第 8 周学习笔记

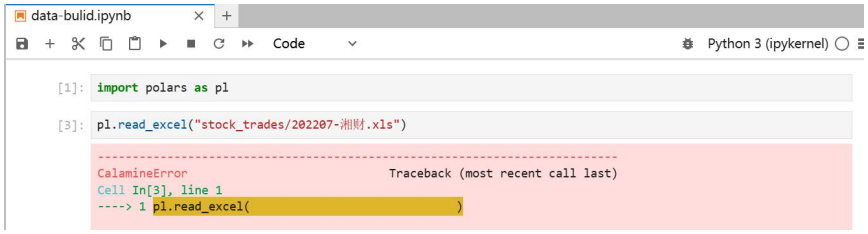
1. 新建 environment.yml 文件

```yaml
name: week08
channels:
  - conda-forge
dependencies:
  - python=3.12
  - wat-inspector
  - xlrd
  - openpyxl
  - fastexcel
  - xlsxwriter
  - pandas
  - pyarrow
  - polars
  - jupyterlab
  - ipywidgets
  - jupyter-ruff
  - pip
  - pip:
    - perspective-python
    - tushare
```

2. 下载案例数据

```
(week08) Administrator@MICROSO-J56DDR4 MINGW64 ~/repo/week08 (main)
$ curl -O https://raw.gitcode.com/cueb-fintech/courses/blobs/8e70be13d8672dd6856
72f6624896ad5320d1110/stock_trades.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 77002    0 77002    0     0  74705      0 --:--:--  0:00:01 --:--:-- 74904
(week08) Administrator@MICROSO-J56DDR4 MINGW64 ~/repo/week08 (main)
$ unzip stock_trades.zip
Archive:  stock_trades.zip
   creating: stock_trades/
  inflating: stock_trades/202207-湘财.xls
  inflating: stock_trades/202208-湘财.xls
  inflating: stock_trades/202209-湘财.xls
  inflating: stock_trades/202210-湘财.xls
  inflating: stock_trades/202211-湘财.xls
  inflating: stock_trades/202212-湘财.xls
  inflating: stock_trades/202301-湘财.xls
```

3. 尝试使用 polars.read_excel() 函数读取名称为 202207-湘财.xls 的文件，观察报错



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

stock_trades > 202207-湘财.xls

	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	
1	20220721	600269	赣粤高速	卖出	股息入账	16:00:00	="0.00"	3.6000	4884
2	20220718	204007	GC007	卖出	拆出质押购回	19:03:27	-580.00	1.6750	580
3	20220718	="002462"	嘉事堂	卖出	证券卖出	09:38:10	-10400.00	13.2	
4	20220718	600408	安泰集团	买入	证券买入	09:44:52	47000.00	3.1900	
5	20220718	600648	外高桥	买入	证券买入	09:44:31	11900.00	12.6066	1500
6	20220718	600269	赣粤高速	买入	证券买入	09:43:38	40700.00	3.6900	
7	20220718	600015	华夏银行	买入	证券买入	09:42:51	30000.00	5.0700	152100.00
8	20220718	601992	金隅集团	卖出	证券卖出	09:39:28	-54000.00	2.5683	138686.00
9	20220718	600894	广日股份	卖出	证券卖出	09:39:06	-21400.00	6.5419	139996.00
10	20220718	601077	渝农商行	卖出	证券卖出	09:38:30	-38300.00	3.5800	137114.00
11	20220711	="002462"	嘉事堂	买入	证券买入	09:38:16	10400.00	13.5100	140504.00
12	20220711	="000900"	现代投资	买入	证券买入	09:33:37	34400.00	4.0500	139320.00
13	20220711	204007	GC007	卖出	质押回购拆出	09:39:27	580.00	1.6750	58000.00
14	20220711	601992	金隅集团	买入	证券买入	09:37:25	54000.00	2.5900	139860.00
15	20220711	600894	广日股份	买入	证券买入	09:36:30	21400.00	6.5400	139956.00
16	20220711	601077	渝农商行	买入	证券买入	09:34:24	38300.00	3.6500	139795.00
17	20220707	204007	GC007	卖出	拆出质押购回	19:17:51	-7580.00	2.4600	758000.00

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

```
[1]: import polars as pl
```

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

```
[8]: df
```

[8]: 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
i64	str	str	str	str	str	str	f64	f64	f64	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.(
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.(

```
[9]: df.shape
[9]: (17, 16)
[10]: df.height
[10]: 17
[11]: df.width
[11]: 16
[12]: df.is_empty()
[12]: False
[13]: df.schema
[13]: Schema([('发生日期', Inte
               ('证券代码', Stri
               ('证券名称', Stri
               ('买卖标志', Stri
```

