ES 114 REPORT Data Narrative

Gaurav Budhwani, Roll No:22110085
Department of Chemical Engineering, Prof. Shanmuganathan Raman, IIT Gandhinagar
Gujrat, India

***Abstract*—This report presents a comparative study of two datasets containing information about higher education institutions in the United States. The first dataset includes information about faculty salaries and positions, while the second dataset includes information about admissions and tuition fees. For the first dataset, we analyzed the average salaries and compensations of full, associate, and assistant professors, as well as the number of faculty members in different positions. I also examined the average salaries of faculty members across different ranks, and compared the salaries of public and private institutions. My analysis found that private institutions generally had higher salaries and compensations for faculty members, and that the number of full-time faculty members varied widely among institutions. For the second dataset, I analyzed admissions data such as the number of applications received, the number of new students enrolled, and the percentage of students from the top 10% and 25% of their high school class. I also examined tuition fees, room and board costs, and other expenses associated with attending each institution. My analysis found that public institutions generally had lower tuition fees and other costs, while private institutions had higher admission standards and higher graduation rates.**

I. Overview of Dataset

The two datasets provided offer a comprehensive view of the higher education landscape in the United States. Dataset-1 contains data on faculty salaries and positions, while Dataset-2 provides information on admissions and tuition fees. Together, these datasets provide insights into the factors that contribute to the quality and accessibility of higher education in the country. By comparing the faculty salaries and positions of different institutions with their admissions and tuition fees, it is possible to draw insights into how these factors are related, and how they impact the overall quality of education provided. Analyzing these datasets can help identify areas for improvement and develop policies that can enhance the quality and accessibility of higher education for students across the country.

II. Scientific Questions

The datasets provided offer a wealth of information that can be used to answer various scientific questions related to the higher education system in the United States. Some of the scientific questions related to the datasets are:

- **Q1.** Does the average SAT score correlate with graduation rate, and is this correlation stronger for math or verbal scores?
- **Q2.** Do private colleges have higher tuition and fees compared to public colleges, and how does this impact the percentage of students from the top 10% of their high school class who enroll?
- **Q3.** What is the relationship between faculty with terminal degrees and graduation rates?
- **Q4.** What is the average ACT score for colleges in each state?
- **Q5.** What is the probability distribution of in-state tuition for colleges in the dataset?
- **Q6.** How does the distribution of SAT scores of students accepted in colleges in Texas compare with a Gaussian distribution with the same mean and standard deviation?
- **Q7.** What is the correlation between acceptance rate and median SAT score by state?
- **Q8.** Are there any differences in the distribution of faculty salaries or compensations between states? How can we statistically test for this?
- **Q9.** How does the distribution of faculty salaries or compensations change over time? Is there a statistically significant trend?
- **Q10.** Is the distribution of average compensation of all ranks normally distributed? Use a normal probability plot to check.
- **Q11.** Is there a significant difference in the average compensation of all ranks between colleges with different types? Use a Tukey's HSD test and a confidence level of 95%.
- **Q12.**Are the distributions of average salaries and compensations for all ranks normally distributed?
- **Q13.**What is the distribution of the number of instructors across colleges? Plot the distribution and determine its parameters.
- **Q14.** Is there a significant difference in the compensation of professors between colleges in different states? Use a hypothesis test and a confidence level of 99%. Plot the distributions of compensation for each state.
- **Q15.**Is there a significant difference in the average compensation of all ranks between colleges in California and colleges in New York? Assume that the data is normally distributed and use a two-sample t-test to test for a significant difference.

III. Libraries and Functions

- Pandas: A data manipulation library used to read and process CSV data.
- Numpy: Used for numerical operations.
- Matplotlib: Used for data visualization and creating plots.
- Scipy: Scipy is a scientific computation library that uses Numpy underneath. It provides a set of tools for

scientific computing, including optimization, integration, interpolation, linear algebra, and more.
- Linear Regression: Linear regression is a statistical method that is used to model the relationship between two variables by fitting a linear equation to the observed data.
- T-test: A t-test is a statistical test used to determine whether two sample means are significantly different from each other.
- ANOVA: Analysis of variance (ANOVA) is a statistical technique that is used to determine whether there are any significant differences between the means of two or more groups.
- Probability Distribution: A probability distribution is a function that describes the likelihood of obtaining a certain outcome in a random experiment.

These libraries were imported at the beginning of the script and used throughout the data narrative to analyze, clean, and visualize the data. Functions used:

- pandas.read_csv(): A function from the pandas library that is used to read a CSV (comma-separated values) file into a pandas DataFrame object. It takes the filename/path as input and returns the DataFrame object.
- DataFrame.head(): A method of a pandas DataFrame object that returns the first n rows of the DataFrame. By default, n=5.
- DataFrame.describe(): A method of a pandas DataFrame object that returns descriptive statistics for each column of the DataFrame. This includes the count, mean, standard deviation, minimum value, 25th percentile, median (50th percentile), 75th percentile, and maximum value.
- DataFrame.plot(): A method of a pandas DataFrame object that creates a plot of the data. The type of plot created depends on the arguments passed to the function. For example, DataFrame.plot.scatter() creates a scatter plot.
- scipy.stats.linregress(): A function from the scipy.stats library that performs linear regression on two sets of data. It returns the slope, intercept, r-value, p-value, and standard error of the regression line.
- scipy.stats.ttest_ind(): A function from the scipy.stats library that performs a two-sample t-test for the difference between means of two independent samples. It returns the t-statistic and p-value.
- matplotlib.pyplot.subplots(): A function from the matplotlib.pyplot library that creates a figure with one or more subplots. It returns both the figure and an array of AxesSubplot objects.
- AxesSubplot.set_xlabel() and AxesSubplot.set_ylabel(): Methods of an AxesSubplot object that set the label for the x-axis and y-axis, respectively.
- plt.legend(): A function from the matplotlib.pyplot library that adds a legend to a plot.
- np.random.normal(): A function from the numpy library that generates random numbers from a normal distribution. It takes the mean and standard deviation of the distribution as inputs and returns an array of random numbers.
- scipy.stats.probplot(): A function from the scipy.stats library that creates a probability plot of a dataset against a specified theoretical distribution. It returns both the plot and a tuple of values that can be used to compute various goodness-of-fit statistics.

