Homework2

Yiheng

7/14/2020

1.Loading and Cleaning

a.

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

b.

```
nrow(ca_pa)
```

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

```
ncol(ca_pa)
```

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

⇒ The dataframe 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
```

The command shows the number of missing values each column has.

d.

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

e.

```r
11275 - nrow(ca_pa)
```

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

⇒ It eliminated 670 rows.

f. They're compatible, since there might be more than one missing values in one row, and the R script below returns `TRUE`:

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

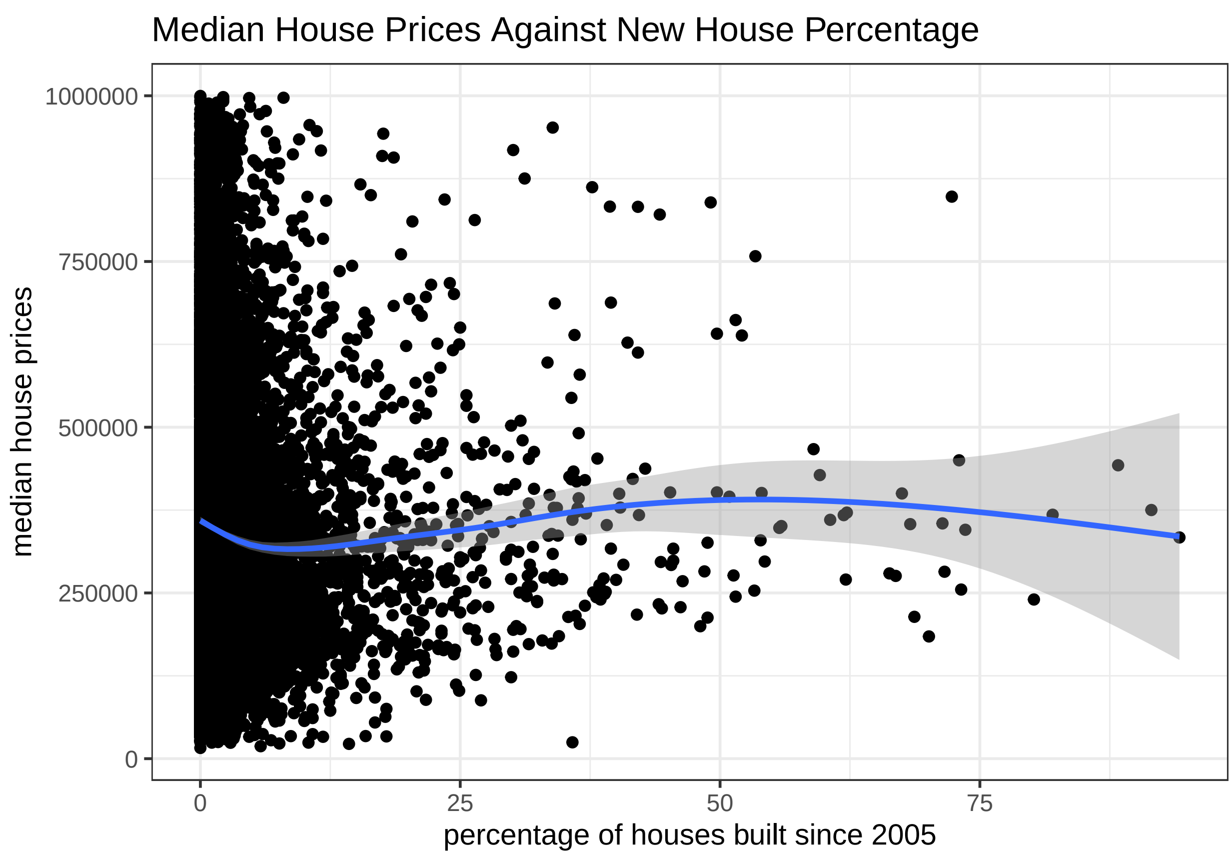
```
## [1] TRUE
```

2. This Very New House

a.

