homework2

黄舟翔 3220103606

2025-06-27

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`.

```
library(tidyverse)
ca_pa <- read.csv("data/calif_penn_2011.csv")
```

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

```
cat(" 原始数据维度:", nrow(ca_pa), " 行 ×", ncol(ca_pa), " 列\n")
```

```
## 原始数据维度: 11275 行 × 34 列
```

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

该代码是在完成每列缺失值数量的统计。

```
na_counts <- colSums(apply(ca_pa,c(1,2),is.na))
cat("\n每列缺失值统计:\n")
```

```
##
## 每列缺失值统计:
```

```
print(na_counts)
```

```
##                           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
```

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.

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

e. How many rows did this eliminate?

```
rows_eliminated <- nrow(ca_pa) - nrow(ca_pa_clean)
cat("\n删除的行数:", rows_eliminated)
```

```
##
## 删除的行数: 670
```

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

是兼容的：na.omit() 删除包含任意 NA 的整行

理论删除行数 = (所有列 NA 总数之和)/列数

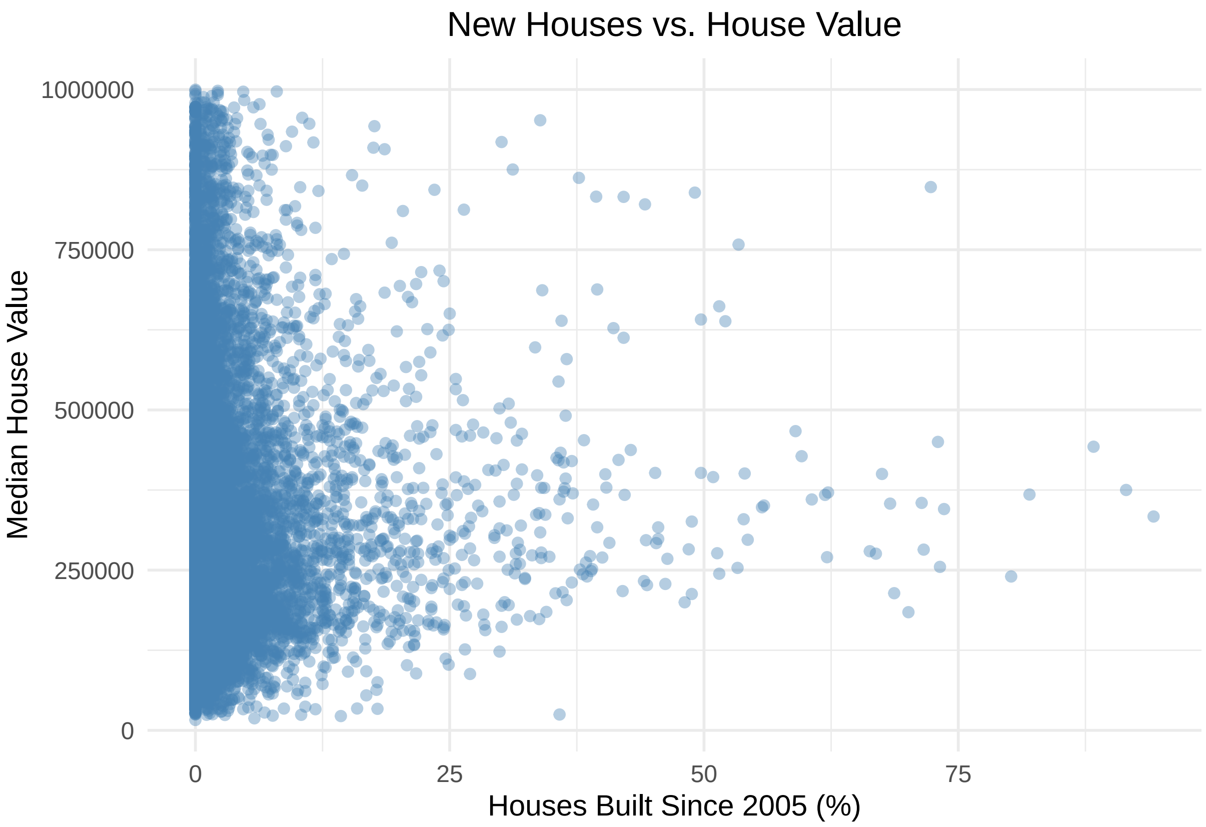
实际因每行可能有多个 NA，但删除行数应与 NA 分布一致

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.

