



# 비즈니스프로그래밍

- Python과 R 기초 -



## 5. 데이터 전처리




- ▶ 5.1 데이터 추출하기
  - ▶ 조건에 맞는 데이터만 추출하는 filter 함수
  - ▶ 필요한 변수를 추출하는 select 함수
  - ▶ dplyr 함수 조합하기

dplyr 함수	기능
filter( )	행 추출: 조건에 맞는 관측 값만 추출
select( )	열 추출: 필요한 변수만 추출

# 조건에 맞는 데이터 추출

- ▶ dplyr 패키지의 filter() 함수는 데이터 세트에서 조건에 맞는 행을 추출하고자 할 때 사용

name	department	worktime
aaa	관리부	45
bbb	영업부	48
ccc	홍보부	50
ddd	영업부	34
eee	영업부	44
fff	인사부	40
ggg	개발부	39




name	department	worktime
bbb	영업부	48
ddd	영업부	34
eee	영업부	44



# 필요한 변수 추출



- ▶ dplyr 패키지의 select( ) 함수는 데이터 세트에 포함되어 있는 여러 변수 중에서 필요한 변수만 추출하여 활용하고자 할 때 사용

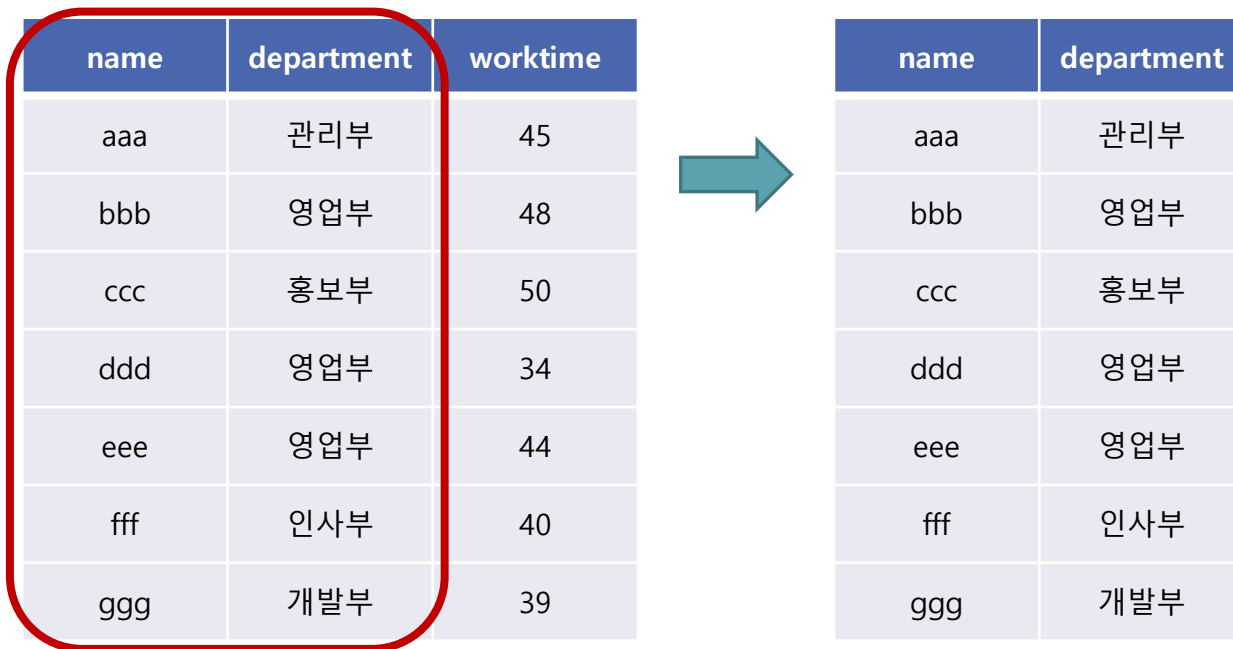


name	department	worktime
aaa	관리부	45
bbb	영업부	48
ccc	홍보부	50
ddd	영업부	34
eee	영업부	44
fff	인사부	40
ggg	개발부	39

name	department
aaa	관리부
bbb	영업부
ccc	홍보부
ddd	영업부
eee	영업부
fff	인사부
ggg	개발부

# dplyr 함수 조합하기

- ▶ dplyr 패키지의 함수를 조합할 때에는 %>% 연산자를 이용하여 조합하여 사용할 수 있다.



