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", sep = ',', header=T)

dim_info <- dim(ca_pa)
paste("Rows:", dim_info[1], "Columns:", dim_info[2])
```

```
## [1] "Rows: 11275 Columns: 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
```

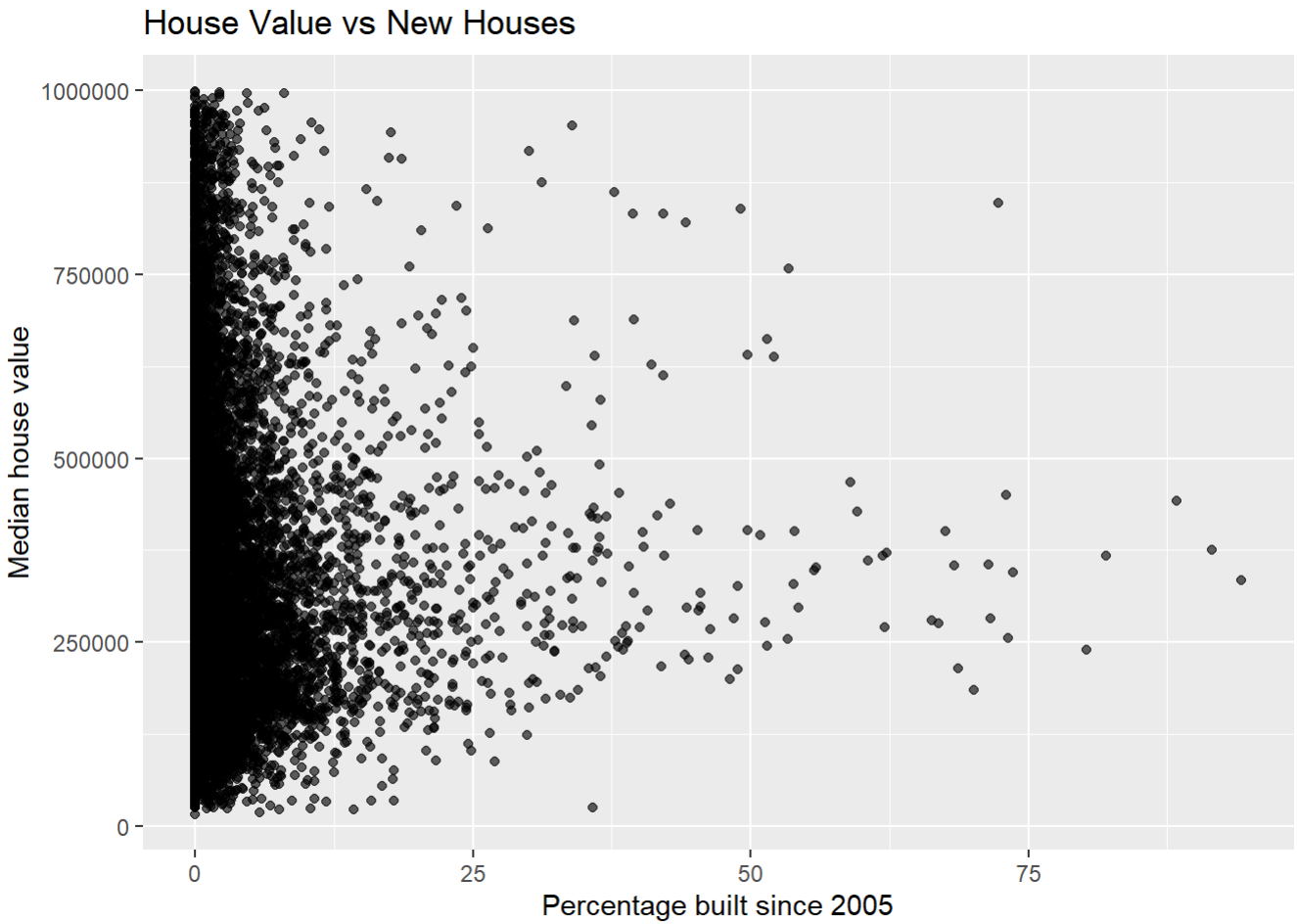
```
ca_pa_clean <- na.omit(ca_pa)

rows_eliminated <- nrow(ca_pa) - nrow(ca_pa_clean)
paste("Rows eliminated:", rows_eliminated)
```

