

Intro to Data Science - Lab 3

IST687 Section M003

Professor Anderson

Enter your name here: Hrishikesh Telang

#Select one of the below and add needed information # 1. I did this homework by myself, with help from the book and the professor.

#Instructions: Be prepared to report the whole class at the conclusion of the breakout
#group. Run the necessary code on your own instance of R-Studio. Save the code: It
#will be useful on your homework!

#1. Get an explanation of the contents of the state.x77 data set: help("state.x77")

```
help("state.x77")
```

#2. Create a dataframe from the built-in state.x77 data set, store in a variable named 'dfStates77'

```
dfStates77 <- state.x77  
dfStates77
```

##	Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost
## Alabama	3615	3624	2.1	69.05	15.1	41.3	20
## Alaska	365	6315	1.5	69.31	11.3	66.7	152
## Arizona	2212	4530	1.8	70.55	7.8	58.1	15
## Arkansas	2110	3378	1.9	70.66	10.1	39.9	65
## California	21198	5114	1.1	71.71	10.3	62.6	20
## Colorado	2541	4884	0.7	72.06	6.8	63.9	166
## Connecticut	3100	5348	1.1	72.48	3.1	56.0	139
## Delaware	579	4809	0.9	70.06	6.2	54.6	103
## Florida	8277	4815	1.3	70.66	10.7	52.6	11
## Georgia	4931	4091	2.0	68.54	13.9	40.6	60
## Hawaii	868	4963	1.9	73.60	6.2	61.9	0
## Idaho	813	4119	0.6	71.87	5.3	59.5	126
## Illinois	11197	5107	0.9	70.14	10.3	52.6	127
## Indiana	5313	4458	0.7	70.88	7.1	52.9	122
## Iowa	2861	4628	0.5	72.56	2.3	59.0	140
## Kansas	2280	4669	0.6	72.58	4.5	59.9	114
## Kentucky	3387	3712	1.6	70.10	10.6	38.5	95
## Louisiana	3806	3545	2.8	68.76	13.2	42.2	12
## Maine	1058	3694	0.7	70.39	2.7	54.7	161
## Maryland	4122	5299	0.9	70.22	8.5	52.3	101
## Massachusetts	5814	4755	1.1	71.83	3.3	58.5	103
## Michigan	9111	4751	0.9	70.63	11.1	52.8	125
## Minnesota	3921	4675	0.6	72.96	2.3	57.6	160
## Mississippi	2341	3098	2.4	68.09	12.5	41.0	50
## Missouri	4767	4254	0.8	70.69	9.3	48.8	108
## Montana	746	4347	0.6	70.56	5.0	59.2	155
## Nebraska	1544	4508	0.6	72.60	2.9	59.3	139

## Nevada	590	5149	0.5	69.03	11.5	65.2	188
## New Hampshire	812	4281	0.7	71.23	3.3	57.6	174
## New Jersey	7333	5237	1.1	70.93	5.2	52.5	115
## New Mexico	1144	3601	2.2	70.32	9.7	55.2	120
## New York	18076	4903	1.4	70.55	10.9	52.7	82
## North Carolina	5441	3875	1.8	69.21	11.1	38.5	80
## North Dakota	637	5087	0.8	72.78	1.4	50.3	186
## Ohio	10735	4561	0.8	70.82	7.4	53.2	124
## Oklahoma	2715	3983	1.1	71.42	6.4	51.6	82
## Oregon	2284	4660	0.6	72.13	4.2	60.0	44
## Pennsylvania	11860	4449	1.0	70.43	6.1	50.2	126
## Rhode Island	931	4558	1.3	71.90	2.4	46.4	127
## South Carolina	2816	3635	2.3	67.96	11.6	37.8	65
## South Dakota	681	4167	0.5	72.08	1.7	53.3	172
## Tennessee	4173	3821	1.7	70.11	11.0	41.8	70
## Texas	12237	4188	2.2	70.90	12.2	47.4	35
## Utah	1203	4022	0.6	72.90	4.5	67.3	137
## Vermont	472	3907	0.6	71.64	5.5	57.1	168
## Virginia	4981	4701	1.4	70.08	9.5	47.8	85
## Washington	3559	4864	0.6	71.72	4.3	63.5	32
## West Virginia	1799	3617	1.4	69.48	6.7	41.6	100
## Wisconsin	4589	4468	0.7	72.48	3.0	54.5	149
## Wyoming	376	4566	0.6	70.29	6.9	62.9	173
##	Area						
## Alabama	50708						
## Alaska	566432						
## Arizona	113417						
## Arkansas	51945						
## California	156361						
## Colorado	103766						
## Connecticut	4862						
## Delaware	1982						
## Florida	54090						
## Georgia	58073						
## Hawaii	6425						
## Idaho	82677						
## Illinois	55748						
## Indiana	36097						
## Iowa	55941						
## Kansas	81787						
## Kentucky	39650						
## Louisiana	44930						
## Maine	30920						
## Maryland	9891						
## Massachusetts	7826						
## Michigan	56817						
## Minnesota	79289						
## Mississippi	47296						
## Missouri	68995						
## Montana	145587						
## Nebraska	76483						
## Nevada	109889						
## New Hampshire	9027						
## New Jersey	7521						

```
## New Mexico      121412
## New York        47831
## North Carolina  48798
## North Dakota    69273
## Ohio            40975
## Oklahoma        68782
## Oregon          96184
## Pennsylvania    44966
## Rhode Island    1049
## South Carolina  30225
## South Dakota    75955
## Tennessee       41328
## Texas           262134
## Utah            82096
## Vermont         9267
## Virginia        39780
## Washington      66570
## West Virginia   24070
## Wisconsin       54464
## Wyoming         97203
```

#3. Summarize the variables in your dfStates77 data set - using the summary() function

```
summary(dfStates77)
```

```
##      Population      Income      Illiteracy      Life Exp
##  Min.   : 365      Min.   :3098      Min.   :0.500      Min.   :67.96
##  1st Qu.: 1080      1st Qu.:3993      1st Qu.:0.625      1st Qu.:70.12
##  Median : 2838      Median :4519      Median :0.950      Median :70.67
##  Mean   : 4246      Mean   :4436      Mean   :1.170      Mean   :70.88
##  3rd Qu.: 4968      3rd Qu.:4814      3rd Qu.:1.575      3rd Qu.:71.89
##  Max.   :21198      Max.   :6315      Max.   :2.800      Max.   :73.60
##      Murder      HS Grad      Frost      Area
##  Min.   : 1.400      Min.   :37.80      Min.   : 0.00      Min.   : 1049
##  1st Qu.: 4.350      1st Qu.:48.05      1st Qu.: 66.25      1st Qu.: 36985
##  Median : 6.850      Median :53.25      Median :114.50      Median : 54277
##  Mean   : 7.378      Mean   :53.11      Mean   :104.46      Mean   : 70736
##  3rd Qu.:10.675      3rd Qu.:59.15      3rd Qu.:139.75      3rd Qu.: 81162
##  Max.   :15.100      Max.   :67.30      Max.   :188.00      Max.   :566432
```

#4. Calculate the total population of the U.S. by adding together the populations of each of the individual states in dfStates77. Store the result in a new variable #called totalPop77.

```
totalPop77 <- sum(dfStates77[, 'Population'])
totalPop77
```

```
## [1] 212321
```

#5. Use R code to read a CSV data file directly from the web. Store the dataset #into a new dataframe, called dfStates17. The URL is: " <https://intro-datascience.s3.us-east-2.amazonaws.com/statesNew.csv>" #Note: Use the function read_csv() to read in the data.

#You will need to run library(readr) or library(tidyverse) before you can run #read_csv(). If that generates an error, then you first need to do
#install.packages("readr") or install.packages("tidyverse")

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.4      v dplyr  1.0.7
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   2.0.1      v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
dfStates17 <- read_csv('https://intro-datascience.s3.us-east-2.amazonaws.com/statesNew.csv')
```

```
## Rows: 50 Columns: 19
```

```
## -- Column specification -----
## Delimiter: ","
## chr  (15): state, slug, code, nickname, website, capital_city, capital_url, ...
## dbl  (3): admission_number, population, population_rank
## date (1): admission_date
```

```
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
dfStates17 <- data.frame(dfStates17)
```

#6. Summarize the variables in your new data set, using the summary command.

```
summary(dfStates17)
```

```
##      state          slug          code          nickname
## Length:50      Length:50      Length:50      Length:50
## Class :character Class :character Class :character Class :character
## Mode  :character Mode  :character Mode  :character Mode  :character
##
##
##      website      admission_date      admission_number      capital_city
## Length:50      Min.   :1787-12-07      Min.   : 1.00      Length:50
## Class :character 1st Qu.:1790-08-06      1st Qu.:13.25      Class :character
## Mode  :character Median :1836-10-05      Median :25.50      Mode  :character
##                  Mean   :1840-03-14      Mean   :25.50
##                  3rd Qu.:1874-03-24      3rd Qu.:37.75
```

```
##           Max.    :1959-08-21    Max.    :50.00
## capital_url      population      population_rank constitution_url
## Length:50        Min.    : 582658    Min.    : 1.00    Length:50
## Class :character  1st Qu.: 1857857    1st Qu.:13.25    Class :character
## Mode  :character  Median : 4510382    Median :25.50    Mode  :character
##                Mean   : 6309648    Mean   :25.50
##                3rd Qu.: 6901760    3rd Qu.:37.75
##                Max.    :38332521    Max.    :50.00
## state_flag_url    state_seal_url    map_image_url
## Length:50         Length:50         Length:50
## Class :character  Class :character  Class :character
## Mode  :character  Mode  :character  Mode  :character
##
##
##
## landscape_background_url skyline_background_url twitter_url
## Length:50           Length:50           Length:50
## Class :character     Class :character     Class :character
## Mode  :character     Mode  :character     Mode  :character
##
##
##
## facebook_url
## Length:50
## Class :character
## Mode  :character
##
##
##
```

#7. The data you now have stored in dfStates17 were collected in 2017. As such,
 #about 40 years passed between the two data collections. Calculate the total 2017
 #population of the U.S. in dfStates17 by adding together the populations of each of
 #the individual states. Store the result in a new variable called totalPop17.

```
totalPop17 <- sum(dfStates17[, 'population'])
totalPop17
```

```
## [1] 315482390
```

#8. Create and interpret a ratio of totalPop77 to totalPop17. Check to ensure that the #result makes sense!

```
ratio <- totalPop77/totalPop17
ratio
```

```
## [1] 0.0006730043
```

#End of first breakout

#Create a function that, given population and area, calculates population density by dividing a population value by an area value. Here is the core of the function:

```

popDensity <- function (pop, area) {
  # Add your code below here:
  # Next, divide pop by area and store the result in a
  # variable called popDens
  popDens <- pop/area

  return(popDens) # This provides the function's output
}

```

#9. After you finish your function, make sure to run all of the lines of code in it so that the function becomes known to R.

#10. Make a fresh copy of state.x77 into dfStates77

