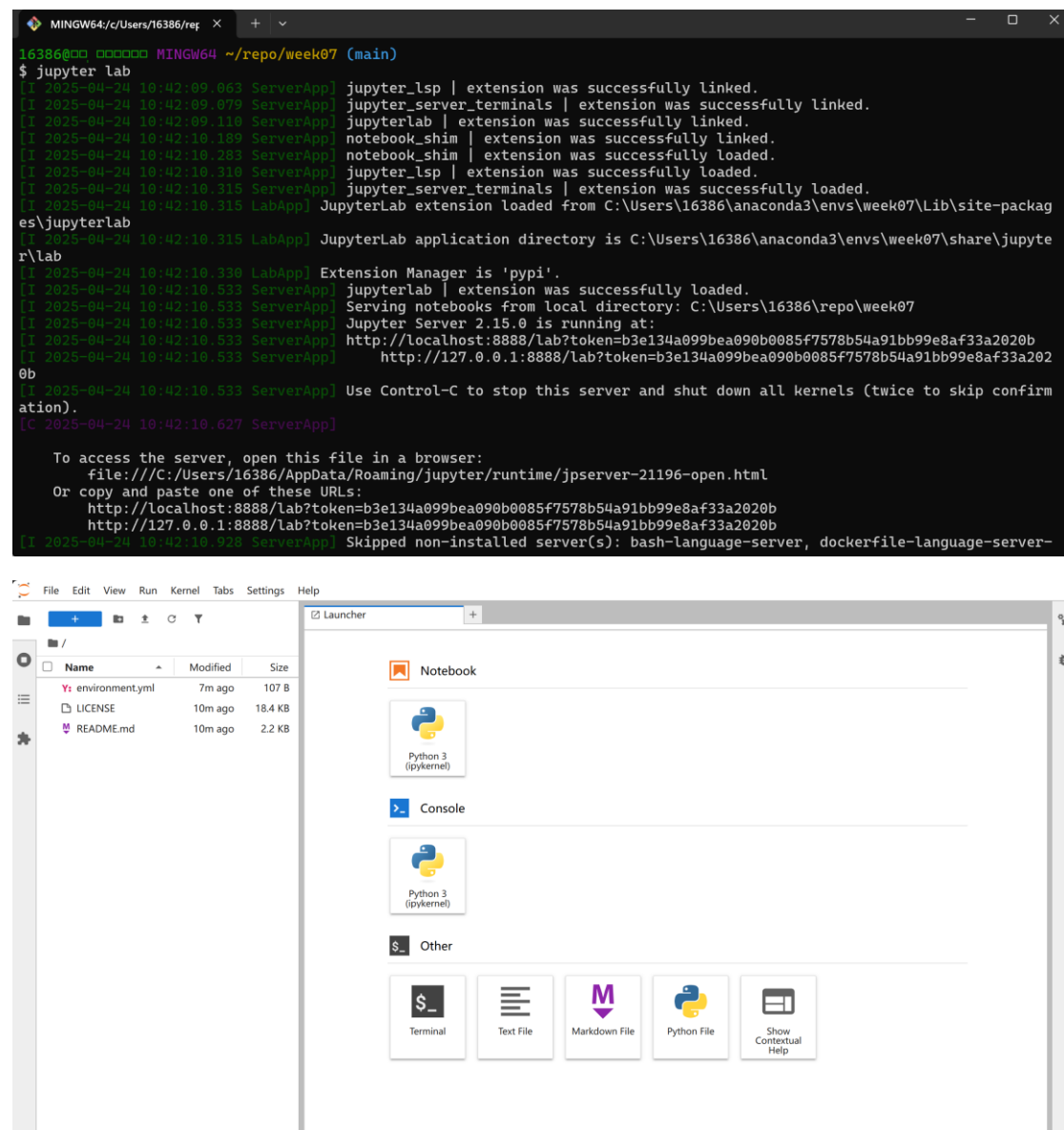


1、启动 jupyter lab 命令



2、在单元格代码，按 Shift+Enter 运行

按 ESC 切换到命令模式，按 Enter 切换到编写模式

命令模式下，j:下一个，k:上一个，a:在上方添加，b:在下方添加，dd:删除，Shift:多选，x:剪切，c:复制，v:粘贴，Shift+M:合并，z:撤销，Shift+Z:重做，Shift+L:显示/隐藏代码行号

编写模式下，Ctrl+Shift+-:切分单元格

序号为 i 的输出，可以用 `j` 变量来引用

序号为 * 表示代码运行中，尚未返回，按 ii 可以打断(终端 ctrl+c)

