

## Algebra Linear, Vetores e Geometria Analitica

Após a coleta de dados, alguns dados foram escolhidos para serem representados em forma de matriz ou vetor:

```
genero = dfcus["gender"].value_counts(dropna=False).sort_index() #Seleciona a quantidade de pessoas de cada genero
enrich = dfcus[dfcus["isEnriched"]==0]["gender"].value_counts(dropna=False).reindex(genero.index, fill_value=0) #Seleciona quantos dados foram melhorados
enrichn = dfcus[dfcus["isEnriched"]==1]["gender"].value_counts(dropna=False).reindex(genero.index, fill_value=0) #Seleciona quantos dados não foram melhorados

matrizgen = np.vstack([genero.values, enrich.values, enrichn.values]) #Monta a matriz com os dados selecionados

print(matrizgen) #Mostra a matriz montada
```

[[244 234 263 259]
[119 128 121 128]
[125 106 142 131]]

```
df_matrizgen = pd.DataFrame( #Estrutura da matriz com índices
    matrizgen, #carrega a matriz
    index = ["Total", "Enriched", "Não Enriched"],
    columns = ["Feminino", "Masculino", "Outro", "Não Selecionado"]
)
```

```
print(df_matrizgen) #Mostra a matriz
```

	Feminino	Masculino	Outro	Não Selecionado
Total	244	234	263	259
Enriched	119	128	121	128
Não Enriched	125	106	142	131

```
statuscamp = dfcamp["status"].value_counts().sort_index() #Seleciona quantas campanhas existem com o mesmo status
typeum = dfcamp[dfcamp["type"]==1]["status"].value_counts().reindex(statuscamp.index, fill_value=0) #Seleciona a quantidade de campanhas promocionais
typedois = dfcamp[dfcamp["type"]==2]["status"].value_counts().reindex(statuscamp.index, fill_value=0) #Seleciona a quantidade de campanhas institucionais

matrizcamp = np.vstack([statuscamp.values, typeum.values, typedois.values]) #Monta a matriz com os dados selecionados

print(matrizcamp) #Mostra a matriz montada
```

```
[[500 497 480 523]
 [265 257 270 264]
 [235 240 210 259]]
```

```
df_matrizcamp = pd.DataFrame( #Estrutura da matriz com índices
    matrizcamp, #carrega a matriz
    index = ["Total", "Promocional", "Institucional"],
    columns = ["Rascunho", "Cancelado", "Publicado", "Completado"]
)
```

```
print(df_matrizcamp) #Mostra a matriz
```

	Rascunho	Cancelado	Publicado	Completado
Total	500	497	480	523
Promocional	265	257	270	264
Institucional	235	240	210	259

```
ordert = dford["orderType"].value_counts() #Seleciona a quantidade de cada tipo de pedido
```

```
vetorord = np.stack(ordert.values) #Coloca os dados em um vetor
```

```
print(vetorord) #Mostra o vetor
```

```
[688 685 627]
```

```
df_ordert = ordert.to_frame().T
```

```
df_ordert.index = [""]
```

```
df_ordert.columns.name = None
```

```
print(df_ordert)
```

DELIVERY	INDOOR	TAKEOUT
688	685	627

Também é possível realizar a transposição das matrizes:

```
df_matrizTgen = pd.DataFrame( #Estrutura da matriz Transposta com indices
    matrizgen.T, #carrega a matriz e transpõe ela
    index = ["Feminino", "Masculino", "Outro", "Não Selecionado"],
    columns = ["Total", "Enriched", "Não Enriched"]
)
```

```
print(df_matrizTgen) #Mostra a matriz transposta
```

	Total	Enriched	Não Enriched
Feminino	244	119	125
Masculino	234	128	106
Outro	263	121	142
Não Selecionado	259	128	131

```
df_matrizTcamp = pd.DataFrame( #Estrutura da matriz transposta com indices
    matrizcamp.T, #carrega a matriz e transpõe ela
    index = ["Rascunho", "Cancelado", "Publicado", "Completado"],
    columns = ["Total", "Promocional", "Institucional"]
)
```

```
print(df_matrizTcamp) #Mostra a matriz transposta
```

	Total	Promocional	Institucional
Rascunho	500	265	235
Cancelado	497	257	240
Publicado	480	270	210
Completado	523	264	259

Com as matrizes feitas, é possível fazer diferentes operações entre elas, como soma e multiplicação.

Como as duas matrizes feitas possuem o mesmo tamanho (3,4), é possível realizar a soma dessas duas matrizes.

```
print(matrizgen)
print("\t+\t")
print(matrizcamp)
print("\t=\n")

matrizS = matrizgen + matrizcamp #Soma entre matrizes

print(matrizS)

[[244 234 263 259]
 [119 128 121 128]
 [125 106 142 131]]
+
[[500 497 480 523]
 [265 257 270 264]
 [235 240 210 259]]
=
[[744 731 743 782]
 [384 385 391 392]
 [360 346 352 390]]
```

Para realizar a multiplicação entre matrizes, o número de colunas da primeira matriz deve ser o mesmo número de linhas da segunda matriz. Como as duas matrizes são iguais, é possível realizar a multiplicação de uma das matrizes com a transposta da outra:

```
print(matrizgen)
print("\tX\t")
print(matrizcamp.T)
print("\t=\n")

matrizV = np.dot(matrizgen, matrizcamp.T) #Multiplicação entre matrizes

print(matrizV)

[[244 234 263 259]
 [119 128 121 128]
 [125 106 142 131]]
      X
[[500 265 235]
 [497 257 240]
 [480 270 210]
 [523 264 259]]
      =
[[499995 264184 235811]
 [248140 130893 117247]
 [251855 133291 118564]]
```