

2025학년도 2학기 출석수업

# 파이썬과 R

한국방송통신대학교 통계·데이터과학과  
안 형 미

# 출석수업 안내

- 강의자료 : 안형미교수 홈페이지 자료실
- 질문 : 안형미교수 홈페이지 상담게시판 - 파이썬과 R
- 강의범위 : 교재 1~5장
  
- 출석수업 과제물 : 출석 수업 종료 후 안내

# 학습목표

- 파이썬과 R을 설치 및 구동할 수 있다.
- 파이썬과 R을 자료 구조 및 입·출력을 이해하고 프로그래밍할 수 있다.
- 수업 내용을 바탕으로 과제를 수행할 수 있다.

# 제 1장 파이썬과 R의 설치와 구동

파이썬과 아나콘다 및 구글 코랩, R과 Rstudio

# 파이썬과 R의 설치

구분	파이썬	R
홈페이지	<a href="http://www.python.org">http://www.python.org</a>	<a href="http://r-project.org">http://r-project.org</a>
최신버전 (2025.09)	python-3.13.7	R-4.5.1
IDE	Anaconda(Spyder 포함), Jupyter Notebook 등	RStudio, Jupyter Notebook 등
장단점	설치 쉬우나 용량이 큼	가볍고 직관적이나 확장성 제한

IDE (Integrated Development Environment) 통합 개발 환경

# R과 Rstudio 설치

- R의 설치
  - <http://r-project.org>
- Rstudio 설치
  - <https://posit.co/downloads/>
  - 2022년 Rstudio ⇨ Posit 회사명 변경
  - 교재의 <http://www.rstudio.com> 접속 후 <FREE & OPEN SOURCE>  
— <DOWNLOAD Rstudio →>

# 파이썬 (아나콘다 설치)

구분	기본 설치	아나콘다	구글 코랩
방식	<a href="http://www.python.org">http://www.python.org</a>	<a href="http://www.anaconda.com">http://www.anaconda.com</a>	<a href="http://colab.google.com">http://colab.google.com</a>
장점	가볍고 최신 버전	패키지 포함, 편리	설치 불필요, GPU 지원
단점	패키지 관리 불편	용량 큼, 업데이트 지연	인터넷 필수, 세션 제한

# 파이썬의 IDE

- Spyder와 Jupyter Notebook의 차이: .py vs .ipynb

구분	Spyder	Jupyter Notebook
파일 형식	.py (파이썬 스크립트 파일)	.ipynb (노트북 파일)
실행 방식	전체 코드 실행 (스크립트 단위)	셀 단위 실행 (코드+결과+설명 함께)
특징	IDE 환경, 디버깅/패키지 관리 용이	코드 + 그래프 + Markdown 문서화 가능
저장 용도	프로그램 작성, 모듈 개발	실험 기록, 교육, 데이터 분석 보고서
공유성	코드 중심 공유	분석 과정 시각적으로 공유



# 연습문제

- <http://www.r-project.org/>를 접속하여 R을 설치하고 R 명령 프롬프트에서  $1+1$ 을 계산하시오.
- <http://rstudio.com/>에 접속하여 Rstudio를 설치하고 R 명령 프롬프트에서  $1+1$ 을 계산하시오.

# 연습문제

- <http://www.anaconda.com>에 접속하여 아나콘다 설치
  - 설치한 아나콘다의 Jupyter Notebook에서  $1+1$  계산하시오.
  - 설치한 아나콘다의 Anaconda Prompt를 사용하여 파이썬을 실행하고  $1+1$ 을 계산하시오.
  - 설치한 Spyder의 오른쪽 아래의 창에서  $1+1$ 을 계산하시오
  - 설치한 Spyder의 왼쪽 창에서  $1+1$ 을 계산하시오.
    - (인쇄를 위해 `print(1+1)`을 명령하고 실행버튼을 클릭)

# 제 2장 파이썬과 R의 기본 사용법

변수명, 명령어 파일, 자료의 종류, 날짜와 시간

# 기본연산자

구분	파이썬	R
할당	= , :=	<- , =
사칙연산	+, -, *, /	+, -, *, /
거듭제곱	**	^
나머지	%	%%
몫(정수 나눗셈)	//	/%
비교(같음, 다름)	==, !=	==, !=
논리 AND	and	& (벡터), && (스칼라)
논리 OR	or	,
부정	not	!

