HW2_IS457_39

39

9/23/2018

PART 1. Warm up (4 pts)

Q1. 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)

Your code below

```
x = seq(0, 18, by = 3)
print(x)
```

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

```
vec = rep(x, each = 3)
print(vec)
```

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

Q2. Calculate the fraction of elements in vec that are more than or equal to 9. (2 pts)

hint: R can do vectorized operations.

Your code below

```
length(vec[vec >= 9])/length(vec)
```

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

```
# As calculated, the faction of elements in vec that are more than or equal to 9 is 57.14%.
```

Q3. 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)

Your code below

```
y = c(1:5)
print(y)
```

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

```
vec_1 = rep(y, y)
print(vec_1)
```

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

PART II. CO2 Data (9 pts)

Q4. 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.)

Your code below:

```
summary(CO2)
```

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

```
##  Min.   : 7.70
##  1st Qu.:17.90
##  Median :28.30
##  Mean   :27.21
##  3rd Qu.:37.12
##  Max.   :45.50
##
```

```
dim(CO2)
```

```
## [1] 84  5
```

```
class(CO2$Plant)
```

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

```
summary(CO2$Plant)
```

```
## Qn1 Qn2 Qn3 Qc1 Qc3 Qc2 Mn3 Mn2 Mn1 Mc2 Mc3 Mc1
##   7   7   7   7   7   7   7   7   7   7   7   7
```

```
class(CO2$Type)
```

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

```
summary(CO2$Type)
```

```
##      Quebec Mississippi
##          42          42
```

```
class(CO2$Treatment)
```

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

```
summary(CO2$Treatment)
```

```
## nonchilled    chilled
##         42         42
```

```
class(CO2$conc)
```

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

```
summary(CO2$conc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      95     175     350     435     675    1000
```

```
class(CO2$uptake)
```

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

```
summary(CO2$uptake)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.70   17.90   28.30   27.21   37.12   45.50
```

```
### Your answer below:
# The dimensions of the CO2 dataset are 84 rows and 5 columns.
# The data type of column Plant is ordered factor, totally 12 classes.
# The data type of column Type is factor, with 2 classes "Quebec" and "Mississippi".
# The data type of column Treatment is factor, with 2 classes "nonchilled" and "chilled".
```

```
# The data type of column conc is numeric.
# The data type of column uptake is numeric.
```

Q5. Show last 8 plants' uptake values (1 pt.)

Your code below:

```
help("tail")
tail(CO2$uptake, n = 8)
```

```
## [1] 14.4 10.6 18.0 17.9 17.9 17.9 18.9 19.9
```

Q6. Show all plants' uptake values except the first 20 plants'. (1 pt.)

