Homework 2

Hyun Suk Ryoo (hr2ee)

Problem 1

A

```
course <- list(Name="Gretchen Martinet",
               Department= "Statistics",
               Courses=c(2559,3080),
               ActiveTeach2559= FALSE,
               ActiveTeach3080=TRUE,
               Enr2559=10,
               Enr3080=c(90,90,88),
               Days2559=c("Tuesday", "Thursday"),
               Days3080=matrix(c("Monday", "Wednesday", "Tuesday",
                                 "Thursday", "Tuesday", "Thursday"),
                               nrow=3, ncol=2, byrow=TRUE))
print(course)
```

```
## $Name
## [1] "Gretchen Martinet"
##
## $Department
## [1] "Statistics"
##
## $Courses
## [1] 2559 3080
##
## $ActiveTeach2559
## [1] FALSE
##
## $ActiveTeach3080
## [1] TRUE
##
## $Enr2559
## [1] 10
##
## $Enr3080
## [1] 90 90 88
##
## $Days2559
## [1] "Tuesday"  "Thursday"
##
## $Days3080
##      [,1]      [,2]
## [1,] "Monday"  "Wednesday"
## [2,] "Tuesday" "Thursday"
## [3,] "Tuesday" "Thursday"
```

Made a list containing different dataframes

B

```
cap_3080 <- course$Enr3080
filled_3080 <- which(cap_3080 == 90)
print(filled_3080)
```

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

Sections 1 and 2 are filled at 90 students. This was done by going through the enrollment vector of 3080 and equating it to 90.

C

```
sum_reports <- sum(sum(course$Enr2559[course$ActiveTeach2559 == TRUE])
                   +
                     sum(course$Enr3080[course$ActiveTeach3080 == TRUE]))
print(sum_reports)
```

```
## [1] 268
```

I added all the numbers in the vector Enr3080 in the list of courses.

Problem 2

A

```
pnames <- c("Mercury", "Venus", "Earth", "Mars",
            "Jupiter", "Saturn", "Uranus", "Neptune")
pdistance <-c(0.39, 0.72, 1, 1.52,
              5.2, 9.54, 19.18, 30.06)
ptype <- c("terrestrial", "terrestrial", "terrestrial", "terrestrial",
           "gas", "gas", "gas", "gas")
pdiameter <-c(0.382, 0.949, 1, 0.532,
              11.209, 9.449, 4.007, 3.883)
protation <- c(58.64, -243.02, 1, 1.03,
               0.41, 0.43, -0.72,0.67)
prings <-c(FALSE, FALSE, FALSE, FALSE,
           TRUE, TRUE, TRUE, TRUE)
pmoons <-c("none", "none", "one", "more than one",
           "more than one", "more than one", "more than one", "more than one")
planet_data <- data.frame(Name=pnames, distance=pdistance,
                          type=ptype, diameter=pdiameter,
                          rotation=protation, rings=prings,
                          moons=pmoons)
print(planet_data)
```

```
##      Name distance        type diameter rotation rings         moons
## 1 Mercury     0.39 terrestrial    0.382    58.64 FALSE          none
## 2   Venus     0.72 terrestrial    0.949  -243.02 FALSE          none
## 3   Earth     1.00 terrestrial    1.000     1.00 FALSE           one
## 4    Mars     1.52 terrestrial    0.532     1.03 FALSE more than one
## 5 Jupiter     5.20         gas   11.209     0.41  TRUE more than one
## 6  Saturn     9.54         gas    9.449     0.43  TRUE more than one
## 7  Uranus    19.18         gas    4.007    -0.72  TRUE more than one
## 8 Neptune    30.06         gas    3.883     0.67  TRUE more than one
```

I made several vectors and made it into a dataframe.

B

```
atleast2Au <- planet_data[which(planet_data$distance > 2), ]
print(atleast2Au)
```

```
##      Name distance type diameter rotation rings         moons
## 5 Jupiter     5.20  gas   11.209     0.41  TRUE more than one
## 6  Saturn     9.54  gas    9.449     0.43  TRUE more than one
## 7  Uranus    19.18  gas    4.007    -0.72  TRUE more than one
## 8 Neptune    30.06  gas    3.883     0.67  TRUE more than one
```

I took which instances of the subset distance has a value greater than 2. Then I found the corresponding items in the original dataset that satisfies this condition. I then put it in a datafram called atlesat2Au

C

```
opposite_rotation <- which(planet_data$rotation <0)
print(opposite_rotation)
```

```
## [1] 2 7
```

The two planets that are rotating the opposite way are Venus (2) and Uranus (7) with a distance of 0.72, 19.18 respectivly.

D

```
diameter_bigger <- planet_data[which(planet_data$diameter > 1), ]
print(diameter_bigger[,c(1,3)])
```

```
##      Name type
## 5 Jupiter  gas
## 6  Saturn  gas
## 7  Uranus  gas
## 8 Neptune  gas
```

I made data frame where the contents conditions were that they have a diameter greater than earth's, which is 1. I then proceeded to take the first and third columns to represent only the name and type.

E

```
morethanone_moon <- planet_data[which(planet_data$moons == "more than one"), ]
print(morethanone_moon[,c(2,3)])
```

```
##   distance        type
## 4     1.52 terrestrial
## 5     5.20         gas
## 6     9.54         gas
## 7    19.18         gas
## 8    30.06         gas
```

I made a data frame where the contents conditions were that they have more than one moon. I then proceeded to take the second and third columns to isolate the distance and type of the original data frame.

