데이터 전처리

데이터 구조 선택

- 분석을 위해 수집한 데이터를 실제 사용 가능한 데이터로 만들기 위해서는 많은 사전작업이 필요함
  - **데이터 전처리** : 데이터 분석에 사용할 수 있는 적합한 데이터를 만들기 위한 사전 작업
- R환경에서 어떠한 데이터 구조로 표현할 것인지 결정해야함(벡터, 행렬, 리스트, 데이터프레임 등)
  - 하나의 차원만 갖고 있다 : 벡터
  - 2개 이상의 차원으로 구성되어 있다 : 행렬/배열/리스트/데이터프레임
    - 행렬이나 배열은 데이터가 모두 동일한 데이터 유형으로 구성되어있는경우
  - 데이터가 서로 다른 유형으로 구성되어 있으면서 테이블 형태로 표현 가능하다면 데이터프레임
    - 어떤 유형의 객체든 수용할 수 있음
    - 가장 널리 사용됨
    - 데이터프레임으로 표현할 수 없으면 리스트가 적합

데이터 유형 및 구조 변환

데이터 유형 변환

- 모든 객체는 Mode를 가짐(메모리에 물리적으로 저장되는 형태)
  - 숫자, 문자, 논리, 리스트, 함수
  - 하나의 모드만을 가질 수 있음
  - mode() 함수 : 인수로 주어진 객체의 모드 반환
- 벡터는 모든 원소가 동일한 모드여야 함
  - 리스트의 원소는 서로 다른 모드를 가질 수 있음
- 모든 객체는 모드와 별도로 클래스라는 추상적 형태를 가짐
  - R은 클래스에 따라 객체를 처리하는 방법을 결정
    - print : 객체의 클래스를 확인한 후 그에 맞는 적당한 하위 print함수 호출
- 문자형/숫자형 -> 다른 데이터 유형
  - 데이터 유형별 변환 함수 이용
  - as.character(x), as.complex(x), as.numeric(x), as.integer(x), as.logical(x)
    - 변환 불가능시 NA 출력
    - 벡터가 인수로 주어지면 벡터 연산 수행
    - 논리값은 숫자로 변환할 수 있음(TRUE = 1, FALSE = 0)

데이터 구조 변환

- 데이터 전처리 작업 중 빈번히 발생
- as.data.frame(x), as.list(x), as.matrix(x), as.vector(x), as.factor(x)
  - 확인 : is.data.frame(x) 등

벡터에서 변환

- as.list(벡터) : 벡터의 각 원소가 리스트의 원소로 변환
    - list(벡터) 함수 이용시 **벡터 전체를 하나의 원소**로 갖는 리스트가 생성
- 행렬
    - 하나의 열로 구성된 행렬로 변환 : cbind(벡터) / as.matrix(벡터)
    - 하나의 행으로 구성된 행렬로 변환 : rbind(벡터)
    - 행과 열의 개수를 지정하여 벡터로부터 행렬을 생성하려면 matrix()함수
- 데이터프레임
    - 하나의 열로 구성 : as.data.frame(벡터)
    - 하나의 행으로 구성 : rbind()함수로 행렬로 변환한 후 as.data.frame() 함수 이용

행렬에서 변환

- 행렬에 as.vector(), as.list(), as.data.frame() 함수 적용

리스트에서 변환

- 벡터로 변환 : unlist()
    - 리스트의 원소에 이름이 있을 때는 이름을 갖는 벡터로 변환
    - as.vector()는 아무런 변화가 없음
    - 리스트에 숫자와 문자가 섞여있으면 모두 문자로 변환됨
- 행렬로 변환 : unlist()로 벡터로 변환한 뒤, 벡터를 행렬로 변환
    - 리스트의 원소 이름을 행렬의 행 또는 열의 이름으로 사용하려면 dimnames인수 추가로 지정
    - matrix()나 as.matrix()함수를 적용하면 변환된 결과를 얻을 수 없음
- 데이터프레임으로 변환 : as.data.frame() 함수 이용
    - 숫자와 문자가 섞여있을 시 숫자는 숫자 벡터로, 문자는 팩터로 변환

데이터프레임에서 변환

- 벡터를 생성 : 데이터프레임 인덱싱 이용
    - 하나의 행을 추출할 시 데이터프레임으로 반환, 열을 추출하면 벡터로 반환
- 전체를 벡터로 변환 : 데이터프레임이 하나의 행 또는 열로 구성되어있을때만 유효
    - as.matrix() 함수로 행렬로 변환한 후, as.vector()함수로 벡터로 변환
    - 데이터프레임의 원소가 모두 숫자이거나 모두 문자일때에만 의미가있음
    - 문자와 숫자가 섞여있으면 모두 문자로 변환
- 리스트로 변환 : as.list()
    - 숫자와 문자가 섞여있으면 숫자는 벡터, 문자는 팩터로 변환
- 행렬로 변환 : as.matrix()
    - 숫자로만 구성되어 있으면 숫자로 구성된 행렬을 얻음
    - 문자로만 구성되어 있으면 문자로 구성된 행렬을 얻음
    - 섞여있으면 모든 값들은 문자로 변환되어 **문자로 구성된 행렬**을 얻음

결합

- 다른 곳에 저장된 데이터를 서로 합쳐야 할 때
- 실습 부분 책 + 정오표 참조

열 결합

- 결합하고자 하는 데이터셋이 동일한 행으로 구성되어 있을 때, **수평으로 결합**
- data.frame(), cbind() 함수로 기준이 되는 데이터셋의 **오른쪽에 열을 추가**하는 형태로 데이터프레임/행렬을 결합

행 결합

- 결합하고자 하는 데이터셋이 동일한 열을 갖고 있을 때, **수직으로 결합**
- rbind()함수로 기준이 되는 데이터셋의 **아래에 행을 추가**하는 형태로 데이터프레임/행렬을 결합

공통 열에 의한 결합

- 결합하고자 하는 데이터셋이 **공통 열**을 기준으로 데이터를 결합
    - 데이터베이스의 JOIN 작업과 동일
    - merge()함수를 이용하여 두 데이터셋의 **교집합/합집합**을 만듬
    - match() 함수 / %in% 연산자도 유사한 작업 수행

서브셋

- 데이터셋으로부터 데이터 일부를 추출하여 서브셋을 만드는 작업
- 후속 분석 작업에 활용
- R의 다양한 인덱싱방법을 이용
- 실습부분 책 참조

$

- 데이터셋으로부터 하나의 원소를 추출
- 데이터프레임에 적용시 벡터 반환
- 이름을 갖는 리스트에 적용시 해당 원소 추출

[[]]

- 하나의 원소를 반환
- 이름 또는 위치로 원소 선택 가능
- 리스트와 데이터프레임에 사용가능

[]

- 데이터셋으로부터 여러개의 원소를 추출

소팅

- 데이터 분석이나 전처리 작업중 흔히 발생
- sort() 함수를 이용해 벡터를 소팅할 수 있음
    - 달리 지정하지 않으면 오름차순으로 소팅
    - 내림차순으로 소팅할려면 decreasing=TRUE 옵션 지정
    - 벡터에 NA가 포함되어 있을 경우 NA값을 맨앞/뒤로 위치시키기

