

# DATA ANALYST

## Intenship Task 6

### DESCRIPTION

This task focuses on data visualization using Python to analyze and interpret patterns in a cleaned student performance dataset. Various charts such as bar, line, histogram, and scatter plots are created to identify trends, distributions, and relationships that help in understanding academic performance effectively.

### PREPARED BY

Reema Safrin M  
(23-01-2026)

### MY WORK

In this task, I used a cleaned student performance dataset to create meaningful data visualizations using Python libraries such as pandas and matplotlib. I analyzed average grades across different assessments using a bar chart, examined performance trends through a line chart, and studied grade distribution with a histogram. A scatter plot was used to understand the relationship between student absences and final grades. Each visualization was supported with clear insights to explain observed patterns and trends. This task helped in transforming numerical data into visual insights and improved my ability to interpret and communicate data-driven findings clearly.

### DATASET

[students\\_perfomence\\_dataset](#)

# Task 6: Data Visualization – Python Charts for Insights

**Dataset Used:** Cleaned Student Performance Dataset

**Objective:**  
To visualize student performance data using Python charts and extract meaningful insights.

```
In [1]: from google.colab import files
        uploaded = files.upload()
```

Choose Files

No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving cleaned\_student\_performance.csv to cleaned\_student\_performance.csv

**This dataset contains student academic performance data used for visualization analysis.**

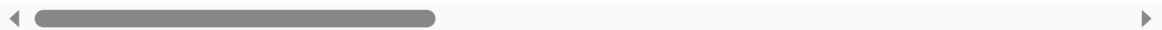
```
In [2]: import pandas as pd
        import matplotlib.pyplot as plt
```

```
In [3]: df = pd.read_csv("cleaned_student_performance.csv")
        df.head()
```

Out[3]:

	school	sex	age	address	family_size	parent_status	mother_education	father_educatio
0	GP	F	18	U	GT3	A	4	
1	GP	F	17	U	GT3	T	1	
2	GP	F	15	U	LE3	T	1	
3	GP	F	15	U	GT3	T	4	
4	GP	F	16	U	GT3	T	3	

5 rows × 35 columns



```
In [4]: df.info()
        df.describe()
```

```

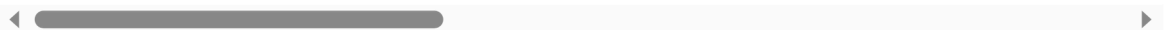
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 649 entries, 0 to 648
Data columns (total 35 columns):
#   Column                                     Non-Null Count  Dtype
---  -
school                                     649 non-null    object
sex                                       649 non-null    object
age                                       649 non-null    int64
address                                   649 non-null    object
family_size                             649 non-null    object
parent_status                           649 non-null    object
mother_education                        649 non-null    int64
father_education                        649 non-null    int64
mother_job                              649 non-null    object
father_job                              649 non-null    object
0  1  2  3  4  5  6                        649 non-null    object
7  8  9  10 reason 11                      649 non-null    object
guardian                                12              649 non-null    int64
traveltime                             13              649 non-null    int64
studytime 14 failures                  649 non-null    int64
15 school_support 16                    649 non-null    object
family_support 17                       649 non-null    object
paid 18                                 649 non-null    object
extra_activities 19                     649 non-null    object
nursery 20                             649 non-null    object
higher_education 21                    649 non-null    object
internet_access 22 romantic_relationship 649 non-null    object
23 family_relationship 649 non-null    int64
24 freetime 25 going_out 649 non-null    int64
26 workday_alcohol 27              649 non-null    int64
weekend_alcohol 28                  649 non-null    int64
health 29 absences 30                649 non-null    int64
grade_1 31 grade_2 32                649 non-null    int64
final_grade 33                      649 non-null    int64
total_score 34                      649 non-null    int64
performance_level 649 non-null    int64
649 non-null    int64
649 non-null    object

dtypes: int64(17), object(18)
memory usage: 177.6+ KB

```

Out[4]:

	age	mother_education	father_education	traveltime	studytime	failures
count	649.000000	649.000000	649.000000	649.000000	649.000000	649.000000
mean	16.744222	2.514638	2.306626	1.568567	1.930663	0.221880
std	1.218138	1.134552	1.099931	0.748660	0.829510	0.593235
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000
25%	16.000000	2.000000	1.000000	1.000000	1.000000	0.000000
50%	17.000000	2.000000	2.000000	1.000000	2.000000	0.000000
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000



