# 강의 : 빅데이터실습

# 교수 : 이희진

# 강의 교재

- 저자 : 김영우
- 출판사 : 이지스 퍼블리싱


# 평가

| 중간 고사 | 기말고사 | 출석 | 기타 |
|---|---|---|---|
| 30% | 40% | 20% | 10% |

# 교과목 개요

본 교과목에서는 유용한 정보를 발견하고 결론을 유추하거나 의사결정을 돕기 위해 데이터를 조사, 정제, 변환환, 모델링하는 데이터 분석 과정에 대해 살펴 본다. 데이터 분석을 위해서는 파이썬 프로그래밍 언어를 사용한다. 특히 파이썬 데이터 분석을 위한 전문 패키지인 판다

스(Pandas)를 활용한 데이터 처리와 분석 기능을 학습한다. 또한 데이터 분석 역량을 향상하기 위해 실제 공공 데이터를 활용하여 분석해 본다. 아울러 다양한 데이터 분석을 위해 텍스트 마이닝, 지도 시각화, 통계 분석 기법을 활용한 가설 검증, 머신러닝을 이용한 예측 분석에 대해서도 학습한다.

# 강의 계획

- 1주차 : 파이썬 기초와 데이터 분석 준비
  - 아나콘다 개발환경 이해
  - Jupyter Notebook 활용
  - 파이썬 프로그래밍 기초
- 2주차 : 데이터 프레임의 이해
  - 데이터 프레임 이해하기
  - 데이터 프레임 만들기
  - 외부 데이터를 이용하여 데이터 프레임 만들기
- 3주차 : 데이터 파악하기, 다루기 쉽게 수정하기
  - 데이터 특징 파악하기
  - 변수명 바꾸기
  - 파생변수 만들기
- 4주차 : 데이터 가공하기
  - 데이터 전처리
  - 조건에 맞는 데이터 추출하기
  - 필요한 변수만 추출하기
  - 순서대로 정렬하기
  - 파생변수 추가하기
  - 집단별 요약하기
  - 데이터 합치기
- 5주차 : 데이터 정제
  - 빠진 데이터(결측치) 처리하기
  - 이상한 데이터(이상치) 처리하기
- 6주차 : 그래프 만들기
  - 파이썬으로 만들 수 있는 그래프 살펴보기
  - 산점도 - 변수간 관계 표현하기
  - 막대 그래프 - 집단간 차이 표현하기
  - 선 그래프 - 시간에 따라 달라지는 데이터 표현하기
  - 상자 그래프 - 짐단간 분포 차이 표현하기
- 7주차 : 1주 ~ 6주 학습 내용 정리
- 8주차 : 중간고사
- 9주차 : 데이터 분석 프로젝트 (한국 복지패널 데이터 분석)
  - 한국복지패널 데이터 분석 준비하기
  - 성별에 따른 원급 차이 분석
  - 나이와 월급 관계
  - 연령대에 따른 원급 차이 분석하기
- 10주차 : 데이터 분석 프로젝트 (한국 복지패널 데이터 분석)
  - 연령대 및 성별 월급 차이 분석
  - 직업별 월급 차이 분석하기

