Домашнее задание 1 (5 баллов).

Все задания ниже имеют равный вес (5/16).

In [100]:
```
import pandas as pd
```

Описание данных

В папке Data находится информация о студентах. Всего 10 групп студентов. Файлы делятся на две категории:

```
* Students_info_i - информация о студентах из группы i
* Students_marks_i - оценки студентов из группы i за экзамены
```

Одно из важных достоинств pandas — это удобные методы реляционного взаимодействия с данными, аналогичные, например, возможностям SQL для слияния и конкатенации таблиц: merge, join, concat. Наличие готовых методов позволяет не реализовывать самостоятельно поэлементную обработку данных и оперировать сразу целыми таблицами данных.

Подробно об этих методах посмотрите тут: https://www.kaggle.com/residentmario/renaming-and-combining#Combining (https://www.kaggle.com/residentmario/renaming-and-combining#Combining)

Задание 1. Соберите всю информацию о студентах в одну таблицу df. В получившейся таблице должна быть информация и оценки всех студентов из всех групп. Напечатайте несколько строк таблицы для демонстрации результата.¶

In [101]:
```
df_info = pd.DataFrame()
df_marks = pd.DataFrame()
for i in range(10):
    df_info = pd.concat([df_info, pd.read_csv(f"Data/Students_info_{i}.csv")])
    df_marks = pd.concat([df_marks, pd.read_csv(f"Data/Students_marks_{i}.csv"
)])
df = pd.merge(df_info, df_marks, on='index', how='outer') #объединим все, но п
роверим, что действительно все объединилось ->

df[df.isna().any(axis=1)] #пустой df, значит, все успешно
```

Out[101]:

index	gender	race/ethnicity	parental level of education	lunch	test preparation course	group	math score	reading score	writing score

In [102]:
```
df.head()
```

Out[102]:

	index	gender	race/ethnicity	parental level of education	lunch	test preparation course	group	math score	reading score	wr s
0	0	female	group B	bachelor's degree	standard	none	group1	72	72	
1	1	female	group C	some college	standard	completed	group1	69	90	
2	2	female	group B	master's degree	standard	none	group1	90	95	
3	3	male	group A	associate's degree	free/reduced	none	group1	47	57	
4	4	male	group C	some college	standard	none	group1	76	78	

Задание 2. Удалите столбец index у полученной таблицы. Напечатайте первые 10 строк таблицы.

In [103]:
```
df.drop('index', axis=1, inplace = True)
df.head(10)
```

Out[103]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	group	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	group1	72	72	74
1	female	group C	some college	standard	completed	group1	69	90	88
2	female	group B	master's degree	standard	none	group1	90	95	93
3	male	group A	associate's degree	free/reduced	none	group1	47	57	44
4	male	group C	some college	standard	none	group1	76	78	75
5	female	group B	associate's degree	standard	none	group1	71	83	78
6	female	group B	some college	standard	completed	group1	88	95	92
7	male	group B	some college	free/reduced	none	group1	40	43	39
8	male	group D	high school	free/reduced	completed	group1	64	64	67
9	female	group B	high school	free/reduced	none	group1	38	60	50

Задание 3. Выведите на экран размеры полученной таблицы

In [104]:
```
df.shape
```

Out[104]: (1000, 9)

Задание 4. Выведите на экран статистические характеристики числовых столбцов таблицы (минимум, максимум, среднее значение, стандартное отклонение)

In [105]:
```
df.describe()
```

Out[105]:

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

Задание 5. Проверьте, есть ли в таблице пропущенные значения

In [106]:
```
df[df.isna().any(axis=1)] #пустой df, значит, пропущенных значений нет
```

Out[106]:

gender	race/ethnicity	parental level of education	lunch	test preparation course	group	math score	reading score	writing score

Задание 6. Выведите на экран средние баллы студентов по каждому предмету (math, reading, writing)

In [107]:
```
def print_mean_scores(df):
    print('Math mean score: ', df['math score'].mean())
    print('Reading mean score: ', df['reading score'].mean())
    print('Writing mean score: ', df['writing score'].mean())

print_mean_scores(df)
```

```
Math mean score:  66.089
Reading mean score:  69.169
Writing mean score:  68.054
```

Задание 7. Как зависят оценки от того, проходил ли студент курс для подготовки к сдаче экзамена (test preparation course)? Выведите на экран для каждого предмета в отдельности средний балл студентов, проходивших курс для подготовки к экзамену и не проходивших курс.

In [108]:
```
df_prepared_students = df[df['test preparation course'] == 'completed']
df_unprepared_students = df[df['test preparation course'] == 'none']

print('RESULTS FOR PREPARED STUDENTS')
print()
print_mean_scores(df_prepared_students)
print('_____________________________')
print()
print('RESULTS FOR UNPREPARED STUDENTS')
print()
print_mean_scores(df_unprepared_students)
```

```
RESULTS FOR PREPARED STUDENTS

Math mean score:  69.69553072625699
Reading mean score:  73.89385474860335
Writing mean score:  74.41899441340782
_____________________________

RESULTS FOR UNPREPARED STUDENTS

Math mean score:  64.0778816199377
Reading mean score:  66.53426791277259
Writing mean score:  64.50467289719626
```

Задание 8. Выведите на экран все различные значения из столбца lunch.

In [109]:
```
set(df['lunch'])
```

Out[109]:
```
{'free/reduced', 'standard'}
```

Задание 9. Переименуйте колонку "parental level of education" в "education", а "test preparation course" в "test preparation" с помощью метода pandas rename https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rename.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rename.html)

In [110]:
```
df.rename(columns={"parental level of education": "education",
                   "test preparation course": "test preparation"}, inplace = True)
df.head()
```

Out[110]:

	gender	race/ethnicity	education	lunch	test preparation	group	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	group1	72	72	74
1	female	group C	some college	standard	completed	group1	69	90	88
2	female	group B	master's degree	standard	none	group1	90	95	93
3	male	group A	associate's degree	free/reduced	none	group1	47	57	44
4	male	group C	some college	standard	none	group1	76	78	75

Зафиксируем минимальный балл для сдачи экзамена

In [111]:
```
passmark = 50
```

Задание 10. Ответьте на вопросы:

* Какая доля студентов сдала экзамен по математике (passmark >= 50)?
* Какая доля студентов, проходивших курс подготовки к экзамену, сдала экзамен по математике?
* Какая доля женщин, не проходивших курс подготовки к экзамену, не сдала экзамен по математике?

In [112]:
```
def proportion_of_passed_students(df, passmark):
    passed_num = df[df['math score']>=50].shape[0]
    total_num = df.shape[0]
    return (passed_num / total_num)
```

In [113]:
```
df_prepared = df[df['test preparation'] == 'completed']
df_unprepared_female = df[(df['test preparation'] == 'none')
                          & (df['gender'] == 'female')]
```

In [114]:
```
print("PROPORTION OF PASSED STUDENTS")
print(proportion_of_passed_students(df, passmark))
print()
print('_____________________________')
print()
print("PROPORTION OF PASSED PREPARED STUDENTS")
print(proportion_of_passed_students(df_prepared, passmark))
print()
print('_____________________________')
print()
print("PROPORTION OF UNPREPARED FEMALES WHO DID NOT PASS")
print(1 - proportion_of_passed_students(df_unprepared_female, passmark))
```

```
PROPORTION OF PASSED STUDENTS
0.865

_____________________________

PROPORTION OF PASSED PREPARED STUDENTS
0.9217877094972067

_____________________________

PROPORTION OF UNPREPARED FEMALES WHO DID NOT PASS
0.20958083832335328
```

Задание 11. С помощью groupby выполните задания ниже. Также выведите время выполнения каждого из заданий.

* Для каждой этнической группы выведите средний балл за экзамен по чтению
* Для каждого уровня образования выведите минимальный балл за экзамен по письму

In [115]:
```
%%time
df[['race/ethnicity', 'reading score']].groupby(['race/ethnicity']).mean()
```

```
Wall time: 6 ms
```

Out[115]:

race/ethnicity	reading score
group A	64.674157
group B	67.352632
group C	69.103448
group D	70.030534
group E	73.028571

In [116]:
```
%%time
df[['education', 'writing score']].groupby(['education']).min()
```

```
Wall time: 3.99 ms
```

Out[116]:

education	writing score
associate's degree	35
bachelor's degree	38
high school	15
master's degree	46
some college	19
some high school	10

Задание 12. Выполните задание 11 с помощью циклов. Сравните время выполнения.

