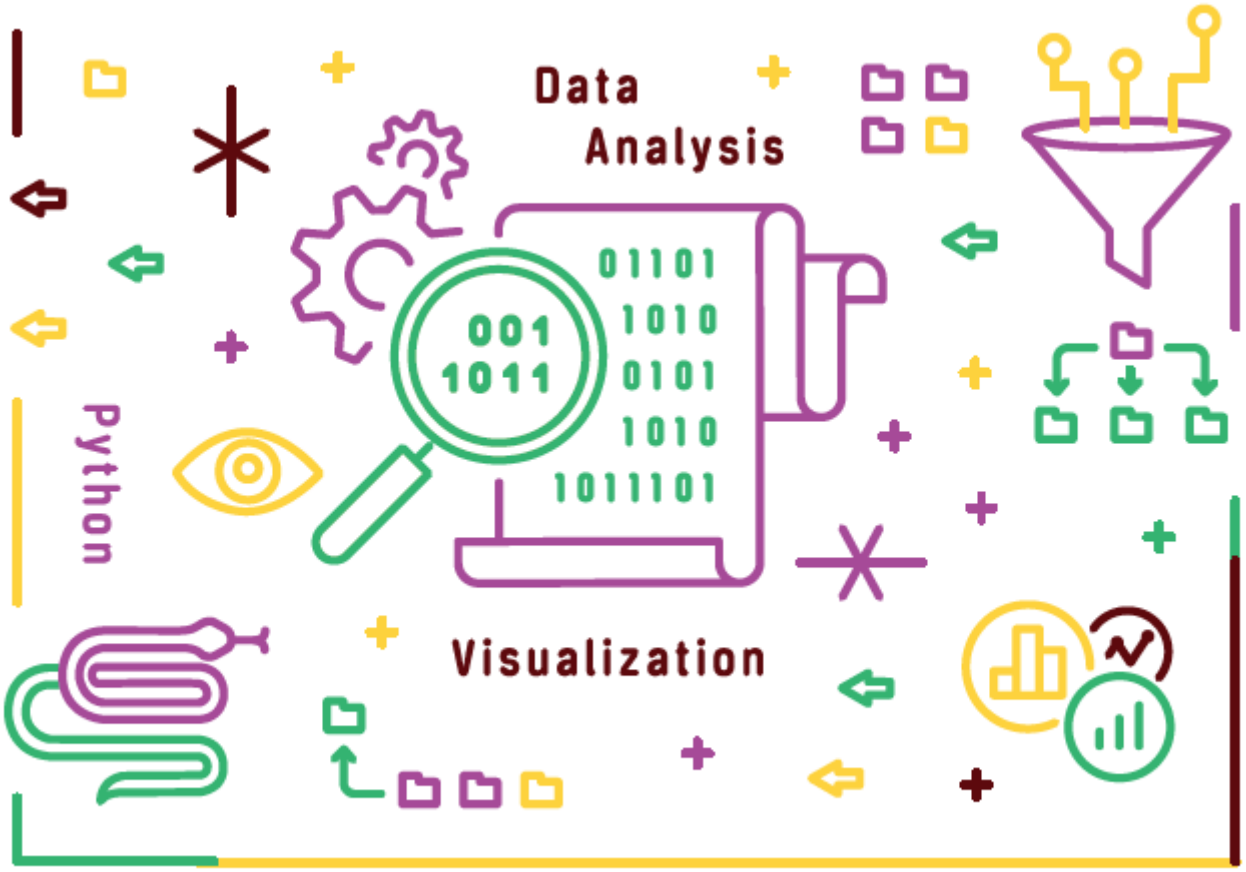

Data
Analysis
01101
1010
0101
1010
1011101
001
1011
Python
Visualization

<6장> 판다스 활용

학습 목표

- 판다스로 데이터를 처리하는 과정을 익힌다.
- 판다스 라이브러리를 활용하여 붓꽃 데이터 셋을 분석하는 방법을 익힌다.
- 판다스 라이브러리를 활용하여 타이타닉 데이터 셋을 분석하는 방법을 익힌다.
- 판다스로 데이터 시각화하는 방법을 익힌다.

