Power BI for Physics

Edmund Ting

July 15, 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 **1**

2 Preparing physics event data for analysis **1**

3 Example: Plotting the Higgs peak **2**

3.1 Applying filters using Power Query M 2

3.2 Using Python for visualisation 2

4 Example: Interactive data filtering using slicers **2**

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 sieve 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.

2 Preparing physics event data for analysis

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 needs to first be processed to extract the relevant information and export it 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 that it is not recommended to export a `csv` if processing a large number of events, as the write rate is extremely slow.)

3 Example: Plotting the Higgs peak

3.1 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"
```

3.2 Using Python for visualisation

4 Example: Interactive data filtering using slicers