A Study of Salaries in Artificial Intelligence, Machine Learning and Data Science Globally

Instructor: Tran Thi Thanh Diu[1]
Student 1: Nguyen Minh Triet (523H0107)[1]
Student 2: Nguyen Phuc Toan (523H0185)[1]

[1] Ton Duc Thang University, Department of Computer Science

Abstract :

In the rapidly growing fields of Artificial Intelligence (AI), Machine Learning (ML), and Data Science, understanding salary trends and recruitment patterns is increasingly important. This report analyzes the *Global Salaries in AI, ML, Data Science* dataset, containing 57,194 records across 11 variables, to provide a comprehensive overview of compensation patterns. The study covers exploratory data analysis, handling of missing and duplicate entries, outlier detection and treatment, and examination of the probability distribution of salaries. Based on these analyses, the report presents key insights into global salary structures within these domains.

Index Terms: Exploratory Data Analysis, Outlier Detection, Probability Distribution, Hypothesis Testing, Correlation, Multiple Linear Regression.

1 Overview

This study investigates global salary trends for AI, ML, and Data Science roles, focusing on the factors that influence compensation and enabling predictive modeling. Specifically, we aim to:

a) Examine salary patterns across countries and job positions.
b) Understand how experience, company size, location, remote work, and employee background relate to salary.
c) Analyze the impact of recent trends in remote work on compensation.

Our analysis is guided by the following questions:

1. How does an employee's experience level (Entry, Mid, Senior, Executive) affect salary?
2. Has the rise of remote work, especially post-COVID-19, significantly influenced salaries?
3. What other factors most strongly drive salary differences?

By conducting this study, we seek to:

1. Provide clear answers to these questions.
2. Reveal the relationships between key factors and salary.
3. Develop a multiple linear regression model to predict salary based on experience, company size, remote work, and other relevant parameters.

2 Exploratory Data Analysis

2.1 Data Summary

2.1.1 Number of Records and Variables For this study, we chose the dataset *"Global Salaries in AI, ML, Data Science"*, which is collected from reputable international recruitment sources and published on Kaggle.

The dataset contains 57,194 records. Each record includes detailed information on job title, salary, country, experience level, contract type, and other relevant factors.

As such, the sample size is sufficient enough to ensure representativeness and statistical reliability for advanced analysis such as probability distribution assessment, statistical testing, outlier removal, and regression modeling.

The dataset comprises 11 variables (columns), as described in Table 1.

2.1.2 Data Types The variables in the dataset fall into two major groups: numerical and categorical.

Numerical variables represent quantitative measurements that support descriptive statistics such as mean, median, and standard deviation. They can be either continuous—values measured on a scale, or discrete—countable values without fractional components. In this dataset, the numerical variables are:

1. `work_year`: the year in which the salary was reported, used to analyze temporal salary trends.
2. `salary`: the original reported salary, expressed in the currency specified by `salary_currency`.
3. `salary_in_usd`: the salary converted to USD; this is the primary variable for statistical analysis, distribution assessment, and modeling.

Categorical variables, in contrast, represent groups, labels, or classifications without inherent numerical meaning. They can be divided into two subtypes:

1. Nominal: categories without an intrinsic order.
2. Ordinal: categories with a meaningful, ranked order.

The categorical variables in this dataset include:

1. `experience_level` (Ordinal): employee seniority (Entry, Mid, Senior, Executive).
2. `employment_type` (Nominal): contract type (Full-time, Part-time, Contract, Freelance).
3. `job_title` (Nominal): job position (e.g., Data Scientist, ML Engineer).
4. `employee_residence` (Nominal): country where the employee resides.
5. `remote_ratio` (Ordinal): degree of remote work (0%, 50%, 100%); treated as ordinal since the categories represent increasing levels of remote work.
6. `company_location` (Nominal): country where the company is based.

Table 1
Overview of dataset variables, their types, meanings, and roles in the analysis.

Variable	Data Type	Meaning	Purpose in Analysis
`work_year`	`int`	Year in which the salary was reported (2020–2024)	Track salary changes over time and evaluate temporal trends
`experience_level`	`enum[4]`	Employee seniority: Entry, Mid, Senior, Executive	Quantify salary differences across experience groups
`employment_type`	`enum[4]`	Employment arrangement: Full-time, Part-time, Contract, Freelance	Explore whether alternative contract types differ from full-time roles in salary
`job_title`	`string`	Reported job position (e.g., Data Scientist, ML Engineer)	Compare salaries across roles and characterize experience-level patterns
`salary`	`float`	Salary in original currency	Used for historical reference before conversion
`salary_currency`	`string`	Currency of the reported salary (USD, EUR, GBP, etc.)	Enables proper conversion to USD for cross-country comparability
`salary_in_usd`	`float`	Salary converted to USD using annual average exchange rates	Key variable for descriptive stats, distribution analysis, outlier removal, and regression
`employee_residence`	`string`	Country where the employee resides	Assess regional salary variations and geographic patterns
`remote_ratio`	`enum[3]`	Degree of remote work: 0% (on-site), 50% (hybrid), 100% (remote)	Measure the influence of remote work intensity on salary
`company_location`	`string`	Country where the employer is based	Compare salaries by company geography and economic region
`company_size`	`enum[3]`	Size category: Small (<50), Medium (50–250), Large (>250)	Analyze how organizational scale relates to compensation

7. `company_size` (Ordinal): organization size (Small, Medium, Large).
8. `salary_currency` (Nominal): currency used in the original salary report.

Understanding this distinction is essential during preprocessing—for example, encoding ordinal variables to preserve their rank, or handling nominal categories appropriately when analyzing relationships and fitting regression models.