Problem 3

A

```
USArrests2 <- USArrests[-3]
print(USArrests2[1:10,])
```

```
##             Murder Assault Rape
## Alabama       13.2     236 21.2
## Alaska        10.0     263 44.5
## Arizona        8.1     294 31.0
## Arkansas       8.8      190 19.5
## California     9.0      276 40.6
## Colorado       7.9      204 38.7
## Connecticut    3.3      110 11.1
## Delaware       5.9      238 15.8
## Florida       15.4      335 31.9
## Georgia       17.4      211 25.8
```

B

```
arrests_tot <- c(sum(USArrests2$Murder),
                 sum(USArrests2$Assault),
                 sum(USArrests2$Rape))
print(arrests_tot)
```

```
## [1]  389.4 8538.0 1061.6
```

I took the sum of every crime and concated into a vector called arrests_tot

C

```
PropRape <- USArrests2$Rape / USArrests2$Assault
USArrests2$PropRape <- PropRape
print(USArrests2[1:10,])
```

```
##             Murder Assault Rape   PropRape
## Alabama       13.2      236 21.2 0.08983051
## Alaska        10.0      263 44.5 0.16920152
## Arizona        8.1      294 31.0 0.10544218
## Arkansas       8.8      190 19.5 0.10263158
## California     9.0      276 40.6 0.14710145
## Colorado       7.9      204 38.7 0.18970588
## Connecticut    3.3      110 11.1 0.10090909
## Delaware       5.9      238 15.8 0.06638655
## Florida       15.4      335 31.9 0.09522388
## Georgia       17.4      211 25.8 0.12227488
```

I first took the proportions then I put them in as a vector. I then proceeded to make a dataframe.

D

```
above20 <- USArrests2[which(PropRape > 0.2), ]
states_20 <- row.names(above20[0])
print(states_20)
```

```
## [1] "Hawaii"    "Iowa"      "Minnesota" "Vermont"   "Wisconsin"
```

You can see from the dataframe above that the states that have a proportion higher than 20% ARE Hawaii, Iowa, Minnesota, Vermont, and Wisconsin. I used the row.names to get the names of the rows (states). I first took which proportions have higher than 20, then I took those indicies and set them as a parameter to find the names and details in the USArrests2 dataframe.

E

```
sorted_murder <- USArrests2[order(USArrests2$Murder, decreasing = TRUE),]
print(sorted_murder)
```

```
##                Murder Assault Rape   PropRape
## Georgia          17.4     211 25.8 0.12227488
## Mississippi      16.1     259 17.1 0.06602317
## Florida          15.4     335 31.9 0.09522388
## Louisiana        15.4     249 22.2 0.08915663
## South Carolina   14.4     279 22.5 0.08064516
## Alabama          13.2     236 21.2 0.08983051
## Tennessee        13.2     188 26.9 0.14308511
## North Carolina   13.0     337 16.1 0.04777448
## Texas            12.7     201 25.5 0.12686567
## Nevada           12.2     252 46.0 0.18253968
## Michigan         12.1     255 35.1 0.13764706
## New Mexico       11.4     285 32.1 0.11263158
## Maryland         11.3     300 27.8 0.09266667
## New York         11.1     254 26.1 0.10275591
## Illinois         10.4     249 24.0 0.09638554
## Alaska           10.0     263 44.5 0.16920152
## Kentucky          9.7     109 16.3 0.14954128
## California        9.0     276 40.6 0.14710145
## Missouri          9.0     178 28.2 0.15842697
## Arkansas          8.8     190 19.5 0.10263158
## Virginia          8.5     156 20.7 0.13269231
## Arizona           8.1     294 31.0 0.10544218
## Colorado          7.9     204 38.7 0.18970588
## New Jersey        7.4     159 18.8 0.11823899
## Ohio              7.3     120 21.4 0.17833333
## Indiana           7.2     113 21.0 0.18584071
## Wyoming           6.8     161 15.6 0.09689441
## Oklahoma          6.6     151 20.0 0.13245033
## Pennsylvania      6.3     106 14.9 0.14056604
## Kansas            6.0     115 18.0 0.15652174
## Montana           6.0     109 16.4 0.15045872
## Delaware          5.9     238 15.8 0.06638655
## West Virginia     5.7      81  9.3 0.11481481
## Hawaii            5.3      46 20.2 0.43913043
## Oregon            4.9     159 29.3 0.18427673
## Massachusetts     4.4     149 16.3 0.10939597
## Nebraska          4.3     102 16.5 0.16176471
## Washington        4.0     145 26.2 0.18068966
## South Dakota      3.8      86 12.8 0.14883721
## Rhode Island      3.4     174  8.3 0.04770115
## Connecticut       3.3     110 11.1 0.10090909
## Utah              3.2     120 22.9 0.19083333
## Minnesota         2.7      72 14.9 0.20694444
## Idaho             2.6     120 14.2 0.11833333
## Wisconsin         2.6      53 10.8 0.20377358
## Iowa              2.2      56 11.3 0.20178571
## Vermont           2.2      48 11.2 0.23333333
## Maine             2.1      83  7.8 0.09397590
## New Hampshire     2.1      57  9.5 0.16666667
## North Dakota      0.8      45  7.3 0.16222222
```

I used the order function with the parameter decreasing=TRUE to sort the dataframe by descending murder counts.

F

```
number_states <- sum(USArrests2$Murder > USArrests2$Rape)
print(number_states)
```

```
## [1] 0
```

I compared the muder and rape cases to which there were no states.

References

1. <https://stackoverflow.com/questions/7793295/ how-to-order-a-data-frame-by-one-descending-and-one-ascending-column>
2. <https://developmentality.wordpress.com/2010/02/12/ r-sorting-a-data-frame-by-the-contents-of-a-column/>
3. https://stackoverflow.com/questions/1813550/count-of-entries-in-data-frame-in-r