2020 빅콘테스트 데이터분석 분야 퓨처스 리그 - 야구 데이터

20-SUMMER ESC 데이터 분석 중간발표

4조 권혁 김내히 임선우 조유림 홍익선

01 조원 소개

02 데이터 설명

- Data Overview
- 분석 관점

03 EDA

- NA Check
- Feature Selection
- Skewness 조정

04 향후 계획

조원 소개

01. 조원 소개

권혁

임선우

홍익선

김내히

조유림

데이터 설명

02. 데이터 설명

⚾ 스포츠 투아이 제공 2016~2020/7/20 KBO 정규시즌 경기별 결과 및 팀/선수 기록
⚾ 연도별 팀, 경기, 선수, 등록선수, 팀투수, 팀타자, 개인투수, 개인타자 총 40개 파일

A	B
T_ID	T_NM
HH	한화
HT	KIA
KT	KT
LG	LG
LT	롯데
NC	NC
OB	두산
SK	SK
SS	삼성
WO	넥센

팀

- Features: 팀 코드, 팀 이름
- 2016~2020 모두 동일한 10개 팀
- 2019~ 넥센 → 키움 팀 이름 변경(팀 코드는 그대로)
- 팀 이름 X, 팀 코드 사용

A	B	C	D	E	F	G
G_ID	GDAY_DS	VISIT_KEY	HOME_KEY	HEADER_NO	GWEEK	STADIUM
20160401HHLG0	20160401	HH	LG	0	금	잠실
20160401HTNC0	20160401	HT	NC	0	금	마산
20160401KTSK0	20160401	KT	SK	0	금	문학
20160401LTWO0	20160401	LT	WO	0	금	고척
20160401OBSS0	20160401	OB	SS	0	금	대구
20160402HHLG0	20160402	HH	LG	0	토	잠실
20160402HTNC0	20160402	HT	NC	0	토	마산

⋮

경기

- Features: 경기 코드, 일자, 참여 팀, 더블헤더 코드, 요일, 구장
- Col 1은 Col 2~5 정보와 동일 ⇒ 사용 X

02. 데이터 설명

- 스포츠 투아이 제공 2016~2020/7/20 KBO 정규시즌 경기별 결과 및 팀/선수 기록
- 연도별 팀, 경기, 선수, 등록선수, 팀투수, 팀타자, 개인투수, 개인타자 총 40개 파일

A	B	C	D	E	F	G
GYEAR	PCODE	NAME	T_ID	POSITION	AGE_VA	MONEY
2016	60100	백창수	LG	내	28	4000만원
2016	60105	배민관	LG	투	25	2700만원
2016	60146	이승현	LG	투	25	3700만원
2016	60164	유경국	LG	투	25	2700만원
2016	60181	김지용	LG	투	28	4000만원
2016	60255	장민익	OB	투	25	2800만원
2016	60263	이재학	NC	투	26	20000만원

⋮

선수

- Features: 시즌, 선수 코드, 선수 이름, 소속 팀, 포지션, 나이, 연봉
- 선수 이름 X, 선수 코드 사용

GDAY_DS	T_ID	P_ID	ENTRY_YN
20160401	HH	60404	Y
20160401	HH	60667	Y
20160401	HH	60757	Y
20160401	HH	60764	N
20160401	HH	60768	N
20160401	HH	60805	Y
20160401	HH	61767	N
20160401	HH	61700	N

⋮

등록 선수

- Features: 일자, 팀 코드, 선수 코드, 등록 여부
- 경기 기록이 있는 선수 목록 선수 데이터에 이미 있으므로 이 데이터는 사용 X

02. 데이터 설명

- 스포츠 투아이 제공 2016~2020/7/20 KBO 정규시즌 경기별 결과 및 팀/선수 기록
- 연도별 팀, 경기, 선수, 등록선수, 팀투수, 팀타자, 개인투수, 개인타자 총 40개 파일

A	B	C
G_ID	GDAY_DS	T_ID
20160401I	20160401	LG
20160401I	20160401	HH
20160401I	20160401	NC
20160401I	20160401	HT
20160401I	20160401	SK
20160401I	20160401	KT

…

AF	AG	AH
P_WHIP_R	P2_WHIP_I	CB_WHIP_
0.642857	1.285714	2.4
1.5	1	0.75
1.333333	1.038462	2.142857
0.5	1.695652	1.875
1	2.357143	2.25
1.8	1.75	0.272727

팀투수, 개인투수

- 팀투수 Features: 경기 코드, 참여 팀 코드, 결과, 투구 수 등(34개 열)
- 개인투수 Features: 팀투수 Features + 선수 코드, 선발, 구원, 종료(38개 열)
- Feature Selection 필요

A	B	C
G_ID	GDAY_DS	T_ID
20160401I	20160401	LG
20160401I	20160401	HH
20160401I	20160401	NC
20160401I	20160401	HT
20160401I	20160401	SK
20160401I	20160401	KT

…

Z	AA	AB
P_HRA_RT	P_AB_CN	P_HIT_CN
0.333333	12	4
0.2	15	3
0.142857	7	1
0.1	10	1
0.375	8	3
0.285714	7	2

팀타자, 개인타자

- 팀타자 Features: 경기 코드, 참여 팀 코드, 타자 수, 타수 등(28개 열)
- 개인타자 Features: 팀타자 Features + 선수 코드, 선발, 타순(31개 열)
- Feature Selection 필요

