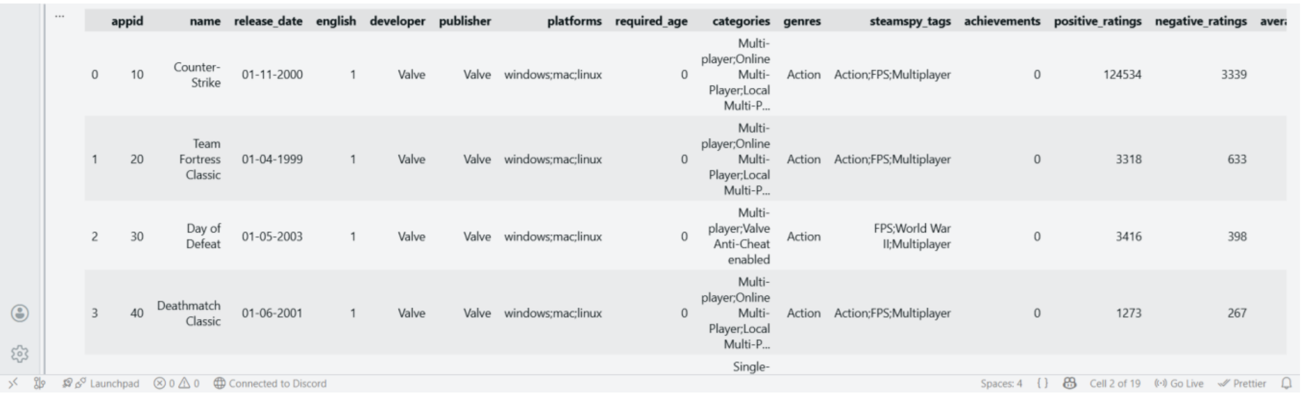
Loading and previewing the data

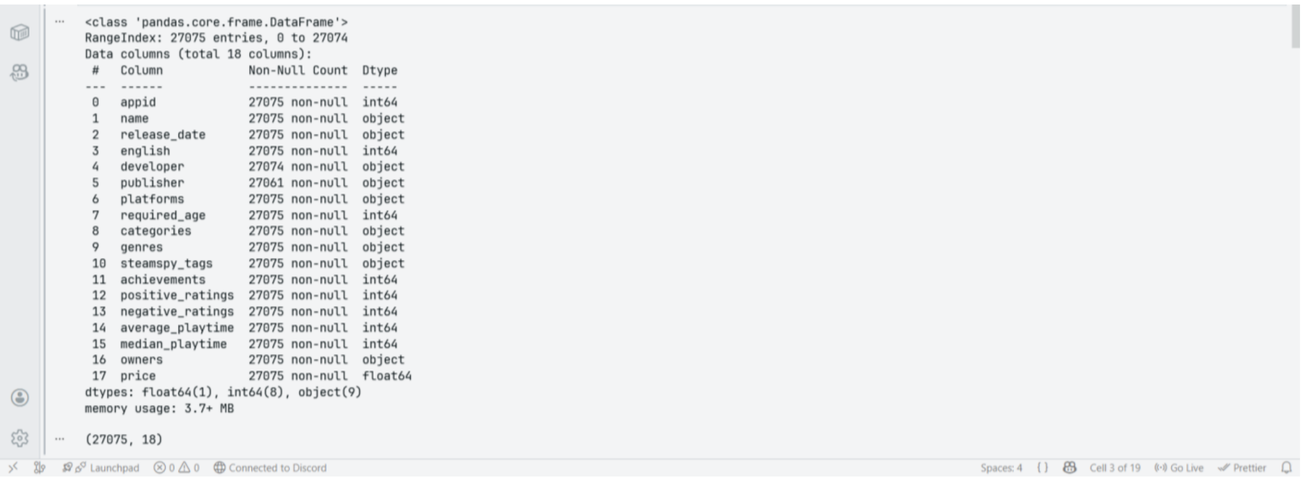
```
import pandas as pd
import numpy as np

df = pd.read_csv("steam.csv")
# Data preview
df.head()
```



	appid	name	release_date	english	developer	publisher	platforms	required_age	categories	genres	steamspy_tags	achievements	positive_ratings	negative_ratings	avera
0	10	Counter-Strike	01-11-2000	1	Valve	Valve	windows;mac;linux	0	Multi-player;Online Multi-Player;Local Multi-P...	Action	Action;FPS;Multiplayer	0	124534	3339	
1	20	Team Fortress Classic	01-04-1999	1	Valve	Valve	windows;mac;linux	0	Multi-player;Online Multi-Player;Local Multi-P...	Action	Action;FPS;Multiplayer	0	3318	633	
2	30	Day of Defeat	01-05-2003	1	Valve	Valve	windows;mac;linux	0	Multi-player;Valve Anti-Cheat enabled	Action	FPS;World War II;Multiplayer	0	3416	398	
3	40	Deathmatch Classic	01-06-2001	1	Valve	Valve	windows;mac;linux	0	Multi-player;Online Multi-Player;Local Multi-P...	Action	Action;FPS;Multiplayer	0	1273	267	
									Single-						

Analysing the data

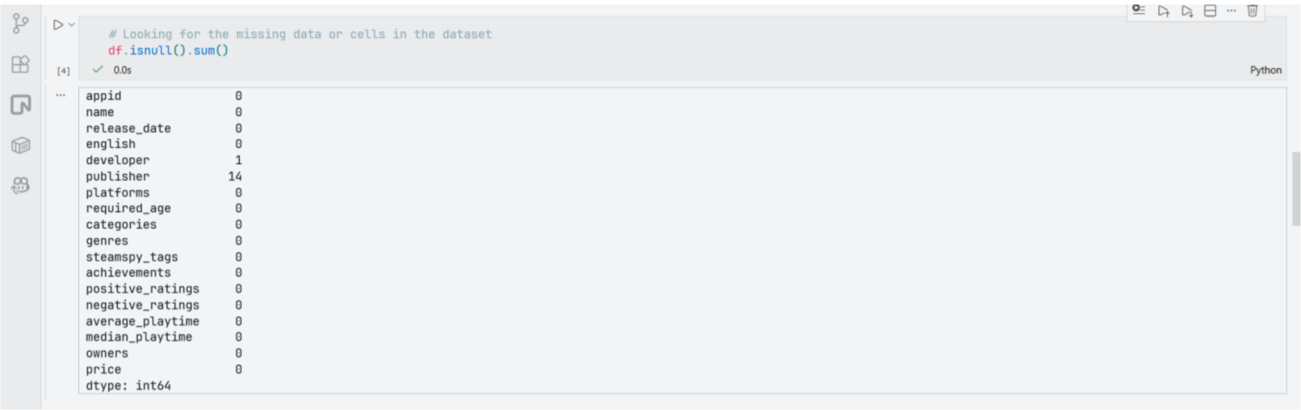
```
# Analyzing the structure of the dataset
df.info()
df.shape
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27075 entries, 0 to 27074
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   appid             27075 non-null  int64
 1   name              27075 non-null  object
 2   release_date      27075 non-null  object
 3   english           27075 non-null  int64
 4   developer         27074 non-null  object
 5   publisher         27061 non-null  object
 6   platforms         27075 non-null  object
 7   required_age      27075 non-null  int64
 8   categories        27075 non-null  object
 9   genres            27075 non-null  object
 10  steamspy_tags     27075 non-null  object
 11  achievements      27075 non-null  int64
 12  positive_ratings  27075 non-null  int64
 13  negative_ratings  27075 non-null  int64
 14  average_playtime  27075 non-null  int64
 15  median_playtime   27075 non-null  int64
 16  owners            27075 non-null  object
 17  price             27075 non-null  float64
dtypes: float64(1), int64(8), object(9)
memory usage: 3.7+ MB

(27075, 18)
```

Analysing the data

```
# Looking for the missing data or cells in the dataset
df.isnull().sum()
```

