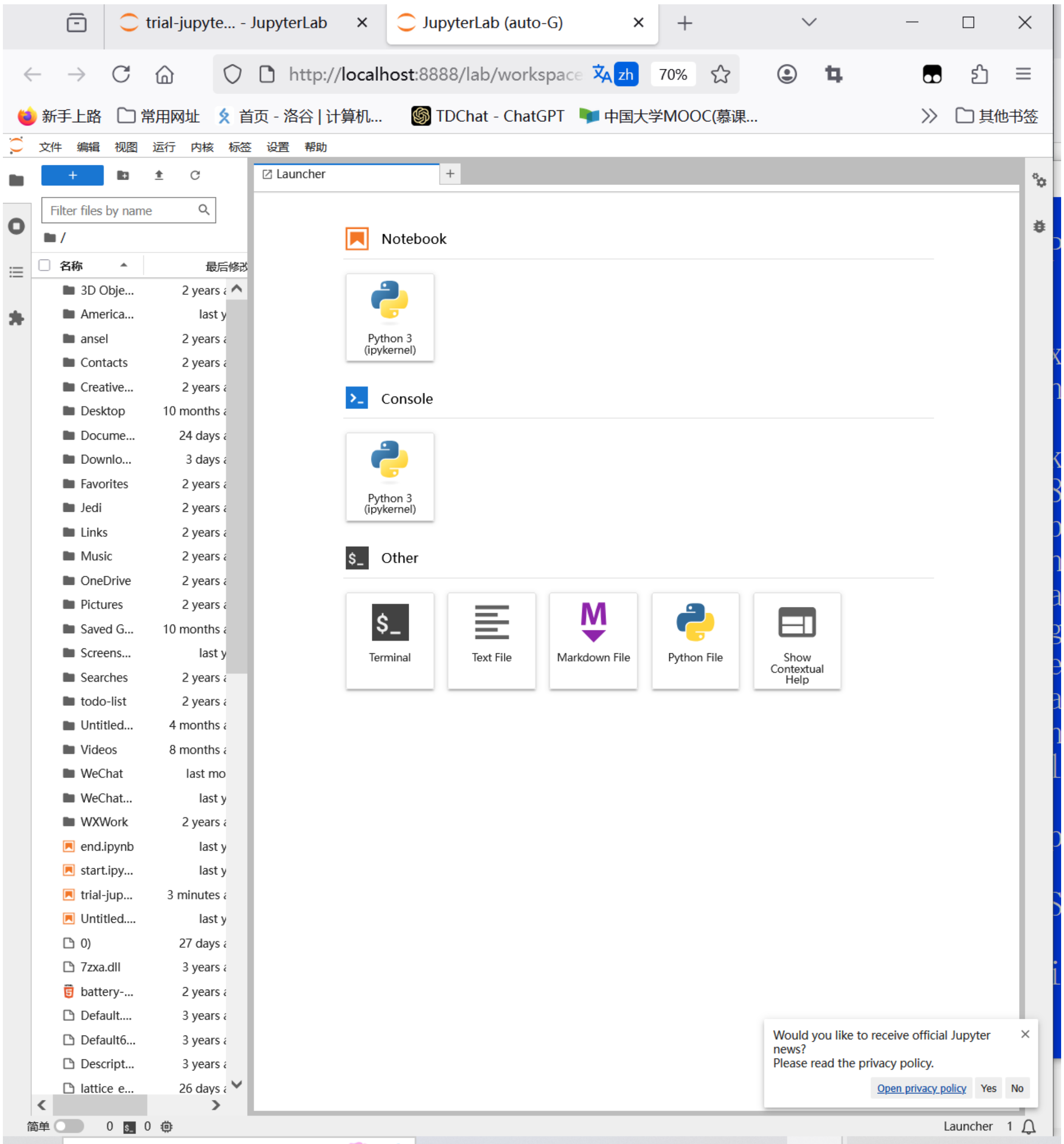


名称	最后修改
3D Obje...	2 years
America...	last y
ansel	2 years
Contacts	2 years
Creative...	2 years
Desktop	10 months
Docume...	24 days
Downlo...	3 days
Favorites	2 years
Jedi	2 years
Links	2 years
Music	2 years
OneDrive	2 years
Pictures	2 years
Saved G...	10 months
Screens...	last y
Searches	2 years
todo-list	2 years
Untitled...	4 months
Videos	8 months
WeChat	last mo
WeChat...	last y
WXWork	2 years
end.ipynb	last y
start.ipy...	last y
trial-jup...	3 minutes
Untitled....	last y
0)	27 days
7zxa.dll	3 years
battery-...	2 years
Default....	3 years
Default6...	3 years
Descript...	3 years
lattice e...	26 days

Would you like to receive official Jupyter news?
Please read the privacy policy.

Open privacy policy Yes No

快捷指令无运行痕迹，盖复制指令作为记录

在单元格（Cell）里编写 Python 代码，按 Shift+Enter 运行 Cell 并下移

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）

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，按 a 在上方添加，按 b 在下方添加，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

在单元格（Cell）的编写模式下，按 Ctrl+Shift+- 切分单元格

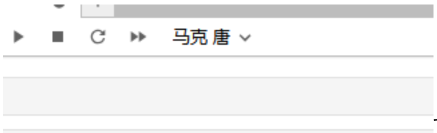
按按钮显示/隐藏 Minimap

运行单元格 (Cell) 注意序号单调递增

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

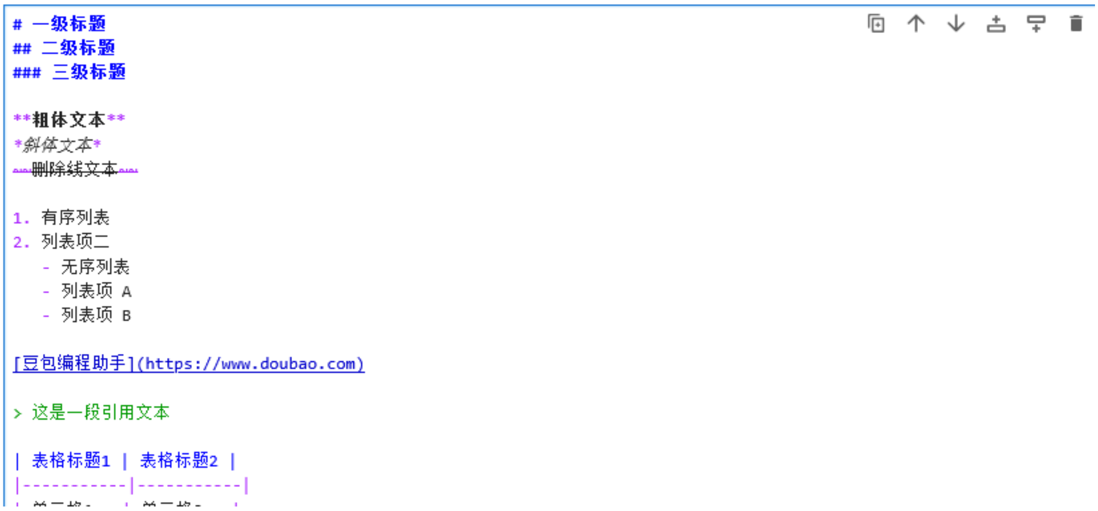
单元格 (Cell) 序号为 * 表示代码运行中, 尚未返回, 按 ii 可以打断 (KeyboardInterrupt) (类似于终端的 Ctrl+C)

