

# Python Coding Challenge

Name: PRIYESHWAR

Mail: [priyesh2664@gmail.com](mailto:priyesh2664@gmail.com)

---

## Loading and Printing rows of the Data

Read the CSV file into a DataFrame and preview the first few rows using head().

```
data=pd.read_csv('annual.csv')
```

```
data.head()
```

Loading and Printing rows of the Data							
In [4]:	<pre>data=pd.read_csv('annual.csv') data.head()</pre>						
	Year	Industry_aggregation_NZSIOC	Industry_code_NZSIOC	Industry_name_NZSIOC	Units	Variable_code	Variable_name
0	2023	Level 1	99999	All industries	Dollars (millions)	H01	Total income
1	2023	Level 1	99999	All industries	Dollars (millions)	H04	Sales, government funding, grants and subsidies
2	2023	Level 1	99999	All industries	Dollars (millions)	H05	Interest, dividends and donations
3	2023	Level 1	99999	All industries	Dollars (millions)	H07	Non-operating income
4	2023	Level 1	99999	All industries	Dollars (millions)	H08	Total expenditure

---

## Summary of Data Frame

Use info() to display data types and count of non-null values per column.

**data.info()**

# Summary of Data Frame

```
In [5]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50985 entries, 0 to 50984
Data columns (total 10 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Year                                50985 non-null  int64
1   Industry_aggregation_NZSIOC         50985 non-null  object
2   Industry_code_NZSIOC                50985 non-null  object
3   Industry_name_NZSIOC                50985 non-null  object
4   Units                              50985 non-null  object
5   Variable_code                       50985 non-null  object
6   Variable_name                       50985 non-null  object
7   Variable_category                   50985 non-null  object
8   Value                              50985 non-null  object
9   Industry_code_ANZSIC06              50985 non-null  object
dtypes: int64(1), object(9)
memory usage: 3.9+ MB
```

## Descriptive Statistical Measures of a DataFrame

Generate basic statistics like mean, median, and quartiles with describe().

**data.describe()**

Descriptive Statistical Measures of a DataFrame																			
In [7]:	data.describe()																		
	<table><thead><tr><th></th><th>Year</th></tr></thead><tbody><tr><td>count</td><td>50985.000000</td></tr><tr><td>mean</td><td>2018.000000</td></tr><tr><td>std</td><td>3.162309</td></tr><tr><td>min</td><td>2013.000000</td></tr><tr><td>25%</td><td>2015.000000</td></tr><tr><td>50%</td><td>2018.000000</td></tr><tr><td>75%</td><td>2021.000000</td></tr><tr><td>max</td><td>2023.000000</td></tr></tbody></table>		Year	count	50985.000000	mean	2018.000000	std	3.162309	min	2013.000000	25%	2015.000000	50%	2018.000000	75%	2021.000000	max	2023.000000
	Year																		
count	50985.000000																		
mean	2018.000000																		
std	3.162309																		
min	2013.000000																		
25%	2015.000000																		
50%	2018.000000																		
75%	2021.000000																		
max	2023.000000																		

## Finding Null Values

Check for missing data by summing nulls in each column using isnull().sum().

**data.isnull().sum()**

Finding Null Values																							
In [8]:	data.isnull().sum()																						
	<table><tbody><tr><td>Year</td><td>0</td></tr><tr><td>Industry_aggregation_NZSIOC</td><td>0</td></tr><tr><td>Industry_code_NZSIOC</td><td>0</td></tr><tr><td>Industry_name_NZSIOC</td><td>0</td></tr><tr><td>Units</td><td>0</td></tr><tr><td>Variable_code</td><td>0</td></tr><tr><td>Variable_name</td><td>0</td></tr><tr><td>Variable_category</td><td>0</td></tr><tr><td>Value</td><td>0</td></tr><tr><td>Industry_code_ANZSIC06</td><td>0</td></tr><tr><td>dtype:</td><td>int64</td></tr></tbody></table>	Year	0	Industry_aggregation_NZSIOC	0	Industry_code_NZSIOC	0	Industry_name_NZSIOC	0	Units	0	Variable_code	0	Variable_name	0	Variable_category	0	Value	0	Industry_code_ANZSIC06	0	dtype:	int64
Year	0																						
Industry_aggregation_NZSIOC	0																						
Industry_code_NZSIOC	0																						
Industry_name_NZSIOC	0																						
Units	0																						
Variable_code	0																						
Variable_name	0																						
Variable_category	0																						
Value	0																						
Industry_code_ANZSIC06	0																						
dtype:	int64																						

## Missing Data Handling

Remove rows containing any null values with `dropna()`.

## data.dropna()

Missing Data Handling								
In [9]: data.dropna()								
	Year	Industry_aggregation_NZSIOC	Industry_code_NZSIOC	Industry_name_NZSIOC	Units	Variable_code	Variable_name	
0	2023	Level 1	99999	All industries	Dollars (millions)	H01	Total income	
1	2023	Level 1	99999	All industries	Dollars (millions)	H04	Sales, government funding, grants and subsidies	
2	2023	Level 1	99999	All industries	Dollars (millions)	H05	Interest, dividends and donations	
3	2023	Level 1	99999	All industries	Dollars (millions)	H07	Non-operating income	
4	2023	Level 1	99999	All industries	Dollars (millions)	H08	Total expenditure	
...	...	...	...	...	...	...	...	
50980	2013	Level 3	ZZ11	Food product manufacturing	Percentage	H37	Quick ratio	
50981	2013	Level 3	ZZ11	Food product manufacturing	Percentage	H38	Margin on cost of goods for resale	
50982	2013	Level 3	ZZ11	Food product manufacturing	Percentage	H39	Return on assets	
50983	2013	Level 3	ZZ11	Food product manufacturing	Percentage	H40	Return on equity	

**Sorting DataFrame values**

Sort the DataFrame based on the 'Value' column using sort\_values().

