Data Science in Pratica

Kevin Roitero & Eddy Maddalena
University of Udine
kevin.roitero@uniud.it, eddy.maddalena@uniud.it

Python per la Data Science

- Python
  - linguaggio di programmazione ad alto livello, interpretato e generale
  - Semplice da imparare, con una sintassi chiara e leggibile
  - vasto ecosistema di librerie scientifiche e di data analysis
- Caratteristiche
  - curva di apprendimento quasi piatta
  - può essere utilizzato per una varietà di applicazioni, dai siti web ai giochi, dall'analisi dei dati all'apprendimento automatico
  - comunità attiva che contribuisce con nuove librerie e strumenti
  - Python può essere facilmente integrato con altri linguaggi come C, C++, JAVA
  - è attualmente uno dei linguaggi di programmazione più popolari e richiesti

Aiuto e Moduli

```
# aiuto sulla funzione log
help(numpy)
```

Python è dotato di una serie di moduli, alcuni caricati di default

```
# installare e aggiornare un modulo (da fare solo una volta)
pip install numpy

# importare un modulo
import numpy as np
```

Operatori e Funzioni

- aritmetici: somma (+), sottrazione (-), prodotto (*), divisione (/), divisione intera (//), modulo (%), esponente (**)
- comparazione: uguaglianza (==), non uguaglianza (!=), minore (<), maggiore (>), minore uguale (<=), maggiore uguale (>=)
- operatori logici: congiunzione (&, AND), disgiunzione (|, OR), negazione (NOT), disgiunzione esclusiva (^)
- indici statistici: media (mean), massimo (max), minimo (min)
- ...
- vedi https://www.programiz.com/python-programming/operators

Valori Speciali

- il valore `None` è usato per rappresentare valori mancanti
- il valore `np.inf` rappresenta il valore infinito
- il valore `np.nan` (not a number) è il risultato di una computazione che non ha senso

Variabili

È possibile utilizzare le variabili per memorizzare valori:

```
# assegnamento
x = 42

# stampa x
print(x)

# stampa tipo di x
type(x)
```

Tipi di Base

Python ha i seguenti tipi di base

```
# float (double-precision number)
x = 108.801

# integer (integer number)
x = 108

# string (stringa di caratteri)
x = "108L"

# bool (Booleano, True oppure False)
x = True
```

Strutture Dati

Python include le seguenti strutture dati:

- list
- tuple
- set
- dictionary
- array (numpy)
- dataframe (pandas)

Liste

Una lista è una sequenza di elementi che possono avere tipi diversi. Gli indici partono da 0.

```
# creare una lista
[1, 3, 5, 7]

# concatenare liste
[1, 3] + [5, 7]

# somma elemento per elemento con liste numpy
import numpy as np
np.array([1, 3]) + np.array([5, 7])
```

Tuple

Una tupla è una sequenza di elementi. Può essere usata per raggruppare più valori insieme.

```
# creare una tupla
x = (1, 3, 5, 7)

# ottenere il primo elemento
x[0]
```

Set

Un set è una collezione non ordinata di elementi unici. È utile quando si vogliono eliminare i duplicati da una lista o confrontare due liste.

```
# creare un set
{1, 3, 5, 7}

# rimuovere duplicati da una lista
set([1, 1, 3, 5, 5, 7])

# intersezione di due set
{1, 3, 5, 7} & {5, 7, 9}
```

Dizionario

Un dizionario è una collezione non ordinata di coppie chiave-valore. È simile a una mappa o a un hash-table in altri linguaggi di programmazione.

```
# creare un dizionario
d = {'a': 1, 'b': 2, 'c': 3}

# ottenere un valore da un dizionario
d['a']

# aggiungere o aggiornare un elemento di un dizionario
d['b'] = 3
d['d'] = 4
```

Array (Numpy)

Un array di numpy è una lista di valori dello stesso tipo. È come una lista base, ma ha funzionalità aggiuntive per l'analisi dei dati.

```
import numpy as np

# creare un array
np.array([1, 2, 3, 4, 5])

# somma elemento per elemento
np.array([1, 2]) + np.array([3, 4])
```

DataFrame (Pandas)

Un dataframe di pandas è una tabella di dati con righe e colonne. È come una tabella in un database SQL o un foglio di calcolo in Excel.

```
import pandas as pd

# creare un dataframe
pd.DataFrame({
    'colonna1': [1, 2, 3],
    'colonna2': ['a', 'b', 'c']
})

# ottenere una colonna
df['colonna1']

# filtrare le righe
df[df['colonna1'] > 2]
```

see https://www.educative.io/blog/pandas-cheat-sheet

Controllo del Flusso

Python fornisce diverse strutture di controllo del flusso, tra cui if, for e while.

```
# if
if x < 0:
    print("x è negativo")

# for
for i in range(5):
    print(i)

# while
while x < 0:
    print("x è ancora negativo")
    x += 1
```

Funzioni

Le funzioni sono blocchi di codice riutilizzabili che eseguono una determinata operazione.

```
# definizione di una funzione
def somma(x, y):
    return x + y

# chiamata di una funzione
somma(3, 4)
```

Seaborn

Seaborn è una libreria di visualizzazione dei dati in Python basata su matplotlib. Fornisce un'interfaccia di alto livello per disegnare grafici statistici attraenti e informativi.

Il Dataset dei Videogiochi

Stiamo lavorando con un dataset che comprende informazioni sui videogiochi, tra cui:

- Rank: posizione per numero di vendite
- Name: nome del gioco
- Platform: piattaforma (i.e. PC, PS4, etc.)
- Year: anno di rilascio
- Genre: genere
- Publisher: editore
- NA_Sales: vendite in Nord America (in milioni)
- EU_Sales: vendite in Europa (in milioni)
- JP_Sales: vendite in Giappone (in milioni)
- Other_Sales: vendite nel resto del mondo (in milioni)
- Global_Sales: totale delle vendite (mondiale)

Dataset

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	3.28	2.96	33
5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1	31.37
6	Tetris	GB	1989	Puzzle	Nintendo	23.2	2.26	4.22	0.58	30.26
7	New Super Mario Bros.	DS	2006	Platform	Nintendo	11.38	9.23	6.5	2.9	30.01
8	Wii Play	Wii	2006	Misc	Nintendo	14.03	9.2	2.93	2.85	29.02
9	New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.59	7.06	4.7	2.26	28.62
10	Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31
11	Nintendogs	DS	2005	Simulation	Nintendo	9.07	11	1.93	2.75	24.76
12	Mario Kart DS	DS	2005	Racing	Nintendo	9.81	7.57	4.13	1.92	23.42
13	Pokemon Gold/Pokemon Silver	GB	1999	Role-Playing	Nintendo	9	6.18	7.2	0.71	23.1
14	Wii Fit	Wii	2007	Sports	Nintendo	8.94	8.03	3.6	2.15	22.72
15	Wii Fit Plus	Wii	2009	Sports	Nintendo	9.09	8.59	2.53	1.79	22
16	Kinect Adventures!	X360	2010	Misc	Microsoft Game Studios	14.97	4.94	0.24	1.67	21.82
17	Grand Theft Auto V	PS3	2013	Action	Take-Two Interactive	7.01	9.27	0.97	4.14	21.4
18	Grand Theft Auto: San Andreas	PS2	2004	Action	Take-Two Interactive	9.43	0.4	0.41	10.57	20.81
19	Super Mario World	SNES	1990	Platform	Nintendo	12.78	3.75	3.54	0.55	20.61
20	Brain Age: Train Your Brain in Minutes a Day	DS	2005	Misc	Nintendo	4.75	9.26	4.16	2.05	20.22

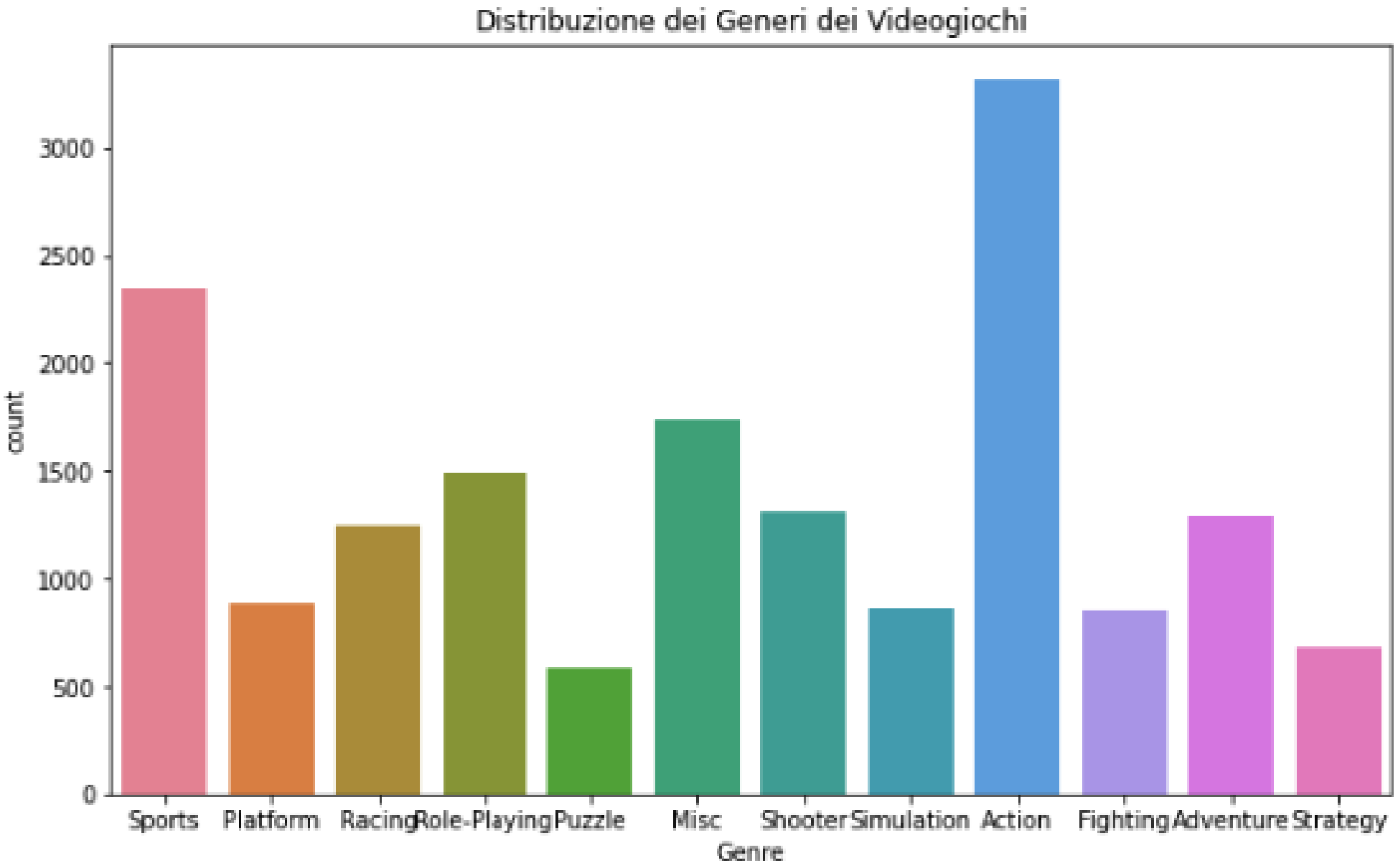
BarPlot dei Generi

```
# Importazione delle librerie
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Caricamento dei dati
df = pd.read_csv('videogames.csv')

# Barplot dei generi
plt.figure(figsize=(10,6))
sns.countplot(data=df, x='Genre')
plt.title('Distribuzione dei Generi dei Videogiochi')
plt.show()
```

