### [2016 - 2017 삼성전자 주식의 수익률을 이용한 기술통계량 분석](https://drive.google.com/file/d/1k2KsY9DObv26gyANSKV6hBrCDydzQT1e/view?usp=sharing)

데이터 분석, 수치 해석, 통계, 시각화 모듈을 Import 한다.

import pandas as pd

import numpy as np

import scipy.stats as stats # 유용한 통계분석을 할 수 있는 모듈

import matplotlib.pyplot as plt

에프엔가이드 API에서 삼성전자 주가 정보를 불러온다. 거래소에서 삼성전자의 코드는 A005930이다. 주가정보날짜는 2016/01/01 - 2017/12/31로 하자.

# 에프엔가이드 삼성전자데이터를 dsf\_data라는 이름으로 저장

dsf\_data = pd.read\_json("http://app.fnguide.com/apiex/api/dsf/dsf001?gicode=A005930&fr\_dt=20160101&to\_dt=20171231")

# 데이터를 pandas dataframe으로 저장

df = pd.DataFrame(dsf\_data["Data"][0])

문자열로 인식되어있는 데이터의 형식을 변경한다. (DATE는 날짜, PRC는 숫자 형식 사용)

df["DATE"] = df["DATE"].apply(pd.to\_datetime, errors='coerce') # If ‘coerce’, then invalid parsing = NaN

df["PRC"] = df["PRC"].apply(pd.to\_numeric, errors='coerce')

데이터를 확인해보자.

print(df.head())

ASKHI BIDLO DATE HEXCD HSICCD OPENPRC PERMCO PRC \

0 1260000 1205000 2016-01-04 유가증권시장 I.013 1260000 A005930 1205000

1 1218000 1186000 2016-01-05 유가증권시장 I.013 1202000 A005930 1208000

2 1208000 1168000 2016-01-06 유가증권시장 I.013 1208000 A005930 1175000

3 1183000 1151000 2016-01-07 유가증권시장 I.013 1166000 A005930 1163000

4 1186000 1163000 2016-01-08 유가증권시장 I.013 1163000 A005930 1171000

RET RETX SHROUT VOL

0 -4.37 -4.37 147299337 306939

1 0.25 0.25 147299337 216002

2 -2.73 -2.73 147299337 366752

3 -1.02 -1.02 147299337 282388

4 0.69 0.69 147299337 257763

각 열별 정보는 다음과 같다.

* ASKHI, BIDLOW: 장종료시점 가장 높은 ask price 와 가장 낮은 bid price.
* DATE: 날짜
* HEXCD: 거래소정보
* HSICCD: 산업분류
* OPENPRC: 시초가 (opening price)
* PERMCO: 회사 ID
* PRC: 종가 (closing price)
* RET: 전날 종가대비 수익률
* RETX: 전날 종가대비 수정 수익률 (배당 등 반영)
* SHROUT: 주식 수
* VOL: 거래량

데이터프레임을 날짜를 행, 기업명을 열, 그리고 이에 대응하는 주가를 데이터로 하는 피벗 테이블을 만듭니다.

df\_pivot = df.pivot(index='DATE', columns='PERMCO', values='PRC')

print(df\_pivot.head())

PERMCO A005930

DATE

2016-01-04 1205000

2016-01-05 1208000

2016-01-06 1175000

2016-01-07 1163000

2016-01-08 1171000

피벗 테이블에서 매월 초에 해당하는 데이터를 뽑습니다.

def first\_day(entry):

return entry[0]

beginning\_price = df\_pivot.resample('M').apply(first\_day)

print(beginning\_price.head())

PERMCO A005930

DATE

2016-01-31 1205000

2016-02-29 1163000

2016-03-31 1197000

2016-04-30 1279000

2016-05-31 1250000

매월 초에 해당하는 데이터를 이용하여, 매월 수익률을 계산합니다.

diff = (beginning\_price - beginning\_price.shift(1)).shift(-1).dropna()

monthly\_return = diff.div(beginning\_price)

print(monthly\_return.head())

PERMCO A005930

DATE

2016-01-31 -0.034855

2016-02-29 0.029235

2016-03-31 0.068505

2016-04-30 -0.022674

2016-05-31 0.066400

매월 수익률을 로그를 이용하여 보정합니다.

log\_monthly\_return = np.log(monthly\_return + 1).dropna()

print(log\_monthly\_return.head())

PERMCO A005930

DATE

2016-01-31 -0.035477

2016-02-29 0.028816

2016-03-31 0.066260

2016-04-30 -0.022935

2016-05-31 0.064288

앞에서 구한 매월 로그 수익률을 그래프로 나타냅니다.

plt.figure(figsize=(12, 8))

plt.plot(log\_monthly\_return)

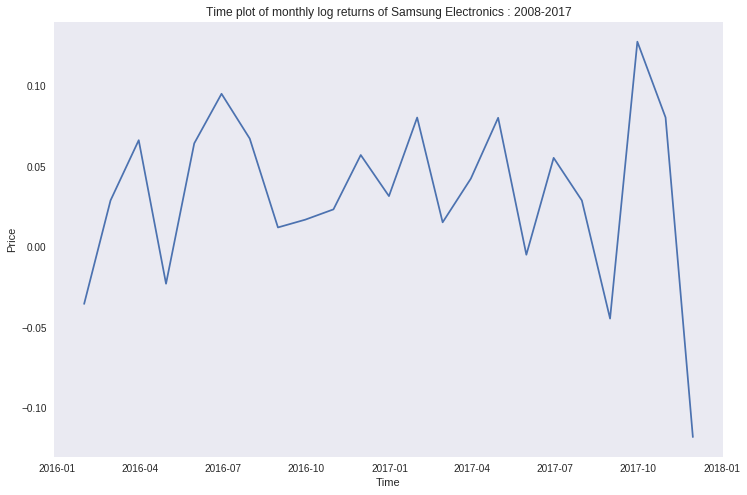
plt.grid()

plt.xlabel('Time')

plt.ylabel('Price')

plt.title('Time plot of monthly log returns of Samsung Electronics : 2008-2017')

plt.show()



데이터의 통계량을 표현합니다.

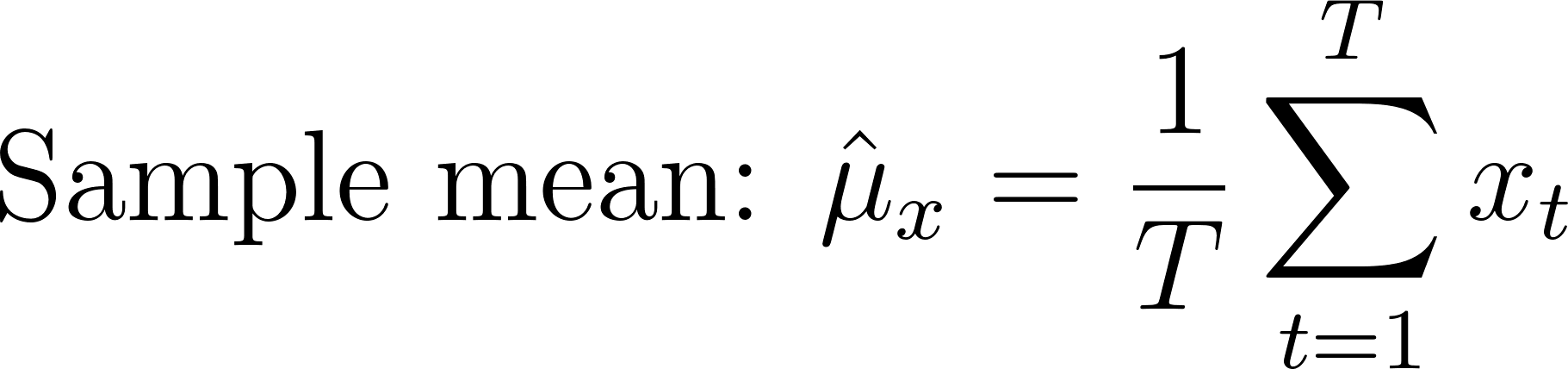
stats.describe(log\_monthly\_return)

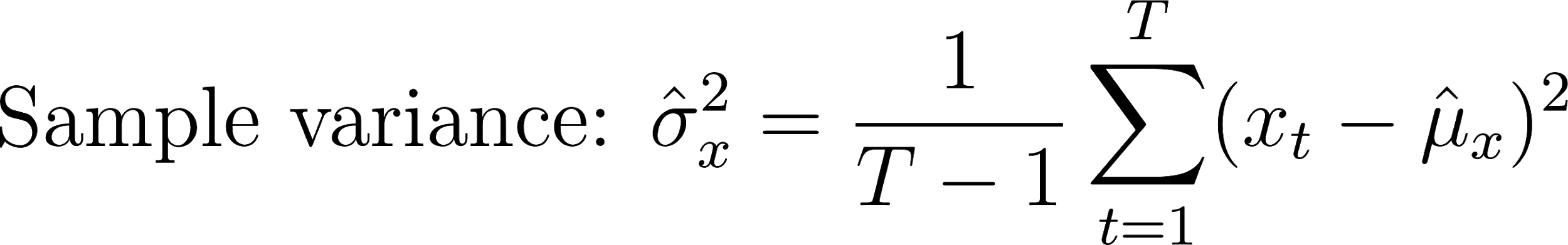
DescribeResult(nobs=23, minmax=(array([-0.11822004]), array([0.12748908])), mean=array([0.03245529]), variance=array([0.00287307]), skewness=array([-0.86265845]), kurtosis=array([1.03457877]))

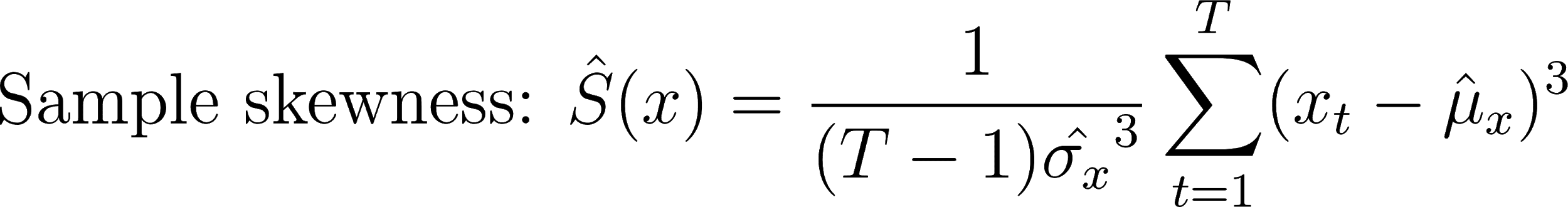
다음과 같은 데이터가 있다고 하자.

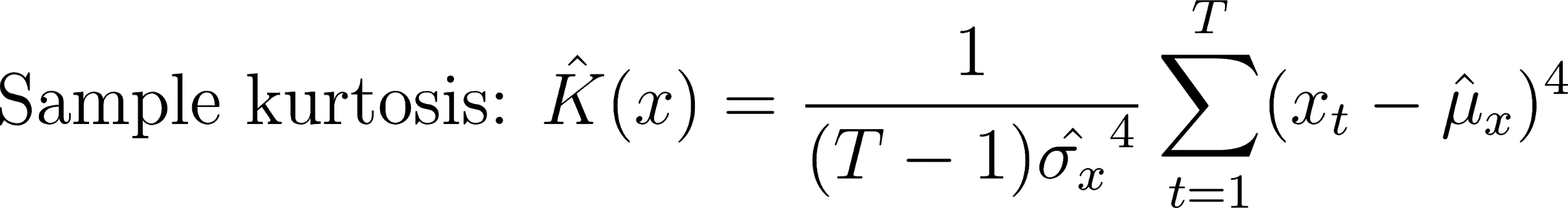
Data: {*x1*, *x2*,, …, *xT* }

이때 표본 평균(sample mean), 표본 분산 (sample variance), 표본 왜도 (sample skewness), 표본 첨도(sample kurtosis)는 다음과 같이 구할 수 있다.

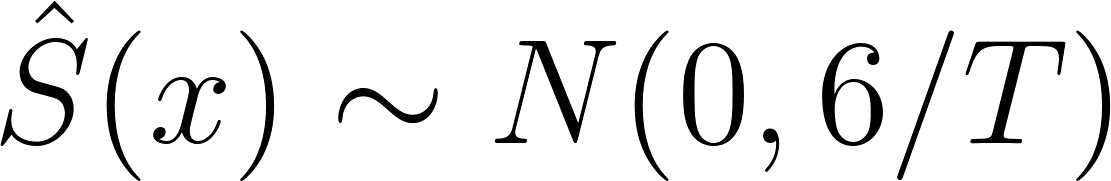
[](https://www.codecogs.com/eqnedit.php?latex=%20%5Ctext%7BSample%20mean%3A%20%20%7D%20%5Chat%7B%5Cmu%7D_x%20%3D%20%5Cfrac%7B1%7D%7BT%7D%20%5Csum_%7Bt%3D1%7D%5E%7BT%7D%20x_t%20%0)

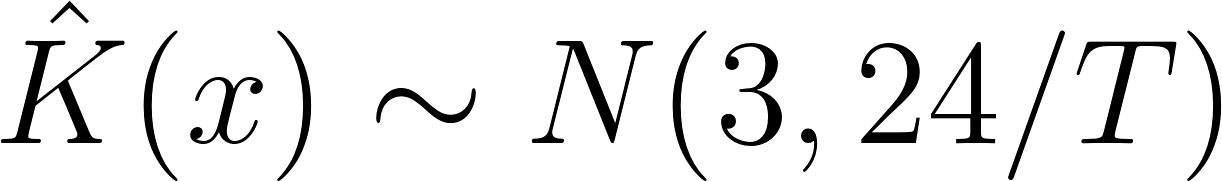
[](https://www.codecogs.com/eqnedit.php?latex=%20%5Ctext%7BSample%20variance%3A%20%20%7D%20%5Chat%7B%5Csigma%7D_x%5E2%20%3D%20%5Cfrac%7B1%7D%7BT-1%7D%20%5Csum_%7Bt%3D1%7D%5E%7BT%7D%20(x_t%20-%20%20%5Chat%7B%5Cmu%7D_x)%5E2%0)

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Ctext%7BSample%20skewness%3A%20%20%7D%20%5Chat%7BS%7D(x)%20%3D%20%5Cfrac%7B1%7D%7B(T-1)%20%5Chat%7B%5Csigma_x%7D%5E3%7D%20%5Csum_%7Bt%3D1%7D%5E%7BT%7D%20(x_t%20-%20%20%5Chat%7B%5Cmu%7D_x)%5E3%0)

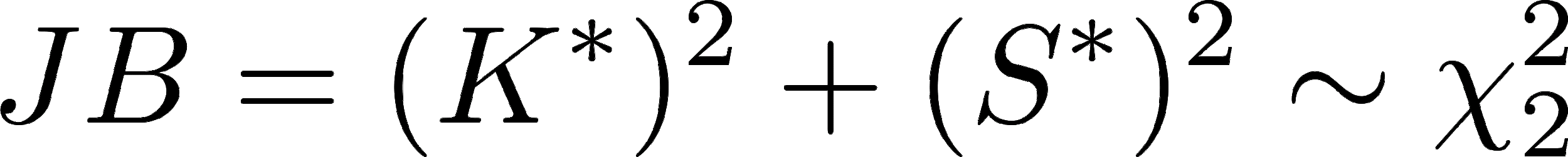
[](https://www.codecogs.com/eqnedit.php?latex=%20%5Ctext%7BSample%20kurtosis%3A%20%20%7D%20%5Chat%7BK%7D(x)%20%3D%20%5Cfrac%7B1%7D%7B(T-1)%20%5Chat%7B%5Csigma_x%7D%5E4%7D%20%5Csum_%7Bt%3D1%7D%5E%7BT%7D%20(x_t%20-%20%20%5Chat%7B%5Cmu%7D_x)%5E4%0)

데이터가 정규분포를 따른다면 표본왜도와 표본첨도의 분포는 다음과 같다.

[](https://www.codecogs.com/eqnedit.php?latex=%5Chat%7BS%7D(x)%20%5Csim%20N(0%2C%206%2FT)%0)

[](https://www.codecogs.com/eqnedit.php?latex=%5Chat%7BK%7D(x)%20%5Csim%20N(3%2C%2024%2FT)%0)

첨도와 왜도를 모두 사용하여 데이터가 정규분포를 따르는지 테스트할 수 있다. Jarque-Bera test가 많이 쓰인다.

[](http://www.texrendr.com/?eqn=JB%20%3D%20(K%5E*)%5E2%20%2B%20(S%5E*)%5E2%20%5Csim%20%5Cchi_2%5E2%20%0)

[](https://www.codecogs.com/eqnedit.php?latex=%5Cchi_2%5E2%20%5Ctext%7B%20denotes%20a%20chi%20squared%20distribution%20with%202%20degrees%20of%20freedom.%7D%0)

K\* 와 S\*는 왜도와 첨도에서 표본평균을 제하고 분산으로 나누어주어 정규화(normalized)한 값들이다. 단일 표본 t-검정을 수행한다. p-value 값이 0.05보다 큰 지 여부를 확인한다.

stats.ttest\_1samp(log\_monthly\_return, popmean=0)

Ttest\_1sampResult(statistic=array([2.90386154]), pvalue=array([0.0082321]))

정규성을 검정합니다. p-value 값이 0.05보다 큰 지 여부를 확인합니다.

# D’Agostino and Pearson’s test similar to JB test

stats.normaltest(log\_monthly\_return, axis=0)

NormaltestResult(statistic=array([6.07655854]), pvalue=array([0.04791727]))

Skewness test를 수행합니다.

T = len(log\_monthly\_return)

tst=stats.skew(log\_monthly\_return)/np.sqrt(6/T) # perform skenwness test

print(tst[0])

-1.688990372456996

양측 검정 통계량의 p-value 값을 계산합니다.

pv=2\*stats.norm.cdf(tst, 0, 1) # Compute two-sided p-value of the test statistic.

print(pv)

[0.09122127]

Kurtosis test를 수행합니다

tst2=(stats.kurtosis(log\_monthly\_return)-4.)/np.sqrt(24/T)

print(tst2[0])

1.0127957199142594