Homework 1: Introduction to Data Processing

Points: 20 | Due: See WebCampus for deadline

Author: Richard Young, Ph.D. | UNLV Lee Business School

Compute: CPU (free tier)

Learning Objectives

1. **Connect** to real-world data sources using Python
2. **Load and explore** datasets using Pandas
3. **Assess data quality** by identifying missing values, duplicates, and outliers
4. **Discover and communicate** an interesting finding from data

---

Why This Matters for Business

Data-Driven Decisions: Amazon processes over 1 million transactions per hour. Before any ML model can predict inventory needs or recommend products, data engineers must load, clean, and understand this data—exactly what you'll practice here.

Quality Control: Netflix discovered that 80% of their data science team's time was spent on data preparation, not modeling. A single missing value in the wrong place crashed their recommendation engine for hours. Data quality assessment isn't glamorous, but it's essential.

Insight Discovery: Target famously identified a pregnant teenager before her father knew—by finding patterns in purchase data. The ability to find "interesting findings" in data is what separates analysts who generate reports from analysts who drive strategy.

Cost of Bad Data: IBM estimates poor data quality costs the U.S. economy $3.1 trillion annually. Learning to identify data issues early prevents costly downstream errors.

Grading

Component	Points	Effort	What We're Looking For
Data Connection	5	*	Load data from any source into a DataFrame
Data Exploration	5	*	Answer questions about dataset structure
Quality Assessment	5	**	Analyze missing values, duplicates, outliers
Interesting Finding	5	**	One insight with evidence and business relevance
Total	**20**		

Effort Key: * Straightforward | ** Requires thinking | *** Challenge

---

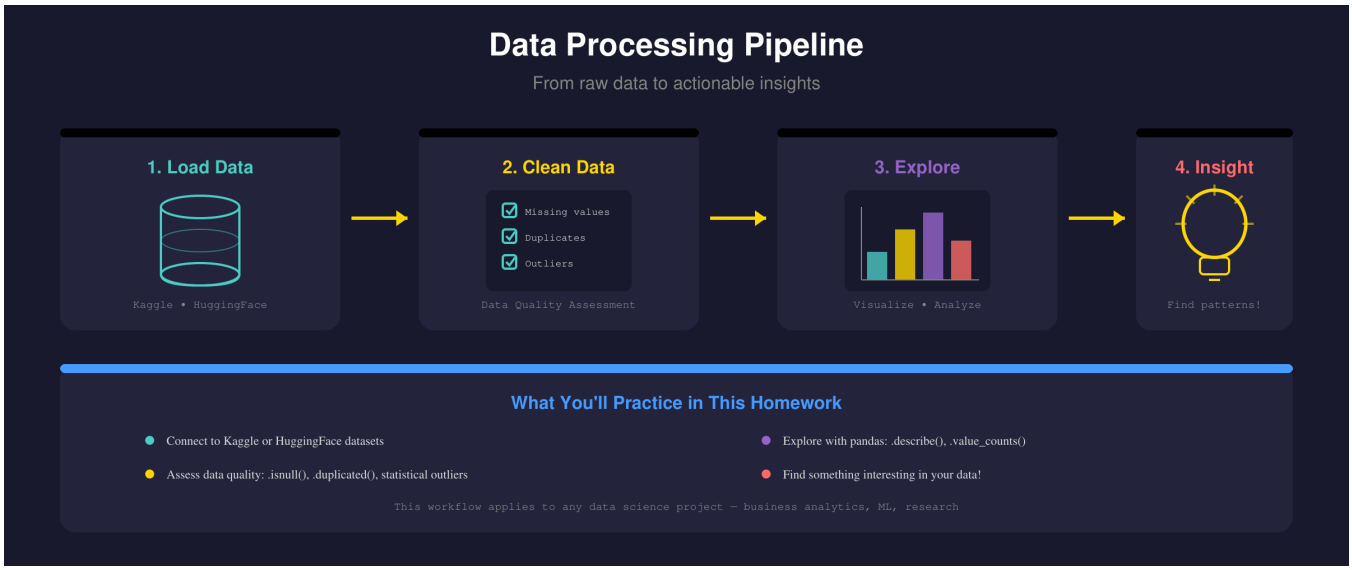

Figure 1: Data Pipeline Flow

Instructions

1. Open `MIS769_HW1_Data_Processing.ipynb` in Google Colab
2. Choose a data source (HuggingFace, Kaggle, or your own CSV)
3. Run the exploration code and answer questions in markdown cells
4. Complete the data quality assessment
5. Find something interesting and write up your insight

What Your Output Should Look Like

Data Loading:

```
✅ DATASET LOADED SUCCESSFULLY
============================================================
Shape: (50,000 rows × 12 columns)
Memory usage: 4.6 MB
Source: HuggingFace - imdb
```

