Processo Seletivo 2024

Raul Myron Silva Amorim – raul.myron@gmail.com

Antes de tudo, por favor, abra o Jupyter Lab e execute lá, onde está tudo de maneira mais organizada e concisa conforme as respostas e as necessidades, organizado em laboratório conforme cada questão com o formato seguido de questões, código, gráficos e resposta. Grato desde já pela atenção.

Respostas:

1. No dataset existem alguns valores faltantes. Antes de começar a manipular os dados, trate essas informações e descreva sucintamente as alterações feitas.

R:

```
print("pré manipulação linhas, colunhas", data.shape)

# aqui estão sendo calculados todos os valores no qual são nulos e sendo inseridos em uma tabela
table_null = data.isnull().sum()
print(table_null[table_null > 0])
#print(table_null.head)
#aqui utilizamos um método do pandas chamado dropna, que remove todos os valores nulos

data_cleaned = data.dropna()  # Remove linhas com nan
data_cleaned = data_cleaned.dropna(axis=1)  # Remove colunas com nan

print(table_null[table_null > 0])
print("em manipulação pós dropna linhas, colunhas", data.shape)
#para realizar knn nós precisamos
#quero que leve em consideração o país e o range de idade, categorizando em códigos

copia_tabela = data.copy()

copia_tabela['Country'] = data['Country'].astype('category').cat.codes
copia_tabela['Age'] = data['Age'].astype('category').cat.codes

#seleciona colunas de valores númericos, tirando id e considerando país e range de idade
```

```
colunas_knn = data.select_dtypes(include=['float64', 'int64']).columns.drop('ID')
colunas_knn = colunas_knn.union(['Country', 'Age'])

print(colunas_knn)

#imputa valores ausentes nas colunas selecionadas usando KNN com 5 vizinhos.
knn_imputer = KNNImputer(n_neighbors=5)
copia_tabela[colunas_knn] = knn_imputer.fit_transform(copia_tabela[colunas_knn])

#salva a tabela
copia_tabela.to_csv('Drugs2.csv', index=False)
income_nulls = copia_tabela['Income (USD)'].isnull().sum()
unico = copia_tabela['Income (USD)'].unique()

print(f"Nulos em Income: {income_nulls}")
print(f"Valores unicos em Income: {unico}")

table_null = copia_tabela.isnull().sum()
print(table_null[table_null > 0])

table_null = copia_tabela.isnull().sum()
colunas_com_valores_ausentes = table_null[table_null > 0].index.tolist()
print(colunas_com_valores_ausentes)
def transforma_cl(valor):
    dicionario_valores = {'CL0': 1, 'CL1': 2, 'CL2': 3, 'CL3': 4, 'CL4': 5, 'CL5': 6, 'CL6': 7}
    return valor.map(dicionario_valores)

#coloco aqui começando em 1, pois ao considerar como começando em 0 pode ocorrer de
0 ser false para alguns algoritmos

drug_columns = ['Alcohol', 'Amphet', 'Amyl', 'Benzos', 'Caff', 'Cannabis','Choc', 'Coke',
'Crack', 'Ecstasy', 'Heroin', 'Ketamine','Legalh', 'LSD', 'Meth', 'Mushrooms', 'Nicotine',
'Semer', 'VSA']

for col in drug_columns:
    if col in copia_tabela.columns:
        copia_tabela[col] = transforma_cl(copia_tabela[col])

print(copia_tabela.head(3))
```

```
cols_faltando = ['Gender', 'Education', 'Country', 'Ethnicity']

for col in cols_faltando:
    if col in copia_tabela.select_dtypes(include=['object']).columns:
        copia_tabela[col] = copia_tabela[col].astype('category').cat.codes

print(copia_tabela.head(3))
#Agora fazendo um ultimo knn pra tudo
table_null = copia_tabela.isnull().sum()
colunas_com_valores_ausentes = table_null[table_null > 0].index.tolist()
print(colunas_com_valores_ausentes)

table_null = copia_tabela.isnull().sum()
print(table_null[table_null > 0])

#knn realizado para os valores em 'Gender', 'Education', 'Country', 'Ethnicity', 'Impulsive' que acabaram virando -1, não sei de outra solução pra isso

knn_imputer = KNNImputer(n_neighbors=2, missing_values=-1)
copia_tabela[cols_faltando] = knn_imputer.fit_transform(copia_tabela[cols_faltando])

#knn para as drogas

knn_imputer = KNNImputer(n_neighbors=5)
copia_tabela[drug_columns] = knn_imputer.fit_transform(copia_tabela[drug_columns])

copia_tabela['Impulsive'].replace('?', np.nan, inplace=True)
copia_tabela['Impulsive'] = copia_tabela['Impulsive'].astype(float)
knn_imputer = KNNImputer(n_neighbors=5)
copia_tabela['Impulsive'] = knn_imputer.fit_transform(copia_tabela[['Impulsive']]).ravel()

table_null = copia_tabela.isnull().sum()
colunas_com_valores_ausentes = table_null[table_null > 0].index.tolist()
print(colunas_com_valores_ausentes)
plt.figure(figsize=(8, 6))
copia_tabela['Gender'].hist(bins=20)
plt.title('Histogram of Gender')
plt.xlabel('Gender')
plt.ylabel('Frequency')
```

```
plt.ylim(0, 200)
plt.show()

#dá pra ver que tem alguns valores estranhos, nesse contexto aqui era pra ser algo binário, a mesma coisa ocorre em 'Gender', 'Education', 'Country', 'Ethnicity', 'Impulsive', pois utilizei KNN, por causa desses valores inconstantes que podem mudar os dados ou interpretação e como temos um grande grupo amostral irei remove-los
#vendo elementos unicos
for col in ['Gender', 'Education', 'Country', 'Ethnicity']:
    print(copia_tabela[col].unique().tolist())
def remove_rows_with_value(df, value):
  return df[~df.eq(value).any(axis=1)]

print(copia_tabela.head(93))

copia_tabela = remove_rows_with_value(copia_tabela, 0.5015957446808511)

copia_tabela.to_csv('Drugs4.csv', index=False)
for col in ['Gender', 'Education', 'Country', 'Ethnicity']:
    print(sorted(copia_tabela[col].unique().tolist()))
copia_tabela = remove_rows_with_value(copia_tabela, 0.5)
for col in ['Gender', 'Education', 'Country', 'Ethnicity']:
    print(sorted(copia_tabela[col].unique().tolist()))
```

Gráfico (se houver):

Legenda/explicação:

Basicamente realizo a manipulação de dados para ficar de maneira razoável, utilizo knn para fazer a filtragem e tratamento de dados além de que realizo uma parte de debug de erro do meu knn pois tive erro com 0.5 em erro de gênero onde fiz a correção excluindo essa coluna.

2. Qual é a distribuição da idade dos indivíduos na amostra? Existem diferenças significativas nas faixas etárias predominantes de consumo entre os grupos de

usuários de diferentes substâncias?

R:

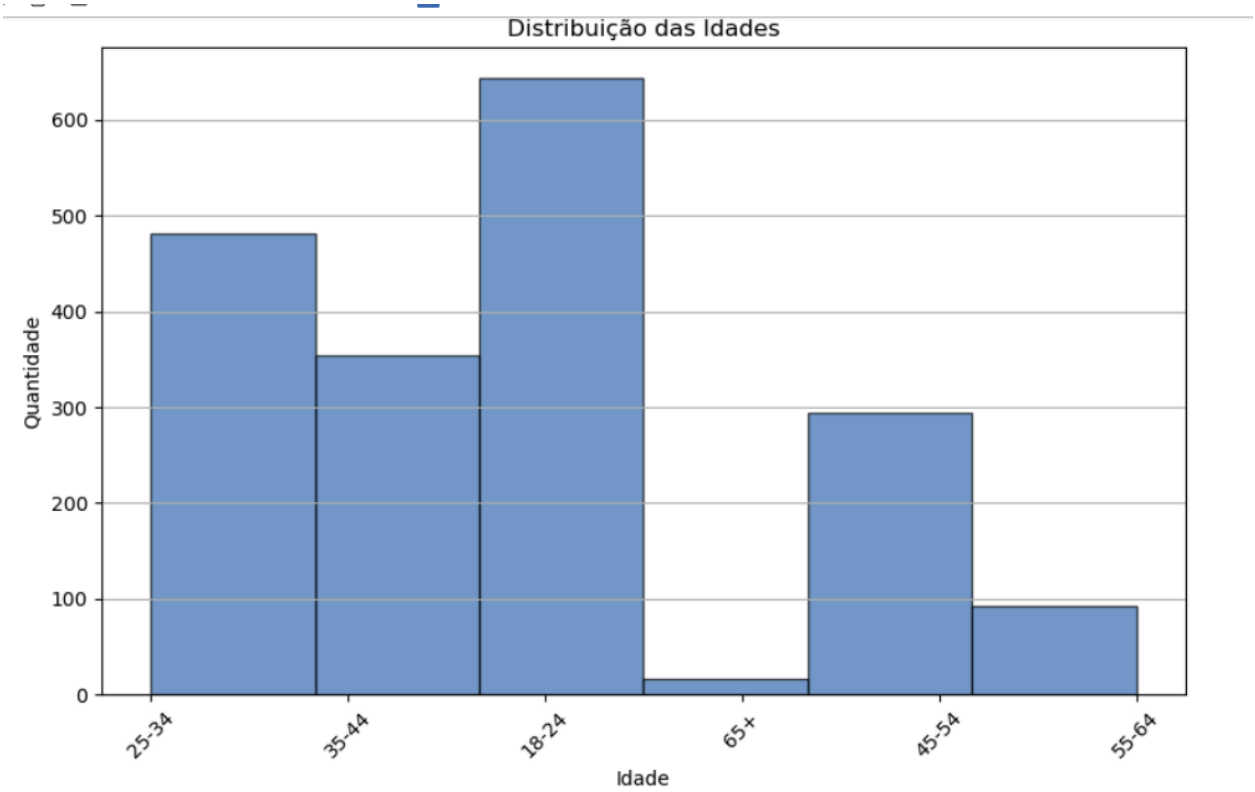


Gráfico (se houver):

Legenda/explicação:

Para responder a parte 2, Existem diferenças significativas nas faixas etárias predominantes de consumo entre os grupos de usuários de diferentes substâncias? Eu irei considerar uma pessoa como usária de substância se ela consumiu no ultimo 1 ano, portanto, CL3, portanto para meu código 'CL0': 1, 'CL1': 2, 'CL2': 3, 'CL3': 4, 'CL4': 5, 'CL5': 6, 'CL6': 7; CL3 é o 4, logo o que é preciso é visualizar o consumo por droga da faixa etária 4o

- CL0 Nunca Usou
- CL1 Usou Mais de Uma Década Atrás
- CL2 Usou nos Últimos Dez Anos
- CL3 Usou no Último Ano (59 vezes)
- CL4 Usou nos Últimos Meses
- CL5 Usou na Última Semana
- CL6 Usou Hoje

2. Existem diferenças significativas nas faixas etárias predominantes de consumo entre os grupos de usuários de diferentes substâncias?

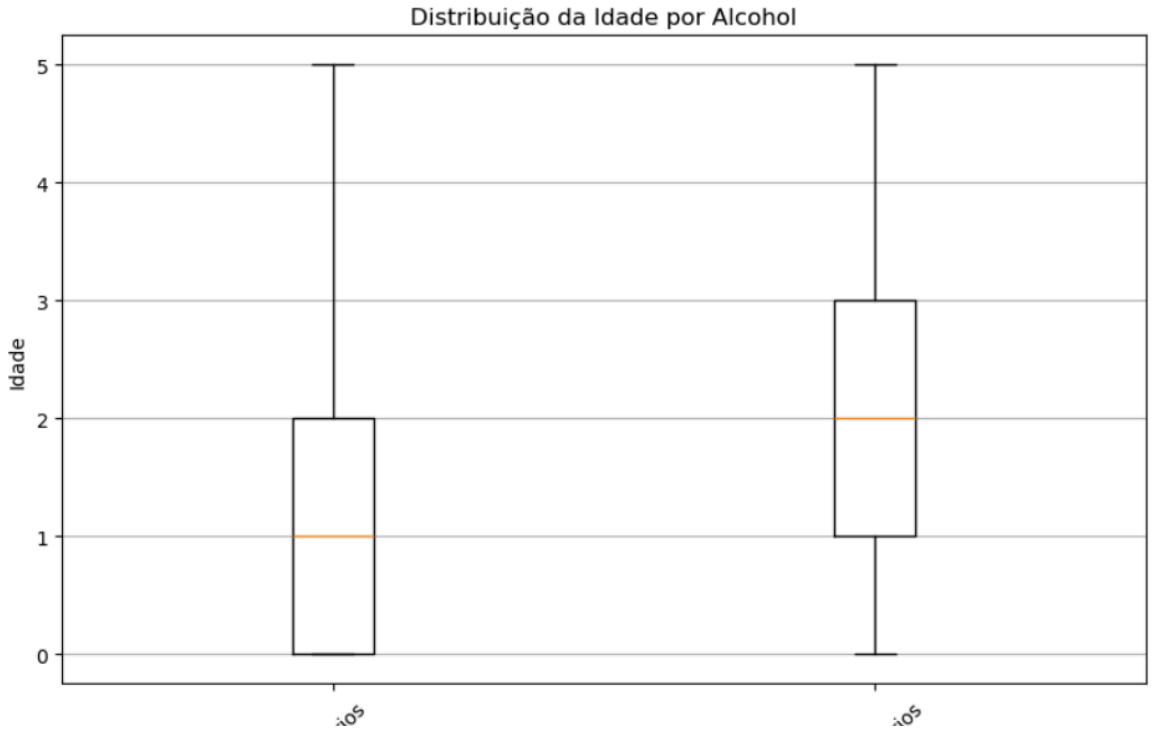
R: drug_columns = ['Alcohol', 'Amphet', 'Amyl', 'Benzos', 'Caff', 'Cannabis','Choc', 'Coke', 'Crack', 'Ecstasy', 'Heroin', 'Ketamine','Legalh', 'LSD', 'Meth', 'Mushrooms', 'Nicotine', 'Semer', 'VSA']

