HW2_IS457_85

Mon Sep 24, 2018

Part 1. Warm up

In this part, we will work with vectors and apply some functions on them.

(1) Create a Vector like this (0 0 0 3 3 3 6 6 6 9 9 9 12 12 12 15 15 15 18 18 18) with functions seq() and rep() and call it "vec" (1 pt)

Ans :

```
vec = rep(seq(0,18,3), each = 3)
vec
```

```
[1]  0  0  0  3  3  3  6  6  6  9  9  9 12 12 12 15 15 15 18 18 18
```

The above code gives us the desired sequence of elements.

(2) Calculate the fraction of elements in vec that are more than or equal to 9. (2 pts) hint: R can do vectorized operations.

Ans :

```
frac = sum(vec>=9)/length(vec)
frac
```

```
[1] 0.5714286
```

The fraction of elements in vec that are more than or equal to 9 is 0.57.

(3) Create a Vector like this (1 2 2 3 3 3 4 4 4 4 5 5 5 5 5) with functions rep() and the : operator (1 pt)

Ans :

```
vec2 = rep(1:5, c(1, 2, 3, 4, 5))
vec2
```

```
[1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
```

As can be seen from the output above the code gives us the desired sequence.

Part 2. CO2 Data

(4) Use R to generate descriptions of the CO2 data which is already available with the base R installation (it is called CO2 in R. Please note that we are using the CO2 dataset and not the similarly named co2 dataset). Print out the summary of each column and the dimensions of the dataset. (2 pts.) (hint: you may find the summary() and dim() useful). Write up your descriptive findings and observations of the R output. (1 pt.)

Ans :

summary(CO2)

```
Plant             Type         Treatment       conc          uptake
 Qn1    : 7   Quebec     :42   nonchilled:42   Min.   :  95   Min.   : 7.70
 Qn2    : 7   Mississippi:42   chilled   :42   1st Qu.: 175   1st Qu.:17.90
 Qn3    : 7                                    Median : 350   Median :28.30
 Qc1    : 7                                    Mean   : 435   Mean   :27.21
 Qc3    : 7                                    3rd Qu.: 675   3rd Qu.:37.12
 Qc2    : 7                                    Max.   :1000   Max.   :45.50
 (Other):42
```

dim(CO2)

```
[1] 84  5
```

The CO2 dataset consists of 84 observations for 5 variables. The variables are: 1) Plant – factor variable for giving unique id for each plant. 2) Type – factor variable for location of the plant. 3) Treatment – factor variable for the treatment given to the plant. 4) conc – numeric variable for concentration of CO2. 5) uptake – numeric variable the uptake rate in the plant observed.

(5) Show last 8 plants' uptake values (1 pt.)

Ans :

tail(CO2, n = 8)

```
Plant        Type Treatment conc uptake
77   Mc2 Mississippi   chilled 1000   14.4
78   Mc3 Mississippi   chilled   95   10.6
79   Mc3 Mississippi   chilled  175   18.0
80   Mc3 Mississippi   chilled  250   17.9
81   Mc3 Mississippi   chilled  350   17.9
82   Mc3 Mississippi   chilled  500   17.9
83   Mc3 Mississippi   chilled  675   18.9
84   Mc3 Mississippi   chilled 1000   19.9
```

To show the last eight uptake values of the plants we provide the value 8 to the n argument of the **tail()** function to override its default value of 6.

(6) Show all plants' uptake values except the first 20 plants'. (1 pt.)

Ans :

CO2$uptake[-seq(1:20)]

```
[1] 45.5 14.2 24.1 30.3 34.6 32.5 35.4 38.7  9.3 27.3 35.0 38.8 38.6 37.5 42.4
[16] 15.1 21.0 38.1 34.0 38.9 39.6 41.4 10.6 19.2 26.2 30.0 30.9 32.4 35.5 12.0
[31] 22.0 30.6 31.8 32.4 31.1 31.5 11.3 19.4 25.8 27.9 28.5 28.1 27.8 10.5 14.9
[46] 18.1 18.9 19.5 22.2 21.9  7.7 11.4 12.3 13.0 12.5 13.7 14.4 10.6 18.0 17.9
[61] 17.9 17.9 18.9 19.9
```

We use subsetting by exclusion method to remove the first 20 plants' values.

