교육부 Ministry of Education

NRF 한국연구재단 National Research Foundation of Korea

탐색적 데이터 분석

Convergence and Open Sharing System COSS 인공지능

전남대학교 CHONNAM NATIONAL UNIVERSITY

성균관대학교 SUNG KYUN KWAN UNIVERSITY

KNU 경북대학교 KYUNGPOOK NATIONAL UNIVERSITY

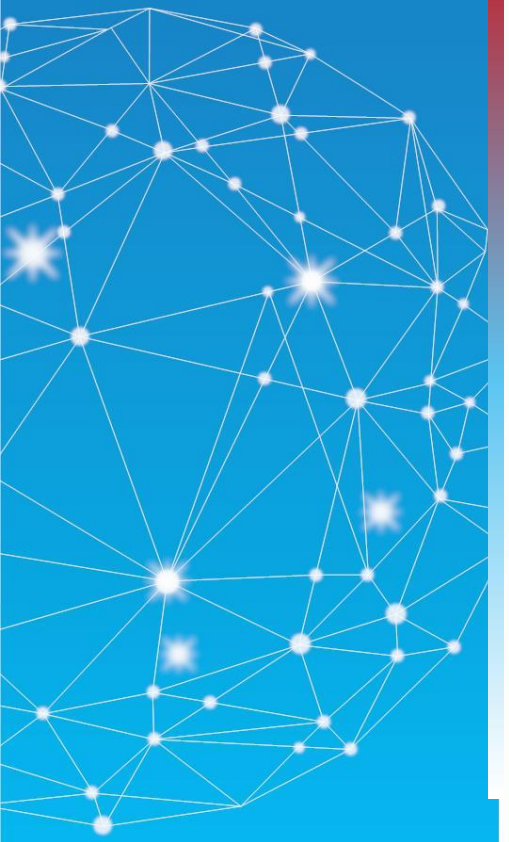
전주대학교 JEONJU UNIVERSITY

서울시립대학교 UNIVERSITY OF SEOUL

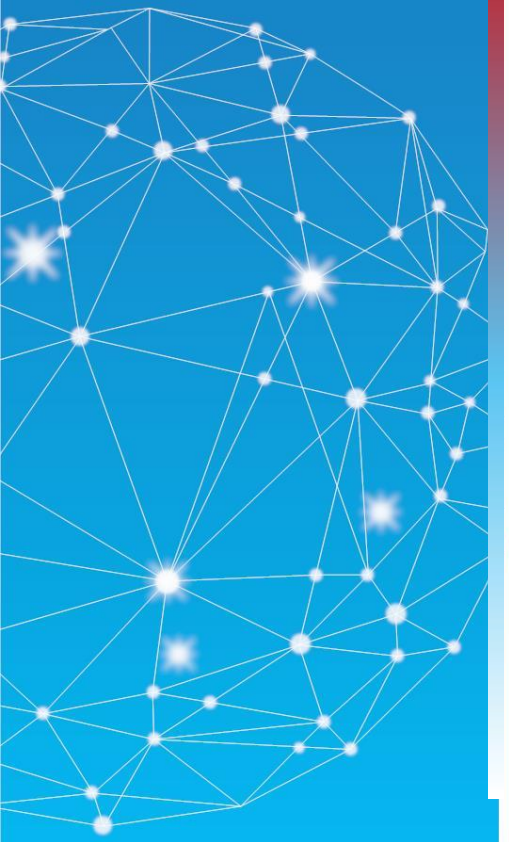
서울과학기술대학교 SEOUL NATIONAL UNIVERSITY OF SCIENCE & TECHNOLOGY

영진전문대학교 YEUNGJIN UNIVERSITY

Contents



Contents



3. 결측치와 이상치

3-1. 결측치와 이상치의 개념 이해

3-2. 해결 방법

3-3. 최빈값

3-4. 이상치 왜 문제인가?

3-5. Titanic 데이터 결측치

3-6. Titanic 데이터 결측치 해결은?

3-7. 결측치 해결 결과 확인

4. EDA 도전!

4-1. Titanic 데이터 가져오기

4-2. 전처리 과정

4-3. 행과 열 확인

4-4. 생존자 확인

4-5. 성별에 따른 생존

4-6. 선실 종류에 따른 생존

4-7. 성별에 따른 선실 별 생존

4-8. 통계량 확인

4-9. 속성 간 상관관계

1. 데이터 분석 목적 이해하기

1-1. 탐색적 데이터 분석
1-2. EDA 필요 이유
1-3. EDA를 위한 도구
1-4. EDA 유형
1-5. 다변수 시각화 유형
1-6. EDA Life Cycle
1-7. 확증적 데이터 분석

1-1 탐색적 데이터 분석

- 탐색적 자료 분석 – Exploratory Data Analysis (EDA)
  - 데이터에 대한 다양한 각도에서의 관찰 및 이해
  - 데이터의 특징, 내재하는 구조관계를 알아보기 위해 수행
- 데이터 분석의 기본 작업
  - 데이터 종류의 확인
  - 처리하기 간편한 형태로 처리
  - 데이터 간의 관계에 대한 이해
- 평균, 편차, 분포 등 통계적 수치와 시각화 기법을 활용
  - 데이터에 내재되어있는 패턴을 파악하기 위한 절차
  - 다양한 방법으로 처리하여 데이터를 검토
- 탐색적 데이터 분석을 통하여 데이터가 갖는 특성 파악

1-2 EDA 필요 이유

- 데이터 표현 현상 이해
  - 데이터의 분포 및 값에 대한 검토를 통하여 데이터에 대한 잠재적인 문제를 발견
- 데이터 수집의 방향성 결정
- 데이터 패턴 분석
  - 문제 정의 단계에서 누락 가능한 패턴 발견
- 가설에 대한 유효성
  - 기존의 가설 수정 및 새로운 가설 제시
- 정교한 데이터 분석과 AI 모델링

1-2 EDA 필요 이유

- 이론 모형을 바로 적용하기 보다는 데이터 그 자체를 잘 정비하고 기본적인 특성을 파악하여 효율적인 모델을 구성

- 자료의 구조 및 특징 파악을 위해 효과적이고 신뢰성있는 자료 요약

- 다양한 각도에서 데이터를 탐색 분석
  - 데이터에 대한 다각적인 이해
  - 잠재적 혹은 사전에 발견하지 못한 문제점, 특성을 파악
  - 가설을 검증

1-3 EDA를 위한 도구

- EDA 단계에서는 데이터 전체의 특성 파악을 우선함
  - 요약 통계
  - 데이터 간의 관계 시각화
  - 데이터 군집화
  - 회기분석
  - 상관계수 등
  - 단변량, 다변량 분석

- 복잡하고 정교한 기계학습 모델보다, 전체 데이터의 분포 및 개별 변수의 특성 파악

- 문제 해결을 위한 “가능성 검토”에 초점을 맞추고 진행

1-4 EDA 유형

- 일변량 비시각화(Univariate non-graphical)
  - 분석되는 데이터가 하나의 변수로 구성되는 가장 간단한 데이터 분석 형식
  - 단일 변수이기 때문에 원인 및 결과를 다루지는 않음
  - 일변량 분석 목적 : 데이터 설명 및 그 안에 존재하는 패턴 찾기

