STRINGFEST ANALYTICS

WORKING FILE – PROCEED WITH CAUTION

FIVE THINGS EXCEL USERS

SHOULD KNOW ABOUT POWER QUERY

Contents

0. Getting Started

Thank you for picking up this white paper. Power Query is the most exciting Excel release in 20 years. You are going to love it; your relationship to data and data cleaning will never be the same.

The objective of this white paper is to orient you to the tool and what it can do for you. We'll see how Power Query busts older limitations of Excel in an easy-to-use modern interface.

Please be aware that Power Query is **not fully available on every version of Excel**. You can see all the versions of Excel that feature Power Query here. I am writing this on a Windows PC with the Office 365 desktop version of Excel installed. For best results following along, you'll be on the same.

Companion workbook

One learns best by doing. The more time you have your hands on the keyboard as you learn Excel, the more you'll remember. To that end, **download the companion file for this white paper here.** This is a real-life dataset consisting of housing prices. Learn more about it here.

I will provide a completed version of this workbook at the end of the white paper.

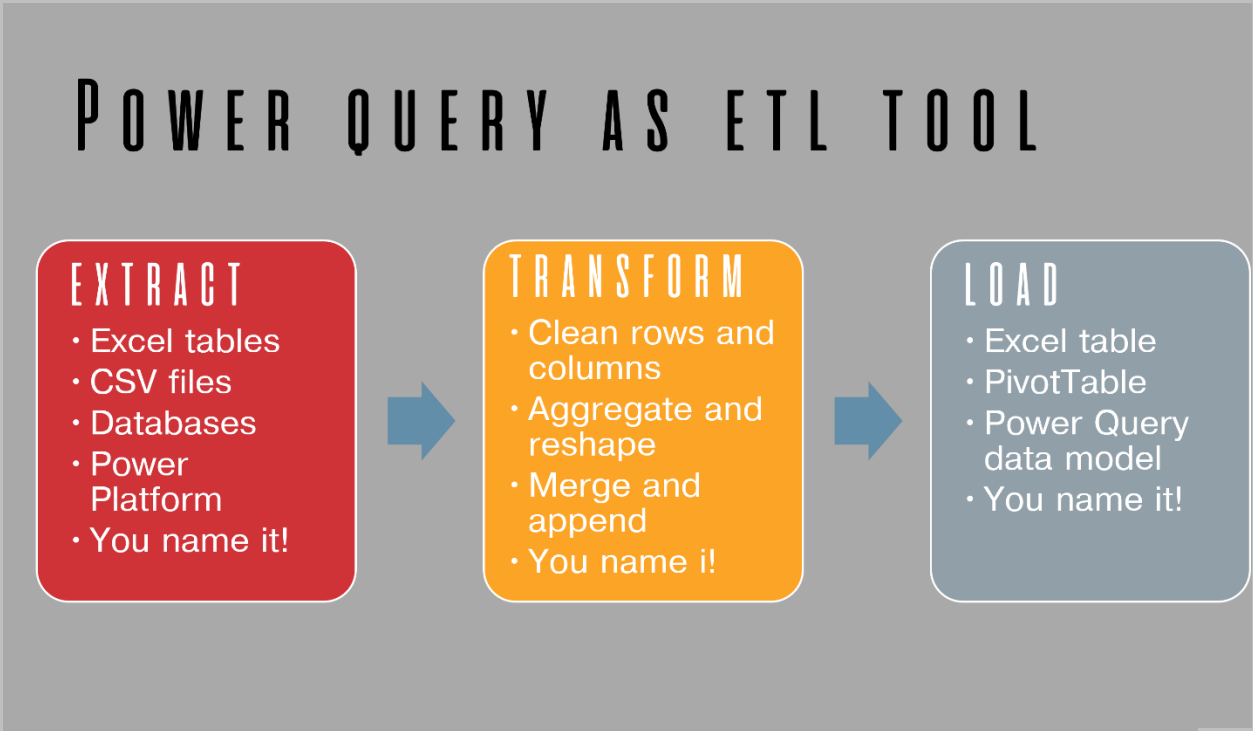
1. It's not just for Excel

While the focus of this white paper is on Excel for Power Query, perhaps the first thing you should know about this tool is **it's not just for Excel.**

You'll find Power Query in Excel, Power BI and Power Automate, among other products in the Microsoft BI stack. Most if not all of the principles you'll read about in this white paper apply equally to these other tools. So you're really getting a bargain here (both from me and Microsoft)! You can learn more about where Power Query is available on the official Power Query page from Microsoft.

