Big Data Case Study

Analyzing Global Video Games Sales Data

Introduction

With the explosion of digital data, industries are turning to Big Data technologies to extract valuable insights from large and complex datasets. This case study demonstrates how Hadoop ecosystem tools — HDFS, Sqoop, and Hive — can be used for data ingestion, storage, processing, and analysis. We analyze a real-world dataset of video game sales, exploring trends in platforms, publishers, and regional performance.

Details of Dataset

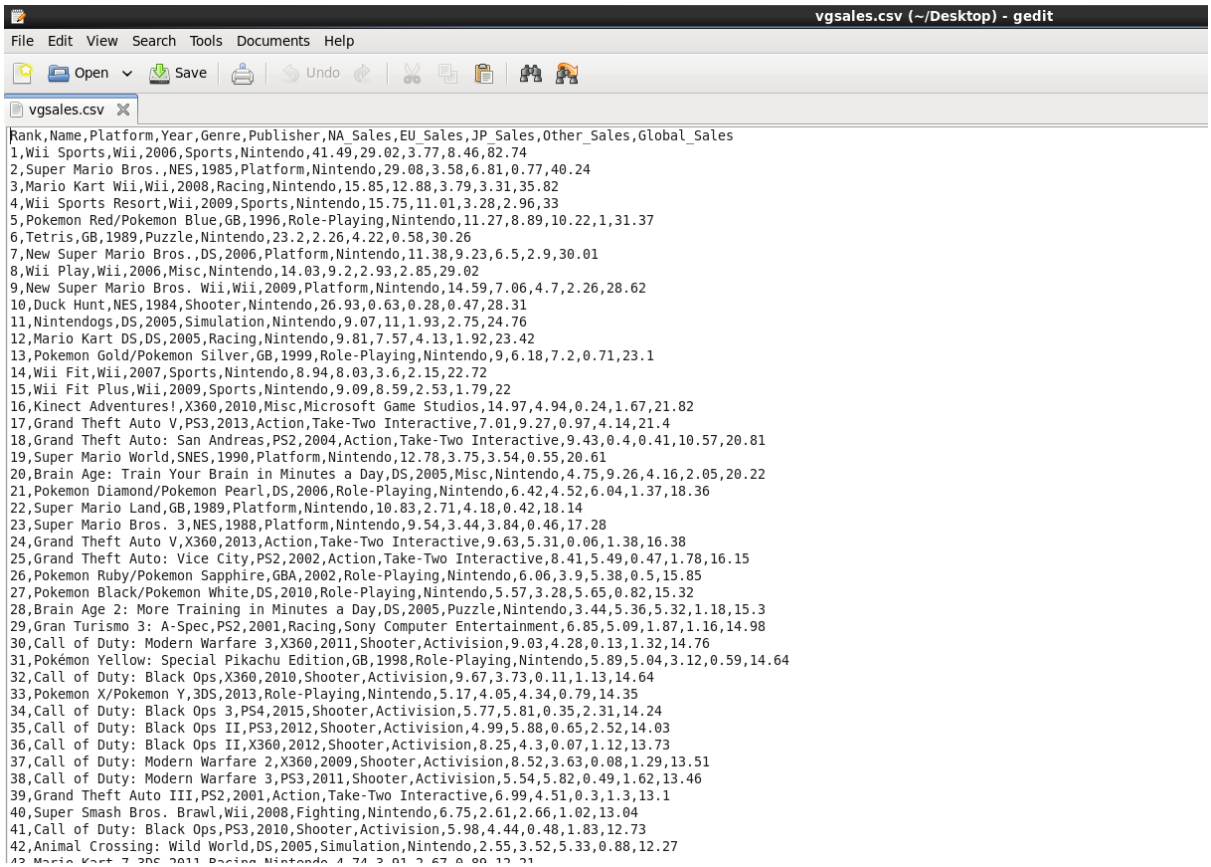
Dataset name: vgsales.csv
Source: Kaggle – Video Game Sales Dataset
Description: Contains sales information of video games across multiple regions.

Column Name	Description
Rank	Rank of the game based on global sales
Name	Name of the video game
Platform	Platform on which the game was released (e.g., PS4, Xbox, PC)
Year	Year of release
Genre	Type of game (e.g., Action, Sports, RPG)
Publisher	Company that published the game
NA_Sales (million)	Sales in North America

Column Name	Description
EU_Sales (million)	Sales in Europe
JP_Sales (million)	Sales in Japan
Other_Sales (million)	Sales in other regions
Global_Sales (million)	Total worldwide sales

Total records: ~16,500
File format: CSV (Comma-separated values)

A part of the dataset is:




```
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
21,Pokemon Diamond/Pokemon Pearl,DS,2006,Role-Playing,Nintendo,6.42,4.52,6.04,1.37,18.36
22,Super Mario Land,GB,1989,Platform,Nintendo,10.83,2.71,4.18,0.42,18.14
23,Super Mario Bros. 3,NES,1988,Platform,Nintendo,9.54,3.44,3.84,0.46,17.28
24,Grand Theft Auto V,X360,2013,Action,Take-Two Interactive,9.63,5.31,0.06,1.38,16.38
25,Grand Theft Auto: Vice City,PS2,2002,Action,Take-Two Interactive,8.41,5.49,0.47,1.78,16.15
26,Pokemon Ruby/Pokemon Sapphire,GBA,2002,Role-Playing,Nintendo,6.06,3.9,5.38,0.5,15.85
27,Pokemon Black/Pokemon White,DS,2010,Role-Playing,Nintendo,5.57,3.28,5.65,0.82,15.32
28,Brain Age 2: More Training in Minutes a Day,DS,2005,Puzzle,Nintendo,3.44,5.36,5.32,1.18,15.3
29,Gran Turismo 3: A-Spec,PS2,2001,Racing,Sony Computer Entertainment,6.85,5.09,1.87,1.16,14.98
30,Call of Duty: Modern Warfare 3,X360,2011,Shooter,Activision,9.03,4.28,0.13,1.32,14.76
31,Pokémon Yellow: Special Pikachu Edition,GB,1998,Role-Playing,Nintendo,5.89,5.04,3.12,0.59,14.64
32,Call of Duty: Black Ops,X360,2010,Shooter,Activision,9.67,3.73,0.11,1.13,14.64
33,Pokemon X/Pokemon Y,3DS,2013,Role-Playing,Nintendo,5.17,4.05,4.34,0.79,14.35
34,Call of Duty: Black Ops 3,PS4,2015,Shooter,Activision,5.77,5.81,0.35,2.31,14.24
35,Call of Duty: Black Ops II,PS3,2012,Shooter,Activision,4.99,5.88,0.65,2.52,14.03
36,Call of Duty: Black Ops II,X360,2012,Shooter,Activision,8.25,4.3,0.07,1.12,13.73
37,Call of Duty: Modern Warfare 2,X360,2009,Shooter,Activision,8.52,3.63,0.08,1.29,13.51
38,Call of Duty: Modern Warfare 3,PS3,2011,Shooter,Activision,5.54,5.82,0.49,1.62,13.46
39,Grand Theft Auto III,PS2,2001,Action,Take-Two Interactive,6.99,4.51,0.3,1.3,13.1
40,Super Smash Bros. Brawl,Wii,2008,Fighting,Nintendo,6.75,2.61,2.66,1.02,13.04
41,Call of Duty: Black Ops,PS3,2010,Shooter,Activision,5.98,4.44,0.48,1.83,12.73
42,Animal Crossing: Wild World,DS,2005,Simulation,Nintendo,2.55,3.52,5.33,0.88,12.27
43,Mario Kart 7,3DS,2011,Racing,Nintendo,4.74,3.91,2.67,0.89,12.21
```

Project Scope

The objective of this project is to:

- Ingest and process the dataset using Hadoop ecosystem components.
- Perform analysis using **Hive** (Big Data warehouse).
- Generate visual insights into sales trends and publisher performance.

Goals

1. Import dataset into **HDFS** file system.
2. Store and query data using **Hive**.
3. Perform exploratory and comparative data analysis.
4. Visualize results using analytical tools like **Matplotlib**, **Seaborn etc.**

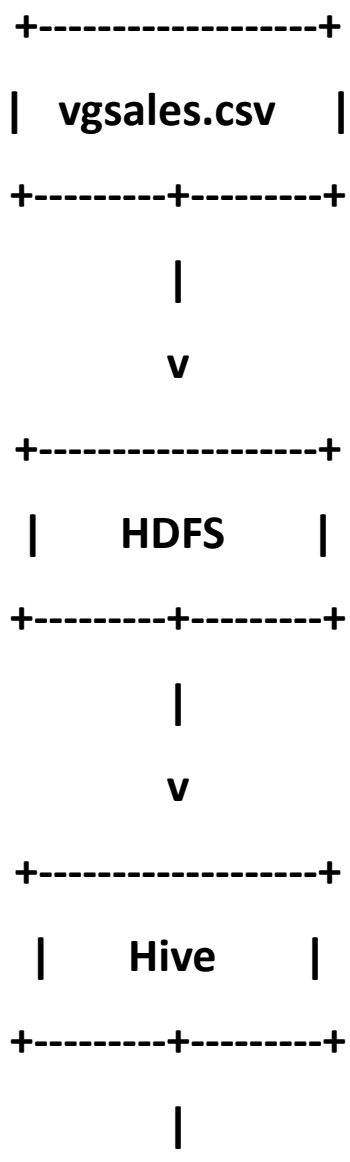
Tools and Working Environment

Tool	Purpose
HDFS	Distributed storage for large-scale data
Hive	Data warehouse for querying using SQL-like syntax
Hadoop	Core distributed processing framework
Python / Matplotlib	Visualization and reporting

Environment Setup:

- Hadoop 3.x
- Hive 3.x
- Cloudera

Data Flow Architecture



v

+------------------+

| Data Visualization|

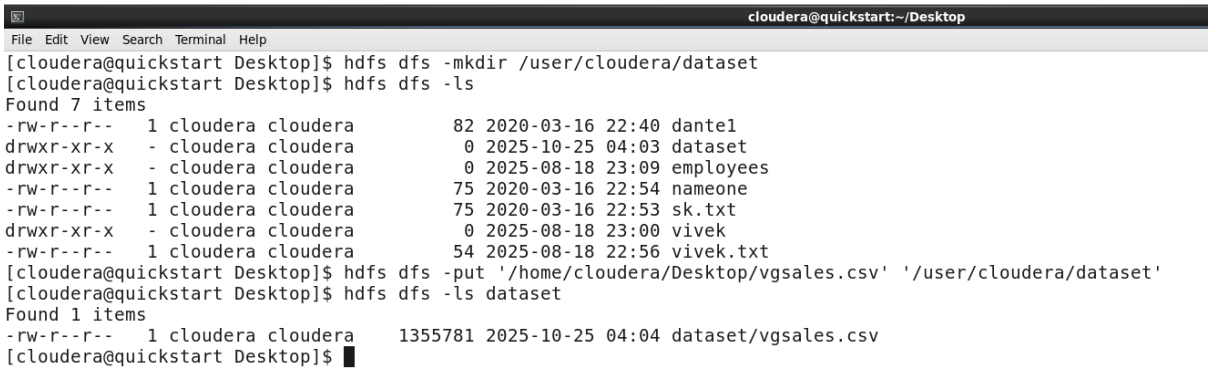
+------------------+

Performing Analysis on Hive

Step 1: Upload Data to HDFS

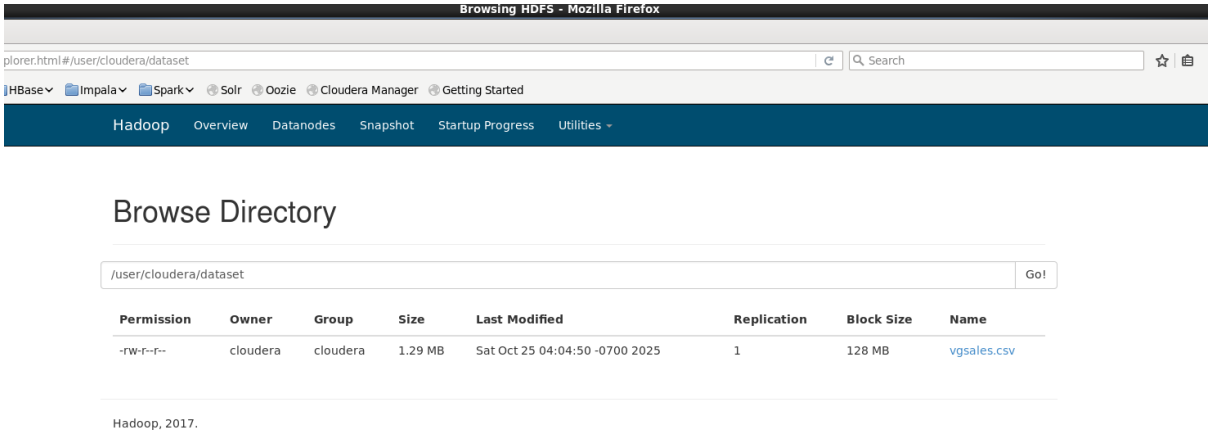
hdfs dfs -mkdir /user/cloudera/dataset/

hdfs dfs -put /home/cloudera/Desktop/vgsales.csv /user/cloudera/dataset



The dataset appears in our hdfs directory.

