Do not remove any of the comments. These are marked by

HW 2 - Due Monday, Feb 19 2018 in moodle and hardcopy in class.

(1). Please upload R code and report to Moodle

with filename: HW2_IS457_YourCourseID.

(2). Turn in hard copy of your report in class.

Class ID:

In this assignment you will practice how to manipulate vector and dataframe,

such as taking subsets and creating new data structure, and end with creating a fantastic plot.

You will work with the mtcars data in R library and a dataset called SFHousing.

Before beginning with the housing data however, you will do some warm up exercises.

PART 1. Warm up (3 pts)

Q1. Create a Vector like this (0 0 2 2 4 4 6 6 8 8 10 10 12 12)

with functions seq() and rep() and call it "vec" (1 pt)

Your code below

```
x = seq(0,12,by = 2)
       vec = rep(x, each =2 )
       vec
```

```
##  [1]  0  0  2  2  4  4  6  6  8  8 10 10 12 12
```

Q2. Calculate the fraction of elements in vec that are more than 4. (2 pts)

hint: R can do vectorized operations.

Your code below

```r
sum(vec > 4)
```

```
## [1] 8
```

```r
sum(vec > 4)/length(vec)
```

```
## [1] 0.5714286
```

PART II. mtcars Data (9 pts)

Q3. Use R to generate descriptions of the mtcars data which is already built in R base.

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.)

Your code below:

```r
data("mtcars")
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
```

```
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
      dim(mtcars)
```

```
## [1] 32 11
```

Your answer below:

```
 # There are 32 observations and 11 columns
```

Q4. Show last 10 cars' mpg values (1 pt.)

Your code below:

```
   mtcars[23:32,c("mpg")]
```

```
##  [1] 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7 15.0 21.4
```

Q5. Show all cars' mpg values except the first 10 cars'. (1 pt.)

Your code below:

```
 class(mtcars)
```

```
## [1] "data.frame"
```

```
      mtcars[-(1:10),c("mpg")]
```

```
##  [1] 17.8 16.4 17.3 15.2 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3
## [15] 19.2 27.3 26.0 30.4 15.8 19.7 15.0 21.4
```

Q6. Calculate the mean of mpg subseted by "vs" variable.(1 pt)

hint: apply function family.

Your code below:

```
  tapply(mtcars$mpg, mtcars$vs, mean)
```

```
##        0        1
## 16.61667 24.55714
```

Q7. Create a logical vector mpg_vs . (2 pts)

For the cars with V-engine (vs = 0), return value TRUE when mpg > 14.

For the cars with straight engine (vs = 1), return value TRUE when mpg > 20.

Your code below:

```
mpg_vs = c(TRUE,FALSE)

(mtcars$vs==0) & (mtcars$mpg>14)
```

```
##  [1]  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## [12]  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE
## [23]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
```

```
(mtcars$vs==1) & (mtcars$mpg>20)
```

```
##  [1] FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE
## [12] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE
## [23] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE
```

Q8. Here is an alternative way to create the same vector in Q2.

First, we create a numeric vector mpg_index that is 14 for each car with V-engine

and 20 for each car with straight engine. To do this, first create a vector of length 2 called

id_val whose first element is 14 and second element is 20. (1 pt)

Your code below:

```
  id_val = c(14,20)
                 id_val
```

```
## [1] 14 20
```

Create the mpg_index vector by subsetting id_val by position, where the

positions could be represented based on vs column in mtcars. (1 pt)

Your code below

```
mpg_index= c(mtcars$vs)
mpg_index[(mpg_index==0)]= id_val[1]
mpg_index[(mpg_index==1)]= id_val[2]
mpg_index
```

```
##  [1] 14 14 20 20 14 20 14 20 20 20 20 14 14 14 14 14 14 20 20 20 20 14 14
## [24] 14 14 20 14 20 14 14 14 20
```

Finally, use mpg_index and mpg column to create the desired vector, and

call it mpg_vs2. (1 pt)

Your code below

```
mpg_vs2 = data.frame(mpg_index,mtcars$mpg)
    mpg_vs2
```

```
##    mpg_index mtcars.mpg
## 1         14       21.0
## 2         14       21.0
## 3         20       22.8
## 4         20       21.4
## 5         14       18.7
## 6         20       18.1
## 7         14       14.3
## 8         20       24.4
## 9         20       22.8
## 10        20       19.2
## 11        20       17.8
## 12        14       16.4
## 13        14       17.3
## 14        14       15.2
## 15        14       10.4
## 16        14       10.4
## 17        14       14.7
## 18        20       32.4
## 19        20       30.4
## 20        20       33.9
## 21        20       21.5
```

```
## 22          14        15.5
## 23          14        15.2
## 24          14        13.3
## 25          14        19.2
## 26          20        27.3
## 27          14        26.0
## 28          20        30.4
## 29          14        15.8
## 30          14        19.7
## 31          14        15.0
## 32          20        21.4
```

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

Load the data into R.

