STA 141A Homework 1

Li Yuan

Contents

Problem 1 Vectors and simulation

Suppose that we have:

-four types of animals: `cat`, `dog`, `cow`, `squirrel`;

-four possible colors: `white`, `black`, `brown`, `red`;

-five possible attributes: `big`, `small`, `angry`, `cute`, `finicky`.

Question (a)

Generate three random samples of size 100 from each of the three groups, so that you have a vector containing 100 animals, a vector containing 100 colors and a vactor containing 100 attributes. Call the resulting vectors of character strings as: Animal, Color, Attribute (1 point).

```
Animal = sample(c('cat', 'dog', 'cow', 'squirrel'), 100, replace = TRUE)
Color = sample(c('white', 'black', 'brown', 'red'), 100, replace = TRUE)
Attribute = sample(c('big', 'small', 'angry', 'cute', 'finicky'), 100, replace = TRUE)
```

Question (b)

Using the `sum()` function and a logical vector, compute the number of animals that are cats or dogs (1 point).

```
sum(Animal == 'cat' | Animal == 'dog')
```

```
## [1] 51
```

Question (c)

Compute the relative frequency of cats, dogs, cows and squirrels in the sample (1 point).

```
mean(Animal == 'cat')
```

```
## [1] 0.29
```

```
mean(Animal == 'dog')
```

```
## [1] 0.22
```

```
mean(Animal == 'cow')
```

```
## [1] 0.25
```

```
mean(Animal == 'squirrel')
```

```
## [1] 0.24
```

Question (d)

Create a contingency table between `Animal` and `Attribute` (1 point).

```
table(Animal, Attribute)
```

```
##           Attribute
## Animal     angry big cute finicky small
##   cat          3  10    3       8     5
##   cow          4   5    4       9     3
##   dog          3   5    5       3     6
##   squirrel     4   7    5       5     3
```

Question (e)

Put the three vectors together in a list of three elements called `mylist`, so that each vector is an element of the list. Use the command `length(mylist[1])` to print the length of the first vector. Is this code actually printing the length of the vector? Explain and write the correct code to print the length of the first vector of the list. (1 point)

```
mylist = list(Animal, Color, Attribute)
length(mylist[1])
```

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

The command `length(mylist[1])` does not print the true length of the first vector `Animal`. It prints the length of the vector as a whole instead of the elements inside the vector. The correct code is shown as follow.

```
length(mylist[[1]])
```

```
## [1] 100
```

Problem 2 Matrices

Consider the following system of linear equations

$$x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 = 7$$

$$2x_1 + x_2 + 2x_3 + 3x_4 + 4x_5 = -1$$

$$3x_1 + 2x_2 + x_3 + 2x_4 + 3x_5 = -3$$

$$4x_1 + 3x_2 + 2x_3 + x_4 + 2x_5 = 5$$

$$5x_1 + 4x_2 + 3x_3 + 2x_4 + x_5 = 17$$

Question (a)

Create the matrix `A` and the vector `y` corresponding to the matrix equation $Ax = y$, where $A \in R^{5\times5}$ and $x, y \in R^5$ (1 point).

```
A = matrix(c(1, 2, 3, 4, 5,
             2, 1, 2, 3, 4,
             3, 2, 1, 2, 3,
             4, 3, 2, 1, 2,
             5, 4, 3, 2, 1), byrow = TRUE, nrow = 5)
y = c(7, -1, -3, 5, 17)
```

Question (b)

Determine if the matrix A is invertible using the `det()` function (1 point).

```
det(A)
```

```
## [1] 48
```

We can see our determinant of the square matrix A is 48, which is not zero. As a result, matrix A is invertible.

Question (c)

Find the solution of the system of linear equations using the `solve()` function (1 point).

```
solve(A, y)
```

```
## [1] -2  3  5  2 -4
```

As we can see from above, the solution of the system of linear equation can be $x_1 = -2$, $x_2 = 3$, $x_3 = 5$, $x_4 = 2$, and $x_5 = -4$.

Problem 3 Data objects

The `dist()` function is an R built-in function that takes a matrix (or a data frame) and returns the distance between the individuals (rows) of the matrix.

