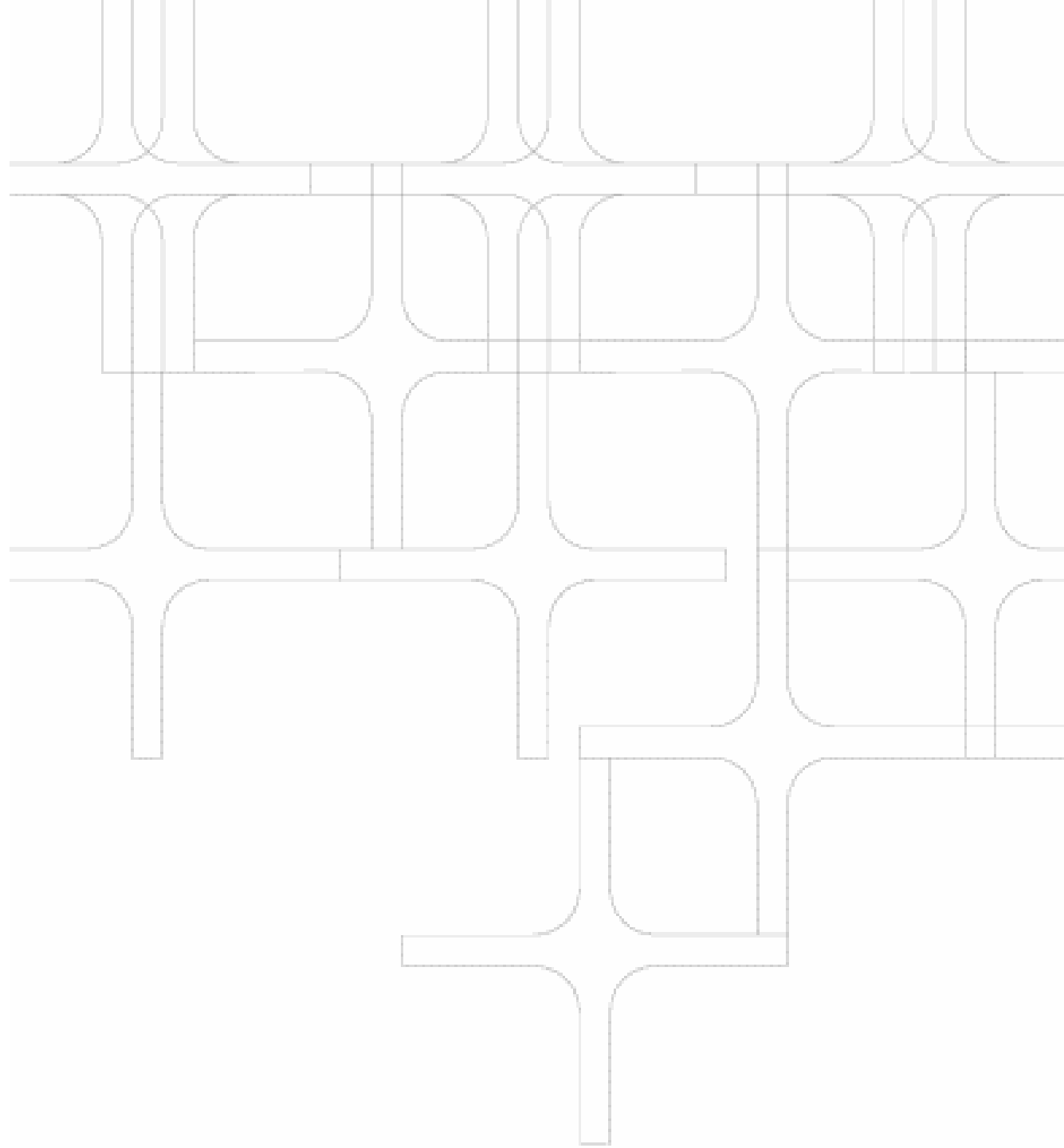


## Chapter. 03

# Exploratory Data Analysis

- 탐색적 데이터 분석
- 기술 통계
- 데이터의 분포 탐색
- 시각적 패턴을 통한 이해
- 주요 EDA 기법



- **탐색적 데이터 분석의 유래**

- 1977년 Tukey의 저서, '탐색적 데이터 분석' 에서 시작
- 요약 통계 및 데이터셋의 시각화에 도움이 되는 간단한 그림 (상자그림, 산점도 같은)

- **통계**

- 추론 중심, 작은 표본을 기반으로 대규모 모집단에 대한 결론을 도출하기 위한 복잡한 프로세스

- **데이터 분석**

- 1962년 John W. Tukey의 'The Future of Data Analysis' 논문, 통계의 개혁 요구
- 통계적 추론을 하나의 구성 요소로 포함하는 과학 분야 '데이터 분석' 제안

- **EDA (Exploratory Data Analysis)**

- 데이터를 분석하고 결과를 내는 과정에서 지속적으로 데이터에 대한 '탐색과 이해'를 가져야 한다는 것을 의미
- 전체적인 데이터의 구조를 시각적으로 분석하거나 분석방향을 가늠
- Matplotlib 같은 시각화 도구를 활용