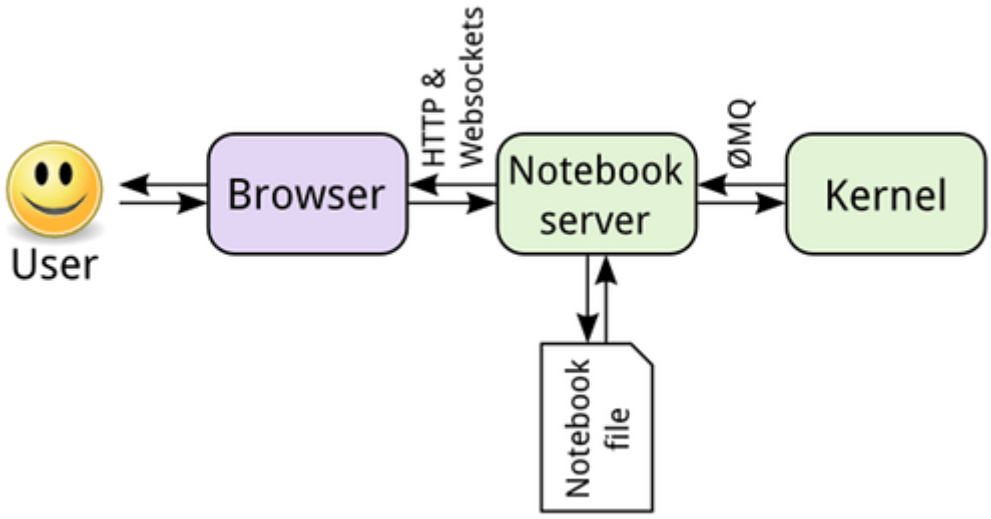
    - 성별 직업 빈도 분석
    - 종교 유무에 따른 이혼율
    - 지역별 연령대 비율
- 11주차 : 텍스트 마이닝
    - 대통령 연설문 텍스트 마이닝
    - 기사 댓글 텍스트 마이닝
- 12주차 : 지도 시각화
    - 시군별 인구 단계 구분도 만들기
    - 서울시 동별 외국인 인구 단계 구분도 만들기
- 13주차 : 통계 분석을 이용한 가설 검증
    - 가설검증 이해하기
    - t 검증 - 두 집단간 평균 비교하기
    - 상관 분석 - 두 변수의 관계 분석하기
- 14주차 : 머신러닝을 이용한 예측 분석
    - 머신러닝 모델 알아보기
    - 소득 예측 모델 만들기
- 15주차 : 기말 평가

# Anaconda

1. Anaconda는 수학이나 과학, 데이터 분석, 기계학습 등에 필요한 파이썬 패키지를 포함하고 있는 파이썬 배포판
2. 아나콘다 다운로드 페이지 (https://www.anaconda.com/products/distribution)에서 설치 가능
3. 2020년 부터 개인 이용자, 대학, 비영리단체, 200인 미만 중소기업에게만 무료이고 정부 및 200인 이상의 기업에게는 유료


# Jupyter Notebook 기본 사용법

---




## 웹 브라우저에서 파이썬 코드를 실행할 수 있는 IDE(Integrated Development Environment, 통합개발환경)

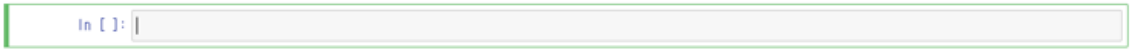
1. cell 단위로 실행 및 실행 결과 확인 가능
2. 각 cell은 커맨드 모드(Command Mode), 에디트 모드(Edit Mode)가 있음
   - [ Enter ] : 커맨드 모드 -> 에디트 모드
   - [ ESC ] : 에디트 모드 -> 커맨드 모드

In [ ]:
```
print('[> Run]')
print('[Insert]-[Insert Cell Below]')
```

---

## 에디트 모드(Edit Mode) : cell에 코드나 Markdown을 입력하거나 수정할 수 있는 상태

- cell을 마우스로 클릭하거나 enter를 누르면 에디트 모드가 됨
- [Sift + Enter] 현재 cell을 실행하고 다음 cell이 없으면 아래에 새로운 cell 추가, 다음 cell이 있으면 이동
- [Ctrl + Enter] 현재 cell을 실행하고 새로운 cell은 만들지 않음
- [ALT + Enter] 현재 cell을 실행하고 아래에 새로운 cell 추가



In [ ]:
```
# 현재 cell을 실행하고 다음 cell 이동, 맨 마지막인 경우엔 추가 하면서 이동
# 실행 후 라인 실행 번호 변화에 주목
print('[Shift + Enter] : 현재 cell을 실행하고 다음 cell 이동, 맨 마지막인 경우엔 추
```

In [ ]:
```
# 현재 cell을 실행
print('[Ctrl + Enter] : 현재 cell을 실행')
```

In [ ]:
```
# 현재 cell을 실행하고 다음 cell 추가
print('[ALT + Enter] : 현재 cell을 실행하고 다음 cell 추가')
```

---

## 커맨드 모드(Command Mode) : cell을 편집할 수 있는 상태

- 편집 모드에서 ESC 키를 누르면 커맨드 모드가 됨
- 상하 화살표 방향 키를 이용하여 다른 cell을 선택할 수 있음
- [ a ] cell 위에 새로운 cell 추가
- [ b ] cell 아래에 cell 추가
- [ c ] cell 복사
- [ v ] cell 아래에 cell 붙여 놓기
- [ x ] cell 잘라내기
- [ dd ] cell 삭제
- [Shift + L] Cell Code에 라인 번호를 부여 [View-Toggle Line Numbers]

In [ ]:

In [ ]:
```
# 파일 이름 변경 (저장된 거 확인, 파일명 변경 확인)
# 셀 소개 (실행 단위)
#  - 파란색: 선택 모드
#  - 녹색: 편집 모드
```

In [ ]:
```
print('Hello World')
# 실행 방법
# - shift + Enter
```

In [ ]:
```
3+5
# - ctrl + Enter (셀 생성 X)
```

In [ ]:
```
# 셀 추가/ 삭제
#
```

In [19]:
```
# 셀 안에 여러 줄 가능
a = 3
print(a)
```

```
3
```

In [ ]:
```
# b라는 변수가 없어서 에러
print(a+b)
```

In [20]:
```
b = 10
# 를 정의하고 나서, 다시 실행
# '실행번호'의 의미 확인
```

In [ ]:
```
import pandas as pd
pd.__version__
```

In [ ]:
```
import matplotlib
matplotlib.__version__
```

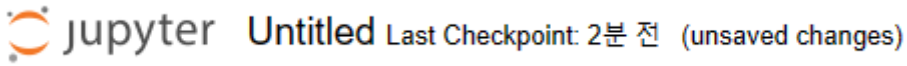
In [ ]:
```
print('error')
print(2
print(3)
```

In [ ]:
```
# Cell Code에 라인 번호를 부여
# 실행 오류시 디버깅 용이
print('[Shift + L] Cell Code에 라인 번호를 부여')
```

---

## Notebook 파일 이름 바꾸기, 저장하기

- 작성한 Notebook 파일 이름을 **Untitle**에서 **week01**로 변경




- [CTRL + S]를 눌러 NoteBook 저장
- Notebook 파일의 확장자는 **.ipynb**
- 웹브라우저 Notebook Tab의 [x]를 눌러 노트북 종료

# Python 코딩을 위한 Jupyter Notebook 기능

1. 코드를 cell 단위로 작성 및 실행
   - cell의 순서와 관계없이 실행할 수 있으며, [ ]로 실행 번호를 표시해 줌
2. 그래프나 표를 실시간으로 확인 가능
3. HTML. PDF 파일로 저장 가능

In [ ]:
```
# random.randint(a, b) : a, b 사이의 랜덤한 정수 값 반환( a, b 포함 )
import random
print(random.randint(1, 50))
```

In [ ]:
```
# random.[Tab]
# random에서 사용가능한 함수 목록 열람
```

In [ ]:
```
# random.randint의 사용법
random.randint?
```

In [ ]:
```
# random.randint의 사용법(함수 정의)
random.randint??
```

In [ ]:
```
# 주석
print(1)
print(2)
```

In [ ]:
```
print(1)
print(2)
print('CTRL + /')
```

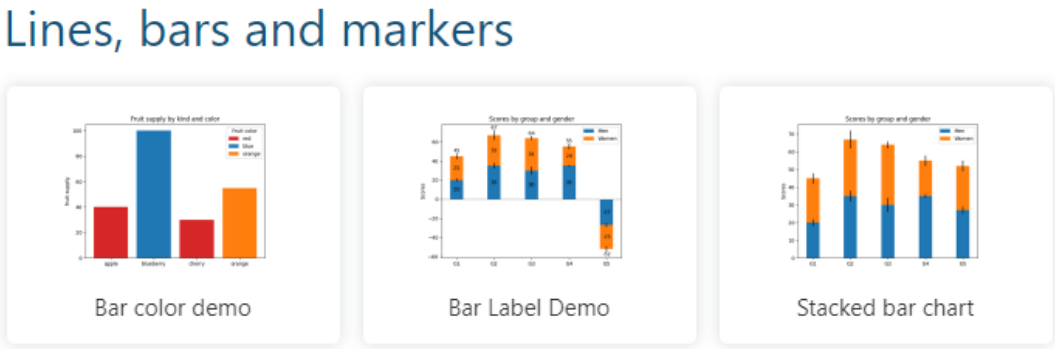
In [ ]:
```
# cell이 실행 중인 경우는 [*]로 표시됨
# cell 실행 중 중단할 경우, 커멘드 모드에서 [ ii ] 또는 [kernel]-[interrupt]
import time
for i in range(100):
    print(i)
    time.sleep(1)
```

---

## 시각화 예제 (Matplotlib) - 그래프

- Matplotlib을 이용하여 다양한 그래프를 그릴 수 있음

- [Examples]에서 가져와 그려봄




In [ ]:
```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

fruits = ['apple', 'blueberry', 'cherry', 'orange']
counts = [40, 100, 30, 55]
bar_labels = ['red', 'blue', '_red', 'orange']
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']

ax.bar(fruits, counts, label=bar_labels, color=bar_colors)

ax.set_ylabel('fruit supply')
ax.set_title('Fruit supply by kind and color')
ax.legend(title='Fruit color')

plt.show()
```

---

## 시각화 예제 (Panads) - 테이블

- Pandas을 이용하여 다양한 표를 그릴 수 있음
- [Documentation]-[User Guide]-[Table Visualization]에서 예제 가져옴

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib as mpl

df = pd.DataFrame([[38.0, 2.0, 18.0, 22.0, 21, np.nan],[19, 439, 6, 452, 226,232]],
                  index=pd.Index(['Tumour (Positive)', 'Non-Tumour (Negative)'], nam
                  columns=pd.MultiIndex.from_product([['Decision Tree', 'Regression'
df.style
```

In [ ]:
```
np.random.seed(0)
df2 = pd.DataFrame(np.random.randn(10,4), columns=['A','B','C','D'])
df2.style
```

In [ ]:
```
def style_negative(v, props=''):
    return props if v < 0 else None
s2 = df2.style.applymap(style_negative, props='color:red;')W
              .applymap(lambda v: 'opacity: 20%;' if (v < 0.3) and (v > -0.3) else
s2
```

---

## HTML, PDF 저장

- 작업한 Notebook을 HTML이나 PDF로 저장한 후 배포 가능

In [ ]:
```
# [File]-[Download as]
# [File] - [Print]
```

# 마크다운 (Markdown) 사용법

## 쉽게 문서를 작성하기 위한 마크업 언어

1. 커맨드 모드에서 [ M ] : Markdown을 편집할 수 있는 cell로 전환
2. 커맨드 모드에서 [ y ] : code를 편집할 수 있는 cell로 전환

마크다운 (Markdown)으로 다음과 같은 문서 작성이 가능하다

- **굵고**, *기울임* 문자 표현이 가능
- 인터넷 링크 연결도 가능 : 동양미래대학교
- 그림을 넣을 수도 있음
- 표를 만들 수 있음

| 학번 | 이름 | 학과 |
|---|---|---|
| **2023001** | 김동양 | 인공지능소프트웨어학과 |
| *2023002* | 이미래 | 컴퓨터소프트웨어공학과 |
| 2023003 | 박대학 | 컴퓨터정보공학과 |

- 마크다운 작성법을 자세히 알고 싶으면 마크다운 참조
- 교재 13장 p.323 참조

# 파이썬 진단

## https://forms.gle/wnauAJyRxk1CxCEo6