命令模式下，按 m 切换至 Markdown 模式，按 y 切换至 Python 模式


```
[ ]: ## 1. 标题层级展示

## 2. 列表的运用
### 无序列表
- 这是无序列表的第一项
- 第二项
- 第三项

### 有序列表
1. 这是有序列表的第一项
2. 第二项
3. 第三项

## 3. 插入图片
这里展示一张美丽的风景图片：

## 4. 表格呈现
| 姓名 | 年龄 | 职业 |
| --- | --- | --- |
| 张三 | 25 | 工程师 |
| 李四 | 30 | 设计师 |
| 王五 | 22 | 学生 |
```

trial-jupyterlab.ipynb Python 3 (ipykernel)

2. 列表的运用

无序列表

- 这是无序列表的第一项
- 第二项
- 第三项

有序列表

- 这是有序列表的第一项
- 第二项
- 第三项

3. 插入图片

这里展示一张美丽的风景图片：

4. 表格呈现

姓名	年龄	职业
张三	25	工程师
李四	30	设计师
王五	22	学生

```
[ ]: <table>
  <thead>
    <tr>
      <th rowspan="2">序号</th>
      <th rowspan="2">产品名称</th>
      <th colspan="3">季度销售数据</th>
      <th rowspan="2">年度总计</th>
      <th rowspan="2">评价</th>
    </tr>
    <tr>
      <th>第一季度</th>
      <th>第二季度</th>
      <th>第三季度</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>智能手机</td>
      <td>1500</td>
      <td>1800</td>
      <td>2200</td>
      <td>5500</td>
      <td><i class="fa-solid fa-thumbs-up text-green-500"></i> 畅销</td>
    </tr>
    <tr>
      <td>2</td>
      <td>平板电脑</td>
      <td>800</td>
      <td>900</td>
      <td>1000</td>
      <td>2700</td>
      <td><i class="fa-solid fa-check text-yellow-500"></i> 稳定</td>
    </tr>
  </tbody>
</table>
```

序号	产品名称	季度销售数据			年度总计	评价
		第一季度	第二季度	第三季度		
1	智能手机	1500	1800	2200	5500	<i>畅销</i>
2	平板电脑	800	900	1000	2700	<i>稳定</i>
3	智能手表	300	350	400	1050	<i>增长中</i>

备注：以上数据为模拟数据，仅供参考。

```
[ ]: |
```

```
[ ]: 多元函数  $f(x_1, x_2, \dots, x_n)$  在点  $(\mathbf{a} = (a_1, a_2, \dots, a_n))$  处的二阶泰勒展开式为:


$$f(\mathbf{x}) = f(\mathbf{a}) + \sum_{i=1}^n \frac{\partial f}{\partial x_i} \bigg|_{\mathbf{a}} (x_i - a_i) + \frac{1}{2} \sum_{i,j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j} \bigg|_{\mathbf{a}} (x_i - a_i)(x_j - a_j) + R_2(\mathbf{x})$$


其中  $R_2(\mathbf{x})$  是余项。

矩阵指数函数的定义: 设  $(A)$  是一个  $(n \times n)$  的方阵, 则矩阵指数函数  $(e^A)$  定义为


$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k = I + A + \frac{1}{2!} A^2 + \frac{1}{3!} A^3 + \dots$$


其中  $(I)$  是  $(n \times n)$  的单位矩阵。

傅里叶变换对: 函数  $(f(t))$  的傅里叶变换  $(F(\omega))$  定义为


$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$


其逆傅里叶变换为


$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

```

多元函数 $f(x_1, x_2, \dots, x_n)$ 在点 $(\mathbf{a} = (a_1, a_2, \dots, a_n))$ 处的二阶泰勒展开式为:

$$f(\mathbf{x}) = f(\mathbf{a}) + \sum_{i=1}^n \frac{\partial f}{\partial x_i} \Big|_{\mathbf{x}=\mathbf{a}} (x_i - a_i) + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j} \Big|_{\mathbf{x}=\mathbf{a}} (x_i - a_i)(x_j - a_j) + R_2(\mathbf{x})$$

其中 $(R_2(\mathbf{x}))$ 是余项。

矩阵指数函数的定义: 设 (A) 是一个 $(n \times n)$ 的方阵, 则矩阵指数函数 (e^A) 定义为

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k = I + A + \frac{1}{2!} A^2 + \frac{1}{3!} A^3 + \dots$$

其中 (I) 是 $(n \times n)$ 的单位矩阵。

傅里叶变换对: 函数 $(f(t))$ 的傅里叶变换 $(F(\omega))$ 定义为

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

其逆傅里叶变换为

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

3、在 IPython 提示符下, 运行代码设置 Tushare Token

```
MINGW64/c/Users/16386/rep x MINGW64/c/Users/16386 x + v
IPython 9.1.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: You can use 'files = !ls *.png'

In [1]: import tushare as ts

In [2]:

In [2]: quit
(week07)
16386@ MINGW64 ~/repo/week07 (main)
$ ipython
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 22:08:16) [MSC 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 'ipython --help-all' or 'less' to view all the IPython configuration options.

In [1]: quit
(week07)
16386@ MINGW64 ~/repo/week07 (main)
$ python
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 22:08:16) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tushare as ts
>>> ts.set_token("1a10a689130905f22fdc2e9ff1c3e9f2bbb3cb1e3fc6d9502647fd3c")
>>> quit
Use quit() or Ctrl-Z plus Return to exit
>>> quit()
(week07)
16386@ MINGW64 ~/repo/week07 (main)
$
```

