EDA와 시각화

데이터의 특징과 데이터에 내제된 관계를 알아내기 위해 그래프와 통계적 분석 방법을 활용하여 데이터를 탐구하는 것

EDA의 주제

1. 저항성의 강조 : 이상치 등 부분적 변동에 대한 민감성 확인
2. 잔차계산 : 관찰 값들이 주 경향에서 벗어난 정도 파악
3. 자료변수의 재표현 : 변수를 적당한 척도로 바꾸는 것
4. 그래프롤 통한 현시성 : 분석 결과를 이해하기 쉽게 시각화하는 것

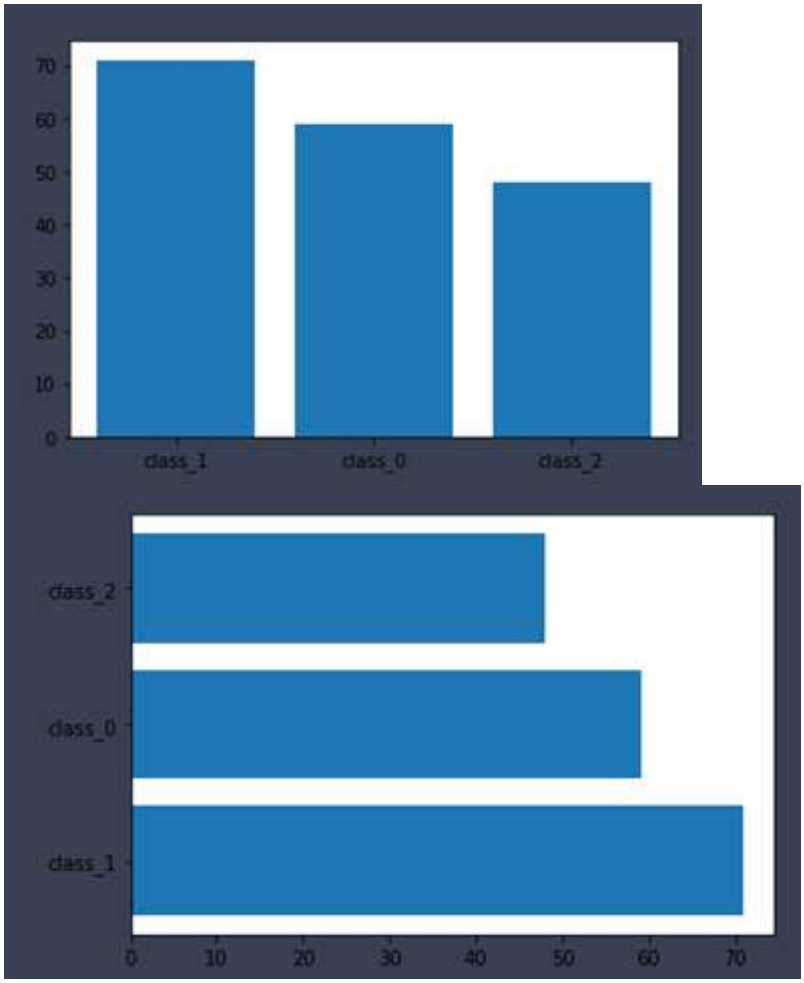
막대그래프

범주형 데이터를 요약하고 시각적으로 비교하는데 효과적인 그래프
막대그래프로 각 범주의 값의 개수 차이를 비교하고, 값 차이가 많이 날 경우 업/다운 샘플링을 통해 개수가 유사하도록 조정할 수 있음.

plt.bar(x,height,weidh=0.8,bottom=None,align='center',data=None)

x	막대의 x 좌표
height	막대의 높이
width	막대의 너비
bottom	막대 바닥변의 y 좌표(누적 막대그래프를 그릴 때 사용할 수 있음)
align	x 좌표에 대한 막대 정렬 {'center','edge'}

```
#범주형 변수인 wine['Class']로 막대그래프 그림
wine['Class'] = wine_load.target
wine['Class'] = wine['Class'].map({0:'class_0',1:'class_1',2:'class_2'})

#갯수 확인
wine_type = wine['Class'].value_counts()
wine_type

#막대그래프 그리기
#'Rectangle' object has no property 'aglign'
plt.bar(wine_type.index, wine_type.values, width = 0.8,
          bottom = None)
plt.show()

#수평 막대그래프
plt.barh(wine_type.index, wine_type.values, height = 0.8, left = None )
plt.show()
```