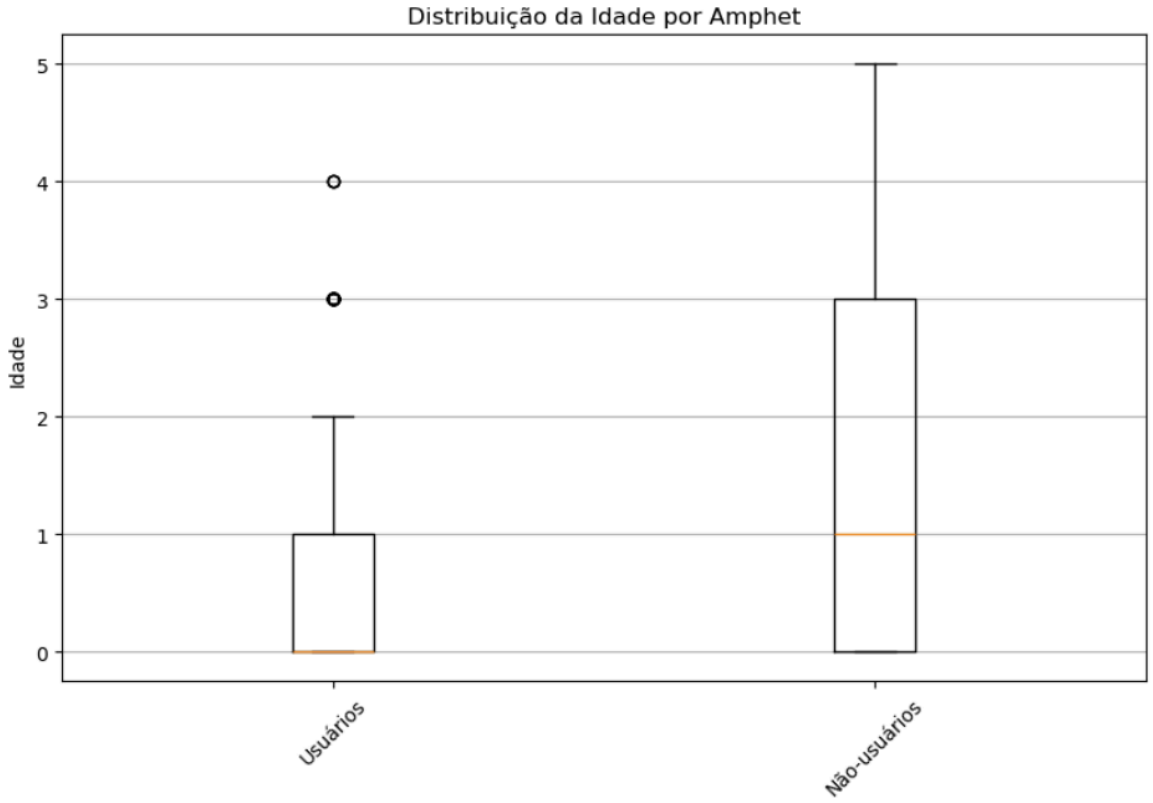
```
#função para categorizar usuários e não-usuários
def categorize_user(value):
    if value >= 4:
        return 'user'
    else:
        return 'non-user'

#função para comparar a distribuição etária entre usuários e não-usuários
def compare_age_distribution(substance, tabela):

    users = tabela[ tabela [f'{substance}_User' ] == 'user']['idades']
    non_users = tabela[ tabela [f'{substance}_User' ] == 'non-user']['idades']

    return users, non_users

for substance in drug_columns:

    categorize_users(copia_tabela, substance)

print("-"*10)
for i in range(len(idades)):
  print(f"Code: {i}, Age: {idades[i]}")
print("-"*10)



for substance in drug_columns:

    users, non_users = compare_age_distribution(substance, copia_tabela)

    print(f"Distribuição das idades para {substance}:")
    print(f"Usuários: {users.value_counts()}")
    print(f"Não-usuários: {non_users.value_counts()}")
```

```
plt.figure(figsize=(10, 6))

plt.boxplot([copia_tabela[copia_tabela[f'{substance}_User'] == 'user']['Age'],
             copia_tabela[copia_tabela[f'{substance}_User'] == 'non-user']['Age']],
            labels=['Usuários', 'Não-usuários'])



plt.title(f'Distribuição da Idade por {substance}')
plt.xlabel('Grupo')
plt.ylabel('Idade')
plt.grid(axis='y')
plt.xticks(rotation=45)
plt.show()
```

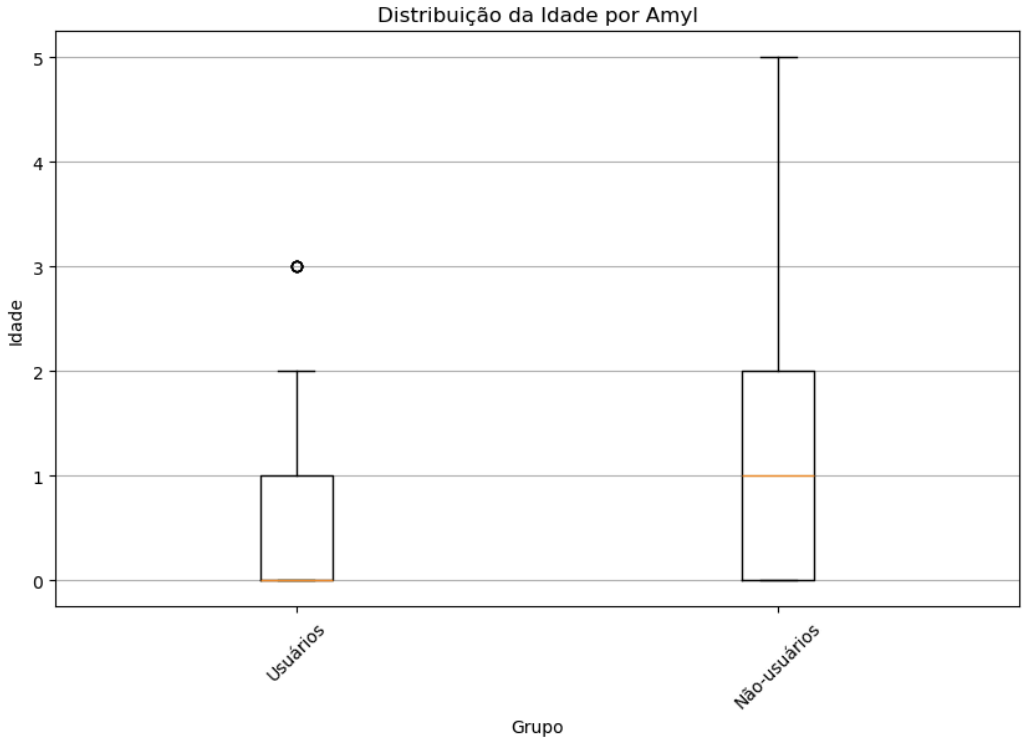
Gráfico (se houver):

```
55-64    13
65+       4
Name: count, dtype: int64
```



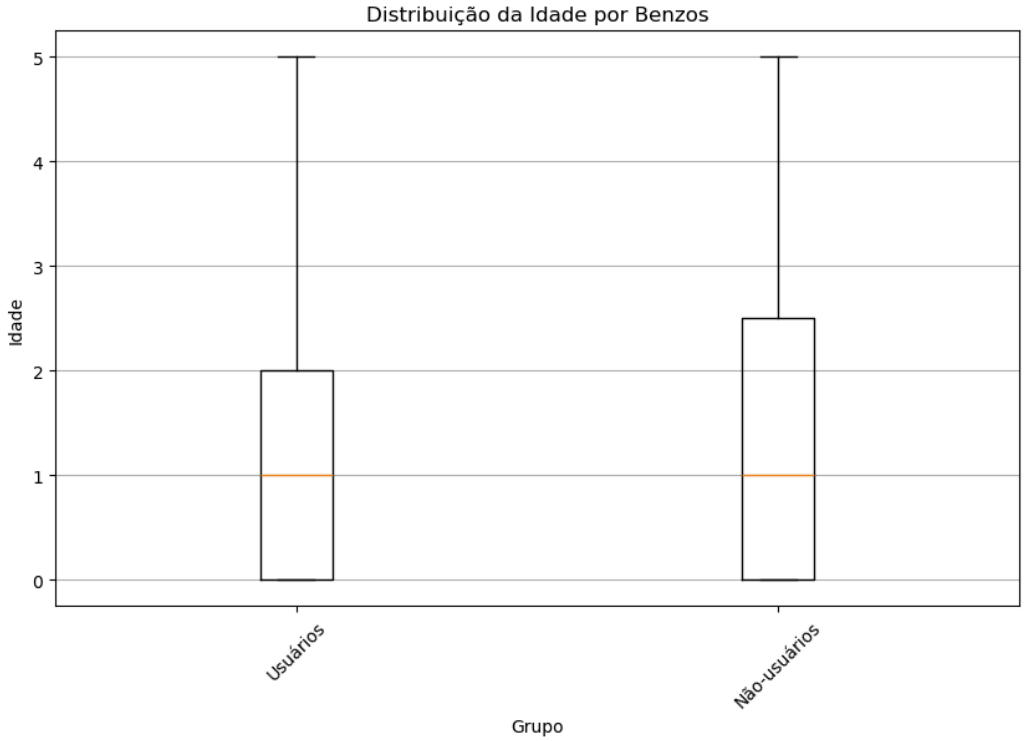



```
Name: count, dtype: int64
```

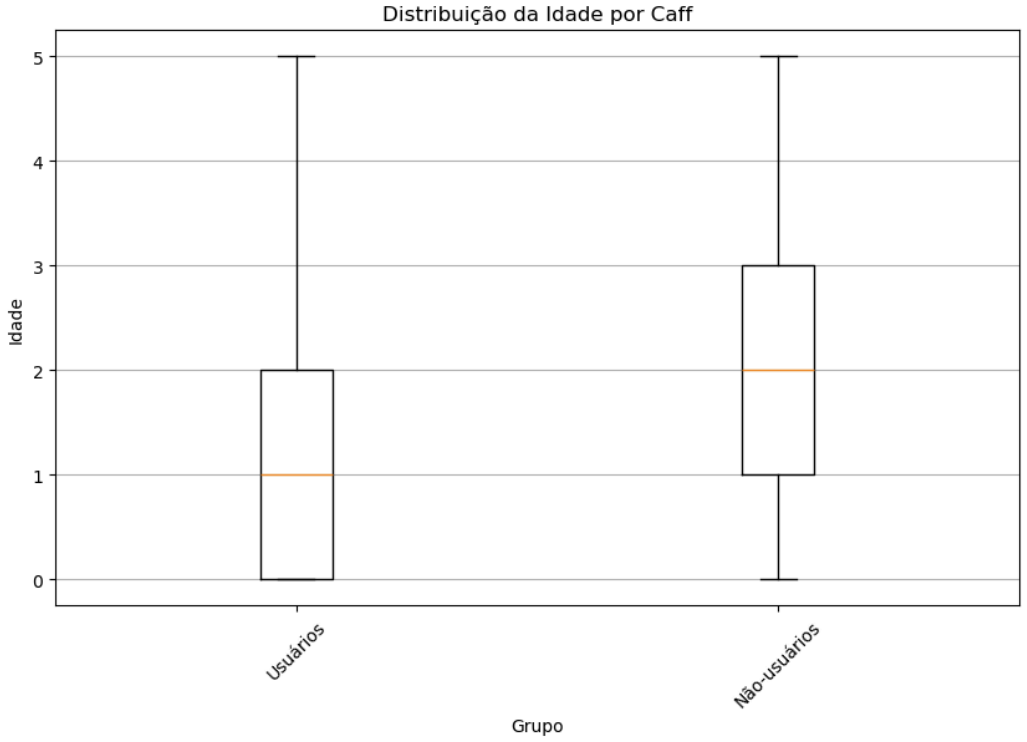



```
Distribuição das idades para Benzos:
```

```
Name: count, dtype: int64
```

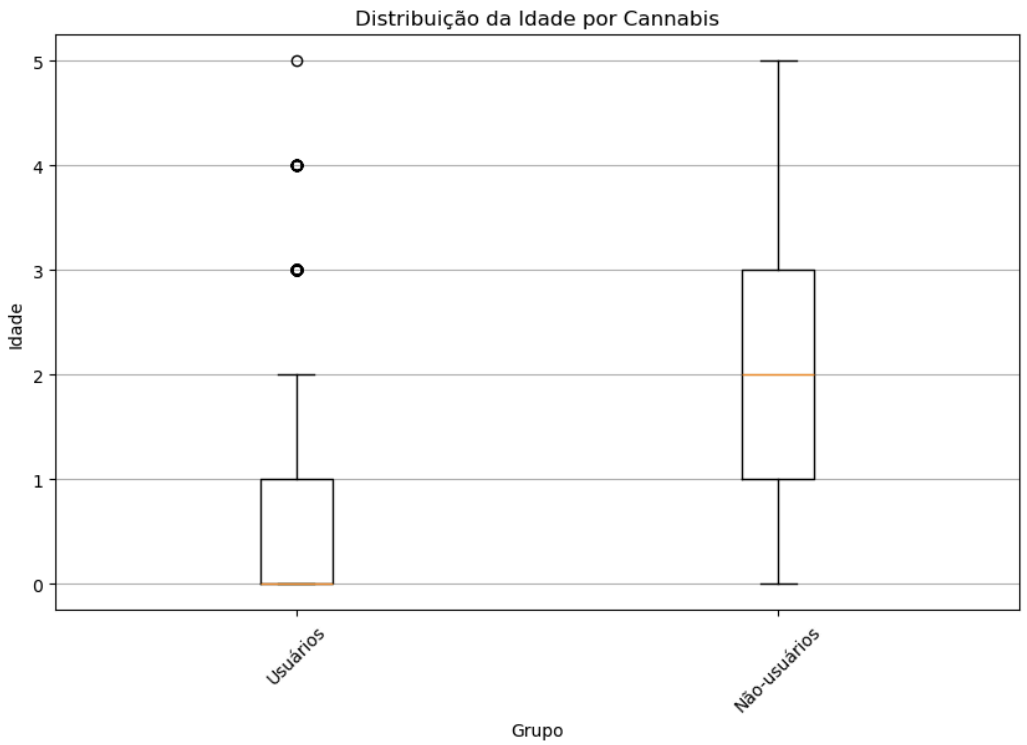



```
Distribuição das idades para Caff:

Name: count, dtype: int64
```



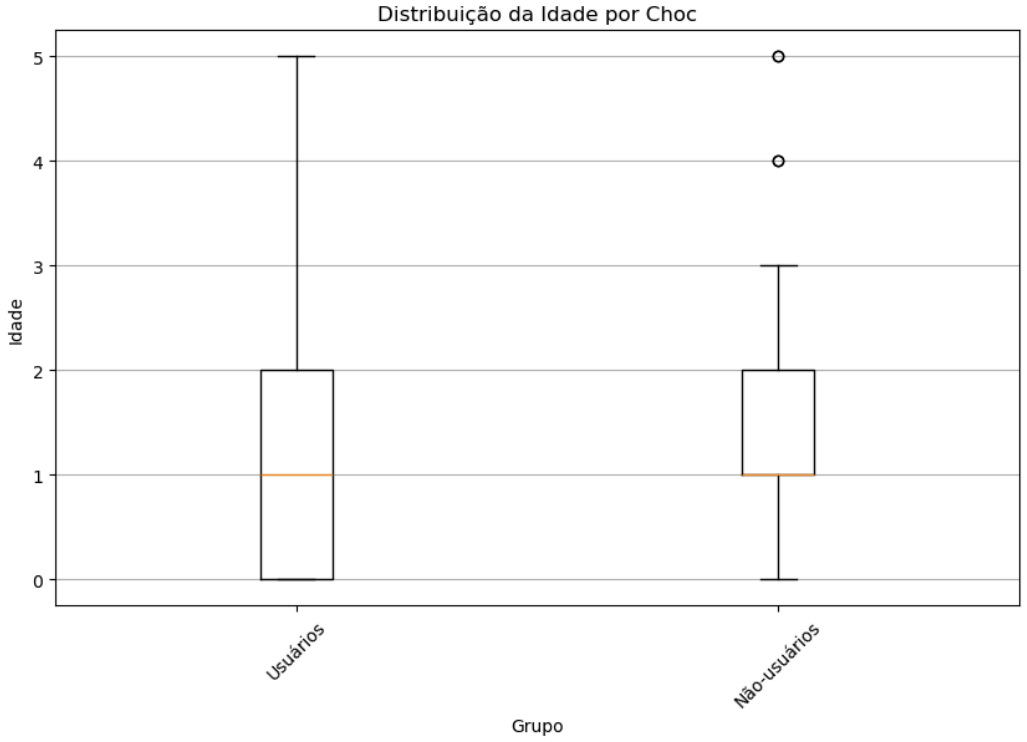

```
Distribuição das idades para Cannabis:
```

```
Name: count, dtype: int64
```