```

dfStates77 <- data.frame(state.x77)
dfStates77

```

##	Population	Income	Illiteracy	Life.Exp	Murder	HS.Grad	Frost
## Alabama	3615	3624	2.1	69.05	15.1	41.3	20
## Alaska	365	6315	1.5	69.31	11.3	66.7	152
## Arizona	2212	4530	1.8	70.55	7.8	58.1	15
## Arkansas	2110	3378	1.9	70.66	10.1	39.9	65
## California	21198	5114	1.1	71.71	10.3	62.6	20
## Colorado	2541	4884	0.7	72.06	6.8	63.9	166
## Connecticut	3100	5348	1.1	72.48	3.1	56.0	139
## Delaware	579	4809	0.9	70.06	6.2	54.6	103
## Florida	8277	4815	1.3	70.66	10.7	52.6	11
## Georgia	4931	4091	2.0	68.54	13.9	40.6	60
## Hawaii	868	4963	1.9	73.60	6.2	61.9	0
## Idaho	813	4119	0.6	71.87	5.3	59.5	126
## Illinois	11197	5107	0.9	70.14	10.3	52.6	127
## Indiana	5313	4458	0.7	70.88	7.1	52.9	122
## Iowa	2861	4628	0.5	72.56	2.3	59.0	140
## Kansas	2280	4669	0.6	72.58	4.5	59.9	114
## Kentucky	3387	3712	1.6	70.10	10.6	38.5	95
## Louisiana	3806	3545	2.8	68.76	13.2	42.2	12
## Maine	1058	3694	0.7	70.39	2.7	54.7	161
## Maryland	4122	5299	0.9	70.22	8.5	52.3	101
## Massachusetts	5814	4755	1.1	71.83	3.3	58.5	103
## Michigan	9111	4751	0.9	70.63	11.1	52.8	125
## Minnesota	3921	4675	0.6	72.96	2.3	57.6	160
## Mississippi	2341	3098	2.4	68.09	12.5	41.0	50
## Missouri	4767	4254	0.8	70.69	9.3	48.8	108
## Montana	746	4347	0.6	70.56	5.0	59.2	155
## Nebraska	1544	4508	0.6	72.60	2.9	59.3	139
## Nevada	590	5149	0.5	69.03	11.5	65.2	188
## New Hampshire	812	4281	0.7	71.23	3.3	57.6	174
## New Jersey	7333	5237	1.1	70.93	5.2	52.5	115
## New Mexico	1144	3601	2.2	70.32	9.7	55.2	120
## New York	18076	4903	1.4	70.55	10.9	52.7	82
## North Carolina	5441	3875	1.8	69.21	11.1	38.5	80
## North Dakota	637	5087	0.8	72.78	1.4	50.3	186
## Ohio	10735	4561	0.8	70.82	7.4	53.2	124
## Oklahoma	2715	3983	1.1	71.42	6.4	51.6	82

## Oregon	2284	4660	0.6	72.13	4.2	60.0	44
## Pennsylvania	11860	4449	1.0	70.43	6.1	50.2	126
## Rhode Island	931	4558	1.3	71.90	2.4	46.4	127
## South Carolina	2816	3635	2.3	67.96	11.6	37.8	65
## South Dakota	681	4167	0.5	72.08	1.7	53.3	172
## Tennessee	4173	3821	1.7	70.11	11.0	41.8	70
## Texas	12237	4188	2.2	70.90	12.2	47.4	35
## Utah	1203	4022	0.6	72.90	4.5	67.3	137
## Vermont	472	3907	0.6	71.64	5.5	57.1	168
## Virginia	4981	4701	1.4	70.08	9.5	47.8	85
## Washington	3559	4864	0.6	71.72	4.3	63.5	32
## West Virginia	1799	3617	1.4	69.48	6.7	41.6	100
## Wisconsin	4589	4468	0.7	72.48	3.0	54.5	149
## Wyoming	376	4566	0.6	70.29	6.9	62.9	173
##	Area						
## Alabama	50708						
## Alaska	566432						
## Arizona	113417						
## Arkansas	51945						
## California	156361						
## Colorado	103766						
## Connecticut	4862						
## Delaware	1982						
## Florida	54090						
## Georgia	58073						
## Hawaii	6425						
## Idaho	82677						
## Illinois	55748						
## Indiana	36097						
## Iowa	55941						
## Kansas	81787						
## Kentucky	39650						
## Louisiana	44930						
## Maine	30920						
## Maryland	9891						
## Massachusetts	7826						
## Michigan	56817						
## Minnesota	79289						
## Mississippi	47296						
## Missouri	68995						
## Montana	145587						
## Nebraska	76483						
## Nevada	109889						
## New Hampshire	9027						
## New Jersey	7521						
## New Mexico	121412						
## New York	47831						
## North Carolina	48798						
## North Dakota	69273						
## Ohio	40975						
## Oklahoma	68782						
## Oregon	96184						
## Pennsylvania	44966						
## Rhode Island	1049						

```
## South Carolina 30225
## South Dakota 75955
## Tennessee 41328
## Texas 262134
## Utah 82096
## Vermont 9267
## Virginia 39780
## Washington 66570
## West Virginia 24070
## Wisconsin 54464
## Wyoming 97203
```

#11. Store the population vector in a variable called tempPop. Adjust the tempPop as needed (based on your analysis at the end of the first breakout)

```
tempPop <- dfStates77[, 'Population'] * 1000
tempPop
```

```
## [1] 3615000 365000 2212000 2110000 21198000 2541000 3100000 579000
## [9] 8277000 4931000 868000 813000 11197000 5313000 2861000 2280000
## [17] 3387000 3806000 1058000 4122000 5814000 9111000 3921000 2341000
## [25] 4767000 746000 1544000 590000 812000 7333000 1144000 18076000
## [33] 5441000 637000 10735000 2715000 2284000 11860000 931000 2816000
## [41] 681000 4173000 12237000 1203000 472000 4981000 3559000 1799000
## [49] 4589000 376000
```

#12. Store the area vector in a variable, called tempArea