Question (a)

Without using for loops or `apply` functions, create a matrix containing 10 rows and 5 columns. The elements of the matrix are integer numbers randomly selected from 1 to 10 (1 point).

```
p3matrix = matrix(sample(1:10, size=50, replace=T), nrow = 10, ncol = 5)
p3matrix
```

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

Question (b)

Apply the `dist()` function to the above matrix and assign the output to an object called `m_dist`. Assume that you want to print the first 6 rows of `m_dist`, and since we learned that we can use the `head()` function for printing the first few rows of a matrix or of a dataframe, try using it on `m_dist`. What does `head(m_dist)` return? What is the class of `m_dist`? Explain in words in addition to writing the code. (2 point)

```
m_dist = dist(p3matrix)
head(m_dist)
```

```
## [1] 10.344080  5.099020  3.464102  9.055385  7.416198  7.745967
```

```
class(m_dist)
```

```
## [1] "dist"
```

As we can see above, the `head(m_dist)` return the first six numbers of the distance matrix of `p3matrix` we created in Question (a).

The class of `m_dist` is dist, which means it is a distance matrix.

Question (c)

Write a code to print the first 6 rows of `m_dist` (2 point). Hint: transform `m_dist` into an object where you can use the `head()` function.

```
head(as.matrix(m_dist))
```

```
##           1         2         3         4         5         6        7
## 1  0.000000 10.344080  5.099020  3.464102  9.055385  7.416198 7.745967
## 2 10.344080  0.000000 11.532563 10.908712  6.403124  9.695360 3.605551
## 3  5.099020 11.532563  0.000000  6.480741 12.083046  7.937254 8.602325
## 4  3.464102 10.908712  6.480741  0.000000  9.055385  5.916080 8.000000
## 5  9.055385  6.403124 12.083046  9.055385  0.000000 11.000000 7.071068
## 6  7.416198  9.695360  7.937254  5.916080 11.000000  0.000000 6.244998
##           8         9        10
## 1  8.426150  6.928203  8.426150
## 2  5.656854  8.660254  9.695360
## 3  8.660254 10.862780  7.416198
## 4  9.539392  6.000000 10.535654
## 5  7.280110  6.782330 12.124356
## 6 10.392305  6.855655  9.695360
```

Problem 4 Data exploration and manipulation

The task is to explore the US census population estimates by county for 2015 from the package `usmap` (load the data frame from `countypop.RData`). The data frame has `3142` rows and `4` variables: `fips` is the 5-digit FIPS code corresponding to the county; `abbr` is the 2-letter state abbreviation; `county` is the full county name; `pop_2015` is the 2015 population estimate (in number of people) for the corresponding county. Each row of the data frame represents a different county or a county equivalent. For the sake of simplicity, when we say a county, that also includes a county equivalent and when we say a state, that also includes the District of Columbia.

Without using extra libraries, without creating new functions, and without using for loops, answer the following questions.

Question (a)

Remove all the rows that contain at least one NA (1 point).

```
countypop = na.omit(countypop)
head(countypop)
```

```
## # A tibble: 6 x 4
##   fips  abbr  county         pop_2015
##   <chr> <chr> <chr>             <dbl>
## 1 01001 AL    Autauga County    55347
## 2 01003 AL    Baldwin County   203709
## 3 01007 AL    Bibb County       22583
## 4 01009 AL    Blount County     57673
## 5 01011 AL    Bullock County    10696
## 6 01013 AL    Butler County     20154
```

Question (b)

What is the total number of counties in the US (1 point)?

```
length(countypop$county)
```

```
## [1] 3139
```

Question (c)

How many unique county names are there (1 point)?

```
length(unique(countypop$county))
```

```
## [1] 1876
```

Question (d)

What are the top 10 most common county names (1 point)?

```
sort(table(countypop$county), decreasing = TRUE)[1:10]
```

```
##
## Washington County  Jefferson County   Franklin County    Jackson County
##                30                25                24                23
##    Lincoln County    Madison County       Clay County Montgomery County
##                23                19                18                18
##     Marion County     Monroe County
##                17                17
```

```
# Extra credit - using dplyr library
head(count(countypop, county, sort = TRUE), 10)
```

```
## # A tibble: 10 x 2
##    county                n
##    <chr>             <int>
##  1 Washington County    30
##  2 Jefferson County     25
##  3 Franklin County      24
##  4 Jackson County       23
##  5 Lincoln County       23
##  6 Madison County       19
##  7 Clay County          18
##  8 Montgomery County    18
##  9 Marion County        17
## 10 Monroe County        17
```

The top 10 most common county names are shown above, which are Washington County, Jefferson County, Franklin County, Jackson County, Lincoln County, Madison County, Clay County, Montgomery County, Marion County, and Monroe County.

