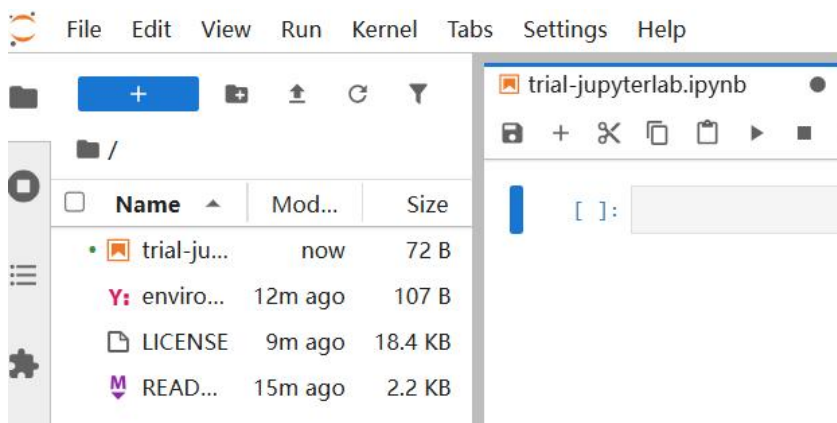
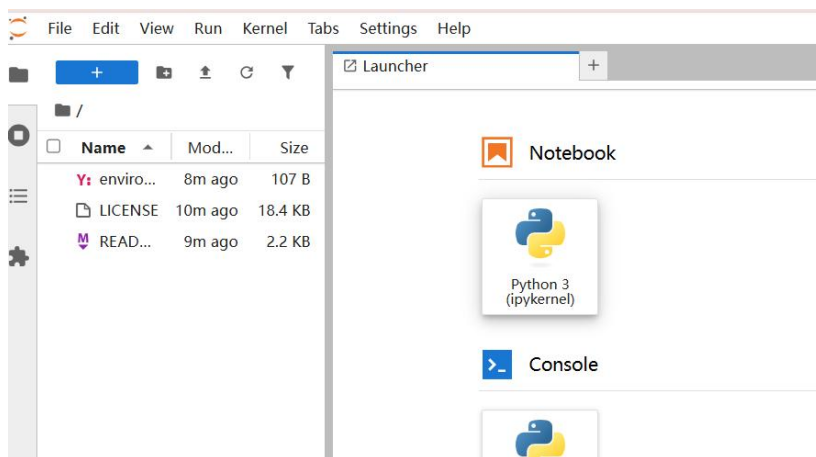
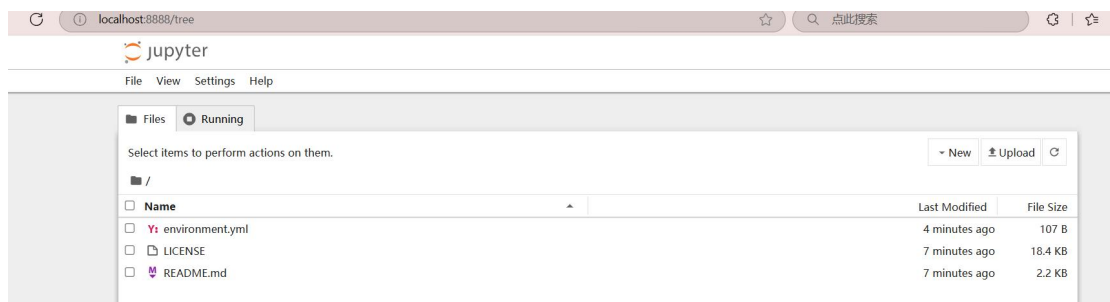
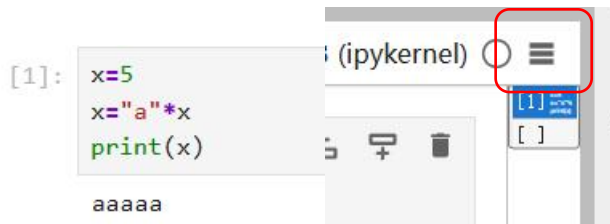


```
! environment.yml U X
! environment.yml
1  name: week07
2  channels:
3    - conda-forge
4  dependencies:
5    - python=3.12
6    - wat-inspector
7    - jupyterlab
```





