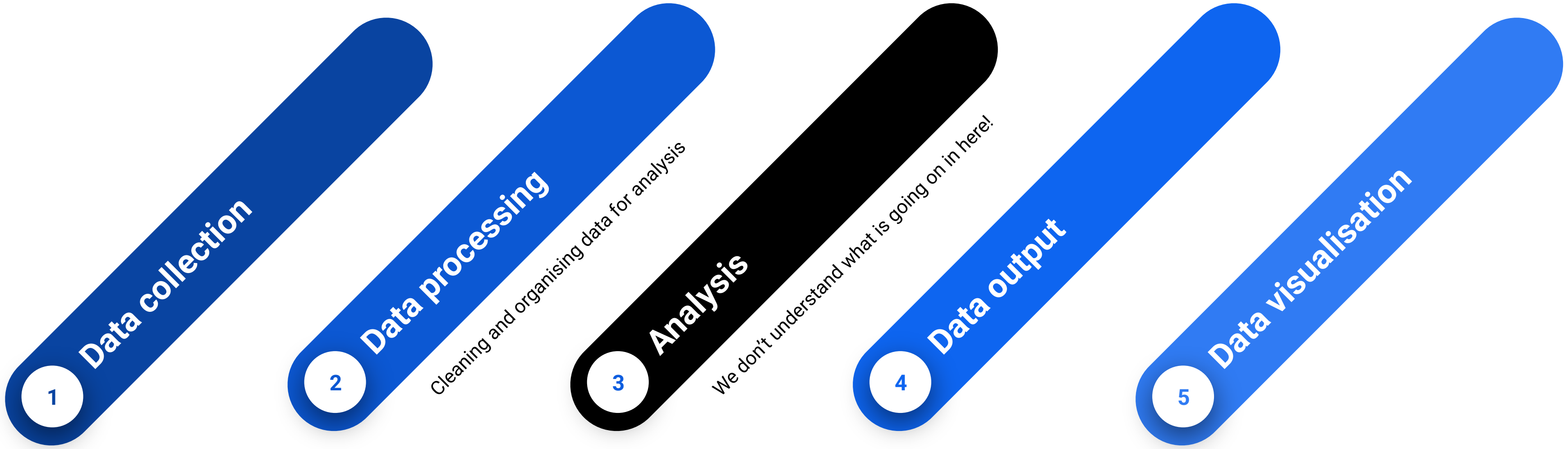
What is Data Analytics?

Data analytics is the process of examining raw data with the goal of discovering useful information that can be used to inform conclusions and support decision-making

Why is testing analytics different?

- Data analytics is normally somewhere between creating a piece of software and a research activity
- There is very rarely a specified “correct answer” or expected behaviour. As a result I’ll be proposing making multiple incomplete attempts to test. If all these incomplete approaches give sensible results then the hypothesis is that the sum of the approaches (the analysis overall) is more likely to be correct
- Without understanding the analysis and reproducing it independently this is often the best we can do (which may still not be good enough)

Model of data analytics processes


1
Data collection
2
Data processing
Cleaning and organising data for analysis
3
Analysis
We don't understand what is going on in here!
4
Data output
5
Data visualisation

My 3 Step Plan of Attack

- Data, Data, Data!
  - Garbage in, garbage out
  - We’ll look at some common ways bad data can cause problems

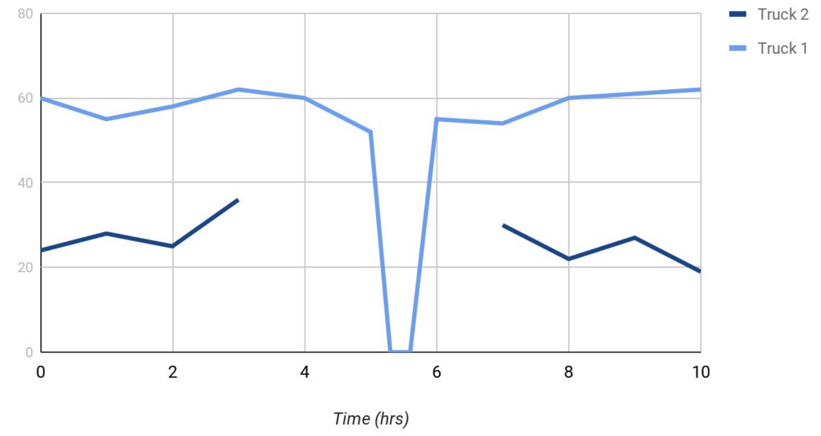
- Black-box approaches to test the analysis step in simplified ways

- General testability priorities
  - Some project/process-based actions that you can take to improve success chances

Data: Gaps

- Can you trust the output of some analysis when there are large gaps in the data? Particularly relevant for time series data.

Vehicle speed



Data: NULL

- Can null or “empty values” be treated as 0? Sometimes this is the default behaviour so watch out!
- Does it make sense for a row of data to be used if it contains null values?
- Do fields treat variants in the same way: “NULL”, “null”, “”
- Null codes

Data: Updating

Is there a process by which data can be updated?

Does old data get deleted or archived?

Can users enter information and is there appropriate validation in place?

Does analysis get re-run whenever the underlying data changes?

Data: Overlapping

- When data overlaps does one value take precedence over another?
- If not does it make sense to include multiple values in calculations?

