Section= SE(8A)

ma’am nasreen akhtar

Data Mining Project

Part 1: Eda on given dataset

|  |  |
| --- | --- |
| **Group Member Roll Number** | **Group Member Name** |
| 19F-0916 | Muhammad Abdullah |
| 19F-0917 | Zainab Ijaz |
| 19F-0965 | Mahnoor |
| 19F-0908 | Maria Fatima |

Table of Contents

[**Introduction:** 3](#_Toc133780662)

[**Problem:** 3](#_Toc133780663)

[**Summary of Dataset:** 3](#_Toc133780664)

[**Tools Used:** 3](#_Toc133780665)

[**Working Using Python:** 3](#_Toc133780666)

[Importing File while considering useful attributes only: 3](#_Toc133780667)

[Data Duplication: 4](#_Toc133780668)

[Null Values: 4](#_Toc133780669)

[Displaying Null Values using Heatmap: 4](#_Toc133780670)

[Missing Values in Sheet D1: 4](#_Toc133780671)

[Missing Values in Sheet D2: 5](#_Toc133780672)

[Missing Values in Sheet D3: 5](#_Toc133780673)

[Missing Values in Sheet D4: 6](#_Toc133780674)

[Missing Values in Sheet D5: 6](#_Toc133780675)

[Missing Values in Sheet D6: 7](#_Toc133780676)

[Missing Values in Sheet D7: 7](#_Toc133780677)

[Handling Null Values: 8](#_Toc133780678)

[Data Reduction: 8](#_Toc133780679)

[Correlation among attributes: 8](#_Toc133780680)

[Correlation in Sheet D1: 8](#_Toc133780681)

[Correlation in Sheet D2: 9](#_Toc133780682)

[Correlation in Sheet D3: 9](#_Toc133780683)

[Correlation in Sheet D4: 10](#_Toc133780684)

[Correlation in Sheet D5: 10](#_Toc133780685)

[Correlation in Sheet D6: 11](#_Toc133780686)

[Correlation in Sheet D7: 11](#_Toc133780687)

[Relation of attributes with main class: 12](#_Toc133780688)

[Grade vs Assignments: 12](#_Toc133780689)

[Grade vs Quizzes: 13](#_Toc133780690)

[Grade vs Sessional I: 13](#_Toc133780691)

[Grade vs Sessional II: 14](#_Toc133780692)

[Performing EDA: 14](#_Toc133780693)

[Sheet D1 EDA Assignment 1: 14](#_Toc133780694)

[Sheet D1 EDA Assignment 2: 15](#_Toc133780695)

[Sheet D1 EDA Assignment 3: 15](#_Toc133780696)

[Sheet D1 EDA Assignment 4: 15](#_Toc133780697)

[Sheet D1 EDA Assignment 5: 16](#_Toc133780698)

[Sheet D1 EDA All Assignments: 16](#_Toc133780699)

[Sheet D1 EDA Quiz 1: 16](#_Toc133780700)

[Sheet D1 EDA Quiz 2: 17](#_Toc133780701)

[Sheet D1 EDA Quiz 3: 17](#_Toc133780702)

[Sheet D1 EDA Quiz 4: 17](#_Toc133780703)

[Sheet D1 EDA Quiz 5: 18](#_Toc133780704)

[Sheet D1 EDA Quiz 6: 18](#_Toc133780705)

[Sheet D1 EDA Quiz 7: 18](#_Toc133780706)

[Sheet D1 EDA All Quizzes: 19](#_Toc133780707)

[Sheet D1 EDA Sessional I: 19](#_Toc133780708)

[Sheet D1 EDA Sessional II: 19](#_Toc133780709)

[Sheet D1 EDA Results: 20](#_Toc133780710)

[**Python Code:** 21](#_Toc133780711)

[**Working Using Orange:** 26](#_Toc133780712)

[Workflow Demonstration: 26](#_Toc133780713)

[Data Importing: 26](#_Toc133780714)

[Covert data into Table from File: 28](#_Toc133780715)

[Heat Map: 29](#_Toc133780716)

[Co-relation: 29](#_Toc133780717)

[Scatter plot: 30](#_Toc133780718)

[Box Plot: 31](#_Toc133780719)

[Preprocessing: 31](#_Toc133780720)

[Data After Preprocessing: 32](#_Toc133780721)

[Boxplot After preprocessing: 33](#_Toc133780722)

[Scatterplot After preprocessing: 33](#_Toc133780723)

# **Introduction:**

This document contains the pre-processing phase (part 1) of the Data Mining Project, which contains Python code and data analysis. We’ve applied different data mining techniques to normalize the data and make it ready for phase-2. We have performed EDA on the given dataset and tried to extract important information from it.

# **Problem:**

To predict students’ grades as “pass” or “fail” before: (a) Mid-II, and, (b) Final exams. For Mid-II grade prediction, use the following features: first four assignments, first four quizzes, and Mid-I score; and, for grade prediction before the final exam, use all the features (take the best 5 assignments and quizzes).

# **Summary of Dataset:**

The dataset contains students’ assessment scores including <Assignments, Quizzes, Mid- I, Mid-II>, and a predictor variable <Grade>. The data has been anonymized to hide the identities of the students and course(s). The data is shared on seven sheets **(D1 to D7),** where each sheet contains a different number of assignments and quizzes. However, only the best 5 assignments and quizzes are included for each student before calculating their grades. Also, note that total marks for assignments and quizzes are given on the top along with their corresponding weights.

# **Tools Used:**

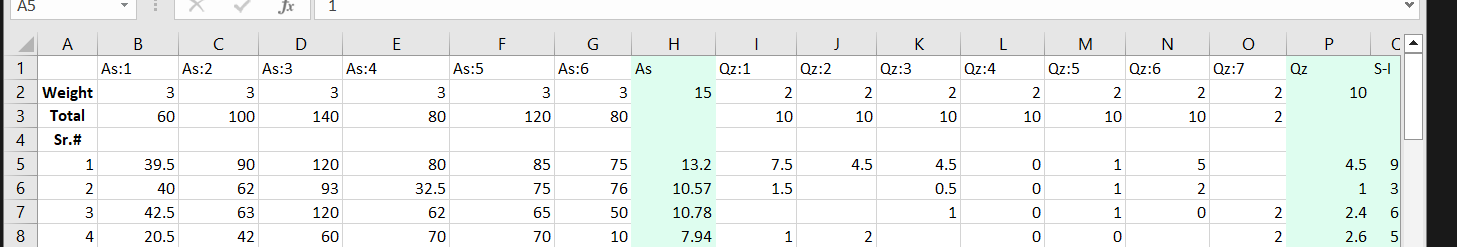
|  |  |
| --- | --- |
| **Tools** | **Purpose** |
| Python | Data normalization using Code. |
| Orange | Data normalization using Software |
| MS Word | For generating report |
|  |  |

# **Working Using Python:**

We’ve tried to perform the following tasks to achieve phase 1:

## Importing File while considering useful attributes only:

The dataset file starts from 1st row, which shows the name of the columns. 2nd row which shows the weightage of tasks. 3rd column which shows the total marks of the tasks. 4th column is null/free. And main data starts from the 5th column. So, due to this, we’ve started taking data while neglecting the 2nd, 3rd, and 4th row which contains nothing which can contribute to our analysis.



## Data Duplication:

We have also performed a data duplication analysis to check the redundant data. Because the dataset is on the student’s record hence, we don’t need to extra check any duplicity. Yet, we need to check for duplicate serial numbers. There were no duplications in the serial numbers and our data was perfectly passed through this phase.

## Null Values:

The given dataset has some missing values, some missing assignments marks, and the same for the quizzes too. There were also some missing values available under the Grade column that also needs to be handled.

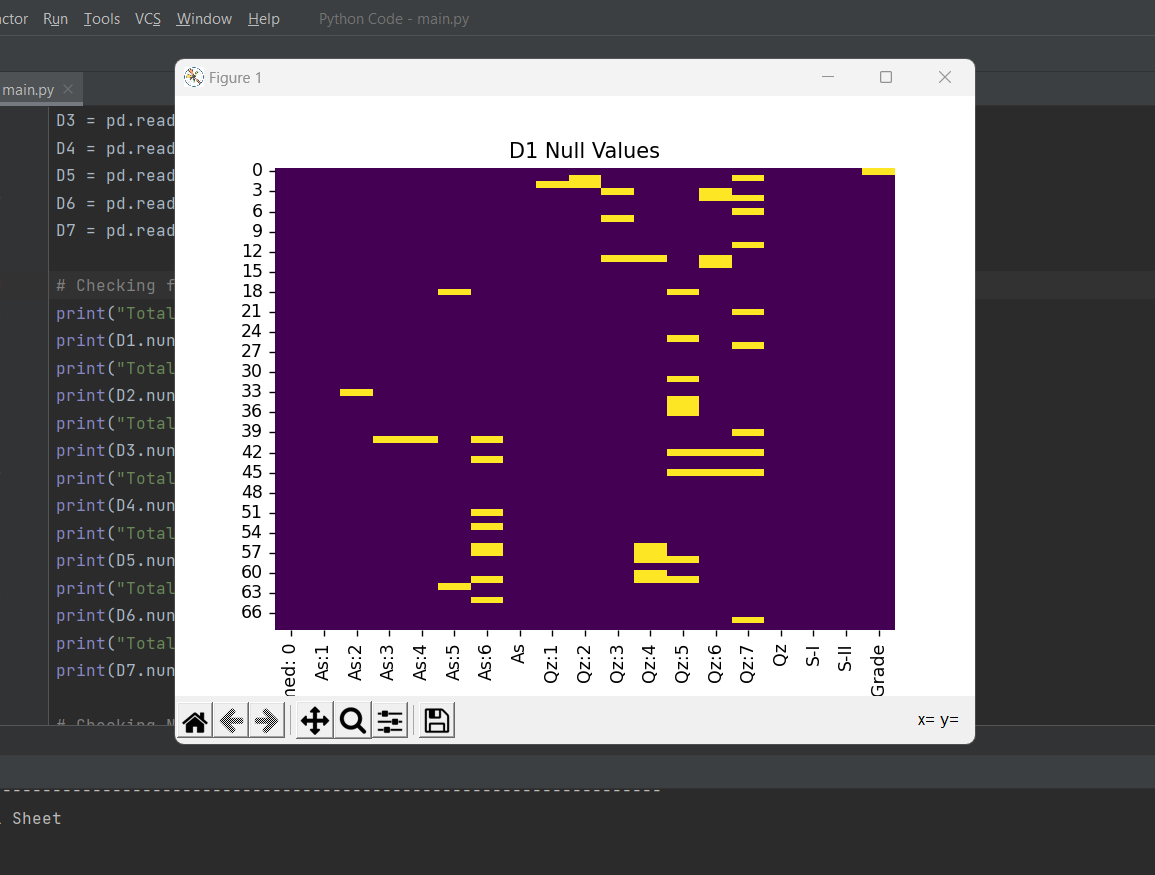
**For missing numerical values,** we’ve opted them to be replaced with the median of that column which also handles the outliers’ issues.

**For missing categorical values,** we’ve opted them to be replaced with the mode of that column.

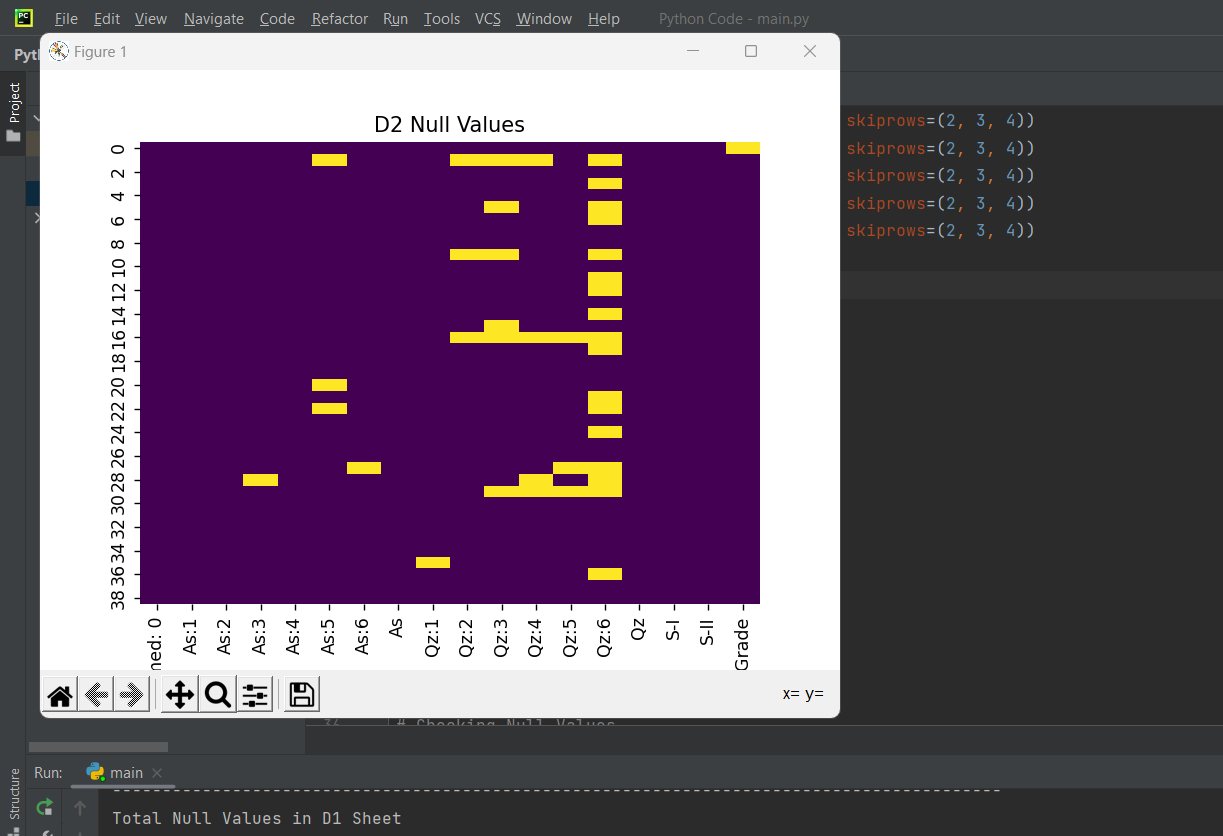
## Displaying Null Values using Heatmap:

We’ve used Heatmap to illustrate the missing values inside the dataset and those missing values are as follows:

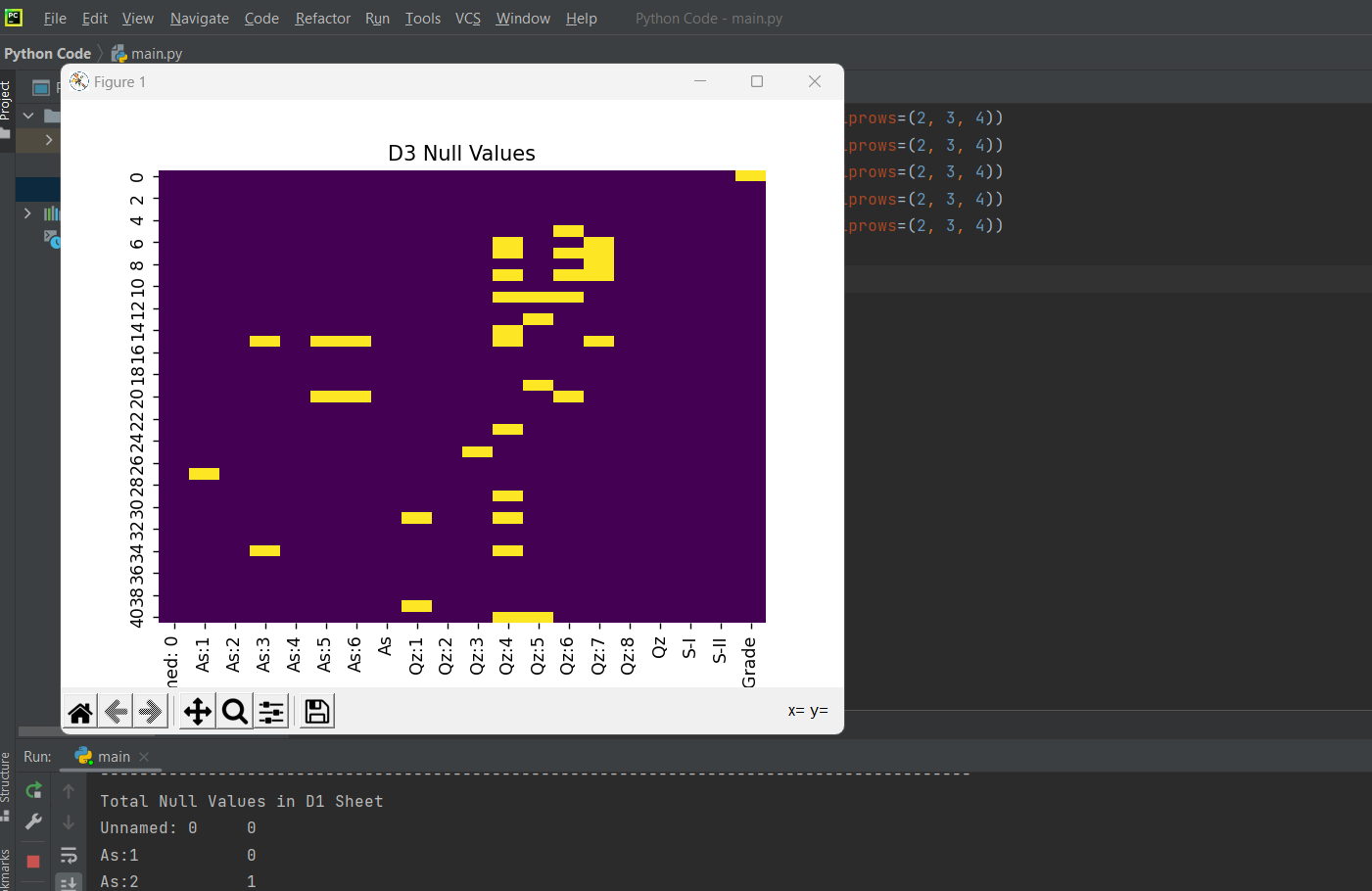
### Missing Values in Sheet D1:



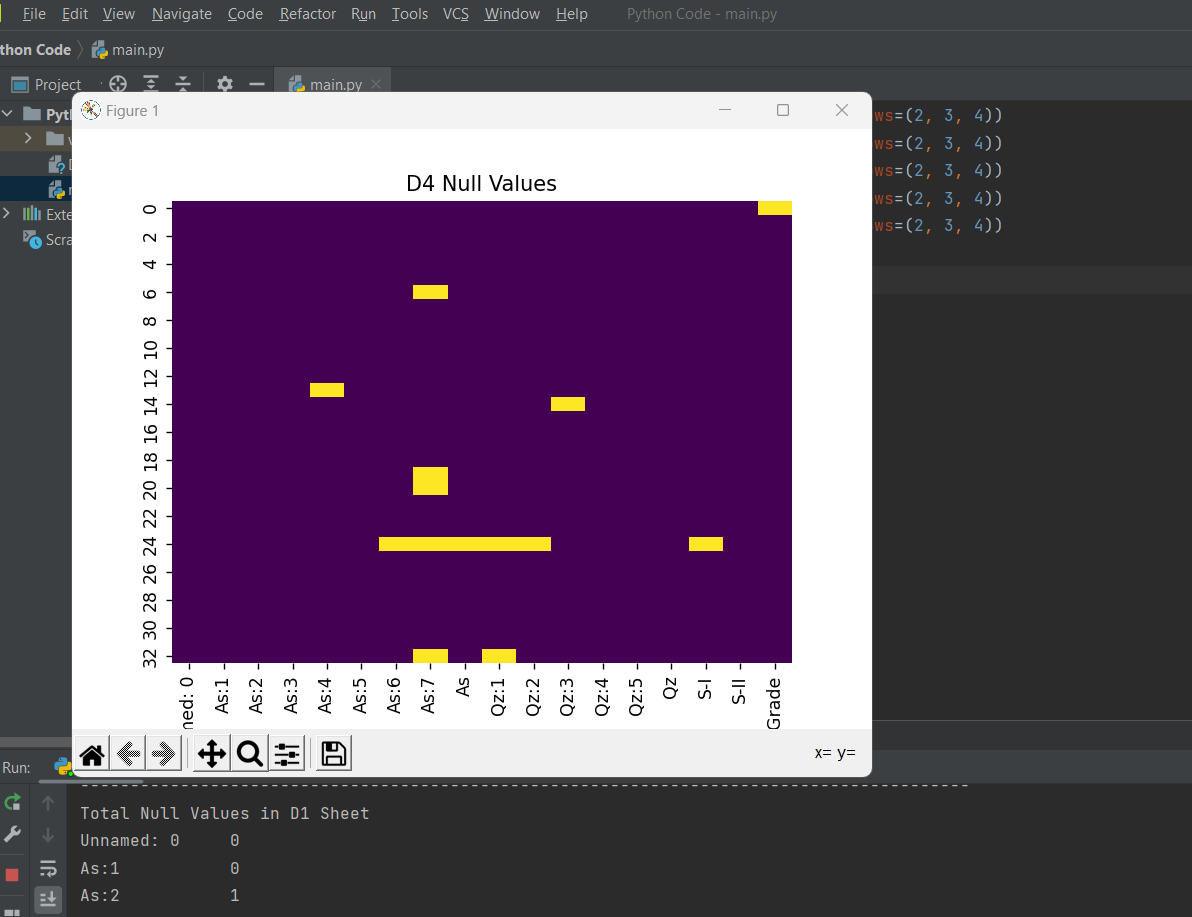
### Missing Values in Sheet D2:



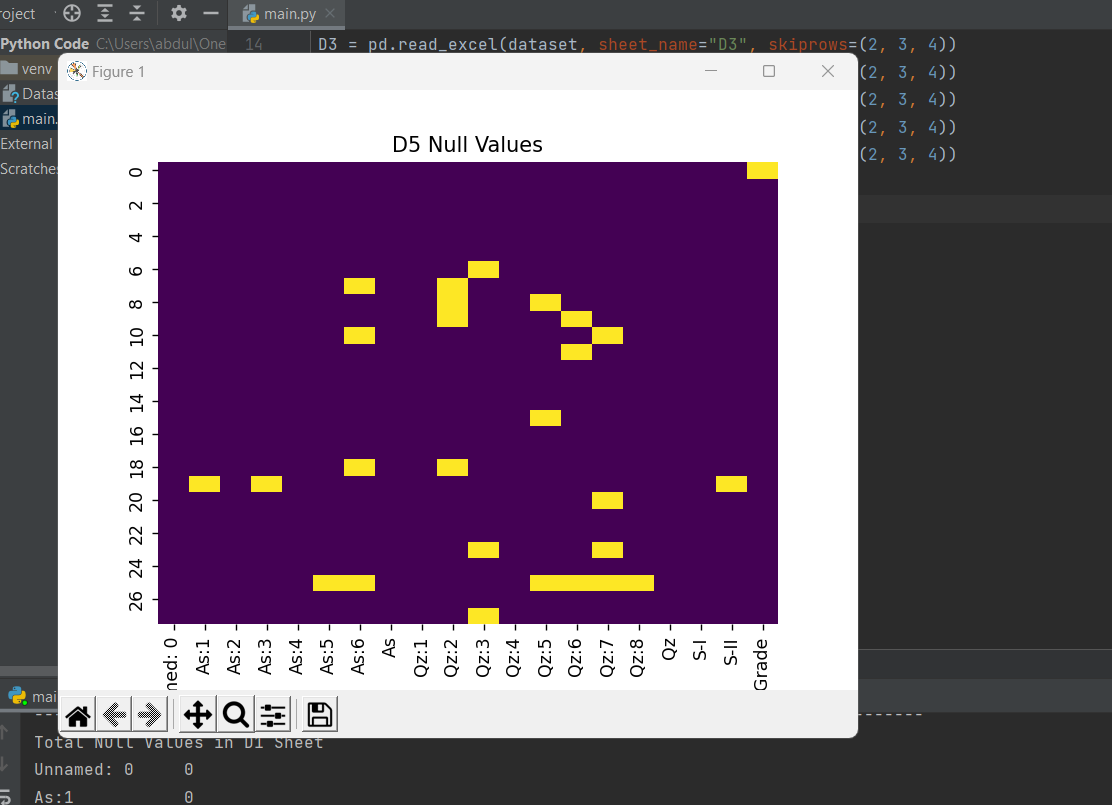
### Missing Values in Sheet D3:



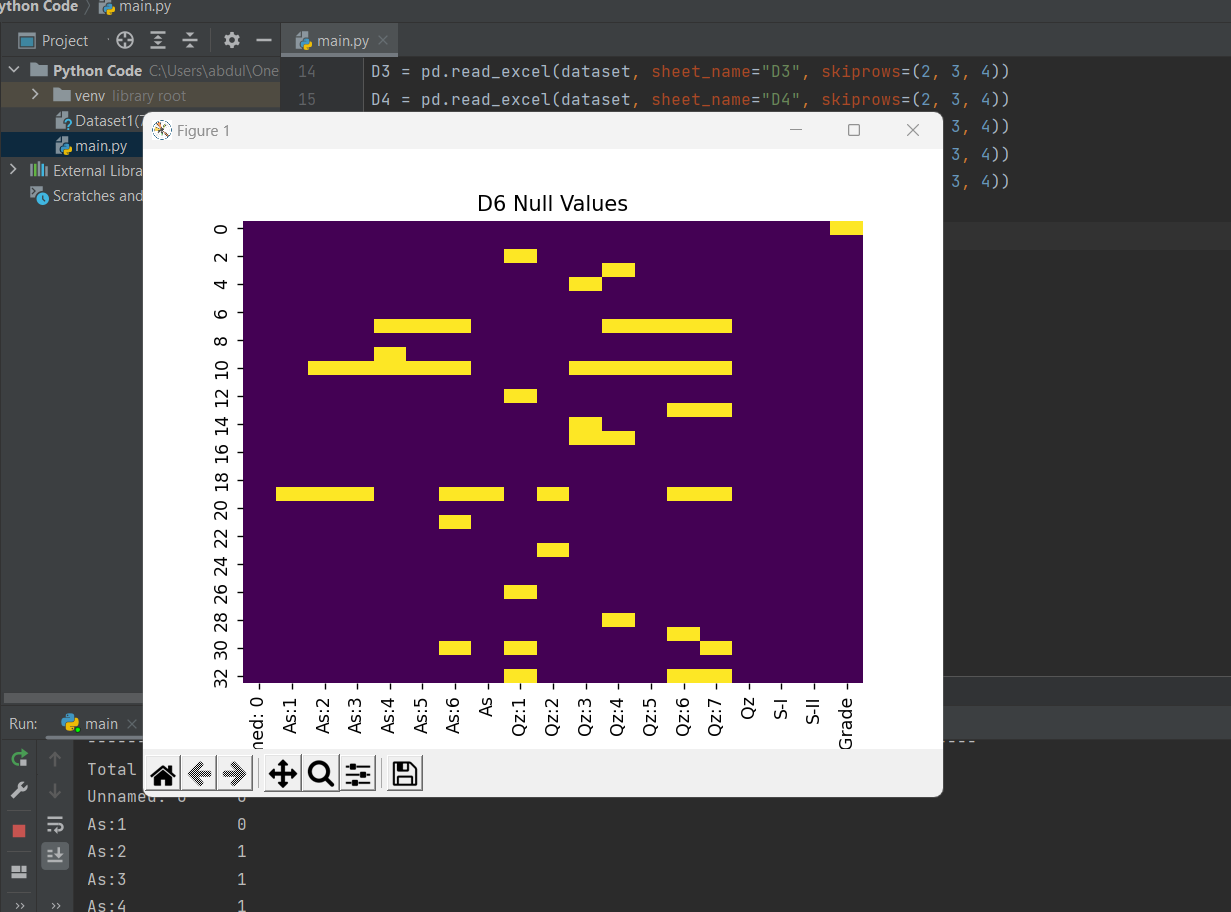
### Missing Values in Sheet D4:



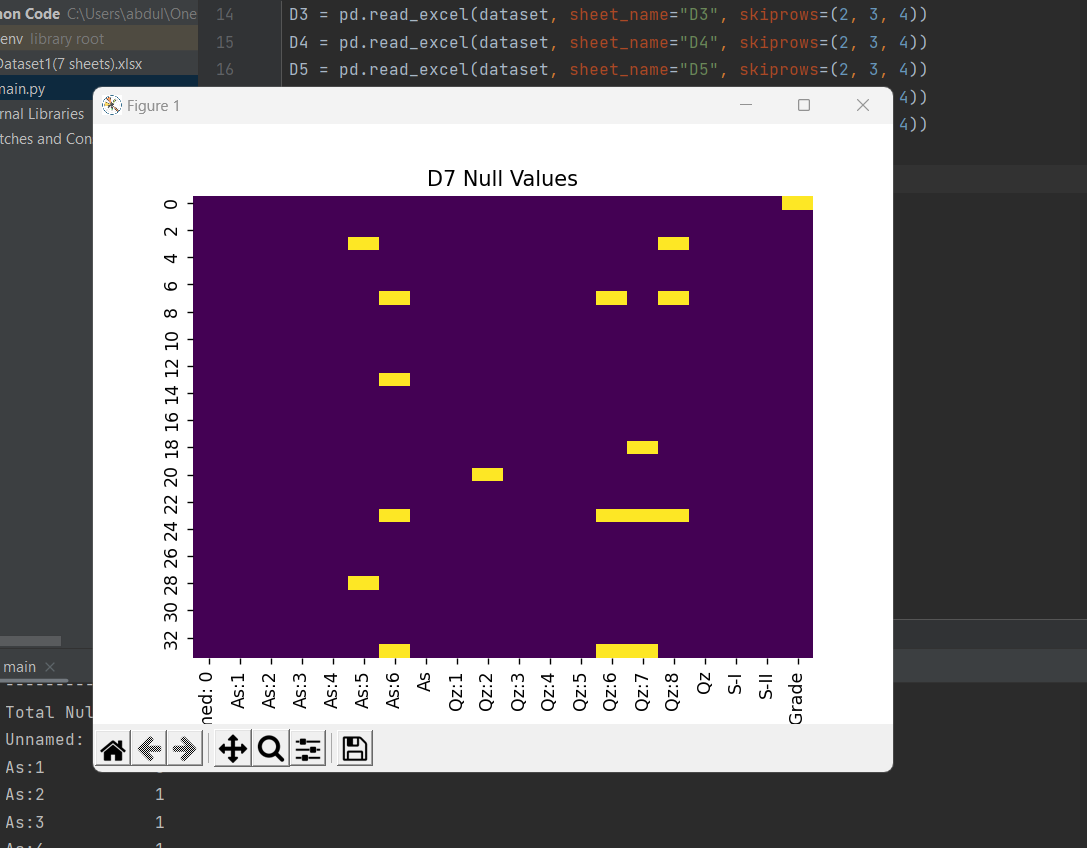
### Missing Values in Sheet D5:



### Missing Values in Sheet D6:



### Missing Values in Sheet D7:



## Handling Null Values:

As mentioned earlier, we’ve handled these null values in the following way:

**For missing numerical values,** we’ve opted them to be replaced with the median of that column which also handles the outliers’ issues.

**For missing categorical values,** we’ve opted them to be replaced with the mode of that column.

## Data Reduction:

In this phase, we performed data reduction on 1st column of the data set as this was the only column that was not playing any role in our dataset. This was just portraying serial numbers of students.

Apart from 1st column, all of the remaining columns were playing important role in the dataset and they were required for further processing of the data.

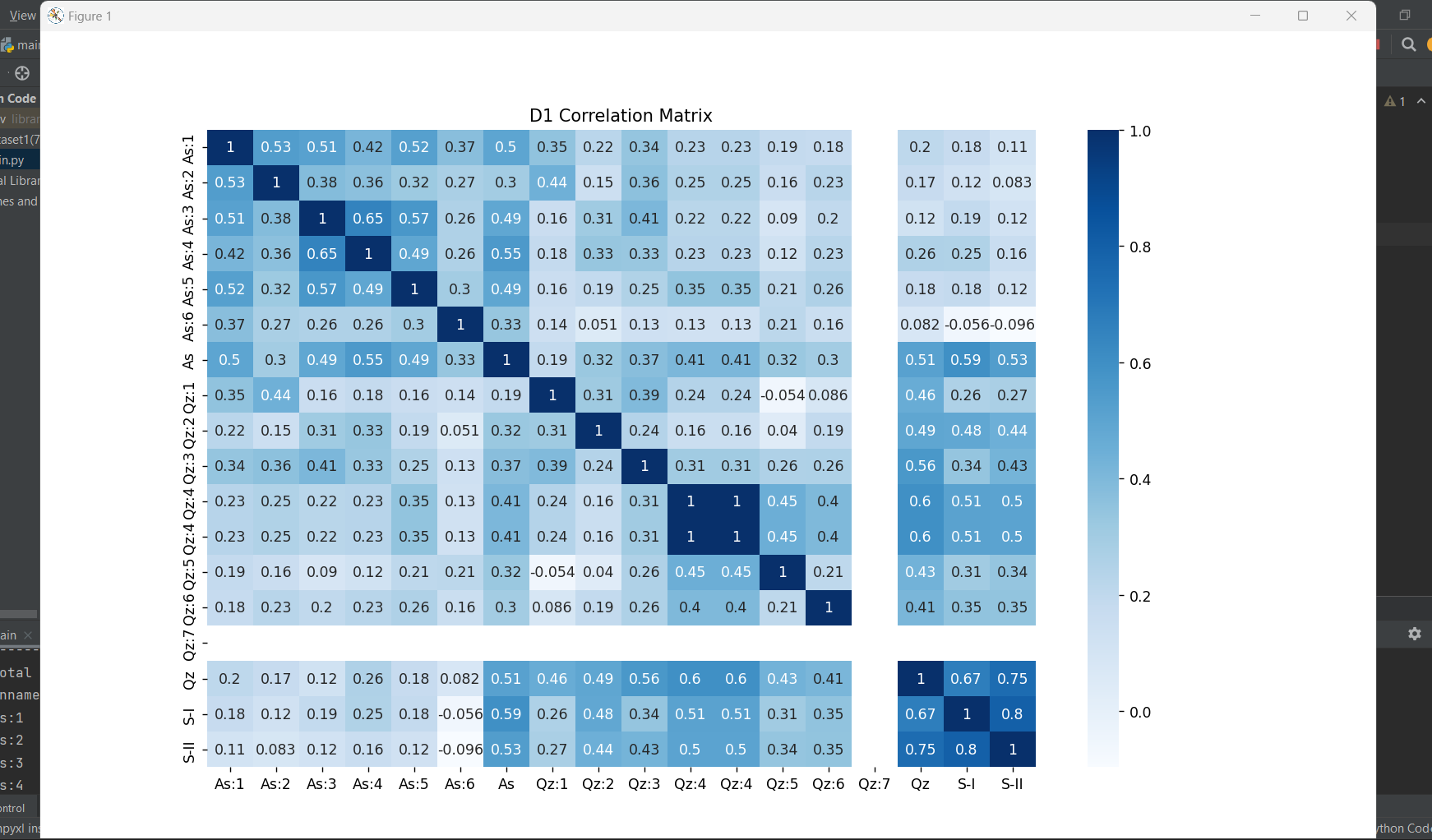
## Correlation among attributes:

After data reduction, our data was ready to step on next level which represents the correlation phase. In this phase, we’ve tried to check out the correlation of attributes (assignments, quizzes, and sessional) to take note of how they are related to each other.

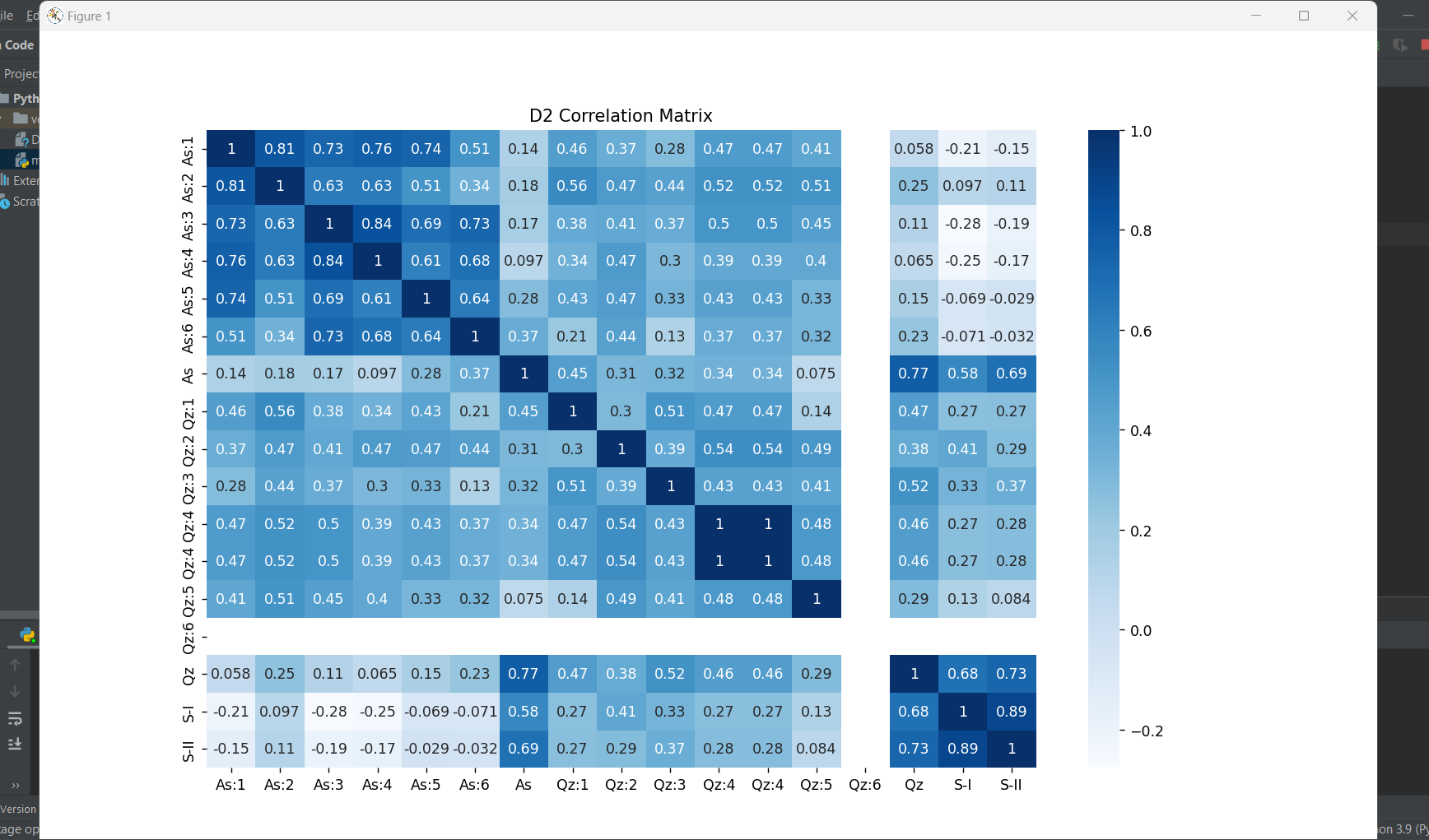
We’ve used the Seaborn library to illustrate the correlation of attributes in all sheets.

