

AFS 505 HW3

Aaron Appleby

2022-09-12

#HW3

1. Simple Lists

a) list 1

```
# Create a list containing different vectors and matrices.
##create a list with the **list()** function.
# we can construct a list by stringing together multiple objects
a = "Soils 502"
b = c(1, 3, 4, 5)
c = matrix(seq(1:6),3,2, byrow = T)
```

```
my_list = list(a, b, c)
# contains copies of a, b, c
my_list
```

```
## [[1]]
## [1] "Soils 502"
##
## [[2]]
## [1] 1 3 4 5
##
## [[3]]
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
## [3,]    5    6
```

Addressing allows for concise data reshaping, extraction and assignment in R.

```
# Extract matrix from list 1.
my_list[3]
```

```
## [[1]]
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
## [3,]    5    6
```

```
cat("\n")
```

```
# Extract a and c twice  
my_list[-2]
```

```
## [[1]]  
## [1] "Soils 502"  
##  
## [[2]]  
##      [,1] [,2]  
## [1,]    1    2  
## [2,]    3    4  
## [3,]    5    6
```

1.

b. list 2

```
# Named List within a list
```

```
name <- list>Last= "Williams", First= "Serena")  
name
```

```
## $Last  
## [1] "Williams"  
##  
## $First  
## [1] "Serena"
```

```
record <- c(801, 136)  
#now assign names for meaning, notice these are character strings  
names(record) <- c("wins", "losses")  
record
```

```
##    wins losses  
##    801    136
```

```
grand.slam.wins <- list(Australian.Open= c(2003, 2005, 2007, 2009, 2010, 2015, 2017), French.Open= c(2003, 2005, 2007, 2009, 2010, 2015, 2017), Wimbledon.Open= c(2003, 2005, 2007, 2009, 2010, 2015, 2017), US.Open= c(2003, 2005, 2007, 2009, 2010, 2015, 2017))  
cat("\n")
```

```
tennis_list <- list(name=name, age=c(36), record=record, grand.slam.wins=grand.slam.wins)  
tennis_list
```

```
## $name  
## $name$Last  
## [1] "Williams"  
##  
## $name$First  
## [1] "Serena"
```

```
##
##
## $age
## [1] 36
##
## $record
##   wins losses
##   801    136
##
## $grand.slam.wins
## $grand.slam.wins$Australian.Open
## [1] 2003 2005 2007 2009 2010 2015 2017
##
## $grand.slam.wins$French.Open
## [1] 2002 2013 2015
##
## $grand.slam.wins$Wimbeldon
## [1] 2002 2003 2009 2010 2012 2015 2016
##
## $grand.slam.wins$US.Open
## [1] 1999 2002 2008 2012 2013 2014
```

```
cat("\n")
```

```
str(grand.slam.wins)
```

```
## List of 4
## $ Australian.Open: num [1:7] 2003 2005 2007 2009 2010 ...
## $ French.Open    : num [1:3] 2002 2013 2015
## $ Wimbeldon      : num [1:7] 2002 2003 2009 2010 2012 ...
## $ US.Open        : num [1:6] 1999 2002 2008 2012 2013 ...
```

grand.slam.wins is not structured as a data frame or matrix because it was input as a list composed of 4 numeric vectors.