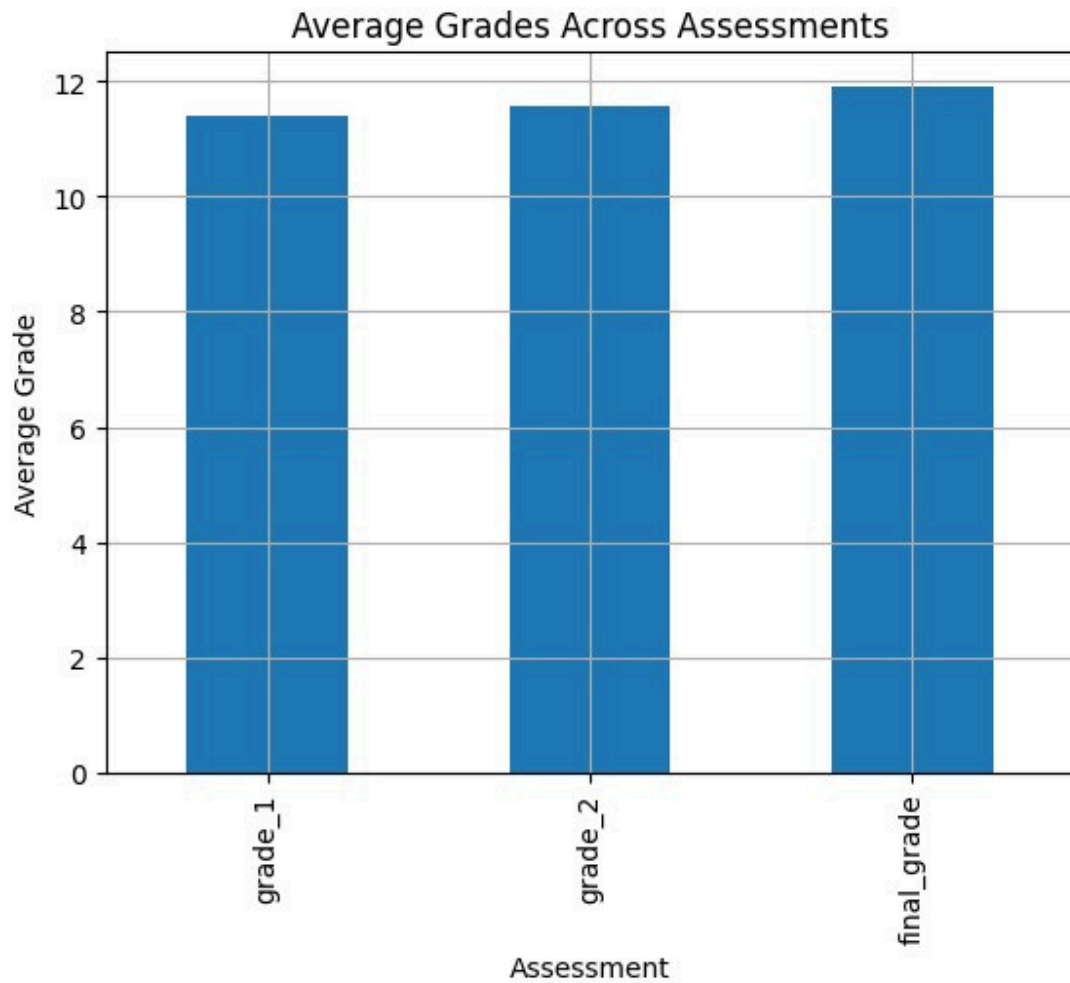
**This dataset contains cleaned student academic performance data, including subject-wise marks.**

In [6]: `df.columns`

Out[6]: Index(['school', 'sex', 'age', 'address', 'family\_size', 'parent\_status', 'mother\_education', 'father\_education', 'mother\_job', 'father\_job', 'reason', 'guardian', 'traveltime', 'studytime', 'failures', 'school\_support', 'family\_support', 'paid', 'extra\_activities', 'nursery', 'higher\_education', 'internet\_access', 'romantic\_relationship', 'family\_relationship', 'freetime', 'going\_out', 'workday\_alcohol', 'weekend\_alcohol', 'health', 'absences', 'grade\_1', 'grade\_2', 'final\_grade', 'total\_score', 'performance\_level'], dtype='object')

