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))`
   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")
dim(ca_pa)
```

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

```
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
```

```
ca_pa_clean <- na.omit(ca_pa)
nrow(ca_pa) - nrow(ca_pa_clean)
```

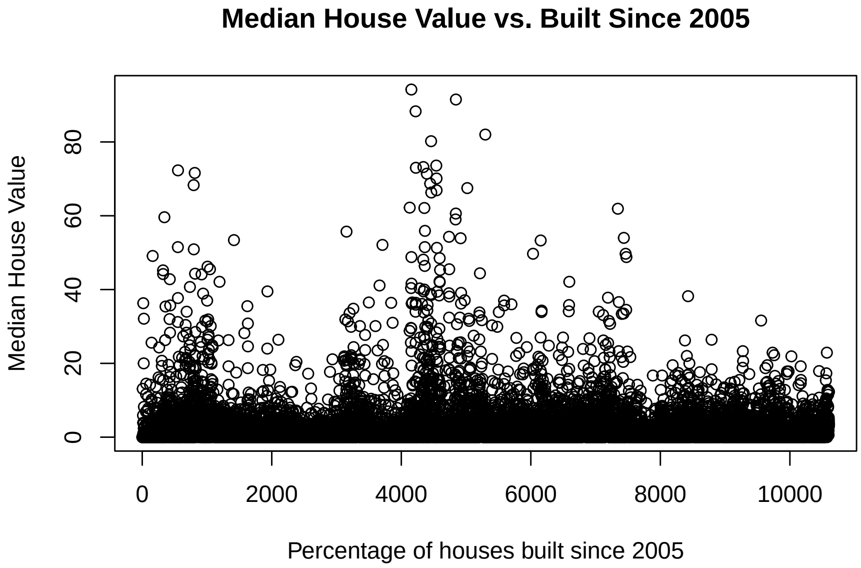
```
## [1] 670
```

- T1 答：(a) 代码见上；(b) 代码见上，11275 行和 34 列；(c) 该命令

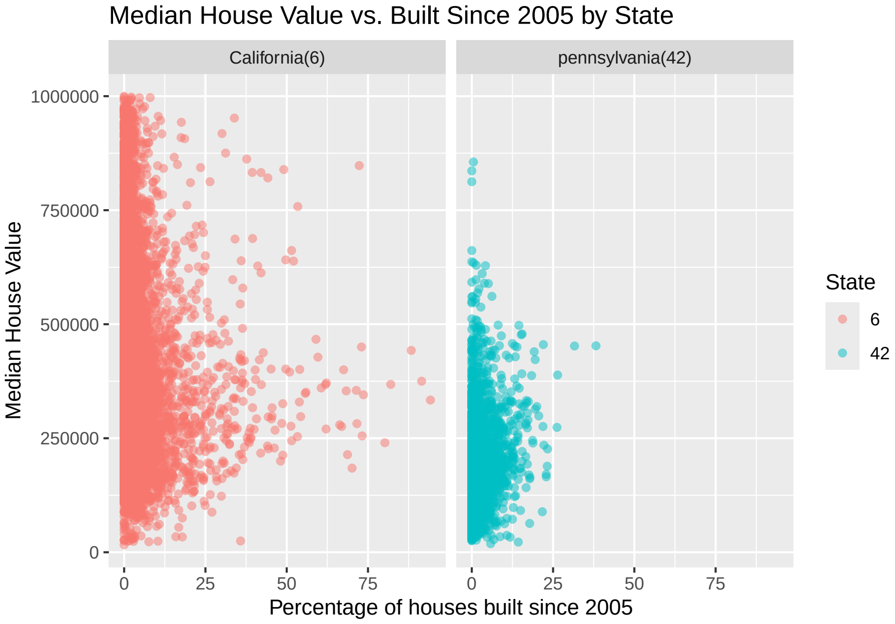
用于计算 ca_pa 中每一列包含的缺失值（NA）的数量；（d）代码见上；（e）代码见上，670 行。

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
# a
plot(ca_pa_clean$Built_2005_or_later, ca_pa_clean$Median_House_Value,
     xlab = "Percentage of houses built since 2005",
     ylab = "Median House Value",
     main = "Median House Value vs. Built Since 2005")
```



```r
# b
ggplot(ca_pa_clean, aes(x = Built_2005_or_later, y = Median_house_value,
                        color = factor(STATEFP))) +
  geom_point(alpha = 0.5) +
  facet_wrap(~ STATEFP, labeller = as_labeller(c(`6` = "California(6)",
                                                 `42` = "pennsylvania(42)"))) +
  labs(x = "Percentage of houses built since 2005",
       y = "Median House Value",
       color = "State") +
  ggtitle("Median House Value vs. Built Since 2005 by State")
```



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.