2.1.3 Initial Observations We display the first five records from the dataset to get a sense of the overall patterns that we work with:

```
[10]: df.head()
```

[10]:

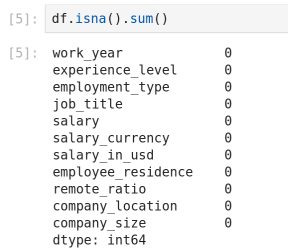
	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2024	MI	FT	Developer	168276	USD	168276	US	0	US	M
1	2024	MI	FT	Developer	112184	USD	112184	US	0	US	M
2	2024	EN	FT	Developer	180000	USD	180000	US	0	US	M
3	2024	EN	FT	Developer	133500	USD	133500	US	0	US	M
4	2024	EN	FT	Developer	122000	USD	122000	US	0	US	M

Figure 1. First five records of the dataset the showing variable values.

From Figure 1, several preliminary characteristics can be observed:

1. All entries' job titles contain "Developer", suggesting that while the dataset contains many technical roles, the first records show only one particular title.
2. Experience levels include Entry and Mid, demonstrating a diverse distribution of employee seniority.
3. All entries list full-time as the employment type, which is typical in the technology sector.
4. All of the salary records are reported in USD without conversion, meaning the original currency matches the converted currency in these records.
5. The remote ratio is 0 for all entries, indicating fully on-site work arrangements.
6. Employee residence and company location are both "US", showing that the first records represent the United States labor market.
7. The company size is consistently medium-sized (M, 50–250 employees), reflecting a workplace environment typical of mid-scale companies.

2.1.4 Handling Missing Data A check for missing values reveals that there are no missing values (NA) in any variable, so no further preprocessing is required. As shown in Figure 2, the output of `df.isna().sum()` confirms the absence of missing data.



```
[5]: df.isna().sum()
```

[5]:

```
work_year             0
experience_level      0
employment_type       0
job_title             0
salary                0
salary_currency       0
salary_in_usd         0
employee_residence    0
remote_ratio          0
company_location      0
company_size          0
dtype: int64
```

Figure 2. Summary of missing values in the dataset (all variables have zero missing entries).

2.1.5 Handling Duplicate Data To determine the number of duplicate entries, we use `df.duplicated().sum()`. The result revealed a significant number of duplicate records in the original dataset. After applying `df.drop_duplicates()`, the number of records was substantially reduced (57,194 → 29,883), likely due to aggregation from multiple sources or web scraping processes, as shown in Figure 3.

Removing duplicates ensures that: 1) descriptive statistics for salaries are not artificially inflated or biased; 2) probability distribution analysis are performed on truly independent observations; 3) outlier detection is not affected by repeated entries; 4) analytical or

```
[7]: df.duplicated().sum()

[7]: np.int64(29883)

[8]: df = df.drop_duplicates()
     df.shape

[8]: (27311, 11)
```

Figure 3. Duplicate data handling: detecting duplicates using `df.duplicated().sum()`, then removing with `df.drop_duplicates()`, and finally the new dataset shape after cleaning.

predictive models are not biased by replicated data.

2.2 Descriptive Statistics

After cleaning the dataset and removing duplicate entries, 27,311 records remain. Next, descriptive statistical analysis was performed for the numerical variables: work year, salary (original and in USD), and remote ratio.

```
[9]: df.describe()
```

[9]:

	work_year	salary	salary_in_usd	remote_ratio
count	27311.000000	2.731100e+04	27311.000000	27311.000000
mean	2023.717989	1.633512e+05	152807.917286	26.511296
std	0.603236	2.892565e+05	75863.899248	43.860317
min	2020.000000	1.400000e+04	15000.000000	0.000000
25%	2024.000000	9.840000e+04	98290.000000	0.000000
50%	2024.000000	1.420000e+05	141340.000000	0.000000
75%	2024.000000	1.950000e+05	193300.000000	100.000000
max	2024.000000	3.040000e+07	800000.000000	100.000000

Figure 4. Descriptive statistics for numerical variables using `df.describe()`.

In Figure 4, it can be observed that the numerical variables exhibit uneven, asymmetric distributions, particularly salary (USD), where the difference between mean (≈ 150,000 USD) and max salary (800,000 USD), which means that the mean is closer to the left side, pointing to a plausible right-skewed distribution. The remote ratio indicates that on-site work remains dominant, while fully remote positions account for roughly 25% of the dataset. The work year is heavily concentrated in 2024, reflecting the recency and reliability of the information.

With this, we have a solid general insight to begin the study.

2.3 Data Visualization

2.3.1 Salary Histogram and KDE Plot The histogram + KDE plot provides a clear representation of the salary distribution. In Figure 5, it is heavily right-skewed, with most of its values falling between 100,000–200,000 USD. This matches the median (≈ 150,000 USD), which represents the typical salary in the dataset.

A long right tail appears for sparse but very large salaries (300,000–400,000 and up to 800,000 USD). These high values pull the mean upward (≈ 160,000 USD) and push it to the left, creating a clear gap between mean and median and confirming that the distribution is far from normal. Because of this skewness, methods such as Z-score which rely on normality assumption are unreliable; and MAD or log-transformation are better suited for this.

The KDE curve reinforces the same pattern: a tall peak around 100,000–200,000 USD and a slow-decaying tail on the right. This

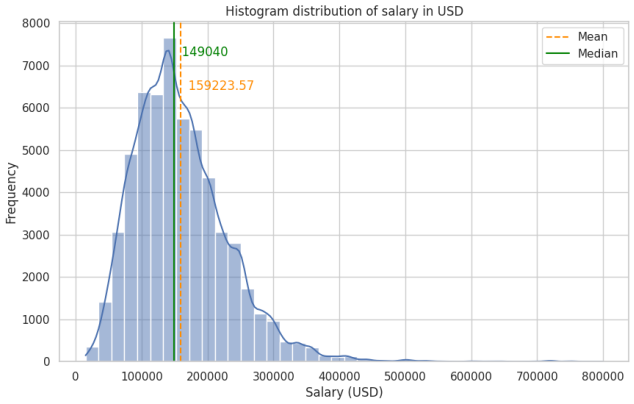


Figure 5. Combined histogram and KDE plot of salary (USD), illustrating the right-skewed distribution and presence of high-value outliers.

shape highlights the small group of extremely high earners, mainly high-experienced employees, and the influence of outliers on summary statistics.

