

# **Relatório do projeto de Programação e Algoritmos em Ciências**

## **Programação e Algoritmos em Ciências**

Joana dos Santos nº 98903

João Pacheco nº 97978

Jorge Marques nº 98941

## Introdução

Neste projeto, inserido na UC de Programação e Algoritmos em Ciências, desenvolvemos um programa de código, recorrendo à linguagem de programação *Python* (v3.8), de forma a realizar uma análise de dados a uma *database* escolhida.

## Objetivos

- 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.

## Base de dados

A base de dados utilizada está disponível no Kaggle<sup>1</sup>, intitulada “*Diabetes Dataset*”, uma base de dados com o objetivo de prever se um paciente tem diabetes com base nos meios de diagnóstico que compõem as diferentes variáveis. Esta base de dados foi escolhida devido ao tamanho da amostra razoável, número de variáveis ideal e pelo interesse, da nossa parte, em fazer a análise de dados a este *dataset* e observar as correlações entre as variáveis que o compõem.

Esta base de dados é então composta por 768 pacientes do sexo feminino, de pelo menos 21 anos de idade e com um *background* genético da linhagem “*Pima Indian*”. Ao todo, a base de dados é composta por 9 variáveis, todas elas variáveis quantitativas contínuas, exceto a variável “*Pregnancies*”, uma variável quantitativa discreta e a variável “*Outcome*”, uma variável qualitativa nominal. Para além disso, é possível classificar as variáveis quanto à sua utilização, pelo que 8 das variáveis são variáveis preditivas enquanto a variável “*Outcome*” é o objeto de estudo, ou seja, a variável que se pretende prever.

Entretanto, foi realizada uma transformação da nossa parte à base de dados, sendo criada e adicionada uma nova variável, designada “*Glycemiavalues*” que categoriza os pacientes tendo em conta a sua concentração de glucose no sangue. Os *cutoffs* para a criação desta variável foram utilizados de acordo com a literatura<sup>2</sup>, sendo os seguintes: até 70 mg/dL (hipoglicemia); entre 70 e 140 mg/dL (normal); entre 140 e 199 mg/dL (pré-diabetes) e maior que 200 mg/dL (diabetes).

Considerámos necessária a adição desta variável porque o teste de tolerância à glucose, que mede os valores desta variável, pode ser utilizado para rastrear a diabetes tipo 2, tornando esta variável um bom preditor para o desenvolvimento da doença.

## Programa

O programa de código desenvolvido permite-nos realizar a análise de dados ao *dataset* referido. Aquando da abertura do programa o utilizador é remetido para um menu de utilização e confrontado com a possibilidade de escolher entre várias opções, sendo incitado a introduzir um número de 0 a 7 em que cada número corresponde a uma das seguintes opções: visualizar a base de dados, obter análises estatísticas para variáveis numéricas individualmente, obter análises estatísticas para variáveis categóricas individualmente, obter análises estatísticas para cada as variáveis em conjunto, ver interações e correlações entre as variáveis, calcular medidas amostrais para as variáveis, extrair um relatório geral da base de dados ou sair do programa, respetivamente. Cada uma das opções, exceto a opção de abandonar o programa e a de extrair um relatório geral da base de dados, vai levar o utilizador a um outro submenu em que terá opções de escolha mais específicas sobre a análise estatística a extrair para a opção escolhida anteriormente, e também uma opção de voltar atrás ao menu principal do programa.

## Dependências do Projeto

Para possibilitar a construção dos gráficos e análise estatística pretendidos neste projeto tivemos de proceder à instalação no terminal de comandos dos pacotes a utilizar, indispensáveis para a execução do código feito, utilizando para tal o comando *pip3 install {package em causa}*.

Neste programa foram utilizados os pacotes “*Pandas*” e o seu módulo “*Pandas profiling*”, “*Numpy*”, o módulo “*Matplotlib.pyplot*” do package “*Matplotlib*” e “*Seaborn*”.

### *Pandas*

*Pandas* é um pacote de manipulação e análise de dados em *Python* para dados tabulares. As funcionalidades do *Pandas* incluem transformações de dados, classificar linhas e obter subconjuntos, calcular estatísticas resumidas e remodelar e unir *dataframes*, fornecendo estruturas de dados rápidas, flexíveis e expressivas projetadas para trabalhar com "dados rotulados ou relacionais" fáceis e intuitivos.

Neste projeto, usamos o package *Pandas* para ler a nossa base de dados.

### *Pandas profiling*

*Pandas profiling* é um módulo *Python* que gera relatórios de perfil a partir de um *dataframe* do *pandas*. *Pandas profiling* estende a *dataframe* do *pandas* com *df.profile\_report()*, que gera automaticamente um relatório interativo univariado e

multivariado padronizado em formato web com todas as informações facilmente disponíveis para compreensão dos dados.

No nosso trabalho usamos *Pandas profiling* para gerar um relatório interativo dos nossos dados

### *Numpy*

*Numpy* significa Numerical *Python* e é uma das bibliotecas científicas mais úteis na programação *Python*. O *Numpy* fornece funções que podem ser chamadas pelo utilizador, o que o torna especialmente útil para manipulações de dados. O *Numpy* fornece um objeto de matriz até 50 vezes mais rápido que as listas tradicionais do *Python*. Uma matriz *Numpy* é semelhante ao tipo de lista embutido do *Python*, mas as matrizes *Numpy* oferecem armazenamento e operações de dados muito mais eficientes à medida que o conjunto de dados cresce. As matrizes *Numpy* são mais compactas, permitem acesso mais rápido na leitura e gravação de itens e são mais convenientes e eficientes em geral.

Neste projeto usamos o pacote *Numpy* para construir uma matriz de correlações.

### *Matplotlib.pyplot*

*Matplotlib* é um pacote para visualização de dados e biblioteca de construção gráfica para *Python* e para a extensão numérica *Numpy*. O pacote *Matplotlib* é uma biblioteca do *Python* utilizada para criar gráficos 2D, apresentando uma série de possibilidades gráficas, como gráficos de barras, linhas, pizza, histogramas, entre muitos outros. O conjunto de funções disponíveis em *matplotlib.pyplot* permite a criação de uma figura e uma área padrão para exibir o gráfico na figura, exigindo apenas que o programador desenhe as linhas na área do gráfico, decore o gráfico com rótulos e assim por diante.

### *Seaborn*

*Seaborn* é uma biblioteca de visualização de dados em *Python* baseada no *Matplotlib*. Ele fornece uma interface de alto nível para desenhar gráficos estatísticos atraentes e informativos. As funções de construção gráfica *Seaborn* operam em *dataframes* e arrays contendo *datasets* inteiros e executam internamente o mapeamento semântico necessário e a agregação estatística para produzir gráficos informativos. *Seaborn* fornece uma API sobre o *Matplotlib* que oferece opções sensatas para estilos de design gráfico e padrões de cores, define funções simples de alto nível para tipos de gráficos estatísticos comuns e integra-se com a funcionalidade fornecida pelos *dataframes* do *Pandas*.

Neste trabalho os pacotes *matplotlib.pyplot* e *seaborn* foram usados na criação de vários gráficos.

## Código

### Leitura dos Dados

Primeiramente, procedemos à leitura da nossa base de dados, um ficheiro .csv, convertendo-o a uma *dataframe* usando o pacote *pandas*, para uma melhor manipulação, visualização e análise dos dados.

Este projeto é composto por dois ficheiros *Python*, denominados *Plots.py* e *main.py*. O ficheiro *Plots.py* contém um conjunto de funções que permitem a construção e visualização dos gráficos e tabelas do programa. No ficheiro *main.py* estas funções são chamadas para serem incorporadas no menu de utilização, tendo ainda neste ficheiro mais funções necessárias para a construção do menu, como, por exemplo funções que pedem e verificam menus e ainda que dão print aos mesmos. Quando o utilizador executa o programa, aparece um menu de utilização principal com 8 opções que permitem uma livre e personalizada visualização e análise da base de dados.

```
[Menu] Escolhe uma das seguintes opções:
0 - Visualização da base de dados
1 - Análise individual de variáveis numéricas
2 - Análise individual de variáveis categóricas
3 - Análise conjunta de variáveis
4 - Interações e Correlações entre as variáveis
5 - Cálculos das medidas amostrais das variáveis
6 - Relatório geral da base de dados
7 - Sair
Opção: █
```

### Funcionalidades do programa

A opção 0 – “Visualização da base de dados” – permite ao utilizador visualizar a *dataframe* diabetes e ainda algumas informações relativas à mesma, como o tipo de variável, de forma imediata.

Base de Dados Diabetes:

	Pregnancies	Glucose	GlycemiaValues	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	Pre-diabetes	72	35	0	33.6	0.627	50	1
1	1	85	Normal	66	29	0	26.6	0.351	31	0
2	8	183	Pre-diabetes	64	0	0	23.3	0.672	32	1
3	1	89	Normal	66	23	94	28.1	0.167	21	0
4	0	137	Normal	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...	...
763	10	101	Normal	76	48	180	32.9	0.171	63	0
764	2	122	Normal	70	27	0	36.8	0.340	27	0
765	5	121	Normal	72	23	112	26.2	0.245	30	0
766	1	126	Normal	60	0	0	30.1	0.349	47	1
767	1	93	Normal	70	31	0	30.4	0.315	23	0

[768 rows x 10 columns]

```
Pretende ainda visualizar:  
0- Tabela com as primeiras 20 observações  
1- Tabela com medidas estatísticas das variáveis  
2- Voltar ao Menu Principal  
Opção: █
```

O submenu da opção 0 permite ainda ao utilizador visualizar 2 tabelas distintas com informações sobre a *dataframe* (primeiras 20 observações e medidas estatísticas das variáveis) ou regressar ao menu inicial caso não seja pretendido visualizar nenhuma informação adicional para além da já disposta.

A opção 1 - “Análise individual de variáveis numéricas” – permite ao utilizador escolher uma ou mais variáveis que pretenda analisar. Caso o utilizador escolha a mesma variável duas vezes ou um número não existente no submenu, aparece uma mensagem a pedir ao utilizador para inserir números válidos.

```
0: Pregnancies  
1: Glucose  
2: BloodPressure  
3: SkinThickness  
4: Insulin  
5: BMI  
6: DiabetesPedigreeFunction  
7: Age  
Escolha a variável que deseja analisar utilizando os números indicados no menu acima: █
```

Após a escolha da variável pelo utilizador é dada a escolha de selecionar mais variáveis ou prosseguir com a análise da variável escolhida.

```
Escolha a variável que deseja analisar utilizando os números indicados no menu acima:2  
Deseja escolher outra variável para análise? Sim ou Não?
```

Escolhidas as variáveis, o utilizador pode agora escolher de entre várias opções gráficas o que pretende visualizar acerca das variáveis escolhidas. Independentemente do número de variáveis escolhidas, depois de feita a seleção das variáveis e das representações gráficas que o utilizador pretende visualizar, o programa apresentará os gráficos por ordem de seleção das variáveis.

```
0: Representação estatística e gráfica  
1: Swarmplot  
2: Histograma com função densidade  
Escolha o gráfico que pretende visualizar: █
```

A opção 2 - “Análise individual de variáveis categóricas” - permite ao utilizador visualizar um gráfico de barras ou um gráfico circular das variáveis categóricas (“*Outcome*” e “*GlycemiaValues*”) da nossa base de dados.

```
0: Gráfico de Barras  
1: Gráfico Circular  
Escolha o gráfico do menu acima que pretende visualizar:
```



Escolhido o gráfico que o utilizador pretende visualizar, é inquirida qual a variável categórica que o utilizador pretende ver:

```
G: GlycemiaValues
O: Outcome
Escolha a variável que pretende analisar, do menu acima:
```

A opção 3 - “Análise conjunta de variáveis” – permite ao utilizador visualizar um conjunto de gráficos para todas as variáveis, de forma conjunta. Após a escolha do gráfico a visualizar pelo utilizador, são feitas um conjunto de perguntas que dão ao utilizador a possibilidade de remover uma ou mais variáveis da análise e, ainda, observar os gráficos feitos em função da variável “Outcome”, “Glycemiavalues” ou de nenhuma destas.

```
0: Boxplot
1: Stripplot
2: Pairplot
3: Histograma + Função densidade
Escolha o gráfico do menu acima que pretende visualizar:
```

```
Deseja eliminar alguma variável da análise? Sim ou Não?s
0: Pregnancies
1: Glucose
2: BloodPressure
3: SkinThickness
4: Insulin
5: BMI
6: DiabetesPedigreeFunction
7: Age
Escolha a variável utilizando os números indicados no menu acima:4
Deseja escolher outra variável? Sim ou Não?
n
Deseja fazer em função da variável 'Outcome'? Sim ou Não?s
Deseja continuar a análise estatística? Escreva 'Sim' para continuar ou 'Não' para terminar
n
```

A opção 4 - “Interações e Correlações entre as variáveis” – permite ao utilizador visualizar interações entre as variáveis em gráficos de dispersão ou de regressão linear e também observar as correlações entre as variáveis numa matriz de correlações.

```
0: Gráficos de dispersão
1: Regressão linear
2: Matriz de correlações
Escolha o gráfico do menu acima que pretende visualizar:
```

Se o utilizador escolher gráficos de dispersão ou de regressão linear, é necessário um input adicional do utilizador para selecionar que 2 variáveis quer escolher para executar o gráfico. Adicionalmente a análise das duas variáveis escolhidas pode ser feita em função das variáveis qualitativas “Outcome”, “Glycemiavalues” ou nenhuma destas, exceto na opção 2, matriz de correlações.

```
Variável do eixo dos xx
0: Pregnancies
1: Glucose
2: BloodPressure
3: SkinThickness
4: Insulin
5: BMI
6: DiabetesPedigreeFunction
7: Age
Escolha a variável que deseja analisar utilizando os números indicados no menu acima:
Variável do eixo dos yy
0: Pregnancies
1: Glucose
2: BloodPressure
3: SkinThickness
4: Insulin
5: BMI
6: DiabetesPedigreeFunction
7: Age
Escolha a variável que deseja analisar utilizando os números indicados no menu acima:
Deseja fazer em função da variável Outcome (O), da variável GlycemiaValues (G)?
```

A opção 5 – “Cálculos das medidas amostrais das variáveis” – permite ao utilizador executar e, caso assim o pretenda, guardar, cálculos estatísticos das medidas amostrais das variáveis, nomeadamente média, média ponderada, mediana, variância e desvio padrão.

```
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 BloodPressure
69.11
Deseja guardar o cálculo num ficheiro? Sim ou Não?
sim
Escolha o nome do seu ficheiro: media_bmi
```

A opção 6 - Relatório geral da base de dados – descarrega um atalho HTML que redireciona o utilizador para uma página web que apresenta um relatório interativo sobre a base de dados, com várias funcionalidades pelas quais o utilizador pode ter uma visão geral da base de dados e ver medidas amostrais para cada variável, interações entre as variáveis, vários métodos estatísticos para determinar correlações entre as variáveis, a contagem de valores em falta e as partes iniciais e finais da base de dados utilizada.

A opção 7 – Sair – termina a execução do programa.



É ainda de realçar que no final de qualquer função executada que corresponde a uma funcionalidade descrita acima, executamos a função `terminar()`, que tem o objetivo de perguntar ao utilizador se deseja terminar o programa ou se pretende voltar ao menu principal.

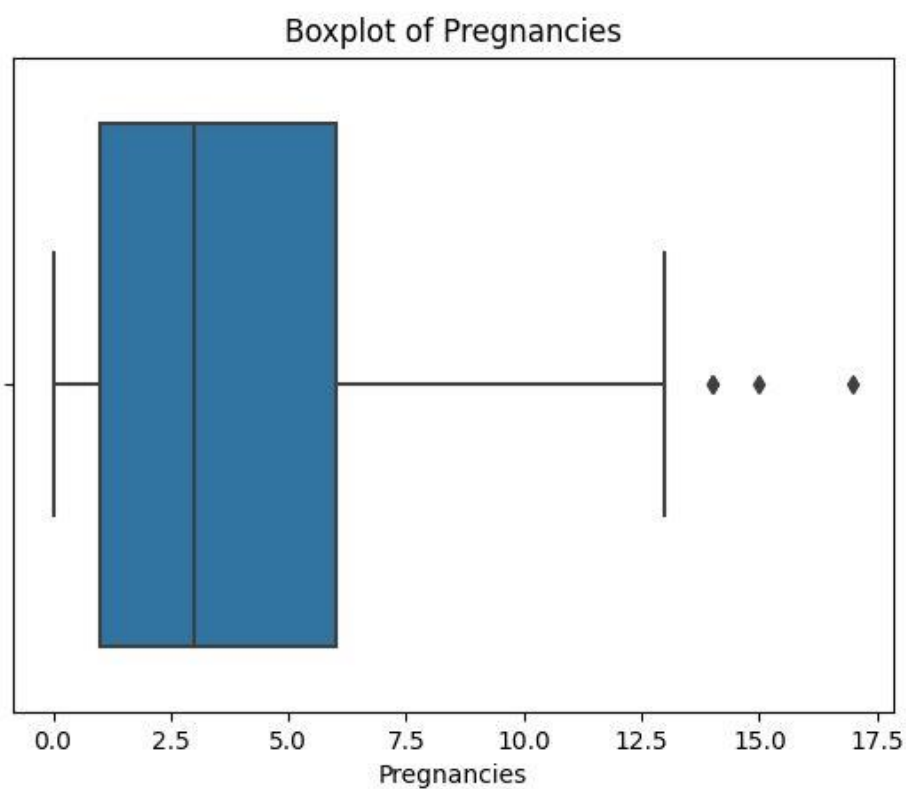
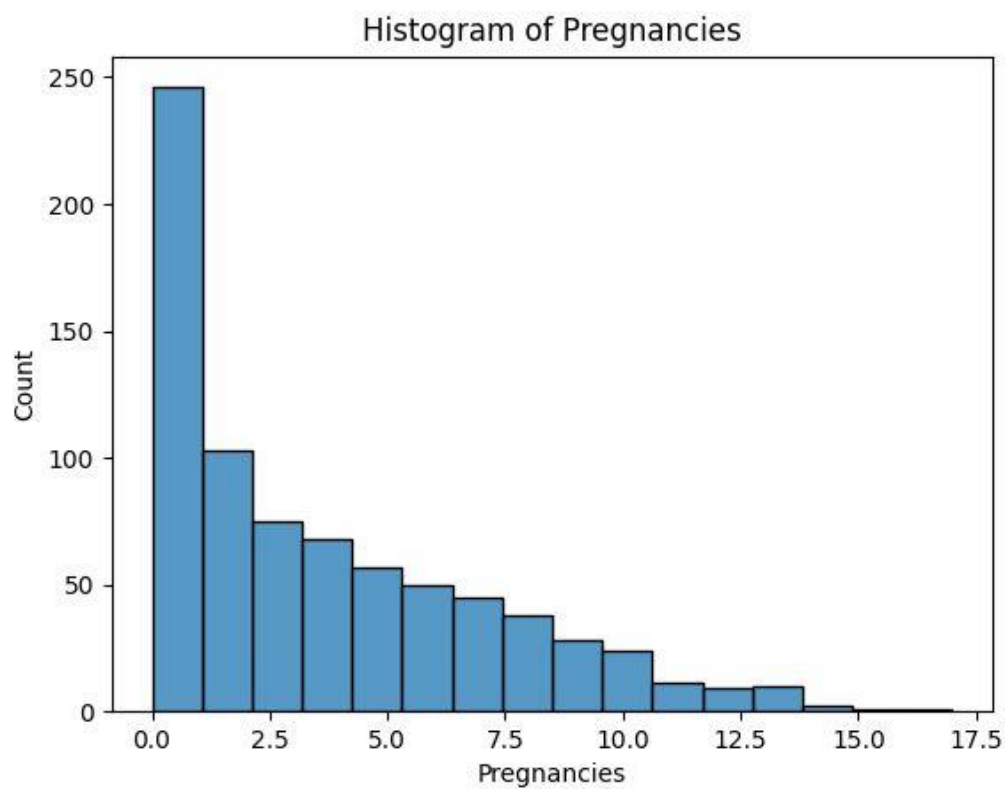
Para além disso, procuramos ainda fazer todas as verificações possíveis através de ciclos *while* de forma a validar os inputs fornecidos pelo utilizador.

Finalmente, realçamos ainda que quando qualquer uma das funções do ficheiro `Plots.py` é chamada através do menu, é automaticamente guardado num ficheiro png. Na opção 5 do menu, correspondente à opção cálculos das medidas amostrais das variáveis, permitimos ao utilizador também guardar num ficheiro txt, caso assim o desejar.

### Exemplos:

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



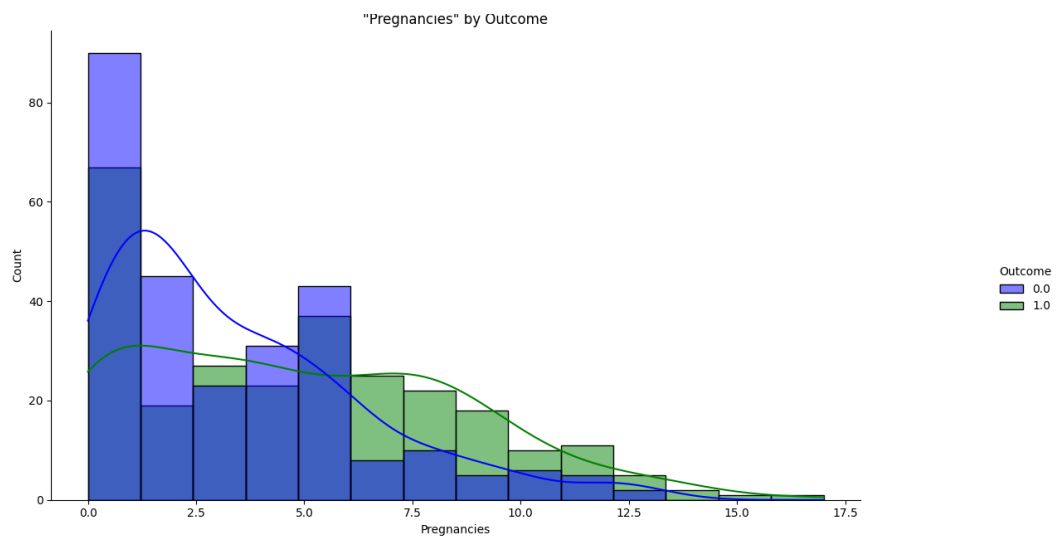
```

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,kde = True)
        plt.title(f"Histogram of {i}")
        plt.show()

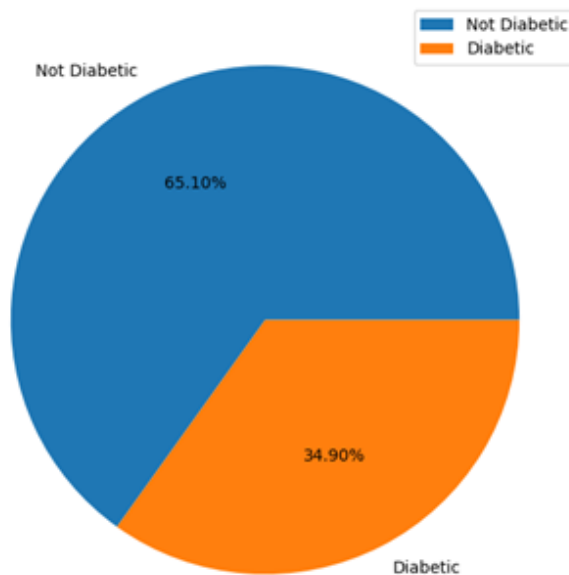
        sns.boxplot(data = dataframe, x = i)
        plt.title(f"Boxplot of {i}")
        plt.show()
  
```

Através desta função é possível escolher uma ou mais variáveis e executar 3 tipos de análise estatística para a(s) variável(eis) selecionada(s), uma tabela, um histograma e um boxplot.



```
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'"{var}" by {vcategorical}')
    plt.plot()
    plt.show()
```

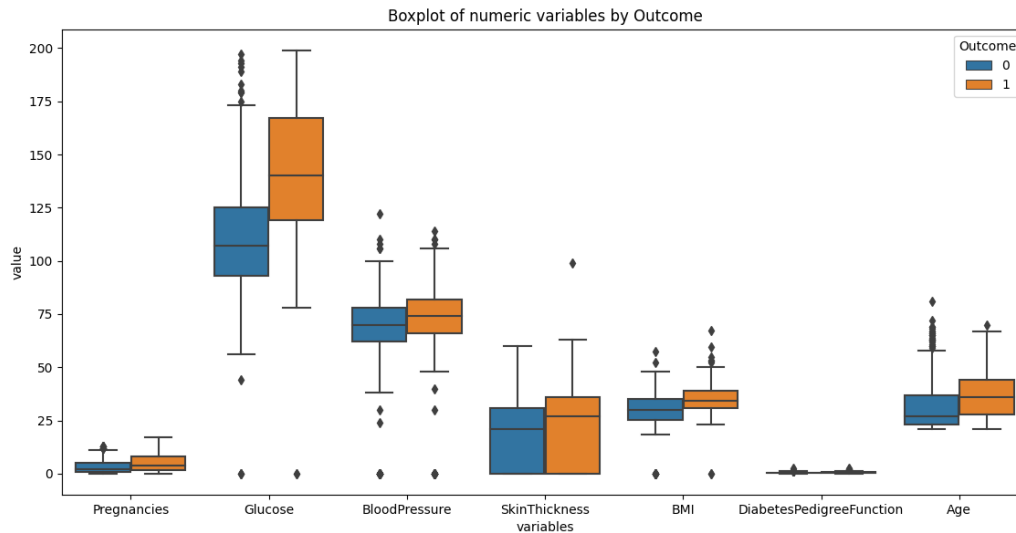
Esta função permite a escolha de uma ou mais variáveis e constrói um histograma com função de densidade para a(s) variável(eis) selecionada(s), com a opção de apresentar o gráfico em função das variáveis categóricas ou de nenhuma através do parâmetro `vcategorical`.



```
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_counts(), labels = labels,
autopct = '%0.02f%%')
    plt.legend()
    plt.show()
```

Esta função cria um gráfico circular para uma das variáveis categóricas “*Outcome*” ou “*Glycemiavalues*” selecionada pelo utilizador, de tamanho 10x7 e com as percentagens de cada um dos valores da variável inbutidas dentro do gráfico circular arredondado a 2 casas decimais.

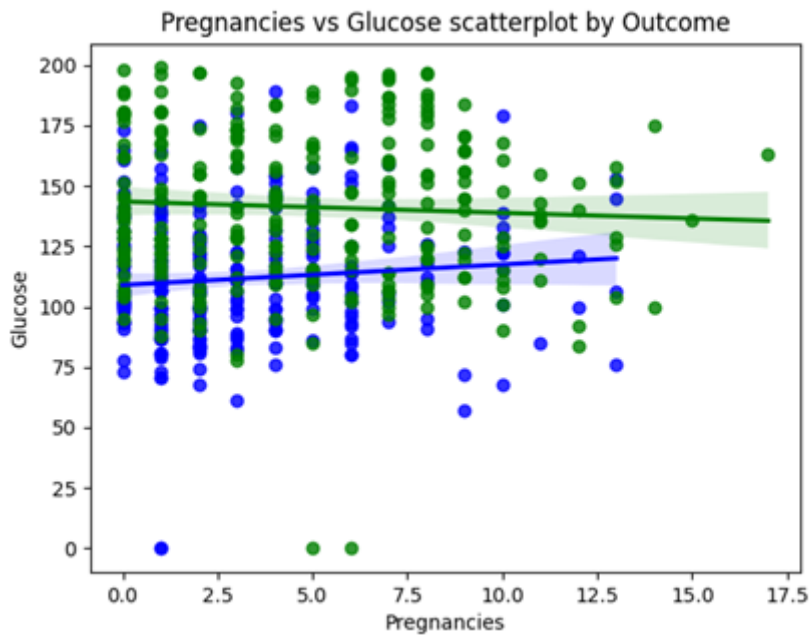


```
def boxplot_all(dataframe, drop_values, has_outcome = False):
    diabetesbp = dataframe.drop(drop_values, axis = 1)
    diabetes_melted = pd.melt(diabetesbp, id_vars = "Outcome", var_name =
"variables", value_name = "value")

    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.show()
```

Esta função é chamada na opção 3 do menu e exibe *boxplots* para todas as variáveis numéricas simultaneamente, permitindo ao utilizador escolher ocultar as variáveis que desejar e, adicionalmente, distinguir dentro das variáveis numéricas a distribuição das observações em função do “*Outcome*” das mesmas.



```

def regressao (dataframe, variavel_1, variavel_2, vcategorical = None):
    if vcategorical == "Outcome":
        sns.regplot(x = variavel_1, y = variavel_2, data =
df_bal[df_bal['Outcome'] == 0], color = 'blue')
        sns.regplot(x = variavel_1, y = variavel_2, data =
df_bal[df_bal['Outcome'] == 1], color = 'green')

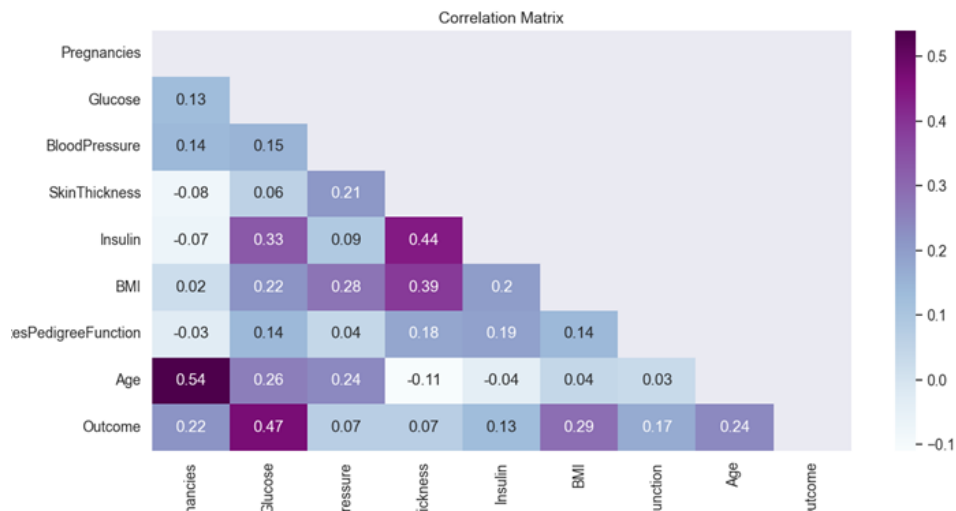
    elif vcategorical=="Glycemiavalues":
        sns.regplot(x = variavel_1, y = variavel_2, data =
df_bal[df_bal['Glycemiavalues'] == "Hypoglycemia"], color = 'yellow')
        sns.regplot(x = variavel_1, y = variavel_2, data =
df_bal[df_bal['Glycemiavalues'] == "Normal"], color = 'green')
        sns.regplot(x = variavel_1, y = variavel_2, data =
df_bal[df_bal['Glycemiavalues'] == "Pre-diabetes"], color = 'blue')

    else:
        sns.regplot(x = variavel_1, y = variavel_2, data = dataframe,
color = 'blue')

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

Esta função cria um gráfico de dispersão com a reta de regressão linear da distribuição de duas variáveis numéricas, representadas nos eixos x e y, permitindo ao utilizador adicionar, opcionalmente, a distinção da distribuição das observações por uma das variáveis categóricas.





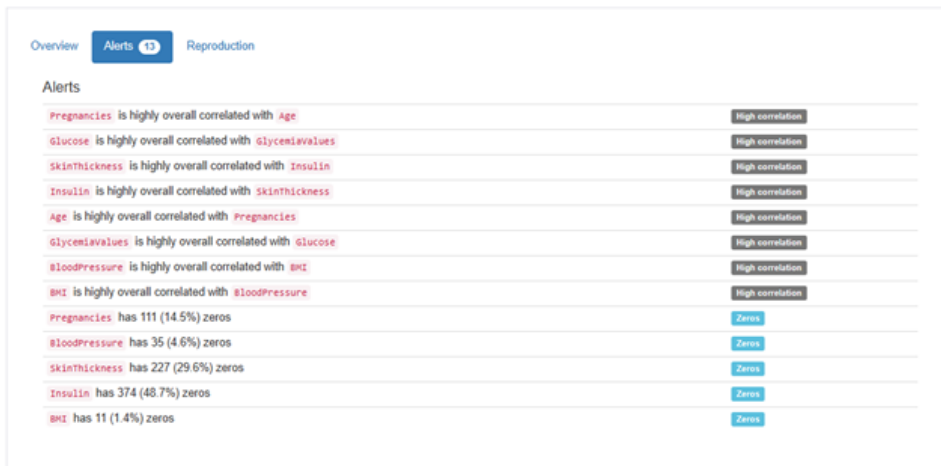
```
def matrcorr(dataframe):
    corr = dataframe.corr().round(2)
    plt.figure(figsize = (14, 10))
    sns.set(font_scale = 1.15)
    mask = np.zeros_like(corr)
    mask[np.triu_indices_from(mask)] = True
    sns.heatmap(corr, annot = True, cmap = 'BuPu', mask = mask, cbar =
True)
    plt.title('Correlation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

Esta função cria uma matriz de correlação que indica, arredondado a duas casas decimais, a força da correlação entre as 9 variáveis iniciais do *dataset*, apresentando um gradiente de cores (*heatmap*) que acompanha estas diferenças de correlação de forma a ajudar visualmente a interpretação da informação gráfica por parte do utilizador.

```
<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
```

```
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()
```

Esta função redireciona o utilizador para um submenu onde este pode escolher o que deseja de entre uma lista de cálculos de medidas amostrais das variáveis, tendo o utilizador a potencialidade de selecionar quais e quantas variáveis pretende para efetuar os cálculos e guardar os resultados obtidos num ficheiro .txt



The screenshot shows a web interface with tabs for 'Overview', 'Alerts' (selected), and 'Reproduction'. Under the 'Alerts' tab, there is a table with two columns: a description of the alert and a button indicating the correlation level.

Alert	Correlation
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

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

Esta função, que recorre ao *package* “*pandas profiling*”, descarrega um atalho HTML que redireciona o utilizador para uma página web que apresenta um relatório interativo sobre a base de dados.

## 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* (<https://github.com/Joanaa27/ProjetoPAC>);
- 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.

## Conclusão

Este projeto conseguiu alcançar, os objetivos propostos, conseguindo atingir o principal foco de desenvolver uma interface estilo menu com a qual é possível ao utilizador interagir de forma livre e personalizada com a base de dados. Para além disso foi realizada com sucesso a leitura e manipulação da base de dados, inclusive criando uma nova variável. Ademais foram desenvolvidas diversas funções para a construção e visualização gráfica e estatística da base de dados, englobadas na interface por nós desenvolvida. Por fim, foi também possível extrair a informação visualizada e obtida pelo utilizador para ficheiros do tipo .png e .txt.

## Referências

1. Akturk, M. (2020, August 5). Diabetes *dataset*. Kaggle. Retrieved December 3, 2022, from <https://www.kaggle.com/datasets/mathchi/diabetes-data-set>
2. 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>
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>