Following are the results of the correlation of attributes:

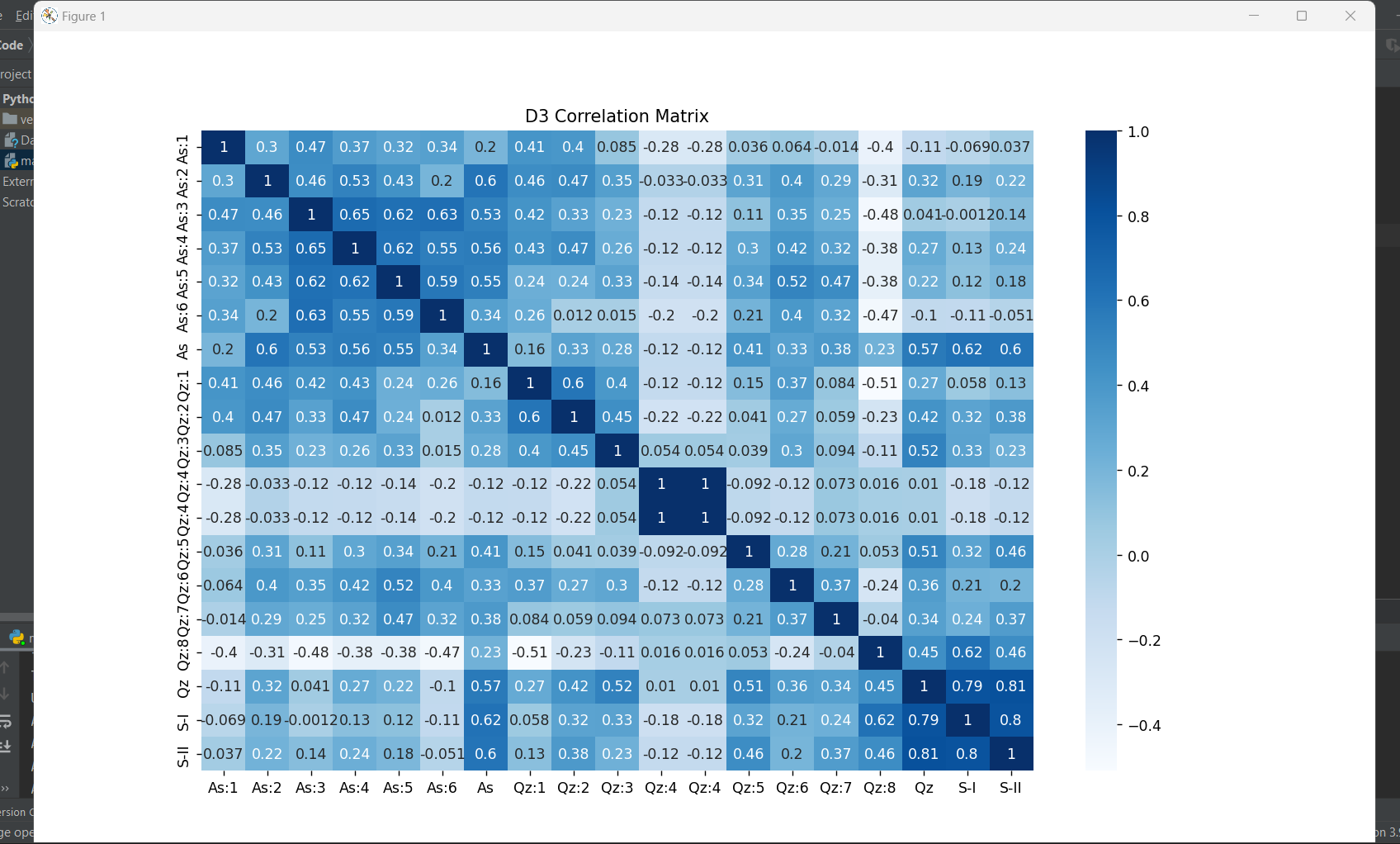
### Correlation in Sheet D1:



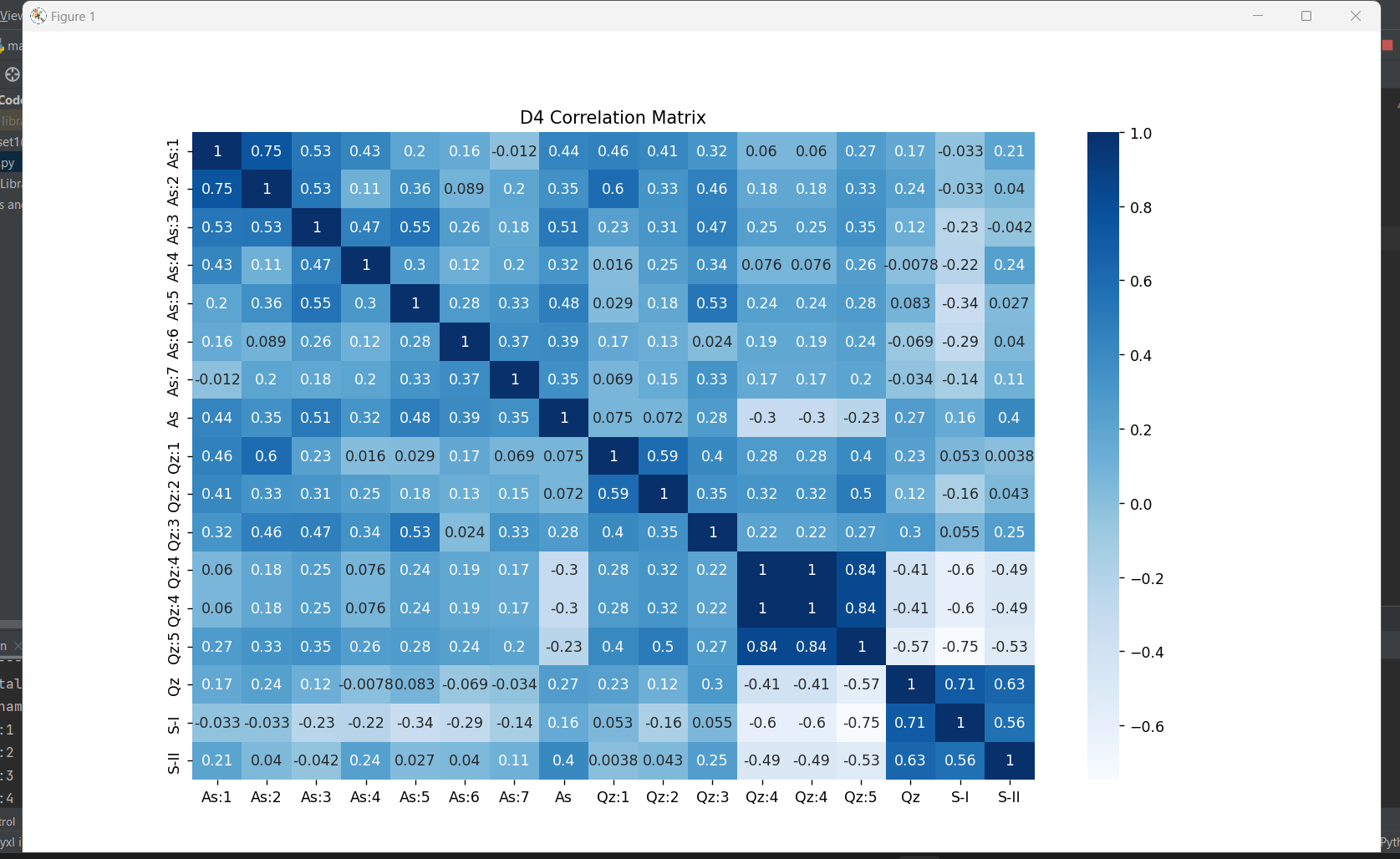
### Correlation in Sheet D2:



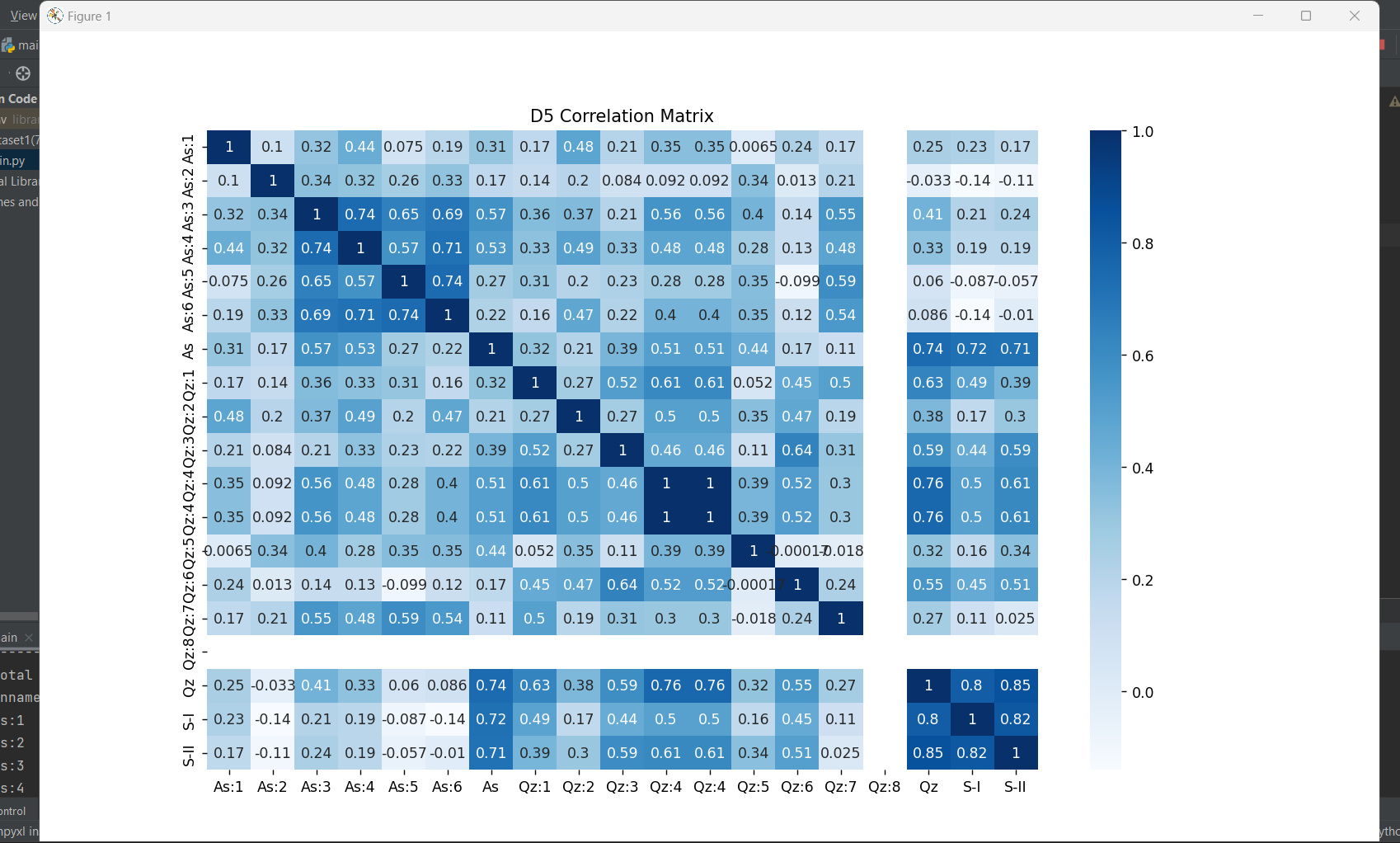
### Correlation in Sheet D3:



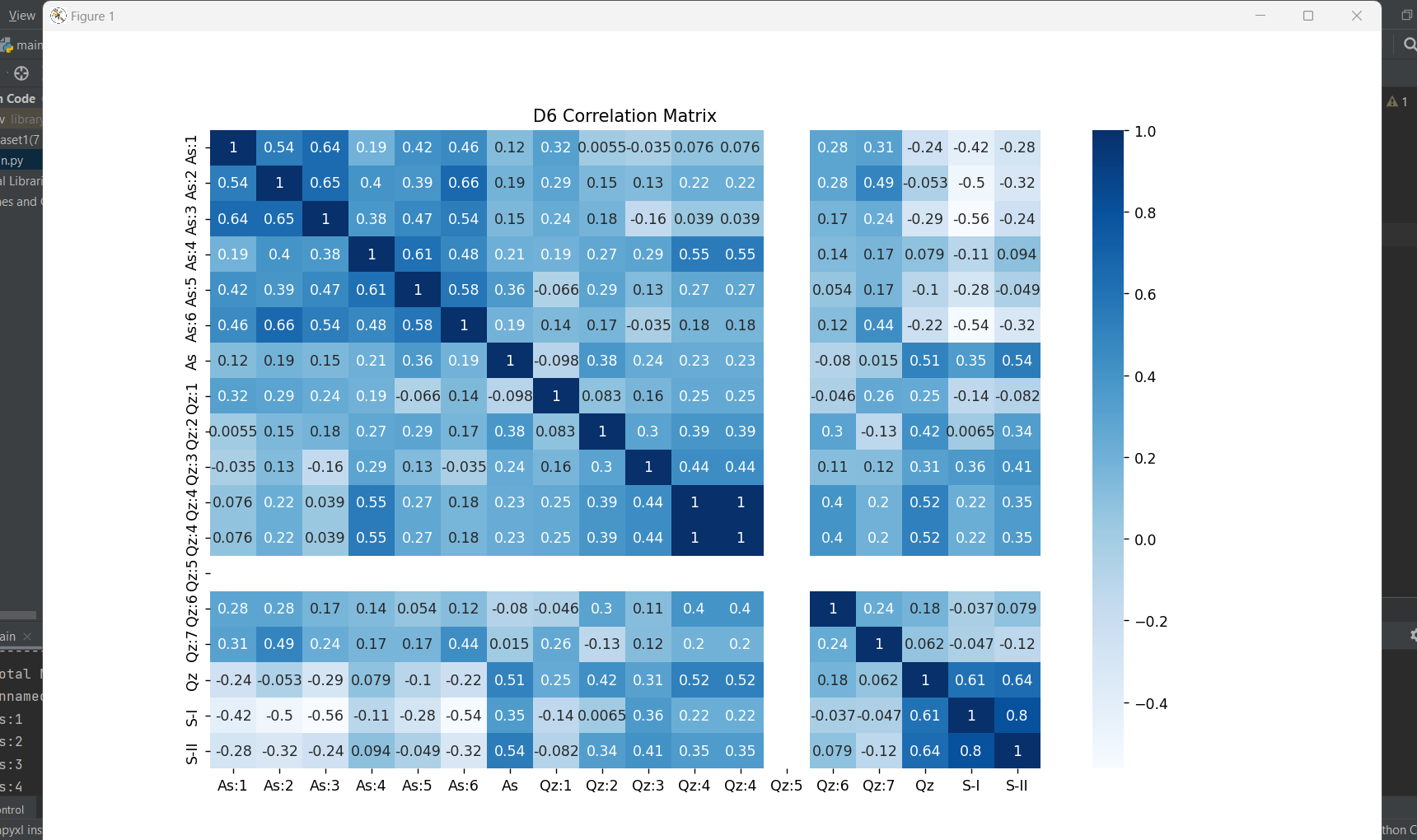
### Correlation in Sheet D4:



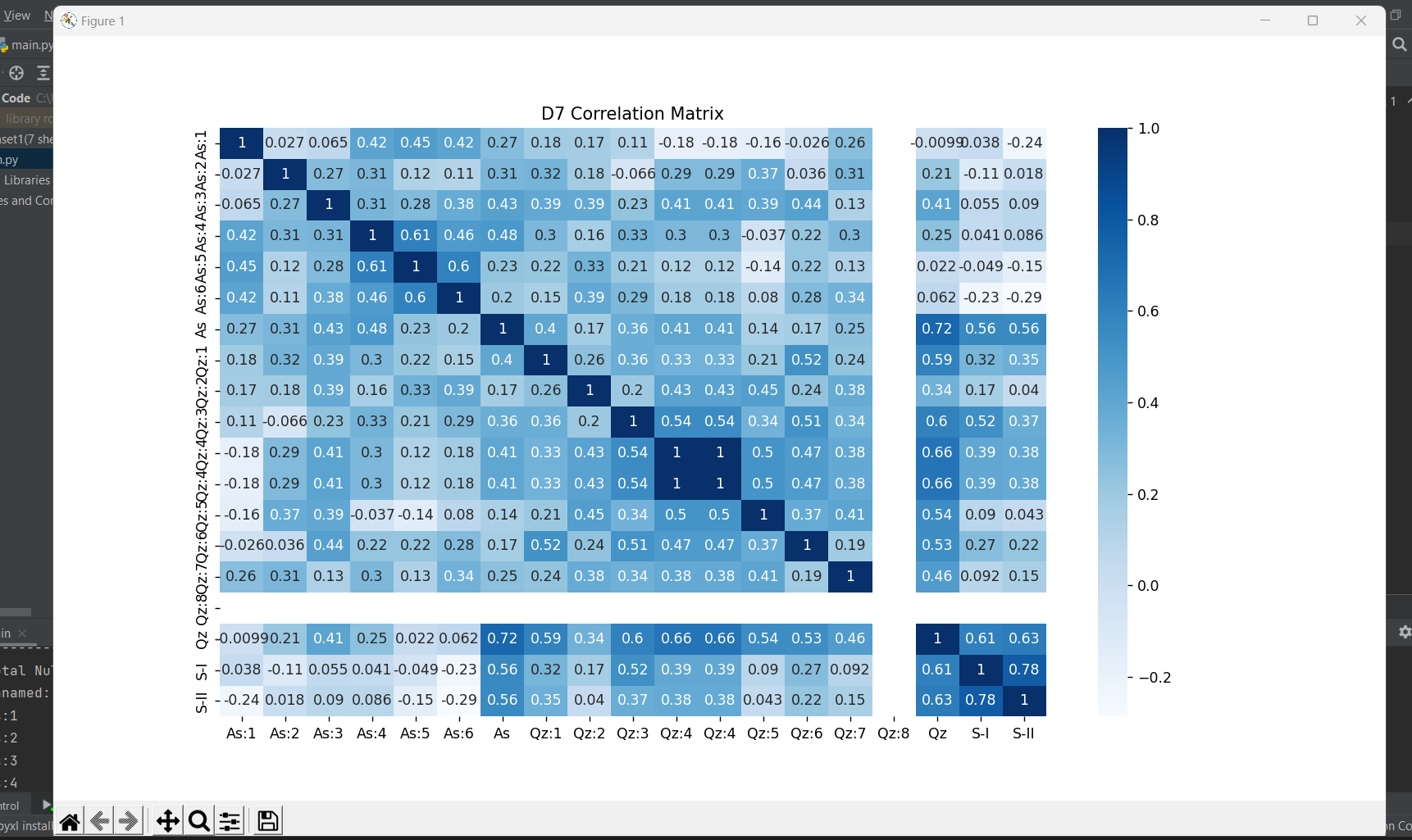
### Correlation in Sheet D5:



### Correlation in Sheet D6:



### Correlation in Sheet D7:

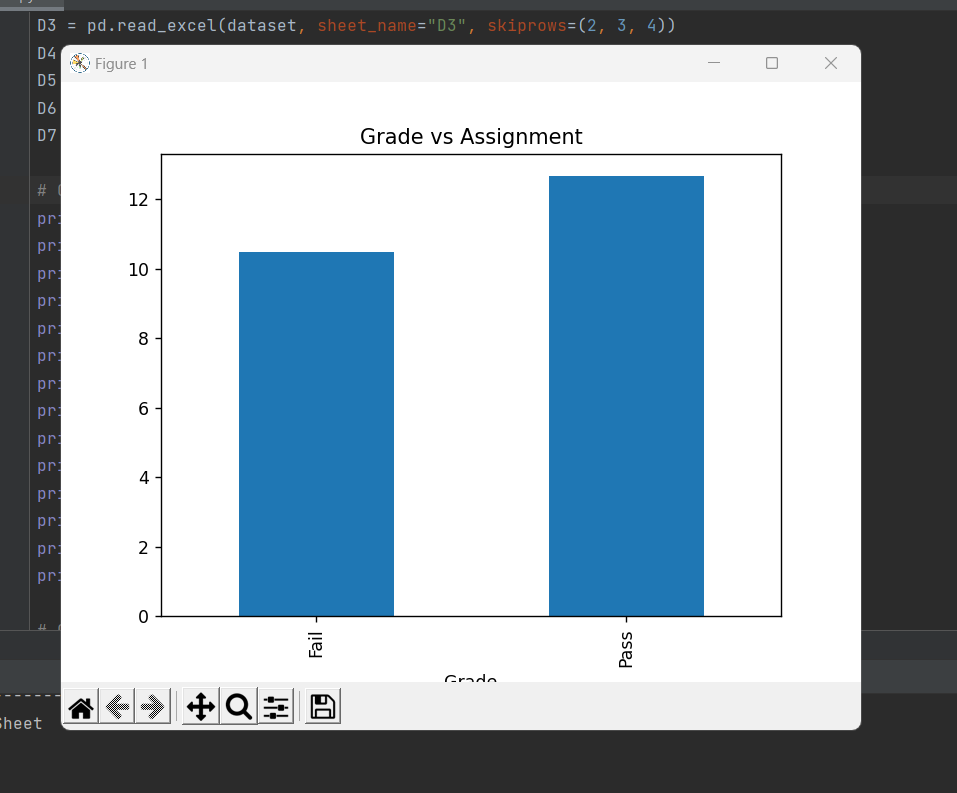


## Relation of attributes with main class:

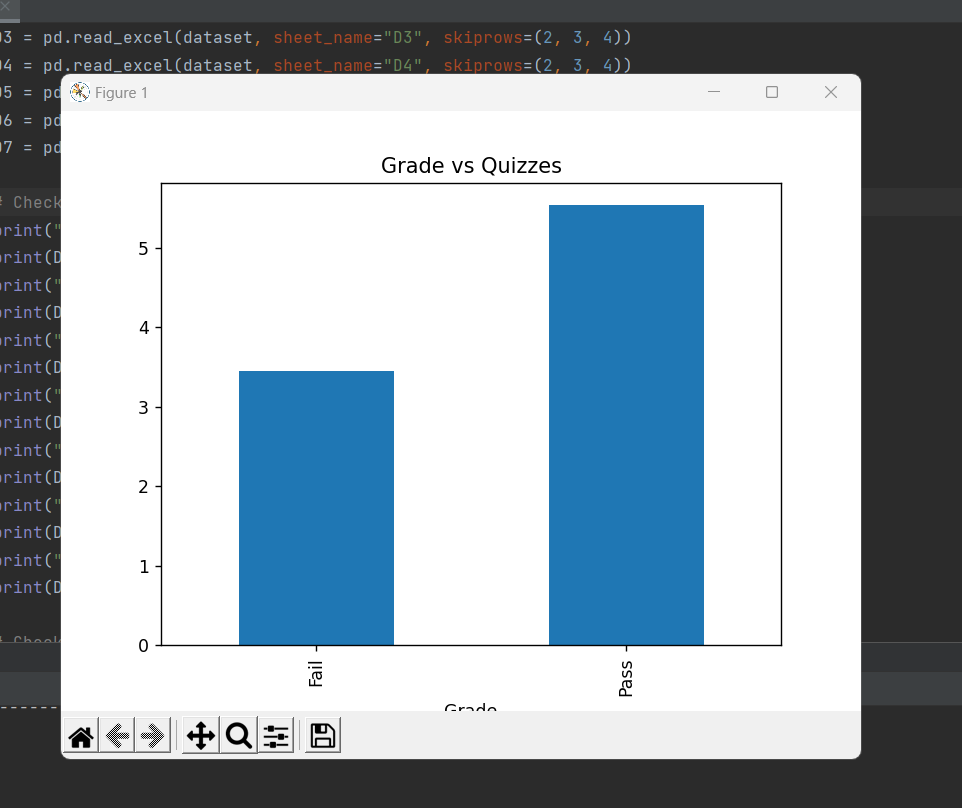
After performing correlation among attributes, now it was time to relate attributes with the main class. As our dataset is on the student’s record, our main class shows binary values, Fail and Pass. We’ve tried to map our Grade column with assignments, quizzes, and sessional exams to have an abstract view of their relation.

