

Homework 2

Shen Dingtao 3170104764

1. Loading and cleaning

(a)

```
ca_pa <- read.csv("data/calif_penn_2011.csv",header=T)
```

(b)

```
dim(ca_pa)
```

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

So the data frame has 11275 rows and 34 columns.

(c)

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

```
##              X              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
```

This command is used to count the number of NA (not a number) values in each column of the data frame.

- The `apply()` function loops through all the elements of the matrix `ca_pa` and applies the `is.na()` function which returns TRUE if the element is not a number and FALSE otherwise.
- The resulting matrix of TRUE and FALSE values is then given as input to `colSums()` function, which counts the number of TRUE values in each column.

(d)

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

(e)

```
nrow(ca_pa)-nrow(new_ca_pa)
```

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

The result shows that the `na.omit()` command eliminate 670 rows.

(f) My answer in (c) and (e) are compatible. We can run the following command to verify this:

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

```
##           X           GEO.id2
##           0           0
##           STATEFP       COUNTYFP
##           0           0
##           TRACTCE       POPULATION
##           0           0
##           LATITUDE      LONGITUDE
##           0           0
## GEO.display.label      Median_house_value
##           0           0
##           Total_units   Vacant_units
##           0           0
##           Median_rooms  Mean_household_size_owners
##           0           0
## Mean_household_size_renters      Built_2005_or_later
##           0           0
##           Built_2000_to_2004      Built_1990s
##           0           0
```

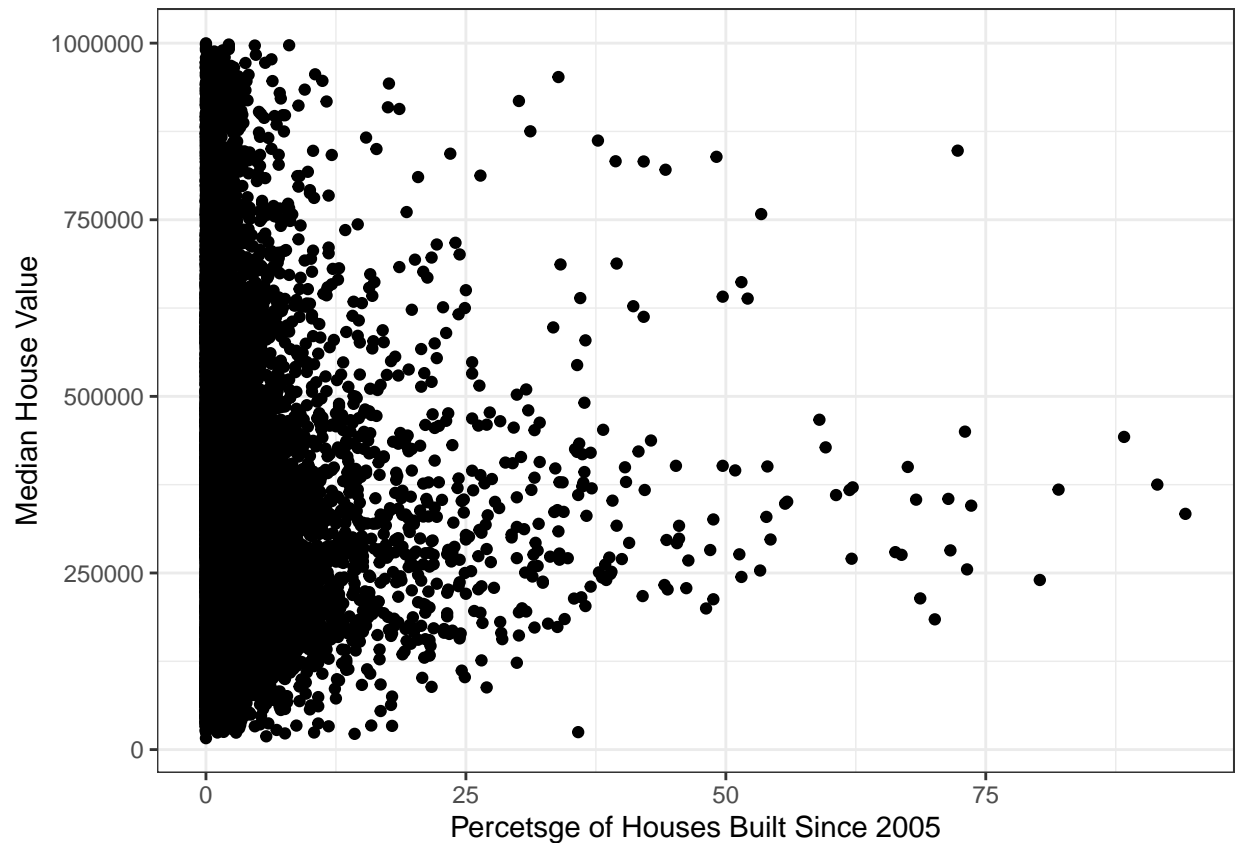
##	Built_1980s	Built_1970s
##	0	0
##	Built_1960s	Built_1950s
##	0	0
##	Built_1940s	Built_1939_or_earlier
##	0	0
##	Bedrooms_0	Bedrooms_1
##	0	0
##	Bedrooms_2	Bedrooms_3
##	0	0
##	Bedrooms_4	Bedrooms_5_or_more
##	0	0
##	Owners	Renters
##	0	0
##	Median_household_income	Mean_household_income
##	0	0

