# Python e seu ecossistema

## Foco: análise de dados

## Introdução a biblioteca Pandas:

- O **Pandas** é uma **biblioteca** construída sobre o NumPy, **focada no manuseio e análise de dados tabulares**.
- Sua **estrutura principal** é o **DataFrame**, que pode ser visto **como uma tabela do Excel** ou um banco de dados SQL em memória.
- Ele é **otimizado** para lidar com **dados heterogêneos**, ou seja, diferentes tipos de informações em uma mesma tabela.
- Exemplos de aplicação:
    - Processamento de dados de transações financeiras
    - Análise e limpeza de grandes bases de dados
    - Manipulação de datasets de clientes, produtos e vendas
    - Compatível com uma grande variedade de formatos/origens para o carregamento de dados
- É uma biblioteca de terceiro, portanto, preciso instalá-la via pip para poder usá-la: (Ambiente local: pip install pandas, No Notebook: !pip install pandas. Obs.: Google Colab ou Ambiente Anaconda já traz instalada por padrão)

**"Se o NumPy é uma super calculadora, o Pandas é uma super planilha."**

## Entendendo a estrutura e interação do pd.Series e do pd.DataFrame

**Estrutura de uma Series:**

```
import pandas as pd

vendas = pd.Series([100, 200, 300, 250], index=['Jan', 'Fev', 'Mar',
'Abr'])
print(vendas)
```

**Análise da estrutura da Series:**

```
vendas.shape
vendas.describe()
vendas.head()
vendas.tail()
```

**Análise de dados estatísticos consolidados:**

```
print(“Quantidade:”, vendas.count())
print("Total:", vendas.sum())
print("Média:", vendas.mean())
print("Mediana:", vendas.median())
print("Máximo:", vendas.max())
print("Mínimo:", vendas.min())
print("Desvio Padrão:", vendas.std())
```

**Acessar dados:**

```
vendas[‘Fev’] #pelo índice definido str
vendas[1] #pelo índice numérico
vendas.loc[‘Fev’] #loc para índice definido
vendas.iloc[1] #pelo índice numérico
```

**Filtrar dados:**

```
#todas vendas cujo valor seja ou maior igual a 200
print(vendas[vendas>=200])

#todas vendas cujo valor seja maior ou igual a 200 e menor que 300
print(vendas[(vendas[vendas>=200]) & (vendas[vendas<300])])

#todas vendas cujo valor seja diferente de 200
print(vendas[vendas!= 200])

#quantas vendas tem o valor diferente de 200?
print(vendas[vendas!= 200].count())
```

**Operadores booleanos para comparação:**

***and (e):*** &
**or (ou):** |
***negation (negação):*** ~
***equal (igual):*** ==
***different (diferente):*** !=

**Criando um DataFrame:**

```
dados = {
    'produto': ['sorvete', 'refrigerante', 'batata frita'],
    'preco': [8.0, 5.5, 7.0],
    'estoque': [120, 85, 60]
}

df = pd.DataFrame(dados)
```

**Representação do DataFrame como uma planilha:**

```
print(df)
df.head()
df.tail()
```

**Análise da estrutura do DataFrame:**

```
df.shape
df.describe()
df.dtypes
```

**Parênteses sobre tipagem de dados:**

Por padrão o Pandas procura identificar automaticamente o tipo de dado (auto tipado), no entanto, podemos converter o tipo de dados explicitamente (casting):

```
df[‘estoque’] = df[‘estoque’].astype(str)
```

**Manipulação de dados do DataFrame:**

```
#acrescentar a coluna que calcula o total que se tem de ativo investido em cada produto do estoque
df['valor_total'] = df['preco'] * df['estoque']
```

**Remover linha:**

```
df = df.drop(3)  # remove a linha de índice 3
df = df.drop([0, 1]) # remove as linhas de índices 0 e 1

#ou

df.drop(3, inplace=True)
df.drop([0, 1], inplace=True)
```

**Remover coluna:**

```
#parâmetro axis representa os eixos, onde 0 é linha e 1 é coluna
df.drop('valor_total', axis=1, inplace=True)
```

# Análise de aluguel de bikes:

Você foi contratado para um trabalho como analista de dados de uma rede de aluguel de bicicletas. A gestão gostaria de entender quais os dias de maior demanda, se o clima interfere no movimento do aluguel.

**A base de dados coletada do sistema transacional da última semana de uma das filiais:**

```
# Dados fictícios: aluguel de bikes por dia da semana
dados_bike = {
    'dia': ['Seg', 'Ter', 'Qua', 'Qui', 'Sex', 'Sab', 'Dom'],
    'alugueis': [32, 41, 35, 39, 60, 150, 140],
    'chuva': [True, False, True, False, False, False, True]
}

df_bike = pd.DataFrame(dados_bike)
```

**Perguntas a serem respondidas:**

***Em quais dias o aluguel foi maior que 50?***

```
maior_que_50 = df_bike[df_bike['alugueis']>50]
print(maior_que_50)
```

***Qual o total de aluguéis nos dias sem chuva?***

```
total_dias_chuva = df_bike[df_bike['chuva'] == False]['alugueis'].sum()
print(total_dias_chuva)
```

