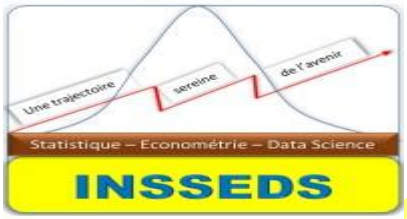
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

REPUBLIQUE DE CÔTE D'IVOIRE

Union – Discipline – Travail



INSTITUT SUPERIEUR DE STATISTIQUE, D'ECONOMETRIE ET DE DATASCIENCE

MASTER 2

STATISTIQUE ECONOMETRIE DATA SCIENCE

MINI PROJET STATISTIQUE INFERENTIELLE

RAPPORT STATISTIQUE SUR LA DEPRESSION DES ETUDIANTS

ETUDE DES FACTEURS INFLUENCANT LA DEPRESSION CHEZ LES ETUDANTS A L'AIDE DE METHODES STATISTIQUES

ANNEE UNIVERSITAIRE : 2024-2025

ETUDIANT :

N'DRI ONESIME

ENSEIGNANT :

AKPOSSO DIDIER MARTIAL

AVANT-PROPOS

L'association du théorique au pratique, des connaissances aux compétences et des savoir-faire aux savoirs est la principale tendance récente dans le secteur technique. Dans ce contexte, l'INSSEDS (Institut Supérieur de la Statistique, d'Econométrie et de la Data Science), dans sa formation en master professionnel en statistique, économie et science des données, impose que les divers crédits soient validés en effectuant un mini-projet à la fin de chaque module. Le projet est donc structuré et supervisé de cette manière, visant principalement à faire de chaque élève un participant dynamique, engagé et libre dans la vie active.

Ce document est un rapport de projet Muni axé sur la statistique inférentielle. Il se divise principalement en trois parties : La section initiale traite de la préparation des données ; la seconde partie est associée aux analyses et inférence statistique.

En règle générale, toutes les analyses et conclusions présentées dans ce rapport relèvent de la responsabilité de l'auteur, qui ne sollicite ni autrui ni l'INSSEDS (Institut Supérieur de Statistique d'Econométrie et de Data Science).

TABLE DES MATIERES

INTRODUCTION

Contexte et justification de l'étude

La dépression est une condition mentale sérieuse qui touche une proportion croissante de la population étudiante. Les pressions académiques, les exigences sociales et les préoccupations personnelles contribuent à cette situation. Comprendre les divers facteurs qui influencent la dépression chez les étudiants est essentiel pour élaborer des stratégies d'intervention efficaces et améliorer le bien-être général des étudiants.

Problématique

La problématique centrale de cette étude est de déterminer les facteurs spécifiques liés au mode de vie, aux études et aux antécédents personnels qui sont les plus fortement associés à la dépression chez les étudiants. En identifiant ces facteurs, nous pouvons mieux cibler les interventions pour prévenir et traiter la dépression.

Principaux résultats attendus

Les principaux résultats attendus de cette étude incluent :
1. Une compréhension approfondie des facteurs associés à la dépression chez les étudiants.
2. Des insights sur les différences de niveaux de satisfaction aux études et au travail entre les étudiants déprimés et non déprimés.
3. Une analyse de l'indépendance entre la dépression et des variables telles que les habitudes alimentaires et la durée du sommeil.

Méthodologie

Cette étude suivra une méthodologie en trois étapes principales :
1. Prétraitement des données : Nettoyage et préparation des données pour l'analyse.
2. Analyse et inférence statistique : Analyse des données pour identifier les patterns et les corrélations, estimation des intervalles de confiance, calcul des moyennes et médianes, et tests de différences de moyennes.
3. Conclusion : Synthèse des résultats et propositions d'interventions basées sur les preuves.

Description du jeu de données : dictionnaire des données

Le jeu de données utilisé dans cette étude contient diverses caractéristiques visant à analyser les niveaux de dépression chez les étudiants:

1. id: Identifiant unique de chaque étudiant.
2. sexe: Sexe (Masculin/Féminin).
3. Age: Âge de l'étudiant.
4. ville: Ville de résidence.
5. profession: Profession (étudiant à temps plein ou autre activité).
6. pression_academique: Pression académique (échelle).
7. pression_liee_au_travail: Pression liée au travail (échelle).
8. moyenne_notes: Moyenne générale des notes.
9. satisfaction_etudes: Satisfaction par rapport aux études (échelle).
10. satisfaction_travail: Satisfaction par rapport au travail (échelle).
11. duree_sommeil: Durée du sommeil (catégorielle, ex. : 5-6 heures, moins de 5 heures).
12. habitudes_alimentaires: Habitudes alimentaires (ex. : saines, modérées).
13. diplome_suivi: Diplôme suivi ou obtenu (ex. : BSc, M.Tech).
14. pensees_suicidaires : Avez-vous déjà eu des pensées suicidaires ? (Oui/Non).
15. nombre_heure_travail_etude : Nombre d'heures de travail ou d'études par jour.
16. stress_financier : Stress financier (échelle ou score).
17. antecedents_familiaux_maladies_mentales : Antécédents familiaux de maladies mentales (Oui/Non).
18. dépression : Dépression (1 pour oui, 0 pour non).

I. PRETRAITEMENT DES DONNEES

- ## Visualisation des données

```
'data.frame':    27901 obs. of 18 variables:
 id                                    : int  2 8 26 30 32 33 52 56 59 62 ...
 sexe                                  : Factor w/ 2 levels "Female","Male": 2 1 2 1 1 2 2 1 2 2 ...
 age                                   : num  33 24 31 28 25 29 30 30 28 31 ...
 ville                                 : Factor w/ 52 levels "3.0","Agra","Ahmedabad",..: 52 4 45 50 17 40 47 7 34 38 ...
 profession                            : Factor w/ 14 levels "Architect","Chef",..: 12 12 12 12 12 12 12 12 12 12 ...
 pression_academique                   : num  5 2 3 3 4 2 3 2 3 2 ...
 pression_liee_au_travail              : num  0 0 0 0 0 0 0 0 0 0 ...
 moyenne_notes                         : num  8.97 5.9 7.03 5.59 8.13 5.7 9.54 8.04 9.79 8.38 ...
 satisfaction_etudes                   : num  2 5 5 2 3 3 4 4 1 3 ...
 satisfaction_travail                  : num  0 0 0 0 0 0 0 0 0 0 ...
 duree_sommeil                         : Factor w/ 5 levels "5-6 hours","7-8 hours",..: 1 1 3 2 1 3 2 3 2 3 ...
 habitudes_alimentaires                : Factor w/ 4 levels "Healthy","Moderate",..: 1 2 1 2 2 1 1 4 2 2 ...
 diplome_suivi                         : Factor w/ 28 levels "B.Arch","B.Com",..: 4 11 6 8 18 28 11 12 3 13 ...
 pensees_suicidaire                    : Factor w/ 2 levels "No","Yes": 2 1 1 2 2 1 1 1 2 2 ...
 nombre_heure_travail_etude            : num  3 3 9 4 1 4 1 0 12 2 ...
 stress_financier                      : num  1 2 1 5 1 1 2 1 3 5 ...
 antecedants_familiaux_maladie_mentale: Factor w/ 2 levels "No","Yes": 1 2 2 2 1 1 1 2 1 1 ...
 depression                            : Factor w/ 2 levels "Non","Oui": 2 1 1 2 1 1 1 1 2 2 ...
```

Ce jeu de données comprend 27901 observations et 18 variables.

1-Conversion des modalités des Données, des types de Données et des variables binaires

```
Id (catégorielle)

Sexe (catégorielle, Homme/femme)

age (entier)

pression_academique (numérique /10)

Pression_liee_au_travail (numérique /10)

Moyenne_notes (float)

satisfaction_etudes (numérique /10)

satisfaction_travail (numérique /10)

pensees_suicidaire (Oui/Non)

nombre_heure_travail_etude (entier)

stress_financier (numérique /10)

antecedants_familiaux_maladie_mentale (Oui/Non)

depression (Oui/Non)
```

Ces variables vous permettront d'explorer et d'analyser divers aspects du mode de vie et de la santé mentale des étudiants, dans le but d'étudier l'association entre ces facteurs et la dépression.

2- Traitement de doublons

```
0 doublons trouvés et supprimés
```

3- Traitement des données manquantes

- **Visualisation des données manquantes**

```
Id                                  0
sexe                                0
age                                 0
ville                               0
profession                          0
..
pensees_suicidaire                  0
nombre_heure_travail_etude          0
stress_financier                    3
antecedants_familiaux_maladie_mentale    0
depression                          0
Length : 18, dtype : int64
```

Une seule variable, stress_financier, présente des valeurs manquantes (3 valeurs manquantes), ce qui représente un faible pourcentage par rapport à l'ensemble des 27,901 observations. Les valeurs manquantes peuvent être imputées ou traitées pour ne pas biaiser les analyses statistiques. Le fait que seules 3 valeurs soient manquantes dans une variable spécifique facilite ce traitement sans impact significatif sur les résultats globaux.

