

Creating GPA database

```
In [1]: import pandas as pd
import numpy as np
# Remove all unwanted warnings
import warnings
warnings.filterwarnings('ignore')

In [2]: data = {'Course': ['PHYS 205', 'MATH 205', 'MATH 204', 'ENGR 201', 'ECON 201', 'ENGR 233', 'ENGR 213',
                        'COEN 243', 'COEN 212', 'ENGR 301', 'ENCS 282', 'ELEC 273', 'COEN 231', 'ENGR 371',
                        'ENGR 251', 'ENGR 242', 'MIAE 221', 'MIAE 211', 'INDU 211', 'ENGR 311'],
              'Letter Grade': ['A+', 'A', 'A+', 'B-', 'A-', 'C+', 'B', 'B+', 'C', 'A-', 'B-', 'C', 'A+', 'B-', 'D', 'A', 'A+', 'A', 'A+'],
              'Year': [2021, 2021, 2021, 2021, 2021, 2022, 2022, 2022, 2022, 2022, 2022, 2022, 2023, 2023, 2023, 2023, 2023, 2023, 2023],
              'Semester': ['Fall', 'Fall', 'Fall', 'Fall', 'Fall', 'Winter', 'Winter', 'Winter', 'Winter', 'Summer', 'Fall', 'Fall', 'Fall',
                          'Winter', 'Winter', 'Winter', 'Fall', 'Fall', 'Fall', 'Fall']}

In [3]: df = pd.DataFrame(data)

In [4]: # Add a GPA column into the data
def grade_to_gpa(letter_grade):
    if letter_grade == 'A+':
        return 4.3
    elif letter_grade == 'A':
        return 4.0
    elif letter_grade == 'A-':
        return 3.7
    elif letter_grade == 'B+':
        return 3.3
    elif letter_grade == 'B':
        return 3.0
    elif letter_grade == 'B-':
        return 2.7
    elif letter_grade == 'C+':
        return 2.3
    elif letter_grade == 'C':
        return 2.0
    elif letter_grade == 'C-':
        return 1.7
    elif letter_grade == 'D+':
        return 1.3
    elif letter_grade == 'D':
        return 1.0
    elif letter_grade == 'D-':
        return 0.7
    elif letter_grade == 'F':
        return 0
    else:
        return None

In [5]: semester_order = ['Winter', 'Summer', 'Fall']

In [6]: # Applying the function to create the GPA column
df['GPA'] = df['Letter Grade'].apply(grade_to_gpa)
df
```

Out[6]:

	Course	Letter Grade	Year	Semester	GPA
0	PHYS 205	A+	2021	Fall	4.3
1	MATH 205	A	2021	Fall	4.0
2	MATH 204	A+	2021	Fall	4.3
3	ENGR 201	B-	2021	Fall	2.7
4	ECON 201	A-	2021	Fall	3.7
5	ENGR 233	C+	2022	Winter	2.3
6	ENGR 213	B	2022	Winter	3.0
7	COEN 243	B+	2022	Winter	3.3
8	COEN 212	C	2022	Winter	2.0
9	ENGR 301	C+	2022	Summer	2.3
10	ENCS 282	A-	2022	Fall	3.7
11	ELEC 273	D-	2022	Fall	0.7
12	COEN 231	C	2022	Fall	2.0
13	ENGR 371	A-	2023	Winter	3.7
14	ENGR 251	B-	2023	Winter	2.7
15	ENGR 242	D	2023	Winter	1.0
16	MIAE 221	A	2023	Fall	4.0
17	MIAE 211	A+	2023	Fall	4.3
18	INDU 211	A	2023	Fall	4.0
19	ENGR 311	A+	2023	Fall	4.3

```
In [7]: # Get my Cumulative GPA
CGPA = df['GPA'].mean()
print("CGPA as of Winter 2024 is:", round(CGPA, 2))

CGPA as of Winter 2024 is: 3.12
```

```
In [8]: # Add courses
winter2024_courses = {'Course': ['MIAE 313', 'INDU 323', 'ENGR 245', 'ACCO 220'],
                    'Letter Grade': ['A+', 'A+', 'A+', 'A+'],
                    'Year': [2024, 2024, 2024, 2024],
                    'Semester': ['Winter', 'Winter', 'Winter', 'Winter']}

new_df = pd.DataFrame(winter2024_courses)
new_df['GPA'] = new_df['Letter Grade'].apply(grade_to_gpa)

# Add new courses to the data frame using concat
df = pd.concat([df, new_df], ignore_index = True)
df
```

Out[8]:

	Course	Letter Grade	Year	Semester	GPA
0	PHYS 205	A+	2021	Fall	4.3
1	MATH 205	A	2021	Fall	4.0
2	MATH 204	A+	2021	Fall	4.3
3	ENGR 201	B-	2021	Fall	2.7
4	ECON 201	A-	2021	Fall	3.7
5	ENGR 233	C+	2022	Winter	2.3
6	ENGR 213	B	2022	Winter	3.0
7	COEN 243	B+	2022	Winter	3.3
8	COEN 212	C	2022	Winter	2.0
9	ENGR 301	C+	2022	Summer	2.3
10	ENCS 282	A-	2022	Fall	3.7
11	ELEC 273	D-	2022	Fall	0.7
12	COEN 231	C	2022	Fall	2.0
13	ENGR 371	A-	2023	Winter	3.7
14	ENGR 251	B-	2023	Winter	2.7
15	ENGR 242	D	2023	Winter	1.0
16	MIAE 221	A	2023	Fall	4.0
17	MIAE 211	A+	2023	Fall	4.3
18	INDU 211	A	2023	Fall	4.0
19	ENGR 311	A+	2023	Fall	4.3
20	MIAE 313	A+	2024	Winter	4.3
21	INDU 323	A+	2024	Winter	4.3
22	ENGR 245	A+	2024	Winter	4.3
23	ACCO 220	A+	2024	Winter	4.3

```
In [9]: # Getting new CGPA after winter 2024
new_cgpa = df['GPA'].mean()
print("CGPA after Winter 2024 is:", round(new_cgpa, 2))

CGPA after Winter 2024 is: 3.31
```

```
In [10]: # Getting annual GPA for year 2023-2024
year2023_2024 = df['GPA'].tail(8)
print("Annual GPA for year 2023-2024 is:", round(year2023_2024.mean(), 2))

Annual GPA for year 2023-2024 is: 4.22
```

```
In [11]: # Sort DataFrame by custom order of semesters and year
df['Semester'] = pd.Categorical(df['Semester'], categories = semester_order, ordered = True)
df = df.sort_values(by = ['Year', 'Semester'])

# Group courses by year and semester and calculate annual GPA
semester_gpa = df.groupby(['Year', 'Semester'])['GPA'].mean()
semester_gpa = round(semester_gpa, 2)

# Convert the series to DataFrame
semester_gpa_df = semester_gpa.reset_index()
semester_gpa_df.columns = ['Year', 'Semester', 'GPA']

# Remove null values
semester_gpa_df.dropna(axis = 0, inplace = True)
semester_gpa_df
```

Out[11]:

	Year	Semester	GPA
2	2021	Fall	3.80
3	2022	Winter	2.65
4	2022	Summer	2.30
5	2022	Fall	2.13
6	2023	Winter	2.47
8	2023	Fall	4.15
9	2024	Winter	4.30

Getting Data using SQL

```
In [12]: import sqlite3

In [13]: collect = sqlite3.connect('grades.db')

In [14]: df.to_sql('grades', collect)

Out[14]: 24

In [15]: %load_ext sql

In [16]: %sql sqlite:///grades.db

• Getting FALL 2023 GPA

In [17]: %sql
SELECT round(avg(gpa), 2)
FROM grades
WHERE semester = 'Fall' AND year = 2023
* sqlite:///grades.db
Done.

Out[17]: round(avg(gpa), 2)
4.15

• Getting FALL 2023 and Winter 2024 GPA

In [18]: %sql
SELECT year, round(AVG(gpa), 2) as annual_gpa
FROM grades
WHERE (year = 2023 AND semester = 'Fall') OR (year = 2024 AND semester = 'Winter')
* sqlite:///grades.db
Done.

Out[18]: Year annual_gpa
2023 4.23
```

Calculating percentage increase between Winter 2023 and Fall 2023

```
In [19]: %sql
SELECT year, round(AVG(gpa), 2) as Winter
FROM grades
WHERE year = 2023 AND semester = 'Winter'
* sqlite:///grades.db
Done.

Out[19]: Year Winter
2023 2.47

In [20]: winter_2023_gpa = 2.47

In [21]: %sql
SELECT year, round(AVG(gpa), 2) as Fall
FROM grades
WHERE year = 2023 AND semester = 'Fall'
* sqlite:///grades.db
Done.

Out[21]: Year Fall
2023 4.15

In [22]: fall_2023_gpa = 4.15

In [23]: # Finding increase in percentage
percentage_increase_2023 = ((fall_2023_gpa - winter_2023_gpa) / winter_2023_gpa) * 100
print("My performance between Winter 2023 and Fall 2023 has increased by:", round(percentange_increase_2023, 2), "%")

My performance between Winter 2023 and Fall 2023 has increased by: 68.02 %

In [24]: %sql
SELECT year, round(AVG(gpa), 2)
FROM grades
WHERE (year = 2023 AND semester = 'Winter') OR (year = 2023 AND semester = 'Fall')
* sqlite:///grades.db
Done.
```

Out[24]:

Year	round(AVG(gpa), 2)
2023	3.43