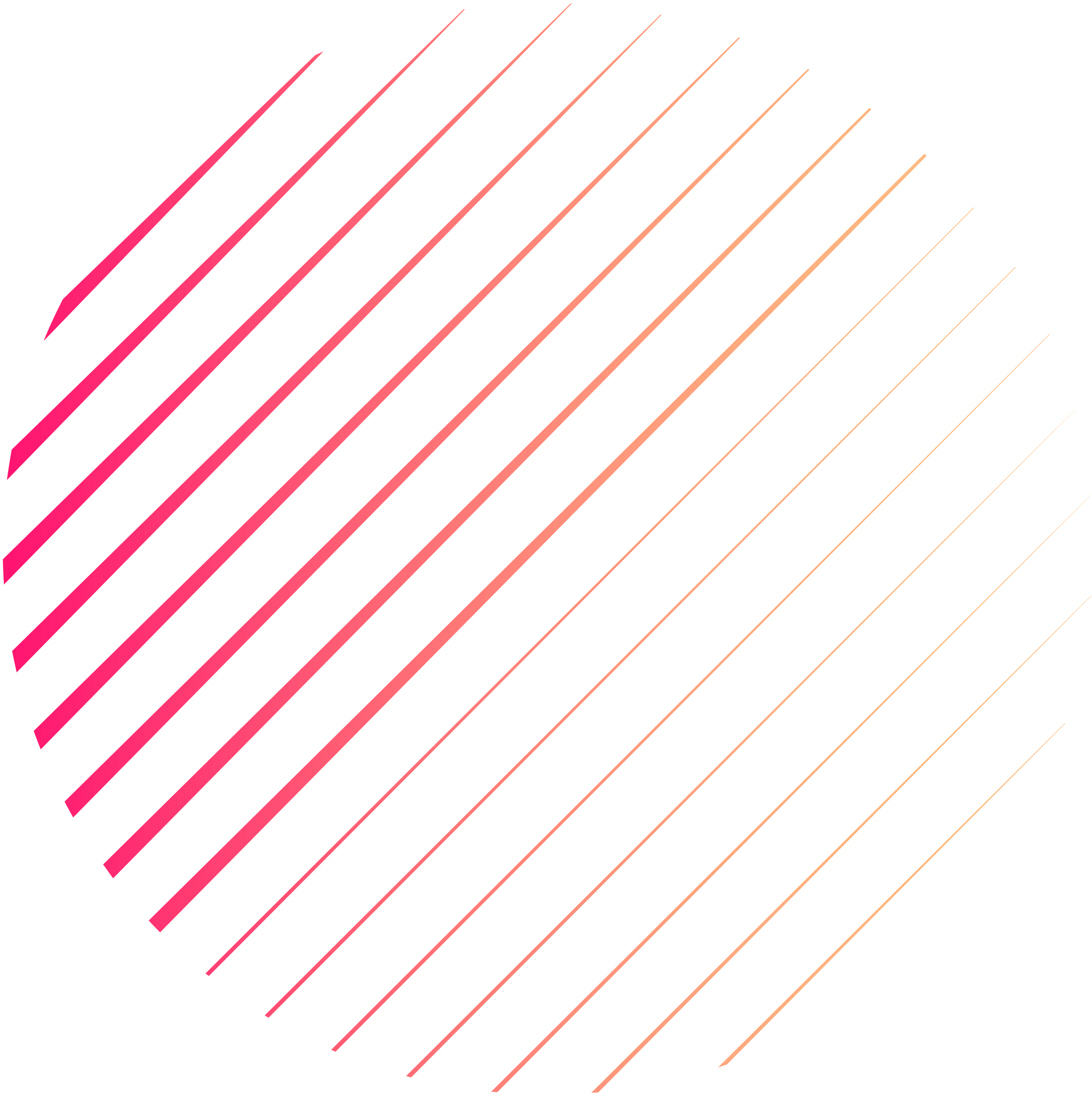
# Visualizing Compensation & Talent: A Data-Driven Look at Employee Salaries, Gender.