# 변수명

구분	파이썬	R
사용 가능 (공통)	문자로 시작, 숫자 붙임 가능 대소문자 구분	문자로 시작, 숫자 붙임 가능 대소문자 구분
사용 불가 (공통)	숫자로 시작, 공백 및 특수문자(% , @ , ! 등) 예약어(if, else, for, while 등)	숫자로 시작, 공백 및 특수문자(% , @ , ! 등) 예약어(if, else, for, while 등)
차이점	. 을 변수명에 사용 불가	. 을 변수명에 사용 가능
예시	gpd19, gpd, if_1 (O) 19gpd, x/y, x.all (X)	gpd19, gpd, if_1, x.all (O) 19gpd, x/y (X)

# 연습문제

gdp2020

#key

2xl

mean.x

var%4

- R에서 변수명으로 사용할 수 없는 것
  - #key, var%4 : 특수문자 사용불가
  - 2xl : 숫자 시작 불가
- 파이썬에서 변수명으로 사용할 수 없는 것
  - #key, var%4 : 특수문자 사용불가
  - 2xl : 숫자 시작 불가
  - mean.x : . 사용불가

# 명령어 파일

구분	파이썬	R
경로구분자	/ , \\\ (\\)	/ , \\\ (\\)
함수	runfile("경로/파일명.py")	source("경로/파일명.r")
예시	runfile("D:\\\\wlang/python/hello.py")	source("D:\\\\wlang/R/hello.r")

# 연습문제

- "Hello Python!"을 인쇄하는 파이썬 프로그램 파일
  - 경로: "D:\lang\python\hello.py"
  - 아나콘다 프롬프트로 실행

```
(base) C:\Users\USER>python D:\lang\python\hello.py
```

- "Hello R!"을 인쇄하는 R 프로그램 파일
  - 경로: "D:\lang\R\hello.r"
  - R에서 실행

```
> source("D:\\lang\\R\\hello.r")
```



# 문자열 합치기

구분	파이썬	R
함수	연산자 "+" 사용	<code>paste(str1, str2, sep="")</code>

## ■ 연습문제

- 파이썬에서 '3'+ '5'의 결과

```
[1]: '3'+ '5'
```

```
[1]: '35'
```

- R에서 두 문자열 'abc'와 'cdf'를 합하여 하나의 문자열로 만들기

```
> paste('abc','cdf')  
[1] "abc cdf"
```

```
> a<-'abc' > paste(a,b)  
> b<-'cdf' [1] "abc cdf"
```

# 날짜와 시간

구분	파이썬	R
함수	datetime 패키지의 datetime	as.Date

## ■ 연습문제: 광복절인 '1945년 8월 15일'의 요일 계산

- 파이썬

```
import datetime as dt
indep_date = dt.datetime(1945,8,15)
print(indep_date.strftime("%A"))
```

Wednesday

- R

```
> format(as.Date('1945-8-15'), "%a")
[1] "Wed"
```

# 제 3장 파이썬과 R의 배열, 리스트, 데이터프레임

벡터와 행렬, 연산

# 벡터와 행렬

구분	파이썬	R
벡터 (vector)	리스트: [1, 2, 3]	c(1, 2, 3)
행렬(matrix)	NumPy 배열: np.array([[1,2],[3,4]])	matrix(c(1,2,3,4), nrow=2, ncol=2)
리스트(list)	[1, "a", True] (자료형 혼합 가능)	list(1, "a", TRUE) (자료형 혼합 가능)
튜플(tuple)	(1, 2, 3) (불변)	없음 (리스트로 대체)
딕셔너리 (dictionary)	{"a": 1, "b": 2}	list(a=1, b=2) 또는 named vector

# 연습문제

- 다음 자료 x1, x2, x3를 파이썬 리스트로 정의하고, 이들을 각 key의 value로 갖는 dictionary를 만드시오

자료	key
x1=['kim', 'lee', 'park']	name
x2=[170, 160, 180]	height
x3=[60, 55, 75]	weight

```
#파이썬 리스트 정의
x1=['kim', 'lee', 'park']
x2=[170, 160, 180]
x3=[60, 55, 75]
|
# 딕셔너리 생성
data = {"name": x1, "height": x2, "weight": x3 }

# 출력
print(data)
```

# 벡터의 연산

구분	파이썬	R
기본벡터	1차원 구조 행벡터	행/열 방향 없는 1차원 구조
합/차	같은 위치의 원소끼리 합/차 연산, 차원 다르면 오류 발생	같은 위치의 원소끼리 합/차 연산, 차원 다르면 재활용규칙 적용
스칼라곱	모든 원소에 곱해짐	모든 원소에 곱해짐
벡터의 곱(*)연산	같은 위치의 원소끼리 곱 연산, 차원 다르면 오류 발생	같은 위치의 원소끼리 곱 연산, 차원 다르면 재활용규칙 적용
항목 추가 및 삭제	numpy 패키지의 insert, append, delete	rbind, cbind