```
## [1] "Rows eliminated: 670"
```

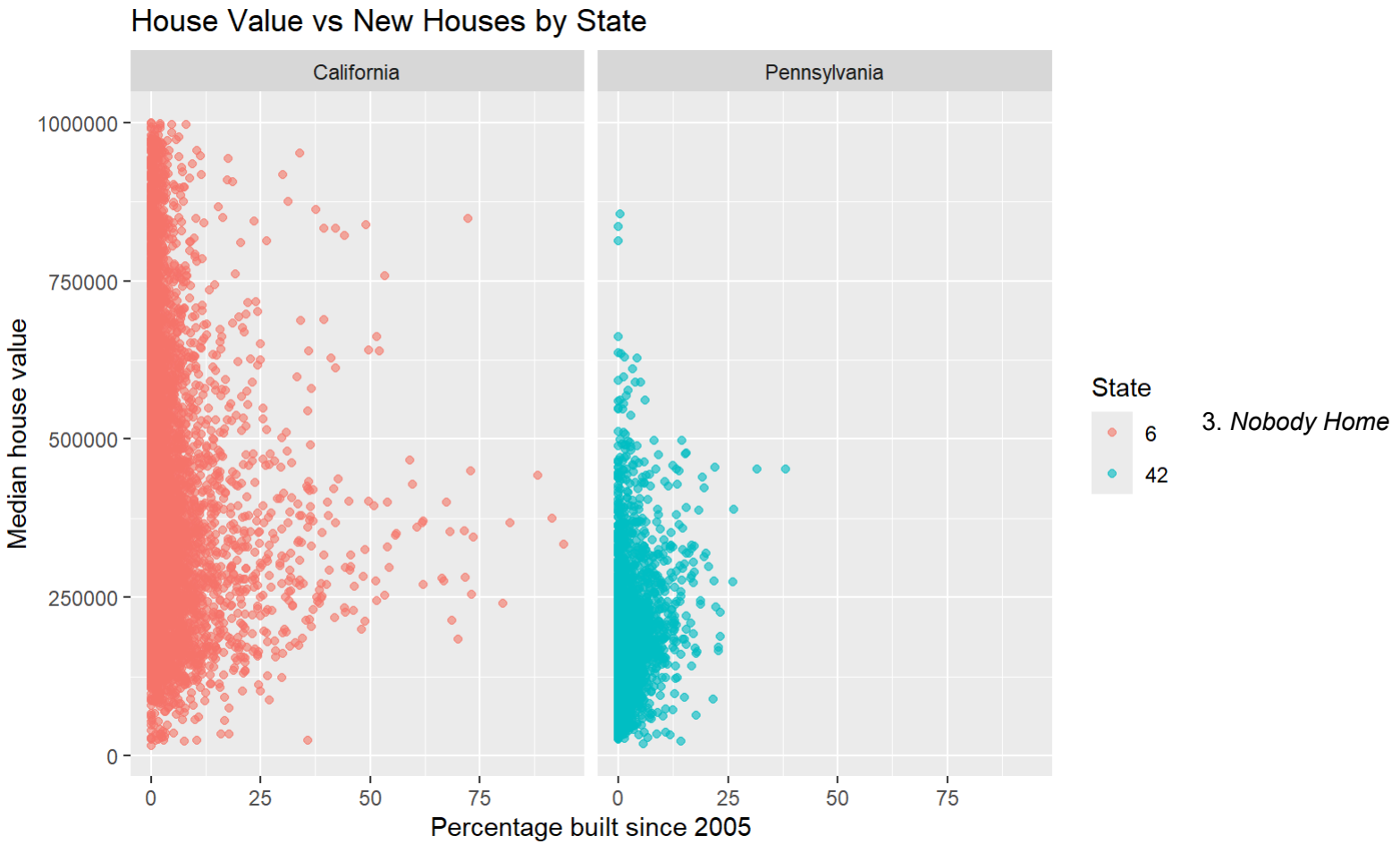
#c This command calculates the number of missing values (NA) in each column of the dataframe. apply(ca_pa, c(1,2), is.na) converts each element to TRUE if it's NA, FALSE otherwise colSums() then sums the TRUE values (NA count) column-wise #f Yes, they are compatible. The value in (e) should equal the number of rows with at least one NA (missing value), while the sum of values in (c) counts all missing values across all cells. Since a row may have multiple NAs, the sum in (c) will typically be larger than (e), but both reflect the presence of missing data.

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.

