

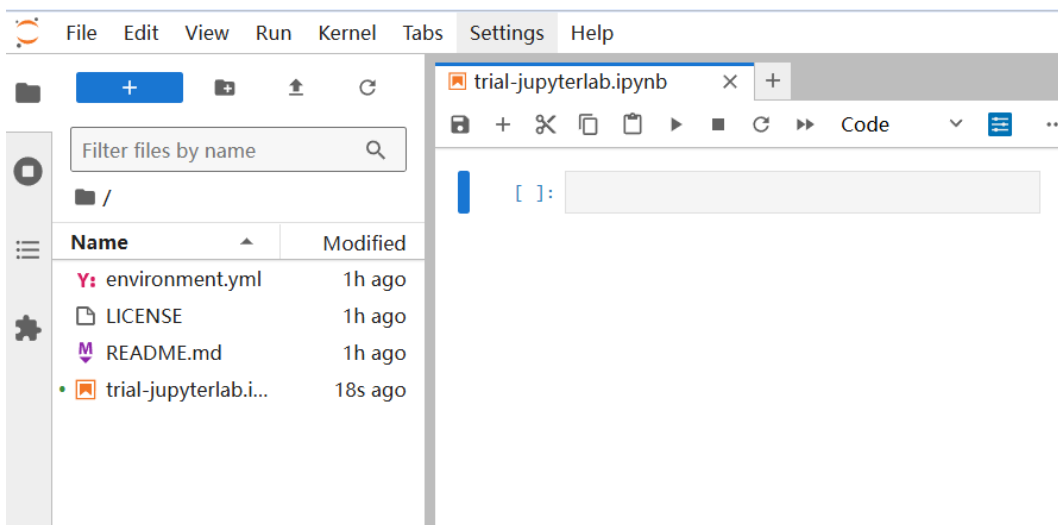
数据可视化和交互

——JupyterLab 、 Perspective

前言：终端（TUI）适合自动化，但在可视化与交互方面较弱；图形界面（GUI）擅长交互，但存在跨平台性和开放性不足的问题。基于浏览器的 Web 技术（如网页 app）兼具可视化、交互性、跨平台性和开放性，是理想选择，但其技术栈（HTML+CSS+JavaScript）对初学者门槛较高。

一、准备工作

- 1.准备环境（注意：安装 jupyterlab）
2. 终端运行 `jupyter lab` 命令，启动后端 (Backend) 服务，在浏览器访问前端 (Frontend) 页面
3. 在 JupyterLab 页面里，新建一个 Notebook，改名为 `trial-jupyterlab.ipynb`



PS：此处内容是可以保存、分享的，但分享内容不可运行

二、JupyterLab 的使用（含 Markdown）

- 1.在单元格 (Cell) 里编写 Python 代码，按 `Shift+Enter` 运行 Cell 并下移
- 2.在单元格 (Cell) 上按 `ESC` 切换到命令模式 (command mode)，按 `Enter` 切换到编写模式 (edit mode)
- 3.在单元格 (Cell) 的**命令模式下**：
 - 按 `j` 选择下一个，按 `k` 选择上一个（或者使用上下键）；
 - 按 `a` 在上方添加，按 `b` 在下方添加（或使用按钮）；
 - 按 `dd` 删除，按住 `Shift` 多选；
 - 按 `x` 剪切，按 `c` 复制，按 `v` 粘贴；
 - 按 `Shift+M` 合并，按 `z` 撤销，按 `Shift+Z` 重做，按 `Shift+L` 显示/隐藏代码行号。
- 4.在单元格 (Cell) 的**编写模式下**，按 `Ctrl+Shift+-` 切分单元格
- 5.按右上角按钮显示/隐藏 Minimap
- 6.运行单元格 (Cell) 注意序号单调递增

- 7.单元格最后一行如果是表达式 (expression) 且运行后返回的对象不是 None，则计输出 (Out)，否则只计输入 (In)，序号为 i 的输出，可以用 `_i` 变量来引用
- 8.单元格 (Cell) 序号为 * 表示代码运行中，尚未返回，按 ii 可以打断 (KeyboardInterrupt) (类似于终端的 Ctrl+C)

```
[2]: import time

while True:
    print('hello')
    time.sleep(3)

hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello

KeyboardInterrupt                                Traceback (most recent call last)
Cell In[2], line 5
      3 while True:
      4     print('hello')
----> 5     time.sleep(3)

KeyboardInterrupt:
```

- 9.在单元格 (Cell) 的命令模式下，按 00 重启后端 Python 解释器 (被 Jupyter 称为 Kernel)，重启后先在菜单里选择 “Edit / Clear Outputs of All Cells” 清空全部页面显示的输出，然后从上至下重新运行一遍代码 (Shift+Enter)（这种操作常见于修改代码后）
- 10.在单元格 (Cell) 的命令模式下，按 m 切换至 Markdown 模式，按 y 切换至 Python 模式
- 11.用 AI 生成一段示例 Markdown 代码，复制粘贴进 Markdown 单元格，运行以呈现 (Render)

```
# 标题示例

## 二级标题
### 三级标题

---

**加粗文本** 和 *斜体文本* 或 删除线

> 引用文本: Markdown 让写作更简单
```

标题示例

二级标题

三级标题

加粗文本 和 *斜体文本* 或 ~~删除线~~

> 引用文本: Markdown 让写作更简单

Markdown 代码运行后复制粘贴到 word 变得可识别

- 12.用 AI 生成一段示例 HTML 代码，复制粘贴进 Markdown 单元格，运行以呈现 (Render)；注意 Markdown 单元格不支持 CSS

```
<h1>欢迎来到我的网页</h1>

<!-- 导航菜单 -->
<nav>
  <ul>
    <li><a href="#home">首页</a></li>
    <li><a href="#about">关于</a></li>
    <li><a href="#contact">联系方式</a></li>
  </ul>
</nav>

<!-- 主要内容区 -->
<main>
  <section id="home">
    <h2>主页内容</h2>
    <p>这是一个简单的HTML文档示例，不包含任何CSS样式。</p>
    <p>HTML定义了网页的结构和内容。</p>
```

欢迎来到我的网页

- 首页
- 关于
- 联系方式

主页内容

这是一个简单的HTML文档示例，不包含任何CSS样式。

HTML定义了网页的结构和内容。

© 2023 我的网站. 保留所有权利.

