
AN
INTRODUCTION
TO
MACHINE
LEARNING
WITH R
DAY 2


DAY 2
Data Handling

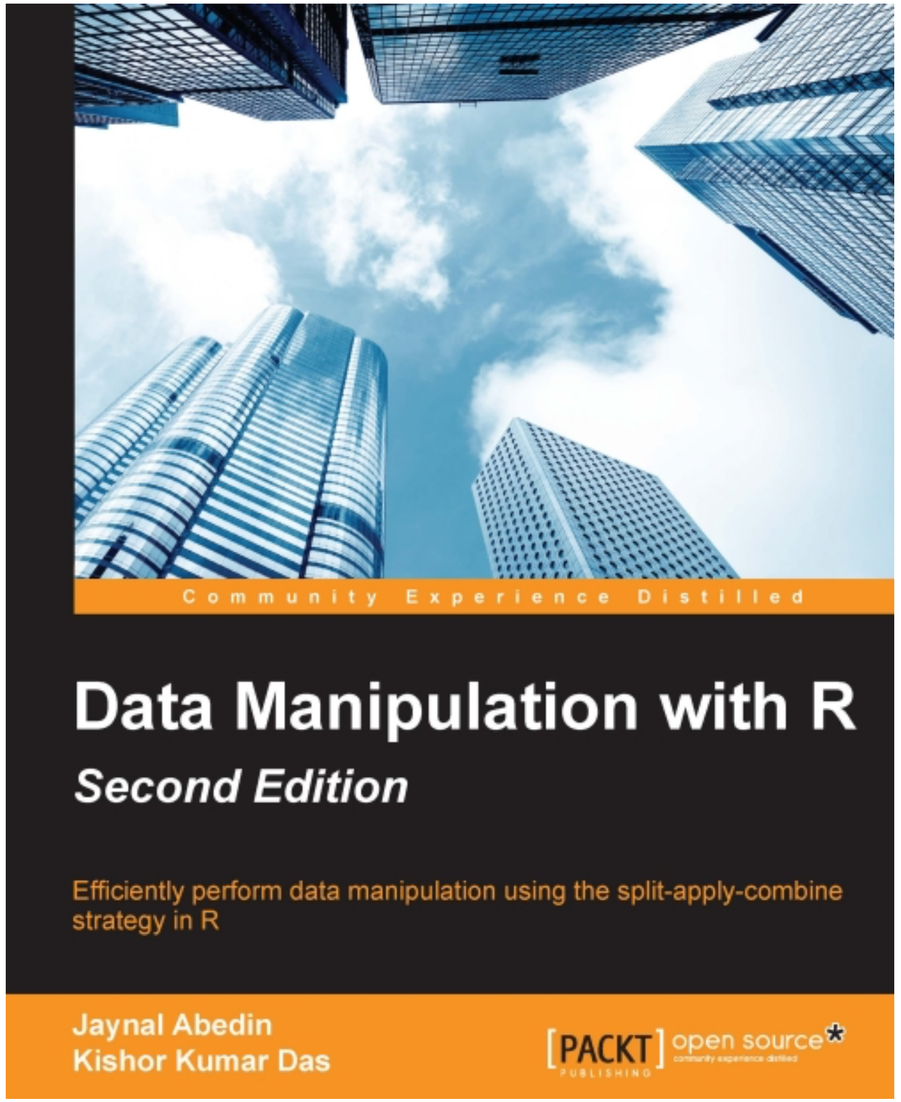

데이터
핸들링

Data Handling

R이 가장 강력한 힘을 보여주는 영역

Data Handling

데이터를 **원하는 형태**로 만들고 수정하는 일련의 작업



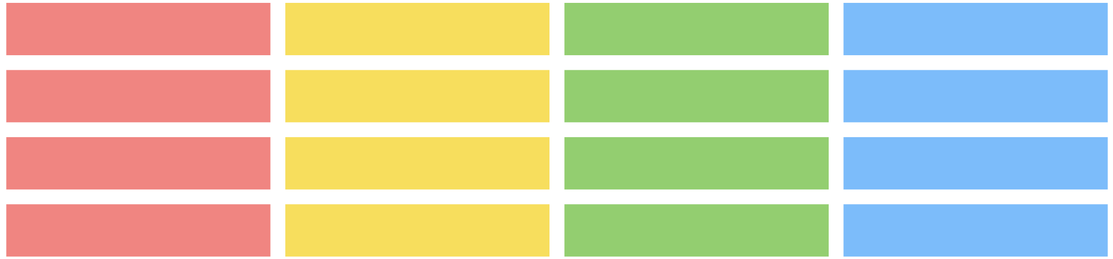
Data Manipulation with R, 2nd Edition
PACKT Publishing