The following images show the relation of Grade with the remaining attributes:

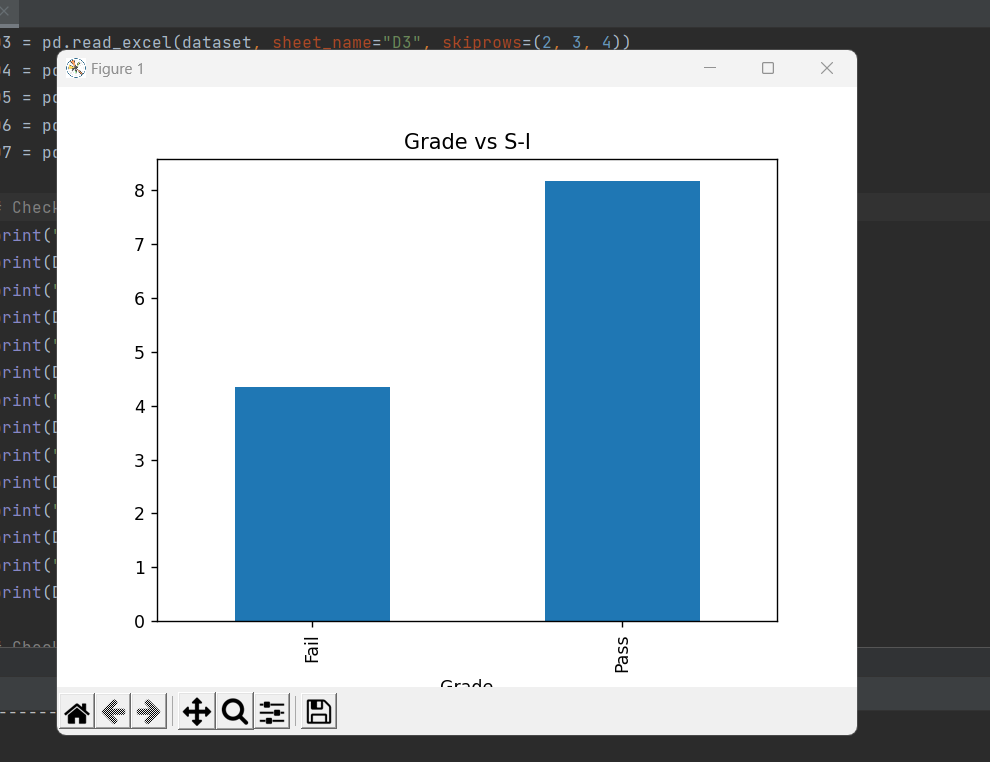
### Grade vs Assignments:



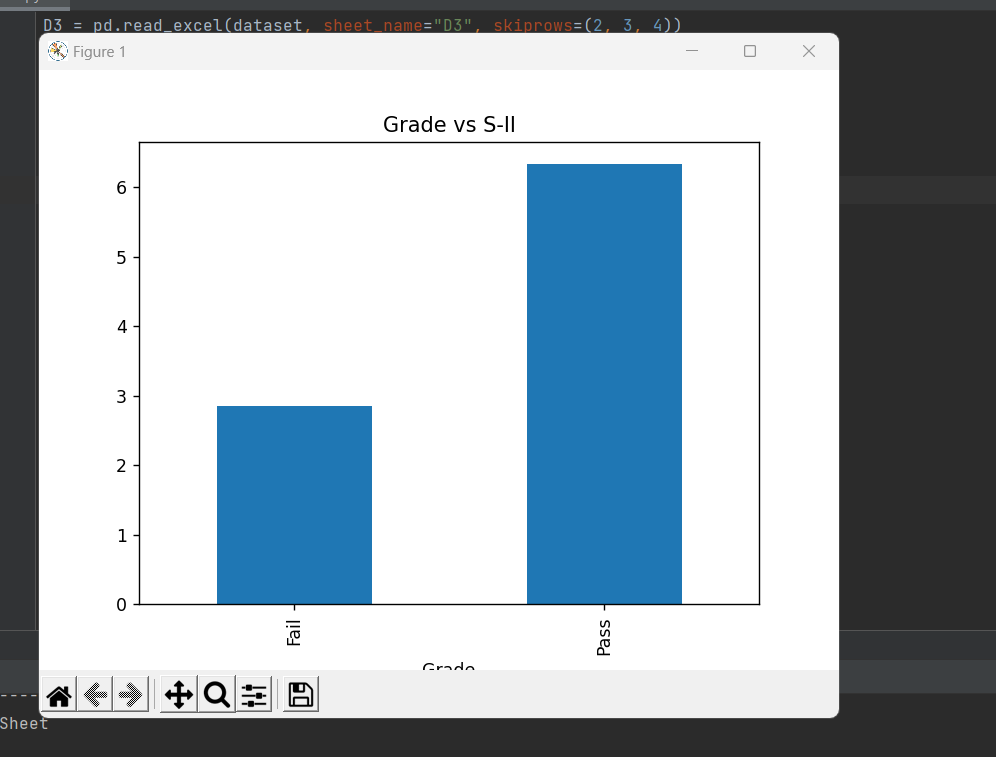
### Grade vs Quizzes:



### Grade vs Sessional I:



### Grade vs Sessional II:

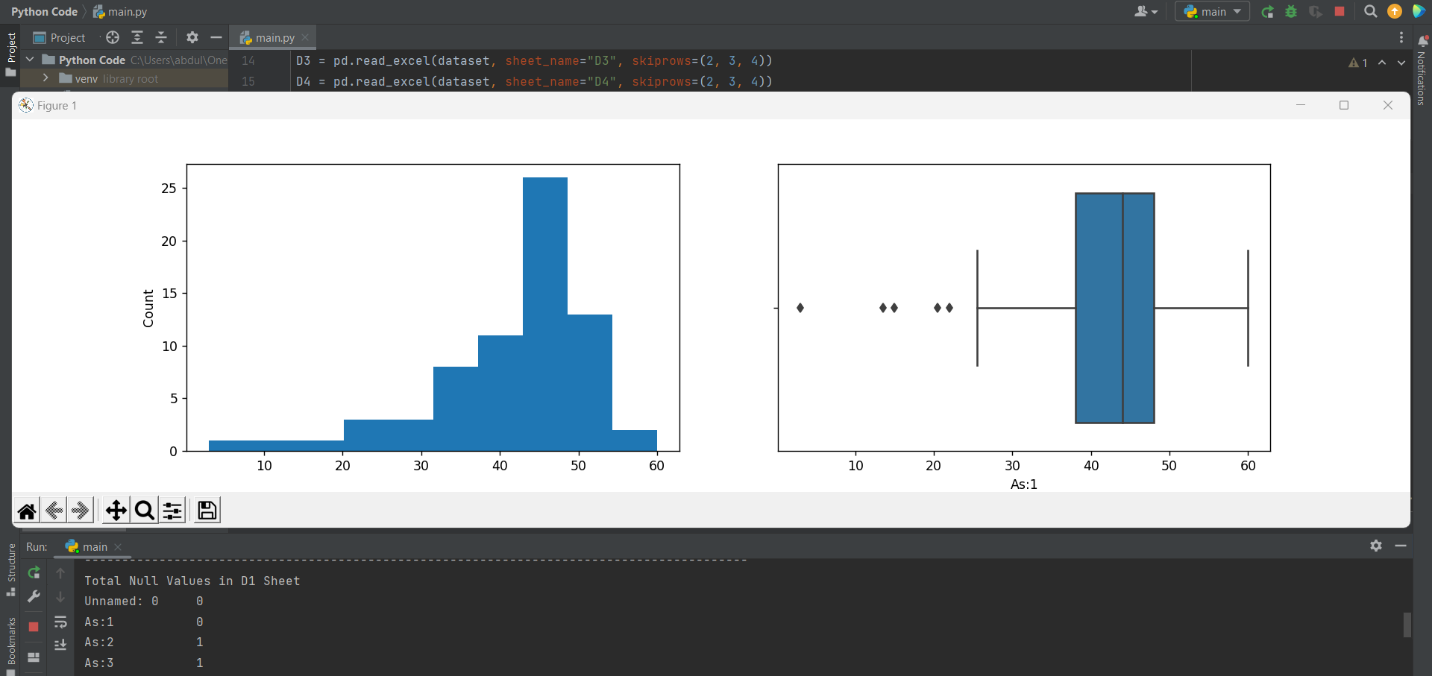


## Performing EDA:

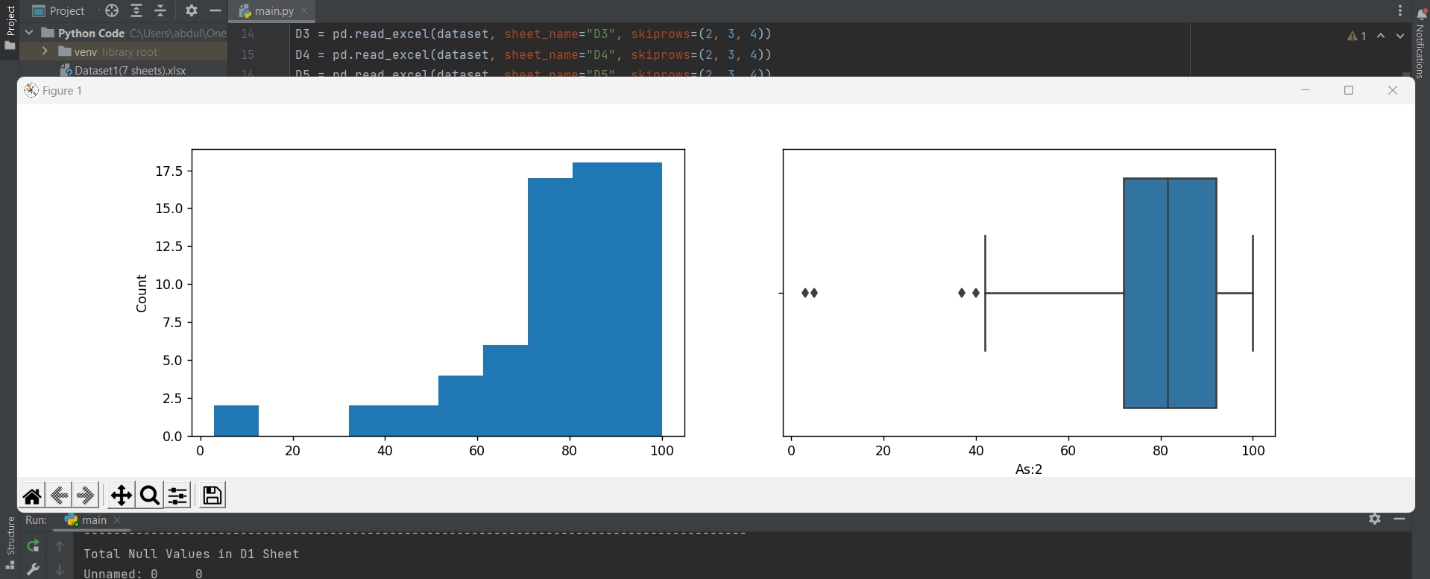
After performing all the previous steps, now it was time to move on to EDA. We’ve performed Univariate EDA on our dataset to extract important information from it, which tells us about the features and their count.

We’ve only performed it for sheet D1 because the process for the other is the same. It only differs in the results. The following images show the EDA of different attributes:

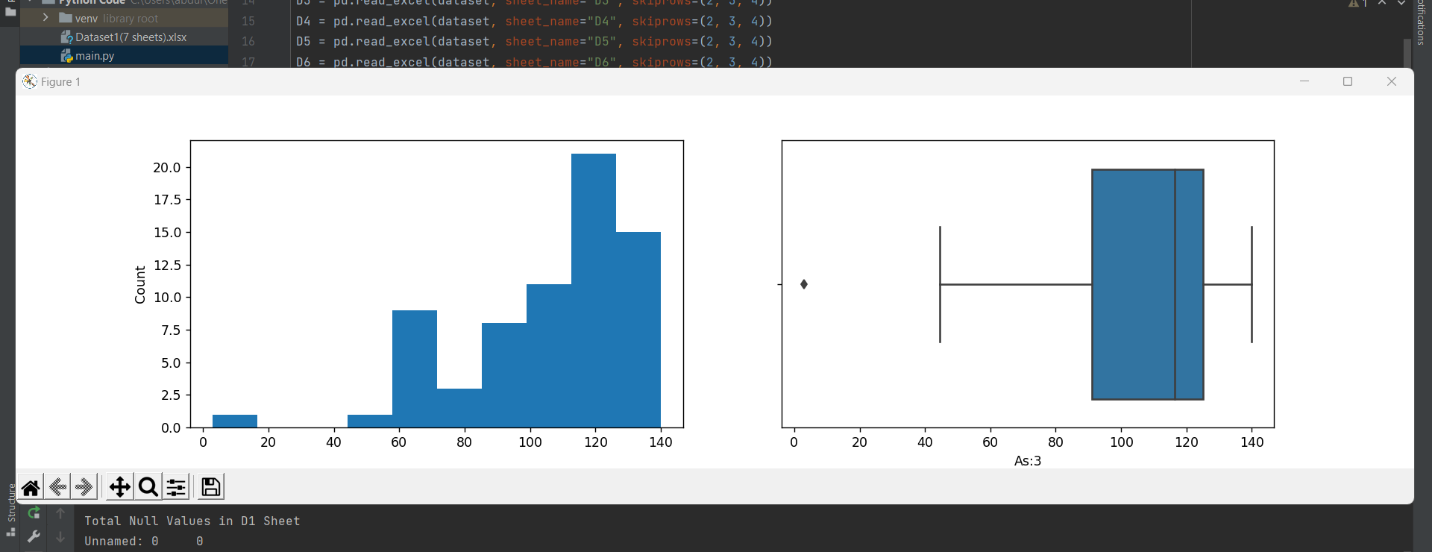
### Sheet D1 EDA Assignment 1:



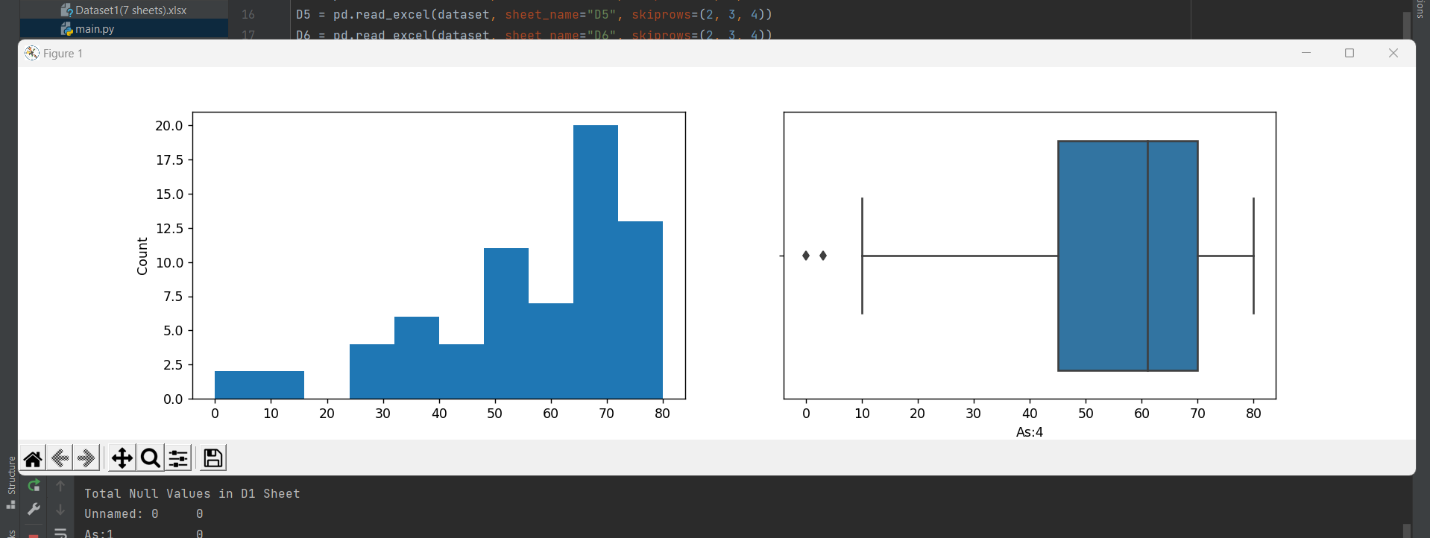
### Sheet D1 EDA Assignment 2:



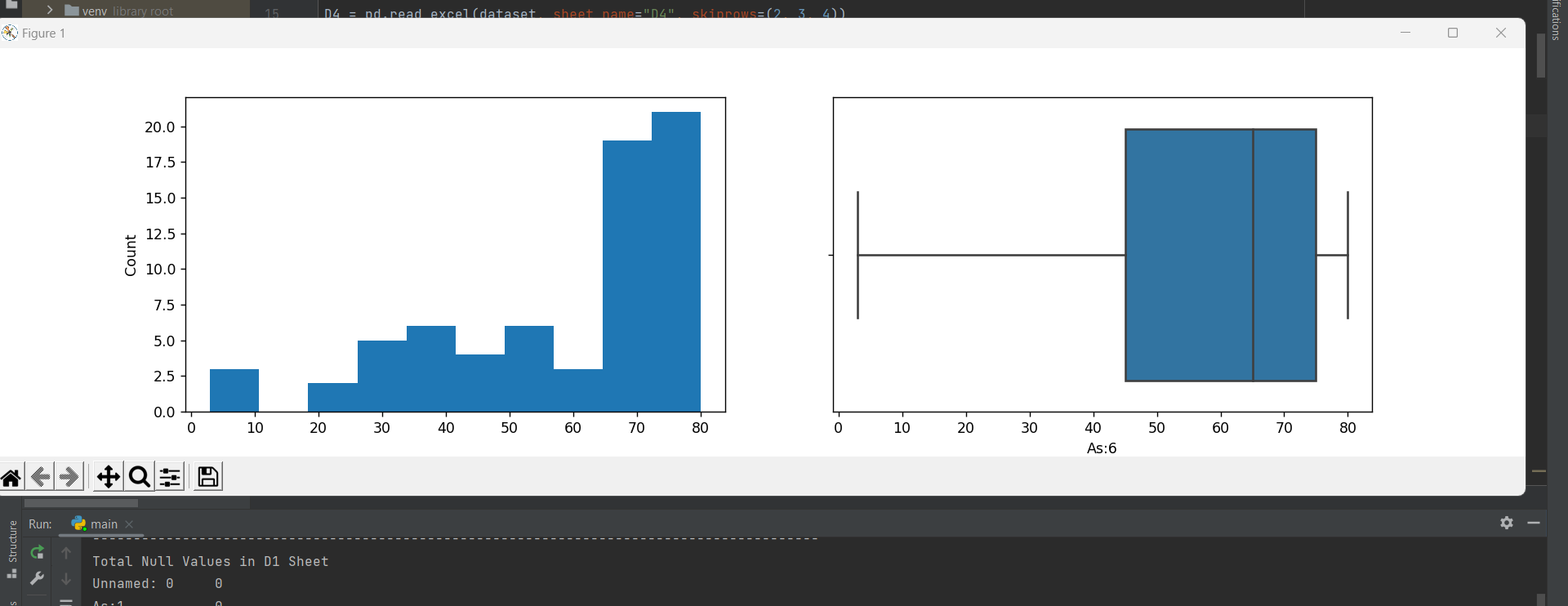
### Sheet D1 EDA Assignment 3:



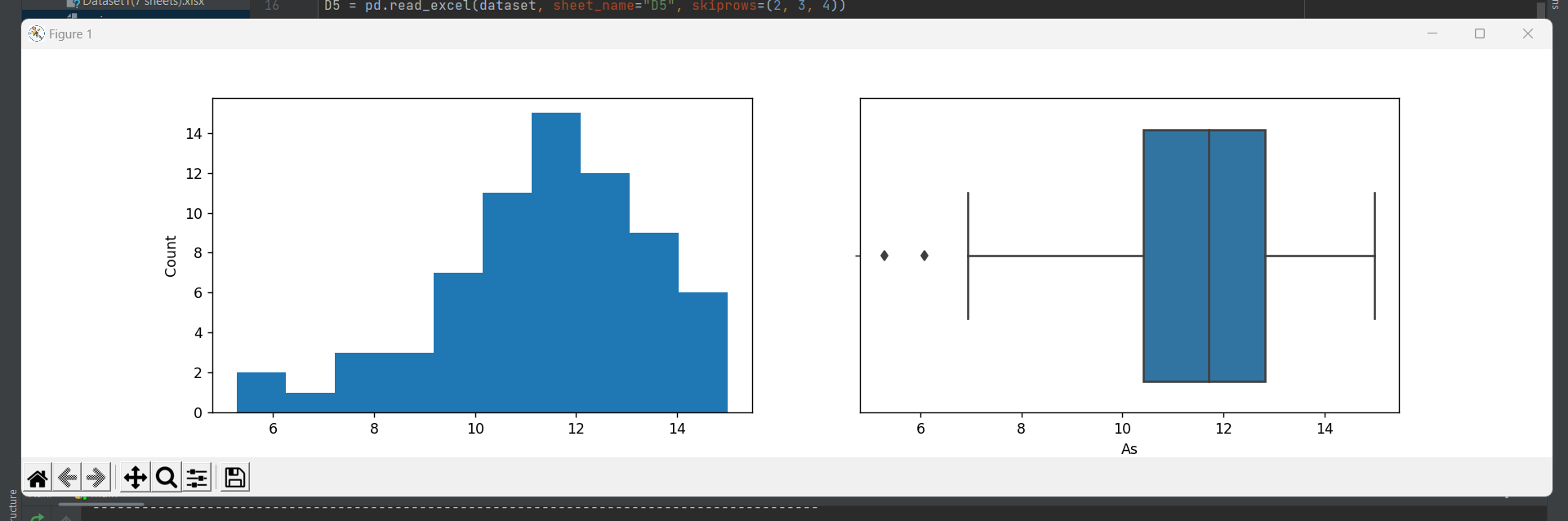
### Sheet D1 EDA Assignment 4:



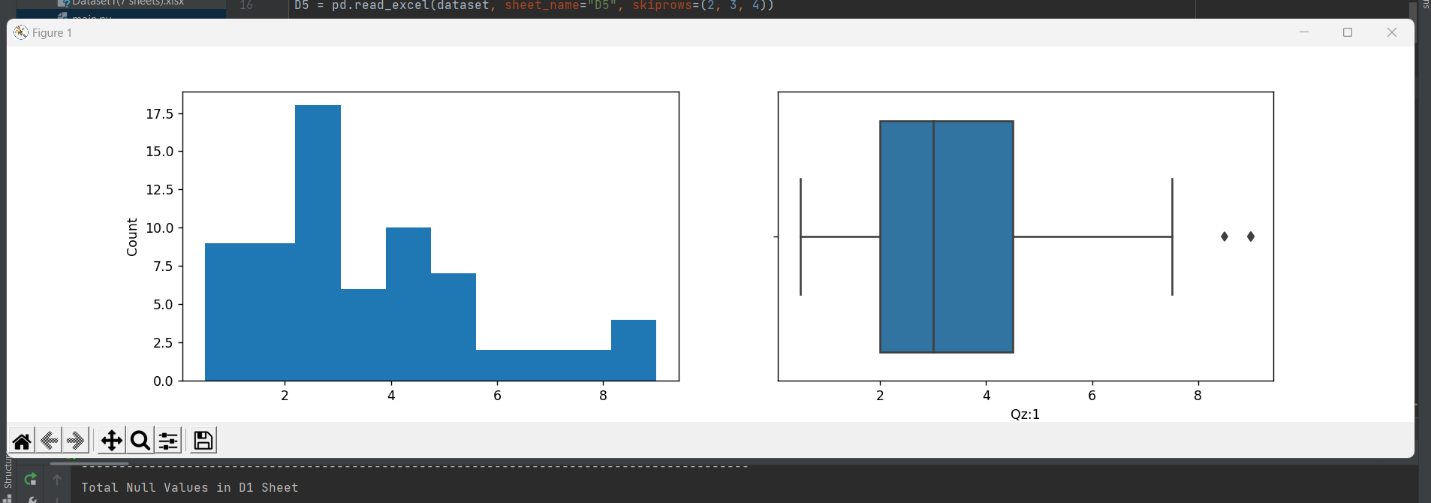
### Sheet D1 EDA Assignment 5:



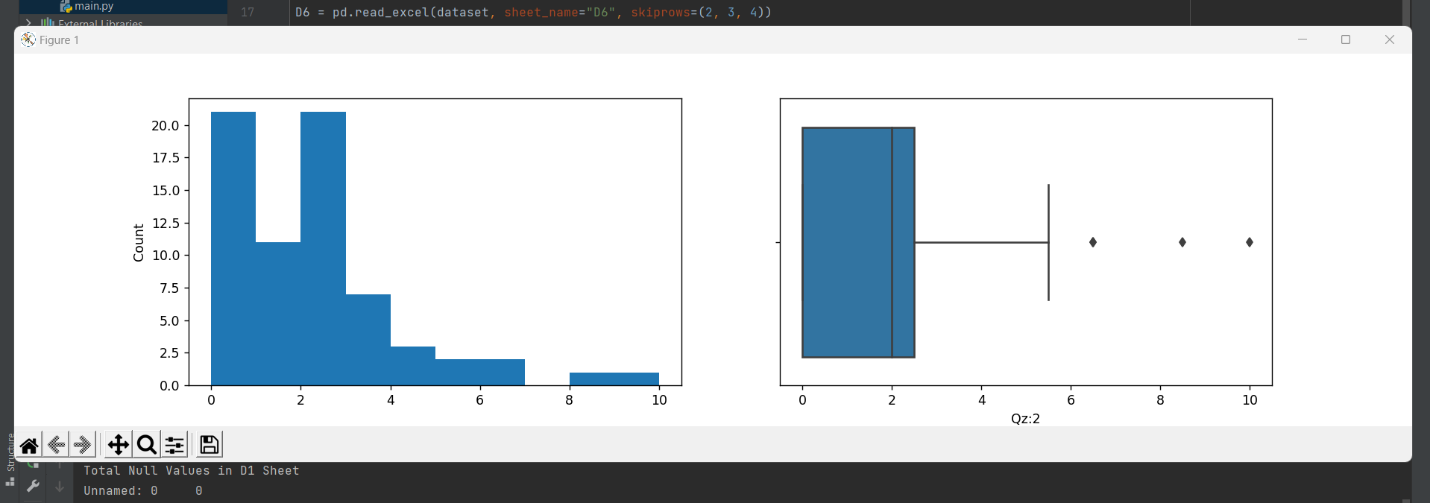
### Sheet D1 EDA All Assignments:



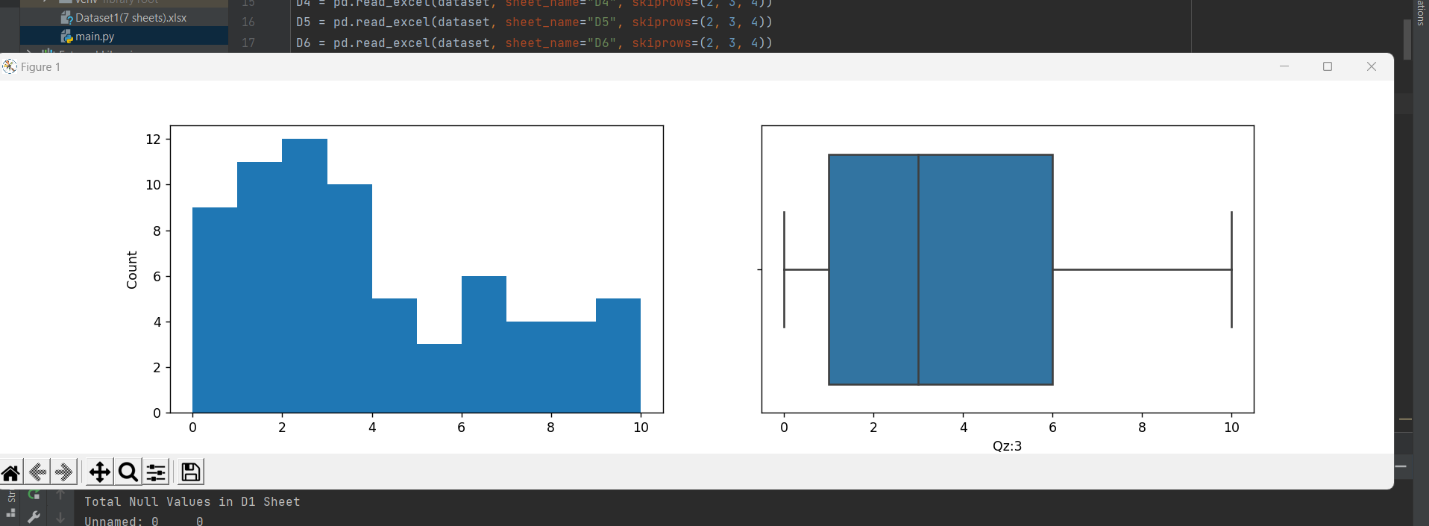
### Sheet D1 EDA Quiz 1:



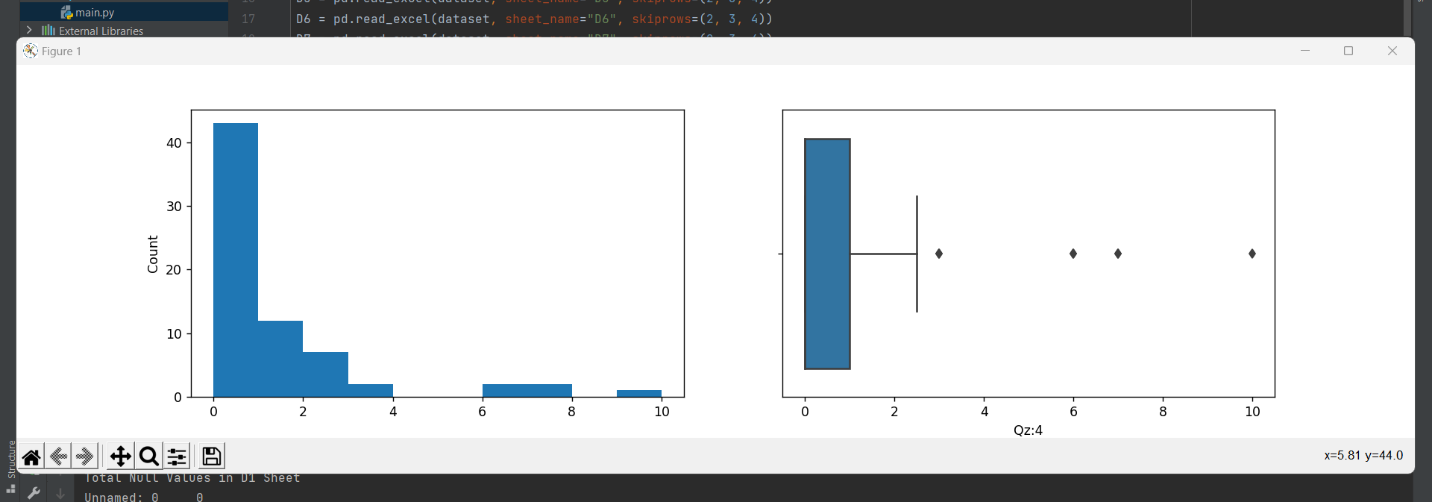
### Sheet D1 EDA Quiz 2:



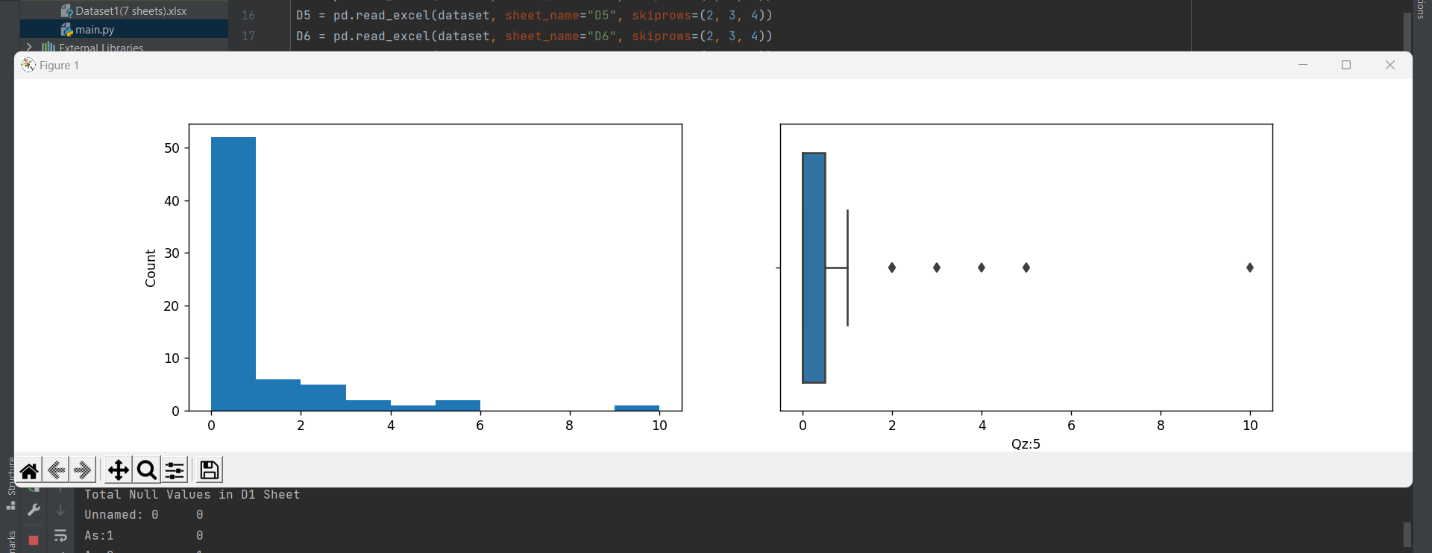
### Sheet D1 EDA Quiz 3:



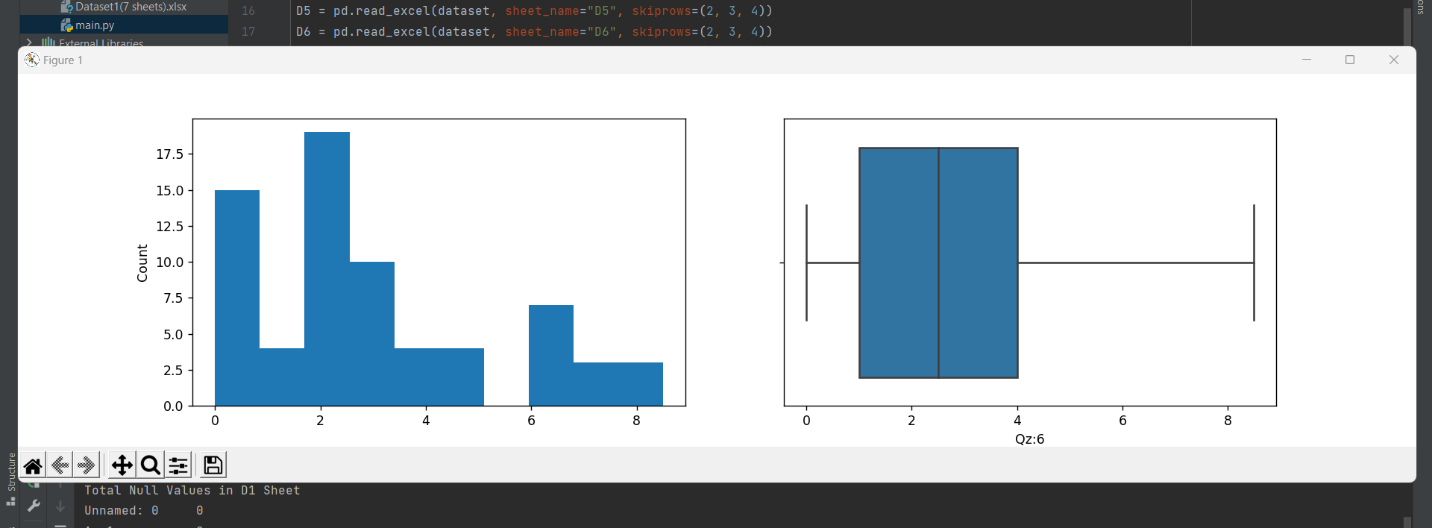
### Sheet D1 EDA Quiz 4:



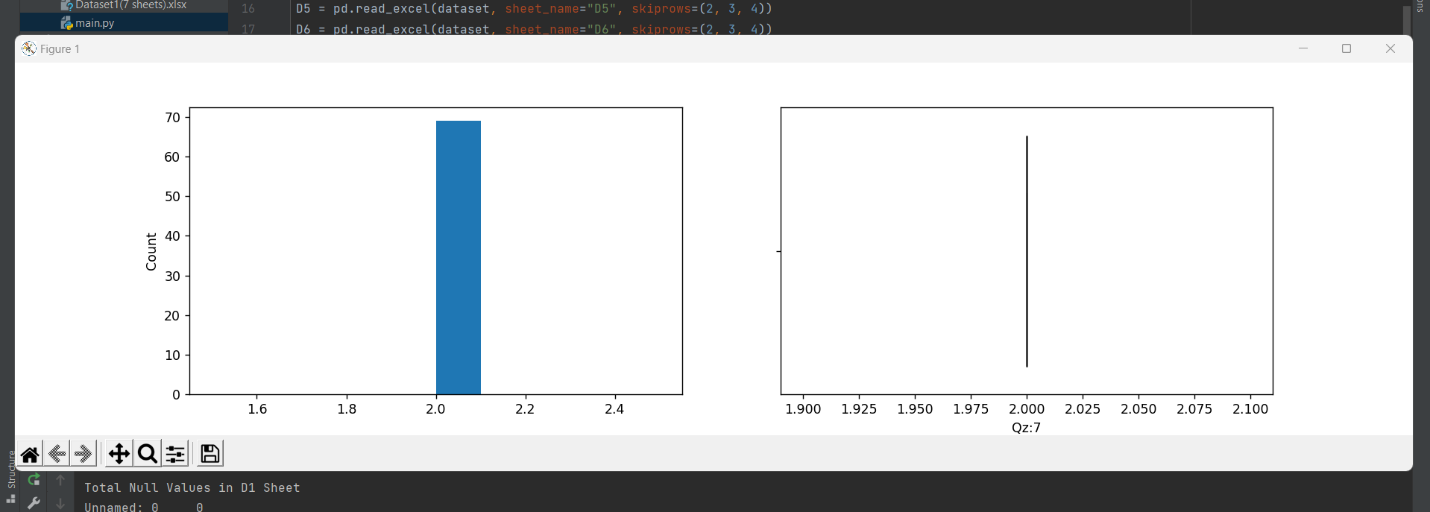
### Sheet D1 EDA Quiz 5:



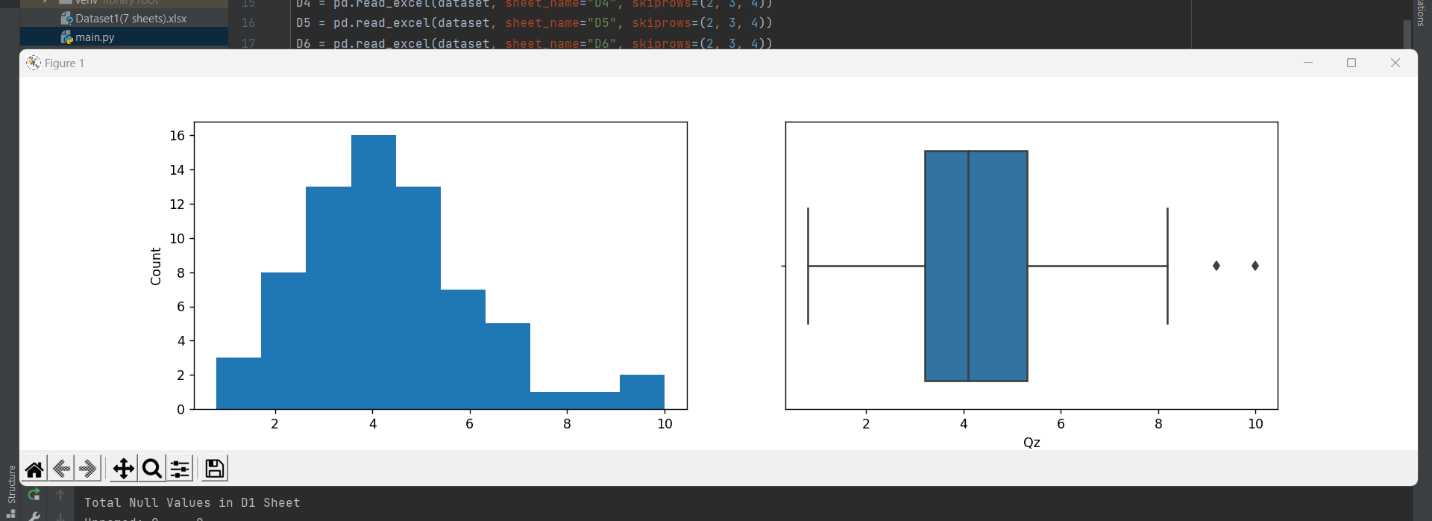
### Sheet D1 EDA Quiz 6:



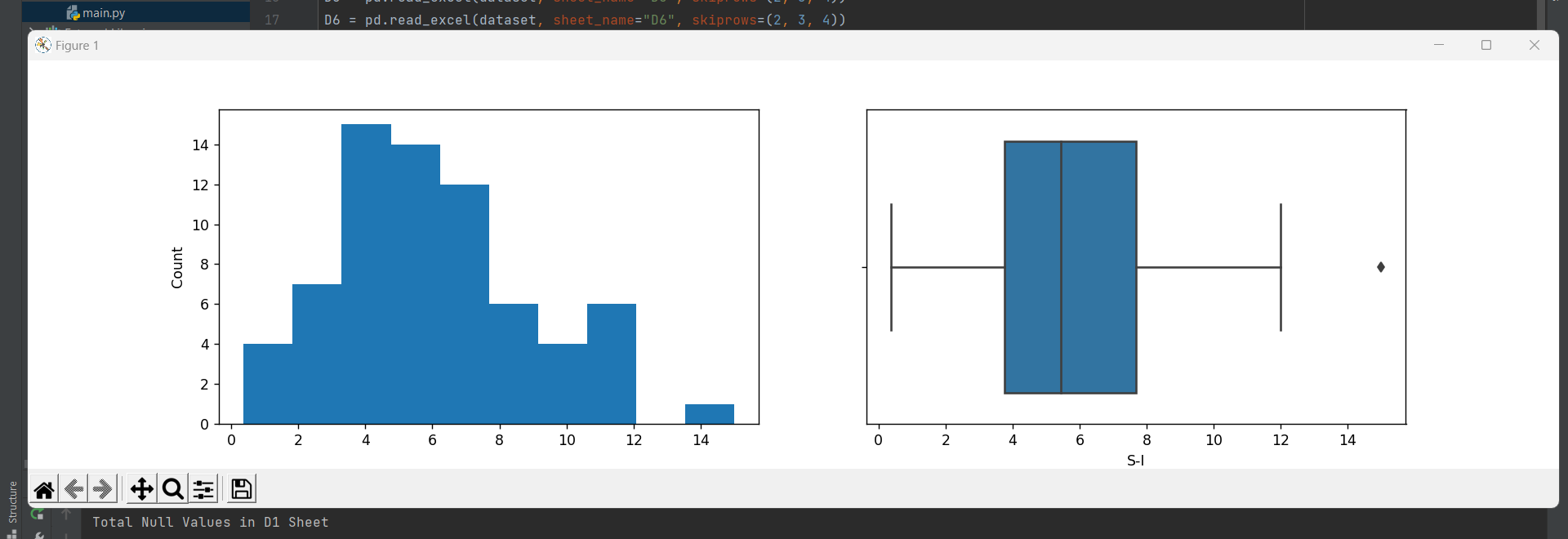
### Sheet D1 EDA Quiz 7:



### Sheet D1 EDA All Quizzes:



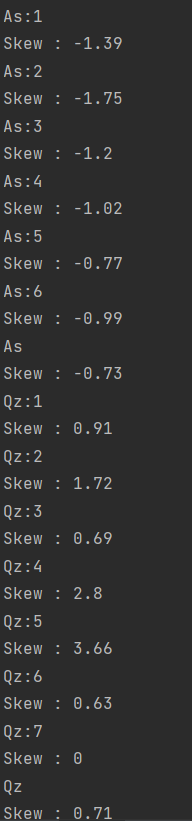
### Sheet D1 EDA Sessional I:



### Sheet D1 EDA Sessional II:



### Sheet D1 EDA Results:



This analysis shows us the skewness of the attributes depending upon their count and allows us to take steps to normalize the values using data transformation and make it eligible to perform Bivariate EDA on it.

But as we know our data only consists of numerical values hence there is no need to perform Bivariate EDA for further analysis. At this time, our data is clean and ready to proceed in the next phase.

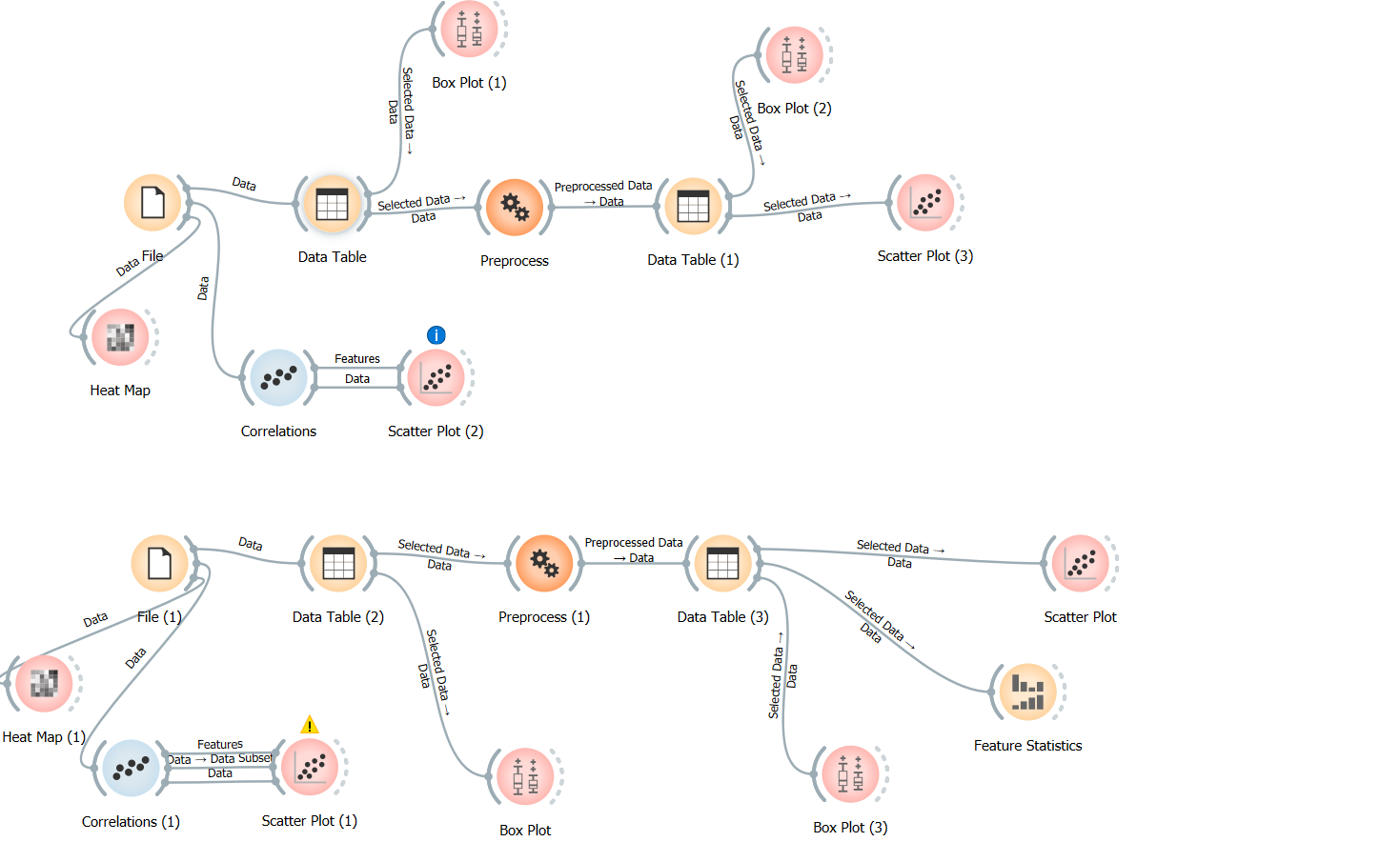
# **Python Code:**

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
import warnings  
  
warnings.filterwarnings('ignore')  
  
# Importing File and Sheets from the given Dataset  
dataset = pd.ExcelFile('Dataset1(7 sheets).xlsx', engine='openpyxl')  
# We don't need Row 2,3 and 4 because it adds nothing to our data.  
D1 = pd.read\_excel(dataset, sheet\_name="D1", skiprows=(2, 3, 4))  
D2 = pd.read\_excel(dataset, sheet\_name="D2", skiprows=(2, 3, 4))  
D3 = pd.read\_excel(dataset, sheet\_name="D3", skiprows=(2, 3, 4))  
D4 = pd.read\_excel(dataset, sheet\_name="D4", skiprows=(2, 3, 4))  
D5 = pd.read\_excel(dataset, sheet\_name="D5", skiprows=(2, 3, 4))  
D6 = pd.read\_excel(dataset, sheet\_name="D6", skiprows=(2, 3, 4))  
D7 = pd.read\_excel(dataset, sheet\_name="D7", skiprows=(2, 3, 4))  
  
# Checking for Duplication  
print("Total Duplications in D1 Sheet")  
print(D1.nunique())  
print("Total Duplications in D2 Sheet")  
print(D2.nunique())  
print("Total Duplications in D3 Sheet")  
print(D3.nunique())  
print("Total Duplications in D4 Sheet")  
print(D4.nunique())  
print("Total Duplications in D5 Sheet")  
print(D5.nunique())  
print("Total Duplications in D6 Sheet")  
print(D6.nunique())  
print("Total Duplications in D7 Sheet")  
print(D7.nunique())  
  
# Checking Null Values  
print("-----------------------------------------------------------------------------------------")  
print("Total Null Values in D1 Sheet")  
print(D1.isnull().sum())  
print("Total Null Values in D2 Sheet")  
print(D2.isnull().sum())  
print("Total Null Values in D3 Sheet")  
print(D3.isnull().sum())  
print("Total Null Values in D4 Sheet")  
print(D4.isnull().sum())  
print("Total Null Values in D5 Sheet")  
print(D5.isnull().sum())  
print("Total Null Values in D6 Sheet")  
print(D6.isnull().sum())  
print("Total Null Values in D7 Sheet")  
print(D7.isnull().sum())  
  
# Displaying Null Values using Heatmap  
print("-----------------------------------------------------------------------------------------")  
sns.heatmap(D1.isnull(), cbar=False, cmap='viridis')  
plt.title("D1 Null Values")  
plt.show()  
sns.heatmap(D2.isnull(), cbar=False, cmap='viridis')  
plt.title("D2 Null Values")  
plt.show()  
sns.heatmap(D3.isnull(), cbar=False, cmap='viridis')  
plt.title("D3 Null Values")  
plt.show()  
sns.heatmap(D4.isnull(), cbar=False, cmap='viridis')  
plt.title("D4 Null Values")  
plt.show()  
sns.heatmap(D5.isnull(), cbar=False, cmap='viridis')  
plt.title("D5 Null Values")  
plt.show()  
sns.heatmap(D6.isnull(), cbar=False, cmap='viridis')  
plt.title("D6 Null Values")  
plt.show()  
sns.heatmap(D7.isnull(), cbar=False, cmap='viridis')  
plt.title("D7 Null Values")  
plt.show()  
  
# Replacing missing values with Median of the corresponding column  
print("-----------------------------------------------------------------------------------------")  
num\_col\_D1 = ['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz', 'S-I', 'S-II']  
for col in num\_col\_D1:  
 D1[col] = pd.to\_numeric(D1[col])  
 D1[col].fillna(D1[col].median(), inplace=True)  
grades\_mode = D1.Grade.mode()  
D1.Grade.fillna(grades\_mode, inplace=True)  
  
num\_col\_D2 = ['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz', 'S-I', 'S-II']  
for col in num\_col\_D2:  
 D2[col] = pd.to\_numeric(D2[col])  
 D2[col].fillna(D2[col].median(), inplace=True)  
grades\_mode = D2.Grade.mode()  
D2.Grade.fillna(grades\_mode, inplace=True)  
  
num\_col\_D3 = ['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz:8', 'Qz', 'S-I', 'S-II']  
for col in num\_col\_D3:  
 D3[col] = pd.to\_numeric(D3[col])  
 D3[col].fillna(D3[col].median(), inplace=True)  
grades\_mode = D3.Grade.mode()  
D3.Grade.fillna(grades\_mode, inplace=True)  
  
num\_col\_D4 = ['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As:7', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4',  
 'Qz:5', 'Qz', 'S-I', 'S-II']  
for col in num\_col\_D4:  
 D4[col] = pd.to\_numeric(D4[col])  
 D4[col].fillna(D4[col].median(), inplace=True)  
grades\_mode = D4.Grade.mode()  
D4.Grade.fillna(grades\_mode, inplace=True)  
  
num\_col\_D5 = ['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz:8', 'Qz', 'S-I', 'S-II']  
for col in num\_col\_D5:  
 D5[col] = pd.to\_numeric(D5[col])  
 D5[col].fillna(D5[col].median(), inplace=True)  
grades\_mode = D5.Grade.mode()  
D5.Grade.fillna(grades\_mode, inplace=True)  
  
num\_col\_D6 = ['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz', 'S-I', 'S-II']  
for col in num\_col\_D6:  
 D6[col] = pd.to\_numeric(D6[col])  
 D6[col].fillna(D6[col].median(), inplace=True)  
grades\_mode = D6.Grade.mode()  
D6.Grade.fillna(grades\_mode, inplace=True)  
  
num\_col\_D7 = ['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz:8', 'Qz', 'S-I', 'S-II']  
for col in num\_col\_D7:  
 D7[col] = pd.to\_numeric(D7[col])  
 D7[col].fillna(D7[col].median(), inplace=True)  
grades\_mode = D7.Grade.mode()  
D7.Grade.fillna(grades\_mode, inplace=True)  
  
# Performing Data-Reduction Techniques  
# Removing 1st Column from data as it only contains serial number that is not helpful for us  
print("-----------------------------------------------------------------------------------------")  
D1 = D1.drop(columns=D1.columns[0])  
D2 = D2.drop(columns=D2.columns[0])  
D3 = D3.drop(columns=D3.columns[0])  
D4 = D4.drop(columns=D4.columns[0])  
D5 = D5.drop(columns=D5.columns[0])  
D6 = D6.drop(columns=D6.columns[0])  
D7 = D7.drop(columns=D7.columns[0])  
  
# Data Cleared. Now proceeding to Checking Correlation of attributes.  
print("-----------------------------------------------------------------------------------------")  
plt.figure(figsize=(13, 13))  
sns.heatmap(D1[['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz', 'S-I', 'S-II']].corr(), cbar=True, annot=True, cmap='Blues')  
plt.title("D1 Correlation Matrix")  
plt.show()  
  
plt.figure(figsize=(13, 13))  
sns.heatmap(D2[['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz', 'S-I', 'S-II']].corr(), cbar=True, annot=True, cmap='Blues')  
plt.title("D2 Correlation Matrix")  
plt.show()  
  
plt.figure(figsize=(13, 13))  
sns.heatmap(D3[['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz:8', 'Qz', 'S-I', 'S-II']].corr(), cbar=True, annot=True, cmap='Blues')  
plt.title("D3 Correlation Matrix")  
plt.show()  
  
plt.figure(figsize=(13, 13))  
sns.heatmap(D4[['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As:7', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4',  
 'Qz:5', 'Qz', 'S-I', 'S-II']].corr(), cbar=True, annot=True, cmap='Blues')  
plt.title("D4 Correlation Matrix")  
plt.show()  
  
plt.figure(figsize=(13, 13))  
sns.heatmap(D5[['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz:8', 'Qz', 'S-I', 'S-II']].corr(), cbar=True, annot=True, cmap='Blues')  
plt.title("D5 Correlation Matrix")  
plt.show()  
  
plt.figure(figsize=(13, 13))  
sns.heatmap(D6[['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz', 'S-I', 'S-II']].corr(), cbar=True, annot=True, cmap='Blues')  
plt.title("D6 Correlation Matrix")  
plt.show()  
  
plt.figure(figsize=(13, 13))  
sns.heatmap(D7[['As:1', 'As:2', 'As:3', 'As:4', 'As:5', 'As:6', 'As', 'Qz:1', 'Qz:2', 'Qz:3', 'Qz:4', 'Qz:4', 'Qz:5',  
 'Qz:6', 'Qz:7', 'Qz:8', 'Qz', 'S-I', 'S-II']].corr(), cbar=True, annot=True, cmap='Blues')  
plt.title("D7 Correlation Matrix")  
plt.show()  
  
# Showing the relation between assignments weightage and Grades  
print("-----------------------------------------------------------------------------------------")  
D1.groupby('Grade')['As'].mean().plot.bar()  
plt.title("Grade vs Assignment")  
plt.show()  
  
# Showing the relation between quizzes weightage and Grades  
D1.groupby('Grade')['Qz'].mean().plot.bar()  
plt.title("Grade vs Quizzes")  
plt.show()  
  
# Showing the relation between S-I weightage and Grades  
D1.groupby('Grade')['S-I'].mean().plot.bar()  
plt.title("Grade vs S-I")  
plt.show()  
  
# Showing the relation between S-II weightage and Grades  
D1.groupby('Grade')['S-II'].mean().plot.bar()  
plt.title("Grade vs S-II")  
plt.show()  
  
# Same can be done for D2 to D7 for further analysis  
  
# Performing EDA on D1  
# Selecting only Numerical Values  
print("-----------------------------------------------------------------------------------------")  
num\_cols = D1.select\_dtypes(include=np.number).columns.tolist()  
for col in num\_cols:  
 print(col)  
 print('Skew :', round(D1[col].skew(), 2))  
 plt.figure(figsize=(15, 4))  
 plt.subplot(1, 2, 1)  
 D1[col].hist(grid=False)  
 plt.ylabel('Count')  
 plt.subplot(1, 2, 2)  
 sns.boxplot(x=D1[col])  
 plt.show()  
# Same can be done for D2 to D7 for further analysis

# **Working Using Orange:**

D1 to D7 process is same just showing the D1:

## Workflow Demonstration:

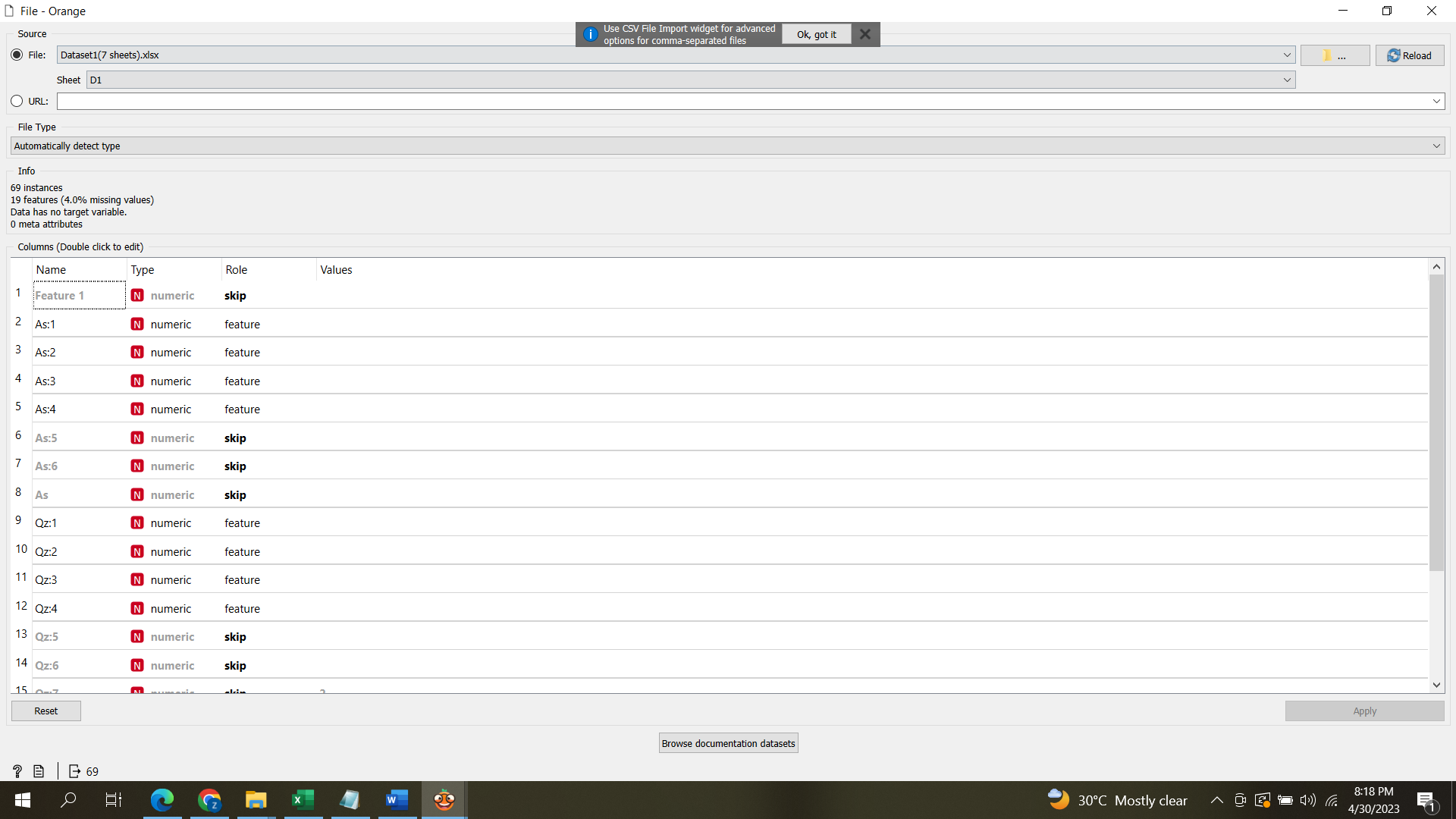


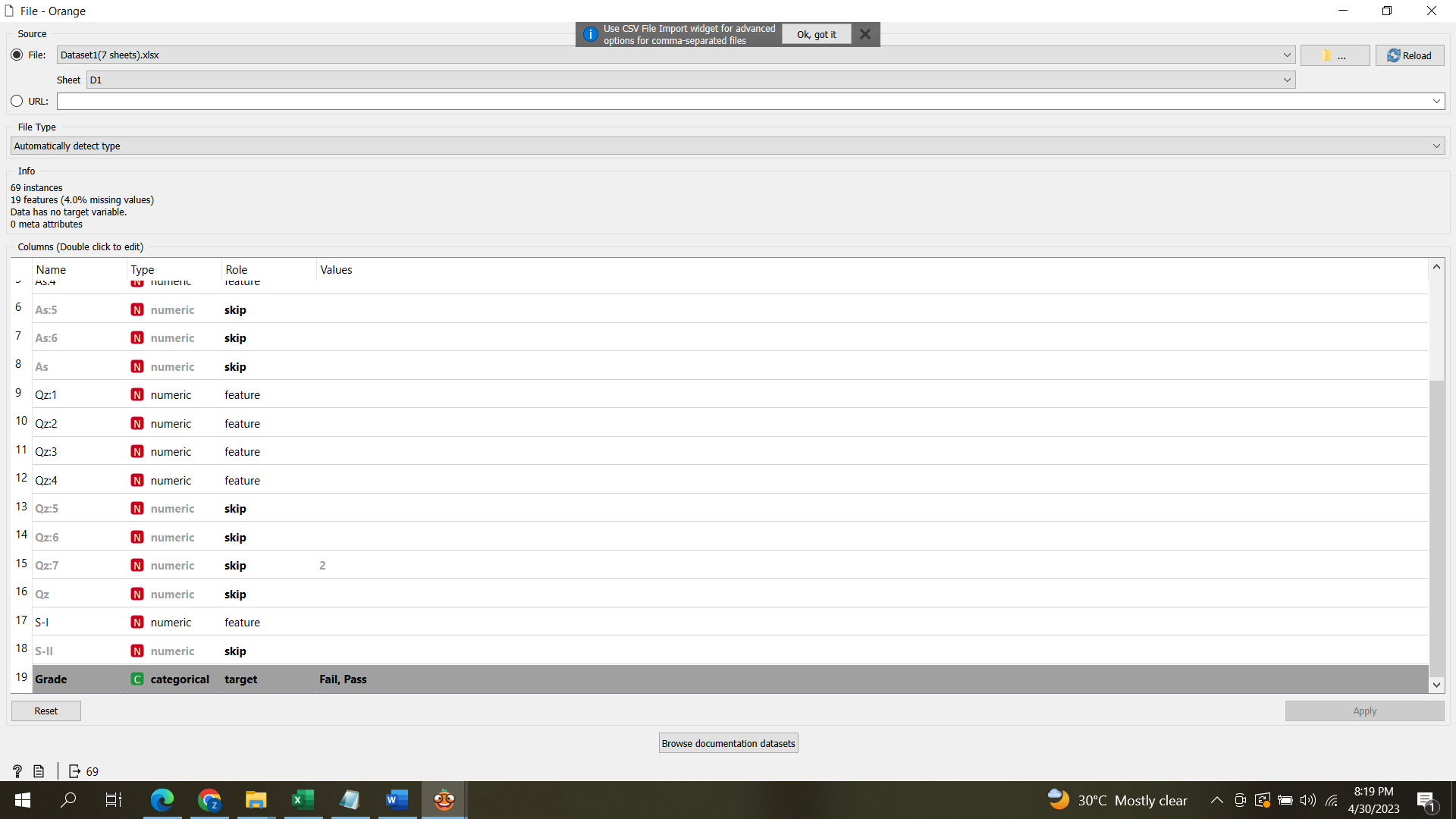
**Before Final**

**Before Mid 2**

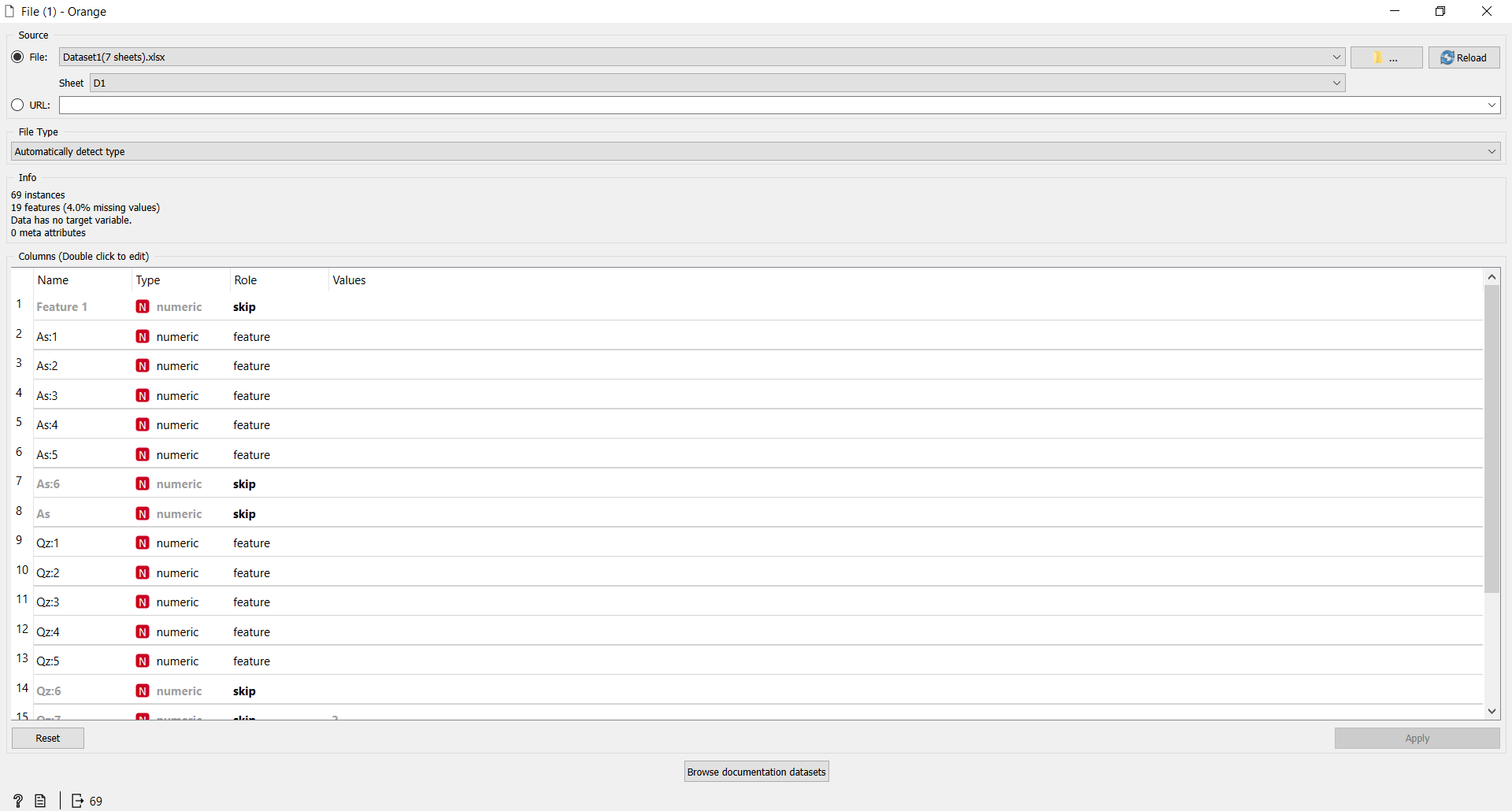
## Data Importing:

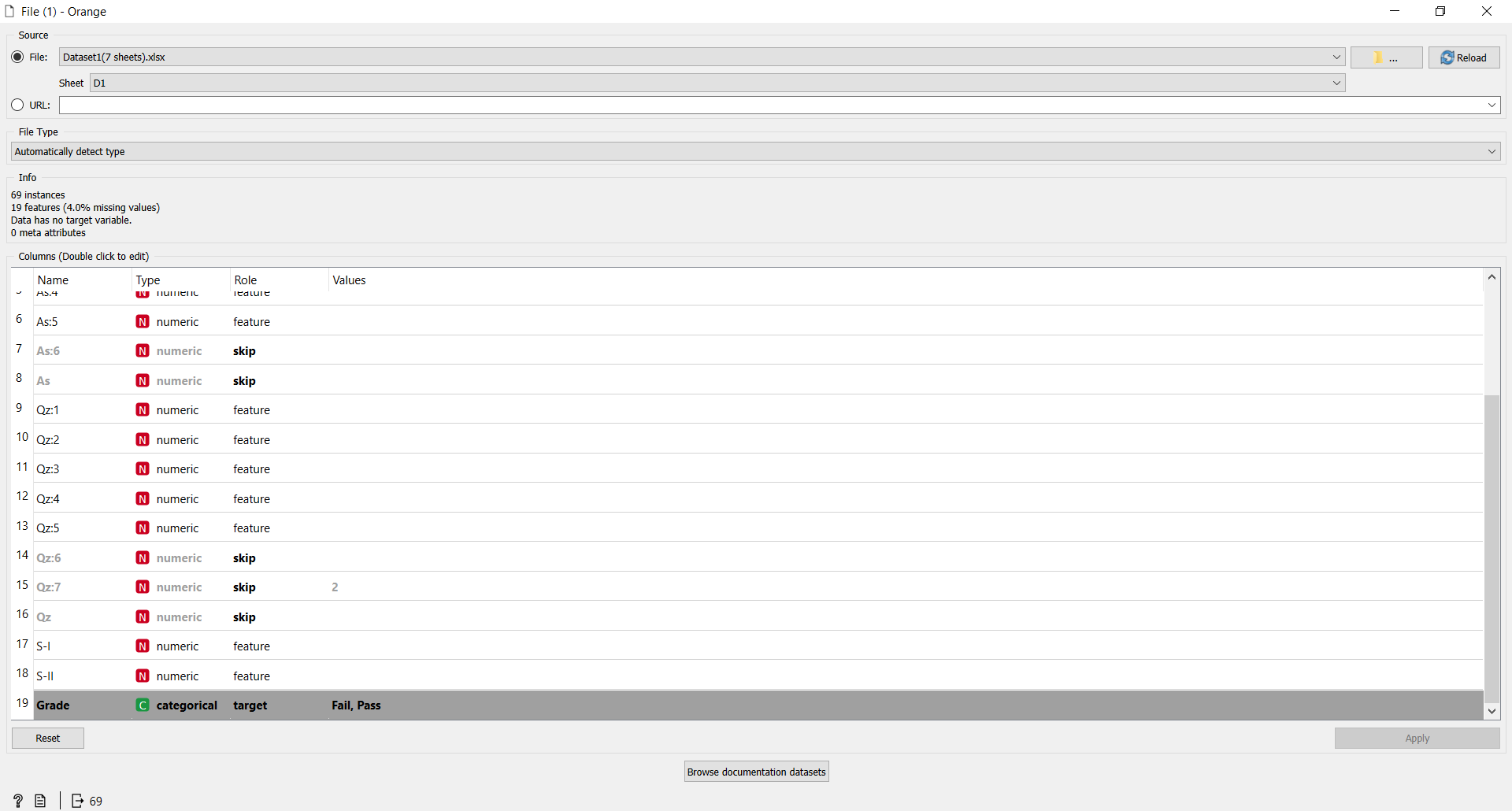
Importing the data from D1 of Assign 1 to 4, Quiz 1 to 4 & S-1 for before mid-2 grade suggestion and set their type as all are feature and grade is target.





Importing the data from D1 of Assign 1 to 5, Quiz 1 to 5 & S-1, S-2 for before final grade suggestion and set their type as all are feature and grade is target.

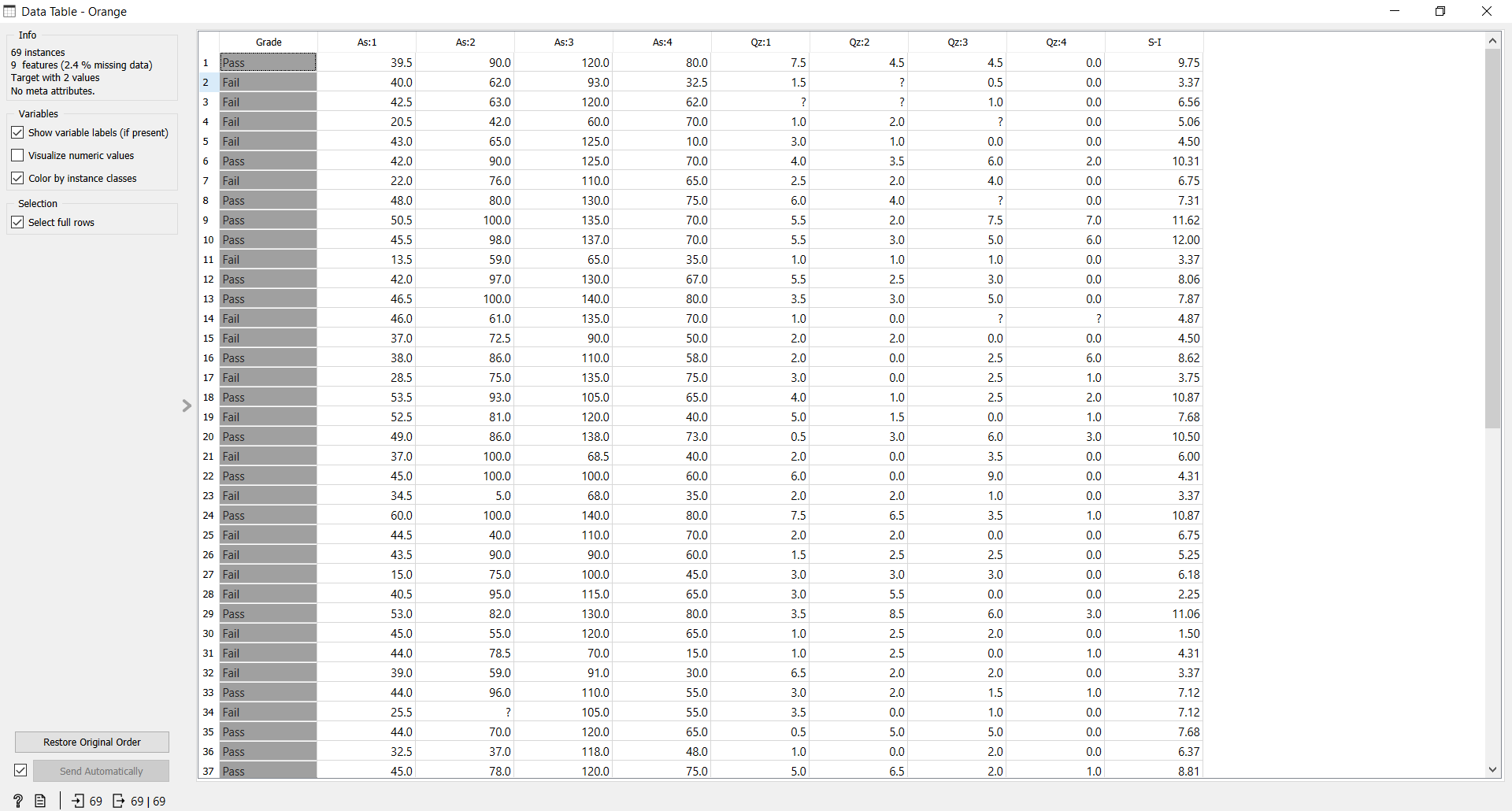




Importing process is same from as D1 from D2 to D7.

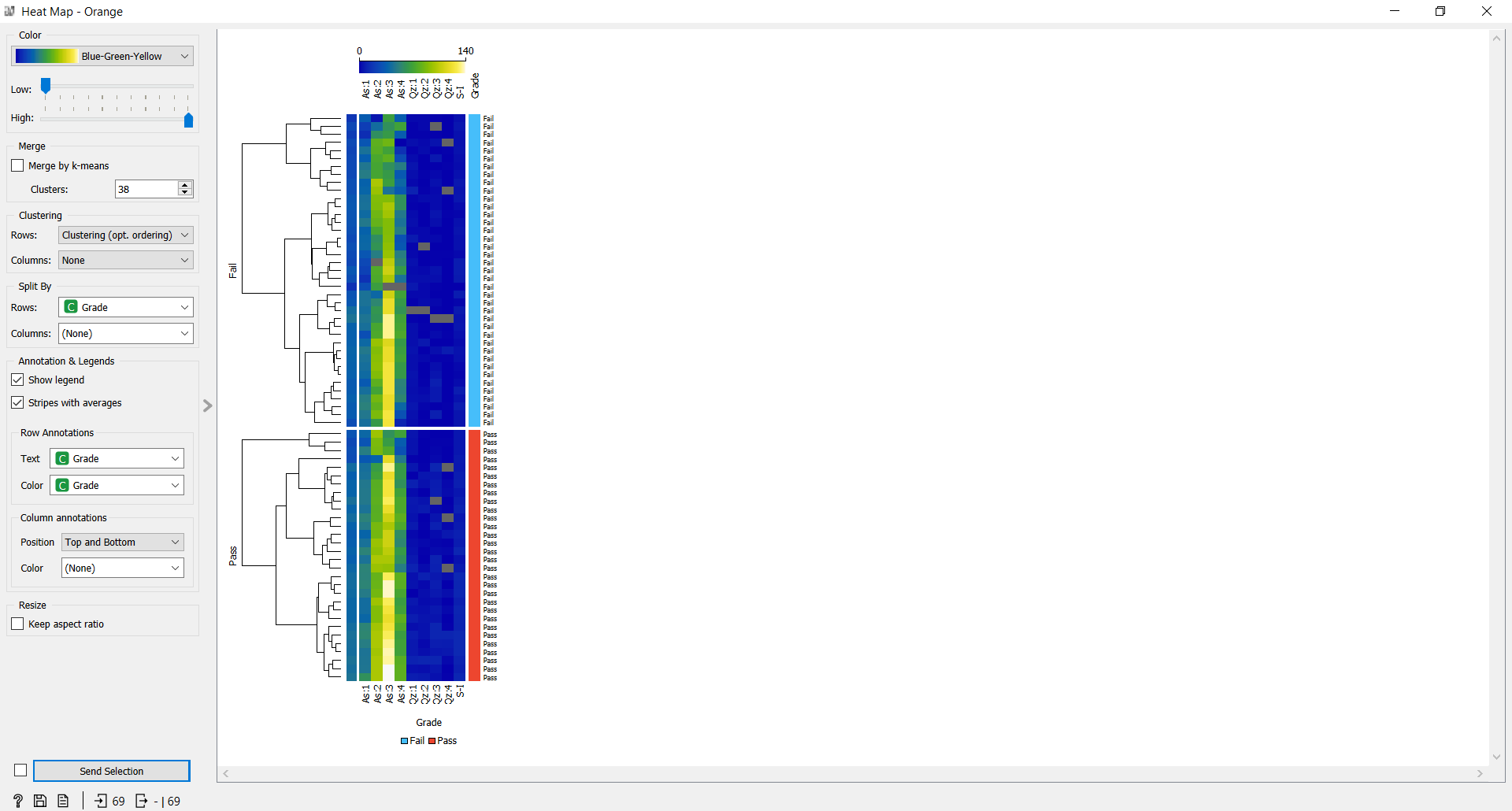
## Covert data into Table from File:

Data imported from file in table format is shown in table as below:



## Heat Map:

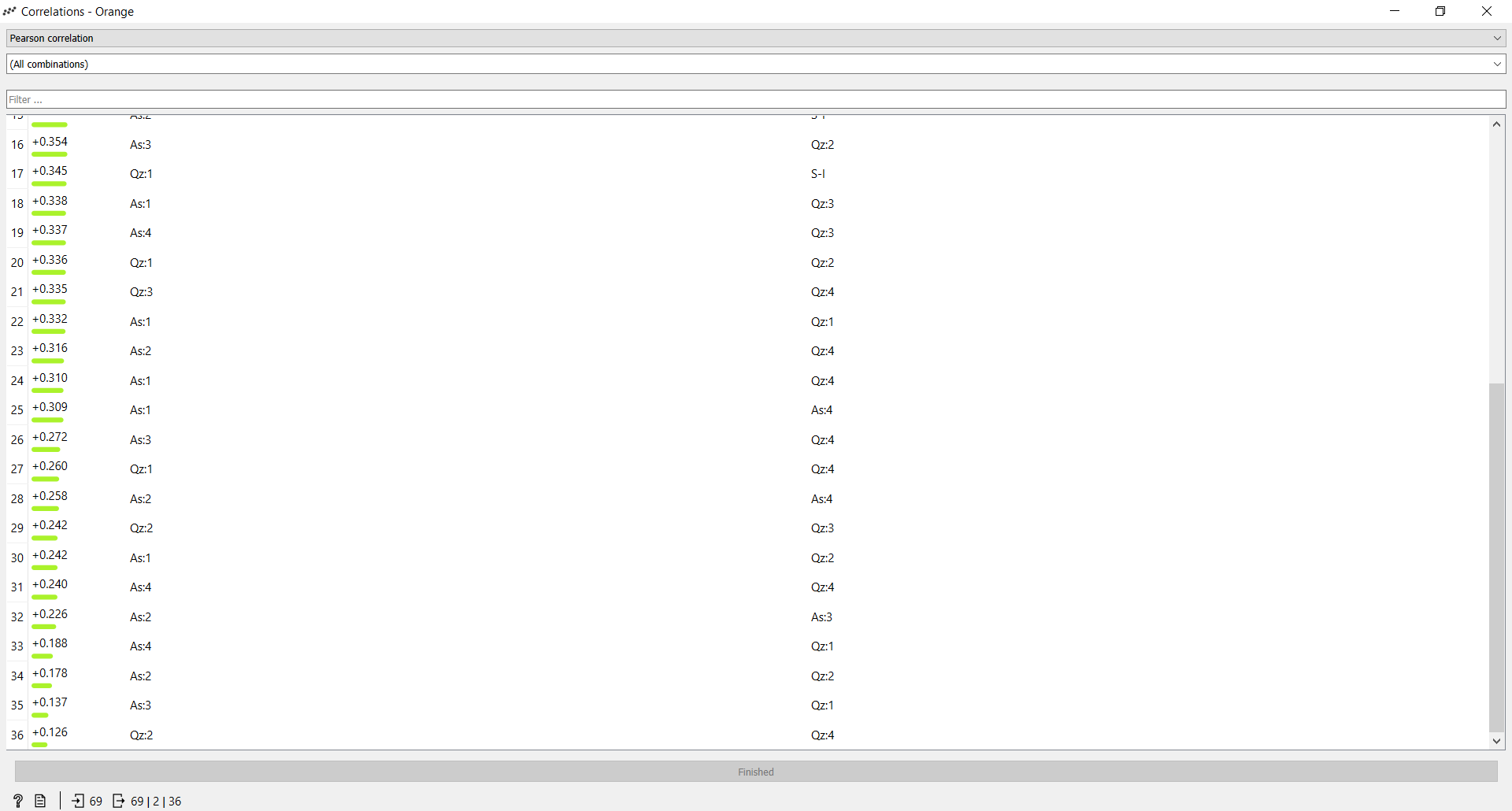
Heat map is a graphical method for visualizing attribute values in a two-way matrix. It only works on datasets containing numeric variables. Heat map shows low expressions in blue and high expressions in yellow and white.



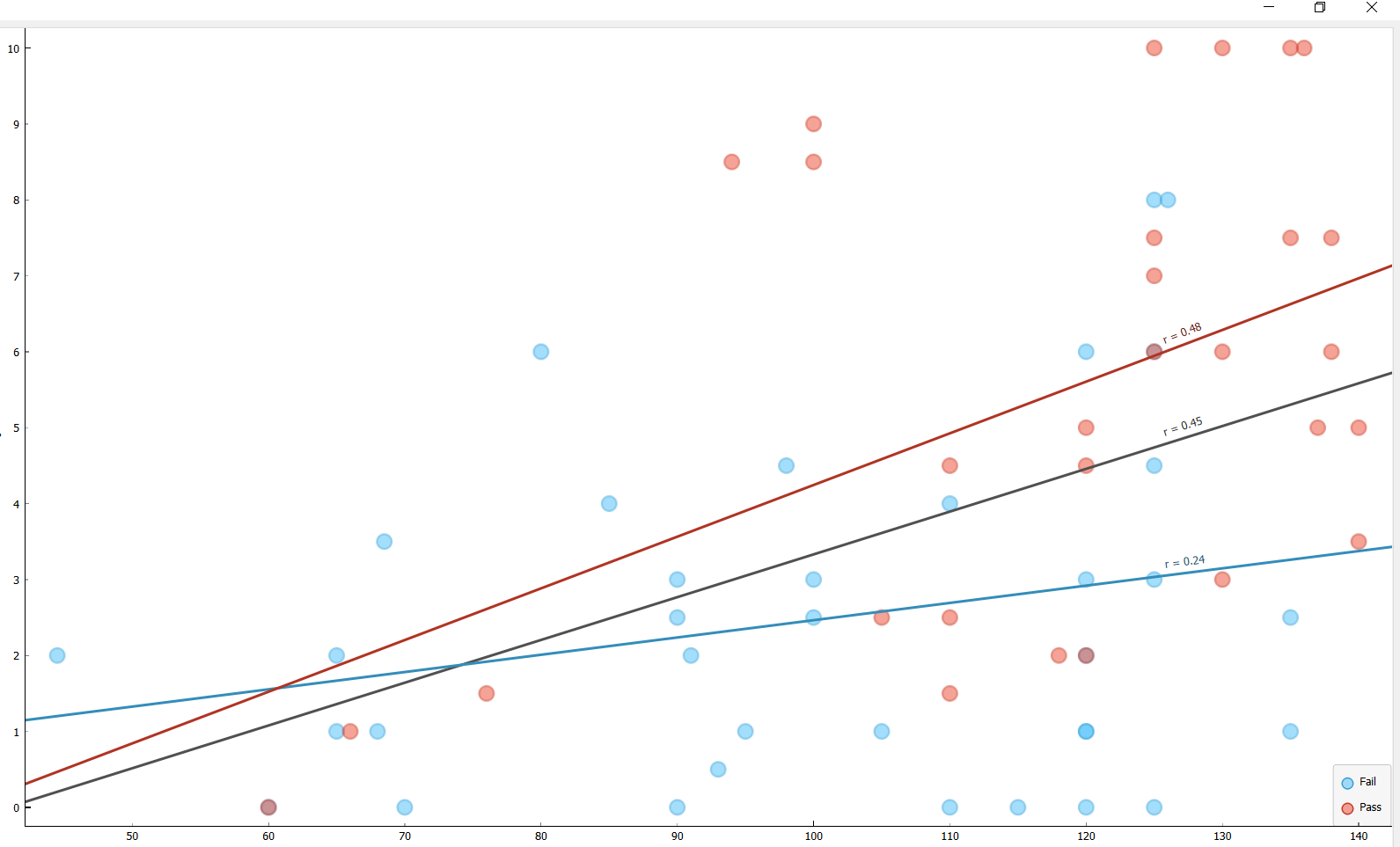
## Co-relation:

Load the File widget and connect it to Correlations (Pearson). Positively correlated feature pairs will be at the top of the list and negatively correlated will be at the bottom.



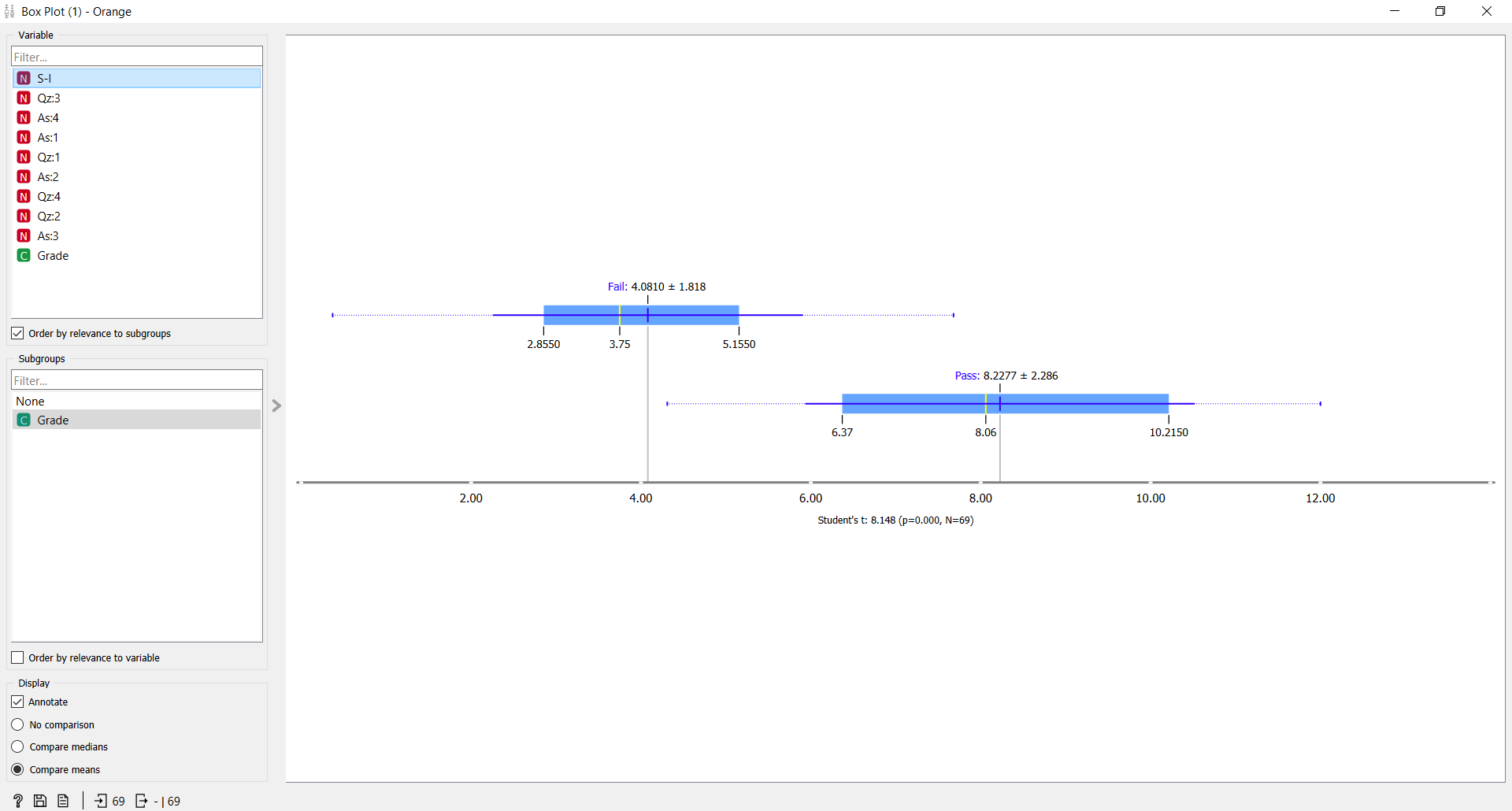


## Scatter plot:



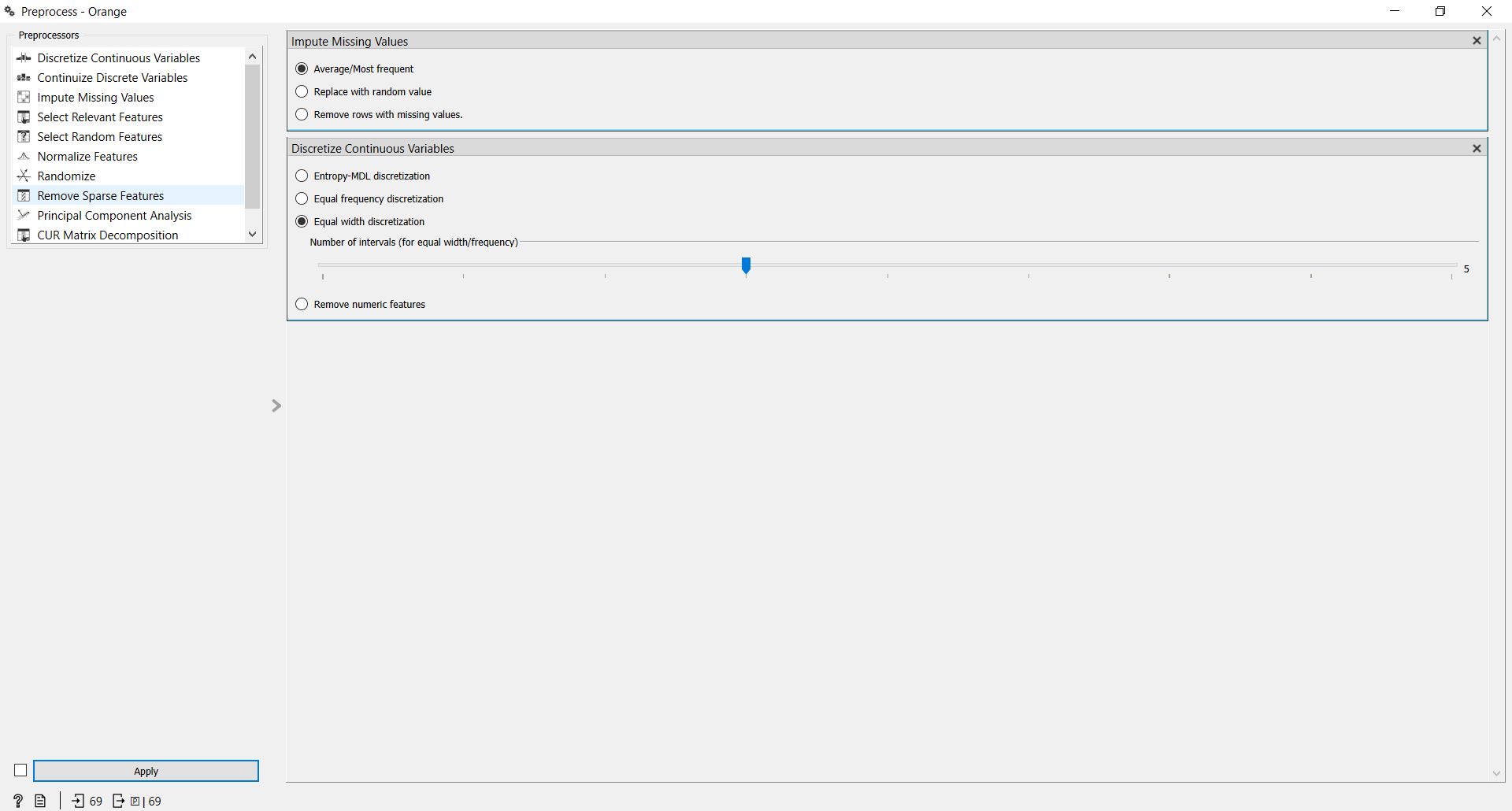
## Box Plot:

S1 & grade

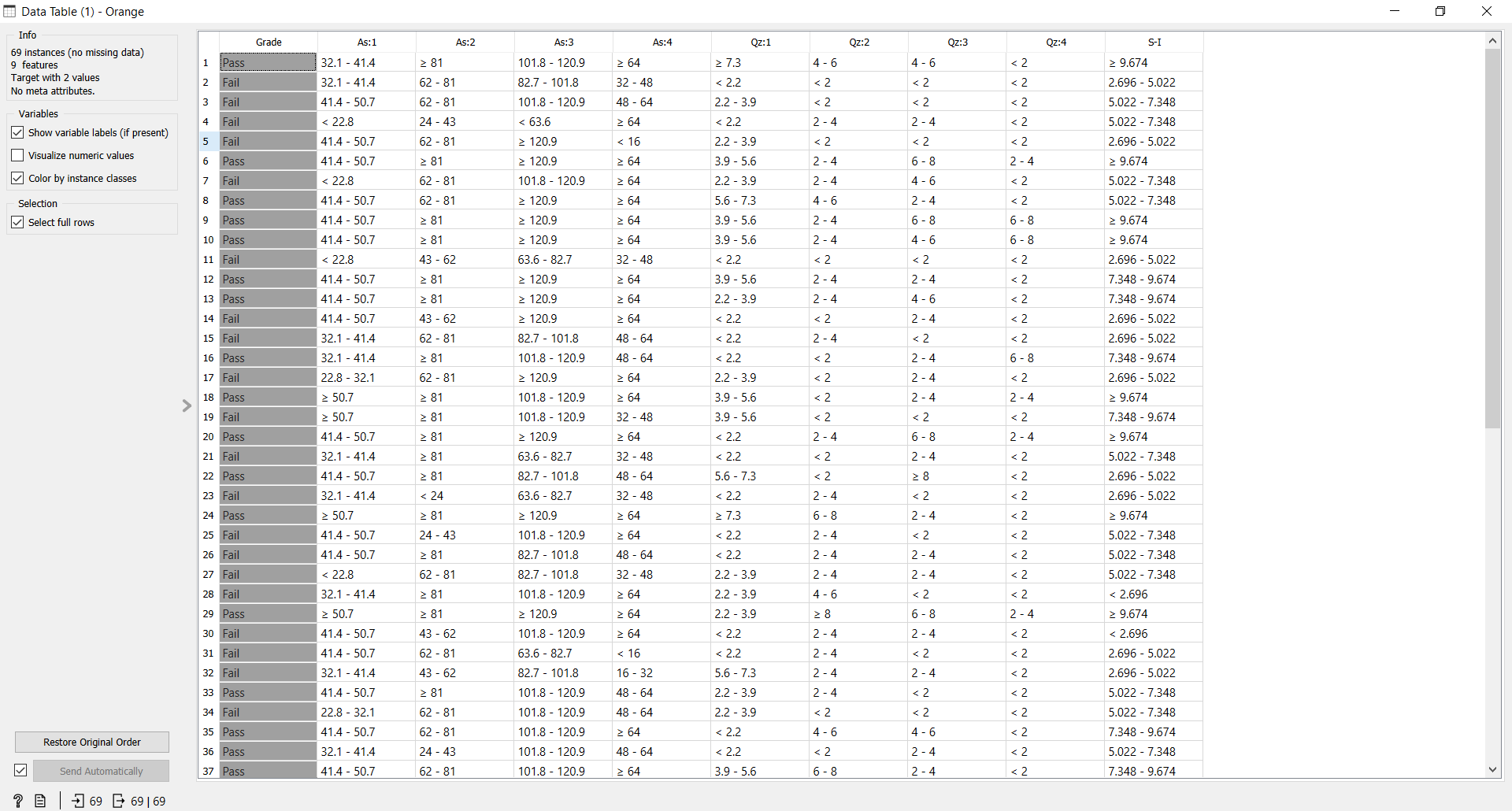


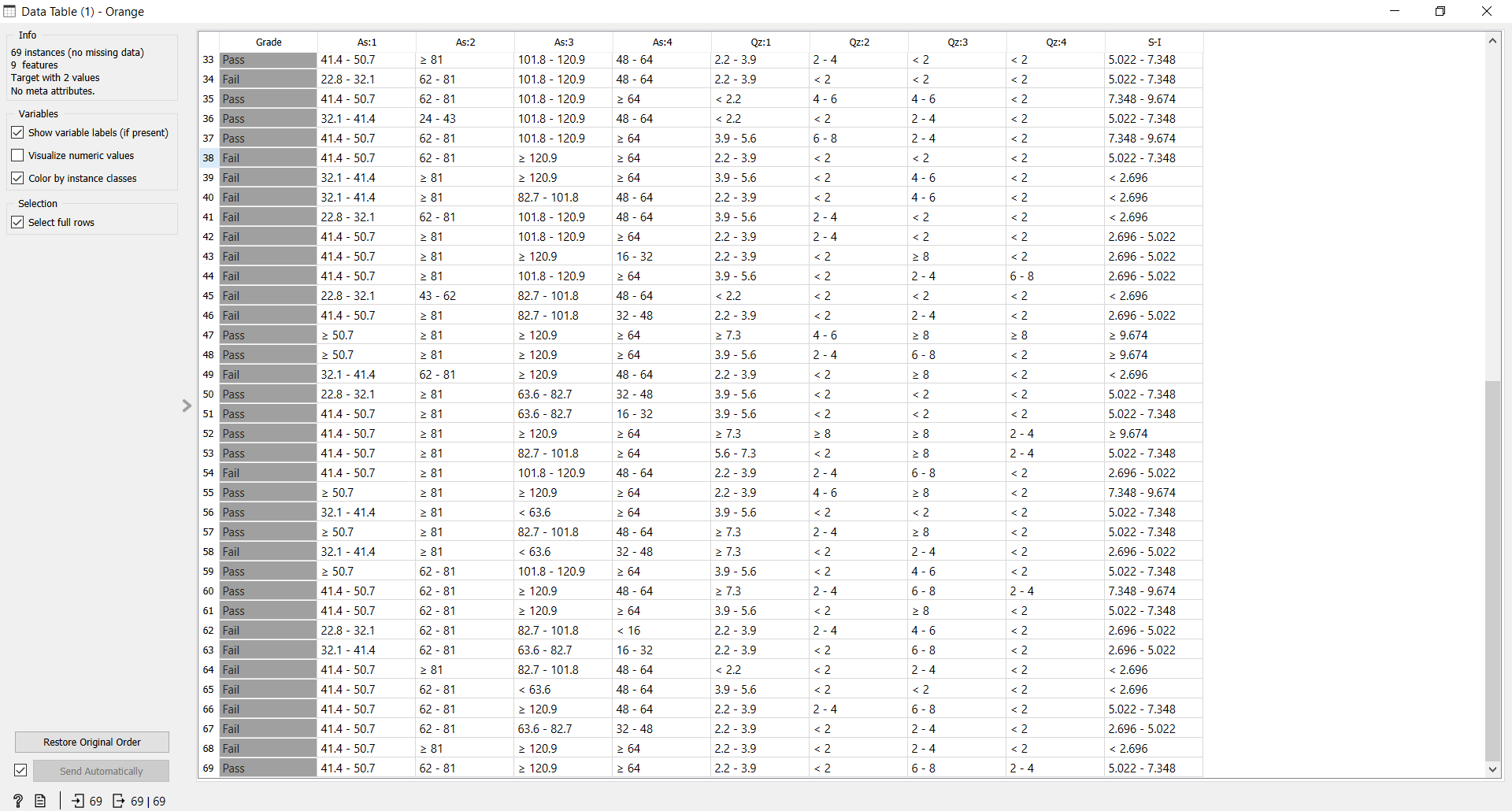
For the sample I just show the grade with S1.

## Preprocessing:



## Data After Preprocessing:





## Boxplot After preprocessing:



## Scatterplot After preprocessing:

