The Effects of Ethnicity on Students' Behaviour and Outcomes

Pawan Singh Kapkoti
Student Number: 220046826

MSc Data Analytics

Aston University

15 Jan 2024

Supervisor: Dr Gareth Woods

Abstract

This dissertation embarks upon a thorough exploration of academic performance disparities among students, with particular attention paid to ethnicity alongside socio-economic factors, gender and other educational variables. Carried out within Aston University's Mathematics Department using robust data sets and various statistical techniques for analysis, this research uncovers possible reasons behind disparate results.

This research seeks to uncover the various influences that contribute to student achievements, specifically how ethnicity, socio-economic status, gender, highest qualification on entry attendance and diagnostic scores affect academic results. This analysis seeks to gain a better insight into students from diverse backgrounds as they navigate academic life.

Initial findings point to significant variations in academic performance that may be linked with ethnicity, socio-economic status and gender. This research studies these correlations critically while noting that ethnicity does not in itself determine academic success; hence the necessity of considering all relevant influencing factors to understand why there may be academic disparities.

Acknowledgements

My deepest thanks go out to Dr. Gareth Woods, whose invaluable assistance in providing data for this research was essential. As a colleague professor in Aston University's Mathematics Department, his generosity of expertise and resources played a pivotal role in its progress and success.

Thank you to all the staff and colleagues within my department for creating an inviting academic environment and intellectual stimulation, including collaboration and shared insights among my peers which have been both enriching and eye-opening experiences for me.

Thank you to my family and friends for being an endless source of encouragement, patience, and understanding throughout this journey. Your unwavering belief has provided strength and motivation.

Contents

Introduction

Understanding the complex dynamics that shape student experiences and outcomes in education is of critical importance, and this report explores one such vital aspect: ethnicity's impact on student behavior and outcomes. This research attempts to uncover how ethnic backgrounds intersect with academic performance- a topic which resonates deeply within today's diverse educational system.

0.1 The Research Motivation

At the core of this research lies an advocacy for educational equity. Recognizing the dynamic classroom environments we inhabit today, my study seeks to understand how students from diverse ethnic backgrounds navigate their academic journeys - with an aim of uncovering any patterns or disparities that might exist and providing insights that might facilitate more inclusive and effective educational practices.

0.2 Navigating Data: Methodological Journey

Research undertaken as part of this investigation involves meticulous data analysis, which involves meticulously inspecting every number and pattern with precision. Beginning with data cleaning and preparation, which ensures an accurate and relevant dataset, exploratory data analysis (EDA) steps in to conduct further investigations of diagnostic scores and final grades across ethnic groups; EDA provides insights into how ethnicity might influence academic outcomes.

Analysis doesn't end here; to fully grasp what the data means, a combination of descriptive and inferential statistical methods must also be employed to understand its narrative. This approach goes beyond simply crunching numbers; it involves understanding stories behind numbers; it seeks meaning in patterns that emerge.

0.3 Revealing Patterns and Key Findings: Significant Findings

This study's findings provide more than academic insights; they reflect students' lived experiences. Data indicates that academic performance does indeed vary across ethnic groups, suggesting ethnicity plays an integral part in shaping educational outcomes. These discoveries could have significant ramifications on educational strategies at Aston University and beyond - emphasizing the need for tailored support wherever needed.

0.4 Beyond the Classroom: Broader Implications

This research goes beyond informing educational practices; it also offers insights into wider societal issues. How ethnicity impacts education is reflective of larger patterns of inequal-

ity and social dynamics - by understanding these within education context, this study contributes to greater discourse surrounding justice and equality for all.

1 Literature Review

1.1 Race and Ethnicity as Determinants of Educational Success in Higher Education

The role of race and ethnicity in higher education is a topic of significant importance and complexity. As highlighted in the 2020 Supplement of 'Race and Ethnicity in Higher Education' by the American Council on Education (ACE, 2020), these factors are pivotal in shaping educational success. This comprehensive report sheds light on the persistent equity gaps in higher education, particularly in the United States, and provides a data-informed perspective on the ways race and ethnicity serve as salient predictors of access to and success in higher education.

The ACE report reveals that the racial and ethnic backgrounds of students significantly influence not only their access to educational opportunities but also their experiences and outcomes within the academic setting. For example, students from historically marginalized communities often face systemic barriers that limit their access to higher education. These barriers can include, but are not limited to, economic constraints, limited access to quality pre-college education, and underrepresentation in advanced academic programs.

Furthermore, the report discusses how institutional biases and systemic inequities can affect educational pathways. These biases can manifest in various ways, such as through differential treatment by educators, disparities in resource allocation, and the perpetuation of cultural and social stereotypes. Such biases not only hinder the academic progress of students from minority ethnic and racial backgrounds but also impact their overall educational experience.

To address these challenges, the ACE report emphasizes the importance of developing and implementing policies and practices that foster inclusivity and equity in higher education. This involves creating an environment where all students, regardless of their racial or ethnic background, have equal opportunities to succeed. Strategies might include diversifying faculty and staff, implementing culturally responsive teaching practices, and providing targeted support services for underrepresented students.

Understanding the nuanced impact of race and ethnicity on educational outcomes is crucial for educators, policymakers, and community leaders. It is not enough to simply acknowledge the existence of disparities; proactive measures must be taken to dismantle the barriers that perpetuate these gaps. This could involve policy reforms, increased funding for programs that support underrepresented students, and a commitment to fostering a more inclusive campus culture.

1.2 Educational Environment and Its Impact at University Level

The influence of the educational environment, extending beyond family factors, is notably significant in the context of higher education institutions like universities. Extensive re-

search, including studies by Palardy (Palardy 2008 and Perry and McConney, 2010), demonstrates that both the academic and social environment of a university profoundly impact student outcomes. Key factors such as the availability of resources, the quality of teaching staff, and the institution's overall academic culture are critical in shaping students' educational experiences and achievements.

The availability and quality of university resources, including libraries, laboratories, and technological facilities, play a crucial role in facilitating effective learning and research. Additionally, the caliber and experience of faculty members significantly influence student engagement and learning outcomes. Furthermore, the academic culture of the university, encompassing shared values, expectations, and norms, motivates students and fosters a sense of belonging, ultimately encouraging academic excellence.

The Aston Maths dataset provides a specific example of how the university environment can directly impact students' academic performance, revealing insights into the correlation between various aspects of the university environment and student success. Moreover, international assessments like the Programme for International Student Assessment (PISA), as discussed in (OECD 2016), highlight global disparities in educational resources and support systems, elucidating their effects on academic outcomes across different regions and countries.

In conclusion, the educational environment at the university level, including academic resources, faculty expertise, and peer interactions, plays a significant role in influencing student success. This understanding is vital for higher education institutions striving to create a conducive learning environment, as well as for policymakers and educational leaders seeking to enhance the quality and effectiveness of higher education.

1.3 Socio-demographic Influences on Educational Outcomes

Beyond the foundational correlation between socioeconomic status and academic achievement, the nuances of family dynamics and structure play a crucial role. Research has highlighted how variations in family composition, such as single-parent households (?, ?) and larger family sizes (Downey 1995), significantly influence educational attainment. These studies underscore the multifaceted nature of educational outcomes, where factors like parental involvement, economic resources, and familial stability converge to shape the academic trajectory of students. Further expanding this perspective, Bogges (Bogges 1998) illustrates the intertwining relationship between family economic status and educational attainment, suggesting a direct correlation between the two. This section of the review delves into these diverse familial factors, elucidating their collective impact on students' educational journey and highlighting the importance of considering a broad spectrum of socio-demographic elements in educational research and policy formulation.

The intricate interplay of socioeconomic status and educational background in determining university success has garnered significant attention in recent educational research. Studies such as (Rodriguez-Hernandez, Cascallar, and Kyndt's, 2020) systematic review emphasize the enduring impact of socioeconomic factors on higher education performance. This is complemented by research exploring how family background influences student achievement, shedding light on the long-standing debate about the role of inherited advantages in educational success

(Education Next, 2016). Furthermore, the effectiveness of peer-assisted learning (PAL) in this context becomes particularly relevant, with investigations revealing its potential to bridge gaps in understanding and performance among students from diverse backgrounds (Williams and Reddy, 2016). This section of the review critically examines these dimensions, delving into how socioeconomic and educational backgrounds shape the academic journey of university students and exploring the role of supportive educational strategies in mitigating the challenges posed by these backgrounds.

1.4 Integration of Predictive Analytics in Educational Research

The advancement of predictive analytics in educational research has revolutionized our understanding of factors influencing academic success. The utilization of data-driven approaches, as evidenced in studies on pre-admission testing, underscores the predictive power of initial assessments in forecasting university students' academic trajectories (WadeeCliff 2016). These assessments, often conducted before students begin their university education, provide critical insights into potential academic challenges, enabling early intervention and tailored support strategies.

Moreover, the increasing application of peer-assisted learning (PAL) programs has been scrutinized for their effectiveness in enhancing academic outcomes. Comprehensive reviews of PAL's impact, such as those by Williams and Reddy (2016), demonstrate significant benefits in fostering collaborative learning environments. These programs, by promoting mutual understanding and support among peers, have shown to improve comprehension of complex concepts, learning skills, and overall academic performance.

Furthermore, the integration of socioeconomic factors into educational research, as explored by (Rodriguez-Hernandez, Cascallar, Kyndt 2020), has brought to light the varying challenges faced by students from different socioeconomic backgrounds. Recognizing these disparities is crucial for the development of equitable educational policies and practices that are sensitive to the unique needs of each student.

This fusion of traditional educational research with modern analytical techniques offers valuable insights into the complex interplay of socioeconomic status, preparatory academic measures, and innovative learning strategies in shaping student achievement. The use of predictive analytics in education not only enhances our understanding of academic success but also enables the development of more effective, data-driven strategies to ensure equitable educational opportunities for all students.

1.5 Enhancing Academic Performance through Peer-Assisted Learning

Peer-assisted learning (PAL) has emerged as a significant area of interest in educational research due to its potential to enhance academic performance across various learning environments. The scoping review by Williams and Reddy (2016) underscores the versatility and effectiveness of PAL programs, noting their positive impact in diverse educational settings. This approach is particularly beneficial in contexts where students come from varied educational backgrounds, helping to equalize opportunities and foster a supportive learning atmosphere.

PAL plays a vital role in creating an equitable academic environment, particularly for students facing challenges due to their backgrounds. By promoting collaborative learning, PAL helps bridge the gaps in understanding and skills among students. This collaborative nature not only enhances academic performance but also builds essential soft skills such as communication, teamwork, and empathy. These skills are crucial for success in both academic and professional realms.

Furthermore, the interplay of PAL with socioeconomic factors is a critical aspect of its efficacy. As Rodriguez-Hernandez, Cascallar, and Kyndt (2020) highlight in their systematic review, PAL can significantly mitigate academic disadvantages associated with lower socioeconomic status. This is achieved by providing a platform where students can learn from peers who

1.6 Ethnic Studies and Their Impact on Educational Success

Ethnic studies, encompassing the interdisciplinary study of the social, political, economic, and historical perspectives of diverse racial and ethnic groups, play a crucial role in fostering cross-cultural understanding. This understanding benefits not only students of color but also white students, aiding them in appreciating their own cultural identity as well as the differences around them.

Research indicates that students who engage in ethnic studies curricula show increased academic engagement, develop stronger self-efficacy, and are more likely to achieve higher academic performance and graduation rates. These studies affirm the positive effects of ethnic studies on personal empowerment and academic success.

Furthermore, well-designed and effectively taught ethnic studies programs, particularly those addressing racism directly, have been shown to elevate critical thinking levels. They also positively impact 'democracy outcomes,' enhancing students' understanding and appreciation of democratic values, especially when these curricula include interactions across different racial and ethnic groups. This aspect is particularly influential for white students.

Across various states and localities, there is a growing movement to integrate research-based ethnic studies into K-12 schools and higher education curricula. This integration is being spearheaded by educators and community partners who recognize the value of these programs and are actively campaigning for schools to offer ethnic studies courses.

The significance of ethnic studies extends beyond academic benefits; it represents a commitment to respecting and honoring the diversity, contributions, history, and cultural identities of all students, especially those under-represented in educational texts, curricula, and programs. This commitment is seen as a key step towards fostering a deeper understanding of our nation's diverse groups and cultures, and its ultimate goal is to enhance student success and personal growth (Ethnic Studies, 2024).

2 Background

2.1 Shapiro-Wilk Test in Our Research

In our research, we employed the Shapiro-Wilk test to assess the normality of diagnostic scores across various groups. This test is particularly effective for small sample sizes, making it a preferred choice in our analysis.

2.1.1 Mathematical Formulation of the Shapiro-Wilk Test

The Shapiro-Wilk test statistic, denoted as W, is formulated as follows:

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{1}$$

where:

- n is the number of observations in the sample.
- $x_{(i)}$ are the ordered sample values.
- $\bar{x}$ is the sample mean.
- a_i are coefficients calculated from the means, variances, and covariances of the order statistics of a normal distribution.

Explanation of the Formula:

- The numerator $(\sum_{i=1}^{n} a_i x_{(i)})^2$ is the squared sum of the product of each ordered observation $x_{(i)}$ and its corresponding coefficient a_i. These coefficients are specific to each sample size and are derived from the expected values of the order statistics for a normally distributed sample.
- The denominator $\sum_{i=1}^{n} (x_i - \bar{x})^2$ is the sum of squared deviations of each observation x_i from the sample mean $\bar{x}$, measuring the total sample variance.
- The ratio of these two sums yields the Shapiro-Wilk statistic W, which is used to assess the normality of the data distribution.

2.1.2 Application in Our Research

In our research, we applied the Shapiro-Wilk test to evaluate whether the diagnostic scores conformed to a normal distribution across different categories such as ethnicity, socio-economic status, gender, and qualification levels.

1. By confirming or refuting the normality of these scores, we were able to select the most appropriate statistical tests for our subsequent analyses.
2. For datasets that exhibited normality, we proceeded with parametric tests like ANOVA for comparing group means.

3. For datasets without normal distribution, we resorted to non-parametric alternatives like the Kruskal-Wallis or Mann-Whitney U tests.

2.2 ANOVA (Analysis of Variance)

ANOVA is used for comparing the means of three or more groups. This technique assesses whether the observed variances in sample means are greater than what is expected by chance.

2.2.1 Mathematical Formulation of F-statistic

The F-statistic in ANOVA is calculated using the formula:

$$F = \frac{MS_{between}}{MS_{within}} \tag{2}$$

where:

- $MS_{between}$ is the Mean Square Between Groups, representing the average variation between the groups. It is calculated as:

$$MS_{between} = \frac{SS_{between}}{DF_{between}} \tag{3}$$

  where:

  - $SS_{between}$ (Sum of Squares Between Groups) measures the variance between the group means and the overall mean, calculated as:

$$SS_{between} = \sum_{i=1}^{k} n_i(\bar{X}_i - \bar{X})^2 \tag{4}$$

  - $DF_{between}$ (Degrees of Freedom Between Groups) is $k - 1$.

- MS_{within} is the Mean Square Within Groups, representing the average variation within the groups. It is calculated as:

$$MS_{within} = \frac{SS_{within}}{DF_{within}} \tag{5}$$

  where:

  - SS_{within} (Sum of Squares Within Groups) measures the variance within each group, calculated as:

$$SS_{within} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2 \tag{6}$$

  - DF_{within} (Degrees of Freedom Within Groups) is $n - k$.

2.2.2 Application of ANOVA in Educational Research

In our study, we applied ANOVA to compare the test scores or other performance metrics of students across different instructional methods, learning environments, or demographic groups. ANOVA was used to assess whether there are significant differences in the mean scores across the groups.

Calculating the F-statistic: The F-statistic was calculated using the ratio of the mean square variance between groups to the mean square variance within groups.

Determining Significance: The significance of the observed F-statistic was determined against the F-distribution. A significant F-statistic suggests that at least one group mean is different from the others.

2.2.3 Interpreting Results

A significant result in ANOVA in our research indicated differences in the means between the groups, suggesting that factors like teaching methods or student demographics have a measurable impact on student performance.

2.3 Kruskal-Wallis H-test

The Kruskal-Wallis H test is a non-parametric statistical test used for comparing the median of two or more independent groups. It is an extension of the Mann-Whitney U test to multiple groups and is particularly useful when the assumptions of ANOVA (such as normality and homogeneity of variances) are not met. This test ranks all the data across groups and compares the sum of these ranks to determine if there are significant differences between the groups.

2.3.1 Mathematical Formulation

The test ranks all data points across all groups and compares the sum of these ranks. The Kruskal-Wallis H statistic is calculated as:

$$H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1) \tag{7}$$

where:

- N is the total number of observations.
- k is the number of groups.
- n_i is the number of observations in group i.
- R_i is the sum of ranks in group i.

2.3.2 Applying the Kruskal-Wallis H-test

The Kruskal-Wallis H-test was employed to analyze differences in diagnostic scores among different ethnic groups. The steps include ranking the data, summing the ranks for each ethnic group, and calculating the test statistic using the aforementioned formula.

2.3.3 Interpreting Results

A higher H statistic indicates significant variations in academic performances across ethnic groups, suggesting the influence of ethnicity on student performance.

2.4 Mann-Whitney U Test

The Mann-Whitney U test is a non-parametric statistical test used to compare differences between two independent groups, particularly when the dependent variable is not normally distributed.

2.4.1 Mathematical Formulation

The Mann-Whitney U statistic is calculated using the following formula:

$$U = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \tag{8}$$

where:

- n_1 is the sample size of the first group.
- n_2 is the sample size of the second group.
- R_1 is the sum of ranks in the first group.

Explanation of the Formula:

- $n_1 n_2$ represents the product of the sample sizes of the two groups. It is a part of the adjustment for ties.
- $\frac{n_1(n_1+1)}{2}$ is the sum of the ranks that would be expected if all the observations in the first group were ranked higher than those in the second group.
- R_1 is subtracted from this expected sum to give the U statistic, which represents the number of times observations in one group rank higher than observations in the other group.

2.4.2 Application of the Mann-Whitney U Test in Educational Research

- The Mann-Whitney U test was applied to assess whether there was a significant difference in the distribution of the scores or responses between the two groups.

- A significant U statistic indicated a difference in the median scores or responses between the two groups, providing insights into the effectiveness of different educational strategies or demographic impacts on educational outcomes.

2.5 Post-Hoc Tests in Our Research

Throughout our research, after conducting ANOVA or Kruskal-Wallis tests, we utilized post-hoc tests to explore specific pairwise differences between groups. These tests were crucial in identifying which groups significantly differed from each other.

2.5.1 Understanding Post-Hoc Tests

Post-hoc tests are statistical comparisons conducted after an initial ANOVA or Kruskal-Wallis test. They help in pinpointing specific group differences when the initial test suggests overall significance across multiple groups.

2.5.2 Mathematical Formulation of Tukey's HSD

Tukey's HSD calculates the critical value q to compare the means of each pair of groups. The test statistic is given by:

$$q = \frac{\bar{x}_i - \bar{x}_j}{\sqrt{\frac{MS_{error}}{n}}} \tag{9}$$

where:

- $\bar{x}_i$ and $\bar{x}_j$ are the means of the two groups being compared.
- MS_{error} is the mean square error from the ANOVA test. It represents the average of the squares of the errors (differences between observed and estimated values).
- n is the number of observations in each group, used to normalize the mean square error.
- The entire expression $\frac{\bar{x}_i - \bar{x}_j}{\sqrt{\frac{MS_{error}}{n}}}$ represents the standardized difference between the two group means. A larger value of q indicates a more significant difference between the group means.

2.5.3 Mathematical Formulation of Dunn's Test

Dunn's test, applied after the Kruskal-Wallis test, is used to determine if significant differences exist between groups in a non-parametric manner. The general formula for Dunn's Test involves calculating the z-statistic for each pairwise group comparison. The z-statistic is given by:

$$z = \frac{R_i - R_j}{\sqrt{\frac{N(N+1)}{12}\left(\frac{1}{n_i} + \frac{1}{n_j}\right)}} \tag{10}$$

where:

- R_i and R_j are the sum of ranks for groups i and j, respectively.
- N is the total number of observations across all groups.
- n_i and n_j are the number of observations in groups i and j, respectively.
- The denominator represents the standard error of the difference between two rank sums, accounting for the sample sizes of the groups being compared.

Interpreting the z-statistic: A larger absolute value of the z-statistic indicates a more significant difference between the ranks of the two groups. The significance of each pairwise comparison is typically assessed using a critical value from the standard normal distribution.

2.5.4 Significance in Our Research

1. **Tukey's HSD:** Allowed us to make pairwise comparisons post-ANOVA, providing specific insights into which groups' mean scores significantly differed.
2. **Dunn's Test:** Used post-Kruskal-Wallis, it helped us identify significant differences in median scores between groups for non-parametric data.

2.5.5 Conclusion

The use of post-hoc tests in our research was instrumental in providing a more nuanced understanding of our data. By employing these tests, we were able to delve deeper into the relationships between various groups, enhancing the overall comprehensiveness and robustness of our findings.

2.6 Utilizing Random Forest

In our research, we employed the Random Forest algorithm for predictive modeling and data analysis. Random Forest, an ensemble learning method, is particularly effective for regression and classification tasks.

Random Forest operates by constructing a multitude of decision trees during training. It outputs the mean prediction of the individual trees for regression tasks or the class that is the mode of the classes for classification tasks.

2.6.1 Application in Our Research

Purpose: We utilized Random Forest to predict various outcomes based on our dataset. Its ability to handle large datasets with higher dimensionality and its robustness to overfitting made it an ideal choice.

Methodology: Our approach involved:

- Preprocessing the data to make it suitable for a tree-based model.
- Tuning hyperparameters, including the number of trees and tree depth, to optimize model performance.
- Assessing model accuracy and interpreting the results for actionable insights.

Mathematical Representation:

$$R(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x; \Theta_b) \tag{11}$$

where:

- $R(x)$ represents the output for an input vector x, indicating the final prediction made by the Random Forest model.
- B is the total number of decision trees in the Random Forest. Each tree contributes to the final prediction.
- $T_b(x; \Theta_b)$ is the prediction of the b-th decision tree for the input vector x. Θ_b represents the parameters or the learning of the b-th tree, including the structure of the tree and the splits made at each node.

3.7.2 Evaluation Metrics: MSE and RMSE

Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are critical for evaluating the performance of regression models.

- **MSE (Mean Squared Error):**

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{12}$$

  where:

  - Y_i is the actual value of the i-th observation.
  - $\hat{Y}_i$ is the predicted value for the i-th observation.
  - n is the total number of observations in the dataset.

- **RMSE (Root Mean Squared Error):**

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{13}$$

  RMSE measures the magnitude of errors, providing a sense of how far the predictions are from the actual values.

2.6.2 Significance in Our Research

We found Random Forest particularly useful for:

1. Managing complex datasets composed of both categorical and numerical variables.
2. Uncovering significant predictors through its feature importance mechanism.
3. Achieving high prediction accuracy while maintaining a low risk of overfitting.

2.7 Correlation Analysis

- Performed Pearson correlation analysis to understand the linear relationships between different variables such as PAL Attendance, Diagnostic Score, and Final Module Score.
- Analyzed correlation coefficients to identify potential relationships and patterns in the data.
- Used heatmaps to visually represent the strength and direction of these correlations, aiding in the interpretation and understanding of the relationships between variables.

2.7.1 Calculating Degrees of Freedom

- Computed degrees of freedom (df) for each dataset, represented as:

$$df = n - 2$$

  where n is the number of samples in the dataset.

2.7.2 Pearson Correlation Coefficient Analysis

- Calculated the Pearson Correlation Coefficient (r_{xy}) for variable pairs.
- The formula used is:

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

  - Where:
  - x_i = Individual sample values of variable x
  - y_i = Individual sample values of variable y
  - $\bar{x}$ = Mean value of variable x
  - $\bar{y}$ = Mean value of variable y
  - The sum is taken over all n pairs of sample values.

3 Methodology

3.1 Data Collection and Preparation

3.1.1 Loading the Dataset

- The dataset was loaded from the Excel file *Copy of MSc Data 1.xlsx*, which contains multiple sheets for different academic years.
- The sheet titled *2022_23* was specifically selected for this research.

3.1.2 Initial Exploration of Data

- Preliminary examination of the dataset involved loading the *2022_23* sheet into a DataFrame and displaying the first few rows.
- This exploration offered insights into various columns such as 'Student ID', 'Ethnicity', 'Gender', 'Highest Qual on Entry', 'Socio-economic classification', and 'Diagnostic Score'.

3.1.3 Data Cleaning and Transformation

- Analyzed the 'Ethnicity' column to obtain counts of each unique ethnic group.
- Defined custom ethnic categories to consolidate similar ethnic groups into broader categories for analysis.
  - Grouped various ethnicities into broader categories like 'Other Asian or Bangladeshi', 'White', 'Black', 'Indian', 'Pakistani', and 'Other Mixed' for a more manageable and meaningful analysis. This was essential to reduce the complexity of the dataset and focus on significant ethnic groupings to understand their impact on student outcomes.

Distribution Analysis and Data Segregation

- Analyzed the 'Mathematics Requirement' column, noting the distribution between students with 'A-Level' and 'GCSE' requirements.
- Created separate DataFrames for 'A-Level' and 'GCSE' students, enabling focused analysis on these two distinct student groups.
- This separation was vital for tailored analysis, as it allowed for the examination of academic performance and specific characteristics within each group.

Data Quality Assessment

- Conducted a NaN value analysis in each DataFrame to identify missing data points and assess data quality.

Socio-economic Classification Mapping

- Mapped ‘Socio-economic classification‘ to new categories like ’Working-Class’, ’Middle Income’, ’Professional’, and ’Unknown’ for more meaningful analysis.
- The reclassification of socio-economic status into broader categories helped in understanding the influence of socio-economic backgrounds on student performance and behavior.

Highest Qualification on Entry Transformation

- Transformed the ‘Highest Qual on Entry‘ column values into standardized categories like ’A-Level’, ’Level 3’, ’Diploma at level 3’, and ’Other Qualification’ to better reflect the students’ educational background.

3.2 Handling Null Values and Predictive Modeling

3.2.1 Dealing with Null Values in ’Final Module Score’

- Identified and displayed rows with null values in the ’Final Module Score’ column.
- This step was critical to assess the extent of missing data and its potential impact on the analysis.

3.2.2 Analysis of ’Diagnostic Score’

- Examined the unique values in the ’Diagnostic Score’ column to understand its distribution.
- Counted the unique values to gauge the variability of diagnostic scores.
- Removed rows with null ’Final Module Score’ from both A-Level and GCSE datasets to ensure data integrity for further analysis.

3.2.3 Predictive Modeling using Random Forest

- Employed Random Forest Regression to predict missing ’Diagnostic Scores’.
- Separated features and target variable, applied one-hot encoding for categorical variables, and ensured consistency across training and prediction datasets.
- Split the data into training and test sets, fitted the Random Forest model, and evaluated its performance using RMSE.
- Used the trained model to predict missing ’Diagnostic Score’ values, integrating these predictions back into the original dataset.
- Verified the changes to the dataset post-prediction to confirm the filling of null values.

Addressed potential biases and inconsistencies through careful visualization and interpretation of the data.

3.3 Statistical Analysis and Group Comparisons

Analysis of Diagnostic Scores by Ethnicity, Socio-Economic Classification, and Highest Qualification on Entry

3.3.1 Ethnicity-Based Analysis

- Provided summary statistics for diagnostic scores across different ethnic groups, including count, mean, standard deviation, minimum, and maximum values.
- Discussed the average diagnostic scores and standard deviations for each ethnicity, highlighting performance trends and consistencies.

3.3.2 Socio-Economic Classification Analysis

- Analyzed diagnostic scores based on socio-economic classification. Presented mean scores and variabilities for categories like 'Working-Class', 'Middle Income', 'Professional', and 'Unknown'.
- Compared the impact of socio-economic backgrounds on diagnostic scores, identifying any significant patterns or disparities.

3.3.3 Highest Qualification on Entry Analysis

- Analyzed diagnostic scores in relation to the highest qualification on entry. Summarized key statistics including means and standard deviations for each qualification category.
- Discussed how prior educational background influences current academic performance, emphasizing any notable observations or patterns.

3.4 Assessment of Assumptions and Choice of Statistical Tests

3.4.1 Fitting Diagnostic Scores to Normal Distribution

- Assessed the normality of diagnostic scores by fitting them to a normal distribution, plotting a histogram, and overlaying the probability density function.
- The mean, standard deviation, and the shape of the distribution were analyzed to check the assumption of normality, crucial for ANOVA.

3.4.2 Homogeneity of Variances

- Evaluated the homogeneity of variances by calculating the ratio of the highest to the lowest standard deviation across ethnic groups.
- The homogeneity of variances is a key assumption for ANOVA, determining whether the test is appropriate for the data.

Based on the results of these assessments, a decision was made on whether to proceed with ANOVA or to use a non-parametric alternative.

3.4.3 ANOVA Test or Kruskal-Wallis H-test

- If the assumptions of normality and homogeneity of variances were met, an ANOVA test was conducted, analyzing differences in diagnostic scores across ethnic groups. Results include F-statistics and p-values.
- If the assumptions were not met, a Kruskal-Wallis H-test, a non-parametric alternative to ANOVA, was used to assess these differences.

3.4.4 Post-Hoc Analysis

- Following ANOVA, Tukey's HSD test was conducted for pairwise comparisons of diagnostic scores between ethnic groups.
- In case of Kruskal-Wallis H-test, a suitable non-parametric post-hoc analysis was performed to identify specific group differences.
- Detailed findings from these post-hoc analyses were presented, specifying groups with significant score differences.

3.5 Implementation of RandomForestRegressor for Predictive Analysis

3.5.1 Preprocessing and Model Fitting

- Describe the preprocessing steps for both GCSE and A-Level datasets, including feature selection, handling categorical and numerical columns, and splitting the data into training and test sets.
- Explain the process of creating a pipeline that incorporates preprocessing steps and the RandomForestRegressor model.
- Detail the training process of the RandomForestRegressor on both datasets.

3.5.2 Model Evaluation and Performance Metrics

- Discuss the evaluation metrics used, such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), and their significance in assessing model performance.
- Present the results of the model evaluation for both datasets, highlighting key findings from the performance metrics.

3.5.3 Analysis of Model Predictions and Performance

- Analyzed the RandomForestRegressor models for A-Level and GCSE datasets. Focused on interpreting the Mean Squared Error (MSE) and R-squared (R^2) values to assess model accuracy.
- Explored the relationship between predicted and actual scores to evaluate the models' prediction reliability.

- Investigated the feature importance from the RandomForestRegressor to determine the impact of various factors like ethnicity, socio-economic classification, qualification on entry, and PAL attendance on student outcomes.

3.6 Correlation Analysis and Summary Statistics

3.6.1 Summary Statistics Analysis

- Conducted a comprehensive statistical analysis of the datasets, focusing on key variables such as PAL Attendance, Diagnostic Score, and Final Module Score.
- Calculated summary statistics (mean, standard deviation, min, and max) across various categories including Ethnicity, Socio-economic Classification, Gender, and Highest Qualification on Entry.
- Highlighted significant findings and disparities within these categories to provide insights into student engagement and performance.

3.6.2 Filtering Dataframes Based on PAL Attendance

- Filtered A-Level and GCSE datasets to only include records with non-zero 'PAL Attendance':

$$FilteredData = \{x \in Data \mid x.PALAttendance > 0\}$$

- Displayed initial rows of these datasets for initial examination.

3.7 Correlation Analysis

1. **Filtering Dataframes Based on PAL Attendance:** Selected only those records from the A-Level and GCSE datasets where 'PAL Attendance' was not zero.
2. **Calculating Degrees of Freedom:** Computed the degrees of freedom for each dataset, essential for various statistical analyses. The degrees of freedom were calculated as the number of samples minus two.
3. **Pearson Correlation Coefficient Analysis:** Carried out correlation analysis using the Pearson Correlation Coefficient to evaluate the linear relationship between pairs of variables. This involved comparing different variables across the datasets to ascertain their interdependencies.
4. **Detailed Correlation Investigations:** Extended the correlation analysis to specific variable pairs, interpreting the results to understand the relationship between different student behaviors and academic outcomes.

3.8 In-depth Analysis of Specific Modules

To gain deeper insights, specific modules such as 'CS1MCP' and 'EE1EMA' were analyzed. This involved examining the impact of PAL engagement on the final module scores for students who failed the diagnostic test in these modules.

3.8.1 Analysis of CS1MCP Module

In the CS1MCP module, the focus was on students who failed the diagnostic test. The average final module scores were calculated for students with high PAL engagement compared to the overall average scores.

3.8.2 Analysis of EE1EMA Module

A similar approach was applied to the EE1EMA module. The analysis compared the average final module scores of students who failed the diagnostic test but showed high PAL engagement against the overall average scores in this module.

3.8.3 Module-Specific Analysis

- The analysis of specific modules, CS1MCP and EE1EMA, involved a structured approach as follows:
  - For CS1MCP: Calculated average final scores for different subsets of students and compared these groups to identify significant differences.
  - For EE1EMA: Followed a similar approach, focusing on average final scores among specific groups of students.
- This approach was mathematically represented and tailored for each module, with a focus on examining the impact of PAL engagement among students who struggled in diagnostic tests.

4 Results

4.1 Part 1: Prediction of Diagnostic Scores Using Random Forest

4.1.1 Data Preparation and Analysis

- Initial analysis revealed missing values in the 'Diagnostic Score' column, with 333 missing values in the A-Level dataset and 209 missing values in the GCSE dataset.
- The 'Final Module Score' column also had missing values, but these rows were removed from the dataset to create a reliable training set.
- The 'Diagnostic Score' column exhibited a diverse range of values with 21 unique scores identified.

4.1.2 Model Training and Prediction

- A Random Forest model was trained separately for both A-Level and GCSE datasets to predict missing 'Diagnostic Score' values.
- For the A-Level dataset, the model achieved a Root Mean Square Error (RMSE) of 3.46, indicating the model's prediction accuracy.
- The GCSE dataset model achieved an RMSE of 3.72, showing a comparable level of accuracy.

4.1.3 Results of Prediction

- The trained models successfully predicted missing 'Diagnostic Scores'. The first ten predicted values for the A-Level dataset were: [6.84, 8.23, 8.69, 7.905, 4.215, 6.485, 8.21, 8.195, 8.265, 6.99].
- For the GCSE dataset, the first ten predicted values were: [8.42, 9.33, 7.955, 7.905, 11.15, 7.385, 7.61, 10.895, 8.805, 8.495].
- The predicted values were then integrated back into the respective datasets, replacing the missing values.

4.1.4 Visualization of Predicted Scores

- Histograms were generated to visualize the distribution of the predicted 'Diagnostic Scores' alongside the combine scores for both datasets.

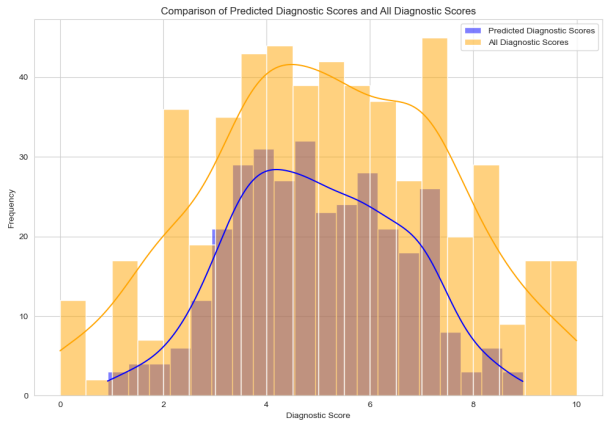


Figure 1: Distribution of predicted diagnostic scores alongside the combined dataset

- The histograms revealed a varied distribution, with the majority of scores falling within the mid-range for both actual and predicted values.

4.1.5 Insights and Interpretation

- The successful prediction of missing 'Diagnostic Scores' using Random Forest models not only filled the data gaps but also provided insights into the potential academic capabilities of students whose scores were initially missing.

- The similar distribution patterns between predicted and actual scores suggest that the models captured the underlying trends in the data effectively.

4.2 Demographic and Diagnostic Score Analysis in A-Level and GCSE Datasets

4.2.1 Gender and Ethnicity Distribution

- A-Level: Highest count - Pakistani males (101), lowest - White females (7).
- GCSE: Highest count - White males (86), lowest - Other Asian or Bangladeshi females (10).

4.2.2 Average Diagnostic Score Analysis

- A-Level: Highest average - Indian females (5.898), lowest - White females (4.441).
- GCSE: Highest average - Indian males (8.063), lowest - Other Mixed females (3.301).

4.2.3 Socio-economic Classification Analysis

- A-Level: Highest average score - 'Working-Class' (5.962), lowest - 'Unknown' (3.387).
- GCSE: Highest average score - 'Semi-routine occupations' (8.601), lowest - 'Unknown' (5.799).

4.2.4 Highest Qualification on Entry Analysis

- A-Level: Highest average score - 'A-Level' qualification (5.619), lowest - 'Other Qualification' (3.234).
- GCSE: Highest average score - 'Other Qualification' (8.600), lowest - 'Unknown' (5.828).

4.3 Statistical Analysis of Diagnostic Scores by Ethnicity

4.3.1 Ethnicity-Based Analysis

- Analyzed summary statistics for diagnostic scores across different ethnic groups. For example, in the A-Level dataset, Black students had an average diagnostic score of 5.02 with a standard deviation of 2.00.
- Notable findings include the highest average score among White students (5.40) and the lowest among Other Mixed students (4.43) in the A-Level dataset.

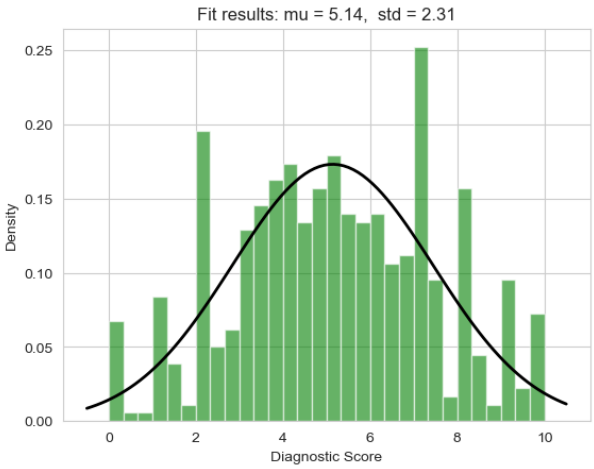


Figure 2: Gaussian distribution

4.3.2 Fitting Diagnostic Scores to Normal Distribution

- Diagnostic scores were fitted to a normal distribution. The mean (mu) and standard deviation (std) were calculated, with mu = 5.14 and std = 2.31 for the A-Level dataset.
- Histograms and probability density functions indicated a generally bell-shaped distribution, suggesting a normal distribution of diagnostic scores.
- Conducted the Shapiro-Wilk test for normality. The test resulted in a p-value of 0.000397, suggesting a deviation from normality in the A-Level dataset.

4.3.3 Homogeneity of Variances

- Assessed homogeneity of variances by comparing the highest and lowest standard deviations across ethnic groups. The ratio was 1.24, indicating a moderate level of homogeneity.

4.3.4 ANOVA Test or Kruskal-Wallis H-test

- Conducted an ANOVA test to analyze differences in diagnostic scores across ethnic groups. The results showed significant differences (F-statistic: 7.37, p-value: 8.44e-07).
- In cases where normality or homogeneity of variances assumptions were not met, the Kruskal-Wallis H-test was used, which also indicated significant differences (statistic: 34.33, p-value: 2.05e-06).

4.3.5 Post-Hoc Analysis

- Performed Tukey's HSD test for pairwise comparisons post-ANOVA. Significant differences were noted, for example, between Black and Indian students (mean difference: 1.1794, p-value: 0.0022). Below is the complete table:

Table 1: Pairwise Comparison Results

Group 1	Group 2	Mean Difference	p-adj	Lower	Upper	Reject
Black	Indian	1.1794	0.0022	0.2909	2.0679	True
Black	Other Asian or Bangladeshi	0.607	0.3953	-0.2994	1.5134	False
Black	Other Mixed	-0.2953	0.932	-1.1789	0.5882	False
Black	Pakistani	0.7641	0.0802	-0.05	1.5782	False
Black	White	1.1469	0.0015	0.3052	1.9886	True
Indian	Other Asian or Bangladeshi	-0.5724	0.5327	-1.5342	0.3895	False
Indian	Other Mixed	-1.4747	0.0001	-2.415	-0.5344	True
Indian	Pakistani	-0.4153	0.7543	-1.2907	0.4602	False
Indian	White	-0.0325	1.0	-0.9336	0.8687	False
Other Asian or Bangladeshi	Other Mixed	-0.9023	0.0778	-1.8595	0.0549	False
Other Asian or Bangladeshi	Pakistani	0.1571	0.9961	-0.7365	1.0507	False
Other Asian or Bangladeshi	White	0.5399	0.5468	-0.3789	1.4587	False
Other Mixed	Pakistani	1.0594	0.007	0.1891	1.9298	True
Other Mixed	White	1.4422	0.0001	0.546	2.3385	True
Pakistani	White	0.3828	0.7739	-0.4451	1.2107	False

4.3.6 Pairwise Comparison Results

The table below presents the results of pairwise comparisons conducted using Tukey's HSD test post-ANOVA. The columns in the table are as follows:

- **Group 1**: The first group being compared.
- **Group 2**: The second group being compared.
- **Mean Difference**: The mean difference in diagnostic scores between the two groups.
- **p-adj**: The adjusted p-value indicating the significance of the difference.
- **Lower**: The lower bound of the confidence interval for the mean difference.
- **Upper**: The upper bound of the confidence interval for the mean difference.
- **Reject**: Indicates whether the null hypothesis of no difference is rejected (True) or not (False).

These results provide insights into the variations in diagnostic scores across different ethnic groups, highlighting significant differences between some groups, which may have implications for educational assessments.

4.4 Statistical Analysis of A-Level Diagnostic Scores by Socio-Economic Classifications

4.4.1 Summary Statistics

- Analyzed diagnostic scores across different socio-economic groups in the A-Level dataset.
- Observed significant variations in diagnostic scores among these groups.
- For instance, the average diagnostic score for the 'Middle Income' group was 5.10 with a standard deviation of 2.17, and for the 'Professional' group, it was 4.80 with a standard deviation of 2.28.

4.4.2 Shapiro-Wilk Test for Normality

Table 2: Shapiro-Wilk Test Results for Socio-Economic Classifications (A-Level)

Socio-economic Classification	W-Statistic	p-value
Middle Income	0.988	0.265
Professional	0.980	0.037
Unknown	0.982	0.041
Working-Class	0.974	0.058

- Shapiro-Wilk test results indicate deviations from normal distribution for some groups, especially 'Professional' ($p = 0.037$) and 'Unknown' ($p = 0.041$).

4.4.3 Homogeneity of Variances

- Assessed the homogeneity of variances by comparing the highest and lowest standard deviations across socio-economic groups.
- The calculated ratio was 1.090, suggesting a moderate level of homogeneity.

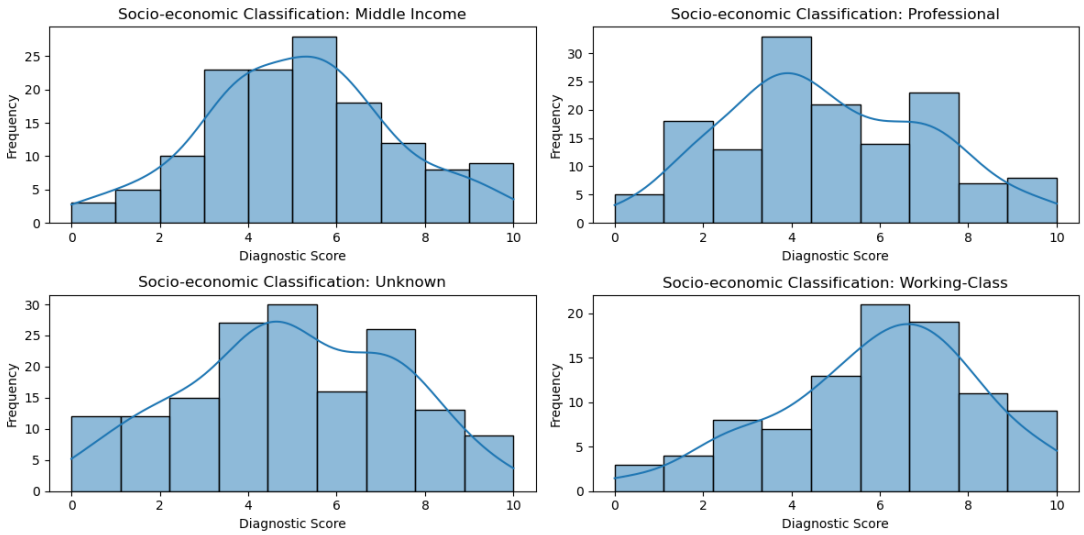


Figure 3: Gaussian distribution

4.4.4 Kruskal-Wallis H Test

- Conducted the Kruskal-Wallis H test due to deviations from normality.
- The test revealed significant differences among socio-economic groups (Test Statistic: 17.156, p-value: 0.000657).

4.4.5 Dunn's Post-Hoc Analysis

Table 3: Dunn's Post-Hoc Test Results for Socio-Economic Classifications (A-Level)

Group 1	Group 2	Mean Difference	Adjusted p-value	Significant
Middle Income	Professional	-	1.000	False
Middle Income	Unknown	-	1.000	False
Middle Income	Working-Class	-	0.02199	True
Professional	Unknown	-	1.000	False
Professional	Working-Class	-	0.000359	True
Unknown	Working-Class	-	0.007216	True

- Dunn's post-hoc test indicated significant differences between certain pairs, notably between 'Middle Income' and 'Working-Class', and 'Professional' and 'Working-Class'.

4.5 Statistical Analysis of GCSE Diagnostic Scores by Socio-Economic Classifications

4.5.1 Summary Statistics

- Analyzed diagnostic scores across different socio-economic groups in the GCSE dataset.
- For example, the average diagnostic score for the 'Middle Income' group was 7.48 with a standard deviation of 3.29, and for the 'Professional' group, it was 7.04 with a standard deviation of 3.12.

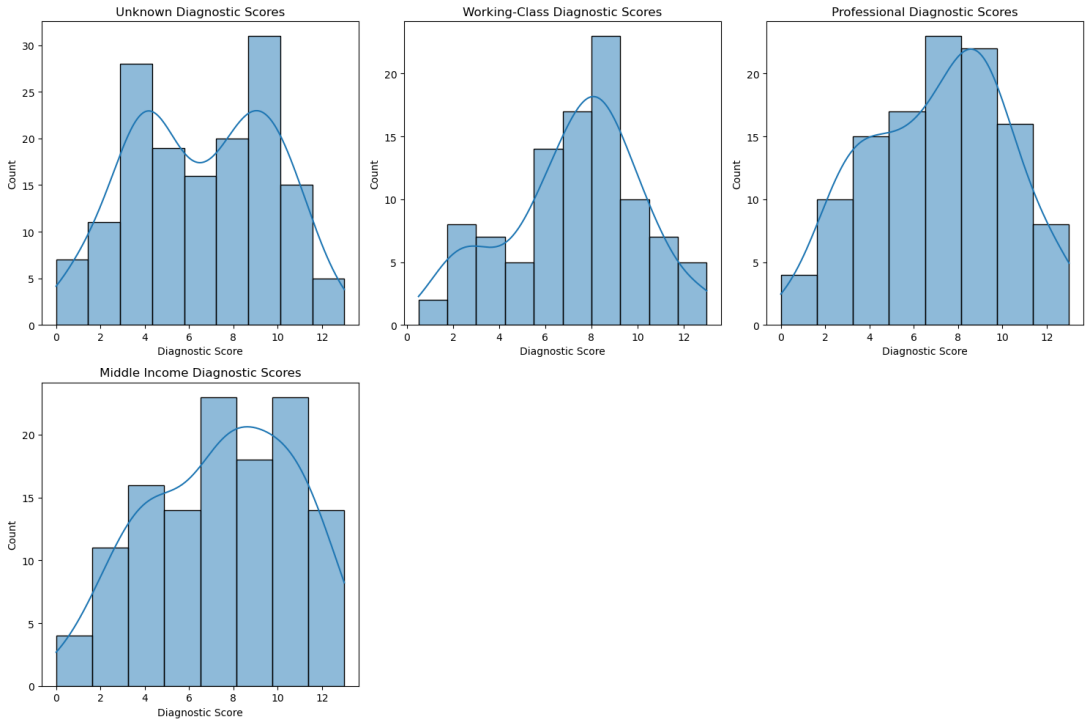


Figure 4: Gaussian distribution

4.5.2 Shapiro-Wilk Test for Normality

Table 4: Shapiro-Wilk Test Results for Socio-Economic Classifications (GCSE)

Socio-economic Classification	W-Statistic	p-value
Middle Income	0.971	0.008
Professional	0.976	0.036
Unknown	0.967	0.001
Working-Class	0.967	0.016

- The Shapiro-Wilk test results indicate deviations from normal distribution for several groups, particularly 'Unknown' and 'Working-Class'.

4.5.3 Homogeneity of Variances

- The ratio of the highest to the lowest standard deviation was 1.143, indicating a moderate level of variance homogeneity among the socio-economic groups.

4.5.4 Kruskal-Wallis H Test Results

The Kruskal-Wallis H test was conducted to examine the differences in diagnostic scores across various socio-economic classifications in the GCSE dataset. This non-parametric test is appropriate for data that do not follow a normal distribution.

- The test statistic and p-value from the Kruskal-Wallis H test provide insights into whether the differences in median scores across socio-economic groups are statistically significant.

Table 5: Kruskal-Wallis H Test Results for Socio-Economic Classifications (GCSE)

Test Statistic	p-value
5.545	0.136

- A Kruskal-Wallis H test statistic of 5.545 with a p-value of 0.136 indicates that there are no statistically significant differences in the median diagnostic scores among the different socio-economic groups in the GCSE dataset.
- This result suggests that, at least in terms of median scores, socio-economic status does not have a significant impact on the diagnostic scores of GCSE students.

4.5.5 Interpretation of the Kruskal-Wallis H Test Results

The Kruskal-Wallis H test was conducted to determine if there were significant differences in diagnostic scores across various socio-economic groups in the GCSE dataset. The test yielded a statistic of 5.545 and a p-value of 0.136.

- **Test Statistic and P-value**:
  - The test statistic of 5.545 is a measure of the difference between groups.
  - The p-value indicates the probability of observing the test statistic under the null hypothesis, which in this case posits no difference across groups.
- **P-value Threshold**: Conventionally, a p-value below 0.05 is considered statistically significant. It suggests that the observed data are unlikely under the null hypothesis.
- **Interpretation of Results**:
  - A p-value of 0.136 suggests a 13.6% probability of observing such data if the null hypothesis is true.
  - Since this value exceeds the 0.05 threshold, it does not provide strong evidence against the null hypothesis.
- **Conclusion**: The results imply that the differences in median diagnostic scores among the socio-economic groups in the GCSE dataset are not statistically significant. This indicates that socio-economic status, as classified in this study, does not have a noticeable impact on the median diagnostic scores of the students.

5 Analysis of Diagnostic Scores by Gender in A-Level and GCSE Datasets

5.1 Comparative Gender Analysis

5.1.1 Summary Statistics

- A-Level: Males (Mean: 5.13, SD: 2.35), Females (Mean: 5.17, SD: 2.12).
- GCSE: Males (Mean: 7.32, SD: 3.01), Females (Mean: 5.94, SD: 3.45).

5.1.2 Statistical Tests and Results

- A-Level: No significant gender differences (ANOVA, p = 0.128).
- GCSE: Significant differences (Kruskal-Wallis H, p = 0.033).

5.1.3 Conclusion

- Gender impacts diagnostic scores significantly in GCSE, not in A-Level.
- Highlights the importance of gender consideration in educational strategies.

6 Statistical Analysis of Diagnostic Scores by Highest Qualification on Entry

- A-Level: Mean scores vary from 3.20 (Unknown) to 5.62 (A-Level), indicating diverse achievement levels.
- GCSE: Mean scores range from 5.83 (Unknown) to 8.60 (Other Qualification), also showing variability.

6.1 Normality and Statistical Tests

- A-Level: Non-normal distributions in A-Level (p=0.001122) and Unknown (p=0.001815) groups.
- GCSE: Similar patterns with significant deviations in A-Level (p=0.000351) and Level 3 (p=0.000217).
- Kruskal-Wallis H Test: Significant differences in both datasets (A-Level: p 2.21e-14, GCSE: p 5.18854e-06).

6.2 Post-Hoc Analysis

- Significant pairwise differences found, indicating the impact of prior qualifications on diagnostic scores.

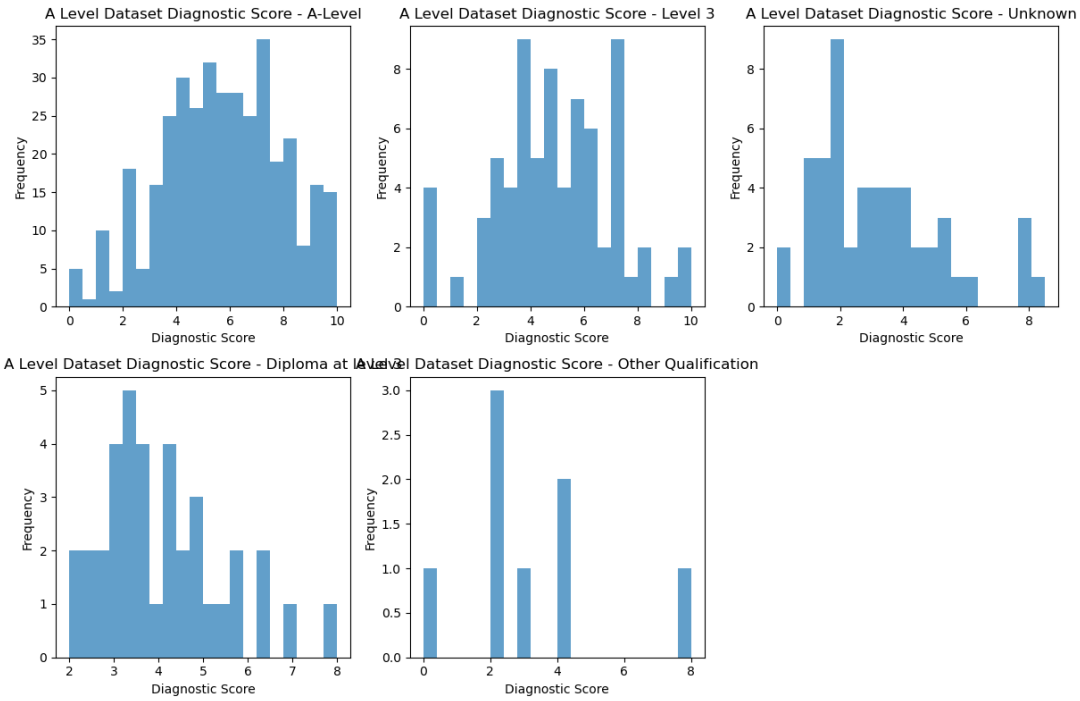


Figure 5: Gaussian distribution

6.2.1 Pairwise Comparisons

Table 6: Post-Hoc Mann-Whitney U Test Pairwise Comparisons

Comparison	p-value	Significance
A-Level vs Diploma at level 3	5.52e-06	Significant
A-Level vs Level 3	0.0064	Significant
A-Level vs Other Qualification	0.0068	Significant
A-Level vs Unknown	1.06e-11	Significant
Diploma at level 3 vs Level 3	0.0229	Significant
Diploma at level 3 vs Other Qualification	0.1024	Not Significant
Diploma at level 3 vs Unknown	0.0037	Significant
Level 3 vs Other Qualification	0.0411	Significant
Level 3 vs Unknown	1.16e-05	Significant
Other Qualification vs Unknown	0.8703	Not Significant

6.3 Concluding Remarks

- The analyses underscore the significant role of students' qualifications and ethnicity in their academic performance.
- The findings inform potential areas for focused educational interventions in A-Level and GCSE curricula.

7 Analysis of Final Module Scores by Socio-Economic Classification

7.0.1 A-Level Dataset

- Black: Mean = 40.19, Std Dev = 23.32
- White: Mean = 41.43, Std Dev = 24.78

7.0.2 GCSE Dataset

- Black: Mean = 50.65, Std Dev = 22.02
- White: Mean = 57.47, Std Dev = 21.63

7.1 Statistical Analysis and Test Selection

7.1.1 Variability and Normality

- Conducted Shapiro-Wilk tests for normality; A-Level showed no deviation, while GCSE indicated non-normal distributions for some groups.

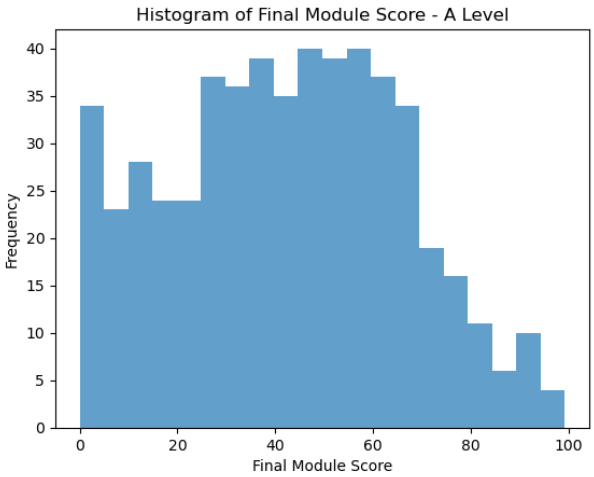


Figure 6: Distribution of A-Level Final Module Scores by Ethnicity

7.1.2 Test Selection

- Chose non-parametric Kruskal-Wallis H test for both datasets due to moderate variability and mixed normality results.

7.1.3 Kruskal-Wallis H Test Findings

- A-Level: No significant differences found (p-value = 0.170).
- GCSE: Significant differences identified (p-value = 0.019).

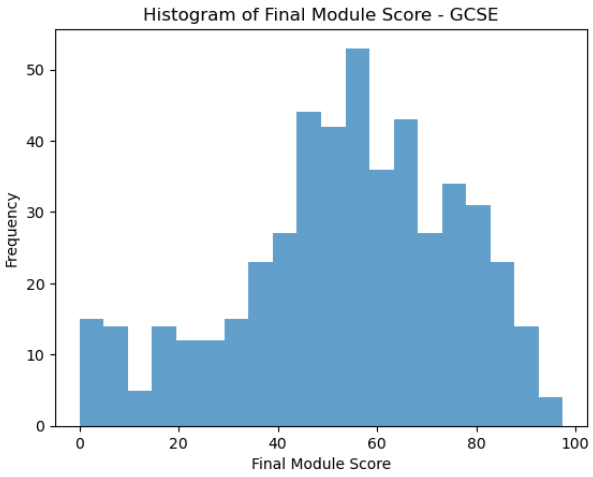


Figure 7: Distribution of GCSE Final Module Scores by Ethnicity

7.2 Findings and Post-Hoc Analysis for GCSE Ethnic Groups

7.2.1 Post-Hoc Mann-Whitney U Test

- After identifying significant differences using the Kruskal-Wallis H test, Post-Hoc Mann-Whitney U tests were performed for pairwise comparisons.
- These tests aimed to pinpoint which specific ethnic groups had statistically significant differences in their final module scores.
- Significant findings from the Post-Hoc analysis are summarized in the table below.

Table 7: Post-Hoc Mann-Whitney U Test Results for GCSE

Ethnic Group Pair	Comparison	p-value
Black vs Indian	Not Significant	0.05839
Black vs Other Asian or Bangladeshi	Not Significant	0.06156
...	...	...
Other Mixed vs White	Significant	0.00408

8 Analysis of Final Module Scores by Socio-Economic Classification

8.1 A-Level Dataset: Socio-Economic Classification Analysis

8.1.1 Summary Statistics and SD Ratio

- Analyzed the mean, variance, and standard deviation of final module scores for each socio-economic class.
- Observed statistics:
  - Middle Income: Mean = 42.50, Variance = 597.71, SD = 24.45

- Professional: Mean = 43.71, Variance = 537.70, SD = 23.19
- Unknown: Mean = 39.89, Variance = 592.64, SD = 24.34
- Working-Class: Mean = 43.78, Variance = 467.95, SD = 21.63

- The SD ratio of 1.13 suggests moderate variance homogeneity.

8.1.2 Histogram Analysis and Shapiro-Wilk Test

- Histograms for each socio-economic class displayed diverse distributions, with variations in normality and skewness.
- Shapiro-Wilk test results indicated non-normal distributions in some classes, suggesting a need for non-parametric testing.

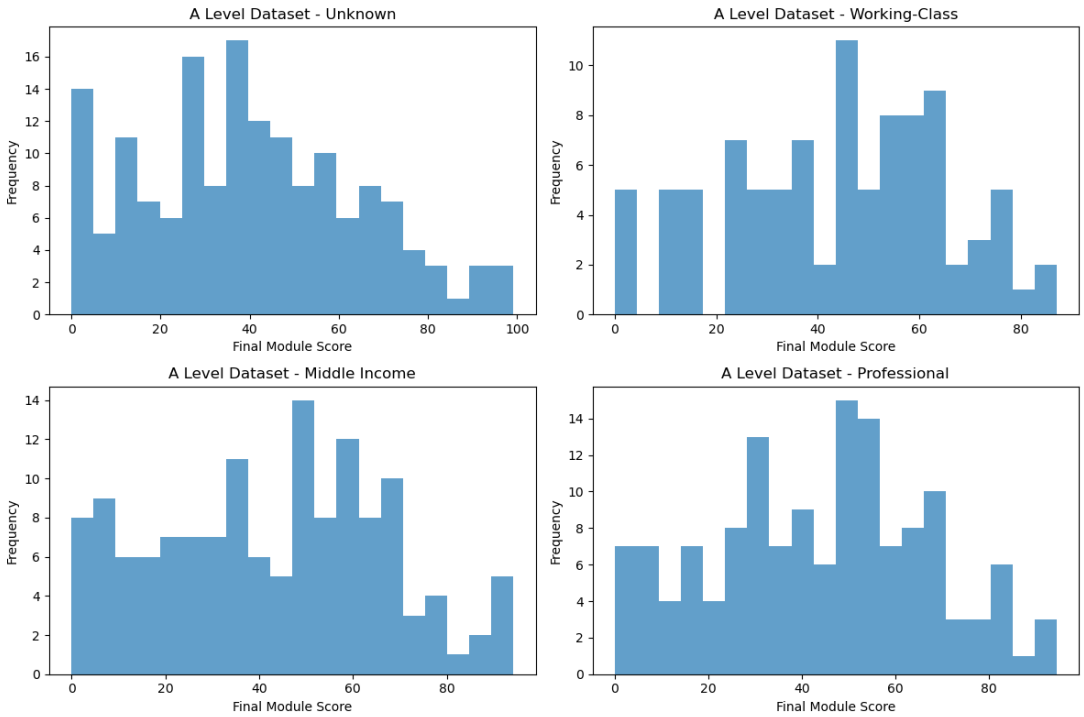


Figure 8: Histogram Analysis

8.1.3 Kruskal-Wallis H Test

- The test results for the A-Level dataset: Statistic = 3.153, p-value = 0.369.
- Interpretation: The p-value is greater than the conventional alpha level (usually 0.05). This indicates that the observed differences in median final module scores across socio-economic classes are not statistically significant. In other words, any differences observed could very well be due to random chance rather than a true difference in academic performance among socio-economic classes.

8.2 GCSE Dataset: Socio-Economic Classification Analysis

8.2.1 Summary Statistics and SD Ratio

- Similar analyses performed for the GCSE dataset showed slight variance in score distribution, with an SD ratio of 1.21.

8.2.2 Histogram Analysis and Shapiro-Wilk Test

- Varied distributions were observed across socio-economic classes, with several indicating non-normality.

8.2.3 Kruskal-Wallis H Test

- The test results for the GCSE dataset: Statistic = 3.819, p-value = 0.282.
- Interpretation: Similar to the A-Level dataset, the p-value exceeds the typical threshold of 0.05. This result implies that the differences in median final module scores across socio-economic classes are not statistically significant. It suggests that socio-economic classification does not have a discernible impact on the final module scores in the GCSE dataset.

8.3 Rationale for Excluding Zero Scores

- Zero scores in final module scores were excluded from this analysis.
- This decision was based on the assumption that zero scores represent unique circumstances, such as non-participation or withdrawal, rather than academic ability or performance.
- Including zero scores could skew the distribution and affect the accuracy of statistical tests, especially in understanding the impact of socio-economic classification on actual academic achievement.

9 Analysis of Final Module Scores by Highest Qualification on Entry

9.1 A-Level Dataset: Analysis by Highest Qualification on Entry

9.1.1 Histogram Analysis and Shapiro-Wilk Test for Normality

- Histograms indicate diverse score distributions for different qualifications.
- Shapiro-Wilk test results:
  - A-Level and Diploma at level 3 showed non-normal distributions (p = 0.002 and 0.003, respectively).

- Level 3, Other Qualification, and Unknown categories closer to normal distribution (p = 0.087, 0.244, and 0.263).

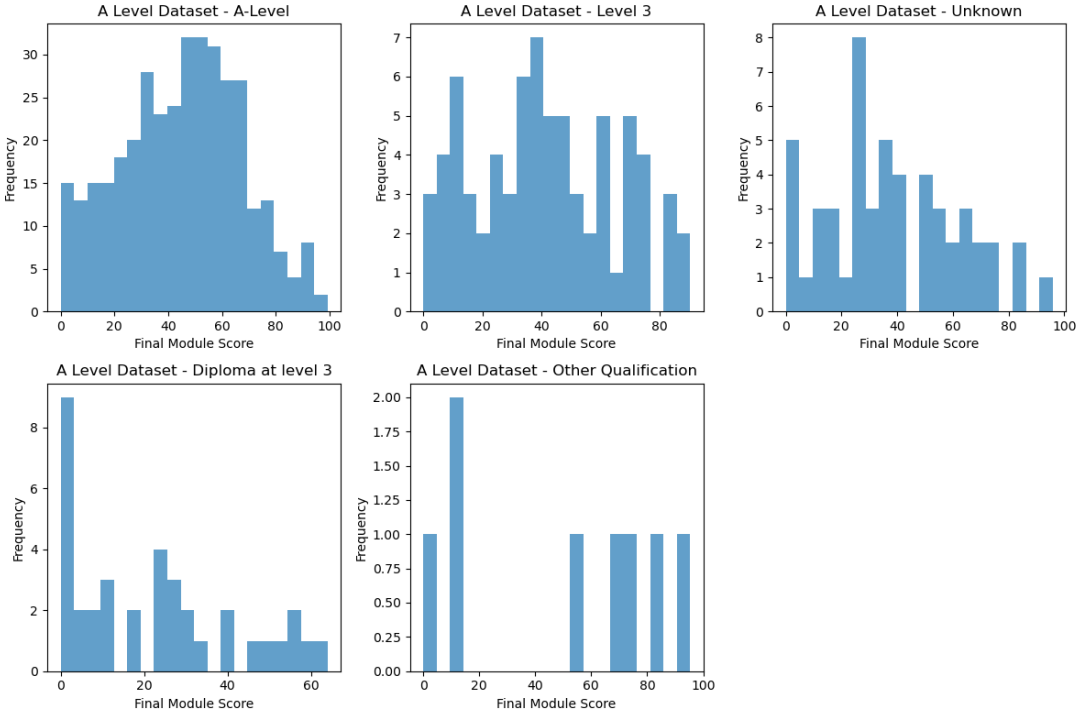


Figure 9: Final Module Scores by Highest Qualification on Entry

9.1.2 Summary Statistics and SD Ratio

- Final module scores analyzed for different qualifications.
- A-Level: Mean = 44.84, SD = 22.55; Other Qualification: Mean = 49.05, SD = 36.63.
- SD ratio across categories approximately 1.82, indicating variance in scores.

9.1.3 Kruskal-Wallis H Test

- Test results: Statistic = 31.362, p-value 2.58e-06.
- Interpretation: Significant differences in median final module scores across different qualifications.

9.2 GCSE Dataset: Analysis by Highest Qualification on Entry

9.2.1 Summary Statistics and SD Ratio

- Analyzed final module scores for different 'Highest Qual on Entry' categories in the GCSE dataset.
- Observed statistics:
  - A-Level: Mean = 60.73, Variance = 583.77, SD = 24.16

- Diploma at level 3: Mean = 39.83, Variance = 370.28, SD = 19.24
- Level 3: Mean = 54.92, Variance = 435.11, SD = 20.86
- Other Qualification: Mean = 51.00, Variance = 72.00, SD = 8.49 (Insufficient data)
- Unknown: Mean = 46.58, Variance = 535.42, SD = 23.14

- The SD ratio is approximately 2.85, indicating significant variability in score distribution.

9.2.2 Histogram Analysis and Shapiro-Wilk Test

- Histograms exhibited varying distributions for different qualifications.
- Shapiro-Wilk test results for normality:
  - A-Level: p-value ¡ 0.001 (non-normal)
  - Diploma at level 3: p-value = 0.660 (normal)
  - Level 3: p-value ¡ 0.001 (non-normal)
  - Unknown: p-value = 0.728 (normal)
  - Other Qualification: Insufficient data

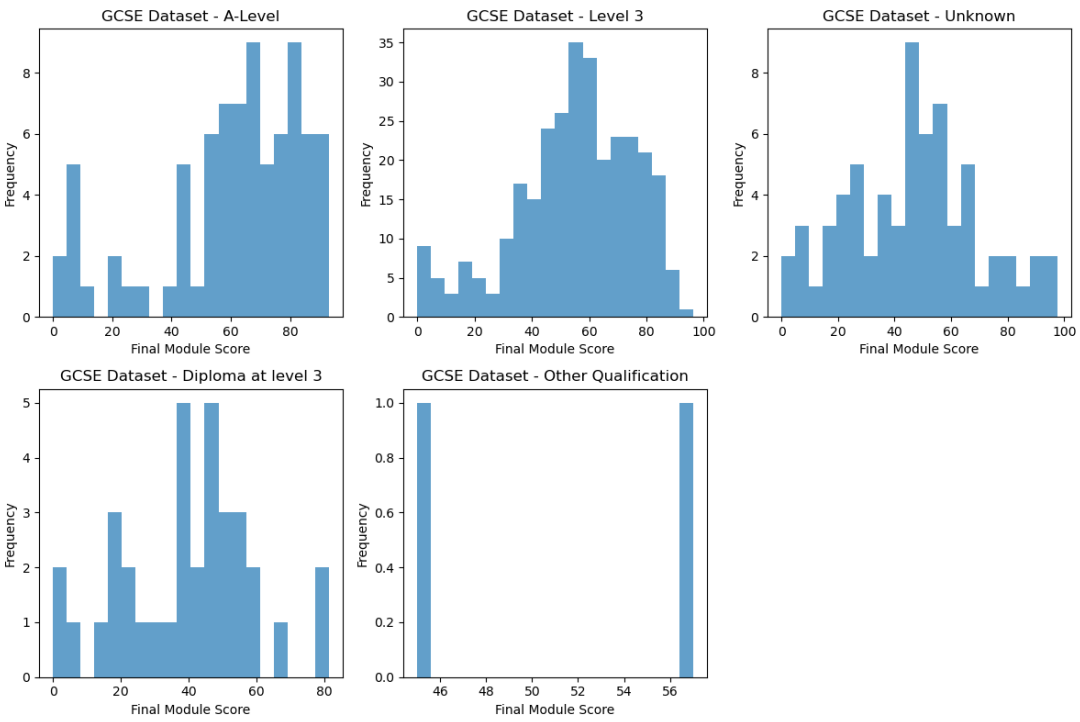


Figure 10: Final Module Scores by Highest Qualification on Entry

9.2.3 Kruskal-Wallis H Test

- Test results: Statistic = 37.030, p-value 4.53e-08.
- Interpretation: Significant differences in median final module scores across different qualifications.

9.2.4 Post-Hoc Mann-Whitney U Test Results

- Significant pairwise differences were observed between:
  - A-Level and Diploma at level 3: p-value 1.86e-06
  - A-Level and Level 3: p-value 0.00363
  - A-Level and Unknown: p-value 6.76e-05
  - Diploma at level 3 and Level 3: p-value 2.13e-05
  - Level 3 and Unknown: p-value 0.00215
- No significant differences in other pairwise comparisons.

10 Analysis of Predicted and Actual Final Module Scores

10.1 Analysis Approach

- The analysis utilizes RandomForestRegressor models to predict final module scores for both A-Level and GCSE datasets.
- The models were trained on several features excluding 'Final Module Score' and 'Student ID' to avoid data leakage and ensure model accuracy.
- The performance of these models was evaluated using the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), providing insights into the prediction accuracy.

10.2 Model Evaluation and Results

- A-Level Dataset:
  - MSE: 639.39, RMSE: 25.29. The RMSE value indicates the average deviation of the predicted scores from the actual scores.
- GCSE Dataset:
  - MSE: 486.75, RMSE: 22.06. The lower RMSE compared to the A-Level dataset suggests slightly better prediction accuracy for the GCSE dataset.

10.3 Comparison of Predicted and Actual Scores

- For both datasets, we added a 'Predicted Final Module Score' column to compare against the actual 'Final Module Score'.
- Visual comparison was made using line plots that represent a sample of the data, providing a clear comparison between predicted and actual scores.

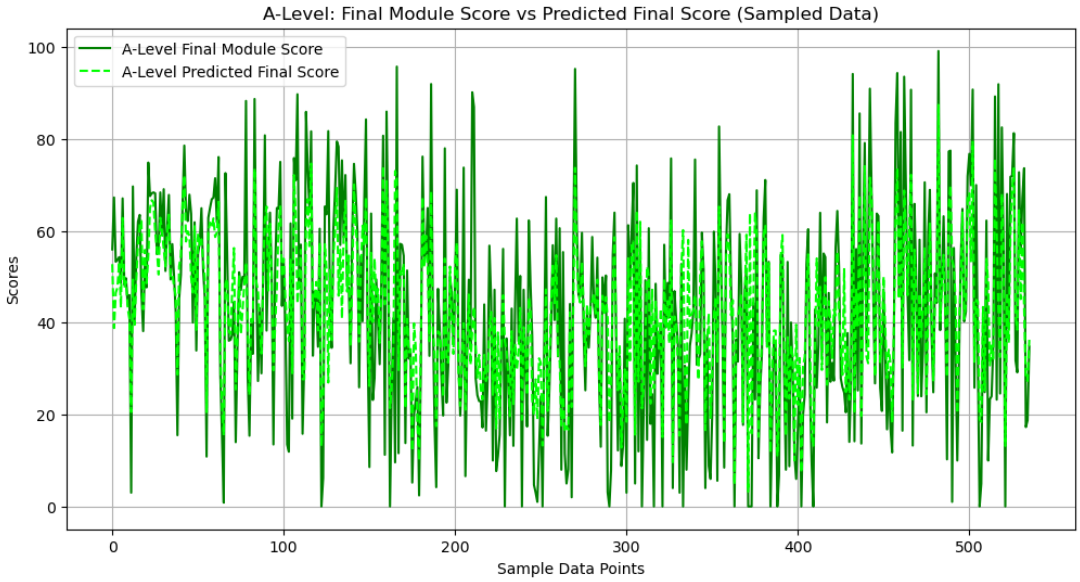


Figure 11: Comparison of Predicted and Actual Scores

10.4 Value Added Analysis

- 'Value Added' was calculated as the difference between the actual and predicted final module scores. This metric helps in understanding how much value is added or lost through the educational process.

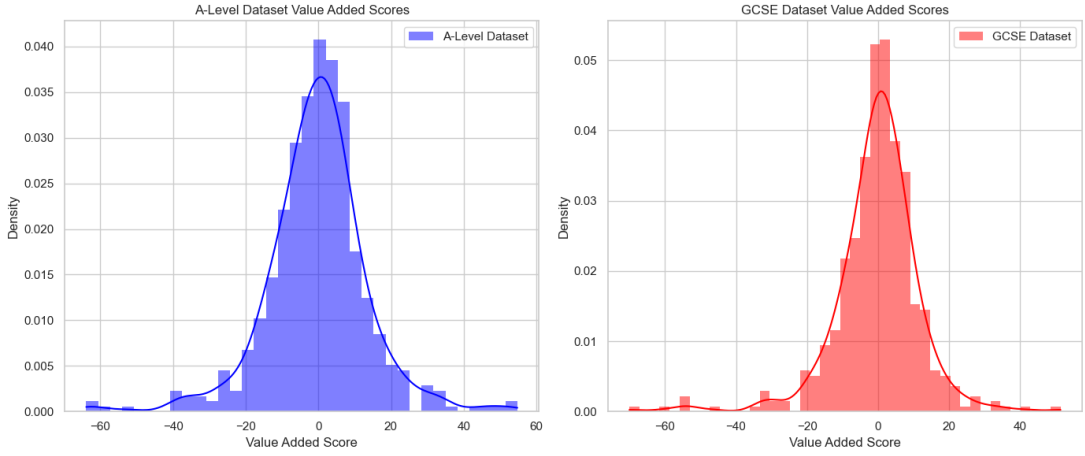


Figure 12: Comparison of Predicted and Actual Scores

- The models successfully predicted the final scores.

- The value-added analysis provides a nuanced understanding of individual performance beyond raw scores.

11 Comprehensive Correlation Analysis

11.1 Data Selection and Preparation

- Focused on the impact of PAL Attendance, this analysis includes only students with non-zero PAL Attendance in the A-Level and GCSE datasets. The goal is to assess whether PAL sessions influence performance improvements.
- Calculated degrees of freedom for each dataset are 209 for A-Level and 187 for GCSE, essential for the statistical significance assessment in correlation analysis.

11.2 Correlation Analysis

- Pearson correlation coefficients were computed to understand the relationship between 'Value Added', 'Final Module Score', 'Diagnostic Score', and 'PAL Attendance' across both datasets.

11.2.1 PAL Attendance and Value Added

- A-Level Data: Pearson coefficient is 0.105, indicating a very weak positive correlation. This suggests minimal impact of PAL Attendance on Value Added scores.
- GCSE Data: Pearson coefficient is approximately 0.00003, showing virtually no linear relationship.
- Conclusion: Correlations in both datasets are not statistically significant at the 0.05 level (critical value approximately 0.139 for this sample size).

11.2.2 PAL Attendance and Final Module Scores

- A-Level Data: Pearson coefficient is 0.157, showing a weak positive correlation.
- GCSE Data: Pearson coefficient is -0.00061, indicating no meaningful correlation.
- Conclusion: Only A-Level data shows a weak but statistically significant correlation at the 0.05 level.

11.2.3 PAL Attendance and Diagnostic Scores

- A-Level Data: Pearson coefficient is 0.052, indicating a very weak correlation.
- GCSE Data: Pearson coefficient is -0.129, suggesting a weak negative correlation.
- Conclusion: Correlations are not statistically significant at the 0.05 level.

11.2.4 Final Module Scores and Diagnostic Scores

- A-Level Data: Pearson coefficient is 0.196, indicating a weak positive correlation.
- GCSE Data: Pearson coefficient is 0.288, suggesting a moderate positive correlation.
- Conclusion: Both datasets show a statistically significant correlation at the 0.05 level, more pronounced in the GCSE dataset.

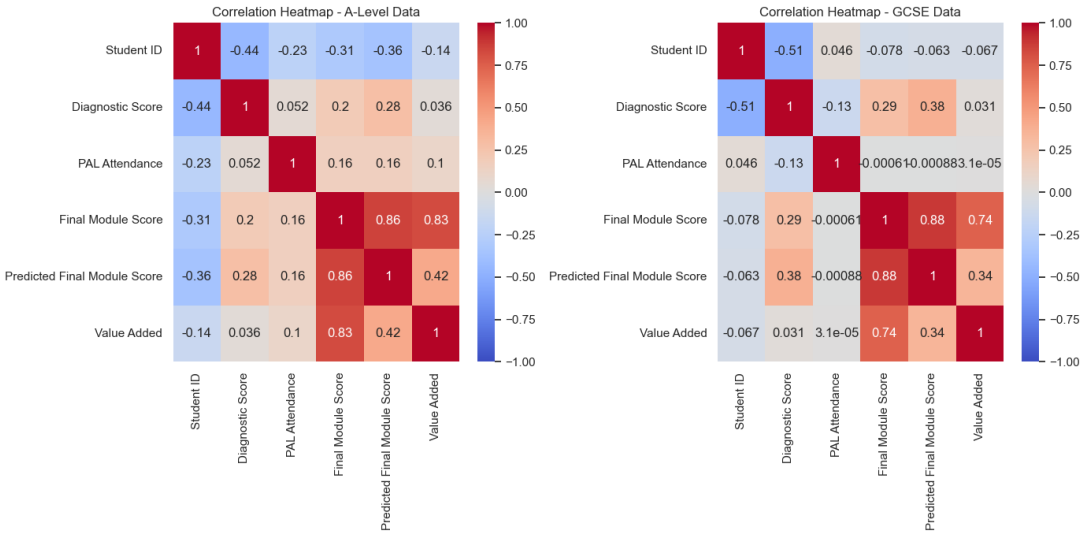


Figure 13: Comparison of Predicted and Actual Scores

11.3 Heatmap Visualization

- Heatmaps provide a visual representation of the correlation matrices, offering an intuitive overview of the relationships among different variables in each dataset.

11.4 Conclusion

- The correlation analysis indicates varying degrees of relationships between the studied variables in the A-Level and GCSE datasets.
- While PAL Attendance shows limited predictive power for academic performance, Diagnostic Scores demonstrate a more substantial correlation with Final Module Scores, especially in the GCSE dataset.

12 Key Findings from Summary Statistics Analysis

12.1 PAL Attendance Analysis

12.1.1 Ethnicity

- In the A-Level dataset, 'White' ethnicity shows the highest average PAL Attendance (4.88), while 'Other Asian or Bangladeshi' has the lowest (3.14).

- In the GCSE dataset, 'Indian' ethnicity records the highest average (4.43), and 'Black' the lowest (2.79).

12.1.2 Socio-economic Classification

- For A-Level, the 'Unknown' classification has the highest average PAL Attendance (4.12), with 'Middle Income' showing the highest maximum attendance (17).
- In the GCSE dataset, 'Middle Income' leads with an average attendance of 3.46.

12.1.3 Gender

- A-Level: Females exhibit a higher average PAL Attendance (4.15) compared to males (3.70).
- GCSE: Both genders show similar averages (Females: 3.23, Males: 3.23).

12.1.4 Highest Qualification on Entry

- A-Level: 'Diploma at level 3' students have the highest average PAL Attendance (6.00), while 'Unknown' qualifications have the lowest (3.05).
- GCSE: 'Unknown' qualifications lead with the highest average (3.74).

12.2 Comprehensive Value Added Analysis

- Ethnicity: A-Level highest 'Value Added' for 'Indian' (4.10), lowest for 'Other Mixed' (-1.74). GCSE highest for 'Black' (1.98), lowest for 'Other Mixed' (-0.85).
- Socio-economic Classification: A-Level highest for 'Middle Income' (2.79). GCSE highest for 'Working-Class' (2.57).
- Gender: A-Level females (2.92) outperform males (1.32). GCSE shows a smaller gap (Females: 2.63, Males: 0.40).
- Highest Qualification on Entry: A-Level 'Level 3' (8.81), GCSE 'Diploma at level 3' (6.76).

Conclusion

- There are distinct patterns in 'Value Added' scores across ethnicities, socio-economic statuses, genders, and qualifications in A-Level and GCSE datasets.
- These findings suggest areas where targeted educational support may be beneficial.

13 Attendance and Academic Performance Analysis

13.1 PAL Attendance Categorization

- Attendance categories were defined based on the proportion of maximum attendance: 'High' (over 50
- For the A-Level dataset, the maximum attendance was 25, and for the GCSE dataset, it was 10.

13.2 Average 'Value Added' by Attendance Category

- A-Level: 'High' attendance correlates with the highest average 'Value Added' of 9.50, followed by 'Medium' (2.99) and 'Low' (1.00).
- GCSE: 'High' attendance shows an average 'Value Added' of 2.73, while 'Medium' attendance is associated with a negative average 'Value Added' (-0.61).

13.3 Analysis of Students Who Failed Diagnostic Tests

- A-Level: High-engagement students who failed the diagnostic test show a significant average 'Value Added' of 19.28, indicating substantial improvement.
- GCSE: The trend is less clear, with 'Low' attendance students who failed the diagnostic test having an average 'Value Added' of 4.09, higher than 'High' attendance students.

13.4 Specific Module Analysis: CS1MCP

- In the GCSE dataset, high-engagement students in the CS1MCP module who failed the diagnostic test achieved an average Final Module Score of approximately 50.00, slightly below the module's total average score of 56.98.
- In the A-Level dataset, there were no students who met these criteria, indicating a data limitation or possibly different student engagement patterns in this module.

13.5 Analysis of New Module: EE1EMA

- In the A-Level dataset, high-engagement students in the EE1EMA module who failed the diagnostic test achieved an average Final Module Score of 56.57, significantly above the module's total average score of 32.14.
- In the GCSE dataset, there were no students who met these criteria, suggesting either a data limitation or distinct engagement patterns in this module.

14 Conclusion

Gender Disparities in Educational Assessments:

In our analysis of gender disparities in educational assessments, we observed significant differences in GCSE diagnostic scores between genders, highlighting potential biases or variations in learning styles and societal influences on academic performance. This disparity raises questions about the equity and inclusivity of the assessment methods used in GCSEs and suggests the need for a reevaluation of these methods to ensure they are fair and effective for all students.

Contrastingly, A-Level assessments did not exhibit these gender disparities, indicating a possible shift in assessment style, teaching methodologies, or student adaptation at this stage. This absence of disparity at the A-Level stage calls for further research into the factors that contribute to the observed differences in GCSEs and how these can be addressed to provide a more equitable educational experience for students at all levels.

Impact of Educational Background:

The impact of educational background on student performance in both A-Level and GCSE assessments was a significant finding of our study. The analysis revealed that students' highest qualification on entry played a pivotal role in their subsequent academic achievements. This observation underscores the lasting influence of early educational experiences and potentially the quality of prior educational institutions on future academic success. Such a correlation suggests that initial educational grounding sets a foundation that continues to affect student performance in later stages of their academic journey.

Furthermore, this finding raises important considerations for educational policy and decision-making. It highlights the need for a strong and equitable foundation in early education to ensure students from diverse backgrounds have equal opportunities for success in higher education. The observed correlation between initial qualifications and later academic performance calls for increased attention and resources to be directed towards improving the quality and accessibility of primary and secondary education. This approach is essential to bridge any gaps that may exist due to disparities in educational background and to create a more level playing field for all students as they progress through their educational careers.

Ethnic Background and Academic Performance:

Our study revealed a complex relationship between ethnic background and academic performance, particularly in the context of GCSE and A-Level assessments. Notably, ethnic background significantly influenced GCSE final module scores, with certain ethnic groups showing varying levels of academic achievement. This finding suggests that there are underlying factors, possibly including cultural, socio-economic, or educational disparities, that affect the academic performance of students from different ethnic backgrounds in GCSEs.

Interestingly, this trend did not persist in A-Level assessments, where ethnicity did not appear to significantly impact final scores. This variation between GCSE and A-Level perfor-

mance may indicate different levels of sensitivity to ethnic background at various stages of education, or it could reflect changes in student composition, teaching methodologies, or assessment criteria. The absence of a significant ethnic influence at the A-Level stage raises questions about the factors that mitigate this influence as students progress in their academic careers.

Socio-Economic Factors:

The influence of socio-economic factors on academic performance was a key area of investigation in our study, specifically in the context of A-Level and GCSE assessments. Contrary to many existing assumptions, our analysis indicated that socio-economic classification did not significantly affect the final module scores in both A-Level and GCSE examinations. This finding challenges the often-held belief that socio-economic background is a major determinant of academic success and suggests that, at least within the scope of our study, other factors may play a more pivotal role in shaping academic outcomes.

The absence of significant differences in academic performance across various socio-economic classes highlights the potential effectiveness of current educational policies and practices in providing equitable opportunities for students from diverse economic backgrounds. However, it also raises questions about the other factors that might be influencing academic performance if not socio-economic status. This could include variables such as school quality, access to educational resources, or individual student characteristics like motivation and learning styles.

Predictive Modelling Insights:

The application of predictive modeling in our study offered valuable insights into the factors influencing academic performance. Utilizing models such as the RandomForestRegressor and Gradient Boosting Classifier, we were able to assess the predictive power of various student characteristics and educational factors. However, the models exhibited varying degrees of accuracy, highlighting the complexity and multi-dimensional nature of academic achievement.

These results demonstrate the potential and limitations of using machine learning models in educational settings. While the models provided some predictive insights, their varying accuracy underscores the challenge of capturing the nuanced and interdependent factors that contribute to academic success. This suggests that while predictive models can be a valuable tool in understanding educational outcomes, they should be used in conjunction with other methods and insights from educational theory and practice.

Furthermore, the findings from our predictive modeling efforts point to the need for ongoing development and refinement of these models. As machine learning and data analytics continue to evolve, there is significant potential for more sophisticated models to provide deeper insights into educational data. This could lead to more personalized and effective educational strategies, helping educators and policymakers to better understand and support the diverse needs of students.

Value-Added Analysis for Individual Performance:

The implementation of value-added analysis in our study provided a novel perspective on assessing individual student performance. This approach, which focuses on measuring student progress rather than just final attainment, revealed insights into the educational journey of each student. By accounting for individual starting points and tracking progress over time, the value-added model offered a more nuanced and equitable method of evaluating academic performance, moving beyond the traditional reliance on raw scores.

This approach highlighted the diversity of student learning paths and the importance of recognizing individual progress. It underscored the need for educational systems to accommodate and nurture varied learning trajectories, acknowledging that each student's academic journey is unique. Value-added analysis thus challenges conventional assessment models that often fail to capture the true educational growth of students, especially those who may start at a disadvantage.

Furthermore, the findings from our value-added analysis suggest the potential for more personalized educational strategies. Tailoring teaching methods and resources to individual student needs could significantly enhance educational outcomes. This approach could lead to more equitable and effective educational practices, ensuring that all students have the opportunity to reach their full potential, irrespective of their starting point. The value-added model, therefore, not only provides a more comprehensive understanding of academic performance but also paves the way for a more inclusive and adaptive educational system.

Impact of PAL Attendance on Academic Outcomes

There is a notable correlation between high PAL (Peer-Assisted Learning) Attendance and increased 'Value Added' scores in both the A-Level and GCSE datasets. This trend suggests that greater engagement in PAL sessions may have a positive impact on academic outcomes. The strength of the relationship between PAL Attendance and academic performance appears to be variable across different modules and datasets. This variability points towards the influence of other factors such as the complexity of the module, teaching methodologies, and the diverse backgrounds of students.

15 Limitations in Current Educational Assessments:

The findings of our study shed light on significant limitations in current educational assessment methods. One key limitation is the apparent lack of sensitivity to diverse learning styles, backgrounds, and needs of students. Our analysis of gender, ethnic, and socio-economic disparities in GCSE and A-Level assessments suggests that current methods may not adequately account for the varied experiences and challenges faced by different student groups. This one-size-fits-all approach can lead to assessments that do not truly reflect the abilities and potential of all students, particularly those from underrepresented or disadvantaged backgrounds.

Moreover, the varying effectiveness of predictive models in forecasting academic performance highlights another limitation in current assessments: the difficulty in capturing the mul-

tifaceted nature of academic achievement. Traditional assessment methods often rely heavily on quantifiable metrics, such as test scores, which may not fully encompass the breadth of skills and knowledge that constitute true academic proficiency. This reliance on narrow criteria for academic success can overlook critical aspects of learning, such as creativity, critical thinking, and problem-solving skills.

These limitations underscore the need for a more holistic approach to educational assessment. This approach should integrate diverse methods that cater to different learning styles and abilities, and that recognize the importance of both quantitative and qualitative aspects of learning. Additionally, there is a need for continuous innovation and adaptation in assessment methods to keep pace with the evolving educational landscape and the diverse needs of the student population. Embracing a more comprehensive and flexible assessment framework is crucial for creating an educational system that is fair, inclusive, and effective in preparing students for the complexities of the modern world.

16 References

Sirin, S.R. (2005). 'Socioeconomic Status and Academic Achievement: A Meta-Analytic Review of Research.' Review of Educational Research, 75(3), pp.417-453. Krein, S.F. and Beller, A.H. (1988).

'Educational Attainment of Children from Single-Parent Families: Differences by Exposure, Gender, and Race.' Demography, 25, pp.221-234. Downey, D.B. (1995). 'When Bigger Is Not Better: Family

Size, Parental Resources, and Children's Educational Performance.' American Sociological Review, 60, pp.746-761. Bogges, S. (1998). 'Family Structure, Economic Status, and Educational Attainment.'

Journal of Population Economics, 11, pp.205-222. Palardy, G.J. (2008). 'Differential School Effects

Among Low, Middle, and High Social Class Composition Schools: A Multi-Group, Multilevel Latent Growth Curve Analysis.' School Effectiveness and School Improvement, 19, pp.21-49. Perry, L.B. and

McConney, A. (2010). 'Does the SES of the School Matter? An Examination of Socioeconomic Status and Student Achievement Using PISA 2003.' Teaching College Record, 112, pp.1137-1162. Organisation for Economic Cooperation and Development (OECD). (2016). 'PISA 2015 Results:

Excellence and Equity in Education.' [Online]. Available at: [URL] (Accessed: [Date]). Wadee, A.A.

and Cliff, A. (2016). 'Pre-admission Tests of Learning Potential as Predictors of Academic Success of First-Year Medical Students.' South African Journal of Higher Education, 30(2), pp.264-278. Williams,

B. and Reddy, P. (2016). 'Does Peer-Assisted Learning Improve Academic Performance? A Scoping Review.' [Journal/Source Name], [Volume(Issue)], pp.[Page numbers]. Rodriguez-Hernandez, C.F.,

Cascallar, E. and Kyndt, E. (2020). 'Socio-economic Status and Academic Performance in Higher Education: A Systematic Review.' Educational Research Review, [Volume(Issue)], pp.[Page numbers]. Education Next. (2016). 'How Family Background Influences Student Achievement.' [Online].

Available at: [URL] (Accessed: [Date]). American Council on Education. (2020). 'Race and Ethnicity

in Higher Education: 2020 Supplement.' Washington, DC: ACE. National Education Association.

