

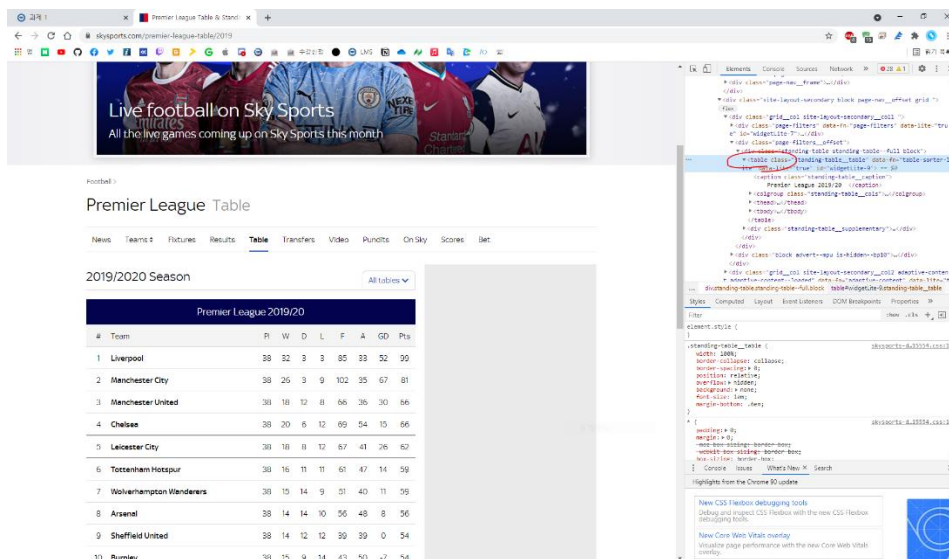


AI+X:R-Py컴퓨팅 과제1

이름: 홍재희 학번: 2021050000 소속: 경제금융학부

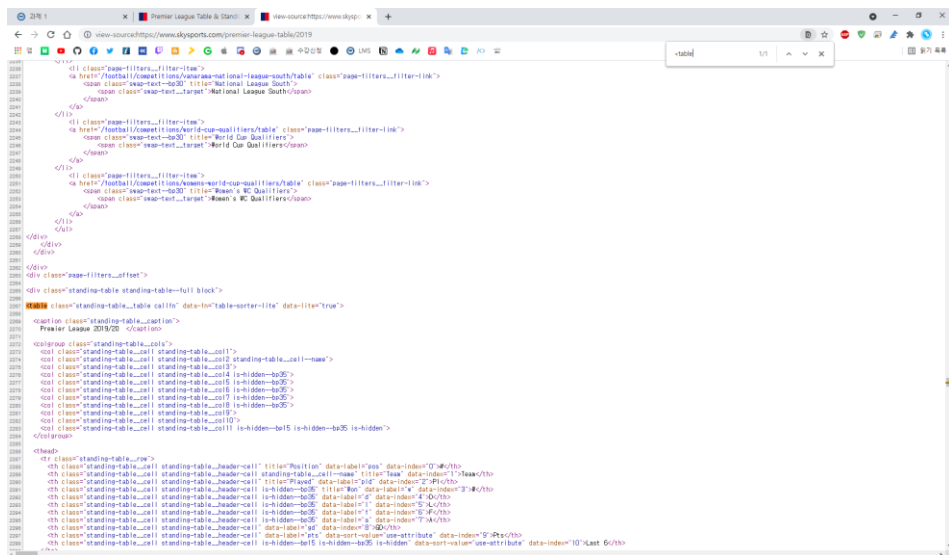
<Part1>

Q1.A



#	Team	R	W	D	L	F	A	GD	Pts
1	Liverpool	38	32	3	3	85	33	52	99
2	Manchester City	38	25	3	9	102	35	67	81
3	Manchester United	38	18	12	8	66	36	30	66
4	Chelsea	38	20	6	12	69	54	15	66
5	Leicester City	38	18	0	12	67	41	26	62
6	Tottenham Hotspur	38	16	11	11	65	47	14	59
7	Wolverhampton Wanderers	38	15	14	9	51	40	11	59
8	Arsenal	38	14	14	10	56	48	8	56
9	Sheffield United	38	14	12	12	39	59	0	54
10	Burnley	38	15	9	14	43	50	-7	54

Q1.B



#	Team	R	W	D	L	F	A	GD	Pts
1	Liverpool	38	32	3	3	85	33	52	99
2	Manchester City	38	25	3	9	102	35	67	81
3	Manchester United	38	18	12	8	66	36	30	66
4	Chelsea	38	20	6	12	69	54	15	66
5	Leicester City	38	18	0	12	67	41	26	62
6	Tottenham Hotspur	38	16	11	11	65	47	14	59
7	Wolverhampton Wanderers	38	15	14	9	51	40	11	59
8	Arsenal	38	14	14	10	56	48	8	56
9	Sheffield United	38	14	12	12	39	59	0	54
10	Burnley	38	15	9	14	43	50	-7	54

Q1.C

개발자 모드에서는 id 속성이 있지만, view-source에는 id 속성이 없다.

View-source에는 class 속성에 callfn이라는 값이 있지만, 개발자 모드의 class 속성에는 callfn 값이 없다.

Q2.

첨부파일 (Part1.py) 참고

실행결과

```
PS C:\Users\jhsol\Documents\과제\과제\컴퓨팅> & D:/Python39/python.exe "C:/Users/
{'#': [], 'Team': [], 'Pl': [], 'W': [], 'D': [], 'L': [], 'F': [], 'A': [], 'GD'
# Team Pl W D L F A GD Pts Last 6
0 1 Liverpool 38 32 3 3 85 33 52 99
1 2 Manchester City 38 26 3 9 102 35 67 81
2 3 Manchester United 38 18 12 8 66 36 30 66
3 4 Chelsea 38 20 6 12 69 54 15 66
4 5 Leicester City 38 18 8 12 67 41 26 62
5 6 Tottenham Hotspur 38 16 11 11 61 47 14 59
6 7 Wolverhampton Wanderers 38 15 14 9 51 40 11 59
7 8 Arsenal 38 14 14 10 56 48 8 56
8 9 Sheffield United 38 14 12 12 39 39 0 54
9 10 Burnley 38 15 9 14 43 50 -7 54
10 11 Southampton 38 15 7 16 51 60 -9 52
11 12 Everton 38 13 10 15 44 56 -12 49
12 13 Newcastle United 38 11 11 16 38 58 -20 44
13 14 Crystal Palace 38 11 10 17 31 50 -19 43
14 15 Brighton and Hove Albion 38 9 14 15 39 54 -15 41
15 16 West Ham United 38 10 9 19 49 62 -13 39
16 17 Aston Villa 38 9 8 21 41 67 -26 35
17 18 Bournemouth 38 9 7 22 40 65 -25 34
18 19 Watford 38 8 10 20 36 64 -28 34
19 20 Norwich City 38 5 6 27 26 75 -49 21
```

Q3.

첨부파일 (Part1.py) 참고

a: 0.3789473684210526 b: 0.005502392344497606

```
a: 0.3789473684210526 b: 0.005502392344497606
```

GD(골득실)가 높을수록, win_prob(승리 확률)이 올라가는 관계라고 생각한다.

Q4.

```
OLS Regression Results
=====
Dep. Variable: win_prob R-squared: 0.872
Model: OLS Adj. R-squared: 0.865
Method: Least Squares F-statistic: 122.4
Date: Tue, 18 May 2021 Prob (F-statistic): 1.85e-09
Time: 03:42:42 Log-Likelihood: 28.446
No. Observations: 20 AIC: -52.89
Df Residuals: 18 BIC: -50.90
Df Model: 1
Covariance Type: nonrobust
=====
coef std err t P>|t| [0.025 0.975]
-----
const 0.3789 0.014 27.552 0.000 0.350 0.408
GD 0.0055 0.000 11.062 0.000 0.004 0.007
=====
Omnibus: 10.433 Durbin-Watson: 1.561
Prob(Omnibus): 0.005 Jarque-Bera (JB): 7.864
Skew: 1.257 Prob(JB): 0.0196
Kurtosis: 4.765 Cond. No. 27.6
=====
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

절편이 0.3789, 기울기가 0.0055이므로, Q3에서 구했던 값과 크게 차이하지 않는다.

Q5.

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
	CRIM	ZN	INDUS CHAS NOX RM AGE DIS RAD TAX PTRATIO B LSTAT MEDV CAT. MEDV
0	0.00632	18	2.31 0 0.538 6.575 NaN 4.09 1 296 15.3 396.9 4.98 24 0
1	0.02731	0	7.07 0 <NA> 6.421 78.9 4.9671 2 242 17.8 396.9 9.14 21.6 0
2	0.02729	0	7.07 0 0.469 7.185 61.1 4.9671 2 242 17.8 392.83 4.03 34.7 1
3	0.03237	0	2.18 0 0.458 6.998 45.8 6.0622 3 222 18.7 394.63 2.94 33.4 1
4	0.06905	0	2.18 0 0.458 7.147 54.2 6.0622 3 222 18.7 396.9 5.33 36.2 1
..
501	0.06263	0	11.93 0 0.573 6.593 69.1 2.4786 1 273 21 391.99 9.67 22.4 0
502	0.04527	0	11.93 0 0.573 6.12 76.7 2.2875 1 273 21 396.9 9.08 20.6 0
503	0.06076	0	11.93 0 0.573 6.976 91 2.1675 1 273 21 396.9 5.64 23.9 0
504	0.10959	0	11.93 0 0.573 6.794 89.3 2.3889 1 273 21 393.45 6.48 22 0
505	0.04741	0	11.93 0 0.573 6.03 80.8 2.505 1 273 21 396.9 <NA> 11.9 0
[506 rows x 15 columns]			
	CRIM	ZN	INDUS CHAS NOX RM AGE DIS RAD TAX PTRATIO B LSTAT MEDV CAT. MEDV
2	0.02729	0.0	7.07 0.0 0.469 7.185 61.1 4.9671 2.0 242.0 17.8 392.83 4.03 34.7 1.0
3	0.03237	0.0	2.18 0.0 0.458 6.998 45.8 6.0622 3.0 222.0 18.7 394.63 2.94 33.4 1.0
4	0.06905	0.0	2.18 0.0 0.458 7.147 54.2 6.0622 3.0 222.0 18.7 396.90 5.33 36.2 1.0
5	0.02985	0.0	2.18 0.0 0.458 6.430 58.7 6.0622 3.0 222.0 18.7 394.12 5.21 28.7 0.0
6	0.08829	12.5	7.87 0.0 0.524 6.012 66.6 5.5605 5.0 311.0 15.2 395.60 12.43 22.9 0.0
..
499	0.17783	0.0	9.69 0.0 0.585 5.569 73.5 2.3999 6.0 391.0 19.2 395.77 15.10 17.5 0.0
501	0.06263	0.0	11.93 0.0 0.573 6.593 69.1 2.4786 1.0 273.0 21.0 391.99 9.67 22.4 0.0
502	0.04527	0.0	11.93 0.0 0.573 6.120 76.7 2.2875 1.0 273.0 21.0 396.90 9.08 20.6 0.0
503	0.06076	0.0	11.93 0.0 0.573 6.976 91.0 2.1675 1.0 273.0 21.0 396.90 5.64 23.9 0.0
504	0.10959	0.0	11.93 0.0 0.573 6.794 89.3 2.3889 1.0 273.0 21.0 393.45 6.48 22.0 0.0
[502 rows x 15 columns]			

(소스코드는 Part2.py 참고)

위는 결측치를 제거하기 전이고 아래는 결측치를 제거한 데이터 프레임입니다.

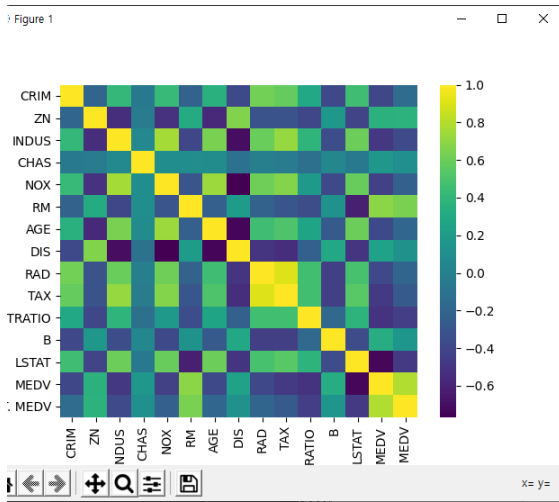
성공적으로 결측치를 제거했음을 확인할 수 있습니다.

Q6.

(소스코드는 Part2.py 참고)

[502 rows x 15 columns]															
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV	CAT. MEDV
count	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000	502.000000
mean	3.641708	11.418327	11.163765	0.069721	0.554802	6.284805	68.514542	3.797274	9.605578	409.095618	18.456574	356.353506	12.681514	22.564343	0.167331
std	8.629979	23.396912	6.873538	0.254930	0.116263	0.705085	28.247125	2.111828	8.717100	168.859125	2.165559	91.587527	7.155966	9.217580	0.373643
min	0.009060	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.730000	5.000000	0.000000
25%	0.082492	0.000000	5.190000	0.000000	0.449000	5.884250	44.550000	2.091150	4.000000	279.250000	17.400000	375.240000	6.950000	17.100000	0.000000
50%	0.262660	0.000000	9.690000	0.000000	0.538000	6.208500	77.150000	3.207450	5.000000	330.000000	19.050000	391.340000	11.395000	21.200000	0.000000
75%	3.689388	12.500000	18.100000	0.000000	0.624000	6.628000	94.100000	5.213925	24.000000	666.000000	20.200000	396.120000	17.057500	25.000000	0.000000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	37.970000	50.000000	1.000000

요약 통계량을 성공적으로 구했습니다.



seaborn으로 heatmap을 구현하여 matplotlib로 보여줬습니다.

Q7.

```
[[ -0.9789119]]
[34.9919874]
0.5394708106425741
41.117857183518886
29.565166011180708
```

(소스코드는 Part2.py참고)

위부터 차례로 coef_, intercept_, R^2, mean_squared_error 그리고 Test set을 예측해서 구한 mean_squared_error 값입니다.

Q8.

```
36.67167770307805
[-0.89162835 -0.00683497]
0.5505131096156948
40.1319573043212
30.328119505426738
```

(소스코드는 Part2.py 참고)

위부터 차례로 coef_, intercept_, R^2, mean_squared_error 그리고 Test set을 예측해서 구한 mean_squared_error 값입니다.