

University of Wales Trinity Saint David

School of Engineering



Mechanical and Manufacturing Engineering

Engineering Applications

Sheunesu Ziki

2205423

05/04/24

Table of Contents

1	Introduction	iii
2	Literature review	iii
2.1	Statistical analysis	iii
2.2	Types of statistical analysis.....	iii
3	Data importation and summary statistics	iii
4	Correlation analysis.....	v
5	Hypothesis test	vii
6	Histcounts and histogram	viii
7	Conclusion	x
8	References	x

1 Introduction

This report will look into performing a number of statistical analyses between a number of student grades and other categories such as their gender and age. These analyses will be performed in Matlab by importing a dataset that contains the student grades and the additional information on the students. This data set will then have its data extracted into Matlab, which will then be used to perform these statistical analyses such as correlation analyses. The objective of performing these statistical analyses is to see the factors that may affect a student's performance such as gender or age.

2 Literature review

2.1 Statistical analysis

Statistical analysis is the process of gathering and examining massive amounts of data in order to spot patterns and derive insightful knowledge is known as statistical analysis (Dixon and Massey Jr., 1951). This is done by using unprocessed data to determine correlations between variables to identify trends and patterns.

2.2 Types of statistical analysis

There are two main types of statistical analysis which are descriptive and inferential analysis. Descriptive statistics provides an overview of the data without making any inferences about the information included in a data set (Dixon and Massey Jr., 1951). When performing descriptive statistics, data visualization is frequently used to show information to others in an easily understood way using graphs, tables, and charts.

Inferential statistics advances the findings of descriptive statistics, by deriving conclusions from the data and offering recommendations (Dixon and Massey Jr., 1951). Businesses frequently utilize inferential statistical analysis to help them make decisions, and scientists frequently use it to discover novel links between variables in their research (Bersenev, Yachie-Kinoshita and Palaniappan, 2024).

3 Data importation and summary statistics

The first step before conducting summary statistics it's to import the data set into Matlab so that the data it contains can be used. Below is figure 1 showing the function used to import the spreadsheet dataset:

```
% Importing data
D = readtable('student_grades.xlsx.csv');
```

Figure 1: Matlab function for importing dataset.

Figure 1 shows that the command `readtable` was used for importing the dataset. This is because the dataset contains both categorical and numerical data. This function allows for both of these data types to be represented. It is also noted that the data is under the variable `D` this is to help it to be easier to recall this data later on in the code and for the data to be established.

Summary statistics is utilized in descriptive statistics to provide a huge amount of information in the most straightforward way. The second step in the task needs for summary statistics to be computed for each numerical variable for the mean, median, mode, range, and standard deviation. These will be performed using the Matlab functions `mean`, `median`, `mode`, `min`, `max`, and `std`. Figure 2 below shows how the functions were used:

```
% calculation of the mean, mode, median, min, max and std
AgeMean = mean(table2array(D(:,3)));
AttdMean = mean(table2array(D(:,4)));
Ex1Mean = mean(table2array(D(:,5)));
Ex2Mean = mean(table2array(D(:,6)));
AgeMedian = median(table2array(D(:,3)));
AttdMedian = median(table2array(D(:,4)));
Ex1Median = median(table2array(D(:,5)));
Ex2Median = median(table2array(D(:,6)));
AgeMode = mode(table2array(D(:,3)));
AttdMode = mode(table2array(D(:,4)));
Ex1Mode = mode(table2array(D(:,5)));
Ex2Mode = mode(table2array(D(:,6)));
AgeMin = min(table2array(D(:,3)));
AttdMin = min(table2array(D(:,4)));
Ex1Min = min(table2array(D(:,5)));
Ex2Min = min(table2array(D(:,6)));
AgeMax = max(table2array(D(:,3)));
AttdMax = max(table2array(D(:,4)));
Ex1Max = max(table2array(D(:,5)));
Ex2Max = max(table2array(D(:,6)));
AgeStd = std(table2array(D(:,3)));
AttdStd = std(table2array(D(:,4)));
Ex1Std = std(table2array(D(:,5)));
Ex2Std = std(table2array(D(:,6)));
O_All = (table2array(D(:,5))+table2array(D(:,6)))/2;
O_AllMean = mean(O_All);
O_AllMedian = median(O_All);
O_AllMode = mode(O_All);
O_AllMin = min(O_All);
O_AllMax = max(O_All);
O_AllStd = std(O_All);
```

Figure 2: Matlab functions for performing summary statistics.

