Programming: more loop examples

Stat 133 with Gaston Sanchez

Traversing a matrix

(or any 2-dim object)

Writing a function

In this slides, we provide several examples of loops and similar operations applied on matrices (or any other 2-dimensional object)

The purpose is to give you a taste of how to use loops (at least conceptually) on rectangular objects.

Traversing cells of a matrix

	col1	*col2*	*col3*
row1	x_{11}	x_{12}	x_{13}
row2	x_{21}	x_{22}	x_{23}
row3	x_{31}	x_{32}	x_{33}
	etc	*etc*	*etc*

```
# looping through a matrix
X <- matrix(runif(12), nrow = 4, ncol = 3)

# looping row by row
for (i in 1:nrow(X)) {
  print(X[i, ])
}
```

```r
# looping through a matrix
X <- matrix(runif(12), nrow = 4, ncol = 3)

# looping row by row
for (i in 1:nrow(X)) {
   print(X[i, ])
}

# looping column by column
for (j in 1:ncol(X)) {
   print(X[ ,j])
}
```

```r
# looping cell by cell
for (i in 1:nrow(X)) {
   for (j in 1:ncol(X)) {
      print(X[i,j])
   }
}
```

```r
# looping cell by cell
for (i in 1:nrow(X)) {
  for (j in 1:ncol(X)) {
    print(X[i,j])
  }
}

# looping cell by cell (equivalent)
for (j in 1:ncol(X)) {
  for (i in 1:nrow(X)) {
    print(X[i,j])
  }
}
```

Row Sums

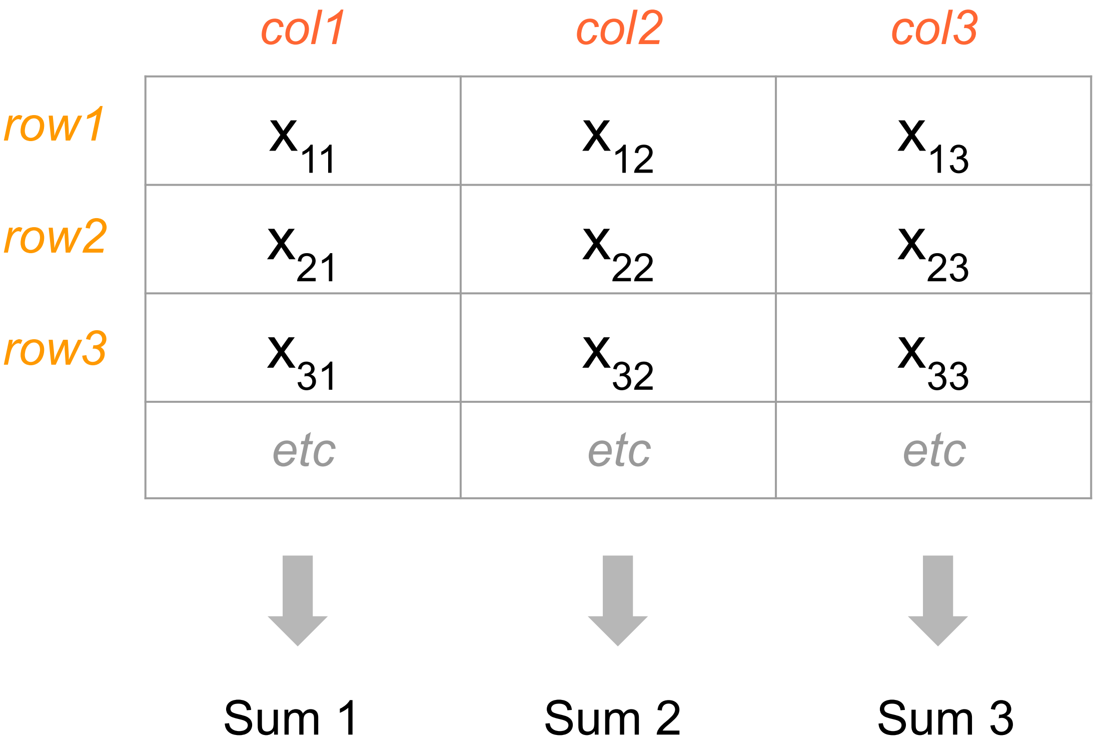
Row Sums

	col1	*col2*	*col3*	
row1	x_{11}	x_{12}	x_{13}	→ Sum 1
row2	x_{21}	x_{22}	x_{23}	→ Sum 2
row3	x_{31}	x_{32}	x_{33}	→ Sum 3
	etc	*etc*	*etc*	

```
# vector of row sums
row_sums <- rep(0, nrow(X))

for (i in 1:nrow(X)) {
  aux <- 0
  for (j in 1:ncol(X)) {
    aux <- aux + X[i,j]
  }
  row_sums[i] <- aux
}
```