실습

apply 함수

큰 데이터를
일괄적으로 처리하기

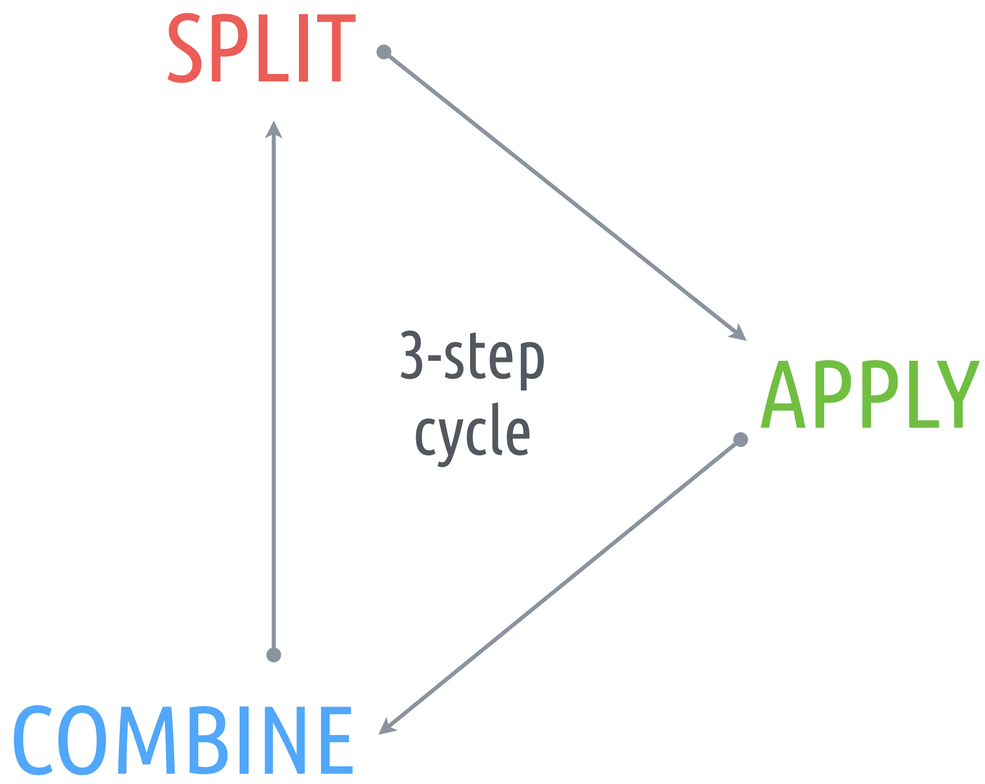
Basic Process



SPLIT

APPLY

COMBINE


SPLIT
APPLY
COMBINE
3-step
cycle

APPLY

APPLY FUNCTIONS

데이터에 원하는 함수를 **일괄적**으로 적용해 결과를 얻을 때 사용하는 함수

APPLY

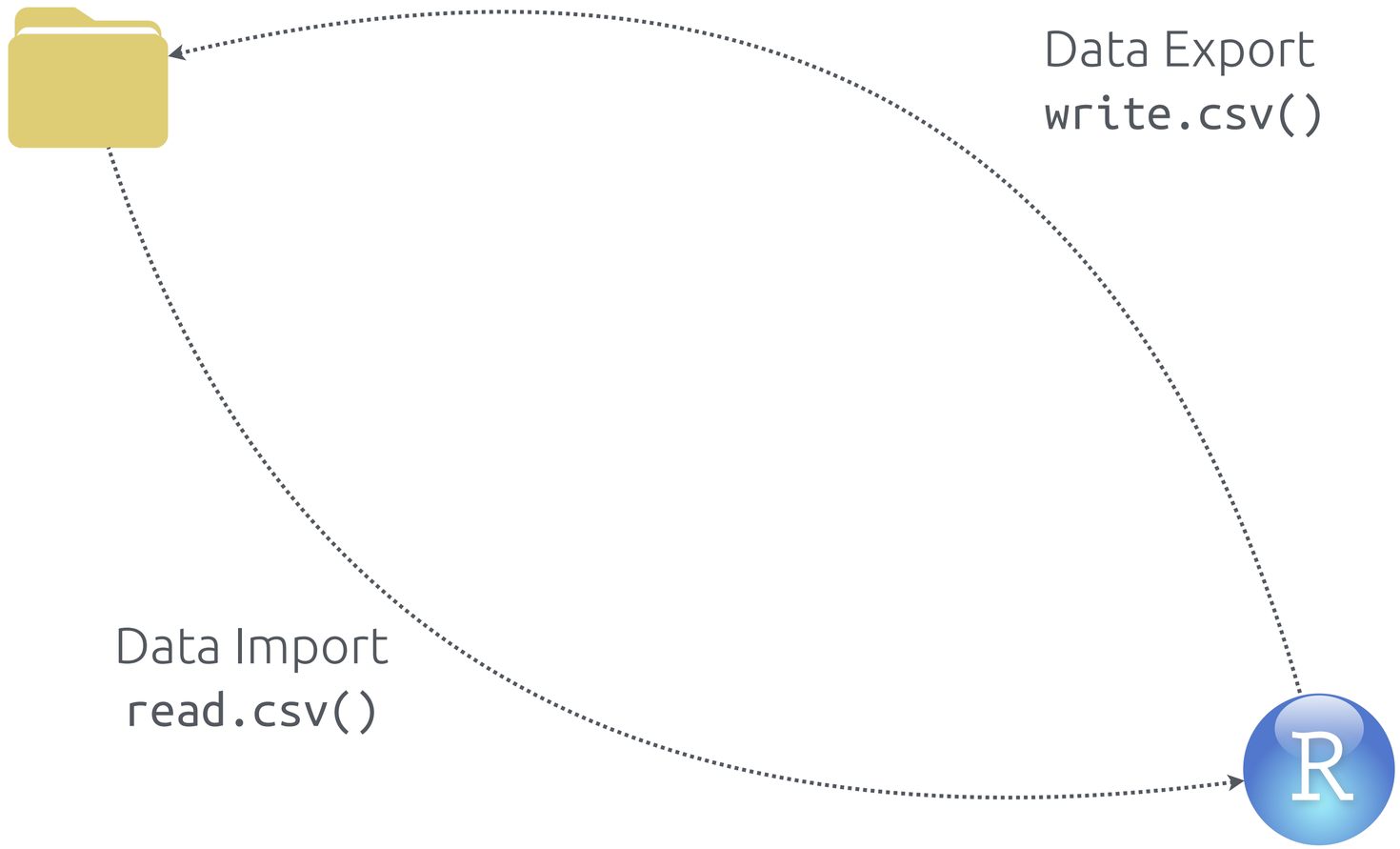
APPLY FUNCTIONS

데이터에 원하는 함수를 **일괄적**으로 적용해 결과를 얻을 때 사용하는 함수

apply	lapply	sapply	tapply

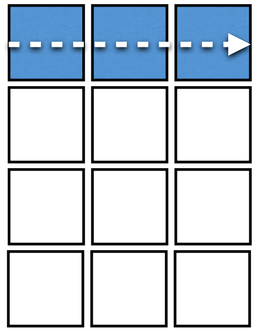
Iris Data

Data I/O

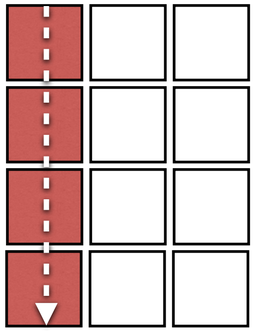


```
apply(data, margin, function, …)
```

