CUAI 데이터 분석 및 EDA 스터디 3조

2024.04.02.

발표자 : 황지민

스터디원 소개 및 만남 인증

김민하 (응통 22)

김부영 (융공 20)

김예은 (경영 20)

성산해 (수학 22)

황지민 (전전 20)

스터디 진행 방향

< 1주차 > 03/18 ~ 03/24

- 통계 분석 기법 학습 (심평원)
- **기상 데이터 분석 유튜브 영상 시청** (기상청 날씨마루)
- 기상 데이터 분석 파이썬 코드 해석 (기상청 날씨마루)

< 2주차 > 03/25 ~ 03/31

- **주가 데이터 분석 실습** (유튜브 영상 참고)

< 3주차 > 04/01 ~ 04/07

- 자전거 수요 예측 실습 (유튜브 영상 참고)

< 중간고사 이후 ~ 기말고사 이전 > - 캐글 / 데이콘 / 공공데이터 활용해 데이터 분석 진행

< 여름방학 > - 공모전 / 프로젝트 진행 예정

날씨마루 파이썬 교육 영상

- 파이썬
  - 가독성과 유지보수성
  - 스크립트 언어
  - 배터리 내장
  - 접착제 언어
  - 다양한 생태계

날씨마루 파이썬 교육 영상

- 넘파이
  - 행렬이나 일반적으로 대규모 다차원 배열을 쉽게 처리할 수 있도록 지원하는 파이썬의 라이브러리
  - 데이터 구조 외에도 수치 계산을 위해 효율적으로 구현된 기능을 제공
- 기본연산
  - 덧셈 : np.add()
  - 뺄셈 : np.subtract()
  - 곱셈 : np.multiply()
  - 나눗셈 : np.divide()
- 기술통계
  - 최솟값 : np.min()
  - 최댓값 : np.max()
  - 평균 : np.mean()
  - 중앙값 : np.median()
  - 분산 : np.var()
  - 표준편차 : np.std()

날씨마루 파이썬 교육 영상

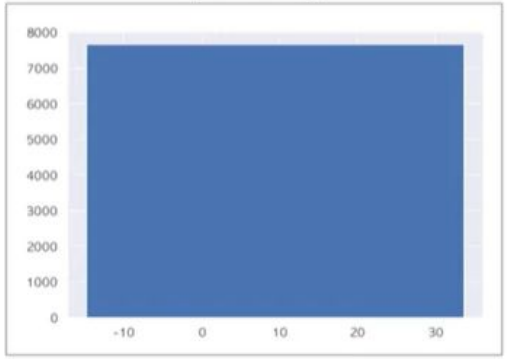
- 파일 불러오기
  - pd.read_csv("파일명")
- 수치형 데이터의 기술통계
  - count : 빈도수
  - mean : 평균
  - std : 표준편차
  - min : 최솟값
  - 25% : 1사분위수
  - 50% : 2사분위수(중앙값)
  - 75% : 3사분위수
  - max : 최댓값
- 범주형 데이터의 기술통계
  - count : 빈도수
  - unique : 중복제거 후의 빈도수
  - top : 최빈값
  - freq : 최빈값의 빈도수

날씨마루 파이썬 교육 영상

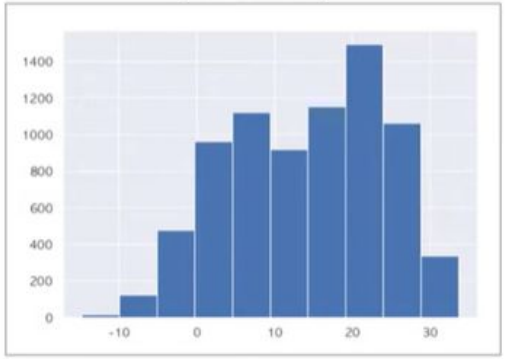
- 히스토그램(Histogram)
  - 표로 되어 있는 도수 분포를 정보 그림으로 나타낸 것
  - 도수분포표를 그래프로 나타낸 것

df[“파일”].hist(bins=10)

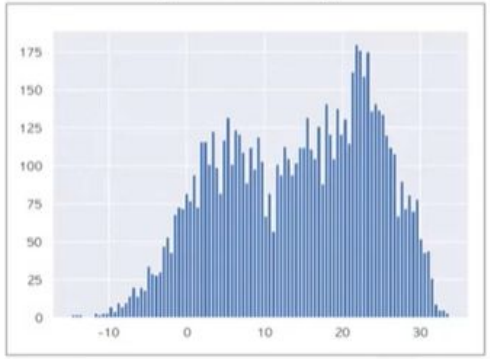
[bins=1]



[bins=10]



[bins=100]



날씨마루 파이썬 교육 영상

- 산점도
  - 두 변수 간의 관계의 방향성과 강도를 확인
  - 선형이나 비선형의 형태와 같은 수학적 모델을 확인
- 회귀분석
  - 관찰된 연속형 변수들에 대해 두 변수 사이의 모형을 구함
  - 두 변수 사이의 적합도를 측정해 내는 분석 방법

날씨마루 파이썬 교육 영상

- 범주형 변수의 빈도수 구하기
  - 하나의 범주형 변수 : series.value_counts()
  - 두 개의 범주형 변수 : pd.crosstab()
- 범주형 변수의 시각화
  - countplot : 범주형 변수의 빈도수를 시각화
  - barplot : 범주형과 수치형 변수를 함께 비교할 때 사용
  - boxplot : 요약수치(최댓값, 최솟값, 사분위수, 이상치)를 그래프로 표현
  - violinplot : 밀도함수 그래프를 마주보게 그림

날씨마루 파이썬 교육 영상

- 딥러닝
  - 인간의 뇌 신경 회로를 모방한 신경 회로망(neural network)을 다층적으로 구성
  - 컴퓨터가 다양한 데이터를 통해 마치 사람처럼 생각하고 배울 수 있도록 하는 기술
- 텐서플로
  - ML 모델을 개발하고 학습시키는 데 도움이 되는 핵심 오픈소스 라이브러리
  - 2015년에 오픈소스로 공개된 구글 브레인 팀의 두 번째 머신 러닝 시스템
- 회귀의 정확도 측정 방법
  - MAE(Mean Absolute Error) : 평균 절대 오차
  - MSE(Mean Square Error) : 평균 제곱 오차
  - RMSE(Root Mean Square Error) : 평균 제곱근 오차

주가 데이터 분석 - 데이터프레임 다루기

2023/03/22 ~ 2024/03/22 삼성전자 & SPC삼립 주가 데이터 이용 (KRX 한국거래소)

```
[ ]  df = pd.read_csv('/content/spc_주가데이터.csv', encoding = 'cp949')
     df
```

	일자	종가	대비	등락률	시가	고가	저가	거래량	거래대금	시가총액	상장주식수
0	2024/03/22	56500	-500	-0.88	57000	57000	56200	3844	217404300	487539008500	8629009
1	2024/03/21	57000	600	1.06	56500	57100	56400	4786	271750700	491853513000	8629009
2	2024/03/20	56400	200	0.36	56200	56700	56000	4811	270531900	486676107600	8629009
3	2024/03/19	56200	-300	-0.53	56600	56900	56200	3305	186495900	484950305800	8629009
4	2024/03/18	56500	300	0.53	56200	56900	55700	9785	551497700	487539008500	8629009
...	...	...	...	...	...	...	...	...	...	...	...
242	2023/03/28	68000	1200	1.80	66800	68000	66500	5384	363175200	586772612000	8629009
243	2023/03/27	66800	-1000	-1.47	67700	67700	66400	8513	568917800	576417801200	8629009
244	2023/03/24	67800	-200	-0.29	68000	68100	67100	7003	473485000	585046810200	8629009
245	2023/03/23	68000	-600	-0.87	68600	68600	67600	5777	393046800	586772612000	8629009
246	2023/03/22	68600	200	0.29	68600	69000	67800	4993	342379600	591950017400	8629009

247 rows × 11 columns

주가 데이터 분석 - 데이터프레임 다루기

1) 데이터프레임 칼럼명 변경

pd.rename(columns = {'기존 컬럼명' : '새로운 컬럼명'})

	일자	종가	대비	등락률	시가	고가	저가	거래량	거래대금	시가총액	상장주식수
0	2024/03/22	56500	-500	-0.88	57000	57000	56200	3844	217404300	487539008500	8629009
1	2024/03/21	57000	600	1.06	56500	57100	56400	4786	271750700	491853513000	8629009

	date	Closing Price	Change	Fluctuation Rate	Opening Price	High Price	Low Price	Trading Volume	Trading Value	Market Capitalization	Number of Listed Shares
0	2024/03/22	56500	-500	-0.88	57000	57000	56200	3844	217404300	487539008500	8629009
1	2024/03/21	57000	600	1.06	56500	57100	56400	4786	271750700	491853513000	8629009

2) 데이터 프레임 파악하기

- head(), tail()
- describe()
- sort_values()
- info()
- drop()

주가 데이터 분석 - 데이터프레임 시각화

시각화를 위한 **matplotlib** 라이브러리

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

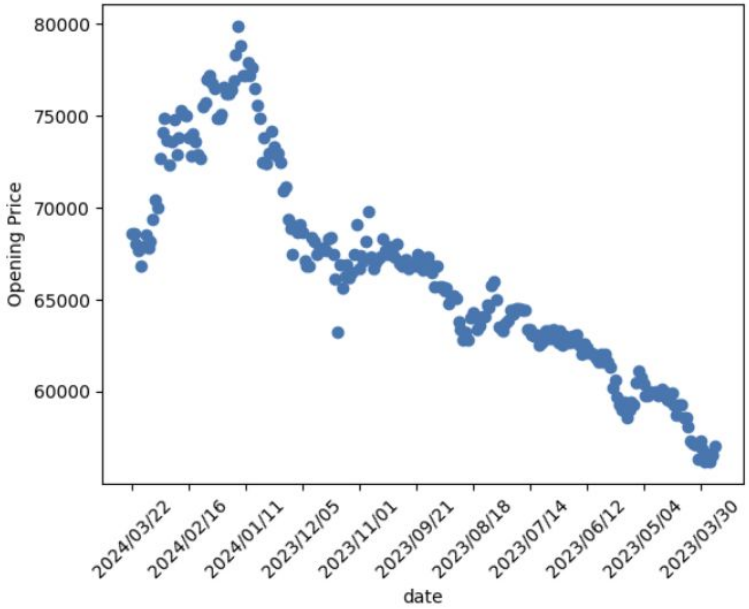
1) **산점도 (Scatter plot)**

: 두 변수 간 관계를 점으로 나타낸 그래프

변수 간 상관관계, 분포 확인

```
plt.scatter(df['date'], df['Opening Price'])
```

- plt.xlabel() / plt.ylabel() : 축 레이블 설정
- plt.xticks() / plt.yticks() : 축 눈금 설정
- plt.show() : 그래프를 화면에 표시



주가 데이터 분석 - 데이터프레임 시각화

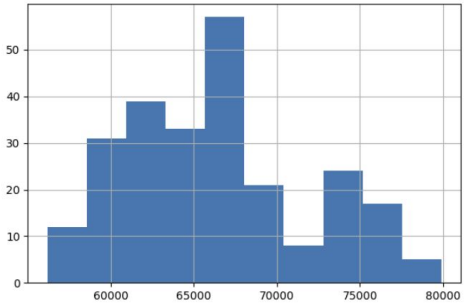
2) **히스토그램 (Histogram)**

: 데이터의 분포를 구간별로 나누어 막대로 나타낸 그래프

데이터의 분포 모양, 중심 경향성, 이상치 여부 파악

```
df['Opening Price'].hist(figsize=(6,4), bins=10)
```

- plt.tight_layout() : subplot들이 겹치지 않도록



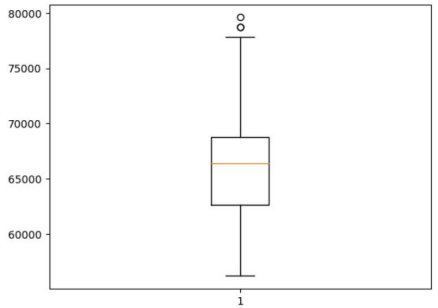
3) **Boxplot**

: 데이터의 중앙값과 사분위수를 상자 모양으로 나타낸 그래프

데이터의 분포 모양, 중심 경향성, 이상치 여부 파악

```
plt.boxplot(df['Closing Price'])
```

- pd.to_numeric() : 데이터를 숫자형식으로 변환



주가 데이터 분석 - 상관관계

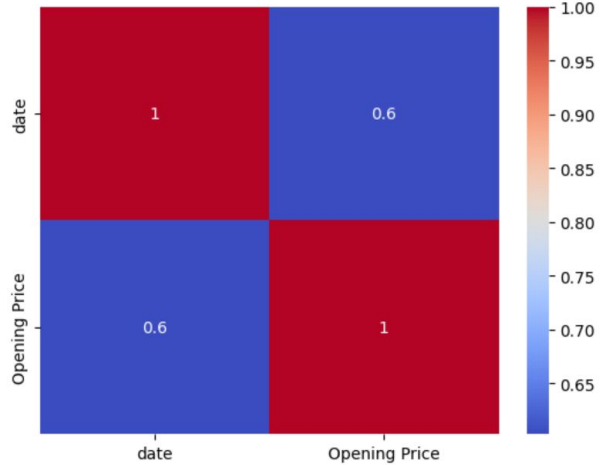
상관계수

- 두 변수 간의 상관관계를 나타내는 수치
- -1: 강한 음의 상관관계 / 0: 상관관계 없음 / 1: 강한 양의 상관관계

---

date와 Opening Price의 상관계수

corr(), sns.heatmap()



date와 모든 변수의 상관계수

corrwith()

date	1.000000
High Price	0.610900
Opening Price	0.602834
Market Capitalization	0.594824
Closing Price	0.594824
Low Price	0.594813
Trading Value	0.387475
Trading Volume	0.326069
Change	-0.037505
Fluctuation Rate	-0.037816
Number of Listed Shares	NaN

주가 데이터 분석 - FinanceDataReader

fdr.StockListing('KRX') -> KRX에 상장된 주식 목록 확인

	Code	ISU_CD	Name	Market	Dept	Close	ChangeCode	Changes	ChagesRatio	Open	High	Low	Volume	Amount	Marcap	Stocks	MarketId
0	005930	KR7005930003	삼성전자	KOSPI		78200	2	-700	-0.89	79600	79800	77800	18660392	1468796757203	466836995410000	5969782550	STK
1	000660	KR7000660001	SK하이닉스	KOSPI		169400	2	-400	-0.24	170500	174800	168500	3417222	582845196400	123323600631000	728002365	STK
2	373220	KR7373220003	LG에너지솔루션	KOSPI		414500	1	1000	0.24	416500	417000	412000	161091	66741857500	96993000000000	234000000	STK
3	207940	KR7207940008	삼성바이오로직스	KOSPI		840000	3	0	0.00	847000	852000	835000	47308	39839206000	59786160000000	71174000	STK
4	005935	KR7005931001	삼성전자우	KOSPI		65700	2	-1000	-1.50	66700	67200	65500	1398641	92530648800	54063656190000	822886700	STK

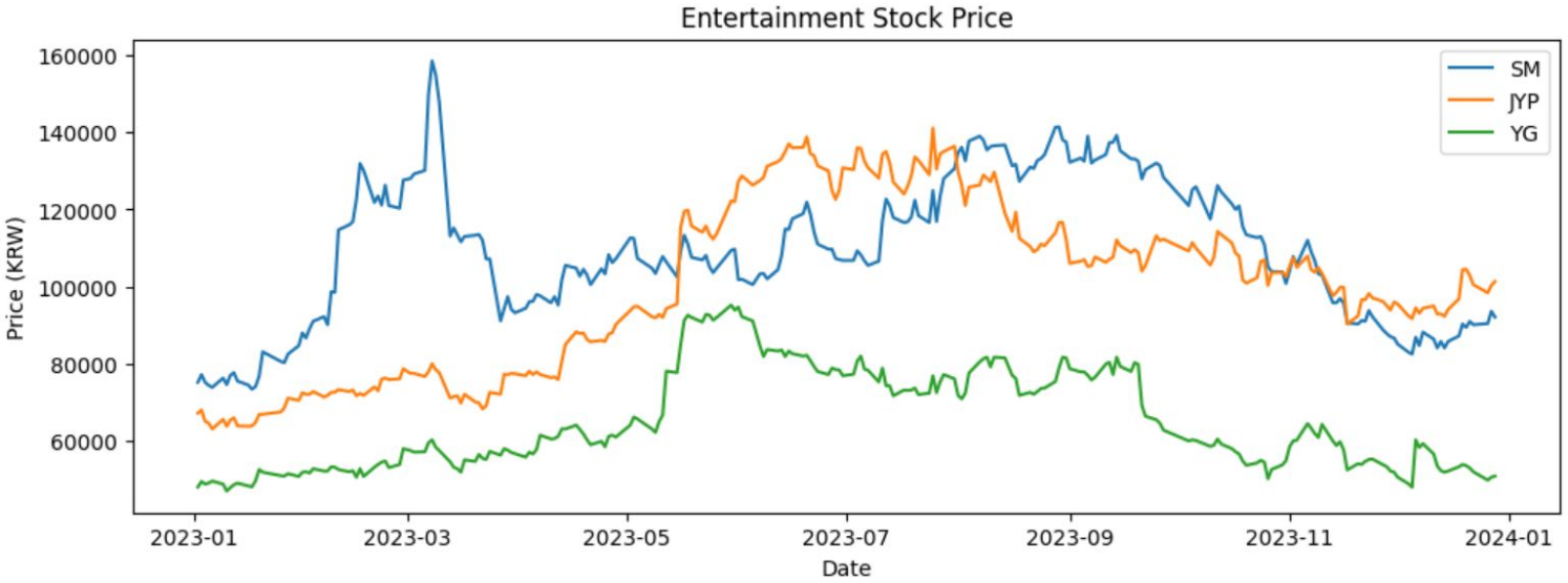
fdr.DataReader(종목코드, 시작일, 종료일)
-> 해당 종목의 주가 정보 확인

fdr.DataReader('041510', '2023-01-01', '2024-01-01')

Date	Open	High	Low	Close	Volume	Change
2023-01-02	77300	78000	73300	75200	425582	-0.019557
2023-01-03	74900	78400	74700	77200	462240	0.026596
2023-01-04	77200	77400	74200	75200	337977	-0.025907
2023-01-05	75600	76900	73600	74400	297806	-0.010638
2023-01-06	74000	74400	71700	73900	441861	-0.006720

주가 데이터 분석 - FinanceDataReader

fdr.DataReader()로 **여러 종목의 주가 정보**를 불러와서
시가, 종가 등의 변동을 **그래프로 시각화 및 비교** 가능



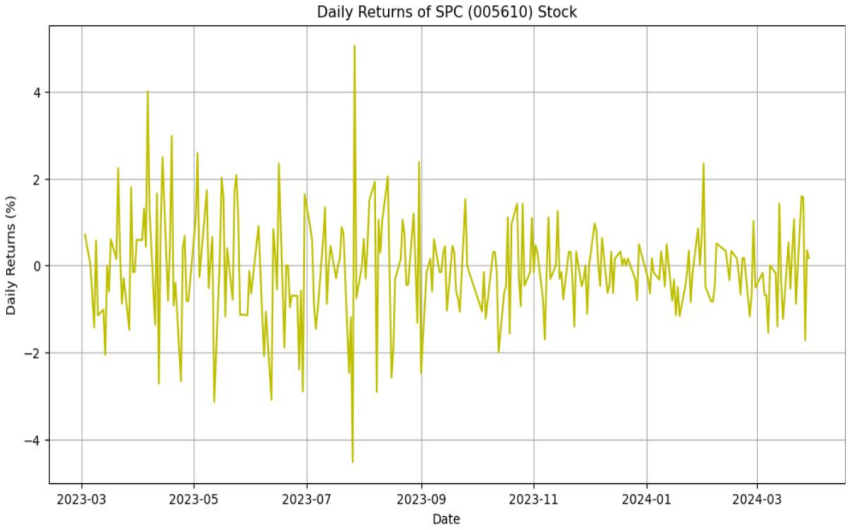
주가 데이터 분석 - 일별 수익률 계산 및 시각화

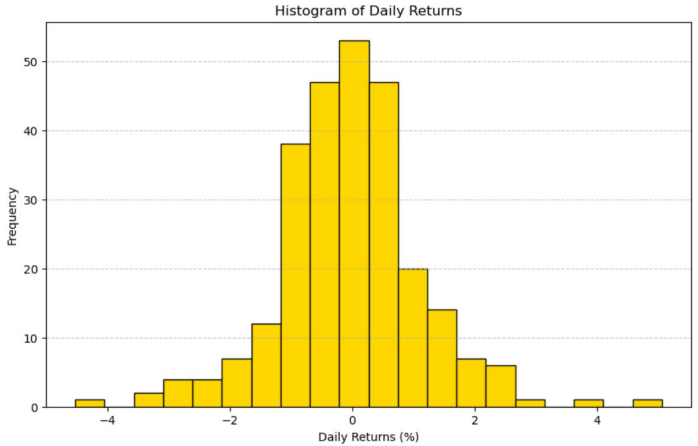
일별 수익률(%) = (현재 가격 - 어제 가격) / 어제 가격 x 100

df = (df['Close'].**pct_change()**) * 100

```
3  df_spc.describe()

count    265.000000
mean      -0.068362
std        1.168431
min       -4.524887
25%       -0.698324
50%       -0.147059
75%        0.504202
max        5.055292
Name: Close, dtype: float64
```





주가 데이터 분석 - 주가등락 구하기

주가가 오를 확률?
(평균 주가 변동)

$$P(X=k)=\frac{n!}{k!\times(n-k)!}\times p^k\times(1-p)^{n-k}$$

이항분포를 이용

- n은 시행횟수
- k는 성공(==주가상승)횟수
- p는 각 시행에서 성공할 확률

1년간 SPC삼립 일별 수익률

```
df1_spc=df_spc.loc['2023-03-01':'2024-03-01']
df2_spc=df1_spc.copy()
df2_spc['ret']=(df2_spc['Close'].pct_change())*100
df3_spc=df2_spc['ret']
r=df3_spc.dropna()
len(r)
```

```
245
```

성공 확률 p

```
positive=0
n = len(r)
for i in range(n):
    if r[i]>0:
        positive +=1

positive_rate=positive/n
positive_rate
```

```
0.4204081632653061
```

ex) 8일 중 5일이 오를 확률?

```
import math
a=math.factorial(8)
b=math.factorial(5)*math.factorial(3)
c=((positive_rate)**5)*((1-positive_rate)**3)

number=a/b
prob=number*c
prob
```

```
0.14318928113438226
```

감 사 합 니 다