02. 데이터 설명

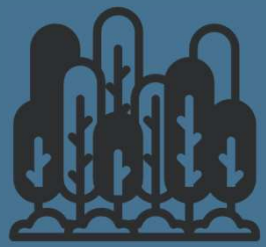
⚾ 분석 관점

나무

- 팀의 능력은 곧 선수 개개인 능력의 합
- 팀별 성적이 아닌 선수 각각의 득점, 타율, 방어율 등을 따로 예측한 뒤 소속 팀별로 합산

VS

숲



- 개별 선수의 능력은 소속 팀으로부터 영향을 받은 것
- 처음부터 X 변수로 팀별 설명변수를 두고 한번에 팀별 승률, 타율, 방어율 예측

02. 데이터 설명

분석 관점

나무

- 팀의 능력은 곧 선수 개개인 능력의 합
- 팀별 성적이 아닌 선수 각각의 득점, 타율, 방어율 등을 따로 예측한 뒤 소속 팀별로 합산

VS

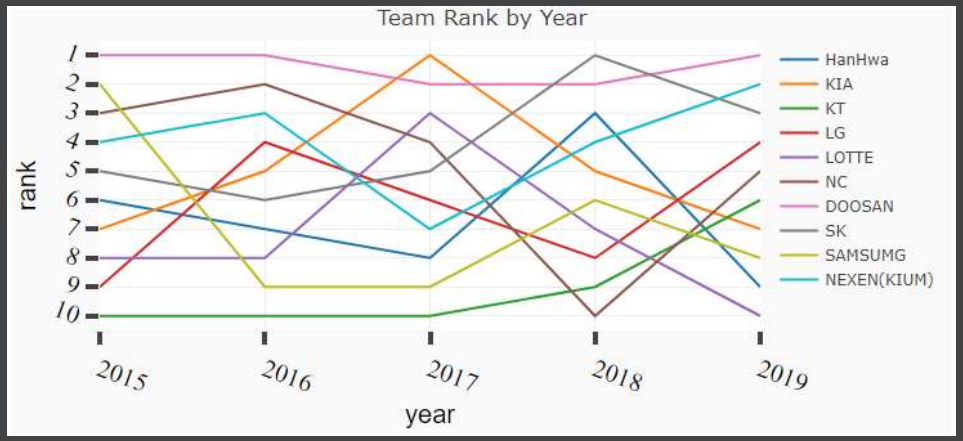
숲

- 개별 선수의 능력은 소속 팀으로부터 영향을 받은 것
- 처음부터 X 변수로 팀별 설명변수를 두고 한번에 팀별 승률, 타율, 방어율 예측

02. 데이터 설명

⚾ 분석 관점

1 팀별 시즌 성적은 변동성이 너무 크다



- 선수 개개인의 성적은 비교적 변동성이 적음

2 야구는 선수들 간 플레이의 독립성이 보장된다

- 동료들의 성적이 개인의 성적에 미치는 영향이 작음
- 특히 타율과 방어율은 더더욱 개인의 독립적인 퍼포먼스가 가장 큰 요인

02. 데이터 설명

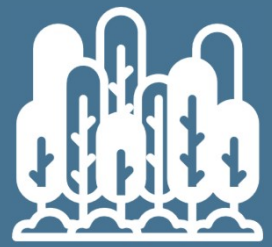
⚾ 분석 관점

나무

- 팀의 능력은 곧 선수 개개인 능력의 합
- 팀별 성적이 아닌 선수 각각의 득점, 타율, 방어율 등을 따로 예측한 뒤 소속 팀별로 합산

VS

숲



- 개별 선수의 능력은 소속 팀으로부터 영향을 받은 것
- 처음부터 X 변수로 팀별 설명변수를 두고 한번에 팀별 승률, 타율, 방어율 예측

EDA

03. EDA

1 Data Concatenation

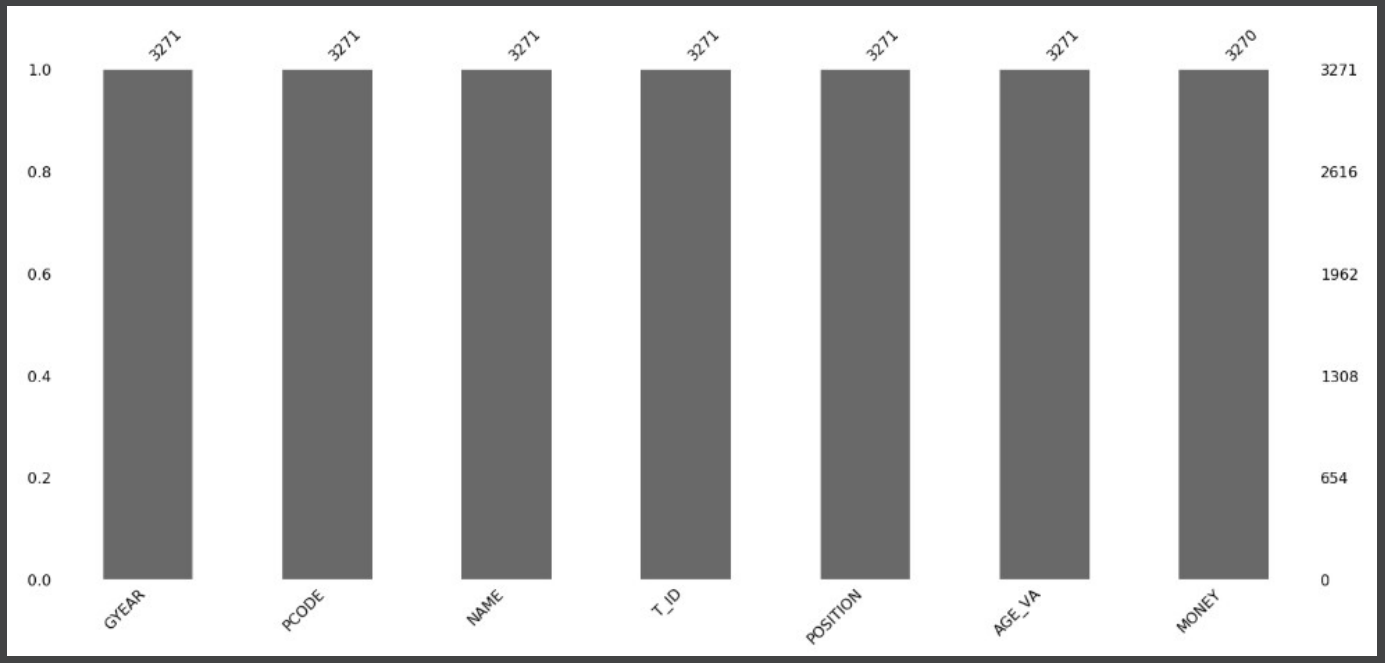
```
game = pd.concat([game_2016,game_2017,game_2018,game_2019,game_2020])
players = pd.concat([players_2016,players_2017,players_2018,players_2019,players_2020])
private_pitcher = pd.concat([private_pitcher_2016,private_pitcher_2017,private_pitcher_2018,private_pitcher_2019,private_pitcher_2020])
private_batter = pd.concat([private_batter_2016,private_batter_2017,private_batter_2018,private_batter_2019,private_batter_2020])
team_pitcher = pd.concat([team_pitcher_2016,team_pitcher_2017,team_pitcher_2018,team_pitcher_2019,team_pitcher_2020])
team_batter = pd.concat([team_batter_2016,team_batter_2017,team_batter_2018,team_batter_2019,team_batter_2020])
```