Data Exploration:

```
📋 COLUMN SUMMARY
------------------------------------------------------------
Column          Type      Non-Null    Example
------------------------------------------------------------
review_id       int64     50,000      12847
text            object    50,000      "This movie was..."
```

```
rating          float64   49,234      4.5
date            datetime  50,000      2024-03-15
category        object    48,892      "Drama"
```

Quality Assessment:

```
 DATA QUALITY REPORT
============================================================
Missing Values:
  rating      766 (1.5%)
  category  1,108 (2.2%)

Duplicates: 234 rows (0.5%)

Outliers (rating):
  Below Q1-1.5*IQR: 0
  Above Q3+1.5*IQR: 127 
```

Interesting Finding:

```
 INSIGHT: Weekend Reviews Are More Positive

Reviews posted on weekends have an average rating of 4.2
compared to 3.8 for weekday reviews (p < 0.01).

Business Implication: Consider timing product launches
and review solicitation campaigns for weekends.
```

What Counts as "Interesting"?

- An unexpected distribution ("90% of reviews are 5-stars")
- A surprising correlation ("longer reviews tend to be more negative")
- A pattern ("complaints spike on Mondays")
- An anomaly ("one product has 10,000 reviews while avg is 50")
- A business insight ("premium products have fewer but longer reviews")

Common Mistakes (and How to Avoid Them)

Mistake	Symptom	Fix
Wrong file path	`FileNotFoundError`	Use `!ls` to check file location
Encoding issues	`UnicodeDecodeError`	Add `encoding='utf-8'` or `'latin-1'`
Memory overflow	Colab crashes	Load sample: `df.sample(10000)` or use `chunksize`

Mistake	Symptom	Fix
Treating strings as numbers	Math operations fail	Use `pd.to_numeric(col, errors='coerce')`
Ignoring data types	Wrong analysis results	Always run `df.dtypes` first
Duplicate column names	Confusing errors	Check with `df.columns.duplicated()`

If you see this error:

`ParserError: Error tokenizing data`

Try: `pd.read_csv(file, on_bad_lines='skip')` or check for inconsistent delimiters.

If HuggingFace is slow: - Use `streaming=True` for large datasets - Download once, then load from cache

---

Questions to Answer

- **Q1:** What is the shape of your dataset? What do the rows and columns represent?
- **Q2:** Which columns have missing values? What percentage?
- **Q3:** What data quality issues did you find? How would you address them?
- **Q4:** What interesting finding did you discover? Why does it matter for business?

Going Deeper (Optional Challenges)

Challenge A: Automated Data Profiling

Use the `pandas-profiling` (now `ydata-profiling`) library to generate a comprehensive HTML report of your dataset. Compare what it finds automatically vs. what you found manually.

Challenge B: Data Pipeline

Write a function that takes any CSV URL and returns a standardized quality report. Test it on 3 different datasets. Can you make it robust to different formats?

Challenge C: Outlier Investigation

Instead of just counting outliers, investigate them. Are they errors or genuine extreme values? Use visualization (box plots, scatter plots) to tell the story.

Quick Reference

```python
# Load data from various sources
import pandas as pd
```

```python
# From CSV
df = pd.read_csv("data.csv")

# From HuggingFace
from datasets import load_dataset
ds = load_dataset("imdb", split="train")
df = ds.to_pandas()

# From URL
df = pd.read_csv("https://example.com/data.csv")

# Basic exploration
df.shape              # (rows, columns)
df.head()             # First 5 rows
df.info()             # Column types and non-null counts
df.describe()         # Statistics for numeric columns

# Data quality checks
df.isnull().sum()                  # Missing values per column
df.duplicated().sum()              # Count duplicates
df[df.duplicated()]                # View duplicate rows

# Outlier detection (IQR method)
Q1 = df['col'].quantile(0.25)
Q3 = df['col'].quantile(0.75)
IQR = Q3 - Q1
outliers = df[(df['col'] < Q1-1.5*IQR) | (df['col'] > Q3+1.5*IQR)]

# Quick statistics
df['col'].value_counts()           # Frequency counts
df.groupby('category')['value'].mean()  # Group statistics
df.corr()                          # Correlation matrix
```

Useful Pandas Methods: | Method | Purpose | Example | |——–|———|———| | `.shape` | Dimensions | `(1000, 5)` | | `.dtypes` | Column types | `int64, object, float64` | | `.isnull().sum()` | Missing counts | `rating: 50` | | `.describe()` | Statistics | mean, std, min, max | | `.value_counts()` | Frequencies | `5-star: 400, 4-star: 300` | | `.nunique()` | Unique values | `categories: 12` |

Submission

Upload to Canvas: - Your completed `.ipynb` notebook with all cells executed

Richard Young, Ph.D.