联系方式: contact@example.com

13. 用 AI 生成一段示例 LaTeX 数学公式代码，复制粘贴进 Markdown 单元格，运行以呈现 (Render)；
注意要用 $(行内模式) 或 $(整行模式) 包围$$

LaTeX 数学公式示例

行内公式

在文本中嵌入公式，例如欧拉公式： $e^{i\pi} + 1 = 0$ 或者质能方程 $E = mc^2$ 。

独立公式块

常用数学公式：

二次方程求根公式

```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```

LaTeX 数学公式示例

行内公式

在文本中嵌入公式，例如欧拉公式： $e^{i\pi} + 1 = 0$ 或者质能方程 $E = mc^2$ 。

独立公式块

常用数学公式：

14. 关闭前端页面，在后端按 Ctrl+C 打断运行中的服务，回到 Bash 提示符

三、通过 Tushare 软件包下载保存数据

1. environment.yml 添加 pip: tushare (conda-forge 没有收录 tushare，只能从 PyPI 安装，) 依赖项

```
! environment.yml
1  name: week07
2  channels:
3    - conda-forge
4  dependencies:
5    - python=3.12
6    - wat-inspector
7    - jupyterlab
8    - pip
9    - pip:
10     - tushare
```

2. 设置 Tushare Token

```
(week07)
71970@13269: ~$ MINGW64 ~/repo/week07 (main)
$ python
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 22:08:16) [MS
C v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tushare as ts
>>> ts.set_token(" ")
>>> quit()
```

PS: 运行 set_token 函数会把 Token 字符串保存在 ~/tk.csv 文件里，今后每次使用 tushare 软件包请求数据时都会自动读取并发送 Token，不需要反复设置。

3. 向 Tushare 服务器请求 IPO 新股列表数据，并保存在本地：

其中请求数据函数返回的对象 df 是 pandas.DataFrame 类型，调用其 to_parquet 方法能够将内存 (memory) 中的 DataFrame 数据按照 Parquet 格式序列化 (serialize) 为字节串 (bytes) 保存到磁盘 (Parquet 是大数据领域的首选格式，压缩效率很高，已经成为业界标准)

```

71970@MINGW64 ~/repo/week07 (main)
$ ipython
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 22:08:16) [MS
C v.1943 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 9.1.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: Use 'F2' or %edit with no arguments to open an empty editor with a temp
orary file.


In [1]: import tushare as ts
In [2]: pro = ts.pro_api()
In [3]: pro
Out[3]: <tushare.pro.client.DataApi at 0x21fa89520c0>
In [4]: type(pro)
Out[4]: tushare.pro.client.DataApi
In [5]: df = pro.new_share()
In [6]: df
Out[6]:
   ts_code sub_code name ... limit_amount funds ballot
0    301595.SZ   301595 太力科技 ...         0.65    0.000    0.00
1    688755.SH   787755 汉邦科技 ...         0.50    0.000    0.00
2    301636.SZ   301636 泽润新能 ...         0.45    0.000    0.00
3    001400.SZ   001400 江顺科技 ...         1.50    5.604    0.01
4    301560.SZ   301560 众捷汽车 ...         0.70    5.016    0.02
...
1995 600989.SH   730989 宝丰能源 ...        22.00   81.550    0.25
1996 300778.SZ   300778 新城市 ...         2.00    5.466    0.02
1997 002953.SZ   002953 日丰股份 ...         1.70    4.526    0.03
1998 603697.SH   732697 有友食品 ...         3.10    6.257    0.05
1999 300772.SZ   300772 运达股份 ...         2.80    4.792    0.04

[2000 rows x 12 columns]

In [7]: type(df)
Out[7]: pandas.core.frame.DataFrame
In [8]: df.to_parquet("new_share.parquet")

```

4. 同上，访问 stock_basic 接口，并将数据保存为 stock_basic.parquet 文件



exchange str N 交易所 SSE上交所 SZSE深交所 BSE北交所

is_hs str N 是否沪深港通标的, N否 H沪股通 S深股通

输出参数

名称	类型	默认显示	描述
ts_code	str	Y	TS代码
symbol	str	Y	股票代码
name	str	Y	股票名称
area	str	Y	地域
industry	str	Y	所属行业
fullname	str	N	股票全称
enname	str	N	英文全称
cnsPELL	str	Y	拼音缩写
market	str	Y	市场类型 (主板/创业板/科创板/CDR)
exchange	str	N	交易所代码
curr_type	str	N	交易货币
list_status	str	N	上市状态 L上市 D退市 P暂停上市
list_date	str	Y	上市日期
delist_date	str	N	退市日期
is_hs	str	N	是否沪深港通标的, N否 H沪股通 S深股通
act_name	str	Y	实控人名称

```

None
5411 920116.BJ 920116 星图测控 ... 20250102 None
5412 920118.BJ 920118 太湖远太 ... 20240822 None
5413 920128.BJ 920128 胜业电气 ... 20241129 None
5414 689009.SH 689009 九号公司-WD ... 20201029 None

[5415 rows x 10 columns] 需要指定 fields 参数获取全部字段

In [10]: df = pro.stock_basic(fields='ts_code,symbol,name,area,industry,ful
         : lname,enname,cnsPELL,market,exchange,curr_type,list_status,list_da
         : te,delist_date,is_hs,act_name,act_ent_type')

In [11]: df
Out[11]:
   ts_code symbol name ... is_hs act_name act_ent_type
0  000001.SZ 000001 平安银行 ... S 无实际控制人
1  000002.SZ 000002 万科A ... S 深圳市人民政府国有资产监督管
理委员会 地方国企
2  000004.SZ 000004 国华网安 ... N 李映彤
   民营企业
3  000006.SZ 000006 深振业A ... S 深圳市人民政府国有资产监督管
理委员会 地方国企
4  000007.SZ 000007 全新好 ... N 王玩虹
   民营企业
...
5410 920111.BJ 920111 聚厚科技 ... N
5411 920116.BJ 920116 星图测控 ... N
5412 920118.BJ 920118 太湖远太 ... N
5413 920128.BJ 920128 胜业电气 ... N
5414 689009.SH 689009 九号公司-WD ... None

[5415 rows x 17 columns]

In [12]: df.to_parquet("stock_basic.parquet") 存储
In [13]:

```

PS:查看上述两个数据文件

```
(week07)
71970@1326: ~$ MINGW64 ~/repo/week07 (main)
$ ls -lh *.parquet
-rw-r--r-- 1 71970 197609 118K 4月 24 20:22 new_share.parquet
-rw-r--r-- 1 71970 197609 420K 4月 24 21:08 stock_basic.parquet
```

四、Perspective（实现交互式可视化）

- 1.修改 environment.yml 文件，添加 perspective-python 和 polars 依赖项
- 2.启动 JupyterLab，新建一个 Notebook，改名为 trial-perspective.ipynb
- 3.调用 polars.read_parquet 函数，分别读取磁盘中的 new_share.parquet 文件和 stock_basic.parquet 文件得到内存中的 **polars.DataFrame** 对象，命名为 d1 和 d2

```
[1]: import polars as pl

[2]: d1 = pl.read_parquet("new_share.parquet")

[3]: d2 = pl.read_parquet("stock_basic.parquet")

