七天入门 PowerBI

帮你轻松迈入 Powerbi 大门

采悟

注册会计师、知乎优秀回答者、
微信公众号「PowerBI 星球」作者

PowerBI 星球 | 出品

前言

本电子书主要精选自微信公众号的相关文章，分模块整理而成，适合刚接触 PowerBI 的新人作为入门学习材料使用，如果你已经入门了或者已经阅读过公众号中的每篇文章，完全可以跳过。

PowerBI 作为一门实用型的技能，不建议一开始就去学习大量的函数公式、计算逻辑等理论性的知识，我的建议是先快速入门，熟悉基本的概念和操作，了解常用函数的用法，然后就可以在自己的工作中实际操练，遇到问题随时请教、交流沟通，在实践中理解背后的逻辑，不断提升，实践获得的技能，为更进一步的学习打下基础，层层深入，这是最快速也是最有效的学习方法。

这本电子书就是带你快速入门的秘籍，其中涉及到软件的安装与使用、数据清洗和数据建模的基本知识，可视化的制作、交互的技巧等，并搭配有练习数据（详见部分文章底部数据下载方式），可以在阅读文章的同时动手操作练习，每天花费两三个小时，七天轻松入门。

现在就开始你的 PowerBI 学习之旅吧。

目录

【第 1 天 | 认识并体验 PowerBI】 3
认识 Power BI 3
PowerBI 与 PowerQuery、PowerPivot 到底是什么关系？ 10
体验 Power BI | 分分钟生成一份可视化图表 15
【第 2 天 | 熟悉 Powerquery】 24
使用 Power Query 是一种什么体验？ 24
数据获取-PoweBI 支持从多种数据源导入数据 33
PowerQuery 常用的数据清洗十三招 36
Power Query 数据丰富 50
认识 M 函数 58
【第 3 天 | 数据建模的基本知识】 63
Power BI 数据建模 63
理解度量值 70
DAX 是什么？ 77
【第 4 天 | 了解常用的 DAX 函数】 82
PowerBI 中最重要的函数 CALCULATE 82
FILTER 函数 89
FILTER 语法 89
FILTER 应用 89
HASONEVALUE 和 SUMX 函数 92
EARLIER 函数 97
【第 5 天 | 了解可视化的制作】 102
使用 PowerBI 制作柱形图 102
树状图 110
地图可视化 115
【第 6 天 | PowerBI 的常用操作】 125
查看图表背后数据的三种方式 125
图表的钻取 127
编辑交互 132
【第 7 天 | 初学者经常遇到的几个问题】 137

【第 1 天 | 认识并体验 PowerBI】

首先来安装 **PowerBI Desktop** 并熟悉它的基本界面，搞清楚 **PowerBI** 与两大核心组件 **Power query** 和 **Power pivot** 是什么关系，并尝试着用它做出一份简单的报表。

认识 **Power BI**

本文简单介绍什么是 Power BI、如何下载安装 PowerBI Desktop 以及 PowerBI 的界面和操作的一般步骤，现在就来迈出我们学习 PowerBI 的第一步吧。

Power BI 是什么?

Power BI 是微软推出的数据分析和可视化工具，我们先来看看微软官方是怎么介绍的：

Power BI 是一套商业分析工具，用于在组织中提供见解。可连接数百个数据源、简化数据准备并提供即席分析。生成美观的报表并进行发布，供组织在 Web 和移动设备上使用。每个人都可创建个性化仪表板，获取针对其业务的全方位独特见解。在企业内实现扩展，内置管理和安全性。

简单来说就是可以从各种数据源中提取数据，并对数据进行整理分析，然后生成精美的图表，并且可以在电脑端和移动端与他人共享的一个神器。

PowerBI 包含桌面版 Power BI Desktop、在线 Power BI 服务和移动端 Power BI 应用，我们学习的时候使用 PowerBI Desktop 已足够，这个是 PowerBI 的精髓技能，简单来说 PowerBI

服务和移动端不过是 PowerBI 的分发共享功能而已，好消息是，PowerBI Desktop 还完全免费。

如何下载安装 PowerBI Desktop?

登录微软 PowerBI 主页：https://powerbi.microsoft.com/zh-cn/，

从产品中选择 Power BI Desktop:

下载的时候选择高级下载选项，可以选中文版本（如果英语不错，也可以直接下英文版本的），根据电脑的操作系统选择 32 位或者 64 位的安装包。

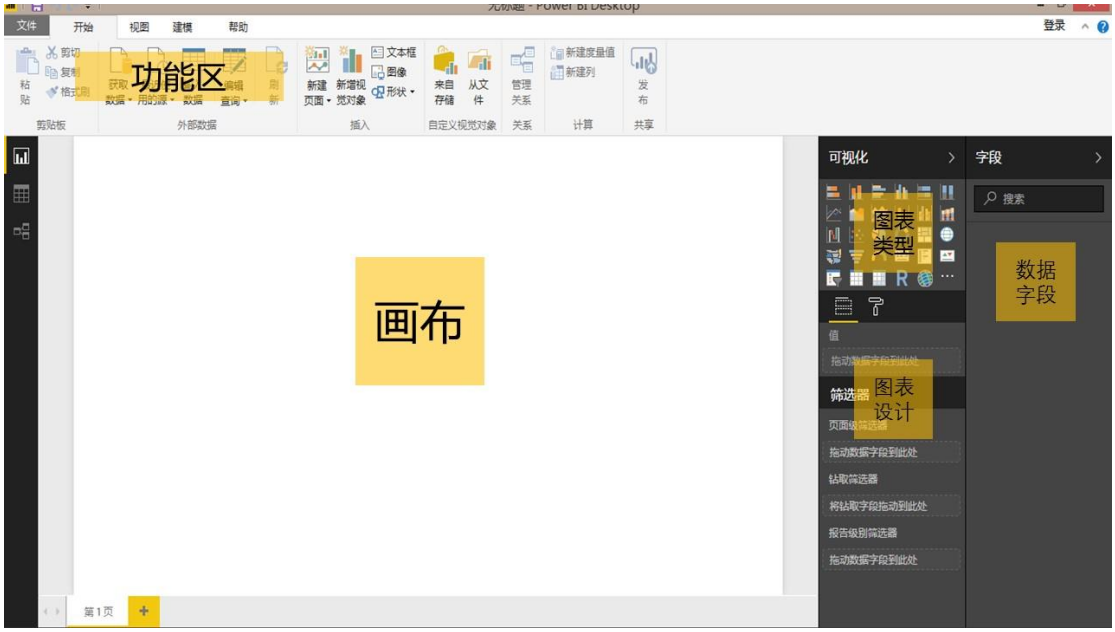
如果是 WIN10 系统，还可以直接在微软 store 里面找到 Power BI Desktop 应用直接安装。

安装完成，启动后会提示你注册登录，暂时不想注册直接关掉就行，如果注册，需要使用

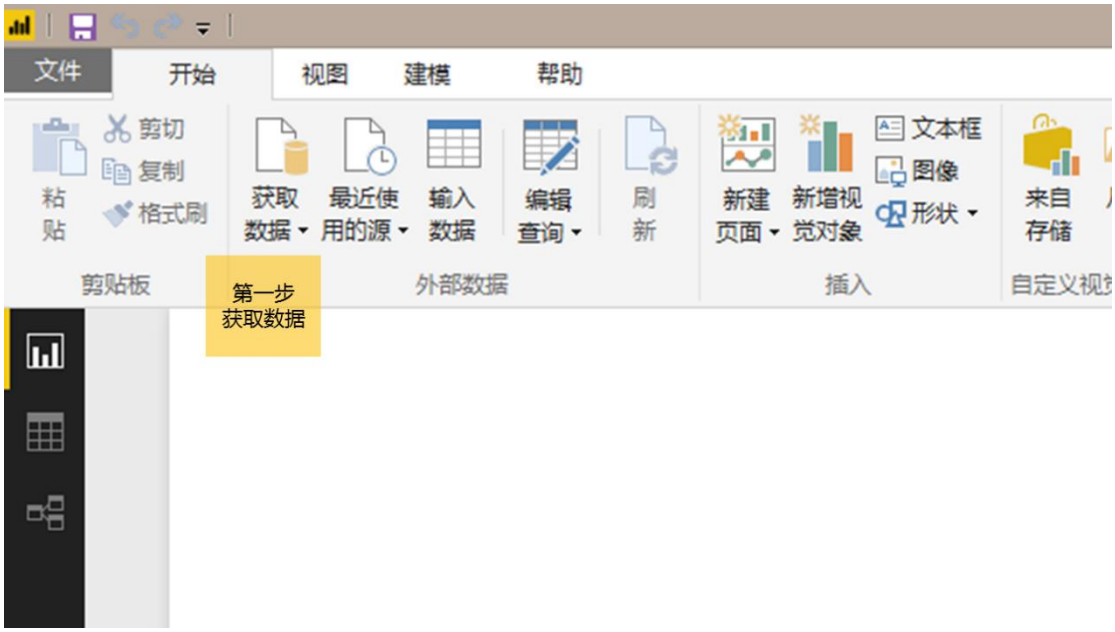
企业邮箱，现在注册还可以享受 2 个月的 PowerBI 专业版体验。

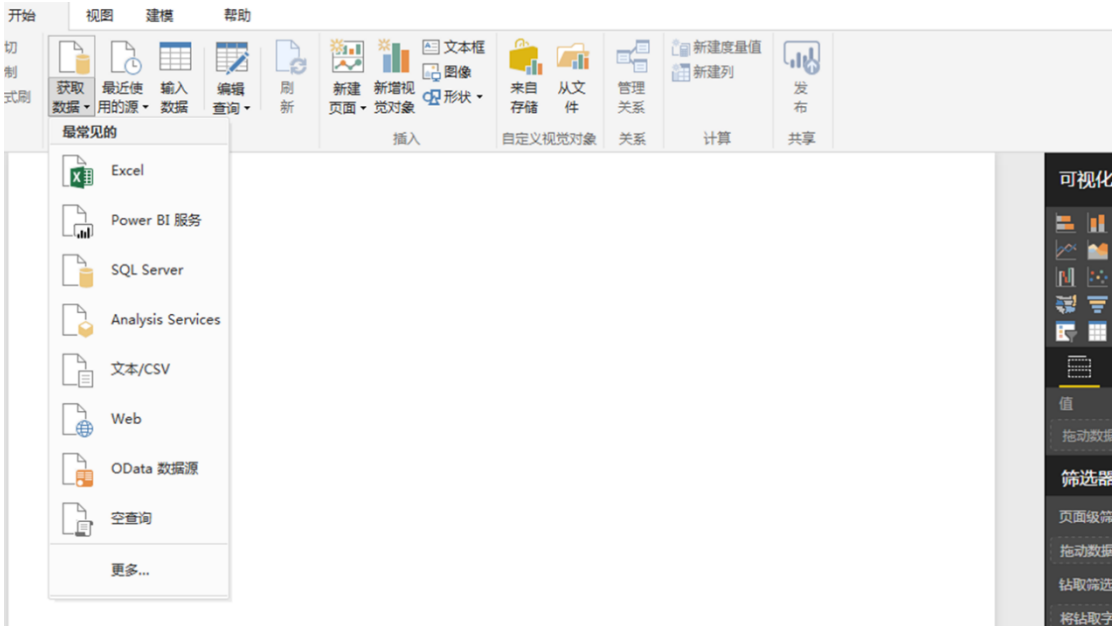
如何使用 Power BI?

进去后界面如下：

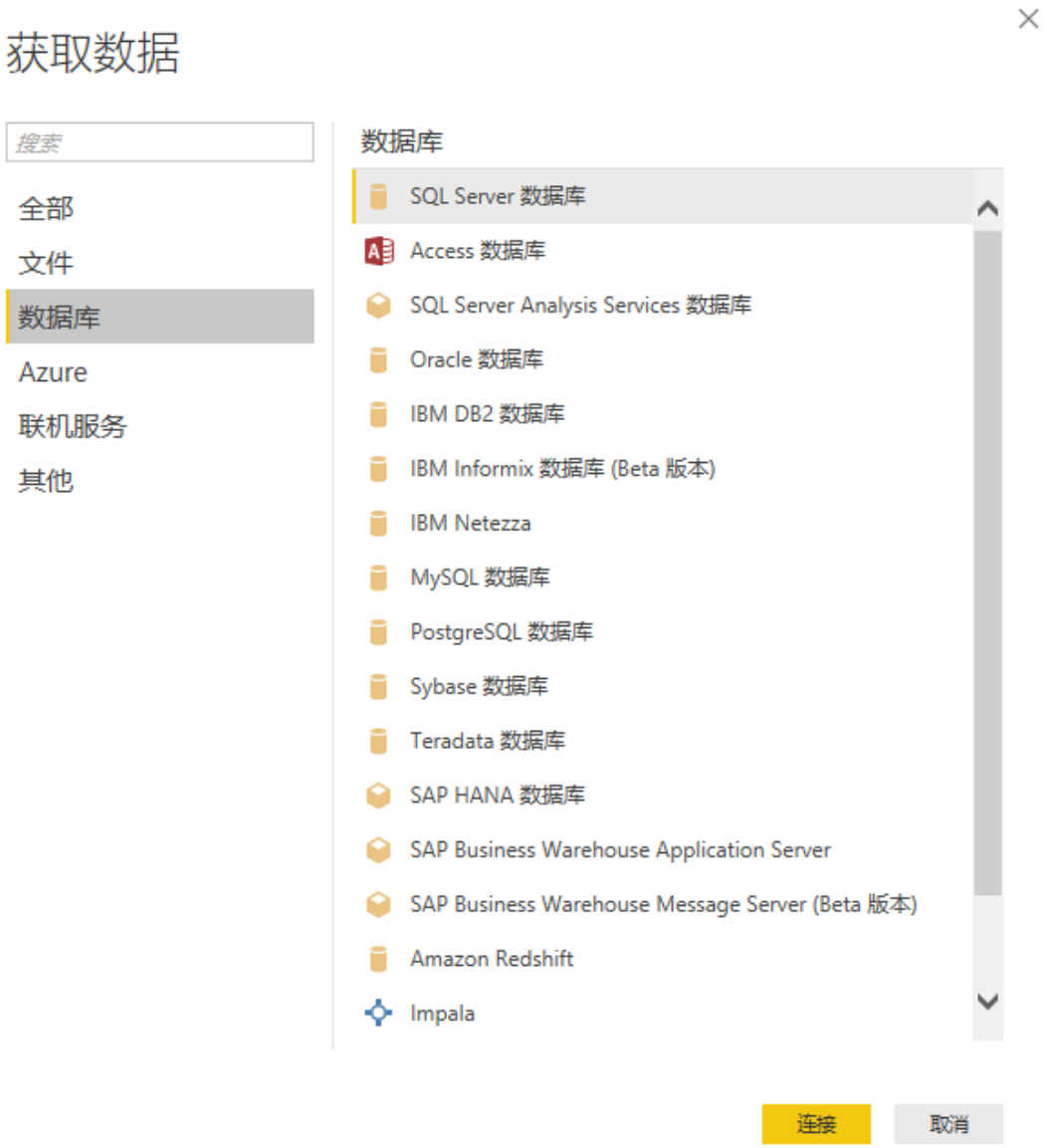


数据处理的第一步就是获取外部数据：



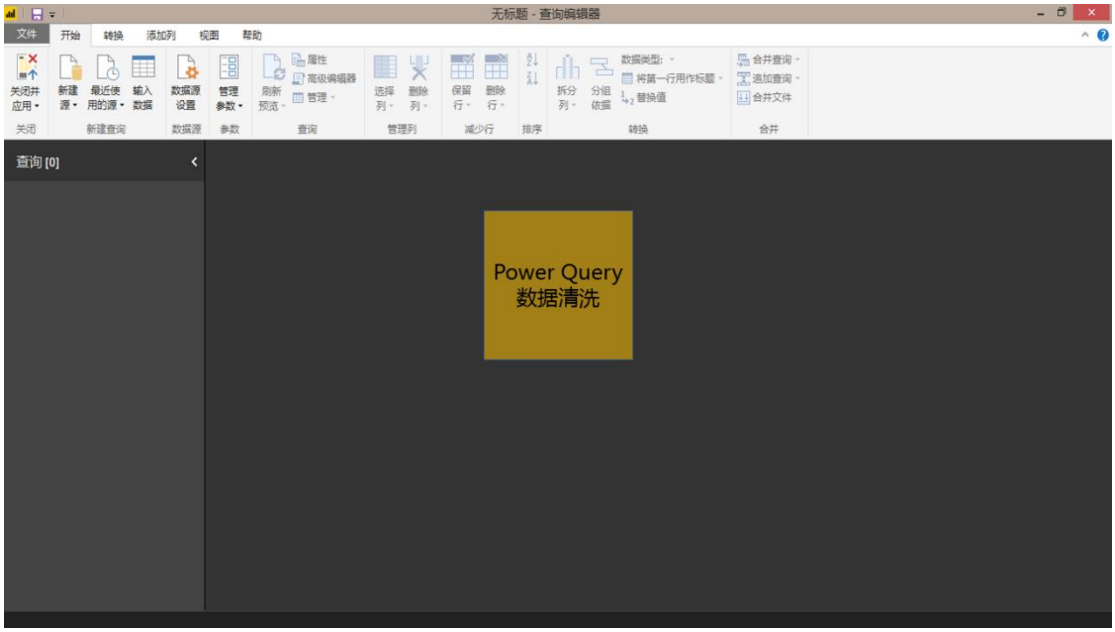


常用的数据格式如 excel、sql、csv/txt 出现在最上面，如果是其他格式的，点击更多，

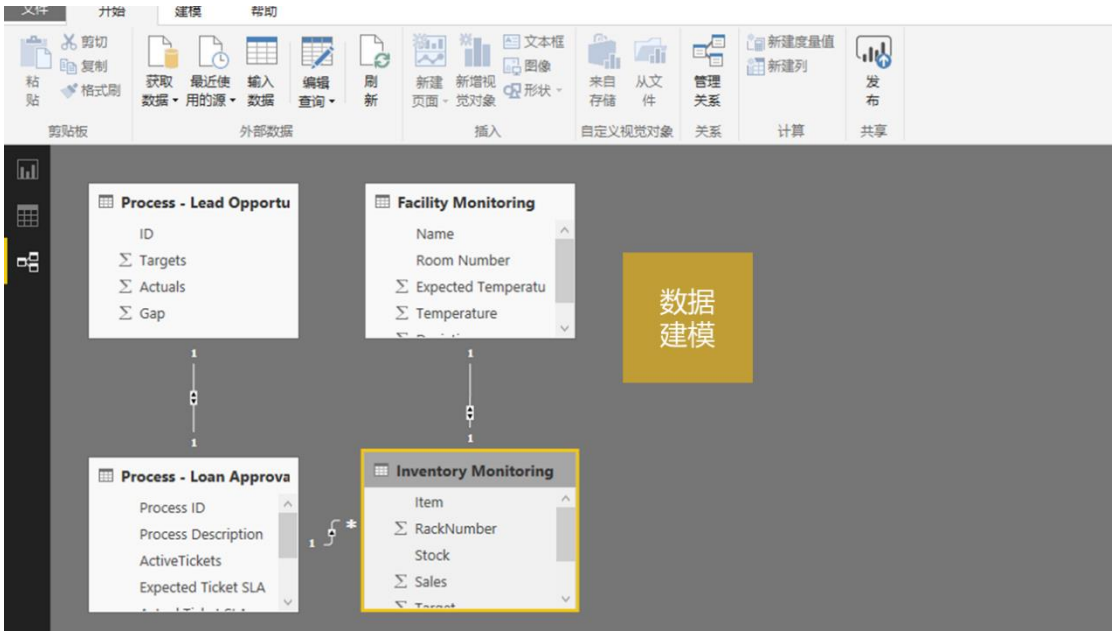


出来很多可选项，基本上目前所有能见到的数据格式都可直接导入。

数据导入后，可以进入内嵌的查询编辑器，这是 Power BI 的一个主要模块，也可以称为 Power Query，数据整理都在这里完成，后期会专门介绍。

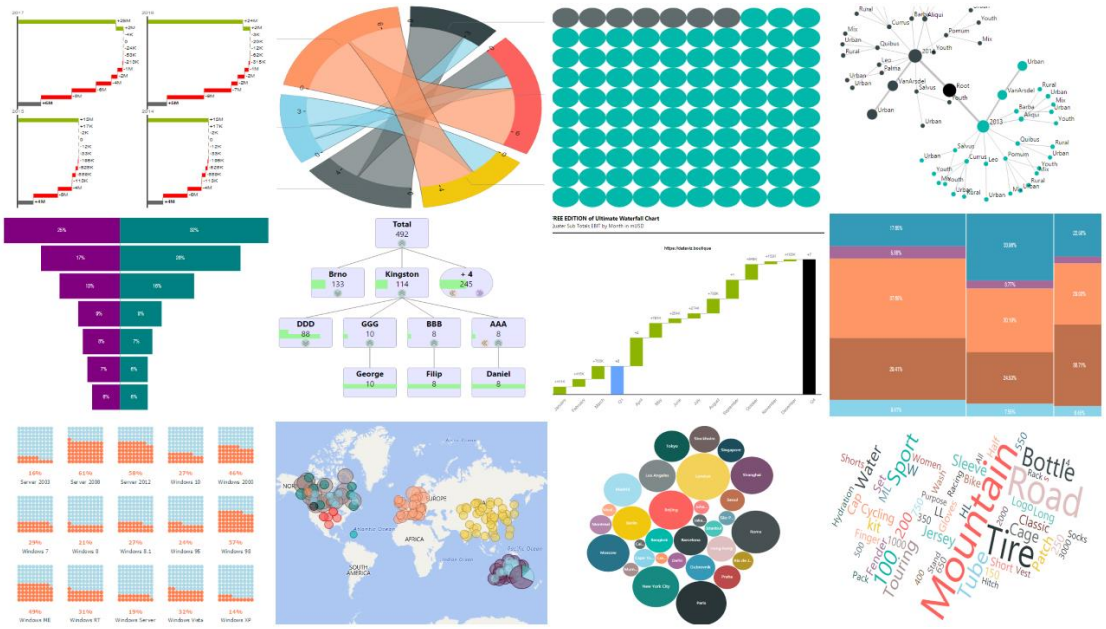


整理后的数据进入数据建模，不要被建模这个词吓着了，其实就是表格之间建立关联，这是我们下一步数据可视化的基础，当然如果只有一个表，就没有必要经过这一步了。



然后就可以进行可视化工作了，来看看可视化的组件：

这里显示的是系统默认自带的图表类型，平时常见的图表都有，另外推荐 PowerBI 一大理由就是还可以加载更多的自定义可视化包，各种酷炫图表，比如这些：



目前已经有 100 多种自定义图表可供免费下载使用，还在不断增加中……

PowerBI 与 PowerQuery、PowerPivot 到底是什么关系?

PowerBI 星球

搞不清楚 Power BI 与 Power Query、Power Pivot 是什么关系？看这篇文章就够了。

刚开始学习 PowerBI 的时候，总是能碰到 Power Query 和 Power Pivot 这两个词（下文简称为 PQ 和 PP)，现在中文里面学习 PowerBI 的资源本来就不是很多，大部分资源还都是介绍 PQ 和 PP 的，那么她们到底和 PowerBI 是什么关系呢?

微软的很多办公工具都是以 Power 开头，最熟悉的当然就是 PowerPoint 了，如果 ppt 可以直译为超级演示，PQ 就是超级查询，PP 就是超级透视，我们先来看一下 PQ。

Power Query

用作数据处理的大众化软件就是万人皆知的 Excel 了，Excel 作为日常办公使用当然没有问题，但在大数据时代，她明显有点扛不住，微软也意识到了这一点，所以从 Excel2010 开

始，推出了一个叫 Power Query 的插件，可以弥补 Excel 的不足，处理数据的能力边界大大提升，Excel2013 也同样可以使用，现在还在用 Excel2010 和 2013 的同学可以从微软官网下载 powerquery 插件使用。

而到了 Excel2016，微软直接把 PQ 的功能嵌入进来，放在数据选项卡下：

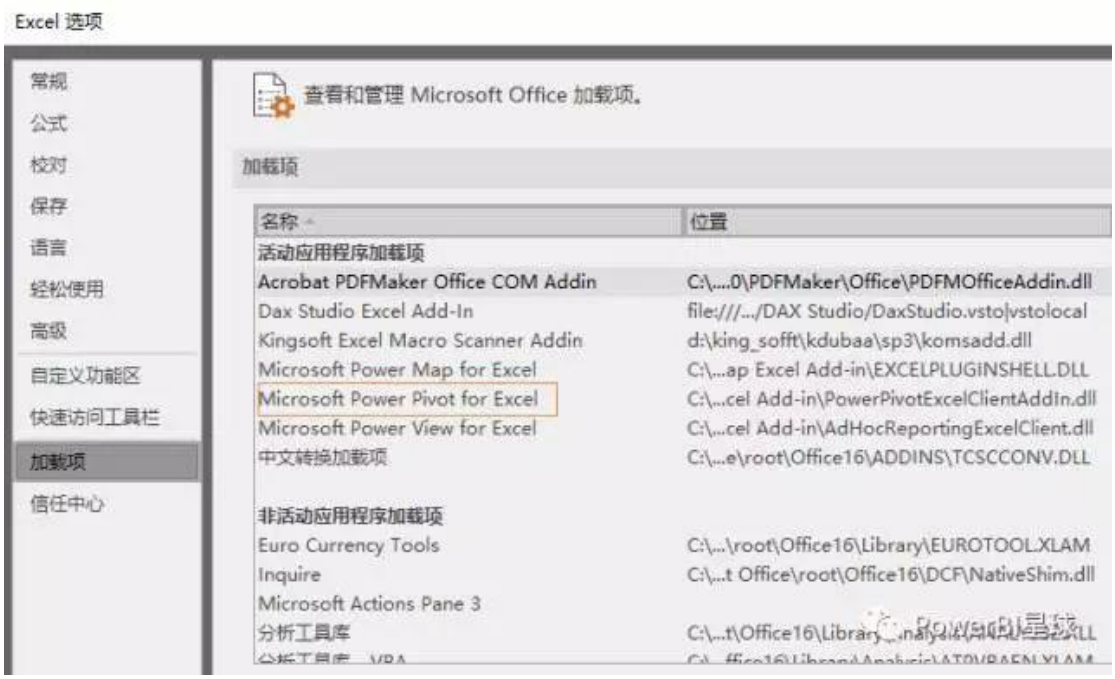
PowerBI 中的获取数据界面是这样的：

是不是非常相似，功能也基本是一样，点击进去后都是进入查询编辑器，所用的也都是 M 语言，所以学习 PP 就是学习 PowerBI 中的数据处理模块，无论在 Excel 中学还是在 Power BI 中学，都是一样的。

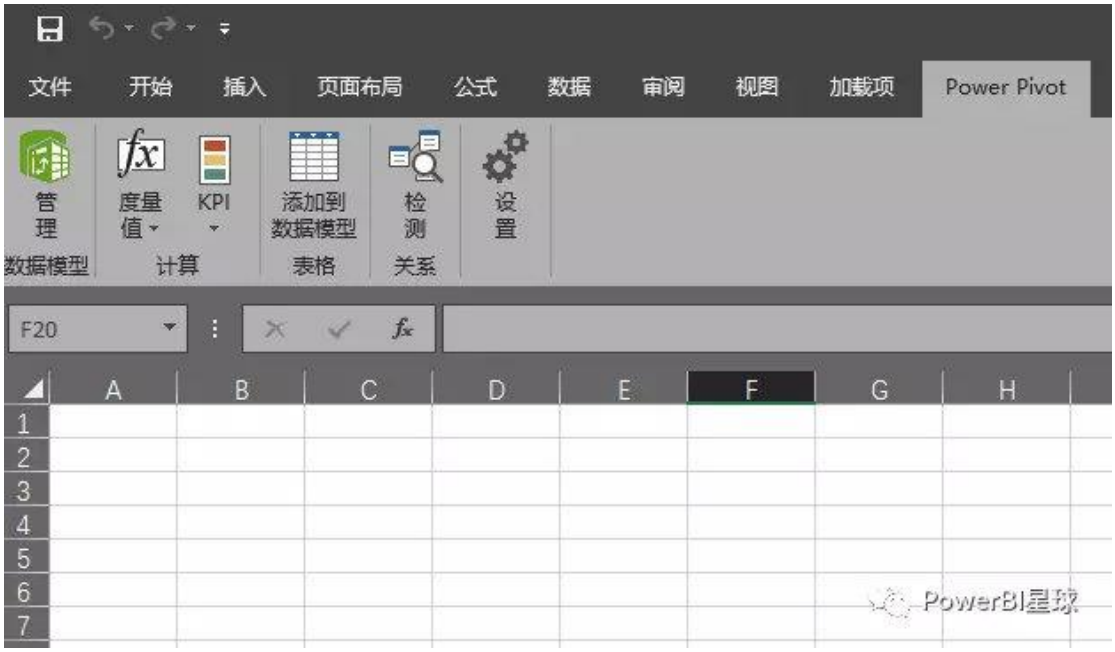
Power Pivot

接触过 Excel 的人肯定都知道数据透视表，英文名是 Pivot Table，按这个翻译 PP 可以叫做超级透视，但其功能要比数据透视表强大很多，所以 PP 被大家称为是数据建模，这个名字一下就显得高大上了吧，不过 PP 确实名副其实，她被称为 Office 软件中近 20 年来最伟大的发明，也是 PowerBI 的灵魂，PP 用到的语言是 DAX，以后会详细介绍。

在 Excel 中也可以使用 PP,首先从选项里面把这个功能加载进来：



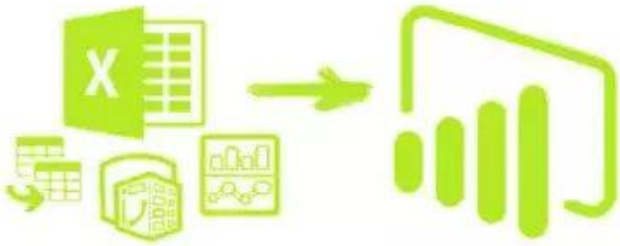
然后 Excel 选项卡下就多了一个 Power Pivot，界面如下：



这个和 PowerBI 中建模选项卡的功能区也非常相似，所以学习 PP 就是学习 Power BI 的数据建模，二者的本质内容是一致的。

刚才看 Excel 选项中加载项的时候，我们看到 Power Pivot 旁边还有两个 Power 兄弟，Power View 和 Power Map，PV 就是数据可视化，PM 就是数据地图，这两项也已经内嵌到 PowerBI 中，且功能更加强大。这两个学习都相对比较简单，就不作介绍，我们学习 PowerBI 的重点就是数据处理和数据建模，学好这两个以后，数据可视化就是水到渠成而已。

从上面的介绍可以看出，**Power Query**、**Power Pivot**、**Power view** 以及 **Power Map** 等全部功能聚集到一起，就成了现在的 **Power BI**。



PowerBI星球

PowerBI

Excel or PowerBI Desktop?

- 既然在 Excel 和 PowerBI Desktop 中都可以学习最核心的组件 PP 和 PQ，那么在哪里学习更好呢，其实都可以，看个人的使用习惯。我个人更推荐直接在 PowerBI Desktop 中学习，理由如下：

- PowerBI Desktop 界面更友好，逼格更高
- PowerBI Desktop 更新速度快，几乎每月都有更新，最新的功能、新的 M 函数和 DAX 函数随时可以调用
- 进行数据处理的最终可能还要通过可视化来展现，发现有趣的见解，这在 PowerBI Desktop 中整个流程一气呵成，且图表库和便捷性要完爆 Excel
- 最后一个原因，对很多人也非常重要的是，PowerBI Desktop 完全免费，而 Excel 需要正版才能体现非一般的感觉

当然主要还是看你的实际需要，如果只是日常的数据处理，比如财务工作中处理各种表格，在 Excel 中完全没有问题的哦。

体验 **Power BI |** 分分钟生成一份可视化图表

微软宣称 Power BI 只要几分钟就可以将数据转化为精美的图表，要不要尝试一下？

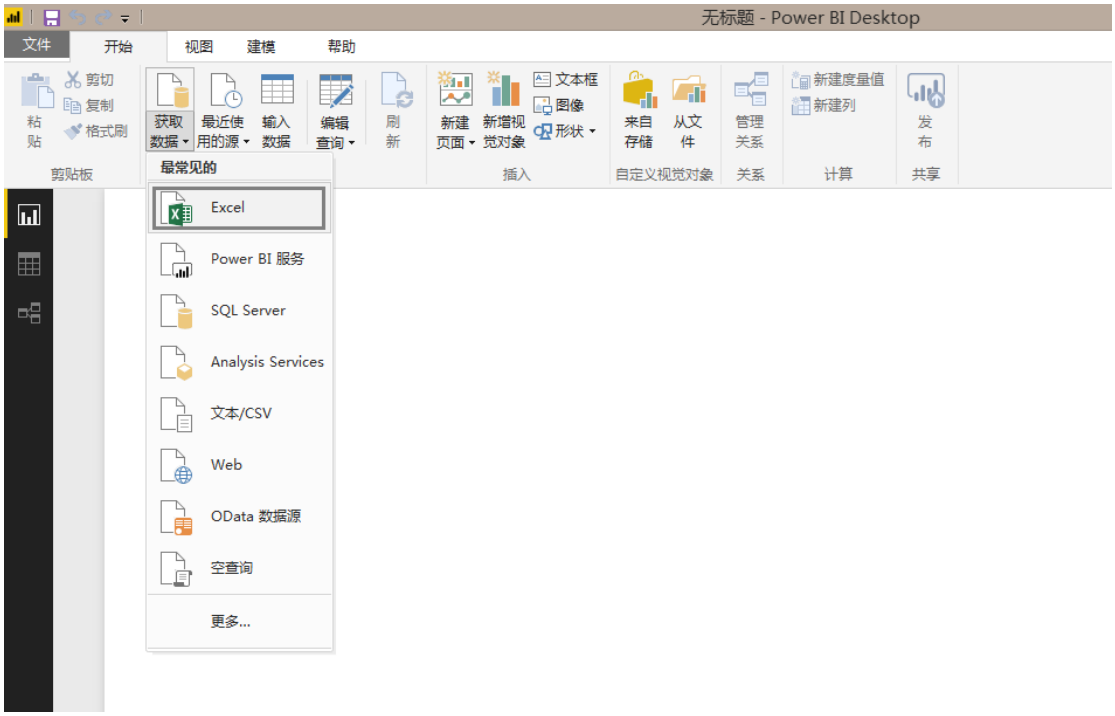
首先我们准备一份数据，Excel 格式

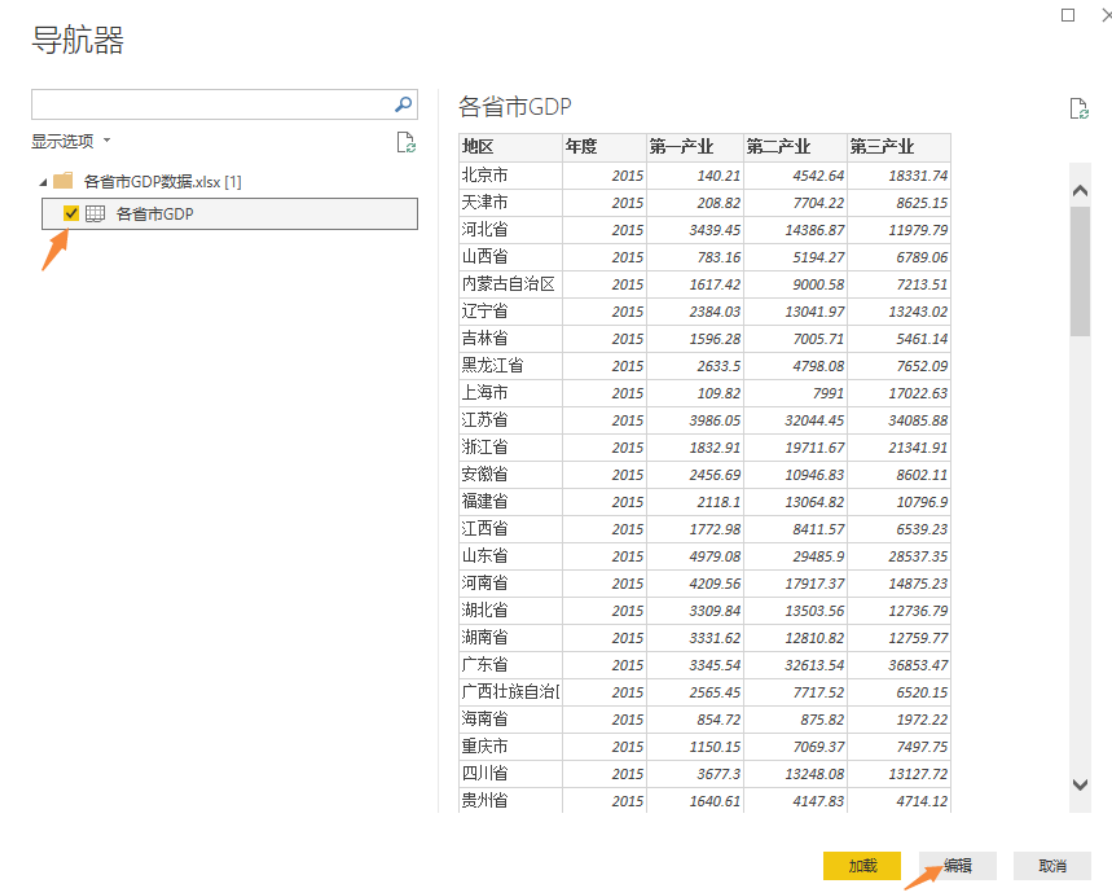
	A	B	C	D	E	F
1	地区	年度	第一产业	第二产业	第三产业	合计（亿元）
2	北京市	2015	140.21	4542.64	18331.74	23014.59
3	天津市	2015	208.82	7704.22	8625.15	16538.19
4	河北省	2015	3439.45	14386.87	11979.79	29806.11
5	山西省	2015	783.16	5194.27	6789.06	12766.49
6	内蒙古自治区	2015	1617.42	9000.58	7213.51	17831.51
7	辽宁省	2015	2384.03	13041.97	13243.02	28669.02
8	吉林省	2015	1596.28	7005.71	5461.14	14063.13
9	黑龙江省	2015	2633.5	4798.08	7652.09	15083.67
10	上海市	2015	109.82	7991	17022.63	25123.45
11	江苏省	2015	3986.05	32044.45	34085.88	70116.38
12	浙江省	2015	1832.91	19711.67	21341.91	42886.49
13	安徽省	2015	2456.69	10946.83	8602.11	22005.63
14	福建省	2015	2118.1	13064.82	10796.9	25979.82
15	江西省	2015	1772.98	8411.57	6539.23	16723.78
16	山东省	2015	4979.08	29485.9	28537.35	63002.33
17	河南省	2015	4209.56	17917.37	14875.23	37002.16
18	湖北省	2015	3309.84	13503.56	12736.79	29550.19
19	湖南省	2015	3331.62	12810.82	12759.77	28902.21
20	广东省	2015	3345.54	32613.54	36853.47	72812.55
21	广西壮族自治	2015	2565.45	7717.52	6520.15	16803.12
22	海南省	2015	854.72	875.82	1972.22	3702.76

