Untitled

Chase Darlington

September 25, 2018

1. (6 pts) Create the following vectors, populated with information about the courses for which you are enrolled this year in addition to one course (any course) that you are not enrolled.

- courseNum: course number of each course
- coursename: course name of each course
- courseProf: name of the instructor for each course
- enrolled: a logical vector indicating whether or not you are enrolled in the course
- anticipatedGrade: your anticipated letter grade in each course, with an NA for any course for which you are not enrolled
- anticipatedHours: your anticipated hours spent on each class per week based on on your experience so far, with an NA for any course for which you are not enrolled.

```
courseNum <- c(1,2,3,4,5,6)
coursename <- c("MATH", "BSDS", "BLAW", "IFM", "MIS", "BIO")
courseProf <- c("Chubb", "Popa", "Griffis", "Shahhosseini", "Rosidi", "Dickens")
enrolled <- c(T,T,T,T,T,F)
anticipatedGrade <- c("A", "E", "D", "B", "C", NA)
anticipatedHours <- c(1, 2, 3, 4, 5, NA)
```

2. (5 pts) Create and print a data frame called MyCourseDataFrame by combining all of the above vectors. Assign the names of each column to be the names of the original vectors. Summarize the type of each column. Do the data frame variables retain their original classes? Formally test this using appropriate R code.

```
MyCourseDataFrame <- data.frame(courseNum, coursename, courseProf, enrolled, anticipatedGrade, anticipa
MyCourseDataFrame
```

```
##   courseNum coursename   courseProf enrolled anticipatedGrade
## 1         1       MATH        Chubb     TRUE                A
## 2         2       BSDS         Popa     TRUE                E
## 3         3       BLAW      Griffis     TRUE                D
## 4         4        IFM Shahhosseini     TRUE                B
## 5         5        MIS       Rosidi     TRUE                C
## 6         6        BIO      Dickens    FALSE             <NA>
##   anticipatedHours
## 1                1
## 2                2
## 3                3
## 4                4
## 5                5
## 6               NA
```

```
class(MyCourseDataFrame[,1])
```

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

```
class(MyCourseDataFrame[,2])
```

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

```
class(MyCourseDataFrame[,3])
```

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

```
class(MyCourseDataFrame[,4])
```

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

```
class(MyCourseDataFrame[,5])
```

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

```
class(MyCourseDataFrame[,6])
```

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

3. (5 pts) Combine the vectors from (1) into a list called MyCourseDataList, where each vector is an element of the list. Assign the names of each element to be the names of the original vectors. Do the elements of the list maintain their original classes? Formally test this using appropriate R code.

The elements lose their original classes and turn into characters

```
MyCourseDataList <- cbind(courseNum, coursename, courseProf, enrolled, anticipatedGrade, anticipatedHou
MyCourseDataList
```

```
##      courseNum coursename courseProf     enrolled anticipatedGrade
## [1,] "1"       "MATH"     "Chubb"        "TRUE"   "A"
## [2,] "2"       "BSDS"     "Popa"         "TRUE"   "E"
## [3,] "3"       "BLAW"     "Griffis"      "TRUE"   "D"
## [4,] "4"       "IFM"      "Shahhosseini" "TRUE"   "B"
## [5,] "5"       "MIS"      "Rosidi"       "TRUE"   "C"
## [6,] "6"       "BIO"      "Dickens"      "FALSE"  NA
##      anticipatedHours
## [1,] "1"
## [2,] "2"
## [3,] "3"
## [4,] "4"
## [5,] "5"
## [6,] NA
```

```
col(MyCourseDataList)
```

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

```
class(MyCourseDataList[,1])
```

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

```
class(MyCourseDataList[,2])
```

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

```
class(MyCourseDataList[,3])
```

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

```
class(MyCourseDataList[,4])
```

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

```
class(MyCourseDataList[,5])
```

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

```
class(MyCourseDataList[,6])
```

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

4. Write code that returns the following values. As always, use 'echo = TRUE' so that your code as well as your output is displayed after each calculation:

- (2 pts) The values in courseNum, excluding the fourth value
- (2 pts) The total number of hours you anticipate spending on coursework per week
- (2 pts) A data frame with only the third row and first two columns of MyCourseDataFrame
- (2 pts) The first value in the second element of MyCourseDataList

```
MyCourseDataList[-4,1]
```

```
## [1] "1" "2" "3" "5" "6"
```

```
colSums(MyCourseDataFrame[6], na.rm=T, dims=1)
```

```
## anticipatedHours
##               15
```

```
MyCourseDataFrame[3, 1:2]
```

```
##   courseNum coursename
## 3         3       BLAW
```

```
MyCourseDataList[1,2]
```

```
## coursename
##     "MATH"
```

5. Convert the anticipatedGrade variable in MyCourseDataFrame into an ordered factor where your best anticipated grade is the maximum and your lowest anticipated grade is the minimum using the function factor(). Note: to get an ordering of values, you'll have to use the argument ordered = TRUE. Look at the documentation of factor() to understand how to do this. Now write code to answer the following questions, and output the answers.

(a) (2 pts) What is the maximum letter grade you anticipate receiving this semester?
(b) (2 pts) What is the minimum number of hours you expect to work per week in a class this semester? Is this in this course?
(c) (2 pts) For (a) and (b), what is the name and course number of each class? Your code should provide the result as a single textual output with both course number and course name separated by a colon, e.g. 'BSDS100: Intro to Data Science with R'

```
MyCourseDataFrame
```

```
##   courseNum coursename   courseProf enrolled anticipatedGrade
## 1         1       MATH        Chubb     TRUE                A
## 2         2       BSDS         Popa     TRUE                E
## 3         3       BLAW      Griffis     TRUE                D
## 4         4        IFM Shahhosseini     TRUE                B
## 5         5        MIS       Rosidi     TRUE                C
## 6         6        BIO      Dickens    FALSE             <NA>
##   anticipatedHours
## 1                1
## 2                2
## 3                3
## 4                4
## 5                5
## 6               NA
```

```
### (a)
f=factor(MyCourseDataFrame$anticipatedGrade)
levels(f) = rev(levels(f))
MyCourseDataFrame[order(MyCourseDataFrame$anticipatedGrade, as.character(f), decreasing = FALSE), ]
```

```
##   courseNum coursename   courseProf enrolled anticipatedGrade
## 1         1       MATH        Chubb     TRUE                A
## 4         4        IFM Shahhosseini     TRUE                B
## 5         5        MIS       Rosidi     TRUE                C
## 3         3       BLAW      Griffis     TRUE                D
## 2         2       BSDS         Popa     TRUE                E
## 6         6        BIO      Dickens    FALSE             <NA>
##   anticipatedHours
## 1                1
## 4                4
## 5                5
## 3                3
## 2                2
## 6               NA
```

```
MyCourseDataFrame[1,]
```

```
##   courseNum coursename courseProf enrolled anticipatedGrade
## 1         1       MATH      Chubb     TRUE                A
##   anticipatedHours
## 1                1
```

```
MyCourseDataFrame[1,5]
```

```
## [1] A
## Levels: A B C D E
```

```
paste(MyCourseDataFrame[1, 1], ":", MyCourseDataFrame[1,2])
```

```
## [1] "1 : MATH"
```

```
### (b)
f=factor(MyCourseDataFrame$anticipatedHours)
MyCourseDataFrame[order(MyCourseDataFrame$anticipatedHours, as.character(f), decreasing = FALSE), ]
```

```
##   courseNum coursename   courseProf enrolled anticipatedGrade
## 1         1       MATH        Chubb     TRUE                A
## 2         2       BSDS         Popa     TRUE                E
## 3         3       BLAW      Griffis     TRUE                D
## 4         4        IFM Shahhosseini     TRUE                B
## 5         5        MIS       Rosidi     TRUE                C
## 6         6        BIO      Dickens    FALSE             <NA>
##   anticipatedHours
## 1                1
## 2                 2
## 3                 3
## 4                 4
## 5                 5
## 6                NA
```

```
MyCourseDataFrame[1,]
```

```
##   courseNum coursename courseProf enrolled anticipatedGrade
## 1         1       MATH      Chubb     TRUE                A
##   anticipatedHours
## 1                1
```

```
MyCourseDataFrame[1,6]
```

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

```
paste(MyCourseDataFrame[1,1], ":", MyCourseDataFrame[1,2])
```

```
## [1] "1 : MATH"
```

```
### I'm not sure what you wanted here. I had difficulty doing this without hardcoding the coordinates.
```