可以用下列代码解决问题

```
p1 <- ggplot(ca_pa_clean, aes(x = Built_2005_or_later, y = Median_house_value)) +
  geom_point(alpha = 0.4, color = "steelblue") +
  labs(title = "New Houses vs. House Value",
       x = "Houses Built Since 2005 (%)",
       y = "Median House Value") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        plot.margin = margin(t=10, r=10, b=10, l=10, unit="pt"))
print(p1)
```

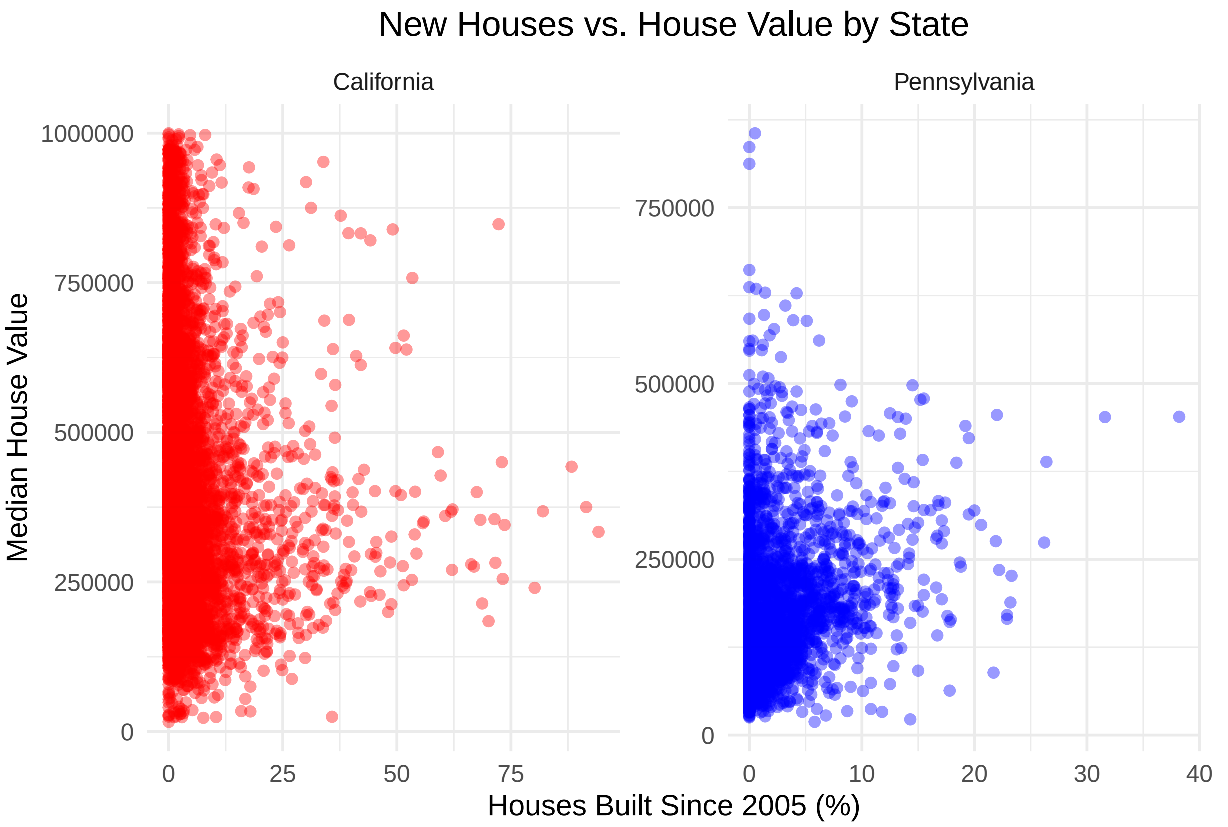


b. Make a new plot, or pair of plots, which breaks this out by state. Note that the state is recorded in the STATEFP variable, with California being state 6 and Pennsylvania state 42.