[ ]:
```

- 4.进行适当的列变换，把实际上是日期类型的列，从 polars.String() 类型转换为 polars.Date() 类型

The screenshot shows a JupyterLab notebook titled 'trial-perspective.ipynb'. It displays two code cells and their corresponding DataFrame outputs. A red arrow points from the 'ipo_date' and 'issue_date' columns in the first DataFrame to the same columns in the second DataFrame, indicating the transformation process.

Cell [4]: d1

shape: (2,000, 12)

ts_code	sub_code	name	ipo_date	issue_date	amount	market_amount	price	pe	limit_amount	funds	ballot
str	str	str	str	str	f64	f64	f64	f64	f64	f64	f64
"301595.SZ"	"301595"	"太力科技"	"20250508"	null	2707.0	0.0	0.0	0.0	0.65	0.0	0.0
"688755.SH"	"787755"	"汉邦科技"	"20250507"	null	2200.0	0.0	0.0	0.0	0.5	0.0	0.0
"301636.SZ"	"301636"	"泽润新能"	"20250428"	null	1597.0	0.0	0.0	0.0	0.45	0.0	0.0
"001400.SZ"	"001400"	"江顺科技"	"20250415"	"20250424"	1500.0	1500.0	37.36	15.32	1.5	5.604	0.01
"301560.SZ"	"301560"	"众捷汽车"	"20250415"	"20250425"	3040.0	1216.0	16.5	21.3	0.7	5.016	0.02
...
"600989.SH"	"730989"	"宝丰能源"	"20190430"	"20190516"	73336.0	66002.0	11.12	22.07	22.0	81.55	0.25
"300778.SZ"	"300778"	"新城市"	"20190425"	"20190510"	2000.0	2000.0	27.33	22.99	2.0	5.466	0.02
"002953.SZ"	"002953"	"日丰股份"	"20190424"	"20190509"	4302.0	3872.0	10.52	16.34	1.7	4.526	0.03
"603697.SH"	"732697"	"有友食品"	"20190423"	"20190508"	7950.0	7155.0	7.87	13.92	3.1	6.257	0.05
"300772.SZ"	"300772"	"运达股份"	"20190417"	"20190426"	7349.0	6614.0	6.52	22.97	2.8	4.792	0.04

Cell [5]: d1.with_columns(
 ipo_date=pl.col.ipo_date.str.to_date("%Y%m%d"),
 issue_date=pl.col.issue_date.str.to_date("%Y%m%d"),
)

shape: (2,000, 12)

ts_code	sub_code	name	ipo_date	issue_date	amount	market_amount	price	pe	limit_amount	funds	ballot
str	str	str	date	date	f64	f64	f64	f64	f64	f64	f64
"301595.SZ"	"301595"	"太力科技"	2025-05-08	null	2707.0	0.0	0.0	0.0	0.65	0.0	0.0
"688755.SH"	"787755"	"汉邦科技"	2025-05-07	null	2200.0	0.0	0.0	0.0	0.5	0.0	0.0

5.把 d1 或 d2 作为参数传递给 perspective.widget.PerspectiveWidget 类型进行初始化，返回的对象会呈现在 Notebook 的 Output 里

```
Launcher trial-perspective.ipynb +
Code
[1]: import polars as pl
    from perspective.widget import PerspectiveWidget

[2]: d1 = pl.read_parquet("new_share.parquet")

[3]: d2 = pl.read_parquet("stock_basic.parquet")

[4]: d1 = d1.with_columns(
    ipo_date=pl.col.ipo_date.str.to_date("%Y%m%d"),
    issue_date=pl.col.issue_date.str.to_date("%Y%m%d"),
)

[5]: d2 = d2.with_columns(
    list_date=pl.col.list_date.str.to_date("%Y%m%d"),
)

[6]: PerspectiveWidget(d1)