La variable "stress_financier" contient 3 données manquantes.

- **Moyenne et Médiane avant traitement**

```
count    27898.00
mean         3.14
std          1.44
min          1.00
25%          2.00
50%          3.00
75%          4.00
max          5.00
Name: stress_financier, dtype: float64
```

Ces statistiques fournissent une vue d'ensemble du stress financier ressenti par les étudiants, avec une majorité se situant dans la tranche modérée. Elles seront utiles pour explorer les corrélations et l'impact de ce stress sur d'autres variables comme la dépression dans les analyses ultérieures.

- **Moyenne et Médiane après traitement**

```
count    27901.00
mean         3.14
std          1.44
min          1.00
25%          2.00
50%          3.00
75%          4.00
max          5.00
Name : stress_financier, dtype : float64
```

Ces statistiques montrent que l'imputation des valeurs manquantes n'a pas affecté de manière significative la distribution des données sur le stress financier. Les données restent cohérentes et fiables pour les analyses statistiques ultérieures, permettant d'explorer les relations entre le stress financier et d'autres variables, comme la dépression.

- **Visualisation des données manquantes**

```
Valeurs manquantes après l'imputation :
Id                                      0
sexe                                    0
age                                     0
ville                                   0
profession                              0                                    ..
pensees_suicidaire                      0
nombre_heure_travail_etude              0
stress_financier                        0
antecedants_familiaux_maladie_mentale   0
depression                              0
Length: 18, dtype: int64
```

Avec ces données complètes, nous sommes maintenant bien placés pour avancer vers la prochaine étape d'analyse et d'inférence statistique. Nous pourrons explorer les relations entre divers facteurs liés au mode de vie, aux études, et à la dépression chez les étudiants, pour en tirer des conclusions éclairées et significatives.

Les données ainsi traiter, Nous avons des données dans le bon format et ainsi prêt pour les analyses et inférence statistique. Le nouveau jeu de données est:

```
'data.frame':27901 obs. of  18 variables:
  id                                    : int  2 8 26 30 32 33 52 56 59 62 ...
  sexe                                  : Factor w/ 2 levels "Femme","Homme": 2 1 2 1
1 2 2 1 2 2 ...
  age                                   : int  33 24 31 28 25 29 30 30 28 31 ...
  ville                                 : Factor w/ 52 levels "3.0","Agra","Ahmedabad",..:
52 4 45 50 17 40 47 7 34 38 ...
  profession                            : Factor w/ 14 levels "Architecte","Avocat",..:
8 8 8 8 8 8 8 8 8 8 ...
  pression_academique                   : int  5 2 3 3 4 2 3 2 3 2 ...
  pression_liee_au_travail              : int  0 0 0 0 0 0 0 0 0 0 ...
  moyenne_notes                         : Factor w/ 332 levels "0","10","5,03",..: 268
64 135 36 209 45 303 203 316 225 ...
  satisfaction_etudes                   : int  2 5 5 2 3 3 4 4 1 3 ...
  satisfaction_travail                  : int  0 0 0 0 0 0 0 0 0 0 ...
  duree_sommeil                             : Factor w/ 5 levels "5-6 heures","7-8
heures",..: 1 1 4 2 1 4 2 4 2 4 ...
  habitudes_alimentaires                : Factor w/ 4 levels "Autres","Mauvais pour la
santé",..: 4 3 4 3 3 4 4 2 3 3 ...
  diplome_suivi                          : Factor w/ 28 levels "B.Arch","B.Com",..: 4
11 6 8 18 28 11 12 3 13 ...
  pensees_suicidaire                    : Factor w/ 2 levels "Non","Oui": 2 1 1 2 2 1
1 1 2 2 ...
  nombre_heure_travail_etude            : int  3 3 9 4 1 4 1 0 12 2 ...
  stress_financier                      : int  1 2 1 5 1 1 2 1 3 5 ...
  antecedants_familiaux_maladie_mentale: Factor w/ 2 levels "Non","Oui": 1 2 2 2 1 1
1 2 1 1 ...
  depression                            : Factor w/ 2 levels "Non","Oui": 2 1 1 2 1 1
1 1 2 2 ...
```

Le prétraitement des données a permis de :

1. Visualiser et comprendre des données :

- Le jeu de données comprend 27,901 observations et 18 variables.

- Ces variables incluent des facteurs démographiques (âge, sexe, ville), académiques (pression académique, moyenne des notes), et des informations sur la santé mentale (pensées suicidaires, dépression).

2. Convertir des types de données :

- Les variables catégorielles et binaires ont été correctement identifiées et converties en types appropriés (facteurs, int, etc.).

3. Traitement des doublons :

- Aucun doublon trouvé, assurant ainsi l'unicité de chaque enregistrement.

4. Traitement des données manquantes :

- Seules 3 valeurs manquantes ont été identifiées dans la variable "stress_financier".

- Imputation effectuée avec succès, ce qui a permis de conserver la cohérence et l'intégrité des données.

5. Statistiques descriptives :

- Les moyennes et médianes avant et après traitement des valeurs manquantes montrent que les valeurs imputées n'ont pas altéré de manière significative les caractéristiques de cette variable.

Grâce à ces étapes, nous avons maintenant des données nettoyées, cohérentes et prêtes pour les analyses et inférences statistiques.

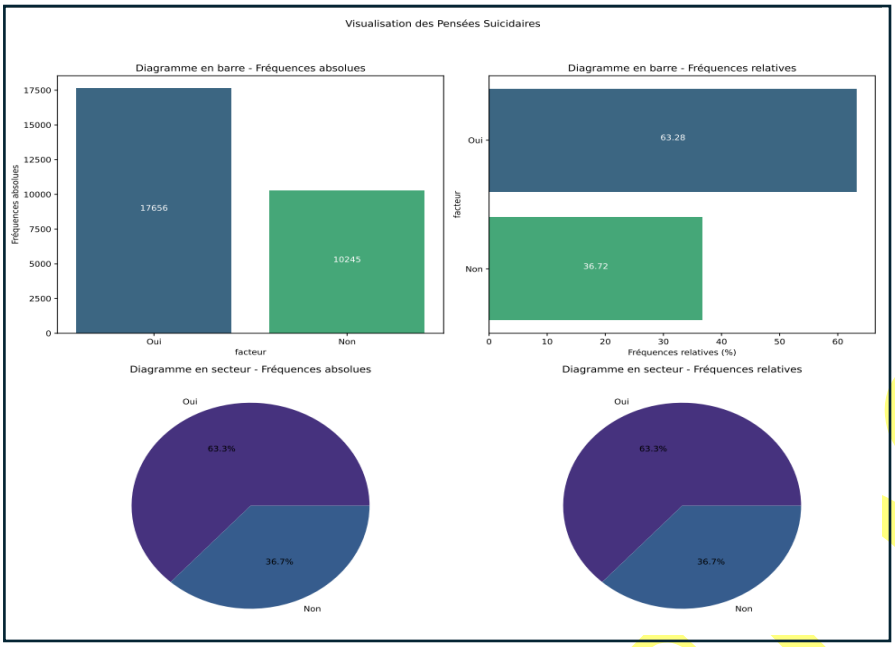
II- ANALYSE ET INFERENCE STATISTIQUE

1. Quel est l'intervalle de confiance pour la proportion d'étudiants ayant déjà eu des pensées suicidaires ?

a) Tableau statistique

pensees_suicidaire	Fréquence	Effectif
Oui	17656	0.63
Non	10245	0.37

b) Graphique variable " Pensees_suicidaire"



c) Estimation proportion

```
Intervalle de confiance à 95% : (0.6271527143195701, 0.6384650054754193)
```

Cet intervalle de confiance à 95% pour la proportion d'étudiants ayant déjà eu des pensées suicidaires nous donne une estimation précise de cette proportion dans la population étudiée. En termes simples, cela signifie que nous sommes 95% certains que la véritable proportion d'étudiants ayant eu des pensées suicidaires se situe entre 62.72% et 63.85%. Ainsi, sur une grande population d'étudiants, on peut dire avec un haut niveau de confiance que la proportion de ceux ayant déjà eu des pensées suicidaires est dans cette fourchette.

