```
1 !pip install -U finance-datareader
```

```
Requirement already up-to-date: finance-datareader in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied, skipping upgrade: requests-file in /usr/local/lib/python3.6/dis
Requirement already satisfied, skipping upgrade: requests>=2.3.0 in /usr/local/lib/python3.6/d
Requirement already satisfied, skipping upgrade: lxml in /usr/local/lib/python3.6/dist-package
Requirement already satisfied, skipping upgrade: pandas>=0.19.2 in /usr/local/lib/python3.6/di
Requirement already satisfied, skipping upgrade: six in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied, skipping upgrade: idna<2.9,>=2.5 in /usr/local/lib/python3.6/di
Requirement already satisfied, skipping upgrade: urllib3<1.25,>=1.21.1 in /usr/local/lib/pytho
Requirement already satisfied, skipping upgrade: chardet<3.1.0,>=3.0.2 in /usr/local/lib/pytho
Requirement already satisfied, skipping upgrade: certifi>=2017.4.17 in /usr/local/lib/python3.
Requirement already satisfied, skipping upgrade: pytz>=2017.2 in /usr/local/lib/python3.6/dist
Requirement already satisfied, skipping upgrade: python-dateutil>=2.6.1 in /usr/local/lib/pyth
Requirement already satisfied, skipping upgrade: numpy>=1.13.3 in /usr/local/lib/python3.6/dis
```

```
1 import FinanceDataReader as fdr
2 # 한국거래소 상장종목 전체
3 df_krx = fdr.StockListing('KRX')
```

```
1 df_krx.head()
2 #Symbol 종목코드
```

	Symbol	Name	Sector	
0	155660	DSR	1차 비철금속 제조업	
1	001250	GS글로벌	상품 종합 도매업	수출입업(시멘트,철강금속,전기전자,섬유,기
2	082740	HSD엔진	일반 목적용 기계 제조업	
3	011070	LG이노텍	전자부품 제조업	
4	010060	OCI	기초 화학물질 제조업	타르제품,카본블랙,무수프탈산,농약원제,석탄화학제

```
1 name=input('주식 코드 기업명 ?') #find 종목코드 및 정보
2 df_krx[df_krx['Name'].str.contains(name)]
```

주식 코드 기업명 ?삼성

	Symbol	Name	Sector	
34	006400	삼성SDI	일차전지 및 축전지 제조업	칼라브라운관
35	207940	삼성바이오로직스	기초 의약물질 및 생물학적 제제 제조업	
36	068290	삼성출판사	서적, 잡지 및 기타 인쇄물 출판업	
37	029780	삼성카드	기타 금융업	신용
38	000810	삼성화재해상보험	보험업	
623	018260	삼성에스디에스	컴퓨터 프로그래밍, 시스템 통합 및 관리업	
624	010140	삼성중공업	선박 및 보트 건조업	
625	016360	삼성증권	금융 지원 서비스업	
932	028260	삼성물산	기타 전문 도매업	
933	005930	삼성전자	통신 및 방송 장비 제조업	IMT2000 서
934	001360	삼성제약	의약품 제조업	
1214	006660	삼성공조	자동차 신품 부품 제조업	
1215	032830	삼성생명	보험업	
1216	009150	삼성전기	전자부품 제조업	영상,음향,통신장ㅂ
1508	028050	삼성엔지니어링	건축기술, 엔지니어링 및 관련 기술 서비스업	
1623	309930	삼성머스트스팩3호	금융 지원 서비스업	
1913	291230	삼성스팩2호	금융 지원 서비스업	

```
1 import pandas as pd
2 import datetime
```

```
1 ss=fdr.DataReader('005930', '1995-01-01')
2 ss.head(3)
```

Date	Open	High	Low	Close	Volume	Change
1996-02-12	1755	1768	1735	1756	65570	NaN
1996-02-13	1728	1742	1708	1742	173490	-0.007973
1996-02-14	1735	1755	1708	1736	160300	-0.003444

셀트리온 주가정보를 불러오시오.

```
1 name=input('주식 코드 기업명 ?') #find 종목코드 및 정보
2 df_krx[df_krx['Name'].str.contains(name)]
```

```
df_krx[df_krx['Name'].str.contains(name)]
```

