CS 839 - Spring 839 - Project Stage 4

Team - Abhinav Garg, Bidyut Hota, Chirayu Garg

Table Merging

We used the two table from Project Stage 3 to form the merged table. We did not use any other table. Merge strategy for each of the attribute are detailed in following table.

Attribute	**Merge Process**
ltable_ID	This is primary index from table A used for entity matching, this attribute is primarily kept to ensure that data lineage is maintained
rtable_ID	This is primary index from table B used for entity matching, this attribute is primarily kept to ensure that data lineage is maintained
Title	The game title that was longer among the matched tuples was taken, this was done with assumption that longer title will have more information
Developer	Union of developer attribute among the matched entities is taken, this attribute can have multiple values and hence they are separated using a '|' character
Publisher	Similar to developer attribute, a union of values from matched entities is used
Platform	Platform attribute from matched entities are first normalized to a base and then union is taken to find all platforms. An example of normalization is: PlayStation 1, PlayStation 2, PlayStation 3 and other variants are replaced by PlayStation
Genre	Union of genre values from matched tuples is taken
ReleaseDate	Preference is given to table A (left table) for release data, if missing then data is taken from table B (right table)
lRating	Rating of game title from left table, rating values are not merged to help later in data analysis
rRating	Rating of game title from right table, rating values are not merged to help later in data analysis

Problems faced: We had difficulty deciding how to combine Ratings from two tables A and B. We decided to keep them both and analyse if there are any differences in the way Games are rated on two sources.

Table E Overview

- Schema: 'ltable_ID', 'rtable_ID', 'Title', 'Developer', 'Publisher', 'Platform', 'Genre', 'ReleaseDate', 'lRating', 'rRating'

- Number of tuples: 3959

- Tuple Sample:

  a6418, b6090, Shining Force EXA, Neverland, Sega, PlayStation, Action|role-playing|Role-Playing, 2007-03-20,8.1,6.9

  a6427, b6098, Shining the Holy Ark, Camelot Software Planning|Sonic! Software Planning, Sega, Sega, Role-Playing,1997-06-30,8.6,7.7

  a4165, b4147, Mario Party 4, Hudson Soft|Hudson, Nintendo, Nintendo, Party/Minigame|Party, 2002-10-21,7.7,7.4

  a3466,b3467,International Superstar Soccer 64,Konami Computer Entertainment Osaka,Konami,Nintendo,Team-Based|Sports|Simulation|Soccer,1997-07-31,8.2,8.0

Data Analysis

We have used OLAP style exploration to understand the distribution of game titles by platform, publisher and release date. We present some of our learnings here.

Rating Analysis

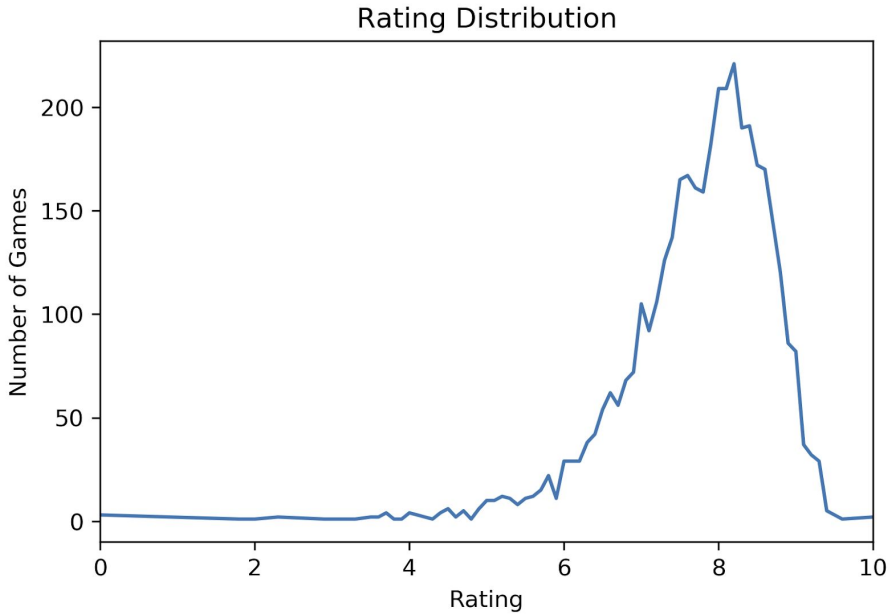

Figure 1: Rating Distribution of Games

We attempted to use ratings from the two data sources to do some analysis, however there is no significant information as both sources have mean rating around 7.5 with a standard deviation of less than 1. Figure 1 shows the rating distribution of games for one of the source (IRating).

Yearly and Monthly Game Release Analysis

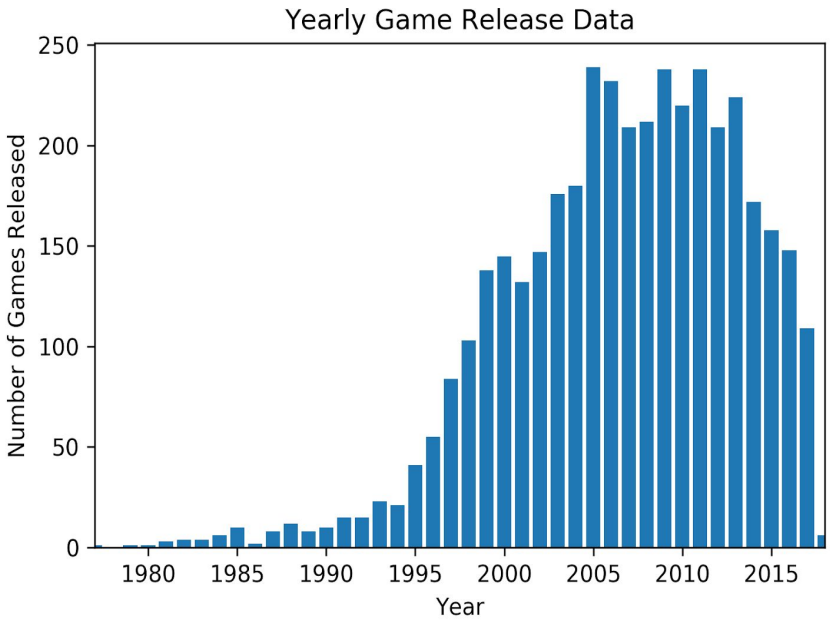

Figure 2: Yearly Game Release Data

Using roll-up operations on the release date for the tuples in table E, it is observed that year-over-year number of games released has increased exponentially from 90's to 2000's. This concurs with the common knowledge that video game popularity has increased significantly over last couple of decades. Figure 2, illustrates above information.

Using drill-down and slice operations on above information, monthly game release information is extracted, as shown in Figure 3 . A very interesting insight here is that for first three financial quarters the highest number of video games are released for the last month of the quarter. This may be primarily done, to show strong earnings forecast for upcoming quarters in the financial report. For last financial quarter, the releases are higher for initial months of October and November to capitalize on holiday market of December.

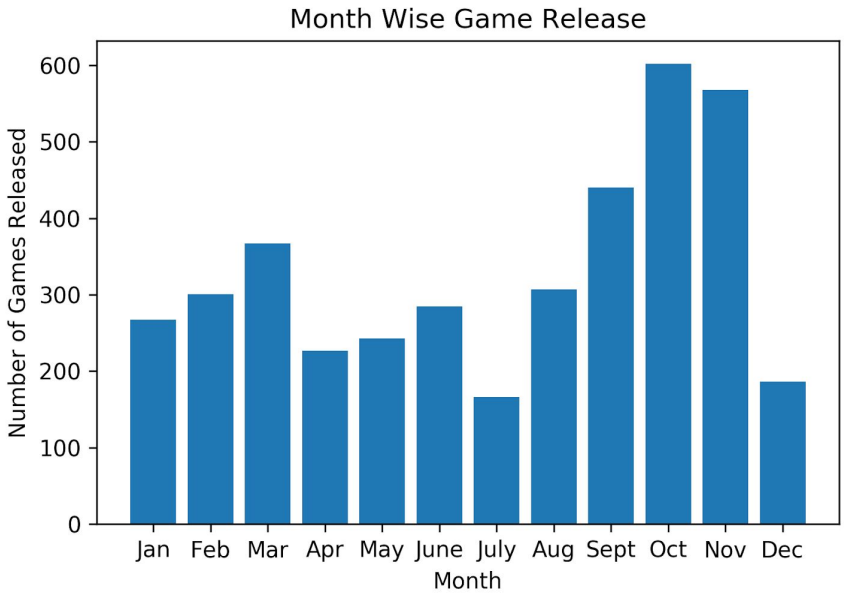

Figure 3: Monthly Game Release Distribution

Platform Popularity Analysis

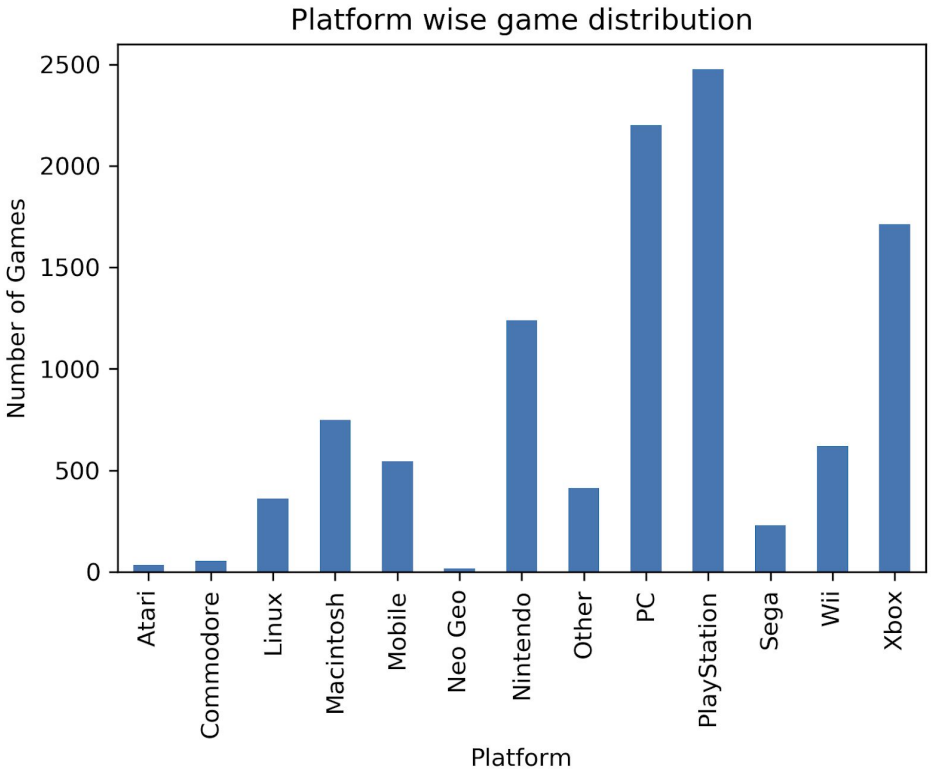

Figure 4: Platform wise game distribution

Platform wise game distribution is extracted using roll-up and slice operations, as shown in Figure 4. PlayStation, PC, Xbox, and Nintendo emerge as the most popular platforms for which games are released.

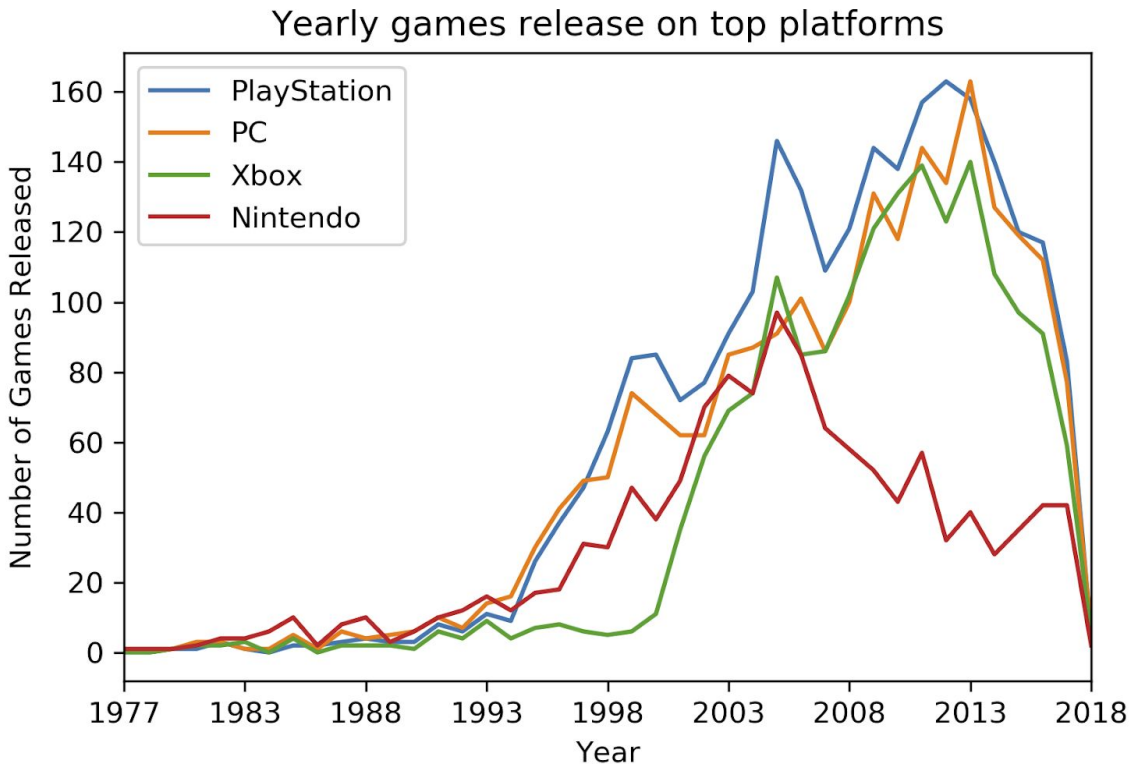

Figure 5: Year wise games released on top platforms

From the previous analysis, we know that 4 most popular platforms are PlayStation, PC, Xbox, and Nintendo. We used the slice and roll-up operations to get the games released on these 4 popular platforms. The result is shown in Figure 5. We observe that these platforms became popular around year 1995, which is also the time around which gaming became popular in general (Figure 2).

Interestingly, while the PlayStation, PC, and Nintendo started becoming popular around 1995, Xbox gained popularity around 2001. We also observe that Nintendo's popularity is declining from year 2005. Also the number of games released in year 2005 for PlayStation, PC, and Xbox does not follow the year-over-year trend, and we see a much higher number of games released in 2005 than expected by following the trend.

Genre popularity Analysis

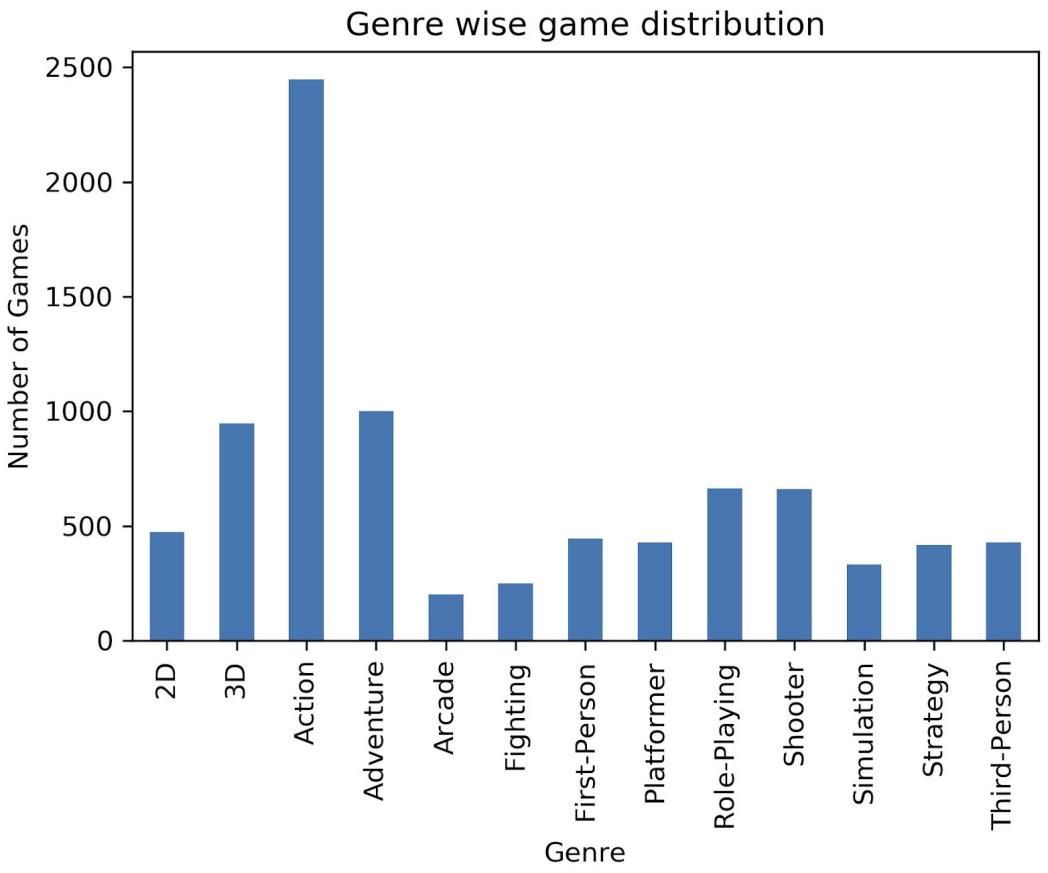

Figure 6: Genre wise game distribution

Extracting Genre wise game distribution using roll-up and slice operations, Action emerge as the most popular Genre of the released games. Figure 6 shows that a much higher number of Acton games are released as compared to other platforms.

Top 10 publishers

Here we find the 10 Publishers who have produced maximum number of games with high rating since year 2000. A game is considered to have high rating if the rating of game is equal or above 9.0. For this operation, used the dice operation for ReleaseDate and IRating attributes (ReleaseDate>=2000 and IRating>=9) and then aggregated using Publisher attribute. The is result is shown in Table 1. It is interesting to note that Nintendo(45 games) has produced much higher number of high rated games, as compared to its closest competitor SCEA(20 games).

Publisher	Number of High Rated Games
Nintendo	45
SCEA	20
Square Enix	19
SCEE	18
Capcom	16
SCE Australia	15
SCEI	14
Konami	12
Sony Computer Entertainment	12
Microsoft Game Studios	11

Table 1 : Publishers with highest number of high rated games produced

Future Work

Given more time we would like to spend some time on unifying certain Genre as some are very closely related. Similarly there are certain game development studios and publishers that can be combined with their parent companies and we will have more concise data for analysis.

We can get the sales data for games from an external source, and explore how the sales data correlates with platform, publisher, and release month. It will be interesting to compare the earnings of game released in months of October, November with earnings of games released in other months.

Python script to merge table

mergeTables

May 9, 2018

In [42]:
```
import pandas as pd
import numpy as np
```

In [43]:
```
df = pd.read_csv("/Users/abhinavgarg/Projects/Data-Science/stage4/data/matchedTuples.
```

In [44]:
```
platformDict = {"PC":"PC", "Xbox 360":"Xbox", "Xbox One":"Xbox", "Xbox":"Xbox",
"PlayStation 4":"PlayStation","PlayStation 3":"PlayStation",
"PlayStation 2":"PlayStation","PlayStation":"PlayStation","Wii":"Wii",
"Macintosh":"Macintosh","PSP":"PlayStation","Game Boy Advance":"Nintendo",
"PlayStation Vita":"PlayStation","Linux":"Linux","DS":"Nintendo","Android":"Mobile",
"iOS (iPhone/iPad)":"Mobile","GameCube":"Nintendo","Wii U":"Wii","Mac":"Macintosh",
"Nintendo 3DS":"Nintendo","Nintendo 64":"Nintendo", "Super Nintendo":"Nintendo",
"Nintendo DS":"Nintendo", "NES":"Nintendo","Nintendo Switch":"Nintendo",
"Nintendo GameCube":"Nintendo","Dreamcast":"Sega","Mobile":"Mobile",
"Game Boy Color":"Nintendo","iOS":"Mobile","3DS":"Nintendo","Game Boy":"Nintendo",
"Sega Mega Drive/Genesis":"Sega", "Saturn":"Sega","Genesis":"Sega",
"Windows Mobile":"Mobile","Sega Master System":"Sega","Sega Saturn":"Sega",
"Nintendo 2DS":"Nintendo","PlayStation VR":"PlayStation","Commodore 64":"Commodore",
"Amiga":"Commodore","BlackBerry":"Mobile","N-Gage":"Mobile","Sega CD":"Sega",
"PlayStation Now":"PlayStation","Sega Game Gear":"Sega","Sega 32X":"Sega",
"Atari 2600":"Atari","Atari ST":"Atari", "Atari 8-bit":"Atari", "Atari 7800":"Atari",
"Atari 5200":"Atari","Atari Jaguar":"Atari", "Atari Lynx":"Atari","NeoGeo":"Neo Geo",
"NeoGeo Pocket Color":"Neo Geo","Neo Geo":"Neo Geo", "Neo Geo Pocket Color":"Neo Geo"
"Neo Geo CD":"Neo Geo","Apple II":"Macintosh","Neo-Geo CD":"Neo Geo",
"Windows Mobile":"Mobile","HTC Vive":"Mobile"
}
```

In [45]:
```
def getMergedPlatform(lplatform,rplatform):
    mergedPlatString = lplatform+"|"+rplatform
    uniquePlatforms = set()
    for it in mergedPlatString.split('|'):
        uniquePlatforms.add(platformDict.get(it,"Other"))
    return '|'.join(list(uniquePlatforms))

def getLargerString(lString,rString):
    if len(lString) > len(rString):
        return lString
    return rString
```

```
        def getMergedSet(lplatform,rplatform):
            mergedPlatString = lplatform+"|"+rplatform
            return '|'.join(list(set(mergedPlatString.split('|'))))

        # def getCommonDate(ldate,rdate):
```

In [47]:
```
        def checkNullValue(left,right):
            if str(left) != "nan" and str(right) != "nan":
                return 0
            if str(left) != "nan" and str(right) == "nan":
                return 1
            if str(left) == "nan" and str(right) != "nan":
                return 2
            return 3

        def getMergedValue(left,right,parameter_type=None):
            ret = checkNullValue(left,right)
            if ret == 3:
                return ""
            if ret == 2:
                return right
            if ret == 1:
                return left
            if parameter_type == 'title':
                return getLargerString(left,right)
            if parameter_type == 'platform':
                return getMergedPlatform(left,right)
            return getMergedSet(left,right)
```

In [56]:
```
        dfMerged = pd.DataFrame()
        dfMerged['ltable_ID'] = pd.Series(dtype=str)
        dfMerged['rtable_ID'] = pd.Series(dtype=str)
        dfMerged['Title'] = pd.Series(dtype=str)
        dfMerged['Developer'] = pd.Series(dtype=str)
        dfMerged['Publisher'] = pd.Series(dtype=str)
        dfMerged['Platform'] = pd.Series(dtype=str)
        dfMerged['Genre'] = pd.Series(dtype=str)
        dfMerged['ReleaseDate'] = pd.Series(dtype=str)
        dfMerged['lRating'] = pd.Series(dtype=float)
        dfMerged['rRating'] = pd.Series(dtype=float)

        for idx, row in df.iterrows():
        #     print(row)
            title = getMergedValue(row['ltable_Title'], row['rtable_Title'],'title')
```

```
            developer = getMergedValue(row['ltable_Developer'], row['rtable_Developer'])
            publisher = getMergedValue(row['ltable_Publisher'], row['rtable_Publisher'])
            platform = getMergedValue(row['ltable_Platform'],
                                      row['rtable_Platform'],'platform')
            genre = getMergedValue(row['ltable_Genre'], row['rtable_Genre'])

            relDateRet = checkNullValue(row['ltable_CleanRDate'],row['rtable_CleanRDate'])
            releasedate = ""
            if relDateRet == 0:
                releasedate = row['ltable_CleanRDate']
            elif relDateRet == 1:
                releasedate = row['ltable_CleanRDate']
            elif relDateRet == 2:
                releasedate = row['rtable_CleanRDate']
            dfMerged.loc[idx] = [row['ltable_ID'],row['rtable_ID'],title,developer,publisher,
                                 platform,genre,releasedate,row['ltable_Rating'],
                                 row['rtable_Rating']]

        #     row[ltable_ID],row[rtable_ID],row[ltable_ID]
```

In [59]:
```
dfMerged.head()
dfMerged.to_csv("/Users/abhinavgarg/Projects/Data-Science/stage4/data/tableE.csv",
                index=False)
```