The values of all columns are all 0, which means that we have successfully purged `ca_pa` from any row containing NA value.

2. *This Very New House*

(a)

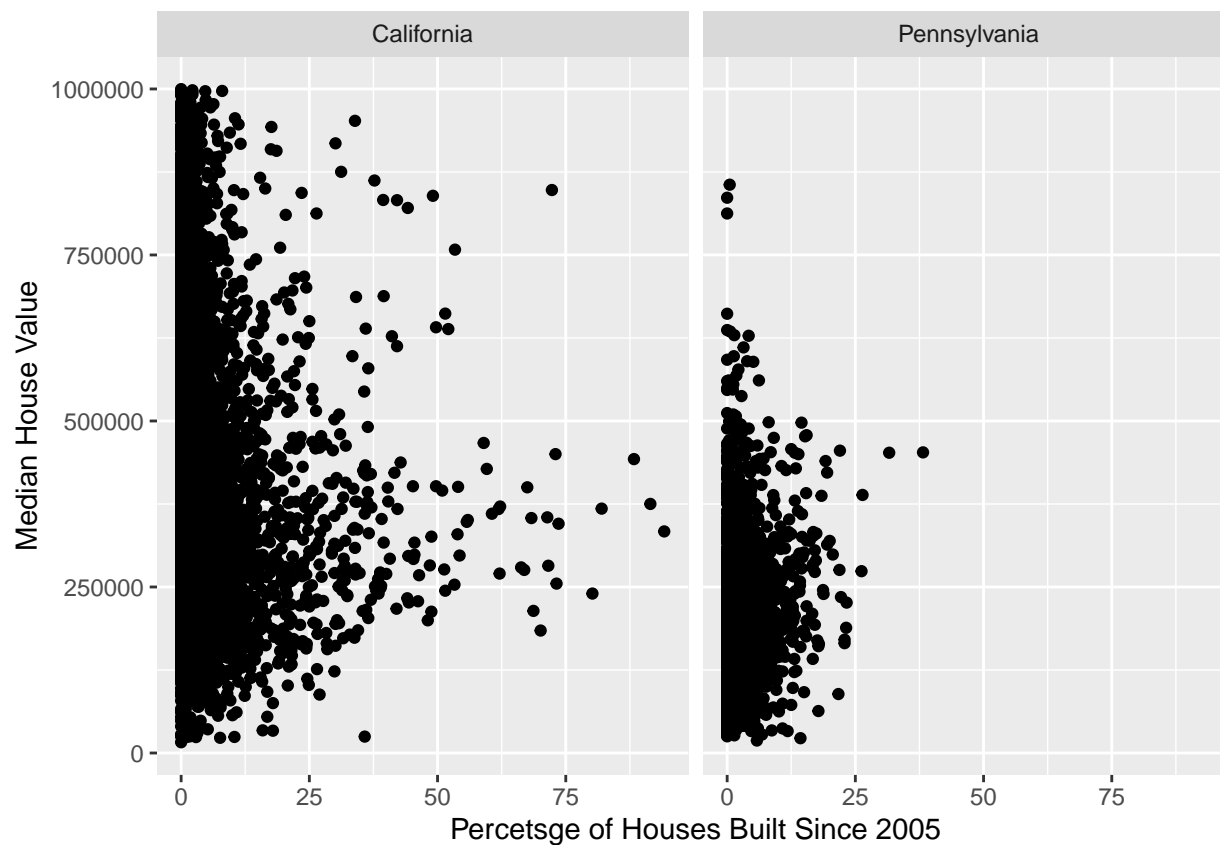
```
ggplot(data = new_ca_pa) +
  geom_point(aes(x = Built_2005_or_later,
    y = Median_house_value), na.rm = TRUE) +
  labs(x = "Percentage of Houses Built Since 2005",
    y = "Median House Value") +
  theme_bw() +
  theme(legend.title=element_blank())
```