运行代码向 Tushare 服务器请求 IPO 新股列表数据, 并保存在本地

```
IPython: Crepo/week07 x MINGW64: c:/Users/16386 x + v
16386@MINGW64 ~/repo/week07 (main)
$ ipython
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 22:08:16) [MSC 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: You can use 'files = !ls *.png'

In [1]: import tushare as ts

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

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

In [4]: id(pro)
Out[4]: 3040332706880

In [5]: pro
Out[5]: <tushare.pro.client.DataApi at 0x2c3e1f37440>

In [6]: pro.new_share()
Out[6]:
```

	ts_code	sub_code	name	ipo_date	issue_date	amount	market_amount	price	pe	limit_amount	funds	ballot
0	301595.SZ	301595	太力科技	20250508	None	2707.0		0.0	0.00	0.00	0.65	0.000 0
1	688755.SH	787755	汉邦科技	20250507	None	2200.0		0.0	0.00	0.00	0.50	0.000 0
2	301636.SZ	301636	泽润新能	20250428	None	1597.0		0.0	0.00	0.00	0.45	0.000 0
3	001400.SZ	001400	江顺科技	20250415	20250424	1500.0	1500.0	37.36	15.32		1.50	5.604 0

```
MINGW64: c:/Users/16386/rep x MINGW64: c:/Users/16386 x + v
[2000 rows x 12 columns]

In [7]: df = pro.new_share()

In [8]: df.to_parquet("new_share.parquet")

ImportError                                Traceback (most recent call last)
Cell In[8], line 1
----> 1 df.to_parquet(

File ~\anaconda3\envs\week07\Lib\site-packages\pandas\util\_decorators.py:333, in deprecate_nonkeyword_arguments.<locals>
>.decorate.<locals>.wrapper(*args, **kwargs)
    327 if len(args) > num_allow_args:
    328     warnings.warn(
    329         msg.format(arguments=_format_argument_list(allow_args)),
    330         FutureWarning,
    331         stacklevel=find_stack_level(),
    332     )
--> 333 return func(*args, **kwargs)

File ~\anaconda3\envs\week07\Lib\site-packages\pandas\core\frame.py:3113, in DataFrame.to_parquet(self, path, engine, co
mpression, index, partition_cols, storage_options, **kwargs)
    3032 """
    3033 Write a DataFrame to the binary parquet format.
    3034 (...) 3109 >>> content = f.read()
    3110 """
    3111 from pandas.io.parquet import to_parquet
--> 3113 return to_parquet(
    3114     self,
```

资源管理器

WEEK07

.ipynb_checkpoints

.gitignore

environment.yml

LICENSE

new_share.parquet

README.md

trial-jupyterlab.ipynb

大纲

时间线

main

environment.yml

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

行 8, 列 12 空格: 2 UTF-8 CRLF () YAML Aide 消耗

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

```
In [6]: df.to_parquet("new_share.parquet")

In [7]: pro.stock_basic()
Out[7]:
```

	ts_code	symbol	name	area	industry	cnsPELL	market	list_date	act_name	act_ent_type
0	000001.SZ	000001	平安银行	深圳	银行	payh	主板	19910403		无实际控制人
1	000002.SZ	000002	万科A	深圳	全国地产	wka	主板	19910129	深圳市人民政府国有资产监督管理委员会	
2	000004.SZ	000004	国华网安	深圳	软件服务	ghwa	主板	19910114	李映彤	民营
3	000006.SZ	000006	深振业A	深圳	区域地产	szya	主板	19920427	深圳市人民政府国有资产监督管理委员会	
4	000007.SZ	000007	全新好	深圳	其他商业	qxh	主板	19920413	王玩虹	民营
...
5409	920111.BJ	920111	聚星科技	None	None	jxkj	北交所	20241111	None	None
5410	920116.BJ	920116	星图测控	None	None	xtck	北交所	20250102	None	None
5411	920118.BJ	920118	太湖远大	None	None	thyd	北交所	20240822	None	None
5412	920128.BJ	920128	胜业电气	None	None	sydq	北交所	20241129	None	None
5413	689009.SH	689009	九号公司-WD	北京	摩托车	jhgs	科创板	20201029	None	None

```
[5414 rows x 10 columns]

In [8]: df = pro.stock_basic()

In [9]: df = pro.stock_basic(fields="ts_code,symbol,name,area")

In [10]: df
Out[10]:
```

