R Learning Module
Subsetting Data

Version info: Code for this page was tested in R version 3.0.2 (2013-09-25)
On: 2013-11-19
With: lattice 0.20-24; foreign 0.8-57; knitr 1.5

1. Subsetting variables

To manipulate data frames in R we can use the bracket notation to access the indices for the observations and the variables. It is easiest to think of the data frame as a rectangle of data where the rows are the observations and the columns are the variables. Just like in matrix algebra, the indices for a rectangle of data follow the RxC principle; in other words, the first index is for Rows and the second index is for Columns [R, C]. When we only want to subset variables (or columns) we use the second index and leave the first index blank. Leaving an index blank indicates that you want to keep all the elements in that dimension. In the first example we create the data frame **hsb3** containing only the variables **id**, **read** and **write**, but all the observations from the original data frame **hsb2.small**. In order to know which variables correspond to which number in the index we use the **names** function, which will list the names of the variables in the order in which they appear in the data frame. From this list we see that **id** is variable 1, **read** is variable 7 and **write** is variable 8. We cannot refer to the variables by their names alone until we have attached the data.

```
hsb2.small <- read.csv("http://www.ats.ucla.edu/stat/data/hsb2_small.csv")

# using the names function to see names of the variables and which column of
# data to which they correspond
names(hsb2.small)
```

```
##  [1] "id"      "female"  "race"    "ses"     "schtyp"  "prog"    "read"
##  [8] "write"   "math"    "science" "socst"
```

```
(hsb3 <- hsb2.small[, c(1, 7, 8)])
```

```
##     id read write
## 1   70   57    52
## 2  121   68    59
## 3   86   44    33
## 4  141   63    44
## 5  172   47    52
## 6  113   44    52
## 7   50   50    59
## 8   11   34    46
## 9   84   63    57
## 10  48   57    55
## 11  75   60    46
## 12  60   57    65
## 13  95   73    60
## 14 104   54    63
## 15  38   45    57
## 16 115   42    49
## 17  76   47    52
## 18 195   57    57
## 19 114   68    65
## 20  85   55    39
## 21 167   63    49
## 22 143   63    63
## 23  41   50    40
## 24  20   60    52
## 25  12   37    44
```

If the variables we want are in consecutive columns, we can use the colon notation rather than list them using the **c** function. In the next example we create the data frame **hsb4** containing the first four variables of **hsb2.small**.

```
(hsb4 <- hsb2.small[, 1:4])
```

```
##     id female race ses
## 1   70      0    4   1
## 2  121      1    4   2
## 3   86      0    4   3
## 4  141      0    4   3
## 5  172      0    4   2
## 6  113      0    4   2
## 7   50      0    3   2
## 8   11      0    1   2
## 9   84      0    4   2
## 10  48      0    3   2
## 11  75      0    4   2
## 12  60      0    4   2
## 13  95      0    4   3
## 14 104      0    4   3
## 15  38      0    3   1
## 16 115      0    4   1
## 17  76      0    4   3
## 18 195      0    4   2
## 19 114      0    4   3
## 20  85      0    4   2
## 21 167      0    4   2
## 22 143      0    4   2
## 23  41      0    3   2
## 24  20      0    1   3
## 25  12      0    1   2
```

2. Subsetting observations

We subset observations by also using the bracket notation but now we use the first index and leave the second index blank. This indicates that we want all the variables for specific observations. In the first example we create the data frame **hsb5**, which contains the first 10 observations of **hsb2.small**.

```
(hsb5 <- hsb2.small[1:10, ])
```

```
##     id female race ses schtyp prog read write math science socst
## 1   70      0    4   1      1    1   57    52   41      47    57
## 2  121      1    4   2      1    3   68    59   53      63    61
## 3   86      0    4   3      1    1   44    33   54      58    31
## 4  141      0    4   3      1    3   63    44   47      53    56
## 5  172      0    4   2      1    2   47    52   57      53    61
## 6  113      0    4   2      1    2   44    52   51      63    61
## 7   50      0    3   2      1    1   50    59   42      53    61
## 8   11      0    1   2      1    2   34    46   45      39    36
## 9   84      0    4   2      1    1   63    57   54      58    51
## 10  48      0    3   2      1    2   57    55   52      50    51
```

We can also subset observations based on logical tests. In the following example we create the data frame **hsb6**, which contains only the observations for which **ses**=1. For a logical equality we need to use the double equal sign notation. We also need to refer to the variable, **ses** in the data frame **hsb2.small**, which we do using **$**.

```
(hsb6 <- hsb2.small[hsb2.small$ses == 1, ])
```

```
##     id female race ses schtyp prog read write math science socst
## 1   70      0    4   1      1    1   57    52   41      47    57
## 15  38      0    3   1      1    2   45    57   50      31    56
## 16 115      0    4   1      1    1   42    49   43      50    56
```

In the previous example we used a logical test to subset the observations, but we only tested for one variable being equal to a single value. We can also subset using a logical test that will test a single variable being equal to the elements in a list, and we do this by using the **%in%** function. In the following example we create the data frame **hsb7**, which contains the observations where **id** is equal to 11, 12, 20, 48, 86 or 195.

```
(hsb7 <- hsb2.small[hsb2.small$id %in% c(12, 48, 86, 11, 20, 195), ])
```

```
##     id female race ses schtyp prog read write math science socst
## 3   86      0    4   3      1    1   44    33   54      58    31
## 8   11      0    1   2      1    2   34    46   45      39    36
## 10  48      0    3   2      1    2   57    55   52      50    51
## 18 195      0    4   2      2    1   57    57   60      58    56
## 24  20      0    1   3      1    2   60    52   57      61    61
## 25  12      0    1   2      1    3   37    44   45      39    46
```

It is also possible to combine logical tests. In the following example we create the data frame **hsb8**, which contains only the observations where **ses**=3 and **female**=0. Here to avoid having to type **hsb2.small** multiple times, we use the **with** function to let R know that it should look for **ses** and **female** inside the **hsb2.small** data frame.

```
(hsb8 <- hsb2.small[with(hsb2.small, ses == 3 & female == 0), ])
```

```
##     id female race ses schtyp prog read write math science socst
## 3   86      0    4   3      1    1   44    33   54      58    31
## 4  141      0    4   3      1    3   63    44   47      53    56
## 13  95      0    4   3      1    2   73    60   71      61    71
## 14 104      0    4   3      1    2   54    63   57      55    46
## 17  76      0    4   3      1    2   47    52   51      50    56
## 19 114      0    4   3      1    2   68    65   62      55    61
## 24  20      0    1   3      1    2   60    52   57      61    61
```

The **subset** function with a logical statement will let you subset the data frame by observations. In the following example the **write.50** data frame contains only the observations for which the values of the variable **write** is greater than 50. Note that one convenient feature of the **subset** function, is R assumes variable names are within the data frame being subset, so there is no need to tell R where to look for **write**.

```
(write.50 <- subset(hsb2.small, write > 50))
```

```
##     id female race ses schtyp prog read write math science socst
## 1   70      0    4   1      1    1   57    52   41      47    57
## 2  121      1    4   2      1    3   68    59   53      63    61
## 5  172      0    4   2      1    2   47    52   57      53    61
## 6  113      0    4   2      1    2   44    52   51      63    61
## 7   50      0    3   2      1    1   50    59   42      53    61
## 9   84      0    4   2      1    1   63    57   54      58    51
## 10  48      0    3   2      1    2   57    55   52      50    51
## 12  60      0    4   2      1    2   57    65   51      63    61
## 13  95      0    4   3      1    2   73    60   71      61    71
## 14 104      0    4   3      1    2   54    63   57      55    46
## 15  38      0    3   1      1    2   45    57   50      31    56
## 17  76      0    4   3      1    2   47    52   51      50    56
## 18 195      0    4   2      2    1   57    57   60      58    56
## 19 114      0    4   3      1    2   68    65   62      55    61
## 22 143      0    4   2      1    3   63    63   75      72    66
## 24  20      0    1   3      1    2   60    52   57      61    61
```

There is no limit to how many logical statements may be combined to achieve the subsetting that is desired. The data frame **write.1** contains only the observations for which the values of the variable **write** is greater than 50 and for which the variable **read** is greater than 60.

```
(write.1 <- subset(hsb2.small, write > 50 & read > 60))
```

```
##     id female race ses schtyp prog read write math science socst
## 2  121      1    4   2      1    3   68    59   53      63    61
## 9   84      0    4   2      1    1   63    57   54      58    51
## 13  95      0    4   3      1    2   73    60   71      61    71
## 19 114      0    4   3      1    2   68    65   62      55    61
## 22 143      0    4   2      1    3   63    63   75      72    66
```

It is possible to subset both rows and columns using the **subset** function. The **select** argument lets you **subset** variables (columns). The data frame **write.2** contains only the variables **write** and **read** and then only the observations of these two variables where the values of variable **write** are greater than 50 and the values of variable **read** are greater than 65.

```
(write.2 <- subset(hsb2.small, write > 50 & read > 60, select = c(write, read)))
```

```
##    write read
## 2     59   68
## 9     57   63
## 13    60   73
## 19    65   68
## 22    63   63
```

In the data frame **write.3** contains only the observations in variables **read** through **science** for which the values in the variable **science** are less than 55.

```
(write.3 <- subset(hsb2.small, science < 55, select = read:science))
```



```
##    read write math science
## 1    57    52   41      47
## 4    63    44   47      53
## 5    47    52   57      53
## 7    50    59   42      53
## 8    34    46   45      39
## 10   57    55   52      50
## 11   60    46   51      53
## 15   45    57   50      31
## 16   42    49   43      50
## 17   47    52   51      50
## 20   55    39   57      53
## 25   37    44   45      39
```

3. Subsetting both variables and observations

We can subset variables and observations by simply combining the two above methods of subsetting. We accomplish this by subsetting using both indices at the same time. In the following example we create the data frame **hsb9** in which we keep only the variables **id**, **female**, **race**, **ses** and **read** and only the observations where **ses**=3. Note again that because we are not using **subset**, we have to let R know where to find the variable **ses** by explicitly pointing to **hsb2.small**.

```
# using the names function to see names of the variables and which column of
# data to which they correspond
names(hsb2.small)
```

```
##  [1] "id"      "female"  "race"    "ses"     "schtyp"  "prog"    "read"
##  [8] "write"   "math"    "science" "socst"
```

```
(hsb9 <- hsb2.small[hsb2.small$ses == 3, c(1:4, 7)])
```

```
##     id female race ses read
## 3   86      0    4   3   44
## 4  141      0    4   3   63
## 13  95      0    4   3   73
## 14 104      0    4   3   54
## 17  76      0    4   3   47
## 19 114      0    4   3   68
## 24  20      0    1   3   60
```

GIS and Visualization

High Performance Computing
Hoffman2 Cluster
Hoffman2 Account Application
Hoffman2 Usage Statistics

GIS
Mapshare
Visualization
3D Modeling

Statistical Computing
Classes
Conferences
Reading Materials

Shared Cluster & Storage
About IDRE

Data Centers

S AYV5U9EUOH011

ABOUT CONTACT NEWS EVENTS OUR EXPERTS