



Estácio

Campus: Rua Manoel João Gonçalves, 410/412 – Alcântara CEP: 24711-080

Curso: Desenvolvimento Full-Stack

Disciplina: RPG0033 - TRATANDO A IMENSIDÃO DOS DADOS

Turma: 9001

Semestre letivo: 2024.4 FLEX

Integrante:

Nome: Samir Campos Lima

Matrícula: 2022.11.47141-1

Link do repositório no GIT: [samircamposlima/Miss-o-Pr-tica-N-vel-3-Mundo-5](https://github.com/samircamposlima/Miss-o-Pr-tica-N-vel-3-Mundo-5)

Micro atividade 1: Descrever como ler um arquivo CSV usando a biblioteca Pandas (Python)

- Procedimentos

- 1) Salve o conjunto de dados em formato CSV que utilizará num local acessível pela ferramenta de escrita de código que utilizará;
- 2) Crie um novo arquivo e:
 - a) Importe a biblioteca pandas;
 - b) Cria uma variável;
 - c) Leia o conteúdo do arquivo CSV, passando como parâmetros o separador de colunas, a engine – com o valor 'python' e o encoding relativo aos dados constantes no arquivo lido (esse último parâmetro pode ser opcional, dependendo do encoding existente);
 - d) Atribua os dados lidos do CSV à variável criada anteriormente; Salve as alterações;
 - e) Imprima/exiba em tela os dados da variável.

Código:

```
import pandas as pd

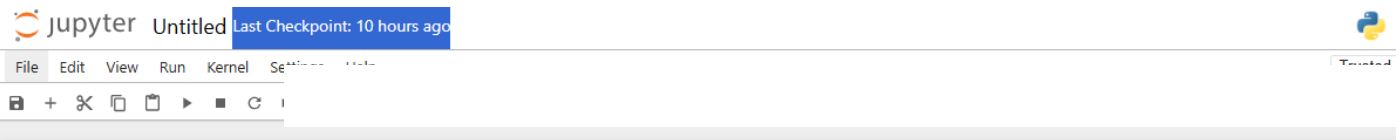
df = pd.read_excel(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail.xlsx', sheet_name='Online Retail', engine='openpyxl', header=0)
print(df.head())

df.to_csv(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail.csv', sep=';', index=False, encoding='utf-8')
display(df)

df = pd.read_csv(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail.csv', sep=';', engine='python')
print(df.head())

with open('Online Retail.csv', 'r') as file:
    print(file.readline())
```

imagem:



```
[1]: import pandas as pd
df = pd.read_excel(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail.xlsx', sheet_name='Online Retail', engine='openpyxl')
print(df.head())
```

	InvoiceNo	StockCode	Description	Quantity	\
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	
1	536365	71053	WHITE METAL LANTERN	6	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	

	InvoiceDate	UnitPrice	CustomerID	Country
0	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

```
[6]: df.to_csv(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail.csv', sep=';', index=False, encoding='utf-8')
display(df)
df = pd.read_csv(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail.csv', sep=';', engine='python')
print(df.head())
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	France

541909 rows × 8 columns

	InvoiceNo	StockCode	Description	Quantity	\
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	
1	536365	71053	WHITE METAL LANTERN	6	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	

	InvoiceDate	UnitPrice	CustomerID	Country
0	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

```
[7]: with open('Online Retail.csv', 'r') as file:
print(file.readline())
```

InvoiceNo;StockCode;Description;Quantity;InvoiceDate;UnitPrice;CustomerID;Country

Micro atividade 2: Descrever como criar um subconjunto de dados a partir de um conjunto existente usando a biblioteca Pandas (Python)

- Procedimentos:

1. No mesmo arquivo/script utilizado na microatividade 1, crie uma nova variável;
2. Atribua, a essa nova variável, um subconjunto de dados contendo apenas parte das colunas (recomenda-se a utilização de 3 colunas) disponíveis no conjunto de dados original;
3. Salve as alterações realizadas;
4. Imprima/exiba em tela os dados da nova variável (que contém o subconjunto de dados).

Código:

```
arquivo_csv = 'Online Retail.csv'
colunas_selecionadas = ['InvoiceNo', 'Description', 'UnitPrice']
df_selecionado = df[colunas_selecionadas]

print(df_selecionado.head())
display(df_selecionado)
```

imagem:

The screenshot shows a JupyterLab window titled 'Untitled' with a last checkpoint of '10 hours ago'. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for file operations and execution. The code cell contains the following Python code:

```
[17]: arquivo_csv = 'Online Retail.csv'
colunas_selecionadas = ['InvoiceNo', 'Description', 'UnitPrice']
df_selecionado = df[colunas_selecionadas]

print(df_selecionado.head())
display(df_selecionado)
```

The output of the code is displayed below the cell, showing the first five rows of the selected data. The data is presented in a table format with columns 'InvoiceNo', 'Description', and 'UnitPrice'.

	InvoiceNo	Description	UnitPrice
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	2.55
1	536365	WHITE METAL LANTERN	3.39
2	536365	CREAM CUPID HEARTS COAT HANGER	2.75
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	3.39
4	536365	RED WOOLLY HOTTIE WHITE HEART.	3.39
...
541904	581587	PACK OF 20 SPACEBOY NAPKINS	0.85
541905	581587	CHILDREN'S APRON DOLLY GIRL	2.10
541906	581587	CHILDRENS CUTLERY DOLLY GIRL	4.15
541907	581587	CHILDRENS CUTLERY CIRCUS PARADE	4.15
541908	581587	BAKING SET 9 PIECE RETROSPOT	4.95

At the bottom of the window, a status bar indicates '541909 rows x 3 columns'. In the bottom right corner, there is a notification: 'Ativar o Windows. Acesse Configurações para ativar o'.

Micro atividade 3: Descrever como configurar o número máximo de linhas a serem exibidas na visualização de um conjunto de dados usando a biblioteca Pandas (Python)

- Procedimentos

1. Abra o arquivo/script utilizado nas microatividades anteriores;
2. Usando as opções de configuração da biblioteca pandas, defina um novo valor para a propriedade “max_rows”, definindo o novo valor para 9999;
3. Salve as alterações;
4. Imprima na tela o conjunto de dados original (criado na microatividade 1) usando o método “to_string()”.

Código:

```
df = pd.read_csv(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail.csv', sep=';', engine='python')

pd.set_option('display.max_rows', 9999)

print(df.head(9999).to_string())
print(pd.get_option('display.max_rows'))
display(df)
```

imagem:

jupyter

Untitled Last Checkpoint: 18 hours ago

File

Edit

View

Run

Kernel

Settings

Help

Trusted

+

✕

📄

📄

▶

■

↺

↻

▶▶

Code

▼

JupyterLab

Python 3 (ipykernel)

[89]:

```
df = pd.read_csv(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail.csv', sep=';', engine='python')
pd.set_option('display.max_rows', 9999)

print(df.head(9999).to_string())
print(pd.get_option('display.max_rows'))
display(df)
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	844068	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
5	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850.0	United Kingdom
6	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850.0	United Kingdom
7	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
8	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
9	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047.0	United Kingdom
10	536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	2010-12-01 08:34:00	2.10	13047.0	United Kingdom
11	536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	2010-12-01 08:34:00	2.10	13047.0	United Kingdom
12	536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	2010-12-01 08:34:00	3.75	13047.0	United Kingdom
13	536367	22310	IVORY KNITTED MUG COSY	6	2010-12-01 08:34:00	1.65	13047.0	United Kingdom
14	536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	2010-12-01 08:34:00	4.25	13047.0	United Kingdom
15	536367	22623	BOX OF VINTAGE JIGSAW BLOCKS	3	2010-12-01 08:34:00	4.95	13047.0	United Kingdom
16	536367	22622	BOX OF VINTAGE ALPHABET BLOCKS	2	2010-12-01 08:34:00	9.95	13047.0	United Kingdom
17	536367	21754	HOME BUILDING BLOCK WORD	3	2010-12-01 08:34:00	5.95	13047.0	United Kingdom
18	536367	21755	LOVE BUILDING BLOCK WORD	3	2010-12-01 08:34:00	5.95	13047.0	United Kingdom
19	536367	21777	RECIPE BOX WITH METAL HEART	4	2010-12-01 08:34:00	7.95	13047.0	United Kingdom
20	536367	48187	DOORMAT NEW ENGLAND	4	2010-12-01 08:34:00	7.95	13047.0	United Kingdom
21	536368	22960	JAM MAKING SET WITH JARS	6	2010-12-01 08:34:00	4.25	13047.0	United Kingdom
22	536368	22913	RED COAT RACK PARIS FASHION	3	2010-12-01 08:34:00	4.95	13047.0	United Kingdom
23	536368	22912	YELLOW COAT RACK PARIS FASHION	3	2010-12-01 08:34:00	4.95	13047.0	United Kingdom
24	536368	22914	BLUE COAT RACK PARIS FASHION	3	2010-12-01 08:34:00	4.95	13047.0	United Kingdom
25	536369	21756	BATH BUILDING BLOCK WORD	3	2010-12-01 08:35:00	5.95	13047.0	United Kingdom
26	536370	22728	ALARM CLOCK BAKELIKE PINK	24	2010-12-01 08:45:00	3.75	12583.0	France
27	536370	22727	ALARM CLOCK BAKELIKE RED	24	2010-12-01 08:45:00	3.75	12583.0	France
28	536370	22726	ALARM CLOCK BAKELIKE GREEN	12	2010-12-01 08:45:00	3.75	12583.0	France
29	536370	21724	PANDA AND BUNNIES STICKER SHEET	12	2010-12-01 08:45:00	0.85	12583.0	France
30	536370	21883	STARS GIFT TAPE	24	2010-12-01 08:45:00	0.65	12583.0	France
31	536370	10002	TINEIATARI E POLITICAL GLOVE	48	2010-12-01 08:45:00	0.85	12583.0	France

jupyter

Untitled Last Checkpoint: 18 hours ago

File

Edit

View

Run

Kernel

Settings

Help

Trusted

+

✕

📄

📄

▶

■

↺

↻

▶▶

Code

▼

JupyterLab

Python 3 (ipykernel)

9988

537224

22405

MONEY BOX POCKET MONEY DESIGN

6

2010-12-05 16:24:00

1.25

13174.0

United Kingdom

9989

537224

22766

PHOTO FRAME CORNICE

4

2010-12-05 16:24:00

2.95

13174.0

United Kingdom

9990

537224

21695

SMALL SILVER FLOWER CANDLE POT

6

2010-12-05 16:24:00

2.95

13174.0

United Kingdom

9991

537224

21693

SMALL HAMMERED SILVER CANDLEPOT

6

2010-12-05 16:24:00

2.95

13174.0

United Kingdom

9992

537224

21889

WOODEN BOX OF DOMINOES

4

2010-12-05 16:24:00

1.25

13174.0

United Kingdom

9993

537224

22767

TRIPLE PHOTO FRAME CORNICE

2

2010-12-05 16:24:00

9.95

13174.0

United Kingdom

9994

537224

21361

LOVE LARGE WOOD LETTERS

1

2010-12-05 16:24:00

12.75

13174.0

United Kingdom

9995

537224

22622

BOX OF VINTAGE ALPHABET BLOCKS

2

2010-12-05 16:24:00

9.95

13174.0

United Kingdom

9996

537224

22623

BOX OF VINTAGE JIGSAW BLOCKS

2

2010-12-05 16:24:00

4.95

13174.0

United Kingdom

9997

537224

21363

HOME SMALL WOOD LETTERS

2

2010-12-05 16:24:00

4.95

13174.0

United Kingdom

9998

537224

21556

CERAMIC STRAWBERRY MONEY BOX

2

2010-12-05 16:24:00

2.55

13174.0

United Kingdom

9999

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	844068	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	France

541909 rows × 8 columns

[]:

Ativar o Windows
Acesse Configurações para

Micro atividade 4: Descrever como exibir as primeiras e últimas “N” linhas de um conjunto de dados usando a biblioteca Pandas (Python)

- Procedimentos

1. Abra o arquivo/script utilizado nas microatividades anteriores;
2. Imprima na tela as apenas as primeiras 10 linhas do conjunto de dados original (criado na microatividade 1);
3. Imprima na tela as apenas as últimas 10 linhas do conjunto de dados original (criado na microatividade 1).