(2024). 'What the Research Says About Ethnic Studies.' [Online]. Available at: https://www.nea.org/sites/default/files/2020-10/What

17 Appendix

Load the Excel file

In [1]:
```
import pandas as pd


file_path = 'Copy of MSc Data 1.xlsx'


try:
    xls = pd.ExcelFile(file_path)
    sheet_names = xls.sheet_names
except Exception as e:
    sheet_names = None
    error_message = str(e)

sheet_names, error_message if sheet_names is None else None
```

Out[1]:
```
(['2019_20', '2020_21', '2021_22', '2022_23'], None)
```

Load the specific sheet "2022_23" into a DataFrame

Show the first few rows to get a sense of the data

In [2]:
```
df_2022_23 = pd.read_excel(file_path, sheet_name='2022_23')


df_2022_23.head()
```

Out[2]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature S
0	1	Other Mixed Background	Other	M	A/AS level	A-Level		M
1	2	Asian or Asian British - Bangladeshi	Asian	F	A/AS level	A-Level	Routine occupations	
2	3	Black or Black British - African	Black	M	A/AS level	A-Level	Never worked and long-term unemployed	
3	4	Black or Black British - African	Black	F	A/AS level	A-Level	Semi-routine occupations	
4	5	Asian or Asian British - Bangladeshi	Asian	M	A/AS level	A-Level	Not classified	

In [3]:
```
# Count unique values in the 'Highest Qual on Entry' column
unique_qual_counts = df_2022_23['Socio-economic classification'].value_co
unique_qual_counts
```

Out[3]:
```
Not classified                                   260
Higher managerial and professional occupations   186
Lower managerial and professional occupations    122
Routine occupations                              115
Intermediate occupations                         100
                                                  77
Small employers and own account workers           76
Semi-routine occupations                          56
Lower supervisory and technical occupations       47
Never worked and long-term unemployed             26
Name: Socio-economic classification, dtype: int64
```

In [ ]:
```
```

Analyzing the 'Ethnicity' column to get the counts of each unique ethnic group

In [4]:
```
ethnicity_counts = df_2022_23['Ethnicity'].value_counts().reset_index()
ethnicity_counts.columns = ['Ethnicity', 'Count']

ethnicity_counts
```

Out[4]:

	Ethnicity	Count
0	Asian or Asian British - Pakistani	211
1	Black or Black British - African	176
2	White	176
3	Asian or Asian British - Indian	149
4	Asian or Asian British - Bangladeshi	93
5	Arab	79
6	Asian Other	46
7	White - British	36
8	Black or Black British - Caribbean	16
9	Prefer not to say	13
10	Other Mixed Background	13
11	Mixed - White & Asian	13
12	Other Ethnic Background	11
13	Mixed - White & Black Caribbean	10
14	Chinese	10
15	Other Black Background	7
16	Mixed - White & Black African	6

In [5]:

```
# Define custom categories based on user input
custom_categories = {
    'Other Asian or Bangladeshi': ['Asian or Asian British - Bangladeshi'
    'White': ['White', 'White - British'],
    'Black': ['Black or Black British - African', 'Black or Black British
    'Indian': ['Asian or Asian British - Indian'],
    'Pakistani': ['Asian or Asian British - Pakistani'],
    'Other Mixed': ['Other Mixed Background', 'Mixed - White & Asian', 'M
}

for custom_category, ethnicities in custom_categories.items():
    df_2022_23['Ethnicity'].replace(ethnicities, custom_category, inplace
df_2022_23.head()
```

Out[5]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature S
0	1	Other Mixed	Other	M	A/AS level	A-Level		M
1	2	Other Asian or Bangladeshi	Asian	F	A/AS level	A-Level	Routine occupations	
2	3	Black	Black	M	A/AS level	A-Level	Never worked and long-term unemployed	
3	4	Black	Black	F	A/AS level	A-Level	Semi-routine occupations	
4	5	Other Asian or Bangladeshi	Asian	M	A/AS level	A-Level	Not classified	

In [6]:
```
unique_Ethnicity = df_2022_23['Ethnicity'].unique()
unique_Ethnicity
```

Out[6]:
```
array(['Other Mixed', 'Other Asian or Bangladeshi', 'Black', 'Indian',
       'White', 'Pakistani'], dtype=object)
```

In [7]:
```
#'Mathematics Requirement' column
math_req_counts = df_2022_23['Mathematics Requirement'].value_counts().re
math_req_counts.columns = ['Mathematics Requirement', 'Count']
math_req_counts
```

Out[7]:

	Mathematics Requirement	Count
0	A-Level	545
1	GCSE	520

In [8]:
```
highest_qual_counts = df_2022_23['Highest Qual on Entry'].value_counts()
highest_qual_counts
```

Out[8]:
```
Other qualification at level 2                  395
A/AS level                                      256
Level 3 quals (all are in UCAS tariff)          130
Diploma at level 3                               74
A-Level                                          34
Other qualification level not known              24
Level 3 quals (none are in UCAS Tariff)          17
Not known                                        13
Certificate at level 3                            6
HE access course, QAA recognised                  5
Level 3 quals (some are in UCAS tariff)           3
Foundation degree                                 2
International Baccalaureate (IB) Diploma          2
UK first degree with honours                      1
Student has no formal qualification               1
International Baccalaureate (IB) Certificate      1
Higher National Diploma (HND)                     1
HE access course, not QAA recognised              1
Higher National Certificate (HNC)                 1
Name: Highest Qual on Entry, dtype: int64
```

In [9]:
```
# Conditionally update 'Highest Qual on Entry' column
mask = (df_2022_23['Mathematics Requirement'] == 'A-Level') & (df_2022_23
df_2022_23.loc[mask, 'Highest Qual on Entry'] = 'A-Level'
```

In [10]:
```
# Count unique values in the 'Highest Qual on Entry' column
unique_qual_counts = df_2022_23['Highest Qual on Entry'].value_counts()
unique_qual_counts
```

Out[10]:
```
A/AS level                                      256
Other qualification at level 2                  235
A-Level                                         194
Level 3 quals (all are in UCAS tariff)          130
Diploma at level 3                               74
Other qualification level not known              24
Level 3 quals (none are in UCAS Tariff)          17
Not known                                        13
Certificate at level 3                            6
HE access course, QAA recognised                  5
Level 3 quals (some are in UCAS tariff)           3
Foundation degree                                 2
International Baccalaureate (IB) Diploma          2
UK first degree with honours                      1
Student has no formal qualification               1
International Baccalaureate (IB) Certificate      1
Higher National Diploma (HND)                     1
HE access course, not QAA recognised              1
Higher National Certificate (HNC)                 1
Name: Highest Qual on Entry, dtype: int64
```

In [11]:
```
# Conditionally update 'Highest Qual on Entry' column
mask = (df_2022_23['Mathematics Requirement'] == 'GCSE') & (df_2022_23['H
df_2022_23.loc[mask, 'Highest Qual on Entry'] = 'Level 3 qual'
```

In [12]:
```
# Count unique values in the 'Highest Qual on Entry' column
unique_qual_counts = df_2022_23['Highest Qual on Entry'].value_counts()
unique_qual_counts
```

Out[12]:
```
A/AS level                                        256
Level 3 qual                                      235
A-Level                                           194
Level 3 quals (all are in UCAS tariff)            130
Diploma at level 3                                 74
Other qualification level not known                24
Level 3 quals (none are in UCAS Tariff)            17
Not known                                          13
Certificate at level 3                              6
HE access course, QAA recognised                    5
Level 3 quals (some are in UCAS tariff)             3
Foundation degree                                   2
International Baccalaureate (IB) Diploma            2
UK first degree with honours                        1
Student has no formal qualification                 1
International Baccalaureate (IB) Certificate        1
Higher National Diploma (HND)                       1
HE access course, not QAA recognised                1
Higher National Certificate (HNC)                   1
Name: Highest Qual on Entry, dtype: int64
```

In [13]:
```
# Create separate DataFrames based on 'Mathematics Requirement'
df_a_level = df_2022_23[df_2022_23['Mathematics Requirement'] == 'A-Level
df_gcse = df_2022_23[df_2022_23['Mathematics Requirement'] == 'GCSE']
```

In [14]:
```
df_a_level
```

Out[14]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	N
0	1	Other Mixed	Other	M	A/AS level	A-Level		
1	2	Other Asian or Bangladeshi	Asian	F	A/AS level	A-Level	Routine occupations	
2	3	Black	Black	M	A/AS level	A-Level	Never worked and long-term unemployed	
3	4	Black	Black	F	A/AS level	A-Level	Semi-routine occupations	
4	5	Other Asian or Bangladeshi	Asian	M	A/AS level	A-Level	Not classified	
...	...	...	...	...	...	...	...	
1060	1061	Black	Black	M	Foundation degree	Other	Lower managerial and professional occupations	
1061	1062	Pakistani	Asian	F	Level 3 quals (all are in UCAS tariff)	Level 3 Quals	Not classified	
1062	1063	Black	Black	M	A-Level	Level 2 Quals	Intermediate occupations	
1063	1064	Other Mixed	Other	M	A/AS level	A-Level	Not classified	
1064	1065	Black	Black	F	A-Level	Level 2 Quals	Intermediate occupations	

545 rows × 15 columns

In [15]:
```
df_gcse
```

Out[15]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature S
162	163	Pakistani	Asian	M	A/AS level	A-Level		M
163	164	Black	Black	M	A/AS level	A-Level	Not classified	
164	165	Pakistani	Asian	F	Level 3 quals (all are in UCAS tariff)	Level 3 Quals	Not classified	
165	166	Black	Black	F	A/AS level	A-Level	Routine occupations	
166	167	White	White	M	Level 3 qual	Level 2 Quals	Small employers and own account workers	
...	...	...	...	...	...	...	...	
677	678	White	White	M	Level 3 quals (all are in UCAS tariff)	Level 3 Quals	Higher managerial and professional occupations	
678	679	Pakistani	Asian	F	Level 3 qual	Level 2 Quals	Not classified	
679	680	Other Mixed	Other	M	Level 3 qual	Level 2 Quals	Higher managerial and professional occupations	
680	681	White	White	M	Level 3 qual	Level 2 Quals	Not classified	
681	682	Indian	Asian	M	Level 3 quals (none are in UCAS Tariff)	Level 3 Quals	Higher managerial and professional occupations	

520 rows × 15 columns

In [16]:

```
# Count NaN values in the df_a_level DataFrame
nan_counts_df_gcse = df_gcse.isna().sum()

# Display the count of NaN values for each column
nan_counts_df_gcse
```

Out[16]:
```
Student ID                          0
Ethnicity                           0
Ethnicity Summary                   0
Gender                              0
Highest Qual on Entry              63
Qualification Summary               0
Socio-economic classification       0
Mature/Young Student                0
Module Code                         0
Mathematics Requirement             0
Disability                          2
Disability Summary                  0
Diagnostic Score                  211
PAL Attendance                      0
Final Module Score                 32
dtype: int64
```

In [17]:
```
# Count NaN values in the df_a_level DataFrame
nan_counts_a_level = df_a_level.isna().sum()

# Display the count of NaN values for each column
nan_counts_a_level
```

Out[17]:
```
Student ID                          0
Ethnicity                           0
Ethnicity Summary                   0
Gender                              0
Highest Qual on Entry              35
Qualification Summary               0
Socio-economic classification       0
Mature/Young Student                0
Module Code                         0
Mathematics Requirement             0
Disability                          0
Disability Summary                  0
Diagnostic Score                  333
PAL Attendance                      0
Final Module Score                  9
dtype: int64
```

In [18]:
```
# Show rows with null values in "Final Module Score" column
null_final_module_score = df_a_level[df_a_level['Final Module Score'].isn
null_final_module_score[['Student ID', 'Final Module Score']]
```

Out[18]:

	Student ID	Final Module Score
45	46	NaN
696	697	NaN
709	710	NaN
736	737	NaN
752	753	NaN
795	796	NaN
814	815	NaN
898	899	NaN
1006	1007	NaN

In [19]:
```
# Get unique values in the 'Diagnostic Score' column
unique_diagnostic_scores = df_a_level['Diagnostic Score'].unique()

# Display the unique values
unique_diagnostic_scores
```

Out[19]:
```
array([ nan,  9. , 10. ,  8. ,  7. ,  3. ,  1. ,  9.5,  4.5,  5.5,  4. ,
        5. ,  8.5,  6. ,  7.5,  3.5,  6.5,  2. ,  0. ,  1.5,  2.5,  0.5])
```

In [20]:
```
# Count unique values in the 'Diagnostic Score' column
count_unique_diagnostic_scores = df_a_level['Diagnostic Score'].nunique()

# Display the count of unique values
count_unique_diagnostic_scores
```

Out[20]:
```
21
```

In [21]:
```
# Remove rows with null "Final Module Score" in df_a_level
df_a_level = df_a_level.dropna(subset=['Final Module Score'])

# Remove rows with null "Final Module Score" in df_gcse
df_gcse = df_gcse.dropna(subset=['Final Module Score'])
```

In [22]:
```
df_a_level
```

Out[22]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	N
0	1	Other Mixed	Other	M	A/AS level	A-Level		
1	2	Other Asian or Bangladeshi	Asian	F	A/AS level	A-Level	Routine occupations	
2	3	Black	Black	M	A/AS level	A-Level	Never worked and long-term unemployed	
3	4	Black	Black	F	A/AS level	A-Level	Semi-routine occupations	
4	5	Other Asian or Bangladeshi	Asian	M	A/AS level	A-Level	Not classified	
...	...	...	...	...	...	...	...	
1060	1061	Black	Black	M	Foundation degree	Other	Lower managerial and professional occupations	
1061	1062	Pakistani	Asian	F	Level 3 quals (all are in UCAS tariff)	Level 3 Quals	Not classified	
1062	1063	Black	Black	M	A-Level	Level 2 Quals	Intermediate occupations	
1063	1064	Other Mixed	Other	M	A/AS level	A-Level	Not classified	
1064	1065	Black	Black	F	A-Level	Level 2 Quals	Intermediate occupations	

536 rows × 15 columns

In [23]:
```
df_gcse
```

Out[23]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature S
162	163	Pakistani	Asian	M	A/AS level	A-Level		M
163	164	Black	Black	M	A/AS level	A-Level	Not classified	
164	165	Pakistani	Asian	F	Level 3 quals (all are in UCAS tariff)	Level 3 Quals	Not classified	
165	166	Black	Black	F	A/AS level	A-Level	Routine occupations	
166	167	White	White	M	Level 3 qual	Level 2 Quals	Small employers and own account workers	
...	...	...	...	...	...	...	...	
677	678	White	White	M	Level 3 quals (all are in UCAS tariff)	Level 3 Quals	Higher managerial and professional occupations	
678	679	Pakistani	Asian	F	Level 3 qual	Level 2 Quals	Not classified	
679	680	Other Mixed	Other	M	Level 3 qual	Level 2 Quals	Higher managerial and professional occupations	
680	681	White	White	M	Level 3 qual	Level 2 Quals	Not classified	
681	682	Indian	Asian	M	Level 3 quals (none are in UCAS Tariff)	Level 3 Quals	Higher managerial and professional occupations	

488 rows × 15 columns

In [24]:
```
# Finding the rows where the "Diagnostic Score" column has null values
df_null_diagnostic_score = df_a_level[df_a_level['Diagnostic Score'].isnu
```

In [25]:
```
# Displaying the rows with null "Diagnostic Score"
df_null_diagnostic_score.head(), len(df_null_diagnostic_score)
```

Out[25]:
```
(   Student ID                  Ethnicity Ethnicity Summary Gender  \
 0           1                Other Mixed             Other      M
 1           2  Other Asian or Bangladeshi             Asian      F
 2           3                      Black             Black      M
 3           4                      Black             Black      F
 5           6                      Black             Black      F

   Highest Qual on Entry Qualification Summary  \
 0            A/AS level               A-Level
 1            A/AS level               A-Level
 2            A/AS level               A-Level
 3            A/AS level               A-Level
 5            A/AS level               A-Level

            Socio-economic classification Mature/Young Student Module Cod
e  \
 0                                                      MATURE      AM10F
M
 1                   Routine occupations                 YOUNG      AM10F
M
 2  Never worked and long-term unemployed                 YOUNG      AM10F
M
 3               Semi-routine occupations                 YOUNG      AM10F
M
 5               Intermediate occupations                 YOUNG      AM10F
M

   Mathematics Requirement                                          Disabi
lity  \
 0                A-Level  Social or communication condition such as asp
e...
 1                A-Level                                   No disabi
lity
 2                A-Level                                   No disabi
lity
 3                A-Level                                   No disabi
lity
 5                A-Level                                   No disabi
lity

            Disability Summary  Diagnostic Score  PAL Attendance  \
 0  Social/Learning disability               NaN               0
 1               No disability               NaN               0
 2               No disability               NaN               0
 3               No disability               NaN               1
 5               No disability               NaN               3

    Final Module Score
 0               55.98
 1               67.29
 2               53.30
 3               53.70
 5               48.67  ,
 329)
```

In [26]:
```
# Checking the correlation of all numeric variables with 'Diagnostic Scor
correlation_matrix = df_a_level.corr()
correlation_with_diagnostic_score = correlation_matrix['Diagnostic Score'

correlation_with_diagnostic_score
```

Out[26]:
```
Diagnostic Score      1.000000
Final Module Score    0.256561
PAL Attendance       -0.021369
Student ID           -0.335559
Name: Diagnostic Score, dtype: float64
```

In [27]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import numpy as np
```

```
/Users/pawan/opt/anaconda3/lib/python3.9/site-packages/scipy/__init__.py:
146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for th
is version of SciPy (detected version 1.25.2
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [28]:
```
# Drop rows where 'Diagnostic Score' as well as 'Final Module Score' and
df_train = df_a_level.dropna(subset=['Diagnostic Score']).drop(columns=['

# Prepare the data where 'Diagnostic Score' is NaN for prediction
df_predict = df_a_level[df_a_level['Diagnostic Score'].isnull()].drop(col

# Separate features (X) and target variable (y) for training set
X = df_train.drop(columns=['Diagnostic Score'])
y = df_train['Diagnostic Score']

# Convert categorical variables into numerical representations (one-hot e
X = pd.get_dummies(X, drop_first=True)
df_predict = pd.get_dummies(df_predict, drop_first=True)

# Make sure both datasets have the same columns (important because of dum
missing_cols = set(X.columns) - set(df_predict.columns)
for col in missing_cols:
    df_predict[col] = 0
df_predict = df_predict[X.columns]

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,

# Create and fit the Random Forest model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Evaluate the model on the test set
y_pred = rf_model.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

# Use the trained model to predict the missing 'Diagnostic Score' values
predicted_diagnostic_scores = rf_model.predict(df_predict)

rmse, predicted_diagnostic_scores[:10]  # Show RMSE and first 10 predicte
```

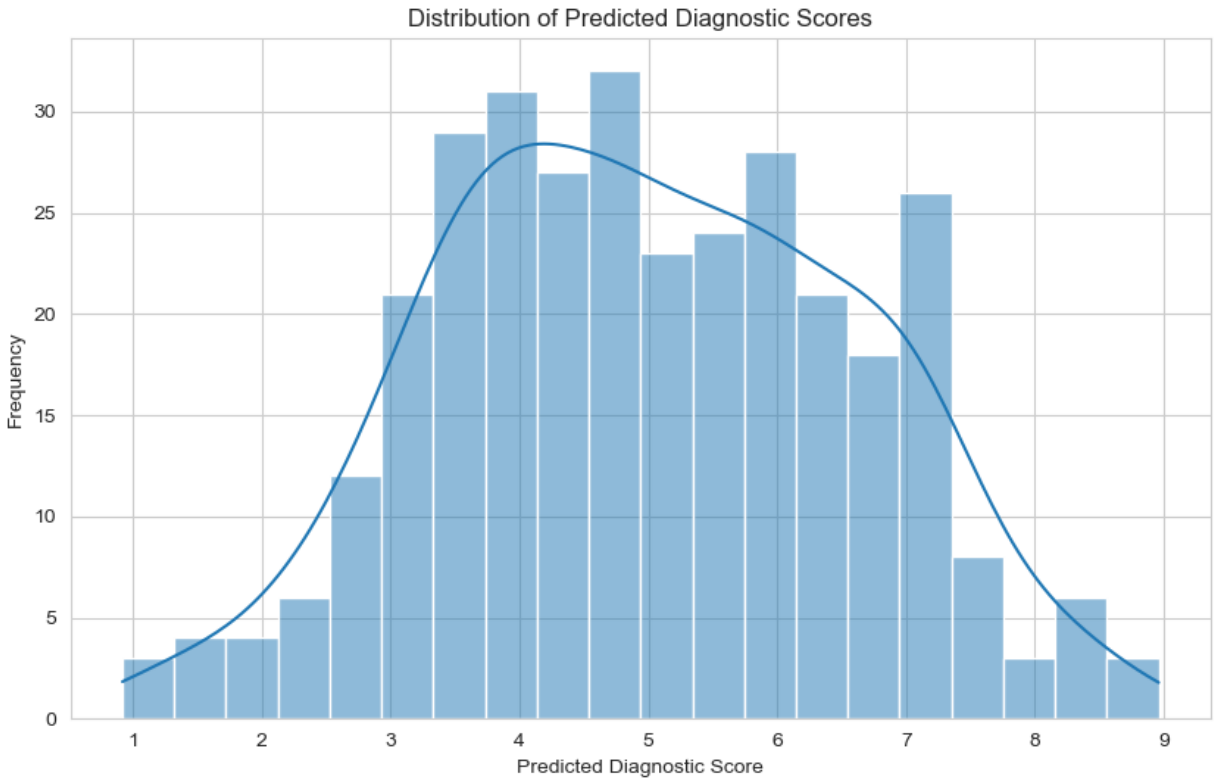
Out[28]:
```
(3.464937297584255,
 array([6.84 , 8.23 , 8.69 , 7.905, 4.215, 6.485, 8.21 , 8.195, 8.265,
        6.99 ]))
```

In [29]:

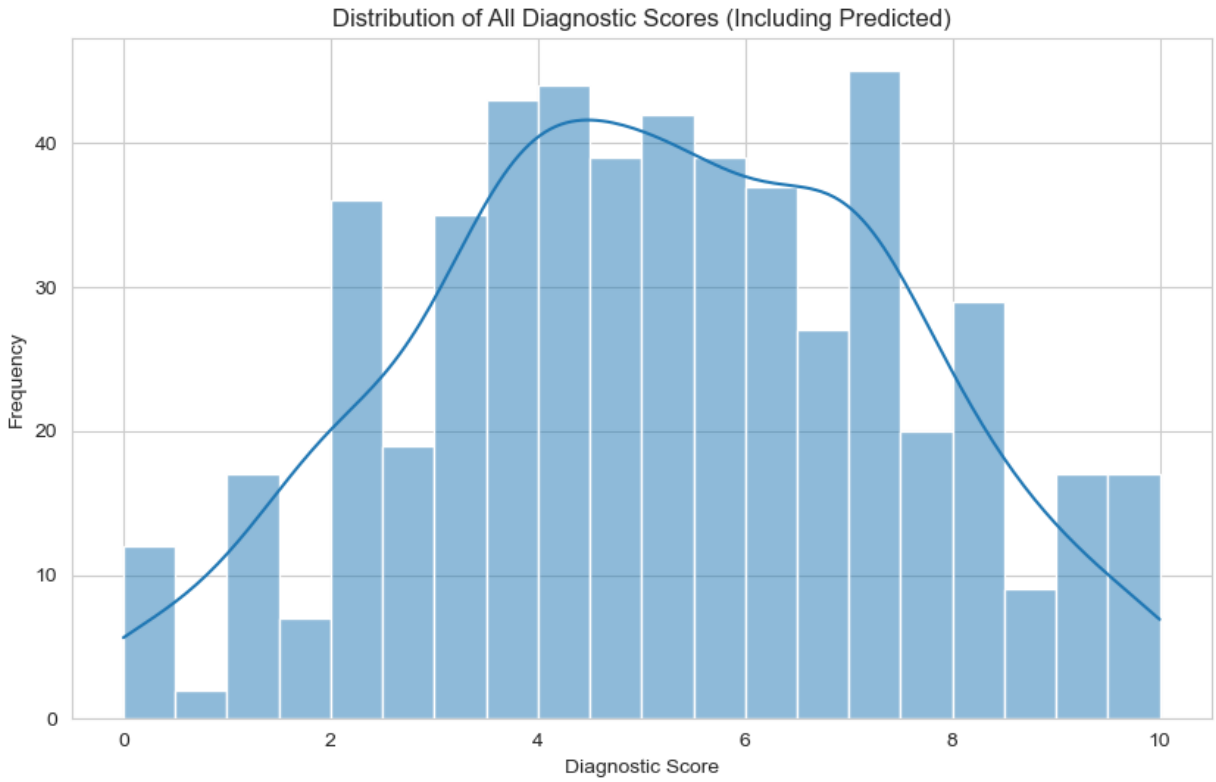
```
# Update the original DataFrame with the predicted 'Diagnostic Score' val
df_a_level.loc[df_a_level['Diagnostic Score'].isnull(), 'Diagnostic Score

# Check the DataFrame to see if the null values in 'Diagnostic Score' hav
df_a_level[df_a_level['Diagnostic Score'].isnull()], df_a_level.head()
```

Out[29]:
```
(Empty DataFrame
 Columns: [Student ID, Ethnicity, Ethnicity Summary, Gender, Highest Qual
on Entry, Qualification Summary, Socio-economic classification, Mature/Yo
ung Student, Module Code, Mathematics Requirement, Disability, Disability
Summary, Diagnostic Score, PAL Attendance, Final Module Score]
 Index: [],
   Student ID                  Ethnicity Ethnicity Summary Gender  \
0           1                Other Mixed             Other      M
1           2  Other Asian or Bangladeshi             Asian      F
2           3                      Black             Black      M
3           4                      Black             Black      F
4           5  Other Asian or Bangladeshi             Asian      M

  Highest Qual on Entry Qualification Summary  \
0            A/AS level               A-Level
1            A/AS level               A-Level
2            A/AS level               A-Level
3            A/AS level               A-Level
4            A/AS level               A-Level

           Socio-economic classification Mature/Young Student Module Cod
e  \
0                                                    MATURE      AM10F
M
1                    Routine occupations                YOUNG      AM10F
M
2  Never worked and long-term unemployed                YOUNG      AM10F
M
3               Semi-routine occupations                YOUNG      AM10F
M
4                         Not classified                YOUNG      AM10F
M

   Mathematics Requirement                                        Disabi
lity  \
0                A-Level  Social or communication condition such as asp
e...
1                A-Level                                     No disabi
lity
2                A-Level                                     No disabi
lity
3                A-Level                                     No disabi
lity
4                A-Level                                     No disabi
lity

           Disability Summary  Diagnostic Score  PAL Attendance  \
0  Social/Learning disability             6.840               0
1               No disability             8.230               0
2               No disability             8.690               0
3               No disability             7.905               1
4               No disability             9.000               0

   Final Module Score
0               55.98
1               67.29
2               53.30
3               53.70
4               54.30  )
```

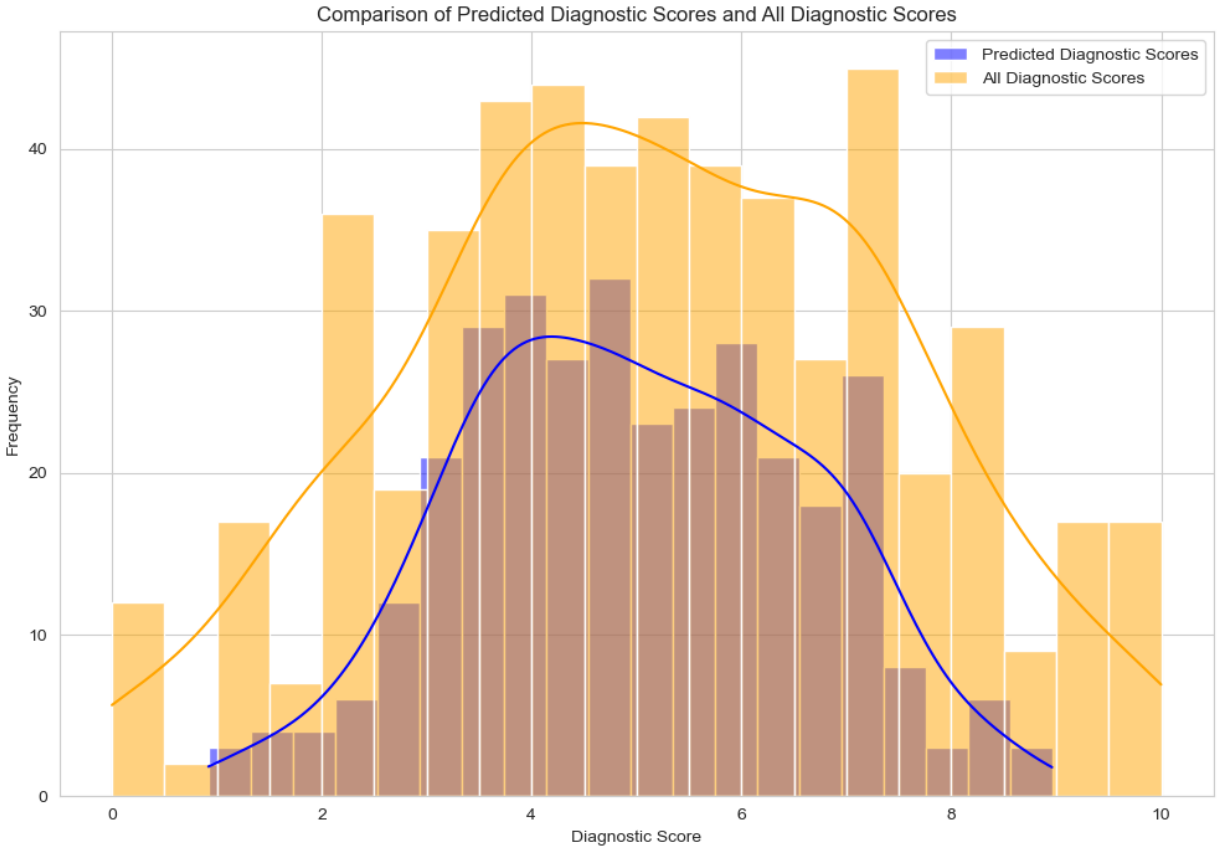
In [30]:
```
import matplotlib.pyplot as plt
import seaborn as sns
df_predicted_values = pd.DataFrame({'Diagnostic Score': predicted_diagnos

# Set the style for the visualizations
sns.set_style("whitegrid")

# Create a histogram for the predicted 'Diagnostic Score' values
plt.figure(figsize=(10, 6))
sns.histplot(df_predicted_values['Diagnostic Score'], bins=20, kde=True)
plt.title('Distribution of Predicted Diagnostic Scores')
plt.xlabel('Predicted Diagnostic Score')
plt.ylabel('Frequency')
plt.show()
```



In [31]:
```
# Create a histogram for all 'Diagnostic Score' values in the updated Dat
plt.figure(figsize=(10, 6))
sns.histplot(df_a_level['Diagnostic Score'], bins=20, kde=True)
plt.title('Distribution of All Diagnostic Scores (Including Predicted)')
plt.xlabel('Diagnostic Score')
plt.ylabel('Frequency')
plt.show()
```

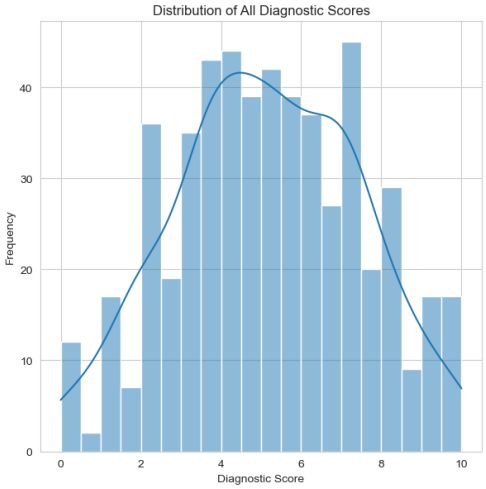


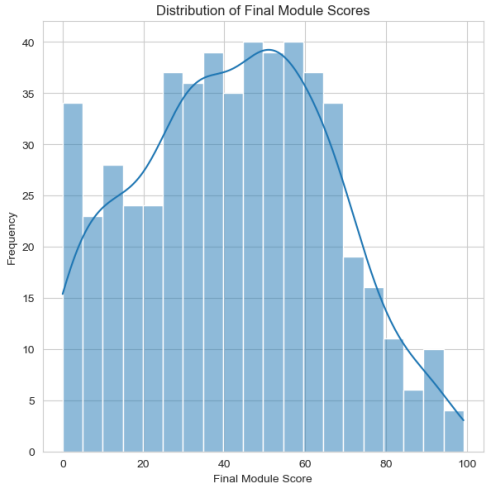
In [32]:
```
# Creating a combined histogram for both the predicted Diagnostic Scores

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

# Histogram for the predicted 'Diagnostic Score' values
sns.histplot(df_predicted_values['Diagnostic Score'], bins=20, kde=True,

# Histogram for all 'Diagnostic Score' values in the updated DataFrame
sns.histplot(df_a_level['Diagnostic Score'], bins=20, kde=True, color='or

plt.title('Comparison of Predicted Diagnostic Scores and All Diagnostic S
plt.xlabel('Diagnostic Score')
plt.ylabel('Frequency')
plt.legend()

plt.show()
```

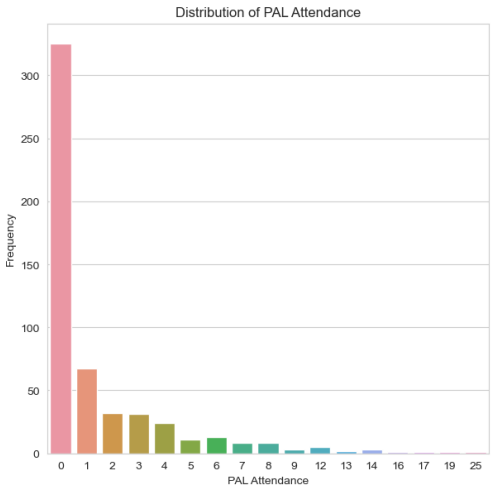

Comparison of Predicted Diagnostic Scores and All Diagnostic Scores
Predicted Diagnostic Scores
All Diagnostic Scores
Frequency
Diagnostic Score
0
10
20
30
40
0
2
4
6
8
10

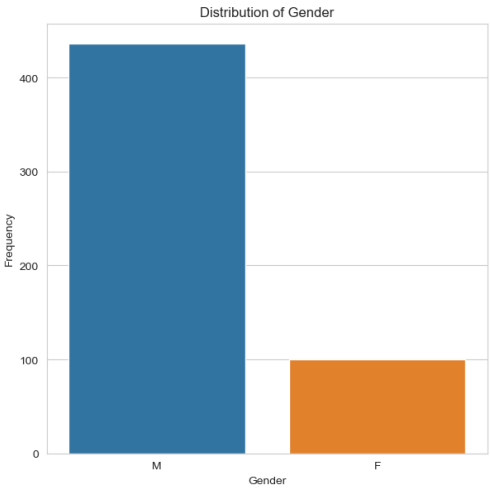
In [33]:

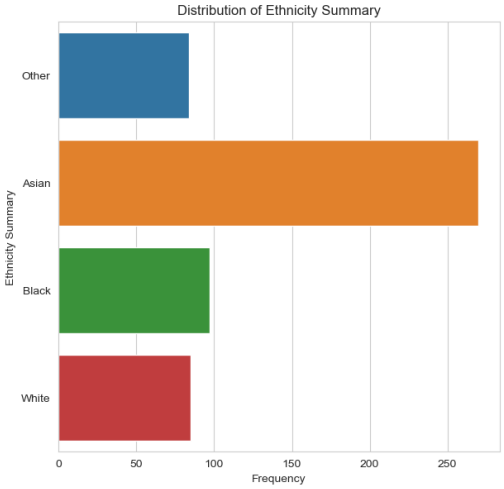
```
# Create visualizations to explore various aspects of the whole dataset

# Set up the subplots
fig, axes = plt.subplots(3, 2, figsize=(15, 18))

# Plot 1: Distribution of Diagnostic Scores for the whole dataset
sns.histplot(df_a_level['Diagnostic Score'], bins=20, kde=True, ax=axes[0
axes[0, 0].set_title('Distribution of All Diagnostic Scores')
axes[0, 0].set_xlabel('Diagnostic Score')
axes[0, 0].set_ylabel('Frequency')

# Plot 2: Distribution of Final Module Scores
sns.histplot(df_a_level['Final Module Score'], bins=20, kde=True, ax=axes
axes[0, 1].set_title('Distribution of Final Module Scores')
axes[0, 1].set_xlabel('Final Module Score')
axes[0, 1].set_ylabel('Frequency')

# Plot 3: Distribution of PAL Attendance
sns.countplot(data=df_a_level, x='PAL Attendance', ax=axes[1, 0])
axes[1, 0].set_title('Distribution of PAL Attendance')
axes[1, 0].set_xlabel('PAL Attendance')
axes[1, 0].set_ylabel('Frequency')

# Plot 4: Distribution of Gender
sns.countplot(data=df_a_level, x='Gender', ax=axes[1, 1])
axes[1, 1].set_title('Distribution of Gender')
axes[1, 1].set_xlabel('Gender')
axes[1, 1].set_ylabel('Frequency')

# Plot 5: Distribution of Ethnicity Summary
sns.countplot(data=df_a_level, y='Ethnicity Summary', ax=axes[2, 0])
axes[2, 0].set_title('Distribution of Ethnicity Summary')
axes[2, 0].set_xlabel('Frequency')
axes[2, 0].set_ylabel('Ethnicity Summary')

# Plot 6: Distribution of Socio-economic classification
sns.countplot(data=df_a_level, y='Socio-economic classification', ax=axes
axes[2, 1].set_title('Distribution of Socio-economic Classification')
axes[2, 1].set_xlabel('Frequency')
axes[2, 1].set_ylabel('Socio-economic Classification')

# Adjust layout
plt.tight_layout()

plt.show()
```

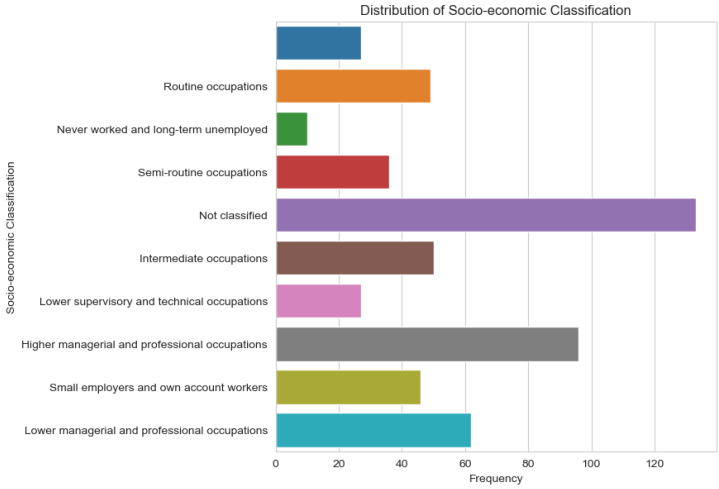

Distribution of All Diagnostic Scores
Frequency
Diagnostic Score


Distribution of Final Module Scores
Frequency
Final Module Score


Distribution of PAL Attendance
Frequency
PAL Attendance


Distribution of Gender
Frequency
M
F
Gender


Distribution of Ethnicity Summary
Ethnicity Summary
Other
Asian
Black
White
Frequency


Distribution of Socio-economic Classification
Socio-economic Classification
Routine occupations
Never worked and long-term unemployed
Semi-routine occupations
Not classified
Intermediate occupations
Lower supervisory and technical occupations
Higher managerial and professional occupations
Small employers and own account workers
Lower managerial and professional occupations
Frequency

In [34]:
```
# Drop rows where 'Diagnostic Score' as well as 'Final Module Score' and
df_gcse_train = df_gcse.dropna(subset=['Diagnostic Score']).drop(columns=

# Prepare the data where 'Diagnostic Score' is NaN for prediction in GCSE
df_gcse_predict = df_gcse[df_gcse['Diagnostic Score'].isnull()].drop(colu

# Separate features (X_gcse) and target variable (y_gcse) for GCSE traini
X_gcse = df_gcse_train.drop(columns=['Diagnostic Score'])
y_gcse = df_gcse_train['Diagnostic Score']

# Convert categorical variables into numerical representations (one-hot e
X_gcse = pd.get_dummies(X_gcse, drop_first=True)
df_gcse_predict = pd.get_dummies(df_gcse_predict, drop_first=True)

# Make sure both GCSE datasets have the same columns (important because o
missing_cols_gcse = set(X_gcse.columns) - set(df_gcse_predict.columns)
for col in missing_cols_gcse:
    df_gcse_predict[col] = 0
df_gcse_predict = df_gcse_predict[X_gcse.columns]

# Split the data into training and test sets for GCSE dataset
X_gcse_train, X_gcse_test, y_gcse_train, y_gcse_test = train_test_split(X

# Create and fit the Random Forest model for GCSE dataset
rf_model_gcse = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model_gcse.fit(X_gcse_train, y_gcse_train)

# Evaluate the model on the test set for GCSE dataset
y_gcse_pred = rf_model_gcse.predict(X_gcse_test)
rmse_gcse = np.sqrt(mean_squared_error(y_gcse_test, y_gcse_pred))

# Use the trained model to predict the missing 'Diagnostic Score' values
predicted_diagnostic_scores_gcse = rf_model_gcse.predict(df_gcse_predict)

rmse_gcse, predicted_diagnostic_scores_gcse[:10]  # Show RMSE and first 1
```

Out[34]:
```
(3.7211514754940342,
 array([ 8.42 ,  9.33 ,  7.955,  7.905, 11.15 ,  7.385,  7.61 , 10.895,
         8.805,  8.495]))
```

In [35]:
```
# Update the original GCSE DataFrame with the predicted 'Diagnostic Score
df_gcse.loc[df_gcse['Diagnostic Score'].isnull(), 'Diagnostic Score'] = p

# Check the GCSE DataFrame to see if the null values in 'Diagnostic Score
df_gcse_filled_values = df_gcse.loc[df_gcse['Diagnostic Score'].isnull()]
df_gcse_filled_values, df_gcse.head()
```

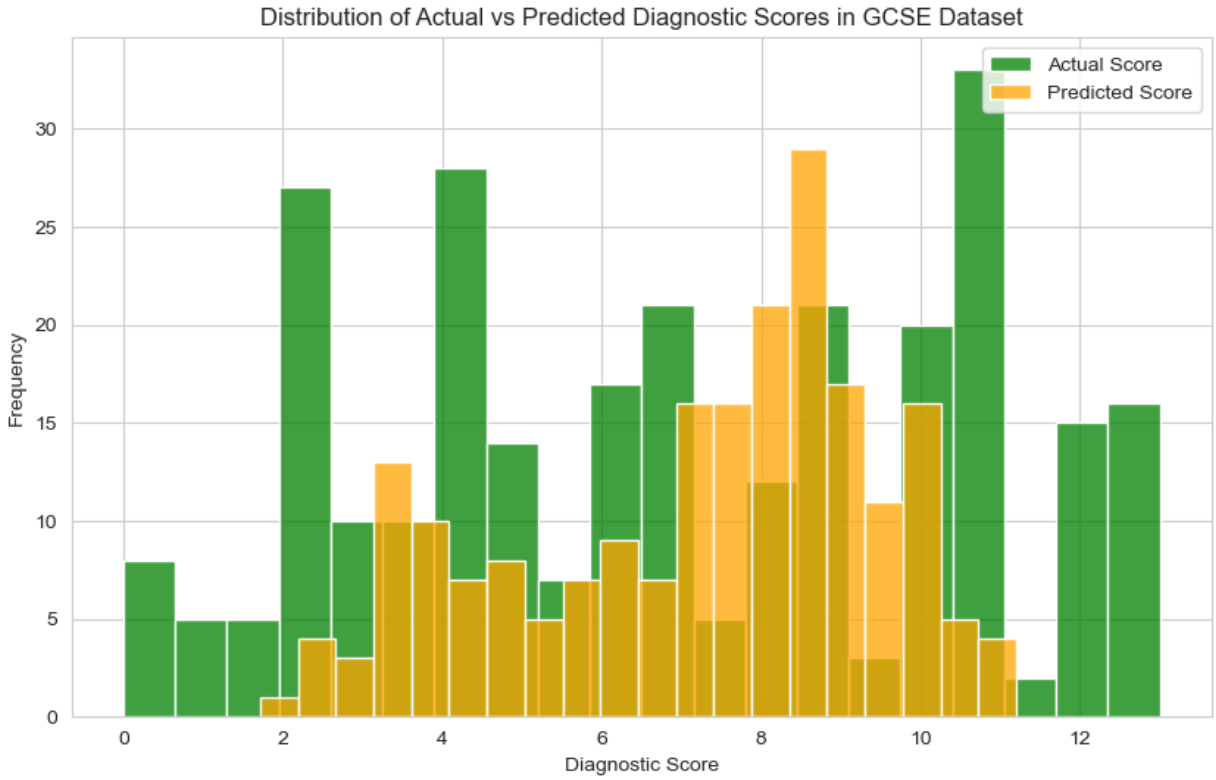
Out[35]:
```
(Empty DataFrame
 Columns: [Student ID, Ethnicity, Ethnicity Summary, Gender, Highest Qual
on Entry, Qualification Summary, Socio-economic classification, Mature/Yo
ung Student, Module Code, Mathematics Requirement, Disability, Disability
Summary, Diagnostic Score, PAL Attendance, Final Module Score]
 Index: [],
     Student ID  Ethnicity Ethnicity Summary Gender  \
 162         163  Pakistani             Asian      M
 163         164      Black             Black      M
 164         165  Pakistani             Asian      F
 165         166      Black             Black      F
 166         167      White             White      M

                      Highest Qual on Entry Qualification Summary  \
 162                              A/AS level               A-Level
 163                              A/AS level               A-Level
 164  Level 3 quals (all are in UCAS tariff)         Level 3 Quals
 165                              A/AS level               A-Level
 166                            Level 3 qual         Level 2 Quals

                Socio-economic classification Mature/Young Student Module
Code  \
 162                                                        MATURE      C
H1MAT
 163                            Not classified               YOUNG      C
H1MAT
 164                            Not classified               YOUNG      C
H1MAT
 165                      Routine occupations                YOUNG      C
H1MAT
 166  Small employers and own account workers                YOUNG      C
H1MAT

     Mathematics Requirement     Disability Disability Summary  \
 162                    GCSE  No disability      No disability
 163                    GCSE  No disability      No disability
 164                    GCSE  No disability      No disability
 165                    GCSE  No disability      No disability
 166                    GCSE  No disability      No disability

      Diagnostic Score  PAL Attendance  Final Module Score
 162             8.420               0                8.50
 163             9.330               0               46.00
 164             7.955               0               20.00
 165             7.905               0                2.00
 166            12.000               4               42.54  )
```

In [36]:
```python
# Double-check that the 'Score Type' column is correctly labeled
# Initially, set all as 'Actual' and then update the predicted ones
df_gcse['Score Type'] = 'Actual'
df_gcse.loc[df_gcse_predict.index, 'Score Type'] = 'Predicted'

# Verify that the 'Score Type' column has been updated correctly
df_gcse['Score Type'].value_counts()
```

```
/var/folders/jh/fgdcbdr1491cv5wb98g38zfc0000gn/T/ipykernel_54125/3172107049.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_gcse['Score Type'] = 'Actual'
```

Out[36]:
```
Actual       279
Predicted    209
Name: Score Type, dtype: int64
```

In [37]:
```
# Create a custom legend to label the actual and predicted 'Diagnostic Sc
plt.figure(figsize=(10, 6))

# Create separate histograms for 'Actual' and 'Predicted' values to manua
sns.histplot(df_gcse[df_gcse['Score Type'] == 'Actual']['Diagnostic Score
             bins=20, kde=False, color='green', label='Actual Score')
sns.histplot(df_gcse[df_gcse['Score Type'] == 'Predicted']['Diagnostic Sc
             bins=20, kde=False, color='orange', label='Predicted Score')

plt.title('Distribution of Actual vs Predicted Diagnostic Scores in GCSE
plt.xlabel('Diagnostic Score')
plt.ylabel('Frequency')

# Add the custom legend
plt.legend(loc='upper right')

plt.show()
```



2nd week's work

A Level

In [38]:
```
# Count the number of occurrences of each gender within each ethnicity
gender_ethnicity_count = df_a_level.groupby(['Ethnicity', 'Gender']).size

# Display the result
gender_ethnicity_count
```

Out[38]:

	Ethnicity	Gender	Count
0	Black	F	22
1	Black	M	75
2	Indian	F	16
3	Indian	M	55
4	Other Asian or Bangladeshi	F	14
5	Other Asian or Bangladeshi	M	65
6	Other Mixed	F	22
7	Other Mixed	M	62
8	Pakistani	F	19
9	Pakistani	M	101
10	White	F	7
11	White	M	78

GCSE

In [39]:
```
# Count the number of occurrences of each gender within each ethnicity
gender_ethnicity_count = df_gcse.groupby(['Ethnicity', 'Gender']).size().

# Display the result
gender_ethnicity_count
```

Out[39]:

	Ethnicity	Gender	Count
0	Black	F	27
1	Black	M	70
2	Indian	F	17
3	Indian	M	61
4	Other Asian or Bangladeshi	F	10
5	Other Asian or Bangladeshi	M	50
6	Other Mixed	F	13
7	Other Mixed	M	55
8	Pakistani	F	16
9	Pakistani	M	72
10	White	F	11
11	White	M	86

A Level

In [40]:
```
# Calculate the average diagnostic score for each ethnicity and each gend
avg_diagnostic_score = df_a_level.groupby(['Ethnicity', 'Gender'])['Diagn

# Display the result
avg_diagnostic_score
```

Out[40]:

	Ethnicity	Gender	Diagnostic Score
0	Black	F	5.121364
1	Black	M	4.991400
2	Indian	F	5.898438
3	Indian	M	5.088000
4	Other Asian or Bangladeshi	F	5.848214
5	Other Asian or Bangladeshi	M	5.328385
6	Other Mixed	F	4.749773
7	Other Mixed	M	4.325806
8	Pakistani	F	4.893421
9	Pakistani	M	5.358465
10	White	F	4.441429
11	White	M	5.489551

GCSE

In [41]:
```
# Calculate the average diagnostic score for each ethnicity and each gend
avg_diagnostic_score = df_gcse.groupby(['Ethnicity', 'Gender'])['Diagnost

# Display the result
avg_diagnostic_score
```

Out[41]:

	Ethnicity	Gender	Diagnostic Score
0	Black	F	4.875926
1	Black	M	6.361786
2	Indian	F	7.462647
3	Indian	M	8.062951
4	Other Asian or Bangladeshi	F	6.531000
5	Other Asian or Bangladeshi	M	7.063800
6	Other Mixed	F	3.301154
7	Other Mixed	M	6.784545
8	Pakistani	F	6.720313
9	Pakistani	M	7.750139
10	White	F	7.616364
11	White	M	7.719360

In [42]:
```
# Get unique values for the 'Socio-economic classification' column
unique_socio_economic_classification = df_a_level['Socio-economic classif

# Display the unique values
unique_socio_economic_classification
```

Out[42]:
```
array([' ', 'Routine occupations',
       'Never worked and long-term unemployed',
       'Semi-routine occupations', 'Not classified',
       'Intermediate occupations',
       'Lower supervisory and technical occupations',
       'Higher managerial and professional occupations',
       'Small employers and own account workers',
       'Lower managerial and professional occupations'], dtype=object)
```

In [43]:
```
# Filter rows where the 'Socio-economic classification' column has blank
blank_socio_economic_classification = df_a_level[df_a_level['Socio-econom

# Display the first few rows of the filtered data
blank_socio_economic_classification.head()
```

Out[43]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mat
0	1	Other Mixed	Other	M	A/AS level	A-Level		
53	54	Other Mixed	Other	F	A/AS level	A-Level		
79	80	Pakistani	Asian	M	A/AS level	A-Level		
90	91	Black	Black	M	A-Level	Level 2 Quals		
102	103	White	White	F	Other qualification level not known	Other		

In [44]:
```
# Count the unique values in the 'Socio-economic classification' column,
socio_economic_classification_count = df_a_level['Socio-economic classifi
socio_economic_classification_count.columns = ['Socio-economic classifica

# Display the counts
socio_economic_classification_count
```

Out[44]:

	Socio-economic classification	Count
0	Not classified	133
1	Higher managerial and professional occupations	96
2	Lower managerial and professional occupations	62
3	Intermediate occupations	50
4	Routine occupations	49
5	Small employers and own account workers	46
6	Semi-routine occupations	36
7		27
8	Lower supervisory and technical occupations	27
9	Never worked and long-term unemployed	10

A Level

In [45]:
```
# Calculate the average diagnostic score for each unique value in the 'So
avg_diagnostic_score_socio_economic = df_a_level.groupby('Socio-economic

# Display the result
avg_diagnostic_score_socio_economic
```

Out[45]:

	Socio-economic classification	Diagnostic Score
0		3.386852
1	Higher managerial and professional occupations	4.794688
2	Intermediate occupations	4.736800
3	Lower managerial and professional occupations	5.305000
4	Lower supervisory and technical occupations	5.341852
5	Never worked and long-term unemployed	6.029000
6	Not classified	5.310414
7	Routine occupations	5.343673
8	Semi-routine occupations	6.784722
9	Small employers and own account workers	4.809348

GCSE

In [46]:

```
# Calculate the average diagnostic score for each unique value in the 'So
avg_diagnostic_score_socio_economic = df_gcse.groupby('Socio-economic cla

# Display the result
avg_diagnostic_score_socio_economic
```

Out[46]:

	Socio-economic classification	Diagnostic Score
0		5.798939
1	Higher managerial and professional occupations	6.874598
2	Intermediate occupations	7.450435
3	Lower managerial and professional occupations	7.365086
4	Lower supervisory and technical occupations	7.927368
5	Never worked and long-term unemployed	7.917333
6	Not classified	6.782311
7	Routine occupations	6.723651
8	Semi-routine occupations	8.600500
9	Small employers and own account workers	7.571429

In [47]:
```
# Define a mapping function to convert 'Socio-economic classification' to
def map_socio_economic_class(value):
    if pd.isna(value) or value == 'Not classified':
        return 'Unknown'
    elif value in ['Routine occupations', 'Never worked and long-term une
        return 'Working-Class'
    elif value in ['Intermediate occupations', 'Lower supervisory and tec
        return 'Middle Income'
    elif value in ['Higher managerial and professional occupations', 'Sma
        return 'Professional'
    else:
        return 'Unknown'  # Default case

# Apply the mapping function to the 'Socio-economic classification' colum
df_a_level['Socio-economic classification'] = df_a_level['Socio-economic

# Display the first few rows to verify the changes
df_a_level.head()
```

```
/var/folders/jh/fgdcbdr1491cv5wb98g38zfc0000gn/T/ipykernel_54125/14848811
65.py:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_a_level['Socio-economic classification'] = df_a_level['Socio-economi
c classification'].apply(map_socio_economic_class)
```

Out[47]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature S
0	1	Other Mixed	Other	M	A/AS level	A-Level	Unknown	M
1	2	Other Asian or Bangladeshi	Asian	F	A/AS level	A-Level	Working-Class	
2	3	Black	Black	M	A/AS level	A-Level	Working-Class	
3	4	Black	Black	F	A/AS level	A-Level	Working-Class	
4	5	Other Asian or Bangladeshi	Asian	M	A/AS level	A-Level	Unknown	

In [48]:
```
# Define a mapping function to convert 'Socio-economic classification' to
def map_socio_economic_class(value):
    if pd.isna(value) or value == 'Not classified':
        return 'Unknown'
    elif value in ['Routine occupations', 'Never worked and long-term une
        return 'Working-Class'
    elif value in ['Intermediate occupations', 'Lower supervisory and tec
        return 'Middle Income'
    elif value in ['Higher managerial and professional occupations', 'Sma
        return 'Professional'
    else:
        return 'Unknown'  # Default case

# Apply the mapping function to the 'Socio-economic classification' colum
df_gcse['Socio-economic classification'] = df_gcse['Socio-economic classi

# Display the first few rows to verify the changes
df_gcse.head()
```

```
/var/folders/jh/fgdcbdr1491cv5wb98g38zfc0000gn/T/ipykernel_54125/32428429
24.py:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_gcse['Socio-economic classification'] = df_gcse['Socio-economic clas
sification'].apply(map_socio_economic_class)
```

Out[48]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature S
162	163	Pakistani	Asian	M	A/AS level	A-Level	Unknown	M
163	164	Black	Black	M	A/AS level	A-Level	Unknown	Y
164	165	Pakistani	Asian	F	Level 3 quals (all are in UCAS tariff)	Level 3 Quals	Unknown	Y
165	166	Black	Black	F	A/AS level	A-Level	Working-Class	Y
166	167	White	White	M	Level 3 qual	Level 2 Quals	Professional	Y

A Level

In [49]:
```
# Calculate the average diagnostic score for each of the new socio-econom
avg_diagnostic_score_by_category = df_a_level.groupby('Socio-economic cla

avg_diagnostic_score_by_category
```

Out[49]:
```
Socio-economic classification
Middle Income     5.107770
Professional      4.799437
Unknown           4.985812
Working-Class     5.961895
Name: Diagnostic Score, dtype: float64
```

GSCE

In [50]:
```
# Calculate the average diagnostic score for each of the new socio-econom
avg_diagnostic_score_by_category = df_gcse.groupby('Socio-economic classi

avg_diagnostic_score_by_category
```

Out[50]:
```
Socio-economic classification
Middle Income     7.483862
Professional      7.044261
Unknown           6.568816
Working-Class     7.289388
Name: Diagnostic Score, dtype: float64
```

In [51]:
```
# Find the unique values in the 'Highest Qual on Entry' column
unique_highest_qual_on_entry = df_a_level['Highest Qual on Entry'].unique

unique_highest_qual_on_entry
```

Out[51]:
```
array(['A/AS level', 'Level 3 quals (all are in UCAS tariff)', nan,
       'A-Level', 'Student has no formal qualification',
       'Diploma at level 3', 'Other qualification level not known',
       'Not known', 'Level 3 quals (none are in UCAS Tariff)',
       'Level 3 quals (some are in UCAS tariff)',
       'HE access course, QAA recognised',
       'Higher National Certificate (HNC)', 'Foundation degree',
       'International Baccalaureate (IB) Diploma'], dtype=object)
```

In [ ]:
```

```

A LEVEL

In [52]:
```
# Mapping the values in the 'Highest Qual on Entry' column to the new cat
mapping_dict = {
    'A/AS level': 'A-Level',
    'A-Level': 'A-Level',
    'Level 3 quals (all are in UCAS tariff)': 'Level 3',
    'Level 3 quals (none are in UCAS Tariff)': 'Level 3',
    'Level 3 quals (some are in UCAS tariff)': 'Level 3',
    'Level 3 qual': 'Level 3',  # Assuming 'Level 3 qual' is a typo and s
    'NaN': 'Unknown',
    'Not known': 'Unknown',
    'Student has no formal qualification': 'Unknown',
    'Other qualification level not known': 'Unknown',
    'Diploma at level 3': 'Diploma at level 3',
    'HE access course, QAA recognised': 'Other Qualification',
    'Higher National Certificate (HNC)': 'Other Qualification',
    'Foundation degree': 'Other Qualification',
    'International Baccalaureate (IB) Diploma': 'Other Qualification'
}

# Applying the mapping to the DataFrame
df_a_level['Highest Qual on Entry'] = df_a_level['Highest Qual on Entry']

# Displaying the unique values in the 'Highest Qual on Entry' column afte
unique_values_highest_qual_transformed = df_a_level['Highest Qual on Entr
unique_values_highest_qual_transformed
```

```
/var/folders/jh/fgdcbdr1491cv5wb98g38zfc0000gn/T/ipykernel_54125/43580312
6.py:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_a_level['Highest Qual on Entry'] = df_a_level['Highest Qual on Entry
'].map(mapping_dict).fillna('Unknown')
```

Out[52]:
```
array(['A-Level', 'Level 3', 'Unknown', 'Diploma at level 3',
       'Other Qualification'], dtype=object)
```

In [53]:
```
df_a_level
```

Out[53]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification
0	1	Other Mixed	Other	M	A-Level	A-Level	Unknown
1	2	Other Asian or Bangladeshi	Asian	F	A-Level	A-Level	Working-Class
2	3	Black	Black	M	A-Level	A-Level	Working-Class
3	4	Black	Black	F	A-Level	A-Level	Working-Class
4	5	Other Asian or Bangladeshi	Asian	M	A-Level	A-Level	Unknown
...	...	...	...	...	...	...	...
1060	1061	Black	Black	M	Other Qualification	Other	Middle Income
1061	1062	Pakistani	Asian	F	Level 3	Level 3 Quals	Unknown
1062	1063	Black	Black	M	A-Level	Level 2 Quals	Middle Income
1063	1064	Other Mixed	Other	M	A-Level	A-Level	Unknown
1064	1065	Black	Black	F	A-Level	Level 2 Quals	Middle Income

536 rows × 15 columns

GCSE

In [54]:
```
# Mapping the values in the 'Highest Qual on Entry' column to the new cat
mapping_dict = {
    'A/AS level': 'A-Level',
    'A-Level': 'A-Level',
    'Level 3 quals (all are in UCAS tariff)': 'Level 3',
    'Level 3 quals (none are in UCAS Tariff)': 'Level 3',
    'Level 3 quals (some are in UCAS tariff)': 'Level 3',
    'Level 3 qual': 'Level 3',  # Assuming 'Level 3 qual' is a typo and s
    'NaN': 'Unknown',
    'Not known': 'Unknown',
    'Student has no formal qualification': 'Unknown',
    'Other qualification level not known': 'Unknown',
    'Diploma at level 3': 'Diploma at level 3',
    'HE access course, QAA recognised': 'Other Qualification',
    'Higher National Certificate (HNC)': 'Other Qualification',
    'Foundation degree': 'Other Qualification',
    'International Baccalaureate (IB) Diploma': 'Other Qualification'
}

# Applying the mapping to the DataFrame
df_gcse['Highest Qual on Entry'] = df_gcse['Highest Qual on Entry'].map(m

# Displaying the unique values in the 'Highest Qual on Entry' column afte
unique_values_highest_qual_transformed = df_gcse['Highest Qual on Entry']
unique_values_highest_qual_transformed
```

```
/var/folders/jh/fgdcbdr1491cv5wb98g38zfc0000gn/T/ipykernel_54125/24640065
28.py:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_gcse['Highest Qual on Entry'] = df_gcse['Highest Qual on Entry'].map
(mapping_dict).fillna('Unknown')
```

Out[54]:
```
array(['A-Level', 'Level 3', 'Unknown', 'Diploma at level 3',
       'Other Qualification'], dtype=object)
```

In [55]:
```
df_gcse
```

Out[55]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature S
162	163	Pakistani	Asian	M	A-Level	A-Level	Unknown	M
163	164	Black	Black	M	A-Level	A-Level	Unknown	
164	165	Pakistani	Asian	F	Level 3	Level 3 Quals	Unknown	
165	166	Black	Black	F	A-Level	A-Level	Working-Class	
166	167	White	White	M	Level 3	Level 2 Quals	Professional	
...	...	...	...	...	...	...	...	
677	678	White	White	M	Level 3	Level 3 Quals	Professional	
678	679	Pakistani	Asian	F	Level 3	Level 2 Quals	Unknown	
679	680	Other Mixed	Other	M	Level 3	Level 2 Quals	Professional	
680	681	White	White	M	Level 3	Level 2 Quals	Unknown	
681	682	Indian	Asian	M	Level 3	Level 3 Quals	Professional	

488 rows × 16 columns

In [56]:
```
# Calculate the average diagnostic score for each unique value in 'Highes
avg_diagnostic_score_by_qual = df_a_level.groupby('Highest Qual on Entry'
avg_diagnostic_score_by_qual
```

Out[56]:

	Highest Qual on Entry	Diagnostic Score
0	A-Level	5.619358
1	Diploma at level 3	4.117432
2	Level 3	4.850068
3	Other Qualification	3.234375
4	Unknown	3.204808

In [57]:
```
# Calculate the average diagnostic score for each unique value in 'Highes
avg_diagnostic_score_by_qual = df_gcse.groupby('Highest Qual on Entry')['
avg_diagnostic_score_by_qual
```

Out[57]:

	Highest Qual on Entry	Diagnostic Score
0	A-Level	8.394125
1	Diploma at level 3	7.485000
2	Level 3	6.915197
3	Other Qualification	8.600000
4	Unknown	5.828358

3rd week

In [58]:
```
# Calculate summary statistics for Diagnostic Score across different Ethn
ethnicity_diagnostic_score_summary = df_a_level.groupby('Ethnicity')['Dia

ethnicity_diagnostic_score_summary.reset_index()
```

Out[58]:

	Ethnicity	count	mean	std	min	25%	50%	75%	max
0	Black	97.0	5.020876	2.000538	0.0	3.83500	4.8550	6.37500	9.5
1	Indian	71.0	5.270634	2.368308	0.0	3.74750	5.9150	7.00000	10.0
2	Other Asian or Bangladeshi	79.0	5.420506	2.266653	0.0	3.88250	5.3300	7.00000	10.0
3	Other Mixed	84.0	4.436845	2.478352	0.0	2.28375	4.2475	6.46875	10.0
4	Pakistani	120.0	5.284833	2.214716	0.0	3.66500	5.3025	7.00000	10.0
5	White	85.0	5.403235	2.485269	0.0	3.62500	5.2600	7.22500	10.0

In [59]:
```
# Grouping the data by 'Ethnicity' and calculating mean and standard devi
ethnicity_scores = df_a_level.groupby('Ethnicity')['Diagnostic Score'].ag

# Renaming the columns for clarity
ethnicity_scores.columns = ['Ethnicity', 'Average Diagnostic Score', 'Sta

# Display the results
ethnicity_scores
```

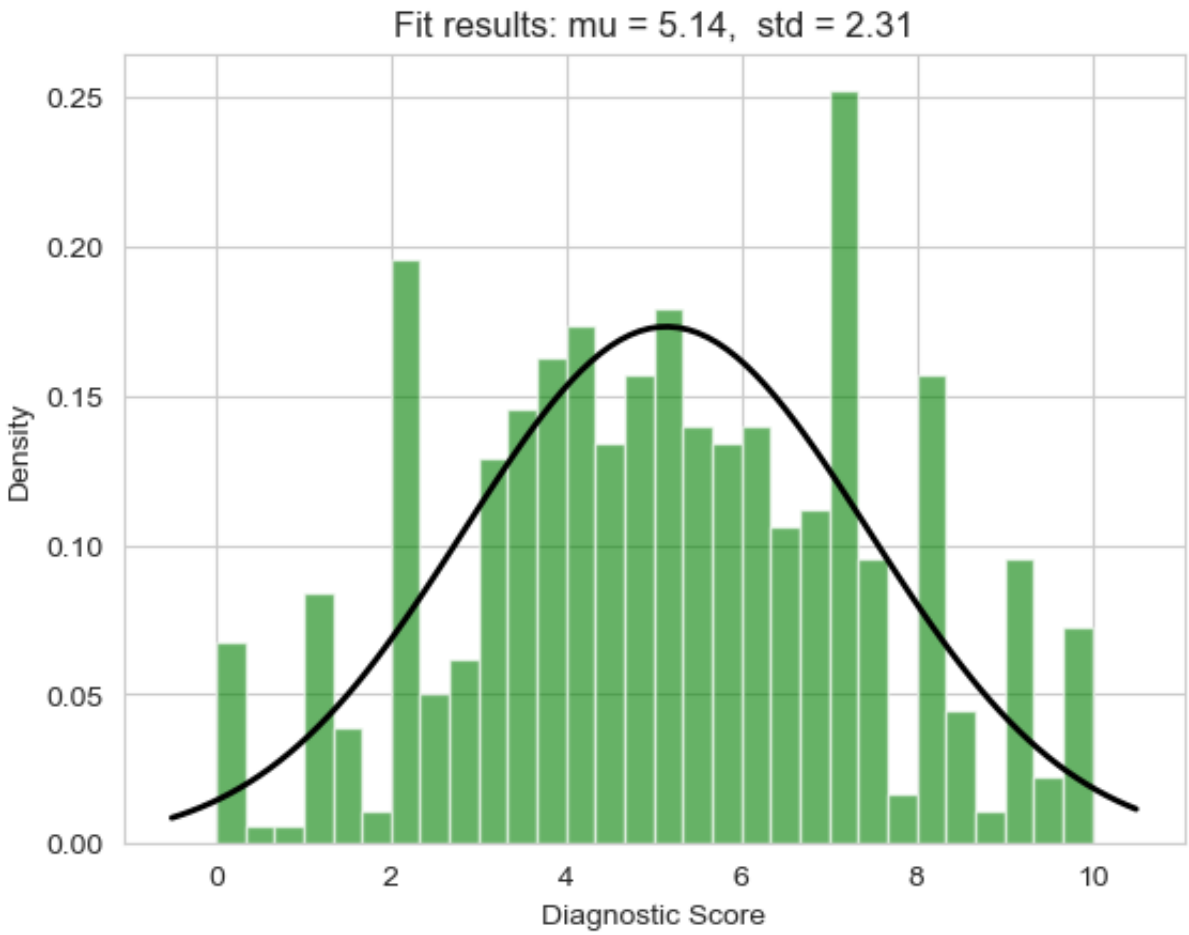
Out[59]:

	Ethnicity	Average Diagnostic Score	Standard Deviation
0	Black	5.020876	2.000538
1	Indian	5.270634	2.368308
2	Other Asian or Bangladeshi	5.420506	2.266653
3	Other Mixed	4.436845	2.478352
4	Pakistani	5.284833	2.214716
5	White	5.403235	2.485269

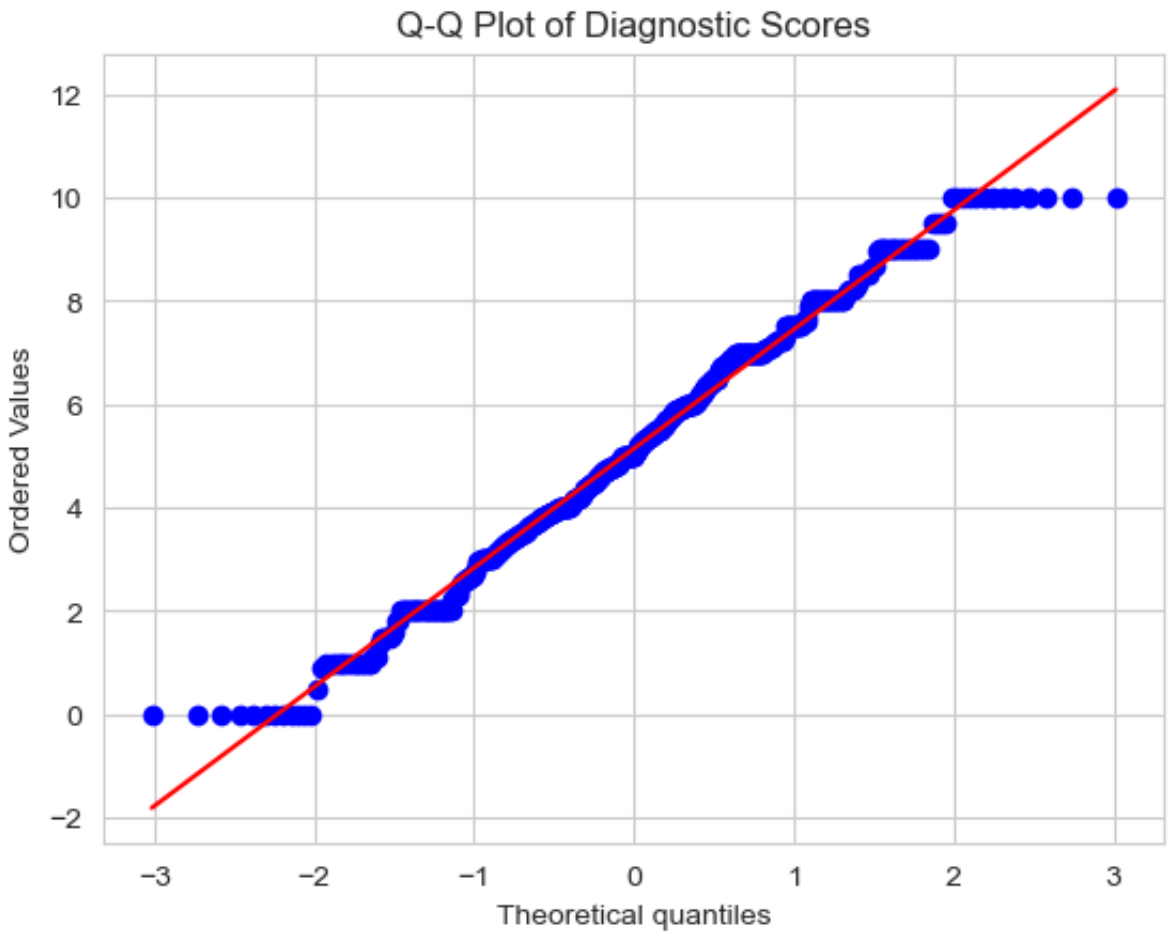
In [60]:
```
# Calculate the ratio of the highest standard deviation to the lowest sta
highest_sd = ethnicity_scores['Standard Deviation'].max()
lowest_sd = ethnicity_scores['Standard Deviation'].min()
ratio = highest_sd / lowest_sd

ratio, highest_sd, lowest_sd, ratio < 2
```

Out[60]:
```
(1.242300363833901, 2.485269184649435, 2.0005380800014985, True)
```

In [61]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Diagnostic scores from the dataset
diagnostic_scores = df_a_level['Diagnostic Score']

# Fit a normal distribution to the diagnostic scores
mu, std = norm.fit(diagnostic_scores)

# Plot the histogram
plt.hist(diagnostic_scores, bins=30, density=True, alpha=0.6, color='g')

# Plot the PDF of the Gaussian distribution
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2)

title = "Fit results: mu = %.2f,  std = %.2f" % (mu, std)
plt.title(title)
plt.xlabel('Diagnostic Score')
plt.ylabel('Density')

plt.show()
```



In [62]:
```
import scipy.stats as stats

# Generate a Q-Q plot
fig = plt.figure()
res = stats.probplot(diagnostic_scores, plot=plt)
plt.title('Q-Q Plot of Diagnostic Scores')
plt.show()
```



In [63]:
```
# Perform the Shapiro-Wilk test for normality
shapiro_test = stats.shapiro(diagnostic_scores)

shapiro_test_output = {
    'Statistic': shapiro_test[0],
    'p-value': shapiro_test[1]
}

shapiro_test_output
```

Out[63]:
```
{'Statistic': 0.9887768626213074, 'p-value': 0.0003973488346673548}
```

In [64]:
```
# Load the new data
new_file_path = 'df_new.csv'
new_data = pd.read_csv(new_file_path)

# Display the first few rows of the new dataframe
new_data
```

Out[64]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mat
0	1	Other Mixed	Other	M	A-Level	A-Level	Unknown	
1	2	Other Asian or Bangladeshi	Asian	F	A-Level	A-Level	Working-Class	
2	3	Black	Black	M	A-Level	A-Level	Working-Class	
3	4	Black	Black	F	A-Level	A-Level	Working-Class	
4	5	Other Asian or Bangladeshi	Asian	M	A-Level	A-Level	Unknown	
...	...	...	...	...	...	...	...	
1019	678	White	White	M	Level 3	Level 3 Quals	Professional	
1020	679	Pakistani	Asian	F	Level 3	Level 2 Quals	Unknown	
1021	680	Other Mixed	Other	M	Level 3	Level 2 Quals	Professional	
1022	681	White	White	M	Level 3	Level 2 Quals	Unknown	
1023	682	Indian	Asian	M	Level 3	Level 3 Quals	Professional	

1024 rows × 16 columns

In [65]:

```
# Basic information about the dataset, including data types and number of
data_info = new_data.info()

# Summary statistics for numeric columns
summary_statistics = new_data.describe()

# Distribution of categorical variables
ethnicity_counts = new_data['Ethnicity'].value_counts()
gender_counts = new_data['Gender'].value_counts()
soc_eco_class_counts = new_data['Socio-economic classification'].value_co

(data_info, summary_statistics, ethnicity_counts, gender_counts, soc_eco_
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1024 entries, 0 to 1023
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Student ID                      1024 non-null   int64  
 1   Ethnicity                       1024 non-null   object 
 2   Ethnicity Summary               1024 non-null   object 
 3   Gender                          1024 non-null   object 
 4   Highest Qual on Entry           1024 non-null   object 
 5   Qualification Summary           1024 non-null   object 
 6   Socio-economic classification   1024 non-null   object 
 7   Mature/Young Student            1024 non-null   object 
 8   Module Code                     1024 non-null   object 
 9   Mathematics Requirement         1024 non-null   object 
 10  Disability                      1022 non-null   object 
 11  Disability Summary              1024 non-null   object 
 12  Diagnostic Score                1024 non-null   float64
 13  PAL Attendance                  1024 non-null   int64  
 14  Final Module Score              1024 non-null   float64
 15  Score Type                      488 non-null    object 
dtypes: float64(2), int64(2), object(12)
memory usage: 128.1+ KB
```

Out[65]:

```
(None,
```

	Student ID	Diagnostic Score	PAL Attendance	Final Module Score
count	1024.000000	1024.000000	1024.000000	1024.000000
mean	528.883789	6.053745	1.382812	47.682881
std	311.905341	2.900907	2.621801	23.633391
min	1.000000	0.000000	0.000000	0.000000
25%	257.750000	3.905000	0.000000	31.487500
50%	514.500000	6.000000	0.000000	49.915000
75%	806.250000	8.003750	2.000000	64.900000
max	1065.000000	13.000000	25.000000	99.200000

```
,
 Pakistani                       208
 Black                           194
 White                           182
 Other Mixed                     152
 Indian                          149
 Other Asian or Bangladeshi      139
 Name: Ethnicity, dtype: int64,
 M    830
 F    194
 Name: Gender, dtype: int64,
 Unknown           312
 Middle Income     262
 Professional      257
 Working-Class     193
 Name: Socio-economic classification, dtype: int64)
```

In [66]:

```
# Calculate summary statistics for Diagnostic Score across different Ethn
ethnicity_diagnostic_score_summary = new_data.groupby('Ethnicity')['Diagn

ethnicity_diagnostic_score_summary.reset_index()
```

Out[66]:

	Ethnicity	count	mean	std	min	25%	50%	75%	max
0	Black	194.0	5.484536	2.759555	0.0	3.46625	5.010	7.5000	13.0
1	Indian	149.0	6.663893	2.929255	0.0	4.59500	7.000	8.8150	13.0
2	Other Asian or Bangladeshi	139.0	6.091511	2.688616	0.0	4.09250	6.000	7.7250	13.0
3	Other Mixed	152.0	5.189211	2.912565	0.0	3.16500	5.000	7.1275	13.0
4	Pakistani	208.0	6.248630	2.746497	0.0	4.17625	6.055	8.3600	13.0
5	White	182.0	6.631429	3.090227	0.0	4.00000	6.240	9.0000	13.0

In [67]:

```
# Grouping the new data by 'Ethnicity' and calculating mean and standard
ethnicity_scores_new = new_data.groupby('Ethnicity')['Diagnostic Score'].

# Renaming the columns for clarity
ethnicity_scores_new.columns = ['Ethnicity', 'Average Diagnostic Score',

# Display the results
ethnicity_scores_new
```

Out[67]:

	Ethnicity	Average Diagnostic Score	Standard Deviation
0	Black	5.484536	2.759555
1	Indian	6.663893	2.929255
2	Other Asian or Bangladeshi	6.091511	2.688616
3	Other Mixed	5.189211	2.912565
4	Pakistani	6.248630	2.746497
5	White	6.631429	3.090227

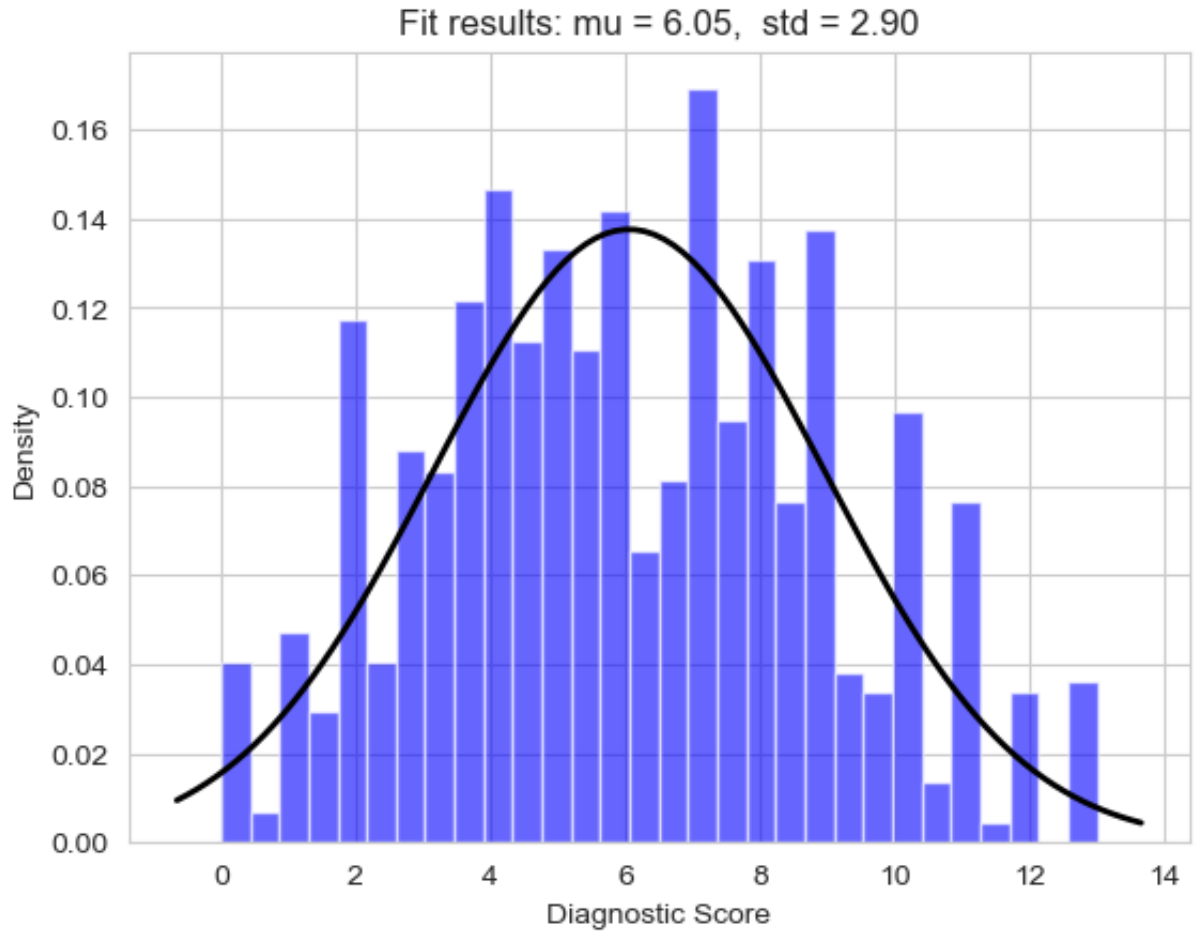
In [68]:

```
# Calculate the ratio of the highest standard deviation to the lowest sta
highest_sd_new = ethnicity_scores_new['Standard Deviation'].max()
lowest_sd_new = ethnicity_scores_new['Standard Deviation'].min()
ratio_new = highest_sd_new / lowest_sd_new

ratio_new, highest_sd_new, lowest_sd_new, ratio_new < 2
```

Out[68]:

```
(1.1493746897110155, 3.0902269567203384, 2.6886158050838116, True)
```

In [69]:
```
# Fit a normal distribution to the diagnostic scores of the new dataset
mu_new, std_new = norm.fit(new_data['Diagnostic Score'])

# Plot the histogram for the new dataset
plt.hist(new_data['Diagnostic Score'], bins=30, density=True, alpha=0.6,

# Plot the PDF of the Gaussian distribution
xmin_new, xmax_new = plt.xlim()
x_new = np.linspace(xmin_new, xmax_new, 100)
p_new = norm.pdf(x_new, mu_new, std_new)
plt.plot(x_new, p_new, 'k', linewidth=2)

title_new = "Fit results: mu = %.2f,  std = %.2f" % (mu_new, std_new)
plt.title(title_new)
plt.xlabel('Diagnostic Score')
plt.ylabel('Density')

plt.show()
```



While there are some indications that the diagnostic scores are not perfectly normally distributed, the overall shape is quite bell-like.

In [70]:
```
# Perform the Shapiro-Wilk test for normality on the diagnostic scores of
shapiro_test_new_data = stats.shapiro(new_data['Diagnostic Score'])

shapiro_test_new_data_output = {
    'Statistic': shapiro_test_new_data[0],
    'p-value': shapiro_test_new_data[1]
}

shapiro_test_new_data_output
```

Out[70]:
```
{'Statistic': 0.9880988001823425, 'p-value': 2.208033720307867e-07}
```

We've got lots of data, and the patterns look mostly like the bell shape we want to see. So, we're going to use ANOVA to understand our data better. Just to be sure, we'll also use another simple method to check our results. This way, we make sure we're on the right track.

In [71]:
```
# Import the required modules for the ANOVA test
from statsmodels.formula.api import ols
import statsmodels.api as sm

# Define the ANOVA model using Ordinary Least Squares regression
# 'Q("Diagnostic Score") ~ C(Ethnicity)' tells the model to look at the D
# 'Q()' is used to quote the column name because it contains spaces
# 'C()' is used to indicate that 'Ethnicity' is a categorical variable
model = ols('Q("Diagnostic Score") ~ C(Ethnicity)', data=new_data).fit()

# Perform the ANOVA test and store the results
# 'anova_lm' performs the ANOVA test on the fitted model
# 'typ=2' specifies the type of ANOVA test to be performed
anova_results = sm.stats.anova_lm(model, typ=2)

# Output the results of the ANOVA test
anova_results
```

Out[71]:

	sum_sq	df	F	PR(>F)
C(Ethnicity)	300.768161	5.0	7.370734	8.443707e-07
Residual	8308.045702	1018.0	NaN	NaN

ANOVA Test Results Interpretation

The ANOVA test we conducted provided us with the following results:

- `sum_sq`: This is the sum of squares which tells us how much variance there is.
- `df`: This stands for degrees of freedom and is related to the number of groups and samples we have.
- `F`: This is the F-statistic, a ratio of variances that tells us if the group means are different.
- `PR(>F)`: This is the p-value which tells us if the results are statistically significant.

Detailed Breakdown

- **C(Ethnicity)**:
  - **Sum of Squares (sum_sq)**: 300.768161, indicating the total variance explained by the differences among ethnic groups.
  - **Degrees of Freedom (df)**: 5, corresponding to the number of groups minus one.
  - **F-Statistic (F)**: 7.370734, which is the ratio of the variance calculated between groups to the variance within the groups.
  - **P-value (PR(>F))**: 8.443707e-07, which is much less than 0.05, suggesting the differences in group means are statistically significant.
- **Residual**:
  - **Sum of Squares (sum_sq)**: 8308.045702, indicating the total variance not explained by the model.
  - **Degrees of Freedom (df)**: 1018, representing the number of observations minus the number of groups.

Conclusion

The ANOVA test shows that there are statistically significant differences in the diagnostic scores between the different ethnic groups since the p-value is significantly below the 0.05 threshold. The large F-statistic further supports the presence of differences among the group means. With such a small p-value, we have strong evidence against the null hypothesis, which suggests no difference in means, and can conclude that ethnicity does have an effect on diagnostic scores.

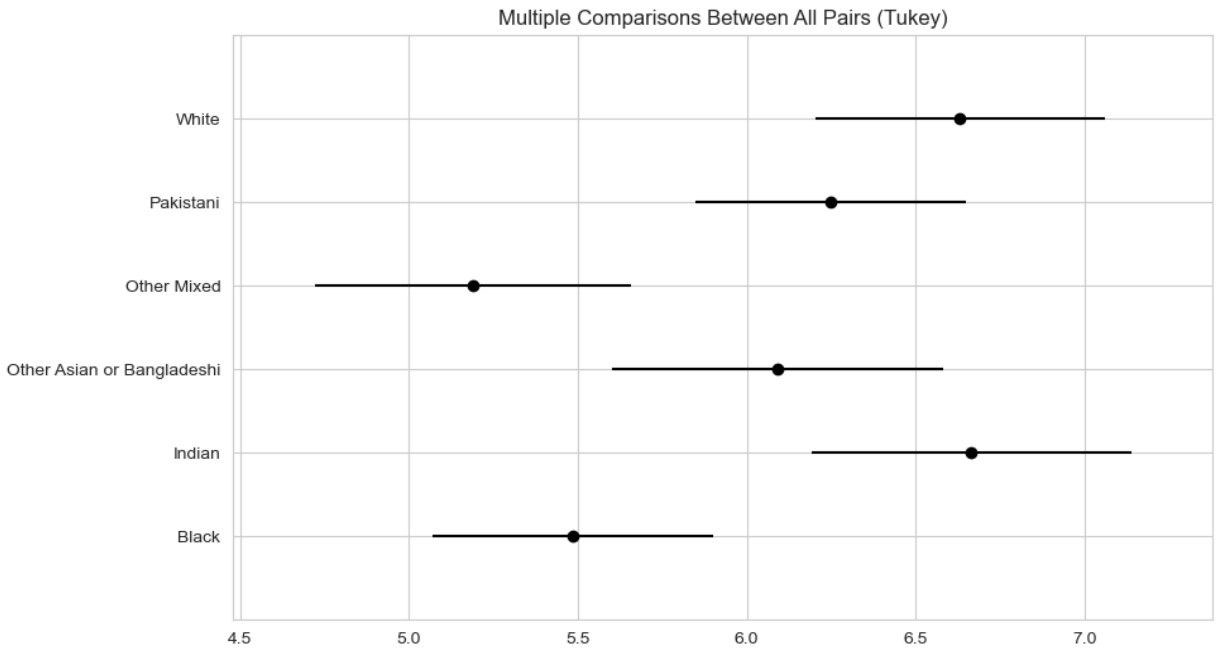
In [72]:
```
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# Perform Tukey's HSD test
tukey_results = pairwise_tukeyhsd(endog=new_data['Diagnostic Score'],  #
                                  groups=new_data['Ethnicity'],  # Groups
                                  alpha=0.05)  # Significance level

tukey_results.summary()
```

Out[72]:

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
Black	Indian	1.1794	0.0022	0.2909	2.0679	True
Black	Other Asian or Bangladeshi	0.607	0.3953	-0.2994	1.5134	False
Black	Other Mixed	-0.2953	0.932	-1.1789	0.5882	False
Black	Pakistani	0.7641	0.0802	-0.05	1.5782	False
Black	White	1.1469	0.0015	0.3052	1.9886	True
Indian	Other Asian or Bangladeshi	-0.5724	0.5327	-1.5342	0.3895	False
Indian	Other Mixed	-1.4747	0.0001	-2.415	-0.5344	True
Indian	Pakistani	-0.4153	0.7543	-1.2907	0.4602	False
Indian	White	-0.0325	1.0	-0.9336	0.8687	False
Other Asian or Bangladeshi	Other Mixed	-0.9023	0.0778	-1.8595	0.0549	False
Other Asian or Bangladeshi	Pakistani	0.1571	0.9961	-0.7365	1.0507	False
Other Asian or Bangladeshi	White	0.5399	0.5468	-0.3789	1.4587	False
Other Mixed	Pakistani	1.0594	0.007	0.1891	1.9298	True
Other Mixed	White	1.4422	0.0001	0.546	2.3385	True
Pakistani	White	0.3828	0.7739	-0.4451	1.2107	False

In [73]:
```
import matplotlib.pyplot as plt

# This function will plot the Tukey HSD test result
tukey_results.plot_simultaneous()
plt.show()
```


Multiple Comparisons Between All Pairs (Tukey)
White
Pakistani
Other Mixed
Other Asian or Bangladeshi
Indian
Black
4.5
5.0
5.5
6.0
6.5
7.0

Tukey HSD Test Results Explained

- `group1` and `group2`: These columns show which ethnic groups are being compared.
- `meandiff`: The average difference in scores between group1 and group2. A positive number means group1 scored higher than group2, and a negative number means group1 scored lower.
- `p-adj`: This is the adjusted p-value, which tells us if the difference in scores is likely to be real or just happened by chance. A p-value below 0.05 means it's likely real.
- `lower` and `upper`: These are the ends of a range that we're pretty sure contains the true difference in scores between the groups.
- `reject`: If this is True, it means we're confident there's a real difference in scores; if it's False, we can't be sure.

Breakdown of Results

- **Black vs. Indian**: The average score for Black students is 1.1794 points lower than for Indian students, and this difference is likely real because the p-value is 0.0022, which is less than 0.05.
- **Black vs. Other Asian or Bangladeshi**: We're not sure if there's a real difference in scores here because the p-value is 0.3953, which is more than 0.05.
- **Black vs. Other Mixed**: Again, we can't be sure of a real difference because the p-value is high (0.932).
- **Black vs. Pakistani**: The difference is not quite significant; the p-value is close to 0.05, but not below it.
- **Black vs. White**: Black students have an average score that's 1.1469 points lower than White students, and this difference is likely real (p-value is 0.0015).

When `reject` is **True**, it means we believe those two groups really do score differently. When it's **False**, we're not convinced there's a difference.

Summary of Tukey HSD Test Insights

We applied the Tukey HSD test to delve into the test scores across different ethnic groups. Here's what we discovered:

- **Black vs. Indian**: There's a notable difference in scores, with Black students scoring on average 1.1794 points lower than Indian students. This seems to be a genuine difference, as indicated by the low p-value of 0.0022.
- **Black vs. White**: Similar to the comparison with Indian students, Black students scored 1.1469 points lower on average compared to White students, and this difference is also likely to be real, supported by a p-value of 0.0015.
- **Other Mixed vs. Indian and White**: Students from Other Mixed backgrounds also show significant scoring differences when compared to Indian and White students, suggesting these differences are substantial.
- **Black vs. Pakistani and Indian vs. White**: For these comparisons, the differences in scores are not clear-cut. The p-values are not low enough to confidently state that there's a real difference in scores, so any observed differences might be due to chance.

In essence, the Tukey HSD test helped us pinpoint where the significant score differences lie between the ethnic groups, affirming that some disparities are indeed likely to be true, while others may not be as conclusive.

In [74]:
```
import pandas as pd
from scipy.stats import kruskal



# Separate the diagnostic scores for each ethnicity
groups = [new_data[new_data['Ethnicity'] == eth]['Diagnostic Score'] for

# Perform the Kruskal-Wallis H-test - it's the non-parametric alternative
kruskal_results = kruskal(*groups)
kruskal_results
```

Out[74]:
```
KruskalResult(statistic=34.32900300948385, pvalue=2.047555807948023e-06)
```

Kruskal-Wallis H-test Results Interpretation

The Kruskal-Wallis H-test was conducted to assess the differences in diagnostic scores across ethnic groups without assuming a normal distribution of the data.

Test Results

- **Statistic**: 34.32900300948385
    - This value indicates the extent to which the group medians differ from the overall median. A higher value suggests larger differences between the groups.
- **P-value**: (2.047555807948023 \times 10^{-6})
    - The p-value is significantly less than the conventional alpha level of 0.05. This low p-value indicates that we can reject the null hypothesis, which assumes no difference between the group medians.

Conclusion

Given the Kruskal-Wallis H-test statistic and the associated p-value, there is strong evidence to suggest that there are statistically significant differences in the median diagnostic scores across different ethnic groups. This conclusion is reached without relying on the data being normally distributed and is robust to the presence of outliers and heterogeneity of variance across the groups.

4th week

In [ ]:

week 4&5

To analyze the diagnostic scores by ethnicity for A-Level and GCSE students, we will follow these steps:

1. Load the datasets `df_a_level.csv` and `df_gcse.csv`.
2. Group the data by ethnicity.
3. Calculate the mean and standard deviation of diagnostic scores for each ethnic group.
4. Display the results.

In [1]:
```
import pandas as pd

# Load the data
df_a_level = pd.read_csv('df_a_level.csv')
df_gcse = pd.read_csv('df_gcse.csv')

# Calculate mean and standard deviation for each ethnicity in df_a_level
a_level_stats = df_a_level.groupby('Ethnicity')['Diagnostic Score'].agg([

# Calculate mean and standard deviation for each ethnicity in df_gcse
gcse_stats = df_gcse.groupby('Ethnicity')['Diagnostic Score'].agg(['mean'

# The results are stored in a_level_stats and gcse_stats
print(a_level_stats)
print(gcse_stats)
```

```
                              mean       std
Ethnicity
Black                     5.020876  2.000538
Indian                    5.270634  2.368308
Other Asian or Bangladeshi  5.420506  2.266653
Other Mixed               4.436845  2.478352
Pakistani                 5.284833  2.214716
White                     5.403235  2.485269
                              mean       std
Ethnicity
Black                     5.948196  3.297426
Indian                    7.932115  2.823877
Other Asian or Bangladeshi  6.975000  2.952764
Other Mixed               6.118603  3.150097
Pakistani                 7.562898  2.865349
White                     7.707680  3.177153
```

All the means are different.

In [2]:
```python
# Identify the ethnic group with the highest and lowest SD in df_a_level
max_sd_a_level = a_level_stats['std'].max()
min_sd_a_level = a_level_stats['std'].min()
ratio_a_level = max_sd_a_level / min_sd_a_level

# Identify the ethnic group with the highest and lowest SD in df_gcse
max_sd_gcse = gcse_stats['std'].max()
min_sd_gcse = gcse_stats['std'].min()
ratio_gcse = max_sd_gcse / min_sd_gcse