목차

01

붓꽃 데이터 분석

1. 붓꽃 데이터 분석

데이터 설명

- 통계학자가 정리한 아이리스 데이터는 붓꽃의 3가지 종을 각각의 특성에 맞게 분류
- 꽃잎의 각 부분의 너비와 길이 등을 측정한 데이터이며 150개의 레코드로 구성

그림 6-1. 붓꽃의 3가지 종

1. 붓꽃 데이터 분석

필드의 이해

- 5개의 필드로 구성

표 6-1. 데이터 정보 : 붓꽃(iris) 정보에 관한 데이터 파일

변수명	변수 설명
Sepal Length	꽃받침의 길이 정보
Sepal Width	꽃받침의 너비 정보
Petal Length	꽃잎의 길이 정보
Petal Width	꽃잎의 너비 정보
Species	꽃의 종류 정보, Setosa/Versicolor/Virginica 3종류로 구분

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
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

그림 6-2. 원본 데이터 형태

1. 붓꽃 데이터 분석

붓꽃 데이터 읽어와서 확인

- (1) 1행에서는 파이썬 라이브러리 판다스(Pandas)를 임포트(Import)
- 3행에서는 판다스의 read_csv를 사용하여 'iris.csv' 파일을 읽어 데이터프레임 변수 iris에 저장
- 4행에서는 iris의 처음 5행을 출력한다. head 함수에 숫자를 입력하면, 숫자만큼의 행을 추출

'붓꽃' 파일 읽어오기

```
1  import pandas as pd
2  filename='iris.csv' # 코랩 파일 경로 : '/content/drive/MyDrive/iris.csv'
3  iris=pd.read_csv(filename)
4  iris.head()
```

〈실행결과〉

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
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

1. 붓꽃 데이터 분석

붓꽃 데이터 읽어와서 확인

- (2) 데이터 전체적인 구조를 확인

데이터의 기본 정보 출력

```
1    iris.info()
```

〈실행결과〉

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entrles, 0 to 149
Data columns (total 5 columns) :
 #     column       Non-Null Count     Dtype
---     ------      --------------     ----
0  SepalLength     150 non-null       float64
1  SepalWidth      150 non-null       float64
2  PetalLength     150 non-null       float64
3  PetalWidth      150 non-null       float64
3  Species         150 non-null       object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

1. 붓꽃 데이터 분석

붓꽃 데이터 읽어와서 확인

- (3) 데이터 기초 통계량을 확인

데이터의 기초 통계량 출력

```
1   iris.describe()
```

〈실행결과〉

	SepalLength	SepalWidth	PetalLength	PetalWidth
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

1. 붓꽃 데이터 분석

붓꽃 데이터 읽어와서 확인

- (4) value_counts( )를 적용하여 품종별 데이터 개수 구하기

품종별 개수 구하기

```
1  count=pd.DataFrame(iris['Species'].value_counts())
2  count
```

〈실행결과〉

	Species
setosa	50
versicolor	50
virginica	49

1. 붓꽃 데이터 분석

■ 붓꽃 데이터 전처리

▪ 결측치 확인하기

- 결측값이면 True, 정상 데이터면 False를 반환

결측치 확인

1

〈코드〉

```
iris.isnull().sum()
```

〈실행결과〉

```
SepalLength     0
SepalWidth      0
PetalLength     0
PetalLidth      0
Species         0
dtype:  int 64
```

1. 붓꽃 데이터 분석

붓꽃 데이터 전처리

- 중복 데이터 확인하기
  - 중복되면 True, 아니면 False를 반환

중복 데이터 확인

〈코드〉

```
1  iris.duplicated().sum()
```

〈실행결과〉

```
1
```

실제 중복 데이터 확인

```
1  index=iris.duplicated()
2  iris.loc[index, :]
```

〈실행결과〉

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
142	5.8	2.7	5.1	1.9	virginica