各省市GDP

数据是从 2006 年到 2015 年 10 年间，中国大陆 31 个省市的三个产业的 GDP,我们用 Power BI 来看看这三个产业结构近年来有什么趋势？

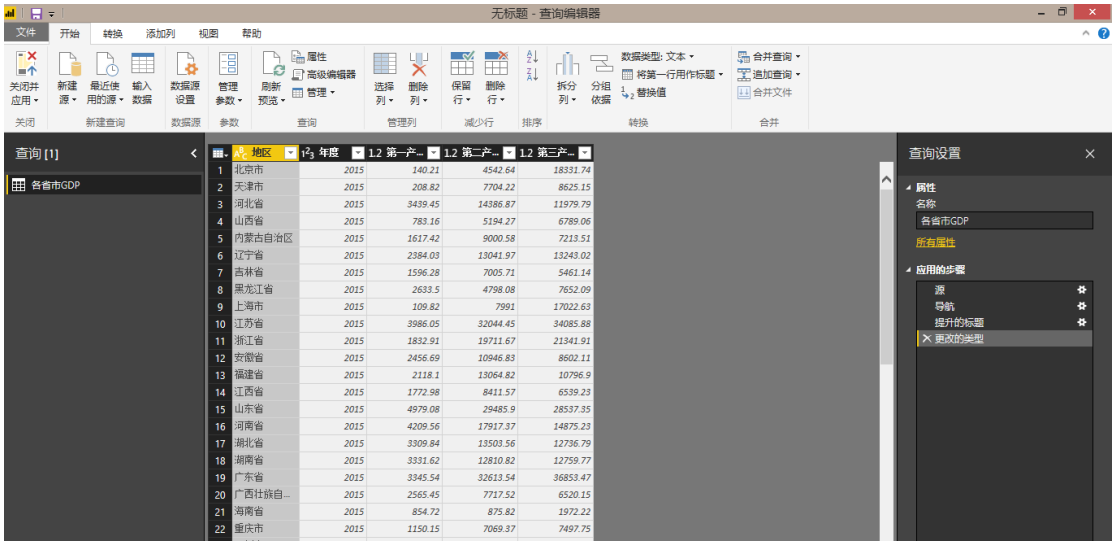
首先第一步，获取数据，选择 Excel 格式导入：



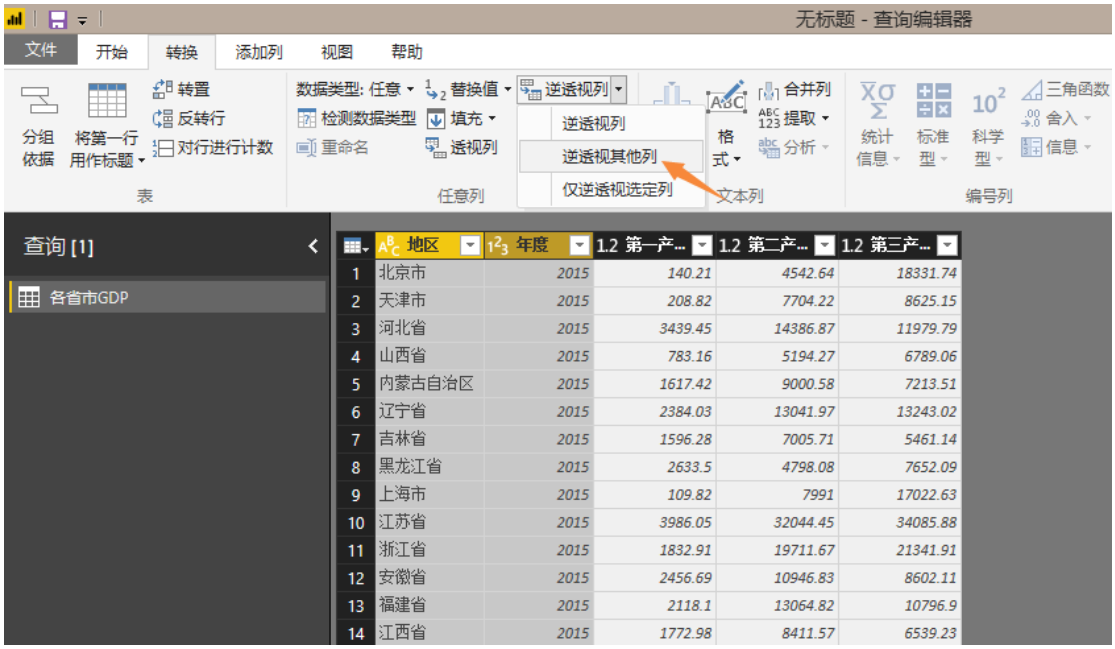


地区	年度	第一产业	第二产业	第三产业
北京市	2015	140.21	4542.64	18331.74
天津市	2015	208.82	7704.22	8625.15
河北省	2015	3439.45	14386.87	11979.79
山西省	2015	783.16	5194.27	6789.06
内蒙古自治区	2015	1617.42	9000.58	7213.51
辽宁省	2015	2384.03	13041.97	13243.02
吉林省	2015	1596.28	7005.71	5461.14
黑龙江省	2015	2633.5	4798.08	7652.09
上海市	2015	109.82	7991	17022.63
江苏省	2015	3986.05	32044.45	34085.88
浙江省	2015	1832.91	19711.67	21341.91
安徽省	2015	2456.69	10946.83	8602.11
福建省	2015	2118.1	13064.82	10796.9
江西省	2015	1772.98	8411.57	6539.23
山东省	2015	4979.08	29485.9	28537.35
河南省	2015	4209.56	17917.37	14875.23
湖北省	2015	3309.84	13503.56	12736.79
湖南省	2015	3331.62	12810.82	12759.77
广东省	2015	3345.54	32613.54	36853.47
广西壮族自治[	2015	2565.45	7717.52	6520.15
海南省	2015	854.72	875.82	1972.22
重庆市	2015	1150.15	7069.37	7497.75
四川省	2015	3677.3	13248.08	13127.72
贵州省	2015	1640.61	4147.83	4714.12

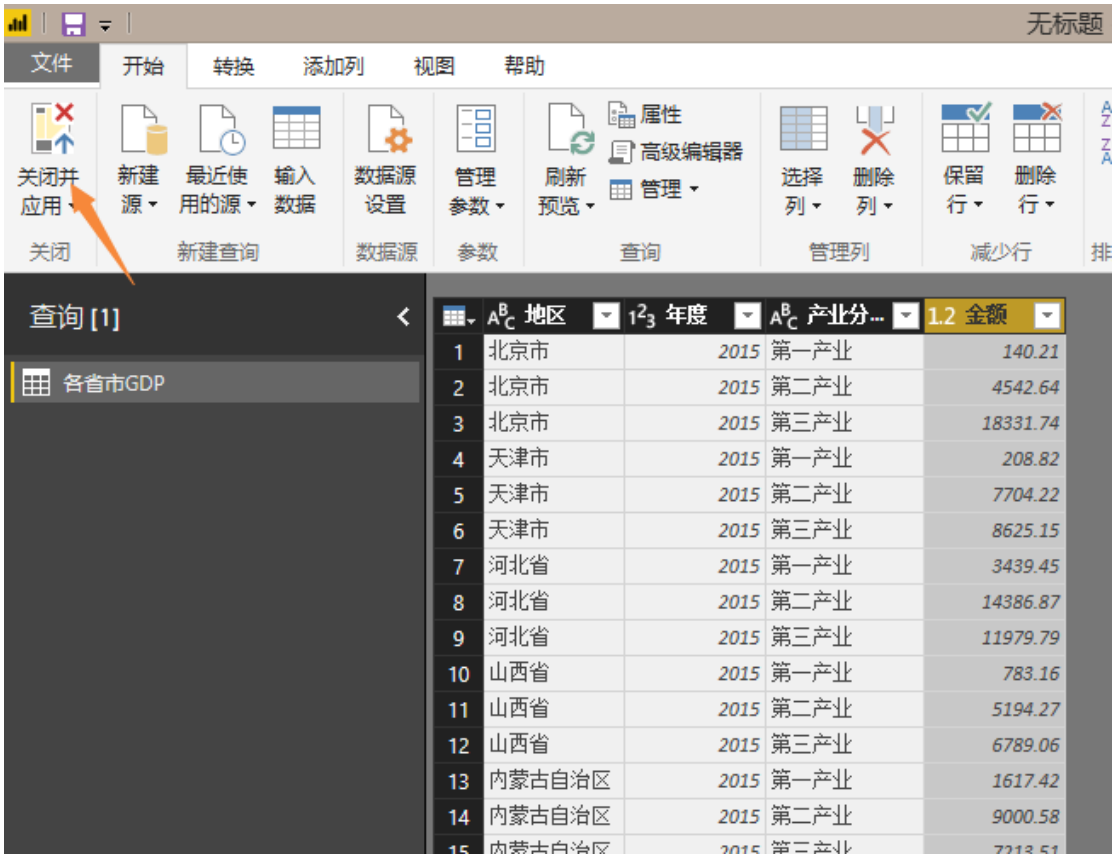
这里选择编辑，进入查询编辑器，



这个表格是二维表，为了分析的方便，需要把二维表转化为一维表，这个操作在 Power BI 里非常简单，这里把三个产业结构的数据转化为一个字段，选中地区和年度列，从转换里找到逆透视其他列：



然后就成了一维表了，把最后两列的名称重命名一下，点击关闭并应用：

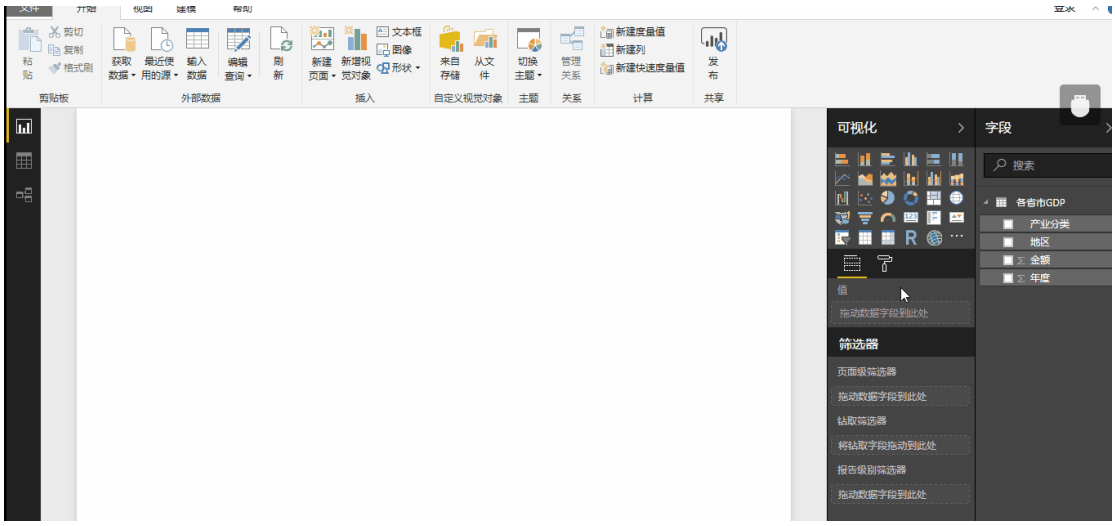


	地区	年度	产业分...	金额
1	北京市	2015	第一产业	140.21
2	北京市	2015	第二产业	4542.64
3	北京市	2015	第三产业	18331.74
4	天津市	2015	第一产业	208.82
5	天津市	2015	第二产业	7704.22
6	天津市	2015	第三产业	8625.15
7	河北省	2015	第一产业	3439.45
8	河北省	2015	第二产业	14386.87
9	河北省	2015	第三产业	11979.79
10	山西省	2015	第一产业	783.16
11	山西省	2015	第二产业	5194.27
12	山西省	2015	第三产业	6789.06
13	内蒙古自治区	2015	第一产业	1617.42
14	内蒙古自治区	2015	第二产业	9000.58

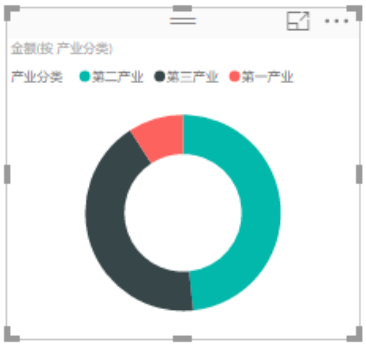
数据上载后，就可以在字段区看到这张表的字段，

到这里数据已经整理完成，开始生成图表。

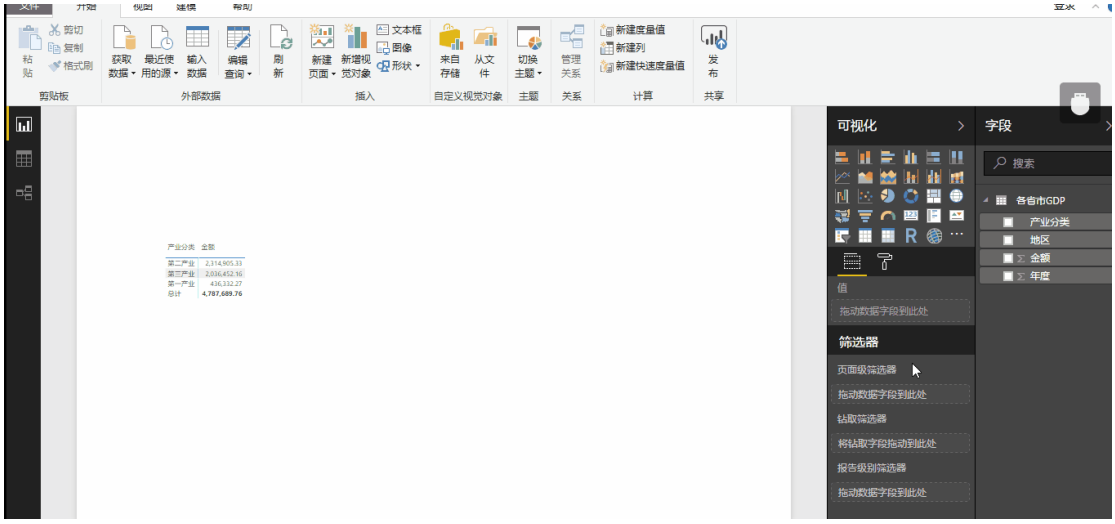
为了直观看出各个产业的数据，我们先做一个产业结构数据的矩阵表：



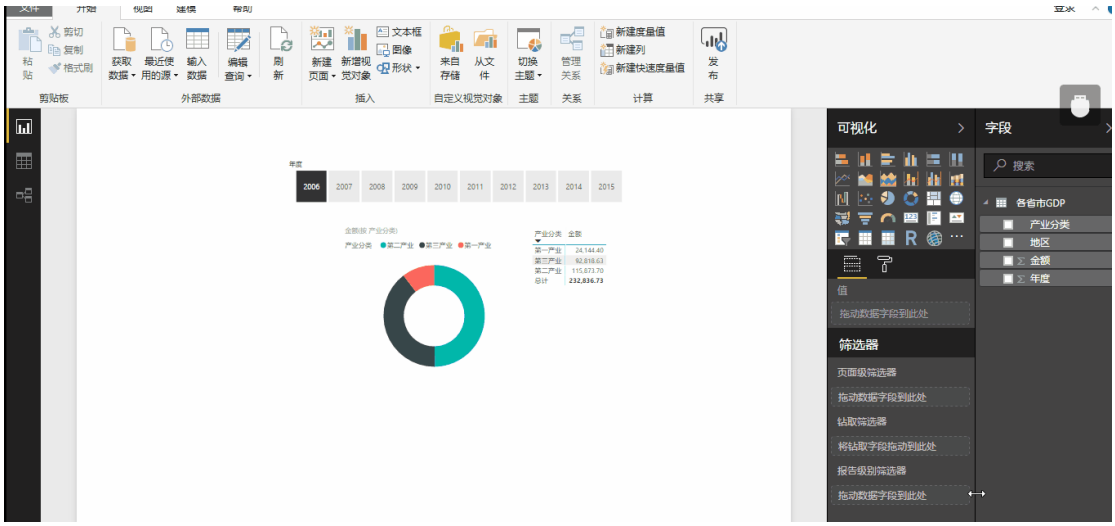
是不是很简单，就是把相应的字段拖进图表编辑框中而已，然后同样拖拽字段生成圆环图：



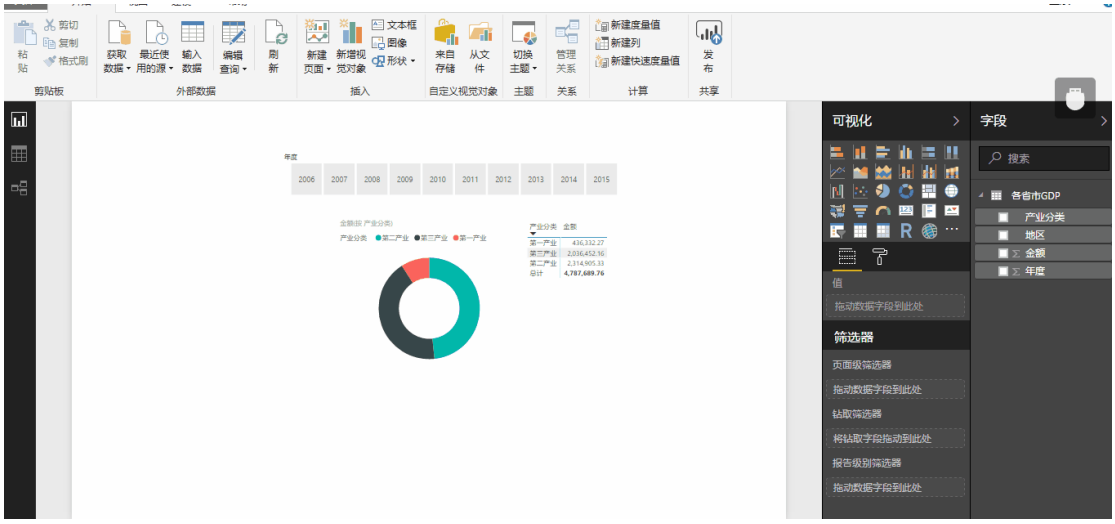
这两个表都是按产业结构分类的，并没有把年度指标考虑进来，下面就来看看如何把年度放进来：



实际上是做了一个切片器，通过点击年份的切片就可以控制另外两个图表的数据，这就是图表间的交互功能，我们先看一下效果：

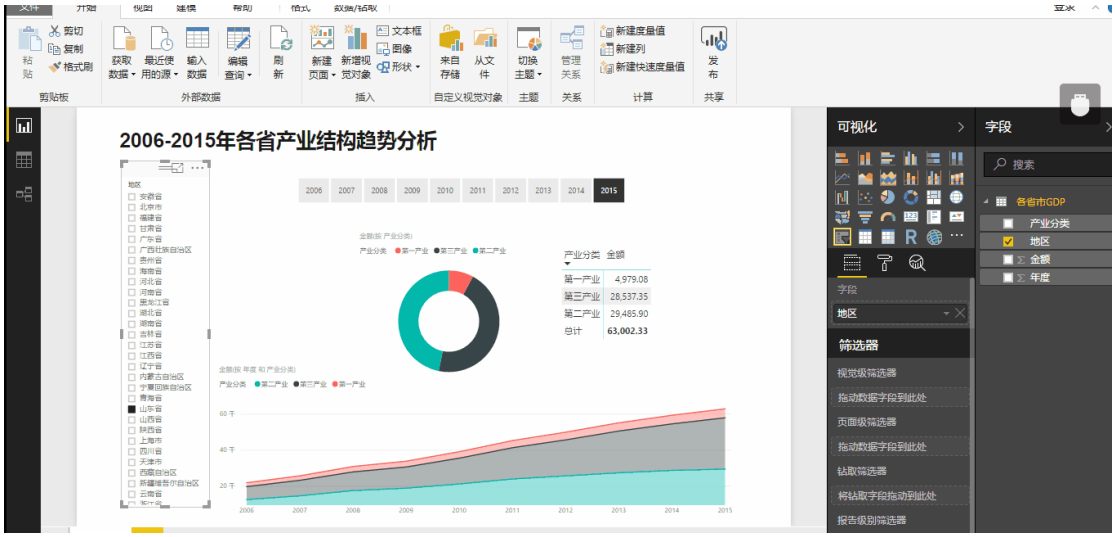


不仅要看每年的数据，还想看这十年来的趋势，那么就放个堆积面积图：



从这里可以看出 GDP 的增长主要来自于第二产业和第三产业规模的扩大，第一产业的规模基本没什么变化，占比逐年降低。

原始数据是大陆 31 个省市的产业数据，到现在为止还没有把省份纬度放进来。和刚才的年度切片器一样，再根据地区做一个各省市的切片器，然后在画布上拖拽调整对齐这几个图表的位置，并在上方写个标题，这套图表就完成了，我们来看一下效果：



是不是很快，如果操作熟练，5 分钟左右就可以生成这样一个可视化图表，微软所言果然不虚。这个图表虽然看起来还有很多地方需要进一步完善，但这个效果已经超过大部分人用传统做法耗费一两个小时的成果。

至此，我们体验了 PowerBI 把数据通过简单点击几下鼠标就生成图表的整套流程，这速度简直要上天了。

Power BI 还不止于此，这个图表我们还可以进一步分享给其他人查看，点击发布，将该图表发布到 PowerBI 网页版，

然后在网页版中选择发布到 web：

就会生成一串网址，把这个网址发给接收的人员，他们就可以随时随地、通过手机平板都

可以查看这个报表了，并且这个报表不是静态的，也是可以点击交互的，

这份图表的网址如下，也可以复制到浏览器器重查看：

http://t.cn/RDkbuwo

看到这里感觉 PowerBI 怎么样，是不是有种相见恨晚的感觉呢，练习数据可通过微信公众号后回复关键字“体验 PowerBI”获取。

【第 2 天 | 熟悉 Powerquery】

熟悉 **Power query** 进行数据处理、**PQ** 的最常用功能和操作，并对 **M** 函数有个基本的了解

使用 **Power Query** 是一种什么体验?

你和 Excel 高手的距离只差了一个 Power Query!!!

先思考一个日常工作中常遇到的问题：如何将多个工作簿的数据合并到一张表上?

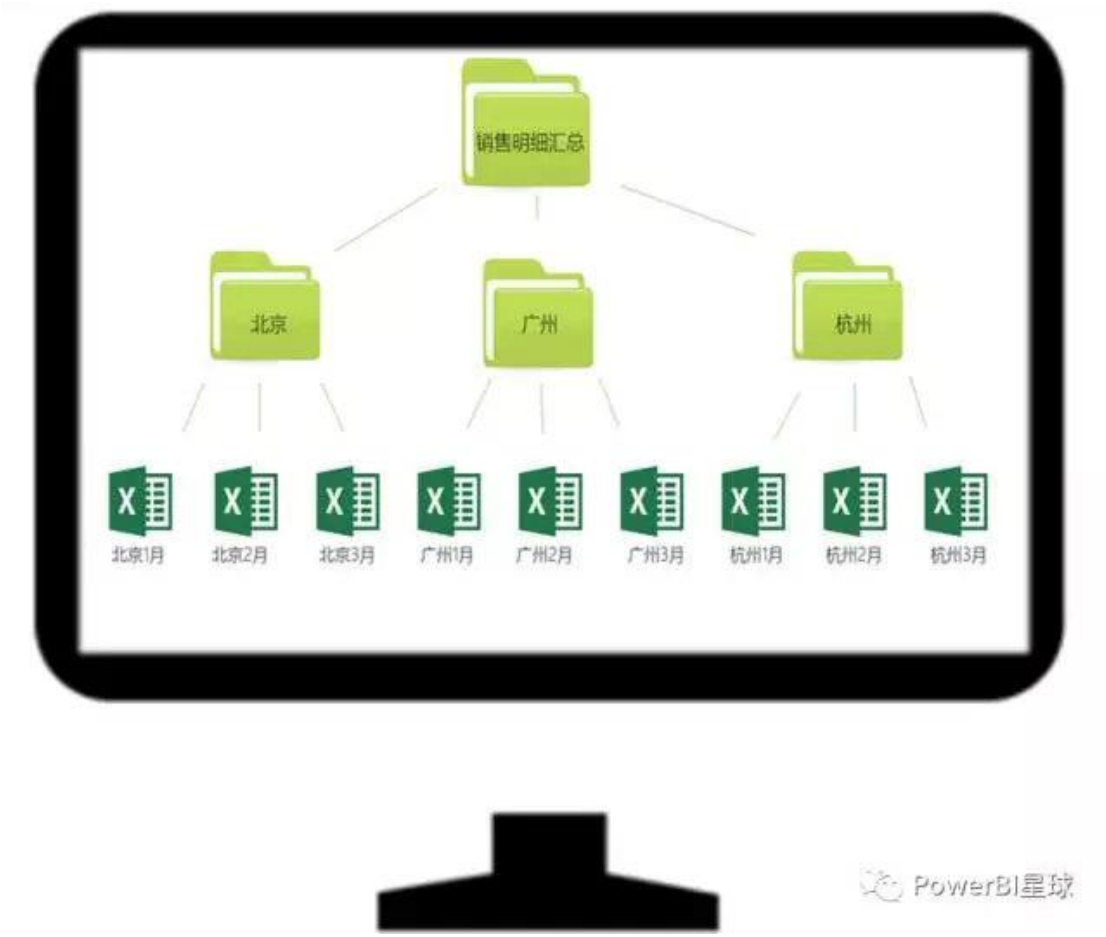
可能不同的人有不同的做法?

普通青年用万能的复制粘贴
二逼青年网上百度 VBA 代码一键汇总
文艺青年找个崇拜自己的实习小 MM 帮忙

其实都不必这么麻烦，我们无需借助高级的数据处理软件，无需学习复杂的 VBA 语言，无需挖空心思找别人帮忙，超级强大的工具就在我们身边，就在我们天天使用的 Excel 里面，那就是 Power Query！

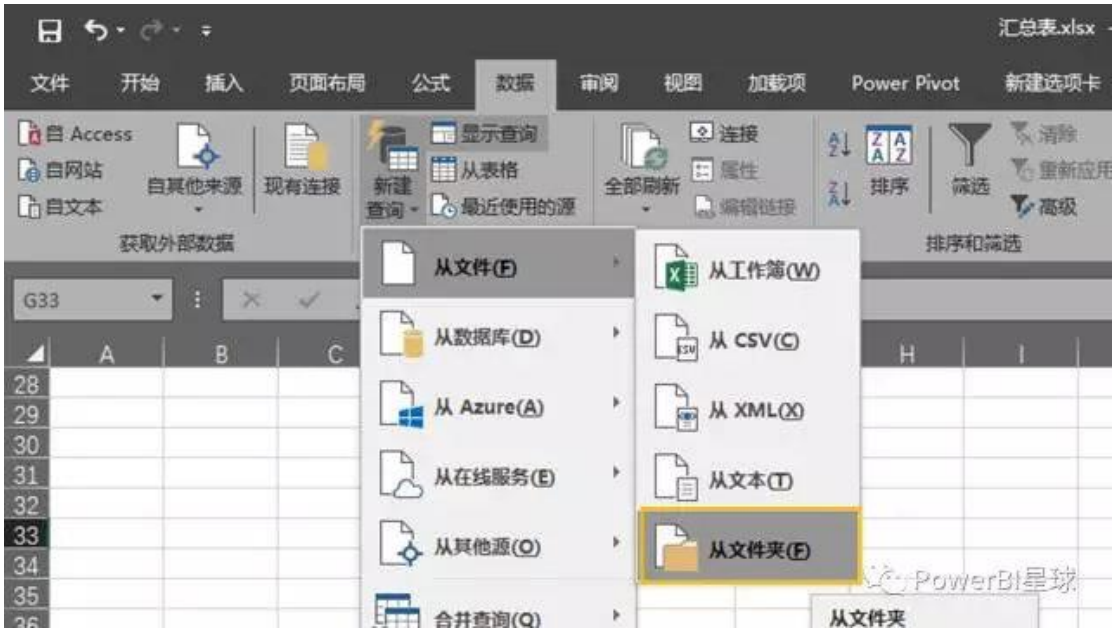
下面来看一下 PQ 是如何汇总多文件的数据的:

假设有一个连锁型零售商店，有北京、广州、杭州三个城市门店，总部每月需要汇总每个城市门店销售明细数据，现在需要汇总 2016 年 1-3 月的销售明细，共 9 个工作簿，保存在一个文件夹内，结构如下：



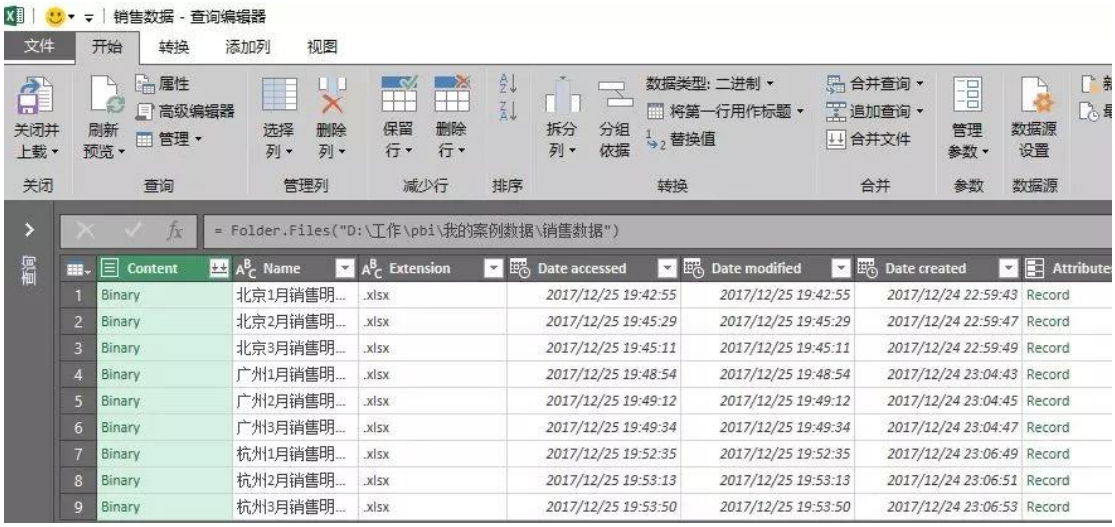
由于只是数据处理的过程，下面的演示就在 Excel2016 进行，使用 Excel2010、Excel2013 的插件以及在 Power BI Desktop 中的操作也都是一样的。

首先我们新建一张空白 Excel 工作簿，点击"数据"选项卡下"新建查询"，从文件夹获取数据：

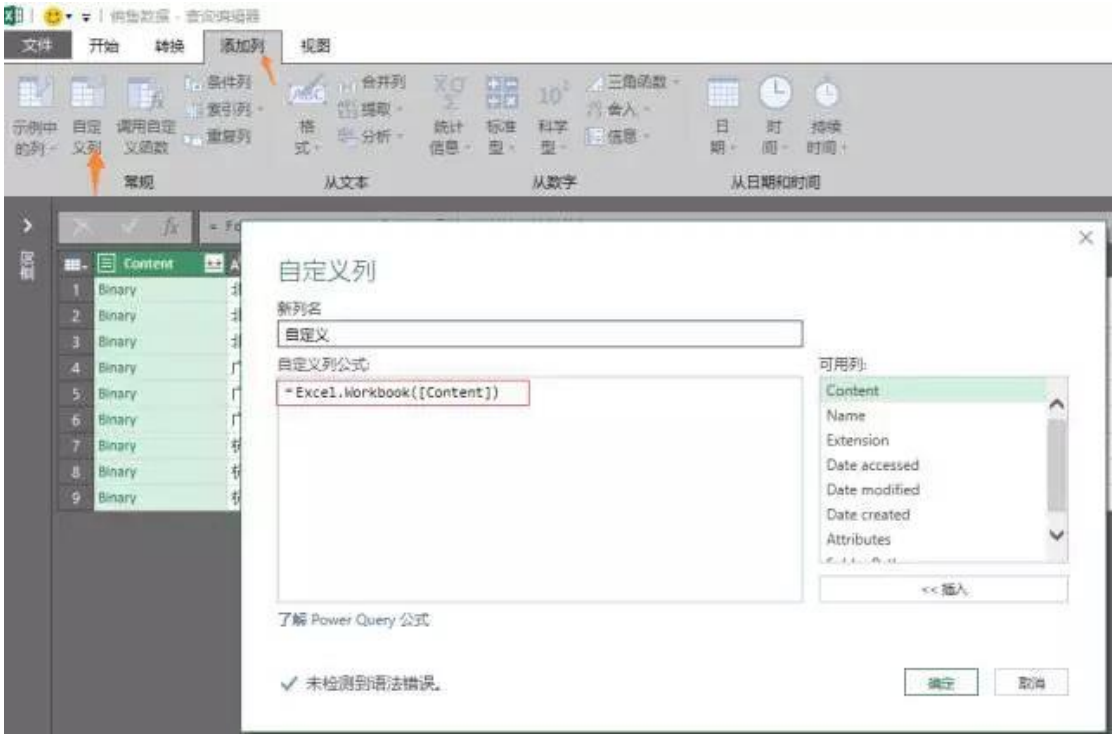


浏览找到该文件夹的路径，确认后出现这个界面，

点击"编辑",进入查询编辑器:



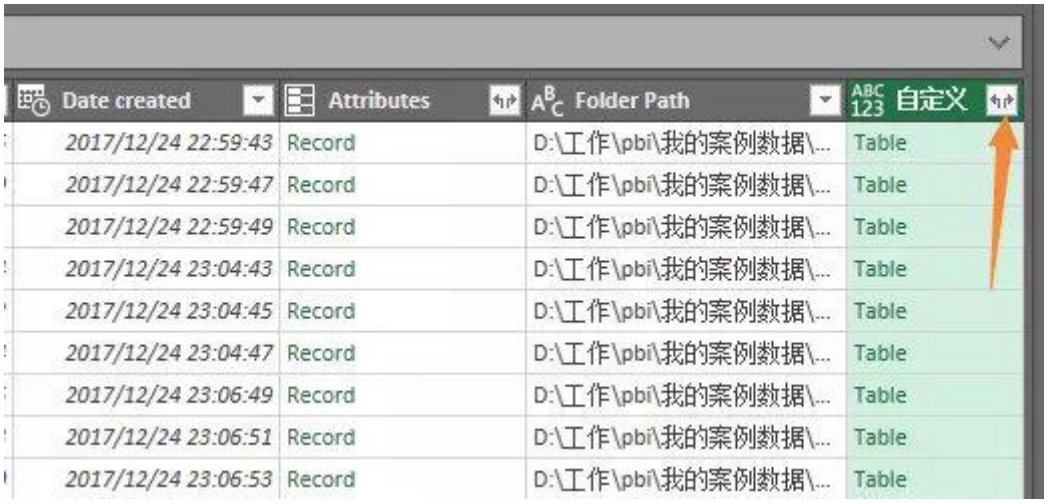
数据就储存在[Content]列,其他列都是每个工作簿的信息，现在要做的就是把 Content 的内容提取出来，点击"添加列"选项卡，添加自定义列，



自定义列中输入公式=Excel.Workbook([Content])，这里要注意严格区分大小写，不能写错了，这就是提取 Excel 格式数据的 M 函数（关于 M 函数后面会单独介绍）。

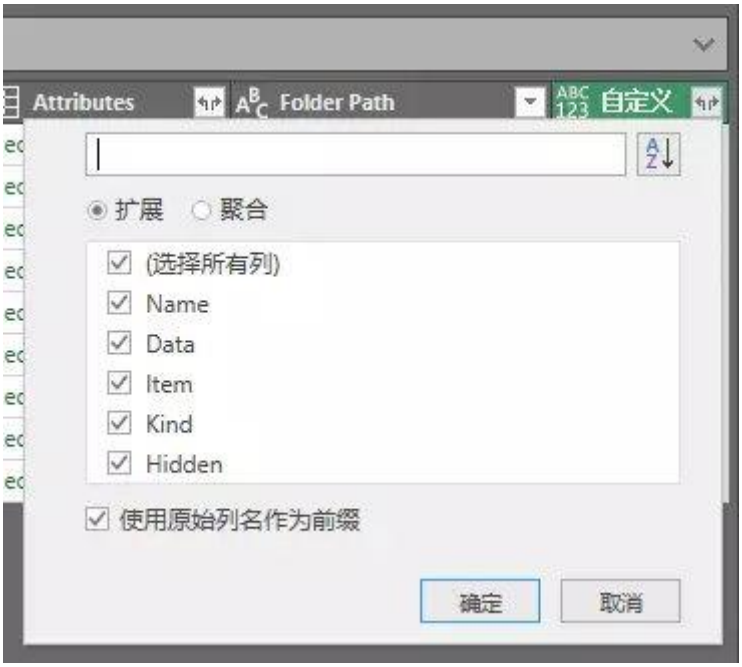
（现在不用添加列也可以，可以直接点击**[Content]**列右上角的向下的箭头，直接展开）

