Pandas

```
! pip install pandas
```

숨겨진 출력 표시

```
import pandas as pd
```

1. pandas 직접 만들기

```
# numpy로 부터 만들기
import numpy as np
```

```
arr = np.arange(9)
arr
```

숨겨진 출력 표시

```
df = pd.DataFrame(arr)
df
```

숨겨진 출력 표시

```
type(df)
```

숨겨진 출력 표시

```
arr = np.arange(9).reshape(3,3)
arr
```

숨겨진 출력 표시

```
df = pd.DataFrame(arr)
df
```

숨겨진 출력 표시

```
df = pd.DataFrame(arr,
                  columns = list('abc'))
df
```

숨겨진 출력 표시

```
df = pd.DataFrame(arr,
                  columns = list('abc'),
                  index = [10,11,12])


# list로 부터 만들기
df = pd.DataFrame([[1,2,3], [4,5,6], [7,8,9]])
df
```

숨겨진 출력 표시

```
# dictionary로 부터 만들기
df = pd.DataFrame({'a': [1,2,3],
                   'b': [4,5,6],
                   'c': [7,8,9]})
df
```

숨겨진 출력 표시

```
! ls
```

숨겨진 출력 표시

2. 외부 데이터(따릉이)로 pandas 만들기

```
! ls
```

숨겨진 출력 표시

```
df = pd.read_excel('18_공공자전거_파생변수.xlsx')
df
```

숨겨진 출력 표시

```
df.head()
```

숨겨진 출력 표시

```
df.shape
```

숨겨진 출력 표시

```
df.columns
```

숨겨진 출력 표시

```
df.index
```

숨겨진 출력 표시

```
df.info()
```

숨겨진 출력 표시

```
df.describe()
```

숨겨진 출력 표시

```
df.median()
```

숨겨진 출력 표시

```
df['이용시간'].describe()
```

숨겨진 출력 표시

```
df['이용시간'].var()
```

숨겨진 출력 표시

```
df.corr()
```

숨겨진 출력 표시

```
df.cov()
```

숨겨진 출력 표시

```
df.hist()
```

숨겨진 출력 표시

```
import warnings
warnings.filterwarnings(action='ignore')
```

```
df.hist()
```

숨겨진 출력 표시

```
df['이용시간'].hist()
```

숨겨진 출력 표시

```
np.log(df['이용시간']).hist()
```

숨겨진 출력 표시

```
df['이용시간'].min()
```

숨겨진 출력 표시

```
np.log(1)
```

숨겨진 출력 표시

```
df['이용시간'].max()
```

숨겨진 출력 표시

```
np.log(162)
```

숨겨진 출력 표시

```
np.log(df['이용시간'].mean())
```

숨겨진 출력 표시

```
df['이용시간'].plot()
```

숨겨진 출력 표시

```
import matplotlib.pyplot as plt
plt.scatter(x= df.index, y= df['이용시간'])
```

숨겨진 출력 표시

```
plt.scatter(df['이용시간'], df['이용거리'])
```

숨겨진 출력 표시

```
# 산점도를 보고 대여일시(요일, 시간, 오전 오후 등 시간대)별로 다른가?
df['대여일시']
```

숨겨진 출력 표시

```
# 대여일자를 날짜 서식으로 바꾸기
pd.to_datetime(df['대여일시'])
```

숨겨진 출력 표시

```
df['대여일시'] = pd.to_datetime(df['대여일시'])
```

```
df['대여일시'].dt.year
```

숨겨진 출력 표시

```
df['대여일시'].dt.month
```

숨겨진 출력 표시

```
df['대여일시'].dt.weekday
```

숨겨진 출력 표시

```
df['대여일시'].dt.hour
```

숨겨진 출력 표시

```
df['요일'] = df['대여일시'].dt.weekday
df['시간'] = df['대여일시'].dt.hour
```

```
df
```

숨겨진 출력 표시

```
df['시간'].unique()
```

숨겨진 출력 표시

```
import numpy as np
np.sort(df['시간'].unique())
```

숨겨진 출력 표시

```
df['시간'].apply(lambda x :'오후' if  x < 18 else '오전')
```

숨겨진 출력 표시

```
df['시간'].apply(lambda x :'오후' if  12 < x < 20 else('밤' if 20 <= x < 24 else '야밤'))
```

숨겨진 출력 표시

```
df['시간대'] = df['시간'].apply(lambda x :'오후' if  12 < x < 20 else('밤' if 20 <= x < 24 else '야
```

```
df
```

숨겨진 출력 표시

```
df.info()
```

숨겨진 출력 표시

```
df.head()
```

숨겨진 출력 표시

```
# 요일 수자를 text로 mapping하기

day_dict = {0: '월', 1: '화', 2: '수', 3: '목', 4: '금', 5: '토', 6: '일'}
```

```
df['요일']
```

숨겨진 출력 표시

```
df['요일'].map(day_dict)
```

숨겨진 출력 표시

```
df['요일'] = df['요일'].map(day_dict)
```

```
df.head()
```

숨겨진 출력 표시

```
# 내용 확인
df.tail()
```

숨겨진 출력 표시

```
# 크기 확인
df.shape
```

숨겨진 출력 표시

```
df.columns
```

숨겨진 출력 표시

```
for col in df.columns:
    print(col, type(col))
```

숨겨진 출력 표시

```
df.index
```

숨겨진 출력 표시

```
df.values
```

숨겨진 출력 표시

```
type(df.values)
```

숨겨진 출력 표시

```
df.values.shape
```

숨겨진 출력 표시

```
len(df)
```

숨겨진 출력 표시

```
len(df.columns)
```

숨겨진 출력 표시

```
len(df.index)
```

숨겨진 출력 표시

```
# 기본 속성 확인
df.describe()                # Series or DataFrame을 return
#df.describe?                  # df.describe(include='all')
```

숨겨진 출력 표시

```
df.info()
```

숨겨진 출력 표시

```
df.dtypes
```

숨겨진 출력 표시

```
plt.boxplot(df['이용거리'])
```

숨겨진 출력 표시

```
plt.boxplot(df[['이용거리', '이용거리']])
```

숨겨진 출력 표시

```
df.nunique()
```

숨겨진 출력 표시

```
df.nunique().plot(kind='bar')
```

숨겨진 출력 표시

```
df.count()
```

숨겨진 출력 표시

```
# 고유한 개수
df['요일'].unique()
```

숨겨진 출력 표시

```
# null 개수
df.isnull()
```

숨겨진 출력 표시

```
df.isnull().sum()  # df.isnull().sum(1)
```

숨겨진 출력 표시

```
  df.notnull().sum()
```

숨겨진 출력 표시

```
  # 속도계산
  df['분속'] = df['이용거리']/df['이용시간']
  df['분속']
```

숨겨진 출력 표시

```
df.sample(5)
```

숨겨진 출력 표시

```
# 실수를 정수로
df['분속'] = df['분속'].astype(int)
df['이용거리'] = df['이용거리'].astype(int)
```

```
df.sample(3)
```

숨겨진 출력 표시

```
# Text 처리
df['대여 대여소명'].str.contains('역')
```

숨겨진 출력 표시

```
df['역'] = df['대여 대여소명'].str.contains('역')
```

```
df['학교'] = df['대여 대여소명'].str.contains('학교')
```

```
df.head()
```

숨겨진 출력 표시

```
# 역 or 학교
df['역'] | df['학교']
```

숨겨진 출력 표시

```
df['역+공원'] = (df['역'] | df['학교'])
df.head()
```

숨겨진 출력 표시

```
# 대여와 반납 대요소가 같은가?
df[df['대여 대여소명'] == df['반납대여소명']]
```

숨겨진 출력 표시

```
df[df['대여 대여소번호'] == df['반납대여소번호']]
```

숨겨진 출력 표시

```
  df['대여=반납'] = (df['대여 대여소명'] == df['반납대여소명'])
  df
```

숨겨진 출력 표시

```
del df['반납대여소명']
```

```
df.head()
```

숨겨진 출력 표시

```
del df['대여 대여소명']
```

```
df.head()
```

숨겨진 출력 표시

```
# 짝수 행
df[::2].head()
```

숨겨진 출력 표시

```
# 홀수 행
df[1::2].head()
```

숨겨진 출력 표시

데이터 현황 분석

EDA(Explanatory Data Analysis)

```
# aggregation 함수로 기본 정보 확인
df['이용거리'].sum()
```