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), 重启后需要从上至下重新运行一遍代码 (Shift+Enter), 运行前建议先在菜单里选择 "Edit / Clear Outputs of All Cells" 清空全部页面显示的输出

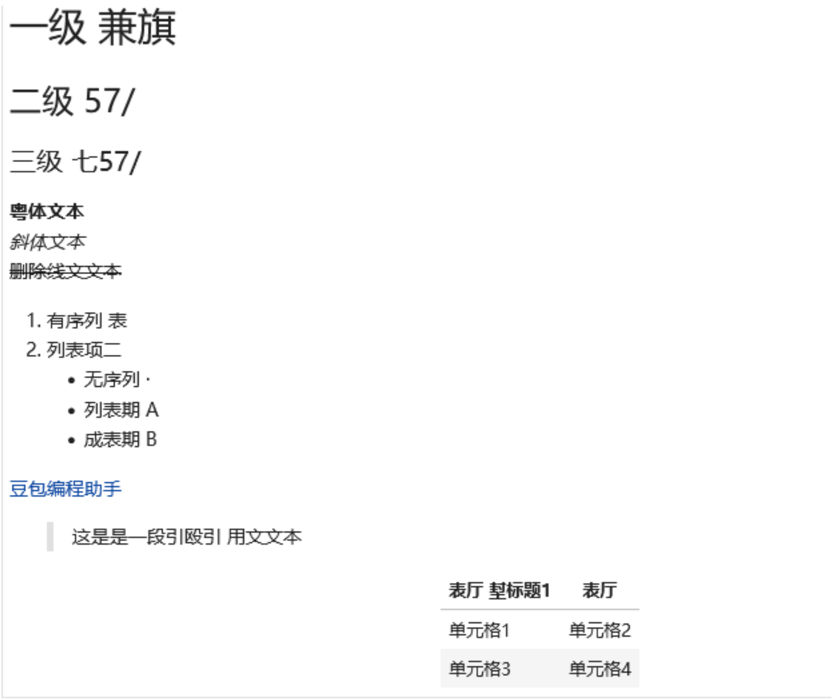


切换至 markdown 模式

Markdown



```
# 一级标题
## 二级标题
### 三级标题

**粗体文本**
*斜体文本*
~~删除线文本~~

1. 有序列表
2. 列表项二
   - 无序列表
   - 列表项 A
   - 列表项 B

[豆包编程助手](https://www.doubao.com)

> 这是一段引用文本

| 表格标题1 | 表格标题2 |
|-----------|-----------|
```



一级 兼旗

二级 57/

三级 七57/

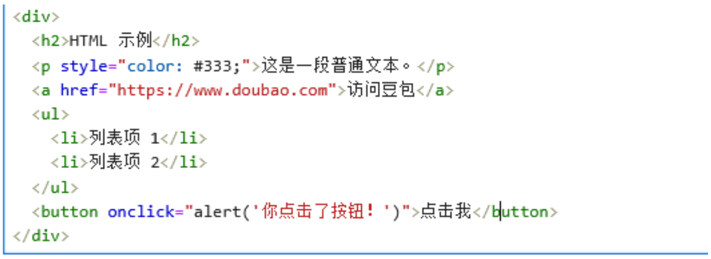
粤体文本
斜体文本
~~删除线文文本~~

1. 有序列 表
2. 列表项二
    - 无序列 ·
    - 列表期 A
    - 成表期 B

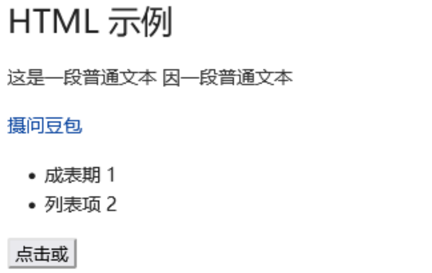
豆包编程助手

> 这是是一段引殴引 用文文本

表厅 埜标题1	表厅
单元格1	单元格2
单元格3	单元格4



```html
<div>
  <h2>HTML 示例</h2>
  <p style="color: #333;">这是一段普通文本。</p>
  <a href="https://www.doubao.com">访问豆包</a>
  <ul>
    <li>列表项 1</li>
    <li>列表项 2</li>
  </ul>
  <button onclick="alert('你点击了按钮！')">点击我</button>
</div>
```



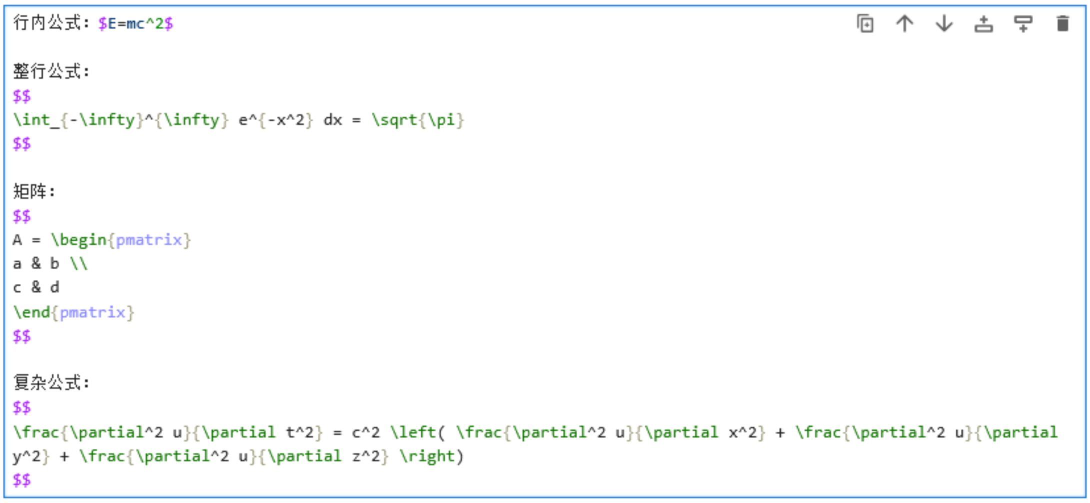
HTML 示例

这是一段普通文本 因一段普通文本

摄问豆包

- 成表期 1
- 列表项 2

点击或



```
行内公式：$E=mc^2$

整行公式：
$$
\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}
$$

矩阵：
$$
A = \begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
$$

复杂公式：
$$
\frac{\partial^2 u}{\partial t^2} = c^2 \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial
y^2} + \frac{\partial^2 u}{\partial z^2} \right)
$$
```