[6]:
```

ts_code	sub_code	name	ipo_date	issue_date	amount	market_amount	price	pe	limit_amount	funds	ballot
301595.SZ	301595	太力科技	2025/5/7	-	2,707.00	0.00	0.00	0.00	0.65	0.00	0.00
688755.SH	787755	汉邦科技	2025/5/6	-	2,200.00	0.00	0.00	0.00	0.50	0.00	0.00
301636.SZ	301636	泽润新能	2025/4/27	-	1,597.00	0.00	0.00	0.00	0.45	0.00	0.00

6. 在 PerspectiveWidget 默认的 Datagrid 视图下，尝试实践：

①修改各种列数据类型 (文本、数值、日期) 的显示风格 (style): 点击变量下的 Edit

PerspectiveWidget(d2)

untitled 5,415 x 17

Area

Datagrid

Style

Format: none

Color: series

Series

ts_code	symbol	name	area	industry	fullname	enname
Edit	Edit	Edit	Edit	Edit	Edit	Edit
000001.SZ	000001	平安银行	深圳	银行	平安银行股份有限公司	Ping An Bank Co., Ltd
000002.SZ	000002	万科A	深圳	全国地产	万科企业股份有限公司	China Vanke Co.,Ltd.
000004.SZ	000004	国华网安	深圳	软件服务	深圳国华网安科技股份有限公司	Shenzhen Guohua Netwo
000006.SZ	000006	深振业A	深圳	区域地产	深圳市振业(集团)股份有限公司	Shenzhen Zhenye(Group
000007.SZ	000007	全新好	深圳	其他商业	深圳市全新好股份有限公司	Shenzhen Quanyinhao C
000008.SZ	000008	神州高铁	北京	运输设备	神州高铁技术股份有限公司	China High Speed Rail
000009.SZ	000009	中国宝安	深圳	电气设备	中国宝安集团股份有限公司	China Baoan Group Co.
000010.SZ	000010	美丽生态	深圳	建筑工程	深圳美丽生态股份有限公司	Shenzhen EcoBeauty Co
000011.SZ	000011	深物业A	深圳	房产服务	深圳市物业发展(集团)股份有限公司	Shenzhen Properties &
000012.SZ	000012	南玻A	深圳	玻璃	中国南玻集团股份有限公司	CSG Holding Co., Ltd.
000014.SZ	000014	沙河股份	深圳	全国地产	沙河实业股份有限公司	Shahe Industrial Co.,
000016.SZ	000016	深康佳A	深圳	家用电器	康佳集团股份有限公司	Konka Group Co.,Ltd
000017.SZ	000017	深中华A	深圳	服饰	深圳中华自行车(集团)股份有限公司	Shenzhen China Bicycl
000019.SZ	000019	深粮控股	深圳	农业综合	深圳市深粮控股股份有限公司	Shenzhen Cereals Hold
000020.SZ	000020	深华发A	深圳	元器件	深圳中恒华发股份有限公司	Shenzhen Zhongheng Hu
000021.SZ	000021	深科技	深圳	元器件	深圳长城开发科技股份有限公司	Shenzhen Kaifa Techno
000025.SZ	000025	特力A	深圳	综合类	深圳市特力(集团)股份有限公司	Shenzhen Tellus Holdi

Group By

Split By

Order By

Where

Columns

- ts_code
- symbol
- name
- area
- industry
- fullname
- enname
- cnsoell

②设置 Group By 选项，选择某些列作为分组依据 (纵向排列)，选择其他某些列进行汇总 (注意汇总方式有多种函数选项)

[7]: PerspectiveWidget(d2)

untitled 9 (5,415) x 3 (17)

	ts_code	industry2	industry
- TOTAL	5,415	电气设备	
- BSE	265	汽车配件	
北交所	265	汽车配件	
- SSE	2,282	半导体	
主板	1,696	汽车配件	
科创板	586	半导体	
- SZSE	2,868	元器件	
主板	1,487	元器件	
创业板	1,381	软件服务	

industry

Style

Foreground

Color Range

Max Value

Background

Style

Minimum Integer Digits

Rounding Increment

Notation

Use Grouping

Sign Display

Datagrid

Group By

Split By

Order By

Where

Columns

market

name

symbol

list_date

列，可以自己新建一个industry

隐藏

③设置 Split By 选项，选择某些列作为拆分依据 (横向排列)

[7]: PerspectiveWidget(d2)

untitled 4 (5,415) x 4 (17)

	主板	创业板	北交所	科创板
	ts_code	ts_code	ts_code	ts_code
- TOTAL	3,183	1,381	265	586
BSE	-	-	265	-
SSE	1,696	-	-	586
SZSE	1,487	1,381	-	-

Datagrid

Group By

Split By

Order By

Where

Columns

④设置 Order By 选项，选择某些列作为排序依据 (注意可以切换 升序/降序)

⑤设置 Where 选项，选择某些列，进一步设置条件，进行数据行 (观测) 方向的过滤

[7]: PerspectiveWidget(d2)

untitled 435 (5,415) x 2 (17)

ts_code	list_date
000001.SZ	1991/4/2
000002.SZ	1991/1/28
000004.SZ	1991/1/13
000006.SZ	1992/4/26
000007.SZ	1992/4/12
000008.SZ	1992/5/6
000009.SZ	1991/6/24
000010.SZ	1995/10/26
000011.SZ	1992/3/29
000012.SZ	1992/2/27
000014.SZ	1992/6/1
000016.SZ	1992/3/26
000017.SZ	1992/3/30
000019.SZ	1992/10/11
000020.SZ	1992/4/27
000021.SZ	1994/2/1

Datagrid

Group By

Split By

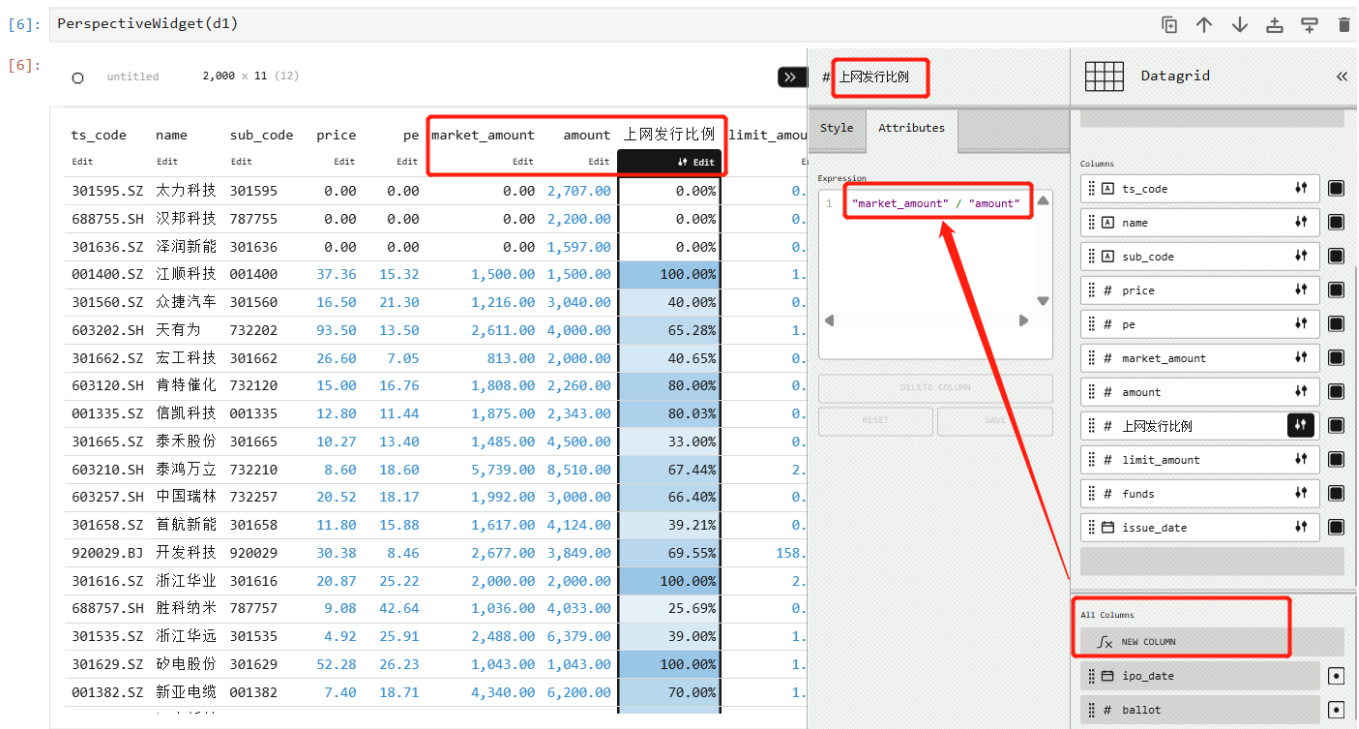
Order By

Where

Columns

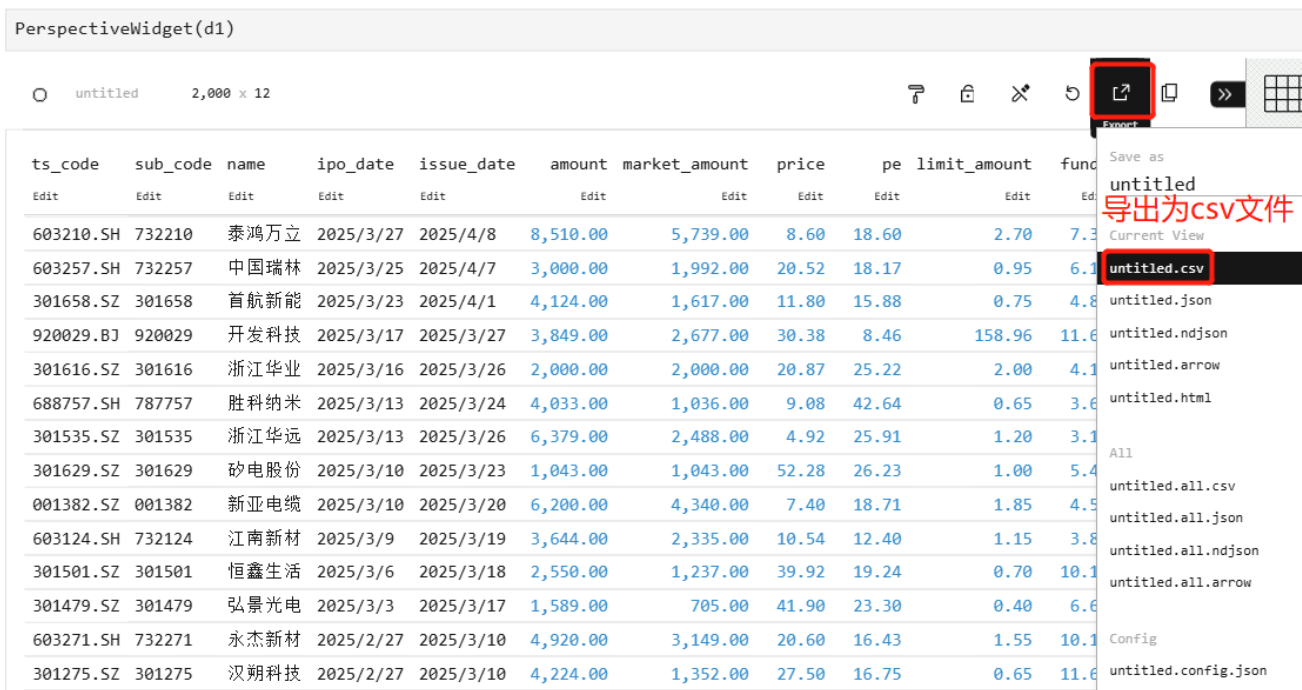
⑥设置 Columns 选项，选择要显示的数据列 (变量)，及其显示的先后顺序

⑦AllColumns 部分，是能够显示但没有显示的数据列(变量)，可以点击 NEW COLUMN 添加衍生计算出的新列，需要用 ExprTK 语言书写表达式代码，变量名用双引号包围，字符串用单引号包围



7. 在 PerspectiveWidget 图形界面依靠鼠标 (手动) 所做的设置 (configure)，可以导出代码，根据导出的代码，可以修改我们的代码，使得我们运行代码直接就能得到我们所需要的视图 (自动化)

①在 PerspectiveWidget 的右上方有按钮，可以把图形界面的数据或设置 (configure) 导出 (export) 为文件



②把设置 (config.json) 复制（右上 copy 按钮）粘贴进 Notebook Cell，保存成字符串 (str)

③也可以把设置 (config.json) 导出为文件，用 `pathlib.Path.read_text` 方法从文件读取出字符串 (str)



The screenshot shows a Jupyter Notebook interface. The main area contains the following code:

