HW6_Salem_Mohamed

Mohamed Salem

October 6, 2019

```
# My countif function
countif.fn <- function(d, c = 1) {
    cnt = sum(d == c)
    return(cnt)
}
```

```
# seed setting and data creation
set.seed(12345)
P4b_data <- matrix(rbinom(10, 1, prob = (30:40)/100), nrow = 10,
    ncol = 10, byrow = FALSE)
print(P4b_data)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    1    1    1    1    1    1    1    1    1     1
##  [2,]    1    1    1    1    1    1    1    1    1     1
##  [3,]    1    1    1    1    1    1    1    1    1     1
##  [4,]    1    1    1    1    1    1    1    1    1     1
##  [5,]    0    0    0    0    0    0    0    0    0     0
##  [6,]    0    0    0    0    0    0    0    0    0     0
##  [7,]    0    0    0    0    0    0    0    0    0     0
##  [8,]    0    0    0    0    0    0    0    0    0     0
##  [9,]    1    1    1    1    1    1    1    1    1     1
## [10,]    1    1    1    1    1    1    1    1    1     1
```

```
# applying the counting function to get proportions
a <- P4b_data
cndtn_mat <- matrix(sapply(a, countif.fn), nrow = length(a[,
    1]), ncol = length(a[1, ]))
prptns_byrow <- rowSums(cndtn_mat)/length(a[, 1])
prptns_bycol <- colSums(cndtn_mat)/length(a[1, ])
print(prptns_bycol)
```

```
##  [1] 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6
```

```
print(prptns_byrow)
```

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

```
# function to produce matrix of bernoulli trials
binom.vec <- function(p) {
    P4b_data_adj <- matrix(rbernoulli(10, p), nrow = 10, ncol = 1)
    P4b_data_adj <- as.numeric(P4b_data_adj)
    return(P4b_data_adj)
}
```

```
# computing and printing prob.s for the correct matrix
prob_vec <- c(seq(0, 1, length.out = 10))
prob_mat <- matrix(sapply(prob_vec, binom.vec), ncol = length(a[1,
    ]))
print(prob_mat)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    0    0    1    1    1    1    1    1    1     1
##  [2,]    0    0    0    0    1    0    1    1    1     1
##  [3,]    0    1    0    1    0    1    1    0    0     1
##  [4,]    0    0    0    1    0    1    1    0    1     1
##  [5,]    0    0    0    0    1    1    0    0    1     1
##  [6,]    0    0    0    0    0    0    1    1    1     1
##  [7,]    0    0    1    0    1    1    1    1    1     1
##  [8,]    0    0    1    0    0    0    1    1    1     1
##  [9,]    0    0    0    0    0    1    1    1    0     1
## [10,]    0    0    0    0    0    1    0    0    1     1
```

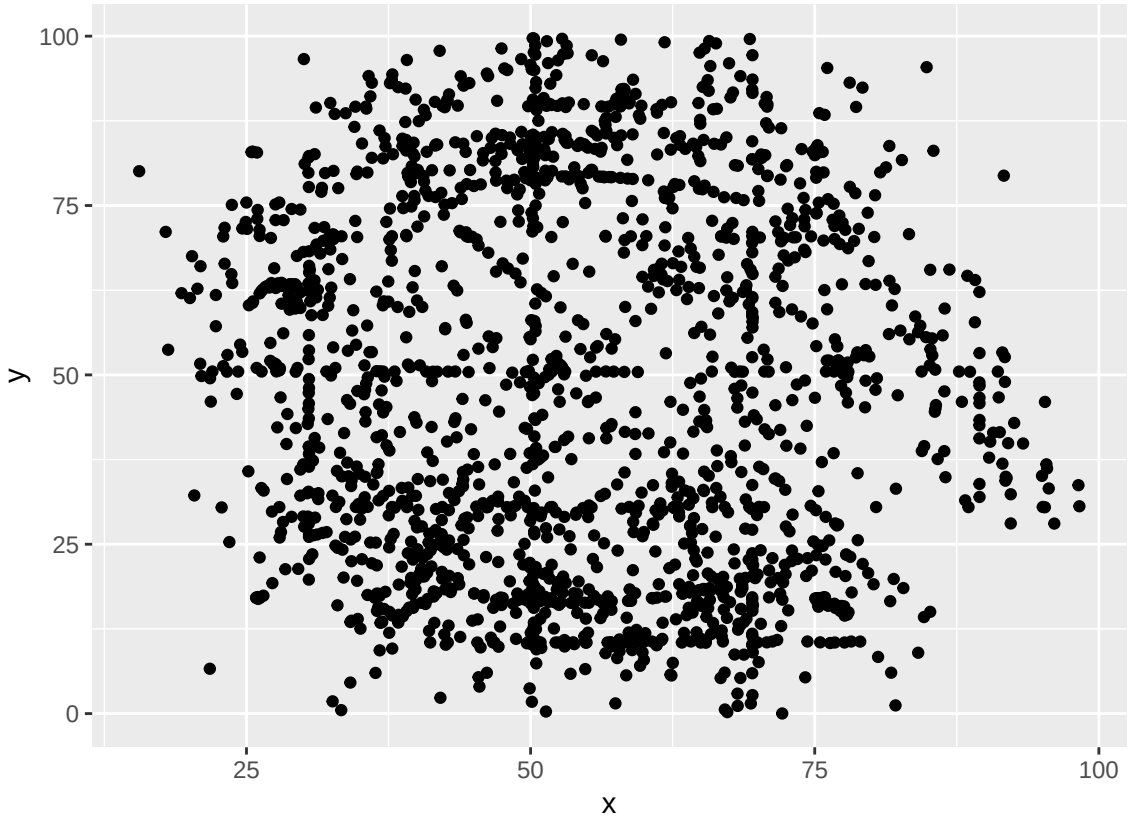
```
prptns_byrow <- rowSums(prob_mat)/length(prob_mat[, 1])
prptns_bycol <- colSums(prob_mat)/length(prob_mat[1, ])
print(prptns_bycol)
```

```
##  [1] 0.0 0.1 0.3 0.3 0.4 0.7 0.8 0.6 0.8 1.0
```

```
print(prptns_byrow)
```