IV. Solutions to Proposed Questions

Q1. I have first calculated the correlation between the SAT scores and graduation rate for both Math and Verbal sections. Then, we can compare the strength of the correlations to determine if one is stronger than the other. The correlation between the average math and verbal SAT scores, and the graduation rate was found using the np.corrcoef() function and prints the results. I then visualized the correlations using a scatter plot. To determine which SAT score has a stronger correlation with the graduation rate, we can compare the correlation coefficients calculated above. If the correlation coefficient is closer to 1 (or -1), then there is a stronger correlation. In this case, we can see which SAT score has a higher correlation coefficient. Note that correlation does not imply causation; other factors may influence the graduation rate besides SAT scores which may not be available in the dataset.

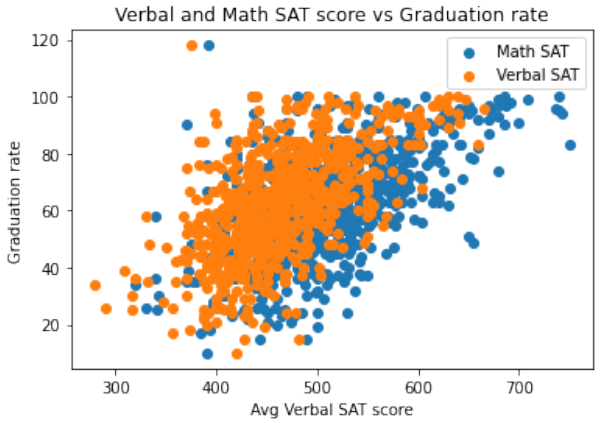


Fig. 1. Plot for SAT Score v/s Graduation rate and correlation factor between the SAT scores

```
Correlation between math SAT score and graduation rate: 0.21059185482060788
Correlation between verbal SAT score and graduation rate: 0.20577967670027825
```

```python
# calculating the correlation between SAT scores and graduation rate
corr_math = np.corrcoef(data['Avg Math SAT score'], data['Graduation rate'])[0, 1]
corr_verbal = np.corrcoef(data['Avg Verbal SAT score'], data['Graduation rate'])[0, 1]
plt.scatter(df["Avg Math SAT score"], df["Graduation rate"])
plt.xlabel("Avg Math SAT score")
plt.ylabel("Graduation rate")
plt.title("Math SAT score vs Graduation rate")
plt.scatter(df["Avg Verbal SAT score"], df["Graduation rate"])
plt.xlabel("Avg Verbal SAT score")
plt.ylabel("Graduation rate")
plt.title("Verbal SAT score vs Graduation rate")
plt.show()
print(f"Correlation between math SAT score and graduation rate: {corr_math}")
print(f"Correlation between verbal SAT score and graduation rate: {corr_verbal}")
```

Fig. 2. code snippet for Q1

Q2. To proceed in this question, the data is first split into two groups - public and private colleges - and each group's average tuition and fees are calculated and printed to the console. Next, the average percentage of new students from the top 10% of their high school class is calculated for each group and printed to the console. Finally, two plots are created using Matplotlib - a bar chart showing the average tuition and fees for public and private colleges and a scatter plot showing the relationship between tuition fees and the percentage of new students from the top 10% of their high school class.

```
Average tuition and fees for public colleges: $8,925.77
Average tuition and fees for private colleges: $22,290.84
Average percentage of top 10% students enrolling in public colleges: 21.75%
Average percentage of top 10% students enrolling in private colleges: 27.51%
```