[27]:
```
s = df.to_series()
```

Type Markdown and LaTeX: α^2

[28]:
```
s.name
```

[28]: '发生日期'

[29]:
```
s.dtype
```

[29]: Int64

[30]:
```
s.shape
```

[30]: (17,)

[31]:
```
s.len()
```

[31]: 17

[32]:
```
s[:5]
```

[32]: shape: (5,)

发生日期
i64

[34]:
```
pl.read_csv("stock_trades/202207-湘财.xls",encoding="gb18030",separator="\t",infer_schema=False)
```

[34]: 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
str	str	str	str	str	str	str	str	str	str	st

4. polars.DataFrame 的计算

[25]:
```
df = pl.read_csv("stock_trades/202207-湘财.xls",encoding="gb18030"
```

[35]:
```
df = pl.DataFrame(
    {
        "foo": ["asdj1", "jjask", "hasna"],
    }
)
df.with_columns(foo2=pl.col("foo").str.strip_chars("a"))
```

```
df = pl.read_csv("stock_trades/202207-湘财.xl
df.with_columns(
    pl.col("发生日期").str.to_date('%Y%m%d'),
    pl.col("证券代码").str.strip_prefix("=").
)
df[:,"证券代码"].unique().to_list()
```

```
['204007',
 '601992',
 '600269',
 '600648',
 '="000900"',
 '="002462"',
 '600408',
```

```
df = df.filter(
    pl.col("业务名称").is_in(["证券买入", "证券卖出"]),
)
df[:,"业务名称"].value_counts()
```

shape: (2, 2)

业务名称	count
str	u32
"证券买入"	9
"证券卖出"	4

```
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("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
```

shape: (13, 16)

5. 命名为 df，检查行列数 (shape)，检查架构 (dtype)，逐列检查值 (value_counts)，发现一些问题，进行清洗

```
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")!="")
    &(pl.col("操作").is_in(["买","卖"]))
    &(~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(),
)
```

```python
def read_df_海通普通(f: str | Path) -> pl.DataFrame:
    df = pl.read_excel(f,
        schema_overrides={
            "成交日期":pl.String,
            "成交时间":pl.String,
            "成交数量":pl.Float64,
            "成交金额":pl.Float64,
            "印花税":pl.Float64,
            "其他费":pl.Float64,
        },
    )
    df = df.filter(
        (pl.col("成交时间")!="")
        &(pl.col("操作").is_in(["买","卖"]))
        &(~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(),
    )
    return df
```

```python
df = [read_df_海通普通(p) for p in Path("stock_trades/").glob("*-海通普通.xlsx")]
df = pl.concat(df)
d2 = df.with_columns(
```

```python
df = [read_df_海通普通(p) for p in Path("stock_trades/").glob("*-海通两融.xlsx")]
df = pl.concat(df)
d3 = df.with_columns(
    券商=pl.lit("海通两融"),
)
d3
```

```python
d1 = d1.select(
    券商=pl.col("券商"),
    交易日期=pl.col("发生日期"),
    交易时间=pl.col("成交时间"),
    证券代码=pl.col("证券代码"),
    证券名称=pl.col("证券名称"),
    买卖标志=pl.col("业务名称").replace({"证券卖出":"卖出", "证券买入":"买入"}),
    成交价格=pl.col("成交价格"),
    成交数量=pl.col("成交数量").abs(),
    成交金额=pl.col("成交金额"),
    手续费=pl.col("手续费"),
    印花税=pl.col("印花税"),
    过户费=pl.col("过户费"),
    其他费=pl.col("其他费"),
    发生金额=pl.col("发生金额"),
)
```

```
d2 = d2.select(
    券商=pl.col("券商"),
    交易日期=pl.col("成交日期"),
    交易时间=pl.col("成交时间"),
    证券代码=pl.col("证券代码"),
    证券名称=pl.col("证券名称"),
    买卖标志=pl.col("操作").replace({"卖":"卖出", "买":"买入"}),
    成交价格=pl.col("成交价格"),
    成交数量=pl.col("成交数量").abs(),
    成交金额=pl.col("成交金额"),
    手续费=pl.col("手续费"),
    印花税=pl.col("印花税"),
    过户费=pl.col("过户费"),
    其他费=pl.col("其他费"),
    发生金额=pl.col("发生金额"),
)
```

```
d3 = d3.select(
    券商=pl.col("券商"),
    交易日期=pl.col("成交日期"),
    交易时间=pl.col("成交时间"),
    证券代码=pl.col("证券代码"),
    证券名称=pl.col("证券名称"),
    买卖标志=pl.col("操作").replace({"卖":"卖出", "买":"买入"}),
    成交价格=pl.col("成交价格"),
    成交数量=pl.col("成交数量").abs(),
    成交金额=pl.col("成交金额"),
    手续费=pl.col("手续费"),
    印花税=pl.col("印花税"),
    过户费=pl.col("过户费"),
    其他费=pl.col("其他费"),
    发生金额=pl.col("发生金额"),
)
```

```
df = pl.concat([d1, d2, d3])
```

```
df.with_columns(
    成交金额2=pl.col("成交价格") * pl.col("成交数量"),
).with_columns(
    成交金额D=pl.col("成交金额") - pl.col("成交金额2"),
).with_columns(
    发生金额D=(
        pl.col("发生金额")
        - (
            pl.when(pl.col("买卖标志") == "买入")
            .then(-pl.col("成交金额"))
            .when(pl.col("买卖标志") == "卖出")
            .then(pl.col("成交金额"))
            - pl.col("手续费")
            - pl.col("印花税")
            - pl.col("过户费")
            - pl.col("其他费")
        )
    ).round(4)
).sort("发生金额D")
```

```
df.write_parquet("stock_trades.parquet")
```

