Homework2

Zhang YunMengGe_3170105497

2020/7/9

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.

```
#load the data
calif_penn<-read_csv("data/calif_penn_2011.csv")
```

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

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   STATEFP = col_character(),
##   COUNTYFP = col_character(),
##   TRACTCE = col_character(),
##   GEO.display.label = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
ca_pa<-as.data.frame(calif_penn)
#head(ca_pa) #view the datafrme
```

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

```
dim(ca_pa)
```

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

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

```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
```

The function of `apply()` is `Retruns a vector or array or list of values obtained by applying a function to margins of an array or matrix`. This operation means judging the missing values of the row and column of the dataframe, and finding the sum of the missing values.

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.

```r
new_ca_pa<-na.omit(ca_pa)
#head(new_ca_pa,10)
```

e. How many rows did this eliminate?

```
n<-nrow(ca_pa)-nrow(new_ca_pa)
n
```

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

f. Are your answers in (c) and (e) compatible? Explain.

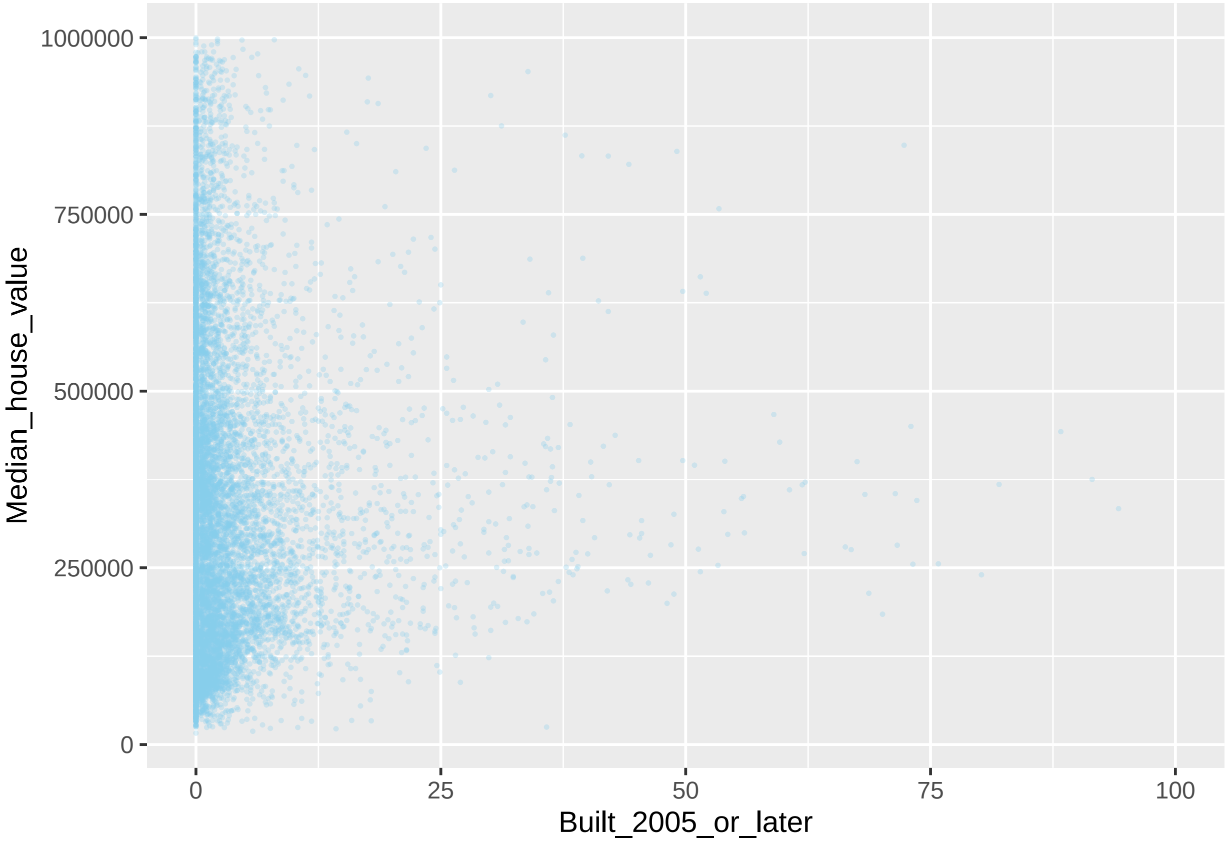
Compatible. `(c)` shows the number of missing values in each column, some of them have missing values at the same position, and the eliminated ones in `(e)` are those with missing values greater than or equal to 1, so they are compatible.

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.

```
ggplot(data=ca_pa)+
  geom_point(aes(x =Built_2005_or_later, y =Median_house_value),color="skyblue",alpha=0.3,size=0.3) +
  labs(x = "Built_2005_or_later",y = "Median_house_value")
```

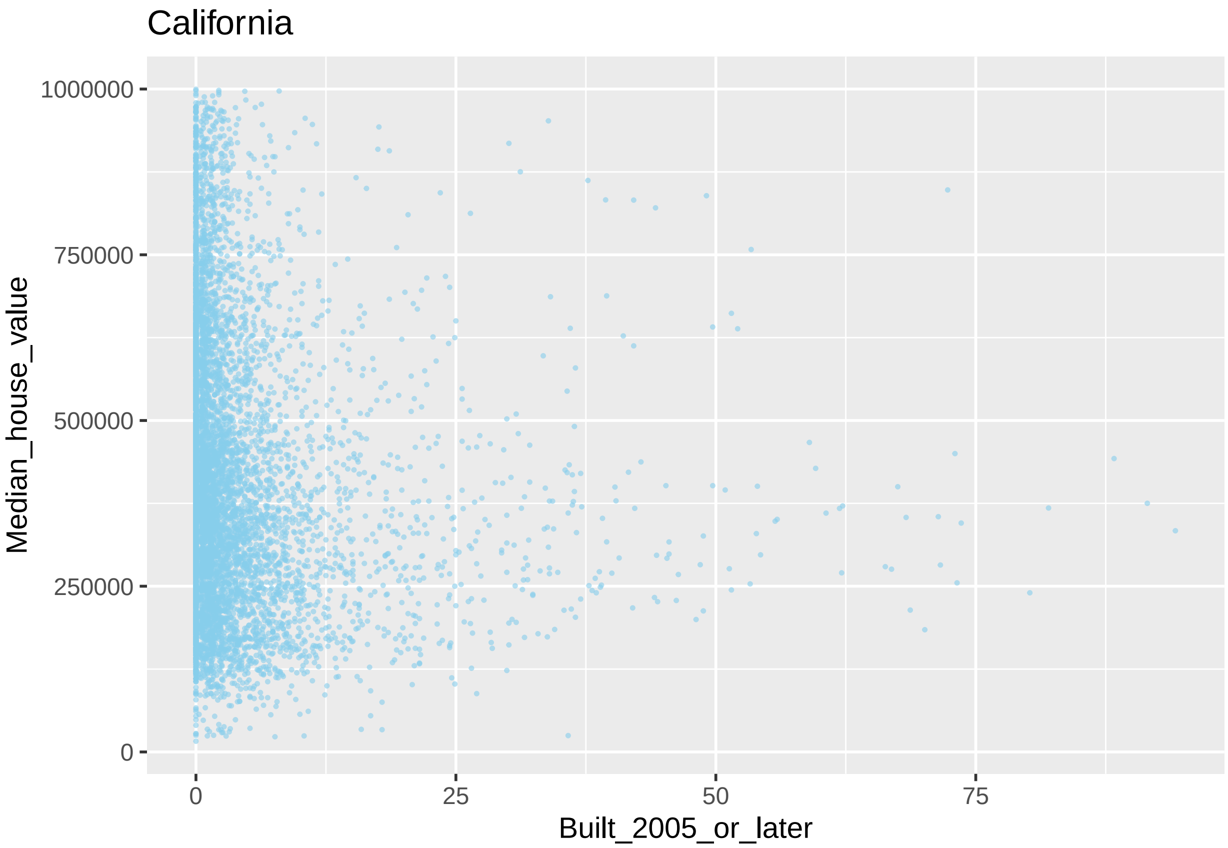
```
