<Project assignment 1>

Dataset Report: Data Science Salary Analysis

1. Why Choose This Dataset?

This dataset provides global salary information for the data science field, including job titles, experience levels, employment types, salaries (in different currencies), and company locations. Since our team members aspire to become data scientists in the future, analyzing this dataset allows us to understand salary trends, the impact of remote work, and salary differences across countries.

2. Key Data Columns and Their Characteristics

- Number of Rows: 3,773
- Description of Each Column:
  I. experience_level (Experience Level): Represents job experience levels such as EN (Entry), MI (Mid-level), SE (Senior), EX (Executive).
  II. job_title (Job Title): Represents various data science-related positions such as Data Scientist, Machine Learning Engineer, etc.
  III. salary_in_usd (Salary in USD): Annual salary converted to USD.
  IV. remote_ratio (Remote Work Ratio):
  0 (Fully On-Site Work) / 50 (Hybrid Work) / 100 (Fully Remote Work)
  V. company_location: Country code representing the company's location (e.g., US for the United States, GB for the United Kingdom).
  VI. company_size (Company Size):
  S (Small company) / M (Medium-sized company) / L (Large company)

3. Statistical Analysis of Key Numerical Variables

```
Statistics for salary_in_usd:
Mean : 137509.51126424596
Median : 135000.0
Mode : 100000
Standard Deviation : 63042.2252169619
Range : 5132 - 450000


Statistics for remote_ratio:
Mean : 46.30267691492181
Median : 0.0
Mode : 0
Standard Deviation : 48.57710890737816
Range : 0 - 100
```

4. Primary Data Types

- experience_level: Categorical (Nominal)
- job_title: Categorical (Nominal)
- salary_in_usd: Numerical (Integer)
- remote_ratio: Numerical (Integer)

- company_location: Categorical (Nominal)
- company_size: Categorical (Ordinal)

5. Missing Data Analysis for Key Columns

The 'Job Title' field contains 10 missing values, while 'Company Size' has 6 missing values. Other fields have no missing data.

experience_level	0
job_title	10
salary_in_usd	0
remote_ratio	0
company_location	0
company_size	6

6. Data Quality Analysis

Using info() and nunique() to analyze data:

(1) Unique Value Count:

- experience_level: 4 different experience levels
- job_title: 93 unique job titles
- company_location: 72 different countries
- company_size: 3 categories (S, M, L)

7.Top 5 Most Common Values

(1) Most Common Job Titles (job_title)

- Data Engineer: 1,041 occurrences
- Data Scientist: 841 occurrences
- Data Analyst: 614 occurrences
- Machine Learning Engineer: 289 occurrences
- Analytics Engineer : 103 occurrences

(2) Top 5 Company Locations (company_location)

- United States (US): 3,051 companies
- United Kingdom (GB): 173 companies
- Canada (CA): 88 companies
- Spain (ES): 77 companies
- India (IN): 60 companies

8. Research Questions

1. Does the remote work ratio affect salary levels?
2. What are the salary differences among experience levels?
3. Which country offers the highest salaries for data scientists?
4. How do small, medium, and large companies compare in terms of salary differences?

<Project assignment 2 - Data Cleaning & Quality Analysis>

1. Data Quality Analysis (Before Cleaning)

```
# Quantify Missing Values
missing_values = df_selected.isnull().sum()
missing_values
```

```
experience_level     0
job_title           10
salary_in_usd        0
remote_ratio         0
company_location     0
company_size         6
```

- Missing Values:
  - job_title: 10 missing values
  - company_size: 6 missing values
  - Other columns have no missing values.
- Unique Value Counts : Unique Value Counts are shown below. All fields have business value and cannot be deleted even if some fields contain few unique values.

```
# Unique Value Counts (to identify low-information columns)
unique_counts = df_selected.nunique()
unique_counts
# All fields have business value and cannot be deleted even if some fields contains few unique values.
```

```
experience_level       4
job_title             93
salary_in_usd       1035
remote_ratio           3
company_location      72
company_size           3
```

