2023 KBO 타자 OPS 예측하기

PLAYDATA AI Boot Camp 22기
6조 :: 송소정, 오유경, 전희진

주제 선정 및 문제 정의

탐색적 데이터 분석

- 데이터 훑어보기
- 데이터 클렌징
- 데이터 시각화
- 상관관계 분석

데이터 전처리

베이스라인 모델 구축

성능 올리기

- Feature Engineering
- Grid Search

결론

- 최종 예측 및 성능 평가
- 프로젝트 리뷰

주제 선정 및 문제 정의

“

야구는 … (중략) …
게임의 특성상 많은 부분이 기록돼 있어
분석 가치가 높다.

”

-한정섭, 정다현 and 김성준. (2022). 머신러닝을 활용한 빅데이터 분석을 통해 KBO 타자의 OPS 예측. 차세대융합기술학회논문지, 6(1), 12-18.

왜 OPS인가?

$$\text{타율(BA)} = \frac{\text{안타수}}{\text{타수}}$$

1B, 2B, 3B, 홈런 모두 '안타'로만 나타냄
4B, 사구 가치 적용되지 않음

↓ 보완

$$\text{장타율} = \frac{1\text{루타}_{*1} + 2\text{루타}_{*2} + 3\text{루타}_{*3} + \text{홈런}_{*4}}{\text{타수}}$$

안타의 종류에 따라 가중치 부여
4B, 사구 가치 적용되지 않음

↓ 4B, 사구 가치 살펴보기

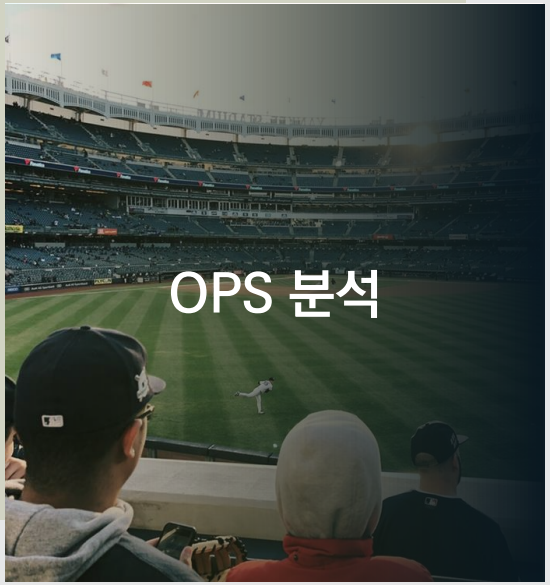
$$\text{출루율} = \frac{\text{안타수}+\text{볼넷}+\text{사구}}{\text{타수}+\text{볼넷}+\text{사구}+\text{희생플라이}}$$

즉, 장타율과 출루율은 야구 득점 메커니즘의 기본인 전진, 출루로 이해할 수 있음

피트 파머가 처음 제시한 개념인 OPS(On-base Plus Slugging)

OPS = 장타율 + 출루율

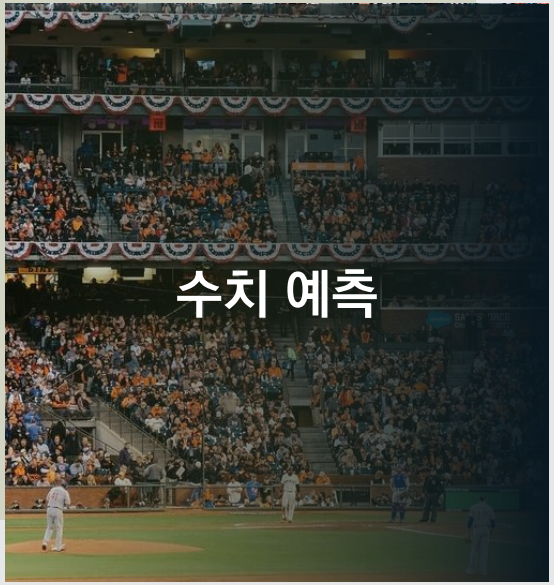
-조영석 and 조용주. (2005). 한국 프로야구에서 OPS와 득점에 관한 연구. Journal of The Korean Data Analysis Society, 7(1), 221-231.



01 statiz 데이터를 활용한 머신러닝

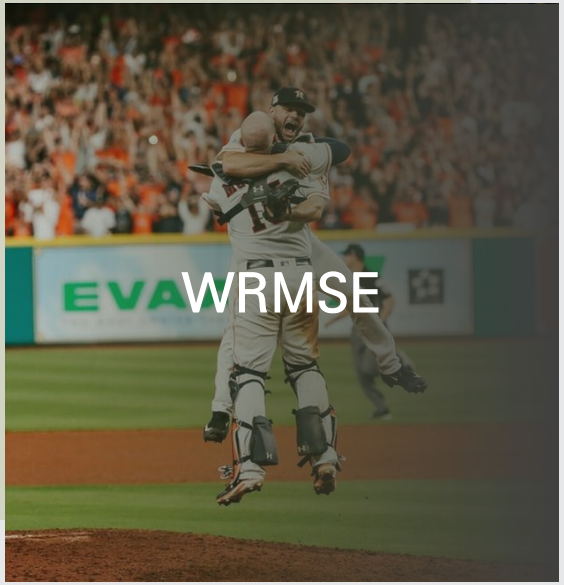
상관관계가 높은 특성들로
2023년 KBO 타자 OPS를 예측합니다.

*OPS : On-base Plus Slugging.
야구에서 타자들을 평가하는 스탯 중 하나로
'출루율 + 장타율'로 계산한다.



02 다양한 모델을 사용해 최적의 기법 찾기

랜덤포레스트, XGBoost 등 수치 예측 문제에
활용되는 다양한 알고리즘을 적용하고
실제 결과를 도출해봄으로써
강의 내용을 실제에 적용해봅니다.



03 WRMSE를 활용한 성능 평가

다양한 머신러닝 기법의
성능을 평가하는 지표로 WRMSE를
사용, 가장 좋은 성능을 내는 모델을
찾아냅니다.

사용 데이터 출처: Dacon , statiz

탐 색 적
데 이 터 분 석

데이터 훑어보기

```
# statiz에서 1982년부터 2022년까지의 타자 데이터를 크롤링 후 csv로 저장한 데이터
# 10023 열 x 26 행으로 이루어져 있음을 알 수 있다
```

```
In [19]: reg.tail()
```

Out[19]:

Target

	선수	G	타석	타수	득점	안타	2타	3타	홈런	루타	...	희타	희비	타율	출루	장타	OPS	wOBA	wRC+	WAR+	생일
10018	김헌곤22삼CF	80	239	224	18	43	8	0	1	54	...	2	3	0.192	0.224	0.241	0.465	0.217	25.5	-1.55	1988-11-09
10019	김영진99삼C	97	212	189	8	26	4	0	2	36	...	12	0	0.138	0.185	0.190	0.376	0.166	-16.1	-1.60	1972-04-09
10020	나균안18롯C	106	203	177	15	22	3	0	2	31	...	9	0	0.124	0.201	0.175	0.376	0.175	-15.1	-1.71	1998-03-16
10021	이종범07KRF	84	282	253	23	44	6	0	1	53	...	10	4	0.174	0.217	0.209	0.426	0.199	8.0	-2.04	1970-08-15
10022	권두조86청1B	108	358	303	14	49	6	1	0	57	...	25	0	0.162	0.237	0.188	0.425	0.198	17.3	-2.47	1951-06-10

5 rows × 28 columns

KBO 기록실(https://www.koreabaseball.com/Record/Player/HitterBasic/Basic1.aspx)

데이터 클렌징

```
# ‘선수’ 칼럼의 데이터를 각각 ‘이름’, ‘시즌’, ‘소속’, ‘포지션’으로 분리
# 개명한 선수, 생일 정보 누락 데이터 등 수정
# 이름+생일 로 선수 별 고유 아이디 부여
# ‘선수’ 칼럼 삭제 후 기존 칼럼들 float으로 type 변경
```

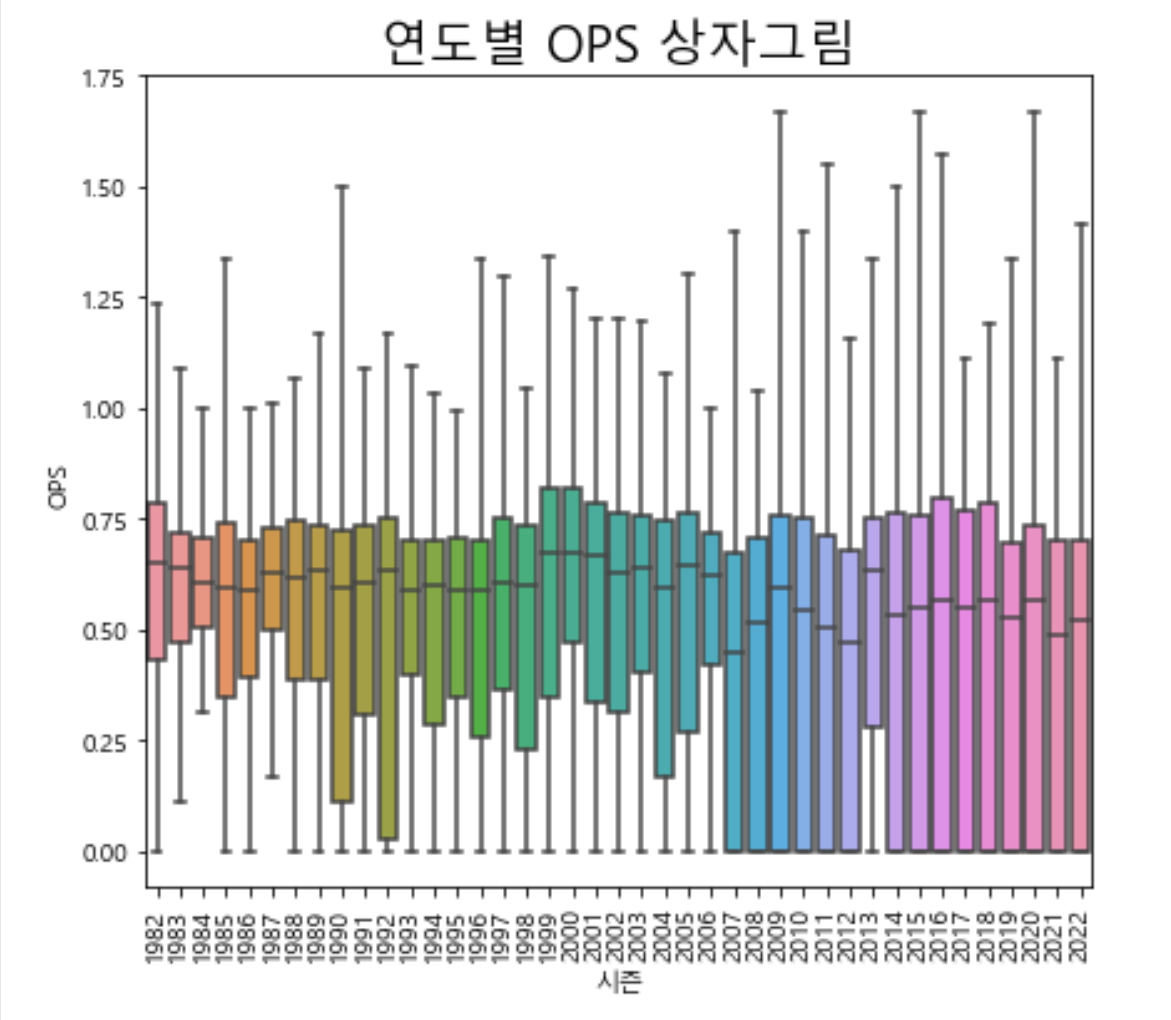
데이터 클렌징

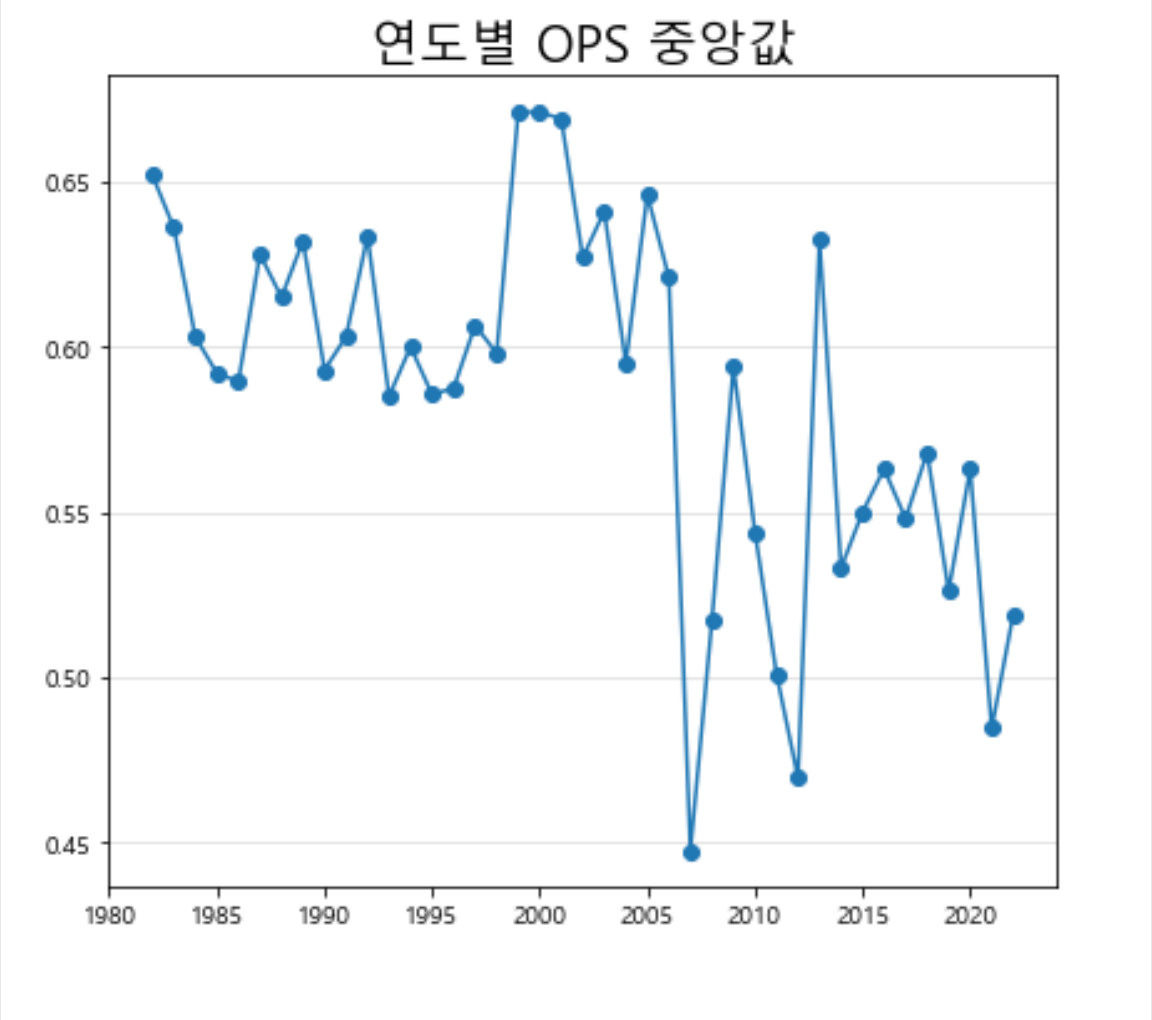
Columns 정렬 후의 모습

	ID	이름	생일	팀	시즌	포지션	나이	G	타석	타수	...	병살	희타	희비	타율	출루	장타	OPS	wOBA	wRC+	WAR+
0	0	이종범	1970-08-15	해	1994	SS	25	124.0	561.0	499.0	...	2.0	1.0	4.0	0.393	0.452	0.581	1.033	0.462	198.3	11.77
1	1	테임즈	1986-11-10	N	2015	1B	30	142.0	595.0	472.0	...	7.0	0.0	7.0	0.381	0.498	0.790	1.288	0.530	222.3	10.71
2	2	심정수	1975-05-05	현	2003	RF	29	133.0	601.0	460.0	...	14.0	0.0	8.0	0.335	0.478	0.720	1.197	0.498	210.7	10.19
3	0	이종범	1970-08-15	해	1997	SS	28	125.0	577.0	484.0	...	9.0	0.0	3.0	0.324	0.428	0.581	1.009	0.431	173.2	9.70
4	0	이종범	1970-08-15	해	1996	SS	27	113.0	525.0	449.0	...	4.0	0.0	2.0	0.332	0.425	0.566	0.991	0.440	184.6	9.52
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
10018	394	김헌곤	1988-11-09	삼	2022	CF	35	80.0	239.0	224.0	...	10.0	2.0	3.0	0.192	0.224	0.241	0.465	0.217	25.5	-1.55
10019	1024	김영진	1972-04-09	삼	1999	C	28	97.0	212.0	189.0	...	7.0	12.0	0.0	0.138	0.185	0.190	0.376	0.166	-16.1	-1.60
10020	2109	나종덕	1998-03-16	롯	2018	C	21	106.0	203.0	177.0	...	2.0	9.0	0.0	0.124	0.201	0.175	0.376	0.175	-15.1	-1.71
10021	0	이종범	1970-08-15	K	2007	RF	38	84.0	282.0	253.0	...	6.0	10.0	4.0	0.174	0.217	0.209	0.426	0.199	8.0	-2.04
10022	466	권두조	1951-06-10	청	1986	1B	36	108.0	358.0	303.0	...	5.0	25.0	0.0	0.162	0.237	0.188	0.425	0.198	17.3	-2.47

10023 rows × 33 columns

데이터 시각화

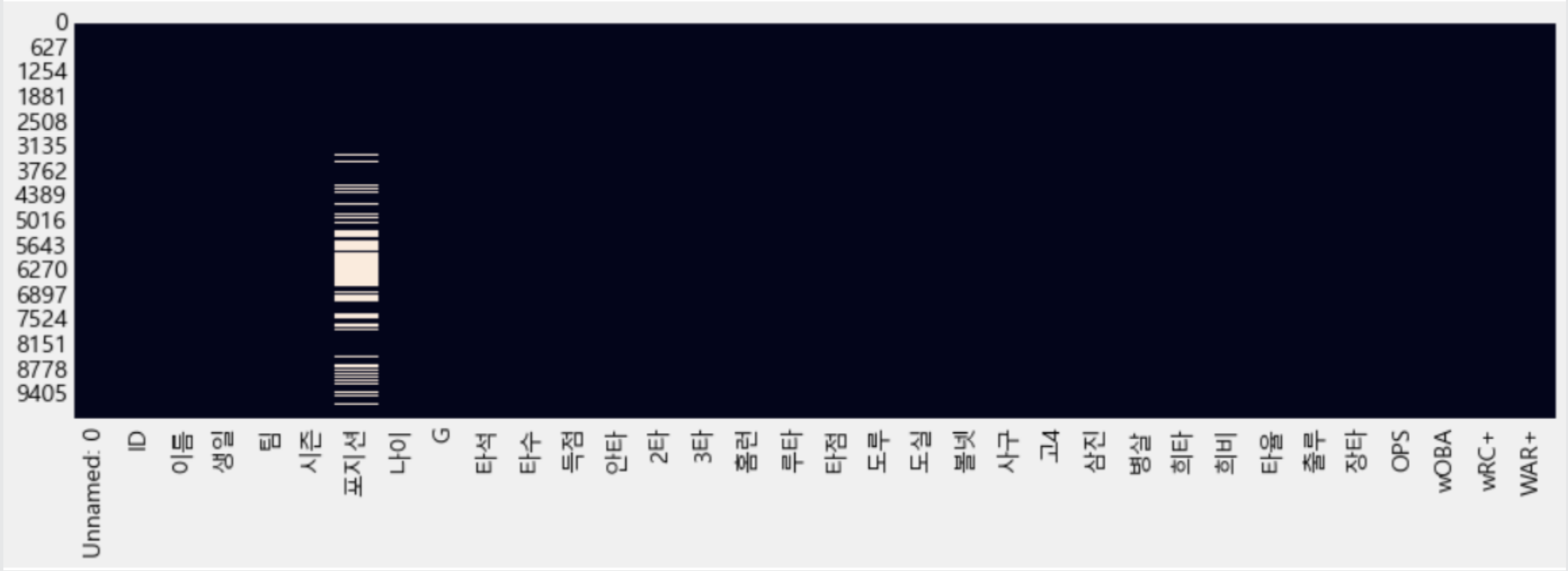

연도별 OPS 상자그림
OPS
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022
시즌


연도별 OPS 중앙값
0.65
0.60
0.55
0.50
0.45
1980 1985 1990 1995 2000 2005 2010 2015 2020

데이터 시각화

```
In [47]: plt.rcParams['figure.figsize']=(14,4)
         sns.heatmap(regular_season_df.isnull(),cbar=False)
         regular_season_df.isnull().sum()
```

결측치 찾기

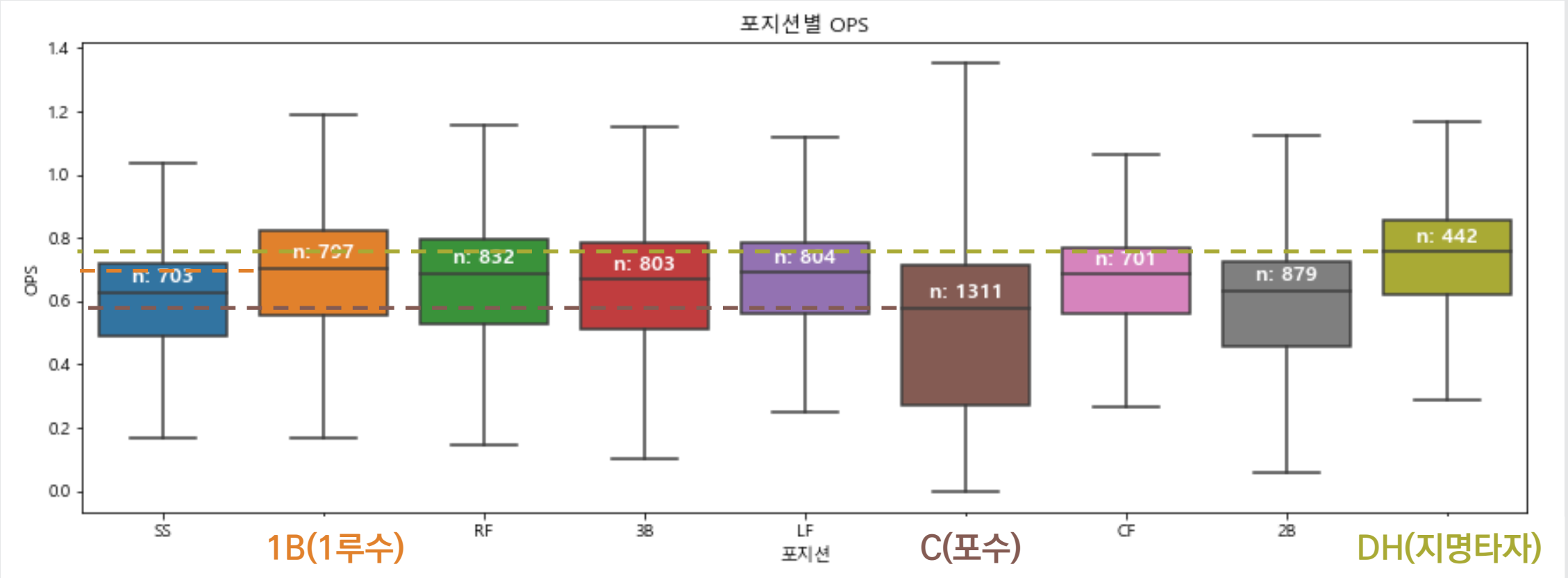


데이터 시각화

결측치 비율	처리 방법
10% 미만	제거 or 어떠한 방법이든지 상관없이 Imputation
10% 이상 20% 미만	hot deck , regression , model based method
20% 이상	model based method , regression

Hair et al. (2006, pp. 49-73)

데이터 시각화



DH : 수비에 전혀 관여하지 않는 지명타자
1루수: 수비 난이도가 낮고, 체격이 좋은 선수들이 단순 포구에 더 유리함

포수: 상대적으로 수비에 신경을 많이 써야하므로 OPS가 낮게 나옴

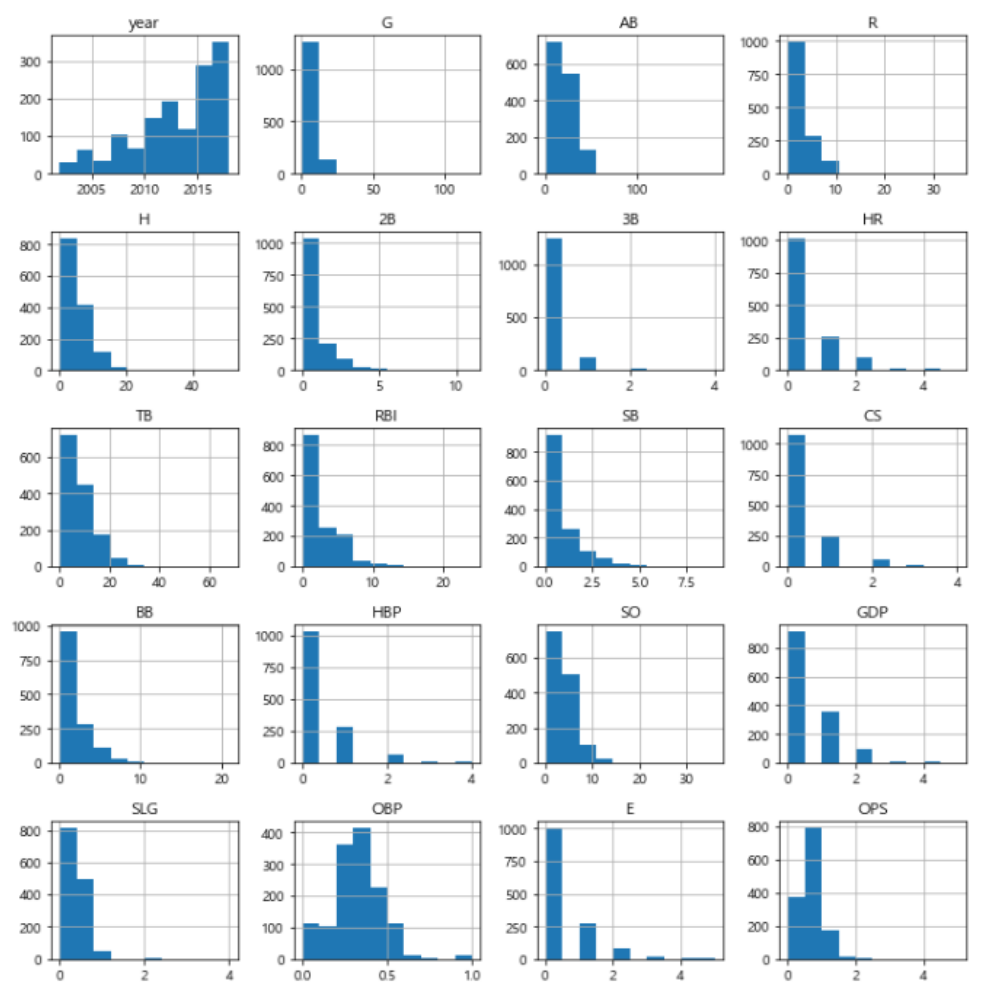
데이터 시각화

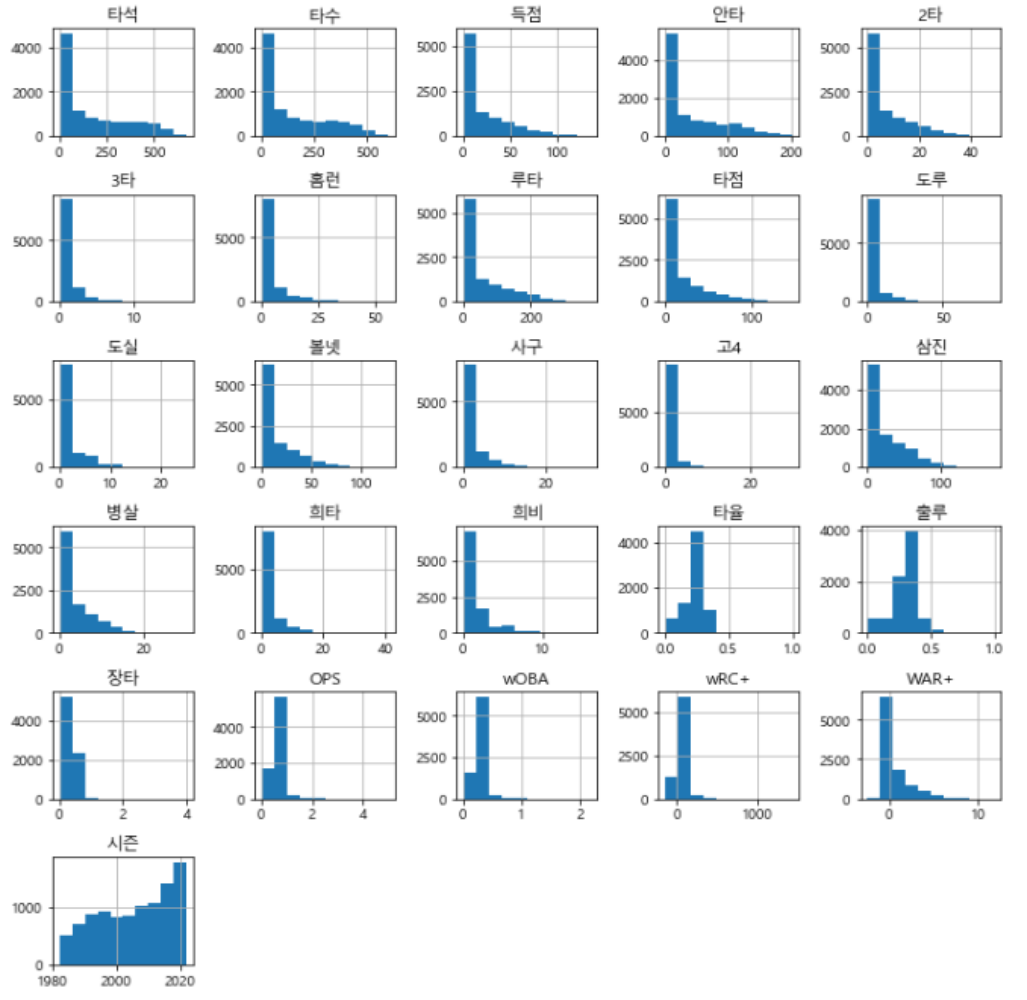
시즌 별 평균 타율 막대그래프

```
In [89]: regular_season_df['타율'].groupby(regular_season_df['시즌']).mean().plot.bar(color='yellowgreen')
Out[89]: <AxesSubplot:xlabel='시즌'>
```

상관관계(1) pre시즌 – regular 시즌

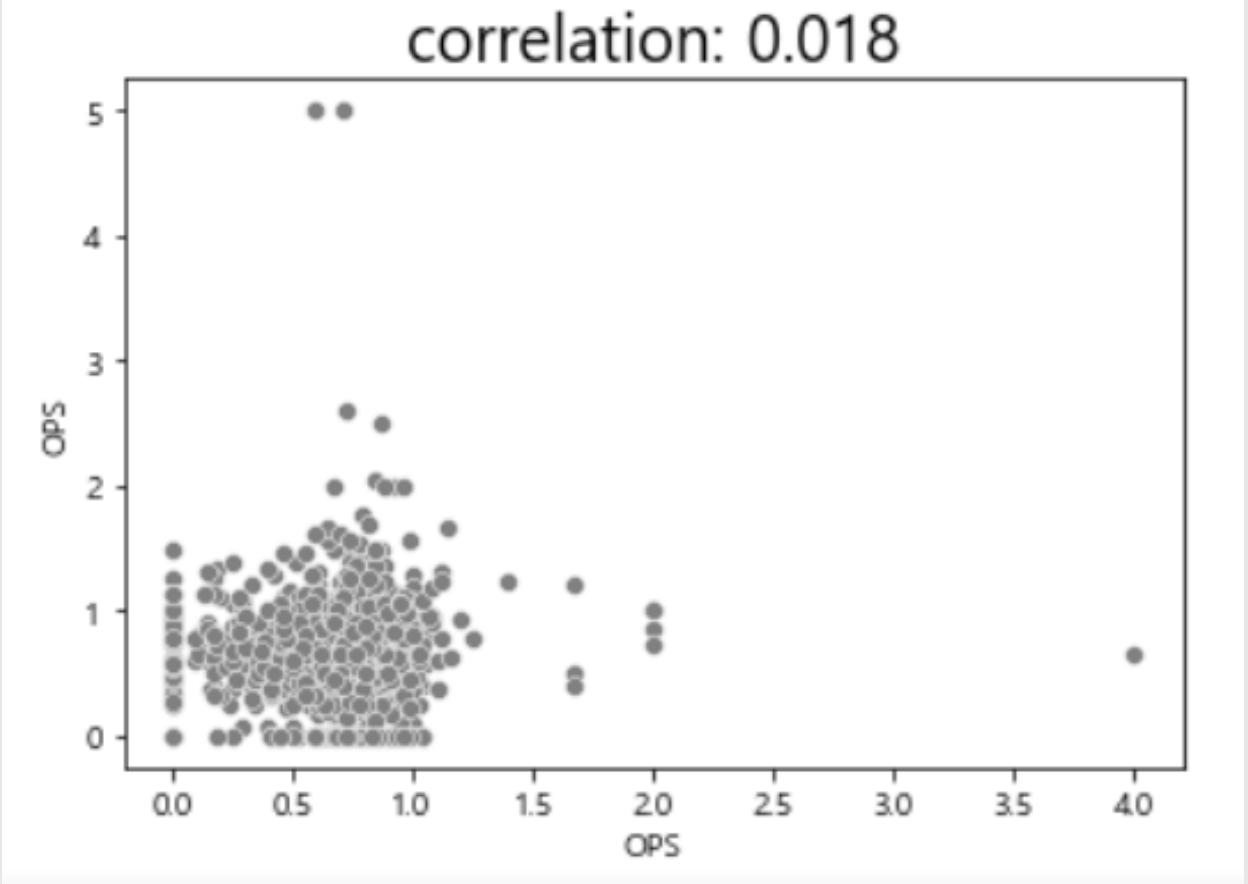
수치형 데이터 분포도 확인





상관관계(1) pre시즌 – regular 시즌

```
In [38]: sns.scatterplot(regular_new['OPS'], pre_new['OPS'], color='grey')
         plt.title('correlation: '+str(np.round(correlation,3)), fontsize=20)
         plt.xlabel=('regular OPS')
         plt.ylabel=('pre OPS')
         plt.show()
```