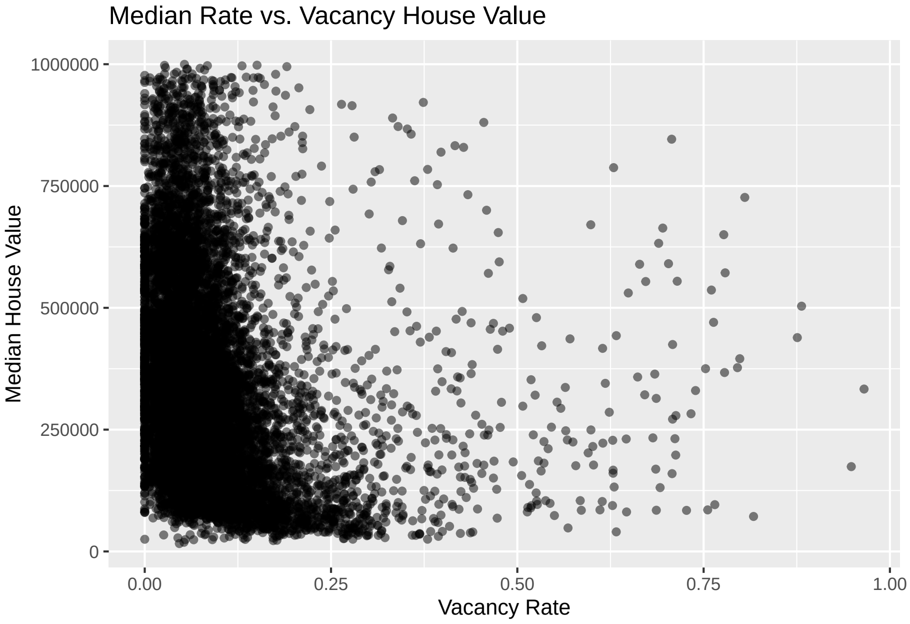
   a. Add a new column to the dataframe which contains the vacancy rate. What are the minimum, maximum, mean, and median vacancy rates?
   b. Plot the vacancy rate against median house value.
   c. Plot vacancy rate against median house value separately for Cal-

ifornia and for Pennsylvania. Is there a difference?