Overall, the distribution's heavy right skew suggests that we need to use more robust techniques when analyzing or modeling this dataset.

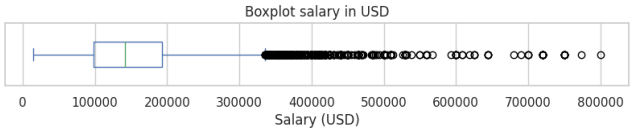


Figure 6. Boxplot of salary (USD) illustrating the interquartile range and high-value outliers.

2.3.2 Salary Boxplot The salary boxplot (Figure 6) highlights the spread of salaries and makes outliers easy to spot. The box represents the interquartile range (IQR), which spans from the first quarter (Q1, 25% ≈ 100,000 USD) to the third quarter (Q3, 75% < 200,000 USD). The whiskers (the lines outside the box) stretch from roughly 15,000 USD up to >300,000 USD, covering the typical salary range.

Beyond the upper whisker, a cluster of high-value points appears, representing salaries from >300.000 to 800,000 USD. These extreme values match the strong right skew seen in Figure 5, and are potentially the outliers that we need to handle using more robust methods.

Overall, the boxplot shows both the typical central range of salaries and the substantial presence of large outliers.

2.3.3 Experience Level Countplot Figure 7 shows how experience levels (Entry, Mid, Senior, Executive) distribute across the top job titles. Executive dominates overall with >6,000 records max in the busiest titles, while Senior usually follows with <3,000 max.

Surprisingly, Entry actually exceeds Mid in several roles; most notably for Data Analyst, where it even surpasses Senior. This reflects the rise of internships and university-backed training pipelines, making companies more open to early-career hires. Still, across all roles, higher experience levels remain the most in-demand.

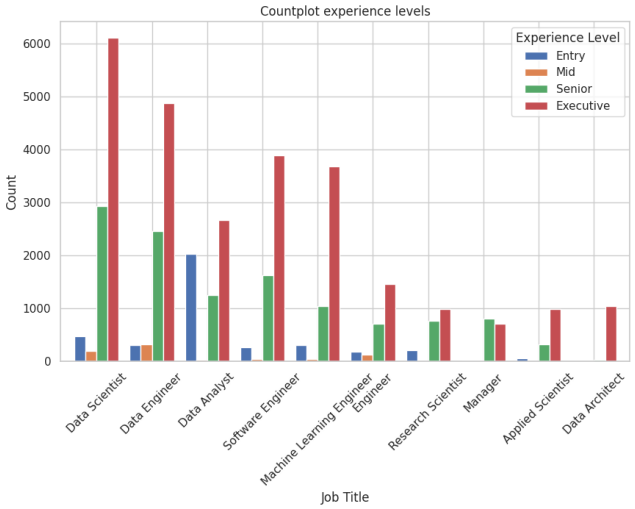


Figure 7. Countplot of employees by experience level, grouped by the top 10 job titles.

Overall, this analysis shows that the dataset primarily focuses on professionals and high-level executives, rather than newcomers or interns.

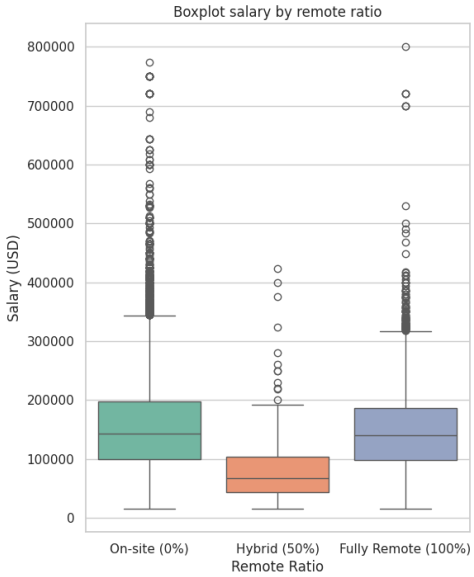


Figure 8. Salary distribution across remote-work categories (On-site, Hybrid, and Fully Remote) based on the remote-ratio variable.

2.3.4 Remote Ratio Boxplot Finally, the remote ratio boxplot (Figure 8) shows a pretty distinct pattern across the three remote-work setups. On-site and fully-remote roles both have noticeably higher salaries, with medians sitting well above the hybrid group. Hybrid salaries drop sharply, forming a much shorter box and whiskers.

On-site and fully remote also share a similar spread: wide IQRs, long upper whiskers, and numerous high-salary outliers (up to 800,000 USD). Hybrid, meanwhile, has fewer extreme values and a lower overall range.

We conclude that hybrid roles consistently pay less, while on-site and fully remote offer more comparable and higher compensation, creating a unique V-shape.

2.4 Outlier Detection and Removal

2.4.1 Theoretical Basis There are four methods we tested for this study: IQR, Z-score, MAD, and Isolation Forest.

(1) Interquartile Range (IQR)—IQR identifies outliers by assuming that the majority of data points lie within the box as we saw with Figure 6, which is from Q1 to Q3:

$$\text{Lower fence} = Q1 - 1.5 \times IQR, \quad \text{Upper fence} = Q3 + 1.5 \times IQR$$

Data points outside this range are considered outliers.

However, IQR works best for approximately symmetric distributions, which in our dataset is not reflected due to the heavy right skew of salary.