(b) With following commands, we make a pair of plots that breaks this out by state, which shows the median house prices against Built_2005_or_later in California(state 6) and Pennsylvania(state 42) respectively.

```
d<-data.frame(STATEFP=c(6,42),state=c("California","Pennsylvania"))
ca_pa2<-left_join(new_ca_pa,d,by="STATEFP")
```

```
ggplot(data = ca_pa2) +
  geom_point(aes(x = Built_2005_or_later,
    y = Median_house_value),na.rm = TRUE) +
  labs(x = "Percetsge of Houses Built Since 2005",
    y = "Median House Value") +facet_wrap(~ state)
```



3. *Nobody Home*

(a) Add a new column to the dataframe which contains the vacancy rate.

```
ca_pa3 <- ca_pa2 %>%
  mutate(Vacant_Rate=Vacant_units/Total_units)
```

Minimum:

```
min(ca_pa3$Vacant_Rate,na.rm = TRUE)
```

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

Maximum:

```
max(ca_pa3$Vacant_Rate,na.rm = TRUE)
```

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

Mean:

```
mean(ca_pa3$Vacant_Rate,na.rm = TRUE)
```

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

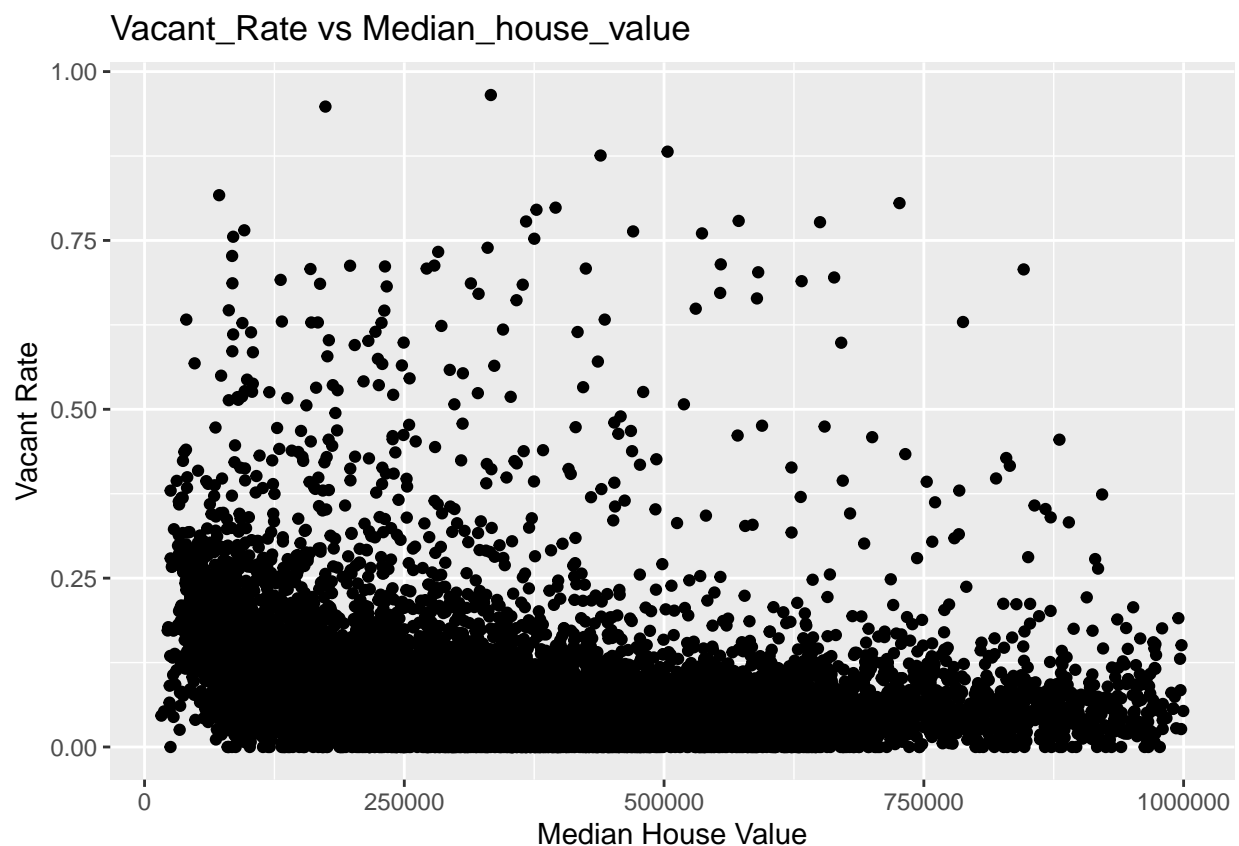
Median:

```
median(ca_pa3$Vacant_Rate,na.rm = TRUE)
```