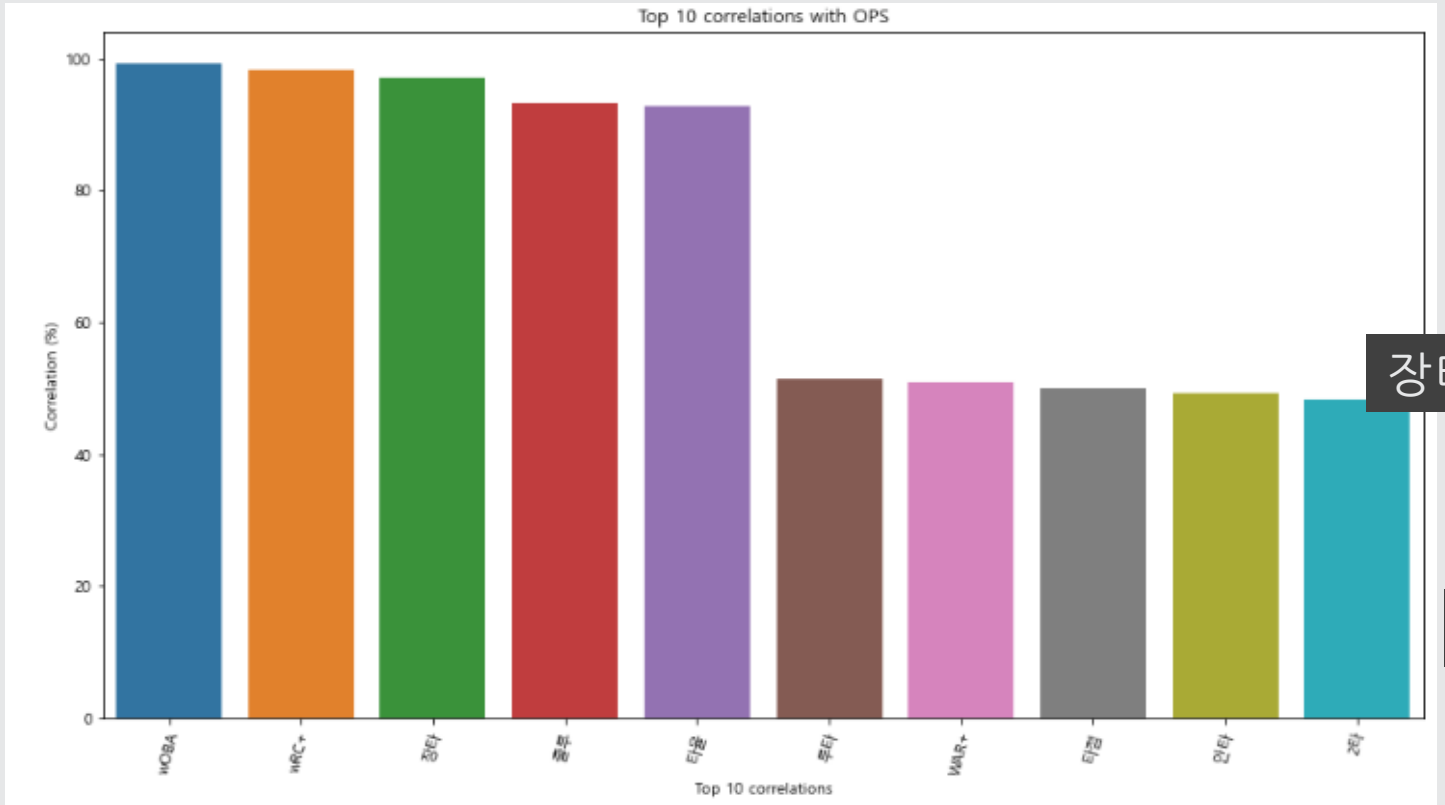


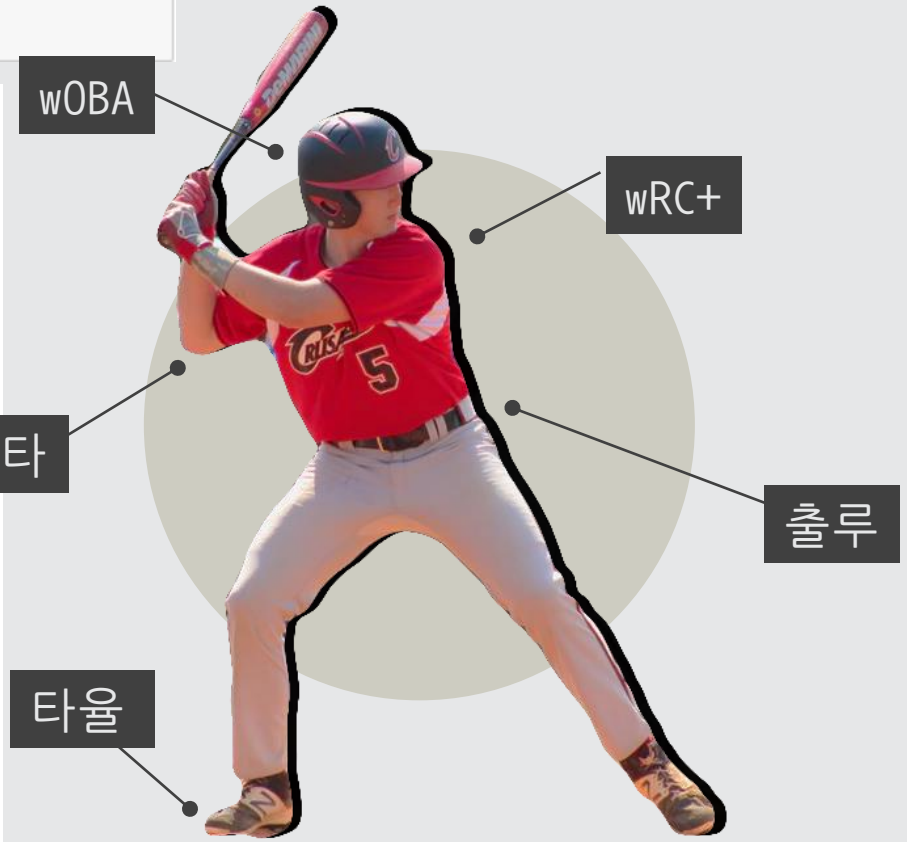
상관관계(2) OPS와의 상관관계

```
In [46]: plt.figure(figsize=(15,8))

         sns.barplot(correlations.tail(11)[:-1].index.astype(str)[::-1], 100*correlations.tail(11)[:-1][::-1])

         # Plot labeling
         plt.xticks(rotation=75) ; plt.xlabel('Top 10 correlations'); plt.ylabel('Correlation (%)');
         plt.title('Top 10 correlations with OPS')
```



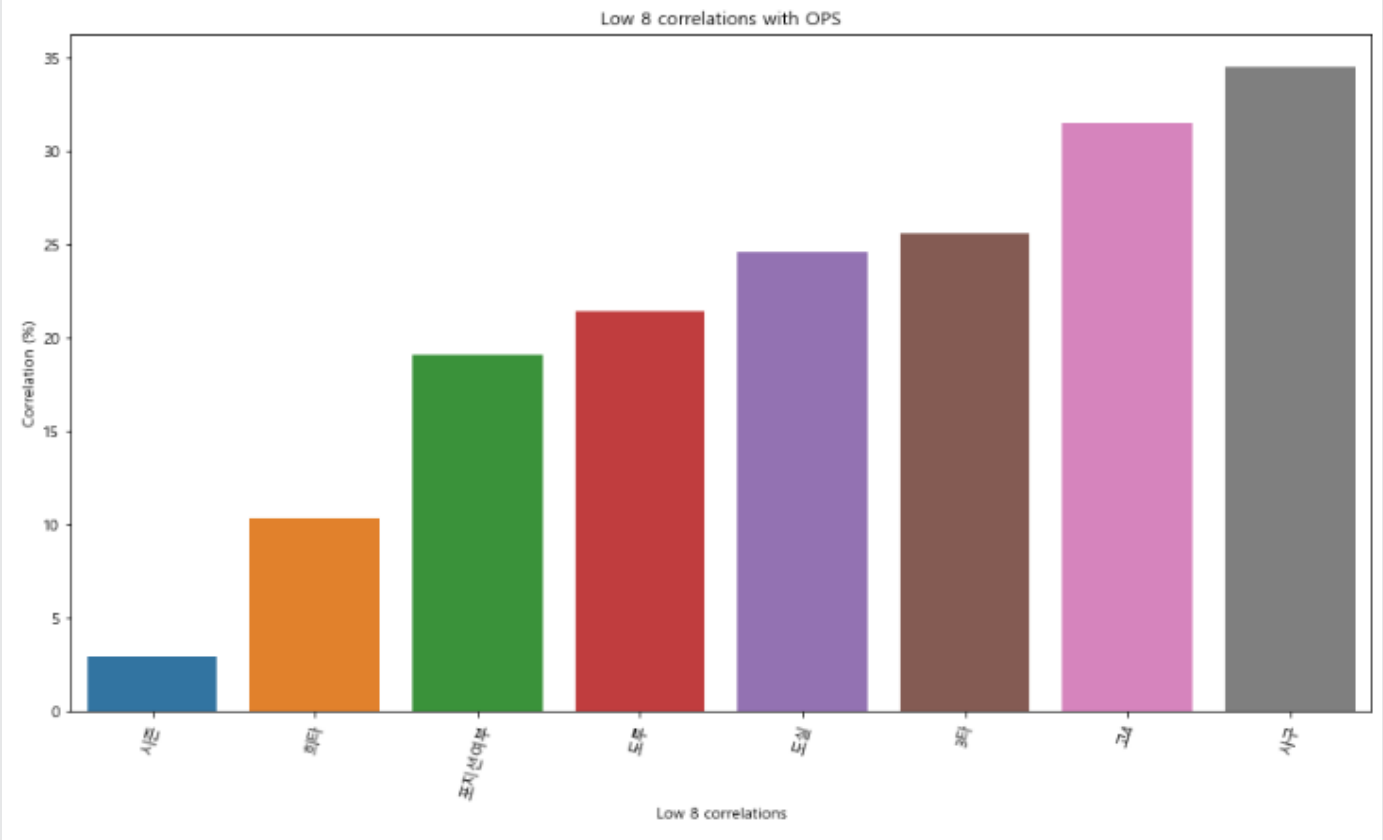


상관관계(2) OPS와의 상관관계

```
In [47]: plt.figure(figsize=(15,8))

         sns.barplot(correlations.head(10)[2:].index.astype(str), 100*correlations.head(10)[2:])

         # Plot labeling
         plt.xticks(rotation=75) ; plt.xlabel('Low 8 correlations'); plt.ylabel('Correlation (%)')
         plt.title('Low 8 correlations with OPS')
```



데 이 터
전 처 리

데이터 전처리: Feature Engineering

수치형이 아닌 변수 중 결측치가 0개 이상인 데이터 추출(df.loc[ ] 이용)

In [58]:
```
# 수치형이 아닌 변수 추출
not_num_cols = [x for x in regular_season_df.columns if x not in num_cols]

# 수치형이 아닌 변수 중 결측치가 하나라도 존재하는 행 출력
# isna().sum(axis=1) -> 열 기준의 결측치 개수
# df.loc[]를 통해 결측치 0개 이상 데이터를 추출
regular_season_df.loc[regular_season_df[not_num_cols].isna().sum(axis=1) > 0,
                      not_num_cols].head()
```

Out [58]:

	이름	생일	팀	포지션
901	정성룡	1964-05-19	해	NaN
913	이시온	1975-09-24	롯	NaN
1300	정문언	1962-08-13	태	NaN
1567	펨버튼	1969-12-17	K	NaN
1584	정문언	1962-08-13	태	NaN

데이터 전처리: Feature Engineering

In [60]:

```
# 삭제할 데이터 추출
drop_idx = regular_season_df.loc[
    # 안타가 0개 이상이면서 장타율이 0인 경우
    ((regular_season_df['안타'] > 0) & (regular_season_df['장타']==0)) |

    # 안타가 0개 이상 혹은 볼넷이 0개 이상 혹은 몸에 맞은 볼이 0개 이상이면서
    # 출루율이 0인 경우
    (((regular_season_df['안타'] > 0) |
      (regular_season_df['볼넷'] > 0) |
      (regular_season_df['사구'] > 0)) &
     (regular_season_df['출루'] == 0))
].index

# 데이터 삭제
regular_season_df = regular_season_df.drop(drop_idx).reset_index(drop=True)
regular_season_df
```

Out [60]:

	Unnamed: 0	ID	이름	생일	팀	시즌	포지션	나이	G	타석	...	병살	희타	희비	타율	출루	장타	OPS	wOBA	wRC+	WAR+
0	0	0	이종범	1970-08-15	해	1994	SS	25	124.0	561.0	...	2.0	1.0	4.0	0.393	0.452	0.581	1.033	0.462	198.3	11.77
1	1	1	테임즈	1986-11-10	N	2015	1B	30	142.0	595.0	...	7.0	0.0	7.0	0.381	0.498	0.790	1.288	0.530	222.3	10.71
2	2	2	심정수	1975-05-05	현	2003	RF	29	133.0	601.0	...	14.0	0.0	8.0	0.335	0.478	0.720	1.197	0.498	210.7	10.19
9262	10021	0	이종범	1970-08-15	K	2007	RF	38	84.0	282.0	...	6.0	10.0	4.0	0.174	0.217	0.209	0.426	0.199	8.0	-2.04
9263	10022	466	권두조	1951-06-10	청	1986	1B	36	108.0	358.0	...	5.0	25.0	0.0	0.162	0.237	0.188	0.425	0.198	17.3	-2.47

9264 rows × 34 columns

베 이 스 라 인

모 델 구 축

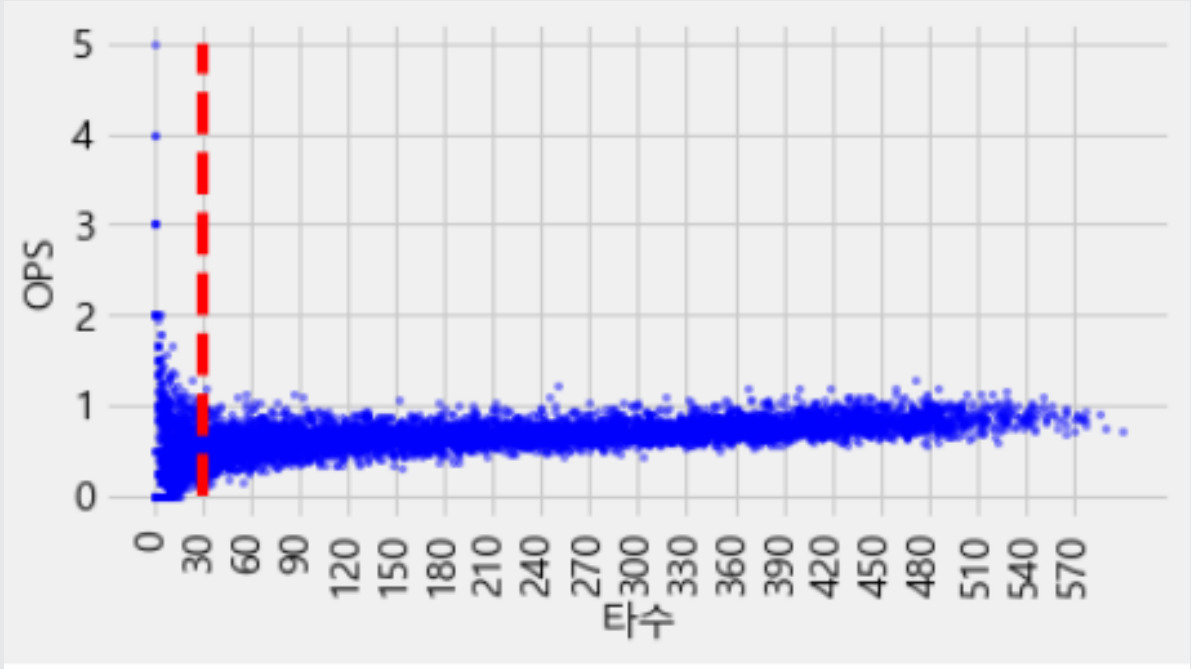
베이스라인 모델구축

Out [83] :

	이름	타수	시즌	OPS
3132	강태율	11.0	2020	1.667
4158	황대인	7.0	2016	1.571
3973	박경호	5.0	1997	1.800
4077	김종문	5.0	1990	1.500
4502	조재호	4.0	2011	1.550
4099	한익희	4.0	1997	2.000
3887	박희찬	4.0	1986	1.800
4168	김종국	3.0	1999	1.933
4323	이동훈	3.0	2005	1.667
4385	김응민	3.0	2015	1.667

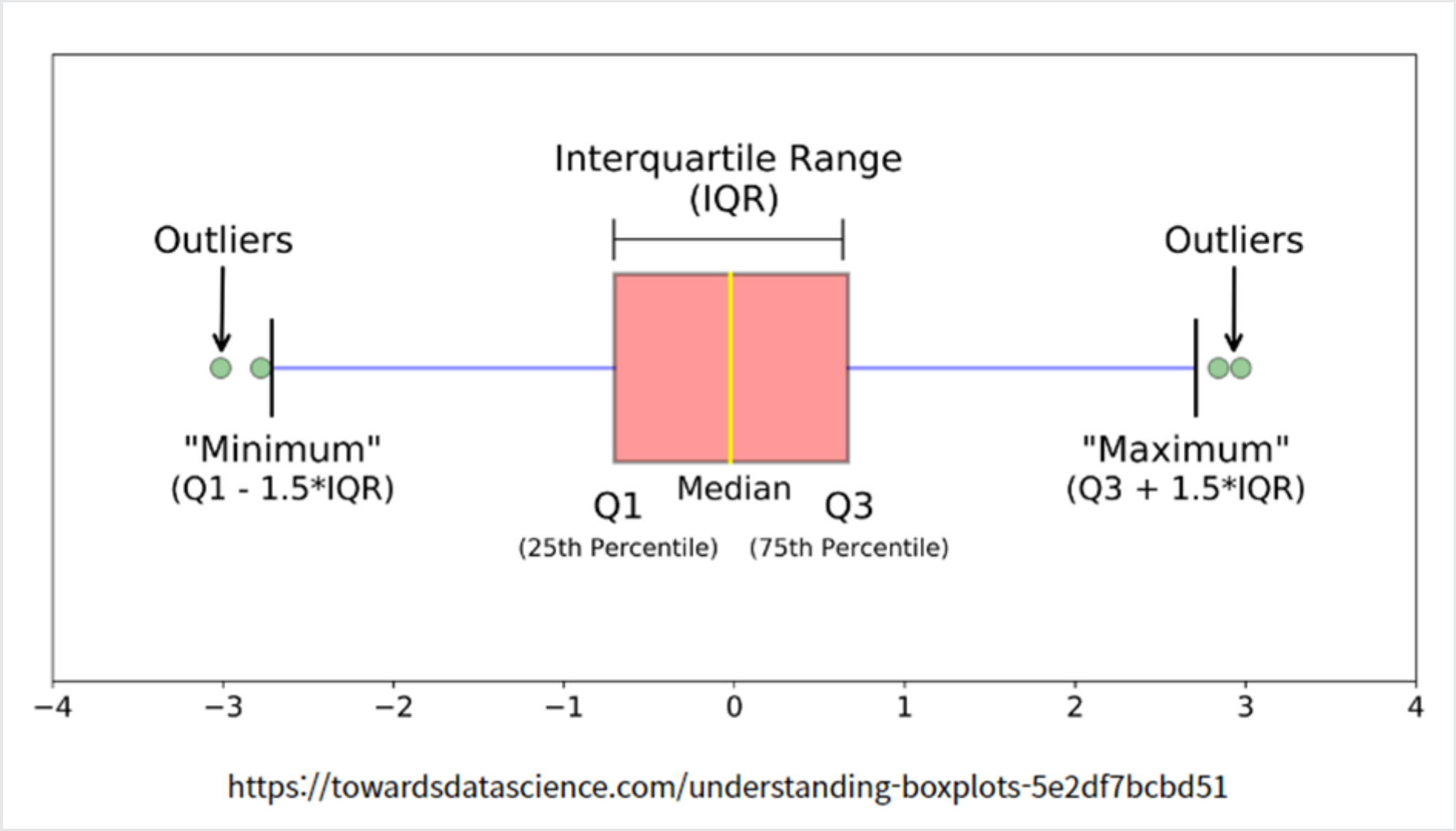
OPS는 2.000을 넘을 수 없고 일반적으로
OPS가 **.900이상**의 타자라면 **좋은 타자**로 생각하며
OPS가 1.000이 넘으면 팀을 대표하는, 리그에서
손에 꼽히는 타자로 여김

베이스라인 모델구축 (규정타석)



30타수를 기준으로 그 미만의 데이터는 삭제하고 사용

베이스라인 모델구축



https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

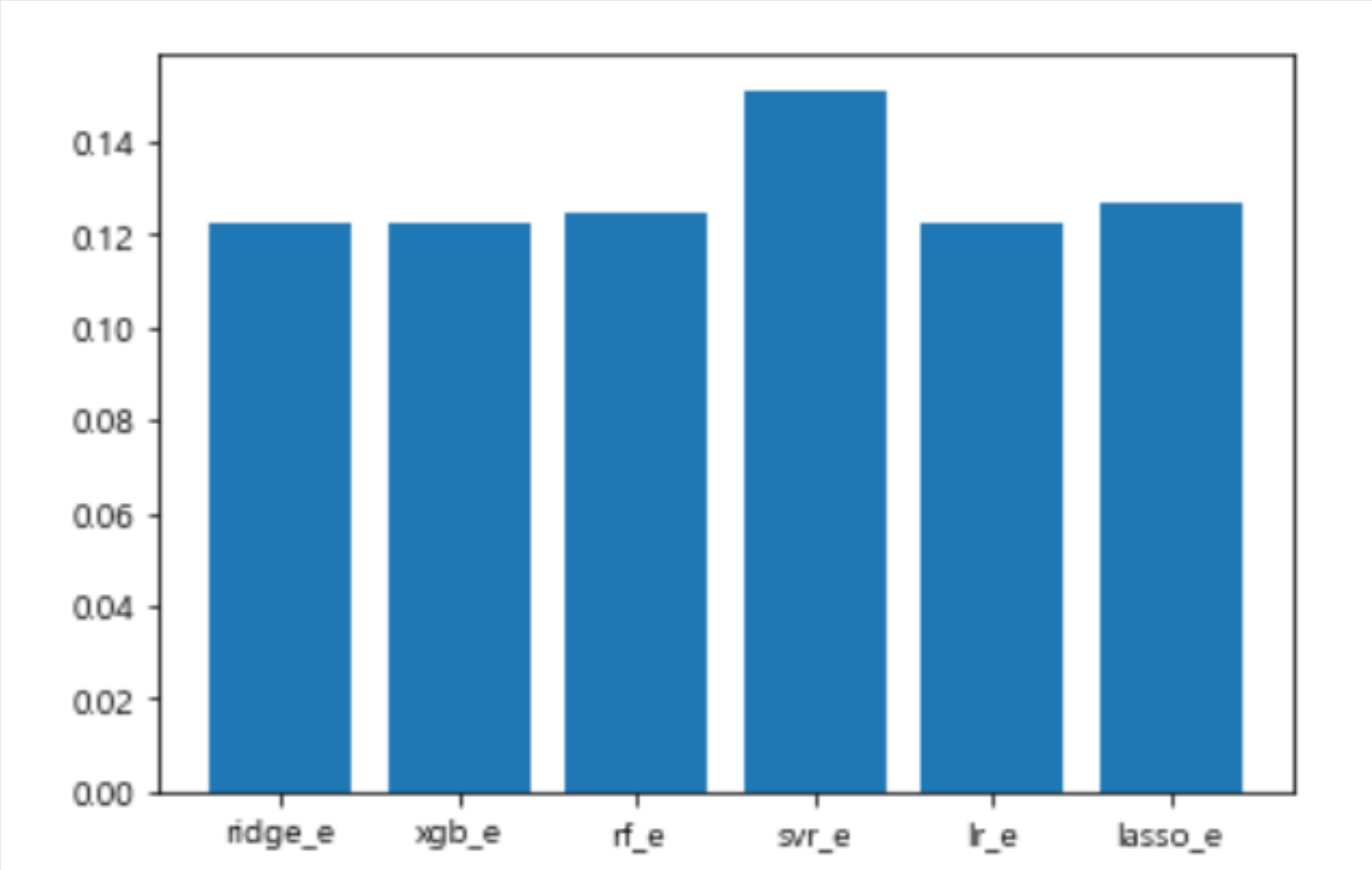
train data: 1982~2016 데이터

(2016년은 2015년까지의 데이터로,

2015년은 2014년까지의 데이터를 사용)

