Indexing in R: A summary with examples and exercises.

These pages summarize basic indexing of vectors and dataframes in R, as used in the course Bioinformatics of High-Thoughput Analysis, 2008, University of Copenhagen. It is by no means a complete description, and ignores most technicalities. Instead it has plenty of examples.
For more technically correct descriptions, please see the documentation on the R homepage, or in the R reference sheet.
The test file used here can be found at:
http://people.binf.ku.dk/albin/teaching/htbinf/R/

Comments and suggestions are welcome, please send them to
Sanne Nygaard (sanne@binf.ku.dk)

Getting started:
Suppose you have a text-file with some information on the members of your local Bridge club. You read this into R:

```
>club_members <- read.table(file='Kortklubben.txt', h=T)
#remember the header!
```

club_members is now a *dataframe*. Let's look at it:

```
>club_members
  Firstname    Lastname Age Gender Points
1     Alice     Knudsen  37      F    278
2      Poul     Knudsen  34      M    242
3     Jokum    Jonassen  26      M    312
4   Theodor Thorvaldsen  72      M    740
5   Babette   Brodersen  18      F    177
6   Lynette   Lauridsen  24      F    195
```

Each column in club_members is a *vector*. To get one specific vector from the dataframe, do:

```
>club_members$Age
[1] 37 34 26 72 18 24
```

Technical detail: If you do the same thing for eg. club_members$Gender, or any other of the text vectors, you will see an output like:

```
[1] F M M M F F
Levels: F M
```

where 'Levels' shows all different words in the vector. You dont need to worry about this, but if you are curious it is because the vector has been made into a *factor*. You can check the details in the R manuals and ?read.table.

Extracting from a vector:

To extract specific parts of a vector, you use square brackets. The syntax is:
Vectorname[criterion]
To get all ages above 30:

```
>club_members$Age[ club_members$Age > 30 ]
[1] 37 34 72
```

Technical detail: the [criterion] is itself a vector:

```
>club_members$Age > 30
[1]  TRUE  TRUE FALSE  TRUE FALSE FALSE
```

So you are in fact extracting all elements where the [criterion] is TRUE

You can also extract by index-number:

```
>club_members$Age[1:3]   #Get the first three elements
[1] 37 34 26
```

Again, [1:3] is actually a vector:

```
>1:3
[1] 1 2 3
```

So writing club_members$Age[c(1,2,3)] would give the same thing.

It is annoying to write club_members$Age constantly. That is why we:

```
>attach(club_members)
```

Now we can use the vector names directly

```
>Age  #Much better
[1] 37 34 26 72 18 24
>Age[ Age > 30 ] #same as before, easier to read
[1] 37 34 72
```

Since the [criterion] is just a vector, we can use values from one vector as a criterion for extracting values from another vector.
Let's get the firstnames of the club members older than thirty

```
>Firstname[ Age > 30 ]
[1] Alice   Poul    Theodor
```

Then get the age of people with the lastname Knudsen

```
>Age[ Lastname == 'Knudsen']
[1] 37 34
```

Note that 'Knudsen' is text, and must be surrounded by ''

How many Points do people under 30 have, on average?

```
>mean( Points[ Age < 30]  )
[1] 228
```

Note that functions use ( ) around their arguments

Technical detail: It is often easiest to understand an R statement 'from the inside'.
Look at:

```
>mean( Points[ Age < 30]  )
```

This really means:
  - First get the vector: Age < 30
  - Then get all the Points[ ] where the above vector is TRUE
  - Then take the mean( )

Hint: If your R commands fail, try testing them this way, from the inside and out. When you find the step that fails, you have found where the error is.

Exercises (answers are at the end of the document):
1) Get the last three values in Points
2) Get all Points above 200
3) Get the ages of all men in the club
4) What is the max( ) Points among men in the club? Among women?
5) What is the average age among men? Among women?
6) What is the name of the person with the lowest score?

