Power BI for Physics

Edmund Ting

July 21, 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

As the first example, let's start with the very basics of a typical analysis in HEP. The problem statement is this: we have data that is distributed across a multi-dimensional space, and the objective is to identify a region in this "phase space" where contributions from a "signal" process is statistically significant with respect to other "background" processes. In this case, the signal corresponds to the production of a Higgs boson that then decays into two photons, while backgrounds consist of other Standard Model processes that also produce photons.

As preparatory work, for all photons in each event, the following variables were exported into a Parquet file: the transverse momentum (p_{T}), pseudorapidity (η), azimuthal angle (ϕ), and Boolean flags indicating whether they pass identification and isolation requirements.

2.1 Applying filters using Power Query M

Importing the Parquet file into Power BI, we obtain the following table:

	photon_pt	photon_eta	photon_phi	photon_e	photon_isTightID	photon_ptcone20
1	List	List	List	List	List	List
2	List	List	List	List	List	List
3	List	List	List	List	List	List
4	List	List	List	List	List	List
5	List	List	List	List	List	List
6	List	List	List	List	List	List
7	List	List	List	List	List	List
8	List	List	List	List	List	List
9	List	List	List	List	List	List
10	List	List	List	List	List	List
11	List	List	List	List	List	List
12	List	List	List	List	List	List
13	List	List	List	List	List	List
14	List	List	List	List	List	List
15	List	List	List	List	List	List
16	List	List	List	List	List	List

Because each event can contain multiple photons, Power Query displays each entry in the table as a **List** object. These are sorted in decreasing order according to the photon p_{T}, i.e. the first element in each list corresponds to the highest p_{T} photon, the second element to the p_{T}–sub-leading photon, and so on.

The Power Query M formula language can be used to filter the source data. The idea is that we start with the full table from the source, select only a subset of its rows based on some condition, and define this as a new table. This procedure is repeated successively until all the desired filters are applied, and the final data table is then passed to Power BI for analysis.

The basic structure of M's syntax looks like this:

```
let
Source = ***point to the input source data***
#"New table" = Table.SelectRows(Source, ***filter condition***)
#"Second new table" = Table.SelectRows(#"New table", ***another filter***)
in
#"Second new table"
```

Importantly, note that the hash symbol is not a comment, but rather how the table variables are defined. The line after `in` determines the table that is imported into Power BI.

For this example, we want to define one new variable (i.e. add a column to the source table) and apply six filter requirements:

- at least two photons in the event
- both of the two leading photons must pass the identification criteria
- the leading photon must have $p_{\mathrm{T}} > 50\,\mathrm{GeV}$, and the subleading photon $p_{\mathrm{T}} > 30\,\mathrm{GeV}$
- both of the two leading photons must pass an isolation check: $p_{\mathrm{T}}^{\mathrm{cone,20}}/p_{\mathrm{T}} < 0.055$
- both of the two leading photons must satisfy $|\eta| < 1.37$ or $|\eta| > 1.52$
- the invariant mass of the two leading photons, calculated as

  $$m_{\gamma\gamma} = \sqrt{2p_{\mathrm{T},1}p_{\mathrm{T},2}(\cosh(\eta_1 - \eta_2) - \cos(\phi_1 - \phi_2))}$$

  must be less than their respective $p_{\mathrm{T}}/0.35$

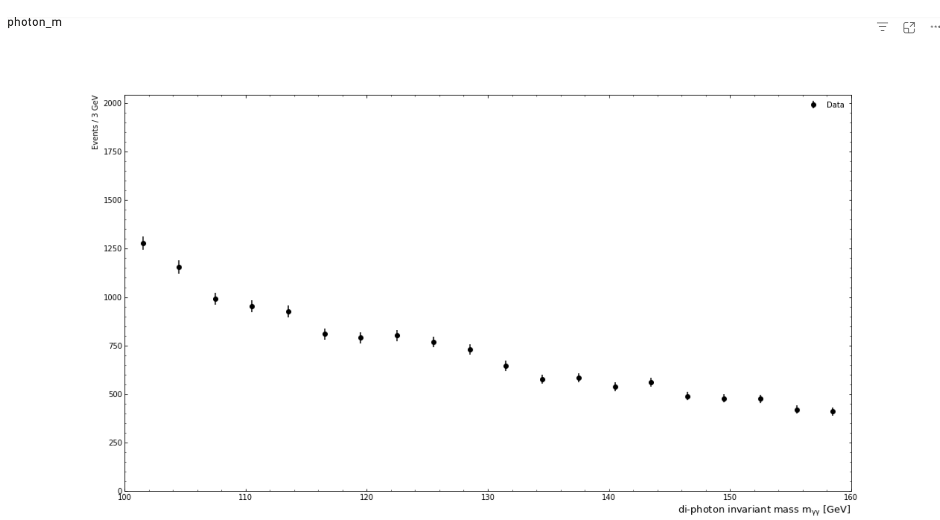
The M code to calculate $m_{\gamma\gamma}$ and implement these selections is as follows:

```
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

For this example, all of the necessary filters are applied in Power Query, and the semantic model is simple, since there is only one table (corresponding to the di-photon invariant mass in each event).

We'll visualise the data using Power BI's integrated Python scripting functionality (the icon for this looks like Py). This is a flexible option, since it allows one to leverage Python's large suite of plotting tools. In this instance, let's plot a histogram of the di-photon invariant mass. The excess of events near 125 GeV is the Higgs peak that we set out to recreate.



3 Example: Interactive data filtering using slicers