# **Documentation for Data Engineering Project**

# **Overview**

This Python script is designed to process a dataset (2024\_State\_Employee\_Pay.csv) containing information about state employees, such as their position titles, agencies, and year-to-date (YTD) gross pay. The script performs data cleaning and analysis to generate insights, including the average gross pay grouped by position title and agency. The results are saved in separate CSV files for further analysis.

## **2. Features**

### **Key Operations Performed:**

1. **Load Data**: Reads the input dataset into memory using Pandas.
2. **Data Cleaning**:
   * Removes duplicate entries.
   * Ensures the YTD gross pay column contains numeric values.
   * Removes rows with invalid or missing YTD gross pay.
   * Drops unnecessary columns, such as employee names.
3. **Data Grouping**:
   * Calculates the average YTD gross pay grouped by position title and agency.
4. **Export Results**:
   * Saves the grouped data into separate CSV files.

## **3. File Inputs and Outputs**

### **Input**

* **File Name**: 2024\_State\_Employee\_Pay.csv
* **Content**:
  + Columns:
    - Position Title: The job title of the employee.
    - Agency Name: The name of the agency employing the individual.
    - YTD Gross Pay: The total gross pay of the employee year-to-date.
    - Employee Name (optional): The name of the employee (if available).

### **Output**

* **Files**:
  1. position\_title\_grouped.csv:
     + Contains average YTD gross pay grouped by Position Title.
  2. agency\_name\_grouped.csv:
     + Contains average YTD gross pay grouped by Agency Name.
* **Location**: By default, the output is saved in the directory specified in the script (c:/Users/tareq/venv/DE Proj/).

## **4. Script Workflow**

### **Step 1: Load the Dataset**

The script begins by reading the input CSV file into a Pandas DataFrame:

python

Copy code

import pandas as pd

data = pd.read\_csv('2024\_State\_Employee\_Pay.csv')

print(f"Data loaded with {data.shape[0]} rows and {data.shape[1]} columns.")

* **Functionality**:
  + Ensures the file is loaded correctly.
  + Prints the dimensions of the dataset for verification.

### **Step 2: Data Cleaning**

#### **2.1 Remove Duplicates**

Duplicate rows are removed to ensure the data is unique:

python

Copy code

data\_cleaned = data.drop\_duplicates()

print(f"Removed duplicates. Rows remaining: {data\_cleaned.shape[0]}")

#### **2.2 Ensure Numeric Values in 'YTD Gross Pay'**

The script converts the YTD Gross Pay column to numeric format:

python

Copy code

if 'YTD Gross Pay' in data\_cleaned.columns:

data\_cleaned['YTD Gross Pay'] = pd.to\_numeric(data\_cleaned['YTD Gross Pay'], errors='coerce')

else:

raise ValueError("'YTD Gross Pay' column is missing!")

* **Handling**:
  + Non-numeric values are coerced to NaN.
  + If the column is missing, an error is raised.

#### **2.3 Remove Invalid Rows**

Rows with NaN values in the YTD Gross Pay column are dropped:

python

Copy code

data\_cleaned = data\_cleaned.dropna(subset=['YTD Gross Pay'])

print(f"Invalid 'YTD Gross Pay' rows removed. Rows remaining: {data\_cleaned.shape[0]}")

#### **2.4 Drop Unnecessary Columns**

If the Employee Name column exists, it is removed:

python

Copy code

if 'Employee Name' in data\_cleaned.columns:

data\_cleaned = data\_cleaned.drop(columns=['Employee Name'])

print("'Employee Name' column dropped.")

### **Step 3: Data Grouping and Aggregation**

The cleaned data is grouped by Position Title and Agency Name to calculate the average YTD gross pay:

python

Copy code

for group\_by in ['Position Title', 'Agency Name']:

group = data\_cleaned.groupby(group\_by)['YTD Gross Pay'].mean().reset\_index()

group = group.rename(columns={'YTD Gross Pay': 'Average Gross Pay'})

group.to\_csv(f'{output\_dir}{group\_by.lower().replace(" ", "\_")}\_grouped.csv', index=False)

print(f"Grouped data by {group\_by} saved.")

* **Output**:
  + Two CSV files are created: one for position titles and another for agencies.

## **5. Notes**

* **Error Handling**:
  + The script raises an error if the YTD Gross Pay column is missing.
  + Non-numeric values in YTD Gross Pay are handled by converting them to NaN.
* **Requirements**:
  + Ensure the input file is in the correct format with the expected column names.

Install Pandas using:  
bash  
Copy code  
pip install pandas

* **Performance**:
  + The script handles large datasets efficiently, but ensure your system has sufficient memory.

## **6. Example Usage**

### **Command to Run the Script:**

1. Place 2024\_State\_Employee\_Pay.csv in the working directory.

Run the script:  
bash  
Copy code  
python Data\ Engineering.py

1. Check the output directory for the generated CSV files.

## **7. Improvements and Extensions**

* **Potential Enhancements**:
  + Add support for command-line arguments to specify input/output paths.
  + Include additional analyses (e.g., median gross pay).
  + Implement logging for better traceability.