

```
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
import seaborn as sns
```

```
import matplotlib.pyplot as plt
correlation_matrix = data.corr()
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    correlation_matrix = data.corr()
NameError: name 'data' is not defined
```

===== RESTART: Shell =====

```
import pandas as pd
file_path = r'C:\Users\biola\Downloads\Life_Expectancy_Data.csv'
data = pd.read_csv(file_path)
print(data.head())
```

	Country	Year	...	Income composition of resources	Schooling
0	Afghanistan	2015	...	0.479	10.1
1	Afghanistan	2014	...	0.476	10.0
2	Afghanistan	2013	...	0.470	9.9
3	Afghanistan	2012	...	0.463	9.8
4	Afghanistan	2011	...	0.454	9.5

```
[5 rows x 22 columns]
print(data.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
```

#	Column	Non-Null Count	Dtype
0	Country	2938 non-null	object
1	Year	2938 non-null	int64
2	Status	2938 non-null	object
3	Life expectancy	2928 non-null	float64
4	Adult Mortality	2928 non-null	float64
5	infant deaths	2938 non-null	int64
6	Alcohol	2744 non-null	float64
7	percentage expenditure	2938 non-null	float64
8	Hepatitis B	2385 non-null	float64
9	Measles	2938 non-null	int64
10	BMI	2904 non-null	float64
11	under-five deaths	2938 non-null	int64
12	Polio	2919 non-null	float64
13	Total expenditure	2712 non-null	float64
14	Diphtheria	2919 non-null	float64
15	HIV/AIDS	2938 non-null	float64
16	GDP	2490 non-null	float64
17	Population	2286 non-null	float64
18	thinness 1-19 years	2904 non-null	float64

```

19  thinness 5-9 years          2904 non-null    float64
20  Income composition of resources  2771 non-null    float64
21  Schooling                  2775 non-null    float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
None
for column in data.select_dtypes(include=['float64', 'int64']).columns:
    if data[column].isnull().sum() > 0:
        data[column] = data[column].fillna(data[column].mean())

```

```

print("Missing values handled.")
Missing values handled.
import matplotlib.pyplot as plt

```

```

target_variable = 'Life expectancy'
for column in numeric_columns:
    if column != target_variable:
        plt.figure(figsize=(8, 6))
        plt.scatter(data[column], data[target_variable], alpha=0.5)
        plt.title(f"Scatter Plot: {column} vs {target_variable}")
        plt.xlabel(column)
        plt.ylabel(target_variable)
        plt.show()

```

Traceback (most recent call last):

```

File "<pyshell#14>", line 1, in <module>
    for column in numeric_columns:
NameError: name 'numeric_columns' is not defined
numeric_columns = data.select_dtypes(include=['float64', 'int64']).columns
print("Numeric columns:", numeric_columns)
Numeric columns: Index(['Year', 'Life expectancy ', 'Adult Mortality', 'infant
deaths',
    'Alcohol', 'percentage expenditure', 'Hepatitis B', 'Measles ', ' BMI ',
    'under-five deaths ', 'Polio', 'Total expenditure', 'Diphtheria ',
    ' HIV/AIDS', 'GDP', 'Population', ' thinness 1-19 years',
    ' thinness 5-9 years', 'Income composition of resources', 'Schooling'],
    dtype='object')
target_variable = 'Life expectancy'
numeric_columns = data.select_dtypes(include=['float64', 'int64']).columns
print("Numeric columns:", numeric_columns)

Numeric columns: Index(['Year', 'Life expectancy ', 'Adult Mortality', 'infant
deaths',
    'Alcohol', 'percentage expenditure', 'Hepatitis B', 'Measles ', ' BMI ',
    'under-five deaths ', 'Polio', 'Total expenditure', 'Diphtheria ',
    ' HIV/AIDS', 'GDP', 'Population', ' thinness 1-19 years',
    ' thinness 5-9 years', 'Income composition of resources', 'Schooling'],
    dtype='object')