(7) Calculate the mean of uptake subseted by the "Treatment" variable.(1 pt) hint: apply function family

Ans :

mean_uptake = tapply(CO2$uptake, CO2$Treatment, mean)
mean_uptake

```
nonchilled    chilled
  30.64286   23.78333
```

The mean uptake value for nonchilled plants are 30.64 and for chilled plants is 23.78.

(8) Create a logical vector uptake treatment . (2 pts)
For the plants with Chilled treatment (Treatment == "chilled"), return value TRUE when uptake > 30.
For the plants with Non-Chilled treatment (Treatment == "nonchilled"), return value TRUE when uptake > 40.

Ans :

uptake_treatment = as.logical((CO2$Treatment == "nonchilled" & CO2$uptake > 40) | (CO2$Treatment == "chilled" & CO2$uptake > 30))
uptake_treatment

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
[14]  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
[27]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
[40]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[53] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[66] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[79] FALSE FALSE FALSE FALSE FALSE FALSE
```

The above logical vector shows the truth values corresponding to the **CO2$uptake** value greater than 30 in case of **CO2$Treatment** value being **"chilled"** and **CO2$uptake** value greater than 40 in case of **CO2$Treatment** value being **"nonchilled"**.

(9) Here is an alternative way to create the same vector in Q8. First, we create a numeric vector uptake_test that is 30 for each plant with chilled treatment and 40 for each plant with non-chilled treatment. To do this, first create a vector of length 2 called test_val whose first element is 40 and second element is 30. (1 pt).

Ans :

test_val = c(40,30)

Create the uptake_test vector by subsetting test_val by position, where the positions could be represented based on the Treatment column in CO2. (1 pt)

uptake_test = test_val[CO2$Treatment]

Finally, use uptake_test and the uptake column to create the desired vector, and call it uptake_treatment2. (1 pt)

uptake_treatment2 = CO2$uptake>uptake_test
uptake_treatment2

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
[14]  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
[27]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
[40]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[53] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
[66] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[79] FALSE FALSE FALSE FALSE FALSE FALSE
```

all.equal(uptake_treatment, uptake_treatment2)

```
[1] TRUE
```

As can be seen from the above code, **uptake_treatment** and **uptake_treatment2** vector have the same element values.

Part 3. San Francisco Housing Data (25 pts.)

Load the data into R.

load(url("https://www.stanford.edu/~vcs/StatData/SFHousing.rda"))

(10) (3 pts.) What objects are in SFHousing.rda? Give the name and class of each.

Ans :

names(housing)

```
[1] "county"  "city"    "zip"     "street"  "price"
[6] "br"      "lsqft"   "bsqft"   "year"    "date"
[11] "long"    "lat"     "quality" "match"   "wk"
```

str(housing)

```
'data.frame':   281506 obs. of  15 variables:
 $ county : Factor w/ 9 levels "Alameda County",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ city   : Factor w/ 163 levels "Alameda","Alamo",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ zip    : Factor w/ 314 levels "94002","94005",..: 79 79 79 79 79 80 79 79 80 79
...
 $ street : chr  "1220 Broadway" "429 Fair Haven Road" "2804 Fernside Boulevard" "1
316 Grove Street" ...
 $ price  : num  509000 504000 526000 637000 393500 ...
 $ br     : int  4 4 2 3 3 4 2 3 3 3 ...
 $ lsqft  : num  4420 6300 4000 2700 1780 ...
 $ bsqft  : int  1834 1411 1272 1168 1610 1720 854 1476 1503 1240 ...
 $ year   : int  1910 1964 1941 1910 NA 1965 1964 1990 1986 1963 ...
 $ date   : POSIXt, format: "2003-04-27 02:00:00" ...
 $ long   : num  -122 -122 -122 -122 -122 ...
 $ lat    : num  37.8 37.8 37.8 37.8 37.8 ...
 $ quality: Factor w/ 13 levels "1","2","3","4",..: 12 6 6 6 6 11 NA 6 9 12 ...
 $ match  : Factor w/ 16 levels "1","2","3","4",..: 8 8 8 8 9 8 NA 9 10 8 ...
 $ wk     : Date, format: "2003-04-21" ...
```

The housing dataset consists of the following objects along with their class.