```
[17]: config = '{"version": "3.6.0", "plugin": "Datagrid", "plugin_config": {"columns": {}, "edit_mode": "READ_ONLY", "scroll_lock": false, "settings": true, "theme": "Pro Light", "title": null, "group_by": [], "split_by": [], "sort": [], "filter": [], "expressions": {}, "columns_config": {"ts_code": "sub_code", "name": "ipo_date", "issue_date": "amount", "market_amount": "price", "pe": "limit_amount", "funds": "ballot"}, "aggregates": {}}'

[10]: print(config)

[11]: from pathlib import Path

[15]: config2 = Path("C:/Users/71970/Desktop/untitled.config.json").read_text(encoding="utf8")

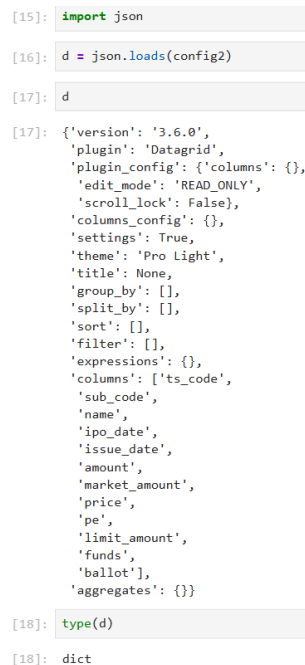
[16]: print(config2)

[6]: PerspectiveWidget(d1)

[6]:
```

A red box highlights the first two lines of code, with a red arrow pointing to the text "复制粘贴到cell (字符串形式)". A yellow box highlights the code for reading the file, with a yellow arrow pointing to the text "下载保存". The file explorer sidebar on the right shows a file named "untitled.config.json" selected.

④可以用 `json.loads` 函数将无结构的字符串 (str) 解析为有结构的 (structured) Python 字典，这样就容易在 Notebook 里美化呈现，也容易进一步通过 Python 代码访问内部的具体设置



The screenshot shows the following code and its output in a Jupyter Notebook:

```
[15]: import json

[16]: d = json.loads(config2)

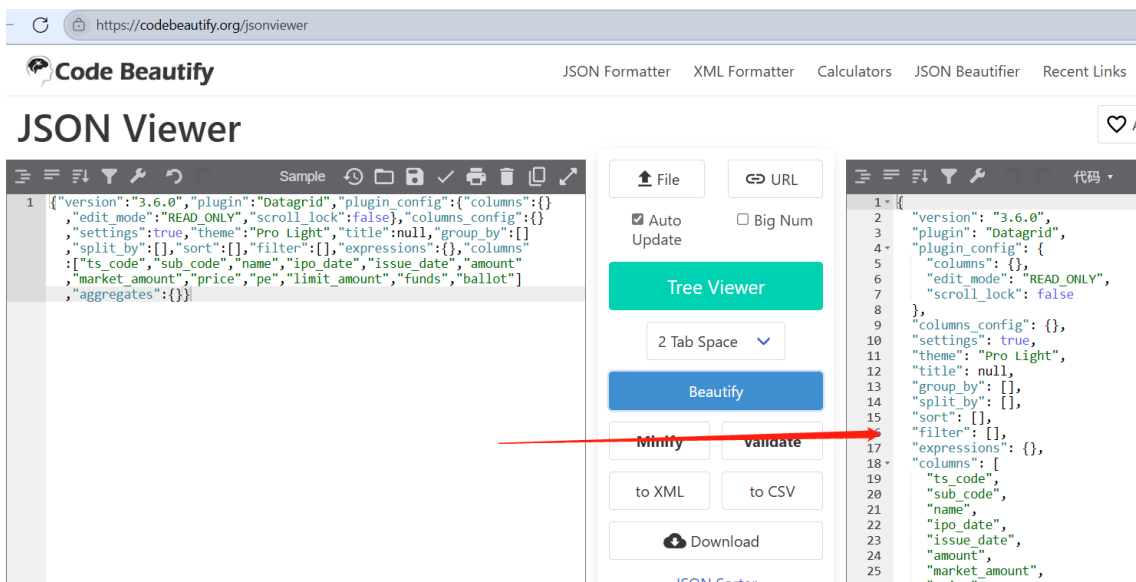
[17]: d

[17]: {'version': '3.6.0',
      'plugin': 'Datagrid',
      'plugin_config': {'columns': {}},
      'edit_mode': 'READ_ONLY',
      'scroll_lock': False,
      'columns_config': {},
      'settings': True,
      'theme': 'Pro Light',
      'title': None,
      'group_by': [],
      'split_by': [],
      'sort': [],
      'filter': [],
      'expressions': {},
      'columns': ['ts_code',
                  'sub_code',
                  'name',
                  'ipo_date',
                  'issue_date',
                  'amount',
                  'market_amount',
                  'price',
                  'pe',
                  'limit_amount',
                  'funds',
                  'ballot'],
      'aggregates': {}}

[18]: type(d)

[18]: dict
```

⑤也可以把复制到剪贴板的 JSON 字符串，粘贴进某个在线的 JSON 工具网站进行美化



⑥根据导出的设置代码，在初始化 PerspectiveWidget 类型时，传入适当的参数进行设置，运行代码，观察是否符合我们的期望

[21]:

PerspectiveWidget(d2)

[21]:

untitled5,415 × 17

ts_code	symbol	name	area	industry	fullname	enname
Edit	Edit	Edit	Edit	Edit	Edit	Edit
000019.SZ	000019	深粮控股	深圳	农业综合	深圳市深粮控股股份有限公司	Shenzhen Cereals Holdings Co.,Ltd
000020.SZ	000020	深华发A	深圳	元器件	深圳中恒华发股份有限公司	Shenzhen Zhongheng Huafa Co.,Ltd.
000021.SZ	000021	深科技	深圳	元器件	深圳长城开发科技股份有限公司	Shenzhen Kaifa Technology Co., Ltd.
000025.SZ	000025	特力A	深圳	综合类	深圳市特力(集团)股份有限公司	Shenzhen Tellus Holding Co.,Ltd.
000026.SZ	000026	飞亚达	深圳	服饰	飞亚达精密科技股份有限公司	FIYTA Precision Technology Co., Ltd.
000027.SZ	000027	深圳能源	深圳	火力发电	深圳能源集团股份有限公司	Shenzhen Energy Group Co., Ltd.
000028.SZ	000028	国药一致	深圳	医药商业	国药集团一致药业股份有限公司	China National Accord Medicines Corpor
000029.SZ	000029	深深房A	深圳	区域地产	深圳经济特区房地产(集团)股份有限公司	Shenzhen Special Economic Zone Real Es
000030.SZ	000030	富奥股份	吉林	汽车配件	富奥汽车零部件股份有限公司	Fawer Automotive Parts Limited Company
000031.SZ	000031	大悦城	深圳	全国地产	大悦城控股集团股份有限公司	Grandjoy Holdings Group Co.,Ltd.
000032.SZ	000032	深桑达A	深圳	建筑工程	深圳市桑达实业股份有限公司	Shenzhen Sed Industry Co.,Ltd.
000034.SZ	000034	神州数码	深圳	软件服务	神州数码集团股份有限公司	Digital China Group Co., Ltd.
000035.SZ	000035	中国天楹	江苏	环境保护	中国天楹股份有限公司	China Tianying Inc.
000036.SZ	000036	华联控股	深圳	全国地产	华联控股股份有限公司	China Union Holdings Ltd.
000037.SZ	000037	深南电A	深圳	火力发电	深圳南山热电股份有限公司	Shenzhen Nanshan Power Co., Ltd.