```r
ggplot(data = ca_pa) +
  geom_point(mapping = aes(x = Built_2005_or_later, y = Median_house_value))+
  geom_smooth(mapping = aes(x = Built_2005_or_later, y = Median_house_value))+
  labs(x = "percentage of houses built since 2005",
       y = "median house prices",
       title = "Median House Prices Against New House Percentage") +
  theme_bw()
```

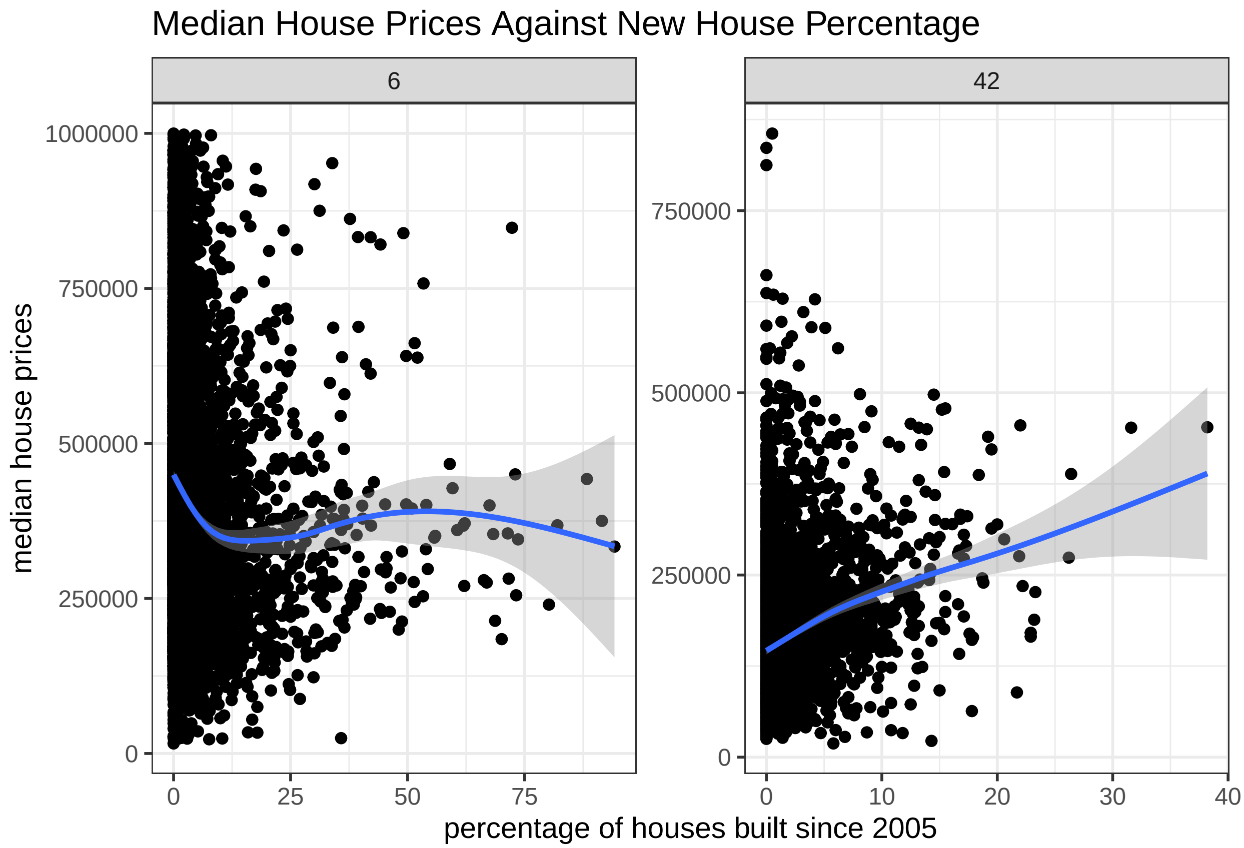
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



b.

