

## 2강

# R 데이터처리(2)

정보통계학과 김성수교수

### 목 차

1. 변수 값 바꾸기 및 결측치 처리
2. 변수이름 바꾸기
3. 변수 값 라벨
4. 변수 값 변환
5. 케이스 선택
- 6~7. dplyr 패키지 활용 (1)(2)



# 1 변수 값 바꾸기 및 결측치 처리



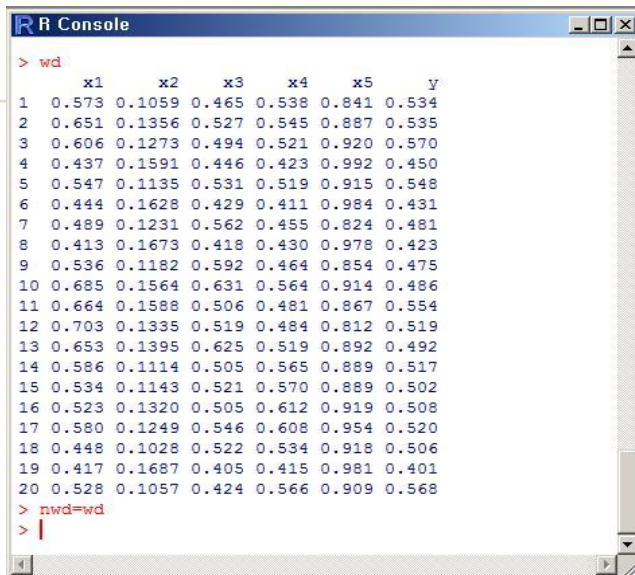
# 변수 값 바꾸기 및 결측치 처리

예1) x2가 0.11 보다 작은 경우를 99로 바꾸고,  
이를 결측치로 처리하는 경우

```
> nwd = wd  
> nwd[nwd$x2 < 0.11, "x2"] = 99  
> nwd[nwd == 99 ] = NA
```

예2) 0.9 보다 큰 경우를 모두 결측치로 처리  
하는 경우,

```
> nwd[nwd > 0.9] = 99  
> nwd[nwd == 99] = NA
```



The R Console window displays the following data table:

|    | x1    | x2     | x3    | x4    | x5    | y     |
|----|-------|--------|-------|-------|-------|-------|
| 1  | 0.573 | 0.1059 | 0.465 | 0.538 | 0.841 | 0.534 |
| 2  | 0.651 | 0.1356 | 0.527 | 0.545 | 0.887 | 0.535 |
| 3  | 0.606 | 0.1273 | 0.494 | 0.521 | 0.920 | 0.570 |
| 4  | 0.437 | 0.1591 | 0.446 | 0.423 | 0.992 | 0.450 |
| 5  | 0.547 | 0.1135 | 0.531 | 0.519 | 0.915 | 0.548 |
| 6  | 0.444 | 0.1628 | 0.429 | 0.411 | 0.984 | 0.431 |
| 7  | 0.489 | 0.1231 | 0.562 | 0.455 | 0.824 | 0.481 |
| 8  | 0.413 | 0.1673 | 0.418 | 0.430 | 0.978 | 0.423 |
| 9  | 0.536 | 0.1182 | 0.592 | 0.464 | 0.854 | 0.475 |
| 10 | 0.685 | 0.1564 | 0.631 | 0.564 | 0.914 | 0.486 |
| 11 | 0.664 | 0.1588 | 0.506 | 0.481 | 0.867 | 0.554 |
| 12 | 0.703 | 0.1335 | 0.519 | 0.484 | 0.812 | 0.519 |
| 13 | 0.653 | 0.1395 | 0.625 | 0.519 | 0.892 | 0.492 |
| 14 | 0.586 | 0.1114 | 0.505 | 0.565 | 0.889 | 0.517 |
| 15 | 0.534 | 0.1143 | 0.521 | 0.570 | 0.889 | 0.502 |
| 16 | 0.523 | 0.1320 | 0.505 | 0.612 | 0.919 | 0.508 |
| 17 | 0.580 | 0.1249 | 0.546 | 0.608 | 0.954 | 0.520 |
| 18 | 0.448 | 0.1028 | 0.522 | 0.534 | 0.918 | 0.506 |
| 19 | 0.417 | 0.1687 | 0.405 | 0.415 | 0.981 | 0.401 |
| 20 | 0.528 | 0.1057 | 0.424 | 0.566 | 0.909 | 0.568 |

The console also shows the following commands and output:

```
> wd  
> nwd=wd  
> |
```

# 변수 값 바꾸기 및 결측치 처리

## 예3) 각 행별, 열별 결측치의 수를 구하는 경우

```
> head(nwd, n=5)
```

|   | x1    | x2     | x3    | x4    | x5    | y     |
|---|-------|--------|-------|-------|-------|-------|
| 1 | 0.573 | NA     | 0.465 | 0.538 | 0.841 | 0.534 |
| 2 | 0.651 | 0.1356 | 0.527 | 0.545 | 0.887 | 0.535 |
| 3 | 0.606 | 0.1273 | 0.494 | 0.521 | NA    | 0.570 |
| 4 | 0.437 | 0.1591 | 0.446 | 0.423 | NA    | 0.450 |
| 5 | 0.547 | 0.1135 | 0.531 | 0.519 | NA    | 0.548 |

```
> rowSums(is.na(nwd))
```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 2  | 1  | 2  |

```
> colSums(is.na(nwd))
```

| x1 | x2 | x3 | x4 | x5 | y |
|----|----|----|----|----|---|
| 0  | 3  | 0  | 0  | 1  | 0 |

```
> mywd = na.omit(nwd)
```

```
> head(mywd)
```

|    | x1    | x2     | x3    | x4    | x5    | y     |
|----|-------|--------|-------|-------|-------|-------|
| 2  | 0.651 | 0.1356 | 0.527 | 0.545 | 0.887 | 0.535 |
| 7  | 0.489 | 0.1231 | 0.562 | 0.455 | 0.824 | 0.481 |
| 9  | 0.536 | 0.1182 | 0.592 | 0.464 | 0.854 | 0.475 |
| 11 | 0.664 | 0.1588 | 0.506 | 0.481 | 0.867 | 0.554 |
| 12 | 0.703 | 0.1335 | 0.519 | 0.484 | 0.812 | 0.519 |
| 13 | 0.653 | 0.1395 | 0.625 | 0.519 | 0.892 | 0.492 |