        - na.last 지정안할 경우(기본값 NA)
            - 결과에는 NA값이 출력되지 않음
        - na.last = TRUE 지정시 NA값이 결과의 맨 뒤에 위치
        - na.last = FALSE 지정시 NA값이 결과의 맨 앞에 위치
- order() 함수 : 벡터가 소팅되었을 때의 원소의 순서를 원소의 인덱스로 반환
    - 벡터 자체를 소팅하는게 아니라, 소팅되었을 경우의 순서를 나타냄
    - 데이터프레임을 소팅할 때 유용
    - 두 개 이상의 벡터를 인수로 제공하면 두 개 이상의 열을 이용한 소팅이 가능
        - 첫번째 벡터로 소팅 순서 결정, 동률 발생 시 나머지 벡터에 의해 소팅
        - 모든 열이 동일하게 오름차순 또는 내림차순의 쇵 순서를 가짐
- xtfrm() 함수 : 어떤 열은 오름차순, 어떤 열은 내림차순으로 정렬할 필요가 있을 경우
    - 주어진 벡터를 오름차순의 소팅 순서를 나타내는 숫자 벡터로 변환
    - 변환한 뒤 음수로 만들면 내림차순으로 소팅방법 변경 가능

변수 변환

할당 연산자를 이용하는 법

- 할당연산자 <-를 이용해서 그냥 덮어쓰면 됨
- 기존의 변수를 이용해 새로운 변수 생성도 가능

Transform() 함수

- 지정된 표현식을 데이터프레임에 차례대로 적용하여 최종 결과 데이터프레임을 **반환**
    - 첫번째 인수 : 데이터프레임
    - 나머지 인수 : 변수를 변환하는 표현식, 할당 연산자 대신 =을 이용함
- 어떤 함수를 일련의 데이터 원소들에 반복적으로 적용
    - 'apply'함수들 이용
    - apply
        - 적용하고자 하는 데이터 객체가 **행렬**일 때 이용
        - 두번째 인수에 지정된 차원에 대해 세번째 인수에 지정된 함수 반복 적용 후 벡터/행렬형식 **반환**
        - 첫 번째 인수 : X, 함수가 적용되느 행렬
        - 두 번째 인수 : MARGIN, 행렬에서 함수가 적용될 차원 지정(1=행, 2=열)
            - 다차원 행렬에 적용시, 결과값이 행렬형식이 될 수 있음
        - 세 번째 인수 : FUN, 함수 지정
            - 추가적인 인수를 필요로 할 경우 세번째 이후 지정가능
    - lapply(list)
        - 적용하고자 하는 데이터 객체가 **리스트 또는 벡터, 데이터프레임**일 때 이용
        - 실행결과를 항상 리스트 형식으로 **반환**
        - 내부에서 호출하는 함수는 사용자 정의/내장함수 다 됨
    - sapply(simplify)
        - 적용하고자 하는 데이터 객체가 **리스트 또는 벡터, 데이터프레임**일 때 이용

        - 실행결과를 될 수 있는대로 벡터 또는 행렬로 반환
        - 실행 결과 반환되는 값이 모두 동일한 길이를 가져야 단순화 됨
        - 내부에서 호출하는 함수는 사용자 정의/내장함수 다 됨
    - mapply
        - 복수의 벡터나 리스트에 대해 함수를 적용
        - 첫 번째 인수 : FUN, 이용하고자 하는 함수 지정
        - 두 번째 이후 : 지정한 함수에 인수로 제공되는 벡터나 리스트 지정
            - 추가적인 인수가 필요할 경우 두번째 이후 지정가능
        - 적용하려는 함수가 벡터 연산을 지원하지 않을 때 유용