```
ggplot(data = ca_pa) +
  geom_point(mapping = aes(x = Built_2005_or_later, y = Median_house_value))+
  geom_smooth(mapping = aes(x = Built_2005_or_later, y = Median_house_value))+
  labs(x = "percentage of houses built since 2005",
       y = "median house prices",
       title = "Median House Prices Against New House Percentage") +
  facet_wrap(~ STATEFP , scales = "free") +
  theme_bw()
```

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



3. Nobody Home

a.

```r
ca_pa <- data.frame(ca_pa, vacancy_rate = ca_pa$Vacant_units/ca_pa$Total_units)
```

```r
min(ca_pa$vacancy_rate)
```

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

```r
max(ca_pa$vacancy_rate)
```

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

```r
mean(ca_pa$vacancy_rate)
```

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

```r
median(ca_pa$vacancy_rate)
```

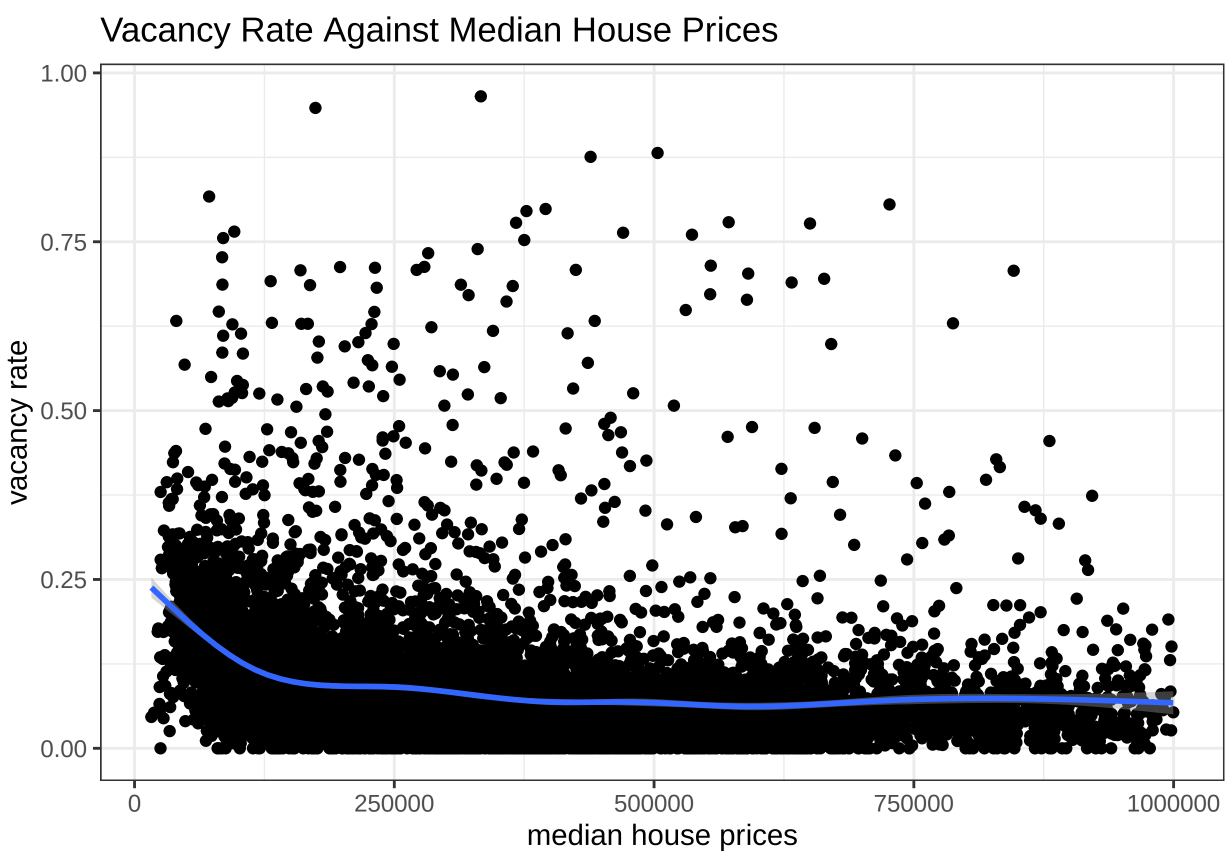
```
## [1] 0.06767283
```

⇒ The minimum vacancy rate is 0;The maximum vacancy rate is 0.965311;
The mean vacancy rate is 0.08888789;The median vacancy rate is 0.06767283.

b.

```
ggplot(data = ca_pa) +
  geom_point(mapping = aes(x = Median_house_value, y = vacancy_rate))+
  geom_smooth(mapping = aes(x = Median_house_value, y = vacancy_rate))+
  labs(x = "median house prices",
       y = "vacancy rate",
       title = "Vacancy Rate Against Median House Prices") +
  theme_bw()
```

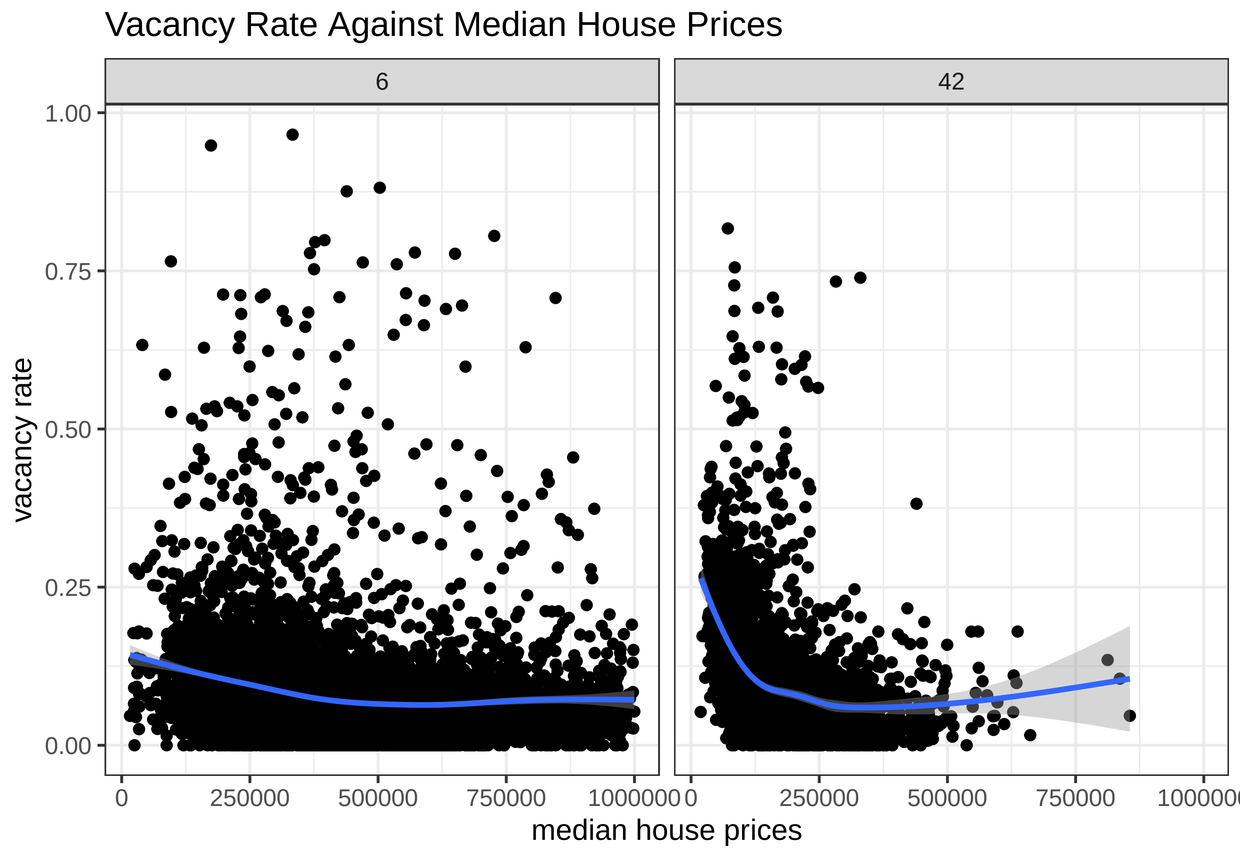
```
## 'geom_smooth()' using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



c.

```
ggplot(data = ca_pa) +
  geom_point(mapping = aes(x = Median_house_value, y = vacancy_rate))+
  geom_smooth(mapping = aes(x = Median_house_value, y = vacancy_rate))+
  labs(x = "median house prices",
       y = "vacancy rate",
       title = "Vacancy Rate Against Median House Prices") +
  facet_wrap(~ STATEFP, scale = "fixed") +
  theme_bw()
```

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



The plot shows that in California, house vacancy rate and number of houses at house prices between 0 and 1000000$ are rather even, while in Pennsylvania there are a lot more houses at low prices(0~250000$) , house vacancy rate is also higher in this range.

4.

a.

```
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)
```

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

Variable `acca` in the block of code gives the row indices which the house is in Alameda County (county1 in STATEFP 6), while variable `accamhv` gives the median house values of these houses. Using `median()` function, the whole block gives the median level of house price in Alameda County.

b.

```
median(ca_pa[which(ca_pa$STATEFP == 6 & ca_pa$COUNTYFP ==1),10])
```

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

c.

```
mean(ca_pa[which((ca_pa$STATEFP == 6 & ca_pa$COUNTYFP ==1)|
              (ca_pa$STATEFP == 6 & ca_pa$COUNTYFP ==85)|
              (ca_pa$STATEFP == 42 & ca_pa$COUNTYFP ==3)),16])
```

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

⇒ The average percentages of housing built since 2005 is 2.437344.

d.

```
#(i)
cor(ca_pa$Median_house_value,ca_pa$Built_2005_or_later)
```

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

```
#(ii)
cor(ca_pa[which(ca_pa$STATEFP==6),10],ca_pa[which(ca_pa$STATEFP==6),16])
```

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

```
#(iii)
cor(ca_pa[which(ca_pa$STATEFP==42),10],ca_pa[which(ca_pa$STATEFP==42),16])
```

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

```
#(iv)
cor(ca_pa[which(ca_pa$STATEFP == 6 & ca_pa$COUNTYFP == 1),10],
  ca_pa[which(ca_pa$STATEFP==6 & ca_pa$COUNTYFP == 1),16])
```

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

```
#(v)
cor(ca_pa[which(ca_pa$STATEFP == 6 & ca_pa$COUNTYFP == 85),10],
  ca_pa[which(ca_pa$STATEFP==6 & ca_pa$COUNTYFP == 85),16])
```