- **데이터 시각화(Data Visualization)**

- 데이터를 쉽게 이해할 수 있도록 시각적으로 표현하고 전달되는 과정.
- 통계치를 그래프로 표현하는 통계 그래픽, 수많은 데이터를 한 장의 그림으로 요약하여 표현하는 인포그래픽, 문서에 사용된 단어의 빈도와 중요도를 시각적으로 표현하는 워드 클라우드 등 다양한 기법과 기능들이 있음
- 그래프라는 수단으로 정보를 명확하고 효과적으로 전달, 데이터의 의미를 쉽게 파악, 숨어있는 정보도 찾고 통찰력을 얻을 수 있음.
- 전체적 데이터의 구조를 시각적으로 분석하거나 분석방향을 가늠하기 위해 탐색적 데이터 분석(EDA)에서 사용

**기술 통계** : 수집한 데이터를 요약, 묘사, 설명하는 통계 기법

- 기술 통계는 표본 자체의 속성을 파악하는 데 주안점을 두는 데이터 분석방법
- 주로 표본에 속한 대상자들의 인구통계학적 속성과 함께 연구문제나 가설에 포함된 개별적인 변인에 대한 표본 대상자의 응답
- 즉 데이터 속성을 특정한 통계량을 사용해 요약해준다

목적	측정방법	개념
데이터의 중심을 이해	산술평균(mean) 중앙값(median) 최빈값(mode)	주어진 수의 합을 수의 개수로 나눈 값 주어진 값들을 크기의 순서대로 정렬했을 때 중앙에 위치하는 값 주어진 값 중에서 가장 많이 나오는 값
데이터가 흩어진 정 도를 이해	분산(variance) 표준편차(standard deviation) 사분위수(quantile)	주어진 값에서 평균을 뺀 값을 제곱하고, 모두 더한 후 전체 개수로 나눈 값(차이값의 제곱의 평균) 분산의 양의 제곱근 데이터 표본을 4개의 동일한 부분으로 나눈 값

데이터의 흩어진 정도 : 변동성

1,      2,      3,      4,      5

편차(deviation)

-2      -1      0      1      2

편차의 제곱

4      1      0      1      4

분산(variance)

$(4 + 1 + 0 + 1 + 4) / 5 = 2$

표준 편차(standard deviation)

$\sqrt{2}$

## 백분위수 Percentiles 를 이용한 분포 탐색

데이터를 백분위수로 나눈 값

데이터 집합을 순서대로 나열했을 때 전체의 특정 비율이 되는 지점

제  $100 \times p$  백분위수 구하는 방법

1. 관측값을 작은 순서로 배열한다.
2. 관측값의 개수( $n$ )에  $p$ 를 곱한다.
  - 1) 만약  $n \times p$ 가 정수이면,  $n \times p$  번째로 작은 관측값과  $n \times p + 1$  번째로 작은 관측값의 평균을 제  $100 \times p$  백분위수로 한다.
  - 2) 만약  $n \times p$ 가 정수가 아니면,  $n \times p$ 에서 정수부분에 1을 더한 값  $m$ 을 구한 수,  $m$  번째 작은 관측값을 제  $n \times p$  백분위수로 한다.

1, 2, 2, 3, 3, 3, 4

$7 * 0.5 = 3.5 \rightarrow 4\text{번째 값이 } 50\text{번째 값} \rightarrow \text{중앙값}$

#### 백분위수 Percentiles 를 이용한 분포 탐색

데이터의 위치와 분포를 이해하는 도구로 활용

1. 데이터를 판정하는 기준선으로 설정
2. 개별 데이터의 전체 데이터 속 위치를 파악하는데 사용
3. 다른 집단, 다른 시간대 데이터와 현재 데이터를 비교해서 변화의 정도, 추세를 파악
4. 이상치 설정 : 매우 높거나 낮은 퍼센타일 값

## 데이터 불러오기

## 3가지 품종의 붓꽃 분류

### Iris dataset uci

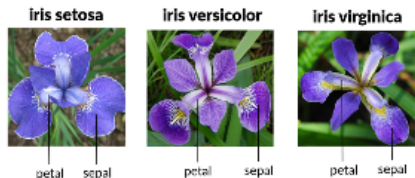
Iris Dataset From Uci Machine Learning

Data Card

Code (8)

Discussion (0)

Suggestions (0)



#### About Dataset

The Iris dataset was used in R.A. Fisher's classic 1936 paper, The Use of Multiple Measurements in Taxonomic Problems, and can also be found on the UCI Machine Learning Repository.

It includes three iris species with 50 samples each as well as some properties about each flower. One flower species is linearly separable from the other two, but the other two are not linearly separable from each other.