- 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

一级标题

二级标题

三级标题

这是一段普通文本，**这是粗体文本**，*这是斜体文本*，***这是粗斜体文本***

这是一个链接：[豆瓣编程助手](#)

这是一个列表：

- 项目1
- 项目2
 - 子项目2.1
 - 子项目2.2
- 项目3

这是一个有序列表：

1. 第一步
2. 第二步
3. 第三步

这是一个代码块：

```
def hello_world():
    print("Hello, World!")
```

简易网页示例

[首页](#) | [产品](#) | [关于](#) | [联系](#)

欢迎访问

这是一个示例页面，展示了基本的 HTML 元素。

特性列表

- 响应式设计
- 语义化结构
- 无障碍支持

产品展示

产品一



$$\int_a^b f(x) dx = F(b) - F(a)$$

行内公式示例: $\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0}$

二元二次方程的求根公式:

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

矩阵乘法示例:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

概率密度函数的性质:

$$\int_{-\infty}^{\infty} f(x) dx = 1 \quad \text{且} \quad f(x) \geq 0 \quad \forall x \in \mathbb{R}$$

```
[I 2025-05-10 22:11:22.007 ServerApp] Starting buffering for 0fab0a02-71
1-8b281195fbbb
[I 2025-05-10 22:11:28.544 ServerApp] Interrupted...
[IPKernelApp] WARNING | Parent appears to have exited, shutting down.
(week07)
DELL@DESKTOP-8NOG904 MINGW64 ~/repo/week07 (main)
$
```

```
-rw-r--r-- 1 DELL 197121 65 5月 10 23:08 tk.csv
lrwxrwxrwx 1 DELL 197121 0 5月 10 23:05 token/
```

```
(base) DELL@DESKTOP-8NOG904
$ cat tk.csv
token
```

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

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

In [3]: type(pro)
Out[3]: tushare.pro.client.DataApi

In [4]: pro
Out[4]: <tushare.pro.client.DataApi at 0x2f46ae3cf20>

In [5]: pro.new_share()
Out[5]:
   ts_code sub_code  name ipo_date issue_date  amount
0  001390.SZ   001390  古麒绒材  20250519      None    500
.00
1  603014.SH   732014  威高血净  20250508      None    411
```

```
In [5]: de.to_parquet("new_share.parquet")
```

```
In [7]: df = pro.stock_basic()
```

```
In [9]: df.columns
Out[9]:
Index(['ts_code', 'symbol', 'name', 'area', 'industry', 'cnsPELL', 'market',
      'list_date', 'act_name', 'act_ent_type'],
      dtype='object')
```