```

```

for column in numeric_columns:
    if column != target_variable:
        plt.figure(figsize=(8, 6))
        plt.scatter(data[column], data[target_variable], alpha=0.5)
        plt.title(f"Scatter Plot: {column} vs {target_variable}")
        plt.xlabel(column)
        plt.ylabel(target_variable)
        plt.show()

```

<Figure size 800x600 with 0 Axes>

Traceback (most recent call last):

```

File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3805, in get_loc
    return self._engine.get_loc(casted_key)
File "index.pyx", line 167, in pandas._libs.index.IndexEngine.get_loc
File "index.pyx", line 196, in pandas._libs.index.IndexEngine.get_loc
File "pandas\_libs\hashtable_class_helper.pxi", line 7081, in
pandas._libs.hashtable.PyObjectHashTable.get_item
File "pandas\_libs\hashtable_class_helper.pxi", line 7089, in
pandas._libs.hashtable.PyObjectHashTable.get_item
KeyError: 'Life expectancy'

```

The above exception was the direct cause of the following exception:

Traceback (most recent call last):

```

File "<pyshell#21>", line 4, in <module>
    plt.scatter(data[column], data[target_variable], alpha=0.5)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py", line 4102, in __getitem__
    indexer = self.columns.get_loc(key)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3812, in get_loc
    raise KeyError(key) from err
KeyError: 'Life expectancy'
print(data.columns)
Index(['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortality',
      'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
      'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditure',
      'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
      ' thinness 1-19 years', ' thinness 5-9 years',
      'Income composition of resources', 'Schooling'],
      dtype='object')
target_variable = 'Life expectancy'
data.columns = data.columns.str.strip().str.lower()
target_variable = 'life expectancy'
print(data.columns)

```

```

Index(['country', 'year', 'status', 'life expectancy', 'adult mortality',
      'infant deaths', 'alcohol', 'percentage expenditure', 'hepatitis b',
      'measles', 'bmi', 'under-five deaths', 'polio', 'total expenditure',
      'diphtheria', 'hiv/aids', 'gdp', 'population', 'thinness 1-19 years',
      'thinness 5-9 years', 'income composition of resources', 'schooling'],
      dtype='object')
for column in numeric_columns:
    if column != target_variable:
        plt.figure(figsize=(8, 6))
        plt.scatter(data[column], data[target_variable], alpha=0.5)
        plt.title(f"Scatter Plot: {column} vs {target_variable}")
        plt.xlabel(column)
        plt.ylabel(target_variable)
        plt.show()

```

<Figure size 800x600 with 0 Axes>

Traceback (most recent call last):

```

File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3805, in get_loc
    return self._engine.get_loc(casted_key)
File "index.pyx", line 167, in pandas._libs.index.IndexEngine.get_loc
File "index.pyx", line 196, in pandas._libs.index.IndexEngine.get_loc
File "pandas\_libs\hashtable_class_helper.pxi", line 7081, in
pandas._libs.hashtable.PyObjectHashTable.get_item
File "pandas\_libs\hashtable_class_helper.pxi", line 7089, in
pandas._libs.hashtable.PyObjectHashTable.get_item
KeyError: 'Year'

```

The above exception was the direct cause of the following exception:

Traceback (most recent call last):

```

File "<pyshell#28>", line 4, in <module>
    plt.scatter(data[column], data[target_variable], alpha=0.5)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py", line 4102, in __getitem__
    indexer = self.columns.get_loc(key)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3812, in get_loc
    raise KeyError(key) from err
KeyError: 'Year'
print(data.columns)
Index(['country', 'year', 'status', 'life expectancy', 'adult mortality',
      'infant deaths', 'alcohol', 'percentage expenditure', 'hepatitis b',
      'measles', 'bmi', 'under-five deaths', 'polio', 'total expenditure',
      'diphtheria', 'hiv/aids', 'gdp', 'population', 'thinness 1-19 years',
      'thinness 5-9 years', 'income composition of resources', 'schooling'],

