

```
1 from google.colab import files
2 uploaded = files.upload()

Choose Files flipkart_pro...0250405.csv
• flipkart_products_20250405.csv(text/csv) - 903362 bytes, last modified: 4/6/2025 - 100% done
Saving flipkart_products_20250405.csv to flipkart_products_20250405 (1).csv
```

```
1 import pandas as pd
2
3 # Replace the file name with your actual uploaded file name
4 df = pd.read_csv('flipkart_products_20250405.csv')
5 df.head()
```

↻

| | Product Name | Price (₹) | Rating (★) | Number of Buyers | Total Sold | Available Stock | Main Category | Sub Category | Discount (%) | Seller | Return Policy | |
|---|---|-----------|------------|------------------|------------|-----------------|---------------|--------------|--------------|------------------|---------------|--------------------------|
| 0 | Krishnamurthy-Devan Laboriosam Ultra Smartphon... | 142247.04 | 3.2 | 7348 | 4812 | 364 | Electronics | Smartphones | 45 | RetailNet | False | https://www.flipkart.com |
| 1 | Nanda-Mahal Dignissimos Lite Laptops 1 | 186922.43 | 4.1 | 2342 | 881 | 145 | Electronics | Laptops | 55 | Flipkart Assured | False | https://www.flipkart.cor |
| 2 | Choudhury LLC Amet Plus Decor 15 | 11843.41 | 5.0 | 739 | 2580 | 206 | Home | Decor | 58 | SuperComNet | True | https://www.flipkart.c |

◀ ▶

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
1 # Extracting a Series object
2 # Check for typos in the column name - It might be 'Product Name' instead of 'product_name'
3 title_series = df['Product Name']
4 print(type(title_series))
5 print(title_series.head())
```

↻

```
<class 'pandas.core.series.Series'>
0    Krishnamurthy-Devan Laboriosam Ultra Smartphon...
1              Nanda-Mahal Dignissimos Lite Laptops 1
2              Choudhury LLC Amet Plus Decor 15
3              Borah LLC Accusantium Lite Smartphones 9
4              Murty Inc Placeat Pro Smartwatches 8
Name: Product Name, dtype: object
```

```
1 # DataFrame info
2 print(type(df))
3 df.info()
```

↻

```
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Product Name          5000 non-null  object
1   Price (₹)             5000 non-null  float64
2   Rating (★)           5000 non-null  float64
3   Number of Buyers      5000 non-null  int64
4   Total Sold            5000 non-null  int64
5   Available Stock       5000 non-null  int64
6   Main Category         5000 non-null  object
7   Sub Category          5000 non-null  object
8   Discount (%)          5000 non-null  int64
9   Seller                5000 non-null  object
10  Return Policy         5000 non-null  bool
11  Product URL           5000 non-null  object
dtypes: bool(1), float64(2), int64(4), object(5)
memory usage: 434.7+ KB
```

```
1 # Selecting column (Series)
2 # Check if 'price' column exists, if not try 'Price'
3 print(df['Price']) if 'Price' in df.columns else print(df['price']) if 'price' in df.columns else print("Column 'price' or 'Price' not f
4
```

```

5 # Selecting multiple columns
6 # Ensure column names used here match the actual names in your DataFrame
7 print(df[['product_name', 'Price']].head()) if 'Price' in df.columns else print(df[['product_name', 'price']].head()) if 'price' in df.c
8
9
10 # Row selection using iloc
11 print(df.iloc[0]) # First row
12
13 # Row selection using loc
14 print(df.loc[0]) # Based on index label

```

Column 'price' or 'Price' not found in DataFrame
 Column 'price' or 'Price' not found in DataFrame

| | |
|------------------|---|
| Product Name | Krishnamurthy-Devan Laboriosam Ultra Smartphon... |
| Price (₹) | 142247.04 |
| Rating (★) | 3.2 |
| Number of Buyers | 7348 |
| Total Sold | 4812 |
| Available Stock | 364 |
| Main Category | Electronics |
| Sub Category | Smartphones |
| Discount (%) | 45 |
| Seller | RetailNet |
| Return Policy | False |
| Product URL | https://www.flipkart.com/Krishnamurthy-Devan-L... |

Name: 0, dtype: object

| | |
|------------------|---|
| Product Name | Krishnamurthy-Devan Laboriosam Ultra Smartphon... |
| Price (₹) | 142247.04 |
| Rating (★) | 3.2 |
| Number of Buyers | 7348 |
| Total Sold | 4812 |
| Available Stock | 364 |
| Main Category | Electronics |
| Sub Category | Smartphones |
| Discount (%) | 45 |
| Seller | RetailNet |
| Return Policy | False |
| Product URL | https://www.flipkart.com/Krishnamurthy-Devan-L... |

Name: 0, dtype: object

```

1 import pandas as pd
2 import numpy as np
3
4 # ... (Your existing code to load the DataFrame) ...
5
6 # Check if 'price' or 'Price' column exists in the DataFrame
7 price_column = 'price' if 'price' in df.columns else 'Price' if 'Price' in df.columns else None
8
9 if price_column:
10     # Apply np.log1p on prices (safe for 0 values)
11     df['log_price'] = np.log1p(df[price_column])
12     df[[price_column, 'log_price']].head()
13 else:
14     print("Column 'price' or 'Price' not found in DataFrame")

```

Column 'price' or 'Price' not found in DataFrame

```

1 # Check for nulls
2 df.isnull().sum()
3
4 # Drop rows with missing values
5 df_cleaned = df.dropna()
6
7 # Fill missing values
8 df_filled = df.fillna({'price': 0})
9
10 # Check which rows have null prices, handling potential column name variations
11 if 'price' in df.columns:
12     null_price_rows = df[df['price'].isnull()]
13 elif 'Price' in df.columns:
14     null_price_rows = df[df['Price'].isnull()]
15 else:
16     print("No 'price' or 'Price' column found for null check")
17     null_price_rows = pd.DataFrame() # Empty DataFrame if column not found
18
19 print(null_price_rows)
20 # Replace null product names with "Unknown"

```

```
12 # Check if 'product_name' or 'Product Name' column exists before filling NaNs
13 product_name_col = 'product_name' if 'product_name' in df.columns else 'Product Name' if 'Product Name' in df.columns else None
14
15 if product_name_col:
16     df[product_name_col] = df[product_name_col].fillna("Unknown")
17 else:
18     print("Column 'product_name' or 'Product Name' not found in DataFrame")
```

```
➡ No 'price' or 'Price' column found for null check
Empty DataFrame
Columns: []
Index: []
```

```
1 # Create a MultiIndex using category and brand
2 if 'category' in df.columns and 'brand' in df.columns:
3     df_multi = df.set_index(['category', 'brand'])
4     print(df_multi.head())
```