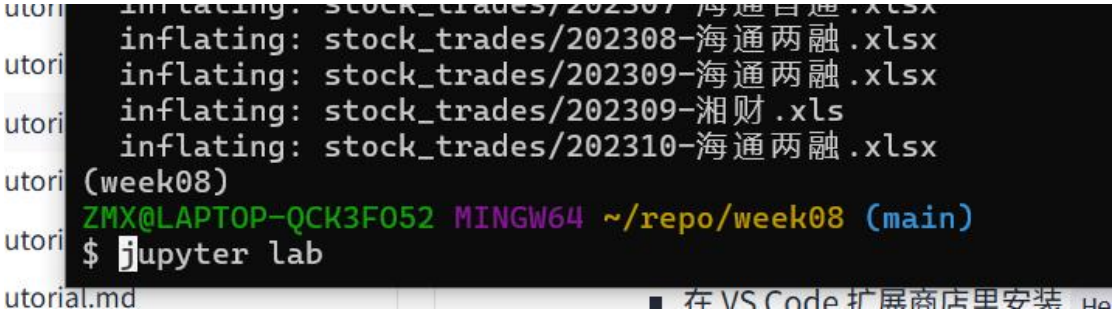
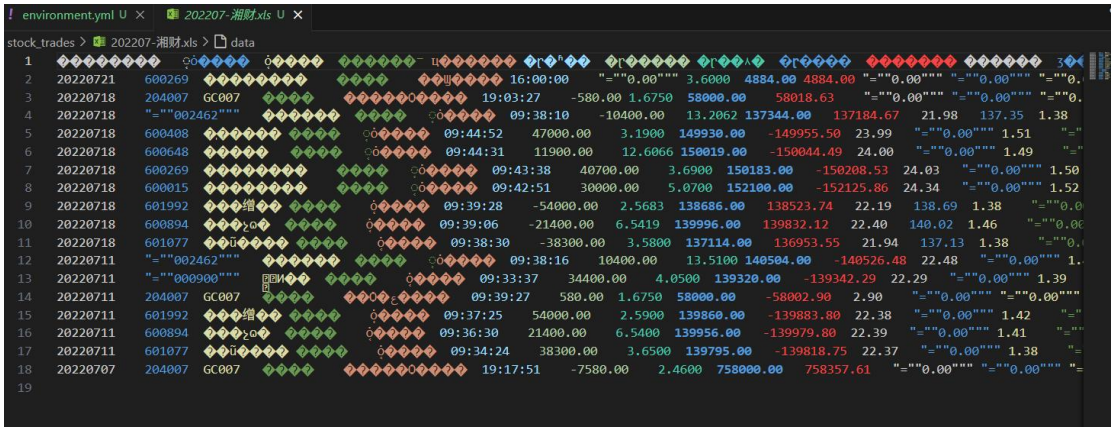
如何合并 使用 jupyter lab

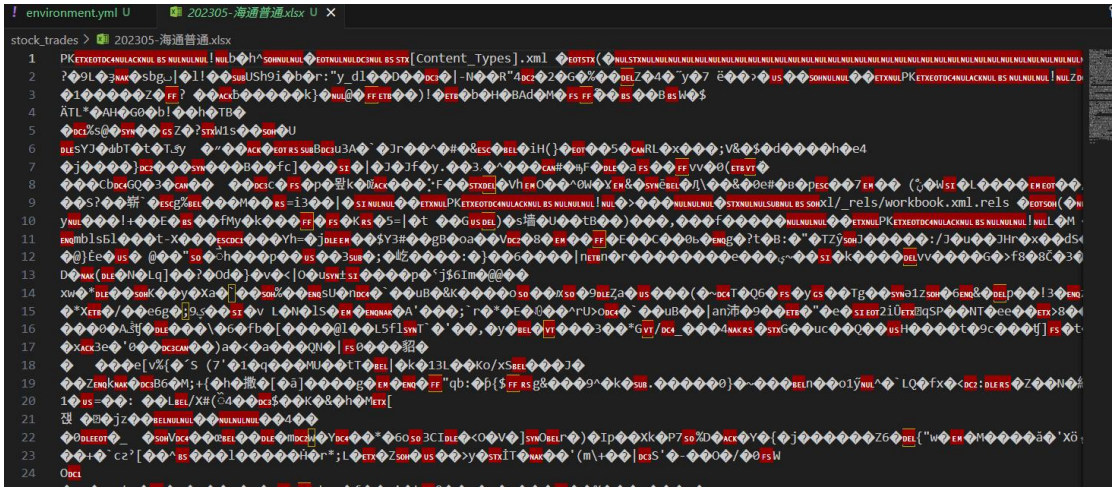


报错 （纯文本）

```
[2]: pl.read_excel("stock_trades/202207-湘财.xls")

---------------------------------------------------------------------------
CalamineError                             Traceback (most recent call last)
Cell In[2], line 1
----> 1 pl.read_excel(                          )

File D:\anaconda3\envs\week08\Lib\site-packages\polars\_utils\deprecation.py:119, in deprecate_renamed_parameter.<locals>.decorate.<locals>.wrapper(*args, **kwargs)
    114 @wraps(function)
    115 def wrapper(*args: P.args, **kwargs: P.kwargs) -> T:
    116     _rename_keyword_argument(
    117         old_name, new_name, kwargs, function.__qualname__, version
    118     )
--> 119     return function(*args, **kwargs)

File D:\anaconda3\envs\week08\Lib\site-packages\polars\_utils\deprecation.py:119, in deprecate_renamed_parameter.<locals>.decorate.<locals>.wrapper(*args, **kwargs)
    114 @wraps(function)
    115 def wrapper(*args: P.args, **kwargs: P.kwargs) -> T:
    116     _rename_keyword_argument(
    117         old_name, new_name, kwargs, function.__qualname__, version
    118     )
```



二进制



environment.yml U 202305-海通普通.xlsx U

stock_trades > 202305-海通普通.xlsx

	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded Text
00000000	50 4B 03 04 14 00 06 00 08 00 00 00 21 00 62 EE	P K . . . . . ! . b .
00000010	9D 68 5E 01 00 00 90 04 00 00 13 00 08 02 5B 43	. h ^ . . . . . [C
00000020	6F 6E 74 65 6E 74 5F 54 79 70 65 73 5D 2E 78 6D	o n t e n t _ T y p e s] . x m
00000030	6C 20 A2 04 02 28 A0 00 02 00 00 00 00 00 00 00	l . . . (. . . . .
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000080	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000090	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
000000A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
000000B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
000000C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
000000D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
000000E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
000000F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000100	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .
00000150	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . .

environment.yml U 202207-湘财.xls U

stock_trades > 202207-湘财.xls > data

	发生日期	证券代码	证券名称	买卖标志	业务名称	成交时间	成交数量	成交价格	成交金额	发生金额	手续费	印花税	过户费	其他费
2	20220721	600269	赣粤高速	卖出	股息入账	16:00:00	"=""0.00"""	3.6000	4884.00	4884.00	"=""0.00"""	"=""0.00"""	"=""0.00"""	"=""0.
3	20220718	204007	GC007	卖出	拆出质押购回	19:03:27	-580.00	1.6750	58000.00	58018.63	"=""0.00"""	"=""0.00"""	"=""0.00"""	"
4	20220718	"=""002462"""	嘉事堂	卖出	证券卖出	09:38:10	-10400.00	13.2062	137344.00	137184.67	21.98	137.35	1.38	"=""0.
5	20220718	600408	安泰集团	买入	证券买入	09:44:52	47000.00	3.1900	149930.00	-149955.50	23.99	"=""0.00"""	1.51	"=""0.
6	20220718	600648	外高桥	买入	证券买入	09:44:31	11900.00	12.6066	150019.00	-150044.49	24.00	"=""0.00"""	1.49	"=""0.00"
7	20220718	600269	赣粤高速	买入	证券买入	09:43:38	40700.00	3.6900	150183.00	-150208.53	24.03	"=""0.00"""	1.50	"=""0.
8	20220718	600015	华夏银行	买入	证券买入	09:42:51	30000.00	5.0700	152100.00	-152125.86	24.34	"=""0.00"""	1.52	"=""0.
9	20220718			卖出	证券卖出	09:39:28	-54000.00	2.5683	138686.00	138523.74	22.19	138.69	1.38	"=""0.00"
10	20220718	600894	广日股份	卖出	证券卖出	09:39:06	-21400.00	6.5419	139996.00	139832.12	22.40	140.02	1.46	"=""0.00"
11	20220718	601077	渝农商行	卖出	证券卖出	09:38:30	-38300.00	3.5800	137114.00	136953.55	21.94	137.13	1.38	"=""0.00"
12	20220711	"=""002462"""	嘉事堂	买入	证券买入	09:38:16	10400.00	13.5100	140504.00	-140526.48	22.48	"=""0.00"""	1.41	"=
13	20220711	"=""000900"""	现代投资	买入	证券买入	09:33:37	34400.00	4.0500	139320.00	-139342.29	22.29	"=""0.00"""	1.39	
14	20220711	204007	GC007	卖出	质押回购拆出	09:39:27	580.00	1.6750	58000.00	-58002.90	2.90	"=""0.00"""	"=""0.00"""	"=""0
15	20220711	601992	金隅集团	买入	证券买入	09:37:25	54000.00	2.5900	139860.00	-139883.80	22.38	"=""0.00"""	1.42	"=""0.
16	20220711	600894	广日股份	买入	证券买入	09:36:30	21400.00	6.5400	139956.00	-139979.80	22.39	"=""0.00"""	1.41	"=""0.
17	20220711	601077	渝农商行	买入	证券买入	09:34:24	38300.00	3.6500	139795.00	-139818.75	22.37	"=""0.00"""	1.38	"=""0.
18	20220707	204007	GC007	卖出	拆出质押购回	19:17:51	-7580.00	2.4600	758000.00	758357.61	"=""0.00"""	"=""0.00"""	"=""0.00"	
19														

Col 2: 证券代码

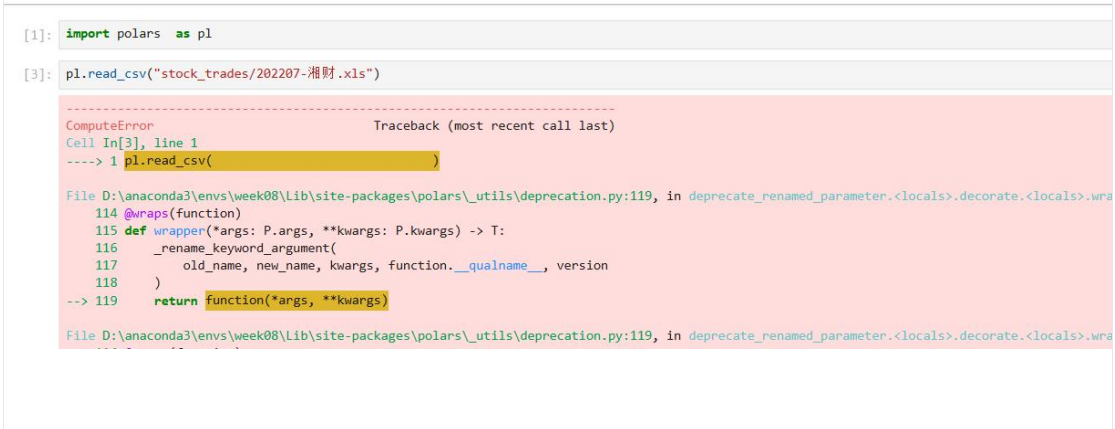
○

在 VS Code 界面右下角 UTF-8 处点击鼠标，在菜单里选择 “Reopen with Encoding”，进一步选择 GB18030 编解码器，就能够正确地看到汉字了

在 VS Code 里可以看出，202207-湘财.xls 文件实际上并不是 Excel 格式，而是 CSV 格式，而且分隔符 (seporator) 不是逗号 (,)，而是 TAB (\t)

尝试使用 polars.read_csv() 函数重新读取 202207-湘财.xls 文件，参照函数文档恰当指定参数 (可以在 Notebook 右键菜单里选择 “Show Contextual Help” 方便查看内置文档)，反复尝试，最终返回正确的 polars.DataFrame 对象，命名为 df

报错：读到乱码



```
[1]: import polars as pl

[5]: pl.read_csv("stock_trades/202207-湘财.xls",encoding="gb18030")

[5]: shape: (17, 1)
```

发生日期 证券代码 证券名称 买卖标志 业务名称 成交时间 成交数量 成交价格 成交金额 发生金额 手续费 印花税 过户费 其他费 备注 币种
str
"20220721 600269 赣粤高速 卖出 股息入账 1...
"20220718 204007 GC007 卖出 拆出质押购...
"20220718 "="002462""" 嘉事堂 卖出 ...
"20220718 600408 安泰集团 买入 证券买入 0...
"20220718 600648 外高桥 买入 证券买入 09...
...
"20220711 204007 GC007 卖出 质押回购拆...
"20220711 601992 金隅集团 买入 证券买入 0...
"20220711 600894 广日股份 买入 证券买入 0...
"20220711 601077 渝农商行 买入 证券买入 0...
"20220707 204007 GC007 卖出 拆出质押购...

正确版

```
[1]: import polars as pl

[6]: pl.read_csv("stock_trades/202207-湘财.xls",encoding="gb18030",separator="\t")

[6]: shape: (17, 16)
```

发生日期	证券代码	证券名称	买卖标志	业务名称	成交时间	成交数量	成交价格	成交金额	发生金额	手续费	印花税	过户费	其他费	备注	币种
i64	str	str	str	str	str	str	f64	f64	f64	str	str	str	str	str	str
20220721	"600269"	"赣粤高速"	"卖出"	"股息入账"	"16:00:00"	"="0.00""	3.6	4884.0	4884.0	"="0.00""	"="0.00""	"="0.00""	"="0.00""	"股息入账:赣粤高速600269;权益股数:40700;"	"人民币"
20220718	"204007"	"GC007"	"卖出"	"拆出质押购回"	"19:03:27"	"-580.00"	1.675	58000.0	58018.63	"="0.00""	"="0.00""	"="0.00""	"="0.00""	"融券购回:18.63实际占款天数：7-888880"	"人民币"
20220718	"="002462""	"嘉事堂"	"卖出"	"证券卖出"	"09:38:10"	"-10400.00"	13.2062	137344.0	137184.67	"21.98"	"137.35"	"1.38"	"="0.00""	"证券卖出"	"人民币"
20220718	"600408"	"安泰集团"	"买入"	"证券买入"	"09:44:52"	"47000.00"	3.19	149930.0	-149955.5	"23.99"	"="0.00""	"1.51"	"="0.00""	"证券买入"	"人民币"
20220718	"600648"	"外高桥"	"买入"	"证券买入"	"09:44:31"	"11900.00"	12.6066	150019.0	-150044.49	"24.00"	"="0.00""	"1.49"	"="0.00""	"证券买入"	"人民币"
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
20220711	"204007"	"GC007"	"卖出"	"质押回购拆出"	"09:39:27"	"580.00"	1.675	58000.0	-58002.9	"2.90"	"="0.00""	"="0.00""	"="0.00""	"融券回购购回日:20220718	"人民