Applying this method removed 6,646 records. However, with that right skew, it likely removed some legitimate high salaries (Senior or Executive experiences), making it too aggressive for this dataset.

```
[42]: target_col = "salary_in_usd"

      q1 = df[target_col].quantile(0.25)
      q3 = df[target_col].quantile(0.75)

      iqr = q3 - q1

      lower = q1 - 1.5 * iqr
      upper = q1 + 1.5 * iqr

      df_iqr = df[(df[target_col] >= lower) & (df[target_col] <= upper)]

      print(f"Cleaned: {len(df)} → {len(df_iqr)}")
      Cleaned: 57194 → 50548
```

Figure 9. Before-and-after sample size after removing outliers using the IQR method.

(2) Z-score—Assuming that the data is approximately normally distributed, Z-score is calculated by dividing the difference between the data point x (the single value of a variable) and the mean μ by the standard deviation σ:

$$Z = \frac{x - \mu}{\sigma}$$

By the empirical rule, roughly 99.7% of normally distributed data falls within $\pm 3\sigma$ of the mean, meaning values with $|Z| > 3$ are considered outliers.

Using this method, 523 records were identified as outliers, showing that Z-score is less aggressive than IQR. However, due to the normality assumption, the strong right-skew of salary reduces the reliability of Z-score significantly.

```
[41]: target_col = "salary_in_usd"

      z = stats.zscore(df[target_col])

      df_z = df[np.abs(z) < 3]

      print(f"Cleaned: {len(df)} → {len(df_z)}")
      Cleaned: 57194 → 56671
```

Figure 10. Before-and-after sample size after removing outliers using the Z-score method.

(3) Median Absolute Deviation (MAD)—MAD (also known as the Modified Z-score method) is a robust outlier detection technique. Unlike the standard Z-score, which relies on the mean and standard deviation and is therefore highly sensitive to extreme values, MAD uses the median and median absolute deviation, making it far more stable under skewed or heavy-tailed data.

Experimentally, this method removed 522 records, striking a good balance between eliminating true outliers and preserving legitimately high salaries (Senior or Executive roles). For this dataset, MAD proved to be the most appropriate method.

```
[43]: target_col = "salary_in_usd"

median = df[target_col].median()
mad = np.median(np.abs(df[target_col] - median))
mod_z = 0.6745 * (df[target_col] - median) / mad

df_mad = df[np.abs(mod_z) < 3.5]

print(f"Cleaned: {len(df)} → {len(df_mad)}")
```

```
Cleaned: 57194 → 56672
```

Figure 11. Before-and-after sample size after removing outliers using the MAD method.

(4) Isolation Forest—Isolation Forest is an anomaly-detection method that works by randomly splitting the data into branches (trees). Outliers are easier to isolate, meaning they require fewer splits, while normal points need more. This makes the algorithm effective for high-dimensional datasets.

Despite its strengths, the method is less interpretable in a statistical context, and its results depend heavily on the `contamination` parameter, which is the assumed proportion of outliers. Here, we set contamination to 0.02 (2%), which led the model to remove 1,079 records. This suggests that the method likely flagged additional false positives compared to simpler statistical techniques.

```
[47]: target_col = "salary_in_usd"

iso = IsolationForest(contamination=0.02, random_state=42)
iso_labels = iso.fit_predict(df[[target_col]])

df_iso = df[iso_labels == 1]

print(f"Cleaned: {len(df)} → {len(df_iso)}")
```

```
Cleaned: 57194 → 56116
```

Figure 12. Before-and-after sample size after removing outliers using the Isolation Forest method.

2.4.2 Experiment Results Lastly, we composed all of the values from our tests into Table 2 and plotted the boxplots to compare with the original (Figure 13).

Table 2
Comparison of outlier detection methods, showing the number of records removed by each technique.

Method	No. of Records Removed
IQR	6,646
Z-score	523
MAD	522
Isolation Forest	1,079

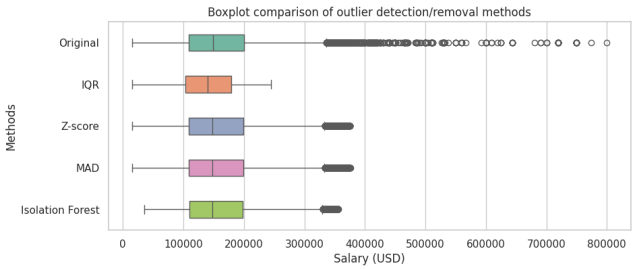


Figure 13. Boxplots showing the effect of different outlier removal methods on salary distribution.

2.4.3 Conclusion MAD is the preferred choice for outlier detection and removal because it handles skewed data effectively while preserves the natural distribution of salary, removes outliers without being too aggressive, and provides the most stable and interpretable results among the four methods.

3 Probability Distribution Analysis

In this task, we examine whether the key variable **salary (USD)** follows a known probability distribution. As noted in Section 2.1.2, this variable has strong variability and extreme values, which makes distributional assessment essential.

It is noting that we avoid removing outliers with MAD before this analysis, which will be explained in Section 3.3.

3.1 Initial Observations

Recall from the previous descriptive analysis (Figure 4), salaries exhibit a strong right skew, caused by a few extremely high values reaching up to 800,000 USD while the majority cluster near 150,000 USD.

The skew is more evident when we re-examine the histogram and KDE (Figure 5): the central mass roughly forms a bell curve, which is typical of a normal distribution, but the long, sparse right tail pulls the distribution away from symmetry. This combination of a bell-shaped core and heavy right tail is characteristic of variables that follow a **log-normal** distribution rather than a normal distribution.

To further assess the skew, we use a Q–Q plot, which compares the quantiles of the observed salary data with those of a theoretical normal distribution:

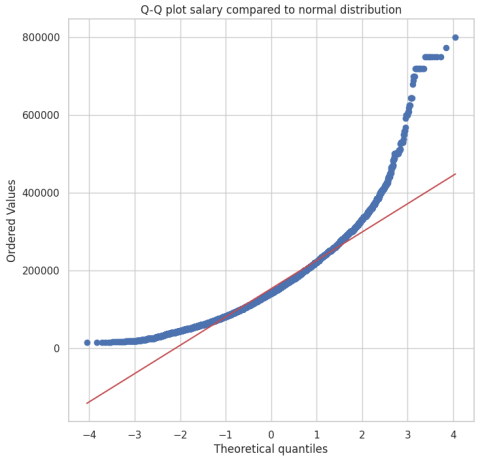


Figure 14. Q–Q plot comparing salary quantiles to a theoretical normal distribution.

In the plot, the points curve upward rather than following the straight reference line, particularly in the right tail. This deviation confirms the presence of strong right skew and reinforces that the salary distribution is far from normal.

3.2 Normality Tests

To formally evaluate normality, two statistical tests are applied:

1. *Sharpio–Wik Test*: evaluates how closely the data follow a normal distribution by comparing the sample's order statis-

tics (the sorted data points) to those expected under normality. If the data deviate substantially from this pattern, the test produces a small p value, which signals that the normality assumption should be rejected.

2. *Kolmogorov–Smirnov Test*: compares the empirical cumulative distribution function (ECDF) of the data, which shows the proportion of values below each point, with the theoretical CDF of a normal distribution. A large maximum difference (D statistic) leads to a small p value, indicating that the data are unlikely to be normally distributed.

The p values from both tests are shown in Figure 15.

```
[6]: sample = df["salary_in_usd"].sample(3_000, random_state=42)

     shapiro_stat, shapiro_p = stats.shapiro(sample)
     print("Shapiro-Wilk test p-value =", shapiro_p)

     scaled = (sample - sample.mean()) / sample.std()
     kstest_stat, kstest_p = stats.kstest(scaled, 'norm')
     print("Kolmogorov-Smirnov test p-value =", kstest_p)

     Shapiro-Wilk test p-value = 3.811588354422095e-37
     Kolmogorov-Smirnov test p-value = 2.355048740125661e-11
```

Figure 15. p values from the Shapiro–Wilk and Kolmogorov–Smirnov tests for salary normality. Values near 0 indicate strong rejection of the normality assumption.

All p values are extremely small, confirming that the salary distribution deviates significantly from normality and that the normality hypothesis should be rejected.

3.3 Log-transformation

To reduce skewness, we apply the natural logarithm transformation:

$$\text{salary'} = \ln(\text{salary})$$

After transformation, the histogram of ln(salary) becomes approximately symmetric, and the Q–Q plot aligns much more closely with the theoretical normal line. The spread of salaries also becomes more consistent across the distribution.

A crucial observation is that if outliers are removed before the log-transformation, the distribution becomes left-skewed. Therefore, log-transforming must be performed prior to MAD outlier removal to maintain approximate symmetry.

3.4 Conclusion

All analyses, visual inspection, statistical testing, and log-transformation indicate that salary follows a log-normal distribution rather than a normal distribution. The raw data shows strong right skew due to high-salary outliers, while the log-transformed data is approximately symmetric and closely aligns with a theoretical normal distribution.

Practical implications:

1. Use median and IQR instead of mean and standard deviation for descriptive summaries.
2. Log-transform salary values before modeling to improve residual behavior and support parametric analyses.
3. Apply non-parametric tests on the original scale, unless log-transformed values are used.
4. Remove outliers after log-transformation to preserve symmetry and avoid artificial skew.

Recognizing the log-normal nature of salary ensures more accurate modeling, robust statistical inference, and meaningful descriptive analysis.

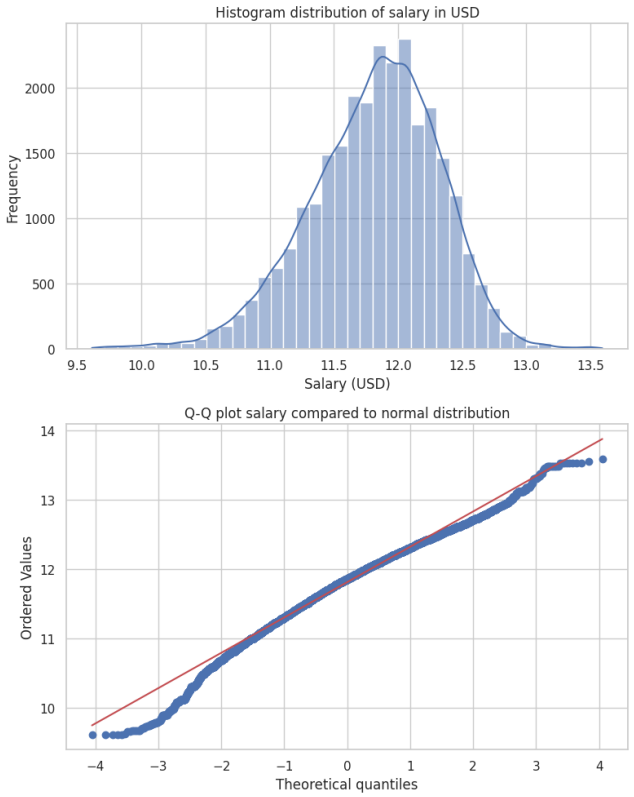


Figure 16. Histogram and Q–Q plot of log-transformed salary values.

4 Hypothesis Testing

In this task, we will perform **hypothesis testing** by proposing a question relevant to the dataset and performing statistical evaluation to answer it.

To begin, we will formulate a question that aims to explore the relationship between remote work and salary, especially as working from home has become increasingly relevant after the COVID-19 pandemic:

> *"Does working fully remotely correlate to a higher salary than on-site or hybrid work?"*

Then, we choose a method to perform the test. There are three methods: t-test, Chi-square and ANOVA. After a quick survey, we will test our hypothesis using ANOVA, the reason for which will be explained in Section 4.1.