```

```

dtype='object')
irrelevant_columns = ['Year']
filtered_columns = [col for col in numeric_columns if col not in irrelevant_columns]
print("Filtered columns for scatter plots:", filtered_columns)
Filtered columns for scatter plots: ['Life expectancy ', 'Adult Mortality', 'infant
deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B', 'Measles ', ' BMI ',
'under-five deaths ', 'Polio', 'Total expenditure', 'Diphtheria ', ' HIV/AIDS',
'GDP', 'Population', ' thinness 1-19 years', ' thinness 5-9 years', 'Income
composition of resources', 'Schooling']
for column in filtered_columns:
    if column != target_variable:
        plt.figure(figsize=(8, 6))
        plt.scatter(data[column], data[target_variable], alpha=0.5)
        plt.title(f"Scatter Plot: {column} vs {target_variable}")
        plt.xlabel(column)
        plt.ylabel(target_variable)
        plt.show()

```

<Figure size 800x600 with 0 Axes>

Traceback (most recent call last):

```

File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3805, in get_loc
    return self._engine.get_loc(casted_key)
File "index.pyx", line 167, in pandas._libs.index.IndexEngine.get_loc
File "index.pyx", line 196, in pandas._libs.index.IndexEngine.get_loc
File "pandas\_libs\hashtable_class_helper.pxi", line 7081, in
pandas._libs.hashtable.PyObjectHashTable.get_item
File "pandas\_libs\hashtable_class_helper.pxi", line 7089, in
pandas._libs.hashtable.PyObjectHashTable.get_item
KeyError: 'Life expectancy '

```

The above exception was the direct cause of the following exception:

Traceback (most recent call last):

```

File "<pyshell#34>", line 4, in <module>
    plt.scatter(data[column], data[target_variable], alpha=0.5)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py", line 4102, in __getitem__
    indexer = self.columns.get_loc(key)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3812, in get_loc
    raise KeyError(key) from err
KeyError: 'Life expectancy '
data.columns = data.columns.str.strip().str.lower()
target_variable = 'life expectancy'
filtered_columns = [col for col in numeric_columns if col not in ['year']]

```

```

for column in filtered_columns:
    if column != target_variable:
        plt.figure(figsize=(8, 6))
        plt.scatter(data[column], data[target_variable], alpha=0.5)
        plt.title(f"Scatter Plot: {column} vs {target_variable}")
        plt.xlabel(column)
        plt.ylabel(target_variable)
        plt.show()

```

<Figure size 800x600 with 0 Axes>

Traceback (most recent call last):

```

File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3805, in get_loc
    return self._engine.get_loc(casted_key)
File "index.pyx", line 167, in pandas._libs.index.IndexEngine.get_loc
File "index.pyx", line 196, in pandas._libs.index.IndexEngine.get_loc
File "pandas\_libs\hashtable_class_helper.pxi", line 7081, in
pandas._libs.hashtable.PyObjectHashTable.get_item
File "pandas\_libs\hashtable_class_helper.pxi", line 7089, in
pandas._libs.hashtable.PyObjectHashTable.get_item
KeyError: 'Year'

```

The above exception was the direct cause of the following exception:

Traceback (most recent call last):

```

File "<pyshell#40>", line 4, in <module>
    plt.scatter(data[column], data[target_variable], alpha=0.5)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py", line 4102, in __getitem__
    indexer = self.columns.get_loc(key)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\indexes\base.py", line 3812, in get_loc
    raise KeyError(key) from err
KeyError: 'Year'
numeric_columns = data.select_dtypes(include=['float64', 'int64']).columns
irrelevant_columns = ['year'] # Add other irrelevant columns here if needed
filtered_columns = [col for col in numeric_columns if col not in irrelevant_columns]
print("Filtered numeric columns:", filtered_columns)
Filtered numeric columns: ['life expectancy', 'adult mortality', 'infant deaths',
'percentage expenditure', 'hepatitis b', 'measles', 'bmi', 'under-five
deaths', 'polio', 'total expenditure', 'diphtheria', 'hiv/aids', 'gdp',
'population', 'thinness 1-19 years', 'thinness 5-9 years', 'income composition of
resources', 'schooling']
target_variable = 'life expectancy'
for column in filtered_columns:

```

```

if column != target_variable:
    plt.figure(figsize=(8, 6))
    plt.scatter(data[column], data[target_variable], alpha=0.5)
    plt.title(f"Scatter Plot: {column} vs {target_variable}")
    plt.xlabel(column)
    plt.ylabel(target_variable)
    plt.show()

```

```

<Figure size 800x600 with 0 Axes>
<matplotlib.collections.PathCollection object at 0x000002C011572000>
Text(0.5, 1.0, 'Scatter Plot: adult mortality vs life expectancy')
Text(0.5, 0, 'adult mortality')
Text(0, 0.5, 'life expectancy')

```

===== RESTART: Shell =====

```

import seaborn as sns
correlation_matrix = data.corr()

```

Traceback (most recent call last):

```

File "<pyshell#49>", line 1, in <module>
    correlation_matrix = data.corr()
NameError: name 'data' is not defined
import pandas as pd
file_path = r'C:\Users\biola\Downloads\Life_Expectancy_Data.csv'
data = pd.read_csv(file_path)
print(data.head())

```

	Country	Year	...	Income composition of resources	Schooling
0	Afghanistan	2015	...	0.479	10.1
1	Afghanistan	2014	...	0.476	10.0
2	Afghanistan	2013	...	0.470	9.9
3	Afghanistan	2012	...	0.463	9.8
4	Afghanistan	2011	...	0.454	9.5

[5 rows x 22 columns]

```

correlation_matrix = data.corr()

```

Traceback (most recent call last):

```

File "<pyshell#54>", line 1, in <module>
    correlation_matrix = data.corr()
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py", line 11049, in corr
    mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py", line 1993, in to_numpy
    result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core

```

```
e\internals\managers.py", line 1694, in as_array
    arr = self._interleave(dtype=dtype, na_value=na_value)
File
"C:\Users\biola\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core
e\internals\managers.py", line 1753, in _interleave
    result[rl.indexer] = arr
ValueError: could not convert string to float: 'Afghanistan'
>>> numeric_data = data.select_dtypes(include=['float64', 'int64'])
>>> correlation_matrix = numeric_data.corr()
>>> print(correlation_matrix)
```

	Year	...	Schooling
Year	1.000000	...	0.209400
Life expectancy	0.170033	...	0.751975
Adult Mortality	-0.079052	...	-0.454612
infant deaths	-0.037415	...	-0.193720
Alcohol	-0.052990	...	0.547378
percentage expenditure	0.031400	...	0.389687
Hepatitis B	0.104333	...	0.231117
Measles	-0.082493	...	-0.137225
BMI	0.108974	...	0.546961
under-five deaths	-0.042937	...	-0.209373
Polio	0.094158	...	0.417866
Total expenditure	0.090740	...	0.246384
Diphtheria	0.134337	...	0.425332
HIV/AIDS	-0.139741	...	-0.220429
GDP	0.101620	...	0.448273
Population	0.016969	...	-0.031668
thinness 1-19 years	-0.047876	...	-0.471652
thinness 5-9 years	-0.050929	...	-0.460632
Income composition of resources	0.243468	...	0.800092
Schooling	0.209400	...	1.000000

```
[20 rows x 20 columns]
>>> import seaborn as sns
>>> import matplotlib.pyplot as plt
>>> plt.figure(figsize=(12, 8))
<Figure size 1200x800 with 0 Axes>
>>> sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f',
cbar=True)
<Axes: >
>>> plt.title("Correlation Heatmap")
Text(0.5, 1.0, 'Correlation Heatmap')
>>> plt.show()
import matplotlib.pyplot as plt
... numeric_columns = data.select_dtypes(include=['float64', 'int64']).columns
... for column in numeric_columns:
...     plt.figure(figsize=(8, 6))
...     plt.boxplot(data[column].dropna(), vert=False, patch_artist=True)
...     plt.title(f"Boxplot of {column}")
...     plt.xlabel(column)
```



```
... plt.show()  
...
```