헬로 데이터 과학: R 시작하기

김진영

# 환경 설정하기

우선 이 책에서 제공하는 R 라이브러리를 로딩하자.

source("dbook.R")

이제 자주 사용하는 패키지를 설치할 차례다. 위 글에서 설명했듯이, R에서는 기본적으로 다양한 통계 및 시각화 기능을 제공하지만, R의 진정한 힘은 패키지 형태로 제공되는 부가적인 기능에서 나온다. 다음은 이 책에서 자주 사용할 패키지를 세 개를 설치하는 명령어다.

load.packages(c("stringr", "ggplot2", "dplyr"))

## [1] loading stringr  
## [1] loading ggplot2  
## [1] loading dplyr

# R의 기본 기능

## 변수 정의하기

R에서는 다음과 같이 다양한 형태의 변수를 제공한다.

# 수치 / 문자열 / 배열 자료형을 정의한다.  
v1 = 1  
v2 = "abc"  
v3 = c(1,2,3)  
  
# 데이터 프레임을 정의한다  
df1 = data.frame(  
 Name=c("Jerry","Tom","Smith"),  
 Math=c(50,60,75))  
df1

## Name Math  
## 1 Jerry 50  
## 2 Tom 60  
## 3 Smith 75

## 데이터 읽고 쓰기

이런 기본 데이터는 R을 공부하는 목적으로는 적합하지만, 일반적으로는 디스크에 파일 형태로 저장된 데이터를 불러와서 작업하게 된다. 여기서는 실습을 위해 mtcars 데이터를 파일로 저장하고 다시 읽어보자. 데이터를 파일로 저장하거나 불러들이는데 write.table() 및 read.table() 함수를 사용하는데, 다음은 mtcars 데이터를 저장하고 이를 mcars라는 이름으로 다시 불러들이는 방법이다. (R에서는 파일 이외에도 관계형 데이터베이스, 웹 주소 등 다양한 형태의 데이터를 사용할 수 있다.)

write.table(mtcars, "mtcars\_new.txt")  
mcars = read.table("mtcars\_new.txt", header=T)

앞으로는 원본 데이터인 mtcars를 두고 mcars를 사용할 것이다. 이처럼 원본 데이터를 가공된 데이터와 별도로 보관하는 것은 좋은 습관인데, 이를 통해 원본 데이터를 언제든지 다시 살펴보고 재가공할 수 있기 때문이다.

어떤 경우에는 R에서 가공한 데이터를 엑셀 등의 프로그램에서 쓰고 싶을 것이다. 이런 경우에는 위에서처럼 데이터를 파일에 저장하고 다시 불러들여도 되지만, 클립보드에 저장하면 더 편리한데 이를 위해서는 위의 파일명 대신 "clipboard"를 사용하면 된다. 이렇게 저장한 데이터를 엑셀로 불러들일 때에는 앞에서 설명한 것처럼 ‘텍스트 마법사’를 사용하면 된다.

write.table(mcars, "clipboard")

## 데이터 살펴보기

이제 실제로 데이터를 불러와서 다루는 방법을 알아보자. R 실습 절차는 앞에서 소개한 엑셀 실습 절차뿐만 아니라 실습 데이터 역시 mtcars 데이터셋을 그대로 사용한다. 이 데이터는 R에 기본적으로 mtcars라는 이름으로 설치되어있기 때문에 그냥 mtcars라는 명령을 실행하면 데이터를 볼 수 있다. 그리고 summary() 함수를 사용하면 각 속성별로 값의 분포를 볼 수 있다.

summary(mtcars)