行内序:$E = mc^2$

整行公式:

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$$

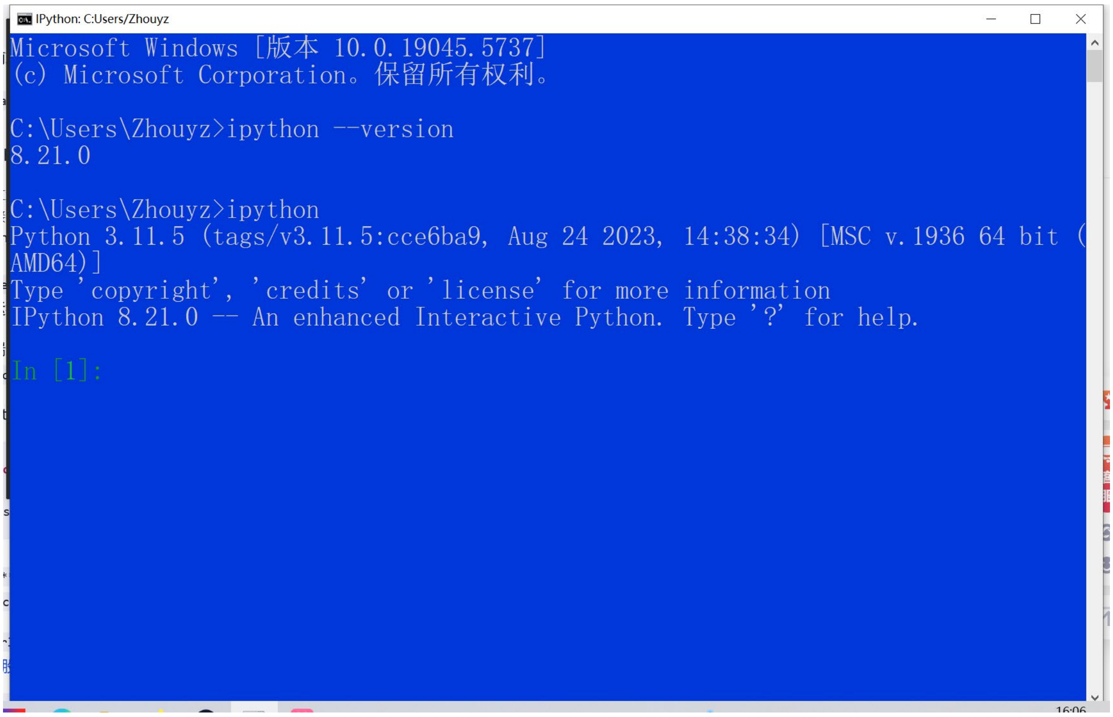
矩阵:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

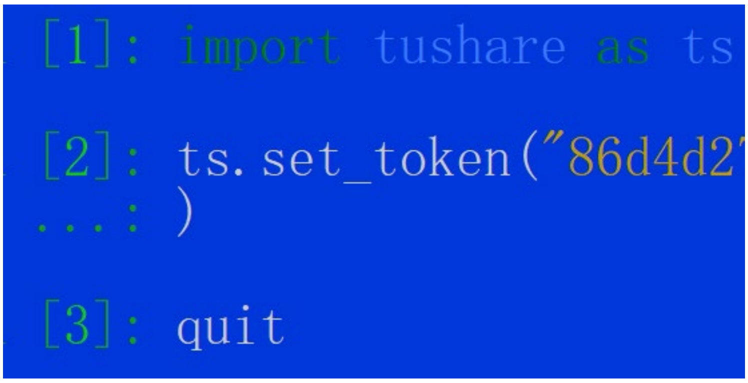
复杂公式:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right)$$

Ipython 调用



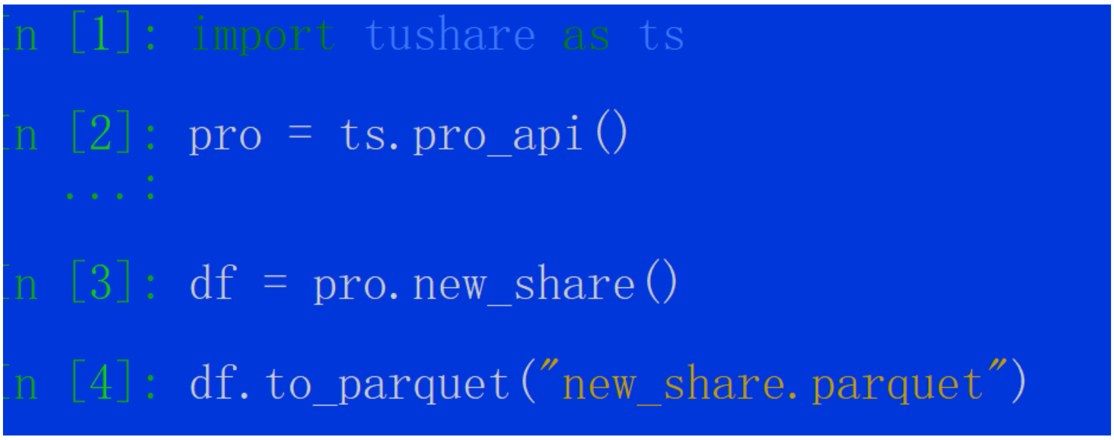
```
Microsoft Windows [版本 10.0.19045.5737]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Zhouyz>ipython --version
8.21.0

C:\Users\Zhouyz>ipython
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (
AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.21.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```



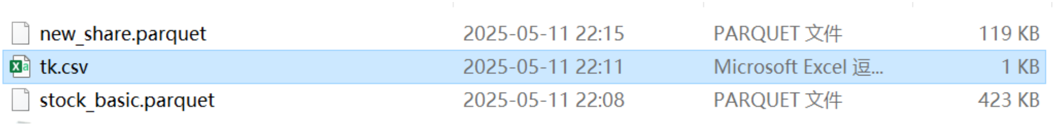
```
[1]: import tushare as ts

[2]: ts.set_token("86d4d2
...: )