The function `(table2array(D(:,3)))` converts the table, `D`, to a homogeneous array, `AgeMean`. The variables in `D` become columns in `AgeMean`. The `(:,3)` means that this function is taking all rows of column 3. This function does not include the table properties.

The final step in the summary statistics is the presentation of the data in a table format. To do this the table will have to be populated with the appropriate variables in the appropriate positions.

Below is the code that was used in creating the table:

```
% creating the data table
Variable = ["Age"; "Attendance"; "1st Exam Score"; "2nd Exam Score"; "O/All"];
Mean = [AgeMean, AttdMean, Ex1Mean, Ex2Mean, O_AllMean]';
Median = [AgeMedian, AttdMedian, Ex1Median, Ex2Median, O_AllMedian]';
Mode = [AgeMode, AttdMode, Ex1Mode, Ex2Mode, O_AllMode]';
Range = [AgeMin AgeMax; AttdMin AttdMax; Ex1Min Ex1Max; Ex2Min Ex2Max; O_AllMin O_AllMax];
Std = [AgeStd, AttdStd, Ex1Std, Ex2Std, O_AllStd]';
T = table(Variable, Mean, Median, Mode, Range, Std);
disp(T)
```

Figure 3: Matlab code showing the creation of the table.

From the above code the variable (Variable) creates the first column and the six rows of the table. The variables Mean, Median, Mode, Range, and Std create our columns and populates the table with the calculations that were done in Figure 2. The function table(Variable, Mean, Median, Mode, Range, Std) creates the table with all the input variables to populate the table, which only works when the variables have the same number of rows. To display the table in the command window the function disp(T) is used as the function displays the value of variable T.

Variable	Mean	Median	Mode	Range	Std
"Age"	36.65	37.5	48	19 53	9.7643
"Attendance"	60.58	57	27	20 100	24.249
"1st Exam Score"	58.27	58	59	7 98	21.673
"2nd Exam Score"	55.08	53.5	51	4 98	22.428
"O/All"	56.675	54.5	25.5	9.5 95	21.452

Figure 4: Image showing the table displayed in the command window.

4 Correlation analysis

The first task is to return in the command window a matrix of the pairwise linear correlation coefficients between the variables age, attendance, exam score 1, exam score 2 and O/All using the function corr. Below is an image showing the code used to complete this task:

```
% creating correlation plot
V = [table2array(D(:,3:6)), O_All];
correlationMatrix = corr(V);
format("shortG")
disp('Correlation Matrix:')
disp(corr(V))
variable_names = {'Age', 'Attendance', 'Exam 1 Score', 'Exam 2 Score', 'O/All'};
corrplot(V, 'VarNames', variable_names, TestR="on")
```

Figure 5: Matlab code showing correlation matrix and corrplot.

In Figure 5 we see the variable V is created by all rows from column three to six of variable D. This is then used to create the variable correlationmatrix using the function `corr(V)`. This function returns a matrix of the pairwise linear correlation coefficient between each pair of columns in the input matrix V.

```
Correlation Matrix:
      1      -0.036206      -0.036398      -0.050609      -0.044842
    -0.036206      1      0.89123      0.82972      0.88394
    -0.036398      0.89123      1      0.89286      0.9719
    -0.050609      0.82972      0.89286      1      0.97378
    -0.044842      0.88394      0.9719      0.97378      1
```

Figure 6: Image showing the correlation Matrix in the command window.

From Figure 5 the function `disp('Correlation Matrix:')` displays in the command window the (Correlation Matrix:) as seen in Figure 6. The function `disp(corr(V))` displays the correlation matrix in the command window as seen in Figure 6. The function `format("shortG")` changes the output display format to display numeric values in a compact form with 5 total digits.

