Homework 2

Zhang Zile,3230104237

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?

```
ca_pa <-read.csv("data/calif_penn_2011.csv", header=T)[,-1] #remove the column of index
nrow(ca_pa) #11275
```

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

```
ncol(ca_pa) #original 34,after removing 33
```

```
## [1] 33
```

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

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

- Count the number of missing data 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.

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

    e. How many rows did this eliminate?

```
nrow(ca_pa)#11275-10605=670
```

```
## [1] 10605
```

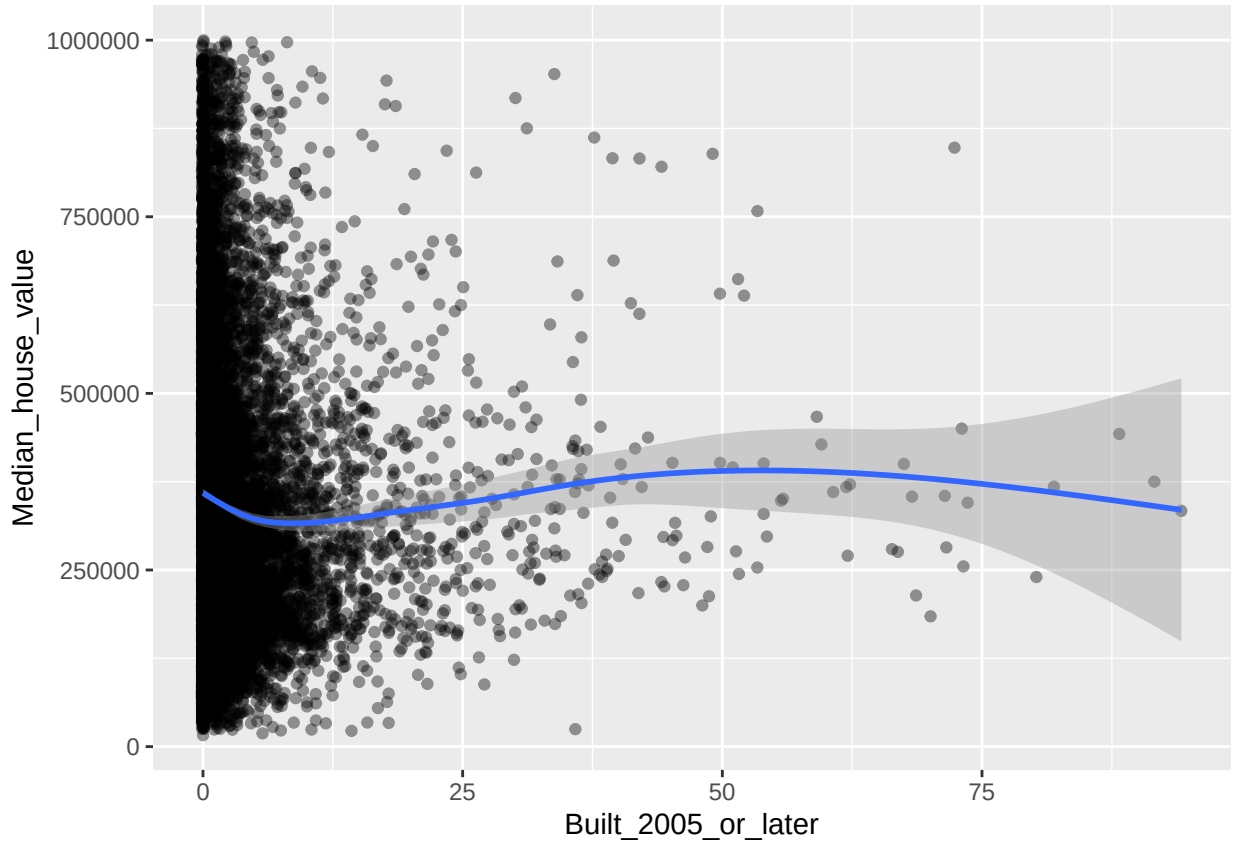
    f. Are your answers in (c) and (e) compatible? Explain.

