Class 10. 성적 예측에 좋은 스탯

질문!

그런데 과연 당해 OPS가 좋았던 선수는 다음 시즌 OPS도 좋을까?

비싼 돈 주고 영입했는데 OPS가 확 나빠진다면??

"선수들의 지난해 OPS와 올해 OPS 간의 상관관계"를 살펴보자

타자들의 지난해 OPS와 올해 OPS 간의 상관관계

x	y
2010년 추신수 OPS	2011년 추신수 OPS
2011년 추신수 OPS	2012년 추신수 OPS
2012년 추신수 OPS	2013년 추신수 OPS
2013년 추신수 OPS	2014년 추신수 OPS
…	…
2010년 트라웃 OPS	2011년 트라웃 OPS
2011년 트라웃 OPS	2012년 트라웃 OPS
2012년 트라웃 OPS	2013년 트라웃 OPS
2013년 트라웃 OPS	2014년 트라웃 OPS
…	…

SQL 만으로 이렇게 조회하는 것은 매우 어렵다

파이썬으로 가지고 와서 이러한 형태가 되도록 처리를 해줘야 함

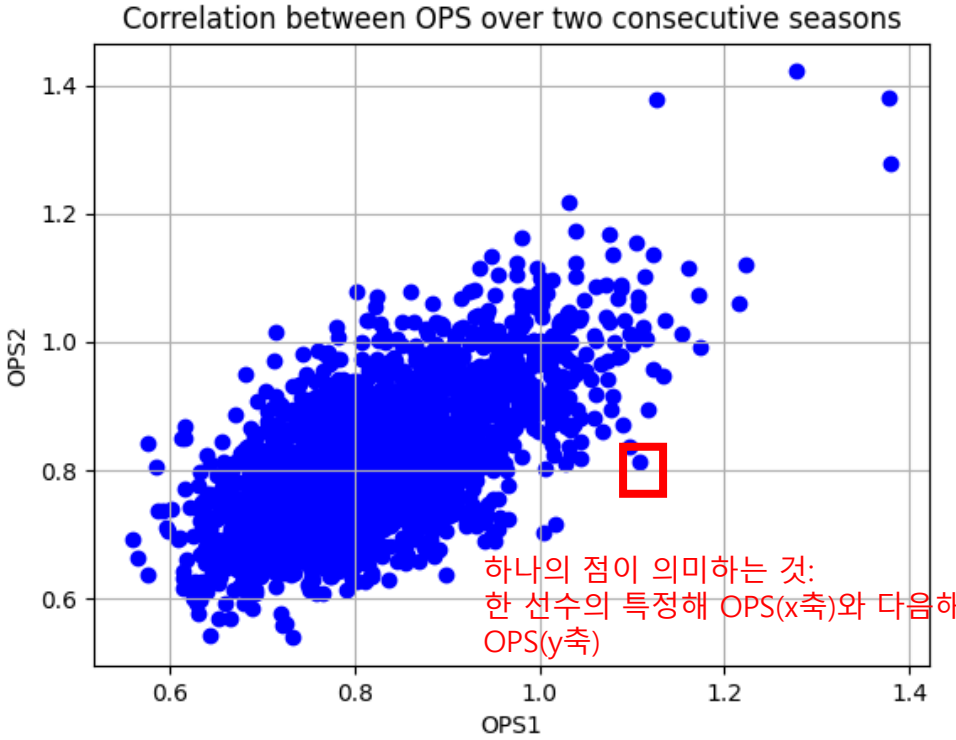
지난해 OPS와 올해 OPS 간의 상관관계

상관계수의 절대값	정도
0 ~ 0.2	상관관계 거의 없음
0.2 ~ 0.4	약한 상관관계
0.4 ~ 0.7	다소 강한 상관관계
0.7 ~ 0.9	강한 상관관계
0.9 ~ 1.0	매우 강한 상관관계

ex5.py

코드 상세 설명!