The variable (`variable_names = {'Age', 'Attendance', 'Exam 1 Score', 'Exam 2 Score', 'O/All'}`) shown in Figure 5 contains the variable names that will be used in the visual correlation plot. The function `corrplot(V, 'VarNames', variable_names, TestR="on")` creates a correlation plot using the variable V and the VarNames argument variable_names and the TestR argument on to flag any significant correlations in the correlation's matrix. Below is an image of the correlation plot produced after implementing the Matlab code in figure 5:

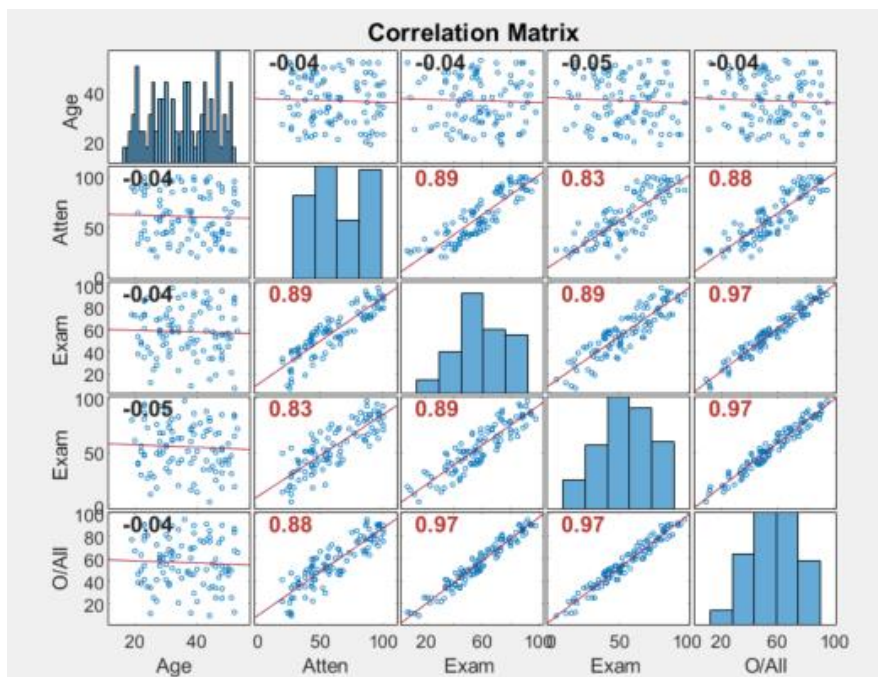


Figure 7: Image showing the correlation plot.

From Figure 7 the numerical values in red are the flagged significant correlations in the correlation's matrix. Figure 7 shows that exam scores and attendance seem to be strongly positively connected (correlation coefficient ≈ 0.89). This means that an improved attendance could lead to improved academic performance. This same outcome is also seen for the overall exam score. There is a negative association (correlation coefficient ≈ -0.04) between exam scores and age in the scatter plot. This suggests that exam performance is not greatly impacted by age alone. This is also the same result for age and overall exam score. This shows that students that have a higher attendance tend to perform better than those who don't, but this is not always the case.

5 Hypothesis test

In this segment of the report an examination of the relationship between the categorical variable gender and the numerical variable overall exam score will be performed in order to see if the gender of the students affect their exam scores. This will be done by performing a hypothesis test known in Matlab as a `ttest2`. A `ttest2` is a two sample test used for comparing information from two separate random samples drawn from various populations (*Air Medical Journal_20240507*). It is used to find out if there is a statistically significant difference between the variables (*Air Medical Journal_20240507*).

The first task in this segment is to organise the data into the categorical and numerical data sets.

```
Gender = table2array(D(:,2));  
male_indices = find(strcmp(Gender, 'Male'));  
female_indices = find(strcmp(Gender, 'Female'));  
male_scores = O_All(male_indices);  
female_scores = O_All(female_indices);
```

Figure 8: Image showing how the data was grouped.