히스토그램

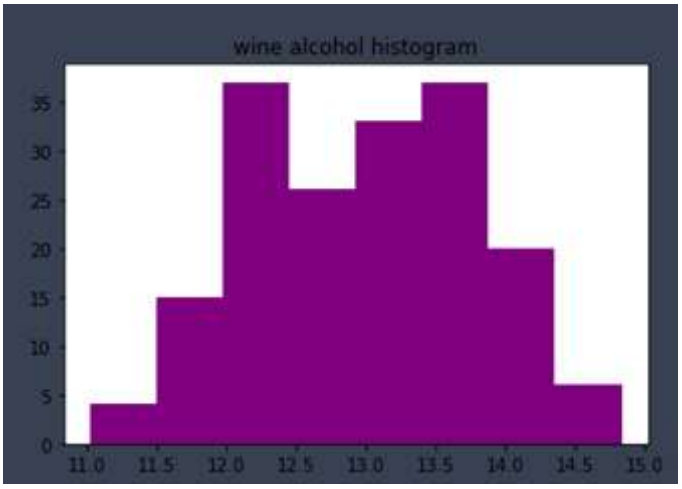
연속형 자료에 대한 도수분포표를 시각화하여 나타낸 것으로 서로 겹치지 않는 특정 구간에 따른 데이터의 빈도수를 표현

각구간은 연속되어 있으므로 히스토그램의 막대는 서로 붙어있으며, 각 구간의 순서는 임의로 변경하여 나타낼 수 없다.

plt.hist('변수명', bins = None, range=None, density = Fasle, data=df)

bin	히스토그램의 구간의 개수 정의
range	bin의 상한값과 하한값(x.min(), x.max()) 형태로 선언
density	True이면 확률밀도함수를 그리고 반환

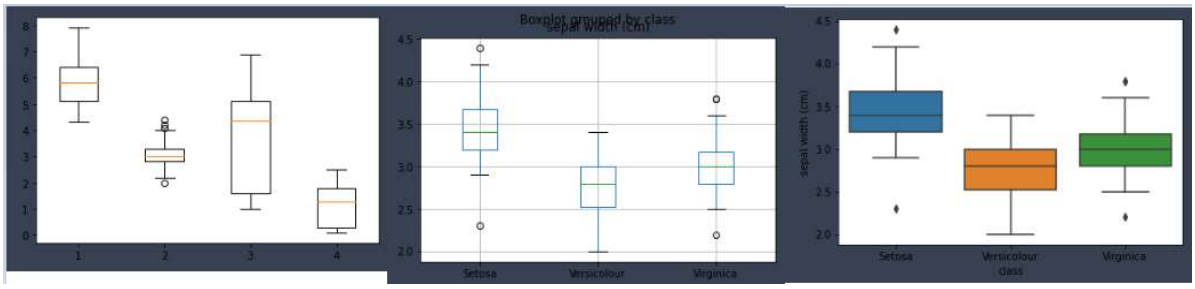
```
plt.title('wine alcohol histogram')
plt.hist('alcohol', bins = 8, range =
(wine['alcohol'].min(),wine['alcohol'].max()), color = 'purple', data = wine)
plt.show()
```