- Yes. Some rows may have more than one 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.

```
  ggplot(ca_pa,aes(Built_2005_or_later,Median_house_value))+
  geom_jitter(width=0.1,alpha=0.4)+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



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.

```
ca_pa$STATEFP <- factor(ca_pa$STATEFP)
ca_pa %>%
    group_by(Built_2005_or_later, STATEFP) %>%
    summarize(median_house_price = mean(Median_house_value), .groups = 'drop') %>%
    mutate(state = case_when(
        STATEFP == "6" ~ "California",
        STATEFP == "42" ~ "Pennsylvania"
    )) %>%
    ggplot(aes(x = Built_2005_or_later,
               y = median_house_price,
               color = state)) +
    geom_point() +
    geom_smooth(method = "lm", se = FALSE)
```

```
## `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?

```
ca_pa <- ca_pa %>% mutate(vacanty_rate = Vacant_units/Total_units)
summary(ca_pa$vacanty_rate)
```

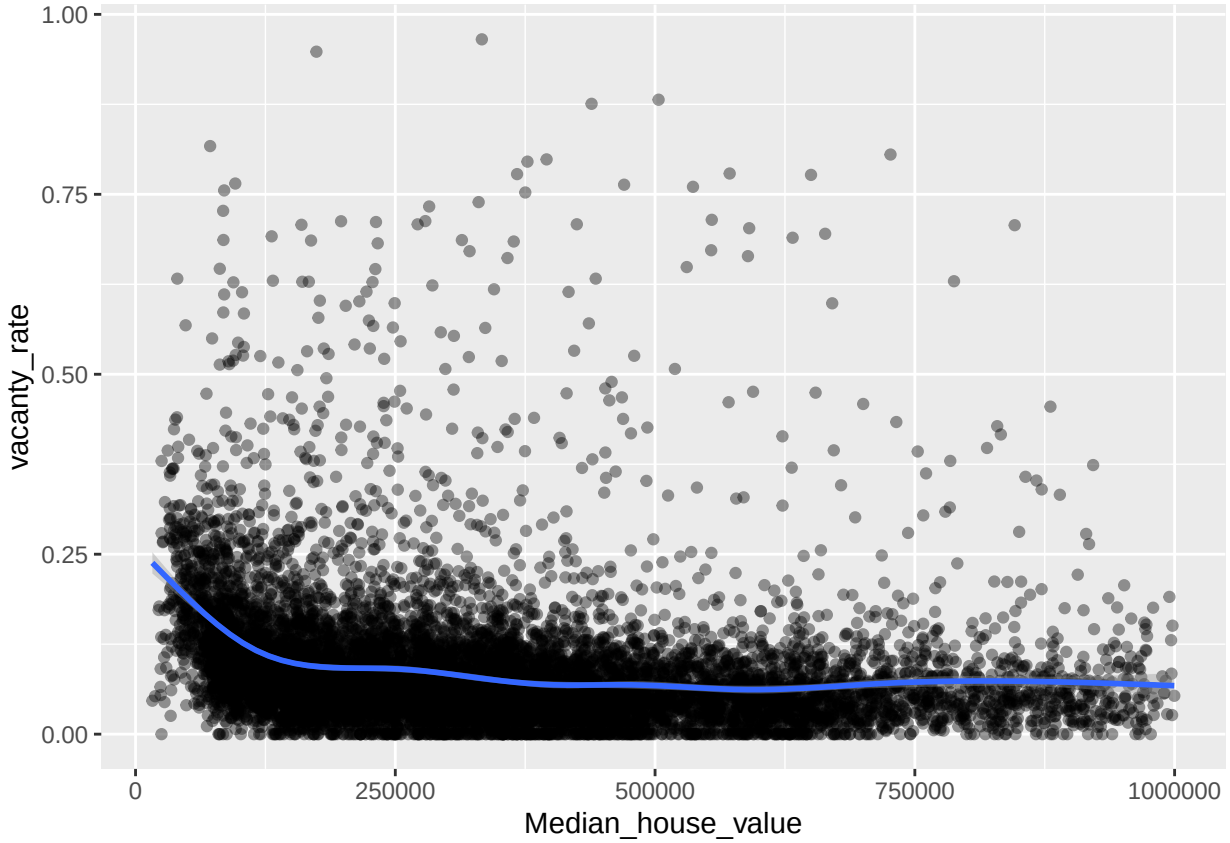
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.03846 0.06767 0.08889 0.10921 0.96531
```

- minimum=0,maximum=0.96531,mean=0.08889,median=0.06767

b. Plot the vacancy rate against median house value.

