Visualização de Dados em Python

Professora: Lorena Barberia

DCP-USP

lorenabarberia@usp.br

Universidade de São Paulo

Tópicos da Aula

Universidade de São Paulo

Visualização de Dados

- Nesta aula, exploraremos as técnicas de visualização de dados usando a linguagem de programação Python.
- A visualização de dados é uma parte fundamental da análise de dados e comunicação de resultados.

Universidade de São Paulo

Tipos de Gráficos

- Na visualização de dados em Python, você pode criar diversos tipos de gráficos, incluindo:
  - Gráficos de barras
  - Gráficos de dispersão
  - Gráficos de linhas
  - Histogramas
  - Gráficos de pizza
  - Mapas
- Cada tipo de gráfico é adequado para diferentes tipos de dados e objetivos de comunicação.

Universidade de São Paulo

Como escolher sua visualização

- A escolha da visualização correta é crucial para comunicar efetivamente seus dados.
- Considere os seguintes fatores ao escolher o tipo de gráfico:
  - Tipo de dados: Variáveis categóricas ou numéricas?
  - Objetivo: Mostrar tendências, distribuição, comparação, etc.
  - Público-alvo: Quem são seus espectadores? Qual é o nível de detalhe necessário?

Universidade de São Paulo

Como escolher sua visualização

- Alguns exemplos:
  - Use gráficos de barras para comparar categorias.
  - Use gráficos de dispersão para mostrar relações entre variáveis numéricas.
  - Use gráficos de linhas para representar tendências ao longo do tempo.
  - Use histogramas para visualizar a distribuição de dados numéricos.

Universidade de São Paulo

Data to Viz

Universidade de São Paulo

Recursos

- **Data-to-viz** → Dicas de visualização para cada tipo de dado
- **Colorbrewer** → Escolha de paleta de cores (sequencial, divergente, qualitativo).
- **Material Design** (Data Visualization) → Princípios a se seguir na visualização de dados
- Livros específicos sobre visualização de dados (com ou sem linguagens de programação).

Universidade de São Paulo

Escolha com cuidado

- Escolher a melhor visualização nem sempre é simples
- Além disso, questões estéticas e de estilo nos gráficos só são desenvolvidas com a prática. Seu primeiro gráfico não vai ser o mais bonito.

Universidade de São Paulo

Matplotlib: Biblioteca de Visualização

- Matplotlib é a principal biblioteca de visualização de dados em Python, com outras bibliotecas sendo construídas sobre ele.
- Matplotlib foi originalmente escrito por John D. Hunter em 2002. Sua interface foi baseada no MATLAB e desde então, tem tido uma comunidade de desenvolvimento ativa, e sua influência perpassa outras bibliotecas.
- Principais recursos incluem suporte para gráficos de barras, gráficos de dispersão, gráficos de linha e muito mais.

Universidade de São Paulo

Figuras e Subfiguras

- Matplotlib segue o conceito de uma figura (figure) contendo um ou mais subplots (subfiguras).
- A figura é a área onde os gráficos são desenhados, enquanto os subplots representam os próprios gráficos.
- É possível criar vários subplots para criar gráficos complexos em uma única figura.

Universidade de São Paulo

Estilos de Programação em Matplotlib

- Matplotlib oferece duas abordagens principais para criar gráficos: **explícita** e **implícita**.

1. Estilo Explícito:

- No estilo explícito, cada elemento do gráfico é especificado separadamente.
- Você tem controle total sobre cada detalhe do gráfico.
- É útil quando se deseja personalização completa.
- Exige mais código para criar gráficos.

Universidade de São Paulo

Estilo Implícito

2. Estilo Implícito:

- No estilo implícito, você cria gráficos de maneira mais simplificada.
- Matplotlib lida automaticamente com muitos detalhes.
- É conveniente para criação rápida de gráficos simples.
- Menos controle detalhado sobre elementos individuais.

Escolha o estilo que melhor atende às suas necessidades e preferências ao criar visualizações de dados com Matplotlib.

Universidade de São Paulo

Customizações

- Na aula de hoje, iremos ver muitos customizações possíveis no Matplotlib (Cores, marcadores, legendas, títulos, etc.). Essas são bem importantes pois serão as mesmas utilizadas em gráficos com outras bibliotecas, como o Pandas e o Seaborn.
- Portanto, se torna importante fixar bem os conceitos de Matplotlib, pois eles serão a base para a maior parte das visualizações que você fizer em Python

Universidade de São Paulo

Visualizações em Pandas

- O Pandas oferece suporte para visualizações simples de dados, e seus métodos e funções foram criados em cima do Matplotlib
- Essas visualizações são úteis para a análise exploratória e inicial de seus conjuntos de dados.
- Vamos explorar algumas das funções do Pandas para criar visualizações.

Universidade de São Paulo

Visualização Básica de Dados em Pandas

- Você pode usar o método ‘plot()‘ em um DataFrame do Pandas para criar visualizações básicas.
- Alguns tipos de gráficos que você pode criar incluem:
  - Gráfico de linha
  - Gráfico de barras
  - Gráfico de dispersão
  - Histograma
- Por exemplo, ‘df[’coluna’].plot(kind=’hist’)‘ cria um histograma da coluna escolhida.

Universidade de São Paulo

Visualizações em Pandas

- No laboratório serão apresentados algumas das principais visualizações presentes na biblioteca Pandas
- Todas as regras que vocês observarem no Matplotlib valem para os gráficos em Pandas, já que as visualizações geradas por este são objetos idênticos ao do Matplotlib.

