Homework2

22560448 邵胜强

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))
   ```

   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.

```
> ca_pa <- read.csv("data/calif_penn_2011.csv", stringsAsFactors = FALSE)
> nrow(ca_pa); ncol(ca_pa)
[1] 11275
[1] 34
> dim(ca_pa)
[1] 11275    34
> colSums(apply(ca_pa, c(1, 2), is.na))
                       X                   GEO.id2                    STATEFP                 COUNTYFP
                       0                         0                          0                        0
                 TRACTCE                POPULATION                   LATITUDE                LONGITUDE
                       0                         0                          0                        0
       GEO.display.label        Median_house_value                Total_units             Vacant_units
                       0                       599                          0                        0
            Median_rooms Mean_household_size_owners Mean_household_size_renters     Built_2005_or_later
                     157                       215                        152                       98
        Built_2000_to_2004               Built_1990s                Built_1980s              Built_1970s
                       98                        98                         98                       98
             Built_1960s               Built_1950s                Built_1940s     Built_1939_or_earlier
                       98                        98                         98                       98
              Bedrooms_0                Bedrooms_1                 Bedrooms_2               Bedrooms_3
                       98                        98                         98                       98
              Bedrooms_4        Bedrooms_5_or_more                     Owners                  Renters
                       98                        98                        100                      100
    Median_household_income     Mean_household_income
                     115                       126
> ## 含义：按“先行后列”判断是否为 NA，再对每一列求 NA 的个数
> ca_pa_clean <- na.omit(ca_pa);
> n_deleted <- nrow(ca_pa) - nrow(ca_pa_clean); n_deleted
[1] 670
```

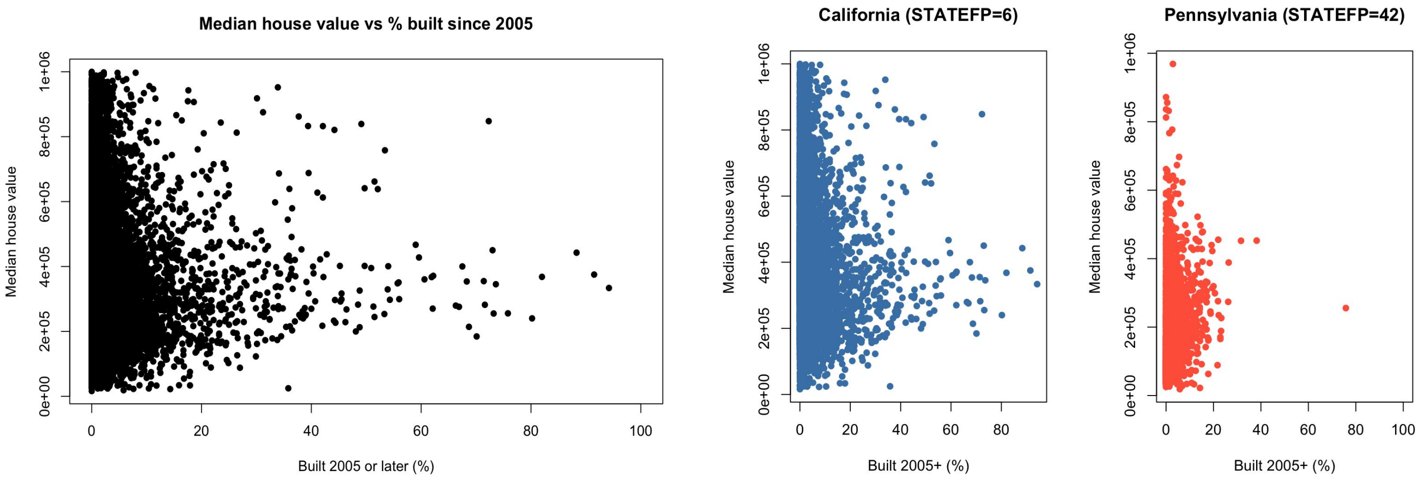
2. *This Very New House*

   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.

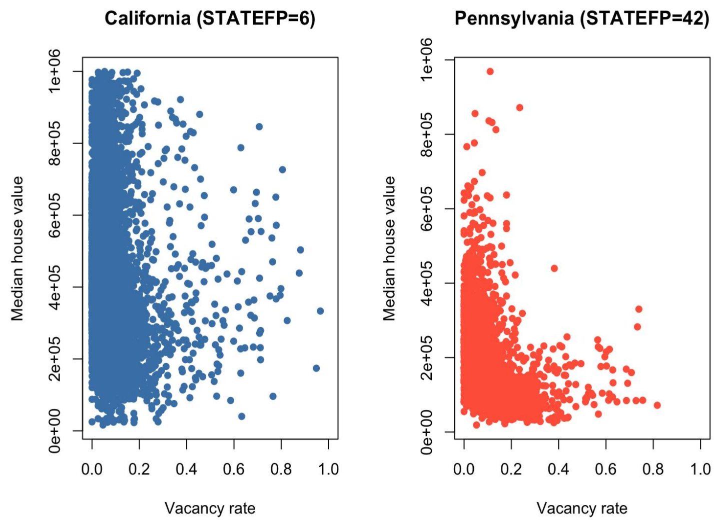
3. *Nobody Home*
   The vacancy rate is the fraction of housing units which are not occupied. The dataframe contains columns giving the total number of housing units for each Census tract, and the number of vacant housing units.

   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?