```
In [10]: df = pro.stock_basic(fields="ts_code,symbol,name,area")
```

```
(base) DELL@DESKTOP-8NOG904 MINGW64 ~/repo/week07 (main)
$ ls -l
total 573
-rw-r--r-- 1 DELL 197121 156 5月 10 23:39 environment.yml
-rw-r--r-- 1 DELL 197121 18805 5月 10 19:08 LICENSE
-rw-r--r-- 1 DELL 197121 120102 5月 10 23:44 new_share.parquet
-rw-r--r-- 1 DELL 197121 2239 5月 10 19:08 README.md
-rw-r--r-- 1 DELL 197121 429181 5月 11 00:14 stock_basic.parquet
-rw-r--r-- 1 DELL 197121 4731 5月 10 22:10 trial-jupyterlab.ipynb
```

豆包

新对话

AI 搜索

帮我写作

AI 编程

图像生成 超能创意 1.0

更多

AI 云盘

历史对话

比较 Parquet 和 CSV 格式

生成示例 Markdown 代码

Python -m 用法

询问原因

Unix 中文文件操作与 Git 状态不...

介绍自身功能

下载电脑版

比较 Parquet 和 CSV 格式

初步了解 Parquet 格式和 CSV 格式的特点和适用领域

Parquet 格式和 CSV 格式是两种常见的数据存储格式。它们在特点和适用领域上有所不同。以下是它们的初步介绍：

特点

Parquet 格式

- 列式存储**：数据按列进行存储，而不是像传统的行式存储那样按行存储。这种方式有利于在查询时只读取需要的列，减少 I/O 开销，提高查询性能。
- 压缩效率高**：支持多种压缩算法，如 Snappy、Gzip 等，可以有效地减少数据存储空间，降低存储成本。
- 支持复杂数据类型**：能够很好地处理嵌套结构的数据，如数组、字典、结构体等，适合存储半结构化或结构化程度较高的数据。
- 数据查询快**：由于列式存储和压缩的特点，在进行特定列的查询、聚合等操作时，速度通常比 CSV 格式快很多。
- 与大数据生态系统集成性好**：Parquet 格式与 Hadoop、Pig、Spark 等大数据处理框架紧密集成，被广泛应用于大数据分析场景中。

发消息、输入 @ 选择技能或 / 选择文件

深度思考 技能

trial-perspective.ipynb

environment.yml

trial-jupyterlab.ipynb

Python 3 (ipykernel)

[1]: import polars as pl

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

[x]: 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
"001390.SZ"	"001390"	"古麒域材"	"20250519"	null	5000.0	0.0	0.0	0.0	2.0	0.0	0.0
"603014.SH"	"732014"	"威高血净"	"20250508"	null	4114.0	2685.0	26.5	24.82	1.1	10.902	0.03
"301595.SZ"	"301595"	"太力科技"	"20250508"	null	2707.0	1178.0	17.05	21.55	0.65	4.615	0.02
"688755.SH"	"787755"	"汉邦科技"	"20250507"	null	2200.0	704.0	22.77	26.35	0.5	5.009	0.03
"301636.SZ"	"301636"	"泽润新能"	"20250428"	null	1597.0	774.0	33.06	17.57	0.45	5.279	0.02
...
"300776.SZ"	"300776"	"帝尔激光"	"20190507"	"20190517"	1654.0	1654.0	57.71	22.99	1.6	9.543	0.01
"300777.SZ"	"300777"	"中简科技"	"20190506"	"20190516"	4001.0	3601.0	6.06	22.98	1.1	2.425	0.04
"603267.SH"	"732267"	"鸿远电子"	"20190430"	"20190515"	4134.0	3721.0	20.24	16.5	1.6	8.367	0.03
"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


```
[8]: 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"),
    )
```

[8]: shape: (2_000, 12)

ts_code	sub_code	name	ipo_date	issue_date
str	str	str	date	date
"001390.SZ"	"001390"	"古麒绒材"	2025-05-19	null
"603014.SH"	"732014"	"威高血净"	2025-05-08	null
"301595.SZ"	"301595"	"太力科技"	2025-05-08	null
"688755.SH"	"787755"	"汉邦科技"	2025-05-07	null
"301636.SZ"	"301636"	"泽润新能"	2025-04-28	null
...
"300776.SZ"	"300776"	"帝尔激光"	2019-05-07	2019-05-17
"300777.SZ"	"300777"	"中简科技"	2019-05-06	2019-05-16
"603267.SH"	"732267"	"鸿远电子"	2019-04-30	2019-05-15
"600989.SH"	"730989"	"宝丰能源"	2019-04-30	2019-05-16
"300778.SZ"	"300778"	"新城市"	2019-04-25	2019-05-10

.6]: PerspectiveWidget(d1)

```
ValueError                                Traceback (most recent call last)
Cell In[16], line 1
----> 1 PerspectiveWidget(d1)

File D:\Anaconda\envs\week07\Lib\site-packages\perspective\widget\widget.py:155
t, binding_mode, **kwargs)
    153 if limit is not None:
    154     self.options.update({"limit": limit})
--> 156 self.load(data, **self.options)

File D:\Anaconda\envs\week07\Lib\site-packages\perspective\widget\widget.py:161
    159 """Load the widget with data."""
    160 # Viewer will ignore **options if `data` is a Table or View.
--> 161 super(PerspectiveWidget, self).load(data, **options)