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

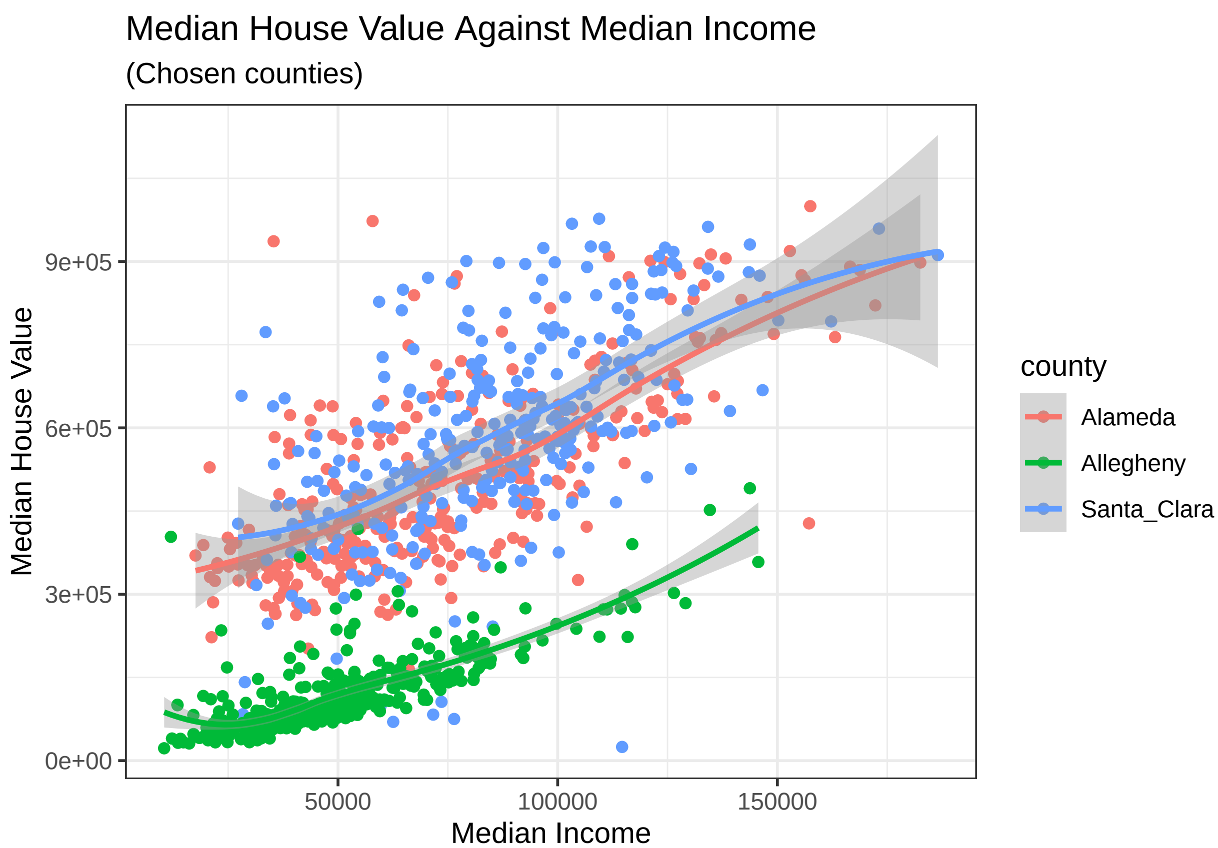
```r
#(vi)
cor(ca_pa[which(ca_pa$STATEFP == 42 & ca_pa$COUNTYFP == 3),10],
  ca_pa[which(ca_pa$STATEFP==42 & ca_pa$COUNTYFP == 3),16])
```

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

e.

```r
#extract related information
Alameda <- data.frame(Median_house_value = ca_pa[which(ca_pa$STATEFP == 6 & ca_pa$COUNTYFP == 1),10], Me

Santa_Clara <- data.frame(Median_house_value = ca_pa[which(ca_pa$STATEFP == 6 & ca_pa$COUNTYFP == 85),1

Allegheny <- data.frame(Median_house_value = ca_pa[which(ca_pa$STATEFP == 42 & ca_pa$COUNTYFP == 3),10]

three_counties <- rbind(Alameda, Santa_Clara, Allegheny)

#plot
ggplot(data = three_counties) +
  geom_point(aes(x = Median_income, y = Median_house_value, color = county)) +
  geom_smooth(aes(x = Median_income, y = Median_house_value, color = county)) +
  labs(x = "Median Income",
       y = "Median House Value",
       title = "Median House Value Against Median Income",
       subtitle = "(Chosen counties)") +
  theme_bw()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



MB.Ch1.11

Given code block:

```
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
```

`table()` function counts the elements by their factor levels, and the factor levels are assumed to be ordered, that's why the first and second output has different orders.

In the third command, `"male"` in the original dataframe `gender` cannot be paired with any of the levels, thus these elements in the output `gender` becomes `NA`:

```
gender <- factor(c(rep("female", 91), rep("male", 92)))
gender <- factor(gender, levels=c("Male", "female"))
is.na(gender[92])
```

```
## [1] TRUE
```

In the last command, expression `exclude=NULL` makes `NA` an extra level, and is the last level printed as `<NA>`.

MB.Ch1.12

```
prop_over <- function(x,c){
  num1 <- 0
  i <- 1
  while (i <= length(x)){
    if (x[i]>c){
      num1 <- num1+1
    }
    i <- i+1
  }
    prop <- num1/length(x)
  return(prop)
}
```

a.

