

# AIB\_13\_Section1\_Project

다음 분기에 어떤 게임을 설계해야 할까?

**AI\_13\_김강호**

# 목차

1. 데이터 전처리 과정
2. 데이터 시각화 과정
3. 데이터 시각화 분석

# 1. 데이터 전처리 과정

# 1 - 1]. 데이터 불러오기

0초

▶

# 데이터 불러오기

```
import pandas as pd
import numpy as np

df = pd.read_csv('/content/sample_data/Section_1_project_vgames2.csv')
df.head()
```

📄

Unnamed: 0

		Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	1	Candace Kane's Candy Factory	DS	2008	Action	Destineer	0.04	0	0	0
1	2	The Munchables	Wii	2009	Action	Namco Bandai Games	0.17	0	0	0.01
2	3	Otome wa Oanesama Boku ni Koi Shiteru Portable	PSP	2010	Adventure	Alchemist	0	0	0.02	0
3	4	Deal or No Deal: Special Edition	DS	2010	Misc	Zoo Games	0.04	0	0	0
4	5	Ben 10 Ultimate Alien: Cosmic Destruction	PS3	2010	Platform	D3Publisher	0.12	0.09	0	0.04

pd.read\_csv()로 데이터 불러온 후 df 저장

# 1 - 2]. 데이터 전처리

## 1. 결측치 처리 : 결측치 개수 조사 & 제거

[4] #. 결측치 개수 조사

```
pd.DataFrame(df.isnull().sum(), columns=["결측치 개수"])
```

	결측치 개수
Name	0
Platform	0
Year	271
Genre	50
Publisher	58
NA_Sales	0
EU_Sales	0
JP_Sales	0
Other_Sales	2

**df.isnull().sum()을 이용하여 dataframe 내 컬럼별 결측치 개수 조사**

# 1 - 2]. 데이터 전처리

## 1. 결측치 처리 : 결측치 개수 조사 & 제거

```
df = df.dropna(axis=0, how='any')
df = df.reset_index(drop = True)
df
```

### Dropna를 통해 결측값이 있는 행 전체를 제거

- 가정 1 : drop값이 있다는 것은 상대적으로 덜 알려진 게임이라고 가정
  - 상대적으로 덜 알려진 게임을 drop하는 것이 게임의 트렌드를 파악하는 데 있어 큰 지장을 주지 않는다고 가정.
- 가정 2 : drop 후에도 여전히 많은 수의 게임이 남아있기 때문에 (drop 전 게임 수: 16598개, drop후 게임 수: 16239개) 게임의 트렌드를 파악하는 데에는 지장이 없다.

# 1 - 2]. 데이터 전처리

## 2. 숫자 데이터 전처리 - 컬럼 'Year' 데이터 전처리

### 'Year'의 자리수를 4자리로 통일

1. Year이 4자리 : 데이터 형태만 변환시켜 사용
2. Year이 2자리
  - Int(Year)이 21보다 같거나 클 때 : Year 앞에 '19'를 붙임
  - 그 외 : Year앞에 '20'를 붙여 년도를 사용
3. Year이 1자리
  - 결측값으로 만든 후에 dropna로 결측값이 있는 행 데이터 제거

```
a = []

for i in df['Year'].values:
    if len(i) == 4:
        a.append(i)
    elif len(i) == 2:
        if int(i) >= 21:
            i = '19' + i
        else:
            i = '20' + i
        a.append(i)
    else:
        a.append(np.NaN)

df['Year'] = a

[15] df = df.dropna(axis=0, how='any')
df.isnull().sum()

Name      0
Platform  0
Year      0
Genre     0
Publisher  0
NA_Sales  0
EU_Sales  0
JP_Sales  0
Other_Sales 0
dtype: int64
```

# 1 - 2]. 데이터 전처리

## 2. 숫자 데이터 전처리 - 컬럼 'Year' 데이터 전처리

```
[16] def to_int(string):  
      return int(string)  
  
df['Year'] = df['Year'].apply(to_int)  
df['Year'].unique()  
  
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead.  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/timestamps.html  
after removing the cwd from sys.path.  
array([2008, 2009, 2010, 2005, 2011, 2007, 2001, 2002, 1997, 2013, 1996, 2004, 2000, 1984, 1995, 1993, 2012, 1987, 1982, 1994, 1990, 1991, 1989, 1986, 2017, 1980, 2020])
```

**'Year'의 dtype을 [object >> 정수형]  
으로 변환**



# 1 - 2]. 데이터 전처리

## 3. 숫자 데이터 전처리 - 출고량['지역명\_Sales'] 데이터 전처리

```
for j in range(5, len(df.columns)):
    a = []

    for i in df[df.columns[j]].values:
        if i[-1] == 'K':
            i = float(i.replace('K', ''))
            a.append(i * 1000)
        elif i[-1] == 'M':
            i = float(i.replace('M', ''))
            a.append(i * 1000000)
        else:
            a.append(float(i) * 1000000)

    df[df.columns[j]] = a
```

**단위를 1개 단위로 통일 후 출고량 컬럼  
들에 저장**

Ex)

1. 0.5(백만개) >> 500,000(개)
2. 350K(개) >> 350,000(개)
3. 0.5M(개) >> 500,000(개)

## 2. 데이터 시각화 과정

## 2 - 1]. 데이터 plotting

```
import matplotlib.pyplot as plt

plt.figure(figsize = (18, 12))

plt.subplot(231)

a = df[(df['Year'] >= 2015)].groupby('Genre').NA_Sales.mean()
a.plot(x = a.index, y = a.values, kind = 'bar', color = 'r')
plt.title('북미, 출시년도 : 2015~', fontsize = 20)
plt.xlabel('게임장르', fontsize = 15)
plt.ylabel('평균 출고량(개)', fontsize = 20)
plt.tick_params(labelsize = 13);

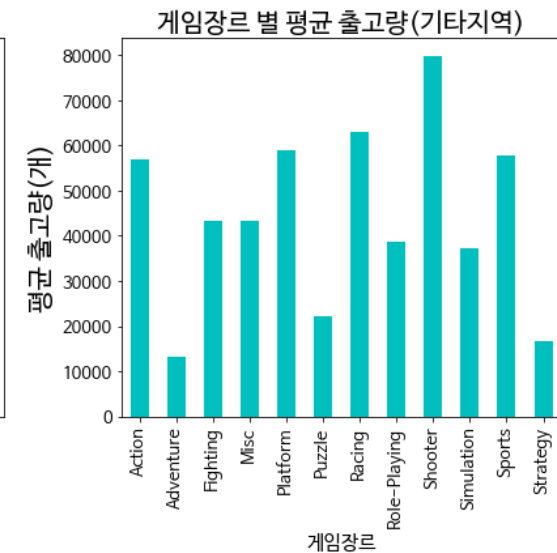
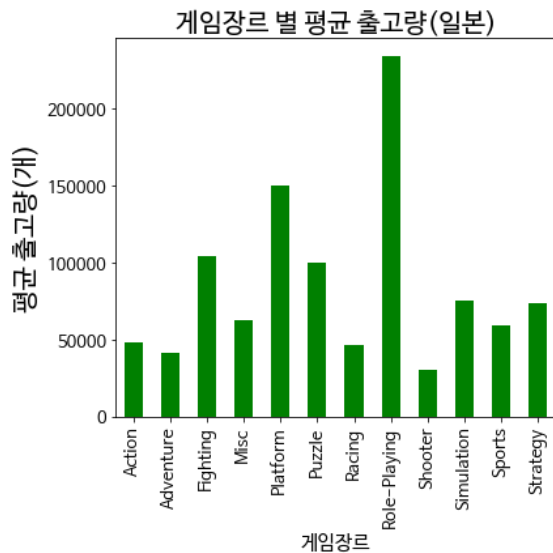
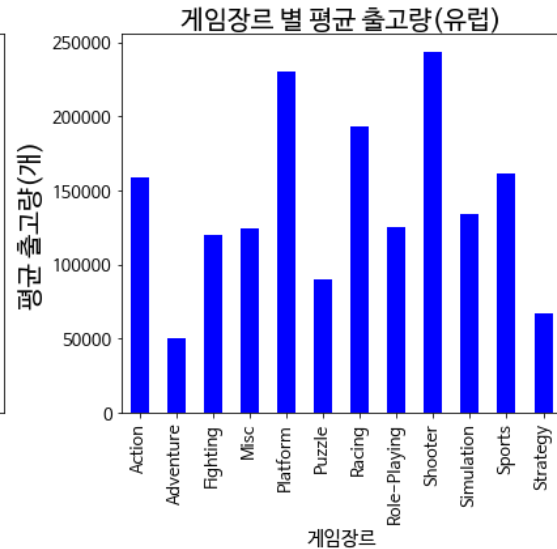
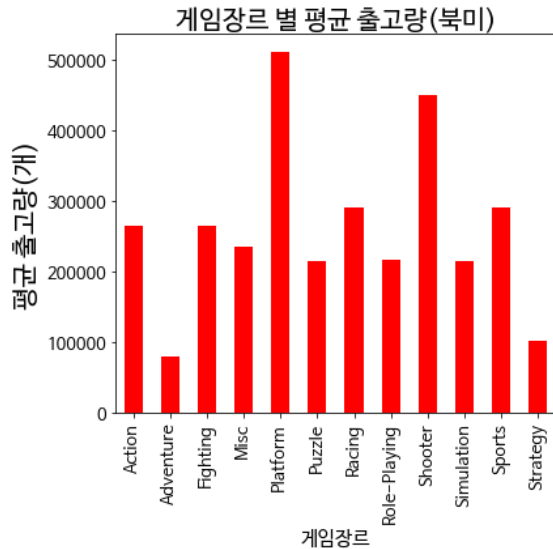
plt.subplot(232)
b = df[(df['Year'] >= 2015)].groupby('Genre').EU_Sales.mean()
b.plot(x = b.index, y = b.values, kind = 'bar', color = 'b')
plt.title('유럽, 출시년도 : 2015~', fontsize = 20)
plt.xlabel('게임장르', fontsize = 15)
plt.ylabel('평균 출고량(개)', fontsize = 20)
plt.tick_params(labelsize = 13);
```

### @ 코드설명

1. Matplotlib.pyplot(plt) 라이브러리 사용
2. Df의 conditioning, groupby 등을 통해 1개의 feature plotting을 위한 series데이터 구성
3. .plot()을 이용해 bar plot 생성
4. 피쳐끼리 비교를 할 수 있도록 Plt.subplot()을 이용해 여러 개의 그래프를 붙여서 그래프 생성

### **3. 데이터 시각화 분석**

# 3 - 1]. 지역별 게임 트렌드 분석



1. 지역 별 게임장르에 대한 평균 출고량(선호도)을 조사

2. 대상 게임 출시년도 : 1980 ~ 2020년

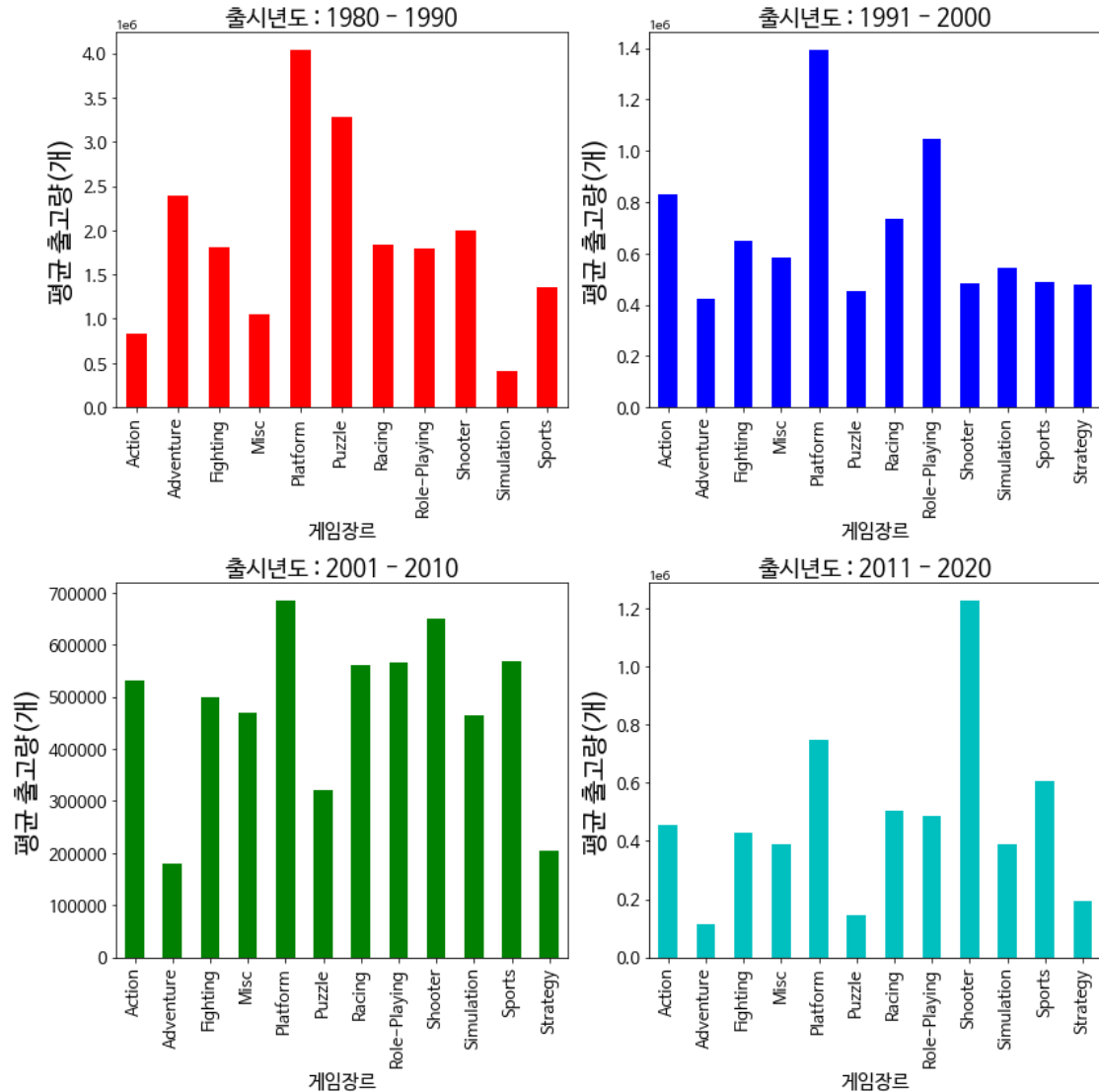
3. 그래프 분석

- 북미, 유럽 : Platform, Shooter 분야 선호
- 일본 : Role-Playing, Platform 분야 선호
- 기타지역 : 특히 Shooter 분야 선호, Racing, Platform, Action, Sports 분야들도 골고루 선호함

4. 결론 :

- 세계시장 - Platform, Shooter 분야의 게임이 유행해 옴
- 일본시장 - Role-playing 분야의 게임이 유행해 옴

## 3 - 2]. 연도별 게임 트렌드 분석



1. 게임 출시년도 별 게임장르에 대한 평균 출고량 (선호도, 인기)을 조사

2. 지역 전체 대상

3. 그래프 분석

- 출시년도가 2010년 이전 : **Platform**분야 게임의 인기가 제일 좋음, **Shooter**분야 게임의 약진
- 출시년도가 2010년 이후 : **Shooter**분야 게임의 인기가 **Platform**분야를 추월함.

4. 결론 :

- 게임 트렌드가 **Platform**분야에서 **Shooter**분야로 변하는 중이다.

# 3 - 3]. 출고량 Top10 게임들의 분석

## 1. 전 세계, 지역별 게임의 출고량 Top10조사

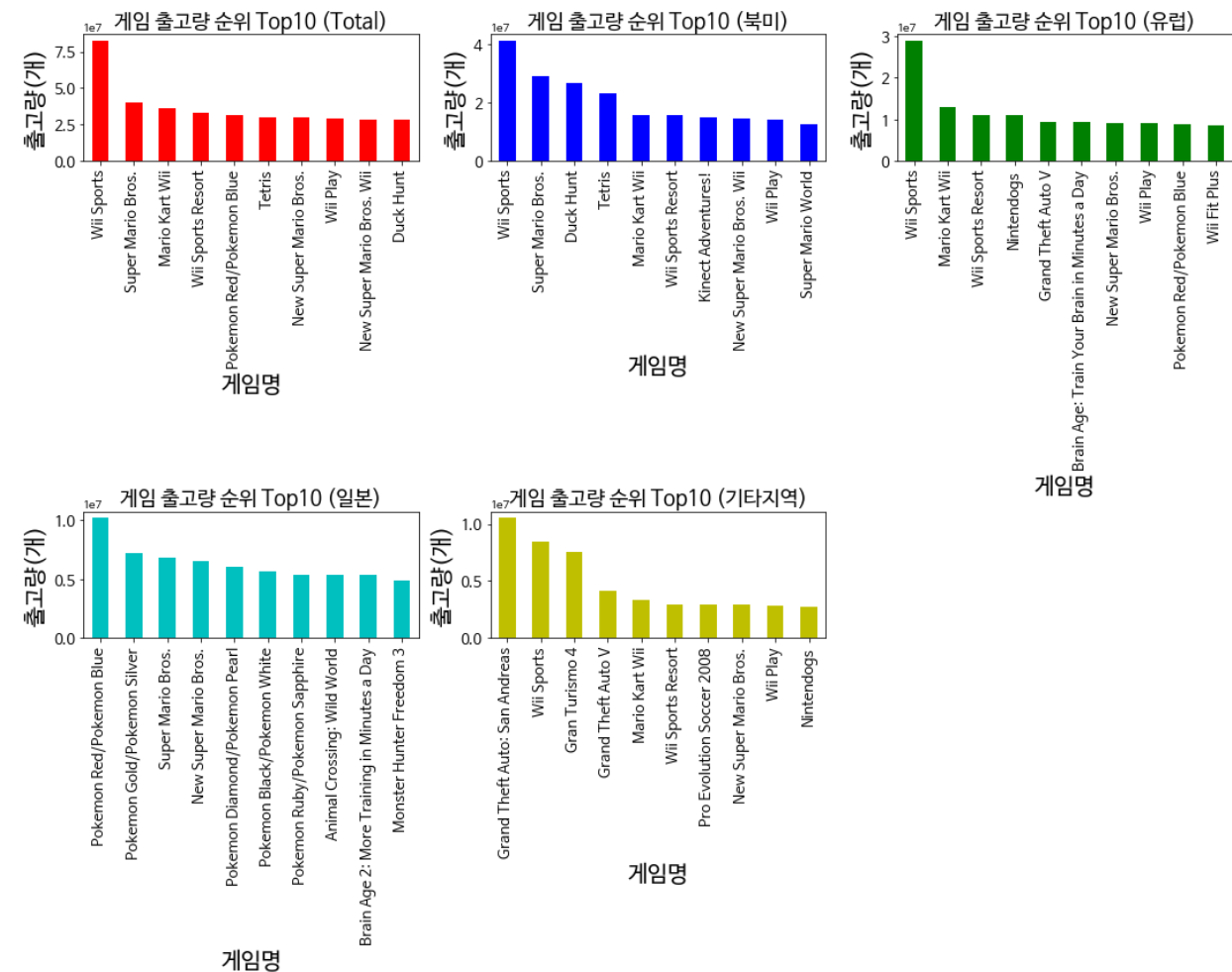
## 2. 게임 출시년도 : 1980 ~ 2020년

## 3. 그래프 분석

- 북미, 유럽 지역 게임 출고량 1위 : wii sports (sports분야)
- 일본 지역 게임 출고량 1위 : 포켓몬 레드/블루 (Role-Playing 분야)
- Top10 게임들의 게임장르 분포
  - 일본 이외 : Action, Sports, Racing 등으로 다양하게 분산 됨
  - 일본 : Role-Playing과 Platform으로 편중 됨.
- 그러나, Top10 게임들의 출시년도 대부분이 2010년 이전임.

## 4. 결론 :

- 출고량 Top10 게임들의 분석결과는 "미래에 어떤 게임을 설계해야 될까?"라는 물음엔 인사이트를 주지 못한다.



# 3 - 4]. 중간 피드백

## 1. 지역별로 게임의 트렌드가 다름

- 세계시장 - Platform, Shooter 분야의 게임이 유행해 옴
- 일본시장 - Role-playing 분야의 게임이 유행해 옴

## 2. 연도별로 게임의 트렌드가 변화해 옴

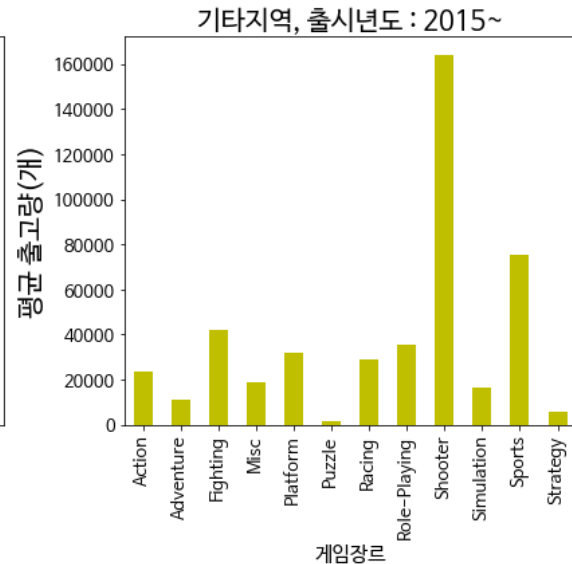
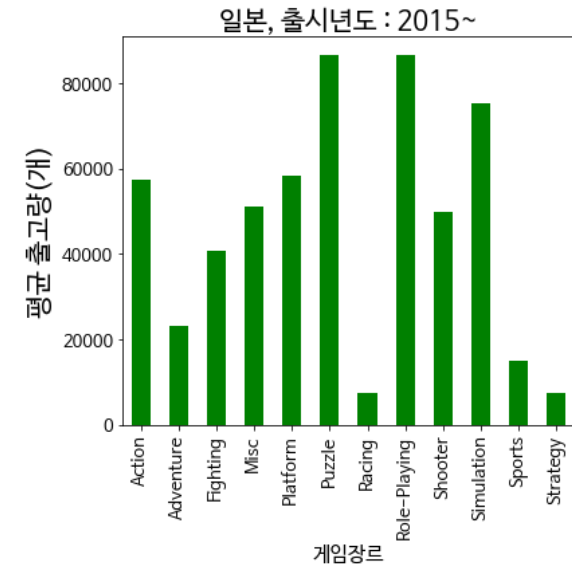
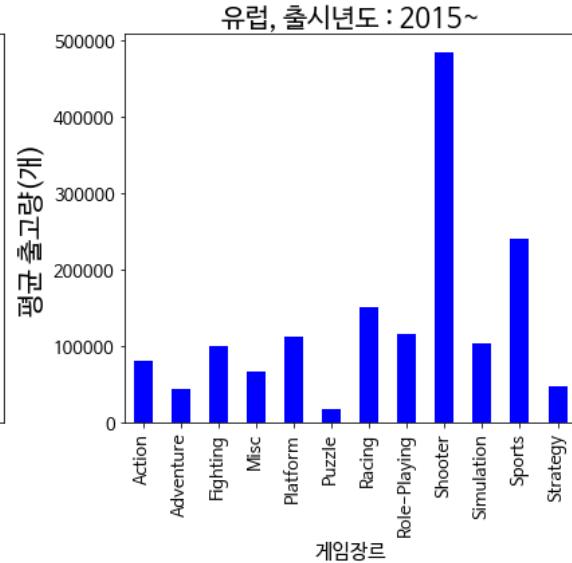
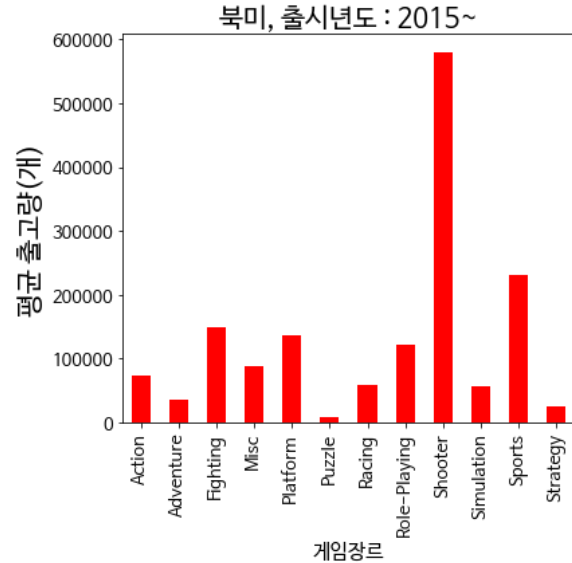
- 게임 트렌드가 Platform분야에서 Shooter분야로 변하는 중이다.

## 3. 결론 :

- 지역별로 출시연도가 최근인 게임들의 평균 출고량을 조사해서 미래의 게임 트렌드를 파악하기
  - 가까운 미래[다음 분기의 게임 설계]를 가까운 과거[최신게임들의 게임 장르별 평균 판매량]를 통해 예측



# 3 - 5]. 최신게임들의 지역별 트렌드 분석



1. 최신게임들의 지역 별 게임장르에 대한 평균 출고량(선호도)을 조사

2. 대상 게임 출시년도 : 2015 ~ 2020년

3. 그래프 분석

- 북미, 유럽, 기타지역 : Shooter, sports 분야 선호 됨
- 일본 : Role-Playing, Puzzle, Simulation 분야 선호 됨

4. 다음분기의 게임 설계 트렌드에 대한 결론 :

- 일본 외 시장 공략 : Shooter, sports 분야의 게임 위주로 개발
- 일본시장 : Role-Playing, Puzzle, Simulation 분야의 게임 위주로 개발