Week 4

Fixing character vectors

```r
if(!file.exists("./Data")){dir.create("./Data")}
fileUrl <- "https://data.baltimorecity.gov/api/views/dz54-2aru/rows.csv?accessType=DOWNLOAD"
download.file(fileUrl,destfile="./Data/cameras.csv", method="curl")
cameraData <- read.csv("./Data/cameras.csv")
```

```r
names(cameraData)
```

```
## [1] "address"                    "direction"
## [3] "street"                     "crossStreet"
## [5] "intersection"               "Location.1"
## [7] "X2010.Census.Neighborhoods"  "X2010.Census.Wards.Precincts"
## [9] "Zip.Codes"
```

tolower() toupper()

```r
tolower(names(cameraData)) #rendo minuscole le lettere
```

```
## [1] "address"                    "direction"
## [3] "street"                     "crossstreet"
## [5] "intersection"               "location.1"
## [7] "x2010.census.neighborhoods"  "x2010.census.wards.precincts"
## [9] "zip.codes"
```

```r
toupper(names(cameraData)) #rendo maiuscole le lettere
```

```
## [1] "ADDRESS"                    "DIRECTION"
## [3] "STREET"                     "CROSSSTREET"
## [5] "INTERSECTION"               "LOCATION.1"
## [7] "X2010.CENSUS.NEIGHBORHOODS"  "X2010.CENSUS.WARDS.PRECINCTS"
## [9] "ZIP.CODES"
```

strsplit

```
splitNames <- strsplit(names(cameraData), "\\.")#tolgo il punto dal nome
splitNames[[6]]
```

```
## [1] "Location" "1"
```

List

```
myList <- list(letters = c("A", "b", "c"), numbers = 1:3 , matrix(1:25, ncol=5))
head(myList)
```

```
## $letters
## [1] "A" "b" "c"
##
## $numbers
## [1] 1 2 3
##
## [[3]]
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    6   11   16   21
## [2,]    2    7   12   17   22
## [3,]    3    8   13   18   23
## [4,]    4    9   14   19   24
## [5,]    5   10   15   20   25
```

```
myList[1]; myList$letters ; myList[[1]]
```

```
## $letters
## [1] "A" "b" "c"
```

```
## [1] "A" "b" "c"
```

```
## [1] "A" "b" "c"
```

Sapplay

```
splitNames[[6]][1];splitNames[[6]][2]
```

```
## [1] "Location"
```

```
## [1] "1"
```

Posso creare una funzione che mi restituisca solo il primo elemento di un vettore es("location","1")>("location")

```
firstelement <- function(x){x[1]}
sapply(splitNames, firstelement)
```

```
## [1] "address"      "direction"    "street"       "crossStreet"  "intersection"
## [6] "Location"     "X2010"        "X2010"        "Zip"
```

sub

per sosituire una cosa

```
names(cameraData)
```

```
## [1] "address"                        "direction"
## [3] "street"                         "crossStreet"
## [5] "intersection"                   "Location.1"
## [7] "X2010.Census.Neighborhoods"    "X2010.Census.Wards.Precincts"
## [9] "Zip.Codes"
```

```
sub("[.]","",names(cameraData))
```

```
## [1] "address"                       "direction"
## [3] "street"                        "crossStreet"
## [5] "intersection"                  "Location1"
## [7] "X2010Census.Neighborhoods"    "X2010Census.Wards.Precincts"
## [9] "ZipCodes"
```

```
sub("X","",names(cameraData))
```

```
## [1] "address"                       "direction"
## [3] "street"                        "crossStreet"
## [5] "intersection"                  "Location.1"
## [7] "2010.Census.Neighborhoods"    "2010.Census.Wards.Precincts"
## [9] "Zip.Codes"
```

Regular expression

cercare parole o lettere in testi

Date class

```
d2 = Sys.Date()
d2
```

```
## [1] "2020-03-04"
```

```
d1 = date()
d1
```

```
## [1] "Wed Mar  4 10:48:32 2020"
```

Formatting dates

```
format(d2,"%a %b %d")
```

```
## [1] "mer mar 04"
```

Creating Dates

```
x = c("1gen1960","2gen1960"); z=as.Date(x,"%d%b%Y")
z
```

```
## [1] "1960-01-01" "1960-01-02"
```

```
z[1]-z[2]
```

```
## Time difference of -1 days
```

```
as.numeric(z[1]-z[2])
```

```
## [1] -1
```