2. cbind, rbind

Addressing applies to matrices and data frames in much the same way as to vectors.

```
# 'data' is a vector
# 'nrow' gives the number of rows
# 'ncol' gives the number of columns
# 'byrow' by default is FALSE, so data is read in column-wise
v1 <- c(3, 5, 6, 12)
v1
```

```
## [1] 3 5 6 12
```

```
cat("\n")
```

```
v2 <- c(T, F, T, T)
v2
```

```
## [1] TRUE FALSE TRUE TRUE
```

```
cat("\n")
```

```
v3 <- c("A", "B", "C", "D", "E")
v3
```

```
## [1] "A" "B" "C" "D" "E"
```

```
cat("\n")
```

```
mat <- matrix(data = seq(2, 32, by = 3), nrow = 4, ncol = 5)
```

```
## Warning in matrix(data = seq(2, 32, by = 3), nrow = 4, ncol = 5): data length
## [11] is not a sub-multiple or multiple of the number of rows [4]
```

```
##default is byrow = F <- fills columns first
mat
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    2   14   26    5   17
## [2,]    5   17   29    8   20
## [3,]    8   20   32   11   23
## [4,]   11   23    2   14   26
```

```
##because there are more spots in the matrix than the length of the sequence, the sequence starts over
```

```
cat("\n")
```

```
mat1 <- matrix(data = c(1, -1, 1, 0, 0, 0, -1, 1), 2, 4, T)
mat1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1   -1    1    0
## [2,]    0    0   -1    1
```

```
##do not need to specify nrow = , ncol = , or byrow = ...can just put number of rows,columns,T
```

2. Matrices

a) cbind

create matrices using the vectors and matrices created above.

```
# Extract a subsets from the previous matrices and bind them into a single matrix.
##avoid naming a column by extracting vector values
```

```
cbind(v1[1:4],v2)
```

```
##          v2
## [1,]   3  1
## [2,]   5  0
## [3,]   6  1
## [4,]  12  1
```

```
cat("\n")
```

```
m3<- matrix(cbind(v3,v2), 5, 2)
```

```
## Warning in cbind(v3, v2): number of rows of result is not a multiple of vector
## length (arg 2)
```

```
m3
```

```
##      [,1] [,2]
## [1,] "A"  "TRUE"
## [2,] "B"  "FALSE"
## [3,] "C"  "TRUE"
## [4,] "D"  "TRUE"
## [5,] "E"  "TRUE"
```

```
##as before, because the matrix is larger than the sequence of v2, the sequence starts over until it runs out
```

```
cat("\n")
```

```
cbind(mat,v1[1:4])
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   2  14  26   5  17   3
## [2,]   5  17  29   8  20   5
## [3,]   8  20  32  11  23   6
## [4,]  11  23   2  14  26  12
```

```
##by just using cbind() the 6th column is named v1, the name is removed by calling the vector values us
```

2. Matrices

c) rbind

create matrices using the vectors and matrices used above, but this time bind them by rows rather than by columns

```
rbind(mat[1:4,1:4],mat1)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2   14   26    5
## [2,]    5   17   29    8
## [3,]    8   20   32   11
## [4,]   11   23    2   14
## [5,]    1   -1    1    0
## [6,]    0    0   -1    1
```

```
cat("\n")
```

```
rbind(v1[1:4],mat1)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    3    5    6   12
## [2,]    1   -1    1    0
## [3,]    0    0   -1    1
```

```
cat("\n")
```

```
rbind(v1,v2,v3)
```

```
## Warning in rbind(v1, v2, v3): number of columns of result is not a multiple of
## vector length (arg 1)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## v1 "3"    "5"    "6"   "12"  "3"
## v2 "TRUE" "FALSE" "TRUE" "TRUE" "TRUE"
## v3 "A"    "B"    "C"    "D"    "E"
```

3.

a) Read in Iris data

```
iris <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data", header=F)
head(iris)
```

```
##      V1 V2 V3 V4      V5
## 1 5.1 3.5 1.4 0.2 Iris-setosa
## 2 4.9 3.0 1.4 0.2 Iris-setosa
## 3 4.7 3.2 1.3 0.2 Iris-setosa
## 4 4.6 3.1 1.5 0.2 Iris-setosa
## 5 5.0 3.6 1.4 0.2 Iris-setosa
## 6 5.4 3.9 1.7 0.4 Iris-setosa
```

```

#first row is not a header <- added header=F
##need to name columns
colnames(iris) <- c("sepal.length.(cm)",
                    "sepal.width.(cm)",
                    "petal.length.(cm)",
                    "petal.width.(cm)",
                    "class")
###tried to add a little style for ease of reading... Enter automatically spaces/lines text
head(iris)