We can verify using web interface as well.



Step 2: Create Table in Hive

```
CREATE  TABLE vgsales (
  Rank INT,
  Name STRING,
  Platform STRING,
  Year INT,
  Genre STRING,
  Publisher STRING,
  NA_Sales FLOAT,
  EU_Sales FLOAT,
  JP_Sales FLOAT,
  Other_Sales FLOAT,
  Global_Sales FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
```

Step 3: Loading data into Table (vgsales) in Hive

LOAD DATA INPATH '/user/cloudera/dataset/vgsales.csv' OVERWRITE INTO TABLE vgsales

```
hive>
    >
    >
    > CREATE TABLE vgsales (
    > RANK INT,
    > NAME STRING,
    > PLATFORM STRING,
    > YEAR FLOAT,
    > GENRE STRING,
    > PUBLISHER STRING,
    > NA_SALES FLOAT,
    > EU_SALES FLOAT,
    > JP_SALES FLOAT,
    > OTHER_SALES FLOAT,
    > GLOBAL_SALES FLOAT
    > )
    > row format delimited fields terminated by ',' stored as TEXTFILE;
OK
Time taken: 1.711 seconds
hive> show tables;
OK
vgsales
vgsales_small
Time taken: 0.05 seconds, Fetched: 2 row(s)
hive> load data inpath '/user/cloudera/dataset/vgsales.csv' overwrite into table
 vgsales;
Loading data to table dataset.vgsales
chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehous
e/dataset.db/vgsales/vgsales.csv': User does not belong to supergroup
Table dataset.vgsales stats: [numFiles=1, numRows=0, totalSize=1355781, rawDataS
ize=0]
OK
```

Step 4: Performing Analysis / Hive Queries

- Displaying some of the data in hive table

```
Time taken: 0.035 seconds
hive> select * from vgsales limit 10;
OK
NULL    Name    Platform        NULL    Genre   Publisher       NULL    NULL   N
ULL     NULL    NULL
1       Wii Sports      Wii     2006.0  Sports  Nintendo        41.49   29.02  3
.77     8.46    82.74
2       Super Mario Bros.       NES     1985.0  Platform        Nintendo       2
9.08    3.58    6.81    0.77    40.24
3       Mario Kart Wii  Wii     2008.0  Racing  Nintendo        15.85   12.88  3
.79     3.31    35.82
4       Wii Sports Resort       Wii     2009.0  Sports  Nintendo        15.75  1
1.01    3.28    2.96    33.0
5       Pokemon Red/Pokemon Blue        GB      1996.0  Role-Playing    Nintendo
11.27   8.89    10.22   1.0     31.37
6       Tetris  GB      1989.0  Puzzle  Nintendo        23.2    2.26    4.22   0
.58     30.26
7       New Super Mario Bros.   DS      2006.0  Platform        Nintendo       1
1.38    9.23    6.5     2.9     30.01
8       Wii Play        Wii     2006.0  Misc    Nintendo        14.03   9.2    2
.93     2.85    29.02
9       New Super Mario Bros. Wii       Wii     2009.0  Platform        Nintendo
14.59   7.06    4.7     2.26    28.62
Time taken: 0.35 seconds, Fetched: 10 row(s)
hive> desc vgsales;
OK
rank                    int
name                    string
platform                string
year                    float
genre                   string
publisher               string
na_sales                float
eu_sales                float
jp_sales                float
other_sales             float
global_sales            float
Time taken: 0.11 seconds, Fetched: 11 row(s)
```

- Getting the total numbers of rows in our hive table.

```
hive> select count(Genre) from vgsales;
Query ID = cloudera_20251023041515_18a53c69-8d20-4303-8ffc-8d29108198b3
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761216649316_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761216649316_000
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761216649316_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-23 04:15:36,857 Stage-1 map = 0%,  reduce = 0%
2025-10-23 04:15:46,991 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.99 sec
2025-10-23 04:15:57,946 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.71 sec
MapReduce Total cumulative CPU time: 3 seconds 710 msec
Ended Job = job_1761216649316_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.71 sec   HDFS Read: 1364411 HDFS Write: 6 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 710 msec
OK
16599
Time taken: 39.179 seconds, Fetched: 1 row(s)
```

- Getting videos games counts per genre.

```
hive> select genre, count(genre) from vgsales group by genre order by 2;
Query ID = cloudera_20251024000404_b05de96c-e3e5-402a-9cda-6ffcd5a659cf
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761216649316_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761216649316_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761216649316_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-24 00:04:24,561 Stage-1 map = 0%,  reduce = 0%
2025-10-24 00:04:34,539 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.04 sec
2025-10-24 00:04:46,492 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.61 sec
MapReduce Total cumulative CPU time: 3 seconds 610 msec
Ended Job = job_1761216649316_0004
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761216649316_0005, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761216649316_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761216649316_0005
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2025-10-24 00:04:57,572 Stage-2 map = 0%,  reduce = 0%
2025-10-24 00:05:06,367 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.19 sec
2025-10-24 00:05:16,143 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 2.91 sec
MapReduce Total cumulative CPU time: 2 seconds 910 msec
Ended Job = job_1761216649316_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.61 sec   HDFS Read: 1363898 HDFS Write: 1224 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 2.91 sec   HDFS Read: 6285 HDFS Write: 401 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 520 msec
OK
XB      1
X360    2
Wii     3
Strategy        674
Sports  2342
Simulation      865
Shooter 1296
Role-Playing    1481
```

- Displaying titles for specific platform and year.
  (helps understand which year has major releases for a specific platform.)

```
hive> select * from vgsales where platform = "PS2" and year = 2010 limit 10;
OK
2711    FIFA Soccer 11  PS2     2010.0  Sports  Electronic Arts 0.11    0.29    0.0     0.36    0.76
3422    Silent Hill: Shattered Memories PS2     2010.0  Action  Konami Digital Entertainment    0.13    0.22    0.01    0.23    0.59
3995    Madden NFL 11   PS2     2010.0  Sports  Electronic Arts 0.41    0.02    0.0     0.07    0.5
4687    pro evolution soccer 2011       PS2     2010.0  Sports  Konami Digital Entertainment    0.04    0.21    0.05    0.11    0.41
4693    MLB 10: The Show        PS2     2010.0  Sports  Sony Computer Entertainment     0.2     0.16    0.0     0.05    0.41
4753    NBA 2K11        PS2     2010.0  Action  Take-Two Interactive    0.34    0.01    0.0     0.06    0.41
5025    NCAA Football 11        PS2     2010.0  Sports  Electronic Arts 0.19    0.15    0.0     0.05    0.38
5072    WWE SmackDown vs. Raw 2011      PS2     2010.0  Fighting        THQ     0.24    0.07    0.0     0.07    0.38
6390    Scooby-Doo! and the Spooky Swamp        PS2     2010.0  Action  Warner Bros. Interactive Entertainment  0.08    0.11    0.0     0.08    0.27
7051    Major League Baseball 2K10      PS2     2010.0  Sports  Take-Two Interactive    0.11    0.09    0.0     0.03    0.23
Time taken: 0.265 seconds, Fetched: 10 row(s)
hive>
```

- Displaying popular genres in our video games sales dataset.
  (helps understand publisher which genre are popular and smartly invest in those genre.)

```
hive> select genre from vgsales where global_sales between 20.0 and 30.0;
OK
Misc
Platform
Shooter
Simulation
Racing
Role-Playing
Sports
Sports
Misc
Action
Action
Platform
Misc
Time taken: 0.132 seconds, Fetched: 13 row(s)
hive>
```

- Displaying genre per publishers.
  (helps understand which genre are popular among a publisher.)

```
hive> select publisher, genre from vgsales group by publisher, genre limit 10;
Query ID = cloudera_20251024001111_f44f2c0d-8d86-47ce-9dd2-8174c33da4ea
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761216649316_0007, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761216649316_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761216649316_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-24 00:11:30,176 Stage-1 map = 0%,  reduce = 0%
2025-10-24 00:11:40,003 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.19 sec
2025-10-24 00:11:50,835 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.01 sec
MapReduce Total cumulative CPU time: 4 seconds 10 msec
Ended Job = job_1761216649316_0007
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.01 sec   HDFS Read: 1364672 HDFS Write: 279 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 10 msec
OK
"Destination Software   Action
"Destination Software   Misc
"Destination Software   Puzzle
"Destination Software   Racing
"Destination Software   Role-Playing
"Destination Software   Simulation
"Destination Software   Sports
"Interworks Unlimited   Sports
"mixi   Action
10TACLE Studios Adventure
Time taken: 31.66 seconds, Fetched: 10 row(s)
hive>
```

- Displaying average sales in a specific region for a genre.

  (helps understand which region to target for maximum sales for a specific genre.)

```
hive> select avg(na_sales) from vgsales where genre = "Action";
Query ID = cloudera_20251024035555_474ad9b9-713c-4ee5-a7e2-53106614ed02
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761216649316_0009, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761216649316_0009/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761216649316_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-24 03:55:46,688 Stage-1 map = 0%,  reduce = 0%
2025-10-24 03:55:56,572 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.33 sec
2025-10-24 03:56:06,322 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.07 sec
MapReduce Total cumulative CPU time: 4 seconds 70 msec
Ended Job = job_1761216649316_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.07 sec   HDFS Read: 1366005 HDFS Write: 19 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 70 msec
OK
0.2653355603449945
Time taken: 31.135 seconds, Fetched: 1 row(s)
hive>
```

- Displaying total sales of each genre for a region.
  (helps understand which genres are more popular in a specific region.)

```
hive> select genre, sum(jp_sales) from vgsales group by genre, 2 order by 2 desc limit 10;
Query ID = cloudera_20251024040101_f9009366-1d00-4609-a085-11a3e34a563d
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761216649316_0011, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761216649316_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761216649316_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-24 04:01:26,041 Stage-1 map = 0%,  reduce = 0%
2025-10-24 04:01:35,859 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.18 sec
2025-10-24 04:01:44,510 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.74 sec
MapReduce Total cumulative CPU time: 3 seconds 740 msec
Ended Job = job_1761216649316_0011
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761216649316_0012, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761216649316_0012/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761216649316_0012
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2025-10-24 04:01:56,055 Stage-2 map = 0%,  reduce = 0%
2025-10-24 04:02:04,787 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 1.13 sec
2025-10-24 04:02:15,594 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 2.86 sec
MapReduce Total cumulative CPU time: 2 seconds 860 msec
Ended Job = job_1761216649316_0012
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.74 sec   HDFS Read: 1364571 HDFS Write: 1498 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 2.86 sec   HDFS Read: 6525 HDFS Write: 260 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 600 msec
OK
XB      0.1899999976158142
X360    0.7100000083446503
Wii     0.4299999997019768
Strategy        49.46000000089407
Sports  135.29999935626984
Simulation      62.660000152885914
Shooter 38.27999994531274
Role-Playing    351.6299996897578
Racing  56.70000000298023
```

- Displaying average sales of a publisher for a region.
  (helps understand a publisher which region to target for maximum sales.)

```
hive> select avg(eu_sales) from vgsales where publisher = "Electronic Arts";
Query ID = cloudera_20251024040404_1973c720-7a0b-450d-839d-e1d1b2b42c61
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1761216649316_0013, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1761216649316_0013/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1761216649316_0013
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-10-24 04:04:47,523 Stage-1 map = 0%,  reduce = 0%
2025-10-24 04:04:56,234 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.2 sec
2025-10-24 04:05:08,102 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.91 sec
MapReduce Total cumulative CPU time: 3 seconds 910 msec
Ended Job = job_1761216649316_0013
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.91 sec   HDFS Read: 1366043 HDFS Write: 20 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 910 msec
OK
0.27481125094834785
Time taken: 31.938 seconds, Fetched: 1 row(s)
hive>
```

Data Visualization

Using Python (Matplotlib / Seaborn):

```
[1]: import pandas as pd
```

```
[2]: import matplotlib.pyplot as plt
     import seaborn as sns
```

```
[31]: sns.set_palette("pastel")
```

```
[3]: df = pd.read_csv(r'C:\Users\Vivek\Desktop\vgsales.csv')
```

```
[4]: df.head()
```

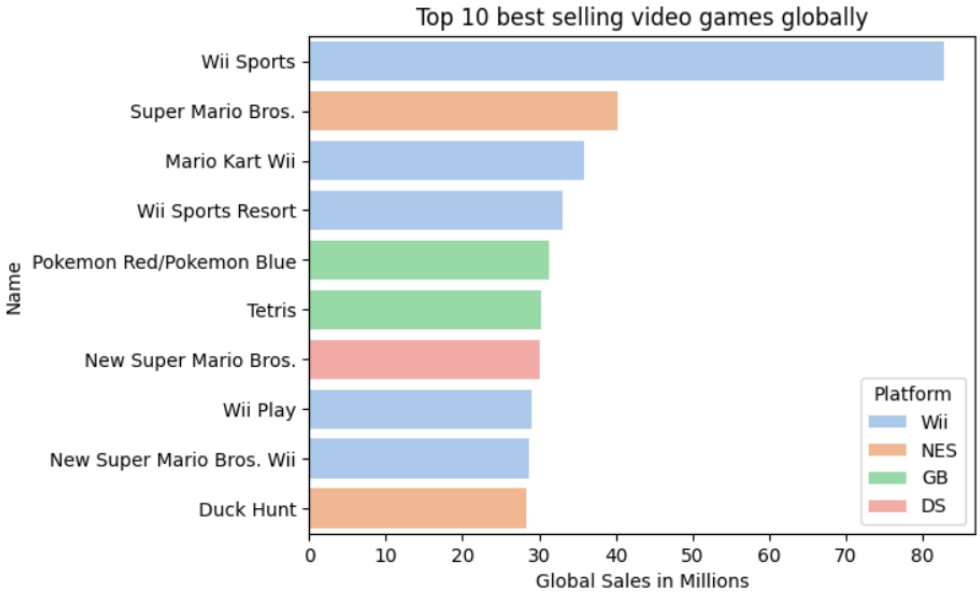
[4]:

	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

Top 10 best selling video games globally

```
[7]: top_10_Games_Globally = df.sort_values(by='Global_Sales', ascending=False).head(10)
```

```
[32]: sns.barplot(data=top_10_Games_Globally, x='Global_Sales', y='Name', hue='Platform', dodge=False)
      plt.title("Top 10 best selling video games globally")
      plt.xlabel("Global Sales in Millions")
      plt.ylabel("Name")
      plt.show()
```



Distribution of games released across the years

```
[13]: year_dist = df['Year'].value_counts()
```

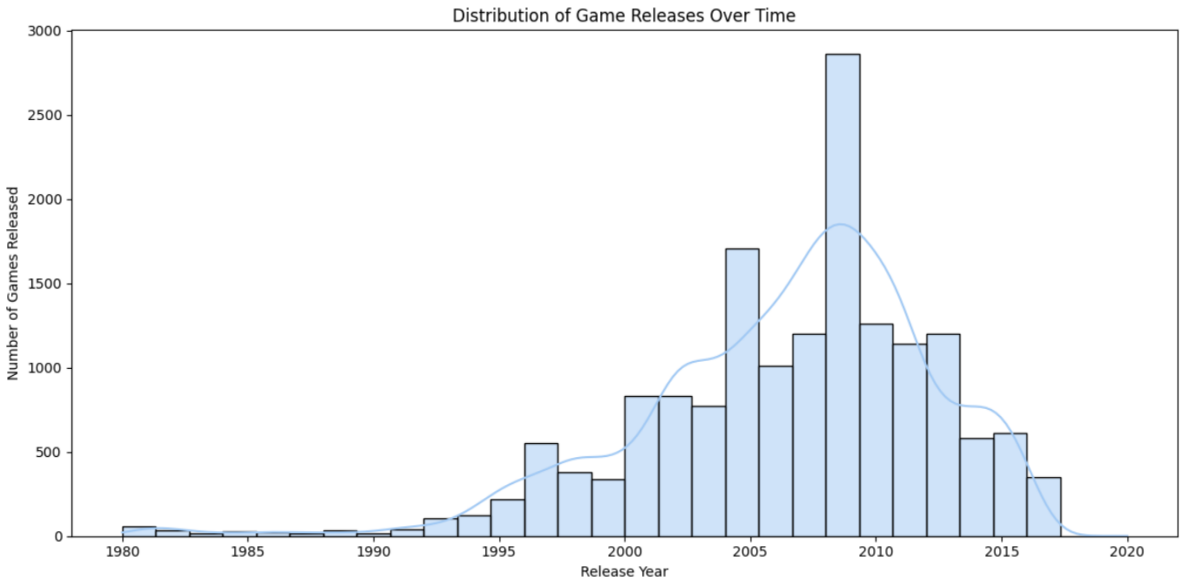
```
[21]: year_dist.head()
```

[13]:
```
year_dist = df['Year'].value_counts()
```

[21]:
```
year_dist.head()
```

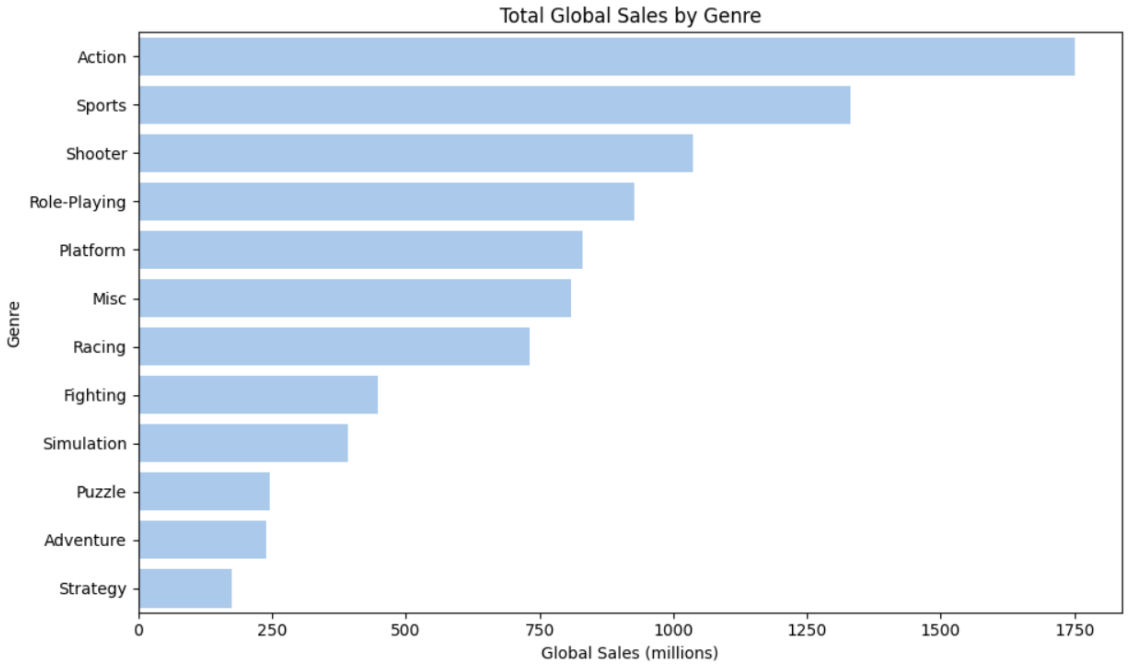
[21]:
```
Year
2009.0    1431
2008.0    1428
2010.0    1259
2007.0    1202
2011.0    1139
Name: count, dtype: int64
```

[33]:
```
plt.figure(figsize=(12, 6))
sns.histplot(df['Year'].dropna(), bins=30, kde=True)
plt.title("Distribution of Game Releases Over Time")
plt.xlabel("Release Year")
plt.ylabel("Number of Games Released")
plt.tight_layout()
plt.show()
```



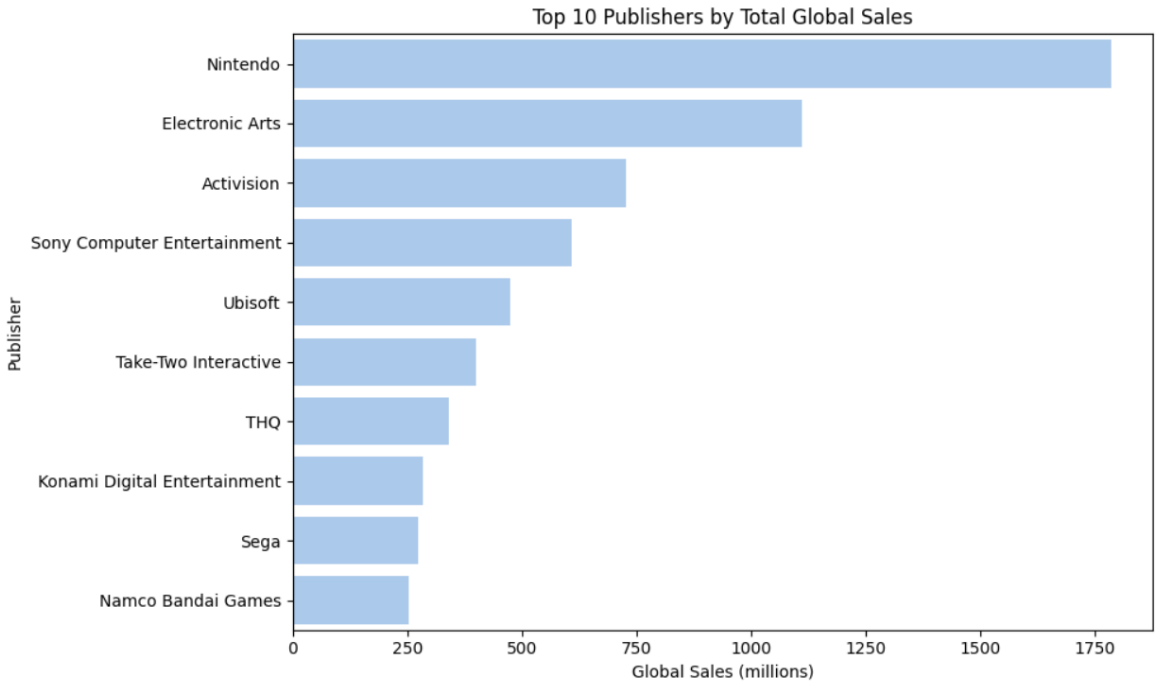
Total Sales by Genre

[37]:
```
genre_sales = df.groupby("Genre")["Global_Sales"].sum().sort_values(ascending=False)
plt.figure(figsize=(10, 6))
sns.barplot(x=genre_sales.values, y=genre_sales.index)
plt.title("Total Global Sales by Genre")
plt.xlabel("Global Sales (millions)")
plt.ylabel("Genre")
plt.tight_layout()
plt.show()
```