Question (e)

Which state has the largest number of counties? Which state has the smallest number of counties (1 point)?

```
# State that had largest number of counties
sort(table(countypop$abbr), decreasing = TRUE)[1]
```

```
##  TX
## 254
```

```
# Extra credit - using dplyr library
head(count(countypop, abbr, sort = TRUE), 1)
```

```
## # A tibble: 1 x 2
##   abbr      n
##   <chr> <int>
## 1 TX      254
```

```r
# State that had smallest number of counties
sort(table(countypop$abbr), decreasing = FALSE)[1]
```

```
## DC
##  1
```

```r
# Extra credit - using dplyr library
tail(count(countypop, abbr, sort = TRUE), 1)
```

```
## # A tibble: 1 x 2
##   abbr      n
##   <chr> <int>
## 1 DC        1
```

As we can see above, Texas (TX) has the most counties, and District of Columbia (DC) has the least counties.

Question (f)

What is the average population of a county in the US (1 point)?

```r
mean(countypop$pop_2015)
```

```
## [1] 102329.2
```

```r
ceiling(mean(countypop$pop_2015))
```

```
## [1] 102330
```

```r
# Extra credit - using dplyr library
summarise(countypop, avg=mean(pop_2015))
```

```
## # A tibble: 1 x 1
##       avg
##     <dbl>
## 1 102329.
```

```r
ceiling(summarise(countypop, avg=mean(pop_2015)))
```

```
## # A tibble: 1 x 1
##      avg
##    <dbl>
## 1 102330
```

The average population of a county in the US is about 102330 people.

Question (g)

Which state has the largest county in terms of population? How many people live in the largest county in terms of population (2 points)?

```r
# State that has the largest county in terms of population
countypop$abbr[which(countypop$pop_2015 == sort(countypop$pop_2015, decreasing = TRUE)[1])]
```

```
## [1] "CA"
```

```r
sort(countypop$pop_2015, decreasing = TRUE)[1]
```

```
## [1] 10170292
```

```r
# Extra credit - using dplyr library
tail(arrange(countypop, pop_2015, sort=TRUE), 1)
```

```
## # A tibble: 1 x 4
##   fips  abbr  county             pop_2015
##   <chr> <chr> <chr>                 <dbl>
## 1 06037 CA    Los Angeles County 10170292
```

The state that has the largest county in terms of population is California (CA). Specifically, it is the Los Angeles County, which has 10,170,292 people live there.

Question (h)