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
from scipy import stats
with sqlite3.connect("lahmansbaseballdb.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT playerID, yearID, CAST((H + BB + HBP) AS REAL)/(AB + BB + HBP + SF) + CAST(((H - "2B" -
"3B" - HR) + 2*"2B" + 3*"3B" + 4*HR) AS REAL)/AB AS OPS
    FROM batting WHERE yearID >= 1990 and (AB + BB + HBP + SH + SF) >= 502 ORDER BY playerID;
    ''')
    result = cur.fetchall()

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

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

before = []
after = []

for i in range(len(df)-1):
    if df.iloc[i+1, 0] == df.iloc[i, 0]:
        if df.iloc[i+1, 1] == df.iloc[i, 1] + 1:
            before.append(df.iloc[i, 2])
            after.append(df.iloc[i+1, 2])

plt.scatter(before, after, c='b')
plt.title('Correlation between OPS over two consecutive seasons')
plt.xlabel('OPS1')
plt.ylabel('OPS2')
plt.grid(True)
plt.savefig('ex5_img.png')
correlation_coefficient = stats.pearsonr(before, after)
print("상관계수:", correlation_coefficient[0])
```



상관계수 : 0.6525293122819918

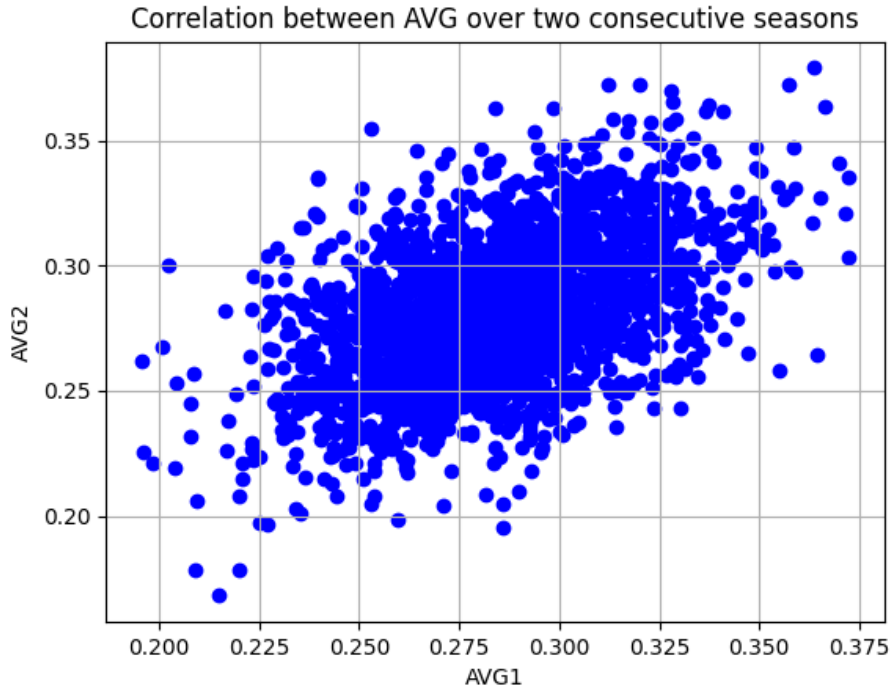
다소 강한 상관관계

특정해에 OPS가 좋았던 선수는 다음 해에도 OPS가 좋을 가능성이 있다

OPS보다 좀 더 다음 해 성적을 예측하기 좋은 스탯은 무엇일까?

조건:
1990년 이후

타율



상관계수 : 0.4905995833790807

다소 강한 상관관계

타율은 다음 시즌의 성적을 예측하는데 있어서 OPS보다 신뢰도가 떨어진다

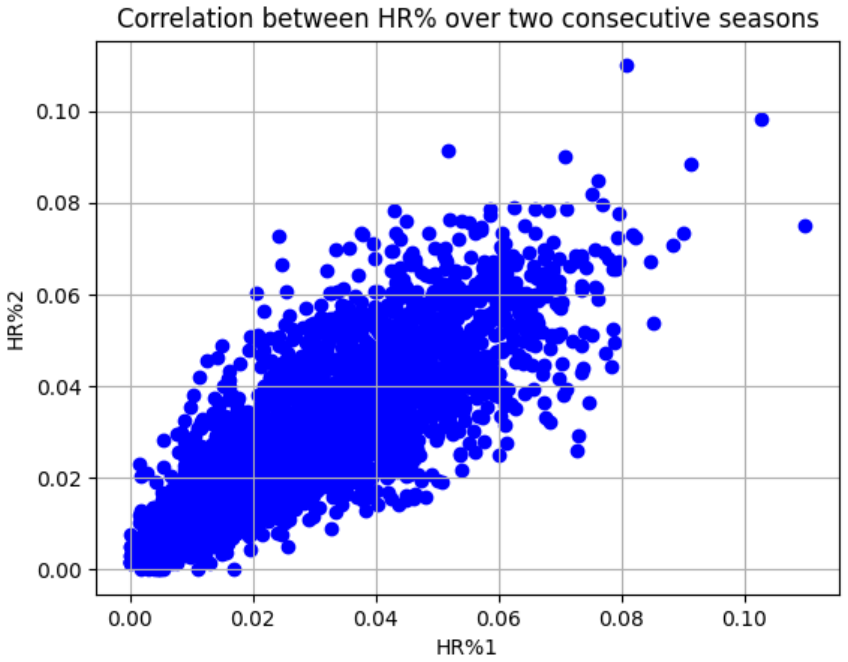
OPS보다 좀 더 다음 해 성적을 예측하기 좋은 스탯은 무엇일까?

HR% = 홈런/타석

타석당 홈런 비율(HR%)

ex7.py

```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
from scipy import stats
with sqlite3.connect("lahmansbaseballdb.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT playerID, yearID, (HR + 0.0)/(AB + BB + HBP + SH + SF) AS "HR%"
    FROM batting WHERE yearID >= 1990 and (AB + BB + HBP + SH + SF) >= 502 ORDER BY playerID;
    ''')
    result = cur.fetchall()

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

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

before = []
after = []

for i in range(len(df)-1):
    if df.iloc[i+1, 0] == df.iloc[i, 0]:
        if df.iloc[i+1, 1] == df.iloc[i, 1] + 1:
            before.append(df.iloc[i, 2])
            after.append(df.iloc[i+1, 2])

plt.scatter(before, after, c='b')
plt.title('Correlation between HR% over two consecutive seasons')
plt.xlabel('HR%1')
plt.ylabel('HR%2')
plt.grid(True)
plt.savefig('ex7_img.png')
correlation_coefficient = stats.pearsonr(before, after)
print("상관계수:", correlation_coefficient[0])
```




```
상관계수: 0.7314803303478029
```

강한 상관관계

이번 시즌에 홈런을 잘 친 타자는 다음 시즌에도 홈런을 잘 칠 확률이 높다 (HR% > OPS > AVG)

정리

OPS와 같이 성적을 평가하기에 좋은 스탯이 있고

HR%와 같이 내년 성적을 예측하기에 좋은 스탯이 있다

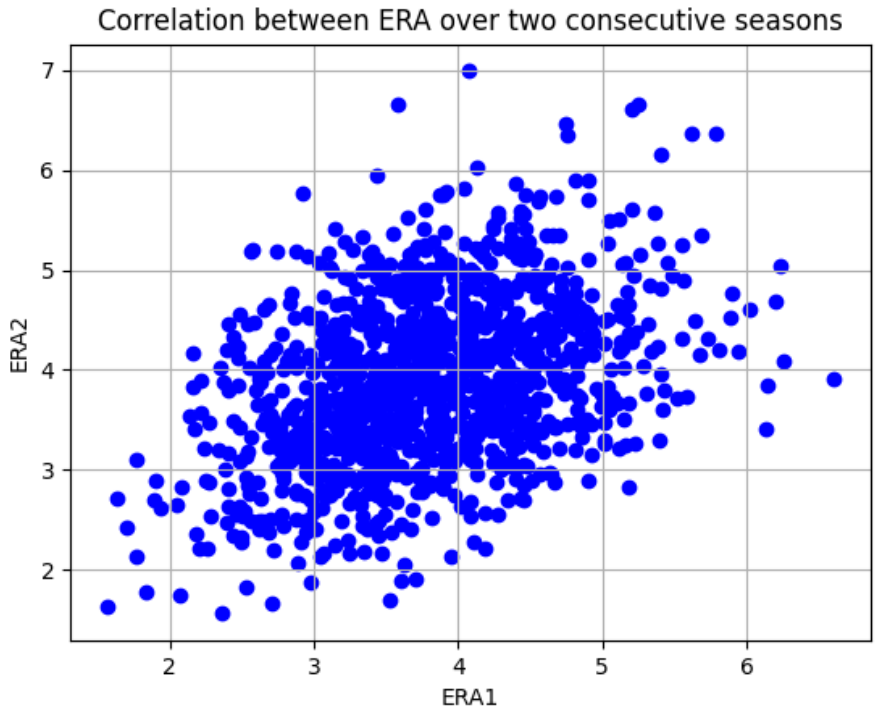
TRY

타자 스탯 중 HR%보다 다음 해 성적을 예측하는데 더 신뢰할 만한 스탯은?

지난해 ERA와 올해 ERA 간의 상관관계

ex8.py

```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
from scipy import stats
with sqlite3.connect("lahmansbaseballdb.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT playerID, yearID, ERA
    FROM pitching WHERE yearID >= 1990 and IPouts/3.0 >= 162 ORDER BY playerID;
    ''')
    result = cur.fetchall()

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

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

before = []
after = []

for i in range(len(df)-1):
    if df.iloc[i+1, 0] == df.iloc[i, 0]:
        if df.iloc[i+1, 1] == df.iloc[i, 1] + 1:
            before.append(df.iloc[i, 2])
            after.append(df.iloc[i+1, 2])

plt.scatter(before, after, c='b')
plt.title('Correlation between ERA over two consecutive seasons')
plt.xlabel('ERA1')
plt.ylabel('ERA2')
plt.grid(True)
plt.savefig('ex8_img.png')
correlation_coefficient = stats.pearsonr(before, after)
print("상관계수:", correlation_coefficient[0])
```




```
상관계수: 0.40008474159986634
```

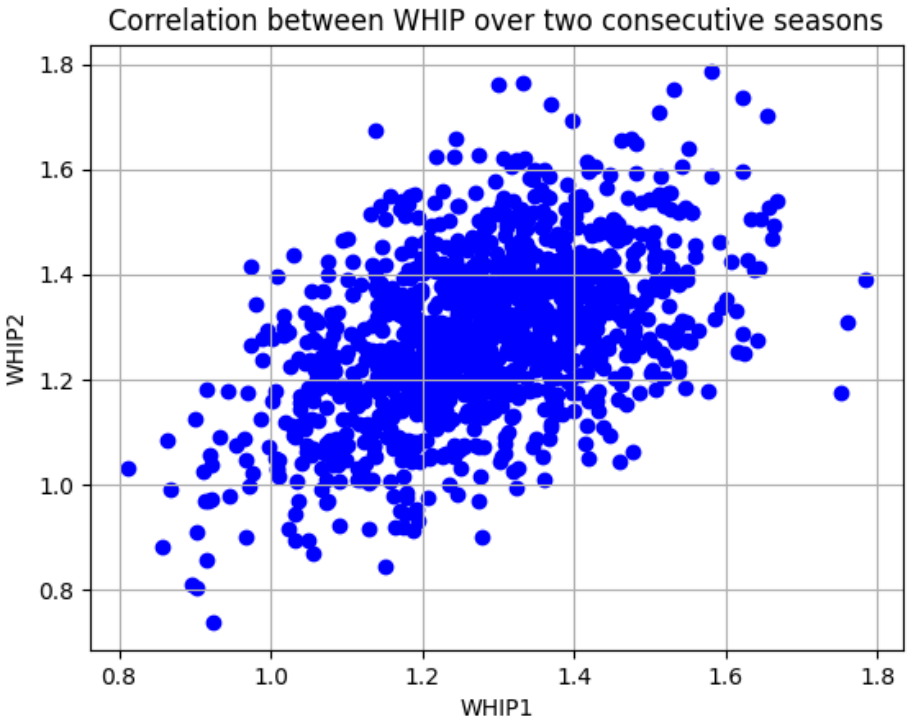
다소 강한 상관관계

특정해에 ERA가 좋았던 선수는 다음 해에도 ERA가 좋을 가능성이 있다

지난해 WHIP와 올해 WHIP 간의 상관관계

ex9.py

```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
from scipy import stats
with sqlite3.connect("lahmansbaseballdb.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT playerID, yearID, (H + BB)/(IPouts/3.0) AS WHIP
    FROM pitching WHERE yearID >= 1990 and IPouts/3.0 >= 162 ORDER BY playerID;
    ''')
    result = cur.fetchall()

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

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

before = []
after = []

for i in range(len(df)-1):
    if df.iloc[i+1, 0] == df.iloc[i, 0]:
        if df.iloc[i+1, 1] == df.iloc[i, 1] + 1:
            before.append(df.iloc[i, 2])
            after.append(df.iloc[i+1, 2])

plt.scatter(before, after, c='b')
plt.title('Correlation between WHIP over two consecutive seasons')
plt.xlabel('WHIP1')
plt.ylabel('WHIP2')
plt.grid(True)
plt.savefig('ex9_img.png')
correlation_coefficient = stats.pearsonr(before, after)
print("상관계수:", correlation_coefficient[0])
```



상관계수: 0.48334696579423914

다소 강한 상관관계

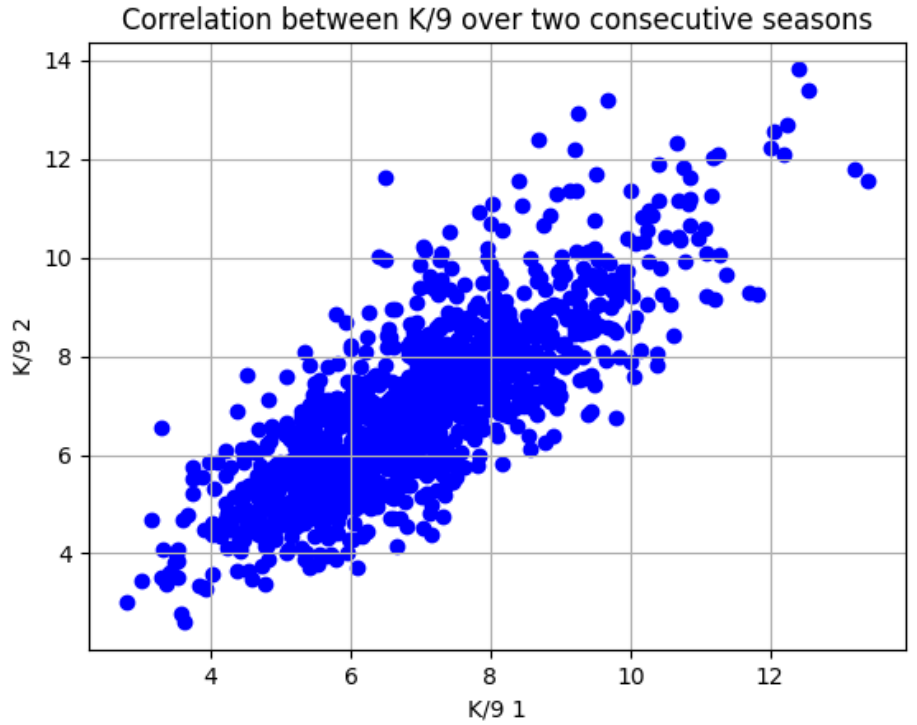
WHIP = (안타 + 볼넷)/이닝

WHIP > ERA

지난해 K/9와 올해 K/9 간의 상관관계

ex10.py

```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import font_manager, rc
from scipy import stats
with sqlite3.connect("lahmansbaseballdb.sqlite") as con:
    cur = con.cursor()
    cur.execute('''
    SELECT playerID, yearID, SO*9/(IPouts/3.0) AS "K/9"
    FROM pitching WHERE yearID >= 1990 and IPouts/3.0 >= 162 ORDER BY playerID;
    ''')
    result = cur.fetchall()

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

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

before = []
after = []

for i in range(len(df)-1):
    if df.iloc[i+1, 0] == df.iloc[i, 0]:
        if df.iloc[i+1, 1] == df.iloc[i, 1] + 1:
            before.append(df.iloc[i, 2])
            after.append(df.iloc[i+1, 2])

plt.scatter(before, after, c='b')
plt.title('Correlation between K/9 over two consecutive seasons')
plt.xlabel('K/9 1')
plt.ylabel('K/9 2')
plt.grid(True)
plt.savefig('ex10_img.png')
correlation_coefficient = stats.pearsonr(before, after)
print("상관계수:", correlation_coefficient[0])
```




```
상관계수: 0.8148357919780918
```

강한 상관관계

K/9 = 탈삼진*9/이닝

K/9 > WHIP > ERA

기본 | 확장 | 방어율 | 가치 | WP | 타석 | 타구

순	이름	팀	정렬 K/9	출장	이닝	ERA	FIP	K/9
1	안우진	22 키	10.67	11	70.0	2.31	2.15	10.67
2	루친스키	22 N	9.71	12	80.2	1.90	2.04	9.71
3	김광현	22 S	9.00	11	71.0	1.39	2.35	9.00
4	이의리	22 K	8.85	11	61.0	3.39	4.38	8.85
5	요키시	22 키	8.53	12	76.0	2.72	2.45	8.53
6	데스파이네	22 K	8.48	12	69.0	3.78	2.60	8.48
7	박세웅	22 롯	8.47	11	68.0	2.78	2.38	8.47
8	반즈	22 롯	8.40	14	90.0	2.60	3.01	8.40
9	고영표	22 K	8.39	11	74.0	2.80	2.56	8.39
10	켈리	22 L	8.30	10	59.2	2.72	2.93	8.30

2022년 6월 9일 스탯티즈 기준 2022 KBO K/9 순위

TRY

투수 스탯 중 K/9보다 다음 해 성적을 예측하는데 더 신뢰할 만한 스탯은?