Performance des étudiants

Visualisation des données

Présenté par Gebran Assaad

Enseignée par DR.ING. Nicole CHALLITA EL BOUTY

2023 - 2024

Plan:

1. Introduction

2. Les Bibliothèques Importées
    - Comparaison des Bibliothèques: Bar Chart

3. Partie analytique:
    - Decompositions des Attributs
    - Attributs Binaires
    - Attributs Nominaux
    - Attributs Ordinaux
    - Attributs Numeriques
    - Groupement des notes
    - Relation entre les attributs ordianux
    - Relation entre les attributs numeriques et nominaux
    - Relation entre les attributs numeriques et ordinaux
    - Relation entre les attributs numeriques et binaires
    - Relation entre les attributs numeriques

4. Conclusion

5. Références

6. Annexe

1. Introduction:

Le présent rapport se penche sur l'analyse approfondie d'un ensemble de données relatif à la consommation d'alcool parmi les étudiants. Cette étude vise à explorer les liens entre la consommation d'alcool des étudiants et leurs performances académiques, tout en mettant en lumière les facteurs qui influencent ces deux aspects cruciaux de la vie estudiantine.

Dans ce rapport, nous utiliserons des techniques de visualisation de données et d'analyse statistique pour explorer les relations entre la consommation d'alcool, les performances académiques, et d'autres variables pertinentes.

L'objectif de cette étude est de fournir des informations utiles pour les éducateurs, les parents et les décideurs, en mettant en lumière l'impact de la consommation d'alcool sur la réussite scolaire, tout en identifiant des facteurs clés qui peuvent être pris en compte pour améliorer les résultats académiques des étudiants.

Ce rapport vise à contribuer à une meilleure compréhension de ces questions importantes, tout en offrant des perspectives exploitables pour améliorer l'expérience éducative des étudiants et promouvoir des comportements sains.

2. Les Bibliothèques Importées:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.gridspec as gs
from sklearn.preprocessing import LabelEncoder
```

Seaborn：Seaborn est une bibliothèque Python de visualisation de données qui simplifie la création de graphiques informatifs et esthétiquement plaisants. Basée sur Matplotlib, Seaborn est largement utilisée dans l'analyse de données et la science des données pour explorer, représenter et communiquer des informations à travers des visualisations statistiques. Cette bibliothèque se distingue par sa simplicité d'utilisation, ses styles prédéfinis et sa capacité à générer des graphiques de haute qualité, en faisant un outil précieux pour les professionnels de la data science et les analystes de données.

Axes3D : Axes3D est un objet utilisé dans la bibliothèque de visualisation Matplotlib en Python. Il permet de créer des graphiques en trois dimensions (3D), ce qui est particulièrement utile pour représenter des données spatiales ou des ensembles de données tridimensionnels. Cela est couramment utilisé dans des domaines tels que la modélisation mathématique, la géospatiale, la visualisation scientifique et l'ingénierie pour représenter des données dans un espace tridimensionnel.

<u>Comparaison des Bibliothèques:</u> Bar Chart

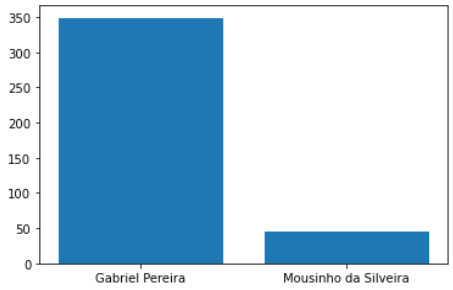
sns.countplot est principalement utilisé pour visualiser la fréquence d'occurrence des catégories dans une variable catégorielle, comme un comptage d'occurrences de chaque catégorie.Il est très simple à utiliser et ne nécessite qu'une seule ligne de code pour créer un bar chart avec des comptages.

Tandis que pour le **plt.bar**, pour voir la fréquence des variables catégorielle , il faut tapper un code pour calculer le nombre d'occurrence et après visualiser le bar chart.

C'est pour cela , il est très simple à utiliser **sns.countplot** et ne nécessite qu'une seule ligne de code pour créer un bar chart avec des comptages.

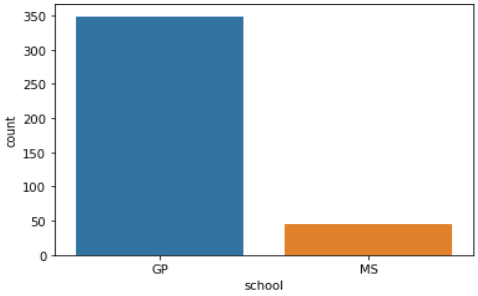
<u>Plt.bar :</u>

```
import matplotlib.pyplot as plt

gp_nb = data[data["school"] == "GP"].count()[1]

ms_nb = data[data["school"] == "MS"].count()[1]

schools = ["Gabriel Pereira", "Mousinho da Silveira"]

students_count = [gp_nb, ms_nb]

plt.bar(schools, students_count)

plt.show()
```



<u>sns.countplot:</u>

```
sns.countplot(data["school"])
```



3. Partie Analytique:

Decompositions des Attributs:

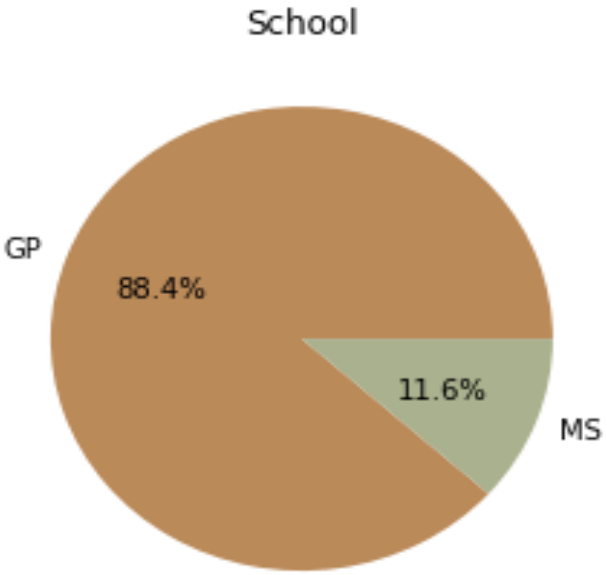
Dans cet ensemble de données, il y a 396 observations et on remarque qu'on a different types de données,

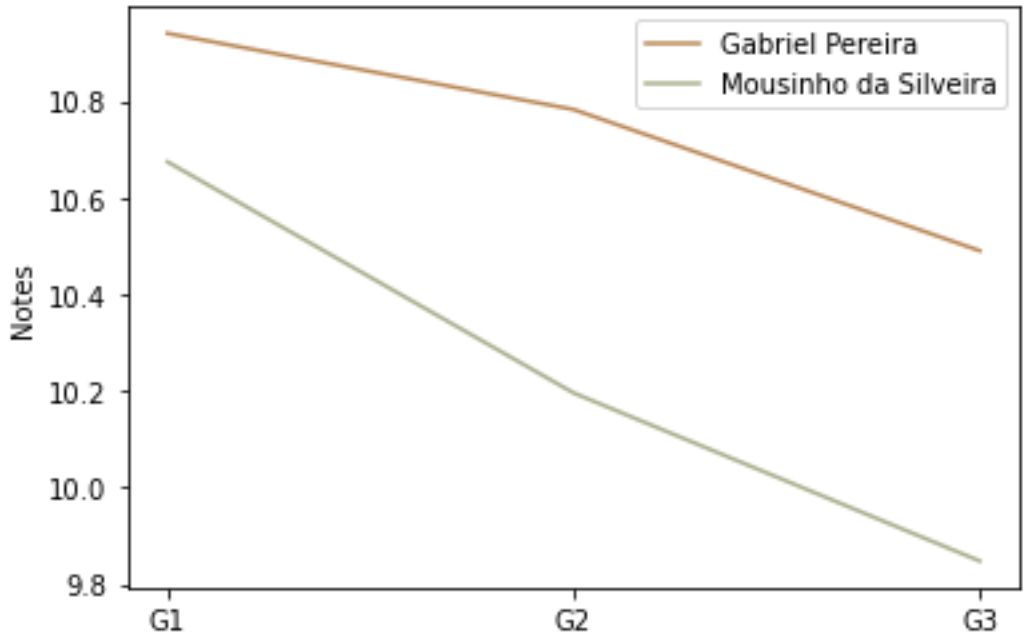
Pour faciliter l'etude de ces données , je les ai divisées de la forme suivante:

- Numeriques = ["age", "absences", "G1", "G2", "G3"]
- Ordinaux = ["Medu", "Fedu", "traveltime", "studytime", "failures", "famrel", "freetime", "goout", "Dalc", "Walc", "health"]
- Nominaux = ["Mjob", "Fjob", "reason", "guardian"]
- Binaires = ["school", "sex", "address", "famsize", "Pstatus", "schoolsup", "famsup", "paid", "activities", "nursery", "higher", "internet", "romantic"]

Attributs Binaires :

Pourcentage des écoles:



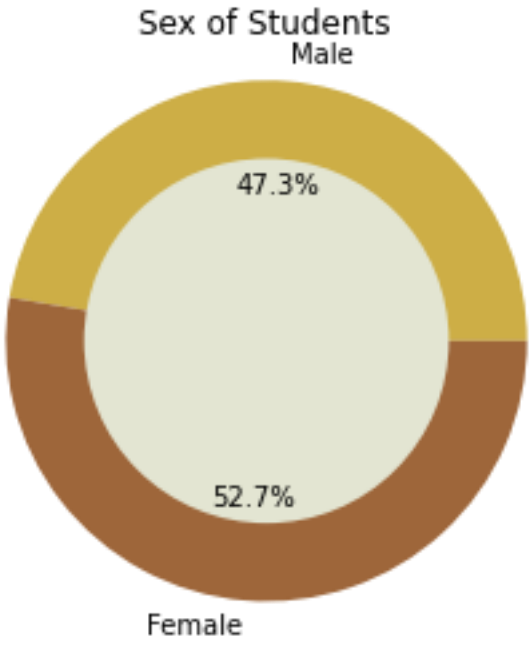


On voit que 88.4% des étudiants sont de l'école Gabriel Pereira tandis que les autres sont de Mousinho da Silveira.

Les notes de maths dans Gabriel Pereira sont plus meilleurs que celle dans Mousinho da Silveira

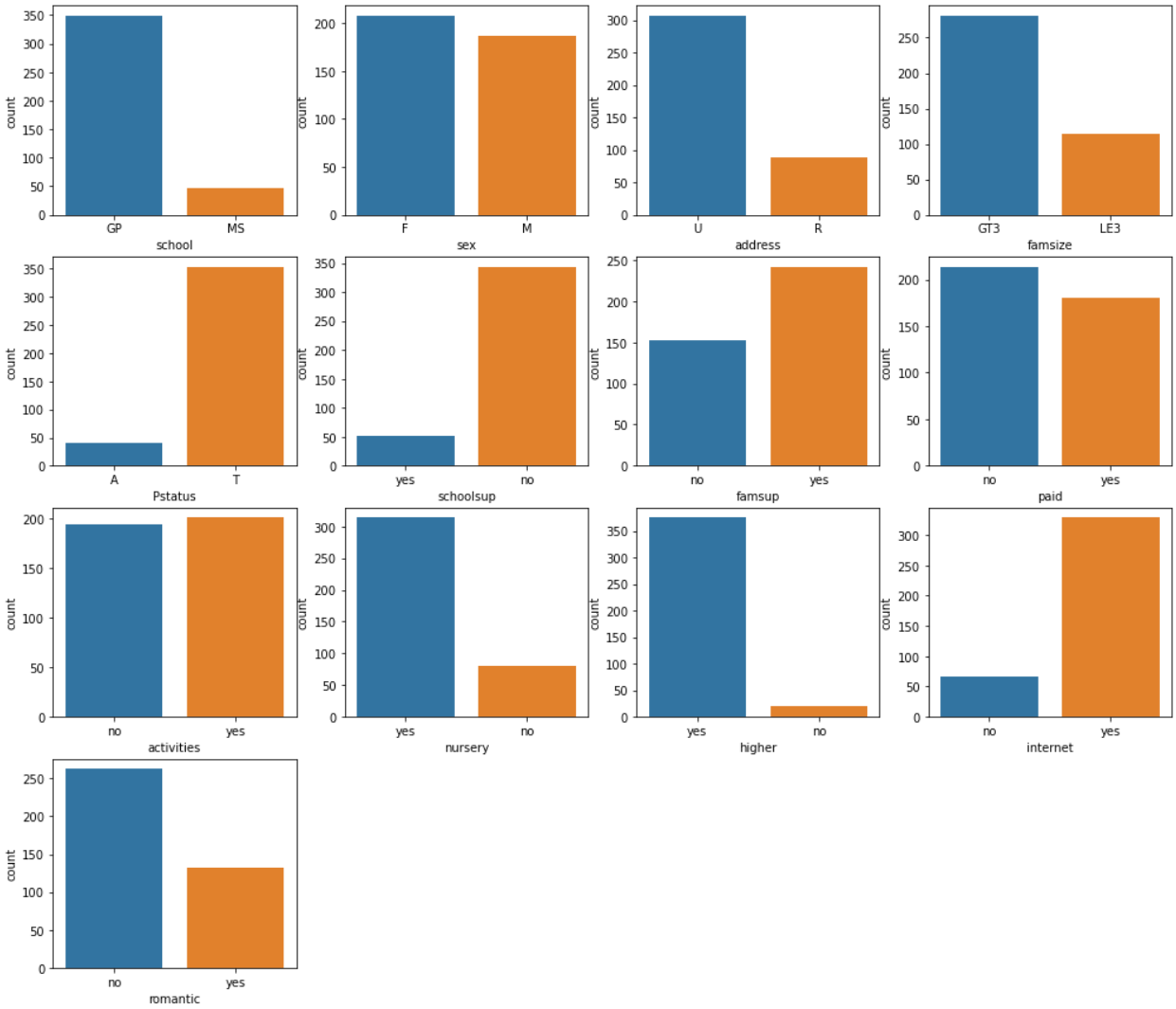
Pourcentage de Sex:

J’ai tracé Donut Plot pour visualizer les pourcentages des garçons et des filles dans les deux écoles.



Le Donut Plot montre que le nombre des filles est supérieur à celui des garçons ayant garçons (47.3%) et filles (52.7%)

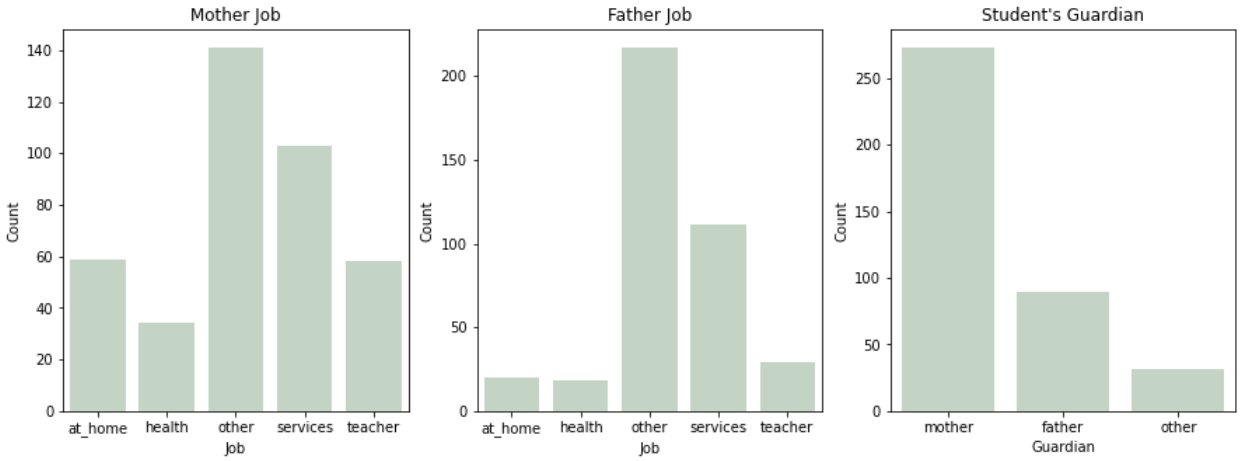
Les Autres Attributs:



On remarque que toutes les variables sont biaisées en faveur d'une valeur par rapport à l'autre, à l'exception de la variable "activities".

Attributs Nominaux:

Métier des Parents:



Mother:270/396=0.68

Father:95/396 = 0.23

Other:0.09

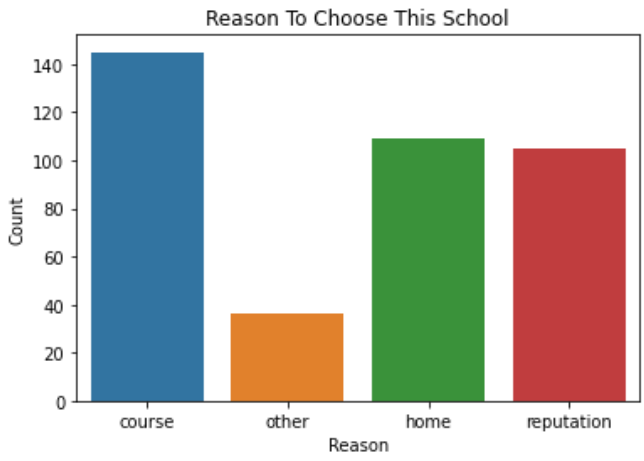
En dehors des professions liées aux services, à l'enseignement et à la santé, d'autres types de professions sont prédominants.

La plupart des étudiants (68 %) sont suivis par leur mère, 23% par leur père, et 9% par d'autres personnes. Ainsi, la mère joue un rôle important dans l'éducation des enfants.

Vous devriez mentionner qu'il y a des élèves qui vont à l'école sans avoir de parents.

Raison de choisir cette école:

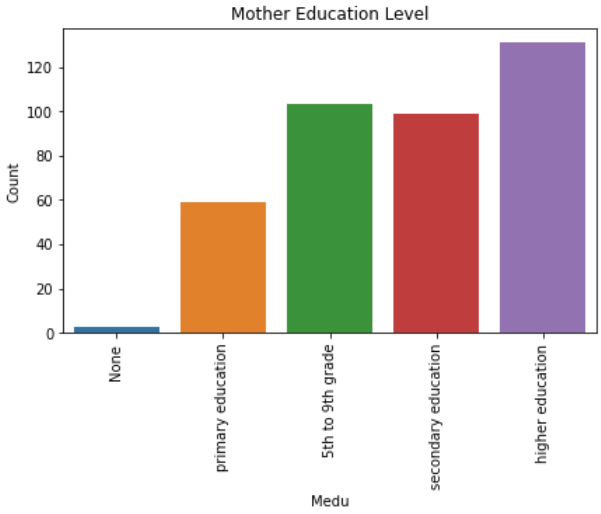
Course: 145/396=0.36 Home:110/396 = 0.27 Reputation:103/396 = 0.26 Other:0.11

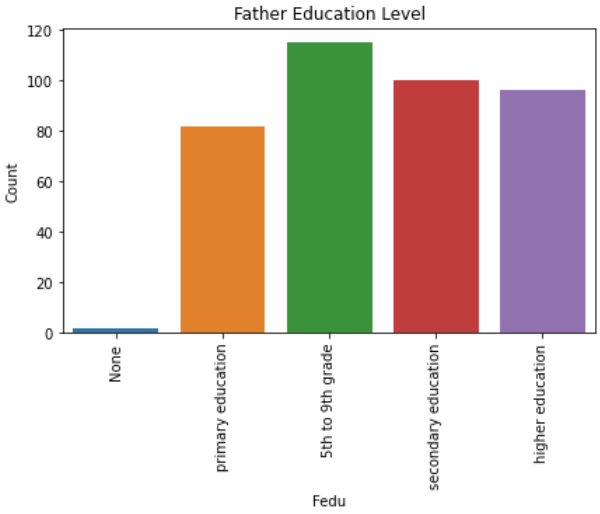


"36 % des étudiants ont choisi l'école en raison du programme d'études, d'autres parce qu'elle était proche de chez eux (27 %), en raison de la réputation de l'école (26 %), et une minorité pour d'autres raisons (11 %)."

Attributs Ordinaux:

Education des Parents:

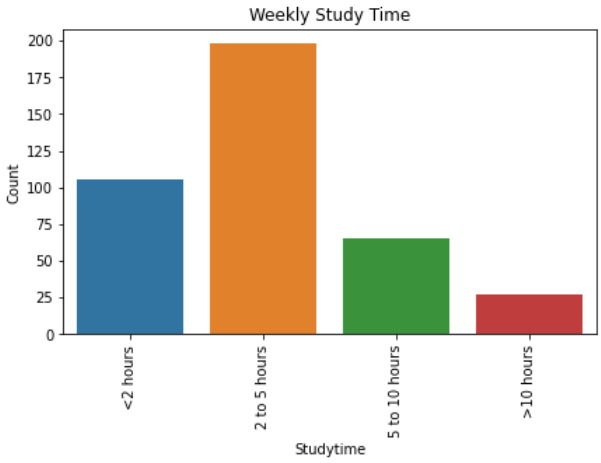


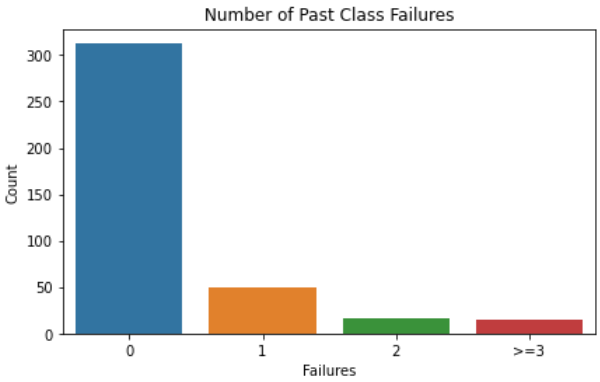


Les niveaux d'éducation (Medu et Fedu) présentent une distribution équilibrée, à l'exception des parents sans éducation. En fait, les parents sans éducation ne représentent qu'un faible pourcentage de tous les parents.