```
ggplot(ca_pa,aes(Median_house_value,vacanty_rate))+geom_jitter(width=0.1,alpha=0.4)+geom_smooth()
```

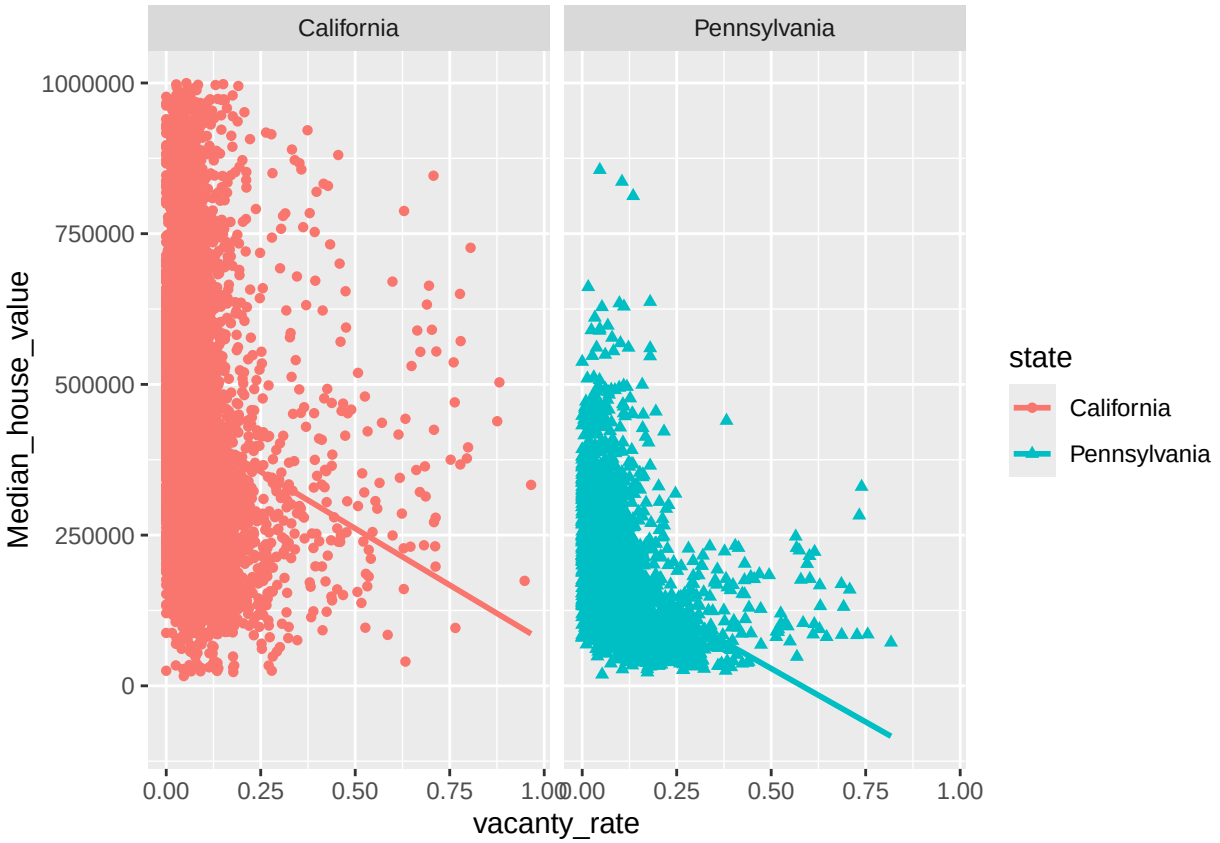
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



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

