Lesson 02
Data Structures
Lists and Arrays

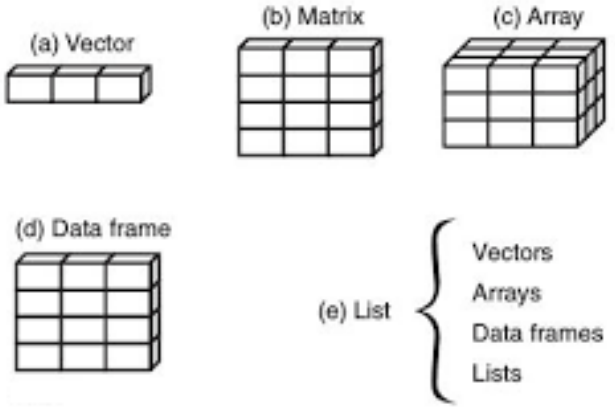
Outline

- Lists
  - Construction
  - Accessing elements
  - Basic attributes
  - Containment
- Arrays
  - Construction
  - Accessing elements

Lists - Definition

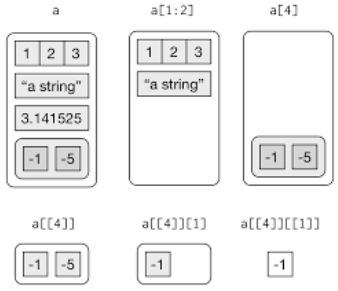
- `list` is a data structure designed to hold a variety of objects of different data types
  - One list member can be `character`, another a `numeric`, yet another a `vector` or even a `data.frame`
  - Lists can include another `list` or lists facilitating containment and recursion
- List containment
  - When one list contains another list as one of its elements
- List recursion
  - When one list references itself through one of its elements

Lists – Financial Data



- List data provided in
  - Variables
  - Vectors
  - Data frames
- Open **`Struct2_Lists_Arrays.r`**
- Data on Target Corporation
  - `character`: Ticker symbol
  - `numeric`: Current stock price
  - `vector`: Company name, Headquarters
  - `data.frame`: Stock prices 6M, 1Y and 5Y ago

Lists – Construction



- Create TGT list

```
tgt_list <- list(tgt_tic, tgt_pr, tgt_vec, tgt_df)
```

- Basic list attributes

`length(tgt_list)` returns `4`

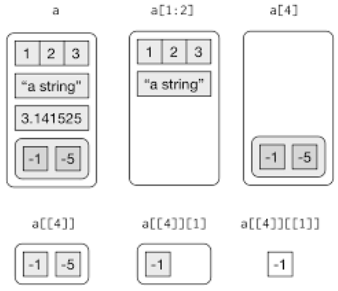
- Accessing individual list elements: [] vs. [[]]

```
# [[]] returns data frame object
tgt_list[[4]]
```

`class(tgt_list[[4]])` returns `data.frame`

```
# First row, second column are TGT price 6M ago
```

`tgt_list[[4]][1,2]` returns `183.58`



Lists – Accessing Elements

- Accessing individual list elements: [] vs. [[]]

```
#[] returns list object
tgt_list[4]
class(tgt_list[4])
```

- Data frame treated as 1-element list

```
# tgt_list[4][1]
```

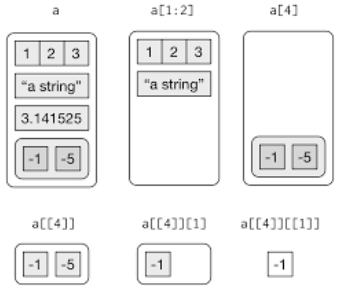
- Could go back to double bracketing [[]]

```
tgt_list[4][[1]]        # Back to being a data frame
class(tgt_list[4][[1]])
```

- If [] required, better to complement with names

```
tgt_list[4][1]$HistPrice$Price[1]
```

Lists – Containment



- Redefine TGT list by including AMZN its last element

```
tgt_list <- list(tgt_tic, tgt_pr,
                 tgt_vec, tgt_df, amzn_list)
tgt_list
length(agg_list)
```

- Accessing elements of the contained list

```
# Access Amazon info through Target list
# Second element of Amazon sublist is the price
tgt_list[[5]][2]
```

- Compare Target and Amazon 5 year returns

Arrays – Construction & Access

- Arrays are multidimensional matrices with data of the same type
  - Example: Sales cube by store, product, promotion and time (4D array)
- Create 3D array of number of stores by size, region and company

```
# Three-dimensional array of stores by region, size and company
# 3D array is filled by traversing the 1st dim, then 2nd and finally 3rd
region <- c("Northeast","South","Midwest","Southwest","West")
size <- c("Small","Medium","Large")
company <- c("BBY","TGT")
stores_array <- array(
  c(52,68,70,51,70,        # Small BBY
    74,76,65,57,51,        # Medium BBY
    59,98,83,38,15,        # Large BBY
    93,75,87,95,66,        # Small TGT
    82,65,75,72,61,        # Medium TGT
    128,261,266,53,307),   # Large TGT
  dim=c(5,3,2),
  dimnames = list(region, size, company)
)
stores_array
stores_array[2,3,1]    # South Large BBY stores
```

Summary

- Worked with `list` data structure
  - Can be used to build repositories of complex structured and semi-structured data
  - Accessing elements with double-brackets [[]]
    - Assures the element of proper data structure
  - Facilitates containment and recursion (see lectures)
- Introduced `array` data structure
  - Used to represent multidimensional data of the same type
- Discussed main concepts for both data structures
  - Construction, attributes, access and operations