Aula Prática 2: Curvas de Carnegie

Nesta aula prática iremos analisar dados de campo elétrico e raios coletados em São Paulo, de forma a:

- Se familiarizar com o formato de dados coletados
- Gerar gráficos para análise inicial das séries temporais
- Interpretar a variabilidade do campo elétrico e como a incidência de raios afeta essa variabilidade

1. Importando os dados

Os dados estão organizados da seguinte forma:

- Campo elétrico vertical:
  - Pasta */campo_eletrico/*
  - Arquivos *campo_bom_YYYY-mm.dat*
    - 1440 linhas: minutos do dia
    - 2 colunas: minuto do dia, campo elétrico do mês sem a presença de tempestades (em V/m)
- Raios:
  - Pasta */raios_minuto/*
  - Arquivos *serie_SP_YYYY-mm.dat*
    - 1440 linhas: minutos do dia
    - 13 colunas: minuto do dia, número médio de raios no mês a diferentes distâncias do sensor (em km)

col2	col3	col4	col5	col6	col7	col8	col9	col10	col11	col12	col13
20	500	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500

> Nota: arquivos *.dat*, assim como *.csv* e às vezes até *.txt*, normalmente estão associados a dados tabulados, podendo ser lidos em editores de texto básicos e também no Microsoft Excel (se importado de forma correta)

Para ler os dados e gerar os gráficos necessários, iremos utilizar duas bibliotecas amplamente utilizadas em Python:

- Pandas: leitura e análise de dados tabulados
- Matplotlib: geração de gráficos

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
```

Além disso, precisamos montar a nossa pasta do Google Drive para conseguir acessar os arquivos de dados, usando a biblioteca própria `google.colab`.

> Outra opção para montar a pasta do Drive é abrir a aba "Arquivos" na barra lateral esquerda e selecionar o botão "Montar Drive". **Nesse caso, o código abaixo não precisa ser rodado**

```
1 from google.colab import drive
2 drive.mount('drive')
```

```
Drive already mounted at drive; to attempt to forcibly remount, call drive.mount("dri
```

Para ler um arquivo, basta usar a função `read_table` do pandas (`pd`) definindo como as colunas estão separadas (parâmetro `sep`).
Neste caso, como os arquivos não possuem nomes de colunas, iremos criá-los também.

> Não esqueça de modificar o caminho `file_path` abaixo, indicando a pasta onde estão os arquivos da aula prática no seu Drive. Use a barra lateral para encontrar a pasta se for necessário

```
1 #@title
2
3 # Definindo o caminho dos arquivos
4 file_path="/content/drive/MyDrive/"
5 # Abrindo um dos arquivos
6 file = pd.read_table(file_path + "CampoEletrico/campo_bom_2015-09.dat", sep="   ", name
7 file.head()
```

	minuto	campo_eletrico
0	0	-23.83
1	1	-24.39
2	2	-24.71
3	3	-24.90

Assim, para ler todos os arquivos de campo elétrico:

```
1 setembro = pd.read_table(file_path + "CampoEletrico/campo_bom_2015-09.dat", sep="   ",
2 outubro = pd.read_table(file_path + "CampoEletrico/campo_bom_2015-10.dat", sep="   ", n
3 novembro = pd.read_table(file_path + "CampoEletrico/campo_bom_2015-11.dat", sep="   ",
4 dezembro = pd.read_table(file_path + "CampoEletrico/campo_bom_2015-12.dat", sep="   ",
5 janeiro = pd.read_table(file_path + "CampoEletrico/campo_bom_2016-01.dat", sep="   ", n
6 fevereiro = pd.read_table(file_path + "CampoEletrico/campo_bom_2016-02.dat", sep="   "
```

```
6 fevereiro = pd.read_table(file_path + "CampoEletrico/campo_bom_2016-02.dat", sep="     ",
7 marco = pd.read_table(file_path + "CampoEletrico/campo_bom_2016-03.dat", sep="   ", nam
```

Clique duas vezes (ou prima Enter) para editar.

Exercício 0 [1 ponto]

Leia os arquivos de raios, lembrando de como as colunas estão distribuídas e configurando o separador entre elas

```
1 r_dezembro = pd.read_table(file_path + "RaiosMinuto/serie_SP_2015-12.dat", sep="     ",
2 r_novembro = pd.read_table(file_path + "RaiosMinuto/serie_SP_2015-11.dat", sep="     ",
3 r_outubro  = pd.read_table(file_path + "RaiosMinuto/serie_SP_2015-10.dat", sep="     ",
4 r_setembro = pd.read_table(file_path + "RaiosMinuto/serie_SP_2015-09.dat", sep="     ",
5
6
```

2. Fazendo gráficos

Com os dados devidamente carregados, agora iremos analisá-los através de gráficos.

Uma primeira abordagem é simplesmente plotar uma coluna em função da outra. No caso dos dados de campo elétrico, esse `plot` (uma função do matplotlib - `plt`) será do próprio campo elétrico (eixo y, segundo argumento) em função do tempo (eixo x, primeiro argumento).

