SEARCHING AND SLICING A VIDEO GAMES DATASET

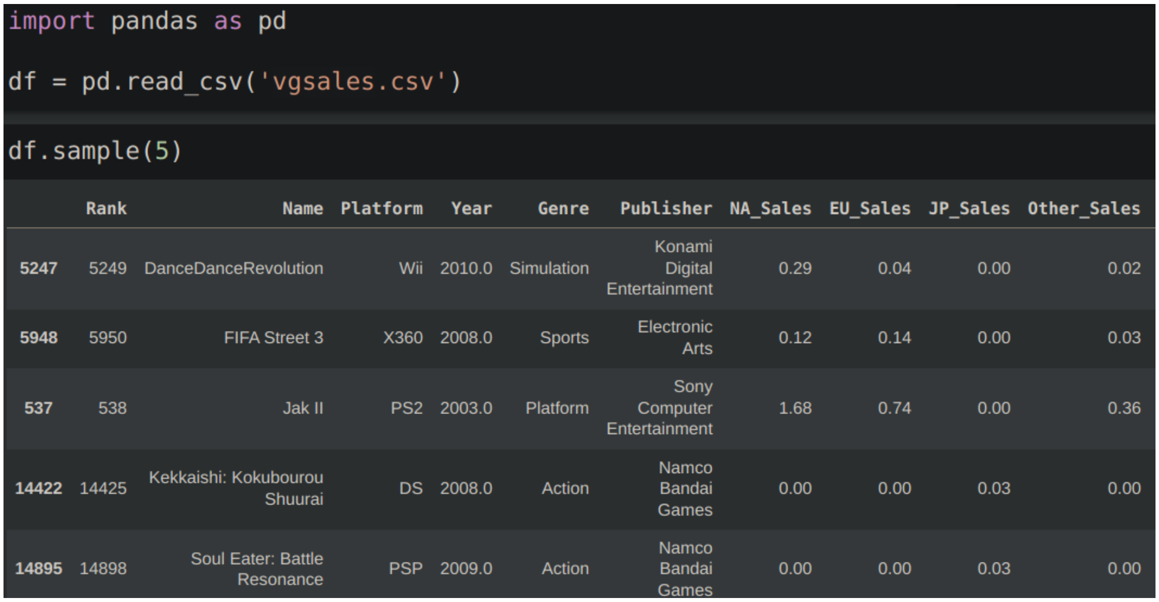
WITH PYTHON
BY DR. ALVIN ANG

CONTENTS

STEP 1: IMPORTING THE DATASET

https://www.alvinang.sg/s/vgsales.csv

https://www.alvinang.sg/s/Searching_and_Slicing_a_Video_Games_Dataset_with_Python_by_Dr_Alvin_Ang.ipynb

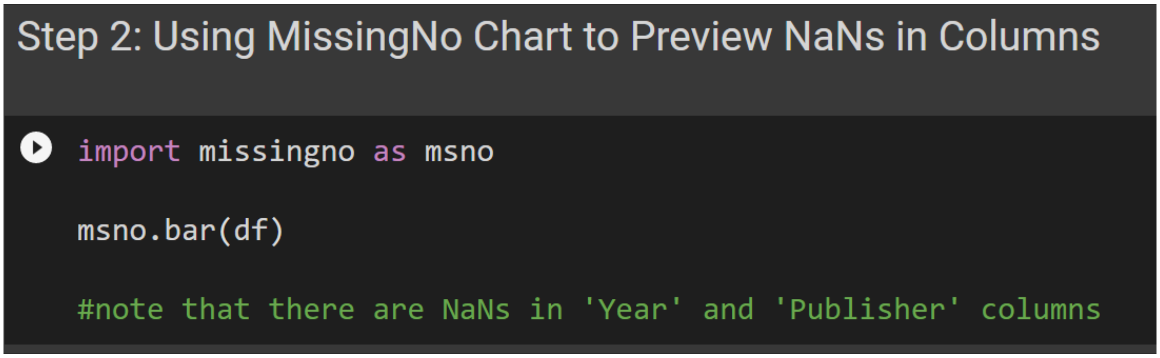


```
import pandas as pd

df = pd.read_csv('vgsales.csv')
```

```
df.sample(5)
```

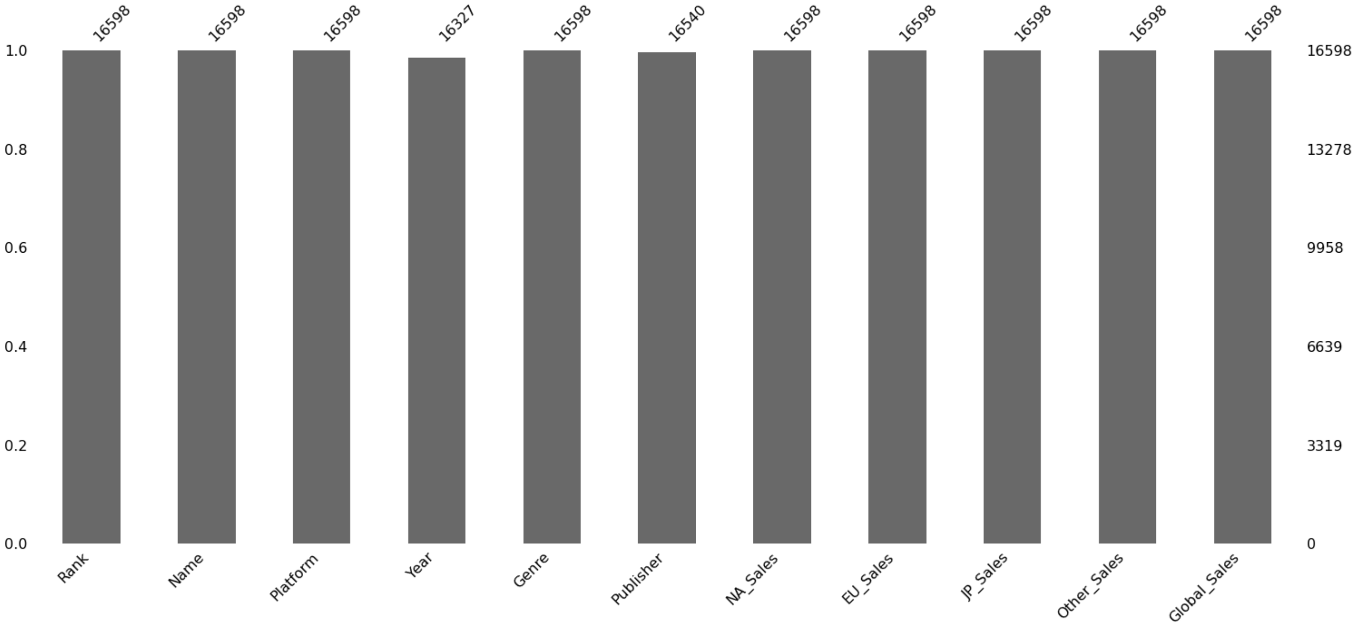
	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
5247	5249	DanceDanceRevolution	Wii	2010.0	Simulation	Konami Digital Entertainment	0.29	0.04	0.00	0.02
5948	5950	FIFA Street 3	X360	2008.0	Sports	Electronic Arts	0.12	0.14	0.00	0.03
537	538	Jak II	PS2	2003.0	Platform	Sony Computer Entertainment	1.68	0.74	0.00	0.36
14422	14425	Kekkaishi: Kokubourou Shuurai	DS	2008.0	Action	Namco Bandai Games	0.00	0.00	0.03	0.00
14895	14898	Soul Eater: Battle Resonance	PSP	2009.0	Action	Namco Bandai Games	0.00	0.00	0.03	0.00

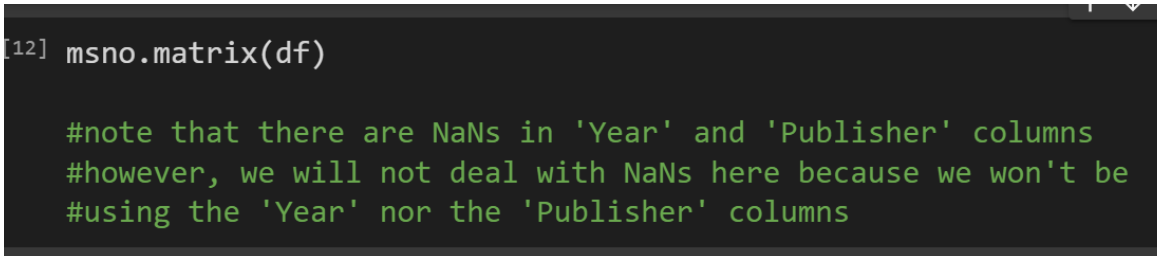
STEP 2: USING MISSINGNO CHART TO PREVIEW NANS IN COLUMNS



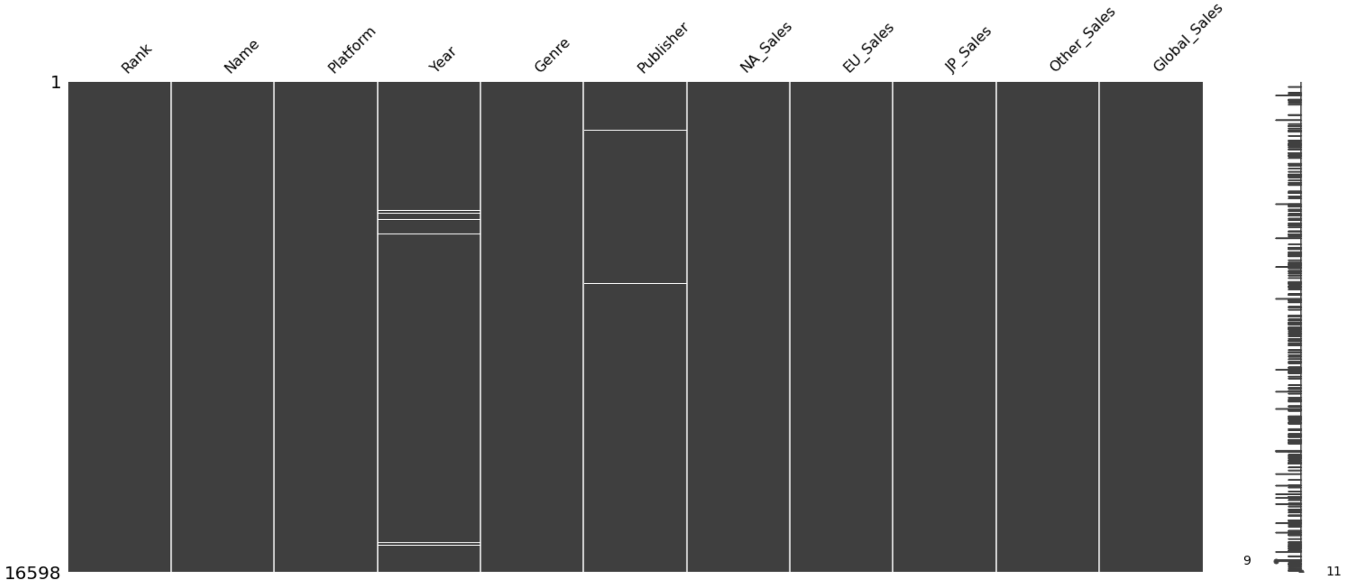
Step 2: Using MissingNo Chart to Preview NaNs in Columns

```
import missingno as msno

msno.bar(df)

#note that there are NaNs in 'Year' and 'Publisher' columns
```

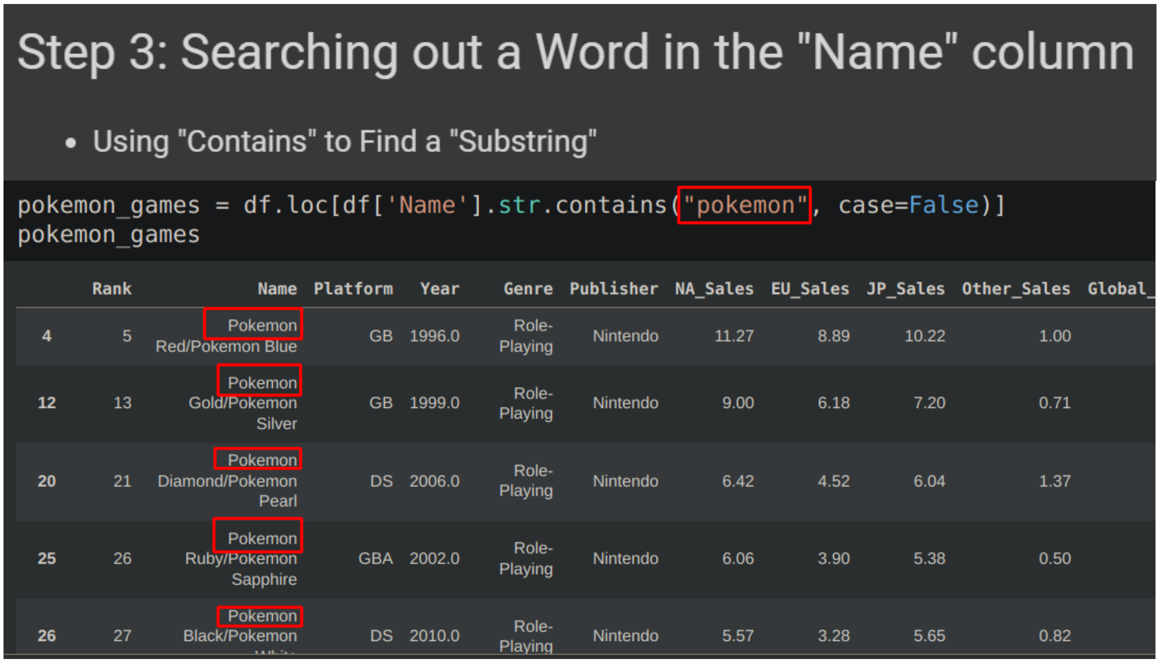




```
msno.matrix(df)

#note that there are NaNs in 'Year' and 'Publisher' columns
#however, we will not deal with NaNs here because we won't be
#using the 'Year' nor the 'Publisher' columns
```



STEP 3: SEARCHING OUT A WORD IN THE 'NAME' COLUMN



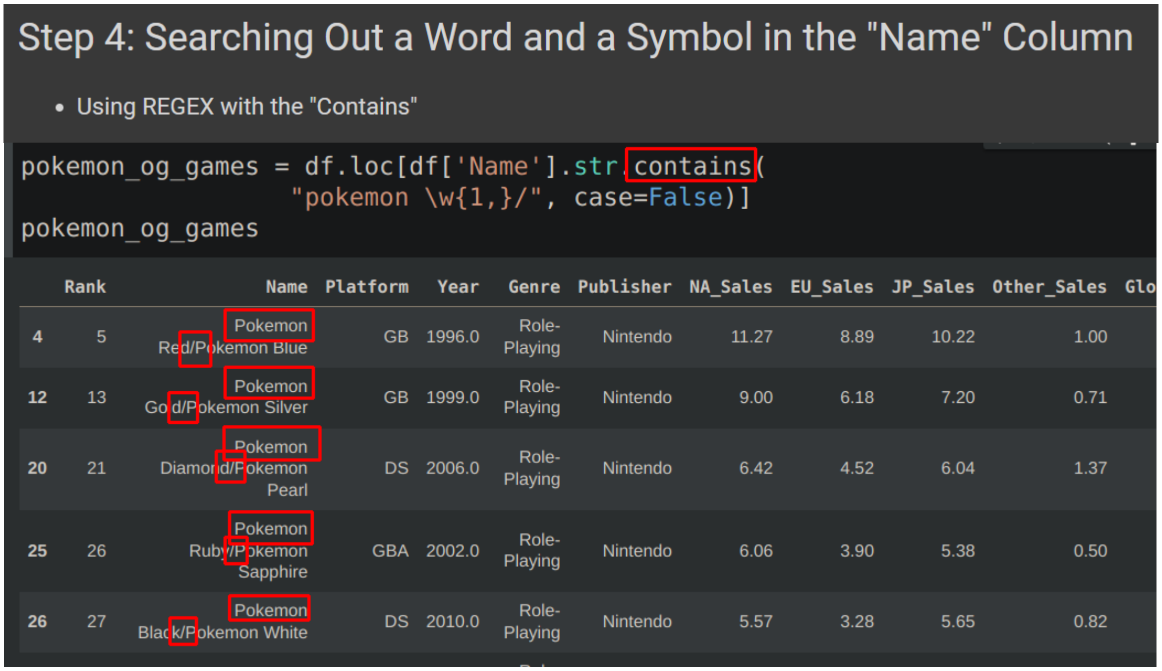
Step 3: Searching out a Word in the "Name" column

- Using "Contains" to Find a "Substring"

```
pokemon_games = df.loc[df['Name'].str.contains("pokemon", case=False)]
pokemon_games
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	
12	13	Pokemon Gold/Pokemon Silver	GB	1999.0	Role-Playing	Nintendo	9.00	6.18	7.20	0.71	
20	21	Pokemon Diamond/Pokemon Pearl	DS	2006.0	Role-Playing	Nintendo	6.42	4.52	6.04	1.37	
25	26	Pokemon Ruby/Pokemon Sapphire	GBA	2002.0	Role-Playing	Nintendo	6.06	3.90	5.38	0.50	
26	27	Pokemon Black/Pokemon	DS	2010.0	Role-Playing	Nintendo	5.57	3.28	5.65	0.82	

STEP 4: SEARCHING OUT A WORD AND A SYMBOL IN THE "NAME" COLUMN



Step 4: Searching Out a Word and a Symbol in the "Name" Column

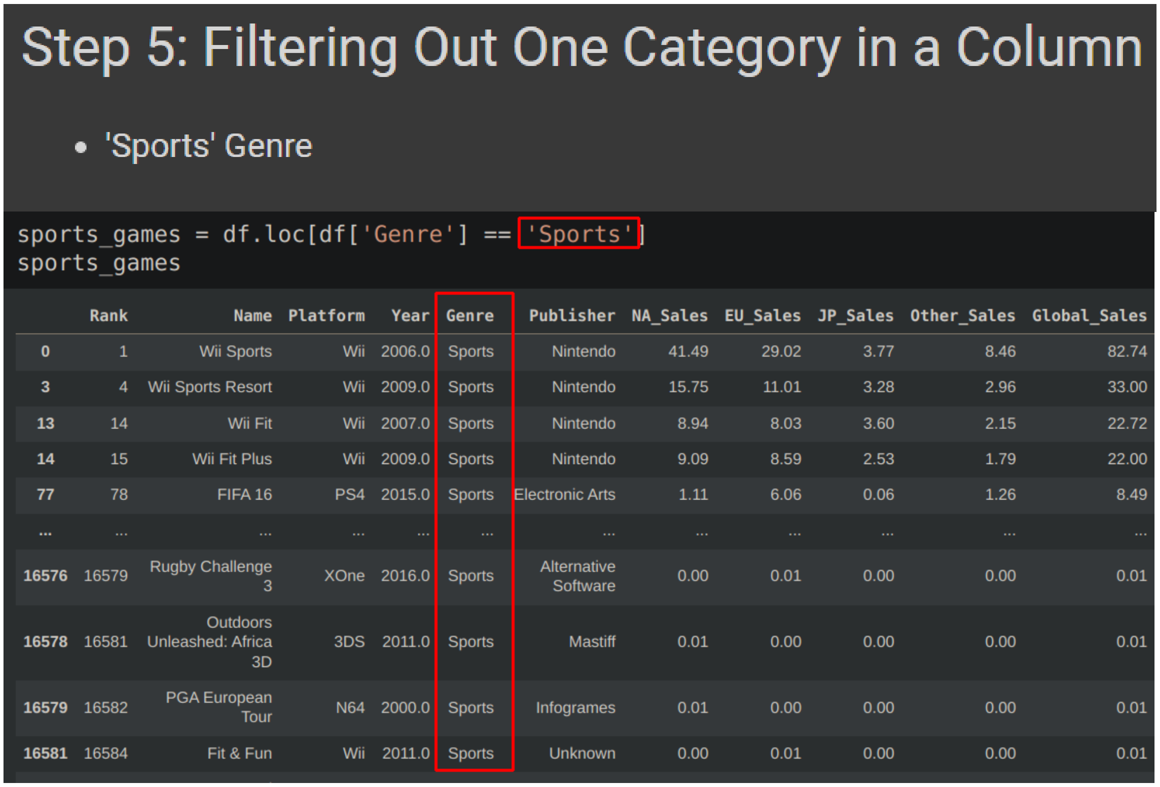
- Using REGEX with the "Contains"

```
pokemon_og_games = df.loc[df['Name'].str.contains(
                        "pokemon \w{1,}/", case=False)]
pokemon_og_games
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Glo
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	
12	13	Pokemon Gold/Pokemon Silver	GB	1999.0	Role-Playing	Nintendo	9.00	6.18	7.20	0.71	
20	21	Pokemon Diamond/Pokemon Pearl	DS	2006.0	Role-Playing	Nintendo	6.42	4.52	6.04	1.37	
25	26	Pokemon Ruby/Pokemon Sapphire	GBA	2002.0	Role-Playing	Nintendo	6.06	3.90	5.38	0.50	
26	27	Pokemon Black/Pokemon White	DS	2010.0	Role-Playing	Nintendo	5.57	3.28	5.65	0.82	

- Used some simple regex to find strings that matched the pattern of "pokemon" + "one character or more" + "/".

- The result of the new mask returned rows including "Pokemon Red/Pokemon Blue", "Pokemon Gold/Pokemon Silver", and more



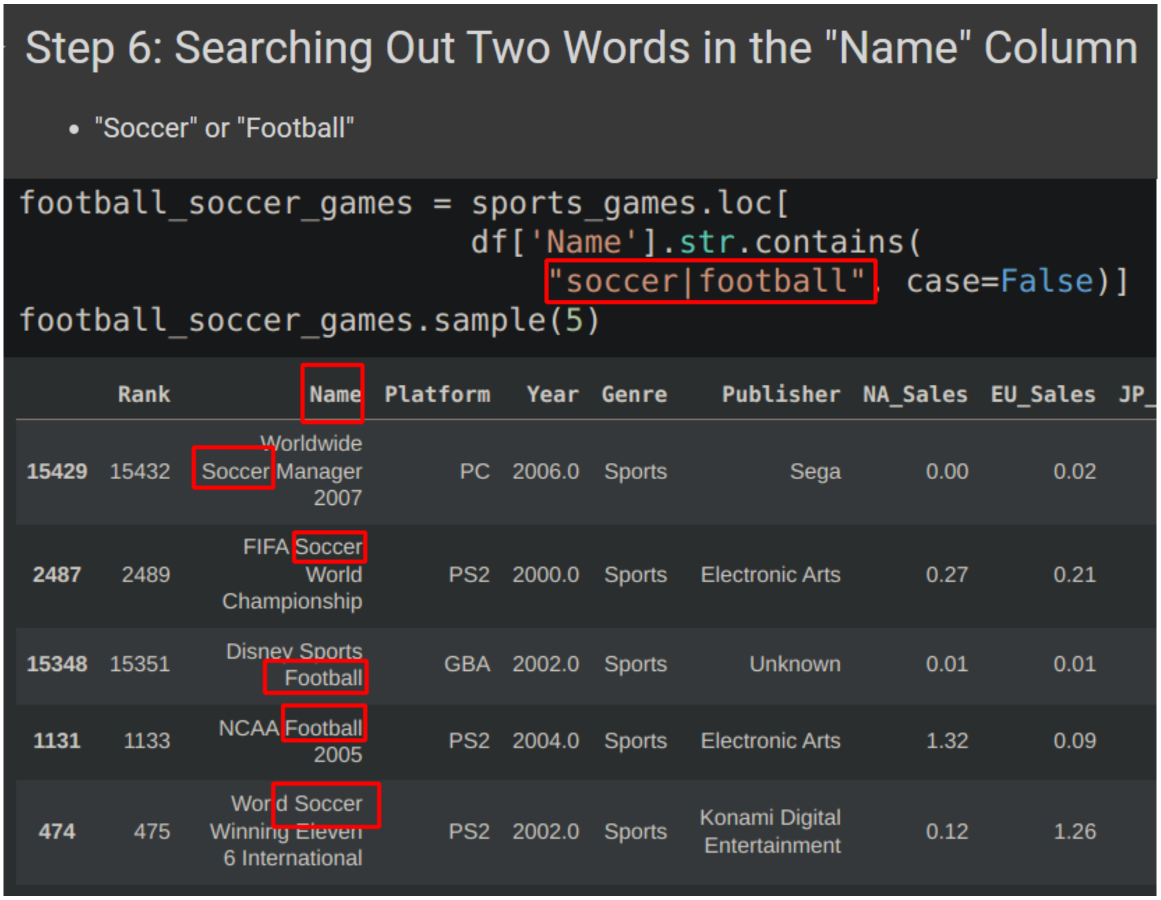
Step 5: Filtering Out One Category in a Column

- 'Sports' Genre

```
sports_games = df.loc[df['Genre'] == 'Sports']
sports_games
```

	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
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
13	14	Wii Fit	Wii	2007.0	Sports	Nintendo	8.94	8.03	3.60	2.15	22.72
14	15	Wii Fit Plus	Wii	2009.0	Sports	Nintendo	9.09	8.59	2.53	1.79	22.00
77	78	FIFA 16	PS4	2015.0	Sports	Electronic Arts	1.11	6.06	0.06	1.26	8.49
...	...	...	...	...	...	...	...	...	...	...	...
16576	16579	Rugby Challenge 3	XOne	2016.0	Sports	Alternative Software	0.00	0.01	0.00	0.00	0.01
16578	16581	Outdoors Unleashed: Africa 3D	3DS	2011.0	Sports	Mastiff	0.01	0.00	0.00	0.00	0.01
16579	16582	PGA European Tour	N64	2000.0	Sports	Infogrames	0.01	0.00	0.00	0.00	0.01
16581	16584	Fit & Fun	Wii	2011.0	Sports	Unknown	0.00	0.01	0.00	0.00	0.01

STEP 6: SEARCHING OUT TWO WORDS IN THE “NAME” COLUMN: SOCCER” OR “FOOTBALL”

