Cruzeiro do Sul Virtual
Educação a Distância

VISUALIZAÇÃO TEMPORAL

Prof. Ismar Frango

Visualização De Informação Temporal

A necessidade de representar a passagem do tempo sempre esteve presente na evolução da humanidade. Os primeiros astrônomos da Antiguidade, assim como os primeiros Historiadores, já buscavam maneiras de comunicar, visualmente, eventos relativos à passagem do tempo.

Dados temporais sempre consistiram em um desafio, tanto no que se diz respeito a como registrar esses dados, bem como nos mecanismos de representação e visualização.

Veremos que a representação visual mais comumente utilizada para visualização de séries temporais são os gráficos de linha, embora outras formas de visualização possam ser exploradas.

Uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo.

As séries temporais muitas vezes podem ser aproximadas por meio de funções matemáticas, ou seja, que pode existir uma lei de formação que determina o comportamento de uma série temporal.

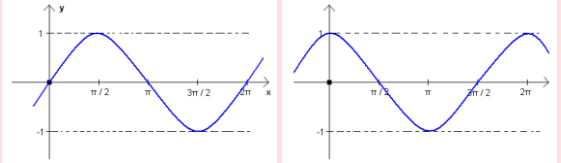
Nesta unidade, iremos estudar as principais estratégias de visualização de informação temporal, começando com uma abordagem histórica, chegando à implementação de algumas representações em Pyhton.

Fonte https://pixabay.com/photos/time-timer-clock-watch-hour-371226/ - Licença Pixabay

Um pouco de História

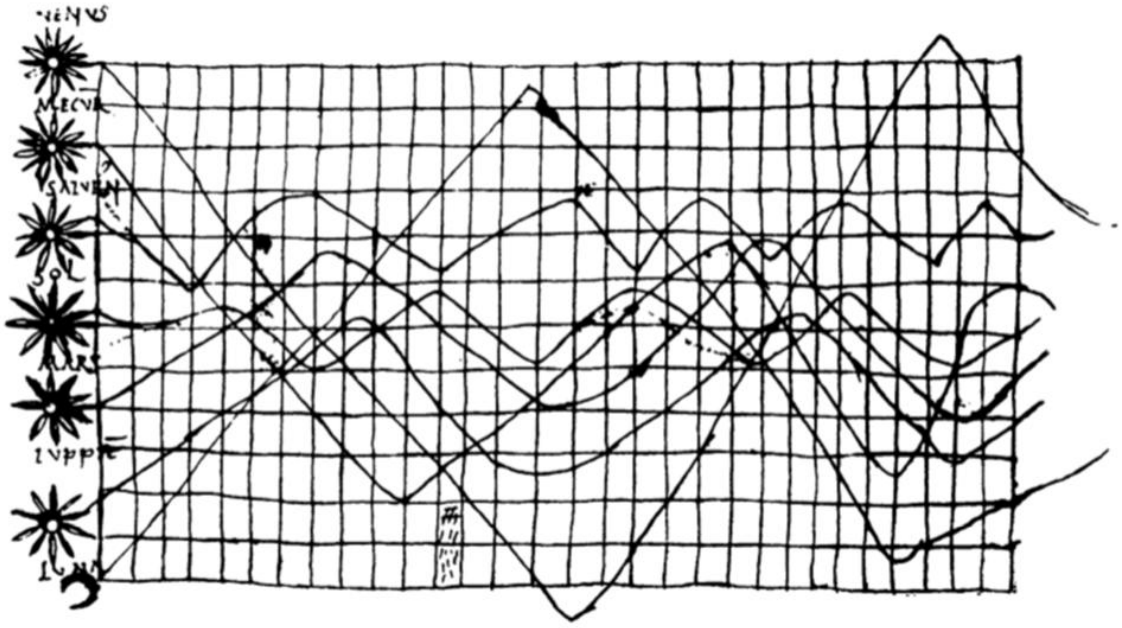
O gráfico a seguir é, possivelmente, a representação visual de dados temporais mais antiga que se tem notícia: trata-se de um gráfico do século X (ou XI), representando as inclinações nas órbitas planetárias como uma função do tempo – note que os traços da figura (próxima página) se aproximam dos gráficos de uma senoide ou cosenoide, por exemplo.

Gráficos das funções seno e cosseno:



Fontes: https://pt.wikipedia.org/wiki/Ficheiro:Funcao_trigonometrica_seno.PNG e https://pt.wikipedia.org/wiki/Ficheiro:Funcao_trigonometrica_coseno.PNG

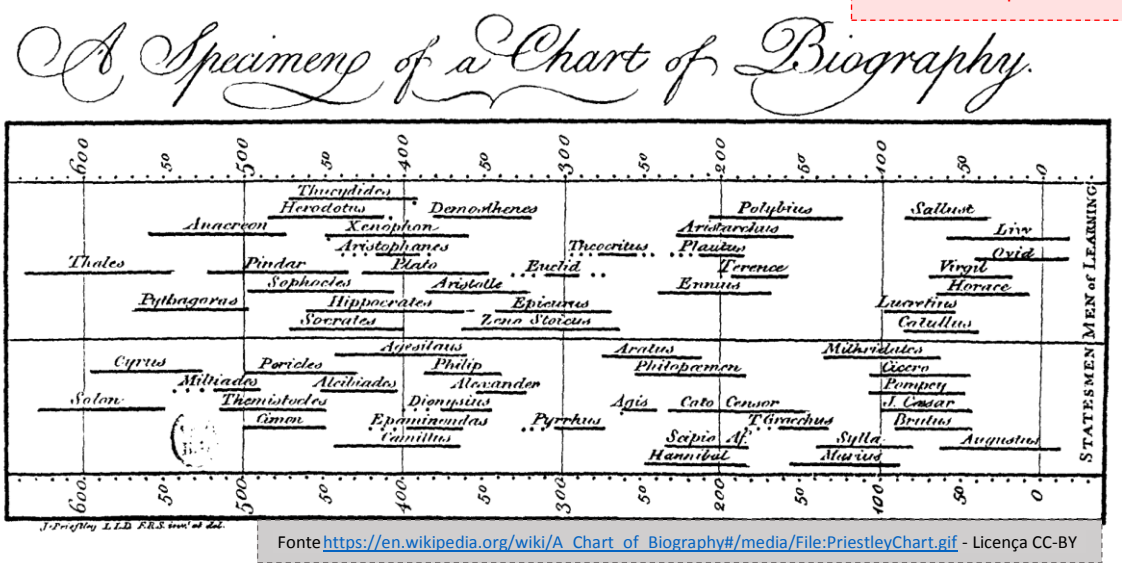
Naquela época, se acreditava que a Lua e o Sol eram também planetas – era o **Geocentrismo**. Durante a inquisição, quem discordasse dessa ideia poderia ser mandado à fogueira – como aconteceu com Giordano Bruno e quase aconteceu com Galileu. Nicolau Copérnico, um dos primeiros da época a contestar o geocentrismo, escondeu por décadas suas ideias do **Heliocentrismo** por medo da Inquisição.



Fonte - H. Gray Funkhouser Osiris Vol. 1 (Jan., 1936), pp. 260-262– Domínio público

Em 1765, Joseph Priestley lança a famosa Chart of Biography, exibindo uma linha do tempo com importantes figuras da história. Note, neste gráfico, a necessidade de representar **intervalos** de tempo:

O gráfico anterior representava observações pontuais (**discretas**) no tempo, ou seja, dados coletados periodicamente – a cada dia ou a cada *n* dias, por exemplo. Intervalos dizem respeito a um tempo **contínuo**.

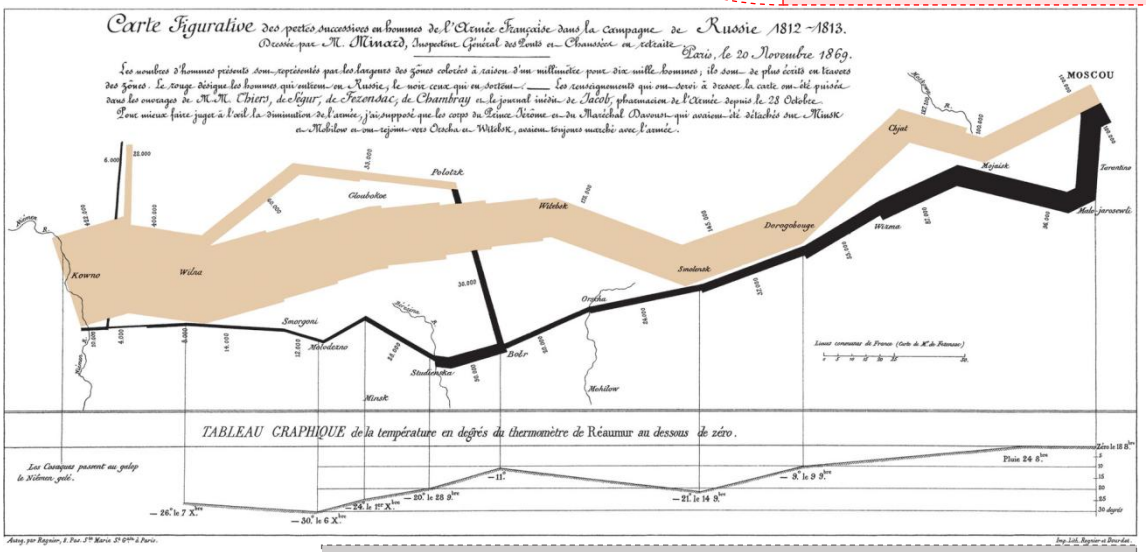


Fonte https://en.wikipedia.org/wiki/A_Chart_of_Biography#/media/File:PriestleyChart.gif - Licença CC-BY

Uma representação visual bastante interessante sobre dados temporais é o conhecido gráfico de Charles Minard, feito em 1869, sobre a desastrosa campanha de Napoleão Bonaparte na tentativa de conquistar a Rússia em 1812. Essa visualização é notável pela representação, em duas dimensões, de

seis dados diferentes: o número de tropas de Napoleão; distância percorrida; latitude e longitude; direção da viagem; localização; e principalmente a temperatura (que teria sido a responsável pela derrota das tropas napoleônicas). Todas em relação a datas específicas.

Esta visualização pode ser considerada também como um gráfico de **fluxo**.

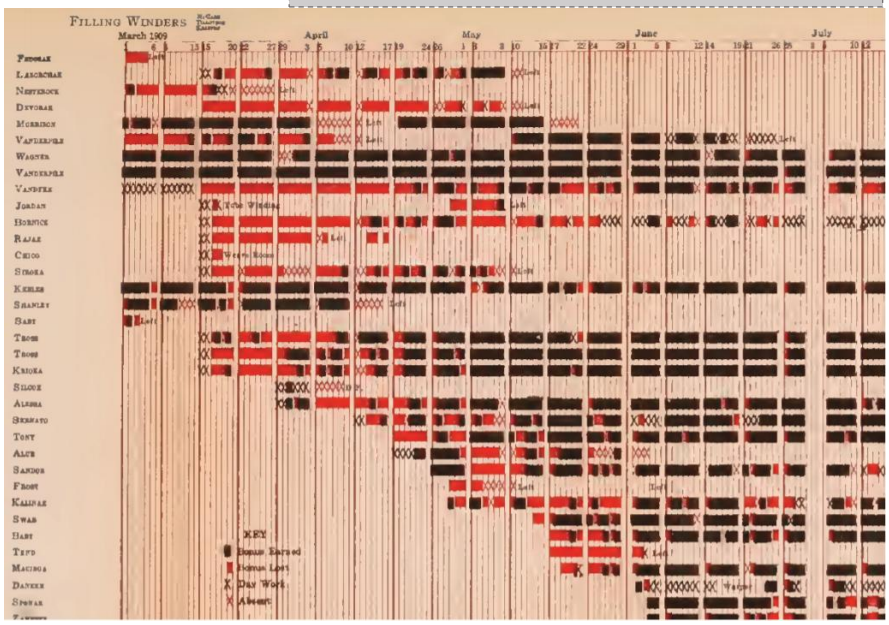


Fonte https://en.wikipedia.org/wiki/Charles_Joseph_Minard#/media/File:Minard.png - Licença CC-BY

O fluxo de soldados, em cor dourada (indo para a invasão) e em preto (bem menor, retornando) permanece como um lembrete do verdadeiro custo de qualquer guerra: as vidas humanas.

Um outro tipo de representação de informação temporal bastante utilizada até hoje para gerenciamento de processos é o Gráfico de Gantt. Criados por Henry Gantt na primeira década do século XX, eles vêm sendo usados para identificar intervalos de tempo associados ao cumprimento de tarefas.

Fonte – H. L. Gantt (1919) "Work, wages, and profits" - Domínio Público



Tipos de dados temporais

Quando tratamos com dados temporais, é importante sabermos classificar os tipos de dados que temos em mãos, para poder obter a visualização mais adequada para cada caso.

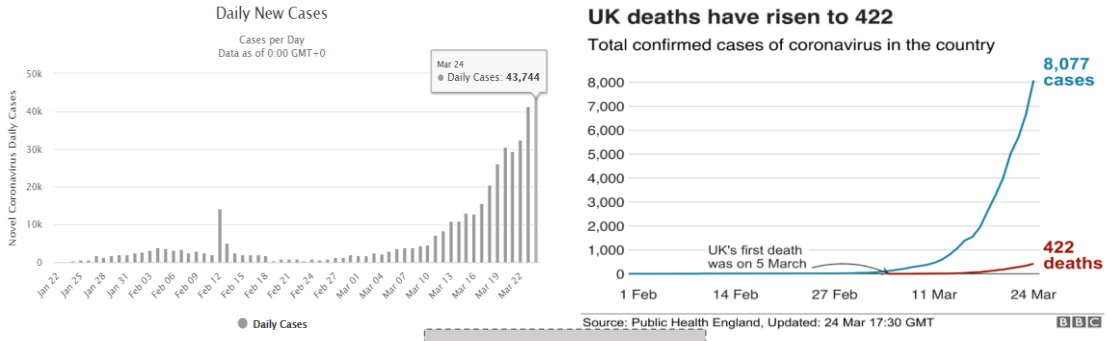
Inicialmente, é importante definir o que é um dado temporal. Consideramos dado temporal todo aquele dado que é alterado com o passar do tempo, ou sobre o qual os aspectos temporais exerçam algum tipo de influência.

Quanto à **escala** do tempo para dados temporais, ela pode ser:

- **Ordinal**: somente a ordem dos dados é conhecida
- **Discreta**: cada elemento tem um antecessor e um sucessor
- **Contínua**: entre quaisquer dois elementos pode haver um terceiro elemento

A escala discreta decorre de dados obtidos de maneira pontual (por exemplo, dados por dia ou por ano), embora ela possa ser apresentada de maneira contínua – essa abordagem é utilizada quando se quer indicar tendências, por exemplo.

Os dois gráficos a seguir representam dados obtidos de maneira discreta (dados diários), mas o gráfico da direita os exibe de forma contínua, de maneira a facilitar a visualização de tendência:



Fontes – World Metrics e BBC

Quanto ao **escopo** dos dados, eles podem ser:

- **Pontuais** – são dados que representam uma data única.
- **Intervalos** – são dados que representam um lapso de tempo, com início e fim.

Um exemplo para discutir o escopo dos dados pode ser visto a seguir, com um recorte de dados relacionados às primeiras corridas de táxi na cidade de Nova Iorque em 2018:

Fonte: Prefeitura de NYC

tpep_pickup_datetime	tpep_dropoff_datetime	fare_amount	passenger_count	trip_distance
2018-01-01 00:00:17	2018-01-01 00:10:55	12	1	3.76
2018-01-01 00:00:16	2018-01-01 00:00:49	55	1	0
2018-01-01 00:00:15	2018-01-01 00:14:17	10.5	1	2.06
2018-01-01 00:00:15	2018-01-01 00:08:21	7	2	1.2
2018-01-01 00:00:14	2018-01-01 00:11:38	14	1	4
2018-01-01 00:00:14	2018-01-01 00:04:32	5	1	0.9
2018-01-01 00:00:13	2018-01-01 00:07:03	6	1	0.9
2018-01-01 00:00:11	2018-01-01 00:06:05	7	1	1.7
2018-01-01 00:00:06	2018-01-01 00:24:34	23.5	1	6.9
2018-01-01 00:00:04	2018-01-01 00:08:13	8	1	1.59
2018-01-01 00:00:04	2018-01-01 00:13:24	13.5	1	3.6
2018-01-01 00:00:03	2018-01-01 00:03:52	5.5	3	0.99
2018-01-01 00:00:03	2018-01-01 00:21:06	20.5	1	6.1
2018-01-01 00:00:02	2018-01-01 00:08:48	7.5	1	1.36
2018-01-01 00:00:00	2018-01-01 00:00:00	27	1	9.14

Os dados apresentados podem ser considerados pontuais, se tomadas as duas primeiras colunas (hora de início e final das corridas de táxi) isoladamente. Porém, as duas colunas tomadas em conjunto podem ser vistas como intervalos.

Quanto ao **arranjo** dos dados temporais, este pode ser:

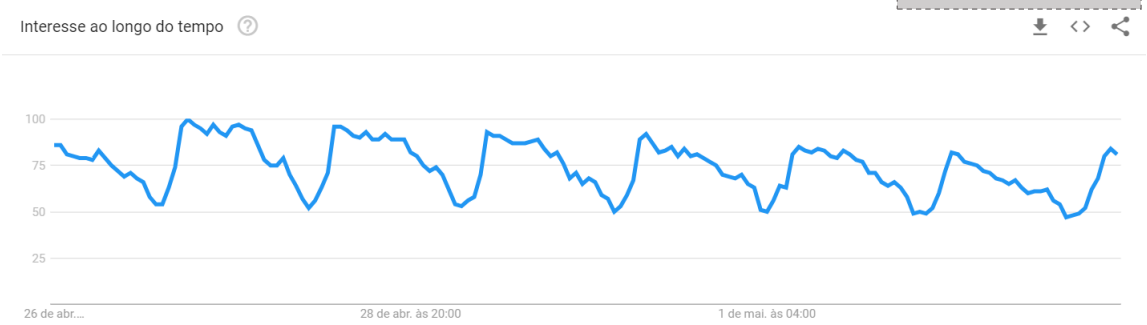
- **Linear** – quando um evento ocorre após o outro.
- **Sazonal** – a sequência temporal é influenciada por fatores sazonais (sempre em um período determinado de tempo). Também chamado de periódico.
- **Cíclico** – quando a sequência de eventos se repete, mas não em um período fixo.

Por exemplo, veja os dados a seguir extraídos do **Google Trends**:

https://trends.google.com
Apresenta gráficos temporais baseados nas buscas recebidas pelo Google.
Os dados em CSV podem ser baixados.

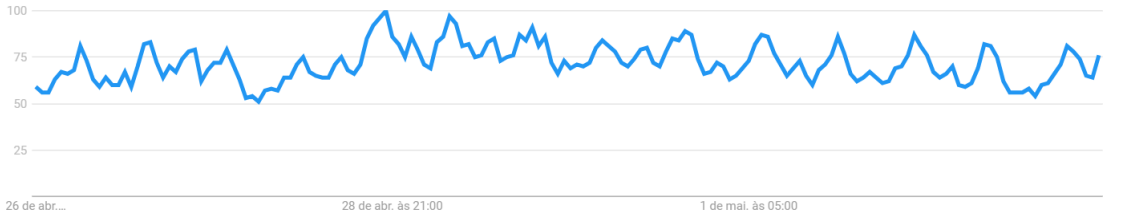
Busca pelo termo “COVID-19” nos EUA num período de 7 dias (26/4/2020 a 2/5/2020). Note que esta parece ser uma **série sazonal**, em que os pontos de mínimo ocorrem pela madrugada a cada dia (ao redor de 4:00 a 5:00).

Fonte: Google Trends



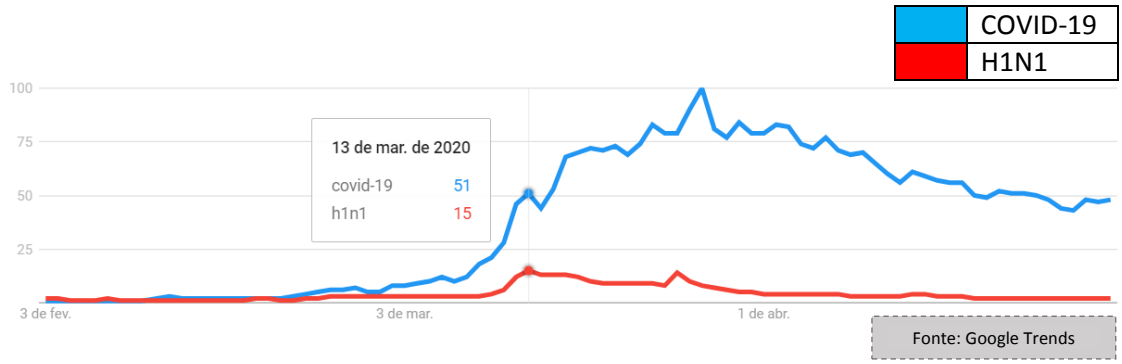
Esse comportamento já não acontece nas buscas pelo mesmo termo (COVID-19) feitas no Brasil, no mesmo período:

Fonte: Google Trends

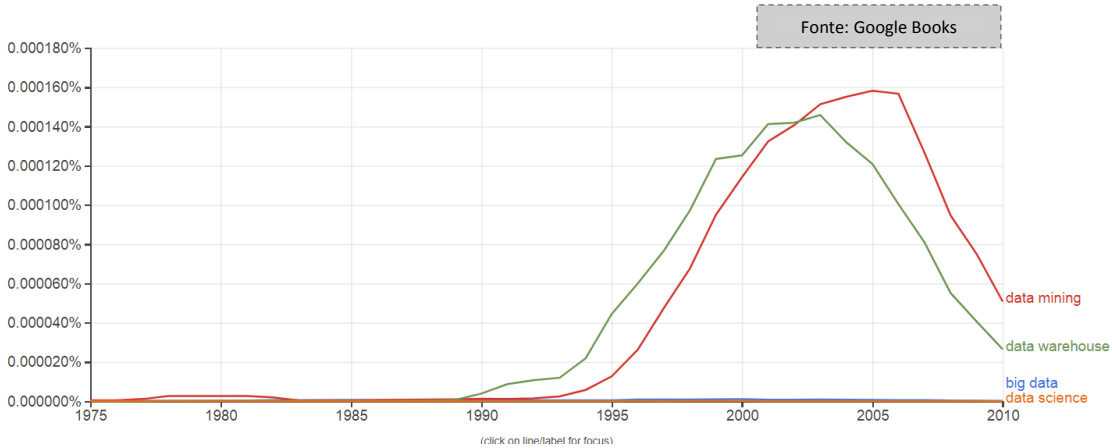


Quando se observa essa série temporal com um *dataset* mais amplo (buscas realizadas num período de 90 dias, a partir de 3/2/2020), percebe-se um comportamento diferente: uma série linear simples, com tendência a ser **crescente** a partir de 29/2/2020 até 13/2/2020 (que coincidiu com um pico de buscas sobre outro termo relacionado a outro vírus, o H1N1. A tendência de alta nas buscas por COVID-19 se mantém, com alguns picos de queda, até atingir o ápice em 27/3/2020, quando reverte a tendência da curva, para uma série **decrescente**.

Esses movimentos coincidem com as datas de notificacão dos primeiros casos do vírus na maior parte dos países, iniciando o movimnto descendente quando o assunto já passa a ser de domínio público, com a pandemia global estabelecida no final de março.



Um outro exemplo de série temporal representada por gráficos de linha pode ser obtida na análise de *n-grams* da base de livros **Google Books**. Uma busca pela presença dos termos “data mining”, “data warehouse”, “big data”e “data science” traz alguns *insights* interessantes: https://books.google.com/ngrams



Essa busca mostra a ocorrência de *n-grams* (termos com n palavras) nos livros cujo conteúdo está disponível pelo Google Books, no período de 1975 a 2010. Vê-se que o termo “data warehouse” assume uma linha de tendência crescente a partir de meados de 1989, enuqanto “data mining” aparece só depois de meados de 1992 (há ocorrências desse termo ao redor do ano 1980, possivelmente com significado distinto do que temos hoje). As curvas ganham padrão descendente a partir de 2002 e 2005. Já os termos “big data” e “data science” ainda não aparecem nessa época.

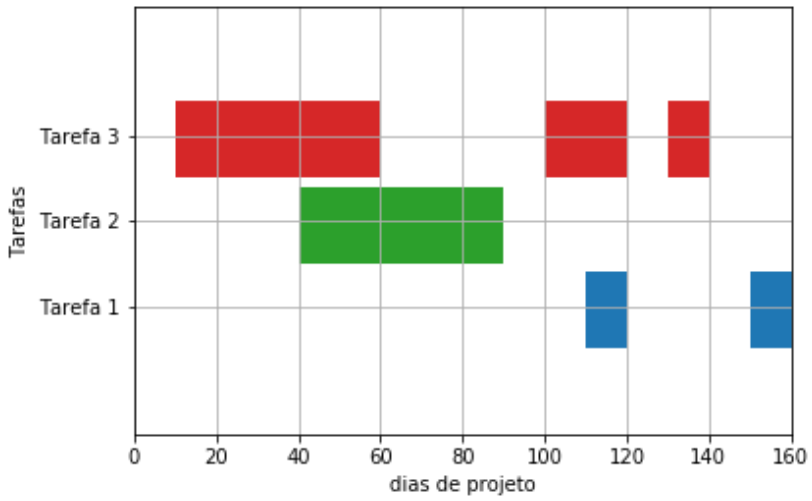
Implementações em Python

Vamos começar mostrando um exemplo de um Diagrama de Gantt feito em Python.

```
import matplotlib.pyplot as plt

fig, gnt = plt.subplots()

gnt.set_ylim(0, 50)
gnt.set_xlim(0, 160)
gnt.set_xlabel('dias de projeto')
gnt.set_ylabel('Tarefas')
gnt.set_yticks([15, 25, 35])
gnt.set_yticklabels(['Tarefa 1', 'Tarefa 2', 'Tarefa 3'])
gnt.grid(True)
gnt.broken_barh([(10, 50), (100, 20), (130, 10)], (30, 9),
                        facecolors =('tab:red'))
gnt.broken_barh([(40, 50)], (20, 9), facecolors =('tab:green'))
gnt.broken_barh([(110, 10), (150, 10)], (10, 9),
                        facecolors ='tab:blue')
```



Note que este exemplo usa basicamente o modulo `pyplot` da biblioteca `matplotlib`. Esta biblioteca é limitada a gráficos mais tradicionais da Estatística Descritiva, não fornecendo suporte direto a visualizações mais elaboradas.

No caso, esse Diagrama de Gantt foi feito usando como subterfúgio o gráfico de barras horizontais descontínuo, por meio do método `broken_barh`. Os dados foram informados de maneira bastante rudimentar, diretamente na chamada deste método.

Já este outro exemplo usa a biblioteca plotly, que facilita bastante a criação de visualizações mais elaboradas do que aquelas fornecidas pela biblioteca `matplotlib`.

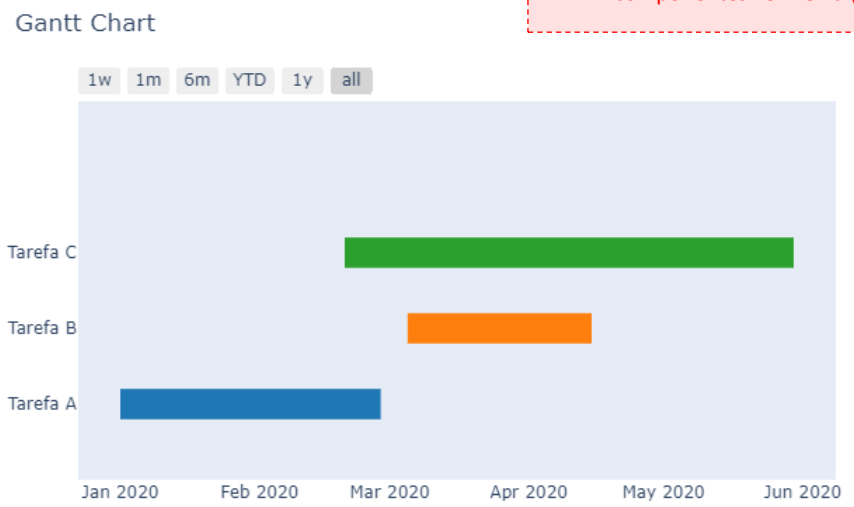
Verifique se a biblioteca está instalada em sua máquina, senão instale-a antes de rodar este e outros exemplos .

```
1. from plotly import figure_factory as ff

2. df = [dict(Task="Tarefa A", Start='2020-01-01', Finish='2020-04-28'),
3. dict(Task="Tarefa B", Start='2020-02-05', Finish='2020-04-15'),
4. dict(Task="Tarefa C", Start='2020-02-20', Finish='2020-05-30')]
5. colors = ['#FF0000', (0.0, 1.0, 0.0), 'rgb(0, 0, 255)']
6. fig = ff.create_gantt(df, colors)
7. fig.show()
```

O exemplo usa o formato: YYYY-MM-DD.

Note que várias formas de estabelecer as cores RGB podem ser usadas. Neste exemplo, a primeira é criada com valores em hexa, a segunda com valores normalizados e a última pela chamada a rgb( ) passando valores de a 255 para as componentes vermelha (R) verde (G) e azul (B).



A biblioteca `plotly`, pelo módulo **`figure_factory`**, gera um Diagrama de Gantt a partir de uma entrada simples, que é um *array* de elementos do tipo *dictionary*, com as entradas `Task`, `Start` e `Finish`), por meio do método `create_gantt`.

Outros valores podem ser informados nos dados de entrada, como por exemplo `Resource` (indicando, por exemplo, um responsável pela tarefa) ou `Complete` (para indicar a % da tarefa já completa).

Para saber mais sobre Diagramas de Gantt usando plotly: https://plotly.com/python/gantt/

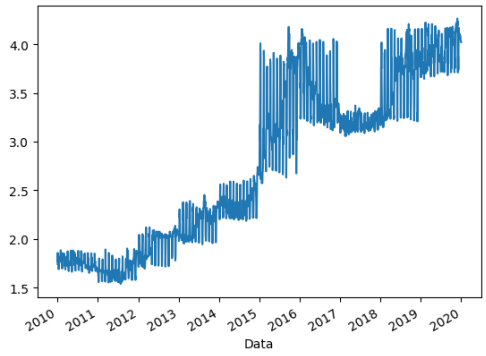
Vamos agora ver um outro exemplo em Pyhton de uma outra representação visual de um série temporal, usando um gráfico de linhas.

Para este exemplo, vamos usar a biblioteca `pandas`, que oferece métodos práticos de leitura de arquivos. No caso, vamos usar o método read_csv para ler o *dataset* contido no arquivo **USD_BRL_hist.csv**.

```
1. from pandas import read_csv
2. from matplotlib import pyplot
3. series = read_csv(r"<informe_sua_pasta>\USD_BRL_hist.csv", header=0,
   index_col=0, parse_dates=True, squeeze=True)
4. series.plot()
5. pyplot.show()
```

É um arquivo (disponível para download no ambiente virtual) contendo as cotações do dólar estadunidense (USD) em Reais (BRL), de 2009 a 2019.
O arquivo tem o seguinte formato:

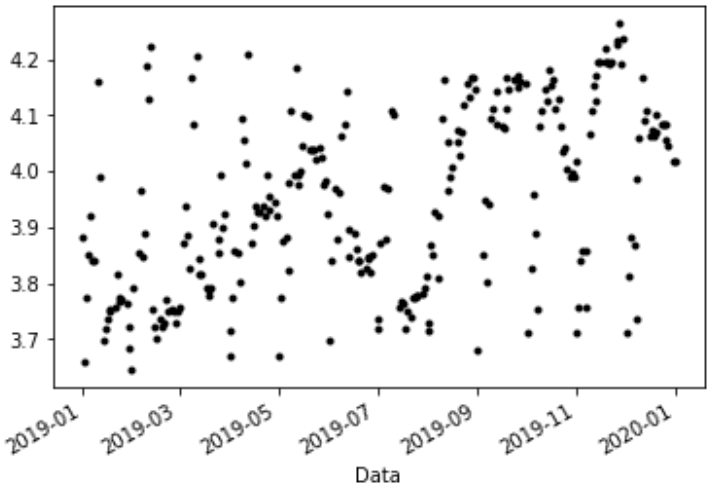
```
"Data","USD_BRL"
"31.12.2019",4.0195
"30.12.2019",4.0195
"27.12.2019",4.046
"26.12.2019",4.056
```



Note que, como a visualização é feita basicamente por meio de um gráfico de linha, o pacote `matplotlib` é suficiente para essa tarefa. Diferentes variações dessa visualização podem ser obtidas, como por exemplo:

```
series2019 = series["01-01-2019":"31-12-2019"]
series2019.plot(style='k.')
pyplot.show()
```

Aqui, extrai-se um subconjunto de dados de 01/01/2019 a 31/12/2019 e altera-se o estilo da plotagem para exibição de pontos (`style='k.'`)



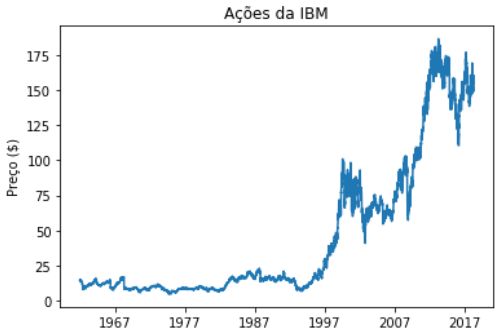
Um exemplo similar (com gráfico de linha para representação visual de dados temporais) pode ser visto a seguir. Neste exemplo, ao invés de carregar as informações a partir de um *dataset* local, iremos consumir um serviço diretamente do código Python.

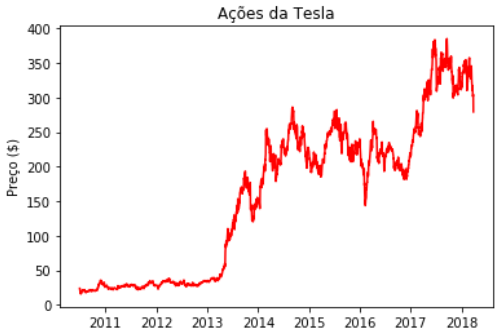
```
1.  import quandl
2.  import pandas as pd
3.  import matplotlib.pyplot as plt
4.  quandl.ApiConfig.api_key = 'Consiga_sua_APIkey'
5.  tesla = quandl.get('WIKI/TSLA')
6.  ibm = quandl.get('WIKI/IBM')
7.  plt.plot(ibm.index, ibm['Adj. Close'])
8.  plt.title('Ações da IBM')
9.  plt.ylabel('Preço ($)');
10.  plt.show()
11.  plt.plot(tesla.index, tesla['Adj. Close'], 'r')
12.  plt.title('Ações da Tesla')
13.  plt.ylabel('Preço ($)')
14.  plt.show()
```

Consiga uma chave gratuita para acessar a API Quandl em http://quandl.com

Verifique se a biblioteca está instalada em seu computador, senão, instale-a.

Neste exemplo, usamos a biblioteca **quandl** para acesso a dados financeiros, no caso, dados de ações em bolsa de valores. Veja os gráficos:





Para saber mais:

https://towardsdatascience.com/time-series-analysis-in-python-an-introduction-70d5a5b1d52a

Cruzeiro do Sul Virtual
Educação a Distância

Para saber mais, leia os capítulos iniciais dos e-books:

PERKOVIC, Ljubomir; VIEIRA, Daniel. Introdução à computação usando Python: um foco no desenvolvimento de aplicações. Rio de Janeiro: LTC, 2016.

McKinney, W. Python Para Análise de Dados: Tratamento de Dados com Pandas, NumPy e IPython. São Paulo: Novatec, 2018