



Quick-R

accessing the power of R

Home | Interface | Input | Manage | Stats | Adv Stats | Graphs | Adv Graphs | Blog

 Search

Data Management

[Creating New Variables](#)
[Operators](#)
[Built-in Functions](#)
[Control Structures](#)
[User-defined Functions](#)
[Sorting Data](#)
[Merging Data](#)
[Aggregating Data](#)
[Reshaping Data](#)
[Subsetting Data](#)
[Data Type Conversion](#)

R in Action



[R in Action](#) (2nd ed) significantly expands upon this material. Use promo code **ria38** for a 38% discount.

Top Menu

[Home](#)
[The R Interface](#)
[Data Input](#)
[Data Management](#)
[Basic Statistics](#)
[Advanced Statistics](#)
[Basic Graphs](#)
[Advanced Graphs](#)
[Blog](#)


DataCamp

FREE COURSE

Introduction to R

[Start Course for Free](#)

Subsetting Data

R has powerful indexing features for accessing object elements. These features can be used to select and exclude variables and observations. The following code snippets demonstrate ways to keep or delete variables and observations and to take random samples from a dataset.

Selecting (Keeping) Variables

```
# select variables v1, v2, v3
myvars <- c("v1", "v2", "v3")
newdata <- mydata[myvars]

# another method
myvars <- paste("v", 1:3, sep="")
newdata <- mydata[myvars]

# select 1st and 5th thru 10th variables
newdata <- mydata[c(1,5:10)]
```

Excluding (DROPPING) Variables

```
# exclude variables v1, v2, v3
myvars <- names(mydata) %in% c("v1", "v2", "v3")
newdata <- mydata[!myvars]

# exclude 3rd and 5th variable
newdata <- mydata[c(-3,-5)]

# delete variables v3 and v5
mydata$v3 <- mydata$v5 <- NULL
```

Selecting Observations

```
# first 5 observations
newdata <- mydata[1:5,]

# based on variable values
newdata <- mydata[ which(mydata$gender=='F'
& mydata$age > 65), ]

# or
attach(mydata)
```

```
newdata <- mydata[ which(gender=='F' & age > 65),]  
detach(mydata)
```

Selection using the Subset Function

The `subset()` function is the easiest way to select variables and observations. In the following example, we select all rows that have a value of age greater than or equal to 20 or age less than 10. We keep the ID and Weight columns.

```
# using subset function  
newdata <- subset(mydata, age >= 20 | age < 10,  
select=c(ID, weight))
```

In the next example, we select all men over the age of 25 and we keep variables weight *through* income (weight, income and all columns between them).

```
# using subset function (part 2)  
newdata <- subset(mydata, sex=="m" & age > 25,  
select=weight:income)
```

Random Samples

Use the `sample()` function to take a random sample of size `n` from a dataset.

```
# take a random sample of size 50 from a dataset mydata  
# sample without replacement  
mysample <- mydata[sample(1:nrow(mydata), 50,  
replace=FALSE),]
```

To Practice

The data frame chapter of [this online introduction to R course](#) has several exercises on subsetting data.