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

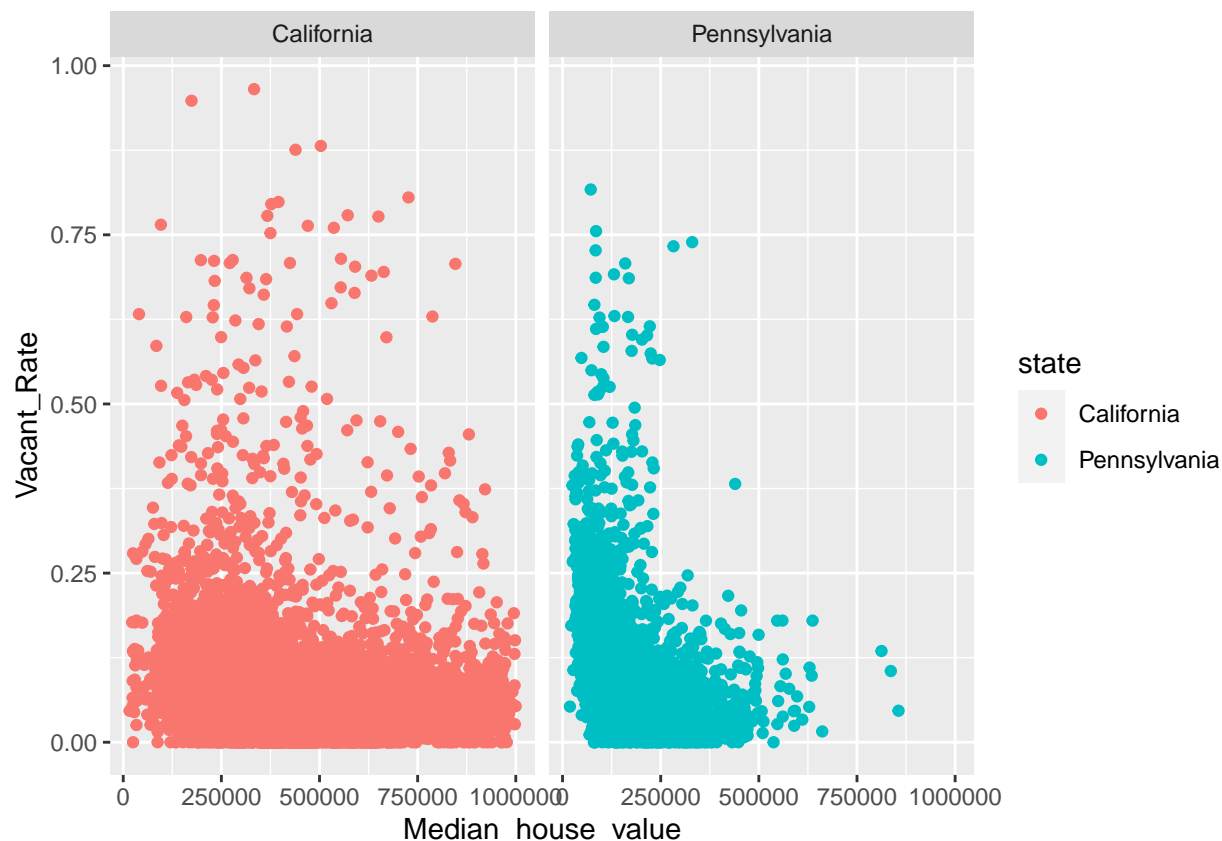
(b) Plot the vacancy rate against median house value:

```
ggplot(data = ca_pa3) +  
  geom_point(aes(x = Median_house_value,  
    y = Vacant_Rate),na.rm = TRUE) +  
  labs(x = "Median House Value",  
    y = "Vacant Rate",title = "Vacant_Rate vs Median_house_value")
```



