April 2025

Statistics Report

CDC Placement Statistics For The Year 2023-24

Under guidance of: Prof. Somesh Kumar (Department of Mathematics)

Team Members:

1. Aayush Srivastava (23EE10080)
2. Aryan Yadav (23CS10003)
3. Ashutosh Ranjan (23CS10004)
4. Mayank Modi (23CS10089)

STATISTICS REPORT

ABOUT THE TOPIC

This project presents a statistical analysis of the CDC placement data of IIT Kharagpur for the academic year 2023-24. The objective is to study the overall distribution of placement offers and salary packages using descriptive statistics such as mean, median, standard deviation, skewness, and kurtosis. The analysis aims to determine the type of distribution the data follows (e.g., lognormal, gamma) and provide insights into the general trends of placement outcomes across the institute.

Source: Data obtained through RTI (Right to Information) request to IIT Kharagpur

STATISTICS REPORT

METHODOLGY

To study the statistical distribution of placement packages, we followed a structured approach involving data binning, frequency calculation, and cumulative analysis. The following steps were executed using Python, with relevant code snippets and graphs provided inline.

1. Data Binning

The raw salary data was categorized into uniform salary bands of ₹50,000 each, starting from ₹0 onwards. This binning continued consistently across the entire range of values, allowing for a detailed and structured analysis of the distribution.

```
bin_width = 50000
min_ctc = df['CTC'].min()
max_ctc = df['CTC'].max()
bins = np.arange(min_ctc, max_ctc + bin_width, bin_width)
```
✓ 0.0s

2. Frequency Distribution

For each bin, we calculated the number of students whose placement packages fell within that range. This gave us the frequency distribution, which was then plotted as a histogram for visual inspection

