

analise_de_dados_oae

May 28, 2022

```
[335]: # importando bibliotecas
import pandas as pd
import numpy as np
import io
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[336]: # importar arquivos
from google.colab import files
uploaded = files.upload()
```

<IPython.core.display.HTML object>

Saving oae_dados_responsaveis.csv to oae_dados_responsaveis.csv

Saving oae_dados_comunidade.csv to oae_dados_comunidade.csv

Saving oae_dados_alunos.csv to oae_dados_alunos.csv

```
[337]: # criando o dataframe alunos
df_alunos = pd.read_csv('oae_dados_alunos.csv')
df_alunos.head(5)
```

```
[337]:
```

	nome	idade	genero	tipo	serie	tempo_moradia	transporte \
0	Karla	10	feminino	aluno	1a5	5a10	carro
1	anônimo	8	masculino	aluno	1a5	2a5	van
2	Jéssica Silva	8	feminino	aluno	1a5	2a5	ônibus
3	João Pedro	13	outro	aluno	6a9	5a10	ônibus
4	Lucas	11	masculino	aluno	1a5	10a20	nenhum

	vale_transp	tempo_desl	qualidade_ens	mudar	ajuda_comunidade	mais_comunidade
0	não	10	boa	sim	sim	sim
1	não	20	regular	sim	não	sim
2	sim	30	regular	sim	sim	sim
3	sim	40	ruim	sim	sim	sim
4	não	6	regular	não	não	sim

```
[338]: # criando o dataframe responsáveis
df_responsaveis = pd.read_csv('oae_dados_responsaveis.csv')
```

```
df_responsaveis.head(5)
```

```
[338]:
```

	nome	idade	genero	tipo	serie	tempo_moradia \
0	Alfredo Alves	42	masculino	responsável	1a5	20a40
1	Maria Aparecida	55	feminino	responsável	6a9	20a40
2	Marcia	60	feminino	responsável	6a9	mais40
3	anônimo	36	masculino	responsável	1a5	20a40
4	Joana	50	feminino	responsável	6a9	mais40

	qualidade_ens	ajuda_comunidade	mais_comunidade	participa_escola
0	boa	sim	sim	nunca
1	boa	sim	sim	algumas vezes
2	regular	sim	sim	nunca
3	regular	sim	sim	nunca
4	boa	sim	sim	algumas vezes

```
[339]: # criando o dataframe comunidade
df_comunidade = pd.read_csv('oae_dados_comunidade.csv')
df_comunidade.head(30)
```

```
[339]:
```

	nome	idade	genero	tipo	tempo_moradia \
0	anônimo	20	feminino	membro	20a40
1	Joana	35	feminino	membro	20a40
2	Maria	66	feminino	membro	mais40
3	anônimo	45	masculino	membro	20a40
4	anônimo	64	feminino	membro	mais40
5	Maria Lopes	50	feminino	membro	6a10
6	André	58	masculino	membro	mais40
7	Luis Souza	53	masculino	membro	6a10
8	Marcoa Alonso	23	masculino	membro	0a5
9	anônimo	41	feminino	membro	20a40
10	Mariangêla Soneda	42	feminino	membro	0a5
11	anônimo	57	feminino	membro	20a40
12	Otávio Gomes	75	masculino	membro	20a40
13	Nilza Souza	62	feminino	membro	40a60

	matricularia_filho	ajuda_comunidade	mais_comunidade
0	sim	sim	sim
1	sim	sim	sim
2	sim	sim	sim
3	sim	sim	sim
4	sim	sim	sim
5	sim	sim	sim
6	sim	sim	sim
7	não	sim	sim
8	sim	sim	sim
9	sim	sim	sim