Data Visualization using AI tools

Y. Adithya Vardhan Reddy

# Objectives:

1. **Uncover the Salary Landscape:** We'll create a captivating histogram to visualize the distribution of salaries, revealing its peaks, valleys, and overall shape. Through this visual journey, we'll uncover hidden patterns and potential outliers that might otherwise remain unseen.

2. **Decipher Gender Pay Dynamics:** Using a persuasive bar plot, we'll engage in a compelling conversation about potential salary differences between male and female employees. Our goal is to illuminate any existing gaps and spark meaningful dialogue about workplace equity.

3. **Explore the Experience-Salary Nexus:** We'll construct an insightful scatter plot to visualize the delicate dance between previous work experience and current salary, revealing whether experience truly translates into financial rewards within our organization.

4. **Slice the Education Pie:** We'll bake a delectable pie chart to showcase the diverse educational backgrounds of our employees, highlighting the unique flavors each slice brings to our workforce.

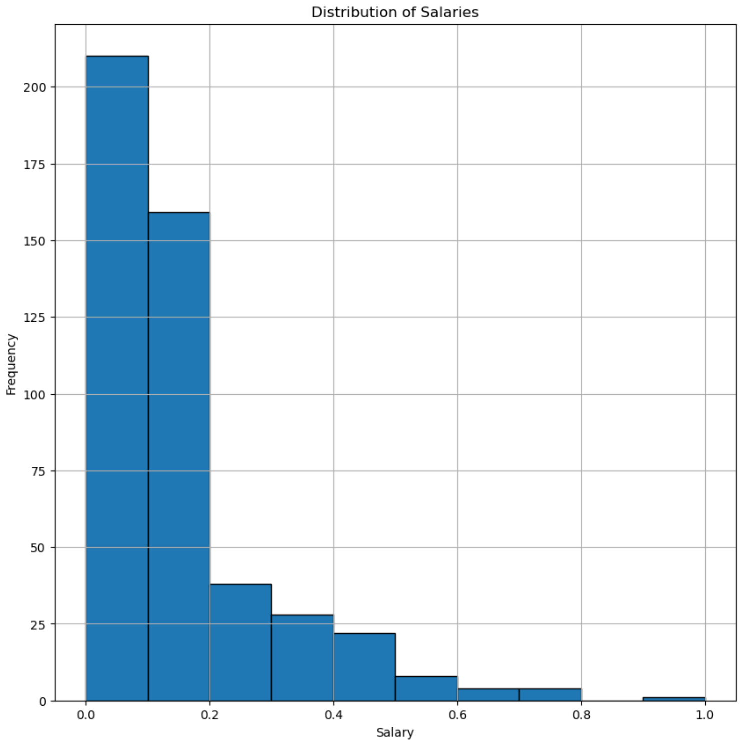
5. **Unleash the AI Storyteller:** With the help of AI-powered libraries like Sweetviz or Dtale, we'll weave a captivating narrative about our data's overarching themes and key insights. These intelligent tools will empower us to uncover hidden stories that might otherwise remain untold.

# Data visualization (charts/graphs):

## 1. Salary Distribution Histogram:

**Objective:** Understand the spread and central tendency of employee salaries.

**Visualization:** Histogram with labeled axes (salary range and frequency), title, and optional annotations for key statistics like mean, median, and percentiles.