```
1
```

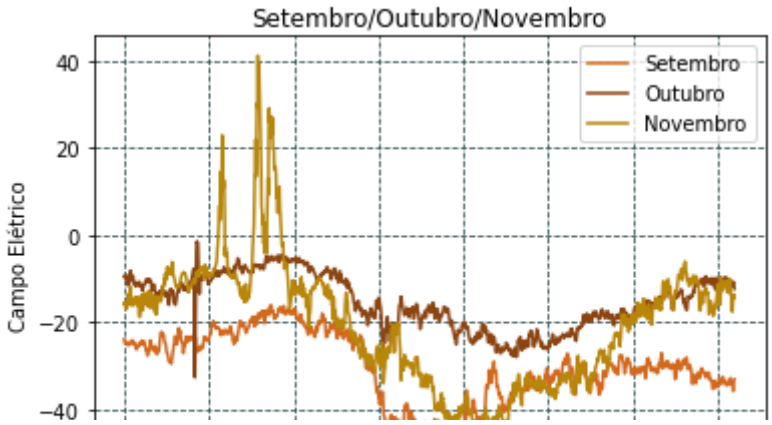
Exercício 1 [2 pontos]

Faça os gráficos de campo elétrico dos meses restantes

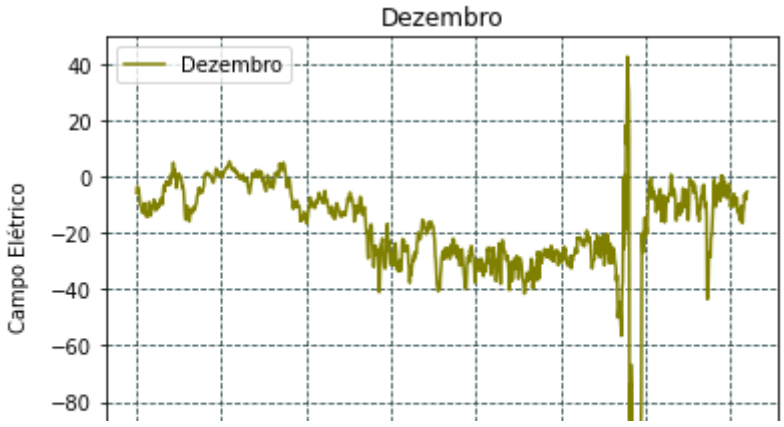
> **DICA: É possível adicionar linhas ao gráfico simplesmente adicionando os argumentos de cada eixo à mesma função `plt.plot`. Elas serão plotadas em cores diferentes; para controlar a cor de cada linha adicione um parâmetro com a cor desejada após os eixos. Para identificar qual linha é qual no gráfico, adicione uma legenda com a função `plt.legend` e o parâmetro `labels` com um nome para cada linha**

```
1 plt.plot(setembro["minuto"], setembro["campo_eletrico"], color='chocolate', label='Sete
2 plt.plot(outubro["minuto"],  outubro["campo_eletrico"], color='saddlebrown', label='Out
3 plt.plot(novembro["minuto"], novembro["campo_eletrico"], color='darkgoldenrod', label='
4 plt.legend()
5 plt.title("Setembro/Outubro/Novembro")
6 plt.ylabel("Campo Elétrico")
7 plt.xlabel("Minutos")
```

```
plt.grid('True', linestyle='--', color='darkslategrey')
plt.show()

```

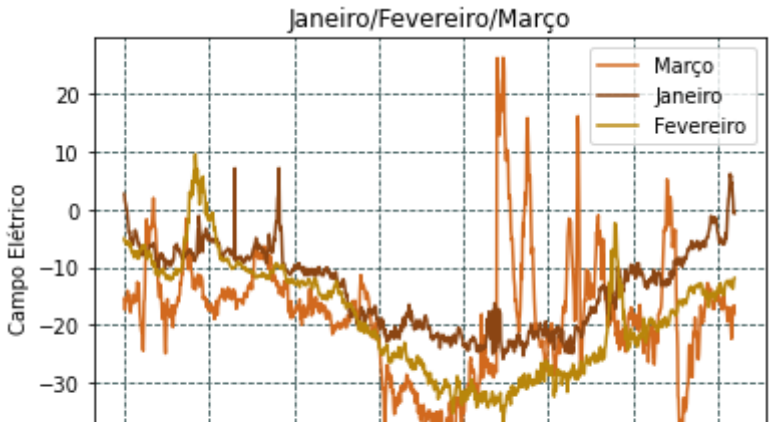



```
import numpy as np
#pd.to_numeric(dezembro["minuto"],downcast=float)
array = np.array([])
for i in range (0,1440):
  array = np.append(array, i)
plt.plot(array, dezembro["campo_eletrico"], color='olive', label='Dezembro')

plt.legend()
plt.title("Dezembro")
plt.ylabel("Campo Elétrico")
plt.xlabel("Minutos")
plt.grid('True', linestyle='--', color='darkslategrey')
plt.show()
```




```
plt.plot(marco["minuto"], marco["campo_eletrico"], color='chocolate', label='Março')
plt.plot(janeiro["minuto"],  janeiro["campo_eletrico"], color='saddlebrown', label='Jan
plt.plot(fevereiro["minuto"], fevereiro["campo_eletrico"], color='darkgoldenrod', label
plt.legend()
plt.title("Janeiro/Fevereiro/Março")
```

```
plt.ylabel("Campo Elétrico")
plt.xlabel("Minutos")
plt.grid('True', linestyle='--', color='darkslategrey')
plt.show()

```



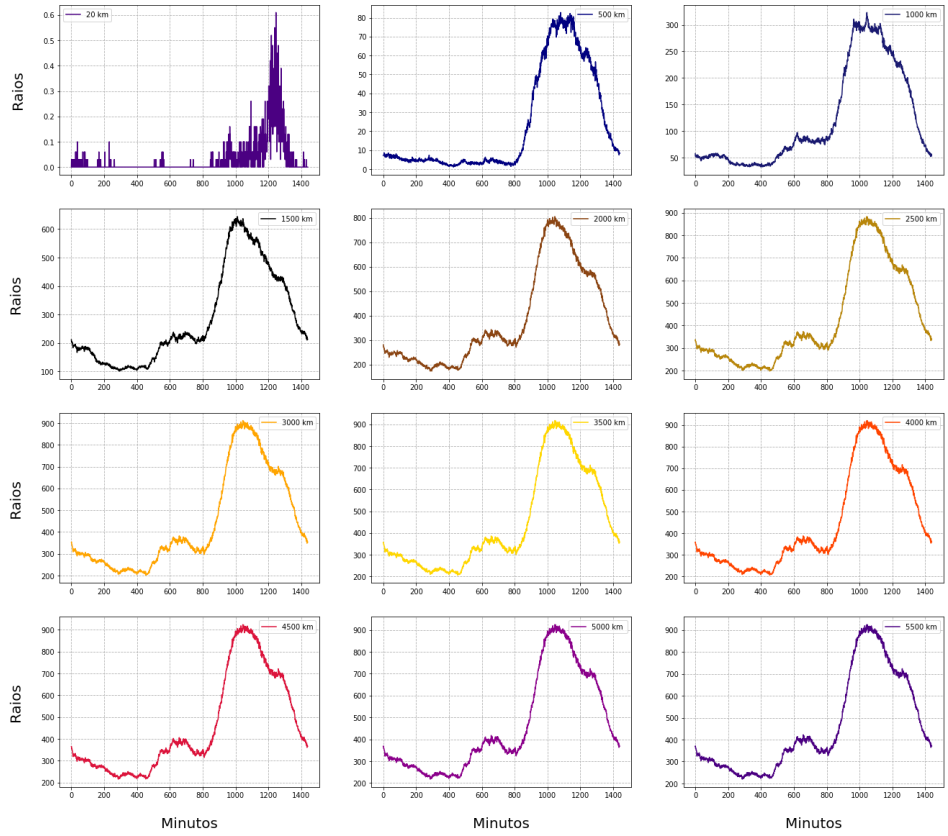
Exercício 2 [3 pontos]

Faça os gráficos de número médio de raios em função do tempo para todas as distâncias e meses

```
import matplotlib.colors as colors
import glob
from PIL import Image

import numpy as np
#pd.to_numeric(dezembro["minuto"],downcast=float)
array = np.array([])
for i in range (0,1440):
  array = np.append(array, i)
fig, ((ax0, ax1, ax2), (ax3, ax4, ax5), (ax6,ax7, ax8), (ax9, ax10, ax11)) = plt.subplo
ax0.plot(array, r_dezembro["20_km"], color='indigo', label='20 km')
ax0.grid('True', linestyle ='--')
ax0.set_ylabel('Raios', fontsize=20, labelpad=25)
ax0.legend()

ax1.plot(array, r_dezembro["500_km"], color='navy', label ='500 km')
ax1.grid('True', linestyle ='--')
ax1.legend()

ax2.plot(array, r_dezembro["1000_km"], color='midnightblue', label = '1000 km')
ax2.grid('True', linestyle ='--')
ax2.legend()

ax3.plot(array, r_dezembro["1500_km"], color='black', label = '1500 km')
ax3.grid('True', linestyle ='--')
ax3.set_ylabel('Raios', fontsize=20, labelpad=25)
ax3.legend()
```

```
ax3.legend()

ax4.plot(array, r_dezembro["2000_km"], color='saddlebrown', label = '2000 km')
ax4.grid('True', linestyle ='--')
ax4.legend()

ax5.plot(array, r_dezembro["2500_km"], color='darkgoldenrod', label = '2500 km')
ax5.grid('True', linestyle ='--')
ax5.legend()

ax6.plot(array, r_dezembro["3000_km"], color='orange', label= '3000 km')
ax6.grid('True', linestyle ='--')
ax6.set_ylabel('Raios', fontsize=20, labelpad=25)
ax6.legend()

ax7.plot(array, r_dezembro["3500_km"], color='gold', label ='3500 km')
ax7.grid('True', linestyle ='--')
ax7.legend()

ax8.plot(array, r_dezembro["4000_km"], color='orangered', label='4000 km')
ax8.grid('True', linestyle ='--')
ax8.legend()

ax9.plot(array, r_dezembro["4500_km"], color='crimson', label = '4500 km')
ax9.grid('True', linestyle ='--')
ax9.set_ylabel('Raios', fontsize=20, labelpad=25)
ax9.set_xlabel('Minutos', fontsize=20, labelpad=25)
ax9.legend()

ax10.plot(array, r_dezembro["5000_km"], color='darkmagenta', label ='5000 km ')
ax10.grid('True', linestyle ='--')
ax10.set_xlabel('Minutos', fontsize=20, labelpad=25)
ax10.legend()

ax11.plot(array, r_dezembro["5500_km"], color='indigo', label = '5500 km')
ax11.grid('True', linestyle ='--')
ax11.set_xlabel('Minutos', fontsize=20, labelpad=25)
ax11.legend()

```

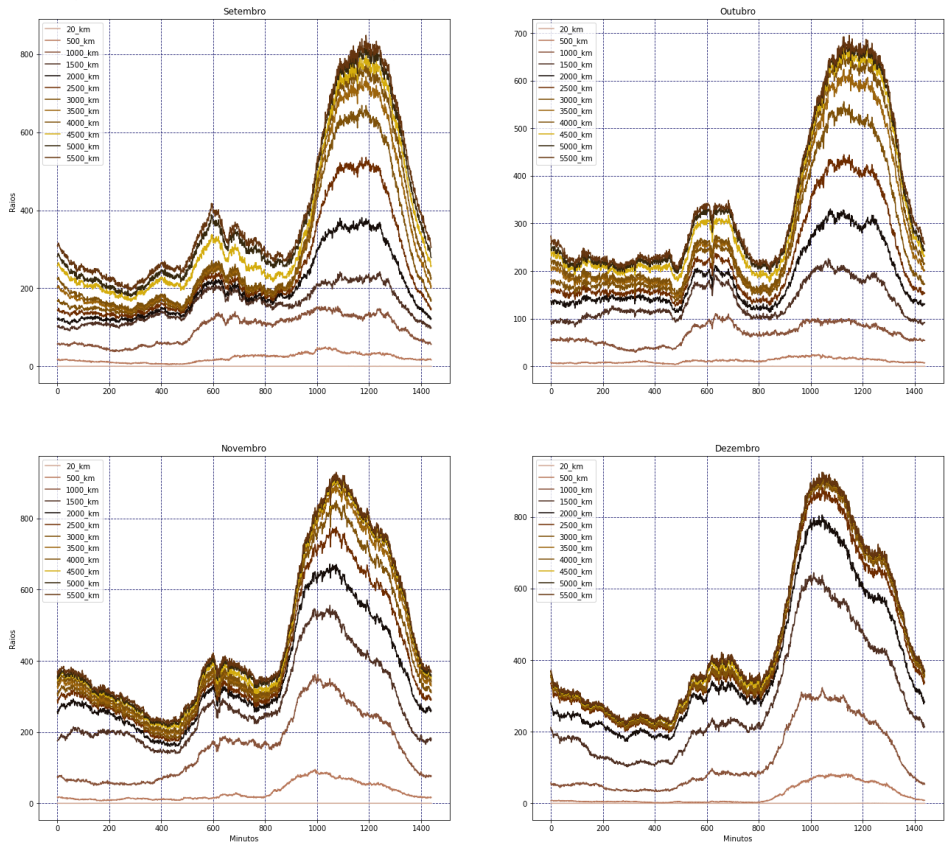
```
<matplotlib.legend.Legend at 0x7f150438b850>
```