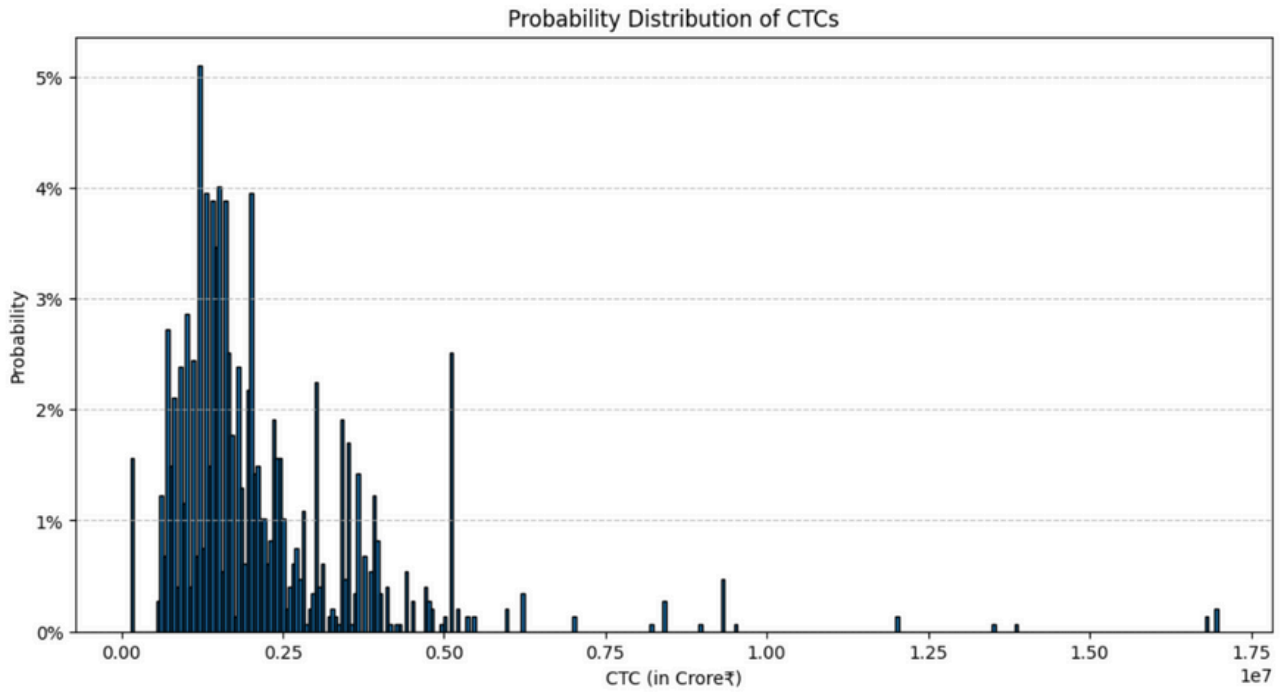
```
plt.figure(figsize=(12, 6))
n, bins, patches = plt.hist(df['CTC'], bins=bins, weights=np.ones(len(df)) / len(df), edgecolor='black')

plt.gca().yaxis.set_major_formatter(plt.FuncFormatter(lambda y, _: '{:.0%}'.format(y)))

plt.xlabel('CTC (in Crore₹)')
plt.ylabel('Probability')
plt.title('Probability Distribution of CTCs')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

STATISTICS REPORT

METHODOLGY(...CONTINUED)



3. Cumulative Distribution Function (CDF)

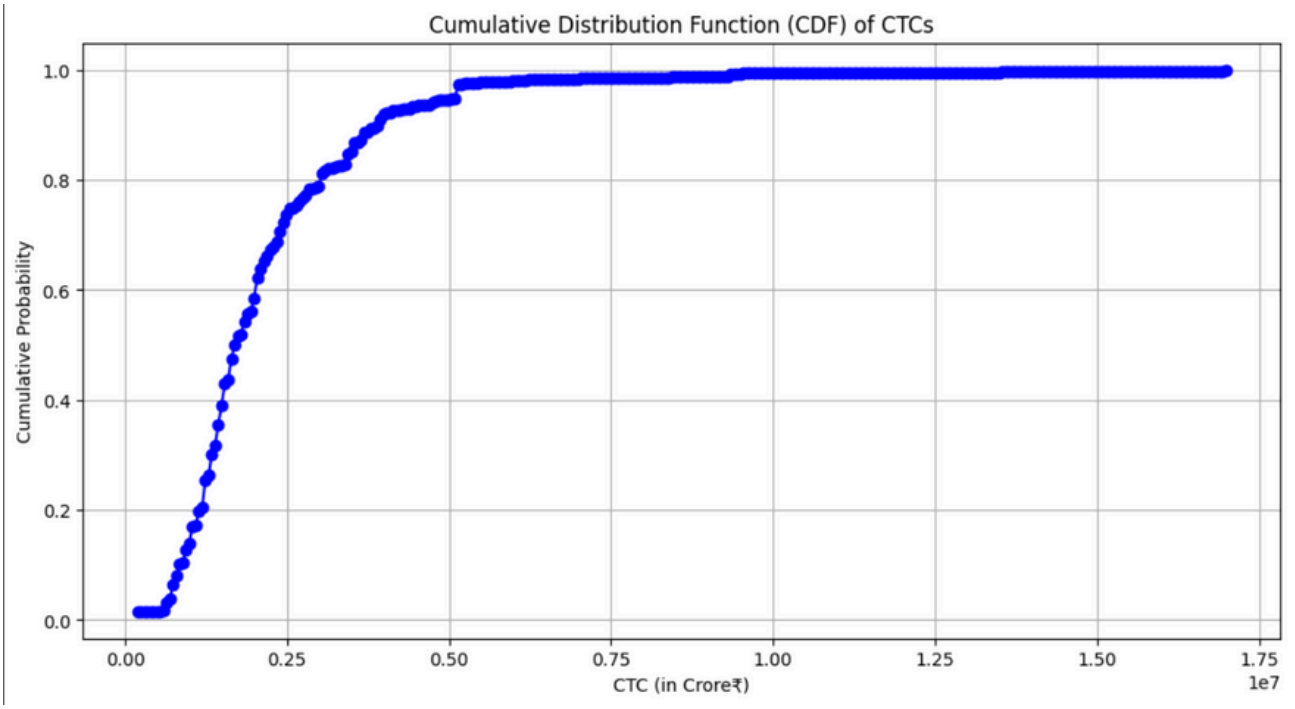
From the frequency data, we computed the cumulative frequency to obtain the CDF. This allowed us to assess how salary packages are distributed across the student population and to visually examine trends in the distribution.