# R의 재활용규칙

- 차원이 다른 두 벡터 (또는 행렬)의 합, 차, 곱(\*) 연산시
  - 짧은 벡터를 자동으로 반복해서 연산에 맞춰주는 규칙
  - 예시

```
> a <- c(1, 2, 3, 4)
> b <- c(10, 20)
> a + b
[1] 11 22 13 24
> c <- c(1, 2, 3, 4, 5)
> d <- c(10, 20)
> c + d
[1] 11 22 13 24 15
```

Warning message:

In c + d : longer object length is not a multiple of shorter object length

# 연습문제

- 파이썬에서  $x=[1,2,3,4]$ 를 사용하여  $x1=[1,2,3,4,5]$ ,  $x2=[2,4,6,8]$ 을 만드시오.

*#리스트의 연산*

```
x=[1,2,3,4]
x1=x+[5]
x2=[num*2 for num in x]
print("x1=",x1)
print("x2=",x2)
```

```
x1= [1, 2, 3, 4, 5]
x2= [2, 4, 6, 8]
```

*#벡터의 연산*

```
import numpy as np
x=[1,2,3,4]
x1=np.append(np.array(x),5)
x2=np.array(x)*2
print("x1=",x1)
print("x2=",x2)
```

```
x1= [1 2 3 4 5]
x2= [2 4 6 8]
```



# 연습문제

- R에서  $x \leftarrow 1:4$ 일 때 이전 문제와 같은  $x1$ ,  $x2$ 를 만드는 프로그램

```
> x<-1:4
> x1<-c(x, 5)
> x2<-x*2
> x1
[1] 1 2 3 4 5
> x2
[1] 2 4 6 8
```

# 연습문제

- R에서 “xm <- matrix(1:12, ncol=6, byrow=T)” 행렬의 세번째 열 위치에 c(10, 20)을 끼워넣어 2x7 행렬을 만드시오.

```
> xm <- matrix(1:12, ncol=6, byrow=T)
> cbind(xm[,1:2], c(10,20), xm[,3:6])
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	1	2	10	3	4	5	6
[2,]	7	8	20	9	10	11	12

- cbind: 주어진 객체들을 “열(column)”로 정렬해 합침

# 연습문제

- 파이썬에서 같은 xm을 정의하고, 같은 결과를 얻어 보시오.

```
#항목 추가  
import numpy as np  
xm = np.arange(1, 13).reshape(2, 6)  
print(np.insert(xm, 2, [10, 20], axis=1))
```

```
[[ 1  2 10  3  4  5  6]  
 [ 7  8 20  9 10 11 12]]
```

- numpy.insert(대상, 위치, 삽입개체, axis=0(행) =1(열))

# 데이터프레임

구분	파이썬	R
기본형태	행렬 혹은 2차원 배열 (pandas DataFrame 기반)	행렬 혹은 2차원 배열 (data.frame 객체)
행(row)	일반적으로 한 개체(관측치)의 값	
열(column)	일반적으로 하나의 변수	
차이점	pandas 패키지에서 제공, 각 열의 길이가 다르면 <b>결측</b> (NaN)으로 채워 맞춤	R의 기본 자료구조 중 하나, 각 열의 길이가 다르면 <b>재활용</b> <b>규칙</b> 을 적용해 강제로 맞춤

# 연습문제

- 앞서 정의한 파이썬 딕셔너리로 파이썬 데이터 프레임을 만드시오.  
이 데이터 프레임에서 첫번째 사람 kim의 키와 몸무게 두 값 (이름제외)만 배열로 추출하시오

```
#파이썬 딕셔너리
x1=['kim', 'lee', 'park']
x2=[170, 160, 180]
x3=[60, 55, 75]
data = {"name": x1, "height": x2, "weight": x3 }
```

```
#데이터 프레임
import pandas as pd
df = pd.DataFrame(data=data)
print(df.iloc[0,[1,2]])
```

```
height    170
weight     60
Name: 0, dtype: object
```

# 연습문제

- R을 사용하여, 다음의 자료로 3개의 열(각 변수명은 name, height, weight)을 갖는 데이터 프레임을 만들고, 첫 번째 사람 kim의 키와 몸무게 두 값(이름제외)만 배열로 추출하시오.

