Assessment of Academic Performances

of WM-ASDS students

MD. RABBI AMIN

11/24/22

WM-ASDSNC01

Objective

The document is a graded assignment for the non-credit course, titled “Statistical Methods” having course code WM-ASDSNC01. It contains an analysis of the historical academic records of students against their exam performance in the current course. It also considers their employment status and income information. As a standard practice, all Personal Identifiable Information (PII) has been removed.

Target Audience

The target audience for this report is our honourable instructor, Dr. XYZ, and the students of this class. The author claims to copyright and assertion of credit whatsoever on the contents, and so, the report would be distributable for learning purposes to all students of the WM-ASDS program.

Tools used

Microsoft Excel – for manual sampling of data

Python 3 – for data loading, cleaning and analysis

IDE - Anaconda Jupyter

List of Contents

List of Figures

List of Tables

List of Code Segments

Dataset

The dataset is collected through an online form. Students input their SSC, HSC, and undergraduate GPAs, distance in kilometres from the university, daily study hours, employment status, monthly income, and exam scores.

All the data in the dataset are of numeric type except for the employment status which is a dichotomous categorical variable, having options as 'Yes' and 'No' as responses.

Data preparation

is the most important and most tedious task prior to analysis. This is how we performed it for this assignment.

Initially, we sampled the data to contain only 20 student records. We performed this sampling manually as the dataset is small, and some students did not input all values, so those partial or blank entries are not suitable for analysis.

After the manual sampling, we loaded the dataset into Python using xlwings library. The initial dataset is of a type list, and so we converted it into a data frame – a data type which is more suitable for analysis. While converting, we gave the column names in the Python code.

After converting it into a data frame, we cleaned the data. We removed the top row that had the excel column headers. Then we dropped the serial number, name, id, and section columns and thus got rid of all PIIs. A snapshot of the cleaned data frame is below.

	SSC_GPA	HSC_GPA	Undergrad_GPA	distance_JU	daily_study	employed	monthly_income	exam_score
1	4.2	4.69	2.6	26	0.2	No	0	19
2	5	4.83	3.49	36	0.5	Yes	40000	10
3	5	5	3.26	28	0.5	Yes	40000	11
4	5	5	3.56	18	0.5	Yes	100000	21
5	5	5	3.82	22	1	No	8000	21
6	5	5	3.26	23	3	yes	42000	16
7	5	5	3.09	30	1	Yes	60000	15
8	5	4.58	3.32	35	0.5	Yes	18000	10

Figure 1. Snapshot of initial dataset

For better analysis, we will add another categorical variable 'performance', which is based on our exam score. If the exam score is less than 12, we call it 'poor', from 13 to 19, we term it 'moderate', and for 20+ score, we term it a 'high' performance. The code below achieves this for us.

```
In [101]: def performance(value):
              if value <= 12:
                  return "poor"
              elif 13 <= value < 20:
                  return "moderate"
              elif value >= 20:
                  return "high"

          df['performance'] = df['exam_score'].map(performance)
```

Code Segment 1. Adding categorical exam performance

Now the data frame looks like below.

	SSC_GPA	HSC_GPA	Undergrad_GPA	distance_JU	daily_study	employed	monthly_income	exam_score	performance
1	4.2	4.69	2.6	26	0.2	No	0	19	moderate
2	5	4.83	3.49	36	0.5	Yes	40000	10	poor
3	5	5	3.26	28	0.5	Yes	40000	11	poor
4	5	5	3.56	18	0.5	Yes	100000	21	high
5	5	5	3.82	22	1	No	8000	21	high
6	5	5	3.26	23	3	yes	42000	16	moderate
7	5	5	3.09	30	1	Yes	60000	15	moderate

Figure 2. Snapshot of dataset after adding categorical performance

The column ‘employed’ and ‘performance ‘ are now categorical data, and so, we encode them. For employed people, we will encode Yes as 1, and No as 0. For ‘performance’, we will encode ‘poor’ as 0, ‘moderate’ as 1, and ‘high’ as 2. Below is the code segment and data frame snapshot after encoding.

```
encode_values = {"employed": {"Yes":1, "yes":1, "No":0, "no":0},
                 "performance":{"poor":0,"moderate":1,"high":2}}
df=df.replace(encode_values)
```

Code Segment 2. Encoding categorical variables

This will transform the two categorical variables ‘employed’ and ‘performance’ as the key-value pair provided in the dictionary-type variable encode values above. The pandas replace() function performs the actual value to code transformation. Below is what we get after encoding.