```
hist, bin_edges = np.histogram(df['CTC'], bins=bins)
cum_counts = np.cumsum(hist)
cum_prob = cum_counts / cum_counts[-1]


plt.figure(figsize=(12, 6))
plt.plot(bin_edges[1:], cum_prob, marker='o', linestyle='-', color='b')
plt.xlabel('CTC (in Crore₹)')
plt.ylabel('Cumulative Probability')
plt.title('Cumulative Distribution Function (CDF) of CTCs')
plt.grid(True)
plt.show()
```

STATISTICS REPORT

METHODOLGY(...CONTINUED)



4. Fitting CDF against Theoretical Distributions

The empirical data thus obtained was fitted against some well know distribution such as Lognormal, Weibull, Gamma, Fisk, Beta etc to obtain the best fit.

```
from scipy.stats import lognorm, gamma, weibull_min, fisk, betaprime
```
✓ 0.0s

```
sorted_ctc = np.sort(ctc_data)
empirical_cdf = np.arange(1, len(sorted_ctc) + 1) / len(sorted_ctc)
distributions = {
    'Lognormal': lognorm,
    'Gamma': gamma,
    'Weibull': weibull_min,
    'Loglogistic': fisk,
    'Beta Prime': betaprime
}
```
✓ 0.0s

STATISTICS REPORT

METHODOLGY(...CONTINUED)

```
fitted_cdfs = {}
for name, dist in distributions.items():
    params = dist.fit(ctc_data)
    fitted_cdfs[name] = dist.cdf(x, *params)
```

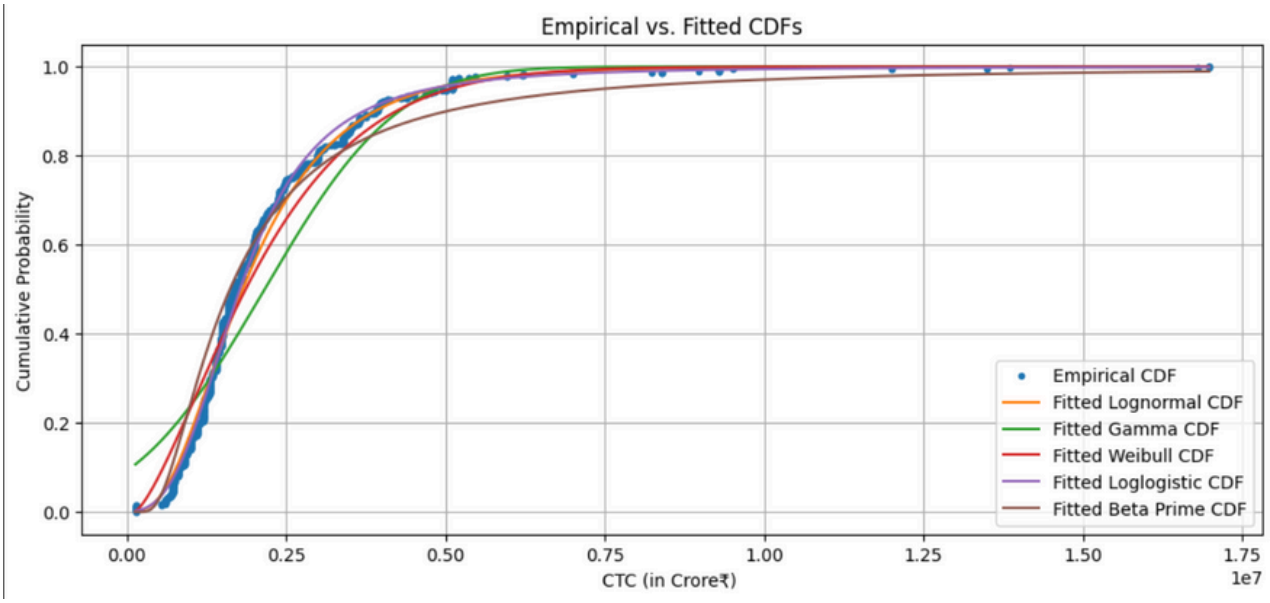
✓ 1.1s