집단 요약

- 집단별 합산을 하거나 집단별 평균을 구하는 작업 등
- 벡터를 집단별로 분할 한 후 데이터 처리
    - split 함수, unstack 함수 이용
    - split
        - 첫 번째 인수 : X, 벡토 또는 데이터 프레임 지정
        - 두 번째 인수 : f, 집단을 나타내느 팩터 지정
        - 벡터 또는 데이터프레임이 집단별로 분할된 **리스트**가 반환
    - unstack
        - 인수 : 분할할 벡터와 집단변수(팩터)를 데이터프레임으로 만든 것
        - 분할된 **리스트가 반환**
        - 주어진 벡터들이 모두 동일한 길이를 가질 때 집단별로 분할된 데이터셋의 형식을 리스트에서 **데이터 프레임로 변환되어 반환**
- 집단별 분할과 동시에 연산 작업을 하면 좀 더 효과적인 데이터 처리 작업 가능
    - tapply
        - 첫 번째 인수 : X, 처리할 대상 벡터
        - 두 번째 인수 : INDEX, 집단을 나타내는 팩터 지정
        - 세 번째 인수 : FUN, 적용할 함수 지정
        - 벡터로부터 집단별로 원소를 추출해 함수를 적용한 후 처리 결과를 모아 벡터/행렬로 반환
        - 두개 집단 간 요약표 만들기 가능
            - 집단 변수 둘을 리스트 형식으로 지정해줌
    - aggregate
        - 집단별로 분할된 데이터셋에 각각 함수를 적용해 집단별 처리 결과 반환
        - 첫 번째 인수 : X, 처리할 대상 벡터 지정
        - 두 번째 인수 : by, 집단을 나타내는 변수를 **리스트 형식**으로 지정
        - 세 번째 인수 : FUN, 적용할 함수
        - 데이터프레임을 대상으로 집단별 데이터 처리도 가능
    - by
        - 데이터프레임을 대상으로 집단별 데이터 처리 수행 가능
        - 데이터프레임의 행을 집단으로 분할하고 집단별로 함수를 적용해 리스트로 반환
        - 첫 번째 인수 : data, 데이터프레임

        - 두 번째 인수 : INDICES, 집단을 나타내는 팩터
        - 세 번째 인수 : FUN, 적용하고자 하는 함수
            - 데이터 프레임을 처리할 수 있는 어떤 함수든 가능
            - 사용자 정의 함수도 가능
    - rowsum
        - 집단별로 합산한 값에만 관심이 있는 경우
        - 첫 번째 인수 : x, 행렬/데이터프레임/벡터
        - 두 번째 인수 : group, 집단을 나타내는 팩터
    - tabulate
        - 각각의 범주별 관측값이 몇개인지 알고 싶을 때
        - 범주를 나타내는 각 정수값에 대응되는 벡터 내 원소의 개수를 **벡터 형식으로 반환**
        - 이해하기 쉽도록 개수에 대응되는 이름을 추가해주는게 좋음
    - table
        - 데이터셋에 집단을 나타내는 범주형 변수나 범주형 변수로 취급될 수 있는 변수가 포함되어있을 경우
        - 각 범주별로 포함된 원소의 개수를 셀 수 있음
        - 2차원 교차표 만들기 가능 (,로 구분
        - 기본적으로 팩터에 대해서만 사용가능
        - cut() 함수를 사용해 숫자 데이터를 구간으로 분할한 후 table() 함수를 적용하면 구간별 빈도 계산 됨
    - xtabs
        - table과는 다르게 범주별 빈도 계산 시 포뮬러 인터페이스 방식 이용
        - 첫 번째 인수 : formula, 범주로 사용할 변수를 포뮬러 형식으로 지정
        - 두 번째 인수 : data,데이터프레이 지정
        - 포뮬러 인터페이스 : 데이터 처리 시 어떤 열을 사용할 것인지를 일종의 수식형식으로 지정하는 방식
            - y~x : y는 x의 함수다
            - y~a+b : a열과b열을 모델에 포함시킴
            - y~a-b : a열은 모델에 포함시키나 b열은 제외
            - y~a:b : a열과 b의 상호작용 효과를 모델에 포함시킴
            - y~a*b : a열과 b열, 그리고 두열의 상호작용 효과를 모델에 포함시킴
            - y~a|b : y는 v를 조건으로 한 a의 조건부 함수