In order to answer the following question, combine the functions `lapply()`, `split()`, `order()`, and `tail()` (or `head()`): What is the largest county in terms of population of each of the states (2 points)?

```r
groups = sort(countypop$abbr)
data = split(countypop[order(countypop$abbr, countypop$pop_2015),], groups)
lapply(data, tail, 1)
```

```
## $AK
## # A tibble: 1 x 4
##   fips  abbr  county                 pop_2015
##   <chr> <chr> <chr>                     <dbl>
## 1 02020 AK    Anchorage Municipality   298695
##
## $AL
## # A tibble: 1 x 4
##   fips  abbr  county           pop_2015
##   <chr> <chr> <chr>               <dbl>
## 1 01073 AL    Jefferson County   660367
##
## $AR
## # A tibble: 1 x 4
##   fips  abbr  county         pop_2015
##   <chr> <chr> <chr>             <dbl>
## 1 05119 AR    Pulaski County   392664
##
## $AZ
## # A tibble: 1 x 4
##   fips  abbr  county          pop_2015
```

```
##   <chr> <chr> <chr>              <dbl>
## 1 04013 AZ    Maricopa County  4167947
##
## $CA
## # A tibble: 1 x 4
##   fips  abbr  county             pop_2015
##   <chr> <chr> <chr>                 <dbl>
## 1 06037 CA    Los Angeles County 10170292
##
## $CO
## # A tibble: 1 x 4
##   fips  abbr  county        pop_2015
##   <chr> <chr> <chr>            <dbl>
## 1 08031 CO    Denver County   682545
##
## $CT
## # A tibble: 1 x 4
##   fips  abbr  county           pop_2015
##   <chr> <chr> <chr>               <dbl>
## 1 09001 CT    Fairfield County   948053
##
## $DC
## # A tibble: 1 x 4
##   fips  abbr  county               pop_2015
##   <chr> <chr> <chr>                   <dbl>
## 1 11001 DC    District of Columbia   672228
##
## $DE
## # A tibble: 1 x 4
##   fips  abbr  county            pop_2015
##   <chr> <chr> <chr>                <dbl>
## 1 10003 DE    New Castle County   556779
##
## $FL
## # A tibble: 1 x 4
##   fips  abbr  county            pop_2015
##   <chr> <chr> <chr>                <dbl>
## 1 12086 FL    Miami-Dade County  2693117
##
## $GA
## # A tibble: 1 x 4
##   fips  abbr  county        pop_2015
##   <chr> <chr> <chr>            <dbl>
## 1 13121 GA    Fulton County  1010562
##
## $HI
## # A tibble: 1 x 4
##   fips  abbr  county          pop_2015
##   <chr> <chr> <chr>              <dbl>
## 1 15003 HI    Honolulu County   998714
##
## $IA
## # A tibble: 1 x 4
##   fips  abbr  county      pop_2015
```

```
##   <chr> <chr> <chr>          <dbl>
## 1 19153 IA    Polk County   467711
##
## $ID
## # A tibble: 1 x 4
##   fips  abbr  county     pop_2015
##   <chr> <chr> <chr>         <dbl>
## 1 16001 ID    Ada County   434211
##
## $IL
## # A tibble: 1 x 4
##   fips  abbr  county      pop_2015
##   <chr> <chr> <chr>          <dbl>
## 1 17031 IL    Cook County  5238216
##
## $IN
## # A tibble: 1 x 4
##   fips  abbr  county        pop_2015
##   <chr> <chr> <chr>            <dbl>
## 1 18097 IN    Marion County   939020
##
## $KS
## # A tibble: 1 x 4
##   fips  abbr  county         pop_2015
##   <chr> <chr> <chr>             <dbl>
## 1 20091 KS    Johnson County   580159
##
## $KY
## # A tibble: 1 x 4
##   fips  abbr  county           pop_2015
##   <chr> <chr> <chr>               <dbl>
## 1 21111 KY    Jefferson County   763623
##
## $LA
## # A tibble: 1 x 4
##   fips  abbr  county                  pop_2015
##   <chr> <chr> <chr>                      <dbl>
## 1 22033 LA    East Baton Rouge Parish   446753
##
## $MA
## # A tibble: 1 x 4
##   fips  abbr  county           pop_2015
##   <chr> <chr> <chr>               <dbl>
## 1 25017 MA    Middlesex County  1585139
##
## $MD
## # A tibble: 1 x 4
##   fips  abbr  county            pop_2015
##   <chr> <chr> <chr>                <dbl>
## 1 24031 MD    Montgomery County  1040116
##
## $ME
## # A tibble: 1 x 4
##   fips  abbr  county            pop_2015
```

```
##   <chr> <chr> <chr>                <dbl>
## 1 23005 ME    Cumberland County   289977
##
## $MI
## # A tibble: 1 x 4
##   fips  abbr  county       pop_2015
##   <chr> <chr> <chr>           <dbl>
## 1 26163 MI    Wayne County  1759335
##
## $MN
## # A tibble: 1 x 4
##   fips  abbr  county          pop_2015
##   <chr> <chr> <chr>              <dbl>
## 1 27053 MN    Hennepin County  1223149
##
## $MO
## # A tibble: 1 x 4
##   fips  abbr  county           pop_2015
##   <chr> <chr> <chr>               <dbl>
## 1 29189 MO    St. Louis County  1003362
##
## $MS
## # A tibble: 1 x 4
##   fips  abbr  county       pop_2015
##   <chr> <chr> <chr>           <dbl>
## 1 28049 MS    Hinds County   242891
##
## $MT
## # A tibble: 1 x 4
##   fips  abbr  county             pop_2015
##   <chr> <chr> <chr>                 <dbl>
## 1 30111 MT    Yellowstone County   157048
##
## $NC
## # A tibble: 1 x 4
##   fips  abbr  county             pop_2015
##   <chr> <chr> <chr>                 <dbl>
## 1 37119 NC    Mecklenburg County  1034070
##
## $ND
## # A tibble: 1 x 4
##   fips  abbr  county      pop_2015
##   <chr> <chr> <chr>          <dbl>
## 1 38017 ND    Cass County   171512
##
## $NE
## # A tibble: 1 x 4
##   fips  abbr  county         pop_2015
##   <chr> <chr> <chr>             <dbl>
## 1 31055 NE    Douglas County   550064
##
## $NH
## # A tibble: 1 x 4
##   fips  abbr  county              pop_2015
```