```
plt.figure(figsize=(12, 8))

plt.plot(sorted_ctc, empirical_cdf, marker='.', linestyle='none', label='Empirical CDF')

for name, cdf in fitted_cdfs.items():
    plt.plot(x, cdf, label=f'Fitted {name} CDF')

plt.xlabel('CTC (in Crore₹) ')
plt.ylabel('Cumulative Probability')
plt.title('Empirical vs. Fitted CDFs')
plt.legend()
plt.grid(True)
plt.show()
```



STATISTICS REPORT

METHODOLGY(...CONTINUED)

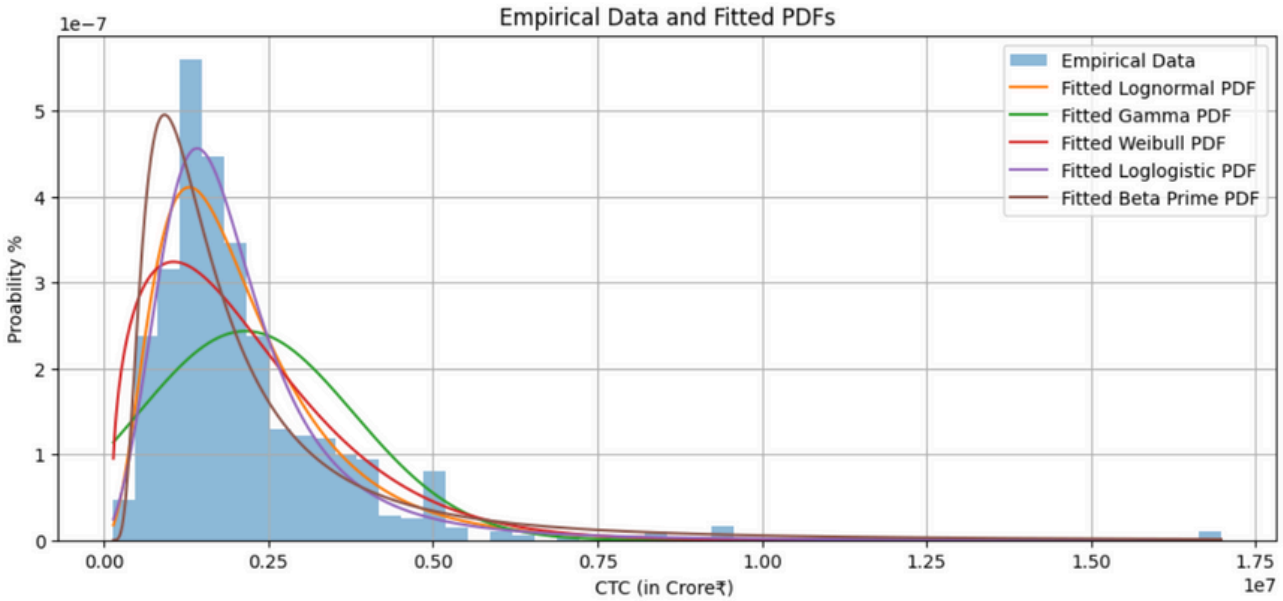
---

5. Fitting PDF against Theoretical Distributions

Similarly, we also fit the PDF of our empirical data against various known theoretical distributions to obtain the best fit for our random variable.

```
fitted_pdfs = {}
for name, dist in distributions.items():
    params = dist.fit(ctc_data)
    fitted_pdfs[name] = dist.pdf(x, *params)


plt.figure(figsize=(12, 6))
plt.hist(ctc_data, bins=50, density=True, alpha=0.5, label='Empirical Data')
for name, pdf in fitted_pdfs.items():
    plt.plot(x, pdf, label=f'Fitted {name} PDF')
plt.xlabel('CTC (in Crore₹) ')
plt.ylabel('Proability %')
plt.title('Empirical Data and Fitted PDFs')
plt.legend()
plt.grid(True)
plt.show()
```



STATISTICS REPORT

OBSERVATIONS

After plotting the empirical cumulative distribution function (CDF), we attempted to fit various standard probability distributions—namely lognormal, Weibull, Gamma, Beta, and others—to model the observed distribution of placement packages.

To determine the most suitable distribution, we employed the Least Sum of Squares (LSS) method. This involved minimizing the sum of squared differences between the values of the empirical CDF and the corresponding fitted CDF values generated by each theoretical distribution.
Mathematically, for each distribution, we computed:

$$\text{LSS} = \sum_{i=1}^{n} [F_{\text{empirical}}(x_i) - F_{\text{fitted}}(x_i)]^2$$

Where:

- $F_{empirical}(x_i)$ is the empirical PDF value at data point x_i
- $F_{fitted}(x_i)$ is the value form the theoretical PDF at the same point.

The following python code was used for its implementation to find the best fitting curve through the Least Sum of Squares method (LSS):

STATISTICS REPORT

OBSERVATIONS (...CONTINUED)

```python
# Computing LSS by empirical cdf and fitted curve
empirical_cdf_interp = np.interp(x, sorted_ctc, empirical_cdf)

ls_errors = {}
for name, fitted_cdf in fitted_cdfs.items():
    error = np.sum((fitted_cdf - empirical_cdf_interp) ** 2)
    ls_errors[name] = error
    print(f"{name} LS error: {error:.6f}")

best_distribution = min(ls_errors, key=ls_errors.get)
print(f"Best fitting distribution: {best_distribution} with LS error: {ls_errors[best_distribution]:.6f}")
```

The obtained output is attached below:

```
LogLogistic LS error: 0.176467
Gamma LS error: 2.808054
Weibull LS error: 0.822884
LogNormal LS error: 0.100936
Beta Prime LS error: 1.306376
Best fitting distribution: LogNormal with LS error: 0.100936
```

Thus LogNormal was chosen as our distribution to fit the data.

STATISTICS REPORT

RESULTS/INSIGHTS

The attached observations suggest that lognormal distribution is the best fit for our data.

Parameters of the obtained log-normal distribution are as follows:

1. μ: 14.548
2. σ: 0.539

The distribution follows the PDF for x>0:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln(x)-\mu}{\sigma}\right)^2}$$

Also, the CDF of the distribution is given by:

$$\frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{\ln x - \mu}{\sigma\sqrt{2}}\right)\right] = \Phi\left(\frac{\ln(x) - \mu}{\sigma}\right)$$

Few of the basic statistical measure were caculated, whose results are as follows:

	Actual (in rupees)	**Lognormal (in rupees)**
Mean	2173483.992	2407490.617
Median	1690000.000	2081229.127
Mode	1200000.000	1555358.401
Standard deviation	1687870.407	1399873.722

STATISTICS REPORT

RESULTS/INSIGHTS (...CONTINUED)

The final graphs of PDF and CDF of lognormal fit are attached below: