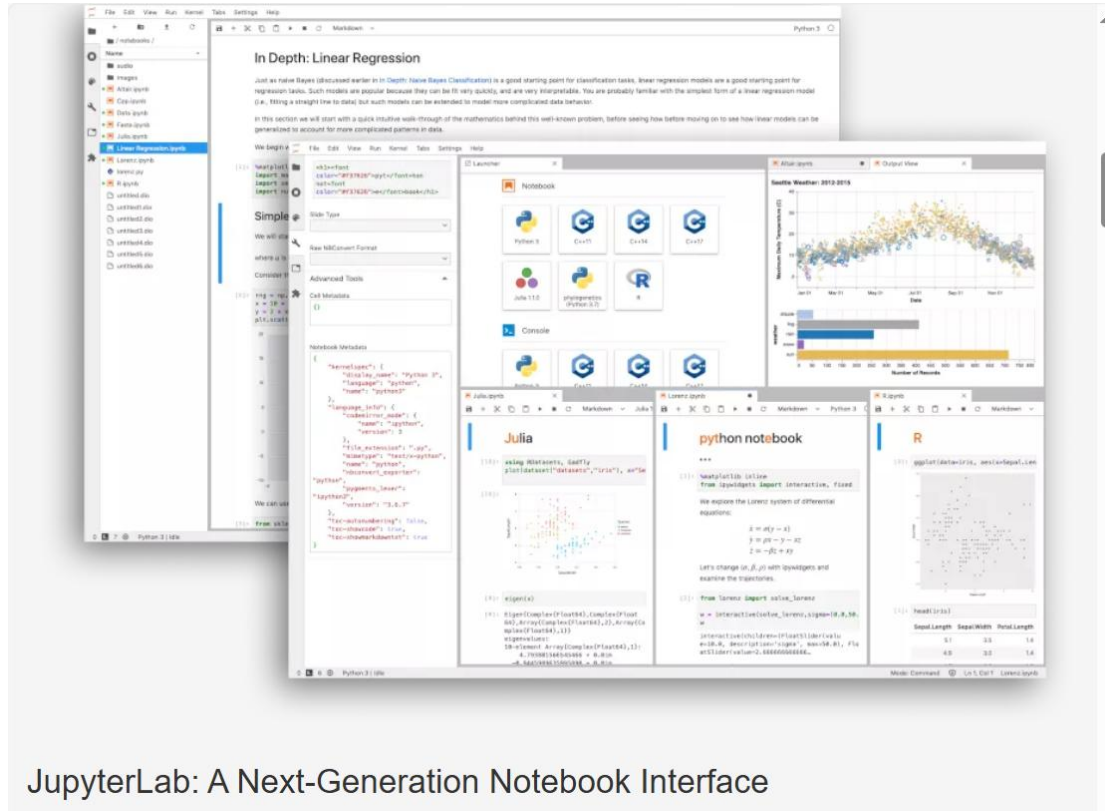
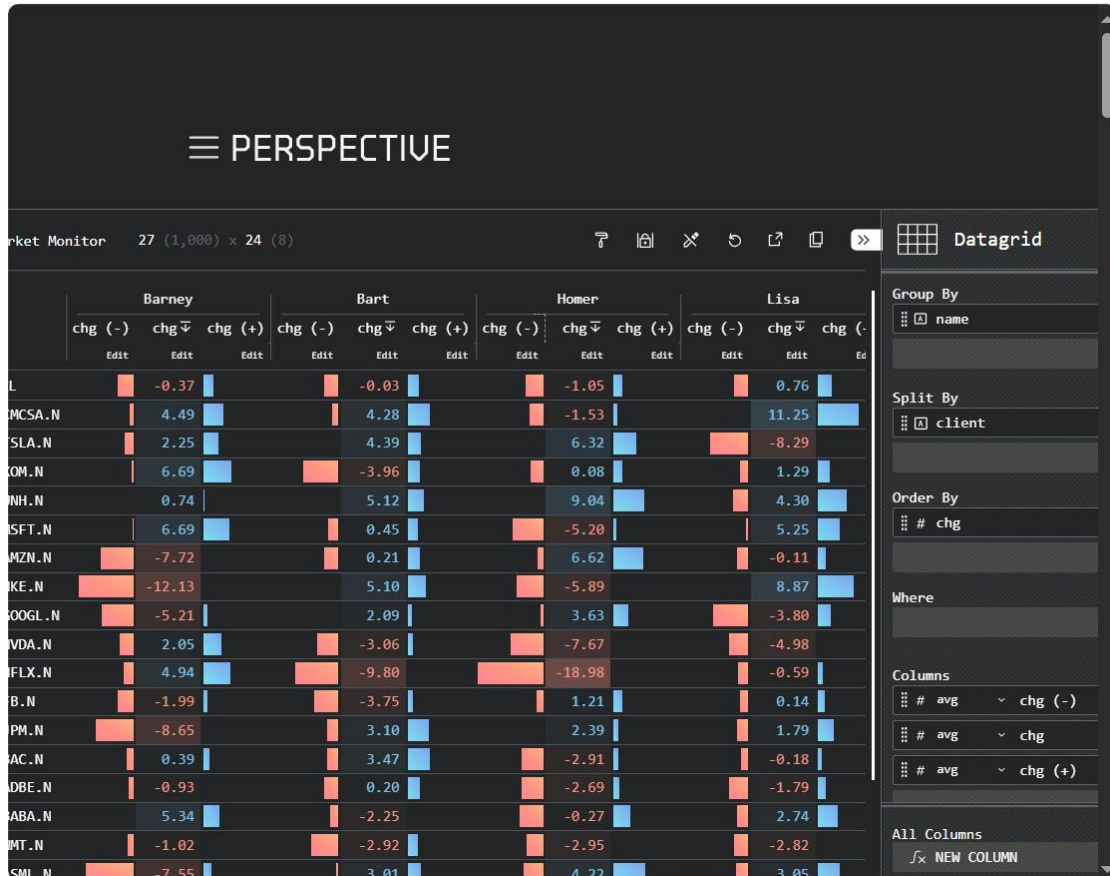


