

Análise por Km - Transmodall

May 10, 2023

1 Importação das bibliotecas

```
[78]: """
1º) Importação do pandas como pd para trabalhar com dados.
"""
import pandas as pd
"""
2º) Importação do numpy como np para trabalhar com matrizes
"""
import numpy as np
"""
3º) Importação do matplotlib.pyplot como plt para fazer gráficos.
"""
import matplotlib.pyplot as plt
"""
4º) De matplotlib.ticker vamos importar o AutoMinorLocator e o MaxNLocator para
    ↪trabalhar com os "ticks"
    dos gráficos.
"""
from matplotlib.ticker import AutoMinorLocator, MaxNLocator
from matplotlib.font_manager import FontProperties
"""
5º) Importação do seaborn e do plotly.express para gráficos
"""
import seaborn as sbn
import plotly.express as px
"""
6º) Análise exploratória otimizada
"""
from pandas_profiling import ProfileReport
```

2 Importação dos dados

```
[79]: Dados = pd.read_csv("TRANSMODALL_CE_PLAN1 - Plan1.csv")
Dados.head()
```

```
[79]:
```

	CIDADE	DISTÂNCIA	VALOR	GASTO	VALOR/IDA	VALOR/KM
0	ACARAU - CE	207.0	3500	2401.2	16.908213	8.454106
1	ACOPIARA - CE	411.0	5800	4603.2	14.111922	7.055961
2	AÇU - RN	372.0	5500	4166.4	14.784946	7.392473
3	ALEXANDRIA - RN	437.0	5500	4894.4	12.585812	6.292906
4	ALMIRO AFONSO - RN	534.0	5500	5980.8	10.299625	5.149813

3 Pré-processamento de dados

```
[80]: Dados.dtypes
```

```
[80]: CIDADE      object
DISTÂNCIA    float64
VALOR        int64
GASTO        float64
VALOR/IDA    float64
VALOR/KM     float64
dtype: object
```

```
[81]: Dados_faltantes = 100*(Dados.isnull().sum()/len(Dados["DISTÂNCIA"]))
Dados_faltantes
```

```
[81]: CIDADE      0.0
DISTÂNCIA    0.0
VALOR        0.0
GASTO        0.0
VALOR/IDA    0.0
VALOR/KM     0.0
dtype: float64
```

```
[82]: Dados.columns
```

```
[82]: Index(['CIDADE ', 'DISTÂNCIA', 'VALOR ', 'GASTO', 'VALOR/IDA', 'VALOR/KM'],
dtype='object')
```

```
[83]: Fretes_abaixo_do_gasto = Dados.loc[Dados["VALOR "] < Dados["GASTO"]]
Frete_abaixo_do_gasto
```

```
[83]:
```

	CIDADE	DISTÂNCIA	VALOR	GASTO	VALOR/IDA	VALOR/KM
4	ALMIRO AFONSO - RN	534.0	5500	5980.8	10.299625	5.149813
7	ANTONIO MARTINS - RN	520.0	5500	5824.0	10.576923	5.288462
8	APODI - RN	450.0	4700	5040.0	10.444444	5.222222
36	ÊRERE -CE	446.0	4600	4995.2	10.313901	5.156951

```
[84]: """
Criação das fontes de texto
"""
```

```

Fonte1 = {"family": "serif", #Familia da fonte
          "weight": "bold", #Peso da fonte
          "color": "black", #cor da fonte
          "size": 12.4} #Tamanho da fonte
font2 = FontProperties(family = "Serif", weight = "bold", style = "normal",
    ↪size = 11)
Fonte3 = {"family": "serif", #Familia da fonte
          "weight": "bold", #Peso da fonte
          "color": "black", #cor da fonte
          "size": 20.4}

"""
Alocando a figura
"""
fig, ax = plt.subplots(figsize = (9, 7))
"""
Plot do gráfico
"""
plt.scatter(2*Dados["DISTÂNCIA"], Dados["VALOR "], color = "purple", alpha = 0.
    ↪5)
plt.plot(2*Dados["DISTÂNCIA"], Dados["GASTO"], label = "Função de gasto bruto",
    ↪color = "red", linewidth = 0.15)
plt.scatter(2*Frete_abixo_do_gasto["DISTÂNCIA"],
    ↪Frete_abixo_do_gasto["VALOR "], color = "darkred", linewidths = 3.1, label
    ↪= "\nALMIRO AFONSO - RN\nANTONIO MARTINS - RN\nAPODI - RN\nÊRERE - CE")
plt.grid(False)
"""
Redefinição da grossura dos eixos e da cor dos mesmos
"""
for axis in ["left", "right", "top", "bottom"]:
    ax.spines[axis].set_linewidth(2)
    ax.spines[axis].set_color("black")
"""
Trabalha com os ticks do gráfico
"""
ax.xaxis.set_minor_locator(AutoMinorLocator())
ax.yaxis.set_minor_locator(AutoMinorLocator())
ax.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsiz
    ↪e = 12.4)
ax.tick_params(which = "minor", direction = "in", width = 2, color = "black")
ax.tick_params(which = "major", direction = "in", color = "black", length=3.4,
    ↪width = 2)
"""
Labels
"""
ax.set_ylabel("Valor do frete (R$)", fontdict = Fonte1)
ax.set_xlabel("Total percorrido (Km)", fontdict = Fonte1)

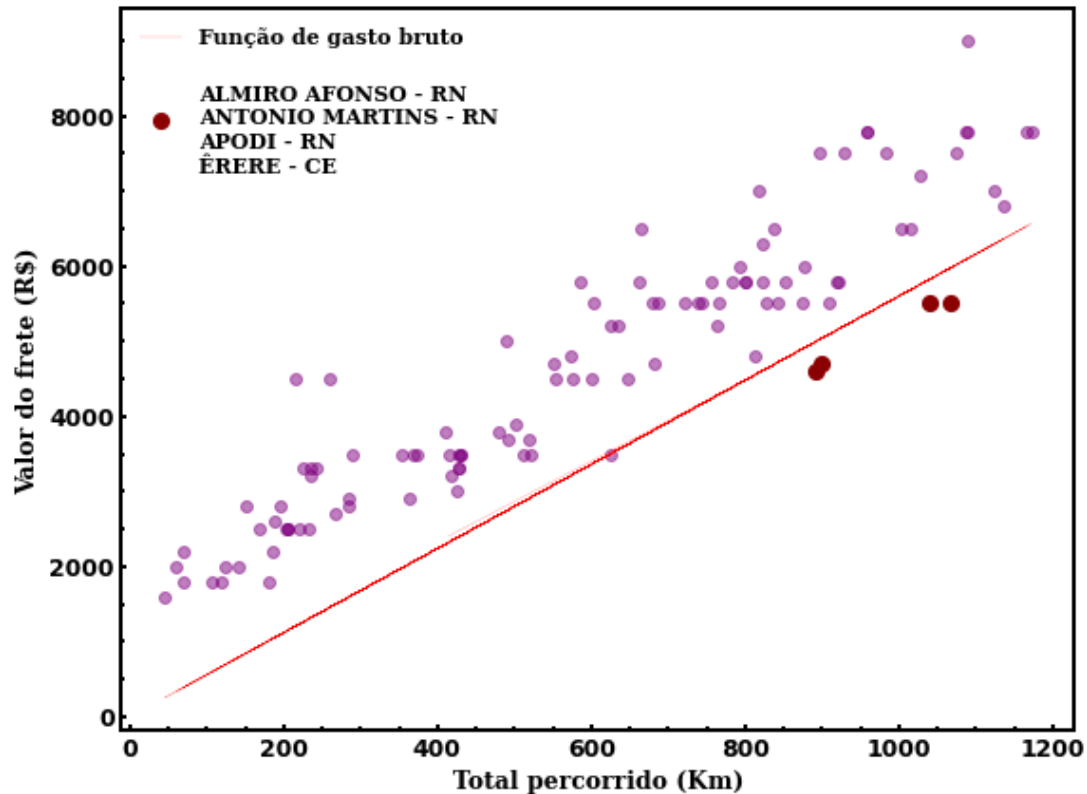
```

```

"""
Tudo em negrito
"""
plt.rcParams["font.weight"] = "bold"
plt.rcParams["axes.labelweight"] = "bold"
"""
Fundo branco
"""
fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
fig.patch.set_facecolor("white")
"""
Mostrar gráfico
"""
plt.legend(frameon = False, prop = font2)
plt.text(1280, 10200, "3 cidades do Rio Grande do Norte são prejuízos",
↪rotation=0, fontdict = Fonte3,
        ha="right", va="top",
        bbox=dict(boxstyle="square",
                  ec=(1, 1, 1),
                  fc=(1, 1, 1),
                  ))
plt.show()

```

3 cidades do Rio Grande do Norte são prejuízos



```
[85]: Dados["GASTO"]
```

```
[85]: 0      2401.20
      1      4603.20
      2      4166.40
      3      4894.40
      4      5980.80
      ...
     103     4480.00
     104     1093.12
     105     5364.80
     106     3707.20
     107     3281.60
      Name: GASTO, Length: 108, dtype: float64
```

```
[86]: Dados["GASTO"] + Dados["VALOR "]*0.25
```

```
[86]: 0      3276.20
      1      6053.20
      2      5541.40
```

```

3      6269.40
4      7355.80
...
103    5930.00
104    1793.12
105    7314.80
106    5157.20
107    4731.60
Length: 108, dtype: float64

```

```
[ ]:
```

```

[87]: Fretes_abaixo_do_esperado = Dados.loc[Dados["VALOR "] < Dados["GASTO"] +
↳Dados["VALOR "]*0.25]
Fretes_abaixo_do_esperado

```

```

[87]:
          CIDADE  DISTÂNCIA  VALOR  GASTO  VALOR/IDA  VALOR/KM
1      ACOPIARA - CE      411.0   5800  4603.2  14.111922  7.055961
2          AÇU - RN      372.0   5500  4166.4  14.784946  7.392473
3      ALEXANDRIA - RN      437.0   5500  4894.4  12.585812  6.292906
4      ALMIRO AFONSO - RN      534.0   5500  5980.8  10.299625  5.149813
5      ALTO DO RODRIGUES - RN      383.0   5500  4289.6  14.360313  7.180157
7      ANTONIO MARTINS - RN      520.0   5500  5824.0  10.576923  5.288462
8          APODI - RN      450.0   4700  5040.0  10.444444  5.222222
12      AREIA BRANCA - RN      341.0   4700  3819.2  13.782991  6.891496
14      BARBALHA-CE      583.0   7800  6529.6  13.379074  6.689537
20      BREJO SANTO - CE      545.0   7800  6104.0  14.311927  7.155963
23      CARIÚS - CE      501.0   6500  5611.2  12.974052  6.487026
28      CHORO - CE      212.0   3000  2374.4  14.150943  7.075472
31      CRATO - CE      586.0   7800  6563.2  13.310580  6.655290
34      DR SEVERINO - RN      461.0   5800  5163.2  12.581345  6.290672
35      EQUADOR-RN      562.0   7000  6294.4  12.455516  6.227758
36      ÊRERE -CE      446.0   4600  4995.2  10.313901  5.156951
40      GOV SEXT ROSADO -RN      323.0   4500  3617.6  13.931889  6.965944
42      IBICUITINGA -CE      256.0   3500  2867.2  13.671875  6.835938
44      ICÓ -CE      406.0   4800  4547.2  11.822660  5.911330
46      IPANGUAÇU - RN      382.0   5200  4278.4  13.612565  6.806283
47      IPAUMIRIM -CE      459.0   5800  5140.8  12.636166  6.318083
48      ITAJÁ - RN      369.0   5500  4132.8  14.905149  7.452575
54      JOÃO CAMARA - RN      514.0   7200  5756.8  14.007782  7.003891
56      JUAZEIRO DO NORTE -CE      543.0   7800  6081.6  14.364641  7.182320
59      LIOMOEIRO DO NORTE -CE      261.0   3500  2923.2  13.409962  6.704981
62      MARCELINO VIEIRA - RN      438.0   6000  4905.6  13.698630  6.849315
64      MARTINS - RN      414.0   5500  4636.8  13.285024  6.642512
65      MAURITI -CE      537.0   7500  6014.4  13.966480  6.983240
76      PATOS - PB      568.0   6800  6361.6  11.971831  5.985915
77      PAU DOS FERROS -RN      400.0   5800  4480.0  14.500000  7.250000

```

80	PILÕES - RN	426.0	5800	4771.2	13.615023	6.807512
82	PORTALEGRE - RN	391.0	5800	4379.2	14.833760	7.416880
84	QUIXERAMOBIM -CE	259.0	3700	2900.8	14.285714	7.142857
86	RAFAEL FERNANDES - RN	421.0	5500	4715.2	13.064133	6.532067
88	RERIUTABA	312.0	3500	3494.4	11.217949	5.608974
90	SANTANA DO MATO - RN	455.0	5500	5096.0	12.087912	6.043956
91	SÃO JOSE DE PIRANHAS-PB	508.0	6500	5689.6	12.795276	6.397638
103	UMARIZAL - RN	400.0	5800	4480.0	14.500000	7.250000

```
[88]: Fretes_abaixo_do_esperado.shape
```

```
[88]: (38, 6)
```

```
[89]: """
Criação das fontes de texto
"""
Fonte1 = {"family": "serif", #Familia da fonte
          "weight": "bold", #Peso da fonte
          "color": "black", #cor da fonte
          "size": 12.4} #Tamanho da fonte
font2 = FontProperties(family = "Serif", weight = "bold", style = "normal",
                      ↪size = 12.4)
Fonte3 = {"family": "serif", #Familia da fonte
          "weight": "bold", #Peso da fonte
          "color": "black", #cor da fonte
          "size": 20.4}

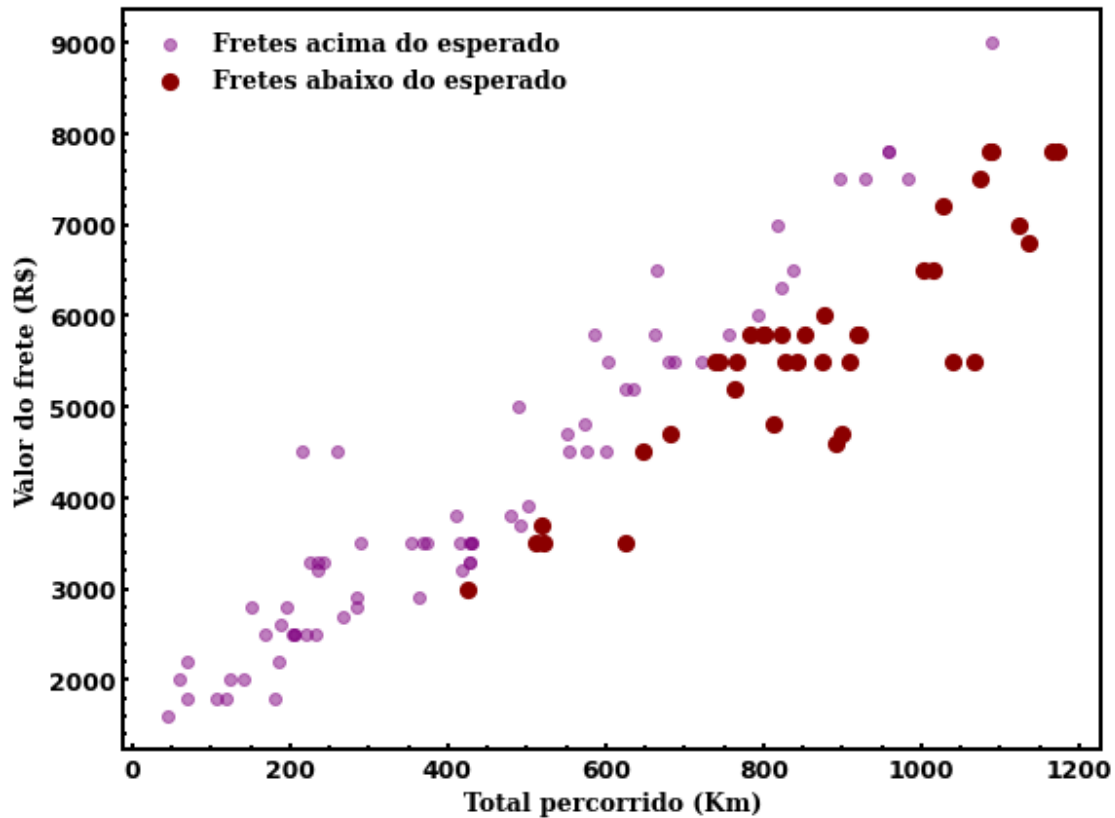
"""
Alocando a figura
"""
fig, ax = plt.subplots(figsize = (9, 7))
"""
Plot do gráfico
"""
plt.scatter(2*Dados["DISTÂNCIA"], Dados["VALOR "], color = "purple", alpha = 0.
           ↪5, label = "Fretes acima do esperado")
#plt.plot(2*Dados["DISTÂNCIA"], Dados["GASTO"] + Dados["VALOR "]*0.25, label =
           ↪"Função de gasto bruto", color = "black", linewidth = 2)
plt.scatter(2*Fretes_abaixo_do_esperado["DISTÂNCIA"],
           ↪Fretes_abaixo_do_esperado["VALOR "], color = "darkred", linewidths = 3.1,
           ↪label = "Fretes abaixo do esperado")
plt.grid(False)
"""
Redefinição da grossura dos eixos e da cor dos mesmos
"""
for axis in ["left", "right", "top", "bottom"]:
    ax.spines[axis].set_linewidth(2)
    ax.spines[axis].set_color("black")
```

```

"""
Trabalha com os ticks do gráfico
"""
ax.xaxis.set_minor_locator(AutoMinorLocator())
ax.yaxis.set_minor_locator(AutoMinorLocator())
ax.tick_params(axis = "both", direction = "in", labelcolor = "black", labelsiz
↳= 12.4)
ax.tick_params(which = "minor", direction = "in", width = 2, color = "black")
ax.tick_params(which = "major", direction = "in", color = "black", length=3.4,
↳width = 2)
"""
Labels
"""
ax.set_ylabel("Valor do frete (R$)", fontdict = Fonte1)
ax.set_xlabel("Total percorrido (Km)", fontdict = Fonte1)
"""
Tudo em negrito
"""
plt.rcParams["font.weight"] = "bold"
plt.rcParams["axes.labelweight"] = "bold"
"""
Fundo branco
"""
fig.patch.set_facecolor("white")
Cor_fundo = plt.gca()
Cor_fundo.set_facecolor("white")
Cor_fundo.patch.set_alpha(1)
fig.patch.set_facecolor("white")
"""
Mostrar gráfico
"""
plt.legend(frameon = False, prop = font2)
plt.text(1200, 10000, "38 cidades estão abaixo do valor esperado", rotation=0,
↳fontdict = Fonte3,
        ha="right", va="top",
        bbox=dict(boxstyle="square",
                  ec=(1, 1, 1),
                  fc=(1, 1, 1),
                  ))
plt.show()

```


38 cidades estão abaixo do valor esperado



[90]: `Dados.shape`

[90]: (108, 6)

```
[91]: Lista_numbers = []
for i in range(0, 71):
    Lista_numbers.append("Acima do esperado")
for i in range(0, 39):
    Lista_numbers.append("Abaixo do esperado")
Lista_numbers = pd.DataFrame(Lista_numbers)
"""
Criação da primeira fonte por meio de um dicionário;
Precisamos definir um dicionário passando 4 parâmetros;
family, weight, color e size
"""
font1 = {"family": "serif", "weight": "bold", "color": "black", "size": 15}
"""
Criação da segunda fonte por meio da FontProperties()
Também passamos alguns parâmetros, como family, weight, style e size.
```

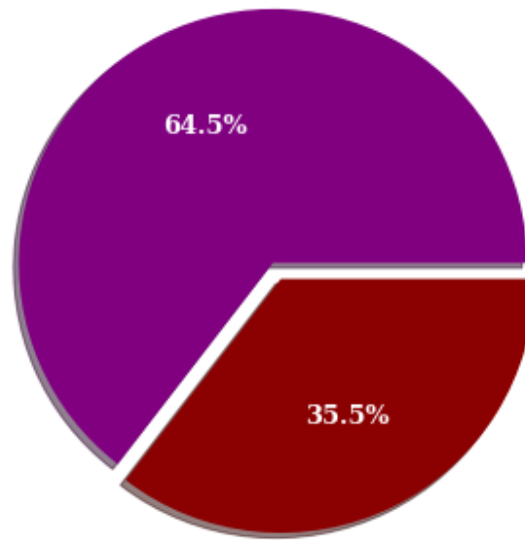
```

"""
font2 = FontProperties(family = "Serif", weight = "bold", style = "normal",
    ↳size = 15)
"""
Definição das cores do gráfico de pizza.
"""
colors_pie = ["purple", "darkred"]
"""
Criando um "local" para alocar a nossa figura
"""
fig, ax = plt.subplots(figsize = (9, 6))
"""
Plot do gráfico do tipo "pie".
"""
Lista_numbers[0].value_counts().plot(kind = "pie", # Tipo de gráfico
    autopct = "%1.1f%", # Mostrar
    ↳porcentagens no gráfico
    shadow = True, # Sombreamento
    explode = [0.02, 0.05], # Separamento
    ↳entre as partes
    textprops = {"family": "serif", #
    ↳Definição da fonte para os textos
                "weight": "bold",
                "color": "white",
                "fontsize": 13},
    colors = colors_pie) # cores do gráfico
"""
Definição de um fundo branco.
"""
fig.patch.set_facecolor("white")
"""
Definindo o título do gráfico
"""
ax.set_title("Acima x Abaixo do esperado", fontdict = font1)
"""
Propriedades da legenda da figura
"""
ax.legend(frameon = False, prop = font2, labelcolor = "black", bbox_to_anchor =
    ↳[0.1, 1])
"""
Definição do label do eixo y
"""
ax.set_ylabel("")
"""
Mostrar gráfico
"""
plt.show()

```

Acima x Abaixo do esperado

■ Acima do esperado
■ Abaixo do esperado



```
[92]: Lista_de_estados = []  
for c in Dados["CIDADE "]:  
    if "RN" in c:  
        Lista_de_estados.append("RN")  
    if "CE" in c:  
        Lista_de_estados.append("CE")  
    if "PB" in c:  
        Lista_de_estados.append("PB")  
Lista_de_estados
```

```
[92]: ['CE',  
      'CE',  
      'RN',  
      'RN',  
      'RN',  
      'RN',  
      'RN',  
      'CE',  
      'RN',  
      'RN',  
      'CE',  
      'CE',  
      'CE',  
      'RN',  
      'RN',  
      'CE',
```

'CE',
'CE',
'CE',
'CE',
'CE',
'PB',
'CE',
'CE',
'RN',
'CE',
'CE',
'RN',
'CE',
'CE',
'CE',
'CE',
'CE',
'CE',
'CE',
'CE',
'RN',
'RN',
'CE',
'CE',
'CE',
'CE',
'RN',
'CE',
'CE',
'CE',
'CE',
'CE',
'RN',
'CE',
'RN',
'CE',
'RN',
'CE',
'CE',
'CE',
'CE',
'RN',
'RN',
'CE',
'RN',
'CE',
'CE',
'CE',
'CE',

'RN',
'CE',
'CE',
'RN',
'CE',
'CE',
'CE',
'RN',
'CE',
'CE',
'CE',
'CE',
'CE',
'RN',
'PB',
'RN',
'CE',
'CE',
'RN',
'CE',
'RN',
'CE',
'CE',
'CE',
'RN',
'CE',
'CE',
'RN',
'PB',
'CE',
'CE',
'RN',
'RN',
'CE',
'RN',
'CE',
'CE',
'CE',
'CE',
'CE',
'RN',
'CE',
'CE',
'CE',
'CE']

```
[93]: Lista_de_estados = pd.DataFrame(Lista_de_estados)
      Lista_de_estados
```

```
[93]:      0
0     CE
1     CE
2     RN
3     RN
4     RN
..    ..
103   RN
104   CE
105   CE
106   CE
107   CE

[108 rows x 1 columns]
```

```
[94]: """
      Criação da primeira fonte por meio de um dicionário;
      Precisamos definir um dicionário passando 4 parâmetros;
      family, weight, color e size
      """

      font1 = {"family": "serif", "weight": "bold", "color": "black", "size":15}
      """

      Criação da segunda fonte por meio da FontProperties()
      Também passamos alguns parâmetros, como family, weight, style e size.
      """

      font2 = FontProperties(family = "Serif", weight = "bold", style = "normal",
      ↪size = 15)
      """

      Definição das cores do gráfico de pizza.
      """

      colors_pie = ["purple", "darkred", "red"]
      """

      Criando um "local" para alocar a nossa figura
      """

      fig, ax = plt.subplots(figsize = (9, 6))
      """

      Plot do gráfico do tipo "pie".
      """

      Lista_de_estados[0].value_counts().plot(kind = "pie", # Tipo de gráfico
      ↪autopct = "%1.1f%%", # Mostrar
      ↪porcentagens no gráfico

      ↪shadow = True, # Sombreamento
      ↪explode = [0.02, 0.05, 0.10], #
      ↪Separamento entre as partes
```

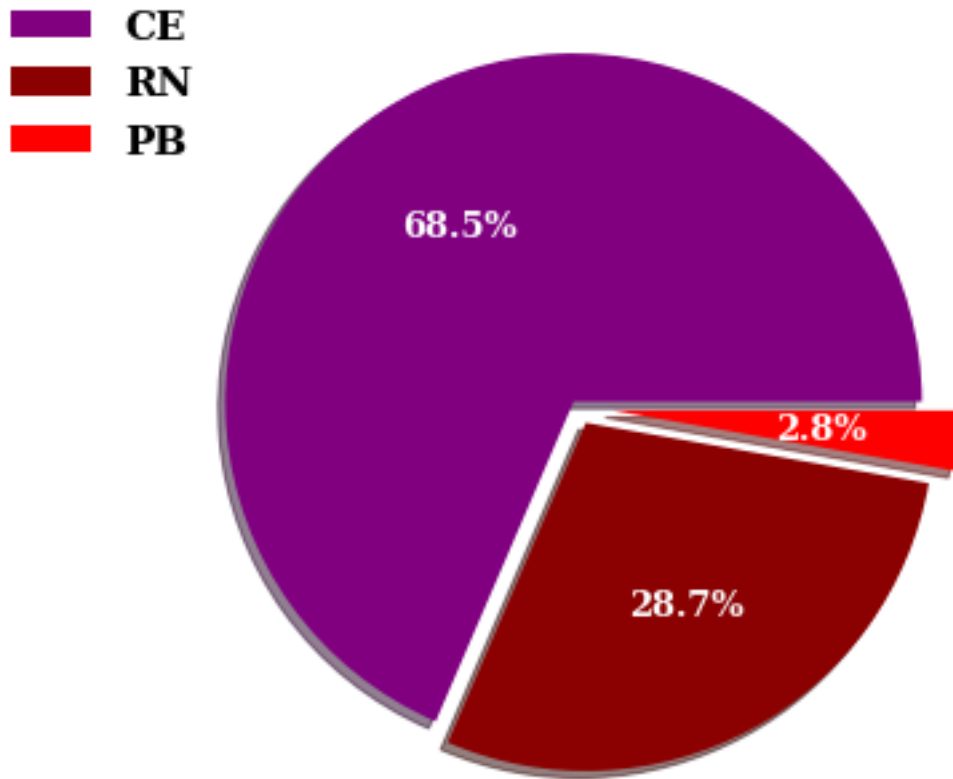
```

textprops = {"family": "serif", #
↳Definição da fonte para os textos
               "weight": "bold",
               "color": "white",
               "fontsize":13},
colors = colors_pie) # cores do gráfico

"""
Definição de um fundo branco.
"""
fig.patch.set_facecolor("white")
"""
Definindo o título do gráfico
"""
ax.set_title("Estados abaixo do esperado", fontdict = font1)
"""
Propriedades da legenda da figura
"""
ax.legend(frameon = False, prop = font2, labelcolor = "black", bbox_to_anchor =
↳[0.1, 1])
"""
Definição do label do eixo y
"""
ax.set_ylabel("")
"""
Mostrar gráfico
"""
plt.show()

```

Estados abaixo do esperado



```
[95]: Fretes_acima_do_esperado = Dados.loc[Dados["GASTO"] + Dados["VALOR "]*0.25 <=
↳Dados["VALOR "]]
Fretes_acima_do_esperado
```

```
[95]:
```

	CIDADE	DISTÂNCIA	VALOR	GASTO	VALOR/IDA	VALOR/KM
0	ACARAU - CE	207.0	3500	2401.20	16.908213	8.454106
6	AMONTADA - CE	121.0	3300	1355.20	27.272727	13.636364
9	AQUIRAZ - CE	90.7	1800	1015.84	19.845645	9.922822
10	ARACATI -CE	209.0	3200	2340.80	15.311005	7.655502
11	ARACOIBA - CE	133.0	2700	1489.60	20.300752	10.150376
..
102	TURURU -CE	93.6	2600	1048.32	27.777778	13.888889
104	URUBURETAMA - CE	97.6	2800	1093.12	28.688525	14.344262
105	VARZEA ALEGRE - CE	479.0	7800	5364.80	16.283925	8.141962
106	VIÇOSA-CE	331.0	5800	3707.20	17.522659	8.761329
107	UBAJARA-CE	293.0	5800	3281.60	19.795222	9.897611

[70 rows x 6 columns]

```
[96]: Lista_de_cidades_acima_do_esperado = []
for c in Dados["CIDADE "]:
    if "CE" in c:
        Lista_de_cidades_acima_do_esperado.append("CE")
    if "RN" in c:
        Lista_de_cidades_acima_do_esperado.append("RN")
    if "PB" in c:
        Lista_de_cidades_acima_do_esperado.append("PB")
    if "PI" in c:
        Lista_de_cidades_acima_do_esperado.append("PI")
Lista_de_cidades_acima_do_esperado = pd.
↳DataFrame(Lista_de_cidades_acima_do_esperado)
```

```
[97]: """
Criação da primeira fonte por meio de um dicionário;
Precisamos definir um dicionário passando 4 parâmetros;
family, weight, color e size
"""

font1 = {"family": "serif", "weight": "bold", "color": "black", "size":15}
"""

Criação da segunda fonte por meio da FontProperties()
Também passamos alguns parâmetros, como family, weight, style e size.
"""

font2 = FontProperties(family = "Serif", weight = "bold", style = "normal",
↳size = 15)
"""

Definição das cores do gráfico de pizza.
"""

colors_pie = ["purple", "darkred", "red", "gray"]
"""

Criando um "local" para alocar a nossa figura
"""

fig, ax = plt.subplots(figsize = (9, 6))
"""

Plot do gráfico do tipo "pie".
"""

Lista_de_cidades_acima_do_esperado[0].value_counts().plot(kind = "pie", # Tipo
↳de gráfico

                                autopct = "%1.1f%%", # Mostrar
↳porcentagens no gráfico

                                shadow = True, # Sombreamento
                                explode = [0.02, 0.05, 0.10, 0.10], #
↳Separamento entre as partes

                                textprops = {"family": "serif", #
↳Definição da fonte para os textos
```

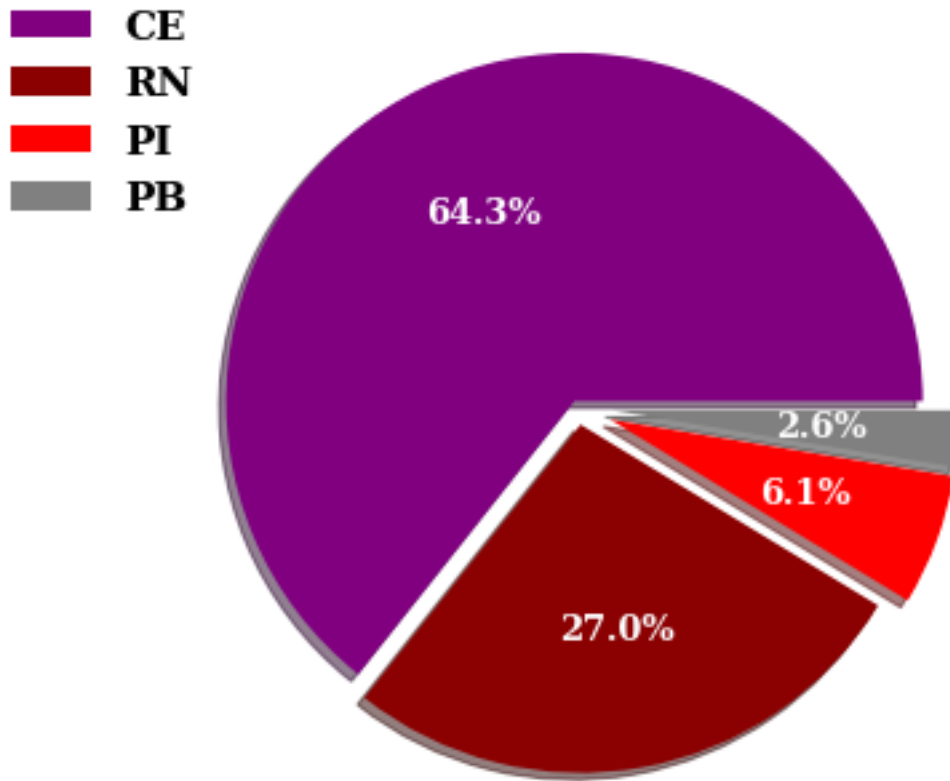
```

        "weight": "bold",
        "color": "white",
        "fontsize":13},
    colors = colors_pie) # cores do gráfico

"""
Definição de um fundo branco.
"""
fig.patch.set_facecolor("white")
"""
Definindo o título do gráfico
"""
ax.set_title("Estados acima do esperado", fontdict = font1)
"""
Propriedades da legenda da figura
"""
ax.legend(frameon = False, prop = font2, labelcolor = "black", bbox_to_anchor = ↵
↵[0.1, 1])
"""
Definição do label do eixo y
"""
ax.set_ylabel("")
"""
Mostrar gráfico
"""
plt.show()

```

Estados acima do esperado



```
[98]: # 5 MELHORES FRETES
Fretes_super_faturado = Dados.loc[Dados["GASTO"] + Dados["VALOR "]*0.5 <=
↪Dados["VALOR "]]
Fretes_super_faturado
```

```
[98]:
```

	CIDADE	DISTÂNCIA	VALOR	GASTO	VALOR/IDA \
6	AMONTADA - CE	121.0	3300	1355.20	27.272727
15	BATURITE-CE	144.0	3500	1612.80	24.305556
19	BOA VISTA DISTRITO(ITAPIPOCA)	118.0	3200	1321.60	27.118644
25	CASCÁVEL -CE	110.0	2500	1232.00	22.727273
26	CAUCAIA -CE	34.6	1800	387.52	52.023121
37	EUSEBIO -CE	70.7	2000	791.84	28.288543
38	FORTALEZA -CE	53.5	1800	599.20	33.644860
41	HORIZONTE - CE	92.3	2200	1033.76	23.835320
49	ITAPAJÉ - CE	112.0	3300	1254.40	29.464286
51	ITAPIPOCA - CE	117.0	3300	1310.40	28.205128
60	MARACANAU -CE	59.4	1800	665.28	30.303030

61	MARANGUAPE - CE	62.3	2000	697.76	32.102729
69	PACAJUS -CE	103.0	2500	1153.60	24.271845
70	PACOTI -CE	130.0	4500	1456.00	34.615385
72	PALMACIA - CE	107.0	4500	1198.40	42.056075
73	PARACURU -CE	30.3	2000	339.36	66.006601
74	PARAIPABA -CE	34.5	2200	386.40	63.768116
79	PENTECOSTE - CE	75.4	2800	844.48	37.135279
81	PINDORETAMA - CE	101.0	2500	1131.20	24.752475
87	REDENÇÃO -CE	102.0	2500	1142.40	24.509804
93	SÃO GONÇALO DO AMARANTE -CE	22.4	1600	250.88	71.428571
100	TRAIRI-CE	83.8	2500	938.56	29.832936
102	TURURU -CE	93.6	2600	1048.32	27.777778
104	URUBURETAMA - CE	97.6	2800	1093.12	28.688525

	VALOR/KM
6	13.636364
15	12.152778
19	13.559322
25	11.363636
26	26.011561
37	14.144272
38	16.822430
41	11.917660
49	14.732143
51	14.102564
60	15.151515
61	16.051364
69	12.135922
70	17.307692
72	21.028037
73	33.003300
74	31.884058
79	18.567639
81	12.376238
87	12.254902
93	35.714286
100	14.916468
102	13.888889
104	14.344262

```
[99]: Fretes_super_faturado.describe()
# Melhor Frete - São Gonçalo do Amarante
```

```
[99]:
```

	DISTÂNCIA	VALOR	GASTO	VALOR/IDA	VALOR/KM
count	24.000000	24.000000	24.000000	24.000000	24.000000
mean	86.475000	2654.166667	968.520000	34.755609	17.377804
std	33.826954	786.836907	378.861886	14.101132	7.050566

min	22.400000	1600.000000	250.880000	22.727273	11.363636
25%	61.575000	2000.000000	689.640000	26.527102	13.263551
50%	95.600000	2500.000000	1070.720000	29.076405	14.538203
75%	110.500000	3225.000000	1237.600000	35.245358	17.622679
max	144.000000	4500.000000	1612.800000	71.428571	35.714286

```
[100]: Dados.describe()
```

```
[100]:
```

	DISTÂNCIA	VALOR	GASTO	VALOR/IDA	VALOR/KM
count	108.000000	108.000000	108.000000	108.000000	108.000000
mean	294.121296	4634.259259	3294.925185	19.625236	9.812618
std	158.183183	1778.330732	1771.243317	10.662557	5.331279
min	22.400000	1600.000000	250.880000	10.299625	5.149813
25%	142.000000	3300.000000	1590.400000	14.085887	7.042944
50%	296.500000	4650.000000	3320.800000	16.099389	8.049695
75%	419.500000	5800.000000	4698.400000	20.331198	10.165599
max	586.000000	9000.000000	6563.200000	71.428571	35.714286

```
[101]: Pior_frete = Dados.loc[Dados["VALOR/KM"] == 5.149813]
Pior_frete
```

```
[101]: Empty DataFrame
Columns: [CIDADE , DISTÂNCIA, VALOR , GASTO, VALOR/IDA, VALOR/KM]
Index: []
```

```
[102]: # 5 piores Fretes
cinco_piores_Fretes = Dados.loc[Dados["VALOR "] < Dados["GASTO"] + Dados["VALOR_
↪"]*0.05]
cinco_piores_Fretes
```

```
[102]:
```

	CIDADE	DISTÂNCIA	VALOR	GASTO	VALOR/IDA	VALOR/KM
4	ALMIRO AFONSO - RN	534.0	5500	5980.8	10.299625	5.149813
7	ANTONIO MARTINS - RN	520.0	5500	5824.0	10.576923	5.288462
8	APODI - RN	450.0	4700	5040.0	10.444444	5.222222
36	ÊRERE -CE	446.0	4600	4995.2	10.313901	5.156951
88	RERIUTABA	312.0	3500	3494.4	11.217949	5.608974

```
[103]: font1 = {"family": "serif", "weight": "bold", "color": "gray", "size":13}
      """
      Criação da segunda fonte por meio da FontProperties()
      Também passamos alguns parâmetros, como family, weight, style e size.
      """
      font2 = FontProperties(family = "Serif", weight = "bold", style = "normal",
↪size = 13)
      """
      Criando um "local" para alocar a nossa figura
      """
```

```

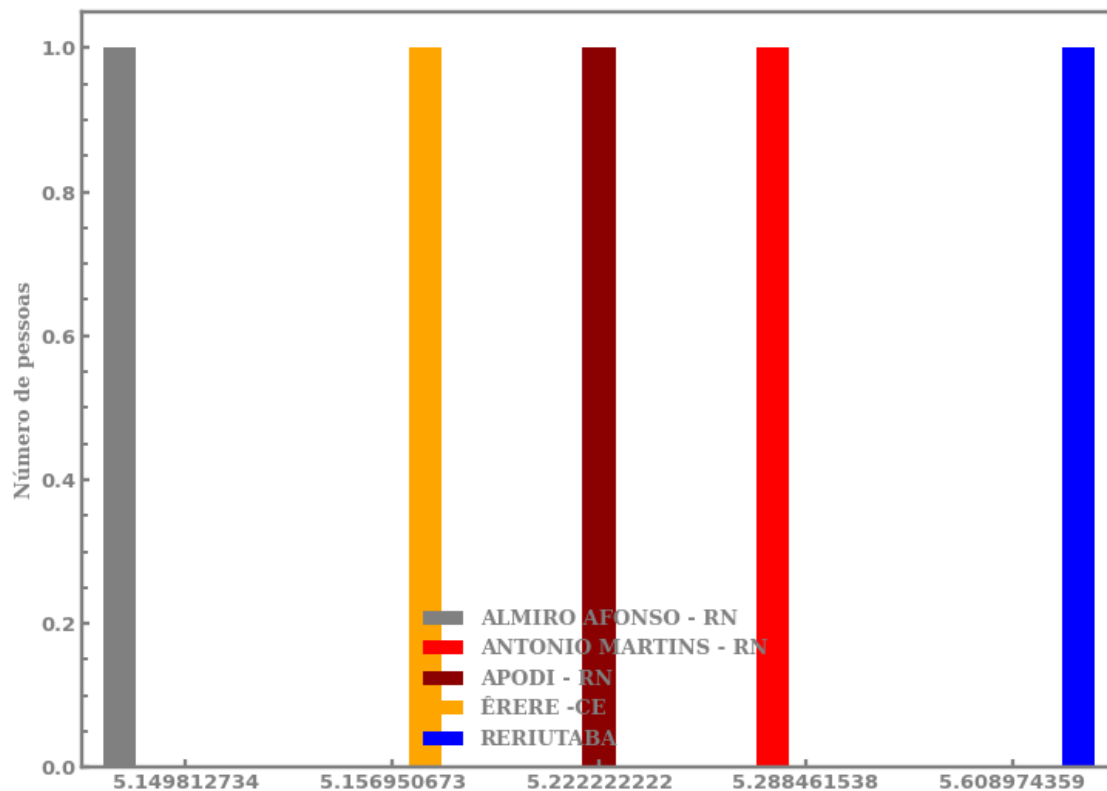
fig, ax = plt.subplots(figsize = (12,9))
"""
plot de gráfico usando o seaborn
"""
sbn.countplot(cinco_piores_Fretes["VALOR/KM"],
               hue = cinco_piores_Fretes["CIDADE "],
               palette = ["gray", "red", "darkred", "orange", "blue"], # Cores
               saturation = 1) # Saturação das cores
"""
Definição de linewidth e da cor para as bordas do gráfico
"""
for axis in ["left", "right", "top", "bottom"]:
    ax.spines[axis].set_linewidth(2.4) # Definindo o linewidth das bordas do
    ↪ gráfico
    ax.spines[axis].set_color("gray") # Definindo a cor das bordas do gráfico
"""
Trabalhando com os ticks no gráfico
"""
ax.xaxis.set_minor_locator(AutoMinorLocator())
ax.yaxis.set_minor_locator(AutoMinorLocator())
ax.tick_params(axis = "both", direction = "in", labelcolor = "gray", labelsiz
    ↪ e = 13, left = True, bottom = True, top = False, right = False)
ax.tick_params(which = "major", direction = "in", color = "gray", length = 5.4,
    ↪ width = 2.3)
ax.tick_params(which = "minor", direction = "in", color = "gray", length = 4,
    ↪ width = 2, bottom = False)
"""
Mostrar tudo que é de texto em negrito.
"""
plt.rcParams["font.weight"] = "bold"
plt.rcParams["axes.labelweight"] = "bold"
"""
Definição de um fundo branco.
"""
fig.patch.set_facecolor("white")
"""
Propriedades da legenda da figura
"""
plt.legend(frameon = False, prop = font2, labelcolor = "gray")
ax.set_xlabel("")
ax.set_ylabel("Número de pessoas", fontdict = font1)
"""
Mostrar gráfico
"""
plt.show()

```

C:\Users\User\anaconda3\lib\site-packages\seaborn_decorators.py:36:

FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```



```
[104]: Fretes_super_faturado
```

```
[104]:
```

	CIDADE	DISTÂNCIA	VALOR	GASTO	VALOR/IDA	\
6	AMONTADA - CE	121.0	3300	1355.20	27.272727	
15	BATURITE-CE	144.0	3500	1612.80	24.305556	
19	BOA VISTA DISTRITO(ITAPIPOCA)	118.0	3200	1321.60	27.118644	
25	CASCÁVEL -CE	110.0	2500	1232.00	22.727273	
26	CAUCAIA -CE	34.6	1800	387.52	52.023121	
37	EUSEBIO -CE	70.7	2000	791.84	28.288543	
38	FORTALEZA -CE	53.5	1800	599.20	33.644860	
41	HORIZONTE - CE	92.3	2200	1033.76	23.835320	
49	ITAPAJÉ - CE	112.0	3300	1254.40	29.464286	
51	ITAPIPOCA - CE	117.0	3300	1310.40	28.205128	
60	MARACANAU -CE	59.4	1800	665.28	30.303030	
61	MARANGUAPE - CE	62.3	2000	697.76	32.102729	
69	PACAJUS -CE	103.0	2500	1153.60	24.271845	

70	PACOTI -CE	130.0	4500	1456.00	34.615385
72	PALMACIA - CE	107.0	4500	1198.40	42.056075
73	PARACURU -CE	30.3	2000	339.36	66.006601
74	PARAIPABA -CE	34.5	2200	386.40	63.768116
79	PENTECOSTE - CE	75.4	2800	844.48	37.135279
81	PINDORETAMA - CE	101.0	2500	1131.20	24.752475
87	REDENÇÃO -CE	102.0	2500	1142.40	24.509804
93	SÃO GONÇALO DO AMARANTE -CE	22.4	1600	250.88	71.428571
100	TRAIRI-CE	83.8	2500	938.56	29.832936
102	TURURU -CE	93.6	2600	1048.32	27.777778
104	URUBURETAMA - CE	97.6	2800	1093.12	28.688525

	VALOR/KM
6	13.636364
15	12.152778
19	13.559322
25	11.363636
26	26.011561
37	14.144272
38	16.822430
41	11.917660
49	14.732143
51	14.102564
60	15.151515
61	16.051364
69	12.135922
70	17.307692
72	21.028037
73	33.003300
74	31.884058
79	18.567639
81	12.376238
87	12.254902
93	35.714286
100	14.916468
102	13.888889
104	14.344262