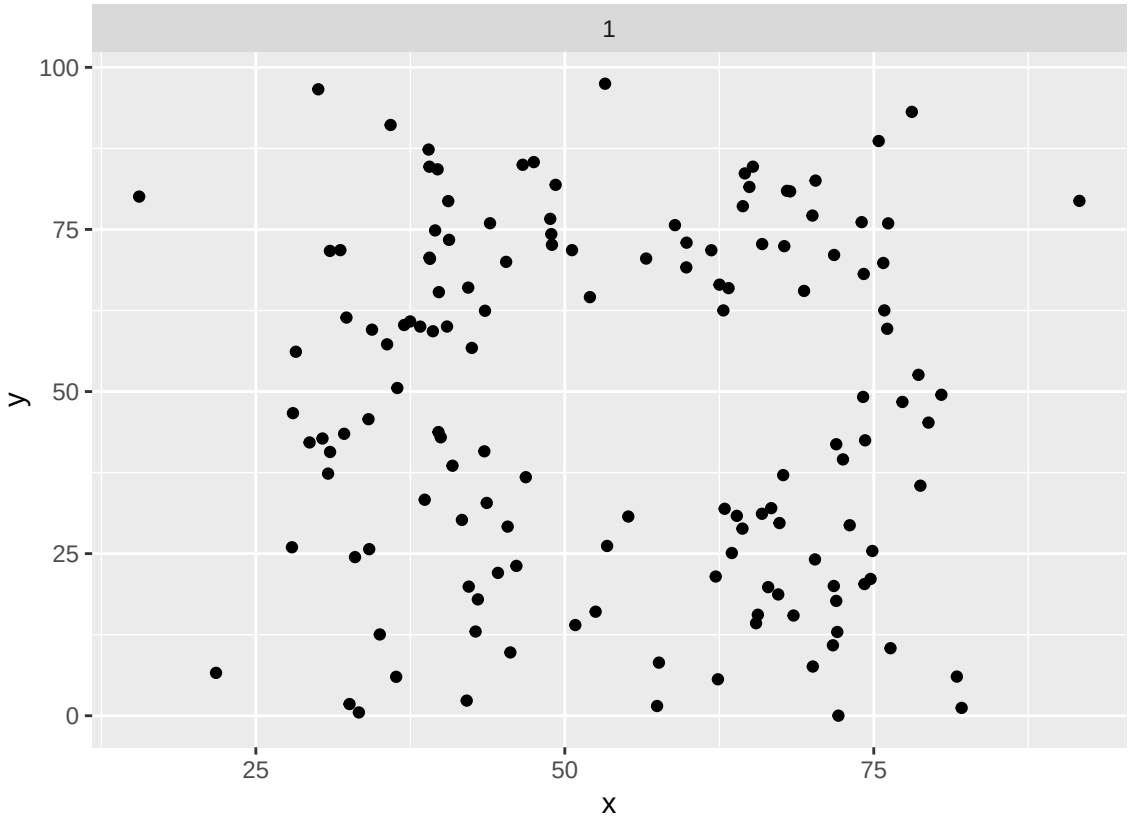
```
##  [1] 0.8 0.5 0.5 0.5 0.4 0.4 0.7 0.5 0.4 0.3
```

```
# Read and import the object
df <- readRDS(file = "D:/Vtech/Statistical Programming/HW4_data.rds")
names(df) <- c("Observer", "x", "y")

# My plotting function
scatterdat <- function(i = 0) {
    if (i %in% c(0)) {
        ggplot(df, aes(x, y)) + geom_point()
    } else {
        ggplot(df[df$Observer == i, ], aes(x, y)) + facet_wrap(~Observer) +
            geom_point()
    }
}

# Plotting all data
scatterdat()
```

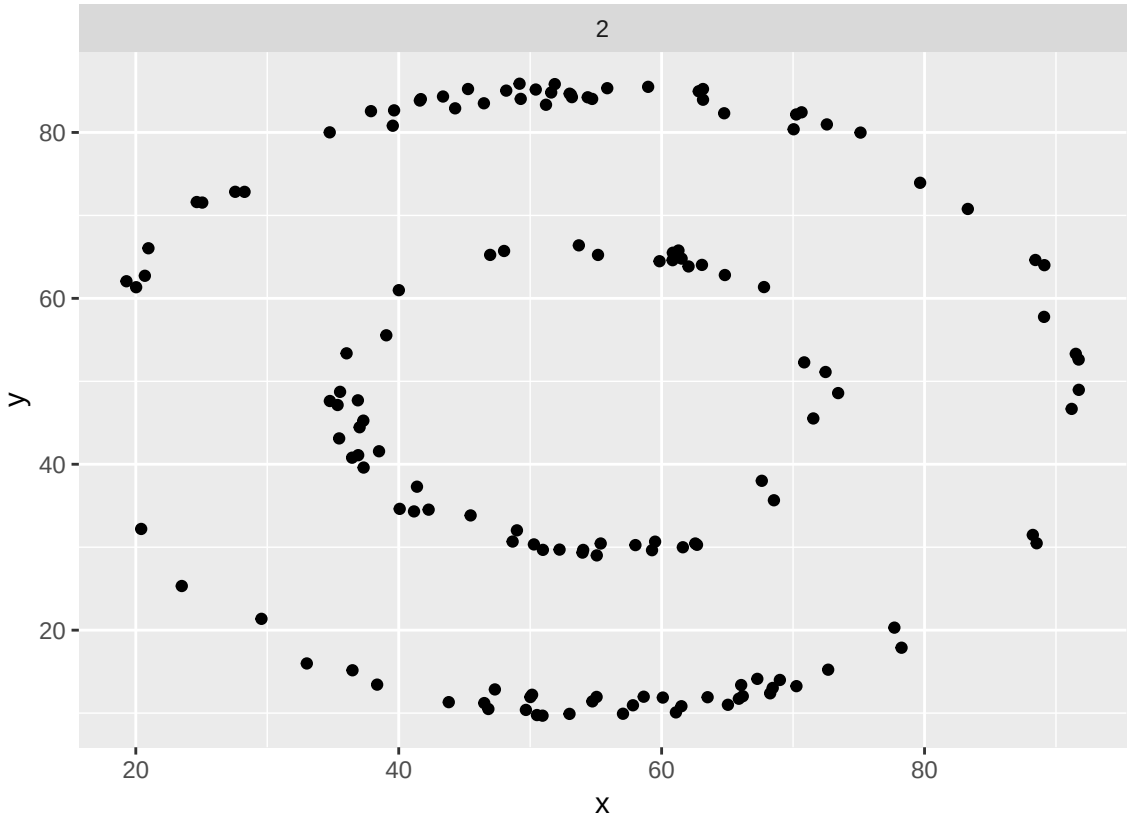



```
# plotting data by observer (P.S: The T-Rex is just AWESOME!)
sapply(sort(unique(df$Observer)), scatterdat, simplify = FALSE)
```