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

Q9. (3 pts.)

What objects are in SFHousing.rda? Give the name and class of each.

Your code below

```
       objects()
```

```
## [1] "cities"    "housing"   "id_val"    "mpg_index" "mpg_vs"    "mpg_vs2"
## [7] "mtcars"    "vec"       "x"
```

```
       class(cities)
```

```
## [1] "data.frame"
```

```
       class(housing)
```

```
## [1] "data.frame"
```

Your answer here

```
#the objects in SFHousing.rda are "cities" and "housing"
#the class of cities is a dataframe
# the class of housing is a dataframe
```

Q10. give a summary of each object, including a summary of each variable and the dimension of the object. (4 pts)

Your code below

```
summary(cities)
```

```
##    longitude         latitude                        county
##  Min.   :-123.5   Min.   :37.01   Santa Clara County :30
##  1st Qu.:-122.5   1st Qu.:37.54   Contra Costa County:29
##  Median :-122.3   Median :37.89   Marin County       :24
##  Mean   :-122.3   Mean   :37.87   San Mateo County   :24
##  3rd Qu.:-122.0   3rd Qu.:38.09   Sonoma County      :23
##  Max.   :-121.6   Max.   :38.80   Alameda County     :17
##  NA's   :6        NA's   :6       (Other)            :16
##   medianPrice        medianSize      numHouses         medianBR
##  Min.   : 324000   Min.   : 861   Min.   :   11.0   Min.   :1.000
##  1st Qu.: 477500   1st Qu.:1322   1st Qu.:  138.5   1st Qu.:3.000
##  Median : 605500   Median :1460   Median :  981.0   Median :3.000
##  Mean   : 711043   Mean   :1565   Mean   : 1727.0   Mean   :2.908
##  3rd Qu.: 800000   3rd Qu.:1672   3rd Qu.: 2409.5   3rd Qu.:3.000
##  Max.   :2200000   Max.   :3140   Max.   :14730.0   Max.   :4.000
##
```

```
summary(housing)
```

```
##                  county                   city            zip
##  Santa Clara County :70424   Oakland      : 14730   94565  :  4595
##  Alameda County     :60410   Santa Rosa   :  9917   94509  :  4302
##  Contra Costa County:59381   Fremont      :  9414   95123  :  4023
##  Solano County      :23404   San Francisco:  8137   95687  :  3652
##  San Mateo County   :22558   Evergreen    :  7947   94533  :  3472
##  Sonoma County      :21676   Antioch      :  7726   (Other):261457
##  (Other)            :23653   (Other)      :223635   NA's   :     5
##     street               price                br            lsqft
##  Length:281506      Min.   :   22000   Min.   :1.000   Min.   :       19
##  Class :character   1st Qu.:  400000   1st Qu.:2.000   1st Qu.:     4000
##  Mode  :character   Median :  530000   Median :3.000   Median :     5760
##                     Mean   :  602000   Mean   :3.024   Mean   :    65939
##                     3rd Qu.:  700000   3rd Qu.:4.000   3rd Qu.:     7701
##                     Max.   :20000000   Max.   :8.000   Max.   :418611600
##                                                        NA's   :21687
##      bsqft              year           date
##  Min.   :    122   Min.   :   0   Min.   :2003-04-27 02:00:00
##  1st Qu.:   1121   1st Qu.:1954   1st Qu.:2004-02-08 02:00:00
##  Median :   1430   Median :1971   Median :2004-10-24 02:00:00
##  Mean   :   1624   Mean   :1966   Mean   :2004-11-01 18:06:12
##  3rd Qu.:   1882   3rd Qu.:1985   3rd Qu.:2005-07-24 02:00:00
##  Max.   :1868120   Max.   :3894   Max.   :2006-06-04 02:00:00
##  NA's   :426       NA's   :9202
##       long             lat
##  Min.   :-123.6   Min.   :36.98
##  1st Qu.:-122.3   1st Qu.:37.50
```

```
##  Median :-122.1   Median :37.77
##  Mean   :-122.1   Mean   :37.78
##  3rd Qu.:-121.9   3rd Qu.:38.00
##  Max.   :-121.5   Max.   :38.85
##  NA's   :23316    NA's   :23316
##                                      quality
##  QUALITY_ADDRESS_RANGE_INTERPOLATION      :170719
##  gpsvisualizer                            : 31084
##  QUALITY_CITY_CENTROID                    : 20473
##  QUALITY_EXACT_PARCEL_CENTROID            : 17208
##  QUALITY_ZIP_CODE_TABULATION_AREA_CENTROID: 14980
##  (Other)                                  :  3726
##  NA's                                     : 23316
##               match               wk
##  Exact           :197044   Min.   :2003-04-21
##  Relaxed         : 30570   1st Qu.:2004-02-01
##  Relaxed; Soundex: 23338   Median :2004-10-18
##  Soundex         :  2573   Mean   :2004-10-26
##  1               :  2244   3rd Qu.:2005-07-18
##  (Other)         :  2421   Max.   :2006-05-29
##  NA's            : 23316
```

```r
dim(cities)
```

```
## [1] 163   7
```

```r
dim(housing)
```

```
## [1] 281506     15
```

```r
View(cities)
```

