

**PROJETO:** PREVENDO DIABETES EM MULHERES

**DISCENTE:** GUILHERME ALVES ()

**DISCENTE:** VINICIUS DE SOUZA SANTOS(BI3008061)

**DOCENTE:** DR. MURILO VARGES DA SILVA

**DISCIPLINA:** MDAEL8 – MINERAÇÃO DE DADOS

# LISTA DE FIGURAS

[Figura 1: Base de dados de prevenção de diabete em mulheres 4](#_Toc130424055)

[Figura 2: Saída do DataCleaning do projeto. 7](#_Toc130424056)

[Figura 3: Saida do algoritmo de Normalização 9](#_Toc130424057)

[Figura 4: Gráficos de Dispersão entre o Resultado e a Espessura da Pele 15](#_Toc130424058)

[Figura 5: Gráficos de setores entre o Resultado e o número de gestações 16](#_Toc130424059)

[Figura 6: Gráficos de Histograma entre o resultado e o IMC 17](#_Toc130424060)

Sumário

[LISTA DE FIGURAS 2](#_Toc130424020)

[1 Objetivo 2](#_Toc130424021)

[1.1 Sobre a base de dados 3](#_Toc130424022)

[2 Pré-processamento 4](#_Toc130424023)

[3 Normalização e Redução de Dados 8](#_Toc130424024)

[3.1 Normalização dos dados: 8](#_Toc130424025)

[3.2 Redução dos dados: 10](#_Toc130424026)

[4 Análise descritiva de dados – Visualização 12](#_Toc130424027)

[4.1 Gráficos em Histograma, Dispersão e Setores 13](#_Toc130424028)

[5 Análise descritiva de dados – Medidas 18](#_Toc130424029)

[Referências Bibliográficas 19](#_Toc130424030)

# 1 Objetivo

Estimular o aluno a aplicar os conhecimentos apresentados no decorrer da disciplina em problemas reais de mineração de dados, utilizando as técnicas de seleção, pré-processamento e transformação de dados, técnicas de visualização de dados, análise descritiva, análise de grupos, classificação e estimação/regressão.