```
ggplot(ca_pa_clean, aes(x = Built_2005_or_later, y = Median_house_value)) +
  geom_point(alpha = 0.6) +
  labs(x = "Percentage built since 2005", y = "Median house value",
       title = "House Value vs New Houses")
```




```
ggplot(ca_pa_clean, aes(x = Built_2005_or_later, y = Median_house_value, color = factor(STATEFP))) +
  geom_point(alpha = 0.6) +
  facet_wrap(~ STATEFP, labeller = labeller(STATEFP = c("6" = "California", "42" = "Pennsylvania"))) +
  labs(x = "Percentage built since 2005", y = "Median house value",
       title = "House Value vs New Houses by State", color = "State")
```



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.

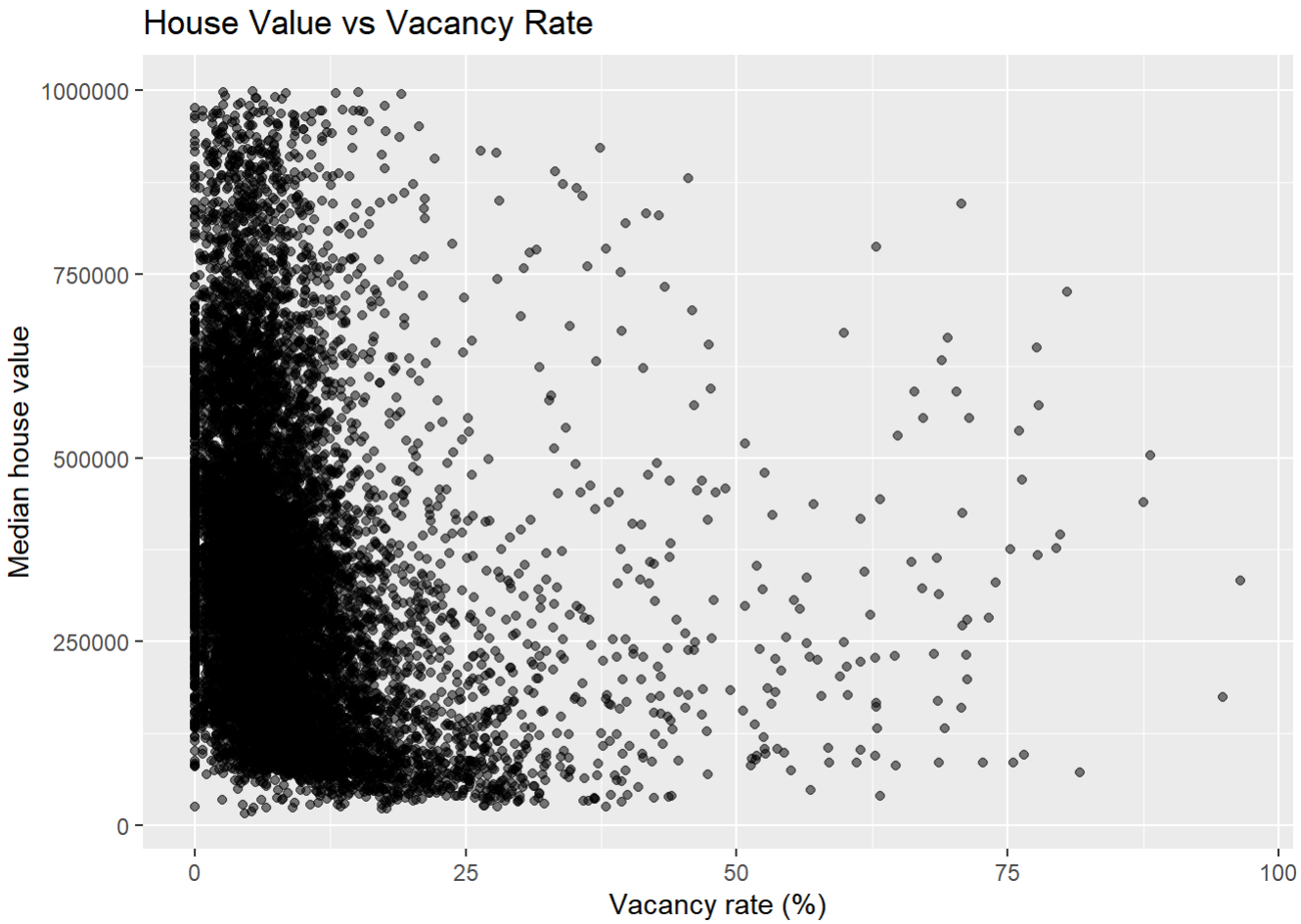
a. Add a new column to the dataframe which contains the vacancy rate. What are the minimum, maximum, mean, and median vacancy rates? b. Plot the vacancy rate against median house value. c. Plot vacancy rate against median house value separately for California and for Pennsylvania. Is there a difference?