## Warning: Removed 599 rows containing missing values (geom_point).
```



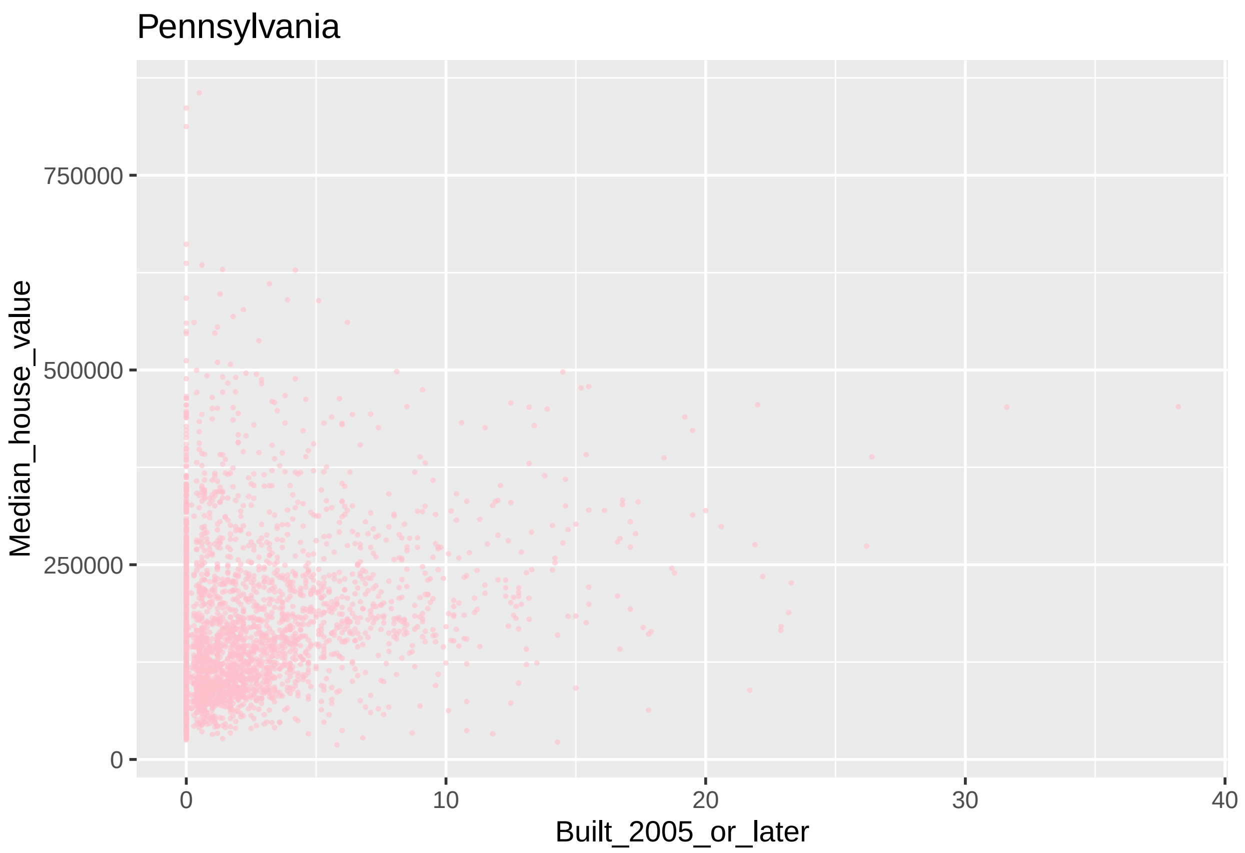
b. Make a new plot, or pair of plots, which breaks this out by state. Note that the state is recorded in the STATEFP variable, with California being state 6 and Pennsylvania state 42.

```
ca_pa_6<-filter(new_ca_pa,STATEFP=="06")
#ca_pa_6
ca_pa_42<-filter(new_ca_pa,STATEFP=="42")
#ca_pa_42
```

```
ggplot(data=ca_pa_6)+
  geom_point(aes(x =Built_2005_or_later, y =Median_house_value),color="skyblue",alpha=0.6,size=0.3) +
  labs(x = "Built_2005_or_later",y = "Median_house_value",title = "California")
```



```
ggplot(data=ca_pa_42)+
  geom_point(aes(x =Built_2005_or_later, y =Median_house_value),color="pink",alpha=0.6,size=0.3) +
  labs(x = "Built_2005_or_later",y = "Median_house_value",title = "Pennsylvania")
```



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.

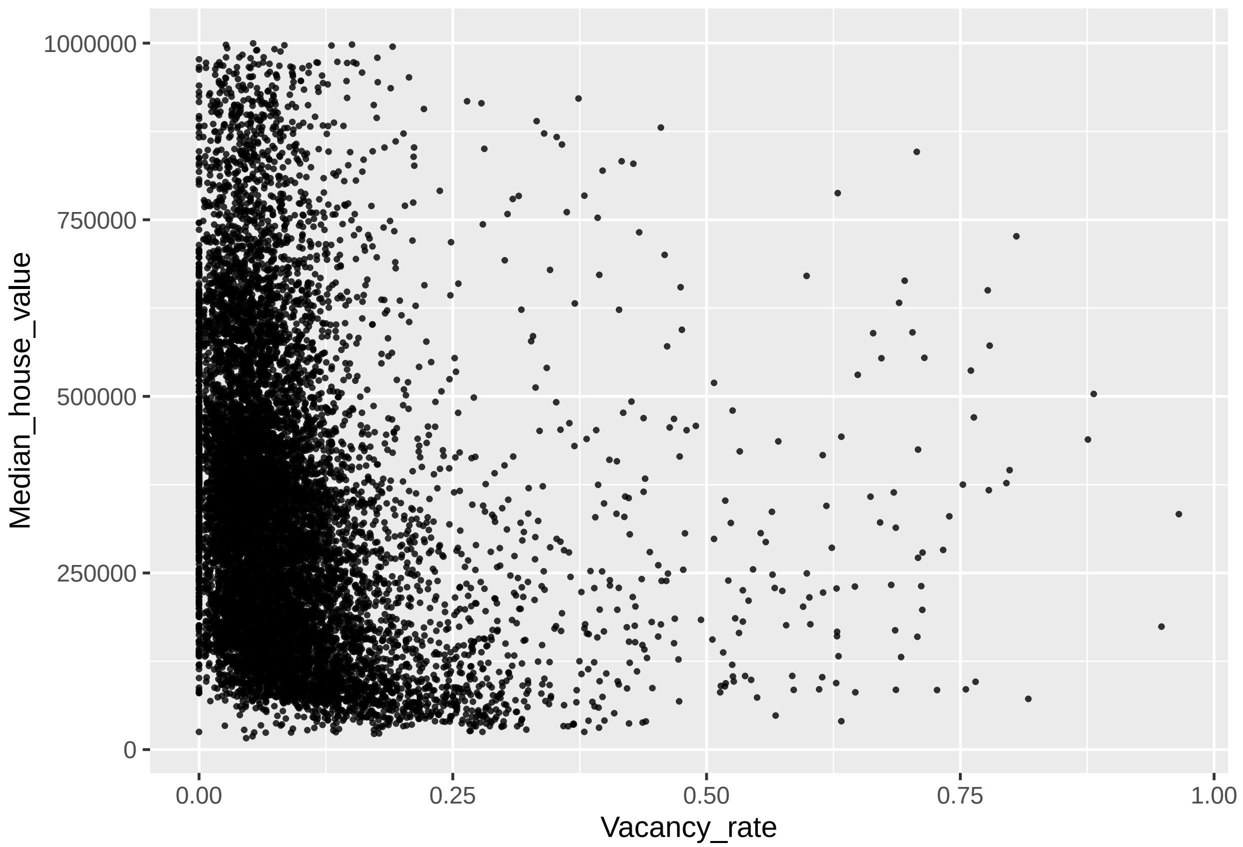
a. Add a new column to the dataframe which contains the vacancy rate. What are the minimum, maximum, mean, and median vacancy rates?