[3]: quit
```

储存 tk

调用数据

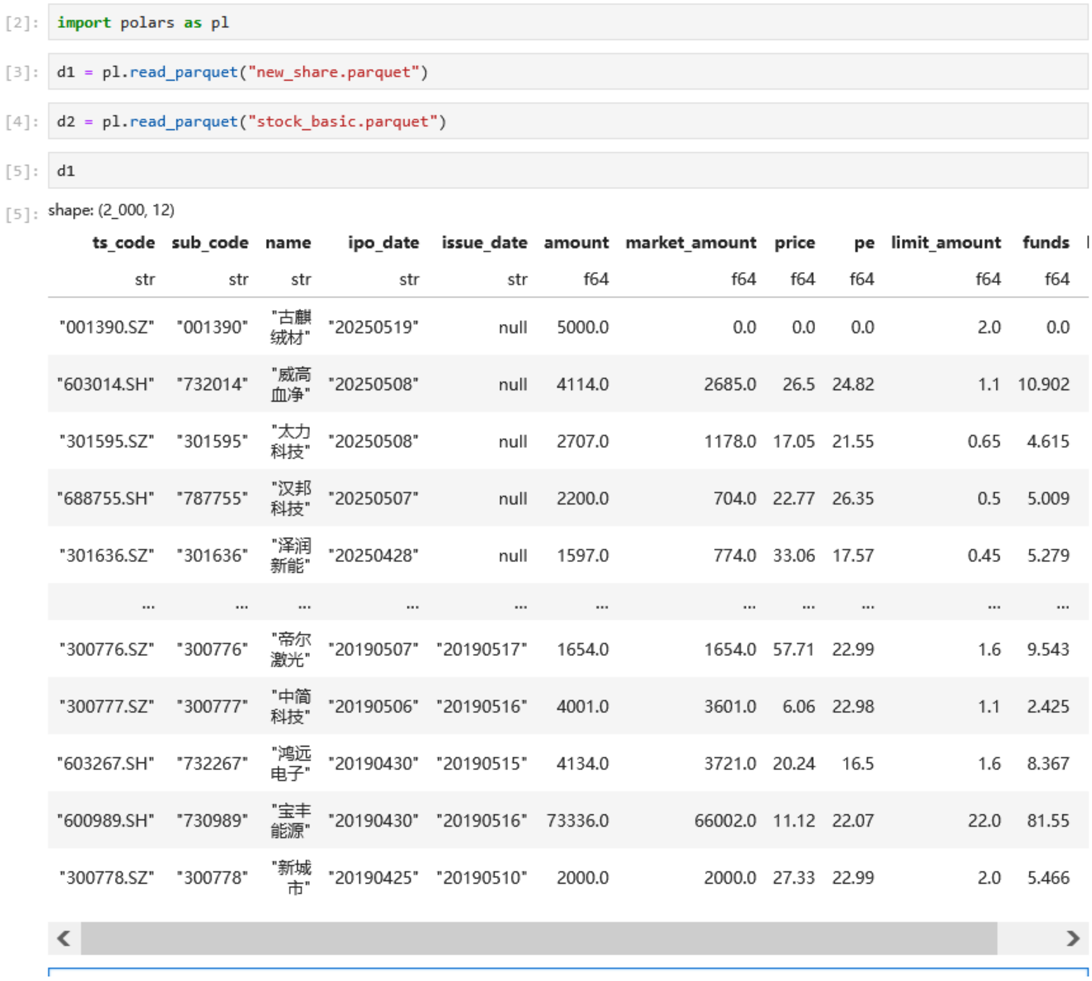


```
n [1]: import tushare as ts

n [2]: pro = ts.pro_api()
   ...:

n [3]: df = pro.new_share()

n [4]: df.to_parquet("new_share.parquet")
```



new_share.parquet	2025-05-11 22:15	PARQUET 文件	119 KB
tk.csv	2025-05-11 22:11	Microsoft Excel 逗...	1 KB
stock_basic.parquet	2025-05-11 22:08	PARQUET 文件	423 KB

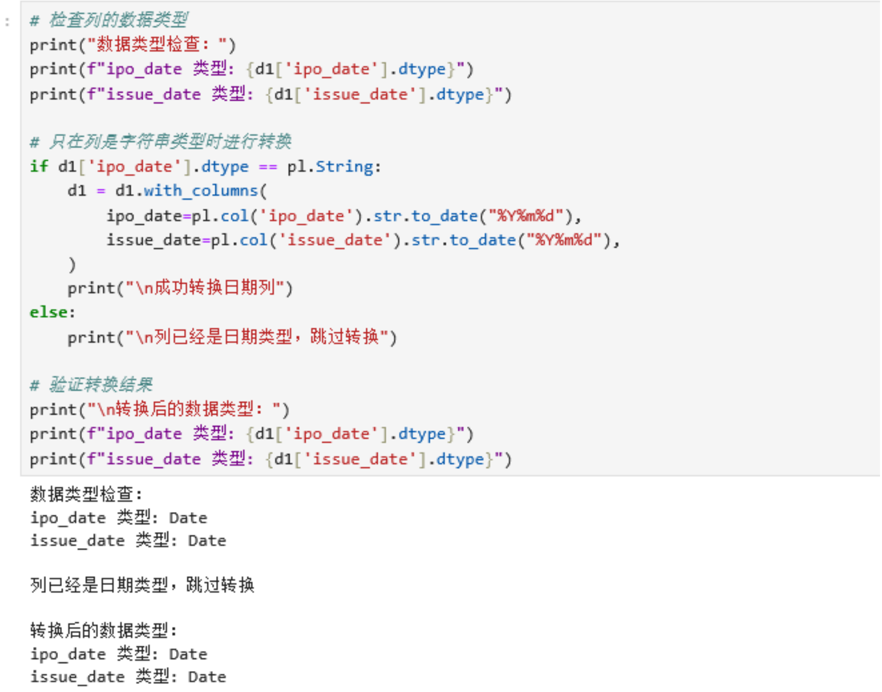
提取数据