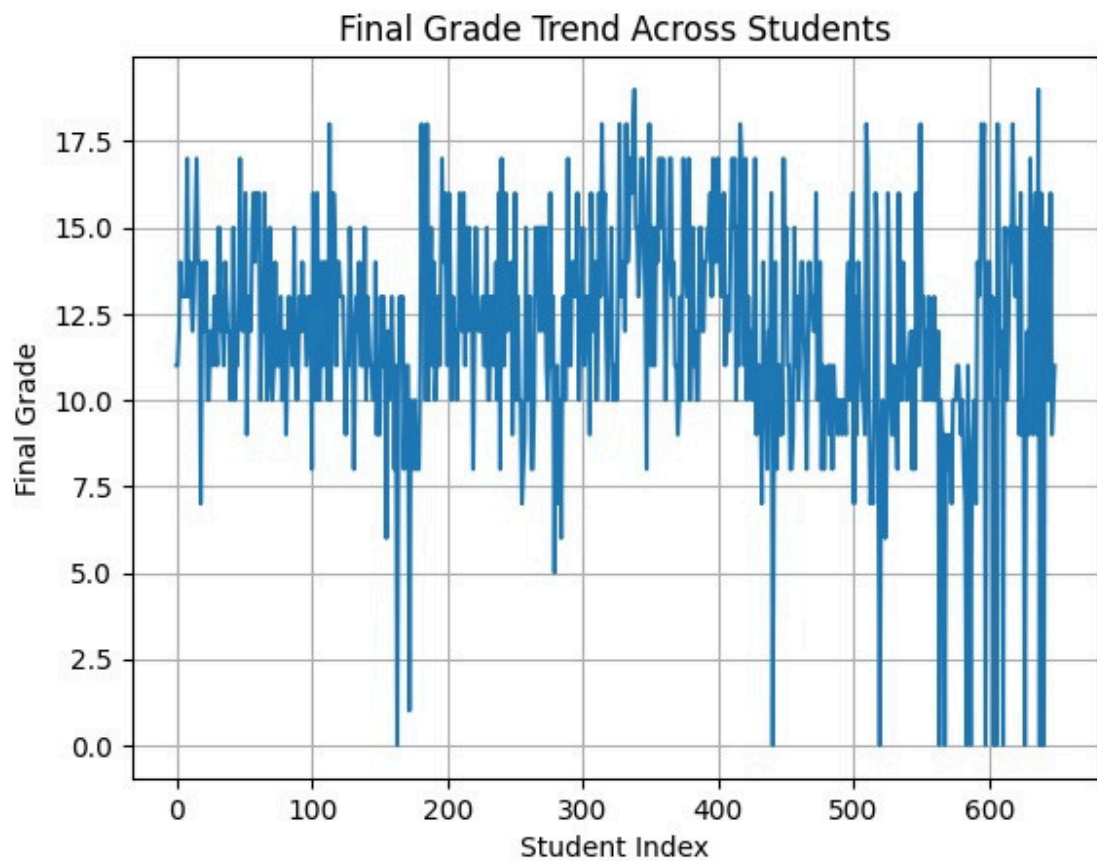
In [7]: `grade_avg = df[['grade_1', 'grade_2', 'final_grade']].mean()`

```
grade_avg.plot(kind='bar')
plt.title("Average Grades Across Assessments")
plt.xlabel("Assessment")
plt.ylabel("Average Grade")
plt.grid(True)
plt.show()
```



