10 Pandas Practice Use Cases

1. Sales Data Analysis Use Case

Operations:

- Count total sales per product category.
- Calculate the total revenue generated by each sales representative.
- Find the product with the highest sales.
- Group data by sales regions and calculate average sales.

2. Employee Data Analysis Use Case

Operations:

- Count the number of employees per department.
- Find the employee with the highest salary.
- Calculate average salary per department.
- Sort employees based on their performance score or salary.

3. Student Performance Analysis Use Case

Operations:

- Count the number of students per class/grade.
- Calculate the average marks per subject.
- Find the student with the highest overall score.
- Identify students who scored below the passing mark in any subject.

4. E-commerce Orders Analysis Use Case

Operations:

- Count total orders per customer.
- Find the customer who placed the highest number of orders.
- Calculate revenue generated per product.
- Identify the top 5 selling products.

5. Hospital Patient Records Analysis Use Case

Operations:

- Count the number of patients per department.
- Find the average hospital stay duration per department.
- Identify the patient with the longest hospital stay.
- Calculate the number of patients admitted per month.

6. Banking Transactions Analysis Use Case

Operations:

- Count the total number of transactions per account.
- Find the account with the highest transaction value.
- Calculate average transaction amount per customer.
- Group transactions by type (deposit, withdrawal) and calculate totals.

7. Retail Store Inventory Analysis Use Case

Operations:

- Count the number of products in each category.
- Find the product with the lowest stock quantity.
- Calculate the average stock value per category.
- Identify products that need restocking (below threshold).

8. Social Media Engagement Analysis Use Case

Operations:

- Count the number of posts per user.
- Find the post with the highest number of likes or comments.
- Calculate average likes per user.
- Identify the most active users based on posting frequency.

9. Airline Flight Data Analysis Use Case

Operations:

- Count the number of flights per airline.
- Calculate average flight delay per airline.
- Identify the route with the maximum number of flights.
- Find the longest and shortest flight duration.

10. Movie Ratings Analysis Use Case

Operations:

- Count the number of movies per genre.
- Calculate the average rating per movie.
- Find the movie with the highest rating.
- Identify top 5 directors based on average movie ratings.

Pandas Treasure Hunt: The Hidden Wealth of

Vijayanagara

Backdrop

It is the 16th century. The Vijayanagara Empire thrives, with bustling bazaars, grand gopurams, and far-reaching trade in spices, silk, and gems. Rumors whisper that King Krishnadevaraya hid a treasure of knowledge and wealth in secret chambers beneath Hampi. The treasure can only be revealed by solving 20 data scrolls preserved by scholars, merchants, and temple priests.

You are a young data explorer chosen to decode these scrolls using pandas, traveling across the empire, solving puzzles, and uncovering secrets. Each solved scroll brings you closer to the treasure.

Scene 1: The Royal Decree of Hampi

As you enter the palace, the royal librarian hands you a scroll listing noble families and their tribute. Some names are written in old Kannada script, and some tribute values seem oddly high. Task: Create a Series and DataFrame with family names and tribute amounts. Use the family names as the index. (Concept: Series/DataFrame creation + custom index)

Scene 2: The Hampi Bazaar

The largest market dazzles with diamonds, horses, spices, and silk. Merchants shout their prices, and you need to find only the stalls selling precious items. Task: Slice the DataFrame to show only the first 10 rows and last 5 rows. (Concept: Indexing & slicing)

Scene 3: The Palace Guard's Riddle

A guard stops you at the inner sanctum. "Who brings tribute from the House of Saluva?" he demands. Only by presenting the right record can you pass. Task: Use .loc to fetch rows where House == "Saluva". Handle cases where multiple entries exist. (Concept: Label-based indexing + handling duplicates)

Scene 4: Tungabhadra River Ferry

The ferryman demands to know which merchants are second, seventh, and last in the scroll of traders. Only then will he row you across the river. Task: Use .iloc to get the 2nd, 7th, and last row. Then reverse the order of the DataFrame. (Concept: Position-based indexing + slicing tricks)

Scene 5: Spice Ships at Goa Port

A Portuguese captain offers you a CSV file of spices exported from Vijayanagara: quantity, type, and price. Some quantities are decimals, but you need integers. Task: Import the CSV, display top 10 rows, and convert "Quantity" to integer type. (Concept: Import CSV + type conversion)