File D:\Anaconda\envs\week07\Lib\site-packages\perspective\viewer\viewer.py:210, in PerspectiveViewer.load(self, data, **options)
    208 else:
    209     client = perspective.Server().new_local_client()
--> 210 self._table = client.table(data, name=name, **options)
    211 self._client = client
    212 # If the user does not set columns to show, synchronize viewer state
    213 # with dataset.
```

ValueError: Unknown input type TypeId(0xa2ed08c824384bb500e5c2441768fa6a)

存在问题 1: 代码报错

解决方法: 询问豆包, 将数据形式转换为 pandas 形式, 后可打开。

```
import polars as pl
from perspective import PerspectiveWidget

# 假设 d1 是你的 Polars DataFrame
# 先将日期列转换为字符串 (可选)
d1 = d1.with_columns(
    pl.col("ipo_date").cast(pl.Utf8).alias("ipo_date_str")
)

# 转换为 Pandas DataFrame
pandas_df = d1.to_pandas()

# 使用 Pandas DataFrame 创建 PerspectiveWidget
widget = PerspectiveWidget(pandas_df)
widget
```

[15]:

untitled 2,000 × 14

🔍 📄 ✂️ ↶ ↷ 📌 📄

Datagrid

<<

Group By

Split By

Order By

Where

Columns

#

market_amount

↑↓

#

price

↑↓

#

pe

↑↓

#

limit_amount

↑↓

#

funds

↑↓

#

ballot

↑↓

A

ipo_date_str

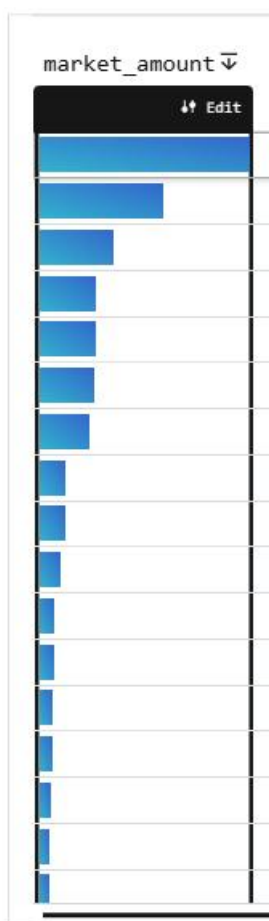
↑↓

A

name

↑↓

market_amount	price	pe	limit_amount	funds	ballot	ipo_date_str	name	sub_code	ts_code	ipo