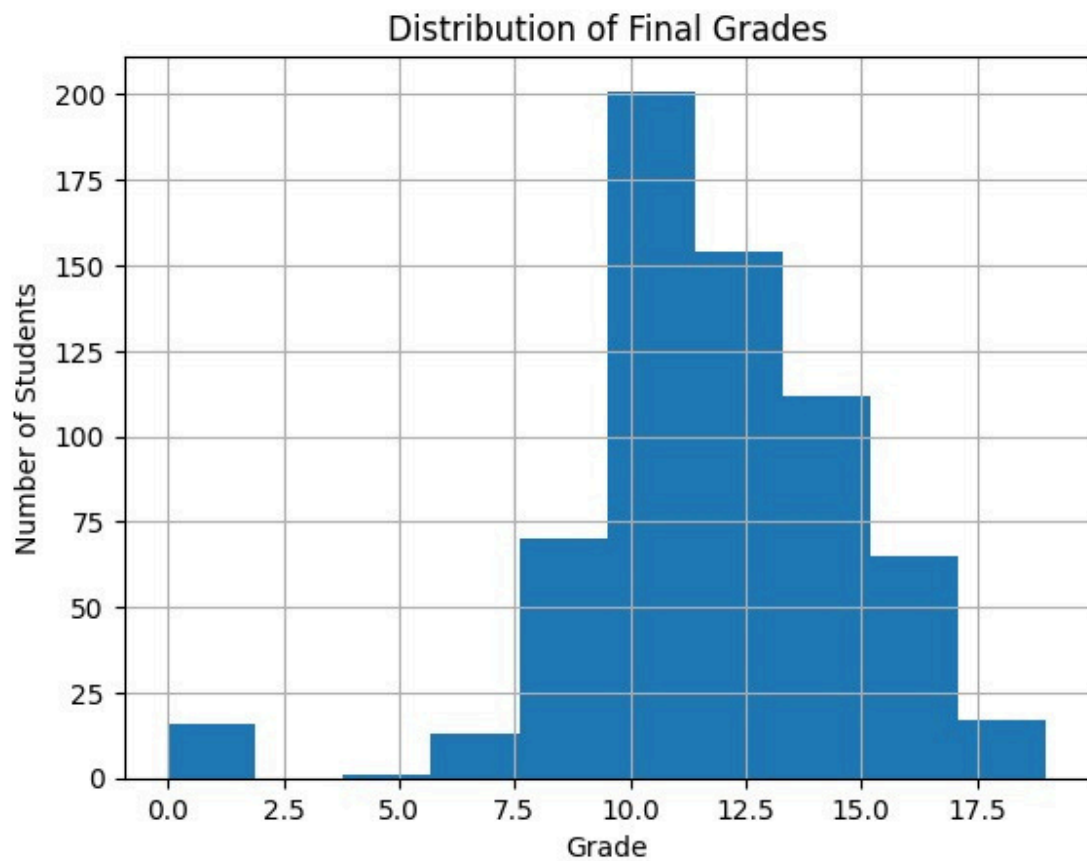
**The bar chart shows that the final grade has the highest average score, indicating improvement in student performance over time.**

```
In [8]: plt.plot(df.index, df['final_grade'])
plt.title("Final Grade Trend Across Students")
plt.xlabel("Student Index")
plt.ylabel("Final Grade")
plt.grid(True)
plt.show()
```



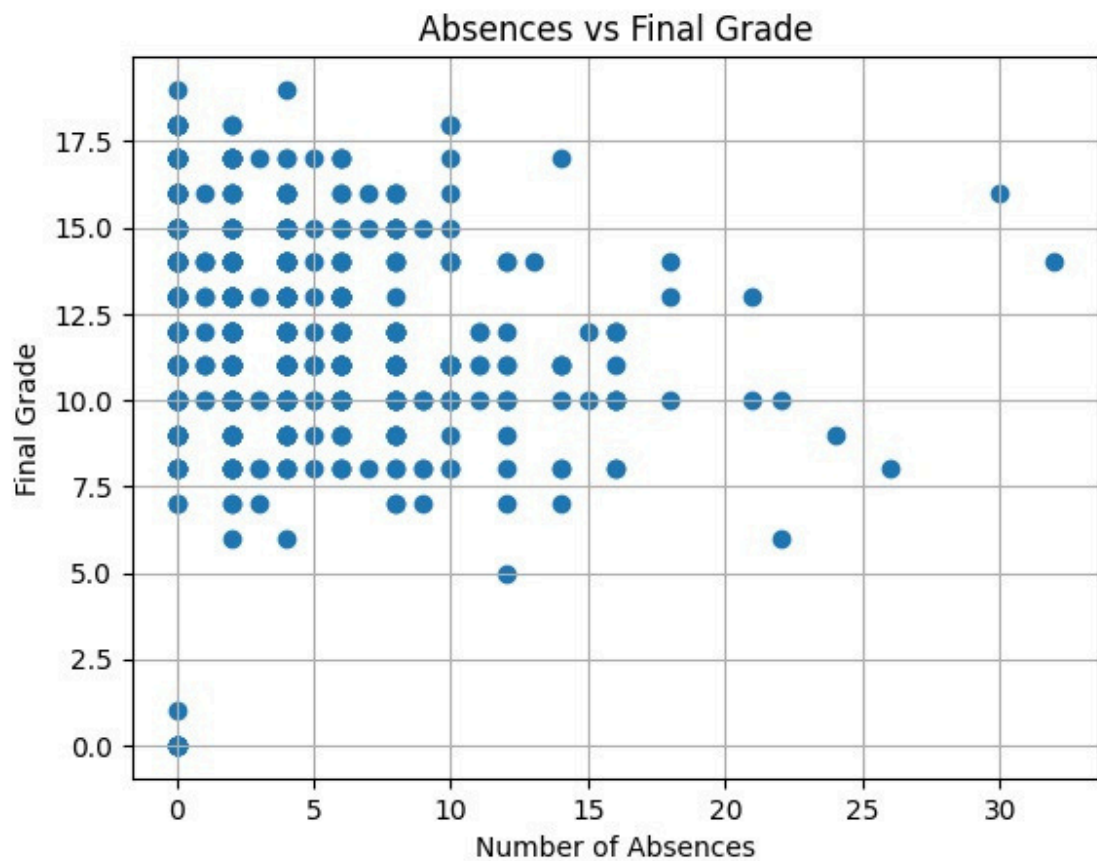
**The line chart shows fluctuations in final grades, reflecting differences in academic performance among students.**

```
In [9]: plt.hist(df['final_grade'], bins=10)
plt.title("Distribution of Final Grades")
plt.xlabel("Grade")
plt.ylabel("Number of Students")
plt.grid(True)
plt.show()
```