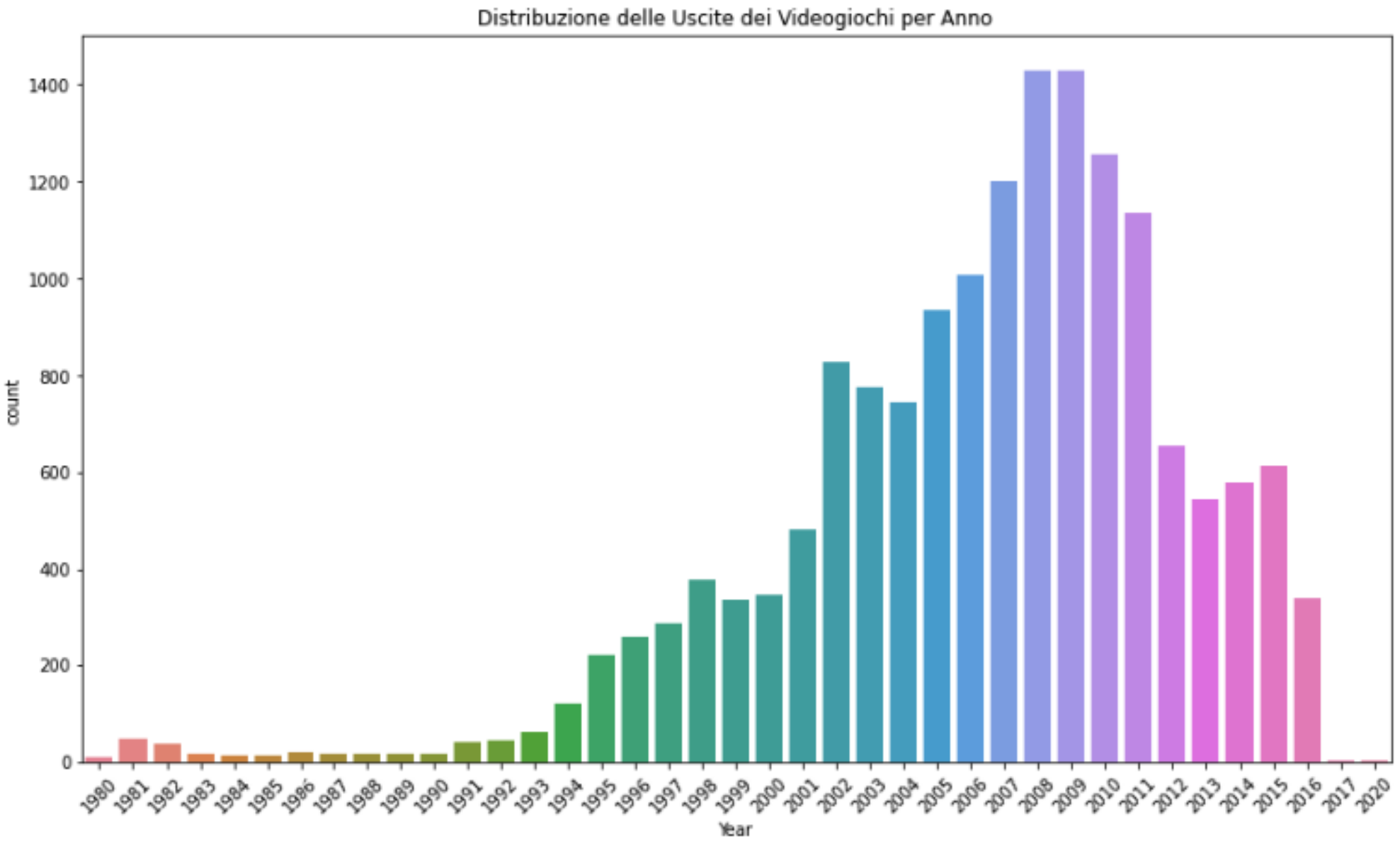
BarPlot dei Generi



BarPlot Per Anno

```
# Barplot per le uscite negli anni
plt.figure(figsize=(14,8))
sns.countplot(data=df, x='Year')
plt.title('Distribuzione delle Uscite dei Videogiochi per Anno')
plt.xticks(rotation=45)
plt.show()
```

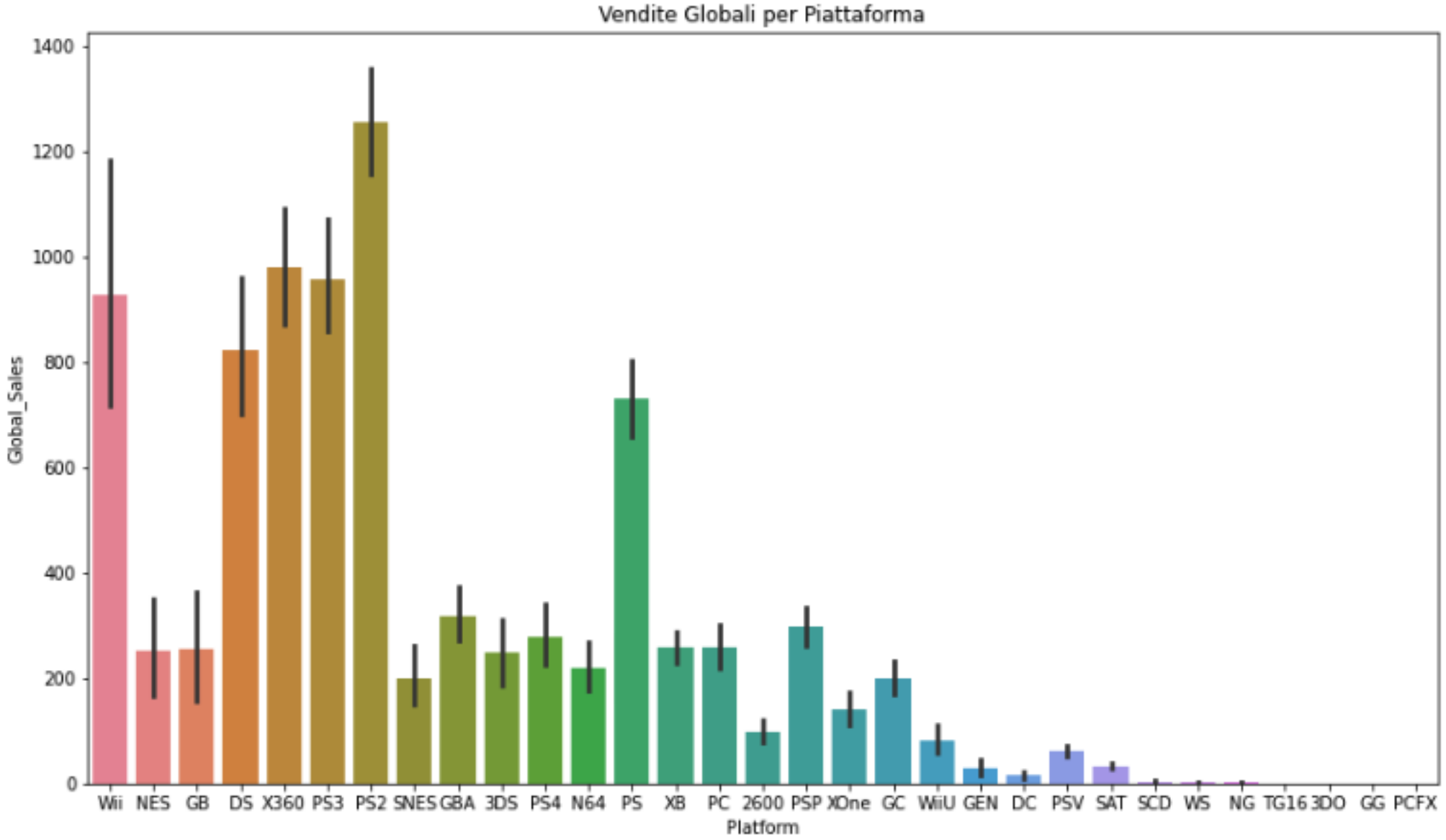
BarPlot Per Anno



Piattaforma

```
# Vendite per piattaforma
plt.figure(figsize=(14,8))
sns.barplot(data=df, x='Platform', y='Global_Sales', estimator=sum)
plt.title('Vendite Globali per Piattaforma')
plt.show()
```