## mpg cyl disp hp   
## Min. :10.40 Min. :4.000 Min. : 71.1 Min. : 52.0   
## 1st Qu.:15.43 1st Qu.:4.000 1st Qu.:120.8 1st Qu.: 96.5   
## Median :19.20 Median :6.000 Median :196.3 Median :123.0   
## Mean :20.09 Mean :6.188 Mean :230.7 Mean :146.7   
## 3rd Qu.:22.80 3rd Qu.:8.000 3rd Qu.:326.0 3rd Qu.:180.0   
## Max. :33.90 Max. :8.000 Max. :472.0 Max. :335.0   
## drat wt qsec vs   
## Min. :2.760 Min. :1.513 Min. :14.50 Min. :0.0000   
## 1st Qu.:3.080 1st Qu.:2.581 1st Qu.:16.89 1st Qu.:0.0000   
## Median :3.695 Median :3.325 Median :17.71 Median :0.0000   
## Mean :3.597 Mean :3.217 Mean :17.85 Mean :0.4375   
## 3rd Qu.:3.920 3rd Qu.:3.610 3rd Qu.:18.90 3rd Qu.:1.0000   
## Max. :4.930 Max. :5.424 Max. :22.90 Max. :1.0000   
## am gear carb   
## Min. :0.0000 Min. :3.000 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:2.000   
## Median :0.0000 Median :4.000 Median :2.000   
## Mean :0.4062 Mean :3.688 Mean :2.812   
## 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:4.000   
## Max. :1.0000 Max. :5.000 Max. :8.000

#str(mtcars)

R의 기본 자료형은 테이블 형태의 데이터 프레임이다. mtcars는 큰 데이터는 아니지만 그래도 한 화면에 들어오지는 않는다. 데이터의 첫 몇 항목만 보기 위해서는 head()라는 함수를 사용한다. head() 함수를 실행하면 아래와 같이 cars의 첫 여섯 줄이 출력된다.

head(mcars)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

만약 첫 10줄을 보고 싶다면 다음과 같이 실행하면 된다. 이렇게 R의 함수는 대부분 다양한 옵션을 제공하는데, 함수에 대한 설명을 보고 싶으면 "? 함수명"을 입력하면 된다.

head(mcars, n=10)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4

head()가 데이터의 첫 부분을 보여준다면 tail()은 마지막을 보여준다. 실행 방법은 head()와 동일하다.

tail(mcars)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.7 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.9 1 1 5 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.5 0 1 5 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.5 0 1 5 6  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.6 0 1 5 8  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.6 1 1 4 2

R에서는 기본적으로 데이터의 각 행과 열에 이름이 주어진다. 주어진 데이터의 각 행과 열에 대한 이름은 rownames() 및 colnames() 함수를 사용하여 볼 수 있다. 다음은 mcars 데이터의 행과 열의 이름을 보는 방법이다.

rownames(mcars)

## [1] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710"   
## [4] "Hornet 4 Drive" "Hornet Sportabout" "Valiant"   
## [7] "Duster 360" "Merc 240D" "Merc 230"   
## [10] "Merc 280" "Merc 280C" "Merc 450SE"   
## [13] "Merc 450SL" "Merc 450SLC" "Cadillac Fleetwood"   
## [16] "Lincoln Continental" "Chrysler Imperial" "Fiat 128"   
## [19] "Honda Civic" "Toyota Corolla" "Toyota Corona"   
## [22] "Dodge Challenger" "AMC Javelin" "Camaro Z28"   
## [25] "Pontiac Firebird" "Fiat X1-9" "Porsche 914-2"   
## [28] "Lotus Europa" "Ford Pantera L" "Ferrari Dino"   
## [31] "Maserati Bora" "Volvo 142E"

colnames(mcars)

## [1] "mpg" "cyl" "disp" "hp" "drat" "wt" "qsec" "vs" "am" "gear"  
## [11] "carb"

데이터를 실제로 다루기 위해서는 각 속성을 선택할 수 있어야 한다. R에서는 데이터 이름과 속성 이름 사이에 $를 사용하여 데이터의 속성을 선택할 수 있다. 다음은 mpg 속성을 보는 명령이다.

mcars$mpg

## [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2  
## [15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4  
## [29] 15.8 19.7 15.0 21.4

# 데이터 준비하기

## 기본 가공법

이번에는 기존 속성의 값을 가공해서 새로운 속성을 만드는 방법을 알아보자. 우선 위에서 살펴본대로 자동차의 이름이 각 행의 이름으로 들어가 있는데, 이를 model이라는 이름의 속성으로 만들어 보자.