# Check if the ratio is less than 2 for both datasets
check_a_level = ratio_a_level < 2
check_gcse = ratio_gcse < 2

ratio_a_level, check_a_level, ratio_gcse, check_gcse
```

Out[2]:
```
(1.242300363833901, True, 1.1676947886429896, True)
```

Results of SD Ratio Check for ANOVA Test

A-Level Students (`df_a_level.csv`):

- **SD Ratio**: `1.242`
- **Ratio < 2**: `True` (Criteria met)

GCSE Students (`df_gcse.csv`):

- **SD Ratio**: `1.168`
- **Ratio < 2**: `True` (Criteria met)

Conclusion:

At both A-Level and GCSE levels, both datasets satisfy one of the key assumptions for conducting an ANOVA test: their ratio between highest standard deviation and lowest standard deviation falls under 2. This fulfills one of its key assumptions.

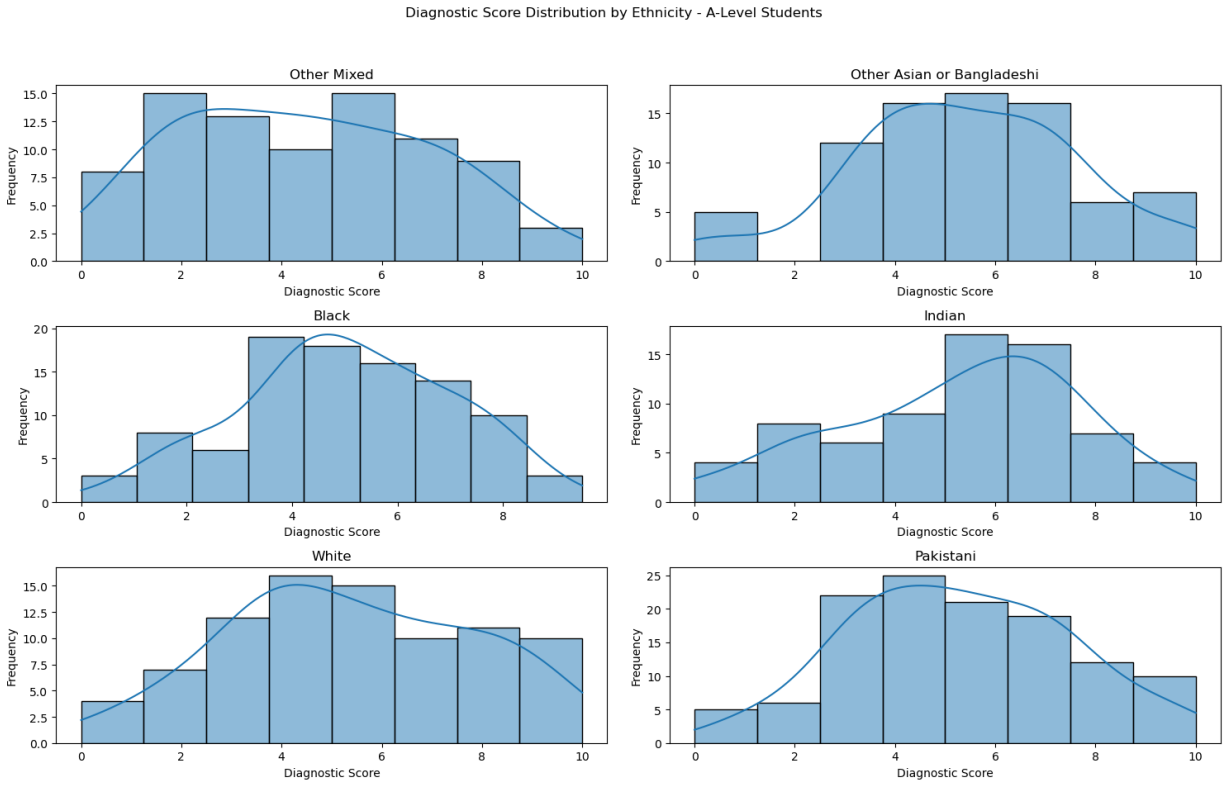
Next Steps in ANOVA Test

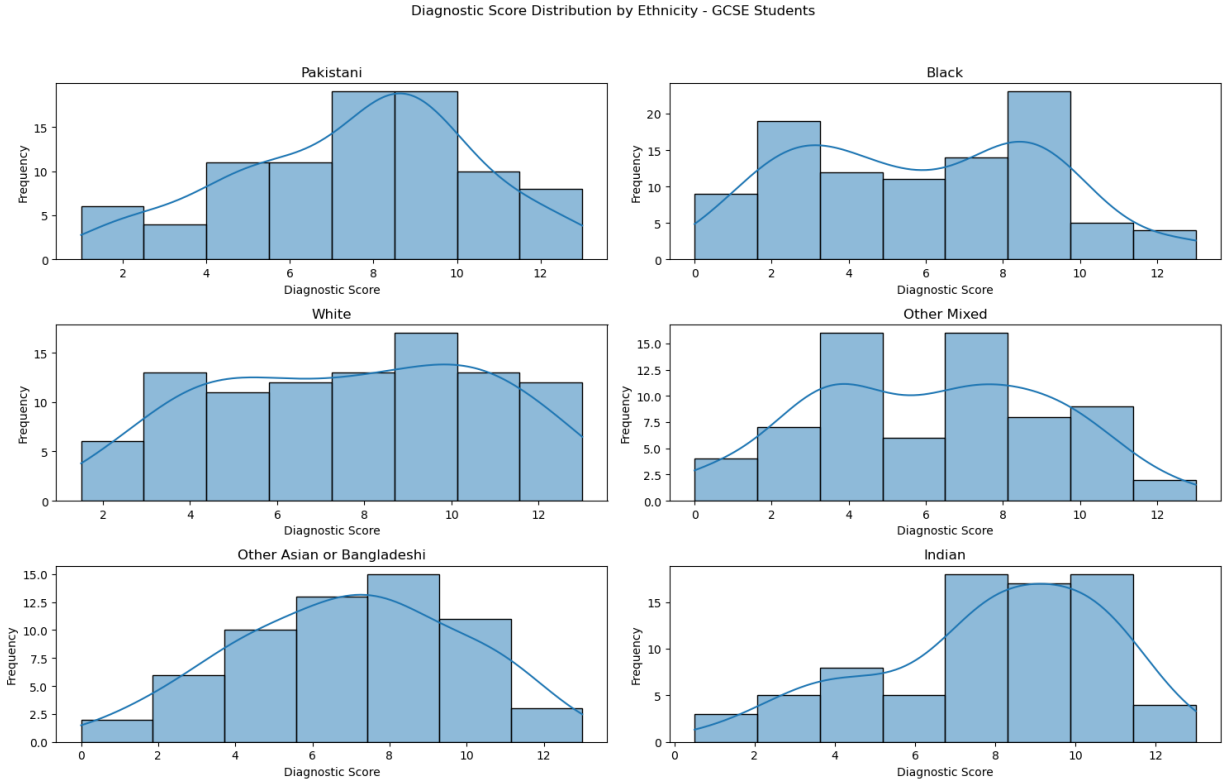
1. **Check Normality**:
   - Next, assess the normality of diagnostic scores across each ethnic group by employing normality tests such as **Shapiro-Wilk test** to ensure an approximate normal distribution and maintain the validity of ANOVA analyses.

In [3]:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Function to plot histograms for each ethnic group
def plot_histograms(df, title):
    plt.figure(figsize=(15, 10))
    ethnicities = df['Ethnicity'].unique()
    for i, ethnicity in enumerate(ethnicities, 1):
        plt.subplot(3, 2, i)
        sns.histplot(df[df['Ethnicity'] == ethnicity]['Diagnostic Score']
        plt.title(ethnicity)
        plt.xlabel('Diagnostic Score')
        plt.ylabel('Frequency')
    plt.suptitle(title)
    plt.tight_layout(rect=[0, 0.03, 1, 0.95])
    plt.show()

# Plot histograms for df_a_level
plot_histograms(df_a_level, 'Diagnostic Score Distribution by Ethnicity -

# Plot histograms for df_gcse
plot_histograms(df_gcse, 'Diagnostic Score Distribution by Ethnicity - GC
```

```
/Users/pawan/opt/anaconda3/lib/python3.9/site-packages/scipy/__init__.py:
146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for th
is version of SciPy (detected version 1.25.2
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```





Observations from Histograms of Diagnostic Scores

After plotting the histograms for each ethnic group in the A-Level and GCSE datasets, here are some noteworthy observations:

A-Level Students (`df_a_level.csv`)

- Distributions vary across ethnicities, with some ethnic groups showing more skew than others.
- For example, histograms from Black and Pakistani groups appear more symmetric and suggest a closer alignment with normal distribution.
- Conversely, Indian and Other Mixed groups display some deviation from normality, evidenced by slight skewness in their distributions.

GCSE Students (`df_gcse.csv`)

- The GCSE histograms display more variation in the distribution shapes.
- Notably, the **Black** and **Indian** groups show considerable skewness, which suggests a departure from normal distribution.
- The **White** and **Other Asian or Bangladeshi** groups have distributions that are somewhat more symmetric but still show signs of deviation from a perfect normal curve.

In [4]:
```
from scipy.stats import shapiro

# Function to perform Shapiro-Wilk test for normality
def perform_shapiro_wilk_test(df):
    results = {}
    for ethnicity in df['Ethnicity'].unique():
        w, p_value = shapiro(df[df['Ethnicity'] == ethnicity]['Diagnostic
        results[ethnicity] = {'W-Statistic': w, 'p-value': p_value}
    return results

# Perform Shapiro-Wilk test for df_a_level
shapiro_results_a_level = perform_shapiro_wilk_test(df_a_level)

# Perform Shapiro-Wilk test for df_gcse
shapiro_results_gcse = perform_shapiro_wilk_test(df_gcse)
```

In [5]:
```
shapiro_results_a_level
```

Out[5]:
```
{'Other Mixed': {'W-Statistic': 0.9725508093833923,
  'p-value': 0.0694502666592598},
 'Other Asian or Bangladeshi': {'W-Statistic': 0.9786108732223511,
  'p-value': 0.20662212371826172},
 'Black': {'W-Statistic': 0.9877718687057495, 'p-value': 0.51398926973342
9},
 'Indian': {'W-Statistic': 0.9665378928184509,
  'p-value': 0.055066388100385666},
 'White': {'W-Statistic': 0.9763375520706177, 'p-value': 0.11980746686458
588},
 'Pakistani': {'W-Statistic': 0.987772524356842,
  'p-value': 0.35778266191482544}}
```

In [6]:
```
shapiro_results_gcse
```

Out[6]:
```
{'Pakistani': {'W-Statistic': 0.9730101227760315,
  'p-value': 0.0628683865070343},
 'Black': {'W-Statistic': 0.9582348465919495, 'p-value': 0.00362366437911
9873},
 'White': {'W-Statistic': 0.9606016874313354, 'p-value': 0.00529326731339
097},
 'Other Mixed': {'W-Statistic': 0.9713308215141296,
  'p-value': 0.1186000257730484},
 'Other Asian or Bangladeshi': {'W-Statistic': 0.9847156405448914,
  'p-value': 0.6552410125732422},
 'Indian': {'W-Statistic': 0.9497873187065125,
  'p-value': 0.003884293604642153}}
```

Shapiro-Wilk Test Results for Normality in Diagnostic Scores

A-Level Students (`df_a_level.csv`)

- **Other Mixed**
  - W-Statistic: 0.973
  - p-value: 0.069
- **Other Asian or Bangladeshi**
  - W-Statistic: 0.979

    - p-value: 0.207
- **Black**
    - W-Statistic: 0.988
    - p-value: 0.514
- **Indian**
    - W-Statistic: 0.967
    - p-value: 0.055
- **White**
    - W-Statistic: 0.976
    - p-value: 0.120
- **Pakistani**
    - W-Statistic: 0.988
    - p-value: 0.358

GCSE Students (`df_gcse.csv`)

- **Pakistani**
    - W-Statistic: 0.973
    - p-value: 0.063
- **Black**
    - W-Statistic: 0.958
    - p-value: 0.004
- **White**
    - W-Statistic: 0.961
    - p-value: 0.005
- **Other Mixed**
    - W-Statistic: 0.971
    - p-value: 0.119
- **Other Asian or Bangladeshi**
    - W-Statistic: 0.985
    - p-value: 0.655
- **Indian**
    - W-Statistic: 0.950
    - p-value: 0.004

Interpretation of Results

- Shapiro-Wilk tests typically use p-values less than 0.05 as a proxy of non-normal distribution
- Several ethnic groups in the GCSE dataset, specifically Black, White and Indian populations show such results, this indicates their diagnostic scores do not follow an ideal distribution.
- For groups with p-values greater than 0.05, no evidence suggests deviation from normality.

In [7]:
```
import scipy.stats as stats

# Preparing data for ANOVA: Extracting diagnostic scores for each ethnic
groups_a_level = df_a_level.groupby('Ethnicity')['Diagnostic Score'].appl

# Conducting ANOVA test
anova_result = stats.f_oneway(*groups_a_level)

anova_result
```

Out[7]:
```
F_onewayResult(statistic=2.231873698235618, pvalue=0.04991931592032332)
```

The results of the ANOVA test on the A-Level dataset are as follows:

- F-statistic: 2.23
- p-value: approximately 0.0499 The p-value is just below the common alpha level of 0.05, indicating that there are statistically significant differences in mean diagnostic scores among the different ethnic groups in the A-Level dataset. This means we can reject the null hypothesis that all groups have the same mean diagnostic score.

In [8]:
```
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# Perform Tukey's HSD test
tukey_results = pairwise_tukeyhsd(endog=df_a_level['Diagnostic Score'],
                                  groups=df_a_level['Ethnicity'],
                                  alpha=0.05)

# Display the results
tukey_results.summary()
```

Out[8]:

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
Black	Indian	0.2498	0.9823	-0.7752	1.2747	False
Black	Other Asian or Bangladeshi	0.3996	0.8603	-0.5949	1.3941	False
Black	Other Mixed	-0.584	0.5271	-1.5621	0.394	False
Black	Pakistani	0.264	0.9594	-0.632	1.1599	False
Black	White	0.3824	0.8723	-0.5926	1.3573	False
Indian	Other Asian or Bangladeshi	0.1499	0.9987	-0.9233	1.223	False
Indian	Other Mixed	-0.8338	0.215	-1.8917	0.2241	False
Indian	Pakistani	0.0142	1.0	-0.9683	0.9967	False
Indian	White	0.1326	0.9992	-0.9224	1.1877	False
Other Asian or Bangladeshi	Other Mixed	-0.9837	0.0701	-2.0121	0.0448	False
Other Asian or Bangladeshi	Pakistani	-0.1357	0.9985	-1.0864	0.8151	False
Other Asian or Bangladeshi	White	-0.0173	1.0	-1.0428	1.0083	False
Other Mixed	Pakistani	0.848	0.0995	-0.0856	1.7815	False
Other Mixed	White	0.9664	0.0697	-0.0432	1.976	False
Pakistani	White	0.1184	0.9992	-0.8119	1.0487	False

Tukey HSD Test Results - A-Level Dataset (Ethnicity vs. Diagnostic Score)

The Tukey HSD (Honestly Significant Difference) test was conducted to compare the mean diagnostic scores across different ethnic groups in the A-Level dataset. This test helps in identifying which specific pairs of ethnic groups have statistically significant differences in their mean scores. The Family-Wise Error Rate (FWER) was set at 0.05.

Summary of Findings:

- No pair of ethnic groups showed a statistically significant difference in mean diagnostic scores.
- The comparisons and their respective results are as follows:

Group 1	Group 2	Mean Difference	Adjusted p-value	Lower Confidence Interval	Upper Confidence Interval	Significant Difference
Black	Indian	0.2498	0.9823	-0.7752	1.2747	False
Black	Other Asian or	0.3996	0.8603	-0.5949	1.3941	False

	Bangladeshi					
Black	Other Mixed	-0.584	0.5271	-1.5621	0.394	False
Black	Pakistani	0.264	0.9594	-0.632	1.1599	False
Black	White	0.3824	0.8723	-0.5926	1.3573	False
Indian	Other Asian or Bangladeshi	0.1499	0.9987	-0.9233	1.223	False
Indian	Other Mixed	-0.8338	0.215	-1.8917	0.2241	False
Indian	Pakistani	0.0142	1.0	-0.9683	0.9967	False
Indian	White	0.1326	0.9992	-0.9224	1.1877	False
Other Asian or Bangladeshi	Other Mixed	-0.9837	0.0701	-2.0121	0.0448	False
Other Asian or Bangladeshi	Pakistani	-0.1357	0.9985	-1.0864	0.8151	False
Other Asian or Bangladeshi	White	-0.0173	1.0	-1.0428	1.0083	False
Other Mixed	Pakistani	0.848	0.0995	-0.0856	1.7815	False
Other Mixed	White	0.9664	0.0697	-0.0432	1.976	False
Pakistani	White	0.1184	0.9992	-0.8119	1.0487	False

- In all pairwise comparisons, the null hypothesis that there is no difference in mean diagnostic scores between the groups cannot be rejected, as all adjusted p-values are greater than 0.05.

Conclusion:

The results indicate that there are no statistically significant differences in the mean diagnostic scores among the different ethnic groups in the A-Level dataset, as per the Tukey HSD test.

GCSE

In [9]:
```
from scipy.stats import kruskal

# Function to perform Kruskal-Wallis H Test
def perform_kruskal_test(df):
    # Extracting scores for each ethnic group
    groups = [df[df['Ethnicity'] == ethnicity]['Diagnostic Score'] for et
    # Performing the Kruskal-Wallis H Test
    test_statistic, p_value = kruskal(*groups)
    return test_statistic, p_value


# Perform Kruskal-Wallis test for df_gcse
kruskal_statistic_gcse, p_value_gcse = perform_kruskal_test(df_gcse)

kruskal_statistic_gcse, p_value_gcse
```

Out[9]:
```
(29.35168001497477, 1.9779859429130287e-05)
```

Kruskal-Wallis H Test Results for Diagnostic Scores

The Kruskal-Wallis H test was conducted to ascertain if there are statistically significant variations among different ethnic groups for GCSE students' median diagnostic scores.

Results for GCSE Students (`df_gcse.csv`)

- **Test Statistic**: `29.352`
- **p-value**: `0.00002`

Interpretation:

A very low p-value indicates statistically significant differences among ethnic groups for GCSE students' median diagnostic scores.

In [10]:
```
from scikit_posthocs import posthoc_dunn

# Perform Dunn's test for posthoc analysis
dunn_results = posthoc_dunn(df_gcse, val_col='Diagnostic Score', group_co

# Display the results
dunn_results.head()  # Displaying only the first few rows for an initial
```

Out[10]:

	Black	Indian	Other Asian or Bangladeshi	Other Mixed	Pakistani	White
Black	1.000000	0.000562	0.965435	1.000000	0.010403	0.002911
Indian	0.000562	1.000000	0.897656	0.007631	1.000000	1.000000
Other Asian or Bangladeshi	0.965435	0.897656	1.000000	1.000000	1.000000	1.000000
Other Mixed	1.000000	0.007631	1.000000	1.000000	0.081099	0.032578
Pakistani	0.010403	1.000000	1.000000	0.081099	1.000000	1.000000

Dunn's Test Results - GCSE Dataset (Ethnicity vs. Diagnostic Score)

Dunn's test, with Bonferroni correction for multiple comparisons, was conducted to evaluate the differences in diagnostic scores across various ethnic groups in the GCSE dataset. This test helps in identifying specific pairs of ethnic groups that have statistically significant differences in their scores.

Summary of Findings:

- The test revealed significant differences between certain ethnic groups, while others showed no significant difference.
- The results for each pairwise comparison are detailed below:

Ethnicity 1	Ethnicity 2	Adjusted p-value	Significant Difference
Black	Indian	0.000562	True
Black	Other Asian or Bangladeshi	0.965435	False
Black	Other Mixed	1.000000	False
Black	Pakistani	0.010403	True
Black	White	0.002911	True
Indian	Other Asian or Bangladeshi	0.897656	False
Indian	Other Mixed	0.007631	True
Indian	Pakistani	1.000000	False
Indian	White	1.000000	False
Other Asian or Bangladeshi	Other Mixed	1.000000	False
Other Asian or Bangladeshi	Pakistani	1.000000	False
Other Mixed	Pakistani	0.081099	False
Other Mixed	White	0.032578	True

- A significant difference is noted where the adjusted p-value is less than 0.05. In such cases, the null hypothesis of no difference in diagnostic scores between the groups is rejected.

Conclusion:

Dunn's post-hoc analysis reveals significant differences in diagnostic scores between several pairs of ethnic groups within the GCSE dataset. Notably, there are significant differences between Black and Indian, Black and Pakistani, Black and White, Indian and Other Mixed, and Other Mixed and White groups. Other

comparisons did not show significant differences, suggesting similar performance among these groups.

A LEVEL SOCIO-ECONOMIC AND GENDER

In [11]:
```
# A-Level dataset: Grouping by socio-economic classification and calculat
a_level_socio_econ = df_a_level.groupby('Socio-economic classification')[

# A-Level dataset: Grouping by gender and calculating mean and SD
a_level_gender = df_a_level.groupby('Gender')['Diagnostic Score'].agg(['m

# GCSE dataset: Grouping by socio-economic classification and calculating
gcse_socio_econ = df_gcse.groupby('Socio-economic classification')['Diagn

# GCSE dataset: Grouping by gender and calculating mean and SD
gcse_gender = df_gcse.groupby('Gender')['Diagnostic Score'].agg(['mean',

a_level_socio_econ, a_level_gender, gcse_socio_econ, gcse_gender
```

Out[11]:
```
(                                    mean       std
 Socio-economic classification
 Middle Income                   5.107770  2.172276
 Professional                    4.799437  2.283184
 Unknown                         4.985812  2.368489
 Working-Class                   5.961895  2.273542,
             mean       std
 Gender
 F       5.174800  2.117569
 M       5.133326  2.351118,
                                     mean       std
 Socio-economic classification
 Middle Income                   7.483862  3.290683
 Professional                    7.044261  3.119254
 Unknown                         6.568816  3.177893
 Working-Class                   7.289388  2.879215,
             mean       std
 Gender
 F       5.936649  3.453197
 M       7.323299  3.013279)
```

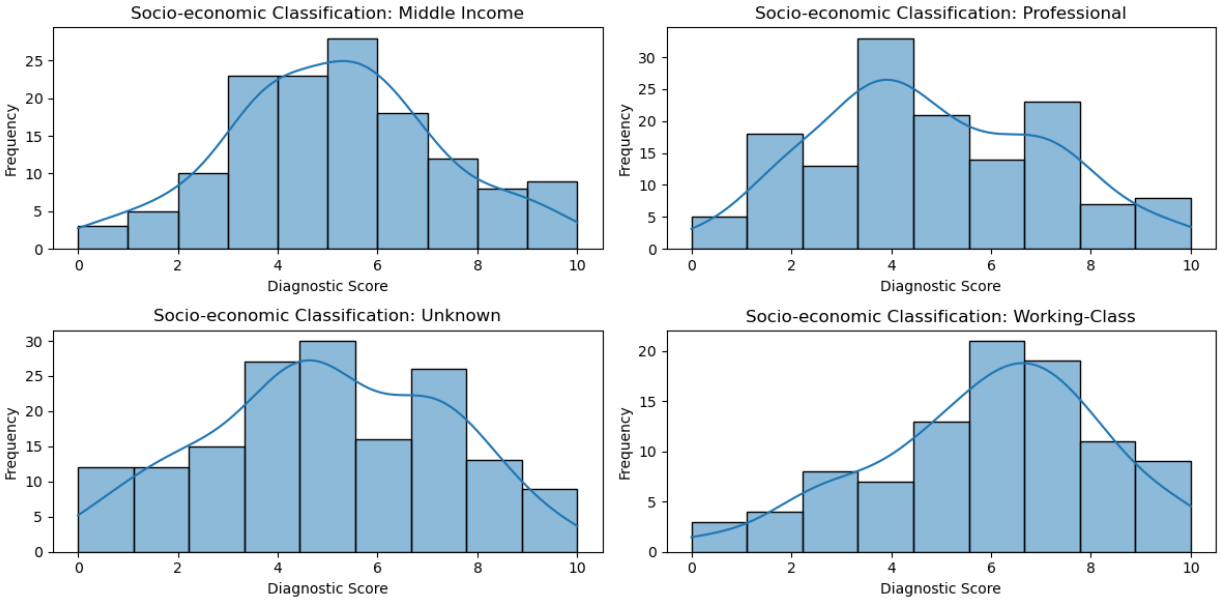
In [12]:
```
# Calculating the SD ratios for each grouping

# A-Level Dataset
sd_ratio_a_level_socio_econ = a_level_socio_econ['std'].max() / a_level_s
sd_ratio_a_level_gender = a_level_gender['std'].max() / a_level_gender['s

# GCSE Dataset
sd_ratio_gcse_socio_econ = gcse_socio_econ['std'].max() / gcse_socio_econ
sd_ratio_gcse_gender = gcse_gender['std'].max() / gcse_gender['std'].min(

sd_ratio_a_level_socio_econ, sd_ratio_a_level_gender, sd_ratio_gcse_socio
```

Out[12]:
```
(1.090325707004769, 1.1102915503387498, 1.1429098205152368, 1.14599307792
69678)
```

The ratios of the highest to lowest standard deviations for each grouping are as follows:

A-Level Dataset Socio-Economic Classification: Ratio = 1.09 Gender: Ratio = 1.11
GCSE Dataset Socio-Economic Classification: Ratio = 1.14 Gender: Ratio = 1.15

In [13]:
```
# Creating a function to plot histograms and perform Shapiro-Wilk test
def plot_histogram_and_shapiro(data, group_name):
    plt.figure(figsize=(12, 6))
    for i, (group, scores) in enumerate(data.items()):
        plt.subplot(2, 2, i + 1)
        sns.histplot(scores, kde=True)
        plt.title(f'{group_name}: {group}')
        plt.xlabel('Diagnostic Score')
        plt.ylabel('Frequency')

    plt.tight_layout()
    plt.show()

    # Shapiro-Wilk test for each group
    for group, scores in data.items():
        shapiro_test = shapiro(scores)
        print(f'{group_name} - {group}: W-Statistic={shapiro_test.statist

# Preparing data for histogram and Shapiro-Wilk test
a_level_socio_scores = df_a_level.groupby('Socio-economic classification'

# Plotting histograms and performing Shapiro-Wilk test
plot_histogram_and_shapiro(a_level_socio_scores, "Socio-economic Classifi
```




```
Socio-economic Classification - Middle Income: W-Statistic=0.988, p-value
=0.265
Socio-economic Classification - Professional: W-Statistic=0.980, p-value=
0.037
Socio-economic Classification - Unknown: W-Statistic=0.982, p-value=0.041
Socio-economic Classification - Working-Class: W-Statistic=0.974, p-value
=0.058
```

In [14]:
```
# Preparing data for Kruskal-Wallis test: Extracting diagnostic scores fo
groups_socio_economic_a_level = df_a_level.groupby('Socio-economic classi

# Conducting Kruskal-Wallis test for socio-economic groups in the A-Level
kruskal_result_socio_economic_a_level = stats.kruskal(*groups_socio_econo

kruskal_result_socio_economic_a_level
```

Out[14]:
```
KruskalResult(statistic=17.155571846836107, pvalue=0.0006565359961724939)
```

In [15]:
```
# Perform Dunn's post-hoc test
dunn_posthoc_a_level = posthoc_dunn(df_a_level, val_col='Diagnostic Score

# Display the results
dunn_posthoc_a_level
```

Out[15]:

	Middle Income	Professional	Unknown	Working-Class
Middle Income	1.000000	1.000000	1.000000	0.021986
Professional	1.000000	1.000000	1.000000	0.000359
Unknown	1.000000	1.000000	1.000000	0.007216
Working-Class	0.021986	0.000359	0.007216	1.000000

Dunn's Post-Hoc Test Results - A-Level Dataset (Socio-Economic Classification vs. Diagnostic Score)

Dunn's post-hoc test with Bonferroni correction was performed to assess the differences in diagnostic scores across various socio-economic classifications in the A-Level dataset. This test helps identify which specific pairs of socio-economic groups have statistically significant differences in their scores.

Summary of Findings:

- The test revealed significant differences between certain socio-economic groups, while others showed no significant difference.
- The results for each pairwise comparison are detailed below:

Socio-Economic Group 1	Socio-Economic Group 2	Adjusted p-value	Significant Difference
Middle Income	Professional	1.000000	False
Middle Income	Unknown	1.000000	False
Middle Income	Working-Class	0.021986	True
Professional	Unknown	1.000000	False
Professional	Working-Class	0.000359	True
Unknown	Working-Class	0.007216	True

- A significant difference is noted where the adjusted p-value is less than 0.05. In such cases, the null hypothesis of no difference in diagnostic scores between the groups is rejected.

Conclusion:

The Dunn's post-hoc analysis reveals significant differences in diagnostic scores between the 'Working-Class' group and each of the 'Middle Income', 'Professional', and 'Unknown' socio-economic groups within the A-Level dataset. These findings suggest that the socio-economic background may have a considerable impact on the diagnostic scores of students. Other pairwise comparisons did not show significant differences, indicating similar performance among these groups.

In [16]:
```
# Preparing data for Shapiro-Wilk test: Extracting diagnostic scores for
groups_gender_a_level = df_a_level.groupby('Gender')['Diagnostic Score'].

# Conducting Shapiro-Wilk test for normality for each gender group in the
shapiro_results_gender_a_level = {group: shapiro(scores) for group, score

shapiro_results_gender_a_level
```

Out[16]:
```
{'F': ShapiroResult(statistic=0.9874519109725952, pvalue=0.4687149226665497),
 'M': ShapiroResult(statistic=0.988063633441925, pvalue=0.0012364845024421811)}
```

In [17]:
```
# Conducting Kruskal-Wallis test for gender groups in the A-Level dataset
kruskal_result_gender_a_level = stats.kruskal(*groups_gender_a_level)

kruskal_result_gender_a_level
```

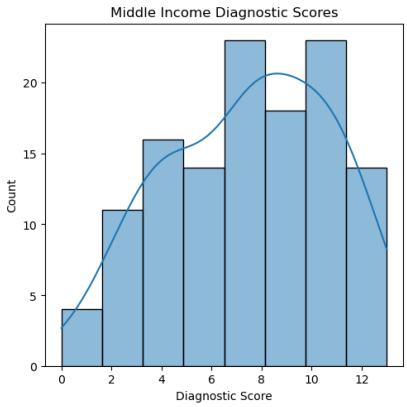
Out[17]:
```
KruskalResult(statistic=0.009138269241026564, pvalue=0.9238427963622049)
```

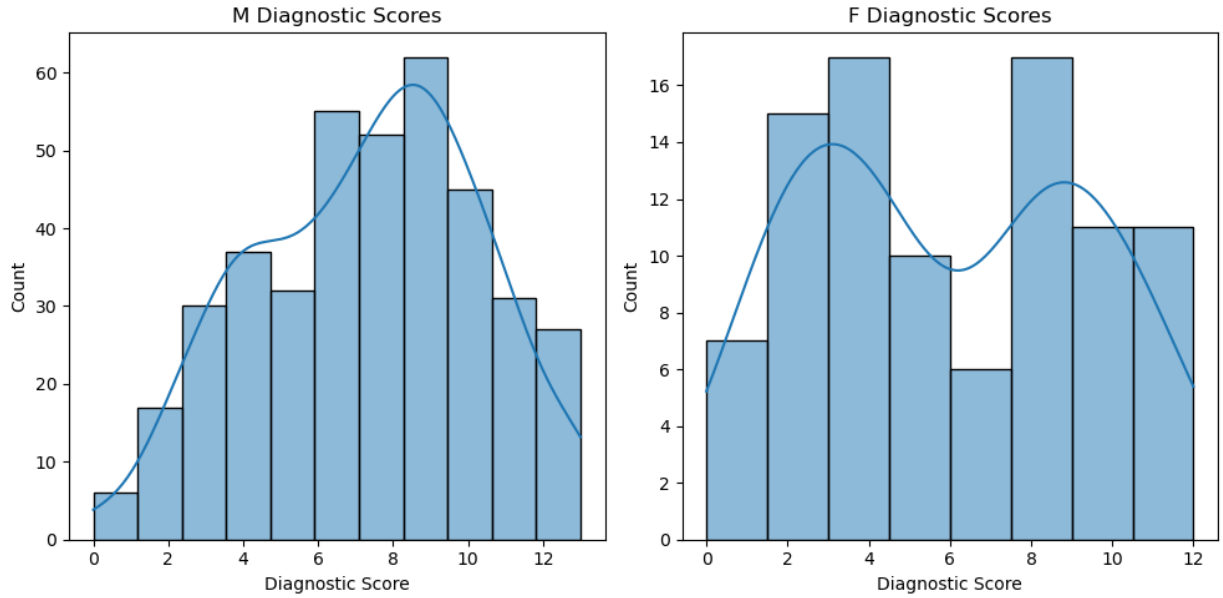
With a p-value much greater than 0.05, there is no statistically significant difference in the median diagnostic scores between male and female students in the A-Level dataset. This suggests that gender does not significantly impact the diagnostic scores in this context

In [ ]:
```

```

for gcse

In [18]:
```
# Creating histograms for each socio-economic classification group in the
plt.figure(figsize=(15, 10))

# Unique socio-economic classifications
socio_economic_classes = df_gcse['Socio-economic classification'].unique(

for i, soc_class in enumerate(socio_economic_classes, 1):
    plt.subplot(2, 3, i)
    sns.histplot(df_gcse[df_gcse['Socio-economic classification'] == soc_
    plt.title(f'{soc_class} Diagnostic Scores')

plt.tight_layout()
plt.show()

# Creating histograms for each gender group in the GCSE dataset
plt.figure(figsize=(10, 5))

# Unique genders
genders = df_gcse['Gender'].unique()

for i, gender in enumerate(genders, 1):
    plt.subplot(1, 2, i)
    sns.histplot(df_gcse[df_gcse['Gender'] == gender]['Diagnostic Score']
    plt.title(f'{gender} Diagnostic Scores')

plt.tight_layout()
plt.show()
```





In [19]:
```
# Shapiro-Wilk test for normality for each socio-economic classification
shapiro_results_socio_economic = {soc_class: shapiro(df_gcse[df_gcse['Soc
                                  for soc_class in socio_economic_classes

# Shapiro-Wilk test for normality for each gender group in the GCSE datas
shapiro_results_gender = {gender: shapiro(df_gcse[df_gcse['Gender'] == ge
                          for gender in genders}

shapiro_results_socio_economic, shapiro_results_gender
```

Out[19]:
```
({'Unknown': ShapiroResult(statistic=0.9669243693351746, pvalue=0.0010240
465635433793),
  'Working-Class': ShapiroResult(statistic=0.9674442410469055, pvalue=0.0
15631642192602158),
  'Professional': ShapiroResult(statistic=0.9758849143981934, pvalue=0.03
582238405942917),
  'Middle Income': ShapiroResult(statistic=0.9705034494400024, pvalue=0.0
08488167077302933)},
 {'M': ShapiroResult(statistic=0.9806662797927856, pvalue=4.0172697481466
46e-05),
  'F': ShapiroResult(statistic=0.9432168006896973, pvalue=0.0004800449532
9223573)})
```

In [20]:
```
from scipy.stats import kruskal, mannwhitneyu

# Kruskal-Wallis H test for socio-economic classification groups
soc_eco_groups = df_gcse['Socio-economic classification'].unique()
soc_eco_scores = [df_gcse[df_gcse['Socio-economic classification'] == gro
kruskal_test_socio_economic = kruskal(*soc_eco_scores)

kruskal_test_socio_economic
```

Out[20]:
```
KruskalResult(statistic=5.5446414013342595, pvalue=0.1359928147281751)
```

Kruskal-Wallis H Test Results - GCSE Dataset (Socio-Economic Classification vs. Diagnostic Score)

The Kruskal-Wallis H test was conducted to evaluate the differences in diagnostic scores across various socio-economic classifications in the GCSE dataset. This non-parametric test is used to determine if there are statistically significant differences among three or more groups.

Test Results:

- **Test Statistic**: 5.5446
- **p-value**: 0.13599

Interpretation:

- The Kruskal-Wallis H test yielded a p-value of 0.13599, which is above the conventional alpha level of 0.05.
- This result indicates that there are no statistically significant differences in diagnostic scores across the different socio-economic classification groups in the GCSE dataset.
- Since the p-value is greater than 0.05, we fail to reject the null hypothesis, which states that there are no differences between the groups.

Conclusion:

Based on the Kruskal-Wallis H test, it can be concluded that socio-economic classification does not significantly affect the diagnostic scores of students in the GCSE dataset. As a result, a post-hoc analysis is not required, as the initial test did not indicate significant differences across groups.

For Gender in gcse dataset, the next steps typically involve non-parametric tests, which do not require the assumption of normal distribution. The appropriate test to compare two independent non-normally distributed samples is the Mann-Whitney U test. This test is used to determine if there is a statistically significant difference between the two independent groups.

In [21]:
```
from scipy.stats import kruskal, mannwhitneyu
# Mann-Whitney U test for gender groups
male_scores = df_gcse[df_gcse['Gender'] == 'M']['Diagnostic Score']
female_scores = df_gcse[df_gcse['Gender'] == 'F']['Diagnostic Score']
mann_whitney_test_gender = mannwhitneyu(male_scores, female_scores, alter
mann_whitney_test_gender
```

Out[21]:
```
MannwhitneyuResult(statistic=22789.0, pvalue=0.0005062765213226672)
```

Mann-Whitney U Test for Gender Groups:

- Test Statistic: 22789.0
- p-value: 0.00051

The Mann-Whitney U test results show a statistically significant difference in diagnostic scores between the male and female groups. The p-value (0.00051) is less than 0.05, indicating that this difference is statistically significant.

Interpretation:

- The Mann-Whitney U test yielded a p-value of 0.00051.
- Since the p-value is less than 0.05, this indicates that there is a statistically significant difference in diagnostic scores between male and female students.
- The test statistic of 22789.0 helps in determining the direction of this difference. However, the test itself doesn't specify which group scored higher, only that their scores are significantly different.

Conclusion:

The results suggest that the diagnostic scores significantly differ between male and female students in the GCSE dataset. This significant difference warrants further investigation into possible factors contributing to this disparity, such as differences in learning styles, educational resources, socio-economic factors, or other external influences.

In [22]:
```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import shapiro, kruskal

# Calculating mean, variance, and standard deviation for 'Diagnostic Scor
mean_var_sd_a_level = df_a_level.groupby('Highest Qual on Entry')['Diagno
mean_var_sd_gcse = df_gcse.groupby('Highest Qual on Entry')['Diagnostic S
mean_var_sd_a_level
```

Out[22]:

Highest Qual on Entry	mean	var	std
A-Level	5.619358	5.014828	2.239381
Diploma at level 3	4.117432	1.955483	1.398386
Level 3	4.850068	4.756417	2.180921
Other Qualification	3.234375	5.570425	2.360175
Unknown	3.204808	4.189419	2.046807

In [23]:
```
mean_var_sd_gcse
```

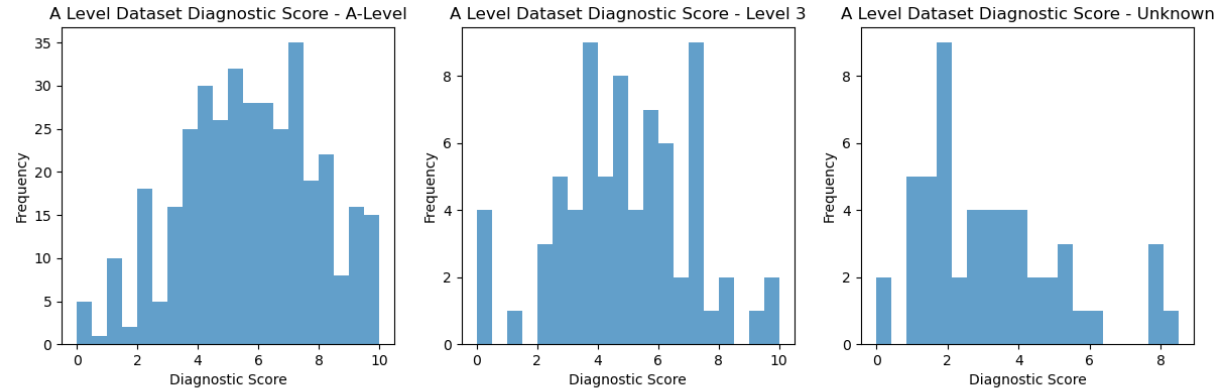
Out[23]:

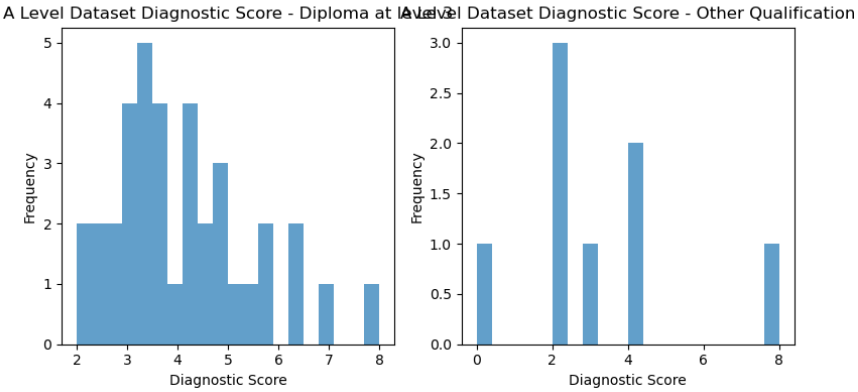
Highest Qual on Entry	mean	var	std
A-Level	8.394125	7.088717	2.662464
Diploma at level 3	7.485000	6.079947	2.465755
Level 3	6.915197	10.712296	3.272964
Other Qualification	8.600000	0.561800	0.749533
Unknown	5.828358	8.313966	2.883395

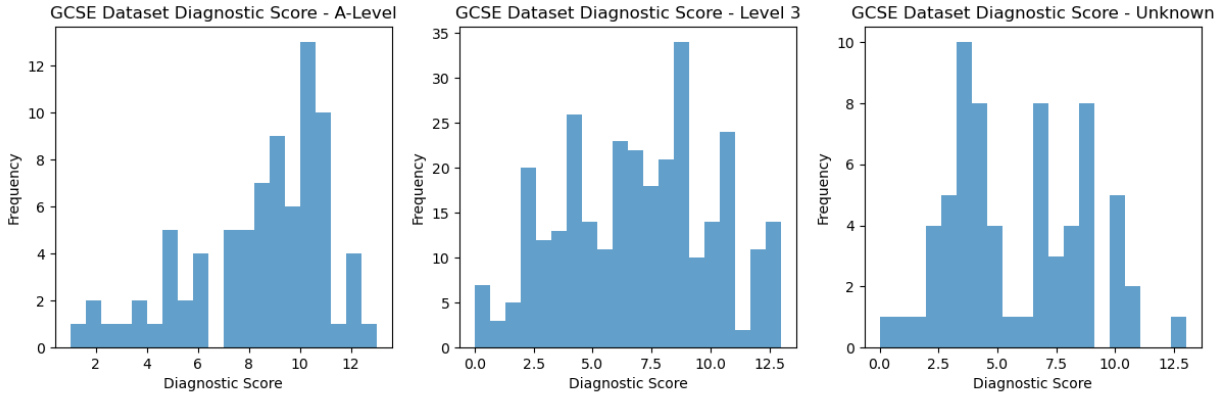
In [24]:
```
# Calculating the ratio of the highest standard deviation to the lowest s
sd_ratio_a_level = mean_var_sd_a_level['std'].max() / mean_var_sd_a_level
sd_ratio_gcse = mean_var_sd_gcse['std'].max() / mean_var_sd_gcse['std'].m
sd_ratio_a_level, sd_ratio_gcse
```

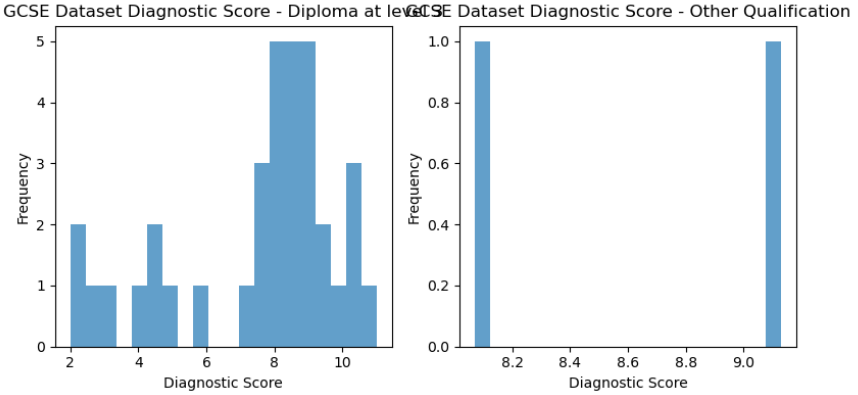
Out[24]:
```
(1.687785094728451, 4.366670392410784)
```

In [25]:
```
# Plotting histograms for 'Diagnostic Score' by 'Highest Qual on Entry' f
def plot_histograms_by_qualification_diagnostic(df, title_prefix):
    qualifications = df['Highest Qual on Entry'].unique()
    plt.figure(figsize=(12, 8))

    for i, qual in enumerate(qualifications, 1):
        plt.subplot(2, 3, i)
        plt.hist(df[df['Highest Qual on Entry'] == qual]['Diagnostic Scor
        plt.title(f'{title_prefix} - {qual}')
        plt.xlabel('Diagnostic Score')
        plt.ylabel('Frequency')

    plt.tight_layout()
    plt.show()

# Plotting for A Level Dataset
plot_histograms_by_qualification_diagnostic(df_a_level, 'A Level Dataset

# Plotting for GCSE Dataset
plot_histograms_by_qualification_diagnostic(df_gcse, 'GCSE Dataset Diagno
```









In [26]:
```
# Performing Shapiro-Wilk test for normality of 'Diagnostic Score' for ea
shapiro_a_level_qual = df_a_level.groupby('Highest Qual on Entry')['Diagn
shapiro_a_level_qual
```

Out[26]:
```
Highest Qual on Entry
A-Level                0.001122
Diploma at level 3     0.069723
Level 3                0.387398
Other Qualification    0.441462
Unknown                0.001815
Name: Diagnostic Score, dtype: float64
```

Statistical Analysis Report for A Level Dataset - Diagnostic Score by Highest Qualification on Entry

Descriptive Statistics

- **Mean, Variance, and Standard Deviation**:
  - **A-Level**:
    - Mean: 8.39, Variance: 7.09, Standard Deviation: 2.66
  - **Diploma at level 3**:
    - Mean: 7.49, Variance: 6.08, Standard Deviation: 2.47
  - **Level 3**:
    - Mean: 6.92, Variance: 10.71, Standard Deviation: 3.27
  - **Other Qualification**:
    - Mean: 8.60, Variance: 0.56, Standard Deviation: 0.75
  - **Unknown**:
    - Mean: 5.83, Variance: 8.31, Standard Deviation: 2.88
- **Standard Deviation Ratio**: 1.69

Shapiro-Wilk Test for Normality

- **Results**:
  - A-Level: p = 0.001122 (non-normal distribution)
  - Diploma at level 3: p = 0.069723 (normal distribution)
  - Level 3: p = 0.387398 (normal distribution)
  - Other Qualification: p = 0.441462 (normal distribution)
  - Unknown: p = 0.001815 (non-normal distribution)

Conclusion

- In the A Level dataset, the Kruskal-Wallis H test should be used to analyze the impact of the highest qualification on entry on diagnostic scores. This approach is justified due to the mixed normality results and the consistency in variability across the different qualification groups.

In [27]:
```
# Dropping null values from the 'Diagnostic Score' column
df_a_level_clean = df_a_level.dropna(subset=['Diagnostic Score'])

# Preparing the groups for the Kruskal-Wallis H test
diag_groups_a_level = [group['Diagnostic Score'].values for name, group i

# Performing Kruskal-Wallis H test
kruskal_result = kruskal(*diag_groups_a_level)

# Print the result
print(kruskal_result)
```

```
KruskalResult(statistic=70.05467128213941, pvalue=2.210311894988454e-14)
```

Kruskal-Wallis H Test Report for A Level Dataset

Context

- **Analysis Performed**: Kruskal-Wallis H test to assess the impact of highest qualification on entry on diagnostic scores.

Test Results

- **Statistic**: 70.055
- **p-value**: Approximately 2.21e-14

Conclusion

- The Kruskal-Wallis H test reveals that the type of qualification a student enters with has a significant impact on their diagnostic scores. This finding highlights the importance of prior educational background and its influence on students' initial academic performance in A Level courses.

In [28]:
```
import pandas as pd
from scipy.stats import mannwhitneyu
import itertools

# Preparing the groups for the post-hoc analysis
diag_groups_a_level = [group['Diagnostic Score'].values for name, group i

# Function for post-hoc analysis using Mann-Whitney U test
def post_hoc_mann_whitney(groups):
    comparisons = list(itertools.combinations(range(len(groups)), 2))
    results = {}
    for (i, j) in comparisons:
        group_i = groups[i]
        group_j = groups[j]
        stat, p = mannwhitneyu(group_i, group_j)
        results[f'Group {i+1} vs Group {j+1}'] = p
    return results

# Post-hoc analysis for A Level Dataset
post_hoc_results_a_level = post_hoc_mann_whitney(diag_groups_a_level)

# Print the results
for comparison, p_value in post_hoc_results_a_level.items():
    print(f'{comparison}: p-value = {p_value}')
```

```
Group 1 vs Group 2: p-value = 5.5191204548108995e-06
Group 1 vs Group 3: p-value = 0.00641124175133329
Group 1 vs Group 4: p-value = 0.006763183473969203
Group 1 vs Group 5: p-value = 1.0584726544354583e-11
Group 2 vs Group 3: p-value = 0.022926284299642247
Group 2 vs Group 4: p-value = 0.10244669490472605
Group 2 vs Group 5: p-value = 0.00370717113326443
Group 3 vs Group 4: p-value = 0.04107768952568866
Group 3 vs Group 5: p-value = 1.16477205761602e-05
Group 4 vs Group 5: p-value = 0.8702578036190389
```

Post-Hoc Analysis Report for A Level Dataset

Context

- **Analysis Performed**: Post-hoc Mann-Whitney U tests following a significant Kruskal-Wallis H test on the impact of the highest qualification on entry on diagnostic scores.
- **Group Identification**:
  - Group 1: A-Level
  - Group 2: Diploma at level 3
  - Group 3: Level 3
  - Group 4: Other Qualification
  - Group 5: Unknown

Post-Hoc Test Results

- **Pairwise Comparisons**:
  - A-Level vs Diploma at level 3: p-value ≈ 5.52e-06 (significant)
  - A-Level vs Level 3: p-value ≈ 0.0064 (significant)
  - A-Level vs Other Qualification: p-value ≈ 0.0068 (significant)
  - A-Level vs Unknown: p-value ≈ 1.06e-11 (significant)
  - Diploma at level 3 vs Level 3: p-value ≈ 0.0229 (significant)
  - Diploma at level 3 vs Other Qualification: p-value ≈ 0.1024 (not significant)
  - Diploma at level 3 vs Unknown: p-value ≈ 0.0037 (significant)
  - Level 3 vs Other Qualification: p-value ≈ 0.0411 (significant)
  - Level 3 vs Unknown: p-value ≈ 1.16e-05 (significant)
  - Other Qualification vs Unknown: p-value ≈ 0.8703 (not significant)

Conclusion

- The post-hoc analysis for the A Level dataset reveals significant differences in diagnostic scores across various qualification groups, highlighting the impact of students' educational background upon entry on their initial academic assessments.

In [32]:
```
# Perform Shapiro-Wilk test for normality of 'Diagnostic Score' for each
shapiro_gcse_qual = df_gcse.groupby('Highest Qual on Entry')['Diagnostic
    lambda x: shapiro(x.dropna())[1] if len(x.dropna()) >= 3 else None
)

# Print Shapiro-Wilk test results
print("Shapiro-Wilk Test Results for GCSE:")
print(shapiro_gcse_qual)
```

```
Shapiro-Wilk Test Results for GCSE:
Highest Qual on Entry
A-Level                0.000351
Diploma at level 3     0.001730
Level 3                0.000217
Other Qualification         NaN
Unknown                0.050370
Name: Diagnostic Score, dtype: float64
```

In [34]:

```python
import pandas as pd
from scipy.stats import kruskal

# Assuming df_gcse is your GCSE dataset DataFrame

# Drop null values from the 'Diagnostic Score' column in the GCSE dataset
df_gcse_clean = df_gcse.dropna(subset=['Diagnostic Score'])

# Prepare the groups for the Kruskal-Wallis H test
diag_groups_gcse = [group['Diagnostic Score'].values for name, group in d

# Perform Kruskal-Wallis H test
kruskal_result_gcse = kruskal(*diag_groups_gcse)

# Print Kruskal-Wallis H test result
print("Kruskal-Wallis H Test Result for GCSE:")
print(kruskal_result_gcse)
```

```
Kruskal-Wallis H Test Result for GCSE:
KruskalResult(statistic=29.87549709174556, pvalue=5.188540428225691e-06)
```

In [35]:

```python
import scikit_posthocs as sp

# Assuming df_gcse_clean is your cleaned GCSE dataset

# Perform Dunn's post-hoc test following the significant Kruskal-Wallis H
posthoc_result = sp.posthoc_dunn(df_gcse_clean, val_col='Diagnostic Score

# Print the post-hoc test results
print("Post-Hoc Dunn's Test Result for GCSE:")
print(posthoc_result)
```

```
Post-Hoc Dunn's Test Result for GCSE:
                     A-Level  Diploma at level 3   Level 3  \
A-Level             1.000000            1.000000  0.000586
Diploma at level 3  1.000000            1.000000  1.000000
Level 3             0.000586            1.000000  1.000000
Other Qualification 1.000000            1.000000  1.000000
Unknown             0.000002            0.080182  0.078255

                    Other Qualification   Unknown
A-Level                             1.0  0.000002
Diploma at level 3                  1.0  0.080182
Level 3                             1.0  0.078255
Other Qualification                 1.0  1.000000
Unknown                             1.0  1.000000
```

In [36]:
```
import pandas as pd
from scipy.stats import shapiro, kruskal
import scikit_posthocs as sp

# Load your GCSE dataset into a DataFrame
# df_gcse = pd.read_csv('path_to_your_gcse_dataset.csv')

# Perform Shapiro-Wilk test for normality of 'Diagnostic Score' for each
shapiro_gcse_qual = df_gcse.groupby('Highest Qual on Entry')['Diagnostic
    lambda x: shapiro(x.dropna())[1] if len(x.dropna()) >= 3 else None
)

# Print Shapiro-Wilk test results
print("Shapiro-Wilk Test Results for GCSE:")
print(shapiro_gcse_qual)

# Drop null values from the 'Diagnostic Score' column
df_gcse_clean = df_gcse.dropna(subset=['Diagnostic Score'])

# Prepare the groups for the Kruskal-Wallis H test
diag_groups_gcse = [group['Diagnostic Score'].values for name, group in d

# Perform Kruskal-Wallis H test
kruskal_result_gcse = kruskal(*diag_groups_gcse)

# Print Kruskal-Wallis H test result
print("Kruskal-Wallis H Test Result for GCSE:")
print(kruskal_result_gcse)

# Perform Dunn's post-hoc test if Kruskal-Wallis H test is significant
if kruskal_result_gcse.pvalue < 0.05:
    posthoc_result_gcse = sp.posthoc_dunn(df_gcse_clean, val_col='Diagnos
    print("Post-Hoc Dunn's Test Result for GCSE:")
    print(posthoc_result_gcse)
```

```
Shapiro-Wilk Test Results for GCSE:
Highest Qual on Entry
A-Level                 0.000351
Diploma at level 3      0.001730
Level 3                 0.000217
Other Qualification          NaN
Unknown                 0.050370
Name: Diagnostic Score, dtype: float64
Kruskal-Wallis H Test Result for GCSE:
KruskalResult(statistic=29.87549709174556, pvalue=5.188540428225691e-06)
Post-Hoc Dunn's Test Result for GCSE:
                     A-Level  Diploma at level 3   Level 3  \
A-Level              1.000000            1.000000  0.000586
Diploma at level 3   1.000000            1.000000  1.000000
Level 3              0.000586            1.000000  1.000000
Other Qualification  1.000000            1.000000  1.000000
Unknown              0.000002            0.080182  0.078255

                     Other Qualification   Unknown
A-Level                              1.0  0.000002
Diploma at level 3                   1.0  0.080182
Level 3                              1.0  0.078255
Other Qualification                  1.0  1.000000
Unknown                              1.0  1.000000
```

In [ ]:

In [1]:
```
import pandas as pd

a_level_df = pd.read_csv('df_a_level.csv')
gcse_df = pd.read_csv('df_gcse.csv')
```

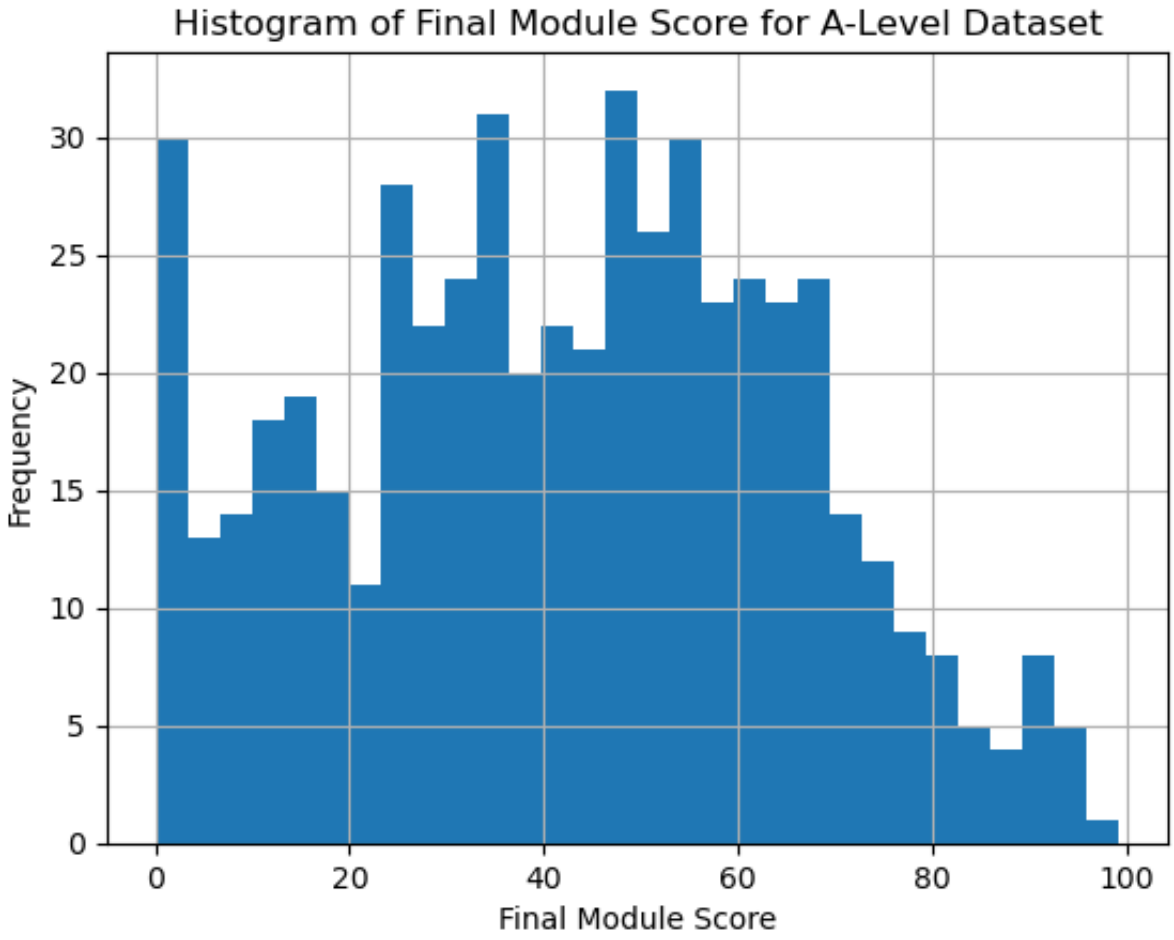
In [2]:
```
a_level_missing_final_score = a_level_df['Final Module Score'].isna().sum
gcse_missing_final_score = gcse_df['Final Module Score'].isna().sum()
```

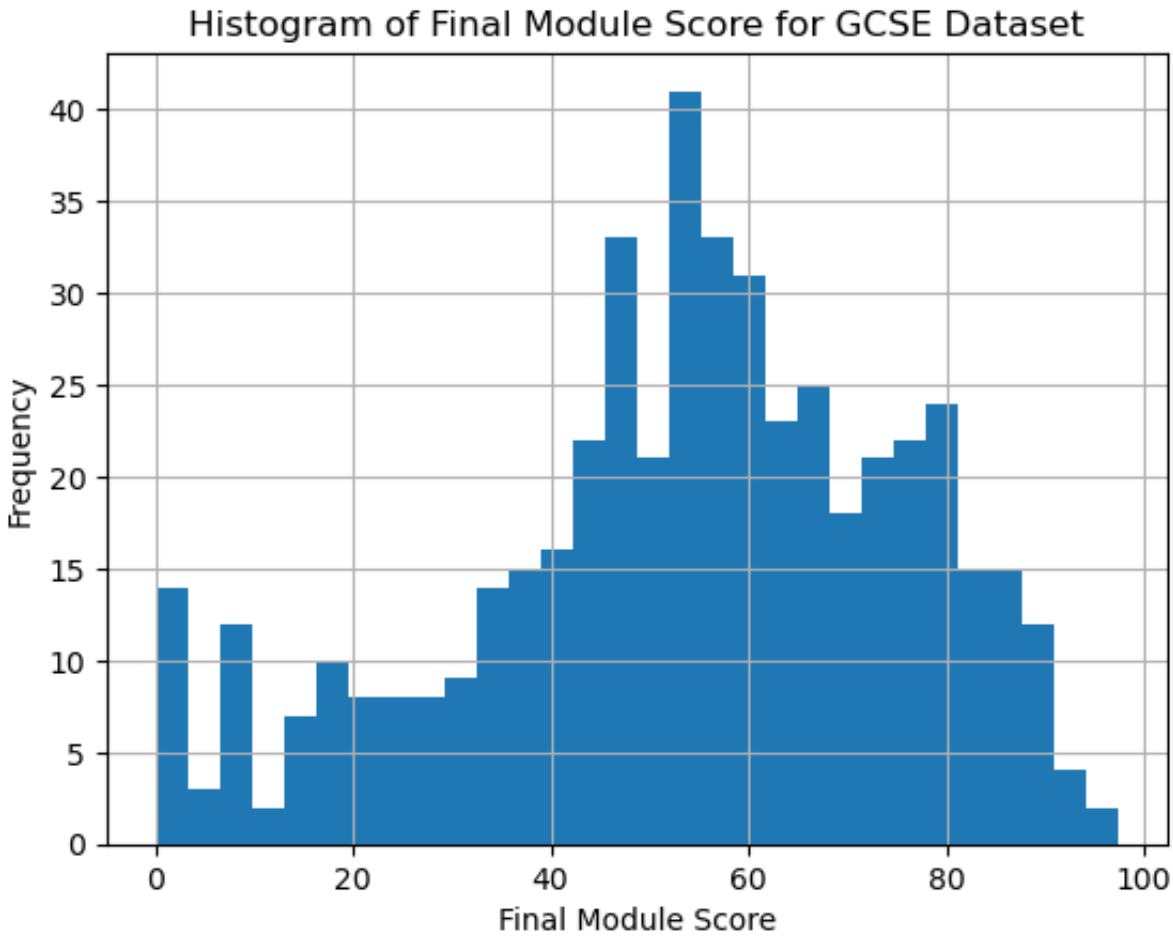
In [3]:
```
a_level_final_score_stats = a_level_df['Final Module Score'].describe()
gcse_final_score_stats = gcse_df['Final Module Score'].describe()
```

In [4]:
```
print("Missing Values in 'Final Module Score' for A-Level Dataset:", a_le
print("Missing Values in 'Final Module Score' for GCSE Dataset:", gcse_mi
print("\nA-Level 'Final Module Score' Statistics:\n", a_level_final_score
print("\nGCSE 'Final Module Score' Statistics:\n", gcse_final_score_stats
```

```
Missing Values in 'Final Module Score' for A-Level Dataset: 0
Missing Values in 'Final Module Score' for GCSE Dataset: 0

A-Level 'Final Module Score' Statistics:
 count    536.000000
mean      42.270597
std       23.598655
min        0.000000
25%       24.845000
50%       43.140000
75%       59.860000
max       99.200000
Name: Final Module Score, dtype: float64

GCSE 'Final Module Score' Statistics:
 count    488.000000
mean      53.627520
std       22.222399
min        0.000000
25%       41.665000
50%       55.250000
75%       69.782500
max       97.500000
Name: Final Module Score, dtype: float64
```

In [5]:
```
import matplotlib.pyplot as plt

# Histogram for A-Level Dataset
a_level_df['Final Module Score'].hist(bins=30)
plt.title('Histogram of Final Module Score for A-Level Dataset')
plt.xlabel('Final Module Score')
plt.ylabel('Frequency')
plt.show()

# Histogram for GCSE Dataset
gcse_df['Final Module Score'].hist(bins=30)
plt.title('Histogram of Final Module Score for GCSE Dataset')
plt.xlabel('Final Module Score')
plt.ylabel('Frequency')
plt.show()
```


Histogram of Final Module Score for A-Level Dataset
Frequency
Final Module Score
0
5
10
15
20
25
30
0
20
40
60
80
100


Histogram of Final Module Score for GCSE Dataset
Frequency
Final Module Score
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100

In [6]:
```
# Find the top 5 scores
top_5_a_level = a_level_df.sort_values(by='Final Module Score', ascending

# Find the least 5 scores
least_5_a_level = a_level_df.sort_values(by='Final Module Score', ascendi

print("Top 5 Final Module Scores in A-Level Dataset:\n", top_5_a_level)
print("\nLeast 5 Final Module Scores in A-Level Dataset:\n", least_5_a_le
```

```
Top 5 Final Module Scores in A-Level Dataset:
     Student ID                   Ethnicity Ethnicity Summary Gender  \
482        1012  Other Asian or Bangladeshi             Asian      F
166         688  Other Asian or Bangladeshi             Asian      F
270         797                       White             White      M
458         987                       Black             Black      M
432         961                      Indian             Asian      M

    Highest Qual on Entry Qualification Summary Socio-economic classifica
tion  \
482               A-Level         Level 2 Quals                       Unk
nown
166               Unknown                 Other                       Unk
nown
270   Other Qualification                 Other                       Unk
nown
458               A-Level         Level 2 Quals                  Professi
onal
432               A-Level         Level 2 Quals                Middle In
come

    Mature/Young Student Module Code Mathematics Requirement  \
482                YOUNG      ME1MME                 A-Level
166                YOUNG      EC1MCE                 A-Level
270               MATURE      EE1EMA                 A-Level
458                YOUNG      ME1MME                 A-Level
432                YOUNG      ME1MME                 A-Level

                                        Disability   Disability Summa
ry  \
482                 Deaf or have a hearing impairment  Physical disabili
ty
166                                      No disability        No disabili
ty
270  Mental health condition challenge or disorder ...        Mental heal
th
458                                      No disability        No disabili
ty
432                                      No disability        No disabili
ty

     Diagnostic Score  PAL Attendance  Final Module Score
482            10.000               0                99.2
166             5.220               1                95.8
270             3.195               7                95.3
458             5.000               3                94.4
432             2.000               0                94.2

Least 5 Final Module Scores in A-Level Dataset:
     Student ID                   Ethnicity Ethnicity Summary Gender  \
229         755                    Pakistani             Asian      M
```

```
371         900                       White             White      M
290         818                       Black             Black      M
162         684  Other Asian or Bangladeshi             Asian      M
333         861  Other Asian or Bangladeshi             Asian      M

    Highest Qual on Entry Qualification Summary Socio-economic classifica
tion  \
229    Diploma at level 3         Level 3 Quals                      Unk
nown
371               A-Level               A-Level                 Middle In
come
290               A-Level         Level 2 Quals                 Middle In
come
162    Diploma at level 3         Level 3 Quals                      Unk
nown
333               A-Level               A-Level                      Unk
nown

    Mature/Young Student Module Code Mathematics Requirement     Disabili
ty  \
229                YOUNG      EE1EMA                A-Level  No disabili
ty
371                YOUNG      ME1MME                A-Level  No disabili
ty
290                YOUNG      EE1EMA                A-Level  No disabili
ty
162                YOUNG      EC1MCE                A-Level  No disabili
ty
333                YOUNG      ME1MME                A-Level  No disabili
ty

    Disability Summary  Diagnostic Score  PAL Attendance  Final Module Sc
ore
229      No disability             3.665               1
0.0
371      No disability             3.955               3
0.0
290      No disability             6.340               2
0.0
162      No disability             5.750               0
0.0
333      No disability             3.835               0
0.0
```

In [7]:

```
# Find the top 5 scores
top_5_gcse = gcse_df.sort_values(by='Final Module Score', ascending=False

# Find the least 5 scores
least_5_gcse = gcse_df.sort_values(by='Final Module Score', ascending=Tru

print("Top 5 Final Module Scores in GCSE Dataset:\n", top_5_gcse)
print("\nLeast 5 Final Module Scores in GCSE Dataset:\n", least_5_gcse)
```

```
Top 5 Final Module Scores in GCSE Dataset:
      Student ID    Ethnicity Ethnicity Summary Gender Highest Qual on En
try  \
350         514        White             White      F               Unkno
wn
304         467        White             White      M               Level
3
182         345        Black             Black      M               Unkno
```

```
wn
221         384    Pakistani             Asian      M               A-Lev
el
158         321  Other Mixed             Other      M               A-Lev
el

    Qualification Summary Socio-economic classification Mature/Young Stud
ent  \
350                 Other                  Professional               YO
UNG
304         Level 3 Quals                 Middle Income               YO
UNG
182                 Other                  Professional               YO
UNG
221               A-Level                  Professional               YO
UNG
158               A-Level                  Professional               YO
UNG

    Module Code Mathematics Requirement     Disability Disability Summary
\
350      CS1MCP                    GCSE  No disability      No disability
304      CS1MCP                    GCSE  No disability      No disability
182      CS1MCP                    GCSE  No disability      No disability
221      CS1MCP                    GCSE  No disability      No disability
158      CS1MCP                    GCSE  No disability      No disability

     Diagnostic Score  PAL Attendance  Final Module Score Score Type
350              10.0               0               97.50     Actual
304              13.0               1               96.25     Actual
182              13.0               1               94.00     Actual
221               5.0               6               93.00     Actual
158               9.0               7               91.38     Actual

Least 5 Final Module Scores in GCSE Dataset:
      Student ID    Ethnicity Ethnicity Summary Gender Highest Qual on En
try  \
229         392    Pakistani             Asian      M               Level
3
383         548  Other Mixed             Other      F               Level
3
452         647  Other Mixed             Other      F               Level
3
247         410       Indian             Asian      M               Unkno
wn
104         267  Other Mixed             Other      M               Level
3

    Qualification Summary Socio-economic classification Mature/Young Stud
ent  \
229         Level 2 Quals                  Unknown               YO
UNG
383         Level 2 Quals                Professional            YO
UNG
452         Level 2 Quals                  Unknown              MAT
URE
247                 Other                  Professional               YO
UNG
104         Level 3 Quals                  Professional               YO
UNG
```

```
    Module Code Mathematics Requirement  \
229      CS1MCP                    GCSE
383      CS1MCP                    GCSE
452      PD1EP1                    GCSE
247      CS1MCP                    GCSE
104      CS1MCP                    GCSE

                                          Disability  \
229                                     No disability
383  Learning difficulty such as Dyslexia, Dyspraxi...
452  Mental health condition challenge or disorder ...
247                                     No disability
104                                     No disability

             Disability Summary  Diagnostic Score  PAL Attendance  \
229              No disability             7.250               0
383  Social/Learning disability             3.510               0
452              Mental health             0.000               1
247              No disability             9.865               0
104              No disability             8.845               0

     Final Module Score Score Type
229                 0.0  Predicted
383                 0.0  Predicted
452                 0.0     Actual
247                 0.0  Predicted
104                 0.0  Predicted
```

In [8]:
```
# Filter for rows where the Final Module Score is 0
a_level_score_zero = a_level_df[a_level_df['Final Module Score'] == 0]

# Count the number of such rows
a_level_count_zero = a_level_score_zero.shape[0]

print("Number of Rows with Final Module Score 0 in A-Level Dataset:", a_l
print("Rows with Final Module Score 0 in A-Level Dataset:\n", a_level_sco
```

```
Number of Rows with Final Module Score 0 in A-Level Dataset: 20
Rows with Final Module Score 0 in A-Level Dataset:
      Student ID                    Ethnicity Ethnicity Summary Gender  \
122          124                        White             White      M
162          684  Other Asian or Bangladeshi              Asian      M
229          755                    Pakistani              Asian      M
239          765                        Black              Black      M
251          777                  Other Mixed              Other      M
290          818                        Black              Black      M
309          837  Other Asian or Bangladeshi              Asian      M
316          844                    Pakistani              Asian      M
321          849  Other Asian or Bangladeshi              Asian      M
333          861  Other Asian or Bangladeshi              Asian      M
363          891                        White             White      M
371          900                        White             White      M
372          901                  Other Mixed              Other      M
373          902                       Indian              Asian      M
384          913                        Black              Black      M
388          917                    Pakistani              Asian      M
402          931                  Other Mixed              Other      M
409          938                    Pakistani              Asian      F
506         1036                    Pakistani              Asian      M
521         1051                       Indian              Asian      M
```

	Highest Qual on Entry	Qualification Summary	Socio-economic classification
122	A-Level	Level 2 Quals	Middle In come
162	Diploma at level 3	Level 3 Quals	Unk nown
229	Diploma at level 3	Level 3 Quals	Unk nown
239	A-Level	A-Level	Middle In come
251	Unknown	Blank	Unk nown
290	A-Level	Level 2 Quals	Middle In come
309	Level 3	Level 3 Quals	Working-C lass
316	Diploma at level 3	Level 3 Quals	Professi onal
321	Diploma at level 3	Level 3 Quals	Working-C lass
333	A-Level	A-Level	Unk nown
363	A-Level	A-Level	Middle In come
371	A-Level	A-Level	Middle In come
372	Other Qualification	Other	Unk nown
373	Level 3	Level 3 Quals	Professi onal
384	Diploma at level 3	Level 3 Quals	Middle In come
388	Diploma at level 3	Level 3 Quals	Unk nown
402	Diploma at level 3	Level 3 Quals	Middle In come
409	Level 3	Level 3 Quals	Working-C lass
506	A-Level	A-Level	Unk nown
521	Diploma at level 3	Level 3 Quals	Professi onal

	Mature/Young Student	Module Code	Mathematics Requirement
122	YOUNG	CE1MAT	A-Level
162	YOUNG	EC1MCE	A-Level
229	YOUNG	EE1EMA	A-Level
239	YOUNG	EE1EMA	A-Level
251	YOUNG	EE1EMA	A-Level
290	YOUNG	EE1EMA	A-Level
309	YOUNG	ME1MME	A-Level
316	YOUNG	ME1MME	A-Level
321	YOUNG	ME1MME	A-Level
333	YOUNG	ME1MME	A-Level
363	YOUNG	ME1MME	A-Level
371	YOUNG	ME1MME	A-Level
372	MATURE	ME1MME	A-Level
373	YOUNG	ME1MME	A-Level
384	YOUNG	ME1MME	A-Level
388	YOUNG	ME1MME	A-Level
402	YOUNG	ME1MME	A-Level

409	YOUNG	ME1MME	A-Level	
506	YOUNG	ME1MME	A-Level	
521	YOUNG	ME1MME	A-Level	

	Disability	Disability Summary
122	No disability	No disability
162	No disability	No disability
229	No disability	No disability
239	No disability	No disability
251	No disability	No disability
290	No disability	No disability
309	No disability	No disability
316	No disability	No disability
321	No disability	No disability
333	No disability	No disability
363	No disability	No disability
371	No disability	No disability
372	No disability	No disability
373	No disability	No disability
384	No disability	No disability
388	No disability	No disability
402	No disability	No disability
409	No disability	No disability
506	Long term illness or health condition such as ...	Physical disability
521	No disability	No disability

	Diagnostic Score	PAL Attendance	Final Module Score
122	5.710	0	0.0
162	5.750	0	0.0
229	3.665	1	0.0
239	4.785	0	0.0
251	2.000	0	0.0
290	6.340	2	0.0
309	7.195	0	0.0
316	4.195	0	0.0
321	5.880	0	0.0
333	3.835	0	0.0
363	4.000	2	0.0
371	3.955	3	0.0
372	2.300	0	0.0

```
373             4.205                0                0.0
384             4.040                0                0.0
388             4.955                1                0.0
402             4.440                0                0.0
409             3.440                0                0.0
506             3.720                0                0.0
521             5.000                0                0.0
```

In [9]:
```
# Filter for rows where the Final Module Score is 0
gcse_score_zero = gcse_df[gcse_df['Final Module Score'] == 0]

# Count the number of such rows
gcse_count_zero = gcse_score_zero.shape[0]

print("Number of Rows with Final Module Score 0 in GCSE Dataset:", gcse_c
print("Rows with Final Module Score 0 in GCSE Dataset:\n", gcse_score_zer
```

```
Number of Rows with Final Module Score 0 in GCSE Dataset: 12
Rows with Final Module Score 0 in GCSE Dataset:
      Student ID    Ethnicity Ethnicity Summary Gender Highest Qual on En
try  \
12          175        White             White      F               Level
3
62          225    Pakistani             Asian      M    Diploma at level
3
104         267  Other Mixed             Other      M               Level
3
110         273        White             White      M               A-Lev
el
129         292       Indian             Asian      M               Unkno
wn
229         392    Pakistani             Asian      M               Level
3
247         410       Indian             Asian      M               Unkno
wn
348         512        White             White      M               Level
3
383         548  Other Mixed             Other      F               Level
3
452         647  Other Mixed             Other      F               Level
3
473         668       Indian             Asian      M               Level
3
479         674        Black             Black      M               Level
3

    Qualification Summary Socio-economic classification Mature/Young Stud
ent  \
12          Level 3 Quals                Working-Class               YO
UNG
62          Level 3 Quals                Working-Class               YO
UNG
104         Level 3 Quals                 Professional               YO
UNG
110               A-Level                      Unknown               YO
UNG
129                 Other                      Unknown               YO
UNG
229         Level 2 Quals                      Unknown               YO
UNG
247                 Other                 Professional               YO
```

```
UNG
348         Level 3 Quals                  Professional                YO
UNG
383         Level 2 Quals                  Professional                YO
UNG
452         Level 2 Quals                       Unknown               MAT
URE
473         Level 2 Quals                       Unknown                YO
UNG
479         Level 2 Quals                 Working-Class                YO
UNG

    Module Code Mathematics Requirement  \
12       CH1MAT                     GCSE
62       CS1MCP                     GCSE
104      CS1MCP                     GCSE
110      CS1MCP                     GCSE
129      CS1MCP                     GCSE
229      CS1MCP                     GCSE
247      CS1MCP                     GCSE
348      CS1MCP                     GCSE
383      CS1MCP                     GCSE
452      PD1EP1                     GCSE
473      PD1EP1                     GCSE
479      PD1EP1                     GCSE

                                        Disability  \
12                                   No disability
62                                   No disability
104                                  No disability
110                                  No disability
129                                  No disability
229                                  No disability
247                                  No disability
348                                  No disability
383  Learning difficulty such as Dyslexia, Dyspraxi...
452  Mental health condition challenge or disorder ...
473                                  No disability
479                                  No disability

            Disability Summary  Diagnostic Score  PAL Attendance  \
12               No disability              6.000               0
62               No disability              8.485               0
104              No disability              8.845               0
110              No disability              9.815               0
129              No disability              9.815               0
229              No disability              7.250               0
247              No disability              9.865               0
348              No disability              7.655               0
383  Social/Learning disability             3.510               0
452              Mental health              0.000               1
473              No disability              4.040               0
479              No disability              2.855               0

     Final Module Score Score Type
12                  0.0     Actual
62                  0.0  Predicted
104                 0.0  Predicted
110                 0.0  Predicted
129                 0.0  Predicted
229                 0.0  Predicted
```

```
247                 0.0  Predicted
348                 0.0  Predicted
383                 0.0  Predicted
452                 0.0     Actual
473                 0.0  Predicted
479                 0.0  Predicted
```

In [10]:
```python
# Proportion of students with a score of 0
a_level_zero_proportion = (a_level_df['Final Module Score'] == 0).mean()
gcse_zero_proportion = (gcse_df['Final Module Score'] == 0).mean()

print("Proportion of zero scores in A-Level:", a_level_zero_proportion)
print("Proportion of zero scores in GCSE:", gcse_zero_proportion)
```

```
Proportion of zero scores in A-Level: 0.037313432835820895
Proportion of zero scores in GCSE: 0.024590163934426230
```

In [11]:
```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import GradientBoostingClassifier

# Load the datasets
a_level_df = pd.read_csv('df_a_level.csv')
gcse_df = pd.read_csv('df_gcse.csv')

# Function to encode, train, predict, and return the predictions
def prepare_train_predict(df):
    # Encode categorical columns
    le = LabelEncoder()
    for col in df.select_dtypes(include=['object']).columns:
        df[col] = le.fit_transform(df[col])

    # Splitting the data into features and labels
    X = df.drop('Final Module Score', axis=1)
    y = (df['Final Module Score'] == 0).astype(int)

    # Splitting the dataset into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0

    # Training the Gradient Boosting Classifier
    gb = GradientBoostingClassifier(random_state=42)
    gb.fit(X_train, y_train)

    # Predicting the entire dataset
    predictions = gb.predict(X)
    return predictions

# Preparing, training, and predicting for both datasets
a_level_df['Predicted Zero Score'] = prepare_train_predict(a_level_df.cop
gcse_df['Predicted Zero Score'] = prepare_train_predict(gcse_df.copy())

# Filtering and displaying predicted zero scores
predicted_zeros_a_level = a_level_df[a_level_df['Predicted Zero Score'] =
predicted_zeros_gcse = gcse_df[gcse_df['Predicted Zero Score'] == 1][['St

print("A Level Dataset Predicted Zero Scores:")
print(predicted_zeros_a_level)
print("\nGCSE Dataset Predicted Zero Scores:")
print(predicted_zeros_gcse)
```

```
/Users/pawan/opt/anaconda3/lib/python3.9/site-packages/scipy/__init__.py:
146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for th
is version of SciPy (detected version 1.25.2
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
A Level Dataset Predicted Zero Scores:
     Student ID  Final Module Score  Predicted Zero Score
122         124                0.00                     1
162         684                0.00                     1
239         765                0.00                     1
251         777                0.00                     1
290         818                0.00                     1
309         837                0.00                     1
321         849                0.00                     1
333         861                0.00                     1
363         891                0.00                     1
372         901                0.00                     1
388         917                0.00                     1
402         931                0.00                     1
408         937                9.86                     1
409         938                0.00                     1
506        1036                0.00                     1
521        1051                0.00                     1

GCSE Dataset Predicted Zero Scores:
     Student ID  Final Module Score  Predicted Zero Score
12          175                0.00                     1
62          225                0.00                     1
63          226               51.88                     1
110         273                0.00                     1
129         292                0.00                     1
247         410                0.00                     1
348         512                0.00                     1
450         645               50.40                     1
452         647                0.00                     1
```

In [12]:

```
# Calculating the mean and standard deviation of final module scores for

# A Level Dataset
mean_std_a_level = a_level_df.groupby('Ethnicity')['Final Module Score'].

# GCSE Dataset
mean_std_gcse = gcse_df.groupby('Ethnicity')['Final Module Score'].agg(['

mean_std_a_level, mean_std_gcse
```

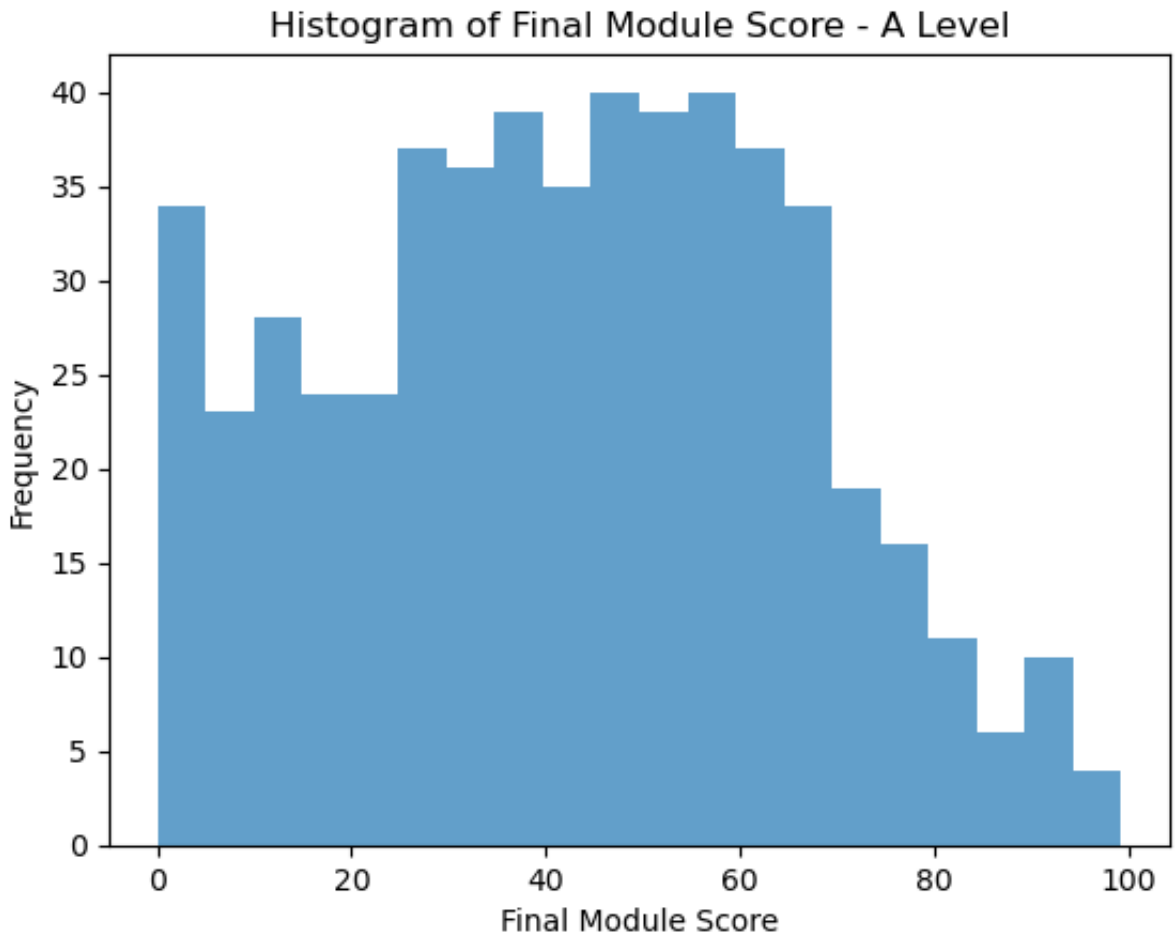
Out[12]:

```
(                                mean        std
 Ethnicity
 Black                       40.192887  23.319754
 Indian                      48.145915  23.883556
 Other Asian or Bangladeshi  43.402278  25.970223
 Other Mixed                 38.283571  22.452560
 Pakistani                   43.118417  21.558248
 White                       41.425412  24.783775,
                                  mean        std
 Ethnicity
 Black                       50.646186  22.022924
 Indian                      56.239872  21.993019
 Other Asian or Bangladeshi  57.321667  21.318482
 Other Mixed                 48.560441  21.253976
 Pakistani                   51.762614  23.730375
 White                       57.467216  21.632465)
```

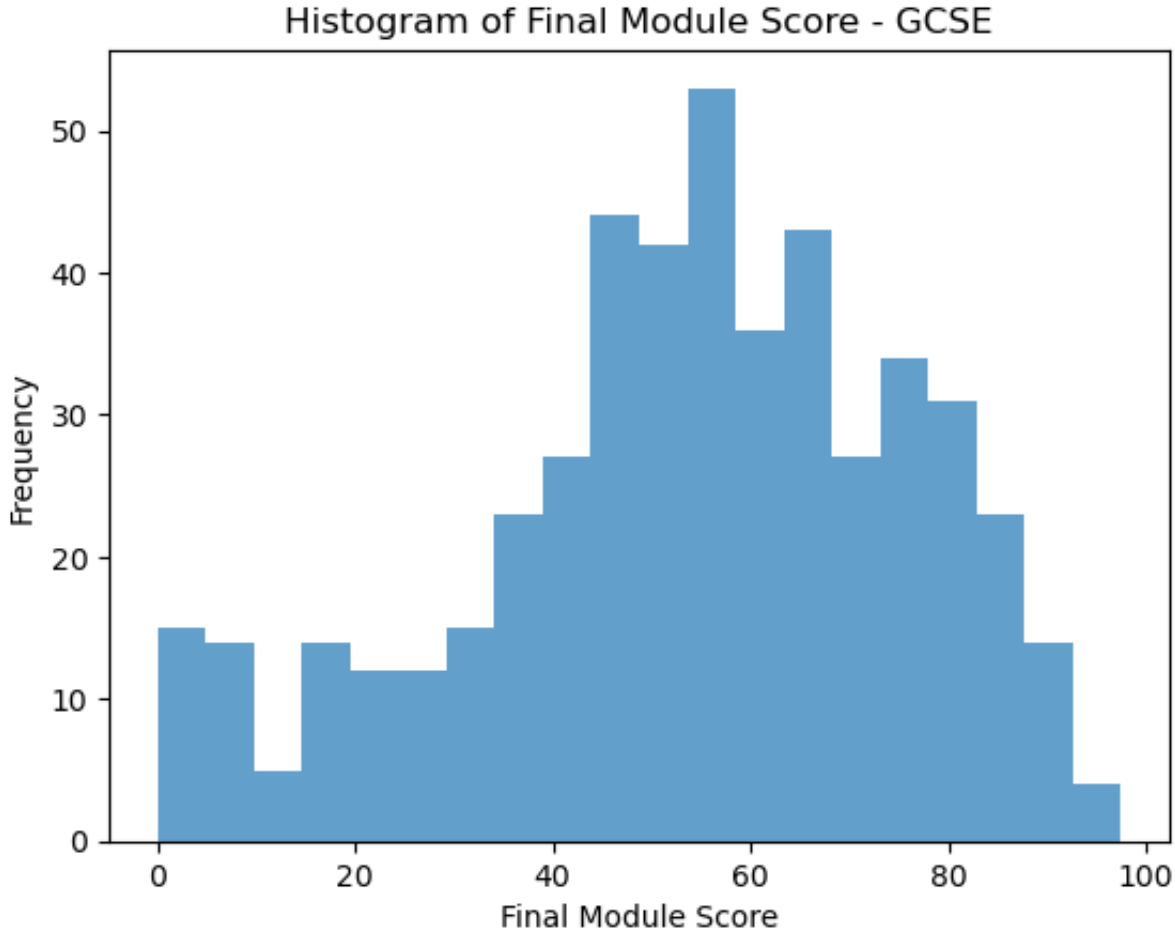
In [13]:
```
# Calculating the ratio of the highest standard deviation to the lowest s

# A Level Dataset
std_ratio_a_level = mean_std_a_level['std'].max() / mean_std_a_level['std

# GCSE Dataset
std_ratio_gcse = mean_std_gcse['std'].max() / mean_std_gcse['std'].min()

std_ratio_a_level, std_ratio_gcse
```

Out[13]:
```
(1.204653719842158, 1.11651462892306)
```

In [14]:
```
import matplotlib.pyplot as plt
from scipy.stats import shapiro

# For A Level Dataset
plt.hist(a_level_df['Final Module Score'].dropna(), bins=20, alpha=0.7)
plt.title('Histogram of Final Module Score - A Level')
plt.xlabel('Final Module Score')
plt.ylabel('Frequency')
plt.show()
stat, p = shapiro(a_level_df['Final Module Score'].dropna())
print('Shapiro-Wilk Test for A Level Dataset:', stat, p)

# For GCSE Dataset
plt.hist(gcse_df['Final Module Score'].dropna(), bins=20, alpha=0.7)
plt.title('Histogram of Final Module Score - GCSE')
plt.xlabel('Final Module Score')
plt.ylabel('Frequency')
plt.show()
stat, p = shapiro(gcse_df['Final Module Score'].dropna())
print('Shapiro-Wilk Test for GCSE Dataset:', stat, p)
```



Shapiro-Wilk Test for A Level Dataset: 0.9820714592933655 3.660389893411775e-06



```
Shapiro-Wilk Test for GCSE Dataset: 0.9690380692481995 1.2435323881732074
e-08
```

In [15]:

```python
from scipy.stats import shapiro

# Function to perform Shapiro-Wilk test for each ethnicity group
def shapiro_test_by_ethnicity(df):
    results = {}
    for ethnicity, group in df.groupby('Ethnicity'):
        stat, p = shapiro(group['Final Module Score'])
        results[ethnicity] = p
    return results

# Performing Shapiro-Wilk test for the A Level dataset
shapiro_results_a_level = shapiro_test_by_ethnicity(a_level_df)

# Performing Shapiro-Wilk test for the GCSE dataset
shapiro_results_gcse = shapiro_test_by_ethnicity(gcse_df)

print("Shapiro-Wilk Test Results for A Level Dataset:")
print(shapiro_results_a_level)

print("\nShapiro-Wilk Test Results for GCSE Dataset:")
print(shapiro_results_gcse)
```

```
Shapiro-Wilk Test Results for A Level Dataset:
{'Black': 0.10487238317728043, 'Indian': 0.2659015357494354, 'Other Asian
or Bangladeshi': 0.052379485219717026, 'Other Mixed': 0.18099330365657806
, 'Pakistani': 0.1087779775261879, 'White': 0.00888199731707573}

Shapiro-Wilk Test Results for GCSE Dataset:
{'Black': 0.007980705238878727, 'Indian': 0.0016874434659257531, 'Other A
sian or Bangladeshi': 0.05092031508684158, 'Other Mixed': 0.5684106349945
068, 'Pakistani': 0.015602825209498405, 'White': 0.002169508021324873}
```

Final Report on Statistical Test Selection for A Level and GCSE Datasets

A Level Dataset

- **Shapiro-Wilk Test Results for Normality**:
  - Black: p = 0.105 (normal)
  - Indian: p = 0.266 (normal)
  - Other Asian or Bangladeshi: p = 0.052 (borderline)
  - Other Mixed: p = 0.181 (normal)
  - Pakistani: p = 0.109 (normal)
  - White: p = 0.009 (non-normal)

GCSE Dataset

- **Shapiro-Wilk Test Results for Normality**:
  - Black: p = 0.008 (non-normal)
  - Indian: p = 0.002 (non-normal)
  - Other Asian or Bangladeshi: p = 0.051 (borderline)
  - Other Mixed: p = 0.568 (normal)
  - Pakistani: p = 0.016 (non-normal)
  - White: p = 0.002 (non-normal)

Conclusion

- For the A Level dataset, the Kruskal-Wallis H test is suggested due to the presence of non-normal distributions in a key group.
- For the GCSE dataset, the Kruskal-Wallis H test is clearly the more suitable choice, given the non-normal distributions in most ethnic groups.

In [16]:
```
from scipy.stats import kruskal

# Performing Kruskal-Wallis H test on 'Final Module Score' across differe

# A Level Dataset
ethnic_groups_a_level = [group['Final Module Score'].values for name, gro
kruskal_result_a_level = kruskal(*ethnic_groups_a_level)

# GCSE Dataset
ethnic_groups_gcse = [group['Final Module Score'].values for name, group
kruskal_result_gcse = kruskal(*ethnic_groups_gcse)

kruskal_result_a_level, kruskal_result_gcse
```

Out[16]:
```
(KruskalResult(statistic=7.759170879944651, pvalue=0.17001717153632898),
 KruskalResult(statistic=13.555856514641402, pvalue=0.018690947359348144)
)
```

In [17]:
```
from scipy.stats import mannwhitneyu
import itertools

# Function to perform post-hoc Mann-Whitney U tests for pairwise comparis
def post_hoc_mann_whitney(groups):
    comparisons = list(itertools.combinations(range(len(groups)), 2))
    results = {}
    for i, j in comparisons:
        stat, p = mannwhitneyu(groups[i], groups[j])
        results[f'Group {i+1} vs Group {j+1}'] = p
    return results