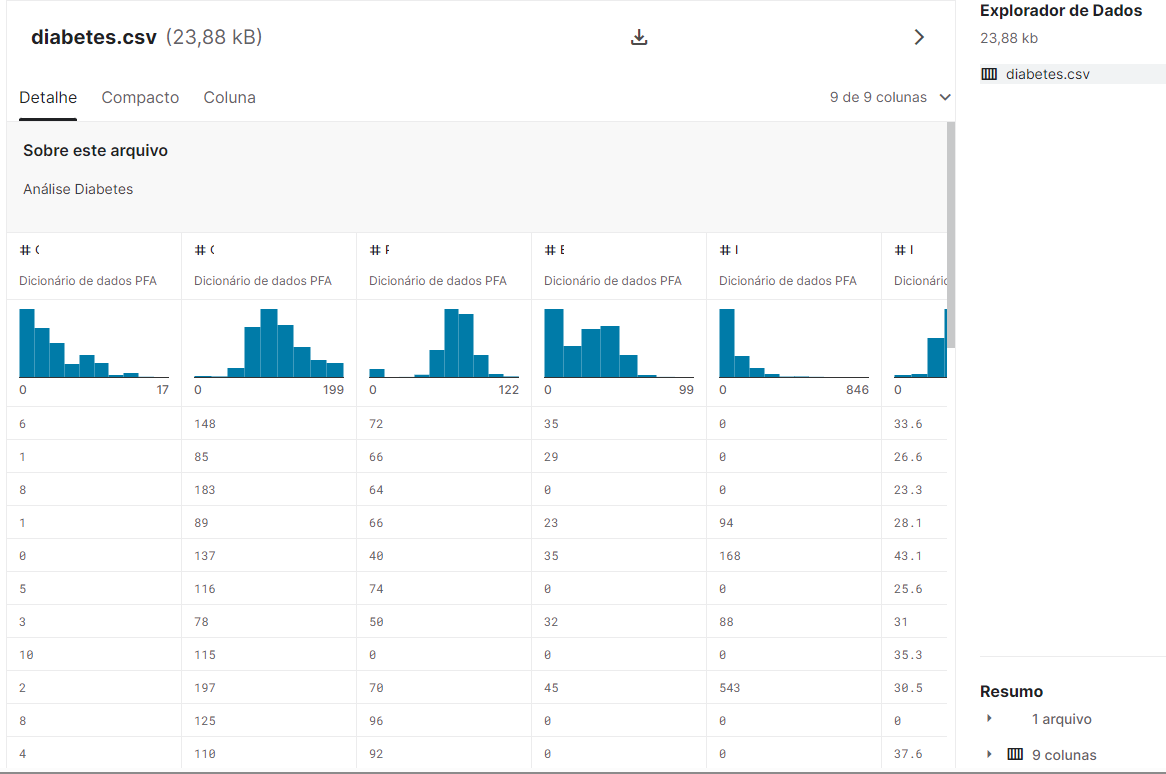
## 1.1 Sobre a base de dados

Este conjunto de dados é originalmente do Instituto Nacional de Diabetes e Doenças Digestivas e Renais. O objetivo do conjunto de dados é prever de forma diagnóstica se um paciente tem diabetes, com base em certas medidas de diagnóstico incluídas no conjunto de dados. Várias restrições foram colocadas na seleção dessas instâncias de um banco de dados maior. Em particular, todos os pacientes aqui são mulheres com pelo menos 21 anos de idade, descendentes dos índios Pima.2 A partir dos dados do arquivo (.csv) Podemos encontrar várias variáveis, algumas delas são independentes (várias variáveis reditivas médicas) e apenas uma variável dependente de destino (resultado).

# 2 Pré-processamento

Nessa primeira etapa do projeto foi feito o pré-processamento dos dados da base de prevenção a diabetes, essa base de dados foi disponibilizada pelo site da Kaggle nela foi abstraída informações relevantes como: Número de gestações, insulina, IMC, espessura da pele entre outros, com essa base de dados vamos fazer a preparação desses dados, conforme observado na figura 1 apresentando a base de dados analisada.

Figura 1: Base de dados de prevenção de diabete em mulheres



fonte: CHAUHAN, 2022

Na aula foi apresentado através de um repositório do professor vários algoritmos de mineração de dados e para esse capítulo usamos apenas o do diretório de processamento.

Para fazer o pré-processamento usamos o algoritmo DataCleaning.py.

import pandas as pd

import numpy as np

def main():

    # Faz a leitura do arquivo

    names = ['Número Gestações','Glucose','pressao Arterial','Expessura da Pele','Insulina','IMC','Função Pedigree Diabete','Idade','Resultado']

    features = ['Número Gestações','Glucose','pressao Arterial','Expessura da Pele','Insulina','IMC','Função Pedigree Diabete','Idade','Resultado']

    output\_file = '0-Datasets/diabetesClear.data'

    input\_file = '0-Datasets/diabetes.data'

    df = pd.read\_csv(input\_file,         # Nome do arquivo com dados

                     names = names,      # Nome das colunas

                     usecols = features, # Define as colunas que serão  utilizadas

                     na\_values='?')      # Define que ? será considerado valores ausentes

    df\_original = df.copy()

    # Imprime as 20 primeiras linhas do arquivo

    print("PRIMEIRAS 20 LINHAS\n")

    print(df.head(20))

    print("\n")

    # Imprime informações sobre dos dados

    print("INFORMAÇÕES GERAIS DOS DADOS\n")

    print(df.info())

    print("\n")

    # Imprime uma analise descritiva sobre dos dados

    print("DESCRIÇÃO DOS DADOS\n")

    print(df.describe())

    print("\n")

    # Imprime a quantidade de valores faltantes por coluna

    print("VALORES FALTANTES\n")

    print(df.isnull().sum())

    print("\n")

    columns\_missing\_value = df.columns[df.isnull().any()]

    print(columns\_missing\_value)

    method = 'mean' # number or median or mean or mode

    for c in columns\_missing\_value:

        UpdateMissingValues(df, c, method)

    print(df.describe())

    print("\n")

    print(df.head(15))

    print(df\_original.head(15))

    print("\n")

    # Salva arquivo com o tratamento para dados faltantes

    df.to\_csv(output\_file, header=False, index=False)

def UpdateMissingValues(df, column, method="mean", number=0):

    if method == 'number':

        # Substituindo valores ausentes por um número

        df[column].fillna(number, inplace=True)

    elif method == 'median':

        # Substituindo valores ausentes pela mediana

        median = df['Density'].median()

        df[column].fillna(median, inplace=True)

    elif method == 'mean':

        # Substituindo valores ausentes pela média

        mean = round(df[column].mean(), 2)  # Modificação na linha da média

        df[column].fillna(mean, inplace=True)

    elif method == 'mode':

        # Substituindo valores ausentes pela moda

        mode = df[column].mode()[0]

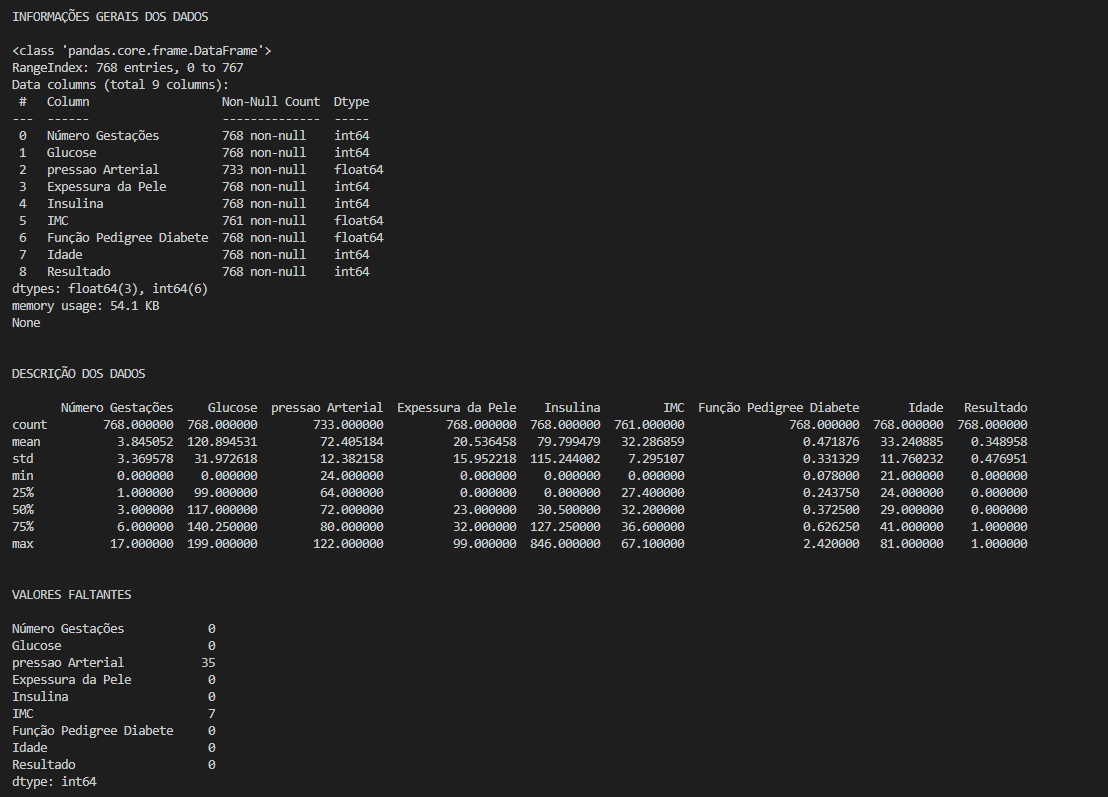
        df[column].fillna(mode, inplace=True)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Ao adaptar esse algoritmo a nossa base de dados obtivemos as seguintes saídas.

Figura 2: Saída do DataCleaning do projeto.



fonte: vscode, 2023

Foi observado duas colunas com dados faltantes e nela foi substituído em uma nova base pela mediana dos valores, foi escolhido a mediada devido os dados da coluna ser assimétrico, ou seja, em um histograma os dados da direita ou esquerda são desiguais conforme verificamos abaixo:

Figura 3: Histograma dos dados de Pressão Arterial

Gráfico, Histograma

Descrição gerada automaticamente

fonte: vscode, 2023

Figura 4: Histograma dos Dados de IMC

Gráfico, Histograma

Descrição gerada automaticamente

fonte: vscode, 2023

# 3 Normalização e Redução de Dados

## 3.1 Normalização dos dados:

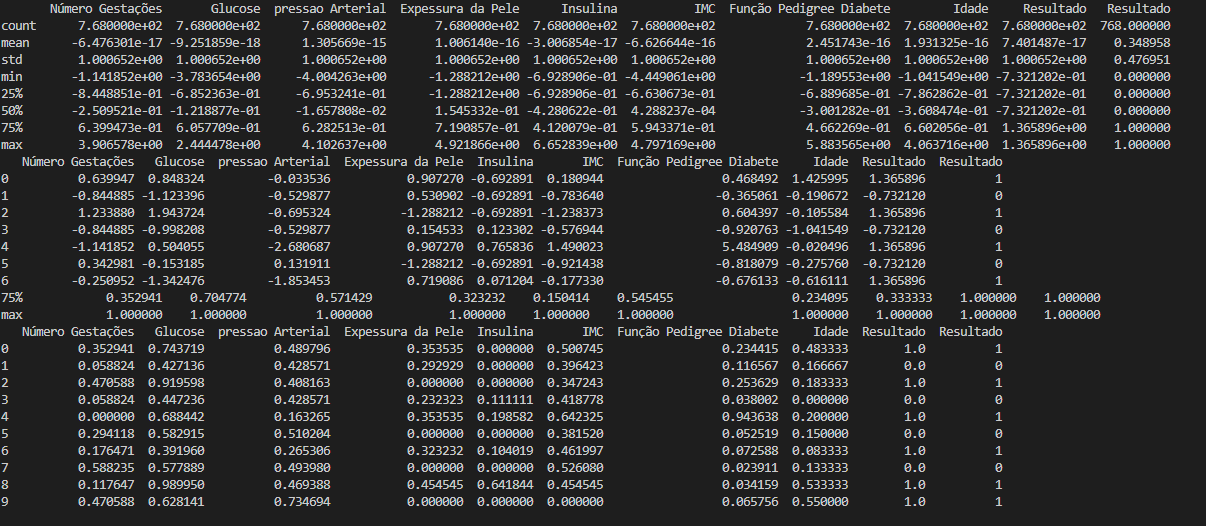
A normalização de dados é o processo de escalar os dados para que todos estejam na mesma escala. Isso é importante porque algumas variáveis podem ter uma escala muito maior do que outras e podem ter um impacto desproporcional na análise. Existem diferentes métodos de normalização, mas um dos mais comuns é a normalização Min-Max, que ajusta os valores para um intervalo entre 0 e 1. Para aplicar a normalização Min-Max à base de dados de prevê diabetes, seguimos esses passos:

* Importar a biblioteca Scikit-learn para a normalização dos dados.
* Dividir a base de dados em duas partes, uma para as variáveis dependentes e outra para a variável de saída.
* Aplicar a normalização Min-Max às variáveis dependentes usando a função MinMaxScaler().
* Concatenar as variáveis dependentes normalizadas com a variável de saída para formar a base de dados normalizada.
* import pandas as pd
* from sklearn.preprocessing import MinMaxScaler
* # Faz a leitura do arquivo
* input\_file = '0-Datasets/diabetesClear.data'
* names = ['Número Gestações','Glucose','pressao Arterial','Expessura da Pele','Insulina','IMC','Função Pedigree Diabete','Idade','Resultado']
* features = ['Número Gestações','Glucose','pressao Arterial','Expessura da Pele','Insulina','IMC','Função Pedigree Diabete','Idade','Resultado']
* target = 'Resultado'
* df = pd.read\_csv(input\_file,    # Nome do arquivo com dados
* names = names) # Nome das colunas
* # Separar as variáveis dependentes da variável de saída
* X = df.drop('Resultado', axis=1)
* y = df['Resultado']
* # Aplicar a normalização Min-Max às variáveis dependentes
* scaler = MinMaxScaler()
* X\_norm = scaler.fit\_transform(X)
* # Criar um novo dataframe com as variáveis dependentes normalizadas e a variável de saída
* df\_norm = pd.DataFrame(X\_norm, columns=X.columns)
* df\_norm['Resultado'] = y
* # Salvar a base de dados normalizada em um novo arquivo csv
* df\_norm.to\_csv('dados\_diabetes\_normalizados.csv', index=False)

O arquivo csv com a base de dados original é lido usando a biblioteca pandas, e as variáveis dependentes (todas as colunas exceto a última) são separadas da variável de saída (a última coluna). Em seguida, a normalização Min-Max é aplicada às variáveis dependentes usando a classe MinMaxScaler da biblioteca Scikit-learn.

O resultado da normalização é armazenado em um novo dataframe chamado df\_norm, que contém as variáveis dependentes normalizadas e a variável de saída original. Finalmente, o dataframe normalizado é salvo em um novo arquivo csv usando a função to\_csv() da biblioteca pandas.

Figura 5: Saida do algoritmo de Normalização



fonte: vscode, 2023

## 3.2 Redução dos dados:

A redução de dados é o processo de reduzir a dimensionalidade dos dados para eliminar variáveis desnecessárias ou redundantes. Isso pode ajudar a reduzir o tempo de processamento e melhorar a precisão do modelo. Uma técnica comum de redução de dados é a Análise de Componentes Principais (PCA), que reduz as variáveis para um número menor de componentes que capturam a maior parte da variação dos dados. Para aplicar a PCA à base de dados fornecida, você pode seguir os seguintes passos:

* Importar a biblioteca Scikit-learn para a redução dos dados.
* Dividir a base de dados em duas partes, uma para as variáveis dependentes e outra para a variável de saída.
* Aplicar a PCA às variáveis dependentes usando a função PCA().
* Selecionar o número de componentes que capturam a maior parte da variação dos dados.
* Transformar as variáveis dependentes reduzidas em uma matriz e concatenar com a variável de saída para formar a base de dados reduzida.
* import pandas as pd
* from sklearn.decomposition import PCA
* # Ler o arquivo csv com a base de dados normalizada
* df\_norm = pd.read\_csv('dados\_diabetes\_normalizados.csv')
* # Separar as variáveis dependentes da variável de saída
* X\_norm = df\_norm.drop('Resultado', axis=1)
* y\_norm = df\_norm['Resultado']
* # Aplicar a PCA às variáveis dependentes normalizadas
* pca = PCA(n\_components=3)
* X\_pca = pca.fit\_transform(X\_norm)
* # Criar um novo dataframe com as variáveis dependentes reduzidas e a variável de saída
* df\_pca = pd.DataFrame(X\_pca, columns=['PCA1', 'PCA2', 'PCA3'])
* df\_pca['Resultado'] = y\_norm
* # Salvar a base de dados reduzida em um novo arquivo csv
* df\_pca.to\_csv('dados\_diabetes\_reduzidos.csv', index=False)

O arquivo csv com a base de dados normalizada é lido usando a biblioteca pandas, e as variáveis dependentes (todas as colunas exceto a última) são separadas da variável de saída (a última coluna).

Em seguida, a PCA é aplicada às variáveis dependentes normalizadas usando a classe PCA da biblioteca Scikit-learn, especificando o número de componentes que se deseja manter. Neste exemplo, estamos mantendo os três primeiros componentes principais.

O resultado da redução é armazenado em um novo dataframe chamado df\_pca, que contém as variáveis dependentes reduzidas e a variável de saída original. Finalmente, o dataframe reduzido é salvo em um novo arquivo csv usando a função to\_csv() da biblioteca pandas.

Em resumo, a normalização e redução de dados são técnicas importantes para pré-processar a base de dados utilizada para diabetes antes de aplicar algoritmos para prever diabetes. A normalização ajuda a garantir que todas as variáveis estejam na mesma escala, enquanto a redução ajuda a reduzir a dimensionalidade dos dados para melhorar a precisão do modelo.

# 4 Análise descritiva de dados – Visualização

A análise descritiva de dados é uma técnica muito utilizada em mineração de dados, que tem como objetivo analisar e entender os dados de um conjunto de dados para obter informações relevantes. Essa técnica permite entender melhor as características dos dados, identificar padrões, tendências e anomalias, além de verificar a qualidade dos dados.

A visualização de dados é uma ferramenta essencial para a análise descritiva de dados, pois permite a representação gráfica dos dados de uma forma que facilita a compreensão e a interpretação das informações contidas no conjunto de dados. Com a visualização, é possível identificar padrões e tendências que não seriam detectados com apenas a análise dos dados em formato tabular.

Entre as principais técnicas de visualização utilizadas na análise descritiva de dados estão o histograma, o gráfico de setores, o gráfico de barras, o gráfico de dispersão e a matriz de correlação. Cada uma dessas técnicas é utilizada para representar diferentes tipos de dados e para identificar diferentes tipos de padrões e tendências.

Em resumo, a análise descritiva de dados e a visualização de dados são técnicas fundamentais em mineração de dados para compreender e extrair informações úteis de grandes conjuntos de dados.

## 4.1 Gráficos em Histograma, Dispersão e Setores

Os gráficos de histograma, dispersão e setores são comumente utilizados na análise descritiva de dados por suas capacidades de visualização eficaz e fácil interpretação dos dados.

O gráfico de histograma é útil para visualizar a distribuição dos dados e pode ajudar a identificar padrões, como picos e caudas longas, bem como a presença de valores atípicos. É particularmente útil quando se trabalha com dados contínuos ou discretos.

O gráfico de dispersão é utilizado para analisar a relação entre duas variáveis. É capaz de identificar padrões, correlações e relacionamentos entre variáveis e pode ser usado para identificar possíveis causas e efeitos.

Já o gráfico de setores é comumente usado para mostrar a proporção de cada categoria em um conjunto de dados. É útil quando se trabalha com dados categóricos e pode ajudar a identificar padrões ou desequilíbrios entre as categorias.

Esses gráficos são considerados eficazes porque são fáceis de entender e interpretar. Eles permitem que os usuários vejam rapidamente as tendências, padrões e correlações nos dados, sem precisar olhar para tabelas de números ou dados brutos. Além disso, esses gráficos podem ser usados para identificar problemas nos dados, como valores discrepantes ou ausentes, o que pode levar a análises mais precisas e confiáveis.

Em resumo, esses gráficos são valiosos para a análise descritiva de dados porque oferecem uma maneira visualmente atraente e intuitiva de explorar os dados e comunicar os resultados de maneira clara e concisa.

Abaixo segue o algoritmo em Python utilizado para a plotagem desses gráficos conforme a base de dados de Prever Diabetes.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv('0-Datasets/diabetesClear.data')

for coluna in df.columns:

    # Histograma

    plt.figure()

    sns.histplot(data=df, x=coluna)

    plt.title(f'Histograma da coluna {coluna}')

    plt.savefig(f'3-Analise Descritiva - Visualização/Graficos/Histograma/historgrama\_{coluna}.png')

    plt.close()

    # Gráfico de setores

    plt.figure()

    df[coluna].value\_counts().plot(kind='pie')

    plt.title(f'Gráfico de setores da coluna {coluna}')

    plt.savefig(f'3-Analise Descritiva - Visualização/Graficos/Grafico de Setores/grafico\_setores\_{coluna}.png')

    plt.close()

    # Dispersão

    plt.figure()

    sns.scatterplot(data=df, x=coluna, y='Outcome')

    plt.title(f'Dispersão da coluna {coluna} em relação à coluna alvo')

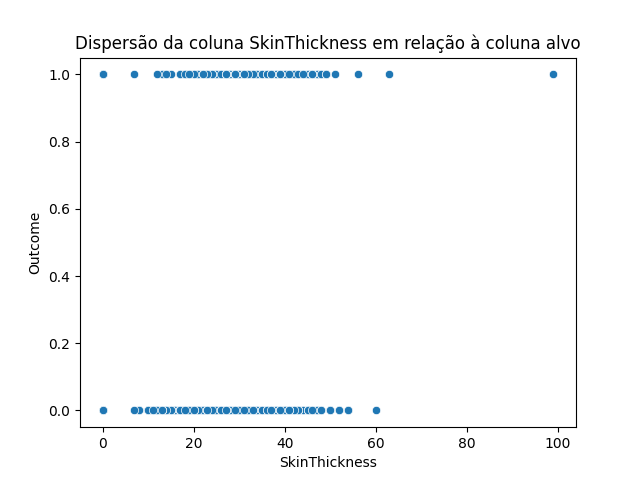
    plt.savefig(f'3-Analise Descritiva - Visualização/Graficos/Dispersao/dispersao\_{coluna}.png')

    plt.close()

Nesse código, assumimos que há uma coluna, "'Outcome'" que será usada como referência para o gráfico de dispersão. Caso não haja, basta remover essa linha e adaptar o código para a sua necessidade.

Abaixo temos alguns exemplos de saída desses gráficos.

Figura 6: Gráficos de Dispersão entre o Resultado e a Espessura da Pele



fonte: elaborado pelo autor (2023)

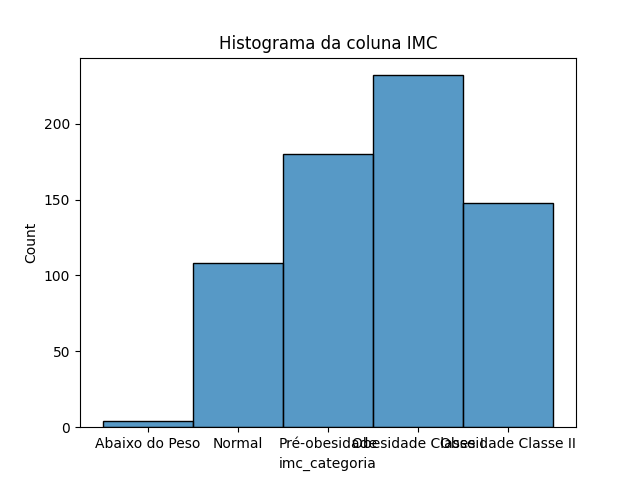
Figura 7: Gráficos de setores entre o Resultado e o IMC categorizado

Gráfico, Gráfico de pizza

Descrição gerada automaticamente

fonte: elaborado pelo autor (2023)

Figura 8: Gráficos de Histograma entre o resultado e o IMC Categorizado



fonte: elaborado pelo autor (2023)

# 5 Análise descritiva de dados – Medidas

Análise descritiva de dados é uma técnica de mineração de dados que busca descrever as características de um conjunto de dados, sem inferir relações causais ou preditivas. Ela é utilizada para explorar os dados, obter insights e compreender melhor as informações disponíveis.

As medidas estatísticas são ferramentas fundamentais na análise descritiva de dados. As medidas de tendência central, dispersão, posição relativa e associação são exemplos de medidas estatísticas utilizadas na análise descritiva de dados.

Medidas de tendência central: São medidas que indicam o ponto central de um conjunto de dados. As três medidas de tendência central mais comuns são a média, a mediana e a moda.

Medidas de dispersão: São medidas que indicam o grau de variação dos dados em relação a uma medida central. As medidas de dispersão mais comuns são o desvio padrão, a variância e o coeficiente de variação.

Medidas de posição relativa: São medidas que indicam a posição de um valor em relação aos demais valores de um conjunto de dados. As medidas de posição relativa mais comuns são o percentil e o quartil.

Medidas de associação: São medidas que indicam a relação entre duas ou mais variáveis. As medidas de associação mais comuns são a correlação e a covariância.

Em resumo, a análise descritiva de dados utiliza medidas estatísticas para resumir e descrever os dados, de forma a obter insights e compreender melhor as informações disponíveis. As medidas de tendência central, dispersão, posição relativa e associação são algumas das medidas estatísticas mais comuns utilizadas na análise descritiva de dados.

# Referências Bibliográficas

CHAUHAN, Aman. Predict Diabetes. Kaggle.com. Disponível em: <https://www.kaggle.com/datasets/whenamancodes/predict-diabities>. Acesso em: 22 fev. 2023.