**A close up of a sign

Description automatically generatedAMERICAN INTERNATIONAL**

**UNIVERSITY-BANGLADESH**

**Faculty of Science and Technology**

**Project**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assignment Title: | Mid Term Project | | | |
| Assignment No: | 01 | | Date of Submission: | 26 April 2025 |
| Course Title: | INTRODUCTION TO DATA SCIENCE | | | |
| Course Code: | CSC4180 | | Section: | E |
| Semester: | Spring | 2024-25 | Course Teacher: | **DR. ABDUS SALAM** |

**Declaration and Statement of Authorship:**

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student’s work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of their arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

*\* Student(s) must complete all details except the faculty use part.*

\*\* Please submit all assignments to your course teacher or the office of the concerned teacher.

|  |  |
| --- | --- |
| Group Name/No.: | 9 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **ID** | **Program** | **Signature** |
| 1 | MAHMUDUS SAMI MAAHI | 22-46446-1 | BSc [CSE] |  |
| 2 | SABIHA IQBAL ETU | 22-47227-1 | BSc [CSE] |  |
| 3 | MUKSHIT SAFI OWASI | 22-47251-1 | BSc [CSE] |  |
| 4 |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| ***Faculty use only*** | | |
| FACULTY COMMENTS | **Marks Obtained** |  |
|  |
|  |
|  | **Total Marks** |  |
|  |
|  |

**Introduction:**

This report is based on the placement\_Data\_Full\_Class – modified.csv dataset. It contains student academic and placement-related information and includes 216 student records and 16 attributes, where each represents detailed information about education, work experiences, and placement outcomes. The dataset helps to understand whether things affect whether a student gets a job after finishing their studies. The dataset is also useful for data preparation tasks like handling missing or invalid values, normalization, data balancing and statistical analysis.

**Feature Description:**

**sl\_no**: Serial number (unique identifier for each student)

**gender**: Student’s gender (M/F)

**ssc\_p**: Secondary Education (10th grade) percentage

**ssc\_b**: Board of Secondary Education (Central/Others)

**hsc\_p**: Higher Secondary (12th grade) percentage

**hsc\_b**: Board of Higher Secondary (Central/Others)

**hsc\_s**: HSC specialization (Commerce, Science, Arts)

**degree\_p**: Undergraduate degree percentage

**degree\_t**: Type of undergraduate degree (Sci&Tech, Comm&Mgmt, Others)

**workex**: Work experience (Yes/No)

**etest\_p**: Employability test percentage

**specialisation**: MBA Specialization (Mkt&Fin or Mkt&HR)

**mba\_p**: MBA percentage

**status**: Placement status (Placed/Not Placed)

**salary**: Salary offered (only for placed students)

**class**: Additional attribute

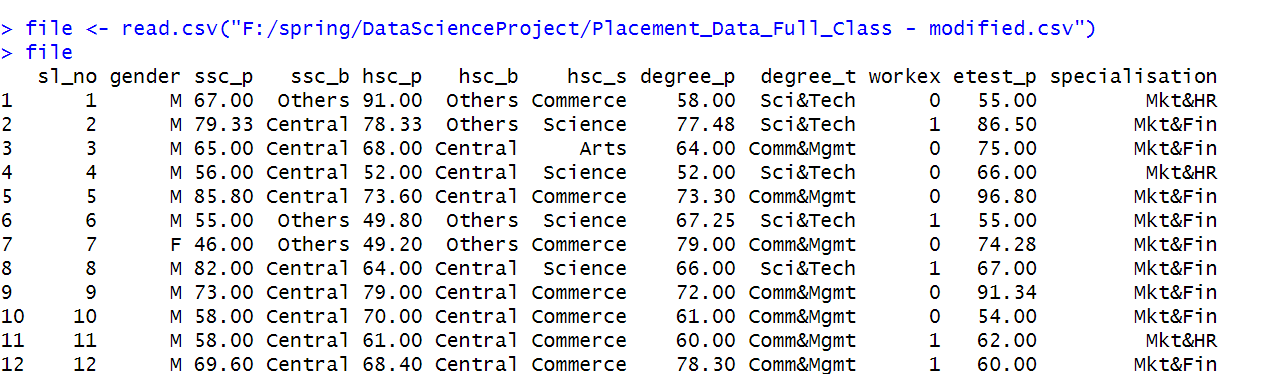
**Data exploration:**

***Required library –***



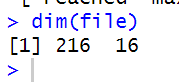
dplyr is a package used for data manipulation in R. This library is included for further use.