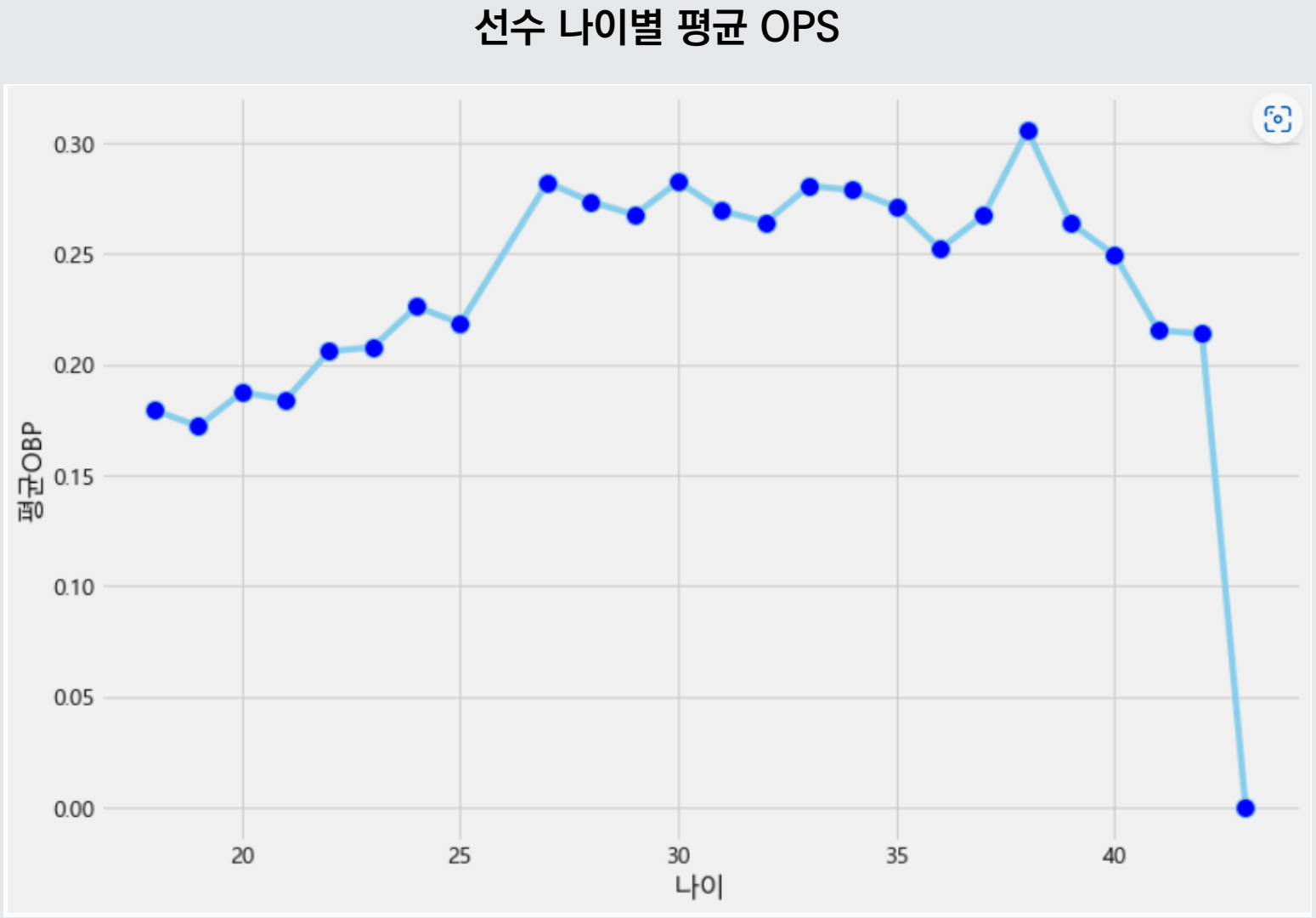
test data : 2017~2022 데이터

베이스라인 모델구축

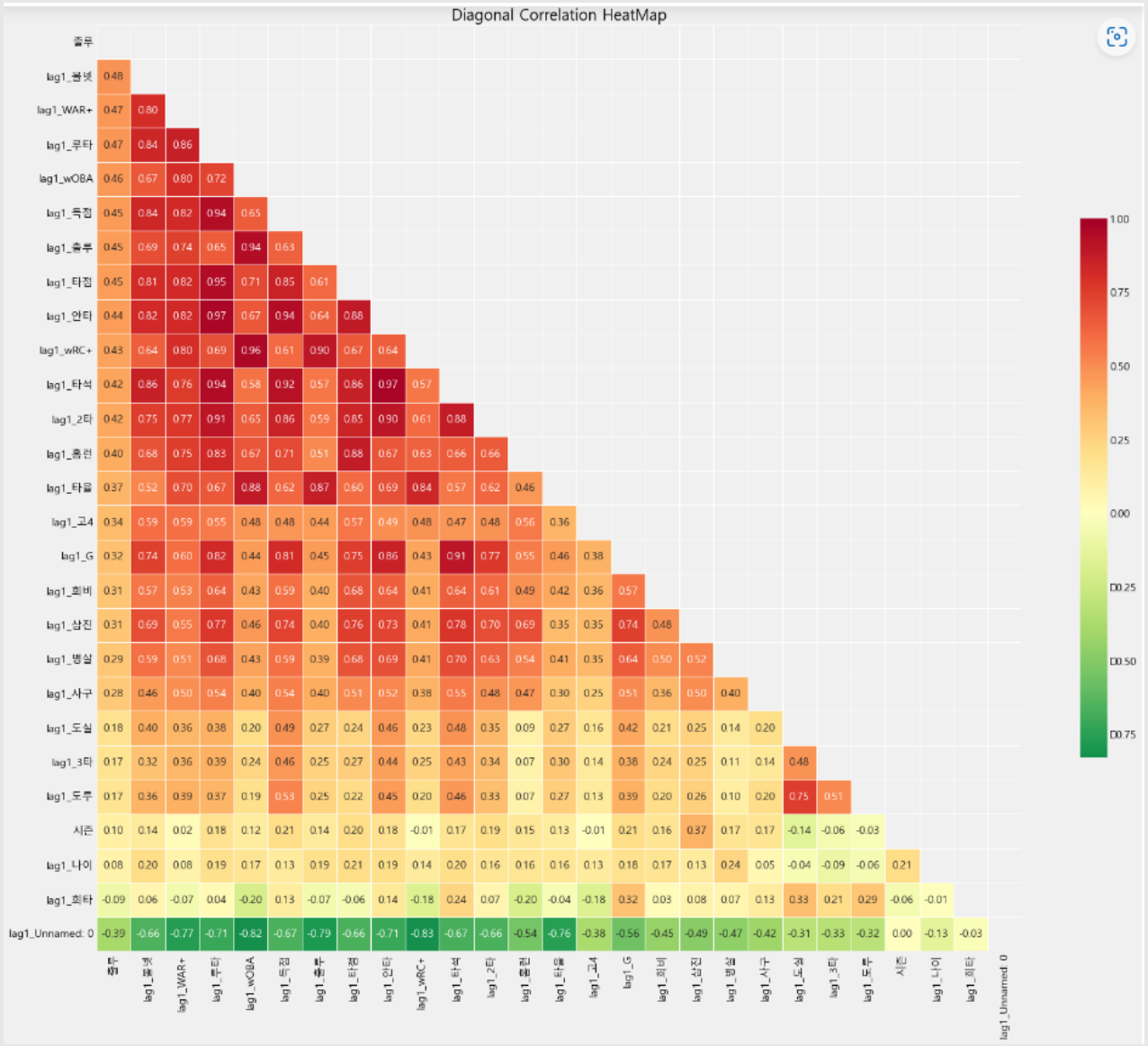

0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
ridge_e
xgb_e
rf_e
svr_e
lr_e
lasso_e

성능 올리기

성능 올리기: Feature Engineering #2



성능 올리기: Feature Engineering #2



시간변수 생성 함수를 통한 지표별 1년전 성적 추출

1년 전 지표의 상관관계 heatmap

성능 올리기: Feature Engineering #2

나잇대 별 평균OBP(타격기록) 추출

In [104]:
```
#나이
regular_season_df['생년'] = regular_season_df['생일'].str.split('-').str[0]
```

In [109]:
```
regular_season_df['나이'] = regular_season_df['시즌'] - ㅐ
                            regular_season_df['생일'].apply(lambda x: int(x[:4]))

regular_season_df
```

In [137]:
```
temp_df
```

Out[137]:

	나이	평균OBP	중위OBP
0	18	0.179500	0.1885
1	19	0.172113	0.1705
2	20	0.187639	0.1880
3	21	0.184176	0.2180
4	22	0.206024	0.2500
5	23	0.207968	0.2480
6	24	0.226257	0.2500
7	25	0.218621	0.2390
8	27	0.282342	0.3340
9	28	0.273626	0.3165
10	29	0.267698	0.3160
11	30	0.282692	0.3220
12	31	0.269563	0.3200
13	32	0.264307	0.3180
14	33	0.280810	0.3280
15	34	0.278849	0.3220
16	35	0.271184	0.3170
17	36	0.252366	0.3220
18	37	0.267450	0.3140
19	38	0.305898	0.3400
20	39	0.263854	0.3280
21	40	0.249533	0.3090
22	41	0.215556	0.2960
23	42	0.214000	0.0000
24	43	0.000000	0.0000

성능 올리기: Feature Engineering #2

희생 플라이(Sacrifice fly: SF): 야구 타격 기록의 일종으로, 타수에 포함되지 않는다. 때문에 타율 계산에서는 볼넷과 마찬가지로 무시되지만, 출루율은 낮아진다.

출처: 위키피디아

출루율(OBP) 계산 공식을 이용해 SF를 도출

$$SF = \frac{H+BB+HBP}{OBP-(AB+BB+HBP)}$$

Out[95]:

	이름	시즌	평희플
0	이종범	1994	0.008016
1	테임즈	2015	0.012712
2	심정수	2003	0.015217
3	이종범	1997	0.006198
4	이종범	1996	0.004454
...	...	...	...
9259	김헌곤	2022	0.013393
9260	김영진	1999	0.000000
9261	나종덕	2018	0.000000
9262	이종범	2007	0.015810
9263	권두조	1986	0.000000

9264 rows × 3 columns

성능 올리기: Feature Engineering #2

결측치 처리 전 출루율 데이터

Out [281] :

	Unnamed: 0	이름	시즌	타수	출루	나이	lag1_출루	lag2_출루	lag3_출루	평균_출루
0	0	박용택	2022	0	0.000	43	NaN	0.339	0.340	0.370399
1	9	호세	2007	86	0.360	42	0.399	NaN	NaN	0.436594
2	10	이병규	2016	1	1.000	42	0.255	0.288	0.384	0.372864
3	11	이병규	2016	1	1.000	42	0.255	0.288	0.384	0.372864
4	12	이병규	2016	1	1.000	33	0.255	0.288	0.384	0.372864
...	...	...	...	...	...	...	...	...	...	...
8284	12660	김종석	1989	53	0.290	18	NaN	NaN	NaN	0.340128
8285	12661	김민재	1991	15	0.133	18	NaN	NaN	NaN	0.309567
8286	12662	고영민	2002	10	0.100	18	NaN	NaN	NaN	0.352874
8287	12663	홍현우	1990	77	0.244	18	NaN	NaN	NaN	0.375992
8288	12664	하주석	2012	127	0.228	18	NaN	NaN	NaN	0.317797