The columns in this dataset are:

Id

SepalLengthCm

SepalWidthCm

PetalLengthCm

PetalWidthCm

Species

#### Usability ⓘ

10.00

#### License

[CC0: Public Domain](#)

#### Expected update frequency

Never

#### Tags

Earth and Nature

Beginner

Intermediate

NumPy

Matplotlib

sklearn

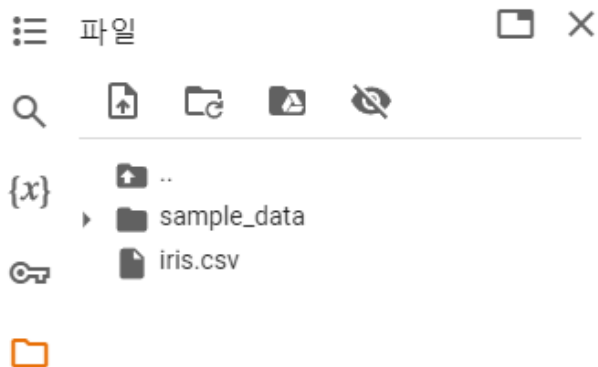


## 데이터 불러오기

## 3가지 품종의 붓꽃 분류

<https://www.kaggle.com/datasets/saurabh00007/iris.csv>

## ▼ 데이터 불러오기



```
1 import pandas as pd
2 df = pd.read_csv('iris.csv')
3 df
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows x 5 columns

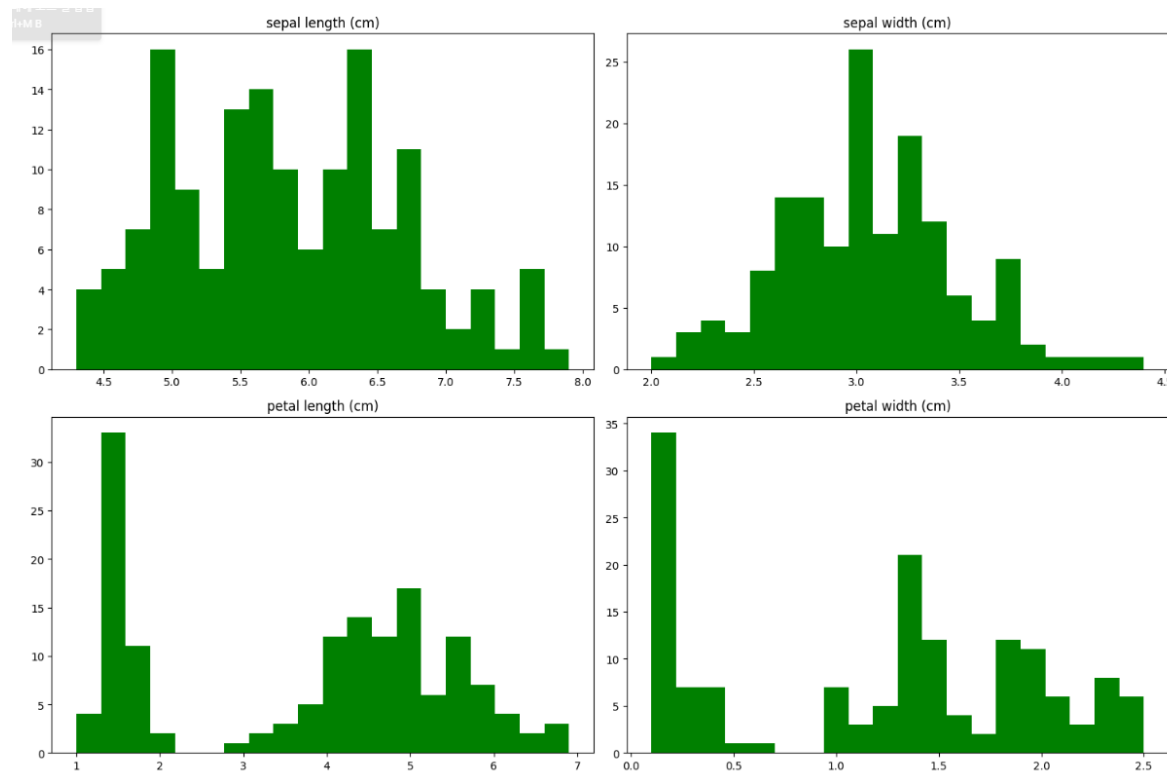
```
1 df.info()
```

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 150 entries, 0 to 149  
Data columns (total 5 columns):  
# Column Non-Null Count Dtype  
--- ---  
0 sepal.length 150 non-null float64  
1 sepal.width 150 non-null float64  
2 petal.length 150 non-null float64  
3 petal.width 150 non-null float64  
4 variety 150 non-null object  
dtypes: float64(4), object(1)  
memory usage: 6.0+ KB

## 데이터의 분포 탐색

### ✓ 5-2) 각 속성별 히스토그램으로 분포 모양 확인하기