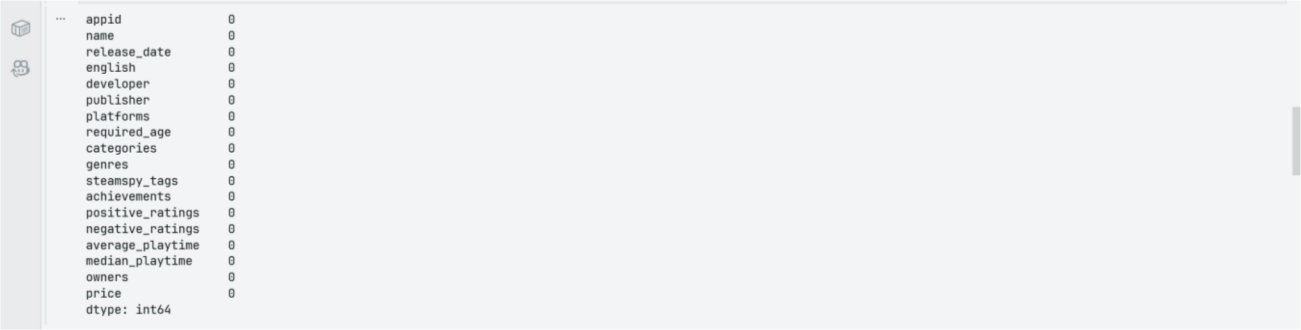


```
# Looking for the missing data or cells in the dataset
df.isnull().sum()
```

```
appid                0
name                 0
release_date         0
english              0
developer            1
publisher           14
platforms            0
required_age         0
categories           0
genres               0
steamspy_tags        0
achievements         0
positive_ratings     0
negative_ratings     0
average_playtime     0
median_playtime      0
owners               0
price                0
dtype: int64
```

Filling missing fields

```
# Missing developer means unknown - so filling them with "Anonymous"
df["developer"] = df["developer"].fillna("Unrated")
# Missing publisher means unknown - so filling them with "Anonymous"
df["publisher"] = df["publisher"].fillna("Anonymous")

# Verfying if the data is cleaned
df.isnull().sum()
```

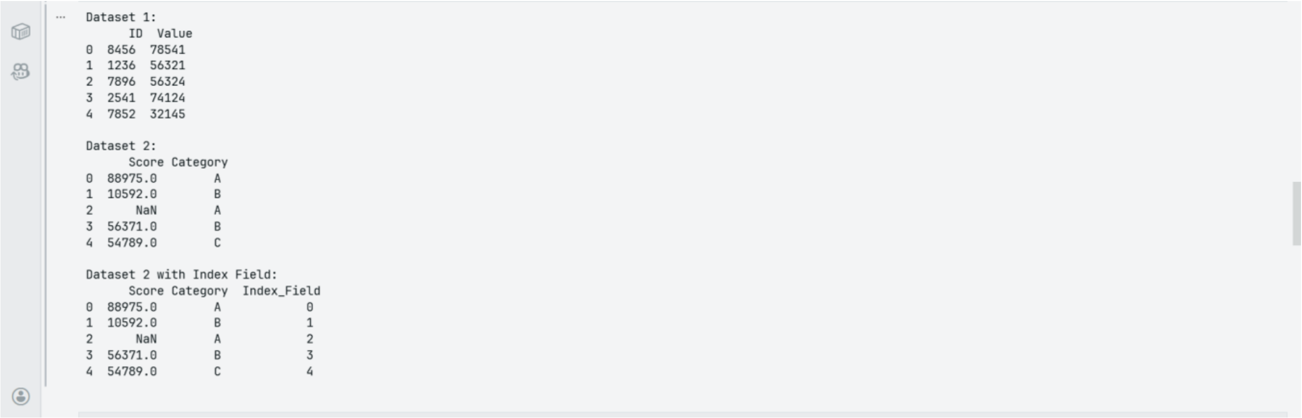


```
appid               0
name                0
release_date        0
english             0
developer           0
publisher           0
platforms           0
required_age        0
categories          0
genres              0
steamspy_tags       0
achievements        0
positive_ratings    0
negative_ratings    0
average_playtime    0
median_playtime     0
owners              0
price               0
dtype: int64
```

Calculating Outliers

```
import pandas as pd
import numpy as np

# Create sample Dataset 1
data1 = {'ID': [8456, 1236, 7896, 2541, 7852],
         'Value': [78541, 56321, 56324, 74124, 32145]}
df1 = pd.DataFrame(data1)

# Create sample Dataset 2 (with a potential missing value and an outlier)
data2 = {'Score': [88975, 10592, np.nan, 56371, 54789],
         'Category': ['A', 'B', 'A', 'B', 'C']}
df2 = pd.DataFrame(data2)
```

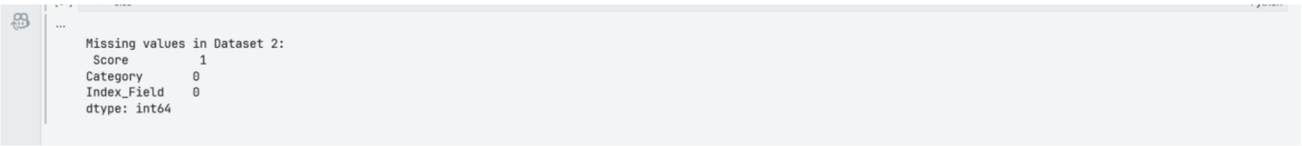
```
print("Dataset 1:\n", df1)
print("\nDataset 2:\n", df2)

df2['Index_Field'] = df1.index
print("\nDataset 2 with Index Field:\n", df2)
```



```
Dataset 1:
      ID  Value
0  8456  78541
1  1236  56321
2  7896  56324
3  2541  74124
4  7852  32145

Dataset 2:
      Score Category
0  88975.0        A
1  10592.0        B
2      NaN        A
3  56371.0        B
4  54789.0        C

Dataset 2 with Index Field:
      Score Category  Index_Field
0  88975.0        A            0
1  10592.0        B            1
2      NaN        A            2
3  56371.0        B            3
4  54789.0        C            4
```

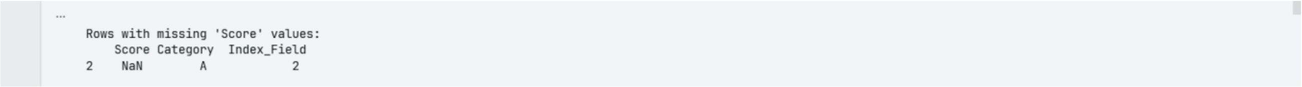
Checking for missing values

```
missing_values = df2.isnull().sum()
print("\nMissing values in Dataset 2:\n", missing_values)
```



```
Missing values in Dataset 2:
 Score          1
Category       0
Index_Field    0
dtype: int64
```

```
missing_rows = df2[df2['Score'].isnull()]
print("\nRows with missing 'Score' values:\n", missing_rows)
```



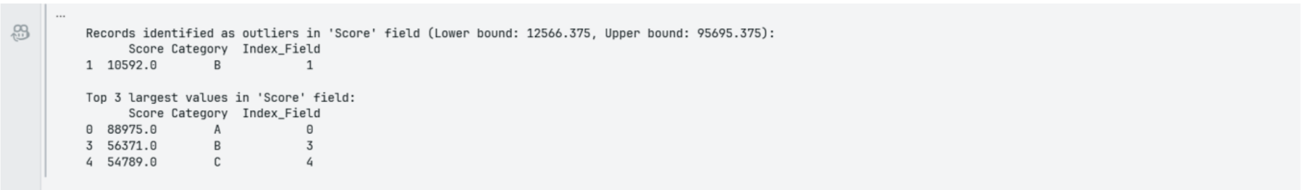
```
Rows with missing 'Score' values:
    Score Category  Index_Field
2    NaN        A            2
```

Calculating IQR

```
# Calculate Q1, Q3, and IQR for the 'Score' column
Q1 = df2['Score'].quantile(0.25)
Q3 = df2['Score'].quantile(0.75)
IQR = Q3 - Q1

# Define bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

```
outliers = df2[(df2['Score'] < lower_bound) | (df2['Score'] > upper_bound)]
print(f"\nRecords identified as outliers in 'Score' field (Lower bound: {lower_bound},
Upper bound: {upper_bound}):\n", outliers)

# Print a listing of the largest values for that field (e.g., top 3 largest)
largest_values = df2.nlargest(3, 'Score')
print("\nTop 3 largest values in 'Score' field:\n", largest_values)
```

Outliers



```
...
Records identified as outliers in 'Score' field (Lower bound: 12566.375, Upper bound: 95695.375):
      Score Category  Index_Field
1  10592.0        B            1

Top 3 largest values in 'Score' field:
      Score Category  Index_Field
0  88975.0        A            0
3  56371.0        B            3
4  54789.0        C            4
```