Object Name	Object class
1. county	Factor
2. city	Factor
3. zip	Factor
4. street	Character
5. price	Numeric

6. br Integer
7. lsqft Numeric
8. bsqft Integer
9. year Integer
10. date POSIXt
11. long Numeric
12. lat Numeric
13. quality Factor
14. match Factor
15. wk Date

(11) Give a summary of each object, including a summary of each variable and the dimension of the object. (4 pts)

Ans :

dim(housing)

```
[1] 281506     15
```

The above output shows the dimensions of the dataset. The summary for each variable is below.

summary(housing$county)

```
Alameda County  Contra Costa County         Marin County 
           60410                59381                10450 
     Napa County San Francisco County     San Mateo County 
            5066                 8137                22558 
  Santa Clara County        Solano County        Sonoma County 
           70424                23404                21676 
```

summary(housing$city)

```
Oakland                  Santa Rosa
                  14730                         9917 
                Fremont                San Francisco 
                   9414                         8137 
              Evergreen                      Antioch 
                   7947                         7726 
                Vallejo                      Concord 
                   7183                         7109 
                Hayward                    Fairfield 
                   6565                         5734 
              Vacaville                     Richmond 
                   5439                         5298 
```

summary(housing$zip)

```
94565   94509   95123   95687   94533   94531   94591   94536   94513 
  4595    4302    4023    3652    3472    3427    3369    3292    3235 
  94587   94583   94521   94558   95035   95125   94806   95127   94553 
   2896    2784    2779    2757    2676    2646    2617    2607    2549 
  94544   95111   94551   95020   95124   94550   94538   94534   94568 
   2524    2494    2467    2446    2390    2376    2279    2274    2240 
  95037   94561   94541   95051   95014   94539   94928   94605   95122 
   2237    2193    2186    2169    2126    2124    2090    2084    2063 
  94560   95403   94526   94590   95136   94523   94804   95008   94585 
   2014    2012    1958    1957    1957    1940    1931    1914    1911 
```

```
  94566   94577   95148   94589   95121   94546   95404   94087   95688
   1892    1871    1856    1853    1851    1838    1797    1787    1787
  95132   94520   95120   94588   95401   95118   95409   94555   94954
   1778    1753    1746    1740    1734    1725    1723    1719    1711
  94080   94611   94015   94547   94501   94518   95116   94603   94510
   1692    1672    1648    1634    1619    1590    1590    1552    1534
  95129   95476   94941   95131   94506   95407   94010   94403   94621
   1532    1521    1488    1470    1454    1451    1426    1417    1414
  94044   94514   94404   94066   94801   94597   94070   94947   95070
   1377    1363    1355    1350    1348    1344    1343    1338    1335
  94602   95492   94598   94803   94578   95135   94901   95032 (Other)
   1330    1324    1312    1310    1291    1289    1288    1285   83020
   NA's
      5
```

summary(housing$street)

```
Length     Class      Mode
   281506 character
```

summary(housing$price)

```
Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
   22000   400000   530000   602000   700000 20000000
```

summary(housing$br)

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   3.024   4.000   8.000
```

summary(housing$lsqft)

```
Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
       19      4000      5760     65939      7701 418611600     21687
```

summary(housing$bsqft)

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
    122    1121    1430    1624    1882 1868120     426
```

summary(housing$year)

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
      0    1954    1971    1966    1985    3894    9202
```

summary(housing$date)

```
Min.                 1st Qu.                Median
"2003-04-27 02:00:00" "2004-02-08 02:00:00" "2004-10-24 02:00:00"
                 Mean               3rd Qu.                  Max.
"2004-11-01 18:06:12" "2005-07-24 02:00:00" "2006-06-04 02:00:00"
```

summary(housing$long)

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 -123.6  -122.3  -122.1  -122.1  -121.9  -121.5   23316
```

summary(housing$lat)

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  36.98   37.50   37.77   37.78   38.00   38.85   23316 
```

summary(housing$quality)

```
                                        1 
                                     2593 
                                        2 
                                      605 
                                        3 
                                      199 
                                        4 
                                       69 
                                        5 
                                        3 
      QUALITY_ADDRESS_RANGE_INTERPOLATION 
                                   170719 
                    QUALITY_CITY_CENTROID 
                                    20473 
      QUALITY_COUNTY_SUBDIVISION_CENTROID 
                                      160 
            QUALITY_EXACT_PARCEL_CENTROID 
                                    17208 
        QUALITY_UNIFORM_LOT_INTERPOLATION 
                                       96 
QUALITY_ZIP_CODE_TABULATION_AREA_CENTROID 
                                    14980 
                            gpsvisualizer 
                                    31084 
                             gpsvisualzer 
                                        1 
                                     NA's 
                                    23316 
```

summary(housing$match)

```
               1                2                3                4 
            2244              823              319               80 
               5          Address        CityLevel            Exact 
               3              287               36           197044 
         Relaxed; Soundex  Relaxed;Soundex          Soundex 
           30570            23338              379             2573 
     StreetLevel        Substring      ZipCodeLevel        cityLevel 
               0              491                2                1 
            NA's 
           23316 
```

summary(housing$wk)

```
Min.      1st Qu.       Median         Mean      3rd Qu. 
"2003-04-21" "2004-02-01" "2004-10-18" "2004-10-26" "2005-07-18" 
        Max. 
"2006-05-29" 
```

(12) After exploring the data (maybe using the summary() function), describe in words the connection between the two objects (e.g., what links them together). (2 pts)

Ans :

The county and zip objects are related in the sense that the county name can be used to determine the range of zip values for that county. These variables are also linked to the city object as the range of zip values and the county can also be decided by the city name.

(13) Describe in words two problems that you see with the data. (2 pts)

Ans :

One issue I can observe in the dataset is the use of different date formats for the **date** and **wk** variables. Using one date format can maintain data consistency in the dataset. The other issue I can find is the existence of garbage values in the **year** variable viz. 0, 1, 2, 3894, 3885, 3881 and so on. These values are either too old or in the future to hold any meaning. We could also have used a **factor** datatype instead of **integer** for the **year** variable.

(14) (2 pts.) We will work with the houses in San Francisco, Fremont, Vallejo, Concord and Livermore only. Subset the housing data frame so that we have only houses in these cities and keep only the variables county, city, zip, price, br, bsqft, and year. Call this new data frame SelectArea. This data frame should have 36686 observations and 7 variables. (Note you may need to reformat any factor variables so that they do not contain incorrect levels)

Ans :

```
cities = c("San Francisco", "Fremont", "Vallejo", "Concord", "Livermore")
col = c("county", "city", "zip", "price", "br", "bsqft", "year")
SelectArea = housing[is.element(housing$city,cities), col]
SelectArea$city = factor(SelectArea$city, levels = cities)
str(SelectArea)
```

```
'data.frame':	36686 obs. of  7 variables:
 $ county: Factor w/ 9 levels "Alameda County",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ city  : Factor w/ 5 levels "San Francisco",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ zip   : Factor w/ 314 levels "94002","94005",..: 105 107 105 105 122 108 107 105
107 107 ...
 $ price : num  538000 455000 422000 459000 438000 ...
 $ br    : int  4 3 3 3 3 1 3 2 3 3 ...
 $ bsqft : int  1871 1401 1390 1645 1688 675 1290 1254 1637 1304 ...
 $ year  : int  1977 1963 1971 1965 1986 1987 1968 1988 NA 1962 ...
```

We subset the **housing** dataset using the **cities** vector to only include the cities we want in our **SelectArea** dataset and then use the **col** vector to include only the columns in the **housing** dataset present in the **col** vector. Finally, we refactor the **city** variable in the **SelectArea** dataset to only have five levels from the **cities** vector.

(15) (3 pts.) We are interested in making plots of price and size of house, but before we do this we will further subset the housing dataframe to remove the unusually large values. Use the quantile function to determine the 95th percentile of price and bsqft and eliminate all of those houses that are above either of these 95th percentiles. Call this new data frame SelectArea (replacing the old one) as well. It should have 33693 observations.

Ans :

Code to calculate the value of 95th percentile of price.
q_value_bsqft = quantile(SelectArea$bsqft, probs = c(0.95), na.rm = TRUE)

Code to calculate the value of 95th percentile of bsqft
q_value_price = quantile(SelectArea$price, probs = c(0.95), na.rm = TRUE)

Code to select only those rows fulfilling all the conditions

SelectArea = SelectArea[SelectArea$price < q_value_price & SelectArea$bsqft < q_value_bsqft & !is.na(SelectArea$bsqft) & !is.na(SelectArea$price),]
Str(SelectArea)