4.1 Theoretical Basis of ANOVA

Analysis of Variance (ANOVA) is a statistical method used to compare the means of three or more independent groups to determine whether at least one group differs significantly from others. ANOVA effectively reduces the risk of Type I error (rejecting the null hypothesis H_0 when it is actually true) while allowing the comparison of three or more categories (*e.g.* groups) per variable, compared to a t-test.

We test our hypothesis using ANOVA by performing an **F-test**:

$$F^* = \frac{\text{MSR}}{\text{MSE}} = \frac{\text{variance between groups}}{\text{variance within groups}}$$

We notice that $F^* \propto \frac{1}{\text{MSE}}$, meaning a higher F-statistic value indicates a greater likelihood that the group means are not equal.

Then we calculate **p-value** using the cumulative distribution function (CDF) of the F-distribution, and finally reject H_0 if $p < \alpha$ (where $\alpha = 1 - \text{CI}$, commonly 0.05).

ANOVA works on three assumptions about the dataset: 1) observations in each group are independent; 2) the data within each group is approximately normally distributed; 3) The variances across groups are roughly equal.

We chose this method for this task because:

1. We are testing the remote ratio variable, which is a categorical feature with three groups (on-site, hybrid, and fully remote), so t-test is unsuitable.
2. Our goal is to determine the effect of working remotely on salary, which involves comparing the group means, whereas the Chi-square test is used for independence testing.
3. Our dataset has a sufficient sample size ($n > 31$) for the distribution to be approximately normal.

4.2 Experiment Results

We begin forming our null and alternate hypotheses:

1. H_0: The mean salary is the same across all remote ratios.
2. H_1: There is at least one remote ratio that has a significantly different mean salary.

The hypothesis is supported if we reject H_0, *i.e.* $p < \alpha$. We set $\alpha = 0.05$ (95% confidence interval).

We perform a one-way ANOVA test using Python's `scipy.stats.f_oneway()` function, which conveniently returns the F-statistic and p-value for the comparison:

```
[6]: groups = [g["salary_in_usd"] for _, g in df.groupby("remote_ratio")]

f_stat, p_value = stats.f_oneway(*groups)
print(
    f"F-statistic = {f_stat}",
    f"\np-value = {p_value}",
)
```

```
F-statistic = 199.2627168301049
p-value = 1.2267506437579236e-86
```

Figure 17. F-statistic and p-value results from performing one-way ANOVA test using `scipy.stats.f_oneway()`.

With $p \approx 1.22 \times 10^{-86} \ll \alpha$, we reject H_0, meaning that, with 95% confidence, there is a significant difference between the salary means of the different remote ratio groups.

4.3 Observations & Conclusion

The significant difference is further illustrated in Figure 8, where hybrid working (50% on-site, 50% remote) has a noticeably lower IQR and median, resulting in a much shorter and lower box. However, the answer to our proposed question is not entirely straightforward, as working fully remotely has roughly the same median salary as working on-site. Furthermore, as noted, the work year was recorded from 2020–2024, which introduced a bias toward hybrid or fully remote work in recent years.

From a practical standpoint, this trend makes sense because:

1. The demand for efficient remote work has increased in recent years, especially during the COVID-19 pandemic when employees were restricted from going outside due to quarantine measures.
2. Many employees prefer working from home because it offers great flexibility and comfort, which is particularly common in software development, data analysis, and other technology or science-related jobs.
3. Improvements in infrastructure and technology have made remote work not only feasible but also as effective and efficient as working directly in an office.

However, this does not imply that remote work will fully replace offices, as face-to-face interaction and social activities still hold an important role in boosting employee confidence and supporting their work.

Conclusion: While we cannot provide a definitive answer to our proposed question, our analysis confirms a significant difference in salary across remote ratio groups and highlights the growing importance and prevalence of fully remote work for many employees.

5 Correlation Analysis

In this task, we will analyze the correlations between multiple variables/features in our dataset. There are two correlation coefficients we can utilize: Pearson and Spearman, but we will focus only on Spearman, which will be discussed later in Section 5.1.

5.1 Theoretical Basis of Spearman Correlation

The *Spearman rank correlation coefficient* (ρ) measures the statistical dependence between two variables. Unlike the *Pearson correlation coefficient* r, it does not assume linearity or a normal distribution. Instead, it evaluates whether the relationship between two variables is monotonic—that is, as one variable increases, the other consistently increases or decreases, though not necessarily at a constant rate.

ρ works especially well for **ordinal categorical** variables and is robust to outliers, since it relies on ranks rather than raw values.

Assuming a dataset with a sample size of n, ρ is computed by ranking the values of each variable:

$$\rho \in [-1, 1] = 1 - \frac{6 \times \text{sumsq}(R(X_i) - R(Y_i))}{n(n^2 - 1)}$$

where:

- $\text{sumsq}(\cdot)$: sum of squares of its elements.
- $R(\cdot)$: the rank of a given variable value.
- X_i, Y_i: the value of two variables X and Y of the i^{th} observation.

The meaning of ρ is as follows:

- $\rho = 1$: perfect positive monotonic relationship (upward trend).
- $\rho = -1$: perfect negative monotonic relationship (downward trend).
- $\rho = 0$: no monotonic relationship (random fluctuations).

5.2 Experiment Results

Our target variable of choice for this task is salary (USD), and we are going to compare it with four different feature variables: work year, remote ratio, experience level, and company size.

