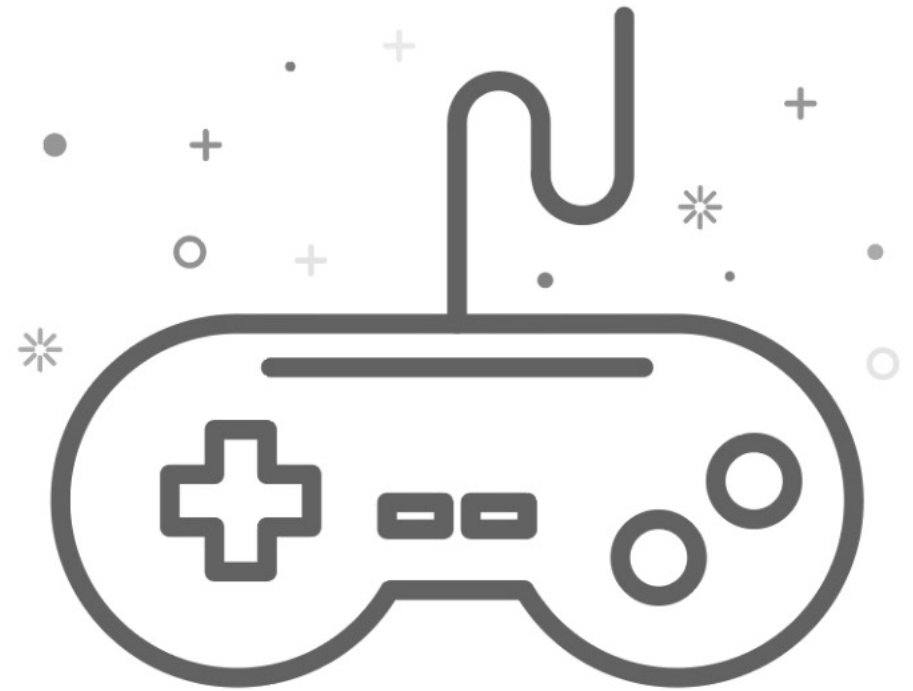


PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

CODE_STATES

AI_14_YEARIM.CHO



PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

CODE_STATES

AI_14_YEARIM.CHO

INDEX



1 기획 배경

1-1 가설

2 DATA DESCRIPTION

3 EDA

4 DATA WRANGLING

- 지역에 따라 선호하는 게임의 장르가 있는가
- 연도별 게임 트렌드가 있는가
- 출고량이 높은 게임에 대한 분석 및 시각화 프로세스

5 가설 검증 및 결과 확인

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

1 기획 배경

게임은 즐거운 것입니다.

게임은 경쟁입니다.

즐겁지 않으면 왜 경쟁을 하며, 경쟁이 없으면 재미도 없습니다.

즐거움과 경쟁은 언제나 붙어있는 것입니다.

하지만, 게임은 또 다른 무언가 이기도 합니다.

그것은 여행의 새로운 세상을 향한 대여정의 입구입니다.

정신을 집중하고, 마음을 열어보세요.

“닌텐도 아메리카 사장 레지 피서메이. E3 닌텐도 스포트라이트 서두 발언 중



PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

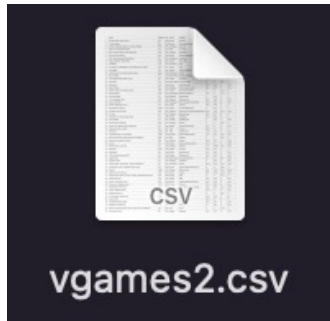
1 기획 배경



PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

1 기획 배경



오늘, 우리에게 제공된 데이터는 ' **vgame2.csv** ' 입니다.

우리는 이 데이터를 사용하여 "다음_분기에_어떤_게임을_설계해야_할까"에 대해 답변하고자 합니다.

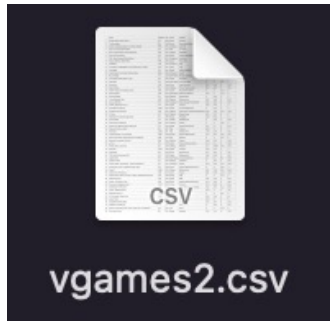
이에, 우리는 다음과 같은 하위 질문을 계획할 수 있습니다.

- 지역에 따라 선호하는 게임의 장르가 있는가
- 연도별 게임 트렌드가 있는가
- 출고량이 높은 게임에 대한 분석 및 시각화 프로세스

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

1-1 가설



우리는 “다음_분기에_어떤_게임을_설계해야_할까 ” 에 답변하기 위한 탐색을 시작하기 전,
새로운 가설을 생성할 수 있습니다.

- 귀무 가설 : 현재 게임 외 새로운 게임을 설계하지 않아도 된다.
- 대립 가설 : 현재 게임 외 새로운 어떤 게임을 설계할 필요가 있다.

또한, 우리는 분석을 시작하기 전, 우리에게 유의미한 결과를 낼 수 있는 **feather** 을 확인 할 필요가 있습니다.

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

2 DATA DESCRIPTION



Data Description

- **Name** : 게임의 이름입니다.
- **Platform** : 게임이 지원되는 플랫폼의 이름입니다.
- **Year** : 게임이 출시된 연도입니다.
- **Genre** : 게임의 장르입니다.
- **Publisher** : 게임을 제작한 회사입니다.
- **NA_Sales** : 북미지역에서의 출고량입니다.
- **EU_Sales** : 유럽지역에서의 출고량입니다.
- **JP_Sales** : 일본지역에서의 출고량입니다.
- **Other_Sales** : 기타지역에서의 출고량입니다.

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

2 DATA DESCRIPTION

- **NA_Sales** : 북미지역에서의 출고량입니다.
- **EU_Sales** : 유럽지역에서의 출고량입니다.
- **JP_Sales** : 일본지역에서의 출고량입니다.
- **Other_Sales** : 기타지역에서의 출고량입니다.

데이터를 확인시, 현 데이터는 우리나라가 아닌 특정 글로벌 지역을 대상으로 수집된 **data set**임을 확인할 수 있습니다.
글로벌을 대상으로 하는 **data set**은 다소 생소한 게임데이터들이 확인될 가능성이 있으나,

큰 범주로 보았을 때는 출고량과 플랫폼, 장르는 유의미한 결과에 영향을 줄 것이라는 생각이 듭니다.

이에, 데이터를 정제하는 과정인 **EDA**와 **DATA WRANGLING**을 통해
필요한 데이터만을 잘 정제하여 결론에 도달하고자 합니다.

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

3 EDA

```
# 라이브러리 불러오기
import numpy as np # 노메딕 컴퓨테이션?
import pandas as pd # 데이터 프레임
import matplotlib.pyplot as plt # 시각화
import seaborn as sns # 시각화

import sklearn #다양한 머신러닝 알고리즘
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

EDA란, 데이터를 분석하기 위하여 다각도로 관찰하며, 이해, 정제하는 단계입니다.

EDA를 거치지 않은 데이터는 많은 결측치와 **missing values** 등의 많은 오류값을 포함하고 있기때문에

EDA과정은 데이터 분석에서 꼭 중요한 요소 입니다.

EDA과정을 위해 필요한 라이브러리와 파일을 불러옵니다.

```
# 파일 불러오기 (local에서 업로드)
from google.colab import files
files.upload()
```


PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

3 EDA

```
game.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Unnamed: 0      16598 non-null  int64
1   Name            16598 non-null  object
2   Platform        16598 non-null  object
3   Year            16327 non-null  float64
4   Genre           16548 non-null  object
5   Publisher       16540 non-null  object
6   NA_Sales        16598 non-null  object
7   EU_Sales        16598 non-null  object
8   JP_Sales        16598 non-null  object
9   Other_Sales     16598 non-null  object
dtypes: float64(1), int64(1), object(8)
memory usage: 1.3+ MB
```



3	Year	16241 non-null	float64
4	Genre	16241 non-null	object
5	Publisher	16241 non-null	object

```
# 중복 데이터 확인 > 중복 데이터가 없다고 뜬다
```

```
game.duplicated().sum()
```

```
#결측값 확인
```

```
game.isnull().sum()
```

```
game.dropna(axis = 0, inplace=True) # 결측값 제거, inplace를 쓰면 변수 없이 저장 가능
```

```
game.reset_index() # index 재정렬
```

```
game.info()
```

- .info() : 데이터의 정보를 보여줍니다. 현재, 결측치가 존재함을 알 수 있습니다.

- .duplicated().sum() : 중복값을 확인합니다. 해당 데이터는 중복 데이터가 없습니다.

- .isnull().sum() : 결측값을 확인합니다.

- .dropna() : 결측값을 삭제합니다.

PROJECT1) PRESENTATION

Goal_다음_분기에_어떤_게임을_설계해야_할까?

3 EDA

NA_Sales	EU_Sales	JP_Sales	Other_Sales
0.02	0	0	0
0.01	0.01	0	0
0.04	0.01	0	0
0	0	0.02	0
0.02	0	0	0
0.11	0.08	0.35	0.03
0	0	0.08	0
0	0	0.16	0
0.34	0.09	0	0.01
0	0	0.03	0
0.65	0.14	0.21	0
0	0	0.12	0.01
0.08	OK	0	0.01M



0.08	OK	0	0.01M
0.25	0.2	0.18	0.06
0	0	0.04	0
0	0	0.04	0
1.22	0	0M	270K

- `sample()` : 무작위로 **data set**을 확인 할 수 있습니다.

- `head()` : 데이터를 상위 순으로 나열합니다.

현재 **sales** 데이터를 확인시, 'K','M' 같은 단위 데이터를 확인할 수 있습니다.

'K'는 10의 세제곱 단위를, 'M'은 10의 여섯 제곱을 의미하며, 현 데이터가 소수점인 것을 보아

Sales 단위는 'M'을 기준으로 작성 됨을 알 수 있습니다.

PROJECT1) PRESENTATION


Goal _다음_분기에_어떤_게임을_설계해야_할까?

3 EDA

#데이터 손실 방지 위해 카피

```
game1 = game.copy()
```

```
game1['NA_Sales']=game1['NA_Sales'].replace({"K": "*1e-3", "M": "*1"}, regex=True).map(pd.eval).astype(float)
game1['EU_Sales']=game1['EU_Sales'].replace({"K": "*1e-3", "M": "*1"}, regex=True).map(pd.eval).astype(float)
game1['JP_Sales']=game1['JP_Sales'].replace({"K": "*1e-3", "M": "*1"}, regex=True).map(pd.eval).astype(float)
game1['Other_Sales']=game1['Other_Sales'].replace({"K": "*1e-3", "M": "*1"}, regex=True).map(pd.eval).astype(float)
```



6	NA_Sales	16141	non-null	float64
7	EU_Sales	16141	non-null	float64
8	JP_Sales	16141	non-null	float64
9	Other_Sales	16141	non-null	float64
10	All_Sales	16141	non-null	float64

'M'을 기준으로 한 data set의 단위를 맞추기 위해,

'K'는 10의 마이너스 세제곱 (과학적 표기법 'le-3') 을 작성 후 map 함수에 (pd.eval : 사칙연산을 한다) 대입하여 숫자를 치환합니다.

후, sales 데이터들은 숫자로써 의미를 갖는 data로, astype() 함수를 통해 'float'(실수:소수점)으로 변환해줍니다.

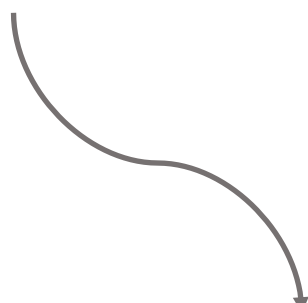
PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

3 EDA

```
game1 = game1[game1.Year >= 1980]  
game1 = game1.sort_values(by='Year' ,ascending=True)  
game1.describe()
```

최초의 게임은 1958년에 생산되었으나,
해당 데이터는 1980년을 기준으로 배열됩니다.
하지만, 1~20까지의 잘못된 데이터가 있으므로
변수를 1980년 이상으로 지정해 삭제합니다.



	Unnamed: 0	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales
count	16141.000000	16141.000000	16141.000000	16141.000000	16141.000000	16141.000000
mean	8292.228486	2006.406356	0.265156	0.147319	0.077951	0.048224
std	4792.105258	5.830780	0.822621	0.508480	0.306771	0.189939
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000
25%	4143.000000	2003.000000	0.000000	0.000000	0.000000	0.000000
50%	8284.000000	2007.000000	0.080000	0.020000	0.000000	0.010000
75%	12444.000000	2010.000000	0.240000	0.110000	0.040000	0.040000
max	16598.000000	2020.000000	41.490000	29.020000	10.220000	10.570000

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING

```
game2=game1[['Genre', 'NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]
#game2 = game - regeion sales
game2=game1.groupby(game1.Genre)['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales'].sum()
game2.head()
```

	NA_Sales	EU_Sales	JP_Sales	Other_Sales
Genre				
Action	855.96	512.52	155.73	183.64
Adventure	101.33	63.32	51.79	16.64
Fighting	220.11	99.72	86.39	36.10
Misc	394.29	207.68	103.71	72.54
Platform	442.46	199.40	129.22	51.15

DATA WRANGLING 란, 데이터를 정리하고 통합하는 단계입니다.

의미있는 **features**들과 도메인 지식, 창의력, 수한 연산등을 활용한 재조합을 통해 의미있는 데이터를 찾고, 세세한 분석, 모델 성능을 향상시키는 단계로 '**feather engineering**' 의 단계를 포함할 수 있습니다.

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 지역에 따라 선호하는 게임의 장르가 있는가

```
import pandas as pd
import matplotlib.pyplot as plt

games2NA=game1.groupby('Genre', as_index=False)['NA_Sales'].sum()

plt.figure(figsize=(12,12))
plt.title('Genre-Sales')
plt.xlabel('Genre')
plt.ylabel('Sales')

plt.rc('font', family = 'AppleGothic')
plt.rc('font', size = 10)
plt.bar(games2NA['Genre'], games2NA['NA_Sales'], width=0.5)
```

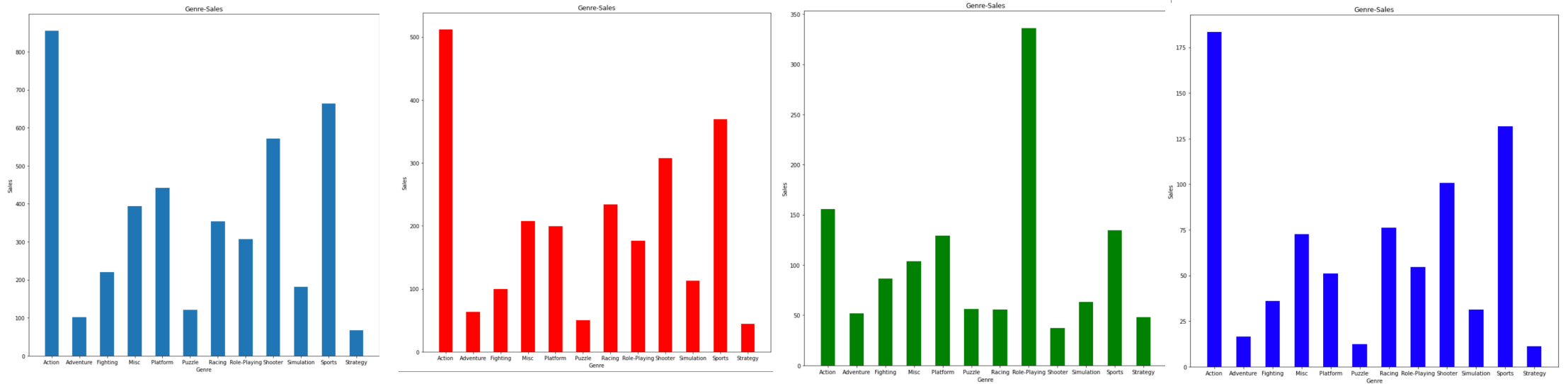
특히나, `groupby()` 함수를 사용할 경우, 원하는 컬럼을 기준으로 통합할 수 있습니다.

그러므로 `groupby()`는 ,DATA WRANGLING를 대표하는 함수입니다.

PROJECT1) PRESENTATION

Goal_다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 지역에 따라 선호하는 게임의 장르가 있는가



1 - 지역에 따라 선호하는 게임의 장르가 있는가

지역판매량과 장르를 비교시, 각 지역마다 값이 다른 점을 확인할 수 있습니다.

그러므로 지역마다 선호하는 장르가 다를 수 있습니다.

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 지역에 따라 선호하는 게임의 장르가 있는가

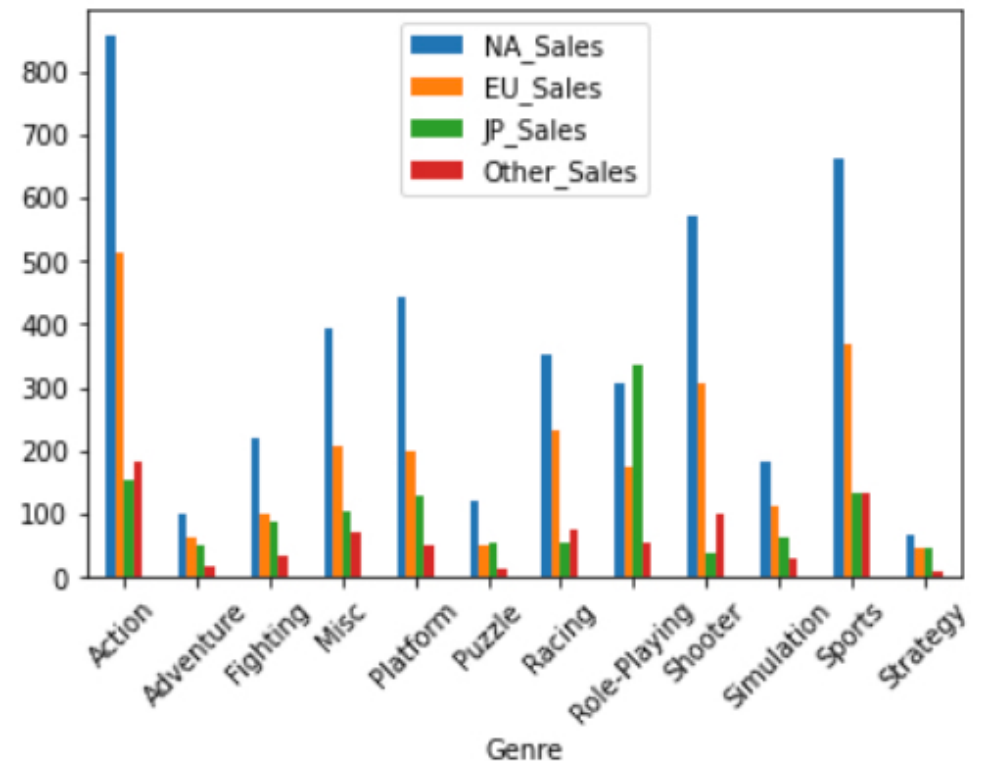
지역별 선호도

NA : ACTION > SPORT > SHOOTER

-EU : ACTION > SPORT > SHOOTER

-JP : ROLE - PLAYING

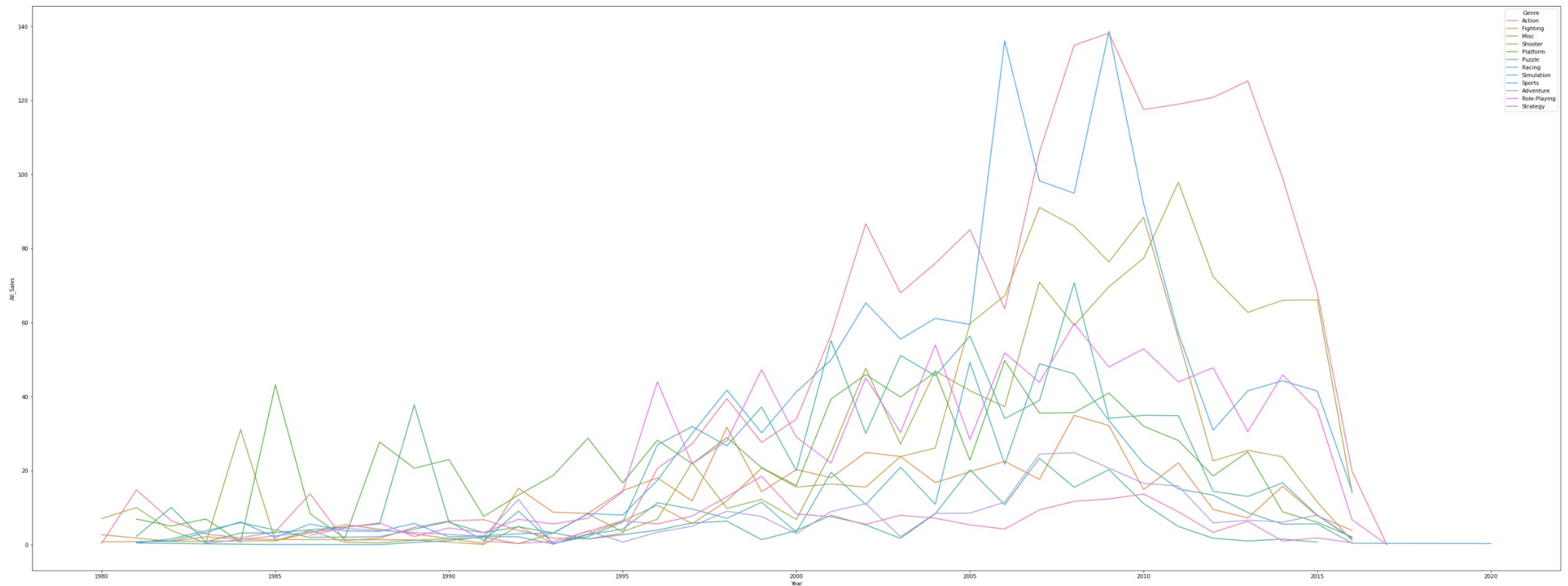
-OT : ACTION > SPORT > SHOOTER



PROJECT1) PRESENTATION

Goal_다음_분기에_어떤_게임을_설계해야_할까?

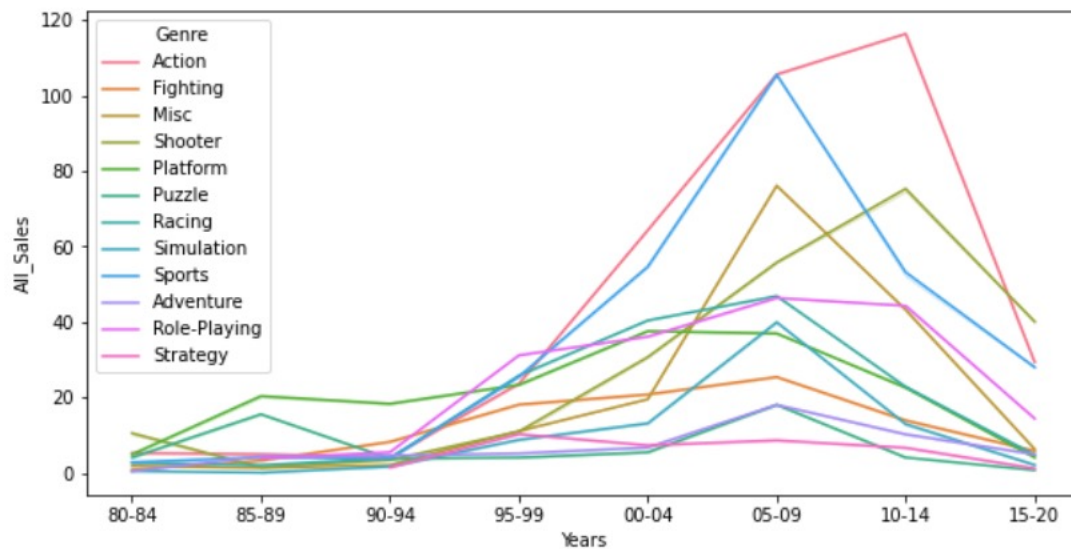
4 DATA WRANGLING : - 연도별 게임 트렌드가 있는가 - 장르



PROJECT1) PRESENTATION

Goal_다음_분기에_어떤_게임을_설계해야_할까?

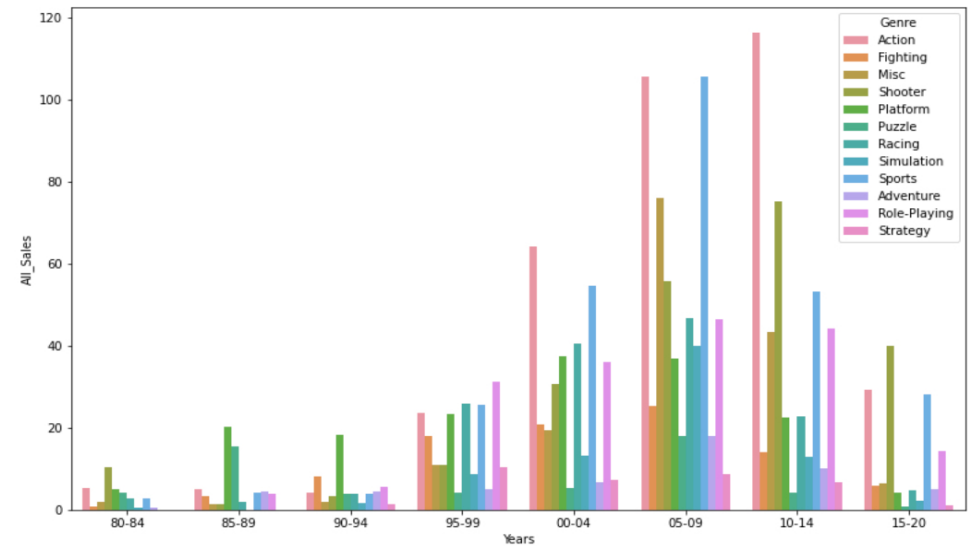
4 DATA WRANGLING : - 연도별 게임 트렌드가 있는가 - 장르



2 - 연도별 게임 트렌드가 있는가

5년 단위로 확인시,

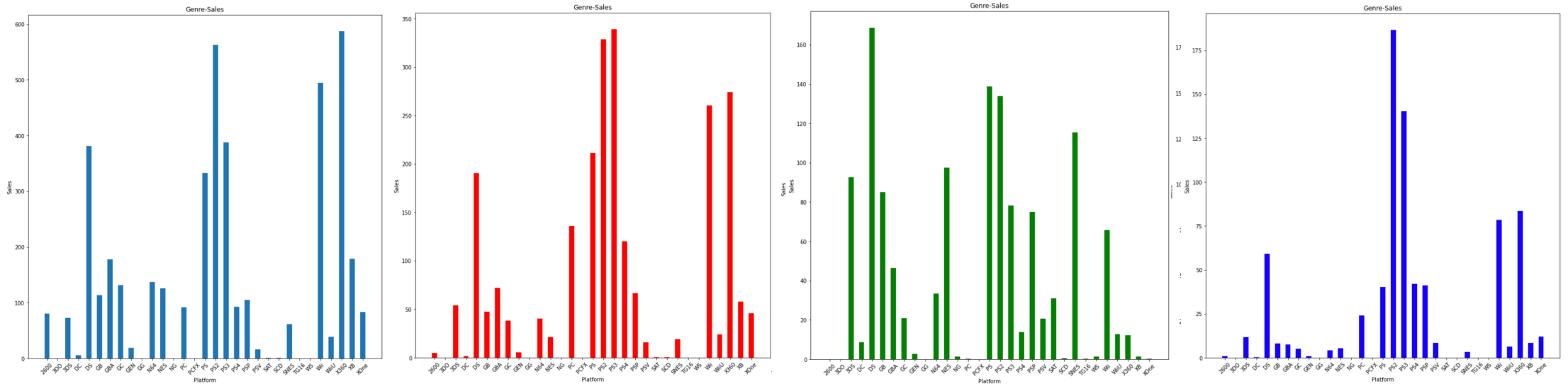
연도에 따라서 장르별 판매량의 격차가 있음을 알 수 있습니다.



PROJECT1) PRESENTATION

Goal_다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 연도별 게임 트렌드가 있는가 - 플랫폼



2 - 연도별 게임 트렌드가 있는가

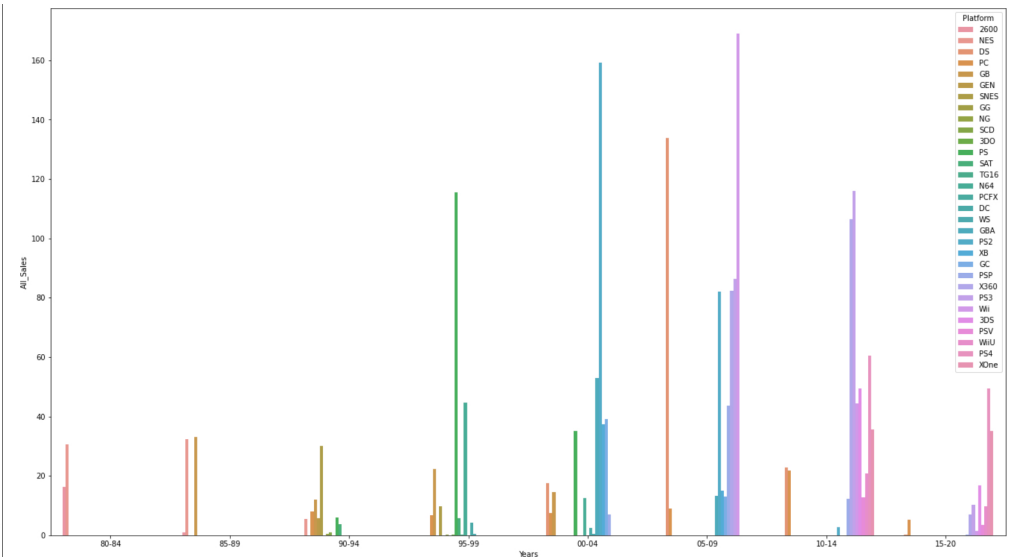
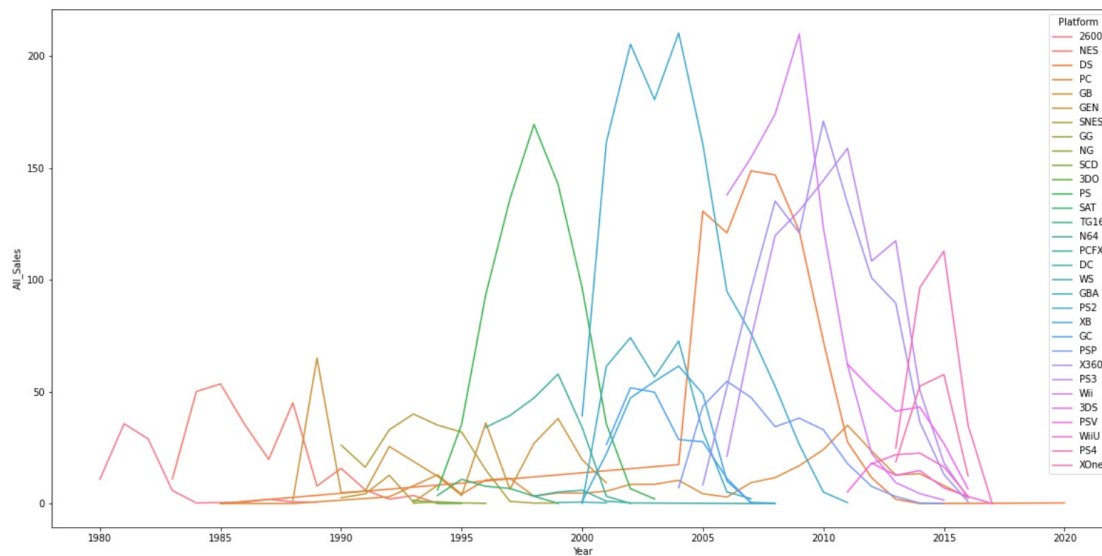
지역별 판매량을 기준으로 확인시

연도에 따라서 플랫폼별 판매량의 격차가 있음을 알 수 있습니다.

PROJECT1) PRESENTATION

Goal_다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 연도별 게임 트렌드가 있는가 - 플랫폼



2 - 연도별 게임 트렌드가 있는가