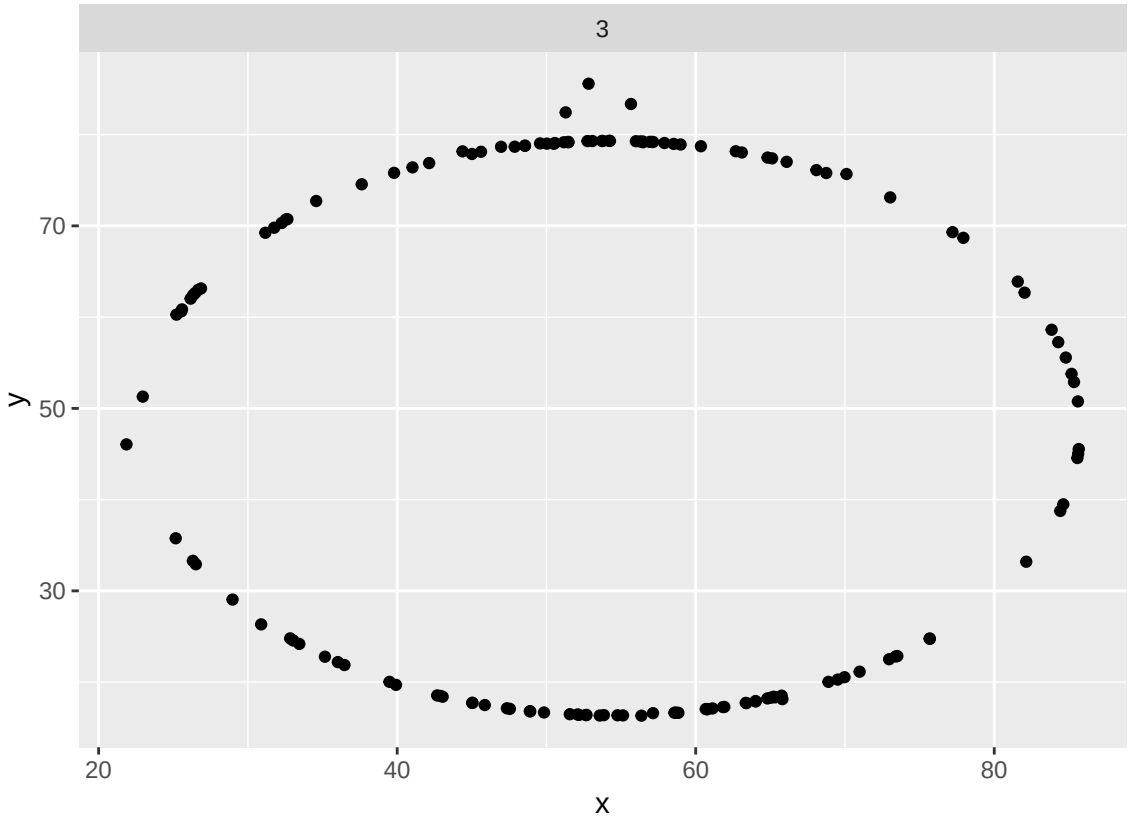
```
## [[1]]
```



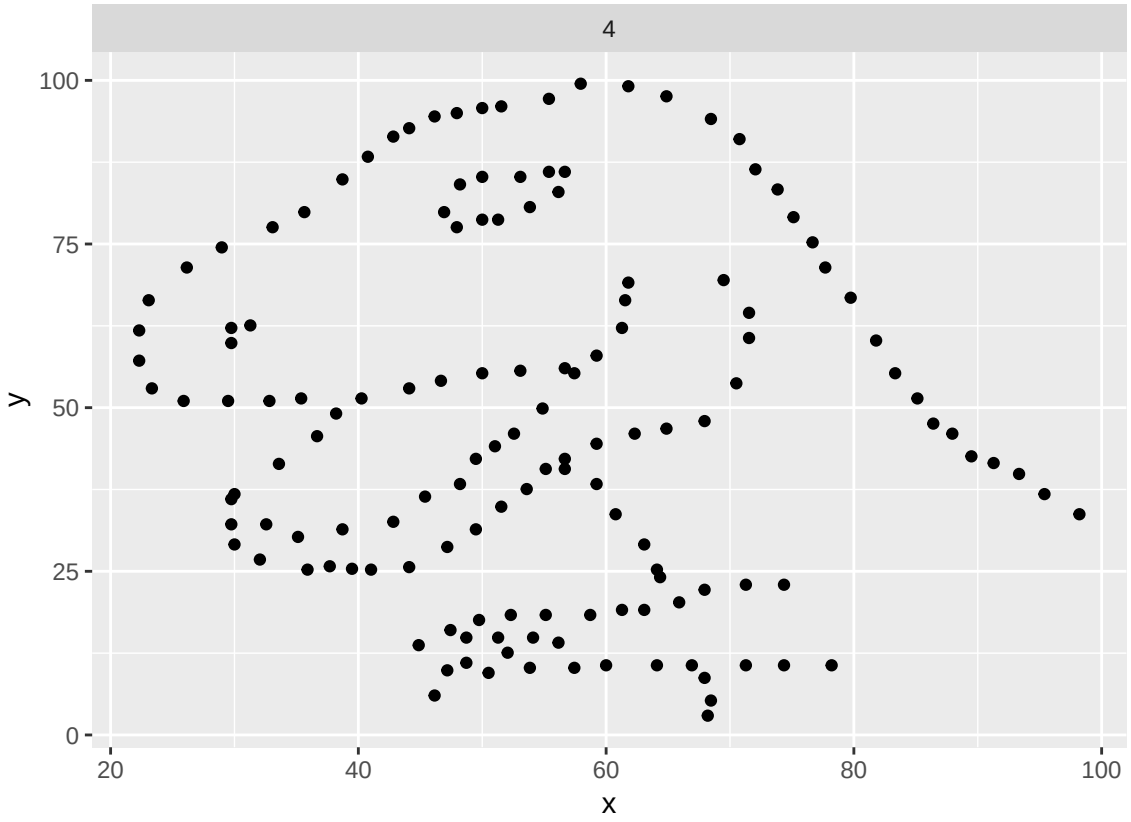

```
##
## [[2]]
```



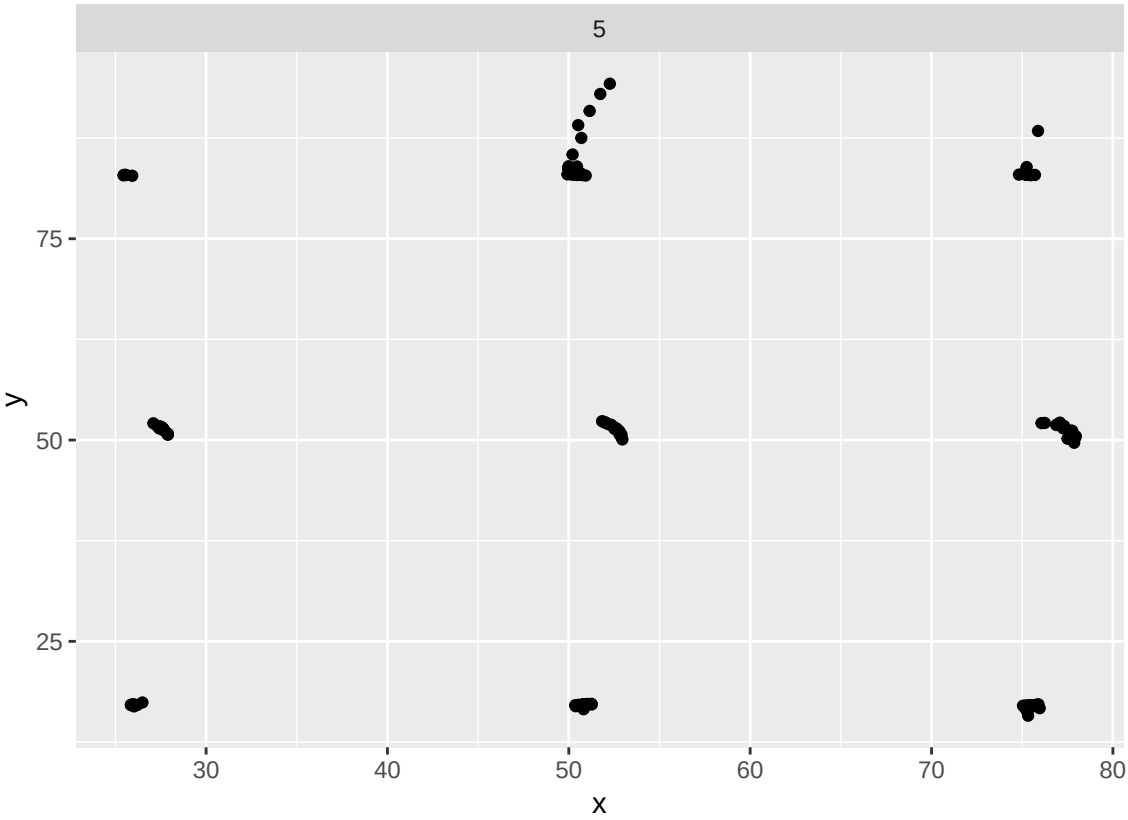

```
##
## [[3]]
```



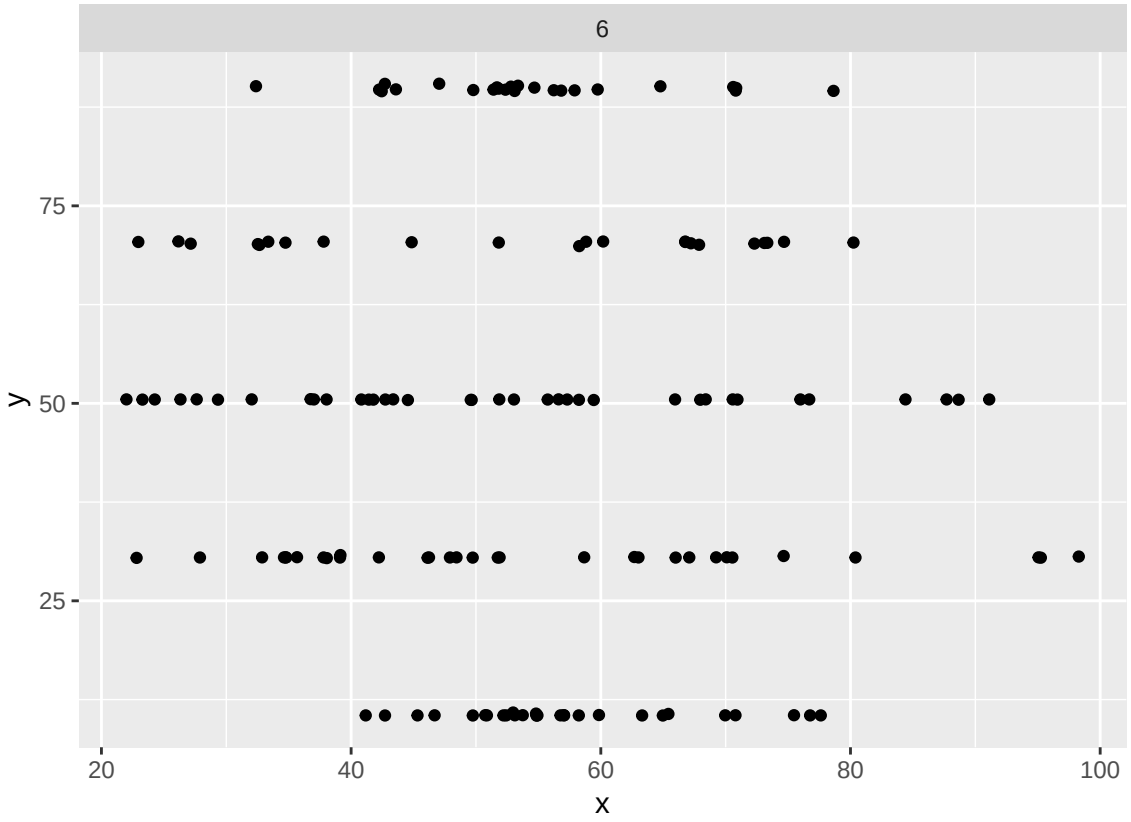

```
##
## [[4]]
```



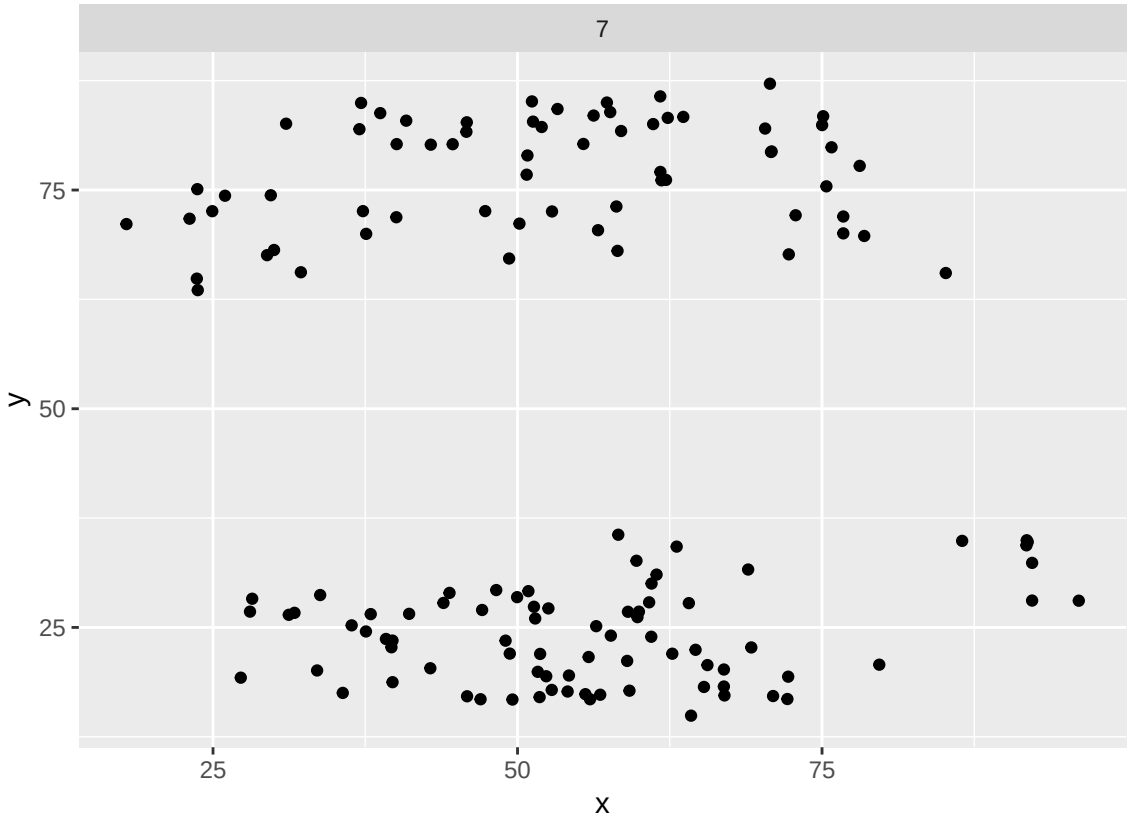

```
##
## [[5]]
```



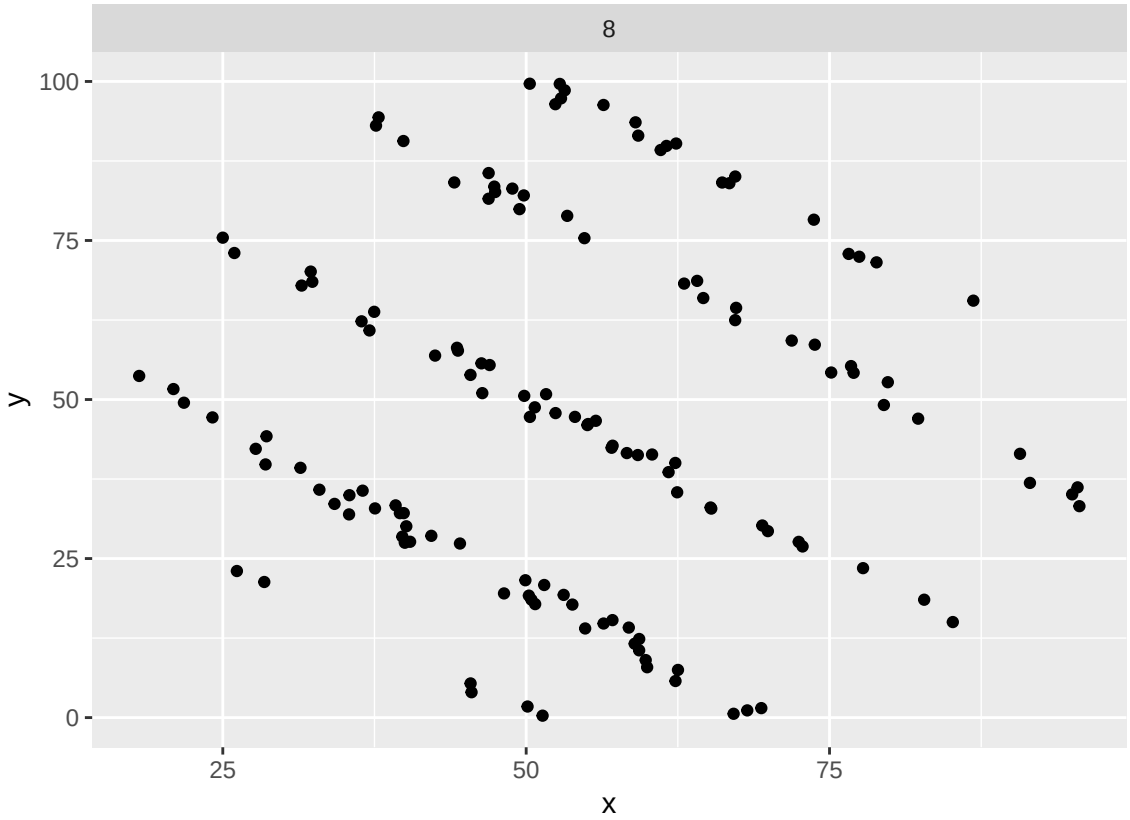

```
##
## [[6]]
```



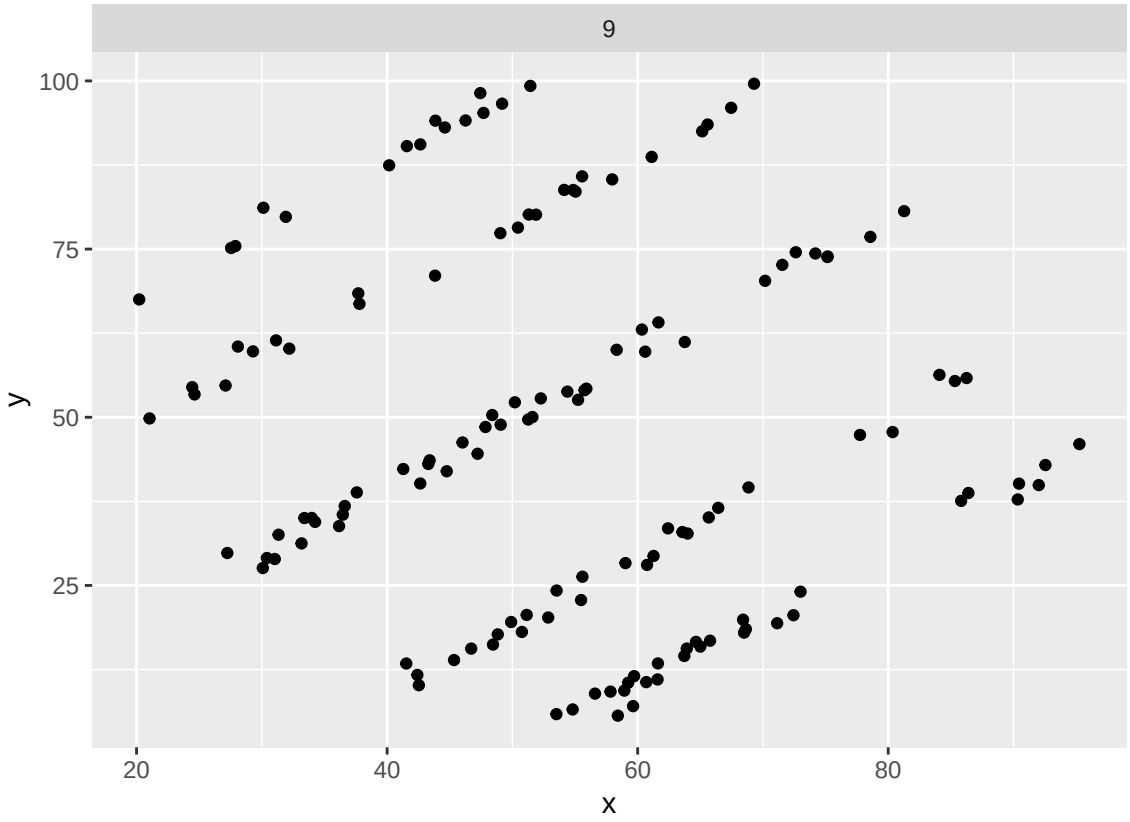

```
##
## [[7]]
```



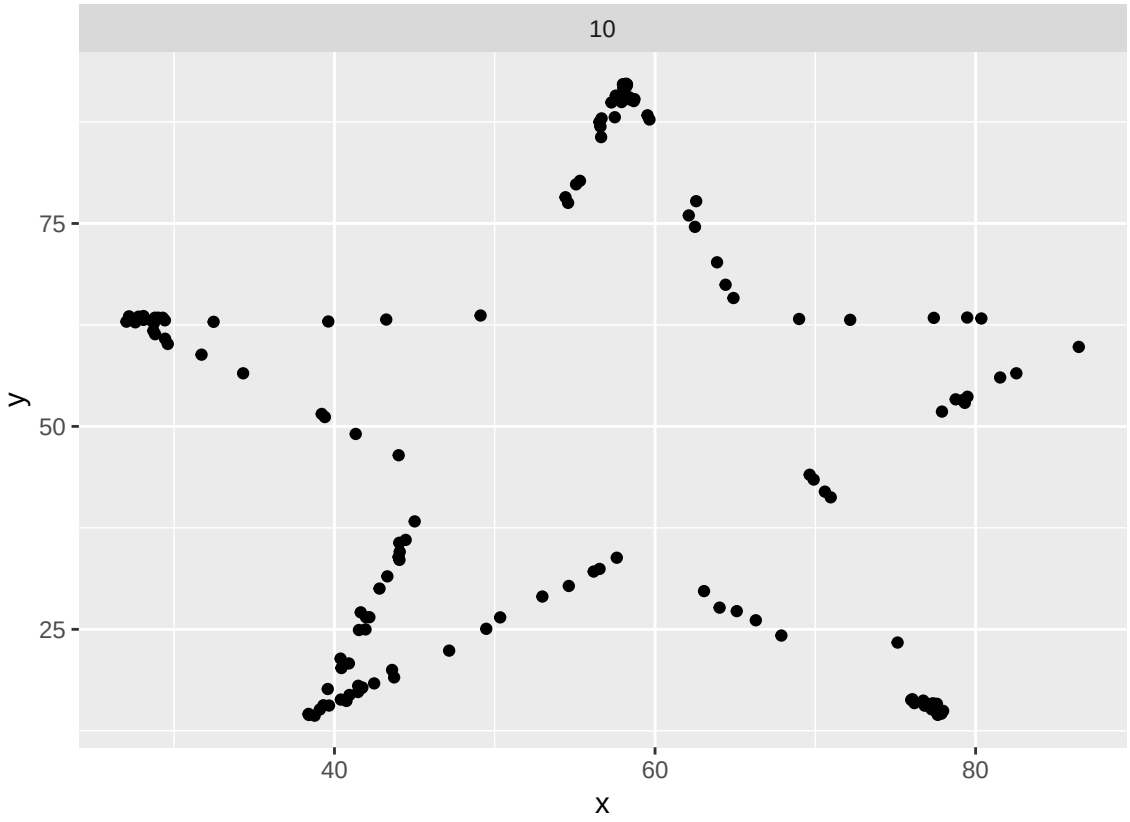

```
##
## [[8]]
```



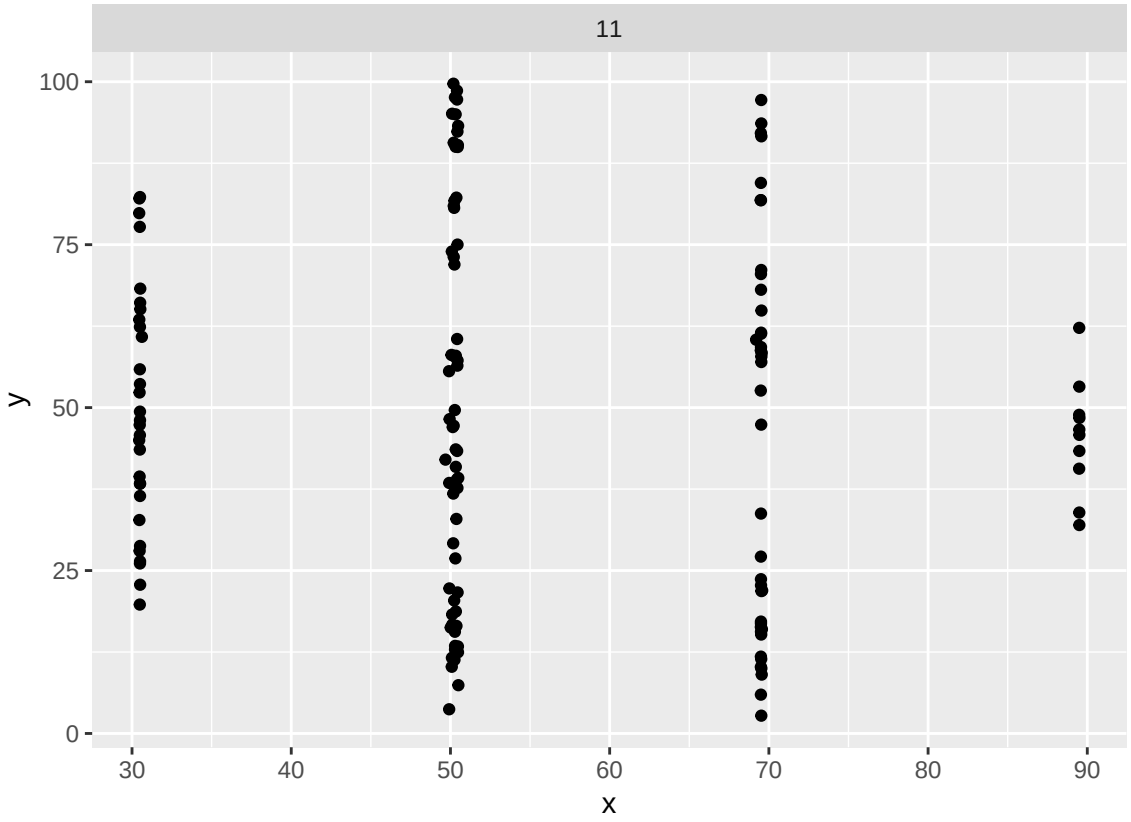

