

第7周 数据可视化与交互（初级）学习笔记

一、核心概念

1. 可视化工具选择

- **JupyterLab**: 基于浏览器的交互式开发环境，支持代码、Markdown、图表混合编排
- **Perspective**: 交互式数据可视化库，支持动态数据过滤/分组/聚合
- 优势: 跨平台、可扩展、适合数据探索

二、环境配置

environment.yml

文件示例

name: week07

channels:

- conda-forge

dependencies:

- python=3.12

- jupyterlab

- pip

- pip:

- tushare

- perspective-python

- polars

- 执行命令:

- conda env create # 首次创建环境

- conda env update # 更新依赖

- jupyter lab # 启动服务

三、JupyterLab 核心操作

模式/操作	快捷键
切换命令模式	`Esc`
切换编辑模式	`Enter`
插入单元格	`a` (上) / `b` (下)
合并单元格	`Shift+M`
运行单元格	`Shift+Enter`
Markdown 渲染	`Ctrl+Enter`

四、Tushare 数据获取

```
import tushare as ts
```

```
# 设置 Token (仅需运行一次)
```

```
ts.set_token("your_token_here")
```

```
# 获取 IPO 数据示例
pro = ts.pro_api()
df = pro.new_share()
df.to_parquet("new_share.parquet")
```

```

- 文件格式对比：

|      | Parquet      | CSV      |
|------|--------------|----------|
| 存储效率 | 列式存储，高压缩率    | 文本格式，体积大 |
| 读取速度 | 快速（支持列裁剪）    | 较慢       |
| 适用场景 | 大数据量/结构化数据分析 | 小型数据集    |

## 五、Perspective 可视化

```
import polars as pl
from perspective import PerspectiveWidget

读取数据
d1 = pl.read_parquet("new_share.parquet")

日期类型转换
d1 = d1.with_columns(
 pl.col("ipo_date").str.strptime(pl.Date, "%Y%m%d")
)

创建交互式视图
widget = PerspectiveWidget(d1)
widget
```

```

常用视图配置：

1. 直方图：

```
widget.configure({
    "group_by": ["bucket_pe"], # 分桶后的市盈率
    "aggregates": {"count": "count"},
    "split_by": [],
    "columns": ["bucket_pe", "count"]
})
```

2. 时间序列分析：

```
widget.aggregate({
    "funds": "avg",
    "pe": "avg"
```

```
})  
...
```

3. 多维度对比:

```
widget.split_by(["exchange"]) # 按交易所拆分  
...
```

六、关键学习点

1. 交互式工作流: 通过配置-导出-代码化的方式实现手动探索到自动化分析的过渡

2. 数据理解: 结合可视化结果与历史事件 (如注册制改革) 深化分析

3. 可视化选择原则:

- 分类数据 → 条形图/Treemap
- 时间序列 → 折线图
- 相关性分析 → 散点图

注: 所有视图配置均可通过界面操作生成 JSON, 使用`json.loads()`解析后复用