(c) Plot vacancy rate against median house value separately for California and for Pennsylvania:

```
ggplot(data = ca_pa3) +  
  geom_point(aes(x = Median_house_value,  
    y = Vacant_Rate,color=state),na.rm = TRUE) +  
  facet_wrap(~ state)
```



4.

(a) The block of code is supposed to calculate the median house value in Alameda County (country 1 in California). It firstly selects the rows of California, whose STATEFP=6, and then selects the rows of Alameda, whose COUNTYFP=1 among the selected rows. Finally select median value of these rows' Median house value to get the median house value in Alameda.

(b) We can obtain the same result as the block of code with following command:

```
ca_pa3 %>% filter(STATEFP==6,COUNTYFP==1) %>% {median(.$Median_house_value,na.rm = TRUE)}
```

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

(c) We can obtain the average percentages of housing built since 2005 for Alameda, Santa Clara and Alameda Counties with following commands: (i) For Alameda:

```
(Alameda_avg <- ca_pa3 %>% filter(STATEFP==6&COUNTYFP==1 )) %>% {mean(.$Built_2005_or_later,na.rm = TRUE)}
```

```
## [1] 2.820468
```

(ii) For Santa Clara:

```
(Alameda_avg <- ca_pa3 %>% filter(STATEFP==6&COUNTYFP==85 )) %>% {mean(.$Built_2005_or_later,na.rm = TRUE)}
```

```
## [1] 3.200319
```

(ii) For Allegheny:

```
(Alameda_avg <- ca_pa3 %>% filter(STATEFP==42&COUNTYFP==3 )) %>% {mean(.$Built_2005_or_later,na.rm = TR  
## [1] 1.474219
```

(d) the correlation between median house value and the percent of housing built since 2005 (i) In the whole data:

```
cor(ca_pa3$Median_house_value,ca_pa3$Built_2005_or_later)
```

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

(ii) In all of California

```
ca_pa3 %>% filter(STATEFP==6) %>%  
{cor(.$Median_house_value,.$Built_2005_or_later)}
```

```
## [1] -0.1153604
```

(iii) In all of Pennsylvania

```
ca_pa3 %>% filter(STATEFP==42) %>%  
{cor(.$Median_house_value,.$Built_2005_or_later)}
```

```
## [1] 0.2681654
```

(iv) In Alameda County

```
ca_pa3 %>% filter(STATEFP==6,COUNTYFP==1) %>%  
{cor(.$Median_house_value,.$Built_2005_or_later)}
```

```
## [1] 0.01303543
```

(v) In Santa Clara County

```
ca_pa3 %>% filter(STATEFP==6,COUNTYFP==85) %>%  
{cor(.$Median_house_value,.$Built_2005_or_later)}
```

```
## [1] -0.1726203
```

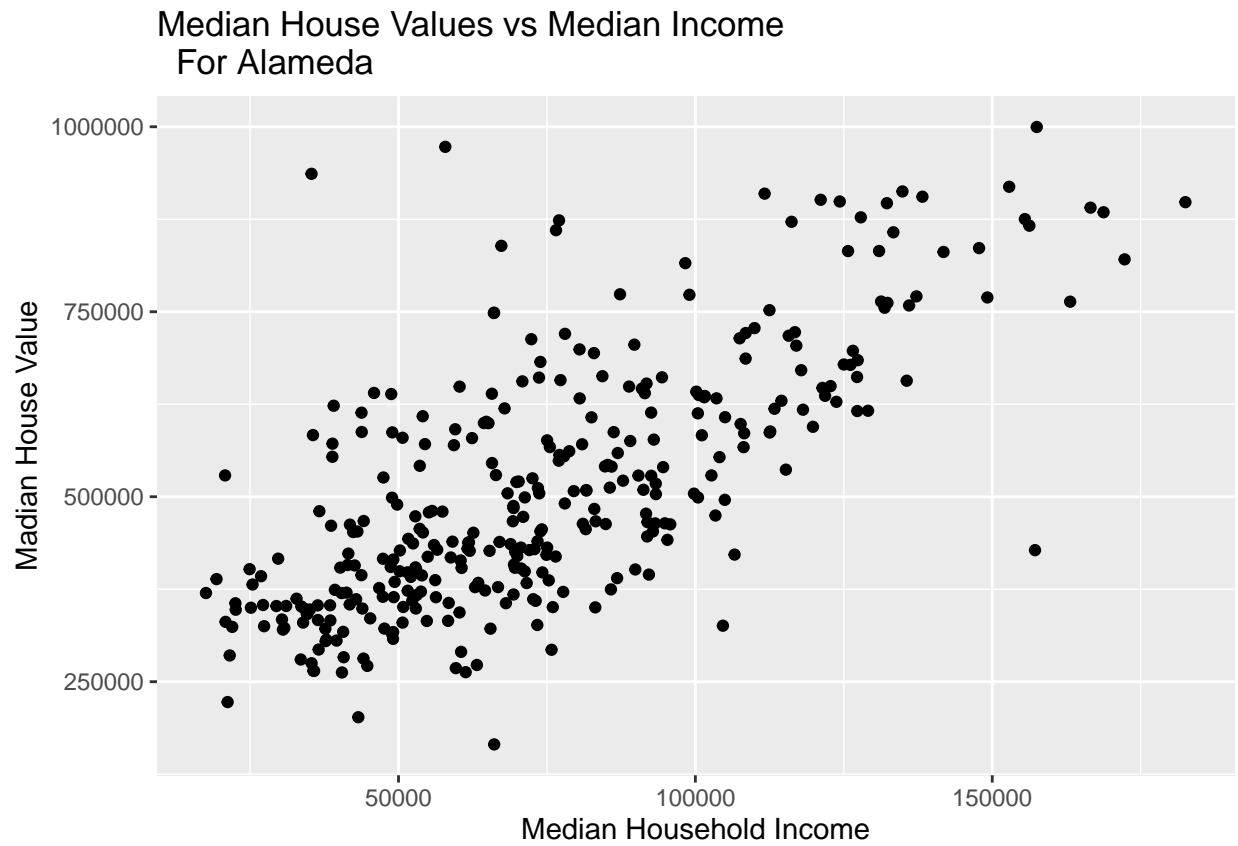
(vi) In Allegheny Count

```
ca_pa3 %>% filter(STATEFP==42,COUNTYFP==3) %>%  
{cor(.$Median_house_value,.$Built_2005_or_later)}
```

```
## [1] 0.1939652
```