可以用下列代码解决

```
ca_pa_clean <- ca_pa_clean %>%
  mutate(State = ifelse(STATEFP == 6, "California",
                        ifelse(STATEFP == 42, "Pennsylvania", "Other")))
p2 <- ggplot(ca_pa_clean, aes(x = Built_2005_or_later,
  y = Median_house_value, color = State)) +
  geom_point(alpha = 0.4) +
  scale_color_manual(values = c("California" = "red", "Pennsylvania" = "blue")) +
  facet_wrap(~State, scales = "free") +
  labs(title = "New Houses vs. House Value by State",
       x = "Houses Built Since 2005 (%)",
       y = "Median House Value") +
  theme_minimal() +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5),
        plot.margin = margin(t=10, r=10, b=10, l=10, unit="pt"))
```

```r
print(p2)
```



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?

可以用以下代码完成：

```r
ca_pa_clean <- ca_pa_clean %>%
  mutate(Vacancy_Rate = Vacant_units / Total_units)

vacancy_stats <- summary(ca_pa_clean$Vacancy_Rate)
print(vacancy_stats)
```

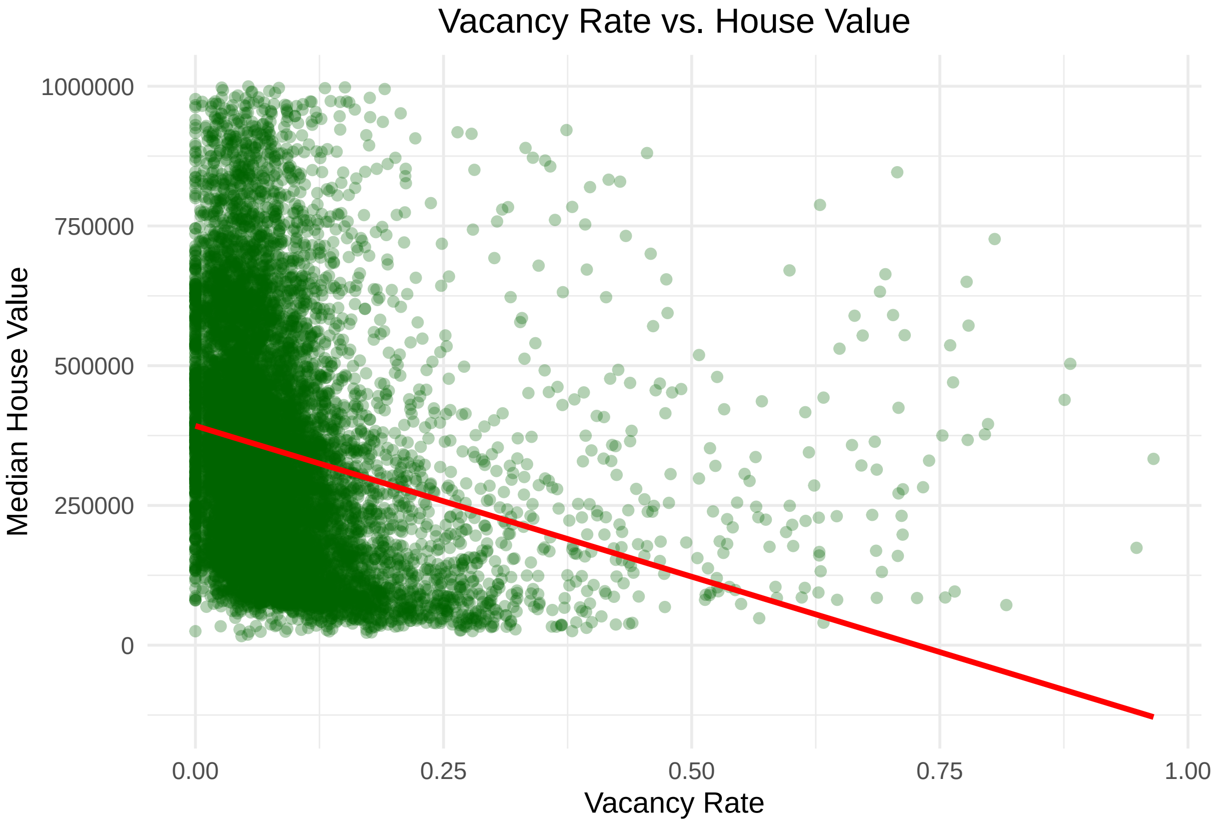
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.03846 0.06767 0.08889 0.10921 0.96531
```

b. Plot the vacancy rate against median house value.

可以用以下代码完成：