取 hang

```
[25]: df[1]

[25]: shape: (1, 16)
```

发生日期	证券代码	证券名称	买卖标志	业务名称	成交时间	成交数量	成交价格	成交金额	发生金额	手续费	印花税	过户费	其他费	备注	币种
i64	str	str	str	str	str	str	f64	f64	f64	str	str	str	str	str	str
20220718	"204007"	"GC007"	"卖出"	"拆出质押购回"	"19:03:27"	"-580.00"	1.675	58000.0	58018.63	"="0.00""	"="0.00""	"="0.00""	"="0.00""	"融券购回:18.63实际占款天数：7-888880"	"人民币"

```
[26]: df[-1]

[26]: shape: (1, 16)
```

发生日期	证券代码	证券名称	买卖标志	业务名称	成交时间	成交数量	成交价格	成交金额	发生金额	手续费	印花税	过户费	其他费	备注	币种
i64	str	str	str	str	str	str	f64	f64	f64	str	str	str	str	str	str
20220707	"204007"	"GC007"	"卖出"	"拆出质押购回"	"19:17:51"	"-7580.00"	2.46	758000.0	758357.61	"="0.00""	"="0.00""	"="0.00""	"="0.00""	"融券购回:357.61实际占款天数：7-888880"	"人民币"

```
[27]: df[3:5]

[27]: shape: (2, 16)
```

发生日期	证券代码	证券名称	买卖标志	业务名称	成交时间	成交数量	成交价格	成交金额	发生金额	手续费	印花税	过户费	其他费	备注	币种
i64	str	str	str	str	str	str	f64	f64	f64	str	str	str	str	str	str

```
[33]: df[:,["证券名称","成交金额"]]
```

```
[33]: shape: (17, 2)
```

证券名称	成交金额
str	f64
"赣粤高速"	4884.0
"GC007"	58000.0
"嘉事堂"	137344.0
"安泰集团"	149930.0
"外高桥"	150019.0
...	...
"GC007"	58000.0
"金隅集团"	139860.0
"广日股份"	139956.0
"渝农商行"	139795.0
"GC007"	758000.0

○

掌握以下几个 检查 polars.DataFrame 对象时常用的属性 (attributes) / 方法 (methods)：

形状/维度：df.shape、df.height、df.width、df.is_empty()

数据模式/架构/类型：df.schema、df.columns、df.dtypes

数据提取/切片：df[...] (取行/取列/取多行/取多列)、df[..., ...] (行列双向限制)、df.row()、df.rows()、df.get_column()、df.to_series()

数据概览/描述：df.glimpse()、df.head()、df.tail()、df.sample()、df.describe()、df.null_count()

转换/导出：df.to_pandas()、df.to_arrow()、df.to_dicts()

$ 币种 <str> '人民币', '人民币', '人民币', '人民币', '人民币', '人民币', '人民

[47]:
```
df[:,"证券名称":"成交数量"].sample(5)
```

[47]: shape: (5, 5)

证券名称	买卖标志	业务名称	成交时间	成交数量
str	str	str	str	str
"华夏银行"	"买入"	"证券买入"	"09:42:51"	"30000.00"
"赣粤高速"	"买入"	"证券买入"	"09:43:38"	"40700.00"
"嘉事堂"	"卖出"	"证券卖出"	"09:38:10"	"-10400.00"
"GC007"	"卖出"	"拆出质押购回"	"19:17:51"	"-7580.00"
"渝农商行"	"卖出"	"证券卖出"	"09:38:30"	"-38300.00"

○

polars.DataFrame 单独的一列数据是 polars.Series；检查 polars.Series 对象 (命名为 s) 有以下常用的属性 (attributes) / 方法 (methods)：

基本属性：s.name、s.dtype、s.shape、s.len()

数据提取/切片：s[...] (取单值/取多值)

数据概览/描述：s.unique()、s.value_counts()、s.describe()、s.null_count()

转换/导出：s.to_pandas()、s.to_arrow()、s.to_list()

```
成交价格: [[3.6,1.675,13.2062,3.19,12.6066,...,1.675,2.59,6.54,3.6
成交金额: [[4884,58000,137344,149930,150019,...,58000,139860,13995
发生金额: [[4884,58018.63,137184.67,-149955.5,-150044.49,...,-5800
...
```

```
[53]: s = df.to_series()

[54]: s.name

[54]: '发生日期'

[55]: s.shape

[55]: (17,)

[56]: s.len()

[56]: 17

[59]: s[3:5]

[59]: shape: (2,)
```

发生日期
i64
20220718
20220718

清洗和转换

```
$ 币种 ´ <str> '人民币', '人民币', '人民币', '人民币', '人民币', '人民币', '人民币', '人民币', '人民币', '人民币'

[67]: pl.read_csv("stock_trades/202207-湘财.xls", encoding="gb18030",separator="\t",infer_schema=False)

[67]: shape: (17, 16)
```

发生日期	证券代码	证券名称	买卖标志	业务名称	成交时间	成交数量	成交价格	成交金额	发生金额	手续费	印花税	过户费	其他费	备注	币种
str	str	str	str	str	str	str	str	str	str	str	str	str	str	str	str
"20220721"	"600269"	"赣粤高速"	"卖出"	"股息入账"	"16:00:00"	"="0.00""	"3.6000"	"4884.00"	"4884.00"	"="0.00""	"="0.00""	"="0.00""	"="0.00""	"股息入账:赣粤高速600269; 权益股数:40700;"	"人民币"
"20220718"	"204007"	"GC007"	"卖出"	"拆出质押购回"	"19:03:27"	"-580.00"	"1.6750"	"58000.00"	"58018.63"	"="0.00""	"="0.00""	"="0.00""	"="0.00""	"融券购回:18.63实际占款天数：7-888880"	"人民币"
"20220718"	"="002462""	"嘉事堂"	"卖出"	"证券卖出"	"09:38:10"	"-10400.00"	"13.2062"	"137344.00"	"137184.67"	"21.98"	"137.35"	"1.38"	"="0.00""	"证券卖出"	"人民币"

polars.DataFrame 的计算，都是整列进行的向量化 (vectorized) 计算，利用 CPU 的 SIMD 指令能够极大地提升计算效率

DataFrame.with_columns() 方法用来添加/修改列

DataFrame.select() 方法用来挑选/计算列

DataFrame.filter() 方法用来过滤行 (计算为 True 的行将被保留)

她们接受的参数都是 Polars Expression —— 存储的是算法逻辑，而非具体数值

Polars 之所以功能强大，就是因为设计有 大量的 Expressions，可以组合使用

构建 Polars Expression 的起点，一般都是通过 polars.col 选择一列或多列，也可以通过 selectors 挑选符合条件的列，然后利用 . 运算符进行链式调用，或者用其他各种运算符组合计算出更进一步的、复杂的 Expression

[76]:
```
df = pl.read_csv("stock_trades/202207-湘财.xls", encoding="gb18030",separator="\t",infer_schema=False)
df = df.with_columns(
    pl.col("发生日期").str.to_date("%Y%m%d"),
    pl.col("证券代码").str.strip_prefix("=").str.strip_chars('"'),
)
df[:,"证券代码"].unique().to_list()
```

[76]:
```
['600269',
 '600015',
 '600408',
 '600648',
 '600894',
 '204007',
 '601077',
 '000900',
 '601992',
 '002462']
```

[95]:
```
df = pl.read_csv(
    "stock_trades/202207-湘财.xls", encoding="gb18030",separator="\t",infer_schema=False,
)
df = df.with_columns(
    pl.col("发生日期").str.to_date("%Y%m%d"),
    pl.col("证券代码").str.strip_prefix("=").str.strip_chars('"'),
)
df = df.filter(pl.col("业务名称").is_not_null())
df[:,"业务名称"].value_counts()
```

