ST 558-651 Homework 1

James Carr

5/22/2021

Subsetting Vectors

```r
#creating vector elements 2 to 12 counting by 2
vec <- seq(2, 12, by=2)

#here we print only the odd indexes
vec[c(1, 3, 5)]
```

```
## [1]  2  6 10
```

```r
#print without first element
vec[-1]
```

```
## [1]  4  6  8 10 12
```

```r
#print without first and third element
vec[c(-1, -3)]
```

```
## [1]  4  8 10 12
```

```r
#print all values but the last value
vec[-length(vec)]
```

```
## [1]  2  4  6  8 10
```

```r
#save the vector in reverse
revVec <- rev(vec)

#print first element 5 times
vec[rep(1, 5)]
```

```
## [1] 2 2 2 2 2
```

Subsetting Matrices

```r
#matrix w/ 4 rows, 3 cols, of uniform data between 0 and 1
unifMat <- matrix(runif(12, 0, 1),nrow=4, ncol=3)

#odd rows replaced by 2 times value
unifMat[c(1, 3),] <- unifMat[c(1, 3),] * 2

#even rows replaced by 1/2 times value
unifMat[c(2, 4),] <- unifMat[c(2, 4),] * 0.5

#convert vec to matrix and print dimensions
matVec <- as.matrix(vec)
dim(matVec)
```

```
## [1] 6 1
```

```r
#using the drop command to subset
drop_true <- unifMat[1, ,drop=TRUE]
drop_false <- unifMat[1, ,drop=FALSE]
```

Using drop = true seems to return the values as a vector

Subsetting Data Frames

```r
#showing 3 ways to access the 4th column of the data set
names(iris)[4]
```

```
## [1] "Petal.Width"
```

```r
iris$Petal.Width
```

```
##   [1] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 0.2 0.2 0.1 0.1 0.2 0.4 0.4 0.3
##  [19] 0.3 0.3 0.2 0.4 0.2 0.5 0.2 0.2 0.4 0.2 0.2 0.2 0.2 0.4 0.1 0.2 0.2 0.2
##  [37] 0.2 0.1 0.2 0.2 0.3 0.3 0.2 0.6 0.4 0.3 0.2 0.2 0.2 0.2 1.4 1.5 1.5 1.3
##  [55] 1.5 1.3 1.6 1.0 1.3 1.4 1.0 1.5 1.0 1.4 1.3 1.4 1.5 1.0 1.5 1.1 1.8 1.3
##  [73] 1.5 1.2 1.3 1.4 1.4 1.7 1.5 1.0 1.1 1.0 1.2 1.6 1.5 1.6 1.5 1.3 1.3 1.3
##  [91] 1.2 1.4 1.2 1.0 1.3 1.2 1.3 1.3 1.1 1.3 2.5 1.9 2.1 1.8 2.2 2.1 1.7 1.8
## [109] 1.8 2.5 2.0 1.9 2.1 2.0 2.4 2.3 1.8 2.2 2.3 1.5 2.3 2.0 2.0 1.8 2.1 1.8
## [127] 1.8 1.8 2.1 1.6 1.9 2.0 2.2 1.5 1.4 2.3 2.4 1.8 1.8 2.1 2.4 2.3 1.9 2.3
## [145] 2.5 2.3 1.9 2.0 2.3 1.8
```

```r
iris[,4]
```

```
##   [1] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 0.2 0.2 0.1 0.1 0.2 0.4 0.4 0.3
##  [19] 0.3 0.3 0.2 0.4 0.2 0.5 0.2 0.2 0.4 0.2 0.2 0.2 0.2 0.4 0.1 0.2 0.2 0.2
##  [37] 0.2 0.1 0.2 0.2 0.3 0.3 0.2 0.6 0.4 0.3 0.2 0.2 0.2 0.2 1.4 1.5 1.5 1.3
##  [55] 1.5 1.3 1.6 1.0 1.3 1.4 1.0 1.5 1.0 1.4 1.3 1.4 1.5 1.0 1.5 1.1 1.8 1.3
##  [73] 1.5 1.2 1.3 1.4 1.4 1.7 1.5 1.0 1.1 1.0 1.2 1.6 1.5 1.6 1.5 1.3 1.3 1.3
##  [91] 1.2 1.4 1.2 1.0 1.3 1.2 1.3 1.3 1.1 1.3 2.5 1.9 2.1 1.8 2.2 2.1 1.7 1.8
## [109] 1.8 2.5 2.0 1.9 2.1 2.0 2.4 2.3 1.8 2.2 2.3 1.5 2.3 2.0 2.0 1.8 2.1 1.8
## [127] 1.8 1.8 2.1 1.6 1.9 2.0 2.2 1.5 1.4 2.3 2.4 1.8 1.8 2.1 2.4 2.3 1.9 2.3
## [145] 2.5 2.3 1.9 2.0 2.3 1.8
```

```
iris[,"Petal.Width"]
```

```
##   [1] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 0.2 0.2 0.1 0.1 0.2 0.4 0.4 0.3
##  [19] 0.3 0.3 0.2 0.4 0.2 0.5 0.2 0.2 0.4 0.2 0.2 0.2 0.2 0.4 0.1 0.2 0.2 0.2
##  [37] 0.2 0.1 0.2 0.2 0.3 0.3 0.2 0.6 0.4 0.3 0.2 0.2 0.2 0.2 1.4 1.5 1.5 1.3
##  [55] 1.5 1.3 1.6 1.0 1.3 1.4 1.0 1.5 1.0 1.4 1.3 1.4 1.5 1.0 1.5 1.1 1.8 1.3
##  [73] 1.5 1.2 1.3 1.4 1.4 1.7 1.5 1.0 1.1 1.0 1.2 1.6 1.5 1.6 1.5 1.3 1.3 1.3
##  [91] 1.2 1.4 1.2 1.0 1.3 1.2 1.3 1.3 1.1 1.3 2.5 1.9 2.1 1.8 2.2 2.1 1.7 1.8
## [109] 1.8 2.5 2.0 1.9 2.1 2.0 2.4 2.3 1.8 2.2 2.3 1.5 2.3 2.0 2.0 1.8 2.1 1.8
## [127] 1.8 1.8 2.1 1.6 1.9 2.0 2.2 1.5 1.4 2.3 2.4 1.8 1.8 2.1 2.4 2.3 1.9 2.3
## [145] 2.5 2.3 1.9 2.0 2.3 1.8
```

```
#create extra column whose entrys are all 1s
iris$Extra.Col <- 1

#printing out mtcars columns in sorted order
mtcars <- mtcars[,sort(names(mtcars))]
mtcars
```

```
##                     am carb cyl  disp drat gear  hp  mpg  qsec vs    wt
## Mazda RX4            1    4   6 160.0 3.90    4 110 21.0 16.46  0 2.620
## Mazda RX4 Wag        1    4   6 160.0 3.90    4 110 21.0 17.02  0 2.875
## Datsun 710           1    1   4 108.0 3.85    4  93 22.8 18.61  1 2.320
## Hornet 4 Drive       0    1   6 258.0 3.08    3 110 21.4 19.44  1 3.215
## Hornet Sportabout    0    2   8 360.0 3.15    3 175 18.7 17.02  0 3.440
## Valiant              0    1   6 225.0 2.76    3 105 18.1 20.22  1 3.460
## Duster 360           0    4   8 360.0 3.21    3 245 14.3 15.84  0 3.570
## Merc 240D            0    2   4 146.7 3.69    4  62 24.4 20.00  1 3.190
## Merc 230             0    2   4 140.8 3.92    4  95 22.8 22.90  1 3.150
## Merc 280             0    4   6 167.6 3.92    4 123 19.2 18.30  1 3.440
## Merc 280C            0    4   6 167.6 3.92    4 123 17.8 18.90  1 3.440
## Merc 450SE           0    3   8 275.8 3.07    3 180 16.4 17.40  0 4.070
## Merc 450SL           0    3   8 275.8 3.07    3 180 17.3 17.60  0 3.730
## Merc 450SLC          0    3   8 275.8 3.07    3 180 15.2 18.00  0 3.780
## Cadillac Fleetwood   0    4   8 472.0 2.93    3 205 10.4 17.98  0 5.250
## Lincoln Continental  0    4   8 460.0 3.00    3 215 10.4 17.82  0 5.424
## Chrysler Imperial    0    4   8 440.0 3.23    3 230 14.7 17.42  0 5.345
## Fiat 128             1    1   4  78.7 4.08    4  66 32.4 19.47  1 2.200
## Honda Civic          1    2   4  75.7 4.93    4  52 30.4 18.52  1 1.615
## Toyota Corolla       1    1   4  71.1 4.22    4  65 33.9 19.90  1 1.835
## Toyota Corona        0    1   4 120.1 3.70    3  97 21.5 20.01  1 2.465
## Dodge Challenger     0    2   8 318.0 2.76    3 150 15.5 16.87  0 3.520
## AMC Javelin          0    2   8 304.0 3.15    3 150 15.2 17.30  0 3.435
## Camaro Z28           0    4   8 350.0 3.73    3 245 13.3 15.41  0 3.840
## Pontiac Firebird     0    2   8 400.0 3.08    3 175 19.2 17.05  0 3.845
## Fiat X1-9            1    1   4  79.0 4.08    4  66 27.3 18.90  1 1.935
## Porsche 914-2        1    2   4 120.3 4.43    5  91 26.0 16.70  0 2.140
## Lotus Europa         1    2   4  95.1 3.77    5 113 30.4 16.90  1 1.513
## Ford Pantera L       1    4   8 351.0 4.22    5 264 15.8 14.50  0 3.170
## Ferrari Dino         1    6   6 145.0 3.62    5 175 19.7 15.50  0 2.770
## Maserati Bora        1    8   8 301.0 3.54    5 335 15.0 14.60  0 3.570
## Volvo 142E           1    2   4 121.0 4.11    4 109 21.4 18.60  1 2.780
```

```r
#removing column called Sepal.Width
iris$Sepal.Width <- NULL
names(iris)
```

```
## [1] "Sepal.Length" "Petal.Length" "Petal.Width"  "Species"      "Extra.Col"
```

```r
#replacing all periods in column name with underscores
names(iris) <- gsub('\\.', '_', names(iris))
names(iris)
```

```
## [1] "Sepal_Length" "Petal_Length" "Petal_Width"  "Species"      "Extra_Col"
```

Subsetting Lists

```r
#creating list 1 and showing 2 ways to access the 2nd element
lst1 <- list(a = vec, b = revVec, c = unifMat)
lst1$b
```

```
## [1] 12 10  8  6  4  2
```

```r
lst1[[2]]
```

```
## [1] 12 10  8  6  4  2
```

```r
#creating a 2nd list with list 1 as its only element
lst2 <- list(lst1)
#subsetting list 2 to access b in list 1
lst2[[1]]$b
```

```
## [1] 12 10  8  6  4  2
```

```r
#test what happens if only using 1 set of square brackets
test <- lst1[1]
test2 <- lst1[[1]]
str(test)
```

```
## List of 1
##  $ a: num [1:6] 2 4 6 8 10 12
```

```r
str(test2)
```

```
##  num [1:6] 2 4 6 8 10 12
```

```r
test[1]
```

```
## $a
## [1]  2  4  6  8 10 12
```

```r
test2[1]
```

```
## [1] 2
```

When using only 1 bracket, it returns the vector as a list of 1 and inside of the list of 1 is the element you want to look at. If you use both brackets, it does not return a list of 1, but it returns the element.

This could be useful if you want to subset a list to combine with another list, I suppose.