- rowSums(is.na(wd)) :  
각 행별로 결측치의 수를 나타냄.
- colSums(is.na(wd)) :  
각 열별로 결측치의 수를 나타냄.
- na.omit(wd) : 결측치를 제거

## 2 변수이름 바꾸기



# 변수이름 바꾸기

- 변수이름을 바꾸고자 할 때 `fix()`, `names()`, `colnames()` 를 이용
- 이외에도 R 패키지 `reshape`가 이용

예) `fix()`를 이용한 대화형 변수이름 바꾸기

The screenshot shows the R environment with a console window on the left and a Data Editor window on the right. The console shows the command `fix(nwd)` being executed. The Data Editor window displays a data frame with 19 rows and 6 columns: `nx1`, `x2`, `x3`, `x4`, `x5`, and `y`. A 'Variable editor' dialog box is open, allowing the user to rename the variable `y` to `yvalue`. The dialog box has a 'variable name' field containing `yvalue` and a 'type' section with `numeric` selected and `character` unselected.

|    | nx1   | x2     | x3    | x4    | x5    | y     |
|----|-------|--------|-------|-------|-------|-------|
| 1  | 0.573 | NA     | 0.465 | 0.538 | 0.841 | 0.534 |
| 2  | 0.651 | 0.1356 | 0.527 | 0.545 | 0.887 | 0.535 |
| 3  | 0.606 | 0.1273 | 0.494 | 0.521 | NA    | 0.57  |
| 4  | 0.437 | 0.1591 | 0.446 | 0.423 | NA    | 0.45  |
| 5  | 0.547 | 0.1135 | 0.531 | 0.519 | NA    | 0.548 |
| 6  | 0.444 | 0.1628 | 0.429 | 0.411 | NA    | 0.431 |
| 7  | 0.489 | 0.1231 | 0.562 | 0.455 | 0.824 | 0.481 |
| 8  | 0.413 |        |       |       |       |       |
| 9  | 0.536 |        |       |       |       |       |
| 10 | 0.685 |        |       |       |       |       |
| 11 | 0.664 |        |       |       |       |       |
| 12 | 0.703 |        |       |       |       |       |
| 13 | 0.653 |        |       |       |       |       |
| 14 | 0.586 | 0.1114 | 0.505 | 0.565 | 0.889 | 0.517 |
| 15 | 0.534 | 0.1143 | 0.521 | 0.57  | 0.889 | 0.502 |
| 16 | 0.523 | 0.132  | 0.505 | 0.612 | NA    | 0.508 |
| 17 | 0.58  | 0.1249 | 0.546 | 0.608 | NA    | 0.52  |
| 18 | 0.448 | NA     | 0.522 | 0.534 | NA    | 0.506 |
| 19 | 0.417 | 0.1687 | 0.405 | 0.415 | NA    | 0.401 |

예) 변수 이름 바꾸기

```
> names(nwd)[6] = "ny"  
> colnames(nwd) = c("a1", "a2", "a3", "a4", "a5", "newy")
```

# 변수이름 바꾸기

- R 패키지 reshape를 이용하여 변수이름을 바꿀 때 rename() 함수를 이용

```
> library(reshape)
> names = c("김", "이", "박")
> ages = c(50,44,35)
> mydata <- data.frame(names,ages)
> names(mydata)
[1] "names" "ages"
> mydata
  names ages
1   김   50
2   이   44
3   박   35
> mydata <- rename(mydata, c(names="name"))
> mydata <- rename(mydata, c(ages="age"))
> mydata
  name age
1   김  50
2   이  44
3   박  35
```

### 3 변수 값 라벨





# 값 라벨 (Value labels)

- 숫자로 입력된 값을 라벨로 바꾸기

예) 변수 job 1 = 근로자, 2 = 사무직, 3 = 전문가

edu 1 = 무학, 2 = 국졸, 3 = 중졸, 4 = 고졸, 5 = 대졸 로 바꾸기

```
> insurance = read.table("c:/Rfolder/data/insurance.txt", header=T)
> insurance$job = factor(insurance$job, levels=c(1:3),
                        labels=c("근로자","사무직","전문가"))
> insurance$edu2 = ordered(insurance$edu, levels=c(1:5),
                          labels=c("무학","국졸","중졸","고졸","대졸"))
> head(insurance)
```