Scene 6: Temple Treasury Records

In Tirupati, temple priests reveal Excel files recording gold, land, and donations in multiple sheets. Each sheet tells only part of the story. Task: Load all sheets into separate DataFrames, then concatenate them into one DataFrame. Add a column indicating "Source" (Gold, Land, Donations). (Concept: Import Excel + concat + adding a column)

Scene 7: Royal Granaries of Andhra

Rice production scrolls show Area and Productivity, but some values seem exaggerated, hinting at errors. Task: Add a Total_Yield column = Area × Productivity. Identify outliers (> mean + 2*std) and filter them out. (Concept: Column operations + statistical filtering)

Scene 8: Horse Traders of Hampi

Arab horses, Persian horses, and fake horses crowd the marketplace. Only genuine Arab horses cost more than 2000 gold coins. Task: Filter rows where Price > 2000 & Origin != "Fake". Sort descending by price. (Concept: Filtering with conditions + sorting)

Scene 9: Monsoon Scrolls of Vijayanagara

Temple inscriptions record rainfall for irrigation tanks in different regions. The priests need the wettest regions for planning. Task: Group by Region and compute mean, max, and min rainfall. Return the top 3 wettest regions. (Concept: GroupBy + multiple aggregation)

Scene 10: Diamond Merchants of Golconda

Two ledgers record diamond trade separately. You must know total trade per region to unlock the merchant guild's secret. Task: Concatenate the two DataFrames vertically. Add a column Ledger_Source. (Concept: Concatenate + source tracking)

Scene 11: Scholar of Hampi

A scholar presents two scrolls: one with student names, another with marks. Some students are missing in one scroll. Task: Merge the DataFrames on Roll_Number. Compare inner, left, and right joins. (Concept: Merge types)

Scene 12: Caravanserai of Bijapur

Merchants' wool and cotton trade data share the same index, but some regions are missing records. Task: Join the DataFrames on index and fill missing values with 0. (Concept: Join + handling NaN)

Scene 13: Navaratri Festival Stalls

Festival stalls sell rice, sweets, silk, and horses. You are asked to analyze only sweets and silk. Task: Use .isin() to filter categories. Compute total revenue for these categories. (Concept: Filtering with isin + aggregation)

Scene 14: Census of Vijayanagara

Officials record populations of provinces. Some provinces are missing entries. Task: Find the top 5 populated provinces. Compute their percentage contribution to total population. (Concept: Sorting + percentage calculation)

Scene 15: Royal Chariot Schedules

Chariot races are organized at festivals. Each race has Departure and Arrival times. Task: Keep only Chariot Name, Departure, Arrival. Add a column Duration = Arrival − Departure in hours. (Concept: Column selection + datetime operations)

Scene 16: Court Poets of Hampi

Revenue from drama performances is recorded by year. Task: Group by Year, compute total revenue, and find the highest-grossing drama per year using groupby().apply(). (Concept: GroupBy + apply)

Scene 17: Royal Kitchen of Vijayanagara

Feast recipe data contains missing Spices. You must ensure no feast is incomplete. Task: Fill missing Spice with "Unknown". Forward-fill Quantity column. (Concept: fillna with method)

Scene 18: Temple Inscriptions

Donation inscriptions are sometimes repeated by mistake. Task: Drop duplicates, reset index, and rename columns appropriately. (Concept: drop_duplicates + reset_index + rename)

Scene 19: Assembly of Data Scrolls

After collecting 19 scrolls, you must merge them into one master dataset. Task: Merge/concatenate all scrolls. Drop duplicates. Summarize Treasure_Value by Region. (Concept: Large-scale merging + cleanup + final aggregation)

Scene 20: Hidden Chamber of Hampi

Finally, at the Virupaksha Temple, you decode the last riddle to reveal the treasure. Task: From the final dataset, compute:

Region with maximum treasure

Average treasure per category

Treasure Key = sum of top 3 treasures

Printing the treasure key unlocks the hidden wealth of Vijayanagara!

Learning Outcomes

By the end, students will:

Progress from basic pandas → advanced problem solving

Handle real-world operations: merging multiple sheets, missing data, custom filtering, datetime, aggregations, and apply functions

Experience an immersive historical narrative, making pandas problem-solving an adventure.