In [1]:
```
#Import pandas so we can import our video game file
#Jupyter Lab 3.44, Python 3
#If you do not have the packahes available to your machine, please follow the usual pr
import pandas as pd
df = pd.read_csv(r'C:\Users\e202271009\Documents\D214\vgsales.csv')
#Your path will be different
#Import Other Packages, these packages allow us to perform statistical analysis and pl
import numpy as np
import scipy as sp
import scipy.stats as stats
import pylab
from statsmodels.formula.api import ols
import statistics
import matplotlib as mpl
import matplotlib.pyplot as mpl
import seaborn as sb
# Scikit
import sklearn
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

In [2]:
```
#Let's view the dataset
#View Data Types
print(df.select_dtypes(include="float").info())
print(df.select_dtypes(include="integer").info())
print(df.select_dtypes(include="object").info())

#View exaample of the information in the dataset
print(df.head(5))
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Year          16327 non-null  float64
 1   NA_Sales      16598 non-null  float64
 2   EU_Sales      16598 non-null  float64
 3   JP_Sales      16598 non-null  float64
 4   Other_Sales   16598 non-null  float64
 5   Global_Sales  16598 non-null  float64
dtypes: float64(6)
memory usage: 778.2 KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Rank    16598 non-null  int64
dtypes: int64(1)
memory usage: 129.8 KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Name       16598 non-null  object
 1   Platform   16598 non-null  object
 2   Genre      16598 non-null  object
 3   Publisher  16540 non-null  object
dtypes: object(4)
memory usage: 518.8+ KB
None
   Rank                      Name Platform    Year         Genre Publisher  \
0     1                Wii Sports      Wii  2006.0        Sports  Nintendo   
1     2         Super Mario Bros.      NES  1985.0      Platform  Nintendo   
2     3            Mario Kart Wii      Wii  2008.0        Racing  Nintendo   
3     4         Wii Sports Resort      Wii  2009.0        Sports  Nintendo   
4     5  Pokemon Red/Pokemon Blue       GB  1996.0  Role-Playing  Nintendo   

   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales  
0     41.49     29.02      3.77         8.46         82.74  
1     29.08      3.58      6.81         0.77         40.24  
2     15.85     12.88      3.79         3.31         35.82  
3     15.75     11.01      3.28         2.96         33.00  
4     11.27      8.89     10.22         1.00         31.37  
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Rank          16598 non-null  int64  
 1   Name          16598 non-null  object 
 2   Platform      16598 non-null  object 
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object 
 5   Publisher     16540 non-null  object 
 6   NA_Sales      16598 non-null  float64
```

```
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

In [3]:
```
#Check for any missing values
df.isna().sum()
```

Out[3]:
```
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 [4]:
```
# Remove nulls from where it says “True” above this cell
df = df.dropna(subset=['Year', 'Publisher'])
```

In [5]:
```
#Check for any missing values
df.isna().sum()
```

Out[5]:
```
Rank            0
Name            0
Platform        0
Year            0
Genre           0
Publisher       0
NA_Sales        0
EU_Sales        0
JP_Sales        0
Other_Sales     0
Global_Sales    0
dtype: int64
```

In [6]:
```
#Drop irrelevant columns from the dataset
df = df.drop(['Name'], axis=1)
```

In [7]:
```
for col in df:
    print(df[col].unique())
```

```
[    1     2     3 ... 16598 16599 16600]
['Wii' 'NES' 'GB' 'DS' 'X360' 'PS3' 'PS2' 'SNES' 'GBA' '3DS' 'PS4' 'N64'
 'PS' 'XB' 'PC' '2600' 'PSP' 'XOne' 'GC' 'WiiU' 'GEN' 'DC' 'PSV' 'SAT'
 'SCD' 'WS' 'NG' 'TG16' '3DO' 'GG' 'PCFX']
[2006. 1985. 2008. 2009. 1996. 1989. 1984. 2005. 1999. 2007. 2010. 2013.
 2004. 1990. 1988. 2002. 2001. 2011. 1998. 2015. 2012. 2014. 1992. 1997.
 1993. 1994. 1982. 2003. 1986. 2000. 1995. 2016. 1991. 1981. 1987. 1980.
 1983. 2020. 2017.]
['Sports' 'Platform' 'Racing' 'Role-Playing' 'Puzzle' 'Misc' 'Shooter'
 'Simulation' 'Action' 'Fighting' 'Adventure' 'Strategy']
['Nintendo' 'Microsoft Game Studios' 'Take-Two Interactive'
 'Sony Computer Entertainment' 'Activision' 'Ubisoft' 'Bethesda Softworks'
 'Electronic Arts' 'Sega' 'SquareSoft' 'Atari' '505 Games' 'Capcom'
 'GT Interactive' 'Konami Digital Entertainment'
 'Sony Computer Entertainment Europe' 'Square Enix' 'LucasArts'
 'Virgin Interactive' 'Warner Bros. Interactive Entertainment'
 'Universal Interactive' 'Eidos Interactive' 'RedOctane' 'Vivendi Games'
 'Enix Corporation' 'Namco Bandai Games' 'Palcom' 'Hasbro Interactive'
 'THQ' 'Fox Interactive' 'Acclaim Entertainment' 'MTV Games'
 'Disney Interactive Studios' 'Majesco Entertainment' 'Codemasters'
 'Red Orb' 'Level 5' 'Arena Entertainment' 'Midway Games' 'JVC'
 'Deep Silver' '989 Studios' 'NCSoft' 'UEP Systems' 'Parker Bros.' 'Maxis'
 'Imagic' 'Tecmo Koei' 'Valve Software' 'ASCII Entertainment' 'Mindscape'
 'Infogrames' 'Unknown' 'Square' 'Valve' 'Activision Value' 'Banpresto'
 'D3Publisher' 'Oxygen Interactive' 'Red Storm Entertainment'
 'Video System' 'Hello Games' 'Global Star' 'Gotham Games'
 'Westwood Studios' 'GungHo' 'Crave Entertainment' 'Hudson Soft' 'Coleco'
 'Rising Star Games' 'Atlus' 'TDK Mediactive' 'ASC Games' 'Zoo Games'
 'Accolade' 'Sony Online Entertainment' '3DO' 'RTL' 'Natsume'
 'Focus Home Interactive' 'Alchemist' 'Black Label Games'
 'SouthPeak Games' 'Mastertronic' 'Ocean' 'Zoo Digital Publishing'
 'Psygnosis' 'City Interactive' 'Empire Interactive' 'Success' 'Compile'
 'Russel' 'Taito' 'Agetec' 'GSP' 'Microprose' 'Play It'
 'Slightly Mad Studios' 'Tomy Corporation' 'Sammy Corporation'
 'Koch Media' 'Game Factory' 'Titus' 'Marvelous Entertainment' 'Genki'
 'Mojang' 'Pinnacle' 'CTO SpA' 'TalonSoft' 'Crystal Dynamics' 'SCi'
 'Quelle' 'mixi, Inc' 'Rage Software' 'Ubisoft Annecy' 'Scholastic Inc.'
 'Interplay' 'Mystique' 'ChunSoft' 'Square EA'
 '20th Century Fox Video Games' 'Avanquest Software'
 'Hudson Entertainment' 'Nordic Games' 'Men-A-Vision' 'Nobilis'
 'Big Ben Interactive' 'Touchstone' 'Spike' 'Jester Interactive'
 'Nippon Ichi Software' 'LEGO Media' 'Quest' 'Illusion Softworks'
 'Tigervision' 'Funbox Media' 'Rocket Company' 'Metro 3D'
 'Mattel Interactive' 'IE Institute' 'Rondomedia'
 'Sony Computer Entertainment America' 'Universal Gamex' 'Ghostlight'
 'Wizard Video Games' 'BMG Interactive Entertainment' 'PQube'
 'Trion Worlds' 'Laguna' 'Ignition Entertainment' 'Takara'
 'Kadokawa Shoten' 'Destineer' 'Enterbrain' 'Xseed Games' 'Imagineer'
 'System 3 Arcade Software' 'CPG Products' 'Aruze Corp' 'Gamebridge'
 'Midas Interactive Entertainment' 'Jaleco' 'Answer Software' 'XS Games'
 'Activision Blizzard' 'Pack In Soft' 'Rebellion' 'Xplosiv'
 'GameMill Entertainment' 'Wanadoo' 'NovaLogic' 'Telltale Games' 'Epoch'
 'BAM! Entertainment' 'Knowledge Adventure' 'Mastiff' 'Tetris Online'
 'Harmonix Music Systems' 'ESP' 'TYO' 'Telegames' 'Mud Duck Productions'
 'Screenlife' 'Pioneer LDC' 'Magical Company' 'Mentor Interactive' 'Kemco'
 'Human Entertainment' 'Avanquest' 'Data Age' 'Electronic Arts Victor'
 'Black Bean Games' 'Jack of All Games' '989 Sports' 'Takara Tomy'
 'Media Rings' 'Elf' 'Starfish' 'Zushi Games' 'Jorudan'
 'Destination Software, Inc' 'New' 'Brash Entertainment'
 'ITT Family Games' 'PopCap Games' 'Home Entertainment Suppliers'
```

```
'Ackkstudios' 'Starpath Corp.' 'P2 Games' 'BPS' 'Gathering of Developers'
'NewKidCo' 'Storm City Games' 'CokeM Interactive' 'CBS Electronics'
'Magix' 'Marvelous Interactive' 'Kalypso Media'
'Nihon Falcom Corporation' 'Wargaming.net' 'Angel Studios'
'Arc System Works' 'Playmates' 'SNK Playmore' 'Hamster Corporation'
'From Software' 'Nippon Columbia' 'Nichibutsu' 'Little Orbit'
'Conspiracy Entertainment' 'DTP Entertainment' 'Hect' 'Mumbo Jumbo'
'Pacific Century Cyber Works' 'Indie Games' 'Liquid Games' 'NEC' 'Axela'
'ArtDink' 'Sunsoft' 'Gust' 'SNK' 'NEC Interchannel' 'FuRyu'
'Xing Entertainment' 'ValuSoft' 'Victor Interactive' 'Detn8 Games'
'American Softworks' 'Nordcurrent' 'Bomb' 'Falcom Corporation'
'AQ Interactive' 'CCP' 'Milestone S.r.l.' 'JoWood Productions'
'Seta Corporation' 'On Demand' 'NCS' 'Aspyr' 'Gremlin Interactive Ltd'
'Agatsuma Entertainment' 'Compile Heart' 'Culture Brain' 'Mad Catz'
'Shogakukan' 'Merscom LLC' 'Rebellion Developments' 'Nippon Telenet'
'TDK Core' 'bitComposer Games' 'Foreign Media Games' 'Astragon' 'SSI'
'Kadokawa Games' 'Idea Factory' 'Performance Designed Products'
'Asylum Entertainment' 'Core Design Ltd.' 'PlayV' 'UFO Interactive'
'Idea Factory International' 'Playlogic Game Factory' 'Essential Games'
'Adeline Software' 'Funcom' 'Panther Software' 'Blast! Entertainment Ltd'
'Game Life' 'DSI Games' 'Avalon Interactive' 'Popcorn Arcade'
'Neko Entertainment' 'Vir2L Studios' 'Aques' 'Syscom'
'White Park Bay Software' 'System 3' 'Vatical Entertainment' 'Daedalic'
'EA Games' 'Media Factory' 'Vic Tokai' 'The Adventure Company'
'Game Arts' 'Broccoli' 'Acquire' 'General Entertainment'
'Excalibur Publishing' 'Imadio' 'Swing! Entertainment'
'Sony Music Entertainment' 'Aqua Plus' 'Paradox Interactive'
'Hip Interactive' 'DreamCatcher Interactive' 'Tripwire Interactive'
'Sting' 'Yacht Club Games' 'SCS Software' 'Bigben Interactive'
'Havas Interactive' 'Slitherine Software' 'Graffiti' 'Funsta' 'Telstar'
'U.S. Gold' 'DreamWorks Interactive' 'Data Design Interactive' 'MTO'
'DHM Interactive' 'FunSoft' 'SPS' 'Bohemia Interactive'
'Reef Entertainment' 'Tru Blu Entertainment' 'Moss' 'T&E Soft' 'O-Games'
'Aksys Games' 'NDA Productions' 'Data East' 'Time Warner Interactive'
'Gainax Network Systems' 'Daito' 'O3 Entertainment' 'Gameloft'
'Xicat Interactive' 'Simon & Schuster Interactive' 'Valcon Games'
'PopTop Software' 'TOHO' 'HMH Interactive' '5pb' 'Cave'
'CDV Software Entertainment' 'Microids' 'PM Studios' 'Paon' 'Micro Cabin'
'GameTek' 'Benesse' 'Type-Moon' 'Enjoy Gaming ltd.' 'Asmik Corp'
'Interplay Productions' 'Asmik Ace Entertainment' 'inXile Entertainment'
'Image Epoch' 'Phantom EFX' 'Evolved Games' 'responDESIGN'
'Culture Publishers' 'Griffin International' 'Hackberry' 'Hearty Robin'
'Nippon Amuse' 'Origin Systems' 'Seventh Chord' 'Mitsui' 'Milestone'
'Abylight' 'Flight-Plan' 'Glams' 'Locus' 'Warp' 'Daedalic Entertainment'
'Alternative Software' 'Myelin Media' 'Mercury Games'
'Irem Software Engineering' 'Sunrise Interactive' 'Elite'
'Evolution Games' 'Tivola' 'Global A Entertainment' 'Edia' 'Athena'
'Aria' 'Gamecock' 'Tommo' 'Altron' 'Happinet' 'iWin' 'Media Works'
'Fortyfive' 'Revolution Software' 'Imax' 'Crimson Cow' '10TACLE Studios'
'Groove Games' 'Pack-In-Video' 'Insomniac Games'
'Ascaron Entertainment GmbH' 'Asgard' 'Ecole' 'Yumedia' 'Phenomedia'
'HAL Laboratory' 'Grand Prix Games' 'DigiCube' 'Creative Core'
'Kaga Create' 'WayForward Technologies' 'LSP Games' 'ASCII Media Works'
'Coconuts Japan' 'Arika' 'Ertain' 'Marvel Entertainment' 'Prototype'
'Phantagram' '1C Company' 'The Learning Company' 'TechnoSoft' 'Vap'
'Misawa' 'Tradewest' 'Team17 Software' 'Yeti' 'Pow' 'Navarre Corp'
'MediaQuest' 'Max Five' 'Comfort' 'Monte Christo Multimedia'
'Pony Canyon' 'Riverhillsoft' 'Summitsoft' 'Milestone S.r.l' 'Playmore'
'MLB.com' 'Kool Kizz' 'Flashpoint Games' '49Games' 'Legacy Interactive'
'Alawar Entertainment' 'CyberFront' 'Cloud Imperium Games Corporation'
```

```
 'Societa' 'Virtual Play Games' 'Interchannel' 'Sonnet' 'Experience Inc.'
 'Zenrin' 'Iceberg Interactive' 'Ivolgamus' '2D Boy' 'MC2 Entertainment'
 'Kando Games' 'Just Flight' 'Office Create' 'Mamba Games' 'Fields'
 'Princess Soft' 'Maximum Family Games' 'Berkeley' 'Fuji'
 'Dusenberry Martin Racing' 'imageepoch Inc.' 'Big Fish Games'
 'Her Interactive' 'Kamui' 'ASK' 'TopWare Interactive' 'Headup Games'
 'KSS' 'Cygames' 'KID' 'Quinrose' 'Sunflowers' 'dramatic create' 'TGL'
 'Encore' 'Extreme Entertainment Group' 'Intergrow' 'G.Rev' 'Sweets'
 'Kokopeli Digital Studios' 'Number None' 'Nexon' 'id Software'
 'BushiRoad' 'Tryfirst' 'Strategy First' '7G//AMES' 'GN Software' "Yuke's"
 'Easy Interactive' 'Licensed 4U' 'FuRyu Corporation'
 'Lexicon Entertainment' 'Paon Corporation' 'Kids Station' 'GOA'
 'Graphsim Entertainment' 'King Records' 'Introversion Software'
 'Minato Station' 'Devolver Digital' 'Blue Byte' 'Gaga'
 'Yamasa Entertainment' 'Plenty' 'Views' 'fonfun' 'NetRevo'
 'Codemasters Online' 'Quintet' 'Phoenix Games' 'Dorart' 'Marvelous Games'
 'Focus Multimedia' 'Imageworks' 'Karin Entertainment' 'Aerosoft'
 'Technos Japan Corporation' 'Gakken' 'Mirai Shounen' 'Datam Polystar'
 'Saurus' 'HuneX' 'Revolution (Japan)' 'Giza10' 'Visco' 'Alvion' 'Mycom'
 'Giga' 'Warashi' 'System Soft' 'Sold Out' 'Lighthouse Interactive'
 'Masque Publishing' 'RED Entertainment' 'Michaelsoft'
 'Media Entertainment' 'New World Computing' 'Genterprise'
 'Interworks Unlimited, Inc.' 'Boost On' 'Stainless Games'
 'EON Digital Entertainment' 'Epic Games' 'Naxat Soft'
 'Ascaron Entertainment' 'Piacci' 'Nitroplus' 'Paradox Development'
 'Otomate' 'Ongakukan' 'Commseed' 'Inti Creates' 'Takuyo'
 'Interchannel-Holon' 'Rain Games' 'UIG Entertainment']
[4.149e+01 2.908e+01 1.585e+01 1.575e+01 1.127e+01 2.320e+01 1.138e+01
 1.403e+01 1.459e+01 2.693e+01 9.070e+00 9.810e+00 9.000e+00 8.940e+00
 9.090e+00 1.497e+01 7.010e+00 9.430e+00 1.278e+01 4.750e+00 6.420e+00
 1.083e+01 9.540e+00 9.630e+00 8.410e+00 6.060e+00 5.570e+00 3.440e+00
 6.850e+00 9.030e+00 5.890e+00 9.670e+00 5.170e+00 5.770e+00 4.990e+00
 8.250e+00 8.520e+00 5.540e+00 6.990e+00 6.750e+00 5.980e+00 2.550e+00
 4.740e+00 7.970e+00 3.800e+00 4.400e+00 6.910e+00 3.010e+00 6.160e+00
 4.230e+00 6.760e+00 4.020e+00 4.890e+00 2.960e+00 4.760e+00 5.990e+00
 4.340e+00 5.080e+00 6.050e+00 6.720e+00 7.030e+00 5.550e+00 3.660e+00
 6.630e+00 4.090e+00 5.840e+00 3.880e+00 5.910e+00 4.360e+00 5.580e+00
 2.010e+00 4.460e+00 5.030e+00 3.540e+00 1.110e+00 1.790e+00 6.820e+00
 3.810e+00 2.910e+00 1.060e+00 9.800e-01 5.800e+00 2.580e+00 2.280e+00
 2.820e+00 7.280e+00 2.900e+00 2.930e+00 2.800e+00 4.100e+00 3.780e+00
 5.390e+00 3.240e+00 4.790e+00 3.830e+00 4.520e+00 3.510e+00 2.850e+00
 3.270e+00 3.680e+00 4.410e+00 3.130e+00 2.470e+00 4.120e+00 4.140e+00
 7.800e-01 2.710e+00 2.770e+00 3.230e+00 3.500e+00 4.150e+00 3.100e+00
 8.400e-01 1.670e+00 2.790e+00 7.900e-01 3.250e+00 3.740e+00 2.640e+00
 4.980e+00 2.570e+00 3.640e+00 3.700e+00 4.010e+00 7.000e-02 3.110e+00
 3.920e+00 4.050e+00 2.450e+00 4.470e+00 2.630e+00 3.180e+00 2.410e+00
 1.880e+00 6.600e-01 2.260e+00 2.490e+00 2.970e+00 2.540e+00 2.950e+00
 3.280e+00 2.700e+00 2.990e+00 4.700e-01 3.140e+00 2.620e+00 3.210e+00
 2.720e+00 2.070e+00 1.970e+00 1.740e+00 2.180e+00 3.020e+00 1.620e+00
 1.920e+00 3.330e+00 1.220e+00 2.300e+00 6.500e-01 2.430e+00 2.320e+00
 1.080e+00 1.900e+00 2.100e+00 9.600e-01 1.640e+00 1.980e+00 3.590e+00
 3.220e+00 1.960e+00 2.660e+00 1.700e+00 6.000e-01 3.400e+00 2.050e+00
 3.420e+00 2.590e+00 3.360e+00 3.060e+00 3.490e+00 3.390e+00 1.850e+00
 2.310e+00 3.980e+00 2.890e+00 0.000e+00 2.740e+00 2.560e+00 1.910e+00
 5.700e-01 2.800e-01 2.360e+00 1.730e+00 3.050e+00 1.870e+00 1.940e+00
 2.080e+00 2.290e+00 2.420e+00 2.600e+00 1.890e+00 1.780e+00 1.550e+00
 3.190e+00 4.180e+00 4.210e+00 3.630e+00 2.000e-01 1.540e+00 2.670e+00
 1.000e-01 2.190e+00 2.030e+00 3.030e+00 2.200e+00 9.200e-01 2.750e+00
 4.000e+00 2.510e+00 2.110e+00 2.230e+00 1.410e+00 3.000e+00 1.460e+00
 8.800e-01 1.300e+00 1.280e+00 2.250e+00 2.020e+00 3.380e+00 2.040e+00
```

```
 3.790e+00 1.400e+00 4.030e+00 1.650e+00 7.100e-01 2.140e+00 1.420e+00
 2.130e+00 2.650e+00 2.350e+00 1.200e-01 1.680e+00 1.120e+00 2.780e+00
 1.380e+00 2.150e+00 1.180e+00 1.330e+00 6.700e-01 1.530e+00 1.150e+00
 9.300e-01 2.120e+00 2.480e+00 1.600e-01 8.700e-01 2.210e+00 1.440e+00
 1.490e+00 1.140e+00 2.400e+00 1.820e+00 1.370e+00 1.930e+00 5.800e-01
 1.590e+00 2.530e+00 2.330e+00 5.000e-02 1.610e+00 2.380e+00 1.570e+00
 1.560e+00 1.230e+00 1.660e+00 1.170e+00 2.840e+00 5.900e-01 2.090e+00
 2.390e+00 1.340e+00 1.130e+00 8.600e-01 1.750e+00 4.600e-01 1.430e+00
 1.630e+00 1.450e+00 1.470e+00 1.990e+00 1.500e+00 8.000e-01 1.360e+00
 5.000e-01 2.500e-01 9.500e-01 1.270e+00 3.000e-02 1.720e+00 7.300e-01
 1.760e+00 1.350e+00 1.480e+00 1.520e+00 2.060e+00 6.800e-01 9.100e-01
 1.690e+00 8.000e-02 1.290e+00 2.170e+00 2.500e+00 1.010e+00 1.580e+00
 1.040e+00 2.220e+00 1.830e+00 6.100e-01 1.840e+00 9.900e-01 1.510e+00
 9.000e-02 4.000e-01 2.520e+00 1.320e+00 2.000e-02 1.050e+00 2.900e-01
 1.190e+00 8.900e-01 3.000e-01 1.200e+00 1.240e+00 1.250e+00 1.070e+00
 1.020e+00 6.900e-01 1.950e+00 2.000e+00 7.600e-01 6.300e-01 9.000e-01
 1.860e+00 4.800e-01 6.400e-01 3.700e-01 1.310e+00 1.500e-01 1.210e+00
 4.900e-01 1.300e-01 1.810e+00 1.260e+00 8.100e-01 7.700e-01 1.000e+00
 1.160e+00 1.390e+00 8.500e-01 5.200e-01 5.100e-01 3.800e-01 6.200e-01
 1.090e+00 1.710e+00 1.030e+00 3.400e-01 1.600e+00 5.400e-01 1.400e-01
 1.000e-02 8.200e-01 8.300e-01 1.100e-01 9.400e-01 1.770e+00 7.000e-01
 9.700e-01 7.500e-01 3.500e-01 7.200e-01 7.400e-01 1.800e-01 1.100e+00
 5.600e-01 2.600e-01 2.100e-01 2.200e-01 5.300e-01 5.500e-01 2.300e-01
 3.900e-01 3.200e-01 4.500e-01 4.100e-01 3.100e-01 2.400e-01 6.000e-02
 4.300e-01 4.400e-01 1.900e-01 4.000e-02 1.700e-01 3.600e-01 3.300e-01
 2.700e-01 4.200e-01]
[2.902e+01 3.580e+00 1.288e+01 1.101e+01 8.890e+00 2.260e+00 9.230e+00
 9.200e+00 7.060e+00 6.300e-01 1.100e+01 7.570e+00 6.180e+00 8.030e+00
 8.590e+00 4.940e+00 9.270e+00 4.000e-01 3.750e+00 9.260e+00 4.520e+00
 2.710e+00 3.440e+00 5.310e+00 5.490e+00 3.900e+00 3.280e+00 5.360e+00
 5.090e+00 4.280e+00 5.040e+00 3.730e+00 4.050e+00 5.810e+00 5.880e+00
 4.300e+00 3.630e+00 5.820e+00 4.510e+00 2.610e+00 4.440e+00 3.520e+00
 3.910e+00 2.830e+00 2.770e+00 2.850e+00 1.000e-02 3.400e+00 3.370e+00
 2.040e+00 3.100e+00 3.870e+00 2.990e+00 4.880e+00 3.690e+00 3.760e+00
 2.150e+00 2.650e+00 3.110e+00 3.150e+00 2.630e+00 1.980e+00 1.940e+00
 3.070e+00 2.360e+00 2.470e+00 2.890e+00 3.420e+00 2.380e+00 1.710e+00
 2.320e+00 1.880e+00 2.860e+00 1.240e+00 6.060e+00 3.530e+00 1.530e+00
 2.300e+00 1.860e+00 5.050e+00 6.420e+00 2.010e+00 2.070e+00 1.720e+00
 1.780e+00 4.500e-01 2.420e+00 3.290e+00 3.300e+00 1.890e+00 2.170e+00
 1.180e+00 1.350e+00 1.900e+00 2.130e+00 2.190e+00 2.090e+00 3.030e+00
 2.930e+00 2.220e+00 1.750e+00 1.040e+00 1.770e+00 2.210e+00 4.320e+00
 3.020e+00 2.750e+00 2.800e+00 2.350e+00 2.640e+00 1.920e+00 2.250e+00
 2.780e+00 4.290e+00 1.840e+00 9.300e-01 2.560e+00 1.300e+00 1.580e+00
 1.200e+00 1.560e+00 1.970e+00 1.260e+00 8.300e-01 6.210e+00 1.620e+00
 1.740e+00 1.830e+00 2.280e+00 0.000e+00 2.050e+00 6.900e-01 1.630e+00
 1.470e+00 1.950e+00 6.000e-01 1.650e+00 1.910e+00 5.700e-01 1.640e+00
 1.110e+00 1.870e+00 2.290e+00 2.510e+00 9.600e-01 1.120e+00 7.700e-01
 1.080e+00 7.900e-01 2.480e+00 2.460e+00 7.500e-01 1.250e+00 9.800e-01
 3.480e+00 7.400e-01 2.020e+00 2.230e+00 6.100e-01 1.690e+00 1.430e+00
 1.160e+00 1.380e+00 1.060e+00 1.360e+00 1.030e+00 1.730e+00 2.600e-01
 1.540e+00 9.900e-01 1.680e+00 2.000e+00 3.140e+00 1.570e+00 1.310e+00
 2.100e+00 1.410e+00 9.100e-01 1.990e+00 1.390e+00 1.150e+00 9.200e-01
 2.400e-01 1.510e+00 1.400e-01 1.290e+00 2.390e+00 1.050e+00 5.000e-01
 1.790e+00 5.800e-01 1.270e+00 1.340e+00 1.020e+00 2.270e+00 2.550e+00
 2.790e+00 4.400e-01 4.800e-01 2.700e-01 2.100e-01 5.100e-01 1.500e+00
 1.520e+00 4.000e-02 1.280e+00 1.550e+00 2.120e+00 2.800e-01 8.900e-01
 1.960e+00 7.600e-01 1.170e+00 4.900e-01 1.610e+00 1.140e+00 6.500e-01
 1.440e+00 1.930e+00 1.370e+00 1.400e+00 8.800e-01 7.300e-01 1.220e+00
 1.130e+00 8.100e-01 9.700e-01 1.800e+00 1.090e+00 6.800e-01 1.420e+00
 6.700e-01 2.400e+00 1.330e+00 1.210e+00 1.660e+00 9.400e-01 3.900e-01
```

```
 3.800e-01 1.010e+00 1.070e+00 1.590e+00 1.900e-01 2.110e+00 3.000e-01
 1.000e-01 7.200e-01 8.700e-01 5.600e-01 1.600e-01 8.600e-01 6.200e-01
 1.000e+00 8.200e-01 8.400e-01 6.400e-01 8.000e-01 4.700e-01 3.200e-01
 7.100e-01 1.850e+00 2.430e+00 5.500e-01 5.300e-01 9.000e-02 1.190e+00
 7.000e-02 2.000e-01 7.000e-01 2.200e-01 2.900e-01 1.700e-01 5.400e-01
 3.600e-01 3.100e-01 5.200e-01 1.100e-01 1.200e-01 1.100e+00 4.600e-01
 8.000e-02 5.900e-01 2.300e-01 7.800e-01 2.000e-02 8.500e-01 2.500e-01
 1.800e-01 1.490e+00 1.500e-01 1.300e-01 9.000e-01 4.100e-01 3.000e-02
 3.700e-01 5.000e-02 3.300e-01 4.200e-01 6.000e-02 9.500e-01 1.230e+00
 6.600e-01 3.400e-01 4.300e-01 3.500e-01]
[3.770e+00 6.810e+00 3.790e+00 3.280e+00 1.022e+01 4.220e+00 6.500e+00
 2.930e+00 4.700e+00 2.800e-01 1.930e+00 4.130e+00 7.200e+00 3.600e+00
 2.530e+00 2.400e-01 9.700e-01 4.100e-01 3.540e+00 4.160e+00 6.040e+00
 4.180e+00 3.840e+00 6.000e-02 4.700e-01 5.380e+00 5.650e+00 5.320e+00
 1.870e+00 1.300e-01 3.120e+00 1.100e-01 4.340e+00 3.500e-01 6.500e-01
 7.000e-02 8.000e-02 4.900e-01 3.000e-01 2.660e+00 4.800e-01 5.330e+00
 2.670e+00 3.600e-01 3.960e+00 1.910e+00 1.100e+00 1.200e+00 3.080e+00
 2.690e+00 1.400e-01 2.540e+00 2.130e+00 8.100e-01 3.800e-01 4.400e-01
 2.120e+00 3.150e+00 1.250e+00 0.000e+00 4.000e-02 2.230e+00 2.470e+00
 1.000e-02 1.690e+00 3.000e+00 2.000e-02 4.360e+00 1.980e+00 1.000e-01
 3.810e+00 2.490e+00 5.000e-02 1.580e+00 3.140e+00 6.600e-01 2.730e+00
 3.630e+00 9.800e-01 2.200e-01 1.450e+00 1.310e+00 7.000e-01 2.420e+00
 6.000e-01 1.400e+00 1.420e+00 1.390e+00 1.270e+00 8.700e-01 1.700e-01
 1.900e-01 9.400e-01 2.100e-01 1.600e+00 1.030e+00 2.500e-01 1.600e-01
 2.060e+00 1.490e+00 1.290e+00 9.000e-02 2.870e+00 3.000e-02 8.300e-01
 7.800e-01 2.330e+00 4.350e+00 2.020e+00 1.360e+00 1.810e+00 1.970e+00
 9.100e-01 9.900e-01 9.500e-01 2.000e+00 1.010e+00 2.780e+00 2.110e+00
 1.090e+00 2.000e-01 3.610e+00 1.570e+00 2.200e+00 1.890e+00 1.700e+00
 1.080e+00 1.500e-01 1.110e+00 8.000e-01 2.900e-01 1.540e+00 1.200e-01
 8.900e-01 4.870e+00 1.520e+00 1.320e+00 1.150e+00 1.440e+00 4.100e+00
 4.600e-01 1.050e+00 1.610e+00 2.600e-01 1.380e+00 7.200e-01 6.200e-01
 1.800e-01 5.700e-01 5.800e-01 3.100e-01 1.760e+00 3.700e-01 2.100e+00
 9.000e-01 5.100e-01 6.400e-01 2.460e+00 9.200e-01 1.070e+00 2.620e+00
 1.120e+00 5.400e-01 7.300e-01 2.700e-01 5.900e-01 3.670e+00 5.500e-01
 4.000e-01 1.750e+00 3.440e+00 3.300e-01 2.550e+00 7.400e-01 8.200e-01
 2.320e+00 7.600e-01 7.700e-01 3.180e+00 2.350e+00 3.190e+00 9.300e-01
 8.800e-01 3.030e+00 4.500e-01 1.160e+00 1.190e+00 3.400e-01 1.130e+00
 6.800e-01 1.960e+00 7.100e-01 1.040e+00 2.680e+00 2.650e+00 9.600e-01
 2.410e+00 5.200e-01 2.430e+00 1.340e+00 1.480e+00 2.340e+00 1.060e+00
 1.210e+00 2.280e+00 1.630e+00 2.050e+00 2.170e+00 1.560e+00 1.350e+00
 6.300e-01 7.500e-01 7.900e-01 5.300e-01 1.530e+00 1.300e+00 1.460e+00
 1.330e+00 3.900e-01 6.900e-01 4.200e-01 5.600e-01 8.400e-01 3.200e-01
 1.710e+00 1.650e+00 6.100e-01 1.510e+00 1.500e+00 1.240e+00 1.180e+00
 1.370e+00 1.000e+00 1.260e+00 4.300e-01 8.500e-01 1.280e+00 6.700e-01
 1.140e+00 8.600e-01 1.170e+00 5.000e-01 1.020e+00 2.300e-01]
[8.460e+00 7.700e-01 3.310e+00 2.960e+00 1.000e+00 5.800e-01 2.900e+00
 2.850e+00 2.260e+00 4.700e-01 2.750e+00 1.920e+00 7.100e-01 2.150e+00
 1.790e+00 1.670e+00 4.140e+00 1.057e+01 5.500e-01 2.050e+00 1.370e+00
 4.200e-01 4.600e-01 1.380e+00 1.780e+00 5.000e-01 8.200e-01 1.180e+00
 1.160e+00 1.320e+00 5.900e-01 1.130e+00 7.900e-01 2.310e+00 2.520e+00
 1.120e+00 1.290e+00 1.620e+00 1.300e+00 1.020e+00 1.830e+00 8.800e-01
 8.900e-01 1.210e+00 2.020e+00 2.300e-01 7.530e+00 7.600e-01 6.500e-01
 2.900e-01 1.030e+00 5.200e-01 7.800e-01 2.120e+00 1.630e+00 3.500e-01
 9.800e-01 1.070e+00 1.500e-01 6.300e-01 7.300e-01 9.600e-01 9.000e-01
 4.100e-01 7.000e-01 8.500e-01 1.800e-01 1.260e+00 6.800e-01 8.000e-02
 4.300e-01 2.010e+00 9.100e-01 3.300e-01 1.750e+00 6.400e-01 1.230e+00
 1.600e-01 3.100e-01 1.900e-01 6.900e-01 6.700e-01 1.100e+00 2.800e-01
 2.200e-01 4.900e-01 5.600e-01 1.730e+00 1.090e+00 9.900e-01 1.060e+00
 3.000e-01 1.040e+00 1.420e+00 6.200e-01 1.470e+00 1.050e+00 1.400e-01
 1.140e+00 7.000e-02 2.100e-01 6.100e-01 5.700e-01 1.700e-01 1.200e-01
```

```
 0.000e+00 5.100e-01 6.000e-01 7.200e-01 8.300e-01 2.400e-01 1.010e+00
 1.740e+00 5.300e-01 4.500e-01 1.100e-01 8.700e-01 4.000e-02 8.000e-01
 3.400e-01 2.000e-01 9.000e-02 8.400e-01 9.400e-01 4.400e-01 6.000e-02
 3.800e-01 1.480e+00 3.700e-01 6.600e-01 1.000e-01 2.500e-01 1.300e-01
 4.800e-01 5.000e-02 2.000e-02 7.400e-01 3.900e-01 8.600e-01 4.000e-01
 7.500e-01 1.930e+00 5.400e-01 2.700e-01 3.600e-01 3.200e-01 2.930e+00
 2.600e-01 2.460e+00 1.000e-02 3.000e-02 1.510e+00 1.680e+00 1.410e+00
 1.820e+00 1.330e+00 9.300e-01]
[8.274e+01 4.024e+01 3.582e+01 3.300e+01 3.137e+01 3.026e+01 3.001e+01
 2.902e+01 2.862e+01 2.831e+01 2.476e+01 2.342e+01 2.310e+01 2.272e+01
 2.200e+01 2.182e+01 2.140e+01 2.081e+01 2.061e+01 2.022e+01 1.836e+01
 1.814e+01 1.728e+01 1.638e+01 1.615e+01 1.585e+01 1.532e+01 1.530e+01
 1.498e+01 1.476e+01 1.464e+01 1.435e+01 1.424e+01 1.403e+01 1.373e+01
 1.351e+01 1.346e+01 1.310e+01 1.304e+01 1.273e+01 1.227e+01 1.221e+01
 1.214e+01 1.198e+01 1.190e+01 1.189e+01 1.166e+01 1.152e+01 1.133e+01
 1.118e+01 1.102e+01 1.095e+01 1.079e+01 1.077e+01 1.069e+01 1.057e+01
 1.055e+01 1.049e+01 1.042e+01 1.026e+01 1.021e+01 9.880e+00 9.870e+00
 9.820e+00 9.760e+00 9.720e+00 9.590e+00 9.520e+00 9.490e+00 9.320e+00
 9.300e+00 9.200e+00 9.090e+00 9.020e+00 8.840e+00 8.760e+00 8.490e+00
 8.420e+00 8.330e+00 8.240e+00 8.110e+00 8.090e+00 8.060e+00 8.050e+00
 7.860e+00 7.840e+00 7.810e+00 7.720e+00 7.690e+00 7.670e+00 7.600e+00
 7.580e+00 7.460e+00 7.450e+00 7.370e+00 7.340e+00 7.310e+00 7.300e+00
 7.270e+00 7.230e+00 7.200e+00 7.160e+00 7.130e+00 7.070e+00 6.960e+00
 6.950e+00 6.910e+00 6.900e+00 6.830e+00 6.820e+00 6.810e+00 6.760e+00
 6.730e+00 6.720e+00 6.690e+00 6.670e+00 6.600e+00 6.590e+00 6.560e+00
 6.510e+00 6.500e+00 6.430e+00 6.410e+00 6.400e+00 6.390e+00 6.360e+00
 6.340e+00 6.310e+00 6.300e+00 6.280e+00 6.270e+00 6.240e+00 6.110e+00
 6.050e+00 6.030e+00 5.990e+00 5.950e+00 5.920e+00 5.840e+00 5.830e+00
 5.820e+00 5.740e+00 5.720e+00 5.650e+00 5.580e+00 5.570e+00 5.550e+00
 5.530e+00 5.510e+00 5.500e+00 5.490e+00 5.470e+00 5.460e+00 5.450e+00
 5.430e+00 5.420e+00 5.360e+00 5.340e+00 5.310e+00 5.300e+00 5.290e+00
 5.270e+00 5.260e+00 5.240e+00 5.210e+00 5.200e+00 5.190e+00 5.180e+00
 5.170e+00 5.150e+00 5.140e+00 5.130e+00 5.120e+00 5.110e+00 5.080e+00
 5.070e+00 5.050e+00 5.020e+00 5.010e+00 5.000e+00 4.980e+00 4.960e+00
 4.940e+00 4.910e+00 4.900e+00 4.880e+00 4.870e+00 4.850e+00 4.840e+00
 4.830e+00 4.820e+00 4.790e+00 4.770e+00 4.760e+00 4.730e+00 4.700e+00
 4.680e+00 4.670e+00 4.640e+00 4.630e+00 4.620e+00 4.610e+00 4.600e+00
 4.580e+00 4.570e+00 4.550e+00 4.530e+00 4.500e+00 4.490e+00 4.480e+00
 4.470e+00 4.450e+00 4.440e+00 4.420e+00 4.410e+00 4.390e+00 4.380e+00
 4.370e+00 4.350e+00 4.340e+00 4.330e+00 4.310e+00 4.260e+00 4.250e+00
 4.240e+00 4.230e+00 4.220e+00 4.210e+00 4.200e+00 4.190e+00 4.170e+00
 4.160e+00 4.140e+00 4.120e+00 4.110e+00 4.100e+00 4.090e+00 4.080e+00
 4.070e+00 4.060e+00 4.050e+00 4.030e+00 4.020e+00 4.010e+00 4.000e+00
 3.990e+00 3.980e+00 3.920e+00 3.910e+00 3.900e+00 3.890e+00 3.880e+00
 3.870e+00 3.850e+00 3.840e+00 3.830e+00 3.820e+00 3.810e+00 3.790e+00
 3.780e+00 3.770e+00 3.760e+00 3.730e+00 3.720e+00 3.710e+00 3.700e+00
 3.690e+00 3.670e+00 3.660e+00 3.650e+00 3.640e+00 3.630e+00 3.620e+00
 3.610e+00 3.600e+00 3.590e+00 3.580e+00 3.560e+00 3.540e+00 3.530e+00
 3.520e+00 3.510e+00 3.500e+00 3.490e+00 3.480e+00 3.460e+00 3.450e+00
 3.440e+00 3.430e+00 3.420e+00 3.410e+00 3.400e+00 3.390e+00 3.380e+00
 3.360e+00 3.340e+00 3.330e+00 3.320e+00 3.310e+00 3.280e+00 3.270e+00
 3.260e+00 3.240e+00 3.230e+00 3.220e+00 3.210e+00 3.200e+00 3.190e+00
 3.180e+00 3.170e+00 3.160e+00 3.150e+00 3.140e+00 3.130e+00 3.120e+00
 3.110e+00 3.090e+00 3.080e+00 3.070e+00 3.050e+00 3.040e+00 3.030e+00
 3.020e+00 3.000e+00 2.990e+00 2.980e+00 2.970e+00 2.960e+00 2.950e+00
 2.940e+00 2.930e+00 2.920e+00 2.910e+00 2.900e+00 2.890e+00 2.880e+00
 2.870e+00 2.860e+00 2.850e+00 2.840e+00 2.830e+00 2.820e+00 2.810e+00
 2.800e+00 2.790e+00 2.780e+00 2.770e+00 2.760e+00 2.750e+00 2.740e+00
 2.730e+00 2.720e+00 2.710e+00 2.700e+00 2.690e+00 2.680e+00 2.670e+00
 2.660e+00 2.650e+00 2.640e+00 2.630e+00 2.620e+00 2.610e+00 2.600e+00
```

```
2.590e+00 2.580e+00 2.570e+00 2.560e+00 2.540e+00 2.520e+00 2.510e+00
2.500e+00 2.490e+00 2.480e+00 2.470e+00 2.460e+00 2.450e+00 2.440e+00
2.430e+00 2.420e+00 2.410e+00 2.400e+00 2.390e+00 2.380e+00 2.370e+00
2.360e+00 2.350e+00 2.340e+00 2.330e+00 2.320e+00 2.310e+00 2.300e+00
2.290e+00 2.280e+00 2.270e+00 2.260e+00 2.250e+00 2.240e+00 2.230e+00
2.220e+00 2.210e+00 2.200e+00 2.190e+00 2.180e+00 2.170e+00 2.160e+00
2.150e+00 2.140e+00 2.130e+00 2.120e+00 2.110e+00 2.100e+00 2.090e+00
2.080e+00 2.070e+00 2.060e+00 2.050e+00 2.040e+00 2.030e+00 2.020e+00
2.010e+00 2.000e+00 1.990e+00 1.980e+00 1.970e+00 1.960e+00 1.950e+00
1.940e+00 1.930e+00 1.920e+00 1.910e+00 1.900e+00 1.890e+00 1.880e+00
1.870e+00 1.860e+00 1.850e+00 1.840e+00 1.830e+00 1.820e+00 1.810e+00
1.800e+00 1.790e+00 1.780e+00 1.770e+00 1.760e+00 1.750e+00 1.740e+00
1.730e+00 1.720e+00 1.710e+00 1.700e+00 1.690e+00 1.680e+00 1.670e+00
1.660e+00 1.650e+00 1.640e+00 1.630e+00 1.620e+00 1.610e+00 1.600e+00
1.590e+00 1.580e+00 1.570e+00 1.560e+00 1.550e+00 1.540e+00 1.530e+00
1.520e+00 1.510e+00 1.500e+00 1.490e+00 1.480e+00 1.470e+00 1.460e+00
1.450e+00 1.440e+00 1.430e+00 1.420e+00 1.410e+00 1.400e+00 1.390e+00
1.380e+00 1.370e+00 1.360e+00 1.350e+00 1.340e+00 1.330e+00 1.320e+00
1.310e+00 1.300e+00 1.290e+00 1.280e+00 1.270e+00 1.260e+00 1.250e+00
1.240e+00 1.230e+00 1.220e+00 1.210e+00 1.200e+00 1.190e+00 1.180e+00
1.170e+00 1.160e+00 1.150e+00 1.140e+00 1.130e+00 1.120e+00 1.110e+00
1.100e+00 1.090e+00 1.080e+00 1.070e+00 1.060e+00 1.050e+00 1.040e+00
1.030e+00 1.020e+00 1.010e+00 1.000e+00 9.900e-01 9.800e-01 9.700e-01
9.600e-01 9.500e-01 9.400e-01 9.300e-01 9.200e-01 9.100e-01 9.000e-01
8.900e-01 8.800e-01 8.700e-01 8.600e-01 8.500e-01 8.400e-01 8.300e-01
8.200e-01 8.100e-01 8.000e-01 7.900e-01 7.800e-01 7.700e-01 7.600e-01
7.500e-01 7.400e-01 7.300e-01 7.200e-01 7.100e-01 7.000e-01 6.900e-01
6.800e-01 6.700e-01 6.600e-01 6.500e-01 6.400e-01 6.300e-01 6.200e-01
6.100e-01 6.000e-01 5.900e-01 5.800e-01 5.700e-01 5.600e-01 5.500e-01
5.400e-01 5.300e-01 5.200e-01 5.100e-01 5.000e-01 4.900e-01 4.800e-01
4.700e-01 4.600e-01 4.500e-01 4.400e-01 4.300e-01 4.200e-01 4.100e-01
4.000e-01 3.900e-01 3.800e-01 3.700e-01 3.600e-01 3.500e-01 3.400e-01
3.300e-01 3.200e-01 3.100e-01 3.000e-01 2.900e-01 2.800e-01 2.700e-01
2.600e-01 2.500e-01 2.400e-01 2.300e-01 2.200e-01 2.100e-01 2.000e-01
1.900e-01 1.800e-01 1.700e-01 1.600e-01 1.500e-01 1.400e-01 1.300e-01
1.200e-01 1.100e-01 1.000e-01 9.000e-02 8.000e-02 7.000e-02 6.000e-02
5.000e-02 4.000e-02 3.000e-02 2.000e-02 1.000e-02]
```

In [8]:

```
#Now we need to convert our categorical data into numeric data using the label encoder
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
label = le.fit_transform(df['Platform'])
print(label)
df.drop("Platform", axis=1, inplace=True)
df["Platform"] = label
df
```

```
[26 11 26 ... 16  4  6]
```

Out[8]:

	Rank	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	F
0	1	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	
1	2	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	
2	3	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82	
3	4	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00	
4	5	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	
...	...	...	...	...	...	...	...	...	...	
16593	16596	2002.0	Platform	Kemco	0.01	0.00	0.00	0.00	0.01	
16594	16597	2003.0	Shooter	Infogrames	0.01	0.00	0.00	0.00	0.01	
16595	16598	2008.0	Racing	Activision	0.00	0.00	0.00	0.00	0.01	
16596	16599	2010.0	Puzzle	7G//AMES	0.00	0.01	0.00	0.00	0.01	
16597	16600	2003.0	Platform	Wanadoo	0.01	0.00	0.00	0.00	0.01	

16291 rows × 10 columns

In [9]:
```
#Next variable
from sklearn.preprocessing import LabelEncoder
le2 = LabelEncoder()
label2 = le2.fit_transform(df['Genre'])
print(label2)
df.drop("Genre", axis=1, inplace=True)
df["Genre"] = label2
df
```

```
[10  4  6 ...  6  5  4]
```

Out[9]:

	Rank	Year	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Platform
0	1	2006.0	Nintendo	41.49	29.02	3.77	8.46	82.74	26
1	2	1985.0	Nintendo	29.08	3.58	6.81	0.77	40.24	11
2	3	2008.0	Nintendo	15.85	12.88	3.79	3.31	35.82	26
3	4	2009.0	Nintendo	15.75	11.01	3.28	2.96	33.00	26
4	5	1996.0	Nintendo	11.27	8.89	10.22	1.00	31.37	5
...	...	...	...	...	...	...	...	...	...
16593	16596	2002.0	Kemco	0.01	0.00	0.00	0.00	0.01	6
16594	16597	2003.0	Infogrames	0.01	0.00	0.00	0.00	0.01	7
16595	16598	2008.0	Activision	0.00	0.00	0.00	0.00	0.01	16
16596	16599	2010.0	7G//AMES	0.00	0.01	0.00	0.00	0.01	4
16597	16600	2003.0	Wanadoo	0.01	0.00	0.00	0.00	0.01	6

16291 rows × 10 columns

In [10]:
```
#Now we need to convert our categorical data into numeric data using the label encoder
from sklearn.preprocessing import LabelEncoder
le3 = LabelEncoder()
label3 = le3.fit_transform(df['Genre'])
print(label3)
df.drop("Publisher", axis=1, inplace=True)
df["Publisher"] = label3
df
```

```
[10  4  6 ...  6  5  4]
```

Out[10]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Platform	Genre	Publ
0	1	2006.0	41.49	29.02	3.77	8.46	82.74	26	10	
1	2	1985.0	29.08	3.58	6.81	0.77	40.24	11	4	
2	3	2008.0	15.85	12.88	3.79	3.31	35.82	26	6	
3	4	2009.0	15.75	11.01	3.28	2.96	33.00	26	10	
4	5	1996.0	11.27	8.89	10.22	1.00	31.37	5	7	
...	...	...	...	...	...	...	...	...	...	...
16593	16596	2002.0	0.01	0.00	0.00	0.00	0.01	6	4	
16594	16597	2003.0	0.01	0.00	0.00	0.00	0.01	7	8	
16595	16598	2008.0	0.00	0.00	0.00	0.00	0.01	16	6	
16596	16599	2010.0	0.00	0.01	0.00	0.00	0.01	4	5	
16597	16600	2003.0	0.01	0.00	0.00	0.00	0.01	6	4	

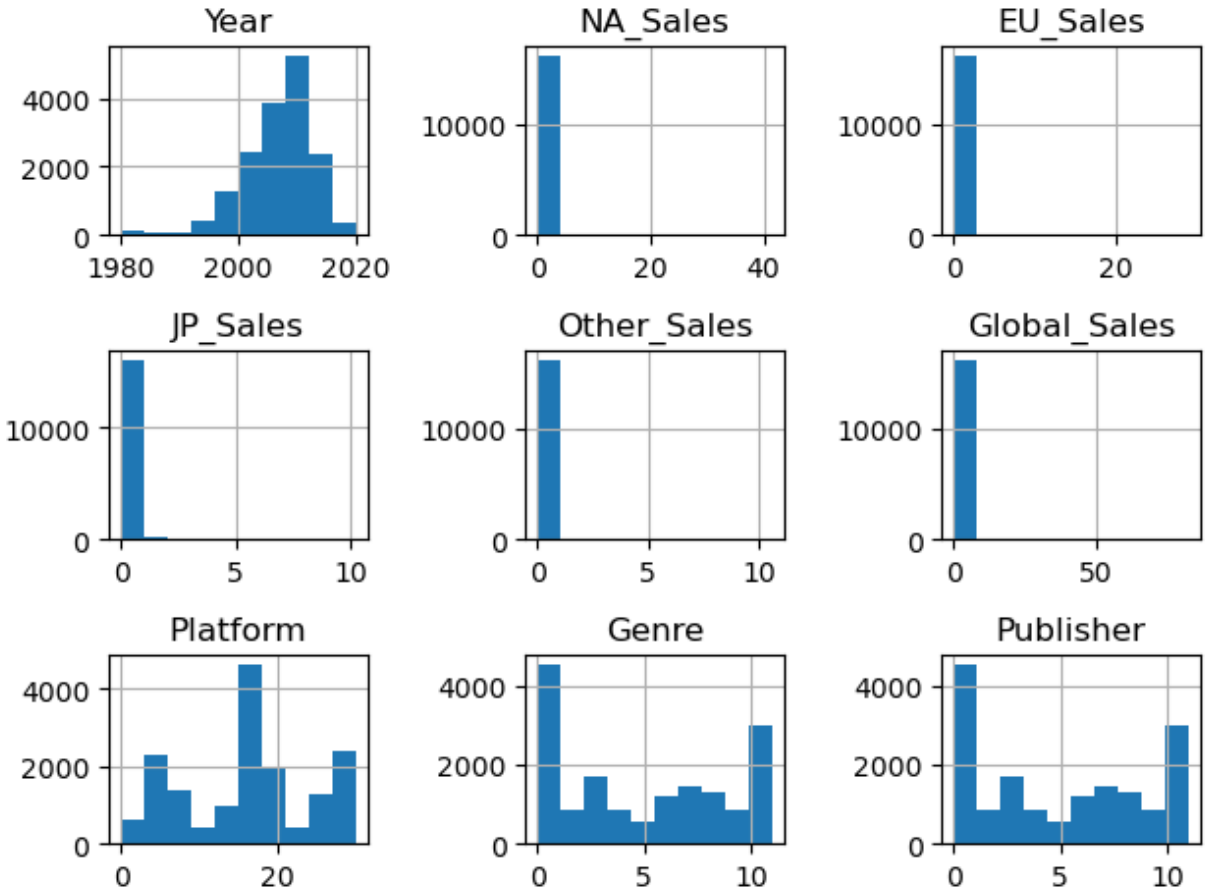
16291 rows × 10 columns

In [11]:
```
#Mean Values in the Distribution
print (df.mean())
#Median Values in the Distribution)
print (df.median())
```

```
Rank            8290.190228
Year            2006.405561
NA_Sales           0.265647
EU_Sales           0.147731
JP_Sales           0.078833
Other_Sales        0.048426
Global_Sales       0.540910
Platform          15.812841
Genre              4.928611
Publisher          4.928611
dtype: float64
Rank            8292.00
Year            2007.00
NA_Sales           0.08
EU_Sales           0.02
JP_Sales           0.00
Other_Sales        0.01
Global_Sales       0.17
Platform          16.00
Genre              5.00
Publisher          5.00
dtype: float64
```

In [12]:
```
#Create Histograms to view our variables
df[['Year' ,
          'NA_Sales' ,
          'EU_Sales' ,
          'JP_Sales' ,
          'Other_Sales' ,
```

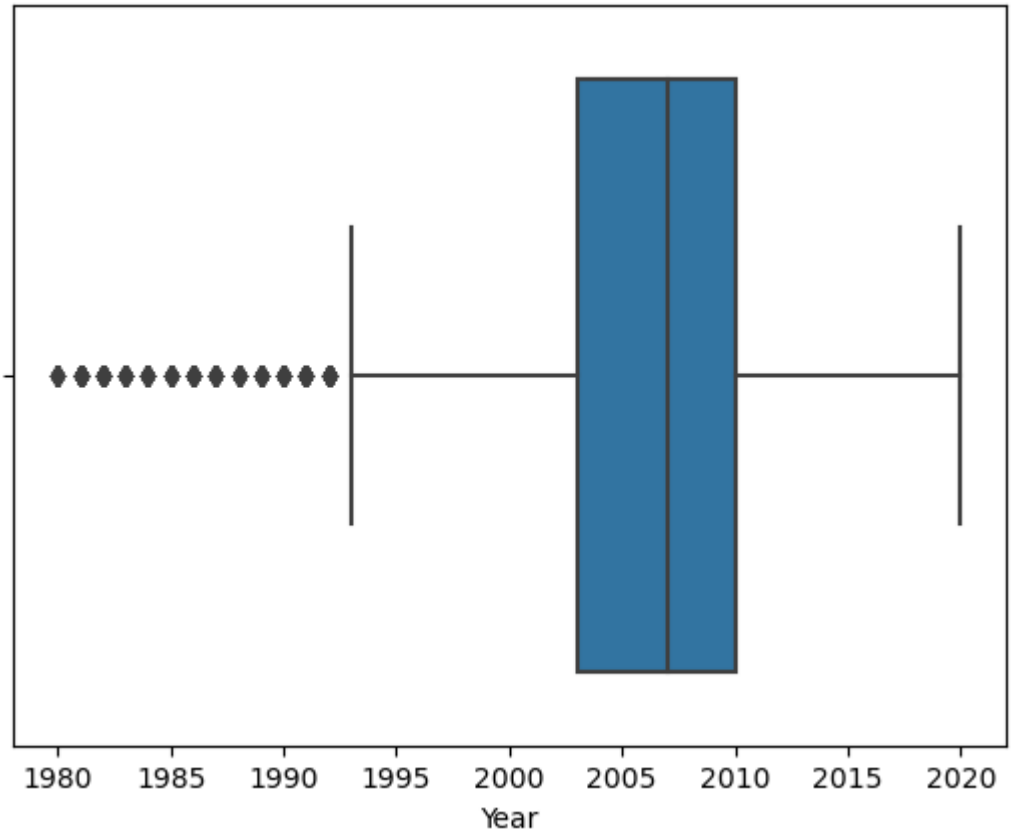
```
          'Global_Sales' ,
          'Platform' , 'Genre' , 'Publisher']].hist()
mpl.savefig('churn_hists.jpg')
mpl.tight_layout()
```



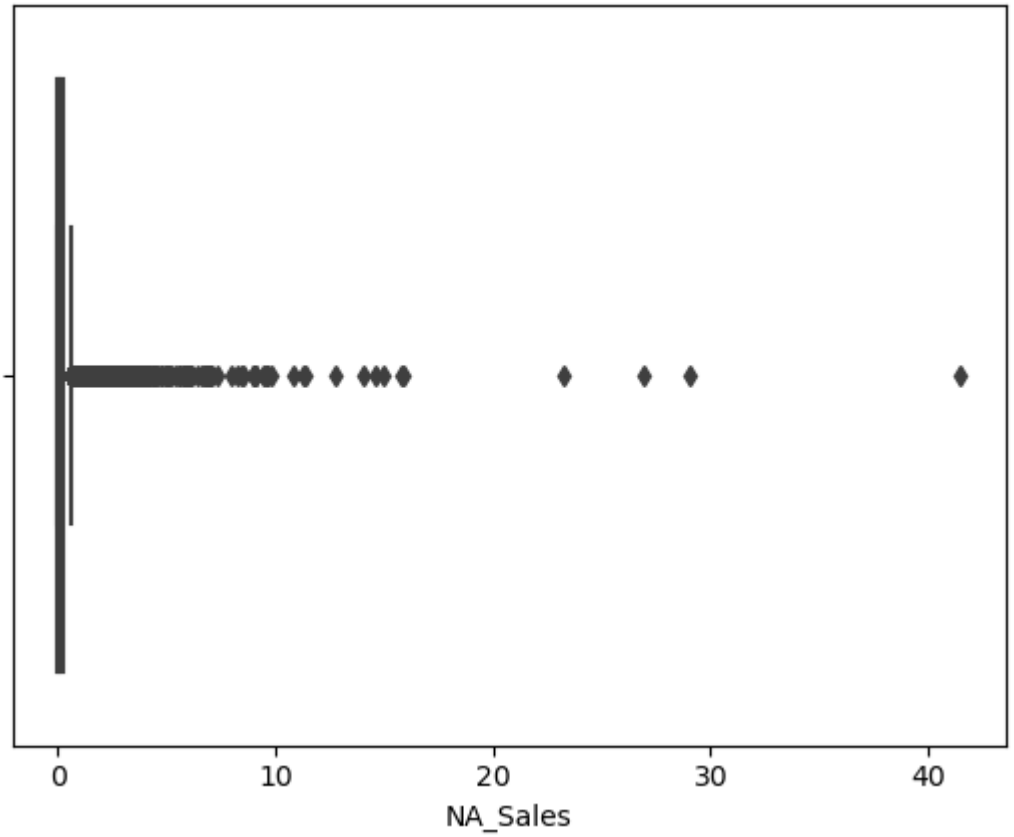
In [13]:
```
#Create Boxplots for our continuous variables
sb.boxplot('Year' , data = df)
mpl.show()

sb.boxplot('NA_Sales' , data = df)
mpl.show()

sb.boxplot('EU_Sales' , data = df)
mpl.show()

sb.boxplot('JP_Sales' , data = df)
mpl.show()

sb.boxplot('Other_Sales' , data = df)
mpl.show()

sb.boxplot('Global_Sales' , data = df)
mpl.show()

sb.boxplot('Platform' , data = df)
mpl.show()

sb.boxplot('Genre' , data = df)
mpl.show()
```

```
sb.boxplot('Publisher' , data = df)
mpl.show()
```

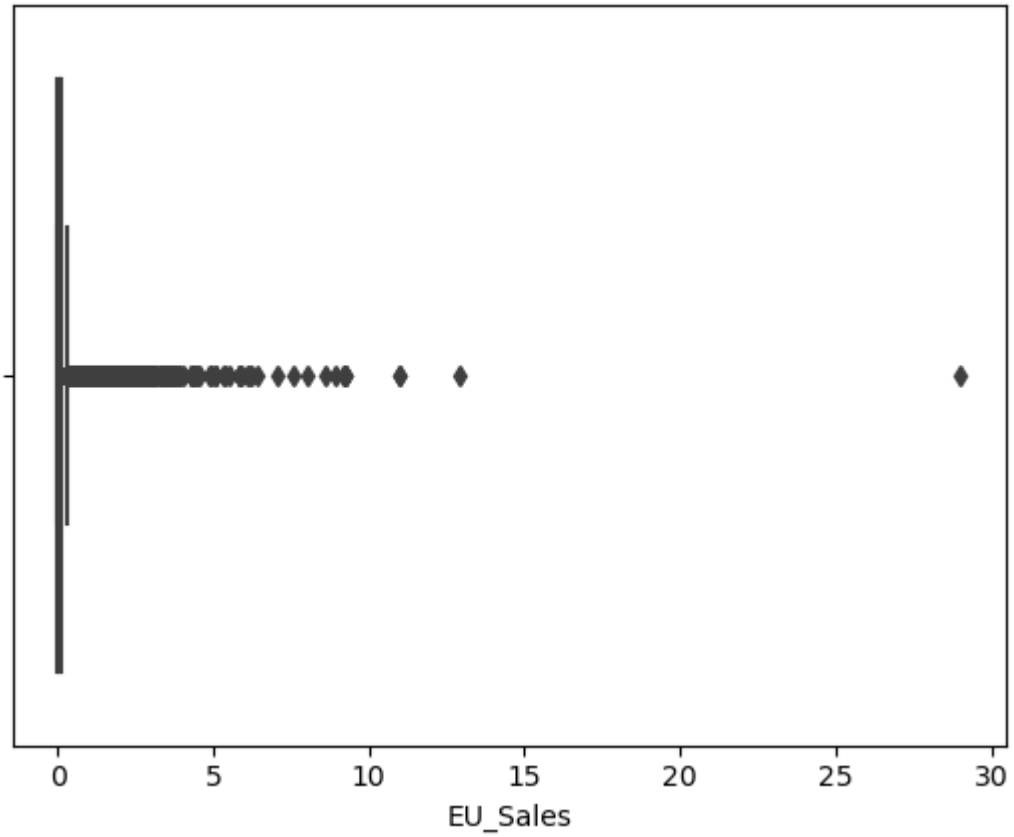
```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



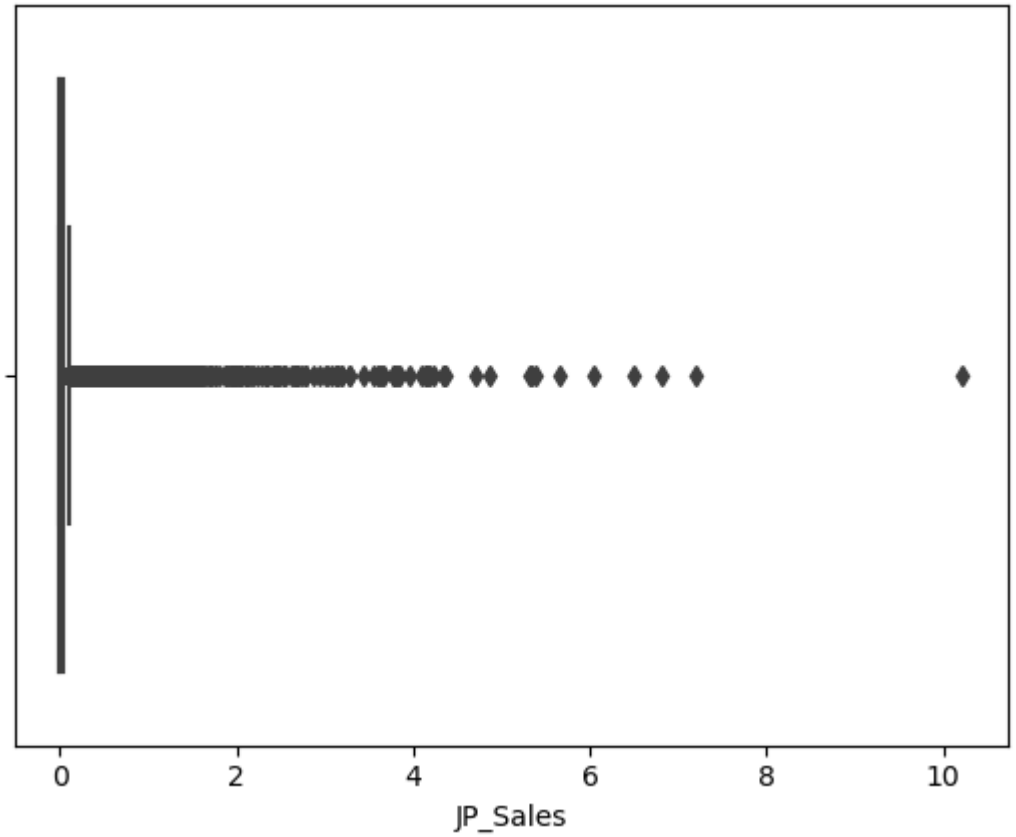

```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



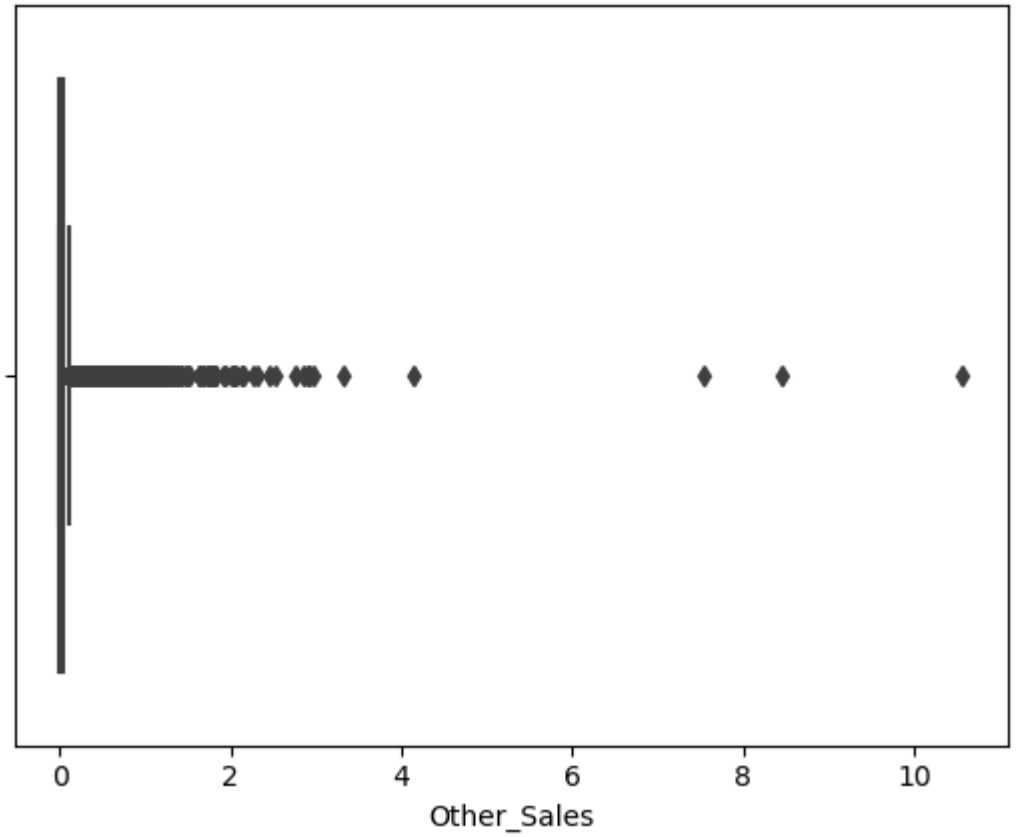

```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



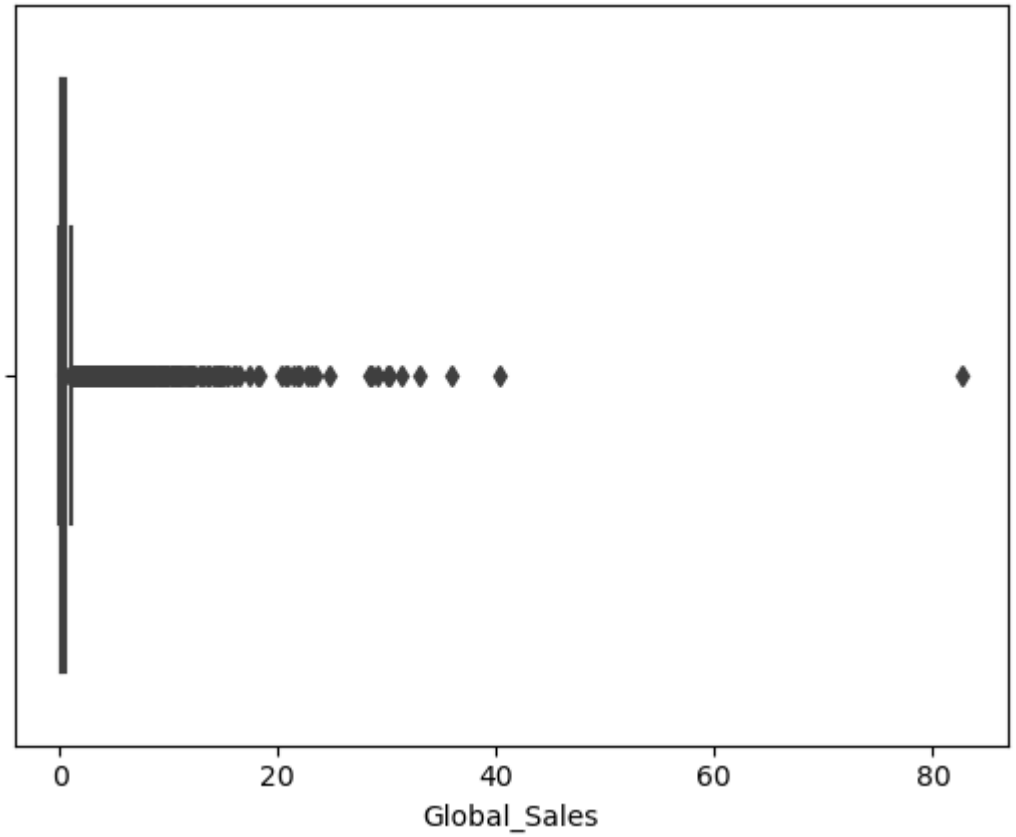
```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



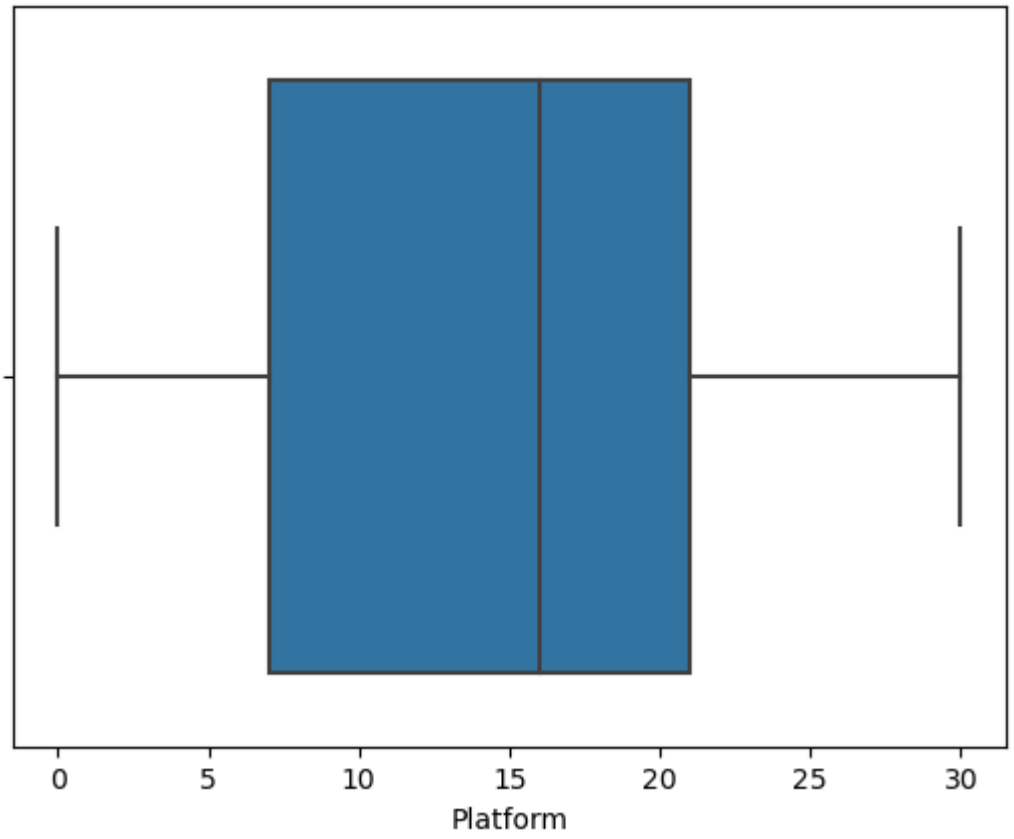