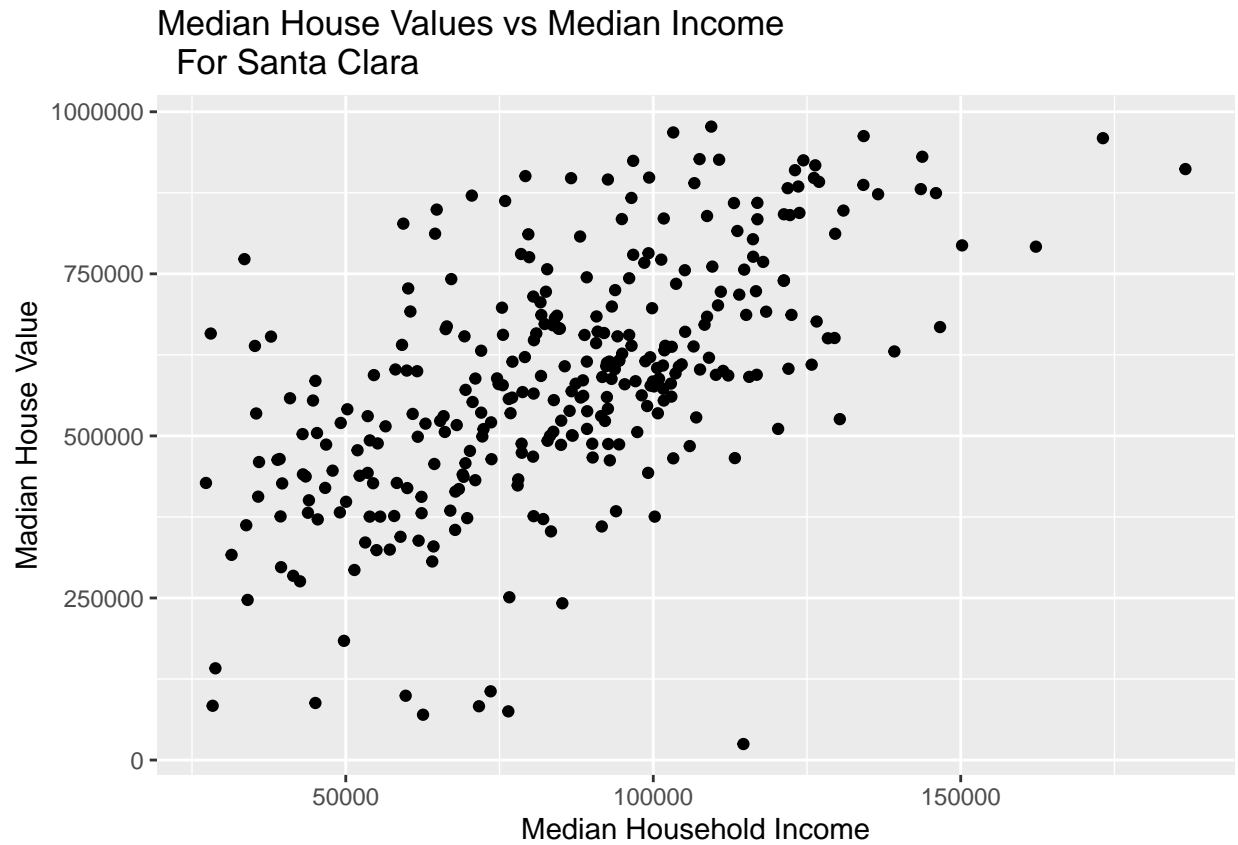
(e) Median house values against median income. (i) For Alameda:


```
ca_pa3 %>% filter(STATEFP==6,COUNTYFP==1) %>%
  ggplot() +
  geom_point(aes(x = Median_household_income,
    y = Median_house_value),na.rm = TRUE) +
  labs(x = "Median Household Income",
    y = "Madian House Value",title = "Median House Values vs Median Income
    For Alameda")
```



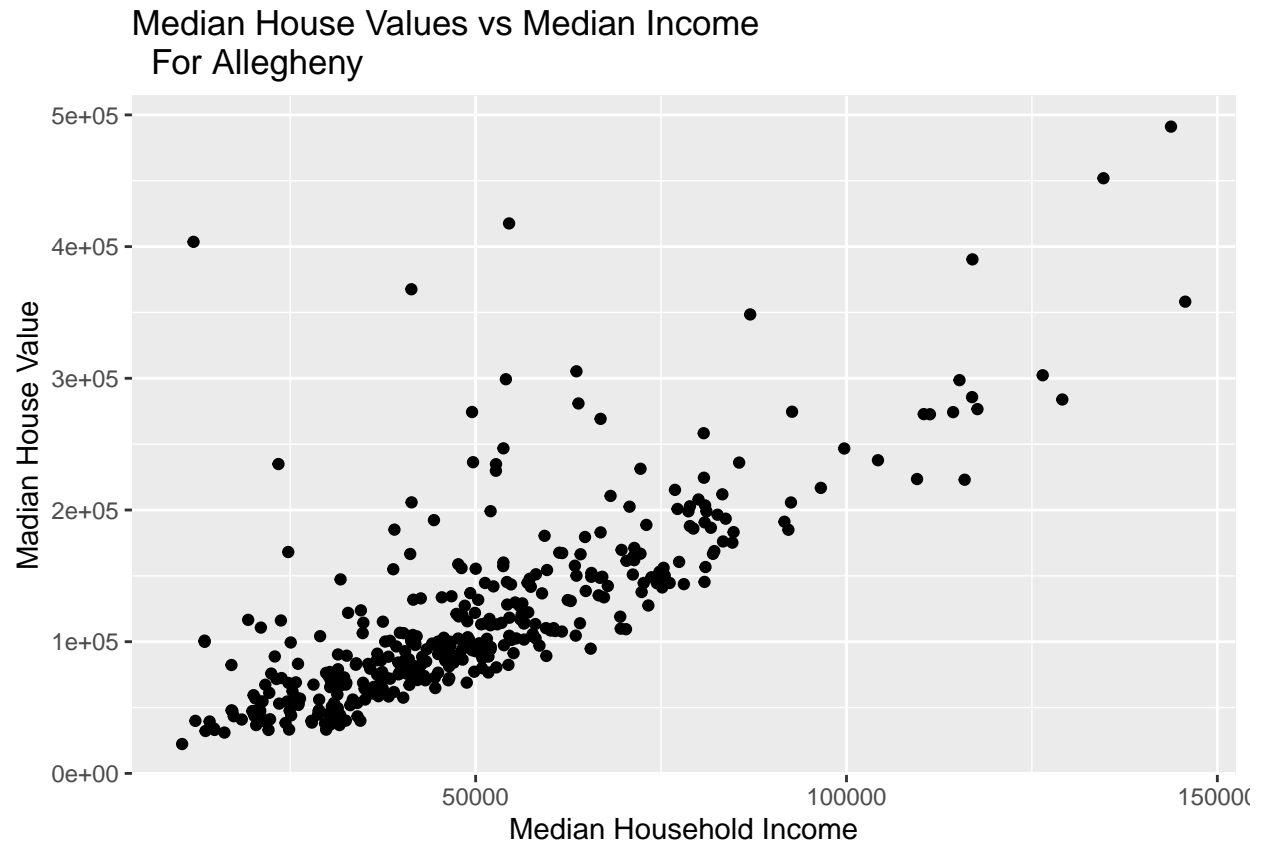
(ii) For Santa Clara:

```
ca_pa3 %>% filter(STATEFP==6,COUNTYFP==85) %>%
  ggplot() +
  geom_point(aes(x = Median_household_income,
    y = Median_house_value),na.rm = TRUE) +
  labs(x = "Median Household Income",
    y = "Madian House Value",title = "Median House Values vs Median Income
    For Santa Clara")
```



(iii) For Allegheny:

```
ca_pa3 %>% filter(STATEFP==42,COUNTYFP==3) %>%
  ggplot() +
  geom_point(aes(x = Median_household_income,
    y = Median_house_value),na.rm = TRUE) +
  labs(x = "Median Household Income",
    y = "Median House Value",title = "Median House Values vs Median Income
    For Allegheny")
```



MB.Ch1.11. Run the following code:

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

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

Explain: table() gives the frequency of each element “female” and “male”.

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

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

Explain: the output gives the frequency in levels, that is “male” and “female”.

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

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

Explain: the output gives the frequency of levels, that is “Male” and “female”. While there is no “Male” in gender, so the value of Male is 0.

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

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

Explain: The argument “exclude=NULL” require to report NA’s, so the result gives =92 with respond to “male” that is not in the levels.

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

MB.Ch1.12.

```
k<-0
Func <- function(x,cutoff_value){
  for(i in x){
    if (i>cutoff_value){
      k<-k+1
    }
  }
  prop <- k/length(x)
  return(prop)
}
```

- (a) For the sequence of numbers 1, 2, . . . , 100, we set the value cutoff 35,67.3 and 89, then the expected proportion result is 0.65, 0.33 and 0.11 correspondingly. And following codes check this.

```
x<-1:100
vc<-35
Func(x,vc)
```

```
## [1] 0.65
```

```
vc<-67.3
Func(x,vc)
```

```
## [1] 0.33
```

```
vc<-89
Func(x,vc)
```

```
## [1] 0.11
```

MB.Ch1.18. Using following commands, we can convert Rabbit to the required form.

```

Dose <- unstack(Rabbit, Dose ~ Animal)[,1]
Treatment <- unstack(Rabbit, Treatment ~ Animal)[,1]
BPchange <- unstack(Rabbit, BPchange ~ Animal)
Rabbit.df <- data.frame(Treatment, Dose, BPchange)
Rabbit.df

```

##	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