**Most students score within the average grade range, while fewer students achieve very high or very low grades.**

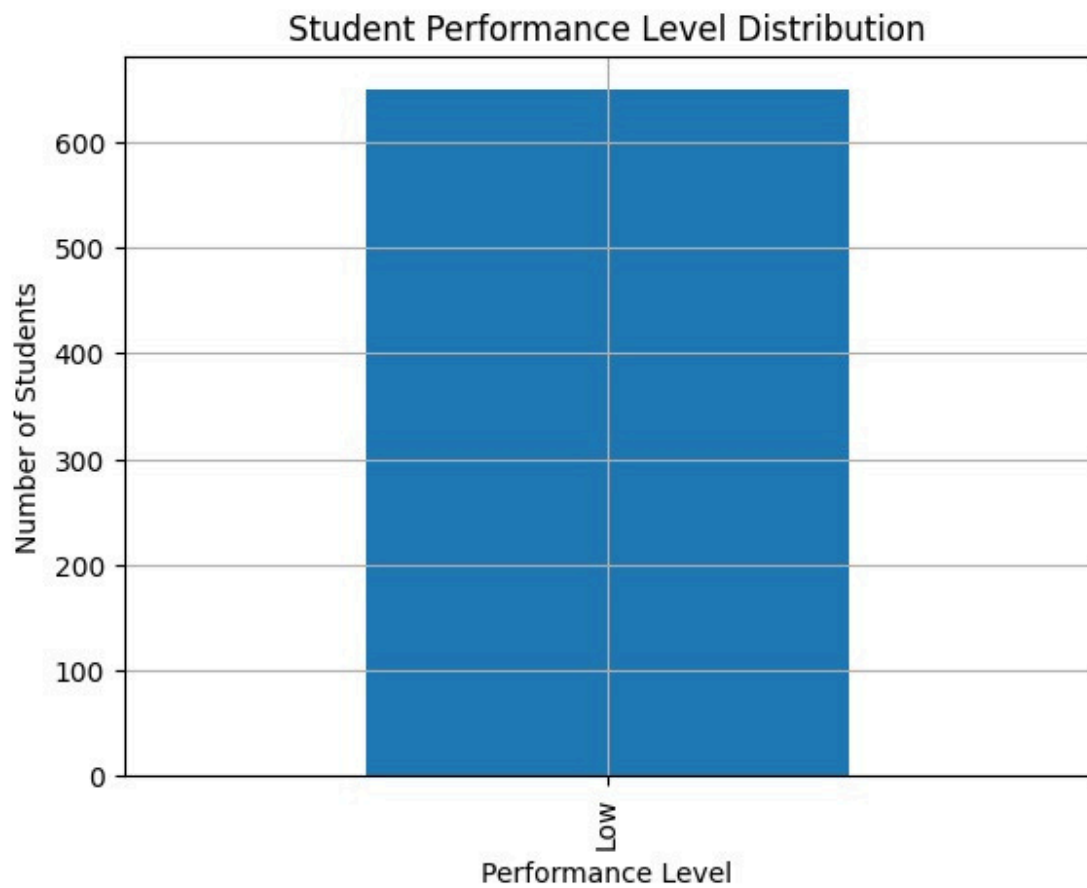
```
In [10]: plt.scatter(df['absences'], df['final_grade'])  
plt.title("Absences vs Final Grade")  
plt.xlabel("Number of Absences")  
plt.ylabel("Final Grade")  
plt.grid(True)  
plt.show()
```



**The scatter plot indicates that students with higher absences tend to have lower final grades, suggesting attendance impacts performance.**

```
In [11]: df['performance_level'].value_counts().plot(kind='bar')
plt.title("Student Performance Level Distribution")
plt.xlabel("Performance Level")
plt.ylabel("Number of Students")
plt.grid(True)
plt.show()
```





## Final Insights

1. Studentsshow improvement from Grade 1 to Final Grade.
2. Final grades are mostly concentrated around average performance levels.
3. Attendance has a noticeable impact on academic results.
4. Visualization helps identify key factors affecting student performance.