

Module 4: Conversational Analysis & Lab 1

Interacting Directly with Public Datasets Using PandasAI • Ingestion to Clean Table

Participant Name: _____

Date: _____

Section 1: The End-to-End Operational Flow

This lab implements a full pipeline that transitions from messy raw online data ingestion to natural language AI analytics. Follow the sequential structure below to build and test your pipeline in a Jupyter Notebook:

1. Ingest

Load the messy remote CSV directly from the GitHub target public stream url.

2. Scan

Run diagnostic checks like `.isnull().sum()` and `.duplicated().sum()` to isolate data anomalies.

3. Purge

Remove invalid rows, strip whitespace, and normalize columns to correct datatypes.

4. Chat

Wrap the pristine dataframe inside a PandasAI `SmartDataframe` layer and execute direct natural language queries.

Section 2: Lab Code Sandbox Workspace

Initialize and execute the following complete framework inside an empty cell in your notebook:

```
import pandas as pd
from pandasai import SmartDataframe

# Step 1: Ingest public live repository stream
url = "https://raw.githubusercontent.com/vizzuhq/ipyvizzu/main/docs/examples/stories/sales/sales.csv"
df_raw = pd.read_csv(url)

print("--- STEP 2: RAW SCAN DIAGNOSTICS ---")
print("Null Values Count:")
print(df_raw.isnull().sum())
print("Duplicate Rows:", df_raw.duplicated().sum())

# Step 3: Purge and Clean Anomaly Elements
df_clean = df_raw.copy()
# (Example standardization step: stripping whitespace from string vectors if present)
for col in df_clean.select_dtypes(include='object').columns:
    df_clean[col] = df_clean[col].str.strip()

print("Pristine Data Table Structural Types:")
print(df_clean.dtypes)

# Step 4: Chat via PandasAI Layer
# Note: Ensure your local API token key is configured prior to instantiation
# smart_df = SmartDataframe(df_clean, config={"llm": your_llm_instance})
```

Section 3: Verification & Analytical Questions

Run the code above in your workspace, review the terminal summaries, and fill out your analytical observations below.

Question 1: Parsing Data Dimensions

After executing Step 1 and 2, write down the baseline column features identified by the inspection. What is the precise metric volume of duplicate rows or empty cells detected?

Question 2: Formulating Conversational Aggregations

Write the exact code syntax required to query your `df` via PandasAI to extract the average revenue generated per product line without manually typing a `.groupby()` or `.mean()` chain:

Question 3: Automated Charting Logic

Write out the text prompt string you would pass to `df.chat()` to automatically generate and output an interactive bar chart plotting total sales across regions:

Question 4: Operational Stream Optimization

What are the key organizational advantages of reading live remote public URLs into an AI-assisted dataframe pipeline over maintaining static localized CSV spreadsheets on individual hard drives?