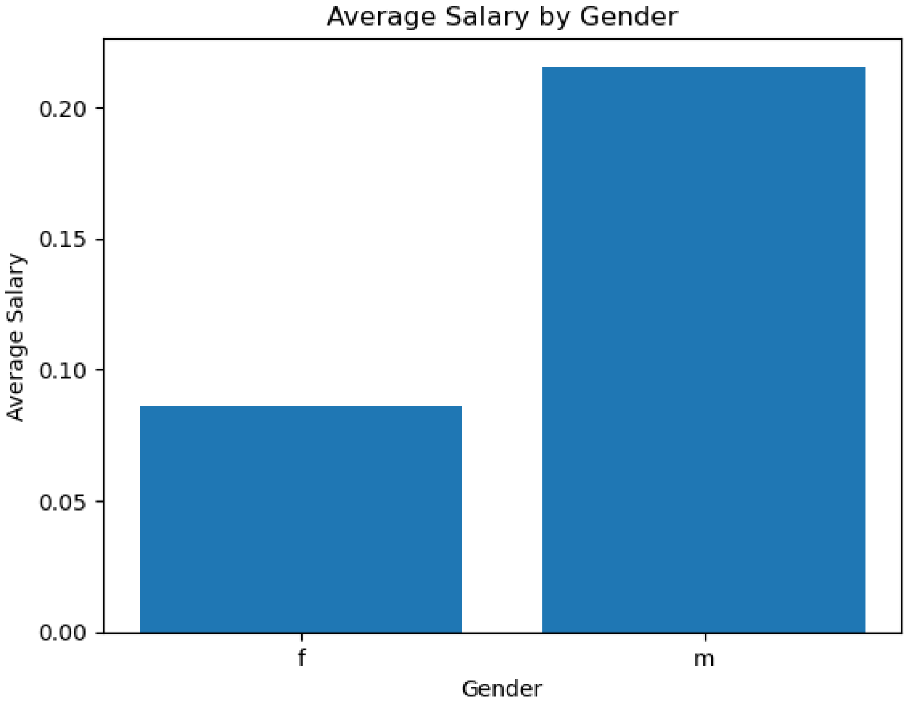
**Python Scripting:**

```
#Salary Distribution Histogram
plt.figure(figsize=(10, 10))

plt.hist(df["salary"], bins=10, edgecolor="black")

plt.xlabel("Salary")

plt.ylabel("Frequency")

plt.title("Distribution of Salaries")

plt.grid(True)

plt.show()
```

## 2. Average Salary by Gender Bar Plot:

**Objective:** Compare average salaries between male and female employees.

**Visualization:** Bar plot with one bar for each gender, labeled with average salaries. Consider adding error bars to display standard deviation or confidence intervals for each group.



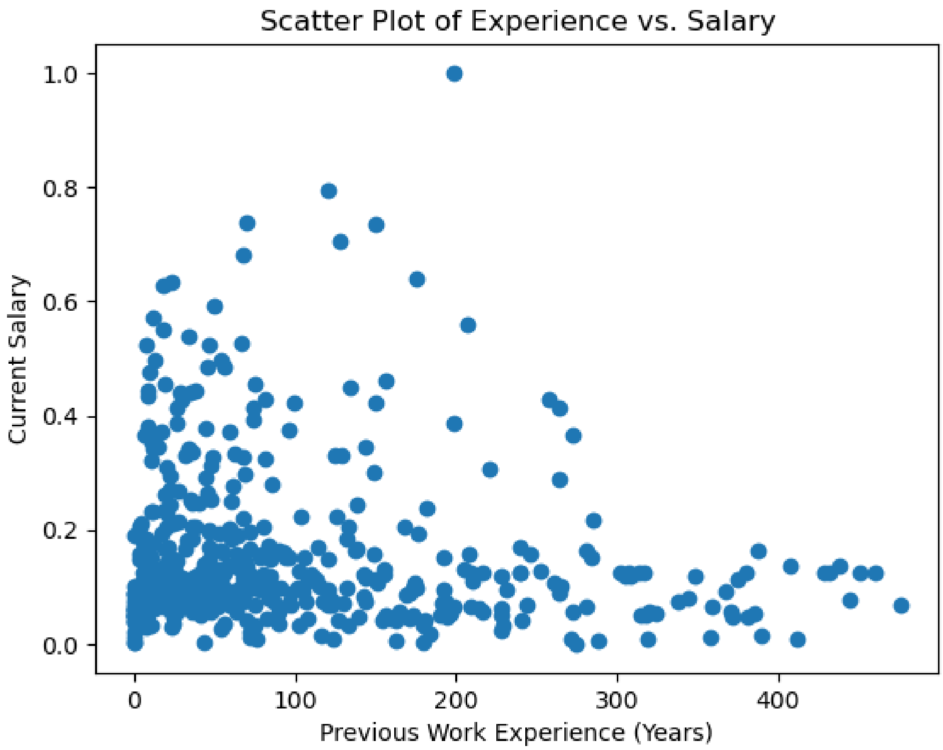
**Python Scripting:**

```
avg_salaries = df.groupby("gender")["salary"].mean()


features = ["prevexp", "salary"]  # Select relevant features
print(df[features])
df[features].describe()
df[features].hist()
plt.scatter(df["prevexp"], df["salary"])
plt.xlabel("Previous Work Experience (Years)")
plt.ylabel("Current Salary")
plt.title("Scatter Plot of Experience vs. Salary")
plt.show()
```