Boxplot

사분위수를 이용하여 수치형 변수값의 분포를 확인하는 그래프

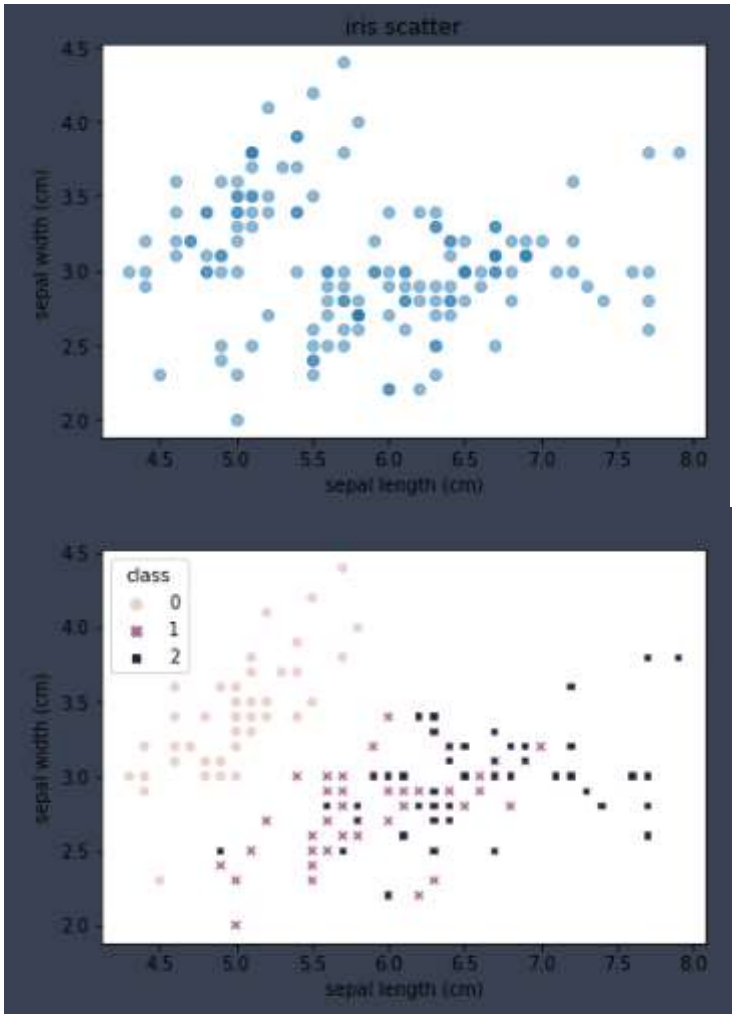
```
import seaborn as sns
import pandas as pd
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt
loaded_iris = load_iris() # print(type(iris)) : <class 'sklearn.utils.Bunch'>
iris = pd.DataFrame(loaded_iris.data, columns = loaded_iris.feature_names)
#iris.target은 sklearn.utils.Bunch.target 함수로 분류 결과를 나타냄. pandas Series로 리턴
iris['class'] = loaded_iris.target
iris['class'] = iris['class'].map({0:'Setosa',1:'Versicolour',2:'Virginica'})

#pyplot으로 그린 boxplot
#class는 문자열 변수로 치환하였으므로 박스플롯으로 표현 불가하여 제거하고 나머지 변수를 그림
plt.boxplot(iris.drop(columns='class'))
plt.show()
#class 분류에 따른 sepal width를 boxplot으로 표현
iris[['sepal width (cm)','class']].boxplot(by='class')
plt.show()
#sns로 그린 boxplot
sns.boxplot(x="class",y="sepal width (cm)", data =iris)
plt.show()
```



산점도(Scatter Plot)

두 개의 *수치형 변수* 각각의 분포와 함께 *두 변수의 관계*를 확인하는 가장 기본적인 그래프
관계의 유형 판단 : 점들이 흩어져 있는 모양보고 판단
관계의 강도 판단 : 적합선에 멀리 퍼져있으면 약한 상관관계, 가까이 퍼져 있으면 강한 상관관계

```
#Scatter Plot
import pandas as pd
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt

loaded_iris = load_iris()
iris = pd.DataFrame(loaded_iris.data, columns = loaded_iris.feature_names)
iris['class'] = loaded_iris.target
iris['class'].map({0:'Setosa', 1:'Versicolour', 2:'Verginica'})

#pyplot 산점도
plt.title('iris scatter')
plt.xlabel('sepal length (cm)')
plt.ylabel('sepal width (cm)')
plt.scatter(x=iris['sepal length (cm)'], y = iris['sepal width (cm)'], alpha =
0.5)
plt.show()

#seaborn 산점도
import seaborn as sns
sns.scatterplot(x='sepal length (cm)', y = 'sepal width (cm)'
                , data=iris, hue='class',style='class')
```