- Check for Duplicate Rows : There are 1406 duplicates in total.

```
duplicate_rows = df_selected[df_selected.duplicated(keep='first')]
print(f"Total duplicated rows: {duplicate_rows.shape[0]}")
duplicate_rows
```

```
Total duplicated rows: 1406
```

2. Data Cleaning Approach

- Drop duplicate rows : Deleting duplicate data is mainly to ensure data accuracy, improve computing efficiency, and reduce data redundancy. There were 3773 records originally. After removing duplicate records (1406 records), there were only 2367 records left.

```
df_cleaned = df_selected.drop_duplicates()
print(f"Total rows BEFORE dropping duplicate rows: {df_selected.shape[0]}")
print(f"Total duplicate rows: {duplicate_rows.shape[0]}")
print(f"Total rows AFTER dropping duplicate rows: {df_cleaned.shape[0]}")
df_cleaned  # DataFrame without duplicate records
```

```
Total rows BEFORE dropping duplicate rows: 3773
Total duplicate rows: 1406
Total rows AFTER dropping duplicate rows: 2367
```

- Handle missing values:

  - Drop rows where “job_title” is missing : 2357 records left

```
# - Drop rows where job_title is missing (essential info)
df_cleaned = df_cleaned.dropna(subset=['job_title'])
print(f"Total rows AFTER dropping row of missing job_title: {df_cleaned.shape[0]}")
df_cleaned  # DataFrame without missing job_title
```

```
Total rows AFTER dropping row of missing job_title: 2357
```

  - Fill missing values in “company_size” with the mode:

company_size is categorical data (such as "S", "M", "L"). Filling with mode() can preserve the consistency of data distribution and avoid human bias. It is more accurate than random filling and does not affect statistical analysis. This

maximizes data utilization without compromising the representativeness of company_size.

```
# - Fill missing values in company_size with the mode
df_cleaned.loc[:, 'company_size'] = df_cleaned['company_size'].fillna(df_cleaned['company_size'].mode()[0])
df_cleaned['company_size'] # DataFrame without missing company_size
```

- Fix outlier:

  - Check Outlier Values of “salary_in_usd”: upper boundary is 315350

```
# Check Outlier Values of "salary_in_usd"
numerical_columns = ["salary_in_usd"]
Q1 = df_cleaned[numerical_columns].quantile(0.25)
Q3 = df_cleaned[numerical_columns].quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
print("lower_bound = ", lower_bound.values[0])
print("upper_bound = ", upper_bound.values[0])
```

```
lower_bound =  -58750.0
upper_bound =  315250.0
```

  - Replace outliers (salary_in_usd > 315350) with the mean:

```
# Calculate the average salary_in_usd
mean_value = int(df_cleaned["salary_in_usd"].mean())

# Replace outliers (those greater than 315250) with the mean value
df_cleaned.loc[df_cleaned["salary_in_usd"] > 315250, "salary_in_usd"] = mean_value
df_cleaned
```

- Convert Data Types for Consistency:

  - The data type of “salary_in_usd” is changed to “integer”

  - “remote_ratio” is displayed as a “percentage”, but the value is still integer, so it can be directly calculated in the future.

```
# The salary data type is changed to "integer"
df_cleaned.loc[:, "salary_in_usd"] = df_cleaned["salary_in_usd"].astype(int)

# remote_ratio is displayed as a percentage, but the value is still int, so it can be directly calculated in the future
df_cleaned.head(10).style.format({"remote_ratio": "{}%"})
```

	experience_level	job_title	salary_in_usd	remote_ratio	company_location	company_size
0	SE	Principal Data Scientist	85847	100%	ES	L
1	MI	ML Engineer	30000	100%	US	S

3. Summary of Cleaned Data:

Summary of Cleaned Data:

- Data Shape : (row, columns) = (2357,6)

```
# Data Shape
final_shape = df_cleaned.shape
final_shape    #(rows,columns)
```

```
(2357, 6)
```

- Data Type

```
final_dtypes = df_cleaned.dtypes
final_dtypes
```

```
experience_level    object
job_title           object
salary_in_usd        int64
remote_ratio         int64
company_location    object
company_size        object
```