```
##
## [[9]]
```



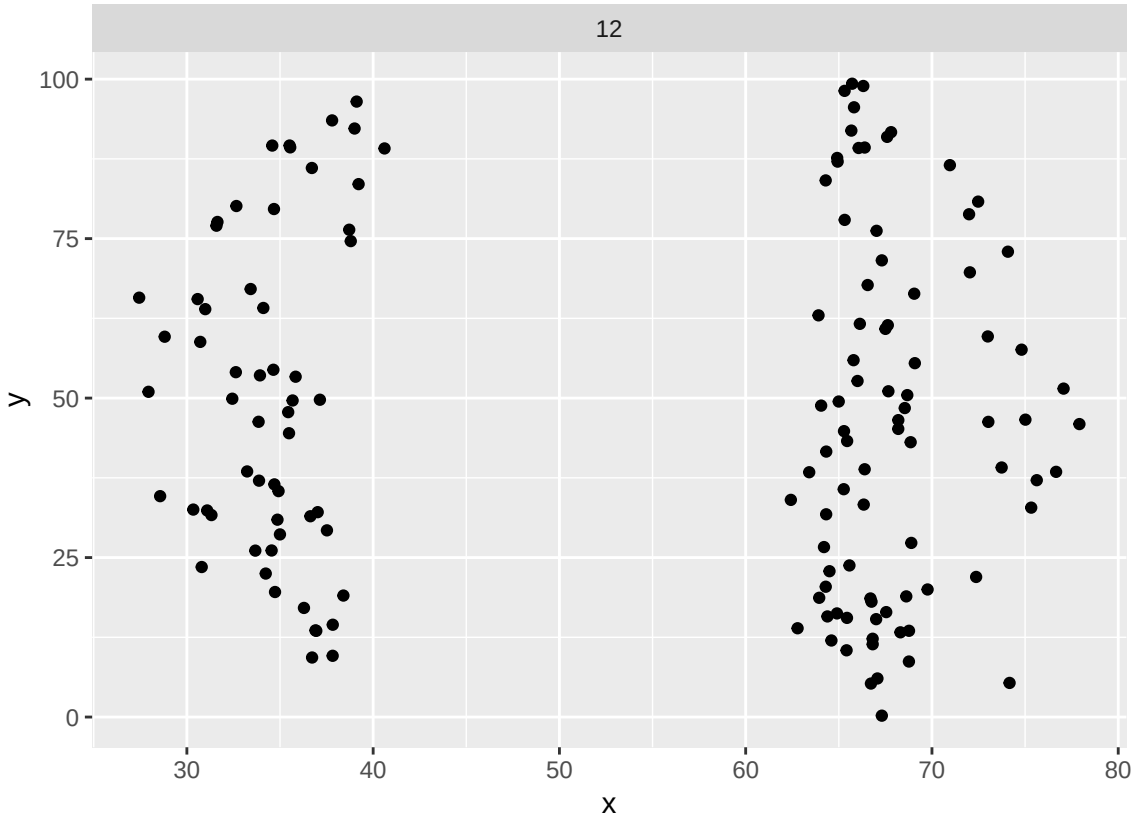

```
##
## [[10]]
```



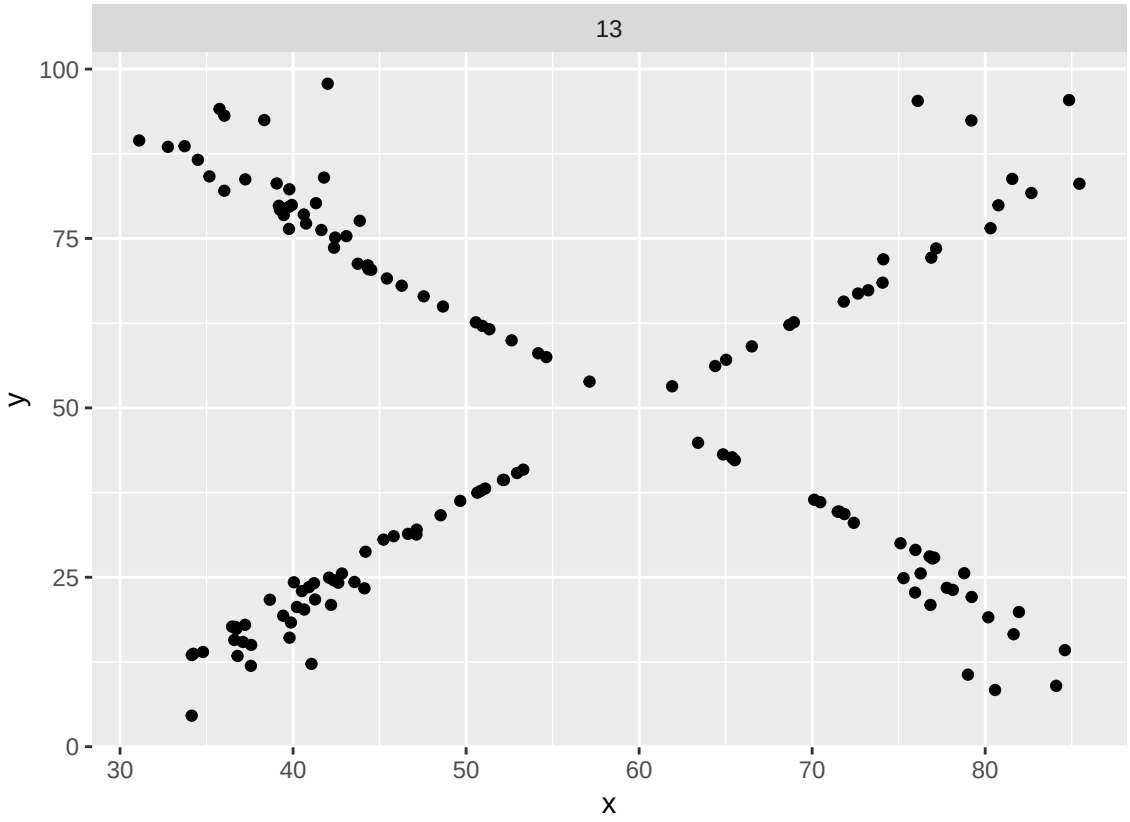

```
##
## [[11]]
```




```
##
## [[12]]
```




```
##
## [[13]]
```




```
# download the files
library(downloader)
download("http://www.farinspace.com/wp-content/uploads/us_cities_and_states.zip",
    dest = "us_cities_states.zip")
unzip("us_cities_states.zip", exdir = ".")

# read in data
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
states <- fread(input = "./us_cities_and_states/states.sql",