At the time of writing, the first sentence on this page to explain Power Query is this: "Power Query is the easiest way to connect, extract, transform and load data from a wide range of sources." Let's drill in on this definition.

2. It's an ETL tool



Let's be honest, there are a lot of terms in tech that sound way more complicated than they really are. Sometimes it seems like these acronyms are used intentionally to create confusion. But peel back the concept and everything is clear.

ETL, or extract, transform, load is one of those terms. You'll hear database administrators and data engineers fawn and hype over their "ETL pipelines" and "ETL tools." Sounds like something only a certified data geek can do, right?

In fact, Power Query serves to democratize the process of ETL right to an Excel spreadsheet. Let's see how that works, and what ETL does.

Don't let the techbros intimidate you: the meaning of ETL is right here in plain English, and can be executed in Excel! Let's walk through each step in Power Query now.

a. (connect to and) Extract

The first step of ETL is to "extract" the data from an outside source. To do that, you of course need to connect to the data. Excel Power Query can connect to an outstanding variety of data sources… not just Excel workbooks!

Some of these include:

- Text and CSV files
- Databases
- SharePoint
- XML, HTML, Web data

For right now, let's just stick with a plain old Excel workbook. If you haven't downloaded the starter file yet, here's another chance. Download it now and open.

Our first order of business is to *extract* the data from the `housing` table. Now it seems a little strange to be "connecting to and extracting" data that we already have, right in this workbook. It's arguably overwrought, even.

What makes this such a powerful idea is that *Power Query is forcing us to keep the raw data intact*. We must take an *extract* of it (There's the E). So let's connect to this data, *then* transform it (There's the T!).

To do so, click anywhere inside the housing table, then go to Data > Get Data > From Table/Range:

INSERT IMAGE HERE

The Power Query editor will now launch in a new window:

INSERT IMAGE HERE

We'll cover the basics of the Power Query editor in this white paper. For a more detailed tour, check out this post.

Congratulations on getting the ETL started! An E for effort… and extract! Let's move to the T.

2. Transform

The next step is to transform the data. This could be a whole lot of things, but it's really all the steps you must take to make this data usable, such as:

- Sorting or filtering rows
- Adding, dropping, renaming or calculating columns
- Grouping by and aggregating categories
- Merging multiple data sources

Let's make a very simple transformation to this data: we will filter it to keep only the records where `prefarea` are set to `yes`. You can do this by clicking the dropdown menu on the column header:

INSERT IMAGE HERE

Click OK. And the data transformation begins!

There's of course a *lot* more you can do with data transformation in Power Query, and most of your time is going to be in this step of ETL. But let's move to that last leg, or L for load (say that 10 times fast!)

3. Load

Finally, click Home on the Power Query editor, then Close & Load. You will see the "transformed extract" of your housing data loaded into Excel:

INSERT IMAGE HERE

Congratulations for completing an entire ETL job!

- You extracted the raw data from an Excel table
- You transformed the data using the Power Query editor
- You loaded the results back to Excel

This is of course a quite diminutive ETL job. But regardless of complexity, Power Query lets you build repeatable data cleaning processes using a variety of sources. **Very** helpful, if I do say myself.

3. It generates code

Guess what… you just produced code in Power Query! Don't believe me?

Let's go back to the Power Query Editor to find out. To do this, go to Data > Queries & Connections.