	ts_code	symbol	name	area
5410	920116.BJ	920116	星图测控	None
5411	920118.BJ	920118	太湖远大	None
5412	920128.BJ	920128	胜业电气	None
5413	689009.SH	689009	九号公司-WD	北京

```
[5414 rows x 4 columns]

In [13]: df = pro.stock_basic(fields="ts_code,symbol,name,area,industry,fullname,enname,cnsPELL,market,exchange,curr_type,list_status,list_date,delist_date,is_hs,act_ent_type")

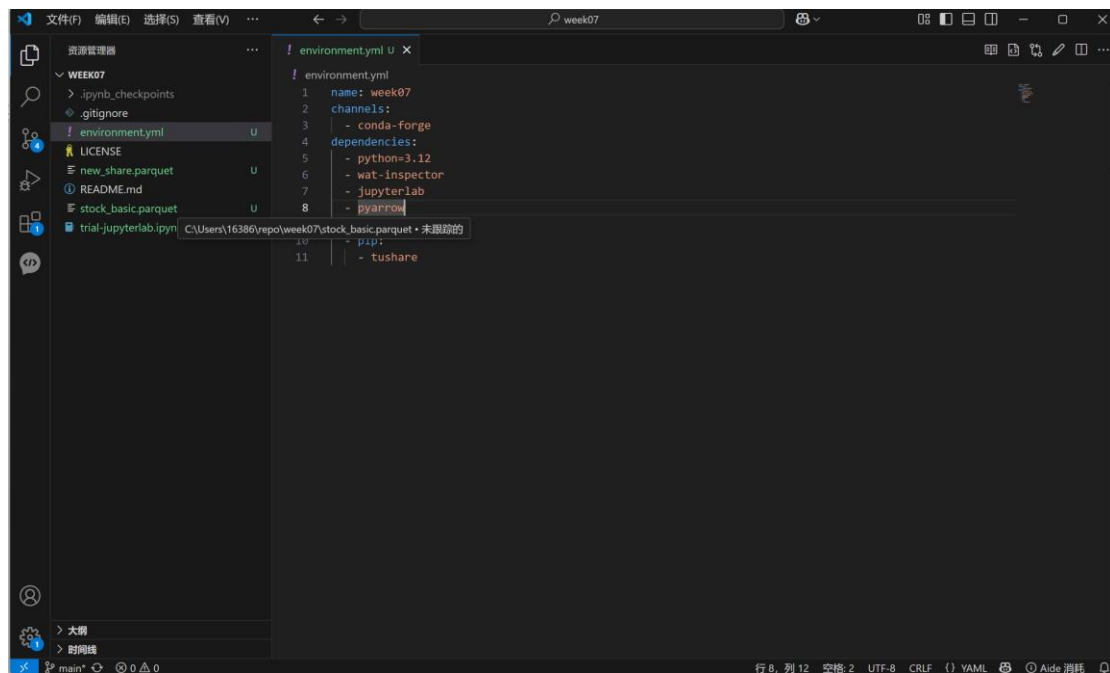
In [14]: df
Out[14]:
```

	ts_code	symbol	name	area	industry	...	list_status	list_date	delist_date	is_hs	act_ent_type	
0	000001.SZ	000001	平安银行	深圳	银行	...	L	19910403		None	S	无
1	000002.SZ	000002	万科A	深圳	全国地产	...	L	19910129		None	S	地方国企
2	000004.SZ	000004	国华网安	深圳	软件服务	...	L	19910114		None	N	民营企业
3	000006.SZ	000006	深振业A	深圳	区域地产	...	L	19920427		None	S	地方国企
4	000007.SZ	000007	全新好	深圳	其他商业	...	L	19920413		None	N	民营企业
...
5409	920111.BJ	920111	聚星科技	None	None	...	L	20241111		None	N	None
5410	920116.BJ	920116	星图测控	None	None	...	L	20250102		None	N	None
5411	920118.BJ	920118	太湖远大	None	None	...	L	20240822		None	N	None
5412	920128.BJ	920128	胜业电气	None	None	...	L	20241129		None	N	None
5413	689009.SH	689009	九号公司-WD	北京	摩托车	...	L	20201029		None	None	None