Código:

```
print(df.head(10))  
  
print(df.tail(10))
```

imagem:

jupyter Untitled Last Checkpoint: 18 hours ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

```
[7]: print(df.head(10))

print(df.tail(10))
```

	InvoiceNo	StockCode	Description	Quantity	\
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	
1	536365	71053	WHITE METAL LANTERN	6	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	
5	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	
6	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	
7	536366	22633	HAND WARMER UNION JACK	6	
8	536366	22632	HAND WARMER RED POLKA DOT	6	
9	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	

	InvoiceDate	UnitPrice	CustomerID	Country
0	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
5	2010-12-01 08:26:00	7.65	17850.0	United Kingdom
6	2010-12-01 08:26:00	4.25	17850.0	United Kingdom
7	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
8	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
9	2010-12-01 08:34:00	1.69	13047.0	United Kingdom

	InvoiceNo	StockCode	Description	Quantity	\
541899	581587	22726	ALARM CLOCK BAKELIKE GREEN	4	
541900	581587	22730	ALARM CLOCK BAKELIKE IVORY	4	
541901	581587	22367	CHILDRENS APRON SPACEBOY DESIGN	8	
541902	581587	22629	SPACEBOY LUNCH BOX	12	
541903	581587	23256	CHILDRENS CUTLERY SPACEBOY	4	
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	

	InvoiceDate	UnitPrice	CustomerID	Country
541899	2011-12-09 12:50:00	3.75	12680.0	France
541900	2011-12-09 12:50:00	3.75	12680.0	France
541901	2011-12-09 12:50:00	1.95	12680.0	France
541902	2011-12-09 12:50:00	1.95	12680.0	France
541903	2011-12-09 12:50:00	4.15	12680.0	France
541904	2011-12-09 12:50:00	0.85	12680.0	France
541905	2011-12-09 12:50:00	2.10	12680.0	France
541906	2011-12-09 12:50:00	4.15	12680.0	France
541907	2011-12-09 12:50:00	4.15	12680.0	France
541908	2011-12-09 12:50:00	4.95	12680.0	France

Ativar o Windows
Acesse Configurações para

Micro atividade 5: Descrever como exibir informações gerais sobre as colunas, linhas e dados de um conjunto de dados usando a biblioteca Pandas (Python)

- Procedimentos

1. Abra o arquivo/script utilizado nas microatividades anteriores;
2. Tendo como base o conjunto de dados original:
 - a. Imprima as informações gerais sobre o conjunto – suas colunas, linhas e dados;
 - b. Descubra a partir do comando acima:
 - i. O total de linhas;
 - ii. O total de colunas;
 - iii. A quantidade de dados nulos, caso existam;
 - iv. O tipo de dado de cada coluna;
 - v. A quantidade de memória utilizada pelo conjunto de dados.

Código:

```
print("Informações gerais:")
print(df.info())

print(f"\nQuantidade de linhas: {df.shape[0]}")

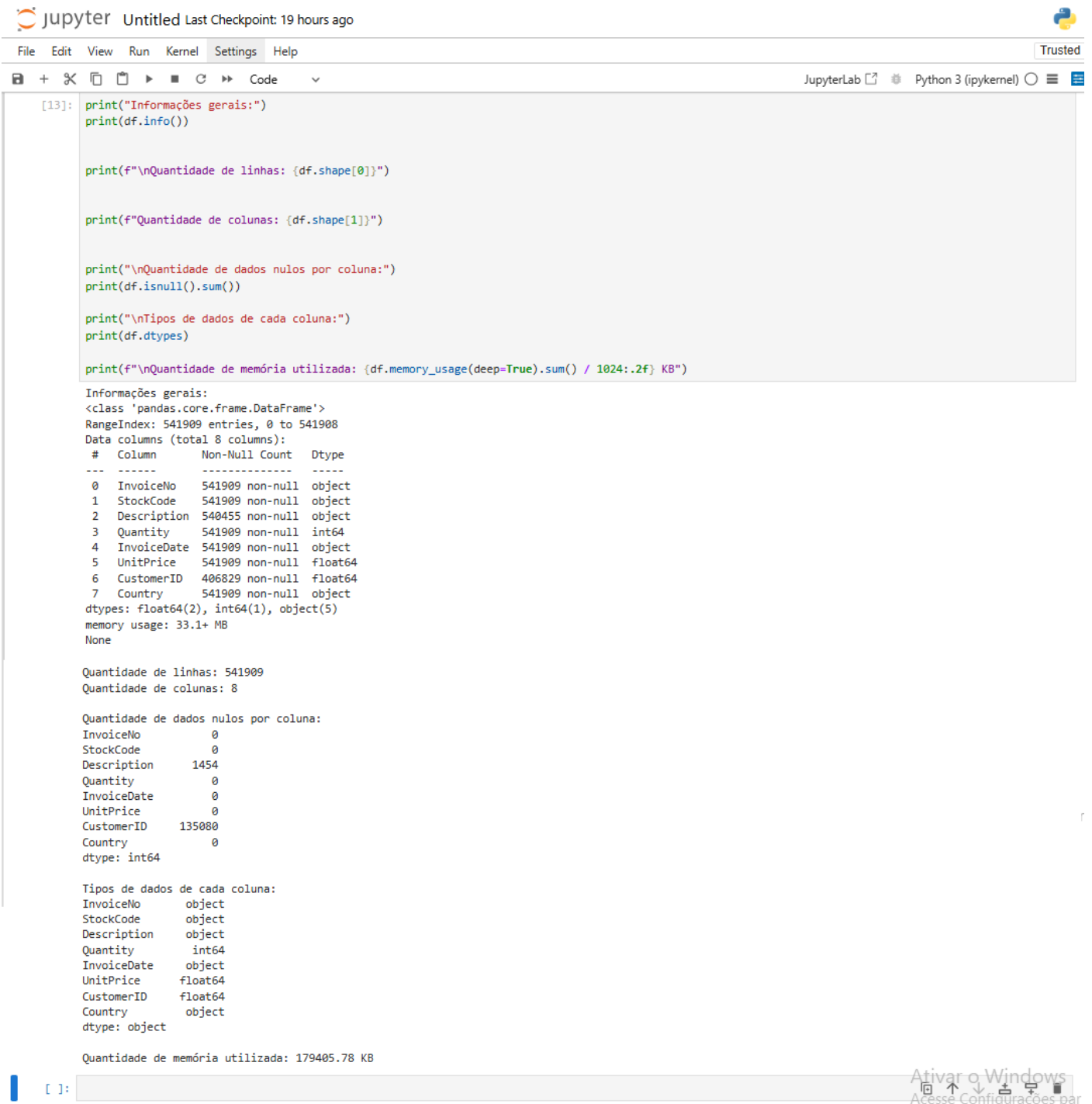
print(f"Quantidade de colunas: {df.shape[1]}")

print("\nQuantidade de dados nulos por coluna:")
print(df.isnull().sum())

print("\nTipos de dados de cada coluna:")
print(df.dtypes)

print(f"\nQuantidade de memória utilizada: {df.memory_usage(deep=True).sum() / 1024:.2f} KB")
```

imagem:



```
[13]: print("Informações gerais:")
print(df.info())

print(f"\nQuantidade de linhas: {df.shape[0]}")

print(f"Quantidade de colunas: {df.shape[1]}")

print("\nQuantidade de dados nulos por coluna:")
print(df.isnull().sum())

print("\nTipos de dados de cada coluna:")
print(df.dtypes)

print(f"\nQuantidade de memória utilizada: {df.memory_usage(deep=True).sum() / 1024:.2f} KB")
```

