

Fundamentos de Python (UFCD 10793)

PPT01

Sandra Liliana Meira de Oliveira



Objetivos Gerais

1. **Compreender a importância da visualização de dados** na análise e comunicação de informação.
2. **Importar e preparar dados com Python** para posterior visualização.
3. **Aplicar boas práticas na criação de representações gráficas** (clareza, tipo adequado de gráfico, leitura fácil, foco no público).
4. **Criar visualizações estáticas e interativas** utilizando bibliotecas especializadas como:
 1. Matplotlib
 2. Seaborn
 3. Folium
 4. Plotly/Dash
5. **Visualizar dados espaciais em mapas** utilizando Folium e GeoJSON.
6. **Desenvolver dashboards simples e eficazes** com ferramentas de visualização.
7. **Analisar e interpretar visualizações** para apoiar a tomada de decisão.
8. **Apresentar resultados de forma clara e estruturada**, usando storytelling com dados.

Conteúdos a Abordar por Categoria

1. Fundamentos da Visualização

- Conceito e importância
- Comunicação com dados
- Tipos de gráficos e quando usá-los
- Boas práticas de visualização

2. Ferramentas de Visualização com Python

- Introdução a Jupyter Notebooks
- Bibliotecas: Matplotlib, Seaborn, Plotly, Folium, Dash
- Integração com Pandas

Conteúdos a Abordar por Categoria

3. Gráficos com Matplotlib

- Gráficos de linha, barra, histograma
- Pie charts, scatter plots, boxplots
- Personalização de gráficos (título, eixos, legenda, cores)

4. Visualizações com Seaborn

- Regressão linear (regplot)
- Gráficos categóricos (boxplot, violinplot, countplot)
- Heatmaps e correlações

Conteúdos a Abordar por Categoria

5. Visualizações Avançadas

- Waffle Charts (categorias proporcionais)
- Word Clouds (visualização textual)

6. Geovisualização com Folium

- Mapas com marcadores
- MarkerCluster e FeatureGroups
- Mapas coropléticos (Choropleth) com dados GeoJSON

7. Dashboards Interativos

- Introdução ao Dash e Streamlit
- Conceitos de interatividade com callbacks
- Criação de painéis com múltiplos gráficos

Ferramentas

Linguagem e Ambiente de Trabalho

- Python 3.x – Linguagem base para programação e visualização de dados.
- Jupyter Notebook – Ambiente interativo para análise, gráficos e documentação.
- Visual Studio Code
- GitHub

Ferramentas

Bibliotecas de Análise de Dados

- Pandas – Manipulação de dados, tabelas, ficheiros e séries temporais.
- NumPy – Arrays e cálculos numéricos.

Ferramentas

Bibliotecas de Visualização Gráfica

- Matplotlib – Gráficos estáticos personalizáveis.
- Seaborn – Gráficos estatísticos.
- PyWaffle – Waffle charts para proporções.
- WordCloud – Nuvens de palavras.

Ferramentas

Visualização Geoespacial

- Folium – Mapas interativos com Leaflet.js.
- GeoJSON – Regiões para mapas coropléticos.

Ferramentas

Visualização Geoespacial

- Folium – Mapas interativos com Leaflet.js.
- GeoJSON – Regiões para mapas coropléticos.

Ferramentas

Dashboards e Interatividade

- Plotly – Gráficos interativos.
- Dash – Dashboards com componentes HTML.
- Streamlit – Dashboards rápidos com Python.
- Voila – Jupyter como app interativa.

Ferramentas

Outros Utilitários

- `openpyxl` – Leitura de ficheiros Excel (.xlsx).
- `%matplotlib inline / notebook` – Para visualização no Jupyter.
- ...

Bibliografia e recursos on-line

Python e Ambientes

- **Python:** <https://www.python.org/>
- **Visual Studio Code:**
<https://code.visualstudio.com/download>
- **Jupyter Notebook:** <https://jupyter.org/>
- **GitHub:** <https://github.com/>

Bibliografia e recursos on-line

Análise de Dados

- **Pandas:** <https://pandas.pydata.org/>
- **NumPy:** <https://numpy.org/>

Visualização Gráfica

- **Matplotlib:** <https://matplotlib.org/>
- **Seaborn:** <https://seaborn.pydata.org/>
- **PyWaffle:** <https://github.com/gyli/PyWaffle>
- **WordCloud:** https://github.com/amueller/word_cloud

Bibliografia e recursos on-line

Visualização Geoespacial

- **Folium:** <https://python-visualization.github.io/folium/>
- **GeoJSON:** <https://geojson.org/>

Dashboards e Interatividade

- **Plotly:** <https://plotly.com/python/>
- **Dash (by Plotly):** <https://dash.plotly.com/>
- **Streamlit:** <https://streamlit.io/> (*opcional*)
- **Voila:** <https://voila.readthedocs.io/>

Bibliografia e recursos on-line

Utilitários e Suporte

- **openpyxl**: <https://openpyxl.readthedocs.io/>
- **IPython Magic Commands** (%matplotlib inline, etc.):
<https://ipython.readthedocs.io/en/stable/interactive/magics.html>

Manual da Ação

Secção própria na Plataforma moodle.