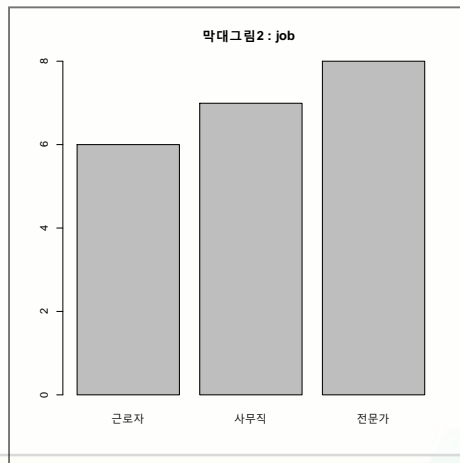
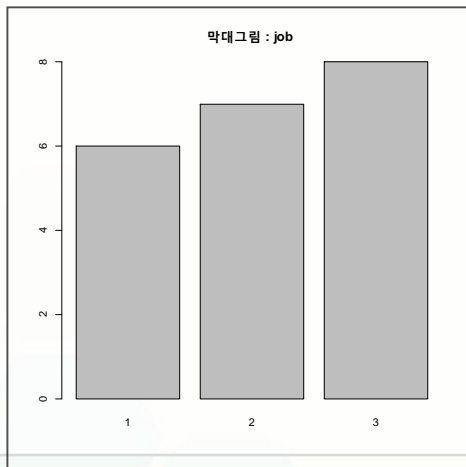
| id | sex | job | religion | edu | amount | salary | edu2   |
|----|-----|-----|----------|-----|--------|--------|--------|
| 1  | 1   | m   | 근로자      | 1   | 3      | 7.0    | 110 중졸 |
| 2  | 2   | m   | 사무직      | 1   | 4      | 12.0   | 135 고졸 |
| 3  | 3   | f   | 사무직      | 3   | 5      | 8.5    | 127 대졸 |
| 4  | 4   | f   | 전문가      | 3   | 5      | 5.0    | 150 대졸 |
| 5  | 5   | m   | 근로자      | 3   | 3      | 4.5    | 113 중졸 |
| 6  | 6   | m   | 사무직      | 1   | 2      | 3.5    | 95 국졸  |

- 명목형(nominal data) :  
factor() 함수
- 순서형(ordered data) :  
ordered() 함수

# 값 라벨 (Value labels)

## 예) 막대그림 그리기

```
> job.freq = table(insurance$job)
> barplot(job.freq)
> title("막대그림 : job ")
> insurance$job = factor(insurance$job, levels=c(1:3),
                        labels=c("근로자", "사무직", "전문가"))
> job.freq2 = table(insurance$job)
> barplot(job.freq2)
> title("막대그림2 : job ")
```



## 4 변수 값 변환(Recode)



# 변수 값 변환(recode)

예) 어느 제약제품의 약 구매 여부를 조사하였다. 나이별 구매내역은 다음과 같다.  
반응변수 purchase 0 = 구매 안함, 1 = 구매함이다. 케이스 수는 100개이다.

이 자료에서 변수 나이(age)의 값을 “40 이하 = 1, 41~60 = 2, 60보다 큰 값 = 3”으로 변환하여보자.

|    | A  | B   | C        |
|----|----|-----|----------|
| 1  | id | age | purchase |
| 2  |    | 1   | 20       |
| 3  |    | 2   | 23       |
| 4  |    | 3   | 24       |
| 5  |    | 4   | 25       |
| 6  |    | 5   | 26       |
| 7  |    | 6   | 27       |
| 8  |    | 7   | 27       |
| 9  |    | 8   | 28       |
| 10 |    | 9   | 29       |
| 11 |    | 10  | 29       |
| 12 |    | 11  | 30       |
| 13 |    | 12  | 30       |
| 14 |    | 13  | 30       |
| 15 |    | 14  | 30       |
| 16 |    | 15  | 32       |

```
> install.packages("xlsx")
> library(xlsx)
> drug = read.xlsx("c:/Rfolder/data/drug.xlsx", 1)
# Replace data in the field : Method 1
> drug$agr = drug$age
> drug$agr[drug$agr >= 20 & drug$agr <= 40 ] = 1
> drug$agr[drug$agr > 40 & drug$agr <= 60 ] = 2
> drug$agr[drug$agr > 60 ] = 3
> drug[c(1,20,40, 95),]
  id age purchase agr
1  1  20         0  1
20 20  34         0  1
40 40  41         0  2
95 95  61         1  3
```

# 변수 값 변환(recode)

예) car 패키지의 recode() 를 이용하는 예

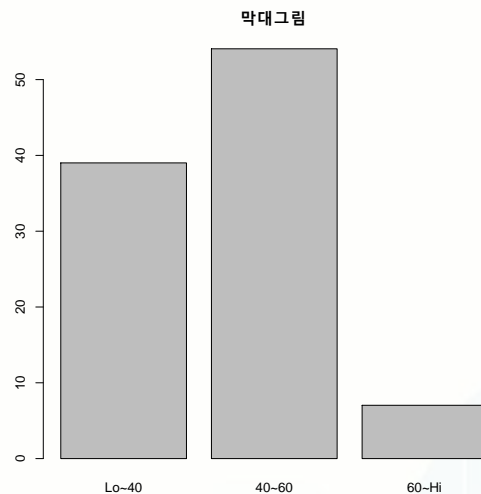
```
> # Use recode function in car package : Method 2
> install.packages("car")
> library(car)
> drug$agr2 = drug$age
> drug$agr2 = recode(drug$age, "lo:40=1; 40:60=2; 60:hi=3")
> drug[c(1,20,40, 80),]
```

|    | id | age | purchase | agr | agr2 |
|----|----|-----|----------|-----|------|
| 1  | 1  | 20  | 0        | 1   | 1    |
| 20 | 20 | 34  | 0        | 1   | 1    |
| 40 | 40 | 41  | 0        | 2   | 2    |
| 80 | 80 | 56  | 0        | 2   | 2    |

```
> drug$agr2 = ordered(drug$agr2, levels=c(1:3),
                      labels=c("Lo~40", "40~60", "60~Hi"))
> agr2.freq=table(drug$agr2)
> agr2.freq
```

| Lo~40 | 40~60 | 60~Hi |
|-------|-------|-------|
| 39    | 54    | 7     |

```
> barplot(agr2.freq, main="막대그림")
```



## 5 케이스 선택



# 케이스 선택

예) insurance 자료에서 다음과 같은 조건을 만족하는 케이스를 추출해보자.

- ① 성별 = m 인 경우,
- ② 성별 = f 이고, 직업 = 2인 경우,

```
> insurance = read.table("c:/Rfolder/data/insurance.txt", header=T)
```

```
> select1 = insurance[insurance$sex=='m',]
```

```
> head(select1, n=3)
```

|   | id | sex | job | religion | edu | amount | salary |     |
|---|----|-----|-----|----------|-----|--------|--------|-----|
| 1 | 1  | m   | 1   |          | 1   | 3      | 7.0    | 110 |
| 2 | 2  | m   | 2   |          | 1   | 4      | 12.0   | 135 |
| 5 | 5  | m   | 1   |          | 3   | 3      | 4.5    | 113 |

```
> select2 = insurance[which(insurance$sex=='f' & insurance$job==2),]
```

```
> head(select2, n=3)
```

|    | id | sex | job | religion | edu | amount | salary |
|----|----|-----|-----|----------|-----|--------|--------|
| 3  | 3  | f   | 2   | 3        | 5   | 8.5    | 127    |
| 9  | 9  | f   | 2   | 3        | 4   | 4.5    | 110    |
| 18 | 18 | f   | 2   | 1        | 5   | 5.0    | 110    |

# 케이스 선택

예) insurance 자료에서 다음과 같은 조건을 만족하는 케이스를 추출해보자.

③ 직업 = 3 이고, 월수입 = 140 이상인 경우

```
> select3 = insurance[which(insurance$job==3 & insurance$salary >= 140),]
```

```
> head(select3, n=3)
```

|    | id | sex | job | religion | edu | amount | salary |     |
|----|----|-----|-----|----------|-----|--------|--------|-----|
| 4  | 4  | f   | 3   |          | 3   | 5      | 5      | 150 |
| 14 | 14 | m   | 3   |          | 2   | 5      | 7      | 150 |

```
> select3 = insurance[insurance$job==3 & insurance$salary >= 140,]
```

```
> head(select3, n=3)
```

|    | id | sex | job | religion | edu | amount | salary |     |
|----|----|-----|-----|----------|-----|--------|--------|-----|
| 4  | 4  | f   | 3   |          | 3   | 5      | 5      | 150 |
| 14 | 14 | m   | 3   |          | 2   | 5      | 7      | 150 |



## 6 dplyr 패키지 활용(1)



# dplyr 패키지 소개

- **dplyr 패키지** : 데이터 처리를 위한 R 패키지
- **개발자** : Hadley Wickham
- **데이터 사전 처리 작업**
  - Filtering (데이터 추출)
  - Selecting columns (변수 선택)
  - Adding new variables (변수 추가)
  - Sorting (정렬)
  - Aggregating (집계)

※ **참고** : foreign" 패키지 (<http://cran.r-project.org/web/packages/plyr/dplyr.pdf>)



# (1) 데이터 추출 (Filtering)

- 실무에 사용되는 데이터는 변수의 수나 케이스가 크기 때문에 분석 전에 이들 내용을 살펴보는 것이 좋음.
- `dim()` 함수 : (케이스의 수, 변수의 수) 표시 ,
- `tbl_df()` 함수: 데이터 프레임. 효율적인 출력결과를 보여줌.

```
> library(dplyr)
> dim(insurance)
```

```
[1] 22 7
```

```
> tbl_df(insurance)
```

```
Source: local data frame [22 x 7]
```

|     | id  | sex | job | religion | edu | amount | salary |
|-----|-----|-----|-----|----------|-----|--------|--------|
| 1   | 1   | m   | 1   |          | 1 3 | 7.0    | 110    |
| 2   | 2   | m   | 2   |          | 1 4 | 12.0   | 135    |
| ... | ... | ... |     |          |     |        |        |
| 21  | 21  | m   | 3   |          | 1 4 | 10.0   | 95     |
| 22  | 22  | m   | 3   |          | 2 3 | 12.0   | 88     |

# (1) 데이터 추출 (Filtering)

예1) tbl\_df()

```
> library(nycflights13)
```

```
> dim(flights)
```

```
[1] 227496    21
```

```
> tbl_df(flights)
```

Source: local data frame [227,496 x 21]

|    | Year | Month | DayofMonth | DayOfWeek | DepTime | ArrTime | UniqueCarrier | FlightNum |
|----|------|-------|------------|-----------|---------|---------|---------------|-----------|
| 1  | 2011 | 1     | 1          | 6         | 1400    | 1500    | AA            | 428       |
| 2  | 2011 | 1     | 2          | 7         | 1401    | 1501    | AA            | 428       |
| 3  | 2011 | 1     | 3          | 1         | 1352    | 1502    | AA            | 428       |
| 4  | 2011 | 1     | 4          | 2         | 1403    | 1513    | AA            | 428       |
| 5  | 2011 | 1     | 5          | 3         | 1405    | 1507    | AA            | 428       |
| 6  | 2011 | 1     | 6          | 4         | 1359    | 1503    | AA            | 428       |
| 7  | 2011 | 1     | 7          | 5         | 1359    | 1509    | AA            | 428       |
| 8  | 2011 | 1     | 8          | 6         | 1355    | 1454    | AA            | 428       |
| 9  | 2011 | 1     | 9          | 7         | 1443    | 1554    | AA            | 428       |
| 10 | 2011 | 1     | 10         | 1         | 1443    | 1553    | AA            | 428       |