숨겨진 출력 표시

```
df['이용거리'].count()
```

숨겨진 출력 표시

```
df['이용거리'].median()
```

숨겨진 출력 표시

```
df['이용거리'].mode()
```

숨겨진 출력 표시

```
df['이용거리'].quantile([0.25, 0.5, 0.75])    # 50% 백분위수의 값이 중위값(median)
```

숨겨진 출력 표시

```
iqr = df['이용거리'].quantile(0.75) - df['이용거리'].quantile(0.25)
ceiling = df['이용거리'].quantile(0.75) + 1.5*iqr
ceiling
```

숨겨진 출력 표시

```
df['이용거리'] > ceiling
```

숨겨진 출력 표시

```
df[df['이용거리'] > ceiling]
```

숨겨진 출력 표시

```
df[df['이용거리'] > ceiling]['이용거리']
```

숨겨진 출력 표시

```
df['이용거리'].min()
```

숨겨진 출력 표시

```
df['이용거리'].max()
```

숨겨진 출력 표시

```
df['이용거리'].mean()
```

숨겨진 출력 표시

```
df['이용거리'].var()
```

숨겨진 출력 표시

```
df['이용거리'].std()
```

숨겨진 출력 표시

```
# 수치형 특정 컬럼 내용 살펴보기
df['이용거리'].value_counts()
```

숨겨진 출력 표시

```
df
```

숨겨진 출력 표시

상세분석

요일마다 이용시간과 거리가 다른가를 분석

```
  df['요일'] == '일'
```

숨겨진 출력 표시

```
  df[df['요일'] == '일']
```

숨겨진 출력 표시

```
df['요일'].value_counts()
```

숨겨진 출력 표시

```
df['요일'].value_counts().plot(kind='bar')
```

숨겨진 출력 표시

```
df.groupby(['요일'])
```

숨겨진 출력 표시

```
for i, j in df.groupby(['요일']):
    i
    j.head()
```

```
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

```
for i, j in df.groupby(['요일']):
    i
    j.head(3)
```

숨겨진 출력 표시

```
for i, j in df.groupby(['요일', '시간대']):
    i
    j.head(3)
```

숨겨진 출력 표시

```
df.groupby(['요일', '시간대'])['이용시간'].mean()
```

숨겨진 출력 표시

```
df.groupby(['요일', '시간대'])['이용시간'].mean().unstack() # .round(2)
```

숨겨진 출력 표시

```
df.groupby(['요일', '시간대'])['이용시간'].mean().unstack().astype(int)
```

숨겨진 출력 표시

```
# 요일별로 크기를 색상으로 (행 기준)
df.groupby(['요일', '시간대'])['이용시간'].mean().unstack().astype(int).style.background_gradient
```

숨겨진 출력 표시

```
# 시간대대별로 크기를 색상으로 (열 기준)
df.groupby(['요일', '시간대'])['이용시간'].mean().unstack().astype(int).style.background_gradient
```

숨겨진 출력 표시

```
df.head()
```

숨겨진 출력 표시

```
cross = df.pivot_table(index=['요일'], columns=['시간대'], values=['역'], aggfunc=['sum', 'mean',
cross
```

숨겨진 출력 표시

```
cross.style.background_gradient()
```

숨겨진 출력 표시

이동평균

Moving average

'대여일시별'로 시계열로(날짜 순서대로)

```
df.sort_values('대여일시')
```

숨겨진 출력 표시

```
df = df.sort_values('대여일시')
```

대여일시를 인덱스로 하면 시계열 분석이 아주 쉬워짐

```
df.set_index(['대여일시'])
```

숨겨진 출력 표시

```
tf = df.set_index(['대여일시'])
tf
```

숨겨진 출력 표시

```
tf.resample('D')
```

숨겨진 출력 표시

```
for i, j in tf.resample('W'):
    i
    j.head(2)
```

숨겨진 출력 표시

```
tf.resample('M')['이용시간'].mean()
```

숨겨진 출력 표시

```
tf.resample('W')['이용시간'].mean()
```

숨겨진 출력 표시

이동평균(Moving average)

```
tf['이용시간'].rolling(window=5).mean()
```

숨겨진 출력 표시

```
tf['이용시간'].rolling(window=7).mean().plot(figsize=(15,3))
```

숨겨진 출력 표시

```
tf.resample('D')['이용시간'].mean().plot(figsize=(15,3))
```

숨겨진 출력 표시

End