第七周作业

1.Fork 第 07 周打卡 仓库至你的名下，然后将你名下的这个仓库 Clone 到你的本地计算机



JupyterLab: A Next-Generation Notebook Interface



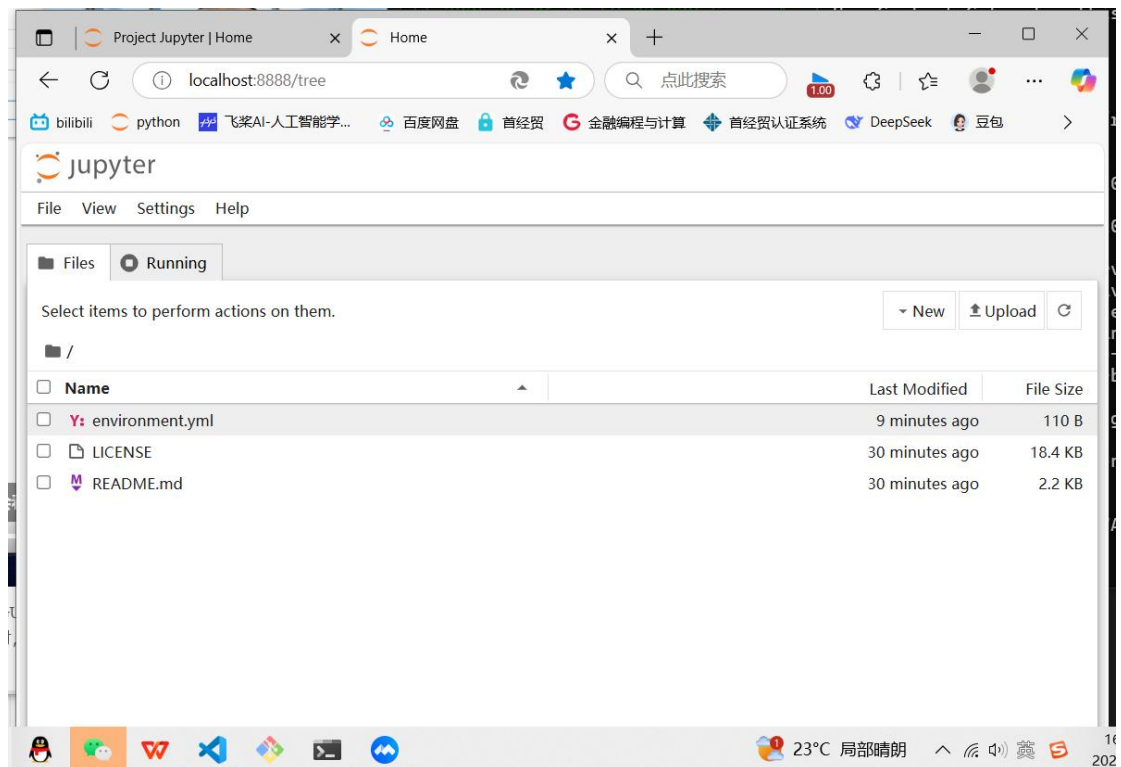
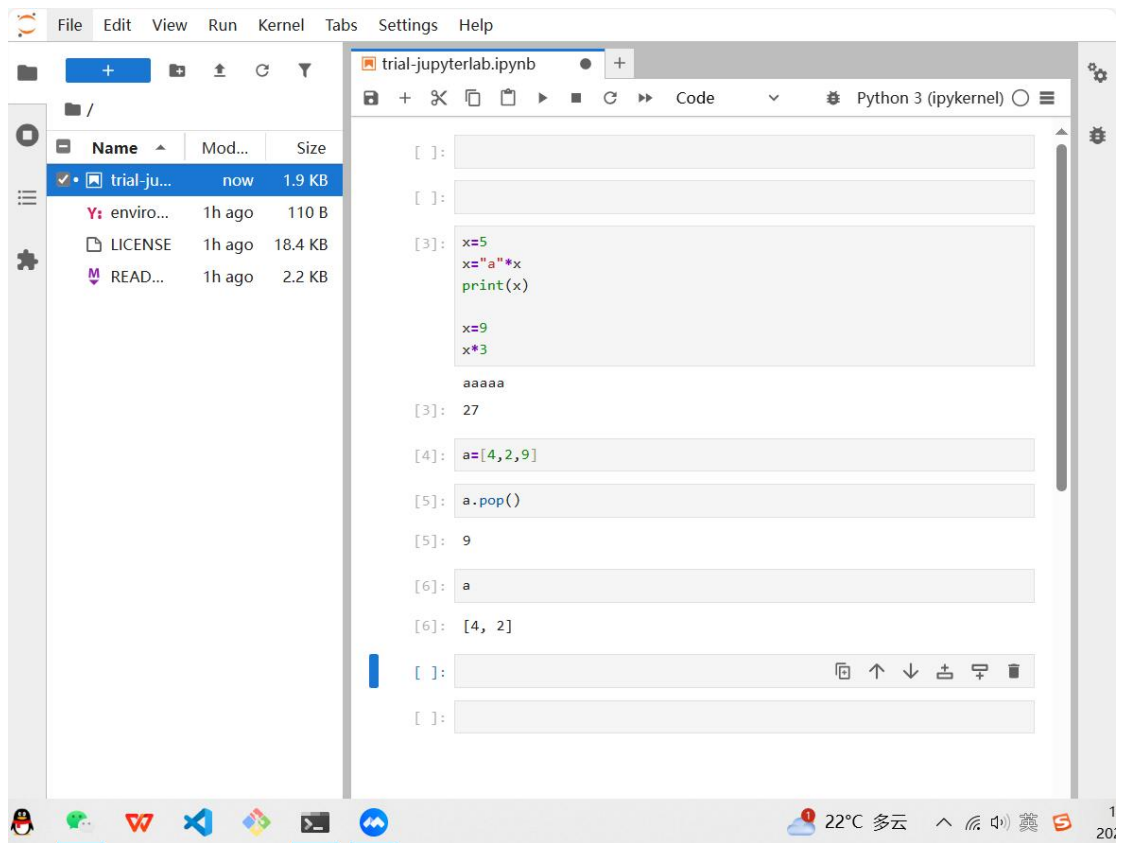
2.用 VS Code 打开项目目录，新建一个 environment.yml 文件，指定安装 Python 3.12 和 jupyterlab，然后运行 conda env create 命令创建 Conda 环境

```
MINGW64:/c/Users/Administr
#
# To activate this environment, use
#
# $ conda activate week07
#
# To deactivate an active environment, use
#
# $ conda deactivate

(base) Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week07 (main)
$ conda activate week07
(week07)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week07 (main)
$ ipython
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 22:0
8:16) [MSC v.1943 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 9.2.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: '?' alone on a line will brings up IPython's help

In [1]: quit
(week07)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week07 (main)
$ python
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 22:0
8:16) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more informati
on.
>>> 
```