***Importing the dataset –***

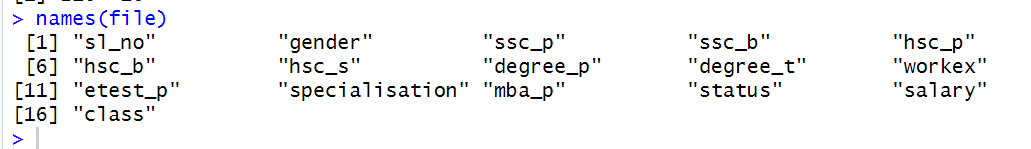


The read.csv() function reads a CSV file and loads the data into a data frame.

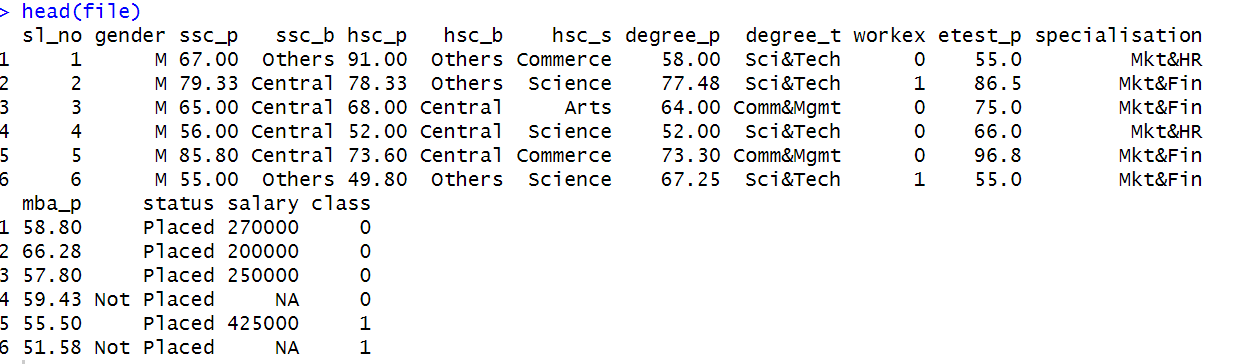
***Dataset explanation –***



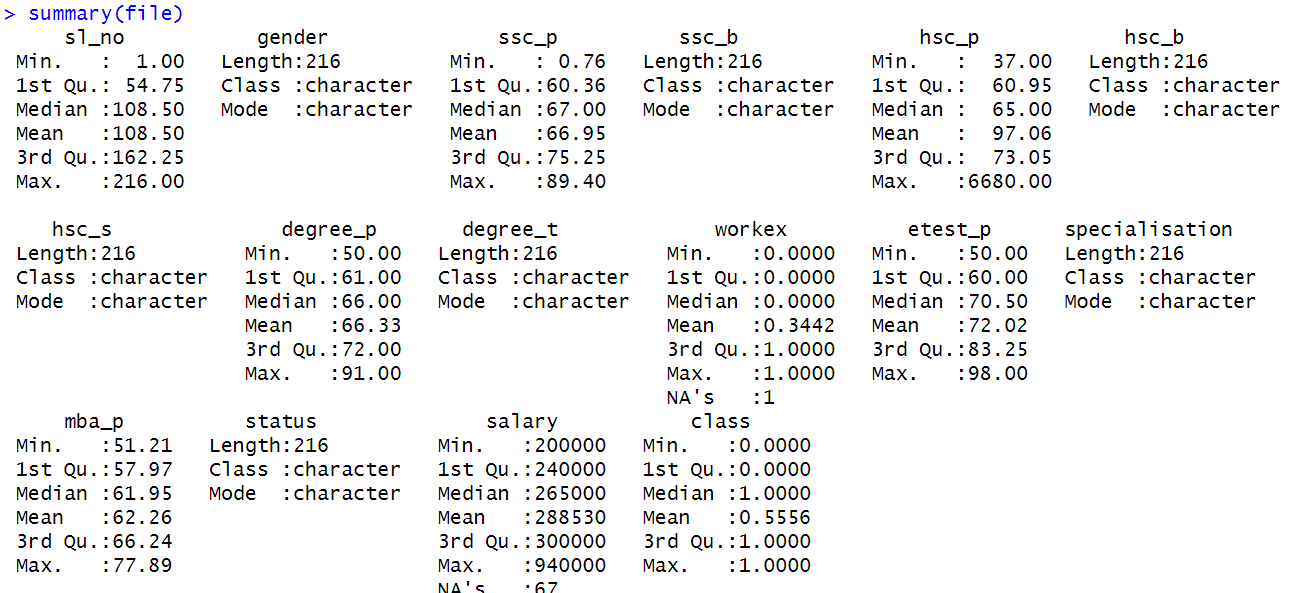
It shows the dimension of the dataset. (216 rows, 16 columns)



names(file) is used to display column names.