```
prop_over(seq(1,100),60)
```

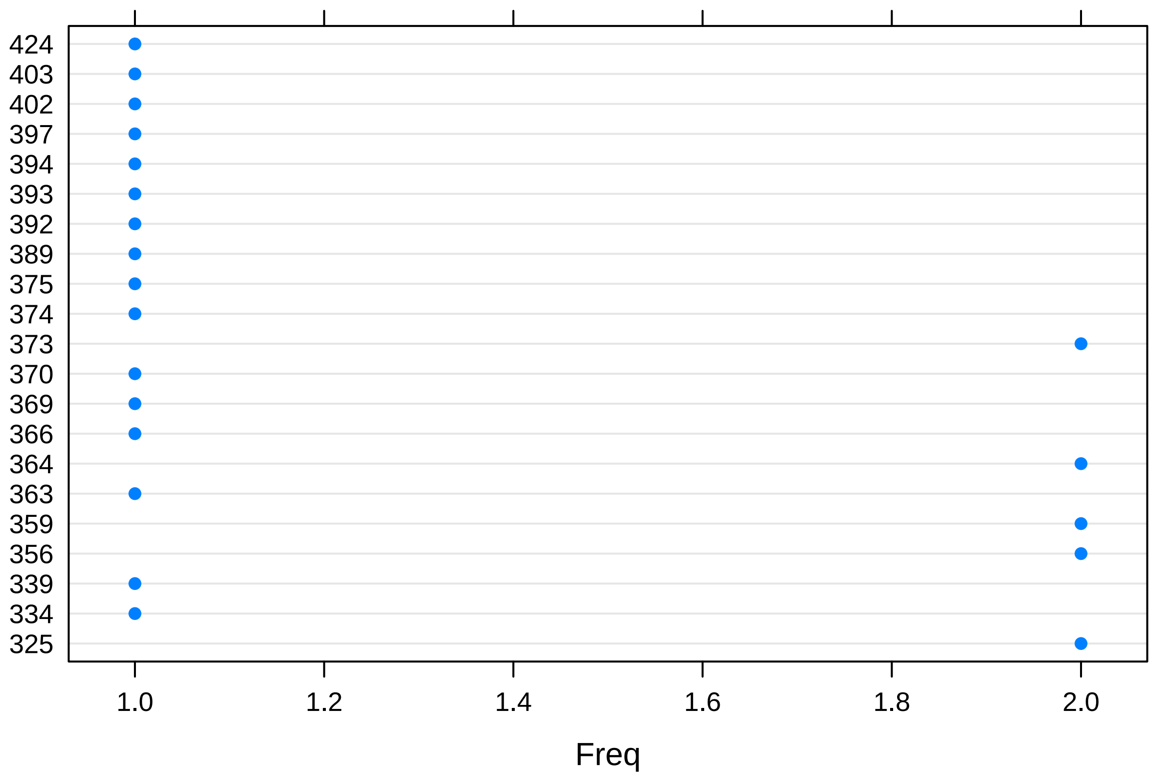
```
## [1] 0.4
```

b.

```r
library(Devore7)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

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

## Loading required package: lattice
```

```r
dotplot(ex01.36)
```




```r
my_ex01.36<- ex01.36$C1
prop_over(my_ex01.36,420)
```

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

MB.Ch1.18

```r
library(MASS)
```

Using `unstack` function:

```r
Value <- unstack(Rabbit, BPchange~Animal)
Treatment <- unstack(Rabbit, Treatment~Animal)[,1]
Dose <- unstack(Rabbit, Dose~Animal)[,1]

Rabbit <- data.frame(Treatment, Dose, Value)
Rabbit
```

```
##    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
```