Say you have a piece of machinery that can have multiple different states, but is only ever in one state at a time. If you have data that indicates that it was in state A (in operation) from 9:00-15:00 and that it was in state B (in maintenance) from 12:00-16:00, then what is the correct output?

You could say that it was in state A for 6 hours and state B for 4 hours, giving it a total running time of 10 hours within a 7 hour period…

Data: Mapping

Single entities can have multiple names and for the purposes of analysis we likely want to make sure we combine these records. For example I’ve seen a highway referred to by five different names in the same dataset:

> Utrechtsebaan / A12 / E30 / E35 / E25

Conversely, multiple entities can have the same name and in this case we likely want to treat them separately. Typically this will involve the use of some unique identifier. It is worth checking that this is truly unique!

Data: Units

- Keep track of units of measurements. Particularly when data is coming from multiple different sources - they will need to be mapped into a common unit
- kilodecanewton-feet is my personal favourite example in the field *(1 kilodecanewton-foot is 3048 newton-metres which is the more conventional unit for torque)*
- Unfortunately this will also include timezones...

Black-box methods

Now we are going to try to probe inside the analysis black box. We have access to the inputs and the outputs. The main difficulty is that **we will rarely know the correct answer/output.**

Without a definitive way to decide “this is working correctly”, it will be much easier to identify situations when it is failing.

Black-box: Go full end-to-end

Get some data or create your own test data and then run it through every step of the process.

After each stage of data ingest, data processing, data output and data visualising - check that the data out looks sensible given the data in.

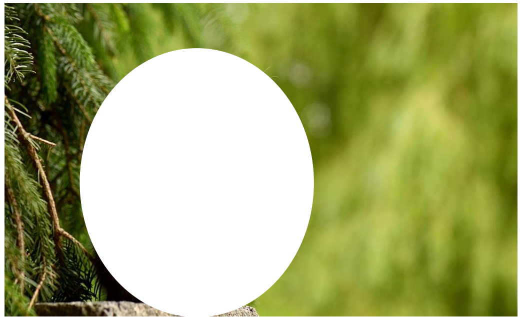
Breaking it down in this way will make it far easier to spot where the error is. It also makes errors easier to identify

Black-box: Construct simple test cases

Even without understanding anything that goes on inside the “Analysis” part of the process, we can usually recognise what should happen in some cases.

For example, say we had an image classifier that identifies whether a picture contains a cat or a dog

96%
Cat!



??

Black-box: Check real-world constraints

We may not know what the correct output is for any given input. We can definitely identify some incorrect outputs however!

If you were analysing truck driver performance using telemetry data, then:

- Negative speed is an error or time travel
- Travelling faster than the speed of sound indicates an error
- A truck carrying a negative amount of weight indicates an error
- Someone driving two different trucks at the same time?
- A driver logging more hours in a month than legally allowed?

Testability: Maximise understanding

As with most areas of testing, having more information is very useful. It will improve the quality of any black-box methods you try to use and make it much easier to recognise mistakes early.

Ideally you want to get some idea of “what looks sensible”

Testability: Advocate automated testing

In general, having some automated unit and system tests is worthwhile. Most data analytics systems will deal with large quantities of data and a lot of complexity.

With too much data to check all by hand, computers are your friend!

Unit tests can be particularly important if there are going to be changes made to the analytics part of any system going forward. Introducing regressions can be very tricky to spot or diagnose otherwise

Testability: Use real-world data

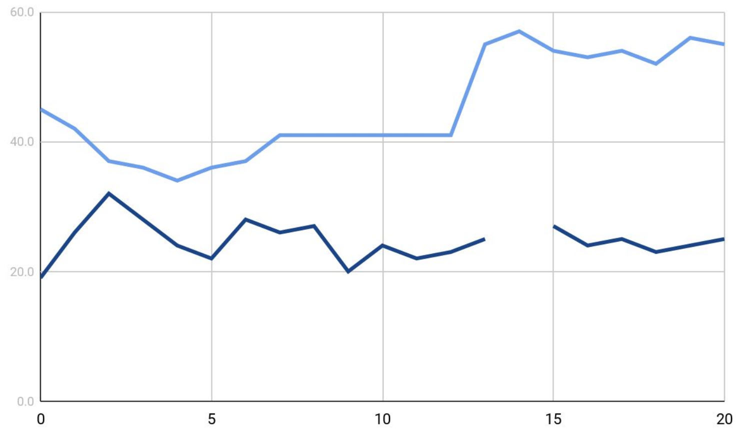
Get as much real world data as you can!

If you are lucky then you may have verified outputs for example if someone has - ie putting in data X will get Y results.

Treat it like it is worth its weight in gold. Version-control of the data and of any analysis output is a good idea. Otherwise evaluating changes to the system over time will be harder.

Testability: Visualise early

Time (hrs)	Dark Blue	Light Blue
0	19.0	45.0
1	26.0	42.0
2	32.0	37.0
3	28.0	36.0
4	24.0	34.0
5	22.0	36.0
6	28.0	37.0
7	26.0	41.0
8	27.0	41.0
9	20.0	41.0
10	24.0	41.0
11	22.0	41.0
12	23.0	41.0
13	25.0	55.0
14	0.0	57.0
15	27.0	54.0



Conclusion

With incomplete understanding you can only do “incomplete tests”. The 3 step plan I’ve outlined here is how I approach this problem.

Data, Black-Box Methods, Testability

Remember there will be occasions where whatever you do will not be enough. Don’t beat yourself up about it!