```r
plot(ca_pa$Built_2005_or_later, ca_pa$Median_house_value,
     xlab = "Built 2005 or later (%)",
     ylab = "Median house value",
     main = "Median house value vs % built since 2005",
     pch = 16)
par(mfrow = c(1, 2))
with(subset(ca_pa, STATEFP == 6),  # California
     plot(Built_2005_or_later, Median_house_value,
          xlab = "Built 2005+ (%)",
          ylab = "Median house value",
          main = "California (STATEFP=6)",
          pch = 16, col = "steelblue"))
with(subset(ca_pa, STATEFP == 42), # Pennsylvania
     plot(Built_2005_or_later, Median_house_value,
          xlab = "Built 2005+ (%)",
          ylab = "Median house value",
          main = "Pennsylvania (STATEFP=42)",
          pch = 16, col = "tomato"))
par(mfrow = c(1, 1))
```




```r
> ca_pa$vacancy_rate <- with(ca_pa, Vacant_units / Total_units)
> min(ca_pa$vacancy_rate, na.rm = TRUE)
[1] 0
> max(ca_pa$vacancy_rate, na.rm = TRUE)
[1] 1
> mean(ca_pa$vacancy_rate, na.rm = TRUE)
[1] 0.08917878
> median(ca_pa$vacancy_rate, na.rm = TRUE)
[1] 0.06766326
> plot(ca_pa$vacancy_rate, ca_pa$Median_house_value,
+      xlab = "Vacancy rate", ylab = "Median house value",
+      main = "Median house value vs vacancy rate",
+      pch = 16)
> par(mfrow = c(1, 2))
> with(subset(ca_pa, STATEFP == 6),
+      plot(vacancy_rate, Median_house_value,
+           xlab = "Vacancy rate", ylab = "Median house value",
+           main = "California (STATEFP=6)", pch = 16, col = "steelblue"))
> with(subset(ca_pa, STATEFP == 42),
+      plot(vacancy_rate, Median_house_value,
+           xlab = "Vacancy rate", ylab = "Median house value",
+           main = "Pennsylvania (STATEFP=42)", pch = 16, col = "tomato"))
> par(mfrow = c(1, 1))
```

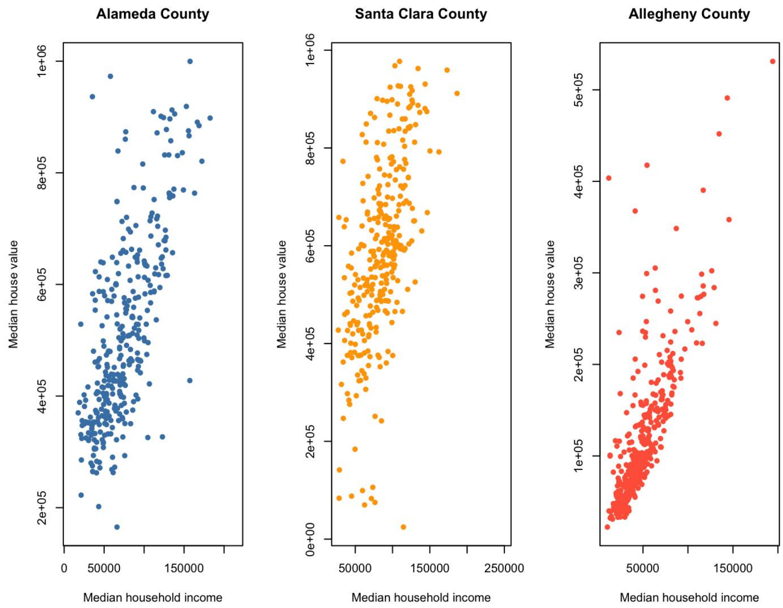


4. The column `COUNTYFP` contains a numerical code for counties within each state. We are interested in Alameda County (county 1 in California), Santa Clara (county 85 in California), and Allegheny County (county 3 in Pennsylvania).

   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,na.rm=TRUE)
[1] 473500
> median(subset(ca_pa, STATEFP == 6 & COUNTYFP == 1)[[col_value]], na.rm = TRUE)
[1] 473500
> mean(ca_pa$Built_2005_or_later[
