UNIVERSIDAD TECNOLÓGICA DE SAN LUIS RIO COLORADO

GUIA

MTRO. AURELIO FLORES

ALUMNO: VICTOR MANUEL GALVAN COVARRUBIAS

ING. EN DESARROLLO Y GESTIÓN DE SOFTWARE

San Luis Rio Colorado, Sonora Octubre, 2022

UNCODING THE FUTURE

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, speech recognition and machine vision.

In general, AI systems work by ingesting large amounts of labeled training data, analyzing the data for correlations and patterns, and using these patterns to make predictions about future states. In this way, a Chabot that is fed examples of text chats can learn to produce lifelike exchanges with people, or an image recognition tool can learn to identify and describe objects in images by reviewing millions of examples.

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Data mining is defined as a process used to extract usable data from a larger set of any raw data. It implies analysing data patterns in large batches of data using one or more software. Data mining has applications in multiple fields, like science and research. As an application of data mining, businesses can learn more about their customers and develop more effective strategies related to various business functions and in turn leverage resources in a more optimal and insightful manner.

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size.

A. Artificial Intelligence
    a. Voice Assistants
    b. Autonomous Vehicles
    c. Facial Recognition
B. Machine learning
    a. Videos Surveillance
    b. Social Media
    c. Email Spamming
C. Data mining
    a. Retail Industry
    b. Crypto currency
    c. Criminal Investigation
D. Big data.
    a. Transportation
    b. Education
    c. Government

A. Pandas

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.
conda install pandas

B. Numpy

NumPy is a Python library used for working with arrays.
It also has functions for working in domain of linear algebra, fourier transform, and matrices.
NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
NumPy stands for Numerical Python.

C. Scipy

SciPy in Python is an open-source library used for solving mathematical, scientific, engineering, and technical problems. It allows users to manipulate the data and visualize the data using a wide range of high-level Python commands. SciPy is built on the Python NumPy extention. SciPy is also pronounced as “Sigh Pi.”

