universidade de aveiro
theoria poiesis praxis

universidade de aveiro
dcm
departamento de ciências médicas

Diabetes Dataset

Programação e Algoritmos em Ciências

João Pacheco nº 97978

Jorge Marques nº 98941

Joana dos Santos nº 98903

Diabetes

Tipo 1

Gestacional

Tipo 2

Base de Dados

Kaggle

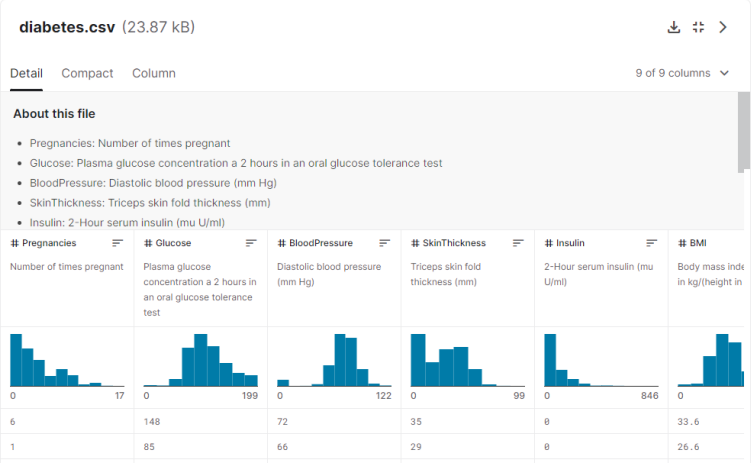
Disponível em:
https://www.kaggle.com/datasets/mathchi/diabetes-data-set

Composição

- 768 pacientes do sexo feminino;
- Pelo menos 21 anos de idade;
- Background genético Pima Indian.

Propósito

Prever se o paciente tem diabetes com base nos meios de diagnóstico que compõem as diferentes variáveis



National Institute of Diabetes and Digestive and Kidney Diseases

Variáveis da base de dados

- Pregnancies
- Glucose
- Blood Pressure
- Skin Thickness
- Insulin
- BMI
- Diabetes Pedigree Function
- Age
- Outcome

+

- Glycemia Values

Legenda:

- Variável qualitativa
- Variável quantitativa

Objetivo

- Leitura e manipulação da base de dados
- Construção e visualização de informação gráfica e estatística da nossa base de dados
- Construção de uma interface estilo menu para o utilizador poder interagir livre e personalizadamente com a base de dados
- Extração em ficheiros do tipo .png e .txt dos resultados obtidos pelo utilizador

Programa

pip3 install “package”

```
import pandas as pd
```

```
from pandas_profiling import ProfileReport
```

```
import numpy as np
```

```
import seaborn as sns
```

```
import matplotlib.pyplot as plt
```

Opção 1

universidade de aveiro dcm departamento de ciências médicas

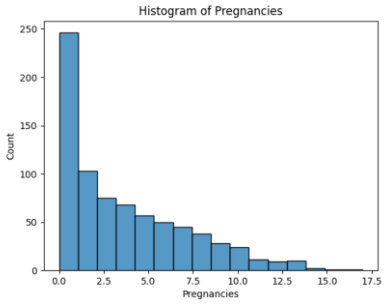
Código

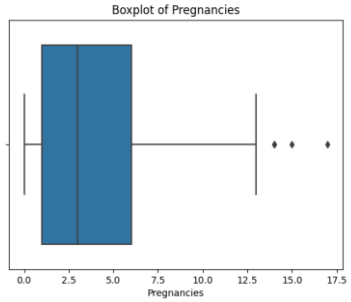
```
def var_num(dataframe, variaveis):
    for i in variaveis:
        fig, ax = plt.subplots()
        ax.axis('off')
        ax.axis('tight')
        df_var = dataframe[i].describe()
        colnames = df_var.axes[0].tolist()
        tabela = ax.table(cellText =
[df_var.values.round(2)],colLabels=colnames, loc =
'center')
        tabela.auto_set_font_size(False)
        tabela.set_fontsize(8)
        plt.title(f"Table of statistical values of {i}")
        plt.show()

        sns.histplot(data = dataframe, x=i)
        plt.title(f"Histogram of {i}")
        plt.show()

        sns.boxplot(data = dataframe, x = i)
        plt.title(f"Boxplot of {i}")
        plt.show()
```

Output

Table of statistical values of Pregnancies

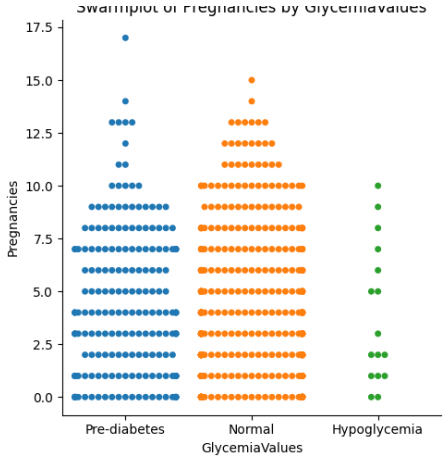
count	mean	std	min	25%	50%	75%	max
768.0	3.85	3.37	0.0	1.0	3.0	6.0	17.0

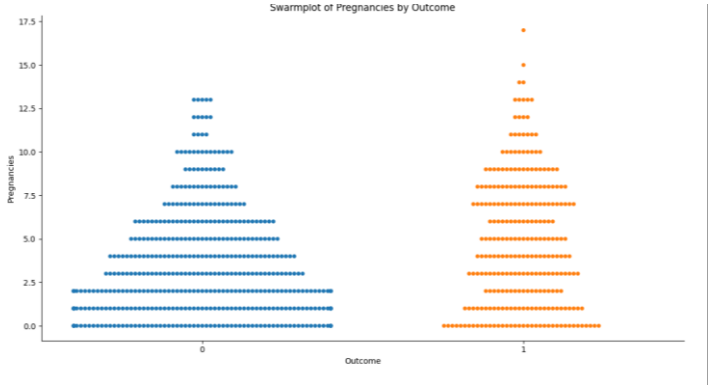




Opção 1


```python
def swarmplotvar(dataframe, variaveis, vcategorical = None):
    for s in variaveis:
        sns.catplot(x = vcategorical, y = s, hue = vcategorical, kind = "swarm", data = dataframe)
        plt.title(f"Swarmplot of {s} by {vcategorical}")
        plt.savefig("swarmplot.png")
        plt.show()
```

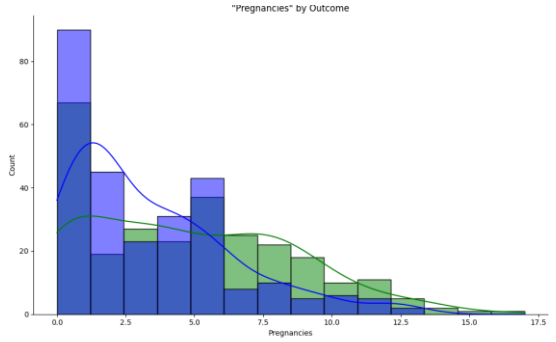


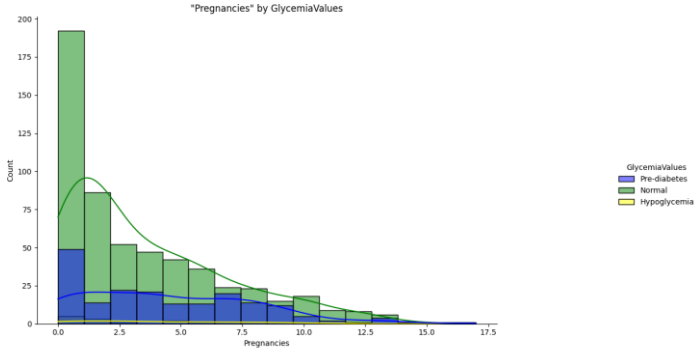


Opção 1

universidade de aveiro dcm departamento de ciências médicas

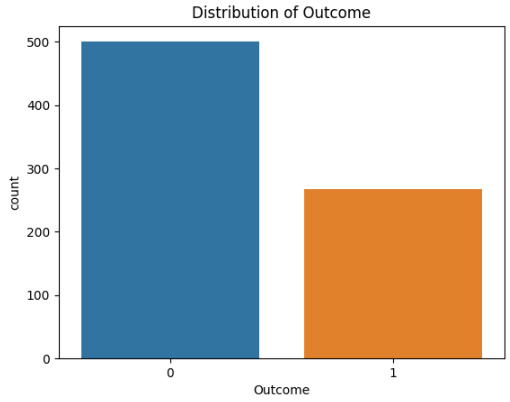
```
def hist_vcat(dataframe, vcategorical,
variaveis):

counter = 0
    for var in variaveis:
        counter += 1
        print(counter, ':', var)
        sns.displot(data = df_bal, kde=True, x
= dataframe[str(var)], hue=vcategorical,
palette=cores)
        plt.title(f'Histogram of "{var}" by
{vcategorical}')
    plt.plot()
    plt.savefig("histogram_variaveis_numericas.
png")
    plt.show()
```

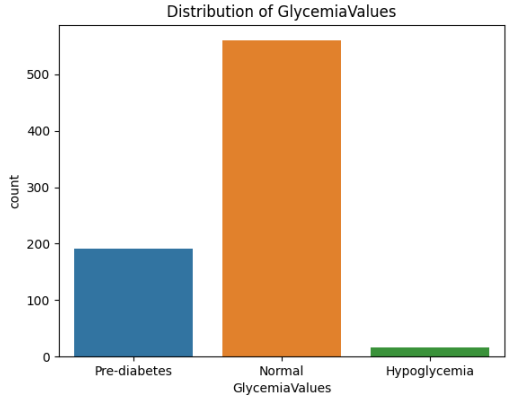




Opção 2


```
def categorica_values(dataframe, vcategorical):
    order = []
    if vcategorical == "Outcome":
        order = [0,1]
    else:
        order = ['Hypoglycemia','Normal','Pre-
diabetes']
    plt.figure()
    sns.countplot(x = dataframe[vcategorical], data =
dataframe, hue_order = order)
    plt.title(f"Distribution of {vcategorical}")
    plt.savefig("barplot_variavel_categorica.png")
    plt.show()
```

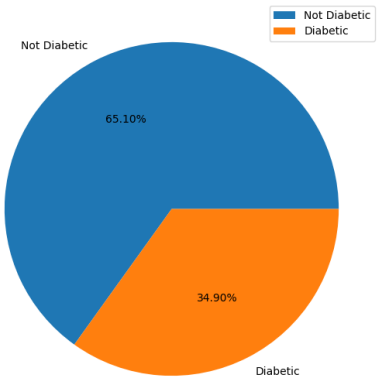


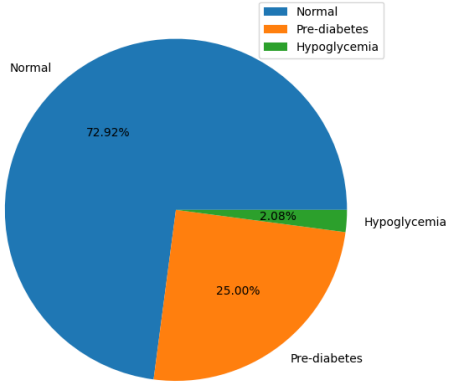


Opção 2

universidade de aveiro dcm departamento de ciências médicas

```
def circular(dataframe, vcategorical):
    labels = []
    if vcategorical == "Outcome":
        labels = {'Not Diabetic', 'Diabetic'}
    else:
        labels = {'Normal':'Normal','Pre-
diabetes':'Pre-
diabetes','Hypoglycemia':'Hypoglycemia'}

    plt.figure(figsize = (10,7))
    plt.pie(dataframe[vcategorical].value_count
s(), labels = labels, autopct = '%0.02f%%')
    plt.legend()
    plt.savefig("circular_var_categorica.png")
    plt.show()
```

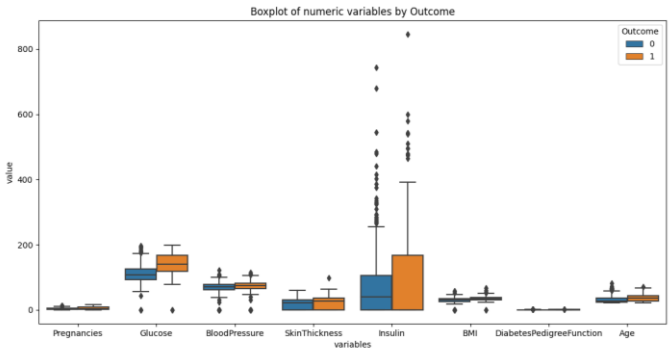


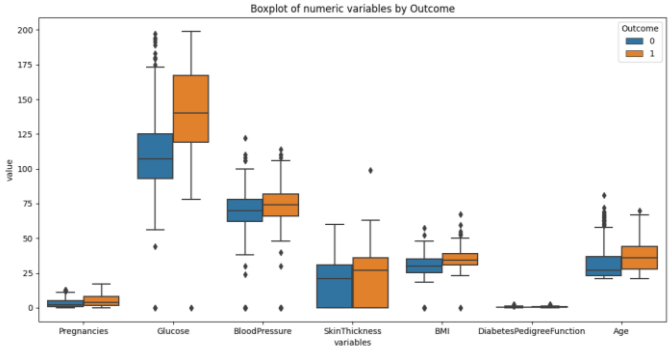


Opção 3

universidade de aveiro dcm departamento de ciências médicas

```
def boxplot_all(dataframe, drop_values,
has_outcome = False):
    diabetesbp = dataframe.drop(drop_values, axis
= 1)
    diabetes_melted = pd.melt(diabetesbp, id_vars
= "Outcome", var_name = "variables", value_name =
"value")

    plt.figure(figsize = (15, 15))

    sns.boxplot(data = diabetes_melted, x =
"variables", y = "value", hue = "Outcome") if
has_outcome else sns.boxplot(data =
diabetes_melted, x = "variables", y = "value")
    plt.title("Boxplot of numeric variables by
Outcome") if has_outcome else plt.title("Boxplot
of numeric variables")
    plt.savefig("boxplot_all.png")
    plt.show()
```

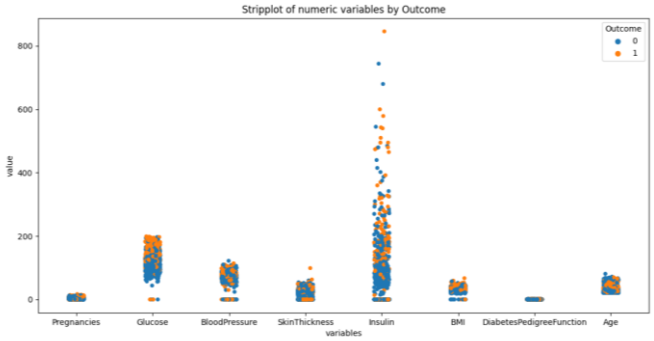


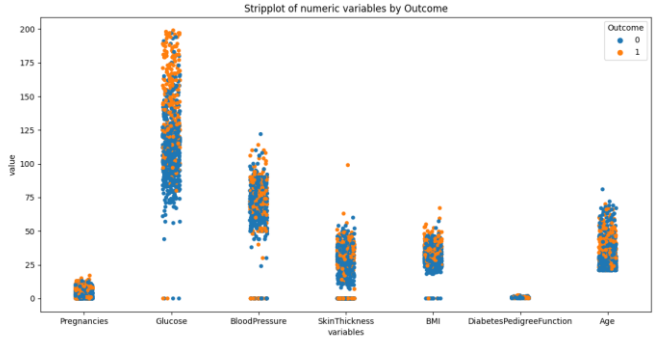


Opção 3

universidade de aveiro dcm departamento de ciências médicas

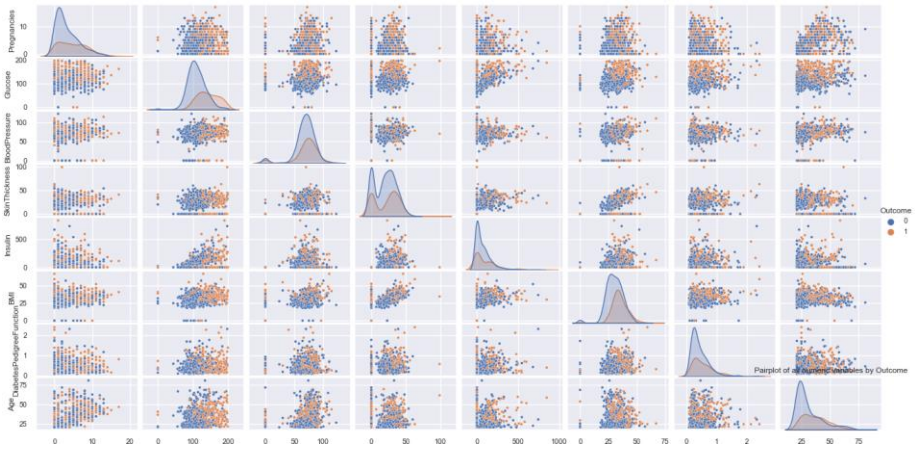
```
def stripvar(dataframe, drop_values,
has_outcome = False):
    diabetesbp= dataframe.drop(drop_values,
axis = 1)
    diabetes_melted = pd.melt(diabetesbp,
id_vars = "Outcome", var_name = "variables",
value_name = "value")

    plt.figure(figsize = (15, 15))
    sns.stripplot(x = "variables", y = "value",
hue = "Outcome", data = diabetes_melted) if
has_outcome else sns.stripplot(x = "variables",
y = "value", data = diabetes_melted)
    plt.title("Stripplot of numeric variables
by Outcome")  if has_outcome else
plt.title("Stripplot of numeric variables")
    plt.savefig("stripplot.png")
    plt.show()
```





Opção 3

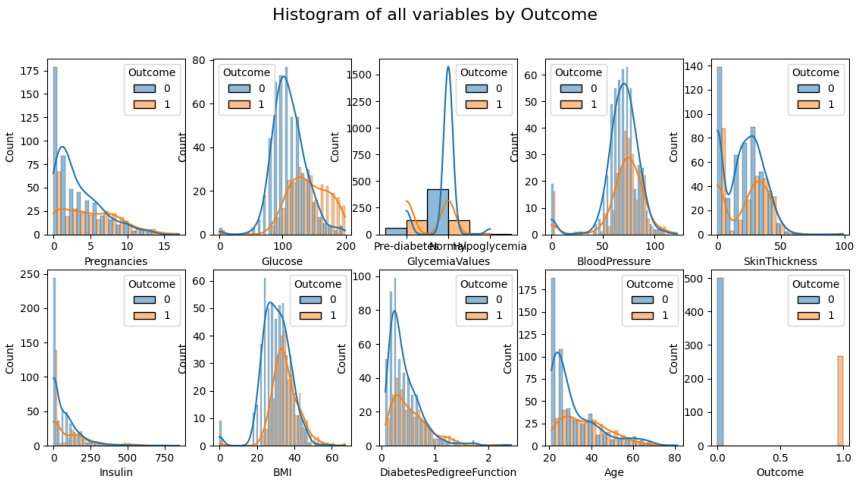

```python
def pairplt(dataframe, vcategorical = None):
    plt.figure()
    sns.set(font_scale = 0.7)
    sns.pairplot(dataframe, hue =
vcategorical, diag_kind = "kde", plot_kws =
{"s": 8})
    plt.title(f"Pairplot of all numeric
variables by {vcategorical}")
    plt.savefig("pairplot.png")
    plt.show()
```



Opção 3

universidade de aveiro dcm departamento de ciências médicas

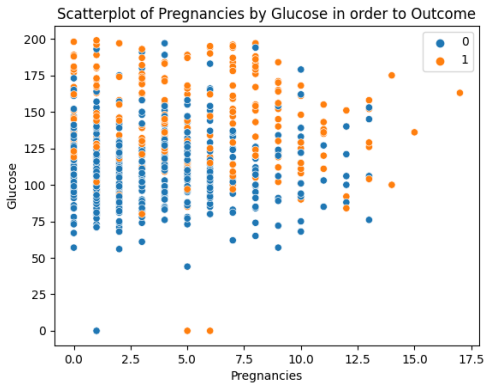
```
def hist_total(dataframe,has_outcome = False):
    variaveis = dataframe.columns
    print(variaveis)
    counter = 1
    for i in variaveis:
        print(counter, ':', i)
        plt.subplot(2,5,counter)
        sns.histplot(data = dataframe, x = datafram
has_outcome else sns.histplot(data = dataframe, x =
        counter += 1

    plt.suptitle("Histogram of all variables by Out
variables", fontsize = 16)
    plt.plot()
    plt.savefig("histogram_all.png")
    plt.show()
```

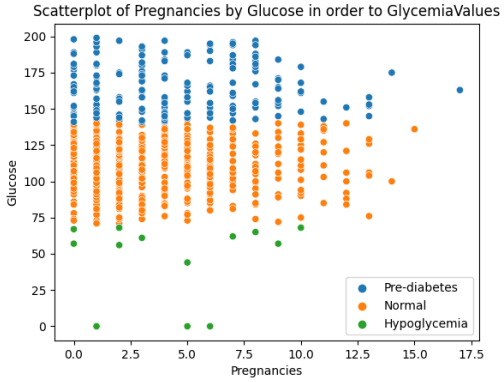


Opção 4

universidade de aveiro dcm departamento de ciências médicas

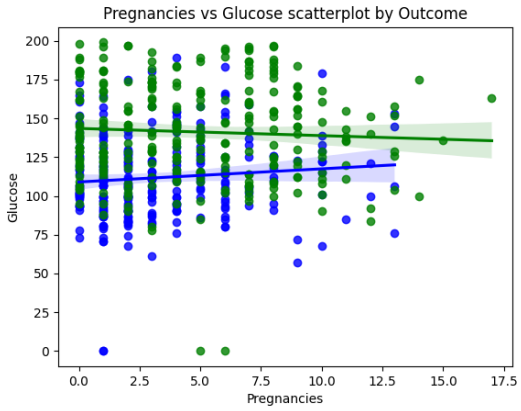
```
def scatterplt(dataframe, variavel_1, variavel_2,
vcategorical = None):
    print(f"Variável no eixo dos xx: {variavel_1}
\nVariável no eixo dos yy: {variavel_2}")
    sns.scatterplot(data = dataframe, x = variavel_1,
y = variavel_2, hue = vcategorical)
    plt.title(f"Scatterplot of {variavel_1} by
{variavel_2} in order to {vcategorical}") if
vcategorical is not None else plt.title(f"Scatterplot
of {variavel_1} by {variavel_2}")
    plt.legend()
    plt.savefig("scatterplot.png")
    plt.show()
```

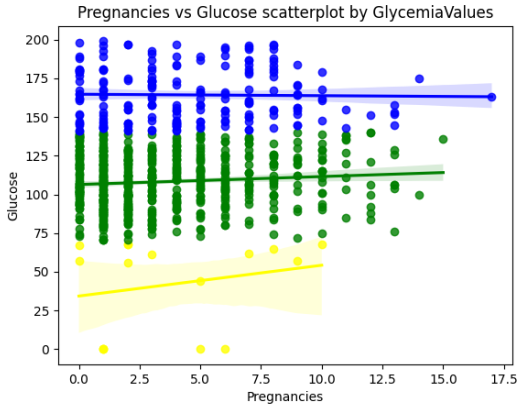




Opção 4

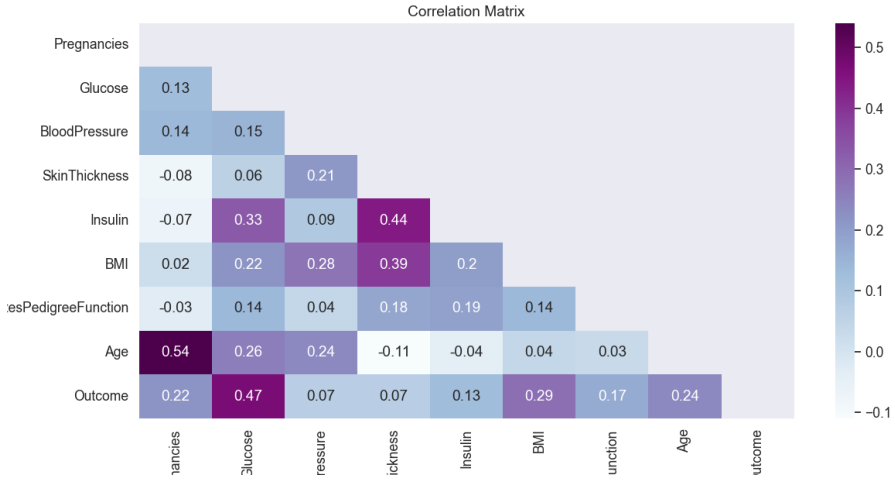

```
def regressao (dataframe, variavel_1, variavel_2,
    if vcategorical == "Outcome":
        sns.regplot(x = variavel_1, y = variavel_
        sns.regplot(x = variavel_1, y = variavel_

    elif vcategorical=="GlycemiaValues":
        sns.regplot(x = variavel_1, y = variavel_
        sns.regplot(x = variavel_1, y = variavel_
        sns.regplot(x = variavel_1, y = variavel_

    else:
        sns.regplot(x = variavel_1, y = variavel_

    plt.title(f"{variavel_1} vs {variavel_2} scat
    plt.savefig("regressao.png")
    plt.show()
```





Opção 4


```python
def matrcorr(dataframe):
    corr = dataframe.corr().round(2)
    plt.figure(figsize = (14, 10))
    sns.set(font_scale = 1.15)
    mask = np.zeros_like(corr)
    mask[np.triu_indices_from(mask)] = True
    sns.heatmap(corr, annot = True, cmap =
'BuPu', mask = mask, cbar = True)
    plt.title('Correlation Matrix')
    plt.savefig("correlation_matrix.png")
    plt.show()
```



Opção 5

universidade de aveiro dcm departamento de ciências médicas

```
elif opcao == 5:
        lista=get_lista_variaveis()
        escolha6=menu_5()
        if escolha6==0:
            list_calcs_to_write=[]
            for c in lista:
                lista_valores=diabetesdf[c]
.values
                print(f"Média da variável
{c}")
                calc_valor =
np.mean(lista_valores).round(2)
                print(calc_valor)
                list_calcs_to_write.append(
f"Media da variavel {c}: {calc_valor} \n")
            save_file(list_calcs_to_write)
            terminar()
```

x4

```
<class 'list'>
    0: Média
    1: Média Ponderada
    2: Mediana
    3: Variância
    4: Desvio Padrão
Escolha o cálculo do menu acima que pretende efetuar:  0
Média da variável Pregnancies
3.85
Deseja guardar o cálculo num ficheiro? Sim ou Não?
s
Escolha o nome do seu ficheiro: MediaPregnancies
```

MediaPregnancies - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

Media da variavel Pregnancies: 3.85

Opção 6

universidade de aveiro dcm departamento de ciências médicas

```
relatorio = ProfileReport(diabetesdf,
title = "Relatório da Análise da Base de
Dados Diabetes")
        diabetesdf.profile_report()
        relatorio.to_file("diabetes_repor
t.html")
```

Overview

Overview | Alerts 13 | Reproduction

Dataset statistics

Number of variables	10
Number of observations	768
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	60.1 KiB
Average record size in memory	80.2 B

Variable types

Numeric	8
Categorical	2

Overview | Alerts 13 | Reproduction

Alerts

Pregnancies is highly overall correlated with Age	High correlation
Glucose is highly overall correlated with GlycemiaValues	High correlation
SkinThickness is highly overall correlated with Insulin	High correlation
Insulin is highly overall correlated with SkinThickness	High correlation
Age is highly overall correlated with Pregnancies	High correlation
GlycemiaValues is highly overall correlated with Glucose	High correlation
BloodPressure is highly overall correlated with BMI	High correlation
BMI is highly overall correlated with BloodPressure	High correlation
Pregnancies has 111 (14.5%) zeros	Zeros
BloodPressure has 35 (4.6%) zeros	Zeros
SkinThickness has 227 (29.6%) zeros	Zeros
Insulin has 374 (48.7%) zeros	Zeros
BMI has 11 (1.4%) zeros	Zeros

Limitações e trabalho futuro

Trabalho futuro:

- Manipular a base de dados tendo a opção de retirar outliers e/ou os zeros.

Limitações:

- Dificuldade inicial de trabalhar em equipa uma vez que eram três pessoas a tentar escrever o mesmo código - ultrapassado com recurso ao GitHub;
- Pouco conhecimento de programação, especialmente a nível prático - ao longo do projeto adquirimos mais conhecimentos ao pesquisar por iniciativa própria como se construía o gráfico desejado, tentando sempre otimizar o código

Obrigado pela atenção!

Referências:

1. Mayo Clinic Staff. (2022, March 24). Glucose tolerance test. Mayo Clinic. Retrieved December 10, 2022, from https://www.mayoclinic.org/tests-procedures/glucose-tolerance-test/about/pac-20394296
2. Diabetes overview. Diabetes Symptoms, Causes, & Treatment | ADA. (n.d.). Retrieved December 12, 2022, from https://diabetes.org/diabetes
3. Waskom, M. L. (2021). Statistical Data Visualization#. seaborn: statistical data visualization. Retrieved December 14, 2022, from https://seaborn.pydata.org/
4. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Matplotlib documentation - Matplotlib 3.6.2 documentation. Retrieved December 13, 2022, from https://matplotlib.org/stable/index.html