Informações gerais:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
Column Non-Null Count Dtype
--- ---
0 InvoiceNo 541909 non-null object
1 StockCode 541909 non-null object
2 Description 540455 non-null object
3 Quantity 541909 non-null int64
4 InvoiceDate 541909 non-null object
5 UnitPrice 541909 non-null float64
6 CustomerID 406829 non-null float64
7 Country 541909 non-null object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
None

Quantidade de linhas: 541909
Quantidade de colunas: 8

Quantidade de dados nulos por coluna:
InvoiceNo 0
StockCode 0
Description 1454
Quantity 0
InvoiceDate 0
UnitPrice 0
CustomerID 135080
Country 0
dtype: int64

Tipos de dados de cada coluna:
InvoiceNo object
StockCode object
Description object
Quantity int64
InvoiceDate object
UnitPrice float64
CustomerID float64
Country object
dtype: object

Quantidade de memória utilizada: 179405.78 KB

Missão Prática | Tratando a imensidão dos dados

- Procedimentos

1. Para essa atividade você deverá, obrigatoriamente, utilizar o conjunto de dados (fornecido anteriormente, na seção “Contextualização”) composto pelas colunas ID;Duration;Date;Pulse;Maxpulse;Calories
OBS: Utilizei a planilha do link que não me direcionou para planilha correta
2. Crie um novo arquivo/script;
3. Leia o conteúdo do CSV fornecido, atentando-se para a necessidade ou não de incluir parâmetros adicionais como os relativos ao separador dos dados, a engine e o encoding;
4. Atribua os dados lidos a uma variável;

Código:

```
import pandas as pd

df = pd.read_csv(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail1.csv', sep=';', engine='python')
display(df)
```

imagem:

Jupyter Missão Prática Last Checkpoint: 8 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

```
[1]: import pandas as pd

df = pd.read_csv(r'C:\Users\Samir\OneDrive\Área de Trabalho\Modulo 3 ultimo semestre\Online Retail1.csv', sep=';', engine='python')
display(df)
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01/12/2010 08:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01/12/2010 08:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01/12/2010 08:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01/12/2010 08:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	20201226	3.39	17850.0	United Kingdom
...
95	536378	22352	LUNCH BOX WITH CUTLERY RETROSPOT	6	01/12/2010 09:37	2.55	14688.0	United Kingdom
96	536378	21212	PACK OF 72 RETROSPOT CAKE CASES	120	01/12/2010 09:37	0.42	14688.0	United Kingdom
97	536378	21975	PACK OF 60 DINOSAUR CAKE CASES	24	NaN	0.55	14688.0	United Kingdom
98	536378	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24	01/12/2010 09:37	0.55	14688.0	United Kingdom
99	536378	84991	60 TEATIME FAIRY CAKE CASES	24	01/12/2010 09:37	0.55	14688.0	United Kingdom

100 rows × 8 columns

5. Verifique se os dados foram importados adequadamente:

- Imprima as informações gerais sobre o conjunto de dados;
- Imprima as primeiras e últimas N linhas do arquivo.

Código:

```
print("Informações gerais sobre o conjunto de dados:")
print(df.info())

print("\nQuantidade de dados nulos por coluna:")
print(df.isnull().sum())

print("\nPrimeiras 5 linhas do arquivo:")
print(df.head())

print("\nÚltimas 5 linhas do arquivo:")
print(df.tail())
```

imagem:

```
[2]: print("Informações gerais sobre o conjunto de dados:")
print(df.info())

print("\nQuantidade de dados nulos por coluna:")
print(df.isnull().sum())

print("\nPrimeiras 5 linhas do arquivo:")
print(df.head())

print("\nÚltimas 5 linhas do arquivo:")
print(df.tail())
```

Informações gerais sobre o conjunto de dados:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 8 columns):
Column Non-Null Count Dtype

0 InvoiceNo 100 non-null int64
1 StockCode 100 non-null object
2 Description 100 non-null object
3 Quantity 100 non-null int64
4 InvoiceDate 96 non-null object
5 UnitPrice 100 non-null float64
6 CustomerID 96 non-null float64
7 Country 100 non-null object
dtypes: float64(2), int64(2), object(4)
memory usage: 6.4+ KB
None

Quantidade de dados nulos por coluna:
InvoiceNo 0
StockCode 0
Description 0
Quantity 0
InvoiceDate 4
UnitPrice 0
CustomerID 4
Country 0
dtype: int64

Primeiras 5 linhas do arquivo:

	InvoiceNo	StockCode	Description	Quantity
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6
1	536365	71053	WHITE METAL LANTERN	6
2	536365	844068	CREAM CUPID HEARTS COAT HANGER	8
3	536365	840296	KNITTED UNION FLAG HOT WATER BOTTLE	6
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6

	InvoiceDate	UnitPrice	CustomerID	Country
0	01/12/2010 08:26	2.55	17850.0	United Kingdom
1	01/12/2010 08:26	3.39	17850.0	United Kingdom
2	01/12/2010 08:26	2.75	17850.0	United Kingdom
3	01/12/2010 08:26	3.39	17850.0	United Kingdom
4	20201226	3.39	17850.0	United Kingdom

Últimas 5 linhas do arquivo:

	InvoiceNo	StockCode	Description	Quantity
95	536378	22352	LUNCH BOX WITH CUTLERY RETROSPOT	6
96	536378	21212	PACK OF 72 RETROSPOT CAKE CASES	120
97	536378	21975	PACK OF 60 DINOSAUR CAKE CASES	24
98	536378	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24
99	536378	84991	60 TEATIME FAIRY CAKE CASES	24

	InvoiceDate	UnitPrice	CustomerID	Country
95	01/12/2010 09:37	2.55	14688.0	United Kingdom
96	01/12/2010 09:37	0.42	14688.0	United Kingdom
97	NaN	0.55	14688.0	United Kingdom
98	01/12/2010 09:37	0.55	14688.0	United Kingdom
99	01/12/2010 09:37	0.55	14688.0	United Kingdom

6. Verifique se os dados foram importados adequadamente:
- Imprima as informações gerais sobre o conjunto de dados;
 - Imprima as primeiras e últimas N linhas do arquivo.

Código:


```
print("Informações gerais sobre o conjunto de dados:")
print(df.info())

print("\nQuantidade de dados nulos por coluna:")
print(df.isnull().sum())


print("\nPrimeiras 5 linhas do arquivo:")
print(df.head())

print("\nÚltimas 5 linhas do arquivo:")
print(df.tail())
```

imagem:








 Jupyter

Missão Prática Last Checkpoint: 39 minutes ago



File Edit View Run Kernel Settings Help

Trusted

 +       Code

JupyterLab Python 3 (ipykernel)

```
[2]: print("Informações gerais sobre o conjunto de dados:")
print(df.info())

print("\nQuantidade de dados nulos por coluna:")
print(df.isnull().sum())

print("\nPrimeiras 5 linhas do arquivo:")
print(df.head())

print("\nÚltimas 5 linhas do arquivo:")
print(df.tail())
```

Informações gerais sobre o conjunto de dados:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 8 columns):
Column Non-Null Count Dtype

0 InvoiceNo 100 non-null int64
1 StockCode 100 non-null object
2 Description 100 non-null object
3 Quantity 100 non-null int64
4 InvoiceDate 96 non-null object
5 UnitPrice 100 non-null float64
6 CustomerID 96 non-null float64
7 Country 100 non-null object
dtypes: float64(2), int64(2), object(4)
memory usage: 6.4+ KB
None

Quantidade de dados nulos por coluna:
InvoiceNo 0
StockCode 0
Description 0
Quantity 0
InvoiceDate 4
UnitPrice 0
CustomerID 4
Country 0
dtype: int64

Primeiras 5 linhas do arquivo:

	InvoiceNo	StockCode	Description	Quantity
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6
1	536365	71053	WHITE METAL LANTERN	6
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6

	InvoiceDate	UnitPrice	CustomerID	Country
0	01/12/2010 08:26	2.55	17850.0	United Kingdom
1	01/12/2010 08:26	3.39	17850.0	United Kingdom
2	01/12/2010 08:26	2.75	17850.0	United Kingdom
3	01/12/2010 08:26	3.39	17850.0	United Kingdom
4	20201226	3.39	17850.0	United Kingdom

Últimas 5 linhas do arquivo:

	InvoiceNo	StockCode	Description	Quantity
95	536378	22352	LUNCH BOX WITH CUTLERY RETROSPOT	6
96	536378	21212	PACK OF 72 RETROSPOT CAKE CASES	120
97	536378	21975	PACK OF 60 DINOSAUR CAKE CASES	24
98	536378	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24
99	536378	84991	60 TEATIME FAIRY CAKE CASES	24

	InvoiceDate	UnitPrice	CustomerID	Country
95	01/12/2010 09:37	2.55	14688.0	United Kingdom
96	01/12/2010 09:37	0.42	14688.0	United Kingdom
97	NaN	0.55	14688.0	United Kingdom
98	01/12/2010 09:37	0.55	14688.0	United Kingdom
99	01/12/2010 09:37	0.55	14688.0	United Kingdom

Ativar o Windows
Acesse Configurações par

7. Crie uma nova variável e atribua a ela uma cópia do conjunto de dados original (variável criada no passo 4);
8. Nessa nova variável, contendo uma cópia dos dados:
 - a. Substitua todos os valores nulos da coluna 'CustomerID' por 0;
 - b. Imprima o conjunto de dados para verificar se a mudança acima foi aplicada com sucesso;

Código:

```
missaoPratica = df.copy()

if 'CustomerID' in missaoPratica.columns:
    missaoPratica['CustomerID'] = missaoPratica['CustomerID'].fillna(0)
else:
    print("A coluna 'CustomerID' não existe no conjunto de dados!")

print("Conjunto de dados após a substituição de valores nulos na coluna 'CustomerID':")
print(missaoPratica.head(5).to_string())
print(missaoPratica.tail(5).to_string())
print("\nQuantidade de dados nulos por coluna:")
print(missaoPratica.isnull().sum())
```

imagem:

```
[3]: missaoPratica = df.copy()

[4]: if 'CustomerID' in missaoPratica.columns:
      missaoPratica['CustomerID'] = missaoPratica['CustomerID'].fillna(0)
      else:
          print("A coluna 'CustomerID' não existe no conjunto de dados!")

      print("Conjunto de dados após a substituição de valores nulos na coluna 'CustomerID':")
      print(missaoPratica.head(5).to_string())
      print(missaoPratica.tail(5).to_string())
      print("\nQuantidade de dados nulos por coluna:")
      print(missaoPratica.isnull().sum())
```

Conjunto de dados após a substituição de valores nulos na coluna 'CustomerID':

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	01/12/2010 08:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	01/12/2010 08:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	01/12/2010 08:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	01/12/2010 08:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	20201226	3.39	17850.0	United Kingdom
95	536378	22352	LUNCH BOX WITH CUTLERY RETROSPOT	6	01/12/2010 09:37	2.55	14688.0	United Kingdom
96	536378	21212	PACK OF 72 RETROSPOT CAKE CASES	120	01/12/2010 09:37	0.42	14688.0	United Kingdom
97	536378	21975	PACK OF 60 DINOSAUR CAKE CASES	24		0.55	14688.0	United Kingdom
98	536378	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24	01/12/2010 09:37	0.55	14688.0	United Kingdom
99	536378	84991	60 TEATIME FAIRY CAKE CASES	24	01/12/2010 09:37	0.55	14688.0	United Kingdom

Quantidade de dados nulos por coluna:

InvoiceNo	0
StockCode	0
Description	0
Quantity	0
InvoiceDate	4
UnitPrice	0
CustomerID	0
Country	0

dtype: int64

9. Transforme os dados da coluna 'Date' em datetime usando o método 'to_datetime';

Código:

```
missaoPratica['InvoiceDate'] = pd.to_datetime(missaoPratica['InvoiceDate'],
format='%d/%m/%Y %H:%M')
```

imagem:

```
[6]: missaoPratica['InvoiceDate'] = pd.to_datetime(missaoPratica['InvoiceDate'], format='%d/%m/%Y %H:%M')
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[6], line 1
----> 1 missaoPratica['InvoiceDate'] = pd.to_datetime(missaoPratica['InvoiceDate'], format='%d/%m/%Y %H:%M')
```