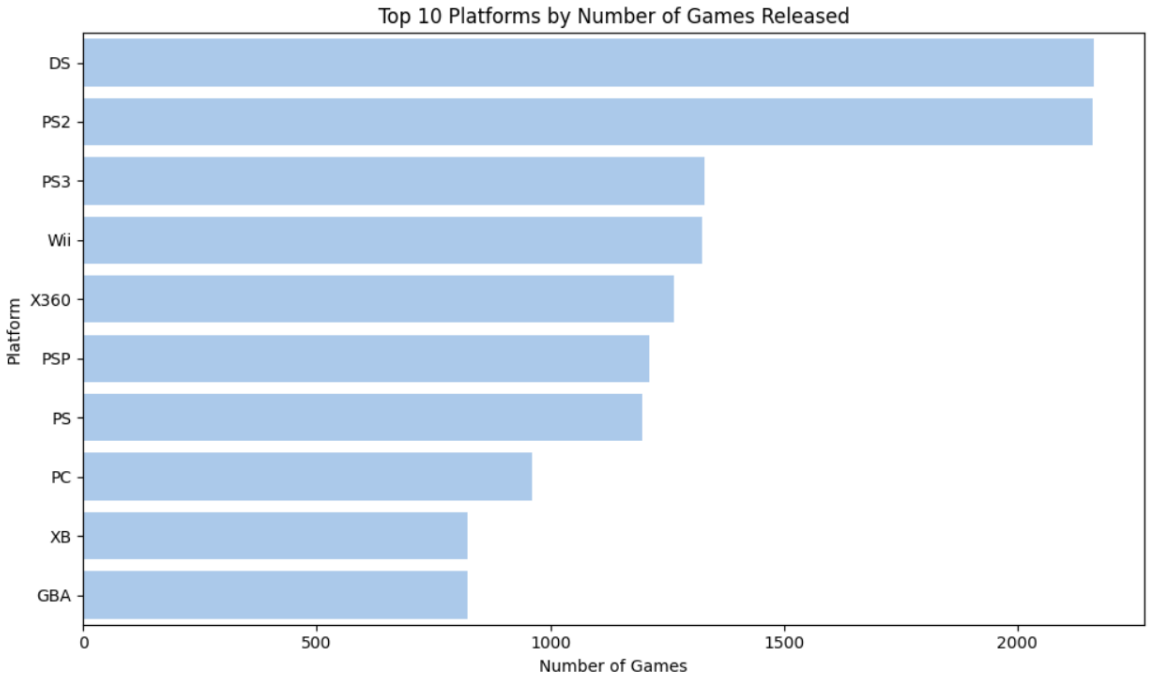
Top 10 Publishers by Total Global Sales

[42]:
```
publisher_sales = df.groupby("Publisher")["Global_Sales"].sum().nlargest(10)
plt.figure(figsize=(10, 6))
sns.barplot(x=publisher_sales.values, y=publisher_sales.index)
plt.title("Top 10 Publishers by Total Global Sales")
plt.xlabel("Global Sales (millions)")
plt.ylabel("Publisher")
plt.tight_layout()
plt.show()
```



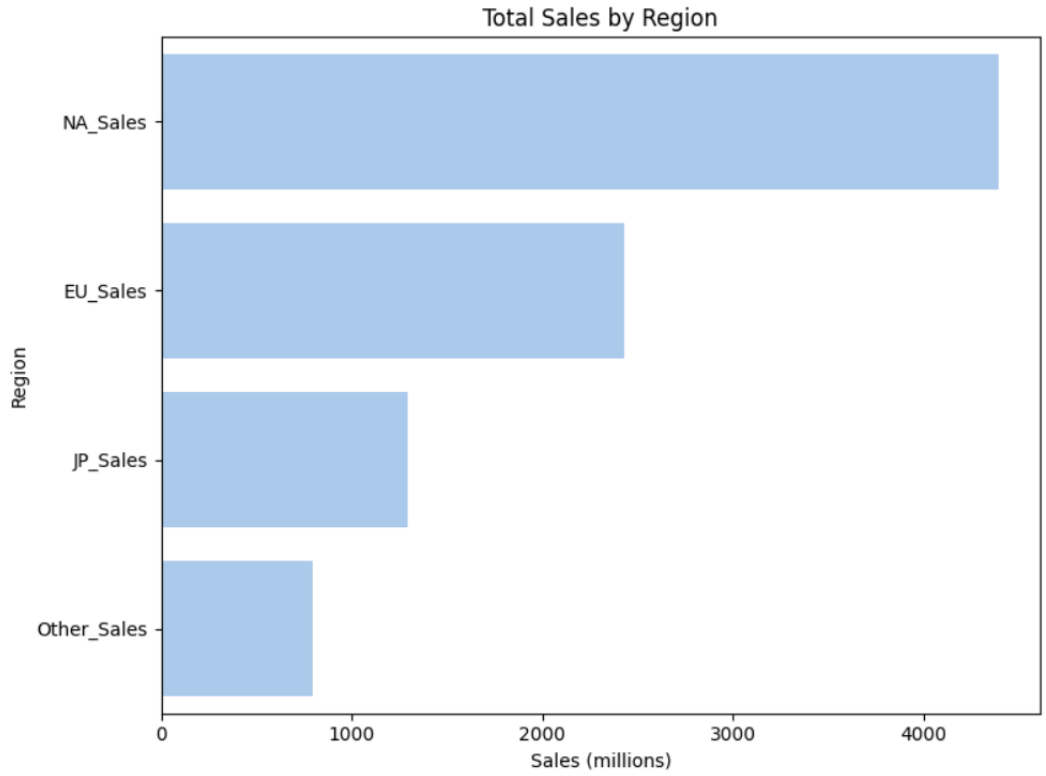
Top 10 platforms by numbers of games released

[35]:
```
platform_counts = df["Platform"].value_counts().head(10)
plt.figure(figsize=(10, 6))
sns.barplot(x=platform_counts.values, y=platform_counts.index)
plt.title("Top 10 Platforms by Number of Games Released")
plt.xlabel("Number of Games")
plt.ylabel("Platform")
plt.tight_layout()
plt.show()
```



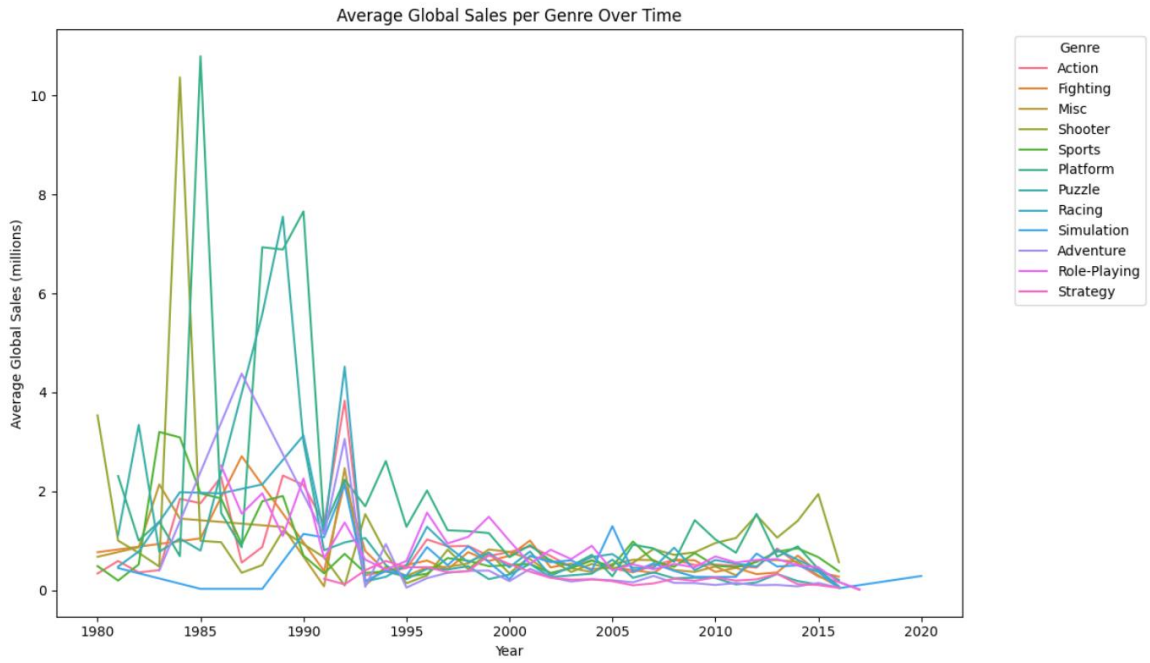
Total Sales by Region

[38]:
```
regions = ["NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales"]
region_sums = df[regions].sum().sort_values(ascending=False)
plt.figure(figsize=(8, 6))
sns.barplot(x=region_sums.values, y=region_sums.index)
plt.title("Total Sales by Region")
plt.xlabel("Sales (millions)")
plt.ylabel("Region")
plt.tight_layout()
plt.show()
```



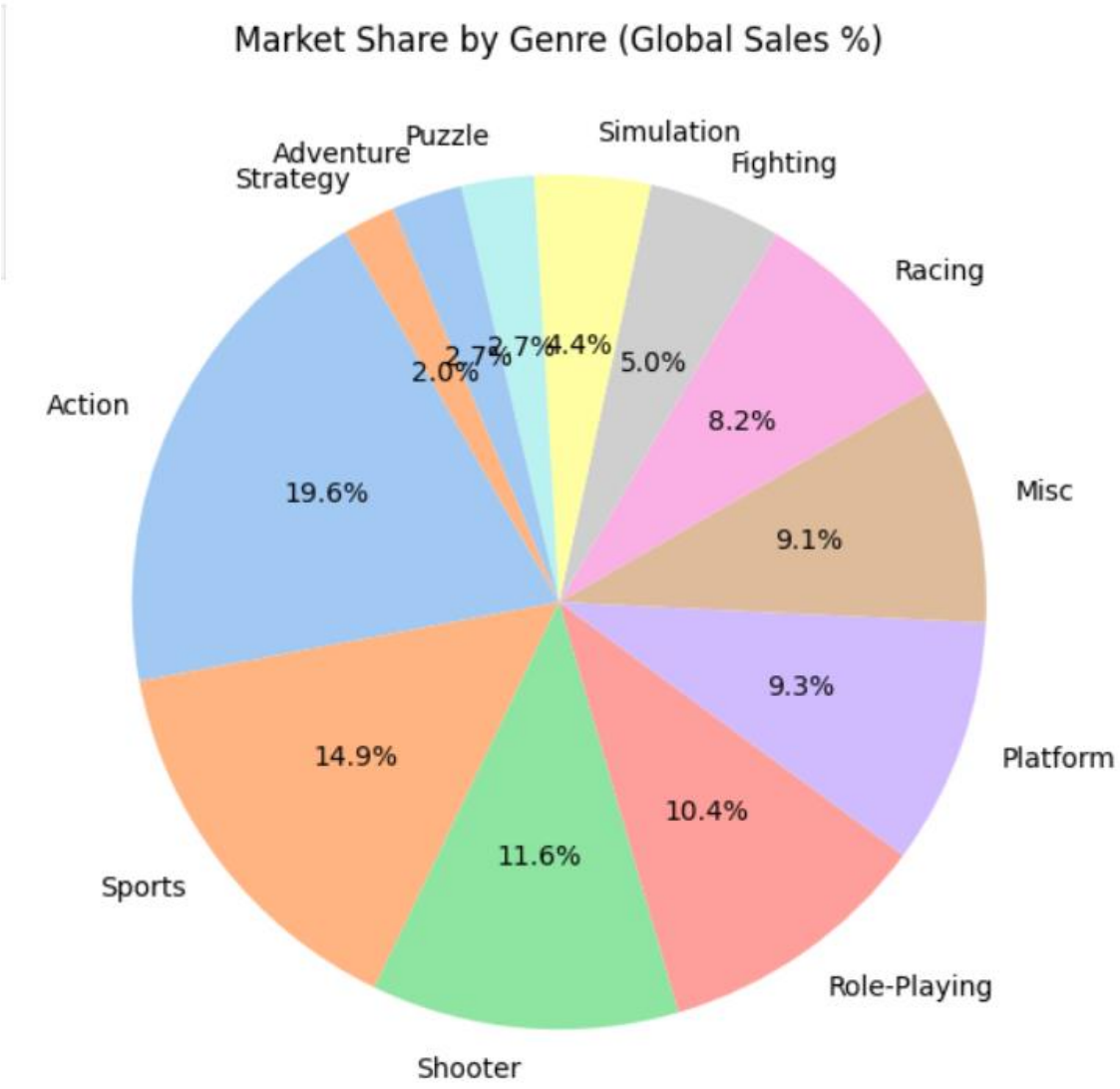
Genre Trends Over Time (Average Global Sales)

[40]:
```
genre_year = df.dropna(subset=['Year']).groupby(['Year', 'Genre'])['Global_Sales'].mean().reset_index()
plt.figure(figsize=(12, 7))
sns.lineplot(data=genre_year, x='Year', y='Global_Sales', hue='Genre')
plt.title("Average Global Sales per Genre Over Time")
plt.xlabel("Year")
plt.ylabel("Average Global Sales (millions)")
plt.legend(title="Genre", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```



Market Share by Genre

[17]:
```
plt.figure(figsize=(6, 6))
genre_sales_pct = genre_sales / genre_sales.sum()
plt.pie(genre_sales_pct, labels=genre_sales_pct.index, autopct='%1.1f%%', startangle=120)
plt.title("Market Share by Genre (Global Sales %)")
plt.tight_layout()
plt.show()
```


Market Share by Genre (Global Sales %)
Puzzle
Simulation
Adventure
Strategy
Fighting
Racing
Action
2.0%
2.7%
2.7%
4.4%
5.0%
8.2%
19.6%
Misc
9.1%
9.3%
Platform
14.9%
10.4%
11.6%
Sports
Role-Playing
Shooter

Conclusion

This case study showcases how Hadoop ecosystem components — **HDFS**, and **Hive** — can be used to process and analyze structured data at scale. Compared to traditional systems like MySQL, **Hive** allows distributed querying, fault tolerance, and better scalability for large datasets. Through visualization, we can derive meaningful insights that support business decisions in the gaming industry.