Temps d'étude et l'échecs des etudiants:

J'ai créé des barplots pour représenter le temps d'étude et les échecs des étudiants.

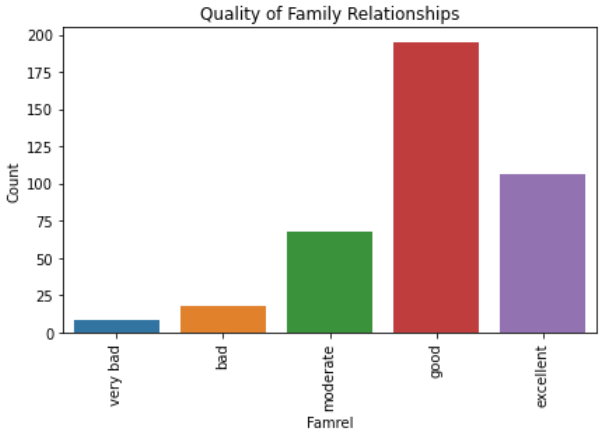


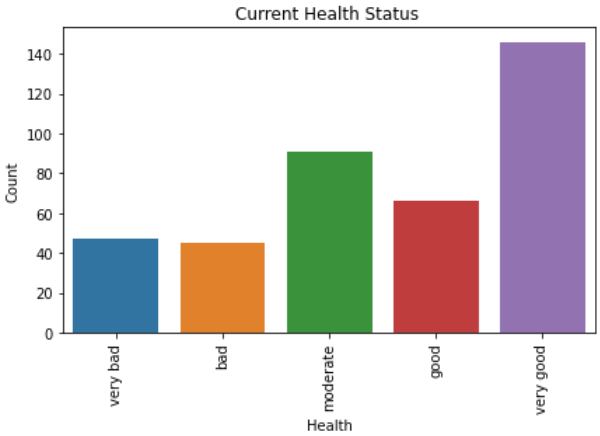


Près de la moitié des étudiants (47 %) étudient de 2 à 5 heures par semaine, 37 % moins de 2 heures par semaine, le reste plus de 5 heures par semaine.

La plupart des étudiants n'ont jamais échoué à un cours.

Qualité des relations familiales et État de santé actuel:





Excellent : 100/396 = 0.25

Good : 200/396 = 0.50

Moderate : 75/396 = 0.18

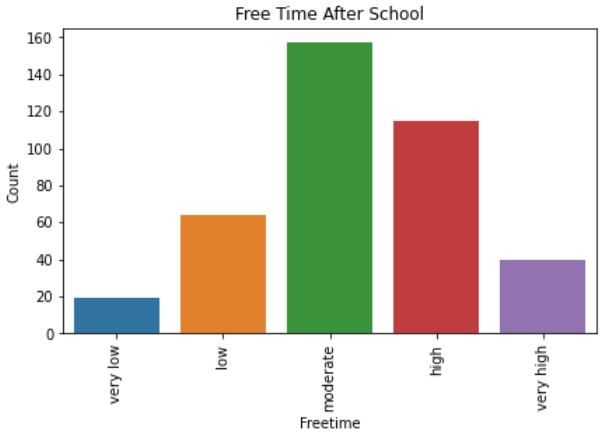
Bad : 25/396 = 0.06

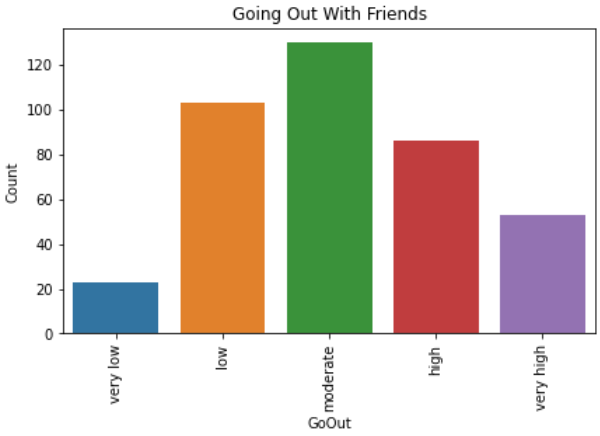
Very bad : 0.01

On distingue que la moitié des étudiants (50 %) se sentent à l'aise avec leur famille, 25 % se sentent très à l'aise avec leur famille, 18 % se sentent assez à l'aise, tandis que le reste des étudiants ne le sont pas.

Environ la moitié des étudiants sont en bonne santé.

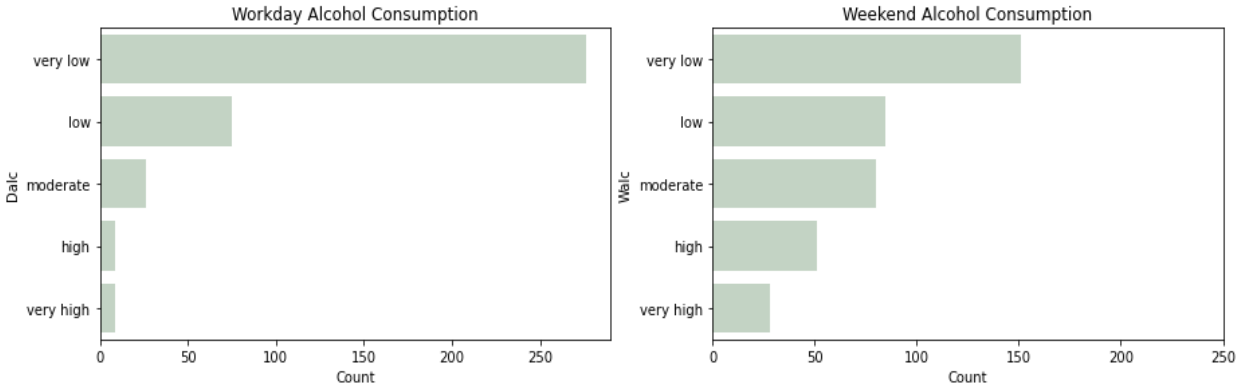
Temps libre et Sorties:





On note que Freetime et GoOut sont normallement distribuées.

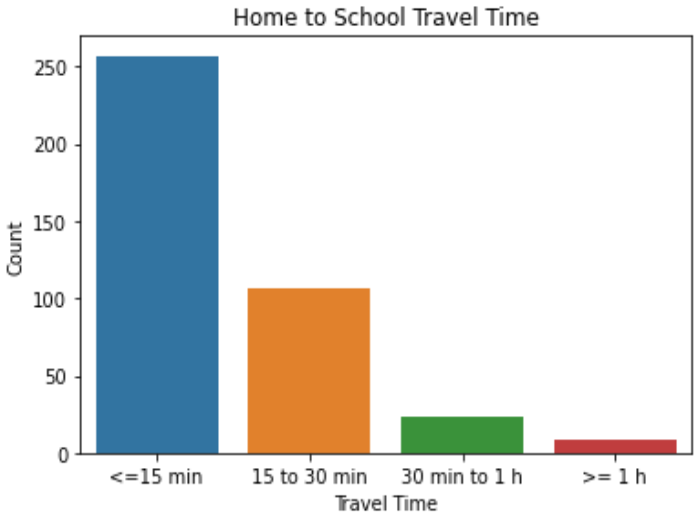
Consommation d'Alcool:



Heureusement, la consommation d'alcool en semaine est minimale. En fait, plus de la moitié des étudiants ne consomment pas, ou consomment très peu d'alcool en semaine.

Mais le week-end la consommation d'alcool augmente, et le groupe d'étudiants qui consomment ou consomment peu d'alcool devient dominant.

Temps de Trajet:

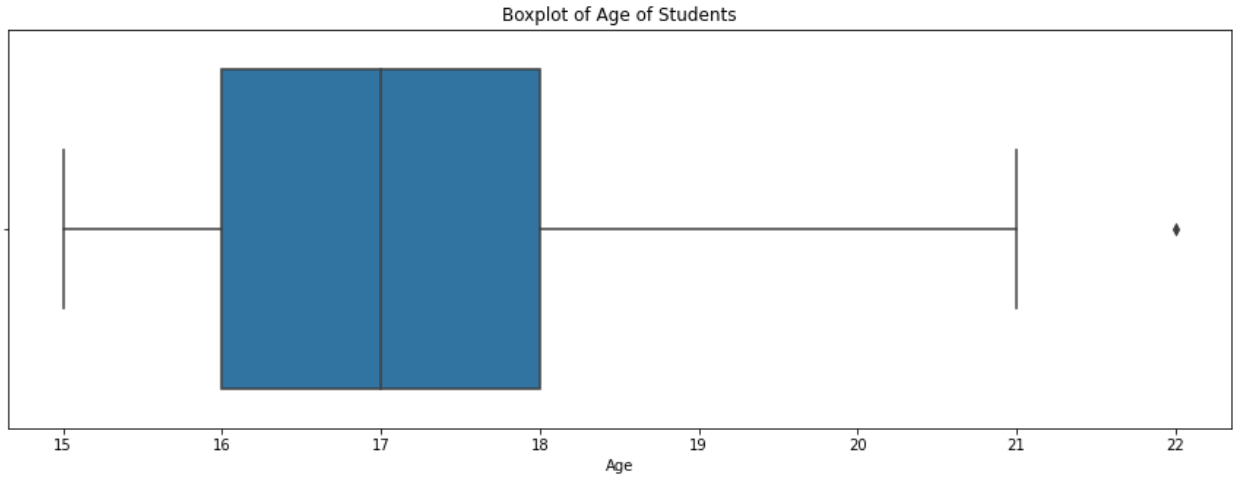


Plus de la moitié des élèves mettent moins de 15 minutes pour se rendre à l'école, environ un tiers met de 15 à 30 minutes, et le reste met plus de 30 minutes."

Attributs Numeriques:

Age des étudiants:

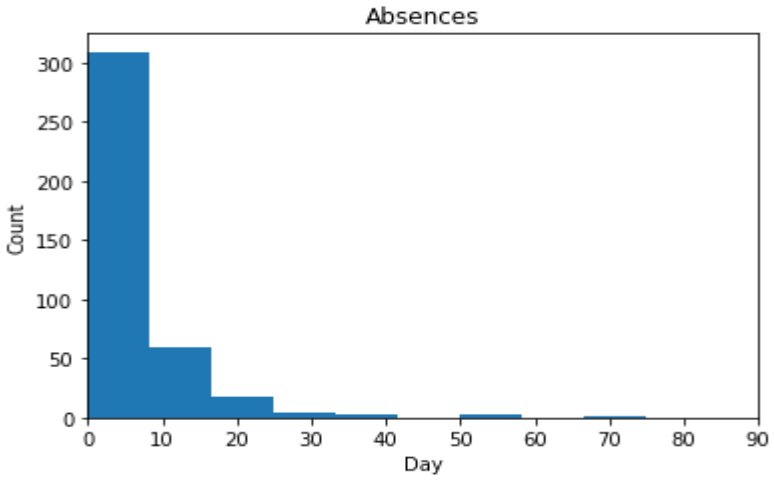
Dans le cadre de cette étude, un boxplot a été réalisé afin d'analyser la répartition des âges des étudiants.



Le boxplot montre qu'il y a un petit nombre d'étudiants plus âgés qui font pencher la distribution vers la droite, tandis que la majorité des étudiants ont des âges plus proches de la médiane.

Absences des étudiants:

J'ai tracé un histogramme pour représenter les niveaux d'absences des étudiants dans le rapport

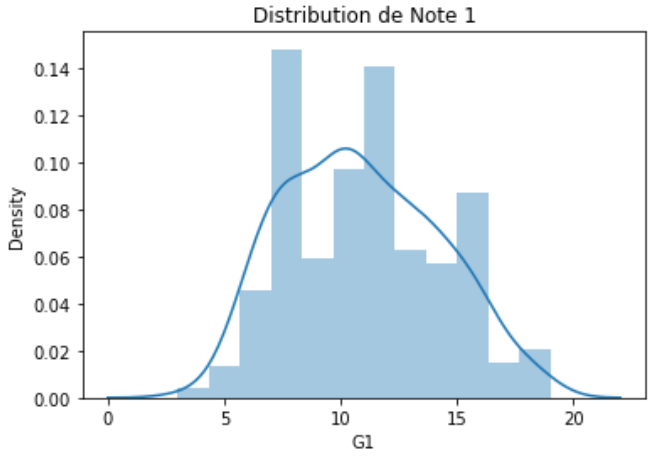


On repère une grande majorité d'étudiants n'ont jamais eu d'absences. Cependant, nous trouvons des exceptions après 10 absences.

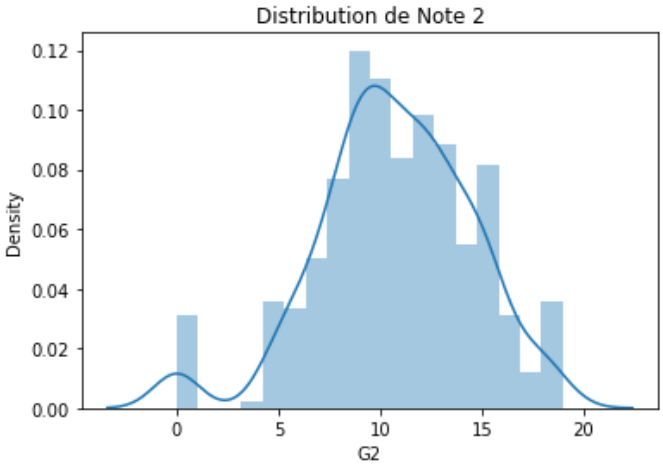
Les notes des trois examens:

Pour visualizer la distribution des notes des étudiants, on a tracé des histogrammes avec le ligne KDE . Il s'agit d'une version lissée et normalisée de la distribution des données. On a presenté ces graphes avec la densité , la surface sous la courbe de densité dans un intervalle représente la probabilité de trouver un point de données dans cet intervalle.

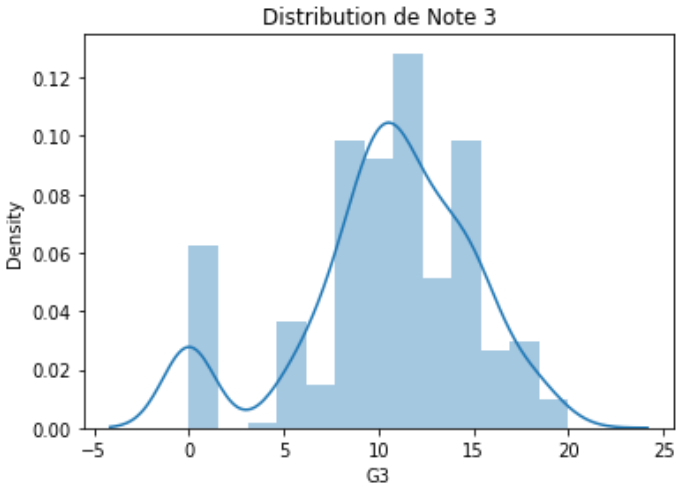
Les notes du premier examen:



Les notes du deuxième examen:



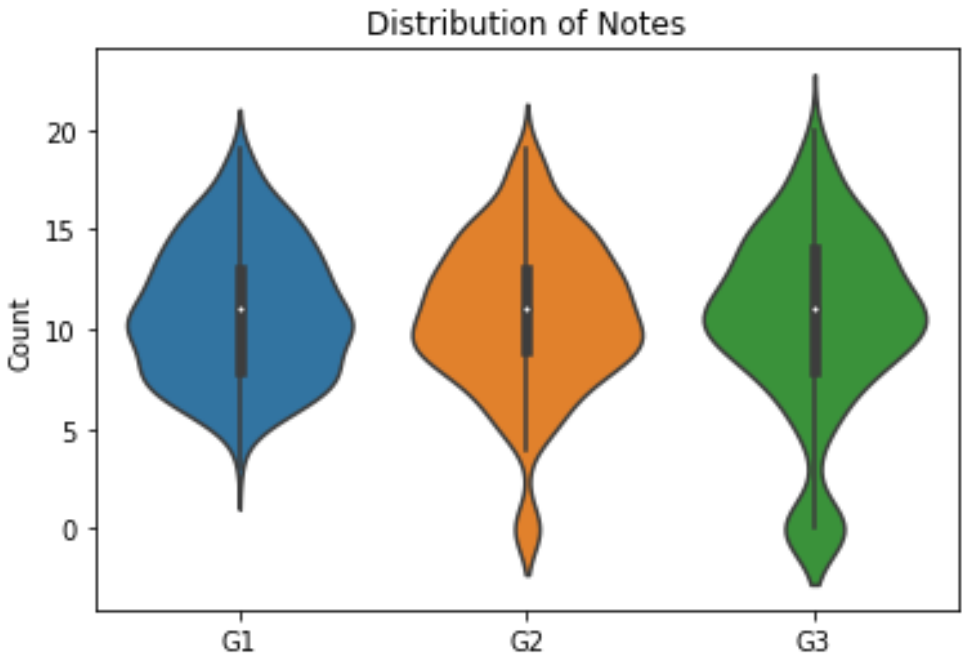
Les notes du troisième examen:



On découvre que les notes suivent une distribution normale, cependant, dans les périodes G2 et G3, il y a un petit groupe d'étudiants qui ont obtenu la note 0, ce qui est une anomalie par rapport à la distribution normale des notes.

Comparaison des Notes:

Dans ce Violin plot, les notes des trois examens sont representées pour les comparer.



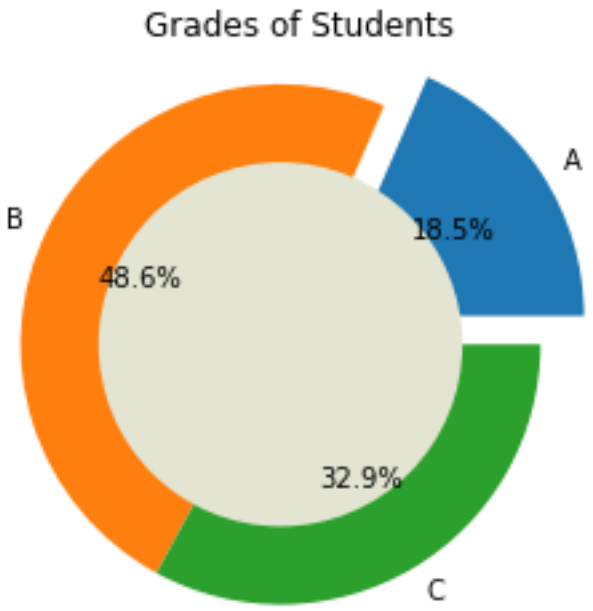
Groupement des Notes:

Pour mieux comprendre et analyser les facteurs qui affectent les notes des étudiants, on a divisé les notes de troisième examen en trois groupes:

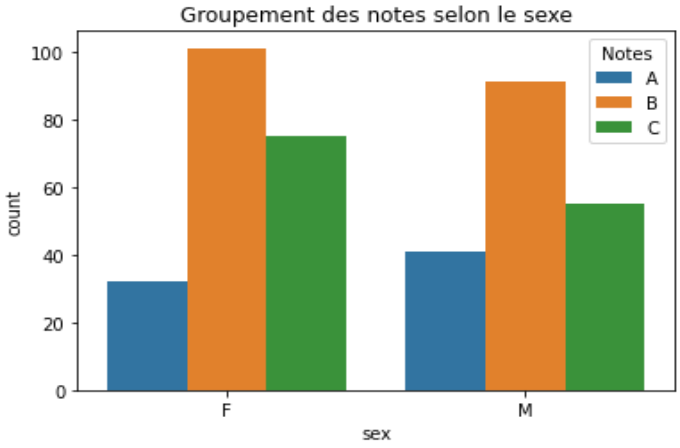
Groupe A: les notes >=15

Groupe B: les notes compris entre 10 et 14

Groupe C: les notes <10

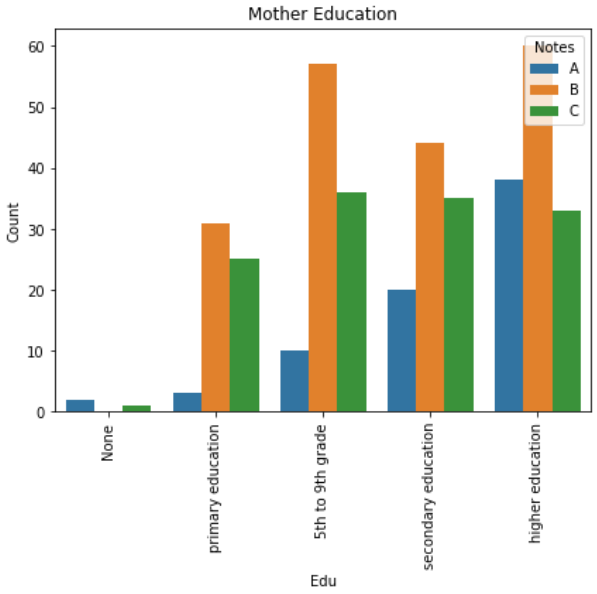


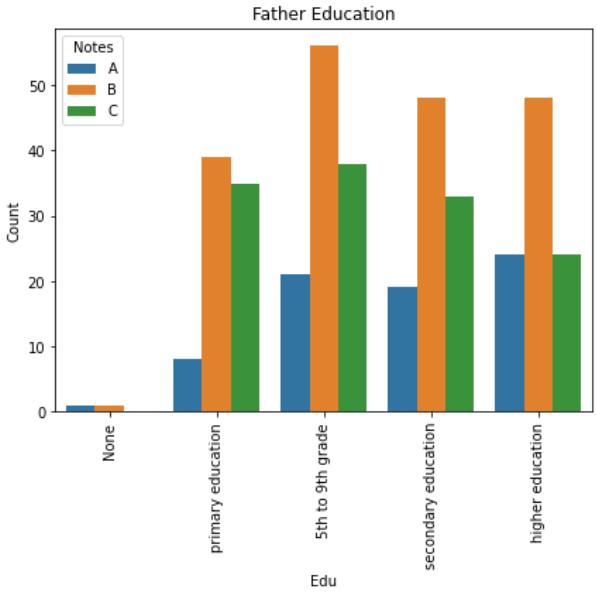
Selon le sexe:



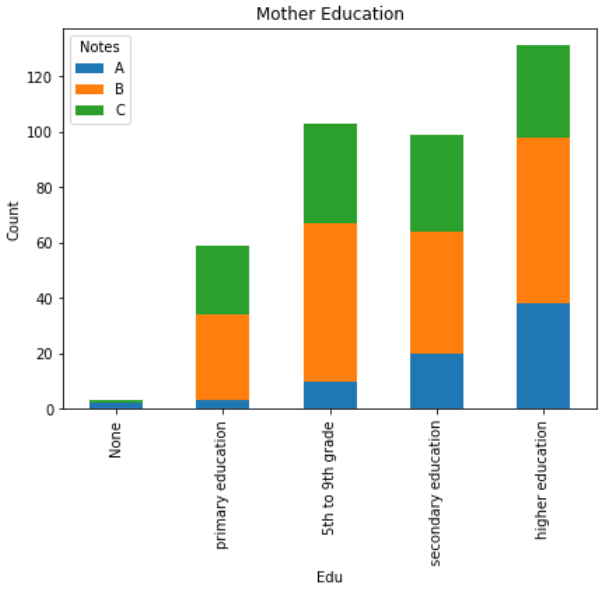
On peut conclure que les garçons ont des notes plus élevées que les filles.

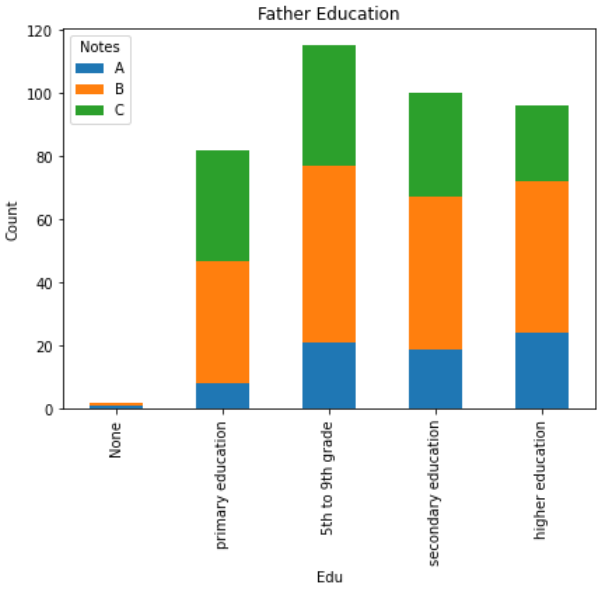
Selon les métiers des parents:





On peut representer cette resultat avec une autre methode: Stacked bar chart

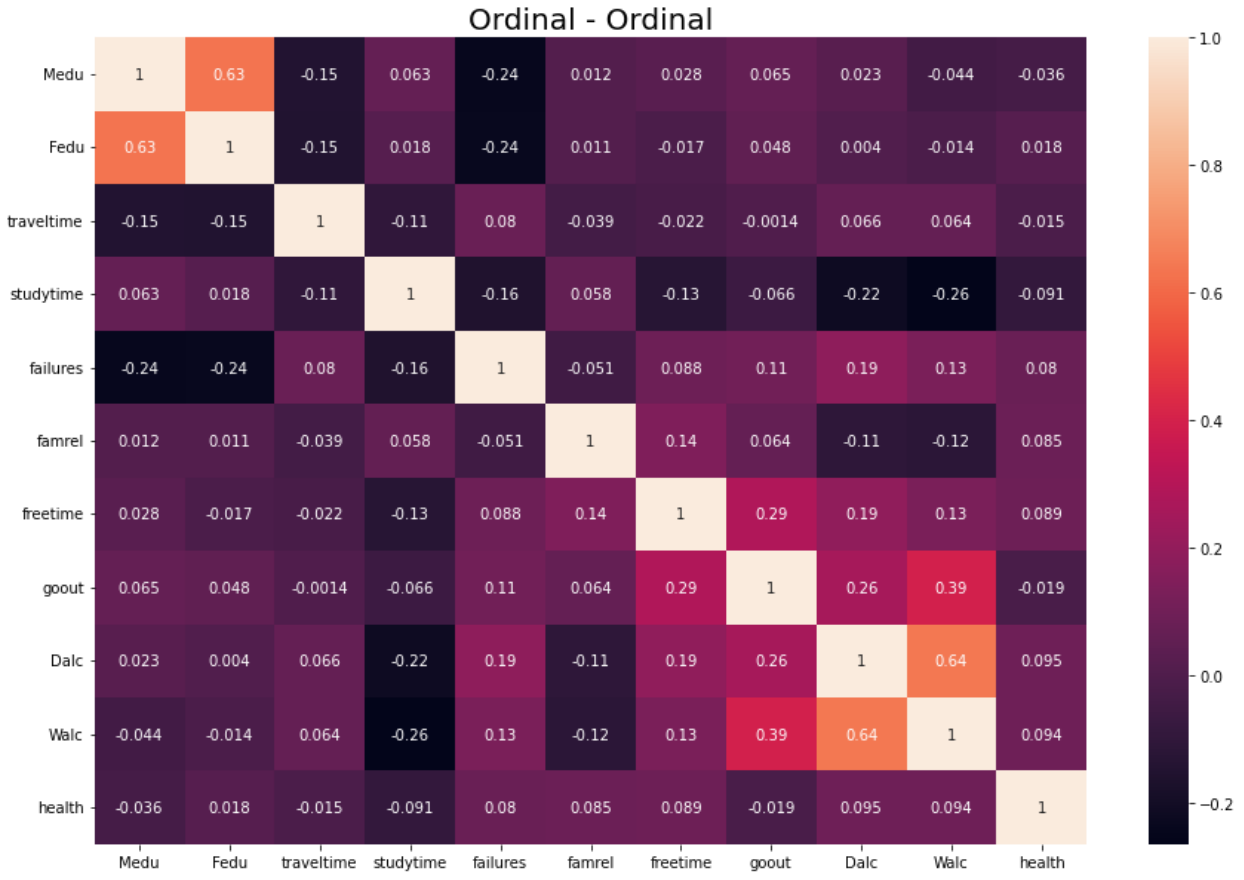




Cela nous indique qu'à mesure que le niveau d'éducation des parents augmente, les notes excellentes (en bleu) des élèves augmentent.

Relations entre les attributs ordinaux:

Pour voir la correlation entre les attributs ordinaux, on a tracé Heatmap qui represente ces correlations avec ses valeurs.



On repère "Medu" et "Fedu" présentent une forte corrélation positive (0,63). Nous pouvons donc dire que les parents ont tendance à avoir le même niveau d'éducation.

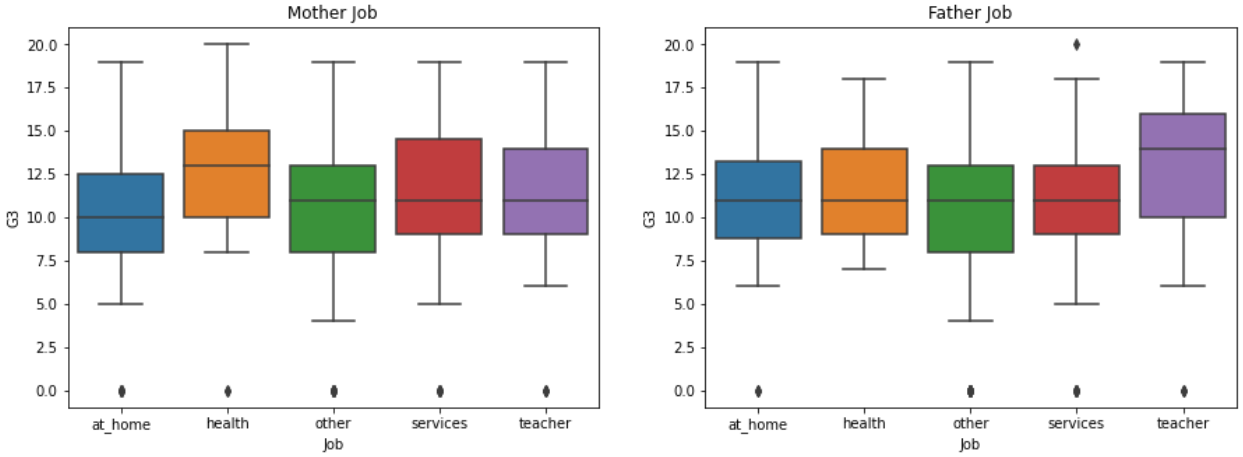
"Dalc" et "Walc" présentent une forte corrélation positive (0,64). Cela signifie que, en général, une forte consommation d'alcool en semaine est associée à une forte consommation d'alcool le week-end.

Walc" et "study time" présentent une corrélation négative modérée (-0,16). Ainsi, ceux qui consacrent plus de temps à étudier ont tendance à boire moins d'alcool le week-end.

Relations entre les attributs numeriques et nominaux:

Le tracé d'une heatmap entre des attributs nominaux (catégoriels) et numériques n'est pas une visualisation courante car les heatmaps sont généralement utilisées pour visualiser les relations entre des variables numériques. Cependant, vous pouvez créer un type similaire de graphique pour explorer la relation entre des attributs catégoriels et numériques. Une approche consiste à créer un graphique en barres groupées ou une boîte groupée pour visualiser la relation. Voici comment créer un graphique en boîtes groupées (grouped box plot) :

Métiers des parents:

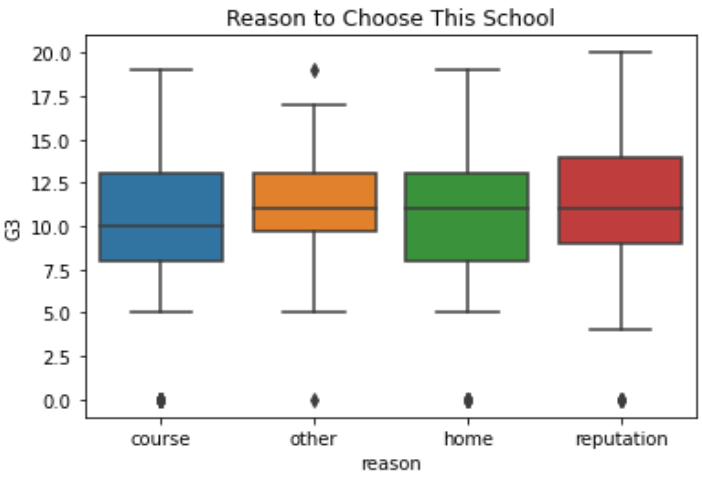


On constate les étudiants dont les mères travaillent dans le domaine de la 'santé' ont tendance à avoir de meilleures notes.

Les étudiants dont les pères travaillent dans le domaine de l'enseignement ont tendance à avoir de meilleures notes.

Raison de choisir cette école:

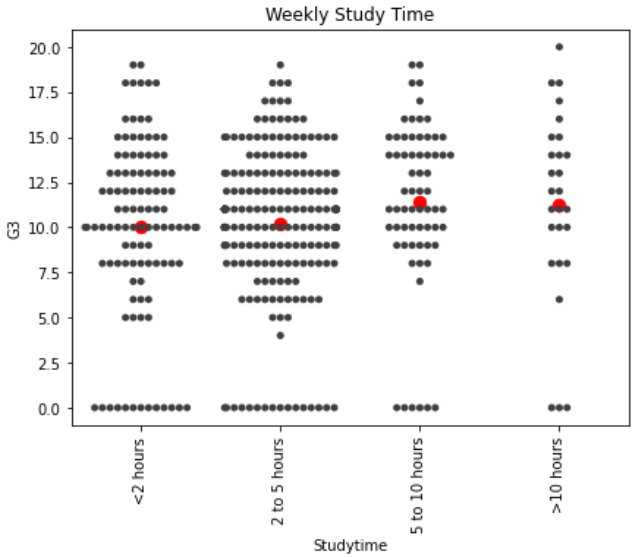
On identifie les étudiants qui ont choisi l'école pour sa réputation ont tendance à avoir de meilleures notes.

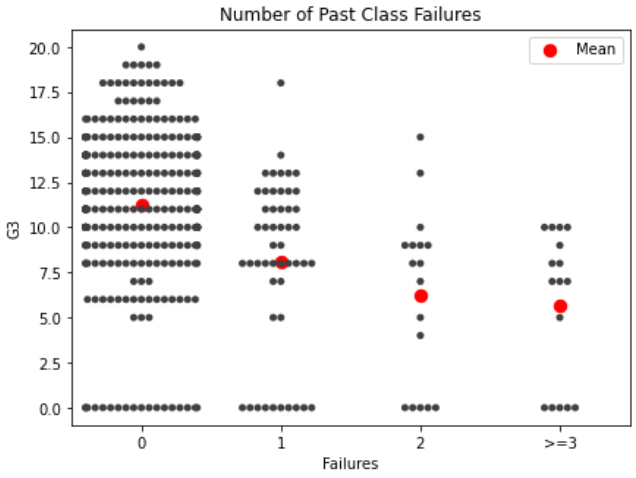


Relations entre les attributs numériques et ordinaux:

Pour conclure comment les attributs ordinaux affectent les notes des étudiants, on a choisit les plus logiques attributs qui peuvent augmenter ou diminuer les notes.

Selon temps d'étude et l'échec des etudiants:

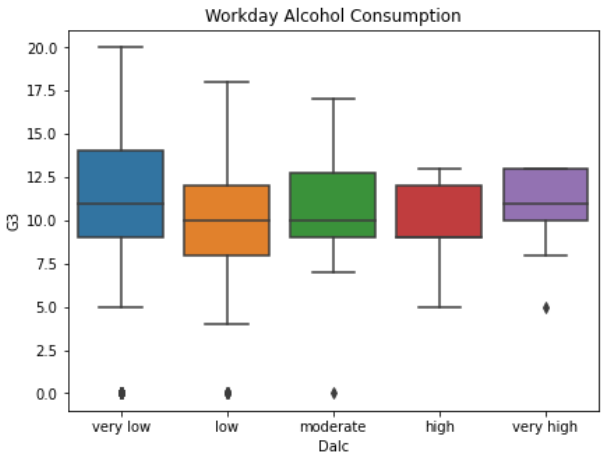


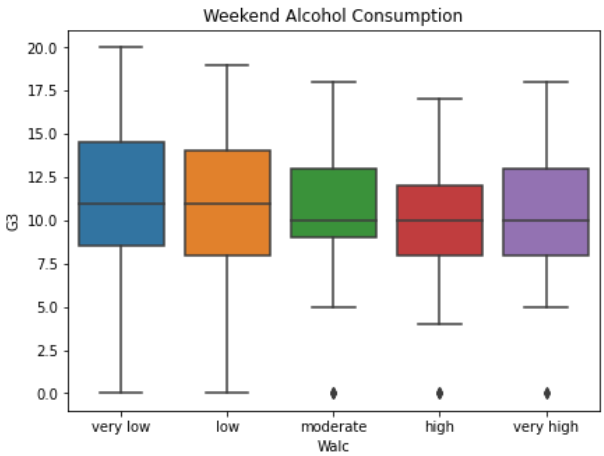


La médiane de G3 diminue quand les échecs des classes précédents augmentent (correlation negative).

La médiane de G3 augmente un peu quand le temps d'étude augmente (correlation faible et positive).

Selon la consommation d'alcool:

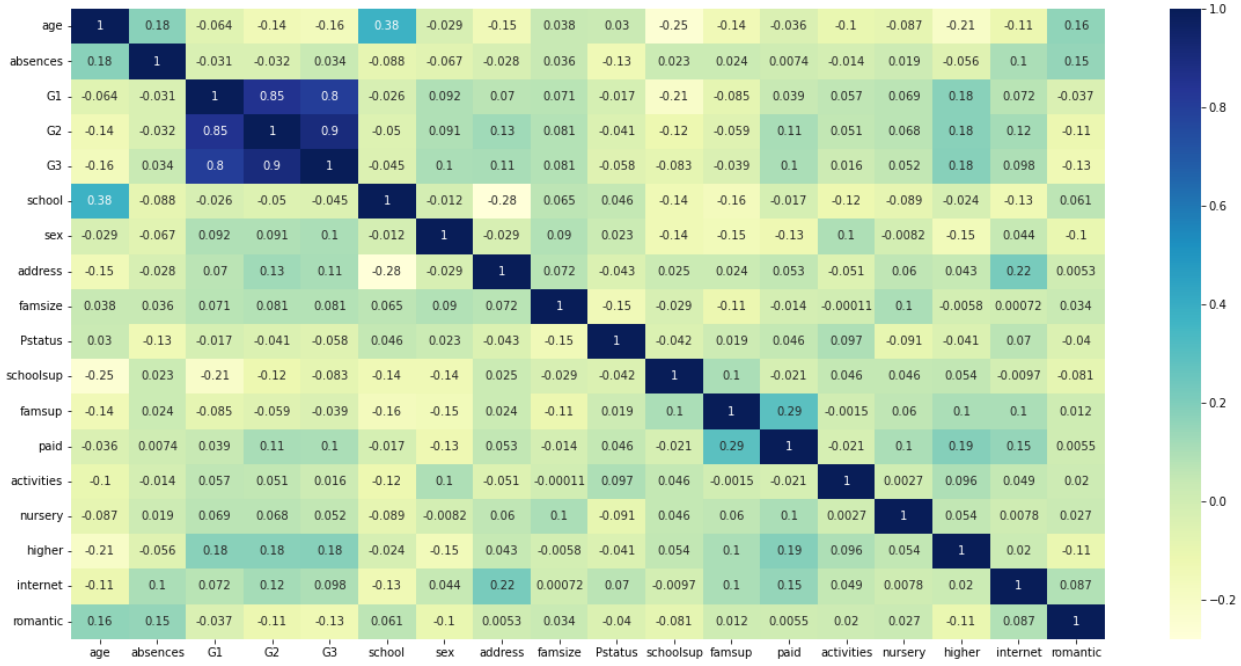




On peut observer que dans les deux cas, à mesure que la consommation d'alcool augmente, les notes ont tendance à diminuer.

Relations entre les attributs numeriques et binaires:

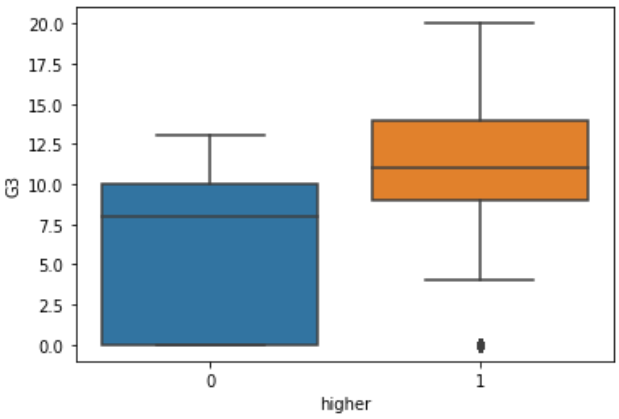
ATTENTION: Avant de commencer à analyser les attributs binaires, il faut les transformer en valeurs 1 ou 0 (exp: No = 0, Yes = 1)



On observe une correlation faible de “higher” avec G1,G2 et G3.

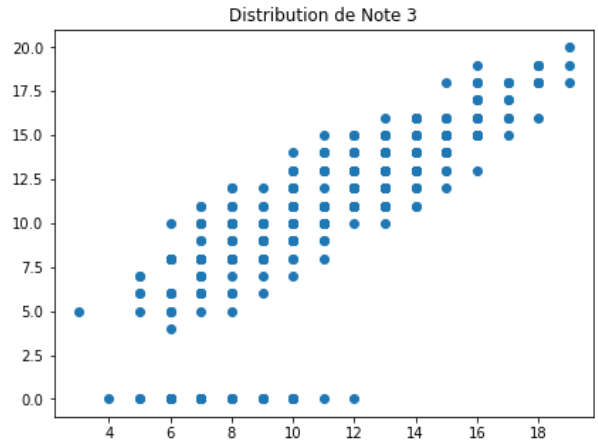
Pour mieux visualizer cette relation, on a tracé cette graphe pour voir l’effect de “higher” sur les notes des etudiants:

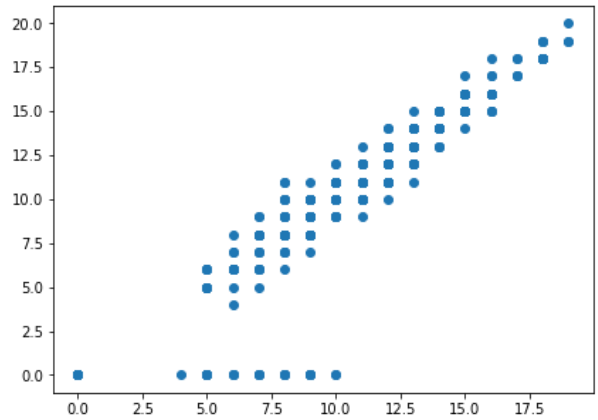
Les étudiants qui souhaitent poursuivre leurs études ont tendance à avoir de meilleures notes.



Relations entre les attributs numeriques:

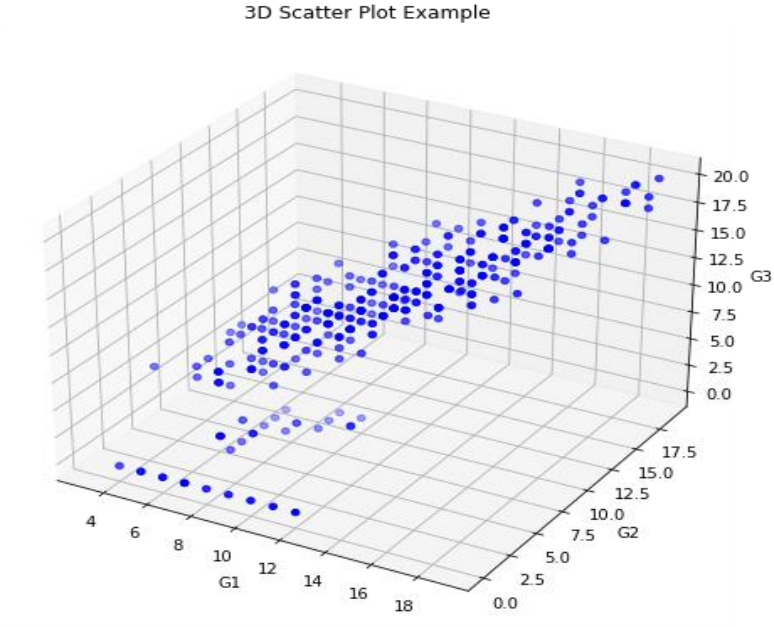
Pour voir la relation entre les notes du premier, deuxième et troisième, j'ai tracé le Scatter Plot de G1 et G2 en fonction de G3:





On remarque une forte relation lineaire entre G1 et G2 avec G3, c'est-à-dire quand G1 et G2 augmentent , G3 augmente

Voici un scatter plot 3D qui montre la corrélation entre les 3 notes:



On peut remarquer qu'il y a une certaine corrélation entre les trois scores.

Detection des valeurs aberrantes: On remarque qu'il existe des valeurs aberrantes dans les scatter plots, pour voir et identifier la cause de cette resultat on a tappé un ligne de code qui calcul la moyenne de G1 et G2 et affiche les caracteristiques des etudiants ayant G3 = 0

Out[287]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3	moyenne
296	GP	F	19	U	GT3	T	4	4	health	other	...	3	4	2	3	2	0	10	9	0	9.5
310	GP	F	19	U	LE3	T	1	2	services	services	...	2	4	2	2	3	0	9	9	0	9.0
316	GP	F	18	U	GT3	T	2	1	services	other	...	3	3	1	2	1	0	8	8	0	8.0
332	GP	F	18	U	GT3	T	3	3	services	services	...	3	4	1	1	4	0	7	0	0	3.5
333	GP	F	18	U	LE3	T	2	2	other	other	...	3	3	1	1	2	0	8	8	0	8.0
334	GP	F	18	R	GT3	T	2	2	at_home	other	...	4	4	1	1	4	0	10	9	0	9.5
337	GP	F	17	U	GT3	T	3	2	other	other	...	3	2	2	3	2	0	7	8	0	7.5
341	GP	M	18	U	GT3	T	4	4	teacher	services	...	3	3	2	2	2	0	10	10	0	10.0
343	GP	F	17	U	GT3	A	2	2	at_home	at_home	...	3	1	1	2	4	0	9	8	0	8.5
367	MS	F	17	R	GT3	T	1	1	other	services	...	2	1	1	2	1	0	7	6	0	6.5
383	MS	M	19	R	GT3	T	1	1	other	services	...	3	2	1	3	5	0	6	5	0	5.5
387	MS	F	19	R	GT3	T	2	3	services	other	...	4	2	1	2	5	0	7	5	0	6.0
389	MS	F	18	U	GT3	T	1	1	other	other	...	1	1	1	1	5	0	6	5	0	5.5

13 rows × 34 columns

Voici une partie des 38 étudiants avec G3 = 0. Nous remarquons que les moyennes G1 et G2 de ces étudiants sont inférieures ou égales à 10, il se peut donc qu'ils n'aient pas atteint un niveau suffisant pour passer l'examen final, ce qui expliquerait automatiquement le fait que G3 = 0.

Conclusion:

Dans le cadre de cette étude, nous avons exploré en détail les corrélations entre la consommation d'alcool des étudiants et leurs performances académiques, en mettant en lumière un certain nombre de facteurs qui influencent ces deux aspects de la vie étudiante.

Nous avons observé une corrélation négative entre la consommation d'alcool et les performances académiques, ce qui suggère que les étudiants qui consomment plus d'alcool ont tendance à obtenir des notes plus basses. Cette constatation met en évidence l'importance de la sensibilisation à la consommation d'alcool excessive chez les étudiants et de l'impact que cela peut avoir sur leur réussite académique.

Par ailleurs, nous avons identifié plusieurs facteurs familiaux et environnementaux qui jouent un rôle significatif dans les performances académiques des étudiants. Les niveaux d'éducation des parents, la fréquentation de l'école, le temps consacré aux études ont tous montré des corrélations importantes avec les notes des élèves.

Il est intéressant de noter que, bien que la majorité des étudiants obtiennent des notes comprises entre 10 et 15, nous avons également identifié un nombre significatif d'étudiants ayant obtenu des notes égales à zéro. Cela peut être dû à un manque d'efforts, à des facteurs extérieurs ou à d'autres circonstances

En conclusion, cette étude met en évidence l'importance de prendre en compte un large éventail de facteurs, y compris la consommation d'alcool, l'environnement familial et le comportement scolaire lors de l'analyse des performances académiques des étudiants. Ces informations peuvent être précieuses pour les éducateurs, les parents et les décideurs cherchant à améliorer la réussite scolaire des étudiants tout en favorisant des comportements sains.

Références:

Dataset: https://www.kaggle.com/datasets/uciml/student-alcohol-consumption/data

Bibliotheques: https://www.geeksforgeeks.org

Codes: https://www.w3schools.com/

Annexe:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.gridspec as gs
from sklearn.preprocessing import LabelEncoder



data= pd.read_csv("C:/Users/Win10RS8/Desktop/PROJET DATA VISUALIZATION/student-mat.csv")
```