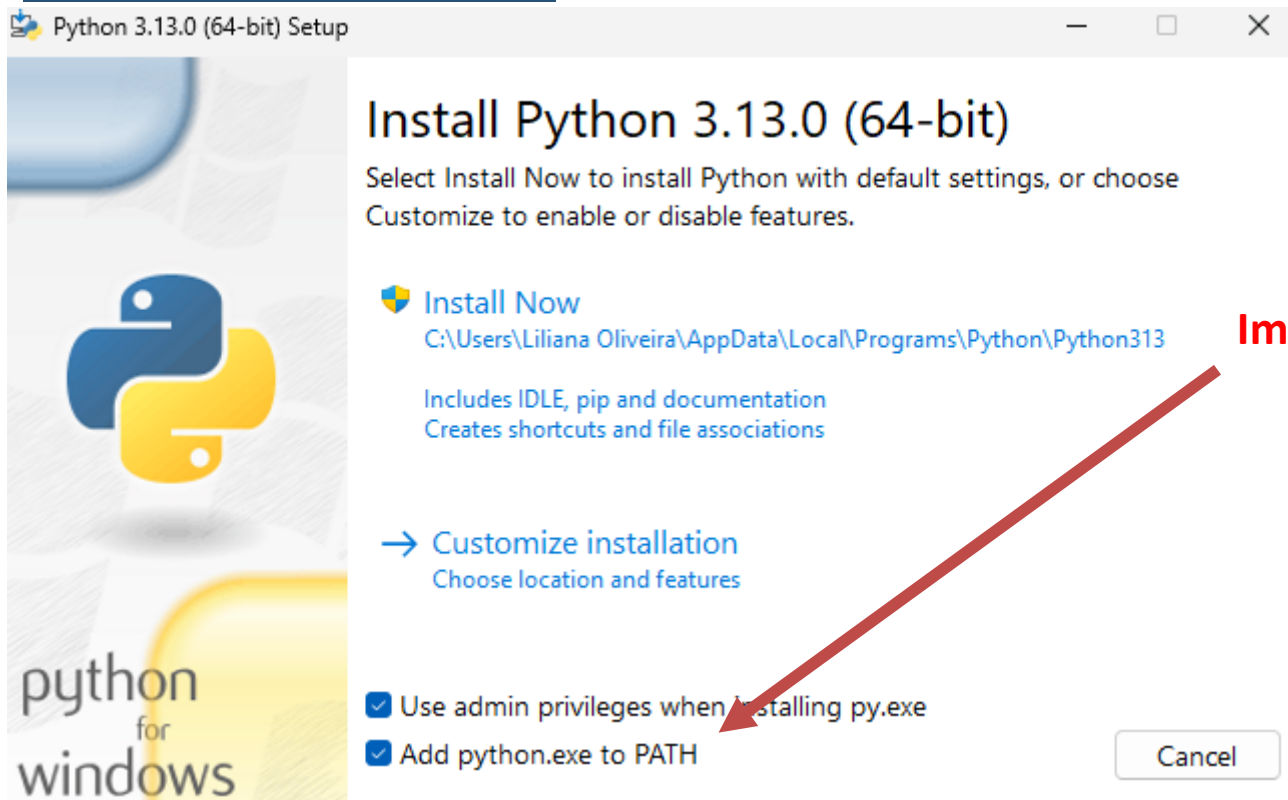
Instalar Ferramentas

Atividade 01

Instalação e Configuração – Python 3

<https://www.python.org/downloads/>

Download Python 3.13.0



Importante

Atividade 02 – o meu primeiro código

1. Abre a linha de comandos;
2. Escreve **python**
3. Pressiona a **Tecla enter**
4. Deverás obter:

```
C:\WINDOWS\system32>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

5. Escreve:

```
print("Hello Python interpreter!")
```

6. Para sair do interpretador python escreve **exit()** e pressiona a tecla **enter** de seguida.

```
C:\WINDOWS\system32>python
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr  4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello Python interpreter!")
Hello Python interpreter!
>>> exit()

C:\WINDOWS\system32>
```

Visual Studio Code

- O VS Code é um IDE poderoso e de qualidade profissional que é gratuito e amigável para iniciantes.
- Pode ser utilizado em projetos simples e complexos
- Podemos continuar a usá-lo à medida que avançamos para projetos maiores e mais complicados.
- O VS Code pode ser instalado em todos os sistemas operativos modernos e suporta a maioria das linguagens de programação, incluindo Python.

Atividade 03

Instalação e Configuração – Visual Studio Code

<https://code.visualstudio.com/>

Download for Windows

Stable Build



Select Additional Tasks

Which additional tasks should be performed?



Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.

Additional icons:

☒ Create a desktop icon

Other:

☒ Add "Open with Code" action to Windows Explorer file context menu

☒ Add "Open with Code" action to Windows Explorer directory context menu

☒ Register Code as an editor for supported file types

☒ Add to PATH (requires shell restart)


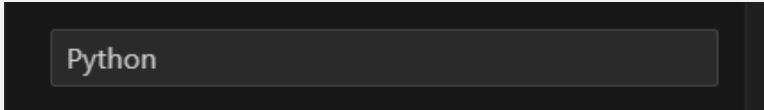
Atividade 04

Instalar extensões python no VS Code

O VS Code trabalha com muitas linguagens de programação diferentes;

para tirar o máximo proveito dele como um programador Python, temos que instalar a extensão Python.

Esta extensão adiciona suporte para escrever, editar e executar programas Python.

1. Abra o Visual Studio Code;
2. Pressiona o ícone 
3. Escreve python na caixa de pesquisa 
4. Clica na extensão Python.
5. Segue os passos anteriores para instalar as extensões jupyter;
6. Cria uma conta (caso ainda não tenhas uma no github on-line)

Atividade 05 – o comando pip

O comando **pip** (acrónimo para "Pip Installs Packages" or "Pip Installs Python") gere a instalação de pacotes no Python.

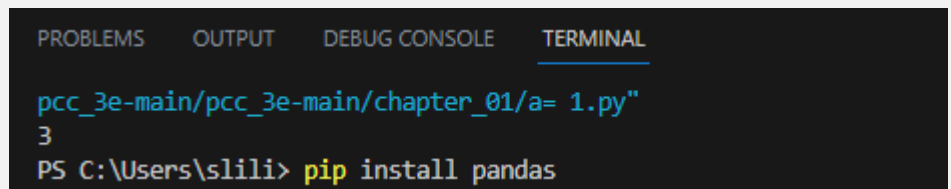
É usado para instalar e atualizar pacotes.

<https://pip.pypa.io/en/stable/>

https://www.w3schools.com/python/python_pip.asp

1. Abre o Visual Studio Code;
2. Clica em “Terminal” no menu do topo;
3. Seleciona a opção “New Terminal”;
4. Para instalar pacotes basta que, no terminal, se escreva `pip install <nome_do_pacote a instalar>`. Deste modo, no terminal criado anteriormente escreve o commando:

pip install pandas



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
pcc_3e-main/pcc_3e-main/chapter_01/a= 1.py"
3
PS C:\Users\slili> pip install pandas
```


Atividade 06 – Repositorios Github

... vamos configurar 😊

Visualização de Dados – Visão Geral

Visualização de Dados

A visualização de dados desempenha um papel crucial na análise e comunicação de informação.

Desde o contexto acadêmico ao mundo empresarial, gráficos bem concebidos e adequadamente contextualizados ajudam a transformar números em insights claros, apoiando a tomada de decisão.

Fundamentos da Comunicação Visual – Cor, Gestalt, Percepção e Storytelling com Dados

Teoria da Cor e Paletas Cromáticas

A cor é um dos aspectos mais poderosos na comunicação visual, ajudando a destacar padrões, classificar categorias e criar hierarquias de informação. Contudo, o uso descuidado da cor pode confundir o leitor e distorcer a mensagem.

- **Cores Primárias, Secundárias e Complementares:**

O círculo cromático ajuda a compreender a relação entre cores. Por exemplo, cores complementares (azul e laranja) criam contraste forte, enquanto cores análogas (azul e verde) produzem harmonia suave.

- **Cores para Dados Categóricos, Contínuos e Divergentes:**

- **Categóricas:** Paletas que atribuem cores claramente distintas a cada categoria.

- **Contínuas:** Gradientes que mostram a progressão de um valor, por exemplo, do claro para o escuro.

- **Divergentes:** Indicadas para dados com um ponto médio, utilizando cores que divergem a partir de um tom neutro (ex.: azul para valores negativos e vermelho para positivos, com branco ao centro).

- **Acessibilidade e Daltonismo:**

É importante garantir que a paleta seja distinguível por pessoas com diferentes tipos de visão. Paletas “colorblind friendly” reduzem a probabilidade de confusão.

Fundamentos da Comunicação Visual – Cor, Gestalt, Percepção e Storytelling com Dados

Princípios da Gestalt Aplicados à Visualização de Dados

A Psicologia da Gestalt estuda como o nosso cérebro organiza a informação visual. Aplicar estes princípios ao design de gráficos ajuda a criar visualizações mais intuitivas:

- **Proximidade:** Elementos próximos tendem a ser percebidos como um grupo.
- **Semelhança:** Semelhanças em cor, forma ou tamanho indicam relação entre elementos.
- **Continuidade:** O olho segue linhas e curvas suaves, tornando gráficos de linha mais fáceis de ler.
- **Figura-Fundo:** Destacar o objeto principal do fundo, evitando ruído visual.

Fundamentos da Comunicação Visual – Cor, Gestalt, Percepção e Storytelling com Dados

Percepção Visual e Limites Humanos

O olho humano é sensível a certos padrões, mas tem limitações:

- Diferenças muito subtis de cor ou textura podem passar despercebidas.
- O excesso de informação visual pode levar a sobrecarga cognitiva.
- É importante simplificar, rotular corretamente e apresentar a informação de forma clara.

Storytelling com Dados

A visualização é comunicação. O storytelling com dados significa estruturar a informação de modo a conduzir o leitor a um insight:

- **Contexto:** Títulos descritivos, legendas, anotações e referências temporais tornam o gráfico mais compreensível.
- **Narrativa:** Apresentar os dados numa sequência lógica.
- **Destaques:** Chamar a atenção para pontos-chave, como picos, variações súbitas ou tendências relevantes.

Fundamentos da Comunicação Visual – Cor, Gestalt, Percepção e Storytelling com Dados

Aprofundar o Conhecimento sobre Design de Informação, Análise Perceptual e Escolha de Paletas de Cor

Ir além do básico implica compreender design de informação (estudar obras de Edward Tufte ou Colin Ware), psicofísica da percepção (o que o olho distingue melhor ou pior) e ferramentas para escolha de paletas (como “ColorBrewer”).

Assim, poderás tomar decisões informadas, escolhendo o tipo de gráfico, paletas cromáticas e níveis de detalhe ideais para o contexto e o público-alvo.

Fundamentos da Comunicação Visual – Cor, Gestalt, Percepção e Storytelling com Dados

Aprofundar o Conhecimento sobre Design de Informação, Análise Perceptual e Escolha de Paletas de Cor

Ir além do básico implica compreender design de informação (estudar obras de Edward Tufte ou Colin Ware), psicofísica da percepção (o que o olho distingue melhor ou pior) e ferramentas para escolha de paletas (como “ColorBrewer”).

Assim, poderás tomar decisões informadas, escolhendo o tipo de gráfico, paletas cromáticas e níveis de detalhe ideais para o contexto e o público-alvo.

Tipos de Gráficos e as Suas Aplicações

2.1 Gráficos de Linhas

- **Aplicação:** Mostrar tendências ao longo do tempo ou de um eixo contínuo (ex.: vendas mensais, evolução de temperatura).
- **Melhor Prática:** Usar marcadores nos pontos-chave se necessário, cores contrastantes para múltiplas séries e legendas claras.

2.2 Gráficos de Barras

- **Aplicação:** Comparar categorias discretas (ex.: vendas por produto, número de queixas por tipo).
- **Melhor Prática:** Ordenar as categorias de forma lógica, usar cores distintas, adicionar rótulos nos eixos.

2.3 Histogramas

- **Aplicação:** Mostrar a distribuição de uma variável contínua (ex.: alturas de uma população, notas de um exame).
- **Melhor Prática:** Escolher um número de “bins” adequado, usar cores suaves e adicionar grelha se necessário.

Tipos de Gráficos e as Suas Aplicações

2.4 Gráficos de Pizza (Circulares)

- **Aplicação:** Mostrar proporções de um todo.
- **Melhor Prática:** Usar com moderação, realçar apenas uma fatia importante, incluir percentagens.

2.5 Gráficos de Dispersão (Scatter Plots)

- **Aplicação:** Visualizar a relação entre duas variáveis (ex.: altura vs peso, preço vs procura).
- **Melhor Prática:** Ajustar a transparência se houver sobreposição, usar cores ou tamanhos diferenciados para mostrar uma terceira dimensão.

2.6 Boxplots e Violinos

- **Aplicação:** Comparar distribuições estatísticas entre grupos (ex.: rendas em diferentes cidades).
- **Melhor Prática:** Legendar cada grupo, cores neutras, destacar medianas e outliers.

Tipos de Gráficos e as Suas Aplicações

2.7 Mapas de Calor e Contornos

- **Aplicação:** Visualizar valores numa grelha (ex.: matriz de correlações, intensidade numa imagem).
- **Melhor Prática:** Usar paleta contínua ou divergente adequada, adicionar colorbar e rótulos claros nos eixos.

2.8 Gráficos 3D

- **Aplicação:** Dados espaciais, topografias, superfícies matemáticas complexas.
- **Melhor Prática:** Evitar ângulos de visualização pouco claros, usar cores adequadas e considerar se uma projecção 2D não seria mais eficaz.

Pandas - Revisão

1. Criação e Estruturas de Dados

- **pd.Series():** Cria uma série unidimensional rotulada (similar a um array com índice).
- **pd.DataFrame():** Cria um DataFrame bidimensional (estrutura tabular com linhas e colunas).
- **pd.read_csv() / pd.to_csv():** Lê/escreve dados de/para ficheiros CSV.
- **pd.read_excel() / pd.to_excel():** Lê/escreve dados de/para ficheiros Excel.
- **pd.read_sql() / pd.to_sql():** Lê/escreve dados de/para uma base de dados SQL.

Pandas - Revisão

2. Visualização e Exploração

- **df.head():** Mostra as primeiras linhas do DataFrame.
- **df.tail():** Mostra as últimas linhas do DataFrame.
- **df.info():** Apresenta um resumo das colunas, tipos de dados e valores nulos.
- **df.describe():** Gera estatísticas descritivas (média, mediana, percentis, etc.) de colunas numéricas.

Pandas - Revisão

3. Seleção e Indexação

- **df['coluna']**: Seleciona uma coluna.
- **df.loc[]**: Seleção baseada em rótulos (nomes de linhas/colunas).
- **df.iloc[]**: Seleção baseada em índices numéricos.
- **df.set_index()**: Define uma coluna como índice.
- **df.reset_index()**: Redefine o índice para numeração padrão.

Pandas - Revisão

4. Manipulação de Dados

- **df.sort_values():** Ordena linhas com base em valores de colunas.
- **df.sort_index():** Ordena com base nos índices.
- **df.drop():** Remove linhas ou colunas.
- **df.rename():** Renomeia colunas ou índices.
- **df.fillna():** Preenche valores ausentes.
- **df.dropna():** Remove linhas ou colunas com valores ausentes.

Pandas - Revisão

5. Agregação e Agrupamento

- **df.groupby():** Agrupa dados para aplicar funções de agregação (soma, média, etc.).
- **df.agg():** Aplica múltiplas funções de agregação.
- **df.pivot_table():** Cria tabelas dinâmicas.
- **df.cumsum():** Calcula a soma cumulativa de colunas numéricas.

Pandas - Revisão

6. Filtros e Condições

- **`df[df['coluna'] > valor]`**: Filtra linhas com base numa condição.
- **`df.query('coluna > valor')`**: Aplica filtros usando uma string de consulta.
- **`df.isnull()` / `df.notnull()`**: Identifica valores nulos/não nulos.

7. Estatísticas e Análise

- **`df.mean()`**: Calcula a média de colunas numéricas.
- **`df.sum()`**: Calcula a soma de colunas ou linhas.
- **`df.corr()`**: Calcula a correlação entre colunas numéricas.
- **`df.value_counts()`**: Conta valores únicos numa coluna.

Pandas - Revisão

8. Manipulação de Strings

- **`df['coluna'].str.lower() / .str.upper()`**: Converte texto para minúsculas/maiúsculas.
- **`df['coluna'].str.contains('texto')`**: Filtra linhas que contêm determinado texto.
- **`df['coluna'].str.replace('a', 'b')`**: Substitui substrings.

9. Trabalhando com Datas

- **`pd.to_datetime()`**: Converte strings para objetos datetime.
- **`df['coluna'].dt.year`**: Extrai o ano de uma coluna datetime.
- **`df['coluna'].dt.dayofweek`**: Obtém o dia da semana.

Pandas - Revisão

10. Visualização com Pandas

- **`df.plot(kind='line')`**: Gera gráficos de linha.
- **`df.plot(kind='bar')`**: Gera gráficos de barras.
- **`df.plot(kind='hist')`**: Gera histogramas.

FT02 e FT03