Column Sums

Column Sums



	col1	*col2*	*col3*
row1	x_{11}	x_{12}	x_{13}
row2	x_{21}	x_{22}	x_{23}
row3	x_{31}	x_{32}	x_{33}
	etc	*etc*	*etc*

Sum 1 Sum 2 Sum 3

```
# vector of column sums
col_sums <- rep(0, ncol(X))

for (j in 1:ncol(X)) {
  aux <- 0
  for (i in 1:nrow(X)) {
    aux <- aux + X[i,j]
  }
  col_sums[j] <- aux
}
```

Built-in functions:

- `rowSums(X)`
- `colSums(X)`

Column Means

Column means

	col1	*col2*	*col3*
row1	x_{11}	x_{12}	x_{13}
row2	x_{21}	x_{22}	x_{23}
row3	x_{31}	x_{32}	x_{33}
	etc	*etc*	*etc*
	↓	↓	↓
	$\bar{X}_1$	$\bar{X}_2$	$\bar{X}_3$

```
X <- matrix(runif(12), nrow = 4, ncol = 3)

x1_mean <- mean(X[ ,1])
x2_mean <- mean(X[ ,2])
x3_mean <- mean(X[ ,3])
```

What if you had many more columns?

```
# one option
X <- matrix(runif(12), nrow = 4, ncol = 3)

x_means <- c()
for (j in 1:ncol(X)) {
  x_means <- c(x_means, mean(X[, j]))
}

x_means
```

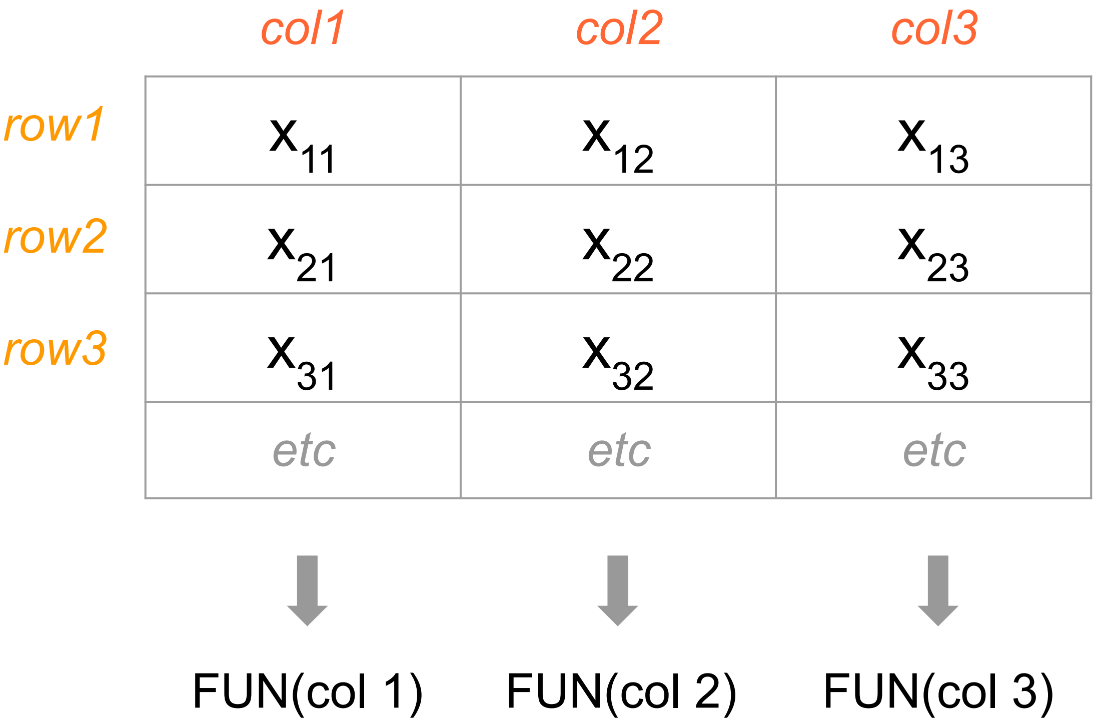
```
# another option
X <- matrix(runif(12), nrow = 4, ncol = 3)

x_means <- rep(0, ncol(X))
for (j in 1:ncol(X)) {
  x_means[j] <- mean(X[, j])
}

x_means
```

Built-in functions:

- `rowMeans(X)`
- `colMeans(X)`

Apply functions

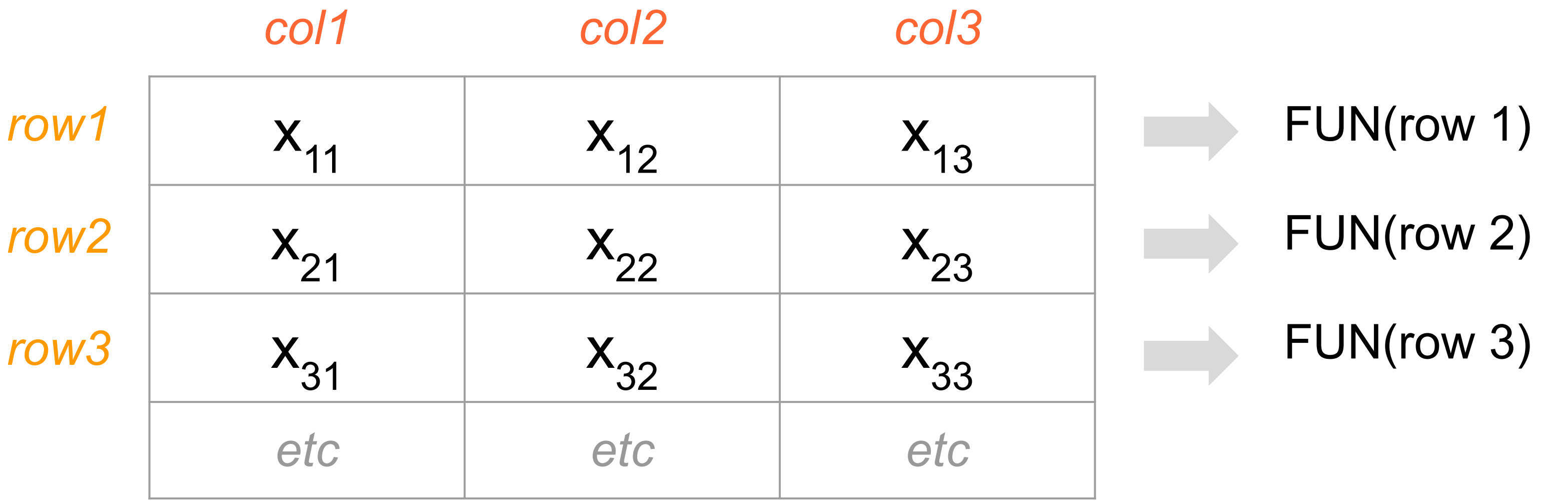
Function applied to columns



	col1	*col2*	*col3*
row1	x_{11}	x_{12}	x_{13}
row2	x_{21}	x_{22}	x_{23}
row3	x_{31}	x_{32}	x_{33}
	etc	*etc*	*etc*

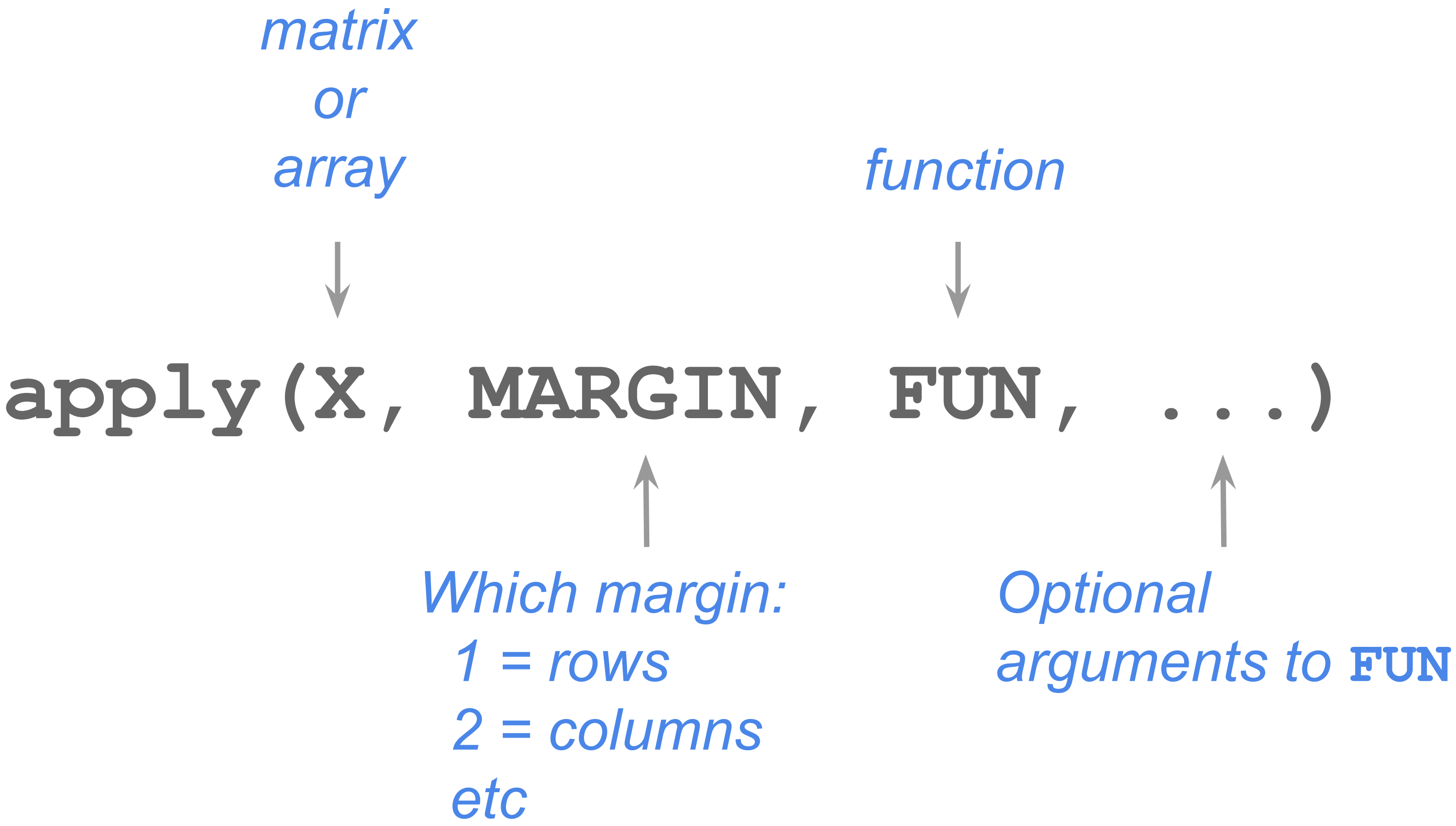
FUN(col 1) FUN(col 2) FUN(col 3)

Function applied to rows



	col1	*col2*	*col3*	
row1	x_{11}	x_{12}	x_{13}	FUN(row 1)
row2	x_{21}	x_{22}	x_{23}	FUN(row 2)
row3	x_{31}	x_{32}	x_{33}	FUN(row 3)
	etc	*etc*	*etc*	

```
apply(X, MARGIN, FUN, ...)
```



matrix
or
array

function

```
apply(X, MARGIN, FUN, ...)
```

Which margin:
1 = rows
2 = columns
etc

Optional
arguments to **`FUN`**

```
X <- matrix(runif(12), nrow = 4, ncol = 3)

# row sums
apply(X, 1, sum)

# row sums removing missing values
apply(X, 1, sum, na.rm = TRUE)
```

```r
X <- matrix(runif(12), nrow = 4, ncol = 3)

# row minima
apply(X, 1, min)

# row minima removing missing values
apply(X, 1, min, na.rm = TRUE)
```

```
X <- matrix(runif(12), nrow = 4, ncol = 3)

# column maxima
apply(X, 2, max)

# column maxima removing missing values
apply(X, 2, max, na.rm = TRUE)
```

More statistics

```
stats <- function(y, na.rm = FALSE) {
  c('min' = min(x, na.rm = na.rm),
    'avg' = mean(x, na.rm = na.rm),
    'max' = max(x, na.rm = na.rm))
}

stats(1:10)

apply(X, 2, stats)
```