Universidade de São Paulo

Seaborn

- Seaborn é uma biblioteca de visualização de dados em Python baseada no Matplotlib.
- É amplamente utilizada para criar gráficos estatísticos de alta qualidade.
- Fácil de usar e fornece um alto nível de abstração para a criação de gráficos complexos com facilidade.
- Possui estilos predefinidos atraentes que melhoram a aparência dos gráficos.

Universidade de São Paulo

Seaborn x Matplotlib

- O Seaborn é uma biblioteca de visualização de dados em Python que é construída sobre o Matplotlib.
- O Matplotlib fornece a base para a criação de gráficos em Python, enquanto o Seaborn simplifica e aprimora a criação de gráficos estatísticos.
- O Seaborn é uma camada de alto nível que abstrai muitos detalhes do Matplotlib, tornando mais fácil a criação de gráficos complexos.
- Você pode usar o Matplotlib em conjunto com o Seaborn para personalizar ainda mais seus gráficos, se necessário.

USP Universidade de São Paulo

Agregação de Dados e Operações em Grupos

- Na análise de dados, muitas vezes precisamos resumir informações de acordo com certos critérios.
- Pandas oferece recursos poderosos para realizar operações em grupos de dados.
- Isso é útil para criar resumos estatísticos, agregações e análises específicas.

Universidade de São Paulo

Split, Apply e Combine

- O ”Split, Apply e Combine” é um padrão amplamente utilizado em análise de dados.
- É uma abordagem para dividir um conjunto de dados, aplicar uma operação a cada grupo e, em seguida, combinar os resultados.

Universidade de São Paulo

Etapas do SAC

1. **Split (Dividir)**: Divide o conjunto de dados em grupos com base em critérios específicos. No Pandas, isso é feito usando ‘groupby()‘.
2. **Apply (Aplicar)**: Aplica uma operação a cada grupo separadamente. Pode ser uma função de agregação, transformação ou uma função personalizada.
3. **Combine (Combinar)**: Combina os resultados das operações em grupos de volta em um único conjunto de dados.

Universidade de São Paulo

Etapas do SAC

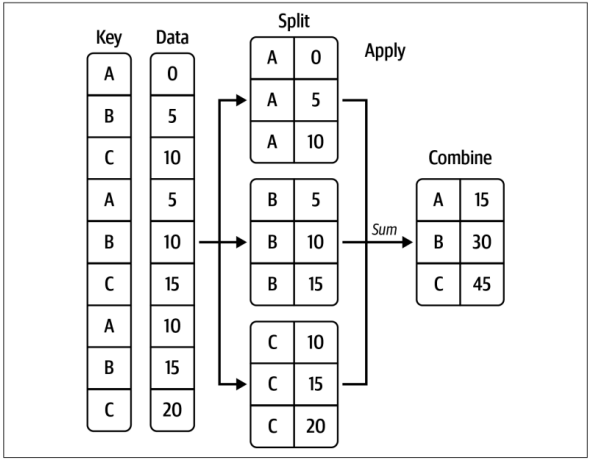

Figure 10-1. Illustration of a group aggregation

Universidade de São Paulo

Exemplo de Split, Apply e Combine em Pandas

Aqui está um exemplo de código Python que demonstra o "Split, Apply e Combine" em Pandas:

```
import pandas as pd

# Criar DataFrame de exemplo
data = {'Grupo': ['A', 'B', 'A', 'B', 'A'],
        'Valores': [10, 15, 7, 12, 8]}
df = pd.DataFrame(data)

# Dividir por 'Grupo', aplicar a média e combinar
resultados = df.groupby('Grupo')['Valores'].mean()
```

Universidade de São Paulo

O Método '.groupby()' no Pandas

- O método '.groupby()' é uma funcionalidade essencial no Pandas para dividir um conjunto de dados em grupos com base em critérios específicos.
- É o primeiro passo no paradigma "Split, Apply e Combine" usado para análise de dados.
- A sintaxe básica do método '.groupby()' é a seguinte:

```
df.groupby('coluna')
```

- onde "coluna" é a coluna pela qual você deseja agrupar seus dados.

Universidade de São Paulo

Operações em Grupos

Após criar grupos usando '.groupby()', você pode realizar várias operações, como:

- Agregação: Calcular estatísticas como média, soma, mínimo e máximo.
- Transformação: Aplicar transformações aos dados em cada grupo.
- Filtragem: Filtrar grupos com base em condições específicas.
- Aplicação de Funções Personalizadas aos grupos.

Universidade de São Paulo

Novas possibilidade de Visualizações

- Utilizar métodos de agregações e operações em grupos abre caminhos para múltiplas visualizações diferentes com os mesmos dados
- Agregar, transformar e fazer diversas operações também permite novas inferências e hipóteses.
- Portanto, ter uma boa ideia de como fazer essas operações é essencial.

Universidade de São Paulo

Considerações Finais

- A aula de hoje buscou dar uma visão geral sobre os pontos principais da visualização de dados e operações em grupo. Na próxima aula, finalmente entraremos no assunto principal da disciplina: O aprendizado de máquina.
- No entanto, o aprendizado de Python não acaba agora, sendo constante. Também retomaremos e abordaremos alguns assuntos de programação que não foram tratados aqui, quando necessário.

Universidade de São Paulo

Laboratório

Agora, vocês irão se reunir em grupos e trabalharão no laboratório da aula de hoje (Lab_Aula5.ipynb). Os laboratórios estão na nossa pasta do drive. Pedimos que façam uma cópia do arquivo em uma subpasta de acordo com o seu turno. Nomeiem essa subpasta de acordo com o grupo (sobrenome1_sobrenome2_lab1) e coloquem as informações dos alunos em uma célula de *Markdown* no início do laboratório (Nome, NUSP, curso e se é da graduação ou pós).

USP Universidade de São Paulo

Dúvidas?

Universidade de São Paulo