```
ca_pa %>%
       mutate(state = case_when(
              STATEFP == "6" ~ "California",
              STATEFP == "42" ~ "Pennsylvania")) %>%
      ggplot(aes(x = vacanty_rate,y = Median_house_value,color = state,shape=state)) +
      geom_point() +
      geom_smooth(method = "lm", se = FALSE)+facet_wrap(~state)
```

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



* The prices of the houses are lower in Pennysylvania, so the points are clustered at the left-bottom. And the variance of the vacancy rate in Pennysylvania is also lower. 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. * It aims to calculate the median number of total units in Alameda County, California. * Fisrtly, pick the row number of Alameda County, California and save it as `acca`. Then using `acca` as index, the information of total units in Alameda County, California is saved in `accamhv`. The method is loop trasversal. Lastly the median number can be calculated from `accamhv`.

b. Give a single line of R which gives the same final answer as the block of code. Note: there are at l

```
    ca_pa %>% filter(STATEFP == 6, COUNTYFP == 1) %>%
    select(Total_units) %>% unlist() %>% median()
```

```
## [1] 1606
```

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

```
  ca_house_2005 <- ca_pa %>%
  filter(
    (STATEFP == 6 & COUNTYFP %in% c(1, 85)) |
    (STATEFP == 42 & COUNTYFP == 3)
  ) %>%
  group_by(STATEFP, COUNTYFP) %>%
```

```
  summarize(
    total_built = sum(Built_2005_or_later, na.rm = TRUE),
    total_units = sum(Total_units, na.rm = TRUE),
    percent_built = (total_built / total_units) * 100,.groups = "drop") %>%
  mutate(county_name = case_when(
    STATEFP == 6 & COUNTYFP == 1 ~ "Alameda",
    STATEFP == 6 & COUNTYFP == 85 ~ "Santa Clara",
    STATEFP == 42 & COUNTYFP == 3 ~ "Allegheny"
  )) %>%
  select(county_name, percent_built)
  ca_house_2005
```

```
## # A tibble: 3 x 2
##   county_name percent_built
##   <chr>               <dbl>
## 1 Alameda             0.173
## 2 Santa Clara         0.188
## 3 Allegheny           0.0968
```

d. The ‘cor‘ function calculates the correlation coefficient between two variables. What is the correla

```
    mycor <- function(x){
      return(cor(x$Median_house_value, x$Built_2005_or_later))
    }
    mycor(ca_pa) #(i)
```

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

```
    ca_pa %>% filter(STATEFP == 6) %>% mycor()    #(ii)
```

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

```
    ca_pa %>% filter(STATEFP == 42) %>% mycor() #(iii)
```

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

```
    ca_pa %>% filter(STATEFP == 6, COUNTYFP == 1) %>% mycor() #(iv)
```

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