```
ca_pa_clean <- ca_pa_clean %>%
  mutate(vacancy_rate = (Vacant_units / Total_units) * 100)

vacancy_stats <- ca_pa_clean %>%
  summarize(
    min = min(vacancy_rate),
    max = max(vacancy_rate),
    mean = mean(vacancy_rate),
    median = median(vacancy_rate)
  )
vacancy_stats
```

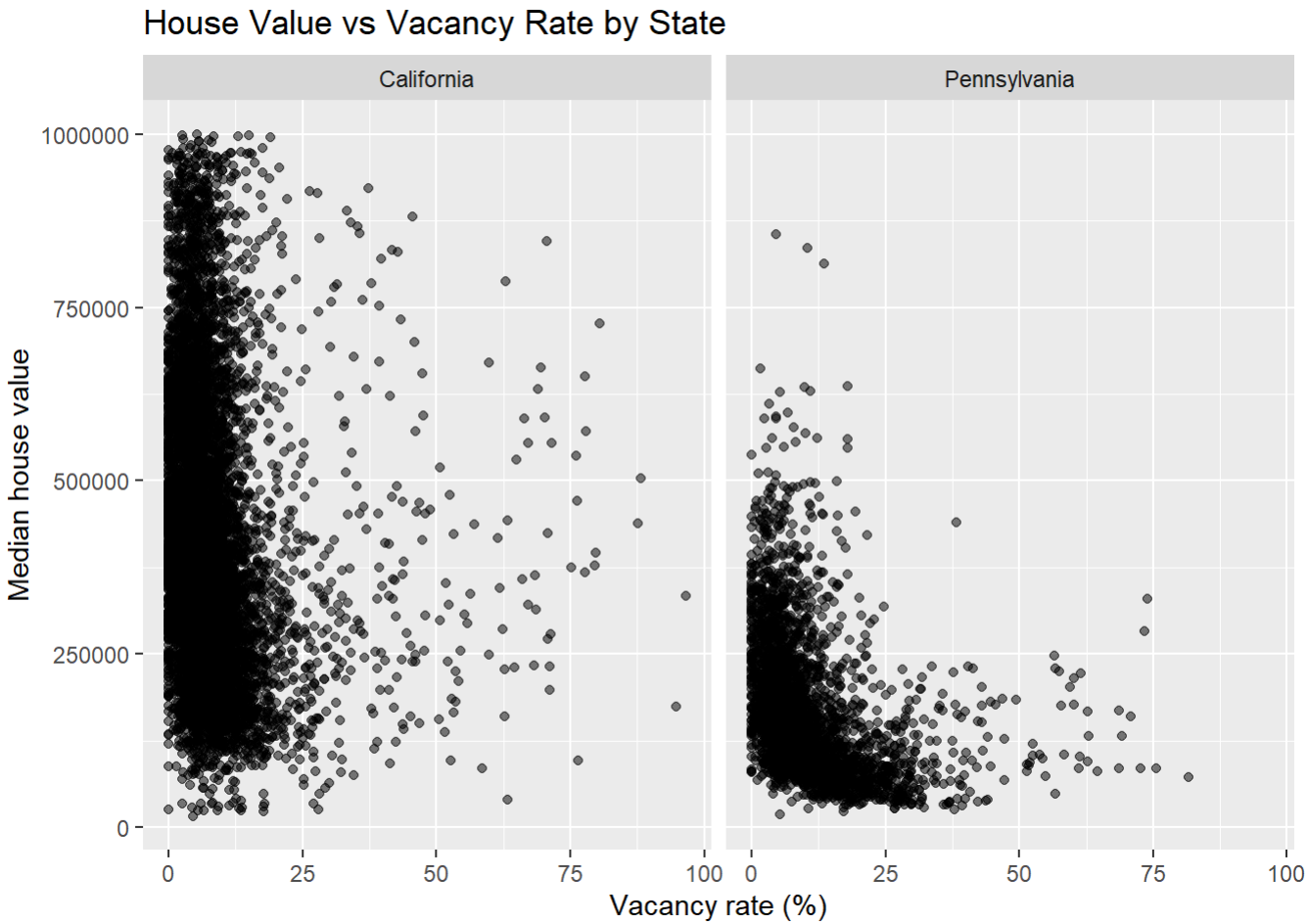
min <dbl>	max <dbl>	mean <dbl>	median <dbl>
0	96.5311	8.888789	6.767283

1 row

```
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",
       title = "House Value vs Vacancy Rate")
```




```
ggplot(ca_pa_clean, aes(x = vacancy_rate, y = Median_house_value)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~ STATEFP, labeller = labeller(STATEFP = c("6" = "California", "42" = "Pennsylvania"))) +
  labs(x = "Vacancy rate (%)", y = "Median house value",
       title = "House Value vs Vacancy Rate by State")
```



There appears to be a difference between states. California shows a tighter cluster of high-value properties with low vacancy rates, while Pennsylvania has more variability and generally lower house values.

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. The code calculates the median house value (column 10) for Alameda County (CA) by: Looping through all tracts to identify those in California (STATEFP=6) and Alameda County (COUNTYFP=1) Storing qualifying tract indices in acca Extracting Median_house_value for these tracts into accamhv Calculating the median of these values

    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.

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

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

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

```
ca_pa_clean %>%
  filter(
    (STATEFP == 6 & COUNTYFP == 1) |  # Alameda
    (STATEFP == 6 & COUNTYFP == 85) | # Santa Clara
    (STATEFP == 42 & COUNTYFP == 3)   # Allegheny
  ) %>%
  group_by(COUNTYFP, STATEFP) %>%
  summarize(avg_new_housing = mean(Built_2005_or_later))
```

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

COUNTYFP <int>	STATEFP <int>	avg_new_housing <dbl>
1	6	2.820468
3	42	1.474219
85	6	3.200319

3 rows

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?

```
# Whole data
cor(ca_pa_clean$Median_house_value, ca_pa_clean$Built_2005_or_later)
```

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

```
# California
ca_data <- filter(ca_pa_clean, STATEFP == 6)
cor(ca_data$Median_house_value, ca_data$Built_2005_or_later)
```

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

```
# Pennsylvania
pa_data <- filter(ca_pa_clean, STATEFP == 42)
cor(pa_data$Median_house_value, pa_data$Built_2005_or_later)
```

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

```
# Alameda
alameda <- filter(ca_pa_clean, STATEFP == 6, COUNTYFP == 1)
cor(alameda$Median_house_value, alameda$Built_2005_or_later)
```

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

```
# Santa Clara
santa_clara <- filter(ca_pa_clean, STATEFP == 6, COUNTYFP == 85)
cor(santa_clara$Median_house_value, santa_clara$Built_2005_or_later)
```

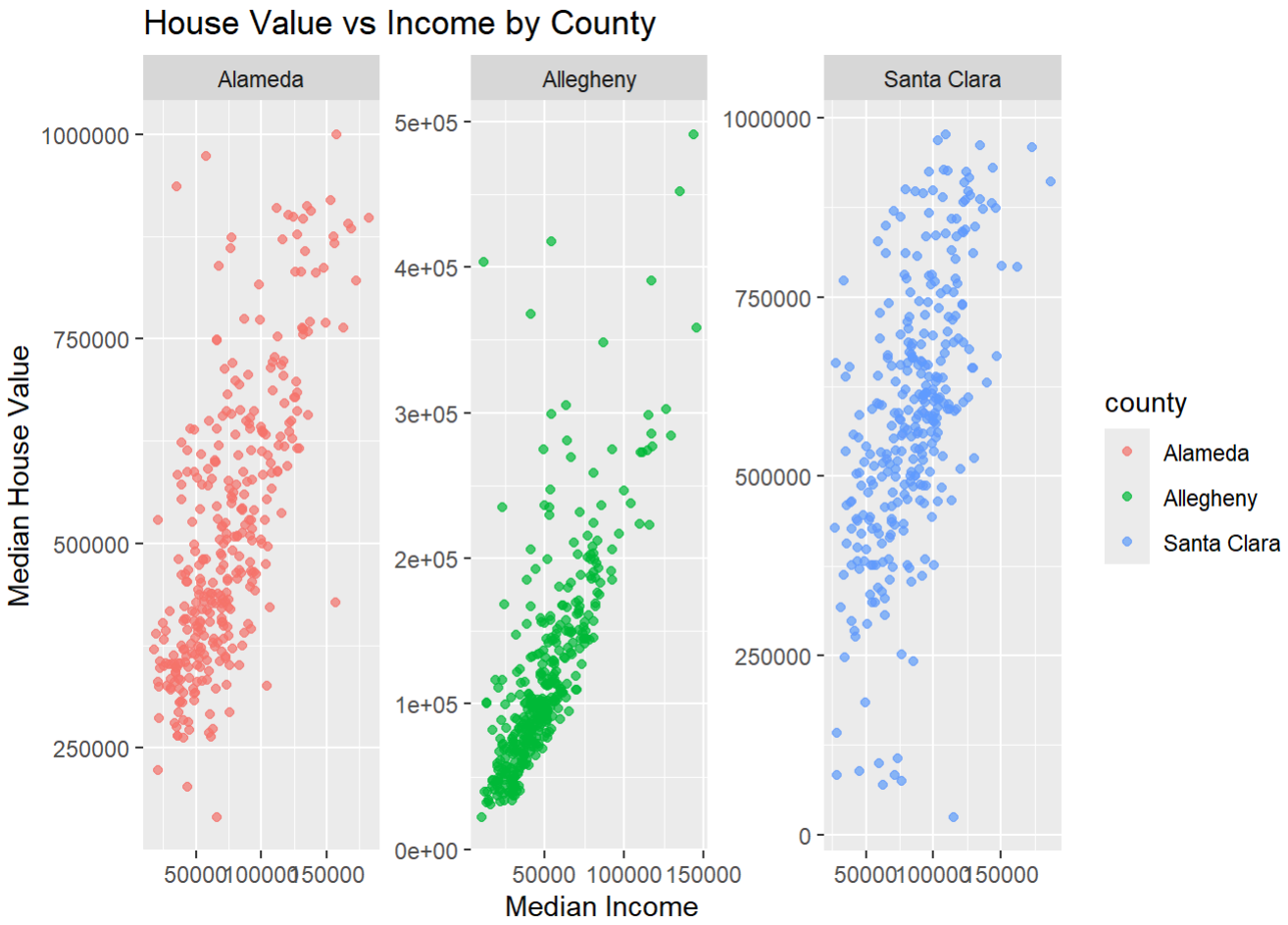
```
## [1] -0.1726203
```

```
# Allegheny
allegheny <- filter(ca_pa_clean, STATEFP == 42, COUNTYFP == 3)
cor(allegheny$Median_house_value, allegheny$Built_2005_or_later)
```

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

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.)

```
county_data <- ca_pa_clean %>%
  filter(
    (STATEFP == 6 & COUNTYFP == 1) |  # Alameda
    (STATEFP == 6 & COUNTYFP == 85) | # Santa Clara
    (STATEFP == 42 & COUNTYFP == 3)   # Allegheny
  ) %>%
  mutate(county = case_when(
    COUNTYFP == 1 ~ "Alameda",
    COUNTYFP == 85 ~ "Santa Clara",
    COUNTYFP == 3 ~ "Allegheny"
  ))

ggplot(county_data, aes(x = Median_household_income, y = Median_house_value, color = county)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~ county, scales = "free") +
  labs(x = "Median Income", y = "Median House Value",
       title = "House Value vs Income by County")
```


House Value vs Income by County
Alameda
Allegheny
Santa Clara
Median House Value
Median Income
1000000
750000
500000
250000
5e+05
4e+05
3e+05
2e+05
1e+05
0e+00
0
50000
100000
150000
county
Alameda
Allegheny
Santa Clara

```
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)
```

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"))
table(gender)
```

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

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

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

```
rm(gender)
```

Explain the output from the successive uses of table(). First table(): Counts factor levels as created (91 female, 92 male) Second table(): Same counts but with level order reversed (male first) Third table(): Shows 0 for “Male” because original “male” doesn’t match new level “Male” (case mismatch) Fourth table(exclude=NULL): Includes NA counts (92) from mismatched “male” values

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.

```
prop_above <- function(x, cutoff) {
  mean(x > cutoff, na.rm = TRUE)
}
prop_above(1:100, 50)
```

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

```
prop_above(1:100, 90)
```

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

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)
```

```
## 载入需要的程序包：MASS
```

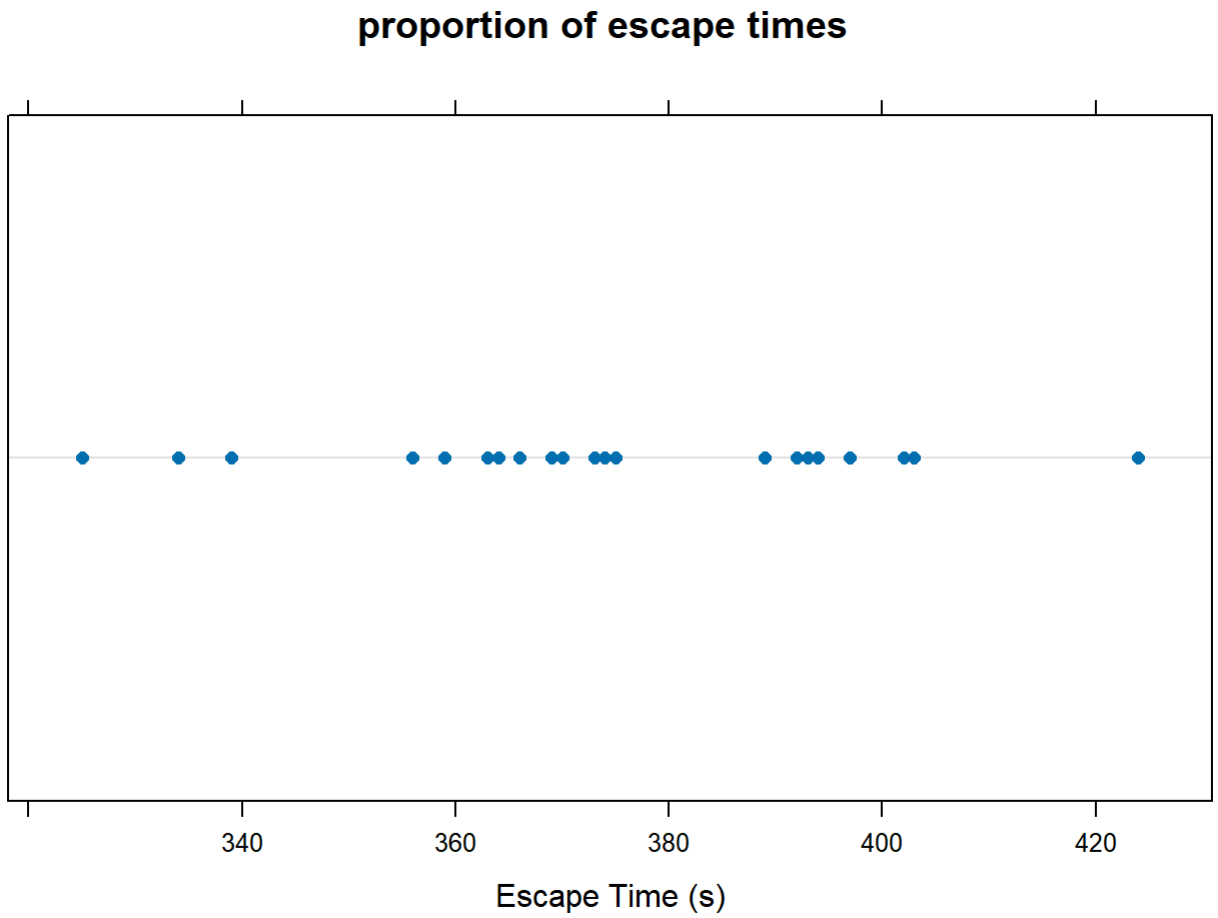
```
##
## 载入程序包：'MASS'
```

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

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

```
## 载入需要的程序包：lattice
```

```
data(ex01.36)
prop_above <- function(x, threshold) {
  mean(x > threshold)
}
library(lattice)
dotplot(~ex01.36, xlab = "Escape Time (s)", main = "proportion of escape times")
```




```
prop_above(ex01.36,420)
```

```
## [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)

Rabbit <- Rabbit %>%
  mutate(group = interaction(Treatment, Dose))

Rabbit_wide <- do.call(cbind, lapply(split(Rabbit, Rabbit$Animal),
                                     function(df) unstack(df, BPchange ~ group)))

Rabbit_final <- data.frame(
  Treatment = sapply(strsplit(rownames(Rabbit_wide), "\\."), `[`, 1),
  Dose = as.numeric(sapply(strsplit(rownames(Rabbit_wide), "\\."), `[`, 2)),
  Rabbit_wide
) %>% `rownames<-`(NULL)

head(Rabbit_final, 4)
```

	Treatment <chr>	Dose <dbl>	res <dbl>	res.1 <dbl>	res.2 <dbl>	res.3 <dbl>	res.4 <dbl>
1	Control	6	0.50	1.00	0.75	1.25	1.5
2	MDL	6	1.25	1.40	0.75	2.60	2.4
3	Control	12	4.50	1.25	3.00	1.50	1.5
4	MDL	12	0.75	1.70	2.30	1.20	2.5

4 rows