Power BI for Physics

Edmund Ting

July 16, 2025

Abstract

Documentation for the Power BI for Physics project. Presents a brief physics preliminary before showcasing each of the Power BI projects in the repository, with descriptions of how its suite of tools (M, DAX, etc.) were used to manipulate and produce visualisations of the data.

Contents

1 Preliminaries

The Large Hadron Collider (LHC) at CERN facilitates the collision of two antiparallel proton beams at extremely high energies. As the protons collide, they can undergo interactions governed by our most complete subatomic theory to date: the Standard Model of particle physics, which describes the fundamental particles and forces in our universe.

The driving motivation behind the physics programme of the LHC is precisely to gain a deeper understanding of the Standard Model. Being built upon the foundation of quantum field theory, the Standard Model is inherently non-deterministic. In fact, there is only a *tiny* probability for something to happen when two protons are made to collide; furthermore, the nature of exactly *what* happens is also probabilistic, as there can be a plethora of interactions that are possible. Thus, statistics lies at the core of experimental high-energy physics (HEP), and the work of the experimentalist is to sift through extremely large volumes of collision data in order to draw meaningful insights about the Standard Model.

The examples in this *Power BI for Physics* project are based on this educational resource by the ATLAS collaboration. The premise is to analyse publicly available LHC data to rediscover the Higgs boson, recreating its momentous discovery by the ATLAS and CMS collaborations in 2012.

1.1 Why Power BI?

Power BI (*Business Intelligence*) is Microsoft's analytics platform (only available on Windows) optimised towards creating interactive and digestible reports from data in the business context.

Still, at its core, it is a tool used for manipulating and visualising data, and so it is also possible to conduct a physics analysis using it. As we shall see in the examples presented later, despite not being a practical tool for analysing large-scale HEP datasets, there are actually some nice features built into the platform particularly when it comes to creating interactive visualisation elements.

1.2 Basic workflow

Using Power BI essentially boils down to a few simple steps:

1. Import the data (from e.g. a local file, SQL server, or data warehouse).
2. Construct a *semantic model* (i.e. one or more connected tables) from the data.
3. Arrange visualisations (histograms, charts, etc.) together to create a report.

There are a few things to keep in consideration for each of these steps. Firstly, the size of the `.pbix` project file will scale with the amount of data imported in step 1. Thus, it's important to trim away as much of the irrelevant data as possible during this step. Power Query M is the premier tool for this: it allows the user to filter and transform the input data prior to loading it into Power BI.

Semantic models represent a structured view of the imported dataset. The idea is that since the data could have been fetched from multiple sources, there can be commonalities or redundancies between them. For example, two tables could contain the same column, which would establish a relationship between their other columns. The interface for doing this is simple (one just drags and drops arrows between tables in a graphical view). Power BI also allows the user to calculate new variables at this step using DAX (*data analysis expressions*).

Finally, build visualisations of the data. Click different buttons, arrange till pretty.

1.3 Preparing physics data for analysis in Power BI

Due to the complex structure of LHC collision events, HEP data is typically stored in the ROOT format. Unfortunately, this is incompatible with Power BI, so relevant information needs to first be extracted from the files and exported into another format.

To parse a ROOT file, the uproot package in Python can be used. This allows for branches in the ROOT file to be extracted as Awkward arrays, which can then be exported into, for example, a `csv` or Apache Parquet file. (Note: it is not recommended to export a `csv` if processing a large number of events, as the write rate is *extremely* slow.)

2 Example: Plotting the Higgs peak

2.1 Applying filters using Power Query M

```
let
Source = Parquet.Document(File.Contents("photons.parquet"), [Compression=null,
    LegacyColumnNameEncoding=false, MaxDepth=null]),
#"Select two photons" = Table.SelectRows(Source, each (List.Count([photon_pt])
    >= 2)),
#"Select Tight ID" = Table.SelectRows(#"Select two photons", each
    ([photon_isTightID]{0} = true and [photon_isTightID]{1} = true)),
#"Select minimum pT" = Table.SelectRows(#"Select Tight ID", each ([photon_pt]{0}
    > 50 and [photon_pt]{1} > 30)),
#"Select isolated" = Table.SelectRows(#"Select minimum pT", each
    ([photon_ptcone20]{0}/[photon_pt]{0} < 0.055 and
    [photon_ptcone20]{1}/[photon_pt]{1} < 0.055)),
#"Select eta" = Table.SelectRows(#"Select isolated", each
    ((Number.Abs([photon_eta]{0}) < 1.37 or Number.Abs([photon_eta]{0}) > 1.52)
    and (Number.Abs([photon_eta]{1}) < 1.37 or Number.Abs([photon_eta]{1}) >
    1.52))),
#"Define invariant mass" = Table.AddColumn(#"Select eta", "photon_m", each
    Number.Sqrt(2*[photon_pt]{0}*[photon_pt]{1} * (Number.Cosh([photon_eta]{0} -
    [photon_eta]{1}) - Number.Cos([photon_phi]{0} - [photon_phi]{1})))),
#"Select invariant mass isolation" = Table.SelectRows(#"Define invariant mass",
    each ([photon_pt]{0}/[photon_m] > 0.35 and [photon_pt]{1}/[photon_m] >
    0.35))
in
#"Select invariant mass isolation"
```

2.2 Using Python for visualisation

3 Example: Interactive data filtering using slicers