[95]: shape: (5, 2)

业务名称	count
str	u32
"证券卖出"	4
"质押回购拆出"	1
"拆出质押购回"	2
"证券买入"	9
"股息入账"	1

[99]:
```
df = pl.read_csv(
    "stock_trades/202207-湘财.xls", encoding="gb18030",separator="\t",infer_schema=False,
)
df = df.with_columns(
    pl.col("发生日期").str.to_date("%Y%m%d"),
    pl.col("证券代码").str.strip_prefix("=").str.strip_chars('"'),
)
df = df.filter(
    pl.col("业务名称").is_in(["证券买入", "证券卖出"]),
)
df
```

[99]: shape: (13, 16)

发生日期	证券代码	证券名称	买卖标志	业务名称	成交时间	成交数量	成交价格	成交金额	发生金额	手续费	印花税	过户费	其他费	备注	币种
date	str	str	str	str	str	str	str	str	str	str	str	str	str	str	str
2022-07-18	"002462"	"嘉事堂"	"卖出"	"证券卖出"	"09:38:10"	"-10400.00"	"13.2062"	"137344.00"	"137184.67"	"21.98"	"137.35"	"1.38"	"=""0.00"""	"证券卖出"	"人民币"
2022-07-18	"600408"	"安泰集团"	"买入"	"证券买入"	"09:44:52"	"47000.00"	"3.1900"	"149930.00"	"-149955.50"	"23.99"	"=""0.00"""	"1.51"	"=""0.00"""	"证券买入"	"人民币"
2022-07-18	"600648"	"外高桥"	"买入"	"证券买入"	"09:44:31"	"11900.00"	"12.6066"	"150019.00"	"-150044.49"	"24.00"	"=""0.00"""	"1.49"	"=""0.00"""	"证券买入"	"人民币"
2022-07-18	"600269"	"赣粤高速"	"买入"	"证券买入"	"09:43:38"	"40700.00"	"3.6900"	"150183.00"	"-150208.53"	"24.03"	"=""0.00"""	"1.50"	"=""0.00"""	"证券买入"	"人民币"

[102]:
```
df = pl.read_csv(
    "stock_trades/202207-湘财.xls", encoding="gb18030",separator="\t",infer_schema=False,
)
df = df.with_columns(
    pl.selectors.all().str.strip_prefix("=").str.strip_chars('"'),
).with_columns(
    pl.col("发生日期").str.to_date("%Y%m%d"),
    pl.col("证券代码").str.strip_prefix("=").str.strip_chars('"'),
    pl.col("成交时间").str.to_time(),
    pl.col("成交数量").cast(pl.Float64),
)
df = df.filter(
    pl.col("业务名称").is_in(["证券买入", "证券卖出"]),
)
df
```

[102]: shape: (13, 16)

发生日期	证券代码	证券名称	买卖标志	业务名称	成交时间	成交数量	成交价格	成交金额	发生金额	手续费	印花税	过户费	其他费	备注	币种
date	str	str	str	str	time	f64	str	str	str	str	str	str	str	str	str
2022-07-18	"002462"	"嘉事堂"	"卖出"	"证券卖出"	09:38:10	-10400.0	"13.2062"	"137344.00"	"137184.67"	"21.98"	"137.35"	"1.38"	"0.00"	"证券卖出"	"人民币"
2022-07-18	"600408"	"安泰集团"	"买入"	"证券买入"	09:44:52	47000.0	"3.1900"	"149930.00"	"-149955.50"	"23.99"	"0.00"	"1.51"	"0.00"	"证券买入"	"人民币"

[104]:
```
df = pl.read_csv(
    "stock_trades/202207-湘财.xls", encoding="gb18030",separator="\t",infer_schema=False,
)
df = df.with_columns(
    pl.selectors.all().str.strip_prefix("=").str.strip_chars('"'),
).with_columns(
    pl.col("发生日期").str.to_date("%Y%m%d"),
    pl.col("证券代码").str.strip_prefix("=").str.strip_chars('"'),
    pl.col("成交时间").str.to_time(),
    pl.col("成交数量","成交价格","成交金额","发生金额","手续费","印花税","过户费","其他费").cast(pl.Float64),
)
df = df.filter(
    pl.col("业务名称").is_in(["证券买入", "证券卖出"]),
)
df
```

[104]: shape: (13, 16)

发生日期	证券代码	证券名称	买卖标志	业务名称	成交时间	成交数量	成交价格	成交金额	发生金额	手续费	印花税	过户费	其他费	备注	币种
date	str	str	str	str	time	f64	f64	f64	f64	f64	f64	f64	f64	str	str
2022-07-18	"002462"	"嘉事堂"	"卖出"	"证券卖出"	09:38:10	-10400.0	13.2062	137344.0	137184.67	21.98	137.35	1.38	0.0	"证券卖出"	"人民币"
2022-07-18	"600408"	"安泰集团"	"买入"	"证券买入"	09:44:52	47000.0	3.19	149930.0	-149955.5	23.99	0.0	1.51	0.0	"证券买入"	"人民币"
2022-07-18	"600648"	"外高桥"	"买入"	"证券买入"	09:44:31	11900.0	12.6066	150019.0	-150044.49	24.0	0.0	1.49	0.0	"证券买入"	"人民币"
2022-07-18	"600269"	"赣粤高速"	"买入"	"证券买入"	09:43:38	40700.0	3.69	150183.0	-150208.53	24.03	0.0	1.5	0.0	"证券买入"	"人民币"
2022-07-18	"600015"	"华夏银行"	"买入"	"证券买入"	09:42:51	30000.0	5.07	152100.0	-152125.86	24.34	0.0	1.52	0.0	"证券买入"	"人民币"
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
2022-07-11	"002462"	"嘉事堂"	"买入"	"证券买入"	09:38:16	10400.0	13.51	140504.0	-140526.48	22.48	0.0	1.41	0.0	"证券买入"	"人民币"

```
)
```

[32]:
```
df = pl.read_excel(
    "stock_trades/202305-海通普通.xlsx",
    schema_overrides={
        "成交日期":pl.String,
        "成交时间":pl.String,
    },
)
df.filter(pl.col("成交时间") !="").with_columns(
    pl.col("成交日期").str.to_date("%Y%m%d"),
    pl.col("成交时间").str.to_time("%H:%M:%S"),
)
```

[32]: shape: (8, 14)

证券代码	证券名称	成交日期	成交时间	成交数量	成交价格	成交金额	发生金额	操作	手续费	印花税	过户费	其他费	备注
str	str	date	time	i64	f64	f64	f64	str	f64	i64	f64	i64	str
"799999"	"指定登记"	2023-05-24	15:00:00	0	0.0	0.0	0.0	"指"	0.0	0	0.0	0	"指定登记指定交易"
"600626"	"申达股份"	2023-05-24	10:06:14	14800	3.37	49876.0	-49881.48	"买"	4.99	0	0.49	0	"申达股份证券买入"
"600178"	"东安动力"	2023-05-24	09:59:17	16400	6.07	99548.0	-99558.97	"买"	9.95	0	1.02	0	"东安动力证券买入"
"603002"	"宏昌电子"	2023-05-24	09:54:48	9800	5.06	49588.0	-49593.46	"买"	4.96	0	0.5	0	"宏昌电子证券买入"
"131810"	"R-001"	2023-05-23	10:05:49	1500	1.865	150000.0	-150001.5	"卖"	1.5	0	0.0	0	"到期日[20230524]，利息[7.66]，金额[1500...
"300107"	"建新股份"	2023-05-23	10:01:57	10000	5.0	50000.0	-50005.0	"买"	5.0	0	0.0	0	"建新股份证券买入"
"002224"	"三力士"	2023-05-23	09:58:07	10800	4.59	49572.0	-49576.96	"买"	4.96	0	0.0	0	"三力士证券买入"
"799998"	"指定撤销"	2023-05-22	15:00:00	0	0.0	0.0	0.0	"撤"	0.0	0	0.0	0	"指定撤销撤消指定"