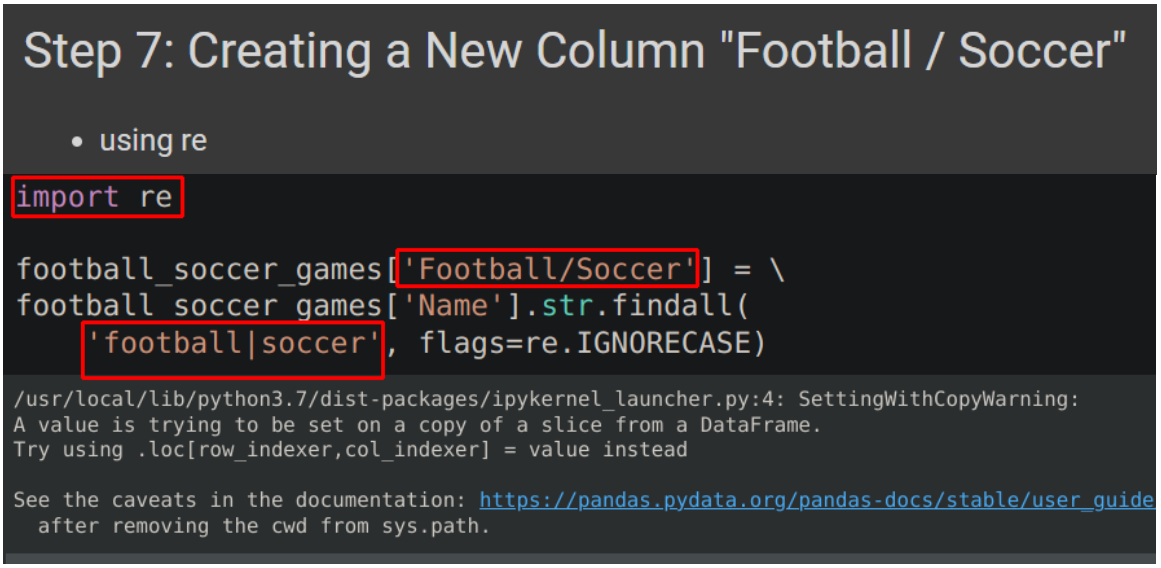


Step 6: Searching Out Two Words in the "Name" Column

- "Soccer" or "Football"

```
football_soccer_games = sports_games.loc[
                        df['Name'].str.contains(
                            "soccer|football", case=False)]
football_soccer_games.sample(5)
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_
15429	15432	Worldwide Soccer Manager 2007	PC	2006.0	Sports	Sega	0.00	0.02	
2487	2489	FIFA Soccer World Championship	PS2	2000.0	Sports	Electronic Arts	0.27	0.21	
15348	15351	Disney Sports Football	GBA	2002.0	Sports	Unknown	0.01	0.01	
1131	1133	NCAA Football 2005	PS2	2004.0	Sports	Electronic Arts	1.32	0.09	
474	475	World Soccer Winning Eleven 6 International	PS2	2002.0	Sports	Konami Digital Entertainment	0.12	1.26	

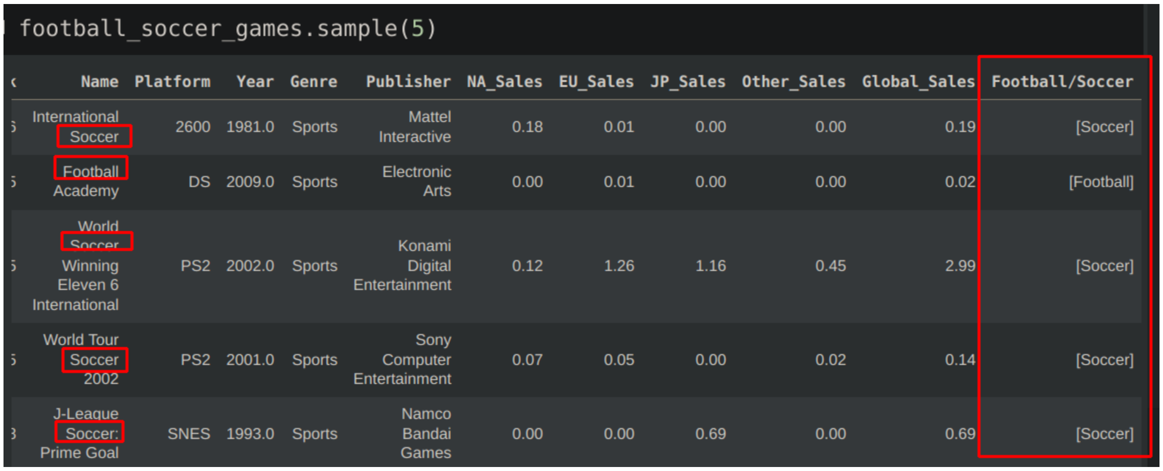


Step 7: Creating a New Column "Football / Soccer"

- using re

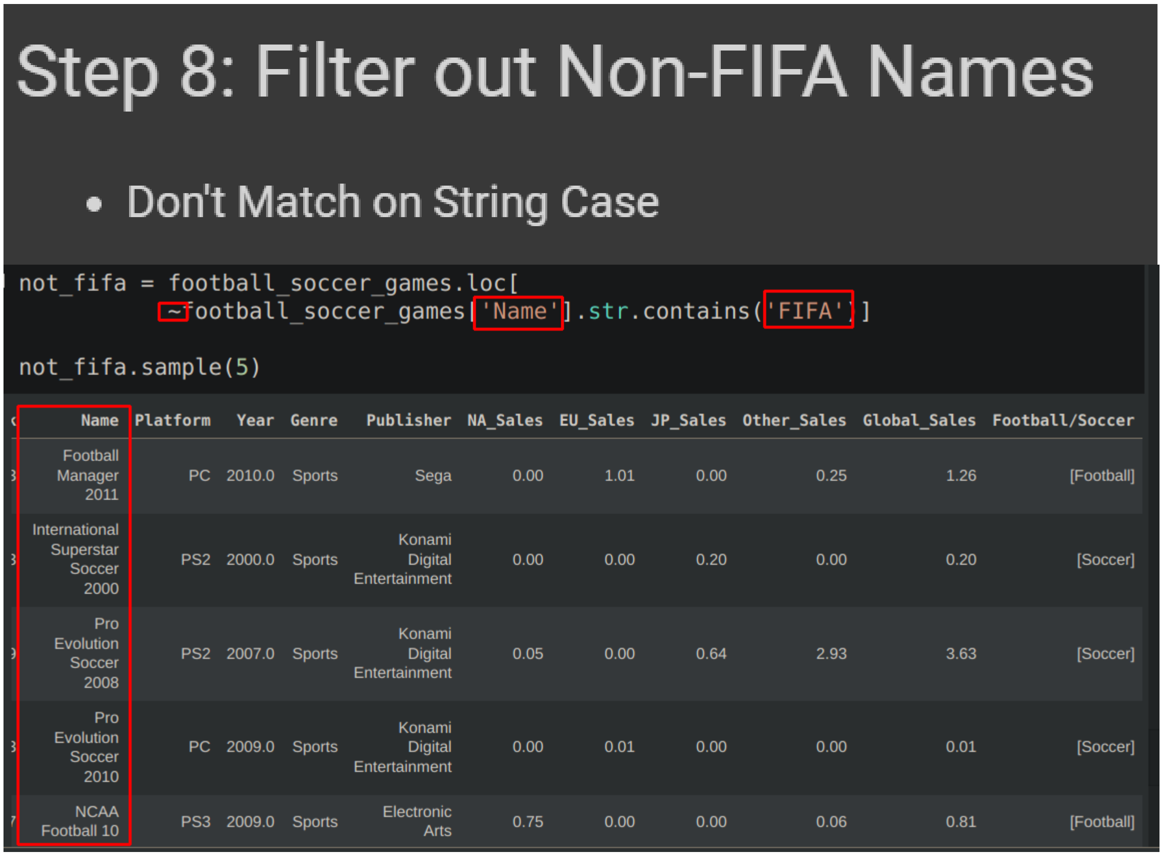
```
import re

football_soccer_games['Football/Soccer'] = \
football_soccer_games['Name'].str.findall(
    'football|soccer', flags=re.IGNORECASE)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
  after removing the cwd from sys.path.
```



```
football_soccer_games.sample(5)
```

	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Football/Soccer
5	International Soccer	2600	1981.0	Sports	Mattel Interactive	0.18	0.01	0.00	0.00	0.19	[Soccer]
5	Football Academy	DS	2009.0	Sports	Electronic Arts	0.00	0.01	0.00	0.00	0.02	[Football]
5	World Soccer Winning Eleven 6 International	PS2	2002.0	Sports	Konami Digital Entertainment	0.12	1.26	1.16	0.45	2.99	[Soccer]
5	World Tour Soccer 2002	PS2	2001.0	Sports	Sony Computer Entertainment	0.07	0.05	0.00	0.02	0.14	[Soccer]
3	J-League Soccer: Prime Goal	SNES	1993.0	Sports	Namco Bandai Games	0.00	0.00	0.69	0.00	0.69	[Soccer]



Step 8: Filter out Non-FIFA Names

- Don't Match on String Case

```
not_fifa = football_soccer_games.loc[
          ~football_soccer_games['Name'].str.contains('FIFA')]

not_fifa.sample(5)
```

Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Football/Soccer
Football Manager 2011	PC	2010.0	Sports	Sega	0.00	1.01	0.00	0.25	1.26	[Football]
International Superstar Soccer 2000	PS2	2000.0	Sports	Konami Digital Entertainment	0.00	0.00	0.20	0.00	0.20	[Soccer]
Pro Evolution Soccer 2008	PS2	2007.0	Sports	Konami Digital Entertainment	0.05	0.00	0.64	2.93	3.63	[Soccer]
Pro Evolution Soccer 2010	PC	2009.0	Sports	Konami Digital Entertainment	0.00	0.01	0.00	0.00	0.01	[Soccer]
NCAA Football 10	PS3	2009.0	Sports	Electronic Arts	0.75	0.00	0.00	0.06	0.81	[Football]

ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.