1. 붓꽃 데이터 분석

붓꽃 데이터 전처리

- 중복 데이터를 모두 확인하기
  - duplicated( ) 결과를 가지고 어떤 데이터 행끼리 중복되는지 확인

중복 데이터 모두 확인

〈코드〉

```
1  result=(iris['SepalLength']==5.8) &
   (iris['PetalWidth']==1.9)
2  iris.loc[result,:]
```

〈실행결과〉

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
```

1. 붓꽃 데이터 분석

붓꽃 데이터 전처리

중복 데이터 삭제하기

- duplibcates( )를 이용하여 나머지 중복 데이터 제거

중복 데이터 삭제

```
1  iris=iris.drop_duplicates()
2  #iris.duplicated().sum()
3  result=(iris['SepalLength']==5.8) & (iris['PetalWidth']==1.9)
4  iris.loc[result,:]
```

〈실행결과〉

	SepalLength	SepalWidth	PetalLength	PetalWidth	Species
101	5.8	2.7	5.1	1.9	virginica

1. 붓꽃 데이터 분석

■ 붓꽃 데이터 그룹핑

'품종' 열을 기준으로 합계 구하기

```
1  iris.groupby('Species').sum()
```

〈실행결과〉

Species	SepalLength	SepalWidth	PetalLength	PetalWidth
setosa	250.3	171.4	73.1	12.3
versicolor	296.8	138.5	213.0	66.3
virginica	323.6	146.0	272.5	99.4

'품종' 열을 기준으로 평균 구하기

```
1  iris.groupby('Species').mean()
```

〈실행결과〉

Species	SepalLength	SepalWidth	PetalLength	PetalWidth
setosa	5.006000	3.428000	1.462000	0.246000
versicolor	5.936000	2.770000	4.260000	1.326000
virginica	6.604082	2.979592	5.561224	2.028571

1. 붓꽃 데이터 분석

■ 붓꽃 데이터 그룹핑

sklearn(사이킷런) 패키지에서 붓꽃 데이터 셋 불러오기

```
1  from sklearn.datasets import load_iris
2  iris_dataset=load_iris()
3  #iris_dataset.keys()
4  df=pd.DataFrame(iris_dataset['data'], columns=iris_dataset['feature_names'])
5  df.head()
```

〈실행결과〉

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

시본 모듈에서 붓꽃 데이터 셋 불러오기

```
1  import seaborn as sns
2  df=sns.load_dataset('iris')
3  df.head()
```

〈실행결과〉

	sepal_length	sepal_width	petal_length	petal_width	species
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

1. 붓꽃 데이터 분석

판다스의 데이터 시각화

- 판다스의 시리즈나 데이터프레임은 'plot'이라는 시각화 메소드를 내장

옵션	종류	옵션	종류
line	선 그래프	kde	커널 밀도 그래프
bar	막대 그래프 – 수직	area	면적 그래프
barh	막대 그래프 – 수평	pie	원형 그래프
his	히스토그램 그래프	scatter	산점도 그래프
box	박스 그래프	hexbin	고밀도 산점도 그래프

1. 붓꽃 데이터 분석

판다스의 데이터 시각화

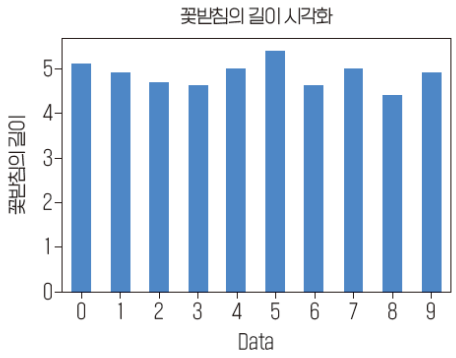
막대 그래프 그리기

- kind = 'bar' 옵션을 통해 구현

꽃 받침 길이의 시각화

```
1  import matplotlib.pyplot as plt
2  import pandas as pd
3  filename='iris.csv'
4  iris=pd.read_csv(filename)
5  iris.head()
6  iris.SepalLength[:10].plot(kind='bar', rot=0)
7  plt.title("꽃받침의 길이 시각화")
8  plt.xlabel("Data")
9  plt.ylabel("꽃받침의 길이")
10 plt.show()
```

〈실행결과〉



붓꽃 종류별 꽃 받침 길이의 평균에 대한 시각화

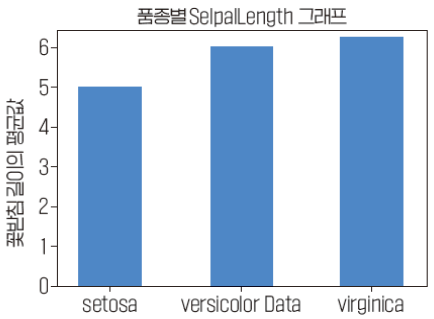
```
1  df2=iris.groupby(iris['Species']).mean()
2  df2
```

〈실행결과〉

Species	SepalLength	SepalWidth	PetalLength	PetalWidth
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

```
1  df2.SepalLength[:].plot(kind='bar', rot=0)
2  plt.title('품종별 SelpalLength 그래프')
3  plt.xlabel('Data')
4  plt.ylabel('꽃받침 길이의 평균값')
5  plt.show()
```

〈실행결과〉



1. 붓꽃 데이터 분석

판다스의 데이터 시각화

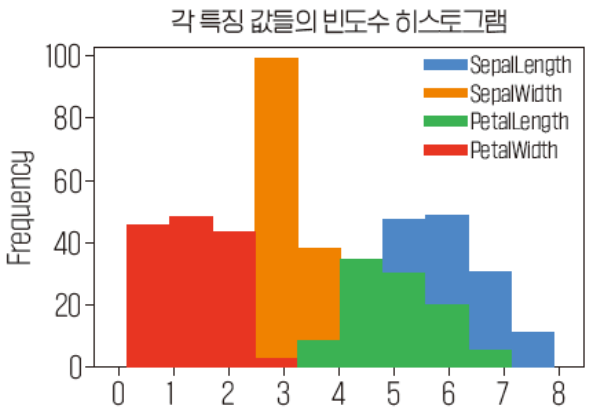
히스토그램 그래프 그리기

- kind = 'hist' 옵션을 통해 구현

각 특징 값들의 빈도수 그래프 그리기

```
1  iris.plot(kind='hist') #iris.plot.hist()
2  plt.title('각 특징 값들의 빈도수 히스토그램')
3  plt.show()
```

〈실행결과〉



1. 붓꽃 데이터 분석

■ 판다스의 데이터 시각화

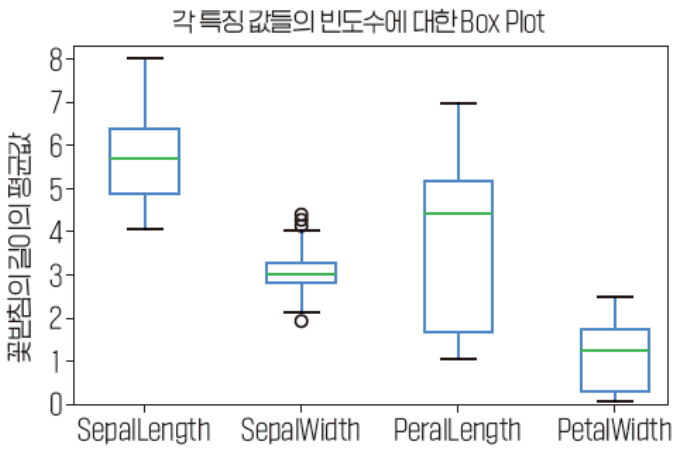
▪ 상자 그래프 그리기

- kind = 'box' 옵션을 통해 구현

각 특징 값들의 빈도수 그래프 그리기