[139]:
```
df = pl.read_excel(
    "stock_trades/202305-海通普通.xlsx",
    schema_overrides={
        "成交日期":pl.String,
        "成交时间":pl.String,
    },
)
df.filter(pl.col("成交时间") !="").filter(
    pl.col("操作").is_in(["买","卖"])
).filter(
    (~pl.col("证券代码").str.starts_with("204")) &
    (~pl.col("证券代码").str.starts_with("1318"))
).with_columns(
    pl.col("成交日期").str.to_date("%Y%m%d"),
    pl.col("成交时间").str.to_time("%H:%M:%S"),
)
```

[139]: shape: (5, 14)

证券代码	证券名称	成交日期	成交时间	成交数量	成交价格	成交金额	发生金额	操作	手续费	印花税	过户费	其他费	备注
str	str	date	time	i64	f64	f64	f64	str	f64	i64	f64	i64	str
"600626"	"申达股份"	2023-05-24	10:06:14	14800	3.37	49876.0	-49881.48	"买"	4.99	0	0.49	0	"申达股份证券买入"
"600178"	"东安动力"	2023-05-24	09:59:17	16400	6.07	99548.0	-99558.97	"买"	9.95	0	1.02	0	"东安动力证券买入"
"603002"	"宏昌电子"	2023-05-24	09:54:48	9800	5.06	49588.0	-49593.46	"买"	4.96	0	0.5	0	"宏昌电子证券买入"
"300107"	"建新股份"	2023-05-23	10:01:57	10000	5.0	50000.0	-50005.0	"买"	5.0	0	0.0	0	"建新股份证券买入"
"002224"	"三力士"	2023-05-23	09:58:07	10800	4.59	49572.0	-49576.96	"买"	4.96	0	0.0	0	"三力士证券买入"

```
)
```

[220]:
```
pl.concat([d1, d2, d3])
```

[220]: shape: (363, 14)

券商	交易日期	交易时间	证券代码	证券名称	买卖标志	成交价格	成交数量	成交金额	手续费	印花税	过户费	其他费	发生金额
str	date	time	str	str	str	f64	f64	f64	f64	f64	f64	f64	f64
"湘财"	2022-07-18	09:38:10	"002462"	"嘉事堂"	"卖出"	13.2062	10400.0	137344.0	21.98	137.35	1.38	0.0	137184.67
"湘财"	2022-07-18	09:44:52	"600408"	"安泰集团"	"买入"	3.19	47000.0	149930.0	23.99	0.0	1.51	0.0	-149955.5
"湘财"	2022-07-18	09:44:31	"600648"	"外高桥"	"买入"	12.6066	11900.0	150019.0	24.0	0.0	1.49	0.0	-150044.49
"湘财"	2022-07-18	09:43:38	"600269"	"赣粤高速"	"买入"	3.69	40700.0	150183.0	24.03	0.0	1.5	0.0	-150208.53
"湘财"	2022-07-18	09:42:51	"600015"	"华夏银行"	"买入"	5.07	30000.0	152100.0	24.34	0.0	1.52	0.0	-152125.86
...	...	...	...	...	...	...	...	...	...	...	...	...	...
"海通两融"	2023-10-18	09:46:15	"300464"	"星徽股份"	"卖出"	5.74	16100.0	92414.0	8.82	46.21	0.0	0.0	92358.97
"海通两融"	2023-10-18	09:55:41	"002661"	"克明食品"	"买入"	9.42	8500.0	80072.0	7.64	0.0	0.0	0.0	-80079.64
"海通两融"	2023-10-09	09:48:02	"002753"	"永东股份"	"买入"	7.02	14200.0	99684.0	9.51	0.0	0.0	0.0	-99693.51
"海通两融"	2023-10-09	09:45:18	"000698"	"沈阳化工"	"卖出"	4.053	23800.0	96460.0	9.2	48.24	0.0	0.0	96402.56
"海通两融"	2023-10-09	09:44:40	"605288"	"凯迪股份"	"卖出"	40.838	2500.0	102094.0	9.74	51.05	1.0	0.0	102032.21

○

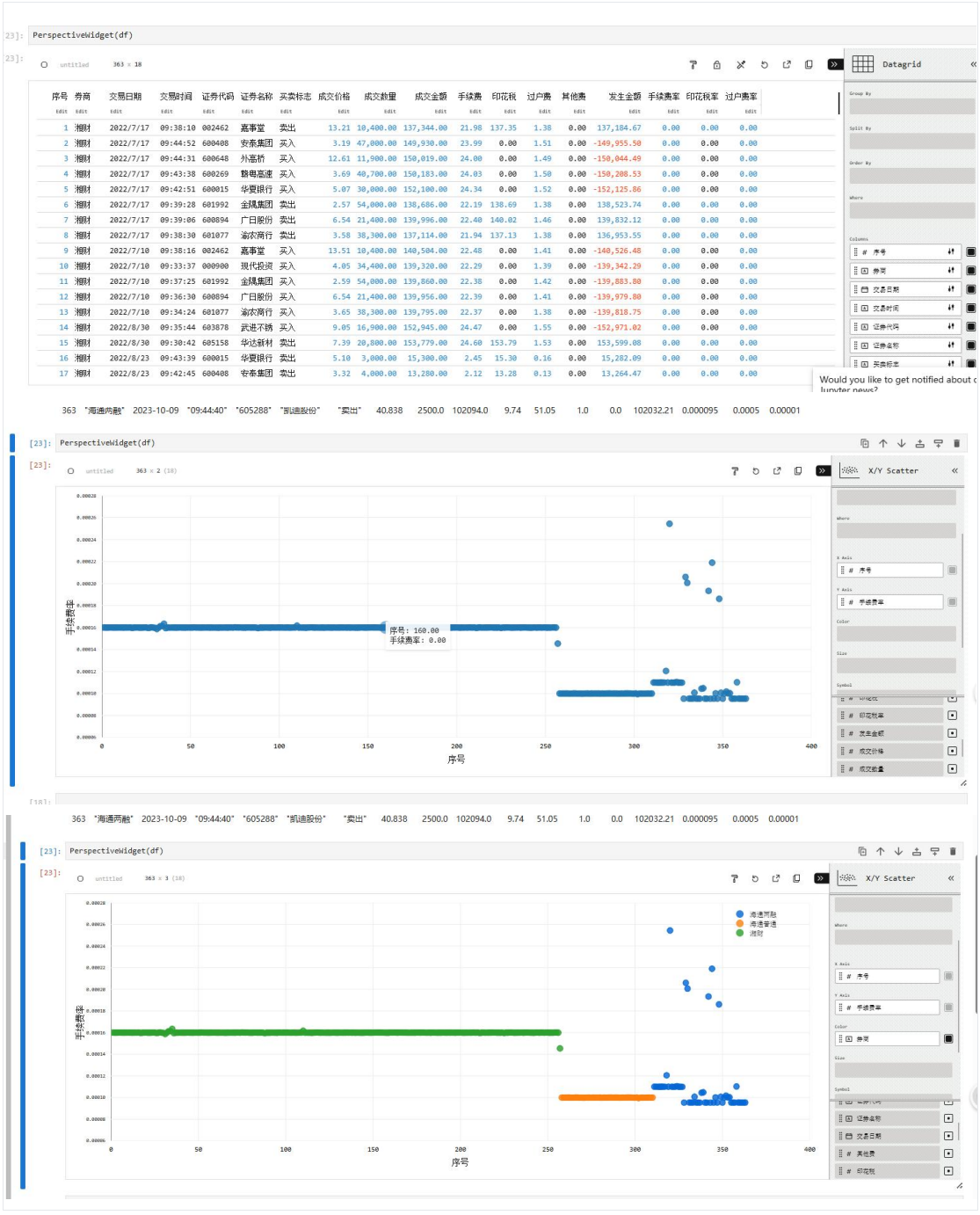
首先，最简单地，我们可以通过计算和做图检查每一笔交易的费率

分别计算每笔交易的 手续费率、印花税率、过户费率

为每笔交易生成一个序号 (index)