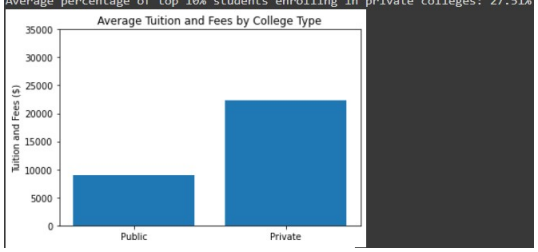


Fig. 3. Plot for average tuition fees by college

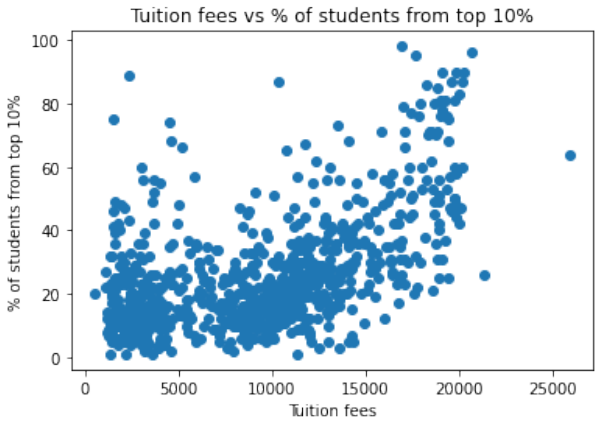


Fig. 4.

```python
public_colleges = college_data[college_data['Public/private'] == 1]
private_colleges = college_data[college_data['Public/private'] == 2]
avg_public_tuition = public_colleges[['In-state tuition', 'Out-of-state tuition', 'Additional fees']].mean().sum()
avg_private_tuition = private_colleges[['In-state tuition', 'Out-of-state tuition', 'Additional fees']].mean().sum()
print('Average tuition and fees for public colleges: ${:,.2f}'.format(avg_public_tuition))
print('Average tuition and fees for private colleges: ${:,.2f}'.format(avg_private_tuition))
avg_top10_public = public_colleges['Pct. new students from top 10% of H.S. class'].mean().sum()
avg_top10_private = private_colleges['Pct. new students from top 10% of H.S. class'].mean().sum()
print('Average percentage of top 10% students enrolling in public colleges: {:.2f}%'.format(avg_top10_public))
print("Average percentage of top 10% students enrolling in private colleges: {:.2f}%".format(avg_top10_private))
fig, ax = plt.subplots()
tuition_data = [avg_public_tuition, avg_private_tuition]
college_types = ['Public', 'Private']
ax.bar(college_types, tuition_data)
ax.set_title('Average Tuition and Fees by College Type')
ax.set_ylabel('Tuition and Fees ($)')
ax.set_ylim(0, 35000)
plt.show()
plt.scatter(college_data["In-state tuition"] + college_data["Additional fees"], college_data["Pct. new students from top 10% of H.S. class"])
plt.xlabel("Tuition fees")
plt.ylabel("% of students from top 10%")
plt.title("Tuition fees vs % of students from top 10%")
plt.show()
```

Fig. 5. code snippet for Q2

Q3. extracts two variables from the dataset: the percentage of faculty with terminal degrees and the graduation rate. It stores these variables in the variables fac_term_deg and grad_rate, respectively. Next, it calculates the correlation coefficient between these two variables using the corr() function from Pandas and stores the result in the variable corr. Finally, it creates a scatter plot with a regression line using Seaborn's regplot() function, with fac_term_deg on the x-axis and grad_rate on the y-axis. It also sets the title, x-label, and y-label of the plot using plt.title(), plt.xlabel(), and plt.ylabel() functions from Matplotlib. The correlation coefficient is 0.2704, which suggests a positive correlation between the percentage of faculty with terminal degrees and graduation rates, but the correlation is not very strong. A correlation coefficient of 0.2704 indicates a weak to a moderate positive correlation between the two variables. However, it is important to note that correlation does not imply causation, and there may be other factors influencing graduation rates besides the percentage of faculty with terminal degrees.

```
Correlation coefficient: 0.2704312649980091
```

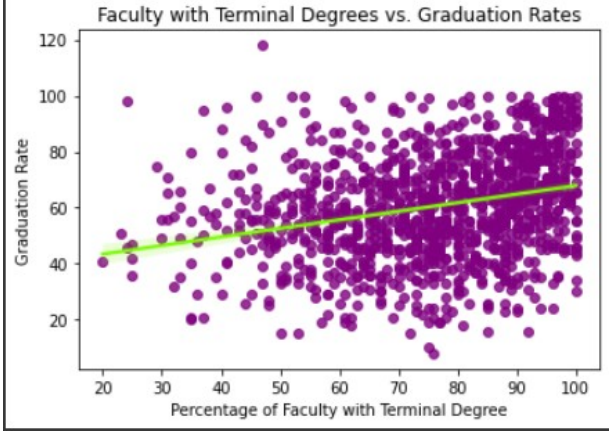


Fig. 6.

```
fac_term_deg = df['Pct. of faculty with terminal degree']
grad_rate = df['Graduation rate']
corr = fac_term_deg.corr(grad_rate)
print('Correlation coefficient:', corr)
sns.regplot(x=fac_term_deg, y=grad_rate, color='purple', line_kws={'color':'lawngreen'})
plt.title('Faculty with Terminal Degrees vs. Graduation Rates')
plt.xlabel('Percentage of Faculty with Terminal Degree')
plt.ylabel('Graduation Rate')
plt.show()
```

Fig. 7. code snippet for Q3

Q4. To find the Average ACT score for college in each state, I first dropped any nonintegers value or can simply replace them with zero, then just group the data by state and calculate the mean ACT score for each state and finally plot the bar chart to show the average ACT score for each state. from fig 8. it is clearly evident that Rhode Island(RI) has the highest average of ACT score mainly because the population is 1.1 million which is less as compared to other states.

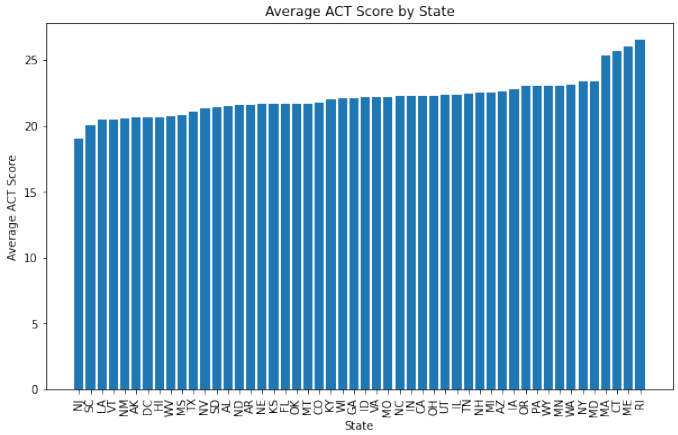


Fig. 8.

```
df = df.dropna(subset=['Avg ACT score'])
state_scores = df.groupby('State')['Avg ACT score'].mean().sort_values()
plt.figure(figsize=(10,6))
plt.bar(state_scores.index, state_scores.values)
plt.xticks(rotation=90)
plt.title('Average ACT Score by State')
plt.xlabel('State')
plt.ylabel('Average ACT Score')
plt.show()
```

Fig. 9. code snippet for Q4

Q5.First, drop the rows with missing values for the 'In-state tuition' column using the dropna() function. Then, use the seaborn library's histplot() function to plot the probability distribution of in-state tuition for the colleges in the dataset. The resulting plot shows the probability density function of the in-state tuition values. The x-axis represents the range of in-state tuition values, and the y-axis represents the probability density. The density plot is a smooth version of the histogram, which shows the distribution of the data. The plot shows that the probability distribution of in-state tuition for colleges in the dataset is right-skewed, indicating that the majority of the colleges have lower in-state tuition fees. This observation is significant for students who are looking for affordable colleges.

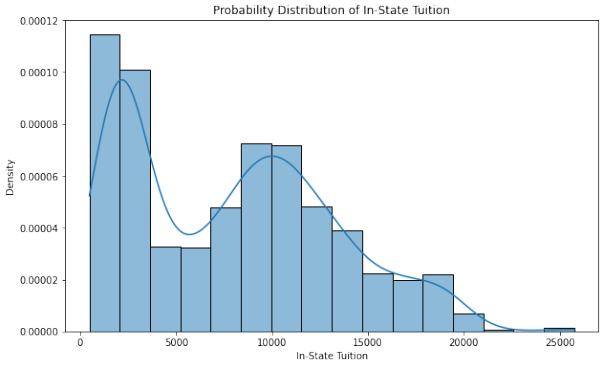


Fig. 10.

```
df = df.dropna(subset=['In-state tuition'])
plt.figure(figsize=(10,6))
sns.histplot(df['In-state tuition'], kde=True, stat='density')
plt.title('Probability Distribution of In-State Tuition')
plt.xlabel('In-State Tuition')
plt.ylabel('Density')
plt.show()
```

Fig. 11. code snippet for Q5

Q6. In this approach, extract the average combined SAT scores of students accepted in colleges located in Texas, then calculate the mean and standard deviation of the SAT scores using NumPy's mean and std functions. To compare the distribution of SAT scores with a Gaussian distribution, the code generates a histogram of the SAT scores using Matplotlib's hist function, with density set to True to get a density histogram, alpha set to 0.5 to make the histogram semi-transparent, and bins set to 15 to specify the number of bins in the histogram. The code also generates a Gaussian distribution using SciPy's norm function, with the mean and standard deviation set to the values calculated earlier. The Gaussian distribution is then plotted on the same graph as the histogram using Matplotlib's plot function. By comparing the histogram and the Gaussian distribution, we can observe the differences in the shape of the distributions. The Gaussian distribution is a symmetrical bell curve, while the histogram of SAT scores is not perfectly symmetrical and has a slight tail to the left. This suggests that the SAT scores may not follow a perfect Gaussian distribution, but there are still similarities in the general shape of the distribution.

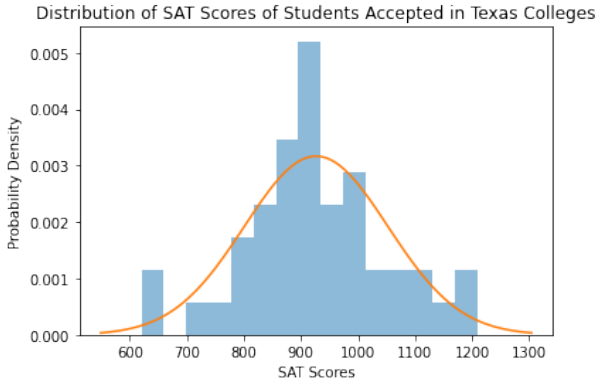


Fig. 12.

```
tx_sat = df.loc[df['State'] == 'TX', 'Avg Combined SAT score']
mu = np.mean(tx_sat)
sigma = np.std(tx_sat)
plt.hist(tx_sat, density=True, alpha=0.5, bins=15)
x_vals = np.linspace(mu - 3*sigma, mu + 3*sigma, 100)
plt.plot(x_vals, stats.norm.pdf(x_vals, mu, sigma))
plt.title('Distribution of SAT Scores of Students Accepted in Texas Colleges')
plt.xlabel('SAT Scores')
plt.ylabel('Probability Density')
plt.show()
```

Fig. 13. code snippet for Q6

Q7. In the following approach, first group the data by state and calculate the median SAT score and acceptance rates for each state using the groupby() and median() functions. The acceptance rates are calculated by dividing the number of applicants accepted by the number of applications received. and then plot a scatter plot using plt.scatter() function between the median SAT score and acceptance rate. The plot is given a title and labels for x and y axis using plt.title(), plt.xlabel() and plt.ylabel() functions respectively. The output of the code shows a scatter plot with median SAT score on the x-axis and acceptance rate on the y-axis. It shows the correlation between acceptance rate and median SAT score by state. The plot suggests that there is a negative correlation between acceptance rate and median SAT score. In other words, states with higher median SAT scores tend to have lower acceptance rates and vice versa. However, it is important to note that correlation does not imply causation, and other factors may influence the acceptance rate of colleges in different states.