```

```

##   sepal.length.(cm) sepal.width.(cm) petal.length.(cm) petal.width.(cm)
## 1                5.1                3.5                1.4                0.2
## 2                4.9                3.0                1.4                0.2
## 3                4.7                3.2                1.3                0.2
## 4                4.6                3.1                1.5                0.2
## 5                5.0                3.6                1.4                0.2
## 6                5.4                3.9                1.7                0.4
##           class
## 1 Iris-setosa
## 2 Iris-setosa
## 3 Iris-setosa
## 4 Iris-setosa
## 5 Iris-setosa
## 6 Iris-setosa

```

3.

b) summaries

```

iris$class<-as.factor(iris$class)
i1 <- iris[iris$class == "Iris-versicolor",]
##the , after "Iris-versicolor" specifies the column
summary(i1)

```

```

##   sepal.length.(cm) sepal.width.(cm) petal.length.(cm) petal.width.(cm)
## Min.   :4.900      Min.   :2.000      Min.   :3.00      Min.   :1.000
## 1st Qu.:5.600      1st Qu.:2.525      1st Qu.:4.00      1st Qu.:1.200
## Median :5.900      Median :2.800      Median :4.35      Median :1.300
## Mean   :5.936      Mean   :2.770      Mean   :4.26      Mean   :1.326
## 3rd Qu.:6.300      3rd Qu.:3.000      3rd Qu.:4.60      3rd Qu.:1.500
## Max.   :7.000      Max.   :3.400      Max.   :5.10      Max.   :1.800
##           class
## Iris-setosa      : 0
## Iris-versicolor:50
## Iris-virginica   : 0
##
##
##

```

```
head(i1)
```

```
##      sepal.length.(cm) sepal.width.(cm) petal.length.(cm) petal.width.(cm)
## 51          7.0          3.2          4.7          1.4
## 52          6.4          3.2          4.5          1.5
## 53          6.9          3.1          4.9          1.5
## 54          5.5          2.3          4.0          1.3
## 55          6.5          2.8          4.6          1.5
## 56          5.7          2.8          4.5          1.3
##              class
## 51 Iris-versicolor
## 52 Iris-versicolor
## 53 Iris-versicolor
## 54 Iris-versicolor
## 55 Iris-versicolor
## 56 Iris-versicolor
```

```
cat("\n")
```

```
i2 <- iris[iris$`petal.width.(cm)` > 1.5 | iris$`sepal.width.(cm)` > 3.0, ]
```

##I am not sure why the column names needed quotes ' ' when class did not... perhaps because of the ()

```
i2
```

```
##      sepal.length.(cm) sepal.width.(cm) petal.length.(cm) petal.width.(cm)
## 1          5.1          3.5          1.4          0.2
## 3          4.7          3.2          1.3          0.2
## 4          4.6          3.1          1.5          0.2
## 5          5.0          3.6          1.4          0.2
## 6          5.4          3.9          1.7          0.4
## 7          4.6          3.4          1.4          0.3
## 8          5.0          3.4          1.5          0.2
## 10         4.9          3.1          1.5          0.1
## 11         5.4          3.7          1.5          0.2
## 12         4.8          3.4          1.6          0.2
## 15         5.8          4.0          1.2          0.2
## 16         5.7          4.4          1.5          0.4
## 17         5.4          3.9          1.3          0.4
## 18         5.1          3.5          1.4          0.3
## 19         5.7          3.8          1.7          0.3
## 20         5.1          3.8          1.5          0.3
## 21         5.4          3.4          1.7          0.2
## 22         5.1          3.7          1.5          0.4
## 23         4.6          3.6          1.0          0.2
## 24         5.1          3.3          1.7          0.5
## 25         4.8          3.4          1.9          0.2
## 27         5.0          3.4          1.6          0.4
## 28         5.2          3.5          1.5          0.2
## 29         5.2          3.4          1.4          0.2
## 30         4.7          3.2          1.6          0.2
## 31         4.8          3.1          1.6          0.2
## 32         5.4          3.4          1.5          0.4
## 33         5.2          4.1          1.5          0.1
## 34         5.5          4.2          1.4          0.2
## 35         4.9          3.1          1.5          0.1
## 36         5.0          3.2          1.2          0.2
```