```
##   <chr> <chr> <chr>                  <dbl>
## 1 33011 NH    Hillsborough County   406678
##
## $NJ
## # A tibble: 1 x 4
##   fips  abbr  county        pop_2015
##   <chr> <chr> <chr>            <dbl>
## 1 34003 NJ    Bergen County   938506
##
## $NM
## # A tibble: 1 x 4
##   fips  abbr  county            pop_2015
##   <chr> <chr> <chr>                <dbl>
## 1 35001 NM    Bernalillo County   676685
##
## $NV
## # A tibble: 1 x 4
##   fips  abbr  county       pop_2015
##   <chr> <chr> <chr>           <dbl>
## 1 32003 NV    Clark County  2114801
##
## $NY
## # A tibble: 1 x 4
##   fips  abbr  county       pop_2015
##   <chr> <chr> <chr>           <dbl>
## 1 36047 NY    Kings County  2636735
##
## $OH
## # A tibble: 1 x 4
##   fips  abbr  county          pop_2015
##   <chr> <chr> <chr>              <dbl>
## 1 39035 OH    Cuyahoga County  1255921
##
## $OK
## # A tibble: 1 x 4
##   fips  abbr  county          pop_2015
##   <chr> <chr> <chr>              <dbl>
## 1 40109 OK    Oklahoma County   776864
##
## $OR
## # A tibble: 1 x 4
##   fips  abbr  county           pop_2015
##   <chr> <chr> <chr>               <dbl>
## 1 41051 OR    Multnomah County   790294
##
## $PA
## # A tibble: 1 x 4
##   fips  abbr  county              pop_2015
##   <chr> <chr> <chr>                  <dbl>
## 1 42101 PA    Philadelphia County  1567442
##
## $RI
## # A tibble: 1 x 4
##   fips  abbr  county            pop_2015
```

```
##   <chr> <chr> <chr>                <dbl>
## 1 44007 RI    Providence County   633473
##
## $SC
## # A tibble: 1 x 4
##   fips  abbr  county            pop_2015
##   <chr> <chr> <chr>                <dbl>
## 1 45045 SC    Greenville County   491863
##
## $SD
## # A tibble: 1 x 4
##   fips  abbr  county           pop_2015
##   <chr> <chr> <chr>               <dbl>
## 1 46099 SD    Minnehaha County   185197
##
## $TN
## # A tibble: 1 x 4
##   fips  abbr  county        pop_2015
##   <chr> <chr> <chr>            <dbl>
## 1 47157 TN    Shelby County   938069
##
## $TX
## # A tibble: 1 x 4
##   fips  abbr  county        pop_2015
##   <chr> <chr> <chr>            <dbl>
## 1 48201 TX    Harris County  4538028
##
## $UT
## # A tibble: 1 x 4
##   fips  abbr  county           pop_2015
##   <chr> <chr> <chr>               <dbl>
## 1 49035 UT    Salt Lake County  1107314
##
## $VA
## # A tibble: 1 x 4
##   fips  abbr  county         pop_2015
##   <chr> <chr> <chr>             <dbl>
## 1 51059 VA    Fairfax County  1142234
##
## $VT
## # A tibble: 1 x 4
##   fips  abbr  county            pop_2015
##   <chr> <chr> <chr>                <dbl>
## 1 50007 VT    Chittenden County   161382
##
## $WA
## # A tibble: 1 x 4
##   fips  abbr  county      pop_2015
##   <chr> <chr> <chr>          <dbl>
## 1 53033 WA    King County  2117125
##
## $WI
## # A tibble: 1 x 4
##   fips  abbr  county           pop_2015
```

```
##   <chr> <chr> <chr>               <dbl>
## 1 55079 WI    Milwaukee County   957735
##
## $WV
## # A tibble: 1 x 4
##   fips  abbr  county         pop_2015
##   <chr> <chr> <chr>             <dbl>
## 1 54039 WV    Kanawha County   188332
##
## $WY
## # A tibble: 1 x 4
##   fips  abbr  county         pop_2015
##   <chr> <chr> <chr>             <dbl>
## 1 56021 WY    Laramie County    97121
```

```
# Extra credit - using dplyr library
countypop %>%
  group_by(abbr) %>%
  filter(pop_2015 == max(pop_2015))
```

```
## # A tibble: 51 x 4
## # Groups:   abbr [51]
##    fips  abbr  county                 pop_2015
##    <chr> <chr> <chr>                     <dbl>
##  1 01073 AL    Jefferson County         660367
##  2 02020 AK    Anchorage Municipality   298695
##  3 04013 AZ    Maricopa County         4167947
##  4 05119 AR    Pulaski County           392664
##  5 06037 CA    Los Angeles County     10170292
##  6 08031 CO    Denver County            682545
##  7 09001 CT    Fairfield County         948053
##  8 10003 DE    New Castle County        556779
##  9 11001 DC    District of Columbia     672228
## 10 12086 FL    Miami-Dade County       2693117
## # ... with 41 more rows
```