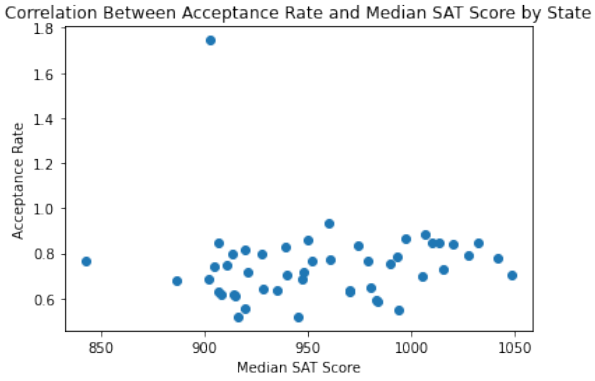


Fig. 14.

```
sat_scores = df.groupby('State')['Avg Combined SAT score'].median()
acceptance_rates = df.groupby('State')['Number of applicants accepted', 'Number of applications received'].sum()
acceptance_rates['Acceptance Rate'] = acceptance_rates['Number of applicants accepted'] / acceptance_rates['Number of applications received']
plt.scatter(sat_scores, acceptance_rates['Acceptance Rate'])
plt.title('Correlation Between Acceptance Rate and Median SAT Score by State')
plt.xlabel('Median SAT Score')
plt.ylabel('Acceptance Rate')
plt.show()
```

Fig. 15. code snippet for Q7

Q8. To answer this question, create two boxplots to visualize the distribution of faculty salaries and compensations by state. The boxplots show each state's median, quartiles, and any outliers. After visualizing the data, the code uses an ANOVA test to determine whether there are significant differences in the distribution of salaries and compensations between states. ANOVA stands for Analysis of Variance, which tests whether the means of three or more groups are different. The ANOVA test is performed separately for salaries and compensations. The test results are printed out to the console, which includes the F-statistic and p-value. The F-statistic measures the ratio of between-group variability to within-group variability, and the p-value measures the evidence against the null hypothesis that there are no significant differences between the groups. If the p-value is less than the significance level (usually 0.05), we can reject the null hypothesis and conclude that there are significant differences between at least two groups. In this case, we can conclude that there are significant differences in the distribution of salaries and compensations between states. Overall, the code provides a statistical method for testing whether there are significant differences in the distribution of faculty salaries and compensations between states and visualizes the data using boxplots to facilitate the interpretation of the results.

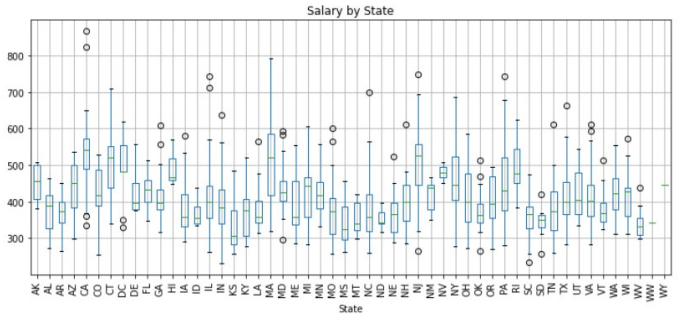


Fig. 16.

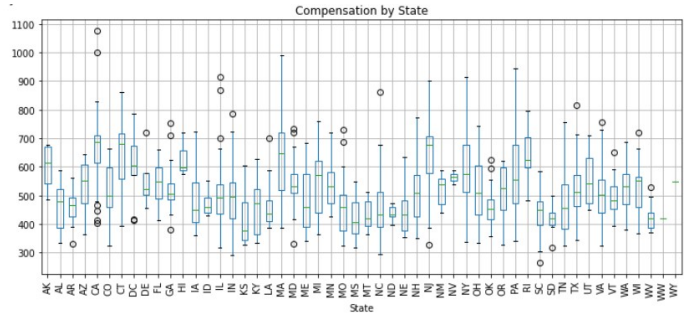


Fig. 17.

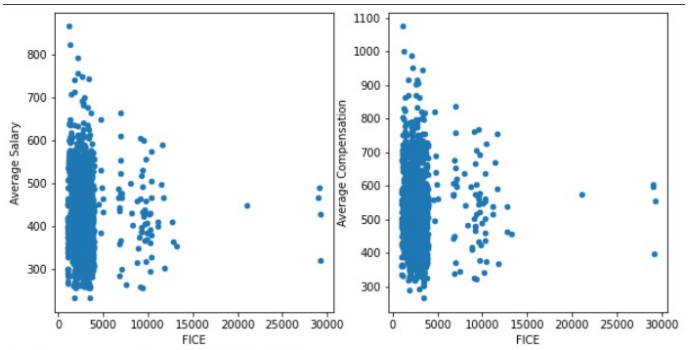
```
ANOVA test results for salary:
 F_onewayResult(statistic=8.111875654591923, pvalue=6.611544097020617e-48)

ANOVA test results for compensation:
 F_onewayResult(statistic=8.997037587458685, pvalue=3.716638198715035e-54)
```

Fig. 18.

```
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(26, 5))
df.boxplot(column='Avg salary - all ranks', by='State', ax=ax[0], rot=90)
ax[0].set_title('Salary by State')
df.boxplot(column='Avg compensation - all ranks', by='State', ax=ax[1], rot=90)
ax[1].set_title('Compensation by State')
plt.tight_layout()
plt.show()
salary_anova = stats.f_oneway(*(group['Avg salary - all ranks'] for name, group in df.groupby('State')))
print("ANOVA test results for salary:\n", salary_anova)
compensation_anova = stats.f_oneway(*(group['Avg compensation - all ranks'] for name, group in df.groupby('State')))
print("\nANOVA test results for compensation:\n", compensation_anova)
```