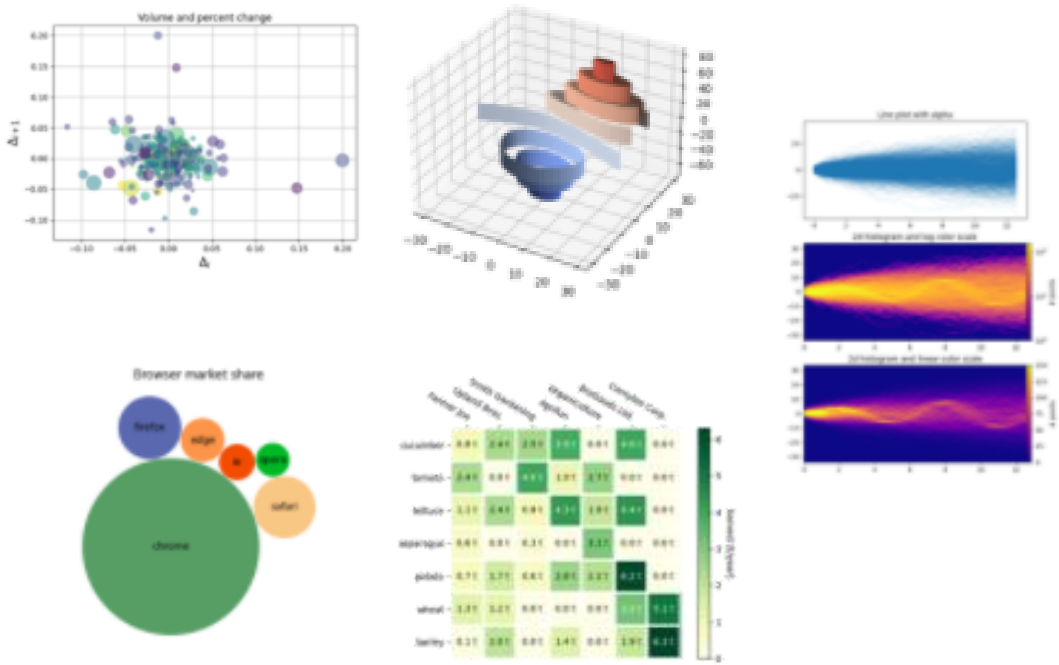
- 일변량 시각화(Univariate graphical)
  - 데이터의 전체 모습을 파악하기 위하여 시각화 기법을 활용
  - 모든 데이터의 값과 분포 시각화 적용
  - 각각의 막대가 개별 값의 범위에 대한 케이스의 빈도와 비율을 나타내는 히스토그램

- 다변량 비시각화(Multivariate nongraphical)
  - 분석 데이터가 2개 이상의 변수로 구성
  - 일반적으로 교차표 또는 통계를 통해 변수간 관계를 나타냄

- 다변량 시각화(Multivariate graphical)
  - 변수 간 관계를 표현 가능한 시각화 기법 활용

1-5 다변수 시각화 유형

- 산점도
  - 한 변수가 다른 변수의 영향을 받는 정도를 수평 및 수직 축에 데이터 포인트를 사용하여 표시
  - 점의 크기, 색상 등을 활용하여 변수간 특성 표현
- 다변량 차트
  - 데이터간의 관계, 인과관계에 대한 시각화
- 런 차트 (Run Chart)
  - 시계열 데이터 표시하는 선 차트를 포함한 그래프
  - 패턴, 추세에 따라 변화 관찰
- 버블 차트
  - 2차원 플롯에 여러 개의 버블 표시 방법
  - 전체의 각 개념이나 부분간 연관
- 히트 맵
  - 데이터의 값(혹은 관계정도)을 색상으로 시각화

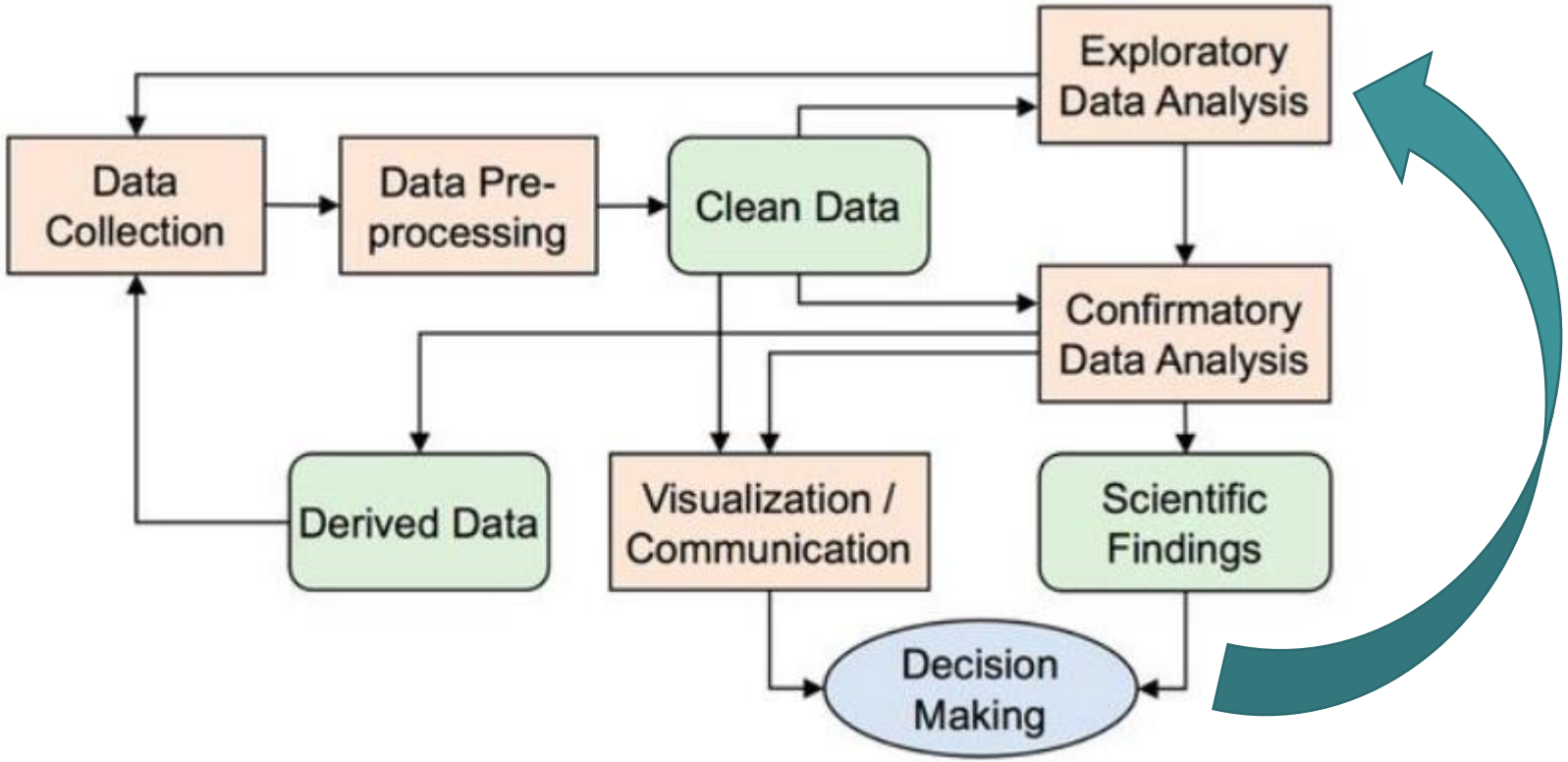


[참고] 그래프 자료 출처

https://matplotlib.org/stable/gallery/index.html

1-6 EDA Life Cycle

- 결과에 따른 데이터 생성
  - 데이터수집 -> 전처리 -> 탐색적 분석
    -> 시각화
    -> 확증적 데이터 분석
    -> 인사이트 도출