Question (i)

What is the average population of the 100 largest counties in the US (2 points)?

```
mean(sort(countypop$pop_2015, decreasing = TRUE)[1:100])
```

```
## [1] 1370079
```

```
# Extra credit - using dplyr library
summarise(slice(arrange(countypop, desc(pop_2015)), 1:100), avg=mean(pop_2015))
```

```
## # A tibble: 1 x 1
##        avg
##      <dbl>
## 1 1370079.
```

Question (j)

How many people live in each of the states (2 points)?

```
aggregate(pop_2015 ~ abbr, data = countypop, sum)
```

```
##    abbr pop_2015
## 1    AK   738432
## 2    AL  4832490
## 3    AR  2978204
## 4    AZ  6774906
## 5    CA 39144818
## 6    CO  5456574
## 7    CT  3590886
## 8    DC   672228
## 9    DE   945934
## 10   FL 20271272
## 11   GA 10214860
## 12   HI  1431603
## 13   IA  3123899
## 14   ID  1654930
## 15   IL 12859995
## 16   IN  6619680
## 17   KS  2911641
## 18   KY  4297380
## 19   LA  4670724
## 20   MA  6794422
## 21   MD  6006401
## 22   ME  1329328
## 23   MI  9922576
## 24   MN  5489594
## 25   MO  6083672
## 26   MS  2992333
## 27   MT  1032949
## 28   NC 10042802
## 29   ND   756927
## 30   NE  1896190
## 31   NH  1330608
## 32   NJ  8958013
## 33   NM  2085109
## 34   NV  2890845
## 35   NY 19795791
## 36   OH 11613423
## 37   OK  3911338
## 38   OR  4028977
## 39   PA 12802503
## 40   RI  1056298
## 41   SC  4896146
## 42   SD   858469
## 43   TN  6600299
## 44   TX 27469114
## 45   UT  2995919
## 46   VA  8382993
## 47   VT   626042
```

```
## 48   WA  7170351
## 49   WI  5771337
## 50   WV  1844128
## 51   WY   586107
```

```
# Extra credit - using dplyr library
countypop %>%
  group_by(abbr) %>%
  summarise(sum(pop_2015))
```

```
## # A tibble: 51 x 2
##    abbr  `sum(pop_2015)`
##    <chr>           <dbl>
##  1 AK             738432
##  2 AL            4832490
##  3 AR            2978204
##  4 AZ            6774906
##  5 CA           39144818
##  6 CO            5456574
##  7 CT            3590886
##  8 DC             672228
##  9 DE             945934
## 10 FL           20271272
## # ... with 41 more rows
```

Question (k)

What is the average population of a county in California (1 point)?

```
mean(countypop[which(countypop$abbr == 'CA'),]$pop_2015)
```

```
## [1] 674910.7
```

```
ceiling(mean(countypop[which(countypop$abbr == 'CA'),]$pop_2015))
```

```
## [1] 674911
```

```
# Extra credit - using dplyr library
summarise(filter(countypop, abbr=="CA"), avg=mean(pop_2015))
```

```
## # A tibble: 1 x 1
##       avg
##     <dbl>
## 1 674911.
```

```
ceiling(summarise(filter(countypop, abbr=="CA"), avg=mean(pop_2015)))
```

```
## # A tibble: 1 x 1
##      avg
##    <dbl>
## 1 674911
```

The average population of a county in California is about 674,911.