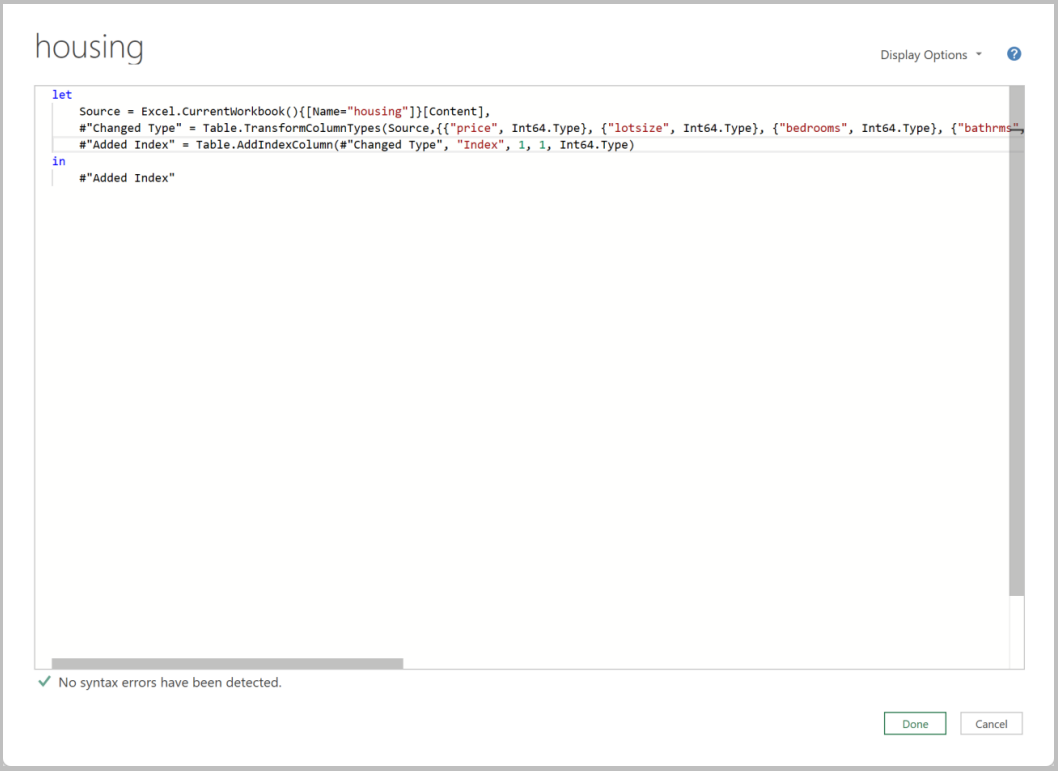
INSERT IMAGE HERE

Here you will see a list of the queries in our workbook. We just have the one, _____. Go ahead and right-click on this in the menu, then select Edit. The Power Query will reappear. Click on the View tab of the ribbon.

INSERT IMAGE HERE

Under View, select Advanced Editor. You should see something like this:

Go to the advanced editor:

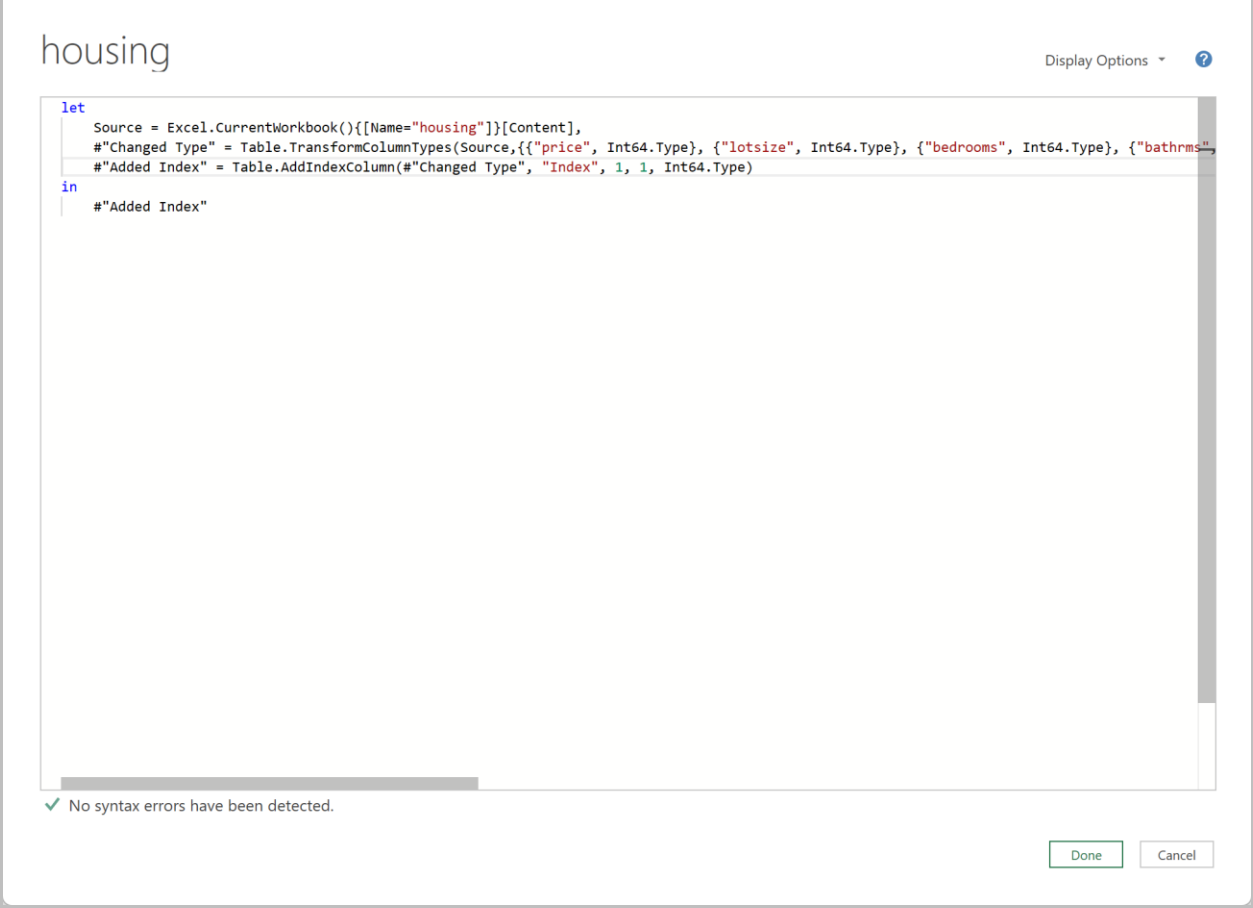


Where did this code come from? The Power Query Editor is a syntax generator. That means that code, or syntax, is created each time you make a change in the Editor. If you've used SPSS or the VBA Macro Recorder before, this type of behavior may be familiar.

But what *kind* of syntax is this? VBA? Python? Something else? In fact, this is the M language developed specifically by Microsoft for Power Query. To learn more, check out this article from Microsoft.

The vast majority of your work in Power Query can be done without ever interacting with the generated M code. But as you progress in your Power Query journey, you may find yourself coming here for certain tasks.

Let's make an edit here ourselves: change the filter so that `prefarea` must now equal no:



You'll see that the filter changes.

4. It's fully reproducible

Picture this:

you're working in an Excel workbook and delete a column. EEverything in it breaks. Or, you are trying to audit through some process and just can't figure out where some numbers came from.

You should have like a recipe… this is what Power Query does…

Instead we will be letting Power Query deal with the code under the hood and just use the menu. For example, let's delete a column. Simply right-click on it and select DELETE COLUMN.

ADD IMAGE HERE

In traditional Excel deleted columns could prove catastrophic. In Power Query it's not a big deal because you always have the source data!

Reproducibility means that an end user should be able to achieve the same result as the author, given access to the same input data and same process script. It's like having a recipe you can follow to get the same dish from the same ingredients.

Power Query avoids this by making everything code. We can see all the steps along the way. Look at the applied steps editor:

Applied Steps is a more accessible view:

INSERT IMAGE HERE.

Not only this but you also know from Thing 2 that this has all been done by code. So for example, if we wanted to see *how* they were filtered. Go back to your query.

INSERT IMAGE HERE

Now we want to delete a column. The great thing with this is we are keeping the source data intact

So actually, there is a recipe.

Go back to the "Source" and you will exactly see the source data as it was.

INSERT IMAGE HERE

Excel is not reproducible is a refrain you'll hear time and again. And it's just not true anymore… a myth, busted…

5. It's an Excel myth buster

44 Excel is a legendary business application. But sometimes, legend passes to myth. And not all these myths are true, at least not anymore. Because, you see, Power Query has dulled so many of the naysayers' claims that it should be considered the ultimate Excel myth buster.

Myth 1: "Excel is not reproducible"

The first one is the reproducibility. We talked about that.

Myth 1: "Excel can't handle over one million rows"

OK, now *this* one may be my favorite myth to bust. The Excel-averse often make it their trump card: we're in the age of "big data," right? Excel can't even go past ~1 million rows… that's peanuts these days! Cue the Excel half-truths now…

It *is* still true that an Excel workbook *itself* can only contain 1,048,576 rows. But this *does not* mean that you can't analyze more than a million rows in an Excel workbook!