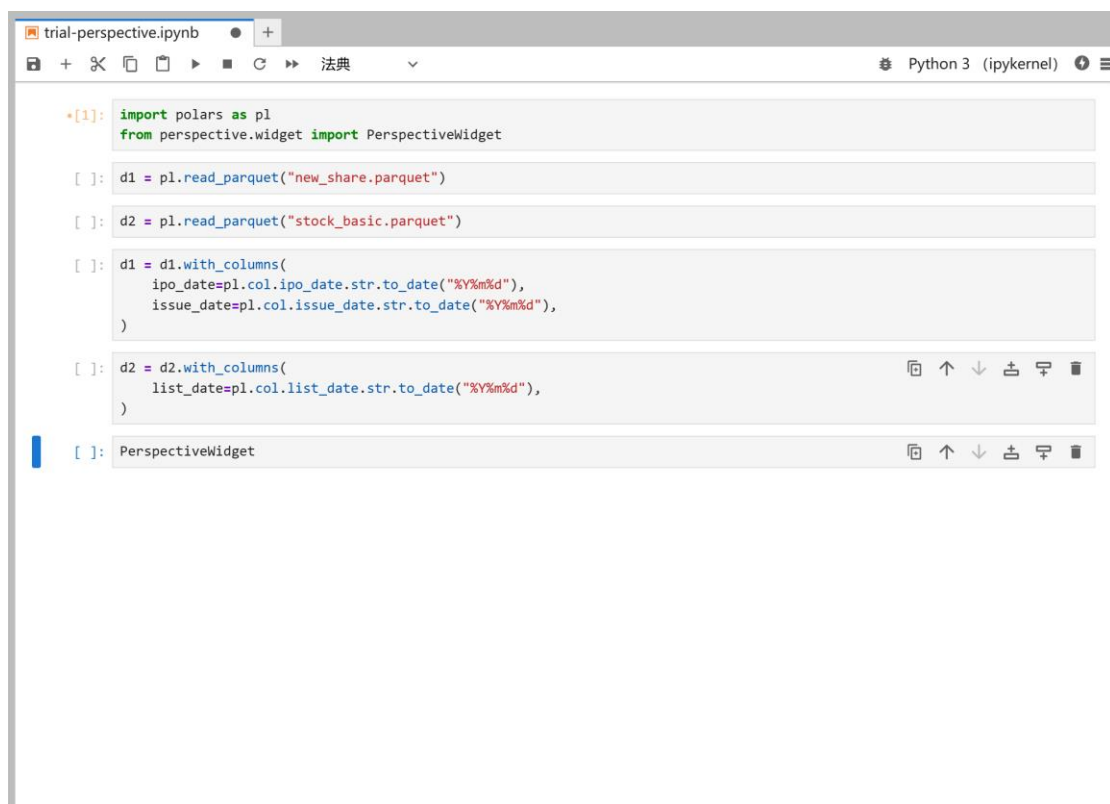
```
[5414 rows x 16 columns]

In [15]: df.to_parquet("stock_basic.parquet")

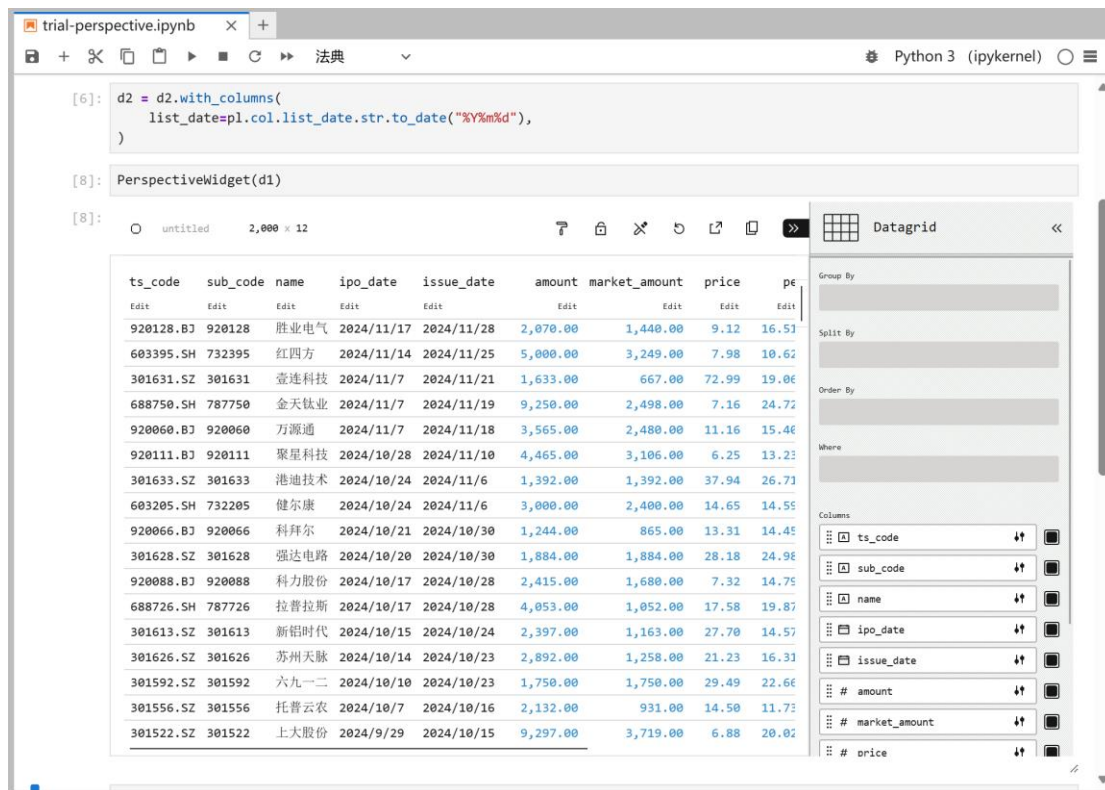
In [16]:
```