In Figure 8 we see the code (`male_indices = find(strcmp(Gender, 'Male'))`) and (`female_indices = find(strcmp(Gender, 'Female'))`) which is used to find from the data set the categorical data (Male) and (Female) which will help in our gender variable. The code (`male_scores = O_All(male_indices)`) and (`female_scores = O_All(female_indices)`) takes the overall exam score that are related to male and female gender categories. Grouping the data makes it easier to perform the hypothesis test.

```

% Two-sample t-test
[h, p_value] = ttest2(male_scores, female_scores);
fprintf('Two-sample t-test results:\n');
fprintf('p-value = %.4f\n', p_value);
% Display of the test results
if h
    fprintf('    There is a significant difference in exam results between genders.\n');
else
    fprintf('    There is no significant difference in exam results between genders.\n');
end

```

Figure 9: Image showing two sample test.

Figure 9 shows the code of how to perform the two sample test. The code `[h, p_value] = ttest2(male_scores, female_scores)` returns the null hypothesis and the p-value, `p`, of the test, using the `male_scores` and the `female_scores`. The null hypothesis means the population means of the two groups are equal and if the p-value is less than the chosen significance level then the null hypothesis will be rejected. The if/else means that if the null hypothesis is higher than the p value then there is a significant difference in exam results between genders and if it is lower than the p value there is no significant difference in exam results between genders.

```

Two-sample t-test results:
p-value = 0.4370
    There is no significant difference in exam results between genders.

```

Figure 10: Image showing the two sample test results.

Figure 10 shows that there is no significant difference between the exam results and the gender which means that gender does not affect the student's exam results.

6 Histcounts and histogram

In Matlab the function `histcounts` is used to partition data into bins and compute the histogram bin counts for a given variable.

```

%histcounts
Edges=0:10:100;
F1 = histcounts(table2array(D(:,5)),Edges);
F2 = histcounts(table2array(D(:,6)),Edges);
F3 = histcounts(O_All,Edges);

```

Figure 11: Image showing code for hitscounts.

The above code returns a vector that indicates the number of elements in the exam 1, exam 2, and overall exam results whose value is equal to each of the mentioned variable's categories. This determines the frequency of each of these variables which will be used in creating the histogram.

A graphical depiction of the frequency distribution of data is called a histogram. Below is the code that is used in creating the histograms:

```

%creating histogram
figure;
subplot(1,3,1);
h2 = histogram('BinEdges',Edges,'BinCounts',F1);
ylim([0 22]);
xlim([0 100]);
xlabel('Class Limits')
ylabel('Frequency')
title('Exam 1 Histogram')
subplot(1,3,2);
h3 = histogram('BinEdges',Edges,'BinCounts',F2);
ylim([0 22]);
xlim([0 100]);
xlabel('Class Limits');
ylabel('Frequency');
title('Exam 2 Histogram');
subplot(1,3,3);
h4 = histogram('BinEdges',Edges,'BinCounts',F3);
ylim([0 22]);
xlim([0 100]);
xlabel('Class Limits')
ylabel('Frequency')
title('O/All Results Histogram ');

```

Figure 12: Image showing code for creating the histograms.

The above code shows how the histograms were created. The function histogram was used with the bin edges that were defined in figure 11 which is a range from 0 to 100 with a step of 10. The bin counts are our variables which are exam1, exam 2, and overall exam results. The ylim sets our limits from 0 to 22.