```
attach(new_ca_pa)
Vacancy_rate<-Vacant_units/Total_units
new_ca_pa<-cbind(new_ca_pa,Vacancy_rate)
summary(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.

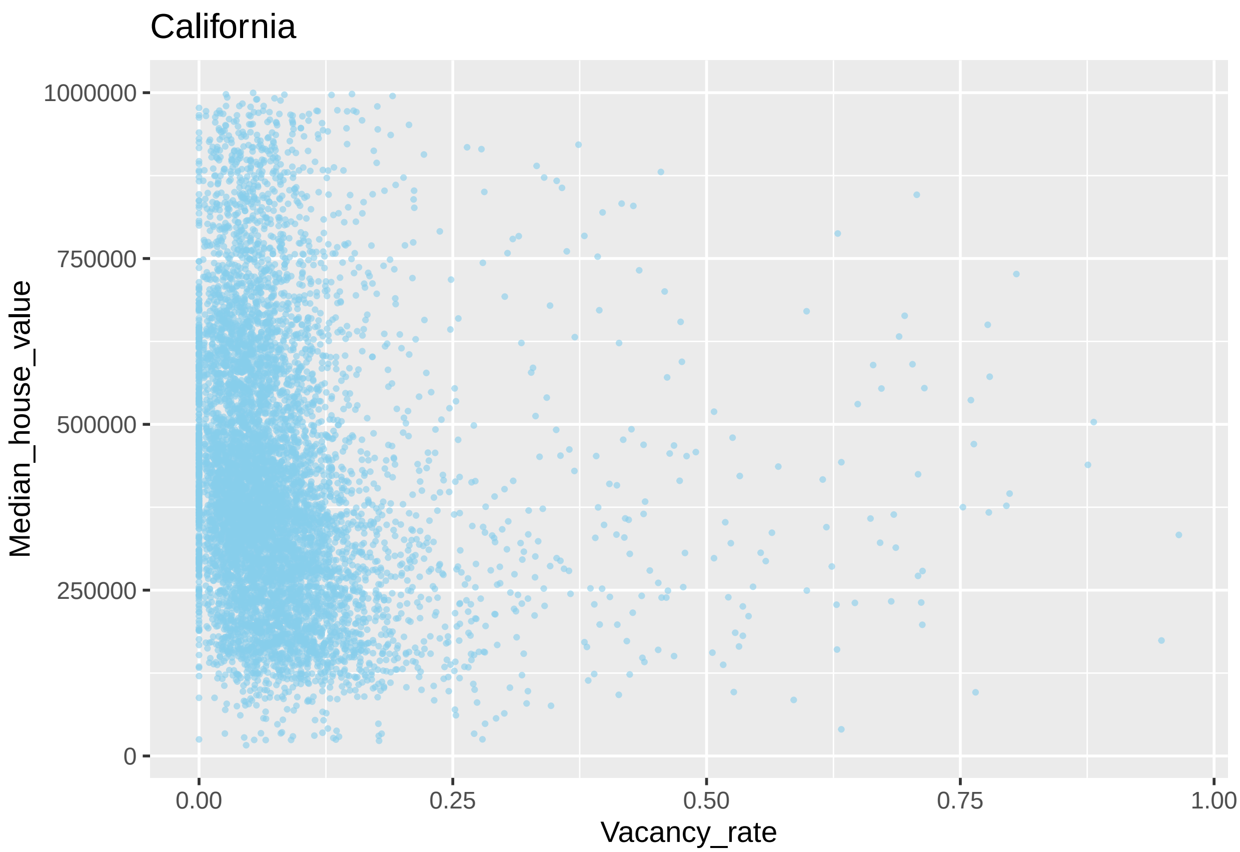
```
ggplot(data=new_ca_pa)+
  geom_point(aes(x =Vacancy_rate, y =Median_house_value),color="black",alpha=0.8,size=0.5) +
  labs(x = "Vacancy_rate",y = "Median_house_value")
```