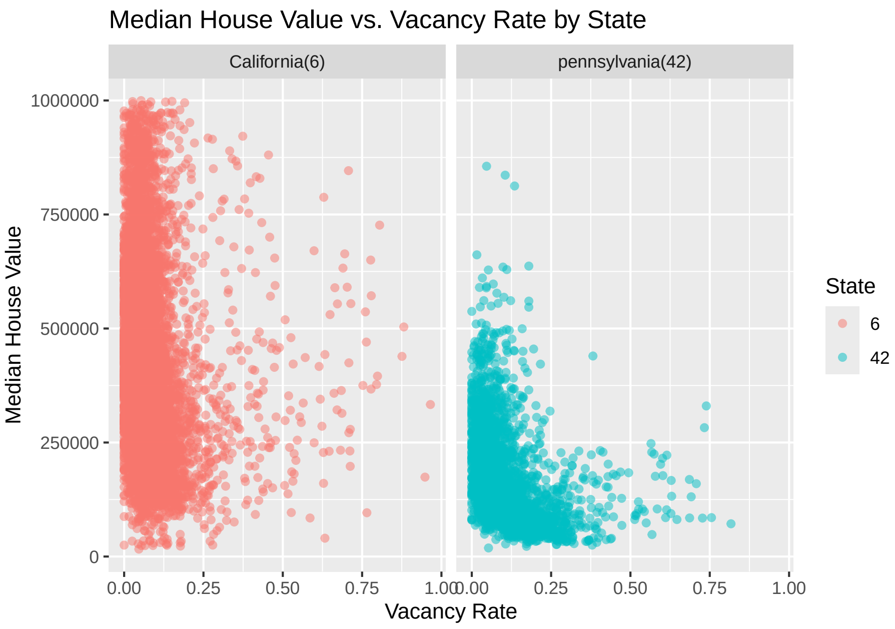
```
# a
ca_pa_clean$Vacancy_rate <- ca_pa_clean$Vacant_units / ca_pa_clean$Total_units
summary(ca_pa_clean$Vacancy_rate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.03846 0.06767 0.08889 0.10921 0.96531
```

```
# b
ggplot(ca_pa_clean, aes(x = Vacancy_rate, y = Median_house_value)) +
  geom_point(alpha = 0.5) +
  labs(x = "Vacancy Rate", y = "Median House Value") +
  ggtitle("Median Rate vs. Vacancy House Value")
```



```
# c
ggplot(ca_pa_clean, aes(x = Vacancy_rate, y = Median_house_value,
                        color = factor(STATEFP))) +
  geom_point(alpha = 0.5) +
  facet_wrap(~ STATEFP, labeller = as_labeller(c(`6` = "California(6)",
                                               `42` = "pennsylvania(42)"))) +
labs(x = "Vacancy Rate", y = "Median House Value", color = "State") +
ggtitle("Median House Value vs. Vacancy Rate by State")
```



- T3 答：（a）代码见上, 空置率最小为 0.00000，最大为 0.96531，平均为 0.08889，中位数为 0.06767；（b）代码见上；（c）代码见上，注意到有区别，且具体来说：首先，在绝对价格上，加利福尼亚州的房价中位数远高于宾夕法尼亚州。其次，在数据分布上，加州的房屋市场更为紧俏，绝大多数地区的空置率都紧密集中在 25% 以下；而宾州的空置率分布则相对更广泛。在关系趋势方面，两个州都表现出房价中位数和空置率之间的负相关性。这种关系主要体现为：最高价位的房产只存在于低空置率的地区。随着空置率的增加，房价的上限随之降低，这表明高空置率对维持高昂的房产价值有很强的抑制作用。

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.)

```r
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)
```

```r
# b
ca_pa_clean %>%
  filter(STATEFP == 6, COUNTYFP == 1) %>%
  summarise(Median_house_value = median(Median_house_value))
```

```
##   Median_house_value
## 1             474050
```

```
median(ca_pa_clean$Median_house_value[ca_pa_clean$STATEFP == 6 &
                                        ca_pa_clean$COUNTYFP == 1])
```

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

```
# c
ca_pa_clean %>%
  filter((STATEFP == 6 & COUNTYFP %in% c(1, 85)) |
           (STATEFP == 42 & COUNTYFP == 3)) %>%
  group_by(STATEFP, COUNTYFP) %>%
  summarise(avg_built_since_2005 = mean(Built_2005_or_later))
```

```
## `summarise()` has grouped output by 'STATEFP'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 3 x 3
## # Groups:   STATEFP [2]
##   STATEFP COUNTYFP avg_built_since_2005
##     <int>    <int>                <dbl>
## 1       6        1                 2.82
## 2       6       85                 3.20
## 3      42        3                 1.47
```

```
# d
cor(ca_pa_clean$Median_house_value, ca_pa_clean$Built_2005_or_later,
    use = "complete.obs")
```

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

```
ca_ca <- filter(ca_pa_clean, STATEFP == 6)
cor(ca_ca$Median_house_value, ca_ca$Built_2005_or_later,
    use = "complete.obs")
```

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

```r
pa_pa <- filter(ca_pa_clean, STATEFP == 42)
cor(pa_pa$Median_house_value, pa_pa$Built_2005_or_later,
    use = "complete.obs")
```

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

```r
acca <- filter(ca_pa_clean, STATEFP == 6, COUNTYFP == 1)
cor(acca$Median_house_value, acca$Built_2005_or_later,
    use = "complete.obs")
```

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

```r
accamhv <- filter(ca_pa_clean, STATEFP == 6, COUNTYFP == 85)
cor(accamhv$Median_house_value, accamhv$Built_2005_or_later,
    use = "complete.obs")
```