## 37	5.5	3.5	1.3	0.2
## 38	4.9	3.1	1.5	0.1
## 40	5.1	3.4	1.5	0.2
## 41	5.0	3.5	1.3	0.3
## 43	4.4	3.2	1.3	0.2
## 44	5.0	3.5	1.6	0.6
## 45	5.1	3.8	1.9	0.4
## 47	5.1	3.8	1.6	0.2
## 48	4.6	3.2	1.4	0.2
## 49	5.3	3.7	1.5	0.2
## 50	5.0	3.3	1.4	0.2
## 51	7.0	3.2	4.7	1.4
## 52	6.4	3.2	4.5	1.5
## 53	6.9	3.1	4.9	1.5
## 57	6.3	3.3	4.7	1.6
## 66	6.7	3.1	4.4	1.4
## 71	5.9	3.2	4.8	1.8
## 78	6.7	3.0	5.0	1.7
## 84	6.0	2.7	5.1	1.6
## 86	6.0	3.4	4.5	1.6
## 87	6.7	3.1	4.7	1.5
## 101	6.3	3.3	6.0	2.5
## 102	5.8	2.7	5.1	1.9
## 103	7.1	3.0	5.9	2.1
## 104	6.3	2.9	5.6	1.8
## 105	6.5	3.0	5.8	2.2
## 106	7.6	3.0	6.6	2.1
## 107	4.9	2.5	4.5	1.7
## 108	7.3	2.9	6.3	1.8
## 109	6.7	2.5	5.8	1.8
## 110	7.2	3.6	6.1	2.5
## 111	6.5	3.2	5.1	2.0
## 112	6.4	2.7	5.3	1.9
## 113	6.8	3.0	5.5	2.1
## 114	5.7	2.5	5.0	2.0
## 115	5.8	2.8	5.1	2.4
## 116	6.4	3.2	5.3	2.3
## 117	6.5	3.0	5.5	1.8
## 118	7.7	3.8	6.7	2.2
## 119	7.7	2.6	6.9	2.3
## 121	6.9	3.2	5.7	2.3
## 122	5.6	2.8	4.9	2.0
## 123	7.7	2.8	6.7	2.0
## 124	6.3	2.7	4.9	1.8
## 125	6.7	3.3	5.7	2.1
## 126	7.2	3.2	6.0	1.8
## 127	6.2	2.8	4.8	1.8
## 128	6.1	3.0	4.9	1.8
## 129	6.4	2.8	5.6	2.1
## 130	7.2	3.0	5.8	1.6
## 131	7.4	2.8	6.1	1.9
## 132	7.9	3.8	6.4	2.0
## 133	6.4	2.8	5.6	2.2
## 136	7.7	3.0	6.1	2.3