3. 在项目目录下，运行 `jupyter lab` 命令，启动 后端 (Backend) 服务，在浏览器里粘贴地址访问 前端 (Frontend) 页面



在 JupyterLab 页面里，新建一个 Notebook，改名为 trial-jupyterlab.ipynb，在里面实践掌握以下功能：

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

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
用豆包 (或 DeepSeek 等任何大模型) 生成一段示例 Markdown 代码, 复制粘贴进 Markdown 单元格, 运行以呈现 (Render)

用豆包 (或 DeepSeek 等任何大模型) 生成一段示例 HTML 代码, 复制粘贴进 Markdown 单元格, 运行以呈现 (Render); 注意不支持 CSS

用豆包 (或 DeepSeek 等任何大模型) 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
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

5.通过 tushare 软件包下载保存一些数据:

在 Tushare 网站上 注册 并登陆, 完善修改个人资料, 浏览阅读 平台介绍 和 数据接口

通过定制的推荐链接 <https://tushare.pro/register?reg=gl5> 完成注册, 将可获赠 2000 平台积分 (有效期一年)。积分达到门槛才有数据接口的使用权限, 否则需要 付费购买积分 (约 200~1000 元/年) 才有权限使用数据接口。本课程的量化投资实战案例, 将主要通过 Tushare 平台获取数据, 请确保拥有足够积分进行实践。

修改 environment.yml 文件, 添加 pip: tushare (注意, conda-forge 没有收录 tushare, 只能从 PyPI 安装, 参考) 依赖项, 运行 `conda env update` 更新 Conda 环境

在终端 (Terminal) 激活 week07 Conda 环境, 运行 `ipython` 命令启动 IPython 交互界面 (IPython 是 Jupyter 项目的一部份, ipython 是 jupyterlab 的依赖项之一)

在 IPython 提示符下, 运行下面的 Python 代码设置 Tushare Token

```
MINGW64:/c/Users/Administr
$ conda activate week07
(week07)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week07 (main)
$ ipython
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 2
2:08:16) [MSC v.1943 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 9.2.0 -- An enhanced Interactive Python. Type '?' for he
lp.
Tip: '?' alone on a line will brings up IPython's help

In [1]: import tushare as ts

In [2]:

In [2]: quit
(week07)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week07 (main)
$ python
Python 3.12.10 | packaged by conda-forge | (main, Apr 10 2025, 2
2:08:16) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more inform
ation.
>>> import tushare as ts
>>> ts.set_token("906917a741a1d700159c5cf8e12959c3be70d053776966
e661fbda6a")
>>> quit()
(week07)
Administrator@DESKTOP-K6SQLQ1 MINGW64 ~/repo/week07 (main)
$
```

```
IPython: C:\repo\week07
Out[5]: <tushare.pro.client.DataApi at 0x1edc0ddc500>
In [6]: pro.new_share()
Out[6]:
```

	ts_code	sub_code	name	...	limit_amount	funds	ballot
0	603014.SH	732014	威高血净	...	1.10	0.000	
0.00							
1	301595.SZ	301595	太力科技	...	0.65	0.000	
0.00							
2	688755.SH	787755	汉邦科技	...	0.50	5.009	
0.00							
3	301636.SZ	301636	泽润新能	...	0.45	5.279	
0.02							
4	920068.BJ	920068	天工股份	...	255.00	2.719	
0.04							
...
1995	300777.SZ	300777	中简科技	...	1.10	2.425	
0.04							
1996	603267.SH	732267	鸿远电子	...	1.60	8.367	
0.03							
1997	600989.SH	730989	宝丰能源	...	22.00	81.550	
0.25							
1998	300778.SZ	300778	新城市	...	2.00	5.466	0
.02							
1999	002953.SZ	002953	日丰股份	...	1.70	4.526	
0.03							

```
[2000 rows x 12 columns]
In [7]:
```



```
In [9]: import wat
```

```
In [10]: wat / df
```