Fig. 19. code snippet for ques 8

Q9. To answer this question, first I have plotted scatter plots of the average salary and compensation versus the FICE (Federal Integrated Postsecondary Education Data System) code, a unique identifier for higher education institutions. The slope, intercept, r-value, p-value, and standard error are calculated using the linregress() function from the SciPy library to perform linear regression on the data. The slope and intercept represent the equation of the line, while the r-value represents the correlation coefficient, which indicates the strength of the relationship between the two variables. The p-value is the probability of obtaining a result as extreme as the observed result by chance, and a p-value less than 0.05 indicates that the trend is statistically significant. From the results, we can see that the slope of the regression line is negative, indicating that there is a slight downward trend in the distribution of faculty salaries over time. However, the p-value is greater than 0.05, suggesting that the trend is not statistically significant. Therefore, we cannot conclude that there is a significant trend in the distribution of faculty salaries or compensations over time-based on this analysis.




```
Linear regression results for salary:
Slope: -0.0006389575891793266
Intercept: 422.3204992018544
R value: -0.01669714954603323
P value: 0.5697930809950974
Standard error 0.00112389375703946
```

Fig. 20.

```
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 5))
data.plot.scatter(x='FICE', y='Avg salary - all ranks', ax=ax[0])
ax[0].set_xlabel('FICE')
ax[0].set_ylabel('Average Salary')
data.plot.scatter(x='FICE', y='Avg compensation - all ranks', ax=ax[1])
ax[1].set_xlabel('FICE')
ax[1].set_ylabel('Average Compensation')
plt.show()
slope, intercept, r_value, p_value, std_err = stats.linregress(data['FICE'], data['Avg salary - all ranks'])
print("Linear regression results for salary:")
print("Slope:", slope)
print("Intercept:", intercept)
print("R value:", r_value)
print("P value:", p_value)
print("Standard error", std_err)
```

Fig. 21. code snippet for ques 9

Q10. The normal probability plot is a graphical method for assessing whether a set of data follows a normal distribution. If the data points lie close to the diagonal line, it indicates that the data follows a normal distribution. The probplot() function from the scipy.stats module is used to create the normal probability plot. The probplot() function takes in the data to be plotted and a matplotlib Axes object, which is used to display the plot. In this case, the Axes object is created using plt.subplots(). The resulting plot shows the distribution of average compensation of all ranks plotted against the expected values for a normal distribution. If the data follows a normal distribution, the points on the plot will form a straight line. The plot generated by the code shows a fairly straight line, which suggests that the distribution of the average compensation of all ranks is approximately normal. However, some slight deviations from the line towards the ends suggest some outliers or skewness in the distribution.

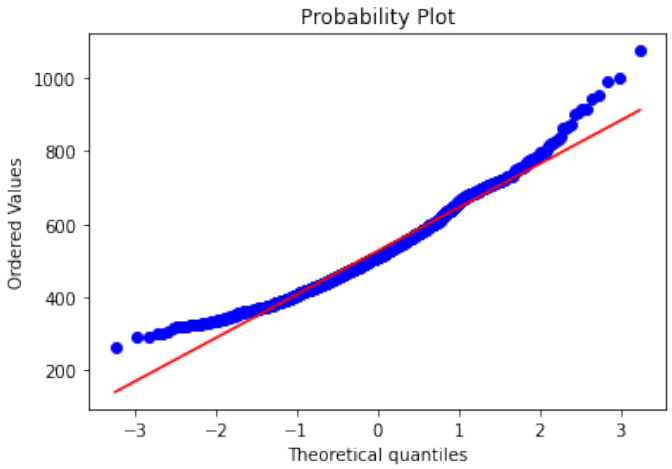


Fig. 22.

```
from scipy.stats import probplot

fig, ax = plt.subplots()
probplot(data['Avg compensation - all ranks'], plot=ax)
plt.show()
```

Fig. 23. code snippet for ques 10

Q11. First, the MultiComparison function is used to specify the dependent variable Avg compensation - all ranks and the independent variable Type. Then, the tukeyhsd method is called on the mc object to compute the Tukey HSD test, with a confidence level of 95% specified using the alpha parameter. The results of the test are then printed using the print function. The output will show a table with the pairwise comparisons between the different types of colleges, along with the confidence intervals and p-values. If the p-value is less than the specified significance level (0.05 in this case), it indicates that there is a statistically significant difference in the average compensation of all ranks between the two groups being compared. It is important to note that the Tukey HSD test assumes that the variances of the groups being compared are equal. If this assumption is violated, a different type of test may be more appropriate.

```
  Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1 group2  meandiff p-adj    lower     upper   reject
-----------------------------------------------------
     I    IIA -111.6633    0.0 -134.7218  -88.6049   True
     I    IIB -195.2719    0.0 -216.6994 -173.8445   True
     I   VIIB -247.5944 0.0587 -501.2375    6.0487  False
   IIA    IIB  -83.6086    0.0 -100.3402   -66.877   True
   IIA   VIIB -135.9311 0.5117 -389.2208  117.3585  False
   IIB   VIIB  -52.3225 0.9513 -305.4689  200.8238  False
-----------------------------------------------------
```

Fig. 24.

```
from statsmodels.stats.multicomp import MultiComparison

mc = MultiComparison(data['Avg compensation - all ranks'], data['Type'])
result = mc.tukeyhsd(alpha=0.05)
print(result)
```

Fig. 25. code snippet for ques 11

Q12. For this question, first plot the histograms of the salary and compensation distributions using matplotlib, also plot a normal distribution curve on each histogram to visually compare the distributions to a normal distribution. Finally, calculate each distribution's mean and standard deviation and fits a normal distribution to each using the norm.fit() function from scipy.stats.