	SSC_GPA	HSC_GPA	Undergrad_GPA	distance_JU	daily_study	employed	monthly_income	exam_score	performance
1	4.2	4.69	2.6	26	0.2	0	0	19	1
2	5	4.83	3.49	36	0.5	1	40000	10	0
3	5	5	3.26	28	0.5	1	40000	11	0
4	5	5	3.56	18	0.5	1	100000	21	2
5	5	5	3.82	22	1	0	8000	21	2
6	5	5	3.26	23	3	1	42000	16	1
7	5	5	3.09	30	1	1	60000	15	1
8	5	4.58	3.32	35	0.5	1	18000	10	0
9	5	5	2.71	25	0.5	1	80000	24	2

Figure 3. Dataset snapshot after encoding

We also made sure that the data types are exactly what we want them to be by explicit type casting. The code we used for this is below.

```
df['SSC_GPA']= df['SSC_GPA'].astype('float')
df['HSC_GPA']= df['HSC_GPA'].astype('float')
df['Undergrad_GPA']= df['Undergrad_GPA'].astype('float')
df['daily_study']= df['daily_study'].astype('float')
df['exam_score']= df['exam_score'].astype('float')
df['distance_JU']= df['distance_JU'].astype('float')
df['monthly_income']= df['monthly_income'].astype('int')
df['performance'] = pd.Categorical(df.performance)
df['employed'] = pd.Categorical(df.employed)
```

Code Segment 3. Explicit type conversion

Before delving into analysis, this snapshot confirms us the types of data in the dataframe.

```
df.dtypes
```

```
SSC_GPA             float64
HSC_GPA             float64
Undergrad_GPA       float64
distance_JU         float64
daily_study         float64
employed           category
monthly_income        int32
exam_score          float64
performance        category
dtype: object
```

Figure 4. Data types in the final data frame

This dataset is more suitable for analysis.

Analysis

In this section, we will consider the exam score and categorical performance as the dependent variable, and the previous results, distance, study hours, employment status and a monthly income as independent variables.

Employment status versus exam performance

Boxplots of exam scores and performance categories against the employment status is portrayed below.

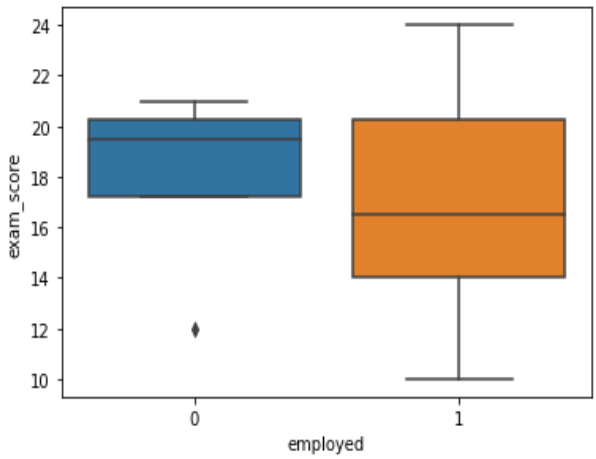
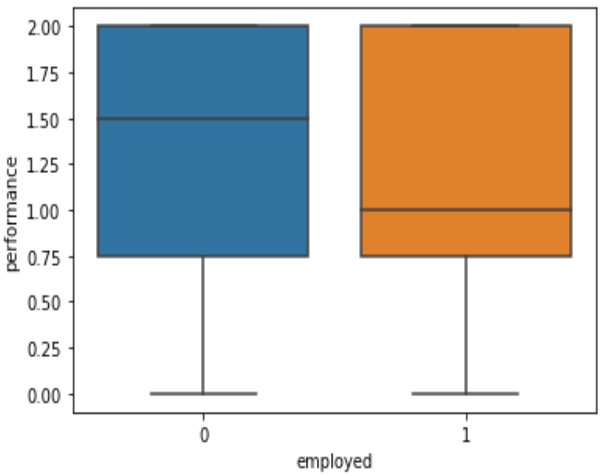
sns.boxplot(data=df, x='employed', y='exam_score')	sns.boxplot(data=df, x='employed', y='performance')
	

Table 1. Exam score and performance by employment status

Here we can see from both the boxplots that the median value of unemployed students are much higher than the employed students. It is understandable that students with employment may get lesser time for exam preparation. But it is also notable from the first boxplot that the higher scores in exam are achieved by those who are employed. This might be because of the fact that they may have some prior experience working with data. But since we do not have enough details to verify this assumption, we cannot further dig into this.

The overall impression is that unemployed students on average perform better than those who are employed, but some employed students also perform outstandingly.

Distance from JU versus exam performance

```
df.distance_JU.describe()
```

```
count    20.000000
mean     33.200000
std      15.453155
min      18.000000
25%      24.500000
50%      30.000000
75%      35.250000
max      90.000000
```

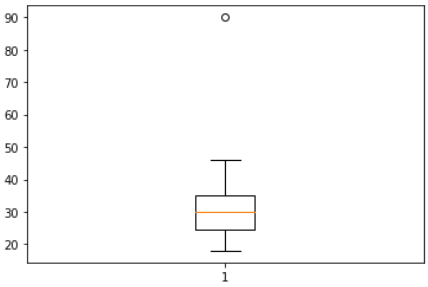
The distance in kilometres from a student's residence to the university impacts the time to reach the university on class days for that student. Larger distance may result in

fatigue for a student which may in turn result in that student's inability to concentrate in classes. This might impact exam preparation and result. So, this is a variable of interest while assessing a student's grades.

Figure 4 displays the sample distribution for this variable.

Figure 5. The numerical summary of the distance variable

in figure 5 shows a minimum value of 18 km, and a maximum distance of 90 km, with an average distance of 33.2 km, and a median distance of 30 km. The inter-quartile distance of the distance variable is only (35.25 – 24.5) = 10.75 km.



So, the whiskers will spread up to (10.75 X 1.5) = 16.125 km on both sides, limited by the existence of data points on either side. So, clearly, the maximum distance will be considered an outlier. Below is the boxplot of this variable of interest, rendered by code plt.boxplot(df.distance_JU).

Figure 6. Boxplot of the distance variable

Now, we have to assess the impact of distance on exam performance, if there is any. To do this, we plot the exam scores against the distance variable. Since both are numeric, a scatter plot would be appropriate for such an analysis.

```
plt.scatter(df.exam_score,df.distance_JU)
plt.xlabel('exam scores')
plt.ylabel('distance (km)')
plt.show()
```

Code Segment 4. Scatterplot of exam scores against distance

The code snippet above renders the following plot.

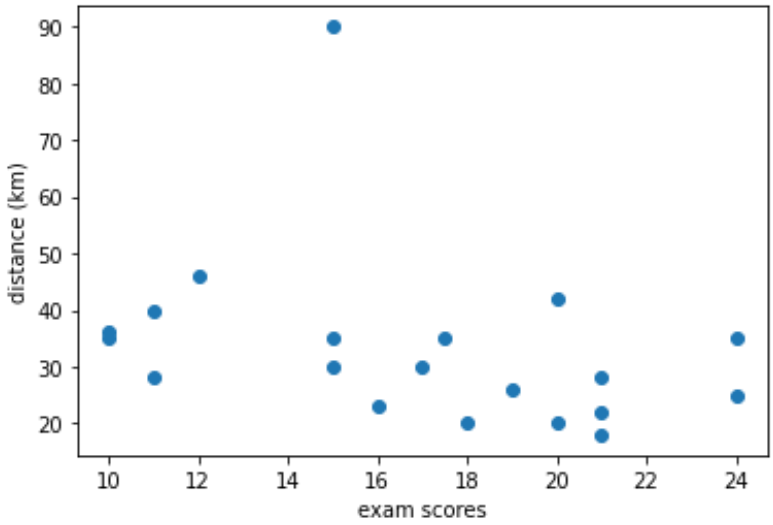


Figure 7. Scatterplot of exam scores against distance

Here, we can see a trend of higher exam scores for lower distances. We can further assess via boxplots for distances versus exam performances.

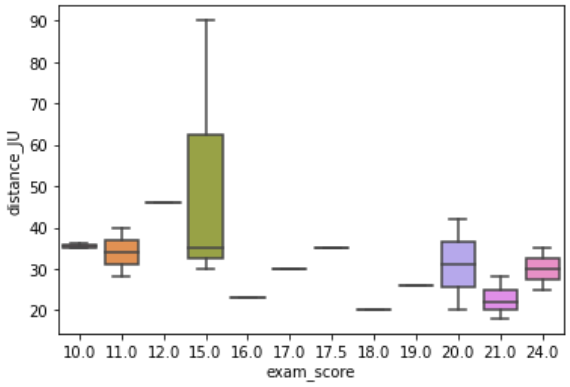
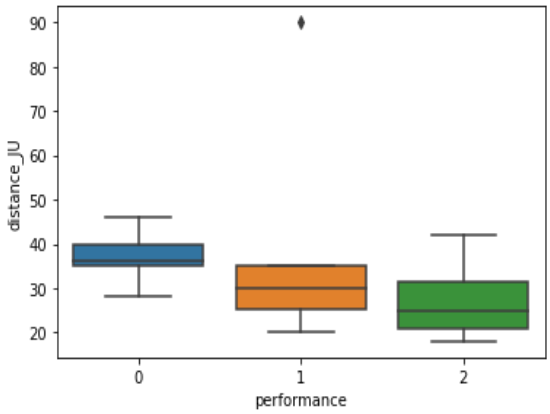
sns.boxplot(data=df, x='exam_score', y='distance_JU')	sns.boxplot(data=df, x='performance', y='distance_JU')
	

Table 2. Boxplot of exam scores (numeric) and exam performance (categorical) against distance

The first boxplot is not informative enough (in fact, it is not appropriate) as both variables are numeric. But, as because we have added a categorical exam performance variable from the exam scores, the second boxplot gives us excellent information. The poor performers in exam (category 0 on x axis) has a median distance of around 36 km, the moderate exam performers has lower distance, with a median hovering around 30 km. The best exam performers has a median distance of approximately 25 km. This information validates what we found earlier in scatter plot we presented in figure 6.

So, although not drastic, but the distance variable has a slight negative impact on students' exam performances. More distance causes fatigue to students, which understandably hampers their concentration level in classes, which in turn results as adverse impact on their exams.

Daily study hours versus exam performance

The daily study in hours is another variable of interest to assess student performance. It has a distribution like below, with mean 1.21, and standard deviation of 0.92.

```
df.daily_study.describe()
```

```
count    20.000000
mean      1.211000
std       0.923659
min       0.010000
25%       0.500000
50%       1.000000
75%       2.000000
max       3.000000
Name: daily_study, dtype: float64
```

Figure 8. Numerical summary of daily study hours variable

We can draw a scatter plot of this variable against the exam scores with the code snippet below.

```
plt.scatter(df.exam_score,df.daily_study)
plt.xlabel('exam scores')
plt.ylabel('daily_study (hours)')
plt.show()
```

Code Segment 5. Scatterplot of daily study hours variable

Interestingly, the plot reveals no particular pattern or trend. Although we expect to have a better exam performance for higher study hours, in this case, we find the almost negligible correlation. This might be because of the fact that the sample size is not large enough, or one exam is not enough to draw a conclusion that these two are not related. But the scatter plot for this sample certainly does not show anything notable. Below is what we get.

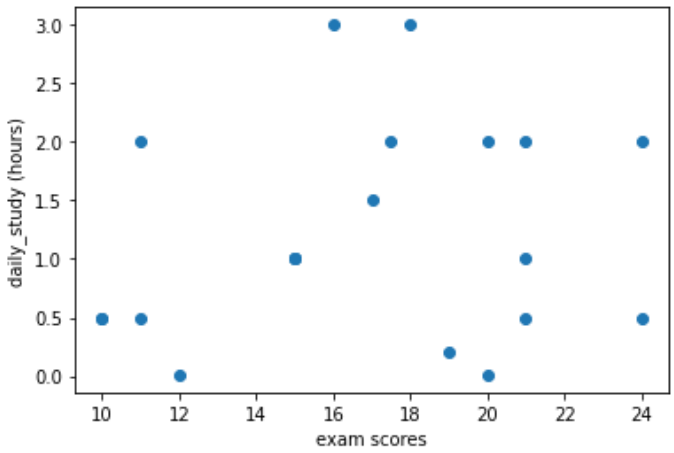


Figure 9. Scatterplot of daily study hours variable

We saw that a boxplot for numeric exam score against another numeric variable is not interpretable to get some valuable insight. So, we will utilize the categorical performance against the study hours again. Below I what we find.

```
sns.boxplot(data=df, x='performance', y='daily_study')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x29b67153d90>
```

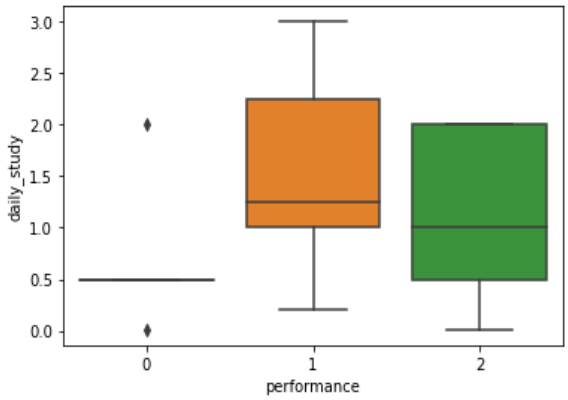


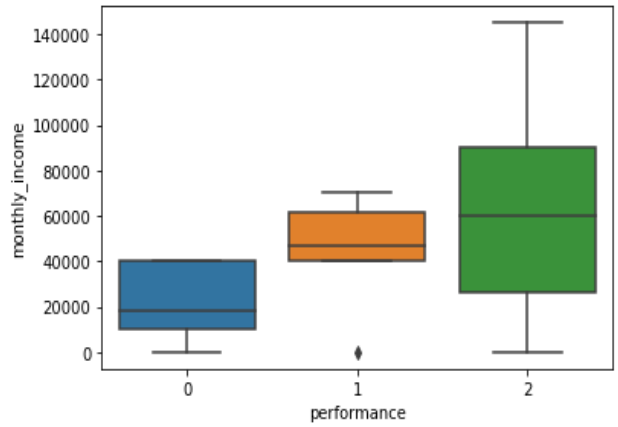
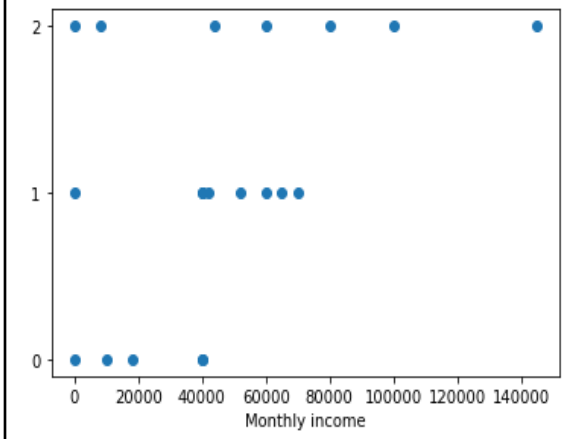
Figure 10. Boxplot of study hours against exam performance categories

This again shows that there is no trend or vivid relation between the two variables. Students with moderate performance studied the most, with a median of almost 1.4 hours of study daily. High-performing students have a median daily study hours around 1 hour.

So, we get no clear relation between these two variables.

Monthly income versus exam performance

Monthly income may indicate stronger socio-economic status, which in turn often facilitates a student's access of better education resources.

sns.boxplot(data=df, x='performance', y='monthly_income')	plt.scatter(df.monthly_income,df.performance) plt.xlabel('Monthly income') plt.yticks(np.arange(min(df.performance), max(df.performance)+1,1)) plt.show()
	

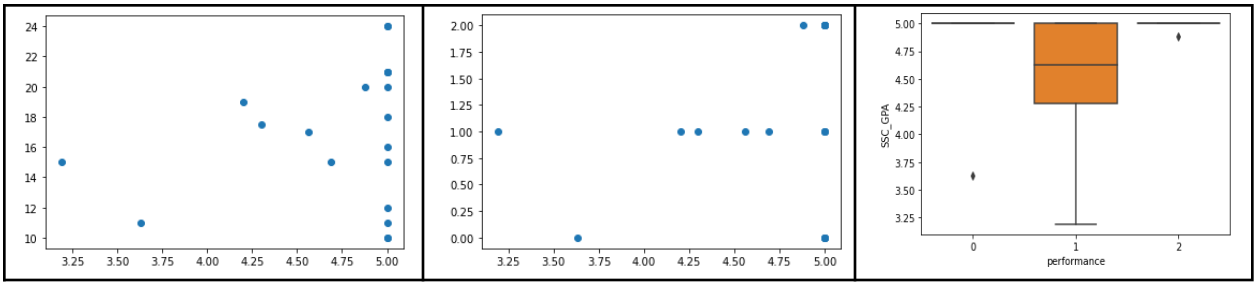
Here we can see the though there are a few exceptions, in general, the high-income students have done better performance in exams. The boxplots shows the high-performer students' median value to be the highest on income scale compared to the other two group medians. But scatterplot also shows that a couple of high-performers also have zero or near-zero monthly income.

Overall, we get a trend that higher income students tend to perform better. This might be due to the fact that either they have better access to quality education materials, or their experience is more than others, which gives them more sense of responsibilities or sense of self-prestige – all of these may give them a boost to perform well in tests.

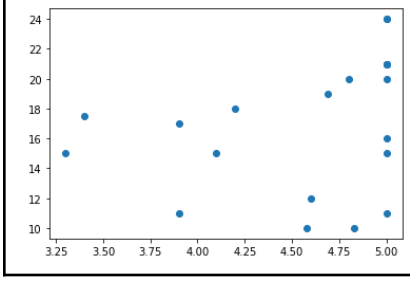
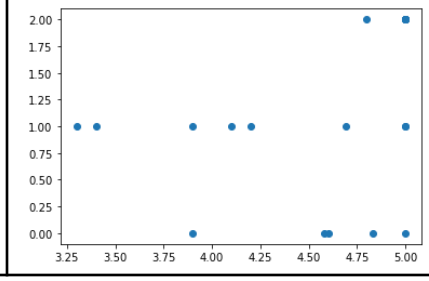
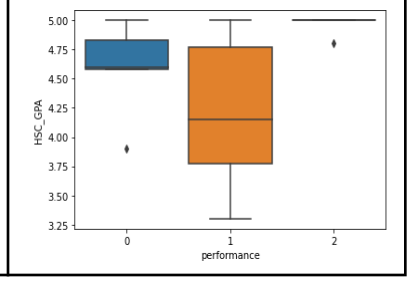
SSC, HSC and Undergraduate results versus exam performance

Past results give us a generic notion on how smart a student is. The table below shows us univariate analysis of SSC, HSC, and Undergraduate results against numeric exam score, and categorical exam performances, respectively.

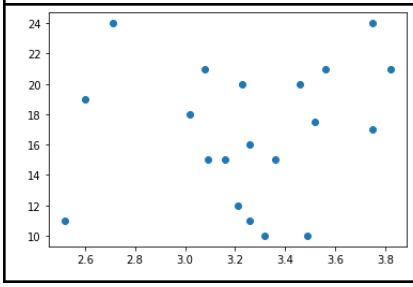
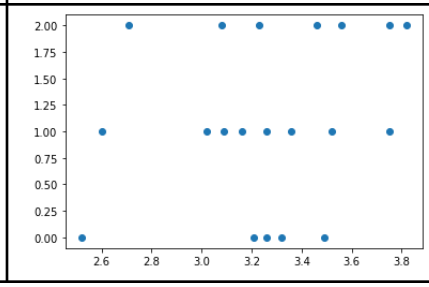
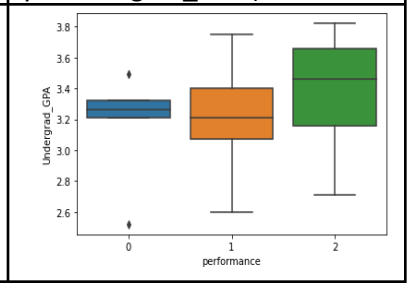
plt.scatter(df.SSC_GPA,df.exam_score)	plt.scatter(df.SSC_GPA,df.performance)	sns.boxplot(data=df, x='performance', y='SSC_GPA')



SSC GPA do not show any trendy relationship with exam performance. The leftmost scatter plot shows that GPA 5 students have all ranges of exam performance. The box plot too shows that higher GPA cannot ensure an excellent exam performance in WM-ASDS program.

plt.scatter(df.HSC_GPA,df.exam_score)	plt.scatter(df.HSC_GPA,df.performance)	sns.boxplot(data=df, x='performance', y='HSC_GPA')
		