주식 코드 기업명 ?셀트리온

	Symbol	Name	Sector	Industry
947	068270	셀트리온	기초 의약물질 및 생물학적 제제 제조업	램시마, 트룩시마, 허쥬마
1061	068760	셀트리온제약	의약품 제조업	무선인터넷서비스및플랫폼

```
df=fdr.DataReader('068270', '2019-01-01')
df2=fdr.DataReader('005930', '2019-01-01')
```

퀴즈1

코스피 지수 삼성전자 005930

데이터 합치기

```
import pandas as pd
import datetime
today=datetime.datetime.today()
# KS11 (KOSPI 지수), 2018년~현재
kospi = fdr.DataReader('KS11', '2019')
#삼성전자 주가(2018.01.01~현재)
ss=fdr.DataReader('005930', '2019-01-01', today)
```

```
import matplotlib.pyplot as plt
```

```
fin=pd.concat([kospi['Close'],ss['Close']], axis=1)
fin.set_axis(['kospi','ss'],axis=1,inplace=True)
fin.head(3)
```

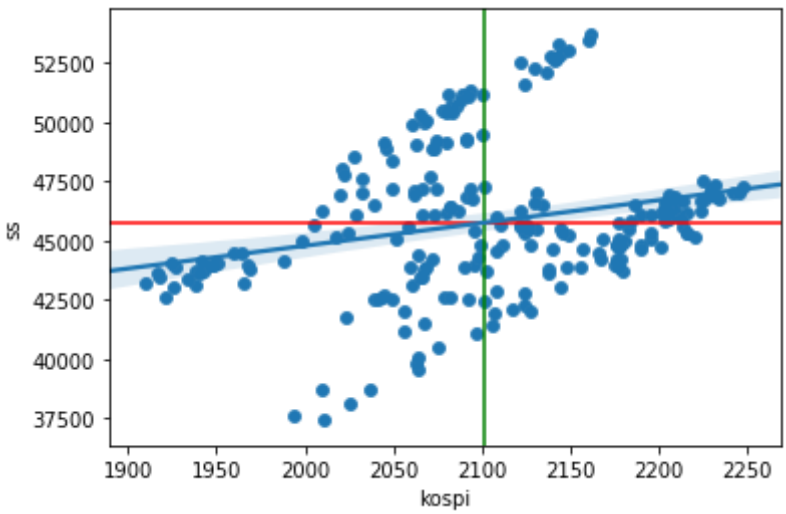
Date	kospi	ss
2019-01-02	2010.00	38750
2019-01-03	1993.70	37600
2019-01-04	2010.25	37450

kospi가 x일 때

1. 산점도 그리고 해석

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

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




```
fin[['kospi','ss']].corr()
```

	kospi	ss
kospi	1.000000	0.254122
ss	0.254122	1.000000

해석

양의 방향으로 선형관계를 갖는다.

2. 기울기 b 측정하기

> 9.5963

3. R 제곱 구하기

```
import statsmodels.api as sm
y=fin['ss']
X=sm.add_constant(fin['kospi'])
model=sm.OLS(y, X).fit()
model.summary()
```

```
/usr/local/lib/python3.6/dist-packages/numpy/core/fromnumeric.py:2495: FutureWarning:

Method .ptp is deprecated and will be removed in a future version. Use numpy.ptp instead.
```

OLS Regression Results

Dep. Variable:	ss	**R-squared:**	0.065
Model:	OLS	**Adj. R-squared:**	0.060
Method:	Least Squares	**F-statistic:**	14.91
Date:	Tue, 19 Nov 2019	**Prob (F-statistic):**	0.000149
Time:	05:47:47	**Log-Likelihood:**	-2052.3
No. Observations:	218	**AIC:**	4109.
Df Residuals:	216	**BIC:**	4115.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.559e+04	5227.563	4.896	0.000	1.53e+04	3.59e+04
kospi	9.5963	2.485	3.862	0.000	4.698	14.494

Omnibus:	9.022	**Durbin-Watson:**	0.032
Prob(Omnibus):	0.011	**Jarque-Bera (JB):**	8.977
Skew:	0.482	**Prob(JB):**	0.0112
Kurtosis:	3.244	**Cond. No.**	5.45e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.45e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

유의수준 > 0.001

R-squared : 6.5%

kospi의 영향을 많이 받지않음.

수익률

1. 산점도 그리고 해석

```
#fin.set_index('Date',inplace=True)
fin0 = fin.diff(axis=0 , periods=1)/fin.shift(periods=1)
```

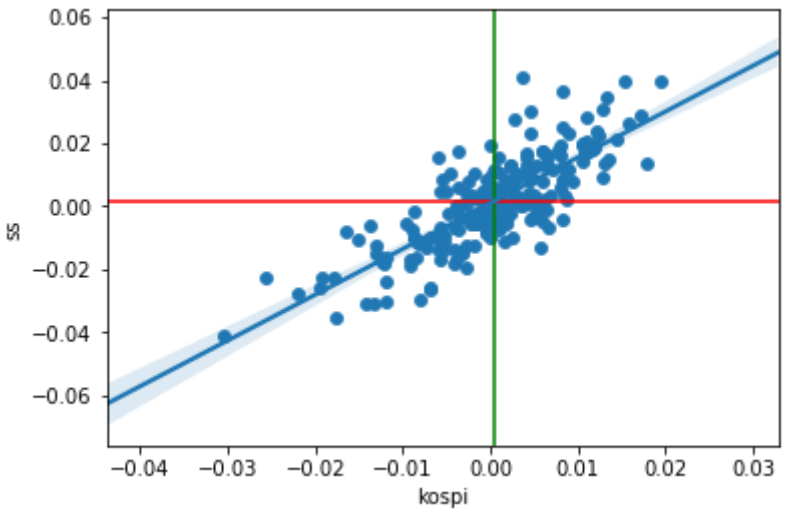
```
fin1 = fin0.dropna()
```

```
fin1.head()
```

Date	kospi	ss
2019-01-03	-0.008109	-0.029677
2019-01-04	0.008301	-0.003989
2019-01-07	0.013357	0.034713
2019-01-08	-0.005807	-0.016774
2019-01-09	0.019474	0.039370

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

sns.scatterplot(x=fin1['kospi'], y=fin1['ss'])
plt.axvline(x=fin1['kospi'].mean() , linestyle= '-', color = 'g')
plt.axhline(y=fin1['ss'].mean() , linestyle= '-', color = 'r')
ax = sns.regplot(x=fin1['kospi'], y=fin1['ss'])
```




```
fin1[['kospi','ss']].corr()
```

	kospi	ss
kospi	1.000000	0.792053
ss	0.792053	1.000000

해석

양의 방향으로 선형관계를 갖는다.

2. 기울기 b 측정하기

1.4527

3. R 제곱 구하기

```
1 import statsmodels.api as sm
2 y=fin1.ss
3 X=sm.add_constant(fin1.kospi)
4 model=sm.OLS(y, X).fit()
5 model.summary()
```

```
/usr/local/lib/python3.6/dist-packages/numpy/core/fromnumeric.py:2495: FutureWarning:

Method .ptp is deprecated and will be removed in a future version. Use numpy.ptp instead.
```

OLS Regression Results

Dep. Variable:	ss	**R-squared:**	0.627
Model:	OLS	**Adj. R-squared:**	0.626
Method:	Least Squares	**F-statistic:**	361.9
Date:	Tue, 19 Nov 2019	**Prob (F-statistic):**	5.63e-48
Time:	05:48:02	**Log-Likelihood:**	718.61
No. Observations:	217	**AIC:**	-1433.
Df Residuals:	215	**BIC:**	-1426.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0011	0.001	1.752	0.081	-0.000	0.002
kospi	1.4527	0.076	19.025	0.000	1.302	1.603

Omnibus:	11.285	**Durbin-Watson:**	1.761
Prob(Omnibus):	0.004	**Jarque-Bera (JB):**	12.530
Skew:	0.451	**Prob(JB):**	0.00190
Kurtosis:	3.755	**Cond. No.**	127.

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

유의수준 > 0.001

R-squared : 62.7%