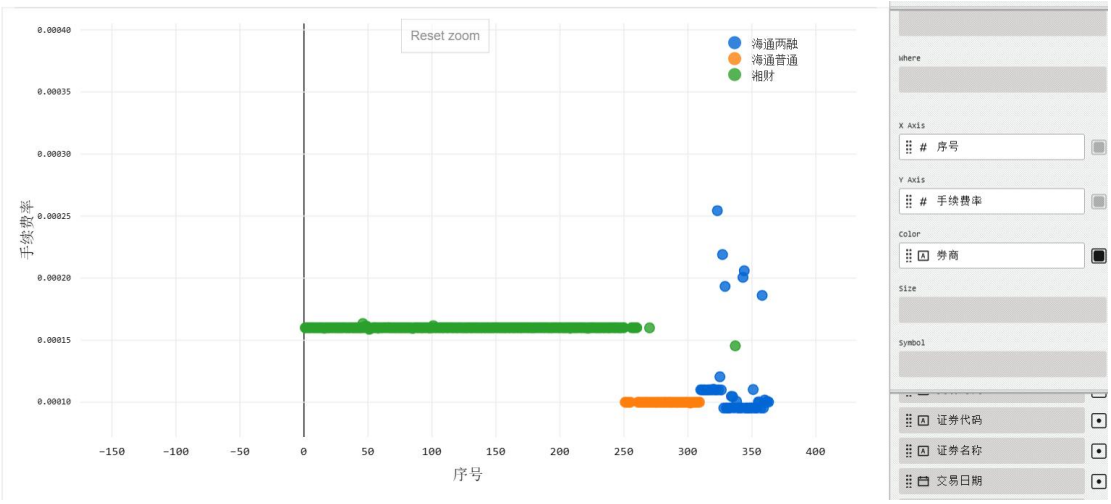
```
df.write_csv("stock_trades.csv")
```

6. 数据计算

```
import polars as pl
from perspective.widget import PerspectiveWidget
```

```
df = pl.read_parquet("stock_trades.parquet")
df = df.sort("交易日期", "交易时间", "证券代码")
df = df.with_columns(
    pl.col("交易时间").cast(pl.String),
    手续费率=pl.col("手续费") / pl.col("成交金额"),
    印花税率=pl.col("印花税") / pl.col("成交金额"),
    过户费率=pl.col("过户费") / pl.col("成交金额"),
)
df = df.with_row_index("序号", 1)
```

```
PerspectiveWidget(df)
```




```
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",
)
```

```
start_date = df["交易日期"].min()
start_date
```

```
datetime.date(2022, 7, 11)
```

```
end_date = df["交易日期"].max()
end_date
```

```
datetime.date(2023, 10, 31)
```

```
k1 = pl.select(
    交易日期=pl.date_range(start_date, end_date)
)
```

```
k2 = df["证券代码"].unique().sort().to_frame()
```

```
k = k1.join(k2, how="cross")
k
```

```
            .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()
)
```

```
from tqdm.notebook import tqdm
```

```
hq = [
    pl.from_pandas(
        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)
]
```