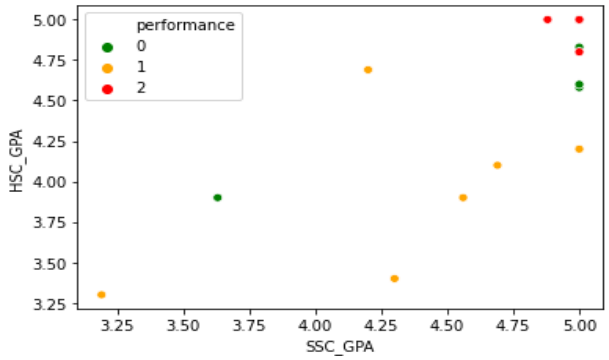
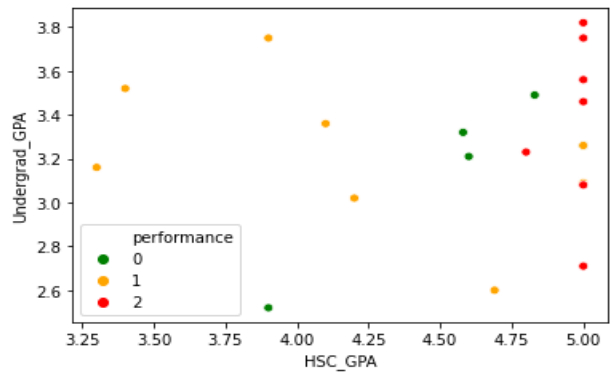
Leftmost scatter plot reveals that students with high HSC GPA (4.5 or more) have given a mixed performance in exam. The top few however are all with a GPA of 5. As we see that lower GPAs, they have achieved moderate to poor results. The box plot too displays this performance-wise distribution. Moderate performers come from highest to lowest GPA holders in HSC. The low and the high performers however are from relatively higher GPAs, with high performers having higher median GPA than that of the low performers. This data shows a very slight downward trend. The lower the HSC result is, the poorer the exam result tend to be.

plt.scatter(df. Undergrad_GPA,df.exam_score)	plt.scatter(df. Undergrad_GPA,df.performance)	sns.boxplot(data=df, x='performance', y='Undergrad_GPA')
		

Undergraduate GPA too shows similar distribution pattern like HSC GPAs. The moderate exam results come from a wide range of GPA holders. The low and good results are from mostly GPA 3+ students, with high performers having higher median GPA.

Bivariate Analysis of past results with exam performance

Many a times, we see that a bivariate analysis may reveal some notable trend. In the following table, we will analyze 2 past results simultaneously with ASDS exam performance. For the ASDS exam, we use categorical performance, and we represent this with colors. For 3 levels, we use 3 colors, which we will add as a legend in the plots.

SSC and HSC GPA together against exam performance	
sns.scatterplot(data=df, x="SSC_GPA", y="HSC_GPA", hue="performance", hue_order=perf_order, palette=p_colors) plt.show()	
HSC and Undergraduate GPA together against exam performance	
sns.scatterplot(data=df, x="HSC_GPA", y="Undergrad_GPA", hue="performance", hue_order=perf_order, palette=p_colors) plt.show()	
SSC and Undergraduate GPA together against exam performance	

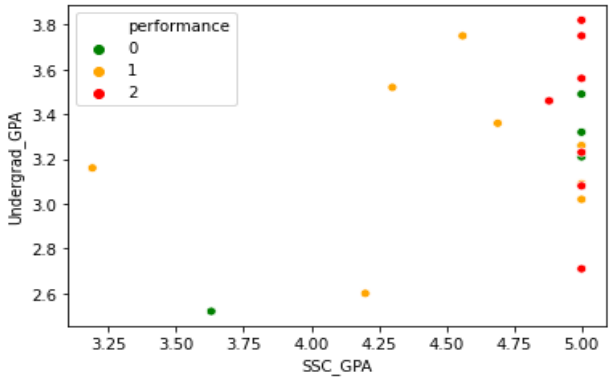
sns.scatterplot(data=df, x="SSC_GPA", y="Undergrad_GPA", hue="performance", hue_order=perf_order, palette=p_colors) plt.show()	

Table 3. Bivariate analysis of past results with ASDS exam performance

Students with high SSC and HSC GA holders show a mixed result, both high and poor exam performance in exams of ASDS.

Students with high HSC GPA has a high performance in ASDS exam although they have a mixed undergraduate GPAs. The mediocre students have a moderate performance.

For high undergraduate and high SSC GPAs, we again have a mixed results in ASDS exam. The plot shows no trend or pattern.

It deems that the bivariate analyses fails to establish any clear relation between past results and ASDS exam performance. However, it was necessary. The inability to find any insight might be due to the fact that the sample size is not large enough, or one exam result may not be a good indicator for establishing any such relations.