+   ca_pa$STATEFP == 6 & ca_pa$COUNTYFP == 1
+ ], na.rm = TRUE)   # Alameda
[1] 2.932778
> mean(ca_pa$Built_2005_or_later[
+   ca_pa$STATEFP == 6 & ca_pa$COUNTYFP == 85
+ ], na.rm = TRUE)   # Santa Clara
[1] 3.160215
> mean(ca_pa$Built_2005_or_later[
+   ca_pa$STATEFP == 42 & ca_pa$COUNTYFP == 3
+ ], na.rm = TRUE)   # Allegheny
[1] 1.883375
```

```
> cor(ca_pa$Median_house_value, ca_pa$Built_2005_or_later, use = "complete.obs")
[1] -0.02052684
> cor(ca_pa$Median_house_value[ca_pa$STATEFP == 6],
+     ca_pa$Built_2005_or_later[ca_pa$STATEFP == 6],
+     use = "complete.obs")
[1] -0.1160322
> cor(ca_pa$Median_house_value[ca_pa$STATEFP == 42],
+     ca_pa$Built_2005_or_later[ca_pa$STATEFP == 42],
+     use = "complete.obs")
[1] 0.2339447
> cor(ca_pa$Median_house_value[ca_pa$STATEFP == 6  & ca_pa$COUNTYFP == 1],
+     ca_pa$Built_2005_or_later[ca_pa$STATEFP == 6  & ca_pa$COUNTYFP == 1],
+     use = "complete.obs")
[1] 0.01432789
> cor(ca_pa$Median_house_value[ca_pa$STATEFP == 6  & ca_pa$COUNTYFP == 85],
+     ca_pa$Built_2005_or_later[ca_pa$STATEFP == 6  & ca_pa$COUNTYFP == 85],
+     use = "complete.obs")
[1] -0.1726203
> cor(ca_pa$Median_house_value[ca_pa$STATEFP == 42 & ca_pa$COUNTYFP == 3],
+     ca_pa$Built_2005_or_later[ca_pa$STATEFP == 42 & ca_pa$COUNTYFP == 3],
+     use = "complete.obs")
[1] 0.1868602
> par(mfrow = c(1, 3))
> with(subset(ca_pa, STATEFP == 6 & COUNTYFP == 1),
+      plot(Median_household_income, Median_house_value,
+           xlab = "Median household income",
+           ylab = "Median house value",
+           main = "Alameda County", pch = 16, col = "steelblue"))
> with(subset(ca_pa, STATEFP == 6 & COUNTYFP == 85),
+      plot(Median_household_income, Median_house_value,
+           xlab = "Median household income",
+           ylab = "Median house value",
+           main = "Santa Clara County", pch = 16, col = "orange"))
> with(subset(ca_pa, STATEFP == 42 & COUNTYFP == 3),
+      plot(Median_household_income, Median_house_value,
+           xlab = "Median household income",
+           ylab = "Median house value",
+           main = "Allegheny County", pch = 16, col = "tomato"))
> par(mfrow = c(1, 1))
```