mcars$model = rownames(mcars)  
rownames(mcars) = NULL

model이라는 속성을 보면 첫 단어가 해당 모델의 제조사 이름이다. stringr 라이브러리의 word() 함수는 문자열 형태의 속성에서 단어를 추출하는 기능을 제공한다. 아래는 model 속성의 첫 단어를 추출하여 maker라는 속성을 만드는 예다. 이 작업을 완료한 후에 데이터를 head() 명령으로 다시 살펴보면 두개의 속성이 추가된 것을 알 수 있다.

mcars$maker = word(mcars$model, 1)  
head(mcars)

## mpg cyl disp hp drat wt qsec vs am gear carb model  
## 1 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4 Mazda RX4  
## 2 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4 Mazda RX4 Wag  
## 3 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1 Datsun 710  
## 4 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1 Hornet 4 Drive  
## 5 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2 Hornet Sportabout  
## 6 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1 Valiant  
## maker  
## 1 Mazda  
## 2 Mazda  
## 3 Datsun  
## 4 Hornet  
## 5 Hornet  
## 6 Valiant

mcars$cyl == 4

## [1] FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE  
## [12] FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE FALSE  
## [23] FALSE FALSE FALSE TRUE TRUE TRUE FALSE FALSE FALSE TRUE

mcars.small = mcars[mcars$cyl == 4, ] # filter rows  
mcars.small.narrow = mcars.small[, c("model","maker", "mpg", "cyl")] # select columns by name  
mcars.small.narrow = mcars.small[, c(12:13, 1:2)] # select columns by index  
mcars.small.narrow

## model maker mpg cyl  
## 3 Datsun 710 Datsun 22.8 4  
## 8 Merc 240D Merc 24.4 4  
## 9 Merc 230 Merc 22.8 4  
## 18 Fiat 128 Fiat 32.4 4  
## 19 Honda Civic Honda 30.4 4  
## 20 Toyota Corolla Toyota 33.9 4  
## 21 Toyota Corona Toyota 21.5 4  
## 26 Fiat X1-9 Fiat 27.3 4  
## 27 Porsche 914-2 Porsche 26.0 4  
## 28 Lotus Europa Lotus 30.4 4  
## 32 Volvo 142E Volvo 21.4 4

## 고급 가공법 (dplyr)

이제 dplyr 라이브러리를 사용하여 데이터를 본격적으로 가공해보자. 데이터 가공에는 행과 열을 선택하는 필터링, 그룹을 만들어 통계 값을 계산하는 집계, 데이터의 순서를 변경하는 정렬 등의 연산이 포함되는데, R에서는 이련 개별 연산에 해당하는 함수를 제공하며, 이들을 '%>%'라는 기호를 사용하여 연결할 수 있다.

우선 데이터에서 필요한 행과 열을 선택하는 과정을 알아보자. 위에서 만든 cars 데이터에서 실린더 개수(cyl)가 4개인 차의 제조사, 모델, 연비를 선택한다고 하자. R에서 이를 실행하는 명령과 그 결과는 다음과 같다(아래 코드에서 주석은 #로 표시한다).

mcars.small.narrow = # 결과를 mcars.small.narrow라는 변수에 저장한다.  
 mcars %>% # mcars 데이터를 사용한다.  
 filter(cyl == 4) %>% # cyl 값이 4인 데이터만 남긴다.  
 select(maker, model, mpg, cyl) # # 주어진 네 속성만 선택한다.  
mcars.small.narrow

## maker model mpg cyl  
## 1 Datsun Datsun 710 22.8 4  
## 2 Merc Merc 240D 24.4 4  
## 3 Merc Merc 230 22.8 4  
## 4 Fiat Fiat 128 32.4 4  
## 5 Honda Honda Civic 30.4 4  
## 6 Toyota Toyota Corolla 33.9 4  
## 7 Toyota Toyota Corona 21.5 4  
## 8 Fiat Fiat X1-9 27.3 4  
## 9 Porsche Porsche 914-2 26.0 4  
## 10 Lotus Lotus Europa 30.4 4  
## 11 Volvo Volvo 142E 21.4 4

이번에는 데이터를 집계하는 과정을 알아보자. 우선 집계를 위한 기준 속성을 선택하고, 집계된 그룹별로 계산할 통계 값을 적어주는 것이 순서다. 아래 명령은 cars 데이터를 제조사별로 집계하고, 각 그룹별로 연비의 평균을 구하는 과정을 보여준다.

makers =   
 mcars %>%   
 group\_by(maker) %>% # 제조사 기준으로 데이터를 집계한다.  
 summarize( maker.mpg = mean(mpg), maker.count = n()) # 제조사별 평균 mpg및 차종수  
head(makers)

## Source: local data frame [6 x 3]  
##   
## maker maker.mpg maker.count  
## (chr) (dbl) (int)  
## 1 AMC 15.2 1  
## 2 Cadillac 10.4 1  
## 3 Camaro 13.3 1  
## 4 Chrysler 14.7 1  
## 5 Datsun 22.8 1  
## 6 Dodge 15.5 1

마지막으로 두 테이블을 특정한 속성값을 기준으로 병합하는 과정을 알아보자. 이는 여러 원본과 가공절차를 거친 데이터를 합쳐서 분석하고자 할 때 필수적인 과정이다. 여기서는 원본 데이터인 cars와 제조사별 집계 데이터인 makers를 제조사명을 기준으로 병합하는 과정을 보여준다. 여기서 by=”maker”는 제조사명을 기준으로 두 테이블을 병합하라는 뜻이다.

mcars.makers = merge(mcars, makers, by="maker")  
head(mcars.makers)

## maker mpg cyl disp hp drat wt qsec vs am gear carb  
## 1 AMC 15.2 8 304 150 3.15 3.435 17.30 0 0 3 2  
## 2 Cadillac 10.4 8 472 205 2.93 5.250 17.98 0 0 3 4  
## 3 Camaro 13.3 8 350 245 3.73 3.840 15.41 0 0 3 4  
## 4 Chrysler 14.7 8 440 230 3.23 5.345 17.42 0 0 3 4  
## 5 Datsun 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## 6 Dodge 15.5 8 318 150 2.76 3.520 16.87 0 0 3 2  
## model maker.mpg maker.count  
## 1 AMC Javelin 15.2 1  
## 2 Cadillac Fleetwood 10.4 1  
## 3 Camaro Z28 13.3 1  
## 4 Chrysler Imperial 14.7 1  
## 5 Datsun 710 22.8 1  
## 6 Dodge Challenger 15.5 1

# 데이터 분석하기

## 요약 통계

table(mcars$cyl)

##   
## 4 6 8   
## 11 7 14

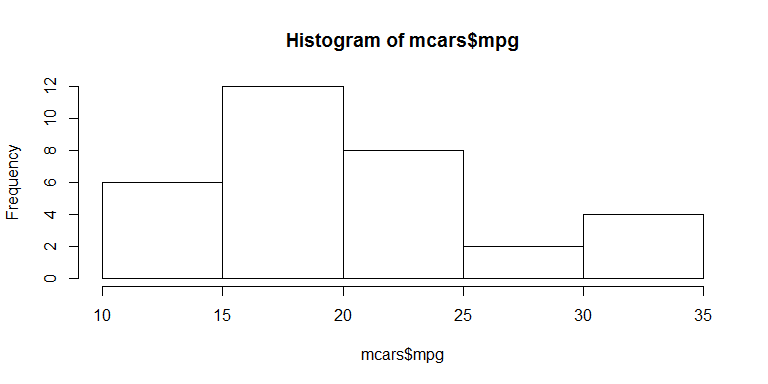
table(mcars$gear, mcars$cyl)

##   
## 4 6 8  
## 3 1 2 12  
## 4 8 4 0  
## 5 2 1 2

## 기본 시각화

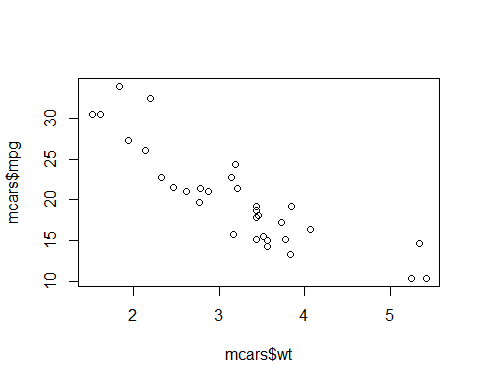
마지막으로 R에서 데이터를 시작화하는 과정을 알아보자. R은 기본적으로 히스토그램, 분산 차트와 같은 다양한 시각화 기능을 제공하며, 고급 시각화 기능이 필요하다면 ggplot2 등의 라이브러리를 사용하면 된다. 우선 아래는 mpg 속성의 값을 시각화하는 과정이다.

hist(mcars$mpg)



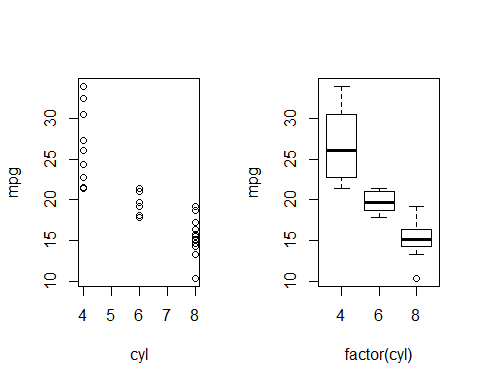
R에서 가장 흔히 사용하는 시각화는 두 속성간의 관계를 한눈에 보여주는 분산 차트이다. 아래 명령은 무게(wt)와 연비(mpg)속성간의 관계를 보여주는 스케터플롯을 보여준다. 무게 증가에 따라 연비가 떨어지는 추세가 확연히 나타난다.

plot(mcars$wt, mcars$mpg)



#plot(mcars[,1:5])

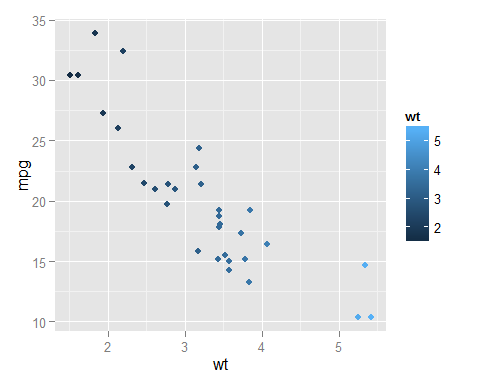
par(mfrow=c(1,2))  
plot(mpg ~ cyl, mcars)  
plot(mpg ~ factor(cyl), mcars)



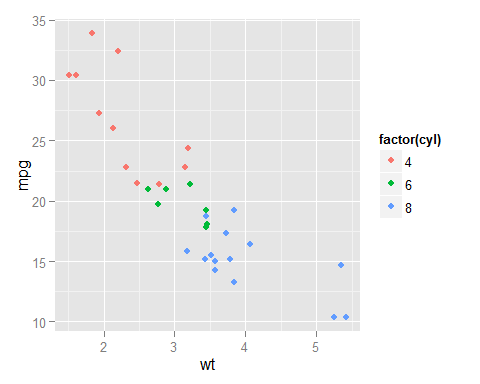
## 고급 시각화 (ggplot2)

때로는 둘 이상의 속성의 관계를 관찰해야 한다. R에서는 ggplot2 라이브러리에서 제공하는 qplot이라는 명령어를 사용하여 세 가지 이상 속성의 관계를 한눈에 볼 수 있다. 아래는 wt와 mpg 속성 간의 관계를 XY 축에 나타내고, cyl 값을 색상으로 표시하는 사례다. 이때 factor() 함수는 숫자를 카테고리 형태의 데이터로 인식하라는 명령이다. 아래 플롯에서는 연비(mpg)가 실린더 개수(cyl) 및 무게(wt)와 높은 상관 관계를 보임을 알 수 있다. 이런 종류의 시각화에는 엑셀에서 지원하지 않기 때문에, R을 사용할 이유가 된다.

qplot(wt, mpg, data=mcars, color=wt)



qplot(wt, mpg, data=mcars, color=factor(cyl))



#qplot(wt, mpg, data=mcars, geom = c("point", "smooth"))