```
[2]: import polars as pl

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

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

[5]: d1

[5]: shape: (2_000, 12)
```

ts_code str	sub_code str	name str	ipo_date str	issue_date str	amount f64	market_amount f64	price f64	pe f64	limit_amount f64	funds f64
"001390.SZ"	"001390"	"古麒绒材"	"20250519"	null	5000.0	0.0	0.0	0.0	2.0	0.0
"603014.SH"	"732014"	"威高血净"	"20250508"	null	4114.0	2685.0	26.5	24.82	1.1	10.902
"301595.SZ"	"301595"	"太力科技"	"20250508"	null	2707.0	1178.0	17.05	21.55	0.65	4.615
"688755.SH"	"787755"	"汉邦科技"	"20250507"	null	2200.0	704.0	22.77	26.35	0.5	5.009
"301636.SZ"	"301636"	"泽润新能"	"20250428"	null	1597.0	774.0	33.06	17.57	0.45	5.279
…	…	…	…	…	…	…	…	…	…	…
"300776.SZ"	"300776"	"帝尔激光"	"20190507"	"20190517"	1654.0	1654.0	57.71	22.99	1.6	9.543
"300777.SZ"	"300777"	"中简科技"	"20190506"	"20190516"	4001.0	3601.0	6.06	22.98	1.1	2.425
"603267.SH"	"732267"	"鸿远电子"	"20190430"	"20190515"	4134.0	3721.0	20.24	16.5	1.6	8.367
"600989.SH"	"730989"	"宝丰能源"	"20190430"	"20190516"	73336.0	66002.0	11.12	22.07	22.0	81.55
"300778.SZ"	"300778"	"新城市"	"20190425"	"20190510"	2000.0	2000.0	27.33	22.99	2.0	5.466

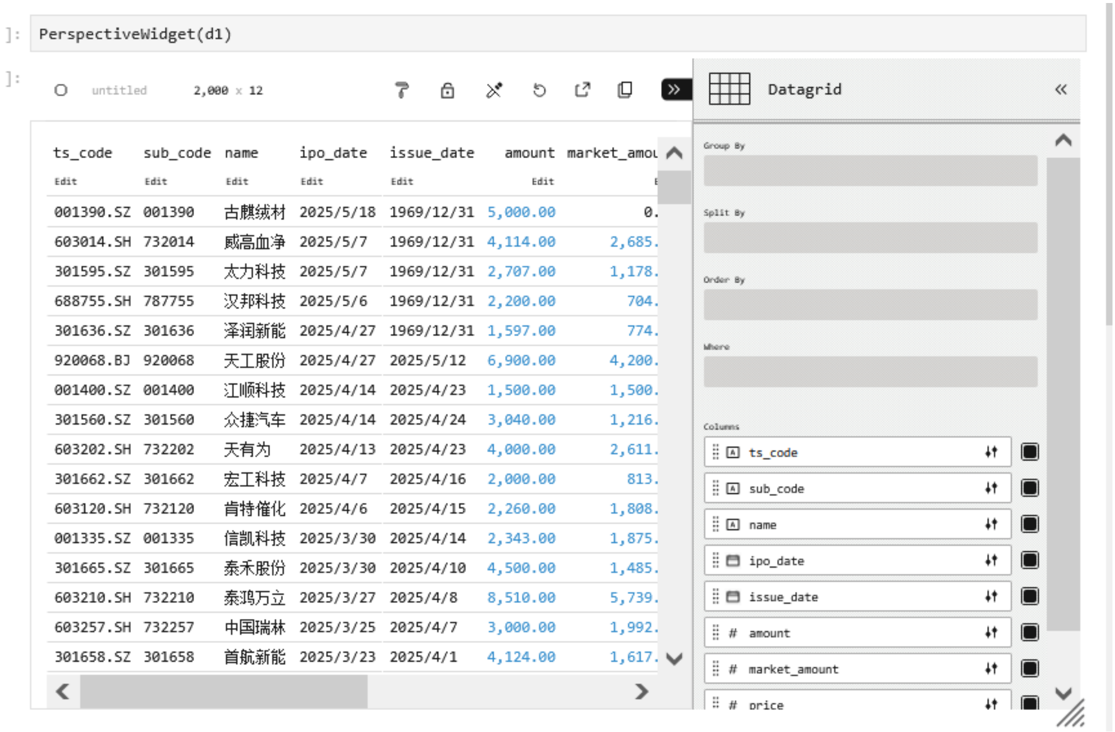
日期数据处理及结果



```
# 检查列的数据类型
print("数据类型检查：")
print(f"ipo_date 类型: {d1['ipo_date'].dtype}")
print(f"issue_date 类型: {d1['issue_date'].dtype}")

# 只在列是字符串类型时进行转换
if d1['ipo_date'].dtype == pl.String:
    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"),
    )
    print("\n成功转换日期列")
else:
    print("\n列已经是日期类型，跳过转换")

# 验证转换结果
print("\n转换后的数据类型：")
print(f"ipo_date 类型: {d1['ipo_date'].dtype}")
print(f"issue_date 类型: {d1['issue_date'].dtype}")
```

```
数据类型检查：
ipo_date 类型: Date
issue_date 类型: Date

列已经是日期类型，跳过转换

转换后的数据类型：
ipo_date 类型: Date
issue_date 类型: Date
```

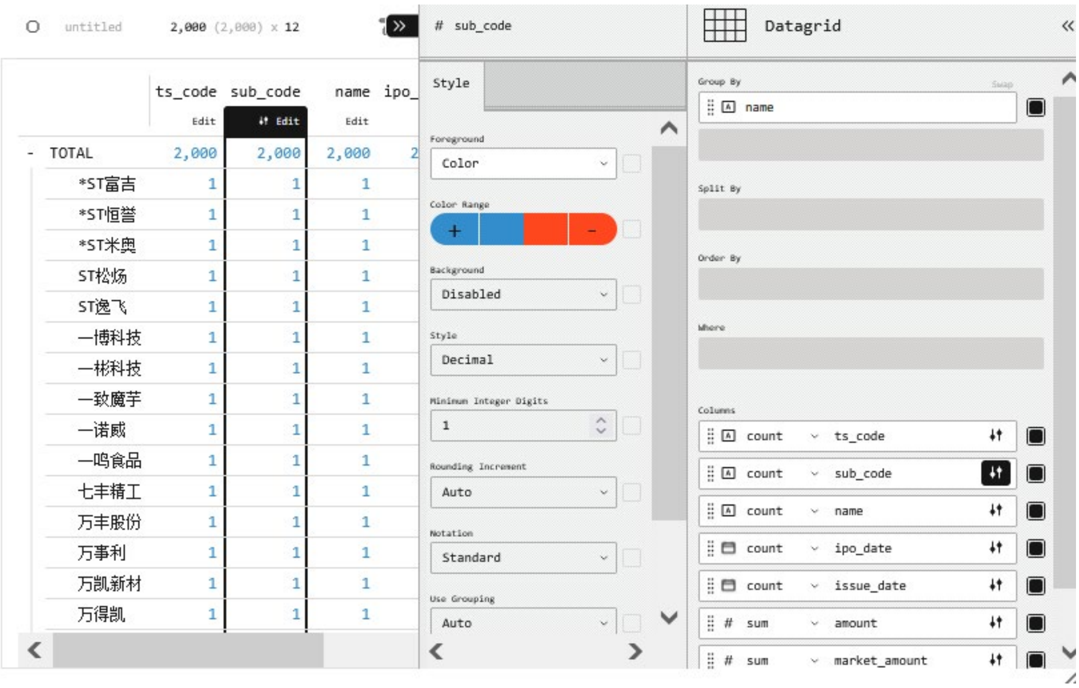
PerspectiveWidget 调用