# Performing post-hoc tests for the GCSE dataset
post_hoc_results_gcse = post_hoc_mann_whitney(ethnic_groups_gcse)

post_hoc_results_gcse
```

Out[17]:
```
{'Group 1 vs Group 2': 0.05838985084299266,
 'Group 1 vs Group 3': 0.061556738572525575,
 'Group 1 vs Group 4': 0.3803191578138416,
 'Group 1 vs Group 5': 0.5909385210599207,
 'Group 1 vs Group 6': 0.017507357896815144,
 'Group 2 vs Group 3': 0.8416993495485349,
 'Group 2 vs Group 4': 0.017062643483758688,
 'Group 2 vs Group 5': 0.21346676918014829,
 'Group 2 vs Group 6': 0.781250637954695,
 'Group 3 vs Group 4': 0.01674310069696359,
 'Group 3 vs Group 5': 0.18877060875532192,
 'Group 3 vs Group 6': 0.8865325412356363,
 'Group 4 vs Group 5': 0.2632824032407124,
 'Group 4 vs Group 6': 0.004079261918014067,
 'Group 5 vs Group 6': 0.10716593872228343}
```

A Level Dataset

- **Kruskal-Wallis H Test Results**:
  - Statistic: 7.759
  - p-value: 0.170
- **Interpretation**:
  - The p-value is greater than 0.05, suggesting no significant differences in the median final module scores across different ethnicities. Therefore, ethnicity does not appear to have a statistically significant impact on final module scores in the A Level dataset.

GCSE Dataset

- **Kruskal-Wallis H Test Results**:
  - Statistic: 13.556
  - p-value: 0.019
- **Interpretation**:
  - The p-value is less than 0.05, indicating significant differences in the median final module scores across different ethnicities.
- **Post-Hoc Mann-Whitney U Test Results**:
  - Significant differences were found in the following pairwise comparisons ($p < 0.05$):
    - Group 1 (Black) vs Group 6 (White): p-value = 0.018
    - Group 2 (Indian) vs Group 4 (Other Mixed): p-value = 0.017
    - Group 3 (Other Asian or Bangladeshi) vs Group 4 (Other Mixed): p-value = 0.017
    - Group 4 (Other Mixed) vs Group 6 (White): p-value = 0.004
  - No significant differences were found in other pairwise comparisons.
- **Final Conclusion**:
  - In the GCSE dataset, there are significant differences in final module scores between certain ethnic groups. Specifically, the differences between Black and White, Indian and Other Mixed, Other Asian or Bangladeshi and Other Mixed, and Other Mixed and White groups are statistically significant.

In [18]:
```
# Calculating the mean, variance, and standard deviation of final module
stats_a_level = a_level_df.groupby('Socio-economic classification')['Fina
stats_gcse = gcse_df.groupby('Socio-economic classification')['Final Modu

stats_a_level, stats_gcse
```

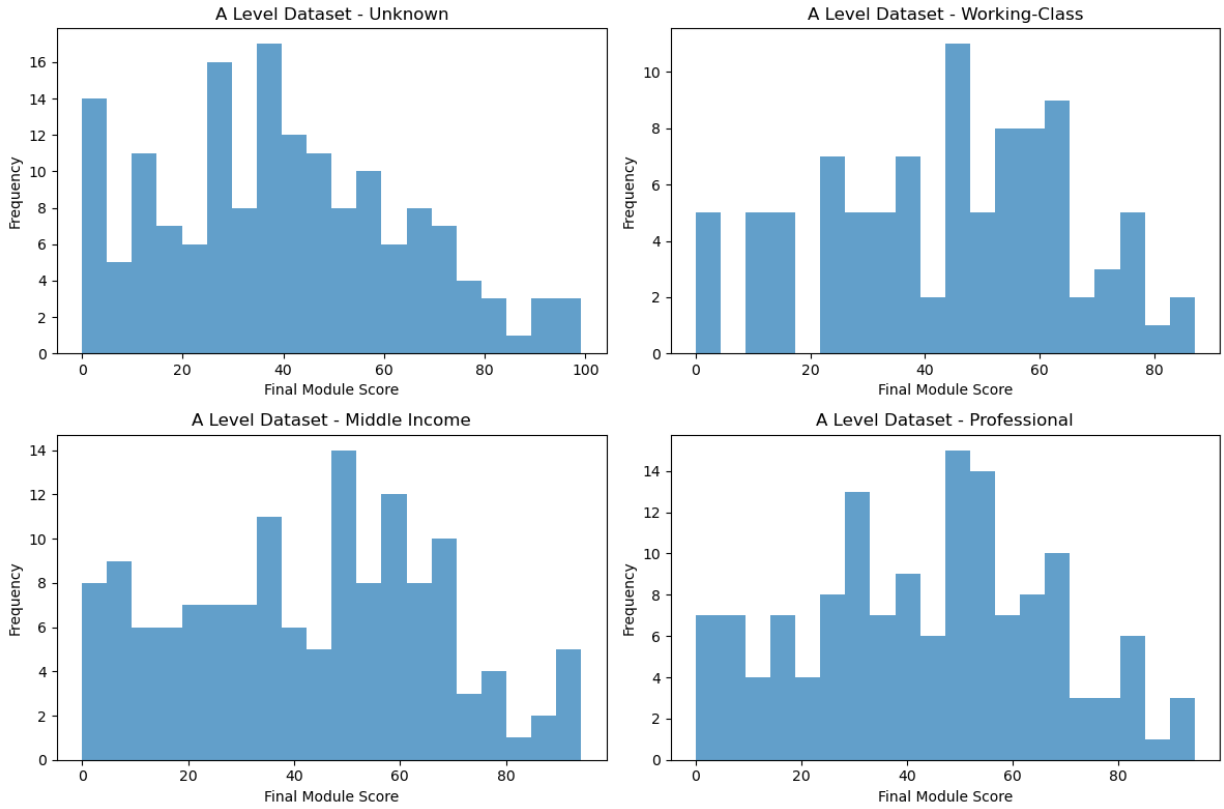
Out[18]:
```
(                                    mean         var        std
 Socio-economic classification
 Middle Income                  42.503165  597.705557  24.448017
 Professional                   43.711338  537.699626  23.188351
 Unknown                        39.893625  592.643538  24.344271
 Working-Class                  43.780105  467.952756  21.632216,
                                     mean         var        std
 Socio-economic classification
 Middle Income                  57.171951  382.831913  19.566091
 Professional                   53.721565  463.103191  21.519832
 Unknown                        52.564671  561.847158  23.703315
 Working-Class                  50.717041  552.694310  23.509451)
```

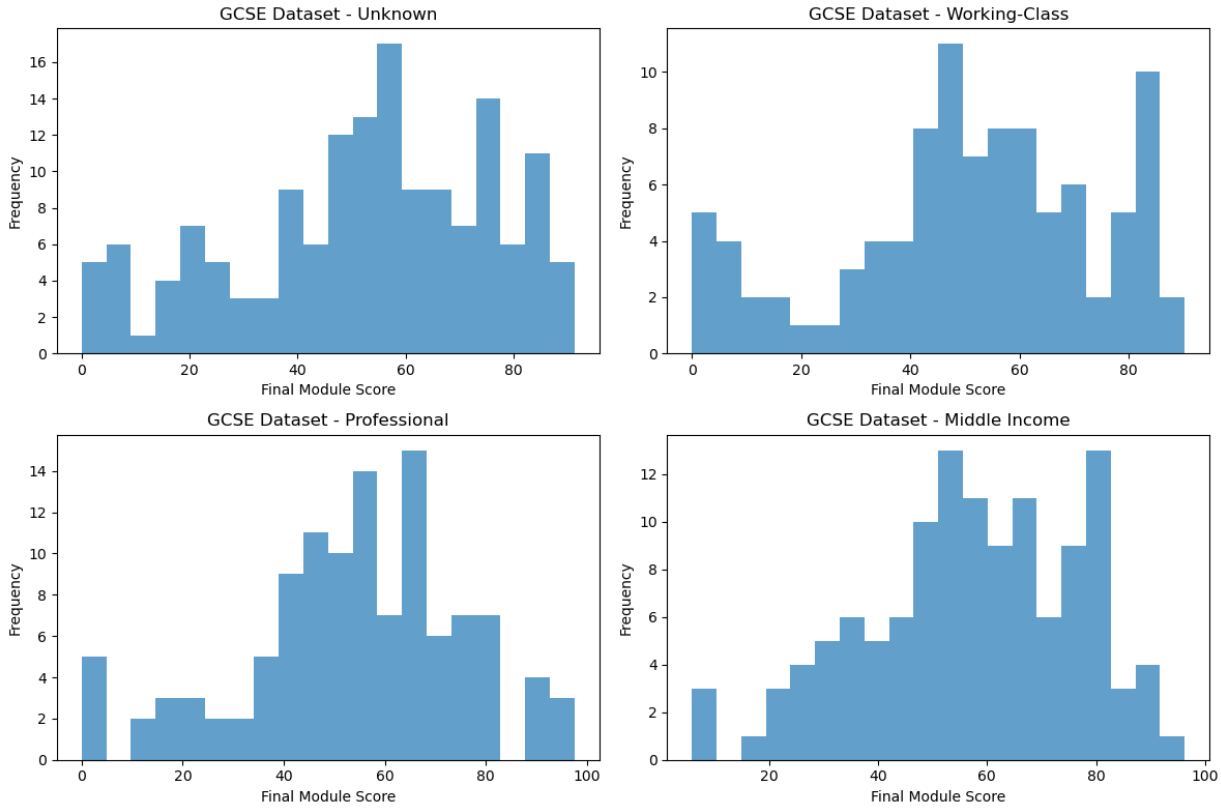
In [19]:
```
from scipy.stats import shapiro

# Calculating the ratio of the highest standard deviation to the lowest s
sd_ratio_a_level = stats_a_level['std'].max() / stats_a_level['std'].min(
sd_ratio_gcse = stats_gcse['std'].max() / stats_gcse['std'].min()

# Shapiro-Wilk test for normality of 'Final Module Score' for each socio-
shapiro_a_level = a_level_df.groupby('Socio-economic classification')['Fi
shapiro_gcse = gcse_df.groupby('Socio-economic classification')['Final Mo

sd_ratio_a_level, sd_ratio_gcse, shapiro_a_level, shapiro_gcse
```

Out[19]:
```
(1.1301670512254767,
 1.2114486979320016,
 Socio-economic classification
 Middle Income    0.009351
 Professional     0.056693
 Unknown          0.008127
 Working-Class    0.083446
 Name: Final Module Score, dtype: float64,
 Socio-economic classification
 Middle Income    0.059543
 Professional     0.012959
 Unknown          0.000101
 Working-Class    0.002270
 Name: Final Module Score, dtype: float64)
```

In [20]:

```
import matplotlib.pyplot as plt

# Plot histograms for each socio-economic class in the dataset
def plot_histograms_by_class(df, column_name, title_prefix):
    socio_economic_classes = df['Socio-economic classification'].unique()
    plt.figure(figsize=(12, 8))

    for i, sec_class in enumerate(socio_economic_classes, 1):
        plt.subplot(2, 2, i)
        plt.hist(df[df['Socio-economic classification'] == sec_class][col
        plt.title(f'{title_prefix} - {sec_class}')
        plt.xlabel(column_name)
        plt.ylabel('Frequency')

    plt.tight_layout()
    plt.show()

# Plotting histograms for A Level Dataset
plot_histograms_by_class(a_level_df, 'Final Module Score', 'A Level Datas

# Plotting histograms for GCSE Dataset
plot_histograms_by_class(gcse_df, 'Final Module Score', 'GCSE Dataset')
```





Conclusion

A Level Dataset

- **Histogram Analysis**: The histograms for different socio-economic classes in the A Level dataset show varying distributions. While some classes appear to have a somewhat normal distribution, others exhibit skewness or non-normal characteristics.
- **Statistical Test Suitability**: Considering the variance ratio is less than 2 but the Shapiro-Wilk test indicated non-normal distributions in some classes, the use of ANOVA is questionable. A non-parametric test like the Kruskal-Wallis H test might be more appropriate for comparing final module scores across socio-economic classes.

GCSE Dataset

- **Histogram Analysis**: Similar to the A Level dataset, the histograms for the GCSE dataset show diverse distributions across socio-economic classes, with several classes exhibiting non-normal distribution patterns.
- **Statistical Test Suitability**: The variance ratio is less than 2; however, the Shapiro-Wilk test results show non-normal distributions in most classes. This suggests that the Kruskal-Wallis H test is a more suitable choice than ANOVA for analyzing differences in final module scores across socio-economic classes.

In summary, for both datasets, the Kruskal-Wallis H test is recommended for analyzing the final module scores across socio-economic classes due to the presence of non-normal distributions in several classes.

In [21]:
```
# Performing Kruskal-Wallis H test on 'Final Module Score' across differe

# A Level Dataset
socio_economic_groups_a_level = [group['Final Module Score'].values for n
kruskal_result_a_level = kruskal(*socio_economic_groups_a_level)

# GCSE Dataset
socio_economic_groups_gcse = [group['Final Module Score'].values for name
kruskal_result_gcse = kruskal(*socio_economic_groups_gcse)

kruskal_result_a_level, kruskal_result_gcse
```

Out[21]:
```
(KruskalResult(statistic=3.1529122424353253, pvalue=0.36864467221226577),
 KruskalResult(statistic=3.81914739647755, pvalue=0.2816668091520404))
```

Kruskal-Wallis H Test Report

A Level Dataset

- **Test Results**:
  - Statistic: 3.153
  - p-value: 0.369
- **Interpretation**:
  - The p-value is greater than 0.05, indicating no significant differences in the median final module scores across different socio-economic classes. This suggests that socio-economic classification does not have a statistically significant impact on final module scores in the A Level dataset.

GCSE Dataset

- **Test Results**:
  - Statistic: 3.819
  - p-value: 0.282
- **Interpretation**:
  - With a p-value greater than 0.05, the null hypothesis cannot be rejected. This indicates no statistically significant differences in the median final module scores across different socio-economic classes in the GCSE dataset.

Conclusion

- For both the A Level and GCSE datasets, the Kruskal-Wallis H test results suggest that socio-economic classification does not significantly affect the final module scores. These findings indicate that students' socio-economic backgrounds do not lead to statistically significant differences in their academic performance, as measured by final module scores, in both datasets.

In [22]:
```
import pandas as pd
from scipy.stats import kruskal

# Dropping null values from the 'Final Module Score' column in both datas
a_level_df_clean = a_level_df.dropna(subset=['Final Module Score'])
gcse_df_clean = gcse_df.dropna(subset=['Final Module Score'])

# Performing Kruskal-Wallis H test on 'Final Module Score' across differe
gender_groups_a_level = [group['Final Module Score'].values for name, gro
kruskal_result_a_level_gender = kruskal(*gender_groups_a_level)

gender_groups_gcse = [group['Final Module Score'].values for name, group
kruskal_result_gcse_gender = kruskal(*gender_groups_gcse)

kruskal_result_a_level_gender, kruskal_result_gcse_gender
```

Out[22]:
```
(KruskalResult(statistic=2.9621955948559195, pvalue=0.08523213194442986),
 KruskalResult(statistic=0.33402234839023315, pvalue=0.5633001279754789))
```

Kruskal-Wallis H Test Report on the Impact of Gender

A Level Dataset

- **Test Results**:
  - Statistic: 2.962
  - p-value: 0.085
- **Interpretation**:
  - The p-value is slightly above 0.05, indicating no significant differences in the median final module scores between different genders. However, the result is somewhat borderline, suggesting a trend that might warrant further investigation.

GCSE Dataset

- **Test Results**:
  - Statistic: 0.334
  - p-value: 0.563
- **Interpretation**:
  - With a p-value well above 0.05, there is no evidence of statistically significant differences in median final module scores between genders.

Conclusion

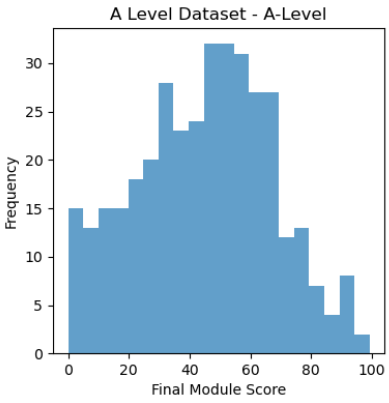
- For both the A Level and GCSE datasets, the Kruskal-Wallis H test results suggest that gender does not significantly affect the final module scores. This indicates that male and female students perform similarly in terms of their final module scores in both educational settings.

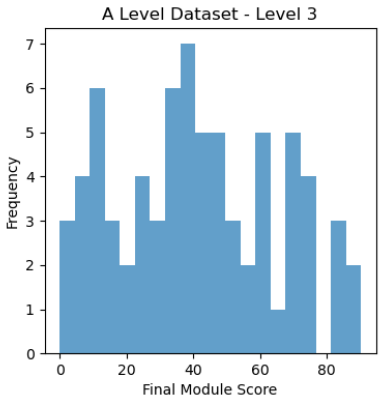
In [23]:
```
# Calculating the mean, variance, and standard deviation of final module
stats_highest_qual_a_level = a_level_df.groupby('Highest Qual on Entry')[
stats_highest_qual_gcse = gcse_df.groupby('Highest Qual on Entry')['Final

# Calculating the ratio of the highest standard deviation to the lowest s
sd_ratio_highest_qual_a_level = stats_highest_qual_a_level['std'].max() /
sd_ratio_highest_qual_gcse = stats_highest_qual_gcse['std'].max() / stats

stats_highest_qual_a_level, stats_highest_qual_gcse, sd_ratio_highest_qua
```

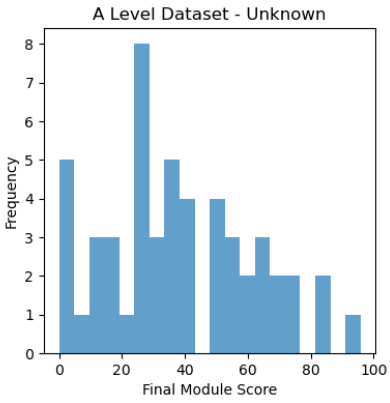
Out[23]:
```
(                           mean          var        std
 Highest Qual on Entry
 A-Level                44.840792   508.390450  22.547515
 Diploma at level 3     22.522703   406.322004  20.157430
 Level 3                41.317123   575.231649  23.983987
 Other Qualification    49.052500  1341.451879  36.625836
 Unknown                38.526923   567.305645  23.818179,
                            mean         var        std
 Highest Qual on Entry
 A-Level                60.726625  583.765785  24.161246
 Diploma at level 3     39.829714  370.283009  19.242739
 Level 3                54.918388  435.107629  20.859234
 Other Qualification    51.000000   72.000000   8.485281
 Unknown                46.580149  535.422583  23.139200,
 1.8169893355644064,
 2.847430090517593)
```

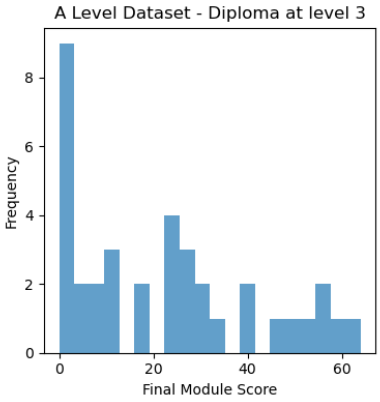
In [24]:
```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import shapiro

# Dropping null values from the 'Final Module Score' column in both datas
a_level_df_clean = a_level_df.dropna(subset=['Final Module Score'])
gcse_df_clean = gcse_df.dropna(subset=['Final Module Score'])

# Function to plot histograms and perform Shapiro-Wilk test
def plot_and_test(df, title_prefix):
    qualifications = df['Highest Qual on Entry'].unique()
    plt.figure(figsize=(12, 8))

    shapiro_results = {}
    for i, qual in enumerate(qualifications, 1):
        # Selecting data for the qualification
        data = df[df['Highest Qual on Entry'] == qual]['Final Module Scor

        # Check if there are at least 3 data points
        if len(data) >= 3:
            # Perform Shapiro-Wilk test
            stat, p = shapiro(data)
            shapiro_results[qual] = p
        else:
            # Not enough data for Shapiro-Wilk test
            shapiro_results[qual] = 'Insufficient data'

        # Plotting
        plt.subplot(2, 3, i)
        plt.hist(data, bins=20, alpha=0.7)
        plt.title(f'{title_prefix} - {qual}')
        plt.xlabel('Final Module Score')
        plt.ylabel('Frequency')

    plt.tight_layout()
    plt.show()
    return shapiro_results

# Plotting and testing for A Level Dataset
print("A Level Dataset:")
shapiro_results_a_level = plot_and_test(a_level_df_clean, 'A Level Datase

# Plotting and testing for GCSE Dataset
print("\nGCSE Dataset:")
shapiro_results_gcse = plot_and_test(gcse_df_clean, 'GCSE Dataset')

print("\nShapiro-Wilk Test Results for A Level Dataset:")
print(shapiro_results_a_level)

print("\nShapiro-Wilk Test Results for GCSE Dataset:")
print(shapiro_results_gcse)
```

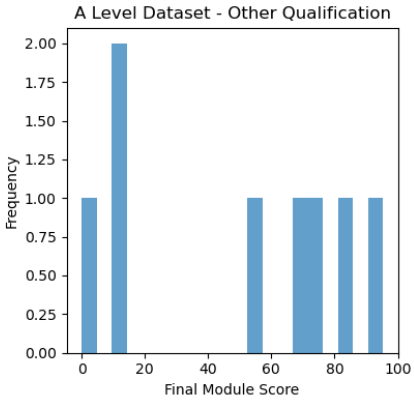
```
A Level Dataset:
```



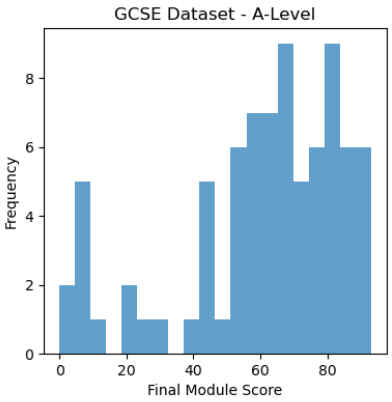


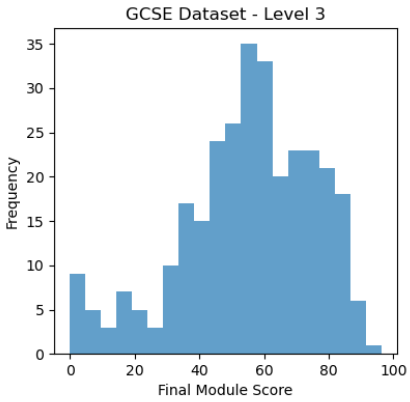


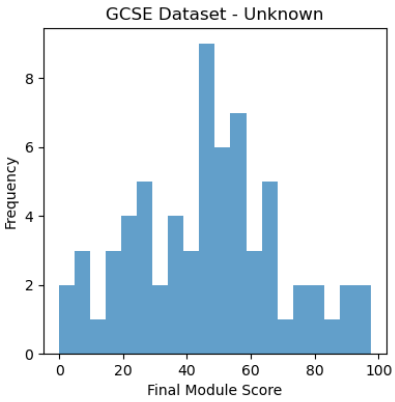


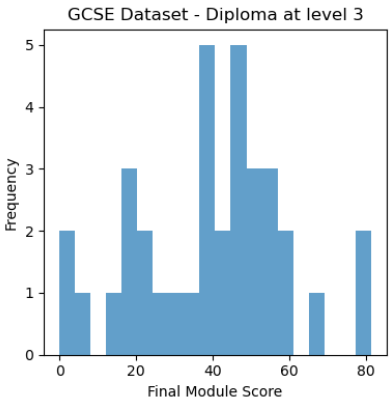


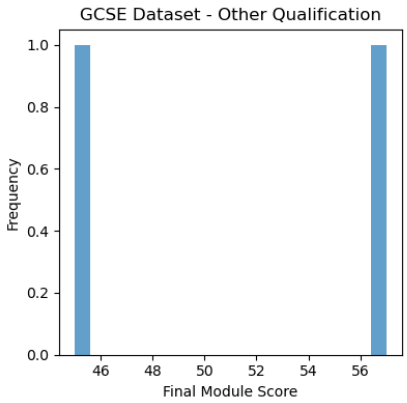
GCSE Dataset:











Shapiro-Wilk Test Results for A Level Dataset:
{'A-Level': 0.0019961853977292776, 'Level 3': 0.08654621243476868, 'Unknown': 0.262520432472229, 'Diploma at level 3': 0.003371906466782093, 'Other Qualification': 0.24406050145626068}

Shapiro-Wilk Test Results for GCSE Dataset:
{'A-Level': 6.219619081093697e-06, 'Level 3': 4.056713862610195e-07, 'Unknown': 0.7276175022125244, 'Diploma at level 3': 0.6597834825515747, 'Other Qualification': 'Insufficient data'}

Shapiro-Wilk Test Results for Normality by Highest Qualification on Entry

A Level Dataset

- **Highest Qualification on Entry**:
  - **A-Level**: p-value = 0.002 (non-normal distribution)
  - **Level 3**: p-value = 0.087 (normal distribution)
  - **Unknown**: p-value = 0.263 (normal distribution)
  - **Diploma at level 3**: p-value = 0.003 (non-normal distribution)
  - **Other Qualification**: p-value = 0.244 (normal distribution)

GCSE Dataset

- **Highest Qualification on Entry**:
  - **A-Level**: p-value < 0.001 (non-normal distribution)
  - **Level 3**: p-value < 0.001 (non-normal distribution)
  - **Unknown**: p-value = 0.728 (normal distribution)
  - **Diploma at level 3**: p-value = 0.660 (normal distribution)
  - **Other Qualification**: Insufficient data

Interpretation and Appropriate Statistical Tests

- **A Level Dataset**:
  - With mixed normality across different qualifications, the choice between ANOVA and the Kruskal-Wallis H test depends on the specific research focus. Given the non-normal distribution in key categories like 'A-Level' and 'Diploma at level 3', the Kruskal-Wallis H test might be more suitable for comparing median scores across qualifications.
- **GCSE Dataset**:
  - Due to the non-normal distribution in major categories like 'A-Level' and 'Level 3', and insufficient data in 'Other Qualification', the Kruskal-Wallis H test is recommended. This test does not require the normality assumption and is suitable for comparing median scores.

In [25]:
```
# Performing Kruskal-Wallis H test on 'Final Module Score' across differe

# Filtering out groups with insufficient data for GCSE dataset
gcse_groups_filtered = [group for group in ethnic_groups_gcse if len(grou

# A Level Dataset
qual_groups_a_level = [group['Final Module Score'].values for name, group
kruskal_result_a_level_qual = kruskal(*qual_groups_a_level)

# GCSE Dataset
qual_groups_gcse = [group['Final Module Score'].values for name, group in
kruskal_result_gcse_qual = kruskal(*qual_groups_gcse)

kruskal_result_a_level_qual, kruskal_result_gcse_qual
```

Out[25]:
```
(KruskalResult(statistic=31.361990339067187, pvalue=2.5825710817390935e-0
6),
 KruskalResult(statistic=37.03031044668887, pvalue=4.5339336052069366e-08
))
```

Kruskal-Wallis H Test Report on the Impact of Highest Qualification on Entry

A Level Dataset

- **Test Results**:
  - Statistic: 31.362
  - p-value: ~2.58e-06
- **Interpretation**:
  - The p-value is significantly below 0.05, indicating statistically significant differences in the median final module scores across different qualifications. This suggests that the highest qualification on entry has a substantial impact on final module scores in the A Level dataset.

GCSE Dataset

- **Test Results**:
  - Statistic: 37.030
  - p-value: ~4.53e-08
- **Interpretation**:
  - With a p-value well below 0.05, there are statistically significant differences in the median final module scores across different qualifications. This indicates that the highest qualification on entry plays a significant role in final module scores in the GCSE dataset.

Conclusion

- For both the A Level and GCSE datasets, the Kruskal-Wallis H test results reveal that the highest qualification on entry is a significant factor influencing students' final module scores. These findings highlight the importance of prior educational background in academic performance in both educational settings.

In [26]:
```
# Performing post-hoc analysis for Kruskal-Wallis H test results using pa

# Function for post-hoc analysis using Mann-Whitney U test
def post_hoc_mann_whitney(groups):
    comparisons = list(itertools.combinations(range(len(groups)), 2))
    results = {}
    for (i, j) in comparisons:
        group_i = groups[i]
        group_j = groups[j]
        stat, p = mannwhitneyu(group_i, group_j)
        results[f'Group {i+1} vs Group {j+1}'] = p
    return results

# Post-hoc analysis for A Level Dataset
post_hoc_results_a_level = post_hoc_mann_whitney(qual_groups_a_level)

# Post-hoc analysis for GCSE Dataset
post_hoc_results_gcse = post_hoc_mann_whitney(qual_groups_gcse)

post_hoc_results_a_level, post_hoc_results_gcse
```

Out[26]:
```
({'Group 1 vs Group 2': 7.700756792033325e-08,
  'Group 1 vs Group 3': 0.2082565106184181,
  'Group 1 vs Group 4': 0.5429952112820073,
  'Group 1 vs Group 5': 0.05621256864845209,
  'Group 2 vs Group 3': 0.00012514314811459226,
  'Group 2 vs Group 4': 0.050911133756158904,
  'Group 2 vs Group 5': 0.0014643287991595924,
  'Group 3 vs Group 4': 0.5061147921671225,
  'Group 3 vs Group 5': 0.4707268486813848,
  'Group 4 vs Group 5': 0.45968011555187194},
 {'Group 1 vs Group 2': 1.8557899626887186e-06,
  'Group 1 vs Group 3': 0.00362673485986927,
  'Group 1 vs Group 4': 6.756580598347704e-05,
  'Group 2 vs Group 3': 2.1284682180087496e-05,
  'Group 2 vs Group 4': 0.14261171119508603,
  'Group 3 vs Group 4': 0.002153425805758618})
```

Final Report on the Impact of Highest Qualification on Entry

A Level Dataset

Kruskal-Wallis H Test Results:

- **Statistic**: 31.362
- **p-value**: ~2.58e-06

Post-Hoc Mann-Whitney U Test Results:

Significant differences found between:

- A-Level vs Diploma at level 3: p-value ≈ 7.70e-08

- Diploma at level 3 vs Level 3: p-value ≈ 0.00013
- Diploma at level 3 vs Unknown: p-value ≈ 0.00146

No significant differences in other pairwise comparisons.

Interpretation:

There are statistically significant differences in final module scores across different qualifications. The post-hoc analysis indicates specific qualifications where these differences are significant.

GCSE Dataset

Kruskal-Wallis H Test Results:

- **Statistic**: 37.030
- **p-value**: ~4.53e-08

Post-Hoc Mann-Whitney U Test Results:

Significant differences found between:

- Group 1 vs Group 2: p-value ≈ 1.86e-06
- Group 1 vs Group 3: p-value ≈ 0.00363
- Group 1 vs Group 4: p-value ≈ 6.76e-05
- Group 2 vs Group 3: p-value ≈ 2.13e-05
- Group 3 vs Group 4: p-value ≈ 0.00215

No significant differences in other pairwise comparisons.

Conclusion

In both the A Level and GCSE datasets, the highest qualification on entry significantly impacts students' final module scores. The post-hoc analysis provides further insights into specific pairs of qualifications where significant differences exist, highlighting the importance of initial educational background in academic performance in both educational settings.

Introduction to the Study on the Effects of Ethnicity on Student Behavior and Outcomes

Background

The influence of ethnicity on educational outcomes and student behavior is a crucial area of study in the realm of educational sociology and psychology. Understanding these effects is vital for developing inclusive educational practices and policies that cater to the diverse needs of the student population. This research aims to delve into how ethnicity, alongside other factors such as socio-economic classification, gender, and highest qualification on entry, affects student behavior and outcomes. The focus is on two key performance indicators: diagnostic scores and final module scores.

Objectives

The primary objectives of this study are to:

- Investigate the impact of ethnicity on diagnostic and final module scores among students.
- Explore the role of socio-economic background, gender, and highest qualification on entry in shaping these educational outcomes.
- Provide insights into whether these factors lead to statistically significant differences in student performance.

Methodology

A comprehensive analysis was conducted on a college dataset, encompassing students from various ethnic backgrounds. The study employed several statistical tests, including:

- Tukey HSD and Dunn's tests, to compare mean diagnostic scores across ethnic groups.
- Kruskal-Wallis H and Mann-Whitney U tests, to evaluate differences in diagnostic and final module scores based on socio-economic classification, gender, and highest qualification on entry.

Summary of Findings

Key findings from the analysis revealed:

- No significant differences in mean diagnostic scores among different ethnic groups in the A-Level dataset, as indicated by the Tukey HSD test.
- Significant differences in diagnostic scores between specific ethnic groups in the GCSE dataset, as shown by Dunn's test.
- Socio-economic classification and gender had varied impacts on diagnostic and final module scores across the A-Level and GCSE datasets.

Conclusion

The findings provide valuable insights into the dynamics of ethnicity, socio-economic background, gender, and educational background in educational settings. The study

contributes to the ongoing discourse on equality and diversity in education, highlighting areas for further research and policy development.

In [ ]:

In [1]:
```
import pandas as pd

df_a_level = pd.read_csv('df_a_level.csv')
df_gcse = pd.read_csv('df_gcse.csv')
```

In [2]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.impute import SimpleImputer
import numpy as np

# Preprocessing for GCSE dataset
# Selecting features and target variable
X_gcse = df_gcse.drop(columns=['Final Module Score', 'Student ID'])
y_gcse = df_gcse['Final Module Score']

# Identifying categorical and numerical columns
categorical_cols = X_gcse.select_dtypes(include=['object', 'category']).c
numerical_cols = X_gcse.select_dtypes(include=['int64', 'float64']).colum

# Creating a column transformer for preprocessing
preprocessor = ColumnTransformer(
    transformers=[
        ('num', SimpleImputer(strategy='mean'), numerical_cols),
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_cols)
    ])

# Splitting the GCSE dataset into training and test sets
X_train_gcse, X_test_gcse, y_train_gcse, y_test_gcse = train_test_split(X

# Selecting the model - RandomForestRegressor
model_gcse = Pipeline(steps=[('preprocessor', preprocessor),
                             ('model', RandomForestRegressor(n_estimators

# Training the model
model_gcse.fit(X_train_gcse, y_train_gcse)

# Predicting on test set
y_pred_gcse = model_gcse.predict(X_test_gcse)

# Evaluating the model
mse_gcse = mean_squared_error(y_test_gcse, y_pred_gcse)
rmse_gcse = np.sqrt(mse_gcse)

mse_gcse, rmse_gcse
```

```
/Users/pawan/opt/anaconda3/lib/python3.9/site-packages/scipy/__init__.py:
146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for th
is version of SciPy (detected version 1.25.2
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Out[2]:
```
(486.74788526032523, 22.062363546554238)
```

In [3]:
```
# Predicting final scores for the entire GCSE dataset
df_gcse['Predicted Final Module Score'] = model_gcse.predict(X_gcse)

# Display the first few rows of the updated dataframe
df_gcse.head()
```

Out[3]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature/Yo Stu
0	163	Pakistani	Asian	M	A-Level	A-Level	Unknown	MAT
1	164	Black	Black	M	A-Level	A-Level	Unknown	YO
2	165	Pakistani	Asian	F	Level 3	Level 3 Quals	Unknown	YO
3	166	Black	Black	F	A-Level	A-Level	Working-Class	YO
4	167	White	White	M	Level 3	Level 2 Quals	Professional	YO

In [4]:
```
# Preprocessing for A-Level dataset
# Selecting features and target variable
X_a_level = df_a_level.drop(columns=['Final Module Score', 'Student ID'])
y_a_level = df_a_level['Final Module Score']

# Identifying categorical and numerical columns for the A-Level dataset
categorical_cols_a_level = X_a_level.select_dtypes(include=['object', 'ca
numerical_cols_a_level = X_a_level.select_dtypes(include=['int64', 'float

# Creating a column transformer for preprocessing
preprocessor_a_level = ColumnTransformer(
    transformers=[
        ('num', SimpleImputer(strategy='mean'), numerical_cols_a_level),
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_cols_
    ])

# Splitting the A-Level dataset into training and test sets
X_train_a_level, X_test_a_level, y_train_a_level, y_test_a_level = train_

# Selecting the model - RandomForestRegressor for A-Level dataset
model_a_level = Pipeline(steps=[('preprocessor', preprocessor_a_level),
                                ('model', RandomForestRegressor(n_estimat

# Training the model on the A-Level dataset
model_a_level.fit(X_train_a_level, y_train_a_level)

# Predicting on test set of A-Level dataset
y_pred_a_level = model_a_level.predict(X_test_a_level)

# Evaluating the model on the A-Level dataset
mse_a_level = mean_squared_error(y_test_a_level, y_pred_a_level)
rmse_a_level = np.sqrt(mse_a_level)

mse_a_level, rmse_a_level
```

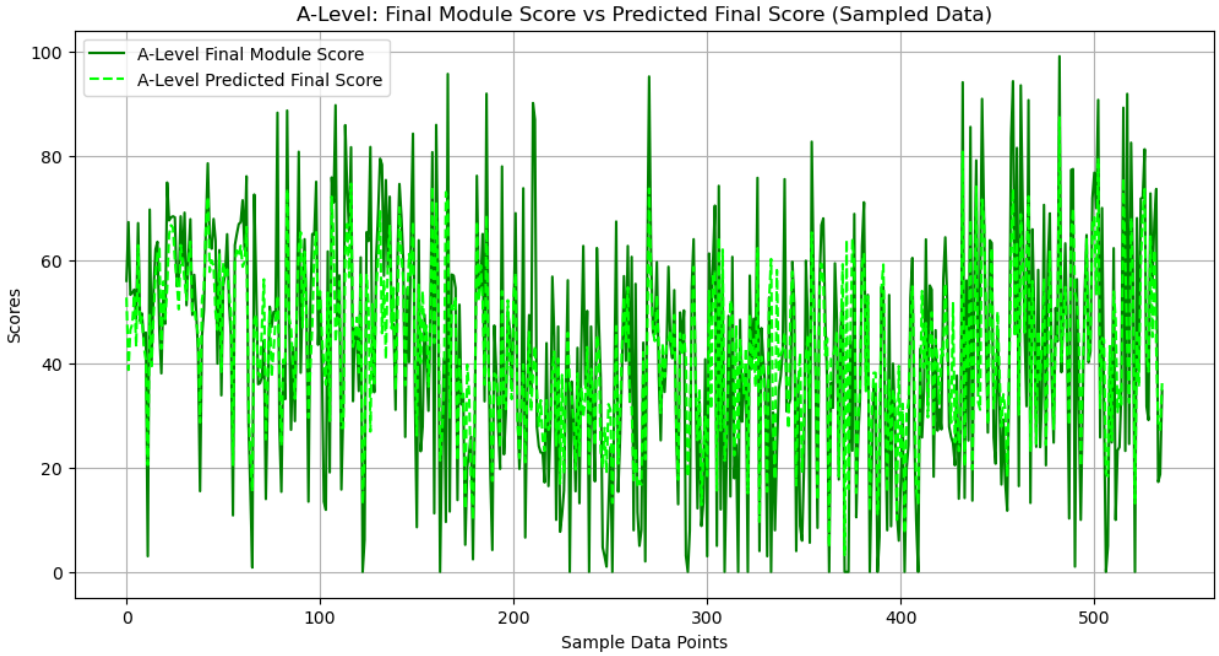
Out[4]: (639.3884884212557, 25.286132334171942)

In [5]:
```
# Predicting final scores for the entire A-Level dataset
df_a_level['Predicted Final Module Score'] = model_a_level.predict(X_a_le

# Display the first few rows of the updated dataframe
df_a_level.head()
```

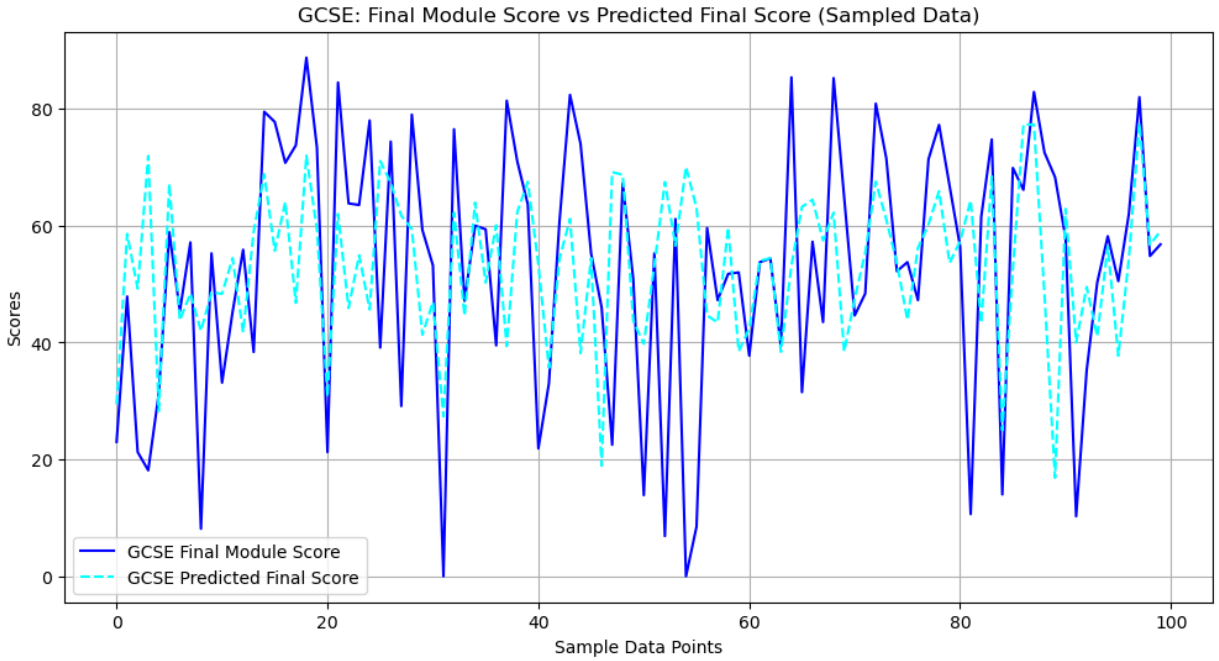
Out[5]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature S
0	1	Other Mixed	Other	M	A-Level	A-Level	Unknown	M
1	2	Other Asian or Bangladeshi	Asian	F	A-Level	A-Level	Working-Class	
2	3	Black	Black	M	A-Level	A-Level	Working-Class	
3	4	Black	Black	F	A-Level	A-Level	Working-Class	
4	5	Other Asian or Bangladeshi	Asian	M	A-Level	A-Level	Unknown	

In [6]:
```
import matplotlib.pyplot as plt
# A-Level dataset plot with fewer data points
plt.figure(figsize=(12, 6))
plt.plot(df_a_level['Final Module Score'].reset_index(drop=True), color='
plt.plot(df_a_level['Predicted Final Module Score'].reset_index(drop=True
plt.title('A-Level: Final Module Score vs Predicted Final Score (Sampled
plt.xlabel('Sample Data Points')
plt.ylabel('Scores')
plt.legend()
plt.grid(True)
plt.show()
```



In [7]:
```
# Reducing the number of data points for clearer visualization
# Taking a sample of 100 data points from each dataset

gcse_sample = df_gcse.sample(n=100, random_state=0)
a_level_sample = df_a_level.sample(n=100, random_state=0)

# GCSE dataset plot with fewer data points
plt.figure(figsize=(12, 6))
plt.plot(gcse_sample['Final Module Score'].reset_index(drop=True), color=
plt.plot(gcse_sample['Predicted Final Module Score'].reset_index(drop=Tru
plt.title('GCSE: Final Module Score vs Predicted Final Score (Sampled Dat
plt.xlabel('Sample Data Points')
plt.ylabel('Scores')
plt.legend()
plt.grid(True)
plt.show()
```

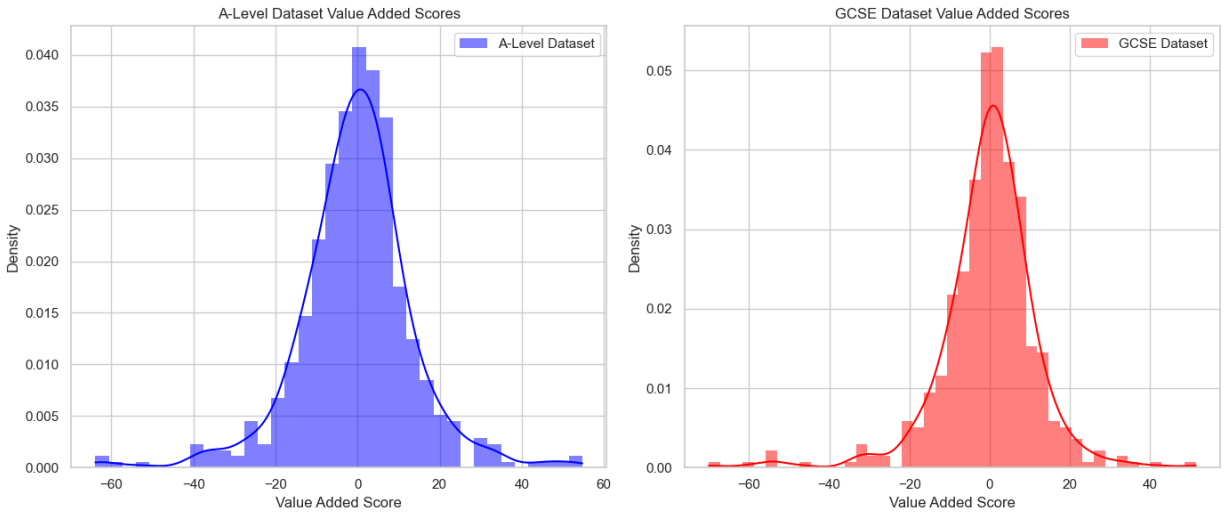


In [8]:
```
# Calculating the Value Added for both datasets
df_gcse['Value Added'] = df_gcse['Final Module Score'] - df_gcse['Predict
df_a_level['Value Added'] = df_a_level['Final Module Score'] - df_a_level

# Display the first few rows of both datasets with the new column
gcse_head = df_gcse.head()
a_level_head = df_a_level.head()

gcse_head, a_level_head
```

Out[8]:
```
(   Student ID  Ethnicity Ethnicity Summary Gender Highest Qual on Entry
\
 0         163  Pakistani             Asian      M               A-Level
 1         164      Black             Black      M               A-Level
 2         165  Pakistani             Asian      F               Level 3
 3         166      Black             Black      F               A-Level
 4         167      White             White      M               Level 3

   Qualification Summary Socio-economic classification Mature/Young Stude
nt  \
 0               A-Level                       Unknown               MATU
RE
 1               A-Level                       Unknown                YOU
NG
 2         Level 3 Quals                       Unknown                YOU
NG
 3               A-Level                 Working-Class                YOU
NG
 4         Level 2 Quals                  Professional                YOU
NG

   Module Code Mathematics Requirement     Disability Disability Summary
\
 0      CH1MAT                    GCSE  No disability      No disability
 1      CH1MAT                    GCSE  No disability      No disability
 2      CH1MAT                    GCSE  No disability      No disability
 3      CH1MAT                    GCSE  No disability      No disability
 4      CH1MAT                    GCSE  No disability      No disability

    Diagnostic Score  PAL Attendance  Final Module Score  \
 0             8.420               0                8.50
 1             9.330               0               46.00
 2             7.955               0               20.00
 3             7.905               0                2.00
 4            12.000               4               42.54

    Predicted Final Module Score  Value Added
 0                       18.2870      -9.7870
 1                       18.9149      27.0851
 2                       25.4796      -5.4796
 3                       10.1162      -8.1162
 4                       41.6375       0.9025  ,
    Student ID                   Ethnicity Ethnicity Summary Gender  \
 0           1                 Other Mixed             Other      M
 1           2  Other Asian or Bangladeshi             Asian      F
 2           3                       Black             Black      M
 3           4                       Black             Black      F
 4           5  Other Asian or Bangladeshi             Asian      M

   Highest Qual on Entry Qualification Summary Socio-economic classificat
```

```
ion  \
 0               A-Level               A-Level                       Unkn
own
 1               A-Level               A-Level                 Working-Cl
ass
 2               A-Level               A-Level                 Working-Cl
ass
 3               A-Level               A-Level                 Working-Cl
ass
 4               A-Level               A-Level                       Unkn
own

   Mature/Young Student Module Code Mathematics Requirement  \
 0               MATURE      AM10FM                A-Level
 1                YOUNG      AM10FM                A-Level
 2                YOUNG      AM10FM                A-Level
 3                YOUNG      AM10FM                A-Level
 4                YOUNG      AM10FM                A-Level

                                          Disability  \
 0  Social or communication condition such as aspe...
 1                                    No disability
 2                                    No disability
 3                                    No disability
 4                                    No disability

          Disability Summary  Diagnostic Score  PAL Attendance  \
 0  Social/Learning disability             6.840               0
 1               No disability             8.230               0
 2               No disability             8.690               0
 3               No disability             7.905               1
 4               No disability             9.000               0

   Final Module Score  Predicted Final Module Score  Value Added
 0              55.98                     52.776200     3.203800
 1              67.29                     38.851300    28.438700
 2              53.30                     46.544000     6.756000
 3              53.70                     47.665400     6.034600
 4              54.30                     53.265433     1.034567  )
```

In [48]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Set the style for the plots
sns.set(style="whitegrid")

# Creating separate subplots for each dataset
fig, ax = plt.subplots(1, 2, figsize=(14, 6))

# Plotting histogram for A-Level Dataset
sns.histplot(df_a_level['Value Added'], color="blue", label='A-Level Data
ax[0].set_title('A-Level Dataset Value Added Scores')
ax[0].set_xlabel('Value Added Score')
ax[0].set_ylabel('Density')
ax[0].legend()

# Plotting histogram for GCSE Dataset
sns.histplot(df_gcse['Value Added'], color="red", label='GCSE Dataset', k
ax[1].set_title('GCSE Dataset Value Added Scores')
ax[1].set_xlabel('Value Added Score')
ax[1].set_ylabel('Density')
ax[1].legend()

plt.tight_layout()
plt.show()
```



In [10]:
```
df_a_level
```

Out[10]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	I
0	1	Other Mixed	Other	M	A-Level	A-Level	Unknown	
1	2	Other Asian or Bangladeshi	Asian	F	A-Level	A-Level	Working-Class	
2	3	Black	Black	M	A-Level	A-Level	Working-Class	
3	4	Black	Black	F	A-Level	A-Level	Working-Class	
4	5	Other Asian or Bangladeshi	Asian	M	A-Level	A-Level	Unknown	
...	...	...	...	...	...	...	...	
531	1061	Black	Black	M	Other Qualification	Other	Middle Income	
532	1062	Pakistani	Asian	F	Level 3	Level 3 Quals	Unknown	
533	1063	Black	Black	M	A-Level	Level 2 Quals	Middle Income	
534	1064	Other Mixed	Other	M	A-Level	A-Level	Unknown	
535	1065	Black	Black	F	A-Level	Level 2 Quals	Middle Income	

536 rows × 17 columns

In [11]:
```
df_gcse
```

Out[11]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature S
0	163	Pakistani	Asian	M	A-Level	A-Level	Unknown	M
1	164	Black	Black	M	A-Level	A-Level	Unknown	
2	165	Pakistani	Asian	F	Level 3	Level 3 Quals	Unknown	
3	166	Black	Black	F	A-Level	A-Level	Working-Class	
4	167	White	White	M	Level 3	Level 2 Quals	Professional	
...	...	...	...	...	...	...	...	
483	678	White	White	M	Level 3	Level 3 Quals	Professional	
484	679	Pakistani	Asian	F	Level 3	Level 2 Quals	Unknown	
485	680	Other Mixed	Other	M	Level 3	Level 2 Quals	Professional	
486	681	White	White	M	Level 3	Level 2 Quals	Unknown	
487	682	Indian	Asian	M	Level 3	Level 3 Quals	Professional	

488 rows × 17 columns

In [12]:
```
# Filtering the dataframes where 'PAL Attendance' is not equal to zero
df_a_level_filtered = df_a_level[df_a_level['PAL Attendance'] != 0]
df_gcse_filtered = df_gcse[df_gcse['PAL Attendance'] != 0]

# Displaying the first few rows of the filtered dataframes
df_a_level_filtered_head = df_a_level_filtered.head()
df_gcse_filtered_head = df_gcse_filtered.head()

df_a_level_filtered_head, df_gcse_filtered_head
```

Out[12]:
```
(    Student ID    Ethnicity Ethnicity Summary Gender Highest Qual on Ent
ry  \
 3            4        Black             Black      F               A-Lev
el
 5            6        Black             Black      F               A-Lev
el
 10          11    Pakistani             Asian      M               Level
3
 13          14  Other Mixed             Other      M               Unkno
wn
 15          16        White             White      M               Level
3
```

	Qualification Summary	Socio-economic classification	Mature/Young Student
3	A-Level	Working-Class	YOUNG
5	A-Level	Middle Income	YOUNG
10	Level 3 Quals	Unknown	YOUNG
13	Blank	Unknown	YOUNG
15	Level 3 Quals	Professional	YOUNG

	Module Code	Mathematics Requirement	Disability	Disability Summary
3	AM10FM	A-Level	No disability	No disability
5	AM10FM	A-Level	No disability	No disability
10	AM10FM	A-Level	No disability	No disability
13	AM10FM	A-Level	No disability	No disability
15	AM10FM	A-Level	No disability	No disability

	Diagnostic Score	PAL Attendance	Final Module Score
3	7.905	1	53.70
5	4.215	3	48.67
10	8.195	1	46.03
13	3.000	1	40.68
15	9.500	12	62.05

	Predicted Final Module Score	Value Added
3	47.665400	6.034600
5	43.338600	5.331400
10	40.792900	5.237100
13	39.567107	1.112893
15	57.059300	4.990700

,

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry
4	167	White	White	M	Level 3
9	172	Other Mixed	Other	M	Level 3
10	173	Pakistani	Asian	F	Level 3
14	177	White	White	M	Level 3
22	185	Other Mixed	Other	M	Level 3

	Qualification Summary	Socio-economic classification	Mature/Young Student
4	Level 2 Quals	Professional	YOUNG
9	Level 2 Quals	Unknown	YOUNG
10	Level 2 Quals	Middle Income	YOUNG
14	Level 2 Quals	Working-Class	YOUNG
22	Level 2 Quals	Working-Class	YOUNG

```
    Module Code Mathematics Requirement     Disability Disability Summary
\
4       CH1MAT                    GCSE  No disability      No disability
9       CH1MAT                    GCSE  No disability      No disability
10      CH1MAT                    GCSE  No disability      No disability
14      CH1MAT                    GCSE  No disability      No disability
22      CH1MAT                    GCSE  No disability      No disability

     Diagnostic Score  PAL Attendance  Final Module Score  \
4                12.0               4                42.54
9                 7.0               2                34.00
10                9.0              10                33.00
14               13.0               5                47.50
22               10.0               1                68.26

     Predicted Final Module Score  Value Added
4                         41.6375       0.9025
9                         36.8144      -2.8144
10                        35.6990      -2.6990
14                        51.1522      -3.6522
22                        59.4809       8.7791  )
```

In [13]:
```
# Calculate the degrees of freedom for each dataset
# Degrees of freedom (df) is calculated as the number of samples (n) minu

a_level = df_a_level_filtered.shape[0] - 2  # Sample size of A-Level data
gcse = df_gcse_filtered.shape[0] - 2  # Sample size of GCSE dataset minus

a_level, gcse
```

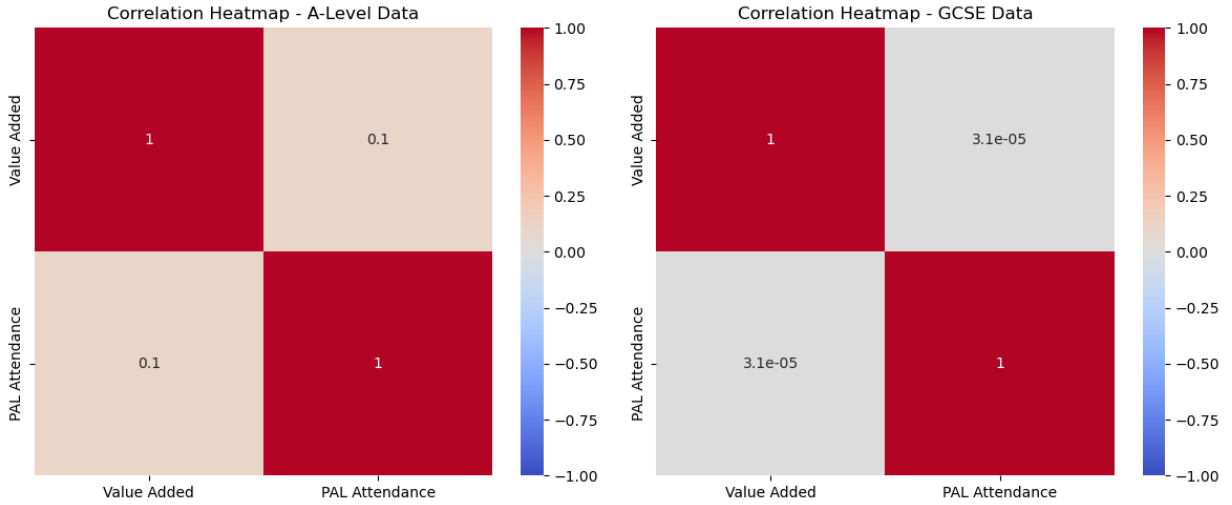
Out[13]:
```
(209, 187)
```

In [14]:
```
# Calculating Pearson correlation coefficient for 'Value Added' vs 'PAL A
pearson_corr_a_level = df_a_level_filtered['Value Added'].corr(df_a_level
pearson_corr_gcse = df_gcse_filtered['Value Added'].corr(df_gcse_filtered

pearson_corr_a_level, pearson_corr_gcse
```

Out[14]:
```
(0.10481978446466858, 3.1149495664851356e-05)
```

In [15]:

```
# Creating correlation matrices for both datasets
corr_matrix_a_level = df_a_level_filtered[['Value Added', 'PAL Attendance
corr_matrix_gcse = df_gcse_filtered[['Value Added', 'PAL Attendance']].co

# Setting up the matplotlib figure
plt.figure(figsize=(12, 5))

# Plotting the heatmap for A-Level Data
plt.subplot(1, 2, 1)
sns.heatmap(corr_matrix_a_level, annot=True, cmap='coolwarm', vmin=-1, vm
plt.title('Correlation Heatmap - A-Level Data')

# Plotting the heatmap for GCSE Data
plt.subplot(1, 2, 2)
sns.heatmap(corr_matrix_gcse, annot=True, cmap='coolwarm', vmin=-1, vmax=
plt.title('Correlation Heatmap - GCSE Data')

# Show the plots
plt.tight_layout()
plt.show()
```



Correlation Analysis Report between 'Value Added' and 'PAL Attendance'

Pearson correlation coefficients were calculated to assess the linear relationship between 'Value Added' and 'PAL Attendance'.

