Data exploration and visualization script using the Pandas, Matplotlib, and Seaborn libraries. It is designed to analyze a dataset named 'ACME-HappinessSurvey2020.csv'. Below is a technical description of the code:

**1. Import Libraries:** The script begins by importing the necessary libraries, including Pandas for data manipulation, Matplotlib for creating plots, and Seaborn for enhancing the visualizations.

**2. Load the Dataset:** The dataset is loaded from a CSV file named 'ACME-HappinessSurvey2020.csv' into a Pandas DataFrame named 'df'.

**3. Data Overview**: The script prints a data overview by displaying the first few rows of the DataFrame using `df.head()`. This provides a quick glimpse of the dataset's structure and content.

**4. Data Summary:** The script then provides a summary of the dataset using `df.info()` and `df.describe()`.

- `df.info()` displays information about the DataFrame, including the number of non-null entries and data types of each column.

- `df.describe()` provides basic statistics (count, mean, std, min, max, etc.) for numerical features in the dataset.

**5. Unique Values in Categorical Features:** The code iterates through each column in the dataset and prints the unique values and their counts for categorical features. This is useful for understanding the diversity and distribution of categorical data.

**6. Data Visualization:** The script proceeds with data visualization using Matplotlib and Seaborn.

Histograms for Numerical Features\*\*: It creates histograms for numerical features using `sns.histplot`. Each numerical feature is plotted with 20 bins and a kernel density estimate (KDE) for visualizing the distribution.

Bar Plots for Categorical Features\*\*: For each categorical feature, the script creates bar plots using `sns.countplot`. The bars represent the counts of each category, and the x-axis labels are rotated for better readability.

Correlation Matrix for Numerical Features\*\*: It generates a correlation matrix using `df.corr()` to explore the relationships between numerical features. The matrix is visualized as a heatmap with annotated correlation coefficients.

Target Variable Analysis\*\*: The code analyzes the distribution of the target variable 'Y' using `sns.countplot`. This is useful for understanding the class distribution in a classification problem.

**7. Plot Display:** After creating all the visualizations, the code calls `plt.tight\_layout()` to ensure proper spacing between subplots and then displays the plots using `plt.show()`.

Overall, this code provides a comprehensive exploration and visualization of the dataset, helping users gain insights into its structure and relationships between variables. It's a valuable starting point for any data analysis project, especially when dealing with both numerical and categorical data.