Primeira Avaliação Parcial: Avaliação em Equipe

Diamonds Dataset

Neste link (https://ggplot2.tidyverse.org/reference/diamonds.html) você encontra os detalhes sobre um dos datasets disponíveis na biblioteca **seaborn**, que já foi utilizada durante as aulas. O código a seguir mostra como carregar o dataset e as características do mesmo.

In [1]:
```
import seaborn as sns
print(sns.__version__)
```

```
Intel MKL WARNING: Support of Intel(R) Streaming SIMD Extensions
4.2 (Intel(R) SSE4.2) enabled only processors has been deprecated.
Intel oneAPI Math Kernel Library 2025.0 will require Intel(R) Adva
nced Vector Extensions (Intel(R) AVX) instructions.
Intel MKL WARNING: Support of Intel(R) Streaming SIMD Extensions
4.2 (Intel(R) SSE4.2) enabled only processors has been deprecated.
Intel oneAPI Math Kernel Library 2025.0 will require Intel(R) Adva
nced Vector Extensions (Intel(R) AVX) instructions.
0.13.1
```

In [2]:
```
diamonds = sns.load_dataset('diamonds')
diamonds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53940 entries, 0 to 53939
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   carat    53940 non-null  float64
 1   cut      53940 non-null  category
 2   color    53940 non-null  category
 3   clarity  53940 non-null  category
 4   depth    53940 non-null  float64
 5   table    53940 non-null  float64
 6   price    53940 non-null  int64
 7   x        53940 non-null  float64
 8   y        53940 non-null  float64
 9   z        53940 non-null  float64
dtypes: category(3), float64(6), int64(1)
memory usage: 3.0 MB
```

In [3]:
```
print(diamonds.head())
```

```
   carat      cut color clarity  depth  table  price     x     y
z
0   0.23    Ideal     E     SI2   61.5   55.0    326  3.95  3.98
2.43
1   0.21  Premium     E     SI1   59.8   61.0    326  3.89  3.84
2.31
2   0.23     Good     E     VS1   56.9   65.0    327  4.05  4.07
2.31
3   0.29  Premium     I     VS2   62.4   58.0    334  4.20  4.23
2.63
4   0.31     Good     J     SI2   63.3   58.0    335  4.34  4.35
2.75
```

Exercício 1

Com base nas dimensões de cada diamante se pode calcular o volume aproximado, considerando que eles como se fossem caixas. Adicione então uma nova coluna que mostre o preço por milímetro cubico de cada diamante.

In [ ]:
```
#Fazer aqui o exercício 1
```

Exercício 2

Utilizando os recursos do **Pandas** mostre em um **Series** o preço médio dos diamantes para cada tipo de corte corte.

In [ ]:
```
# Fazer aqui o exercício 2
```

No seguinte link [High School Student Performance & Demographics (https://www.kaggle.com/datasets/dillonmyrick/high-school-student-performance-and-demographics)](https://www.kaggle.com/datasets/dillonmyrick/high-school-student-performance-and-demographics), podem ser encontrados dois datasets sobre desempenho de estudantes de ensino médio.

In [4]:
```
import pandas as pd

# impotando datasets
math = pd.read_csv('datasets/datasets/student_math_clean.csv')
port = pd.read_csv('datasets/datasets/student_portuguese_clean.csv'
```

In [5]:
```
# dados de matemática
math.info()
math.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 34 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   student_id             395 non-null    int64 
 1   school                 395 non-null    object
 2   sex                    395 non-null    object
 3   age                    395 non-null    int64 
 4   address_type           395 non-null    object
 5   family_size            395 non-null    object
 6   parent_status          395 non-null    object
 7   mother_education       395 non-null    object
 8   father_education       395 non-null    object
 9   mother_job             395 non-null    object
 10  father_job             395 non-null    object
 11  school_choice_reason   395 non-null    object
 12  guardian               395 non-null    object
 13  travel_time            395 non-null    object
 14  study_time             395 non-null    object
 15  class_failures         395 non-null    int64 
 16  school_support         395 non-null    object
 17  family_support         395 non-null    object
 18  extra_paid_classes     395 non-null    object
 19  activities             395 non-null    object
 20  nursery_school         395 non-null    object
 21  higher_ed              395 non-null    object
 22  internet_access        395 non-null    object
 23  romantic_relationship  395 non-null    object
 24  family_relationship    395 non-null    int64 
 25  free_time              395 non-null    int64 
 26  social                 395 non-null    int64 
 27  weekday_alcohol        395 non-null    int64 
 28  weekend_alcohol        395 non-null    int64 
 29  health                 395 non-null    int64 
 30  absences               395 non-null    int64 
 31  grade_1                395 non-null    int64 
 32  grade_2                395 non-null    int64 
 33  final_grade            395 non-null    int64 
dtypes: int64(13), object(21)
memory usage: 105.0+ KB
```

Out[5]:

	student_id	school	sex	age	address_type	family_size	parent_status	mother_education
0	1	GP	F	18	Urban	Greater than 3	Apart	higher education
1	2	GP	F	17	Urban	Greater than 3	Living together	primary education (4th grade)
2	3	GP	F	15	Urban	Less than or equal to 3	Living together	primary education (4th grade)
3						Greater	Living	

	4	GP	F	15	Urban	than 3	together	higher educatior
4	5	GP	F	16	Urban	Greater than 3	Living together	secondary educatior

5 rows × 34 columns

In [6]:
```
#dados de português
port.info()
port.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 649 entries, 0 to 648
Data columns (total 34 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   student_id             649 non-null    int64 
 1   school                 649 non-null    object
 2   sex                    649 non-null    object
 3   age                    649 non-null    int64 
 4   address_type           649 non-null    object
 5   family_size            649 non-null    object
 6   parent_status          649 non-null    object
 7   mother_education       649 non-null    object
 8   father_education       649 non-null    object
 9   mother_job             649 non-null    object
 10  father_job             649 non-null    object
 11  school_choice_reason   649 non-null    object
 12  guardian               649 non-null    object
 13  travel_time            649 non-null    object
 14  study_time             649 non-null    object
 15  class_failures         649 non-null    int64 
 16  school_support         649 non-null    object
 17  family_support         649 non-null    object
 18  extra_paid_classes     649 non-null    object
 19  activities             649 non-null    object
 20  nursery_school         649 non-null    object
 21  higher_ed              649 non-null    object
 22  internet_access        649 non-null    object
 23  romantic_relationship  649 non-null    object
 24  family_relationship    649 non-null    int64 
 25  free_time              649 non-null    int64 
 26  social                 649 non-null    int64 
 27  weekday_alcohol        649 non-null    int64 
 28  weekend_alcohol        649 non-null    int64 
 29  health                 649 non-null    int64 
 30  absences               649 non-null    int64 
 31  grade_1                649 non-null    int64 
 32  grade_2                649 non-null    int64 
 33  final_grade            649 non-null    int64 
dtypes: int64(13), object(21)
memory usage: 172.5+ KB
```

Out[6]:

	student_id	school	sex	age	address_type	family_size	parent_status	mother_education
0	1	GP	F	18	Urban	Greater than 3	Apart	higher education
1	2	GP	F	17	Urban	Greater than 3	Living together	primary education (4th grade)
2	3	GP	F	15	Urban	Less than or equal to 3	Living together	primary education (4th grade)
3	4	GP	F	15	Urban	Greater than 3	Living together	higher education
4	5	GP	F	16	Urban	Greater than 3	Living together	secondary education

5 rows × 34 columns

Exercício 3

Utilizando os datasets anteriores identifique:

1. Quantos alunos de sexo masculino e feminino tem quem cada dataset.
2. Qual a média final dos alunos em cada disciplina.
3. Qual a media final dos alunos cujos parentes (ao menos um) tem nível superior comparada à dos alunos em que nenhum dos pais tem esse tipo de formação.