```
> x1<-c('kim', 'lee', 'park')
> x2<-c(170, 160, 180)
> x3<-c(60,55,75)
> data<-data.frame('name'=x1, 'height'=x2, 'weight'=x3)
> data[1,2:3]
  height weight
1    170     60
```

# 제 4장 파이썬과 R의 구조적 프로그래밍

반복문, 조건문, 함수 작성, 파이썬 클래스

# 연습문제 : 반복문 (for문, while문)

- 1에서 5사이의 자연수의 값을 순서대로 인쇄하는 프로그램을 **for문**과 **while 문**을 이용하여 파이썬과 R로 작성하시오

구분	파이썬	R
for 문	<pre>for n in range(1, 6):     print(n)</pre>	<pre>for (n in 1:5){     print(n) }</pre>
while 문	<pre>n = 1 while n &lt;= 5:     print(n)     n = n+1</pre>	<pre>n &lt;- 1 while (n &lt;= 5) {     print(n)     n &lt;- n + 1 }</pre>



# 연습문제 : 반복문 (while - continue 문)

- 1에서 5사의 자연수의 값을 순서대로 인쇄하되, 4일 경우 건너뛰는 프로그램 작성( while 문과 continue (next) 문을 사용)

구분	파이썬	R
continue (next) 문	<pre>n = 1 while n &lt;= 5:     if n == 4:         n = n + 1         continue     print(n)     n = n + 1</pre>	<pre>n &lt;- 1 while (n &lt;= 5) {     if (n == 4) {         n &lt;- n + 1         next     }     print(n)     n &lt;- n + 1 }</pre>

# 연습문제 : 반복문 (while - break 문)

- 1에서 5사의 자연수의 값을 순서대로 인쇄하되, 4 이후에는 인쇄하지 않는 프로그램 작성( while 문과 break문을 사용)

구분	파이썬	R
break 문	<pre>n = 1 while n &lt;= 5:     if n == 4:         n = n+1         break     print(n)     n = n+1</pre>	<pre>n &lt;- 1 while (n &lt;= 5) {     if (n == 4) {         n &lt;- n + 1         break     }     print(n)     n &lt;- n + 1 }</pre>

# 연습문제 : 조건문을 이용한 함수 작성

- 숫자를 입력받아 이 숫자가 양수인지 음수인지 0인지 판단하여 출력해주는 프로그램 mysign을 작성하시오.

구분	파이썬	R
조건문을 이용한 함수작성	<pre>def mysign(x):     if x &gt; 0:         print(f"{x} : 양수")     elif x &lt; 0:         print(f"{x} : 음수")     else:         print("0입니다")</pre>	<pre>mysign &lt;- function(x) {     if (x &gt; 0) {         cat(paste0(x, " : 양수\n"))     } else if (x &lt; 0) {         cat(paste0(x, " : 음수\n"))     } else {         cat("0입니다\n")     } }</pre>

# 연습문제 : 조건문을 이용한 함수 작성

- 주당 근무시간이 40시간 이내면 시간당 임금이 1만원이고, 40시간이 초과되는 부분에 대해서는 1.5배를 지급하는 경우, 주당 근무시간을 입력 받으면 출력으로 임금이 계산되는 함수 프로그램 mywage를 작성하시오.

구분	파이썬	R
조건문을 이용한 함수 작성	<pre>def mywage(h):     r1= 10000     r2 = r1 * 1.5     if h &lt;= 40:         wage = h * r1     else:         wage = 40 * r1 + (h - 40) * r2     return wage</pre>	<pre>mywage &lt;- function(h) {     r1 &lt;- 10000     r2 &lt;- r1 * 1.5     if (h &lt;= 40) {         wage &lt;- h * r1     } else {         wage &lt;- 40 * r1 + (h - 40) * r2     }     return(wage) }</pre>

# 연습문제 : 파이썬 클래스

- 다음 R의 example 함수문을 파이썬 클래스문으로 만드시오.

구분	파이썬	R
파이썬 클래스	<pre>class Example:     def __init__(self, name):         self.a = f"Hello {name}!"         self.b = f"Good-bye {name}!"  aaa =Example("David") print(aaa.a) print(aaa.b)</pre>	<pre>example &lt;- function(name){   a = paste("Hello", name, "!")   b = paste("Good-bye", name, "!")   result &lt;-list(a=a, b=b)   return(result) } aaa&lt;-example("David") aaa\$a aaa\$b</pre>

# 제 5장 파이썬과 R을 이용한 자료의 입력 및 출력

텍스트 파일, csv 파일, 문자열

# 연습문제 : 텍스트 파일의 입력과 출력

- R 내장 데이터프레임 iris를 텍스트 파일로 저장
  - 따옴표 사용하지 않기 (quote=F)
  - 행번호 저장하지 않기 (row.names=F)
  - 인코딩은 UTF-8 (fileEncoding = 'UTF-8')