    skip = 23, sep = "'", sep2 = ",", header = FALSE, select = c(2,
        4))
cities <- fread(input = "./us_cities_and_states/cities.sql",
    skip = 0, sep = "'", sep2 = ",", header = FALSE, select = c(2,
        4))
cities_ext <- fread(input = "./us_cities_and_states/cities_extended.sql",
    skip = 0, sep = "'", sep2 = ",", header = FALSE, select = c(2,
        4, 12))
```

```
cities_ext <- filter(cities_ext, (V4 != "PR" & V4 != "DC"))

# Some diagnostics to check if everything's fine
sort(unique(cities_ext$V4))
```

```
##  [1] "AK" "AL" "AR" "AZ" "CA" "CO" "CT" "DE" "FL" "GA" "HI" "IA" "ID" "IL"
## [15] "IN" "KS" "KY" "LA" "MA" "MD" "ME" "MI" "MN" "MO" "MS" "MT" "NC" "ND"
## [29] "NE" "NH" "NJ" "NM" "NV" "NY" "OH" "OK" "OR" "PA" "RI" "SC" "SD" "TN"
## [43] "TX" "UT" "VA" "VT" "WA" "WI" "WV" "WY"
```

```
length(unique(cities_ext$V4))
```

```
## [1] 50
```

```
city_cnt <- data.frame(table(cities_ext$V4))
print(table(cities_ext$V4))
```

```
##
##   AK   AL   AR   AZ   CA   CO   CT   DE   FL   GA   HI   IA   ID   IL   IN
##  273  838  709  532 2651  659  438   98 1487  972  139 1060  325 1587  989
##   KS   KY   LA   MA   MD   ME   MI   MN   MO   MS   MT   NC   ND   NE   NH
##  756  961  725  703  619  489 1170 1031 1170  533  405 1090  407  620  284
##   NJ   NM   NV   NY   OH   OK   OR   PA   RI   SC   SD   TN   TX   UT   VA
##  733  426  253 2207 1446  774  484 2208   91  539  394  795 2650  344 1238
##   VT   WA   WI   WV   WY
##  309  732  898  859  195
```

```
# letter counting function
ltr.cntr <- function(a, b) {
    a <- as.character(a)
    b <- as.character(b)
    r <- which(strsplit(a, "")[[1]] == b)
    r <- sum(r > 0)
    return(r)
}
```

