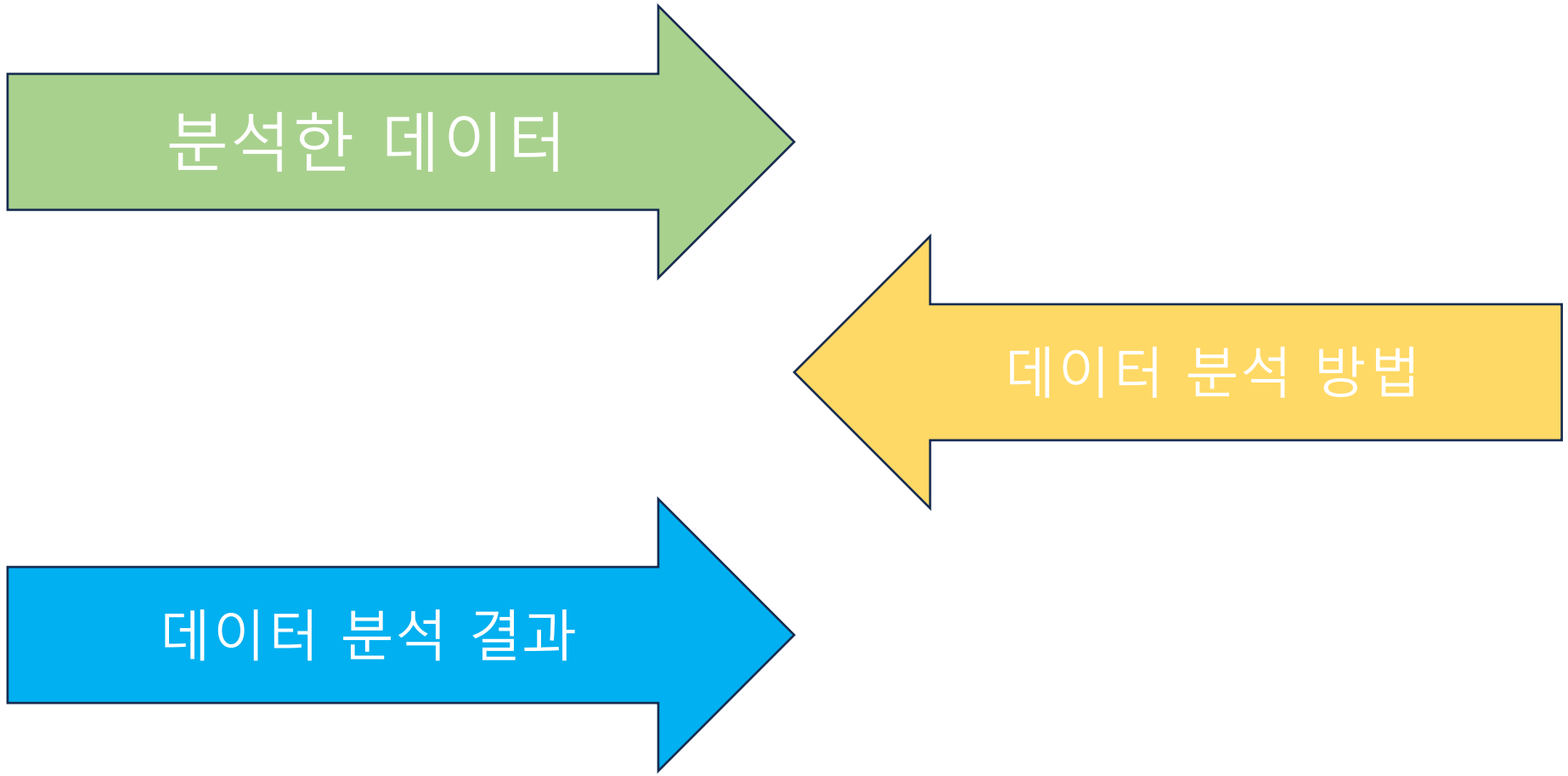
League of Legends 데이터 분석

김태훈

목차



분석한 데이터

- 2023_LOL_esports_match_data_from_OracleElixir.csv
- https://oracleselixir.com/tools/downloads
- 2023 리그오브레전드 이스포츠 대회 경기 데이터
- 경기에서 사용한 챔피언, 킬 ,데스, 더블 킬,
  몬스터 처치 횟수, 타워 파괴 횟수 등이 csv파일로
  정리되어있음

데이터 분석 방법 – 사용한 라이브러리

사용 라이브러리	이유
pandas	대용량 데이터 처리에 필요 효율적으로 데이터의 평균구하기 csv 파일 읽기
NumPy	사이킷런에서 linear regression에 필요한 데이터를 Numpy.array로 변경
matplotlib	그래프 그릴 때 필요
seaborn	그래프를 효율적으로 그릴 때 필요
scikit learn	Linear Regression을 사용하기 위해 필요

데이터 분석 방법 – 승률과 평균킬 구하기

M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
position	playernam	playerid	teamname	teamid	champion	ban1	ban2	ban3	ban4	ban5	gamelengt	result	kills
top	Wylenz	oe:player:6	Klanik Esp	oe:team:0a	Jax	Sylas	Caitlyn	Wukong	Akali	Yone	2612	1	
jng	Julbu	oe:player:f	Klanik Esp	oe:team:0a	Poppy	Sylas	Caitlyn	Wukong	Akali	Yone	2612	1	
mid	Sintax	oe:player:b	Klanik Esp	oe:team:0a	Taliyah	Sylas	Caitlyn	Wukong	Akali	Yone	2612	1	
bot	Axelent	oe:player:8	Klanik Esp	oe:team:0a	Ezreal	Sylas	Caitlyn	Wukong	Akali	Yone	2612	1	
sup	Wixo	oe:player:b	Klanik Esp	oe:team:0a	Karma	Sylas	Caitlyn	Wukong	Akali	Yone	2612	1	
top	Anathar	oe:player:0	MS Comp	oe:team:14	Sejuani	Galio	Lucian	Fiora	Viktor	Azir	2612	0	
jng	nicolaiy	oe:player:a	MS Comp	oe:team:14	Viego	Galio	Lucian	Fiora	Viktor	Azir	2612	0	
mid	Kuroneel	oe:player:7	MS Comp	oe:team:14	Syndra	Galio	Lucian	Fiora	Viktor	Azir	2612	0	
bot	Scripter	oe:player:0	MS Comp	oe:team:14	Zeri	Galio	Lucian	Fiora	Viktor	Azir	2612	0	
sup	Zimba	oe:player:4	MS Comp	oe:team:14	Yuumi	Galio	Lucian	Fiora	Viktor	Azir	2612	0	
team			Klanik Esp	oe:team:0a	de5e44c2	Sylas	Caitlyn	Wukong	Akali	Yone	2612	1	
team			MS Comp	oe:team:14	ad76b8d9	Galio	Lucian	Fiora	Viktor	Azir	2612	0	
top	S0bek	oe:player:c	beGenius	oe:team:60	K'Sante	Fiora	Kindred	Sejuani	Nautilus	Leona	2436	0	
jng	Stormax	oe:player:9	beGenius	oe:team:60	Xin Zhao	Fiora	Kindred	Sejuani	Nautilus	Leona	2436	0	
mid	Frostsize	oe:player:4	beGenius	oe:team:60	Ryze	Fiora	Kindred	Sejuani	Nautilus	Leona	2436	0	

데이터 분석 방법 – 승률과 평균킬 구하기

```
teamList = match_data_origin['teamid'].unique()

match_data_filtered = match_data_origin.loc[match_data_origin['position']=='team',['result','teamid','kills','doublekills','triplekill
match_data_filtered = match_data_filtered.dropna(axis=0)


match_data_filtered
```

1. 주어진 데이터에서 teamid 열을 추출하여 중복을 없애 teamList 추출
2. 팀의 각 경기 당 승리/패배 여부, 킬, 더블 킬, 드래곤 처치 횟수 등 추출
3. Nan제거(=match_data_filtered)

데이터 분석 방법 – 승률과 평균킬 구하기

	result	teamid	kills	doublekills	triplekills	quadrakills	pentakills	minionkills	monsterkills	dragons	...	heralds	opp_heralds	barons	opp_barons	t
10	1	oe:team:0ade5e44c23039bca133eee58ec1b83	13	1.0	1.0	0.0	0.0	1039.0	247	4.0	...	2.0	0.0	1.0	0.0	
11	0	oe:team:14ad76b8d9e647d4b29c3d26ecd29c9	7	1.0	0.0	0.0	0.0	1008.0	218	3.0	...	0.0	2.0	0.0	1.0	
22	0	oe:team:607baf04091e515e195644cb08ec21c	20	2.0	0.0	0.0	0.0	1111.0	225	3.0	...	2.0	0.0	2.0	1.0	
23	1	oe:team:2d9e95ecefb34c32ff1a997ebe372f9	16	1.0	0.0	0.0	0.0	1102.0	261	4.0	...	0.0	2.0	1.0	2.0	
34	1	oe:team:1dc6411295a36bd29c29b1096ba859d	20	2.0	2.0	0.0	0.0	876.0	231	4.0	...	0.0	2.0	0.0	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
119687	1	oe:team:1d1d0b0c76290957c7c9aa69a3ae4e7	27	4.0	1.0	0.0	0.0	724.0	185	4.0	...	1.0	1.0	1.0	0.0	
119698	1	oe:team:1d1d0b0c76290957c7c9aa69a3ae4e7	26	2.0	1.0	0.0	0.0	700.0	156	4.0	...	2.0	0.0	1.0	0.0	
119699	0	oe:team:8776b7033f1bc10634e50d90f861c1d	11	1.0	0.0	0.0	0.0	674.0	139	0.0	...	0.0	2.0	0.0	1.0	
119710	1	oe:team:1d1d0b0c76290957c7c9aa69a3ae4e7	40	4.0	0.0	0.0	0.0	654.0	173	0.0	...	1.0	1.0	1.0	0.0	
119711	0	oe:team:8776b7033f1bc10634e50d90f861c1d	21	4.0	0.0	0.0	0.0	671.0	150	4.0	...	1.0	1.0	0.0	1.0	

16610 rows × 25 columns

데이터 분석 방법 – 승률과 평균킬 구하기

```
winRate = defaultdict()
for team in teamList:
    teamFiltered = match_data_filtered.loc[match_data_filtered['teamid'] == team]
    winRate[team] = teamFiltered['result'].mean()


winRate
```

✓ 0.5s

1. 방금 추출한 팀 리스트를 순회하면서 match_data_filtered에서 그 팀의 데이터만 추출(=teamFiltered)
2. 승리는 1, 패배는 0이므로, teamFiltered['result']의 평균을 구하면 승률이 나옴.
3. 이것을 winRate[각 팀]에 저장

데이터 분석 방법 – 승률과 평균킬 구하기

```
defaultdict(None,
            {'oe:team:0ade5e44c23039bca133eee58ec1b83': 0.4444444444444444,
             'oe:team:14ad76b8d9e647d4b29c3d26ecd29c9': 0.16666666666666666,
             'oe:team:607baf04091e515e195644cb08ec21c': 0.59375,
             'oe:team:2d9e95ecefb34c32ff1a997ebe372f9': 0.4634146341463415,
             'oe:team:1dc6411295a36bd29c29b1096ba859d': 0.803921568627451,
             'oe:team:e05fc2a9f99e906ea564a672d75f839': 0.4444444444444444,
             'oe:team:e0ec1fed6e0949a0aef5f506ccb76ce': 0.4716981132075472,
             'oe:team:7380b73561abdd86c2391e9481acd4b': 0.4772727272727273,
             'oe:team:f9d5cf621532c5ac51e0947d7098b7f': 0.5555555555555556,
             'oe:team:ad003d7f74930eeab29a6029c3b9f52': 0.44680851063829785,
             'oe:team:4054ca135a8ad4383af9b6bed4fabcf': 0.07692307692307693,
             'oe:team:3eb6b5da983552dd32f84d7f0e99717': 0.6428571428571429,
             'oe:team:57ab4379bc2ebd5a0f7f53e791060cd': 0.23076923076923078,
             'oe:team:db4e98d03fec12b08cb7a80b739c446': 0.5263157894736842,
             'oe:team:294ae8f82e2061ab10ef6a3459bc02d': 0.3684210526315789,
             'oe:team:65f7d8b9f9b44f3476f64abf64c3c19': 0.4,
             'oe:team:479dc9903b6f233af09b0227170a0cc': 0.8461538461538461,
             'oe:team:0e99b04d101a93f6de17a4bf5de8b70': 0.42857142857142855,
             'oe:team:35d8c122cd2fd58cdb1208f823023af': 0.2,
             'oe:team:150ed90e9ca3262dbe6f3253b42ebb7': 0.717948717948718,
             'oe:team:853a1d4037366856c872b104ea2686d': 0.42857142857142855,
             'oe:team:1f75fda1d2b3b7aea236e4fd8a5dd4c': 0.7906976744186046,
             'oe:team:1ab54595dac94e054c4f3831f7ccb98': 0.4375,
             'oe:team:b62ca0a6c7279dc53fb2afa1946d284': 0.0,
...
             'oe:team:c94055c4eae45033b3e176ed7c35585': 0.38461538461538464,
```

데이터 분석 방법 – 승률과 평균킬 구하기

```
for team in teamList:
    teamFiltered = match_data_filtered.loc[match_data_filtered['teamid'] == team]
    killDict[team] = teamFiltered['kills'].mean()
    doubleDict[team] = teamFiltered['doublekills'].mean()
    tripleDict[team] = teamFiltered['triplekills'].mean()
    quadraDict[team] = teamFiltered['quadrakills'].mean()
    pentaDict[team] = teamFiltered['pentakills'].mean()
    minionDict[team] = teamFiltered['minionkills'].mean()
    monsterDict[team] = teamFiltered['monsterkills'].mean()
    dragonDict [team] = teamFiltered['dragons'].mean()
    oppDragonDict[team] = teamFiltered['opp_dragons'].mean()
    elementaldrakesDict[team] = teamFiltered['elementaldrakes'].mean()
    oppElementaldrakes[team] = teamFiltered['opp_elementaldrakes'].mean()
    elders[team]= teamFiltered['elders'].mean()
    oppElders [team]= teamFiltered['opp_elders'].mean()
    heralds[team]= teamFiltered['heralds'].mean()
    opp_heralds[team]= teamFiltered['opp_heralds'].mean()
    barons[team]= teamFiltered['barons'].mean()
    opp_barons[team]= teamFiltered['opp_barons'].mean()
    towers[team]= teamFiltered['towers'].mean()
    opp_towers [team]= teamFiltered['opp_towers'].mean()
    turretplates[team]= teamFiltered['turretplates'].mean()
    opp_turretplates [team]= teamFiltered['opp_turretplates'].mean()
    inhibitors[team]= teamFiltered['inhibitors'].mean()
    opp_inhibitors[team]= teamFiltered['opp_inhibitors'].mean()
```

킬 수, 드래곤 처치 횟수 등도 같은 방법으로 처리함

데이터 분석 방법 – 승률과 평균킬 구하기

```
for team in teamList:
    relationDf.loc[-1] = [team,winRate[team],killDict[team],doubleDict[team],tripleDict[team],quadraDict[team],pentaDict[team],minionDict[team],monsterDict[team],dragonDict[team],op
    relationDf.index = relationDf.index + 1

relationDf = relationDf.sort_index()
```

✓ 1.1s Python

```
relationDf
```

✓ 0.0s Python

	teamid	winRate	Kills	doubleKills	tripleKill	quadraKill	pentaKill	minionKill	monsterKill	dragons	...	heralds	opp_heralds	barons	opp_barons
0	oe:team:f9f16e15a84c85d050aa09b4a21c757	0.444444	17.333333	1.333333	0.111111	0.000000	0.000000	759.000000	189.111111	3.000000	...	0.555556	1.444444	0.555556	1.000000
1	oe:team:ccf3ac6ecc7f9a46f87a052deebe32e	0.000000	7.777778	0.888889	0.000000	0.000000	0.000000	686.111111	143.888889	0.555556	...	0.555556	1.444444	0.111111	1.000000
2	oe:team:b3407aff8f36af4d7492aceb5b6bfd6	0.454545	18.272727	1.727273	0.272727	0.000000	0.000000	685.181818	157.090909	1.636364	...	0.818182	1.181818	0.363636	0.818182
3	oe:team:27b676a1e4c8cf0fa2eb95072cf19a6	0.142857	11.571429	0.428571	0.000000	0.000000	0.000000	706.142857	148.142857	1.571429	...	0.428571	1.571429	0.142857	1.142857
4	oe:team:c94055c4eae45033b3e176ed7c35585	0.384615	13.076923	1.615385	0.230769	0.000000	0.000000	827.384615	178.461538	2.461538	...	1.384615	0.615385	0.615385	0.923077
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

구한 데이터를 팀별로 합침

데이터 분석 방법 – 승률과 평균킬 구하기

```
for team in teamList:
    relationDf.loc[-1] = [team,winRate[team],killDict[team],doubleDict[team],tripleDict[team],quadraDict[team],pentaDict[team],minionDict[team],monsterDict[team],dragonDict[team],op
    relationDf.index = relationDf.index + 1

relationDf = relationDf.sort_index()
```

✓ 1.1s Python

```
relationDf
```

✓ 0.0s Python

	teamid	winRate	Kills	doubleKills	tripleKill	quadraKill	pentaKill	minionKill	monsterKill	dragons	...	heralds	opp_heralds	barons	opp_barons
0	oe:team:f9f16e15a84c85d050aa09b4a21c757	0.444444	17.333333	1.333333	0.111111	0.000000	0.000000	759.000000	189.111111	3.000000	...	0.555556	1.444444	0.555556	1.000000
1	oe:team:ccf3ac6ecc7f9a46f87a052deebe32e	0.000000	7.777778	0.888889	0.000000	0.000000	0.000000	686.111111	143.888889	0.555556	...	0.555556	1.444444	0.111111	1.000000
2	oe:team:b3407aff8f36af4d7492aceb5b6bfd6	0.454545	18.272727	1.727273	0.272727	0.000000	0.000000	685.181818	157.090909	1.636364	...	0.818182	1.181818	0.363636	0.818182
3	oe:team:27b676a1e4c8cf0fa2eb95072cf19a6	0.142857	11.571429	0.428571	0.000000	0.000000	0.000000	706.142857	148.142857	1.571429	...	0.428571	1.571429	0.142857	1.142857
4	oe:team:c94055c4eae45033b3e176ed7c35585	0.384615	13.076923	1.615385	0.230769	0.000000	0.000000	827.384615	178.461538	2.461538	...	1.384615	0.615385	0.615385	0.923077
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

구한 데이터를 팀별로 합침

데이터 분석 방법 – 승률과 평균킬 구하기

```
col = ["Kills","doubleKills","tripleKill","quadraKill","pentaKill","minionKill","monsterKill",'dragons','opp_dragons','ele
sb.set(font_scale=1.5)
pos = []
for i in range(5):
    for j in range(5):
        pos.append((i,j))

fig,ax = plt.subplots(nrows=5,ncols=5,figsize=(30,30),constrained_layout=True)

for i in range(len(col)):
    sb.regplot(data=relationDf,x=col[i],y='winRate',ax=ax[pos[i][0],pos[i][1]])
    ax[pos[i][0],pos[i][1]].set_title(col[i])
```

✓ 11.8s

Regplot을 이용하여 각종 지표와 승률의 산점도(scatterplot)과 추세선(trendline)을 그림

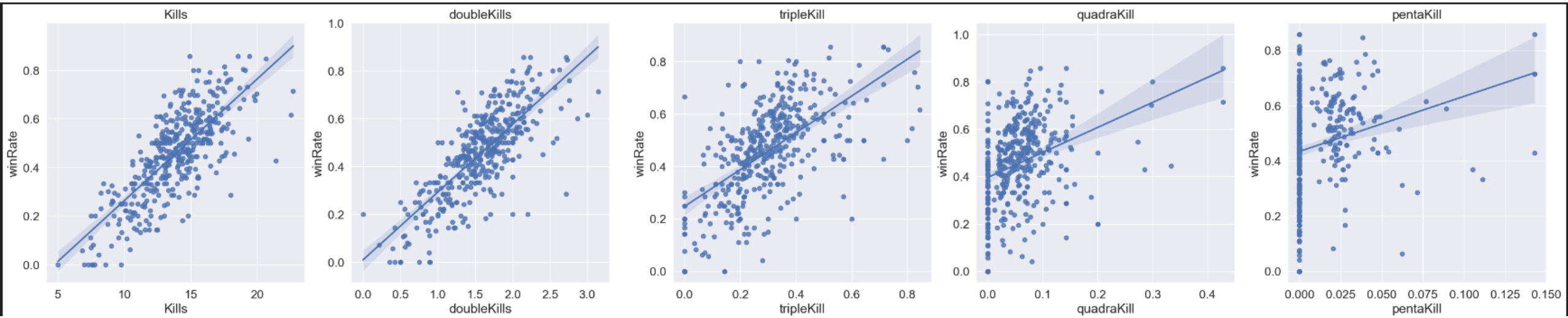
데이터 분석 방법 – 승률과 평균킬 구하기

```
#https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=dotorimj2&logNo=222117972039

trendData = pd.DataFrame(columns = ['category','coefficient','intercept','r2 score'])

for c in col:
    model=LinearRegression()
    var = np.array(relationDf[c]).reshape((-1,1))
    winRate = np.array(relationDf['winRate']).reshape((-1,1))
    model.fit(var,winRate)
    coef = model.coef_[0][0]
    intercept = model.intercept_[0]
    r2 = model.score(var,winRate)
    print(c + ' - coefficient of determination and r2 score is ',coef,r2)
    trendData.loc[-1] = [c,coef,intercept,r2]
    trendData.index +=1


trendData
```

✓ 0.0s

Scikit learn의 Linear Regression을 이용하여 각 지표와 승률의 정확한 상관관계 분석

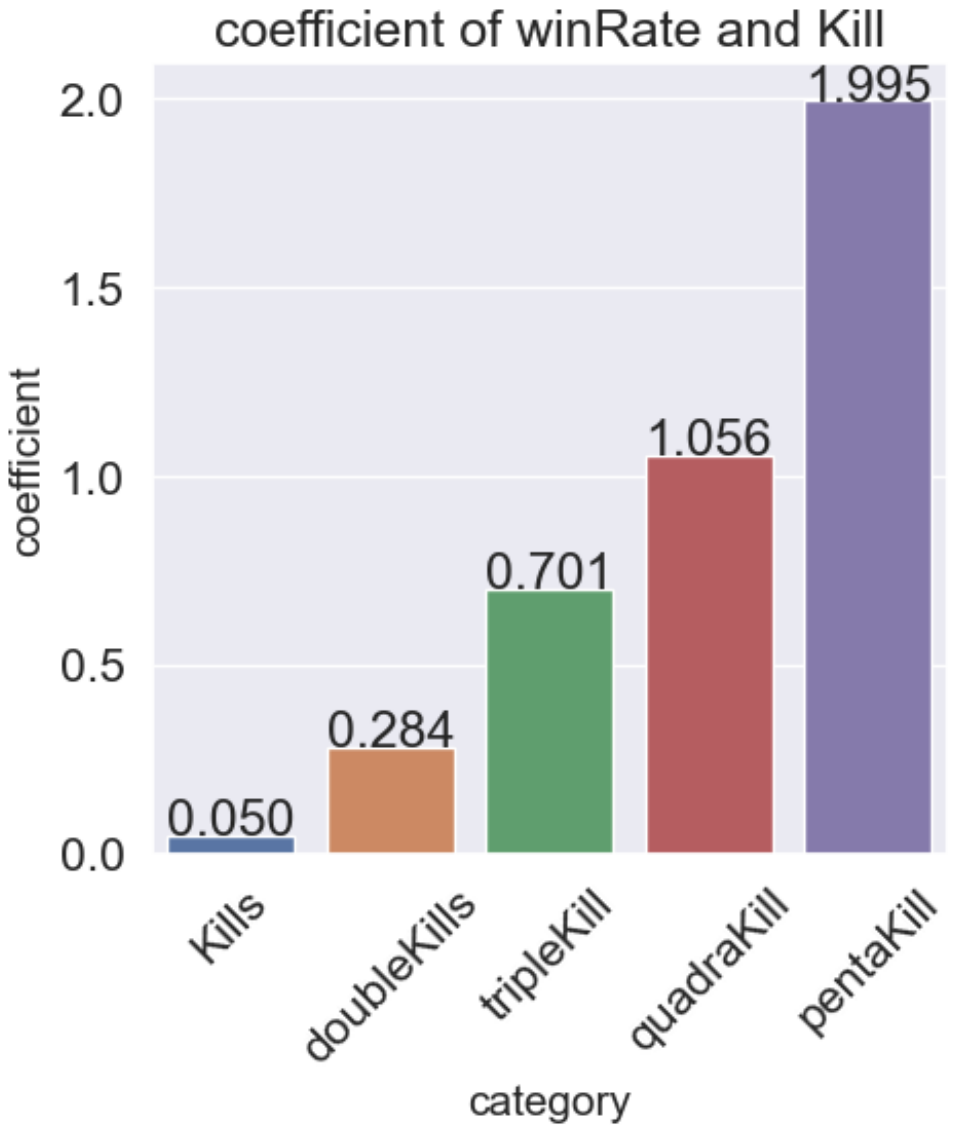
데이터 분석 결과 – 킬과 승률



category	coefficient	intercept	r2 score	form
Kills	0.050	-0.237	0.584	0.05x - 0.237
doubleKills	0.284	0.009	0.595	0.284x + 0.009
tripleKill	0.701	0.248	0.375	0.701x + 0.248
quadraKill	1.056	0.396	0.118	1.056x + 0.396
pentaKill	1.995	0.434	0.057	1.995x + 0.434

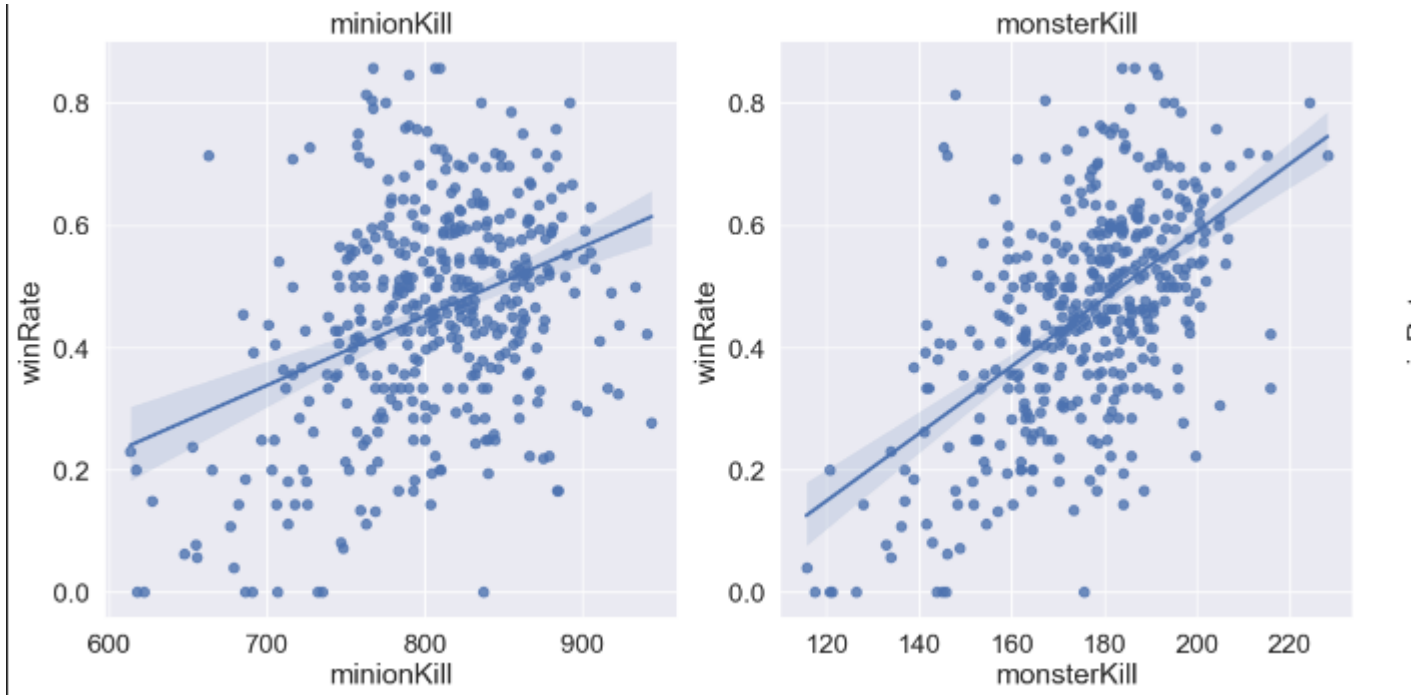
*넷째 자리에서 반올림

데이터 분석 결과 – 킬과 승률



*넷째 자리에서 반올림

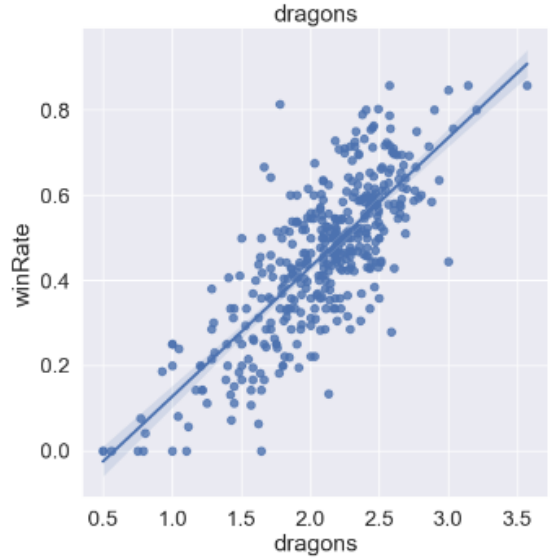
데이터 분석 결과 – 미니언, 몬스터와 승률

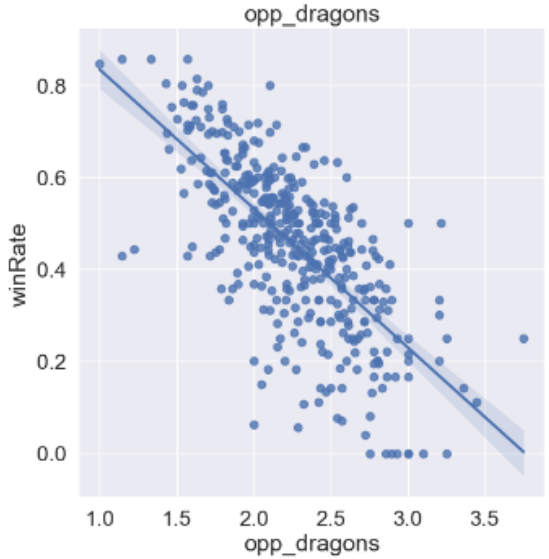


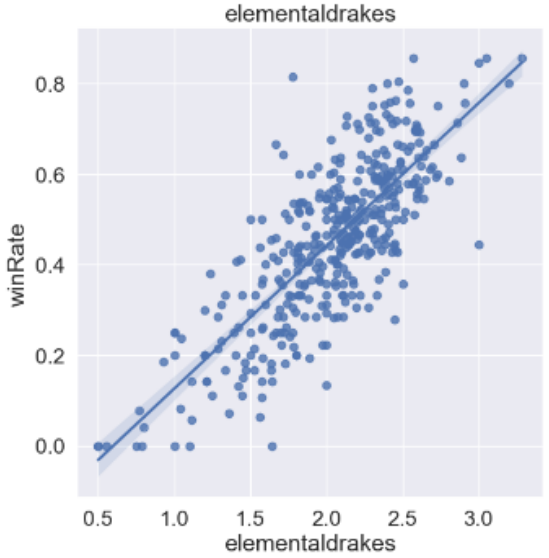
category	coefficient	intercept	r2 score	form
minionKill	0.001	-0.455	0.139	0.001x - 0.455
monsterKill	0.006	-0.512	0.313	0.006x - 0.512

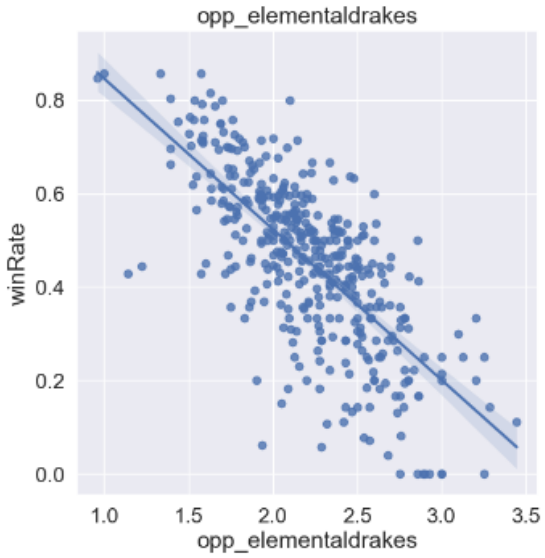
*넷째 자리에서 반올림

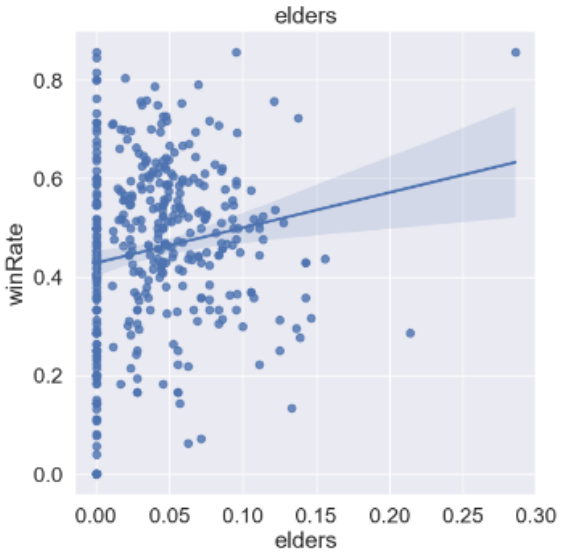
데이터 분석 결과 – 몬스터와 승률

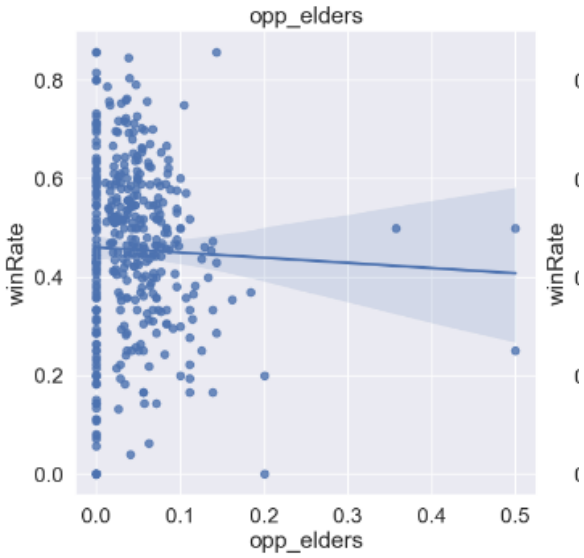




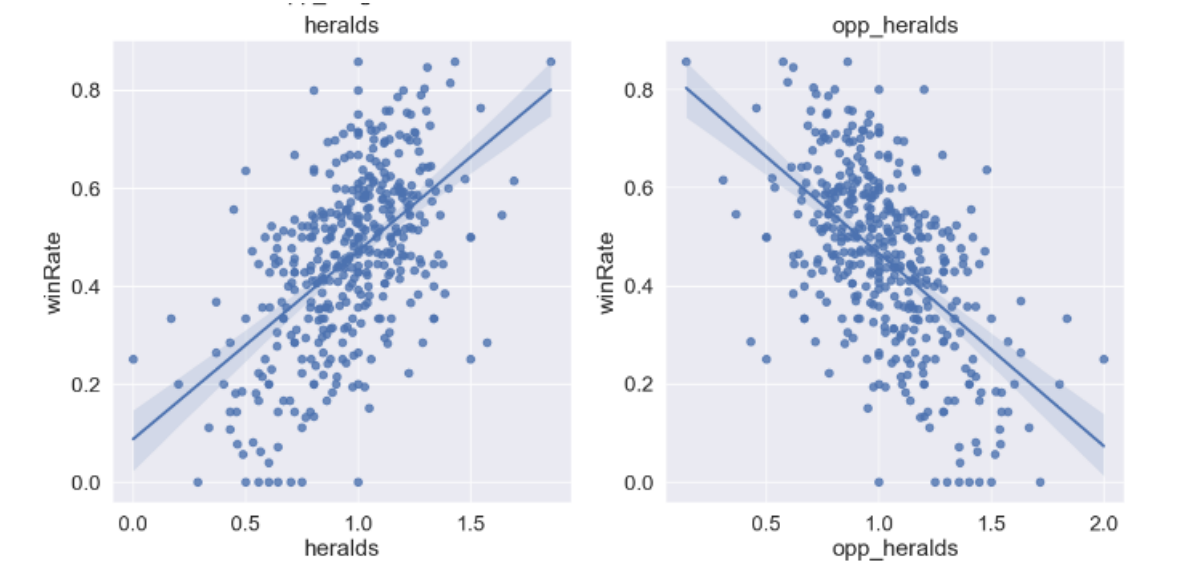


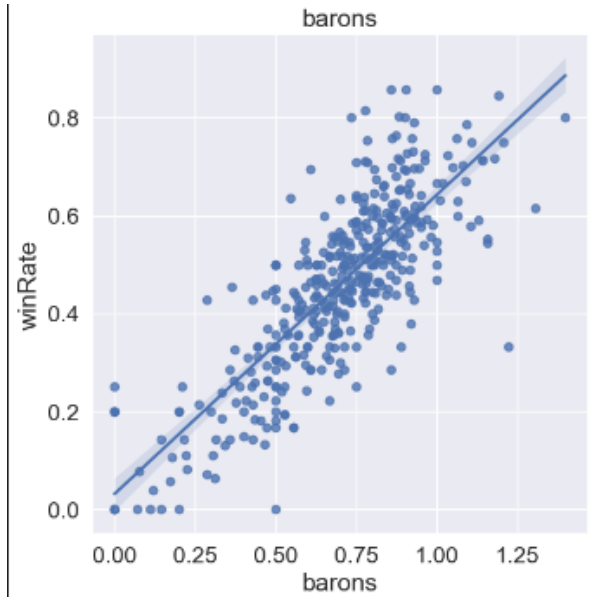






데이터 분석 결과 – 몬스터와 승률



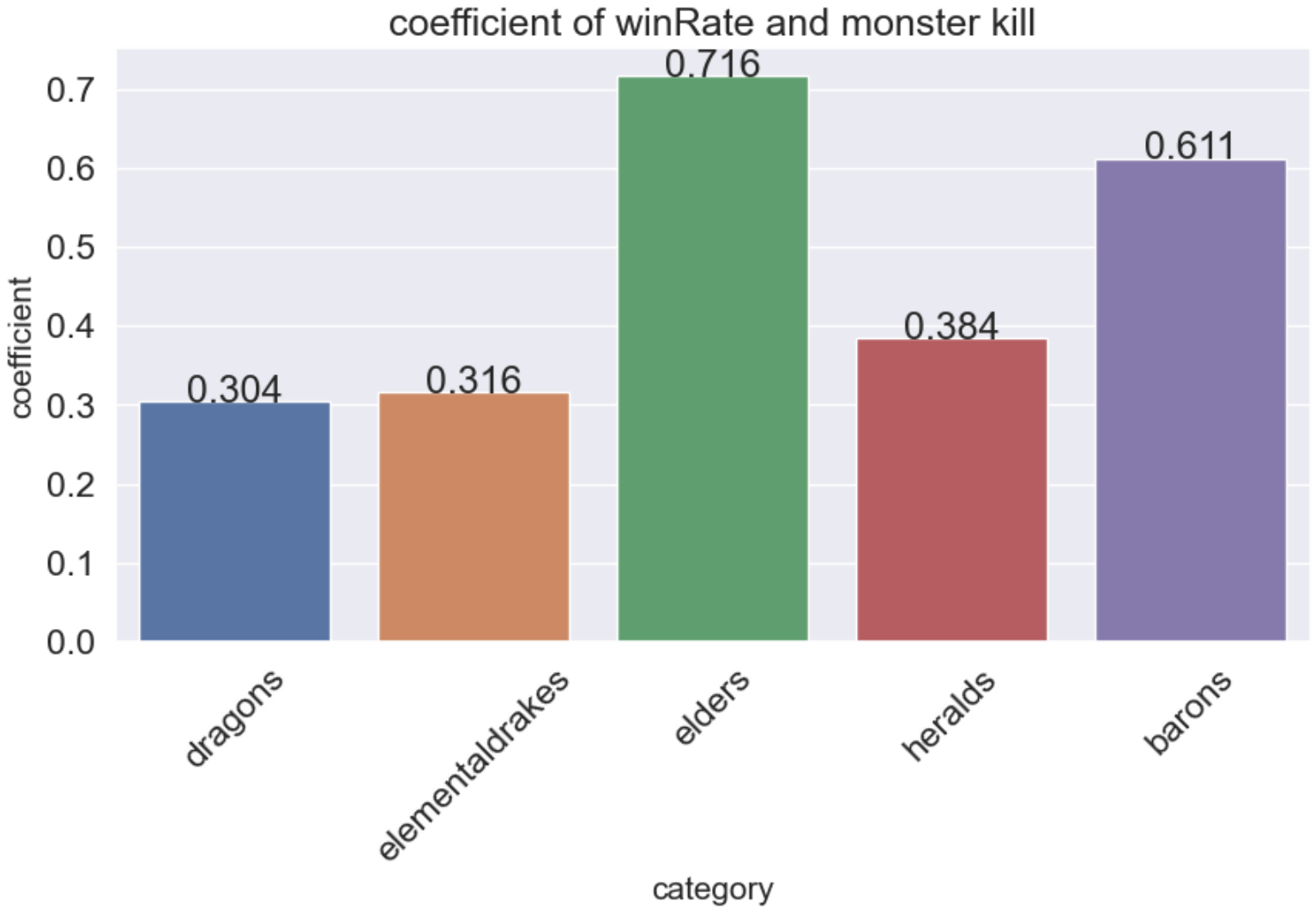


데이터 분석 결과 – 몬스터와 승률

category	coefficient	intercept	r2 score	form
dragons	0.304	-0.178	0.609	0.304x + -0.178
opp_dragons	-0.302	1.137	0.490	-0.302x + 1.137
elementaldrakes	0.316	-0.190	0.621	0.316x + -0.19
opp_elementaldrakes	-0.323	1.169	0.519	-0.323x + 1.169
elders	0.716	0.428	0.026	0.716x + 0.428
opp_elders	-0.105	0.460	0.001	-0.105x + 0.46
heralds	0.384	0.087	0.305	0.384x + 0.087
opp_heralds	-0.393	0.859	0.321	-0.393x + 0.859
barons	0.611	0.031	0.651	0.611x + 0.031
opp_barons	-0.643	0.961	0.545	-0.643x + 0.961

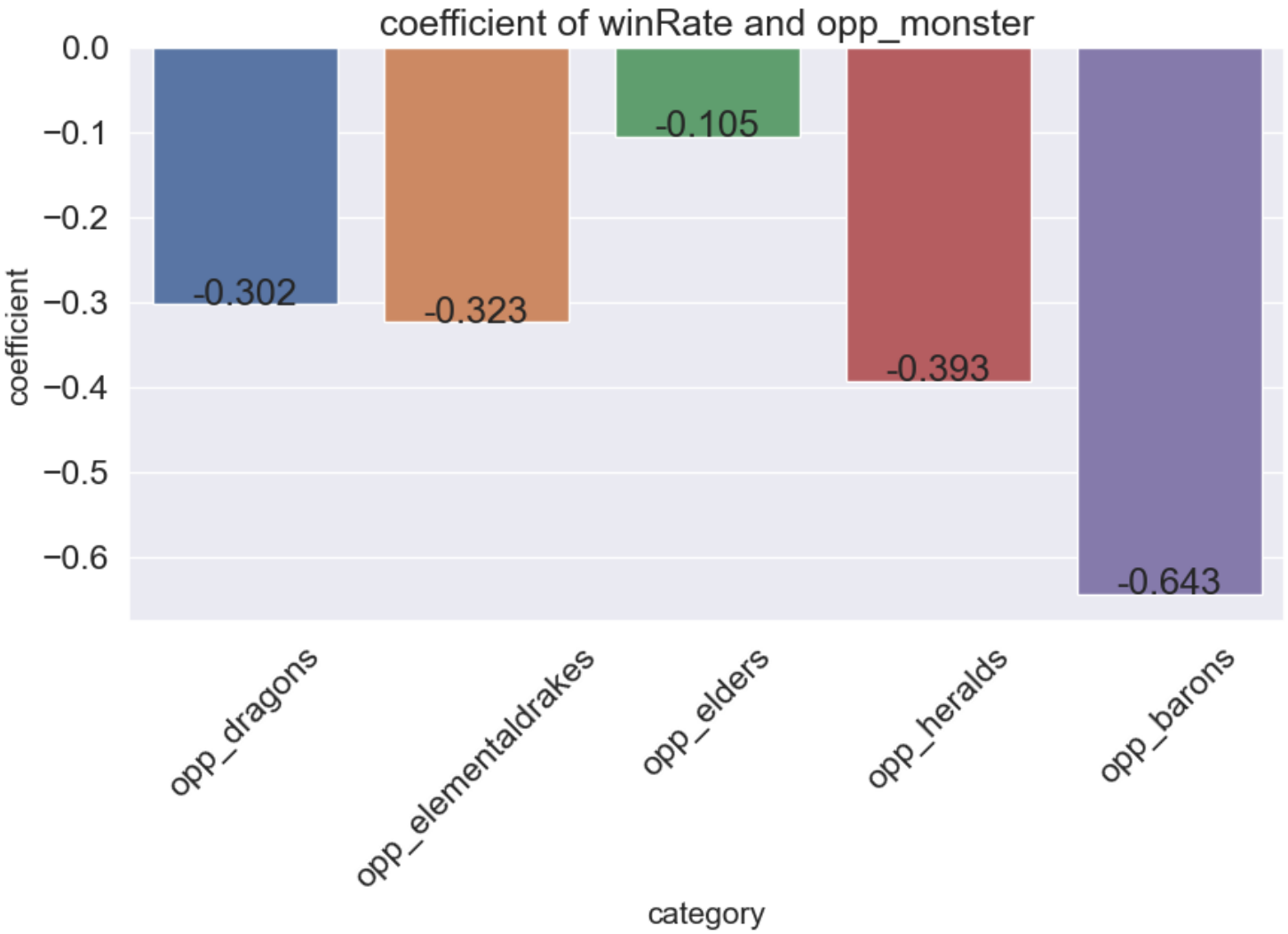
*넷째 자리에서 반올림

데이터 분석 결과 – 몬스터와 승률



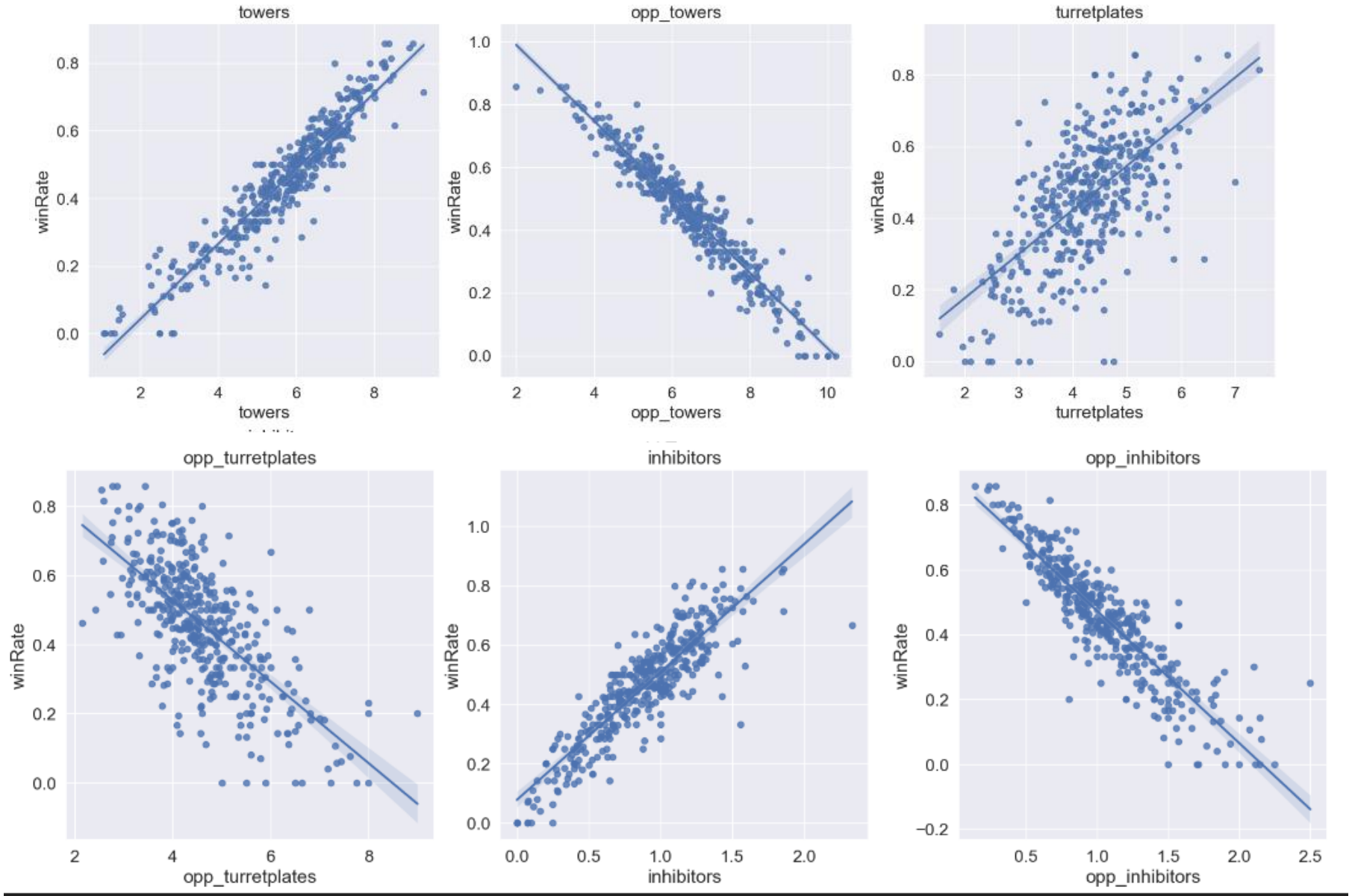
*넷째 자리에서 반올림

데이터 분석 결과 – 몬스터와 승률



*넷째 자리에서 반올림

데이터 분석 결과 – 파괴와 승률

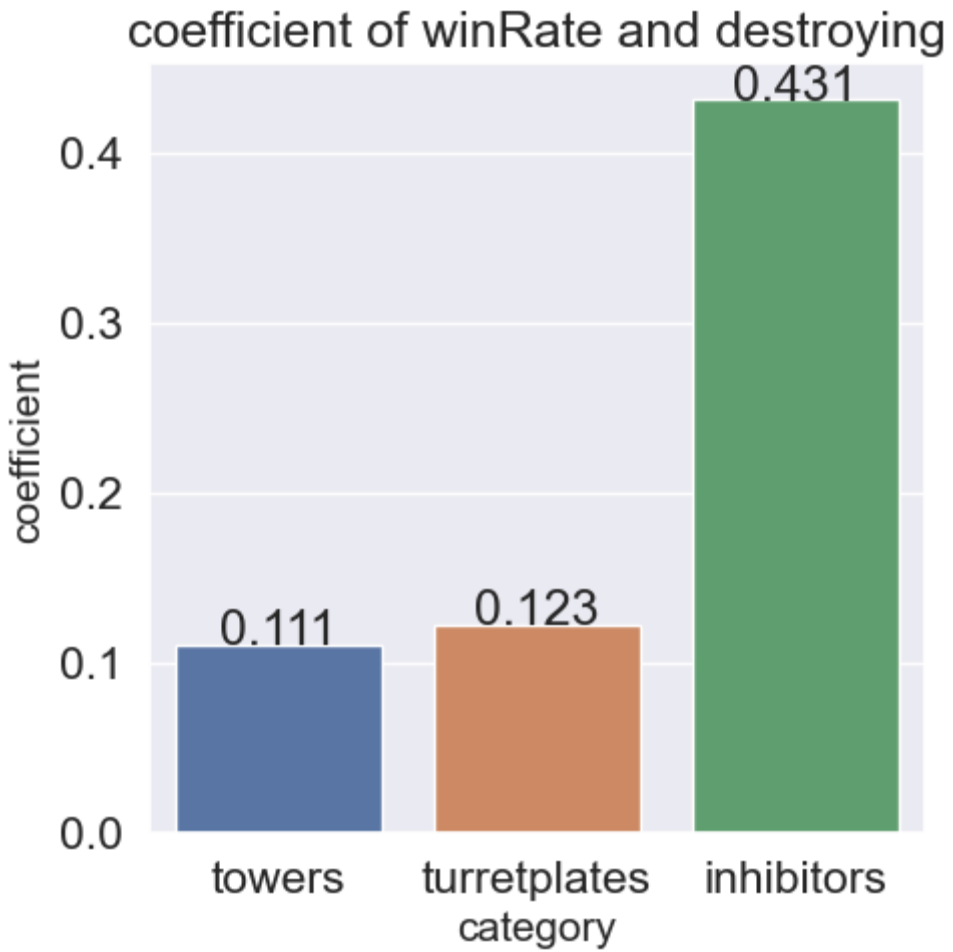


데이터 분석 결과 – 파괴와 승률

category	coefficient	intercept	r2 score	form
towers	0.111	-0.181	0.885	0.111x + -0.181
opp_towers	-0.121	1.230	0.919	-0.121x + 1.23
turretplates	0.123	-0.070	0.428	0.123x + -0.07
opp_turretplates	-0.118	0.999	0.468	-0.118x + 0.999
inhibitors	0.431	0.078	0.780	0.431x + 0.078
opp_inhibitors	-0.408	0.881	0.802	-0.408x + 0.881

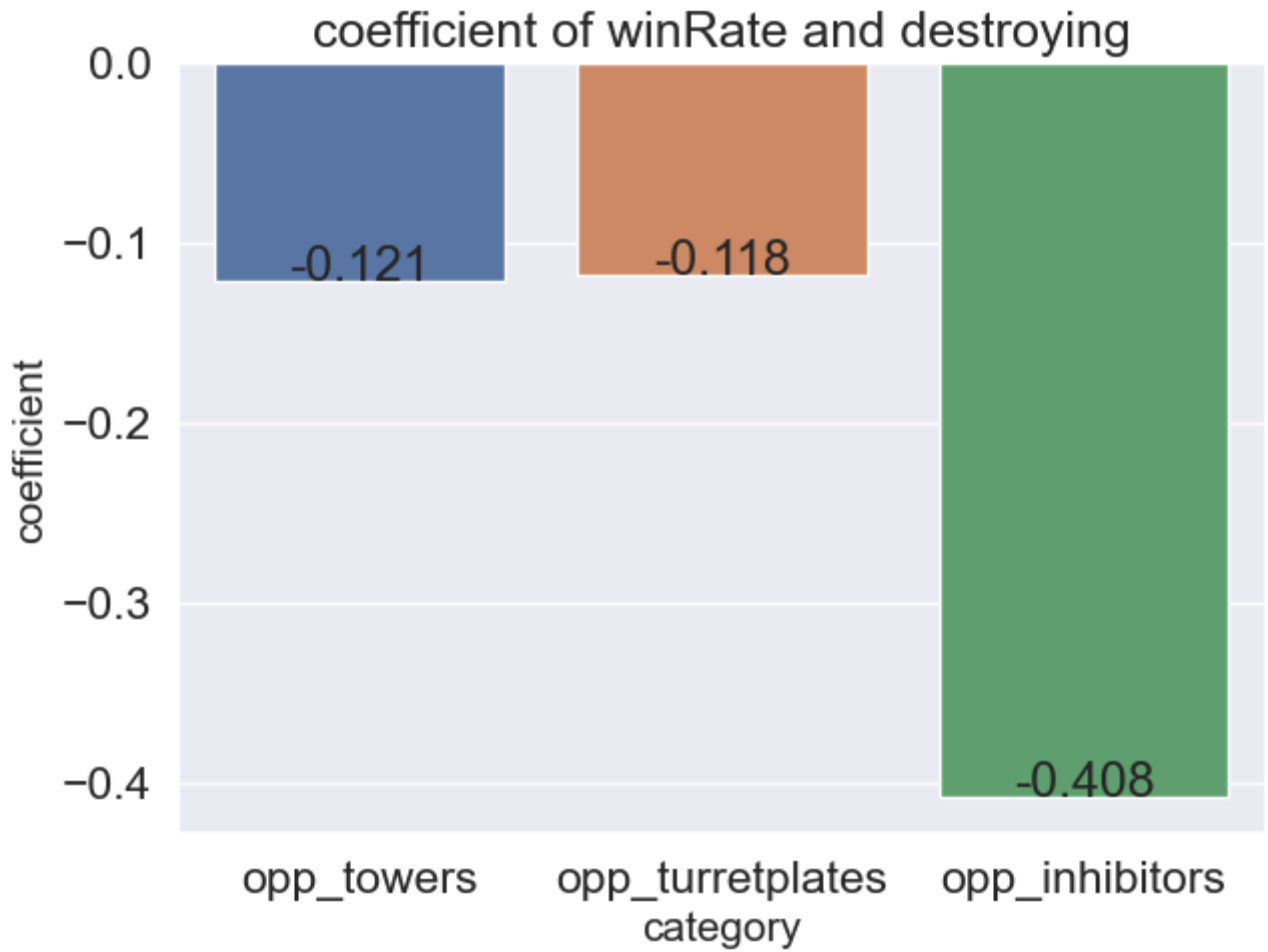
*넷째 자리에서 반올림

데이터 분석 결과 – 파괴와 승률



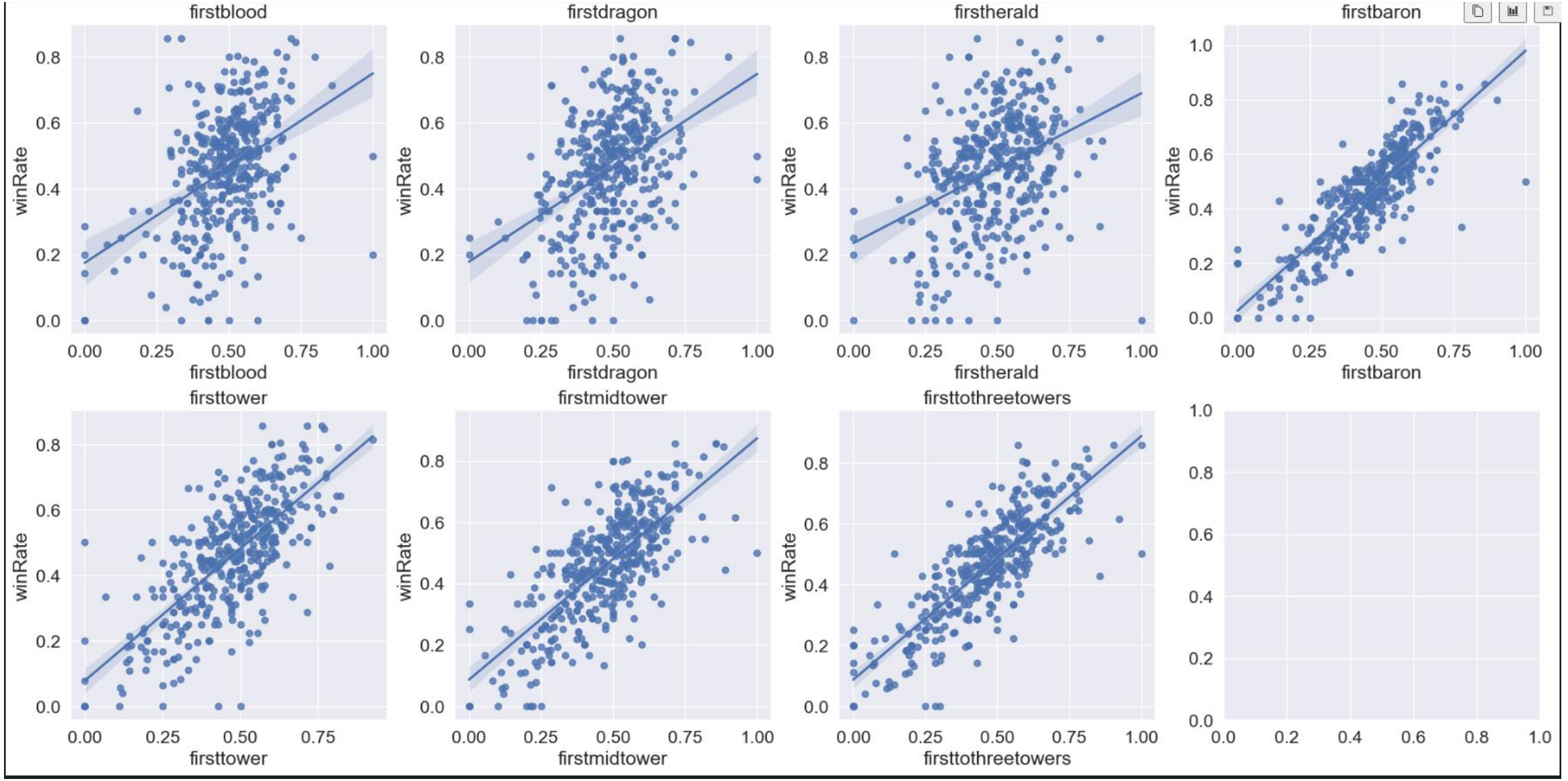
*넷째 자리에서 반올림

데이터 분석 결과 – 파괴와 승률



*넷째 자리에서 반올림

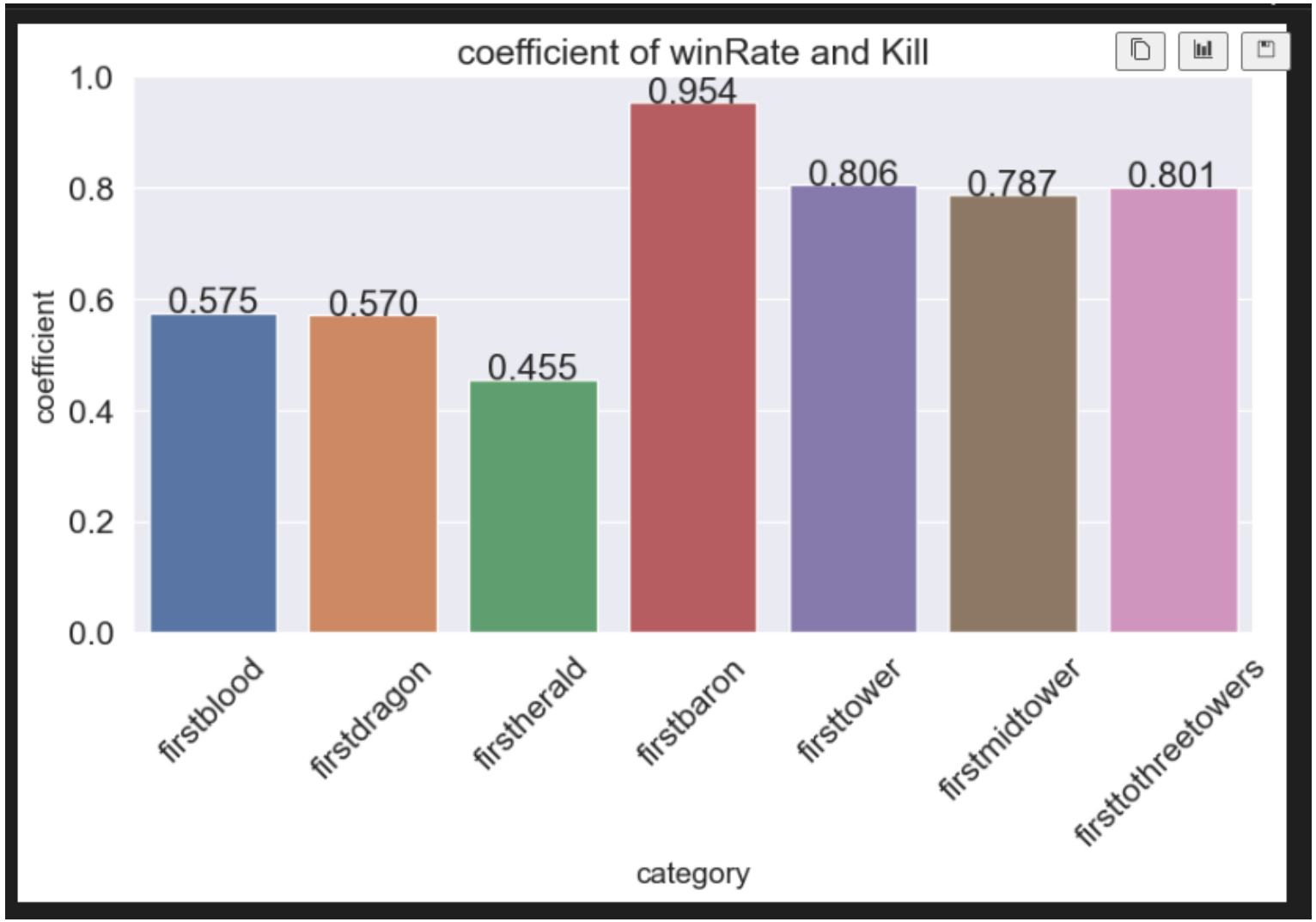
데이터 분석 결과 – 무엇을 첫번째로 하는 것이 좋을까



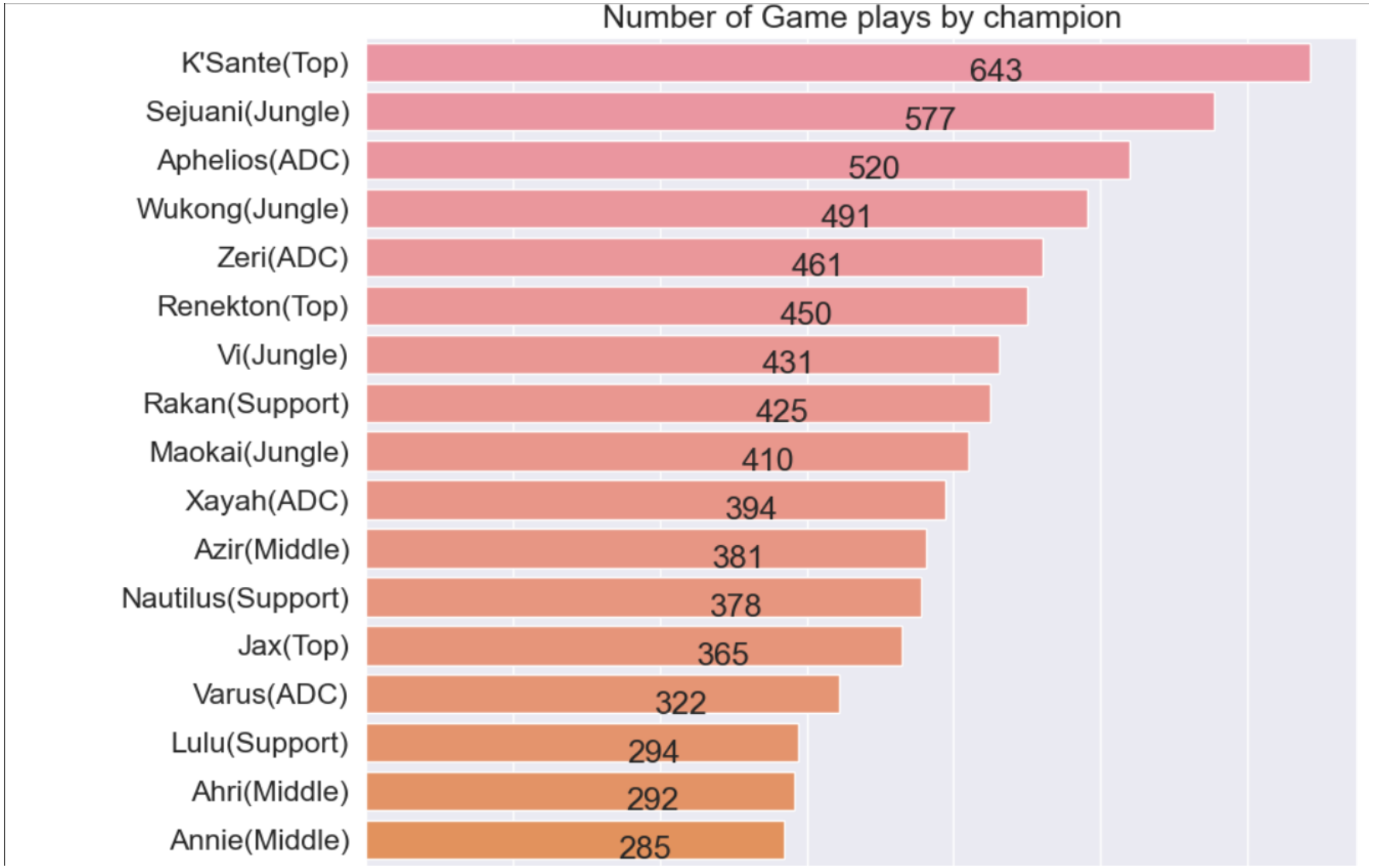
데이터 분석 결과 – 무엇을 첫번째로 하는 것이 좋을까

Category	Coefficient	Intercept	R2score	form
firstblood	0.575	0.175	0.185	0.575x + 0.175
firstdragon	0.570	0.179	0.182	0.57x + 0.179
firstherald	0.455	0.235	0.144	0.455x+0.235
firstbaron	0.954	0.026	0.730	0.954x+0.026
firsttower	0.806	0.078	0.522	0.806x+0.078
firstmidtower	0.787	0.087	0.529	0.787x+0.087
firsttothreetowers	0.801	0.087	0.671	0.801x+0.087

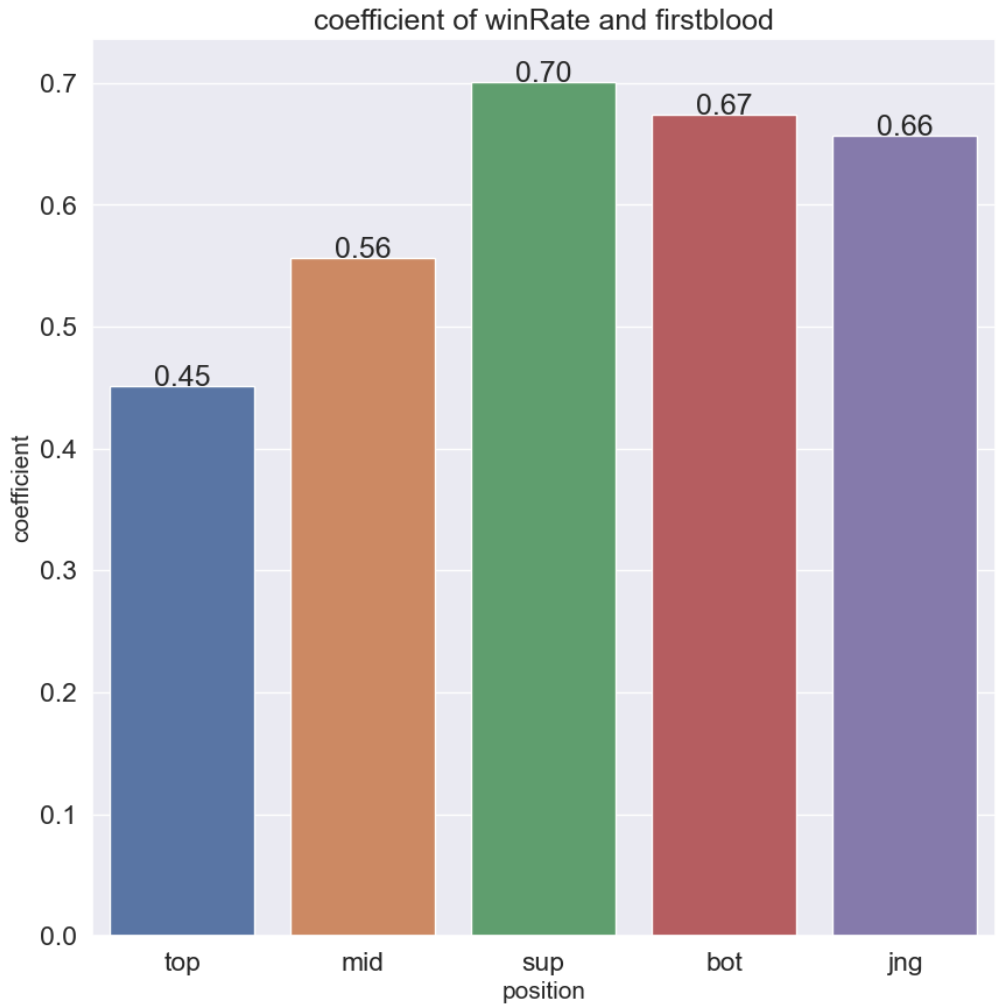
데이터 분석 결과 – 무엇을 첫번째로 하는 것이 좋을까



부록)데이터 분석 결과 – 챔피언 플레이 수



부록)데이터 분석 결과 – firstblood와 승률



*셋째 자리에서 반올림

Reference

링크	목적
Do it! 쉽게 배우는 파이썬 데이터 분석,김영우 저, 이지스 퍼블리싱	데이터 분석의 전반적인 방법
혼자 공부하는 머신러닝+딥러닝, 박해선 저, 한빛미디어	선형회귀 및 로지스틱 회귀를 구하는 방법
https://seaborn.pydata.org/	그래프 그리기
https://blog.naver.com/PostView.naver?blogId=kiddwannabe&logNo=222655678945&categoryNo=0&parentCategoryNo=0&currentPage=1	Seaborn 막대그래프에 값표시
https://mindscale.kr/course/python-visualization-basic/grid/	한번에 여러 그래프 그리기
https://blog.naver.com/PostView.naver?blogId=kiddwannabe&logNo=222655678945&categoryNo=0&parentCategoryNo=0&currentPage=1	Seaborn 막대그래프에 값표시
https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=dotorimj2&logNo=222117972039	산점도와 선형회귀선 그리기

Github

https://github.com/minchoCoin/LOL_data_analysis_prj

감사합니다

김태훈