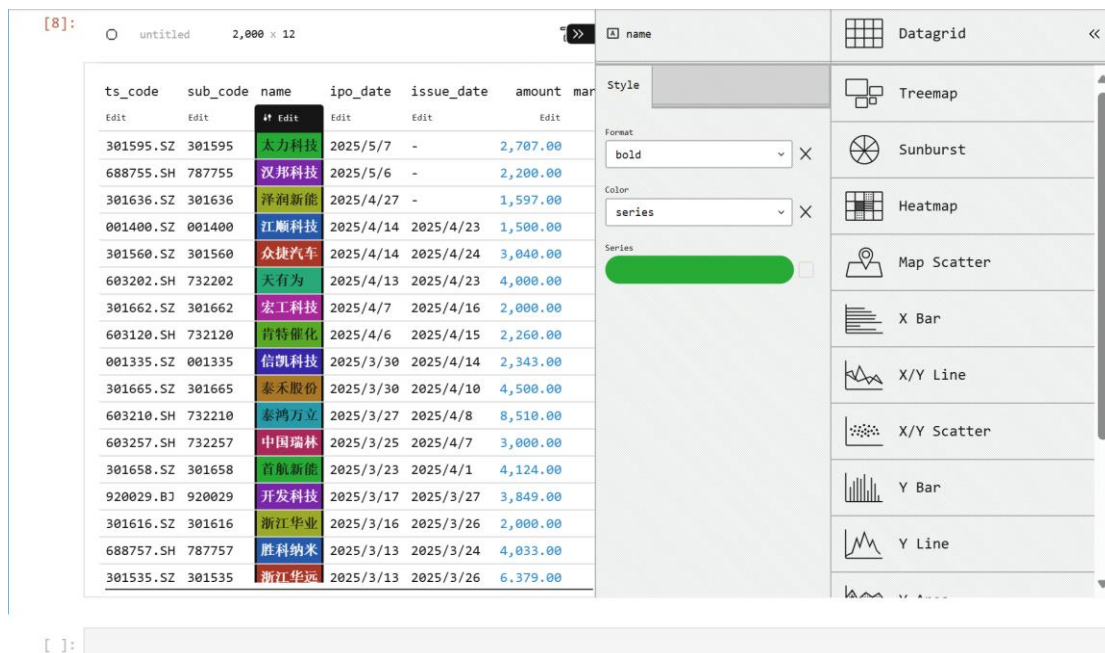
4、调用 `polars.read_parquet` 函数，分别读取 `new_share.parquet` 和 `stock_basic.parquet` 得到 `polars.DataFrame` 对象，命名为 `d1` 和 `d2`