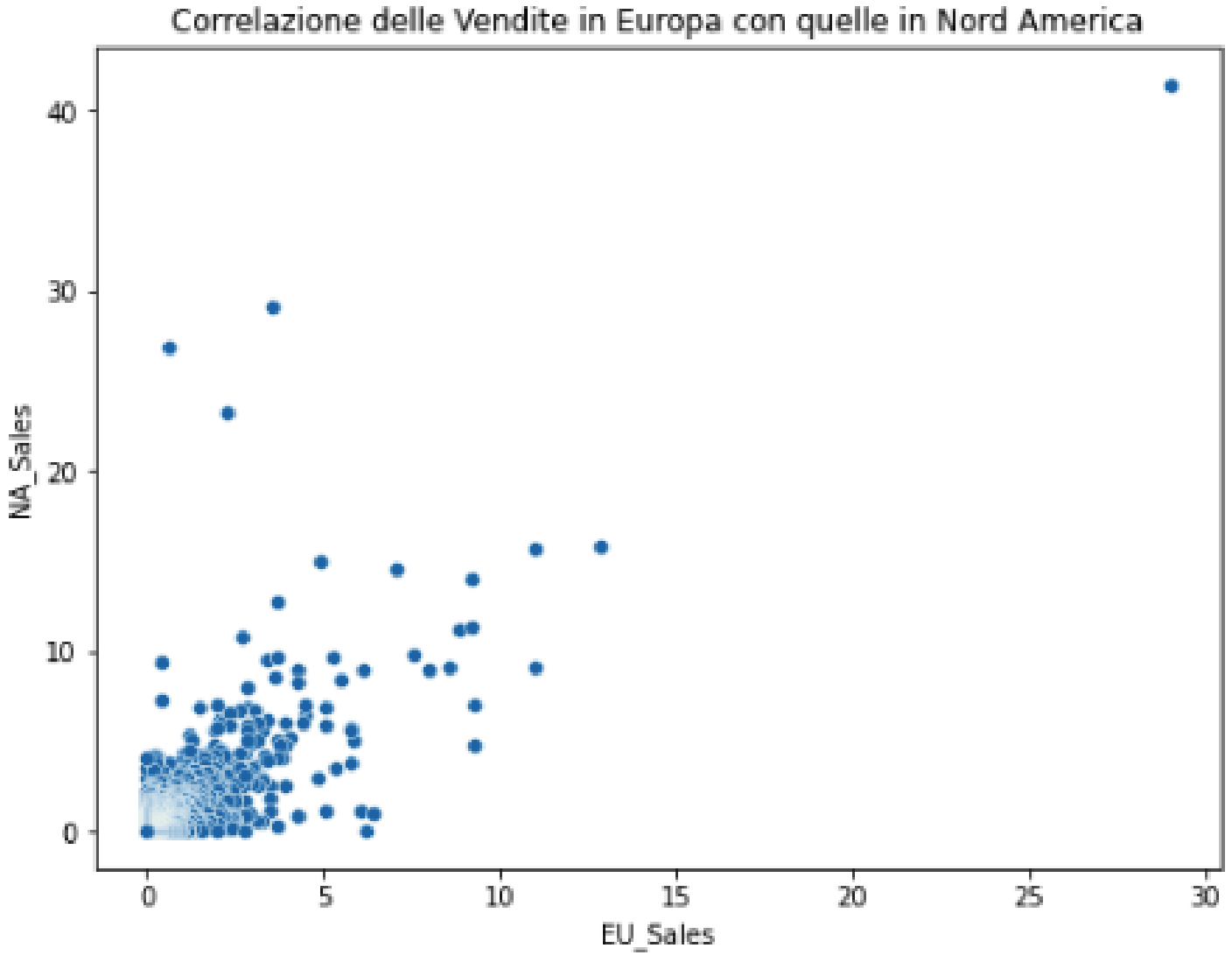
Piattaforma



Correlazione Generi

```
# Correlazione delle vendite EU con NA
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x='EU_Sales', y='NA_Sales')
plt.title('Correlazione delle Vendite in Europa con quelle in Nord America')
plt.show()
```

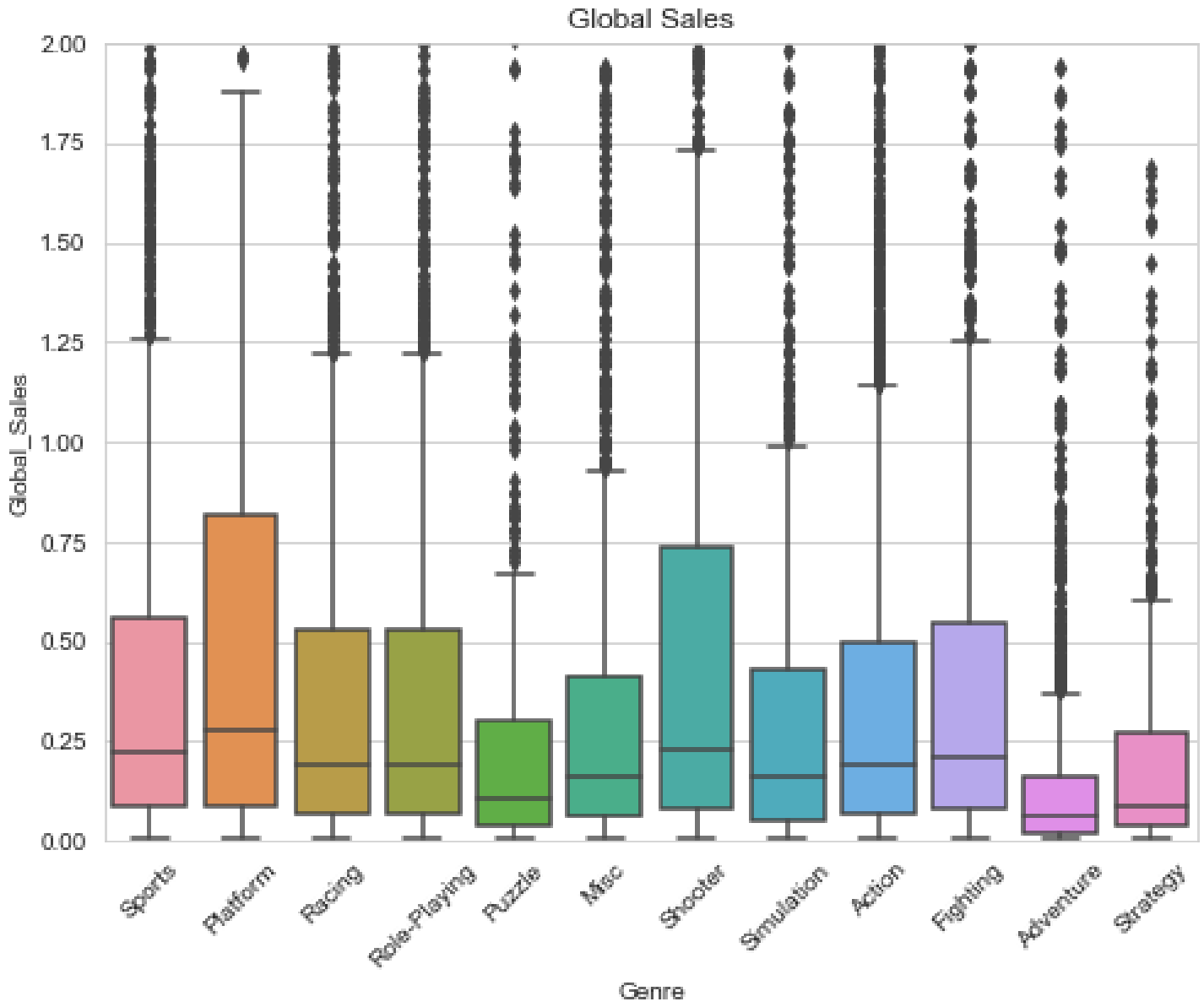
Correlazioni Generi



Vendite globali per Genere

```
# distribuzione delle vendite globali
plt.figure(figsize=(8,6))
sns.boxplot(x='Genre', y='Global_Sales', data=df)
plt.title('Global Sales')
plt.ylim(0, 2)
plt.xticks(rotation=45)
plt.show()
```

Vendite globali per Genere



Trend delle Vendite per Genere

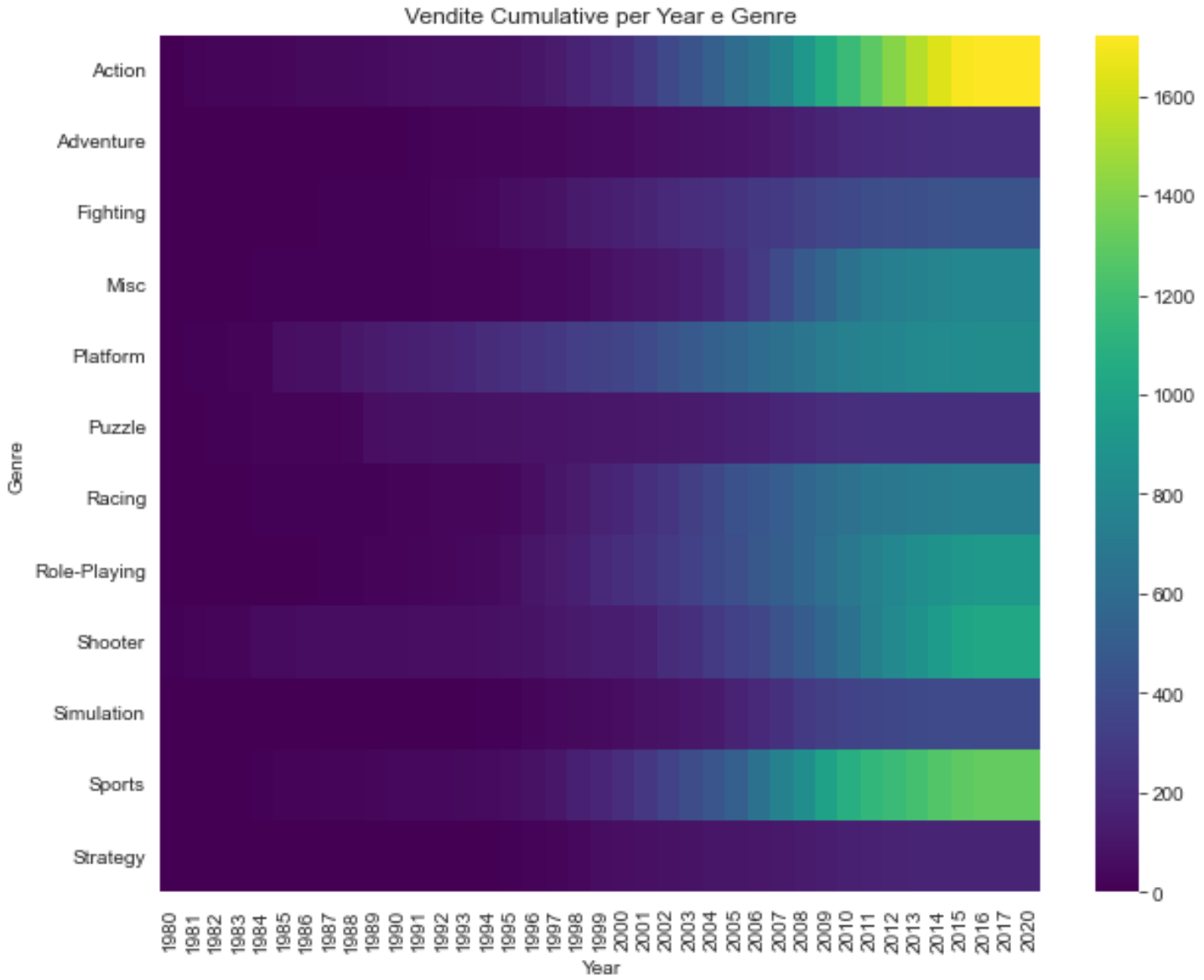
```
# trend delle vendite per genere
plt.figure(figsize=(8,6))
# filtriamo il dataset
action_df = df[df['Genre'] == 'Action']
# raggruppiamo e sommiano
yearly_sales = action_df.groupby('Year')['Global_Sales'].sum().reset_index()
# plottiamo
sns.lineplot(x='Year', y='Global_Sales', data=yearly_sales)
plt.title('Trend for Global Sales di Giochi Action')
plt.show()
```

Trend delle Vendite per Genere

HeatMap

```
# Heatmap
# creiamo una pivot table
pivot = df.pivot_table(values='Global_Sales', index='Genre', columns='Year', aggfunc='sum', fill_value=0)
# somma cumulativa
cumulative_sales = pivot.cumsum(axis=1)
# heatmap
plt.figure(figsize=(10, 8))  # increase figure size for better visibility
sns.heatmap(cumulative_sales, cmap='viridis')
plt.title('Vendite Cumulative per Year e Genre')
plt.show()
```

HeatMap


Vendite Cumulative per Year e Genre
Genre
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role-Playing
Shooter
Simulation
Sports
Strategy
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2020
Year
1600
1400
1200
1000
800
600
400
200
0

Cloud

```
# Importazione delle librerie
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Concatenazione dei titoli dei giochi in un'unica stringa
titles = ' '.join(df['Name'])

# Creazione del Word Cloud
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(titles)

# Visualizzazione del Word Cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.title('Word Cloud dei Titoli dei Videogiochi')
plt.show()
```

Cloud

Word Cloud dei Titoli dei Videogiochi

Risorse

- Galleria di esempi di Seaborn: https://seaborn.pydata.org/examples/index.html
- Dataset Kaggle: https://www.kaggle.com/datasets
- Dataset Italiani: https://github.com/italia/awesome-italian-public-datasets
- Dataset Huggingface: https://huggingface.co/datasets
- Collezione di collezioni di dataset: https://medium.com/analytics-vidhya/top-100-open-source-datasets-for-data-science-cd5a8d67cc3d

Esercizi

- provate ora a replicare i nostri risultati eseguendo il notebook fornito
- sperimentate con il dataset dei videogiochi
- cosa succede se voglio correlare le vendite EU con quelle Globali? È corretto?
- come posso vedere le vendite massime dei videogiochi nel corso degli anni?
- ponentevi delle domande “interessanti” e provate a rispondere
- provate ad analizzare un altro dataset di vostro interesse