

Module 6: Basic Math & Data Filtering

Segmenting Open Datasets & Resolving Calculations in

Jupyter

Part 1: Workspace Setup

Connecting Jupyter to Live Open Repositories

The Target Online Retail Stream

UCI Online Dataset Profile

A classic multi-national transaction matrix mapping product sales. Since this file represents raw real-world data, it contains errors and missing data clusters.

Remote Endpoint URL:

```
https://raw.githubusercontent.com/datagy/data/main/online\_retail\_subset.csv
```

Logical Slicing Vectors

Data filtering is essential to subset raw rows before applying mathematical aggregations. In this module, we partition data across three primary pillars:

- **Temporal Ranges:** Isolate specific calendar dates.
- **Geographical Slices:** Segment by country codes.
- **Product Categories:** Sift based on ID groups.

Interactive Analysis in Jupyter



State Preservation Benefits

Jupyter Notebook is the preferred workspace for mathematical calculations due to its cell-by-cell execution model.

Unlike standard scripts, Jupyter retains memory-allocated DataFrames, allowing you to filter outliers, calculate averages, and slice geography iteratively without re-importing the raw source dataset.

Part 2: Logical Slicing Pipelines

Purging Inconsistencies and Outliers Natively in Memory

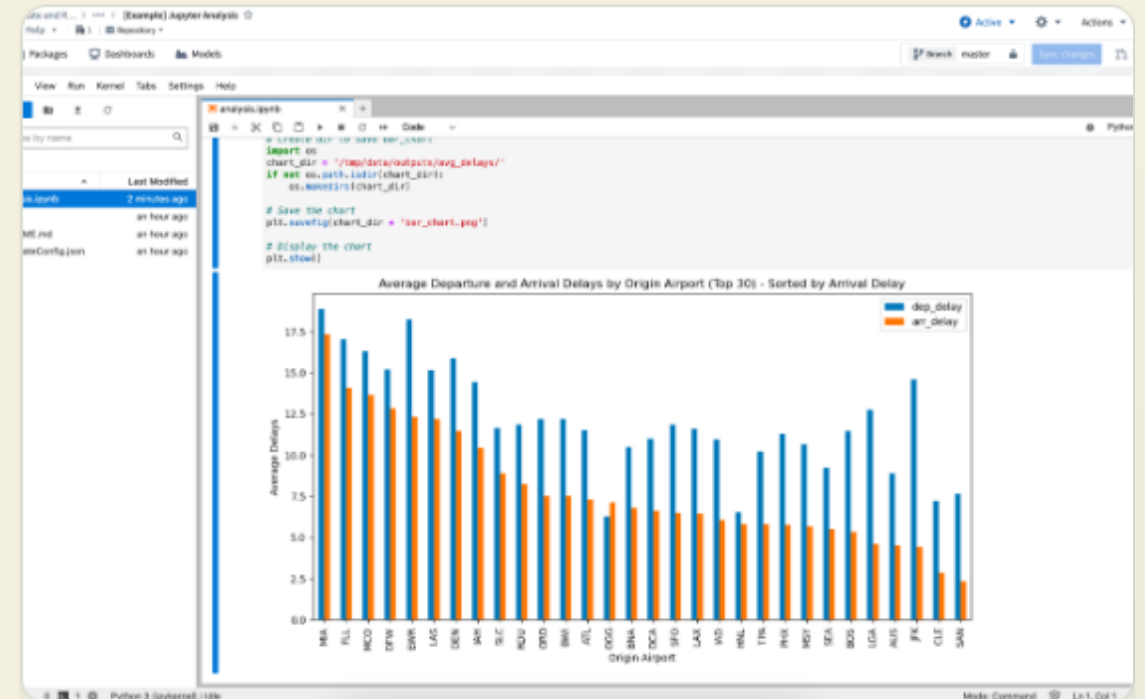
Purging Cancellations & Returns

Filtering Out Negative Values

In retail files, cancellations are often stored as negative quantities. If unchecked, they distort the true mean. We write simple filters to drop them:

```
# Exclude transactional cancellations
df_clean = df_retail[
    (df_retail['Quantity'] > 0) &
    (df_retail['UnitPrice'] > 0)
]
```

By chaining these boolean conditions inside a single query expression, we create a pristine operational base.



Multidimensional In-Memory Slicing



Geographic Slices

Isolate local regions to calculate regional metrics:
`df[df['Country'] == 'United Kingdom']`



Temporal Windows

Restrict transactional rows to specific timeline segments:
`df[df['InvoiceDate'] >= '2011-01-01']`



Category Filters

Isolate target product segments easily:
`df[df['StockCode'].isin(['85123A', '22423'])]`

Part 3: Descriptive Math

Quantifying Performance and Business Averages

Calculating Central Tendency

$$A = \Sigma / n$$

Group Average Logic

Automating Averages via Vector Math

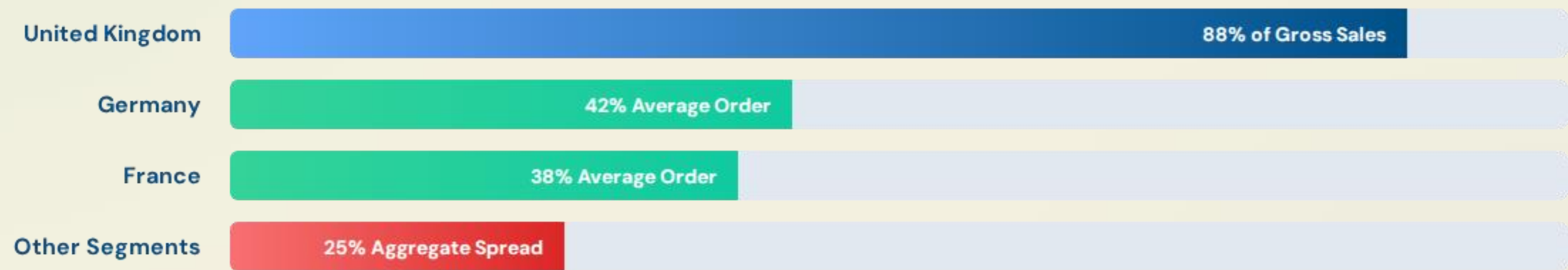
Rather than looping over each transaction to compute a sum, Pandas relies on underlying C-optimized vector processing to aggregate data points instantly.

$$\text{Group Average} = \frac{\sum \text{Subset Values}}{\text{Total Row Count (n)}}$$

Descriptive Math Commands

Calculation	Jupyter Vector Command	Strategic Business Application
Mean (Average)	<code>df['Revenue'].mean()</code>	Evaluate standard baseline order values.
Median	<code>df['Revenue'].median()</code>	Identify typical sales values unaffected by outlier bulk orders.
Sum	<code>df['Revenue'].sum()</code>	Calculate gross consolidated sales across specified cohorts.
Standard Deviation	<code>df['Revenue'].std()</code>	Measure variance and consistency in seasonal transaction sizes.

Revenue Distribution across Key Regions



Segmenting the Online Retail Subset by geographical columns reveals that the UK commands top gross revenue, while European continental clusters lead in average basket value.

Questions?

Module 6: Basic Math & Data Filtering Complete

www.anaconda.com | support@anaconda.com

Image Sources



<https://dataplatfom.cloud.ibm.com/docs/api/content/wsj/getting-started/images/gs-notebook-cell01.png?context=cpdaas&locale=en>

Source: dataplatfom.cloud.ibm.com



<https://build.palantir.com/images/a1792be0a3f6d65d.jpg>

Source: build.palantir.com