0.00	0.00	0.00	2.00	0.00	0.00	2025-05-19	古麒绒材	001390	001390.SZ	20
2,685.00	26.50	24.82	1.10	10.90	0.03	2025-05-08	威高血净	732014	603014.SH	20
1,178.00	17.05	21.55	0.65	4.62	0.02	2025-05-08	太力科技	301595	301595.SZ	20
704.00	22.77	26.35	0.50	5.01	0.03	2025-05-07	汉邦科技	787755	688755.SH	20
774.00	33.06	17.57	0.45	5.28	0.02	2025-04-28	泽润新能	301636	301636.SZ	20
4,200.00	3.94	15.19	255.00	2.72	0.04	2025-04-28	天工股份	920068	920068.BJ	20
1,500.00	37.36	15.32	1.50	5.60	0.01	2025-04-15	江顺科技	001400	001400.SZ	20
1,216.00	16.50	21.30	0.70	5.02	0.02	2025-04-15	众捷汽车	301560	301560.SZ	20
2,611.00	93.50	13.50	1.25	37.40	0.03	2025-04-14	天有为	732202	603202.SH	20
813.00	26.60	7.05	0.45	5.32	0.02	2025-04-08	宏工科技	301662	301662.SZ	20
1,808.00	15.00	16.76	0.90	3.39	0.03	2025-04-07	肯特催化	732120	603120.SH	20
1,875.00	12.80	11.44	0.90	3.00	0.02	2025-03-31	信凯科技	001335	001335.SZ	20
1,485.00	10.27	13.40	0.70	4.62	0.02	2025-03-31	泰禾股份	301665	301665.SZ	20
5,739.00	8.60	18.60	2.70	7.32	0.04	2025-03-28	泰鸿万立	732210	603210.SH	20
1,992.00	20.52	18.17	0.95	6.16	0.03	2025-03-26	中国瑞林	732257	603257.SH	20
1,617.00	11.80	15.88	0.75	4.87	0.02	2025-03-24	首航新能	301658	301658.SZ	20
2,677.00	30.38	8.46	158.96	11.69	0.17	2025-03-18	开发科技	920029	920029.BJ	20