#PieChart School

```
gp_nb = data[data["school"] == "GP"].count()[1]
ms_nb = data[data["school"] == "MS"].count()[1]
school = [gp_nb,ms_nb]
plt.pie(school , labels = ["GP","MS"],autopct='%1.1f%%', colors = ["#BB8A59","#AAB18E"])
plt.title("School")
plt.show()
```

#LineChart

```
GP = data[data["school"] =="GP"]
MS = data[data["school"] == "MS"]
x=['G1','G2','G3']
values1 = [np.mean(GP["G1"]),np.mean(GP["G2"]),np.mean(GP["G3"])]
values2 = [np.mean(MS["G1"]),np.mean(MS["G2"]),np.mean(MS["G3"])]
```

```
plt.ylabel("Notes")
plt.plot(x,values1, color = "#BB8A59")
plt.plot(x,values2, color = "#AAB18E")
plt.legend(["Gabriel Pereira","Mousinho da Silveira"])
plt.show()
```

#Donut Chart

```
nb_male = data[data["sex"] == "M"].count()[1]
nb_female = data[data["sex"] == "F"].count()[1]
sex = [nb_male,nb_female]
fig, ax = plt.subplots()
ax.pie(sex, labels=["Male","Female"], autopct='%1.1f%%',colors = ["#CDAE46","#9E663A"])
# Draw a circle at the center to create the donut chart
circle = plt.Circle((0, 0), 0.70, fc='#E3E5D2',alpha = 1)  # Adjust the second parameter for the size of the hole
fig.gca().add_artist(circle)
# Add a title
plt.title('Sex of Students')
# Equal aspect ratio ensures that pie is drawn as a circle.
ax.axis('equal')
# Display the chart
plt.show()
```

#Les attributs Binaires

```
import matplotlib.gridspec as gs
fig = plt.figure(figsize = (17,15))
```

```
g = gs.GridSpec(nrows = 4, ncols = 4, figure = fig)
i = 0
j = 0
for feature in binary:
    if j == 4:
        i = i + 1
        j = 0
    ax1 = plt.subplot(g[i,j])
    ax1 = sns.countplot(df[feature])
    j = j + 1
```

#BarPlot (Mjob-Fjob-Guardian)

```
fig,(ax1, ax2 , ax3) = plt.subplots(1, 3, figsize=(15, 5))
ax1 = sns.countplot(data["Mjob"],ax = ax1, color ="#C0D6C2")
ax1.set_xlabel("Job")
ax1.set_ylabel("Count")
ax1.set_title("Mother Job")
ax2 = sns.countplot(data["Fjob"],ax = ax2,order =
["at_home","health","other","services","teacher"],color ="#C0D6C2")
ax2.set_xlabel("Job")
ax2.set_ylabel("Count")
ax2.set_title("Father Job")
ax3 = sns.countplot(data["guardian"],ax = ax3, color ="#C0D6C2")
ax3.set_xlabel("Guardian")
ax3.set_ylabel("Count")
ax3.set_title("Student's Guardian")
plt.show()
```

#BarPlot (Reason)

```
sns.countplot(data["reason"])
plt.xlabel("Reason")
plt.ylabel("Count")
plt.title("Reason To Choose This School")
plt.show()
```

#BarPlot (Traveltime)

```
sns.countplot(data["traveltime"])
custom_ticks = ['<=15 min', '15 to 30 min', '30 min to 1 h', ">= 1 h"]  # Custom tick labels
tick_positions = [0, 1, 2, 3]
plt.xticks(tick_positions,custom_ticks)
plt.xlabel("Travel Time")
plt.ylabel("Count")
plt.title("Home to School Travel Time")
plt.show()
```

#BarPlot (Medu-Fedu)

```
fig,(ax1, ax2) = plt.subplots(1, 2, figsize=(15, 4))
ax1 = sns.countplot(data["Medu"],ax = ax1)
custom_ticks = ['None', 'primary education', '5th to 9th grade', "secondary education","higher education"]  # Custom tick labels
```

```
tick_positions = [0, 1, 2, 3, 4]

ax1.set_xticks(tick_positions,custom_ticks,rotation = 90)

ax1.set_xlabel("Medu")

ax1.set_ylabel("Count")

ax1.set_title("Mother Education Level")

ax2 = sns.countplot(data["Fedu"],ax = ax2)

custom_ticks = ['None', 'primary education', '5th to 9th grade', "secondary education","higher education"]  # Custom tick labels

tick_positions = [0, 1, 2, 3, 4]

ax2.set_xticks(tick_positions,custom_ticks,rotation = 90)

ax2.set_xlabel("Fedu")

ax2.set_ylabel("Count")

ax2.set_title("Father Education Level")

plt.show()
```

#BarPlot(Studytime-Failures)

```
fig,(ax1, ax2) = plt.subplots(1, 2, figsize=(15, 4))

ax1 = sns.countplot(data["studytime"],ax = ax1)

custom_ticks = ['<2 hours', '2 to 5 hours', '5 to 10 hours', ">10 hours"]  # Custom tick labels

tick_positions = [0, 1, 2, 3]

ax1.set_xticks(tick_positions,custom_ticks,rotation = 90)

ax1.set_xlabel("Studytime")

ax1.set_ylabel("Count")

ax1.set_title("Weekly Study Time")

ax2 = sns.countplot(data["failures"],ax = ax2)

custom_ticks = ['0', '1', '2', ">=3"]  # Custom tick labels

tick_positions = [0, 1, 2, 3]
```

```
ax2.set_xticks(tick_positions,custom_ticks)
ax2.set_xlabel("Failures")
ax2.set_ylabel("Count")
ax2.set_title("Number of Past Class Failures")
plt.show()
```

#BarPlot (Famrel – Health)

```
fig,(ax1, ax2) = plt.subplots(1, 2, figsize=(15, 4))
ax1 = sns.countplot(data["famrel"],ax = ax1)
custom_ticks = ['very bad', 'bad', 'moderate', "good","excellent"]  # Custom tick labels
tick_positions = [0, 1, 2, 3, 4]
ax1.set_xticks(tick_positions,custom_ticks,rotation = 90) # or we can write rotation = "vertical"
ax1.set_xlabel("Famrel")
ax1.set_ylabel("Count")
ax1.set_title("Quality of Family Relationships")
ax2 = sns.countplot(data["health"],ax = ax2)
custom_ticks = ['very bad', 'bad', 'moderate', "good","very good"]  # Custom tick labels
tick_positions = [0, 1, 2, 3, 4]
ax2.set_xticks(tick_positions,custom_ticks,rotation = 90)
ax2.set_xlabel("Health")
ax2.set_ylabel("Count")
ax2.set_title("Current Health Status ")
```

```
plt.show()
```

#BarPlot(Freetime-Goout)

```
fig,(ax1, ax2) = plt.subplots(1, 2, figsize=(15, 4))
ax1 = sns.countplot(data["freetime"],ax = ax1)
custom_ticks = ['very low', 'low', 'moderate', "high","very high"]  # Custom tick labels
tick_positions = [0, 1, 2, 3, 4]
ax1.set_xticks(tick_positions,custom_ticks,rotation = 90) # or we can write rotation = "vertical"
ax1.set_xlabel("Freetime")
ax1.set_ylabel("Count")
ax1.set_title("Free Time After School")
ax2 = sns.countplot(data["goout"],ax = ax2)
custom_ticks = ['very low', 'low', 'moderate', "high","very high"]  # Custom tick labels
tick_positions = [0, 1, 2, 3, 4]
ax2.set_xticks(tick_positions,custom_ticks,rotation = 90)
ax2.set_xlabel("GoOut")
ax2.set_ylabel("Count")
ax2.set_title("Going Out With Friends")
plt.show()
```

#Horizontal BarPlot (Dalc-Walc)

```
fig,(ax1, ax2) = plt.subplots(1, 2, figsize=(15, 4))
ax1 = sns.countplot(y = data["Dalc"],ax = ax1, color ="#C0D6C2")
custom_ticks = ['very low', 'low', 'moderate', "high","very high"]  # Custom tick labels
tick_positions = [0, 1, 2, 3, 4]
ax1.set_yticks(tick_positions,custom_ticks) # or we can write rotation = "vertical"
ax1.set_ylabel("Dalc")
```

```
ax1.set_xlabel("Count")
ax1.set_title("Workday Alcohol Consumption")
ax2 = sns.countplot( y = data["Walc"],ax = ax2, color ="#C0D6C2")
custom_ticks = ['very low', 'low', 'moderate', "high","very high"]  # Custom tick labels
tick_positions = [0, 1, 2, 3, 4]
ax2.set_yticks(tick_positions,custom_ticks)
ax2.set_ylabel("Walc")
ax2.set_xlabel("Count")
ax2.set_title("Weekend Alcohol Consumption")
ax2.set_xlim(0,250)
plt.show()
```

#BoxPlot (Age)

```
# Create a boxplot
plt.figure(figsize=(15, 5))
sns.boxplot(data["age"], x='age')
plt.xlabel("Age")
plt.title("Boxplot of Age of Students")
plt.show()
```

#Histogramme (Absences)

```
plt.hist(data["absences"] ,bins = 9)
plt.xlim(0,90)
plt.xlabel("Day")
plt.ylabel("Count")
plt.title("Absences")
```

#Histogrammes (G1-G2-G3)

```
sns.distplot(data["G1"])
plt.title("Distribution de Note 1")
plt.show()

sns.distplot(data["G2"])
plt.title("Distribution de Note 2")
plt.show()

sns.distplot(data["G3"])
plt.title("Distribution de Note 3")
plt.show()
#Violin Plot (G1-G2-G3)
sns.violinplot(data =data[["G1","G2","G3"]])
plt.title("Distribution of Notes")
plt.ylabel("Count")
plt.show()


#Groupement des notes
df = pd.DataFrame(data)
# Define the conditions and corresponding labels
conditions = [
    (df['G3'] >= 15) & (df['G3'] <= 20),
    (df['G3'] >= 10) & (df['G3'] <= 14),
    df['G3'] < 10
]
labels = ['A', 'B', 'C']
# Use numpy.select to create the 'Notes' column
```

```
df['Notes'] = np.select(conditions, labels, default=None)

# Display the updated DataFrame

print(df)
```

#PiePlot(Notes)

```
nb_A = data[data["Notes"] == "A"].count()[1]

nb_B = data[data["Notes"] == "B"].count()[1]

nb_C = data[data["Notes"] == "C"].count()[1]

notes = [nb_A,nb_B,nb_C]

fig, ax = plt.subplots()

ax.pie(notes, labels=["A","B","C"], autopct='%1.1f%%',explode = [0.2,0,0])


# Draw a circle at the center to create the donut chart

circle = plt.Circle((0, 0), 0.70, fc='white')  # Adjust the second parameter for the size of the hole

fig.gca().add_artist(circle)

# Add a title

plt.title('Grades of Students')

# Equal aspect ratio ensures that pie is drawn as a circle.

ax.axis('equal')

# Display the chart

plt.show()
```

#Grouped BarPlot(Sex-Notes)

```
p = sns.countplot(data = data , x = data["sex"],hue = data["Notes"],hue_order = ["A","B","C"])

plt.title("Groupement des notes selon le sexe")

plt.show()
```

#Grouped BarPlot(Edu-Notes)

```
fig , (p1,p2) = plt.subplots(1,2,figsize = (15,5))

p1 = sns.countplot(data = data , x = "Medu",hue = "Notes",ax = p1,hue_order = ["A","B","C"])

custom_ticks = ['None', 'primary education', '5th to 9th grade', "secondary education","higher education"]  # Custom tick labels

tick_positions = [0, 1, 2, 3, 4]

p1.set_xticks(tick_positions,custom_ticks,rotation = 90)

p1.set_xlabel("Job")

p1.set_ylabel("Count")

p1.set_title("Mother Education")


p2 =sns.countplot(data = data , x = "Fedu",hue = "Notes",ax = p2,hue_order = ["A","B","C"])

custom_ticks = ['None', 'primary education', '5th to 9th grade', "secondary education","higher education"]  # Custom tick labels

tick_positions = [0, 1, 2, 3, 4]

p2.set_xticks(tick_positions,custom_ticks,rotation = 90)

p2.set_xlabel("Job")

p2.set_ylabel("Count")

p2.set_title("Father Education")

plt.show()
```

#Stacked BarPlot (Edu-Notes)

```
fig , (p1,p2) = plt.subplots(1,2,figsize = (15,5))

# Créez un DataFrame à partir de vos données

# Comptez le nombre d'étudiants par éducation maternelle et par note

count_data1 = df.groupby(['Medu', 'Notes']).size().unstack()

# Créez un graphique à barres empilées
```

```
count_data1.plot(kind='bar', stacked=True,ax = p1)

custom_ticks = ['None', 'primary education', '5th to 9th grade', "secondary education","higher education"]

tick_positions = [0, 1, 2, 3, 4]

p1.set_xticks(tick_positions,custom_ticks,rotation = 90)

p1.set_xlabel("Edu")

p1.set_ylabel("Count")

p1.set_title("Mother Education")

count_data2 = df.groupby(['Fedu', 'Notes']).size().unstack()

count_data2.plot(kind = "bar",stacked = True,ax = p2)

custom_ticks = ['None', 'primary education', '5th to 9th grade', "secondary education","higher education"]  # Custom tick labels

tick_positions = [0, 1, 2, 3, 4]

p2.set_xticks(tick_positions,custom_ticks,rotation = 90)

p2.set_xlabel("Edu")

p2.set_ylabel("Count")

p2.set_title("Father Education")

plt.show()
```

#Heatmap(ord-ord)

```
fig = plt.figure(figsize = (15,10))

sns.heatmap(df[ordin].corr(method = "spearman"), annot = True)

plt.title("Ordinal - Ordinal", size = 20)
```

#Grouped Boxplots (Mjob-Fjob)

```
fig,(ax1, ax2) = plt.subplots(1, 2, figsize=(15, 5))

ax1 = sns.boxplot(x=data["Mjob"],y = data["G3"],ax = ax1)

ax1.set_xlabel("Job")

ax1.set_title("Mother Job")
```

```
ax2 = sns.boxplot(data["Fjob"],y = data["G3"],ax = ax2,order =
["at_home","health","other","services","teacher"])

ax2.set_xlabel("Job")

ax2.set_title("Father Job")

plt.show()
```

#Grouped Boxplot (Reason)

```
sns.boxplot(x = data["reason"], y = data["G3"])

plt.title("Reason to Choose This School")

plt.show()
```

#SwarmPlot (Studytime - Failures)

```
fig,(ax1, ax2) = plt.subplots(1, 2, figsize=(15, 5))

ax1 = sns.swarmplot(data["studytime"],y= data["G3"],ax = ax1,color = "0.25")

custom_ticks = ['<2 hours', '2 to 5 hours', '5 to 10 hours', ">10 hours"]  # Custom tick
labels

tick_positions = [0, 1, 2, 3]

ax1.set_xticks(tick_positions,custom_ticks,rotation = 90)

ax1.set_xlabel("Studytime")

ax1.set_title("Weekly Study Time")

ax2 = sns.swarmplot(x = data["failures"],y = data["G3"],ax = ax2,color = "0.25")

custom_ticks = ['0', '1', '2', ">=3"]  # Custom tick labels

tick_positions = [0, 1, 2, 3]

ax2.set_xticks(tick_positions,custom_ticks)

ax2.set_xlabel("Failures")

ax2.set_title("Number of Past Class Failures")

mean1 = data.groupby("studytime")["G3"].mean().values

mean2 = data.groupby("failures")["G3"].mean().values
```

```
ax1.scatter(tick_positions, mean1,  color='red', s=70, label='Mean')
ax2.scatter(tick_positions, mean2,  color='red', s=70, label='Mean')
plt.legend()
plt.show()
```

#Grouped Boxplot(Dalc-Walc)

```
fig,(ax1, ax2) = plt.subplots(1, 2, figsize=(15, 5))
ax1 = sns.boxplot(x = data["Dalc"],y = data["G3"],ax = ax1)
custom_ticks = ['very low', 'low', 'moderate', "high","very high"]  # Custom tick labels
tick_positions = [0, 1, 2, 3, 4]
ax1.set_xticks(tick_positions,custom_ticks) # or we can write rotation = "vertical"
ax1.set_xlabel("Dalc")
ax1.set_title("Workday Alcohol Consumption")
ax2 = sns.boxplot( x = data["Walc"],y = data["G3"],ax = ax2)
custom_ticks = ['very low', 'low', 'moderate', "high","very high"]  # Custom tick labels
tick_positions = [0, 1, 2, 3, 4]
ax2.set_xticks(tick_positions,custom_ticks)
ax2.set_xlabel("Walc")
ax2.set_title("Weekend Alcohol Consumption")
plt.show()
```

#Transform binary attributs

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for feature in binary:
    data[feature] = le.fit_transform(data[feature])
    print(feature, le.classes_)
```

#Heatmap(Binary -Num)

```
fig = plt.subplots(figsize = (20,10))
sns.heatmap( data[num+binary].corr(),annot = True,cmap = "YlGnBu")
```

#BarPlot(Higher)

```
sns.boxplot(x = data["higher"] , y = data["G3"])
plt.show()
```

#ScatterPlot(G1-G2)

```
plt.subplots(1, 2, figsize=(15, 5))
plt.subplot(1,2,1)
plt.scatter(data["G1"],data["G3"])
plt.title("Distribution de Note 3"
plt.subplot(1,2,2)
plt.scatter(data["G2"],data["G3"])
plt.show()
```

#3D Scatter

```
# Create a 3D scatter plot
fig = plt.figure(figsize = (10,8))
ax = fig.add_subplot(111 ,projection='3d')
ax.scatter(data["G1"], data["G2"], data["G3"], c="blue", marker='o')
ax.set_xlabel('G1')
ax.set_ylabel('G2')
```

```
ax.set_zlabel('G3')
plt.title('3D Scatter Plot Example')
plt.show()

#Afficher les donnees ayant G3 = 0
data["moyenne"] = (data["G1"]+ data["G2"])/2
data[data["G3"] == 0]
data[data["G3"] == 0].iloc[25:38]
```

Références:

Dataset: https://www.kaggle.com/datasets/uciml/student-alcohol-consumption/data

Bibliotheques: https://www.geeksforgeeks.org

Codes: https://www.w3schools.com/