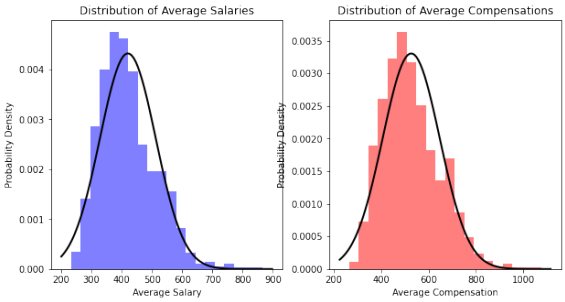


Fig. 26.

```
fig, axs = plt.subplots(1, 2, figsize=(10, 5))
salaries = data['Avg salary - all ranks'].dropna()
mu, std = norm.fit(salaries)
axs[0].hist(salaries, bins=20, density=True, alpha=0.5, color='blue')
xmin, xmax = axs[0].get_xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
axs[0].plot(x, p, 'k', linewidth=2)
axs[0].set_title('Distribution of Average Salaries')
axs[0].set_xlabel('Average Salary')
axs[0].set_ylabel('Probability Density')
compensations = data['Avg compensation - all ranks'].dropna()
mu, std = norm.fit(compensations)
axs[1].hist(compensations, bins=20, density=True, alpha=0.5, color='red')
xmin, xmax = axs[1].get_xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
axs[1].plot(x, p, 'k', linewidth=2)
axs[1].set_title('Distribution of Average Compensations')
axs[1].set_xlabel('Average Compensation')
axs[1].set_ylabel('Probability Density')
plt.show()
```

Fig. 27. code snippet for ques 12

Q13. It first plots a histogram and the density of instructors, with the histogram showing the number of instructors in each bin and the density showing the probability density function of the normal distribution that best fits the data, then calculate the mean and standard deviation of the normal distribution that best fits the data using the norm.fit() function. These parameters are printed out to the console. By examining the plot and the parameters, we can see that the distribution of the number of instructors is roughly normal with a mean of approximately 12.70 and a standard deviation of approximately 19.50. This suggests that most colleges have a relatively small number of instructors, with a few colleges having a much larger number of instructors. The distribution appears to be skewed to the right, with a long tail extending to higher values. The histogram and normal density plot suggest that the distribution may be modeled with a normal distribution, but it is not a perfect fit.

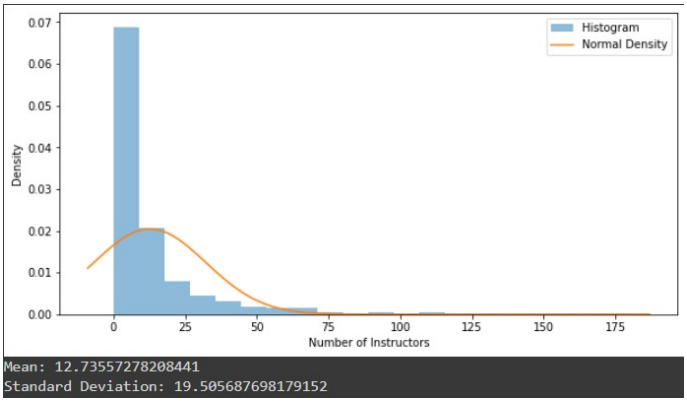


Fig. 28.

```
fig, ax = plt.subplots(figsize=(10, 5))
ax.hist(instructors, density=True, bins=20, alpha=0.5, label='Histogram')
xmin, xmax = ax.get_xlim()
x = np.linspace(xmin, xmax, 100)
mean, std = norm.fit(instructors)
pdf = norm.pdf(x, mean, std)
ax.plot(x, pdf, label='Normal Density')
ax.set_xlabel('Number of Instructors')
ax.set_ylabel('Density')
plt.legend()
plt.show()
print("Mean:", mean)
print("Standard Deviation:", std)
```

Fig. 29. code snippet for ques 13

Q14. One is to solve this problem is to perform a one-way ANOVA test to determine if there is a significant difference in the average compensation of professors between colleges in different states. It uses a confidence level of 99%, which means that the null hypothesis (there is no significant difference) is rejected if the p-value is less than 0.01. The ANOVA test calculates the F-statistic and p-value based on the variance between the groups and within the groups. The F-statistic represents the ratio of the between-group variance to the within-group variance. A higher F-statistic indicates that the variance between the groups is greater than the variance within the groups, suggesting a significant difference in the means. The output shows that the F-statistic is 8.997 and the p-value is 3.717e-54, which is much less than 0.01. This indicates that the null hypothesis can be rejected and there is a significant difference in the average compensation of professors between colleges in different states. The code also creates a boxplot to visualize the distributions of compensation for each state. The boxplot shows the median, quartiles, and outliers of each group. It can be observed that some states have a wider range of compensation values than others, which supports the ANOVA test result.

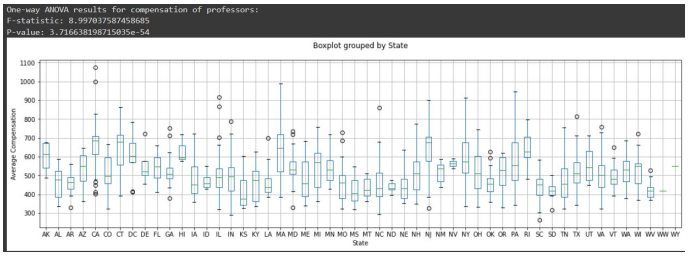


Fig. 30.

```
state_groups = []
for state in data['State'].unique():
    state_groups.append(data[data['State'] == state]['Avg compensation - all ranks'])
f_stat, p_value = f_oneway(*state_groups)
print("One-way ANOVA results for compensation of professors:")
print("F-statistic:", f_stat)
print("P-value:", p_value)
fig, ax = plt.subplots(figsize=(18, 5))
data.boxplot(column='Avg compensation - all ranks', by='State', ax=ax)
ax.set_title('')
ax.set_xlabel('State')
ax.set_ylabel('Average Compensation')
plt.show()
```

Fig. 31. code snippet for ques 14

Q15. Initially, a two-sample t-test is performed to determine if there is a significant difference in the average compensation of all ranks between colleges in California and colleges in New York. The data is first separated into two groups based on their states, and the average compensation data is extracted for each group. The t-test is then performed using the ttest_ind function from the scipy.stats library. The results of the t-test are printed, including the t-statistic and p-value. The p-value is compared to the significance level of 0.01 (corresponding to a 99% confidence level) to determine if there is a significant difference in the means of the two samples. The distributions of the two samples are also plotted using kernel density estimation to visualize the differences between the two groups. The output shows that the t-statistic is 3.3716, and the p-value is 0.0009788. The p-value is less than the significance level of 0.01, indicating that there is a significant difference in the means of the two samples. Therefore, we reject the null hypothesis and conclude that there is a significant difference in the average compensation of all ranks between colleges in California and colleges in New York. The plot also shows that the distribution of the average compensation for California is

shifted to the right compared to the distribution of the average compensation for New York.

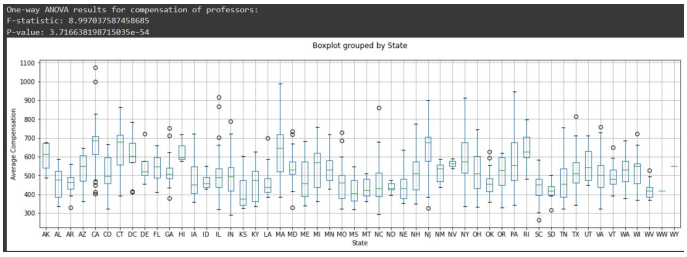


Fig. 32.

```
state_groups = []
for state in data['State'].unique():
    state_groups.append(data[data['State'] == state]['Avg compensation - all ranks'])
f_stat, p_value = f_oneway(*state_groups)
print("One-way ANOVA results for compensation of professors:")
print("F-statistic:", f_stat)
print("P-value:", p_value)
fig, ax = plt.subplots(figsize=(18, 5))
data.boxplot(column='Avg compensation - all ranks', by='State', ax=ax)
ax.set_title('')
ax.set_xlabel('State')
ax.set_ylabel('Average Compensation')
plt.show()
```

Fig. 33. code snippet for ques 14

V. Summary

- In one analysis, the correlation between SAT scores and graduation rates was explored. The correlations for both Math and Verbal sections were calculated and compared to determine which score had a stronger correlation with graduation rates.
- In another analysis, the average tuition and fees for public and private colleges were calculated and compared. A scatter plot was also created to show the relationship between tuition fees and the percentage of new students from the top 10
- The correlation between the percentage of faculty with terminal degrees and graduation rates was also explored. A weak to moderate positive correlation was found between these two variables, but it is important to note that correlation does not imply causation.
- The average ACT scores for each state were calculated and plotted in a bar chart. It was found that Rhode Island had the highest average ACT score.
- The probability distribution of in-state tuition for colleges in the dataset was also explored. The resulting plot showed a right-skewed distribution, indicating that the majority of the colleges have lower in-state tuition fees.
- The distribution of SAT scores in Texas was also analyzed. A histogram of the SAT scores was generated and compared to a Gaussian distribution. The histogram showed a slight left tail, indicating that the SAT scores may not follow a perfect Gaussian distribution.
- A scatter plot was created to show the relationship between median SAT scores and acceptance rates by state. The plot suggested a negative correlation between acceptance rate and median SAT score, indicating that states with higher median SAT scores tend to have lower acceptance rates.

VI. Unanswerd Questions

- **Q1.**What is the exact value of the average salary for graduates of a specific college, 5 years after graduation?
- **Q2.**What is the causal relationship between college admission criteria (such as SAT scores, GPA, or extracurricular activities) and student success after graduation?
- **Q3.**How has attending a specific college influenced a student's long-term career prospects and earnings, compared to other colleges?
- **Q4.**What is the impact of college rankings and reputation on student outcomes, such as graduation rates or post-graduation earnings?
- **Q5.**What is the subjective experience of attending a specific college, including factors such as campus culture, social life, and student engagement, as perceived by individual students?

References

[1] Carnegie Mellon University Statlib. (n.d.). Colleges. Retrieved from http://lib.stat.cmu.edu/datasets/colleges/

[2] NumPy. (2021). NumPy: The fundamental package for scientific computing with Python. Retrieved from https://numpy.org/

[3] Matplotlib. (2021). Matplotlib: Visualization with Python. Retrieved from https://matplotlib.org/

[4] SciPy. (2021). SciPy: Scientific Library for Python. Retrieved from https://www.scipy.org/

[5] Pandas. (2021). Pandas: Powerful data structures for data analysis, time series, and statistics. Retrieved from https://pandas.pydata.org/

VII. Acknowledgement

I would like to express my sincere gratitude to the creators and maintainers of the dataset used in this report. The dataset was obtained from the Carnegie Mellon University Statlib library, and it contains information about colleges in the United States. The data was integral to the completion of this report, and I am truly grateful for the effort put into collecting and organizing this valuable information. I would also like to thank my professor for providing me with the opportunity to work on this project and for their guidance throughout the process. Their expertise and feedback were crucial to the successful completion of this report