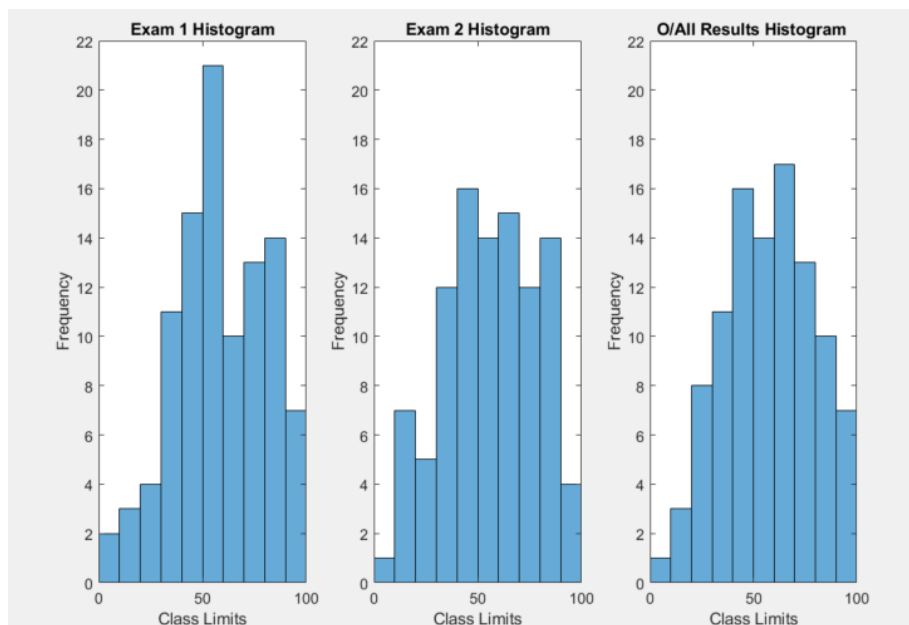


Figure 13: Image showing the histograms created.

The above image shows the histograms that were created using the code from figure 12.

7 Conclusion

This report has proved that there are factors that affect students grades and some that do not. the key findings are that gender and age do not affect student's exam results in terms of doing well whereas attendance affects a student's results in terms of doing well. This has been seen in many investigations worldwide that if a student does not attend a lecture regularly they are prone to have low marks compared to those that attend regularly. In the same worldwide investigations, we see that age and gender don not affect performance as people who have attended education in older age have performed the same or better than those of younger age. Although these statistical methods are good we see some limitation such as in summary statistics where the data may be over simplified which results in a loss of detail.

8 References

Air Medical Journal_20240507.

Bersenev, D., Yachie-Kinoshita, A. and Palaniappan, S.K. (2024) *Replicating a High-Impact Scientific Publication Using Systems of Large Language Models* Cold Spring Harbor Laboratory.

Dixon, W.J. and Massey Jr., F.J. (1951) *Introduction to statistical analysis* New York, NY, US: McGraw-Hill, pp. x, 370.

9 Appendix

```
clc
clear
% Importing data
D = readtable('student_grades.xlsx.csv');
% calculation of the mean, mode, median, min, max and std
AgeMean = mean(table2array(D(:,3)));
AttdMean = mean(table2array(D(:,4)));
Ex1Mean = mean(table2array(D(:,5)));
Ex2Mean = mean(table2array(D(:,6)));
AgeMedian = median(table2array(D(:,3)));
AttdMedian = median(table2array(D(:,4)));
Ex1Median = median(table2array(D(:,5)));
Ex2Median = median(table2array(D(:,6)));
AgeMode = mode(table2array(D(:,3)));
AttdMode = mode(table2array(D(:,4)));
Ex1Mode = mode(table2array(D(:,5)));
Ex2Mode = mode(table2array(D(:,6)));
AgeMin = min(table2array(D(:,3)));
AttdMin = min(table2array(D(:,4)));
Ex1Min= min(table2array(D(:,5)));
Ex2Min = min(table2array(D(:,6)));
AgeMax = max(table2array(D(:,3)));
AttdMax = max(table2array(D(:,4)));
Ex1Max = max(table2array(D(:,5)));
Ex2Max = max(table2array(D(:,6)));
AgeStd = std(table2array(D(:,3)));
AttdStd = std(table2array(D(:,4)));
```

```

Ex1Std = std(table2array(D(:,5)));
Ex2Std = std(table2array(D(:,6)));
O_All = (table2array(D(:,5))+table2array(D(:,6)))/2;
O_AllMean = mean(O_All);
O_AllMedian = median(O_All);
O_AllMode = mode(O_All);
O_AllMin = min(O_All);
O_AllMax = max(O_All);
O_AllStd = std(O_All);
% creating the data table
Variable = ["Age";"Attendance";"1st Exam Score";"2nd Exam Score";"O/All"];
Mean = [AgeMean,AttdMean,Ex1Mean,Ex2Mean,O_AllMean]';
Median = [AgeMedian,AttdMedian,Ex1Median,Ex2Median,O_AllMedian]';
Mode = [AgeMode,AttdMode,Ex1Mode,Ex2Mode,O_AllMode]';
Range = [AgeMin AgeMax;AttdMin AttdMax;Ex1Min Ex1Max;Ex2Min Ex2Max;O_AllMin
O_AllMax];
Std = [AgeStd,AttdStd,Ex1Std,Ex2Std,O_AllStd]';
T = table(Variable,Mean,Median,Mode,Range,Std);
disp(T)
% creating correlation plot
V = [table2array(D(:,3:6)),O_All];
correlationMatrix = corr(V);
disp('Correlation Matrix:')
disp(corr(V))
variable_names = {'Age', 'Attendance', 'Exam 1 Score', 'Exam 2 Score', 'O/All'};
corrplot(V, 'VarNames', variable_names,TestR="on")
%hypothesis test
Gender = table2array(D(:,2));
male_indices = find(strcmp(Gender, 'Male'));
female_indices = find(strcmp(Gender, 'Female'));
male_scores = O_All(male_indices);
female_scores = O_All(female_indices);
% Two-sample t-test
[h, p_value] = ttest2(male_scores, female_scores);
fprintf('Two-sample t-test results:\n');
fprintf('p-value = %.4f\n', p_value);
% Display of the test results
if h
    fprintf(' There is a significant difference in exam results between
    genders.\n');
else
    fprintf(' There is no significant difference in exam results between
    genders.\n');
end
%histcounts
Edges=0:10:100;
F1 = histcounts(table2array(D(:,5)),Edges);
F2 = histcounts(table2array(D(:,6)),Edges);
F3 = histcounts(O_All,Edges);
%creating histogram
figure;
subplot(1,3,1);
h2 = histogram('BinEdges',Edges,'BinCounts',F1);
ylim([0 22]);
xlim([0 100]);
xlabel('Class Limits')
ylabel('Frequency')
title('Exam 1 Histogram')
subplot(1,3,2);

```

```

h3 = histogram('BinEdges',Edges,'BinCounts',F2);
ylim([0 22]);
xlim([0 100]);
xlabel('Class Limits');
ylabel('Frequency');
title('Exam 2 Histogram');
subplot(1,3,3);
h4 = histogram('BinEdges',Edges,'BinCounts',F3);
ylim([0 22]);
xlim([0 100]);
xlabel('Class Limits')
ylabel('Frequency')
title('O/All Results Histogram ');
%ogives
cumulativeExam1 = cumsum(F1);
R = F1/sum(F1);
RCF = cumsum(R);
ICRF = cumsum(R, 'reverse');
ICRF1 = cumsum(R);
cumulativeExam2 = cumsum(F2);
relativeExam2 = F2/sum(F2);
RCF2 = cumsum(relativeExam2);
ICRF2 = cumsum(relativeExam2, 'reverse');
cumulativeO_All = cumsum(F3);
relativeO_All = F3/sum(F3);
RCF3 = cumsum(relativeO_All);
ICRF3 = cumsum(relativeO_All, 'reverse');
figure;
hold on
subplot(1,3,1);
plot(1:10:100,RCF,'Marker','o','MarkerFaceColor','b')
plot(1:10:100,ICRF,'Marker','o','MarkerFaceColor','b')
subplot(1,3,2);
plot(1:10:100,RCF2,'Marker','o','MarkerFaceColor','b')
plot(1:10:100,ICRF2,'Marker','o','MarkerFaceColor','b')
subplot(1,3,3);
plot(1:10:100,RCF3,'Marker','o','MarkerFaceColor','b')
plot(1:10:100,ICRF3,'Marker','o','MarkerFaceColor','b')

```