```
File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:1063, in to_datetime(arg, errors, dayfirst, yearfirst, utc, format, exact, unit, infer_
datetime_format, origin, cache)
    1061     result = arg.tz_localize("utc")
    1062 elif isinstance(arg, ABCSeries):
--> 1063     cache_array = _maybe_cache(arg, format, cache, convert_listlike)
    1064     if not cache_array.empty:
    1065         result = arg.map(cache_array)
```

```
File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:247, in _maybe_cache(arg, format, cache, convert_listlike)
    245 unique_dates = unique(arg)
    246 if len(unique_dates) < len(arg):
--> 247     cache_dates = convert_listlike(unique_dates, format)
    248     # GH#45319
    249     try:
```

```
File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:433, in _convert_listlike_datetimes(arg, format, name, utc, unit, errors, dayfirst, yearfirst, exact)
    431 # 'format' could be inferred, or user didn't ask for mixed-format parsing.
    432 if format is not None and format != "mixed":
--> 433     return _array_strptime_with_fallback(arg, name, utc, format, exact, errors)
    435 result, tz_parsed = objects_to_datetime64(
    436     arg,
    437     dayfirst=dayfirst,
    (...)
    441     allow_object=True,
    442 )
    444 if tz_parsed is not None:
    445     # We can take a shortcut since the datetime64 numpy array
    446     # is in UTC
```

```
File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:467, in _array_strptime_with_fallback(arg, name, utc, fmt, exact, errors)
    456 def _array_strptime_with_fallback(
    457     arg,
    458     name,
    (...)
    462     errors: str,
    463 ) -> Index:
    464     """
    465     Call array_strptime, with fallback behavior depending on 'errors'.
    466     """
--> 467     result, tz_out = array_strptime(arg, fmt, exact=exact, errors=errors, utc=utc)
    468     if tz_out is not None:
    469         unit = np.datetime_data(result.dtype)[0]
```

```
File strptime.pyx:501, in pandas._libs.tslibs.strptime.array_strptime()
File strptime.pyx:451, in pandas._libs.tslibs.strptime.array_strptime()
File strptime.pyx:583, in pandas._libs.tslibs.strptime._parse_with_format()

ValueError: time data "20201226" doesn't match format "%d/%m/%Y %H:%M", at position 1. You might want to try:
- passing 'format' if your strings have a consistent format;
- passing 'format='ISO8601'' if your strings are all ISO8601 but not necessarily in exactly the same format;
- passing 'format='mixed'', and the format will be inferred for each element individually. You might want to use 'dayfirst' alongside this.
```

- a. Nesse ponto, você deverá ter esbarrado em um erro, informando agora que o valor "20201226" não corresponde ao formato "%Y/%m/%d" . Você precisará, agora, na coluna 'Date', transformar especificamente esse valor, atualmente uma string, para o formato datetime. Para isso você deverá combinar os métodos 'replace' e 'to_datetime';

Código:

```
def adicionar_hora_se_necessario(data):
    if isinstance(data, str) and len(data) == 8: # Verifica se é uma string
com 8 caracteres
        return data + ' 00:00'
    else:
        return data

missaoPratica['InvoiceDate'] =
missaoPratica['InvoiceDate'].apply(adicionar_hora_se_necessario)
missaoPratica['InvoiceDate'] = missaoPratica['InvoiceDate'].fillna('120201226')
missaoPratica['InvoiceDate'] = missaoPratica['InvoiceDate'].replace('20201226
00:00', '26/12/2020 00:00')

print(missaoPratica)
```

imagem:

```
if isinstance(data, str) and len(data) == 8: # verifica se é uma string com 8 caracteres
    return data + ' 00:00'
else:
    return data

missaoPratica['InvoiceDate'] = missaoPratica['InvoiceDate'].apply(adicionar_hora_se_necessario)
missaoPratica['InvoiceDate'] = missaoPratica['InvoiceDate'].fillna('120201226')
missaoPratica['InvoiceDate'] = missaoPratica['InvoiceDate'].replace('20201226 00:00', '26/12/2020 00:00')

print(missaoPratica)
```

	InvoiceNo	StockCode	Description	Quantity	\
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	
1	536365	71053	WHITE METAL LANTERN	6	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	
..	
95	536378	22352	LUNCH BOX WITH CUTLERY RETROSPOT	6	
96	536378	21212	PACK OF 72 RETROSPOT CAKE CASES	120	
97	536378	21975	PACK OF 60 DINOSAUR CAKE CASES	24	
98	536378	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24	
99	536378	84991	60 TEATIME FAIRY CAKE CASES	24	
	InvoiceDate	UnitPrice	CustomerID	Country	
0	01/12/2010 08:26	2.55	17850.0	United Kingdom	
1	01/12/2010 08:26	3.39	17850.0	United Kingdom	
2	01/12/2010 08:26	2.75	17850.0	United Kingdom	
3	01/12/2010 08:26	3.39	17850.0	United Kingdom	
4	26/12/2020 00:00	3.39	17850.0	United Kingdom	
..	
95	01/12/2010 09:37	2.55	14688.0	United Kingdom	
96	01/12/2010 09:37	0.42	14688.0	United Kingdom	
97	1900/01/01 00:00	0.55	14688.0	United Kingdom	
98	01/12/2010 09:37	0.55	14688.0	United Kingdom	
99	01/12/2010 09:37	0.55	14688.0	United Kingdom	

[100 rows x 8 columns]

10. Ainda na nova variável:

- Substitua os valores nulos da coluna 'Date' por '1900/01/01';
- Imprima o conjunto de dados e confira se a mudança foi aplicada com sucesso;

Código:

```
missaoPratica['InvoiceDate'] = missaoPratica['InvoiceDate'].fillna('1900/01/01 00:00')
print(missaoPratica)
print(missaoPratica.isnull().sum())
```

imagem:

```
[5]: missaoPratica['InvoiceDate'] = missaoPratica['InvoiceDate'].fillna('1900/01/01 00:00')
print(missaoPratica)
print(missaoPratica.isnull().sum())
```

	InvoiceNo	StockCode	Description	Quantity	\
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	
1	536365	71053	WHITE METAL LANTERN	6	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	
..	
95	536378	22352	LUNCH BOX WITH CUTLERY RETROSPOT	6	
96	536378	21212	PACK OF 72 RETROSPOT CAKE CASES	120	
97	536378	21975	PACK OF 60 DINOSAUR CAKE CASES	24	
98	536378	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24	
99	536378	84991	60 TEATIME FAIRY CAKE CASES	24	

	InvoiceDate	UnitPrice	CustomerID	Country
0	01/12/2010 08:26	2.55	17850.0	United Kingdom
1	01/12/2010 08:26	3.39	17850.0	United Kingdom
2	01/12/2010 08:26	2.75	17850.0	United Kingdom
3	01/12/2010 08:26	3.39	17850.0	United Kingdom
4	20201226	3.39	17850.0	United Kingdom
..
95	01/12/2010 09:37	2.55	14688.0	United Kingdom
96	01/12/2010 09:37	0.42	14688.0	United Kingdom
97	1900/01/01 00:00	0.55	14688.0	United Kingdom
98	01/12/2010 09:37	0.55	14688.0	United Kingdom
99	01/12/2010 09:37	0.55	14688.0	United Kingdom