## 137	6.3	3.4	5.6	2.4
## 138	6.4	3.1	5.5	1.8
## 139	6.0	3.0	4.8	1.8
## 140	6.9	3.1	5.4	2.1
## 141	6.7	3.1	5.6	2.4
## 142	6.9	3.1	5.1	2.3
## 143	5.8	2.7	5.1	1.9
## 144	6.8	3.2	5.9	2.3
## 145	6.7	3.3	5.7	2.5
## 146	6.7	3.0	5.2	2.3
## 147	6.3	2.5	5.0	1.9
## 148	6.5	3.0	5.2	2.0
## 149	6.2	3.4	5.4	2.3
## 150	5.9	3.0	5.1	1.8
##	class			
## 1	Iris-setosa			
## 3	Iris-setosa			
## 4	Iris-setosa			
## 5	Iris-setosa			
## 6	Iris-setosa			
## 7	Iris-setosa			
## 8	Iris-setosa			
## 10	Iris-setosa			
## 11	Iris-setosa			
## 12	Iris-setosa			
## 15	Iris-setosa			
## 16	Iris-setosa			
## 17	Iris-setosa			
## 18	Iris-setosa			
## 19	Iris-setosa			
## 20	Iris-setosa			
## 21	Iris-setosa			
## 22	Iris-setosa			
## 23	Iris-setosa			
## 24	Iris-setosa			
## 25	Iris-setosa			
## 27	Iris-setosa			
## 28	Iris-setosa			
## 29	Iris-setosa			
## 30	Iris-setosa			
## 31	Iris-setosa			
## 32	Iris-setosa			
## 33	Iris-setosa			
## 34	Iris-setosa			
## 35	Iris-setosa			
## 36	Iris-setosa			
## 37	Iris-setosa			
## 38	Iris-setosa			
## 40	Iris-setosa			
## 41	Iris-setosa			
## 43	Iris-setosa			
## 44	Iris-setosa			
## 45	Iris-setosa			
## 47	Iris-setosa			

```
## 48      Iris-setosa
## 49      Iris-setosa
## 50      Iris-setosa
## 51  Iris-versicolor
## 52  Iris-versicolor
## 53  Iris-versicolor
## 57  Iris-versicolor
## 66  Iris-versicolor
## 71  Iris-versicolor
## 78  Iris-versicolor
## 84  Iris-versicolor
## 86  Iris-versicolor
## 87  Iris-versicolor
## 101  Iris-virginica
## 102  Iris-virginica
## 103  Iris-virginica
## 104  Iris-virginica
## 105  Iris-virginica
## 106  Iris-virginica
## 107  Iris-virginica
## 108  Iris-virginica
## 109  Iris-virginica
## 110  Iris-virginica
## 111  Iris-virginica
## 112  Iris-virginica
## 113  Iris-virginica
## 114  Iris-virginica
## 115  Iris-virginica
## 116  Iris-virginica
## 117  Iris-virginica
## 118  Iris-virginica
## 119  Iris-virginica
## 121  Iris-virginica
## 122  Iris-virginica
## 123  Iris-virginica
## 124  Iris-virginica
## 125  Iris-virginica
## 126  Iris-virginica
## 127  Iris-virginica
## 128  Iris-virginica
## 129  Iris-virginica
## 130  Iris-virginica
## 131  Iris-virginica
## 132  Iris-virginica
## 133  Iris-virginica
## 136  Iris-virginica
## 137  Iris-virginica
## 138  Iris-virginica
## 139  Iris-virginica
## 140  Iris-virginica
## 141  Iris-virginica
## 142  Iris-virginica
## 143  Iris-virginica
## 144  Iris-virginica
```

```
## 145 Iris-virginica
## 146 Iris-virginica
## 147 Iris-virginica
## 148 Iris-virginica
## 149 Iris-virginica
## 150 Iris-virginica
```

```
summary(i2)
```

```
##  sepal.length.(cm) sepal.width.(cm) petal.length.(cm) petal.width.(cm)
##  Min.    :4.400      Min.    :2.500      Min.    :1.000      Min.    :0.100
##  1st Qu.:5.100      1st Qu.:3.000      1st Qu.:1.500      1st Qu.:0.200
##  Median :5.900      Median :3.200      Median :4.800      Median :1.600
##  Mean   :5.947      Mean   :3.233      Mean   :3.753      Mean   :1.243
##  3rd Qu.:6.700      3rd Qu.:3.400      3rd Qu.:5.500      3rd Qu.:2.000
##  Max.    :7.900      Max.    :4.400      Max.    :6.900      Max.    :2.500
##
##      class
##  Iris-setosa      :42
##  Iris-versicolor:10
##  Iris-virginica  :47
##
##
##
```

```
cat("\n")
```

```
#separate by class and look at petal width
i3 <- iris[iris$class == "Iris-setosa",]
summary(i3)
```

```
##  sepal.length.(cm) sepal.width.(cm) petal.length.(cm) petal.width.(cm)
##  Min.    :4.300      Min.    :2.300      Min.    :1.000      Min.    :0.100
##  1st Qu.:4.800      1st Qu.:3.125      1st Qu.:1.400      1st Qu.:0.200
##  Median :5.000      Median :3.400      Median :1.500      Median :0.200
##  Mean   :5.006      Mean   :3.418      Mean   :1.464      Mean   :0.244
##  3rd Qu.:5.200      3rd Qu.:3.675      3rd Qu.:1.575      3rd Qu.:0.300
##  Max.    :5.800      Max.    :4.400      Max.    :1.900      Max.    :0.600
##
##      class
##  Iris-setosa      :50
##  Iris-versicolor: 0
##  Iris-virginica  : 0
##
##
##
```

```
i4 <- iris[iris$class == "Iris-virginica",]
summary(i4)
```

```
##  sepal.length.(cm) sepal.width.(cm) petal.length.(cm) petal.width.(cm)
##  Min.    :4.900      Min.    :2.200      Min.    :4.500      Min.    :1.400
##  1st Qu.:6.225      1st Qu.:2.800      1st Qu.:5.100      1st Qu.:1.800
```

```
## Median :6.500      Median :3.000      Median :5.550      Median :2.000
## Mean   :6.588      Mean    :2.974      Mean    :5.552      Mean    :2.026
## 3rd Qu.:6.900      3rd Qu.:3.175      3rd Qu.:5.875      3rd Qu.:2.300
## Max.   :7.900      Max.    :3.800      Max.    :6.900      Max.    :2.500
##           class
## Iris-setosa      : 0
## Iris-versicolor: 0
## Iris-virginica  :50
##
##
##
```