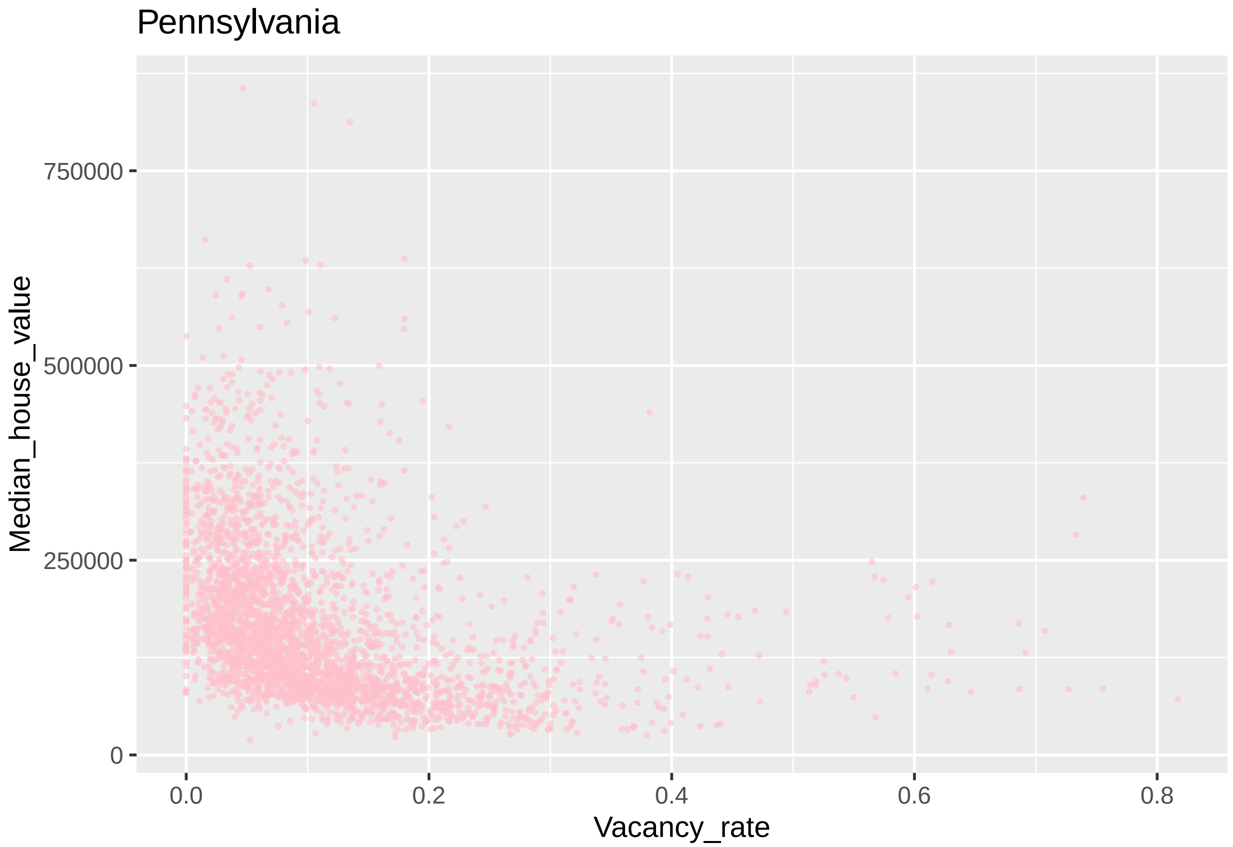
### c. Plot vacancy rate against median house value separately for California and for Pennsylvania. Is there a difference?

```
ca_pa_6va<-filter(new_ca_pa,STATEFP=="06")
#ca_pa_6va
ca_pa_42va<-filter(new_ca_pa,STATEFP=="42")
#ca_pa_42va
```

```
ggplot(data=ca_pa_6va)+
  geom_point(aes(x =Vacancy_rate, y =Median_house_value),color="skyblue",alpha=0.6,size=0.5) +
  labs(x = "Vacancy_rate",y = "Median_house_value",title = "California")
```



```
ggplot(data=ca_pa_42va)+
  geom_point(aes(x =Vacancy_rate, y =Median_house_value),color="pink",alpha=0.6,size=0.5) +
  labs(x = "Vacancy_rate",y = "Median_house_value",title = "Pennsylvania")
```



In California, the vacancy rate basically does not change with the change in the median house price, while in Pennsylvania, the vacancy rate shows a downward trend as the median house price increases.