수평선 그래프

한계점, 평균값 그리는데 사용

plt.hlines(y,xmin,xmax,colors=None,linestyles='solid')

수직선 그래프

plt.vlines(x,ymin,ymax,colors=None,linestyles='solid')

함수식 그래프

plt.plot(x축, 함수식, data=df, c='color')

최소제곱 다항식 (회귀선)

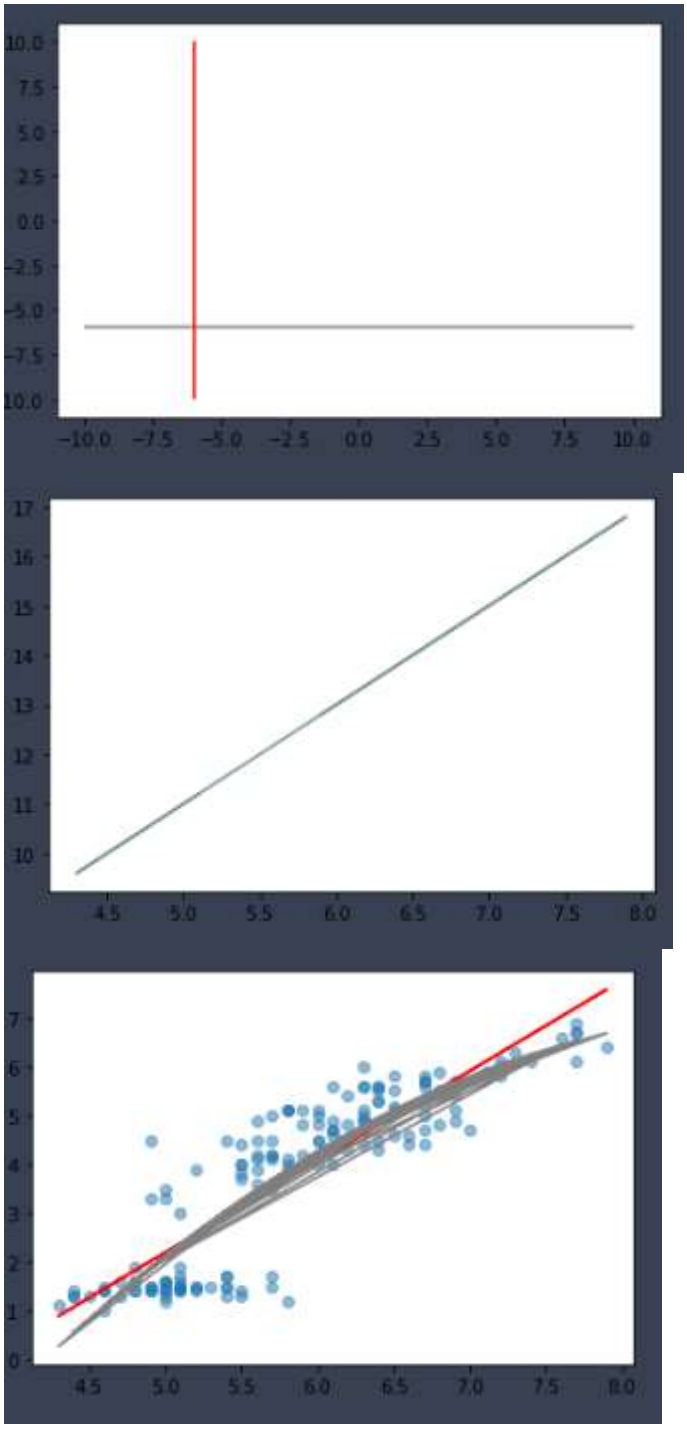
numpy.polyfit(X,Y,차수)

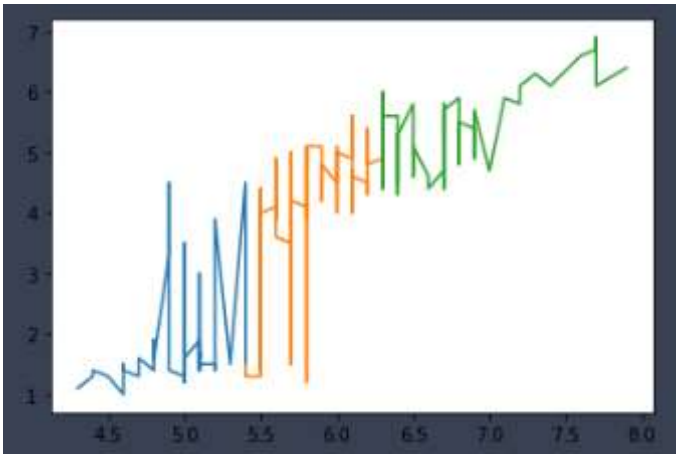
꺾은선 그래프

시간의 변화에 따라 값이 지속적으로 변화할 때 유용한 그래프
X축이 시점, Y축이 값을 의미
시점에 대한 변화를 보여주는 그래프 이므로 X축 값에 대한 정렬이 필요.

```
#수평선 수직선 그래프
plt.hlines(-6,-10,10,color='grey')
plt.vlines(-6,-10,10,color='red')

#함수식 그래프
#2x+1의 그래프를 그림
def linear_func(x) :
    return 2*x + 1

X = iris['sepal length (cm)']
plt.plot(X,linear_func(X),c='#789395')
plt.show()

#회귀선 그래프(numpy)
import numpy as np
X,Y = iris['sepal length (cm)'], iris['petal length (cm)']
#1차원 최소제곱 다항식 생성
b1, b0 = np.polyfit(X,Y,1)
plt.scatter(x=X, y=Y,alpha = 0.5)
#1차선 그래프
plt.plot(X,b1*X+b0, color='red')

#2차선 그래프
#차원 최소제곱 다항식 생성, 결과값은 차수개수 +1개가 출력, 높은 차수의 계수부터 출력
#차수개수가 2개이니 3개의 결과값이 생성되고, x**2의 계수부터 출력
c2, c1, c0 = np.polyfit(X,Y,2)
plt.plot(X,c0+c1*X+c2*X**2, color='grey' )
plt.show()

#꺾은선 그래프
iris2 = pd.DataFrame(load_iris().data, columns =
load_iris().feature_names).sort_values(by = 'sepal length (cm)')
iris2['class'] = load_iris().target
```

```
#카테고리에 따라 분리된 그래프 그리기
plt.plot('sepal length (cm)','petal length (cm)'
         , data=iris2.loc[iris2['class']==0])
plt.plot('sepal length (cm)','petal length (cm)'
         , data=iris2.loc[iris2['class']==1])
plt.plot('sepal length (cm)','petal length (cm)'
         , data=iris2.loc[iris2['class']==2])
plt.show()
```





산점도 행렬

두 개 이상의 변수가 있는 데이터에서 변수들 간의 산점도를 그린 그래프

산점도 행렬 해석방법

1. 대각선의 히스토그램을 통해 이상치를 확인한다.
2. 종속변수와 설명변수들 간의 관계를 시각적으로 판단한다.
3. 종속변수가 수치형인 경우 각 설명변수와의 직선 상관관계를 비교한다.
4. 종속변수가 범주형인 경우 종속변수를 잘 구분하는 변수를 파악한다.
5. 설명변수 간의 직선 함수관계를 파악하여 다중공선성 문제를 진단한다.

KDE 그래프
히스토그램과 함께 Non-parametric 밀도 추정 방법 중 하나
bin의 크기와 시작 및 종료 위치에 따라서 그래프가 달라지는 히스토그램의 문제점을 개선한 방법, 커널 함수를 사용하여 데이터의 분포를 smooth하게 나타낸 것이다.

scatter_matrix(data, alpha=0.5,figsize=(8.8), diagonal='hist')

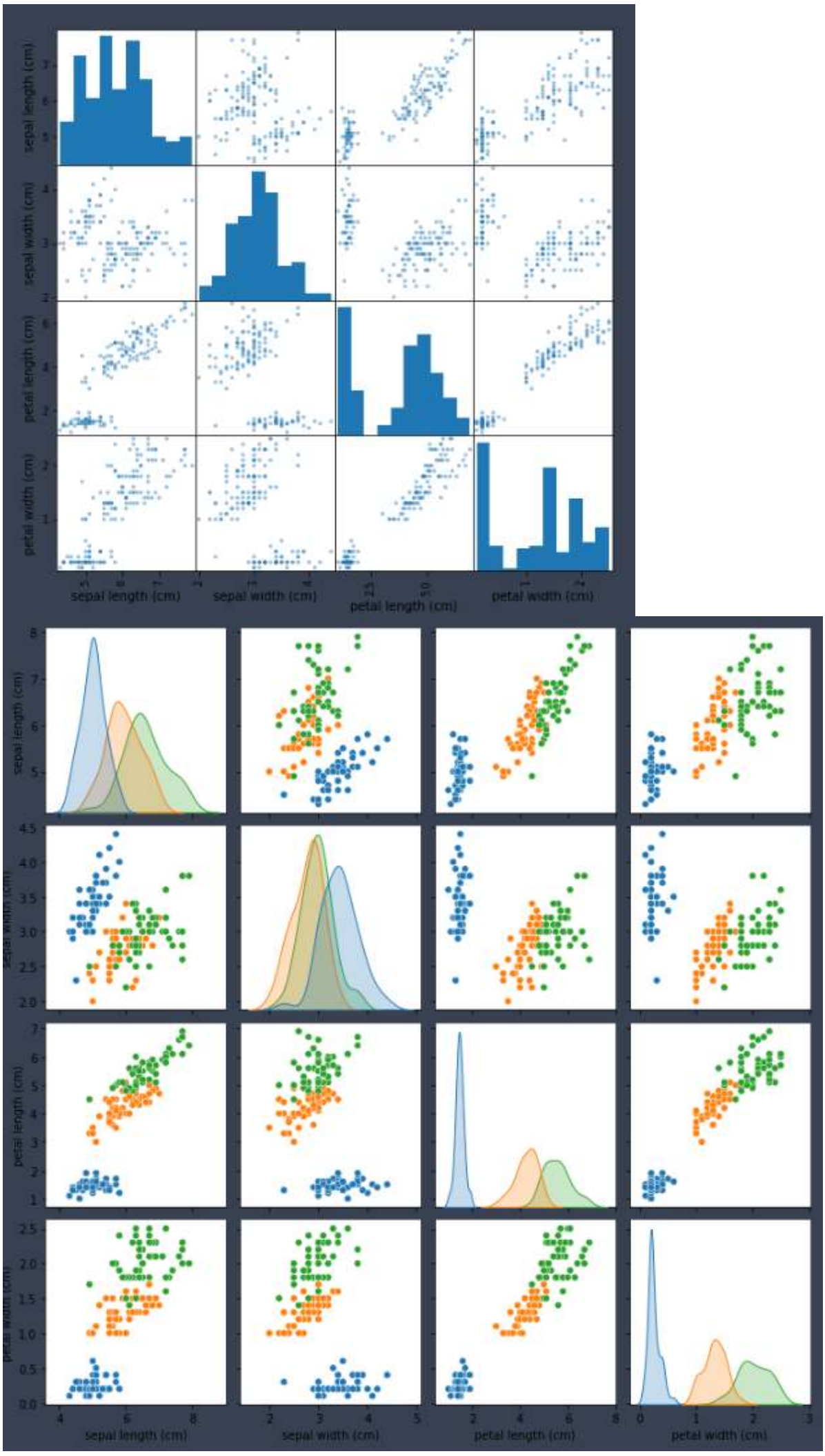
data	**데이터프레임**
alpha	투명도(0~1)
figsize	그래프 크기(x,y)
diagonal	대각선 밀도 그래프 종류{hist/kde}

sns.pairplot(iris,diag_kind='auto', hue='Class')

data	**데이터 프레임**
diag_kind	대각선 밀도 그래프 종류 {auto, hist, kde}
hue	색을 구분할 타겟변수

```
#산점도 행렬
import pandas as pd
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
from sklearn.datasets import load_iris
iris = load_iris()
iris = pd.DataFrame(iris.data, columns=iris.feature_names)
```

```
iris['class']=load_iris().target
iris['class']=iris['class'].map({0:'s1',1:'v1',2:'v2'})

#pyplot으로 산점도 행렬 그리기
scatter_matrix(iris,alpha=0.5, figsize = (8,8),diagonal='hist')
plt.show()

#seaborn으로 산점도 행렬 그리기
import seaborn as sns
sns.pairplot(iris,diag_kind='auto',hue='class')
plt.show()
```


sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)

상관계수 행렬 그래프

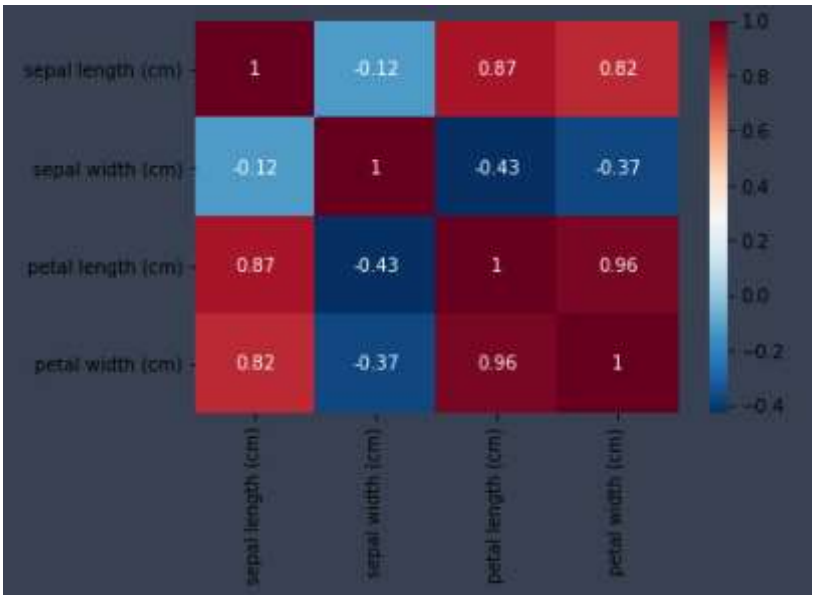
다수의 변수 간 상관관계를 파악하거나 독립변수 간 다중공선성을 파악할 수 있음.
상관관계는 -1~1 사이의 숫자 값으로 출력되고, 0에 가까울 수록 상관관계가 없고 -1에 가까울수록 음의 상관관계, 1에 가까울수록 양의 상관관계를 가짐.

상관계수 구하기
data = data.corr(method = 'pearson')
method : 상관분석 방법 {'pearson', 'kendall' , 'spearman'}

상관계수 행렬 그래프
sns.heatmap(data, xticklabels - data.columns, yticklabels=data.columns,
cmap='RdBu_r',annot = True)

data	상관행렬을 그릴 데이터의 상관계수 데이터프레임
xticklabels	x축의 라벨명
yticklabels	y축의 라벨명
cmap	히트맵의 색깔 지정
annot	True일 경우 상관관계를 텍스트로 표시

```
#상관계수 행렬 그래프
#method = {pearson, kendall, spearman}
iris_corr = iris.drop(columns='class').corr(method='pearson')
sns.heatmap(iris_corr,xticklabels = iris_corr.columns
            , yticklabels = iris_corr.columns
            , cmap='RdBu_r', annot=True)
plt.show()
```



판다스 Profile Report

구분	내용

구분	내용
Overview	데이터세트의 통계정보 및 컬럼의 체크 사항
Variables	컬럼의 통계정보와 Null 정보, 히스토그램 또는 막대그래프 등
Interactions	컬럼쌍별 산점도
Correlations	상관계수 결정 방식 별 상관행렬 그래프
Missing values	값의 개수 및 Null 값의 존재 여부 확인
Sample	가장 처음과 마지막의 10개의 값
Duplicate rows	중복 행

```
import pandas as pd
from sklearn.datasets import load_iris
import pandas_profiling

iris = load_iris()
iris = pd.DataFrame(iris.data, columns = iris.feature_names )
iris['class'] = load_iris().target
iris['class'] = iris['class'].map({0:'S1',1:' V1',2:'V2'})

iris.profile_report()
```