```
PerspectiveWidget(d1)
```

ts_code Edit	sub_code Edit	name Edit	ipo_date Edit	issue_date Edit	amount Edit	market_amou
001390.SZ	001390	古麒绒材	2025/5/18	1969/12/31	5,000.00	0.
603014.SH	732014	威高血净	2025/5/7	1969/12/31	4,114.00	2,685.
301595.SZ	301595	太力科技	2025/5/7	1969/12/31	2,707.00	1,178.
688755.SH	787755	汉邦科技	2025/5/6	1969/12/31	2,200.00	704.
301636.SZ	301636	泽润新能	2025/4/27	1969/12/31	1,597.00	774.
920068.BJ	920068	天工股份	2025/4/27	2025/5/12	6,900.00	4,200.
001400.SZ	001400	江顺科技	2025/4/14	2025/4/23	1,500.00	1,500.
301560.SZ	301560	众捷汽车	2025/4/14	2025/4/24	3,040.00	1,216.
603202.SH	732202	天有为	2025/4/13	2025/4/23	4,000.00	2,611.
301662.SZ	301662	宏工科技	2025/4/7	2025/4/16	2,000.00	813.
603120.SH	732120	肯特催化	2025/4/6	2025/4/15	2,260.00	1,808.
001335.SZ	001335	信凯科技	2025/3/30	2025/4/14	2,343.00	1,875.
301665.SZ	301665	泰禾股份	2025/3/30	2025/4/10	4,500.00	1,485.
603210.SH	732210	泰鸿万立	2025/3/27	2025/4/8	8,510.00	5,739.
603257.SH	732257	中国瑞林	2025/3/25	2025/4/7	3,000.00	1,992.
301658.SZ	301658	首航新能	2025/3/23	2025/4/1	4,124.00	1,617.

基础调试以 groupby 为例子



untitled 2,000 (2,000) x 12

	ts_code Edit	sub_code Edit	name Edit	ipo_
- TOTAL	2,000	2,000	2,000	2
*ST富吉	1	1	1	
*ST恒誉	1	1	1	
*ST米奥	1	1	1	
ST松炀	1	1	1	
ST逸飞	1	1	1	
一博科技	1	1	1	
一彬科技	1	1	1	
一致魔芋	1	1	1	
一诺威	1	1	1	
一鸣食品	1	1	1	
七丰精工	1	1	1	
万丰股份	1	1	1	
万事利	1	1	1	
万凯新材	1	1	1	
万得凯	1	1	1	

sub_code

Style

Foreground: Color

Color Range: + -

Background: Disabled

Style: Decimal

Minimum Integer Digits: 1

Rounding Increment: Auto

Notation: Standard

Use Grouping: Auto

Datagrid

Group By: name

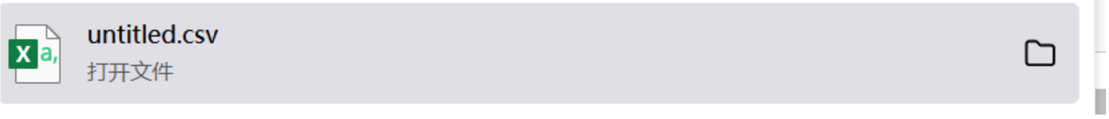
Split By

Order By

Where

Columns:
- count ts_code
- count sub_code
- count name
- count ipo_date
- count issue_date
- sum amount
- sum market_amount

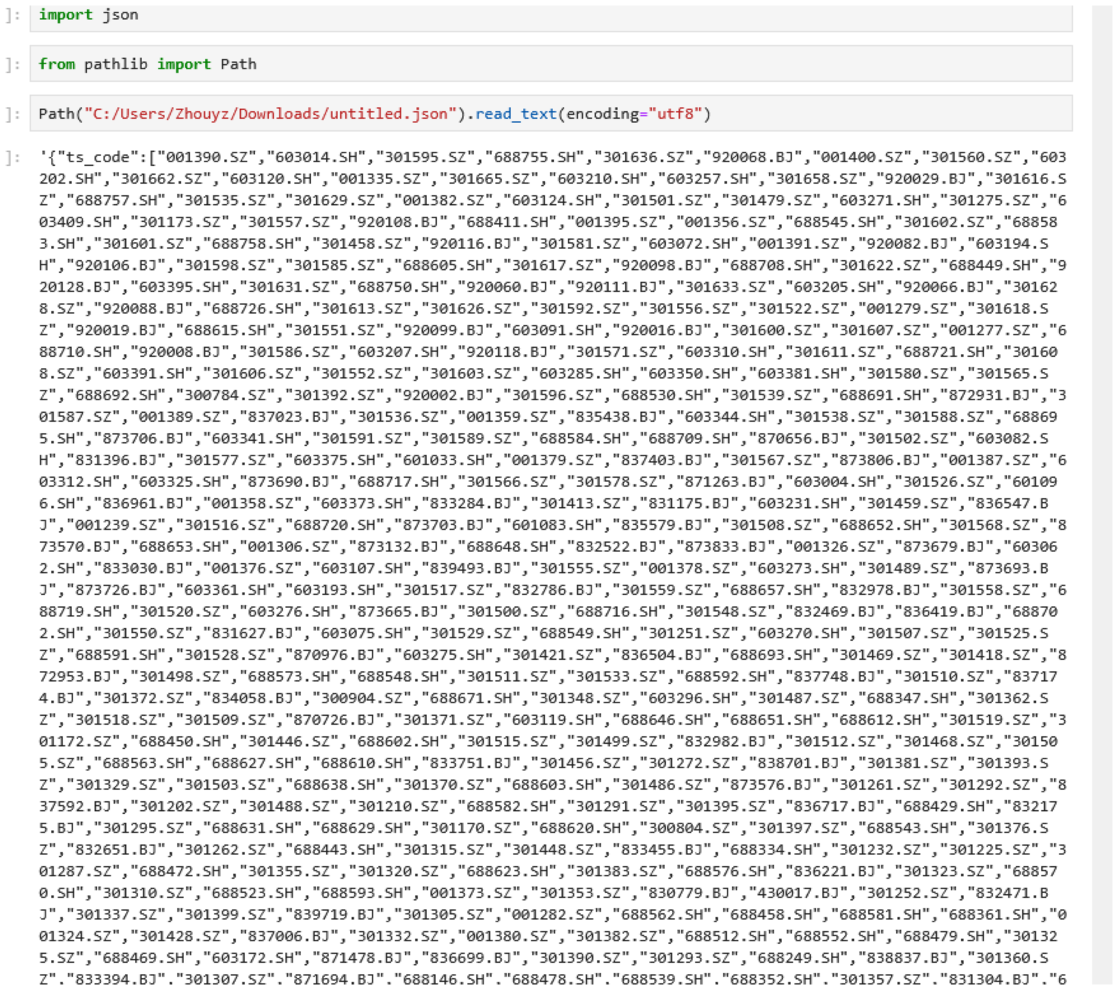
文件导出



untitled.csv

打开文件

读取 json 文件



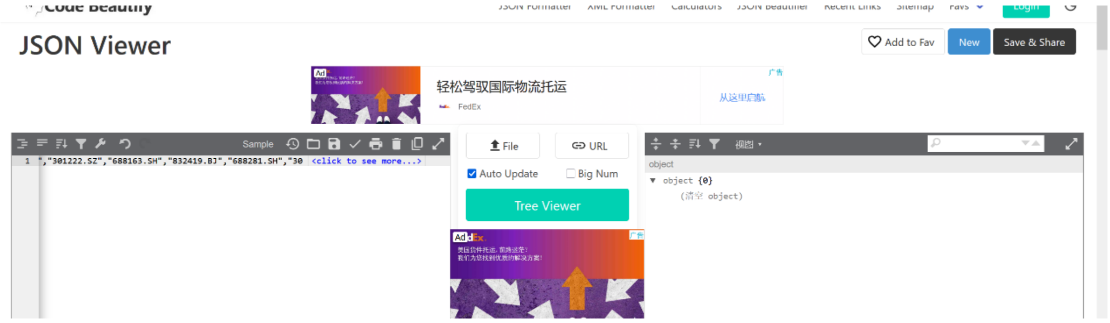
```
]: import json
```

```
]: from pathlib import Path
```

```
]: Path("C:/Users/Zhouyz/Downloads/untitled.json").read_text(encoding="utf8")
```

```
]: '{"ts_code":["001390.SZ","603014.SH","301595.SZ","688755.SH","301636.SZ","920068.BJ","001400.SZ","301560.SZ","603
   202.SH","301662.SZ","603120.SH","001335.SZ","301665.SZ","603210.SH","603257.SH","301658.SZ","920029.BJ","301616.S
   Z","688757.SH","301535.SZ","301629.SZ","001382.SZ","603124.SH","301501.SZ","301479.SZ","603271.SH","301275.SZ","6
   03409.SH","301173.SZ","301557.SZ","920108.BJ","688411.SH","001395.SZ","001356.SZ","688545.SH","301602.SZ","68858
   3.SH","301601.SZ","688758.SH","301458.SZ","920116.BJ","301581.SZ","603072.SH","001391.SZ","920082.BJ","603194.S
   H","920106.BJ","301598.SZ","301585.SZ","688605.SH","301617.SZ","920098.BJ","688708.SH","301622.SZ","688449.SH","9
   20128.BJ","603395.SH","301631.SZ","688750.SH","920060.BJ","920111.BJ","301633.SZ","603205.SH","920066.BJ","30162
   8.SZ","920088.BJ","688726.SH","301613.SZ","301626.SZ","301592.SZ","301556.SZ","301522.SZ","001279.SZ","301618.S
   Z","920019.BJ","688615.SH","301551.SZ","920099.BJ","603091.SH","920016.BJ","301600.SZ","301607.SZ","001277.SZ","6
   88710.SH","920008.BJ","301586.SZ","603207.SH","920118.BJ","301571.SZ","603310.SH","301611.SZ","688721.SH","30160
   8.SZ","603391.SH","301606.SZ","301552.SZ","301603.SZ","603285.SH","603350.SH","603381.SH","301580.SZ","301565.S
   Z","688692.SH","300784.SZ","301392.SZ","920002.BJ","301596.SZ","688530.SH","301539.SZ","688691.SH","872931.BJ","3
   01587.SZ","001389.SZ","837023.BJ","301536.SZ","001359.SZ","835438.BJ","603344.SH","301538.SZ","301588.SZ","68869
   5.SH","873706.BJ","603341.SH","301591.SZ","301589.SZ","688584.SH","688709.SH","870656.BJ","301502.SZ","603082.S
   H","831396.BJ","301577.SZ","603375.SH","601033.SH","001379.SZ","837403.BJ","301567.SZ","873806.BJ","001387.SZ","6
   03312.SH","603325.SH","873690.BJ","688717.SH","301566.SZ","301578.SZ","871263.BJ","603004.SH","301526.SZ","60109
   6.SH","836961.BJ","001358.SZ","603373.SH","833284.BJ","301413.SZ","831175.BJ","603231.SH","301459.SZ","836547.B
   J","001239.SZ","301516.SZ","688720.SH","873703.BJ","601083.SH","835579.BJ","301508.SZ","688652.SH","301568.SZ","8
   73570.BJ","688653.SH","001306.SZ","873132.BJ","688648.SH","832522.BJ","873833.BJ","001326.SZ","873679.BJ","60306
   2.SH","833030.BJ","001376.SZ","603107.SH","839493.BJ","301555.SZ","001378.SZ","603273.SH","301489.SZ","873693.B
   J","873726.BJ","603361.SH","603193.SH","301517.SZ","832786.BJ","301559.SZ","688657.SH","832978.BJ","301558.SZ","6
   88719.SH","301520.SZ","603276.SH","873665.BJ","301500.SZ","688716.SH","301548.SZ","832469.BJ","836419.BJ","68870
   2.SH","301550.SZ","831627.BJ","603075.SH","301529.SZ","688549.SH","301251.SZ","603270.SH","301507.SZ","301525.S
   Z","688591.SH","301528.SZ","870976.BJ","603275.SH","301421.SZ","836504.BJ","688693.SH","301469.SZ","301418.SZ","8
   72953.BJ","301498.SZ","688573.SH","688548.SH","301511.SZ","301533.SZ","688592.SH","837748.BJ","301510.SZ","83717