### 3. Prevexp vs. Salary Scatter Plot:

**Objective:** Explore the relationship between previous work experience and current salary.

**Visualization:** Scatter plot with dots representing each employee, with prevexp on the x-axis and salary on the y-axis. Add a trendline (e.g., linear regression) if appropriate to highlight the general correlation.



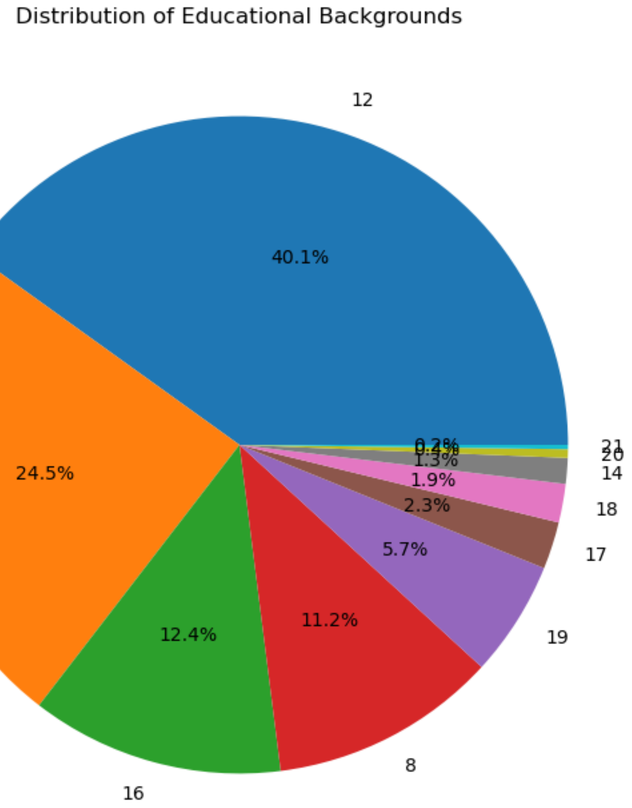
**Python Scripting:**

```
# Visualization
plt.hist(df["salary"])  # Example: create a histogram of salaries
plt.show()
```

## 4. Educational Background Pie Chart:

**Objective:** Visualize the proportion of employees with different educational backgrounds.

**Visualization:** Pie chart with each slice representing a different educational level (e.g., high school, bachelor's, master's, etc.) labeled with percentages. Use distinct and clear colors for each segment.