```
1 fig, ((ax0, ax1), (ax2, ax3)) = plt.subplots(2,2, figsize=(22,20))
2 marrom = ['#d1a895', '#b9795c', '#89533a','#503022', '#160d09', '#6E2C00','#7e5109', '#
3 lista =["20_km", "500_km", "1000_km", "1500_km", "2000_km", "2500_km", "3000_km", "3500
4 for i in range (0,12):
5   ax0.plot(array, r_setembro[lista[i]], label='{}'.format(lista[i]), color= marrom[i])
6   ax0.legend()
7   ax1.plot(array, r_outubro[lista[i]],label='{}'.format(lista[i]), color= marrom[i])
```

```
  ax1.legend()
  ax2.plot(array, r_novembro[lista[i]],label='{}'.format(lista[i]), color= marrom[i])
  ax2.legend()
  ax3.plot(array, r_dezembro[lista[i]],label='{}'.format(lista[i]), color= marrom[i])
  ax3.legend()
ax0.grid('True', linestyle='--', color='midnightblue')
ax1.grid('True', linestyle='--', color='midnightblue')
ax2.grid('True', linestyle='--', color='midnightblue')
ax3.grid('True', linestyle='--', color='midnightblue')
ax0.set_ylabel('Raios')
ax2.set_ylabel('Raios')
ax2.set_xlabel('Minutos')
ax3.set_xlabel('Minutos')
ax0.set_title('Setembro')
ax1.set_title('Outubro')
ax2.set_title('Novembro')
ax3.set_title('Dezembro')


```

```
Text(0.5, 1.0, 'Dezembro')
```



3. Interpretando os resultados

Exercício 3 [2 pontos]

Identifique a partir de que distância os raios começam a contribuir para o circuito elétrico global

A contribuição dos raios em escala global ocorre para maiores distância, com o distanciamento das fontes, maiores áreas superficiais podem afetadas pelos raios podem ser observadas, por causa de sua abertura angular, o detector é mais apto em cobrir maiores áreas, com o afastamento das fontes, que em muitos casos pode significar a observação de mais fontes elétricas. Com os gráficos de **Raios x Minutos** podemos observar que a partir de algumas centenas de quilómetros o comportamento elétrico passa a ter um comportamento mais padronizado e a partir de determinadas distâncias as curvas começam a ter frequências de incindência cada vez mais similares, como se estivessem limitadas, a distância que delimita esse comportamento varia para cada mês, e são aproximadamente 3500 km, 3500 km, 4500 km e 2500 km para os meses de Setembro, Outubro, Novembro e Dezembro, respectivamente.

Exercício 4 [1 ponto]

Essa contribuição aumenta com a distância?

Sim, as curvas são mais coincidentes para maiores distâncias e também parecem afetar mais o campo elétrico.

Exercício 5 [1 ponto]

Essa contribuição é igual em todos os meses? Explique.

Não, pois observa-se pequenas variações entre os meses, tanto na intensidade do campo elétrico, quanto na incidência de raios. Variação essa mais visível para o mês de Dezembro, em que percebe-se uma "aglutinação" mais acentuada e coincidente das curvas, como se a partir de 2500 km a atividade elétrica fosse mais limitada e não pudesse aumentar muito mais que o observado para 2500 km. Em Setembro também é possível observar dois picos menores entre 500 e 800 minutos, com o passar dos meses eles parecem coincidir em um único pico de atividade. Portanto, apesar de observar a influência no circuito elétrico global, as variações sazoanais regionais também afetam as curvas de longas distâncias.

✓ 3 s concluído à(s) 09:35