Notebook KeyError

```
import numpy as np
import pandas as pd


df = pd.read_csv("/content/bestsellers with categories.csv")
df
```

	Name	Author	User Rating	Reviews	Price	Year	Genre
0	10-Day Green Smoothie Cleanse	JJ Smith	4.7	17350	8	2016	Non Fiction
1	11/22/63: A Novel	Stephen King	4.6	2052	22	2011	Fiction
2	12 Rules for Life: An Antidote to Chaos	Jordan B. Peterson	4.7	18979	15	2018	Non Fiction
3	1984 (Signet Classics)	George Orwell	4.7	21424	6	2017	Fiction
4	5,000 Awesome Facts (About Everything!) (Natio...	National Geographic Kids	4.8	7665	12	2019	Non Fiction
...	...	...	...	...	...	...	...
545	Wrecking Ball (Diary of a Wimpy Kid Book 14)	Jeff Kinney	4.9	9413	8	2019	Fiction
546	You Are a Badass: How to Stop Doubting Your Gr...	Jen Sincero	4.7	14331	8	2016	Non Fiction
547	You Are a Badass: How to Stop Doubting Your Gr...	Jen Sincero	4.7	14331	8	2017	Non Fiction
548	You Are a Badass: How to Stop Doubting Your Gr...	Jen Sincero	4.7	14331	8	2018	Non Fiction
549	You Are a Badass: How to Stop Doubting Your Gr...	Jen Sincero	4.7	14331	8	2019	Non Fiction

550 rows × 7 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.head()
```

	Name	Author	User Rating	Reviews	Price	Year	Genre
0	10-Day Green Smoothie Cleanse	JJ Smith	4.7	17350	8	2016	Non Fiction
1	11/22/63: A Novel	Stephen King	4.6	2052	22	2011	Fiction
2	12 Rules for Life: An Antidote to Chaos	Jordan B. Peterson	4.7	18979	15	2018	Non Fiction
3	1984 (Signet Classics)	George Orwell	4.7	21424	6	2017	Fiction
4	5,000 Awesome Facts (About Everything!) (Natio...	National Geographic Kids	4.8	7665	12	2019	Non Fiction

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.shape
```

```
(550, 7)
```

```
df.columns
```

```
Index(['Name', 'Author', 'User Rating', 'Reviews', 'Price', 'Year', 'Genre'], dtype='object')
```

```
df.describe()
```

	User Rating	Reviews	Price	Year
count	550.000000	550.000000	550.000000	550.000000
mean	4.618364	11953.281818	13.100000	2014.000000
std	0.226980	11731.132017	10.842262	3.165156
min	3.300000	37.000000	0.000000	2009.000000
25%	4.500000	4058.000000	7.000000	2011.000000
50%	4.700000	8580.000000	11.000000	2014.000000
75%	4.800000	17253.250000	16.000000	2017.000000
max	4.900000	87841.000000	105.000000	2019.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550 entries, 0 to 549
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Name         550 non-null    object
 1   Author       550 non-null    object
 2   User Rating  550 non-null    float64
 3   Reviews      550 non-null    int64
 4   Price        550 non-null    int64
 5   Year         550 non-null    int64
 6   Genre        550 non-null    object
dtypes: float64(1), int64(3), object(3)
memory usage: 30.2+ KB
```

```
# Clean column names (strip spaces, lowercase, replace spaces with underscores)
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
```

```
# Handle missing values (drop rows with any nulls for simplicity)
df.dropna(inplace=True)


# Remove duplicates
df.drop_duplicates(inplace=True)


# Normalize text fields (example: author/title/category to lowercase and stripped)
text_columns = ['name', 'author', 'genre']
for col in text_columns:
    if col in df.columns:
        df[col] = df[col].str.strip().str.lower()


# Optional – convert 'year' to int and 'price' to float
if 'year' in df.columns:
    df['year'] = pd.to_numeric(df['year'], errors='coerce')
if 'price' in df.columns:
    df['price'] = pd.to_numeric(df['price'], errors='coerce')


# Display cleaned data info
print("\nCleaned Data Info:")
print(df.info())
```

```
Cleaned Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550 entries, 0 to 549
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   name         550 non-null    object
 1   author       550 non-null    object
 2   user_rating  550 non-null    float64
 3   reviews      550 non-null    int64
 4   price        550 non-null    int64
 5   year         550 non-null    int64
 6   genre        550 non-null    object
```

```
dtypes: float64(1), int64(3), object(3)
memory usage: 30.2+ KB
None
```

```
# Save cleaned data to the current working directory
df.to_csv('cleaned_bestsellers.csv', index=False)
print("\nCleaned data saved as 'cleaned_bestsellers.csv' in the current folder.")
```

```
Cleaned data saved as 'cleaned_bestsellers.csv' in the current folder.
```

```
# renaming columns
df.rename(columns={"Name": "Title", "Year": "Publication Year", "User Rating": "Rating"}, inplace=True)
```

```
# Converting data types
df["price"] = df["price"].astype(float)
```

```
# analyzing author popularity
author_counts = df['author'].value_counts()
print(author_counts)
```

```
author
jeff kinney                          12
suzanne collins                      11
gary chapman                         11
rick riordan                         11
american psychological association   10
                                     ..
maurice sendak                        1
cheryl strayed                        1
the staff of the late show with       1
geneen roth                           1
ken follett                           1
Name: count, Length: 248, dtype: int64
```

```
# avg rating by genre
avg_rating_by_genre = df.groupby("genre")["user_rating"].mean()
print(avg_rating_by_genre)
```

```
genre
fiction        4.648333
non fiction    4.595161
Name: user_rating, dtype: float64
```

```
# Export top selling authors to a CSV file
author_counts.head(10).to_csv("top_authors.csv")

# Export average rating by genre to a CSV file
avg_rating_by_genre.to_csv("avg_rating_by_genre.csv")
```

```
author_counts
```

author	count
jeff kinney	12
suzanne collins	11
gary chapman	11
rick riordan	11
american psychological association	10
...	...
maurice sendak	1
cheryl strayed	1
the staff of the late show with	1
geneen roth	1
ken follett	1

248 rows × 1 columns

dtype: int64

```
avg_rating_by_genre
```

genre	user_rating
fiction	4.648333
non fiction	4.595161

dtype: float64