

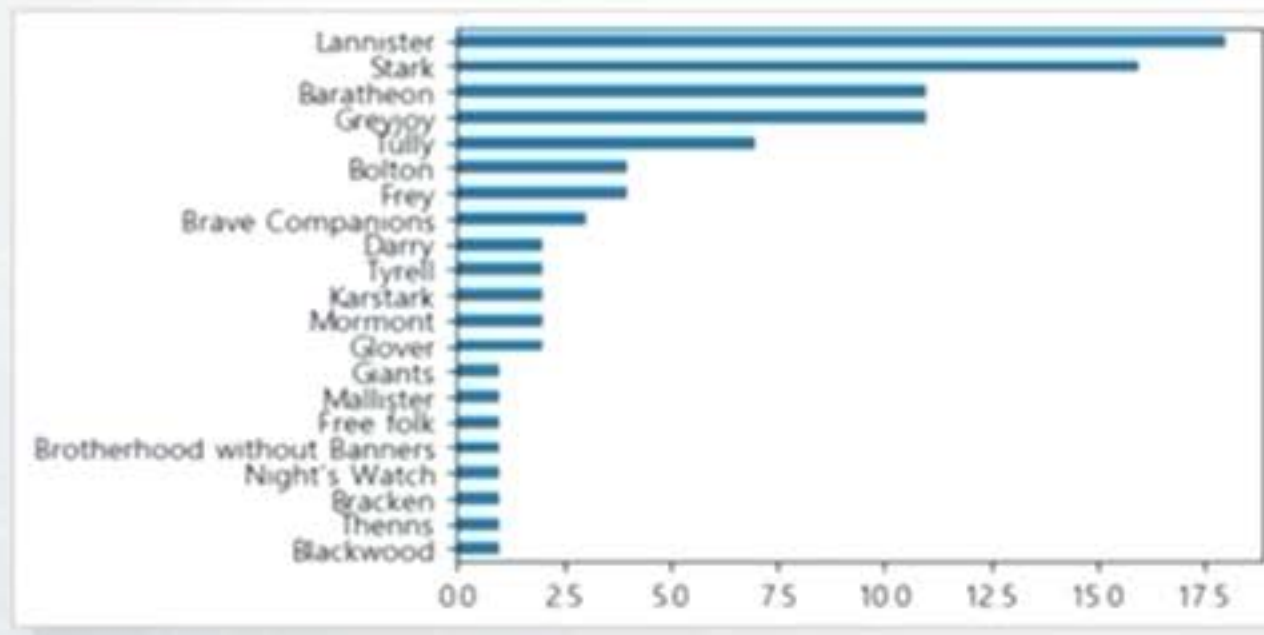
# 빅데이터 실습

10주차 3차시

왕좌의 게임 데이터 분석 [2]

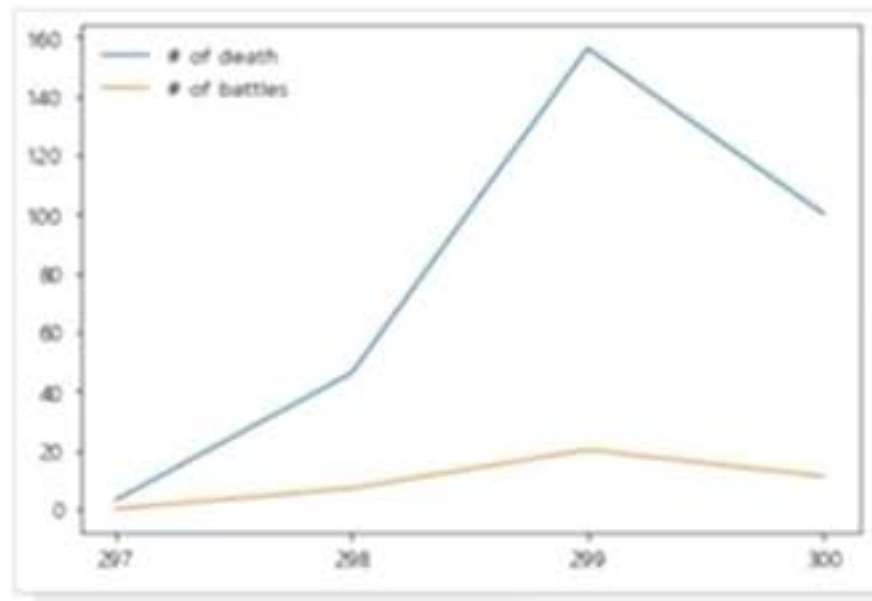
## 데이터 시각화

### 1/ 왕좌의 게임 데이터 분석 [3] - 가문별 전쟁 참여 횟수 분석 및 시각화



## 데이터 시각화

### 2/ 왕좌의 게임 데이터 분석 [4] - 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화





01

# 왕좌의 게임 데이터 분석 [ 3 ]



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

📄 + ✂ 📄 ⬆ ⬆ ▶ Run ■ ↺ ▶ Code ▾ 🖨

In [ ]:

**battles**

전쟁에 대한 정보

4) (실습 #1) 가문 별로 전쟁을 참여한 횟수 시각화

가문 정보는 총 8개 컬럼으로 나뉘어 제공

In [139]: battles

Out [139]:

	name	year	battle_number	attacker_king	defender_king	attacker_1	attacker_2	a
0	Battle of the Golden Tooth	298	1	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	
1	Battle at the Mummer's Ford	298	2	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	
2	Battle of Riverrun	298	3	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	
3	Battle of the Green Fork	298	4	Robb Stark	Joffrey/Tommen Baratheon	Stark	NaN	
4	Battle of the Whispering Wood	298	5	Robb Stark	Joffrey/Tommen Baratheon	Stark	Tully	

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

attacker\_1 ~ defender\_4(총 8개 컬럼) 선택

### 4) (실습 #1) 가문 별로 전쟁을 참여한 횟수 시각화

```
In [139]: # 1. 필요한 컬럼만 선택
# attacker_1~ defender_4, 총 8개 컬럼
battles
```

Out [139]:

	name	year	battle_number	attacker_king	defender_king	attacker_1	attacker_2	a
0	Battle of the Golden Tooth	298	1	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	
1	Battle at the Mummer's Ford	298	2	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	
2	Battle of Riverrun	298	3	Joffrey/Tommen Baratheon	Robb Stark	Lannister	NaN	
3	Battle of the Green Fork	298	4	Robb Stark	Joffrey/Tommen Baratheon	Stark	NaN	
	Battle of the							



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

① attacker\_1 ~ defender\_4까지 모두 나열

### 4) (실습 #1) 가문 별로 전쟁을 참여한 횟수 시각화

In [140]:

8개 컬럼

```
er_2', 'attacker_3', 'attacker_4', 'attacker_1', 'attacker_2', 'attacker_3', 'attacker_4']
```

Out [140]:

	attacker_1	attacker_2	attacker_3
0	Lannister	NaN	NaN
1	Lannister	NaN	NaN
2	Lannister	NaN	NaN
3	Stark	NaN	NaN
4	Stark	Tully	NaN
5	Stark	Tully	NaN
6	Lannister	NaN	NaN

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

① attacker\_1 ~ defender\_4까지 모두 나열

```
er_2', 'attacker_3', 'attacker_4', 'defender_1', 'defender_2', 'defender_3', 'defender_4']]
```

Out[141]:

	attacker_1	attacker_2	attacker_3	attacker_4	defender_1	defender_2	defender_3	defender_4
0	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	N
1	Lannister	NaN	NaN	NaN	Baratheon	NaN	NaN	N
2	Lannister	NaN	NaN	NaN	Baratheon	NaN	NaN	N
3	Stark	NaN	NaN	NaN	Lannister	NaN	NaN	N
4	Stark	Tully	NaN	NaN	Lannister	NaN	NaN	N
5	Stark	Tully	NaN	NaN	Lannister	NaN	NaN	N
6	Lannister	NaN	NaN	NaN	Darry	NaN	NaN	N
7	Greyjoy	NaN	NaN	NaN	Stark	NaN	NaN	N
8	Greyjoy	NaN	NaN	NaN	Stark	NaN	NaN	N
9	Greyjoy	NaN	NaN	NaN	Stark	NaN	NaN	N
10	Stark	NaN	NaN	NaN	Greyjoy	NaN	NaN	N

**Lannister가문이 Tully 가문 공격**



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

① attacker\_1 ~ defender\_4까지 모두 나열

**Stark와 Tully 가문이 합동으로 Lannister 가문 공격**

14	Stark	Tully	NaN	NaN	Lannister	NaN	NaN	N
15	Baratheon	NaN	NaN	NaN	Baratheon	NaN	NaN	N
16	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	N
17	Stark	NaN	NaN	NaN	Lannister	NaN	NaN	N
18	Stark	NaN	NaN	NaN	Lannister	NaN	NaN	N
19	Baratheon	NaN	NaN	NaN	Lannister	NaN	NaN	N
20	Darry	NaN	NaN	NaN	Lannister	NaN	NaN	N
21	Stark	NaN	NaN	NaN	Lannister	NaN	NaN	N

Brotherhood

Dragon

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

① attacker\_1 ~ defender\_4까지 모두 나열

20	Barry	NaN	NaN	NaN	Lannister	NaN	NaN	N
21	Stark	NaN	NaN	NaN	Lannister	NaN	NaN	N
22	Brotherhood without Banners	NaN	NaN	NaN	Brave Companions	NaN	NaN	N
23	Lannister	NaN	NaN	NaN	Stark	NaN	NaN	N
24	Lan	Free folk와 Thenns, Giants 합동 공격						N
25	Frey	Bolton	NaN	NaN	Stark	NaN	NaN	N
26	Frey	NaN	NaN	NaN	Mallister	NaN	NaN	N
27	Free folk	Thenns	Giants	NaN	Night's Watch	Baratheon	NaN	N
28	Bolton	NaN	NaN	NaN	Greyjoy	NaN	NaN	N
29	Brave Companions	NaN	NaN	NaN	NaN	NaN	NaN	N
30	Baratheon	Karstark	Mormont	Glover	Greyjoy	NaN	NaN	N
31	Greyjoy	NaN	NaN	NaN	Tyrell	NaN	NaN	N
32	Stark	NaN	NaN	NaN	Tyrell	NaN	NaN	N



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

① attacker\_1 ~ defender\_4까지 모두 나열

In [ ]:

In [ ]:

In [ ]:

### 4) (실습 #1) 가문 별로 전쟁을 참여한 횟수 시각화

```
In [141]: # 1. 필요한 컬럼만 선택
# attacker_1~ defender_4, 총 8개 컬럼
# #1 모두 나열 방식
battles[['attacker_1', 'attacker_2', 'attacker_3', 'attacker_4', 'defender_1', 'defender_2', 'defender_3', 'defender_4']]
```

Out [141]:

가장 기본적인 방식으로 8개 컬럼을 모두 나열할 수 있습니다.

1 Lannister NaN NaN NaN Baratheon NaN NaN N



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

② loc를 활용한 컬럼 슬라이싱

In [ ]:

In [ ]:

In [ ]:

### 4) (실습 #1) 가문 별로 전쟁을 참여한 횟수 시각화

```
In [141]: # 1. 필요한 컬럼만 선택
# attacker_1~ defender_4. 총 8개 컬럼
# #1 모두 나열 방식
battles[['attacker_1', 'attacker_2', 'attacker_3', 'attacker_4', 'defender_1', 'defender_2', 'defender_3', 'defender_4']]
# #2. 컬럼을 슬라이싱
```

8개 컬럼이 연속되어 있으므로, 컬럼명을 슬라이싱하여 얻을 수 있습니다.

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

② loc를 활용한 컬럼 슬라이싱

### 4) (실습 #1) 가문 별로 전쟁을 참여한 횟수 시각화

```
In [142]: # 1. 필요한 컬럼만 선택
# attacker_1~ defender_4. 총 8개 컬럼
# #1 모두 나열 방식
battles[['attacker_1', 'attacker_2', 'attacker_3', 'attacker_4', 'defender_1', 'defender_2', 'defender_3', 'defender_4']]
# #2. 컬럼을 슬라이싱
battles.loc[:, 'attacker_1': 'defender_4']
```

**battles.loc[ ]** 컬럼 슬라이싱

Out [142]:

	attacker_1	attacker_2	attacker_3	attacker_4	defender_1	defender_2	defender_3	defender_4
0	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	N
1	Lannister	NaN	NaN	NaN	Baratheon	NaN	NaN	N
2	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	N
3	Stark	NaN	NaN	NaN	Lannister	NaN	NaN	N
4	Stark	Tully	NaN	NaN	Lannister	NaN	NaN	N
5	Stark	Tully	NaN	NaN	Lannister	NaN	NaN	N



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

③ columns를 활용한 컬럼명 슬라이싱

4) (실습 #1) 가문 별로 전쟁을 참여한 횟수 시각화

**battles.columns**

컬럼 목록 구한 후 색인

```
In [144]: # 1. 필요한 컬럼만 선택
# attacker_1~ defender_4. 총 8개 컬럼
# #1 모두 나열 방식
#battles[['attacker_1', 'attacker_2', 'attacker_3', 'attacker_4', 'defender_1', 'defender_2', 'defender_3', 'defender_4']]
# #2. 컬럼을 슬라이싱
#battles.loc[:, 'attacker_1': 'defender_4']
# #3. 슬라이싱의 또다른 방법
battles.columns[5:13]

Out[144]: Index(['attacker_1', 'attacker_2', 'attacker_3', 'attacker_4', 'defender_1', 'defender_2', 'defender_3', 'defender_4'],
              dtype='object')
```

In [ ]:

In [ ]:



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

③ columns를 활용한 컬럼명 슬라이싱

```
In [145]: # 1. 필요한 컬럼만 선택
# attacker_1~ defender_4, 총 8개 컬럼
# #1 모두 나열 방식
# battles[['attacker_1', 'attacker_2', 'attacker_3', 'attacker_4', 'defender_1', 'defender_2', 'defender_3', 'defender_4']]
# #2. 컬럼을 슬라이싱
# battles.loc[:, 'attacker_1': 'defender_4']
# #3. 슬라이싱의 또다른 방법
battles[battles.columns[5:13]]
```

Out [145]:

	attacker_1	attacker_2	attacker_3	attacker_4	defender_1	defender_2	defender_3	defender_4
0	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	N
1	Lannister	NaN	NaN	NaN	Baratheon	NaN	NaN	N
2	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	N
3	Stark	NaN	NaN	NaN	Lannister	NaN	NaN	N

Columns를 슬라이싱하여 얻은 컬럼 목록을 색인기호에 넣습니다.

6	Lannister	NaN	NaN	NaN	Darry	NaN	NaN	N
---	-----------	-----	-----	-----	-------	-----	-----	---

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 필요한 컬럼만 선택

④ 정규 표현식 활용

`battles.filter()`

regex 인자값 설정 → 특정 패턴 가진 컬럼명만 선택 가능

```
battles.filter(regex = 'attacker_[1-4]|defender_[1-4]')
```

Out[145]:

	attacker_1	attacker_2	attacker_3	attacker_4	defender_1	defender_2	defender_3	defender_4
0	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	N
1	Lannister	NaN	NaN	NaN	Baratheon	NaN	NaN	N
2	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	N
3	Stark	NaN	NaN	NaN	Lannister	NaN	NaN	N

패턴을 가지고 있는 경우에는 정규 표현식을 사용할 수 있습니다.



# 왕좌의 게임 데이터 분석 및 시각화 실습

✓ 각 가문들이 출현한 횟수 집계

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

Code

```
# 4. 정규표현식을 사용
```

```
battle_house = battles.filter(regex = 'attacker_[1-4]|defender_[1-4]')
```

```
In [149]: # 각 가문들이 출현한 횟수를 집계
battle_house.attacker_1
```

Out [149]:

	attacker_1	attacker_2	attacker_3	attacker_4	defender_1	defender_2	defender_3	defender_4
0	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	NaN
1	Lannister	NaN	NaN	NaN	Baratheon	NaN	NaN	NaN
2	Lannister	NaN	NaN	NaN	Tully	NaN	NaN	NaN
3	Stark	NaN	NaN	NaN	Lannister	NaN	NaN	NaN
4	Stark	Tully	NaN	NaN	Lannister	NaN	NaN	NaN
5	Stark	Tully	NaN	NaN	Lannister	NaN	NaN	NaN
6	Lannister	NaN	NaN	NaN	Darry	NaN	NaN	NaN
7	Greyjoy	NaN	NaN	NaN	Stark	NaN	NaN	NaN

직접 코드를 짜는 것보다 만들어진 함수를 잘 활용하는 것을 권장합니다.

9	Greyjoy	NaN	NaN	NaN	Stark	NaN	NaN	NaN
---	---------	-----	-----	-----	-------	-----	-----	-----



# 왕좌의 게임 데이터 분석 및 시각화 실습

✓ 각 가문들이 출현한 횟수 집계

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

📁 + ✂ 📄 ⬆ ⬇ ▶ Run ■ ↺ ⏮ Code ⏭

# 4. 정규표현식을 사용

```
battle_house = battles.filter(regex = 'attacker_[1-4]|defender_[1-4]')
```

```
In [150]: # 각 가문들이 출현한 횟수를 집계  
battle_house.attacker_1.value_counts()
```

```
Out [150]: Lannister      8  
           Stark          8  
           Greyjoy       7  
           Baratheon     6  
           Bolton        2  
           Frey           2  
           Darry          1  
           Brave Companions 1  
           Bracken        1  
           Free folk      1  
           Brotherhood without Banners 1  
           Name: attacker_1, dtype: int64
```

value\_counts()

각 가문의 출연 빈도수 세기

In [ ]:

5) (실습 #2) 년도별로 사망자의 숫자와 전쟁이 벌어진 횟수 시각화

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 각 가문들이 출현한 횟수 집계

```
In [153]: # 각 가문들이 출현한 횟수를 집계  
battle_house.attacker_1.value_counts() +
```

```
Out [153]: Baratheon      NaN  
Bolton      3.0  
Bracken     NaN  
Brave Companions  NaN  
Brotherhood without Banners  NaN  
Darry       NaN  
Free folk   NaN  
Frey        3.0  
Greyjoy     8.0  
Karstark    NaN  
Lannister   9.0  
Stark       NaN  
Thenns     NaN  
Tully      NaN  
dtype: float64
```

attacker_1	
Stark	8
Lannister	8
Greyjoy	7
Baratheon	6
Frey	2
Bolton	2
Free folk	1
Darry	1
Brave Companions	1
Brotherhood with out Banners	1
Bracken	1



attacker_2	
Tully	3
Karstark	2
Greyjoy	1
Bolton	1
Thenns	1
Lannister	1
Frey	1

같은 인덱스 라벨끼리 더해지면서 둘 중 하나라도 없으면 NaN값으로 표시됩니다.

Name: attacker\_4, dtype: int64



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 각 가문들이 출현한 횟수 집계

```
In [154]: # 각 가문들이 출현한 횟수를 집계
battle_house.attacker_1.value_counts().add(battle_house.attacker_2.value_counts(),
                                             fill_value = 0)
```

```
Out[154]: Baratheon      6.0
Bolton      3.0
Bracken     1.0
Brave Companions  1.0
Brotherhood without Banners  1.0
Darry       1.0
Free folk   1.0
Frey        3.0
Greyjoy     8.0
Karstark     2.0
Lannister    9.0
Stark        8.0
Thenns       1.0
Tully        3.0
dtype: float64
```

**add(fill\_value=0)**

값이 없는 경우, 0으로 처리

```
In [152]: battle_house.attacker_4.value_counts()
```

```
Out[152]: Glover      2
Name: attacker_4, dtype: int64
```



# 왕좌의 게임 데이터 분석 및 시각화 실습

✓ 각 가문들이 출현한 횟수 집계

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

Code

fill\_value = 0)

```
Out[154]: Baratheon      6.0  
Bolton      3.0  
Bracken     1.0  
Brave Companions 1.0  
Brotherhood without Banners 1.0  
Darry       1.0  
Free folk   1.0  
Frey        3.0  
Greyjoy     8.0  
Karstark    2.0  
Lannister   9.0  
Stark       8.0  
Thenns  
Tully  
dtype: float64
```

house\_to\_battle\_count 변수 누적된 값 저장

```
In [152]: house_to_battle_count = Series()  
fd
```

```
Out[152]: Glover      2  
Name: attacker_4, dtype: int64
```

# 왕좌의 게임 데이터 분석 및 시각화 실습

✓ 각 가문들이 출현한 횟수 집계

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

Code

fill\_value = 0)

```
Out[154]:
```

Baratheon	6.0
Bolton	3.0
Bracken	1.0
Brave Companions	1.0
Brotherhood without Banners	1.0
Darry	1.0
Free folk	1.0
Frey	3.0
Greyjoy	8.0
Karstark	2.0
Lannister	9.0
Stark	8.0
Thenns	1.0
Tully	3.0

dtype: float64

컬럼들을 각각 순회

```
In [152]: house_to_battle_count = Series()  
for col in battle_house.columns:
```

```
Out[152]:
```

Glover	2
--------	---

Name: attacker\_4, dtype: int64



## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ 각가문들이 출현한 횟수 집계

## Default값 바뀌었음을 의미

Default가 'object'에서 'float64'로 바뀌니 dtype을 명시하라는 의미입니다.

Frey	4.0
------	-----

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ house\_to\_battle\_count를 그래프 시각화



x축이 범주형 데이터인 경우에는 막대 그래프가 더 적합합니다.



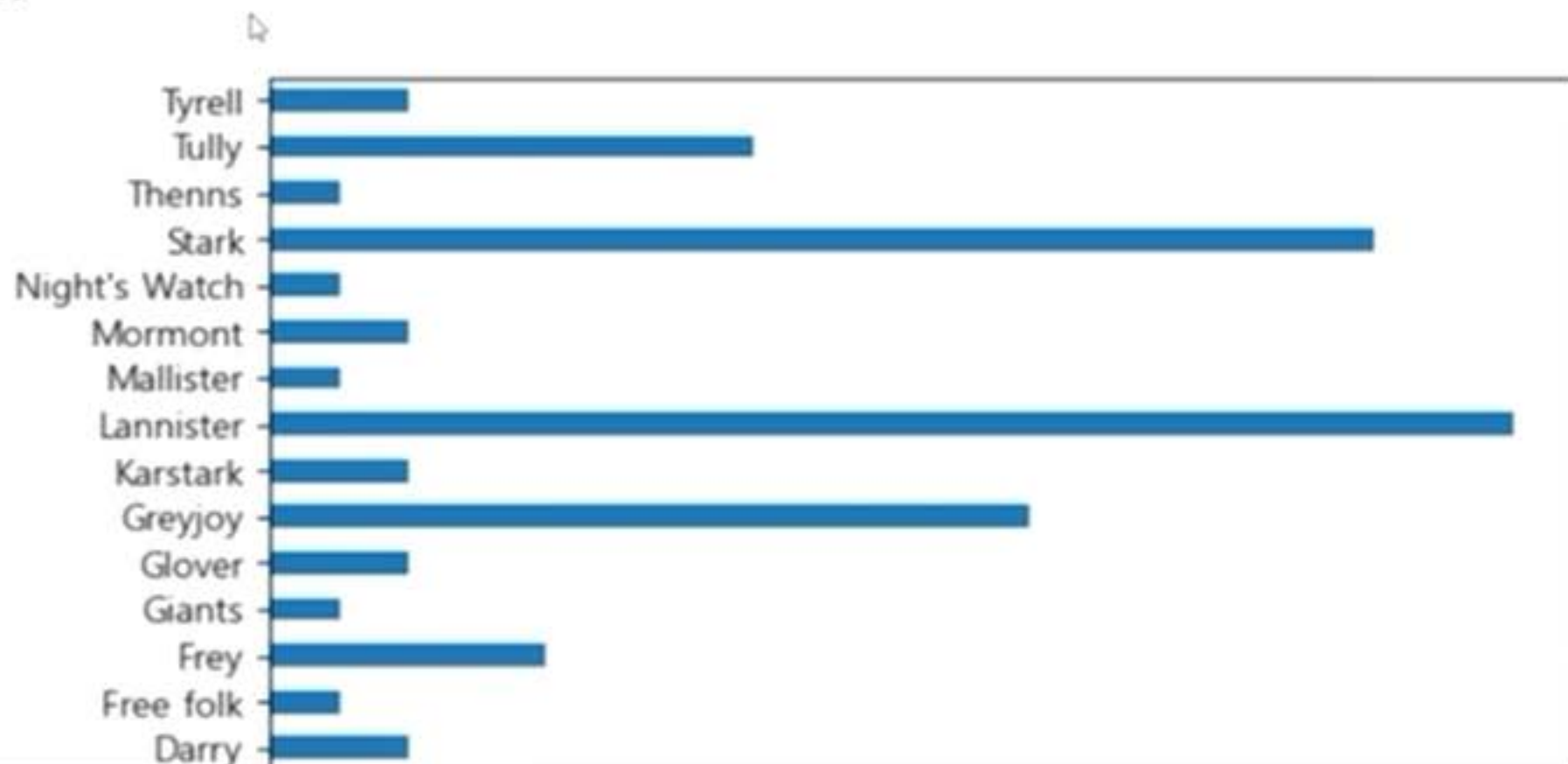
# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ house\_to\_battle\_count()를 그래프 시각화

```
In [159]: house_to_battle_count.plot(kind = 'barh')
```

```
Out[159]: <AxesSubplot:>
```



집계를 정렬해서 보고 싶은 경우, plot함수 수행 전에 정렬을 합니다.

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 가문별 전쟁 참여 횟수 분석 및 시각화

✓ house\_to\_battle\_count()를 그래프 시각화

```
In [160]: house_to_battle_count
```

```
house_to_battle_count.sort_values().plot(kind = 'barh')
```

5) (실습 #2) 년도별로 사망자의 숫자와 전 **sort\_values()** 오름차순 정렬

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

집계를 정렬해서 보고 싶은 경우, plot함수 수행 전에 정렬을 합니다.



02

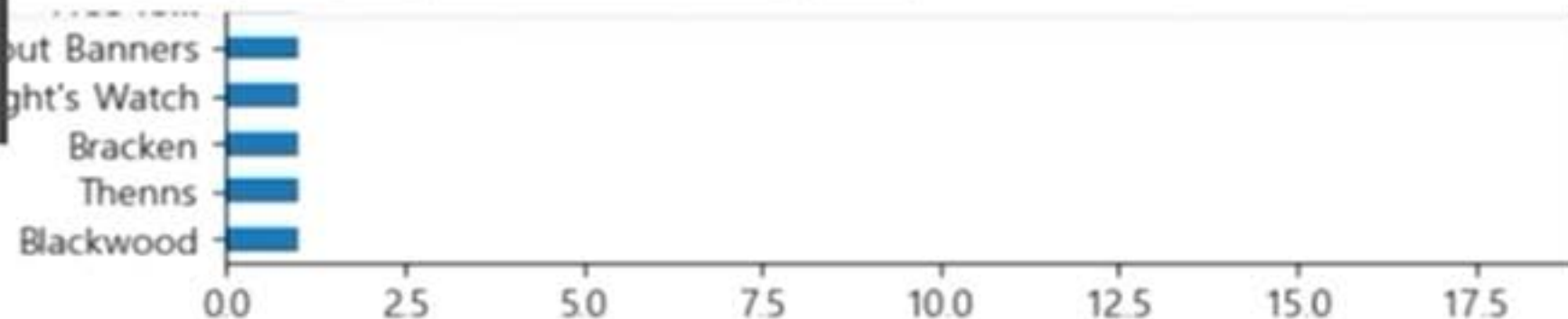
## 왕좌의 게임 데이터 분석 [ 4 ]



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

✓ 연도별 사망자 수 구하기



### 5) (실습 #2) 연도별로 사망자의 숫자와 전쟁이 벌어진 횟수 시각화

```
In [164]: #1. 연도별 사망자수 구하기  
death['Death Year'].value_counts()
```

```
Out[164]: 299.0    156  
          300.0    100  
          298.0     46  
          297.0     3  
          Name: Death Year, dtype: int64
```

```
In [ ]:
```

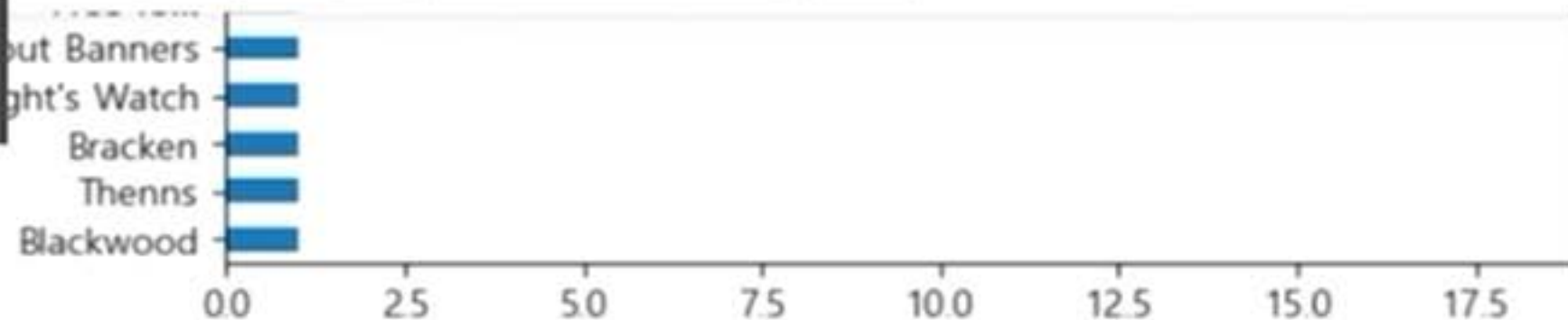
```
In [ ]:
```



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

✓ 연도별 사망자 수 구하기



`death['Death Year'].value_counts()`

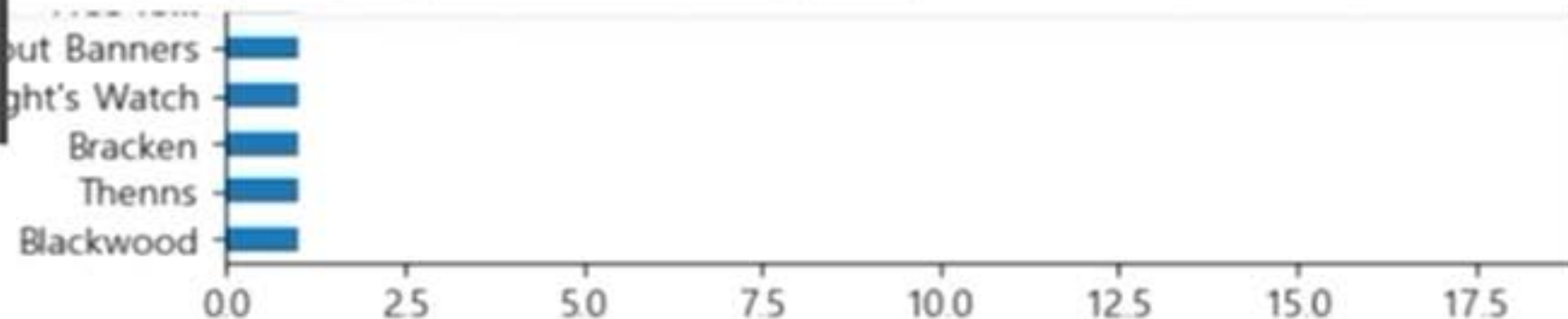
```
In [163]: #1. 연도별 사망자수 구하기  
death['Death Year'].value_counts()
```

```
Out[163]: 0      NaN  
1      299.0  
2      NaN  
3      300.0  
4      NaN  
...  
912     NaN  
913     300.0  
914     300.0  
915     300.0  
916     300.0
```

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

✓ 연도별 사망자 수 구하기



**sort\_index()**

연도별로 정렬

횟수 시각화

```
In [165]: #1. 연도별 사망자수 구하기  
death['Death Year'].value_counts().sort_index()
```

```
Out[165]: 297.0      3  
          298.0     46  
          299.0    156  
          300.0    100  
          Name: Death Year, dtype: int64
```

In [ ]:

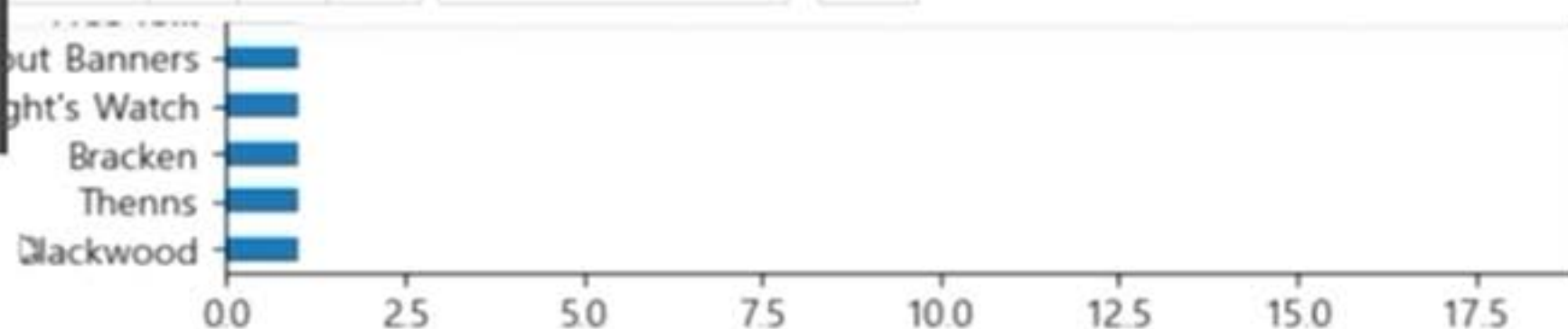
In [ ]:



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

✓ 연도별 전쟁 횟수 구하기



5) (실습 #2) 연도별로 사망자의 숫자와 전쟁이 벌어진 횟수 시각화

In [166]: `battles.year.value_counts()` 연도별 전쟁 횟수

In [\*]: `# 2. 연도별 전쟁횟수 구하기`  
`battles`

In [ ]:

In [ ]:

In [ ]:

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

✓ 연도별 전쟁 횟수 구하기

5) (실습 #2) 연도별로 사망자의 숫자와 전쟁이 벌어진 횟수 시각화

**pivot\_table 사용**

사망자 수 구하기

```
= death['Death Year'].value_counts().sort_index()
```

```
In [169]: # 2. 연도별 전쟁 횟수 구하기
# battles.year.value_counts()
battles.pivot_table(index = 'year', aggfunc = 'count',
                    values = '|')
```

```
Out[169]: 299    20
          300    11
          298     7
          Name: year, dtype: int64
```

In [ ]:

aggfunc을 적용할 values는 null값이 없는 컬럼으로 선택합니다.

In [ ]:

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

### ✓ 연도별 전쟁 횟수 구하기

```
In [168]: #1. 연도별 사망자수 구하기
```

```
death_for_year = death['Death Year'].value_counts().sort_index()
```

```
In [170]: battles.info()
```

```
In [171]: # 2. 연도별 전쟁횟수 구하기
```

```
#battles.year.value_counts()
```

```
battles.pivot_table(index = 'year', aggfunc = 'count',  
                     values = 'name')
```

```
Out[171]:
```

	name
year	
298	7
299	20
300	11

### pivot\_table 장점

자동으로 정렬되어 출력

### pivot\_table 단점

values의 null값이 없는  
컬럼을 찾아야 함



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

✓ 두 분석 결과를 하나의 데이터프레임으로 합치기

```
In [173]: # 3. 두 분석결과를 하나의 데이터프레임으로 합치기  
pd.concat([death_for_year, battle_for_year], axis = 1)
```

```
Out[173]: 297.0      3  
          298.0     46  
          299.0    156  
          300.0    100  
          Name: Death Year, dtype: int64
```

```
In [174]: battle_for_year
```

```
Out[174]:
```

	name
year	
298	7
299	20
300	11

pd.concat

데이터프레임 합치기

death_for_year	
297	3
298	46
299	156
300	100

battles_for_year	
298	7
299	20
300	11

컬럼 방향으로 합칠 것이기 때문에 axis=1로 합니다.

# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

✓ 두 분석 결과를 하나의 데이터프레임으로 합치기

```
In [178]: # 3. 두 분석결과를 하나의 데이터프레임으로 합치기  
data = pd.concat([death_for_year, battle_for_year], axis = 1)  
data.columns = ['# of death', '# of battles']
```

Out [178]:

	Death Year	name
297.0	3	NaN
298.0	46	7.0
299.0	156	20.0
300.0	100	11.0

컬럼명 변경

In [ ]:

# 왕좌의 게임 데이터 분석 및 시각화 실습

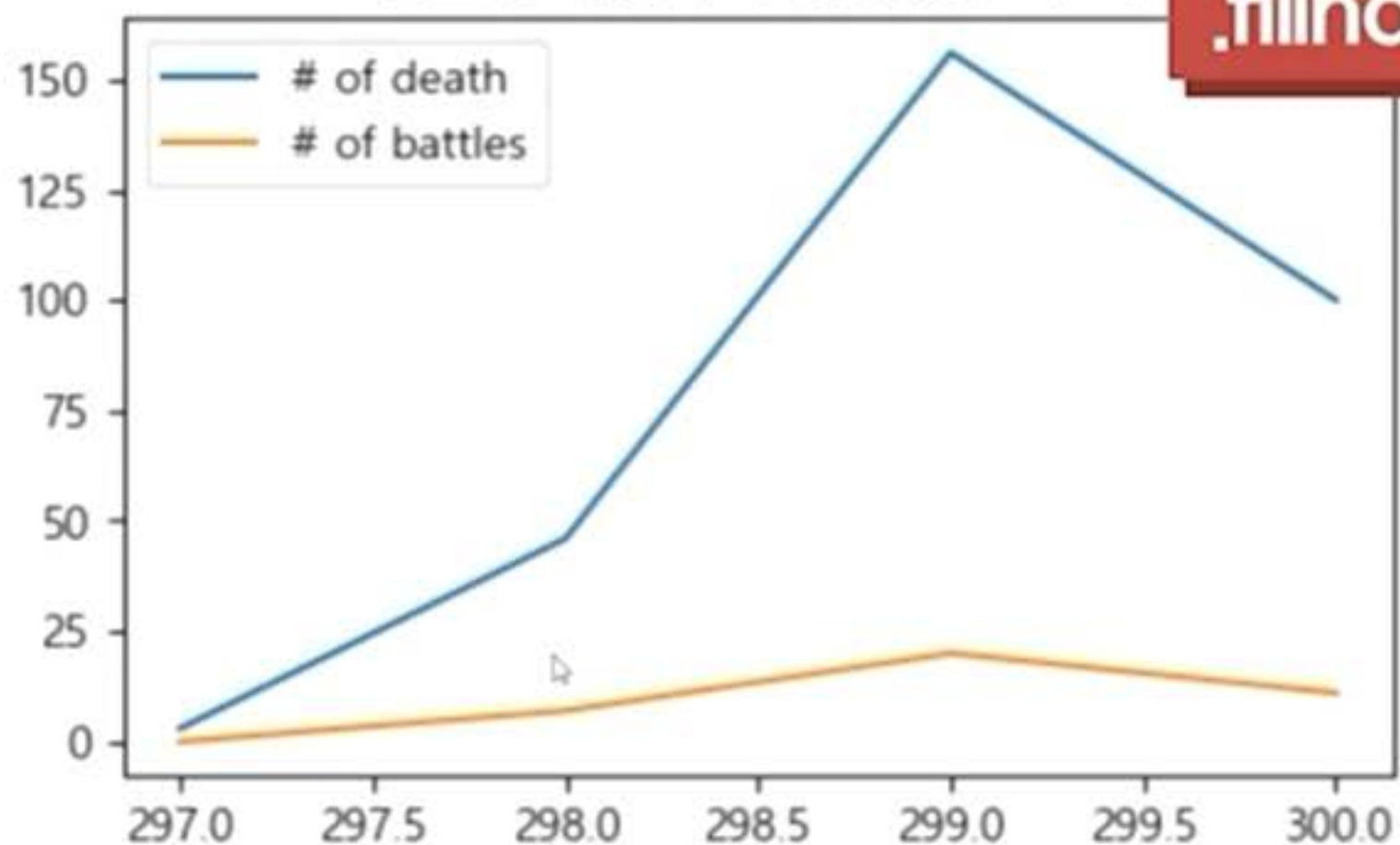
## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

Code

✓ 연도별 전쟁 횟수와 사망자 수 시각화

Out[186]: <AxesSubplot:title={'center': '년도별 사망

년도별 사망자와 전쟁횟수 추이



.fillna()

결측값 채움

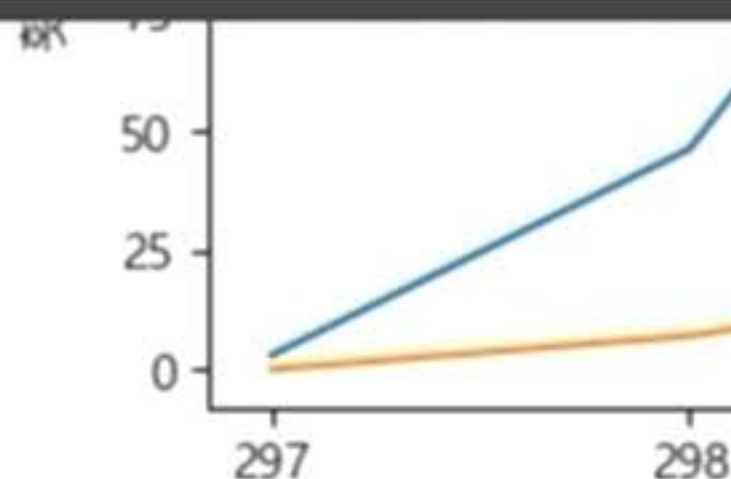
In [ ]:



# 왕좌의 게임 데이터 분석 및 시각화 실습

## ◎ 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화

✓ 연도별 전쟁 횟수와 사망자 수의 상관 관계 분석



`.corr()`

상관 관계 분석

```
In [189]: data.corr()
```

```
Out[189]:
```

	# of death	# of battles
# of death	1.00000	0.98991
# of battles	0.98991	1.00000

상관값이 1에 가까울수록 상관 관계가 높다는 의미입니다.

## 데이터 시각화



### 학습완료

- 1/ 왕좌의 게임 데이터 분석 [ 3 ]
  - 가문별 전쟁 참여 횟수 분석 및 시각화
- 2/ 왕좌의 게임 데이터 분석 [ 4 ]
  - 연도별 전쟁 횟수와 사망자 수 비교 분석 및 시각화