**data.sort\_values(by='Value')**

Sorting DataFrame values								
In [12]:	data.sort_values(by='Value')							
	Year	Industry_aggregation_NZSIOC	Industry_code_NZSIOC	Industry_name_NZSIOC	Units	Variable_code	Variable	
27424	2018	Level 4	QQ111	Hospitals	Percentage	H40	Return o	assets
47238	2013	Level 4	CC411	Printing	Percentage	H40	Return o	assets
47202	2013	Level 3	CC41	Printing	Percentage	H40	Return o	assets
3270	2023	Level 4	KK121	Life Insurance	Percentage	H40	Return o	assets
32685	2016	Level 4	AA131	Dairy Cattle Farming	Percentage	H40	Return o	assets
...	...	...	...	...	...	...	...	...
49887	2013	Level 4	LL122	Non-Residential Property Operation	Dollars (millions)	H27	Addition	assets
31347	2017	Level 4	LL122	Non-Residential Property Operation	Dollars (millions)	H27	Addition	assets
40219	2015	Level 3	KK11	Finance	Dollars (millions)	H26	Fixed tan	assets
40221	2015	Level 3	KK11	Finance	Dollars (millions)	H28	Disposat	assets
40618	2015	Level 4	LL122	Non-Residential Property Operation	Dollars (millions)	H28	Disposat	assets

## Merge Data Frames

Combine sales and products tables on product\_id using inner, left, right, and outer joins.

### Inner Join

```
sales = pd.read_csv("sales.csv")
```

```
products = pd.read_csv("products.csv")
```

```
inner = pd.merge(sales, products, on="product_id", how="inner")
```

```
print( inner)
```

## Merge Data Frames

### Inner Join

```
In [17]: sales = pd.read_csv("sales.csv")
products = pd.read_csv("products.csv")

inner = pd.merge(sales, products, on="product_id", how="inner")
print( inner)
```

	sale_id	product_id	quantity	sale_date	product_name	category
0	1	101	2	2023-01-01	Laptop	Electronics
1	6	101	1	2023-01-06	Laptop	Electronics
2	2	102	1	2023-01-02	Keyboard	Electronics
3	3	103	4	2023-01-03	Mouse	Electronics
4	7	104	3	2023-01-07	Desk	Furniture

### Left Join

```
left = pd.merge(sales, products, on="product_id", how="left")  
print(left)
```

Left Join						
In [18]:	<pre>left = pd.merge(sales, products, on="product_id", how="left") print(left)</pre>					
	sale_id	product_id	quantity	sale_date	product_name	category
0	1	101	2	2023-01-01	Laptop	Electronics
1	2	102	1	2023-01-02	Keyboard	Electronics
2	3	103	4	2023-01-03	Mouse	Electronics
3	4	108	1	2023-01-04	NaN	NaN
4	5	109	2	2023-01-05	NaN	NaN
5	6	101	1	2023-01-06	Laptop	Electronics
6	7	104	3	2023-01-07	Desk	Furniture

### Right Join

```
right = pd.merge(sales, products, on="product_id", how="right")  
print(right)
```

Right Join						
In [19]:	<pre>right = pd.merge(sales, products, on="product_id", how="right") print(right)</pre>					
	sale_id	product_id	quantity	sale_date	product_name	category
0	1.0	101	2.0	2023-01-01	Laptop	Electronics
1	6.0	101	1.0	2023-01-06	Laptop	Electronics
2	2.0	102	1.0	2023-01-02	Keyboard	Electronics
3	3.0	103	4.0	2023-01-03	Mouse	Electronics
4	7.0	104	3.0	2023-01-07	Desk	Furniture
5	NaN	105	NaN	NaN	Chair	Furniture
6	NaN	106	NaN	NaN	Monitor	Electronics
7	NaN	107	NaN	NaN	Notebook	Stationery

## Outer Join

```
outer = pd.merge(sales, products, on="product_id", how="outer")  
print( outer)
```

Outer Join						
In [20]:						
outer = pd.merge(sales, products, on="product_id", how="outer") print( outer)						
	sale_id	product_id	quantity	sale_date	product_name	category
0	1.0	101	2.0	2023-01-01	Laptop	Electronics
1	6.0	101	1.0	2023-01-06	Laptop	Electronics
2	2.0	102	1.0	2023-01-02	Keyboard	Electronics
3	3.0	103	4.0	2023-01-03	Mouse	Electronics
4	4.0	108	1.0	2023-01-04	NaN	NaN
5	5.0	109	2.0	2023-01-05	NaN	NaN
6	7.0	104	3.0	2023-01-07	Desk	Furniture
7	NaN	105	NaN	NaN	Chair	Furniture
8	NaN	106	NaN	NaN	Monitor	Electronics
9	NaN	107	NaN	NaN	Notebook	Stationery

## Visualizing DataFrame

Plot total quantity sold over time with a line chart using Seaborn's lineplot().

```
import matplotlib.pyplot as plt  
import seaborn as sns  
plt.figure(figsize=(10, 6))  
sns.lineplot(data=outer, x='sale_date', y='quantity', marker='o')  
plt.title('Total Quantity Sold Over Time')  
plt.xlabel('Sale Date')  
plt.ylabel('Quantity Sold')  
plt.xticks(rotation=45)  
plt.tight_layout()  
plt.show()
```



