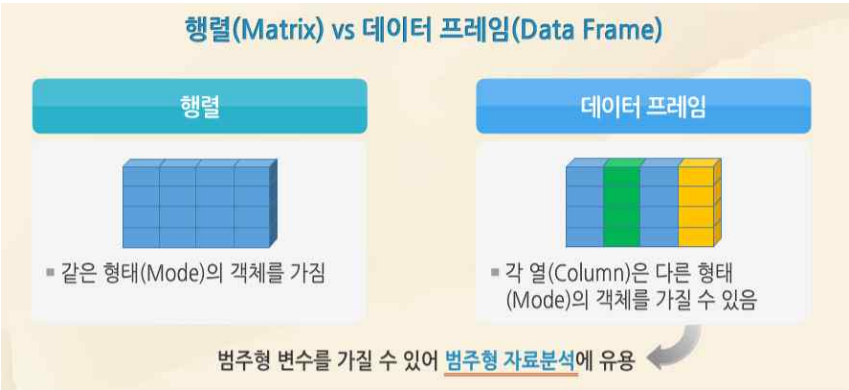
5. 데이터 프레임(data.Frame)

(1) 데이터 프레임이란?

데이터 프레임은 처리할 데이터를 마치 엑셀의 **스프레드시트**(spreadsheet)[1]와 같이 표(행렬) 형태로 정리한 모습을 하고 있다. 데이터 프레임의 각 열에는 관측 값의 이름이 저장되고, 각 행에는 매 관측 단위마다 실제 얻어진 값이 저장된다. 예를 들어, 다음과 같은 형태가 데이터 프레임에 저장되는 데이터의 전형적인 예다.

이름	성별	월급	동호회 가입
A	M	360	T
B	F	430	F
C	M	510	F
D	F	390	F

데이터 프레임은 행렬과 비슷한 형태로 되어 있으나, 다른 속성을 갖는다. 행렬은 같은 유형의 객체를 가지는 반면, 데이터 프레임은 각 열들이 서로 다른 유형의 객체(문자형, 숫자형, 논리형)를 가질 수 있다. 따라서 데이터 프레임은 범주형 변수를 가질 수도 있기 때문에 **범주형 자료분석**[2]에 유용하게 사용된다.



이처럼 자연스럽게 자료를 표현하는 데이터 유형이기 때문에 데이터 프레임은 R에서 가장 중요한 데이터 구조이다.

① **각 열마다 같은 길이의 벡터를 갖는 구조**(반면에 list 다른 길이를 갖는 구조)

② 데이터 프레임의 각각의 **열은 각각의 변수와 대응**하며, 분석이나 모형 설정에 적합한

---

1) 표 형식으로 데이터의 조직, 분석(계산), 저장을 가능케 하는 상호작용 컴퓨터 애플리케이션으로 **계산 기능을 수행하는 프로그램의 총칭**

2) 관측 결과가 몇 개의 범주 또는 항목의 형태로 나타나는 자료

자료 객체이다.

③ 엑셀에서 만든 **csv 파일**을 읽으면 자동으로 데이터 프레임이다.

(2) 데이터 프레임의 생성

```
data.frame(
   # value 또는 name=value로 표현된 데이터 값.
   # 주어진 문자열을 factor로 저장할 것인지 또는 문자열로 저장할 것인지를 지정한다.
   # 즉, stringsAsFactors=F로 설정하면 Factor가 아닌 문자형(Character)로 입력할 수 있다.
   # 기본값은 TRUE다. 따라서 이 인자를 지정하지 않으면 문자열은 factor로 저장된다[3].
)
```

데이터 프레임은 data.frame( )에 '**컬럼명=데이터**' 형태로 데이터를 나열하여 생성한다.

```
x <- data.frame(이름=c('A', 'B', 'C', 'D'), 성별=c('M', 'F', 'M', 'F'), 월급=c(360, 430, 510, 390),
동호회=c(T, F, F, F) ) ; x
   이름 성별 월급  동호회
1    A    M  360   TRUE
2    B    F  430  FALSE
3    C    M  510  FALSE
4    D    F  390  FALSE
```