- 연도별로 따로 저장된 기존의 데이터 concatenate
- 매년 동일한 팀 데이터는 한 개만 사용, 필요한 정보가 없어 보이는 등록선수 데이터는 사용 X
- 파일 수: 40개 → 7개!
- team, game, players, private_pitcher, private_batter, team_pitcher, team_batter

03. EDA

NA Check

2 NA값 확인




```
print(players.shape)
print(players.isna().sum())
print(np.sum(players=="?", axis=1).value_counts())
```

```
(3271, 7)
GYEAR       0
PCODE       0
NAME        0
T_ID        0
POSITION    0
AGE_VA      0
MONEY       1
dtype: int64
0    3271
dtype: int64
```

- 선수 데이터에서 NA 값 한 개 발견

03. EDA

NA Check

2 NA값 확인

```
players[players["MONEY"].isna()]
```

	GYEAR	PCODE	NAME	T_ID	POSITION	AGE_VA	MONEY
95	2016	62322	밴헤켄	WO	투	37	NaN

'넥센맨' 밴 헤켄의 계약

년도	총액	계약금	연봉	인센티브
2012	30만$	3만$	27만$	0
2013	31만$	3만$	28만$	0
2014	38만$	3만$	35만$	0
2015	80만$	5만$	65만$	10만$
2016	10만$	0	0	10만$

- 2016 시즌 밴 헤켄 선수의 연봉이 Missing Value
- 넥센에서 뛰다가 일본으로 간 밴 헤켄이 일본에서 부진하자 거의 공짜로 다시 데려왔다고 한다
- KBO 연봉 기록에 인센티브는 합산 X

03. EDA

Feature Selection

3 새로운 변수 생성

EX)

```
private_batter['AVG'] = private_batter['HIT']/private_batter['AB']
private_batter['AVG'] = private_batter['AVG'].fillna(0)

team_batter['AVG'] = team_batter['HIT']/team_batter['AB']
team_batter['AVG'] = team_batter['AVG'].fillna(0)
```

```
private_batter['OBP'] = (private_batter['HIT']+private_batter['BB']+private_batter['IB']+private_batter['HP'])/private_batter['PA']
private_batter['OBP'] = private_batter['OBP'].fillna(0)

team_batter['OBP'] = (team_batter['HIT']+team_batter['BB']+team_batter['IB']+team_batter['HP'])/team_batter['PA']
team_batter['OBP'] = team_batter['OBP'].replace
team_batter['OBP'] = team_batter['OBP'].fillna(0)
```

```
private_batter["SLG"]=(private_batter["HIT"] + 2*private_batter["H2"] + 3*priva
te_batter["H3"] + 4*private_batter["HR"])/private_batter["AB"]
private_batter['SLG'].fillna(0, inplace=True)
```

NEW

- AVG 타율
- ERA 방어율
- OBP 출루율
- SLG 장타율
- OPS 출루율 + 장타율
- SB_trial 도루 시도 횟수
- SB_SR 도루 성공률
- PA - PB 타석 수 - 타수
- SH + SF 희타 + 희비

03. EDA

Feature Selection

4 Feature Selection - 경기, 선수

(경기)

```
game.drop(['G_ID','HEADER_NO','GWEEK'],axis=1, inplace=True)
#G_ID는 아래의 것들로 설명됨
#더블헤더 경기는 거의 없으므로 삭제
#경기 요일은 경기력에 영향을 미치지 않을 것으로 판단

game.head()
```

	GDAY_DS	VISIT_KEY	HOME_KEY	STADIUM
0	20160401	HH	LG	잠실
1	20160401	HT	NC	마산
2	20160401	KT	SK	문학
3	20160401	LT	WO	고척
4	20160401	OB	SS	대구

(선수)

```
players.drop(['NAME','MONEY'],axis=1, inplace=True)
#NAME 대신 PCODE 사용
#MONEY는 개인 능력이 좋으면 이후에 알아서 따라오는 것 - 개인능력치들과 corr이 높을 것으로 예상되므로 삭제
players.head()
```