데이터 나누는 방향 적용할 함수



margin = 1



margin = 2

DATA FRAME

MATRIX

ARRAY

apply()

VECTOR

LIST

ARRAY

```
lapply(data, function, …)
```

데이터 적용할 함수

```
sapply(data, function, …)
```

데이터 적용할 함수

dplyr

데이터 핸들링을 위한 툴박스

Toolbox for Data Handling

dplyr


패키지
저장소
install.packages()
패키지 설치 요청
library()
패키지 불러오기
Download
PC

DPLYR

FILTER

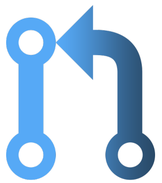
SELECT

ARRANGE

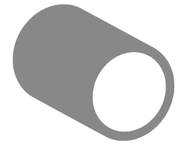
MUTATE

GROUP BY

SUMMARISE



JOIN

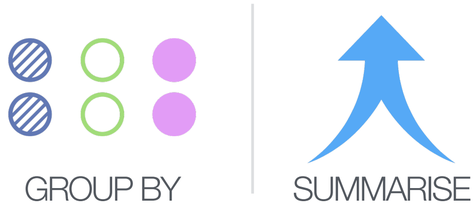


PIPE

실습

dplyr

2013년
뉴욕 항공 데이터



No.	Region	Nationality	Population
1	Europe	England	53,800,000
2	Europe	Germany	80,680,000
3	Asia	Korea	50,000,000
4	Asia	Japan	126,3000,000
5	N. America	Canada	36,200,000
6	N. America	USA	324,000,000

Population

GROUP BY

SUMMARISE

No.	Region	Nationality	Population
1	Europe	England	53,800,000
2	Europe	Germany	80,680,000
3	Asia	Korea	50,000,000
4	Asia	Japan	126,300,000
5	N. America	Canada	36,200,000
6	N. America	USA	324,000,000

Population

No.	Region	Nationality	Population
1	Europe	England	53,800,000
2	Europe	Germany	80,680,000

No.	Region	Nationality	Population
3	Asia	Korea	50,000,000
4	Asia	Japan	126,300,000

No.	Region	Nationality	Population
5	N. America	Canada	36,200,000
6	N. America	USA	324,000,000

No.	Region	Nationality	Population
1	Europe	England	53,800,000
2	Europe	Germany	80,680,000

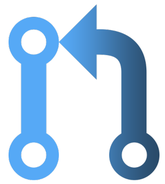
No.	Region	Nationality	Population
3	Asia	Korea	50,000,000
4	Asia	Japan	126,300,000

No.	Region	Nationality	Population
5	N. America	Canada	36,200,000
6	N. America	USA	324,000,000

No.	Region	Population Average
1	Europe	67,240,000
2	Asia	88,150,000
3	N.America	180,100,000

No.	Region	Population Average
1	Europe	67,240,000
2	Asia	88,150,000
3	N.America	180,100,000

Summarized_Population



JOIN

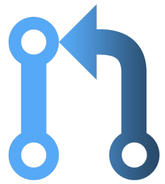
No.	Region	Nationality
1	Europe	England
2	Europe	Germany
3	Europe	France
4	Europe	Italy

Nation

+

Nationality	Capital
Denmark	Copenhagen
England	London
France	Paris
Germany	Berlin
Hungary	Budapest
Italy	Rome

Capital



JOIN

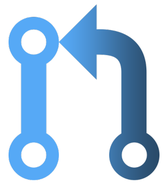
No.	Region	Nationality
1	Europe	England
2	Europe	Germany
3	Europe	France
4	Europe	Italy