Your code below:

```
CO2[-(1:20),]
```

```
##    Plant        Type  Treatment conc uptake
## 21   Qn3      Quebec nonchilled 1000   45.5
## 22   Qc1      Quebec    chilled   95   14.2
## 23   Qc1      Quebec    chilled  175   24.1
## 24   Qc1      Quebec    chilled  250   30.3
## 25   Qc1      Quebec    chilled  350   34.6
## 26   Qc1      Quebec    chilled  500   32.5
## 27   Qc1      Quebec    chilled  675   35.4
## 28   Qc1      Quebec    chilled 1000   38.7
## 29   Qc2      Quebec    chilled   95    9.3
## 30   Qc2      Quebec    chilled  175   27.3
## 31   Qc2      Quebec    chilled  250   35.0
## 32   Qc2      Quebec    chilled  350   38.8
## 33   Qc2      Quebec    chilled  500   38.6
## 34   Qc2      Quebec    chilled  675   37.5
## 35   Qc2      Quebec    chilled 1000   42.4
## 36   Qc3      Quebec    chilled   95   15.1
## 37   Qc3      Quebec    chilled  175   21.0
## 38   Qc3      Quebec    chilled  250   38.1
## 39   Qc3      Quebec    chilled  350   34.0
## 40   Qc3      Quebec    chilled  500   38.9
## 41   Qc3      Quebec    chilled  675   39.6
## 42   Qc3      Quebec    chilled 1000   41.4
## 43   Mn1 Mississippi nonchilled   95   10.6
## 44   Mn1 Mississippi nonchilled  175   19.2
## 45   Mn1 Mississippi nonchilled  250   26.2
## 46   Mn1 Mississippi nonchilled  350   30.0
## 47   Mn1 Mississippi nonchilled  500   30.9
## 48   Mn1 Mississippi nonchilled  675   32.4
## 49   Mn1 Mississippi nonchilled 1000   35.5
```

```
## 50   Mn2 Mississippi nonchilled   95   12.0
## 51   Mn2 Mississippi nonchilled  175   22.0
## 52   Mn2 Mississippi nonchilled  250   30.6
## 53   Mn2 Mississippi nonchilled  350   31.8
## 54   Mn2 Mississippi nonchilled  500   32.4
## 55   Mn2 Mississippi nonchilled  675   31.1
## 56   Mn2 Mississippi nonchilled 1000   31.5
## 57   Mn3 Mississippi nonchilled   95   11.3
## 58   Mn3 Mississippi nonchilled  175   19.4
## 59   Mn3 Mississippi nonchilled  250   25.8
## 60   Mn3 Mississippi nonchilled  350   27.9
## 61   Mn3 Mississippi nonchilled  500   28.5
## 62   Mn3 Mississippi nonchilled  675   28.1
## 63   Mn3 Mississippi nonchilled 1000   27.8
## 64   Mc1 Mississippi    chilled   95   10.5
## 65   Mc1 Mississippi    chilled  175   14.9
## 66   Mc1 Mississippi    chilled  250   18.1
## 67   Mc1 Mississippi    chilled  350   18.9
## 68   Mc1 Mississippi    chilled  500   19.5
## 69   Mc1 Mississippi    chilled  675   22.2
## 70   Mc1 Mississippi    chilled 1000   21.9
## 71   Mc2 Mississippi    chilled   95    7.7
## 72   Mc2 Mississippi    chilled  175   11.4
## 73   Mc2 Mississippi    chilled  250   12.3
## 74   Mc2 Mississippi    chilled  350   13.0
## 75   Mc2 Mississippi    chilled  500   12.5
## 76   Mc2 Mississippi    chilled  675   13.7
## 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
```

```
CO2$uptake[-(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
## [15] 42.4 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
## [29] 35.5 12.0 22.0 30.6 31.8 32.4 31.1 31.5 11.3 19.4 25.8 27.9 28.5 28.1
## [43] 27.8 10.5 14.9 18.1 18.9 19.5 22.2 21.9  7.7 11.4 12.3 13.0 12.5 13.7
## [57] 14.4 10.6 18.0 17.9 17.9 17.9 18.9 19.9
```

Q7. Calculate the mean of uptake subseted by the "Treatment" variable.(1 pt)

hint: apply function family.

Your code below:

```
mean(CO2$uptake[CO2$Treatment == "nonchilled"])
```

```
## [1] 30.64286
```

```
# The mean of uptake where "Treatment" is "nonchilled" is 30.64.
mean(CO2$uptake[CO2$Treatment == "chilled"])
```

```
## [1] 23.78333
```

```
# The mean of uptake where "Treatment" is "chilled" is 23.78.
```

Q8. 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 == "non-chilled"), return value TRUE when uptake > 40.

Your code below:

```
uptake_treatment = rep(TRUE, 84)
ind = length(CO2$uptake)
print(ind)
```

```
## [1] 84
```

```
for (i in 1:ind){
  if (CO2$Treatment[i] == "chilled"){
    if (CO2$uptake[i] > 30){
      uptake_treatment[i] = TRUE
    }
    else{
      uptake_treatment[i] = FALSE
    }
  }
  else if (CO2$Treatment[i] == "nonchilled"){
    if (CO2$uptake[i] > 40){
      uptake_treatment[i] = TRUE
    }
    else{
      uptake_treatment[i] = FALSE
```

```
    }
  }
}
print(uptake_treatment)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [12]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [23] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
## [34]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
## [45] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Q9. 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 treatement. 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)

Your code below:

```
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)

Your code below

```
uptake_test = ifelse(CO2$Treatment =="chilled", test_val[2], test_val[1])
```

Finally, use uptake_test and the uptake column to create the desired vector, and

call it uptake_treatment2. (1 pt)

Your code below

```
uptake_treatment2 = ifelse(CO2$uptake>uptake_test, TRUE, FALSE)
print(uptake_treatment2)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [12]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [23] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
## [34]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
## [45] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

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

Load the data into R.

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

Q10. (3 pts.)

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

Your code below

```
class(cities)
```

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

```
objects(cities)
```

```
## [1] "county"      "latitude"    "longitude"   "medianBR"    "medianPrice"
## [6] "medianSize"  "numHouses"
```

```
class(housing)
```

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

```
objects(housing)
```

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

```
### Your answer here
# There are two objects in SFHousing.rda, "cities" and "housing". Both of the two objects
# are data frame.
```

Q11. give a summary of each object, including a summary of each variable and the dimension of the object. (4 pts)

Your code below

```
objects(cities)
```

```
## [1] "county"      "latitude"    "longitude"   "medianBR"    "medianPrice"
## [6] "medianSize"  "numHouses"
```

```
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
##
```

```
dim(cities)
```

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

```
class(cities$longitude)
```

```
## [1] "array"
```

```
class(cities$latitude)
```

```
## [1] "array"
```

```
class(cities$county)
```

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

```r
class(cities$medianPrice)
```

```
## [1] "array"
```

```r
class(cities$medianSize)
```

```
## [1] "array"
```

```r
class(cities$numHouses)
```

```
## [1] "array"
```

```r
class(cities$medianBR)
```

```
## [1] "array"
```

```r
objects(housing)
```

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

```r
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
```

```
dim(housing)
```

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

```
class(housing$county)
```

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

```
class(housing$city)
```

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

```
class(housing$zip)
```

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

```
class(housing$street)
```

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

```
class(housing$price)
```

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

```
class(housing$br)
```

```
## [1] "integer"
```

```
class(housing$lsqft)
```

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

```
class(housing$bsqft)
```

```
## [1] "integer"
```

```
class(housing$year)
```

```
## [1] "integer"
```

```
class(housing$date)
```

```
## [1] "POSIXt"  "POSIXct"
```

```
class(housing$long)
```

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

```r
class(housing$lat)
```

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

```r
class(housing$quality)
```

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

```r
class(housing$match)
```

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

```r
# The object "cities" is consisted of 163 rows and 7 columns.
# There are 7 variables in the object of cities, which are listed as "longitude" (array),
# "latitude" (array), "county" (factor), "medianPrice" (array), "medianSize" (array),
# "numHouse" (array), "medianBR" (array)
# The object "housing" is consisted of 281506 rows and 15 columns.
# There are 15 varibles in the object of housing, which are listed as "county" (factor), "city" (factor
# "zip" (factor), "street" (character), "price" (numeric), "br" (integer), "lsqft" (numeric),
# "bsqft" (integer), "year" (integer), "date" (POSIXt), "long" (numeric), "lat" (numeric),
# "quality" (factor), "match" (factor).
```

Q12. 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

```r
# The variable "county" is the common factor shared by two objects while linking them.
```

Q13. Describe in words two problems that you see with the data. (2 pts)

Write your response here

```r
# (1) In object "housing", the variable "long" contains 23315 NA values.
# (2) In object "housing", the variable "quality" has some outliers.
```

Q14. (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)

Your code below

```
SelectArea = housing[housing$city %in% c("San Francisco", "Fremont", "Vallejo",
                                         "Concord", "Livermore"), c(1,2,3,5,6,8,9)]
dim(SelectArea)
```

```
## [1] 36686     7
```

```
summary(SelectArea)
```

```
##                   county                 city           zip
##  Alameda County      :14256   Fremont      :9414   94591  : 3369
##  San Francisco County: 8137   San Francisco:8137   94536  : 3292
##  Solano County       : 7183   Vallejo      :7183   94521  : 2779
##  Contra Costa County : 7110   Concord      :7109   94551  : 2467
##  Marin County        :    0   Livermore    :4843   94550  : 2376
##  Napa County         :    0   Alameda      :   0   94538  : 2279
##  (Other)             :    0   (Other)      :   0   (Other):20124
##      price                br            bsqft              year
##  Min.   :   48000   Min.   :1.000   Min.   :    122   Min.   :   2
##  1st Qu.:  379000   1st Qu.:2.000   1st Qu.:   1066   1st Qu.:1954
##  Median :  505000   Median :3.000   Median :   1336   Median :1970
##  Mean   :  566248   Mean   :2.905   Mean   :   1537   Mean   :1963
##  3rd Qu.:  660000   3rd Qu.:4.000   3rd Qu.:   1736   3rd Qu.:1985
##  Max.   :10875000   Max.   :8.000   Max.   :1868120   Max.   :2005
##                                     NA's   :45        NA's   :1347
```

Q15. (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.

Your code below

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

```
##     95%
## 1100000
```

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

```
##  95%
## 2698
```

```
SelectArea = SelectArea[SelectArea$price < 1100000 & SelectArea$bsqft < 2698,]
SelectArea <- SelectArea[!is.na(SelectArea$bsqft), ]
dim(SelectArea)
```

```
## [1] 33693     7
```

```
summary(SelectArea)
```

```
##                  county                city           zip
##  Alameda County      :13200   Fremont      :8853   94536  : 3203
##  Solano County       : 6905   Vallejo      :6905   94591  : 3130
##  Contra Costa County : 6856   Concord      :6856   94521  : 2561
##  San Francisco County: 6732   San Francisco:6732   94551  : 2363
##  Marin County        :    0   Livermore    :4347   94538  : 2260
##  Napa County         :    0   Alameda      :   0   94550  : 1984
##  (Other)             :    0   (Other)      :   0   (Other):18192
##      price               br            bsqft           year
```

```
##  Min.   :  48000   Min.   :1.000   Min.   : 277   Min.   :   2
##  1st Qu.: 367500   1st Qu.:2.000   1st Qu.:1042   1st Qu.:1954
##  Median : 485000   Median :3.000   Median :1290   Median :1970
##  Mean   : 503028   Mean   :2.824   Mean   :1369   Mean   :1963
##  3rd Qu.: 620000   3rd Qu.:3.000   3rd Qu.:1640   3rd Qu.:1984
##  Max.   :1098000   Max.   :8.000   Max.   :2697   Max.   :2005
##                                                   NA's   :976
```

Q16. (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
summary(price_per_sqft)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   47.29  286.50  353.10  384.15  444.31 2351.94
```

```
SelectArea = cbind(SelectArea, price_per_sqft)
summary(SelectArea)
```

```
##                   county                 city           zip
##  Alameda County      :13200   Fremont      :8853   94536  : 3203
##  Solano County       : 6905   Vallejo      :6905   94591  : 3130
##  Contra Costa County : 6856   Concord      :6856   94521  : 2561
##  San Francisco County: 6732   San Francisco:6732   94551  : 2363
##  Marin County        :    0   Livermore    :4347   94538  : 2260
##  Napa County         :    0   Alameda      :   0   94550  : 1984
##  (Other)             :    0   (Other)      :   0   (Other):18192
##      price               br            bsqft           year
##  Min.   :  48000   Min.   :1.000   Min.   : 277   Min.   :   2
##  1st Qu.: 367500   1st Qu.:2.000   1st Qu.:1042   1st Qu.:1954
##  Median : 485000   Median :3.000   Median :1290   Median :1970
##  Mean   : 503028   Mean   :2.824   Mean   :1369   Mean   :1963
##  3rd Qu.: 620000   3rd Qu.:3.000   3rd Qu.:1640   3rd Qu.:1984
##  Max.   :1098000   Max.   :8.000   Max.   :2697   Max.   :2005
##                                                   NA's   :976
##  price_per_sqft
##  Min.   :  47.29
##  1st Qu.: 286.50
##  Median : 353.10
##  Mean   : 384.15
##  3rd Qu.: 444.31
##  Max.   :2351.94
##
```

Q17. (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.

Your code below

```
br_new = ifelse(SelectArea$br>5, 5, SelectArea$br)
```

Q18. (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.)

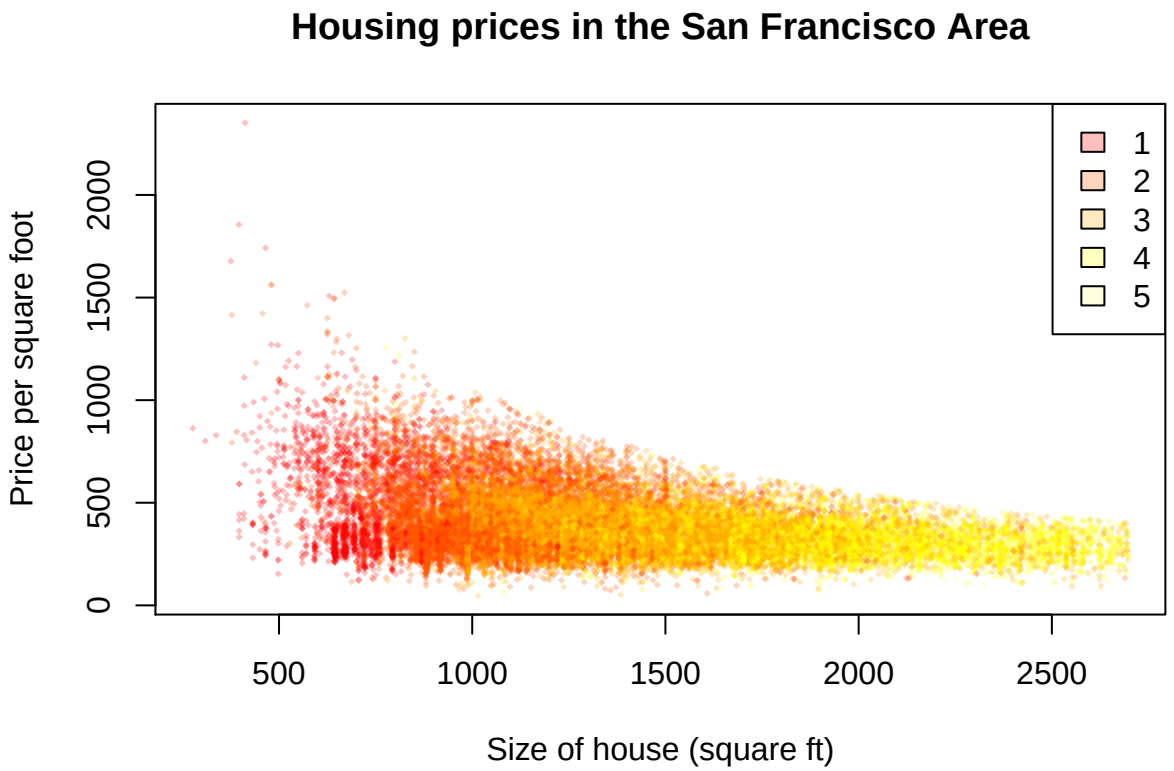
Your code below

```
help("heat.colors")
rCols = heat.colors(5, alpha = 0.25)
brCols = rCols[br_new]
```

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.)
# The relationship between price per square foot and size of house is generally in
# negative-correlation, which means the bigger the house is, the lower the price per square
# foot is.
```