```
[100 rows x 8 columns]
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

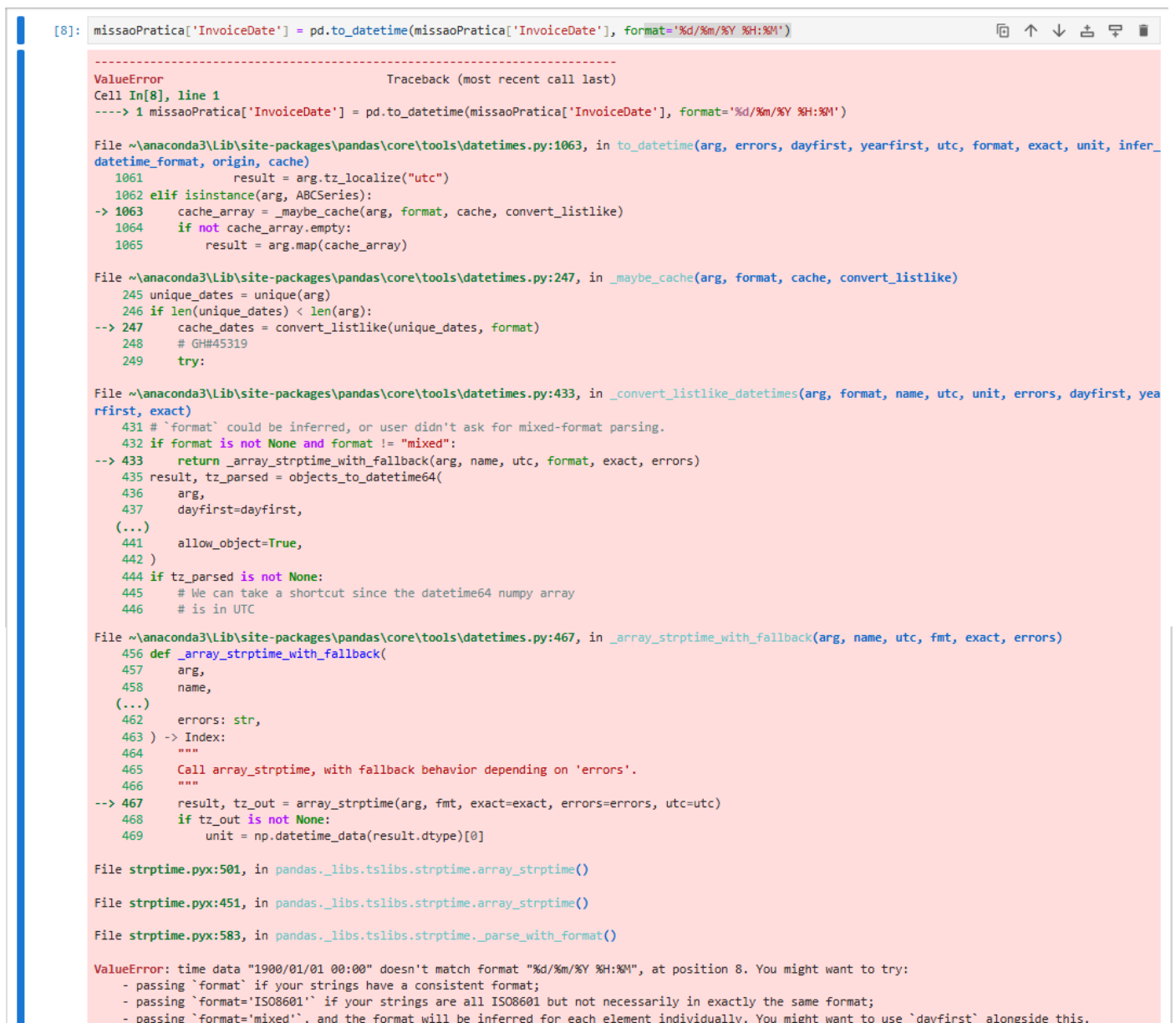
Ativar o Windows
Acesse Configurações para

11. Transforme os dados da coluna 'Date' em datetime usando o método 'to_datetime';
- a. Tendo seguido todas as instruções anteriores, ao executar o passo anterior você deverá ter encontrado outro erro informando que o valor '1900/01/01' não corresponde ao formato '%Y/%m/%d'. Para resolver esse problema:

Código:

```
missaoPratica['InvoiceDate'] = pd.to_datetime(missaoPratica['InvoiceDate'],
format='%d/%m/%Y %H:%M')
```

imagem:



```
[8]: missaoPratica['InvoiceDate'] = pd.to_datetime(missaoPratica['InvoiceDate'], format='%d/%m/%Y %H:%M')
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[8], line 1
----> 1 missaoPratica['InvoiceDate'] = pd.to_datetime(missaoPratica['InvoiceDate'], format='%d/%m/%Y %H:%M')

File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:1063, in to_datetime(arg, errors, dayfirst, yearfirst, utc, format, exact, unit, infer_datetime_format, origin, cache)
    1061     result = arg.tz_localize("utc")
    1062 elif isinstance(arg, ABCSeries):
-> 1063     cache_array = _maybe_cache(arg, format, cache, convert_listlike)
    1064     if not cache_array.empty:
    1065         result = arg.map(cache_array)

File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:247, in _maybe_cache(arg, format, cache, convert_listlike)
    245 unique_dates = unique(arg)
    246 if len(unique_dates) < len(arg):
-> 247     cache_dates = convert_listlike(unique_dates, format)
    248     # GH#45319
    249     try:

File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:433, in _convert_listlike_datetimes(arg, format, name, utc, unit, errors, dayfirst, yearfirst, exact)
    431 # 'format' could be inferred, or user didn't ask for mixed-format parsing.
    432 if format is not None and format != "mixed":
-> 433     return _array_strptime_with_fallback(arg, name, utc, format, exact, errors)
    435 result, tz_parsed = objects_to_datetime64(
    436     arg,
    437     dayfirst=dayfirst,
    (...)
    441     allow_object=True,
    442 )
    444 if tz_parsed is not None:
    445     # We can take a shortcut since the datetime64 numpy array
    446     # is in UTC

File ~\anaconda3\Lib\site-packages\pandas\core\tools\datetimes.py:467, in _array_strptime_with_fallback(arg, name, utc, fmt, exact, errors)
    456 def _array_strptime_with_fallback(
    457     arg,
    458     name,
    (...)
    462     errors: str,
    463 ) -> Index:
    464     """
    465     Call array_strptime, with fallback behavior depending on 'errors'.
    466     """
-> 467     result, tz_out = array_strptime(arg, fmt, exact=exact, errors=errors, utc=utc)
    468     if tz_out is not None:
    469         unit = np.datetime_data(result.dtype)[0]

File strptime.pyx:501, in pandas._libs.tslibs.strptime.array_strptime()

File strptime.pyx:451, in pandas._libs.tslibs.strptime.array_strptime()

File strptime.pyx:583, in pandas._libs.tslibs.strptime._parse_with_format()

ValueError: time data "1900/01/01 00:00" doesn't match format "%d/%m/%Y %H:%M", at position 8. You might want to try:
- passing 'format' if your strings have a consistent format;
- passing 'format='ISO8601'' if your strings are all ISO8601 but not necessarily in exactly the same format;
- passing 'format='mixed'', and the format will be inferred for each element individually. You might want to use 'dayfirst' alongside this.
```