출처: https://www.mdpi.com/2220-9964/6/11/368/html 수정

1-6 EDA Life Cycle

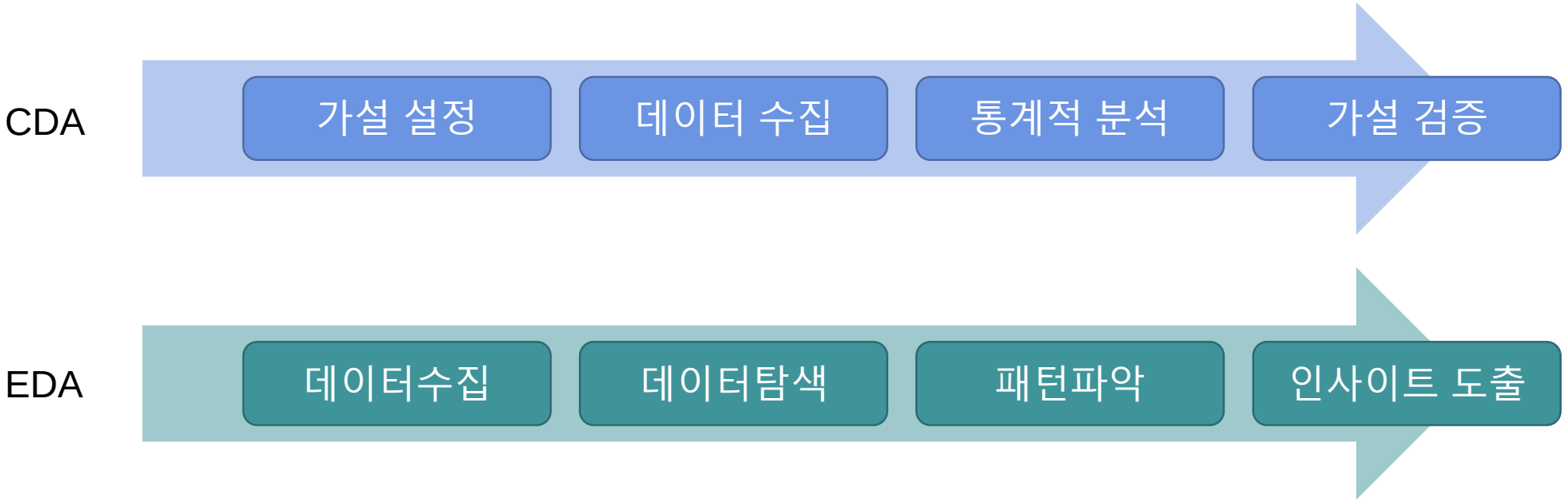
- Data pre-processing
  - outlier제거, 정규화 등 분석 전 데이터 처리

- Exploratory data analysis
  - 탐색적 데이터 분석
  - 변수-변수의 관계등 데이터 자체의 특성을 확인하기 위한 분석, 간단한 기술 통계량 계산과 다양한 그래프 활용, 모든 데이터 분석의 시작단계

- Confirmatory data analysis
  - 확증적 데이터 분석
  - 미리 설정한 가설을 확인하기 위한 분석, 추정과 검정등을 활용, 연구의 데이터 분석 방법

1-7. 확증적 데이터 분석

- 확증적 데이터 분석(Confirmatory Data Analysis)
  - 목적을 가지고 데이터를 확보하여 분석
  - 관측된 형태나 효과의 재현성 평가, 유의성 검정, 신뢰구간 검정 등 통계적 추론
  - 연역적 방법과 같이 선이론-후조사 하는 탐색방법

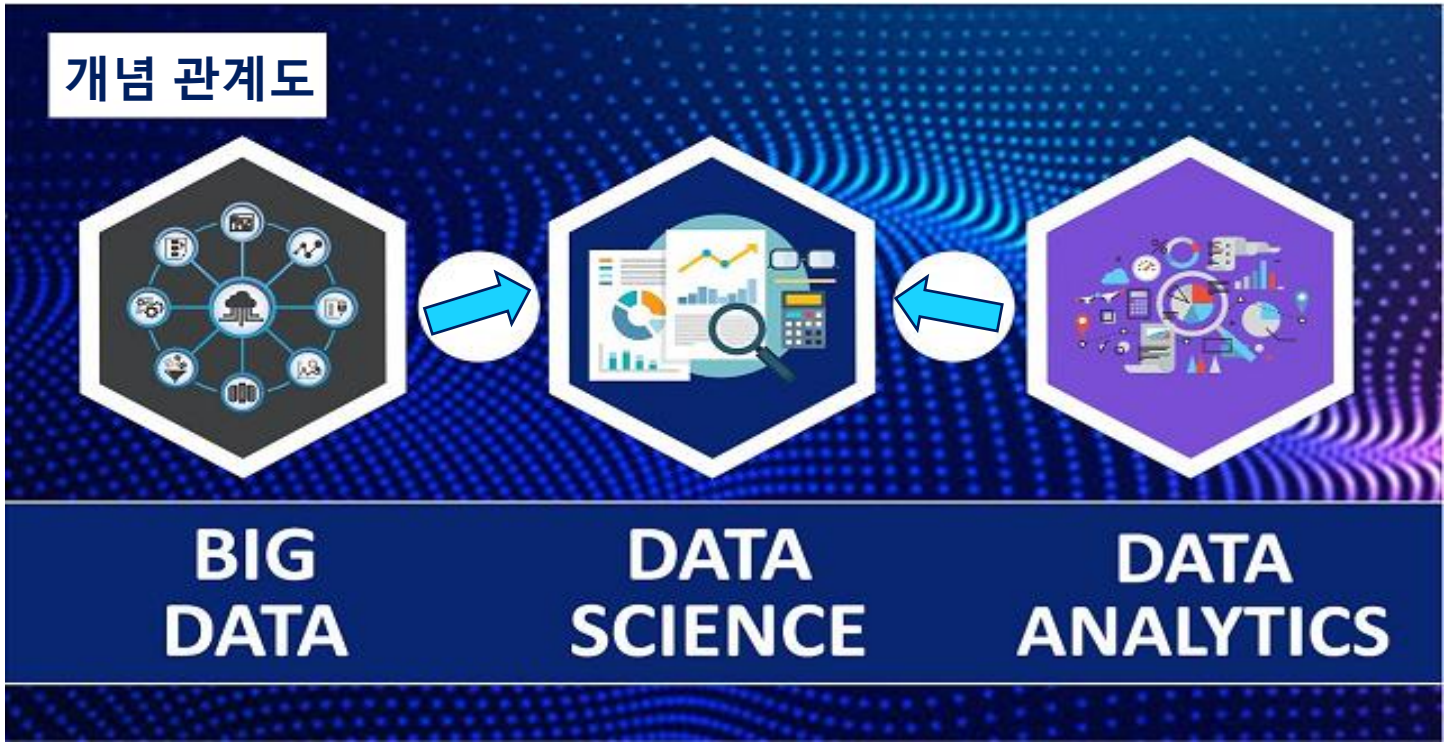


2. 데이터 처리 과정

2-1. 데이터의 관계도
2-2. 일반적 데이터 과학 프로세스
2-3. 데이터 특성
2-4. 데이터 특성을 통한 분석
2-5. 처리 과정

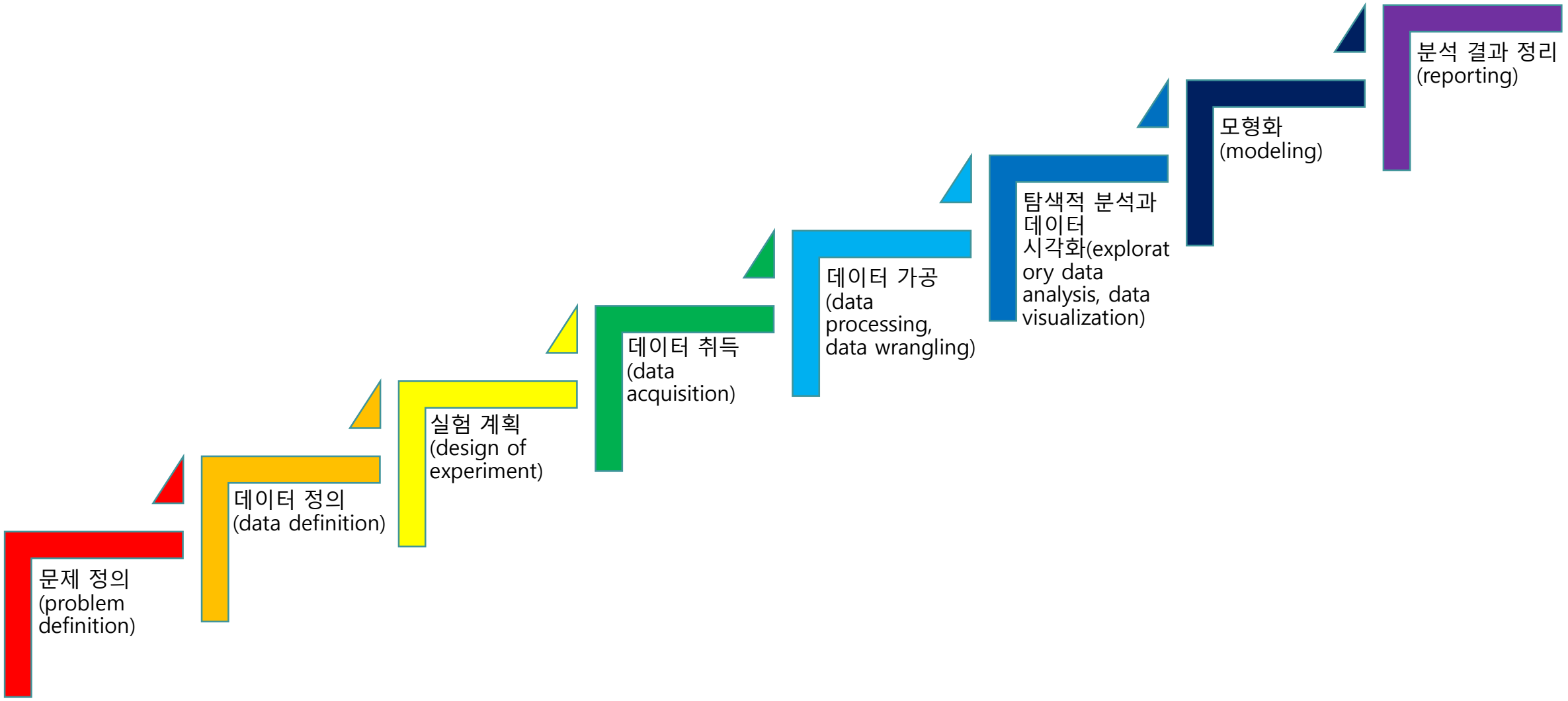
2-1 데이터의 관계도

- 데이터 관련 작업
  - 데이터의 가치를 찾는 것!
  - 단순 분류나 분석이 아닌 데이터 속에 담긴 패턴이나 미래 예측에 도움이 되는 정보를 찾는것



출처 : https://www.intellipaat.com 수정

2-2 일반적 데이터 과학 프로세스



2-2 일반적 데이터 과학 프로세스

- 문제 정의
  - 데이터를 얻기전, 우리가 해결하고 싶은 것에대한 **명확한 정의**
  - 내가 작업중인 문제를 어떻게 구성할 것인가?
  - 현실의 구체적인 문제를 명확하게 표현하고 통계, 수리적 언어로 번역

- 데이터 정의
  - 변수(variable), 지표(metric)등을 정의

- 실험 계획 (design of experiment, or sampling)
  - 어떤 처리의 효과를 알아내기위한 통제된 환경에서의 실험이 필요
  - 모집단을 대표하는 표본을 얻기 위한 표본화 (sampling)

2-2 일반적 데이터 과학 프로세스

- 데이터 취득
  - 다양한 형태, 다양한 시스템에서 저장된 원 데이터를 분석 시스템으로 가져옴

- 데이터 가공
  - 데이터를 분석하기 적당한 형태 (column: variable, row: observation)으로 가공하는 작업
  - 전처리

- 탐색적 분석 과 데이터 시각화
  - 시각화와 간단한 기초 통계량 계산을 통하여 데이터의 패턴을 발견하고 이상치를 점검하는 분석

2-2 일반적 데이터 과학 프로세스

- 모형화
  - 모수 추정, 가설 검정등의 활동과 모형분석, 예측 분석등을 포괄

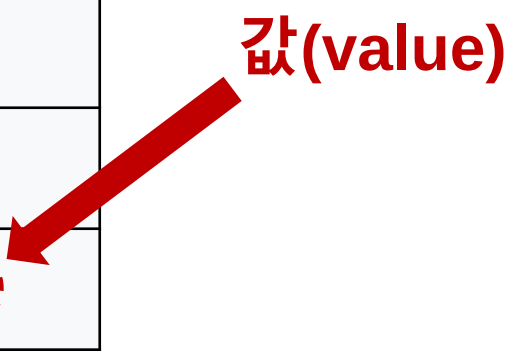
- 분석 결과 정리
  - 분석 결과를 현실적 언어로 이해하기 쉽도록 번역해내는 작업

- 순차적으로 진행되는게 이상적이지만, 실제로는 각 단계별로 왔다갔다 하면서 진행됨

2-3 데이터 특성

- Data Feature
  - 속성 (attribute) + 값 (value)
  - 데이터의 모든 측면중 모델이 사용할 “특징”값

속성 / Object	속성1	속성2	속성3
Object1	1	2	0
Object2	1	2	0
Object3	2	0	0
Object4	0	0	1



2-4 데이터 특성을 통한 분석

- 데이터 특성 분석을 시도
  - 타이타닉 데이터 분석을 통해 생존자 예측이 가능할까?

출처 : https://www.nationalgeographic.org/thisday/apr15/titanic-sinks/

2-5 처리 과정 I

- 문제: 타이타닉 데이터 분석을 통한 생존자 예측
- 1단계
  - 데이터 취득
    - ✓ https://www.kaggle.com/c/titanic/data
      - 학습 데이터: train.csv
      - 테스트 데이터: test.csv
  - 분석 도구: Python

```
# 데이터 수집
import pandas as pd
import numpy as np

train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
```

2-5 처리 과정 II

- 2단계
  - 데이터 확인 및 분석

```
1 train.info()
```

```
<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
```

```
1 test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Pclass       418 non-null    int64
 2   Name         418 non-null    object
 3   Sex          418 non-null    object
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object
 10  Embarked     418 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

2-5 처리 과정 II (계속)

- 데이터 의미 이해: 보통 데이터와 함께 제공됨
  - PassengerId - 승객 번호
  - Survived - 생존 여부 (0 = 사망, 1 = 생존)
  - Pclass - 티켓 클래스 (1 = 1등석, 2 = 2등석, 3 = 3등석)
  - Name - 승객 이름
  - Sex - 성별
  - Age - 나이
  - SibSp - 함께 탑승한 형제자매(Sibling) / 배우자(Spouse)
  - Parch - 함께 탑승한 부모님(Parent) / 아이들 의 수(Child)
  - Ticket - 티켓 번호
  - Fare - 탑승 요금
  - Cabin - 객실 번호
  - Embarked - 선착장 (C=Cherbourg, Q=Queenstown, S=Southampton)

2-5 처리 과정 Ⅲ

- 3단계
  - ➢ 데이터 가공
    - ✓ 예측을 위하여 인공지능 알고리즘 적용
    - ✓ 결측치/이상치 제거
      - 데이터 분석에 있어서 중요한 과정 중 하나
      - NaN (Not a number)
      - NA (Not available)

2-5 처리 과정 IV

- 4단계
  - ➢ 예측 준비 완료 (모델 생성)
    - ✓ 분석 방법에 따라
      - 남녀
      - 티켓 클래스
      - 가족 동승 여부
      - 나이
      - 기타 등등
    - ✓ 제공된 데이터 간의 관계 분석

3. 결측치와 이상치

3-1. 결측치와 이상치의 개념 이해
3-2. 해결 방법
3-3. 최빈값
3-4. 이상치 왜 문제인가?
3-5. Titanic 데이터 결측치
3-6. Titanic 데이터 결측치 해결은?
3-7. 결측치 해결 결과 확인

3-1 결측치와 이상치의 개념 이해

- 결측치
  - 값이 누락되어 비어있는 값
    - ✓ 수집과정의 오류등
  - 문자로는 Null이나 NaN으로 표시
    - ✓ NaN
      - Not a Number
  - 분석과정(함수호출) 등에 문제가 발생하므로 확인하여 처리

3-1 결측치와 이상치의 개념 이해

- 이상치(Outlier)
  - 값이 있지만 정상 범주에서 크게 벗어난 값
  - 보통은 오류지만 드물게 실제 데이터의 극단적인 값이 있을 수 있음
  - 분석 결과에 왜곡을 발생시키므로 제거하거나 별도의 처리가 필요
  - Boxplot을 그렸을 때 lower fence, upper fence를 벗어나는 값

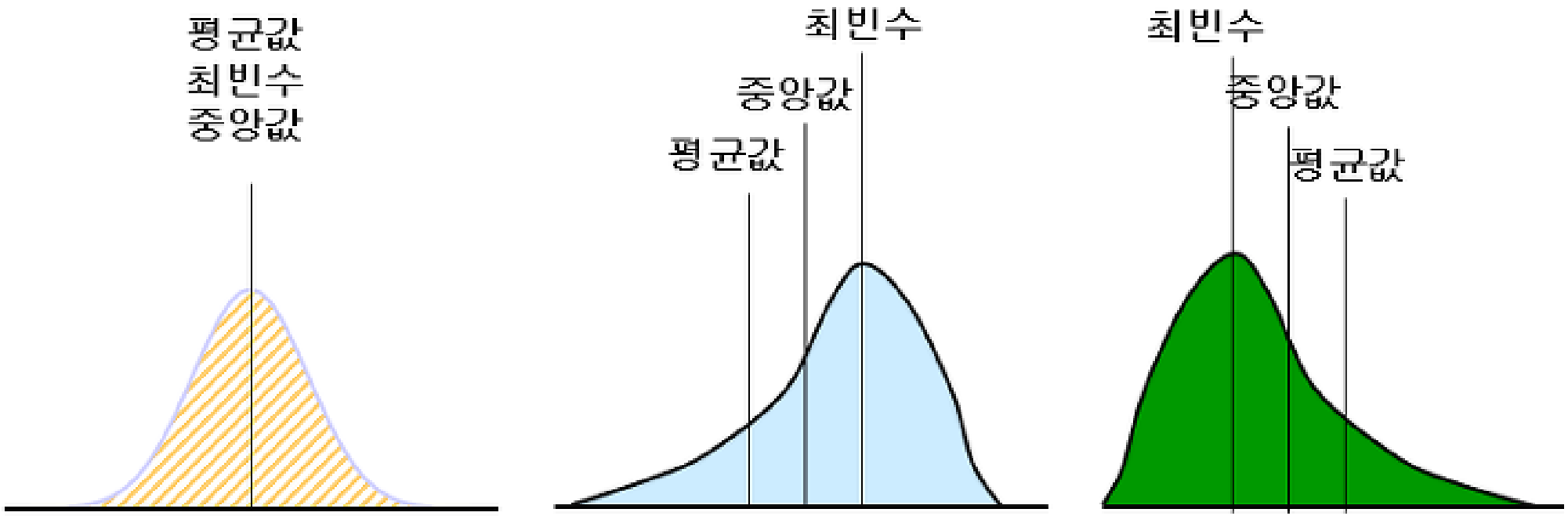
3-2 해결 방법

- 결측치 / 이상치 처리 방식
  - 데이터 삭제
    - ✓ 데이터 분석에 영향이 없어나 미비할때
  - 다른 값으로 채우는 방식
    - ✓ 최빈값, 평균값(mean), 중앙값(median) 등을 사용
  - 아래와 같다면?
    - ✓ 1번행 제거
    - ✓ 2번행 유사데이터를 활용
      - 유사그룹의 값 채우기

	first_name	last_name	age	sex	preTestScore	postTestScore
0	Jason	Miller	42.0	m	4.0	25.0
1	NaN	NaN	NaN	NaN	NaN	NaN
2	Tina	Ali	36.0	f	NaN	NaN
3	Jake	Milner	24.0	m	2.0	62.0
4	Amy	Cooze	73.0	f	3.0	70.0

3-3 최빈값

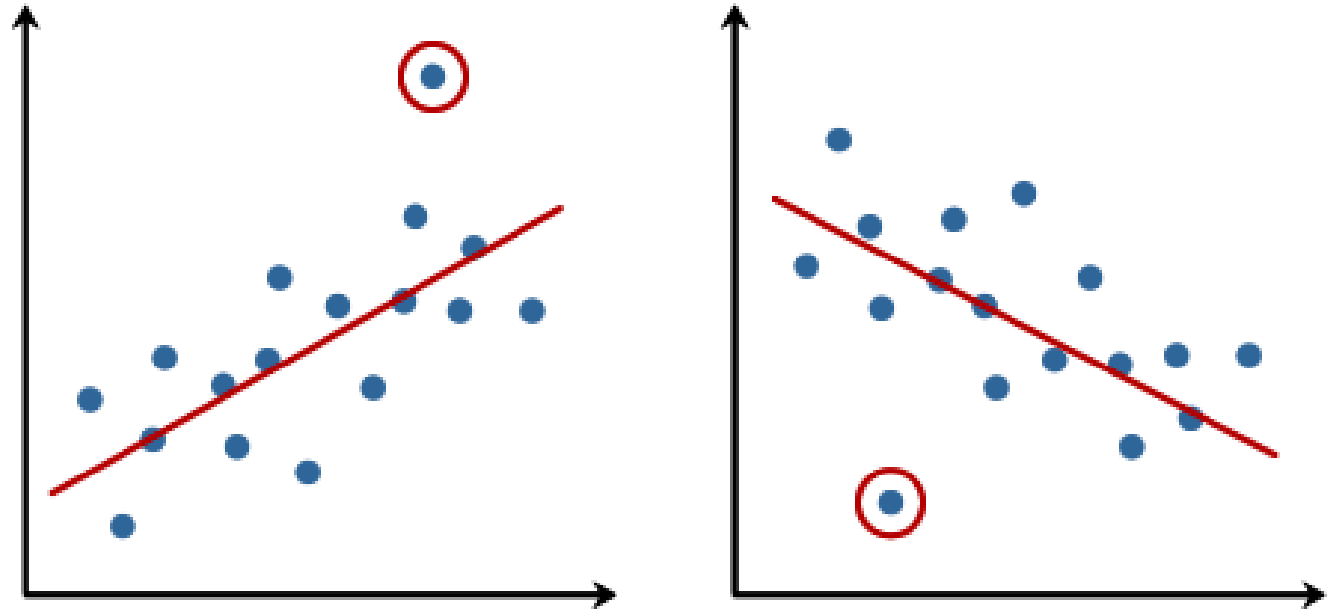
- 가장 많은 빈도수를 갖는 값
  - 데이터 분포에 따라 평균, 중앙과 다른 값을 가질 수 있음



출처 : http://www.itcomm.co.kr

3-4 이상치 왜 문제인가?

- 전반적 데이터 신뢰도에 영향을 미침
  - 분석 결과 및 모델에 왜곡을 발생시킴



3-5 Titanic 데이터 결측치

- null 값을 가진 자료 확인
  - Age와 cabin에 집중되어 있음

```
1 # train의 칼럼별 결측치 합계
2 train.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
1 # test의 칼럼별 결측치 합계
2 test.isnull().sum()
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

3-6 Titanic 데이터 결측치 해결은? (1)

- 나이 (Age)
  - 같은 성별 찾기
  - 같은 성별의 평균 나이 계산
  - 계산된 평균값으로 결측치 제거
  - 나이는 전체 분포를 따라간다고 추정

```
1 # train의 칼럼별 결측치 합계
2 train.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
1 # test의 칼럼별 결측치 합계
2 test.isnull().sum()
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

```
1 train["Age"].fillna(train.groupby("Sex")["Age"].transform("mean"), inplace=True)
2 test["Age"].fillna(test.groupby("Sex")["Age"].transform("mean"), inplace=True)
```

3-6 Titanic 에서 결측치 해결은? (2)

- Cabin
  - 객실 번호가 갖는 의미
    - ✓ 별 의미가 없는 데이터
  - 결측치를 해결할 방법 고려
  - 결측치가 지나치게 많음
  - 속성 제거

```
1 # train의 칼럼별 결측치 합계
2 train.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
1 # test의 칼럼별 결측치 합계
2 test.isnull().sum()
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

```
1 train = train.drop(['Cabin'],axis=1)
2 test  = test.drop(['Cabin'],axis=1)
```

3-6 Titanic 에서 결측치 해결은? (3)

- Embarked
  - 선착장 (C=Cherbourg, Q=Queenstown, S=Southampton)
  - 자료 확인

```
1 train.Embarked.value_counts(dropna=False)
```

```
S      644
C      168
Q       77
NaN      2
Name: Embarked, dtype: int64
```

```
1 # train의 칼럼별 결측치 합계
2 train.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
1 # test의 칼럼별 결측치 합계
2 test.isnull().sum()
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

  - 전체 데이터중 S의 비중이 70%이상
    - ✓ 가장 높은 가능성
  - 가장 많은 값 S로 결측치 해결

```
1 for dataset in [train, test]:
2     dataset['Embarked'] = dataset['Embarked'].fillna('S')
3     dataset['Embarked'] = dataset['Embarked'].astype(str)
```

3-6 Titanic 에서 결측치 해결은? (4)

- 테스트 데이터셋의 Fair 처리는?
  - 티켓의 종류에 따라 운임이 결정
  - Pclass(좌석등급)를 확인
  - 요금이 누락된 데이터의 티켓 종류
    - ✓ Pclass: 3

```
1 # train의 칼럼별 결측치 합계
2 train.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
1 # test의 칼럼별 결측치 합계
2 test.isnull().sum()
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

```
1 print (train[['Pclass', 'Fare']].groupby(['Pclass'], as_index=False).mean())
2 print(test[test["Fare"].isnull()]["Pclass"])
```

```
   Pclass       Fare
0       1  84.154687
1       2  20.662183
2       3  13.675550
152     3
Name: Pclass, dtype: int64
```

3-6 Titanic 에서 결측치 해결은? (4) (계속)

- 테스트 데이터셋의 Fair
  - Pclass가 3인 경우의 평균 운임비에 해당하는 값 적용
  - 다른 데이터를 활용한 결측치 해결

```
1 for dataset in [test]:
2     dataset['Fare'] = dataset['Fare'].fillna(13.675550)
```

3-7 결측치 해결 결과 확인

- isnull( ) 확인
  - 더 이상 결측치가 없는것을 확인

- 모든 속성에 대한 값이 주어졌다면
  - 결측치 해결 완료

```
1 test.isnull().sum()
```

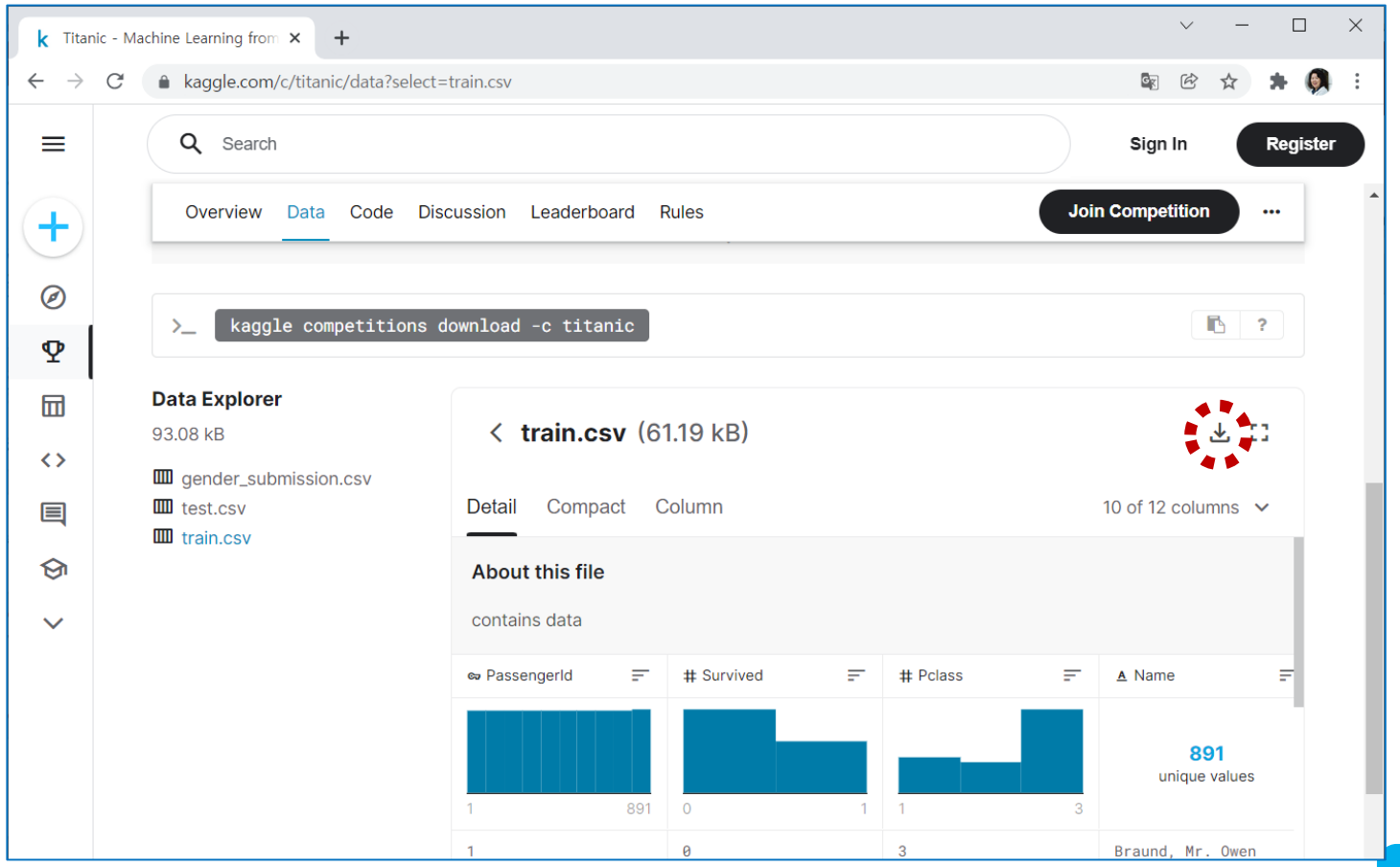
```
PassengerId    0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

4. EDA 도전!

4-1. Titanic 데이터 가져오기
4-2. 전처리 과정
4-3. 행과 열 확인
4-4. 생존자 확인
4-5. 성별에 따른 생존
4-6. 선실 종류에 따른 생존
4-7. 성별에 따른 선실 별 생존
4-8. 통계량 확인
4-9. 속성 간 상관관계

4-1 Titanic 데이터 가져오기

- Kaggle
  - 데이터 분석 및 머신러닝 학습/경진 플랫폼
  - https://www.kaggle.com/c/titanic/data?select=train.csv
  - 다운로드하여 사용
  - 가장 심플한형태의 EDA실습



4-2 전처리 과정

- 데이터 알아보기
  - 다운로드 받은 데이터의 값을 파이썬에서 불러와서 확인

```
# Titanic Data Set 불러오기
# 출처: https://www.kaggle.com/c/titanic/data

import pandas as pd

data = pd.read_csv('train.csv')
data.head()
```

	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

4-3 행과 열 확인

- 12개의 속성
  - 12개의 열로 적용
- 891개의 행
  - 891건의 데이터
- 다시말해, 891명의 사람에 대한 12가지 속성 데이터
- Data Type (dtype)
  - 구성된 자료의 자료형
    - ✓ object

```
1  # data에 어떤 값들이 존재하는지와 그 크기를 알아보자
2  print('Columns: ', end ='')
3  print(data.columns)
4  print('Data size: ', end = '')
5  print(data.shape)
```

```
Columns: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
Data size: (891, 12)
```

4-4 생존자 확인

- Survived
  - 0: 사망
  - 1: 생존

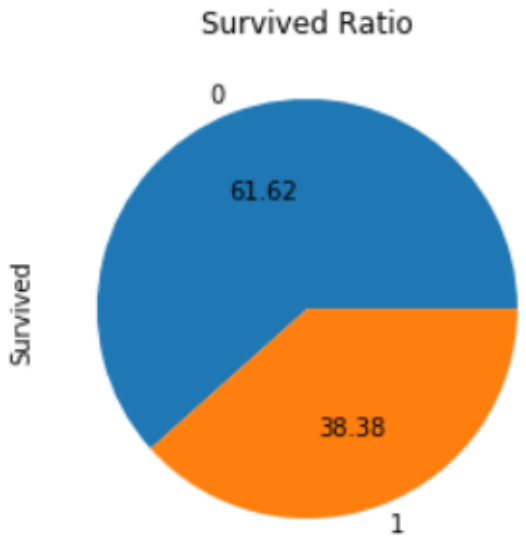
- 342명 생존
  - 생존률 38.38%

```
#사망자와 생존자의 비율을 Survived를 통해 알아보자
#Survived가 0이라는 것은 해당 승객이 사망했음을, 1이라는 것은 생존했음을 뜻한다.

data['Survived'].value_counts()
```

```
0    549
1    342
Name: Survived, dtype: int64
```

```
# 값을 확인했다면, 간단하게 시각화해보자.
ax = data['Survived'].value_counts().plot.pie(autopct ='%.2f')
ax.set_title('Survived Ratio')
```

```
Text(0.5, 1.0, 'Survived Ratio')
```



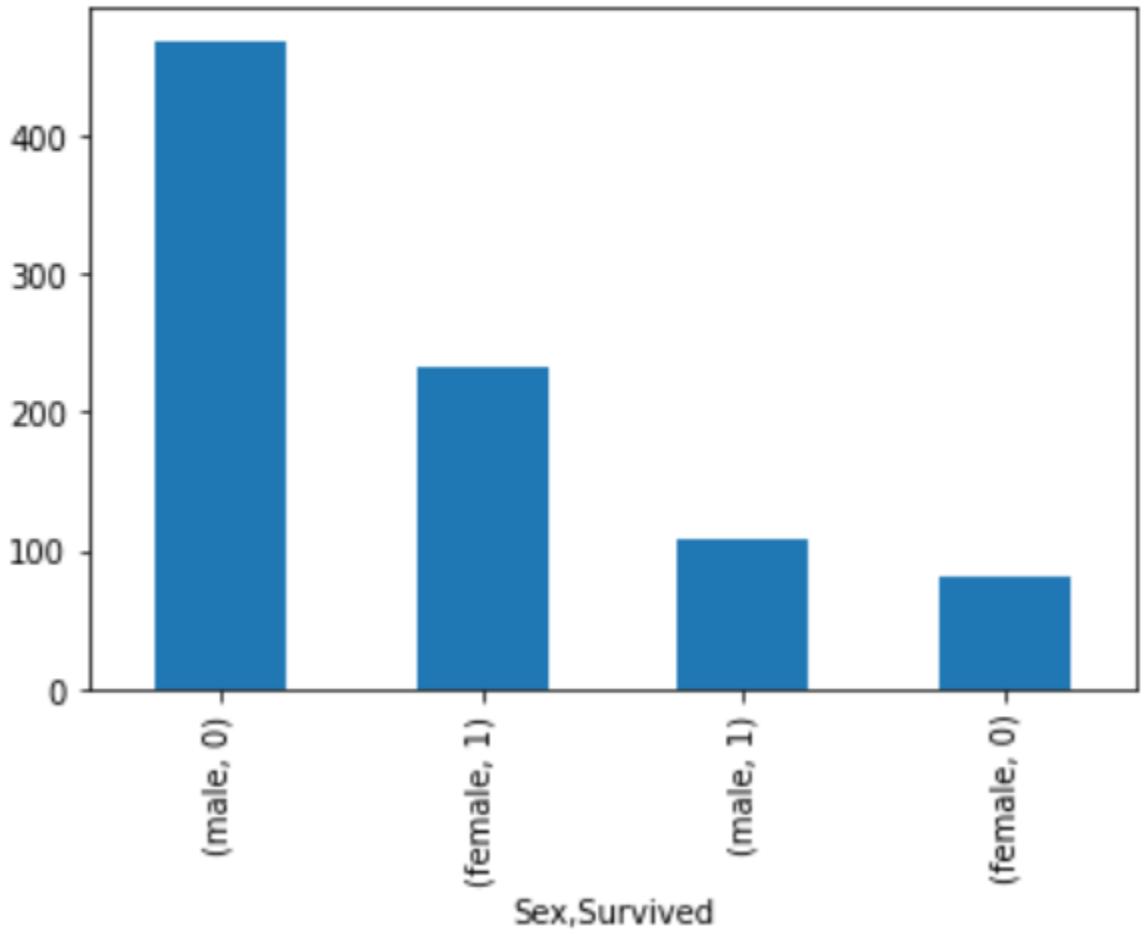
4-5 성별에 따른 생존

- 남성은 사망자(0)가 더 많음
- 여성은 생존자(1)가 더 많음

- 여성 생존율이 높음

```
1  # 성별에 따른 생존률은 어떨까?
2  ax = data[['Sex', 'Survived']].value_counts().plot.bar()
3  ax
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7feb1e938290>
```



4-6 선실 종류에 따른 생존

- 선실 등급별 탑승자와 생존률을 확인
  - Pclass

- 높은등급 선실 탑승자의 생존률이 더 높음

```
1  # 다음으로 선실의 등급인 Pclass도 분석해보자
2  # 1은 1등실을, 2는 2등실을, 3은 3등실을 나타낸다.
3
4  data['Pclass'].value_counts()
```

```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```

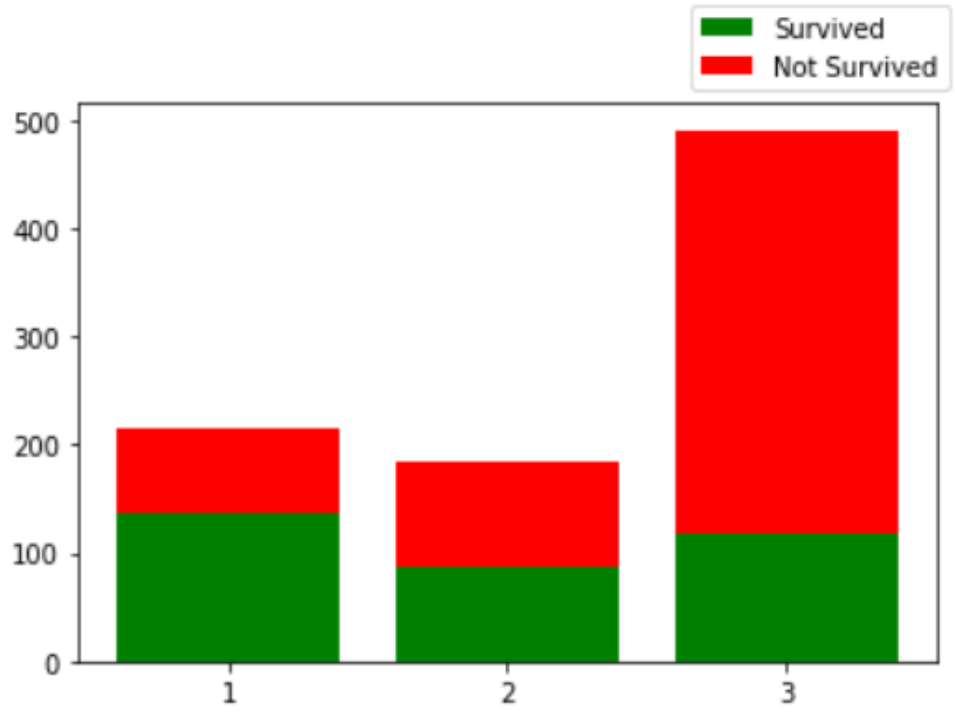
```
1  # 선실 등급 별 생존자 수도 알아보자
2  pd.crosstab(data.Pclass, data.Survived)
```

Survived	0	1
Pclass		
1	80	136
2	97	87
3	372	119

4-6 선실 종류에 따른 생존

- 선실 등급별 생존자/사망자 비율의 시각화
  - Stacked bar chart 활용
  - 등급이 낮을수록 사망률이 높음

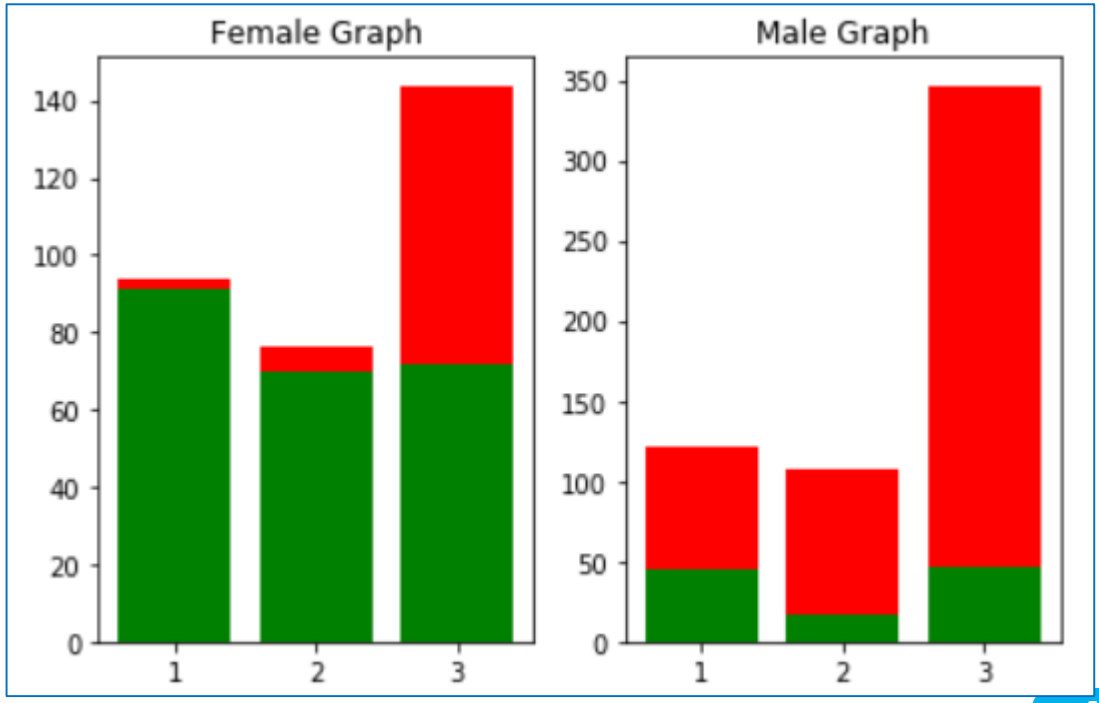
```python
import numpy as np
import matplotlib.pyplot as plt

survived = [136, 87, 119]
not_survived = [80, 97, 372]

Pclass = ["1","2","3"]

plt.bar(Pclass, survived, color="green", label="Survived")
plt.bar(Pclass, not_survived, color="red",
        bottom=np.array(survived), label="Not Survived")

plt.legend(loc="lower left", bbox_to_anchor=(0.7,1.0))
plt.show()
```



4-7 성별에 따른 선실 별 생존

- subplot( )
  - 성별로 구분하여 선실별 생존률을 확인
  - 1~2등급 선실을 사용한 여성의 생존률이 매우 높음

```python
import numpy as np
import matplotlib.pyplot as plt

Pclass = ["1","2","3"]
f_survived = [91, 70, 72]
f_not_survived = [3, 6, 72]

m_survived = [45, 17, 47]
m_not_survived = [77, 91, 300]

plt.subplot(1, 2, 1)                # nrows=1, ncols=2, index=1
plt.bar(Pclass, f_survived, color="green", label="Survived")
plt.bar(Pclass, f_not_survived, color="red",
        bottom=np.array(f_survived), label="Not Survived")
plt.title('Female Graph')

plt.subplot(1, 2, 2)                # nrows=1, ncols=2, index=2
plt.bar(Pclass, m_survived, color="green", label="Survived")
plt.bar(Pclass, m_not_survived, color="red",
        bottom=np.array(m_survived), label="Not Survived")
plt.title('Male Graph')

plt.tight_layout()
plt.show()
```



4-8 통계량 확인

- describe( )
  - 각 컬럼의 통계 요약자료 확인

```
1  # 다음으로 각 column의 통계량을 확인해보자
2  # describe 함수는 수치형 데이터에 대한 요약만을 제공하므로,
3  # Name과 같은 column은 빠져있는 것을 확인할 수 있다.
4
5  data.describe()
```

	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

4-9 속성 간 상관관계

- corr( ) : 상관계수(correlation)를 구하는 함수
- 높은 상관관계
  - Fare & Pclass
  - SibSp & Parch
  - Pclass & Survived

```
1  # 이번에는 각 Column 간 상관계수를 확인해보자
2  data.corr()
3
4  # Pclass-Survived, 그리고 Fare-Pclass 간 큰 상관계수가 나옴
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

교육부 Ministry of Education

NRF 한국연구재단 National Research Foundation of Korea

thank you

본 과제(결과물)는 교육부와 한국연구재단의 재원으로 지원을 받아 수행된
디지털신기술인재양성 혁신공유대학사업의 연구결과입니다.

Convergence and Open Sharing System COSS 인공지능

전남대학교 CHONNAM NATIONAL UNIVERSITY

성균관대학교 SUNG KYUN KWAN UNIVERSITY

KNU 경북대학교 KYUNGPOOK NATIONAL UNIVERSITY

전주대학교 JEONJU UNIVERSITY

서울시립대학교 UNIVERSITY OF SEOUL

서울과학기술대학교 SEOUL NATIONAL UNIVERSITY OF SCIENCE &TECHNOLOGY

영진전문대학교 YEUNGJIN UNIVERSITY