```
'data.frame':	33693 obs. of  7 variables:
 $ county: Factor w/ 9 levels "Alameda County",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ city  : Factor w/ 5 levels "San Francisco",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ zip   : Factor w/ 314 levels "94002","94005",..: 105 107 105 105 122 108 107 105 
107 107 ...
 $ price : num  538000 455000 422000 459000 438000 ...
 $ br    : int  4 3 3 3 3 1 3 2 3 3 ...
 $ bsqft : int  1871 1401 1390 1645 1688 675 1290 1254 1637 1304 ...
 $ year  : int  1977 1963 1971 1965 1986 1987 1968 1988 NA 1962 ...
```

We use the **q_value_bsqft** and **q_value_price** variables to store the value of the 95th percentile of **price** and **bsqft** variables of the **SelectArea** dataset respectively. Finally, we subset the dataset **SelectArea** by using these variables in our conditional expression.

(16) (2 pts.) Create a new vector that is called price_per_sqft by dividing the sale price by the square footage. Add this new variable to the data frame.

Ans :

SelectArea$price_per_sqft = SelectArea$price/SelectArea$bsqft
head(SelectArea$price_per_sqft)

```
[1] 287.5468 324.7680 303.5971 279.0274 259.4787 320.0000
```

As can be seen from the output above the new variable **price_per_sqft** is added to the dataset **SelectArea**.

(17) (2 pts.) Create a vector called br_new, that is the number of bedrooms in the house, except when the number is greater than 5, set it (br_new) to 5.
Ans :

br_new = SelectArea$br
br_new[br_new>5] = 5
head(br_new)

```
[1] 4 3 3 3 3 1
```

(18) (4 pts. 2 + 2 - see below)
Use the heat.colors function to create a vector of 5 colors, call this vector rCols. When you call this function, set the alpha argument to 0.25.
Create a vector called brCols where each element's value corresponds to the color in rCols indexed by the number of bedrooms in the br_new. For example, if the element in br_new is 3 then the color will be the third color in rCols. (2 pts.)

Ans :

rCols = heat.colors(5, alpha = 0.25)
rCols

```
[1] "#FF000040" "#FF550040" "#FFAA0040" "#FFFF0040" "#FFFF8040"
```

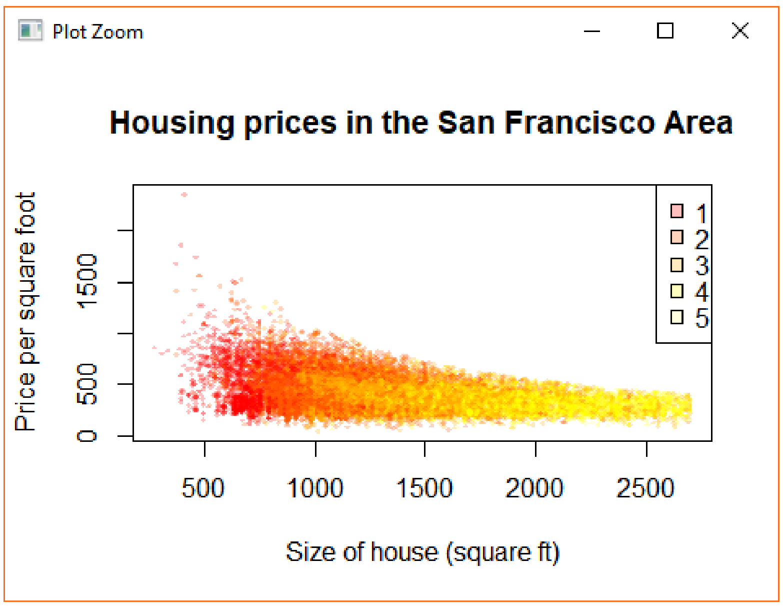
brCols = rCols[br_new]
head(brCols)

```
[1] "#FFFF0040" "#FFAA0040" "#FFAA0040" "#FFAA0040" "#FFAA0040" "#FF000040"
```

We are now ready to make a plot!

```
plot(price_per_sqft ~ bsqft, data = SelectArea,
    main = "Housing prices in the San Francisco Area",
    xlab = "Size of house (square ft)",
    ylab = "Price per square foot",
    col = brCols, pch = 18, cex = 0.5)
legend(legend = 1:5, fill = rCols, "topright")
```

What's your interpretation of the plot? e.g., the trend? the cluster? the comparison? (1 pt.)



From the above plot it can be observed that as the size of the house (square ft) increases we see a decrease in the price per square foot of the house. Also, the houses with more bedrooms tend to be cheaper than the less bedrooms having the same square footage of the house. We can also observe that most of the houses have 1, 2 bedrooms are clustered below the 1250 square feet size range which makes sense as bigger houses tend to have more bedrooms.