Homework2

向晏平

The data set calif_penn_2011.csv contains information about the housing stock of California and Pennsylvania, as of 2011. Information as aggregated into "Census tracts", geographic regions of a few thousand people which are supposed to be fairly homogeneous economically and socially.

1. *Loading and cleaning*

a. Load the data into a dataframe called `ca_pa`.

```
ca_pa <- read.csv("data/calif_penn_2011.csv")[,-1]
```

b. How many rows and columns does the dataframe have?

```
nrow(ca_pa)
```

```
## [1] 11275
```

```
ncol(ca_pa)
```

```
## [1] 33
```

c. Run this command, and explain, in words, what this does:

```
colSums(apply(ca_pa,c(1,2),is.na))
```

```
##                    GEO.id2                    STATEFP
##                          0                          0
##                   COUNTYFP                    TRACTCE
```

```
##                          0                          0
##                 POPULATION                   LATITUDE
##                          0                          0
##                  LONGITUDE           GEO.display.label
##                          0                          0
##          Median_house_value                Total_units
##                        599                          0
##               Vacant_units               Median_rooms
##                          0                        157
##  Mean_household_size_owners Mean_household_size_renters
##                        215                        152
##         Built_2005_or_later          Built_2000_to_2004
##                         98                         98
##                Built_1990s                Built_1980s
##                         98                         98
##                Built_1970s                Built_1960s
##                         98                         98
##                Built_1950s                Built_1940s
##                         98                         98
##       Built_1939_or_earlier                 Bedrooms_0
##                         98                         98
##                 Bedrooms_1                 Bedrooms_2
##                         98                         98
##                 Bedrooms_3                 Bedrooms_4
##                         98                         98
##          Bedrooms_5_or_more                     Owners
##                         98                        100
##                    Renters     Median_household_income
##                        100                        115
##       Mean_household_income
##                        126
```

apply() 函数按指定方向作用对象一个函数，并返回与作用方向相关的一个向量或数组，在这里方向参数 MARGIN 选择 2 或 c(1,2) 所得结果相

同。外层的 colSums() 函数对返回的数组按列求和，得到每个变量存在的缺失值总数。

d. The function `na.omit()` takes a dataframe and returns a new dataframe, omitting any row containing an NA value.Use it to purge the data set of rows with incomplete data.

```
ca_pa_omit <- na.omit(ca_pa)
```

e. How many rows did this eliminate?

```
nrow(ca_pa)-nrow(ca_pa_omit)
```

```
## [1] 670
```

f. Are your answers in (c) and (e) compatible? Explain. 二者答案并不冲突，(c) 统计各列的缺失值总和，(e) 删除存在缺失值的行，两者并无相等的关系。

2. *This Very New House*

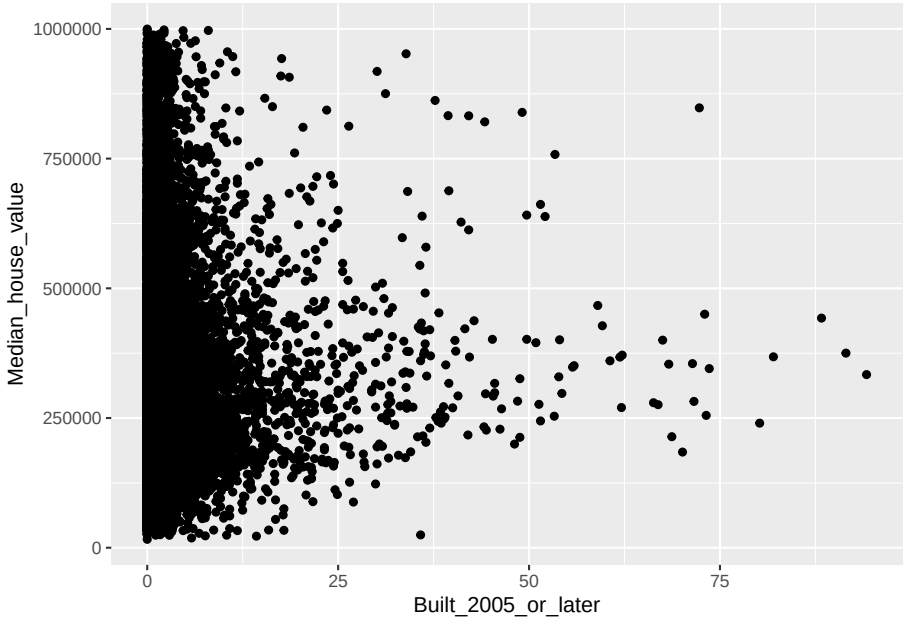
a. The variable `Built_2005_or_later` indicates the percentage of houses in each Census tract built since 2005. Plot median house prices against this variable.

```
str(ca_pa_omit)
```

```
## 'data.frame':    10605 obs. of  33 variables:
##  $ GEO.id2                   : num  6e+09 6e+09 6e+09 6e+09 6e+09 ...
##  $ STATEFP                   : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ COUNTYFP                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ TRACTCE                   : int  400200 400300 400400 400500 400600 400700 40080
##  $ POPULATION                : int  1974 4865 3703 3517 1571 4206 3594 2302 5678 41
##  $ LATITUDE                  : num  37.8 37.8 37.8 37.8 37.8 ...
##  $ LONGITUDE                 : num  -122 -122 -122 -122 -122 ...
```

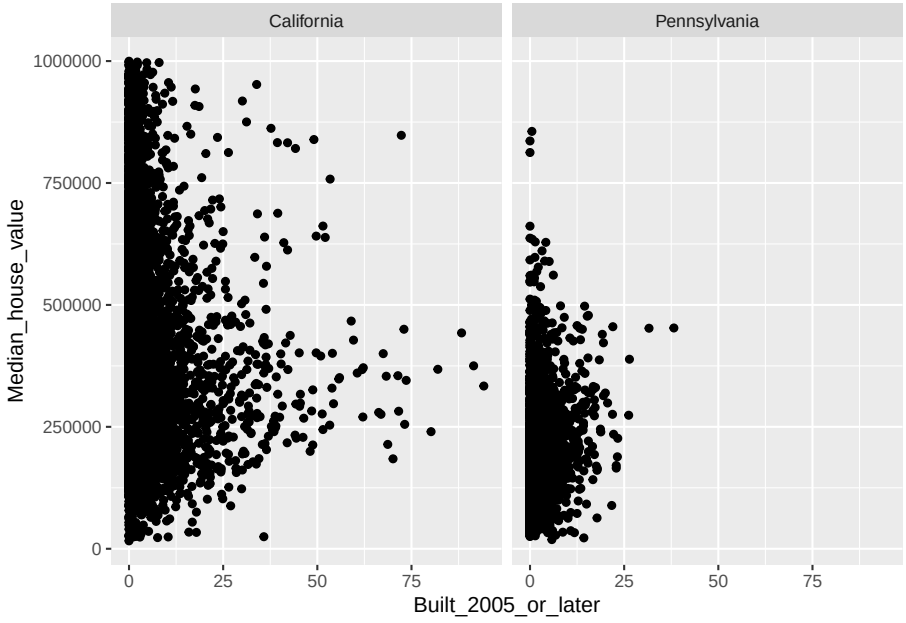
```
##  $ GEO.display.label          : Factor w/ 11275 levels "Census Tract 1, Berks County
##  $ Median_house_value         : int  909600 748700 773600 579200 480800 460800 47350
##  $ Total_units                : int  929 2655 1911 1703 781 1977 1738 1202 2665 2182
##  $ Vacant_units               : int  37 134 68 71 65 236 257 80 500 148 ...
##  $ Median_rooms               : num  6 4.6 5 4.5 4.8 4.3 4.3 4.4 4.6 3.5 ...
##  $ Mean_household_size_owners : num  2.53 2.45 2.04 2.66 2.58 2.72 2.17 2.7 2.75 2.3
##  $ Mean_household_size_renters: num  1.81 1.66 2.19 1.72 2.18 2.15 1.93 1.92 2.08 1.
##  $ Built_2005_or_later        : num  0 0 0 0 0 0 13.1 0 0 0.4 ...
##  $ Built_2000_to_2004         : num  1.2 0 0.2 0.2 0 0.6 4.1 2.2 2.2 1.1 ...
##  $ Built_1990s                : num  0 2.3 1.3 1.1 1.2 1.8 1.6 0.6 0 1.7 ...
##  $ Built_1980s                : num  1.3 3.2 0 1.9 1.4 2.2 2.4 5.9 0.5 0.5 ...
##  $ Built_1970s                : num  6.1 5.2 4.9 3.7 1 3.3 7.8 0 4.3 4.1 ...
##  $ Built_1960s                : num  6.5 8.3 4.3 5.8 6.5 0.8 3.7 5.5 11.2 3.6 ...
##  $ Built_1950s                : num  1 5.3 8 6 19.7 9.4 7.5 9.1 11.3 13.4 ...
##  $ Built_1940s                : num  10.8 7.8 10.4 7.5 17 9.7 13.3 14.7 8.5 11.8 ...
##  $ Built_1939_or_earlier      : num  73.2 68 71.1 73.8 53.1 72.4 46.5 62 62.1 63.3 .
##  $ Bedrooms_0                 : num  3 11.5 5.2 4.9 3.5 8.2 8.9 14.2 6.1 11 ...
##  $ Bedrooms_1                 : num  16.4 28.4 27.7 30.2 20.4 22.3 25 20.1 29.3 44 .
##  $ Bedrooms_2                 : num  27.4 29.2 33.7 38.1 40.1 43.2 37.5 39.4 35.4 24
##  $ Bedrooms_3                 : num  34.4 20.4 21.9 19.3 30.7 16.7 25 18.3 25.3 16.1
##  $ Bedrooms_4                 : num  17.5 7.9 7.3 5.4 4.6 6.5 2.1 5.5 3.9 3.7 ...
##  $ Bedrooms_5_or_more         : num  1.2 2.7 4.2 2.1 0.8 3.1 1.4 2.5 0 0.8 ...
##  $ Owners                     : num  66 45.1 45 43.6 51 32.2 28.3 31.7 35.1 16.8 ...
##  $ Renters                    : num  34 54.9 55 56.4 49 67.8 71.7 68.3 64.9 83.2 ...
##  $ Median_household_income    : int  111667 66094 87306 62386 55658 38646 52837 5909
##  $ Mean_household_income      : int  195229 105877 106248 74604 73933 56705 66822 59
##  - attr(*, "na.action")= 'omit' Named int  1 25 41 43 114 132 135 145 154 155 ...
##   ..- attr(*, "names")= chr  "1" "25" "41" "43" ...
```

```
ggplot(data = ca_pa_omit) +
  geom_point(aes(x = Built_2005_or_later, y = Median_house_value ))
```



b. Make a new plot, or pair of plots, which breaks this out by state. Note that the state is recorded in the `STATEFP` variable, with California being state 6 and Pennsylvania state 42.

```
ggplot(data = ca_pa_omit) +
  geom_point(aes(x = Built_2005_or_later, y = Median_house_value )) +
  facet_wrap(vars(STATEFP),labeller = labeller(STATEFP = c("6" = "California","42" = "P
```



3. *Nobody Home*
   The vacancy rate is the fraction of housing units which are not occupied. The dataframe contains columns giving the total number of housing units for each Census tract, and the number of vacant housing units.

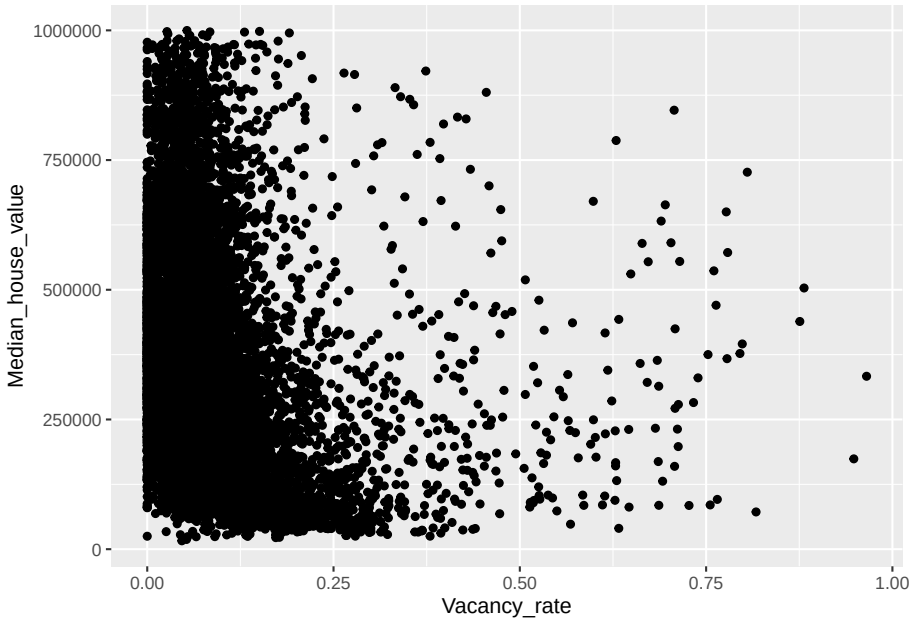
   a. Add a new column to the dataframe which contains the vacancy rate. What are the minimum, maximum, mean, and median vacancy rates?

```
ca_pa_add <- ca_pa_omit %>%
  mutate(Vacancy_rate = Vacant_units / Total_units)
summary(ca_pa_add$Vacancy_rate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.03846 0.06767 0.08889 0.10921 0.96531
```

   b. Plot the vacancy rate against median house value.

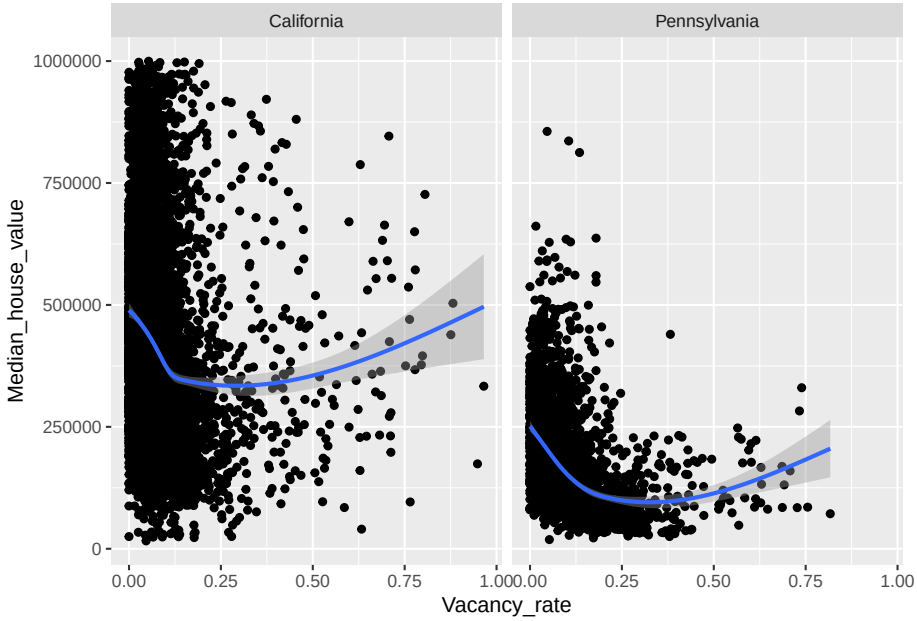
```
ca_pa_add %>%
  ggplot(aes(x = Vacancy_rate, y = Median_house_value )) +
  geom_point()
```



c. Plot vacancy rate against median house value separately for California and for Pennsylvania. Is there a difference?

```
ca_pa_add %>%
  ggplot(aes(x = Vacancy_rate, y = Median_house_value )) +
  geom_point() +
  geom_smooth() +
  facet_wrap(vars(STATEFP),labeller = labeller(STATEFP = c("6" = "California","42" = "P
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Pennsylvania 州的空置率在较低房屋价格时较高，California 州没有这样明显的趋势，由此可见 C 州可能较富裕。

4. The column `COUNTYFP` contains a numerical code for counties within each state. We are interested in Alameda County (county 1 in California), Santa Clara (county 85 in California), and Allegheny County (county 3 in Pennsylvania).

a. Explain what the block of code at the end of this question is supposed to accomplish, and how it does it.

题后代码块将 Alameda County 对应的行代码先挑出来，取出其对应的第 10 列值，求其中位数。

b. Give a single line of R which gives the same final answer as the block of code. Note: there are at least two ways to do this; you just have to find one.

```
my_accamhv <- ca_pa %>%
  filter(COUNTYFP == 1, STATEFP == 6) %>%
```

```
  select(10)
median(my_accamhv[,1])
```

```
## [1] 1560
```

c. For Alameda, Santa Clara and Allegheny Counties, what were the average percentages of housing built since 2005?

```
my_ac <- ca_pa %>%
  filter(COUNTYFP %in% c(1,85,3)) %>%
  select(COUNTYFP,Built_2005_or_later) %>%
  na.omit()
ave <- my_ac %>%
  group_by(COUNTYFP) %>%
  summarise(ave = mean(Built_2005_or_later)) %>%
  ungroup()
ave
```

```
## # A tibble: 3 x 2
##   COUNTYFP   ave
##      <int> <dbl>
## 1        1  3.13
## 2        3  1.88
## 3       85  3.05
```

d. The `cor` function calculates the correlation coefficient between two variables. What is the correlation between median house value and the percent of housing built since 2005 in (i) the whole data, (ii) all of California, (iii) all of Pennsylvania, (iv) Alameda County, (v) Santa Clara County and (vi) Allegheny County?

```
cor(ca_pa_omit$Median_house_value,ca_pa_omit$Built_2005_or_later)
```

```
## [1] -0.01893186
```

```
my_cor1 <- ca_pa_omit %>%
  group_by(STATEFP) %>%
  summarise(cor1 = cor(ca_pa_omit$Median_house_value,ca_pa_omit$Built_2005_or_later)) %
  ungroup()
my_cor1[,"cor1"]
```

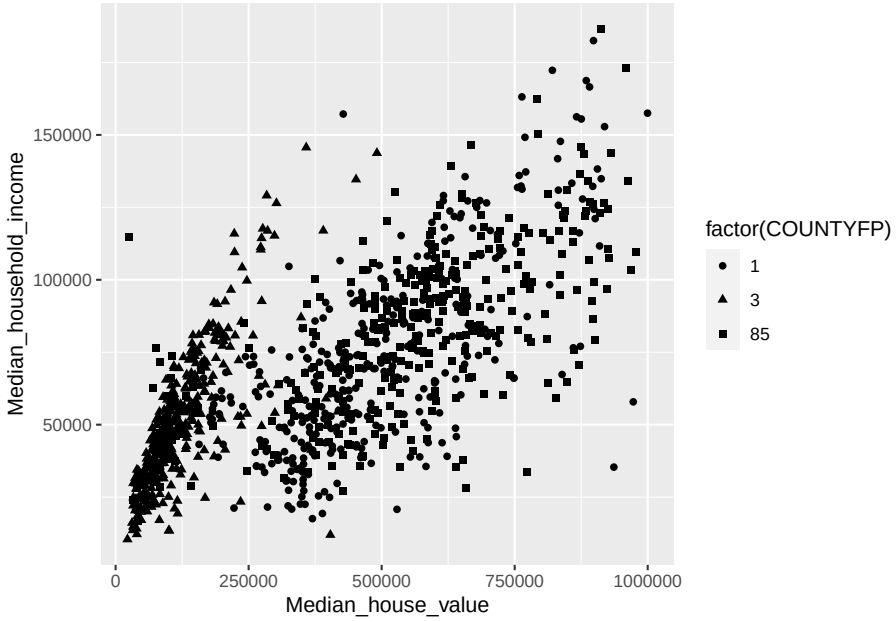
```
## # A tibble: 2 x 1
##      cor1
##     <dbl>
## 1 -0.0189
## 2 -0.0189
```

```
my_cor2 <- ca_pa_omit %>%
  group_by(COUNTYFP) %>%
  summarise(cor2 = cor(ca_pa_omit$Median_house_value,ca_pa_omit$Built_2005_or_later)) %
  ungroup()
my_cor2[my_cor2$COUNTYFP %in% c(1,3,85),"cor2"]
```

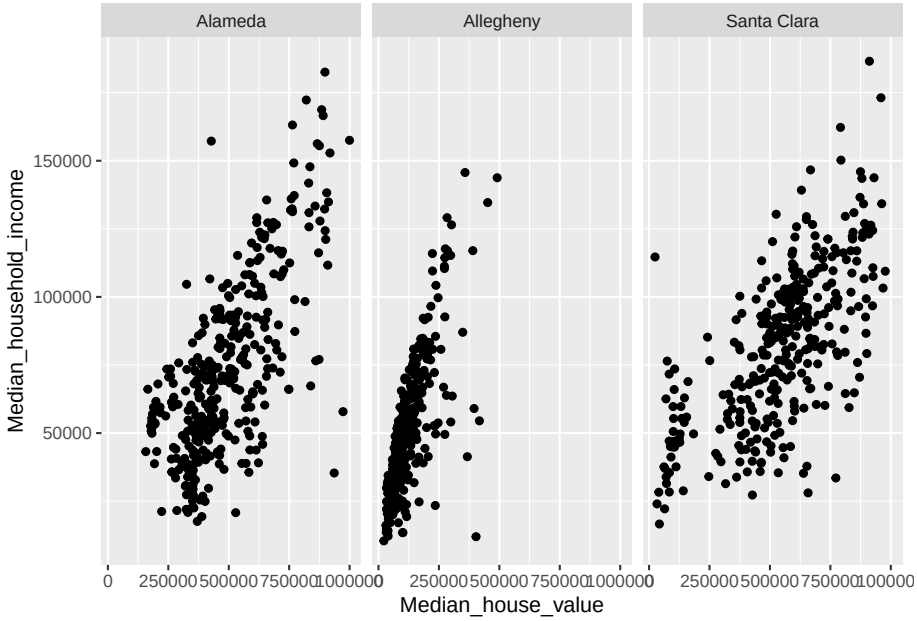
```
## # A tibble: 3 x 1
##      cor2
##     <dbl>
## 1 -0.0189
## 2 -0.0189
## 3 -0.0189
```

e. Make three plots, showing median house values against median income, for Alameda, Santa Clara, and Allegheny Counties. (If you can fit the information into one plot, clearly distinguishing the three counties, that's OK too.)

```
ca_pa_omit %>%
  filter(COUNTYFP %in% c(1,85,3)) %>%
  ggplot(aes(x = Median_house_value,y = Median_household_income,
             shape = factor(COUNTYFP))) +
  geom_point()
```




```
ca_pa_omit %>%
  filter(COUNTYFP %in% c(1,85,3)) %>%
  ggplot(aes(x = Median_house_value,y = Median_household_income)) +
  geom_point()  +
  facet_wrap(~COUNTYFP,labeller = labeller(COUNTYFP =c("1"="Alameda",
                                                       "85"="Santa Clara","3"="Alleghen
```




```
acca <- c()
for (tract in 1:nrow(ca_pa)) {
  if (ca_pa$STATEFP[tract] == 6) {
    if (ca_pa$COUNTYFP[tract] == 1) {
      acca <- c(acca, tract)
    }
  }
}
accamhv <- c()
for (tract in acca) {
  accamhv <- c(accamhv, ca_pa[tract,10])
}
median(accamhv)
```

MB.Ch1.11. Run the following code:

```
gender <- factor(c(rep("female", 91), rep("male", 92)))
table(gender)
```

```
## gender
## female   male
##     91     92
```

```
gender <- factor(gender, levels=c("male", "female"))
table(gender)
```

```
## gender
##   male female
##     92     91
```

```
gender <- factor(gender, levels=c("Male", "female"))
# Note the mistake: "Male" should be "male"
table(gender)
```

```
## gender
##   Male female
##      0     91
```

```
table(gender, exclude=NULL)
```

```
## gender
##   Male female   <NA>
##      0     91     92
```

```
rm(gender)  # Remove gender
```

Explain the output from the successive uses of table(). table() 函数给出变量值不同观测的频数，参数 levels 指定 factor 的输出顺序，如果没有对应则输出 0.factor() 将不在 levels 中的值全部指定为 NA，从而后两个输出中没有 male 而 NA 个数与原 male 相同。

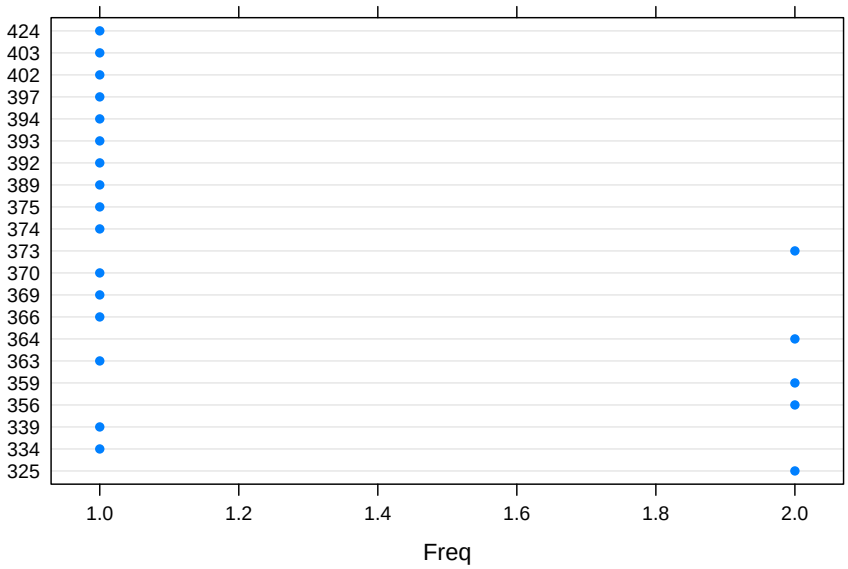
MB.Ch1.12. Write a function that calculates the proportion of values in a vector x that exceed some value cutoff.

(a) Use the sequence of numbers 1, 2, . . . , 100 to check that this function gives the result that is expected.

```
x <- 1:100
my_fun1 <- function(x){
  l <- length(x)
  table(x)/l
}
```

(b) Obtain the vector ex01.36 from the Devore6 (or Devore7) package. These data give the times required for individuals to escape from an oil platform during a drill. Use dotplot() to show the distribution of times. Calculate the proportion of escape times that exceed 7 minutes.

```
library(Devore7)
data(ex01.36)
dotplot(ex01.36)
```




```
sum(ex01.36[1] > 420)/length(ex01.36[[1]])
```

```
## [1] 0.03846154
```

MB.Ch1.18. The Rabbit data frame in the MASS library contains blood pressure change measurements on five rabbits (labeled as R1, R2, . . . ,R5) under various control and treatment conditions. Read the help file for more information. Use the unstack() function (three times) to convert Rabbit to the following form:

Treatment Dose R1 R2 R3 R4 R5
1 Control 6.25 0.50 1.00 0.75 1.25 1.5
2 Control 12.50 4.50 1.25 3.00 1.50 1.5
....

```
data("Rabbit")
attach(Rabbit)
R <- cbind(Treatment = unstack(Rabbit, Treatment ~ Animal)[,1],
      Dose = unstack(Rabbit, Dose ~ Animal)[,1],
      unstack(Rabbit, BPchange ~ Animal))
detach(Rabbit)
R
```

```
##    Treatment   Dose    R1    R2    R3    R4   R5
## 1    Control   6.25  0.50  1.00  0.75  1.25  1.5
## 2    Control  12.50  4.50  1.25  3.00  1.50  1.5
## 3    Control  25.00 10.00  4.00  3.00  6.00  5.0
## 4    Control  50.00 26.00 12.00 14.00 19.00 16.0
## 5    Control 100.00 37.00 27.00 22.00 33.00 20.0
## 6    Control 200.00 32.00 29.00 24.00 33.00 18.0
## 7        MDL   6.25  1.25  1.40  0.75  2.60  2.4
## 8        MDL  12.50  0.75  1.70  2.30  1.20  2.5
## 9        MDL  25.00  4.00  1.00  3.00  2.00  1.5
## 10       MDL  50.00  9.00  2.00  5.00  3.00  2.0
## 11       MDL 100.00 25.00 15.00 26.00 11.00  9.0
## 12       MDL 200.00 37.00 28.00 25.00 22.00 19.0
```

```
#require(reshape)
#recast(Rabbit, Treatment + Dose ~ Animal, measure.var="BPchange")
```