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
```

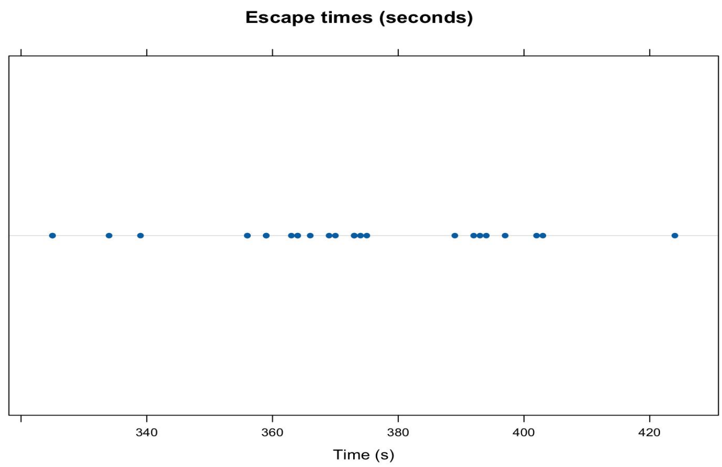
```
> gender <- factor(c(rep("female", 91), rep("male", 92)))
> table(gender)
gender
female   male
    91     92
> #因子水平自动按字母序记录为 c("female","male")。
> gender <- factor(gender, levels=c("male", "female"))
> table(gender)
gender
  male female
    92     91
> # 显式设定水平顺序为 c("male","female")，计数未变，但列出顺序变为 male 在前。
> gender <- factor(gender, levels=c("Male", "female"))
> # Note the mistake: "Male" should be "male"
> table(gender)
gender
  Male female
     0     91
> # "Male" 与原数据 "male" 大小写不一致 → "Male" 在数据中实际不存在
> table(gender, exclude=NULL)
gender
  Male female   <NA>
     0     91     92
> # 把 NA 也统计出来
> rm(gender)
```

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.

(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.

```
install.packages("Devore7")
library(Devore7)
data("ex01.36", package = "Devore7")
x <- as.numeric(ex01.36[[1]])
install.packages("lattice")
library(lattice)
dotplot(x, main = "Escape times (seconds)", xlab = "Time (s)")
prop_over_7min <- prop_above(x, 420)
prop_over_7min  # 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

```
> library(MASS)
> data("Rabbit", package = "MASS")
> df <- Rabbit
> df <- df[order(df$Treatment, df$Dose, df$Animal), ]
> bp_wide    <- unstack(df, BPchange ~ Animal)
> dose_wide <- unstack(df, Dose ~ Animal)
> treat_wide <- unstack(df, Treatment ~ Animal)
> Dose       <- dose_wide[[1]]
> Treatment <- treat_wide[[1]]
> out <- data.frame(
+   Treatment = as.character(Treatment),
+   Dose      = as.numeric(as.character(Dose)),
+   bp_wide[, c("R1","R2","R3","R4","R5")]
+ )
> out <- out[order(out$Treatment, out$Dose), ]
> row.names(out) <- NULL
> out
   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
```