Appendix: R Script

```
knitr::opts_chunk$set(echo = TRUE)
rm(list = ls())
library(dplyr)
Animal = sample(c('cat', 'dog', 'cow', 'squirrel'), 100, replace = TRUE)
Color = sample(c('white', 'black', 'brown', 'red'), 100, replace = TRUE)
Attribute = sample(c('big', 'small', 'angry', 'cute', 'finicky'), 100, replace = TRUE)
sum(Animal == 'cat' | Animal == 'dog')
mean(Animal == 'cat')
mean(Animal == 'dog')
mean(Animal == 'cow')
mean(Animal == 'squirrel')
table(Animal, Attribute)
mylist = list(Animal, Color, Attribute)
length(mylist[1])
length(mylist[[1]])
A = matrix(c(1, 2, 3, 4, 5,
             2, 1, 2, 3, 4,
             3, 2, 1, 2, 3,
             4, 3, 2, 1, 2,
             5, 4, 3, 2, 1), byrow = TRUE, nrow = 5)
y = c(7, -1, -3, 5, 17)
det(A)
solve(A, y)
p3matrix = matrix(sample(1:10, size=50, replace=T), nrow = 10, ncol = 5)
p3matrix
m_dist = dist(p3matrix)
head(m_dist)
class(m_dist)
head(as.matrix(m_dist))
load('countypop.RData')
countypop = na.omit(countypop)
head(countypop)
length(countypop$county)
length(unique(countypop$county))
sort(table(countypop$county), decreasing = TRUE)[1:10]
# Extra credit - using dplyr library
head(count(countypop, county, sort = TRUE), 10)
# State that had largest number of counties
sort(table(countypop$abbr), decreasing = TRUE)[1]
# Extra credit - using dplyr library
head(count(countypop, abbr, sort = TRUE), 1)

# State that had smallest number of counties
sort(table(countypop$abbr), decreasing = FALSE)[1]
# Extra credit - using dplyr library
tail(count(countypop, abbr, sort = TRUE), 1)
mean(countypop$pop_2015)
ceiling(mean(countypop$pop_2015))
# Extra credit - using dplyr library
summarise(countypop, avg=mean(pop_2015))
ceiling(summarise(countypop, avg=mean(pop_2015)))
```

```r
# State that has the largest county in terms of population
countypop$abbr[which(countypop$pop_2015 == sort(countypop$pop_2015, decreasing = TRUE)[1])]
sort(countypop$pop_2015, decreasing = TRUE)[1]
# Extra credit - using dplyr library
tail(arrange(countypop, pop_2015, sort=TRUE), 1)
groups = sort(countypop$abbr)
data = split(countypop[order(countypop$abbr, countypop$pop_2015),], groups)
lapply(data, tail, 1)
# Extra credit - using dplyr library
countypop %>%
  group_by(abbr) %>%
  filter(pop_2015 == max(pop_2015))
mean(sort(countypop$pop_2015, decreasing = TRUE)[1:100])
# Extra credit - using dplyr library
summarise(slice(arrange(countypop, desc(pop_2015)), 1:100), avg=mean(pop_2015))
aggregate(pop_2015 ~ abbr, data = countypop, sum)
# Extra credit - using dplyr library
countypop %>%
  group_by(abbr) %>%
  summarise(sum(pop_2015))
mean(countypop[which(countypop$abbr == 'CA'),]$pop_2015)
ceiling(mean(countypop[which(countypop$abbr == 'CA'),]$pop_2015))
# Extra credit - using dplyr library
summarise(filter(countypop, abbr=="CA"), avg=mean(pop_2015))
ceiling(summarise(filter(countypop, abbr=="CA"), avg=mean(pop_2015)))
```