5년 단위로 확인시,

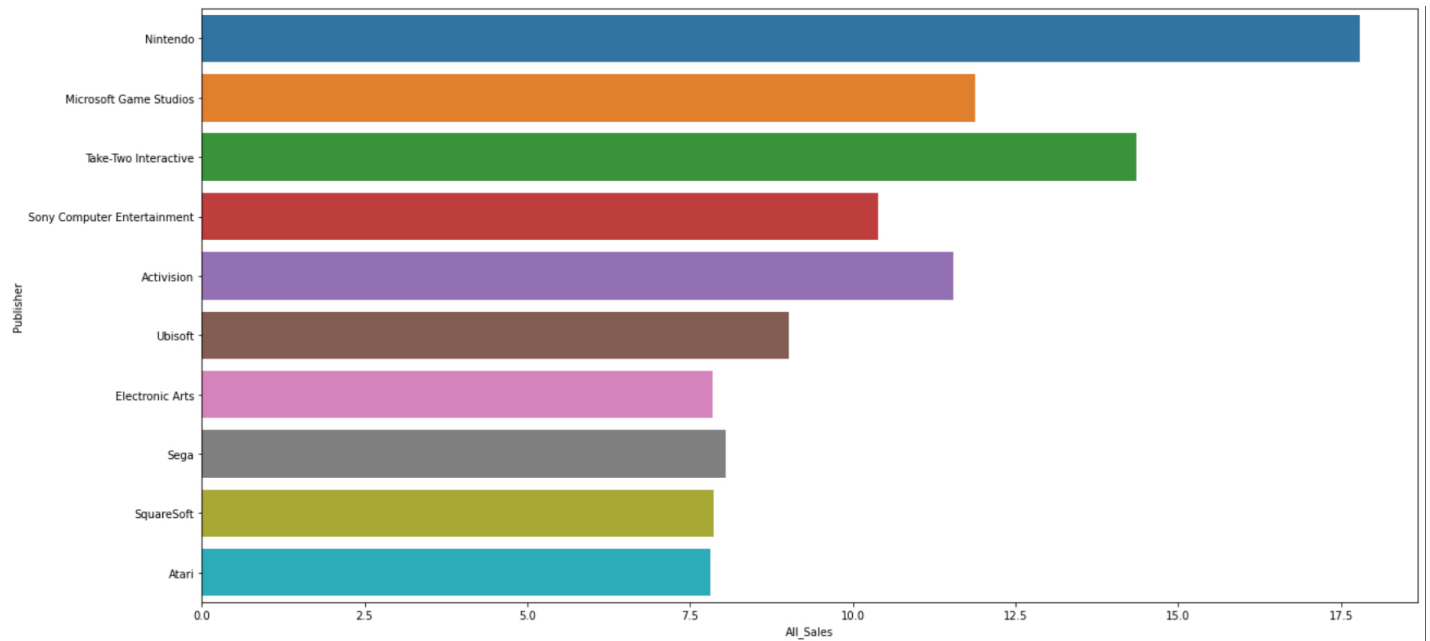
연도에 따라서 플랫폼별 판매량의 격차가 있음을 알 수 있습니다.

그러므로, 플랫폼 또한 트렌드에 영향을 받음을 알 수 있습니다.

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스



3 - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스

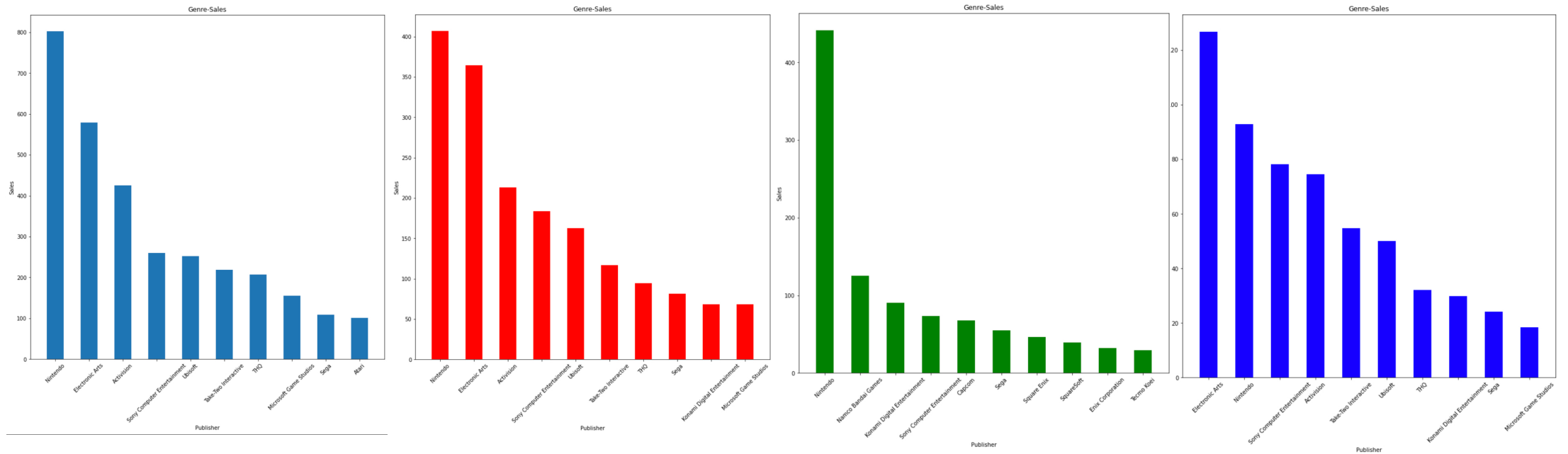
플랫폼의 경우,

전체적으로 닌텐도가 가장 선두이다.

PROJECT1) PRESENTATION

Goal_다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스



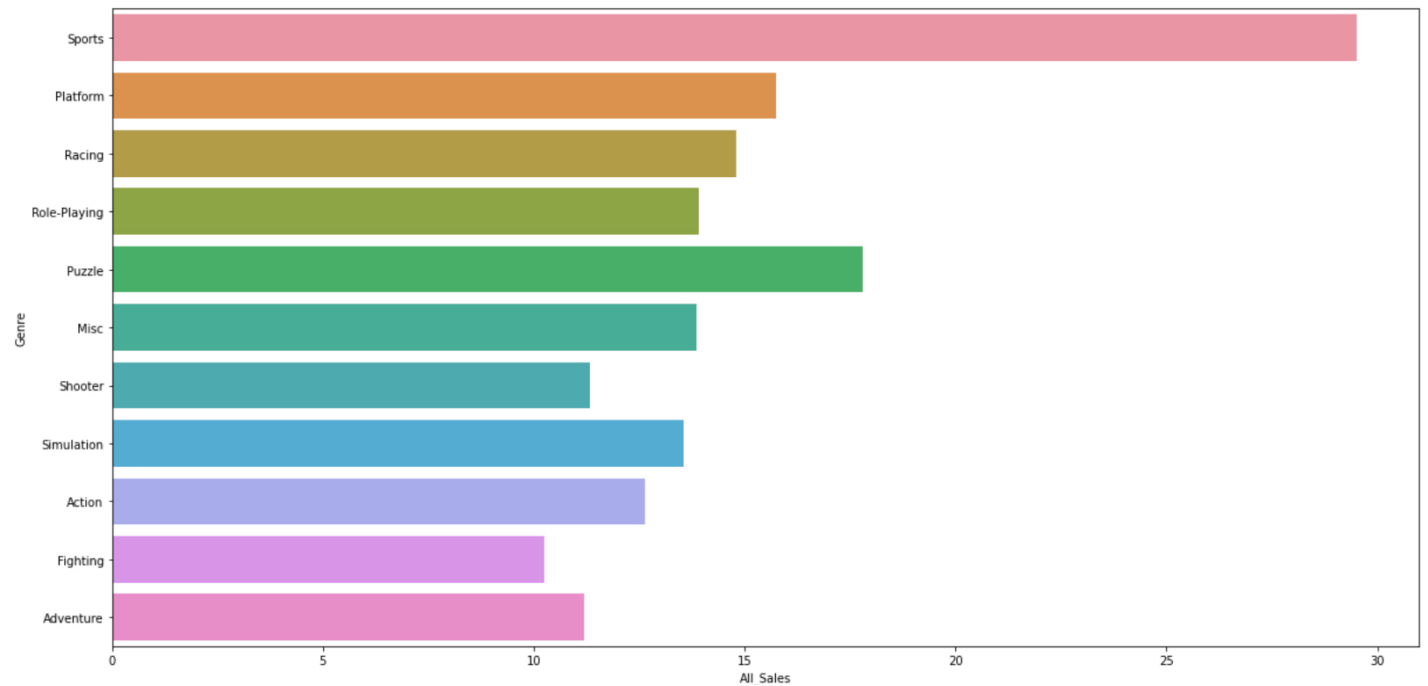
3 - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스

지역별 판매량으로 확인시 출판사 중 닌텐도가 선두를 달리는 부분을 확인 할 수 있다.

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스



3 - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스

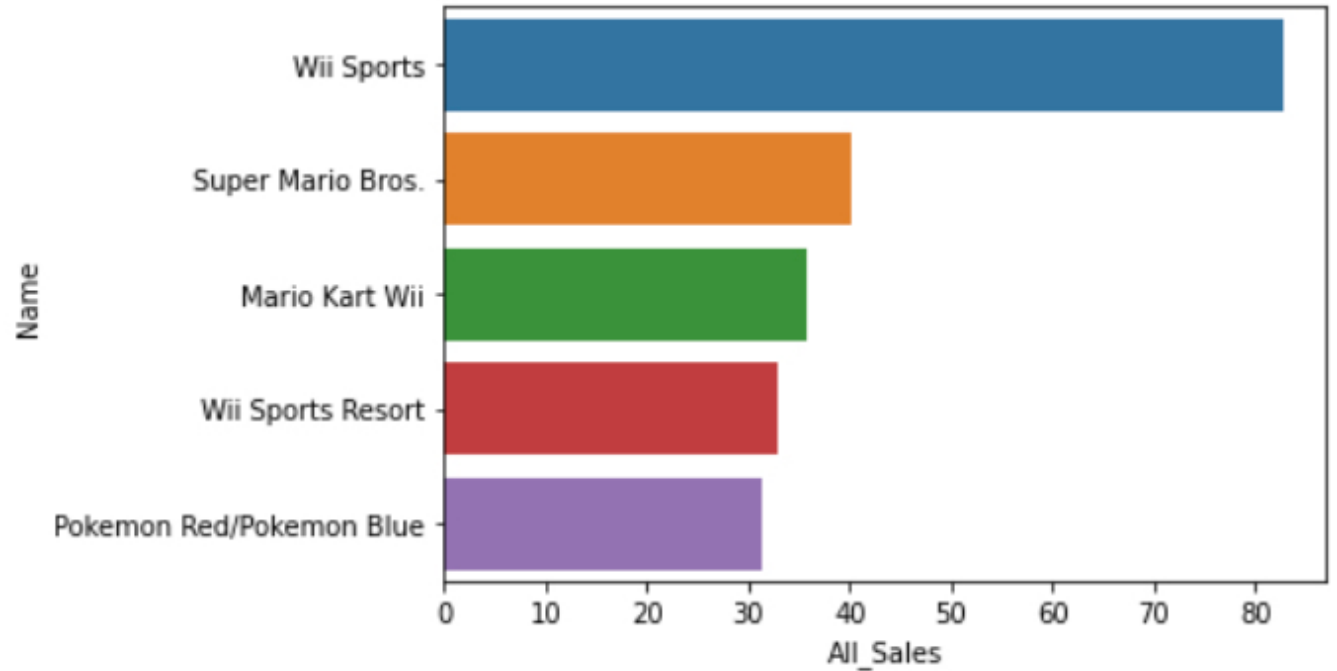
장르의 경우,

스포츠의 판매량이 가장 높음을 알 수 있다.

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스



3 - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스

게임 이름으로 확인시,

상위 **5**위의 게임 모두 사람의 움직임이 가미된 게임이 상위임을 알 수 있다.

PROJECT1) PRESENTATION

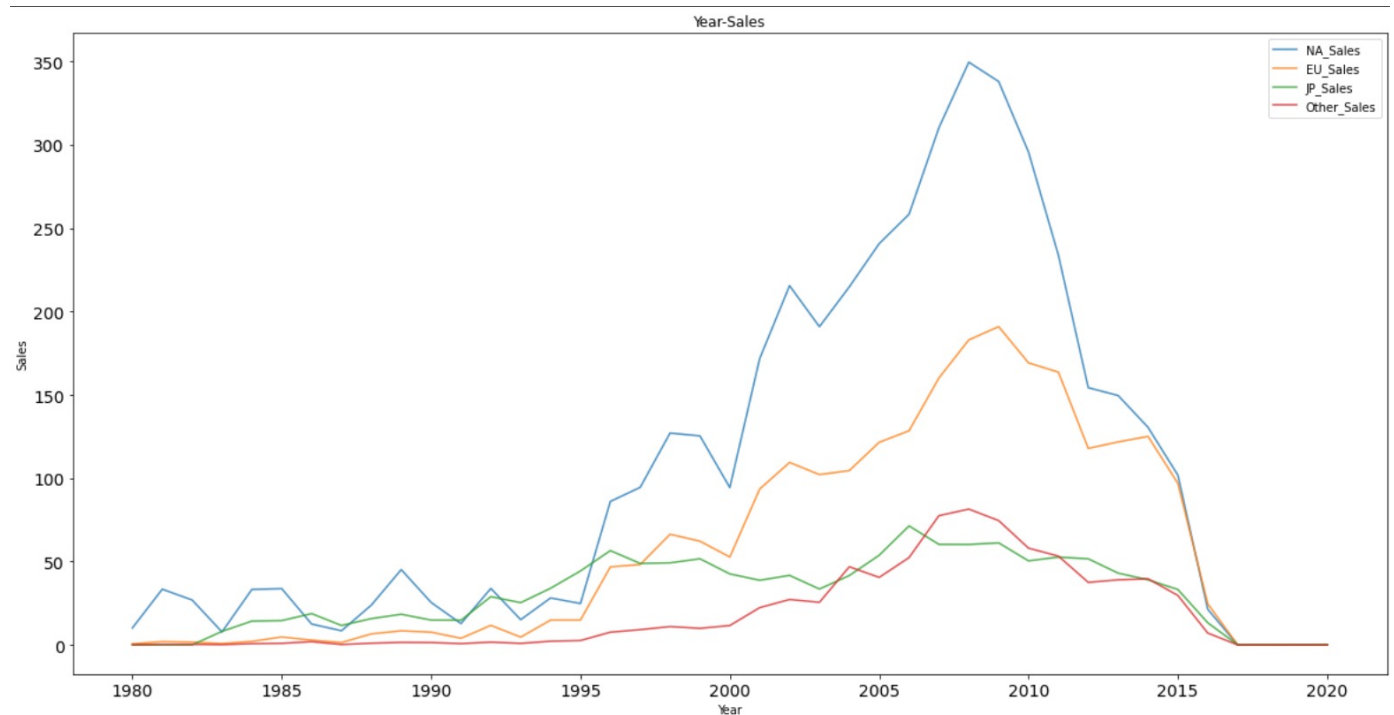
Goal_다음_분기에_어떤_게임을_설계해야_할까?

4 DATA WRANGLING : - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스

3 - 출고량이 높은 게임에 대한 분석 및 시각화 프로세스

지역판매량을 통해 연도별 판매량을 확인시,

NA가 압도적으로 판매량이 많다가, 최근 전체적으로 판매량이 감소함을 알 수 있다.



PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

5 가설 검증 및 결과 확인

- 귀무 가설 : 현재 게임 외 새로운 게임을 설계하지 않아도 된다.
- 대립 가설 : 현재 게임 외 새로운 어떤 게임을 설계할 필요가 있다.

지금까지의 데이터를 종합했을 때,

지역별 판매량에 차이가 있습니다.

*특히 과거의 **NA**의 판매량이 높았던 점에서 적극적인 고객이었음을 알 수 있습니다.*

지역별 장르에 대한 선호도가 다릅니다.

***JP**를 제외한 나머지의 국가에서 **ACTION**에 대한 선호도가 높습니다.*

지역별 플랫폼에 대한 선호도가 다릅니다.

지역별 출판사에 대한 선호도는 다르나, 대부분 닌텐도에 대한 선호도가 높습니다

PROJECT1) PRESENTATION

Goal _다음_분기에_어떤_게임을_설계해야_할까?

5 가설 검증 및 결과 확인

- 귀무 가설 : 현재 게임 외 새로운 게임을 설계하지 않아도 된다.
- 대립 가설 : 현재 게임 외 새로운 어떤 게임을 설계할 필요가 있다.

종합적으로 **JP**를 제외한 나머지 지역에서 비슷한 트렌드를 따르는 경향을 보임을 알 수 있으며,
게임의 트렌드의 변화를 확인시,
새로운 게임에 대해 거부감 적게 다가갈 수 있는 소비자임을 알 수 있습니다.

상위 **5**개의 게임을 나열시, 사람의 몸을 사용하거나 증강현실 등의 게임이 성행하는 것으로 보입니다.
이에, 상위 플랫폼과 출판사를 통해 게임을 런칭하고,
제한된 예산을 고려하여 **JP**를 제외한 나머지 지역을 타겟화한

증강현실, **ACTION**을 활용한 새로운 게임을 출시한다면, **TO**(위협/ 기회) 적 시장에서 성공적인 게임 출시를 기대할 수 있을 것으로 보입니다.

PROJECT1) PRESENTATION

Goal_다음_분기에_어떤_게임을_설계해야_할까?

5 가설 검증 및 결과 확인

- 귀무 가설 : 현재 게임 외 새로운 게임을 설계하지 않아도 된다.
- 대립 가설 : 현재 게임 외 새로운 어떤 게임을 설계할 필요가 있다.

지금까지의 데이터를 종합했을 때,

지역별 판매량에 차이가 있다. 특히 과거의 **NA**의 판매량이 높았던 점에서

지역별 장르에 대한 선호도가 다르다.

지역별 플랫폼에 대한 선호도가 다르다.

지역별 출판사에 대한 선호도는 다르나, 대부분 닌텐도에 대한 선호도가 높다

종합적으로 트렌드를 따르는 경향을 보임을 알 수 있으며,

상위 **5**개의 게임을 나열시 사람의 몸을 사용하거나 증강현실 등의 게임이 성행하는 것으로 보인다.