> Sometimes data require more complex storage than simple vectors, and thankfully R provides a host of data structures. The most common are the data.frame, matrix and list, followed by the array.
>
> — Jared P. Lander (Lander, 2017)

Data Science for Operational Researchers using R

04 – Data Frames and Tibbles

https://github.com/JimDuggan/Data-Science-for-OR

The data frame

- A data frame is based on a list, where the elements of that list containing equal length vectors
- The list elements then become columns in the data frame
- Created with the function data.frame()
- The data frame, with its row and column structure, will be familiar to anyone who has used a spreadsheet, where each column is a variable, and every row is an observation
- It can also be operated on using matrix notation (a 2-dimensional vector)

```
d <- data.frame(Number=1:5,
                Letter=LETTERS[1:5],
                Flag=c(T,F,T,F,NA),
                stringsAsFactors = F)
d
#>   Number Letter  Flag
#> 1      1      A  TRUE
#> 2      2      B FALSE
#> 3      3      C  TRUE
#> 4      4      D FALSE
#> 5      5      E    NA
```

```
summary(d)
#>      Number     Letter             Flag
#>  Min.   :1   Length:5           Mode :logical
#>  1st Qu.:2   Class :character   FALSE:2
#>  Median :3   Mode  :character   TRUE :2
#>  Mean   :3                      NA's :1
#>  3rd Qu.:4
#>  Max.   :5
```

Subset examples

```
d[1:2,]
#>   Number Letter  Flag
#> 1      1      A  TRUE
#> 2      2      B FALSE
```

```
d[d$Flag == T,]
#>    Number Letter Flag
#> 1       1      A TRUE
#> 3       3      C TRUE
#> NA     NA   <NA>   NA
```

```
d[1:2,c("Letter","Flag")]
#>   Letter  Flag
#> 1      A  TRUE
#> 2      B FALSE
```

Adding columns (with $)

```
d1 <- d
d1$letter <- letters[1:5]
d1
#>   Number Letter  Flag letter
#> 1      1      A  TRUE      a
#> 2      2      B FALSE      b
#> 3      3      C  TRUE      c
#> 4      4      D FALSE      d
#> 5      5      E    NA      e
```

Adding columns using transform()

```
df1 <- subset(mtcars,mpg>32,select=c("mpg","disp"))
df1 <- transform(df1,kpg=mpg*1.6)
df1
#>                  mpg disp   kpg
#> Fiat 128        32.4 78.7 51.84
#> Toyota Corolla 33.9 71.1 54.24
```

The subset() function

subset(x, subset,select) returns subsets of vectors, matrices or data frames which meet specified conditions.

The main arguments provided to this function when using with data frames are:

- x, the object to be subsetted
- subset, a logical expression indicating which rows should be kept
- select, which indicates the columns to be selected from the data frame. If this is not present, all columns are returned.

```
subset(mtcars,mpg>32,select=c("mpg","disp"))
#>                  mpg disp
#> Fiat 128        32.4 78.7
#> Toyota Corolla 33.9 71.1
```

Challenge 4.1

- Using the subset function to list all cars with an mpg greater than the mean from the data frame mtcars

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Challenge 4.2

- Add a new column to mtcars (using $) that converts mpg to kpg. Assume a constant 1.6 for the transformation.

```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb   kpg
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4 33.60
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4 33.60
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1 36.48
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1 34.24
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2 29.92
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1 28.96
```

Introducing tibbles

Tibbles are data frames, however they alter some data frame behaviours to make working with packages in the tidyverse a little easier

- Printing, where tibbles by default only show the first 10 rows, and limit the visible columns to those than fit on the screen
- Subsetting, where a tibble is always returned, and also partial matching is not supported

```
library(tibble)
```

```
d1 <- tibble(Number=1:5,
             Letter=LETTERS[1:5],
             Flag=c(T,F,T,F,NA))
d1
#> # A tibble: 5 x 3
#>   Number Letter Flag
#>    <int> <chr>  <lgl>
#> 1      1 A      TRUE
#> 2      2 B      FALSE
#> 3      3 C      TRUE
#> 4      4 D      FALSE
#> 5      5 E      NA
```

Moving between both

```
str(as_tibble(d))
#> tibble [5 x 3] (S3: tbl_df/tbl/data.frame)
#>  $ Number: int [1:5] 1 2 3 4 5
#>  $ Letter: chr [1:5] "A" "B" "C" "D" ...
#>  $ Flag  : logi [1:5] TRUE FALSE TRUE FALSE NA
str(as.data.frame(d1))
#> 'data.frame':    5 obs. of  3 variables:
#>  $ Number: int  1 2 3 4 5
#>  $ Letter: chr  "A" "B" "C" "D" ...
#>  $ Flag  : logi  TRUE FALSE TRUE FALSE NA
```

Using data frames in a pipeline

```
mtcars_1 <- mtcars |>                                  # the original data frame
            subset(select=c("mpg","disp")) |>          # select 2 columns
            transform(kpg=mpg*1.6,                     # Add first column
                      dm_ratio=disp/mpg) |>            # Add second column
            head()                                     # Subset 1st 6 records
mtcars_1
#>                   mpg disp   kpg dm_ratio
#> Mazda RX4          21.0  160 33.60    7.619
#> Mazda RX4 Wag      21.0  160 33.60    7.619
#> Datsun 710         22.8  108 36.48    4.737
#> Hornet 4 Drive     21.4  258 34.24   12.056
#> Hornet Sportabout 18.7  360 29.92   19.251
#> Valiant            18.1  225 28.96   12.431
```

Challenge 4.3

Use the `subset()` function to generate the following tibbles from the tibble `ggplot2::mpg`. Use the R pipe operator (`|>`) where necessary.

```
# The car with the maximum displacement, with a subset of features
max_displ
#> # A tibble: 1 x 6
#>   manufacturer model     year displ   cty class
#>   <chr>        <chr>    <int> <dbl> <int> <chr>
#> 1 chevrolet    corvette  2008     7    15 2seater
```

```
# All 2seater cars, with selected columns
two_seater
#> # A tibble: 5 x 6
#>   class   manufacturer model    displ  year   hwy
#>   <chr>   <chr>        <chr>    <dbl> <int> <int>
#> 1 2seater chevrolet    corvette   5.7  1999    26
#> 2 2seater chevrolet    corvette   5.7  1999    23
#> 3 2seater chevrolet    corvette   6.2  2008    26
#> 4 2seater chevrolet    corvette   6.2  2008    25
#> 5 2seater chevrolet    corvette   7    2008    24
```