	value:	ts_code	sub_code	name	...	limit_amount	funds
	ballot						
0	603014.SH	732014	威高血净	...	1.10	0.000	
0.00							
1	301595.SZ	301595	太力科技	...	0.65	0.000	
0.00							
2	688755.SH	787755	汉邦科技	...	0.50	5.009	
0.00							
3	301636.SZ	301636	泽润新能	...	0.45	5.279	
0.02							
4	920068.BJ	920068	天工股份	...	255.00	2.719	
0.04							
...
1995	300777.SZ	300777	中简科技	...	1.10	2.425	
0.04							
1996	603267.SH	732267	鸿远电子	...	1.60	8.367	
0.03							
1997	600989.SH	730989	宝丰能源	...	22.00	81.550	
0.25							
1998	300778.SZ	300778	新城市	...	2.00	5.466	0
.02							
1999	002953.SZ	002953	日丰股份	...	1.70	4.526	
0.03							

```
[2000 rows x 12 columns]
```

```
type: pandas.core.frame.DataFrame
```

```
parents: pandas.core.generic.NDFrame, pandas.core.base.PandasObject, pandas.core.accessor.DirNamesMixin, pandas.core.indexing.IndexingMixin, pandas.core.arraylike.OpsMixin
```

```
len: 2000
```



```
IPython 9.2.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: Put a ';' at the end of a line to suppress the printing of output.

In [1]: import tushare as ts

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

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

In [4]: df
Out[4]:
```

	ts_code	sub_code	name	...	limit_amount	funds	ballot
0	603014.SH	732014	威高血净	...	1.10	0.000	
0.00							
1	301595.SZ	301595	太力科技	...	0.65	0.000	
0.00							
2	688755.SH	787755	汉邦科技	...	0.50	5.009	
0.00							
3	301636.SZ	301636	泽润新能	...	0.45	5.279	
0.02							
4	920068.BJ	920068	天工股份	...	255.00	2.719	
0.04							
...
1995	300777.SZ	300777	中简科技	...	1.10	2.425	
0.04							
1996	603267.SH	732267	鸿远电子	...	1.60	8.367	
0.03							
1997	600989.SH	730989	宝丰能源	...	22.00	81.550	
0.25							
1998	300778.SZ	300778	新城市	...	2.00	5.466	0
.02							
1999	002953.SZ	002953	日丰股份	...	1.70	4.526	
0.03							

```
IPython: C:repo/week07

[5411 rows x 10 columns]

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

In [8]: df = pro.stock_basic(fields="ts_code , symbol, name , area,industry,fullname,ename,cnspell,market,exchange,curr_type,list_status,list_date,delist_date,is_hs,act_name,act_ent_type")

In [9]: df
Out[9]:
```

	ts_code	symbol	...	act_name	act_ent_type
0	000001.SZ	000001	...	无实际控制人	
1	000002.SZ	000002	...	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	...	李映彤	民营
3	000006.SZ	000006	...	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	...	王玩虹	民营
...
5406	920489.BJ	920489	...	None	None
5407	920682.BJ	920682	...	None	None
5408	920799.BJ	920799	...	None	None
5409	920819.BJ	920819	...	None	None
5410	689009.SH	689009	...	None	None

```
[5411 rows x 17 columns]

In [10]: df.to_parquet("stock_basic.parquet")

In [11]:
```

6.通过 perspective-python 软件包查看 polars.DataFrame 数据，实践交互式可视化：

修改 environment.yml 文件，添加 perspective-python 和 polars 依赖项，运行 conda env update 更新 Conda 环境

启动 JupyterLab，新建一个 Notebook，改名为 trial-perspective.ipynb

调用 polars.read_parquet 函数，分别读取磁盘 (disk) 中的 new_share.parquet 文件和 stock_basic.parquet 文件，得到内存 (memory) 中的 polars.DataFrame 对象，命名为 d1 和 d2

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

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

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

修改各种列数据类型 (文本、数值、日期) 的显示风格 (`style`)

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)

设置 `Split By` 选项，选择某些列作为拆分依据 (横向排列)

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)

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

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

在 `All Columns` 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

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)

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，或复制 (`copy`) 到剪贴板

把设置 (`config.json`) 复制到剪贴板，粘贴进 `Notebook Cell`，保存成字符串 (`str`)