   4.BJ","301372.SZ","834058.BJ","300904.SZ","688671.SH","301348.SZ","603296.SH","301487.SZ","688347.SH","301362.S
   Z","301518.SZ","301509.SZ","870726.BJ","301371.SZ","603119.SH","688646.SH","688651.SH","688612.SH","301519.SZ","3
   01172.SZ","688450.SH","301446.SZ","688602.SH","301515.SZ","301499.SZ","832982.BJ","301512.SZ","301468.SZ","30150
   5.SZ","688563.SH","688627.SH","688610.SH","833751.BJ","301456.SZ","301272.SZ","838701.BJ","301381.SZ","301393.S
   Z","301329.SZ","301503.SZ","688638.SH","301370.SZ","688603.SH","301486.SZ","873576.BJ","301261.SZ","301292.SZ","8
   37592.BJ","301202.SZ","301488.SZ","301210.SZ","688582.SH","301291.SZ","301395.SZ","836717.BJ","688429.SH","83217
   5.BJ","301295.SZ","688631.SH","688629.SH","301170.SZ","688620.SH","300804.SZ","301397.SZ","688543.SH","301376.S
   Z","832651.BJ","301262.SZ","688443.SH","301315.SZ","301448.SZ","833455.BJ","688334.SH","301232.SZ","301225.SZ","3
   01287.SZ","688472.SH","301355.SZ","301320.SZ","688623.SH","301383.SZ","688576.SH","836221.BJ","301323.SZ","68857
   0.SH","301310.SZ","688523.SH","688593.SH","001373.SZ","301353.SZ","830779.BJ","430017.BJ","301252.SZ","832471.B
   J","301337.SZ","301399.SZ","839719.BJ","301305.SZ","001282.SZ","688562.SH","688458.SH","688581.SH","688361.SH","0
   01324.SZ","301428.SZ","837006.BJ","301332.SZ","001380.SZ","301382.SZ","688512.SH","688552.SH","688479.SH","30132
   5.SZ","688469.SH","603172.SH","871478.BJ","836699.BJ","301390.SZ","301293.SZ","688249.SH","838837.BJ","301360.S
   Z"."833394.BJ"."301307.SZ"."871694.BJ"."688146.SH"."688478.SH"."688539.SH"."688352.SH"."301357.SZ"."831304.BJ"."6
```

https://codebeautify.org/jsonviewer 美化网站



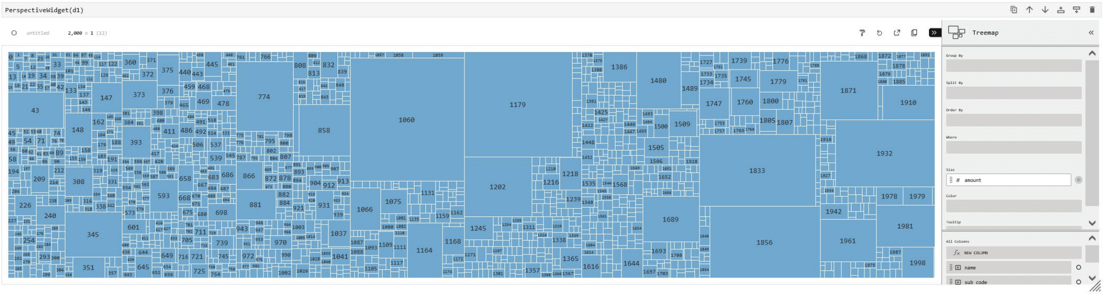
JSON Viewer

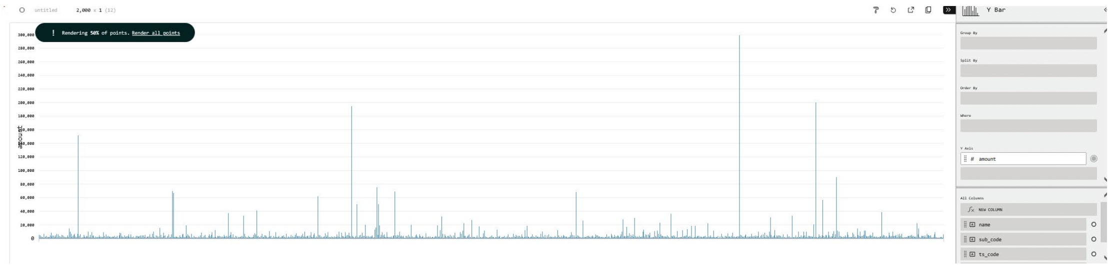
```
1 ","301222.SZ","688163.SH","832419.BJ","688281.SH","30 <click to see more...>
```

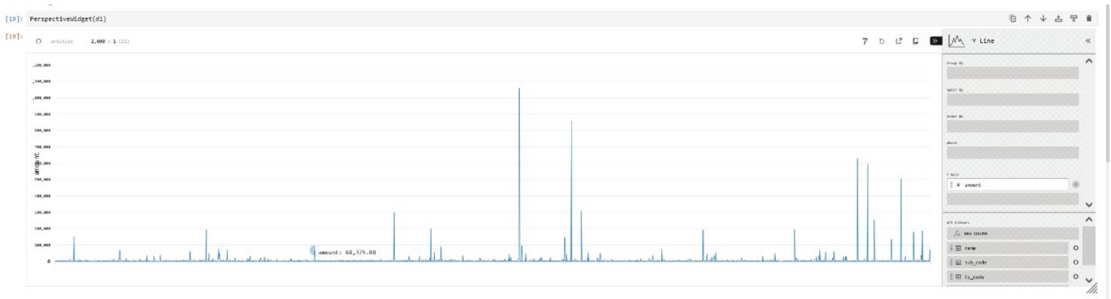
☑ Auto Update ☐ Big Num

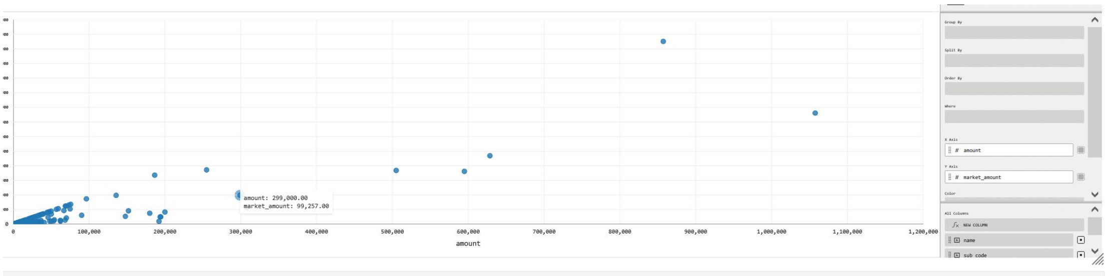
```
object
▼ object {0}
    (清空 object)
```

数据可视化以 amount 为例：









Treemap（树形结构图）

图表特点：通过不同大小的矩形展示数据的分类占比构成情况，颜色可用于体现第二个维度的数据（文本或数值）。不同矩形的面积大小直观反映了数据在各分类中的占比，颜色的差异帮助区分不同类别或体现其他维度的信息。

学习到的内容：适用于展示数据的层次结构和比例关系，能够快速了解各部分在整体中所占的比重。在处理具有分类层次的数据时非常有效，比如组织结构、产品类别分布等数据。

Y Bar（条形图 / 柱状图）

图表特点：横轴为第一个维度，通过 Group By 控制，纵轴（条形的高度）为第二个维度，支持多变量并列显示。可以清晰地比较不同类别之间的数值大小，通过条形的

长度直观展示数据的差异。还可以用于绘制直方图，对连续型数值变量进行分桶处理后展示其分布情况。

学习到的内容：适用于比较不同类别之间的数据差异，无论是离散型数据还是经过分桶处理的连续型数据。在统计分析中，直方图对于了解数据的分布特征非常有帮助，比如数据的集中趋势、离散程度等。

Y Line（折线图）

图表特点：常用于绘制时间序列数据，横轴通常是时间，用 Group By 控制，纵轴为连续型数值变量（经过汇总），支持多序列同时显示。通过折线的走势可以清晰地看到数据随时间的变化趋势，多序列的展示便于比较不同数据系列之间的差异。

学习到的内容：在分析时间序列数据时非常有用，能够直观地展示数据的变化情况，帮助发现数据的趋势、周期性等特征。结合实际财经事件，可以更好地理解数据变化背后的原因，增强对现实背景的理解。

X/Y Scatter（散点图）

图表特点：用于观察两个数值型连续变量之间的相关关系，通过 Group By 进行数据分组汇总，每个组对应一个散点，散点的坐标由 X Axis 和 Y Axis 对应的数值确定。散点的分布情况可以反映变量之间的相关性，如正相关、负相关或无相关性。还可以通过颜色、大小、符号等进一步体现更多的变量维度，或按分类变量进行拆分，展示更细致的规律。

学习到的内容：有助于发现变量之间的潜在关系，判断数据是否具有相关性以及相关性的强弱和方向。在处理数值型数据时，通过散点图可以初步探索数据的特征，为进一步的数据分析提供线索。同时，注意散点分布的均匀性，对于异常数据或变量单位问题可以通过变换等方式进行调整。