Combining criteria:
Criteria can be combined with & (and), | (or) :
Get the lastnames of members who are over thirty, and have less than 300 points

```
Lastname[ Age > 30 & Points < 300]
[1] Knudsen Knudsen
```

Get the Firstname of anyone who is female, or younger than thirty

```
>Firstname[ Gender == 'F' | Age < 30]
[1] Alice   Jokum   Babette Lynette
```

Get the firstname of all men who are younger than 50 and have more than 300 points

```
>Firstname[ Gender == 'M' & Age < 50 & Points > 300]
[1] Jokum
```

Technical detail: How does the above work? As you know, the [criterion] must be a vector. If you look at Lastname[Age > 30 & Points < 300], both Age > 30 and Points < 300 are vectors. The combination [Age > 30 & Points < 300] is also a vector, which is TRUE only when both of the two individual vectors are true:

```
> Age > 30
[1]  TRUE  TRUE FALSE  TRUE FALSE FALSE
> Points < 300
[1]  TRUE  TRUE FALSE FALSE  TRUE  TRUE
> Age > 30 & Points < 300
[1]  TRUE  TRUE FALSE FALSE FALSE FALSE
```

Technical detail: You may have seen '&&' used instead of '&'. Do NOT do this. They are not the same.

Exercises:
7) Get the firstnames of all men older than thirty
8) Get the firstnames of women with the lastname Knudsen
9) Get the points for women younger than 25 with the lastname Lauridsen
10) Get the firstnames of anyone with the lastname Knudsen or Thorvaldsen

Extracting from a dataframe (or matrix):

All of the exercises so far have been on vectors. These are one-dimensional.
If we look at the whole dataframe, it is two-dimensional: It has rows and columns.

```
> club_members
  Firstname    Lastname Age Gender Points
1     Alice     Knudsen  37      F    278
2      Poul     Knudsen  34      M    242
3     Jokum    Jonassen  26      M    312
4   Theodor Thorvaldsen  72      M    740
5   Babette   Brodersen  18      F    177
6   Lynette   Lauridsen  24      F    195
```

So to index a whole dataframe (or matrix) we need to tell both which rows and which columns we want.
The syntax is: dataframename[rows , columns]

Get the first three rows of the first two columns

```
> club_members[1:3 , 1:2]  # Both 1:3  and  1:2 are
vectors
  Firstname Lastname
1     Alice  Knudsen
2      Poul  Knudsen
3     Jokum Jonassen
```

Get the Age and Points for the first three rows

```
> club_members[1:3 , c('Age','Points')]  #again, we use
two vectors
  Age Points
1  37    278
2  34    242
3  26    312
```

Get all of the last row

```
> club_members[6, ]
  Firstname  Lastname Age Gender Points
6   Lynette Lauridsen  24      F    195
```

Note that when we dont specify the columns, we get all of them!

Exercises:

11) Get the full names (firstname and lastname) for the first three rows
12) Get the full names (firstname and lastname) for everyone
13) Get the last two columns for the last two rows
14) Get the lastname and age for the last four rows

Both 'rows' and 'columns' can be expressions (criteria) just like we did for vectors!
Get full rows for all men

```
> club_members[ Gender == 'M',  ]
  Firstname    Lastname Age Gender Points
2      Poul     Knudsen  34      M    242
3     Jokum    Jonassen  26      M    312
4   Theodor Thorvaldsen  72      M    740
```

Again, we didn't specify the columns

Get only the names for all men

```
>club_members[ Gender == 'M', c('Firstname','Lastname')]
  Firstname    Lastname
2      Poul     Knudsen
3     Jokum    Jonassen
4   Theodor Thorvaldsen
```

Get the names for anyone with more than 250 points

```
> club_members[ Points > 250, c('Firstname','Lastname')]
  Firstname    Lastname
1     Alice     Knudsen
3     Jokum    Jonassen
4   Theodor Thorvaldsen
```

Get the names of members who are over thirty, and have less than 300 points

```
>club_members[Age > 30 & Points < 300,
 c('Firstname','Lastname')]
  Firstname Lastname
1     Alice  Knudsen
2      Poul  Knudsen
```

Note that the above command was split over two lines. If you do that, R will actually use a '+' prompt (instead of a '>') to show you that it is waiting for you to complete the command.

Get all information on anyone who is female, or younger than thirty

```
>club_members[ Gender == 'F' | Age < 30 , ]
  Firstname  Lastname Age Gender Points
1     Alice   Knudsen  37      F    278
3     Jokum  Jonassen  26      M    312
5   Babette Brodersen  18      F    177
6   Lynette Lauridsen  24      F    195
```

Get the names of all men who are younger than 50 and have more than 300 points

```
>club_members[Gender=='M' & Age < 50 & Points > 300, 1:2]
  Firstname Lastname
3     Jokum Jonassen
```

Notice that we used number indexes instead of names for the columns here.
You can choose whichever you prefer.

Exercises

15) Get the names for all women
16) Get the names of all men older than thirty
17) Get the age for anyone with more than 250 points
18) Get the firstname and age of anyone with the lastname Knudsen or Thorvaldsen
19) Get the names and age for all women with more than 180 points.

That's it! Now you are ready to enter the strange and wondrous world of High Throughput bioinformatics! *Bon Voyage...* ;-)

Answers for exercises
(There may be more than one way to do it. Any way that works is ok)

1) Get the last three values in Points

```
>Points[4:6]
[1] 740 177 195
#Alternative ways (check the R reference sheet for
details):
>Points[c(4,5,6)]
>Points[-(1:3)]  #negative indexing, anything other than
1:3
```

2) Get all Points above 200

```
>Points[ Points > 200]
[1] 278 242 312 740
```

3) Get the ages of all men in the club

```
>Age[ Gender == 'M']
```

4) What is the max( ) Points among men in the club? Among Women?

```
> max( Points[ Gender == 'M'] )
[1] 740
> max( Points[ Gender == 'F'] )
[1] 278
```

5) What is the average age among men? Among women?

```
> mean(Age[Gender=='M'])
[1] 44
> mean(Age[Gender=='F'])
[1] 26.33333
```

6) What is the Firstname of the person with the lowest score?

```
>Firstname[ Points == min(Points) ]
[1] Babette
```

7) Get the firstnames of all men older than thirty

```
> Firstname[Gender == 'M' & Age > 30]
[1] Poul    Theodor
```

8) Get the firstnames of women with the lastname Knudsen

```
> Firstname[Gender == 'F' & Lastname == 'Knudsen']
[1] Alice
```

9) Get the points for women younger than 25 with the lastname Lauridsen

```
> Points[Gender == 'F' & Age < 25 & Lastname ==
'Lauridsen']
[1] 195
```

10) Get the firstnames of anyone with the lastname Knudsen or Thorvaldsen

```
>Firstname[Lastname == 'Knudsen' | Lastname ==
'Thorvaldsen']
[1] Alice   Poul    Theodor
```

11) Get the full names (firstname and lastname) for the first three rows

```
> club_members[1:3 , c('Firstname','Lastname')]
  Firstname Lastname
1     Alice  Knudsen
2      Poul  Knudsen
3     Jokum Jonassen
```

12) Get the full names (firstname and lastname) for everyone

```
> club_members[ , c('Firstname','Lastname')]
  Firstname    Lastname
1     Alice     Knudsen
2      Poul     Knudsen
3     Jokum    Jonassen
4   Theodor Thorvaldsen
5   Babette   Brodersen
6   Lynette   Lauridsen
```

13) Get the last two columns for the last two rows

```
> club_members[c(5,6) , c(4,5)]
  Gender Points
5      F    177
6      F    195
#equally good:
> club_members[5:6 , c('Gender','Points')]
```

14) Get the lastname and age for the last four rows

```
> club_members[3:6 , c('Lastname','Age')]
     Lastname Age
3    Jonassen  26
4 Thorvaldsen  72
5   Brodersen  18
6   Lauridsen  24
#you could also do:
> club_members[-c(1,2) , c(2,3)]   #Using negative
indexing for the rows
```

15) Get the names for all women

```
> club_members[ Gender == 'F', c
('Firstname','Lastname') ]
  Firstname  Lastname
1     Alice   Knudsen
5   Babette Brodersen
6   Lynette Lauridsen
```

16) Get the names of all men older than thirty

```
> club_members[ Gender == 'M' & Age > 30 , c
('Firstname','Lastname') ]
  Firstname    Lastname
2      Poul     Knudsen
4   Theodor Thorvaldsen
```

17) Get the age for anyone with more than 250 points

```
> club_members[Points > 250 , c('Age')]
[1] 37 26 72
# Note that we are only getting a vector out here,
# so we might as well use the Age vector directly:
> Age[Points > 250]
[1] 37 26 72
#use whichever syntax you prefer...
```

18) Get the firstname and age of anyone with the lastname Knudsen or Thorvaldsen

```
> club_members[Lastname=='Knudsen' |
Lastname=='Thorvaldsen' , c(1,3)]
  Firstname Age
1     Alice  37
2      Poul  34
4   Theodor  72
```

19) Get the names and age for all women with more than 180 points.

```
> club_members[Gender == 'F' & Points > 180 , 1:3]
  Firstname  Lastname Age
1     Alice   Knudsen  37
6   Lynette Lauridsen  24
```