Lists

> Created by Woo Hyun (Ray) Hwang

Lists

Creating Lists

Lists are the most common type of data in R. Data frames are also a type of a list.

```
members <- list(leaders = c("park", "choi"), assistants = "kang",
                workers = c("lee", "kim", "hong", "song"))
members #3 components : leaders, assistants, workers, each a character string
class(members)
names(members)
```

Lists are also a type of vector, so it can be made using vector function.

```
z <- vector(mode="list")
z[["abc"]] <- 1
z
```

List indexing

Basics are using `[[]]`.

```
members[[1]] #refers to the sections : a vector
members[[2]]
members[1] #refers to the subsection : a list
members[2]
class(members[[1]])
class(members[1])
#Better to index using names
members["leaders"]
members$leaders #this is the same as members[1]
```

List updating

```
members$leaders[1] <- "huh" #changing the first element of leaders to huh
members[[1]][2] <- "choi jr." #changing the 2nd element of leaders to choi jr
members
```

Creating a list of lists

```
salaries <- list(leaders = c(250, 200), assistant = 100,
                 members = c(300, 200, 180, 120, 100))
team <- list(m=members, s=salaries) #list of 2 diff. lists
team
```

Adding/Deleting/Combining elements on a list

```
#Adding elements
z <- list(a="abc", b=12)
z
z$c <- "sailing" #adding subsection c, adding sailing to c
z
z[[4]] <- 28 #can add using index, add 4th subsection with element 28
z[5:7] <- c(FALSE, TRUE, TRUE) #note that it is [5:7] not [[5:7]]
z
#Deleting elements : add as NULL
z$b <- NULL #deletes z$b
length(z) #length of the list
#Combining lists
c(list("Joe", 55000, T), list(5))
```

`lapply()` and `sapply()`

Applies the same function to all elements on list. Just like `apply()` on matrices, `lapply()` does the same to lists. It returns a list.

```
lapply(salaries, median)
```

`sapply()` does the same calculation as `lapply()`, but returns a simple format such as a vector or a matrix.

```
sapply(salaries, median)
class(sapply(salaries, median)) #it is a numeric vector with names
```

`vapply()` can be used to get results with specific format.

```
vapply(salaries, range, c(max=0, min=0))
```

Creating a list of lists with `c()`

```
team <- c(m=members, s=salaries) #different format from list()
team
class(team); length(list)
```

`unlist()`

`unlist()` destroys the structure of the list.

```
#median(salaries) #produces an error
unlist(salaries)
class(unlist(salaries)) #it becomes a numeric vector with names
median(unlist(salaries))
```

Lists inside a list

```
b <- list(u=5, v=12)
c <- list(w=13)
a <- list(b, c)
length(a)
```

Applications: making a summary function `summarize()`

The objective is to provide summary of (mean, sd, min, Q1, med, Q3, max, outliers).

```
summarize <- function(x) {
  mean.sd <- c(mean=mean(x), sd=sd(x))
  fivenum <- quantile(x, seq(0, 1, 0.25))
  iqr <- fivenum[4]-fivenum[2]
  lowerbound <- fivenum[2]-1.5*iqr
  upperbound <- fivenum[4]+1.5*iqr
  outliers <- c(x[x>upperbound], x[x<lowerbound])
  list <- list(mean.sd=mean.sd, fivenum=fivenum, outliers=outliers)
  return(list)
}
set.seed(1)
x <- rnorm(1000)
summarize(x)
class(summarize(x)) #it is a list

#We want to round the summary to 2 digits
lapply(summarize(x), round, 2)
```

Application : Card sorting

4 sets of 52 playing cards (208 in total) with families S, D, H, C Choosing 52 cards without replacement Want to figure the order of families

```
kinds <- c("S", "D", "H", "C")
set.seed(1)
cards <- sample(rep(kinds, 13*4), 52, replace=F)
cards
order <- list(S=NULL, D=NULL, H=NULL, C=NULL)
for(i in 1:52) {
  r <- cards[i]
```

```
  order[[r]] <- c(order[[r]], i)
}
order
sapply(order, length)
length(order)
length(unlist(order))
```