	GYEAR	PCODE	T_ID	AGE_VA
0	2016	60100	LG	28
1	2016	60105	LG	25
2	2016	60146	LG	25
3	2016	60164	LG	25
4	2016	60181	LG	28

03. EDA

Feature Selection

4 Feature Selection - 개인투수, 팀투수

```
private_pitcher = private_pitcher[['TB_SC','INN2','BF','PA-AB','AB','HIT','H2','H3','HR','SB_SR','KK','WP','ER','ERA']]
```

```
private_pitcher.columns
```

```
Index(['TB_SC', 'INN2', 'BF', 'PA-AB', 'AB', 'HIT', 'H2', 'H3', 'HR', 'SB_SR',
       'KK', 'WP', 'ER', 'ERA'],
      dtype='object')
```

- 경기 코드, 소속 팀 등 종속변수(방어율)와 연관성이 떨어지는 지표 제거
- PA 대신 (PA - AB) 변수 사용 - 타석 수보다 투수가 내어준 사사구 수(BB, IB, HP 종합)가 자책점과 관련
- SB, CS 대신 SB_SR 변수 사용 - 도루, 도루 실패 횟수를 이용해 도루 성공률 산출, 이용

→ 12개 설명변수, 2개 종속변수

03. EDA

Feature Selection

4 Feature Selection - 개인타자, 팀타자

```
private_batter = private_batter[['TB_SC','PA-AB','AB','RUN', 'RBI', 'SH+SF','KK','SB_trial','HIT', 'AVG']]
```

```
private_batter.columns
```

```
Index(['TB_SC', 'PA-AB', 'AB', 'RUN', 'RBI', 'SH+SF', 'KK', 'SB_trial', 'HIT',
       'AVG'],
      dtype='object')
```

- 경기 코드, 소속 팀 등 종속변수(타율)와 연관성이 떨어지는 지표 제거
- H2, H3, HR 등 타율의 분자 구성 요소, 즉 예측의 대상이 되는 지표 제거
- SB, CS 대신 SB_trial 변수 사용 - 도루 시도 횟수가 많다는 건 출루를 많이 했다는 것
- BB, IB, HP를 따로 보지 않고 이들을 종합적으로 고려하는 (PA-AB) 변수 사용
- SH와 SF는 큰 차이가 없다는 판단, 합쳐서 사용

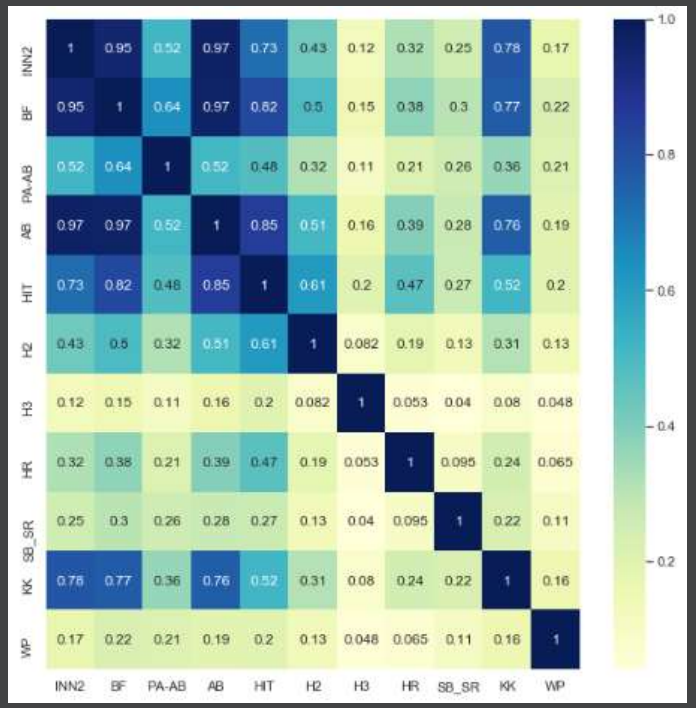
→ 8개 설명변수, 2개 종속변수

03. EDA

Feature Selection

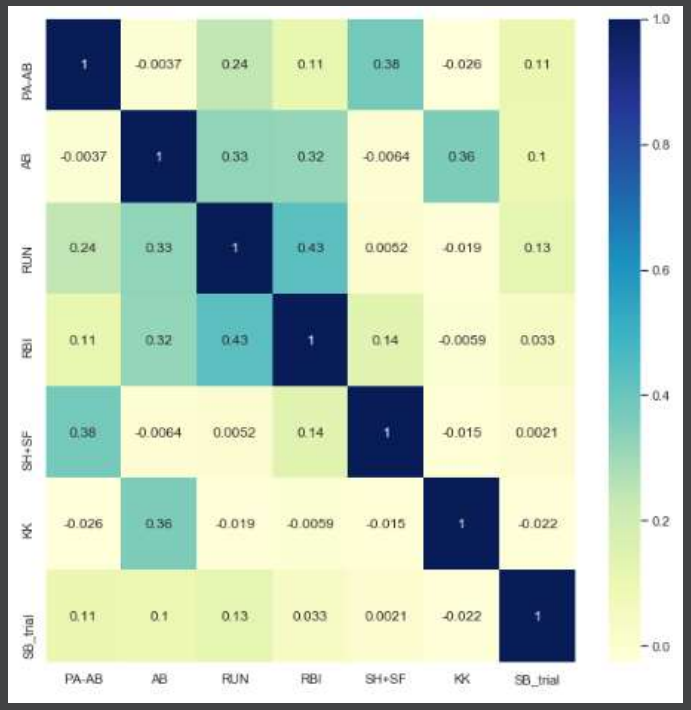
5 설명변수 간 Correlation

- 개인투수



높은 상관관계를 보이는 변수가 몇 가지 보인다. 더 제거해줘도 될 듯

- 개인타자



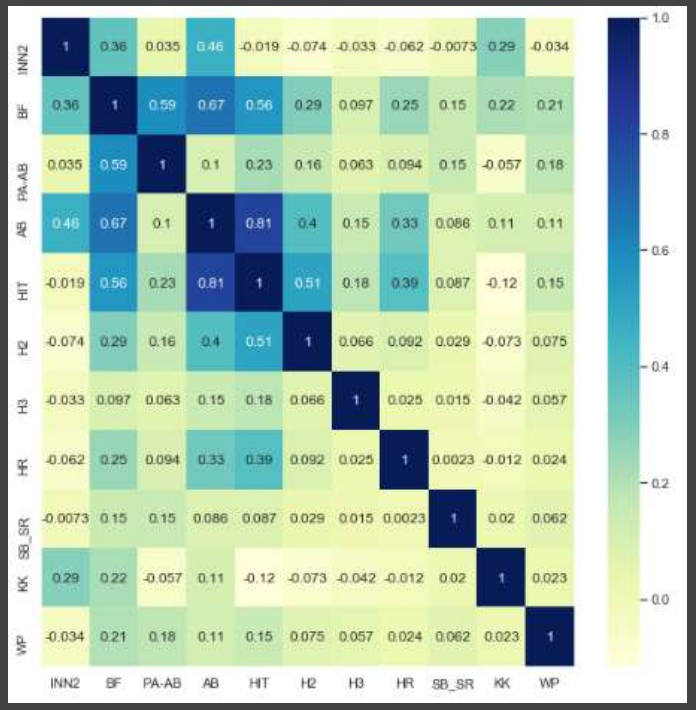
얘는 그냥 이대로 써도 되겠다

03. EDA

Feature Selection

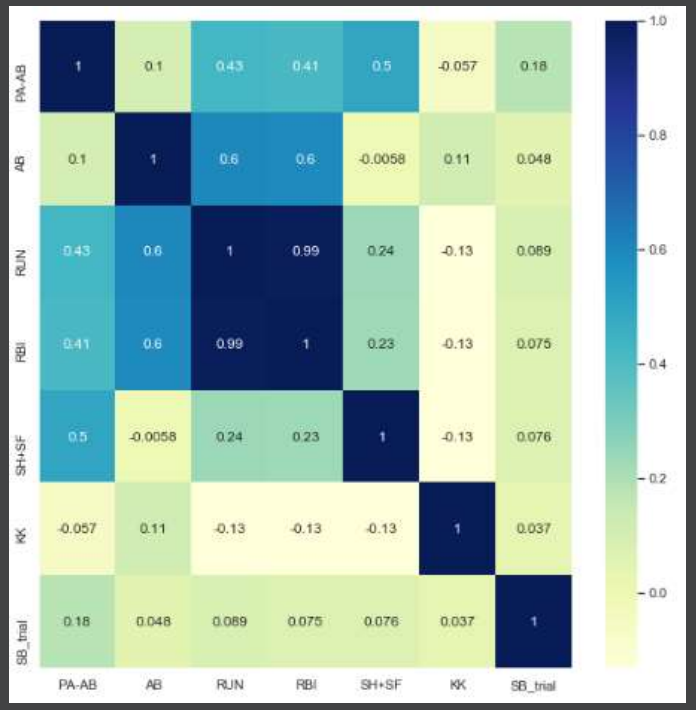
5 설명변수 간 Correlation

- 팀투수



높은 상관관계 -> 변수 선택

- 팀타자

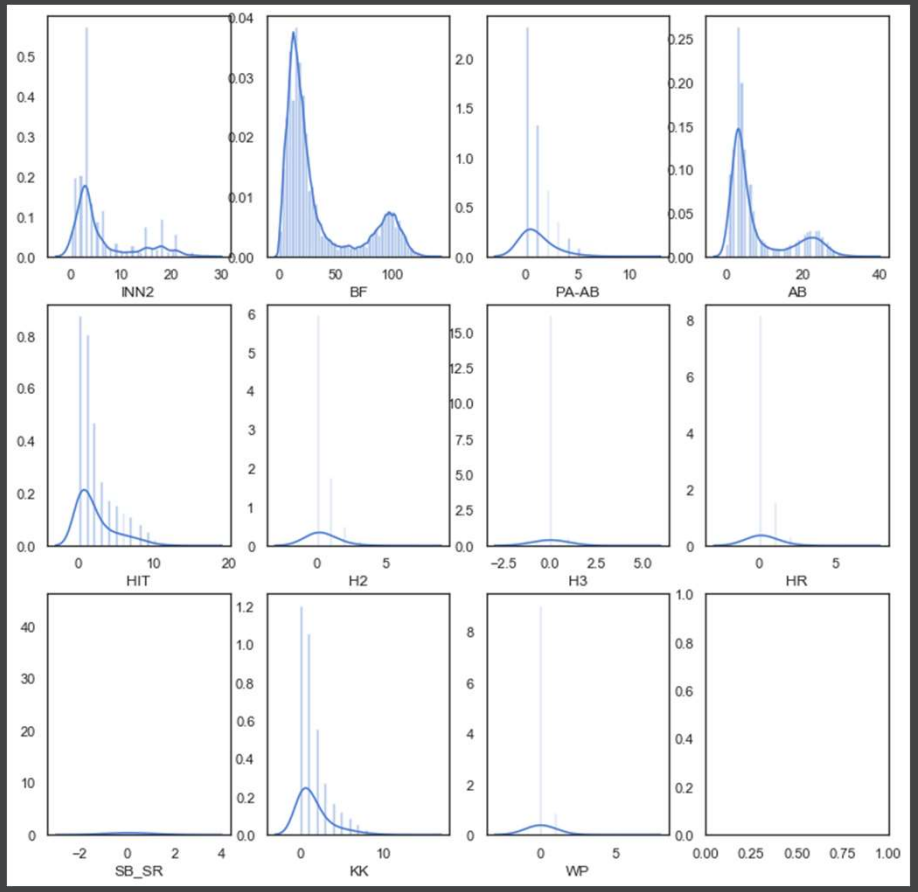


높은 상관관계 -> 변수 선택

03. EDA

⚾ Skewness 조정

6 Skewness 판단(개인투수)