```
1 import matplotlib.pyplot as plt
2
3 feature_names = iris.feature_names
4
5 # 히스토그램 그리기
6 plt.figure(figsize=(15, 10))
7 for i in range(len(feature_names)):
8     plt.subplot(2, 2, i+1) # 4개의 그래프가 들어갈 수 있는 그리드 생성
9     plt.hist(iris_data[:, i], bins=20, color='green')
10    plt.title(feature_names[i])
11
12 plt.tight_layout()
13 plt.show()
14
```



<https://www.kaggle.com/datasets/saurabh00007/iriscsv>

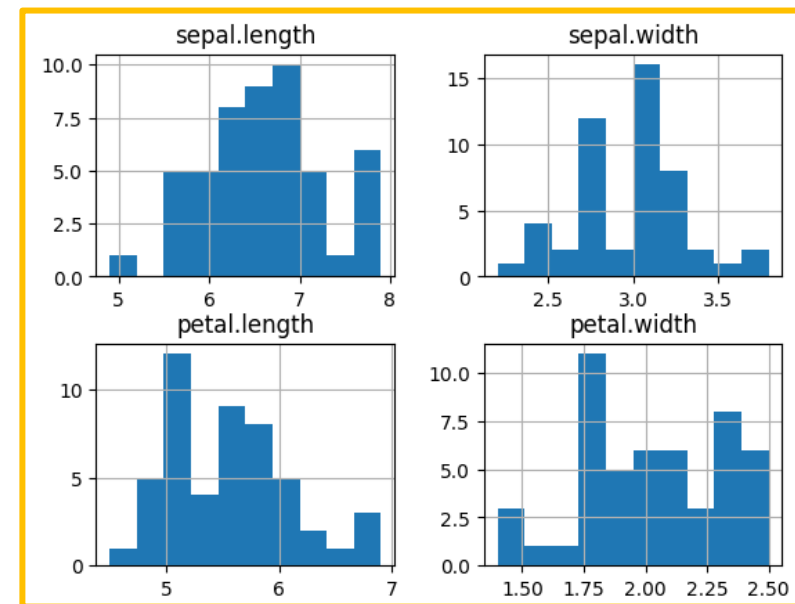
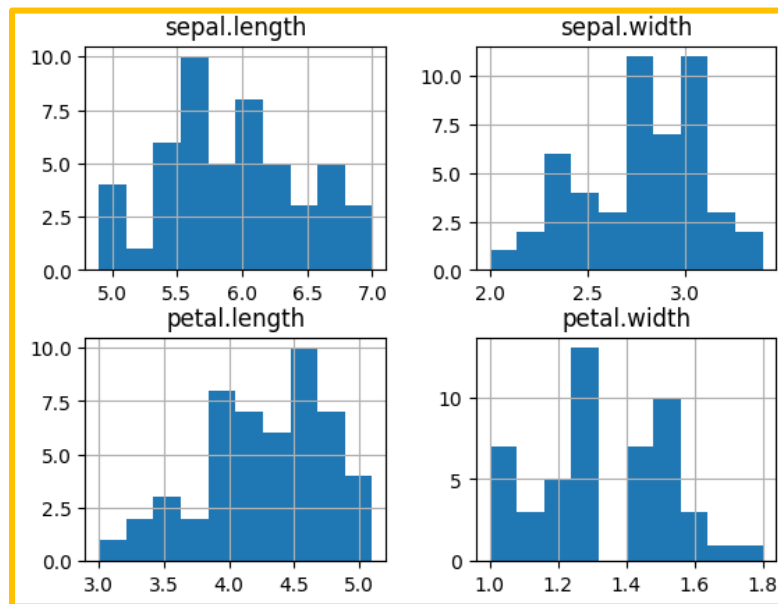
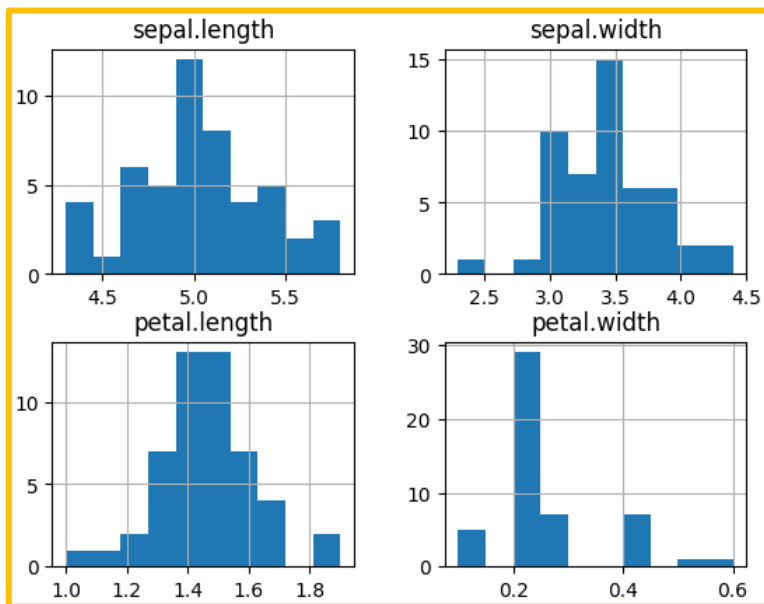
## 데이터의 분포 탐색

✓ 1) 품종별로 그룹화 한 후 히스토그램 그리기

```
1 # 그룹화
2 df1 = df.groupby('variety')
3 df1.head()
```

```
1 # 히스토그램 그리기
2 df1.hist()
```

```
variety
Setosa      [[Axes(0.125, 0.545217:0.336957x0.334783), Axes...
Versicolor  [[Axes(0.125, 0.545217:0.336957x0.334783), Axes...
virginica    [[Axes(0.125, 0.545217:0.336957x0.334783), Axes...
dtype: object
```




## 기술통계 분석

```

1 import seaborn as sns
2 import pandas as pd
3
4 # 아이리스 데이터셋 로드
5 iris = sns.load_dataset('iris')
6
7 # 품종별로 기술통계 실행
8 grouped_iris = iris.groupby('species')
9 description = grouped_iris.describe()
10 description
11

```



	sepal_length								sepal_width								petal_length								petal_width							
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	max	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max			
species																																
setosa	50.0	5.006	0.352490	4.3	4.800	5.0	5.2	5.8	50.0	3.428	...	1.575	1.9	50.0	0.246	0.105386	0.1	0.2	0.2	0.3	0.6											
versicolor	50.0	5.936	0.516171	4.9	5.600	5.9	6.3	7.0	50.0	2.770	...	4.600	5.1	50.0	1.326	0.197753	1.0	1.2	1.3	1.5	1.8											
virginica	50.0	6.588	0.635880	4.9	6.225	6.5	6.9	7.9	50.0	2.974	...	5.875	6.9	50.0	2.026	0.274650	1.4	1.8	2.0	2.3	2.5											

3 rows × 32 columns

3 rows × 32 columns

✓ 2) 품종별로 평균, 중간값, 표준편차, 백분위 구하기

```
1 print(df1.mean())
```

	sepal.length	sepal.width	petal.length	petal.width
variety				
Setosa	5.006	3.428	1.462	0.246
Versicolor	5.936	2.770	4.260	1.326
virginica	6.588	2.974	5.552	2.026

```
[10] 1 print(df1.median())
```

	sepal.length	sepal.width	petal.length	petal.width
variety				
Setosa	5.0	3.4	1.50	0.2
Versicolor	5.9	2.8	4.35	1.3
virginica	6.5	3.0	5.55	2.0

```
[11] 1 print(df1.std())
```

	sepal.length	sepal.width	petal.length	petal.width
variety				
Setosa	0.352490	0.379064	0.173664	0.105386
Versicolor	0.516171	0.313798	0.469911	0.197753
virginica	0.635880	0.322497	0.551895	0.274650

```
[12] 1 print(df1.quantile())
```

	sepal.length	sepal.width	petal.length	petal.width
variety				
Setosa	5.0	3.4	1.50	0.2
Versicolor	5.9	2.8	4.35	1.3
virginica	6.5	3.0	5.55	2.0

### 4-1) 각 속성 별로 value counts 구하기

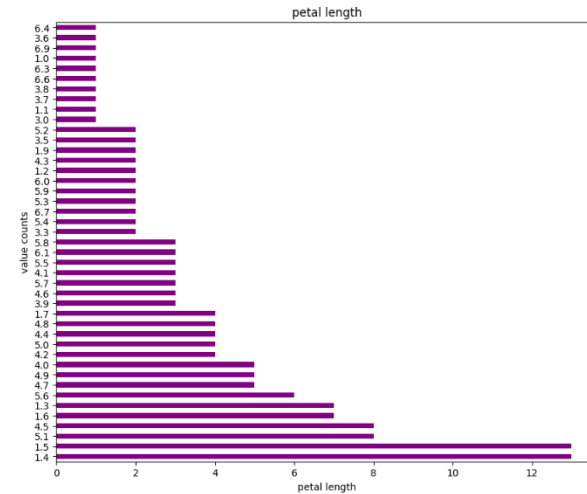
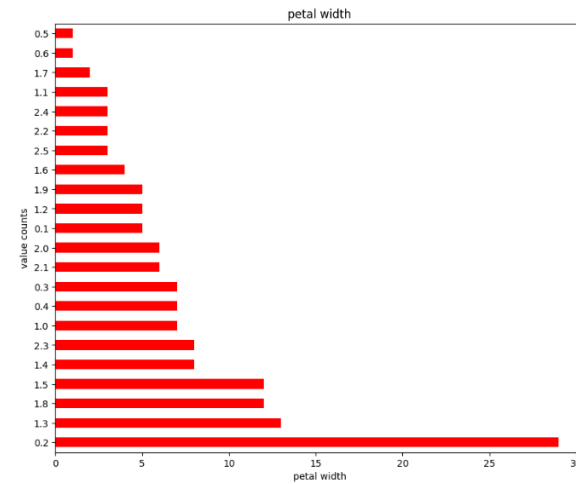
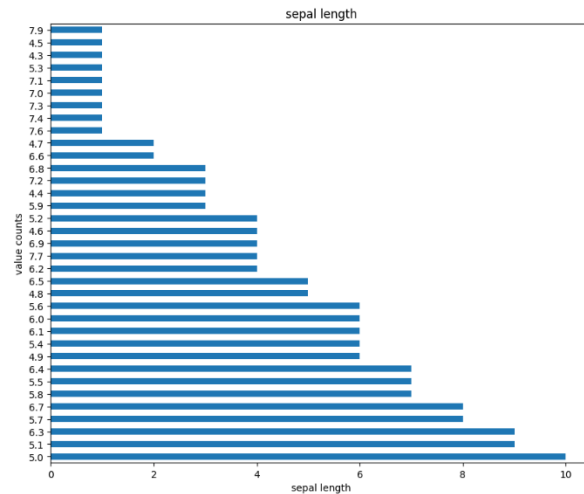
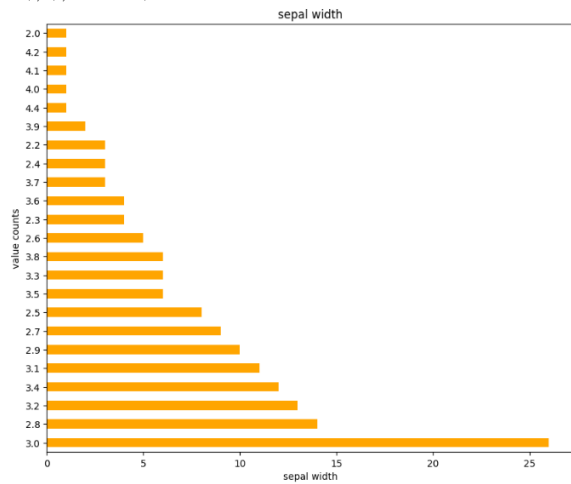
```
1 print("sepal length 값들의 갯수")
2 print(df['sepal.length'].value_counts())
```

```
sepal length 값들의 갯수
5.0    10
5.1     9
6.3     9
5.7     8
6.7     8
5.8     7
```

### 4-2) 각 속성 별로 value counts를 막대 그래프로 그리기

```
1 import matplotlib.pyplot as plt
2
3 f, axes = plt.subplots(4,1)
4 f.set_size_inches((10, 40))
5 plt.subplots_adjust(wspace = 0.3, hspace = 0.3)
6
7 ax1=df['sepal.width'].value_counts().plot(kind='barh',ax=axes[0],color='orange')
8 ax1.set_title('sepal width')
9 ax1.set_xlabel('sepal width')
10 ax1.set_ylabel('value counts')
11
12 ax2=df['sepal.length'].value_counts().plot(kind='barh',ax=axes[1])
13 ax2.set_title('sepal length')
```

Text (0, 0.5, 'value counts')



## 5) 각 속성 별 IQR (Inter Quartile Range) 계산하기

```
1 percentile75_sl=df['sepal.length'].quantile(0.75)
2 percentile25_sl=df['sepal.length'].quantile(0.25)
3 IQR_sl = percentile75_sl - percentile25_sl
4 print('IQR of sepal length:',IQR_sl)
5
6 percentile75_sw=df['sepal.width'].quantile(0.75)
7 percentile25_sw=df['sepal.width'].quantile(0.25)
8 IQR_sw = percentile75_sw - percentile25_sw
9 print('IQR of sepal width:',IQR_sw)
10
11 percentile75_pl=df['petal.length'].quantile(0.75)
12 percentile25_pl=df['petal.length'].quantile(0.25)
13 IQR_pl = percentile75_pl - percentile25_pl
14 print('IQR of petal length:',IQR_pl)
15
16 percentile75_pw=df['petal.width'].quantile(0.75)
17 percentile25_pw=df['sepal.length'].quantile(0.25)
18 IQR_pw = percentile75_pw - percentile25_pw
19 print('IQR of petal width:',IQR_pw)
```

```
IQR of sepal length: 1.3000000000000007
IQR of sepal width: 0.5
IQR of petal length: 3.4999999999999996
IQR of petal width: -3.3
```

## 5-1) 각 속성별 40 백분위수 계산하기

```
1 import numpy as np
2 from sklearn.datasets import load_iris
3
4 # Iris 데이터셋 로드
5 iris = load_iris()
6 iris_data = iris.data
7
8 # Iris 데이터셋의 각 feature에 대한 40번째 백분위수 계산
9 percentiles_40 = np.percentile(iris_data, 40, axis=0)
10
11 print(percentiles_40)
12
```

```
[5.6  3.   3.9  1.16]
```

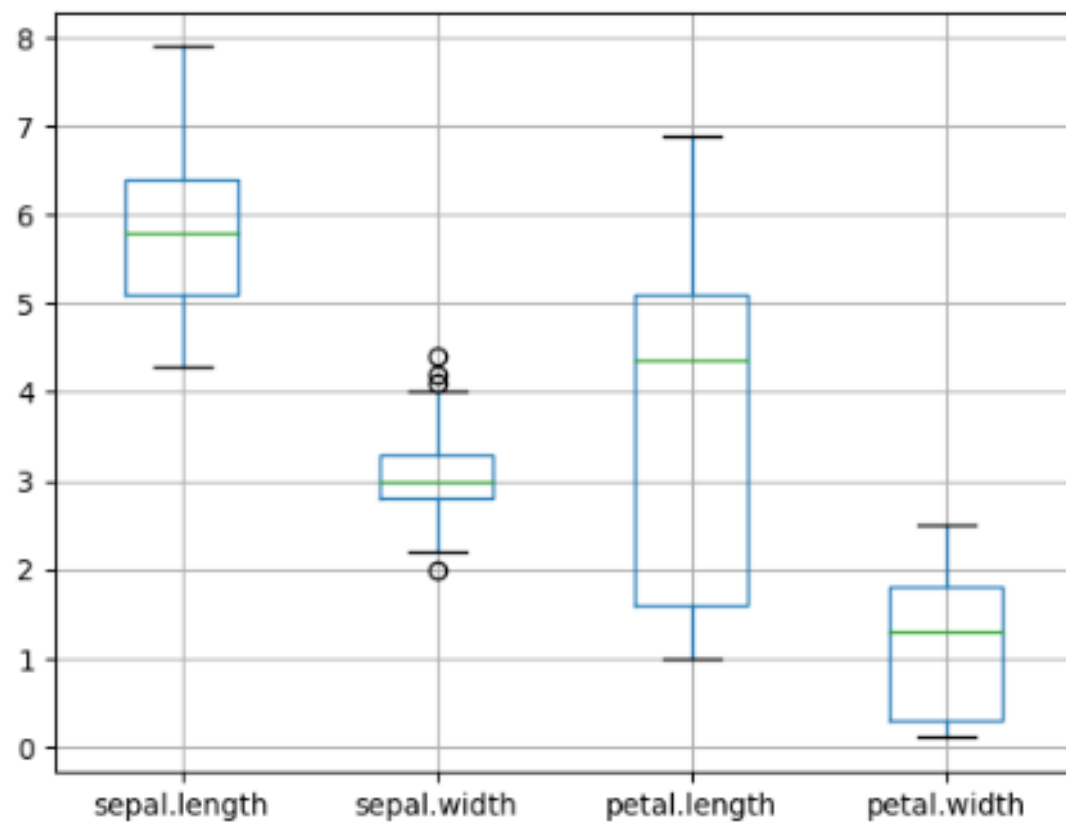
## ✓ 5-3) Boxplot 그리기



```
1 df.boxplot()
```



&lt;Axes: &gt;



## 이진, 범주형 데이터의 탐색

이진 데이터: Yes/No, True/False 등 두 가지 범주를 가지는 데이터

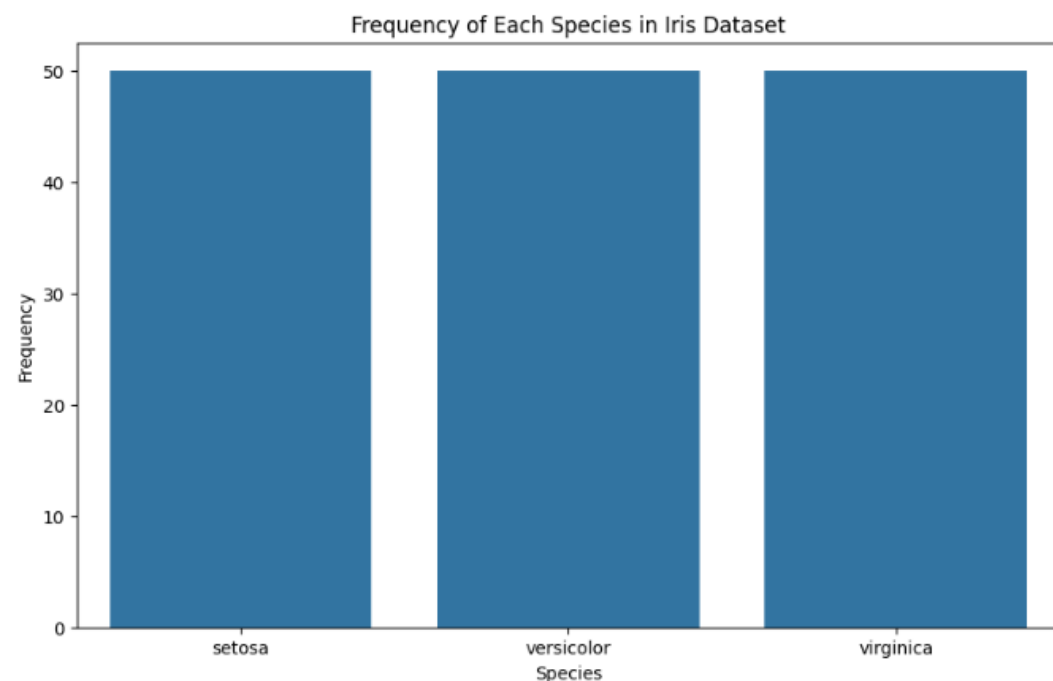
범주형 데이터 : 두 개 이상의 범주를 가지는 데이터

- 각 범주의 빈도수를 계산
- 각 범주의 비율 계산
- 빈도수 또는 비율을 바차트로 시각화
- 범주형 데이터의 경우 파이차트나 교차테이블, 스택바 차트 등으로 범주 간 관계 탐색



## ▼ 6-1) 바플롯으로 범주형 데이터의 탐색하기

```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3
4 # 범주형 데이터인 Species 추가
5 df['species'] = pd.Categorical.from_codes(iris.target, iris.target_names)
6
7 # 범주별 빈도수 계산
8 species_counts = df['species'].value_counts()
9
10 # 빈도수 시각화
11 plt.figure(figsize=(10, 6))
12 sns.barplot(x=species_counts.index, y=species_counts.values)
13 plt.title('Frequency of Each Species in Iris Dataset')
14 plt.xlabel('Species')
15 plt.ylabel('Frequency')
16 plt.show()
17
```



## 상관관계 Correlation

두 변수 사이의 관계를 측정하고, 묘사하기 위해 이용하는 기법 (Gravetter, Wallnau, 2009)

상관 계수 : 두 변수 사이의 관계를 반영하는  $-1 \sim +1$  사이의 수치

양의 상관 관계(직접 상관관계) 와 음의 상관 관계(간접 상관관계)

상관계수 크기	일반적인 해석
0.8 ~ 1.0	매우 강한
0.6 ~ 0.8	강한
0.4 ~ 0.6	중간 정도
0.2 ~ 0.4	약한
0.0 ~ 0.2	매우 약한

X	Y	관계 유형	값의범위	예시
증가	증가	직접/양의상관관계	0~ +1	공부를 더 많이 하면, 시험점수도 오를 것이다
감소	감소	직접/양의상관관계	0~ +1	예금을 더 적게 하면, 이자도 적을 것이다
증가	감소	간접/음의상관관계	-1 ~ 0	더 많이 운동하면, 몸무게는 감소할 것이다
감소	증가	간접/음의상관관계	-1 ~ 0	더 적은 시간 동안 문제를 풀면, 문제를 틀릴 가능성이 높아진다

### 6-2) 상관관계 계산하기

```
1 df.corr()
```

```
<ipython-input-33-2f6f6606aa2c>:1: FutureWarning: The default value of nume
df.corr()
```

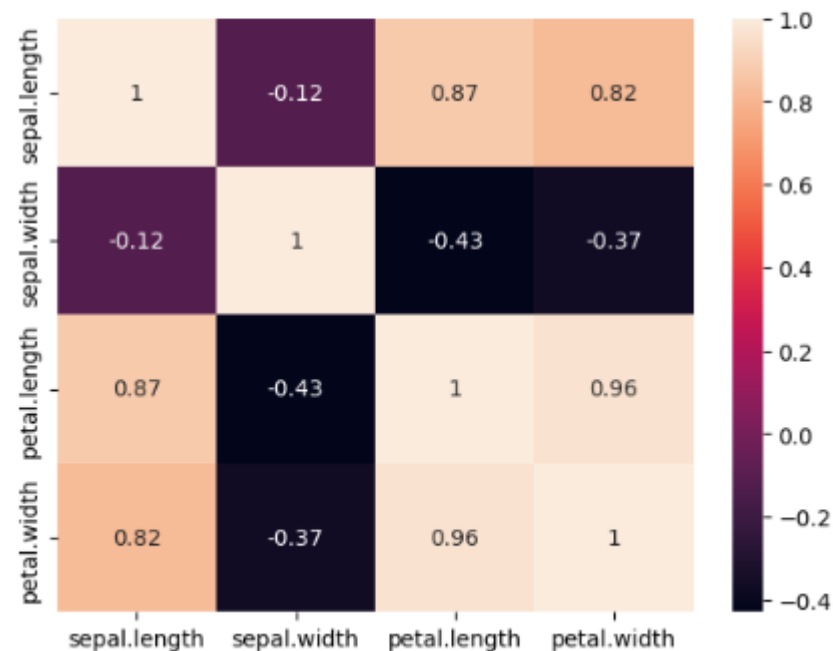
	sepal.length	sepal.width	petal.length	petal.width
sepal.length	1.000000	-0.117570	0.871754	0.817941
sepal.width	-0.117570	1.000000	-0.428440	-0.366126
petal.length	0.871754	-0.428440	1.000000	0.962865
petal.width	0.817941	-0.366126	0.962865	1.000000

### 6-3) 상관관계 Heatmap으로 그리기 (seaborn 활용)

```
1 import seaborn as sns
2 sns.heatmap(df.corr(), annot=True)
```

```
<ipython-input-34-a7434eca7f55>:2: FutureWarning: The default value of nume
sns.heatmap(df.corr(), annot=True)
```

```
<Axes: >
```



## Chapter. 03

# Exploratory Data Analysis

- 탐색적 데이터 분석
- 기술 통계
- 데이터의 분포 탐색
- 시각적 패턴을 통한 이해
- 주요 EDA 기법