```
p3 <- ggplot(ca_pa_clean, aes(x = Vacancy_Rate, y = Median_house_value)) +
  geom_point(alpha = 0.3, color = "darkgreen") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Vacancy Rate vs. House Value",
       x = "Vacancy Rate",
       y = "Median House Value") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        plot.margin = margin(10, 10, 10, 10))
print(p3)
```

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

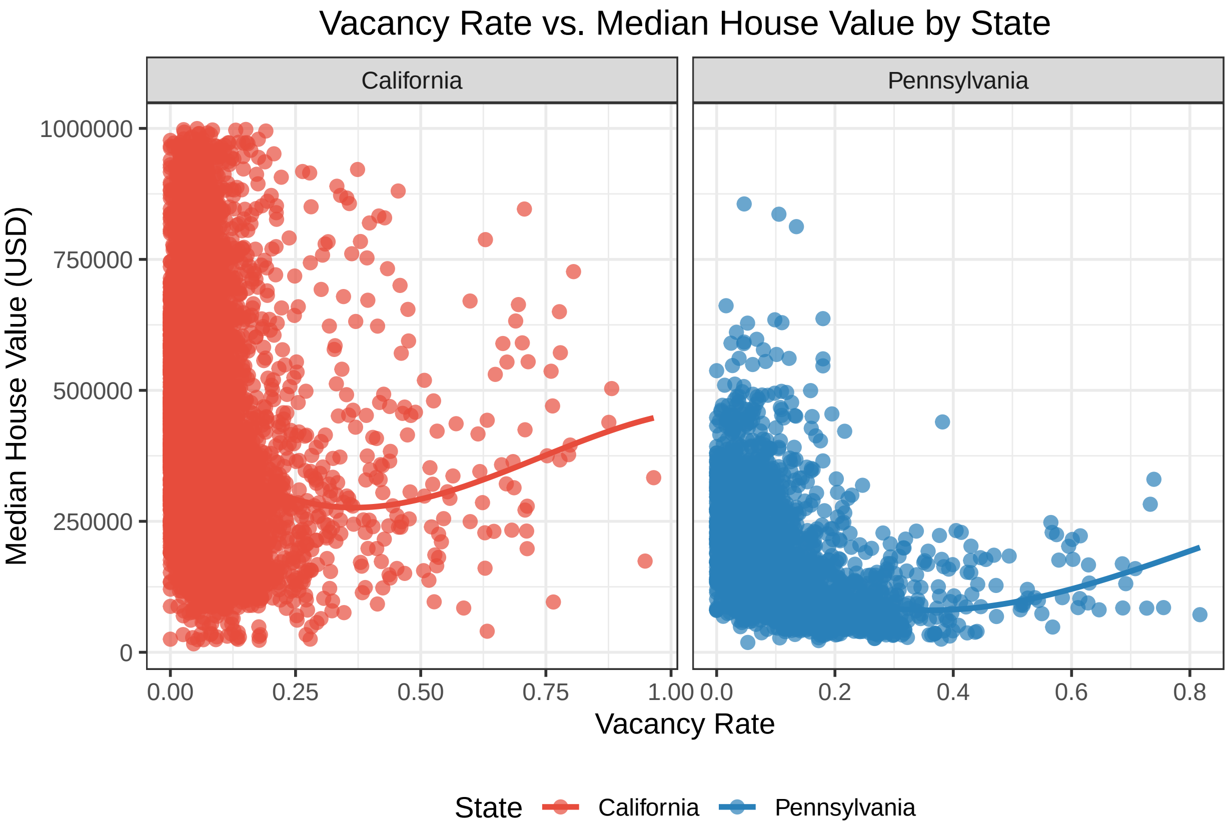


c. Plot vacancy rate against median house value separately for California and for Pennsylvania. Is there a difference?

可以用下列代码完成

```r
ca_pa_clean <- ca_pa_clean %>%
  mutate(state = case_when(
    STATEFP == 6 ~ "California",
    STATEFP == 42 ~ "Pennsylvania",
    TRUE ~ "Other"
  )) %>%
  filter(state != "Other")  # 仅保留两个州的数据

# 分面散点图
ggplot(ca_pa_clean, aes(y = Median_house_value, x = Vacancy_Rate)) +
  geom_point(aes(color = state), alpha = 0.7, size = 2) +
  geom_smooth(aes(color = state), method = "loess", se = FALSE) +
  facet_wrap(~state, scales = "free_x") +
  scale_color_manual(values = c("California" = "#E74C3C",
                                "Pennsylvania" = "#2980B9")) +
  labs(title = "Vacancy Rate vs. Median House Value by State",
       y = "Median House Value (USD)",
       x = "Vacancy Rate",
       color = "State") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5),
        legend.position = "bottom")
```

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



```
ca_stats <- ca_pa_clean %>%
  filter(state == "California") %>%
  summarise(correlation = cor(Median_house_value, Vacancy_Rate, use = "complete.obs"))

pa_stats <- ca_pa_clean %>%
  filter(state == "Pennsylvania") %>%
  summarise(correlation = cor(Median_house_value, Vacancy_Rate, use = "complete.obs"))

cat("\n差异观察:\n")
```

```
##
## 差异观察:
```

```
cat("- 加利福尼亚: 房价与空置率相关性:", round(ca_stats$correlation, 3),
    "(高房价区域空置率普遍较低, 低端市场空置率较高)\n")
```

```
## - 加利福尼亚: 房价与空置率相关性: -0.158 (高房价区域空置率普遍较低, 低端市场空置率较高)
```

```r
cat("- 宾夕法尼亚：房价与空置率相关性:", round(pa_stats$correlation, 3),
    "(相关性较弱，极端高空置率现象较少)\n")
```

```
## - 宾夕法尼亚：房价与空置率相关性: -0.336 (相关性较弱，极端高空置率现象较少)
```

```r
cat("- 对比：加州房价对空置率的影响更明显，高端房产市场空置率显著低于宾州")
```

```
## - 对比：加州房价对空置率的影响更明显，高端房产市场空置率显著低于宾州
```

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.

可以用以下代码完成

```r
acca <- c()
for (tract in 1:nrow(ca_pa_clean)) {
  if (ca_pa$STATEFP[tract] == 6) {           # 筛选加州（州代码 6）
    if (ca_pa$COUNTYFP[tract] == 1) {        # 筛选阿拉米达县（县代码 1）
      acca <- c(acca, tract)                # 记录符合条件的行号
    }
  }
}
accamhv <- c()
for (tract in acca) {
  accamhv <- c(accamhv, ca_pa[tract, 10])    # 提取第 10 列（房屋价值中位数）
}
median(accamhv)                             # 计算所有区的中位数
```

```
## [1] NA
```

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.

可以用以下代码解决

```
median(ca_pa$Median_house_value[ca_pa$STATEFP == 6 & ca_pa$COUNTYFP == 1], na.rm = TRUE)
```

```
## [1] 473500
```

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

```
ca_pa %>%
  filter(
    (STATEFP == 6 & COUNTYFP == 1) |     # Alameda, CA
    (STATEFP == 6 & COUNTYFP == 85) |    # Santa Clara, CA
    (STATEFP == 42 & COUNTYFP == 3)      # Allegheny, PA
  ) %>%
  group_by(STATEFP, COUNTYFP) %>%
  summarise(
    Avg_Percent_New_Housing = mean(Built_2005_or_later, na.rm = TRUE),
    .groups = "drop"
  )
```

```
## # A tibble: 3 x 3
##   STATEFP COUNTYFP Avg_Percent_New_Housing
##     <int>    <int>                   <dbl>
## 1       6        1                    2.93
## 2       6       85                    3.16
## 3      42        3                    1.88
```

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?

```
cor_list <- list(
  whole = cor(ca_pa$Median_house_value, ca_pa$Built_2005_or_later,
              use = "complete.obs"),
  california = cor(ca_pa$Median_house_value[ca_pa$STATEFP==6],
                   ca_pa$Built_2005_or_later[ca_pa$STATEFP==6], use = "complete.obs"),
  pennsylvania = cor(ca_pa$Median_house_value[ca_pa$STATEFP==42],
                     ca_pa$Built_2005_or_later[ca_pa$STATEFP==42],
                     use
                     = "complete.obs"),
```

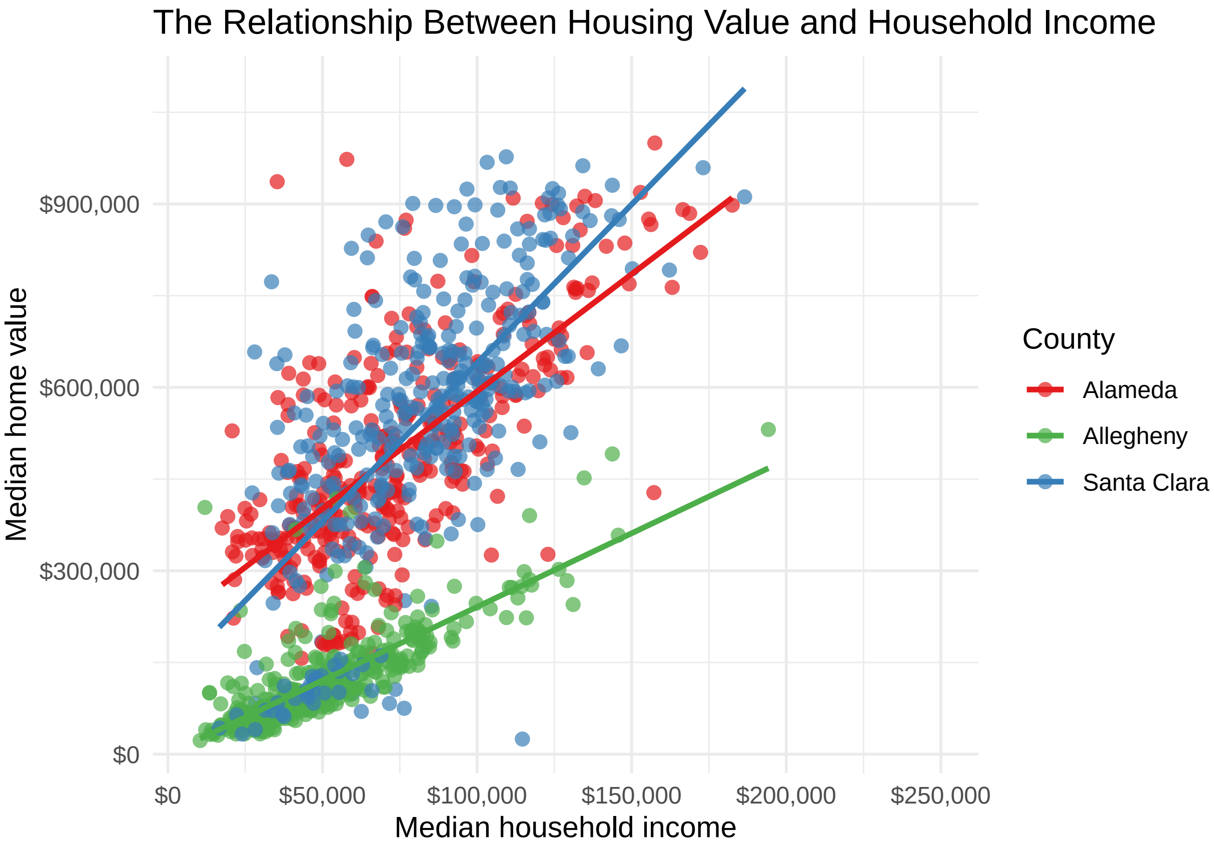
```r
  alameda = cor(ca_pa$Median_house_value[ca_pa$COUNTYFP==1],
                ca_pa$Built_2005_or_later[ca_pa$COUNTYFP==1],
                use = "complete.obs"),
  santa_clara = cor(ca_pa$Median_house_value[ca_pa$COUNTYFP==85],
                    ca_pa$Built_2005_or_later[ca_pa$COUNTYFP==85],
                    use = "complete.obs"),
  allegheny = cor(ca_pa$Median_house_value[ca_pa$COUNTYFP==3],
                  ca_pa$Built_2005_or_later[ca_pa$COUNTYFP==3], use = "complete.obs")
)
print(cor_list)
```

```
## $whole
## [1] -0.02052684
##
## $california
## [1] -0.1160322
##
## $pennsylvania
## [1] 0.2339447
##
## $alameda
## [1] -0.0357215
##
## $santa_clara
## [1] -0.08501218
##
## $allegheny
## [1] 0.1846541
```

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
library(ggplot2)
ca_pa %>%
  filter(COUNTYFP %in% c(1, 85, 3)) %>%
  mutate(County = case_when(
    COUNTYFP == 1 ~ "Alameda",
    COUNTYFP == 85 ~ "Santa Clara",
    COUNTYFP == 3 ~ "Allegheny"
  )) %>%
  ggplot(aes(x = Median_household_income, y = Median_house_value, color = County)) +
  geom_point(alpha = 0.7, size = 2) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_continuous(labels = scales::dollar) +
  scale_x_continuous(labels = scales::dollar) +
  labs(
    title = "The Relationship Between Housing Value and Household Income",
    x = "Median household income",
    y = "Median home value",
    color = "County"
  ) +
  theme_minimal() +
  scale_color_manual(values = c("Alameda" = "#E41A1C",
                               "Santa Clara" = "#377EB8",
                               "Allegheny" = "#4DAF4A"))
```

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



MB.Ch1.11.

Run the following code.Explain the output from the successive uses of table().

创建了一个包含91个”female”和92个”male”的因子向量默认情况下：水平(levels)按字母顺序排序：“female”在前，“male”在后,table()统计了每个水平的真实数量.

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

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

使用levels参数显式指定新顺序：“male”在前，“female”在后table()输出：遵循新的水平顺序（先显示”male”）, 统计数量不变，但顺序改变了

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

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

将”male”改为”Male”所有原”male”值（92个）不匹配新水平，转为NA原”female”值（91个）匹配新水平”female”新水平”Male”无匹配数据，显示为0

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

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

exclude=NULL参数：强制包含所有NA值输出包含三个部分：“Male”：无匹配数据（0个）“female”：原female值（91个）：原male值转换的缺失值（92个）

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

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

清除 gender

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

MB.Ch1.12.

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

```r
#' @param x 输入数值向量
#' @@param cutoff 截止值
#' @return 超过截止值的元素比例
exceeding_proportion <- function(x, cutoff) {
  if (length(x) == 0) return(0)  # 处理空向量情况
  sum(x > cutoff) / length(x)
}
```

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

```r
# 测试各种截止值情况
test_vector <- 1:100

# 当截止值 =0（所有元素）
exceeding_proportion(test_vector, 0)   # 预期：1.0
```

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

```r
# 当截止值 =50（50 个元素）
exceeding_proportion(test_vector, 50)  # 预期：0.5
```

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

```r
# 当截止值 =100（无元素）
exceeding_proportion(test_vector, 100) # 预期：0.0
```

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

```
# 当截止值 =101（无元素）
exceeding_proportion(test_vector, 101) # 预期：0.0
```

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

```
# 边缘情况测试
exceeding_proportion(numeric(0), 5)     # 空向量返回 0
```

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

```
exceeding_proportion(c(5, 5, 5), 5)     # 严格大于，返回 0
```

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

(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.

```
library(Devore7)
data("ex01.36", package = "Devore7")
escape_times_seconds <- ex01.36[[1]]

cutoff_seconds <- 420  # 7 分钟转换为秒

# 计算超过 420 秒的比例
prop_over_420sec <- exceeding_proportion(escape_times_seconds, cutoff_seconds)
count_above <- sum(escape_times_seconds > cutoff_seconds)
n_total <- length(escape_times_seconds)

cat("撤离时间分析结果(秒单位):\n")
```

```
## 撤离时间分析结果(秒单位):
```

```
cat("总观测数:", n_total, "\n")
```

```
## 总观测数: 26
```

```r
cat("超过420秒的观测数:", count_above, "\n")
```

```
## 超过420秒的观测数: 1
```