분할 - 적용 - 결합

- SAC(Split, Apply, Combine) : 데이터를 분할하고 특정함수를 적용하고 결과를 모아 결합하는 일련의 과정

dplyr 패키지

- 데이터프레임 형식의 데이터셋을 대상으로 이러한 일련의 데이터 처리 과정을 지원하는 유용한 함수 제공

filter

- 데이터프레임으로부터 조건과 일치하는 행을 추출
- 첫 번째 인수 : 데이터프레임
- 두 번째 이후 : 추출 조건 지정
- filter(airquality, Month == 6)
- AND 조건 : , 또는 & 연산자
- OR 조건 : | 연산자

slice

- 단순히 특정 위치의 행을 추출
- 첫 번째 인수 : 데이터프레임
- 두 번째 인수 : 추출하고자 하는 행 번호

arrange

- 지정한 열을 기준으로 오름차순으로 데이터프레임을 소팅
- 첫 번째 인수 : 데이터프레임
- 두 번째 이후 : 소팅 기준 열 지정
    - 두 개 이상 지정시 동률이 발생할 때 추가로 지정한 열에 의해 소팅
    - 내림차순으로 소팅시 열 이름에 desc()함수 씌워서 함께 사용
- arrange(airquality, Temp, Month, Day)

select

- 열 추출
- 첫 번째 인수 : 데이터프레임
- 두 번째 이후 : 추출할 열 지정
    - 여러 열을 추출하려면 , 연산자로 구분, 인접한 열 추출 시 : 연산자 이용가능
    - 추출에서 제외하려면 앞에 -부호
- 열 이름 변경 가능
- distinct()함수와 함께 사용 시 특정 열에 포함된 중복되지 않은 고유한 값 추출 가능
    - select문을 인자로 받음
    - Unique 함수와 유사

rename

- 열의 이름 변경
- rename(airquality, Solar = Solar.R)

mutate

- 새로운 열을 추가 할 때 사용
- 첫 번째 인수 : 데이터프레임
- 두 번째 이후 : **새롭게 생성할 변수**에 대한 표현식 지정

Summarise

- 평균, 중위수, 표준편차, 최소값, 최대값 같은 요약 통계량 계산
- 첫 번째 인수 : 데이터프레임
- mean, median, sd, min, max 등 통계 계산 함수 사용 가능
- n() : 데이터셋의 행의 개수
- n_distint(x) : 중복되지 않는 고유한 값의 개수
- first(x), last(x), nth(x, n)

Others

- sample_n() : 지정한 숫자만큼의 표본 추출
- sample_frac() : 지정한 비율만큼의 표본 추출, 복원추출 옵션 = replace=TRUE
- group_by() : 주어진 데이터셋으로 부터 집단변수의 레벨별로 분할된 그룹화된 객체 생성
    - 첫 번째 인수 : 데이터프레임
    - 두 번째 인수 : 집단변수
    - 외관상으로는 기존의 데이터셋과 구별 불가능, 이후 적용되는 함수를 통해 그룹화된 특성 발현
    - 생성 후 다른 함수들을 적용해 집단별 처리 수행 가능
- 파이프 연산자(%>%)
    - 한 함수로부터의 출력 결과나 데이터를 다음 함수의 첫 번째 인수로 전달
        - 좌항의 output이 우항의 input으로 전달
    - 중간에 데이터프레임을 생성할 필요가 없음
- tbl_df() : 데이터프레임을 화면 크기에 맞춰서 출력
    - 데이터프레임을 tbl 객체로 변환함(내부에 데이터프레임 객체를 품음)
    - as.data.frame으로 다시 순수한 데이터프레임으로 변환 가능

형태 변환

- 와이드 포맷 : 너비는 넓고 길이는 짧게 데이터를 배열
    - 변수들이 각각 하나의 열로 표현되어 한행에 관련된 변수와 값들이 차례대로 옆으로 넓게 배열
    - 한 분석대상 = 하나의 행
- 롱 포맷 : 너비는 좁고 길이는 길게 데이터를 배열
    - 모든 변수 목록을 나타내는 열 하나(variable), 값을 나타내는 열 하나(value)
    - 하나의 행 : 하나의 변수에 대한 측정값
    - 분석 대상의 변수가 여러개일 경우 변수값들은 여러 행에 걸쳐 나타남
    - 한 분석대상 = 여러 행

Reshape 패키지

- melt() : 와이드 -> 롱
    - 첫 번째 인수 : 와이드포맷의 데이터 셋
    - 식별자를 지정하지 않으면 임의로 지정
    - 팩터가 포함되어 있으면 이를 식별자로 사용
        - 직접 지정하려면 id.vars 인수에 식별자 지정
    - 숫자형 데이터를 갖는 모든 열을 대상으로 롱포맷 변환 수행 -> 식별자 지정
    - 인수를 명시적으로 지정해 롱포맷의 구성 요소를 보다 명확히 정의 가능
    - variable.name 인수 : 변수목록을 나타내는 열 이름 지정, default = variable
    - value.name 인수 : 측정 변수값을 나타내는 열 이름 지정, defalut = value
- dcast() : 롱 -> 와이드
    - 첫 번째 인수 : data, 롱포맷의 데이터셋
    - 두 번째 인수 : formula, 변수를 배치하는 포뮬라
        - x_variable ~ y_variable 형태로 지정, 앞은 **식별자 변수**, 뒤는 **측정 변수**
        - .은 변수가 없음을 나타냄

        - ...은 포뮬러에 지정되지 않은 다른 모든 변수
    - 측정변수 값이 어떤 열인지를 명시적으로 지정하는게 좋음(value.var 인수)
    - 하나의 셀에 여러개의 값이 들어가게 되는 경우 문제 발생
        - 해당 값들을 어떻게 할지 지정해줘야 함 : fun.aggregate 옵션으로 함수 지정

tidyr 패키지

- reshape2 패키지가 제공하는 기능을 데이터프레임 형식에 대해서만 사용할 수 있도록, 제한하지만 성능 향상 및 사용법 간소화
- melt() -> gather()
    - 첫 번째 인수 : data, 변환 대상 데이터셋
    - 두 번째 인수 : key, 변수명을 포함할 열의 이름 지정
    - 세 번째 인수 : value, 값을 포함할 열의 이름 지정
    - 네 번째 인수 : 롱포맷으로 변환할 변수 리스트 지정, 연속된 변수 지정시 : 연산자 사용 가능
        - 포함되지 않은 건 식별자로 감
        - 제외할 열을 -앞에 부텽서 제외가능(식별자 변수로 사용됨)
        - 인덱스, 직접 변수 이름 입력도 가능
    - 파이프 연산자 이용가능(%>%)
- dcast() -> spread()
    - 첫 번째 인수 : data, 변환 대상 데이터셋
    - 두 번째 인수 : key, 변수명을 포함하고 있는 열의 이름 지정
    - 세 번째 인수 : value, 값을 포함하고 있는 열의 이름 지정
- separate() : 하나의 열을 여러개의 열로 분할
    - 첫 번째 인수 : data, 대상 데이터셋
    - 두 번째 인수 : col, 분할할 현재 열 이름
    - 세 번째 인수 : into, 분할 후에 새롭게 생성할 열 이름
    - sep 인수 : 분할에 사용할 문자를 나탄는 정규표현식 / 분할 위치를 나타내는 숫자
        - 기본값 : 문자, 숫자가 아닌 값을 나타내는 정규표현식
- unite() : 여러개의 열을 하나로 통합
    - 첫 번째 인수 : data, 데이터셋
    - 두 번째 인수 : col, 결합 후에 새롭게 생성할 열 이름
    - 세 번째 이후 : 결합할 열 이름 차례로 지정
    - sep 인수 : 결합한 값들을 연결하는데 사용할 문자 지정

표준패키지

- stack() : 인수로 주어진 데이터프레임을 두 개의 열을 갖는 롱포맷 데이터프레임으로 변환
- unstack() : 팩터 변수의 각 레벨을 열로 하는 와이드포맷의 데이터프레임 생성