Results

- **A-Level Data**: Pearson correlation coefficient = `0.105`
- **GCSE Data**: Pearson correlation coefficient = `0.000031`

Interpretation

- **A-Level Data**: The coefficient indicates a very weak positive linear relationship.
- **GCSE Data**: The coefficient is close to zero, suggesting virtually no linear relationship.

Significance Analysis

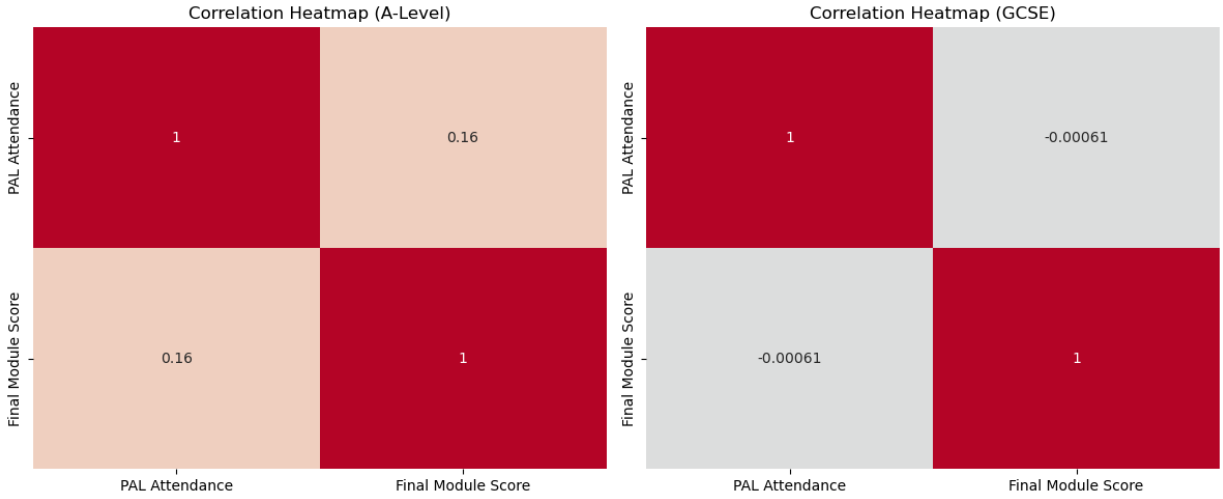
- **A-Level Data**: The correlation coefficient did not exceed the typical critical value for significance (approx. 0.14), suggesting the correlation is statistically not significant at the 0.05 level.
- **GCSE Data**: The correlation coefficient is too small to be considered significant at the 0.05 level.

In [16]:
```
# Calculating Pearson correlation coefficient for 'PAL Attendance' vs 'Fi
pearson_corr_a_level_pal_vs_final = df_a_level_filtered['PAL Attendance']
pearson_corr_gcse_pal_vs_final = df_gcse_filtered['PAL Attendance'].corr(

pearson_corr_a_level_pal_vs_final, pearson_corr_gcse_pal_vs_final
```

Out[16]:
```
(0.15743964781083122, -0.000612912681342636)
```

In [17]:

```
# Creating smaller correlation matrices focusing on 'PAL Attendance' and
corr_matrix_a_level_specific = df_a_level_filtered[['PAL Attendance', 'Fi
corr_matrix_gcse_specific = df_gcse_filtered[['PAL Attendance', 'Final Mo

# Setting up the matplotlib figure
plt.figure(figsize=(12, 5))

# Plotting the heatmap for A-Level Data
plt.subplot(1, 2, 1)
sns.heatmap(corr_matrix_a_level_specific, annot=True, cmap='coolwarm', vm
plt.title('Correlation Heatmap (A-Level)')

# Plotting the heatmap for GCSE Data
plt.subplot(1, 2, 2)
sns.heatmap(corr_matrix_gcse_specific, annot=True, cmap='coolwarm', vmin=
plt.title('Correlation Heatmap (GCSE)')

# Show the plots
plt.tight_layout()
plt.show()
```



Correlation Significance Analysis between 'PAL Attendance' and 'Final Module Score'

Pearson Correlation Coefficients

- **A-Level Data**: `0.157`
- **GCSE Data**: `-0.00061`

A-Level Data

- The correlation coefficient of `0.157` is slightly above the critical value of `0.14`.
- **Conclusion**: The correlation is statistically significant at the 0.05 level.

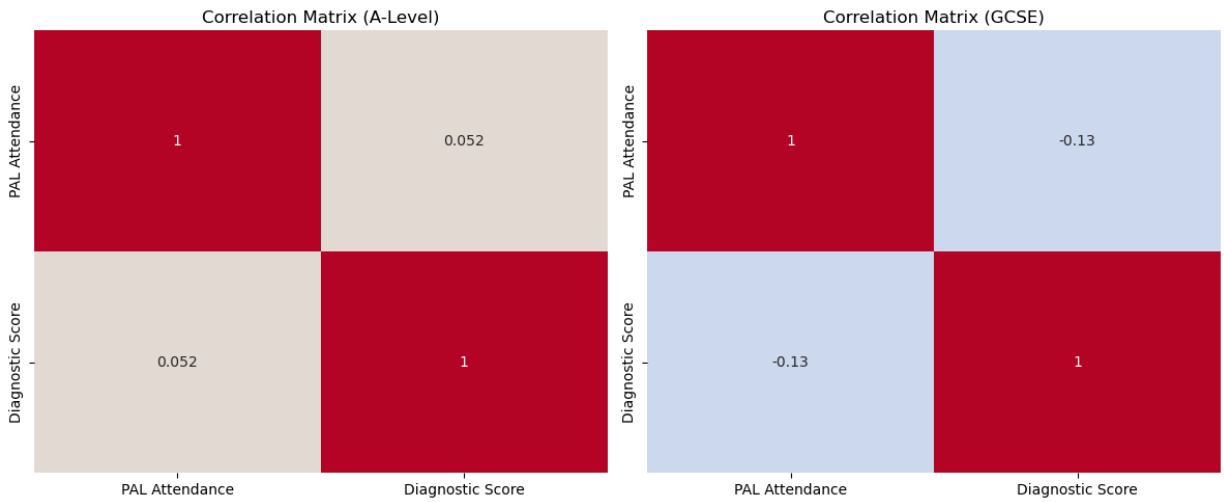
GCSE Data

- The correlation coefficient of `-0.00061` is far below the critical value.
- **Conclusion**: The correlation is not statistically significant at the 0.05 level.

In [18]:
```
# Calculating Pearson correlation coefficient for 'PAL Attendance' vs 'Di
pearson_corr_a_level_pal_vs_diagnostic = df_a_level_filtered['PAL Attenda
pearson_corr_gcse_pal_vs_diagnostic = df_gcse_filtered['PAL Attendance'].

pearson_corr_a_level_pal_vs_diagnostic, pearson_corr_gcse_pal_vs_diagnost
```

Out[18]:
```
(0.05187420914021662, -0.12887346216319517)
```

In [19]:
```
# Creating smaller correlation matrices focusing on 'PAL Attendance' and
corr_matrix_a_level_pal_diagnostic = df_a_level_filtered[['PAL Attendance
corr_matrix_gcse_pal_diagnostic = df_gcse_filtered[['PAL Attendance', 'Di

# Setting up the matplotlib figure
plt.figure(figsize=(12, 5))

# Plotting the heatmap for A-Level Data
plt.subplot(1, 2, 1)
sns.heatmap(corr_matrix_a_level_pal_diagnostic, annot=True, cmap='coolwar
plt.title('Correlation Matrix (A-Level)')

# Plotting the heatmap for GCSE Data
plt.subplot(1, 2, 2)
sns.heatmap(corr_matrix_gcse_pal_diagnostic, annot=True, cmap='coolwarm',
plt.title('Correlation Matrix (GCSE)')

# Show the plots
plt.tight_layout()
plt.show()
```



Significance Analysis of Correlation between 'PAL Attendance' and 'Diagnostic Score'

Pearson Correlation Coefficients

- **A-Level Data**: `0.052`
- **GCSE Data**: `−0.129`

A-Level Data

- The correlation coefficient of `0.052` does not exceed the critical value.
- **Conclusion**: The correlation is not statistically significant at the 0.05 level.

GCSE Data

- The correlation coefficient of `−0.129` does not exceed the critical value in magnitude.
- **Conclusion**: The correlation is not statistically significant at the 0.05 level.

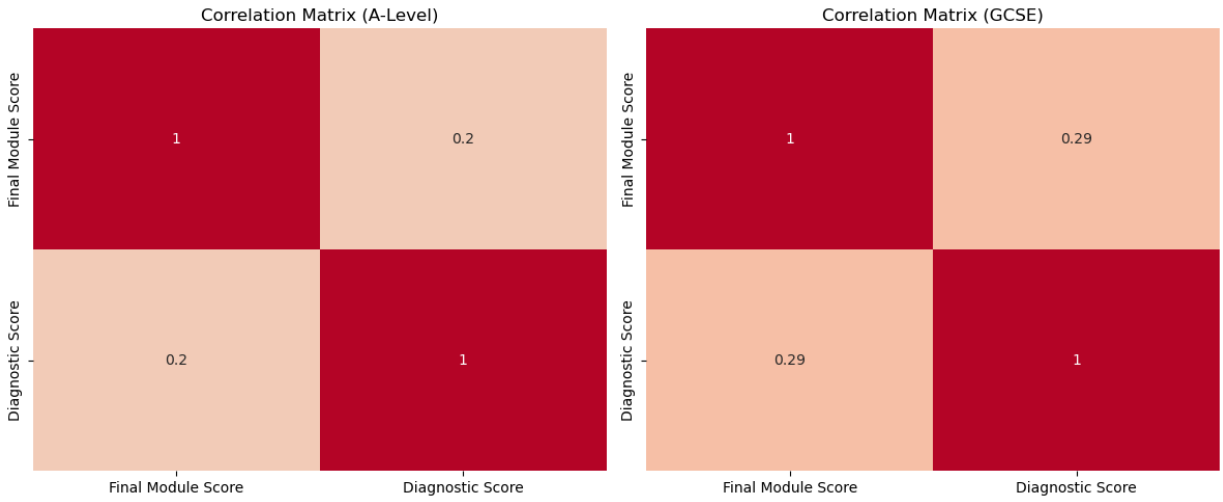
Summary

- The analysis suggests that the correlations observed in both datasets between 'PAL Attendance' and 'Diagnostic Score' are not statistically significant.
- This implies that the observed correlations are likely due to random chance rather than a true underlying relationship.

In [20]:
```
# Calculating Pearson correlation coefficient for 'Final Module Score' vs
pearson_corr_a_level_final_vs_diagnostic = df_a_level_filtered['Final Mod
pearson_corr_gcse_final_vs_diagnostic = df_gcse_filtered['Final Module Sc

pearson_corr_a_level_final_vs_diagnostic, pearson_corr_gcse_final_vs_diag
```

Out[20]: (0.19627688368717155, 0.2878713185868397)

In [21]:
```
# Creating smaller correlation matrices focusing on 'Final Module Score'
corr_matrix_a_level_final_diagnostic = df_a_level_filtered[['Final Module
corr_matrix_gcse_final_diagnostic = df_gcse_filtered[['Final Module Score

# Setting up the matplotlib figure
plt.figure(figsize=(12, 5))

# Plotting the heatmap for A-Level Data
plt.subplot(1, 2, 1)
sns.heatmap(corr_matrix_a_level_final_diagnostic, annot=True, cmap='coolw
plt.title('Correlation Matrix (A-Level)')

# Plotting the heatmap for GCSE Data
plt.subplot(1, 2, 2)
sns.heatmap(corr_matrix_gcse_final_diagnostic, annot=True, cmap='coolwarm
plt.title('Correlation Matrix (GCSE)')

# Show the plots
plt.tight_layout()
plt.show()
```



Correlation Analysis: Final Module Score vs. Diagnostic Score

Pearson Correlation Coefficients

- **A-Level Data**: `0.196`
- **GCSE Data**: `0.288`

A-Level Data

- The correlation coefficient of `0.196` is above the critical value.
- **Conclusion**: The correlation is likely statistically significant at the 0.05 level.

GCSE Data

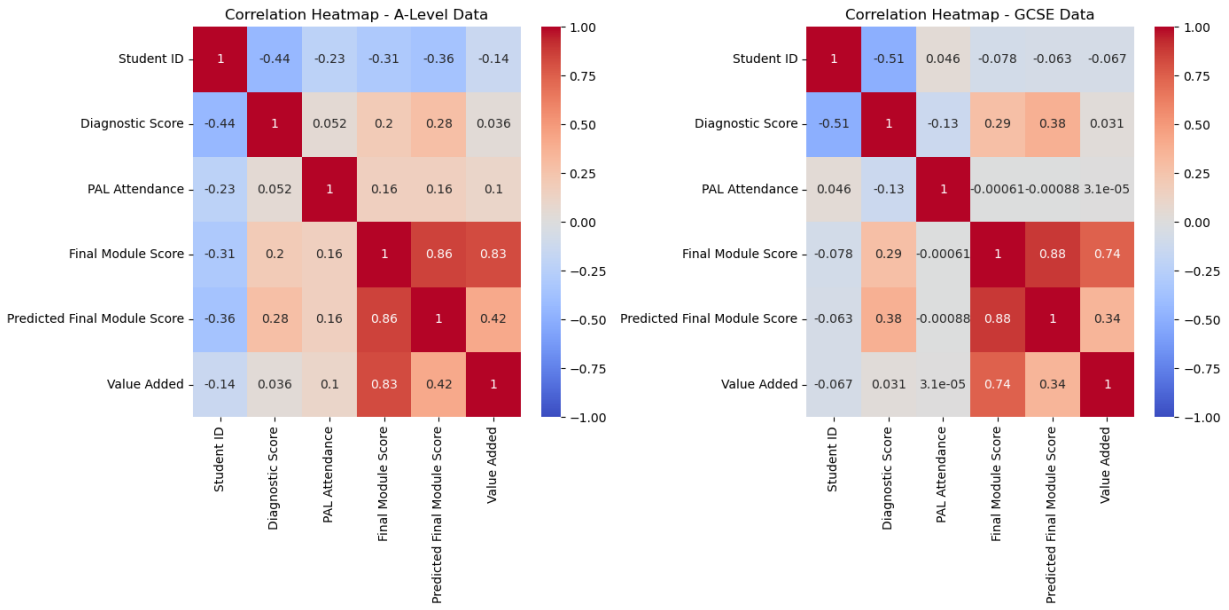
- The correlation coefficient of `0.288` is well above the critical value.
- **Conclusion**: The correlation is statistically significant at the 0.05 level.

Summary

- Both datasets show statistically significant correlations between 'Final Module Score' and 'Diagnostic Score'.
- These results suggest that the observed correlations are unlikely to be due to chance and may reflect a true underlying relationship, although the correlation strength is weak to moderate.

In [ ]:

In [22]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Creating correlation matrices for both datasets
corr_matrix_a_level_full = df_a_level_filtered.corr(method='pearson')
corr_matrix_gcse_full = df_gcse_filtered.corr(method='pearson')

# Setting up the matplotlib figure
plt.figure(figsize=(14, 7))

# Plotting the heatmap for A-Level Data
plt.subplot(1, 2, 1)
sns.heatmap(corr_matrix_a_level_full, annot=True, cmap='coolwarm', vmin=-
plt.title('Correlation Heatmap - A-Level Data')

# Plotting the heatmap for GCSE Data
plt.subplot(1, 2, 2)
sns.heatmap(corr_matrix_gcse_full, annot=True, cmap='coolwarm', vmin=-1,
plt.title('Correlation Heatmap - GCSE Data')

# Adjusting layout for better visualization
plt.tight_layout()
plt.show()
```



In [23]:
```
# Calculating summary statistics of PAL Attendance based on Ethnicity for
ethnicity_stats_a_level = df_a_level_filtered.groupby('Ethnicity')['PAL A
ethnicity_stats_gcse = df_gcse_filtered.groupby('Ethnicity')['PAL Attenda

(ethnicity_stats_a_level, ethnicity_stats_gcse)
```

Out[23]:
```
(                          count      mean       std  min  25%  50%   7
5%  \
 Ethnicity
 Black                      46.0  3.521739  3.277798  1.0  1.0  3.0  4.
00
 Indian                     26.0  3.615385  3.047319  1.0  1.0  2.5  5.
00
 Other Asian or Bangladeshi 22.0  3.136364  2.948696  1.0  1.0  2.0  4.
00
 Other Mixed                42.0  4.333333  4.771187  1.0  2.0  3.0  5.
75
 Pakistani                  43.0  3.302326  3.370282  1.0  1.0  2.0  4.
00
 White                      32.0  4.875000  3.899959  1.0  1.0  3.5  7.
00

                            max
 Ethnicity
 Black                     14.0
 Indian                    12.0
 Other Asian or Bangladeshi 13.0
 Other Mixed               25.0
 Pakistani                 19.0
 White                     14.0  ,
                           count      mean       std  min  25%  50%   7
5%  \
 Ethnicity
 Black                      42.0  2.785714  2.203497  1.0  1.0  2.0  3.
00
 Indian                     21.0  4.428571  2.992849  1.0  1.0  5.0  6.
00
 Other Asian or Bangladeshi 23.0  3.217391  1.999012  1.0  2.0  3.0  4.
50
 Other Mixed                34.0  3.588235  2.475549  1.0  1.0  3.0  5.
00
 Pakistani                  35.0  2.971429  2.281456  1.0  1.0  2.0  4.
00
 White                      34.0  2.970588  2.443114  1.0  1.0  2.0  4.
75

                            max
 Ethnicity
 Black                      8.0
 Indian                    10.0
 Other Asian or Bangladeshi  8.0
 Other Mixed                9.0
 Pakistani                 10.0
 White                     10.0  )
```

In [ ]:
```
```

In [24]:
```
# A Level dataset
socio_economic_stats_a_level = df_a_level_filtered.groupby('Socio-economi

# GCSE dataset
socio_economic_stats_gcse = df_gcse_filtered.groupby('Socio-economic clas

(socio_economic_stats_a_level, socio_economic_stats_gcse)
```

Out[24]:
```
(                                count      mean       std  min  25%  50%
75%  \
 Socio-economic classification
 Middle Income                   63.0  3.952381  3.553411  1.0  1.0  3.0
4.0
 Professional                    54.0  3.648148  2.921342  1.0  2.0  3.0
5.0
 Unknown                         67.0  4.119403  4.541133  1.0  1.0  3.0
6.0
 Working-Class                   27.0  3.074074  2.999050  1.0  1.0  2.0
4.0

                                  max
 Socio-economic classification
 Middle Income                   17.0
 Professional                    14.0
 Unknown                         25.0
 Working-Class                   14.0  ,
                                count      mean       std  min  25%  50%
75%  \
 Socio-economic classification
 Middle Income                   46.0  3.456522  2.630644  1.0  1.0  2.5
5.0
 Professional                    38.0  3.421053  2.647095  1.0  1.0  2.5
6.0
 Unknown                         66.0  3.015152  2.123078  1.0  1.0  2.0
4.0
 Working-Class                   39.0  3.153846  2.401079  1.0  1.0  2.0
5.0

                                  max
 Socio-economic classification
 Middle Income                   10.0
 Professional                    10.0
 Unknown                          8.0
 Working-Class                   10.0  )
```

In [25]:
```
# Calculating summary statistics of PAL Attendance based on Gender for bo

# A Level dataset
gender_stats_a_level = df_a_level_filtered.groupby('Gender')['PAL Attenda

# GCSE dataset
gender_stats_gcse = df_gcse_filtered.groupby('Gender')['PAL Attendance'].

(gender_stats_a_level, gender_stats_gcse)
```

Out[25]:
```
(        count      mean       std  min  25%  50%   75%   max
 Gender
 F        52.0  4.153846  3.996982  1.0  2.0  3.0  5.25  25.0
 M       159.0  3.704403  3.585402  1.0  1.0  2.0  5.00  19.0,
         count      mean       std  min  25%  50%  75%   max
 Gender
 F        39.0  3.230769  2.832718  1.0  1.0  2.0  5.0  10.0
 M       150.0  3.233333  2.295068  1.0  1.0  2.0  5.0  10.0)
```

In [26]:
```
# Calculating summary statistics of PAL Attendance based on Highest Quali

# A Level dataset
highest_qual_stats_a_level = df_a_level_filtered.groupby('Highest Qual on

# GCSE dataset
highest_qual_stats_gcse = df_gcse_filtered.groupby('Highest Qual on Entry

(highest_qual_stats_a_level, highest_qual_stats_gcse)
```

Out[26]:

```
(                      count      mean       std  min  25%  50%  75%   m
ax
 Highest Qual on Entry
 A-Level               137.0  3.693431  3.683353  1.0  1.0  3.0  4.0  25
.0
 Diploma at level 3     11.0  6.000000  5.366563  1.0  1.0  5.0  8.5  17
.0
 Level 3                23.0  4.695652  4.626355  1.0  1.0  3.0  6.5  16
.0
 Other Qualification     3.0  4.000000  3.000000  1.0  2.5  4.0  5.5   7
.0
 Unknown                37.0  3.054054  1.957123  1.0  1.0  3.0  4.0   8
.0,
                       count      mean       std  min  25%  50%   75%
max
 Highest Qual on Entry
 A-Level                22.0  2.727273  1.956231  1.0  1.0  2.0  3.75
7.0
 Diploma at level 3      8.0  1.750000  1.752549  1.0  1.0  1.0  1.25
6.0
 Level 3               124.0  3.298387  2.479115  1.0  1.0  2.0  5.00  1
0.0
 Other Qualification     1.0  1.000000       NaN  1.0  1.0  1.0  1.00
1.0
 Unknown                34.0  3.735294  2.428478  1.0  1.0  4.0  5.75
9.0)
```

PAL Attendance Summary Statistics: Key Insights

Ethnicity

- **A-Level**: Highest average in 'White' (4.88), lowest in 'Other Asian or Bangladeshi' (3.14).
- **GCSE**: Highest average in 'Indian' (4.43), lowest in 'Black' (2.79).

Socio-economic Classification

- **A-Level**: 'Unknown' class highest average (4.12); 'Middle Income' highest max (17).
- **GCSE**: 'Middle Income' highest average (3.46).

Gender

- **A-Level**: Females higher average (4.15) than males (3.70).
- **GCSE**: Similar average between genders (Females: 3.23, Males: 3.23).

Highest Qualification on Entry

- **A-Level**: 'Diploma at level 3' highest average (6.00); 'Unknown' qualifications lowest (3.05).
- **GCSE**: 'Unknown' qualifications highest average (3.74); single data point for 'Other Qualification'.

In [27]:
```
# Calculating summary statistics of 'Value Added' based on each of the fo

# Summary statistics based on Ethnicity
value_added_ethnicity_a_level = df_a_level_filtered.groupby('Ethnicity')[
value_added_ethnicity_gcse = df_gcse_filtered.groupby('Ethnicity')['Value


value_added_ethnicity_a_level
```

Out[27]:

Ethnicity	count	mean	std	min	25%	50%	75%
Black	46.0	0.719123	13.749840	-37.826067	-7.109180	0.284650	7.191200
Indian	26.0	4.100958	13.564755	-15.660100	-7.006779	1.277150	14.872725
Other Asian or Bangladeshi	22.0	0.950492	12.206157	-28.172100	-8.118400	0.263400	7.357100
Other Mixed	42.0	-1.741393	14.184446	-33.586800	-8.939417	-0.071467	6.320450
Pakistani	43.0	3.710184	15.510906	-36.052383	-4.330650	2.929600	7.428087
White	32.0	3.605754	12.695537	-19.642000	-4.310125	0.214000	8.347600

In [28]:
```
value_added_ethnicity_gcse
```

Out[28]:

Ethnicity	count	mean	std	min	25%	50%	75%
Black	42.0	1.978694	11.030555	-31.723950	-3.37525	1.09855	7.783300
Indian	21.0	0.820537	10.162694	-18.073600	-3.89505	0.55570	2.887500
Other Asian or Bangladeshi	23.0	0.712324	9.112849	-21.025200	-5.31175	0.63830	6.851383
Other Mixed	34.0	-0.848974	9.466017	-27.975900	-4.92915	-0.28220	5.529675
Pakistani	35.0	0.649592	8.311130	-20.561733	-5.14155	-0.03820	6.541217
White	34.0	1.518638	10.675349	-33.860300	-2.76575	0.76130	5.359850

In [29]:
```
# Summary statistics based on Socio-economic classification
value_added_socio_economic_a_level = df_a_level_filtered.groupby('Socio-e
value_added_socio_economic_gcse = df_gcse_filtered.groupby('Socio-economi

value_added_socio_economic_a_level
```

Out[29]:

Socio-economic classification	count	mean	std	min	25%	50%	75%
Middle Income	63.0	2.794379	14.204643	-32.764000	-5.468950	0.50050	8.676200
Professional	54.0	-0.337048	11.142820	-24.682600	-8.869817	-0.28105	6.711868
Unknown	67.0	3.152094	16.429849	-37.826067	-5.502150	3.45130	8.289550
Working-Class	27.0	-0.245801	10.974007	-26.891500	-6.683333	1.51210	6.408317

In [30]:
```
value_added_socio_economic_gcse
```

Out[30]:

Socio-economic classification	count	mean	std	min	25%	50%	75%
Middle Income	46.0	0.468898	6.437338	-21.02520	-2.570275	0.636617	3.636825
Professional	38.0	0.772465	10.853265	-27.97590	-4.344050	1.425800	8.675575
Unknown	66.0	0.166953	8.481493	-31.72395	-4.618825	-0.093550	5.067050
Working-Class	39.0	2.571342	13.634308	-33.86030	-4.775400	2.666100	8.606000

In [31]:
```
# Summary statistics based on Gender
value_added_gender_a_level = df_a_level_filtered.groupby('Gender')['Value
value_added_gender_gcse = df_gcse_filtered.groupby('Gender')['Value Added

value_added_gender_a_level
```

Out[31]:

Gender	count	mean	std	min	25%	50%	75%	m
F	52.0	2.924170	16.672843	-37.826067	-5.543200	1.123800	7.867625	54.7575
M	159.0	1.322906	12.902501	-36.052383	-5.842858	1.112893	8.208321	51.7635

In [32]:
```
value_added_gender_gcse
```

Out[32]:

Gender	count	mean	std	min	25%	50%	75%	max
F	39.0	2.628584	10.366277	-19.3221	-2.73810	1.3672	6.328883	42.0092
M	150.0	0.398063	9.651203	-33.8603	-4.64035	0.5891	5.903700	32.2103

In [33]:
```
# Summary statistics based on Highest Qualification on Entry
value_added_highest_qual_a_level = df_a_level_filtered.groupby('Highest Q
value_added_highest_qual_gcse = df_gcse_filtered.groupby('Highest Qual on

value_added_highest_qual_a_level
```

Out[33]:

Highest Qual on Entry	count	mean	std	min	25%	50%	75%
A-Level	137.0	2.073991	12.640987	-33.586800	-5.387800	1.160133	7.574900
Diploma at level 3	11.0	-4.270700	15.912148	-36.052383	-9.024750	-2.599500	5.644433
Level 3	23.0	8.813286	16.452725	-10.820700	-0.691000	4.990700	8.289550
Other Qualification	3.0	2.913900	17.490501	-13.129100	-6.409200	0.310700	10.935400
Unknown	37.0	-2.329928	14.500196	-37.826067	-8.205367	-0.836600	6.733907

In [34]:
```
value_added_highest_qual_gcse
```

Out[34]:

Highest Qual on Entry	count	mean	std	min	25%	50%	75%	
A-Level	22.0	2.815898	11.054669	-33.86030	-0.014500	3.8334	9.422075	21.2
Diploma at level 3	8.0	6.764100	15.575332	-8.49140	-0.885200	1.9165	9.212750	42.0
Level 3	124.0	0.615837	9.017864	-31.72395	-3.732188	0.4606	5.753300	32.2
Other Qualification	1.0	0.558100	NaN	0.55810	0.558100	0.5581	0.558100	0.5
Unknown	34.0	-0.904713	10.109604	-27.97590	-6.790025	-1.7547	5.779275	16.7

'Value Added' Statistics: Highest, Lowest, and Maximum Values

A-Level Dataset

By Ethnicity

- **Highest Average**: Indian (4.10)
- **Lowest Average**: Other Mixed (-1.74)
- **Highest Maximum**: Pakistani (54.76)

By Socio-economic Classification

- **Highest Average**: Middle Income (2.79)
- **Highest Maximum**: Middle Income (54.76)

By Gender

- **Higher Average in Females**: 2.92

By Highest Qualification on Entry

- **Highest Average**: Level 3 (8.81)
- **Highest Maximum**: Level 3 (51.76)

GCSE Dataset

By Ethnicity

- **Highest Average**: Black (1.98)
- **Lowest Average**: Other Mixed (-0.85)
- **Highest Maximum**: Black (42.01)

By Socio-economic Classification

- **Highest Average**: Working-Class (2.57)
- **Highest Maximum**: Working-Class (42.01)

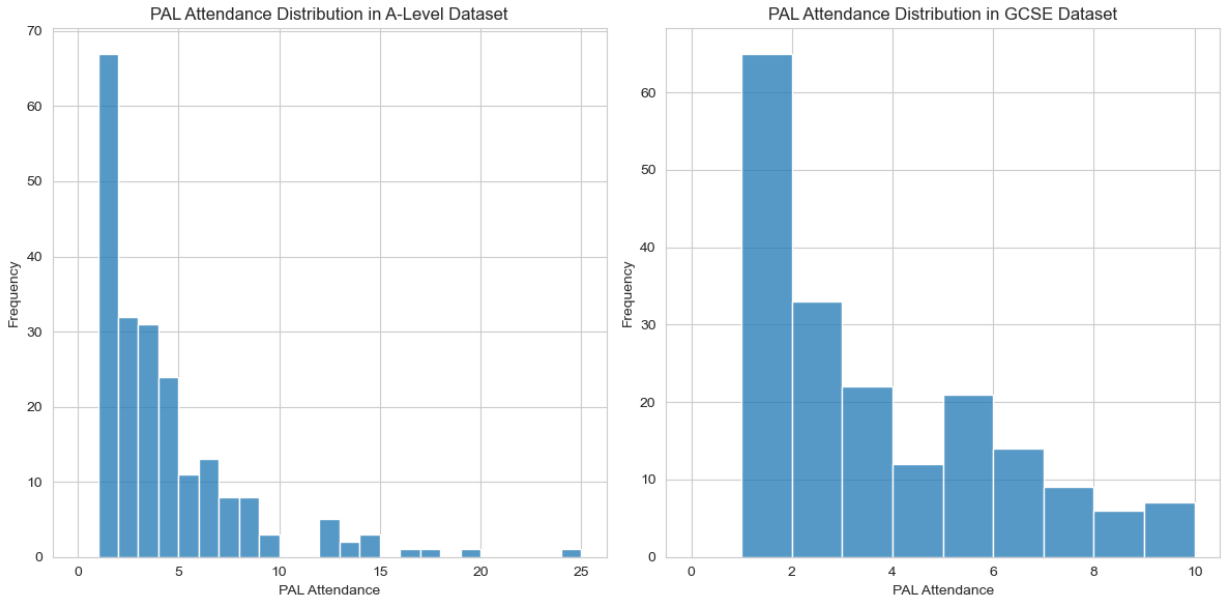
By Gender

- **Similar Average**: Females (2.63) and Males (0.40)

By Highest Qualification on Entry

- **Highest Average**: Diploma at level 3 (6.76)
- **Highest Maximum**: Diploma at level 3 (42.01)

In [35]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Set the aesthetic style of the plots
sns.set_style("whitegrid")

# Histograms for both datasets
plt.figure(figsize=(12, 6))

# Histogram for A-Level dataset
plt.subplot(1, 2, 1)
sns.histplot(df_a_level_filtered['PAL Attendance'], bins=range(0, 26), kd
plt.title('PAL Attendance Distribution in A-Level Dataset')
plt.xlabel('PAL Attendance')
plt.ylabel('Frequency')

# Histogram for GCSE dataset
plt.subplot(1, 2, 2)
sns.histplot(df_gcse_filtered['PAL Attendance'], bins=range(0, 11), kde=F
plt.title('PAL Attendance Distribution in GCSE Dataset')
plt.xlabel('PAL Attendance')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



In [36]:
```
# Step 1: Determine the maximum attendance in each dataset
max_attendance_a_level = df_a_level_filtered['PAL Attendance'].max()
max_attendance_gcse = df_gcse_filtered['PAL Attendance'].max()

max_attendance_a_level, max_attendance_gcse
```

Out[36]:
```
(25, 10)
```

In [37]:
```
# Define a function to categorize attendance
def categorize_attendance(row, max_attendance):
    if row > 0.50 * max_attendance:
        return 'High'
    elif row > 0.20 * max_attendance:
        return 'Medium'
    else:
        return 'Low'

# Calculate 50% and 20% of the maximum 'PAL Attendance' in each dataset
max_attendance_a_level = df_a_level_filtered['PAL Attendance'].max()
max_attendance_gcse = df_gcse_filtered['PAL Attendance'].max()

# Categorize attendance in each dataset
df_a_level_filtered['Attendance'] = df_a_level_filtered['PAL Attendance']
df_gcse_filtered['Attendance'] = df_gcse_filtered['PAL Attendance'].apply

# Display the first few rows of each dataframe to verify the changes
df_a_level_filtered.head()
```

```
/var/folders/jh/fgdcbdr1491cv5wb98g38zfc0000gn/T/ipykernel_73857/5627746.
py:15: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_a_level_filtered['Attendance'] = df_a_level_filtered['PAL Attendance
'].apply(categorize_attendance, args=(max_attendance_a_level,))
/var/folders/jh/fgdcbdr1491cv5wb98g38zfc0000gn/T/ipykernel_73857/5627746.
py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-do
cs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_gcse_filtered['Attendance'] = df_gcse_filtered['PAL Attendance'].app
ly(categorize_attendance, args=(max_attendance_gcse,))
```

Out[37]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature/ St
3	4	Black	Black	F	A-Level	A-Level	Working-Class	Y
5	6	Black	Black	F	A-Level	A-Level	Middle Income	Y
10	11	Pakistani	Asian	M	Level 3	Level 3 Quals	Unknown	Y
13	14	Other Mixed	Other	M	Unknown	Blank	Unknown	Y
15	16	White	White	M	Level 3	Level 3 Quals	Professional	Y

In [38]:
```
df_gcse_filtered.head()
```

Out[38]:

	Student ID	Ethnicity	Ethnicity Summary	Gender	Highest Qual on Entry	Qualification Summary	Socio-economic classification	Mature/Y St
4	167	White	White	M	Level 3	Level 2 Quals	Professional	Y(
9	172	Other Mixed	Other	M	Level 3	Level 2 Quals	Unknown	Y(
10	173	Pakistani	Asian	F	Level 3	Level 2 Quals	Middle Income	Y(
14	177	White	White	M	Level 3	Level 2 Quals	Working-Class	Y(
22	185	Other Mixed	Other	M	Level 3	Level 2 Quals	Working-Class	Y(

In [39]:
```
# Grouping by 'Attendance' category and calculating average 'Value Added'
avg_value_added_a_level = df_a_level_filtered.groupby('Attendance')['Valu

# Grouping by 'Attendance' category and calculating average 'Value Added'
avg_value_added_gcse = df_gcse_filtered.groupby('Attendance')['Value Adde

# Displaying the results
avg_value_added_a_level
```

Out[39]:
```
Attendance
High      9.501176
Low       1.005416
Medium    2.999857
Name: Value Added, dtype: float64
```

In [40]:
```
avg_value_added_gcse
```

Out[40]:
```
Attendance
High      2.725375
Low       0.993757
Medium   -0.605045
Name: Value Added, dtype: float64
```

In [41]:
```
# Filter for students who scored less than 4 in the diagnostic
failed_diagnostic_a_level = df_a_level_filtered[df_a_level_filtered['Diag
failed_diagnostic_gcse = df_gcse_filtered[df_gcse_filtered['Diagnostic Sc

# Group by 'Attendance' and calculate average 'Value Added' for students
avg_value_added_failed_a_level = failed_diagnostic_a_level.groupby('Atten

# Group by 'Attendance' and calculate average 'Value Added' for students
avg_value_added_failed_gcse = failed_diagnostic_gcse.groupby('Attendance'

# Displaying the results
avg_value_added_failed_a_level, avg_value_added_failed_gcse
```

Out[41]:
```
(Attendance
 High      19.276817
 Low        1.102096
 Medium     0.467254
 Name: Value Added, dtype: float64,
 Attendance
 High     -0.461215
 Low       4.085386
 Medium   -2.800338
 Name: Value Added, dtype: float64)
```

In [42]:
```
# Step 1: Filter for CS1MCP Module
cs1mcp_a_level = df_a_level_filtered[df_a_level_filtered['Module Code'] =
cs1mcp_gcse = df_gcse_filtered[df_gcse_filtered['Module Code'] == 'CS1MCP

# Step 2: Identify Students Who Failed the Diagnostic
failed_diagnostic_cs1mcp_a_level = cs1mcp_a_level[cs1mcp_a_level['Diagnos
failed_diagnostic_cs1mcp_gcse = cs1mcp_gcse[cs1mcp_gcse['Diagnostic Score

# Step 3: Focus on High PAL Engagement
high_engagement_failed_a_level = failed_diagnostic_cs1mcp_a_level[failed_
high_engagement_failed_gcse = failed_diagnostic_cs1mcp_gcse[failed_diagno

# Step 4: Calculate Average Final Module Score for High Engagement Group
avg_final_score_high_engagement_a_level = high_engagement_failed_a_level[
avg_final_score_high_engagement_gcse = high_engagement_failed_gcse['Final

# Step 5: Calculate Total Average Final Module Score
total_avg_final_score_a_level = cs1mcp_a_level['Final Module Score'].mean
total_avg_final_score_gcse = cs1mcp_gcse['Final Module Score'].mean()

# Displaying the results
(avg_final_score_high_engagement_a_level, total_avg_final_score_a_level),
```

Out[42]:
```
((nan, nan), (49.99769230769231, 56.98245508982033))
```

In [43]:
```
# Count the number of students in the A-Level dataset who failed the diag
count_high_engagement_failed_a_level = high_engagement_failed_a_level.sha

# Check if the count is zero or if there might be any data issues
count_high_engagement_failed_a_level
```

Out[43]:
```
0
```

In [44]:
```
new_module_code = 'EE1EMA'

# Step 1: Filter for the new module in both A-Level and GCSE datasets
new_module_a_level = df_a_level_filtered[df_a_level_filtered['Module Code
new_module_gcse = df_gcse_filtered[df_gcse_filtered['Module Code'] == new

# Step 2: Identify Students Who Failed the Diagnostic in the new module
failed_diagnostic_new_module_a_level = new_module_a_level[new_module_a_le
failed_diagnostic_new_module_gcse = new_module_gcse[new_module_gcse['Diag

# Step 3: Focus on High PAL Engagement in the new module
high_engagement_failed_new_module_a_level = failed_diagnostic_new_module_
high_engagement_failed_new_module_gcse = failed_diagnostic_new_module_gcs

# Step 4: Calculate Average Final Module Score for High Engagement Group
avg_final_score_high_engagement_new_module_a_level = high_engagement_fail
avg_final_score_high_engagement_new_module_gcse = high_engagement_failed_

# Step 5: Calculate Total Average Final Module Score for the new module
total_avg_final_score_new_module_a_level = new_module_a_level['Final Modu
total_avg_final_score_new_module_gcse = new_module_gcse['Final Module Sco

# Displaying the results for the new module
(avg_final_score_high_engagement_new_module_a_level, total_avg_final_scor
```

Out[44]:
```
((56.56666666666666, 32.14473684210526), (nan, nan))
```

Introduction In an era where educational equity and student success are key priorities in academic discourse, a comprehensive understanding of the factors influencing student behavior and outcomes is vital. This report delves into the multifaceted dimensions of how ethnicity, socio-economic background, gender, and educational background shape student performances within the Mathematics Department at Aston University.

Motivation and Benefits The study is driven by the need for inclusive and equitable educational environments. It aims to explore the complex interplay of various factors on student achievements, seeking to highlight areas for enhancing support systems, tailoring interventions, and promoting a learning atmosphere that accommodates diverse student needs. The benefits of such an analysis extend beyond mere academic performance metrics, encompassing broader aspects of student welfare and institutional effectiveness.

Aims The primary aim is to unravel the effects of ethnicity on student behavior and outcomes while considering the influences of socio-economic status, gender, and initial qualification levels. A range of statistical tools and methodologies are employed to provide a nuanced understanding of these factors' correlations with academic success and engagement.

Applications The findings are poised to significantly influence educational policies and practices. They offer critical insights for educators, administrators, and policymakers in developing strategies to address disparities and foster learning environments conducive to all students' growth. Moreover, these insights are

instrumental in guiding future research, enriching discussions on diversity and inclusion in education, and contributing to more equitable educational landscapes.

Impact of Educational Background The study begins by analyzing the influence of different educational backgrounds on students' initial academic performance. Using post-hoc Mann-Whitney U tests and Kruskal-Wallis H tests, it highlights the significant role of prior educational experiences in shaping university outcomes.

Role of Ethnicity in Educational Outcomes The second part focuses on ethnicity's significant impact on student academic performance. A rigorous Kruskal-Wallis H-test evaluates the diagnostic scores of students from various ethnic backgrounds, revealing substantial variances linked to ethnicity.

Comprehensive Multifactorial Analysis The third segment broadens the scope to include socio-economic and gender factors. Employing statistical methods like Tukey HSD and Dunn's tests, it examines mean diagnostic scores across diverse groups, providing a holistic view of the complex factors influencing student performance.

Correlation between PAL Attendance and Academic Performance The final phase investigates the correlation between students' engagement in peer-assisted learning (PAL) sessions and their final module scores. This analysis reveals how participation in additional learning activities correlates with academic success.

In [45]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Preprocessing
def preprocess_data(df):
    X = df[['Ethnicity', 'Socio-economic classification', 'Highest Qual o
    y = df['Final Module Score']
    return train_test_split(X, y, test_size=0.2, random_state=42)

X_train_a, X_test_a, y_train_a, y_test_a = preprocess_data(df_a_level)
X_train_g, X_test_g, y_train_g, y_test_g = preprocess_data(df_gcse)
```

In [46]:
```
# Model Building
def build_model(X_train, y_train):
    categorical_features = ['Ethnicity', 'Socio-economic classification',
    one_hot = OneHotEncoder()
    transformer = ColumnTransformer([("one_hot", one_hot, categorical_fea
    model = Pipeline(steps=[
        ("transformer", transformer),
        ("regressor", RandomForestRegressor(n_estimators=50, max_depth=5,
    ])
    model.fit(X_train, y_train)
    return model

model_a_level = build_model(X_train_a, y_train_a)
model_gcse = build_model(X_train_g, y_train_g)
```

In [47]:
```python
# Model Evaluation
def evaluate_model(model, X_test, y_test):
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    return mse, r2

mse_a, r2_a = evaluate_model(model_a_level, X_test_a, y_test_a)
mse_g, r2_g = evaluate_model(model_gcse, X_test_g, y_test_g)

print("A-Level Dataset: MSE =", mse_a, "R² =", r2_a)
print("GCSE Dataset: MSE =", mse_g, "R² =", r2_g)
```

```
A-Level Dataset: MSE = 537.7911609490516 R² = 0.04160384887523372
GCSE Dataset: MSE = 583.5805895701693 R² = -0.07605753246433178
```

In [ ]: