

3. R and Atomic Vectors

CT1100 - J. Duggan

R

- R's mission is to enable the best and most thorough exploration of data possible (Chambers 2008).
- It is a dialect of the S language, developed at Bell Laboratories
- ACM noted that S “will forever alter the way people analyze, visualize, and manipulate data”

```
v <- 1:10
```

```
v
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
summary(v)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00   3.25   5.50   5.50   7.75   10.00
```

R Studio IDE (also available through <https://rstudio.cloud>)

R Code

```
01 Introduction.R x
Source on Save
Go to file/function
Run
Source
83 s4<-sample(3,20,replace=TRUE)
84 s2=s4
85
86 # Gather data on the frequency of (whole) numbers in a vector
87 t1<-table(s1)
88 props<-prop.table(t1)
89
90
91
92
93
94
95
96
97
98
99
100
101 (Top Level)
R Script
```

Environment/State

Environment	History	Git
Global Environment	Import Dataset	
Values		
b1	logi [1:2] FALSE TRUE	
b2	logi [1:5] FALSE TRUE FALSE TRUE FALSE	
c1	chr [1:5] "Odd" "Even" "Odd" "Even" "Odd"	
ind	2L	
index	5L	
props	table [1:3(1d)] 0.4 0.3 0.3	
r	24	
s	num [1:101] 51.4 55 57.2 57.3 58.6 ...	
s1	int [1:20] 2 1 1 3 1 3 2 1 1 1 ...	
s2	int [1:20] 2 3 2 2 2 2 1 2 2 1 ...	

Interactive console

Files	Plots	Packages	Help	Viewer
New Folder	Delete	Rename	More	
Home	Desktop	GitHub	CTS102	01 Vectors
Name	Size	Modified		
..				
01 Introduction.R	1.9 KB	Sep 3, 2015, 2:23 PM		
01 Vectors.pdf	892 KB	Sep 3, 2015, 2:26 PM		

File System

Data Structures in R

	Homogenous	Heterogenous
1d	Atomic Vector	List
2d	Matrix	Data Frame/Tibble
nd	Array	

- The basic data structure in R is the Vector
- Vectors come in two flavours:
 - Atomic vectors
 - Lists
- With atomic vectors, all elements have the same type: logical, integer, double (numeric) or character

Atomic Vectors - Examples

```
dbl_var <- c(2.9, 3.1, 4.8)
```

```
dbl_var
```

```
## [1] 2.9 3.1 4.8
```

```
log_var <- c(TRUE, TRUE, FALSE, TRUE, FALSE)
```

```
log_var
```

```
## [1] TRUE TRUE FALSE TRUE FALSE
```

```
str_var <- c("Dublin", "London", "Edinburgh")
```

```
str_var
```

```
## [1] "Dublin"      "London"      "Edinburgh"
```

Useful Comparison - Excel

- Atomic vectors are similar to rows/columns in Excel
- Think of them as adjacent cells of data that can be processed

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2			Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
3		Clifden	196	106.3	78.9	99.2	64.5	50.1	58.9	146.3	117.5	117.9	160	183.4	
4		Dublin Airport	93.1	36.9	100	68.9	19.1	4.8	40	48	43.8	42.6	131.2	81	
5		Difference	102.9	69.4	-21.1	30.3	45.4	45.3	18.9	98.3	73.7	75.3	28.8	102.4	
6															
7			Sum	Max	Min	Mean	SD								
8		Clifden	1379	196	50.1	114.92	48.427								
9		Dublin Airport	709.4	131.2	4.8	59.117	36.48								
10															
11															

```
dublin_air
```

```
##      Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct
##  93.1   36.9  100.0   68.9   19.1    4.8   40.0   48.0   43.8   42.6
```

Summary of data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2			Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
3		Clifden	196	106.3	78.9	99.2	64.5	50.1	58.9	146.3	117.5	117.9	160	183.4	
4		Dublin Airport	93.1	36.9	100	68.9	19.1	4.8	40	48	43.8	42.6	131.2	81	
5		Difference	102.9	69.4	-21.1	30.3	45.4	45.3	18.9	98.3	73.7	75.3	28.8	102.4	
6															
7			Sum	Max	Min	Mean	SD								
8		Clifden	1379	196	50.1	114.92	48.427								
9		Dublin Airport	709.4	131.2	4.8	59.117	36.48								
10															
11															

```
summary(dublin_air)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      4.80   39.23   45.90   59.12   84.03  131.20
```

```
summary(clifden)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      50.1   75.3   111.9   114.9   149.7   196.0
```

Subsetting Atomic Vectors

- Subsetting data is a key activity in data science
- R's subsetting operators are powerful and fast
- For atomic vectors, the operator `[` is used
- In R, the index for a vector starts at 1

```
dublin_air[1]
```

```
## Jan  
## 93.1
```

```
dublin_air[1:6]
```

```
## Jan Feb Mar Apr May Jun  
## 93.1 36.9 100.0 68.9 19.1 4.8
```


Subsetting Vectors - (1) Positive Integer

Positive integers return elements at the specified position

```
clifden[10]
```

```
##      Oct
```

```
## 117.9
```

```
clifden[3:6]
```

```
##   Mar   Apr   May   Jun
```

```
## 78.9 99.2 64.5 50.1
```

Subsetting Vectors - (2) Using character vectors

Return elements with matching names

```
clifden[1:6]
```

```
##      Jan      Feb      Mar      Apr      May      Jun
## 196.0 106.3   78.9   99.2   64.5   50.1
```

```
clifden["Apr"]
```

```
##      Apr
##    99.2
```

```
clifden[c("Apr", "Jan")]
```

```
##      Apr      Jan
##    99.2 196.0
```

Vectorisation

- A powerful feature of R is that it supports vectorisation
- For example, we can subtract two vectors - just like Excel.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2			Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	
3		Clifden	196	106.3	78.9	99.2	64.5	50.1	58.9	146.3	117.5	117.9	160	183.4	
4		Dublin Airport	93.1	36.9	100	68.9	19.1	4.8	40	48	43.8	42.6	131.2	81	
5		Difference	102.9	69.4	-21.1	30.3	45.4	45.3	18.9	98.3	73.7	75.3	28.8	102.4	
6															
7			Sum	Max	Min	Mean	SD								
8		Clifden	1379	196	50.1	114.92	48.427								
9		Dublin Airport	709.4	131.2	4.8	59.117	36.48								
10															
11															

```
clifden[1:9] - dublin_air[1:9]
```

```
##      Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep
## 102.9   69.4  -21.1   30.3   45.4   45.3   18.9   98.3   73.7
```

Logical Vectors

- Logical vectors can be generated using conditional expressions, and for filtering

```
mean(clifden)
```

```
## [1] 114.9167
```

```
clifden1 <- clifden[1:6]
```

```
clifden1 > mean(clifden)
```

```
##   Jan   Feb   Mar   Apr   May   Jun
```

```
## TRUE FALSE FALSE FALSE FALSE FALSE
```

```
clifden1[clifden1 > mean(clifden)]
```

```
## Jan
```

```
## 196
```

Summary

- Atomic vectors, a key type in R. Very similar to rows/columns ($N \times 1$ or $1 \times N$) of data in Excel
- All elements are the same type (coercion)
- Different ways to filter, including logical vectors
- A data frame can be viewed as a set of atomic vectors bound together

```
head(match)
```

```
## # A tibble: 6 x 9
##   Time   Half Team   Scorer           Number From   Type
##   <dbl> <dbl> <chr>  <chr>           <dbl> <chr>  <chr>
## 1     1     1  Dublin Paul Mannion         13 Play   Point
## 2     2     1  Kerry  Sean O'Shea         11 Play   Point
## 3     3     1  Dublin Dean Rock          14 Play   Point
## 4     4     1  Dublin Dean Rock          14 Free   Point
## 5    10     1  Kerry  David Clifford       13 Play   Point
## 6    13     1  Kerry  Sean O'Shea         11 FortyFive Point
```