```
1  iris.plot(kind='box')
2  plt.title('각 특징 값들의 빈도수에 대한 Box Plot')
3  plt.xlabel('특징')
4  plt.ylabel('데이터 값')
5  plt.show()
```

〈실행결과〉



1. 붓꽃 데이터 분석

판다스의 데이터 시각화

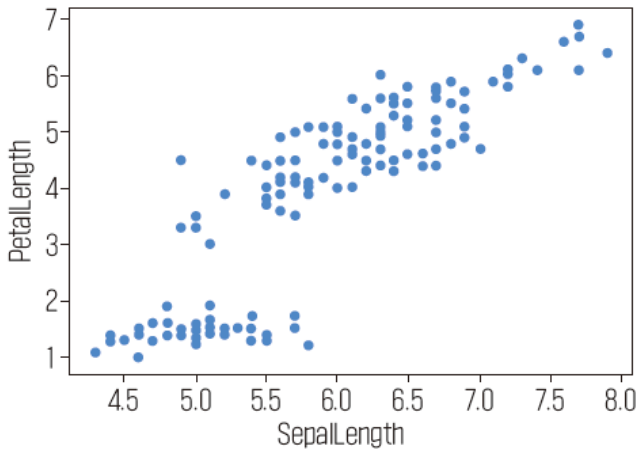
산점도 그래프 그리기

- kind = 'scatter' 옵션을 통해 구현

산점도 그래프 그리기

```
1  iris.plot(x='SepalLength', y='PetalLength', kind='scatter')
2  plt.show()
3
```

〈실행결과〉



02

타이타닉 데이터 분석

2. 타이타닉 데이터 분석

데이터 설명

- 데이터 사이언스나 머신러닝 분야에서 입문자용으로 가장 많이 사용하는 예제 중 하나
- 데이터 분석 경연 사이트인 캐글(Kaggle)에서 입문자용으로 가장 많이 사용
- 당시 사망자와 생존자를 구분하는 요인 분석을 통해 승객들의 생존 여부를 예측

2. 타이타닉 데이터 분석

필드의 이해

- 타이타닉 데이터 셋에 포함된 12개의 변수

표 6-2. 데이터 정보 : 타이타닉(Titanic) 정보에 관한 데이터 파일

변수명	변수 설명
PassengerId	승객 번호
Survived	생존 여부 : 0=사망, 1=생존
Pclass	객실 등급 : 1=등급, 2=등급, 3=등급
Name	승객 이름
Sex	성별
Age	나이
SibSp	함께 탑승한 형제와 배우자의 수
Parch	함께 탑승한 부모, 아이의 수
Ticket	티켓 번호
Fare	탑승 요금
Cabin	객실 번호
Embarked	탑승 항구 : C=Cherbourg/Q=Queenstown/S=Southampton

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

그림 6-3. 원본 데이터 형태

2. 타이타닉 데이터 분석

타이타닉 데이터 읽어와서 확인

- (1) 1행에서는 파이썬 라이브러리 판다스(Pandas)를 임포트(Import)
- 3행에서는 판다스의 read_csv를 사용하여 데이터프레임 변수 titanic에 저장
- 3행에서는 titanic의 처음 5행을 출력

'타이타닉' 파일 읽어오기

```
1  import pandas as pd
2  titanic=pd.read_csv('train.csv')
3  titanic.head()
```

〈실행결과〉

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

2. 타이타닉 데이터 분석

■ 타이타닉 데이터 읽어와서 확인

- (2) 데이터 전체적인 구조 확인

데이터의 기본 정보 출력

```
1   titanic.info()
```

〈실행결과〉

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

2. 타이타닉 데이터 분석

■ 타이타닉 데이터 읽어와서 확인

- (3) 데이터 기초 통계량 확인

데이터의 기초 통계량 출력

```
1   titanic.describe()
```

〈실행결과〉

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

2. 타이타닉 데이터 분석

■ 타이타닉 데이터 읽어와서 확인

- (4) 요금 기준으로 오름차순 정렬한 결과 확인

요금(Fare) 기준 내림차순 정렬

```
1  titanic.sort_values('Fare',ascending=False)
```

〈실행결과〉

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	C
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B101	C
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B51 B53 B55	C
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2	19950	263.0000	C23 C25 C27	S
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	C23 C25 C27	S
...	...	...	...	...	...	...	...	...	...	...	...	...

2. 타이타닉 데이터 분석

■ 타이타닉 데이터 읽어와서 확인

- (5) 생존자별 인원수를 확인

생존자별 인원수 출력

1

〈코드〉

