

Arruma_dados

June 17, 2020

```
[1]: import pandas as pd
import numpy as np
import pickle
import os
from scipy.fftpack import fft, fftfreq
import matplotlib as mpl
import matplotlib.pyplot as plt

%load_ext autoreload
%autoreload 2

from ipynb.fs.full.formata_dados import *
from ipynb.fs.full.processa_dados import *

[2]: #pasta na qual estão os arquivos de dados
path = "/mnt/datasets/jupyter_lab/arg/Public_GitHub/Data/EXPERIMENTO OFICIAL_
→2019/"

[3]: #dicionário
## chave: nome do arquivo, valor: dados
#### Cada voluntário participou do experimento 3 dias, ingerindo 2 vezes cada_
→uma das 4 bebidas
#####(agua, suco com aspartame, suco com acucar. suco com sucralose) em cada_
→dia.
##### 11 voluntários = 264 dados
##### 64 dados descartados -> 200 dados válidos
dados_voluntarios_completos = le_dados_EEG(path)

[4]: lista_dados = list(dados_voluntarios_completos.values())
nome_dos_arquivos = list(dados_voluntarios_completos.keys())
output = cria_dicionario_output()

[5]: #referencias das bebidas
agua = 0
acucar = 1
aspartame = 2
```

```
sucralose = 3
dict_output = enumera_dicionario(['agua', 'acucar', 'aspartame', 'sucralose'])
```

```
[6]: #referencia das datas
dicionario_datas = inverte_enumera_dicionario(['07.11','08.11','11.11','12.
↪11','13.11','14.11','27.11','28.11','02.12','03.12','04.12'])
inverte_enumera_dicionario = enumera_dicionario(['07.11','08.11','11.11','12.
↪11','13.11','14.11','27.11','28.11','02.12','03.12','04.12'])
```

```
[7]: #referencia dos individuos
dict_individuo = enumera_dicionario(['Camila Mendes', 'Camila_
↪Gomes','Beatriz','Carol Monica',
                                'Isabella','Isa Peron','Lais','Maria_
↪Clara','Orlando','Rafaela'])
camilamendes = 0
camila = 1 #camila gomes
beatriz = 2
carolmonica = 3
isabella = 4
isaperon = 5
lais = 6
mariaclara = 7
orlando = 8
rafa = 9
```

```
[8]: #Enumera as bebidas ingeridas e nomeia os voluntarios participantes de cada_
↪experimento
lista_output, individuos = arruma_output(nome_dos_arquivos, output, acucar,
↪agua, sucralose, aspartame)
```

```
[9]: #Deixa todos os vetores com o mesmo tamanho
lista_dados_padronizados, data_experimento = padroniza_dados(lista_dados)
```

Os dados foram padronizados

```
[10]: #enumera os voluntarios participantes de cada experimento
lista_individuos = gera_lista_individuos(individuos, camilamendes, camila,
↪beatriz, carolmonica, isabella, isaperon, lais, mariaclara, orlando, rafa)
```

```
[11]: #Se False: Carrega dados
#Se True: Gera dados
gera_dados = False

if gera_dados:
    dados_originais = completa_dados(lista_dados_padronizados,lista_output,
↪dicionario_datas, lista_individuos, data_experimento)
```

```

    with open("../Public_GitHub/Data/Pickles/dados_originais.pkl", "wb") as f:
        pickle.dump([dados_originais], f)
    else:
        with open("../Public_GitHub/Data/Pickles/dados_originais.pkl", "rb") as f:
            dados_originais = pickle.load(f)
            dados_originais = dados_originais[0]

```

```

[12]: # ordena os dados originais por data e por individuo - coluna 9218 = data,
      #       coluna 9219 = individuo
      dados_originais_ordenados = dados_originais.sort_values([9218, 9219])

```

```

[13]: # transforma os dados originais de pandas.DataFrame para numpy.array
      dados_numpy = np.array(dados_originais_ordenados)

```

```

[14]: ok, errado = valida_dados(dados_numpy[:, :-4])

```

```

[15]: #como a funcao "valida_dados" retorna os indices dos vetores, aqui os vetores
      #       corretos estao sendo filtrados
      dados_numpy_ok = dados_numpy[ok]

```

```

[16]: #separa os dados por bebida
      dados_separados_por_output = separa_dados_por_output(dados_numpy_ok, dict_output)

```

```

[17]: #Se True: salva os dados gerados como pickle
      salva_dados = True
      if salva_dados:
          with open("../Public_GitHub/Data/Pickles/dados_ok.pkl", "wb") as f:
              pickle.dump([dados_numpy_ok], f)
          with open("../Public_GitHub/Data/Pickles/dados_agua_ok.pkl", "wb") as f:
              pickle.dump([dados_separados_por_output['agua']], f)
          with open("../Public_GitHub/Data/Pickles/dados_aspartame_ok.pkl", "wb") as f:
              pickle.dump([dados_separados_por_output['aspartame']], f)
          with open("../Public_GitHub/Data/Pickles/dados_sucralose_ok.pkl", "wb") as f:
              pickle.dump([dados_separados_por_output['sucralose']], f)
          with open("../Public_GitHub/Data/Pickles/dados_acucar_ok.pkl", "wb") as f:
              pickle.dump([dados_separados_por_output['acucar']], f)

```

```

        with open("../Public_GitHub/Data/Pickles/todos_os_dados_originais.pkl",
↪ "wb") as todos_os_dados:
            pickle.dump([dados_numpy], todos_os_dados)
    else:
        print("Seus novos dados não foram salvos")

```

```

[18]: #nomeia as colunas de output
tabela_dados = pd.DataFrame(dados_numpy_ok)
tabela_dados.rename(columns={9216: 'Canal'}, inplace = True)
tabela_dados.rename(columns={9217: 'Bebida'}, inplace = True)
tabela_dados.rename(columns={9218: 'Data'}, inplace = True)
tabela_dados.rename(columns={9219: 'Individuo'}, inplace = True)

```

```

[19]: #nomeia os outputs
for i in list(range(4)):
    tabela_dados['Bebida'] = tabela_dados['Bebida'].replace(i, dict_output[i])
for i in range(len(inverte_enumeracao_dicionario)):
    tabela_dados['Data'] = tabela_dados['Data'].replace(i,
↪ inverte_enumeracao_dicionario[i])
for i in range(len(dict_individuo)):
    tabela_dados['Individuo'] = tabela_dados['Individuo'].replace(i,
↪ dict_individuo[i])

```

```

[20]: #tabela_dados

```