000039.SZ	000039	中集集团	深圳	运输设备	中国国际海运集装箱(集团)股份有限公司	China International Marine Containers
000040.SZ	000040	*ST旭蓝	深圳	新型电力	东旭蓝天新能源股份有限公司	Tunghsu Azure Renewable Energy Co., Lt

[32]:

PerspectiveWidget(d2, config=d)

TypeError

Cell In[32], line 1

-----> 1 PerspectiveWidget(d2, config=d)

Traceback (most recent call last)

理论上是可以恢复area这一列的显示设置的，

此处报错，实际运用中要再调整

8. 把 PerspectiveWidget 切换为 Treemap 视图，尝试设置各种选项 (configure)，观察数据可视化的实际效果

Treemap (树形结构图) 用不同大小的矩形来体现数据的分类占比构成情况，还可以用矩形的颜色来体现第二个维度的数据 (文本或数值都可以)

可切换多种视图

limit_amount

funds

ballot

0.65	0.00	0.00
0.50	0.00	0.00
0.45	0.00	0.00
1.50	5.60	0.01
0.70	5.02	0.02
1.25	37.40	0.03
0.45	5.32	0.02
0.90	3.39	0.03
0.90	3.00	0.02
0.70	4.62	0.02
2.70	7.32	0.04
0.95	6.16	0.03
0.75	4.87	0.02

Datagrid

Treemap

Sunburst

Heatmap

Map Scatter

X Bar

X/Y Line

X/Y Scatter

The Data Visualisation Catalogue

About · Blog · Shop · Resources

Treemap

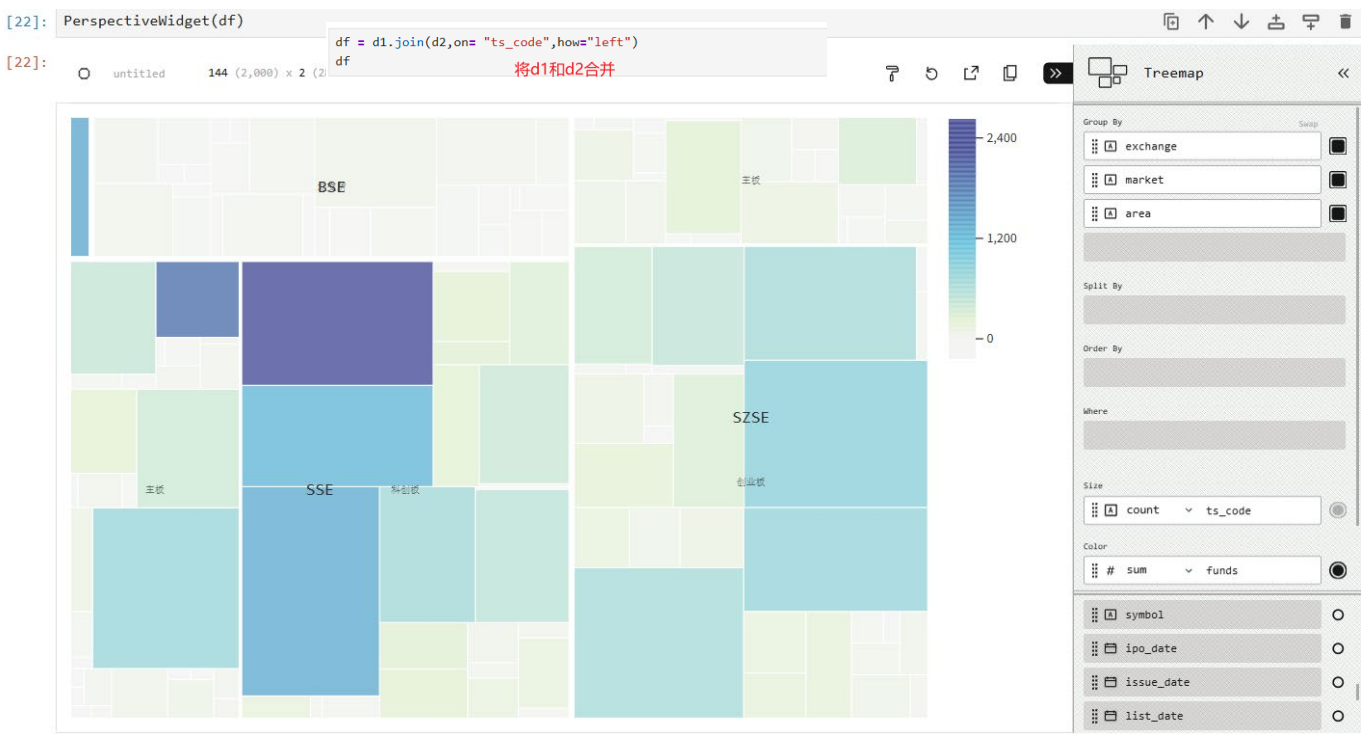
Description

Treemaps are an alternative way of visualising the hierarchical structure of a Tree Diagram while also displaying quantities for each category via area size. Each category is assigned a rectangle area with the subcategory rectangles nested inside.

Anatomy

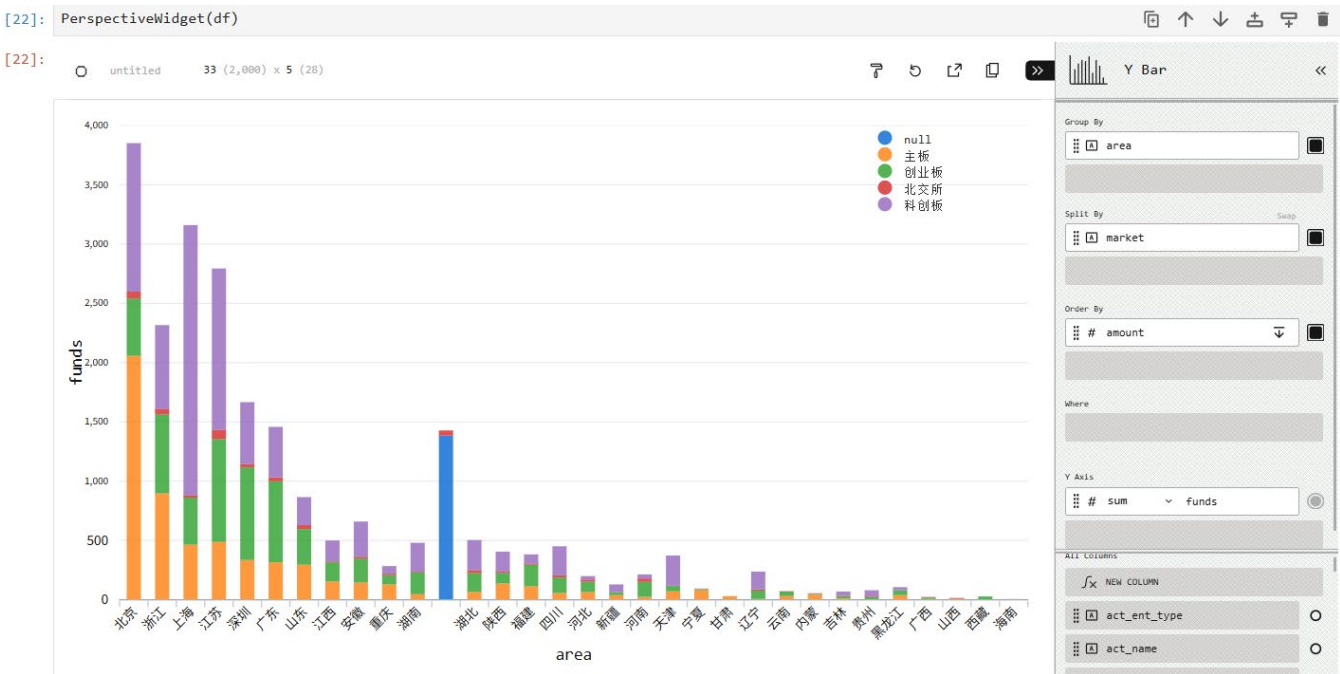
学习了解更多可视化结构图的概念、适用情形以及实现代码：[The Data Visualisation Catalogue](#)

效果如下：



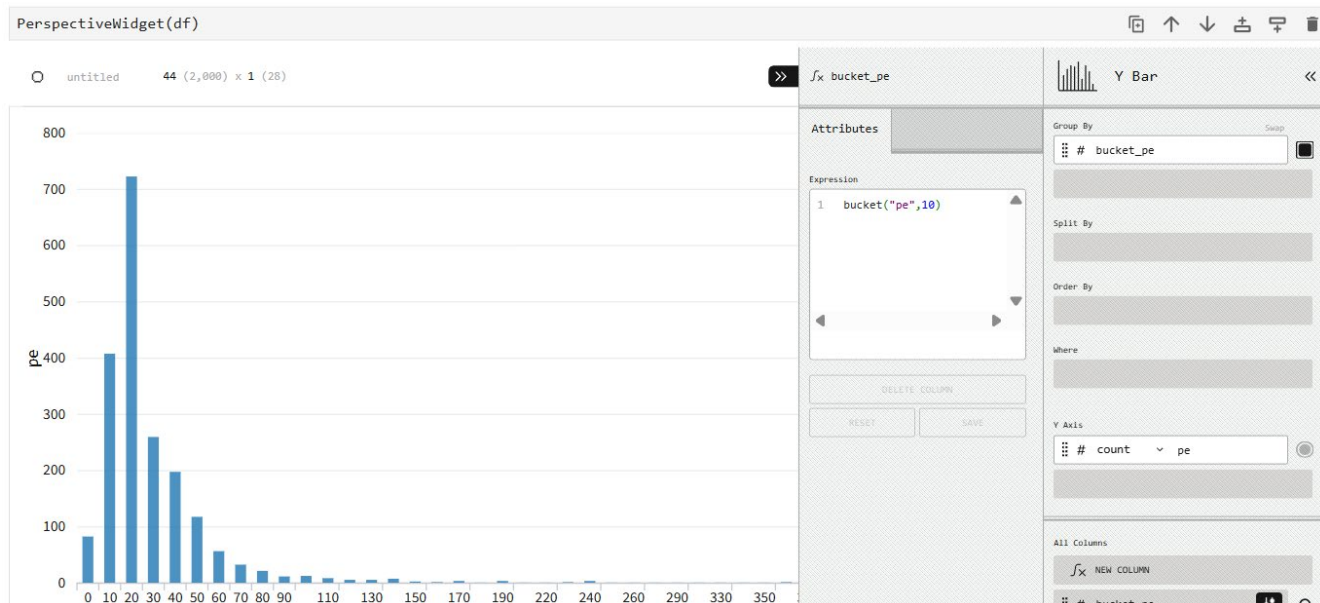
9. 把 PerspectiveWidget 切换为 Y Bar 视图，尝试设置各种选项，观察数据可视化的实际效果：

Y Bar (条形图/柱状图) 的横轴 (不同的条形) 是第一个维度，用 Group By 控制；纵轴 (条形的高度) 是第二个维度，用 Y Axis 控制 (支持多变量并列显示)；还可以把每个条形进一步拆分为多个颜色，用 Split By 控制



Y Bar 视图还可以用来实现一类很重要的统计制图 —— 直方图 (histogram)。对于数据表中的某一列连续型数值变量 (比如新股发行的市盈率 pe)，我们经常希望观察其分布 (distribution)。可以用 bucket 函数对连续变量进行 “分桶” (比如表达式 bucket("pe", 10))，生成一个新的离散变量 (比如命名为 bucket_pe)，然后把离散变量设置为 Y Bar 的横轴 (Group By)，把任意其他一列变量

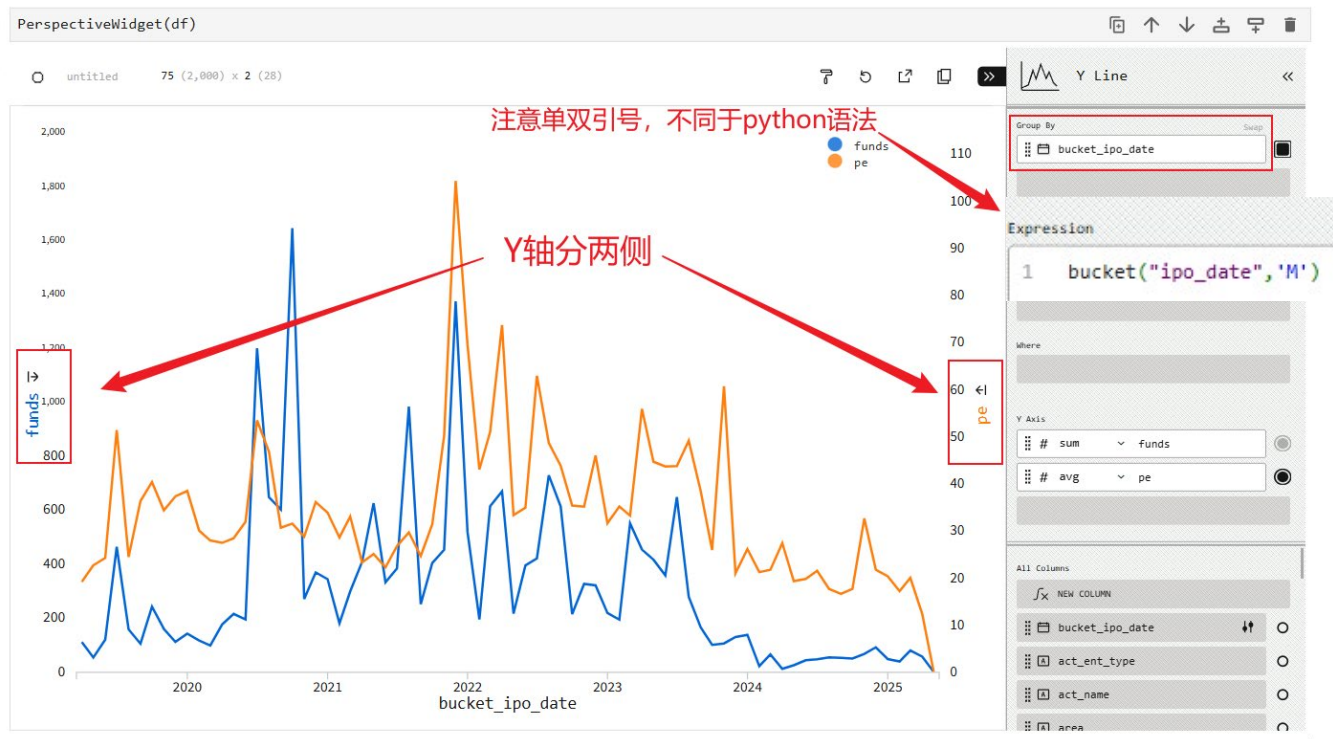
用 `count` (计数) 函数汇总，设置为纵轴 (Y Axis)。这样看到的就是直方图。“分桶” 在有的地方也叫 “分箱” (bin)，其粒度大小需要根据数据适当调节。



10. PerspectiveWidget 切换为 **Y Line** 视图，尝试设置各种选项，观察数据可视化的实际效果：

Y Line (折线图) 常用来绘制**时间序列**，横轴通常是时间，用 `Group By` 控制，纵轴 (折线的 Y 坐标) 通常是连续型数值变量 (经过汇总)，用 `Y Axis` 控制 (支持多序列同时显示)，还可以进一步拆分为多条序列，用 `Split By` 控制

使用示例数据，观察最近几年 A 股 IPO 市场的 融资额 (funds) 与 市盈率 (pe) 变化情况。

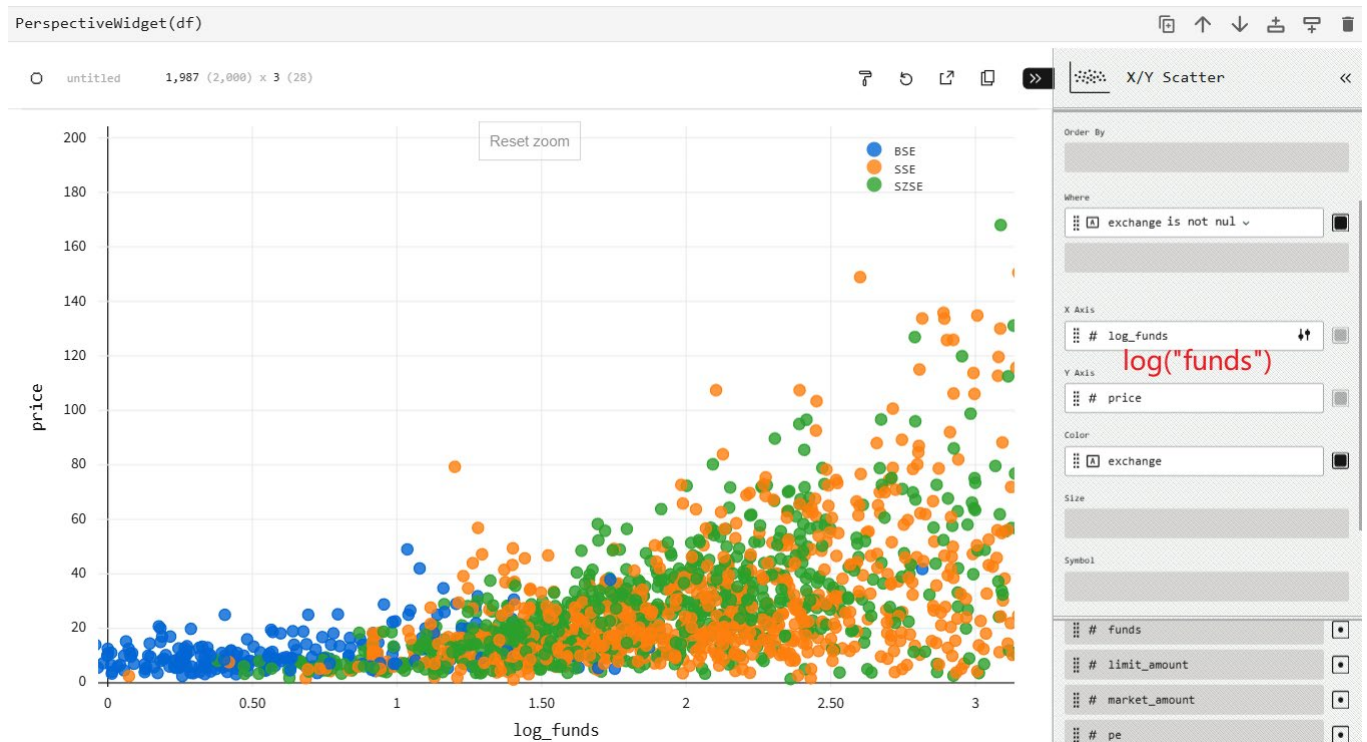


可以询问 AI 在某个时间段内发生了哪些影响 A 股 (或 IPO) 的重大国内外财经事件，由此加强对数据和现实背景的理解。

11. 切换为 **X/Y Scatter** 视图，尝试设置各种选项，观察数据可视化的实际效果：

X/Y Scatter (散点图) 常用来观察两个数值型连续变量之间的相关关系 (**correlation**)。数据首先可以进行分组汇总，每一个组对应一个散点，用 **Group By** 控制。然后把两个连续型数值变量分别设置为 **X Axis** 和 **Y Axis**，其汇总数值将作为每个散点的坐标

- ① 散点的分布如果特别不均匀，则意味着变量单位可能有问题，或者需要经过变换 (比如取对数)
- ② 散点的分布如果杂乱无规律，则意味着 **X** 与 **Y** 没有相关性
- ③ 散点的分布如果看起来能够拟合成一条直线 (即回归线，**regression**)，则意味着 **X** 与 **Y** 具有正的或负的相关性，意味着可能存在某些规律
- ④ 散点图上可以进一步体现更多的变量维度，比如可以把更多变量映射为散点的不同颜色 (**Color**)、大小 (**Size**)、符号 (**Symbol**)、标签 (**Label**)、提示框 (**Tooltip**) 等



- ⑤ 还可以把用于分类的类别变量 (类别不宜太多) 设置为 **SplitBy**，从而把一个散点图拆分为多个小散点图 (smallmultiple)；散点可以以个股为单位 (不汇总)，也可以按行业等 (industry) 汇总。