确认后就出现了一个自定义列：



Date created	Attributes	Folder Path	自定义
2017/12/24 22:59:43	Record	D:\工作\pbi\我的案例数据\...	Table
2017/12/24 22:59:47	Record	D:\工作\pbi\我的案例数据\...	Table
2017/12/24 22:59:49	Record	D:\工作\pbi\我的案例数据\...	Table
2017/12/24 23:04:43	Record	D:\工作\pbi\我的案例数据\...	Table
2017/12/24 23:04:45	Record	D:\工作\pbi\我的案例数据\...	Table
2017/12/24 23:04:47	Record	D:\工作\pbi\我的案例数据\...	Table
2017/12/24 23:06:49	Record	D:\工作\pbi\我的案例数据\...	Table
2017/12/24 23:06:51	Record	D:\工作\pbi\我的案例数据\...	Table
2017/12/24 23:06:53	Record	D:\工作\pbi\我的案例数据\...	Table

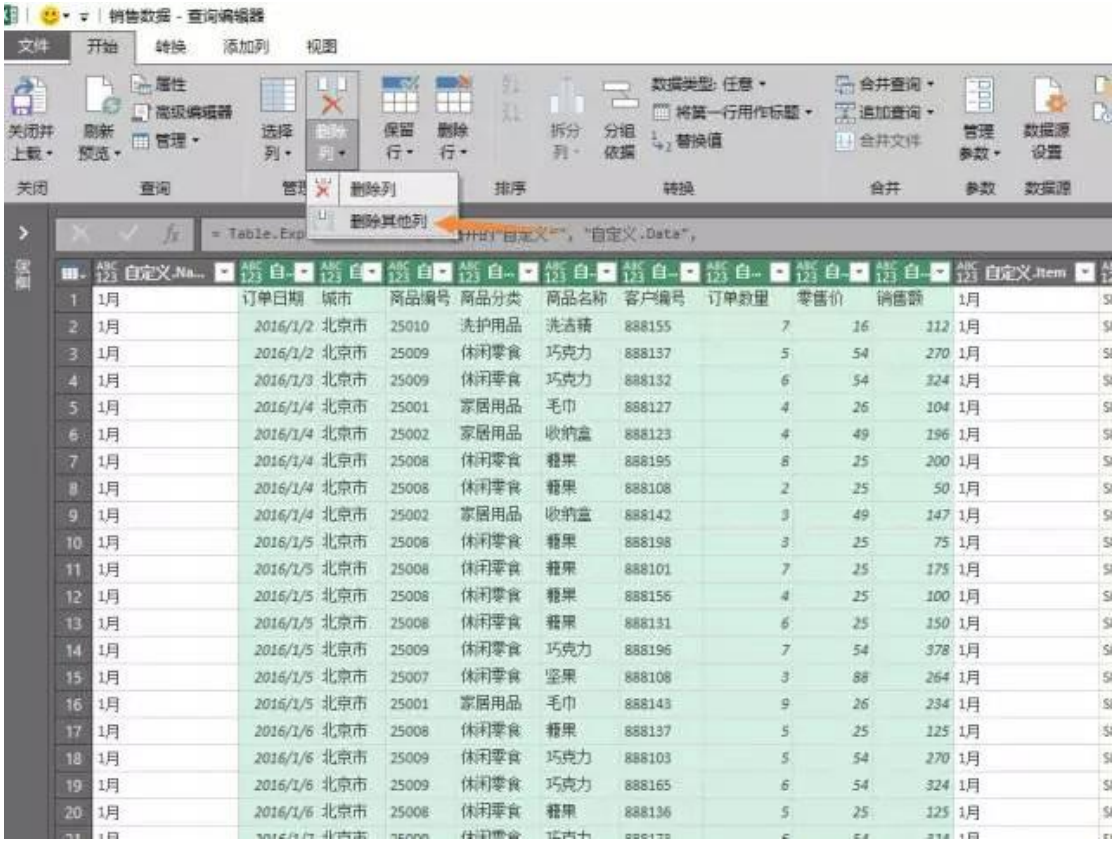
点击自[定义列]右上角的双箭头展开数据，出现这个窗口，



直接点击确定，出现了如下这个界面：

自定义.Na...	自定义.Data	自定义.Item	自定义.Kind	自定义.Hidden
1月	Table	1月	Sheet	FALSE
2月	Table	2月	Sheet	FALSE
3月	Table	3月	Sheet	FALSE
1月	Table	1月	Sheet	FALSE
2月	Table	2月	Sheet	FALSE
3月	Table	3月	Sheet	FALSE
1月	Table	1月	Sheet	FALSE
2月	Table	2月	Sheet	FALSE
3月	Table	3月	Sheet	FALSE

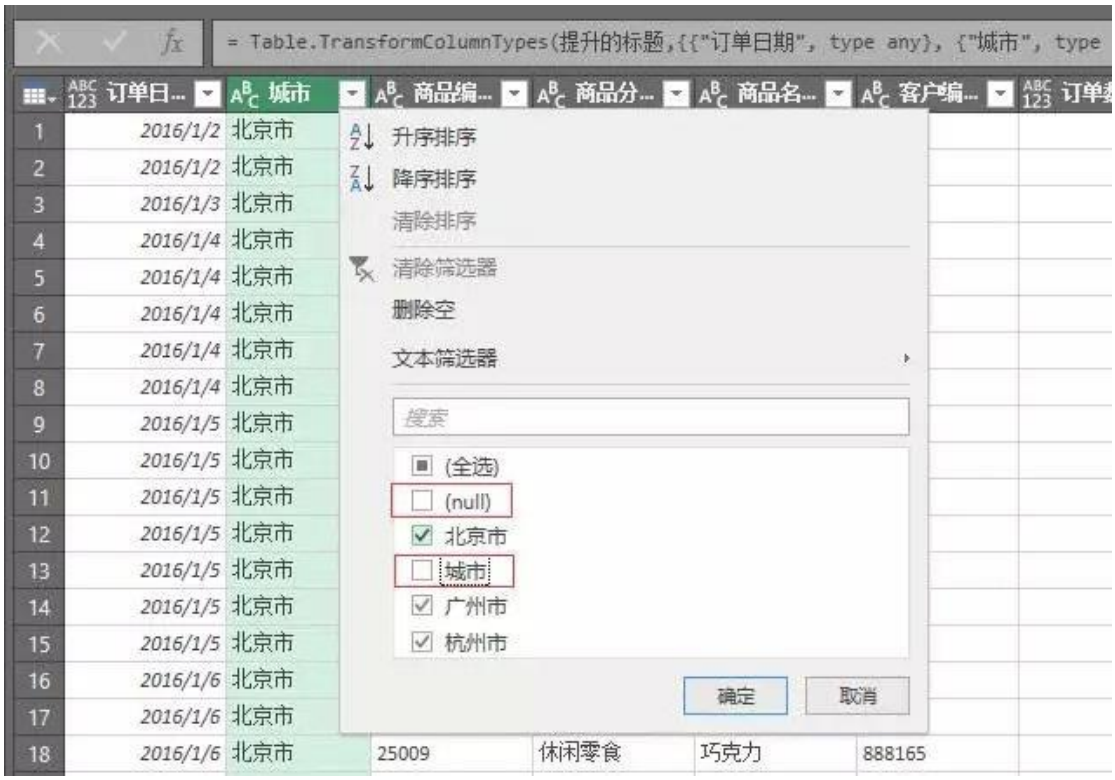
又新增加了几列，继续点击[自定义.Data]列的右上角的双箭头，然后还是直接点确认，数据就全部出来了：



然后只留下各门店上报的数据了，可以看到列的标题是系统添加的，其实应该用第一行作为列的标题，我们直接点击"转换"选项卡下的将第一行作为标题：

然后标题就提升上去了。

数据导入过程中 9 个表格的标题行是重复的，另外表格中可能有空行，所有把标题行和空行筛选出去，像在 Excel 中一样，点击城市的倒三角，去掉这两个勾选：



数据汇总完成，点击上载数据：

然后大功告成，数据就全部汇总到这个 Excel 表格中了。

看着好像步骤挺多，其实动手做起来，所有这些步骤只需一分钟而已，中间除了那个简单的 M 函数，一直都是点点鼠标，是不是非常简单呢。

更简单的是，上面操作的所有步骤都被记录下来，下个月销售记录更新的时候，比如把各个门店的 **4** 月份的明细数据放到相应的文件夹里面，连点鼠标都不用了，直接刷新数据，然后 **4** 月的数据就全部汇总到这个表格了。

如果你说这些其实通过 VBA 或者简单的复制粘贴还都可以做出来，那么如果有 100 家门店，每家门店全年 12 个月的数据呢，复制粘贴显然不现实，如果数据量大用 VBA 估计也会把电脑卡死。而在 PQ 中呢，就是打开文件点击刷新，这个文件夹下无论多少文件，无论数据量有多大，汇总也是秒秒钟的事情。

这体验爽爆了吧**:)**

这只是举个简单的例子，实际上 Excel 能处理的工作，PQ 都能处理，并且能更简单、更节省时间的处理；而 Excel 处理不了的大数据，PQ 同样能够轻松应对。

Power Query 的优势:

操作简单，即使是小白，无需掌握复杂的函数即可处理大部分数据处理工作

数据量无限制，具体多少取决于电脑配置，上亿行不是问题

自动化，处理过程全记录，每次数据源更新后刷新即可，无需重复劳动

数据处理一般可简化为三个过程，数据获取、数据整理、数据丰富，后面几篇文章会分别介绍。

动手才是最好的学习！案例数据可在本公众号回复“**PQ** 体验”获取。

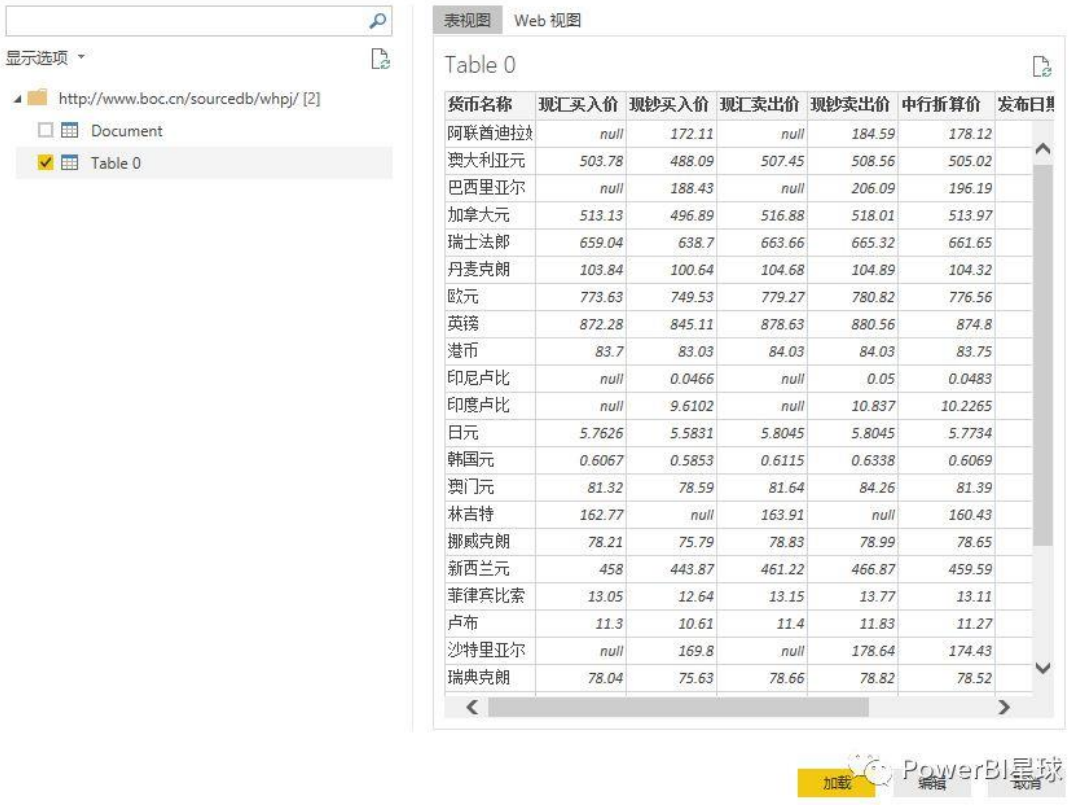
数据获取-PoweBI 支持从多种数据源导入数据

PowerBI 的强大绝不仅是最后生成炫酷的可视化报告，她在第一步数据获取上就显示出了强大的威力，利用 Power Query 的强大数据处理功能，几乎可以从任何来源、任何结构、任何形式上获取数据，

数据的获取上不仅支持微软自己的数据格式，比如 Excel、SQL Server、Access 等;还支持 SAP、Oracle、MySQL、DB2 等几乎能见到的所有类型的数据格式，总有一种适合你;

不仅能能从本地获取数据，还能从网页抓取数据。选择从 Web 获取数据，只要在弹出的 URL 窗口中输入网址，网页上的数据就可直接抓取到，用这种方法我们可以实时抓取股票涨跌、外汇牌价等等交易数据，现在我们尝试一下，比如从中国银行网站上抓取外汇牌价信息，先输入网址：

点击确定以后，出现预览窗口，



点击编辑，进入查询编辑器，

	货币名...	1.2 现汇买...	1.2 现钞买...	1.2 现汇卖...	1.2 现钞卖...	1.2 中行折...	发布日...	发布时...
1	阿联酋迪拉姆	null	172.11	null	184.59	178.12	2017/12/26	20:59:24
2	澳大利亚元	503.78	488.09	507.45	508.56	505.02	2017/12/26	20:59:24
3	巴西里亚尔	null	188.43	null	206.09	196.19	2017/12/26	20:59:24
4	加拿大元	513.13	496.89	516.88	518.01	513.97	2017/12/26	20:59:24
5	瑞士法郎	659.04	638.7	663.66	665.32	661.65	2017/12/26	20:59:24
6	丹麦克朗	103.84	100.64	104.68	104.89	104.32	2017/12/26	20:59:24
7	欧元	773.63	749.53	779.27	780.82	776.56	2017/12/26	20:59:24
8	英镑	872.28	845.11	878.63	880.56	874.8	2017/12/26	20:59:24
9	港币	83.7	83.03	84.03	84.03	83.75	2017/12/26	20:59:24
10	印尼卢比	null	0.0466	null	0.05	0.0483	2017/12/26	20:59:24
11	印度卢比	null	9.6102	null	10.837	10.2265	2017/12/26	20:59:24
12	日元	5.7626	5.5831	5.8045	5.8045	5.7734	2017/12/26	20:59:24
13	韩国元	0.6067	0.5853	0.6115	0.6338	0.6069	2017/12/26	20:59:24
14	澳门元	81.32	78.59	81.64	84.26	81.39	2017/12/26	20:59:24
15	林吉特	162.77	null	163.91	null	160.43	2017/12/26	20:59:24

PowerBI星球

外汇数据抓取完成，剩下的就是数据整理的过程了，而且这些抓取的信息可以随时刷新来更新数据的。这只是抓取外汇牌价的第一页，其实抓取多个页面数据也是可以的，等后面介绍过 M 函数以后再专门写一篇。

以后再也不需要手动从网页上复制数据再粘贴到表格中了。

其实每个人接触到的数据格式很有限，熟悉自己的数据类型知道如何导入到 PowerBI 以后，下一步就是进行数据处理的过程，这才是我们真正需要掌握的核心技巧。

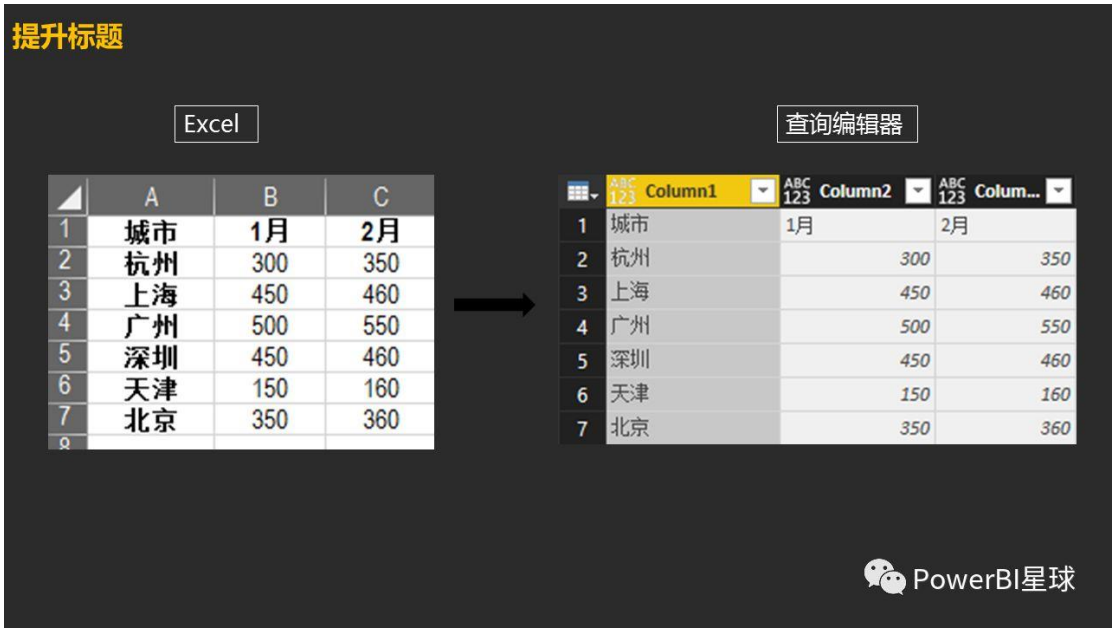
PowerQuery 常用的数据清洗十三招

对导入的数据进行数据整理的过程一般称为「数据清洗」，之所以称之为清洗，是因为在数据分析师眼中，杂乱的数据就是脏数据，只有被清洗成干净的数据后才可以进行分析使用，下面我们就来认识 Power Query 的强大数据清洗功能。

01

提升标题

在 Excel 中第一行为标题行，从第二行开始才是数据，但在 PQ 中，从第一行开始就需要是数据记录，标题在数据之上，



因此从 Excel 导入数据的第一步就是要提升标题，点击「转换」的将第一行作为标题，标题提升就完成了。

点击将第一行作为标题旁边的下拉按钮，还有个将标题作为第一行，实际上就是拉低标题，这个功能也特别有用，下面会介绍到。

02

更改数据类型

更改数据类型虽然很简单，但设置正确的数据类型非常重要，后期数据建模和可视化过程中，很有可能会出现一些意想不到的错误，最后发现是数据类型设置的不对，所以一开始就养成设置数据类型的好习惯。有两种方式设置数据类型，

03

删除错误/空值

数据导入后，有可能出现错误（Error）或者空值（null），

做数据分析之前，要想达到右边的效果，需要把它们去掉，操作如下：

04

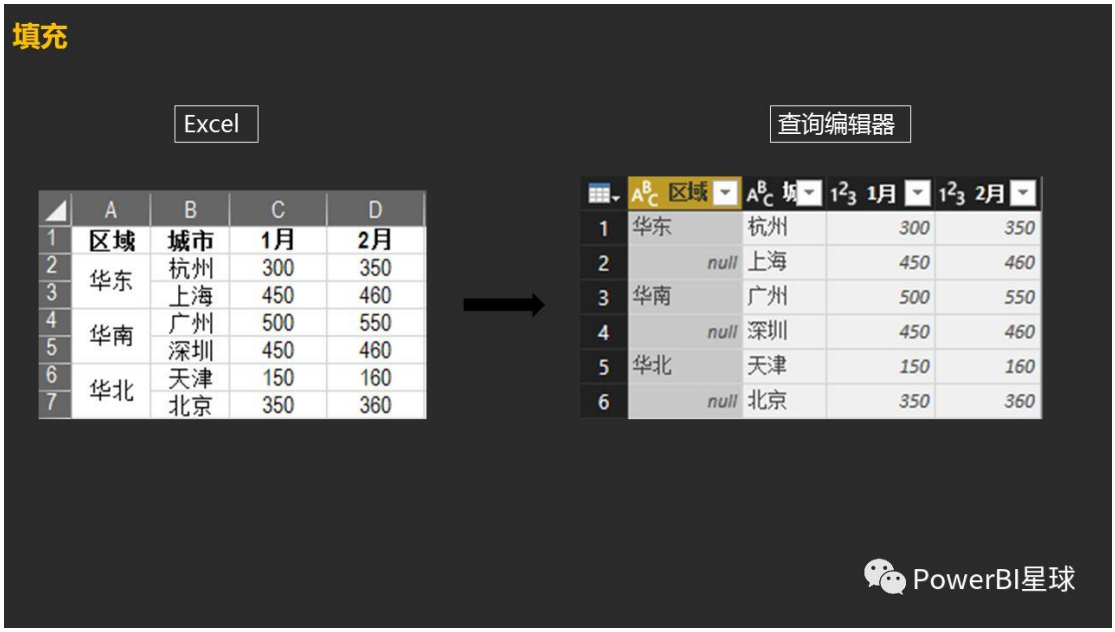
删除重复项

删除重复项在 PQ 中非常简单，选中需要删重的列，右键选择「删除重复项」即可：

05

填充

在 Excel 数据中经常会见到合并单元格的情况，导入后就变成了空值，像下面这样：



在 PQ 中直接向下填充即可：

06

合并列

在 PQ 中选择需要合并的列，然后在「转换」中找到"合并列"，弹出合并列窗口：

比如刚才的表格，把[区域]和[城市]合并：

07

拆分

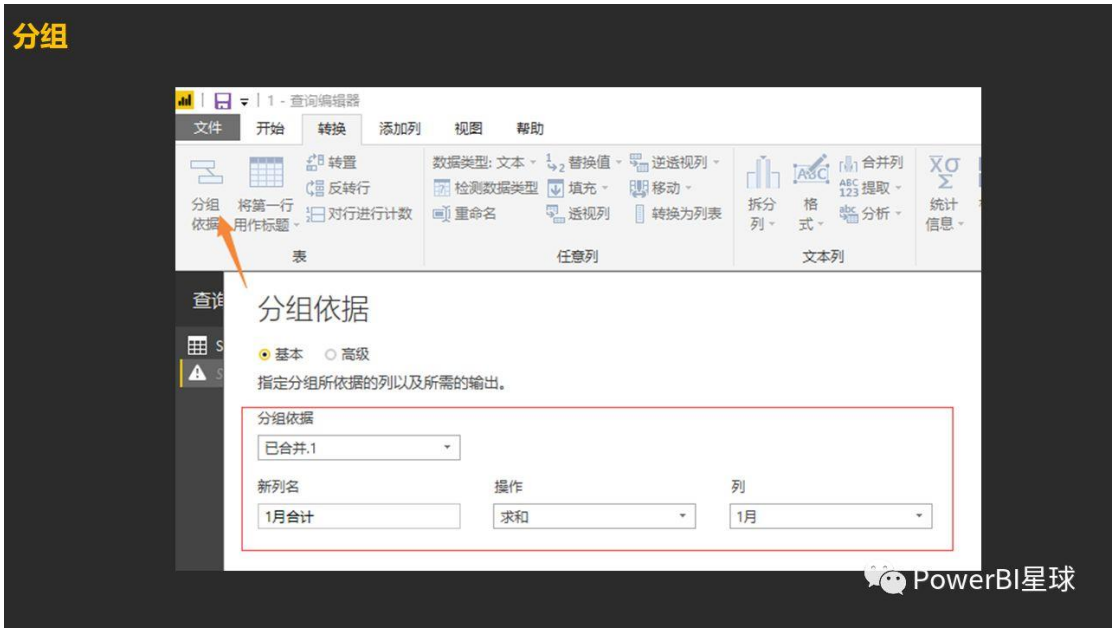
拆分相当于是合并列的反动作，不过功能更丰富，可以选着按字符数，也可以选择按分隔符，如果列中包含多个分隔符，还可以选择按哪个位置的分隔符来拆分：

我们把刚才的合并列再拆分一下，又变成合并前的格式了：

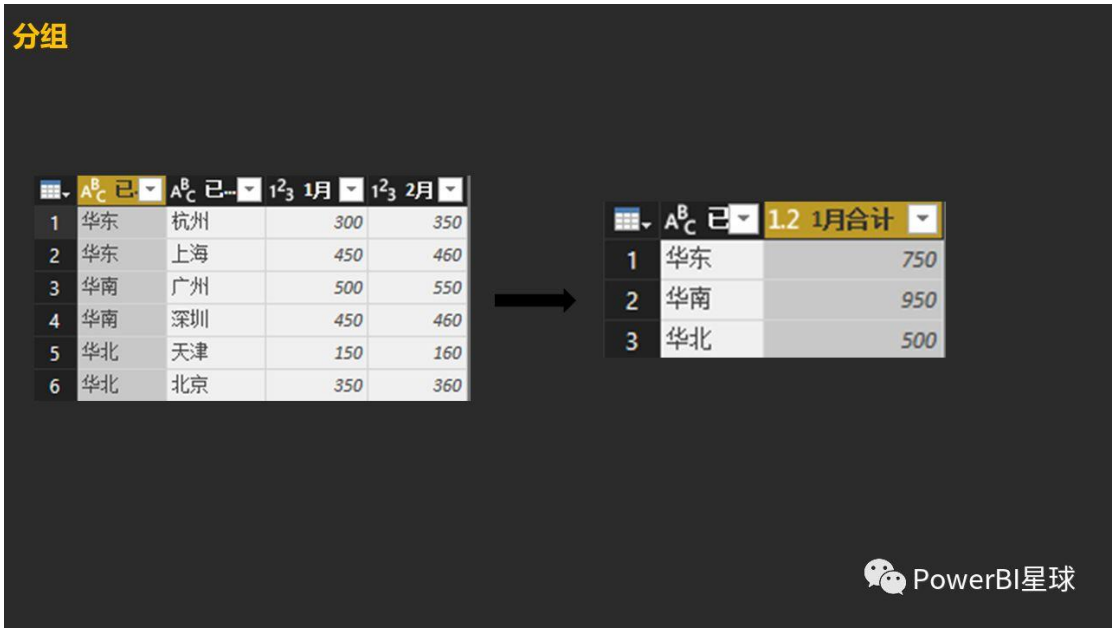
08

分组

相当于 Excel 中的分类汇总功能，比如刚才的数据，我们要计算各区域 1 月份的合计金额，点击分组依据：



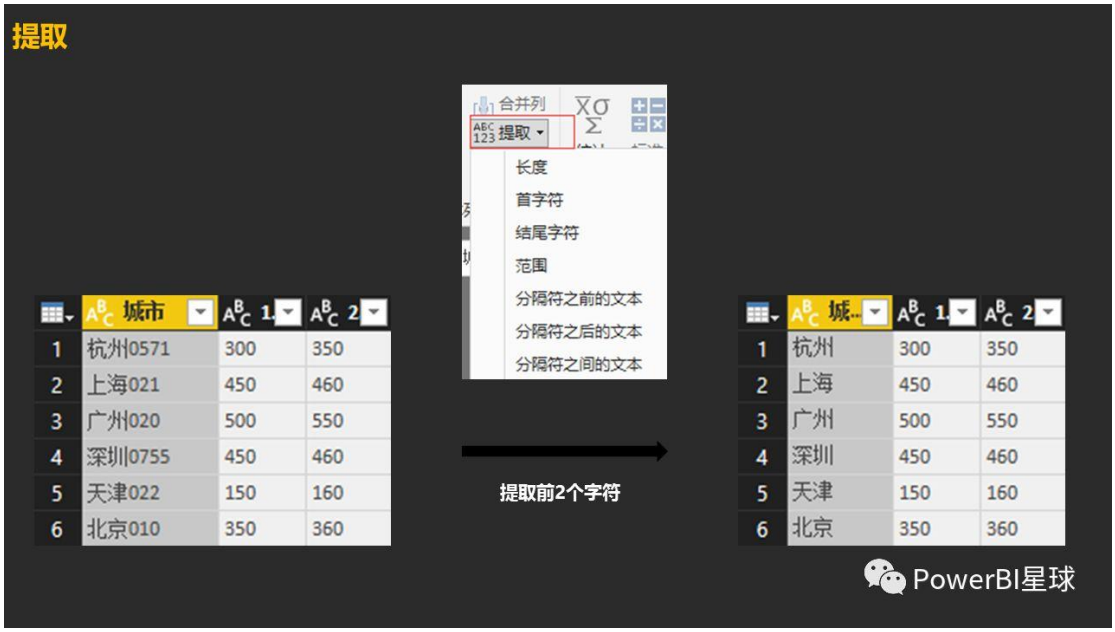
把相应参数填上以后，分组完成：



09

提取

PQ 的提取功能可以按照长度、首字符、尾字符、范围等来提取，比如下面这个例子，提取前 2 个字符：



	城市	1	2
1	杭州0571	300	350
2	上海021	450	460
3	广州020	500	550
4	深圳0755	450	460
5	天津022	150	160
6	北京010	350	360

	城...	1	2
1	杭州	300	350
2	上海	450	460
3	广州	500	550
4	深圳	450	460
5	天津	150	160
6	北京	350	360

10

行列转置

数据处理有的时候需要行列互相转换一下，比如把各各城市变成列标签，月份变成行标签，点击行列转置，

是不是发现哪里有些不对劲？月份不见了，这是因为转置的时候，只转数据的部分，月份并不在数据区，我们要想保留月份，先要把月份降下来，这里用到我们前面介绍的"将标题作为第一行"，

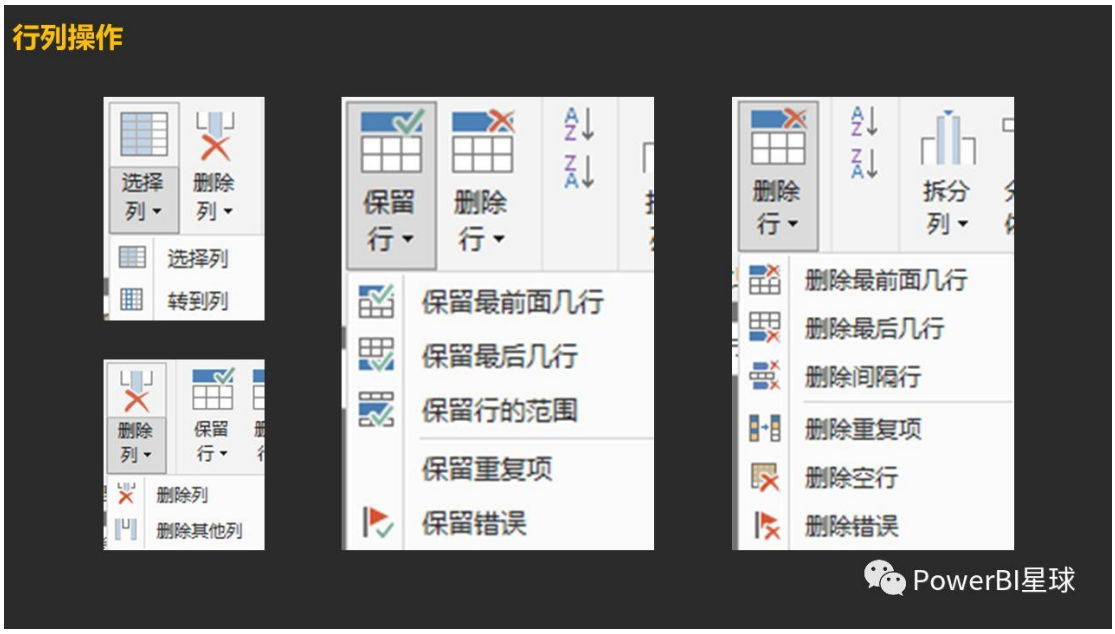
标题下降以后，再进行转置就可以了，

转置完成，不过还要再把第一行城市提升为标题。

11

行列操作

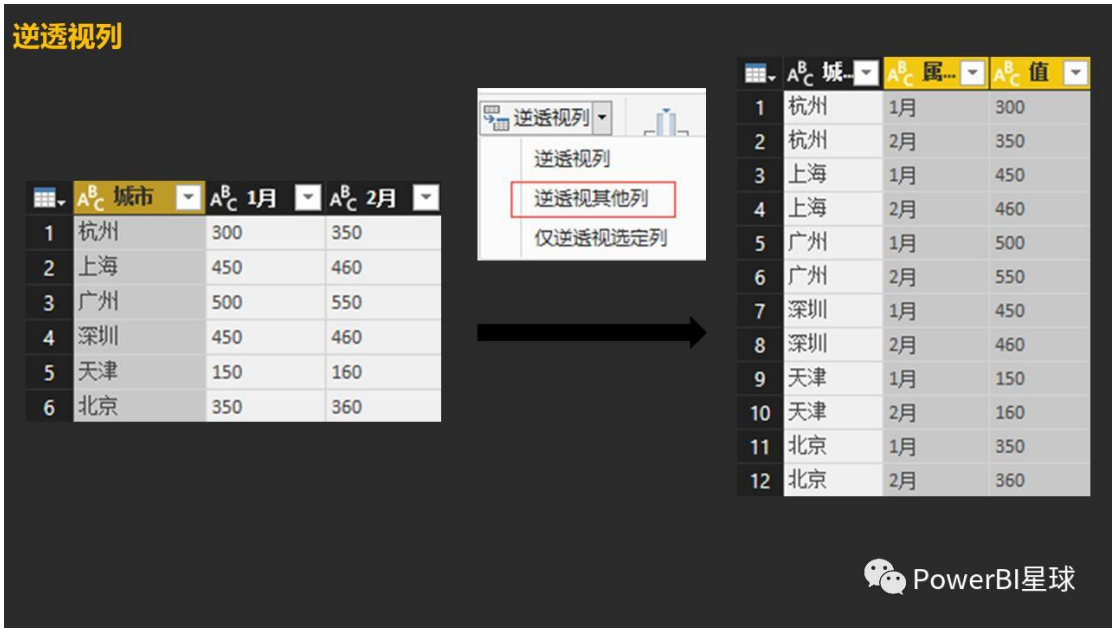
PQ 的行列操作十分灵活，非常大规模的数据操作。



12

逆透视列

这是 PQ 非常便捷的逆天功能，由于数据分析的需要，我们经常要将二维表变为一维表，之前在 Excel 中需要很多操作步骤才能完成，而通过逆透视功能，可以一键降为一维表，



	城市	1月	2月
1	杭州	300	350
2	上海	450	460
3	广州	500	550
4	深圳	450	460
5	天津	150	160
6	北京	350	360

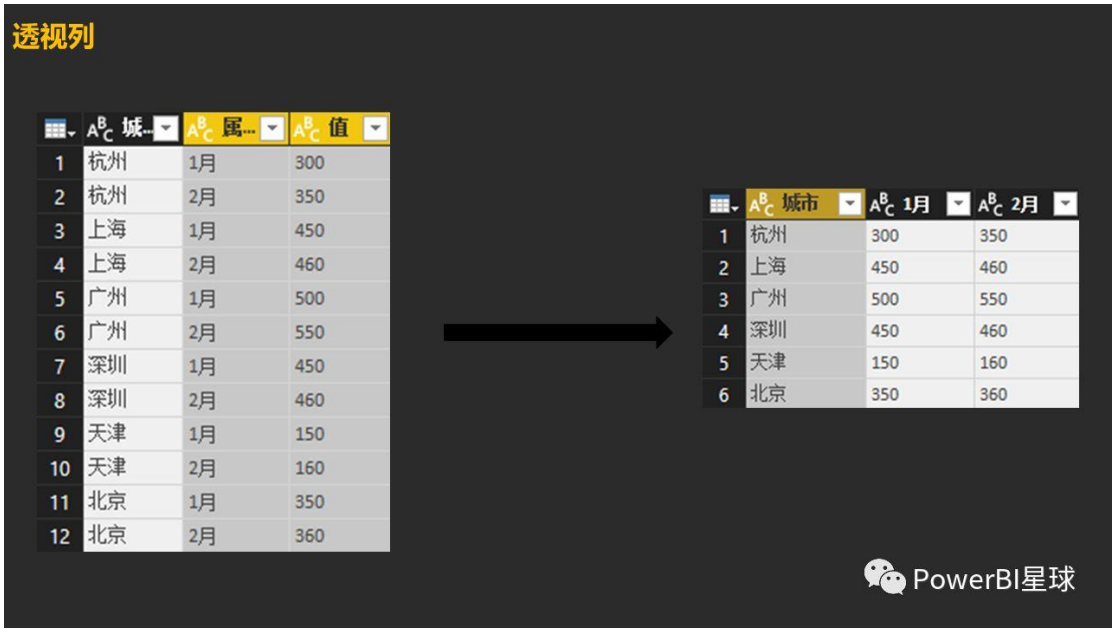
	城...	属...	值
1	杭州	1月	300
2	杭州	2月	350
3	上海	1月	450
4	上海	2月	460
5	广州	1月	500
6	广州	2月	550
7	深圳	1月	450
8	深圳	2月	460
9	天津	1月	150
10	天津	2月	160
11	北京	1月	350
12	北京	2月	360

13

透视列

做分析需要一维表，而为了展现的需要，常常还要把一维表变成二维表，也就是 Excel 中的数据透视，在 PQ 中同样可以一键透视，比如把刚才的一维表变成原样，聚合方式选择“不要聚合”。