Since both experience level and company size's data types are categorical strings, we need to encode them in the correct ordinal order.

```
[6]: ordinal_encodes = {
         "experience_level": {"EN": 1,"MI": 2,"SE": 3,"EX": 4},
         "company_size": {"S": 1, "M": 2, "L": 3},
     }

     for col, encodes in ordinal_encodes.items():
         df[f"_{col}"] = df[col].map(encodes)

     df[["experience_level", "_experience_level", "company_size", "_company_size"]].tail()
```

[6]:

	experience_level	_experience_level	company_size	_company_size
57189	SE	3	L	3
57190	MI	2	L	3
57191	EN	1	S	1
57192	EN	1	L	3
57193	SE	3	L	3

Figure 18. Ordinal encoding for categorical variables, showing the last five entries, with before and after comparison.

We can begin calculating ρ for each combinations of feature and target variables. For this, we use Python's `scipy.stats.spearmanr()`, which conveniently returns both ρ and p (the probability of seeing our observed statistic, as mentioned in Section 4.1). The results are then shown in Figure 19.

```
[7]: feature_cols = ["work_year", "remote_ratio", "_experience_level", "_company_size"]
     target_col = "salary_in_usd"

[8]: for col in feature_cols:
         rho, p_value = stats.spearmanr(df[target_col], df[col])
         print(f"[{col}] Spearman rho = {rho}; p-value = {p_value}")

     [work_year] Spearman rho = 0.049996593119945254; p-value = 1.538695350683143e-16
     [remote_ratio] Spearman rho = -0.03447090858291726; p-value = 1.2707063690329234e-08
     [_experience_level] Spearman rho = 0.354728047363505; p-value = 0.0
     [_company_size] Spearman rho = -0.014309101645989976; p-value = 0.018235690366930526
```

Figure 19. ρ and p values from performing Spearman correlation analysis using `scipy.stats.spearmanr()`.

All variables have $p < \alpha = 0.05$, meaning the observations we observe are statistically significant rather than random. In other words, each feature shows a meaningful monotonic relationship with salary instead of arising from noise in the data.

Furthermore, experience level has the highest positive ρ, which means it has a plausible and stable upward trend with salary; meanwhile other variables have $\rho \approx 0$, corresponding unstable fluctuations, especially for remote work which matches our previous observations that hybrid work is significantly lower.

5.3 Plotting Visualizations

A correlation heatmap (Figure 20) is useful to display all ρ values of variable pairs in a single matrix.

While our main goal is to examine the monotonic relationship with salary, it is worth noting that both remote ratio and work year show the highest negative ρ. This further supports our earlier explanations regarding the bias toward increased remote work in recent years.

To better visualize monotonicity, we revisit the boxplots, this time adding regression lines to show trends or fluctuations (Figure 21).

All boxplots match our observation that p values are extremely small, indicating that the detected trends are meaningful rather than random noise.

Key insights:

1. Experience level shows a roughly linear upward trend, consistent with having the highest ρ among the four variables tested.
2. The regression lines of both remote ratio and company size form a unique V shape, aligning with its ρ being close to 0.

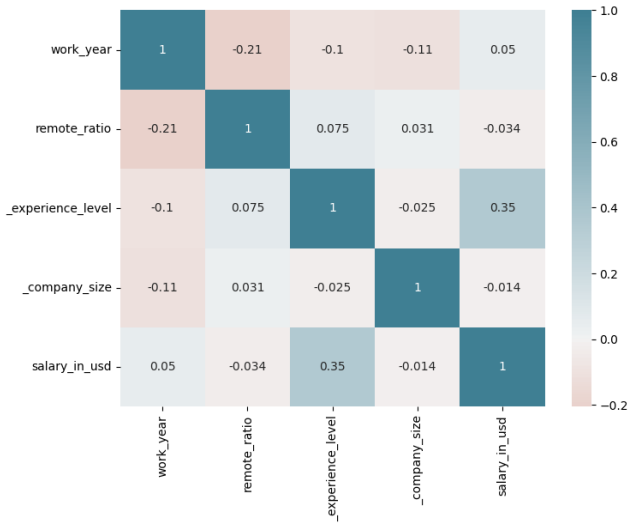


Figure 20. Correlation heatmap of ρ between all variable pairs.

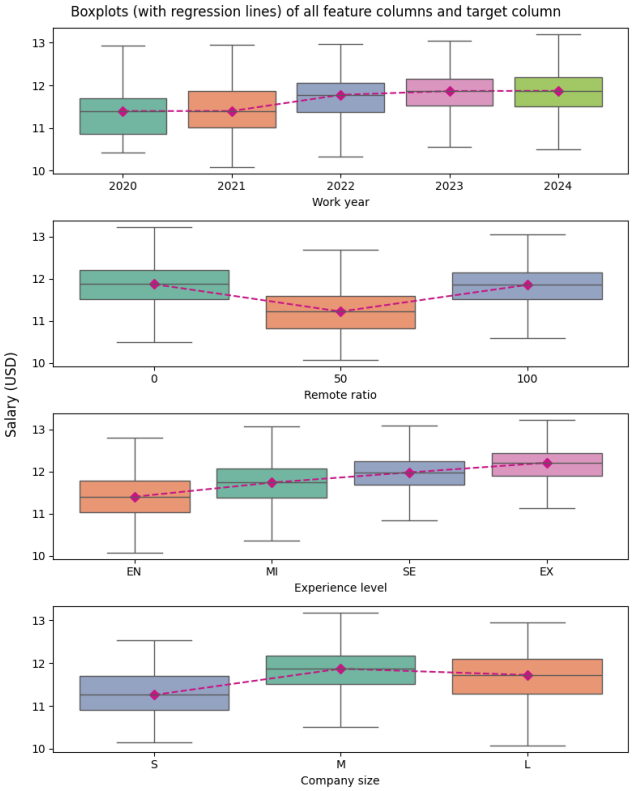


Figure 21. Boxplots of all features against salary, with dashed regression lines representing trends and monotonicity.

3. Work year also shows an upward trend, though not consistently.

5.4 Observations & Conclusions

Our statistical results reflect practical real-world patterns: salary increases as employees gain more experience during their work, and companies generally prefer hiring more experienced employees for higher-responsibility roles, naturally resulting in higher pay; remote work has become increasingly relevant and serves as a flexible, viable working model; work year correlates with salary due to skill growth and inflation; and company size tends to scale with salary, but visualizations indicate that the change from M to L is downward.

Conclusion: Our correlation analysis shows that salary has monotonic relationships with work year, and experience level, while remote ratio and company size show little to no monotonicity.

6 Multiple Linear Regression

In this task, we aim to understand how multiple factors together influence salary. *Multiple Linear Regression* (MLR) models the relationship between a continuous target variable (salary) and several input features simultaneously, allowing us to quantify feature effects, make predictions, and understand multivariate relationships.

7 Theoretical Basis

MLR is an extended version of *Linear Regression* used to predict a continuous target variable y for multiple input features $X_1, X_2, \ldots, X_p$. It provides a way to quantify the effect of each feature on the target while controlling for the others; make predictions on new, unseen data; and understand complex relationships in multivariate datasets.

The MLR model can be written as:

$$y^i = \beta_0 + X_1^i\beta_1 + X_2^i\beta_2 + \cdots + X_p^i\beta_p + \epsilon^i$$

where:

- X^i, y^i: independent (features) and dependent (target) variables at the i^{th} observation.
- β: regression coefficients (parameters).
- β_0: intercept (bias).
- ϵ^i: random error term.

The goal of MLR is to find $\beta_0, \beta_1, \ldots, \beta_p$ that minimizes the mean squared error (MSE) (*i.e.*, fitting the model):

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{n}\|y - \hat{y}\|_2^2$$

where:

- y_i: ground truth.
- $\hat{y_i}$: model prediction.

Once finished, we can evaluate its performance using various metrics:

1. Mean Absolute Error (MAE): $\text{MAE} = \frac{1}{n}\|y - \hat{y}\|$
2. Mean Squared Error (MSE): $\text{MSE} = \frac{1}{n}\|y - \hat{y}\|_2^2$
3. Coefficient of Determination (R^2): $R^2 = 1 - \frac{\|y-\hat{y}\|_2^2}{\|y-\bar{y}\|_2^2}$

However, for MLR to be viable when predicting, a few assumptions are required: 1) the relationship between features and the target is roughly linear; 2) observations are independent; 3) the variance of errors is constant; 4) ϵ_i are approximately normally distributed; 5) features are not highly correlated with each other.

7.1 Experiment Results

Before fitting an MLR model, we need to preprocess our data to ensure that all variables contribute to the final prediction and that discrepancies like outliers and duplicates will not affect the performance negatively:

Step 1: Drop redundant columns (`salary, salary_currency`), only `salary_in_usd` is needed.
Step 2: Remove duplicates with `df.drop_duplicates()` and outliers using MAD.
Step 3: Ordinal encode experience level and company size.
Step 4: Create X and y for features and target variables, and log transform y to reduce right skew.
Step 5: Apply one-hot encoding to nominal categorical features.
Step 6: Split X and y into train and validation sets.

Finally, we fit `scikit-learn`'s `LinearRegression` model and evaluate the performance:

```
[11]:
y_hat = model.predict(X_test)

mae = mean_absolute_error(y_test, y_hat)
print(f"MAE =", mae)

mse = mean_squared_error(y_test, y_hat)
print(f"MSE =", mse)

r2 = r2_score(y_test, y_hat)
print(f"R2 score =", r2)
```

```
MAE = 0.32367433460127365
MSE = 0.16563423016070797
R2 score = 0.3277028307619283
```

Figure 22. MAE, MSE and R^2 scores of `LinearRegression` after fitting with preprocessed data.

From Figure 22, we notice that our model is under-performing, indicated by a low R^2 score. This makes sense, as many features do not have a linear relationship with salary, as seen with Figure 21. Still, the model is able to capture meaningful patterns rather than memorizing noise in the dataset.

7.2 Conclusion

All in all, we showed that a simple MLR can capture meaningful relationships between features and salary, though performance is limited due to non-linear dependencies. Future improvements could include removing highly correlated features or applying non-linear models such as deep learning / neural networks to better capture complex patterns.

8 Summary

This report presents a comprehensive analysis of salary data, exploring the factors that influence compensation in the context of experience, remote work, and organizational characteristics. The following key insights summarize our findings:

1. Experience level strongly correlates with salary. Employees at higher levels such as Senior and Executive consistently earn

more than those at Entry or Mid levels. Statistical analysis, including Spearman correlation and multiple linear regression, confirms a stable upward trend in salary with increasing experience.

2. The rise of remote work has introduced nuanced effects on salaries. ANOVA results show significant differences across remote ratios. Fully remote and on-site roles exhibit similar median salaries, both generally higher than hybrid positions. This suggests that while remote work has become more prevalent, its impact on pay varies depending on work structure.
3. Additional variables such as company size, work year, and job title also contribute to salary differences. Company size shows a moderate association, with larger organizations typically offering higher compensation, although some inconsistencies exist. Work year correlates positively with salary, reflecting experience growth and inflation trends.

Overall, the analysis indicates that salary is primarily driven by experience level, with remote work and organizational factors also playing significant roles. Recognizing these relationships allows for more informed decision-making regarding compensation, career planning, and organizational policies.

Acknowledgements

This research received support during the Data Analysis and Visualization (505067) course, instructed by Professor Tran Thi Thanh Diu, PhD at the University of Ton Duc Thang, Department of Computer Science.

Contributions

Table 3
Contributions of team members to this study.

Team Member	Contributions
Nguyen Minh Triet	Exploratory Data Analysis, Visualization, Outlier Detection and Removal, Probability Distribution Analysis, Presentation Writing
Nguyen Phuc Toan	Hypothesis Testing, Correlation Analysis, Modeling Multiple Linear Regression, Report Writing