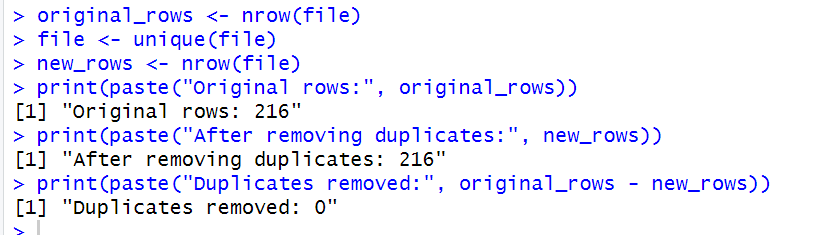
head(files) show the first 6 rows of the dataset.



The summary() function provides a statistical summary of an object, which includes measures like minimum, maximum, median, and quartiles for each variable.

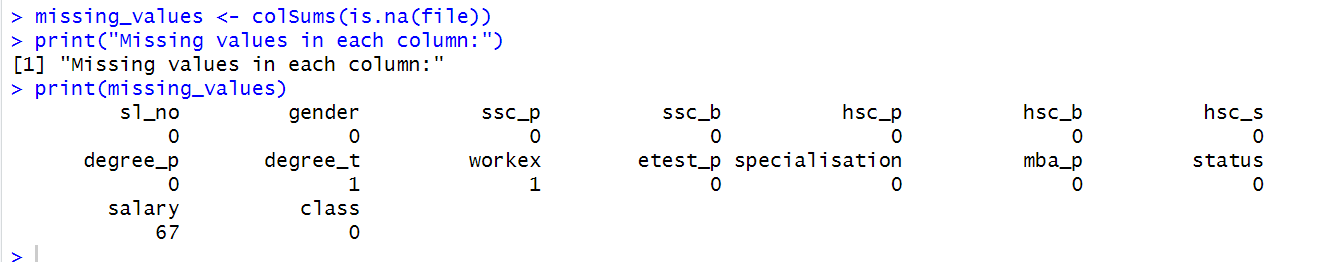
**Data pre-processing :**

***Handling duplicate values -***



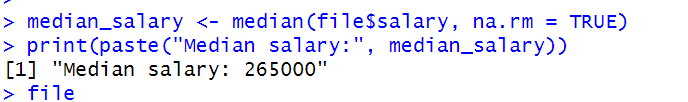
To ensure the dataset was clean, duplicate rows were identified and removed using the unique() function. First, the total number of original rows was recorded. After removing duplicates, the new number of rows was compared with the original to find out how many duplicates were eliminated. Removing duplicates is important because repeated records can affect the accuracy of data analysis.

***Finding missing values -***

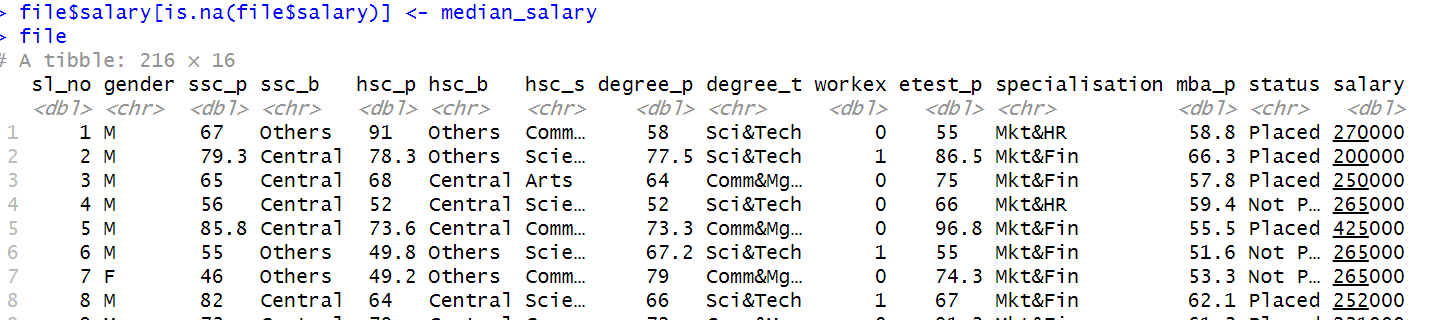


To find missing values in the dataset, the is.na() function was used to check each cell for missing (NA) entries. Then, colSums(is.na(file)) was applied to count how many missing values were present in each column. Finally, the missing values were printed to clearly see which columns had missing data and how many entries were missing. Finding missing values is important because missing data can affect the accuracy and quality of the analysis.