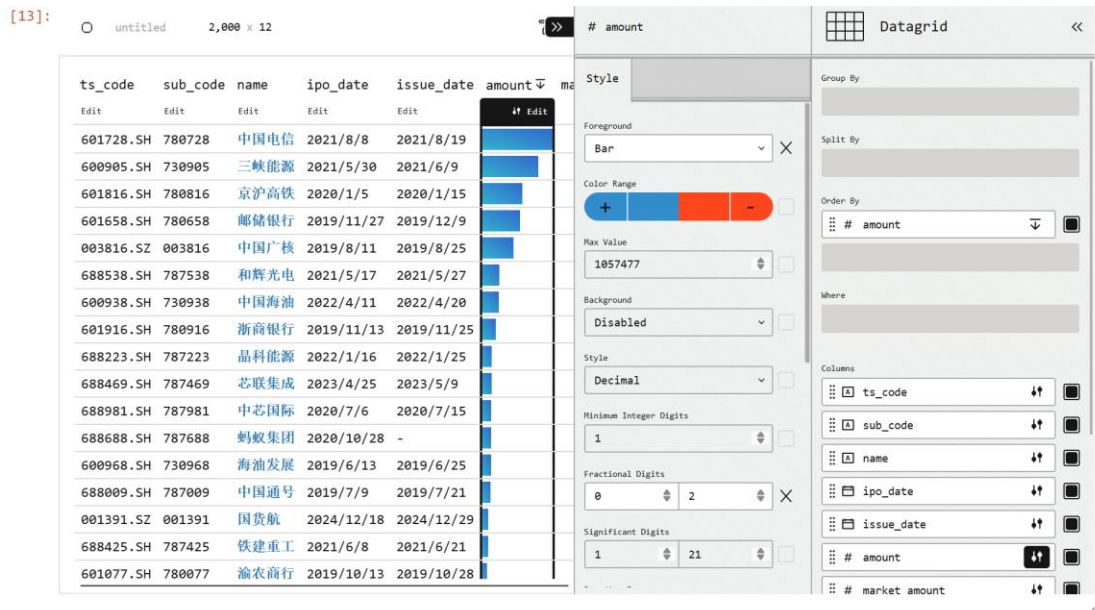


把 `d1` 作为参数传递给 `PerspectiveWidget` 类型进行初始化，返回的对象会呈现在结果里

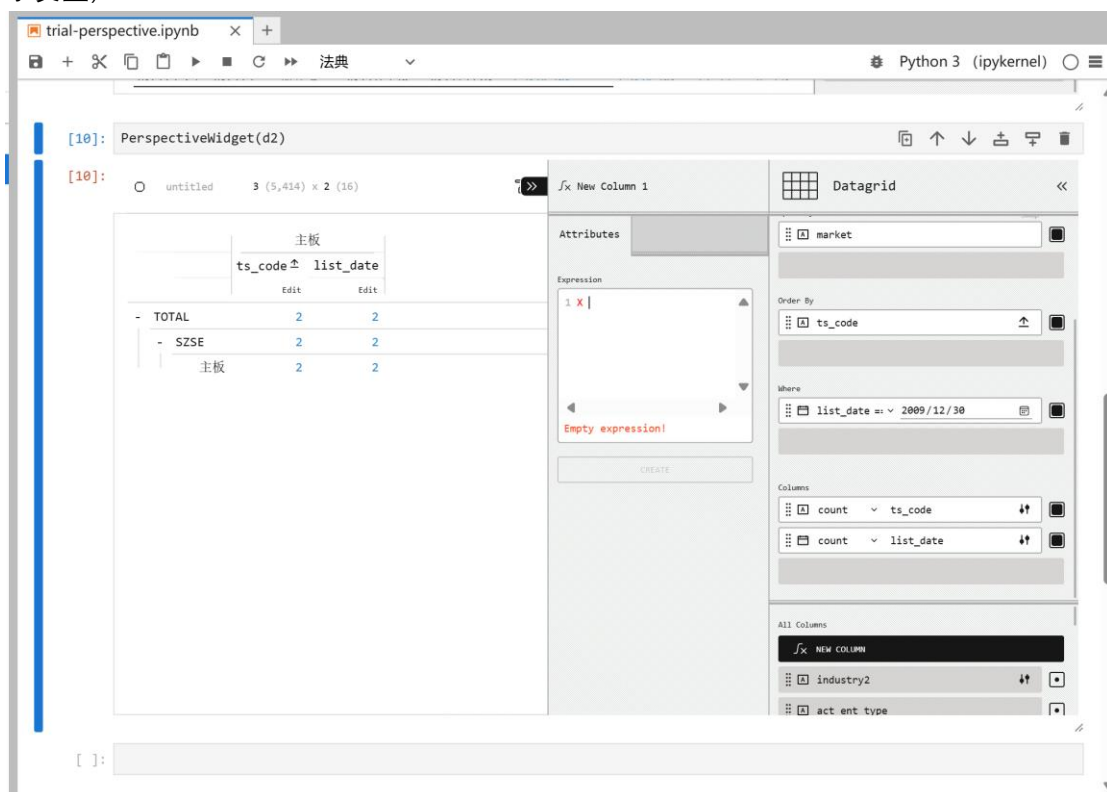


Datagrid 视图下修改显示风格





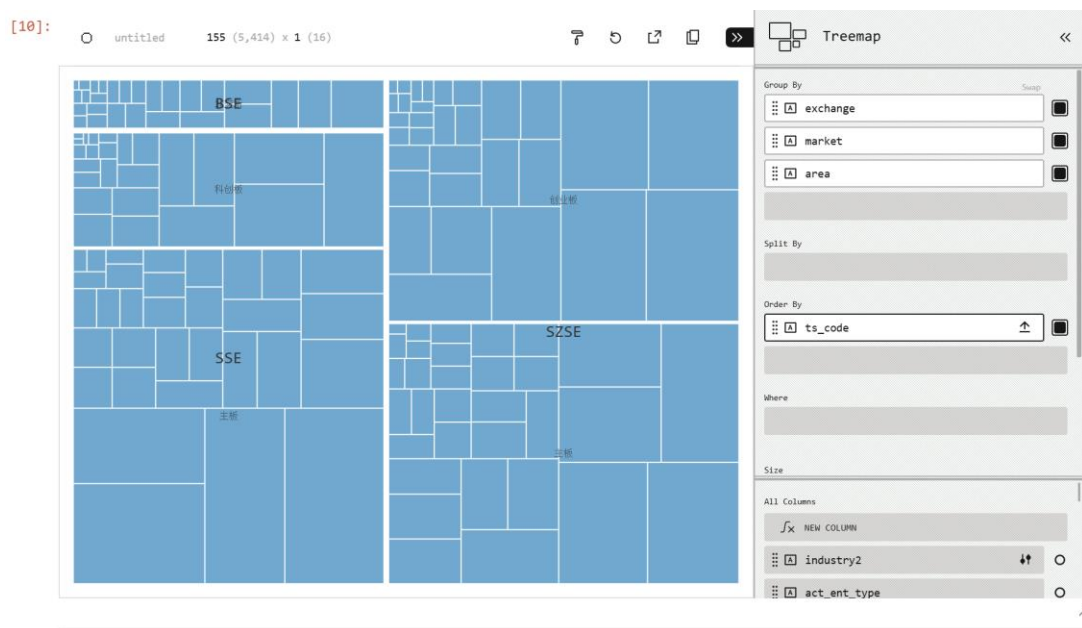
Datagrid 视图下设置 Group By (纵向排列), Split By (横向排列), Order By (作为排序依据, 升序/降序), Where (选择某些列, 进一步设置条件, 进行数据行方向的过滤), Columns (显示变量)