***Qual a média de aluguéis nos dias chuvosos?***

```
media_aluguel_dias_chuvosos = df_bike[df_bike['chuva']]['alugueis'].mean()
print(media_aluguel_dias_chuvosos)
```

***Crie uma nova coluna que classifica o movimento: 'baixo', 'médio', 'alto'.***

```
def verificar_movimento(alugueis):
  if alugueis > 50:
    return 'alto'
  else:
    return 'baixo'

df_bike['movimento'] = df_bike['alugueis'].apply(verificar_movimento)
print(df_bike)
```

# Carregamento de dados de fontes externas:

**O formato CSV:**
CSV é a sigla para Comma-Separated Values, que em português significa valores separados por vírgula. É um formato simples de arquivo de texto usado para armazenar dados em forma de tabela, onde cada linha representa um registro e cada valor (coluna) é separado por uma vírgula. Ele é muito usado porque pode ser aberto facilmente por programas como Excel, Google Sheets e ferramentas de programação como o Python com a biblioteca Pandas.

**Estudo de caso (Banco de dados monitoramento da diabetes - Registro de dados de saúde e intensidade/esforço de exercícios físicos) - Etapas do ETL (Extract, Transform, Load):**

```
import pandas as pd

#carregamento dos dados (fonte: Arquivo CSV)
df = pd.read_csv('/content/drive/MyDrive/Datasets/glicose_data_suja.csv')

#verificar estrutura
df.shape
df.dtypes
df.describe()

#conhecendo os dados
df.head()
df.tail()
```

**Analisando as diversidades dos dados e fazendo o tratamentos:**

```
#remover primeira linha que contem todos dados faltantes
df.drop(0, inplace=True)

#verificar universo de ocorrências dos dados da coluna
df['Dia Semana'].unique()

#verificar se existe algum dado faltante na coluna (NA - Not available)
df['Dia Semana'].isna().sum()
```

**Dicas sobre tratamento de dados faltantes (NAs):**
Ambos (**NA**, **NaN**) significam que **não há dado presente naquela célula**.

- Tipos **object** (rótulos): é necessário substituir os NAs pela **moda** (o que mais aparece). Para descobrir a moda de uma coluna: **moda_dia_semana = df[‘Dia Semana’].mode()[0]** (posição zero, para pegar o dado já no formato str)

- Tipos **numéricos**: é necessário substituir os NAs pela **mediana**. Para descobrir a mediana de uma coluna: **mediana_resultado = df['Resultado'].median()**
- Então, **substituir no DataFrame os valores NAs** de cada coluna. Ex.: **df[Dia Semana'] = df['Dia Semana'].fillna(moda_dia_semana)** e **df['Resultado'] = df['Resultado'].fillna(mediana_resultado)**

**Seguir analisando os dados e tratando os faltantes conforme as respectivas regras para cada tipo de dados.**

**Vamos responder as perguntas para este dataset?**

***Quantas vezes a dose de insulina foi maior que 10?***

```
filtro = df['Dose Insulina'] > 10
df[filtro]
```

***Quantas vezes correu-se e mesmo assim precisou de dose de insulina?***

```
filtro = (df['corrida']>0) & (df['Dose Insulina']>0)
df[filtro]['Dose Insulina'].count()
```

***Quantas vezes a Dose Insulina foi maior que 10 quando se jogou padel ou praticou-se Musculacao H ou praticou-se Musculacao R?***

```
filtro = (df['Dose Insulina']>10) & ((df['padel']>0) | (df['musculacao H']>0) | (df['musculacao R']>0))
df[filtro]['Dose Insulina'].count()
```

# Exercício Avaliativo Final – Análise de Dados de Astronautas:

**Cenário:**
A agência espacial fictícia UFN CosmosLab precisa analisar os perfis dos astronautas que participaram do seu mais recente programa de treinamento para missões em microgravidade. Você foi contratado como analista de dados júnior para explorar as informações coletadas e entregar insights iniciais à equipe de planejamento.

Você recebeu um arquivo CSV contendo os dados de 20 astronautas, incluindo nome, idade, especialidade, tempo em gravidade zero, planeta favorito, e nota na avaliação do treinamento.

**Arquivo CSV:**
astronautas_estacao_espacial.csv

**Objetivo:**
Utilizar as habilidades adquiridas com a biblioteca Pandas para ler, tratar e analisar os dados.

**Analise o dataset, faça os tratamentos se necessário e responda as seguintes perguntas:**

1. Quantos astronautas têm idade registrada e quantos não têm?
2. Qual especialidade aparece com mais frequência na tripulação?
3. Qual é o planeta favorito mais comum?
4. Entre os astronautas com mais de 500 horas em gravidade zero, qual foi a média da avaliação no treinamento?

**Entrega do exercício:**
Você pode entregar suas respostas dentro de um notebook do Google Colab, com comentários explicando seus resultados de forma clara, como se estivesse apresentando para o seu chefe. Compartilhe com: robertson@ersistemas.info e alexz@prof.ufn.edu.br