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.

```
acca <- c()
for (tract in 1:nrow(ca_pa)) {
  if (ca_pa$STATEFP[tract] == 6) { #Filter out California
    if (ca_pa$COUNTYFP[tract] == 1) { #Filter out Alameda County in California
      acca <- c(acca, tract)  #Generate the number of rows where Alameda County is located
    }
  }
}
accamhv <- c()
for (tract in acca) {
  accamhv <- c(accamhv, ca_pa[tract,10]) #Generate the median house price in Alameda County Median_hous
}
median(accamhv) #Calculate the median of house prices in Alameda County
```

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.

```
median(new_ca_pa[which((STATEFP=="06")&(COUNTYFP=="001")),10])
```

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

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

```
ca_pa.1.1<-new_ca_pa[which((STATEFP=="06")&(COUNTYFP=="001")),]
ca_pa.1.2<-new_ca_pa[which((STATEFP=="06")&(COUNTYFP=="085")),]
ca_pa.2.1<-new_ca_pa[which((STATEFP=="42")&(COUNTYFP=="003")),]
mean1=mean(ca_pa.1.1$Built_2005_or_later/ca_pa.1.1$Total_units)
mean1
```

```
## [1] 0.002583202
```

```
mean2=mean(ca_pa.1.2$Built_2005_or_later/ca_pa.1.2$Total_units)
mean2
```

```
## [1] 0.001943991
```

```
mean3=mean(ca_pa.2.1$Built_2005_or_later/ca_pa.2.1$Total_units)
mean3
```

```
## [1] 0.001185988
```

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

```
cor(Median_house_value,Built_2005_or_later/Total_units)
```

```
## [1] -0.008761551
```

(ii) all of California

```
cor(ca_pa_6va$Median_house_value,ca_pa_6va$Built_2005_or_later/ca_pa_6va$Total_units)
```

```
## [1] -0.06240432
```

(iii) all of Pennsylvania

```
cor(ca_pa_42va$Median_house_value,ca_pa_42va$Built_2005_or_later/ca_pa_42va$Total_units)
```

```
## [1] 0.1980093
```

(iv) Alameda County

```
cor(ca_pa.1.1$Median_house_value,ca_pa.1.1$Built_2005_or_later/ca_pa.1.1$Total_units)
```

```
## [1] -0.009840453
```

(v) Santa Clara County

```
cor(ca_pa.1.2$Median_house_value,ca_pa.1.2$Built_2005_or_later/ca_pa.1.2$Total_units)
```

```
## [1] -0.1732909
```