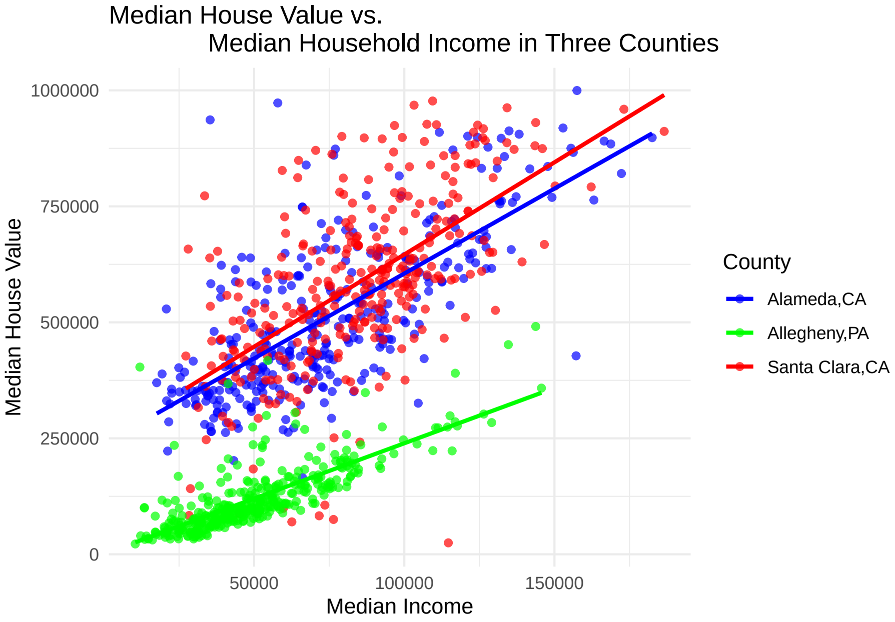
```
## [1] -0.1726203
```

```r
allegheny <- filter(ca_pa_clean, STATEFP == 42, COUNTYFP == 3)
cor(allegheny$Median_house_value, allegheny$Built_2005_or_later,
    use = "complete.obs")
```

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

```r
# e
three_counties <- ca_pa_clean %>%
  filter((STATEFP == 6 & COUNTYFP %in% c(1, 85)) |
           (STATEFP == 42 & COUNTYFP == 3)) %>%
  mutate(CountyName = case_when(
    STATEFP == 6 & COUNTYFP == 1 ~ "Alameda,CA",
    STATEFP == 6 & COUNTYFP == 85 ~ "Santa Clara,CA",
    STATEFP == 42 & COUNTYFP == 3 ~ "Allegheny,PA",
    TRUE ~ "Other"
  ))
ggplot(three_counties, aes(x = Median_household_income, y = Median_house_value,
                           color = CountyName)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Median House Value vs.
            Median Household Income in Three Counties",
       x = "Median Income", y = "Median House Value",
       color = "County") +
  theme_minimal() +
  scale_color_manual(values = c("Alameda,CA" = "blue",
                              "Santa Clara,CA" = "red",
                              "Allegheny,PA" = "green"))
```

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



* T4 答：（a）该代码块的目的是计算加利福尼亚州阿拉米达县（COUNTYFP=1）中房屋建造于 2005 年或之后的房屋的中位数房价。它通过遍历每个地块，检查其州和县代码，并将符合条件的地块的房价存储在 accamhv 向量中，最后计算并返回该向量的中位数；（b）代码见上，也得到 Median_house_value 为 474050；（c）代码见上，平均百分比依次为 2.82,3.20,1.47；（d）代码见上；（e）代码见上。

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().