위에 보인 출력에서는 각 열의 데이터 유형을 보여주지 않아 실제로 어떤 유형으로 데이터가 저장되는지는 쉽게 알 수 없다. 이 경우 str( )을 사용하여 데이터의 구조를 살펴볼 수 있다.

```
str(x)     # data.frame 객체(x)의 내부 구조(structure)를 살펴본다.
'data.frame': 4 obs. of  4 variables:
 $ 이름  : chr  "A" "B" "C" "D"
 $ 성별  : chr  "M" "F" "M" "F"
 $ 월급  : num  360 430 510 390
 $ 동호회: logi  TRUE FALSE FALSE FALSE
```

[실습] 각 열마다 **결측치**(NA)가 있다면, **열의 길이**가 같지 않다면?

① x <- data.frame(수학=c(73, 83, 92, 87,), 영어=c(79, 84, , 91, 83), 성별=c('M', 'F', 'M', 'F', 'M'))

---

3) R에서 factor 형은 범주형 변수에 사용되는데, 범주형 변수란 가질 수 있는 값이 미리 고정되고 또 알려진 변수를 말한다. 즉, 요인(factor) 자료형은 범주형 데이터를 R에서 저장하는 자료형이다.

② x <- data.frame(수학=c(73, 83, 92, 87, NA), 영어=c(79, 84, NA, 91, 83), 성별=c('M', 'F', 'M', 'F', 'M'))

```
   수학  영어  성별
1   73   79    M
2   83   84    F
3   92   NA    M
4   87   91    F
5   NA   83    M
```

(3) data.frame의 데이터에 접근 방법

데이터 프레임의 각 열은 x$colname과 같이 **열 이름**으로 접근할 수 있으며, **행이나 열의 색인(index)**을 사용해서도 데이터에 접근할 수 있다.

x$colname	데이터 프레임 x의 컬럼 이름(열변수) colname에 저장된 데이터
x[m, n]	**데이터 프레임 x의 m행 n열에 저장된 데이터**(성분). m과 n을 벡터로 지정하여 다수의 행과 열을 동시에 가져올 수 있으며 m, n에는 색인뿐만 아니라 행 이름이나 열 이름을 지정할 수 있다. x[,n]과 같이 행을 지정하지 않고 특정 열만 가져올 경우 반환 값은 데이터 프레임이 아니라 해당 열의 데이터 유형이 된다. 이러한 반환을 원하지 않으면 drop=F를 지정하여 **데이터 프레임**을 반환하도록 할 수 있다.

```
x <- data.frame(이름=c('이**', '김**', '박**', '정**'), 성별=c('M', 'F', 'M', 'F'), 월급=c(360, 430, 510, 390), 동호회=c(T, F, F, F) )

   이름  성별  월급  동호회
1  이**    M   360   TRUE
2  김**    F   430   FALSE
3  박**    M   510   FALSE
4  정**    F   390   FALSE

x[4, 3]  # data.frame x의 4행 3열의 성분
```

```
[1] 390
x[4, 3, drop=F]    # data.frame x의 4번째 사람의 월급을 쉽게 파악
   월급
4  390
x$성별    # data.frame x의 2열의 모든 데이터
[1] "M" "F" "M" "F“
x[ , 2]
[1] "M" "F" "M" "F“
x[  , c("성별")]
[1] "M" "F" "M" "F“

x[ ,c("이름", "월급")]    # data.frame x의 "이름"과 "월급" 열 데이터. x[ ,c(1,3)]와 동일
   이름  월급
1  이**  360
2  김**  430
3  박**  510
4  정**  390
x[ ,c(1,3)]

x[2, ]    # data.frame x의 2행의 모든 데이터, 2번째 사람의 모든 데이터
    이름  성별  월급  동호회
2   김**   F   430   FALSE

x[c(2, 4), 3]    # data.frame x의 2행과 4행의 3열의 성분들
[1] 430 390
x[c(2, 4), 3, drop=F]
   월급
2  430
4  390

x[-1, -2]    # data.frame x의 1행과 2열을 제외한 모든 성분들, 제외할 행 또는 열을 -(minus)를 이용하여 표시할 수 있다.
   이름 월급  동호회
2   B   430  FALSE
3   C   510  FALSE
```

```
4    D   390  FALSE
```

[실습] 위 데이터 프레임에서 다음과 같이 출력하는 R-코드를 작성하여라.

```
   이름  월급
2  김**  430
4  정**  390
```

(4) 데이터 프레임의 정렬

	성별	월급	지출
1	M	360	320
2	F	420	270
3	M	480	260
4	F	320	280
5	F	465	350
6	M	390	230

예를 들어, 다음과 같은 월급을 순서대로 정렬해야 한다면 sort( ) 함수를 사용하면 간단하다.

```
월급 <- c(360, 420, 480, 320, 465, 390) ; sort(월급)
```

만일 내림차순으로 정렬하고 싶다면 [decreasing=T] 옵션을 사용한다.

```
sort(월급, decreasing=T)
```

하지만 데이터 프레임은 **sort( )**로 정렬하기가 난감하다. **x$월급**을 sort( )로 정렬할 수는 있겠지만, 데이터 프레임 전체가 정렬되지는 않는다.

```
x <- data.frame(성별=c('M', 'F', 'M', 'F', 'F', 'M'), 월급=c(360, 420, 480, 320, 465, 390), 지출=c(320, 270, 260, 280, 350, 230) ); x
  성별 월급 지출
1    M  360  320
2    F  420  270
3    M  480  260
4    F  320  280
5    F  465  350
6    M  390  230
str(x)
'data.frame':   6 obs. of  3 variables:
 $ 성별: chr  "M" "F" "M" "F" ...
 $ 월급: num  360 420 480 320 465 390
 $ 지출: num  320 270 260 280 350 230
sort( x$월급 )
[1] 320 360 390 420 465 480    # 오름차순
```

이럴 때는 order( ) 함수를 사용한다. order( )는 각 요소의 **상대적인 순서(순위)**를 반환한다.

```
order( x$월급 )      # 각 요소의 상대적인 순위
[1] 4 1 6 2 5 3
x$월급
[1] 360 420 480 320 465 390
```

(4)320-(1)360-(6)390-(2)420-(5)465-(3)480 과 같이 값을 순서대로 정렬한 후, 그 값이 데이터의 몇 번째 위치에 존재하는지를 알려준다. 즉, x를 월급을 기준으로 정렬하고 싶다면 **4행, 1행, 6행, 2행, 5행, 3행**의 순서로 배열하면 되는 것이다. 그러므로

```
x <- x[ order( x$월급 ), ] ; x

  성별 월급 지출
4    F  320  280
1    M  360  320
6    M  390  230
2    F  420  270
5    F  465  350
3    M  480  260
```

"행 번호가 거슬린다면" **초기화**를 하면 **월급을 기준으로 오름차순으로 정렬**된다.

```
rownames(x) <- NULL ; x        # NULL은 대문자!!!

   성별 월급 지출
1    F  320  280
2    M  360  320
3    M  390  230
4    F  420  270
5    F  465  350
6    M  480  260
```

또한 **성별에 따른 월급을 오름차순으로 정렬**하고 싶다면, 즉 두 개 이상의 인자에 따른 정렬은 order( ) 안에 순서대로 적어 주면 된다.

```
x <- x[ order( x$성별, x$월급 ), ] ; x  (○)
x <- x[ order( x$월급, x$성별 ), ] ; x  # 월급을 기준으로 오름차순으로 정렬
   성별 월급 지출
1    F  320  280
4    F  420  270
5    F  465  350
2    M  360  320
3    M  390  230
6    M  480  260
```

만약, **성별에 따른 지출을 내림차순으로 정렬**하고 싶다면 [decreasing=T]를 사용할 경우, 성별도 **내림차순 정렬**(남성부터 정렬)될 것이므로 까다롭다.
이럴 땐 숫자에 (-)를 곱해 더 큰 수가 더 작은 수가 되도록 만들어주면 된다.

```
x <- x[ order( x$성별, x$지출 ), ] ; x
   성별 월급  지출
4    F   420  270
1    F   320  280
5    F   465  350
3    M  390  230
6    M  480  260
2    M  360  320
```

```
x <- x[ order( x$성별, x$지출, decreasing=T ), ] ; x
    성별 월급 지출
2     M  360  320
6     M  480  260
3     M  390  230
5     F  465  350
1     F  320  280
4     F  420  270

x <- x[ order( x$성별, -x$지출 ), ] ; x

   성별  월급   지출
5    F   465   350
1    F   320   280
4    F   420   270
2    M   360   320
6    M   480   260
3    M   390   230
```

[실습] 2020년 OECD 국가의 주요지표 (한글파일 참조)

(1) 데이터 프레임을 생성하여라.

data.frame(국가별=c( ), 1인당GDP=c( ), GDP성장률=c( ), 실업률=c( ))

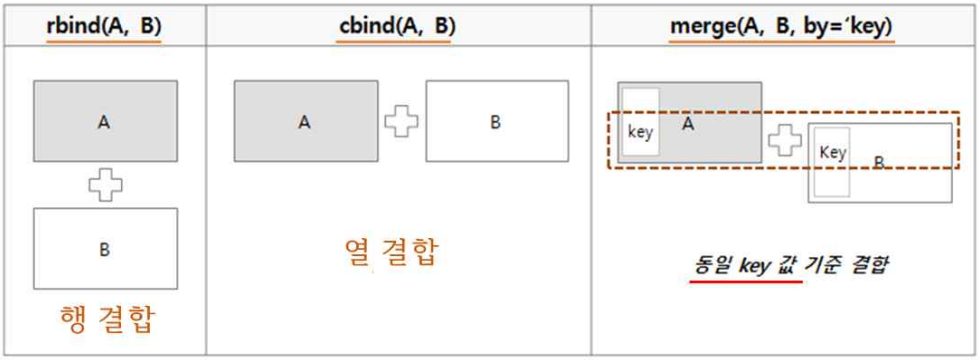
(2) 데이터 프레임을 **1인당 GDP**을 기준으로 **내림차순**으로 정렬하여라.

(3) 국가별에 따른 **실업률**을 **오름차순**으로 정렬하여라.

(4) (1)의 데이터 프레임에 **장래인구 열**을 결합하여라.

(5) 데이터 프레임의 결합 : rbind( ), cbind( ), merge( )

분석을 진행하다 보면 하나의 데이터 셋에서 변수를 생성, 제거, 변환하는 작업도 있지만, 새로운 데이터 셋을 기존의 데이터 셋과 결합하는 경우가 많다. **rbind(), cbind(), merge()** 함수를 활용해서 데이터 프레임을 결합하는 방법에 대해서 알아보자.



데이터 프레임에서 행결합 rbind( )와 열결합 cbind( )는 행렬에서의 데이터 결합과 동일하며, 여기서는 key값 기준으로 결합하는 merge( ) 함수에 대해서 추가로 알아보자.

	성별	월급	지출
1	M	360	320
2	F	420	270
3	M	480	260
4	F	320	280
5	F	465	350
6	M	390	230

나이
35
29
33
30
31
34

나이	도시
35	서울
29	전주
33	대전
30	부산
31	대구
34	울산

① 열 결합

```
x <- data.frame(성별=c('M', 'F', 'M', 'F', 'F', 'M'), 월급=c(360, 420, 480, 320, 465, 390), 지출=c(320, 270, 260, 280, 350, 230) ); x
y <- cbind( x, list( 나이=c(35, 29, 33, 30, 31, 34) ) ) ; y
y <- cbind( x, 나이=c(35, 29, 33, 30, 31, 34) ) ; y
```

```
  성별 월급 지출 나이
1    M  360  320   35
2    F  420  270   29
3    M  480  260   33
4    F  320  280   30
5    F  465  350   31
6    M  390  230   34
```

```
z <- cbind( x, 나이=c(35, 29, 33, 30, 31, 34), 도시=c('서울', '전주', '대전', '부산', '대구', '울산') ) ; z
```

② 열(행) 삭제

```
y[ ,-4]     # 음수 부호는 R에게 해당 열을 삭제(제외)하게 지시한다.
z[ ,-5]
```

③ 행 결합

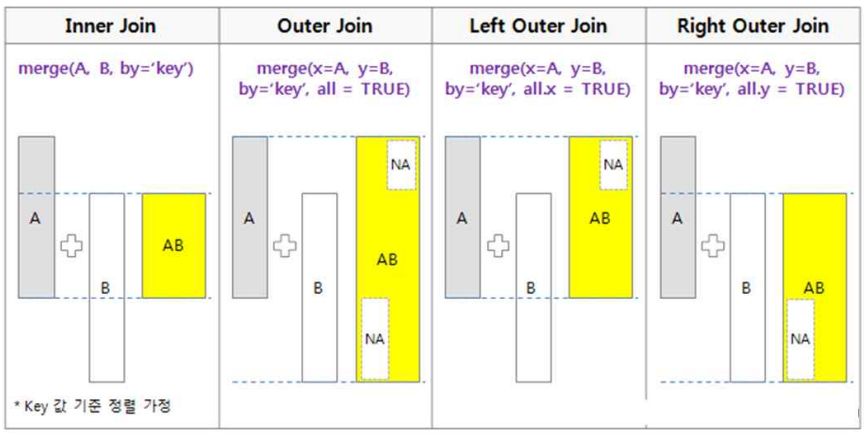
```
w <- rbind( x, c(성별='M', 월급=415, 지출=335) ) ; w
w <- rbind( x, c('M', 415, 335) ) ; w
```

```
   성별  월급  지출
1    M  360   320
2    F  420   270
3    M  480   260
4    F  320   280
5    F  465   350
6    M  390   230
7    M  415   335
```

④ merge( ) 함수

R에서 데이터프레임을 합성하는 명령은 merge이다. merge 명령을 사용하면 두 데이터 프레임에 **공통적인 이름을 키(key)**로 사용해서 데이터를 합친다.

merge(A, B, by='key')에는 기준을 어느 쪽에 두고 어디까지 포함하느냐에 따라 Inner Join, Outer Join, Left Outer Join, Right Outer Join 등의 4가지 종류가 있다. 이를 도식화하면 아래와 같다.



[그림] merge( ) 함수의 join 종류[4)]

4) 이미지 출처 https://rfriend.tistory.com/51

2018년도 1인당 GDP 주요국가 (1위~10위)[5)]

국가	국내총생산(GDP)(당해년 가격) (10억US$)	1인당 GDP(당해년 가격) (달러)	GDP 성장률(%)	소비자물가지수 (2010=100)	인터넷 이용률 (%)
룩셈부르크	69.5	114,340	2.6	113.1	97.1
스위스	705.5	82,839	2.5	99.2	
노르웨이	434.8	81,807	1.4	117.7	96.5
아일랜드	375.9	77,450	6.7	105.6	84.5
아이슬란드	25.9	73,191	4.6	125.2	99.0
미국	20,494.1	62,641	2.9	115.2	
덴마크	351.3	60,596	1.4	109.5	97.6
오스트레일리아	1,432.2	57,305	2.8	117.9	
스웨덴	551.0	54,112	2.4	108.6	92.1
네덜란드	912.9	52,978	2.7	112.9	94.7

2018년도 인터넷 이용률 주요국가 (1위~10위)

국가	국내총생산(GDP) (당해년 가격) (10억US$)	1인당 GDP (당해년 가격) (달러)	GDP 성장률(%)	소비자 물가지수 (2010=100)	인터넷 이용률(%)
아이슬란드	25.9	73,191	4.6	125.2	99.0
덴마크	351.3	60,596	1.4	109.5	97.6
룩셈부르크	69.5	114,340	2.6	113.1	97.1
노르웨이	434.8	81,807	1.4	117.7	96.5
한국	1,720.9	33,346	2.7	104.5	95.9
영국	2,825.2	42,491	1.4	117.6	94.9
네덜란드	912.9	52,978	2.7	112.9	94.7
스웨덴	551.0	54,112	2.4	108.6	92.1
독일	3,996.8	48,196	1.4	111.2	89.7
에스토니아	30.3	22,928	3.9	119.4	89.4

5) 자료출처: OECD 국가의 주요지표
http://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT_2KAAG01

```
x <- data.frame(
      key=c('룩셈부르크', '스위스', '노르웨이', '아일랜드', '아이슬란드', '미국', '덴마크', '오스트레일리아', '스웨덴', '네덜란드'),
      GDP=c(114340, 82839, 81807, 77450, 73191, 62641, 60596, 57305, 54112, 52978)
                    )
y <- data.frame(
      key=c('아이슬란드', '덴마크', '룩셈부르크', '노르웨이', '한국', '영국', '네덜란드', '스웨덴', '독일', '에스토니아'),
      인터넷=c(99.0, 97.6, 97.1, 96.5, 95.9, 94.9, 94.7, 92.1, 89.7, 89.4)
                    )

merge(x, y) # 또는 merge(x, y, by='key'), 나열 순서는 key의 가나다 순...
       key    GDP  인터넷
1   네덜란드  52978   94.7
2   노르웨이  81807   96.5
3     덴마크  60596   97.6
4 룩셈부르크 114340   97.1
5     스웨덴  54112   92.1
6 아이슬란드  73191   99.0
```

다음 인수를 사용하면 합치는 방식을 변경할 수 있다.

- all=TRUE 이면 두 데이터프레임의 키 열의 모든 데이터를 포함
- all.x=TRUE 이면 첫 번째 데이터프레임의 키 열의 모든 데이터를 포함
- all.y=TRUE 이면 두 번째 데이터프레임의 키 열의 모든 데이터를 포함

```
merge(x, y, all=T)       # 두 데이터 프레임 x, y 의 키 열의 모든 데이터를 포함
           key    GDP  인터넷
1     네덜란드  52978   94.7
2     노르웨이  81807   96.5
3       덴마크  60596   97.6
4   룩셈부르크 114340   97.1
5         미국  62641     NA
6       스웨덴  54112   92.1
7       스위스  82839     NA
8   아이슬란드  73191   99.0
9     아일랜드  77450     NA
```

```
10 오스트레일리아    57305      NA
11            독일       NA    89.7
12       에스토니아       NA    89.4
13            영국       NA    94.9
14            한국       NA    95.9
```

```
merge(x, y, all.x=T) # 첫 번째 데이터프레임 x의 키 열의 모든 데이터를 포함
           key     GDP   인터넷
1       네덜란드   52978    94.7
2       노르웨이   81807    96.5
3         덴마크   60596    97.6
4      룩셈부르크  114340    97.1
5           미국   62641      NA
6         스웨덴   54112    92.1
7         스위스   82839      NA
8      아이슬란드   73191    99.0
9       아일랜드   77450      NA
10 오스트레일리아   57305      NA
```

```
merge(x, y, all.y=T) # 두 번째 데이터프레임 y의 키 열의 모든 데이터를 포함
          key     GDP   인터넷
1      네덜란드   52978    94.7
2      노르웨이   81807    96.5
3        덴마크   60596    97.6
4     룩셈부르크  114340    97.1
5       스웨덴   54112    92.1
6     아이슬란드   73191    99.0
7          독일      NA    89.7
8     에스토니아      NA    89.4
9          영국      NA    94.9
10         한국      NA    95.9
```