name	department	worktime
aaa	관리부	45
bbb	영업부	48
ccc	홍보부	50
ddd	영업부	34
eee	영업부	44
fff	인사부	40
ggg	개발부	39

name	department
aaa	관리부
bbb	영업부
ccc	홍보부
ddd	영업부
eee	영업부
fff	인사부
ggg	개발부



## 5. 데이터 전처리



- ▶ 5.2 데이터 정렬하기
  - ▶ 하나의 기준으로 정렬하기
  - ▶ 여러 개의 정렬 기준 적용하기
  - ▶ 추출과 정렬 조합하기

dplyr 함수	기능
arrange( )	정렬: 지정한 변수를 오름차순 or 내림차순 정렬



# 데이터 정렬하기



- ▶ `arrange()` 함수는 데이터를 오름차순 또는 내림차순으로 정렬할 때 사용
- ▶ 내림차순으로 정렬 → `desc()`를 함께 사용



## 5. 데이터 전처리



- ▶ 5.3 데이터 변형하기
  - ▶ 파생 변수 추가하기
  - ▶ 새로운 데이터 세트에 파생 변수 추가하기

dplyr 함수	기능
mutate( )	변수 추가





# 데이터 변형하기



- ▶ `mutate()` 함수는 데이터 세트에 파생 변수(열)를 만들어 추가할 때 사용
- ▶ 기존의 변수를 가공한 후 그 결과 값을 기존 변수나 새로운 변수에 할당할 수 있다.
- ▶ 할당하지 않으면 추가되지 않는다는 점 주의!



## 5. 데이터 전처리



### ▶ 5.4 데이터 요약하기

- ▶ 전체 데이터의 평균
- ▶ 그룹별 평균
- ▶ 합계와 개수 구하기

dplyr 함수	기능
summarise( )	통계치 산출: 빈도, 최대값, 최소값, 합계, 평균 등
group_by( )	집단별로 나누기: 그룹별 통계치 산출



# 데이터 요약하기



- ▶ summarise( ) 함수는 mean( ), max( ), min( ), n( ), sum( ) 함수 등 통계 함수와 함께 집단별로 데이터를 요약할 때 사용
- ▶ group\_by( ) 함수는 특정 변수에 대해 변수 항목별로 집단화
- ▶ sum( ) 함수를 사용하여 합계를 구하고, n( ) 함수를 사용하여 개수를 구할 수 있다.



## 5. 데이터 전처리



### ▶ 5.5 데이터 결합하기

▶ 가로로 결합하기

▶ 세로로 결합하기

dplyr 함수	기능
inner_join( )	기준으로 정한 변수의 값이 동일할 때만 결합
left_join( )	특정 변수 값을 기준으로 다른 데이터의 값을 추가
full_join( )	기준으로 정한 변수의 전체 데이터를 결합
bind_rows( )	세로로 결합



# 가로로 결합하기



- ▶ inner\_join은 가장 간단한 조인 유형으로 key가 되는 변수 값이 동일할 때만 가로로 결합
- ▶ left\_join은 왼쪽에 있는 데이터 프레임의 key를 기준으로 결합
- ▶ full\_join은 기준으로 정한 변수를 기준으로 모든 데이터를 가로로 결합

df\_1

이름	출생년도
aaa	1984
bbb	1989
ccc	1991



df\_2

이름	출생지
aaa	서울
ccc	인천
ddd	대전



df\_inner

이름	출생년도	출생지
aaa	1984	서울
ccc	1991	인천

df\_left

이름	출생년도	출생지
aaa	1984	서울
bbb	1989	<NA>
ccc	1991	인천

df\_full

이름	출생년도	출생지
aaa	1984	서울
bbb	1989	<NA>
ccc	1991	인천
ddd	<NA>	대전



# 세로로 결합하기



- ▶ `bind_rows`는 기존 데이터 세트에 변수가 같은 데이터를 추가하는 기능

**df\_a**

이름	출생년도
aaa	1984
bbb	1989
ccc	1985
ddd	1991



**df\_b**

이름	출생년도
eee	1986
fff	1990
ggg	1988

**df\_bindrow**

이름	출생년도
aaa	1984
bbb	1989
ccc	1985
ddd	1991
eee	1986
fff	1990
ggg	1988



## 6. 데이터 정제



- ▶ 6.1 결측 데이터 처리
  - ▶ 결측치 확인
  - ▶ 결측치가 있는 행 제거
  - ▶ 결측치를 다른 값으로 대체



# 결측치 확인



- ▶ 결측치(missing value): 데이터에서 누락되거나 비어있는 값
- ▶ 실제데이터는 수집과정에서 발생한 오류로 인해 결측치를 포함하는 경우가 많다.
- ▶ 결측치 포함 데이터는 분석 과정에서 함수의 오류로 인해 정상적인 결과를 얻을 수 없으므로 결측치를 제거하여 데이터를 정제하는 과정이 필요
- ▶ R에서 결측치는 대문자 NA로 표기





# 결측치 확인



- ▶ 결측치 확인

```
is.na(데이터이름)
```

- ▶ 결측치의 개수 확인

```
table(is.na(데이터이름))
```

- ▶ 특정변수의 결측치 수 확인

```
table(is.na(데이터이름$변수이름))
```



# 결측치가 있는 행 제거



- ▶ `is.na()` 함수에 `dplyr` 패키지의 `filter()` 함수를 적용하면 데이터에서 결측치 NA가 포함된 행을 제거할 수 있다. 논리연산자 `!(not)`을 이용

- ▶ 결측치가 없는 행 출력하기

```
데이터이름 %>% filter(!is.na(변수이름))
```

- ▶ 여러 변수의 결측치 제거하기

```
데이터이름 %>% filter(!is.na(변수1이름) & !is.na(변수2이름))
```

- ▶ 여러 변수의 결측치 한번에 제거하기

```
na.omit(데이터이름)
```

- ▶ 함수 파라미터로 결측치 제거하기

```
, na.rm = T
```

# 결측치를 다른 값으로 대체

- ▶ 결측치를 모두 제거하면 분석 결과에 영향을 줄 수 있는 경우에 결측치를 제거하는 대신 평균이나 최빈값과 같은 대표값으로 결측치를 대체하여 자료를 분석할 수 있다.

```
데이터이름$변수이름 <- ifelse(is.na(데이터이름$변수이름), mean, 데이터이름$변수이름)
```

- ▶ 참고: R에서는 최빈값 함수가 따로 없으므로 함수 조합으로 구현하여 값을 구한 후 대체해야 한다.



## 6. 데이터 정제



- ▶ 6.2 이상 데이터 처리
  - ▶ 이상치 찾기
  - ▶ 이상치 제거하기



# 이상치 찾기



- ▶ 이상치(outlier): 데이터의 정상 범주에서 크게 벗어난 값
- ▶ 실제데이터는 수집과정의 오류나 드물게 발생하는 극단적인 현상으로 인해 이상치를 포함할 수 있다.
- ▶ 분석결과에 영향을 주어 결과를 편향되게 할 수 있으므로 데이터 정제과정에서 분석 상황에 따라 결측치로 처리하여 분석대상에서 제외하거나 대표값으로 대체할 수 있다.
- ▶ 범주형의 이상치 찾기 ➔ `table( )`
- ▶ 연속형의 이상치 찾기 ➔ `boxplot( )`



# 상자 그림



- ▶ 상자 그림 Box Plot : 데이터의 분포(퍼져 있는 형태)를 직사각형 상자 모양으로 표현한 그래프
  - ▶ 상자 그림을 보면 분포를 알 수 있기 때문에 평균만 볼 때보다 데이터의 특징을 더 자세히 이해할 수 있다.

상자 그림	값	설명
수염 아래 경계선	최소값	이상치를 제외한 데이터 중 최소값
상자 아래 세로선	아래 수염(whisker)	하위 0~25% 내에 해당하는 값
상자 밑면	1사분위수(Q1)	하위 25% 위치 값
상자 내 굵은 선	2사분위수(Q2)	하위 50% 위치 값 (중앙값)
상자 윗면	3사분위수(Q3)	하위 75% 위치 값
상자 위 세로선	위 수염(whisker)	하위 75~100% 내에 해당하는 값
수염 위 경계선	최대값	이상치를 제외한 데이터 중 최대값
상자 밖 점 표식	극단치	Q1, Q3 밖 1.5 IQR을 벗어난 값

- ▶ 이상치를 제외한 나머지 데이터의 통계값

```
boxplot(데이터이름$변수이름)$stat
```



# 이상치 제거하기



- ▶ 정상범위 확인 후, 이 범위를 벗어나는 이상치를 결측치(NA)로 처리

```
data <- ifelse(data < 정상범위최소값 | data > 정상범위최대값, NA, data)
```