A pairplot shows all such relationships among all combination of variables. The color is for categorical exam performances, while x and y axes are combinations of all variables.

```
sns.pairplot(df, hue="performance")

plt.show()
```

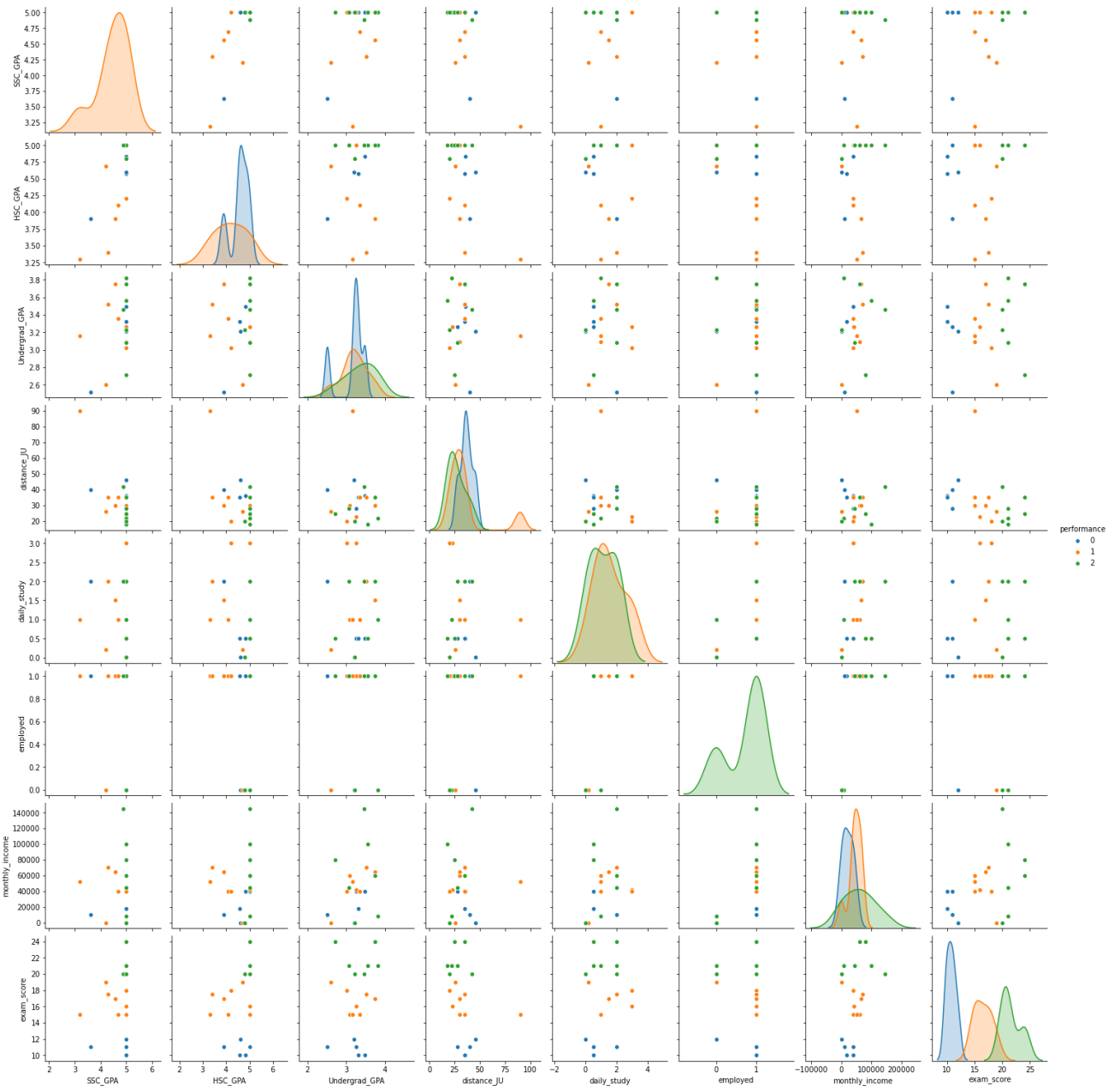


Figure 11. Pair plot to show all distributions

A careful observation usually reveals interpretable correlations, if any, among the variables. In our case, we get the density curves per performance category, and for each variable. But a clear segregation among the performance categories based on variables is not so vivid.

Conclusion

The following are our summary of findings from the analysis.

- Students with no job has a higher median value in exams, but some students with job have shown outstanding performance as well.
- The lower the distance that the students have to travel to get in to the university, the better their exam performances are. In other words, higher distance has slightly adverse impact on exam performance.
- Daily study hours do not depict any influence over exam performance. This is unexpected. This might well be because of the fact that the sample size is not large enough, or one exam performance may not be enough to draw a conclusion of non-relation between these two variables.
- Students with higher income shows a tendency to do better in exams. However, some low or no-income students also performed well.
- Students with higher HSC and undergraduate GPAs tend to perform well in exams. SSC GPA do not show any correlation.