D. Scikit-learn

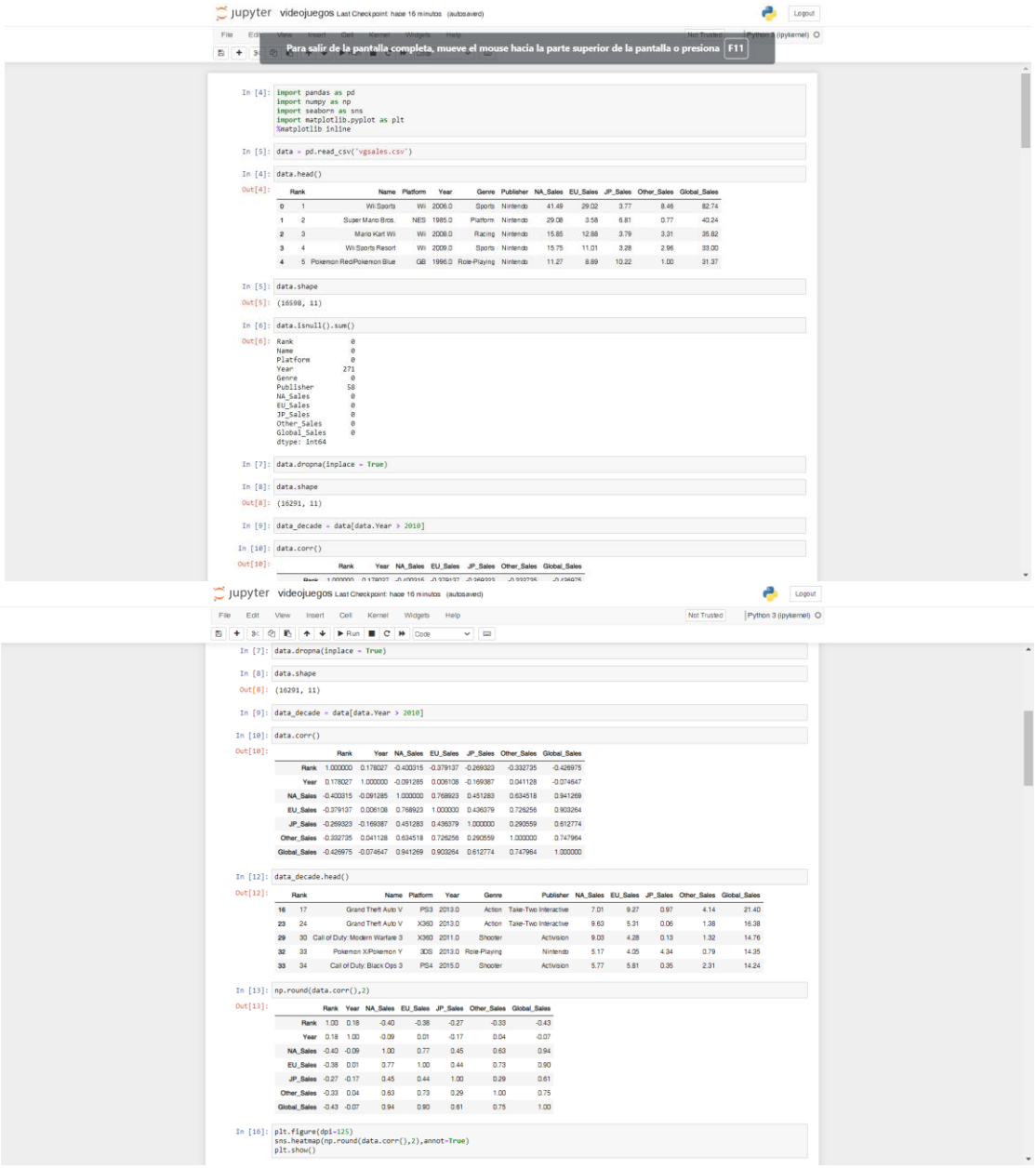
Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

E. Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

F. TensorFlow

TensorFlow gives us an interactive multiplatform programming interface which is scalable and much stable when compared to other **deep learning** libraries available, which are still very experimental.



jupyter videojuegos Last Checkpoint: hace 16 minutos (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Para salir de la pantalla completa, mueve el mouse hacia la parte superior de la pantalla o presiona F11

```
In [4]: import pandas as pd
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
In [5]: data = pd.read_csv('vgsales.csv')
```

```
In [4]: data.head()
```

Out[4]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37

```
In [5]: data.shape
```

```
Out[5]: (16598, 11)
```

```
In [6]: data.isnull().sum()
```

```
Out[6]: Rank            0
        Name            0
        Platform        0
        Year          271
        Genre           0
        Publisher      58
        NA_Sales        0
        EU_Sales        0
        JP_Sales        0
        Other_Sales     0
        Global_Sales    0
        dtype: int64
```

```
In [7]: data.dropna(inplace = True)
```

```
In [8]: data.shape
```

```
Out[8]: (16291, 11)
```

```
In [9]: data_decade = data[data.Year > 2010]
```

```
In [10]: data.corr()
```

Out[10]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.178027	-0.400315	-0.379137	-0.269323	-0.332735	-0.426975
Year	0.178027	1.000000	-0.091285	0.006108	-0.169387	0.041128	-0.074647
NA_Sales	-0.400315	-0.091285	1.000000	0.768923	0.451283	0.634518	0.941269
EU_Sales	-0.379137	0.006108	0.768923	1.000000	0.436379	0.726256	0.903264
JP_Sales	-0.269323	-0.169387	0.451283	0.436379	1.000000	0.290559	0.612774
Other_Sales	-0.332735	0.041128	0.634518	0.726256	0.290559	1.000000	0.747964
Global_Sales	-0.426975	-0.074647	0.941269	0.903264	0.612774	0.747964	1.000000

```
In [12]: data_decade.head()
```

Out[12]:

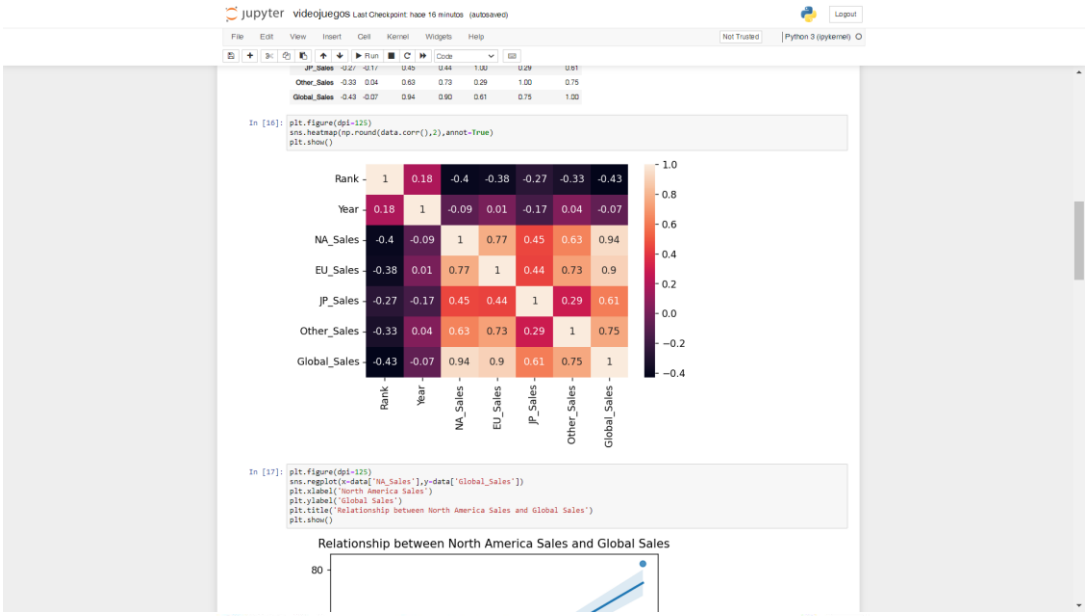
	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
16	17	Grand Theft Auto V	PS3	2013.0	Action	Take-Two Interactive	7.01	9.27	0.97	4.14	21.40
23	24	Grand Theft Auto V	X360	2013.0	Action	Take-Two Interactive	9.63	5.31	0.06	1.38	16.38
29	30	Call of Duty: Modern Warfare 3	X360	2011.0	Shooter	Activision	9.03	4.28	0.13	1.32	14.76
32	33	Pokemon X/Pokemon Y	3DS	2013.0	Role-Playing	Nintendo	5.17	4.05	4.34	0.79	14.35
33	34	Call of Duty: Black Ops 3	PS4	2015.0	Shooter	Activision	5.77	5.81	0.35	2.31	14.24

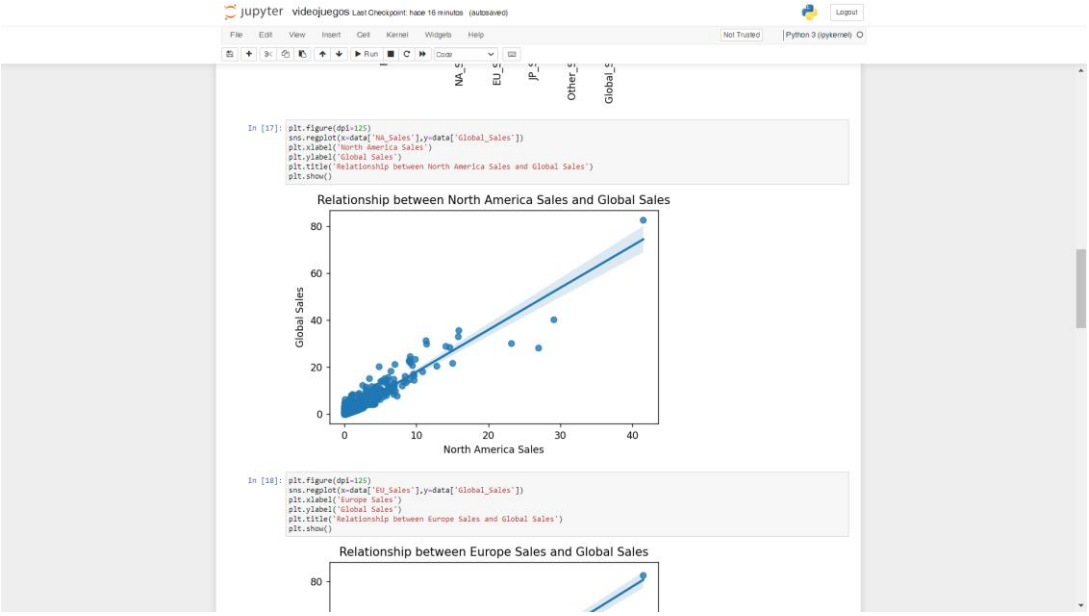
```
In [13]: np.round(data.corr(),2)
```

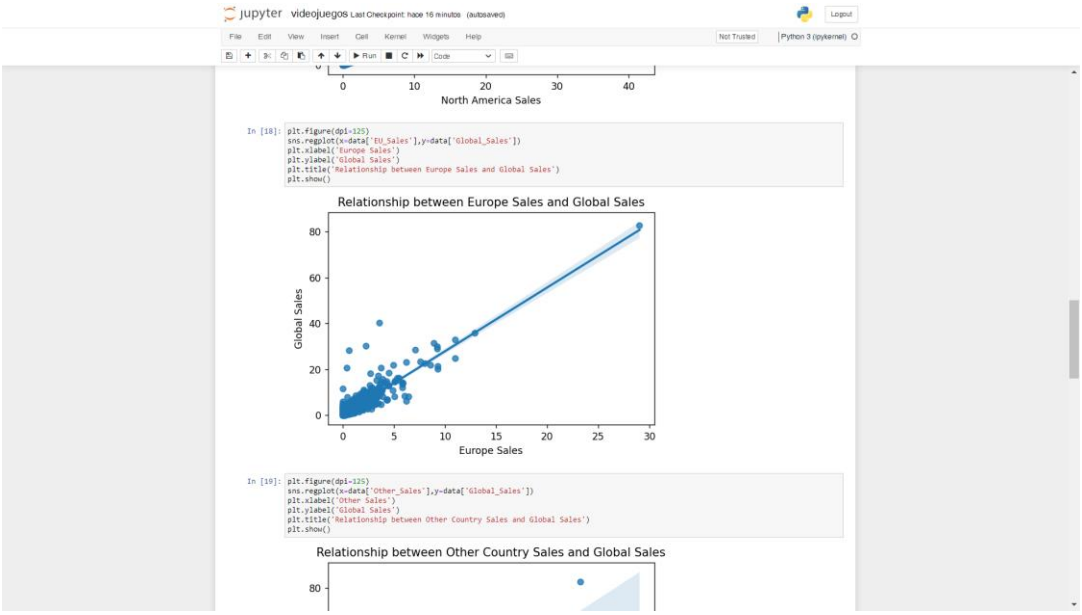
Out[13]:

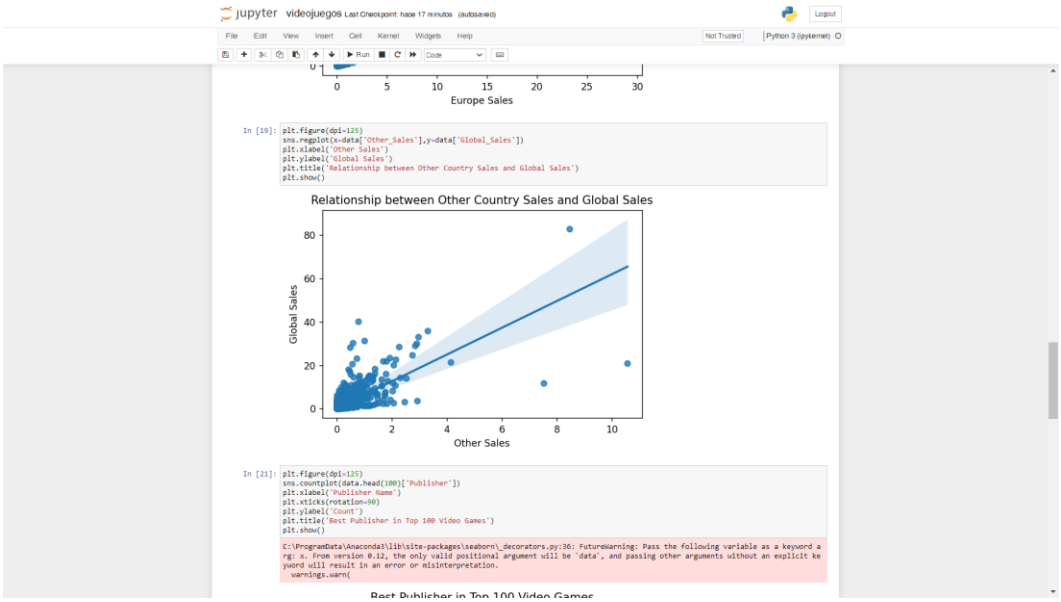
	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.00	0.18	-0.40	-0.38	-0.27	-0.33	-0.43
Year	0.18	1.00	-0.09	0.01	-0.17	0.04	-0.07
NA_Sales	-0.40	-0.09	1.00	0.77	0.45	0.63	0.94
EU_Sales	-0.38	0.01	0.77	1.00	0.44	0.73	0.90
JP_Sales	-0.27	-0.17	0.45	0.44	1.00	0.29	0.61
Other_Sales	-0.33	0.04	0.63	0.73	0.29	1.00	0.75
Global_Sales	-0.43	-0.07	0.94	0.90	0.61	0.75	1.00

```
In [16]: plt.figure(dpi=125)
         sns.heatmap(np.round(data.corr(),2),annot=True)
         plt.show()
```


jupyter videojuegos Last Checkpoint: hace 16 minutos (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Other_Sales 0.33 0.04 0.63 0.73 0.29 1.00 0.75
Global_Sales 0.43 0.07 0.94 0.90 0.61 0.75 1.00
In [16]:
plt.figure(dpi=125)
sns.heatmap(np.round(data.corr(),2),annot=True)
plt.show()
Rank 1 0.18 -0.4 -0.38 -0.27 -0.33 -0.43
Year 0.18 1 -0.09 0.01 -0.17 0.04 -0.07
NA_Sales -0.4 -0.09 1 0.77 0.45 0.63 0.94
EU_Sales -0.38 0.01 0.77 1 0.44 0.73 0.9
JP_Sales -0.27 -0.17 0.45 0.44 1 0.29 0.61
Other_Sales -0.33 0.04 0.63 0.73 0.29 1 0.75
Global_Sales -0.43 -0.07 0.94 0.9 0.61 0.75 1
Rank Year NA_Sales EU_Sales JP_Sales Other_Sales Global_Sales
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4
In [17]:
plt.figure(dpi=125)
sns.regplot(x=data['NA_Sales'],y=data['Global_Sales'])
plt.xlabel('North America Sales')
plt.ylabel('Global Sales')
plt.title('Relationship between North America Sales and Global Sales')
plt.show()
Relationship between North America Sales and Global Sales
80


jupyter videojuegos Last Checkpoint: hace 16 minutos (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [17]:
plt.figure(dpi=125)
sns.regplot(x=data['NA_Sales'],y=data['Global_Sales'])
plt.xlabel('North America Sales')
plt.ylabel('Global Sales')
plt.title('Relationship between North America Sales and Global Sales')
plt.show()
Relationship between North America Sales and Global Sales
Global Sales
0 20 40 60 80
0 10 20 30 40
North America Sales
In [18]:
plt.figure(dpi=125)
sns.regplot(x=data['EU_Sales'],y=data['Global_Sales'])
plt.xlabel('Europe Sales')
plt.ylabel('Global Sales')
plt.title('Relationship between Europe Sales and Global Sales')
plt.show()
Relationship between Europe Sales and Global Sales
80


jupyter videojuegos Last Checkpoint: hace 16 minutos (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0 10 20 30 40
North America Sales
In [18]:
plt.figure(dpi=125)
sns.regplot(x=data['EU_Sales'],y=data['Global_Sales'])
plt.xlabel('Europe Sales')
plt.ylabel('Global Sales')
plt.title('Relationship between Europe Sales and Global Sales')
plt.show()
Relationship between Europe Sales and Global Sales
Global Sales
0 20 40 60 80
0 5 10 15 20 25 30
Europe Sales
In [19]:
plt.figure(dpi=125)
sns.regplot(x=data['Other_Sales'],y=data['Global_Sales'])
plt.xlabel('Other Sales')
plt.ylabel('Global Sales')
plt.title('Relationship between Other Country Sales and Global Sales')
plt.show()
Relationship between Other Country Sales and Global Sales
80


jupyter videojuegos Last Checkpoint: hace 17 minutos (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0 5 10 15 20 25 30
Europe Sales
In [19]:
plt.figure(dpi=125)
sns.regplot(x=data['Other_Sales'],y=data['Global_Sales'])
plt.xlabel('Other Sales')
plt.ylabel('Global Sales')
plt.title('Relationship between Other Country Sales and Global Sales')
plt.show()
Relationship between Other Country Sales and Global Sales
Global Sales
0 20 40 60 80
0 2 4 6 8 10
Other Sales
In [21]:
plt.figure(dpi=125)
sns.countplot(data.head(100)['Publisher'])
plt.xlabel('Publisher Name')
plt.xticks(rotation=90)
plt.ylabel('Count')
plt.title('Best Publisher in Top 100 Video Games')
plt.show()
C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(
Best Publisher in Top 100 Video Games


jupyter videojuegos Last Checkpoint: hace 17 minutos (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0 2 4 6 8 10
Other Sales
In [21]:
plt.figure(dpi=125)
sns.countplot(data.head(100)['Publisher'])
plt.xlabel('Publisher Name')
plt.xticks(rotation=90)
plt.ylabel('Count')
plt.title('Best Publisher in Top 100 Video Games')
plt.show()
C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(
Best Publisher in Top 100 Video Games
Count
0 10 20 30 40 50
Nintendo
Microsoft Game Studios
Take-Two Interactive
Activision
Ubisoft
Bethesda Softworks
Electronic Arts
Sega
SquareSoft
Atari
jupyter videojuegos Last Checkpoint: hace 17 minutos (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Count
0.0 2.5 5.0 7.5 10.0
Take-Two Interactive
Activision
Nintendo
Ubisoft
Microsoft Game Studios
Bethesda Softworks
Electronic Arts
Sony Computer Entertainment
Sony Computer Entertainment Europe
Warner Bros. Interactive Entertainment
Sega
Namco Bandai Games
Capcom
Konami Digital Entertainment
Publisher Name
In [6]:
from wordcloud import WordCloud
ModuleNotFoundError Traceback (most recent call last)
Input In [6], in <cell line: 1>()
----> 1 from wordcloud import WordCloud
ModuleNotFoundError: No module named 'wordcloud'
In [ ]: