Homework 2

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`.
    b. How many rows and columns does the dataframe have?
    c. Run this command, and explain, in words, what this does:

    ```
    colSums(apply(ca_pa,c(1,2),is.na))
    Explain: It applies function is.na to dataframe
    ca_pa then sum the results by cols.
    ```

```r
ca_pa<-readr::read_csv("data/calif_penn_2011.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```r
dim(ca_pa)# answer of b
```

```
## [1] 11275    34
```

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

```
##                           X1                     GEO.id2
##                            0                           0
##                      STATEFP                    COUNTYFP
##                            0                           0
##                      TRACTCE                  POPULATION
##                            0                           0
##                     LATITUDE                   LONGITUDE
##                            0                           0
##            GEO.display.label          Median_house_value
##                            0                         599
##                  Total_units                Vacant_units
##                            0                           0
##                 Median_rooms  Mean_household_size_owners
##                          157                         215
## Mean_household_size_renters          Built_2005_or_later
##                          152                          98
##            Built_2000_to_2004                 Built_1990s
##                           98                          98
##                  Built_1980s                 Built_1970s
##                           98                          98
```

```
##                  Built_1960s                  Built_1950s
##                           98                           98
##                  Built_1940s         Built_1939_or_earlier
##                           98                           98
##                   Bedrooms_0                   Bedrooms_1
##                           98                           98
##                   Bedrooms_2                   Bedrooms_3
##                           98                           98
##                   Bedrooms_4            Bedrooms_5_or_more
##                           98                           98
##                       Owners                      Renters
##                          100                          100
##      Median_household_income        Mean_household_income
##                          115                          126
```

```
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.
e. How many rows did this eliminate?
f. Are your answers in (c) and (e) compatible? Explain.
```

```r
ca_pa_new<-na.omit(ca_pa)
nrow(ca_pa_new)# answer of e
```

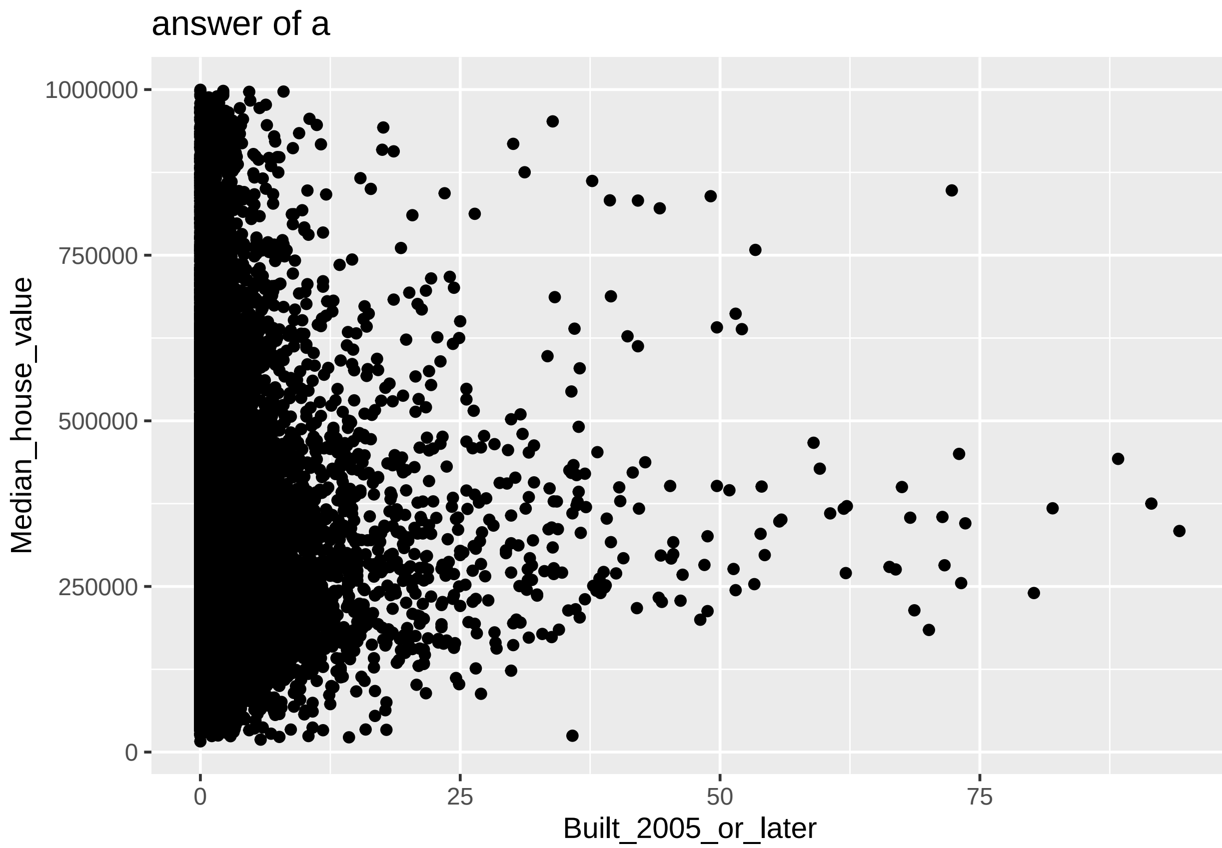
```
## [1] 10605
```

```r
#Explain of f:They're compatible because some rows have more than one NA.
```

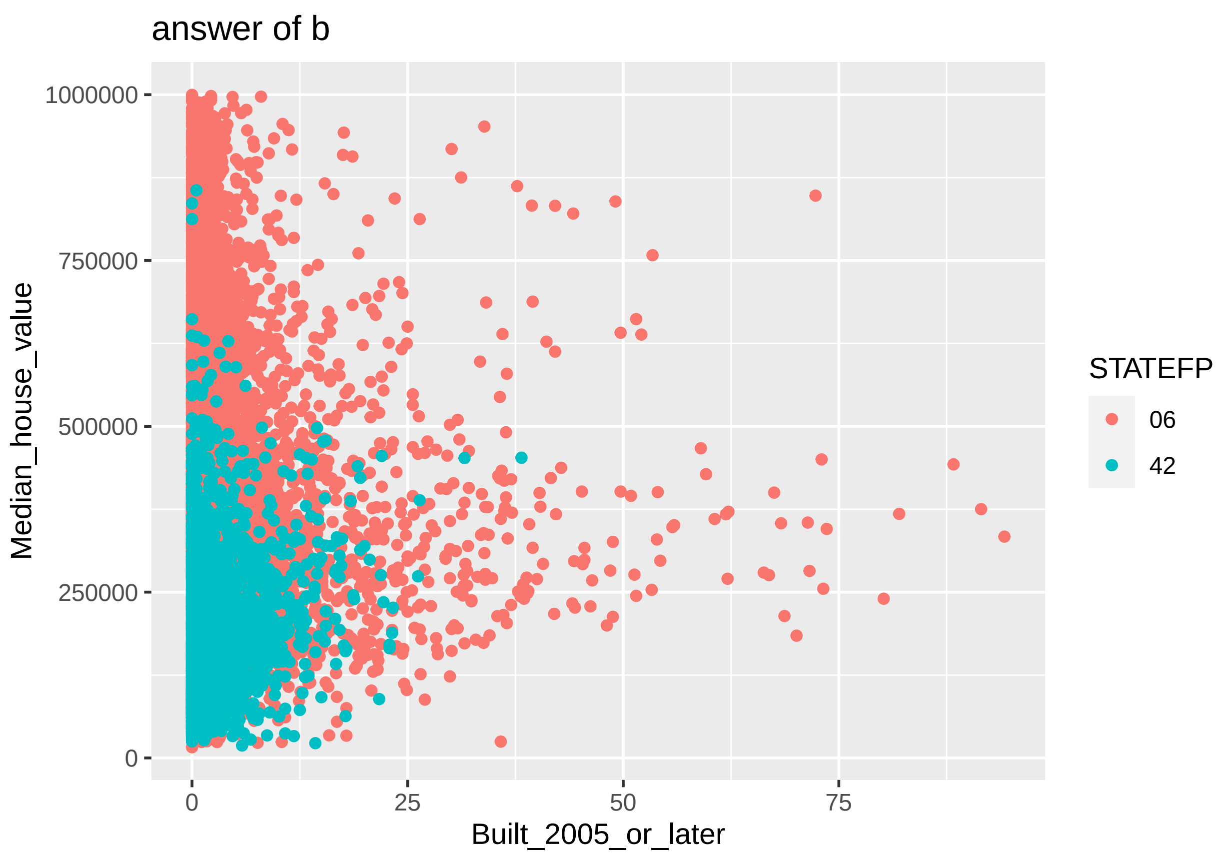
2. *This Very New House*

    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.
    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.

```r
ggplot(data = ca_pa_new) +
geom_point(aes(x =  Built_2005_or_later, y = Median_house_value))+
labs(title = "answer of a")
```




```
ggplot(data = ca_pa_new) +
geom_point(aes(x =  Built_2005_or_later, y = Median_house_value, color = STATEFP))+
labs(title = "answer of b")
```



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

```
ca_pa_new<-ca_pa_new %>%
mutate(vacancy_rate = Vacant_units/Total_units)
min(ca_pa_new$vacancy_rate)
```

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

```
max(ca_pa_new$vacancy_rate)
```

```
## [1] 0.965311
```

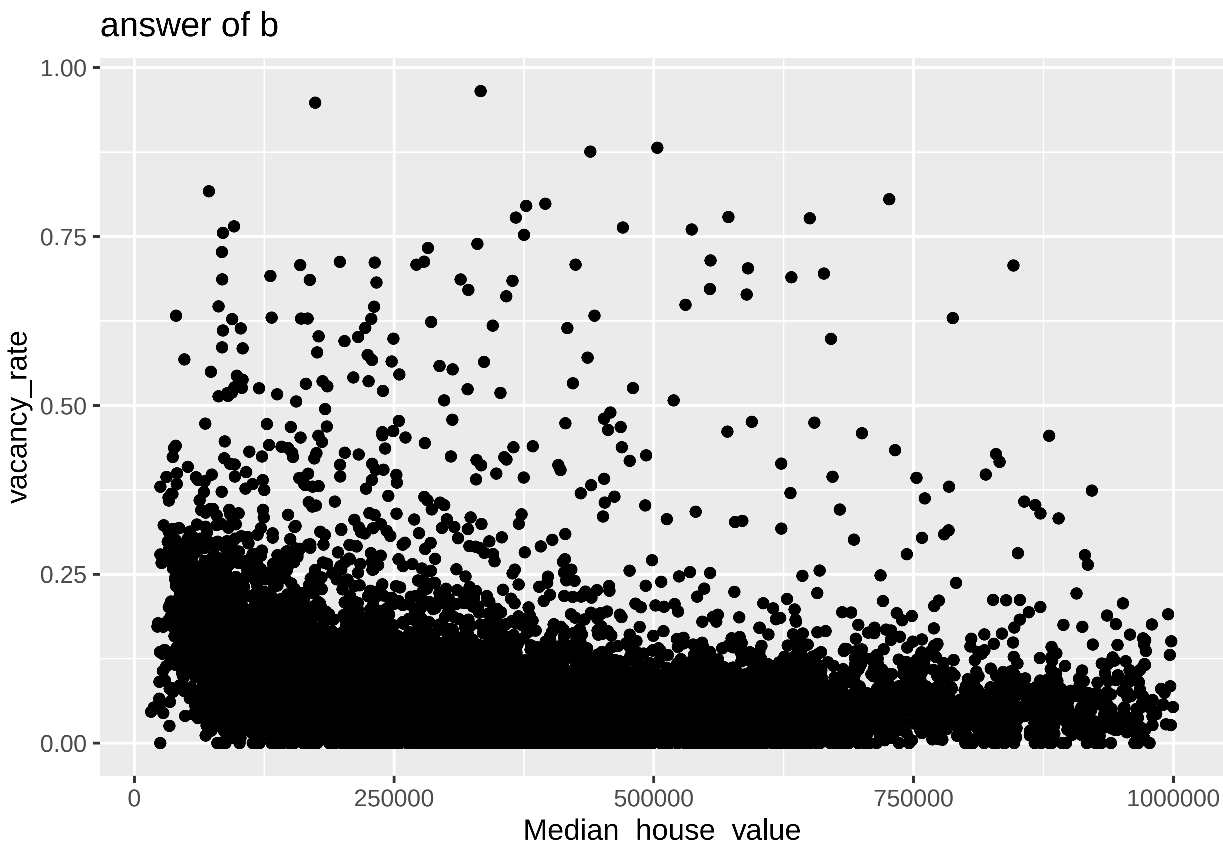
```
mean(ca_pa_new$vacancy_rate)
```

```
## [1] 0.08888789
```

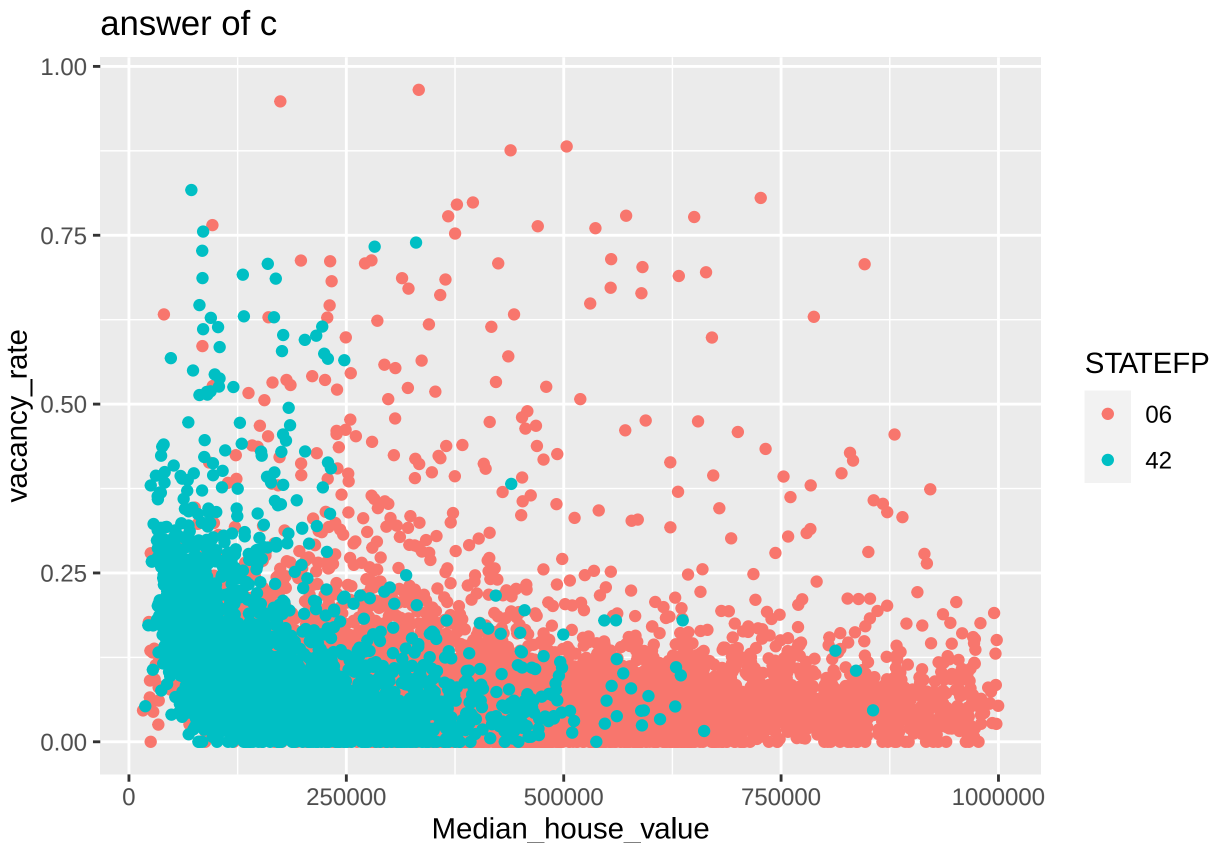
```r
median(ca_pa_new$vacancy_rate)
```

```
## [1] 0.06767283
```

```r
ggplot(data = ca_pa_new) +
geom_point(aes(x = Median_house_value, y = vacancy_rate))+
labs(title = "answer of b")
```




```r
ggplot(data = ca_pa_new) +
geom_point(aes(x = Median_house_value, y = vacancy_rate,color = STATEFP))+
labs(title = "answer of c")
```




```
#Dif: The Median_house_value of Pennsylvania is obviously lower than that of California.
```

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

```
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)
```

Explain of a: It returns the median of Median_house_value of Alameda County in California.

```
# answer of b
filter(ca_pa_new, STATEFP  == '06' & COUNTYFP == '001' )%>%
dplyr::select(Median_house_value) %>% unlist() %>% median()
```

```
## [1] 474050
```

```
# answer of c
filter(ca_pa_new, (STATEFP == '06'&COUNTYFP=='001')|(STATEFP == '06'&COUNTYFP=='085')
   |STATEFP == '42'&COUNTYFP=='003') %>% dplyr::select(Built_2005_or_later) %>%
  unlist() %>% mean()
```

```
## [1] 2.437344
```

```
# answer of d
x<-ca_pa_new%>% dplyr::select(Median_house_value)
y<-ca_pa_new%>% dplyr::select(Built_2005_or_later)
cor(x,y)
```

```
##                    Built_2005_or_later
## Median_house_value         -0.01893186
```

```
x<-ca_pa_new %>% filter(STATEFP == '06') %>% dplyr::select(Median_house_value)
y<-ca_pa_new %>% filter(STATEFP == '06') %>% dplyr::select(Built_2005_or_later)
cor(x,y)
```

```
##                    Built_2005_or_later
## Median_house_value          -0.1153604
```

```
x<-ca_pa_new %>% filter(STATEFP == '42') %>% dplyr::select(Median_house_value)
y<-ca_pa_new %>% filter(STATEFP == '42') %>% dplyr::select(Built_2005_or_later)
cor(x,y)
```

```
##                    Built_2005_or_later
## Median_house_value           0.2681654
```

```
x<-ca_pa_new %>% filter(STATEFP == '06'&COUNTYFP=='001') %>% dplyr::select(Median_house_value)
y<-ca_pa_new %>% filter(STATEFP == '06'&COUNTYFP=='001') %>% dplyr::select(Built_2005_or_later)
cor(x,y)
```

```
##                    Built_2005_or_later
## Median_house_value          0.01303543
```

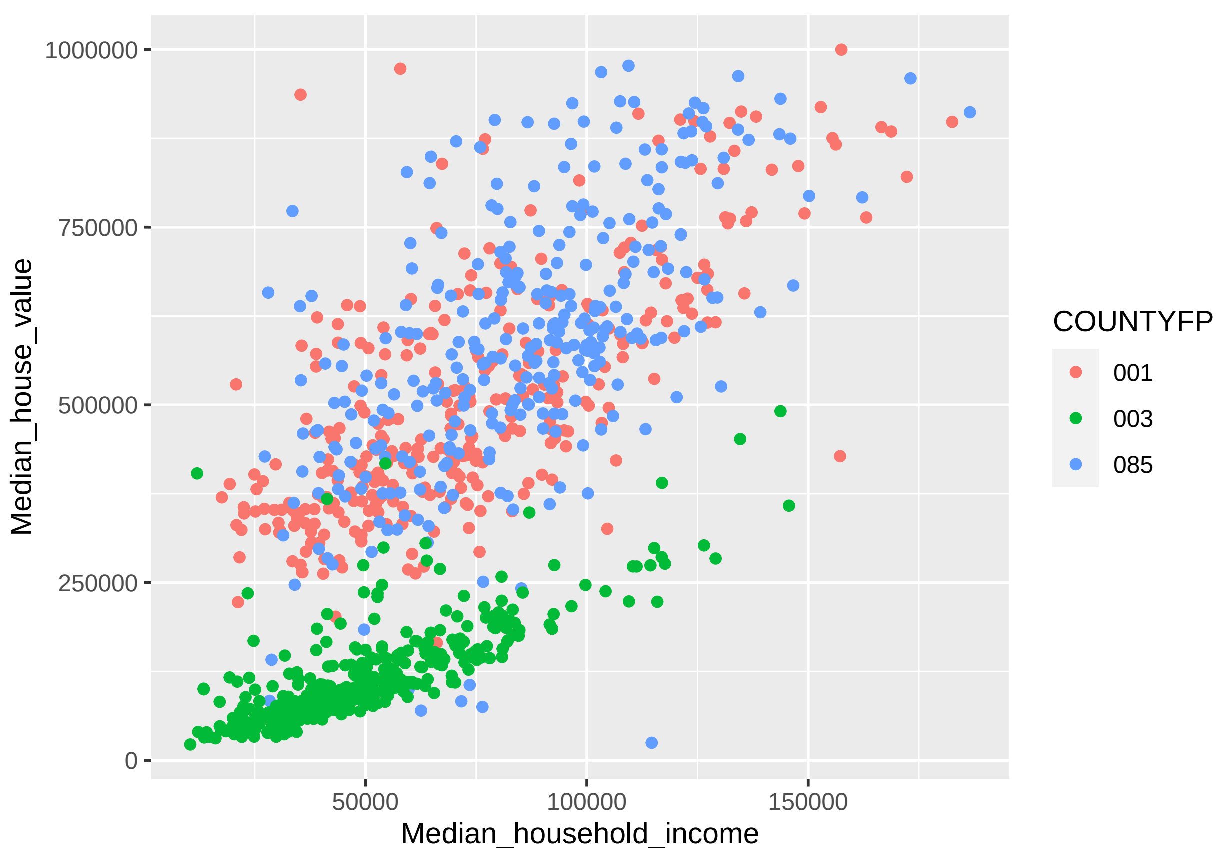
```
x<-ca_pa_new %>% filter(STATEFP == '06'&COUNTYFP=='085') %>% dplyr::select(Median_house_value)
y<-ca_pa_new %>% filter(STATEFP == '06'&COUNTYFP=='085') %>% dplyr::select(Built_2005_or_later)
cor(x,y)
```

```
##                     Built_2005_or_later
## Median_house_value           -0.1726203
```

```
x<-ca_pa_new %>% filter(STATEFP == '42'&COUNTYFP=='003') %>% dplyr::select(Median_house_value)
y<-ca_pa_new %>% filter(STATEFP == '42'&COUNTYFP=='003') %>% dplyr::select(Built_2005_or_later)
cor(x,y)
```

```
##                     Built_2005_or_later
## Median_house_value            0.1939652
```

```
# answer of e
ggplot(data = ca_pa_new%>% filter((STATEFP == '06'&COUNTYFP=='001')|
(STATEFP == '06'&COUNTYFP=='085')|STATEFP == '42'&COUNTYFP=='003')) +
geom_point(aes(x = Median_household_income, y = Median_house_value,color=COUNTYFP))
```



MB.Ch1.11. Run the following code:

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

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

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

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

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

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

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

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

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

Explain the output from the successive uses of table(). Explain: The factor levels are assumed to be ordered. The order varies from ("female", "male") to c("male", "female") , c("Male", "female","male") and c("Male", "female",NULL) , as results table() got output like this.

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.

```r
proportion <- function(x,y) {
s <- 0
for (v in x) {
  if(v>y){
    s<-s+1
  }
}
proportion<-s/length(x)
return(proportion)
}# answer of a
proportion(c(1:100),50)
```

```
## [1] 0.5
```

(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.

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

. . . .

```
x<-unstack(Rabbit,BPchange~Animal)
y<-unstack(Rabbit,Dose~Animal)
z<-unstack(Rabbit,Treatment~Animal)
data.frame("Treatment"=z[,1],"Dose"=y[,1],x)
```

```
##    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
```