2,000 × 14

	index	ts_code	symbol	name	area	industry	fullname	enname	cnsPELL	market	exchange
	Edit	Edit	Edit	Edit	Edit	Edit	Edit	Edit	Edit	Edit	Edit
- TOTAL	14,642,166	5,412	5,412	5,412	5,412	5,412	5,412	5,412	5,412	5,412	5,412
主板	7,417,740	3,180	3,180	3,180	3,180	3,180	3,180	3,180	3,180	3,180	3,180
创业板	3,002,190	1,380	1,380	1,380	1,380	1,380	1,380	1,380	1,380	1,380	1,380
北交所	1,369,630	266	266	266	266	266	266	266	266	266	266
科创板	2,852,606	586	586	586	586	586	586	586	586	586	586

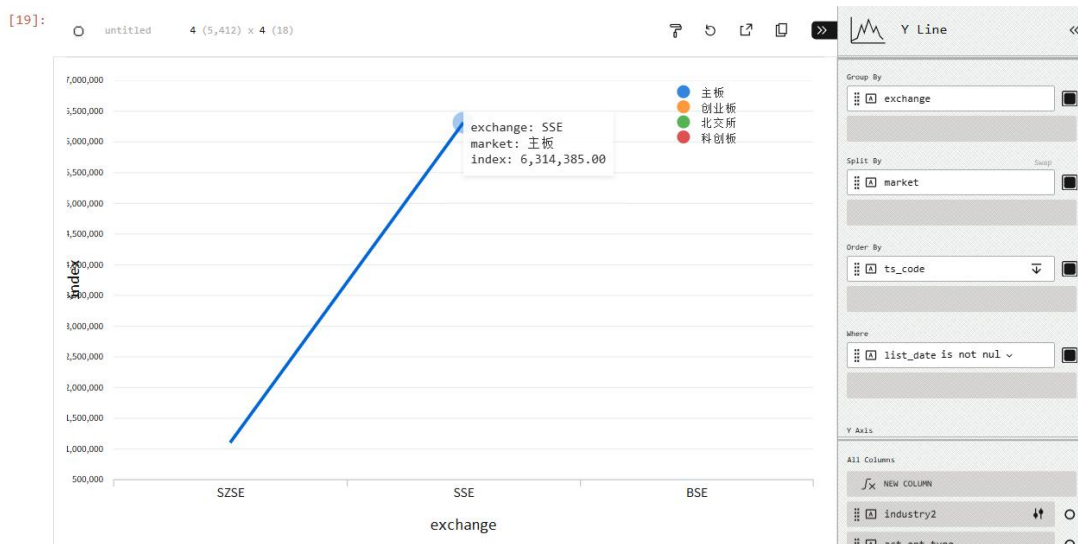
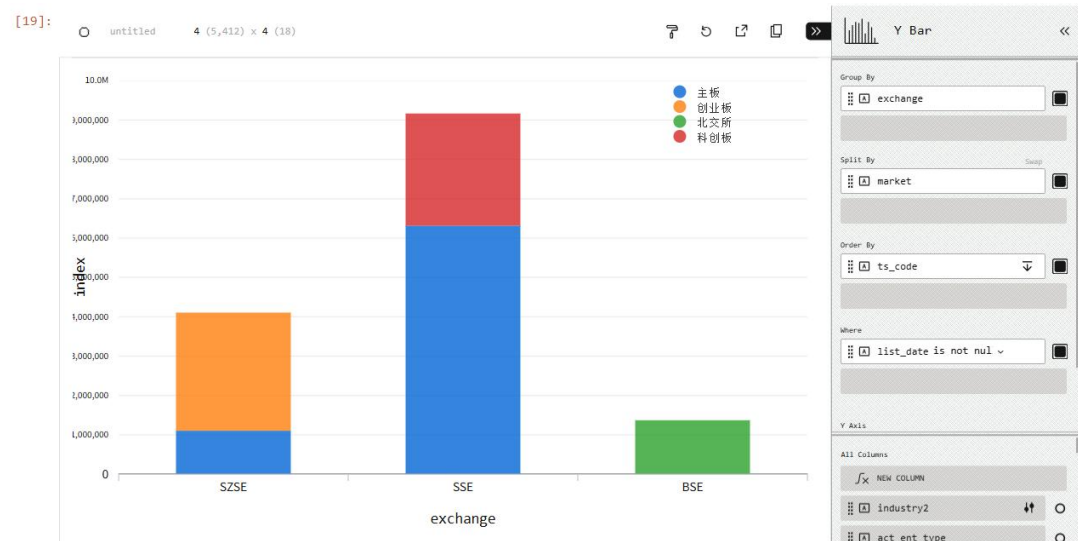
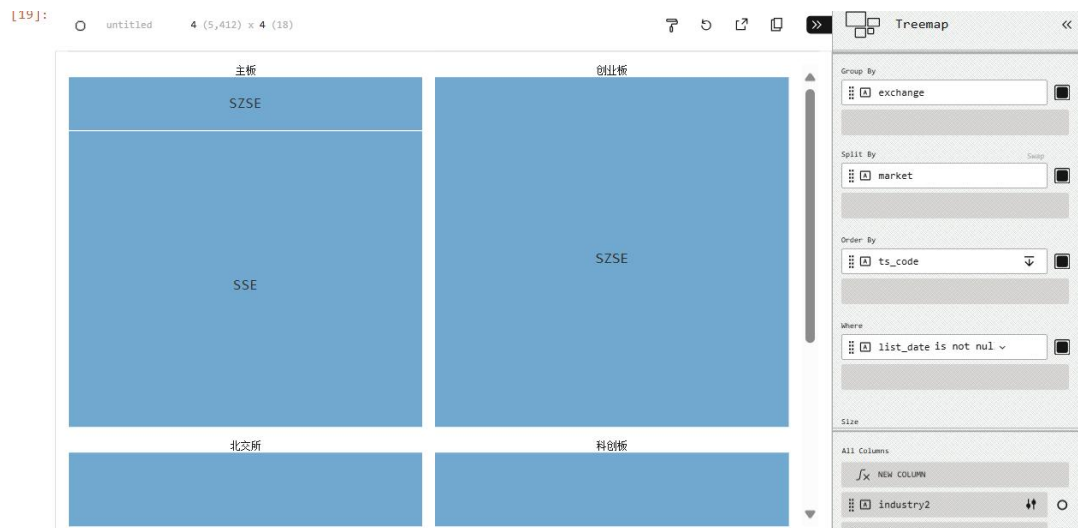
	ts_code	industry	industry2
	Edit	Edit	↕ Edit
- TOTAL	5,412	111	电气设备
主板	3,180	110	电气设备
创业板	1,380	78	软件服务
北交所	266	51	汽车配件
科创板	586	33	半导体

Columns			
⋮ A	count	ts_code	↕
⋮ A	distinct	industry	↕
⋮ A	dominant	industry2	↕

	ts_code	industry	industry2
	Edit	↕ Edit	Edit
- TOTAL	5,412		电气设备
- BSE	266		汽车配件
北交所	266		汽车配件
- SSE	2,280		半导体
主板	1,694		汽车配件
科创板	586		半导体
- SZSE	2,866		元器件
主板	1,486		元器件
创业板	1,380		软件服务

	主板	创业板	北交所	科创板
	ts_code	ts_code	ts_code	ts_code
	Edit	Edit	Edit	Edit
- TOTAL	3,180	1,380	266	586
BSE	-	-	266	-
SSE	1,694	-	-	586
SZSE	1,486	1,380	-	-

Group By	
⋮ A	exchange
Split By	
⋮ A	market



[19]:

