

## Types of data storage in R

Vector: The most basic type of unidimensional data structure that contains element of the same type e.g. logical, integer, double, or character. The type and length of a vector can be checked with the commands `typeof()` and `length()` respectively.

```
df<- seq(1, 10, by=1.5)
typeof(df)
[1] "double"
length(df)
[1] 7
```

Matrix: A homogeneous collection of rectangular two-dimensional datasets with fixed number of rows and columns that allow easy operations e.g. addition, multiplication. All columns are of same type.

```
> df <- matrix(1:16, nrow = 4, dimnames = list(c("this","is","really",
"great"), c("rstudio","is","my", "boss")))
> df
```

	rstudio	is	my	boss
this	1	5	9	13
is	2	6	10	14
really	3	7	11	15
great	4	8	12	16

Matrices can also be formed out of vectors using the `rbind` and `cbind` function:

```
c1=seq(1, 10, by=1.5)
c2=seq(2, 20, by=1.5)
c3=seq(3, 30, by=1.5)
df<-cbind(c1, c2, c3)
```

```
> head(df)
      c1  c2  c3
[1,] 1.0 2.0 3.0
[2,] 2.5 3.5 4.5
[3,] 4.0 5.0 6.0
[4,] 5.5 6.5 7.5
[5,] 7.0 8.0 9.0
[6,] 8.5 9.5 10.5
```

Data frames: It is used for storing two-dimensional data tables arranged in Excel like row and column format. A df is the same as a matrix except that it can contain dissimilar data types (e.g.numeric, character or factor) in multiple columns.

```
# Create a df with different types of variables
days <- c('sunday','monday','tuesday','wednesday', 'thurshday')
weeks <- c(1, 2,3 , 4, 5)
years <- c(2000,2005,2008,2009, 2012)
countries <- c('Atrich', 'Swiss', 'France', 'Ireland', 'Ecosse')
df <- data.frame(days,weeks,years,countries)
df
```

Now check the class of the variables using the sapply function:

```
sapply(dff, class)
      days      weeks      years  countries
"character" "numeric" "numeric" "character"
```

Tibbles: A more modern alternative of df with some advantages: it doesn't automatically convert strings to factors, doesn't create row names, and must have same column lengths (unlike dataframes). Printing a tibble:

```
class(iris)
[1] "data.frame"
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1          3.5          1.4          0.2  setosa
2           4.9          3.0          1.4          0.2  setosa
3           4.7          3.2          1.3          0.2  setosa
4           4.6          3.1          1.5          0.2  setosa
5           5.0          3.6          1.4          0.2  setosa
6           5.4          3.9          1.7          0.4  setosa

> library(tibble)
> iris1<-as_tibble(iris)
> head(iris1)
# A tibble: 6 × 5
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
      <dbl>      <dbl>      <dbl>      <dbl> <fct>
1         5.1         3.5         1.4         0.2 setosa
2         4.9         3          1.4         0.2 setosa
3         4.7         3.2         1.3         0.2 setosa
4         4.6         3.1         1.5         0.2 setosa
5         5          3.6         1.4         0.2 setosa
```

