

练习题01

低折扣订单明细统计 + 商品类别销量 Top3 横向条形图

【实操任务描述】

电商运营部门希望从订单明细中分析两件事：

1. 低折扣 (`discount < 0.95`) 订单主要集中在哪些商品类别 (`category`)，并统计各类别的订单明细条数；
2. 统计各商品类别的总销量 (`quantity`)，并可视化展示销量最高的 Top3 类别。

请你使用提供的 `order_items.csv` 与 `products.csv` 完成以下练习任务。

【任务要求】

步骤 1：导入库并设置中文显示

使用 `pandas` 进行数据处理，使用 `matplotlib` 绘图，并完成中文字体与负号显示配置。

```
import pandas as pd
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
```

步骤 2：读取数据

使用 `read_csv()` 分别读取 `order_items.csv` 与 `products.csv`。

```
items = pd.read_csv("order_items.csv")
products = pd.read_csv("products.csv")
```

步骤 3：合并数据补充商品信息

以 `product_id` 为键，对两张表执行内连接 (`inner join`)，在订单明细表中补充：

- 商品类别：`category`
- 商品单价：`unit_price`

```
items = items.merge(
    products[["product_id", "category", "unit_price"]],
    on="product_id",
    how="inner"
)
```

步骤 4: 统计低折扣订单中各类别的订单明细条数 (重点语法题)

筛选出折扣小于 0.95 的订单记录, 并按 `category` 分组统计每个类别对应的订单明细条数。
要求结果字段命名为: `order_count`。

```
items.query("discount < 0.95").groupby("category").size().reset_index(name="order_count")
```

步骤 5: 按商品类别统计总销量

以 `category` 分组, 对 `quantity` 求和得到各类别的总销量, 并将汇总字段命名为 `total_quantity`。

```
category_sales = (  
    items.groupby("category")["quantity"]  
        .sum()  
        .reset_index(name="total_quantity")  
)
```

步骤 6: 排序并截取 Top3 类别

将类别销量按 `total_quantity` 从高到低排序, 截取销量最高的前 3 个类别, 用于后续可视化展示。

```
top3 = category_sales.sort_values(  
    by="total_quantity",  
    ascending=False  
) .head(3)
```

步骤 7: 绘制横向条形图并完善图表元素

绘制“各商品类别销量 Top3 横向条形图”, 要求如下:

- 使用 `plt.figure(figsize=(8, 5))` 设置画布大小
- 使用 `plt.barh()` 绘制横向条形图:
 - 横轴为销量 `total_quantity`
 - 纵轴为商品类别 `category`
- 设置图表标题为“各商品类别销量 Top3”
- 设置 X 轴标签为“销售量 (件)”, Y 轴标签为“类别”
- 使用 `invert_yaxis()` 将销量最高的类别显示在最上方
- 在每个条形右侧添加对应的销量数值标注
- 使用 `tight_layout()` 优化布局后, 调用 `show()` 显示图表

```
plt.figure(figsize=(8, 5))  
  
plt.barh(top3["category"], top3["total_quantity"])
```

```
plt.title("各商品类别销量 Top3")
plt.xlabel("销售量 (件)")
plt.ylabel("类别")

plt.gca().invert_yaxis()

for index, value in enumerate(top3["total_quantity"]):
    plt.text(value + 1, index, str(value), va="center")

plt.tight_layout()
plt.show()
```