Subsetting and Sorting

Contents

1 Subsetting and Sorting

1.1 Subsetting -quick review

```
set.seed(13435)
X <- data.frame("var1"=sample(1:5),"var2"=sample(6:10),"var3"=sample(11:15))
X
```

var1	var2	var3
3	8	14
1	7	15
5	6	13
4	10	12
2	9	11

```
X <- X[sample(1:5),] # kind with random order?
X
```

	var1	var2	var3
5	2	9	11
4	4	10	12
1	3	8	14
2	1	7	15
3	5	6	13

```
X$var2[c(1,3)] = NA
X
```

	var1	var2	var3
5	2	NA	11
4	4	10	12
1	3	NA	14
2	1	7	15
3	5	6	13

```
X[,1] # get first col
```

```
## [1] 2 4 3 1 5
```

```
X[,"var1"] # get col with name
```

```
## [1] 2 4 3 1 5
```

```
X[1:2,"var2"] # get 1st and snd row of col named "var2"
```

```
## [1] NA 10
```

1.2 Logicals ands and ors

```
X[(X$var1 <= 3 & X$var3 > 11),] # get all rows that meets the condition
```

var1	var2	var3
3	NA	14
1	7	15

```
X[(X$var1 <=3 | X$var3 > 15),] # logical or
```

	var1	var2	var3
5	2	NA	11
1	3	NA	14
2	1	7	15

1.3 Dealing with missing values

```
X[which(X$var2 >8),]
```

	var1	var2	var3
4	4	10	12

1.4 Sorting

```
sort(X$var1)
```

```
## [1] 1 2 3 4 5
```

```
sort(X$var1, decreasing = T)
```

```
## [1] 5 4 3 2 1
```

```r
sort(X$var2, na.last=TRUE) # put NA values at the last
```

```
## [1]  6  7 10 NA NA
```

1.5 Ordering

```r
X[order(X$var1),]
```

	var1	var2	var3
2	1	7	15
5	2	NA	11
1	3	NA	14
4	4	10	12
3	5	6	13

```r
X[order(X$var1,X$var3),]
```

	var1	var2	var3
2	1	7	15
5	2	NA	11
1	3	NA	14
4	4	10	12
3	5	6	13

The order() function in R returns a permutation of the vector that rearranges it into ascending or sorted order.

So, X[order(X$var1, X$var3),] is reordering the rows of the data frame X first by the values of var1 in ascending order. For rows where var1 is equal, it will then order by var3.

The result will be a data frame where the rows are sorted first by var1, and then by var3.

1.6 Ordering with plyr

```r
library(plyr)
```

The pieces are the "ply" (similar to how plywood is layers of wood veneer) and the "r" represents the R programming language.

The plyr package is a set of tools for manipulating data in R. It provides a number of functions that make it easier to manipulate and reshape data.

```r
arrange(X,var1)
```

var1	var2	var3
1	7	15
2	NA	11
3	NA	14
4	10	12
5	6	13

```
arrange(X,desc(var1))
```

var1	var2	var3
5	6	13
4	10	12
3	NA	14
2	NA	11
1	7	15

1.7 Adding rows and columns

```
X$var4 <- rnorm(5)
X
```

	var1	var2	var3	var4
5	2	NA	11	-0.4150458
4	4	10	12	2.5437602
1	3	NA	14	1.5545298
2	1	7	15	-0.6192328
3	5	6	13	-0.9261035

```
Y <- cbind(X,rnorm(5))
Y
```

	var1	var2	var3	var4	rnorm(5)
5	2	NA	11	-0.4150458	-0.6654995
4	4	10	12	2.5437602	-0.0216674
1	3	NA	14	1.5545298	-0.1741195
2	1	7	15	-0.6192328	0.2390044
3	5	6	13	-0.9261035	-1.8324596

1.8 Notes and further resources

R programming in the Data Science Track Andrew Jaffe's lecture notes 1http://www.biostat.jhsph.edu/~ajaffe/lec_winterR/I