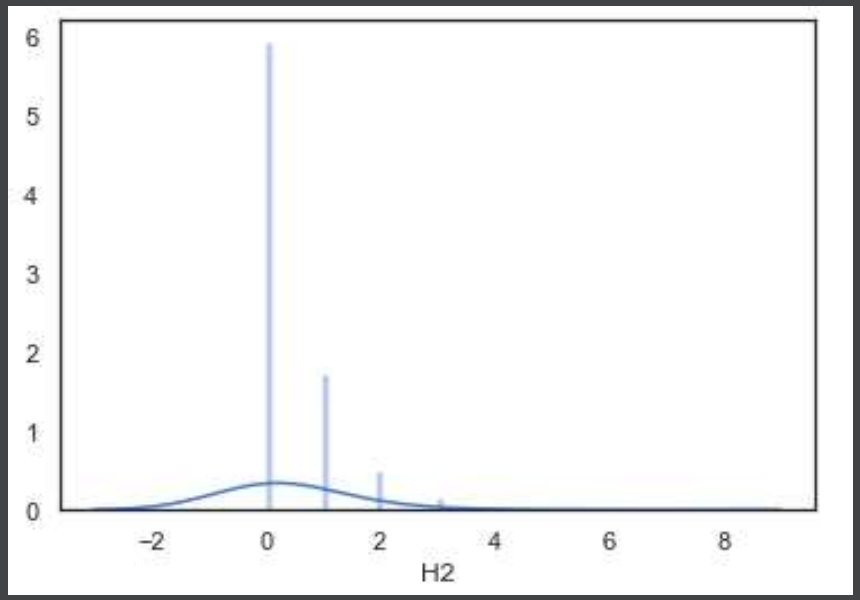

```
for i in range(1,12):
    if abs(skew(private_pitcher[private_pitcher.columns[i]])) >=2:
        print("skewness of "+private_pitcher.columns[i]+" = {}".format(skew(private_pitcher[private_pitcher.columns[i]])))
```

```
skewness of H2 = 2.1513056603411855
skewness of H3 = 5.621236832287975
skewness of HR = 2.646374266035877
skewness of SB_SR = 2.412737064572256
skewness of WP = 3.629523222206262
```

03. EDA

⚾ Skewness 조정

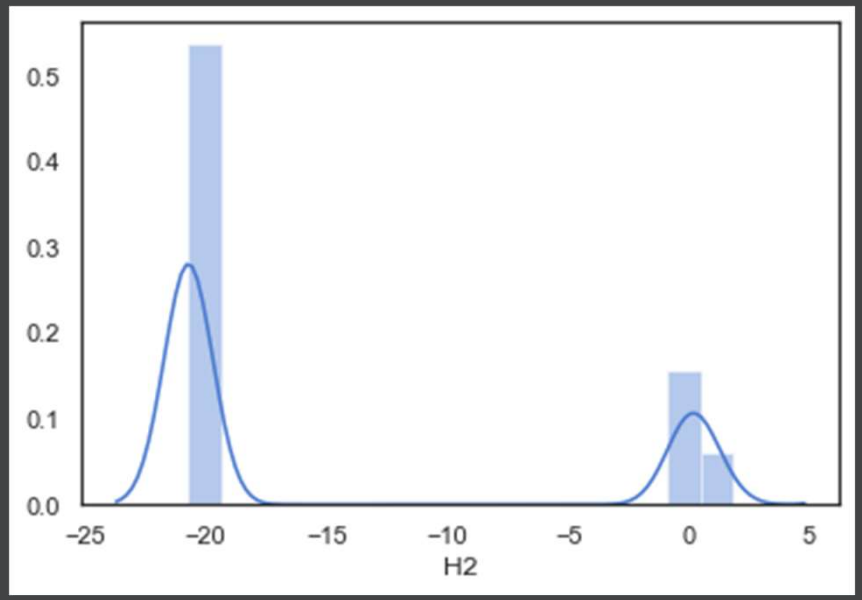
6 Skewness 판단(개인투수)



→

보류

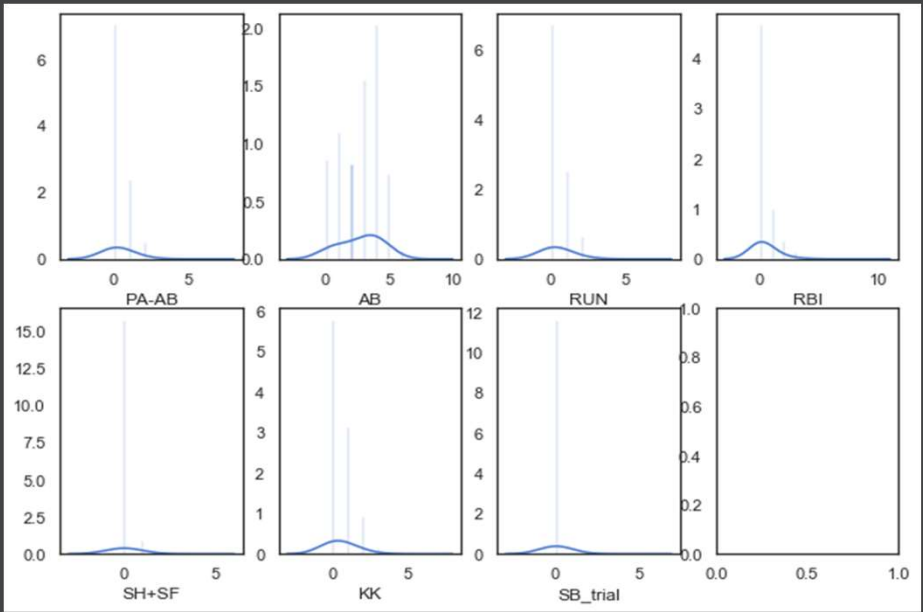
- Log transformation



03. EDA

Skewness 조정

6 Skewness 판단(개인타자)



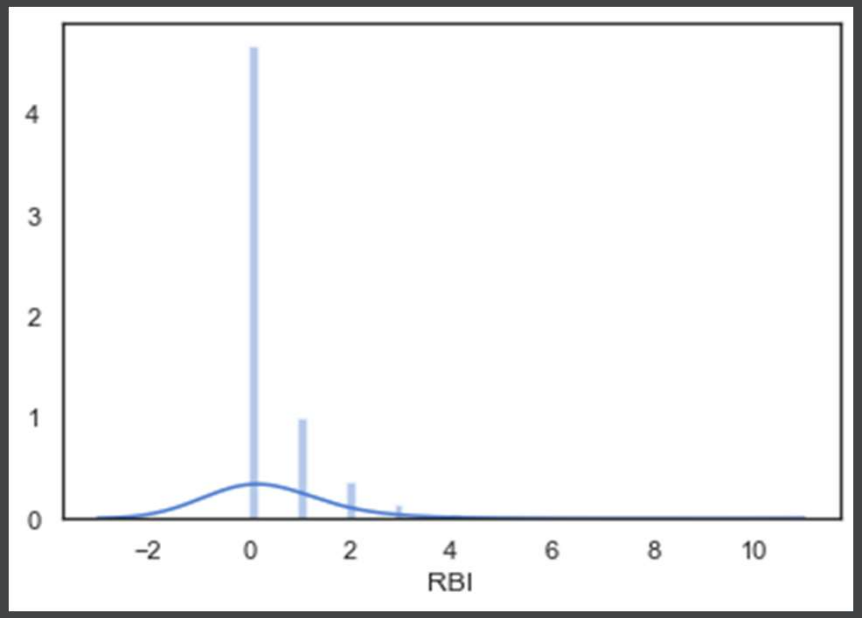

```
for i in range(1,8):
    if abs(skew(private_batter[private_batter.columns[i]])) >=2:
        print("skewness of "+private_batter.columns[i]+" = {}".format(skew(private_batter[private_batter.columns[i]])))
```

```
skewness of RBI = 2.627981398087 52
skewness of SH+SF = 4.436637073090549
skewness of SB_trial = 4.093284336939171
```

03. EDA

⚾ Skewness 조정

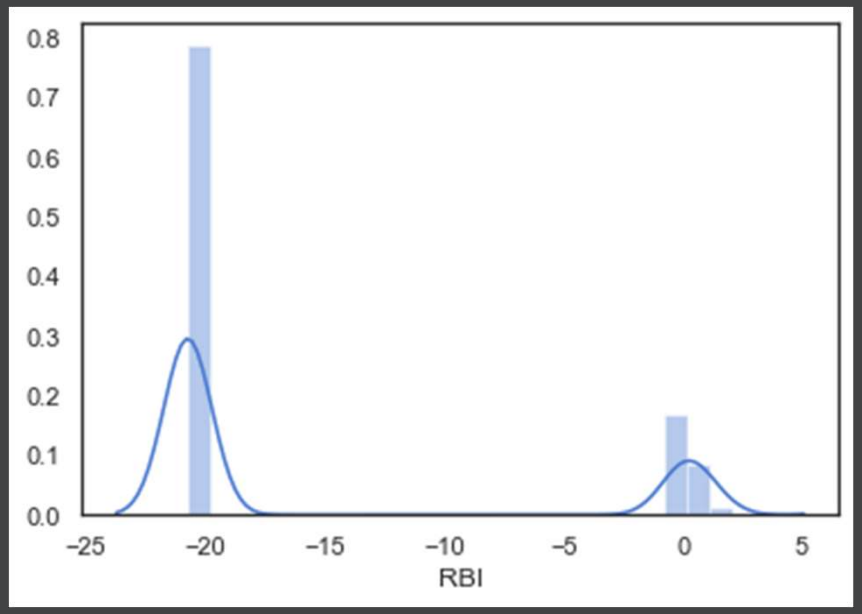
6 Skewness 판단(개인타자)



→

보류

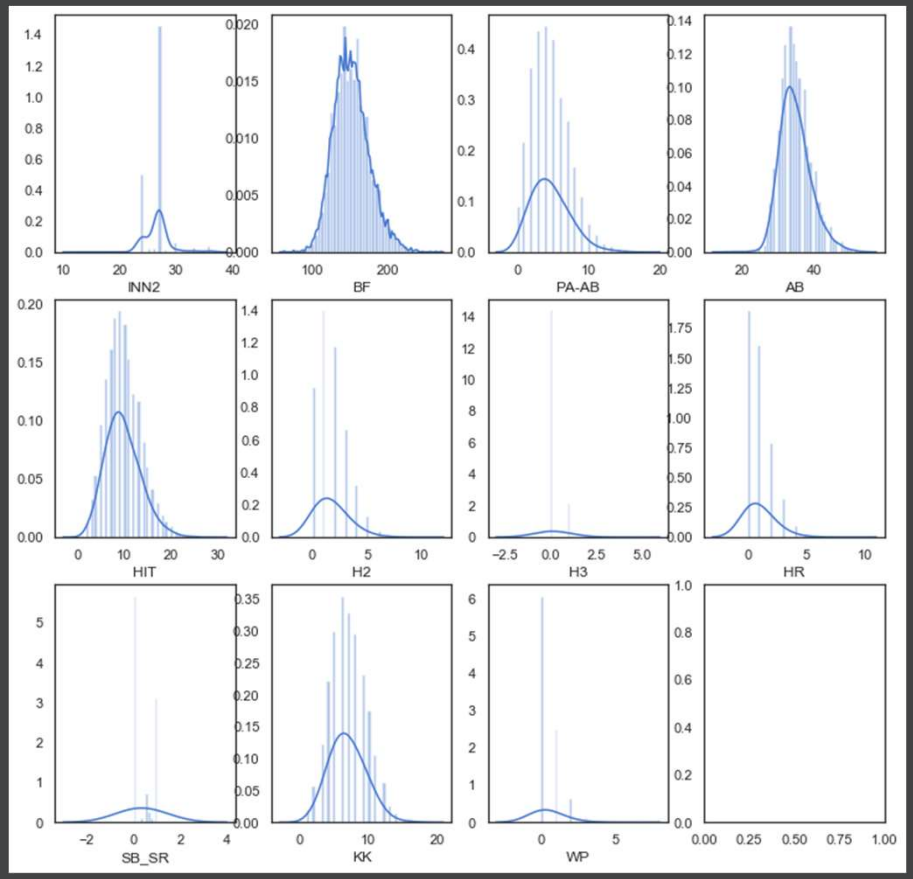
- Log transformation



03. EDA

⚾ Skewness 조정

6 Skewness 판단(팀투수)



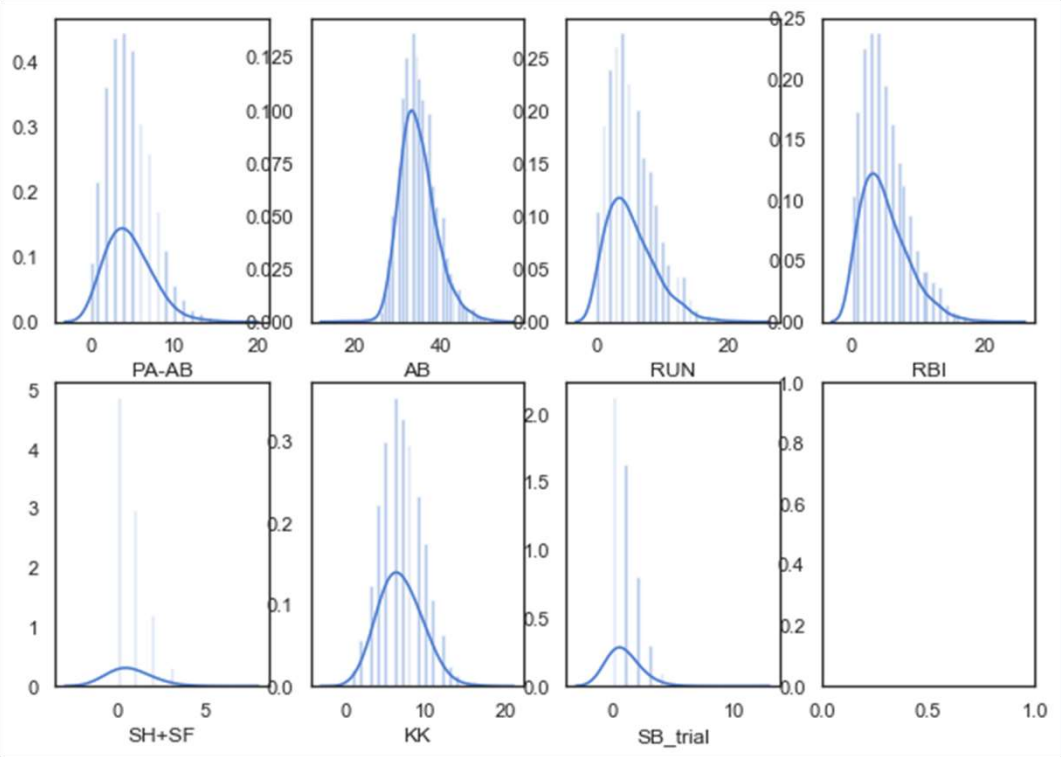

```
for i in range(1,12):
    if abs(skew(team_pitcher[team_pitcher.columns[i]])) >=2:
        print("skewness of "+team_pitcher.columns[i]+" = {}".format(skew(team_pitcher[private_pitcher.colu
```

```
skewness of H3 = 2.6381593398406924
```

03. EDA

⚾ Skewness 조정

6 Skewness 판단(팀타자)



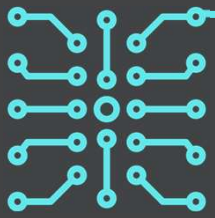
- Skewness의 절댓값이 2를 넘는 변수가 없음!

향후 계획

04. 향후 계획

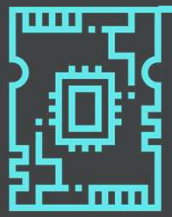
High Correlation

- 서로 상관계수가 높은 설명변수들 존재
- 어떤 변수들을 살릴 것인가?!



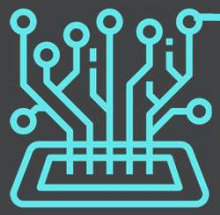
Skewness 조정

- Skewness 조정 후 양봉 분포가 되는 경우...
- 그대로..? 아니면 Transformation..?



변수 추가

- 사용 가능한 변수가 적은 상황이라 판단
- 타 사이트 데이터 추출 및 파생변수 생성
  (ex) BABIP



Modeling

- 경기 전 존재하지 않는 X 변수들을 어떻게 생성?
- 가장 적절한 모델링 기법 선택

감사합니다