Q11. 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)

Write your response here

the cities object holds information about different communities eg alameda, antioch etc stored in variables such as longitude,latitude, county etc. These variables together are used to describe each community

the housing object holds information about different houses using variables such as county,city,zip,street etc to describe attributes of each house

#What both objects have in common is the variable "county"

Q12. Describe in words two problems that you see with the data. (2 pts)

Write your response here

#the first problem I observed is that there are many missing values in both the "cities"" and "housing" datasets.

secondly, I initally could not understand what information the housing dataset was trying to pass across(what things the variables were describing)

Q13. (2 pts.)

We will work the houses in Oakland, Sant Rosa, Campbell, and Sunnyvale 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 20706 observations

and 7 variables. (Note you may need to reformat any factor variables so that they

do not contain incorrect levels)

Your code below

```
SelectArea = (housing[housing$city == "Oakland" | housing$city == "Sant Rosa" |
           housing$city == "Campbell" | housing$city == "Sunnyvale", c("county",  "city","zip",
                                 "price","br","bsqft", "year")])

View(SelectArea)
summary(SelectArea)
```

```
##                    county            city            zip
##  Alameda County      :14729   Oakland  :14730   94605  : 2084
##  Santa Clara County  : 5976   Sunnyvale: 4062   95008  : 1914
##  Contra Costa County :    1   Campbell : 1914   94087  : 1787
##  Marin County        :    0   Alameda  :    0   94603  : 1552
```

```
##   Napa County          :    0   Alamo    :    0   94621  : 1414
##   San Francisco County:    0   Albany   :    0   94611  : 1357
##   (Other)              :    0   (Other)  :    0   (Other):10598
##       price                br              bsqft            year
##   Min.   :  53000    Min.   :1.000   Min.   :  336   Min.   :1885
##   1st Qu.: 366000    1st Qu.:2.000   1st Qu.: 1026   1st Qu.:1924
##   Median : 495000    Median :3.000   Median : 1283   Median :1947
##   Mean   : 540766    Mean   :2.767   Mean   : 1457   Mean   :1948
##   3rd Qu.: 661000    3rd Qu.:3.000   3rd Qu.: 1720   3rd Qu.:1968
##   Max.   :6750000    Max.   :8.000   Max.   :12582   Max.   :3880
##                                      NA's   :15      NA's   :398
```

Q14. (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 19064 observations.

Your code below

```r
quantile(SelectArea$price,0.95)
```

```
##    95%
## 960000
```

```r
quantile(SelectArea$bsqft, 0.95, na.rm=TRUE)
```

```
##    95%
## 2758.5
```

```r
SelectArea = SelectArea[(SelectArea$price < 960000) & (SelectArea$bsqft < 2758.5), ]
SelectArea = SelectArea[!apply(is.na(SelectArea),1, all ), ]
View(SelectArea)
```

Q15. (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.