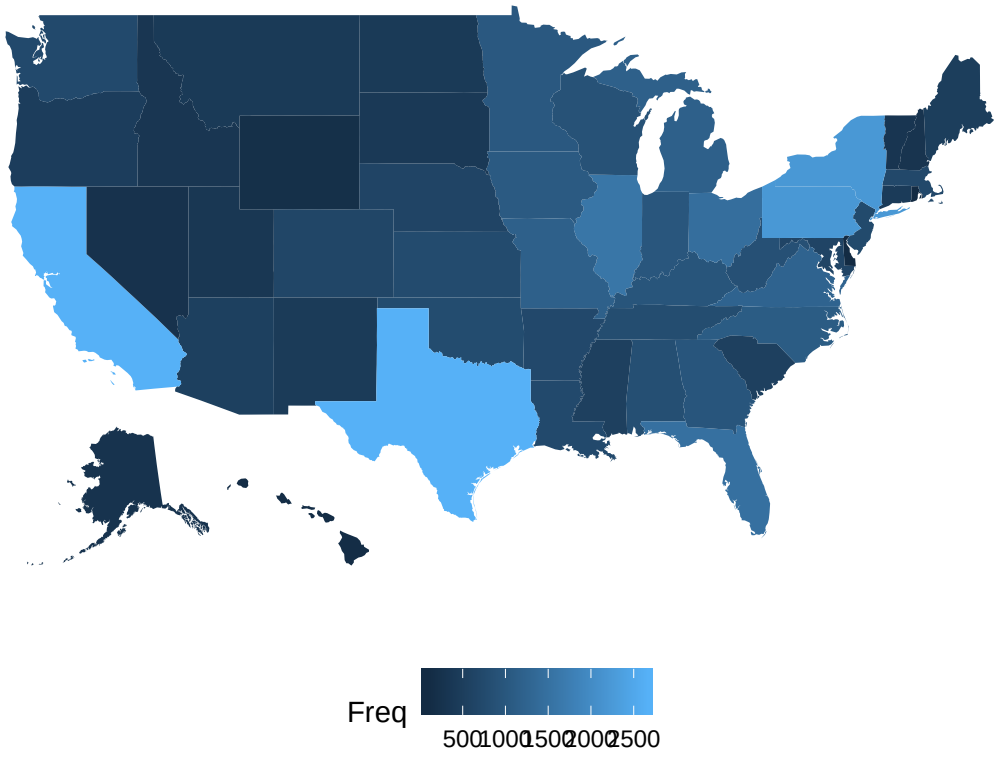
```
# appllying the letter counting function
letter_count <- data.frame(matrix(NA, nrow = 50, ncol = 27))
states_nodc <- filter(states, V4 != "DC")
letter_count[, 1] <- tolower(states_nodc[, 1])
names(letter_count) <- c("State", letters)
vec_feed <- as.character(letters)
state_vec <- tolower(states_nodc[, 1])

for (j in 2:27) {
    letter_count[, j] <- apply(letter_count, 1, ltr.cntr, vec_feed[j -
        1])
}

print(letter_count)
```

```
##             State a b c d e f g h i j k l m n o p q r s t u v w x y z
## 1          alaska 3 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
## 2         alabama 4 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## 3        arkansas 3 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 2 0 0 0 0 0 0 0
## 4         arizona 2 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1
```

```
## 5      california 2 0 1 0 0 1 0 0 2 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 0
## 6        colorado 1 0 1 1 0 0 0 0 0 0 0 1 0 0 3 0 0 1 0 0 0 0 0 0 0 0
## 7     connecticut 0 0 3 0 1 0 0 0 1 0 0 0 0 2 1 0 0 0 0 2 1 0 0 0 0 0
## 8        delaware 2 0 0 1 2 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0
## 9         florida 1 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0
## 10        georgia 1 0 0 0 1 0 2 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0
## 11         hawaii 2 0 0 0 0 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
## 12           iowa 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0
## 13          idaho 1 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
## 14       illinois 0 0 0 0 0 0 0 0 3 0 0 2 0 1 1 0 0 0 1 0 0 0 0 0 0 0
## 15        indiana 2 0 0 1 0 0 0 0 2 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0
## 16         kansas 2 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 2 0 0 0 0 0 0
## 17       kentucky 0 0 1 0 1 0 0 0 0 0 2 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0
## 18      louisiana 2 0 0 0 0 0 0 0 0 2 0 0 1 0 1 1 0 0 0 1 0 1 0 0 0 0 0
## 19  massachusetts 2 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 4 2 1 0 0 0 0 0
## 20       maryland 2 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 1 0
## 21          maine 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
## 22       michigan 1 0 1 0 0 0 1 1 2 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
## 23      minnesota 1 0 0 0 1 0 0 0 1 0 0 0 0 1 2 1 0 0 0 1 1 0 0 0 0 0 0
## 24       missouri 0 0 0 0 0 0 0 0 0 2 0 0 0 1 0 1 0 0 0 1 2 0 1 0 0 0 0
## 25    mississippi 0 0 0 0 0 0 0 0 0 4 0 0 0 0 1 0 0 2 0 0 4 0 0 0 0 0 0
## 26        montana 2 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 0 0 0 0 0 1 0 0 0 0
## 27 north carolina 2 0 1 0 0 0 0 0 1 1 0 0 1 0 2 2 0 0 2 0 1 0 0 0 0 0 0
## 28   north dakota 2 0 0 1 0 0 0 0 1 0 0 1 0 0 1 2 0 0 1 0 2 0 0 0 0 0 0
## 29       nebraska 2 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0
## 30  new hampshire 1 0 0 0 2 0 0 2 1 0 0 0 0 1 1 0 1 0 1 1 0 0 0 1 0 0 0
## 31     new jersey 0 0 0 0 3 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 1 0 1 0
## 32     new mexico 0 0 1 0 2 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 0 0
## 33         nevada 2 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0
## 34       new york 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0
## 35           ohio 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0
## 36       oklahoma 2 0 0 0 0 0 0 0 1 0 0 1 1 1 0 2 0 0 0 0 0 0 0 0 0 0 0
## 37         oregon 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 2 0 0 1 0 0 0 0 0 0 0 0
## 38   pennsylvania 2 0 0 0 1 0 0 0 1 0 0 1 0 3 0 1 0 0 1 0 0 1 0 0 1 0
## 39   rhode island 1 0 0 2 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0 0 0 0 0 0 0
## 40 south carolina 2 0 1 0 0 0 0 0 1 1 0 0 1 0 1 2 0 0 1 1 1 1 0 0 0 0 0
## 41   south dakota 2 0 0 1 0 0 0 0 1 0 0 1 0 0 0 2 0 0 0 1 2 1 0 0 0 0 0
## 42      tennessee 0 0 0 0 4 0 0 0 0 0 0 0 0 2 0 0 0 0 2 1 0 0 0 0 0 0
## 43          texas 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0
## 44           utah 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
## 45       virginia 1 0 0 0 0 0 0 1 0 3 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0
## 46        vermont 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 1 0 0 0 0
## 47     washington 1 0 0 0 0 0 0 1 1 1 0 0 0 0 2 1 0 0 0 1 1 0 0 1 0 0 0
## 48      wisconsin 0 0 1 0 0 0 0 0 0 2 0 0 0 0 2 1 0 0 0 2 0 0 0 1 0 0 0
## 49  west virginia 1 0 0 0 1 0 1 0 3 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 0 0
## 50        wyoming 0 0 0 0 0 0 0 1 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0
```

```
library(fiftystater)
data("fifty_states")

crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)
# map_id creates the aesthetic mapping to the state name column in
# your data
```

```
# A few matching commands to associate city counts with states
df <- merge(states_nodc[, c("V4", "V2")], city_cnt[, c("Var1", "Freq")],
    by.x = "V4", by.y = "Var1")
df[, 2] <- tolower(df$V2)
df <- merge(crimes[, c("state", "Murder", "Assault", "UrbanPop", "Rape")],
    df[, c("V2", "Freq")], by.x = "state", by.y = "V2")

p <- ggplot(df, aes(map_id = state)) + # map points to the fifty_states shape data
geom_map(aes(fill = Freq), map = fifty_states) + expand_limits(x = fifty_states$long,
    y = fifty_states$lat) + coord_map() + scale_x_continuous(breaks = NULL) +
    scale_y_continuous(breaks = NULL) + labs(x = "", y = "") + theme(legend.position = "bottom",
    panel.background = element_blank())

p
```




```
# finding a specified number of letter repeats
letter_count_gt3 <- letter_count
letter_count_gt3 <- data.frame(letter_count_gt3 >= 3)
letter_count_gt3$State <- letter_count$State
letter_count_gt3$tot <- rowSums(letter_count_gt3[, 2:27])
letter_count_gt3$tot[letter_count_gt3$tot > 0] <- 1
letter_count_gt3 <- data.frame(cbind(letter_count_gt3$State, letter_count_gt3$tot))

# matching command to link states on the map to number of letter
# repeats
df <- merge(crimes[, c("state", "Murder", "Assault", "UrbanPop", "Rape")],
    letter_count_gt3[, c("X1", "X2")], by.x = "state", by.y = "X1")
```

```
p <- ggplot(df, aes(map_id = state)) + # map points to the fifty_states shape data
geom_map(aes(fill = X2), color = "black", map = fifty_states) + expand_limits(x = fifty_states$long,
    y = fifty_states$lat) + coord_map() + scale_x_continuous(breaks = NULL) +
    scale_y_continuous(breaks = NULL) + labs(x = "", y = "") + scale_fill_grey() +
    theme(legend.position = "bottom", panel.background = element_blank())

p
```