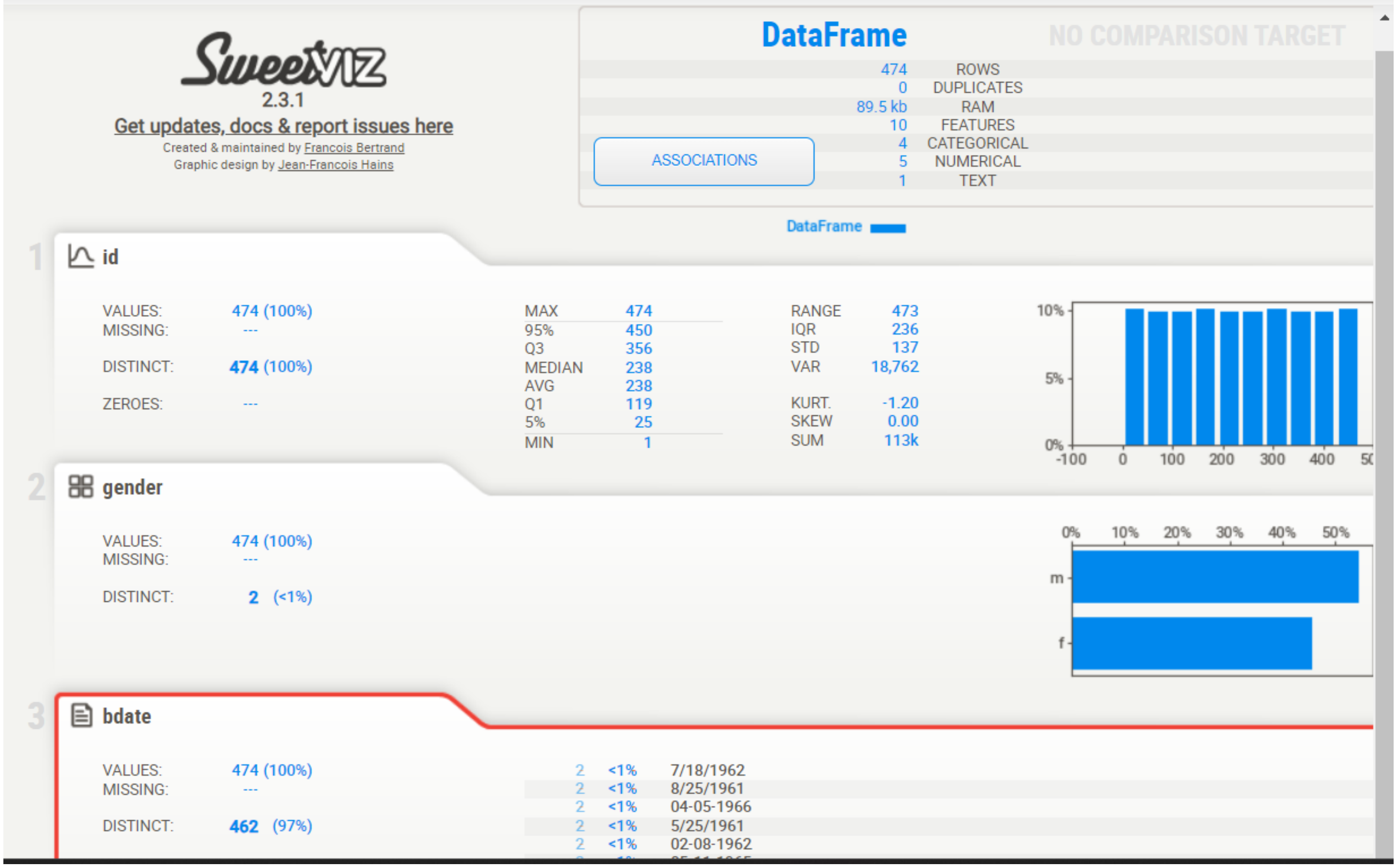


**Python Scripting:**

```
educ_counts = df["educ"].value_counts()

#Educational Background Create a pie chart

plt.figure(figsize=(8, 8))

plt.pie(educ_counts.values, labels=educ_counts.index, autopct="%1.1f%%")

plt.title("Distribution of Educational Backgrounds")

plt.show()
```

## 5. AI Libraries for Data Summary:

**Objective:** Generate a comprehensive overview of the data using machine learning-powered tools.



SweetVIZ
2.3.1
Get updates, docs & report issues here
Created & maintained by Francois Bertrand
Graphic design by Jean-Francois Hains

**DataFrame** NO COMPARISON TARGET

| | |
|---|---|
| 474 | ROWS |
| 0 | DUPLICATES |
| 89.5 kb | RAM |
| 10 | FEATURES |
| 4 | CATEGORICAL |
| 5 | NUMERICAL |
| 1 | TEXT |