How can this be? You guessed it — Power Query. Again, with this tool we can *extract* data from a variety of sources ranging from CSV files to databases and more. It's true that we can't *load* more than a million rows, but we are free to aggregate and summarize the data in Power Query, then load those results to Excel. And who wants to skim > 1 million rows, anyway?

For a more serious demonstration on how to blow past the alleged million-row Excel limit, check out this post from Master Data Analysis on analyzing **fifty million rows** using Excel Power Query.

And for something more humorous, check out my meme-video on the topic…

Orlando over at Master Data Analysis even has an example of analyzing 50 million records using Power Query.

Myth 2: "Excel has no true NULL"

If you've used relational databases you're familiar with the concept of a missing or NULL value. Missing values do not equal zero! We don't know why they are missing, they just are! But without a reserved keyword for NULLs, it's easy to mess up their storage and handling. Some users may keep them blank, others may hard-code a value like NA.

NULLs are also often helpful for conditional logic checks: if something doesn't meet some statement, we may want a truly missing value to be returned:

Whether it's a web page, Salesforce data, even tweets, Power Query greatly expands the type of data we can work with. I mean, look at all these things!

Again, this objection is true in classic Excel. But Power Query does feature a dedicated NULL, and you can use it as a conditional output. So another one bites the dust. Let's keep going, shall we?

Myth 3: "Excel only works with structured data"

Now this one I can be a little more sympathetic to, because in general Excel really isn't the best tool for working with unstructured data. That said, it's worth pointing out that Power Query does make it a little more possible and should expand our horizons about what Excel can do.

If you're not familiar with the idea of unstructured or its opposite, structured data, think of how spreadsheets are laid out — in rows and columns. Generally there are some patterns and rules you can use to determine what type of data goes in those rows and columns. This is structured data in a nutshell. Excel thrives in this land of rows and columns.

Then there is the messier data like text, images, audio and so forth. Not real simple to cram this into a spreadsheet, right?

Again, I agree that Excel shouldn't be the go-to solution when working with unstructured data.

That said, Power Query does make it possible to do these things. For example, we can Power Query to perform basic sentiment analysis or parse so-called "semi-structured" JSON data.

I hope these Power Query possibilities are getting you to "think outside the cell…"

As you progress with Power Query, think about what other Excel myths you're able to bust! Feel free to drop me a line with how it's going…

Conclusion and next steps

Congrats for taking your first steps using Excel Power Query! I hope you see the vast possibilities for automating away the boring stuff and focusing on the data tasks that matter.

Companion solution file

First of all, here is the worked-through download file to accompany this white paper. Were you able to follow all the twists and turns along with me? If you weren't, you know where to look to see every step applied… (Hint… it's the Applied Steps menu!)

First Steps in Power Query course

If you're excited for more, I invite you to take my First Steps in Power Query for Excel course:

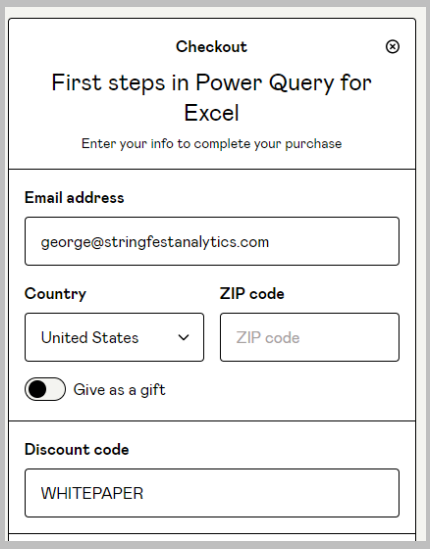
This course contains nearly 3 hours of video instruction on the basics of Power Query for Excel, along with in-depth demo notes, drills, datasets and more. Learn the fundamentals of Power Query in under half a day, then use these techniques day after day.

Promocode for course

As a special thank-you for reading this white paper, I'm offering **25% off the course** with promocode WHITEPAPER.

To redeem, click under "Buy this" on the Gumroad listing and enter under "Discount code."

Looking forward to seeing you in class!



Thank you

Thanks for picking up this white paper as a guide for adding Power Query to your Excel toolkit.

I invite you to continue reading my newsletter and perusing my blog for more analytics content. You're also welcome to get in touch if I can help your organization with the services listed here.

I offer a full-day Excel Power Query workshop to corporate teams up to 20 people in-person or 35 online.

One last thing: please give me a follow on Twitter and LinkedIn. Your support is appreciated.