```
> data(iris)
> write.table(iris, file="iris.txt",
+             quote=F,
+             row.names=F,
+             fileEncoding = 'UTF-8')
```

# 연습문제 : 텍스트 파일의 입력과 출력

- 앞서 저장된 "iris.txt" 파일을 파이썬에서 데이터프레임으로 저장
  - 자료의 첫 줄은 변수 이름 (header=0)
  - 구분자는 공백 (sep=" ")
  - 경로 지정 (C:\Users\USER\Documents\iris.txt)

```
import pandas as pd
iris = pd.read_csv("C:\\Users\\USER\\Documents\\iris.txt", sep=" ", header=0)
```



# 연습문제: csv파일의 입력과 출력

- 앞서 파이썬에 입력한 iris 데이터프레임을 iris.csv 파일로 저장
  - 인코딩은 UTF-8 (encoding="UTF-8")
  - 따옴표 사용하지 않음 (quoting=3)
  - 행번호 저장하지 않기 (index=False)

```
#csv 파일의 입력과 출력
import pandas as pd
iris = pd.read_csv("C:\\Users\\USER\\Documents\\iris.txt", sep=" ", header=0)

iris.to_csv("C:\\Users\\USER\\Documents\\iris.csv", index=False, quoting=3, encoding="UTF-8")
```

# 연습문제: csv파일의 입력과 출력

- iris.csv 파일을 R에서 iris\_csv로 입력하고, iris2.csv로 저장
  - 따옴표 사용하지 않기 (quote=F)
  - 행번호 저장하지 않기 (row.names=F)
  - 인코딩은 UTF-8 (fileEncoding = 'UTF-8')

```
> iris_csv<-read.csv("iris.csv",sep=";",header=T)
> write.csv(iris_csv,file="iris2.csv",
+           quote=F,
+           row.names=F,
+           fileEncoding = 'UTF-8')
```

# 연습문제: 문자열

- 다음 주소에서 충남 연기군을 세종시로 바꾸시오.

구분	파이썬	R
문자열 바꾸기	<pre>addr = [ "충남 연기군 조치원읍 신흥리 123", "충남 연기군 조치원읍 교리 9-1", "충남 당진시 수청동 1002"]  [str.replace("충남 연기군", "세종시") for str in addr]</pre>	<pre>addr &lt;- c("충남 연기군 조치원읍 신흥리 123", "충남 연기군 조치원읍 교리 9-1", "충남 당진시 수청동 1002")  gsub("충남 연기군",'세종시', addr)</pre>

# 연습문제: 정규식

- 정규식 **010W-[0-9]{4}W-1234** 의 해석
  - **010** : 문자열이 010으로 시작
  - **W-** : 정규식에서 특별한 의미가 있는 문자 (-)를 문자 그대로 (하이픈 -) 인식시키고자 할 때 W를 붙여줌 (이스케이프)
  - **[0-9]** : 0에서 9까지의 정수 집합 중 하나
  - **[0-9]{4}** : “0에서 9까지의 정수 집합 중 하나”가 4자리
  - **1234** : 문자열이 1234로 끝나야 함
- 일치하는 문자열 “010-1234-1234”, “010-0000-1234” 등

# 출석수업 과제물

- 수업시간에 공지한 과제물 파일을 참고하여 제출
- 제출기한 : 출석수업 날짜별로 상이함

# 과제물 제출 시 주의사항

- 표지 또는 첫 페이지 상단에 **과목명, 이름, 학번, 소속 지역대학**을 명확히 표기한다.
- **문제 번호**를 기재하고 그에 대한 답안을 작성한다.
- **하나의 파일**로 작성하여 **PDF 형식으로 제출**하며, 보조 파일은 제출하지 않는다.
- 화면을 캡처한 자료를 사용한 경우, **글씨가 식별 가능**해야 하며, 독해 불가로 인한 감점은 반복되지 않는다.
- 온라인 자료나 타인의 작성 내용을 복사해서 포함하거나, 과제와 무관한 내용이 포함될 경우 **0점** 처리된다.

**감사합니다.**