点击确定就变回原来的二维表了，



以上这些是最基本也是最常用的数据清洗功能，看上去也都很简单，动手操作才能更好的掌握，熟练运用以后基本能搞定大部分数据处理的工作，这些不仅是学习 Power BI 的必备技能，更能让你的 Excel 水平短时间内大大提升。

Power Query 数据丰富

上一篇文章都是在原表数据基础上的分分合合，但做数据分析的时候还经常需要在原有数据的基础上增加一些辅助数据，比如加入新列、新行，或者从其他表中添加进来更多维度的数据，这些就是数据丰富的过程。

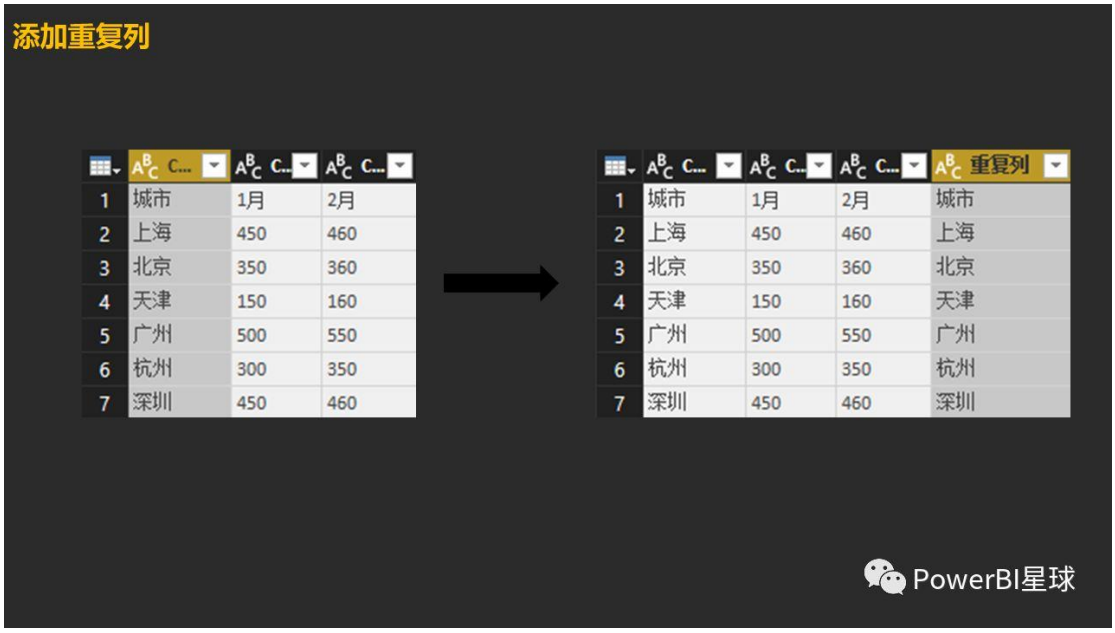
01

添加列

Power Query 中添加列有四种形式，重复列、索引列、条件列、自定义列：

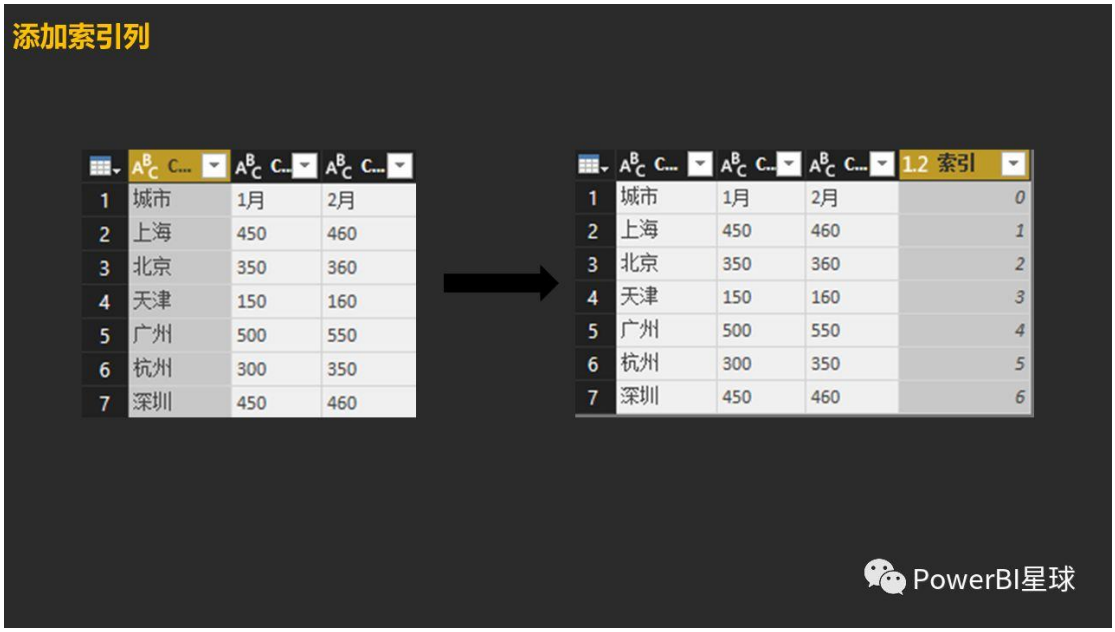
(一)添加重复列

重复列就是把选中的列复制一列，以便对该列的数据进行处理而不损坏原有列的数据，



(二)添加索引列

索引列就是为每行增加个序号，记录每一行所在的位置，可以从 0 或者 1 开始，



(三)添加条件列

添加一列根据指定条件从其他列计算的数据，打开窗口看看，其实就是 if 函数，

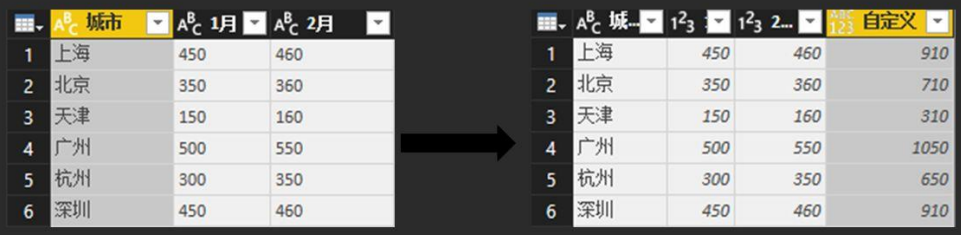
根据上面的条件，得到新的一列，

(四)添加自定义列

自定义列就是用 M 函数生成新的一列，

比如添加一列求 1 月和 2 月的和，把两列相加，PQ 中的字段用[ ]框住，不需要手动输入，直接点击右边的字段名就可。

添加自定义列



	城市	1月	2月
1	上海	450	460
2	北京	350	360
3	天津	150	160
4	广州	500	550
5	杭州	300	350
6	深圳	450	460

	城...	1...	2...	自定义
1	上海	450	460	910
2	北京	350	360	710
3	天津	150	160	310
4	广州	500	550	1050
5	杭州	300	350	650
6	深圳	450	460	910

PowerBI星球

02

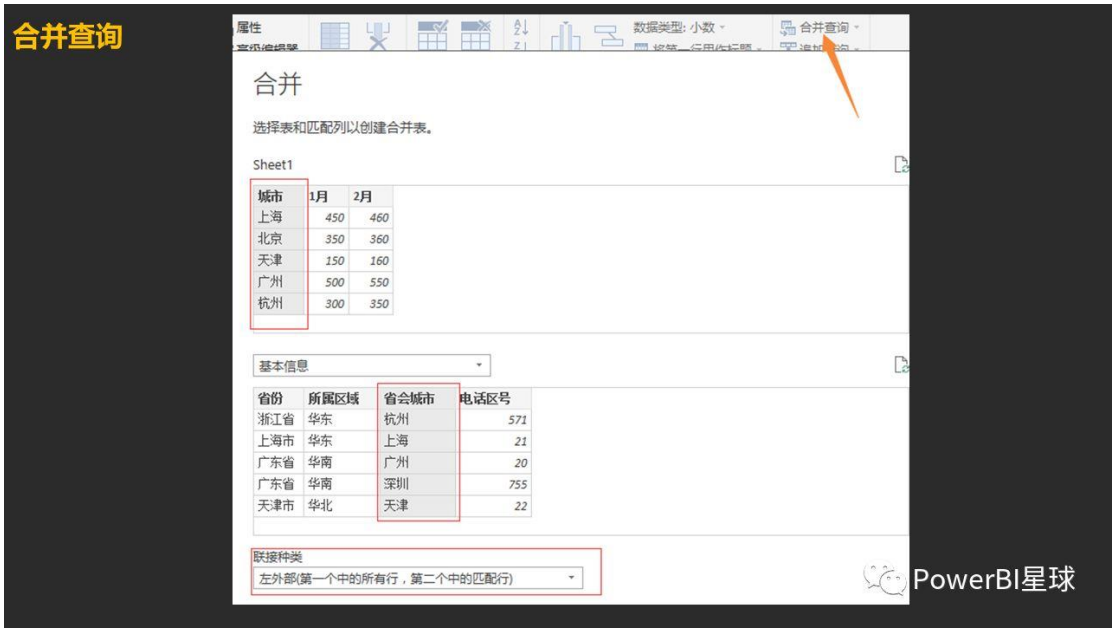
追加查询

追加查询是在现有记录的基础上，在下边添加新的行数据，是一种纵向合并，比如有两个表格式相同，需要合并为一个表，点击“追加查询”，

03
合并查询

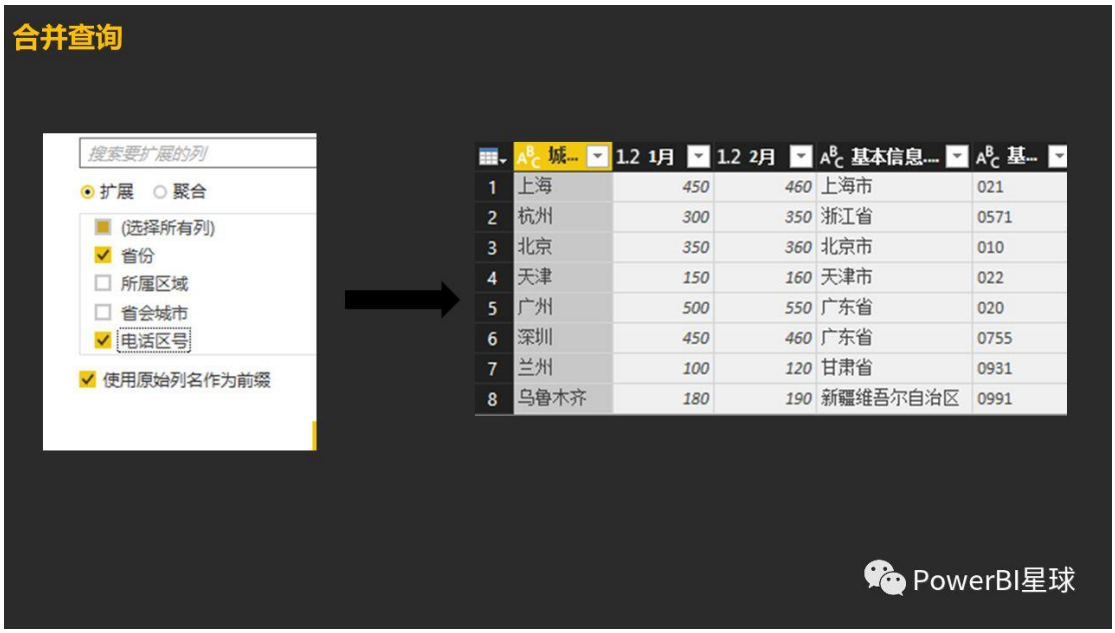
如果说追加查询是纵向合并，那么合并查询就是横向合并，相当于 Excel 的 VLOOKUP 功能，就是匹配其他表格中的数据，不过 PQ 中的合并查询要比 VLOOKUP 功能强大的多，并且操作也更简单。

比如我们想从基础信息表中找到每个省市对应的省市和电话区号，点击"合并查询"



选择两个表需要匹配的字段，这两个表都是[城市]列，下方联结种类选择左外部：第一个表的所有行，第二个表的匹配行，就得到了下面这张表，

然后点击展开，勾选我们需要的字段，合并查询就完成了，增加了每个城市对应的省份和区号。



通过以上的数据丰富以及前面文章中的数据清洗的学习，基本上见识到了所有的界面功能，熟练掌握应该说大部分的数据工作都能够应对，几乎不用任何函数语言。

性价比本来就是 Power Query 的一大优势，即使不打算学习 PowerBI，PQ 也值得你抽出几个小时来学习。任何人都可以通过很短时间的练习就可在数据处理上获得突破性的提升，不知不觉间已经站在了 Excel 的肩膀上。

认识 M 函数

前面我们学习 PQ 的时候都是用鼠标操作，虽然通过这些操作能完成大部分的数据处理，但是毕竟还有些复杂的工作是处理不了的，如果想彻底驾驭 PQ，必须得掌握点高级玩法。就像学习 Excel 一样，做个表格我们只要会简单的操作就可以了，但要想学好 Excel，必须会点公式、VBA 是一样的道理，PQ 的高级技能也是需要写公式函数的，在 PQ 中用的函数称之为 M 函数。

在之前的 PQ 操作中，其实 M 函数无处不在，比如做数据清洗的每一个步骤，背后都有 M 函数的影子。打开高级编辑器，可以看到所有这些步骤的 M 语言。

如果我们不进行鼠标操作，直接在编辑器中编写这些语言，也是可以得到最终的结果的，有了 M 函数，PQ 的数据处理具有很强的可读性和可移植性。

01 | 为什么要学习 M 函数

- 有些复杂的操作必须借助 M 函数
- M 函数更加灵活，简洁高效

02 | M 函数基本规范

- M 函数对大小写敏感，每一个字母必须按函数规范书写，第一个字母都是大写
- 表被称为 Table，每行的内容是一个 Record，每列的内容是一个 List
- 行标用大括号{ }，比如取第一行的内容：=表{0} //PQ 的第一行从 0 开始
- 列标用中括号[ ]，比如取自定义列的内容：=表[自定义]
- 取第一行自定义列的内容：=表{0}[自定义]

03 | 常用的 M 函数

- 聚合函数：

求和：List.Sum()

求最小值：List.Min()

求最大值：List.Max()

求平均值：List.Average()

- 文本函数：

求文本长度：Text.Length()

去文本空格：Text.Trim()

取前 n 个字符：Text.Start(文本,n)

取后 n 个字符：Text.End(文本,n)

- 提取数据函数：

从 Excel 表中提取数据：Excel.Workbook()

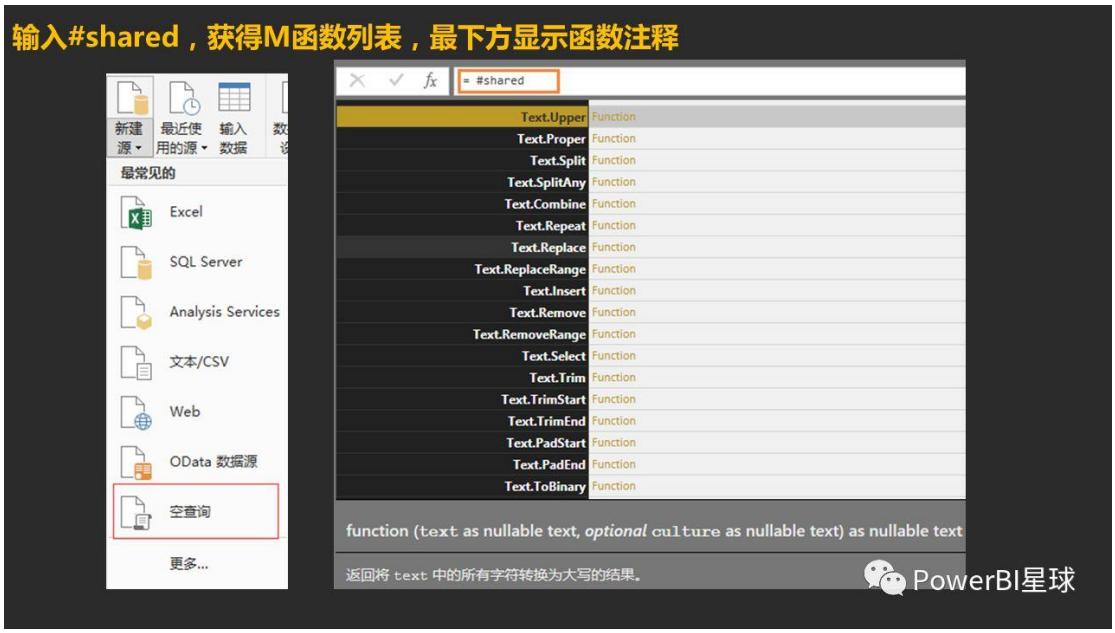
从 Csv/Txt 中提取数据：Csv.Document()

- 条件函数：

if　else then　（相当于 Excel 中的 IF)

04 | 从哪里查找 M 函数

新建一个空查询，在公式标记栏中输入#shared，就把所有的 M 函数显示出来了，点击某个函数，最下方便出现该函数的注释：



05 | 学习建议

虽然 M 函数很强大，但是不建议一头扎进去学习她，毕竟对于一个之前没有接触过编程的人来说，学习成本还是挺高的，并且大部分函数并不常用。

我的建议是先能够读懂 **M** 语言，并把常用的函数，比如文本函数、字符串函数、日期函数等浏览一遍，知道大概都有什么函数，分别是哪些功能，然后在数据处理过程中碰到鼠标操作难以完成的问题，能想到有哪个 **M** 函数可以利用，直接查找和并根据注释使用、或者会修改相应的 **M** 函数即可。

如果熟练使用鼠标操作功能，又能灵活运用 M 函数，你将在数据处理的路上一骑绝尘、所向披靡。在大数据已经进入日常工作和生活的今天，拥有 Power Query 这个利器，我们就能用最少的时间来处理数据，留下更多的时间去分析数据，去发现数据背后的规律，这才是我们学习 PQ、学习 PowerBI 的最终目标。

【第 3 天 | 数据建模】

了解 **PowerBI** 中如何进行数据建模，以及 **PowerBI** 中的核心概念：度量值，并初步了解数据建模语言：**DAX.**

Power BI 数据建模

数据建模并没有那么高深，你同样可以学会！这篇文章通过一个实例创建一个简单的数据建模，并引出两个重要的概念：度量值和 **DAX**.

之前谈论 PowerBI 与 Power Pivot 的关系时就提到，Power BI 数据建模其实就相当于 Excel

的 Power Pivot 插件，不过这个插件已内嵌到 PowerBI Desktop 中了，两者的功能基本相同。Pivot 是透视的意思，那么 PBI 的数据建模也和透视有许多共通之处。

使用的数据透视表的都知道，透视表只能从单个表中取数，如果想把其他表中的数据也放进来，只能先利用 Vlookup 把其他表的数据合并过来，然后再把这个字段放到透视表中。这只适用于数据非常简单的情况，如果数据量大或者维度很多，用透视表就无法满足需求了。

Power BI 突破了这个限制，可以从多个表格、多种来源的数据中，根据不同的维度、不同的逻辑来聚合分析数据；而提取数据的前提是要将这些数据表建立关系，这个建立关系的过程就是数据建模。

以一个实例来理解数据建模。

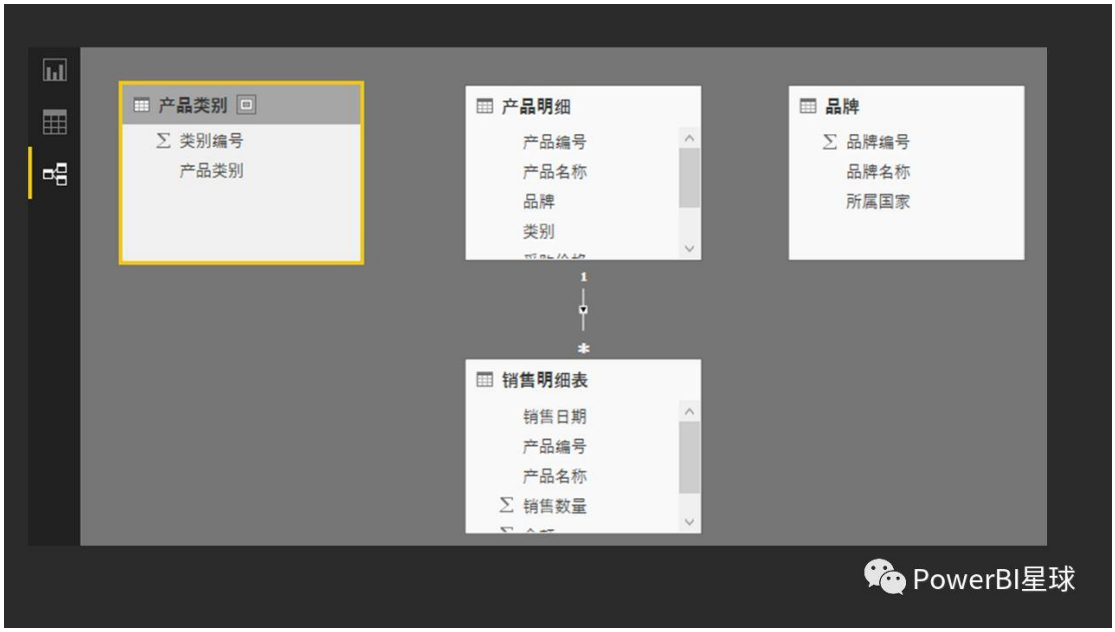
比如有个电子产品专卖店，销售产品有三类：手机、电脑、平板，每一类又分别来自三个品牌：小米、苹果、三星，那么这个店销售的产品共计 9 个，其销售明细也是记录这些产品每天的销售数据，

销售日期	产品编号	产品名称	销售数量	金额
2015年5月1日	1001	小米手机	4	8000
2015年5月2日	1001	小米手机	6	12000
2015年5月2日	2001	苹果手机	2	16000
2015年5月2日	1003	小米平板	1	1500
2015年5月3日	1001	小米手机	3	6000
2015年5月3日	2001	苹果手机	1	8000
2015年5月4日	3001	三星手机	1	4200
2015年5月4日	1001	小米手机	3	6000
2015年5月5日	2001	苹果手机	2	16000
2015年5月6日	2002	苹果电脑	1	10000
2015年5月7日	1001	小米手机	5	10000
2015年5月8日	2001	苹果手机	2	16000
2015年5月8日	3002	三星电脑	1	4000

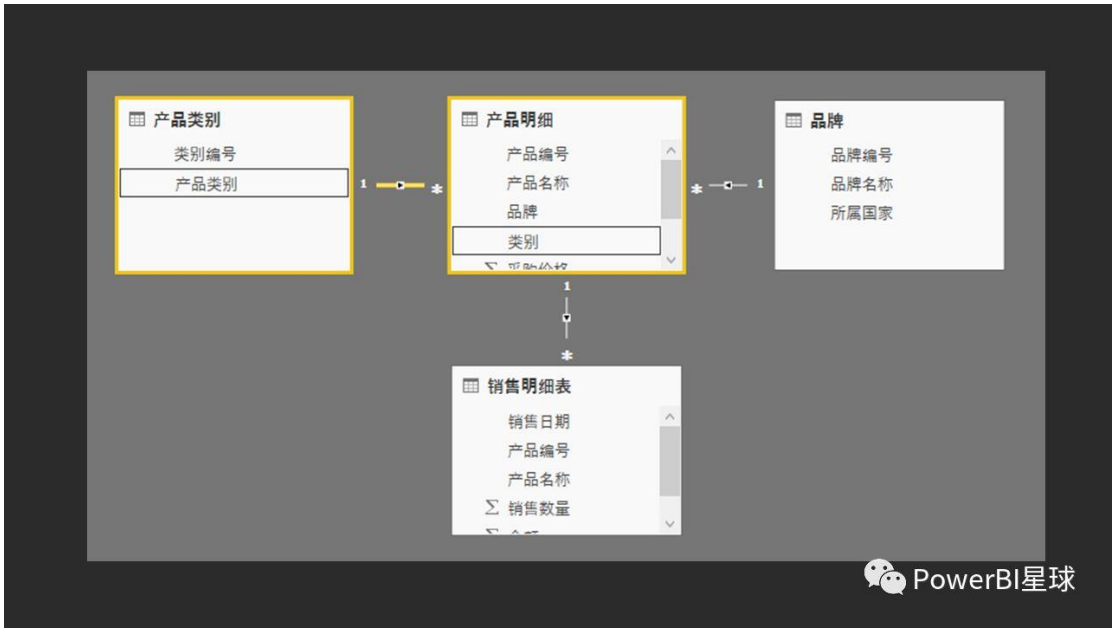
为了能分析每个品牌的销售金额，或者分析每个产品类别的销售情况，其实还应该设计个产品明细表以及对应的品牌表和种类表，像这样的：

从这四个表中很容易就能想到它们之间的关系，品牌表和产品类别表分别和产品明细表中的品牌与产品种类相对应，而产品明细表中的产品编号和销售明细表中的产品编号相对应。

下面就演示一下在 Power BI Desktop 中建立一个模型，导入以后点击关系，出现这 4 张表，



可以看出产品明细表和销售明细表之间已经有一条线，这是由于表格导入后，PowerBI 会自动检测关系并联接，没有检测到的表，可以点击一个表中的字段托到另一个表的对应字段上，就可以建立关系了，把类别表、品牌表和产品明细表建立关系后，关系图如下：



一个简单的数据模型就建立好了，可以点进去看看建立模型的相关参数。

点击关系连接线，两边的表对应的连接字段会框选，双击关系线，进入编辑关系窗口：

编辑关系窗口可以看出关联的两个表和对应的字段，也可以更改联结的字段；下面还有两个可选项，基数和交叉筛选方向。

基数

- 基数就是两个连接字段的对应关系，分为多对一、一对一和一对多，一对多和多对一其实是一样的，实际上就是两种关系：
- 多对一（*：1）：这是最常见的类型，代表一个表中的关系列有重复值，而在另一个表中是单一值
- 一对一（1：1）：两个表是一对一的关系，列中的每个值在两个表中都是唯一的
- 具有唯一值的表通常称为“查找表”，而具有多个值的表称为“引用表”。在上述的关系图上，产品明细表上类别手机、平板、电脑都不是唯一的，每个品牌都有这种类型，是个引用表；但类别表上，几种类别都是唯一值，因此这两个表是多对一的关系，类别表也就是查找表。

交叉筛选方向

表示数据筛选的流向，有两种类型：

- 双向：两个表可以互相筛选
- 单向：一个表只能对另一个表筛选，而不能反向

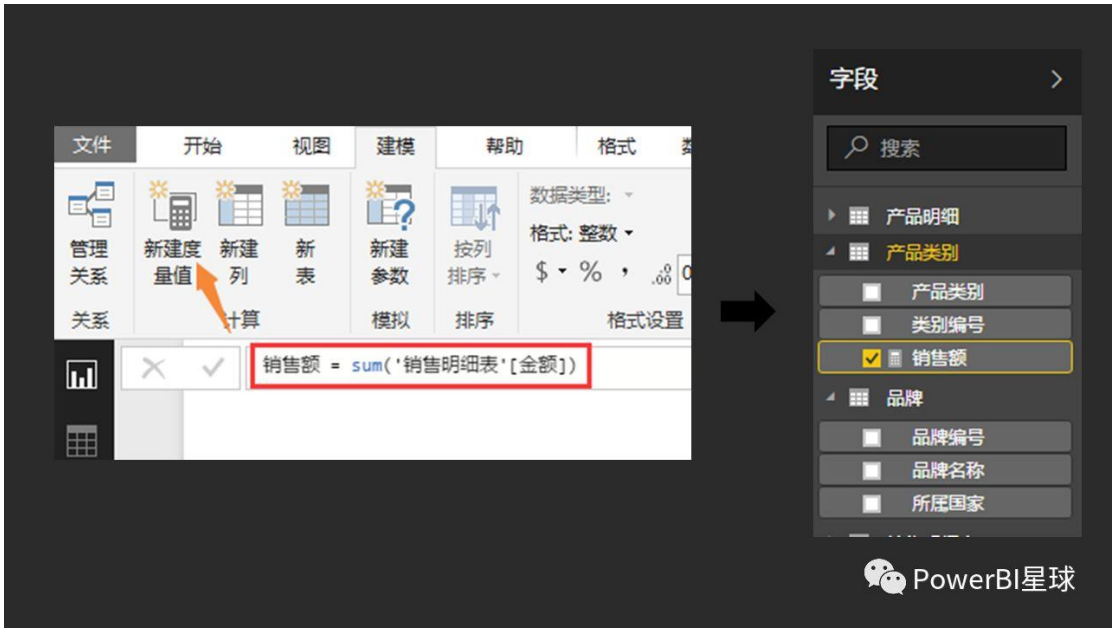
这个稍微有点抽象，以后可以根据实例来理解。

根据刚才建立的数据模型，可以做一下分析，比如统计各品牌产品的销售额：

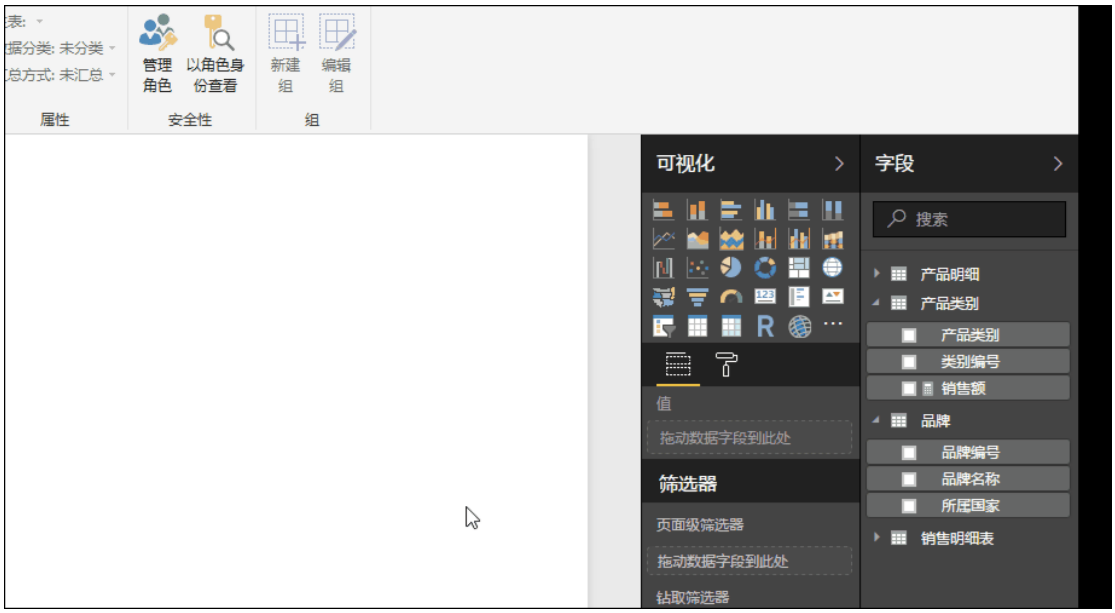
在销售明细表中并不能直接统计出按品牌的销售额，可以先建一个度量值，在建模选项卡下，点击新建度量值，公式栏输入：

```
销售额 = sum('销售明细表'[金额])
```

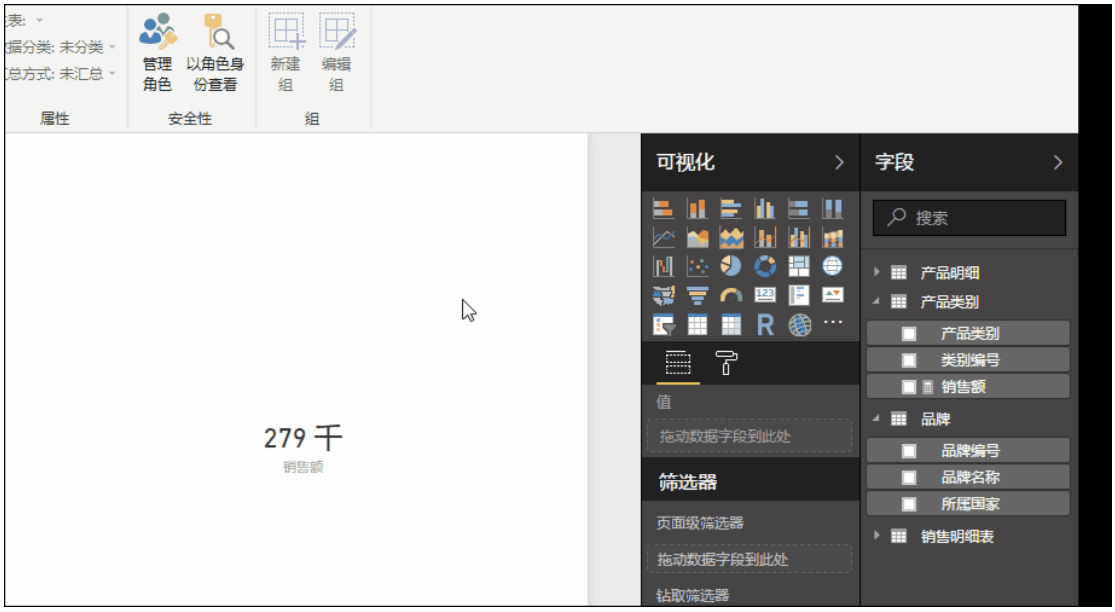
然后[销售额]这个度量值就建立了，在右边字段区可以看到。



为了在画布上直观的看到各品牌销售额，在可视化里添加"卡片图"，把度量值字段放进去，可以看到卡片图的数字出来了，



这个数字是整体销售金额，因为还没有做任何筛选，为了看出各品牌的销售金额，现在添加一个品牌的切片器，



点击不同的品牌，数值跟着变化，通过这个例子，可以看出：

- 展现的数字并不是一个表得出的，根据之前建立的关系模型，销售明细表中的数据被品牌表中的[品牌名称]字段所筛选，展现出来不同品牌的销售额，这就是数据模型的威力。
- 品牌销售额是通过[销售额]这个度量值，加入到卡片图中，并可与切片器交互，展现不同的数据。

通过这个实例，还看到了以前从未见过的的概念：度量值，这可以说是 PowerBI 数据建模的灵魂，创建度量值的公式称为 **DAX** 公式（看起来和 Excel 公式非常相似），刚才创建的这个度量值只是一个简单的 sum 函数，并没有任何的过滤条件，但是却可以根据切片器的筛选而展现不同的数值，所以度量值被称为移动的公式，这里只是简单介绍，有个印象即可。

学习数据建模的更多知识，可以说都是依据度量值的逻辑以及建立度量值的 DAX 公式来展开，是下一步学习 Power BI 的重点。

这篇文章简单介绍了数据建模的概念，并引出下一步学习的重点，Power BI 的大门已经推开，后面的学习会更加精彩。

理解度量值

本文学习 PowerBI 最重要的概念：度量值

初学 Power BI 一般都会对度量值比较困惑，毕竟对长期接触 Excel 的人来说，这是个全新的概念，但一旦理解了她，肯定会立即喜欢上。

简单来说，度量值是用 DAX 公式创建一个虚拟字段的数据值，她不改变源数据，也不改变数据模型，如果你不在报表上使用她，甚至不知道她是什么样子的，而一旦被拖拽到报表上，便发挥巨大的作用，她可以随着切片器的筛选而闪转腾挪、变化万端，所以度量值一般在报表交互时使用。

说了这么多，你也许觉得更晕，不过没问题，这个概念不是靠语言能解释的很清楚的，在 PowerBI 报表上体验一下可能立即就明白了。

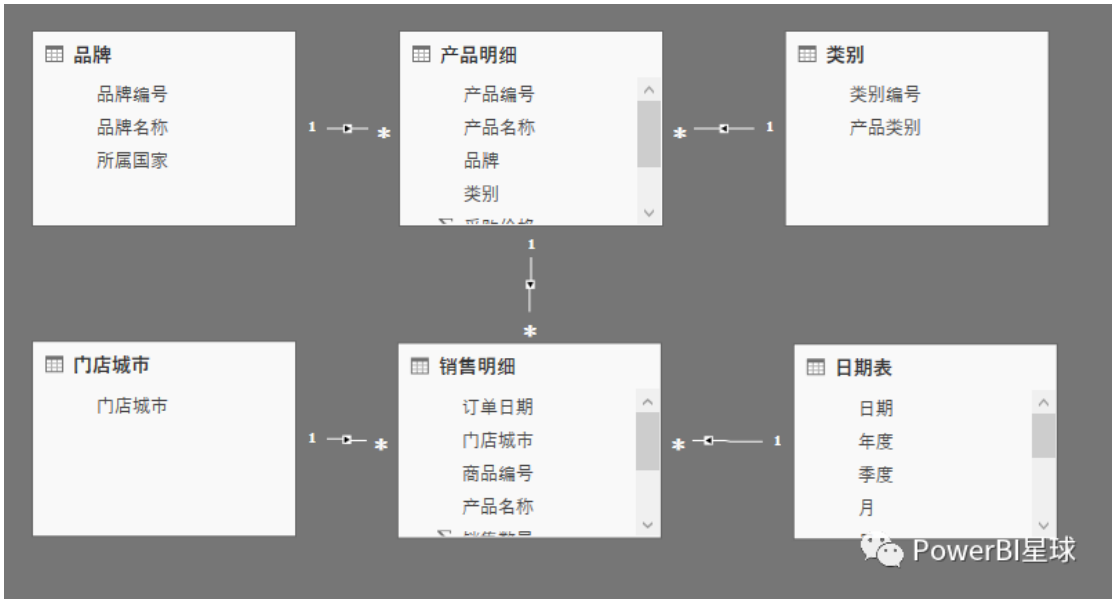
案例数据：

全国连锁电子产品专卖店的 2016 至 2017 年的销售明细，销售产品有三类：手机、电脑、平板，每一类又分别来自三个品牌：小米、苹果、三星，这个店销售的产品共计 9 个，全国 8 个城市门店设有门店。