8289 rows × 10 columns

성능 올리기: Feature Engineering #2

결측치 처리 후 출루율 데이터

	이름	시즌	타수	출루	나이	lag1_출루	lag2_출루	lag3_출루	평균_출루
0	백인천	1982	250	0.497	39	0.480480	0.480480	0.480480	0.480480
1	윤동균	1982	284	0.428	33	0.375937	0.375937	0.375937	0.375937
2	김우열	1982	210	0.428	33	0.379865	0.379865	0.379865	0.379865
3	권두조	1982	265	0.329	31	0.296421	0.296421	0.296421	0.296421
4	김봉연	1982	269	0.405	30	0.354692	0.354692	0.354692	0.354692
...	...	...	...	...	...	...	...	...	...
8284	김도영	2022	195	0.309	19	0.328716	0.329831	0.324324	0.308756
8285	박찬혁	2022	161	0.274	19	0.311481	0.312596	0.307089	0.274286
8286	이재현	2022	191	0.250	19	0.299338	0.300453	0.294946	0.250000
8287	한태양	2022	58	0.246	19	0.297415	0.298530	0.293023	0.246154
8288	조세진	2022	85	0.186	19	0.267362	0.268476	0.262969	0.186047

결측치 처리

그 해의 평균 출루율 +
각 선수별 출루율 평균치

성능 올리기: Feature Engineering #2

결측치 처리 전 출루율 데이터

	이름	시즌	타수	장타	lag1_장타	lag2_장타	lag3_장타	평균_장타
0	박용택	2022	0	0.000	NaN	0.396	0.344	0.451161
1	호세	2007	86	0.337	0.487	NaN	NaN	0.586466
2	이병규	2016	1	1.000	0.323	0.313	0.455	0.449750
3	최동수	2013	2	0.000	0.321	0.381	0.306	0.400786
4	조인성	2017	29	0.138	0.248	0.391	0.396	0.407741
...	...	...	...	...	...	...	...	...
7419	김종석	1989	53	0.264	NaN	NaN	NaN	0.389439
7420	김민재	1991	15	0.133	NaN	NaN	NaN	0.331961
7421	고영민	2002	10	0.100	NaN	NaN	NaN	0.370552
7422	홍현우	1990	77	0.247	NaN	NaN	NaN	0.452927
7423	하주석	2012	127	0.205	NaN	NaN	NaN	0.377625

성능 올리기: Feature Engineering #2

결측치 처리 후 출루율 데이터

	id	이름	시즌	타수	장타	lag1_장타	lag2_장타	lag3_장타	평균_장타
0	40	백인천	1982	250	0.740	0.726950	0.726950	0.726950	0.726950
1	826	윤동균	1982	284	0.532	0.415345	0.415345	0.415345	0.415345
2	835	김우열	1982	210	0.533	0.458580	0.458580	0.458580	0.458580
3	1997	권두조	1982	265	0.287	0.265752	0.265752	0.265752	0.265752
4	2061	김봉연	1982	269	0.636	0.478322	0.478322	0.478322	0.478322

성능 올리기: Grid Search (Random Forest)

```
# 랜덤 포레스트의 parameter 범위를 정의
RF_params = {
    'n_estimators': [50,100,150,200,300,500,1000],
    'max_features': ['auto', 'sqrt'],
    'max_depth' : [1,2,3,5,6,10],
    'min_samples_leaf': [1, 2, 4],
    'min_samples_split': [2, 3, 5, 10]}
```

성능 올리기: Grid Search (XGB Boost)

```
XGB_params = {
    'min_child_weight': [1,3, 5,10],
    'gamma': [0.3,0.5, 1, 1.5, 2, 5],
    'subsample': [0.6, 0.8, 1.0],
    'colsample_bytree': [0.6, 0.8, 1.0],
    'max_depth': [3, 4, 5,7,10]}
```

결 론

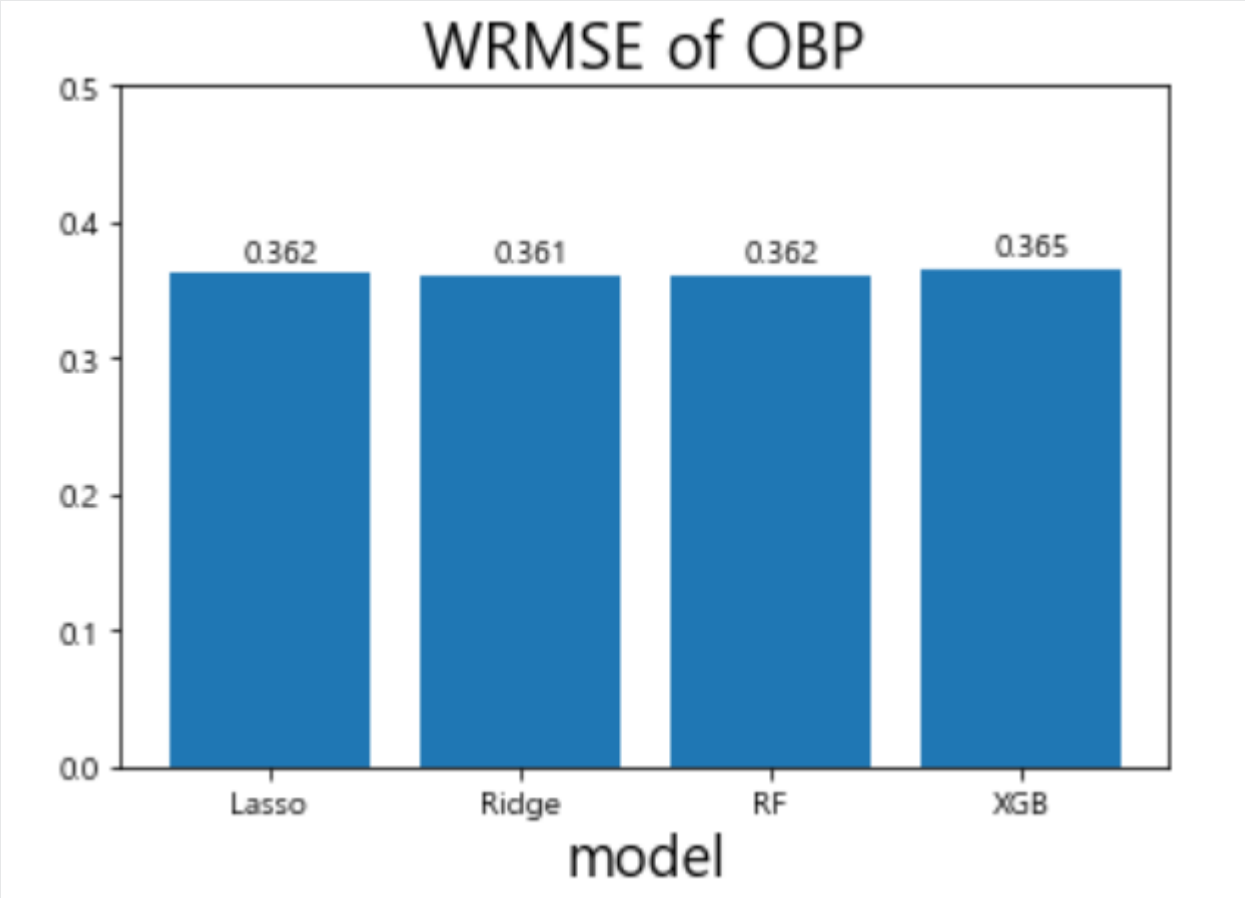
최종 예측 및 성능평가

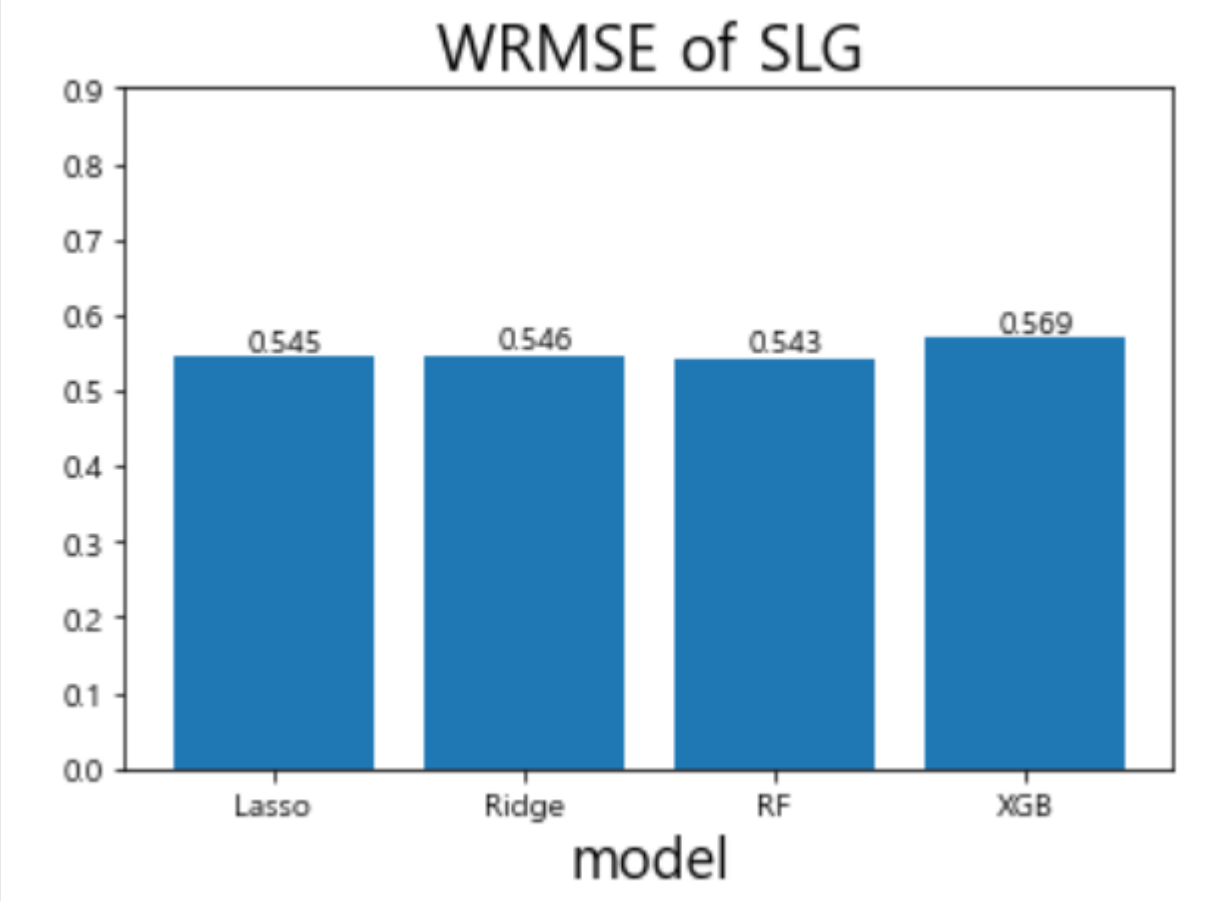
WRMSE (Weighted Root Mean Squared Error) **사용**

기존의 RMSE를 사용하지만, 각 데이터 별로 타수가 다르기 때문에 가중치를 줘야함

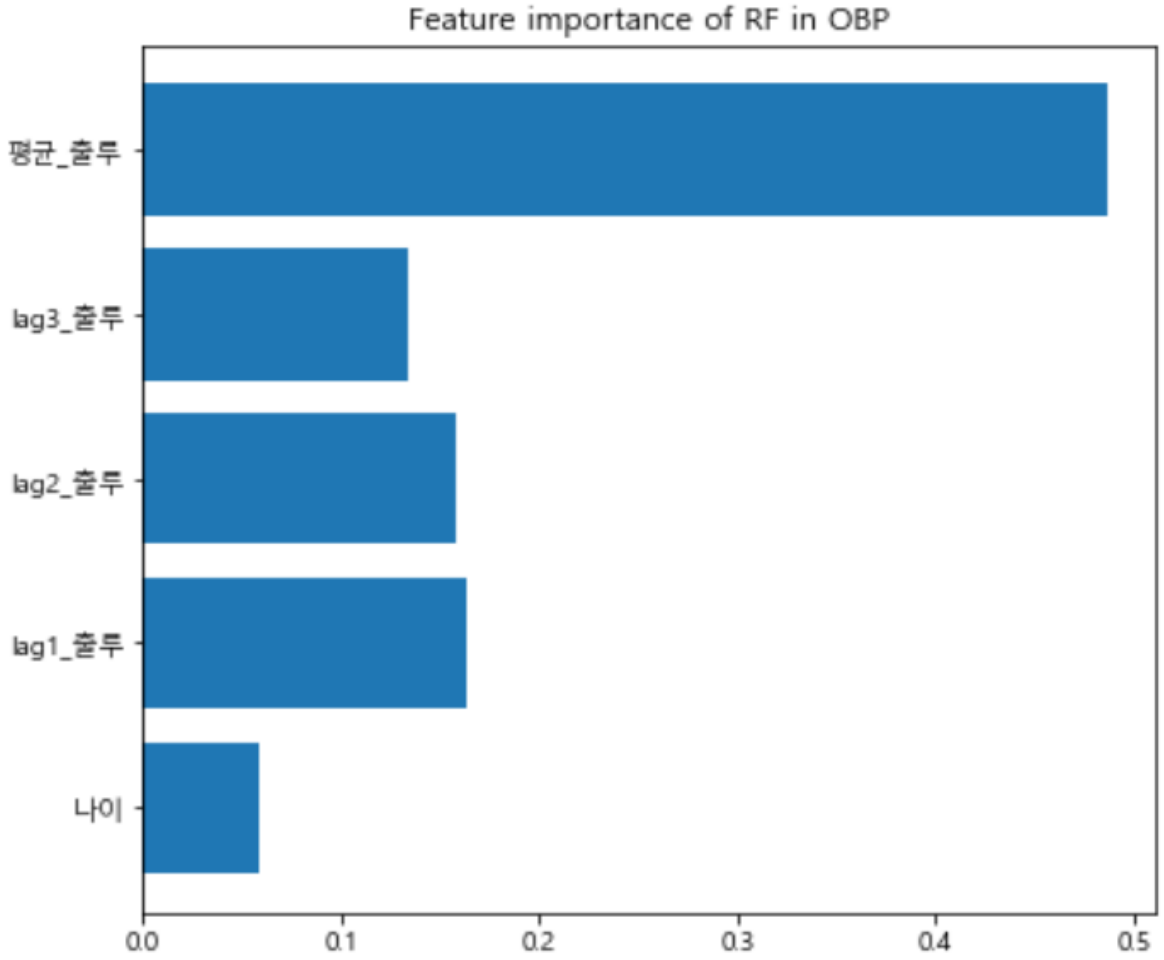
$$\mathrm{WRSME} = \sqrt{\frac{\sum((\text{실제값} - \text{예측치})^2 \times \text{타수})}{\sum \text{타수}}}$$

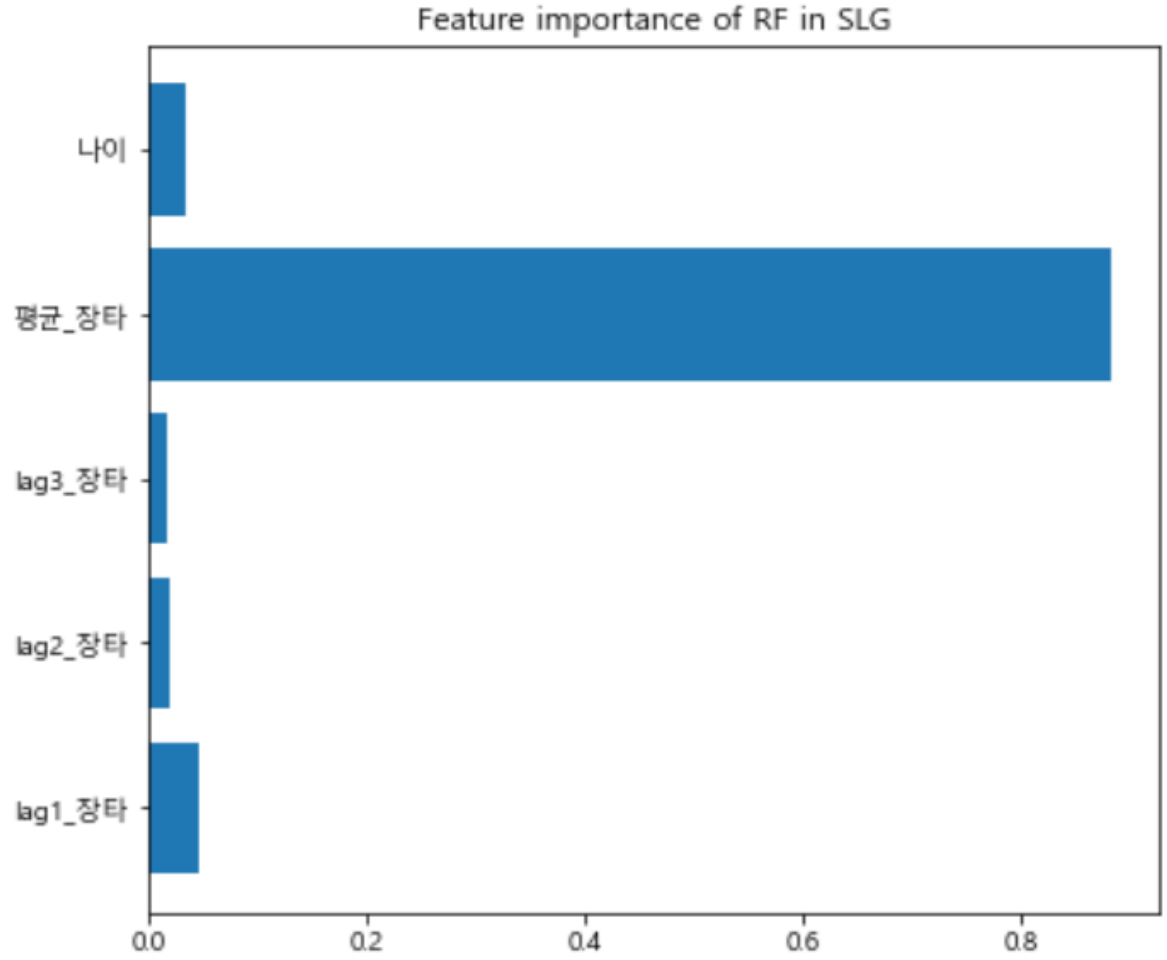
최종 예측 및 성능평가


WRMSE of OBP
0.5
0.4
0.3
0.2
0.1
0.0
0.362
0.361
0.362
0.365
Lasso
Ridge
RF
XGB
model


WRMSE of SLG
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.545
0.546
0.543
0.569
Lasso
Ridge
RF
XGB
model

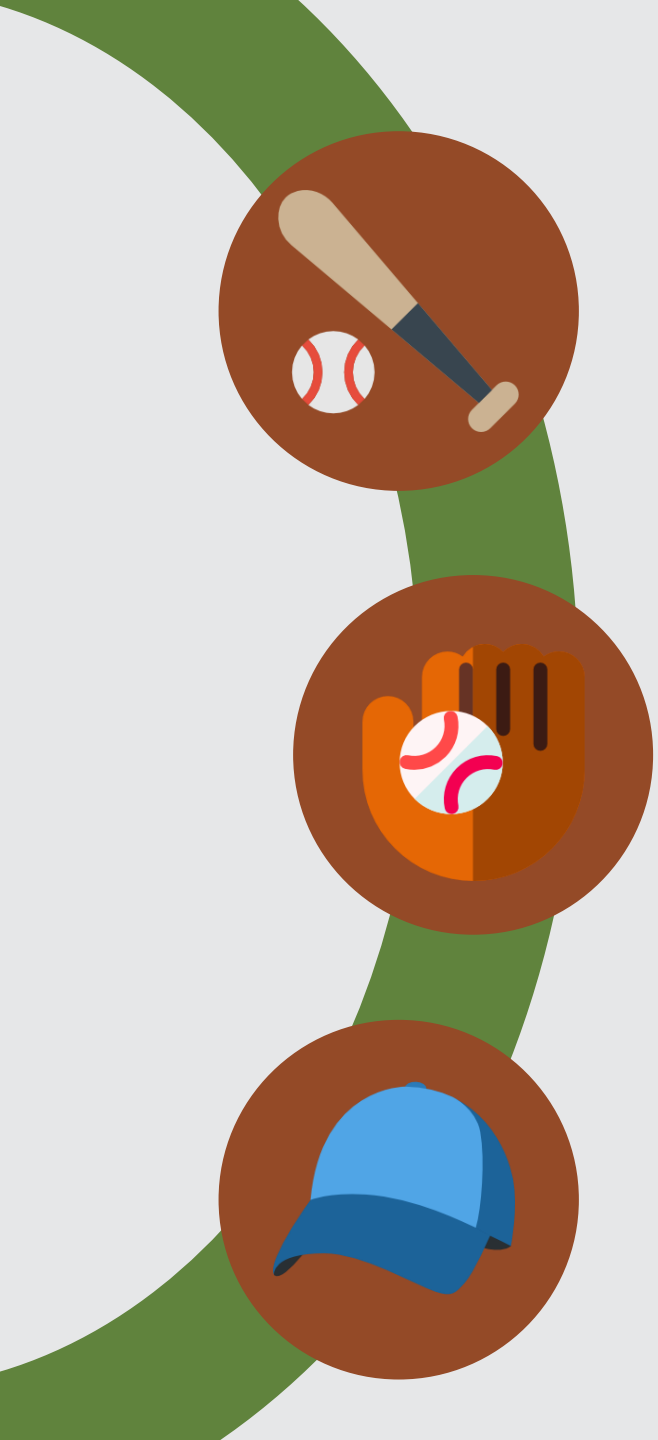
최종 예측 및 성능평가


Feature importance of RF in OBP
평균_출루
lag3_출루
lag2_출루
lag1_출루
나이
0.0
0.1
0.2
0.3
0.4
0.5


Feature importance of RF in SLG
나이
평균_장타
lag3_장타
lag2_장타
lag1_장타
0.0
0.2
0.4
0.6
0.8

최종 예측 및 성능평가

	Unnamed: 0	이름	나이_x	평균_출루	평균_장타	예상ops
0	67	로하스	33	0.388437	0.593401	0.981838
1	65	러프	37	0.403759	0.563965	0.967724
2	97	샌즈	36	0.390382	0.574468	0.964850
3	206	최형우	40	0.402351	0.534443	0.936794
4	4	강백호	24	0.408734	0.521959	0.930692
5	78	박병호	37	0.377166	0.543824	0.920990
6	56	나성범	34	0.381586	0.537150	0.918736
7	200	최정	36	0.390547	0.526430	0.916977
8	38	김재환	35	0.380723	0.529662	0.910386
9	66	로맥	38	0.375571	0.531600	0.907171



프로젝트 리뷰

송소정

- 프로젝트 주제를 두 번이나 변경해서 시간이 촉박했지만 그거대로 배울점이 많았다고 생각한다.
- 머신러닝 성능을 올리는 가장 큰 방법은 데이터를 들여다 보는 것이라는 말을 확실히 깨달았다.
- 이번에도 야구 데이터를 가지고 진행을 했는데 다음에도 아마 스포츠 쪽으로 진행할 것 같은 예감이 든다.

오유경

- **좋은점** : 초반에 다른 주제에 데이터양이 방대했는데 KBO 타자 OPS예측으로 주제를 바꿈으로써 데이터질과 양이 명확해서 비교적 수월해졌다.

- **아쉬운점** : 야구 용어가 생소해서 이해하는데 시간이 걸렸고 스스로 모델링하는 부분이 어려워서 도움을 많이 받았다

전희진

- 좋은 점: 비교적 친근한 주제여서 흥미롭게 할 수 있었다.
  새로운 알고리즘과 코드들을 배울 수 있는 좋은 기회였다.
- 아쉬운 점: 중간에 주제를 변경했는데, 그만큼 시간이 부족해 아쉬움이 남는다.
  팀 프로젝트인데 1인분을 못 한 것 같아 팀원들에게 미안하고 고맙다!!
  차후에 처음 주제였던 교통 데이터 분석도 완성해보고 싶다.


THANK
YOU