用 Perspective 的 X/Y Scatter 视图，将序号 (index) 作为 X 坐标，某项费率作为 Y 坐标，不同券商区分颜色，也可以根据 买卖标志 分别做图，其他感兴趣的值可以显示在悬浮框 (tooltip) 里

确认每项费率是怎么计算的，检查券商有没有多收，比较哪个券商的费率更优惠



○

第二，这期间的交割单涉及多支股票，我们可以计算每支股票是否都已完成平仓(首次买入算开仓，全部卖光算平仓)，或者说，有哪些股票截至期末仍未完全平仓

使用 df.group_by().agg() 进行分组汇总

group_by() 分组和 agg() 汇总都接受一个/多个 Expression 作为输入

可以按 证券代码 或/和 证券名称 分组

可以按 成交数量 汇总，首先需要根据 买卖标志 决定正负号，然后汇总求和，命名为 结余数量

最后，按照 结余数量 排序：结余数量 为负的，是在交割单期初之前就有持仓；结余数量 为负的，是在交割单期末之后仍有持仓

为简化起见，我们把 结余数量 为负的，或者说交割单期初之前就有持仓的股票，从 df 里剔除

使用 DataFrame.filter() 选择出准备剔除的股票

使用 DataFrame.join(how="anti") 进行基于匹配的剔除

[28]:
```
df.group_by("证券代码","证券名称").agg(pl.len(),pl.col("买卖标志"))
```

[28]: shape: (152, 4)

证券代码	证券名称	len	买卖标志
str	str	u32	list[str]
"600844"	"丹化科技"	2	["买入", "卖出"]
"002672"	"东江环保"	2	["买入", "卖出"]
"000655"	"金岭矿业"	2	["买入", "卖出"]
"002381"	"双箭股份"	3	["买入", "买入", "卖出"]
"300429"	"强力新材"	4	["买入", "卖出", … "卖出"]
…	…	…	…
"600735"	"新华锦"	2	["买入", "卖出"]
"600729"	"重庆百货"	2	["买入", "卖出"]
"603002"	"宏昌电子"	2	["买入", "卖出"]
"600727"	"鲁北化工"	2	["买入", "卖出"]
"605258"	"协和电子"	2	["买入", "卖出"]

```
[45]: d1= df.join(
          df.group_by("证券代码","证券名称")
          .agg(
              结余数量 =
                  pl.when(pl.col("买卖标志") == "卖出")
                  .then(-pl.col("成交数量"))
                  .when(pl.col("买卖标志") == "买入")
                  .then(pl.col("成交数量"))
                  .sum()
          )
          .filter(pl.col("结余数量") < 0 ),
          on ="证券代码",
          how ="anti",
      )
```

```
[48]: d1
```

[48]: shape: (358, 18)

序号	券商	交易日期	交易时间	证券代码	证券名称	买卖标志	成交价格	成交数量	成交金额	手续费	印花税	过户费	其他费	发生金额	手续费率	印花税率	过户费率
u32	str	date	str	str	str	str	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64
1	"湘财"	2022-07-11	"09:33:37"	"000900"	"现代投资"	"买入"	4.05	34400.0	139320.0	22.29	0.0	1.39	0.0	-139342.29	0.00016	0.0	0.00001
2	"湘财"	2022-07-11	"09:34:24"	"601077"	"渝农商行"	"买入"	3.65	38300.0	139795.0	22.37	0.0	1.38	0.0	-139818.75	0.00016	0.0	0.00001
3	"湘财"	2022-07-11	"09:36:30"	"600894"	"广日股份"	"买入"	6.54	21400.0	139956.0	22.39	0.0	1.41	0.0	-139979.8	0.00016	0.0	0.00001
4	"湘财"	2022-07-11	"09:37:25"	"601992"	"金隅集团"	"买入"	2.59	54000.0	139860.0	22.38	0.0	1.42	0.0	-139883.8	0.00016	0.0	0.00001
5	"湘财"	2022-07-11	"09:38:16"	"002462"	"嘉事堂"	"买入"	13.51	10400.0	140504.0	22.48	0.0	1.41	0.0	-140526.48	0.00016	0.0	0.00001
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
359	"海通两融"	2023-10-31	"09:31:53"	"002956"	"西麦食品"	"卖出"	14.13	5000.0	70650.0	6.74	35.35	0.0	0.0	70607.91	0.000095	0.0005	0.0
360	"海通两融"	2023-10-31	"09:39:57"	"603214"	"爱婴室"	"买入"	15.84	3100.0	49104.0	5.0	0.0	0.51	0.0	-49109.51	0.000102	0.0	0.00001
361	"海通两融"	2023-10-31	"09:40:55"	"300132"	"青松股份"	"买入"	5.21	9600.0	50016.0	5.0	0.0	0.0	0.0	-50021.0	0.0001	0.0	0.0
362	"海通两融"	2023-10-31	"09:43:13"	"002492"	"恒基达鑫"	"买入"	5.91	8400.0	49644.0	5.0	0.0	0.0	0.0	-49649.0	0.000101	0.0	0.0
363	"海通两融"	2023-10-31	"09:44:45"	"002111"	"威海广泰"	"买入"	9.24	5400.0	49896.0	5.0	0.0	0.0	0.0	-49901.0	0.0001	0.0	0.0

○

第三，我们可以计算和做图观察这段期间累计的股票持仓数量变化情况 (注意，不同股票的持股数量相加是没有意义的，为简化起见，我们现在暂不考虑股价和市值)，以及每天持有股票数量 (支数) 的变化情况

现在要考虑的是每一天、每一支股票的动态情况及其汇总，所以我们先计算时间范围 (k1)，再计算股票范围 (k2)，再计算二者的笛卡尔积 (k1.join(k2, how="cross"))

用得到的笛卡尔积 (k) 与交割单数据做左匹配 (left join)，即保留全部的 k，未匹配到的行赋空值 (null)

对于 成交数量 列，买入 取正值，卖出 取负值，空值 (null) 取值 0 (when)，由此衍生计算一列 结余数量，在每支股票范围内 (over)，沿交易日期计算其累计的 (cum_sum) 成交数量 作为 结余数量

把 结余数量 为 0 的行全部剔除，便于统计每天的持股

用 Perspective 的 Y Area 视图，横轴 Group By 设定为 交易日期，纵轴 Y Axis 可以查看每日总的 结余数量 (注意，其实是不可加的)，也可以查看每日的持股数量 (即持有的 证券代码 的数量)

"688660"

[60]:
```
K= k1.join(k2, how= "cross")
```

[69]:
```
k.join(d1,left_on= ["日期","证券代码"], right_on= ["交易日期","证券代码"],how= "left").sort("日期","证券代码").with_columns(
    结余数量 =
            pl.when(pl.col("买卖标志") == "卖出")
            .then(-pl.col("成交数量"))
            .when(pl.col("买卖标志") == "买入")
            .then(pl.col("成交数量"))
            .otherwise(0).cum_sum().over("证券代码")
)
```

[69]: shape: (72_671, 19)

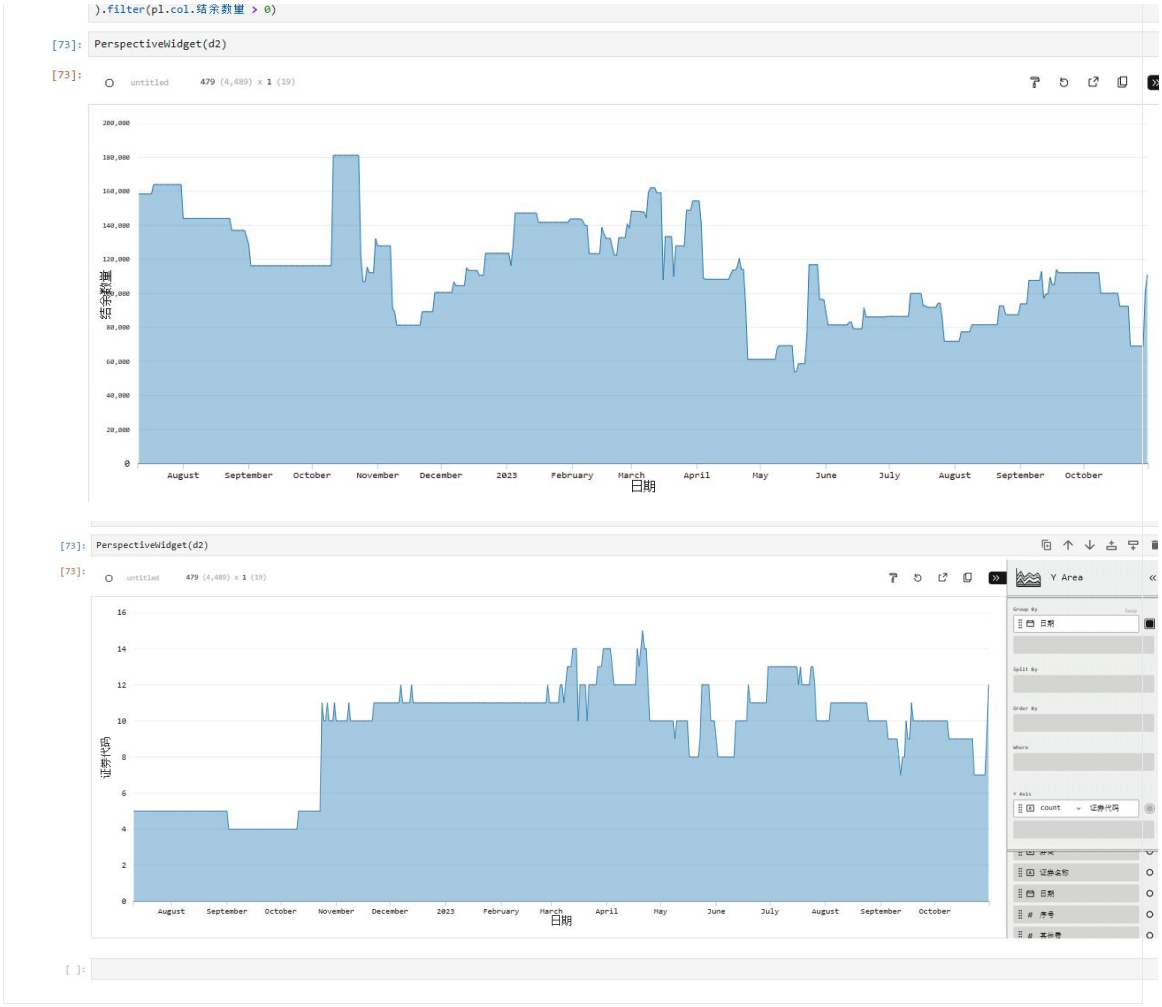
日期	证券代码	序号	券商	交易时间	证券名称	买卖标志	成交价格	成交数量	成交金额	手续费	印花税	过户费	其他费	发生金额	手续费率	印花税率	过户费率	结余数量
date	str	u32	str	str	str	str	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64
2022-07-11	"000096"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	0.0
2022-07-11	"000532"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	0.0
2022-07-11	"000559"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	0.0
2022-07-11	"000599"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	0.0
2022-07-11	"000655"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2023-10-31	"688299"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	0.0
2023-10-31	"688321"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	0.0
2023-10-31	"688360"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	0.0
2023-10-31	"688393"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	0.0
2023-10-31	"688660"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	20200.0

[62]:
```
k= k1.join(k2, how= "cross")
```

[71]:
```
k.join(d1,left_on= ["日期","证券代码"], right_on= ["交易日期","证券代码"],how= "left").sort("日期","证券代码").with_columns(
    结余数量 =
            pl.when(pl.col("买卖标志") == "卖出")
            .then(-pl.col("成交数量"))
            .when(pl.col("买卖标志") == "买入")
            .then(pl.col("成交数量"))
            .otherwise(0).cum_sum().over("证券代码")
).filter(pl.col.结余数量 > 0)
```

[71]: shape: (4_489, 19)

日期	证券代码	序号	券商	交易时间	证券名称	买卖标志	成交价格	成交数量	成交金额	手续费	印花税	过户费	其他费	发生金额	手续费率	印花税率	过户费率	结余数量
date	str	u32	str	str	str	str	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64	f64
2022-07-11	"000900"	1	"湘财"	"09:33:37"	"现代投资"	"买入"	4.05	34400.0	139320.0	22.29	0.0	1.39	0.0	-139342.29	0.00016	0.0	0.00001	34400.0
2022-07-11	"002462"	5	"湘财"	"09:38:16"	"嘉事堂"	"买入"	13.51	10400.0	140504.0	22.48	0.0	1.41	0.0	-140526.48	0.00016	0.0	0.00001	10400.0
2022-07-11	"600894"	3	"湘财"	"09:36:30"	"广日股份"	"买入"	6.54	21400.0	139956.0	22.39	0.0	1.41	0.0	-139979.8	0.00016	0.0	0.00001	21400.0
2022-07-11	"601077"	2	"湘财"	"09:34:24"	"渝农商行"	"买入"	3.65	38300.0	139795.0	22.37	0.0	1.38	0.0	-139818.75	0.00016	0.0	0.00001	38300.0
2022-07-11	"601992"	4	"湘财"	"09:37:25"	"金隅集团"	"买入"	2.59	54000.0	139860.0	22.38	0.0	1.42	0.0	-139883.8	0.00016	0.0	0.00001	54000.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2023-10-31	"300132"	361	"海通两融"	"09:40:55"	"青松股份"	"买入"	5.21	9600.0	50016.0	5.0	0.0	0.0	0.0	-50021.0	0.0001	0.0	0.0	9600.0
2023-10-31	"300215"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	20000.0
2023-10-31	"600525"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	8500.0
2023-10-31	"603214"	360	"海通两融"	"09:39:57"	"爱婴室"	"买入"	15.84	3100.0	49104.0	5.0	0.0	0.51	0.0	-49109.51	0.000102	0.0	0.00001	3100.0
2023-10-31	"688660"	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	null	20200.0



第四，我们可以从 Tushare 获取行情数据，与每日持股数据匹配，由此计算每日持股的市值的动态变化，并能够由此计算每日投资收益率，并与股市指数的每日收益率 (基准收益) 相对照

调用 Tushare 的 daily 接口，需要指定股票代码和起止时间，但交割单里的证券代码 (如 002462) 不含交易所代码，与 Tushare 的编码不符，因此需要先根据沪深交易所的编码规律转换出含交易所代码的证券代码 (如 002462.SZ)