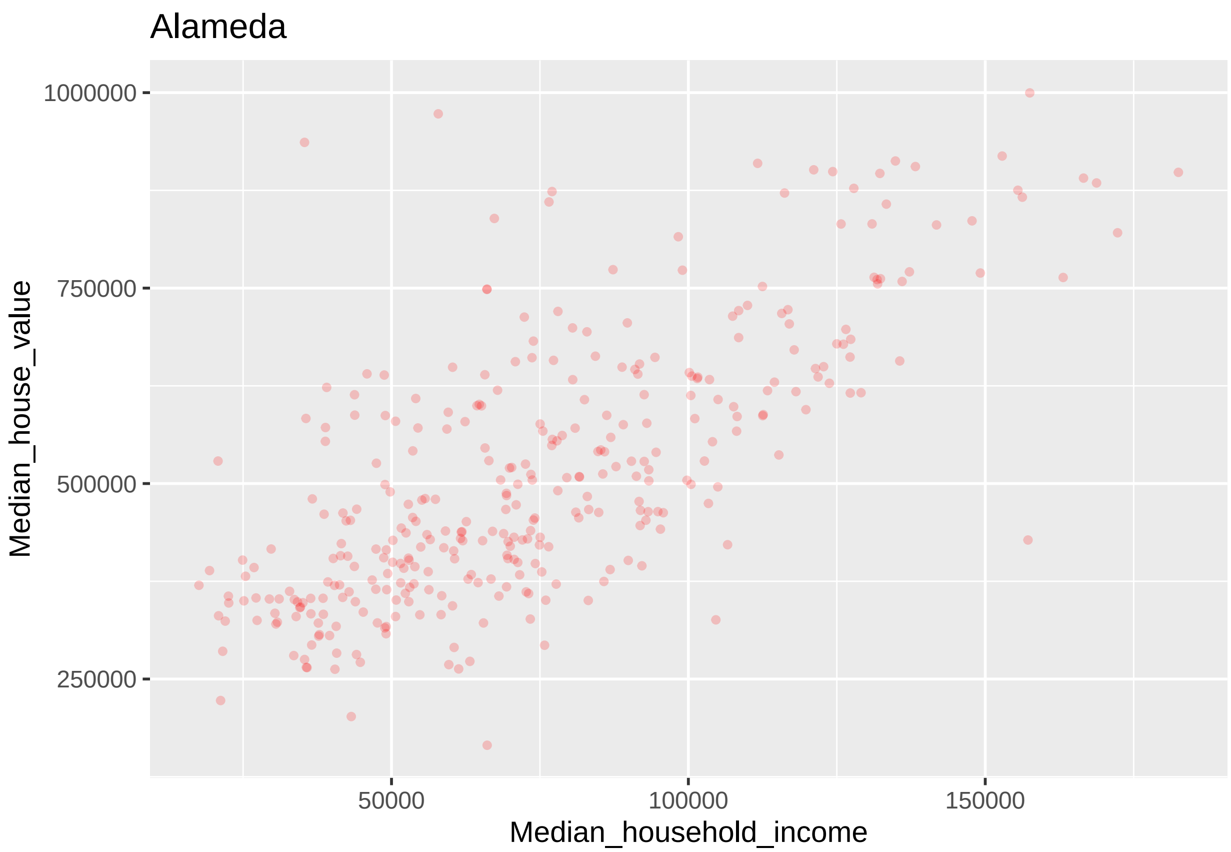
(vi) Allegheny County

```
cor(ca_pa.2.1$Median_house_value,ca_pa.2.1$Built_2005_or_later/ca_pa.2.1$Total_units)
```

```
## [1] 0.04878986
```

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.)

```
ggplot(data=ca_pa.1.1)+
  geom_point(aes(x =Median_household_income, y =Median_house_value),color="red",alpha=0.2,size=1) +
  labs(x = "Median_household_income",y = "Median_house_value",title = "Alameda")
```



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()` uses the cross-classifying factors to build a contingency table of the counts at each combination of factor levels.

The `Male` factor has no corresponding data, the frequency is zero, and the `male` factor that is not named is counted in `NA`.

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

```
proportion<-function(vector,x){
  n=length(vector)
  p=0
  for (i in 1:n) {
    if(vector[i]<=x)
      {
        p=p+1
      }
    else{
    p=p
    }
  }
  1-p/n
}
```

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

```
a=seq(1:100)
proportion(a,60)
```

```
## [1] 0.4
```

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

….

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
RabbitData<-as.data.frame(Rabbit)
#RabbitData
```

```
RabbitDose<-unstack(RabbitData,Dose~Animal)[,1]
RabbitDose
```

```
##  [1]   6.25  12.50  25.00  50.00 100.00 200.00   6.25  12.50  25.00  50.00
## [11] 100.00 200.00
```

```
RabbitTreatment<-unstack(RabbitData,Treatment~Animal)[,1]
RabbitTreatment
```

```
##  [1] "Control" "Control" "Control" "Control" "Control" "Control" "MDL"
##  [8] "MDL"     "MDL"     "MDL"     "MDL"     "MDL"
```

```
RabbitBPchange<-unstack(RabbitData,BPchange~Animal)
RabbitBPchange
```

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

```
Rabbitform<-data.frame(RabbitTreatment,RabbitDose,RabbitBPchange)
Rabbitform
```

```
##    RabbitTreatment RabbitDose    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
```