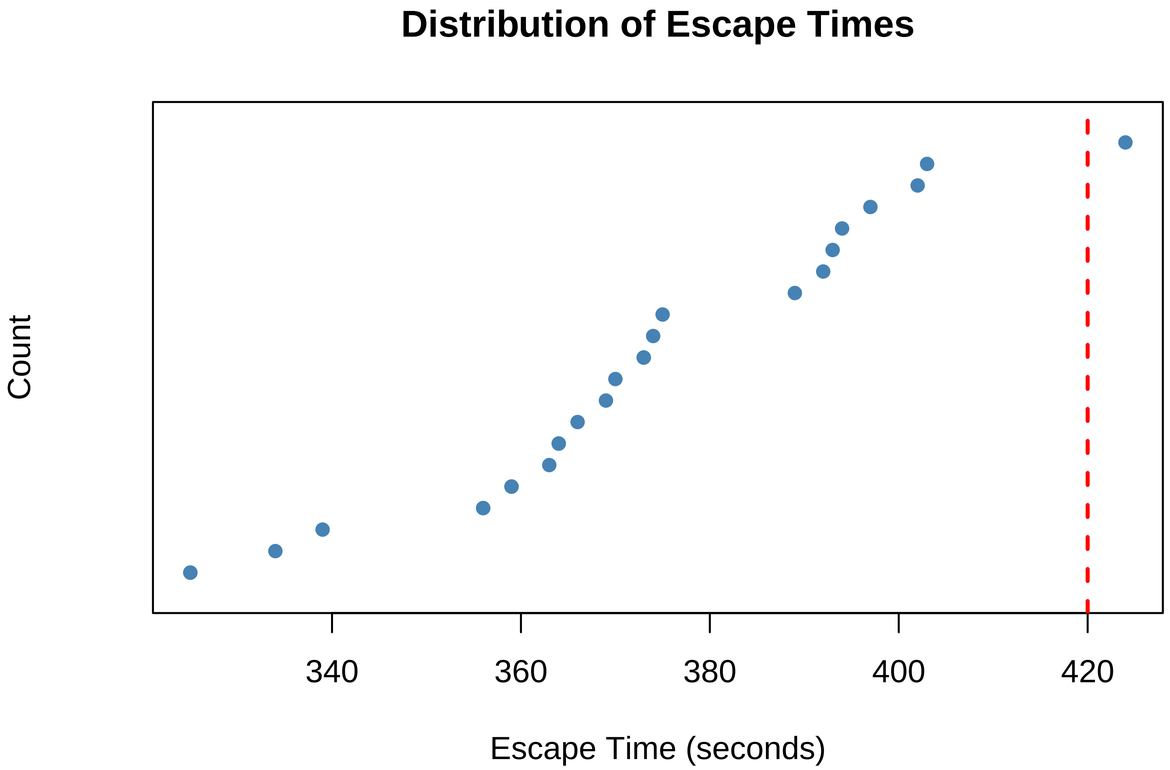
```r
cat("超过420秒的比例:", round(prop_over_420sec, 4), "\n")
```

```
## 超过420秒的比例: 0.0385
```

```r
cat("分数表示:", count_above, "/", n_total, "\n\n")
```

```
## 分数表示: 1 / 26
```

```r
custom_dotplot <- function(x, cutoff, unit = "seconds") {

  unique_vals <- sort(unique(round(x, 2)))
  y_pos <- seq_along(unique_vals)

  y_vals <- sapply(x, function(val) {
    match_idx <- which.min(abs(unique_vals - val))
    y_pos[match_idx]
  })

  plot(1, type = "n",
       xlim = range(x),
       ylim = c(0, max(y_pos) + 1),
       main = "Distribution of Escape Times",
       xlab = paste("Escape Time (", unit, ")", sep = ""),
       ylab = "Count", yaxt = "n")

  points(x, y_vals, pch = 16, col = "steelblue")

  abline(v = cutoff, col = "red", lty = 2, lwd = 2)

}
custom_dotplot(escape_times_seconds, cutoff_seconds, "seconds")
```



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)

final_result <- aggregate(BPchange ~ Treatment + Dose + Animal, data = Rabbit, FUN = mean)

final_result_wide <- reshape(final_result,
                             timevar = "Animal",
                             idvar = c("Treatment", "Dose"),
                             direction = "wide")

rabbit_cols <- grep("BPchange.", colnames(final_result_wide), value = TRUE)
```

```
rabbit_names <- gsub("BPchange.", "", rabbit_cols)

required_columns <- c("Treatment", "Dose", rabbit_cols)
final_result_wide <- final_result_wide[, required_columns]

colnames(final_result_wide) <- c("Treatment", "Dose", rabbit_names)

final_result_wide <- final_result_wide[order(final_result_wide$Treatment), ]

print(final_result_wide)
```

```
##    Treatment   Dose    R1    R2    R3    R4   R5
## 1    Control   6.25  0.50  1.00  0.75  1.25  1.5
## 3    Control  12.50  4.50  1.25  3.00  1.50  1.5
## 5    Control  25.00 10.00  4.00  3.00  6.00  5.0
## 7    Control  50.00 26.00 12.00 14.00 19.00 16.0
## 9    Control 100.00 37.00 27.00 22.00 33.00 20.0
## 11   Control 200.00 32.00 29.00 24.00 33.00 18.0
## 2        MDL   6.25  1.25  1.40  0.75  2.60  2.4
## 4        MDL  12.50  0.75  1.70  2.30  1.20  2.5
## 6        MDL  25.00  4.00  1.00  3.00  2.00  1.5
## 8        MDL  50.00  9.00  2.00  5.00  3.00  2.0
## 10       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
```