以恰当的形式向 Tushare 接口传入参数，获取每支股票在时间范围内的每日 开盘价、收盘价、最高价、最低价、成交量 数据，股票数量较多，需要循环调取，可以使用 tqdm 软件包显示进度条，全部获取后合并，保存为 daily.parquet 文件

将行情数据与交割单数据匹配，检查每一行的 成交价格 是否落在 最高价 与 最低价 之间，检查每一行的交割单 成交数量 占股票 成交量 的比例，以防交割单数据造假

将行情数据与每日持股数据匹配，将非交易日缺失的价格数据填充为最近数值 (fill_null().over())，按 交易日期 分组，汇总计算每日总的 持股市值，用 Perspective 的 X/Y Line 视图观察每日 持股市值 的动态变化

要计算投资者的投资收益率，除了要知道股票交易情况外，还要知道总的资金情况，即本金。假设在期初，投资者的本金是 100 万；先根据交割单计算每日总的 发生金额；再生成一个 转账金额 列，只有第一行取值 100 万，其余行全部为零；再把每日的 发生金额 和 转账金额 加在一起，沿日期做累加，就得到每日的 现金余额；再把每日 现金余额 和每日 持股市值 加在一起，就得到每日的 总资产；用 Perspective 的 X/Y Line 视图观察每日 总资产 的动态变化

调用 Tushare 的 index_daily 接口，获取 “沪深 300 指数” (000300.SH) 在起止时间之内的每日 涨跌幅 数据，这是每日的 净收益率 (net rate of return)，除以 100 (因为单位是 %) 再加 1 转换为每日的 总收益率 (gross rate of return)，沿日期做累乘，就得到投资指数的每日 累计收益率 (cumulative return)，再与本金 100 万相乘，就能够得到如果全部投资于 “沪深 300 指数” 每日的总资产变化情况，命名为 沪深 300；用 Perspective 的 X/Y Line 视图观察每日 沪深 300 的动态变化

最后，如果想在 Perspective 的 X/Y Line 视图里 同时 显示两种 (甚至更多种) 投资的动态变化，还需要做一种数据变形，因为现在这两个曲线的数值分别在两个列里 (总资产 和 沪深 300)，属于 宽形 (wide form)，如果有更多曲线，就要加更多的列进去 (变宽)，会改变表格的架构 (schema)，不利于存储和分析。我们需要把数据变为 长形 (long form)，值都放在同一列 (value column) (在 Perspective 里设置为 Y Axis)，列名放在另一列 (variable column) 用于区分 (在 Perspective 里设置为 Split By)。从 “长形” 变为 “宽形” 叫做 pivot，从 “宽形” 变为 “长形” 叫做 unpivot/melt

[76]:
```
hq= pro.daily(ts_code="002462.SZ",start_date="20220711", end_date="20231031")
```

[77]:
```
hq
```

[77]:

	ts_code	trade_date	open	high	low	close	pre_close	change	pct_chg	vol	amount
0	002462.SZ	20231031	14.75	14.90	14.59	14.70	14.75	-0.05	-0.3390	65859.96	96984.271
1	002462.SZ	20231030	13.88	14.89	13.88	14.75	13.93	0.82	5.8866	123932.16	180119.372
2	002462.SZ	20231027	13.70	13.98	13.51	13.93	13.64	0.29	2.1261	35782.00	49386.168
3	002462.SZ	20231026	13.49	13.68	13.40	13.64	13.62	0.02	0.1468	19215.00	26005.866
4	002462.SZ	20231025	13.65	13.77	13.58	13.62	13.67	-0.05	-0.3658	18484.00	25274.163
...	...	...	...	...	...	...	...	...	...	...	...
313	002462.SZ	20220715	13.61	13.66	13.12	13.13	13.59	-0.46	-3.3848	32967.65	44114.064
314	002462.SZ	20220714	13.54	13.75	13.50	13.59	13.54	0.05	0.3693	21967.00	29851.164
315	002462.SZ	20220713	13.55	13.63	13.39	13.54	13.61	-0.07	-0.5143	22793.00	30714.624
316	002462.SZ	20220712	13.65	13.69	13.41	13.61	13.65	-0.04	-0.2930	29679.00	40146.310
317	002462.SZ	20220711	13.20	13.96	13.07	13.65	13.18	0.47	3.5660	62827.00	85869.110

318 rows × 11 columns

[79]:
```
hq= pro.daily(ts_code="002462.SZ",start_date=format(start_date,"%Y%m%d"), end_date=format(end_date,"%Y%m%d"))
hq= pl.from_pandas(hq)
hq
```

[79]: shape: (318, 11)

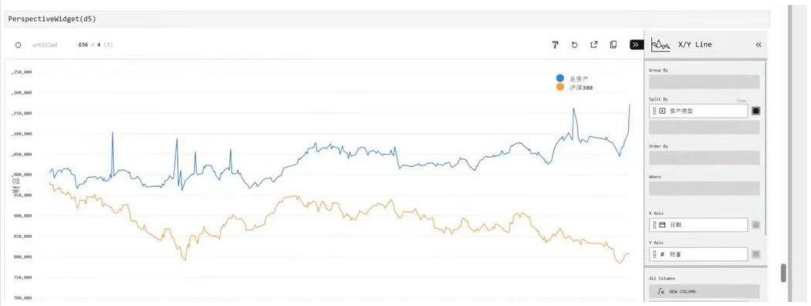
ts_code	trade_date	open	high	low	close	pre_close	change	pct_chg	vol	amount
str	str	f64	f64	f64	f64	f64	f64	f64	f64	f64
"002462.SZ"	"20231031"	14.75	14.9	14.59	14.7	14.75	-0.05	-0.339	65859.96	96984.271
"002462.SZ"	"20231030"	13.88	14.89	13.88	14.75	13.93	0.82	5.8866	123932.16	180119.372
"002462.SZ"	"20231027"	13.7	13.98	13.51	13.93	13.64	0.29	2.1261	35782.0	49386.168
"002462.SZ"	"20231026"	13.49	13.68	13.4	13.64	13.62	0.02	0.1468	19215.0	26005.866
"002462.SZ"	"20231025"	13.65	13.77	13.58	13.62	13.67	-0.05	-0.3658	18484.0	25274.163
...	...	...	...	...	...	...	...	...	...	...
"002462.SZ"	"20220715"	13.61	13.66	13.12	13.13	13.59	-0.46	-3.3848	32967.65	44114.064
"002462.SZ"	"20220714"	13.54	13.75	13.5	13.59	13.54	0.05	0.3693	21967.0	29851.164
"002462.SZ"	"20220713"	13.55	13.63	13.39	13.54	13.61	-0.07	-0.5143	22793.0	30714.624
"002462.SZ"	"20220712"	13.65	13.69	13.41	13.61	13.65	-0.04	-0.293	29679.0	40146.31
"002462.SZ"	"20220711"	13.2	13.96	13.07	13.65	13.18	0.47	3.566	62827.0	85869.11

[81]:
```
ts_codes = d1.select(
    证券代码=(
        pl.when(pl.col("证券代码").str.head(1).is_in(["0","3"]))
        .then(pl.format("{}.SZ",pl.col("证券代码")))
        .when(pl.col("证券代码").str.head(1) == "6")
        .then(pl.format("{}.SH",pl.col("证券代码")))
    )
).to_series().unique().sort().to_list()
```

[83]:
```
from tqdm.notebook import tqdm
```

[84]:
```
hq= [pro.daily(ts_code=ts_code,start_date=format(start_date,"%Y%m%d"), end_date=format(end_date,"%Y%m%d")) for ts_code in tqdm(ts_codes)]
```

100% 149/149 [00:07<00:00, 21.72it/s]


PerspectiveWidget(d5)
X/Y Line