Homework 2

Xinran Ren, 3220103492

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`.

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

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

```r
dim(ca_pa)
```

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

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

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

- This command does the following:
  - apply `(ca_pa, c(1,2), is.na)` creates a TRUE/FALSE matrix where:
    * TRUE indicates a missing value (NA)
    * FALSE indicates valid data
  - `colSums()` sums the TRUE values (counted as 1) column-wise
- Output: A vector showing the number of missing values in each column.

   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.

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

   e. How many rows did this eliminate?

```r
rows_eliminated <- nrow(ca_pa) - nrow(ca_pa_clean)
rows_eliminated
```

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

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

- Yes, the answers are compatible. Here's why:

- Let T = sum of all column-wise NA counts from (c)($i.e.$Total NAs in dataset)
- Let R = rows eliminated from (e)($i.e.$Rows with ≥ 1 NA)
- Relationship: T ≥ R because:
  - Each removed row has ≥ 1 NA
- Equality T = R holds only if:
  - Every removed row has exactly 1 NA (no row has multiple NAs).

- Verification:

```
total_nas <- sum(colSums(is.na(ca_pa)))  # T from (c)
rows_eliminated <- nrow(ca_pa) - nrow(ca_pa_clean)  # R from (e)
total_nas >= rows_eliminated  # Always TRUE
```

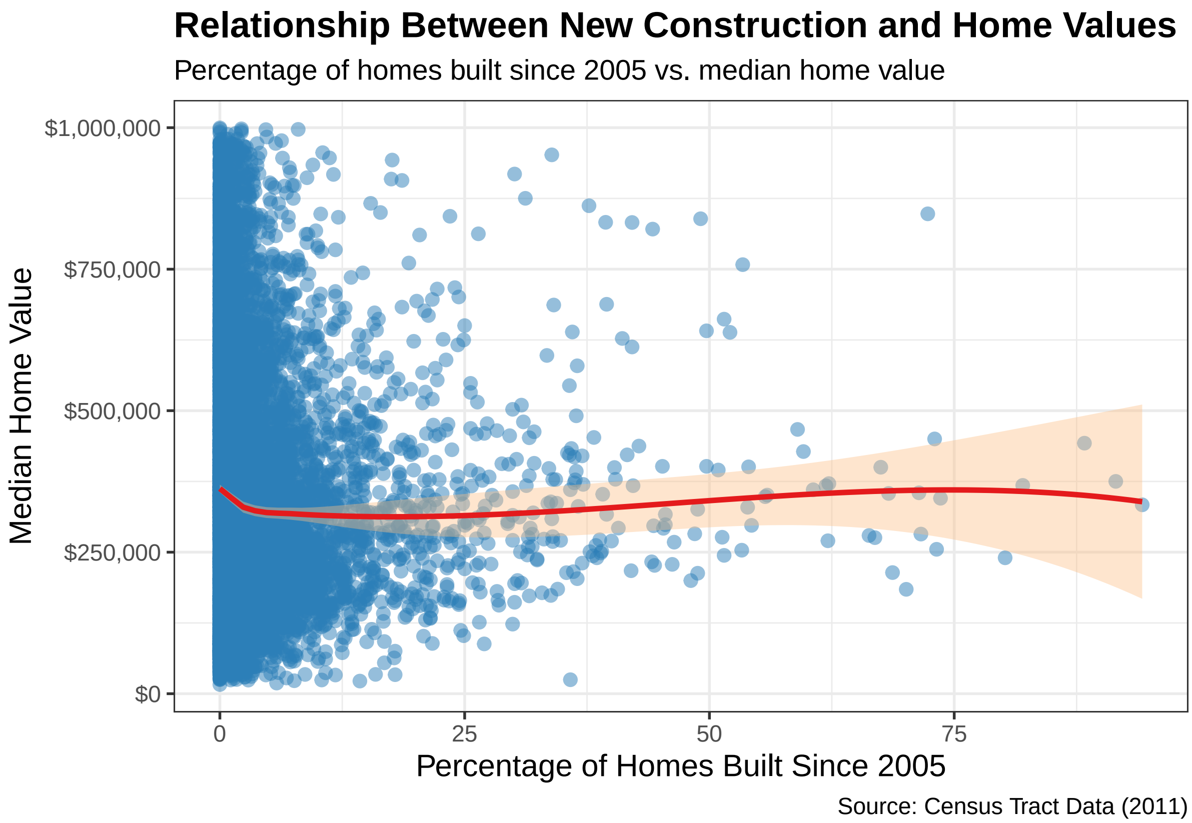
```
## [1] TRUE
```

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.

```
p1 <- ggplot(ca_pa_clean, aes(x = Built_2005_or_later, y = Median_house_value)) +
  geom_point(alpha = 0.5, color = "#2c7fb8", size = 2) +
  geom_smooth(method = "loess", color = "#e41a1c", se = TRUE, fill = "#fdc086") +
  scale_y_continuous(labels = scales::dollar) +  # Format y-axis as currency
  labs(title = "Relationship Between New Construction and Home Values",
   subtitle = "Percentage of homes built since 2005 vs. median home value",
   x = "Percentage of Homes Built Since 2005",
   y = "Median Home Value",
   caption = "Source: Census Tract Data (2011)") +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 14),
    axis.title = element_text(size = 12))

print(p1)
```

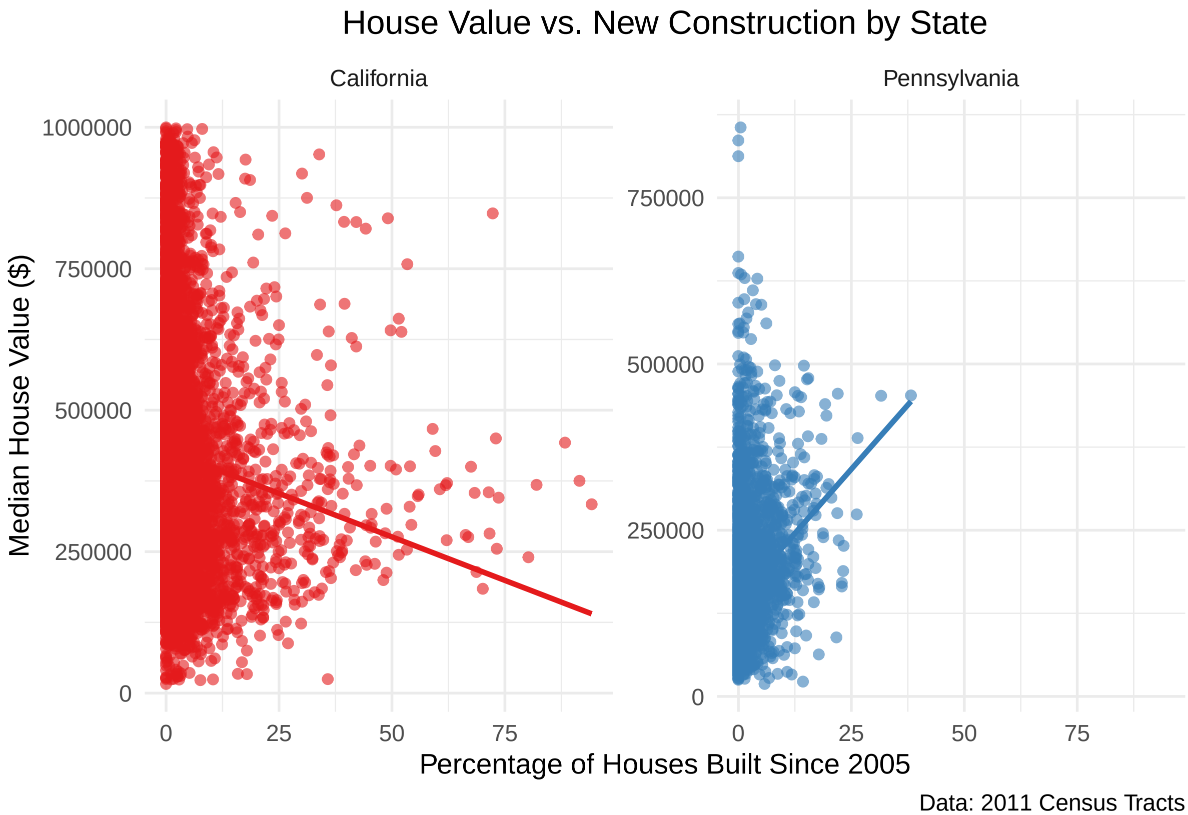
```
## `geom_smooth()` using formula = 'y ~ x'
```



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.

```
# Create state variable
ca_pa_clean$State <- ifelse(ca_pa_clean$STATEFP == 6, "California", "Pennsylvania")

# Create faceted plot with regression lines
p2 <- ggplot(ca_pa_clean, aes(x = Built_2005_or_later, y = Median_house_value)) +
  geom_point(aes(color = State), alpha = 0.6) +
  geom_smooth(aes(color = State), method = "lm", se = FALSE) +
  scale_color_manual(values = c("California" = "#E41A1C",
                                "Pennsylvania" = "#377EB8")) +
  facet_wrap(~ State, scales = "free_y") +
  labs(title = "House Value vs. New Construction by State",
   x = "Percentage of Houses Built Since 2005",
   y = "Median House Value ($)",
   caption = "Data: 2011 Census Tracts") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
    legend.position = "none")

print(p2)
```

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



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?

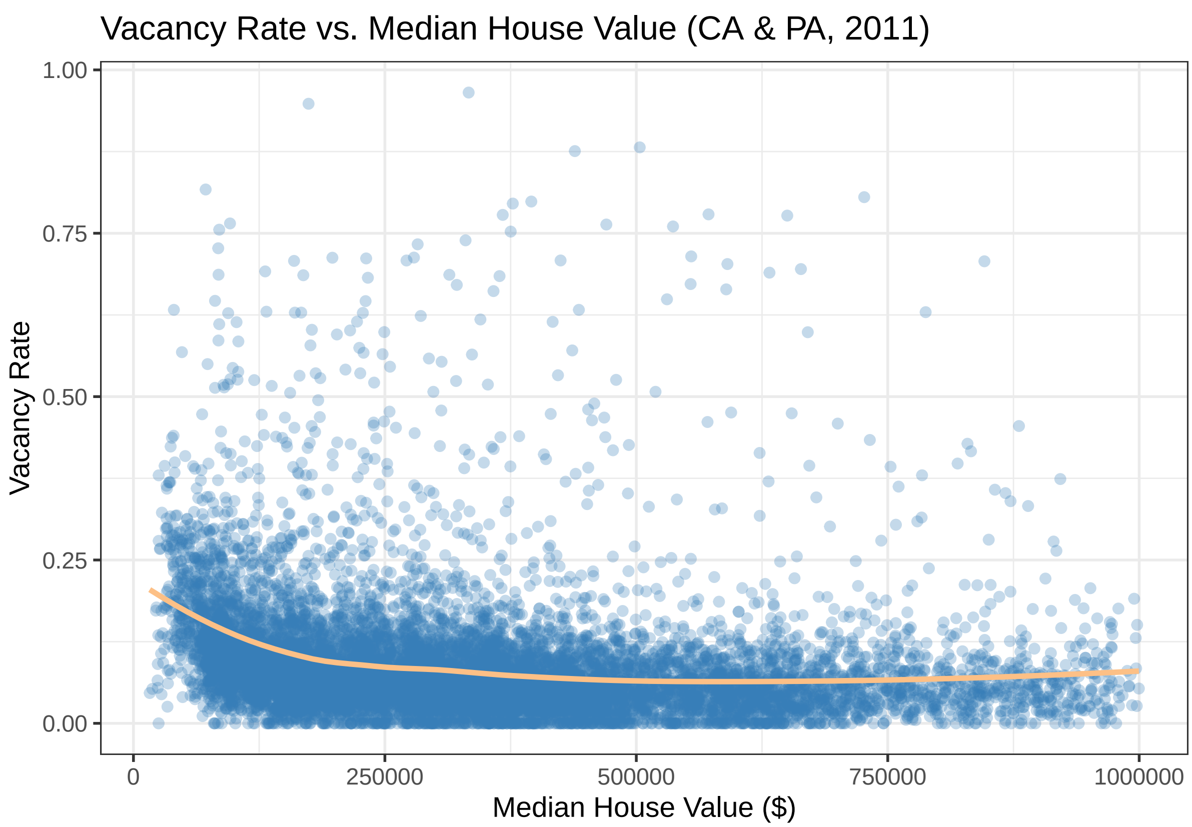
```r
# Add the vacancy rate column:
ca_pa_clean <- ca_pa_clean %>%
  mutate(vacancy_rate = Vacant_units / Total_units)
# Compute summary statistics:
summary_stats <- ca_pa_clean %>%
  summarise(
min = min(vacancy_rate, na.rm = TRUE),
max = max(vacancy_rate, na.rm = TRUE),
mean = mean(vacancy_rate, na.rm = TRUE),
median = median(vacancy_rate, na.rm = TRUE)
  )
print(summary_stats)
```

```
##   min      max       mean     median
## 1   0 0.965311 0.08888789 0.06767283
```

b. Plot the vacancy rate against median house value.

```
ggplot(ca_pa_clean, aes(x = Median_house_value, y = vacancy_rate)) +
geom_point(alpha = 0.3, color = "#377EB8") +
geom_smooth(method = "loess", color = "#FDC086", se = FALSE) +
labs(
  x = "Median House Value ($)",
  y = "Vacancy Rate",
  title = "Vacancy Rate vs. Median House Value (CA & PA, 2011)"
) +
theme_bw()
```

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

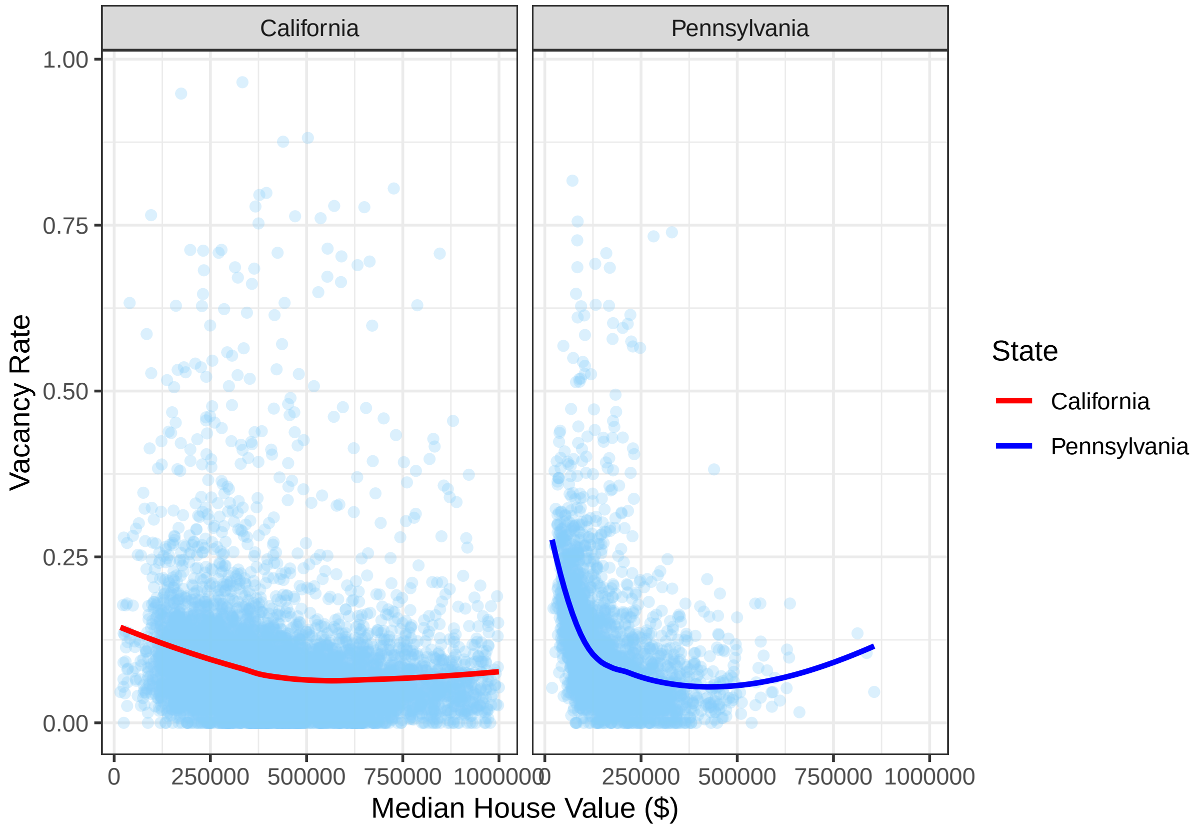


c. Plot vacancy rate against median house value separately for California and for Pennsylvania. Is there a difference?

```
# Convert state codes to factor
ca_pa_clean <- ca_pa_clean %>%
  mutate(State = as.factor(ifelse(STATEFP == 6, "California", "Pennsylvania")))

# Plot with distinct elements
ggplot(ca_pa_clean, aes(x = Median_house_value, y = vacancy_rate)) +
  geom_point(color = "#87CEFA", alpha = 0.3) +
```

```
geom_smooth(aes(color = State), method = "loess", se = FALSE, linewidth = 1) +
scale_color_manual(values = c("California" = "red", "Pennsylvania" = "blue")) +
facet_wrap(~State) +  # Separate panels
labs(x = "Median House Value ($)", y = "Vacancy Rate") +
theme_bw()
```

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



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:
     - Identifies all census tracts in Alameda County, California (where STATEFP == 6 and COUNTYFP == 1) and stores their row indices in acca.
     - Extracts the median house values (from column 10) for these tracts into accamhv.
     - Computes the median of these values.
   - Purpose: Calculate the median median house value for Alameda County tracts.

   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.

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

```
## NULL
```

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)
```

MB.Ch1.11. Run the following code:

```r
# Creates a factor with 91 "female" and 92 "male" (total 183 observations).
# Default factor levels are ordered alphabetically: "female", then "male".
# table() counts observations per level → 91 females, 92 males.
gender <- factor(c(rep("female", 91), rep("male", 92)))
table(gender)
```

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

```r
# Reassigns the factor with new level order: male first, then female.
# Data remains unchanged (still 92 males and 91 females).
# table() now displays counts in the new level order: males first (92), then females (91).
gender <- factor(gender, levels=c("male", "female"))
table(gender)
```

```
## gender
```

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

```
# Reassigns factor with new levels: "Male" (capital "M") and "female".
# Critical error: "Male" not = "male" (R is case-sensitive).
# Original "male" values don't match any level: converted to NA.
gender <- factor(gender, levels=c("Male", "female"))
# Note the mistake: "Male" should be "male"
table(gender)
```

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

```
#exclude = NULL forces table() to include NA values.
table(gender, exclude=NULL)
```

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

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

Explain the output from the successive uses of `table()`.

- We answer it by "#".

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_above <- function(x, cutoff, na.rm = FALSE) {
  mean(x > cutoff, na.rm = na.rm)
}

# Test with sequence 1:100
test_vector <- 1:100
proportion_above(test_vector, 50)  # Should return 0.5 (50 values > 50)
```

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

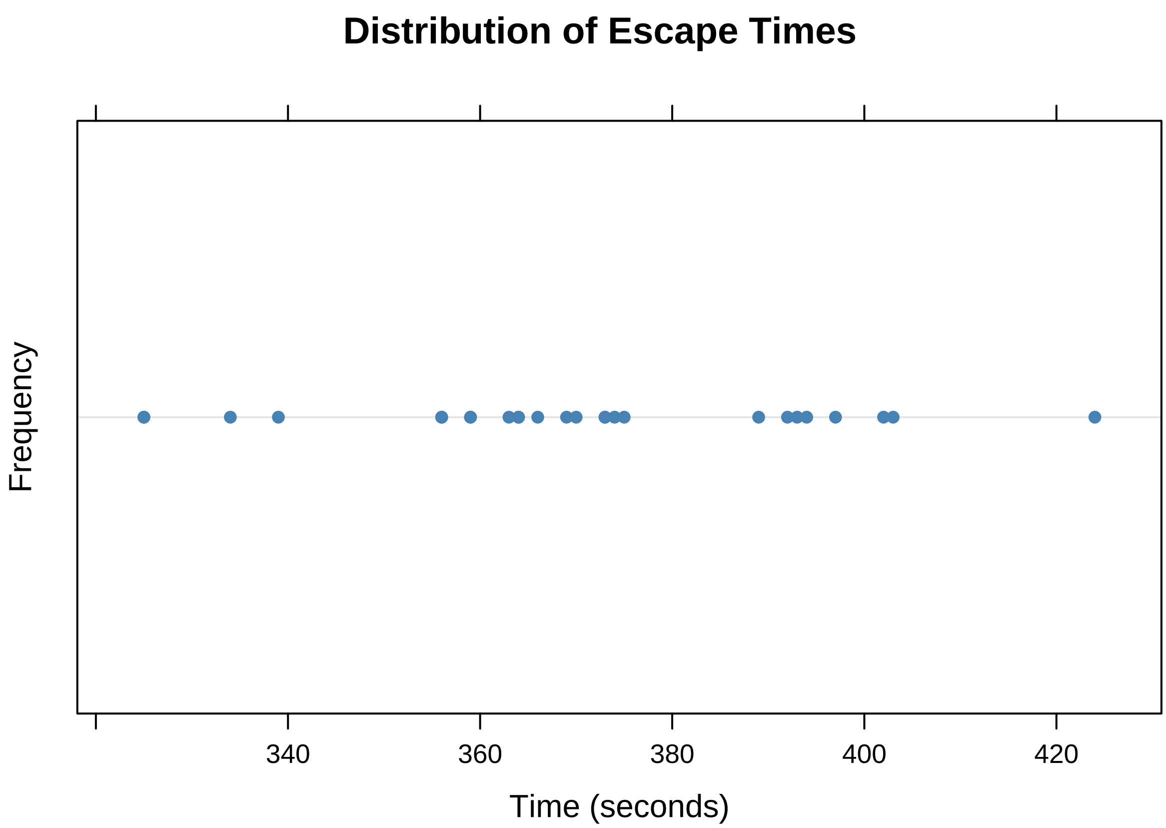
(b) Obtain the vector `ex01.36` from the Devore6 (or Devore7) package. These data give the times required for individuals to escape from an oil platform during a drill. Use `dotplot()` to show the distribution of times. Calculate the proportion of escape times that exceed 7 minutes.

```
#install.packages("Devore7")  # Only needed once
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
```

```r
# Load escape times data
data(ex01.36)

# Create the Dot Plot
dotplot(~ex01.36,
        xlab = "Time (seconds)",
        ylab = "Frequency",
        main = "Distribution of Escape Times",
        pch = 16,
        col = "steelblue")
```



```r
# Calculate the proportion of escape times that exceed 7 minutes
proportion_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

....

```r
library(MASS)
result <- local({
  bp <- unstack(Rabbit, BPchange ~ Animal)
  dose <- unstack(Rabbit, Dose ~ Animal)$R1
  treatment <- unstack(Rabbit, Treatment ~ Animal)$R1
  unique(data.frame(Treatment = treatment, Dose = dose, bp)[order(treatment, dose), ])
})

rownames(result) <- NULL
result
```

```
##    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
```