2) A) la moyenne et la médiane des heures de travail ou d'études pour les étudiants souffrant de dépression.

a) Extraction des données mettant en évidence les heures de travail ou d'études pour les étudiants souffrant de dépression

	Dépression	Nombre_Heures_Travail_Etude
0	Oui	3
3	Oui	4
8	Oui	12
9	Oui	2
10	Oui	11
.....	.....	.....
27887	Oui	7
27888	Oui	10
27891	Oui	2
27899	Oui	10
27900	Oui	2

b) Estimation

- **Test de normalité**

```
H0 : la distribution suit une loi normale
H1 : la distribution ne suit pas une loi normale
Statistique de test = 0.910375834022112
P-valeur = 3.657302435548303e-70
L'échantillon ne semble pas suivre une distribution normale (rejeter H0). En rejetant H0, nous concluons que la distribution des heures de travail ou d'études pour les étudiants souffrant de dépression ne suit pas une loi normale
```

- **Estimation de la moyenne**

```
Intervalle de confiance à 95% pour la moyenne d'Heures_Travail_Etude : (7.753059194417238, 7.857188112144955)
```

Cet intervalle de confiance à 95% pour la moyenne des heures de travail ou d'études nous donne une estimation précise de cette moyenne dans la population étudiée. En termes simples, cela signifie que nous sommes 95% certains que la véritable moyenne des heures de travail ou d'études pour les étudiants souffrant de dépression se situe entre 7,753 et 7,857 heures. Ainsi, sur une grande population d'étudiants, on peut dire avec un haut niveau de confiance que la moyenne des heures de travail ou d'études pour ceux souffrant de dépression est dans cette fourchette.

- **Estimation de la Médiane**

```
Intervalle de confiance à 95% pour la médiane d'Heures_Travail_Etude_med :
(9.0, 9.0)
```

Cet intervalle de confiance à 95% pour la médiane des heures de travail ou d'études nous donne une estimation précise de cette médiane dans la population étudiée. En termes simples, cela signifie que nous sommes 95% certains que la véritable médiane des heures de travail ou d'études pour les étudiants souffrant de dépression est de 9 heures. Ainsi, sur une grande population d'étudiants, on peut dire avec un haut niveau de confiance que la médiane des heures de travail ou d'études pour ceux souffrant de dépression est de 9 heures.

B) la moyenne et la médiane du stress financier pour les étudiants avec et sans dépression.

a) Extraction de la colonne stress_financier et dépression

– **Visualisation des 5 premières lignes :**

	stress_financier	dépression
0	1	Oui
1	2	Non
2	1	Non
3	5	Oui
4	1	Non

b) Test de normalité

```
H0 : la distribution suit une loi normale
H1 : la distribution ne suit pas une loi normale
Statistique de test = 0.880089841144436
P-valeur = 2.8391351357830143e-88
L'échantillon ne semble pas suivre une distribution normale (rejeter H0)
En rejetant H0, nous concluons que la distribution des heures de travail ou d'études
pour les étudiants souffrant de dépression ne suit pas une loi normale.
```

c) Estimation de la moyenne

```
Intervalle de confiance à 95% pour la moyenne stress_financier_moy :
(3.12339432278413, 3.1569513637504034)
```

Cet intervalle de confiance à 95% pour la moyenne du stress financier nous donne une estimation précise de cette moyenne dans la population étudiée. En termes simples, cela signifie que nous sommes 95% certains que la véritable moyenne du stress financier pour les étudiants se situe entre 3,123 et 3,157. Ainsi, sur une grande population d'étudiants, on peut dire avec un haut niveau de confiance que la moyenne du stress financier est dans cette fourchette.

d) Estimation de la médiane

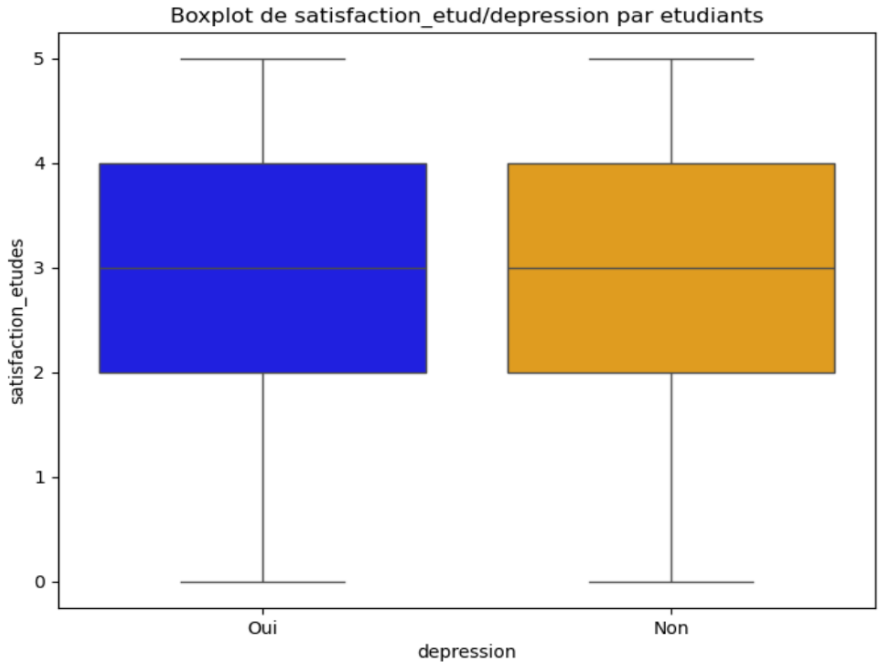
```
Intervalle de confiance à 95% pour la médiane d'stress_financier_med
: (3.0, 3.0)
```

Cet intervalle de confiance à 95% pour la médiane du stress financier nous donne une estimation très précise de cette médiane dans la population étudiée. En termes simples, cela signifie que nous sommes 95% certains que la véritable médiane du stress financier pour les étudiants se situe à 3,0.

3) Différence de moyennes

A) satisfaction des études diffère-t-elle significativement entre les étudiants souffrant de dépression et ceux qui n'en souffrent pas ?

1. Comparer graphiquement les deux sous population



Ce diagramme permet de visualiser comment les niveaux de satisfaction des études diffèrent entre les étudiants souffrant de dépression et ceux qui n'en souffrent pas.
Le groupe "Oui" est représenté par une boîte bleue.
Le groupe "Non" est représenté par une boîte orange.
La médiane de la satisfaction des études semble légèrement plus élevée pour le groupe "Oui" que pour le groupe "Non".
Les quartiles montrent une distribution similaire de satisfaction des études pour les deux groupes, bien que le groupe "Non" ait une légère variation plus large.

2. Estimation des statistiques de base par sous-population

Dépression		Moyenne par dépression
Non		3.215564
Oui		2.751469

Name : satisfaction_etudes, dtype : float64

En moyenne, la satisfaction des études est plus faible chez les étudiants souffrant de dépression par rapport à ceux qui n'en souffrent pas.

3. Test de Shapiro pour vérifier la normalité par sous-population

$H0$: Les distributions suivent la loi normale

$H1$: Les distributions ne suivent pas la loi normale

Test de Shapiro pour les Oui : ShapiroResult (statistic=0.8928755891364766, pvalue=6.177610170128729e-74)

Pvalue < 0.05 donc on rejette H0 donc Les distributions ne suivent pas la loi normale

Test de Shapiro pour les Non : ShapiroResult (statistic=0.8973825038048788, pvalue=8.085526933724967e-66)

Pvalue < 0.05 donc on rejette H0 donc Les distributions ne suivent pas la loi normale.

4. Test de l'égalité des variances

$H0$: Les variances sont égales

$H1$: Les variances ne sont pas égales

Statistic T : 4.541608079214673 P-value : 0.03308050143618153

Conclusion : la p-value < 0.05 donc on rejette $H0$, les variances sont significativement différentes.

Les distributions ne suivant pas une loi normale et l'égalité des variances n'étant pas respecté, Nous procéderons à un test Non paramétrique qu'est le Test de Wilcoxon-Mann-Whitney.

5. Test égalité des moyennes

$H0$: Les moyennes sont égales

$H1$: Les moyennes ne sont pas égales

Statistique observée : -0.464095054440409

P-Value : 0.0. La différence de satisfaction des études entre les deux groupes est statistiquement significative (p < 0.05). On rejette H0 et on conclut que la différence de satisfaction des études entre les étudiants souffrant de dépression et ceux qui n'en souffrent pas est statistiquement significative avec un niveau de signification de 5%.

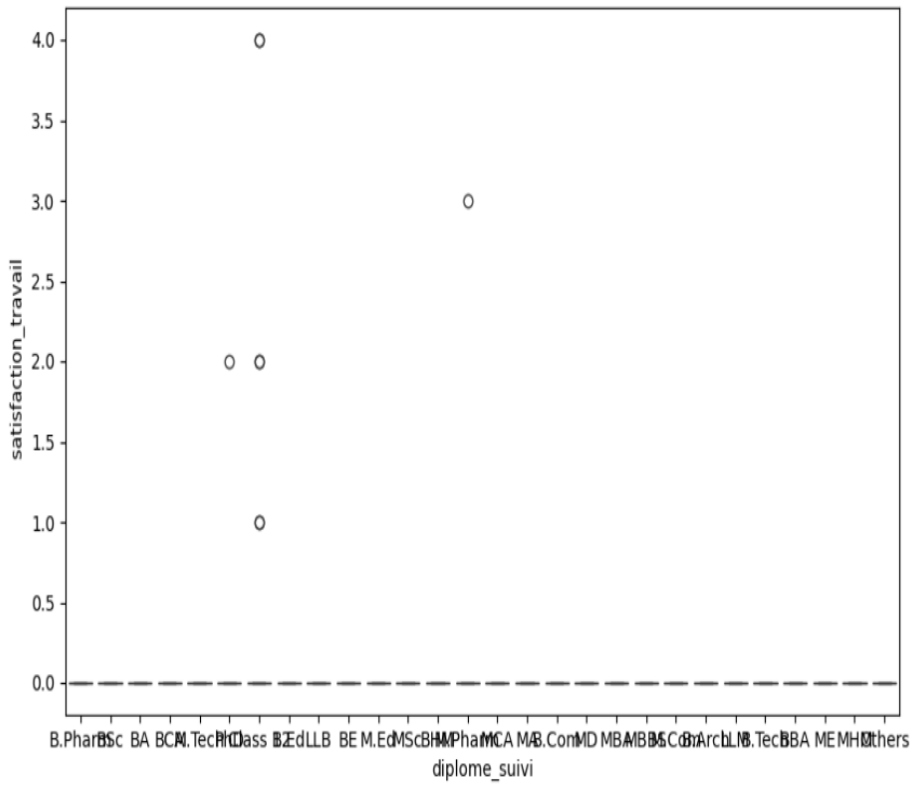
$H0$: Les moyennes sont égales
$H1$: La moyenne satis du groupe "Oui" > La moyenne satis du groupe "Non"

Statistique observée : -0.464095054440409
P-Value : 1.0. La moyenne de satisfaction des études du groupe 'Oui' n'est pas significativement plus grande que celle du groupe 'Non' (p > 0.05). On rejette H0 et on conclut que la différence de satisfaction des études entre les étudiants souffrant de dépression et ceux qui n'en souffrent pas étant statistiquement significative avec un niveau de signification de 5% et La moyenne de satisfaction des études du groupe 'Oui' n'étant pas significativement plus grande que celle du groupe 'Non'.

Alors on peut conclure que La moyenne de satisfaction des études du groupe 'Oui' est significativement plus petite que celle du groupe 'Non'.

B) Les niveaux de satisfaction au travail diffèrent-ils significativement selon le diplôme suivi ?

1. **Comparer graphiquement les sous-populations**



2. Estimer les statistiques de base par sous-population

```
diplome_suivi
B.Arch       0.000000
B.Com        0.000000
B.Ed         0.000000
B.Pharm      0.000000
B.Tech       0.000000
BA           0.000000
BBA          0.000000
BCA          0.000000
BE           0.000000
BHM          0.000000
BSc          0.000000
Class 12     0.002303
LLB          0.000000
LLM          0.000000
M.Com        0.000000
M.Ed         0.000000
M.Pharm      0.005155
M.Tech       0.000000
MA           0.000000
MBA          0.000000
MBBS         0.000000
MCA          0.000000
MD           0.000000
ME           0.000000
MHM          0.000000
MSc          0.000000
Others       0.000000
PhD          0.003831
Name: satisfaction_travail, dtype: float64
```

Class 12 **: La moyenne de satisfaction au travail est de 0.002303. Bien que cette valeur soit très faible, elle montre qu'il y a une légère satisfaction exprimée parmi les étudiants de ce groupe.**

M.Pharm : La moyenne de satisfaction au travail pour ce groupe est de 0.005155, légèrement plus élevée que pour Class 12, ce qui montre une présence très modeste de satisfaction au travail parmi les étudiants de M.Pharm.

PhD : La moyenne de satisfaction au travail est de 0.003831. Bien que faible, cette valeur est plus élevée que pour les diplômes avec 0.000000 mais inférieure à M.Pharm.

3. Tester la normalité des données dans chaque sous-population

*H*0 : Les distributions suivent la loi normale

*H*1 : Les distributions ne suivent pas la loi normale

```
Test de Shapiro-Wilk pour le groupe PhD :
Statistique=0.02039, P-value=0.00000
Conclusion : Les données ne suivent pas une distribution normale.

Test de Shapiro-Wilk pour le groupe Class 12 :
Statistique=0.00948, P-value=0.00000
Conclusion : Les données ne suivent pas une distribution normale.

Test de Shapiro-Wilk pour le groupe M. Pharm :
Statistique=0.01865, P-value=0.00000
Conclusion : Les données ne suivent pas une distribution normale.
```

4. Tester l'égalité des variances

H0 : : Les variances sont égales

H1 : Les variances ne sont pas égales

Bartlett test: Statistic=inf, P-value=0.00000

P-value < 0.05 (5%) : On rejette l'hypothèse nulle et on conclut que les variances des groupes sont significativement différentes. Cela signifie qu'il y a une hétérogénéité des variances entre les groupes.

5. Tester la significativité du facteur : Tester l'égalité des moyennes

*H*0 : Toutes les moyennes sont égales

*H*1 : Au moins une des moyennes est différente des autres

Kruskal-Wallis test: Statistic=25.33084, P-value=0.55593

P-value > 0.05 (5%) : On ne peut rejeter l'hypothèse nulle. Il n'y a pas de preuve suffisante pour dire que les distributions des groupes sont différentes.

Les données suggèrent qu'il n'y a pas de différence statistiquement significative dans les niveaux de satisfaction au travail entre les différents groupes de diplômes suivis. Les variations observées dans les moyennes de satisfaction au travail pour les différents diplômes ne sont pas assez importantes pour être considérées comme différentes d'un point de vue statistique. Cela signifie que, selon les données et les tests effectués, la satisfaction au travail des étudiants ne dépend pas significativement du diplôme qu'ils suivent.

4) INDEPENDANCE

A) La dépression est-elle indépendante des habitudes alimentaires (saines/modérées) ?

a. Extraction des données

	habitudes_alimentaires	dépression
0	sain	Oui
1	Modéré	Non
2	sain	Non
3	Modéré	Oui
4	Modéré	Non
…	……	……
27894	sain	Non
27895	Modéré	Non
27897	sain	Non
27899	sain	Oui
27900	sain	Oui

Ce tableau contient les modalités de la variable « habitudes_alimentaires » et la variable « dépression » où l'on a également 27901 observations de l'ensemble de ces données.

b. Construction du tableau de contingence

dépression / Habitudes_alimentaires	Non	Oui
Modéré	4363	5558
Sain	4178	3473

c. Vérifions la règle de Cochran : cette règle permet de vérifier si 80% des effectifs théoriques sont ≥ 5

Valeurs attendues sous forme de tableau :

dépression / Habitudes_alimentaires	Non	Oui
Modéré	4822.175108	5098.824892
Sain	3718.824892	3932.175108

Pourcentage des effectifs théoriques ≥ 5 : 100.00%
La règle de Cochran est respectée.

La règle de Cochran étant respectée nous procéderons au test de khi-deux.

d. Test de khi-deux

H0: p1=p2=.....= pn
H1: pi≠pj
Chi2 value : 194.96457710906392
P-value : 2.6226631443018764e-44

P-value < 0.05 donc on rejette H0 et on conclut que La dépression n'est pas indépendante des habitudes alimentaires (saines/modérées). On peut conclure qu'il existe une association statistiquement significative entre les habitudes alimentaires et la dépression. En d'autres termes, les données suggèrent que le type d'habitudes alimentaires a un impact sur les niveaux de dépression.

B) La durée du sommeil (par exemple, moins de 5 heures, 5-6 heures, 7-8 heures) est-elle indépendante de la dépression ?

a. Extraction des données

	Duree_sommeil	dépression
0	5-6 heures	Oui
1	5-6 heures	Non
2	Moins de 5 heures	Non
3	7-8 heures	Oui
4	5-6 heures	Non
….		….
27896	5-6 heures	Non
27897	Moins de 5 heures	Non
27898	5-6 heures	Non
27899	Moins de 5 heures	Oui
27900	Moins de 5 heures	Oui

[27901 rows x 2 columns]

Ce tableau contient les modalités de la variable « Duree_sommeil » et la variable « dépression » où l'on a également 27901 observations de l'ensemble de ces données.

b. Construction du tableau de contingence

	dépression	
duree_sommeil	Non	Oui
5-6 heures	2666	3517
7-8 heures	2975	4371
Autres	9	9
moins de 5 heures	2949	5361
plus de 8 heures	2966	3078

- **3517 et 2666 Etudiants qui respectivement souffrent et ne souffrent pas de dépression ont une durée de 5 à 6 heures de sommeil.**
- **4371 et 2975 Etudiants qui respectivement souffrent et ne souffrent pas de dépression ont une durée de 7 à 8 heures de sommeil.**
- **5361 et 2949 Etudiants qui respectivement souffrent et ne souffrent pas de dépression ont moins de 5 heures de sommeil.**
- **3078 et 2966 Etudiants qui respectivement souffrent et ne souffrent pas de dépression ont plus de 8 heures de sommeil.**
- **9 et 9 Etudiants qui respectivement souffrent et ne souffrent pas de dépression ont d'autres heures de sommeil.**

c. **Vérifions la règle de Cochran : cette règle permet de vérifier si 80% des effectifs théoriques sont ≥ 5.**

Valeurs attendues sous forme de tableau :

	dépression	
duree_sommeil	Non	Oui
5-6 heures	2562.861367	3620.138633
7-8 heures	3044.926347	4301.073653
Autres	7.461023	10.538977
moins de 5 heures	3444.505573	4865.494427
plus de 8 heures	2505.245690	3538.754310

Pourcentage des effectifs théoriques ≥ 5 : 100%
La règle de Cochran est respectée.

d. **Test de khi-deux**

H0: p1=p2=……= pn
H1: pi≠pj
Chi2 value : 276.8483796551745
P-value : 1.065310789284643e-58

Une valeur p très petite (comme 1.065310789284643e-58) indique que nous pouvons rejeter l'hypothèse nulle avec une grande confiance. L'hypothèse nulle dans ce test est que la durée du sommeil et la dépression sont indépendantes. Puisque la p-value est extrêmement petite, cela suggère qu'il y a une relation significative entre la durée du sommeil et la dépression. En d'autres termes, selon les données de notre test Chi2, nous pouvons conclure que la durée du sommeil n'est pas indépendante de la dépression. Cela signifie qu'il existe une association entre la quantité de sommeil et la présence de symptômes dépressifs.

III- CONCLUSION

Les résultats de cette étude montrent que la dépression chez les étudiants est influencée par plusieurs facteurs liés à leur mode de vie, à leurs études et à leurs antécédents personnels. La proportion d'étudiants ayant déjà eu des pensées suicidaires est alarmante, ce qui souligne la nécessité de prendre des mesures préventives et d'intervenir de manière proactive. Les analyses statistiques révèlent des associations significatives entre la dépression et des variables telles que les habitudes alimentaires, la durée du sommeil et la satisfaction aux études.

Les étudiants souffrant de dépression tendent à travailler ou à étudier en moyenne plus d'heures que leurs pairs non déprimés, et ils ressentent également un niveau de stress financier plus élevé. La satisfaction des études est nettement plus faible chez les étudiants déprimés, bien que la satisfaction au travail ne semble pas être influencée de manière significative par le diplôme suivi. Enfin, des habitudes alimentaires saines et une durée de sommeil adéquate semblent jouer un rôle dans la prévention de la dépression.

Recommandations

1. Programmes de soutien psychologique : Mettre en place des services de conseil et de soutien psychologique facilement accessibles pour les étudiants. Ces programmes devraient inclure des séances de thérapie, des ateliers de gestion du stress et des groupes de soutien.

2. Promotion de modes de vie sains : Encourager les étudiants à adopter des habitudes alimentaires saines et à maintenir une durée de sommeil adéquate. Des campagnes de sensibilisation et des ateliers sur la nutrition et l'importance du sommeil peuvent être utiles.

3. Gestion de la charge de travail : Aider les étudiants à équilibrer leur charge de travail et leurs études. Des sessions de gestion du temps et des cours sur les techniques d'étude efficaces peuvent aider à réduire la pression académique.

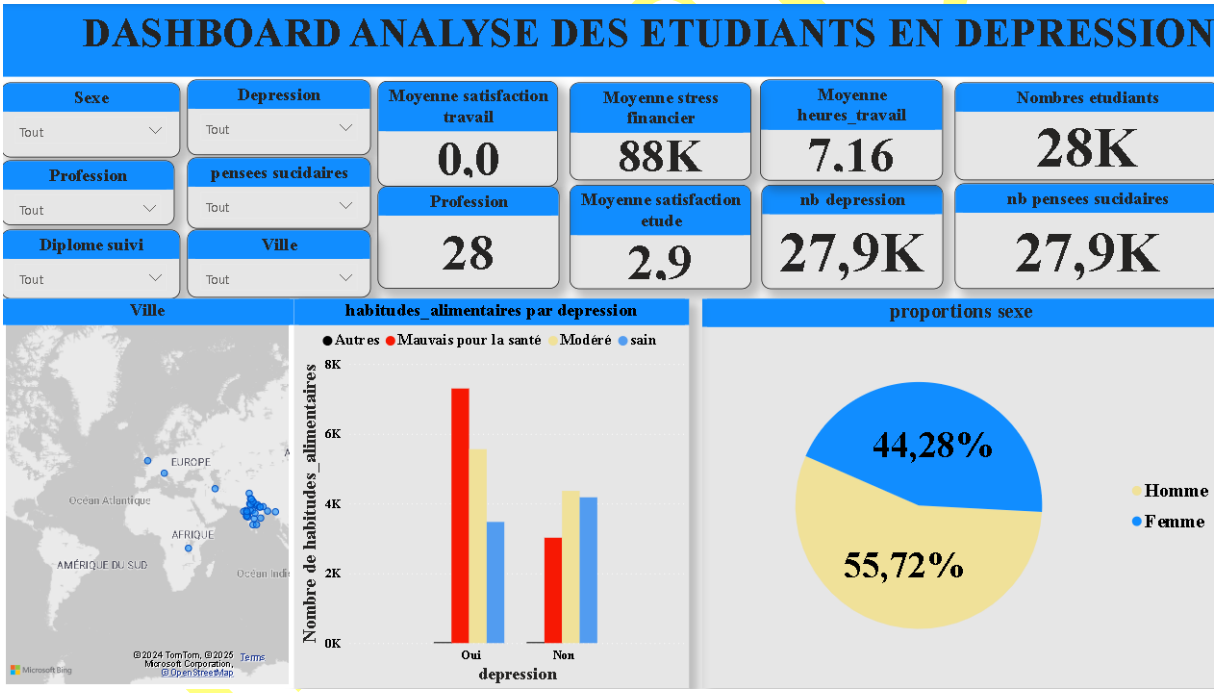
4. Soutien financier : Fournir des aides financières et des conseils sur la gestion des finances pour réduire le stress financier des étudiants. Les bourses

d'études, les subventions et les programmes de prêt à taux réduit peuvent alléger le fardeau financier.

5. Création d'un environnement de soutien : Favoriser une culture de soutien et d'inclusivité sur le campus. Encourager les interactions sociales positives et la participation à des activités communautaires pour renforcer le sentiment d'appartenance et de soutien mutuel.

6. Suivi et évaluation : Mettre en place des systèmes de suivi et d'évaluation pour mesurer l'efficacité des interventions et des programmes de soutien. Ajuster les stratégies en fonction des résultats obtenus pour garantir une amélioration continue du bien-être des étudiants.

POWER BI



CODE PYTHON ET POWER-QUERY

I. PRETRAITEMENT DES DONNEES

-
```
import pandas as pd
file_path = "C:/Users/HP/Downloads/INSSEDS/cours/Inference statistique/Mini projet stat inferentielle/Student_Depression.csv"
etud_dep = pd.read_csv(file_path, sep=",")
etud_dep
```

```
- let
    Source = Csv.Document(File.Contents("C:\Users\HP\Downloads\INSSEDS\cours\Inference statistique\depression etudiant.csv"),[Delimiter=";", Columns=18, Encoding=1252, QuoteStyle=QuoteStyle.None]),
    #"En-têtes promus" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    #"Valeur remplacée" = Table.ReplaceValue(#"En-têtes promus",".",",",Replacer.ReplaceText,{"age"}),
    #"Type modifié" = Table.TransformColumnTypes(#"Valeur remplacée",{{"age", Int64.Type}}),
    #"Valeur remplacée1" = Table.ReplaceValue(#"Type modifié",".",",",Replacer.ReplaceText,{"pression_academique"}),
    #"Type modifié1" = Table.TransformColumnTypes(#"Valeur remplacée1",{{"pression_academique", Int64.Type}}),
    #"Valeur remplacée2" = Table.ReplaceValue(#"Type modifié1",".",",",Replacer.ReplaceText,{"pression_liee_au_travail"}),
    #"Type modifié2" = Table.TransformColumnTypes(#"Valeur remplacée2",{{"pression_liee_au_travail", Int64.Type}}),
    #"Valeur remplacée3" = Table.ReplaceValue(#"Type modifié2",".",",",Replacer.ReplaceText,{"moyenne_notes"}),
    #"Type modifié3" = Table.TransformColumnTypes(#"Valeur remplacée3",{{"moyenne_notes", type number}}),
    #"Valeur remplacée4" = Table.ReplaceValue(#"Type modifié3",".",",",Replacer.ReplaceText,{"satisfaction_etudes"}),
    #"Type modifié4" = Table.TransformColumnTypes(#"Valeur remplacée4",{{"satisfaction_etudes", Int64.Type}}),
    #"Valeur remplacée5" = Table.ReplaceValue(#"Type modifié4",".",",",Replacer.ReplaceText,{" satisfaction_travail"}),
    #"Type modifié5" = Table.TransformColumnTypes(#"Valeur remplacée5",{{" satisfaction_travail", Int64.Type}}),
    #"Valeur remplacée6" = Table.ReplaceValue(#"Type modifié5","Yes","Oui",Replacer.ReplaceText,{"pensees_suicidaire"}),
    #"Valeur remplacée7" = Table.ReplaceValue(#"Valeur remplacée6","No","Non",Replacer.ReplaceText,{"pensees_suicidaire"}),
    #"Valeur remplacée8" = Table.ReplaceValue(#"Valeur remplacée7",".",",",Replacer.ReplaceText,{"nombre_heure_travail_etude"}),
    #"Type modifié6" = Table.TransformColumnTypes(#"Valeur remplacée8",{{"nombre_heure_travail_etude", Int64.Type}}),
    #"Valeur remplacée9" = Table.ReplaceValue(#"Type modifié6",".",",",Replacer.ReplaceText,{"stress_financier"}),
    #"Type modifié7" = Table.TransformColumnTypes(#"Valeur remplacée9",{{"stress_financier", Int64.Type}}),
    #"Valeur remplacée10" = Table.ReplaceValue(#"Type modifié7","Yes","Oui",Replacer.ReplaceText,{"antecedants_familiaux_maladie_mentale"}),
    #"Valeur remplacée11" = Table.ReplaceValue(#"Valeur remplacée10","No","Non",Replacer.ReplaceText,{"antecedants_familiaux_maladie_mentale"}),
    #"Valeur remplacée12" = Table.ReplaceValue(#"Valeur remplacée11","hours","heures",Replacer.ReplaceText,{"duree_sommeil"}),
    #"Valeur remplacée13" = Table.ReplaceValue(#"Valeur remplacée12","Less than 5 heures","moins de 5 heures",Replacer.ReplaceText,{"duree_sommeil"}),
    #"Valeur remplacée14" = Table.ReplaceValue(#"Valeur remplacée13","More than 8 heures","plus de 8 heures",Replacer.ReplaceText,{"duree_sommeil"}),
    #"Valeur remplacée15" = Table.ReplaceValue(#"Valeur remplacée14","Others","Autres",Replacer.ReplaceText,{"duree_sommeil"}),
```

```
    #"Valeur remplacée16" = Table.ReplaceValue(#"Valeur remplacée15","Healthy","sain",Replacer.ReplaceText,{"habitudes_alimentaires"}),
    #"Valeur remplacée17" = Table.ReplaceValue(#"Valeur remplacée16","Moderate","Modéré",Replacer.ReplaceText,{"habitudes_alimentaires"}),
    #"Valeur remplacée18" = Table.ReplaceValue(#"Valeur remplacée17","Others","Autres",Replacer.ReplaceText,{"habitudes_alimentaires"}),
    #"Valeur remplacée19" = Table.ReplaceValue(#"Valeur remplacée18","Unhealthy","Mauvais pour la santé",Replacer.ReplaceText,{"habitudes_alimentaires"}),
    #"Valeur remplacée20" = Table.ReplaceValue(#"Valeur remplacée19","Male","Homme",Replacer.ReplaceText,{"sexe"}),
    #"Valeur remplacée21" = Table.ReplaceValue(#"Valeur remplacée20","Female","Femme",Replacer.ReplaceText,{"sexe"}),
    #"Valeur remplacée22" = Table.ReplaceValue(#"Valeur remplacée21","Architect","Architecte",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée23" = Table.ReplaceValue(#"Valeur remplacée22","Civil Engineer","Ingénieur Civil",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée24" = Table.ReplaceValue(#"Valeur remplacée23","Content Writer","Rédacteur de contenu",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée25" = Table.ReplaceValue(#"Valeur remplacée24","Digital Marketer","Responsable marketing Digital",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée26" = Table.ReplaceValue(#"Valeur remplacée25","Doctor","Medecin",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée27" = Table.ReplaceValue(#"Valeur remplacée26","Educational Consultant","Consultant éducatif",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée28" = Table.ReplaceValue(#"Valeur remplacée27","Lawyer","Avocat",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée29" = Table.ReplaceValue(#"Valeur remplacée28","Pharmacist","Pharmacien",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée30" = Table.ReplaceValue(#"Valeur remplacée29","Student","Etudiant",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée31" = Table.ReplaceValue(#"Valeur remplacée30","Teacher","Enseignant",Replacer.ReplaceText,{"profession"}),
    #"Valeur remplacée32" = Table.ReplaceValue(#"Valeur remplacée31","UX/UI Designer","Designer UX/UI",Replacer.ReplaceText,{"profession"}),
    #"Doublons supprimés" = Table.Distinct(#"Valeur remplacée32", {"id"}),
    #"Lignes vides supprimées" = Table.SelectRows(#"Doublons supprimés", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null})))
in
    #"Lignes vides supprimées"
```

```
- print(etud_dep.isnull().sum())

- etud_dep['stress_financier'].describe()

- # Imputation par la médiane
  for column in etud_dep.select_dtypes(include=['float64', 'int64']).columns:
      median_value = etud_dep[column].median()
      etud_dep[column].fillna(median_value, inplace=True)

  # Afficher les valeurs manquantes après l'imputation
  print("\nValeurs manquantes après l'imputation :")
  print(etud_dep.isnull().sum())
```

- ```
  etud_dep['stress_financier'].describe()
  ```

- ```
  import pandas as pd
  file_path = "C:/Users/HP/Downloads/INSSEDS/cours/Inference statistique/Mini projet stat inferentielle/Depression etudiant.csv"
  Dep_etud = pd.read_csv(file_path, sep=";")
  Dep_etud
  ```

