```
1 import pandas as pd
2 df = pd.read_csv('http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/baseball.csv')
```

11.07

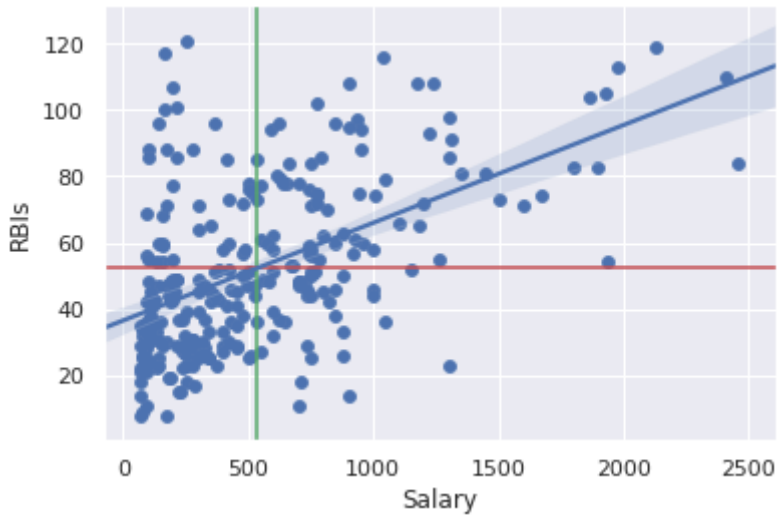
숨겨진 셀 23개

11.12

과업1

연봉값을 하는 선수 구하기

```
1 import seaborn as sns
2 import matplotlib.pyplot as plt
3
4 sns.scatterplot(x=df1['Salary'], y=df1['RBIs'])
5 plt.axvline(x=df1['Salary'].mean() , linestyle= '-', color = 'g')
6 plt.axhline(y=df1['RBIs'].mean() , linestyle= '-', color = 'r')
7 ax = sns.regplot(x=df1['Salary'], y=df1['RBIs'])
```




```
1 df1[['Salary','RBIs']].corr()
```

	Salary	RBIs
Salary	1.00000	0.51724
RBIs	0.51724	1.00000

```
1 df1.head(5)
```

	Player_Name	Team	TimesatBat	Hits	HomeRuns	Runs	RBIs	Walks	YearsinMLB
1	Ashby, Alan	Houston	315	81	7	24	38	39	14
2	Trevino, Alex	Los Angeles	202	53	4	31	26	27	9
3	Bochy, Bruce	San Diego	127	32	8	16	22	14	8
5	Brenly, Bob	San Francisco	472	116	16	60	62	74	6
6	Diaz, Bo	Cincinnati	474	129	10	50	56	40	10

```
import statsmodels.api as sm
y=df1['Salary']
X=sm.add_constant(df1[['RBIs', 'Walks', 'TimesatBat']])
model=sm.OLS(y, X).fit()
model.summary()
```

```
/usr/local/lib/python3.6/dist-packages/numpy/core/fromnumeric.py:2495: FutureWarning: Method .
  return ptp(axis=axis, out=out, **kwargs)
```

OLS Regression Results

Dep. Variable:	Salary	**R-squared:**	0.334
Model:	OLS	**Adj. R-squared:**	0.327
Method:	Least Squares	**F-statistic:**	43.39
Date:	Tue, 12 Nov 2019	**Prob (F-statistic):**	9.41e-23
Time:	05:04:57	**Log-Likelihood:**	-1926.5
No. Observations:	263	**AIC:**	3861.
Df Residuals:	259	**BIC:**	3875.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-55.8817	70.143	-0.797	0.426	-194.006	82.242
RBIs	5.8777	1.458	4.032	0.000	3.007	8.748
Walks	6.5073	1.359	4.788	0.000	3.831	9.184
TimesatBat	0.0319	0.274	0.116	0.907	-0.507	0.571

Omnibus:	36.802	**Durbin-Watson:**	1.821
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	60.937
Skew:	0.799	**Prob(JB):**	5.86e-14
Kurtosis:	4.735	**Cond. No.**	1.35e+03

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.35e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

유의확률이 0.05보다 작으므로 유의함.

Salary가 만불이 증가하면 RBIsrk 0.0295 약 3단위만큼 커져야함.

> Salary 만 보면 됨. P>|t|가 0.00 이므로 매우 유의함.

> 걸음 1단위에 6만5천불 타점 1단위당 5만9천불 정도 오른다.

Salary 만 보면 됨. P>|t|가 0.00 이므로 매우 유의함.

```
import math
math.sqrt(0.268) #결정 계수 min70%
```

```
0.5176871642217914
```

자료 불러오기

숨겨진 셀 9개

2016

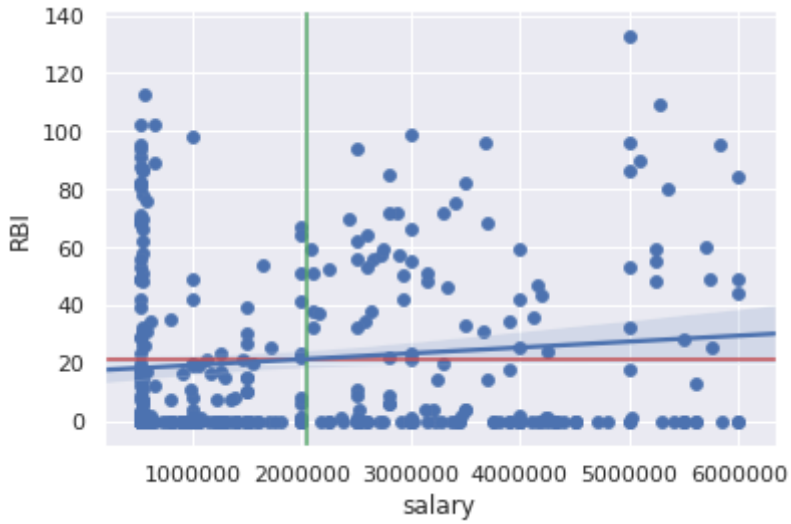
```
import pandas as pd

bat216=bat2[(bat2.yearID==2016)]
bat16 = bat216[(bat216.salary <= bat216.salary.quantile(0.75)) & (bat216.salary >= bat216.salary
bat16.shape
```

```
(369, 48)
```

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(x=bat16['salary'], y=bat16['RBI'])
plt.axvline(x=bat16['salary'].mean() , linestyle= '-', color = 'g')
plt.axhline(y=bat16['RBI'].mean() , linestyle= '-', color = 'r')
ax = sns.regplot(x=bat16['salary'], y=bat16['RBI'])
```



```
1 bat16.corr()
```

	yearID	salary	birthYear	birthMonth	birthDay	deathYear	deathMonth
yearID	NaN	NaN	NaN	NaN	NaN	NaN	NaN
salary	NaN	1.000000	-0.221827	-0.023874	-0.066686	-1.0	1.
birthYear	NaN	-0.221827	1.000000	-0.120074	-0.141645	-1.0	1.
birthMonth	NaN	-0.023874	-0.120074	1.000000	0.035979	-1.0	1.
birthDay	NaN	-0.066686	-0.141645	0.035979	1.000000	-1.0	1.
deathYear	NaN	-1.000000	-1.000000	-1.000000	-1.000000	1.0	-1.
deathMonth	NaN	1.000000	1.000000	1.000000	1.000000	-1.0	1.
deathDay	NaN	1.000000	1.000000	1.000000	1.000000	-1.0	1.
weight	NaN	0.116491	-0.149995	-0.049184	0.110609	-1.0	1.
height	NaN	-0.004337	-0.062646	-0.039868	0.011677	-1.0	1.
stint	NaN	NaN	NaN	NaN	NaN	NaN	NaN
G	NaN	0.121730	0.181103	0.097180	-0.069707	1.0	-1.
AB	NaN	0.103596	0.254445	0.071894	-0.089635	-1.0	1.
R	NaN	0.095104	0.262388	0.087132	-0.108881	-1.0	1.
H	NaN	0.110980	0.259310	0.070682	-0.110759	-1.0	1.
2B	NaN	0.113721	0.249404	0.054872	-0.133383	-1.0	1.
3B	NaN	0.039705	0.191217	0.099558	-0.032754	NaN	NaN
HR	NaN	0.087291	0.251468	0.079391	-0.088794	NaN	NaN
RBI	NaN	0.111421	0.261070	0.081403	-0.108784	-1.0	1.
SB	NaN	0.070735	0.184227	0.100162	-0.084441	NaN	NaN
CS	NaN	0.049199	0.234491	0.035896	-0.118747	NaN	NaN
BB	NaN	0.127744	0.212559	0.086061	-0.062223	-1.0	1.
SO	NaN	0.061562	0.245830	0.064506	-0.035000	-1.0	1.
IBB	NaN	0.151623	0.170041	0.041156	0.026272	NaN	NaN
HBP	NaN	0.070628	0.133461	-0.009050	-0.063349	NaN	NaN
SH	NaN	0.016802	0.104744	0.025801	0.015702	-1.0	1.
SF	NaN	0.119720	0.204643	0.065376	-0.077117	NaN	NaN
GIDP	NaN	0.132085	0.174469	0.118080	-0.082233	-1.0	1.
B23	NaN	0.107482	0.252042	0.064462	-0.123900	-1.0	1.
RR	NaN	0.104416	0.265266	0.085483	-0.110297	-1.0	1.

```
bat16['B23'] = bat16['2B'] + bat16['3B']
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/i
  """Entry point for launching an IPython kernel.
```

```
import statsmodels.api as sm
y=bat16['salary']
X=sm.add_constant(bat16[['GIDP','SF','BB','B23','G','AB','SH','IBB']])
model=sm.OLS(y, X).fit()
model.summary()
```

```
/usr/local/lib/python3.6/dist-packages/numpy/core/fromnumeric.py:2495: FutureWarning: Method .
  return ptp(axis=axis, out=out, **kwargs)
```

OLS Regression Results

Dep. Variable:	salary	**R-squared:**	0.037
Model:	OLS	**Adj. R-squared:**	0.015
Method:	Least Squares	**F-statistic:**	1.705
Date:	Tue, 12 Nov 2019	**Prob (F-statistic):**	0.0957
Time:	05:51:39	**Log-Likelihood:**	-5792.1
No. Observations:	369	**AIC:**	1.160e+04
Df Residuals:	360	**BIC:**	1.164e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.738e+06	2.04e+05	8.526	0.000	1.34e+06	2.14e+06
GIDP	6.083e+04	3.81e+04	1.598	0.111	-1.4e+04	1.36e+05
SF	1.713e+04	6.83e+04	0.251	0.802	-1.17e+05	1.52e+05
BB	5145.6040	1.03e+04	0.499	0.618	-1.51e+04	2.54e+04
B23	2.063e+04	2.62e+04	0.786	0.432	-3.1e+04	7.22e+04
G	4752.2447	4405.726	1.079	0.281	-3911.949	1.34e+04
AB	-3801.7945	2298.408	-1.654	0.099	-8321.788	718.199
SH	1.743e+04	4.57e+04	0.382	0.703	-7.24e+04	1.07e+05
IBB	7.867e+04	5.07e+04	1.551	0.122	-2.11e+04	1.78e+05

Omnibus:	36.828	**Durbin-Watson:**	2.105
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	42.475
Skew:	0.801	**Prob(JB):**	5.98e-10
Kurtosis:	2.557	**Cond. No.**	699.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.