***Handling missing values –***

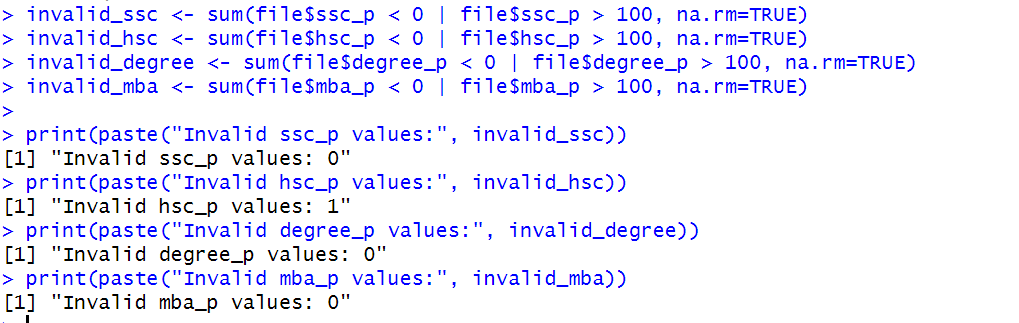


In this step, the median salary was calculated using the median() function, while ignoring any missing values (na.rm = TRUE). The calculated median was then printed to check its value. This was done because the salary column had some missing entries, and using the median is a good way to fill in missing numeric values without being affected by outliers. Finding the median first helps to prepare for replacing missing salary values in the next step, which ensures the dataset stays complete and ready for analysis.



After calculating the median salary, this line was used to fill in the missing (NA) values in the salary column. The is.na(file$salary) part checks which rows have missing salaries, and those values are replaced with the previously calculated median. This ensures that the dataset has no empty salary entries.

***Checking for invalid values –***



In this part, it was checked that if there were any wrong values in the percentage columns like ssc\_p, hsc\_p, degree\_p, and mba\_p. A percentage should always be between 0 and 100. So, a condition was used to find if any values were less than 0 or greater than 100. Then, it was counted how many wrong values there were and printed them.

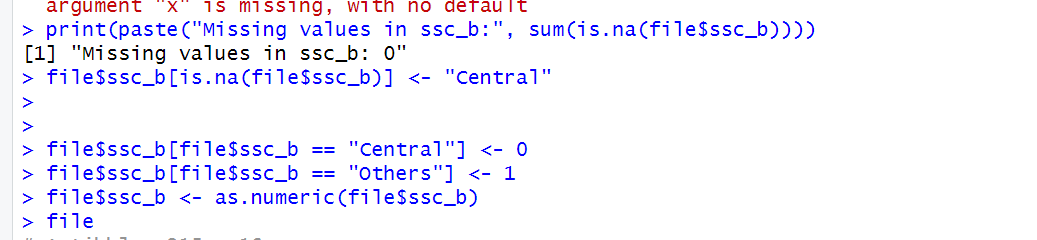
***Correcting invalid percentage values by replacing them with the median -***

A computer code with blue text

AI-generated content may be incorrect.

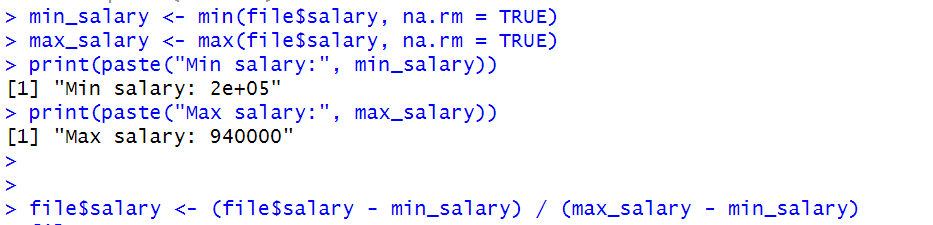
In this step, we first corrected invalid percentage values in the columns ssc\_p, hsc\_p, degree\_p, and mba\_p. Any value less than 0 or greater than 100 was considered invalid and replaced with the median of that respective column, instead of removing the record.