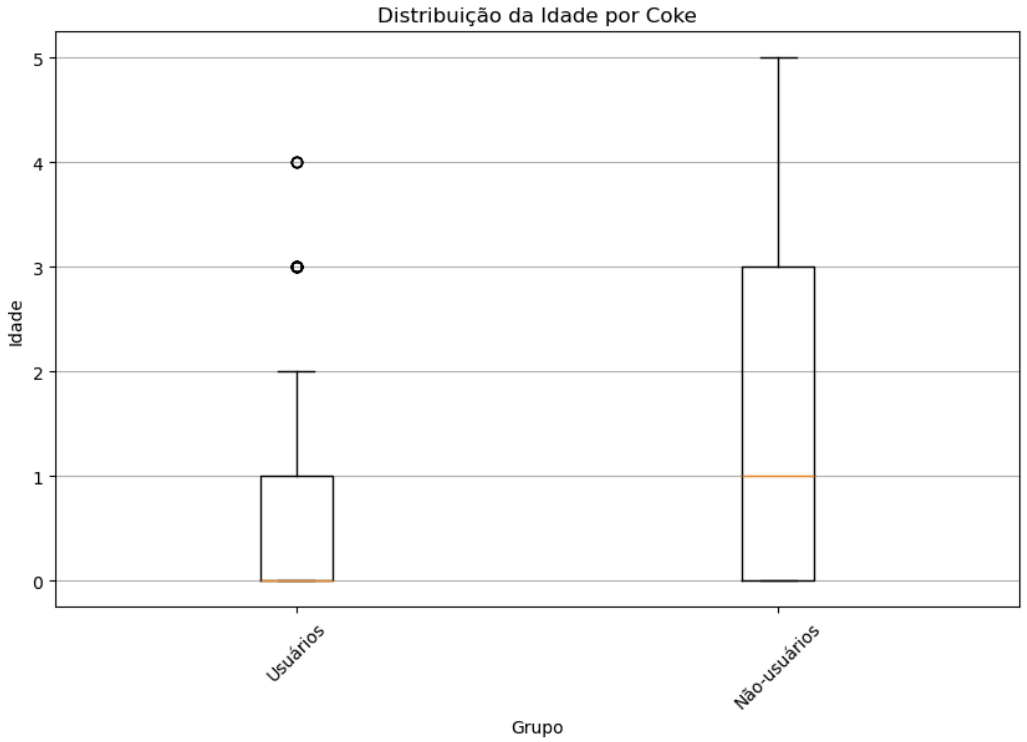



```
Distribuição das idades para Choc:
```

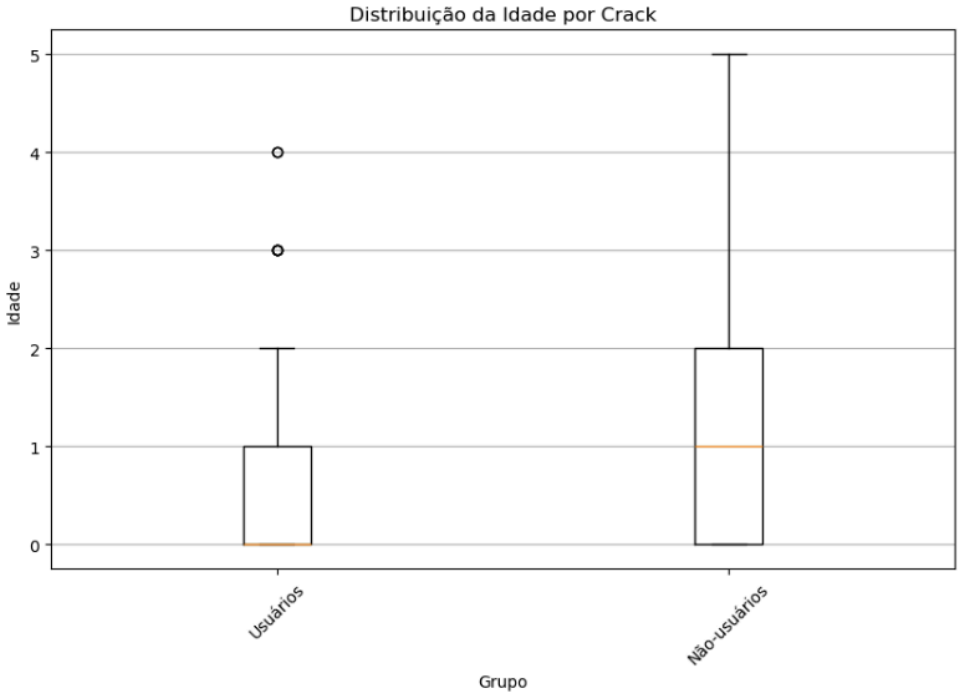
```
Name: count, dtype: int64
```




```
Distribuição das idades para Coke:
```

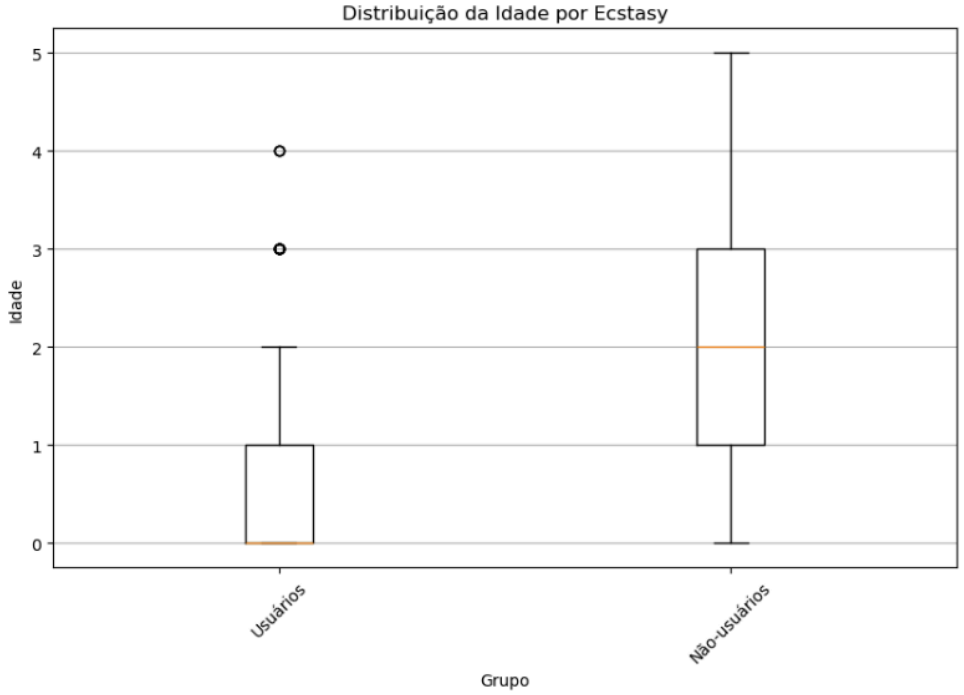



```
Distribuição das idades para Crack:
Usuários: idades
18-24    42
```

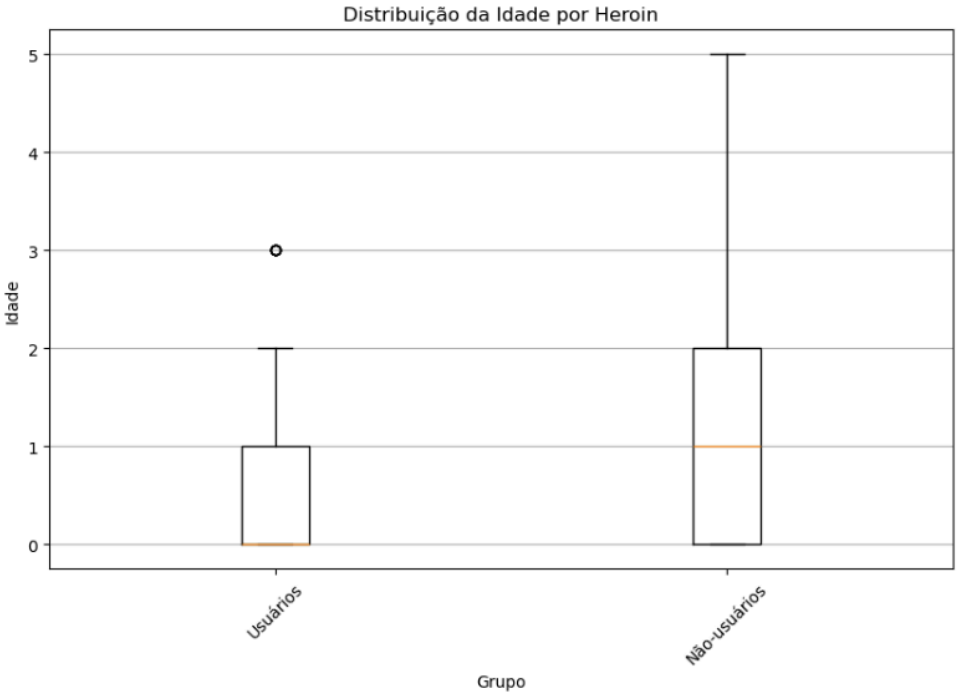


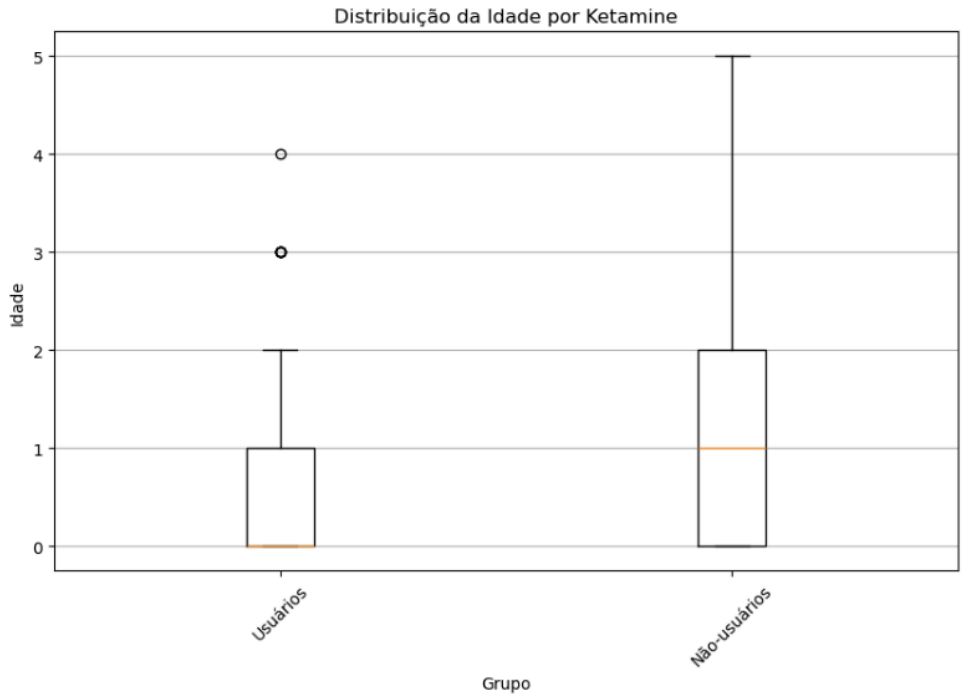

```
Distribuição das idades para Ecstasy:
Usuários: idades
18-24    325
25-34    126
```

```
35-44    310
45-54    275
55-64     90
65+       17
Name: count, dtype: int64
```

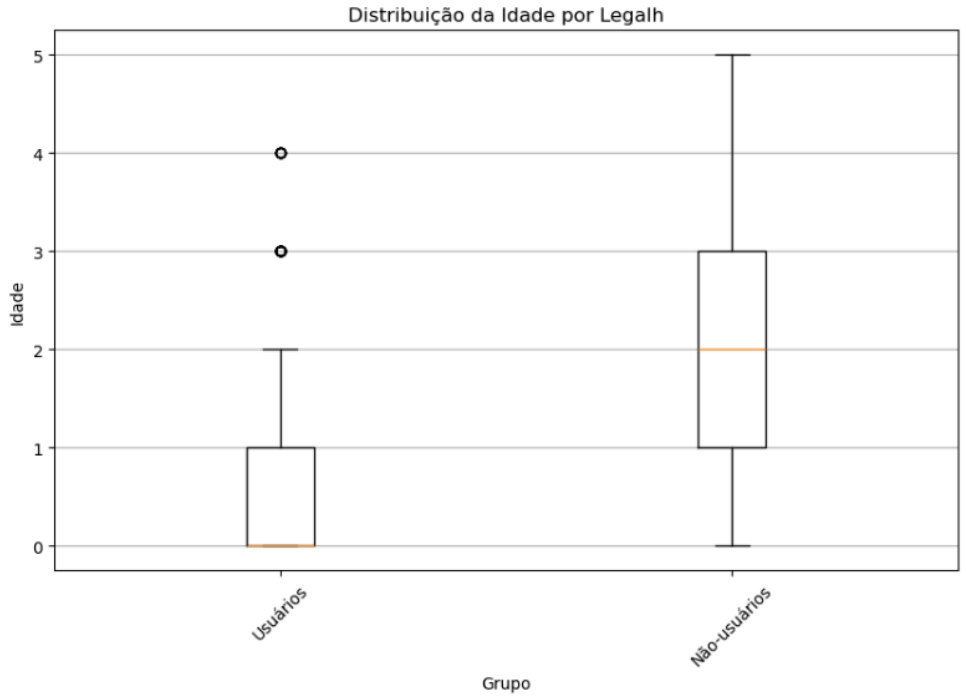


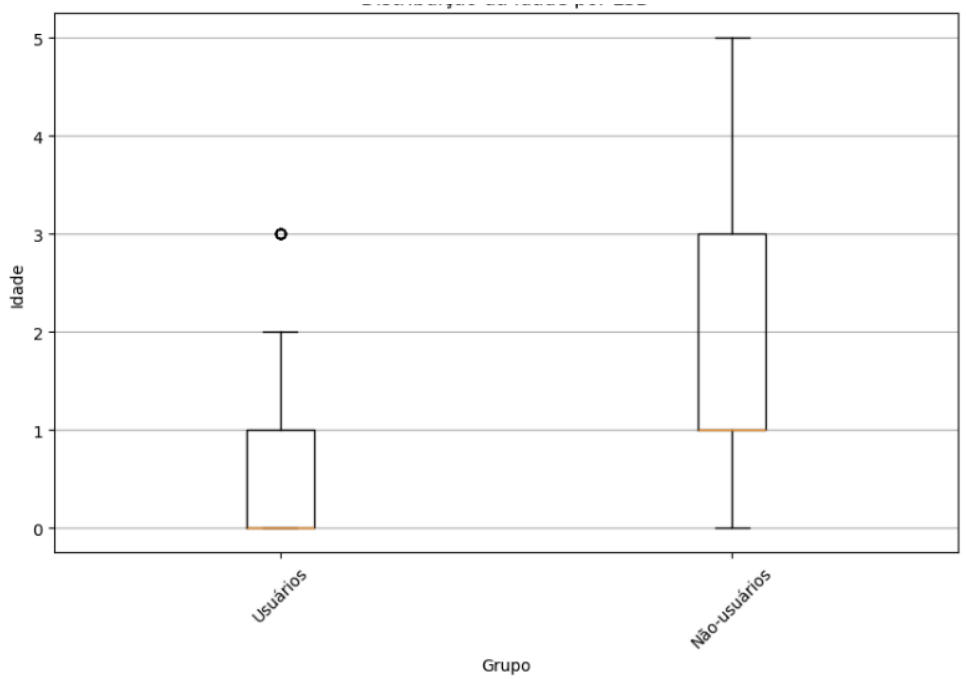

```
45-54    288
55-64     93
65+       17
Name: count, dtype: int64
```



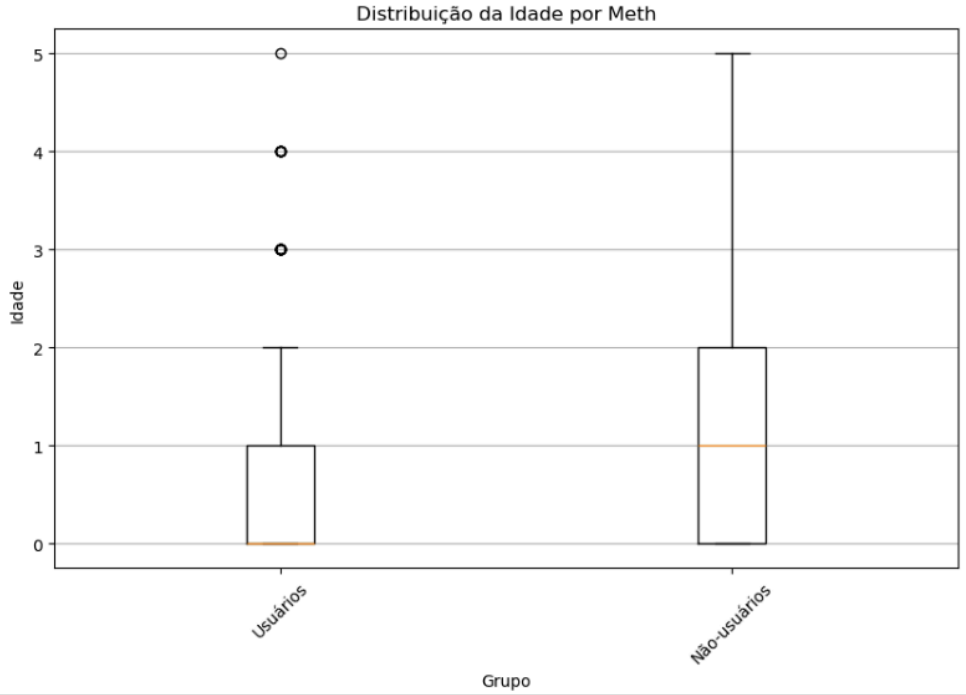


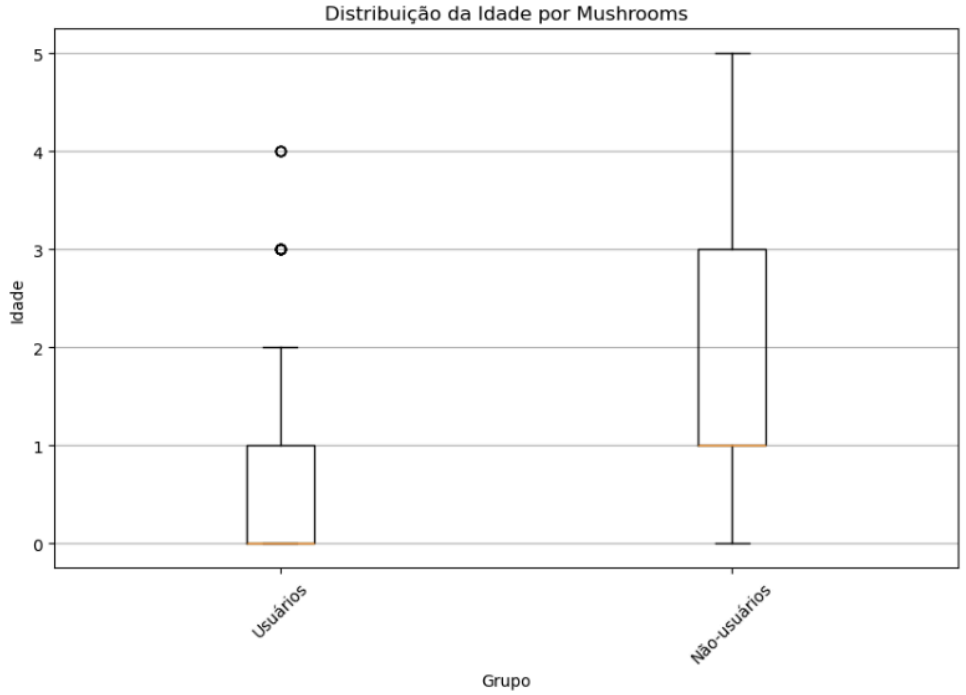

```
Distribuição das idades para Legalh:
Usuários: idades
18-24    361
25-34    120
35-44     51
45-54    269
55-64     86
65+       17
Name: count, dtype: int64
```



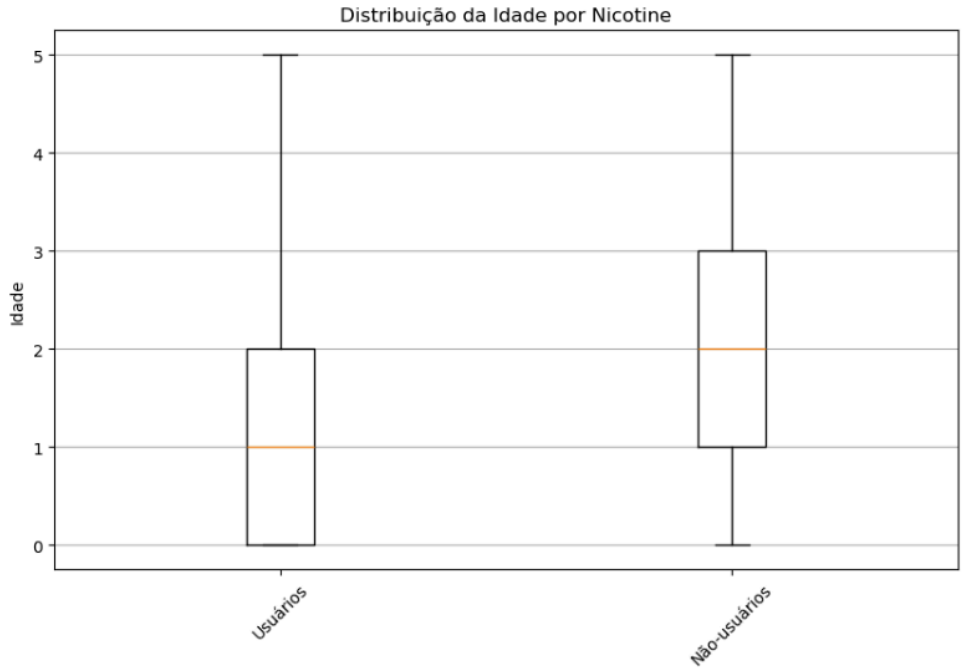



```
Distribuição das idades para Meth:
Usuários: idades
18-24    179
25-34     84
35-44     30
45-54    275
55-64     86
65+       16
Name: count, dtype: int64
```

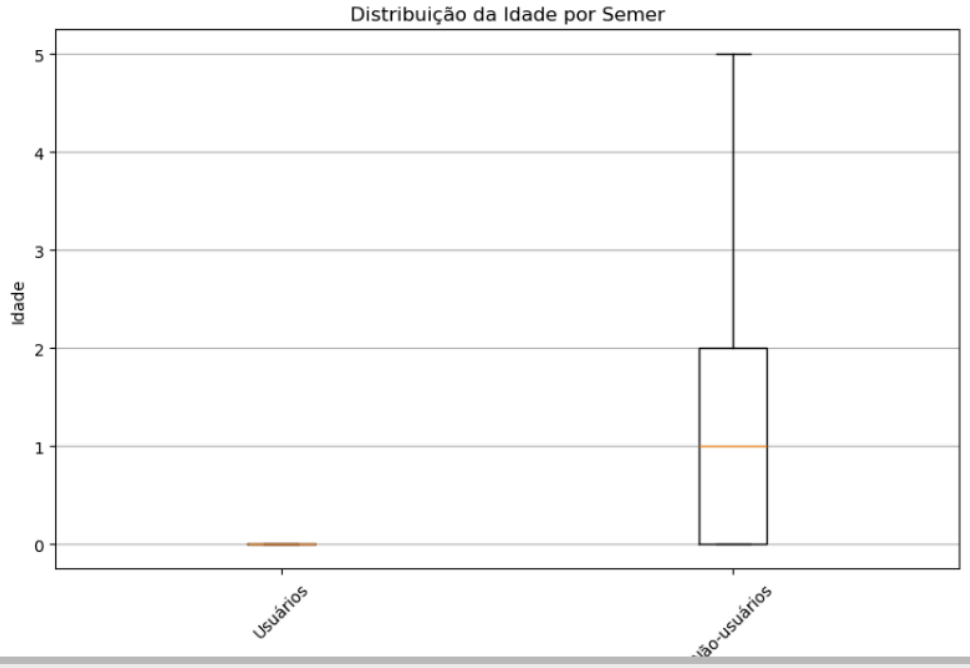




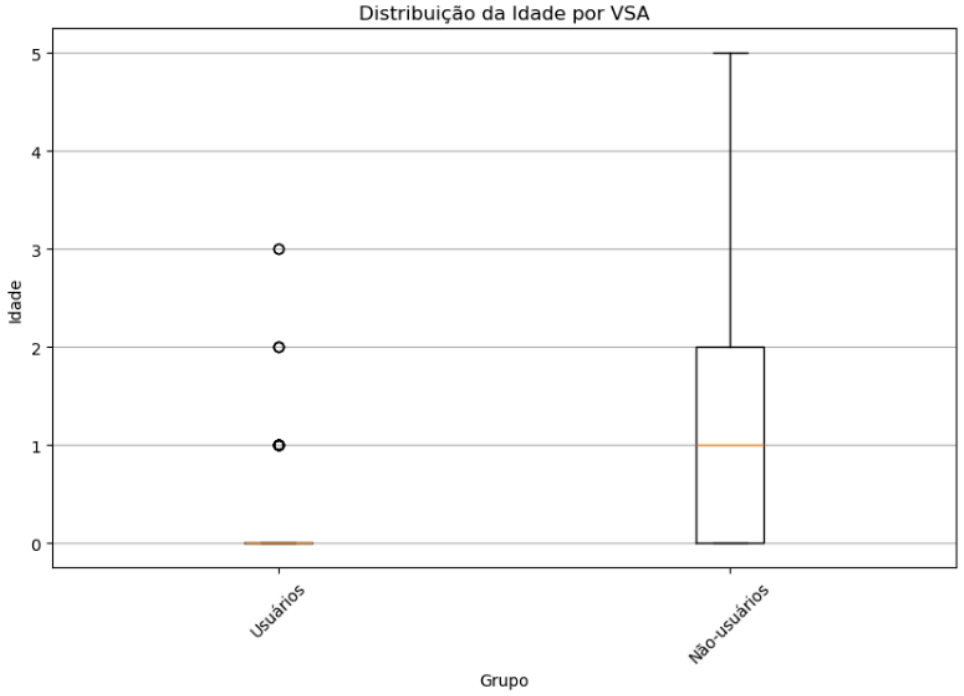

```
Distribuição das idades para Nicotine:
Usuários: idades
18-24    486
25-34    272
45-54    193
35-44    190
18-24    157
55-64     62
65+       12
Name: count, dtype: int64
```



```
25-34    481
35-44    354
45-54    294
55-64     93
65+       17
Name: count, dtype: int64
```




```
Name: count, dtype: int64
```



3. Há uma relação entre o nível educacional e o consumo de substâncias?

Legenda/explicação:

```
18-24    640
25-34    481
35-44    354
45-54    294
55-64     93
65+       17
```

3. Há uma relação entre o nível educacional e o consumo de substâncias?

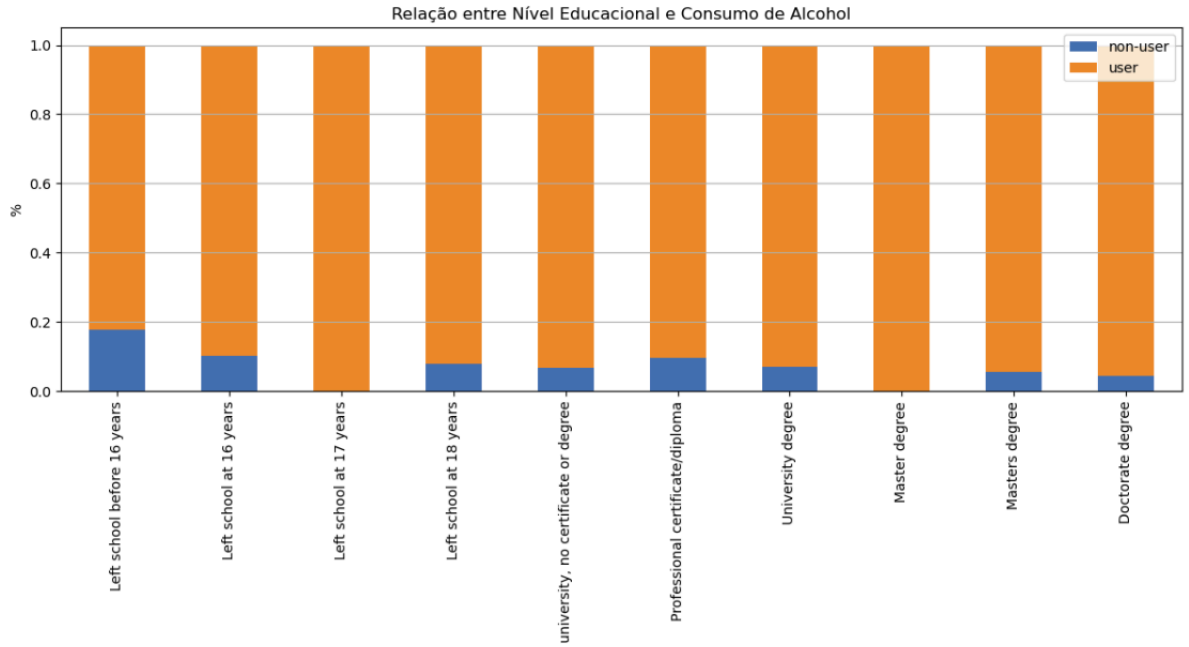
```
from scipy.stats import t

educativo_dic = {
    0: "Doctorate degree",
    1: "Left school at 16 years",
    2: "Left school at 17 years",
    3: "Left school at 18 years",
    4: "Left school before 16 years",
    5: "Master degree",
    6: "Masters degree",
    7: "Professional certificate/diploma",
    8: "Some college or university, no certificate or degree",
    9: "University degree"
}

edu_em_ordem = [
    "Left school before 16 years",
    "Left school at 16 years",
    "Left school at 17 years",
    "Left school at 18 years",
    "Some college or university, no certificate or degree",
    "Professional certificate/diploma",
    "University degree",
    "Master degree",
    "Masters degree",
    "Doctorate degree"
]

copia_tabela['educativo'] = copia_tabela['Education'].map(educativo_dic)
```

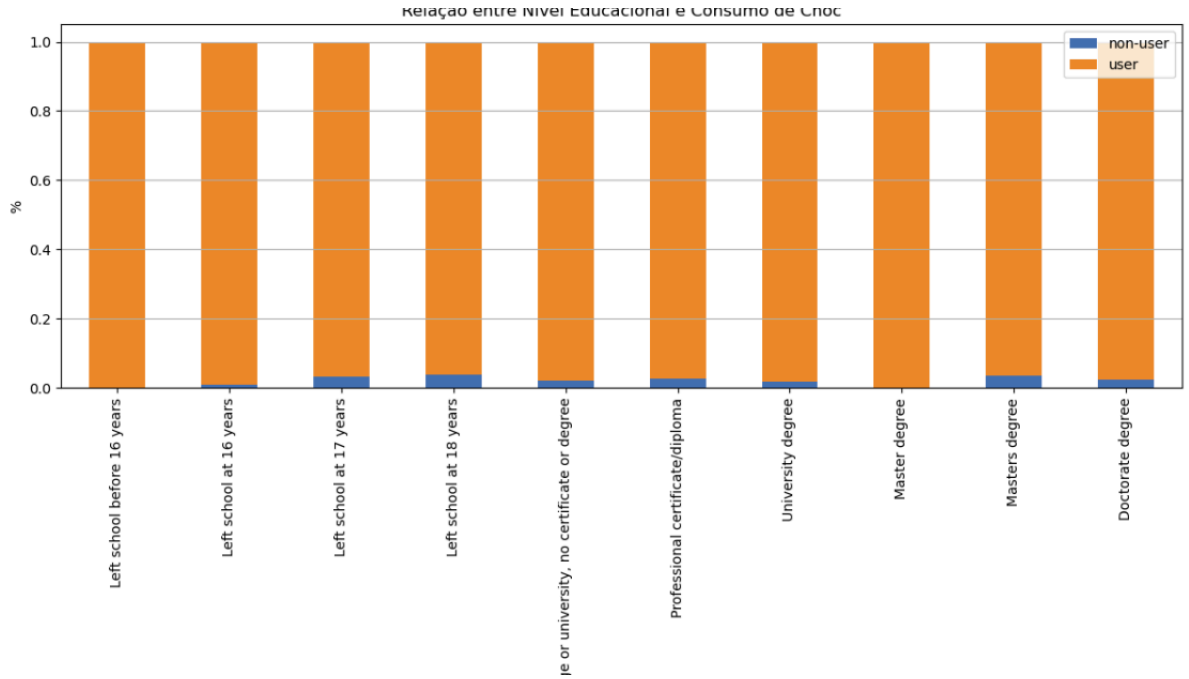
```
#função para categorizar todos os usuários em um DataFrame para uma substância
def categorize_users(df, drug):
    df[f'{drug}_User'] = df[drug].apply(categorize_user)

#aplicar a função para cada substância
for substance in drug_columns:
    categorize_users(copia_tabela, substance)

#função para comparar a distribuição educacional entre usuários e não-usuários

def compare_education_distribution(substance, df):
    users = df[df[f'{substance}_User'] == 'user']['educativo']
    non_users = df[df[f'{substance}_User'] == 'non-user']['educativo']
    return users, non_users

#analisar cada substância e imprimir os resultados

for substance in drug_columns:

                                    cross_tab          =          pd.crosstab(copia_tabela['educativo'],
copia_tabela[f'{substance}_User']).reindex(edu_em_ordem)
    cross_tab_norm = cross_tab.div(cross_tab.sum(1), axis=0)
    cross_tab_norm.plot(kind='bar', stacked=True, figsize=(12, 8))

    plt.title(f'Relação entre Nível Educacional e Consumo de {substance}')
    plt.ylabel('%')
    plt.legend(loc='upper right')
    plt.grid(axis='y')
    plt.tight_layout()
    plt.show()
```

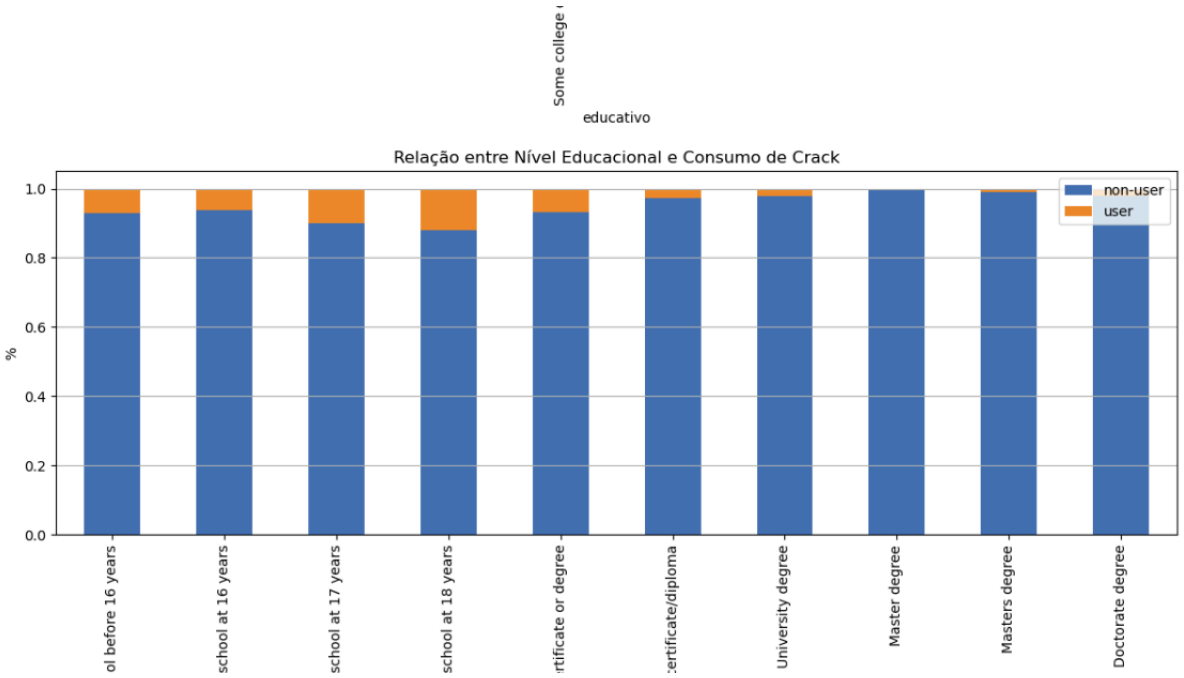
R:

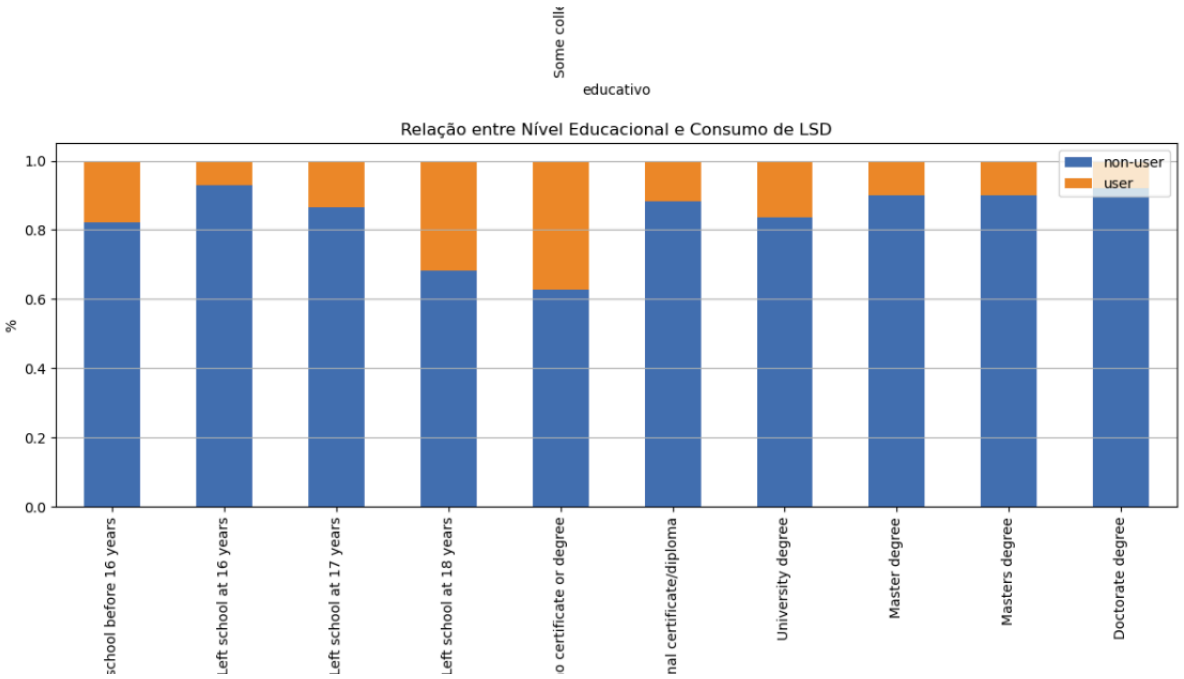
Gráfico (se houver):

```
plt.show()
```


Relação entre Nível Educacional e Consumo de Alcohol
non-user
user
%
1.0
0.8
0.6
0.4
0.2
0.0
Left school before 16 years
Left school at 16 years
Left school at 17 years
Left school at 18 years
university, no certificate or degree
Professional certificate/diploma
University degree
Master degree
Masters degree
Doctorate degree


Relação entre Nível Educacional e Consumo de Choc
non-user
user
%
1.0
0.8
0.6
0.4
0.2
0.0
Left school before 16 years
Left school at 16 years
Left school at 17 years
Left school at 18 years
je or university, no certificate or degree
Professional certificate/diploma
University degree
Master degree
Masters degree
Doctorate degree





Legenda/explicação:

4. Como o gênero influencia no consumo de drogas alucinógenas (LSD, Ecstasy, Ketamine, Cannabis e Mushrooms)? Explique.

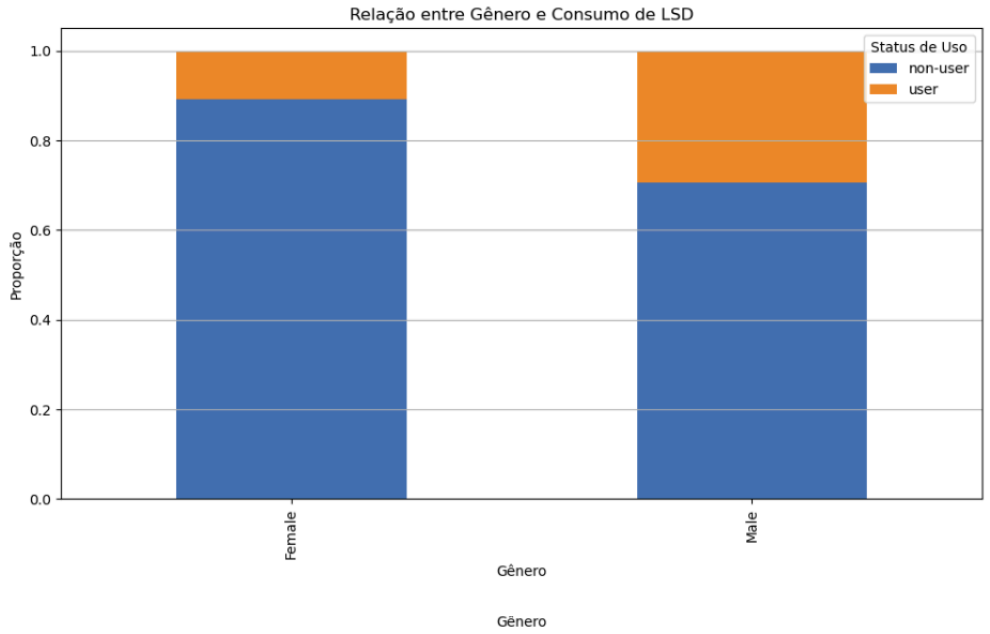
```
hallucinogens = ['LSD', 'Ecstasy', 'Ketamine', 'Cannabis', 'Mushrooms']

#função para categorizar todos os usuários em um DataFrame para uma substância
def categorize_users(tabela, drug):
    tabela[f'{drug}_User'] = tabela[drug].apply(categorize_user)

#aplicar a função para cada substância alucinógena
```

```
for substance in hallucinogens:
    categorize_users(copia_tabela, substance)

#função para analisar a relação entre gênero e consumo de drogas alucinógenas
def analyze_gender_relationship(df, substance):
    cross_tab = pd.crosstab(df['Gender'], df[f'{substance}_User'])
    return cross_tab

#analisar cada substância e imprimir os resultados
for substance in hallucinogens:

    cross_tab = analyze_gender_relationship(copia_tabela, substance)
    cross_tab.index = cross_tab.index.map({0: 'Female', 1: 'Male'})
    print(f"Análise de {substance}:")
    print(cross_tab)

    # Plotar a relação entre gênero e consumo de substâncias

    cross_tab_norm = cross_tab.div(cross_tab.sum(1), axis=0)
    cross_tab_norm.plot(kind='bar', stacked=True, figsize=(10, 6))

    plt.title(f'Relação entre Gênero e Consumo de {substance}')
    plt.xlabel('Gênero')
    plt.ylabel('Proporção')
    plt.legend(title='Status de Uso', loc='upper right')
    plt.grid(axis='y')
    plt.tight_layout()
    plt.show()
```

R:

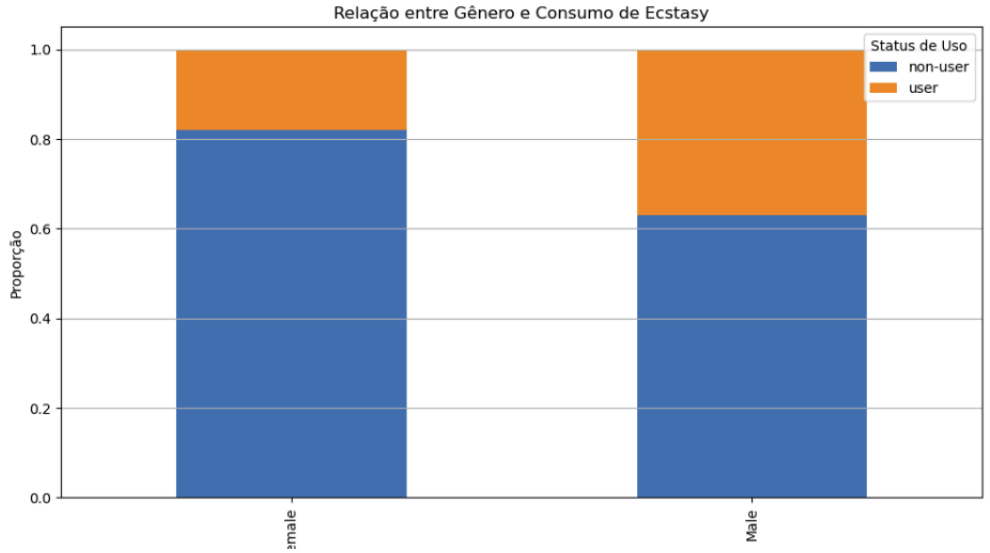
Gráfico (se houver):

```
LSD_User  non-user  user
Gender
Female         836   102
Male           666   278
```



Gênero

```
Análise de Ecstasy:
Ecstasy_User  non-user  user
Gender
Female             770   168
Male               595   349
```



Legenda/explicação:

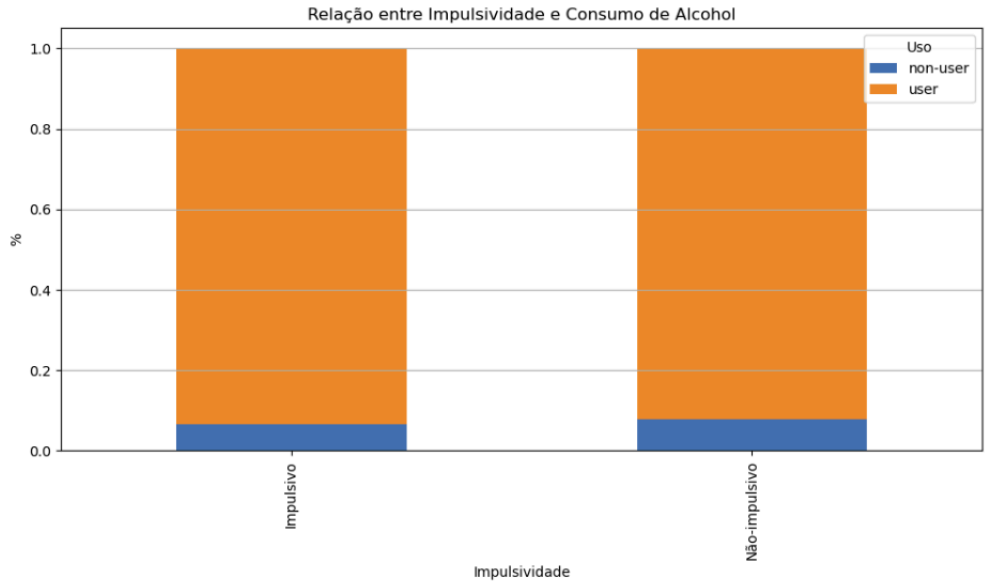
Conclusão: Creio que não dá pra se explicar esses dados com somente esses fatores que foram apresentados aqui, entretanto pelo o que se foi apresentado Homens consomem mais drogas.

5. Qual é a proporção de participantes que se auto-classificam como impulsivos (score superior a zero)? Existe uma correlação entre a impulsividade e o consumo de substâncias?

R:

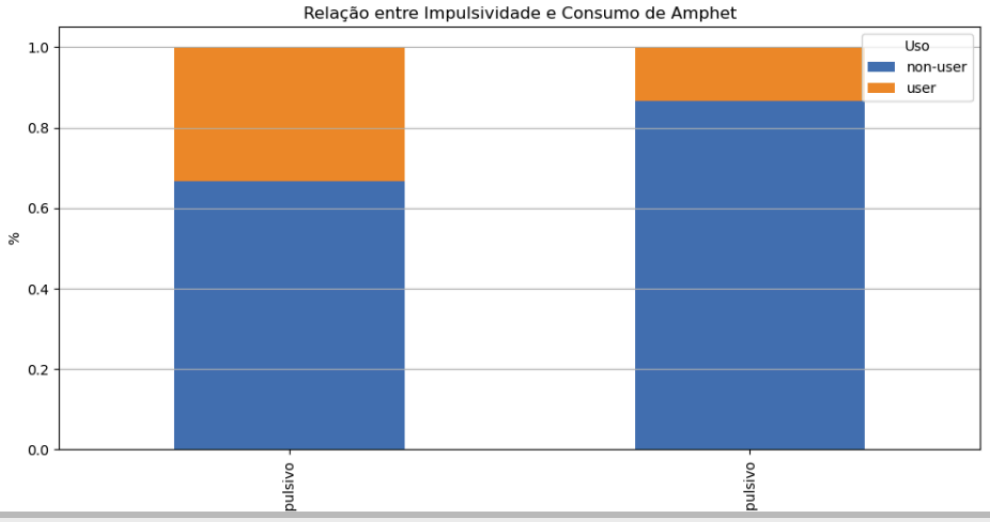
Gráfico (se houver):

Não-impulsivo 74 881

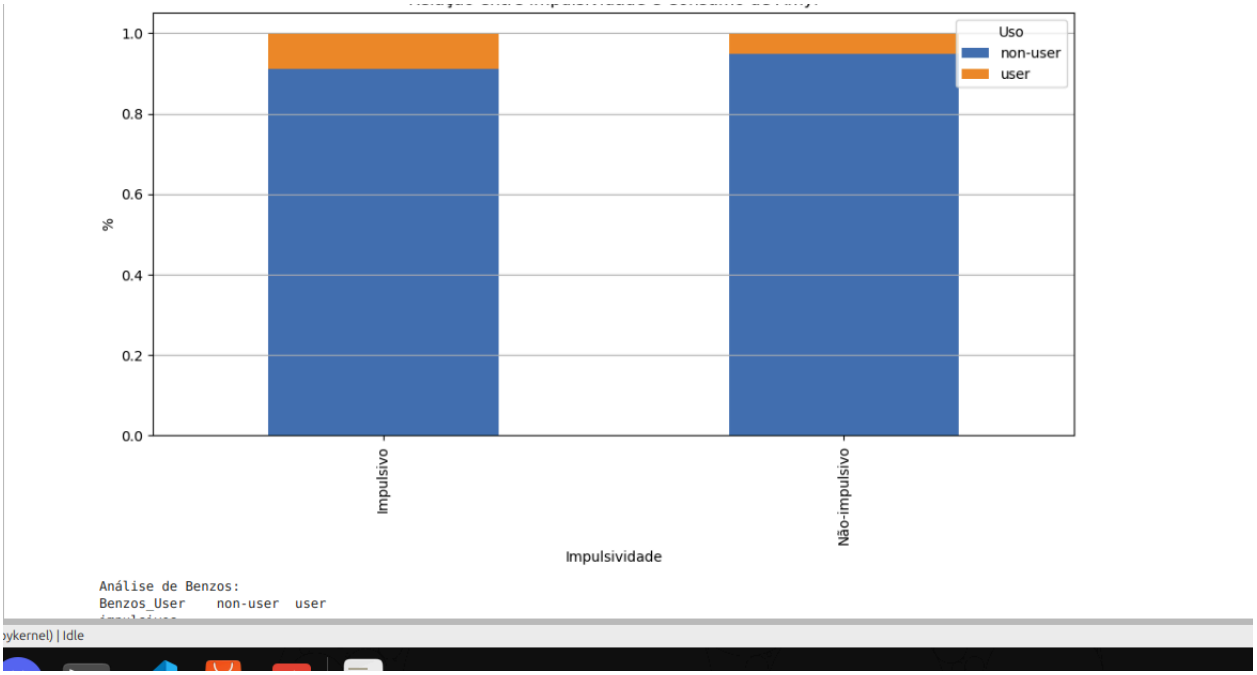


Análise de Amphet:
Amphet_User non-user user
impulsivos
Impulsivo 618 309

Análise de Amphet:
Amphet_User non-user user
impulsivos
Impulsivo 618 309
Não-impulsivo 828 127

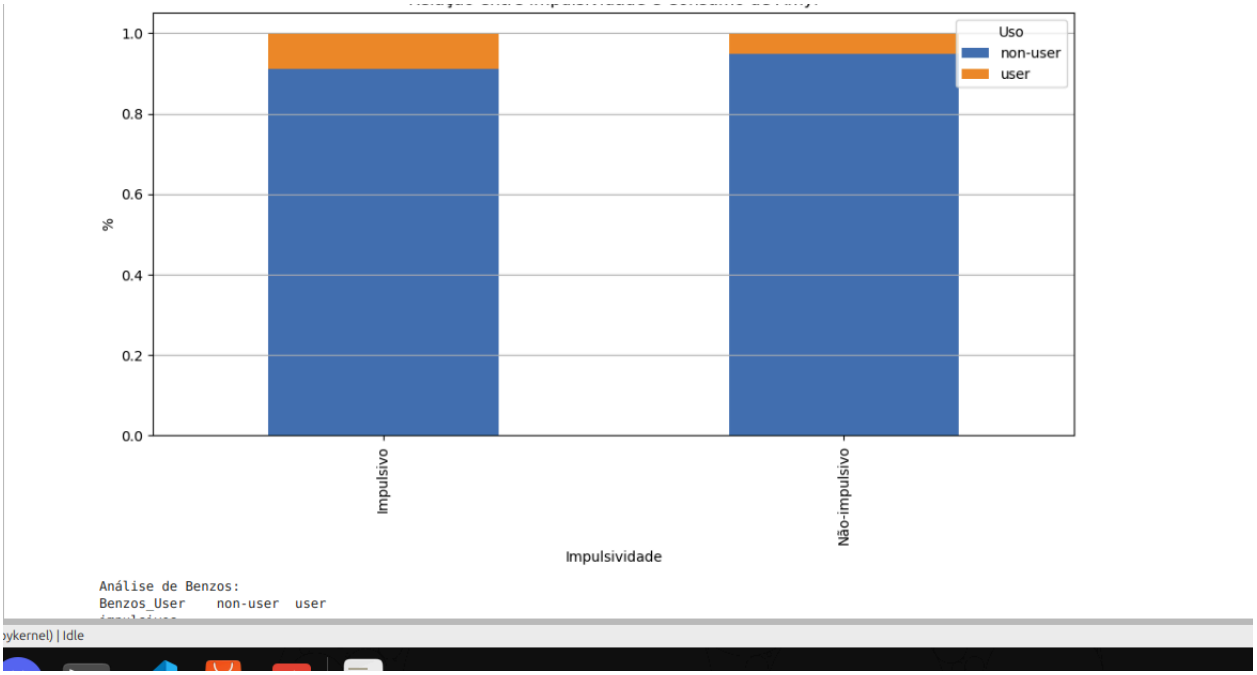


ykernel) | Idle



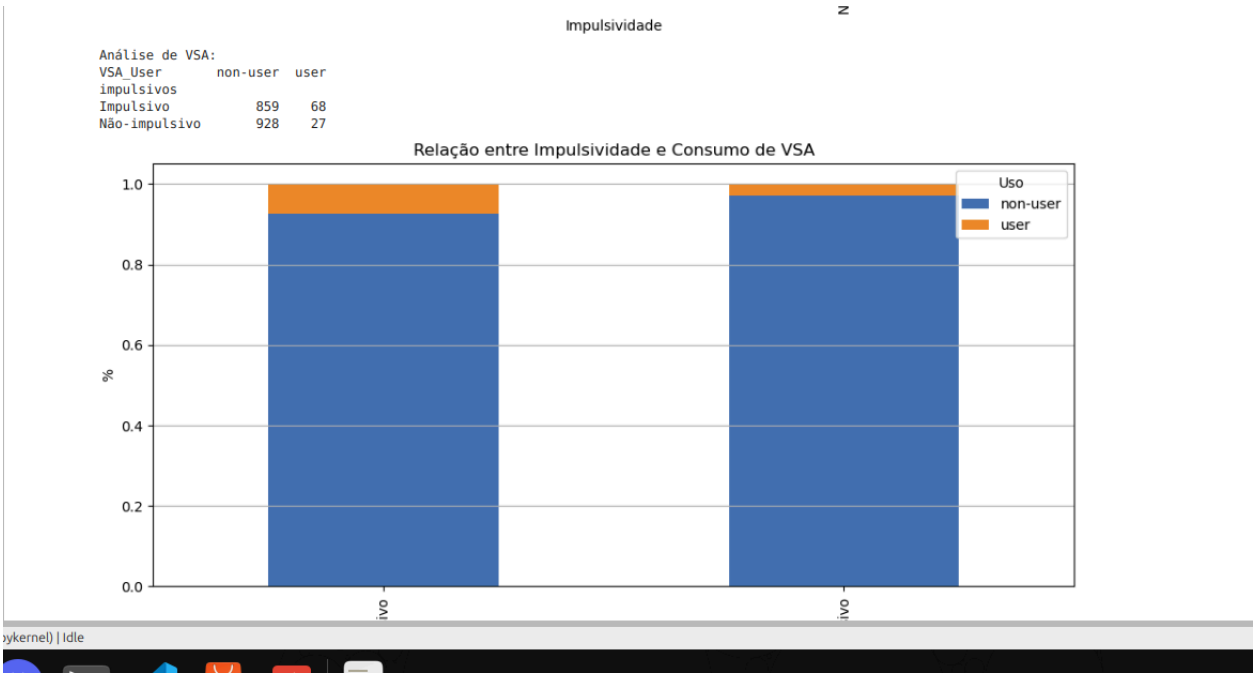

```
Análise de Benzos:
Benzos_User    non-user  user
```

ykernel) | Idle




```
Análise de Benzos:
Benzos_User    non-user  user
```

ykernel) | Idle



Legenda/explicação: Dá pra ver que pessoas impulsivas tendem consumir mais drogas, especialmente quando se trata de LSD, BENZOS, AMPHET, COGUMELOS, NICOTINA, LEGALH, KETAMINE E ECTASY

Impulsividade X Consumo de substâncias:

Alcohol: 0.04

Amphet: 0.29

Amyl: 0.13

Benzos: 0.22

Caff: 0.05

Cannabis: 0.31

Choc: -0.02

Coke: 0.26

Crack: 0.19

Ecstasy: 0.26

Heroin: 0.20

Ketamine: 0.18

Legalh: 0.27

LSD: 0.23

Meth: 0.18

Mushrooms: 0.26

Nicotine: 0.25

Semer: 0.01

VSA: 0.18

6. Classifique as variáveis entre qualitativas (ordinal ou nominal), ou quantitativas (discreta, contínuas).

R:

1 .Variáveis Qualitativas Nominais: São categóricas e não têm uma ordem intrínseca.

- Country
- Gender
- Ethnicity

2. Variáveis Qualitativas Ordinais: São categóricas, mas têm uma ordem natural.

- Age
- Education

3. Variáveis Quantitativas Discretas: São numéricas e contáveis em unidades discretas.

- Impulsive
- Ser usuário da droga (cl1, cl2, cl3, ...)

4. Variáveis Quantitativas Contínuas: São numéricas e podem assumir qualquer valor em um intervalo contínuo.

- Income (USD)

Gráfico (se houver):

Legenda/explicação:

7. Qual é a proporção de consumo de substâncias legais versus ilícitas na amostra (considere a definição de legalidade segundo a legislação brasileira)?

R: legais_lista = ['Alcohol', 'Caff', 'Nicotine', 'Choc', 'Cannabis', 'Mushrooms', 'Ketamine', 'Benzos','Amphet','Legalh']
ilegais_lista = ['Amyl', 'Coke', 'Crack', 'Ecstasy', 'Heroin', 'LSD', 'Meth', 'Semer', 'VSA']
for substance in legal_substances + illegal_substances:
copia_tabela[f'{substance}_User'] = copia_tabela[substance].apply(categorize_user)

#função para calcular a proporção de usuários

def calculate_proportion(tabela, substances):

user_counts = tabela[[f'{substance}_User' for substance in substances]].apply(lambda

```
row: 'user' in row.values, axis=1).sum()

    return user_counts / tabela.shape[0]

# Calcular proporções

legais = calculate_proportion(copia_tabela, legais_lista)
ilegais = calculate_proportion(copia_tabela, ilegais_lista)

print(f"Proporção de consumo de substâncias legais: {legais:.2%}")
print(f"Proporção de consumo de substâncias ilícitas: {ilegais:.2%}")
```

Gráfico (se houver):

Legenda/explicação:

Proporção de consumo de substâncias legais: 99.84%
Proporção de consumo de substâncias ilícitas: 44.31%

8. Quais fatores predizem a probabilidade de um indivíduo consumir crack (Crack)?

R:

Eu acho que essa questão seria bem complementada com uma matriz de confusão mas devido a complexidade tive problemas, para determinar se uma pessoa tem ou não uma prbabidade de consumir outra drogas ou cracks, deve-se considerar:

Gender (foi possível observar que homens tendem a consumir mais drogas), Age (pessoas mais novas tendem a ter consumir mais drogas), Country (pessoas em países com políticas anti-drogas ou com carteis muito fortes tendem a utilizar mais drogas), Education (quanto maior o nível educacional tendem a reduzir o consumo de drogas ilícitas), Psicológicos: Impulsividade (Impulsive) (foi demonstrado como impulsividade é maior a chance do consumo de drogas) e por fim o Consumo de outras substâncias Ilícitas geralmente faz com que pessoas.

Gráfico (se houver):

Legenda/explicação:

9. Qual é a média das pontuações Nscore, Escore, Oscore, AScore, Cscore? Calcule a correlação entre elas.

R:

```
mean_scores = copia_tabela[['Nscore', 'Escore', 'Oscore', 'AScore', 'Cscore']].mean()
print("Média das Pontuações:")
print(mean_scores)

# Calcular a correlação entre as pontuações
correlation_matrix = copia_tabela[['Nscore', 'Escore', 'Oscore', 'AScore', 'Cscore']].corr()
print()

print("Correlação entre as Pontuações:")
print(correlation_matrix)
```

Gráfico (se houver):

Legenda/explicação:

```
print(correlation_matrix)
Média das Pontuações:
Nscore   -0.001305
Escore    0.000434
Oscore    0.002106
AScore   -0.000959
Cscore   -0.001630
dtype: float64

Correlação entre as Pontuações:
          Nscore    Escore    Oscore    AScore    Cscore
Nscore  1.000000 -0.430577  0.011855 -0.217035 -0.390458
Escore -0.430577  1.000000  0.244351  0.158404  0.308911
Oscore  0.011855  0.244351  1.000000  0.040426 -0.058386
AScore -0.217035  0.158404  0.040426  1.000000  0.249885
Cscore -0.390458  0.308911 -0.058386  0.249885  1.000000
```

10. Analise a relação entre o nível de educação (Education) e o consumo de diferentes substâncias ilícitas (como LSD, Amphet, Cannabis, etc.). Identifique se há uma correlação significativa entre essas variáveis e, em caso afirmativo, explore a natureza dessa correlação (positiva/negativa).

R:

```
data4 = 'Drugs4.csv'
data4 = pd.read_csv(data4)

for substance in ilegais_lista:
    data4[f'{substance}_User'] = data4[substance].apply(lambda x: 1 if x >= 4 else 0)

#transforma em dummy
data4 = pd.get_dummies(data4, columns=['Education'], drop_first=True)

# Identificar colunas de educação após a conversão para dummy p/ regressão linear
education_columns = [col for col in data4.columns if 'Education_' in col]

#calcular a correlação entre nível de educação e consumo de substâncias ilícitas
correlation_matrix = data4[education_columns + [f'{substance}_User' for substance in ilegais_lista]].corr()

#filtra as correlações significativas
corr = correlation_matrix.loc[education_columns, [f'{substance}_User' for substance in ilegais_lista]]

#mostra correlações significativas
print("Correlação entre Nível de Educação e Consumo de Substâncias Ilícitas:")

print(corr)
```

Gráfico (se houver):

```
Correlação entre Nível de Educação e Consumo de Substâncias Ilícitas:
              Amphet_User  Amyl_User  Coke_User  Crack_User  Ecstasy_User  \
Education_1.0    -0.043596  -0.017935  -0.038595    0.022522     -0.063790
Education_2.0     0.020649  -0.001969   0.024069    0.036839     -0.002251
Education_3.0     0.070487   0.026369   0.009270    0.091267      0.075364
Education_4.0    -0.005027  -0.016744  -0.012689    0.018062     -0.006762
Education_5.0    -0.022787  -0.020144  -0.021373   -0.015291      0.004165
Education_6.0    -0.119407  -0.019711  -0.075004   -0.063820     -0.105390
Education_7.0    -0.072264  -0.017379  -0.023245   -0.032175     -0.076913
Education_8.0     0.253123   0.053015   0.162148    0.076613      0.248044
Education_9.0    -0.089964   0.000460  -0.056635   -0.061620     -0.086799

              Heroin_User  Legalh_User  LSD_User  Meth_User  Semer_User  \
Education_1.0    -0.011259    -0.064481 -0.076116  -0.010531   -0.009360
Education_2.0     0.019600     0.009392 -0.021708   0.021476   -0.005083
Education_3.0     0.055152     0.091333  0.068231   0.093110   -0.009510
Education_4.0     0.022546    -0.013283 -0.007112   0.014506   -0.004908
Education_5.0    -0.018893    -0.031828 -0.018535  -0.033062   -0.002919
Education_6.0    -0.063201    -0.125172 -0.106177  -0.106521   -0.016485
Education_7.0    -0.042613    -0.067271 -0.087425  -0.046721   -0.016273
Education_8.0     0.100672     0.300288  0.257130   0.176118    0.065987
Education_9.0    -0.040624    -0.111124 -0.057283  -0.082968   -0.023365

              VSA_User
Education_1.0 -0.043086
Education_2.0  0.009427
Education_3.0  0.117434
Education_4.0 -0.008272
Education_5.0 -0.016843
Education_6.0 -0.033192
Education_7.0 -0.066120
Education_8.0  0.106920
Education_9.0 -0.040178
```

Legenda/explicação:

Code: 0, edu: Doctorate degree

Code: 1, edu: Left school at 16 years

Code: 2, edu: Left school at 17 years

Code: 3, edu: Left school at 18 years

Code: 4, edu: Left school before 16 years

Code: 5, edu: Master degree

Code: 6, edu: Masters degree

Code: 7, edu: Professional certificate/ diploma

Code: 8, edu: Some college or university, no certificate or degree

Code: 9, edu: University degree

11. Treine uma árvore de decisão para prever se um indivíduo consome uma determinada substância (por exemplo, álcool, anfetaminas, cannabis) com base em suas características demográficas e pontuações de personalidade. Utilize a acurácia

para avaliar os seus resultados.

R:
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

data4 = 'Drugs4.csv'
data4 = pd.read_csv(data4)

features = ['Gender', 'Age', 'Education', 'Country', 'Ethnicity', 'Nscore', 'Escore', 'Oscore', 'AScore', 'Cscore', 'Impulsive']

# TESTAR SE ELE CONSOME CHOCOLATE
data4['Choc_User'] = data4['Choc'].apply(lambda x: 1 if x >= 4 else 0)

#classificando os dados e features
X = data4[features]
y = data4['Choc_User']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42) #dont ask me about these paramethers :C

#treinamento
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)

#predição
y_pred = clf.predict(X_test)

#acurácia
accuracy = accuracy_score(y_test, y_pred)
print(f"Acurácia: {accuracy:.2f}")

#resultado :D
print(classification_report(y_test, y_pred))
```

Gráfico (se houver):

```
Acurácia: 0.96
              precision    recall  f1-score   support

           0       0.07      0.09      0.08        11
           1       0.98      0.97      0.98       554

    accuracy                           0.96       565
   macro avg       0.52      0.53      0.53       565
weighted avg       0.96      0.96      0.96       565
```

Legenda/explicação:
98% TENDE A CONSUMIR CHOCOLATE ACURÁCIA DE 96% :)

12. Explore a correlação entre a idade (variável Age) e a experimentação de diferentes substâncias ilícitas. Verifique se há uma tendência de aumento ou diminuição do consumo conforme a idade avança.

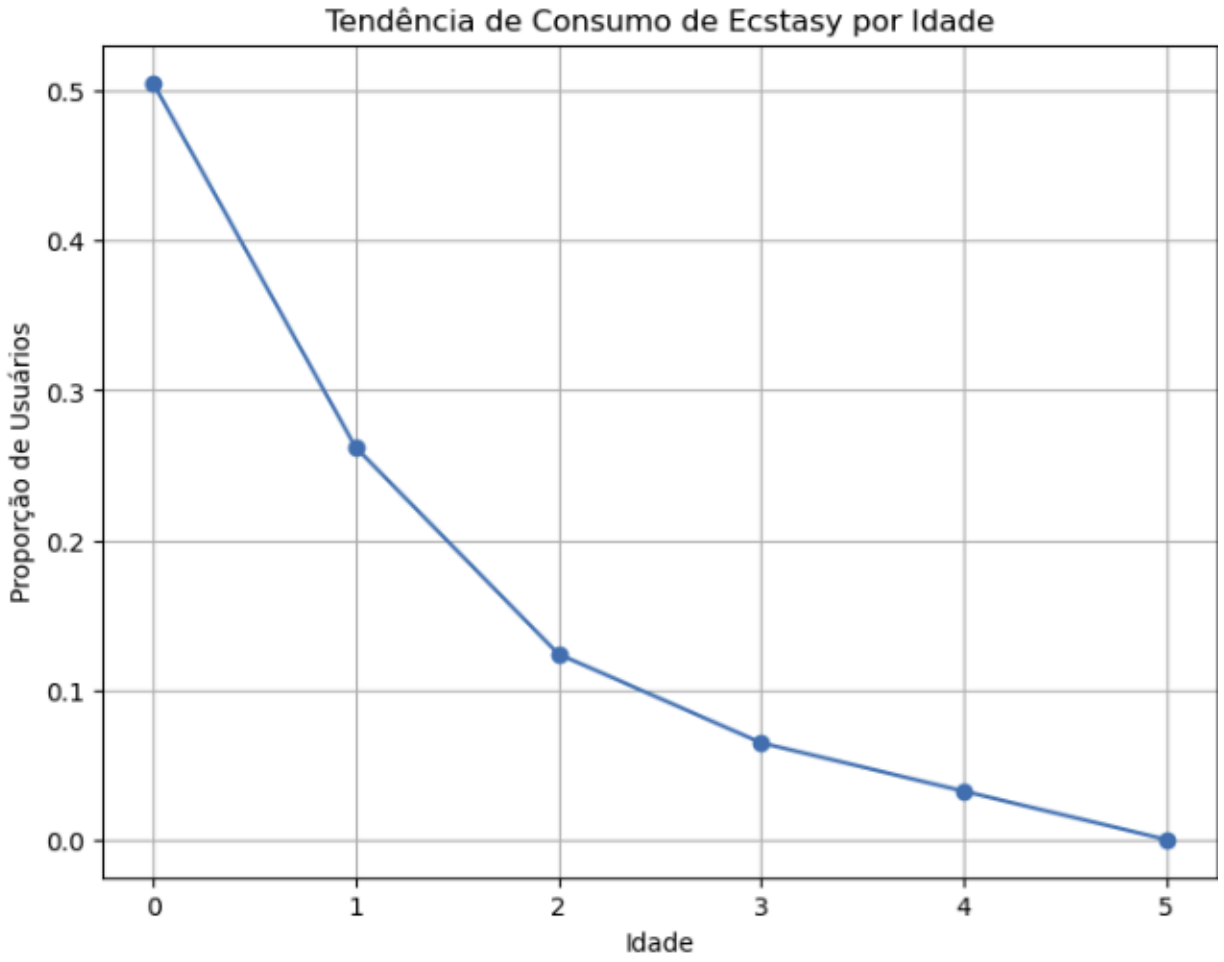
R:

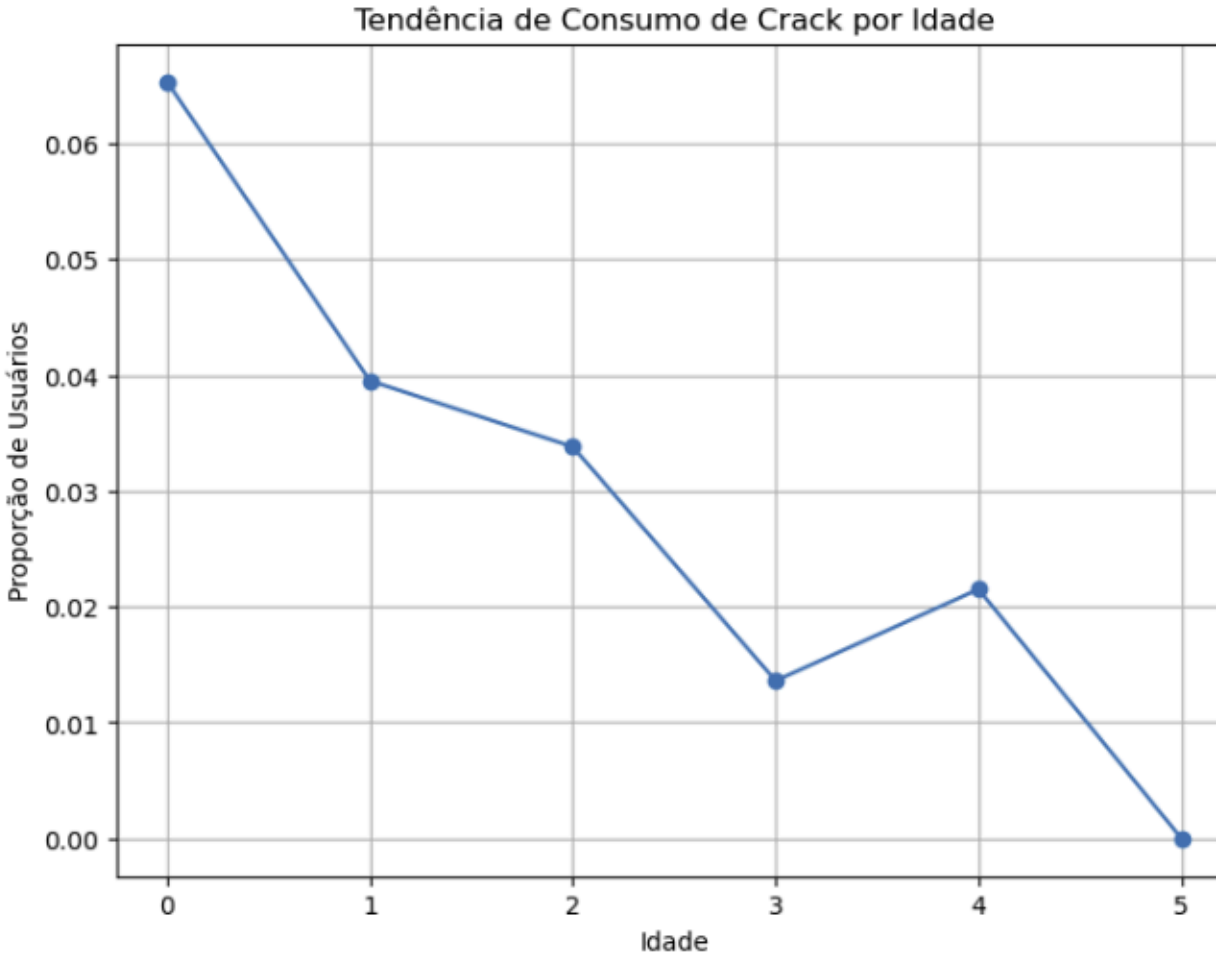
```
data4 = 'Drugs4.csv'
data4 = pd.read_csv(data4)

for substance in ilegais_lista:
    data4[f'{substance}_User'] = data4[substance].apply(lambda x: 1 if x >= 4 else 0)