II - Analyse et Inférence statistique

- ```
  import pandas as pd

  def akposso_ql_tableau(column):
      # Création d'une série contenant les fréquences absolues
      T = column.value_counts()

      # Conversion en DataFrame
      tab = pd.DataFrame({'Effectif': T, 'Frequence': T / T.sum()})

      return tab

  # Générer le tableau statistique pour la colonne 'pensees_suicidaire'
  tableau = akposso_ql_tableau(Dep_etud['pensees_suicidaire'])

  # Afficher le tableau
  print(tableau)
  ```

- ```
  import pandas as pd
  import matplotlib.pyplot as plt
  import seaborn as sns

  def akposso_ql_graph(facteur, output_file):
  # Création d'un DataFrame contenant les fréquences absolues et relatives de chaque modalité
  df = pd.DataFrame(pd.value_counts(facteur).reset_index())
  df.columns = ['facteur', 'Freq']
  df['freq_relatives'] = round(100 * df['Freq'] / df['Freq'].sum(), 2)

  # Créer une figure avec plusieurs sous-graphes
  fig, axes = plt.subplots(2, 2, figsize=(14, 12))
  fig.suptitle('Visualisation des Pensées Suicidaires')

  # Diagramme en barre vertical avec les fréquences absolues
      sns.barplot(ax=axes[0, 0], x='facteur', y='Freq', data=df, palette='viridis')
  axes[0, 0].set_title("Diagramme en barre - Fréquences absolues")
  axes[0, 0].set_ylabel("Fréquences absolues")
  for index, row in df.iterrows():
          axes[0, 0].text(row.name, row.Freq/2, row.Freq, color='white', ha="center")

  # Diagramme en barre horizontal avec les fréquences relatives
      sns.barplot(ax=axes[0, 1], x='freq_relatives', y='facteur', data=df, palette='viridis', orient='h')
  axes[0, 1].set_title("Diagramme en barre - Fréquences relatives")
  ```

```
axes[0, 1].set_xlabel("Fréquences relatives (%)")
for index, row in df.iterrows():
          axes[0, 1].text(row.freq_relatives/2, row.name, row.freq_rela-
  tives, color='white', ha="center")

# Diagramme en secteur avec les fréquences absolues
      axes[1, 0].pie(df['Freq'], labels=df['facteur'],autopct='%1.1f%%',
  colors=sns.color_palette('viridis'))
axes[1, 0].set_title("Diagramme en secteur - Fréquences absolues")

# Diagramme en secteur avec les fréquences relatives
      axes[1, 1].pie(df['freq_relatives'], labels=df['facteur'], auto-
  pct='%1.1f%%', colors=sns.color_palette('viridis'))
axes[1, 1].set_title("Diagramme en secteur - Fréquences relatives")

plt.tight_layout(rect=[0, 0.03, 1, 0.95])

# Enregistrer le graphique en PDF
plt.savefig(output_file, format='pdf')

# Afficher le graphique
plt.show()

  # Utiliser la fonction pour générer et enregistrer les graphiques en
  PDF
  akposso_ql_graph(Dep_etud['pensees_suicidaire'],"resultat_graph.pdf")

-  # L'intervalle de confiance d'un sex-ratio sur 27901individus est:
  # Condition : nb : nb_echec et nb_succes > 10
  nb_succes = 17656
  nb_essais = 27901
  nb_echecs = nb_essais - nb_succes

  import scipy.stats as stats
  import numpy as np
  def binomial_confidence_interval(succes, essais, alpha=0.05):
  """
      Calcule l'intervalle de confiance pour une proportion à l'aide
  d'un test
  binomial.

  :param succes: Nombre de succès.
  :param essais: Nombre total d'essais.
  :param alpha: Niveau de signification (1 - niveau de confiance).

  :return: Intervalle de confiance bas et haut.
  """
  # Calcul de la proportion
  p = succes / essais
  # Z-score pour le niveau de confiance donné (par exemple, 1.96 pour
  95%)
  z = stats.norm.ppf(1 - alpha/2)

  # Largeur de l'intervalle de confiance
  width = z * np.sqrt(p*(1-p) / essais)
  return p - width, p + width
```

```
successes = 17656
trials = 27901
conf_interval = binomial_confidence_interval(successes, trials)
print(f"Intervalle de confiance à 95% : {conf_interval}")
```

-
```
import pandas as pd

# Filtrer les étudiants souffrant de dépression
etudiants_depression = Dep_etud[Dep_etud['depression'] == 'Oui']

# Créer un tableau avec les résultats
tableau_depression = pd.DataFrame({
'Depression': etudiants_depression['depression'],
      'Nombre_Heures_Travail_Etude':                    etudiants_depres-
sion['nombre_heure_travail_etude']
})

# Afficher le tableau
print(tableau_depression)
```

-
```
from scipy.stats import shapiro
# Effectuer le test de Shapiro-Wilk sur la colonne 'age'
statistique, p_valeur = shapiro(tableau_depression['Nombre_Heures_Tra-
vail_Etude'].dropna())
print(f"Statistique de test = {statistique}")
print(f"P-valeur = {p_valeur}")
# Interprétation
alpha = 0.05
if p_valeur > alpha:
 print("L'échantillon semble suivre une distribution normale (ne pas
rejeter H0)")
else:
 print("L'échantillon ne semble pas suivre une distribution normale
(rejeter H0)")
```

-
```
import numpy as np
 from sklearn.utils import resample
 def  bootstrap_confidence_interval(data,  num_bootstraps=1000,  al-
pha=0.05, stat_function=np.mean):
  """
  Retourne un intervalle de confiance bootstrap pour une statistique
donnée (par défaut : la moyenne).

  :param data: Données à échantillonner.
  :param num_bootstraps: Nombre d'échantillons bootstrap à créer.
  :param alpha: Niveau pour l'intervalle de confiance (par exemple,
alpha = 0.05 pour un IC à 95%).
  :param stat_function: Fonction statistique à appliquer sur les échan-
tillons bootstrap.

  :return: Un tuple représentant l'intervalle de confiance bas et haut.
  """

  # Obtenir des statistiques à partir d'échantillons bootstrap
```

```
  bootstrap_samples = [resample(data, replace=True, n_samples=len(data)) for _ in
 range(num_bootstraps)]
  bootstrap_estimates = [stat_function(sample) for sample in bootstrap_samples]

  # Calculer l'intervalle de confiance à partir des estimations bootstrap
  lower = np.percentile(bootstrap_estimates, 100*alpha/2.)
  upper = np.percentile(bootstrap_estimates, 100*(1-alpha/2.))

  return (lower, upper)
 Nombre_Heures_Travail_Etude_moy = tableau_depression['Nombre_Heures_Travail_Etude'].dropna().values
 confidence_interval = bootstrap_confidence_interval(Nombre_Heures_Travail_Etude_moy)
 print(f"Intervalle de confiance à 95% pour la moyenne d'Heures_Travail_Etude : {confidence_interval}")
```

- ```
  def bootstrap_confidence_interval_for_median(data, num_bootstraps=1000, alpha=0.05):
    """
    Retourne un intervalle de confiance bootstrap pour la médiane.

    :param data: Données à échantillonner.
    :param num_bootstraps: Nombre d'échantillons bootstrap à créer.
    :param alpha: Niveau pour l'intervalle de confiance (par exemple, alpha = 0.05 pour un IC à
     95%).

    :return: Un tuple représentant l'intervalle de confiance bas et haut.
    """

    # Obtenir des médianes à partir d'échantillons bootstrap
    bootstrap_samples = [resample(data, replace=True, n_samples=len(data)) for _ in
     range(num_bootstraps)]
    bootstrap_medians = [np.median(sample) for sample in bootstrap_samples]

    # Calculer l'intervalle de confiance à partir des médianes bootstrap
    lower = np.percentile(bootstrap_medians, 100*alpha/2.)
    upper = np.percentile(bootstrap_medians, 100*(1-alpha/2.))

    return (lower, upper)
   Nombre_Heures_Travail_Etude_med = tableau_depression['Nombre_Heures_Travail_Etude'].dropna().values
   confidence_interval_for_median = bootstrap_confidence_interval_for_median(Nombre_Heures_Travail_Etude_med)
   print(f"Intervalle de confiance à 95% pour la médiane d'Heures_Travail_Etude_med : {confidence_interval_for_median}")
  ```

- ```
  import pandas as pd
   # Extraire deux colonnes spécifiques
   colonnes_extraites = Dep_etud[['stress_financier', 'depression']]
  ```

```
 # Afficher les colonnes extraites
 print(colonnes_extraites.head())  # Affiche les 5 premières lignes par
défaut
```

- ```
  from scipy.stats import shapiro
   # Effectuer le test de Shapiro-Wilk sur la colonne 'age'
   statistique,  p_valeur  =  shapiro(colonnes_extraites['stress_finan-
  cier'].dropna())
   print(f"Statistique de test = {statistique}")
   print(f"P-valeur = {p_valeur}")
   # Interprétation
   alpha = 0.05
   if p_valeur > alpha:
    print("L'échantillon semble suivre une distribution normale (ne pas
  rejeter H0)")
   else:
    print("L'échantillon ne semble pas suivre une distribution normale
  (rejeter H0)")
  ```

- ```
  import numpy as np
   from sklearn.utils import resample
   def  bootstrap_confidence_interval(data,  num_bootstraps=1000,  al-
  pha=0.05, stat_function=np.mean):
    """
    Retourne un intervalle de confiance bootstrap pour une statistique
  donnée (par défaut : la moyenne).

    :param data: Données à échantillonner.
    :param num_bootstraps: Nombre d'échantillons bootstrap à créer.
    :param alpha: Niveau pour l'intervalle de confiance (par exemple,
  alpha = 0.05 pour un IC à 95%).
    :param stat_function: Fonction statistique à appliquer sur les échan-
  tillons bootstrap.

    :return: Un tuple représentant l'intervalle de confiance bas et haut.
    """

    # Obtenir des statistiques à partir d'échantillons bootstrap
    bootstrap_samples  =  [resample(data,  replace=True,  n_sam-
  ples=len(data)) for _ in
   range(num_bootstraps)]
    bootstrap_estimates = [stat_function(sample) for sample in boot-
  strap_samples]

    # Calculer l'intervalle de confiance à partir des estimations boot-
  strap
    lower = np.percentile(bootstrap_estimates, 100*alpha/2.)
    upper = np.percentile(bootstrap_estimates, 100*(1-alpha/2.))

    return (lower, upper)
   stress_financier_moy      =      colonnes_extraites['stress_finan-
  cier'].dropna().values
   confidence_interval  =  bootstrap_confidence_interval(stress_finan-
  cier_moy)
  ```

```
print(f"Intervalle de confiance à 95% pour la moyenne d'stress_financier_moy : {confidence_interval}")


- def bootstrap_confidence_interval_for_median(data, num_bootstraps=1000, alpha=0.05):
    """
    Retourne un intervalle de confiance bootstrap pour la médiane.

    :param data: Données à échantillonner.
    :param num_bootstraps: Nombre d'échantillons bootstrap à créer.
    :param alpha: Niveau pour l'intervalle de confiance (par exemple, alpha = 0.05 pour un IC à
    95%).

    :return: Un tuple représentant l'intervalle de confiance bas et haut.
    """

    # Obtenir des médianes à partir d'échantillons bootstrap
    bootstrap_samples = [resample(data, replace=True, n_samples=len(data)) for _ in
    range(num_bootstraps)]
    bootstrap_medians = [np.median(sample) for sample in bootstrap_samples]

    # Calculer l'intervalle de confiance à partir des médianes bootstrap
    lower = np.percentile(bootstrap_medians, 100*alpha/2.)
    upper = np.percentile(bootstrap_medians, 100*(1-alpha/2.))

    return (lower, upper)
  stress_financier_med = colonnes_extraites['stress_financier'].dropna().values
  confidence_interval_for_median = bootstrap_confidence_interval_for_median(stress_financier_med )
  print(f"Intervalle de confiance à 95% pour la médiane d'stress_financier_med  : {confidence_interval_for_median}")


- import pandas as pd
  import numpy as np
  import matplotlib.pyplot as plt
  import seaborn as sns
  from scipy.stats import ttest_ind, levene, shapiro
  file_path = "C:/Users/HP/Downloads/INSSEDS/cours/Inference statistique/Mini projet stat inferentielle/Depression etudiant.csv"
  Dep_etud = pd.read_csv(file_path, sep=";")
  # 1ère étape : Comparaison graphique des deux sous-populations
  plt.figure(figsize=(8,6))
  sns.boxplot(x='depression', y='satisfaction_etudes', data=Dep_etud,palette={'Oui': 'b', 'Non': 'orange'})
  plt.title('Boxplot de satisfaction_etud/depression par etudiants')
  output_file_path = "C:/Users/HP/Downloads/boxplot_satisfaction_etudes_depression.png"
  plt.savefig(output_file_path)
  plt.show()
  print("Chemin d'accès du fichier enregistré :", output_file_path)
```

```
- print("Moyenne par depression:")
  print(Dep_etud.groupby('depression')['satisfaction_etudes'].mean())

- Oui_Dep = Dep_etud[Dep_etud['depression'] == 'Oui']['satisfac-
tion_etudes'].values
  print("\nTest de Shapiro pour les Oui:", shapiro(Oui_Dep))

- Non_Dep = Dep_etud[Dep_etud['depression'] == 'Non']['satisfac-
tion_etudes'].values
  print("Test de Shapiro pour les Non:", shapiro(Non_Dep))

- import pandas as pd
  from scipy.stats import bartlett
  # Séparer les données selon le sexe
  groupe_Oui = Dep_etud[Dep_etud["depression"] == "Oui"]["satisfac-
tion_etudes"]
  groupe_Non = Dep_etud[Dep_etud["depression"] == "Non"]["satisfac-
tion_etudes"]
  # Effectuer le test de Bartlett pour comparer les variances
  T, p_value = bartlett(groupe_Oui, groupe_Non)
  print(f"Statistic T: {T}")
  print(f"P-value: {p_value}")

- # Fonction pour calculer la statistique de test (différence de moyennes)
  def diff_moyennes(g1, g2):
      return np.mean(g1) - np.mean(g2)

  # Calcul de la statistique observée
  stat_obs = diff_moyennes(groupe_Oui, groupe_Non)

  # Nombre de permutations
  n_permutations = 10000

  # Permutations
  diff_permutations = []
  for _ in range(n_permutations):
      data_permutee = np.random.permutation(Dep_etud['satisfac-
tion_etudes'])
      perm_groupe_Oui = data_permutee[:len(groupe_Oui)]
      perm_groupe_Non = data_permutee[len(groupe_Oui):]
      diff_permutations.append(diff_moyennes(perm_groupe_Oui,
perm_groupe_Non))

  # Calcul de la p-value
  p_value_permutation = np.sum(np.abs(diff_permutations) >=
np.abs(stat_obs)) / n_permutations

  # Affichage des résultats
  print(f"Statistique observée: {stat_obs}")
  print(f"Valeur p: {p_value_permutation}")

  if p_value_permutation < 0.05:
```

```
    print("La différence de satisfaction des études entre les deux
groupes est statistiquement significative (p < 0.05).")
 else:
    print("La différence de satisfaction des études entre les deux
groupes n'est pas statistiquement significative (p > 0.05).")
```

```
- import pandas as pd
  import numpy as np

  # Fonction pour calculer la statistique de test (différence de
moyennes)
  def diff_moyennes(g1, g2):
      return np.mean(g1) - np.mean(g2)

  # Calcul de la statistique observée
  stat_obs = diff_moyennes(groupe_Oui, groupe_Non)

  # Nombre de permutations
  n_permutations = 10000

  # Permutations
  diff_permutations = []
  for _ in range(n_permutations):
      data_permutee     =     np.random.permutation(Dep_etud['satisfac-
tion_etudes'])
      perm_groupe_Oui = data_permutee[:len(groupe_Oui)]
      perm_groupe_Non = data_permutee[len(groupe_Oui):]
      diff_permutations.append(diff_moyennes(perm_groupe_Oui,
perm_groupe_Non))

  # Calcul de la p-value avec alternative="greater"
  p_value_permutation = np.sum(np.array(diff_permutations) >= stat_obs)
/ n_permutations

  # Affichage des résultats
  print(f"Statistique observée: {stat_obs}")
  print(f"Valeur p: {p_value_permutation}")

  if p_value_permutation < 0.05:
      print("La moyenne de satisfaction des études du groupe 'Oui' est
significativement plus grande que celle du groupe 'Non' (p < 0.05).")
  else:
      print("La moyenne de satisfaction des études du groupe 'Oui' n'est
pas significativement plus grande que celle du groupe 'Non' (p >
0.05).")
```

```
- import pandas as pd
  import matplotlib.pyplot as plt
  import seaborn as sns
  plt.figure(figsize=(10,6))
  sns.boxplot(x='diplome_suivi',       y='       satisfaction_travail',
data=Dep_etud)
  output_file_path = "C:/Users/HP/Downloads/boxplot_satisfaction_trav-
ail.png"
  plt.savefig(output_file_path)
  plt.show()
```

```
print("Chemin d'accès du fichier enregistré :", output_file_path)
```

```
- from scipy.stats import shapiro, bartlett, kruskal
  import statsmodels.api as sm
  from statsmodels.formula.api import ols
  means = Dep_etud.groupby('diplome_suivi')[' satisfaction_travail'].mean()
  print (means)
```

```
- import pandas as pd
  from scipy.stats import shapiro
  # Test de normalité de Shapiro-Wilk pour chaque groupe
  for diplome in Dep_etud['diplome_suivi'].unique():
      stat, p = shapiro(Dep_etud[' satisfaction_travail'][Dep_etud['diplome_suivi'] == diplome])
      print(f"\nTest de Shapiro-Wilk pour le groupe {diplome} :")
      print(f"Statistique={stat:.5f}, P-value={p:.5f}")
      if p < 0.05:
          print("Conclusion : Les données ne suivent pas une distribution normale.")
      else:
          print("Conclusion : Les données suivent une distribution normale.")
```

```
- groups = [Dep_etud[' satisfaction_travail'][Dep_etud['diplome_suivi'] == diplome] for diplome in
  Dep_etud['diplome_suivi'].unique()]
  stat, p = bartlett(*groups)
  print(f"\nBartlett test: Statistic={stat:.5f}, P-value={p:.5f}")
```

```
- stat, p = kruskal(*groups)
  print(f"Kruskal-Wallis test: Statistic={stat:.5f}, P-value={p:.5f}")
```

```
- import pandas as pd
  # Filtrer les données pour ne garder que les modalités 'sain' et 'modérés'
  filtered_Dep_etud = Dep_etud[Dep_etud['habitudes_alimentaires'].isin(['sain','Modéré'])]
  # Sélectionner les colonnes 'habitudes_alimentaires' et 'depression'
  result_Dep_etud = filtered_Dep_etud[['habitudes_alimentaires', 'depression']]
  print(result_Dep_etud)
```

```
- from scipy.stats import chi2_contingency
  tab = pd.crosstab(result_Dep_etud['habitudes_alimentaires'], result_Dep_etud['depression'])
  print(tab)
```

```
- import numpy as np
  import pandas as pd
  from scipy.stats import chi2_contingency

  # Calcul du chi-deux
```

```
    chi2, p, dof, expected = chi2_contingency(tab)
    print("\nValeurs attendues sous forme de tableau :")
    expected_df = pd.DataFrame(expected, columns=tab.columns, in-
    dex=tab.index)
    print(expected_df)

    # Vérification de la règle de Cochran
    expected_count_check = (expected >= 5).sum() / expected.size * 100
    print(f"\nPourcentage des effectifs théoriques ≥ 5: {ex-
    pected_count_check:.2f}%")

    if expected_count_check >= 80:
        print("La règle de Cochran est respectée.")
    else:
        print("La règle de Cochran n'est pas respectée.")


- print(f"Chi2 value: {chi2}")
    print(f"P-value: {p}")


- import pandas as pd
    # Sélectionner les colonnes 'duree_de_sommeil' et 'depression'
    donnee_sommeil = Dep_etud[['duree_sommeil', 'depression']]
    print(donnee_sommeil)


- from scipy.stats import chi2_contingency
    tab_1 = pd.crosstab(Dep_etud['duree_sommeil'],Dep_etud['depression'])
    print(tab_1)


- # Calcul du chi-deux
    chi2, p, dof, expected = chi2_contingency(tab_1)
    print("\nValeurs attendues sous forme de tableau :")
    expected_df = pd.DataFrame(expected, columns=tab_1.columns, in-
    dex=tab_1.index)
    print(expected_df)

    # Vérification de la règle de Cochran
    expected_count_check = (expected >= 5).sum() / expected.size * 100
    print(f"\nPourcentage des effectifs théoriques ≥ 5: {ex-
    pected_count_check:.2f}%")

    if expected_count_check >= 80:
        print("La règle de Cochran est respectée.")
    else:
        print("La règle de Cochran n'est pas respectée.")


- print(f"Chi2 value: {chi2}")
    print(f"P-value: {p}")
```