- b. Substitua, na coluna 'Date', o valor '1900/01/01' por 'NaN';
- c. Utilizando o método 'to_datetime', repita o passo de transformação dos dados da coluna 'Date' para datetime;
- d. Imprima o conjunto de dados para verificar se as mudanças acima foram aplicadas com sucesso;

Código:

```
missaoPratica['InvoiceDate'] =
missaoPratica['InvoiceDate'].replace('1900/01/01 00:00', pd.NA)

print(missaoPratica)
```

imagem:

The screenshot shows a JupyterLab window titled "Missão Prática Last Checkpoint: 6 minutes ago". The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar. The code cell contains the following Python code:

```
[8]: missaoPratica['InvoiceDate'] = missaoPratica['InvoiceDate'].replace('1900/01/01 00:00', pd.NA)

print(missaoPratica)
```

The output of the code is a pandas DataFrame with 100 rows and 8 columns. The first 10 rows are shown, followed by an ellipsis, and then rows 95 through 99. The columns are InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, and Country.

	InvoiceNo	StockCode	Description	Quantity
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6
1	536365	71053	WHITE METAL LANTERN	6
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6
..
95	536378	22352	LUNCH BOX WITH CUTLERY RETROSPOT	6
96	536378	21212	PACK OF 72 RETROSPOT CAKE CASES	120
97	536378	21975	PACK OF 60 DINOSAUR CAKE CASES	24
98	536378	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24
99	536378	84991	60 TEATIME FAIRY CAKE CASES	24

The second part of the output shows the 'InvoiceDate' column with values like '01/12/2010 08:26' and '26/12/2020 00:00'. The 'UnitPrice' column has values like 2.55, 3.39, and 0.55. The 'CustomerID' column has values like 17850.0 and 14688.0. The 'Country' column has the value 'United Kingdom'.

[100 rows x 8 columns]

12. Após o passo anterior, execute novamente a transformação de todos os dados da coluna 'Date' para o formato datetime (usando o `to_datetime`). Imprima o conjunto de dados atual para verificar se todas as transformações foram executadas com sucesso;

Código:

```
missaoPratica['InvoiceDate'] = pd.to_datetime(missaoPratica['InvoiceDate'],
format='%d/%m/%Y %H:%M')
print(missaoPratica)
```

imagem:

```
[10]: missaoPratica['InvoiceDate'] = pd.to_datetime(missaoPratica['InvoiceDate'], format='%d/%m/%Y %H:%M')
print(missaoPratica)
```

	InvoiceNo	StockCode	Description	Quantity	\
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	
1	536365	71053	WHITE METAL LANTERN	6	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	
..	
95	536378	22352	LUNCH BOX WITH CUTLERY RETROSPOT	6	
96	536378	21212	PACK OF 72 RETROSPOT CAKE CASES	120	
97	536378	21975	PACK OF 60 DINOSAUR CAKE CASES	24	
98	536378	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24	
99	536378	84991	60 TEATIME FAIRY CAKE CASES	24	

	InvoiceDate	UnitPrice	CustomerID	Country
0	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	2020-12-26 00:00:00	3.39	17850.0	United Kingdom
..
95	2010-12-01 09:37:00	2.55	14688.0	United Kingdom
96	2010-12-01 09:37:00	0.42	14688.0	United Kingdom
97	NaT	0.55	14688.0	United Kingdom
98	2010-12-01 09:37:00	0.55	14688.0	United Kingdom
99	2010-12-01 09:37:00	0.55	14688.0	United Kingdom

[100 rows x 8 columns]

13. Por fim, remova os registros contendo valores nulos. Nesse ponto, apenas a coluna 'Date' possui um registro que atende a essa premissa (em 4 linha).

Logo,

utilize-a como base para realizar a transformação solicitada;

Código:

```
missaoPratica = missaoPratica.dropna(subset=['InvoiceDate'])
print(missaoPratica)
```

imagem:

[11]:

```
missaoPratica = missaoPratica.dropna(subset=['InvoiceDate'])
print(missaoPratica)
```

	InvoiceNo	StockCode	Description	Quantity	\
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	
1	536365	71053	WHITE METAL LANTERN	6	
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	
..	
94	536378	21559	STRAWBERRY LUNCH BOX WITH CUTLERY	6	
95	536378	22352	LUNCH BOX WITH CUTLERY RETROSPOT	6	
96	536378	21212	PACK OF 72 RETROSPOT CAKE CASES	120	
98	536378	21977	PACK OF 60 PINK PAISLEY CAKE CASES	24	
99	536378	84991	60 TEATIME FAIRY CAKE CASES	24	

	InvoiceDate	UnitPrice	CustomerID	Country
0	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	2020-12-26 00:00:00	3.39	17850.0	United Kingdom
..
94	2010-12-01 09:37:00	2.55	14688.0	United Kingdom
95	2010-12-01 09:37:00	2.55	14688.0	United Kingdom
96	2010-12-01 09:37:00	0.42	14688.0	United Kingdom
98	2010-12-01 09:37:00	0.55	14688.0	United Kingdom
99	2010-12-01 09:37:00	0.55	14688.0	United Kingdom

[96 rows x 8 columns]