#correlação entre idade e consumo de substâncias ilícitas
correlation_matrix = data4[['Age'] + [f'{substance}_User' for substance in ilegais_lista]].corr()

print("Correlação entre Idade e Consumo de Substâncias Ilícitas:")

# Visualizar as tendências
for substance in ilegais_lista:

    for i in range(len(idades)):
        print(f"Code: {i}, Age: {idades[i]}")

    plt.figure(figsize=(8, 6))
    data4.groupby('Age')[f'{substance}_User'].mean().plot(kind='line', marker='o')
    plt.title(f'Tendência de Consumo de {substance} por Idade')
```

```
plt.xlabel('Idade')
plt.ylabel('Proporção de Usuários')
plt.grid(True)
plt.show()
```

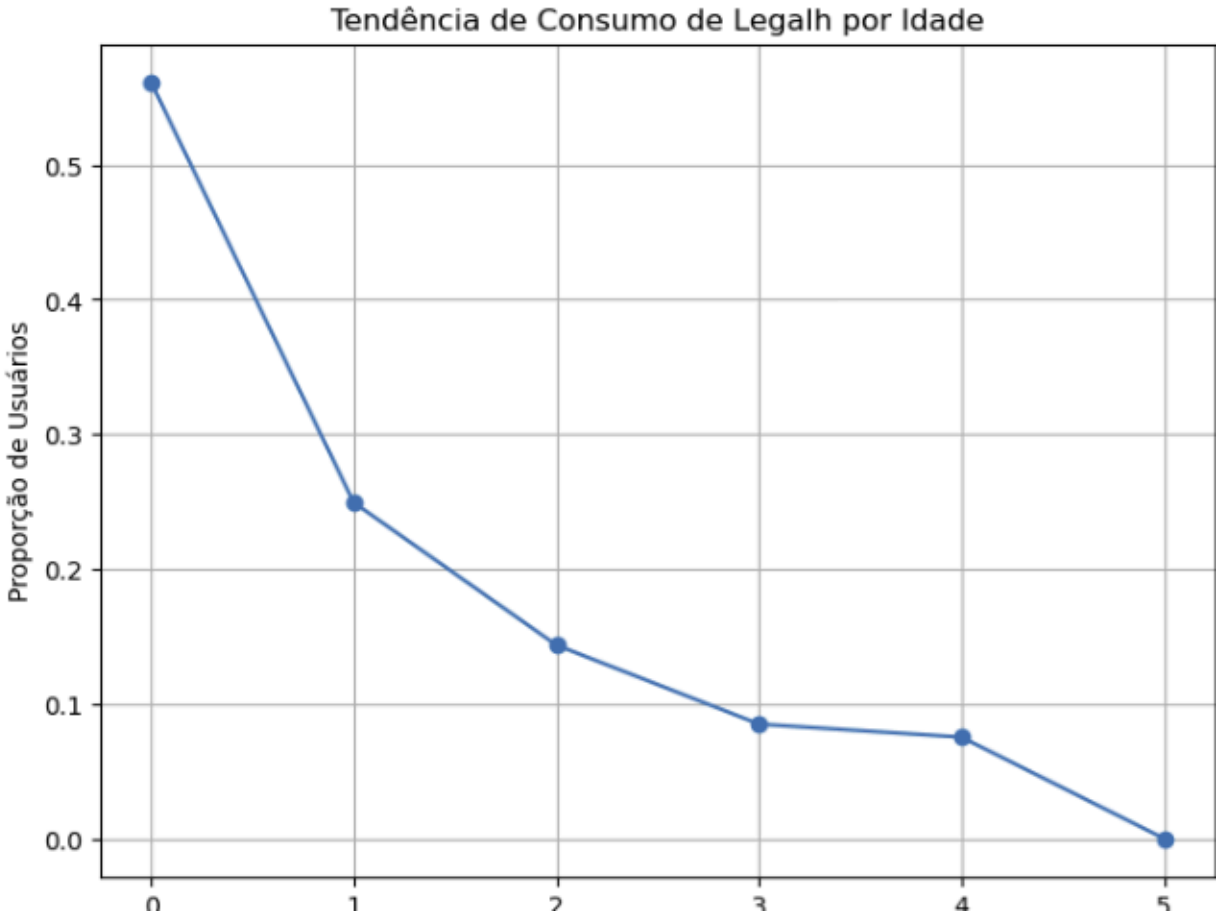
Gráfico (se houver):

```
Code: 5, Age: 65+
```

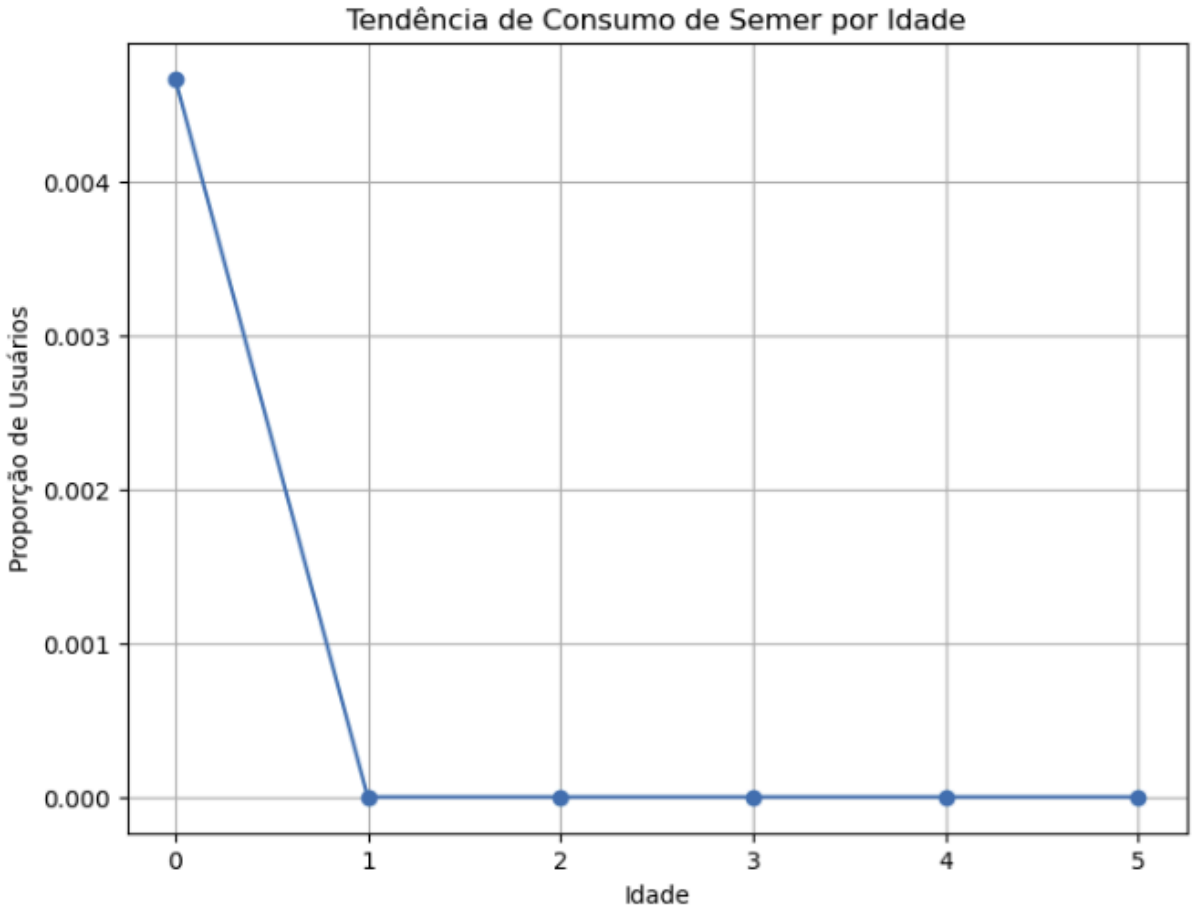



Tendência de Consumo de Crack por Idade
Proporção de Usuários
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0
1
2
3
4
5
Idade

Code: 4, Age: 55-64
Code: 5, Age: 65+



Code: 5, Age: 65+



Legenda/explicação:

Code: 0, Age: 18-24

Code: 1, Age: 25-34

Code: 2, Age: 35-44

Code: 3, Age: 45-54

Code: 4, Age: 55-64

Code: 5, Age: 65+

13. Quais são as 3 drogas mais utilizadas para cada país presente na amostra? E quais são as 3 menos utilizadas?

R:

País: Australia

Top 3 Drogas Mais Utilizadas:

Choc_User 1.000000

Alcohol_User 0.981481

Caff_User 0.981481

Name: Australia, dtype: float64

Top 3 Drogas Menos Utilizadas:

Semer_User 0.018519

Crack_User 0.037037
Heroin_User 0.055556
Name: Australia, dtype: float64

País: Canada
Top 3 Drogas Mais Utilizadas:
Caff_User 0.977011
Choc_User 0.931034
Alcohol_User 0.850575
Name: Canada, dtype: float64
Top 3 Drogas Menos Utilizadas:
Semer_User 0.000000
VSA_User 0.034483
Amyl_User 0.057471
Name: Canada, dtype: float64

País: EUA
Top 3 Drogas Mais Utilizadas:
Caff_User 1.0000
Choc_User 1.0000
Alcohol_User 0.9375
Name: EUA, dtype: float64
Top 3 Drogas Menos Utilizadas:
VSA_User 0.0
Ketamine_User 0.0
Amyl_User 0.0
Name: EUA, dtype: float64

País: New Zealand
Top 3 Drogas Mais Utilizadas:
Alcohol_User 1.0
Caff_User 1.0
Choc_User 1.0
Name: New Zealand, dtype: float64
Top 3 Drogas Menos Utilizadas:
Ecstasy_User 0.0
Amphet_User 0.0
Heroin_User 0.0
Name: New Zealand, dtype: float64

País: Other
Top 3 Drogas Mais Utilizadas:
Choc_User 0.991525

Caff_User 0.974576
Alcohol_User 0.949153
Name: Other, dtype: float64
Top 3 Drogas Menos Utilizadas:
Semer_User 0.000000
Heroin_User 0.033898
Crack_User 0.050847
Name: Other, dtype: float64

País: Republic of Ireland
Top 3 Drogas Mais Utilizadas:
Choc_User 1.0
Caff_User 1.0
Alcohol_User 0.9
Name: Republic of Ireland, dtype: float64
Top 3 Drogas Menos Utilizadas:
Crack_User 0.00
Semer_User 0.00
Amyl_User 0.05
Name: Republic of Ireland, dtype: float64

País: UK
Top 3 Drogas Mais Utilizadas:
Choc_User 0.981766
Caff_User 0.960653
Alcohol_User 0.928023
Name: UK, dtype: float64
Top 3 Drogas Menos Utilizadas:
Semer_User 0.000000
Heroin_User 0.008637
Crack_User 0.010557
Name: UK, dtype: float64

País: USA
Top 3 Drogas Mais Utilizadas:
Caff_User 0.974122
Choc_User 0.964880
Alcohol_User 0.929760
Name: USA, dtype: float64
Top 3 Drogas Menos Utilizadas:
Semer_User 0.003697
Amyl_User 0.060998
Crack_User 0.088725

```
Name: USA, dtype: float64

data = 'Drugs4.csv'
data4 = pd.read_csv(data)


drugs_lista = ['Alcohol', 'Amphet', 'Amyl', 'Benzos', 'Caff', 'Cannabis', 'Choc', 'Coke',
'Crack', 'Ecstasy', 'Heroin', 'Ketamine',
              'Legalh', 'LSD', 'Meth', 'Mushrooms', 'Nicotine', 'Semer', 'VSA']

paises = {
    0: 'Australia',
    1: 'Canada',
    2: 'EUA',
    3: 'New Zealand',
    4: 'Other',
    5: 'Republic of Ireland',
    6: 'UK',
    7: 'USA'
}

data4['Country'] = data4['Country'].map(country_mapping)

for substance in drugs_lista:
    data4[f'{substance}_User'] = data4[substance].apply(categorize_user)

#agrupa país e calcula média de uso de cada substância

country_drug_use = data4.groupby('Country')[[f'{substance}_User' for substance in
drug_columns]].mean()

#função para encontrar as 3 drugs

def top_bottom_3(tabela):

    top_3 = tabela.sort_values(ascending=False).head(3)
    bot_3 = tabela.sort_values(ascending=True).head(3)

    return top_3, bot_3
```

```
#função para cada país

paises = {}

for country in country_drug_use.index:

    top_3, bottom_3 = top_bottom_3(country_drug_use.loc[country])

    paises[country] = {
        'top_3': top_3,
        'bottom_3': bottom_3
    }

print('----------------------')
for country, result in results.items():
    print(f"País: {country}")
    print("Top 3 Drogas Mais Utilizadas:")
    print(result['top_3'])
    print("Top 3 Drogas Menos Utilizadas:")
    print(result['bottom_3'])
    print('----------------------')
```

Gráfico (se houver):

Legenda/explicação: