

Análise de Dados em Python

Relatório dos Dados da Secretaria de Saúde de Sucuri do Sul

Relatório do Projeto Guiado em Análise de
Dados, trabalho elaborado pelo professor
Thiago Vaz como pré-requisito do 1º ofício de
relatório utilizando a linguagem de
programação Python

Jundiaí

2023

Guilherme Vinícius Bispo da Silva

Índice:

Sumário Página 2

Objetivo Página 3

Resumo Página 3

Introdução Página 3

Metodologia Página 4

Conclusão Página 24

Referências Página 25

1. Objetivo:

O presente relatório tem por objetivo analisar um banco de dados com informações dos usuários do sistema municipal de saúde de Sucuri do Sul para, a partir dos dados, relatar tudo o que tange às informações pertinentes ao quadro de dados desse município, como a distribuição de sexo, a média das idades, a altura dos usuários, o tipo sanguíneo, tal qual, o peso e o IMC (Índice de Massa Corporal) dos registrados.

2. Resumo:

Este relatório é o resultado de pesquisas e análises, relacionadas com o tema Análise de Dados: um trabalho dedicado ao aprofundamento sobre os dados coletados, sendo o referido, dos usuários do sistema de saúde de Sucuri do Sul. Para este efeito, fez-se um estudo acerca da distribuição de sexo, do tipo sanguíneo e da implementação do IMC (Índice de Massa Corporal), que leva em consideração o peso e a altura, a fim de compreender a classificação dos usuários. Sob esse viés, concentra-se algumas considerações que serão discutidas ao longo da Metodologia e compactadas na conclusão do Projeto Guiado.

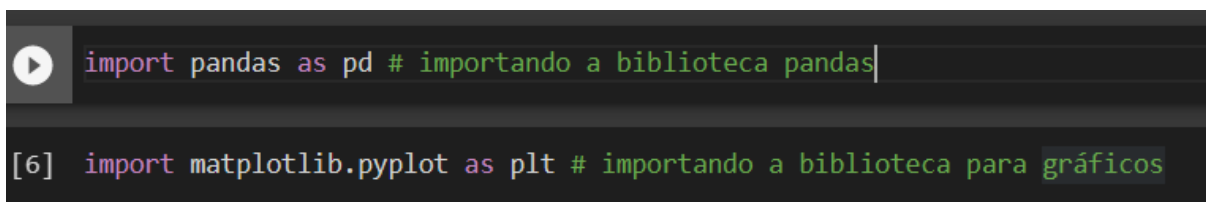
3. Introdução:

A Análise de Dados na linguagem de programação Python, a respeito dos dados da secretaria de saúde do município de Sucuri do Sul, realizada no dia 10/01/2023 até 20/01/2023, guiado pelo professor Thiago Vaz, é de suma importância nos conhecimentos do ramo de programação, uma vez que fornece base para outros projetos dessa vertente. Assim, com a coleta desses dados, compreendia-se necessário abstrair informações relativas às variáveis desse quadro de dados: sexo, idade, altura, peso e tipo de sangue. Deste modo, analisando histogramas e levantando referências para a pesquisa. Neste sentido, a implementação do parâmetro IMC (Índice de Massa Corporal) ajudou a constar as variáveis peso e altura na análise, sobressaindo as classificações dos usuários, segundo esse índice. Por fim, cabia-se uma abordagem da disposição dos registrados por sexo e idade, haja vista, em torno da representação das informações dos limites, mais novo, mais velho e, conseqüentemente, a porcentagem entre homens e mulheres.

4. Metodologia:

Tendo como princípio norteador a linguagem de programação Python, no que diz respeito a manipulação e modulação dos códigos; a biblioteca “Pandas” para a importação dos dados e a “Matplotlib” para a visualização gráfica dos dados acessados; o arquivo de dados da Secretaria de Saúde de Sucuri do Sul, a fim de constar a base de dados a ser analisada; a plataforma Google Collaboratory, com intenção de executar o código descrito; e o Google Drive, com o intuito de compactar o arquivo de dados, foi possibilitado a Análise de Dados desse município de Sucuri do Sul.

4.1. Importando as Bibliotecas:



```
import pandas as pd # importando a biblioteca pandas  
  
[6] import matplotlib.pyplot as plt # importando a biblioteca para gráficos
```

1ª Biblioteca Importada - Pandas

A biblioteca Pandas possibilita importar dados armazenados em diversos formatos, neste caso csv, tratando-os e transformando-os em um Data Frame (Quadro de Dados) para realizar as análises necessárias.

2ª Biblioteca Importada - Matplotlib

A biblioteca Matplotlib possibilita a visualização de dados e plotagem gráfica, ou seja, é através dela que acontecerá a plotagem do histograma para análise.

Comandos utilizados:

import: esse comando em Python é usado para obter acesso a outros módulos, um conjunto de funções e variáveis. Assim, quando precisamos incluir ou usar essas funções de módulos em nosso código, usamos o comando import e podemos agregar facilmente as funções e variáveis do módulo.

as: esse comando em Python substitui o nome da biblioteca importada pelo *import* para usá-las pelo apelido referido, como por exemplo *pandas* ser chamado pelo apelido *pd*.

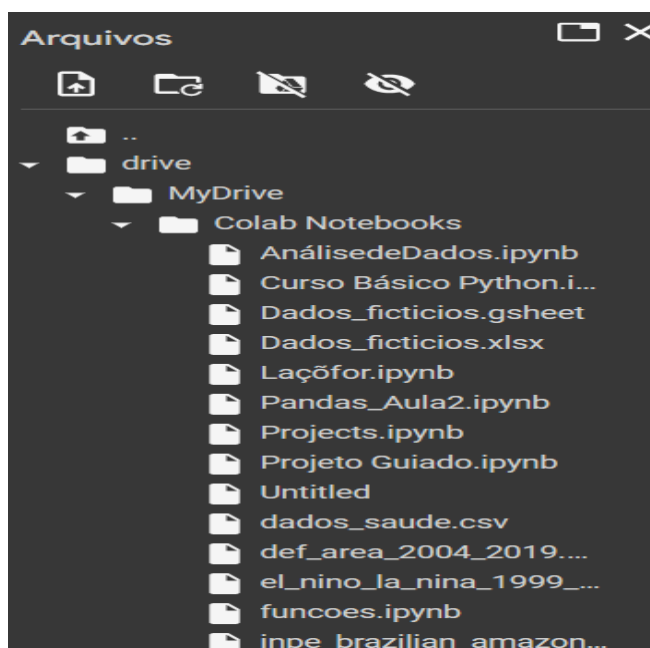
4.2. Importando o *Data Frame* (Quadro de Dados):

```
pg = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/dados_saude.csv') # importação dos dados
pg # print para ver a base de dados
```

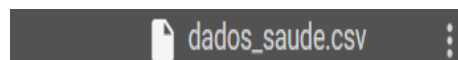
	Unnamed: 0	Sexo	Idade	Altura	Peso	Tipo_Sangue
0	0	0.0	57.0	1.72	67.083	6.0
1	1	0.0	49.0	1.68	65.677	7.0
2	2	1.0	48.0	1.75	71.709	6.0
3	3	1.0	27.0	1.71	67.290	0.0
4	4	0.0	34.0	1.87	67.008	6.0
...
49995	49995	0.0	43.0	1.83	70.012	7.0
49996	49996	0.0	54.0	1.62	64.619	3.0
49997	49997	0.0	23.0	1.72	62.639	4.0
49998	49998	0.0	43.0	1.72	74.847	4.0
49999	49999	1.0	42.0	1.78	63.979	7.0

50000 rows x 6 columns

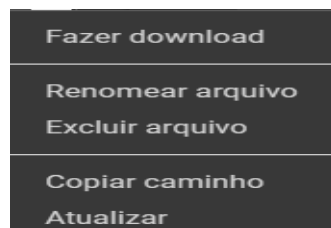
O nome do *Data Frame* (Quadro de Dados) foi tido como *pg*, abreviação de *Projeto Guiado*, uma variável que recebe a leitura (*read*) do arquivo em csv com a cópia do endereço do arquivo no Google Drive, acoplado na mesma pasta dos projetos do Google Collaboratory, veja as imagens a seguir:



A pasta do Drive com o arquivo a ser copiado na próxima imagem:



Clicando nos 3 pontos:



Selecionar a opção “Copiar caminho” e colar entre aspas dentro do parênteses do `.read_csv()`

4.3. Analisando o *Data Frame* (Quadro de Dados):

	Unnamed: 0	Sexo	Idade	Altura	Peso	Tipo_Sangue
0	0	0.0	57.0	1.72	67.083	6.0
1	1	0.0	49.0	1.68	65.677	7.0
2	2	1.0	48.0	1.75	71.709	6.0
3	3	1.0	27.0	1.71	67.290	0.0
4	4	0.0	34.0	1.87	67.008	6.0
...
49995	49995	0.0	43.0	1.83	70.012	7.0
49996	49996	0.0	54.0	1.62	64.619	3.0
49997	49997	0.0	23.0	1.72	62.639	4.0
49998	49998	0.0	43.0	1.72	74.847	4.0
49999	49999	1.0	42.0	1.78	63.979	7.0

50000 rows x 6 columns

Legenda dos dados:

Sexo	Legenda
Masculino	0
Feminino	1

Legenda - Sexo

Tipo Sanguíneo	A+	A-	B+	B-	O+	O-	AB+	AB-
Legenda	0	1	2	3	4	5	6	7

Legenda - Tipo Sanguíneo

Peso: está em quilogramas.

Altura: está em metros.

Idade: está em anos.

OBS: Nota-se que é uma base de dados com 50000 linhas e 6 colunas (foi criado uma nova coluna que não serve para a análise, a “Unnamed: 0”, porque é um novo

Index, algo que já há a presença), logo o próximo tópico é excluir tal coluna desnecessária.

4.4. Excluindo coluna desnecessária do *Data Frame* (*Quadro de Dados*):

```
pg = pg.drop('Unnamed: 0', axis = 1) # excluindo a coluna desnecessária criada
pg # print para ver a base de dados
```

Sobre o comando `drop()`:

Esse comando serve para excluir uma linha ou uma coluna, método que dentro do parênteses recebe o nome da coluna ou linha, tal qual, o *axis*, que é o do eixo horizontal (zero) - “linha” ou vertical (1) - “coluna”.

Data Frame:

	Sexo	Idade	Altura	Peso	Tipo_Sangue
0	0.0	57.0	1.72	67.083	6.0
1	0.0	49.0	1.68	65.677	7.0
2	1.0	48.0	1.75	71.709	6.0
3	1.0	27.0	1.71	67.290	0.0
4	0.0	34.0	1.87	67.008	6.0
...
49995	0.0	43.0	1.83	70.012	7.0
49996	0.0	54.0	1.62	64.619	3.0
49997	0.0	23.0	1.72	62.639	4.0
49998	0.0	43.0	1.72	74.847	4.0
49999	1.0	42.0	1.78	63.979	7.0

50000 rows x 5 columns

Data Frame sem a coluna desnecessária.

OBS: 5 colunas fazem parte do *Data Frame*.

4.5. Breve descrição do *Data Frame* (*Quadro de Dados*):

Comando utilizado:

```
pg.describe() # informações importantes sobre os dados, como ter 50000 linhas, a média, o desvio padrão, o mínimo e o máximo
```

.describe():

Esse comando em Python serve para visualizar alguns detalhes estatísticos básicos como percentual, média, desvio padrão, mínimo, máximo etc. de um quadro de dados.

Data Frame:

	Sexo	Idade	Altura	Peso	Tipo_Sangue
count	50000.000000	50000.000000	50000.000000	50000.000000	50000.000000
mean	0.500900	38.534800	1.719859	69.935756	3.48902
std	0.500004	12.113041	0.060319	4.976680	2.28966
min	0.000000	18.000000	1.450000	49.791000	0.00000
25%	0.000000	28.000000	1.680000	66.580000	1.00000
50%	1.000000	39.000000	1.720000	69.945000	3.00000
75%	1.000000	49.000000	1.760000	73.285000	5.00000
max	1.000000	59.000000	1.980000	91.139000	7.00000

Usuários Cadastrados:

50000 (número do registro de 0 a 49999)

Idade:

Menor usuário cadastrado tem 18 anos.

Maior usuário cadastrado tem 59 anos.

Peso:

O usuário mais leve pesa 49,79 kg.

O usuário mais pesado pesa 91,14 kg.

Altura:

O menor usuário tem 1,45 m de altura.

O maior usuário tem 1,98m de altura.

Comando utilizado:

```
pg.shape # sabendo o tamanho do dataframe, nesse caso tem 50000 linhas e 5 colunas  
(50000, 5)
```

.shape:

Esse comando em Python serve para determinar a extensão do *Data Frame*, retornando entre parênteses a quantidade de linhas e colunas, respectivamente.

OBS: Todas as saídas são do tipo float (flutuante), ou seja, valores numéricos com (.) e não há a presença de nenhum dado vazio. Tal referência é analisada pelo atributo a seguir:

```
pg.info() # sabendo as informações da base de dados, observa-se que todas as saídas são do tipo float (flutuante)
```

#	Column	Non-Null Count	Dtype
0	Sexo	50000 non-null	float64
1	Idade	50000 non-null	float64
2	Altura	50000 non-null	float64
3	Peso	50000 non-null	float64
4	Tipo_Sangue	50000 non-null	float64

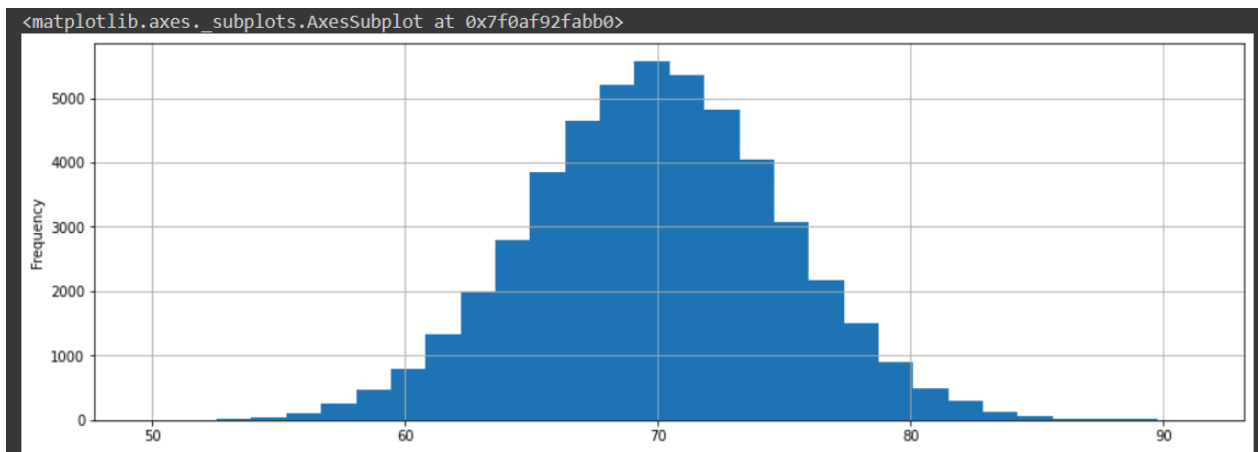
dtypes: float64(5)

4.6. Histograma do Peso:

Código Utilizado:

```
pg.Peso.plot.hist(figsize = (15, 5), bins=30, grid = True)
```

Histograma:



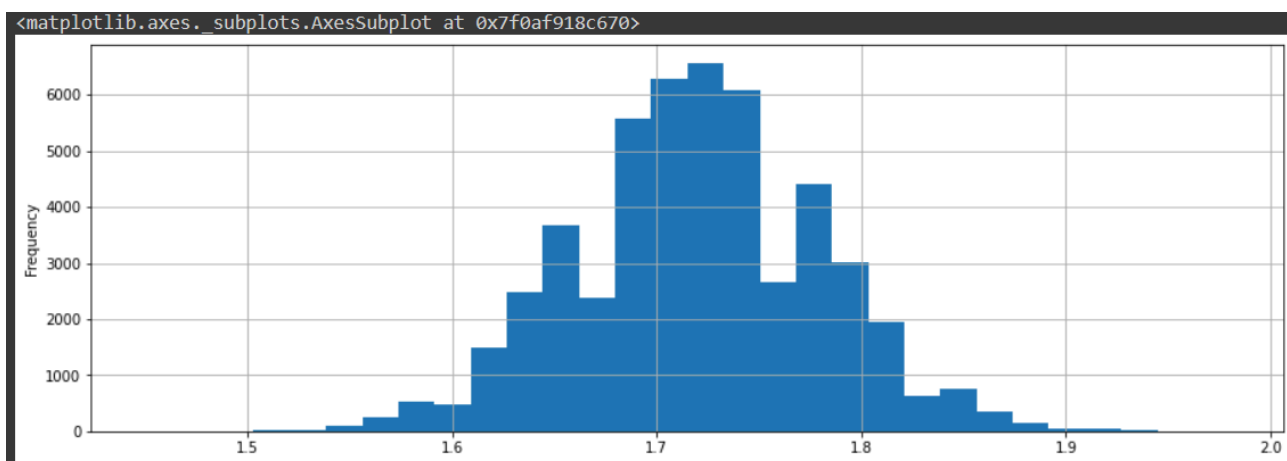
Distribuição Gaussiana

4.7. Histograma da Altura:

Código Utilizado:

```
pg.Altura.plot.hist(figsize = (15, 5), bins=30, grid = True)
```

Histograma:



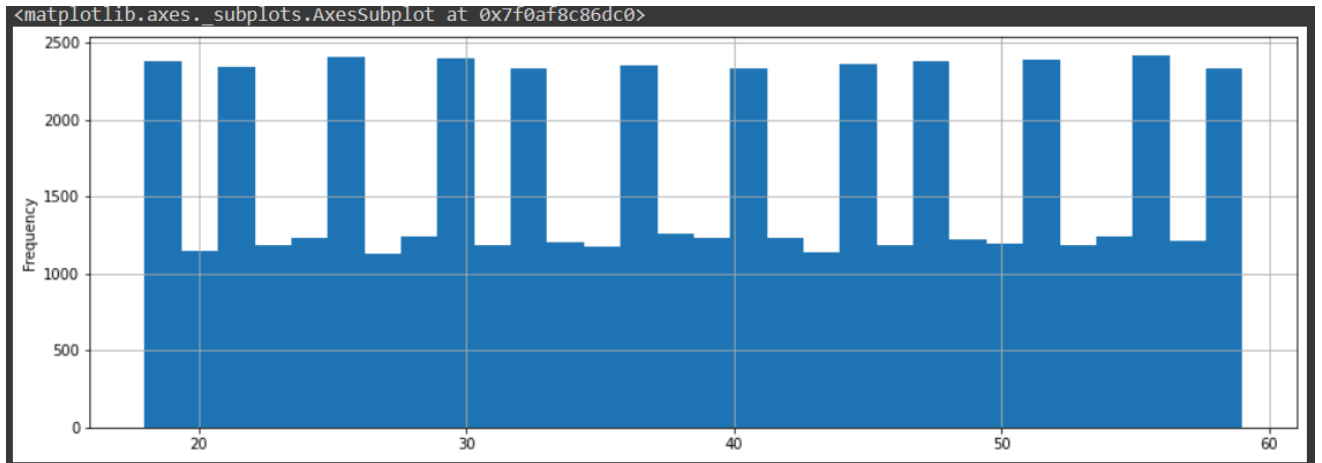
Distribuição Gaussiana

4.8. Histograma da Idade:

Código Utilizado:

```
pg.Idade.plot.hist(figsize = (15, 5), bins=30, grid = True)
```

Histograma:



Distribuição não Gaussiana

Explicação:

Para plotar o histograma, basta colocar o nome do *Data Frame* (nesse caso, *pg*), inserir o ponto (.) e colocar o nome da coluna, inserir o ponto (.) novamente, e utilizar do recurso *plot.hist()* para configurar a saída gráfica. Por fim, dentro deste recurso, há 3 parâmetros: o *figsize = (largura, altura)* - ajusta o gráfico; o *bins = numero_de_barras* - configurar a quantidade de barras e o *grid = True* - adicionar grade para visualizar melhor.

4.9. Sexo dos Usuários:

Quantidades:

Comando utilizado:

```
pg.Sexo.value counts() # para obter as quantidades
```

Saída:

```
1.0    25045
0.0    24955
Name: Sexo, dtype: int64
```

Porcentagens:

Comando utilizado:

```
pg.Sexo.value_counts(normalize=True)*100 # para obter as porcentagens
```

Saída:

```
1.0     50.09
0.0     49.91
Name: Sexo, dtype: float64
```

.value_counts():

Esse comando em Python serve para retornar a distribuição dos valores.

Histograma com o .value_counts():

Comando utilizado:

```
pg.Sexo.value_counts().plot.bar(figsize = (10, 5)) # gráficos de barras dos sexos do usuário
```

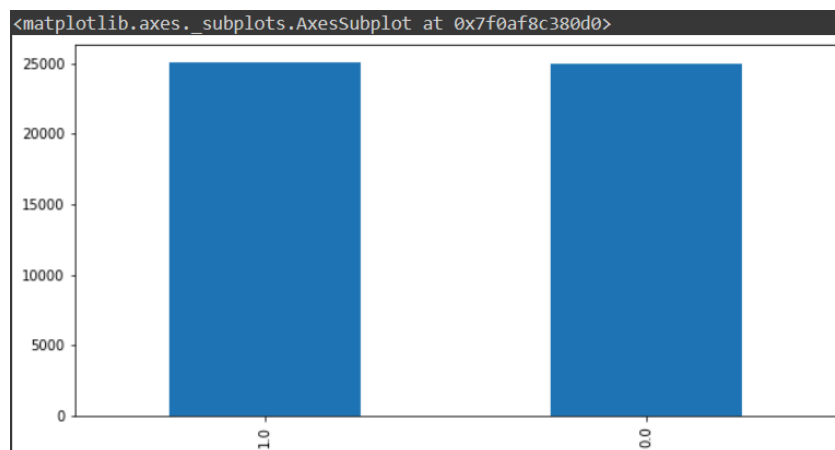


Tabela para determinar as quantidades e percentuais:

Sexo	Quantidade	Porcentagem
Masculino	24955	49,91%
Feminino	25045	50,09%

Considerações:

A porcentagem de mulheres permanece acima dos homens, mas com pouca diferença de 0,18%. Assim, há uma distribuição bem equilibrada, com 24955 registros masculinos e 25045 registros femininos.

4.0.1. Tipo de Sangue dos Usuários:

Quantidades:

Comando utilizado:

```
pg.Tipo_Sangue.value_counts() # para obter as quantidades
```

```
5.0    6324
3.0    6310
1.0    6298
2.0    6272
0.0    6270
6.0    6206
7.0    6181
4.0    6139
Name: Tipo_Sangue, dtype: int64
```

Saída:

Porcentagens:

Comando utilizado:

```
pg.Tipo_Sangue.value_counts(normalize = True) * 100 # para obter as porcentagens
```

```

5.0    12.648
3.0    12.620
1.0    12.596
2.0    12.544
0.0    12.540
6.0    12.412
7.0    12.362
4.0    12.278
Name: Tipo_Sangue, dtype: float64

```

Saída:

.value_counts():

Esse comando em Python serve para retornar à distribuição dos valores.

Histograma com o .value_counts():

Comando utilizado:

```

pg.Tipo_Sangue.value_counts().plot.bar(figsize = (10, 5)) # gráficos de barras dos tipos sanguíneos dos usuários

```

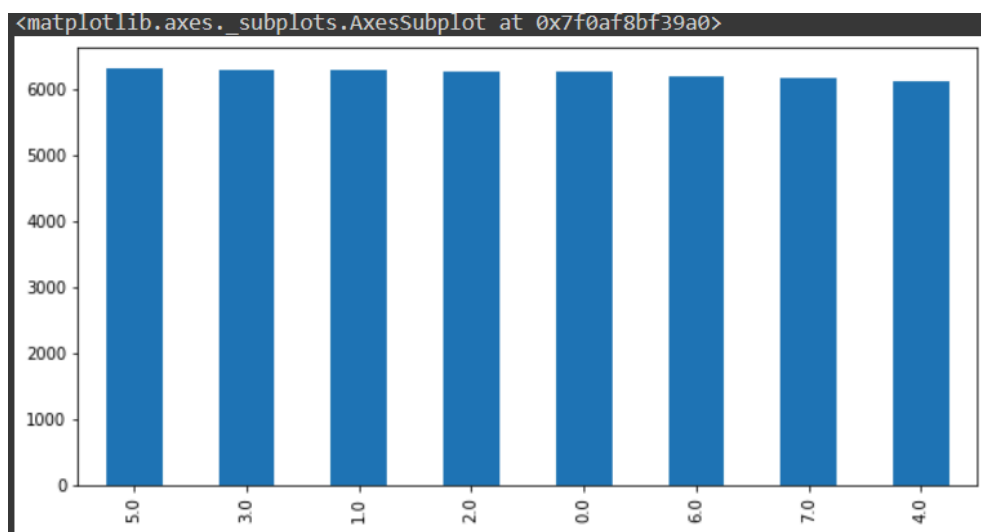


Tabela para determinar as quantidades e percentuais:

Tipo Sanguíneo	A+	A-	B+	B-	O+	O-	AB+	AB-
Quantidade	6270	6298	6272	6310	6139	6324	6206	6181
Porcentagem	12,54%	12,596%	12,544 %	12,62%	12,278%	12,648 %	12,412 %	12,362 %

Considerações:

A porcentagem do tipo sanguíneo dos usuários permanece equilibrada, baseando-se cada tipo de sangue num registro marcado por 12000 apontamentos e estabilidade no percentual de 12%. Por fim, pode-se notar que o tipo de sangue mais recorrente dos registrados é o 5.0 (O-) e o menos recorrente é o 4.0 (O+).

4.0.2. IMC (Índice de Massa Corporal) dos Usuários:

IMC	CLASSIFICAÇÃO	OBESIDADE (GRAU)
MENOR QUE 18,5	MAGREZA	0
ENTRE 18,5 E 24,9	NORMAL	0
ENTRE 25,0 E 29,9	SOBREPESO	I
ENTRE 30,0 E 39,9	OBESIDADE	II
MAIOR QUE 40,0	OBESIDADE GRAVE	III

$$IMC = \frac{Peso}{altura^2}$$

Criando coluna:

```
pg['IMC'] = pg['Peso'] / (pg['Altura'] ** 2) # criando nova coluna do IMC
```

Para criar uma nova coluna, basta adicionar o nome de tal entre “[]”, atribuindo a uma variável pela utilização do “=”, assim, realizando o cálculo do lado direito do operador.

Data Frame com a nova coluna:

	Sexo	Idade	Altura	Peso	Tipo_Sangue	IMC
0	0.0	57.0	1.72	67.083	6.0	22.675433
1	0.0	49.0	1.68	65.677	7.0	23.269912
2	1.0	48.0	1.75	71.709	6.0	23.415184
3	1.0	27.0	1.71	67.290	0.0	23.012209
4	0.0	34.0	1.87	67.008	6.0	19.162115
...
49995	0.0	43.0	1.83	70.012	7.0	20.905969
49996	0.0	54.0	1.62	64.619	3.0	24.622390
49997	0.0	23.0	1.72	62.639	4.0	21.173269
49998	0.0	43.0	1.72	74.847	4.0	25.299824
49999	1.0	42.0	1.78	63.979	7.0	20.192842

50000 rows x 6 columns

4.0.3. Obesidade Grave:

Comando utilizado:

```
obesidade_grave = pg.query('IMC > 40.0') # criando a query
```

```
obesidade_grave
```

Printando Data Frame

.query():

Esse comando em Python serve para consultar resultados de pesquisa.

```
obesidade_grave
```

Saída:

```
Sexo  Idade  Altura  Peso  Tipo_Sangue  IMC
```

Considerações:

Nenhum usuário sofre de obesidade grave, ou seja, IMC > 40.0.

4.0.4. Magreza:

Comando utilizado:

```
magreza = pg.query('IMC < 18.5') # criando a query
```

```
magreza
```

Printando Data Frame

```
magreza.head()
```

magreza.head() para otimizar o espaço e a visualização.

.head():

Esse comando em Python serve para meta-informações escritas em cabeçalhos de resposta, sem ter que transportar todo o conteúdo.


```
magreza.head()
```

	Sexo	Idade	Altura	Peso	Tipo_Sangue	IMC
26	0.0	32.0	1.81	60.398	4.0	18.435945
27	0.0	25.0	1.85	61.712	7.0	18.031264
160	1.0	51.0	1.84	58.905	4.0	17.398689
188	0.0	25.0	1.83	59.597	3.0	17.795993
272	1.0	48.0	1.78	56.200	5.0	17.737659

Saída:

Quantidades e Porcentagens:

```
magreza.Sexo.value_counts() # para obter as quantidades

0.0    209
1.0    209
Name: Sexo, dtype: int64

magreza.Sexo.value_counts(normalize=True)*100 # para obter as porcentagens

0.0    50.0
1.0    50.0
Name: Sexo, dtype: float64
```

Considerações:

As pessoas que sofrem de 'Magreza', segundo o IMC, não apresentam uma distribuição do tipo sexo muito diferente dos usuários em geral, já que são 219 usuários para cada tipo sexo, e 418 usuários no geral, configurando numa porcentagem de 50% para cada segmento.

4.0.5. Obesidade Grau II:

Comando utilizado:

```
obesidade_grauII = pg.query('IMC >= 30.0 & IMC <= 39.9') # criando a query
obesidade_grauII
```

Printando Data Frame

```
obesidade_grauII.head()
```

obesidade_grauII.head() para otimizar o espaço e a visualização.

```
obesidade_grauII.head()
```

	Sexo	Idade	Altura	Peso	Tipo_Sangue	IMC
23	0.0	43.0	1.58	75.191	6.0	30.119772
112	0.0	19.0	1.64	81.722	6.0	30.384444
166	0.0	39.0	1.72	88.790	7.0	30.012845
172	0.0	45.0	1.58	75.887	0.0	30.398574
179	0.0	49.0	1.56	73.178	3.0	30.069855

Saída:

Quantidades e Porcentagens dos Tipos de Sangue:

```
obesidade_grauII.Tipo_Sangue.value_counts() # para obter as quantidades
```

```
7.0    57
0.0    55
5.0    55
3.0    54
6.0    53
2.0    52
4.0    48
1.0    47
Name: Tipo_Sangue, dtype: int64
```

```
obesidade_grauII.Tipo_Sangue.value_counts(normalize=True)*100 # para obter as porcentagens
```

```
7.0    13.539192
0.0    13.064133
5.0    13.064133
3.0    12.826603
6.0    12.589074
2.0    12.351544
4.0    11.401425
1.0    11.163895
Name: Tipo_Sangue, dtype: float64
```

Considerações:

As pessoas que sofrem de ‘Obesidade Grau II’, segundo o IMC, não apresentam uma distribuição de tipo sanguíneo muito diferente dos usuários em geral, já que o tipo de sangue mais presente nessas pessoas é o 7.0 (AB-), com 13,54% aproximadamente em relação ao geral, e pouca diferença em relação ao menos presente, o 1.0 (A-), com 11,16% aproximadamente.

4.0.6. Médias de Peso e Idade das pessoas que sofrem de “Magreza”:

Comando utilizado:

```
magreza.value_counts() # quantas pessoas sofrem de magreza
```

Sexo	Idade	Altura	Peso	Tipo_Sangue	IMC	
0.0	18.0	1.79	59.197	1.0	18.475391	1
1.0	39.0	1.78	56.635	0.0	17.874953	1
	34.0	1.71	52.776	5.0	18.048630	1
	33.0	1.91	66.373	0.0	18.193854	1
	32.0	1.84	55.109	4.0	16.277469	1
						..
0.0	43.0	1.82	59.348	7.0	17.916918	1
		1.81	58.395	1.0	17.824547	1
		1.72	54.466	7.0	18.410627	1
	42.0	1.92	66.252	1.0	17.972005	1
1.0	59.0	1.84	59.182	1.0	17.480506	1

Length: 418, dtype: int64

418 pessoas sofrem de “Magreza”.

```
magreza.describe()
```

	Sexo	Idade	Altura	Peso	Tipo_Sangue	IMC
count	418.000000	418.000000	418.000000	418.000000	418.000000	418.000000
mean	0.500000	38.138756	1.822775	59.472141	3.444976	17.895650

A média de peso desse grupo é de 59,48 kg aproximadamente.

A média de idade desse grupo é de 38 anos aproximadamente.

Justificativa:

```
magreza.Peso.describe()
```

count	418.000000
mean	59.472141
std	3.253882
min	49.791000
25%	57.437500
50%	59.295500
75%	61.657500
max	67.329000

Name: Peso, dtype: float64

A média se aproxima de 59,48 kg, com o mínimo de 49,79 kg e o máximo de 67,33 kg.

```
magreza.Idade.describe()

count    418.000000
mean      38.138756
std       12.049255
min       18.000000
25%       28.000000
50%       37.000000
75%       49.000000
max       59.000000
Name: Idade, dtype: float64
```

A média se aproxima de 38 anos, com o mínimo de 18 anos e o máximo de 59 anos.

4.0.7. Obesidade Grau II Feminina - Mulher com mais Peso:

Comando utilizado:

```
obesas_grauII = pg.query('Sexo == 1.0 & IMC >= 30.0 & IMC <= 39.9') # criando a query
```

Em análise ao código, observa-se a presença de uma nova variável, *obesas_grauII*, atribuída a uma query que consulta o “Sexo” ser igual a 1.0 (Feminino) e o IMC ser da classificação de Obesidade Grau II, tal qual, as duas condições unidas pelo &, na qual une essas duas circunstâncias.

```
obesas_grauII.head()
```

```
obesas_grauII.head()
```

	Sexo	Idade	Altura	Peso	Tipo_Sangue	IMC
279	1.0	40.0	1.60	76.831	5.0	30.012109
418	1.0	39.0	1.62	81.092	3.0	30.899253
1000	1.0	48.0	1.64	85.354	4.0	31.734830
1121	1.0	26.0	1.56	75.087	6.0	30.854290
1182	1.0	49.0	1.61	78.807	7.0	30.402762

```
obesas_grauII.describe()
```

	Sexo	Idade	Altura	Peso	Tipo_Sangue	IMC
count	216.0	216.000000	216.000000	216.000000	216.000000	216.000000
mean	1.0	37.981481	1.594259	78.917398	3.592593	31.048009
std	0.0	11.620336	0.038087	3.740629	2.300807	1.000353
min	1.0	18.000000	1.480000	67.414000	0.000000	30.004244
25%	1.0	29.000000	1.570000	76.526750	2.000000	30.346262
50%	1.0	39.000000	1.590000	78.901500	4.000000	30.729719
75%	1.0	48.000000	1.620000	81.115250	6.000000	31.455639
max	1.0	59.000000	1.680000	89.540000	7.000000	35.268787

A mulher com mais peso, pesa 89,54 kg.

Registro da mulher mais obesa:

```
mais_obesa = pg.query('Sexo == 1.0 & Peso == 89.54') # criando a query
```

Em análise ao código, observa-se a presença do mesmo critério do anterior, uma query que consulta o “Sexo” ser igual a 1.0 (Feminino), mas com a busca pelo máximo peso obtido pelo método `.describe()`.

mais obesa

Chamada da nova variável

Registro:

15096

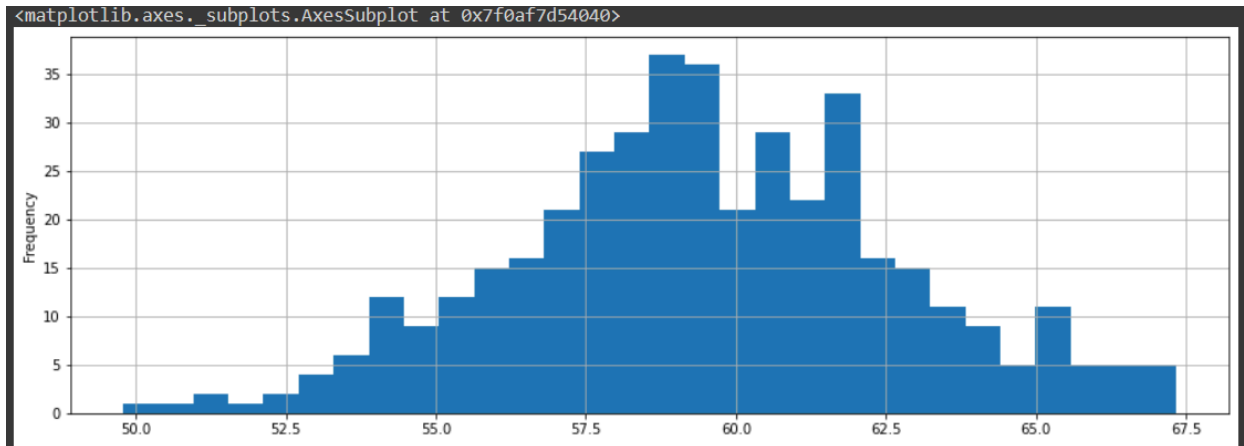
	Sexo	Idade	Altura	Peso	Tipo_Sangue	IMC
15096	1.0	43.0	1.61	89.54	0.0	34.54342

4.0.8. Histograma das pessoas com “Magreza” e dos usuários em geral:

Magreza:

```
# Histograma com o peso das pessoas que sofrem de Magreza:
magreza.Peso.plot.hist(figsize = (15, 5), bins=30, grid = True)
```

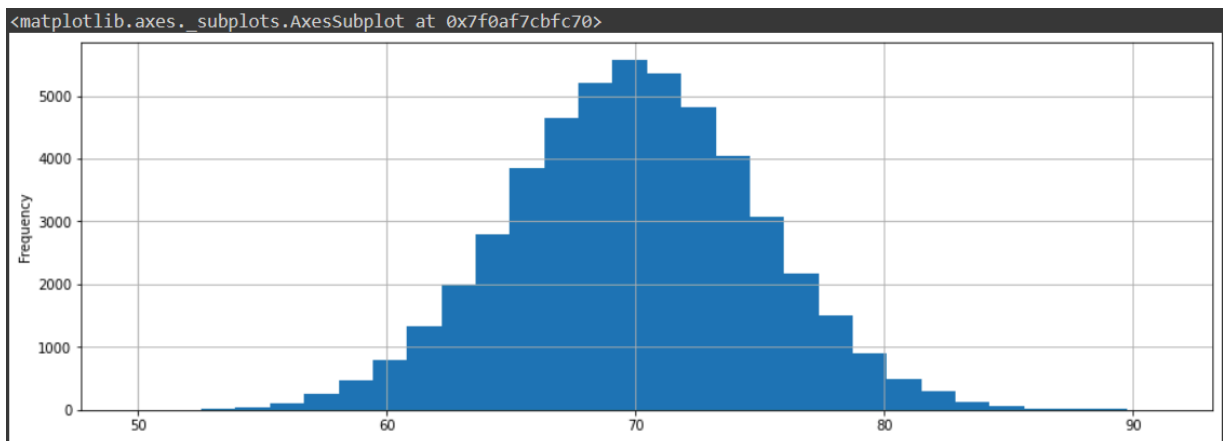
Histograma:



Usuários em geral:

```
# Histograma com o peso dos usuários em geral:
pg.Peso.plot.hist(figsize = (15, 5), bins=30, grid = True)
```

Histograma:



Análise:

O histograma com o peso das pessoas que sofrem de Magreza, não alcança a casa dos 67,5 kg e se aproxima menos de uma distribuição gaussiana do que a dos usuários em geral, que por sua vez, quase chega com pouca frequência a casa dos

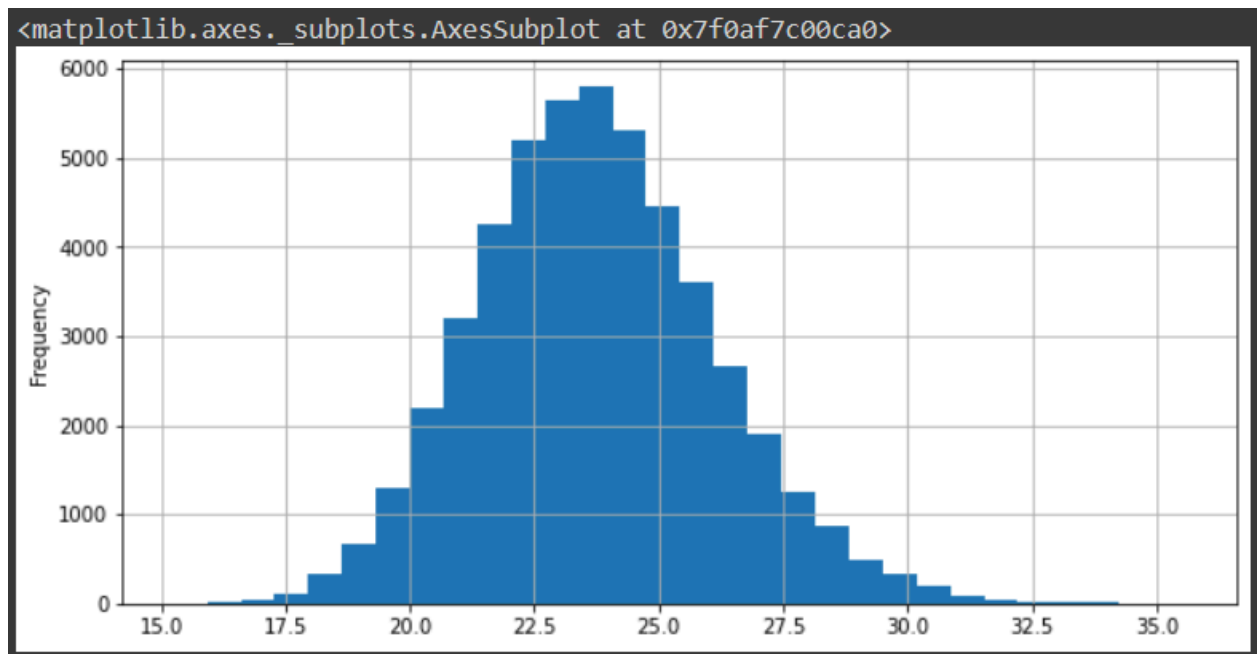
90 kg e se estabelece consideravelmente na casa dos 60 kg aos 80 kg. Assim, pode-se dizer que o histograma das pessoas com Magreza apresenta dos 50 kg aos 67,5 kg, e dos usuários em geral na casa dos 55 kg aos 90 kg.

4.0.9. Histograma do IMC dos usuários em geral:

Comando utilizado:

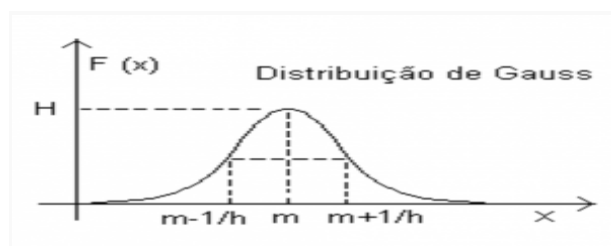
```
pg.IMC.plot.hist(figsize=(10, 5), bins=30, grid=True)
```

Histograma:



Análise:

O histograma do IMC dos usuários em geral apresenta uma distribuição normal (gaussiana), já que tem o formato de um sino. Veja a imagem abaixo:



5.0.0. Conclusão:

Por conseguinte, a toda essa Análise de Dados na linguagem de programação Python, observa-se que referente aos dados com informações dos usuários do sistema municipal de saúde de Sucuri do Sul, há uma porcentagem de mulheres que permanece acima dos homens, mas com pouca diferença de 0,18%, compreendendo-se numa distribuição bem equilibrada, com 24955 registros masculinos e 25045 registros femininos.

Ademais, a porcentagem do tipo sanguíneo dos usuários permanece equilibrada, baseando-se cada tipo de sangue num registro marcado por 12000 apontamentos e estabilidade no percentual de 12%. Por fim, pode-se notar que o tipo de sangue mais recorrente dos registrados é o 5.0 (O-) e o menos recorrente é o 4.0 (O+).

Mediante ao exposto, o histograma de Peso apresenta uma distribuição gaussiana (tem formato de sino), o histograma de Altura se aproxima de uma distribuição gaussiana e o histograma de Idade não se modula como uma distribuição gaussiana.

Destarte, nenhum usuário sofre de obesidade grave, ou seja, $IMC > 40.0$. Assim como, as pessoas que sofrem de “Magreza”, segundo o IMC, não apresentam uma distribuição do tipo sexo muito diferente dos usuários em geral, já que são 219 usuários para cada tipo sexo, e 418 usuários no geral, configurando numa porcentagem de 50% para cada segmento.

Diante disso, as pessoas que sofrem de “Obesidade Grau II”, segundo o IMC, não apresentam uma distribuição de tipo sanguíneo muito diferente dos usuários em geral, já que o tipo de sangue mais presente nessas pessoas é o 7.0 (AB-), com 13,54% aproximadamente em relação ao geral, e pouca diferença em relação ao menos presente, o 1.0 (A-), com 11,16% aproximadamente.

Em síntese, a média de peso das pessoas que sofrem de magreza é de 59,48 kg aproximadamente e a média de idade desse grupo é de 38 anos. Tal qual, a mulher com mais peso, na classificação de obesidade grau II, pesa 89,54 kg e seu registro é 15096.

Portanto, o histograma com o peso das pessoas que sofrem de magreza, não alcança a casa dos 67,5 kg e se aproxima menos de uma distribuição gaussiana do que a dos usuários em geral, que por sua vez, quase chega com pouca frequência a casa dos 90 kg e se estabelece consideravelmente na casa dos 60 kg aos 80 kg. Assim, pode-se dizer que o histograma das pessoas com Magreza apresenta dos 50 kg aos 67,5 kg, e dos usuários em geral na casa dos 55 kg aos 90 kg. Assim sendo, o histograma do IMC dos usuários em geral apresenta uma distribuição normal (gaussiana), já que tem o formato de um sino.

Em última análise, há 50000 usuários cadastrados (número do registro de 0 a 49999). O menor usuário cadastrado tem 18 anos e o maior tem 59 anos. O usuário mais leve pesa 49,79 kg e o mais pesado pesa 91,14 kg. O menor usuário tem 1,45 m de altura e o maior usuário tem 1,98m de altura.

6.0.0. Referências:

- [Distribuição Gaussiana – Seara da Ciência \(ufc.br\)](#)