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`)

可以用 `json.loads` 函数将无结构的 (`unstructured`) 字符串 (`str`) 解析为有结构的 (`structured`) Python 字典 (`dict`)，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

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 (比如 [链接](#)) 进行美化

根据导出的设置代码，在初始化 (`init`) `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

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)

点击 [链接 1](#) 或 [链接 2](#) 学习了解更多 `Treemap` 的概念、适用情形以及实现代码

把 `PerspectiveWidget` 切换为 `Y Bar` 视图，尝试设置各种选项 (`configure`)，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

点击 [链接 1](#) 或 [链接 2](#) 学习了解更多 `Bar Chart` 的概念、适用情形以及实现代码

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系列连续型数值变量 (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**)，把任意其他一系列变量用 `count` (计数) 函数汇总，设置为纵轴 (**Y Axis**)。这样看到的的就是直方图。“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。

把 **PerspectiveWidget** 切换为 **Y Line** 视图，尝试设置各种选项 (configure)，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

点击 [链接 1](#) 或 [链接 2](#) 学习了解更多 **Line Chart** 的概念、适用情形以及实现代码

使用我们的示例数据，可以尝试观察最近几年 **A 股 IPO** 市场的 融资额 (funds) 与 市盈率 (pe) 变化情况。为了加深对数据的理解 和 验证，可以询问豆包 (或 DeepSeek 等任何大模型)，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，由此加强我们对现实背景的理解

也可以使用示例数据，观察对比最近几年不同交易所 (exchange) 或市场 (market) 的平均中签率 (ballot) 情况

把 **PerspectiveWidget** 切换为 **X/Y Scatter** 视图，尝试设置各种选项 (configure)，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

点击 [链接 1](#) 或 [链接 2](#) 学习了解更多 **Scatter Plot** 的概念、适用情形以及实现代码

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)

散点的分布如果杂乱无规律，则意味着 **X** 与 **Y** 没有相关性

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

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

我们经常还可以把用于分类的类别变量 (类别不宜太多) 设置为 **Split By**，从而把一个散点图拆分为多个小散点图 (small multiple)，从而更细致地观察是否存在规律

在我们的示例数据中，融资额 (funds)、市盈率 (pe)、中签率 (ballot) 是数值型连续变量，适合用散点图观察他们的规律，散点可以以个股为单位 (不汇总)，也可以按 行业 (industry) 汇总，或者按 上市时间 (ipo_date) 汇总 (每月分桶)，都可以大胆尝试探索

FileEditViewRunKernelTabsSettingsHelp

trial-perspective.ipynbPython 3 (ipykernel)

Code

"689009.SH" "689009" "九号公司-WD" "北京" "摩托车" "九号有限公司" "Ninebot Limited" "jhgs" "科创板" "SSE" "CNY" "L"

[11]: PerspectiveWidget(d1)

[11]:

untitled2,000 × 12

ts_codesub_codenameipo_dateissue_dateamountmarket_amountprice

EditEditEditEditEditEditEdit

603210.SH732210泰鸿万立2025/3/272025/4/88,510.005,739.008.6018

603257.SH732257中国瑞林2025/3/252025/4/73,000.001,992.0020.5218

301658.SZ301658首航新能2025/3/232025/4/14,124.001,617.0011.8015

920029.BJ920029开发科技2025/3/172025/3/273,849.002,677.0030.388

301616.SZ301616浙江华业2025/3/162025/3/262,000.002,000.0020.8725

688757.SH787757胜科纳米2025/3/132025/3/244,033.001,036.009.0842

301535.SZ301535浙江华远2025/3/132025/3/266,379.002,488.004.9225

301629.SZ301629矽电股份2025/3/102025/3/231,043.001,043.0052.2826

001382.SZ001382新亚电缆2025/3/102025/3/206,200.004,340.007.4018

603124.SH732124江南新材2025/3/92025/3/193,644.002,335.0010.5412

301501.SZ301501恒鑫生活2025/3/62025/3/182,550.001,237.0039.9219

301479.SZ301479弘景光电2025/3/32025/3/171,589.00705.0041.9023

603271.SH732271永杰新材2025/2/272025/3/104,920.003,149.0020.6016

Group By

Split By

Order By

Where

Columns

ts_codesub_codenameipo_date

20:00