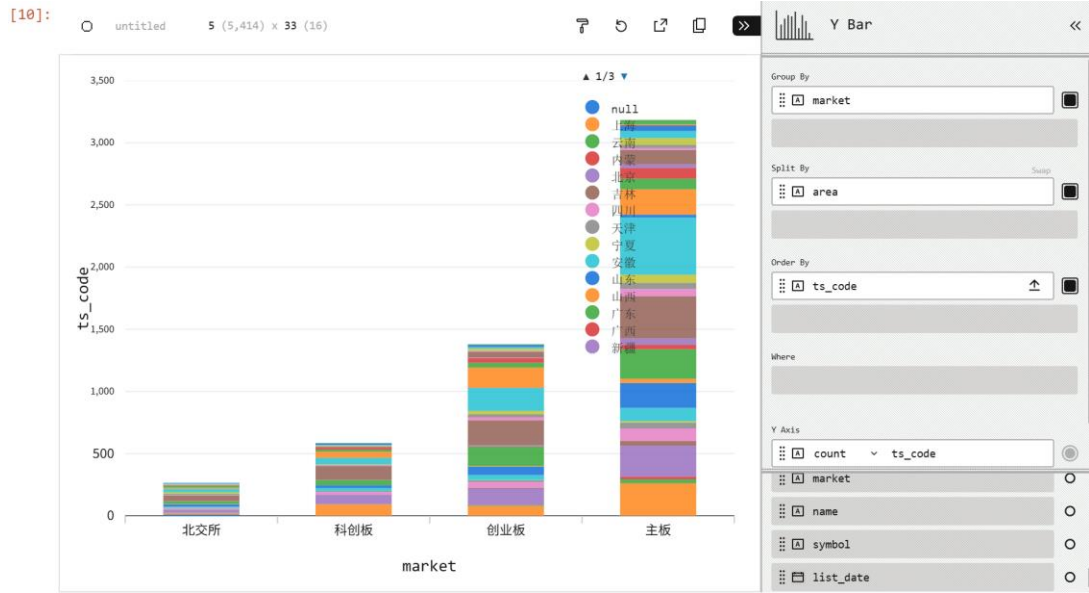
设置 config.json, 用 pathlib.Path.read_text 方法从文件读取字符串

```
[13]: config = {'version':"3.5.1", "plugin": "Datagrid", "plugin_config": {"columns": {}, "edit_mode": "READ_ONLY", "scroll_lock": false},
[14]: print(config)
{"version": "3.5.1", "plugin": "Datagrid", "plugin_config": {"columns": {}, "edit_mode": "READ_ONLY", "scroll_lock": false}, "columns_config": {}, "settings": true, "theme": "Pro Light", "title": null, "group_by": [], "split_by": [], "sort": [{"amount", "asc"}], "filter": [], "expressions": {}, "columns": ["ts_code", "sub_code", "name", "ipo_date", "issue_date", "amount", "market_amount", "price", "pe", "limit_amount", "funds", "ballot"], "aggregates": {}}
[17]: from pathlib import Path
[ ]: config2 = Path("C:/Users/16386/Downloads/untitled.config.json").read_text(encoding="utf8")
[ ]: print(config2)
[21]: import json
[ ]: json.loads(config2)
```

Treemap 视图



Y Bar 视图（横轴用 Group By 控制，纵轴 Y Axis 控制，还可以把每个条形进一步拆分为多个颜色，用 Split By 控制）



Y Line (横轴是时间, 用 Group By 控制, 纵轴是连续型数值变量(经过汇总), 用 Y Axis 控制, 还可以进一步拆分为多条序列, 用 Split By 控制)