Variables not shown: TailNum (chr), ActualElapsedTime (int), AirTime (int),  
ArrDelay (int), DepDelay (int), Origin (chr), Dest (chr), Distance (int),  
TaxiIn (int), TaxiOut (int), Cancelled (int), CancellationCode (chr),  
Diverted (int)

```
>
```

# (1) 데이터 추출 (Filtering)

예2) sex = 'm'이고, edu = 3인 케이스를 추출하는 예 (AND 조건문은 콤마(,)나 & 연산자를 사용)

```
> sel_1 = filter(insurance, sex=='m', edu == 3)
> sel_1
```

|   | id | sex | job | religion | edu | amount | salary |
|---|----|-----|-----|----------|-----|--------|--------|
| 1 | 1  | m   | 1   | 1        | 3   | 7.0    | 110    |
| 2 | 5  | m   | 1   | 3        | 3   | 4.5    | 113    |
| 3 | 19 | m   | 2   | 3        | 3   | 7.0    | 85     |
| 4 | 22 | m   | 3   | 2        | 3   | 12.0   | 88     |

예3) sex = 'm' 또는 edu = 3인 케이스를 추출하는 예 (OR 조건문은 | 연산자를 사용)

```
> sel_2 = filter(insurance, sex=='m' | edu == 3)
> sel_2[c(1:3, 6:8),]
```

|   | id | sex | job | religion | edu | amount | salary |
|---|----|-----|-----|----------|-----|--------|--------|
| 1 | 1  | m   | 1   | 1        | 3   | 7.0    | 110    |
| 2 | 2  | m   | 2   | 1        | 4   | 12.0   | 135    |
| 3 | 5  | m   | 1   | 3        | 3   | 4.5    | 113    |
| 6 | 10 | m   | 1   | 3        | 5   | 17.0   | 200    |
| 7 | 11 | f   | 1   | 1        | 3   | 22.0   | NA     |
| 8 | 12 | m   | 2   | 1        | 2   | 5.5    | 105    |

## (2) 변수 선택

- 변수선택은 select() 사용

```
> head(insurance)
```

|   | id | sex | job | religion | edu | amount | salary |
|---|----|-----|-----|----------|-----|--------|--------|
| 1 | 1  | m   | 1   | 1        | 3   | 7.0    | 110    |
| 2 | 2  | m   | 2   | 1        | 4   | 12.0   | 135    |
| 3 | 3  | f   | 2   | 3        | 5   | 8.5    | 127    |
| 4 | 4  | f   | 3   | 3        | 5   | 5.0    | 150    |
| 5 | 5  | m   | 1   | 3        | 3   | 4.5    | 113    |
| 6 | 6  | m   | 2   | 1        | 2   | 3.5    | 95     |

```
> sel_3 = select(insurance, sex, job, amount, salary)
```

```
> head(sel_3, 3)
```

|   | sex | job | amount | salary |
|---|-----|-----|--------|--------|
| 1 | m   | 1   | 7.0    | 110    |
| 2 | m   | 2   | 12.0   | 135    |
| 3 | f   | 2   | 8.5    | 127    |

## (2) 변수 선택

- filter() 함수와 같이 사용 가능함.  
인접한 열을 추출할 때는 연산자 : 를 사용.

```
> sel_4 = filter(select(insurance, sex, job, amount, salary), job==1)
```

```
> head(sel_4, 3)
```

|   | sex | job | amount | salary |
|---|-----|-----|--------|--------|
| 1 | m   | 1   | 7.0    | 110    |
| 2 | m   | 1   | 4.5    | 113    |
| 3 | m   | 1   | 17.0   | 200    |

```
> sel_5 = select(insurance, job:salary)
```

```
> head(sel_5, 3)
```

|   | job | religion | edu | amount | salary |
|---|-----|----------|-----|--------|--------|
| 1 | 1   |          | 1 3 | 7.0    | 110    |
| 2 | 2   |          | 1 4 | 12.0   | 135    |
| 3 | 2   |          | 3 5 | 8.5    | 127    |

### (3) 변수 추가

- 새로운 변수를 만들어 추가하는 경우에 mutate() 함수가 이용됨.

```
> insu_add = mutate(insurance, amopersal = amount/salary)
```

```
> head(insu_add)
```

|   | id | sex | job | religion | edu | amount | salary | amopersal  |
|---|----|-----|-----|----------|-----|--------|--------|------------|
| 1 | 1  | m   | 1   |          | 3   | 7.0    | 110    | 0.06363636 |
| 2 | 2  | m   | 2   |          | 4   | 12.0   | 135    | 0.08888889 |
| 3 | 3  | f   | 2   |          | 5   | 8.5    | 127    | 0.06692913 |
| 4 | 4  | f   | 3   |          | 5   | 5.0    | 150    | 0.03333333 |
| 5 | 5  | m   | 1   |          | 3   | 4.5    | 113    | 0.03982301 |
| 6 | 6  | m   | 2   |          | 2   | 3.5    | 95     | 0.03684211 |

※ 참고 : cbind() 를 사용하는 경우

```
> amopersal = insurance$amount / insurance$salary
```

```
> insu_add2 = cbind(insurance, amopersal)
```



## (4) 정렬

- 크기순으로 정렬하기 위해서는 arrange() 함수가 이용됨.

예1) sex와 job을 기준으로 오름차순으로 정렬하는 예

```
> sel_3_arr = arrange(sel_3, sex, job)
```

```
> head(sel_3_arr,3)
```

|   | sex | job | amount | salary |
|---|-----|-----|--------|--------|
| 1 | f   | 1   | 22.0   | NA     |
| 2 | f   | 1   | 7.0    | 88     |
| 3 | f   | 2   | 8.5    | 127    |

예2) sex와 job을 기준으로 내림차순으로 정렬하는 예

```
> sel_3_arrd = arrange(sel_3, desc(sex), desc(job))
```

```
> head(sel_3_arrd,3)
```

|   | sex | job | amount | salary |
|---|-----|-----|--------|--------|
| 1 | m   | 3   | 4.0    | 102    |
| 2 | m   | 3   | 4.5    | 130    |
| 3 | m   | 3   | 7.0    | 150    |

## (5) 요약결과

- 데이터를 정렬한 뒤, 그룹별 요약 결과 보기 : `group_by()` 함수, `summarise()` 함수

예) 변수 `job` 에 따라 `amount` 와 `salary` 의 평균을 보기

```
> ins_job = arrange(insurance, job)
> ins_job_g = group_by(ins_job, job)
> ins_job_gm = summarize(ins_job_g, amMean=mean(amount), aSal=mean(salary))
> ins_job_gm
```

Source: local data frame [4 x 3]

|   | job | amMean    | aSal     |
|---|-----|-----------|----------|
| 1 | 1   | 10.583333 | NA       |
| 2 | 2   | 6.571429  | 109.5714 |
| 3 | 3   | 7.000000  | 118.3750 |
| 4 | NA  | 6.000000  | 110.0000 |

## (6) chain 기능

- 여러 명령을 연결해서 한번에 수행할 수 있도록 하는 기능
- 연결 기능은 %>% (“then” 즉, “그리고”로 명명)로 이루어짐.

예) insurance 자료에서 amount, salary 두 변수를 선택하여, salary > 130 이상인 케이스를 추출하는 예

```
> b1 = select(insurance, amount, salary)
```

```
> b2 = filter(b1, salary > 130)
```

```
> b2
```

|   | amount | salary |
|---|--------|--------|
| 1 | 12     | 135    |
| 2 | 5      | 150    |
| 3 | 17     | 200    |
| 4 | 7      | 150    |
| 5 | 6      | 138    |

```
> nb2 = insurance %>%
```

```
  select(amount, salary) %>%
```

```
  filter(salary > 130 )
```

```
> nb2
```

|   | amount | salary |
|---|--------|--------|
| 1 | 12     | 135    |
| 2 | 5      | 150    |
| 3 | 17     | 200    |
| 4 | 7      | 150    |
| 5 | 6      | 138    |

## 7 dplyr 패키지 활용(2)

Ref : <https://rpubs.com/justmarkham/dplyr-tutorial>



# Loading dplyr and an example dataset (1)

```
> library(dplyr)
> library(hflights)
> # hflights is flights from two Houston airports in 2011
> data(hflights)
> head(hflights)
```

|      | Year | Month | DayOfMonth | DayOfWeek | DepTime | ArrTime | UniqueCarrier | FlightNum |
|------|------|-------|------------|-----------|---------|---------|---------------|-----------|
| 5424 | 2011 | 1     | 1          | 6         | 1400    | 1500    | AA            | 428       |
| 5425 | 2011 | 1     | 2          | 7         | 1401    | 1501    | AA            | 428       |
| 5426 | 2011 | 1     | 3          | 1         | 1352    | 1502    | AA            | 428       |
| 5427 | 2011 | 1     | 4          | 2         | 1403    | 1513    | AA            | 428       |
| 5428 | 2011 | 1     | 5          | 3         | 1405    | 1507    | AA            | 428       |
| 5429 | 2011 | 1     | 6          | 4         | 1359    | 1503    | AA            | 428       |

|      | TailNum | ActualElapsedTime | AirTime | ArrDelay | DepDelay | Origin | Dest | Distance |
|------|---------|-------------------|---------|----------|----------|--------|------|----------|
| 5424 | N576AA  | 60                | 40      | -10      | 0        | IAH    | DFW  | 224      |
| 5425 | N557AA  | 60                | 45      | -9       | 1        | IAH    | DFW  | 224      |
| 5426 | N541AA  | 70                | 48      | -8       | -8       | IAH    | DFW  | 224      |
| 5427 | N403AA  | 70                | 39      | 3        | 3        | IAH    | DFW  | 224      |
| 5428 | N492AA  | 62                | 44      | -3       | 5        | IAH    | DFW  | 224      |
| 5429 | N262AA  | 64                | 45      | -7       | -1       | IAH    | DFW  | 224      |

|      | TaxiIn | TaxiOut | Cancelled | CancellationCode | Diverted |
|------|--------|---------|-----------|------------------|----------|
| 5424 | 7      | 13      | 0         |                  | 0        |
| 5425 | 6      | 9       | 0         |                  | 0        |
| 5426 | 5      | 17      | 0         |                  | 0        |
| 5427 | 9      | 22      | 0         |                  | 0        |
| 5428 | 9      | 9       | 0         |                  | 0        |
| 5429 | 6      | 13      | 0         |                  | 0        |

# Loading dplyr and an example dataset (2)

```
> flights = tbl_df(hflights)
```

```
> flights
```

```
Source: local data frame [227,496 x 21]
```

|    | Year | Month | DayofMonth | DayOfWeek | DepTime | ArrTime | UniqueCarrier | FlightNum | TailNum |
|----|------|-------|------------|-----------|---------|---------|---------------|-----------|---------|
| 1  | 2011 | 1     | 1          | 6         | 1400    | 1500    | AA            | 428       | N576AA  |
| 2  | 2011 | 1     | 2          | 7         | 1401    | 1501    | AA            | 428       | N557AA  |
| 3  | 2011 | 1     | 3          | 1         | 1352    | 1502    | AA            | 428       | N541AA  |
| 4  | 2011 | 1     | 4          | 2         | 1403    | 1513    | AA            | 428       | N403AA  |
| 5  | 2011 | 1     | 5          | 3         | 1405    | 1507    | AA            | 428       | N492AA  |
| 6  | 2011 | 1     | 6          | 4         | 1359    | 1503    | AA            | 428       | N262AA  |
| 7  | 2011 | 1     | 7          | 5         | 1359    | 1509    | AA            | 428       | N493AA  |
| 8  | 2011 | 1     | 8          | 6         | 1355    | 1454    | AA            | 428       | N477AA  |
| 9  | 2011 | 1     | 9          | 7         | 1443    | 1554    | AA            | 428       | N476AA  |
| 10 | 2011 | 1     | 10         | 1         | 1443    | 1553    | AA            | 428       | N504AA  |

```
... ..  
Variables not shown: ActualElapsedTime (int), AirTime (int), ArrDelay (int),  
DepDelay  
  (int), Origin (chr), Dest (chr), Distance (int), TaxiIn (int), TaxiOut (int),  
  Cancelled (int), CancellationCode (chr), Diverted (int)
```

※ tbl\_df creates a “local data frame”

※ Local data frame is simply a wrapper for a data frame that prints nicely

# Filter : Keep rows matching criteria(1)

```
> # base R approach to view all flights on January 1  
> flights[flights$Month==1 & flights$DayofMonth==1, ]
```

```
> # dplyr approach  
> # note: you can use comma or &(ampersand) to represent AND condition  
> filter(flights, Month==1, DayofMonth==1)
```

Source: local data frame [552 x 21]

|    | Year | Month | DayofMonth | DayOfWeek | DepTime | ArrTime | UniqueCarrier | FlightNum | TailNum |
|----|------|-------|------------|-----------|---------|---------|---------------|-----------|---------|
| 1  | 2011 | 1     | 1          | 6         | 1400    | 1500    | AA            | 428       | N576AA  |
| 2  | 2011 | 1     | 1          | 6         | 728     | 840     | AA            | 460       | N520AA  |
| 3  | 2011 | 1     | 1          | 6         | 1631    | 1736    | AA            | 1121      | N4WVAA  |
| 4  | 2011 | 1     | 1          | 6         | 1756    | 2112    | AA            | 1294      | N3DGAA  |
| 5  | 2011 | 1     | 1          | 6         | 1012    | 1347    | AA            | 1700      | N3DAAA  |
| 6  | 2011 | 1     | 1          | 6         | 1211    | 1325    | AA            | 1820      | N593AA  |
| 7  | 2011 | 1     | 1          | 6         | 557     | 906     | AA            | 1994      | N3BBAA  |
| 8  | 2011 | 1     | 1          | 6         | 1824    | 2106    | AS            | 731       | N614AS  |
| 9  | 2011 | 1     | 1          | 6         | 654     | 1124    | B6            | 620       | N324JB  |
| 10 | 2011 | 1     | 1          | 6         | 1639    | 2110    | B6            | 622       | N324JB  |

.. ..  
Variables not shown: ActualElapsedTime (int), AirTime (int), ArrDelay (int), DepDelay (int), Origin (chr), Dest (chr), Distance (int), TaxiIn (int), TaxiOut (int), Cancelled (int), CancellationCode (chr), Diverted (int)

# Filter : Keep rows matching criteria(2)

```
> # use pipe for OR condition  
> filter(flights, UniqueCarrier=="AA" | UniqueCarrier=="UA")
```

Source: local data frame [5,316 x 21]

|    | Year | Month | DayofMonth | DayOfWeek | DepTime | ArrTime | UniqueCarrier | FlightNum | TailNum |
|----|------|-------|------------|-----------|---------|---------|---------------|-----------|---------|
| 1  | 2011 | 1     | 1          | 6         | 1400    | 1500    | AA            | 428       | N576AA  |
| 2  | 2011 | 1     | 2          | 7         | 1401    | 1501    | AA            | 428       | N557AA  |
| 3  | 2011 | 1     | 3          | 1         | 1352    | 1502    | AA            | 428       | N541AA  |
| 4  | 2011 | 1     | 4          | 2         | 1403    | 1513    | AA            | 428       | N403AA  |
| 5  | 2011 | 1     | 5          | 3         | 1405    | 1507    | AA            | 428       | N492AA  |
| 6  | 2011 | 1     | 6          | 4         | 1359    | 1503    | AA            | 428       | N262AA  |
| 7  | 2011 | 1     | 7          | 5         | 1359    | 1509    | AA            | 428       | N493AA  |
| 8  | 2011 | 1     | 8          | 6         | 1355    | 1454    | AA            | 428       | N477AA  |
| 9  | 2011 | 1     | 9          | 7         | 1443    | 1554    | AA            | 428       | N476AA  |
| 10 | 2011 | 1     | 10         | 1         | 1443    | 1553    | AA            | 428       | N504AA  |

... ..  
Variables not shown: ActualElapsedTime (int), AirTime (int), ArrDelay (int),  
DepDelay  
(int), Origin (chr), Dest (chr), Distance (int), TaxiIn (int), TaxiOut (int),  
Cancelled (int), CancellationCode (chr), Diverted (int)



# select : Pick columns by name(1)

```
> # base R approach to select DepTime, ArrTime, and FlightNum columns  
> flights[, c("DepTime", "ArrTime", "FlightNum")]
```

```
> # dplyr approach  
> select(flights, DepTime, ArrTime, FlightNum)
```

Source: local data frame [227,496 x 3]

|    | DepTime | ArrTime | FlightNum |
|----|---------|---------|-----------|
| 1  | 1400    | 1500    | 428       |
| 2  | 1401    | 1501    | 428       |
| 3  | 1352    | 1502    | 428       |
| 4  | 1403    | 1513    | 428       |
| 5  | 1405    | 1507    | 428       |
| 6  | 1359    | 1503    | 428       |
| 7  | 1359    | 1509    | 428       |
| 8  | 1355    | 1454    | 428       |
| 9  | 1443    | 1554    | 428       |
| 10 | 1443    | 1553    | 428       |
| .. | ...     | ...     | ...       |



# Chaining

- Usual way to perform multiple operations in one line is by nesting
- Can write commands in a natural order by using the %>% infix operator (which can be pronounced as “then”)

```
> a1 = select(flights, UniqueCarrier, DepDelay)
> a2 = filter(a1, DepDelay > 60)
```

```
> na2 = flights %>%
  select(UniqueCarrier, DepDelay) %>%
  filter(DepDelay > 60)
```

```
> na2
```

Source: local data frame [10,242 x 2]

|   | UniqueCarrier | DepDelay |
|---|---------------|----------|
| 1 | AA            | 90       |
| 2 | AA            | 67       |
| 3 | AA            | 74       |
| 4 | AA            | 125      |
| 5 | AA            | 82       |
| 6 | AA            | 99       |
| 7 | AA            | 70       |
| 8 | AA            | 61       |

# arrange : Reorder rows

```
> # base R approach to select UniqueCarrier and DepDelay columns and sort by DepDelay  
> flights[order(flights$DepDelay), c("UniqueCarrier", "DepDelay")]
```

```
> flights %>%  
  select(UniqueCarrier, DepDelay) %>%  
  arrange(DepDelay)
```

Source: local data frame [227,496 x 2]

|    | UniqueCarrier | DepDelay |
|----|---------------|----------|
| 1  | OO            | -33      |
| 2  | MQ            | -23      |
| 3  | XE            | -19      |
| 4  | XE            | -19      |
| 5  | CO            | -18      |
| 6  | EV            | -18      |
| 7  | XE            | -17      |
| 8  | CO            | -17      |
| 9  | XE            | -17      |
| 10 | MQ            | -17      |
| .. | ...           | ...      |

```
# use `desc` for descending  
flights %>%  
  select(UniqueCarrier, DepDelay) %>%  
  arrange(desc(DepDelay))
```

# mutate : Add new variables

```
> # base R approach to create a new variable Speed (in mph)
> flights$Speed <- flights$Distance / flights$AirTime*60
> flights[, c("Distance", "AirTime", "Speed")]
```

```
> flights %>%
  select(Distance, AirTime) %>%
  mutate(Speed = Distance/AirTime*60)
```

Source: local data frame [227,496 x 3]

|    | Distance | AirTime | Speed    |
|----|----------|---------|----------|
| 1  | 224      | 40      | 336.0000 |
| 2  | 224      | 45      | 298.6667 |
| 3  | 224      | 48      | 280.0000 |
| 4  | 224      | 39      | 344.6154 |
| 5  | 224      | 44      | 305.4545 |
| 6  | 224      | 45      | 298.6667 |
| 7  | 224      | 43      | 312.5581 |
| 8  | 224      | 40      | 336.0000 |
| 9  | 224      | 41      | 327.8049 |
| 10 | 224      | 45      | 298.6667 |
| .. | ...      | ...     | ...      |

# summarise : Reduce variables to values(1)

- group\_by : creates the groups that will be operated on
- summarise : uses the provided aggregation function to summarise each group

```
> flights %>%  
  group_by(Dest) %>%  
  summarise(avg_delay = mean(ArrDelay, na.rm=TRUE))
```

Source: local data frame [116 x 2]

|    | Dest | avg_delay  |
|----|------|------------|
| 1  | ABQ  | 7.226259   |
| 2  | AEX  | 5.839437   |
| 3  | AGS  | 4.000000   |
| 4  | AMA  | 6.840095   |
| 5  | ANC  | 26.080645  |
| 6  | ASE  | 6.794643   |
| 7  | ATL  | 8.233251   |
| 8  | AUS  | 7.448718   |
| 9  | AVL  | 9.973988   |
| 10 | BFL  | -13.198807 |
| .. | ...  | ...        |

# summarise : Reduce variables to values(2)

- summarise\_each : allows you to apply the same summary function to multiple columns at once

```
> flights %>%  
  group_by(UniqueCarrier) %>%  
  summarise_each(funs(mean), Cancelled, Diverted)
```

Source: local data frame [15 x 3]

|    | UniqueCarrier | Cancelled   | Diverted    |
|----|---------------|-------------|-------------|
| 1  | AA            | 0.018495684 | 0.001849568 |
| 2  | AS            | 0.000000000 | 0.002739726 |
| 3  | B6            | 0.025899281 | 0.005755396 |
| 4  | CO            | 0.006782614 | 0.002627370 |
| 5  | DL            | 0.015903067 | 0.003029156 |
| 6  | EV            | 0.034482759 | 0.003176044 |
| 7  | F9            | 0.007159905 | 0.000000000 |
| 8  | FL            | 0.009817672 | 0.003272557 |
| 9  | MQ            | 0.029044750 | 0.001936317 |
| 10 | OO            | 0.013946828 | 0.003486707 |
| 11 | UA            | 0.016409266 | 0.002413127 |
| 12 | US            | 0.011268986 | 0.001469868 |
| 13 | WN            | 0.015504047 | 0.002293629 |
| 14 | XE            | 0.015495599 | 0.003449550 |
| 15 | YV            | 0.012658228 | 0.000000000 |