订单日期	门店城市	商品编号	产品名称	销售数量	零售价	销售额
2016年1月1日	北京市	1001	小米手机	12	2000	24000
2016年1月2日	北京市	1001	小米手机	27	2000	54000
2016年1月5日	北京市	2001	苹果手机	9	8000	72000
2016年1月7日	北京市	1001	小米手机	21	2000	42000
2016年1月9日	北京市	1001	小米手机	15	2000	30000
2016年1月10日	北京市	2001	苹果手机	6	8000	48000
2016年1月11日	北京市	1001	小米手机	12	2000	24000
2016年1月12日	北京市	1001	小米手机	27	2000	54000
2016年1月12日	北京市	1001	小米手机	18	2000	36000
2016年1月14日	北京市	1001	小米手机	12	2000	24000
2016年1月15日	北京市	2001	苹果手机	9	8000	72000
2016年1月15日	北京市	2001	苹果手机	24	8000	192000
2016年1月16日	北京市	2001	苹果手机	15	8000	120000
2016年1月16日	北京市	1001	小米手机	12	2000	24000
2016年1月17日	北京市	1001	小米手机	21	2000	42000

PowerBI星球

还有几个维度表和一个日期表，模型关系图如下：

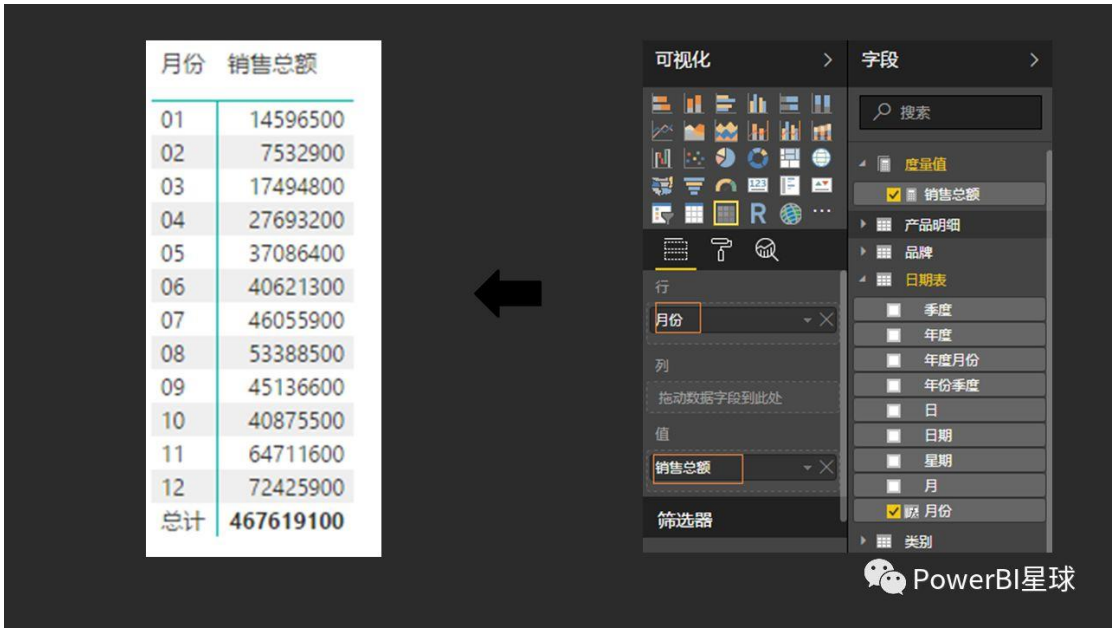


我们现在根据这个模型分析累计销售同比增长率。

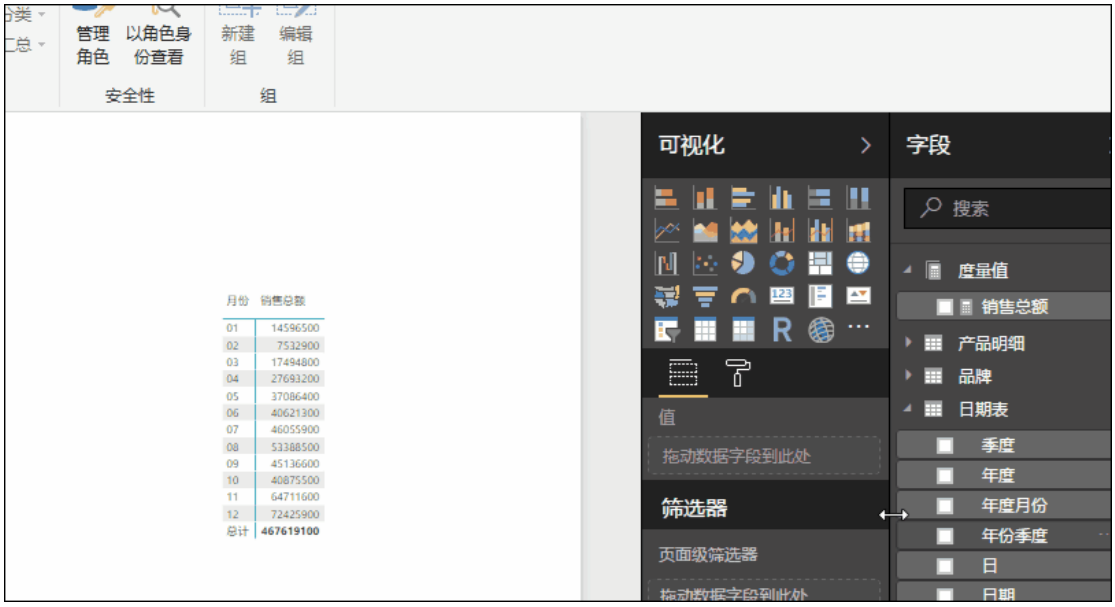
先新建一个度量值[销售总额]：

```
销售总额 = sum('销售明细'[销售额])
```

然后在画布上添加矩阵表格可视化组件，把[月份]和刚创建的度量值分别放入行和值，每月的销售总额就出来了：



因为没有加入年度筛选，这时候的每月销售额实际上是 2016 和 2017 年的两个月份的合计额，所以现在加入一个年度切片器，



这个时候，选中年份，这个销售总额就是该年度当月的销售总额了。

因为我们是要分析本年累计销售金额的同比增长率，那么需要先得到本年累计销售金额和上年累计销售金额，新建两个度量值：

```
本年累计销售额 = TOTALYTD([销售总额],'日期表'[日期])

上年累计销售额 = TOTALYTD([销售总],SAMEPERIODLASTYEAR('日期表'[日期]))
```

这里用了两个 DAX 函数，TOTALYTD 函数是统计从年初到本月的金额，SAMEPERIODLASTYEAR 返回的是上年同期的日期表，关于 DAX 函数先有个印象就行，后面会专门介绍。

把这两个度量值也放入矩阵中，选中 2017 年，本年和上年截至每个月份的累计销售额就出来了，

月份	销售总额	本年累计销售额	上年累计销售额
01	12569100	12569100	2027400
02	5604500	18173600	3955800
03	12760800	30934400	8689800
04	14661800	45596200	21721200
05	19456000	65052200	39351600
06	21901600	86953800	58071300
07	24112400	111066200	80014800
08	25028600	136094800	108374700
09	25327400	161422200	128183900
10	25089700	186511900	143969700
11	29553700	216065600	179127600
12	30004500	246070100	221549000
总计	**246070100**	**246070100**	**221549000**

然后再添加个度量值[同比增长率]，

```
累计同比增长率 = divide([本年累计销售额],[上年累计销售额])-1
```

把该度量值拖入矩阵，同比增长率就出来了，这就是利用这个数据模型实现的结果。

月份	销售总额	本年累计销售额	上年累计销售额	累计同比增长率
01	12569100	12569100	2027400	5.20
02	5604500	18173600	3955800	3.59
03	12760800	30934400	8689800	2.56
04	14661800	45596200	21721200	1.10
05	19456000	65052200	39351600	0.65
06	21901600	86953800	58071300	0.50
07	24112400	111066200	80014800	0.39
08	25028600	136094800	108374700	0.26
09	25327400	161422200	128183900	0.26
10	25089700	186511900	143969700	0.30
11	29553700	216065600	179127600	0.21
12	30004500	246070100	221549000	0.11
总计	**246070100**	**246070100**	**221549000**	**0.11**

PowerBI星球

你也许觉得做个同比增长率也不是很麻烦，在 Excel 里面做也可以很快做出来，但是如果不只比较销售总额，还要分类别、分品牌来比较呢，是不是要在 Excel 里面做很多遍，况

且这只是两年的数据，如果是 20 年呢，如果在 100 个城市设有门店呢，估计用传统 Excel 技能做要累吐血也做不完吧，而在刚才建立的模型里，无论有多少个维度，无论按哪个维度比较，都是点点鼠标的事：

这就是度量值的魅力，只是简单建立了四个度量值，然后就神奇的实现了多维度的比较，这要归功于度量值的最重要的特征：上下文。

上下文就是度量值所处的环境，筛选表的行列标签、切片器的选中，都是度量值的上下文，比如北京 2017 年截至 5 月的苹果手机累计销售额 3424000，它的上下文就是下面这 5 个维度：

[城市]="北京市"

[品牌]="苹果"

[类别]="手机"

[年度]=2017

[月份]=5 月

理解了上下文，就理解了度量值的最重要的特征，因此她被称为移动的公式，随着上下文环境的不同而展示不同的数据，度量值另外还有两个主要特征：

- 度量值不浪费内存，只有被拖到图表上才执行运算，如果数据量非常大的时候这点非常有利
- 度量值可以循环使用，比如上面的建立的度量值：

  本年累计销售额 = TOTALYTD([销售总额],'日期表'[日期])

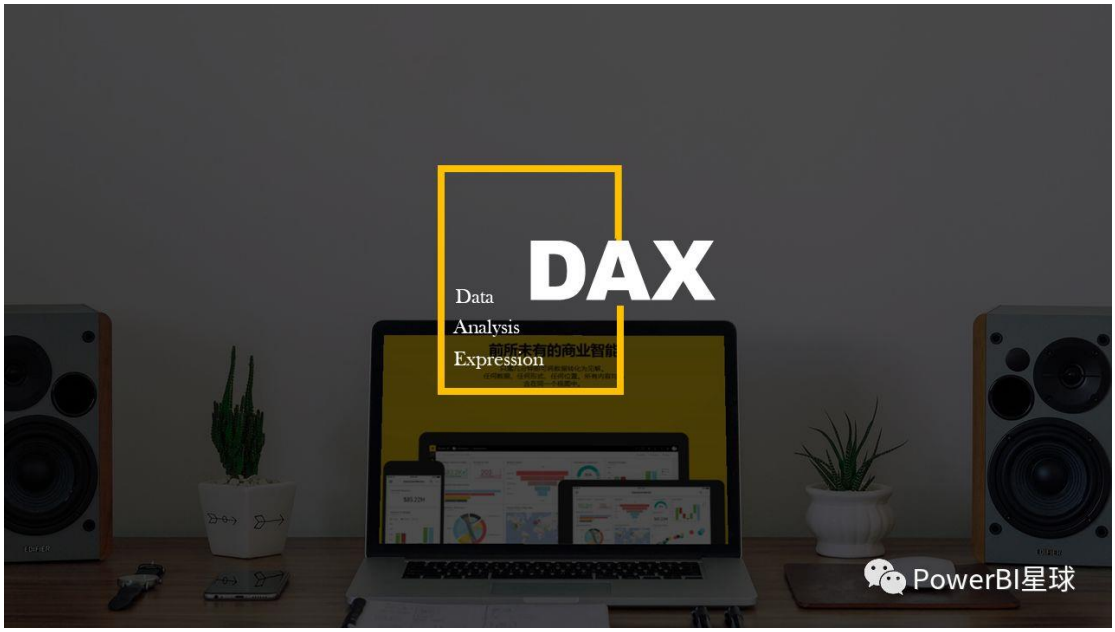
  累计同比增长率 = divide([本年累计销售额],[上年累计销售额])-1,

就是直接调用之前建立好的度量值，所以以后在模型中新建度量值的时候，推荐从最简单的度量值开始建。

看到这里是不是对度量值的认识比较清晰了，如果动手做一遍会更加理解上下文。建立合理的数据模型，加上灵活运用度量值，以后无论遇到什么数据分析都不是难题了。

度量值的创建离不开函数，这里用到的函数称为 DAX，后面会主要介绍，并通过 DAX 实例来加深对度量值的理解。

DAX 是什么?



DAX 是英文 Data Analysis Expression 的缩写，意思是数据分析表达式，从名称上就可以看出，DAX 公式是用作数据分析的，事实上也确实如此，从数据分析层面认识 DAX 会更有助于我们理解它.

---

01 | 认识 DAX

数据分析就是从茫茫数据中提取有用的信息，执行一定的运算，形成结论的过程，而 DAX 的主要功能正是查询和运算，DAX 查询函数负责筛选出有用的数据集合，然后利用 DAX 的聚合函数执行计算。

从前面的文章我们知道，度量值的运算依赖上下文，上下文又分为外部上下文和内部上下文，外部上下文就是外部可以看得见的筛选：标签和切片器，而内部上下文就是创建度量

值的 DAX 公式，它的查询筛选函数可以扩大、限制或者重置外部上下文。利用外部上下文和 DAX 的有效配合，度量值的灵活性大大增强，可以筛选出真正需要的数据集合并执行聚合运算。如果说度量值是 Power BI 数据建模的灵魂，那么 DAX 就是度量值的灵魂。

02 | DAX 的使用范围

DAX 可以在 Power BI Desktop 的数据建模和 Power Pivot 中使用，除了可以创建度量值，还可以新建列，比如想把这个日期表上的月用两位数字表示，即 7 月用 07 表示，新建列然后编辑栏输入：

```
月份 = format('日期表'[日期],"MM")
```

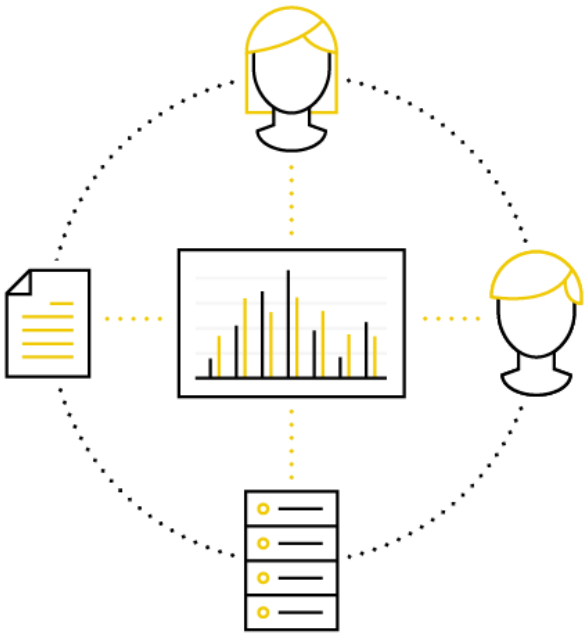
然后就增加了一列[月份]，

日期	年度	季度	月	日	年度月份	年份季度	星期	月份
2016年7月1日	2016	3	7	1	201607	2016Q3	5	07
2016年7月2日	2016	3	7	2	201607	2016Q3	6	07
2016年7月3日	2016	3	7	3	201607	2016Q3	7	07
2016年7月4日	2016	3	7	4	201607	2016Q3	1	07
2016年7月5日	2016	3	7	5	201607	2016Q3	2	07
2016年7月6日	2016	3	7	6	201607	2016Q3	3	07
2016年7月7日	2016	3	7	7	201607	2016Q3	4	07
2016年7月8日	2016	3	7	8	201607	2016Q3	5	07
2016年7月9日	2016	3	7	9	201607	2016Q3	6	07
2016年7月10日	2016	3	7	10	201607	2016Q3	7	07
2016年7月11日	2016	3	7	11	201607	2016Q3	1	07
2016年7月12日	2016	3	7	12	201607	2016Q3	2	07
2016年7月13日	2016	3	7	13	201607	2016Q3	3	07
2016年7月14日	2016	3	7	14	201607	2016Q3	4	07

不过除非特别有必要，不建议用新建列的方式做数据丰富，这样更占用内存，如果想增加一列，可以在源数据上，回到查询编辑器里面增加一列然后上载到数据模型中使用。

03 | DAX 参数的基本格式

- 表名用单引号' '括着 // '日期表'
- 字段用中括号[ ]括着 //[日期]
- 度量值也是用中括号[ ]
- 引用字段始终要包含表名，以和度量值区分开



04 | DAX 常用函数

聚合函数

- SUM
- AVERAGE
- MIN
- MAX

这几个函数的使用和在 Excel 中完全一致，DAX 中还有一类特有的函数非常有用，和这几个函数很像，后面加个 X，

- SUMX
- AVERAGEX
- MINX
- MAXX
- RANKX

这几个函数可以循环访问表的每一行，并执行计算，所以也被称为迭代函数。

其他常见的聚合函数：

- COUNT:计数
- COUNTROWS：计算行数
- DISTINCTCOUNT:计算不重复值的个数

时间智能函数

- PREVIOUSYEAR/Q/M/D:上一年/季/月/日

- NEXTYEAR/Q/M/D:下一年/季/月/日
- TOTALYTD/QTD/MTD:年/季/月初至今
- SAMEPERIODLASTYEAR:上年同期
- PARALLELPERIOD:上一期
- DATESINPERIOD:指定期间的日期

关于时间智能函数，会专门写篇文章介绍，利用它可以灵活的筛选出一段我们需要的时间区间，做同比、环比、滚动预测、移动平均等数据分析时，都会用到这类函数。

筛选函数

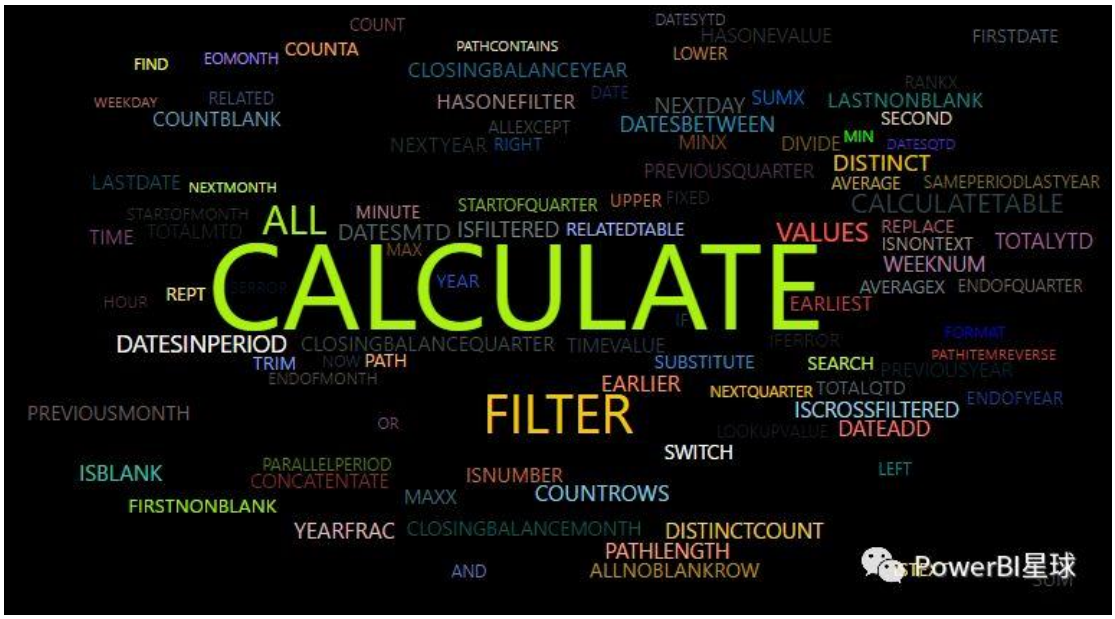
- FILTER:筛选
- ALL:所有值，可以清除筛选
- ALLEXCEPT:保留指定列
- VALUES:返回不重复值

这几个函数，就是典型的 DAX 查询函数，通过筛选来操纵上下文的范围。

最后放出 DAX 最重要也是最常用的函数:**CALCULATE**,这个函数简直就是 DAX 本身，学好这个函数就基本上掌握了 DAX,它值得专门写篇文章好好介绍。

【第 4 天 | 了解常用的 DAX 函数】

PowerBI 中最重要的函数 CALCULATE



介绍 DAX 的时候，特别强调过一个重要的函数:CALCULATE,本文就来揭秘这个函数的计算原理以及它是如何影响上下文的。

CALCULATE 的语法结构

语法：

CALCULATE(<expression>,<filter1>,<filter2>...)

- 第一个参数是计算表达式，可以执行各种聚合运算
- 从第二个参数开始，是一系列筛选条件，可以为空；如果多个筛选条件，用逗号分隔
- 所有晒选条件的交集形成最终的筛选数据集合
- 根据筛选出的数据集合执行第一个参数的聚合运算并返回运算结果

前文提到 DAX 函数可以更改外部上下文，现在通过实例来理解 DAX 中最精髓的函数 CALCULATE 的计算原理，并看看它是如何更改外部上下文的。

CALCULATE 应用实例

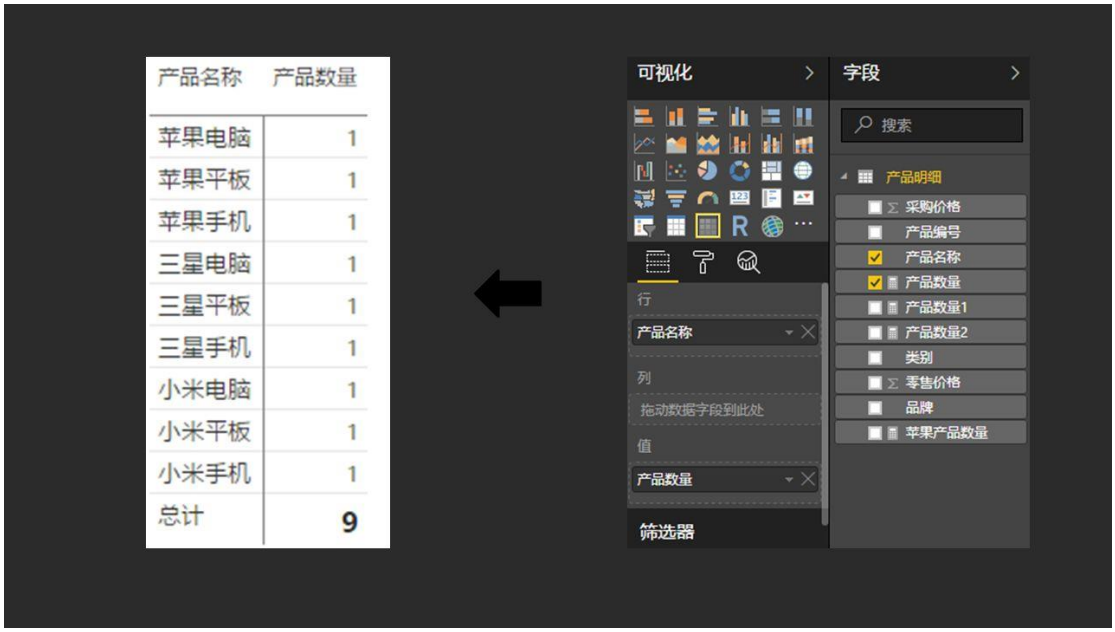
导入下面这个产品明细表：

产品编号	产品名称	品牌	类别	采购价格	零售价格
1001	小米手机	小米	手机	1800	2000
1002	小米电脑	小米	电脑	4700	5200
1003	小米平板	小米	平板	1300	1500
2001	苹果手机	苹果	手机	6000	8000
2002	苹果电脑	苹果	电脑	8000	10000
2003	苹果平板	苹果	平板	2200	3000
3001	三星手机	三星	手机	3600	4200
3002	三星电脑	三星	电脑	3500	4000
3003	三星平板	三星	平板	1700	2000

新建一个度量值求每种产品的数量：

```
产品数量 = COUNTROWS('产品明细')
```

因为每种产品的只有 1 行，所以求产品明细表的行数就相当于求各种产品的数量，把产品名称和该度量值拖拽入矩阵表，



这里外部上下文就是表格每行的行标签。

01 | 筛选条件为空，不影响外部上下文

现在开始用 CALCULATE 函数创建一个度量值：

```
产品数量 1 = CALCULATE([产品数量])
```

只是用了第一个参数，筛选条件为空，因为没有内部筛选所以完全依赖外部上下文，出来的结果也和原度量值一致。

产品名称	产品数量	产品数量1
苹果电脑	1	1
苹果平板	1	1
苹果手机	1	1
三星电脑	1	1
三星平板	1	1
三星手机	1	1
小米电脑	1	1
小米平板	1	1
小米手机	1	1
总计	**9**	**9**

另外，介绍 CALCULATE 语法的时候说，第一个参数为聚合运算表达式，为什么这里没有用聚合函数而只用了一个度量值呢？这是因为度量值[产品数量]本身就是一个聚合函数运算，实际上度量值[产品数量 1]等同于这个：

```
产品数量 1 = CALCULATE(COUNTROWS('产品明细'))
```

DAX 函数可以直接引用已经创建好的度量值，可以使 DAX 函数看起来更简洁、更具可读性，这也是建议从最简单的度量值开始建的原因。

02 | 添加限制条件，缩小上下文

建一个度量值[产品数量 2]，

```
产品数量 2 = CALCULATE([产品数量],'产品明细'[品牌]="苹果")
```

产品名称	产品数量	产品数量1	产品数量2
苹果电脑	1	1	1
苹果平板	1	1	1
苹果手机	1	1	1
三星电脑	1	1	
三星平板	1	1	
三星手机	1	1	
小米电脑	1	1	
小米平板	1	1	
小米手机	1	1	
总计	**9**	**9**	**3**

发现只有苹果的产品计数显示出来，而其他品牌的数据没有了，这是因为 CALCULATE 的第二个参数的限制，只筛选品牌为"苹果"的，限制了外部的上下文，非苹果的产品都不再运算。

03 | 结合 ALL 函数，扩大上下文

新建度量值[产品数量 3]，

```
产品数量 3 = CALCULATE([产品数量],ALL('产品明细'))
```

产品名称	产品数量	产品数量1	产品数量2	产品数量3
苹果电脑	1	1	1	9
苹果平板	1	1	1	9
苹果手机	1	1	1	9
三星电脑	1	1		9
三星平板	1	1		9
三星手机	1	1		9
小米电脑	1	1		9
小米平板	1	1		9
小米手机	1	1		9
总计	**9**	**9**	**3**	**9**

这次的数据居然是所有产品的数量，这是因为筛选条件使用了 ALL 函数，ALL('产品明细')的意思是清除产品明细表里的所有筛选，外部筛选器不起作用了，每行统计的都是该表中的所有产品。

每行的数据都是 9，你可能觉得这个 ALL 函数没什么用，运算的数据没有什么意义，会误导人，实际上当然不是这样，这个数据使用的地方非常多，比如我们想计算每个产品数量占总产品数量的比重，直接写个度量值：

```
产品占比=[产品数量]/[产品数量 3]
```

产品名称	产品数量	产品数量1	产品数量2	产品数量3	产品占比
苹果电脑	1	1	1	9	0.11
苹果平板	1	1	1	9	0.11
苹果手机	1	1	1	9	0.11
三星电脑	1	1		9	0.11
三星平板	1	1		9	0.11
三星手机	1	1		9	0.11
小米电脑	1	1		9	0.11
小米平板	1	1		9	0.11
小米手机	1	1		9	0.11
总计	**9**	**9**	**3**	**9**	**1.00**

产品占比就计算出来了，这就是统计总数的一个功能。

04 | 重置上下文

新建度量值[产品数量 4]，

```
产品数量 4 = CALCULATE([产品数量],
                        all('产品明细'[产品名称]),
                        '产品明细'[类别]="手机")
```

先用 ALL 函数清除外部上下文，然后又新增了一个筛选条件，类别为"手机"的产品数量，那么结果会是什么样的呢，

产品名称	产品数量	产品数量1	产品数量2	产品数量3	产品数量4
苹果电脑	1	1	1	9	3
苹果平板	1	1	1	9	3
苹果手机	1	1	1	9	3
三星电脑	1	1		9	3
三星平板	1	1		9	3
三星手机	1	1		9	3
小米电脑	1	1		9	3
小米平板	1	1		9	3
小米手机	1	1		9	3
总计	**9**	**9**	**3**	**9**	**3**

每一行产品的数量都是 3，正好符合建立这个度量值的逻辑，被 ALL 清除行标签的外部筛选后，从全部产品中统计品类为"手机"的产品的数量，所以每行都返回 3.

通过以上几个简单的例子，可以领会到 CALCULATE 的计算逻辑，通过从第二个参数开始的筛选条件，得到一个数据集合，并利用第一个参数执行聚合运算，这不就是 **DAX** 要实现的功能：提取有用数据并执行聚合运算吗，所以说 **CALCULATE** 几乎就是 **DAX** 本身，它就是实现 **DAX** 功能的引擎，并能灵活的操控外部上下文，后面的数据分析也都离不开 CALCULATE 的身影。

FILTER 函数

介绍过 CALCULATE 函数之后，有必要再介绍它的最佳搭档：FILTER 函数。

CALCULATE 函数的第二个及之后的参数是筛选条件，而 FILTER 函数正是为筛选而生。

FILTER 语法

FILTER(<table>,<filter>)

- 第一个参数<table>是要筛选的表
- 第二个参数<filter>是筛选条件
- 返回的是一张表，不能单独使用，需要与其他函数结合使用

FILTER 应用

前一篇文章中介绍 CALCULATE 的时候，没有使用 FILTER 函数也实现了筛选，你可能会问，既然不用 FILTER 函数也可以，为什么还要使用 FILTER 呢?

那是因为前文中的筛选条件都是最简单的筛选，没有必要使用强大的 FILTER，其实用 FILTER 也是可以实现的，比如这个度量值[产品数量 2],

```
= CALCULATE([产品数量],'产品明细'[品牌]="苹果")
```

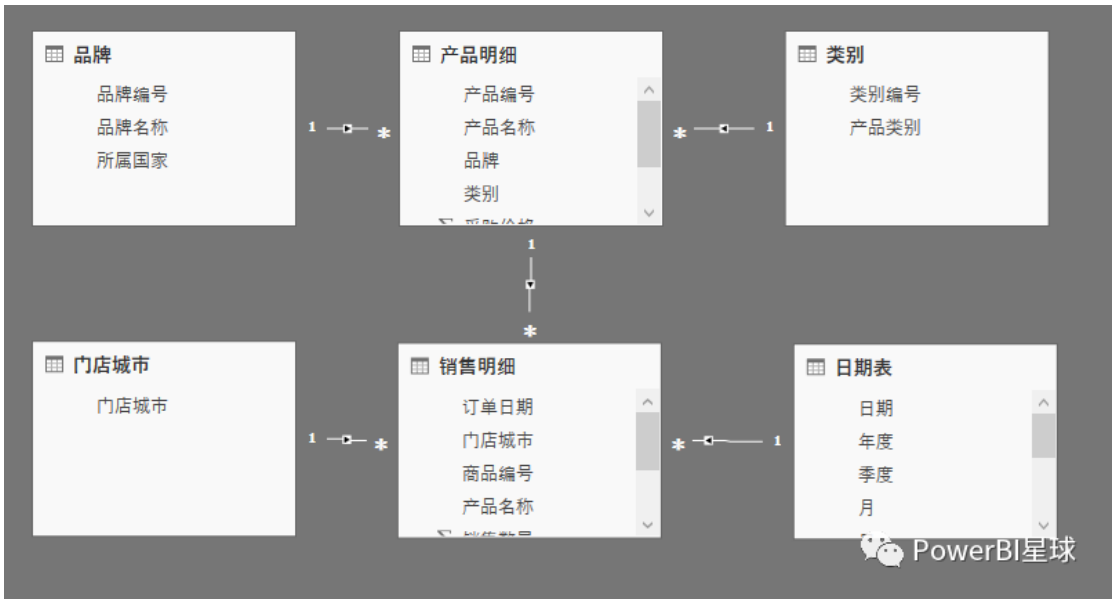
等同于:

```
= CALCULATE([产品数量],
                    FILTER(ALL('产品明细'[品牌]),
                                '产品明细'[品牌]="苹果"))
```

这两种方式返回的结果完全一致。

在这个简单的例子里看起来好像用 FILTER 写起来更复杂，不用 FILTER 的情况下显得更简洁，但是如果我们要做更复杂的运算，通过简单的布尔表达式根本无法实现，必须借助 FILTER。

依然用介绍度量值的那篇文章中的案例数据（可点击阅读原文查看），有这 6 张表，



现在想找出年销售额超过 2000 万的城市销售金额，这个利用简单的表达式已无法解决，来看 FILTER 如何实现。

第一个度量值还是之间建好的，

```
销售总额 = sum('销售明细'[销售额])
```

再建一个度量值[大于 2000 万的城市销售金额]，

```
= CALCULATE([销售总额],
                FILTER(ALL('门店城市'),
                        [销售总额]>20000000))
```

这个度量值先利用 FILTER 函数，筛选出销售大于 2000 万的城市有哪几个，然后再汇总这些城市的销售额，返回结果如下，

产品名称	销售总额	大于2000万的城市销售金额
苹果电脑	89350000	20580000
苹果平板	17532000	
苹果手机	207080000	156584000
三星电脑	20740000	
三星平板	8550000	
三星手机	44662800	
小米电脑	20144800	
小米平板	14803500	
小米手机	44756000	
总计	**467619100**	**451026000**

超过 2000 万销售额的城市的销售额总计等于 451026000，但发现上面明细的两个数字之和并不等于总计，这正是 PowerBI 中数据模型的特点，在这个表格中，每个数据都是独立运算的，和其他数字没有关系。

上面两个数字代表的是，在销售额超过 2000 万的城市中，单个产品的销售额也大于 2000 万的只有苹果电脑和苹果手机。

为什么会有产品的销售额也大于 2000 万这个筛选条件呢，不要忘了还有外部的筛选上下文，在这个表中即是行标签，这两个数字正是内部上下文和外部上下文相结合，筛选计算

的结果。

这就是 FILTER 的一个简单应用，这里要特别记住这个函数嵌套组合，

CALCULATE(...,FILTER(ALL(...),...))

做各种分析时会经常用到这个组合。

HASONEVALUE 和 SUMX 函数

上一篇文章最后留了个问题，就是计算大于 2000 万的城市销售额时，总计金额不等于明细之和，看起来比较奇怪，

产品名称	销售总额	大于2000万的城市销售金额
苹果电脑	89350000	20580000
苹果平板	17532000	
苹果手机	207080000	156584000
三星电脑	20740000	
三星平板	8550000	
三星手机	44662800	
小米电脑	20144800	
小米平板	14803500	
小米手机	44756000	
总计	**467619100**	**451026000**

其实总计和上面的这两个数字都有自己的逻辑，分别代表不同的上下文环境中的销售额，按照常规的汇总逻辑来理解二者关系是没有意义的。但是为了避免误导，有时还是需要避免出现这种状况，这里有两个解决的思路，

（一）明细项目和总计项目不同时显示

（二）让总计额等于明细项之和

下面按这两种思路分别来解决。

（一）明细项目和总计项目不同时显示

也即是让明细项的金额为空值，或者让总计为空值，这里要用到一个新函数：HASONEVALUE，是个逻辑判断函数，如果有单一值，返回 ture,否则返回 false，具体语法如下：

HASONEVALUE(<columnName>)

参数只有一个：列名，

作用：判断外部上下文中是否为该列中的唯一值，做切片器交互时十分有用。

下面就看看这个函数如何解决上面的问题。

不同时显示又分为不显示总计值和不显示明细项的值

不显示明细项的值

新建度量值[方法 1]

```
= IF(HASONEVALUE('产品明细'[产品名称]),
    BLANK(),
    [大于 2000 万的城市销售金额])
```

这个度量值的意思是，如果外部上下文在[产品名称]中，用空值表示，否则运算度量值，结果显示如下，

产品名称	销售总额	大于2000万的城市销售金额	方法1
苹果电脑	89350000	20580000	
苹果平板	17532000		
苹果手机	207080000	156584000	
三星电脑	20740000		
三星平板	8550000		
三星手机	44662800		
小米电脑	20144800		
小米平板	14803500		
小米手机	44756000		
总计	**467619100**	**451026000**	**451026000**

如果只想要销售额大于 2000 万的城市的汇总销售金额，可以用这种方法。

不显示总计值

如果想要总计值为空，和上面的度量值类似，BLANK 换个位置就行了，新建度量值[方法2]，

```
= IF(HASONEVALUE('产品明细'[产品名称]),
    [大于 2000 万的城市销售金额],
    BLANK())
```

结果如下，和我们期望的一致。

产品名称	销售总额	大于2000万的城市销售金额	方法1	方法2
苹果电脑	89350000	20580000		20580000
苹果平板	17532000			
苹果手机	207080000	156584000		156584000
三星电脑	20740000			
三星平板	8550000			
三星手机	44662800			
小米电脑	20144800			
小米平板	14803500			
小米手机	44756000			
总计	**467619100**	**451026000**	**451026000**	

（二）让总计额等于明细项之和

实现这种方式还需要再放出一个函数：SUMX.

SUMX(<table>, <expression>)

第一个参数为被运算的表 table

第二个参数是对表中的每一行计算的表达式

之前也提到过，这是一个迭代函数，可以对表进行逐行运算，用该函数建一个度量值[方法3],

```
= SUMX('产品明细',
                [大于 2000 万的城市销售金额])
```

结果如下，总计正好等于明细项之和，

产品名称	销售总额	大于2000万的城市销售金额	方法1	方法2	方法3
苹果电脑	89350000	20580000		20580000	20580000
苹果平板	17532000				
苹果手机	207080000	156584000		156584000	156584000
三星电脑	20740000				
三星平板	8550000				
三星手机	44662800				
小米电脑	20144800				
小米平板	14803500				
小米手机	44756000				
总计	**467619100**	**451026000**	**451026000**		**177164000**

SUMX 函数十分强大，有的时候甚至可以代替 CALCULATE，比如用 SUMX 可以一步实现方法 3 的效果，而无需借助 CALCULATE 建立的度量值 [大于 2000 万的城市销售金额]。

使用 SUMX 新建度量值[SUMX 方法]，

```
= SUMX('产品明细',
        SUMX(FILTER('门店城市', [销售总额]>20000000),[销售总额])
)
```

结果如下，

产品名称	销售总额	大于2000万的城市销售金额	方法1	方法2	方法3	SUMX方法
苹果电脑	89350000	20580000		20580000	20580000	20580000
苹果平板	17532000					
苹果手机	207080000	156584000		156584000	156584000	156584000
三星电脑	20740000					
三星平板	8550000					
三星手机	44662800					
小米电脑	20144800					
小米平板	14803500					
小米手机	44756000					
总计	**467619100**	**451026000**	**451026000**		**177164000**	**177164000**

通过 SUMX 嵌套实现的效果和之前的方法完全一致,其中内层嵌套的 SUMX 公式和之前用 CALCULATE 建的度量值是等效的。不过 SUMX 的计算原理是逐行运算，如果数据量特别大，这种方式对内存的消耗比较严重，使用这种方式候需要慎重。

通过以上这几种方法，实现了总计和明细之和不相等的问题，实际分析中遇到相似的情况，可根据具体逻辑关系和展现需求选择不同的方式来处理。

EARLIER 函数

前面利用 PowerBI 做数据分析的时候都是对整列的字段进行操作，并没有做更细化的分析，比如分析数据的每一行、提取某一行的数据，这在 Excel 中很容易实现，因为 Excel 公式是对单元格操作。那么在 PowerBI 中能不能按行分析呢，答案当然是肯定的，并且比 Excel 更加灵活，借助这个函数:**EARLIER.**

EARLIER 语法

EARLIER(<column>, <number>)

- 第一个参数是列名
- 第二个参数一般可省略
- EARLIER 函数提取本行对应的该列的值，实际上就是提取本行和参数列交叉的单元格

这个函数是指定行上下文重要工具，下面靠几个实例来理解这个函数。

EARLIER 应用实例

有一张订单表：

序号	订单日期	产品名称	销售数量	销售金额
1	2016/1/1	小米手机	12	24000
2	2016/1/2	小米手机	27	54000
3	2016/1/5	苹果手机	9	72000
4	2016/1/7	小米手机	21	42000
5	2016/1/9	小米手机	15	30000
6	2016/1/10	苹果手机	6	48000
7	2016/1/11	小米手机	12	24000
8	2016/1/12	苹果手机	7	56000
9	2016/1/12	小米手机	18	36000
10	2016/1/14	小米手机	12	24000
11	2016/1/15	苹果手机	9	72000

利用 EARLIER 做以下几种分析：

01 | 求两个订单的时间间隔

相当于用下一个订单的日期减去当前订单的日期，为了相减的方便，新建列[下个订单日期]，先把下一行的订单日期提取过来，输入 DAX 公式：

```
= SUMX(FILTER('订单表','订单表'[序号]=
                        EARLIER('订单表'[序号])+1),
                '订单表'[订单日期])
```

这个公式的意思是，利用 EARLIER 获取当前行的序号，然后找到当前序号+1 的那一行的订

单日期，结果如下，

序号	订单日期	产品名称	销售数量	销售金额	下个订单日期
1	2016/1/1	小米手机	12	24000	2016/1/2
2	2016/1/2	小米手机	27	54000	2016/1/5
3	2016/1/5	苹果手机	9	72000	2016/1/7
4	2016/1/7	小米手机	21	42000	2016/1/9
5	2016/1/9	小米手机	15	30000	2016/1/10
6	2016/1/10	苹果手机	6	48000	2016/1/11
7	2016/1/11	小米手机	12	24000	2016/1/12
8	2016/1/12	苹果手机	7	56000	2016/1/12
9	2016/1/12	小米手机	18	36000	2016/1/14
10	2016/1/14	小米手机	12	24000	2016/1/15
11	2016/1/15	苹果手机	9	72000	

然后新建一列，用两列日期相减得到间隔的天数，

```
间隔 = IF([下个订单日期]=BLANK(),
                BLANK(),
                [下个订单日期]-[订单日期])
```

这里用 IF 判断主要因为最后一列为空值，避免出现不合理的数值，

序号	订单日期	产品名称	销售数量	销售金额	下个订单日期	间隔
1	2016/1/1	小米手机	12	24000	2016/1/2	1
2	2016/1/2	小米手机	27	54000	2016/1/5	3
3	2016/1/5	苹果手机	9	72000	2016/1/7	2
4	2016/1/7	小米手机	21	42000	2016/1/9	2
5	2016/1/9	小米手机	15	30000	2016/1/10	1
6	2016/1/10	苹果手机	6	48000	2016/1/11	1
7	2016/1/11	小米手机	12	24000	2016/1/12	1
8	2016/1/12	苹果手机	7	56000	2016/1/12	0
9	2016/1/12	小米手机	18	36000	2016/1/14	2
10	2016/1/14	小米手机	12	24000	2016/1/15	1
11	2016/1/15	苹果手机	9	72000		

02 | 求每个订单日期的累计销售金额

新建列[累计销售额]，

```
= SUMX (FILTER('订单表','订单表'[序号]<=EARLIER('订单表'[序号])),
        '订单表'[销售金额])
```

序号	订单日期	产品名称	销售数量	销售金额	累计销售额
1	2016/1/1	小米手机	12	24000	24000
2	2016/1/2	小米手机	27	54000	78000
3	2016/1/5	苹果手机	9	72000	150000
4	2016/1/7	小米手机	21	42000	192000
5	2016/1/9	小米手机	15	30000	222000
6	2016/1/10	苹果手机	6	48000	270000
7	2016/1/11	小米手机	12	24000	294000
8	2016/1/12	苹果手机	7	56000	350000
9	2016/1/12	小米手机	18	36000	386000
10	2016/1/14	小米手机	12	24000	410000
11	2016/1/15	苹果手机	9	72000	482000

利用 EARLIER 求当前行的序号，然后把小于等于当前序号的所有行的销售额累加。

03 | 求截至目前订单，每种产品的销量

新建列[产品累计销量]，

```
= SUMX(FILTER(
        '订单表','订单表'[序号]<=EARLIER('订单表'[序号])
        &&'订单表'[产品名称]=EARLIER('订单表'[产品名称])),
    '订单表'[销售数量])
```

不仅利用 EARLIER 筛选小于当前行的序号，还利用它求得当前行的产品名称，然后同时符合这两个条件的销量才累加，结果正是我们期望的，

序号	订单日期	产品名称	销售数量	销售金额	产品累计销量
1	2016/1/1	小米手机	12	24000	12
2	2016/1/2	小米手机	27	54000	39
3	2016/1/5	苹果手机	9	72000	9
4	2016/1/7	小米手机	21	42000	60
5	2016/1/9	小米手机	15	30000	75
6	2016/1/10	苹果手机	6	48000	15
7	2016/1/11	小米手机	12	24000	87
8	2016/1/12	苹果手机	7	56000	22
9	2016/1/12	小米手机	18	36000	105
10	2016/1/14	小米手机	12	24000	117
11	2016/1/15	苹果手机	9	72000	31

以上几个实例都是先获得本行的记录，然后做各种聚合预算，本行的记录可以称为行上下文，EARLIER 函数正是获取行上下文的最好工具之一。

【第 5 天 | 了解可视化的制作】

通过三个简单的图表，学习在 **PowerBI** 中如何制作可视化

使用 **PowerBI** 制作柱形图

前面通过大量时间学习那些可能会让你感觉枯燥的数据处理、数据建模等内容，到这里我们开始涉足 PowerBI 中最让人心动的环节：数据可视化。

第一篇先在 PowerBI 中制作常见的柱形图，简单来说，柱形图就是利用水平的柱子表示不同分类数据的大小，与之类似的是条形图，它就是竖的柱形图，或者说把柱形图顺时针转

动 90 度就成了条形图，使用以及作图方式类似，因此本文介绍的柱形图的各项设置也适用于条形图，对它不再作单独介绍。

在 PowerBI 的可视化组件中，第一排全是这柱形图和条形图，

微软这么安排当然不是偶然的，因为这些图表的应用实在是太常用了，常用到简直不能称为数据可视化。（可视化不应该是玫瑰图、弦图、桑基图等第一眼看不懂只觉得很炫酷的图吗？什么，弦图、桑基图没听说过？以后会介绍的）

但不能因为柱形图很大众化很普通就轻视它，实际上，柱形图加上近亲条形图大约可以占据图表展示的半壁江山，也就是一半以上的图表用这些柱子展示就够了，并且通过适当的设置，柱形图也可以有高颜值哦。

从第一排图表类型可以看出，柱形图又分为：

堆积柱形图

簇状柱形图

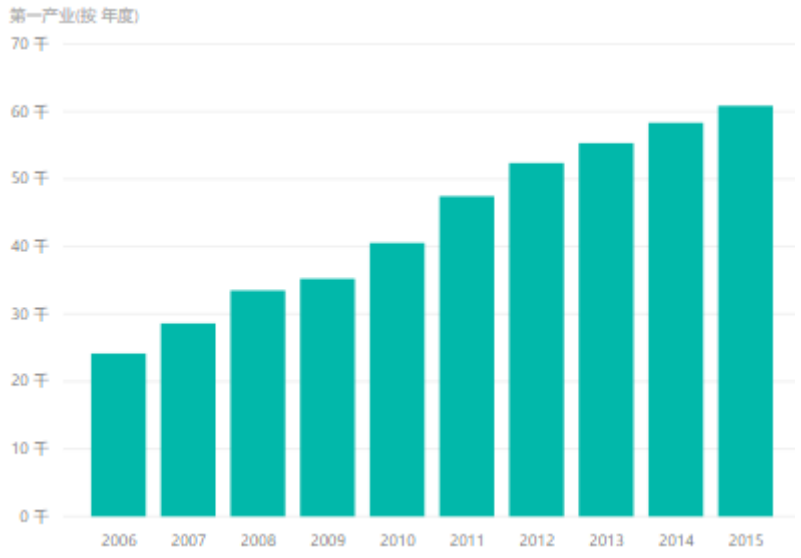
百分比堆积柱形图

可能对这几类柱形图还分不清楚，下面就来制作一个柱形图实际看看分别是什么样的。

有一份整理好的 2006-2015 年各省市三个产业的产值表，

地区	年度	第一产业	第二产业	第三产业
甘肃省	2015	954.09	2494.77	3341.46
青海省	2015	208.93	1207.31	1000.81
宁夏回族自治区	2015	237.76	1379.6	1294.41
新疆维吾尔自治区	2015	1559.08	3596.4	4169.32
北京市	2014	158.99	4544.8	16627.04
天津市	2014	199.9	7731.85	7795.18
河北省	2014	3447.46	15012.85	10960.84
山西省	2014	788.89	6293.91	5678.69
内蒙古自治区	2014	1627.85	9119.79	7022.55
辽宁省	2014	2285.75	14384.64	11956.19

在画布上放入堆积柱形图，[年度]字段拖入轴中，[第一产业]拖入到值上，一个柱形图就出来了，

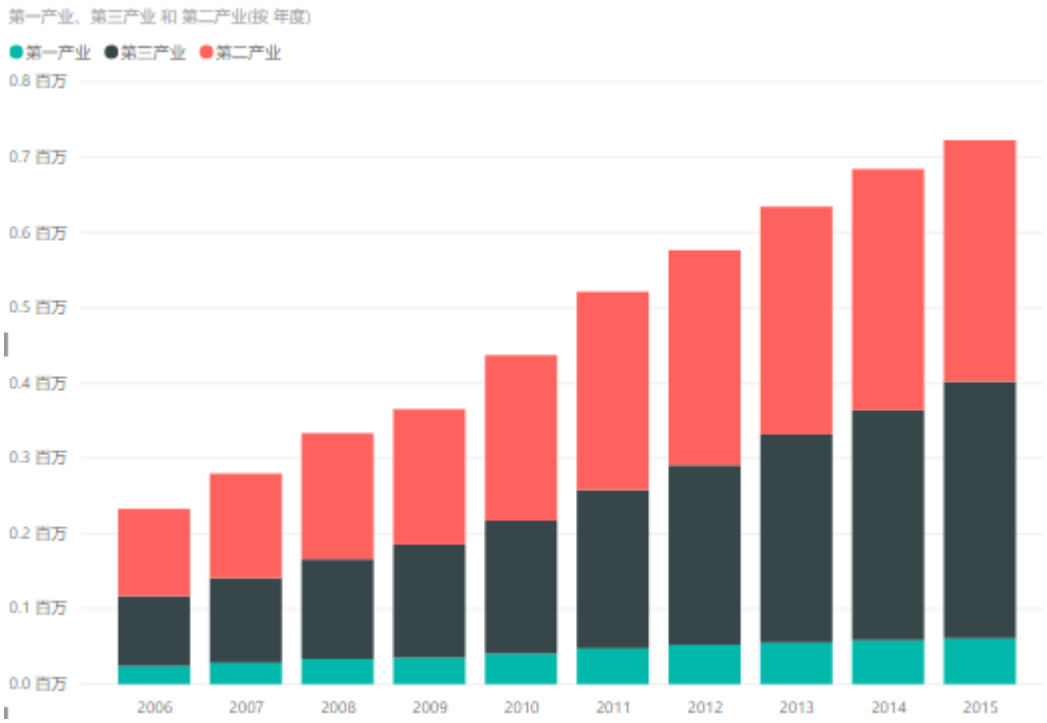


▲柱形图

因为只有一个序列数值，所以堆积柱形图就是普通的柱形图，加入更多序列才显示出堆积的效果，比如把字段[第二产业]、[第三产业]也拖入到值中，

在这个框里拖动各个字段上下的位置，可以改变图表中各字段数值显示的顺序。

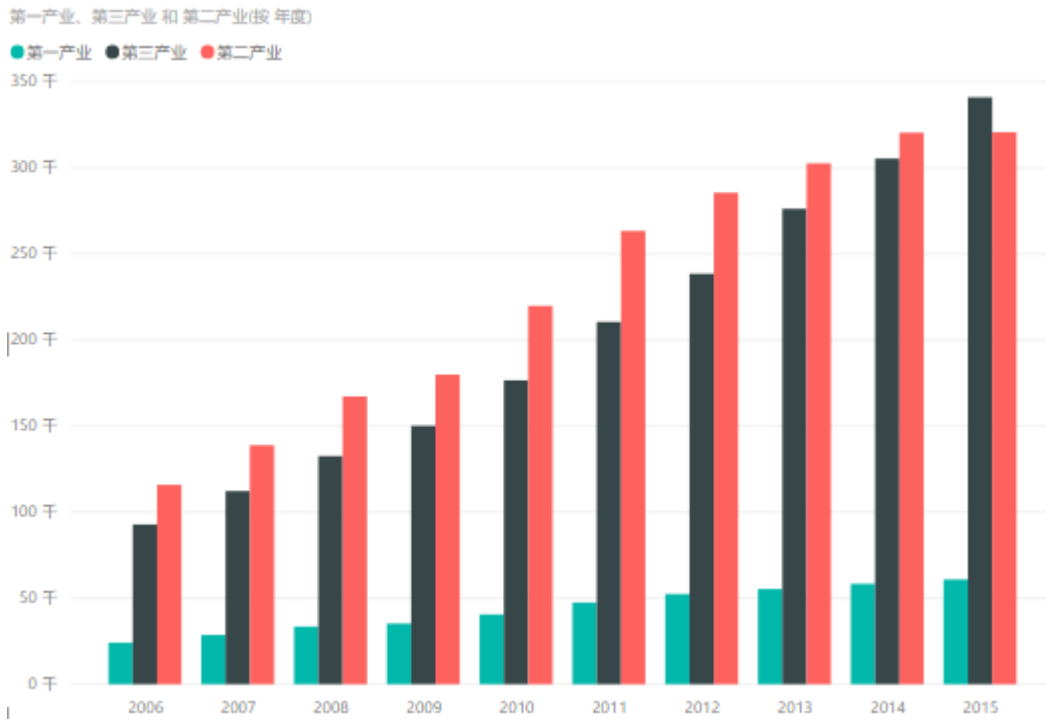
三个序列加入以后就变成了堆积柱形图，显示如下，



堆积柱形图的特征：

- 不同的序列在一根柱子上显示
- 可以直接比较总量的大小
- 分类序列的数值比较功能弱化

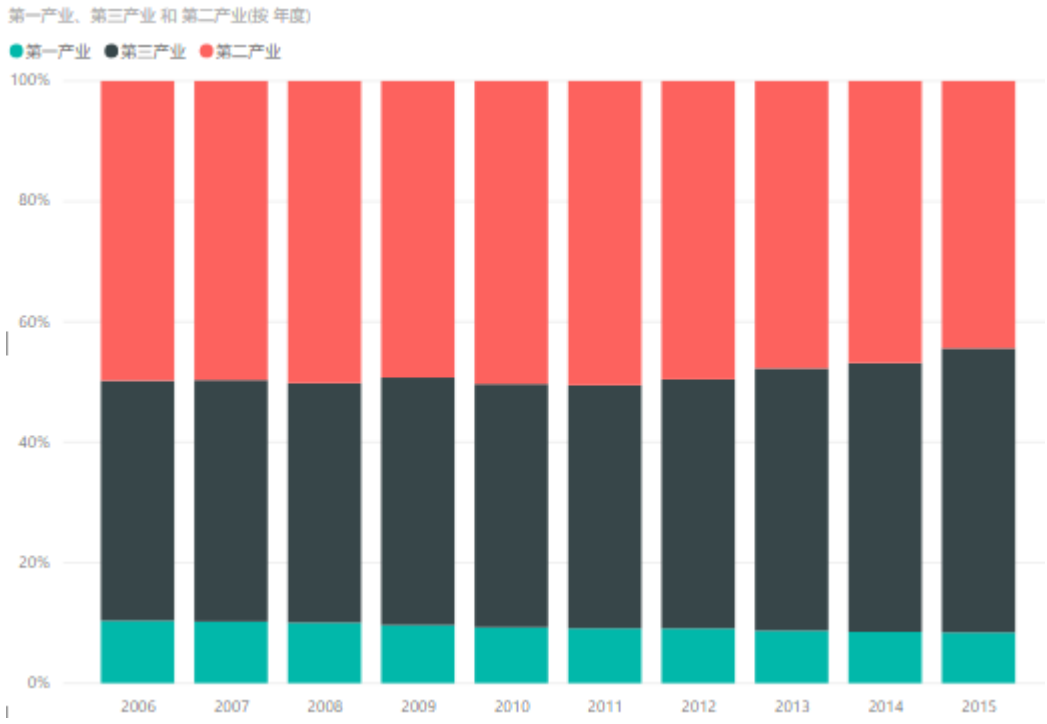
选中刚才制作的堆积柱形图，在可视化区域点击簇状柱形图，刚才的图表就变成了簇状柱形图，



簇状柱形图的特征：

- 不同序列使用不同的柱子
- 可以比较各序列的数值大小
- 总量比较功能弱化

再次选中该图表，然后点击百分比堆积柱形图，图表就变成这样了，



百分比堆积柱形图的特征：

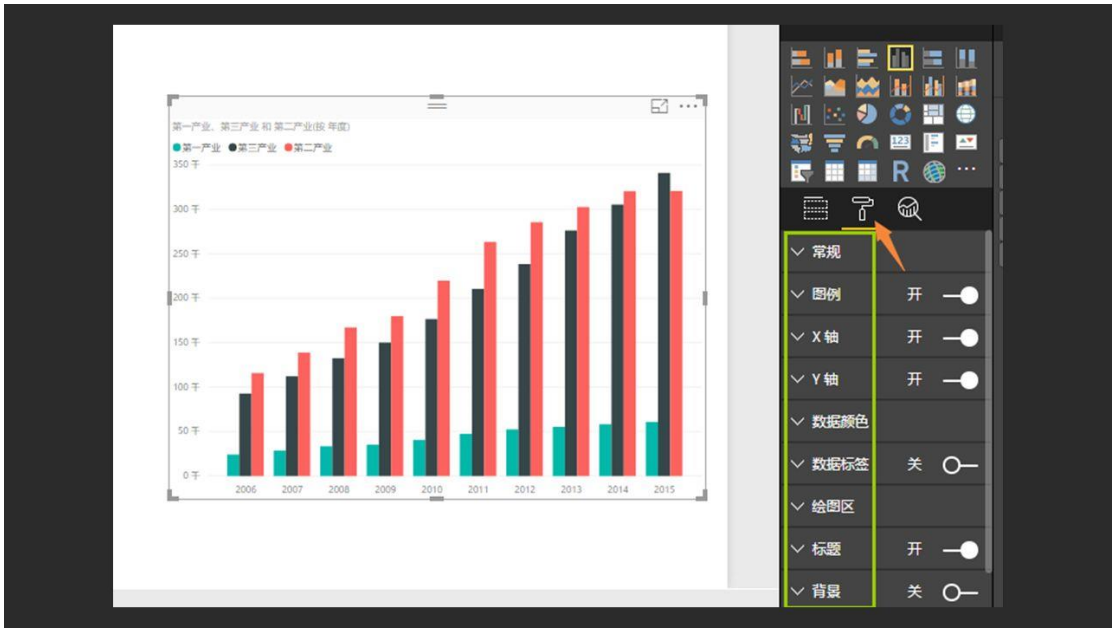
- 与堆积柱形图类似，不同序列在一根柱子上显示
- 显示各序列的相对大小，Y 轴标签变为百分比
- 无法比较总量，每根柱子都一样高

以后就可以分清楚各类柱形图了吧。

柱形图格式设置

从上面 PowerBI 默认出来的图表效果可以看到，虽然比 Excel 中默认做出来的图表好看了

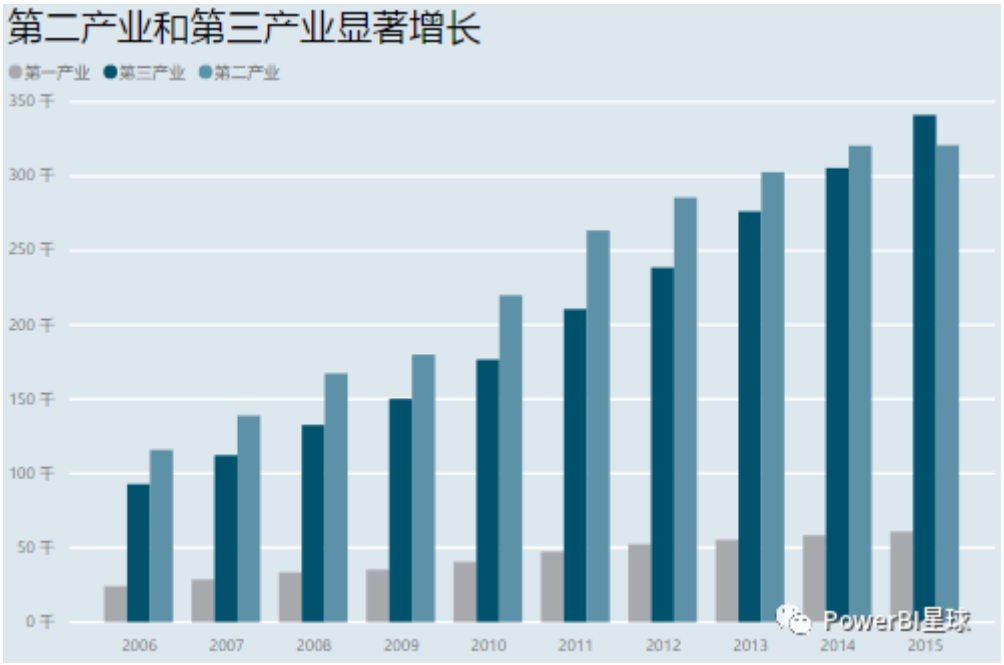
一点点，但离美还差了一大截，完全达不到我们"卖弄"的需求，好在 PowerBI 中进行格式的美化非常简单，下面就以上面的堆积柱形图为例，选中该图表，点击格式，



就可以显示每一项图表要素的设置，有开关按钮的要素可以点击打开，然后点击向下的箭头，就可以显示每项要素的细节设置，比如对刚才的簇状柱形图进行如下设置：

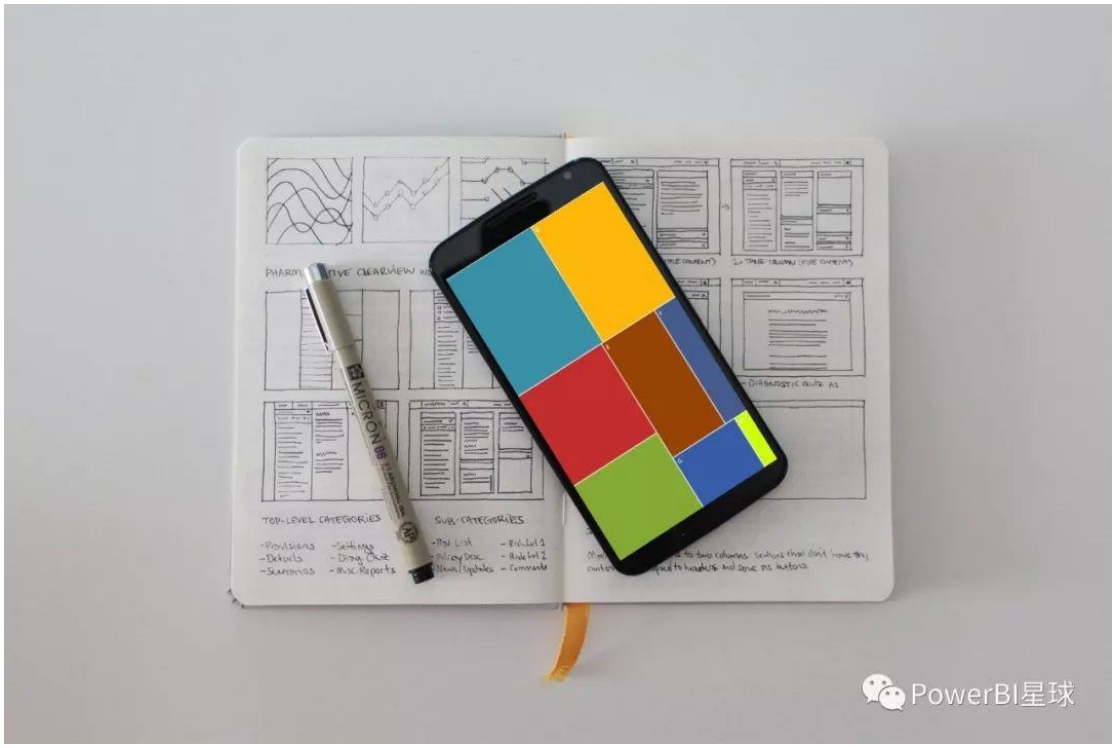
- 修改标题文字、并增大字号
- Y 轴网格线变为白色，宽度设为 2 磅
- 三个序列数据颜色更换
- 背景颜色更换

变成了下面的风格，



就这样简单点击几下，典型的 ECO 图表风格就大致出来了，一个合格的柱形图也大抵如此：纵横轴清晰、有图例、有数据单位、和图表呼应的标题，当然还有最养眼的色彩搭配，关于图表的美化后面会继续深入。

树状图

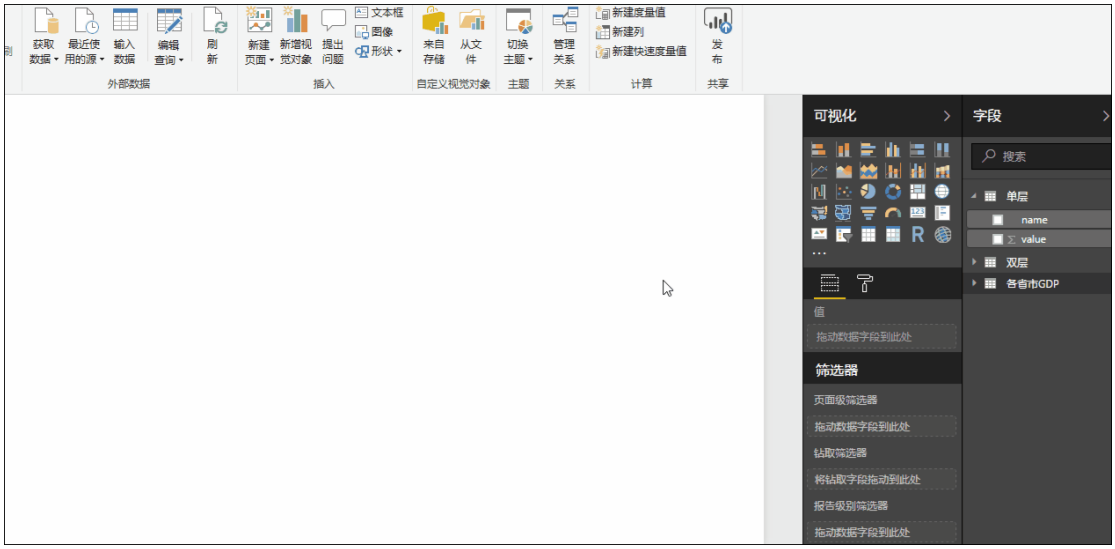


树状图，顾名思义应该就是像树的形状的图形，但今天所讲的图表一点都不像树，严格来说，应该称为矩形树图。

不过既然可以称为树图，当然还是和树有关系的，它把整体的数据想象成一颗树，而其中每一个数据就是一个枝叶，不过这个枝叶是放在一个矩形中，然后每一个数据矩形，错落有致的排放在一个整体的大矩形中。

单层树状图

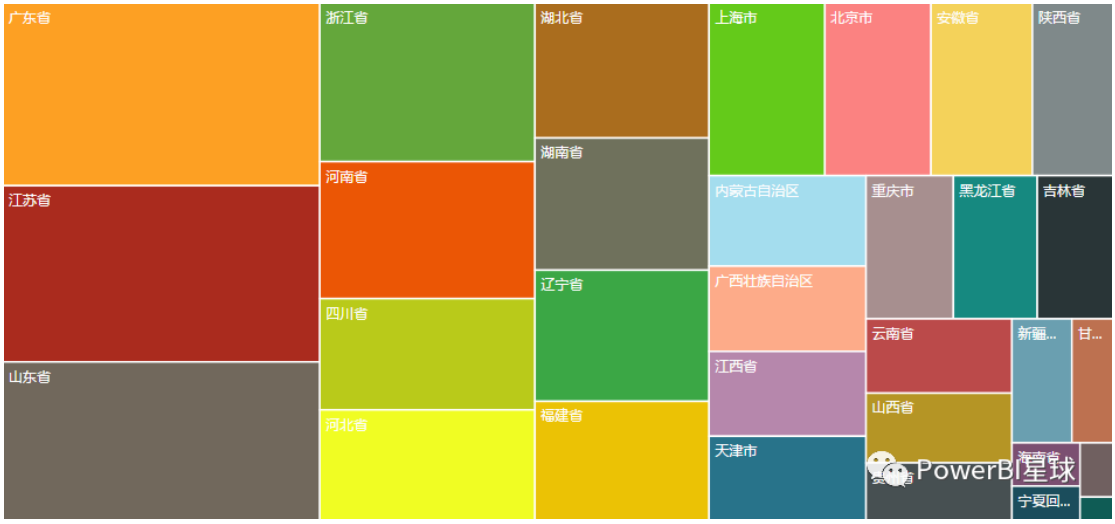
先看看 PowerBI 中树状图的生成过程，



无论如何拖拽，整体上都保持一个矩形的形状,可以根据画布的空间大小和矩形排列美感选择一个合适的大矩形，

矩形树图通过每个矩形的大小、位置和颜色来区分各个数据之间的权重关系，以及占总体的比例，使你一目了然的看到整个数据集。

利用树状图来看看各省市的 GDP 排名，



树状图默认各个矩形从上到下、从左向右按面积降序排列的，因此可以很直观的看到每个省市的 GDP 在全国排在什么位置。

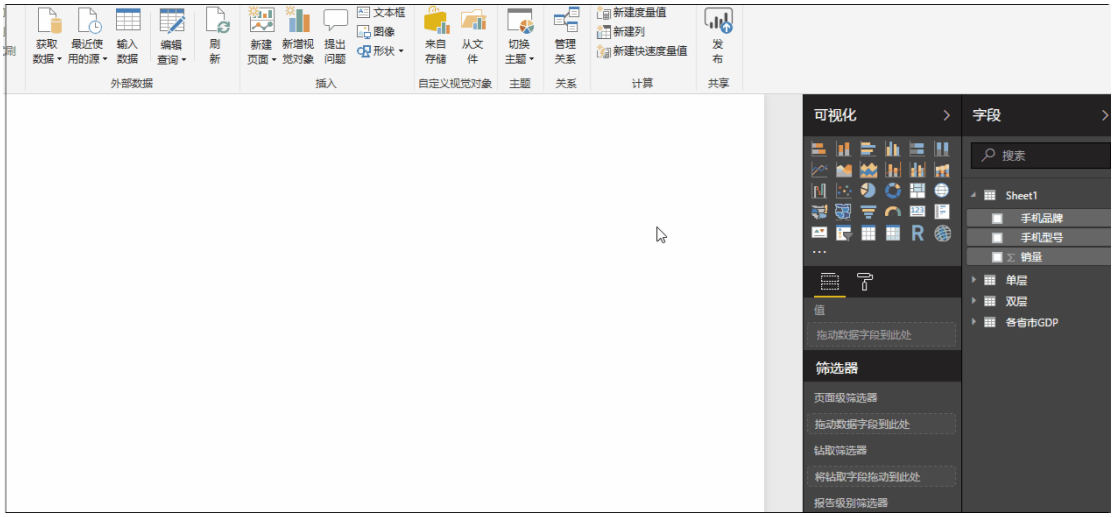
树状图不仅可以表示上面单层数据关系，还可以用来展现双层结构。

双层树状图

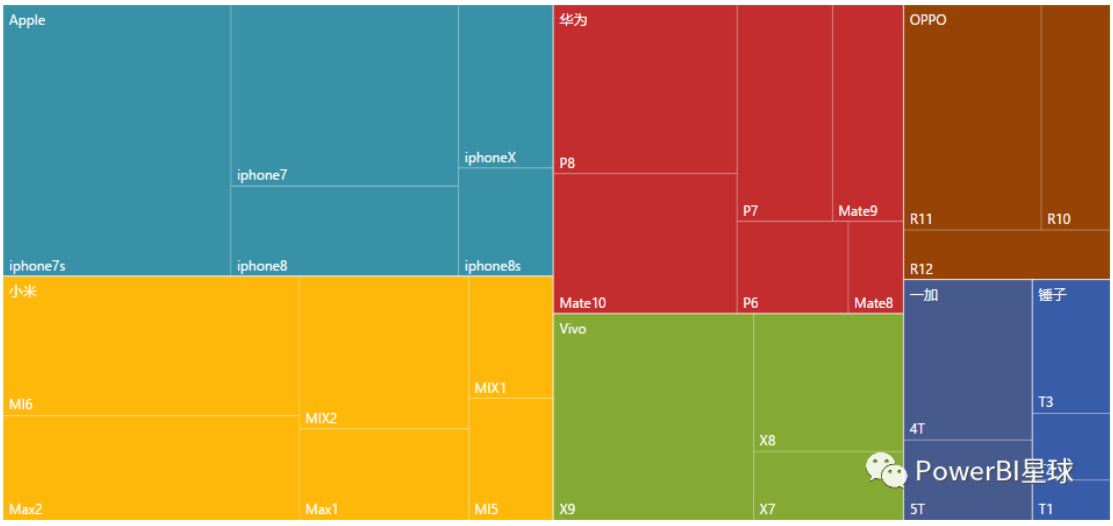
比如模拟一份各品牌手机的销量数据，

手机品牌	手机型号	销量
华为	P6	100
华为	P7	200
华为	P8	300
华为	Mate8	50
华为	Mate9	150
华为	Mate10	250
小米	MI5	100
小米	MI6	400

依然在树状图中进行可视化，



在单层树状图的基础上添加一个明细字段到【详细信息】中，就成了双层树状图，



每一种颜色的矩形代表一个手机品牌，从这个矩形的大小可以判断各个品牌的整体销量；而在各个品牌矩形内部，又根据各种型号的销量分成一系列小矩形，很直观的表现了两个层级的数据结构。

树状图的使用场景

- 要显示大量的分层数据
- 条形图不能有效地处理大量值
- 要显示每个部分与整体之间的比例
- 要显示层次结构中指标在各个类别层次的分布的模式
- 要使用大小和颜色编码显示属性
- 要发现模式、离群值、最重要因素和异常

平时看到其他图表如条形图、折线图等，在图表内部总有空白区域没有被完全利用上，而在树状图中，没有任何空白区域，每一处都用在了表现数据的关系上，从这方面看，树状图绝对算是空间利用率最高的图表了。

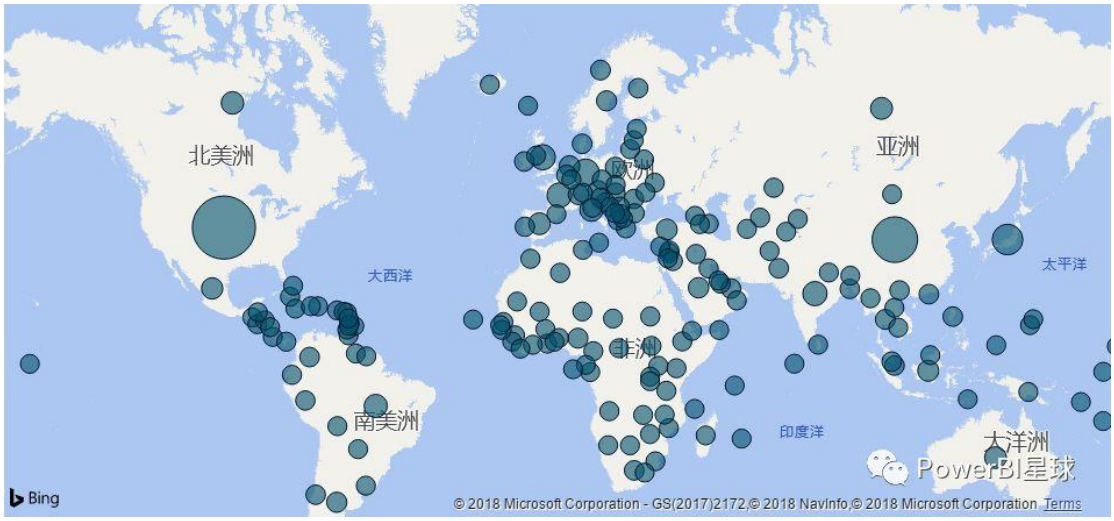
地图可视化

利用内置微软自己的必应地图，在 PowerBI 中可以轻松生成地图来实现各种数据可视化。

在 PowerBI 的默认可视化对象中，有三种地图对象：气泡地图、着色地图和 **ArcGIS Map**。

气泡地图-Bubble Map

在地图上利用气泡的大小来表示不同地区的数据，比如展现世界各国的 GDP，

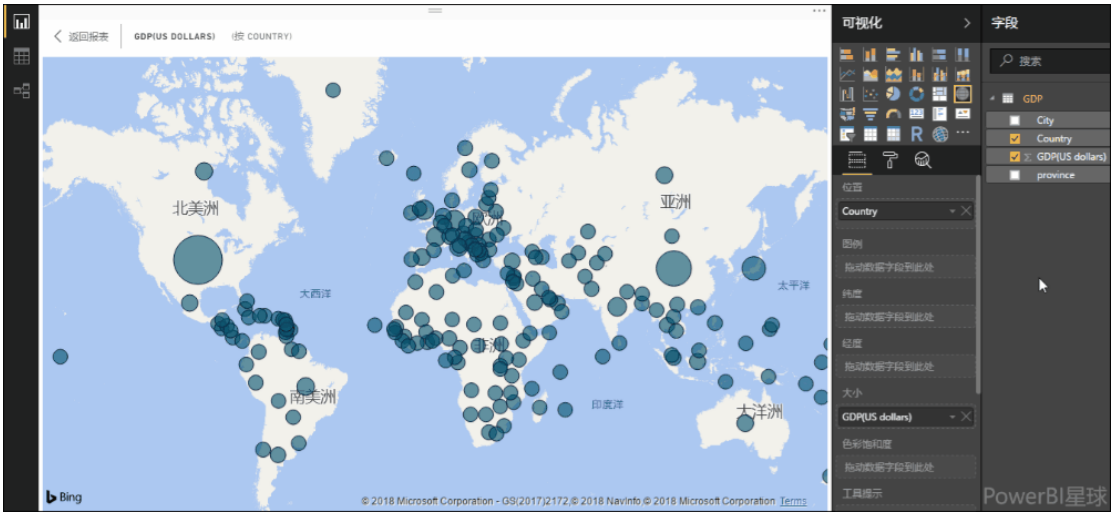


可以明显看到世界地图上美国的气泡最大，表示 GDP 最高，而中国和日本的次之。气泡的颜色也可以自定义。

制作步骤如下，

如果有多个层次的位置列，比如上图中是国家层级的位置，再加上省自治区、市级的位置数据，就可以在地图上向下钻取。

以中国数据为例，在中国的层级下添加各个省份的 GDP 数据列，并且在河南省下添加各个地市的 GDP 数据列，然后把这两个字段列拖到位置框中，单击图表右上角的向下箭头，启用"深化"，就可以在地图中向下钻取到下一层级的数据了，GIF 动画如下：



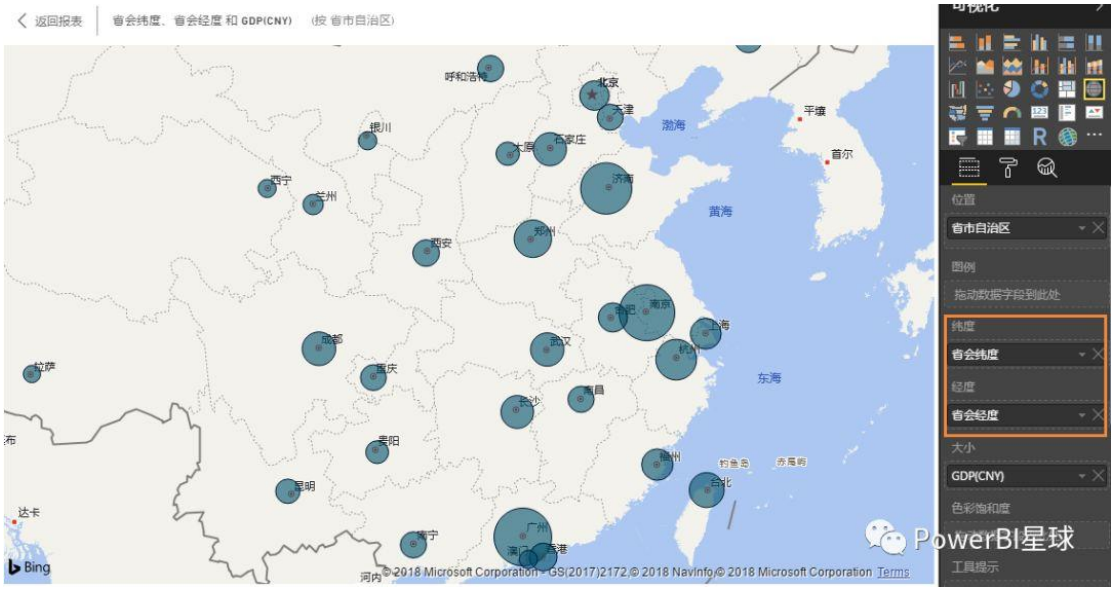
必应地图对于地名的识别有时会有偏差，尤其是对于非英语名称的地理位置，中国地名是否为全称也会有影响，建议都写全称，比如"北京"应该写为"北京市"。

在刚才的中国地图中，台湾和澳门的数据都没有显示出来，如果要保证位置的精确性，把位置的具体经纬度添加进来是必要的。

国家	省市自治区	省会	省会经度	省会纬度	GDP(CNY)
China	北京市	北京	116.47	39.9	24899.26
China	上海市	上海	121.48	31.23	27466.15
China	天津市	天津	117.18	39.15	17885.39
China	重庆市	重庆	106.53	29.53	17558.76
China	黑龙江省	哈尔滨	126.68	45.75	15386.09
China	吉林省	长春	125.32	43.87	14886.23
China	辽宁省	沈阳	123.4	41.83	22037.88
China	内蒙古自治区	呼和浩特	111.8	40.82	18632.57
China	河北省	石家庄	114.47	38.03	31827.86
China	山西省	太原	112.57	37.87	12928.34
China	山东省	济南	117	36.63	67008.19
China	河南省	郑州	113.7	34.8	40160.01
China	陕西省	西安	108.9	34.27	19165.39

PowerBI星球

把省会的经度和纬度分别拖入到【经度】和【纬度】框中，

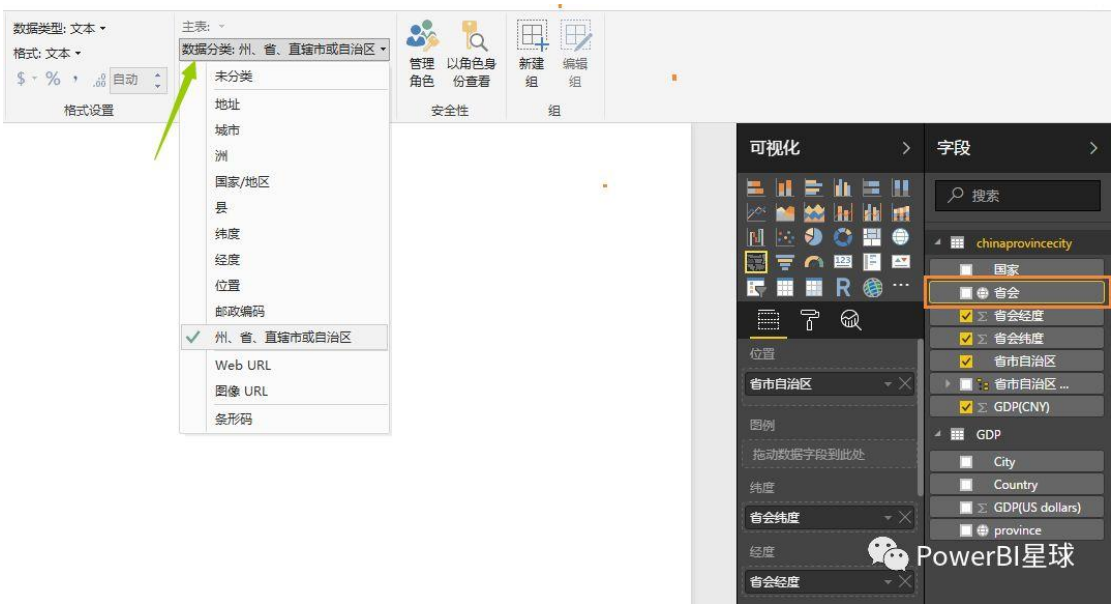


这样台湾和澳门的气泡图数据都展现出来了。

着色地图-Filled Map

着色地图和上面的气泡图制作过程基本一样，只是展现数据的方式不是使用气泡，而是用颜色填充的方式，颜色越深表示数值越大，

着色地图对位置信息的要求比上面的气泡地图要更严格，提供经纬度数据会更容易正常显示，另外还需要把相应字段先设置地理信息分类，并选择和地理位置对应的层级，



设置好以后，该字段前面会显示地球状的地理标识。

以后无论使用何种地图可视化，推荐先按上述方式设置好地理信息的字段。

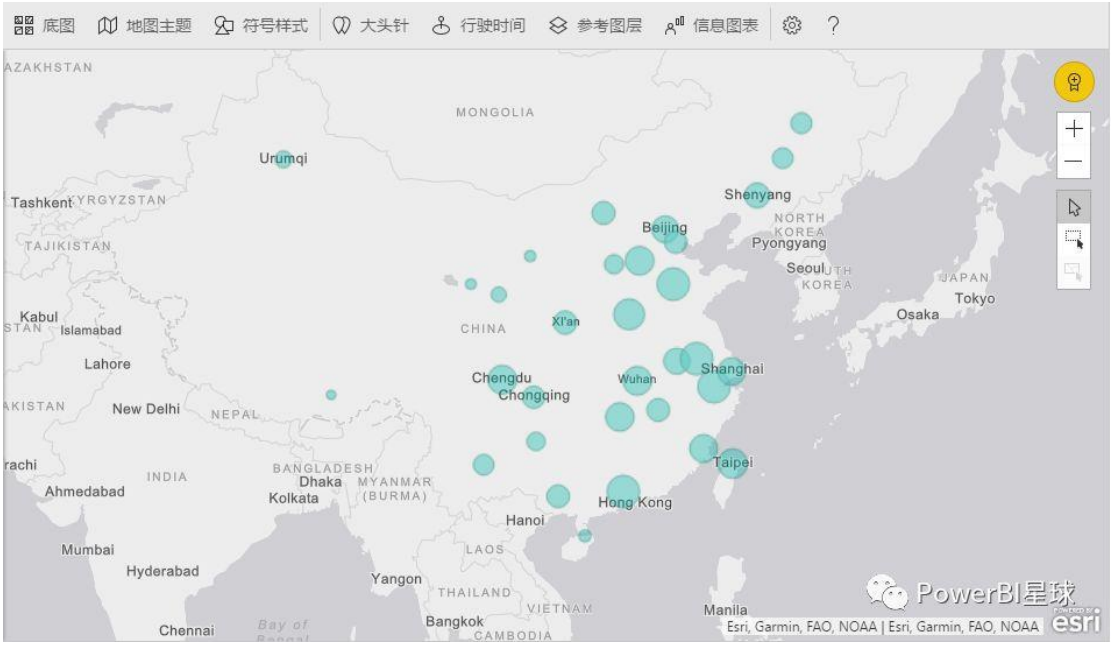
着色地图在 PowerBIDesktop 中总是无法正常显示，可以把数据导入到 PowerBI 在线服务，然后使用着色地图可视化工具可以正常展现出来。

上面的地图还可以设置背景的主题样式，现在可用的样式有以下几种，

ArcGIS Map

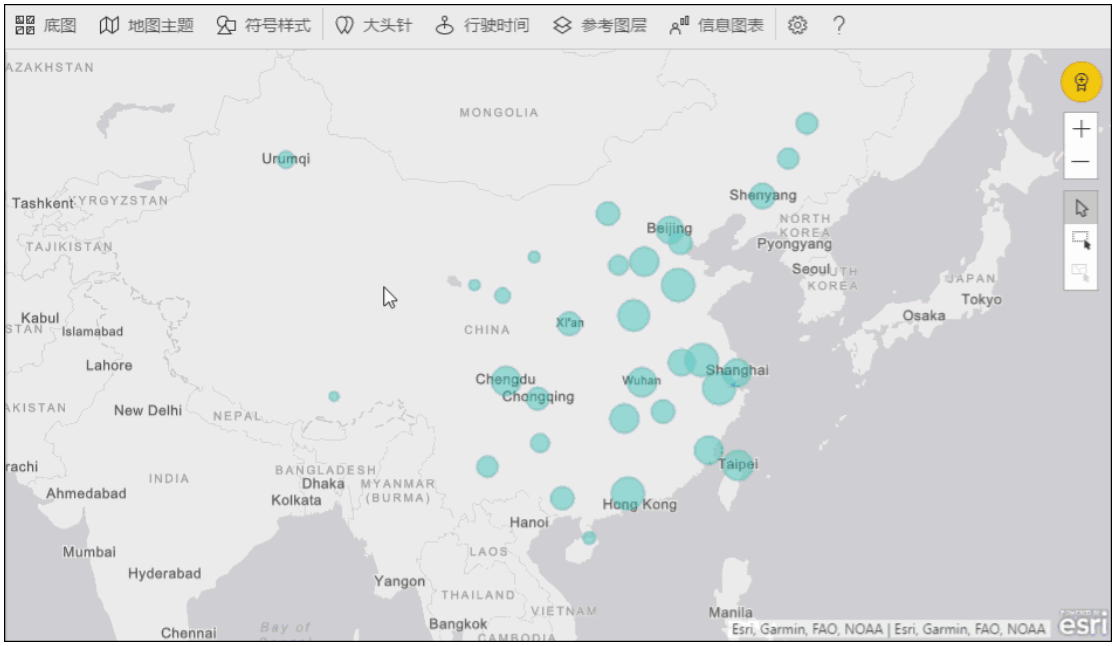
PowerBI 内置的 ArcGISMap，它不是由必应地图提供，而是由 ESRI 提供的，所以第一次使用时会弹出 ESRI 使用条款对话框，点击同意即可。

依然使用上面的数据，使用 ArcGIS Map 可视化，看起来和气泡地图没有什么区别，点击右上角的三个点点，选择编辑，各种强大的功能就在最顶端展示出来了。



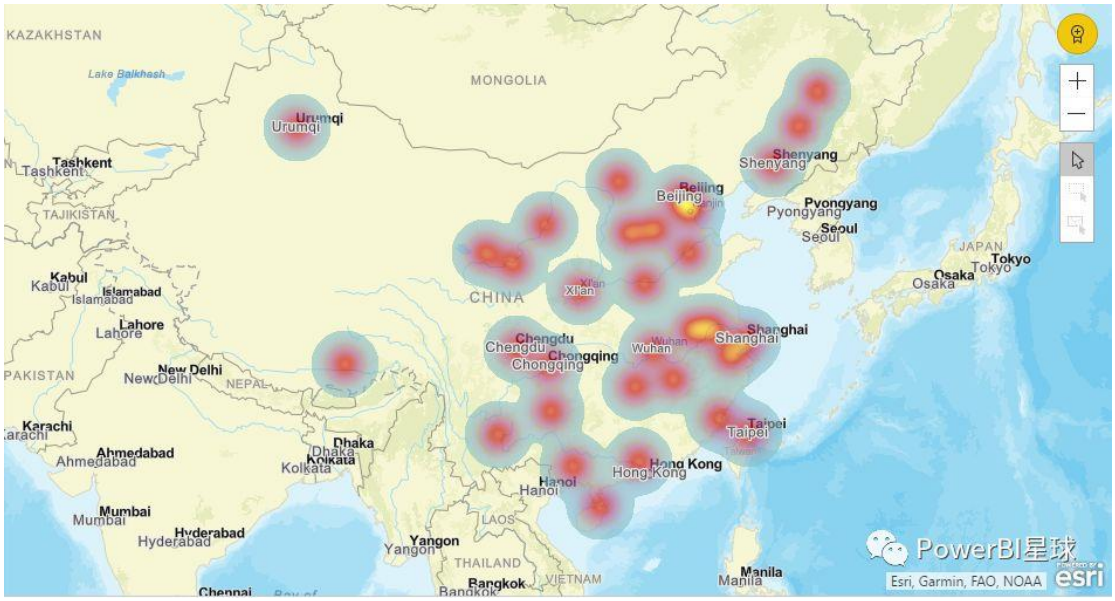
底图

就是地图的背景画布样式，提供了以下 4 种画布，



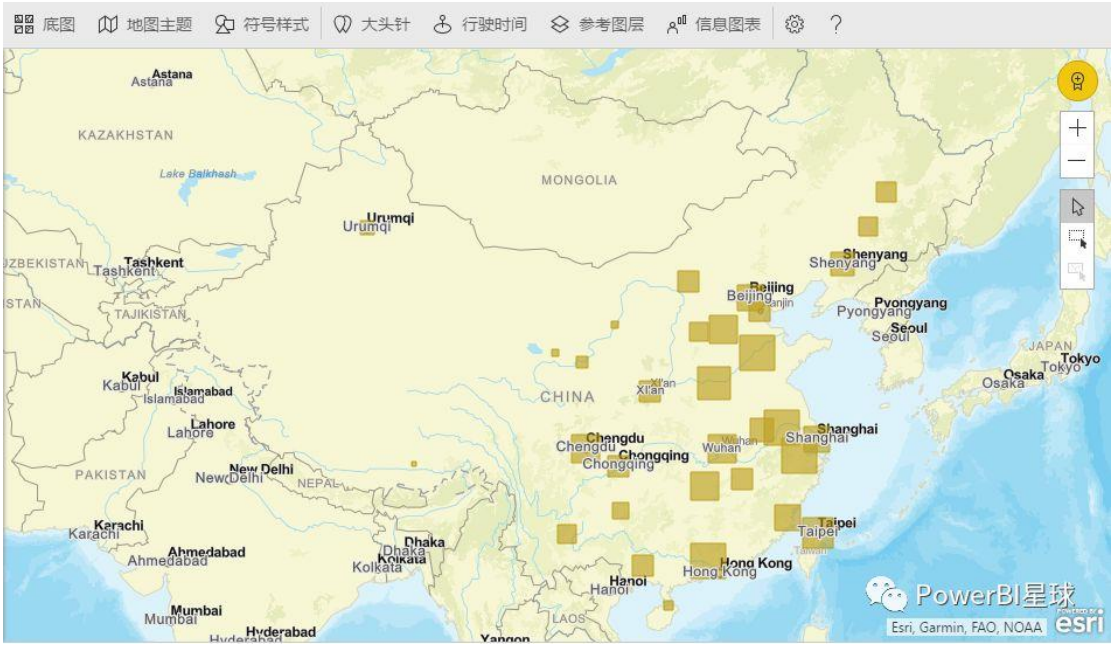
地图主题

提供了四个地图主题。位置主题就是只显示位置，气泡大小一样；而大小主题就是根据数值的不同显示大小不等的气泡，看看之前没有见过的热图主题，



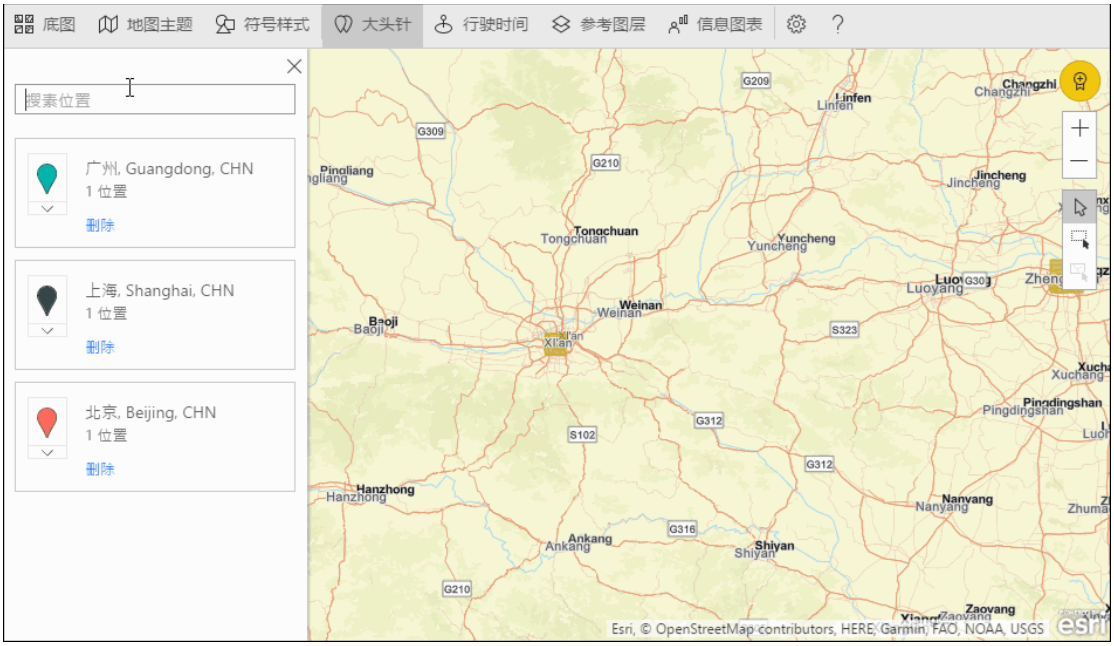
符号样式

使用符号样式可以对地图上数据的显示方式进行精细调整。比如不想用圆形的气泡，改称方形的，



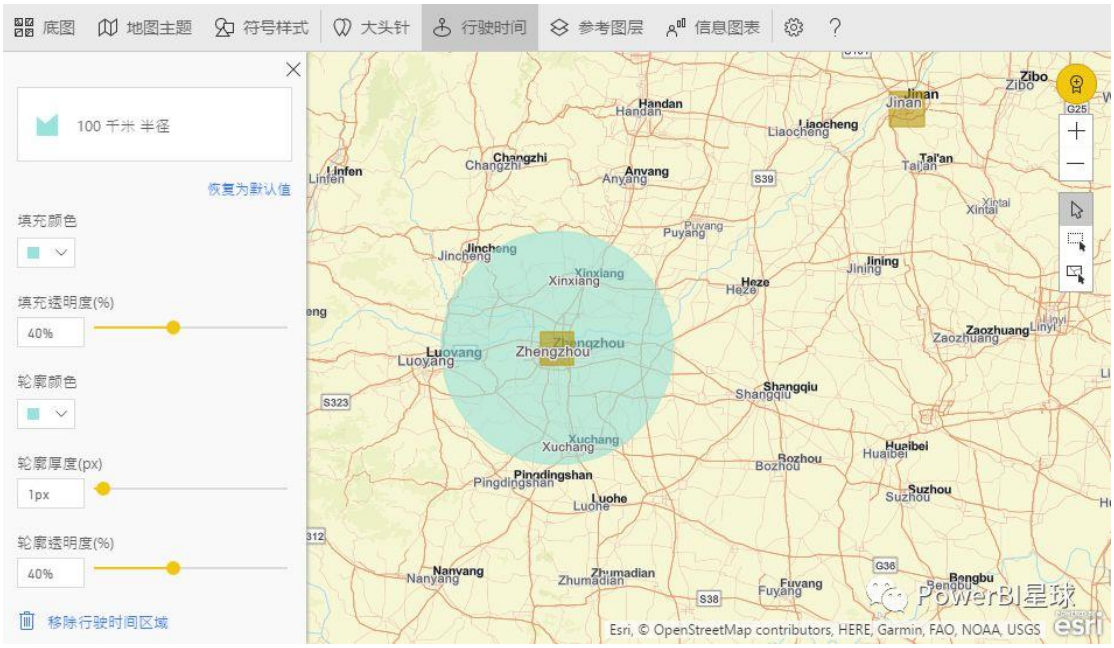
大头针

在搜索框中键入关键字（如地址、地点和兴趣点），再从下拉列表中进行选择。 地图上会显示一个符号，点击会快速定位并会自动缩放到相应位置。



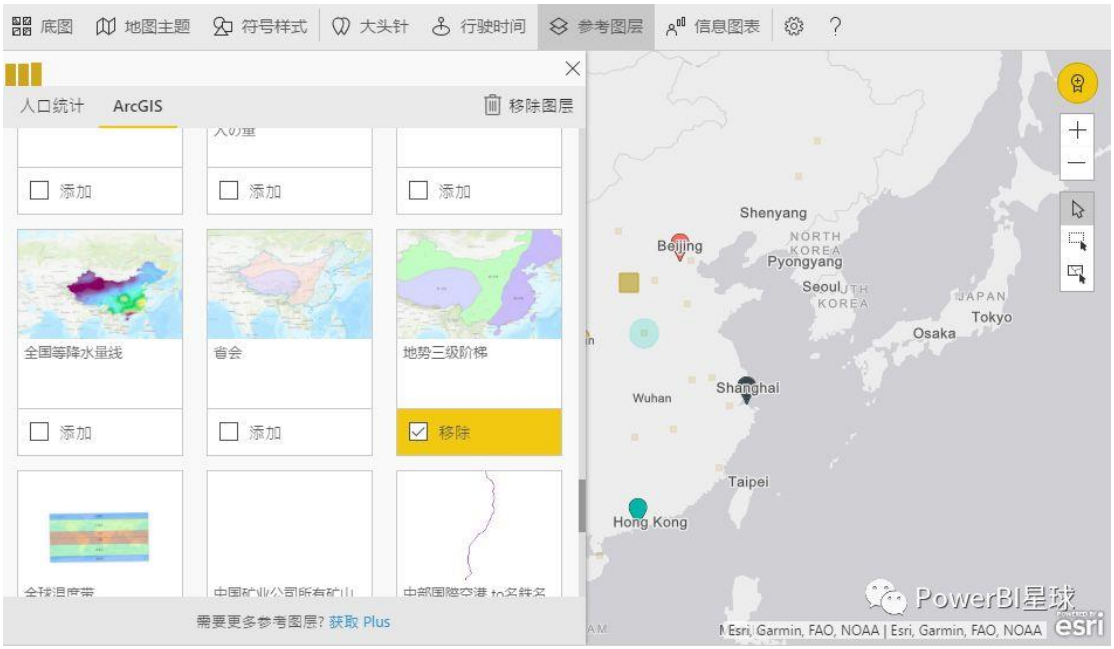
行驶时间

可以选择一个位置，然后确定在指定半径区域或驾驶时间内可用的其他地图功能。

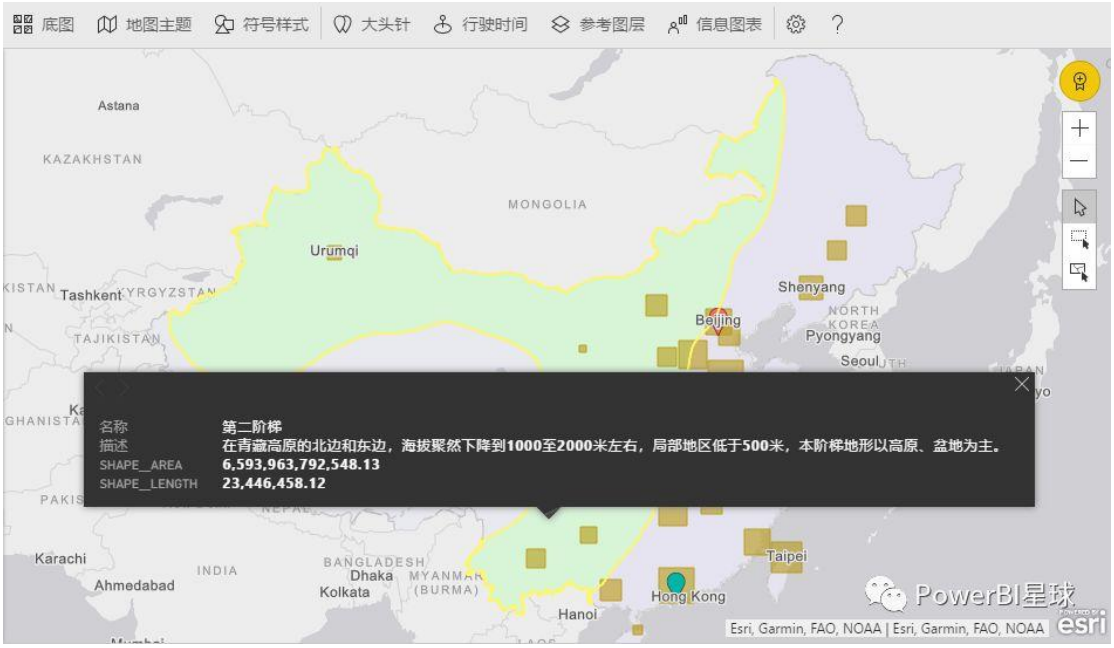


参考图层

可以在地图上添加各种统计信息层，比如在中国地图上添加地势三级阶梯，



在图层上还可以交互，点击展示出相关信息，



还有其他设置，大家可以尝试动手摸索。

总之，ArcGIS 地图超越地图点表示法的地图绘制技术提升到新的水平。 从基本地图、位置类型、主题、符号样式和引用层中进行选择，创建丰富多彩的信息性地图可视化效果。

以上三种就是 PowerBI 默认的地图可视化对象，基本可以满足日常需要，如果想用更个性的地图工具，还有不少自定义地图可以使用，以后再接着介绍。

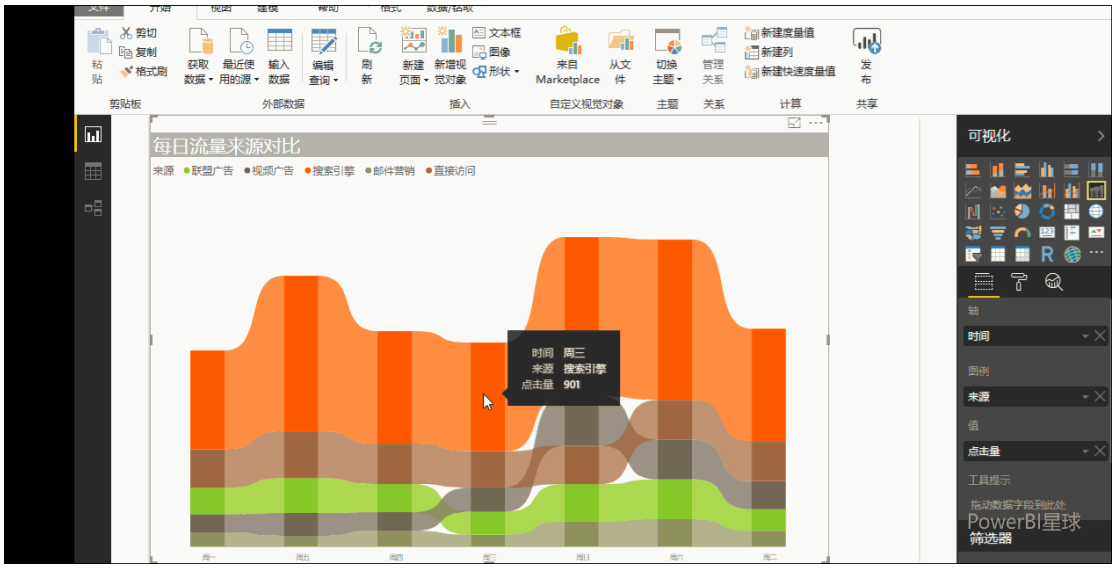
本文生成地图的数据可以在公众号中回复“**地图练习数据**”下载。

【第 6 天 | PowerBI 常用操作】

查看图表背后数据的三种方式

图表很直观，但有时候我们不仅想看图，也想更进一步查看生成该图表的明细数据，在 PowerBI 中有三种方式。

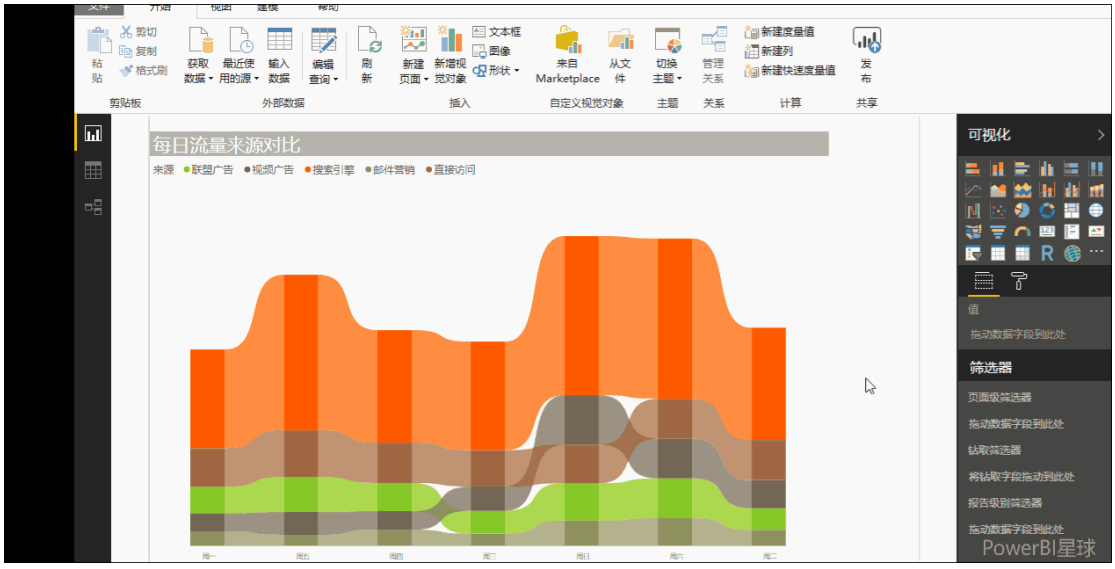
(一)在图表上单击右键，选择“查看数据”



数据可以显示在图表右侧，也可以放在图表下边，数据显示出来后，点击右上角的三个点，还可以导出数据。

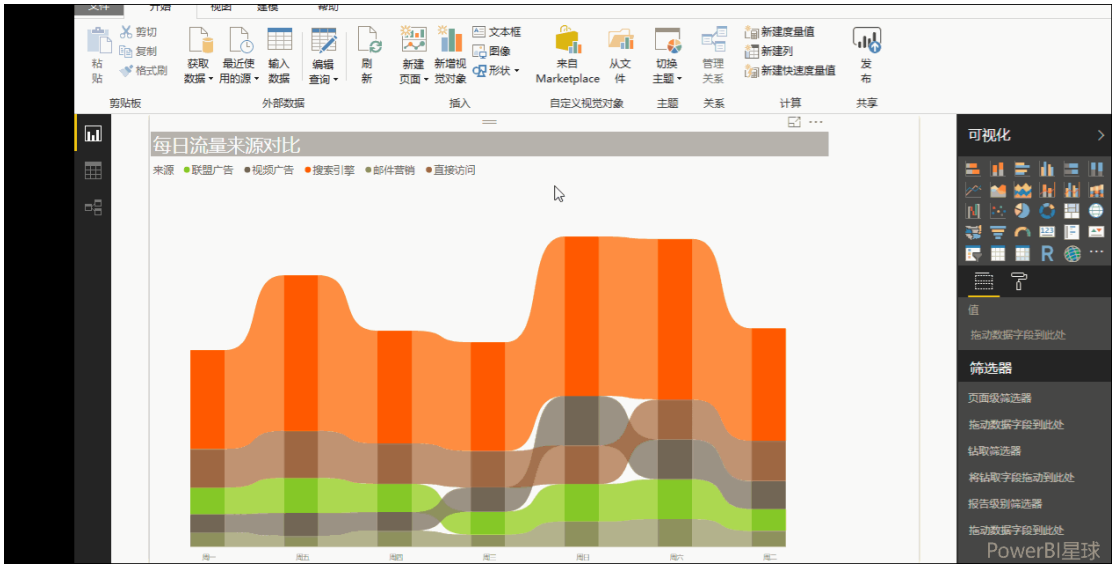
若要隐藏数据，返回之前状态，点击“返回报表”即可。

(二)选中图表，然后点击右上角三个点，选择“显示数据”



这种方式，还可以直接导出数据，在“查看数据”上边的功能就是“导出数据”。

(三)选中图表，点击上方【数据/钻取】选项卡“查看数据”



以上三种查看数据的方式，都可以导出数据，平时我们是用自己的数据做的图表，可能觉得导出数据没有什么意义，但有时别人直接发过来一个 pbix 的文件，如果想从这个文件中获取基础数据，导出数据的功能就派上用场了。

利用这种方式导出数据有行数的限制，最多 3 万行，如果基础数据超出 3 万行，这种方式就不行了，当然也有其他解决的办法，以后再介绍。

图表的钻取

（**PDF** 不显示动图，本文推荐在公众号中阅读）

查看可视化图表的时候，我们可能想深入了解某个视觉对象的更详细信息，或者进行更细粒度的分析，比如看到 2017 年的总体数据，同时想知道每个季度甚至每个月的数据，通过 PowerBI 的钻取功能，可以点击鼠标轻松实现。

当图表中的数据存在层级结构时，可以在图表上直接下钻展示下一层级的数据，最常见的层级结构就是日期数据，从年度、季度、月份到日期，甚至到小时、分钟和秒，只要具体的日期数据的层次结构足够详细。

下面就来看看什么是钻取功能，以及如何进行钻取。

为了能够直观看到层次的变化，先用一个矩阵表来展现，把日期表中的日期拖入到【行】框中，Power BI 会自动添加时间层次结构，

如果不想要某个层次，点击右边的叉删除即可。把销售额也拖入到值中，显示的是年度数据

年	销售额
2016	221549000
2017	246070100
总计	**467619100**

PowerBI星球

钻取到下级层级的数据有两种方式：

（一）通过图表右上角的向下箭头“启用深化”。

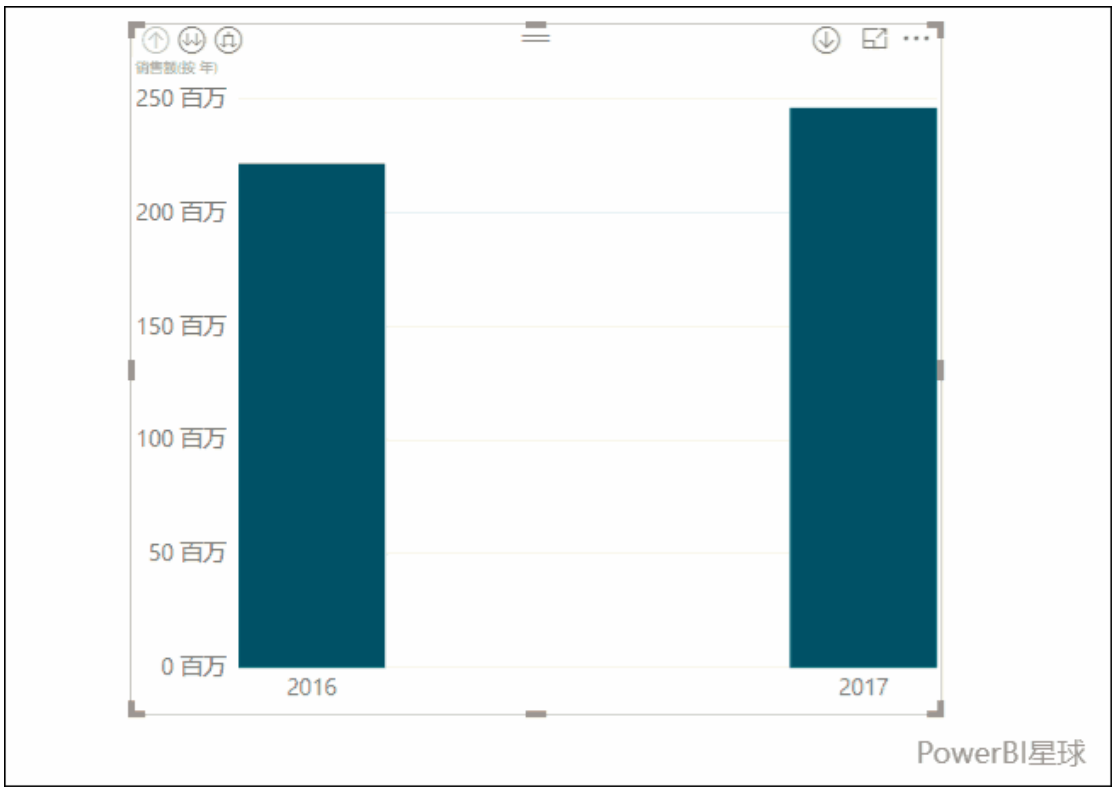
启用“深化”以后，直接点击需要钻取的数据对象就可以了，操作过程和效果如下：

点击左上角的向上箭头，可以向上一级汇总。

另外左上角还有两个按钮，点击第二个按钮，显示下一层级的数据，

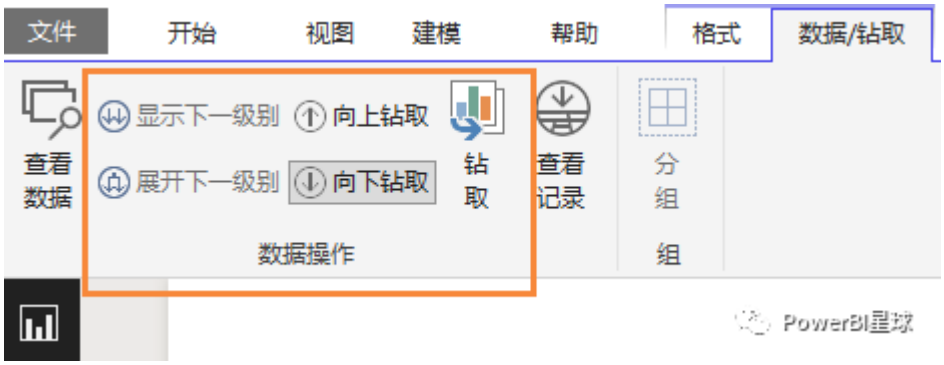
点击左上角第三个按钮，可以同时显示所有层级的数据，

按照以上的操作方式，在柱形图中也可以进行钻取，



（二）使用顶部 Power BI"数据/钻取"选项卡。

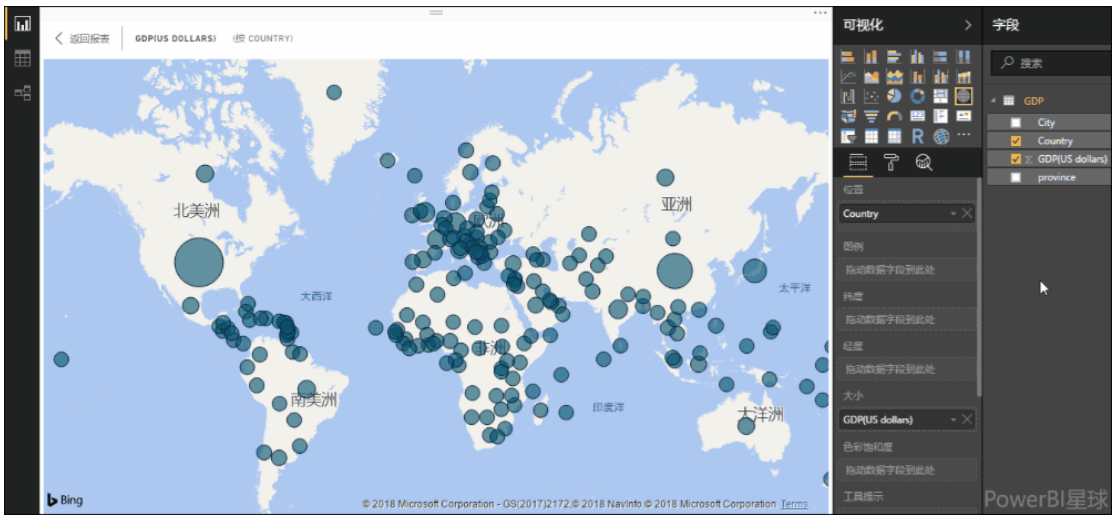
点击选中图表，上方按钮区出现"数据/钻取"功能，



通过点击这几个按钮，同样可以实现第一种方法的钻取效果，大家可以自己动手尝试。

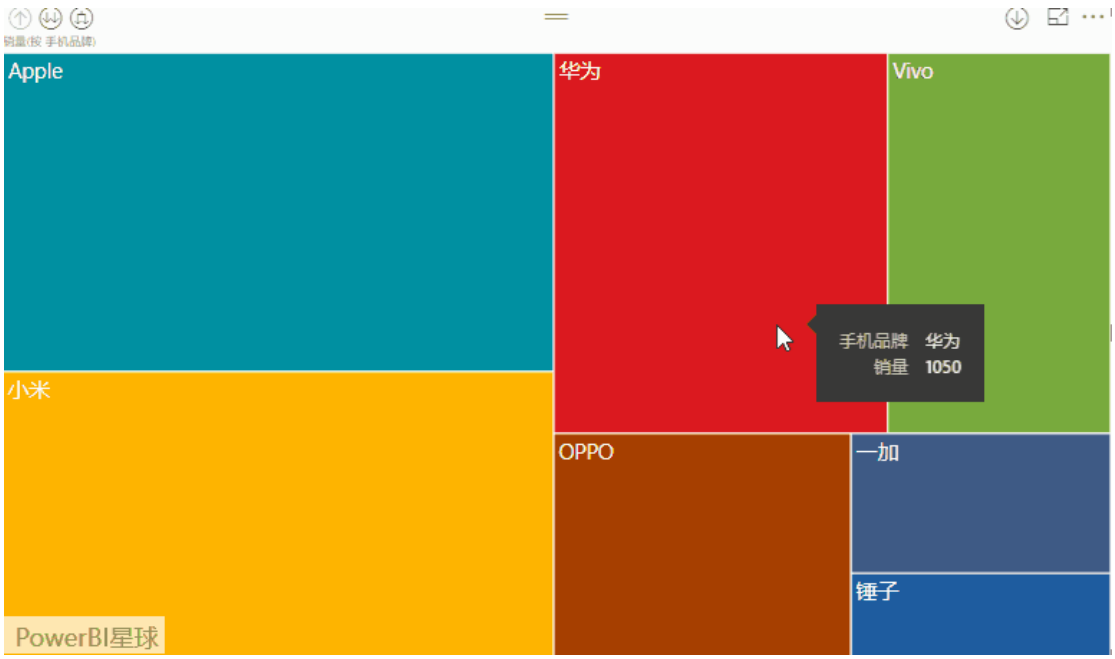
这两种方法都可以完成同样的钻取操作，平时可以选用你使用起来更顺手的那一种。

上面是日期层次的数据钻取，另外一种常见的分层级数据类型就是地理数据，从国家、省份到城市，可以一直往下细分，之前介绍地图可视化对象的时候，也展现过钻取功能，



请参考：地图可视化

还有前面介绍的树状图，也经常用来钻取数据，



请参考：树状图

其实只要是数据结构有层级关系，无论是什么类型，都可以进行钻取操作。

最后需要注意的是，如果向可视化效果添加日期字段没有自动创建层次结构，则可能是因为"日期"字段实际上并未保存为日期类型。 回到"数据"视图下打开该表，选择包含日期的列，然后在"建模"选项卡中将"数据类型"更改为"日期"或"日期/时间"就可以了。

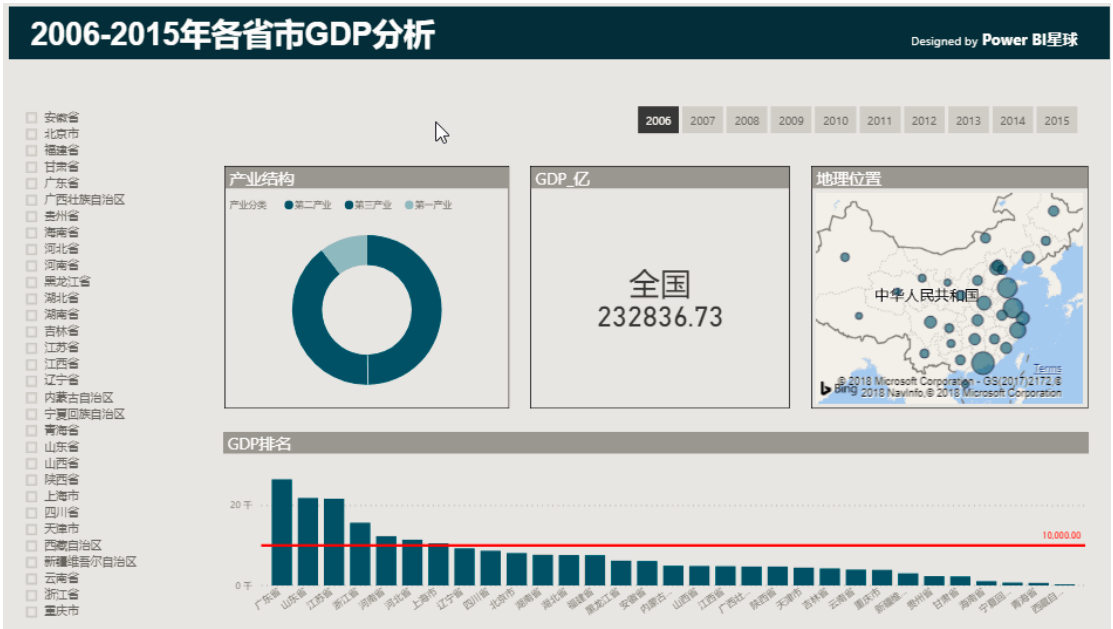
编辑交互

（**PDF** 不显示动图，本文推荐在公众号中阅读）

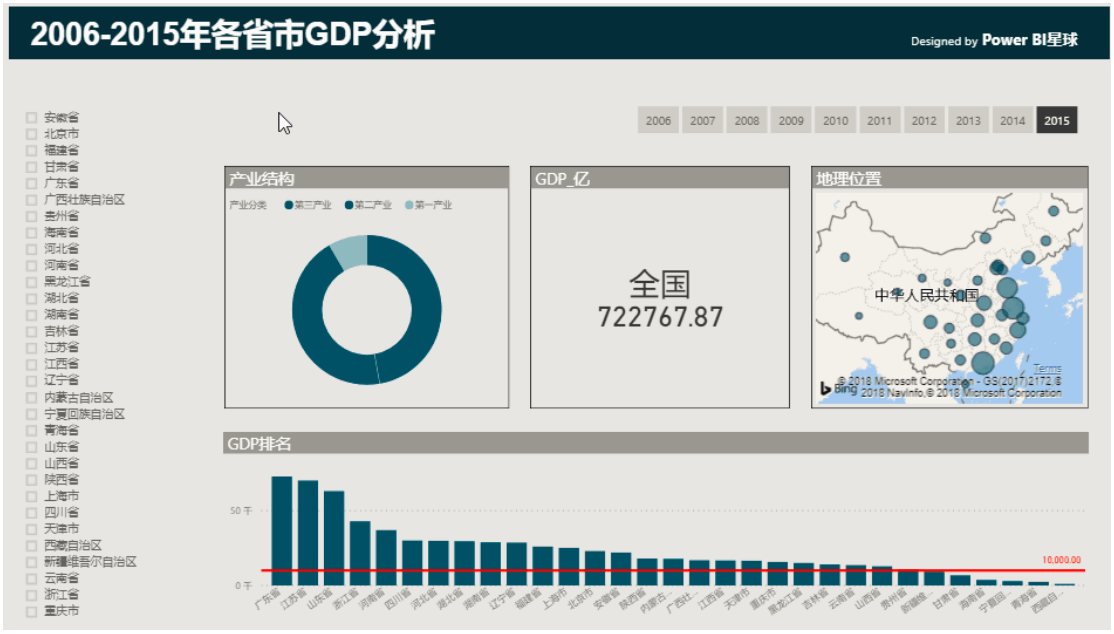
PowerBI 可视化与传统图表的一大区别，就是可视化分析是动态的，通过页面上筛选、钻取、突出显示等交互功能，可以快速进行访问、发现、探索数据背后的规律。但是，不必要的交互会将优异的分析变成无用分析，本文介绍如何恰到好处地使用交互式视图。

下面以这个报表为例，看看 PowerBI 可视化报表的交互效果

默认情况下，筛选条件应用于该报表上的所有视图，



不仅切片器可以筛选，每一个可视化对象，同时也可以是其他图表的筛选条件；点击任何一个筛选条件，其他图表动态响应，从不同的角度展现数据，但如果筛选使用不当，展现出来的可视化也会令人困惑，比如下面这个，



下方的柱形图展示的所有省份的 GDP 排名，可是一旦利用左方的省份切片器进行筛选，柱形图只展示了被选中的省市数据，其他数据被筛选掉了，显然与我们的初衷不符，那如何进行设置呢？

选中任何一个可视化对象，在 Desktop 中，点击“格式”>“交互”，选择“编辑交互”，

可以看到每一个可视化对象上方都出现了两个图标：

筛选

不起作用

针对现在选中的筛选条件，对每个可视化对象进行设置：

- 如果需要响应该筛选，则选择“筛选”图标
- 如果不希望被筛选，则选择“不起作用”图标

对柱形图设置为对省市切片器“不起作用”

这样无论省份怎么选择，均不影响柱形图了。

如果选中圆环图或者柱形图等，这些图表右上角还有出现另外一个图标：

突出显示

下面看看突出显示的效果，

可以看出柱形图被选中的数据突出显示出来，突出显示与上面的筛选功能相比，最大的优点就是不仅被筛选出来，还可以保留其余数据点的上下文。

交互的动态效果有助于进行数据探索，但是要记住，每一个可视化对象只有在必要的时候才使用交互式视图，暂时抛开其他因素，交互性的优劣也决定了该可视化报表是差劲的还是优异的。

【第 7 天 | 初学者经常遇到的几个问题】

刚开始学习 PowerBI 时，不可避免的会遇到各种各样的错误，入门的最后一天，就介绍一下几个常见的错误以及解决的办法，帮你从一开始就避免这类错误；如果你在以后使用 PowerBI 的过程中遇到了类似错误，也可以回头来看看是不是可以解决你的困惑。

01 | DAX 书写错误

刚开始在 DAX 编辑框建度量值时，总是眼前一堆红色的波浪线有没有，

```
1  本周 =
2  VAR curyearweek=SELECTEDVALUE('日期表'[年度周数])
3  RETURN
4  CALCULATE(
5      [销售金额],
6      FILTER(
7          ALL('日期表',
8          '日期表'[年度周数]=curyearweek
9          &&'日期表'[日期]<=MAX('日期表'[日期])
10     )
11 )
```

PowerBI星球

经常有新人问我说，就是按照文章中的 DAX 一个个字符敲的，为什么会报错，其实出现这个问题一般都是少了右括号，忘记输入逗号或者有中文字符等，所以在编写 DAX 时一定要切记：

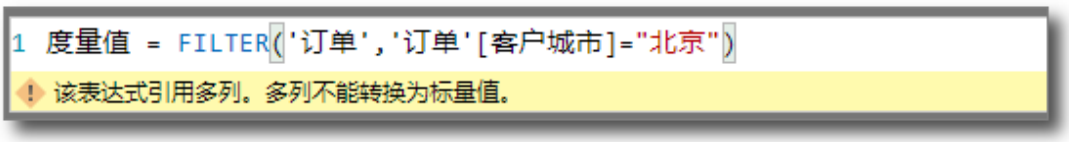
- 每一个函数左右括号 () ,要配对；
- 引用表的单引号 ''，不要漏；
- 参数之间的分隔逗号 ，不能缺少；
- 以上字符均要求为英文字符；

另外一定要按照格式规范书写 DAX（请参考：送你一份 DAX 格式指南），上图中出现错误，其实还是比较容易排查出错误在那里的，如果全部写在一行，有这些错误就更加头大了。

这个问题一般都是出现在刚开始学习时，写的多了基本都不会遇到或者出现这种提示可以马上知道是哪里错了。**所以学习 DAX 时不要只看不练，或者练习时复制别人写好的 DAX，一定要自己亲自动手去写，锻炼肌肉记忆，才能在需要编写 DAX 的时候行云流水。**

02 | 函数使用错误

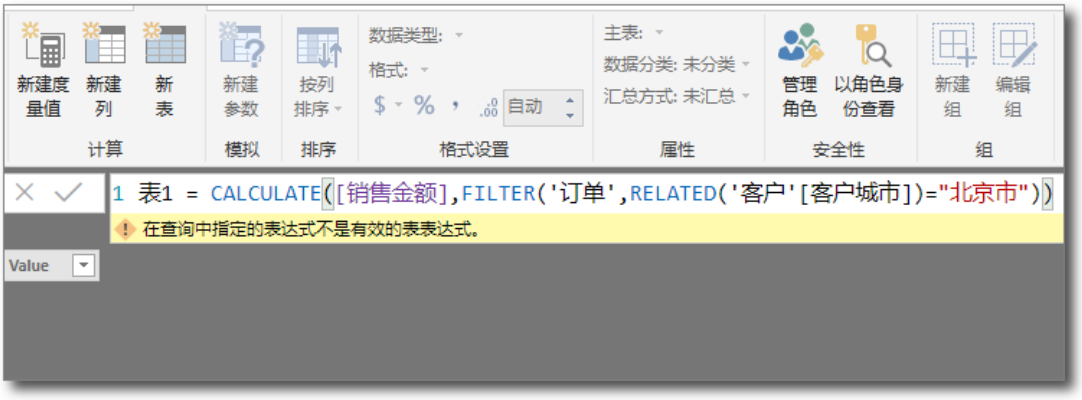
有些函数返回的表，若用于新建度量值，肯定返回错误，比如，



度量值需要返回的是一个值，而 FILTER 函数返回的表，所以它不能单独用于建度量值，但该函数可以作为其他函数的参数使用，来建立度量值。

虽然上图中的 FILTER 函数建度量值时出错，但是却可以建表，该表达式将返回所有北京的客户的订单表。

也有些函数返回的是值，若用于建表，也会报错，比如，



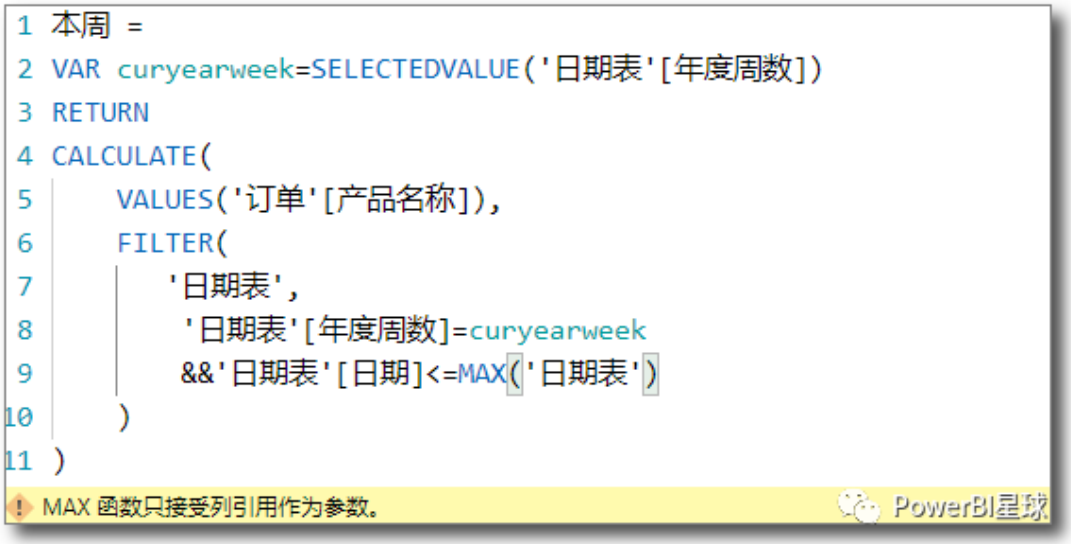
CALCULATE 函数返回的是一个值，不能用于建表。不过如果你确实想建一个只有一个值的表，也是可以做到的，可以在这个表达式外层套一个大括号{},正常返回一个表，

该表只有一个数据，为北京客户贡献的销售额。

通过这种方式，也可以用于在数据视图下，查看度量值的返回结果。

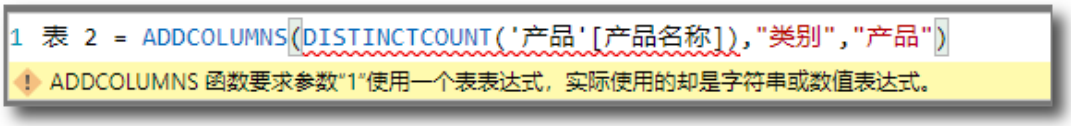
03 | 参数使用错误

参数错误的时候，有时并不会有红色波浪线提醒，比如下图，



提示 MAX 函数的参数只接受列，该错误就是参数应该为列的时候，使用了表。

还有的函数应该引用表，却用了列或者值，



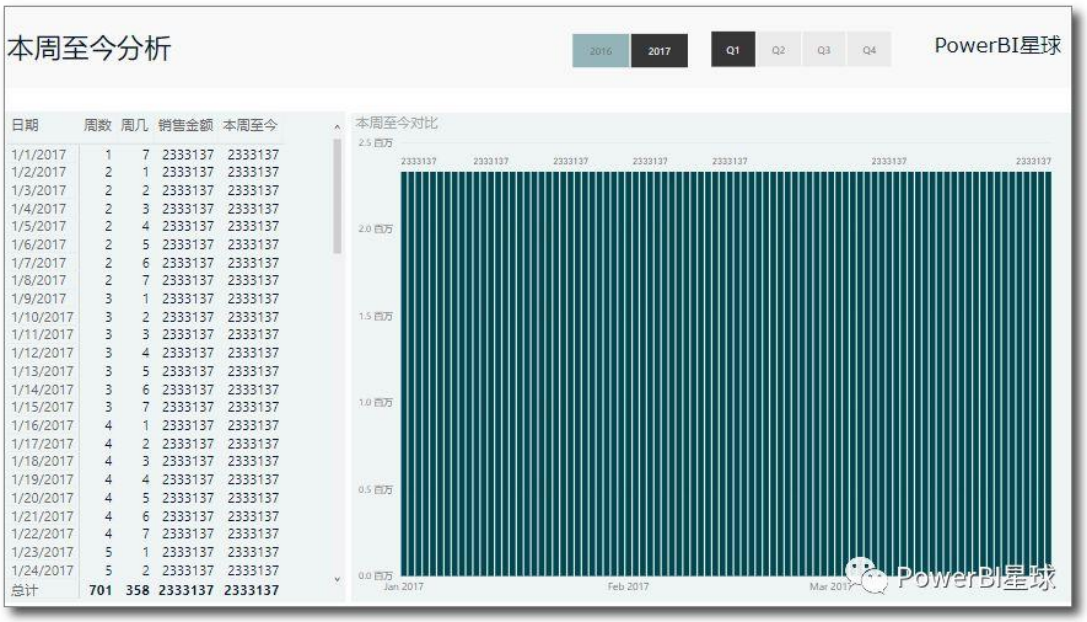
出现这种情况就是按照错误提示进行更改，再仔细研究一下该函数的介绍文档，弄清楚该函数需要有几个参数，以及每个参数分别是什么类型。

关于函数和参数的错误，主要是要理解 DAX 各类函数及其参数的用法，

请参考：DAX 中的表函数和值函数

04 | 显示结果不符合预期

写度量值时没有任何错误提示，把它拖入到图表中也没有错误，可是显示的数据却很奇怪，完全不是想象中的，



日期	周数	周几	销售金额	本周至今
1/1/2017	1	7	2333137	2333137
1/2/2017	2	1	2333137	2333137
1/3/2017	2	2	2333137	2333137
1/4/2017	2	3	2333137	2333137
1/5/2017	2	4	2333137	2333137
1/6/2017	2	5	2333137	2333137
1/7/2017	2	6	2333137	2333137
1/8/2017	2	7	2333137	2333137
1/9/2017	3	1	2333137	2333137
1/10/2017	3	2	2333137	2333137
1/11/2017	3	3	2333137	2333137
1/12/2017	3	4	2333137	2333137
1/13/2017	3	5	2333137	2333137
1/14/2017	3	6	2333137	2333137
1/15/2017	3	7	2333137	2333137
1/16/2017	4	1	2333137	2333137
1/17/2017	4	2	2333137	2333137
1/18/2017	4	3	2333137	2333137
1/19/2017	4	4	2333137	2333137
1/20/2017	4	5	2333137	2333137
1/21/2017	4	6	2333137	2333137
1/22/2017	4	7	2333137	2333137
1/23/2017	5	1	2333137	2333137
1/24/2017	5	2	2333137	2333137
总计	701	358	2333137	2333137

所有的数据都相同，没有按上下文进行计算？

这个问题一般都是维度表和事实表没有建立关系，或者没有建立正确的关系，因此度量值没有按照外部下上文正确计算。

遇到这种情况去建模视图下，更改一下关系就可以了。

这里也建议伙伴们，导入数据之后，先在建模视图下建立正确的关系图，然后再去着手去写度量值。如果系统自动建立的，也要检查一下，自动建立的关系是否与我们的分析逻辑一致。

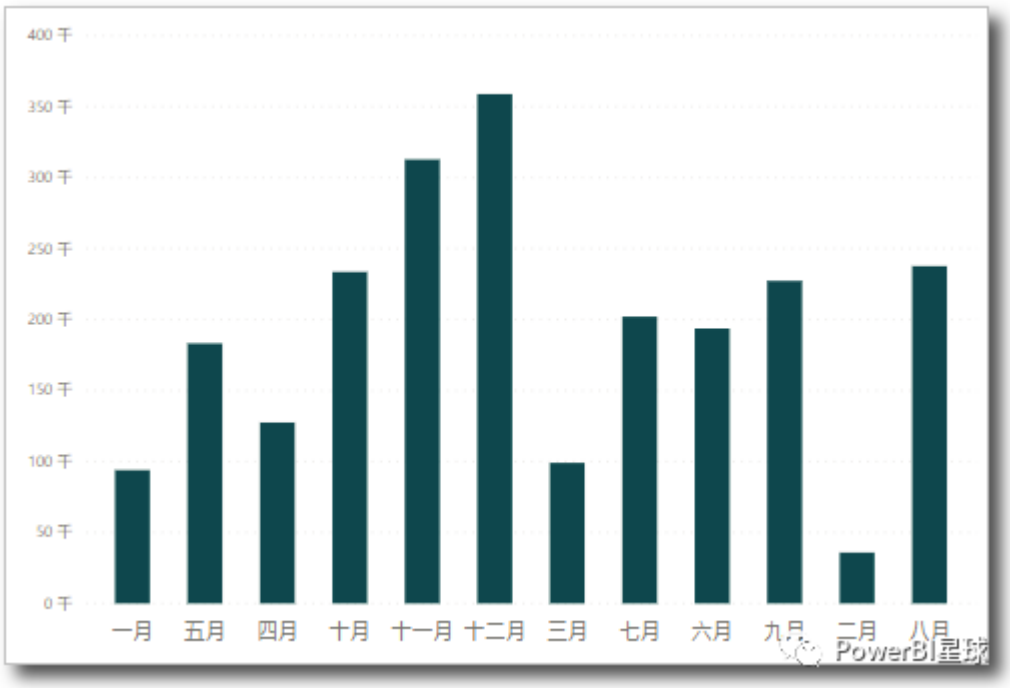
05 | 使用度量值作为筛选维度

还有不少人问我，为什么我建立的度量值无法拖入到图表里，我仔细一看，原来 TA 是想用这个度量值拖到图表的轴上，或者是切片器中。

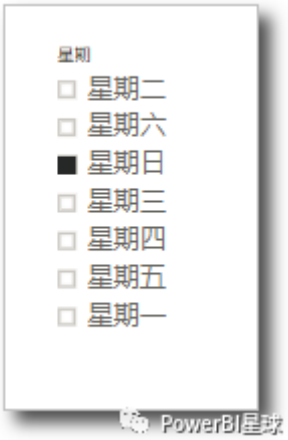
度量值作为动态的公式，只有在一定的上下文环境中才有确定的返回结果，所以 PowerBI 中默认它是不能作为筛选维度的。如果确实需要用这个维度，就去构建一个表吧。

06 | 排序错误

当把文本字段放到坐标轴或者切片器上，显示的顺序都乱了，完全不是自己想要的，就像下面这些，



↑月份坐标轴



↑星期切片器

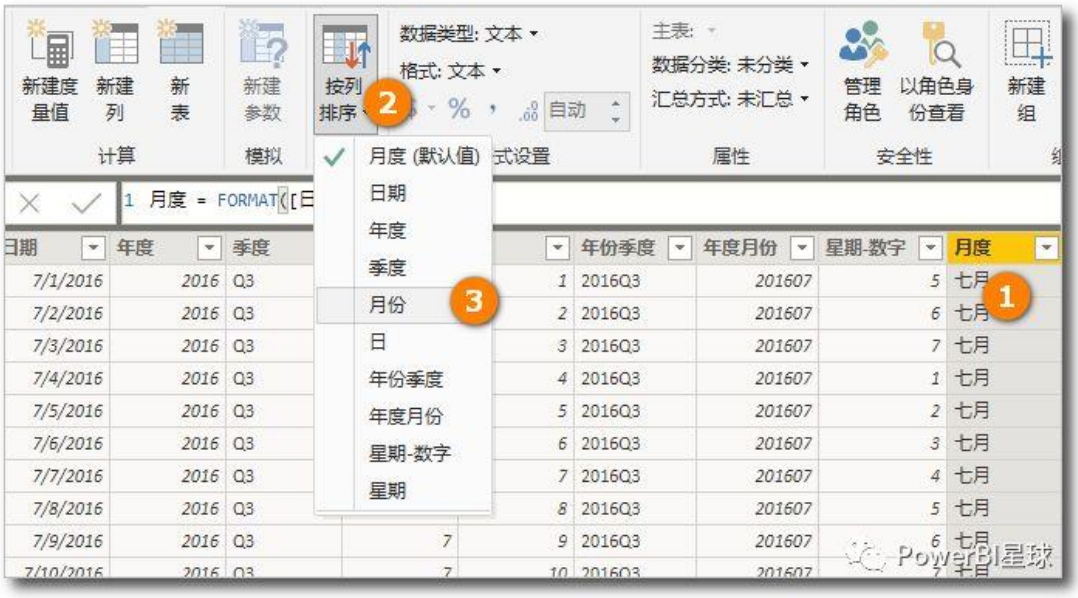
以上这些都是文本类的数据，文本数据默认是按照字母来排序的，但是这样的排序明显与我们的初衷不符，在图表的排序或者切片器的设置中，又没有办法直接调整过来，那怎么办呢？

解决办法：PowerBI 中有个功能，可以轻松排序，它就是：**按列排序。**

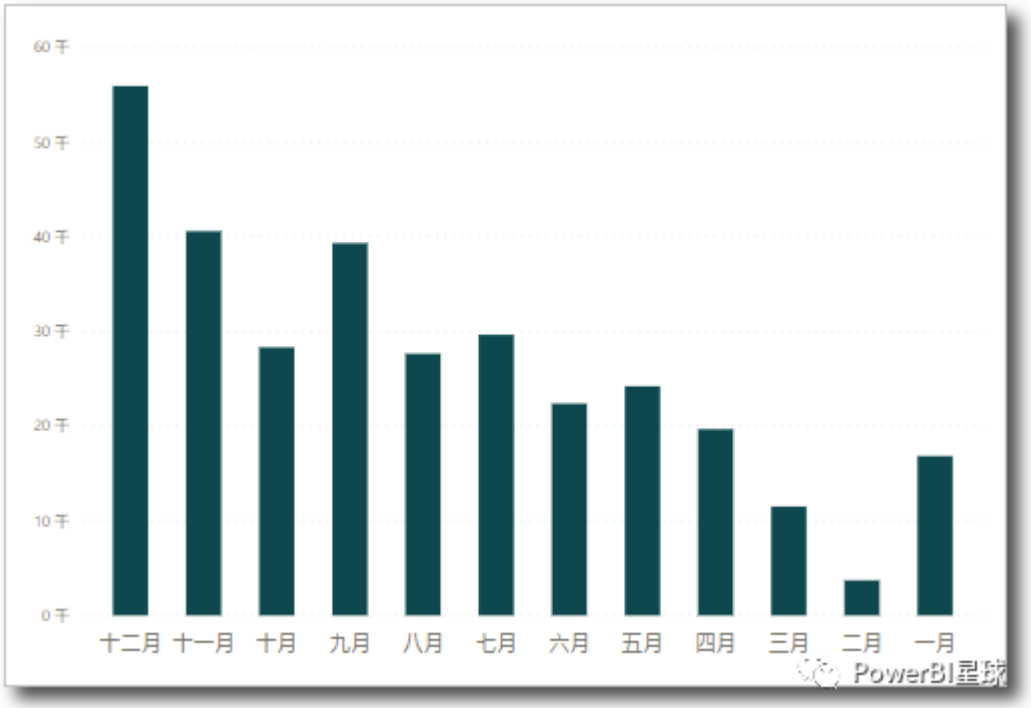
以调整中文的月份字段为例，来看看这个功能是怎么使用的。

日期表中已经有数字的月份，那么可以选中中文字段[月度]，点击“按列排序”，下拉框中选

择“月份”，



然后中文的月度，就按照数字顺序来进行排列了，



其他文本的排序都可以用这种方式来进行设置，不过首先一定要有对应的数字序列。

当我们添加有顺序要求的文本数据时，一定要同时给这些文本添加序号，比如上一篇文章（PowerBI 技巧：动态切换数据单位）中的金额单位，在输入数据的时候，同时添加了序号，

如果不这样做，在切片器中不会按照从小到大的单位顺序来排列（当然这个例子中用单位值也可以排序）。

所以养成为文本数据添加数字索引的好习惯，在任何场景下的文本排序都不再是问题。

有问题不可怕，让我们一起解决它。

公众号:PowerBI 星球

恭喜你完成了《七天入门 PowerBI》的学习内容，这不是结束，而是一个全新的开始，你已经踏入 PowerBI 的大门，一个新的领域在等着你去探索！

现在你可以在自己的工作中实际操练，从简单的开始，遇到困惑可随时提问、交流，根据需要可以选择继续学习 PowerBI 的某些模块，更多文章，请关注微信公众号「**PowerBI 星球**」。

欢迎加入知识星球

问题随时答疑

微信扫一扫，或者添加采悟私人微信号: **PowerBI001**

了解详情