In [117]:
```
import collections
```

In [118]:
```
%%time

#Выполним поиск средней оценки по чтению для каждой расы с помощью циклов.
#Было интересно отсортировать ключи словаря, чтобы максимально приблизиться к
 результату из пункта 11.
#Время исполнения этой ячейки примерно в 26.5 раз выше (конечно, оно меняетс
я), чем время исполнения аналогичной операции с использование groupby
race_mean = {}
for index, row in df.iterrows():
    race = row['race/ethnicity']
    race_mean.setdefault(race, [0, 0])
    race_mean[race][0] += 1
    race_mean[race][1] += row['reading score']

race_mean = collections.OrderedDict(sorted(race_mean.items()))
for key in race_mean.keys():
    print(key, 'MEAN SCORE: ', race_mean[key][1]/race_mean[key][0])
```

```
group A MEAN SCORE:  64.67415730337079
group B MEAN SCORE:  67.35263157894737
group C MEAN SCORE:  69.10344827586206
group D MEAN SCORE:  70.03053435114504
group E MEAN SCORE:  73.02857142857142
Wall time: 185 ms
```

In [119]:
```
%%time

#Здесь время исполнения тоже существенно выше: примерно в 32 раза

min_education = {}
for index, row in df.iterrows():
    education = row['education']
    min_education.setdefault(education, 101) #Так как мы знаем максимальный ба
лл, то можем установить такое значение
    min_education[education] = min(min_education[education], row['writing scor
e'])

min_education = collections.OrderedDict(sorted(min_education.items()))
for key in min_education.keys():
    print(key, 'MIN SCORE: ', min_education[key])
```

```
associate's degree MIN SCORE:  35
bachelor's degree MIN SCORE:  38
high school MIN SCORE:  15
master's degree MIN SCORE:  46
some college MIN SCORE:  19
some high school MIN SCORE:  10
Wall time: 168 ms
```

Задание 13. Выведите на экран средние баллы студентов по каждому предмету в зависимости от пола и уровня образования. То есть должно получиться количество групп, равных 2 * (число уровней образования), и для каждой такой группы выыведите средний балл по каждому из предметов.

Это можно сделать с помощью сводных таблиц (pivot_table):

https://www.kaggle.com/kamilpolak/tutorial-how-to-use-pivot-table-in-pandas (https://www.kaggle.com/kamilpolak/tutorial-how-to-use-pivot-table-in-pandas)

In [120]:
```
mean_score = pd.pivot_table(df,index=['gender','education']) #Получилось 12 гр
упп. Действительно, у нас 6 уровней образования, как видно из пункта 11
mean_score
```

Out[120]:

gender	education	math score	reading score	writing score
female	associate's degree	65.250000	74.120690	74.000000
	bachelor's degree	68.349206	77.285714	78.380952
	high school	59.351064	68.202128	66.691489
	master's degree	66.500000	76.805556	77.638889
	some college	65.406780	73.550847	74.050847
	some high school	59.296703	69.109890	68.285714
male	associate's degree	70.764151	67.433962	65.405660
	bachelor's degree	70.581818	68.090909	67.654545
	high school	64.705882	61.480392	58.539216
	master's degree	74.826087	73.130435	72.608696
	some college	69.009259	64.990741	63.148148
	some high school	67.840909	64.693182	61.375000

Задание 14. Сколько студентов успешно сдали экзамен по математике?

Создайте новый столбец в таблице df под названием Math_PassStatus и запишите в него F, если студент не сдал экзамен по математике (балл за экзамен < passmark), и P иначе.

Посчитайте количество студентов, сдавших и не сдавших экзамен по математике.

Сделайте аналогичные шаги для экзаменов по чтению и письму.

In [121]:
```
df['Math_PassStatus'] = df.apply(lambda row: 'P' if row['math score'] >= 50 el
se 'F', axis=1)
df['Reading_PassStatus'] = df.apply(lambda row: 'P' if row['reading score'] >=
50 else 'F', axis=1)
df['Writing_PassStatus'] = df.apply(lambda row: 'P' if row['writing score'] >=
50 else 'F', axis=1)
df
```

Out[121]:

	gender	race/ethnicity	education	lunch	test preparation	group	math score	reading score	writing scor
0	female	group B	bachelor's degree	standard	none	group1	72	72	7
1	female	group C	some college	standard	completed	group1	69	90	8
2	female	group B	master's degree	standard	none	group1	90	95	9
3	male	group A	associate's degree	free/reduced	none	group1	47	57	4
4	male	group C	some college	standard	none	group1	76	78	7
...	...	...	...	...	...	...	...	...	.
995	female	group E	master's degree	standard	completed	group10	88	99	9
996	male	group C	high school	free/reduced	none	group10	62	55	5
997	female	group C	high school	free/reduced	completed	group10	59	71	6
998	female	group D	some college	standard	completed	group10	68	78	7
999	female	group D	some college	free/reduced	none	group10	77	86	8

1000 rows × 12 columns

In [122]:
```
df['Math_PassStatus'].value_counts()
```

Out[122]:
```
P    865
F    135
Name: Math_PassStatus, dtype: int64
```

In [123]:
```
df['Reading_PassStatus'].value_counts()
```

Out[123]:
```
P    910
F     90
Name: Reading_PassStatus, dtype: int64
```

In [124]:
```
df['Writing_PassStatus'].value_counts()
```

Out[124]:
```
P    886
F    114
Name: Writing_PassStatus, dtype: int64
```

Задание 15. Сколько студентов успешно сдали все экзамены?

Создайте столбец OverAll_PassStatus и запишите в него для каждого студента 'F', если студент не сдал хотя бы один из трех экзаменов, а иначе 'P'.

Посчитайте количество студентов, которые сдали все экзамены.

In [125]:
```
df['OverAll_PassStatus'] = df.apply(lambda row: 'F' if ((row['Math_PassStatus'
] == 'F') or
                                                        (row['Reading_PassStat
us'] == 'F') or
                                                        (row['Writing_PassStat
us'] == 'F'))
                                    else 'P', axis=1)
```

In [126]:
```
df
```

Out[126]:

	gender	race/ethnicity	education	lunch	test preparation	group	math score	reading score	writing scor
0	female	group B	bachelor's degree	standard	none	group1	72	72	7
1	female	group C	some college	standard	completed	group1	69	90	8
2	female	group B	master's degree	standard	none	group1	90	95	9
3	male	group A	associate's degree	free/reduced	none	group1	47	57	4
4	male	group C	some college	standard	none	group1	76	78	7
...	...	...	...	...	...	...	...	...	.
995	female	group E	master's degree	standard	completed	group10	88	99	9
996	male	group C	high school	free/reduced	none	group10	62	55	5
997	female	group C	high school	free/reduced	completed	group10	59	71	6
998	female	group D	some college	standard	completed	group10	68	78	7
999	female	group D	some college	free/reduced	none	group10	77	86	8

1000 rows × 13 columns

In [127]:
```
df['OverAll_PassStatus'].value_counts() #Сдали все экзамены 812 человек
```

Out[127]:
```
P    812
F    188
Name: OverAll_PassStatus, dtype: int64
```

Задание 16. Переведем баллы в оценки

Система перевода баллов в оценки

больше 90 = A

80-90 = B

70-80 = C

60-70 = D

50-60 = E

меньше 50 = F (Fail)

Создайте вспомогательную функцию, которая будет по среднему баллу за три экзамена выставлять оценку студенту по данным выше критериям.

Создайте столбец Grade и запишите в него оценку каждого студента.

Выведите количество студентов, получивших каждую из оценок.

In [128]:
```
def GetGrade(average_mark): #К сжалению, в системе перевода нет указания на ст
рогое-нестрогое равенство, поэтому вариант такой
    result = ""
    if average_mark < 50:
        result = "F"
    elif average_mark >= 50 and average_mark < 60:
        result = "E"
    elif average_mark >= 60 and average_mark < 70:
        result = "D"
    elif average_mark >= 70 and average_mark < 80:
        result = "C"
    elif average_mark >= 80 and average_mark <= 90:
        result = "B"
    else:
        result = "A"
    return result

df['Grade'] = (df['math score'] + df['writing score'] + df['reading score']) /
3
df['Grade'] = df['Grade'].apply(GetGrade)
df
```

Out[128]:

	gender	race/ethnicity	education	lunch	test preparation	group	math score	reading score	writing scor
0	female	group B	bachelor's degree	standard	none	group1	72	72	7
1	female	group C	some college	standard	completed	group1	69	90	8
2	female	group B	master's degree	standard	none	group1	90	95	9
3	male	group A	associate's degree	free/reduced	none	group1	47	57	4
4	male	group C	some college	standard	none	group1	76	78	7
...	...	...	...	...	...	...	...	...	.
995	female	group E	master's degree	standard	completed	group10	88	99	9
996	male	group C	high school	free/reduced	none	group10	62	55	5
997	female	group C	high school	free/reduced	completed	group10	59	71	6
998	female	group D	some college	standard	completed	group10	68	78	7
999	female	group D	some college	free/reduced	none	group10	77	86	8

1000 rows × 14 columns

Спасибо за проверку!