```
tempArea <- dfStates77[, 'Area']
tempArea
```

```
## [1] 50708 566432 113417 51945 156361 103766 4862 1982 54090 58073
## [11] 6425 82677 55748 36097 55941 81787 39650 44930 30920 9891
## [21] 7826 56817 79289 47296 68995 145587 76483 109889 9027 7521
## [31] 121412 47831 48798 69273 40975 68782 96184 44966 1049 30225
## [41] 75955 41328 262134 82096 9267 39780 66570 24070 54464 97203
```

#13. Now use tempPop and tempArea to call your function: popDensity(tempPop, tempArea)

```
popDensity(tempPop, tempArea)
```

```
## [1] 71.2905261 0.6443845 19.5032491 40.6198864 135.5708904 24.4877898
## [7] 637.5976964 292.1291625 153.0227399 84.9103714 135.0972763 9.8334482
## [13] 200.8502547 147.1867468 51.1431687 27.8772910 85.4224464 84.7095482
## [19] 34.2173351 416.7424932 742.9082545 160.3569354 49.4520047 49.4967862
## [25] 69.0919632 5.1240839 20.1874926 5.3690542 89.9523651 975.0033240
## [31] 9.4224624 377.9139052 111.5004713 9.1955019 261.9890177 39.4725364
## [37] 23.7461532 263.7548370 887.5119161 93.1679074 8.9658350 100.9727062
## [43] 46.6822312 14.6535763 50.9334197 125.2136752 53.4625207 74.7403407
## [49] 84.2574912 3.8681934
```

#14. Store the results from the previous task in a column of the dfStates77 dataframe, called popDensity.


```
dfStates77$popDensity <- popDensity(tempPop, tempArea)
dfStates77
```

##	Population	Income	Illiteracy	Life.Exp	Murder	HS.Grad	Frost
## Alabama	3615	3624	2.1	69.05	15.1	41.3	20
## Alaska	365	6315	1.5	69.31	11.3	66.7	152
## Arizona	2212	4530	1.8	70.55	7.8	58.1	15
## Arkansas	2110	3378	1.9	70.66	10.1	39.9	65
## California	21198	5114	1.1	71.71	10.3	62.6	20
## Colorado	2541	4884	0.7	72.06	6.8	63.9	166
## Connecticut	3100	5348	1.1	72.48	3.1	56.0	139
## Delaware	579	4809	0.9	70.06	6.2	54.6	103
## Florida	8277	4815	1.3	70.66	10.7	52.6	11
## Georgia	4931	4091	2.0	68.54	13.9	40.6	60
## Hawaii	868	4963	1.9	73.60	6.2	61.9	0
## Idaho	813	4119	0.6	71.87	5.3	59.5	126
## Illinois	11197	5107	0.9	70.14	10.3	52.6	127
## Indiana	5313	4458	0.7	70.88	7.1	52.9	122
## Iowa	2861	4628	0.5	72.56	2.3	59.0	140
## Kansas	2280	4669	0.6	72.58	4.5	59.9	114
## Kentucky	3387	3712	1.6	70.10	10.6	38.5	95
## Louisiana	3806	3545	2.8	68.76	13.2	42.2	12
## Maine	1058	3694	0.7	70.39	2.7	54.7	161
## Maryland	4122	5299	0.9	70.22	8.5	52.3	101
## Massachusetts	5814	4755	1.1	71.83	3.3	58.5	103
## Michigan	9111	4751	0.9	70.63	11.1	52.8	125
## Minnesota	3921	4675	0.6	72.96	2.3	57.6	160
## Mississippi	2341	3098	2.4	68.09	12.5	41.0	50
## Missouri	4767	4254	0.8	70.69	9.3	48.8	108
## Montana	746	4347	0.6	70.56	5.0	59.2	155
## Nebraska	1544	4508	0.6	72.60	2.9	59.3	139
## Nevada	590	5149	0.5	69.03	11.5	65.2	188
## New Hampshire	812	4281	0.7	71.23	3.3	57.6	174
## New Jersey	7333	5237	1.1	70.93	5.2	52.5	115
## New Mexico	1144	3601	2.2	70.32	9.7	55.2	120
## New York	18076	4903	1.4	70.55	10.9	52.7	82
## North Carolina	5441	3875	1.8	69.21	11.1	38.5	80
## North Dakota	637	5087	0.8	72.78	1.4	50.3	186
## Ohio	10735	4561	0.8	70.82	7.4	53.2	124
## Oklahoma	2715	3983	1.1	71.42	6.4	51.6	82
## Oregon	2284	4660	0.6	72.13	4.2	60.0	44
## Pennsylvania	11860	4449	1.0	70.43	6.1	50.2	126
## Rhode Island	931	4558	1.3	71.90	2.4	46.4	127
## South Carolina	2816	3635	2.3	67.96	11.6	37.8	65
## South Dakota	681	4167	0.5	72.08	1.7	53.3	172
## Tennessee	4173	3821	1.7	70.11	11.0	41.8	70
## Texas	12237	4188	2.2	70.90	12.2	47.4	35
## Utah	1203	4022	0.6	72.90	4.5	67.3	137
## Vermont	472	3907	0.6	71.64	5.5	57.1	168
## Virginia	4981	4701	1.4	70.08	9.5	47.8	85
## Washington	3559	4864	0.6	71.72	4.3	63.5	32
## West Virginia	1799	3617	1.4	69.48	6.7	41.6	100
## Wisconsin	4589	4468	0.7	72.48	3.0	54.5	149

## Wyoming	376	4566	0.6	70.29	6.9	62.9	173
##	Area	popDensity					
## Alabama	50708	71.2905261					
## Alaska	566432	0.6443845					
## Arizona	113417	19.5032491					
## Arkansas	51945	40.6198864					
## California	156361	135.5708904					
## Colorado	103766	24.4877898					
## Connecticut	4862	637.5976964					
## Delaware	1982	292.1291625					
## Florida	54090	153.0227399					
## Georgia	58073	84.9103714					
## Hawaii	6425	135.0972763					
## Idaho	82677	9.8334482					
## Illinois	55748	200.8502547					
## Indiana	36097	147.1867468					
## Iowa	55941	51.1431687					
## Kansas	81787	27.8772910					
## Kentucky	39650	85.4224464					
## Louisiana	44930	84.7095482					
## Maine	30920	34.2173351					
## Maryland	9891	416.7424932					
## Massachusetts	7826	742.9082545					
## Michigan	56817	160.3569354					
## Minnesota	79289	49.4520047					
## Mississippi	47296	49.4967862					
## Missouri	68995	69.0919632					
## Montana	145587	5.1240839					
## Nebraska	76483	20.1874926					
## Nevada	109889	5.3690542					
## New Hampshire	9027	89.9523651					
## New Jersey	7521	975.0033240					
## New Mexico	121412	9.4224624					
## New York	47831	377.9139052					
## North Carolina	48798	111.5004713					
## North Dakota	69273	9.1955019					
## Ohio	40975	261.9890177					
## Oklahoma	68782	39.4725364					
## Oregon	96184	23.7461532					
## Pennsylvania	44966	263.7548370					
## Rhode Island	1049	887.5119161					
## South Carolina	30225	93.1679074					
## South Dakota	75955	8.9658350					
## Tennessee	41328	100.9727062					
## Texas	262134	46.6822312					
## Utah	82096	14.6535763					
## Vermont	9267	50.9334197					
## Virginia	39780	125.2136752					
## Washington	66570	53.4625207					
## West Virginia	24070	74.7403407					
## Wisconsin	54464	84.2574912					
## Wyoming	97203	3.8681934					

#15. Use `which.max()` and `which.min()` to reveal which is the most densely populated and which is the

least densely populated state. #Make sure that you understand the number that is revealed as well as the name of the state.

```
dfStates77[which.max(dfStates77$popDensity),]
```

```
##           Population Income Illiteracy Life.Exp Murder HS.Grad Frost Area
## New Jersey      7333   5237         1.1   70.93    5.2   52.5   115 7521
##           popDensity
## New Jersey    975.0033
```

```
dfStates77[which.min(dfStates77$popDensity),]
```

```
##           Population Income Illiteracy Life.Exp Murder HS.Grad Frost Area
## Alaska          365   6315         1.5   69.31   11.3   66.7   152 566432
##           popDensity
## Alaska    0.6443845
```

#16. Using tidyverse, sort the dataframe using the popDensity attribute, then using #the slice function, show the first row in the sorted database.

```
library('tidyverse')
sortedDF <- dfStates77 %>% arrange(popDensity)
sortedDF <- slice(sortedDF, 1)
sortedDF
```

```
##           Population Income Illiteracy Life.Exp Murder HS.Grad Frost Area
## Alaska          365   6315         1.5   69.31   11.3   66.7   152 566432
##           popDensity
## Alaska    0.6443845
```

#17. How was the dataframe sorted (was the minimum first or the maximum)?

```
#It was the minimum first.
```