```
    ca_pa %>% filter(STATEFP == 6, COUNTYFP == 85) %>% mycor() #(v)
```

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

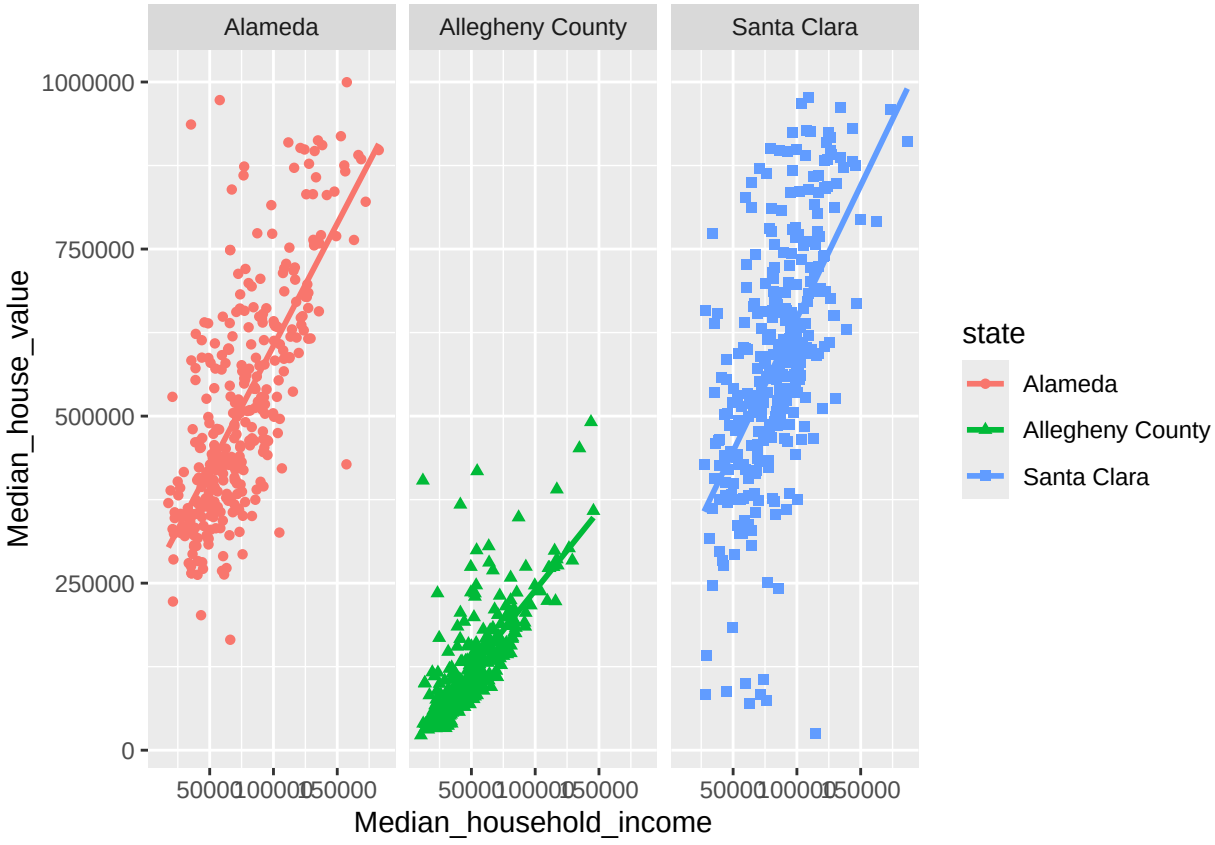
```
    ca_pa %>% filter(STATEFP == 42, COUNTYFP == 3) %>% mycor() #(vi)
```

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

e. Make three plots, showing median house values against median income, for Alameda, Santa Clara, and Al

```
ca_pa %>%
  filter(
    (STATEFP == "6" & COUNTYFP %in% c("1", "85")) |
    (STATEFP == "42" & COUNTYFP == "3")
  ) %>%
  mutate(state = case_when(
    STATEFP == "6" & COUNTYFP == "1" ~ "Alameda",
    STATEFP == "6" & COUNTYFP == "85" ~ "Santa Clara",
    STATEFP == "42" & COUNTYFP == "3" ~ "Allegheny County"
  )) %>%
  ggplot(aes(x = Median_household_income,
             y = Median_house_value,
             color = state,
             shape = state)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~ state)
```

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




```
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"))
# 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().

Table uses the cross-classifying factors to build a contingency table of the counts at each combination of factor levels.

- First line makes `gender` a factor with two levels `female`, `male`.
- Third line just changes the levels order.
- Fifth line takes the `levels` to be "Male" and "female", but there is no Male in gender.

- Eighth line outputs the count number of NA, which is the number of "female".

MB.Ch1.12. Write a function that calculates the proportion of values in a vector x that exceed some value cutoff.

```
cutoff_func <- function(x, cutoff) {mean(x > cutoff, na.rm = TRUE)}#x>cutoff will return a type of bool
```

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

```
test <- seq(1,100)
cutoff_func(test,25)
```

```
## [1] 0.75
```

(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")
```

```
## The following package(s) will be installed:
## - Devore7 [0.7.6]
## These packages will be installed into "E:/R-Learning/renv/library/windows/R-4.5/x86_64-w64-mingw32".
##
## # Installing packages --------------------------------------------------------
## - Installing Devore7 ...                        OK [copied from cache in 0.26s]
## Successfully installed 1 package in 0.29 seconds.
```

```
library(Devore7,warn.conflicts = FALSE)
```

```
## Loading required package: MASS
```

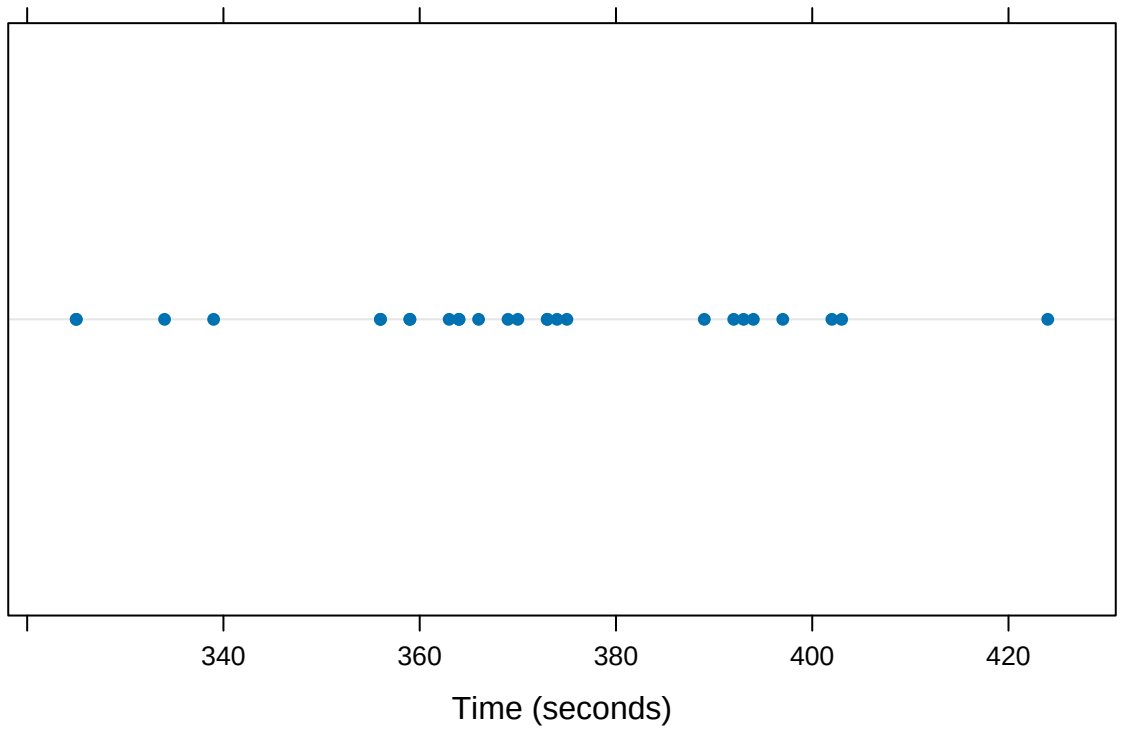
```
##
## Attaching package: 'MASS'
```

```
## 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)
library(lattice)
dotplot(~ex01.36, main = "Escape Times from Oil Platform", xlab = "Time (seconds)")
```




```
proportion <- mean(ex01.36 > 420) #1 minute = 60 seconds
proportion
```

```
## [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)
Dose <- unstack(Rabbit, Dose ~ Animal)[,1]
Treatment <- unstack(Rabbit, Treatment ~ Animal)[,1]
BPchange <- unstack(Rabbit, BPchange ~ Animal)
data.frame(Treatment, Dose, BPchange)
```

```
##    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
```