```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



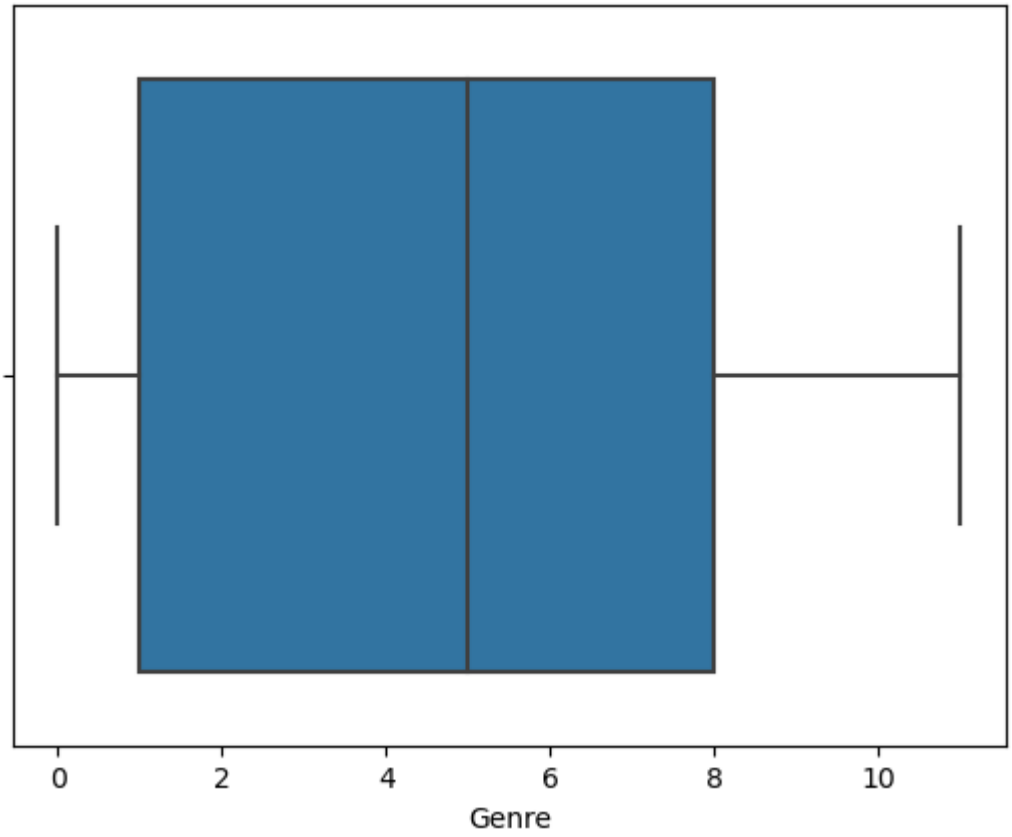

```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



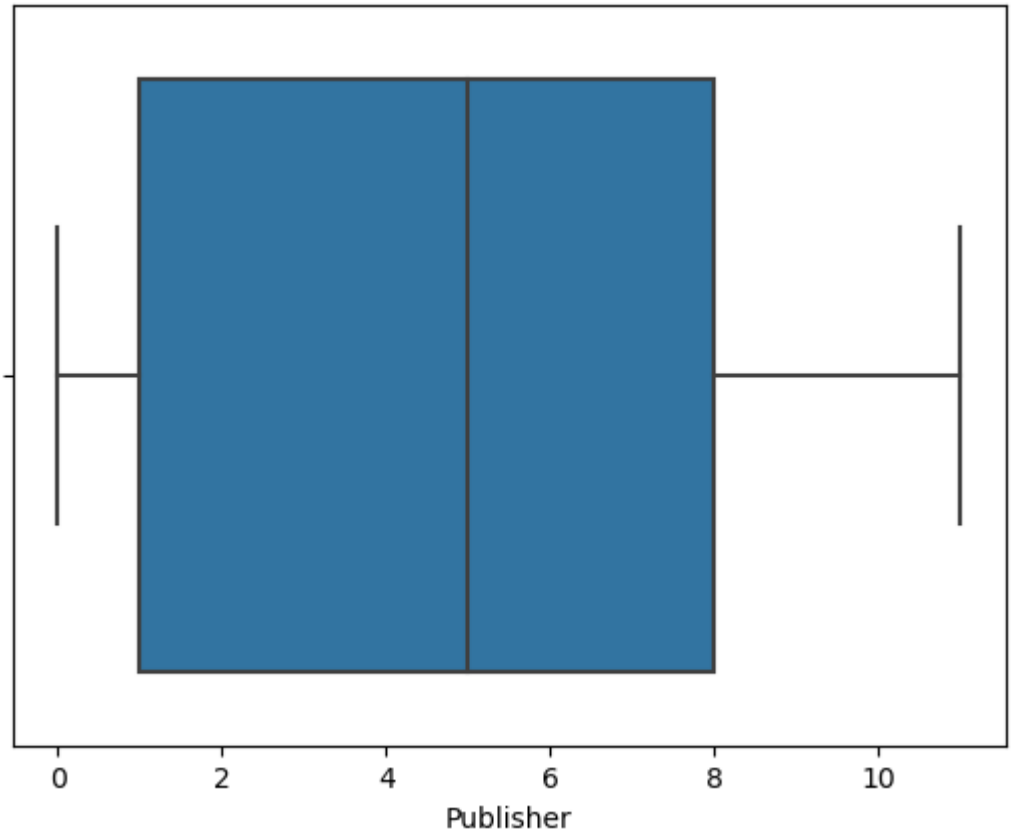
```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```



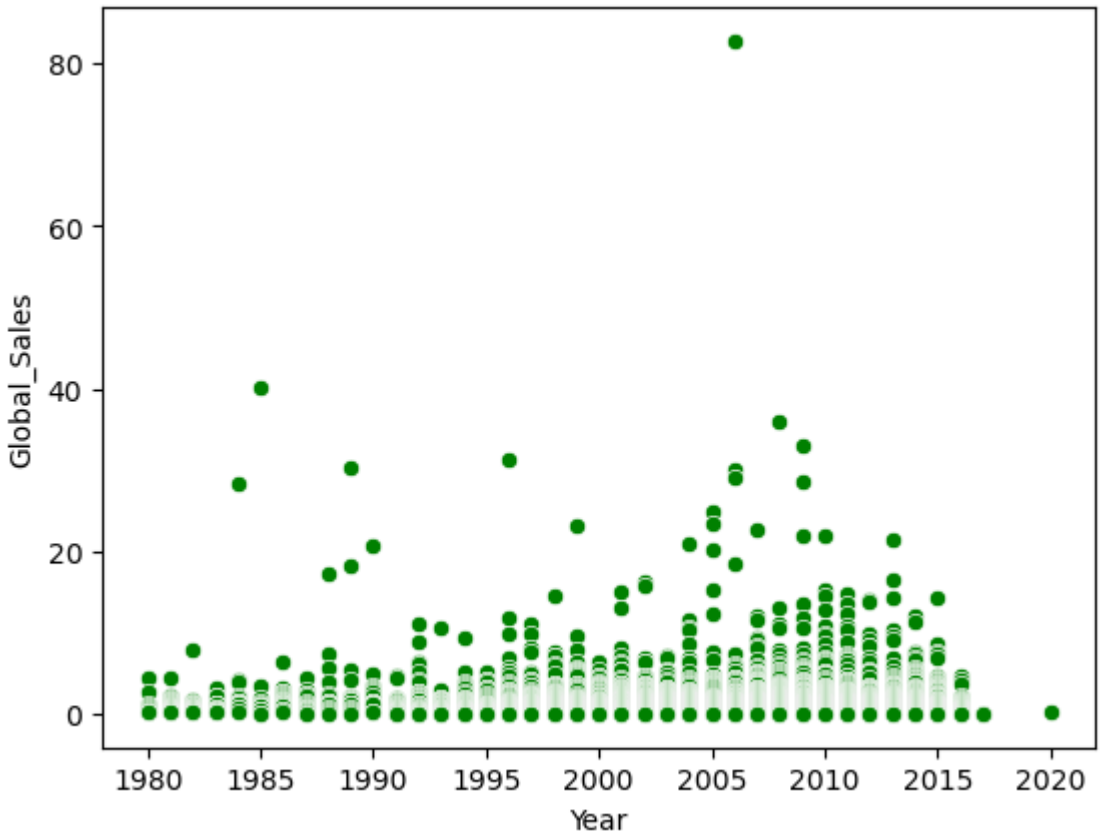

```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```

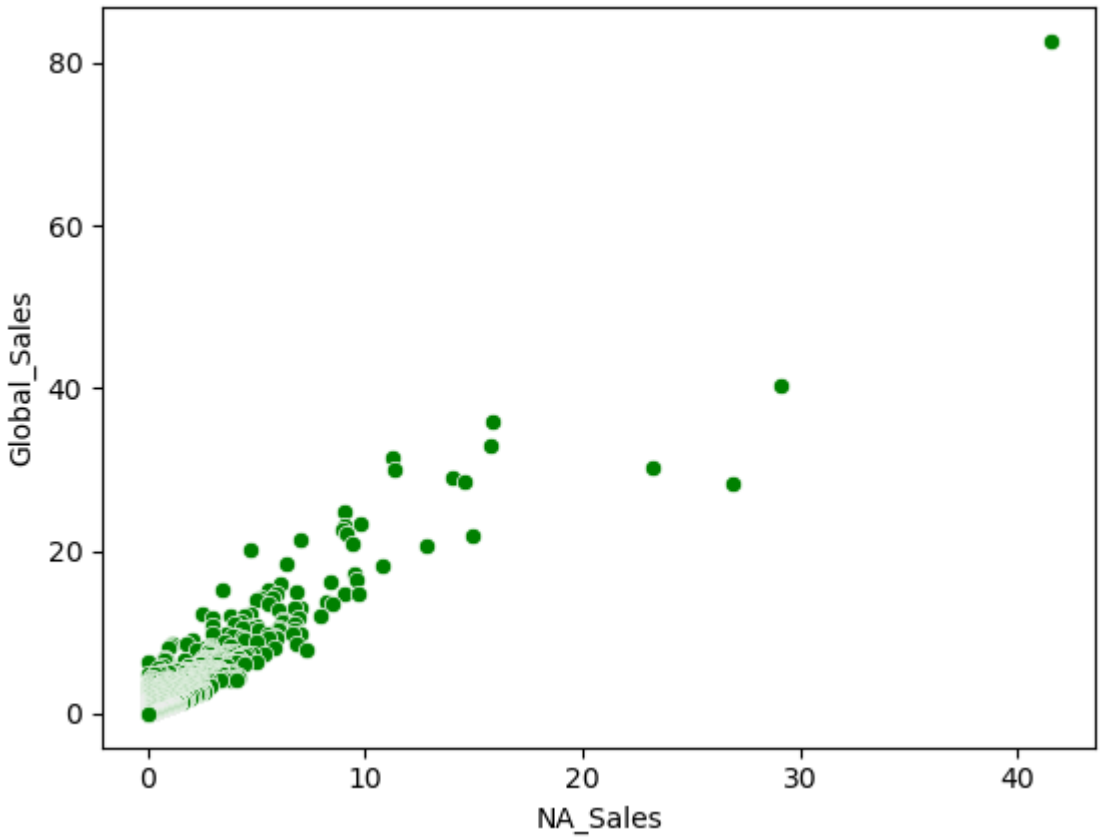


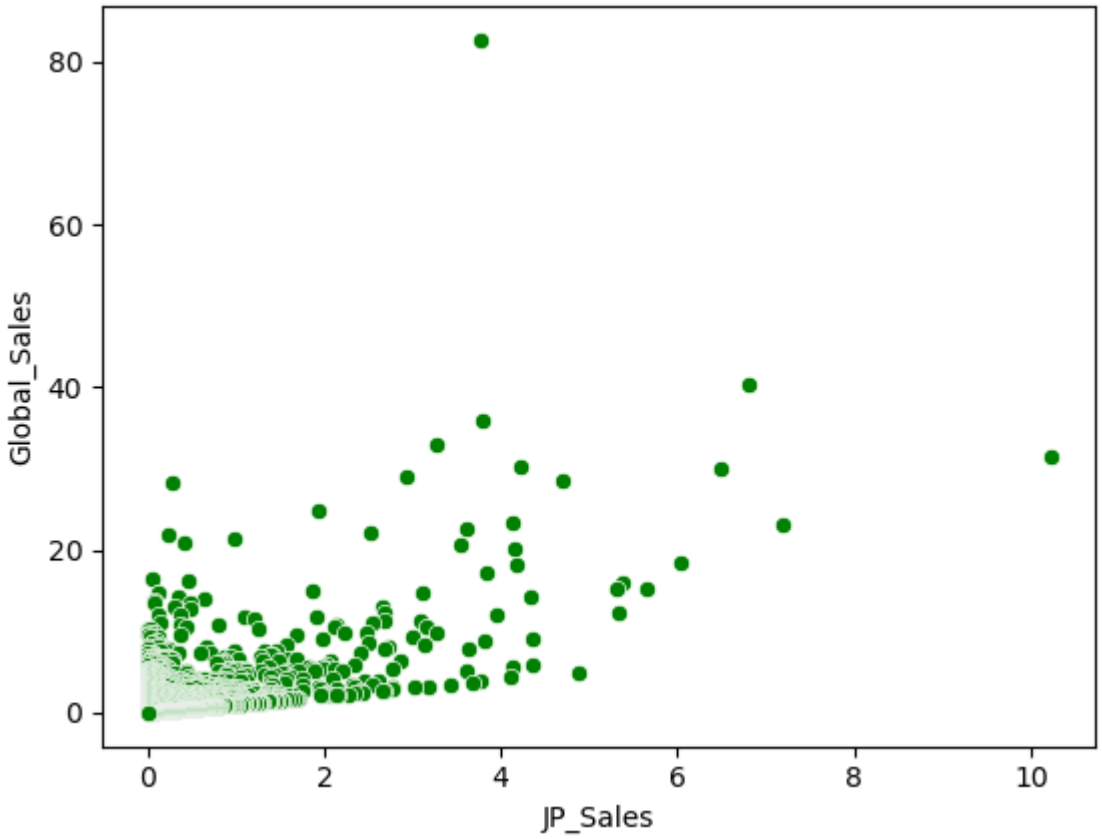

```
C:\Users\e202271009\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarn
ing: Pass the following variable as a keyword arg: x. From version 0.12, the only val
id positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  warnings.warn(
```

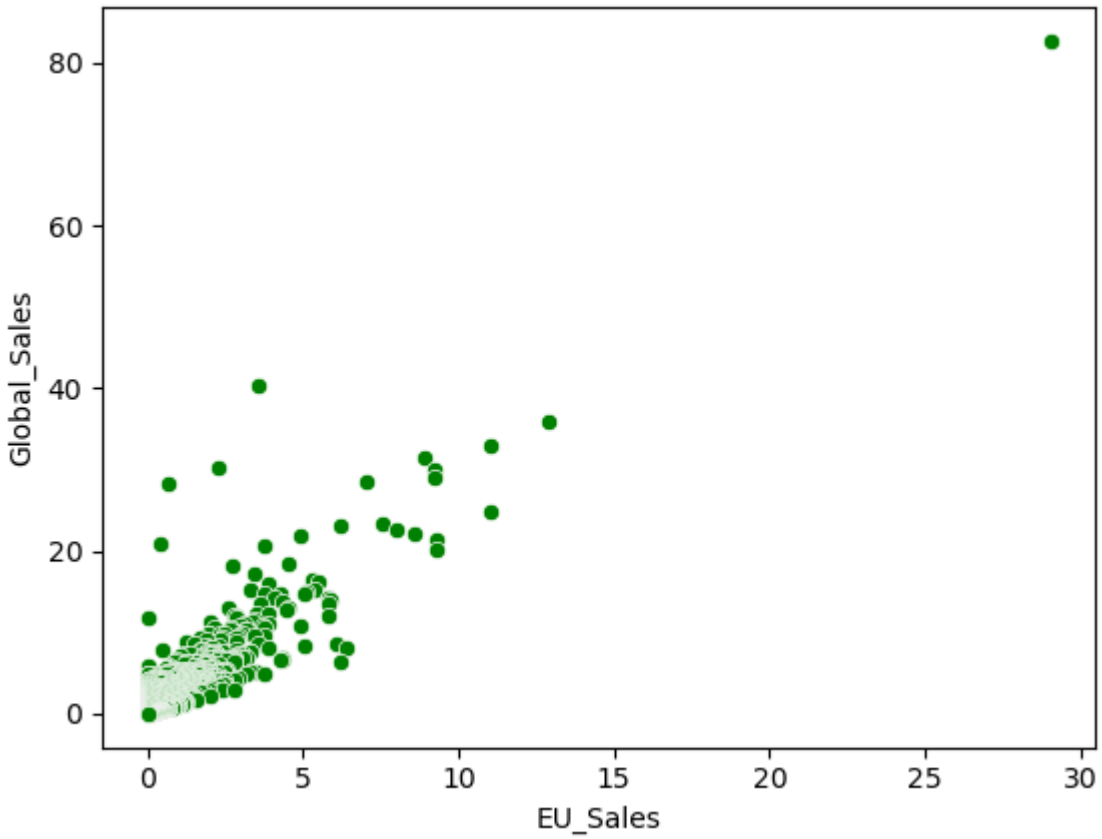


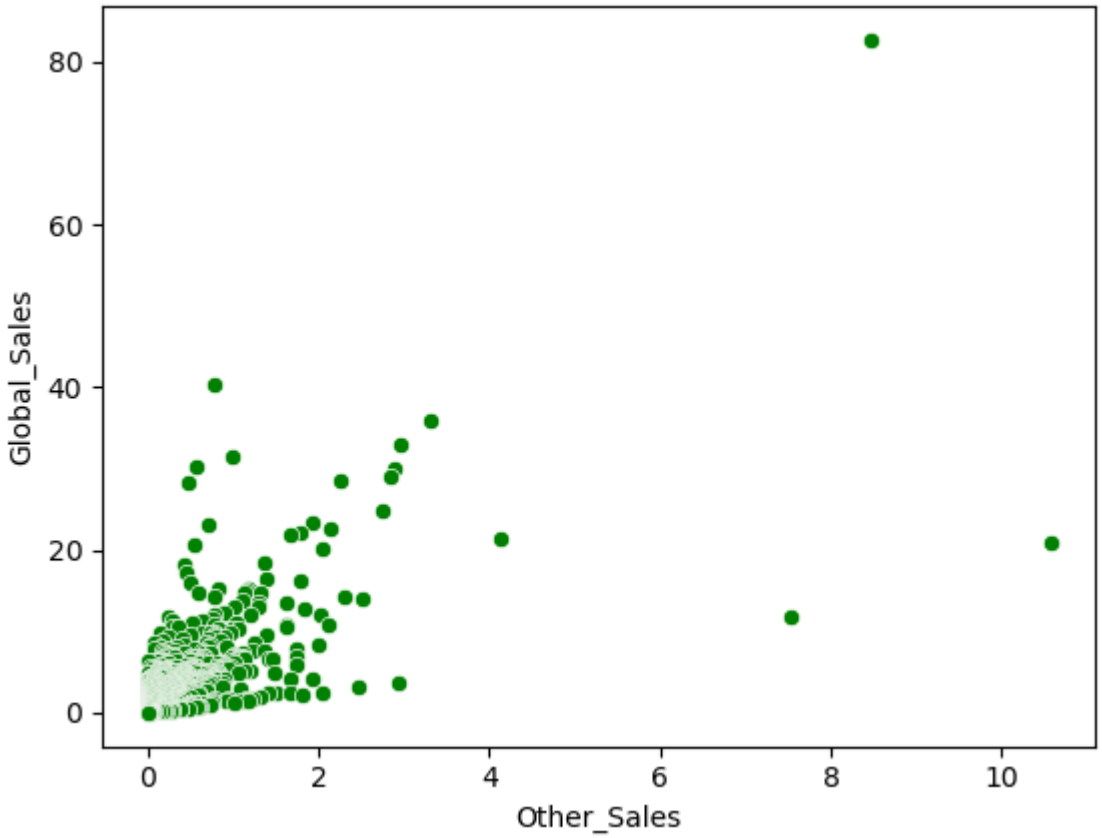
In [14]:
```
#Scatterplots to show direct or inverse relationships between the target & independent
sb.scatterplot(x=df['Year'], y=df['Global_Sales'],
               color='green')
mpl.show()
sb.scatterplot(x=df['NA_Sales'], y=df['Global_Sales'],
               color='green')
mpl.show()
sb.scatterplot(x=df['JP_Sales'], y=df['Global_Sales'],
               color='green')
mpl.show()
sb.scatterplot(x=df['EU_Sales'], y=df['Global_Sales'],
               color='green')
mpl.show()
sb.scatterplot(x=df['Other_Sales'], y=df['Global_Sales'],
               color='green')
mpl.show()
sb.scatterplot(x=df['Platform'], y=df['Global_Sales'],
               color='green')
mpl.show()
sb.scatterplot(x=df['Genre'], y=df['Global_Sales'],
               color='green')
mpl.show()
sb.scatterplot(x=df['Publisher'], y=df['Global_Sales'],
               color='green')
mpl.show()
```

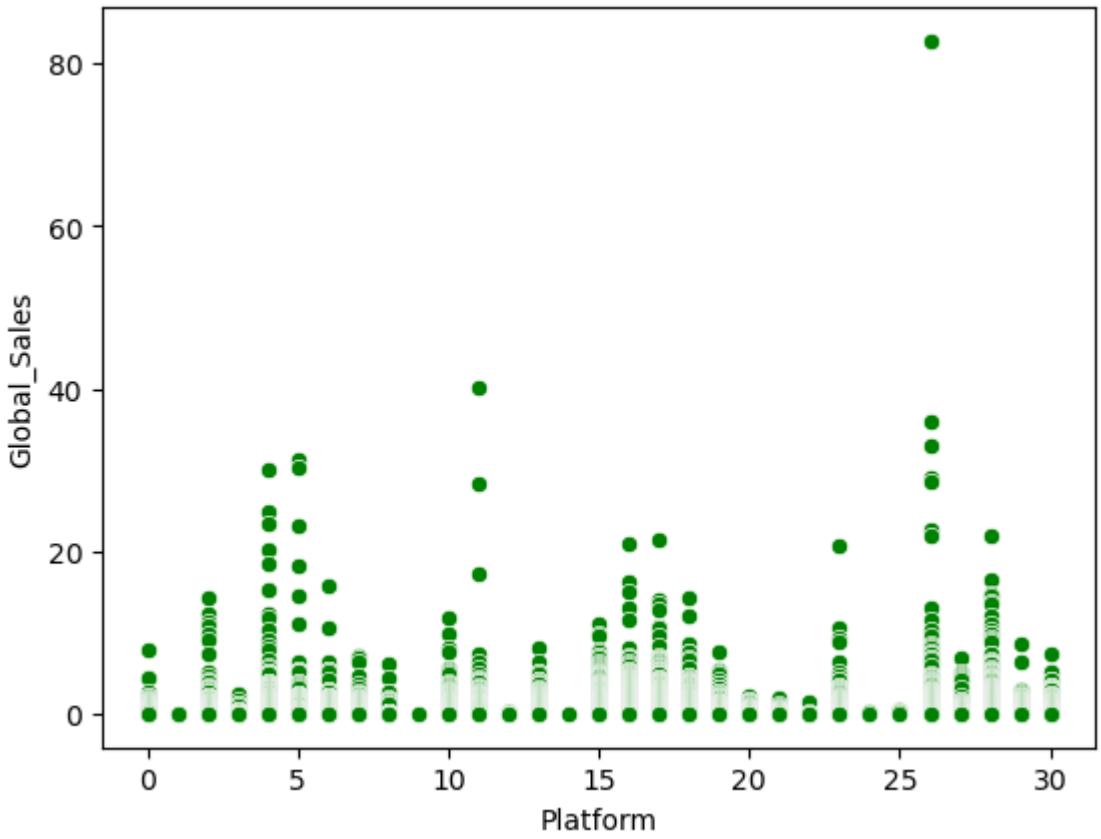


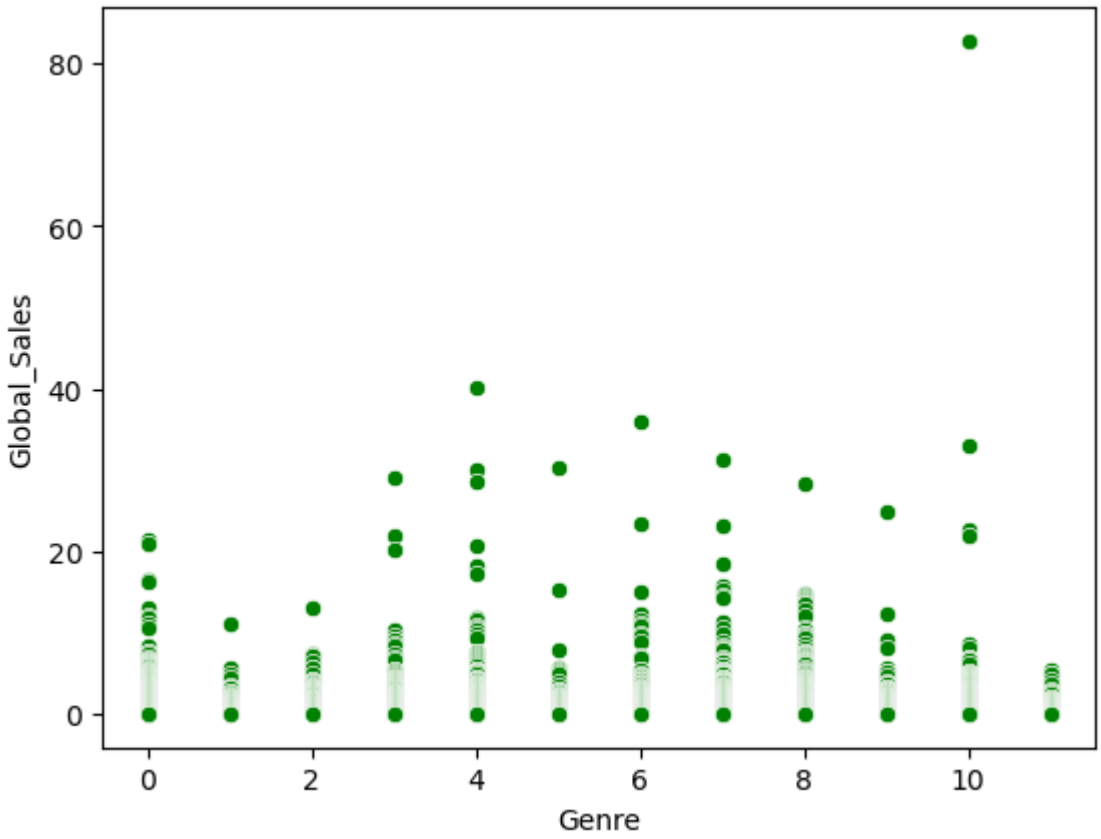

Global_Sales
NA_Sales
0
20
40
60
80
0
10
20
30
40

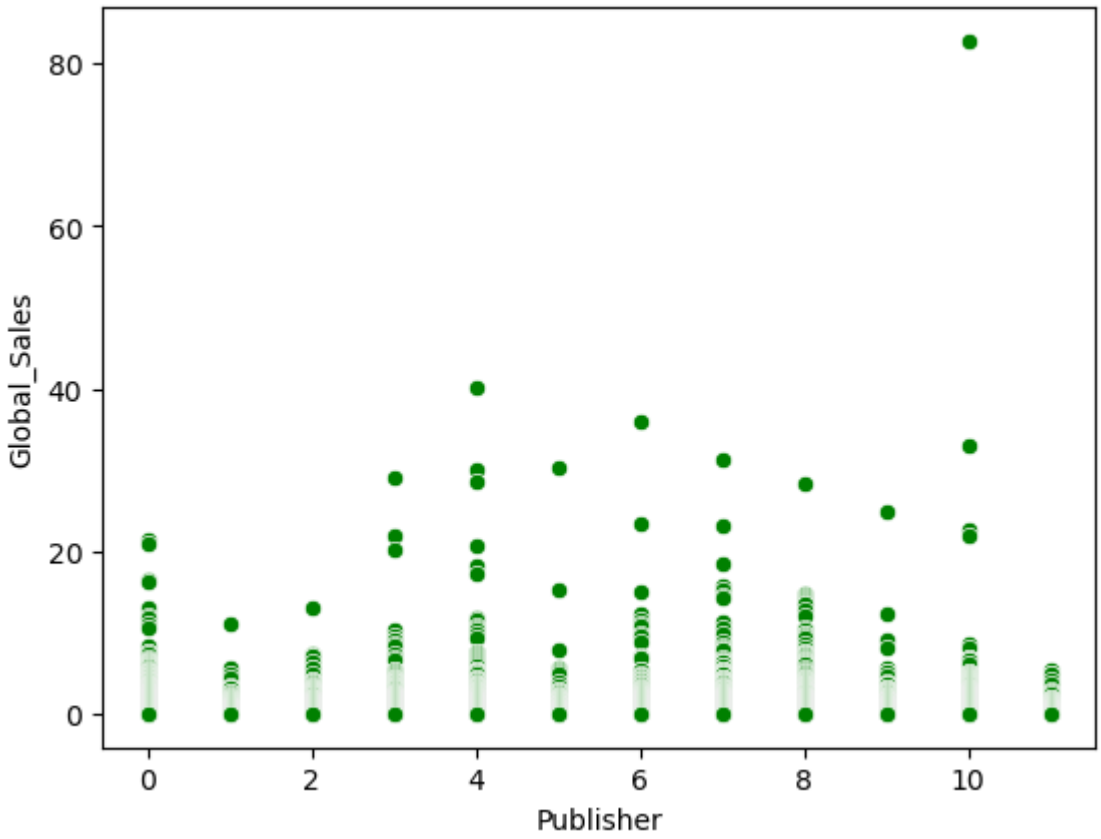

Global_Sales
JP_Sales
0
20
40
60
80
0
2
4
6
8
10


Global_Sales
EU_Sales
0
20
40
60
80
0
5
10
15
20
25
30


Global_Sales
Other_Sales
0
20
40
60
80
0
2
4
6
8
10


Global_Sales
Platform
0
20
40
60
80
0
5
10
15
20
25
30


Global_Sales
Genre
0
20
40
60
80
0
2
4
6
8
10



In [15]:
```
df.to_csv('Sales_Cleaned.csv', index = False)
```

In [16]:
```
# Create initial estimated regression equation that could be used to predict Global_Sa
LMR = ols("Global_Sales ~ Year + Publisher + Platform + NA_Sales + EU_Sales + JP_Sales
print(LMR.params)
print(LMR.summary())
```

```
Intercept      6.994607e-03
Year          -3.251709e-06
Publisher     -6.028122e-07
Platform      -9.288370e-06
NA_Sales       9.999536e-01
EU_Sales       9.999913e-01
JP_Sales       9.998387e-01
Other_Sales    9.996273e-01
Genre         -6.028122e-07
dtype: float64
                            OLS Regression Results
==============================================================================
Dep. Variable:           Global_Sales   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 2.095e+08
Date:                Thu, 09 Mar 2023   Prob (F-statistic):               0.00
Time:                        10:17:09   Log-Likelihood:                 62490.
No. Observations:               16291   AIC:                        -1.250e+05
Df Residuals:                   16283   BIC:                        -1.249e+05
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0070      0.015      0.473      0.636      -0.022       0.036
Year        -3.252e-06   7.36e-06     -0.442      0.659   -1.77e-05    1.12e-05
Publisher   -6.028e-07   5.49e-06     -0.110      0.913   -1.14e-05    1.02e-05
Platform    -9.288e-06   4.99e-06     -1.860      0.063   -1.91e-05    5.02e-07
NA_Sales        1.0000   8.16e-05   1.22e+04      0.000       1.000       1.000
EU_Sales        1.0000      0.000   6829.269      0.000       1.000       1.000
JP_Sales        0.9998      0.000   6580.169      0.000       1.000       1.000
Other_Sales     0.9996      0.000   3125.488      0.000       0.999       1.000
Genre       -6.028e-07   5.49e-06     -0.110      0.913   -1.14e-05    1.02e-05
==============================================================================
Omnibus:                      213.542   Durbin-Watson:                   1.612
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              384.363
Skew:                           0.048   Prob(JB):                     3.44e-84
Kurtosis:                       3.746   Cond. No.                     7.39e+18
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
[2] The smallest eigenvalue is 1.2e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

In [17]:

```
#Try again without the other sales information
LMR = ols("Global_Sales ~ Year + Publisher + Platform + Genre", data=df).fit()
print(LMR.params)
print(LMR.summary())
```

```
Intercept    43.608796
Year         -0.021537
Publisher     0.002306
Platform      0.007691
Genre         0.002306
dtype: float64
```

OLS Regression Results

Dep. Variable:	Global_Sales	R-squared:	0.007
Model:	OLS	Adj. R-squared:	0.007
Method:	Least Squares	F-statistic:	40.34
Date:	Thu, 09 Mar 2023	Prob (F-statistic):	5.80e-26
Time:	10:17:11	Log-Likelihood:	-30376.
No. Observations:	16291	AIC:	6.076e+04
Df Residuals:	16287	BIC:	6.079e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	43.6088	4.305	10.129	0.000	35.170	52.048
Year	-0.0215	0.002	-10.032	0.000	-0.026	-0.017
Publisher	0.0023	0.002	1.405	0.160	-0.001	0.006
Platform	0.0077	0.001	5.180	0.000	0.005	0.011
Genre	0.0023	0.002	1.405	0.160	-0.001	0.006

Omnibus:	33764.736	Durbin-Watson:	0.061
Prob(Omnibus):	0.000	Jarque-Bera (JB):	245097764.986
Skew:	17.360	Prob(JB):	0.00
Kurtosis:	602.895	Cond. No.	7.30e+18

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
[2] The smallest eigenvalue is 1.23e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

In [18]:
```
#Stepwise Backwards reduction with a P value cutoff of p > .05 from the regression mod
#After removing all variables with a P value > .05, we are left with the following var
```

In [19]:
```
#Now we will use Variance Inflation Factor (VIF) to reduce our model. For the output,
from patsy import dmatrices
from statsmodels.stats.outliers_influence import variance_inflation_factor

#find design matrix for linear regression model using 'rating' as response variable
y, X = dmatrices('Global_Sales ~ Year + Platform + NA_Sales + EU_Sales + JP_Sales + Ot

#calculate VIF for each explanatory variable
VIF = pd.DataFrame()
VIF['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
VIF['variable'] = X.columns

#view VIF for each explanatory variable
VIF
```

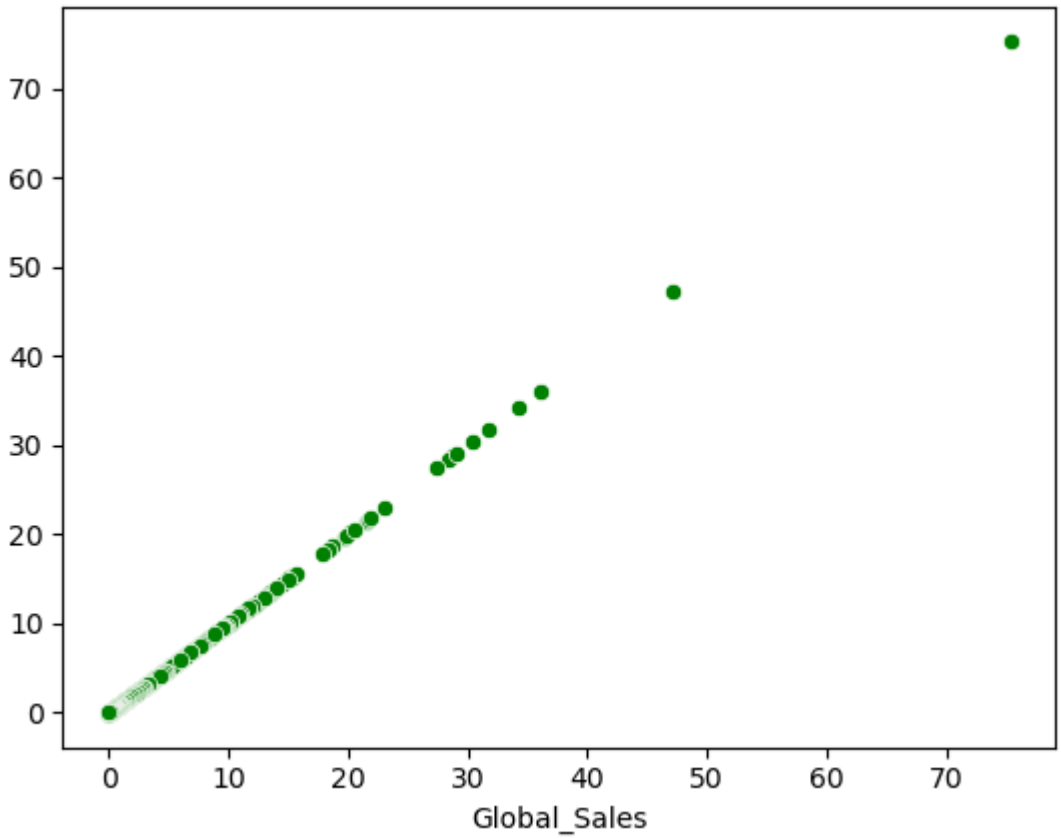
Out[19]:

	VIF	variable
0	127983.692443	Intercept
1	1.082921	Year
2	1.040894	Platform
3	2.690268	NA_Sales
4	3.319718	EU_Sales
5	1.340652	JP_Sales
6	2.206489	Other_Sales

In [20]:
```
# Evcerything above 3 VIF was removed to have a very strict cutoff
```

In [21]:
```
# Create reduced OLS multiple regression using the ols feature. We are compiling our f
LM_Reduced = ols("Global_Sales ~ Year + Platform + NA_Sales + JP_Sales + Other_Sales",
print(LM_Reduced.params)
print(LM_Reduced.summary())

# Extract Clean dataset
df.to_csv('Capstone_Data.csv')

Residuals = df['Global_Sales'] = LM_Reduced.predict(df[['Global_Sales', 'Year', 'Platf
sb.scatterplot(x=df['Global_Sales'],y=Residuals,color='green')
mpl.show()
```

```
Intercept     -9.174632
Year           0.004572
Platform       0.000382
NA_Sales       1.292622
JP_Sales       1.194827
Other_Sales    2.042369
dtype: float64
                            OLS Regression Results
==============================================================================
Dep. Variable:           Global_Sales   R-squared:                       0.968
Model:                            OLS   Adj. R-squared:                  0.968
Method:                 Least Squares   F-statistic:                 9.911e+04
Date:                Thu, 09 Mar 2023   Prob (F-statistic):               0.00
Time:                        10:17:21   Log-Likelihood:                -2352.8
No. Observations:               16291   AIC:                             4718.
Df Residuals:                   16285   BIC:                             4764.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      -9.1746      0.780    -11.756      0.000     -10.704      -7.645
Year            0.0046      0.000     11.746      0.000       0.004       0.005
Platform        0.0004      0.000      1.429      0.153      -0.000       0.001
NA_Sales        1.2926      0.004    347.606      0.000       1.285       1.300
JP_Sales        1.1948      0.008    149.567      0.000       1.179       1.210
Other_Sales     2.0424      0.015    135.762      0.000       2.013       2.072
==============================================================================
Omnibus:                    23368.617   Durbin-Watson:                   2.059
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        204487635.835
Skew:                          -7.354   Prob(JB):                         0.00
Kurtosis:                     551.668   Cond. No.                     7.15e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
ified.
[2] The condition number is large, 7.15e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```



In [22]:
```
# Here we are calculating our residual error from the reduced model
print(np.sqrt(LM_Reduced.mse_resid))
```

```
0.2796173489745299
```

In [23]:
```
# Create reduced OLS multiple regression using the ols feature without other sales inf
LM_Reduced = ols("Global_Sales ~ Year + Platform", data=df).fit()
print(LM_Reduced.params)
print(LM_Reduced.summary())

# Extract Clean dataset
df.to_csv('Capstone_Data.csv')

Residuals = df['Global_Sales'] = LM_Reduced.predict(df[['Global_Sales', 'Year', 'Platf
sb.scatterplot(x=df['Global_Sales'],y=Residuals,color='green')
mpl.show()
```

```
Intercept    44.427467
Year         -0.021935
Platform      0.007792
dtype: float64
```

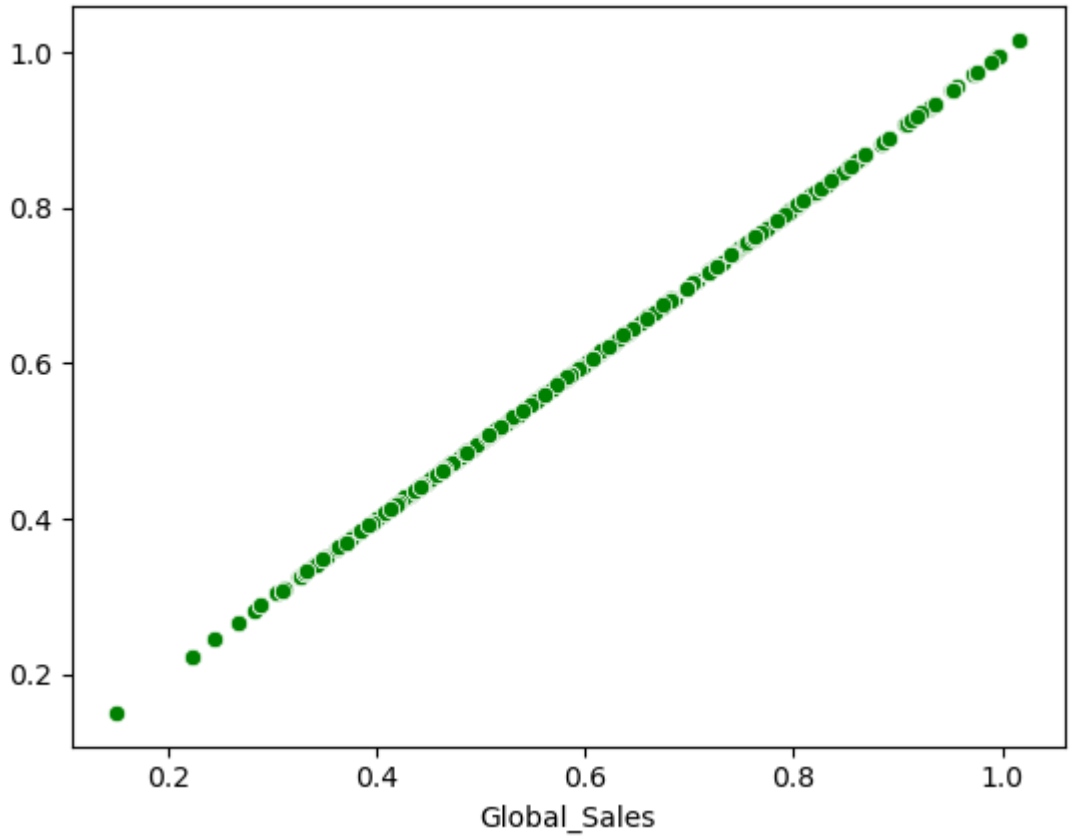
OLS Regression Results

Dep. Variable:	Global_Sales	R-squared:	0.007
Model:	OLS	Adj. R-squared:	0.007
Method:	Least Squares	F-statistic:	61.49
Date:	Thu, 09 Mar 2023	Prob (F-statistic):	2.49e-27
Time:	10:17:27	Log-Likelihood:	-30112.
No. Observations:	16291	AIC:	6.023e+04
Df Residuals:	16288	BIC:	6.025e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	44.4275	4.197	10.585	0.000	36.201	52.654
Year	-0.0219	0.002	-10.476	0.000	-0.026	-0.018
Platform	0.0078	0.001	5.341	0.000	0.005	0.011

Omnibus:	33127.105	Durbin-Watson:	0.102
Prob(Omnibus):	0.000	Jarque-Bera (JB):	182607991.288
Skew:	16.713	Prob(JB):	0.00
Kurtosis:	520.592	Cond. No.	7.00e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7e+05. This might indicate that there are strong multicollinearity or other numerical problems.



In [24]:
```
# Here we are calculating our residual error from the reduced model
print(np.sqrt(LM_Reduced.mse_resid))
```

```
1.536513586801373
```

In [25]:
```
df
```

Out[25]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Platform	Genre	Publ
0	1	2006.0	41.49	29.02	3.77	8.46	0.629189	26	10	
1	2	1985.0	29.08	3.58	6.81	0.77	0.972930	11	4	
2	3	2008.0	15.85	12.88	3.79	3.31	0.585319	26	6	
3	4	2009.0	15.75	11.01	3.28	2.96	0.563385	26	10	
4	5	1996.0	11.27	8.89	10.22	1.00	0.684895	5	7	
...	...	...	...	...	...	...	...	...	...	
16593	16596	2002.0	0.01	0.00	0.00	0.00	0.561079	6	4	
16594	16597	2003.0	0.01	0.00	0.00	0.00	0.546937	7	8	
16595	16598	2008.0	0.00	0.00	0.00	0.00	0.507395	16	6	
16596	16599	2010.0	0.00	0.01	0.00	0.00	0.370017	4	5	
16597	16600	2003.0	0.01	0.00	0.00	0.00	0.539144	6	4	

16291 rows × 10 columns

In [ ]:
```

```