- Ch1.11 答： 1. 第一次使用: 首次创建因子时，默认按照字母顺序排列，因此 “female” 在 “male” 之前。输出为 `female` 91, `male` 92。2. 第二次使用: 通过指定 levels 参数（`levels=c("Male", "female")`），将因子水平更改为 `male` 和 `female`，因此输出顺序也相应更改。输出为 `male` 92, `female` 91。 3. 第三次使用: 指定 levels 参数（`levels=c("Male",` "female"))，将因子水平更改为 Male 和 female，由于 “Male” 与 “male” 不匹配，因此 “Male” 被视为一个新的水平，从而输出为 Male 0, female 91。 4. 第四次使用: 通过设置 exclude = NULL，将所有因子水平都包含在输出中，即使它们没有观察值。于是 table() 函数也要统计 NA 值。因此输出为 Male 0,female 91, <NA> 92 。这 92 次其实对应于那些与新水平 "Male" 不匹配的原始 "male" 值。

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.

```
proportion_exceeding <- function(x, cutoff) {
  num_exceeding <- sum(x > cutoff)
  proportion <- num_exceeding / length(x)
  return(proportion)
}
# 测试函数
test_vector <- 1:100
proportion_exceeding(test_vector, 90) # 应返回 0.1
```

```
## [1] 0.1
```

```
proportion_exceeding(test_vector, 50) # 应返回 0.5
```

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

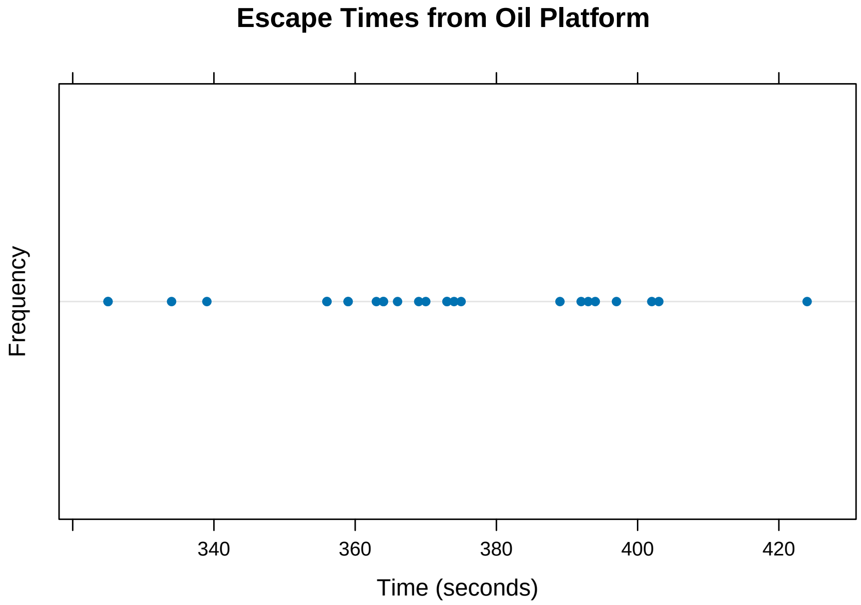
(b) Obtain the vector ex01.36 from the Devore6 (or Devore7) package. These data give the times required for individuals to escape from an oil platform during a drill. Use dotplot() to show the distribution of times. Calculate the proportion of escape times that exceed 7 minutes.

```
# 请预先安装 Devore7 包
if (!requireNamespace("Devore7", quietly = TRUE)) {
  install.packages("Devore7")
}
```

```
library(Devore7)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:DAAG':
##
##     hills

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

## Loading required package: lattice
```

```
data(ex01.36)
dotplot(~ C1, data = ex01.36, main = "Escape Times from Oil Platform",
        xlab = "Time (seconds)", ylab = "Frequency")
```



```
proportion_exceeding(ex01.36$C1, 420) # 超过 7 分钟（420 秒）的比例
```

```
## [1] 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)

# 第一次 unstack: 按兔子 ID 将血压变化分开
unstacked1 <- unstack(Rabbit, BPchange ~ Animal)
# 第二次 unstack: 按兔子 ID 将剂量分开
unstacked2 <- unstack(Rabbit, Dose ~ Animal)
# 第三次 unstack: 按兔子 ID 将治疗方法分开
unstacked3 <- unstack(Rabbit, Treatment ~ Animal)

# 合并结果
final_df <- data.frame(
  Treatment = unstacked3$R1,
  Dose = unstacked2$R1,
  R1 = unstacked1$R1,
  R2 = unstacked1$R2,
  R3 = unstacked1$R3,
  R4 = unstacked1$R4,
  R5 = unstacked1$R5
)

print(final_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
```