10	sim	sim	sim
11	sim	sim	sim
12	sim	sim	sim
13	não	não	não

```
[340]: #Exibindo informações sobre o conjunto de dados (alunos)
df_alunos.info()
# Computa estatísticas de atributos numéricos (alunos)
df_alunos.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24 entries, 0 to 23
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   nome                  24 non-null    object
1   idade                 24 non-null    int64
2   genero                24 non-null    object
3   tipo                  24 non-null    object
4   serie                 24 non-null    object
5   tempo_moradia         24 non-null    object
6   transporte            24 non-null    object
7   vale_transp           24 non-null    object
8   tempo_desl            24 non-null    int64
9   qualidade_ens         24 non-null    object
10  mudar                 24 non-null    object
11  ajuda_comunidade      24 non-null    object
12  mais_comunidade       24 non-null    object
dtypes: int64(2), object(11)
memory usage: 2.6+ KB
```

```
[340]:          idade  tempo_desl
count  24.000000  24.000000
mean    11.583333  25.000000
std     3.775397  20.214072
min     7.000000   3.000000
25%     8.750000   9.500000
50%    10.500000  22.500000
75%    13.250000  31.250000
max    20.000000  90.000000
```

```
[341]: #Exibindo informações sobre o conjunto de dados (responsáveis)
df_responsaveis.info()
# Computa estatísticas de atributos numéricos (responsáveis)
df_responsaveis.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
```

```
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   nome                   25 non-null   object
1   idade                  25 non-null   int64
2   genero                 25 non-null   object
3   tipo                   25 non-null   object
4   serie                  25 non-null   object
5   tempo_moradia          25 non-null   object
6   qualidade_ens           25 non-null   object
7   ajuda_comunidade       25 non-null   object
8   mais_comunidade        25 non-null   object
9   participa_escola       25 non-null   object
dtypes: int64(1), object(9)
memory usage: 2.1+ KB
```

```
[341]:          idade
count    25.000000
mean     43.600000
std       7.421815
min      30.000000
25%      39.000000
50%      42.000000
75%      49.000000
max      60.000000
```

```
[342]: #Exibindo informações sobre o conjunto de dados (comunidade)
df_comunidade.info()
# Computa estatísticas de atributos numéricos (comunidade)
df_comunidade.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   nome                   14 non-null   object
1   idade                  14 non-null   int64
2   genero                 14 non-null   object
3   tipo                   14 non-null   object
4   tempo_moradia          14 non-null   object
5   matricularia_filho     14 non-null   object
6   ajuda_comunidade       14 non-null   object
7   mais_comunidade        14 non-null   object
dtypes: int64(1), object(7)
memory usage: 1.0+ KB
```

```
[342]:
```

	idade
count	14.000000
mean	49.357143
std	16.079644
min	20.000000
25%	41.250000
50%	51.500000
75%	61.000000
max	75.000000

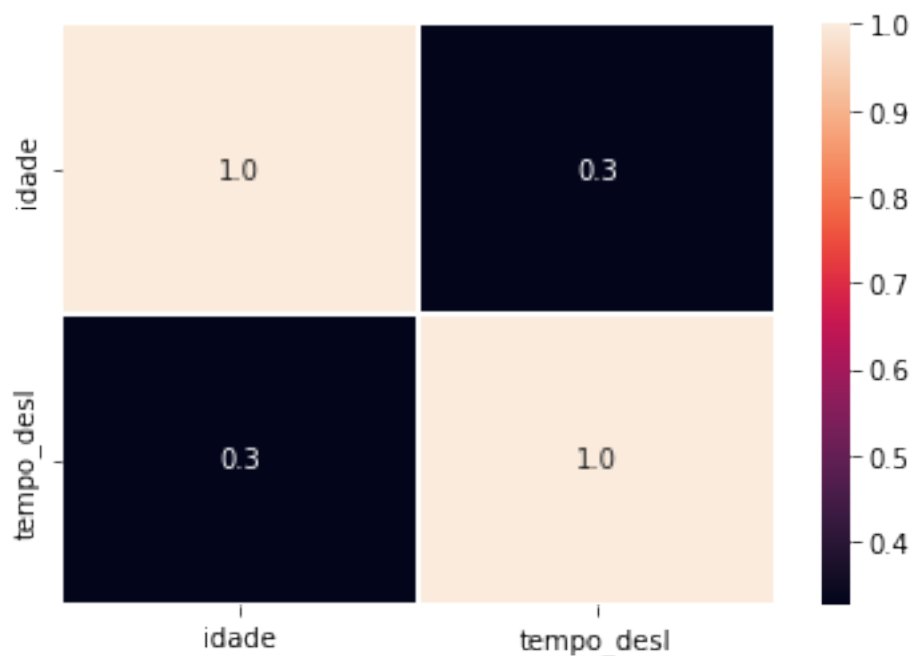
Todos os campos ficaram completos, pois o aluno não pode enviar o formulário até que todas as respostas estejam respondidas. O único campo que pode ficar em branco é o 'nome', porém é salvo como 'anônimo'. O campo idade só aceita números entre 1 e 100.

1 Análise dos dados coletados dos alunos

1.0.1 Corelação entre idade e tempo de deslocamento

```
[343]: correlacao = df_alunos.corr()
mapa_correlacao = sns.heatmap(correlacao, annot = True, fmt=".1f", linewidths=.
↪6)
mapa_correlacao
```

```
[343]: <matplotlib.axes._subplots.AxesSubplot at 0x7faf3f610bd0>
```



1.0.2 A porcentagem de alunos que preferiram não se identificar

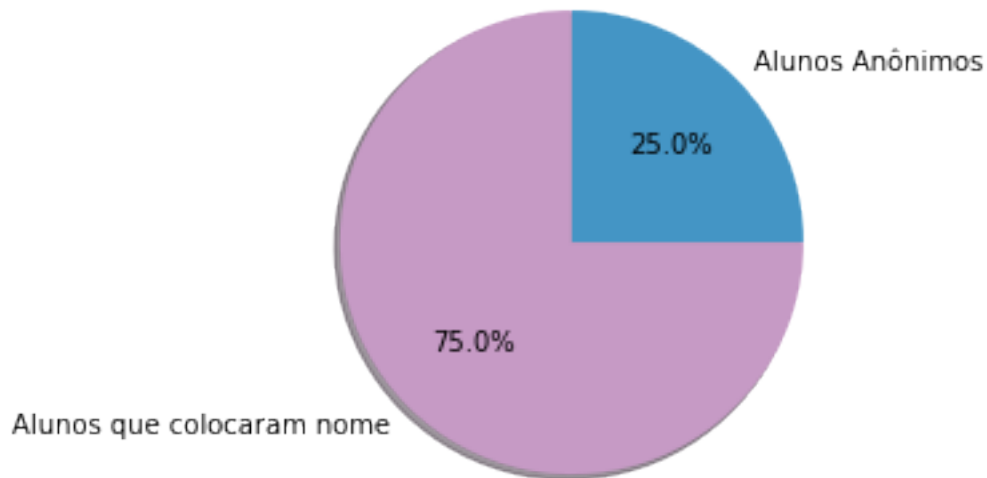
```
[344]: # definindo os dados
df_alunos_lista = list(df_alunos['nome'])
alunos_anonimos = df_alunos_lista.count('anônimo')

total_alunos = df_alunos['nome'].count()

alunos_com_nome = total_alunos - alunos_anonimos

[345]: # montando o gráfico
x = [alunos_anonimos, alunos_com_nome]
mylabels = ["Alunos Anônimos", "Alunos que colocaram nome"]
mycolors = ['#4495c5', '#c69ac5']

plt.pie(x, labels = mylabels, shadow = True, autopct = '%1.1f%%', colors =
→mycolors)
plt.show()
```



1.0.3 Quantidade de alunos que responderam o questionário com relação a idade

```
[346]: # definindo os dados
agrupar_idade = pd.DataFrame(df_alunos.groupby('idade').size())
tamanho_alunos = df_alunos.groupby('idade').size()
```

```

occur_idade = np.array(agrupar_idade.index)
occur_idade_lista = list(occur_idade)

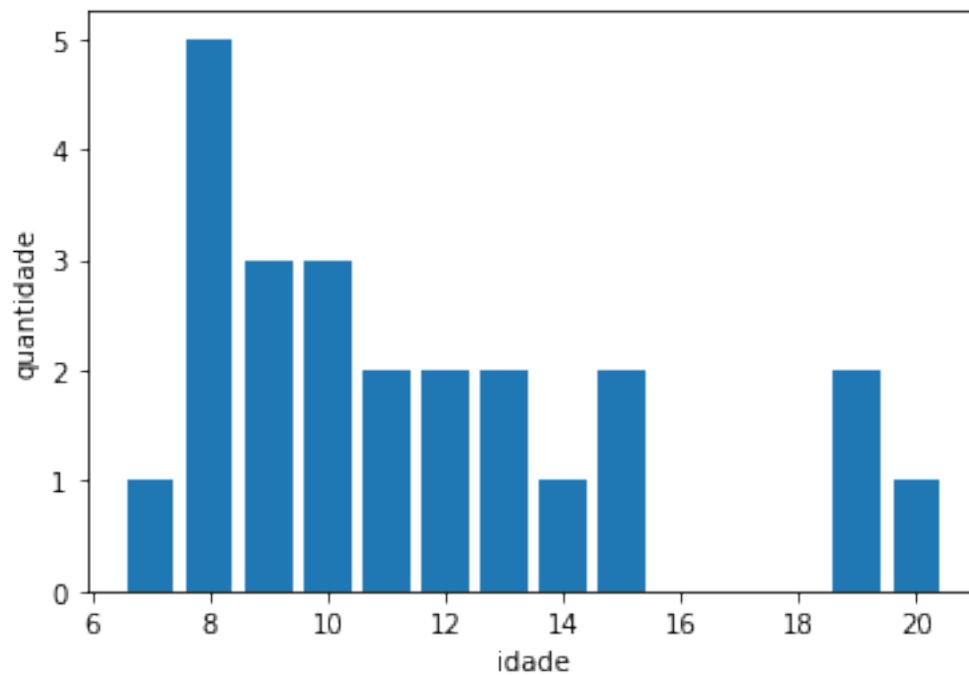
occur_quant = np.array(tamanho_alunos)
occur_quant_lista = list(occur_quant)

```

```

[347]: # montando o gráfico
plt.bar(occur_idade_lista, occur_quant_lista)
plt.xlabel('idade')
plt.ylabel('quantidade')
plt.show()

```



1.0.4 Tempo médio de deslocamento de acordo com o tipo de transporte

```

[348]: # definindo os dados
df_transporte = df_alunos.iloc[:, [6,8]]

```

```

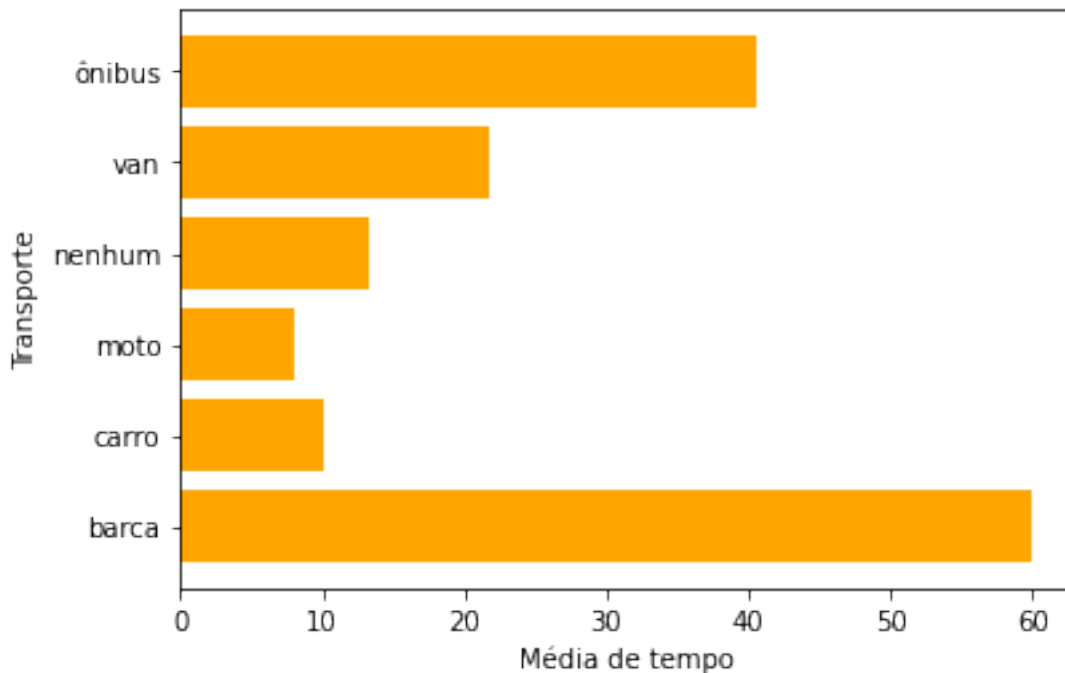
[349]: # definindo os dados
grup_transp = df_transporte.groupby('transporte').mean()

grup_transp_lista = list(grup_transp.iloc[:,0])

grup_transp_media_lista = list(np.array(grup_transp.index))

```

```
[350]: # montando o gráfico
plt.barh(grup_transp_media_lista, grup_transp_lista, color = 'orange')
plt.ylabel('Transporte')
plt.xlabel('Média de tempo')
plt.show()
```



2 Análise dos dados coletados dos responsáveis

2.0.1 A porcentagem de responsáveis que preferiram não se identificar

```
[351]: # definindo os dados
df_responsaveis_lista = list(df_responsaveis['nome'])
responsaveis_anonimos = df_responsaveis_lista.count('anônimo')

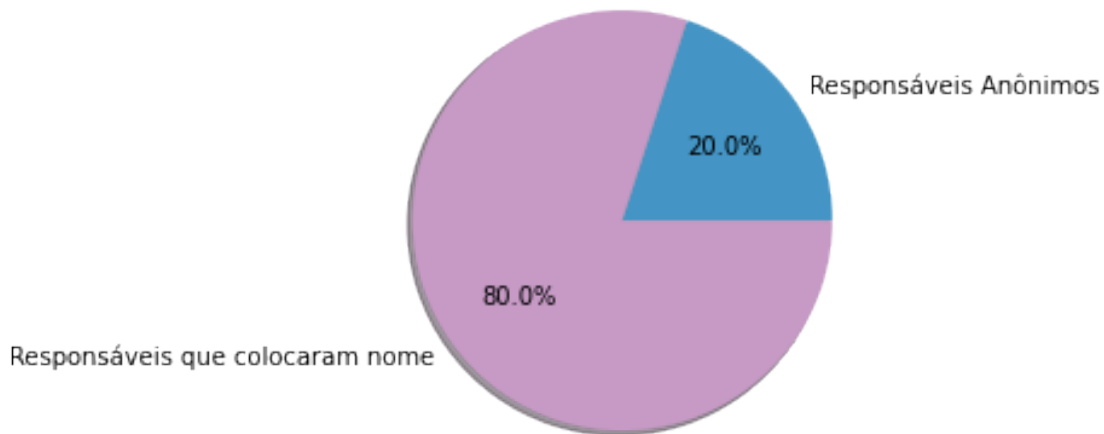
total_responsaveis = df_responsaveis['nome'].count()

responsaveis_com_nome = total_responsaveis - responsaveis_anonimos
```

```
[352]: # montando o gráfico
xa = [responsaveis_anonimos, responsaveis_com_nome]
mylabels = ["Responsáveis Anônimos", "Responsáveis que colocaram nome"]
mycolors = ['#4495c5', '#c69ac5']
```



```
plt.pie(xa, labels = mylabels, shadow = True, autopct = '%1.1f%%', colors = mycolors)
plt.show()
```



2.0.2 Responsáveis que participam das atividades escolares

```
[353]: # definindo os dados

grup_responsaveis_participa = df_responsaveis.groupby('participa_escola').size()

sempre_grup_responsaveis_participa = grup_responsaveis_participa['sempre']
algumas__grup_responsaveis_participa = grup_responsaveis_participa['algumas_
    vezes']
nunca_grup_responsaveis_participa = grup_responsaveis_participa['nunca']
frequentemente_grup_responsaveis_participa = grup_responsaveis_participa['frequentemente']
```

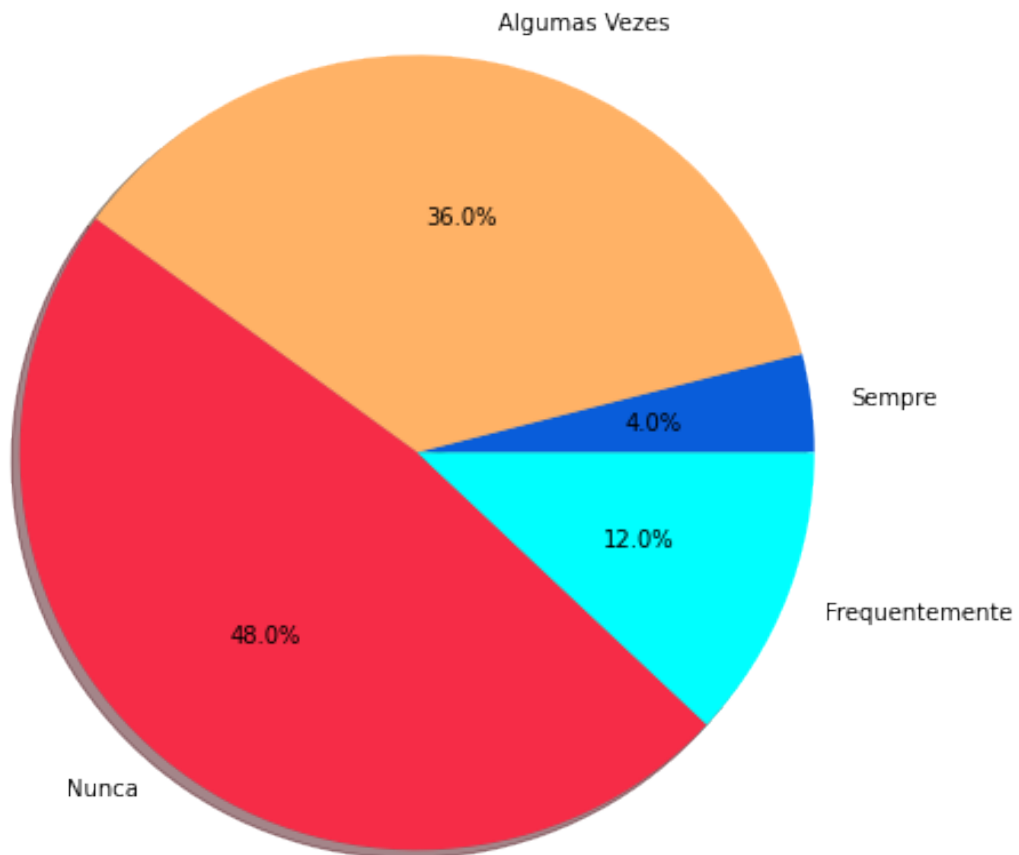
```
[354]: # montando o gráfico

xa = [sempre_grup_responsaveis_participa, algumas__grup_responsaveis_participa,
    nunca_grup_responsaveis_participa,
    frequentemente_grup_responsaveis_participa]
mylabels = ["Sempre", "Algumas Vezes", "Nunca", "Frequentemente"]
mycolors = ['#095dda', '#FFB266', '#f62c47', '#00FFFF' ]

plt.figure(figsize=(8,8))
plt.title('Pais que participam das atividades da escola')
```

```
plt.pie(xa, labels = mylabels, shadow = True, autopct = '%1.1f%%', colors = mycolors)
plt.show()
```

Pais que participam das atividades da escola



3 Análise dos dados coletados dos membros da comunidade

3.0.1 A porcentagem dos membros da comunidade que preferiram não se identificar

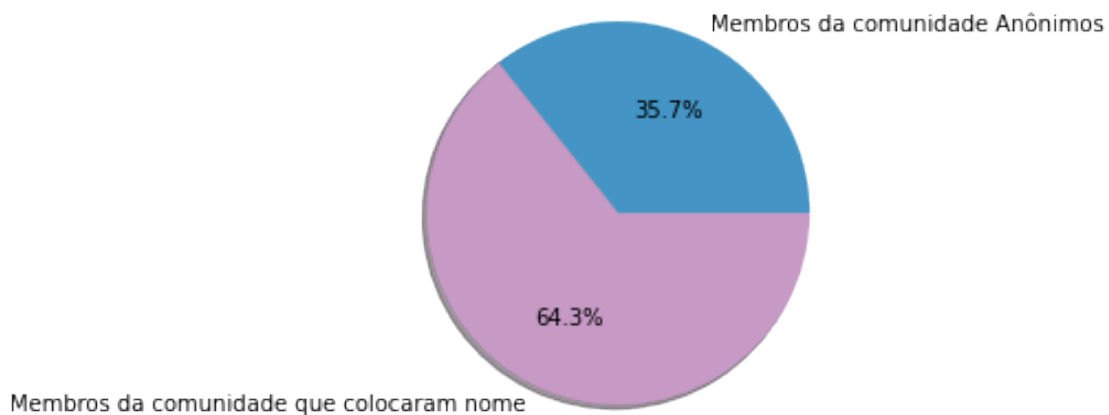
```
[355]: # definindo os dados
df_comunidade_lista = list(df_comunidade['nome'])
comunidade_anonimos = df_comunidade_lista.count('anônimo')
```

```
total_comunidade = df_comunidade['nome'].count()

comunidade_com_nome = total_comunidade - comunidade_anonimos
```

```
[356]: # montando o gráfico
xa = [comunidade_anonimos, comunidade_com_nome]
mylabels = ["Membros da comunidade Anônimos", "Membros da comunidade que
→colocaram nome"]
mycolors = ['#4495c5', '#c69ac5']

plt.pie(xa, labels = mylabels, shadow = True, autopct = '%1.1f%%', colors =
→mycolors)
plt.show()
```



3.0.2 Membros da comunidade que matriculariam o filho na escola

```
[357]: # definindo os dados
df_comunidade_matriculariam = df_comunidade.groupby('matricularia_filho').size()
df_comunidade_matriculariam_sim = df_comunidade_matriculariam['sim']
df_comunidade_matriculariam_nao = df_comunidade_matriculariam['não']
```

```
[358]: # montando o gráfico
y = [df_comunidade_matriculariam_sim, df_comunidade_matriculariam_nao]
x = ['Matriculariam', 'Não matriculariam']
plt.barh(x, y, color = '#66ff66')
plt.title("Matricularia um filho nessa escola?")
plt.show()
```



4 Análise em conjunto Alunos, responsáveis

4.0.1 Com relação a qualidade de ensino

```
[359]: # definindo os dados
grup_qualidade_alunos = df_alunos.groupby('qualidade_ens').size()
otima_qualidade_alunos = grup_qualidade_alunos['ótima']
boa_qualidade_alunos = grup_qualidade_alunos['boa']
regular_qualidade_alunos = grup_qualidade_alunos['regular']
ruim_qualidade_alunos = grup_qualidade_alunos['ruim']
pessima_qualidade_alunos = grup_qualidade_alunos['péssima']

grup_qualidade_responsaveis = df_responsaveis.groupby('qualidade_ens').size()
otima_qualidade_responsaveis = grup_qualidade_responsaveis['ótima']
boa_qualidade_responsaveis = grup_qualidade_responsaveis['boa']
regular_qualidade_responsaveis = grup_qualidade_responsaveis['regular']
```

```
[360]: # montando o gráfico
labels = ['Alunos', 'Responsáveis']
otima = [otima_qualidade_alunos, otima_qualidade_responsaveis]
boa = [boa_qualidade_alunos, boa_qualidade_responsaveis]
regular = [regular_qualidade_alunos, regular_qualidade_responsaveis]
ruim = [ruim_qualidade_alunos, ]
pessima = [pessima_qualidade_alunos, 0]
```

```

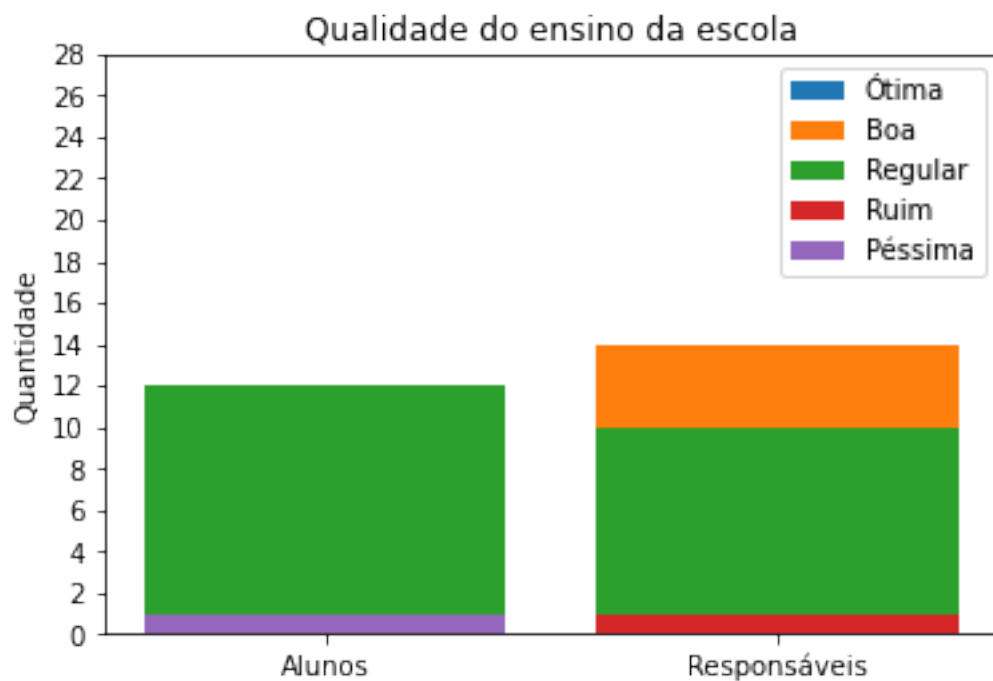
fig, ax = plt.subplots()

ax.bar(labels, otima, label='Ótima')
ax.bar(labels, boa, label='Boa')
ax.bar(labels, regular, label='Regular')
ax.bar(labels, ruim, label='Ruim')
ax.bar(labels, pessima, label='Péssima')

ax.set_ylabel('Quantidade')
ax.set_title('Qualidade do ensino da escola')
plt.yticks(range(0,30,2))
plt.figure(figsize=(8,8))
ax.legend()

plt.show()

```



<Figure size 576x576 with 0 Axes>

5 Análise em conjunto Alunos, responsáveis e comunidade

5.0.1 Gráfico que mostra a relação da escola com a comunidade

```
[361]: # definindo os dados
grup_ajuda_alunos = df_alunos.groupby('ajuda_comunidade').size()
sim_ajuda_alunos = grup_ajuda_alunos['sim']
nao_ajuda_alunos = grup_ajuda_alunos['não']

grup_ajuda_responsaveis = df_responsaveis.groupby('ajuda_comunidade').size()
sim_ajuda_responsaveis = grup_ajuda_responsaveis['sim']
nao_ajuda_responsaveis = grup_ajuda_responsaveis['não']

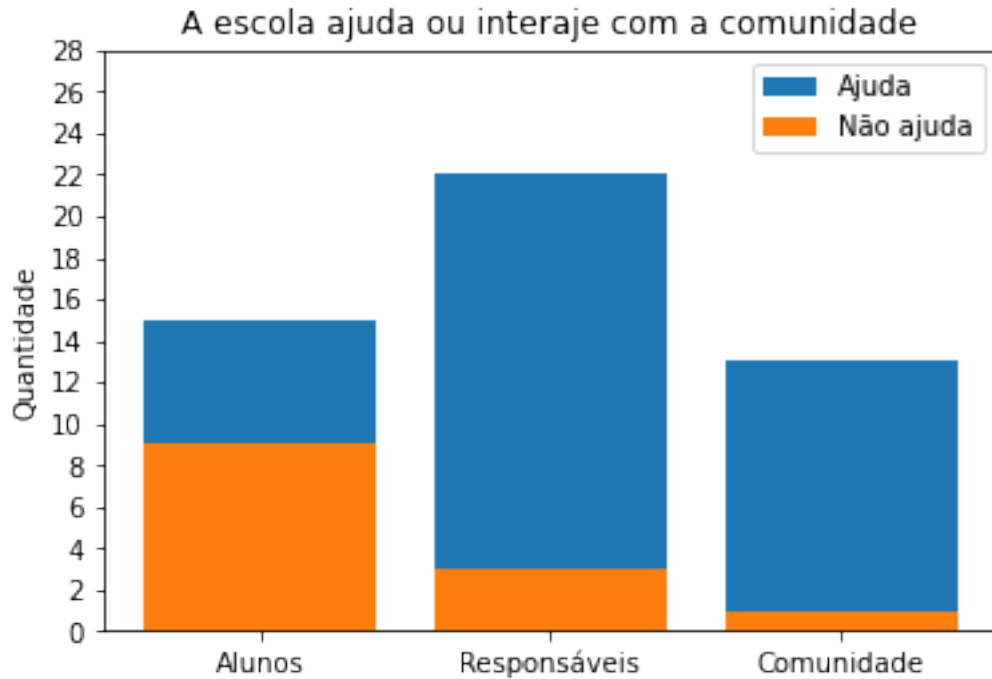
grup_ajuda_comunidade = df_comunidade.groupby('ajuda_comunidade').size()
sim_ajuda_comunidade = grup_ajuda_comunidade['sim']
nao_ajuda_comunidade = grup_ajuda_comunidade['não']

[362]: # montando o gráfico
labels = ['Alunos', 'Responsáveis', 'Comunidade']
ajuda_com = [sim_ajuda_alunos, sim_ajuda_responsaveis, sim_ajuda_comunidade]
nao_ajuda_com = [nao_ajuda_alunos, nao_ajuda_responsaveis, nao_ajuda_comunidade]

fig, ax = plt.subplots()

ax.bar(labels, ajuda_com, label='Ajuda')
ax.bar(labels, nao_ajuda_com, label='Não ajuda')
ax.set_ylabel('Quantidade')
ax.set_title('A escola ajuda ou interaje com a comunidade')
plt.yticks(range(0,30,2))
ax.legend()

plt.show()
```



```
[363]: !wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('analise_de_dados_oae.ipynb')
```

File 'colab_pdf.py' already there; not retrieving.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/Colab
Notebooks/analise_de_dados_oae.ipynb to pdf
[NbConvertApp] Writing 53207 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', './notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', './notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 56073 bytes to /content/drive/My
Drive/analise_de_dados_oae.pdf
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

```
[363]: 'File ready to be Downloaded and Saved to Drive'
```