```
pd.DataFrame(titanic['Survived'].value_counts())
```

〈실행결과〉

	Survived
0	549
1	342

2. 타이타닉 데이터 분석

■ 타이타닉 데이터 전처리

- 결측치 확인하기

결측치 확인

1

〈코드〉

```
titanic.isnull().sum()
```

〈실행결과〉

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
```

2. 타이타닉 데이터 분석

■ 타이타닉 데이터 전처리

- 객실번호(Cabin) 컬럼 삭제

결측데이터가 포함된 컬럼 삭제

```
1  titanic.drop(['Cabin'],axis=1,inplace=True)
2  titanic.columns
```

〈실행결과〉

```
Index([ ' PassengerId ' ,  ' Survived ' ,  ' Pclass ' ,  ' Name ' ,  ' Sex ' ,  ' Age ' ,
' SibSp ' ,  ' Parch ' ,  ' Ticket ' ,  ' Fare ' ,  ' Embarked ' ], dtype= ' object ' )
```

2. 타이타닉 데이터 분석

타이타닉 데이터 전처리

- 결측데이터 최빈값으로 대체하기

최빈값 찾기

```
1  titanic['Embarked'].value_counts()
```

〈실행결과〉

```
S    644
C    168
Q     77
```

최빈값 대체

〈코드〉

```
1  titanic['Embarked']=titanic['Embarked'].fillna('S')
2  titanic.isnull().sum()
```

〈실행결과〉

```
PassengerId  0
Survived     0
Pclass       0
Name         0
Sex          0
Age        177
SibSp        0
Parch        0
Ticket       0
Fare         0
Cabin        0
Embarked     0
```

2. 타이타닉 데이터 분석

타이타닉 데이터 전처리

- 결측데이터 평균값으로 대체

평균값 대체

〈코드〉

```
1  avg=titanic['Age'].mean()
2  titanic['Age']=titanic['Age'].fillna(avg)
```

〈실행결과〉

```
PassengerId   0
Survived      0
Pclass        0
Name          0
Sex           0
Age           0
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin         0
Embarked      0
dtype:  int64
```

2. 타이타닉 데이터 분석

■ 타이타닉 데이터 그룹핑하기

항구별(Emparked) Pclass 컬럼의 평균값을 출력

```
1  titanic['Pclass'].groupby(titanic["Embarked"]).mean()
```

〈실행결과〉

```
Embarked
C        1.886905
Q        2.909091
S        2.346749
Name: Pcalss, dtype:  float64
```

Pclass, Sex 컬럼 열을 기준으로 평균 구하기

```
1  titanic.groupby(["Pclass", "Sex"]).mean()
```

〈실행결과〉

Pclass	Sex	PassengerId	Survived	Age	SibSp	Parch	Fare
1	female	469.212766	0.968085	34.141405	0.553191	0.457447	106.125798
	male	455.729508	0.368852	39.287717	0.311475	0.278689	67.226127
2	female	443.105263	0.921053	28.748661	0.486842	0.605263	21.970121
	male	447.962963	0.157407	30.653908	0.342593	0.222222	19.741782
3	female	399.729167	0.500000	24.068493	0.895833	0.798611	16.118810
	male	455.515850	0.135447	27.372153	0.498559	0.224784	12.661633

2. 타이타닉 데이터 분석

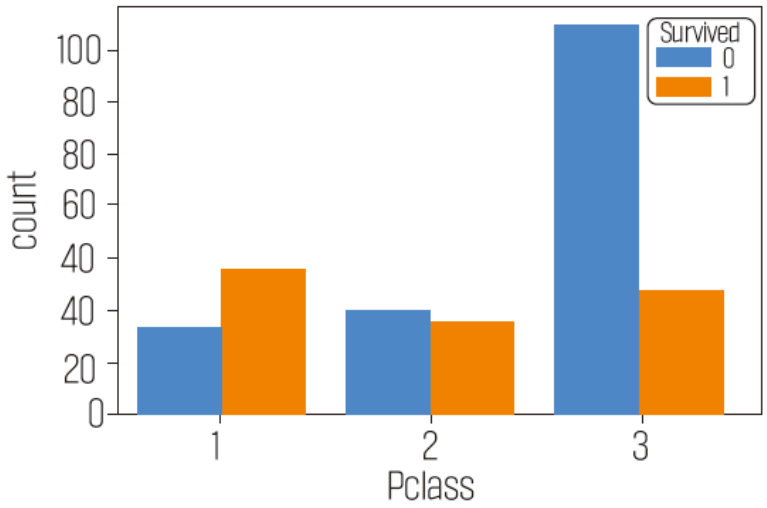
판다스의 데이터 시각화

- 좌석 등급별 생존자 확인

시본 모듈로 막대 그래프 그리기

```
1  import seaborn as sns
2  sns.countplot('Pclass',hue='Survived',data=titanic)
```

〈실행결과〉



2. 타이타닉 데이터 분석

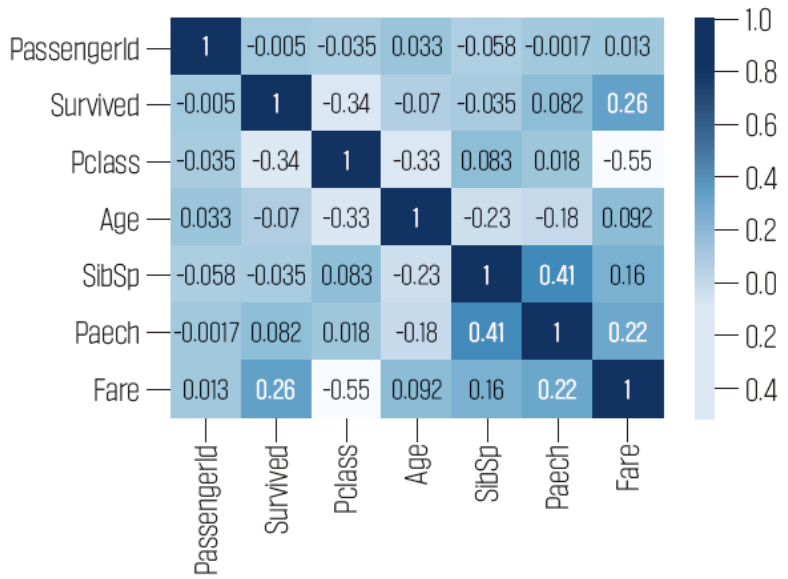
■ 판다스의 데이터 시각화

▪ 상관관계 확인

시본 모듈로 히트맵 그래프 그리기

```
1  sns.heatmap(data=titanic.corr(),annot=True,cmap='Blues')
```

〈실행결과〉



2. 타이타닉 데이터 분석

판다스의 데이터 시각화

상관 분석에 대한 이해

- 상관 분석 개념
  - 두 변수가 어떤 선형적 관계에 있는지를 분석하는 방법-두 변수는 서로 독립적이거나 상관된 관계일 수 있는데 , 두 변수의 관계의 강도를 상관관계라고 한다.
  - 단순 상관 분석-두 변수가 어느 정도 강한 관계에 있는지 측정한다.

- 상관 계수
  - 변수 간 관계의 정도(0~1)와 방향(+, -)을 하나의 수치로 요약하는 지수로 -1에서 +1 사이의 값을 가진다.
  - 상관 계수가 +면 양의 상관관계이며, 한 변수가 증가하면 다른 변수도 증가한다.
  - 상관 계수가 -면 음의 상관관계이며, 한 변수가 증가할 때 다른 변수는 감소한다.
  - 상관 계수는 데이터프레임.corr() 함수로 구한다.

- 상관 분석 결과의 시각화
  - 두 변수의 관계를 보여 주는 산점도나 히트맵을 많이 사용한다.