

▼ Análise de custo e venda de produtos

▼ Módulos

```
import pandas as pd      # análise de dados
import numpy as np       # operações matemáticas
import scipy as sp       # operações matemáticas
import sympy as sm       # cálculos computacionais
import csv
```

▼ Carregamento dos dados

```
# transformando os arquivos txt para csv e carregando-os em DataFrames
arquivo_txt = '/content/CustoProducao.txt'
arquivo_csv = 'CustoProducao.csv'

with open(arquivo_txt, 'r', encoding='latin1') as txt_file:
    colunas = txt_file.readline().strip().split(';')

    with open(arquivo_csv, 'w', newline='', encoding='latin1') as csv_file:
        csv_writer = csv.DictWriter(csv_file, fieldnames=colunas, delimiter=';')
        csv_writer.writeheader()

        for linha in txt_file:
            valores = linha.strip().split(';')
            linha_dict = {coluna: valor for coluna, valor in zip(colunas, valores)}
            csv_writer.writerow(linha_dict)

arquivo1_txt = '/content/PrecosMensalUF.txt'
arquivo1_csv = 'PrecosMensal.csv'

with open(arquivo1_txt, 'r', encoding='latin1') as txt_file:
    colunas = txt_file.readline().strip().split(';')

    with open(arquivo1_csv, 'w', newline='', encoding='latin1') as csv_file:
        csv_writer = csv.DictWriter(csv_file, fieldnames=colunas, delimiter=';')
        csv_writer.writeheader()

        for linha in txt_file:
            valores = linha.strip().split(';')
            linha_dict = {coluna: valor for coluna, valor in zip(colunas, valores)}
            csv_writer.writerow(linha_dict)

#lendo e carregando o arquivos para variáveis
custo_producao = pd.read_csv ('/content/CustoProducao.csv',encoding='latin1', sep = ';', decimal= ',')
venda_producao = pd.read_csv ('/content/PrecosMensal.csv',encoding='latin1', sep = ';', decimal= ',')
```

▼ Estabilidade de custo de produção por produto

```
# convertendo para numérico
custo_producao['vlr_renda_fator_unidade'] = pd.to_numeric(custo_producao['vlr_renda_fator_unidade'])
custo_producao['vlr_custo_fixo_unidade'] = pd.to_numeric(custo_producao['vlr_custo_fixo_unidade'])
custo_producao['vlr_custo_variavel_unidade'] = pd.to_numeric(custo_producao['vlr_custo_variavel_unidade'])

# filtrando o que será utilizado
custo_producao = custo_producao[custo_producao['vlr_renda_fator_unidade'] >= 0.5]
contagem_produto = custo_producao['produto'].value_counts()
produtos10 = set(contagem_produto[contagem_produto >= 10].index)

custo_producao = custo_producao[custo_producao['produto'].isin(produtos10)]

#calculando o custo total de produção
custo_producao['vlr_total_unidade'] = custo_producao['vlr_custo_fixo_unidade'] + custo_producao['vlr_custo_variavel_unidade'] + custo_prc
```

- Estipulando a variabilidade e definindo o ranqueamento

```
#calculando variância, desvio padrão e aptitude total
var_custo_produto = custo_producao.groupby('produto')['vlr_total_unidade'].var()
```

```

dp_custo_produto = custo_producao.groupby('produto')['vlr_total_unidade'].std()
ampl_custo_produto = custo_producao.groupby('produto')['vlr_total_unidade'].agg(lambda x: np.max(x) - np.min(x))

#concatenando os resultados e cruzando as informações por meio do cálculo da média.
variabilidade_custo = pd.concat([var_custo_produto, dp_custo_produto, ampl_custo_produto])

media_variabilidade_custo = variabilidade_custo.groupby('produto').mean()

#ordenação crescente
custos_estaveis = media_variabilidade_custo.sort_values(ascending=True)

custos_estaveis.head(5)

      produto
SORGO GRANIFERO      90.054676
AMENDOIM             92.286050
MILHO                138.166111
TRITICALE            233.699894
AVEIA                260.703473
Name: vlr_total_unidade, dtype: float64

```

▼ Estabilidade de venda por produto

```

# convertendo para numérico
venda_producao['valor_produto_kg'] = pd.to_numeric(venda_producao['valor_produto_kg'])

# filtrando o que será utilizado
contagem_produto = venda_producao['produto'].value_counts()
produtos10 = set(contagem_produto[contagem_produto >= 10].index)

venda_producao = venda_producao[venda_producao['produto'].isin(produtos10)]

• Estipulando a variabilidade e definindo o ranqueamento

#calculando variância, desvio padrão e aptitude total
var_venda_produto = venda_producao.groupby('produto')['valor_produto_kg'].var()
dp_venda_produto = venda_producao.groupby('produto')['valor_produto_kg'].std()
ampl_venda_produto = venda_producao.groupby('produto')['valor_produto_kg'].agg(lambda x: np.max(x) - np.min(x))

#concatenando os resultados e cruzando as informações por meio do cálculo da média.
variabilidade_venda = pd.concat([var_venda_produto, dp_venda_produto, ampl_venda_produto])

media_variabilidade_venda = variabilidade_venda.groupby('produto').mean()

#ordenação crescente
vendas_estaveis = media_variabilidade_venda.sort_values(ascending=True)

vendas_estaveis.head(5)

      produto
CANA DE ACUCAR      0.047322
MACAUBA             0.097491
BARU                0.116670
COCO DE BABACU      0.138034
MILHETO             0.204069
Name: valor_produto_kg, dtype: float64

```