Your code below

```
price_per_sqft = SelectArea$price/SelectArea$bsqft
SelectArea= cbind(SelectArea, price_per_sqft)
```

Q16 (2 pts.)

Create a vector called br_new, that is the number of bedrooms in the house, except

when the number is greater than 6, set it (br_new) to 6.

Your code below

```
 br_new = SelectArea$br
      View(br_new)
      br_new[br_new <= 6]= 6
```

Q17. (4 pts. 2 + 2 - see below)

Use the rainbow function to create a vector of 6 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.

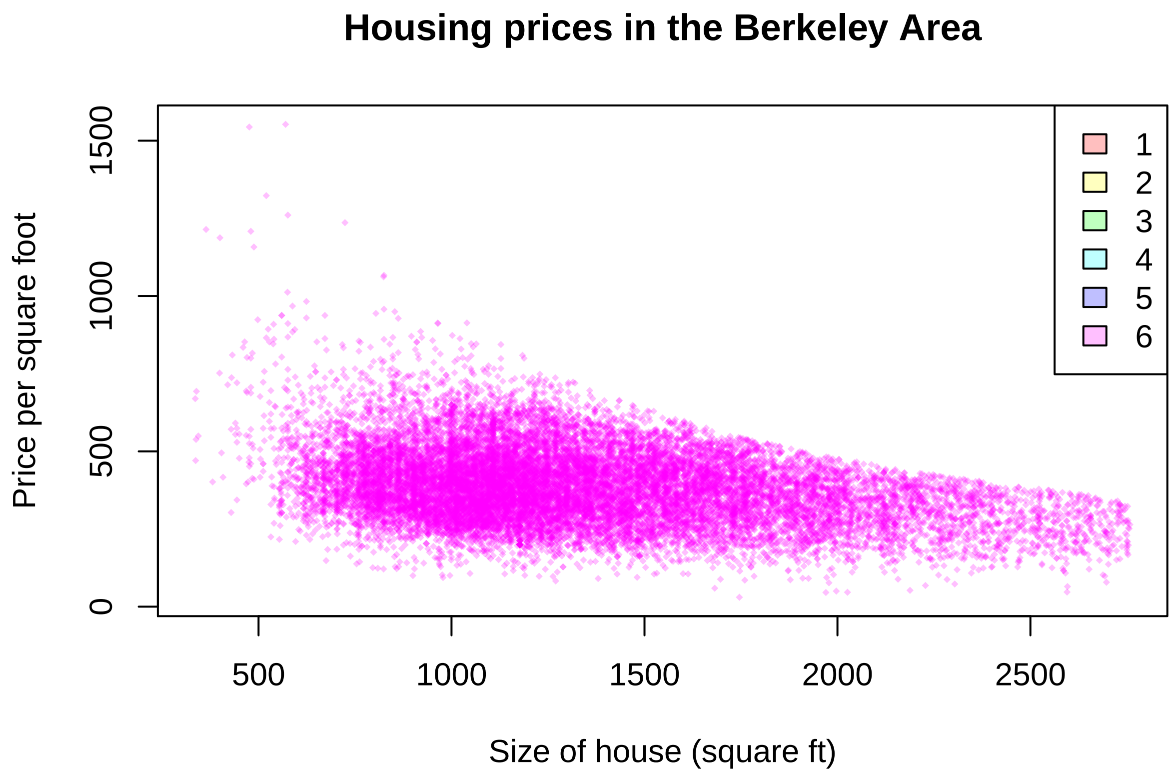
(2 pts.)

Your code below

```
rCols = rainbow(6, alpha = 0.25)
brCols = rCols[br_new]
View(brCols)
```

We are now ready to make a plot!

```
plot.new()
plot(price_per_sqft ~ bsqft, data = SelectArea,
     main = "Housing prices in the Berkeley Area",
     xlab = "Size of house (square ft)",
     ylab = "Price per square foot",
     col = brCols, pch = 18, cex = 0.5)
legend(legend = 1:6, fill = rCols, "topright")
```



what's your interpretation of the plot?

e.g., the trend? the cluster? the comparison? (1 pt.)

price_per_sqft and sqft(size of the house) have a fairly neutral relationship. The increase in size of the house does not cause rise in price_per_sqft rather the smaller houses seem to have higher cost per sqft.

There also seems to be a cluster of houses bewteen 1000 - 1500 sqft.