Nation

+

Nationality	Capital
Denmark	Copenhagen
England	London
France	Paris
Germany	Berlin
Hungary	Budapest
Italy	Rome

Capital



JOIN

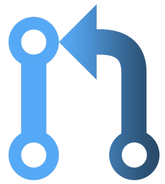
No.	Region	Nationality
1	Europe	England
2	Europe	Germany
3	Europe	France
4	Europe	Italy

Nation

+

Nationality	Capital
Denmark	Copenhagen
England	London
France	Paris
Germany	Berlin
Hungary	Budapest
Italy	Rome

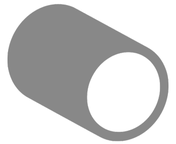
Capital



JOIN

No.	Region	Nationality	Capital
1	Europe	England	London
2	Europe	Germany	Berlin
3	Europe	France	Paris
4	Europe	Italy	Rome

Nation_Capital

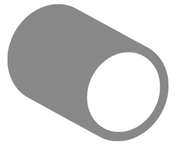


PIPE

%>%

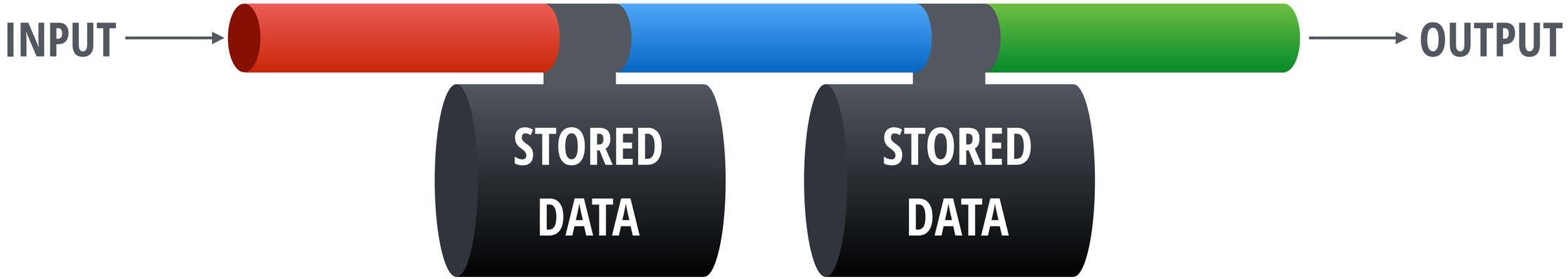
PIPE OPERATOR

일련의 데이터 핸들링 과정을
읽기 쉽고 더 빠르게
작성할 수 있도록 돕는다.

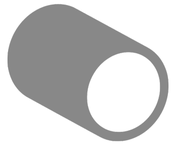


PIPE

WITHOUT **PIPE OPERATOR**

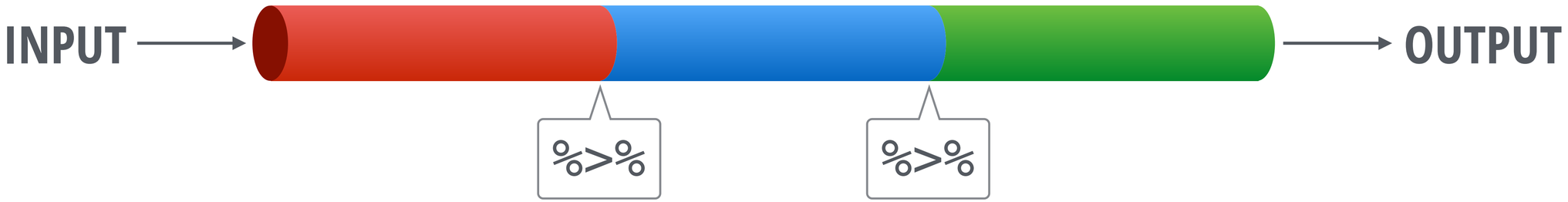


저장된 데이터들이 **메모리를 잡아먹는다.**

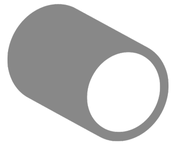


PIPE

WITH **PIPE OPERATOR**

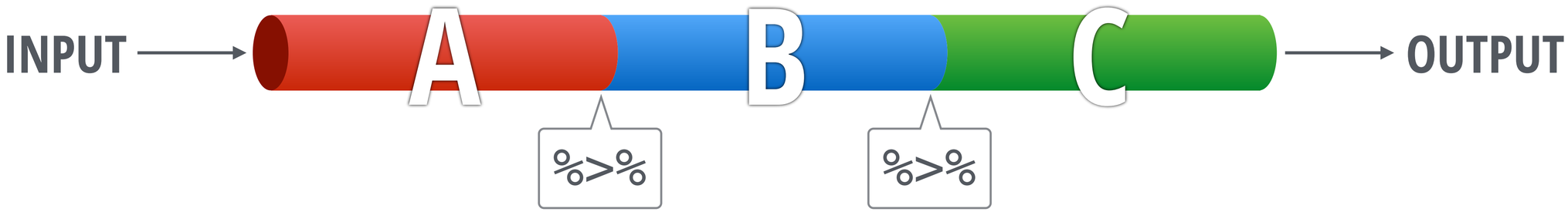


중간에 데이터 저장이 없기 때문에
메모리 관점에서 효율적이다.



PIPE

WITH **PIPE OPERATOR**



reshape2

데이터 레이아웃 바꾸기

reshape2

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	82

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77
B	영어	82

reshape2

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	82

WIDE LAYOUT

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77
B	영어	82

LONG LAYOUT

Layouts

WIDE LAYOUT

하나의 행에 **여러 데이터**가 포함되어 있다.
각각의 칼럼이 각각의 변수를 의미하지 않는다.

Layouts

하나의 행에는 **하나의 관측치**만 포함되어 있다.

각 변수는 개별의 칼럼으로 존재한다.

Basic Concepts of reshape2

데이터를 **녹여서** 원하는 모양의 **거푸집에 붓는** 과정

Melting

```
melt(data, id.vars, measure.vars,
     variable.name = "name",
     na.rm = FALSE, factorsAsStrings = TRUE)
```

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	82

학생	과목	점수
A	국어	80

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	82

학생	과목	점수
A	국어	80
B	국어	68

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	82

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	82

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	82

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	82

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77
B	영어	82

Casting

```
dcast(data, formula, …)
```

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77
B	영어	82

학생	국어 점수	수학 점수	영어 점수
A	80		
B			

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77
B	영어	82

학생	국어 점수	수학 점수	영어 점수
A	80		
B	68		

MELT

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77
B	영어	82

학생	국어 점수	수학 점수	영어 점수
A	80	72	
B	68		

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77
B	영어	82

학생	국어 점수	수학 점수	영어 점수
A	80	72	
B	68	94	

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77
B	영어	82

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	

학생	과목	점수
A	국어	80
B	국어	68
A	수학	72
B	수학	94
A	영어	77
B	영어	82

학생	국어 점수	수학 점수	영어 점수
A	80	72	77
B	68	94	82

실습

reshape2

뉴욕 대기질 데이터

과제

다이아몬드 가격

dplyr 패키지를 이용해서
데이터를 핸들링해보자.

과제

다이아몬드 가격

dplyr 패키지를 이용해서
데이터를 핸들링해보자.

모든 결과물은
파이프 연산자를 활용해야 한다.

데이터명 : `diamonds`

1. 다이아몬드의 투명도가 `I1` 인것을 제외
2. 칼럼명이 `x, y, z` 인 칼럼 제외
3. 기존 데이터의 가격은 달러 기준이므로 원화로 바꿔준다. 환율은 1170원.
4. 캐럿 기준으로 오름차순 정렬
5. 커팅 수준과 색깔, 투명도로 데이터를 그룹화하여 각 그룹의 원화 평균을 산출


THX :)