ASSOCIATIONS

DataFrame ▬

1 id

| | | | | | |
|---|---|---|---|---|---|
| VALUES: | 474 (100%) | MAX | 474 | RANGE | 473 |
| MISSING: | --- | 95% | 450 | IQR | 236 |
| | | Q3 | 356 | STD | 137 |
| DISTINCT: | 474 (100%) | MEDIAN | 238 | VAR | 18,762 |
| | | AVG | 238 | | |
| ZEROES: | --- | Q1 | 119 | KURT. | -1.20 |
| | | 5% | 25 | SKEW | 0.00 |
| | | MIN | 1 | SUM | 113k |

2 gender

VALUES: 474 (100%)
MISSING: ---
DISTINCT: 2 (<1%)

3 bdate

VALUES: 474 (100%)
MISSING: ---
DISTINCT: 462 (97%)

| | | |
|---|---|---|
| 2 | <1% | 7/18/1962 |
| 2 | <1% | 8/25/1961 |
| 2 | <1% | 04-05-1966 |
| 2 | <1% | 5/25/1961 |
| 2 | <1% | 02-08-1962 |

### Python Scripting:

#AI libraries for Data Summary

report = sv.analyze(df)

report.show_notebook( w=None, h=None, scale=None, layout='widescreen', filepath=None, file_layout=None, file_scale=None)

## Python Scripting:

```
import pandas as pd  #For data manipulation and analysis
import numpy as np  #For numerical operations
import matplotlib.pyplot as plt  # For data visualization
import sweetviz as sv # AI library for text summarization
from sklearn.preprocessing import MinMaxScaler

df = pd.read_csv("ED.csv")  # Replace with your file path
df.head()  # View first few rows
df.describe()  # Summary statistics


# Apply transformations if needed (e.g., scaling, normalization)
# Example: Normalize salary
scaler = MinMaxScaler()
df["salary"] = scaler.fit_transform(df[["salary"]])
df["salbegin"].describe()  # Summary statistics

#Salary Distribution Histogram
plt.figure(figsize=(10, 10))
plt.hist(df["salary"], bins=10, edgecolor="black")
plt.xlabel("Salary")
plt.ylabel("Frequency")
plt.title("Distribution of Salaries")
plt.grid(True)
plt.show()


# Select relevant features for analysis
features = ["gender", "salary"]
print(df[features])


# Calculate average salaries by gender
```

```
avg_salaries = df.groupby("gender")["salary"].mean()


features = ["prevexp", "salary"]  # Select relevant features
print(df[features])
df[features].describe()
df[features].hist()
plt.scatter(df["prevexp"], df["salary"])
plt.xlabel("Previous Work Experience (Years)")
plt.ylabel("Current Salary")
plt.title("Scatter Plot of Experience vs. Salary")
plt.show()

# Visualization
plt.hist(df["salary"])  # Example: create a histogram of salaries
plt.show()


# Count the frequency of each educational background
educ_counts = df["educ"].value_counts()



#Educational Background Create a pie chart
plt.figure(figsize=(8, 8))
plt.pie(educ_counts.values, labels=educ_counts.index, autopct="%1.1f%%")
plt.title("Distribution of Educational Backgrounds")
plt.show()


#AI libraries for Data Summary
report = sv.analyze(df)
report.show_notebook(  w=None,  h=None, scale=None,  layout='widescreen', filepath=None,
file_layout=None,              file_scale=None)
```

# Conclusion:

- **Salary Distribution:** The salary distribution is visualized as a histogram, revealing its overall shape and central tendency.

- **Gender and Salary:** Average salaries are calculated for each gender, with a bar plot illustrating potential differences.

- **Experience and Salary:** The relationship between previous work experience and current salary is explored through a scatter plot to identify any patterns or correlations.

- **Educational Background:** The frequency of different educational backgrounds is counted and presented as a pie chart to visualize their distribution.

- **Interactive Report:** Sweetviz generates an interactive HTML report, providing a comprehensive overview of key insights and visualizations.