Iris petal width increases in alphabetical order (the order listed in summary()).
As iris petal width decreases, sepal width increases.

4. Cars

a) Read in cars data

```
car<- read.table("car.txt", header=T, stringsAsFactors=T)
##because car data is .txt not .csv and separated by something different than a comma, data can not be read
##data consists entirely of categorical data, stringsAsFactors=T turns all data into factors
head(car)
```

```
## price.buy price.maint doors seats trunk safety rating
## 1      high      med      2 more med high acc
## 2     vhigh     high      3  2 small low unacc
## 3      med     high 5more      4 small low unacc
## 4      high      med      2  4 small high acc
## 5      high      med      2 more small med unacc
## 6      low     vhigh      2  2 big low unacc
```

```
summary(car)
```

```
## price.buy price.maint doors seats trunk safety
## high :451 high :451 2 :451 2 :599 big :601 high:598
## low :446 low :452 3 :452 4 :600 med :601 low :594
## med :452 med :447 4 :450 more:601 small:598 med :601
## vhigh:451 vhigh:447 5more:447 NA's: 7
## NA's : 3
## rating
## acc : 407
## good : 71
## unacc:1256
## vgood: 66
##
```

4.

b) remove missing values

```
#remove rows with NA positions
dim(na.omit(car))
```

```
## [1] 1790    7
```

```
summary(na.omit(car))
```

```
## price.buy price.maint doors seats trunk safety
## high :447 high :449 2 :449 2 :595 big :598 high:596
## low :443 low :451 3 :452 4 :595 med :599 low :593
## med :451 med :446 4 :445 more:600 small:593 med :601
## vhigh:449 vhigh:444 5more:444
## rating
## acc : 406
## good : 71
## unacc:1247
## vgood: 66
```

```
##na.omit removes rows with NA
```

```
# dim(car[!is.na(car), ])
##for some reason this added rows (a lot of rows)
# dim(car[is.na(car),])
##shows 10 entries with NA (that can be removed by na.omit())
```

4.

c) remove duplicates

```
summary(car[!duplicated(car), ])
```

```
## price.buy price.maint doors seats trunk safety
## high :432 high :432 2 :432 2 :576 big :576 high:573
## low :432 low :432 3 :432 4 :576 med :576 low :573
## med :432 med :431 4 :432 more:576 small:576 med :575
## vhigh:432 vhigh:430 5more:432 NA's: 7
## NA's : 3
## rating
## acc : 384
## good : 69
## unacc:1210
## vgood: 65
##
```

```
dim(car[!duplicated(car), ])
```

```
## [1] 1728    7
```