# Project Instructions: Exploratory Data Analysis (EDA) in Python

## Objective

The objective of this project is to select a dataset, perform data cleaning and pre-processing, conduct exploratory data analysis (EDA), and present your findings. This project will help you understand the dataset, uncover underlying patterns, and generate insights that could guide further analysis or decision-making.

## Steps and Guidelines

1. **Select a Dataset**
    - Choose a dataset that interests you. The dataset can be from a public source such as Kaggle, UCI Machine Learning Repository, or any other reliable source.
    - Ensure the dataset is sufficiently large and has a variety of features (columns) to analyze. Aim for at least 500 rows and 5 columns.
2. **Project Setup**
    - Create a new directory for your project.
    - Use a Jupyter Notebook for your analysis. Name your notebook `EDA_Project_yourname.ipynb`.
    - Create a README file that briefly describes the dataset and the steps you plan to take in your analysis.
3. **Data Import and Cleaning**
    - Import the necessary libraries: `pandas`, `numpy`, `matplotlib`, `seaborn`, etc.
    - Load the dataset into a pandas DataFrame.
    - Perform initial data inspection: check the shape of the data, data types, and summary statistics.
    - Identify and handle missing values. Decide whether to drop, fill, or interpolate missing data based on the context.
    - Detect and remove duplicate rows if any.
    - Convert data types if necessary (e.g., dates should be in datetime format).
4. **Exploratory Data Analysis (EDA)**
    - **Descriptive Statistics**:
        - Provide summary statistics for numerical columns (mean, median, standard deviation, etc.).
        - Provide summary statistics for categorical columns (frequency counts, unique values, etc.).
    - **Data Visualization**:
        - Create histograms or density plots for numerical features to understand their distributions.
        - Create bar plots for categorical features to visualize the frequency of categories.
        - Use box plots to identify outliers and understand the spread of the data.
        - Create scatter plots to explore relationships between numerical features.
        - Use heatmaps to visualize correlations between numerical features.

- **Group Analysis**:
    - Perform group-by operations to aggregate data based on categorical features.
- **Feature Analysis**:
    - Identify and analyze key features that might be important for understanding the dataset.
    - Explore relationships between features using pair plots, correlation matrices, and pivot tables.
    - Perform any additional analyses that might be relevant to your dataset (e.g., time series analysis for time-related data).

**5.Advanced Python Techniques**

- **Lambda Functions**:
    - Use lambda functions for simple data transformations.
    - Example: Apply a lambda function to create a new column that categorizes numerical data into bins.
- **User-Defined Functions**:
    - Write custom functions to perform repetitive tasks or complex calculations.
    - Example: Create a function to calculate the range of salary as low, medium, high.
- **List Comprehensions**:
    - Use list comprehensions for efficient data processing and transformation.
    - Example: Generate a list of column names that have missing values.

**6.Insights and Conclusions**

- Summarize your key findings from the EDA.
- Discuss any patterns, anomalies, or interesting relationships you discovered.
- Highlight any potential areas for further analysis or questions that emerged from your EDA.

**7.Documentation and Presentation**

- Ensure your Jupyter Notebook is well-documented. Include markdown cells to explain each step, the rationale behind your choices, and your findings.
- Visualizations should have clear titles, axis labels, and legends where necessary.
- Prepare a brief presentation (5-10 slides) summarizing your project. Include key findings, interesting visualizations, and potential next steps.
- Submit your Jupyter Notebook, the dataset, the README file, and the presentation slides.

**Submission Deadline**

- Please submit your project by [Insert Deadline Here].

If you have any questions or need further assistance, feel free to reach out during office hours or via email.

---

## Evaluation Rubric for Python EDA Project

**Total Marks: 20**

| Criteria | Description | Marks | Scoring Details |
|---|---|---|---|
| **1. Dataset Selection (2 Marks)** | | | |
| Relevance | The dataset should be relevant and appropriate for the analysis. | 1 | 1: Highly relevant, 0.5: Somewhat relevant, 0: Not relevant |
| Complexity and Variety | The dataset should have sufficient complexity and variety (e.g., 500 rows, 5 columns). | 1 | 1: Meets requirements, 0.5: Partially meets, 0: Does not meet |
| **2. Data Cleaning (2 Marks)** | | | |
| Missing Values Handling | Proper identification and handling of missing values. | 1 | 1: Effectively handled, 0.5: Partially handled, 0: Not handled |
| Duplicate and Inconsistent Data | Detection and resolution of duplicate and inconsistent data. | 1 | 1: Effectively handled, 0.5: Partially handled, 0: Not handled |
| **3. Exploratory Data Analysis (6 Marks)** | | | |
| Descriptive Statistics | Calculation of basic statistics for numerical and categorical columns. | 2 | 2: Comprehensive, 1: Partial, 0: Missing or incorrect |
| Data Visualization | Use of relevant and clear visualizations. | 2 | 2: Relevant and clear, 1: Partial clarity, 0: Missing or unclear |
| Feature Analysis | Thorough analysis of key features and their relationships. | 2 | 2: Thorough, 1: Partial, 0: Missing or not insightful |
| **4. Group Analysis (2 Marks)** | | | |
| Group-by Operations | Perform group-by operations to aggregate data based on categorical features. | 2 | 2: Effectively performed, 1: Partially performed, 0: Not performed |
| **5. Advanced Python Techniques (4 Marks)** | | | |
| Lambda Functions | Use lambda functions for simple data transformations. | 1 | 1: Effectively used, 0.5: Partially used, 0: Not used |

| Criteria | Description | Marks | Scoring Details |
|---|---|---|---|
| User-Defined Functions | Write custom functions for repetitive tasks or complex calculations. | 1 | 1: Effectively written, 0.5: Partially written, 0: Not written |
| List Comprehensions | Use list comprehensions for efficient data processing and transformation. | 2 | 2: Effectively used, 1: Partially used, 0: Not used |
| **6. Insights and Conclusions (2 Marks)** | | | |
| Significance of Insights | Insights should be meaningful and relevant. | 1 | 1: Highly significant, 0.5: Somewhat significant, 0: Not significant |
| Clarity of Conclusions | Conclusions should be clearly stated and supported by the analysis. | 1 | 1: Clear and well-supported, 0.5: Somewhat clear, 0: Unclear or unsupported |
| **7. Documentation and Presentation (2 Marks)** | | | |
| Quality of Documentation & | The Jupyter Notebook should be well-documented with comments. | 1 | 1: Clear and thorough, 0.5: Partially clear, 0: Unclear or missing |
| Quality of Presentation | The presentation should summarize key findings effectively. | 1 | 1: Clear and well-organized, 0.5: Partially clear, 0: Unclear or disorganized |

## Summary of Marks:

- **Dataset Selection**: 2 marks
- **Data Cleaning**: 2 marks
- **Exploratory Data Analysis**: 6 marks
- **Group Analysis**: 2 marks
- **Advanced Python Techniques**: 4 marks
- **Insights and Conclusions**: 2 marks
- **Documentation and Presentation**: 2 marks