***Converting ssc\_b from categorical to numeric –***



We handled missing values in the ssc\_b column, which represents the type of board (either Central or Others). First, we checked for missing values and found any occurrences of NA. We then replaced these missing values with Central as a default value, assuming it’s the most common category. After handling the missing values, we converted the ssc\_b column from categorical values to numeric values for easier analysis and modeling. We assigned "Central" a value of 0 and "Others" a value of 1, then converted the entire column to a numeric format.

***Normalizing –***



**Normalizing salary value using min-max**

In this part we normalized the salary column using the Min-Max normalization technique to scale the salary values between 0 and 1. First, we calculated the minimum and maximum salary values in the dataset. Then, using the Min-Max formula, we transformed the salary values by subtracting the minimum salary and dividing by the range (maximum salary - minimum salary). This ensures that all salary values are on the same scale.

***Balancing dataset –***

A computer screen shot of a computer code

AI-generated content may be incorrect.

In this step, we checked how many students were placed (class = 1) and not placed (class = 0) by counting their numbers using the sum() function. Then, we compared the two counts using an if-else condition to identify which group had more students (majority class) and which had fewer (minority class). We saved the counts into variables for further balancing. This step was important because we needed to balance the dataset later, and first identifying the majority and minority classes was necessary to do that correctly.

***Converting workex to categorical –***

A close-up of a number

AI-generated content may be incorrect.

In this step, we converted the workex attribute from numeric values (0 and 1) to categorical values (No and Yes). We replaced 0 with No and 1 with Yes.

A close-up of a math equation

AI-generated content may be incorrect.

In this step, we first used set.seed(123) to make sure the random sampling gives the same result every time the code is run. Then, we found the rows where students belonged to the majority class using the which() function. From the majority of students, we randomly selected a number of samples equal to the number of minority students using sample(). We combined these sampled majority students with all minority students using c(), and finally updated the dataset by keeping only these selected rows. This was done to create a balanced dataset.

***Split into training and test sets –***

A computer code with blue text

AI-generated content may be incorrect.

In this step, we split the balanced dataset into a training set and a testing set. First, we set a seed using set.seed(123) to make sure the random split is the same every time the code is run. We calculated 80% of the total rows for training and selected random rows using the sample() function. The selected rows became the train\_data, and the remaining rows became the test\_data. This split is important to train the model on one part of the data and test its performance on unseen data to check how well it generalizes.

A computer code with blue text

AI-generated content may be incorrect.

In this step, we checked the class distribution in both the training and testing sets by using the table() function on the class column. We printed how many students were placed and not placed in each set. This check was important to confirm that even after splitting, both the training and testing sets remained balanced, so that the model would learn and perform properly without being biased toward any class.

***Central tendency measure for ssc\_p –***

A computer code with blue text

AI-generated content may be incorrect.

We calculated the central tendency measures (mean, median, and mode) for the ssc\_p (SSC Percentage) to understand the typical performance of students in this column. The mean was calculated using the mean() function, the median using median(), and the mode by finding the most frequent value using table() and which.max(). These measures help us summarize the data and understand the general trend of SSC percentages.

***Spread measure for ssc\_p –***

A computer code with blue text

AI-generated content may be incorrect.

We calculated the spread measures for ssc\_p (SSC Percentage) to understand the variability in the data. The range was computed by subtracting the minimum value from the maximum value. The Interquartile Range (IQR) was calculated using the 25th and 75th percentiles (Q1 and Q3), showing the spread of the middle 50% of the data. Variance and standard deviation were calculated using var() and sd() to measure how spread out the values are from the mean. These measures help us understand the data distribution and identify potential outliers or variations in student performance.

***Central tendency and spread for MBA percentage –***

A screenshot of a computer code

AI-generated content may be incorrect.

We calculated the central tendency measures (mean, median, and mode) to determine the typical performance of the students. The mean provides an average score, the median shows the middle value when sorted, and the mode identifies the most common score. To understand the variability in the data, we calculated the spread measures, including range, interquartile range (IQR), variance, and standard deviation. These measures highlight the extent to which the MBA percentages deviate from the average. By using functions like max(), min(), IQR(), var(), and sd(), we gained insight into the consistency and dispersion of MBA percentages.