분석프로그래밍 I.06
데이터 합치기, 구조 변형하기

국민대학교 경영학부 빅데이터경영통계학과

2018. 4. 09(월)

1 지난 시간 복습 : 데이터프레임을 집단별로 분리한 후, 함수 적용하기

1.1 데이터프레임을 인자로 받는 함수들

- 여러 가지 요약 통계치를 한꺼번에 계산해 주는 함수들

  `summary()`, `psych::describe()`, `Hmisc::describe()`, `pastecs::stat.desc()`

- 상관계수, 공분산 행렬 계산

  `cor()`, `cov()`, `psych::corr.test()`

- 산점도 등 플롯

  `corrgram::corrgram()`, `car::scatterplotMatrix()`

1.2 by와 데이터프레임을 인자로 받는 함수들

```
by(df, df$am, summary)
by(df, df$am, psych::describe)
by(df, df$am, Hmisc::describe)

by(df, df$am, cov)

by(df, df$am, corr.test)

# Compare to dplyr way
df %>% group_by(am) %>% do(data.frame(psych::describe(.)))
```

Listing 1: by 활용 예.

1.3 벡터를 인자로 받는 요약/집계 함수

- 집중경향치

  `mean()`, `median()`, `getmode()`[1]

```r
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

Listing 2: `getmode()` 함수 정의

- 변산성 측정

  `var()`, `sd()`, `IQR()`

- 왜도(대칭성), 첨도(뾰족한 정도)

  `psych::skew()`, `psych::kurtosi()`

  `e1071::skew()`, `e1071::kurtosis()`

- 그 밖의 함수들

  `min()`, `max()`, `quantile( , probs=0.5)`

1.4 `aggregate` : 데이터프레임을 집단별로 나눈 후 컬럼에 함수 적용하기

```r
aggregate(df, list(am=df$am), min)
aggregate(df[,c("mpg","disp","wt")], list(am=df$am), min)
aggregate(cbind(mpg,disp,wt) ~ am, df, min)

aggregate(df[,c("mpg","disp","wt")], list(am=df$am, cyl=df$cyl), min)
```

[1] `https://www.tutorialspoint.com/r/r_mean_median_mode.htm` 참조

```
aggregate(cbind(mpg,disp,wt) ~ am+cyl, df, min)

library(magrittr)
library(dplyr)
df %>% select(mpg,disp,wt,am,cyl) %>% group_by(am,cyl) %>% summarise(
    min(mpg), min(disp), min(wt))
```

Listing 3: aggregate 함수 활용하기

2 여러 데이터 프레임 합치기 : rbind, cbind, merge

2.1 행 합치기 : rbind

- 열이름, 순서, 갯수가 일치할 때 : cbind(df1, df2)
- 열이름이 다르거나, 순서가 불일치 할 때

```
colnames(df2)[colnames(df2)=="name"] = "customerName"
cbind(df1, df2)

df3 <- df2[colnames(df1)]
cbind(df1, df3)

# rename from package dplyr
dply::rename(df1, "name"="customerName")
```

Listing 4: cbind 함수 활용하기

- 열이름은 같지만, 순서 또는 갯수가 다를 때

  plyr::rbind.fill(df1, df2), dplyr::bind_rows(df1, df2)

2.2 열 합치기 : cbind

- 행이름, 순서, 갯수가 같을 때 : cbind(df1, df2)

- 행의 순서 또는 갯수가 다를 때 :

  `merge(df1, df2, by="row.names", all=T)`

- 식별자가 다를 때 : `merge(df1, df2, by.x="id1", by.y="id2", all=T)`

- 식별자는 같지만, 위치 또는 갯수가 다를 때 :

  `merge(df1, df2, by="id", all=T), dplyr::full_join(df1, df2, by="id")`

2.3 두 데이터 프레임의 정보를 식별자(id)를 통해 합칠 때

- inner-join, outer-join, left-join, right-join

```
options(stringsAsFactors=F)
dfCustomer <- data.frame(
  id = c(1,2,3,4,5),
  name = c("김희선","박보검","설현","김수현","전지현"),
  addr = c("서울시","부산시","인천시","강릉시","목포시")
)
dfPurchase <- data.frame(
  name = c("김희선","박보검","김희선","설현","김수현","박보검"),
  product = c("삼푸","텔레비전","통닭","핸드폰","바지","삼푸")
)
dfProduct <- data.frame(
  product = c("삼푸","텔레비전","통닭","핸드폰","바지","PC","연필"),
  price =c(13800, 560000,20000,250000,80000,1380000,2000)
)

merge(dfCustomer, dfPurchase, by="name")
library(dplyr)
inner_join(dfCustomer, dfPurchase, by="name")
full_join(dfCustomer, dfPurchase, by="name")
left_join(dfCustomer, dfPurchase, by="name")
right_join(dfCustomer, dfPurchase, by="name")

```

```r
inner_join(dfPurchase, dfProduct, by="product")
full_join(dfPurchase, dfProduct, by="product")
left_join(dfPurchase, dfProduct, by="product")
right_join(dfPurchase, dfProduct, by="product")

#Filtering Join
semi_join(dfProduct, dfPurchase)
anti_join(dfProduct, dfPurchase)

# Purchase data
df1 <- inner_join(dfCustomer, dfPurchase, by="name")
df2 <- left_join(df1, dfProduct, by="product")

# Product data
left_join(dfProduct, dfPurchase, by="product")
```

Listing 5: □□□□_join 함수 활용하기

2.4 데이터 프레임의 구조 변경하기 : long-form, wide-form

표 1, Class06_Join_Reshape.R, R_wide long_published160404.pdf 참조.

표 1: 데이터 형태 변환을 위한 함수들

함수	패키지	long-form	wide-form
stack/unstack	utils	stack	unstack
reshape	stats	reshape("long", ...)	reshape("wide", ...)
melt/dcast	reshape2	melt	dcast
gather/spread	tidyr	gather	spread