Class 13. KBO 데이터베이스

KBO 데이터베이스 테이블 구성

DB명
KBO.sqlite

데이터 출처
KBO 기록실

테이블 리스트
1) batting_old - 1982년~2001년 타자 기록
2) batting_new - 2002년 ~2021년 타자 기록
3) pitching_old - 1982년~2001년 투수 기록
4) pitching_new - 2002년 ~2021년 투수 기록

KBO 데이터베이스 테이블 구성

batting_old 테이블

yearID	RANK	Player	Team	AVG	G	PA	AB	H	2B
시즌연도	팀내타율순위	선수이름	소속팀	타율	경기	타석	타수	안타	2루타

3B	HR	RBI	SB	CS	BB	HBP	SO	GDP	E
3루타	홈런	타점	도루성공	도루실패	볼넷	사구	삼진	병살타	실책

batting_new 테이블

yearID	RANK	Player	Team	AVG	G	PA	AB	R	H
시즌연도	팀내타율순위	선수이름	소속팀	타율	경기	타석	타수	득점	안타

2B	3B	HR	TB	RBI	SAC	SF	BB	IBB	HBP
2루타	3루타	홈런	루타	타점	희생번트	희생플라이	볼넷	고의사구	사구

SO	GDP	SLG	OBP	OPS	MH	RISP	PH-BA
삼진	병살타	장타율	출루율	장타율+출루율	멀티히트	득점권타율	대타타율

KBO 데이터베이스 테이블 구성

pitching_old 테이블

yearID	RANK	Player	Team	ERA	G	CG	SHO	W	L
시즌연도	팀내ERA순위	선수이름	소속팀	평균자책점	경기	완투	완봉	승	패

SV	HLD	WPCT	TBF	IP	H	HR	BB	HBP	SO
세이브	홀드	승률	타자수	이닝	피안타	홈런	볼넷	사구	삼진

R	ER
실점	자책점

pitching_new 테이블

yearID	RANK	Player	Team	ERA	G	W	L	SV	HLD
시즌연도	팀내ERA순위	선수이름	소속팀	평균자책점	경기	승	패	세이브	홀드

WPCT	IP	H	HR	BB	HBP	SO	R	ER	WHIP
승률	이닝	피안타	홈런	볼넷	사구	삼진	실점	자책점	이닝당 출루허용률

CG	SHO	QS	BSV	TBF	NP	AVG	2B	3B	SAC
완투	완봉	퀄리티스타트	블론세이브	타자수	투구수	피안타율	2루타	3루타	희생번트

SF	IBB	WP	BK
희생플라이	고의사구	폭투	보크

KBO 데이터베이스 활용하기

이용규 선수 기록 조회하기

```
SELECT * FROM batting_new WHERE Player = '이용규';
```

yearID	RANK	Player	Team	AVG	G	PA	AB	R	H	2B	3B	HR	TB	RBI	SAC	SF	BB	IBB	HBP	SO	GDP	SLG	OBP	OPS	MH	RISP	PH-BA
2004	21	이용규	LG	0.129	52	70	62	3	8	1	0	0	9	2	1	1	4	0	2	21	0	0.145	0.203	0.348	1	0.154	0.143
2005	7	이용규	KIA	0.266	124	479	414	57	110	17	2	5	146	37	14	3	39	0	9	64	6	0.353	0.34	0.693	28	0.208	0.0
2006	2	이용규	KIA	0.318	125	552	485	78	154	25	9	1	200	39	7	1	50	3	9	48	6	0.412	0.391	0.803	46	0.25	0.0
2007	6	이용규	KIA	0.28	118	491	439	61	123	17	8	0	156	27	5	3	37	0	7	40	4	0.355	0.344	0.699	29	0.279	0.0
2008	1	이용규	KIA	0.312	106	473	417	62	130	24	6	0	166	38	6	0	47	1	3	37	4	0.398	0.385	0.783	41	0.314	0.0
2009	9	이용규	KIA	0.266	50	201	169	32	45	8	3	0	59	14	6	0	22	1	4	21	1	0.349	0.364	0.713	9	0.333	1.0
2010	3	이용규	KIA	0.307	129	555	472	74	145	19	1	3	175	51	7	3	64	2	9	50	10	0.371	0.398	0.769	38	0.353	0.0
2011	3	이용규	KIA	0.333	111	503	421	84	140	16	2	3	169	33	7	3	63	2	9	33	5	0.401	0.427	0.828	39	0.247	0.0
2012	6	이용규	KIA	0.283	125	580	491	86	139	14	2	2	163	37	10	3	66	0	10	38	9	0.332	0.377	0.709	38	0.292	0.0
2013	5	이용규	KIA	0.295	100	453	390	74	115	20	1	2	143	22	10	2	44	0	7	37	4	0.367	0.375	0.742	32	0.287	0.0
2014	7	이용규	한화	0.288	104	418	358	62	103	12	4	0	123	20	2	1	52	0	5	46	5	0.344	0.385	0.729	27	0.208	0.286
2015	2	이용규	한화	0.341	124	585	493	94	168	15	7	4	209	42	11	4	68	5	9	45	4	0.424	0.427	0.851	49	0.35	0.0
2016	3	이용규	한화	0.352	113	530	452	98	159	20	4	3	196	41	7	1	63	1	7	29	7	0.434	0.438	0.872	54	0.374	0.0
2017	13	이용규	한화	0.263	57	200	179	31	47	8	1	0	57	12	1	1	17	0	2	20	4	0.318	0.332	0.65	10	0.196	0.4
2018	6	이용규	한화	0.293	134	575	491	82	144	14	1	1	163	36	8	5	59	0	12	62	8	0.332	0.379	0.711	44	0.327	0.0
2020	5	이용규	한화	0.286	120	487	419	60	120	14	2	1	141	32	2	1	59	0	6	36	9	0.337	0.381	0.718	31	0.294	0.0
2021	3	이용규	키움	0.296	133	547	459	88	136	16	8	1	171	43	6	6	71	2	5	46	5	0.373	0.392	0.765	38	0.276	0.333

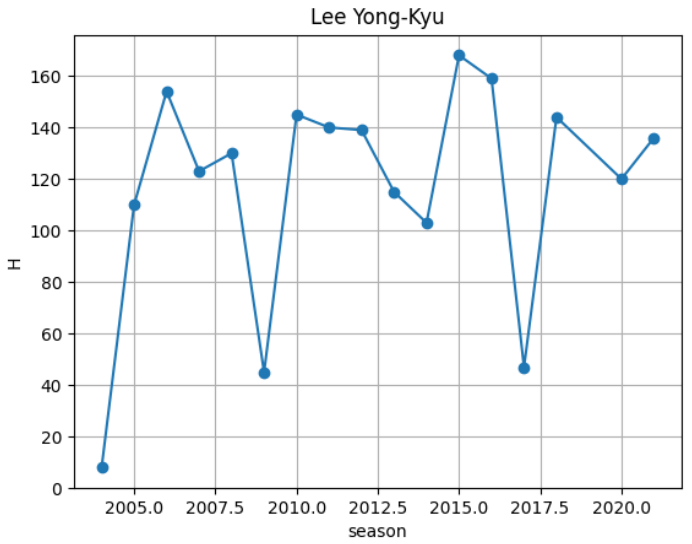
KBO 데이터베이스 활용하기

이용규 선수 안타 개수 추이 선 그래프 그리기 - 1

yearID	H
2004	8
2005	110
2006	154
2007	123
2008	130
2009	45
2010	145
2011	140
2012	139
2013	115
2014	103
2015	168
2016	159
2017	47
2018	144
2020	120
2021	136

```
SELECT yearID, H FROM batting_new WHERE Player = '이용규';
```

```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

with sqlite3.connect("KBO.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT yearID, H FROM batting_new WHERE Player = '이용규';
    ''')
    result = cur.fetchall()

cols = [column[0] for column in cur.description] # 컬럼명 가져오기

df = pd.DataFrame.from_records(data=result, columns=cols)

plt.plot(df['yearID'], df['H'], marker='o')
plt.title('Lee Yong-Kyu')
plt.xlabel('season')
plt.ylabel('H')
plt.grid(True)
plt.savefig('KBO1.png')
```



문제점!
연도가 소수점이 있게 표시 되었고
전체 연도가 보이지 않음

KBO 데이터베이스 활용하기

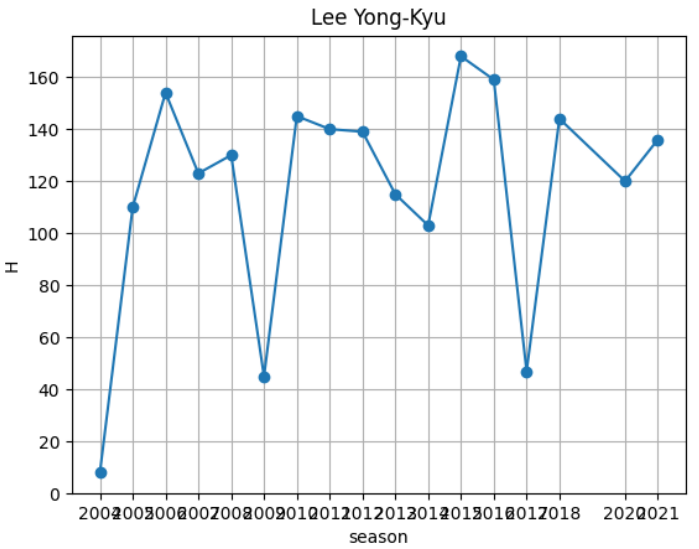
이용규 선수 안타 개수 추이 선 그래프 그리기 - 2

yearID	H
2004	8
2005	110
2006	154
2007	123
2008	130
2009	45
2010	145
2011	140
2012	139
2013	115
2014	103
2015	168
2016	159
2017	47
2018	144
2020	120
2021	136

```
SELECT yearID, H FROM batting_new WHERE Player = '이용규';
```

```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

with sqlite3.connect("KBO.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT yearID, H FROM batting_new WHERE Player = '이용규';
    ''')
    result = cur.fetchall()

cols = [column[0] for column in cur.description] # 컬럼명 가져오기

df = pd.DataFrame.from_records(data=result, columns=cols)

plt.plot(df['yearID'], df['H'], marker='o')
plt.title('Lee Yong-Kyu')
plt.xlabel('season')
plt.ylabel('H')
plt.xticks(df['yearID'].values)
plt.grid(True)
plt.savefig('KBO2.png')
```

추가!



문제점!
연도가 모두 보이지만, 겹쳐져서 보임

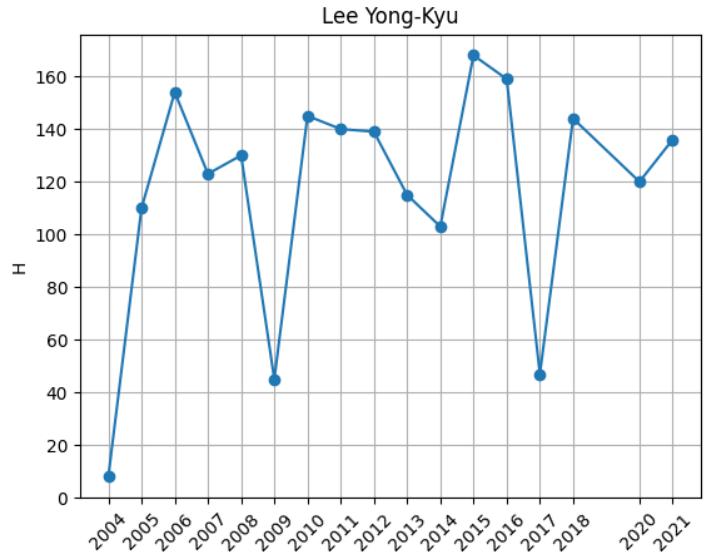
KBO 데이터베이스 활용하기

이용규 선수 안타 개수 추이 선 그래프 그리기 - 3

yearID	H
2004	8
2005	110
2006	154
2007	123
2008	130
2009	45
2010	145
2011	140
2012	139
2013	115
2014	103
2015	168
2016	159
2017	47
2018	144
2020	120
2021	136

```
SELECT yearID, H FROM batting_new WHERE Player = '이용규';
```

```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

with sqlite3.connect("KBO.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT yearID, H FROM batting_new WHERE Player = '이용규';
    ''')
    result = cur.fetchall()

cols = [column[0] for column in cur.description] # 컬럼명 가져오기

df = pd.DataFrame.from_records(data=result, columns=cols)

plt.plot(df['yearID'], df['H'], marker='o')
plt.title('Lee Yong-Kyu')
plt.xlabel('season')
plt.ylabel('H')
plt.xticks(df['yearID'].values, rotation=45) 추가!
plt.grid(True)
plt.savefig('KBO3.png')
```



KBO 데이터베이스 활용하기

이용규 선수 안타 개수 추이 선 그래프 그리기 – 4

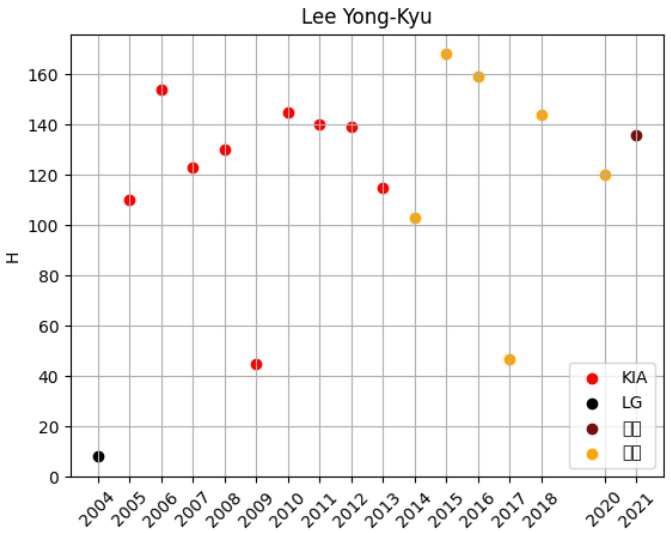
소속 팀 기준으로 데이터 포인트를 다른 색으로 나타내보자

yearID	Team	H
2004	LG	8
2005	KIA	110
2006	KIA	154
2007	KIA	123
2008	KIA	130
2009	KIA	45
2010	KIA	145
2011	KIA	140
2012	KIA	139
2013	KIA	115
2014	한화	103
2015	한화	168
2016	한화	159
2017	한화	47
2018	한화	144
2020	한화	120
2021	키움	136

```
SELECT yearID, Team, H FROM batting_new WHERE Player = '이용규';
```

```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

with sqlite3.connect("KBO.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT yearID, Team, H FROM batting_new WHERE Player = '이용규';
    ''')
    result = cur.fetchall()

cols = [column[0] for column in cur.description] # 컬럼명 가져오기

df = pd.DataFrame.from_records(data=result, columns=cols)

team_colors = {'LG':'black', 'KIA':'red', '한화':'orange', '키움':'#760c0c'}

group_df = df.groupby('Team')

for key, group in group_df:
    print(key)
    print(group)
    plt.scatter(group['yearID'], group['H'], marker='o', color=team_colors[key], label=key)

plt.title('Lee Yong-Kyu')
plt.xlabel('season')
plt.ylabel('H')
plt.xticks(df['yearID'].values, rotation=45)
plt.grid(True)
plt.legend()
plt.savefig('KBO4.png')
```

수정!



문제점!
한글이 깨짐

KBO 데이터베이스 활용하기

이용규 선수 안타 개수 추이 선 그래프 그리기 – 5

소속 팀 기준으로 데이터 포인트를 다른 색으로 나타내보자

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'NanumGothic'   추가!

with sqlite3.connect("KBO.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT yearID, Team, H FROM batting_new WHERE Player = '이용규';
    ''')
    result = cur.fetchall()

cols = [column[0] for column in cur.description] # 컬럼명 가져오기

df = pd.DataFrame.from_records(data=result, columns=cols)

team_colors = {'LG':'black', 'KIA':'red', '한화':'orange', '키움':'#760c0c'}

group_df = df.groupby('Team')

plt.figure(figsize=(10, 6))   추가!

for key, group in group_df:
    print(key)
    print(group)
    plt.scatter(group['yearID'], group['H'], marker='o', color=team_colors[key], label=key)

plt.title('이용규')
plt.xlabel('시즌')
plt.ylabel('안타')   수정!
plt.xticks(df['yearID'].values, rotation=45)
plt.grid(True)
plt.legend()
plt.savefig('KBO5.png')
```

한글 폰트 설치(터미널)

1. 나눔폰트 설치

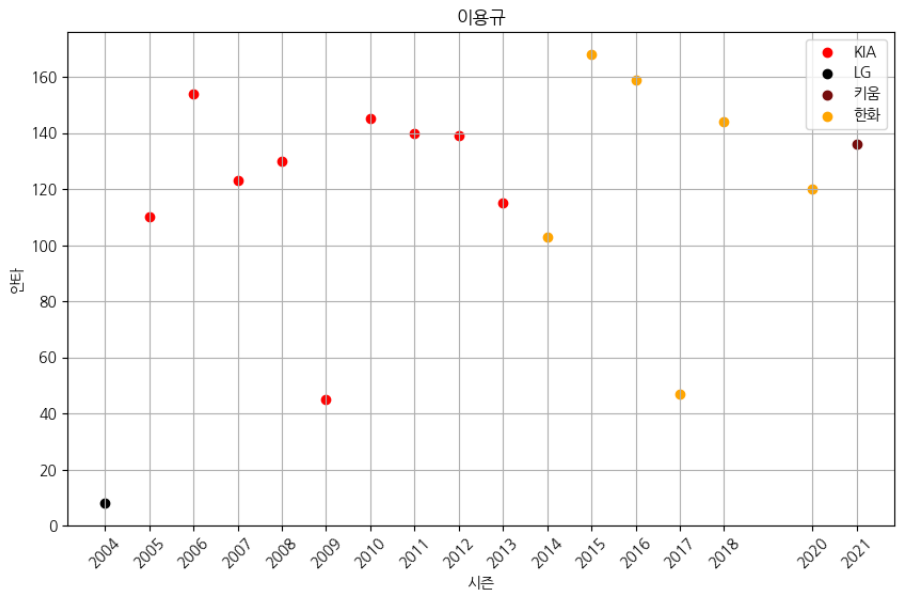
sudo apt-get install –y fonts-nanum fonts-nanum-coding fonts-nanum-extra

2. matplotlib 라이브러리 캐시 삭제

rm –rf ~/.cache/matplotlib/*

3. 폰트 캐시 재생성

fc-cache -fv



KBO 데이터베이스 활용하기

2021년 홈런 TOP 10 조회하기

그룹의 조건을 두 개로 지정해줘야 함
동명이인이 있을 수 있기 때문에

```
SELECT Team, Player, SUM(HR) FROM batting_new WHERE yearID = 2021 GROUP BY Team, Player ORDER BY SUM(HR) DESC LIMIT 10;
```

```
Team  Player   SUM(HR)
----  ------   -------
SSG   최정          35
NC    나성범        33
NC    알테어        32
SSG   한유섬        31
NC    양의지        30
삼성  피렐라        29
두산  양석환        28
두산  김재환        27
삼성  오재일        25
삼성  구자욱        22
```

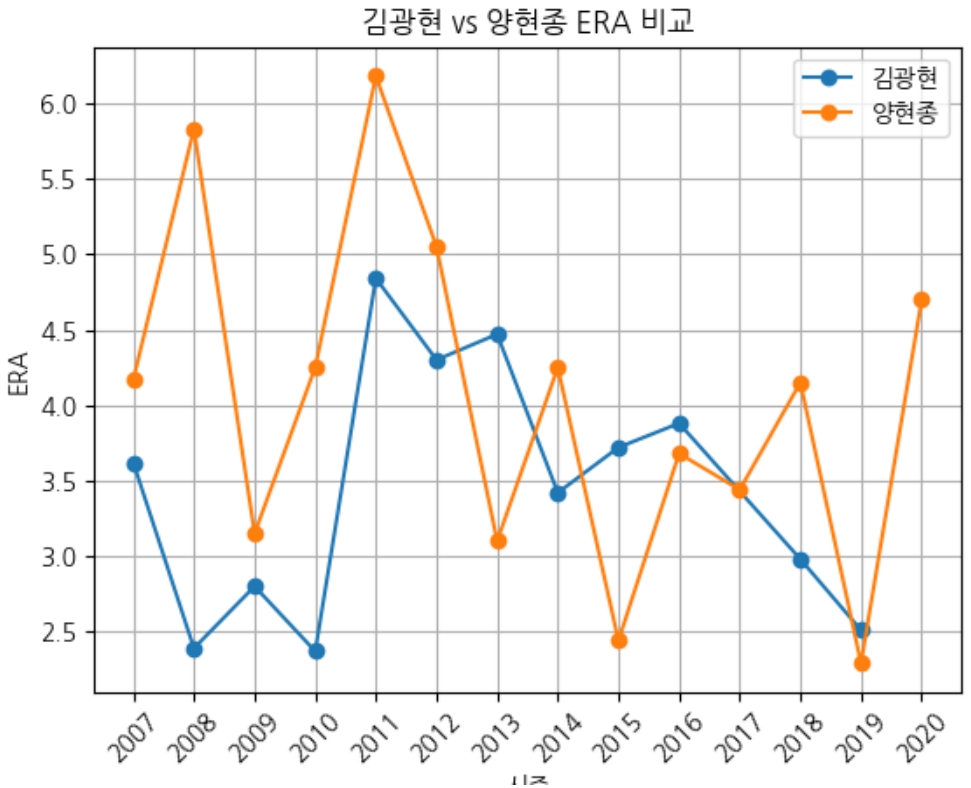
순위	선수명	팀명	AVG	G	PA	AB	R	H	2B	3B	HR	TB	RBI	SAC	SF
1	최정	SSG	0.278	134	555	436	92	121	17	1	35	245	100	1	12
2	나성범	NC	0.281	144	623	570	96	160	29	1	33	290	101	0	4
3	알테어	NC	0.272	143	565	492	83	134	19	2	32	253	84	0	5
4	한유섬	SSG	0.278	135	519	442	71	123	18	1	31	236	95	1	6
5	양의지	NC	0.325	141	570	480	81	156	29	2	30	279	111	0	10
6	피렐라	삼성	0.286	140	621	553	102	158	25	2	29	274	97	0	3
7	양석환	두산	0.273	133	546	488	66	133	22	0	28	239	96	0	7
8	김재환	두산	0.274	137	566	475	86	130	23	2	27	238	102	0	5
9	오재일	삼성	0.285	120	484	418	64	119	20	0	25	214	97	0	8
10	박동원	키움	0.249	131	481	413	61	103	21	0	22	190	83	5	3
10	구자욱	삼성	0.306	139	610	543	107	166	30	10	22	282	88	1	12

출처: KBO 기록실

KBO 데이터베이스 활용하기

김광현, 양현종 데이터 비교(ERA)

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'NanumGothic'

with sqlite3.connect("KBO.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT * FROM pitching_new WHERE Player IN ('양현종', '김광현');
    ''')
    result = cur.fetchall()

cols = [column[0] for column in cur.description] # 컬럼명 가져오기

df = pd.DataFrame.from_records(data=result, columns=cols)

df_kim = df[df['Player']=='김광현']
df_yang = df[df['Player']=='양현종']

plt.plot(df_kim['yearID'], df_kim['ERA'], marker='o', label='김광현')
plt.plot(df_yang['yearID'], df_yang['ERA'], marker='o', label='양현종')
plt.grid(True)
plt.legend()
plt.title('김광현 vs 양현종 ERA 비교')
plt.xlabel('시즌')
plt.ylabel('ERA')
plt.xticks(list(range(2007, 2021)), rotation=45)
plt.savefig('kim_yang_compare1.png')
```



김광현이 양현종보다 ERA가 좋았던 것은 총 몇 회?

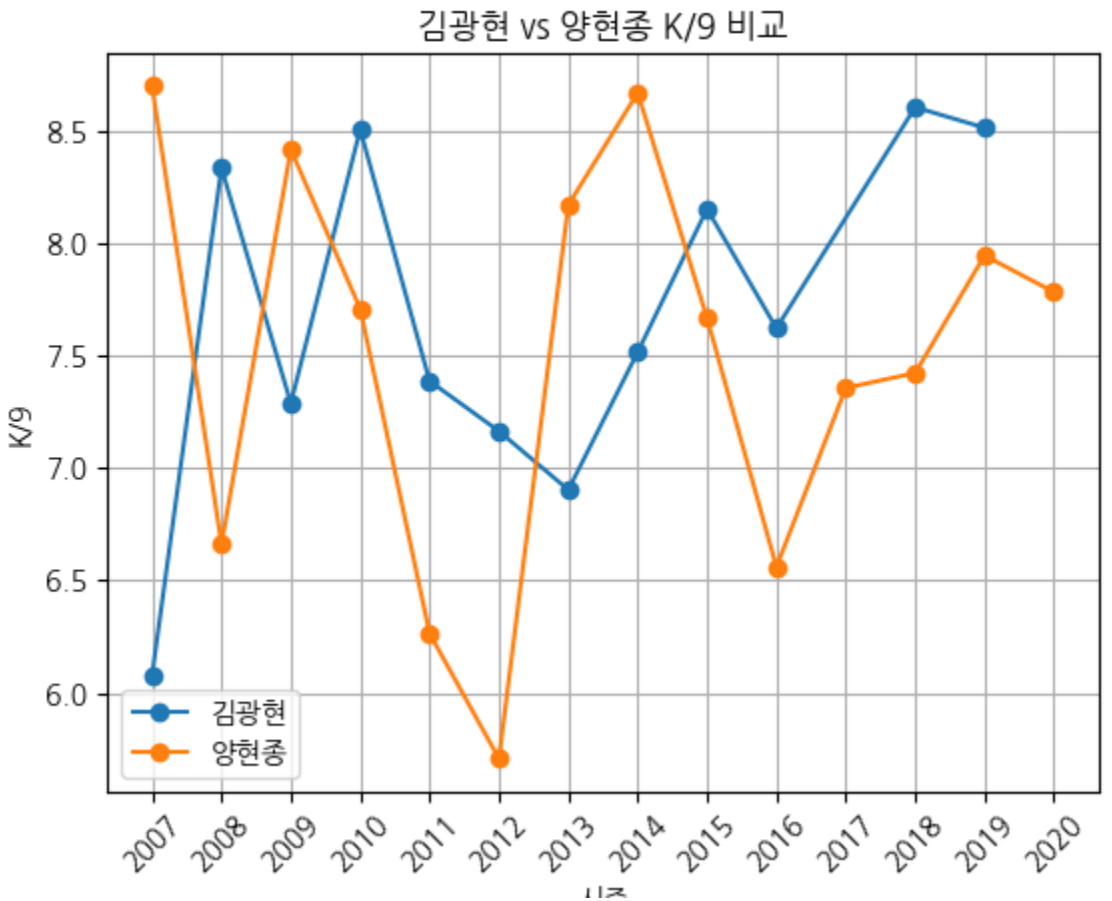
함께 시즌을 소화한 횟수 13시즌
김광현이 양현종보다 ERA가 좋았던 시즌은 총 8시즌

김광현의 ERA가 양현종보다 좋을 확률 = 8/13 = 62%

KBO 데이터베이스 활용하기

김광현, 양현종 데이터 비교(K/9)

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'NanumGothic'

with sqlite3.connect("KBO.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT * FROM pitching_new WHERE Player IN ('양현종', '김광현');
    ''')
    result = cur.fetchall()

cols = [column[0] for column in cur.description] # 컬럼명 가져오기

df = pd.DataFrame.from_records(data=result, columns=cols)

df['K/9'] = df['SO'] * 9 / df['IP']

df_kim = df[df['Player']=='김광현']
df_yang = df[df['Player']=='양현종']

plt.plot(df_kim['yearID'], df_kim['K/9'], marker='o', label='김광현')
plt.plot(df_yang['yearID'], df_yang['K/9'], marker='o', label='양현종')
plt.grid(True)
plt.legend()
plt.title('김광현 vs 양현종 K/9 비교')
plt.xlabel('시즌')
plt.ylabel('K/9')
plt.xticks(list(range(2007, 2021)), rotation=45)
plt.savefig('kim_yang_compare2.png')
```



김광현이 양현종보다 K/9가 좋았던 것은 총 몇 회?

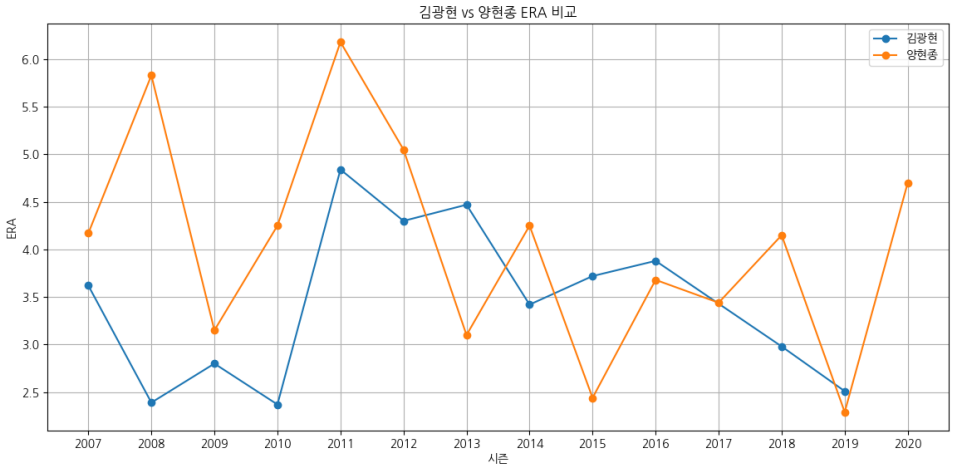
함께 시즌을 소화한 횟수 13시즌
김광현이 양현종보다 K/9가 좋았던 시즌은 총 8시즌

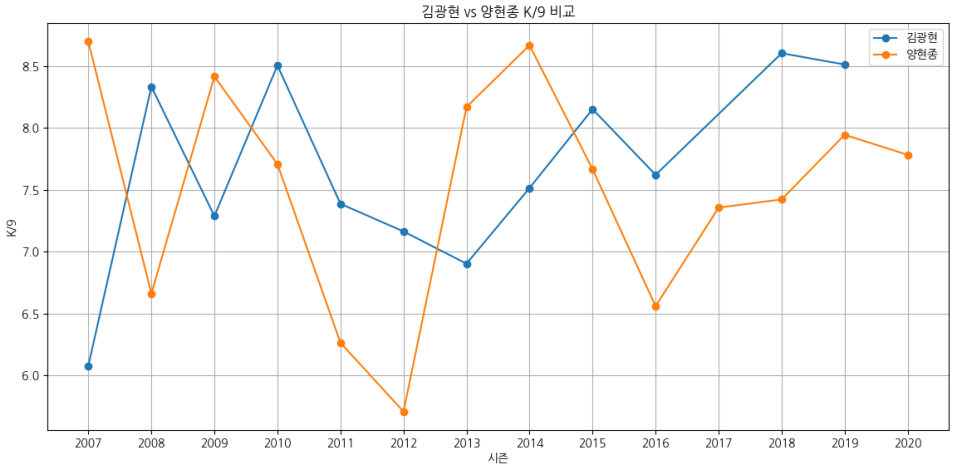
김광현이 양현종보다 삼진을 잘 잡을 확률 = 8/13 = 62%

```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'NanumGothic'

with sqlite3.connect("KBO.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT * FROM pitching_new WHERE Player IN ('양현종', '김광현');
    ''')
    result = cur.fetchall()

cols = [column[0] for column in cur.description] # 컬럼명 가져오기

df = pd.DataFrame.from_records(data=result, columns=cols)

df['K/9'] = df['SO'] * 9 / df['IP']

df_kim = df[df['Player']=='김광현']
df_yang = df[df['Player']=='양현종']

fig = plt.figure(figsize=(14, 14))

ax1 = fig.add_subplot(2, 1, 1)
ax2 = fig.add_subplot(2, 1, 2)

ax1.plot(df_kim['yearID'], df_kim['ERA'], marker='o', label='김광현')
ax1.plot(df_yang['yearID'], df_yang['ERA'], marker='o', label='양현종')

ax2.plot(df_kim['yearID'], df_kim['K/9'], marker='o', label='김광현')
ax2.plot(df_yang['yearID'], df_yang['K/9'], marker='o', label='양현종')

ax1.xaxis.set_ticks(list(range(2007, 2021)))
ax1.set_title('김광현 vs 양현종 ERA 비교')
ax1.set_xlabel('시즌')
ax1.set_ylabel('ERA')
ax1.legend(loc='best')
ax1.grid(True)

ax2.xaxis.set_ticks(list(range(2007, 2021)))
ax2.set_title('김광현 vs 양현종 K/9 비교')
ax2.set_xlabel('시즌')
ax2.set_ylabel('K/9')
ax2.legend(loc='best')
ax2.grid(True)

plt.savefig('kim_yang_compare3.png')
```

KBO 데이터베이스 활용하기

김광현, 양현종 데이터 비교
(ERA와 K/9를 2개의 서브플롯으로 그리기)





KBO 데이터베이스 활용하기

최정 홈런, 타율 그래프 그리기
(값의 범위가 다른 두 개 그래프 함께 그리기)

```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'NanumGothic'

with sqlite3.connect("KBO.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT yearID, AVG, HR FROM batting_new WHERE Player = '최정';
    ''')
    result = cur.fetchall()

cols = [column[0] for column in cur.description] # 컬럼명 가져오기

df = pd.DataFrame.from_records(data=result, columns=cols)

fig = plt.figure(figsize=(14, 14))

ax1 = fig.add_subplot(1, 1, 1)

ax1.plot(df['yearID'], df['AVG'], marker='o', label='타율')
ax2 = ax1.twinx()

ax2.plot(df['yearID'], df['HR'], marker='*', label='홈런', color='red')

ax1.xaxis.set_ticks(list(range(2005, 2021)))
ax1.set_title('최정 시즌 타율과 홈런수')
ax1.set_xlabel('시즌')
ax1.set_ylabel('타율')
ax1.legend(loc='upper left')
ax1.grid(True)

ax2.set_ylabel('홈런')
ax2.legend(loc='upper right')

plt.savefig('choi_jung.png')
```