Test1 OVM

Olivia Mahony

10/3/2021

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

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

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr)  
library(ggplot2)

## 1

Using the midwest data frame produce a data table that shows output for the Ohio (OH) only. Produce correct output by using two methods. First use the piping method and then use the assignment method.

midwest%>%  
 filter(state == "OH")%>%  
print

## # A tibble: 88 × 28  
## PID county state area poptotal popdensity popwhite popblack popamerindian  
## <int> <chr> <chr> <dbl> <int> <dbl> <int> <int> <int>  
## 1 2009 ADAMS OH 0.035 25371 725. 25212 47 67  
## 2 2010 ALLEN OH 0.024 109755 4573. 96177 12313 202  
## 3 2011 ASHLAND OH 0.025 47507 1900. 46686 460 49  
## 4 2012 ASHTABULA OH 0.041 99821 2435. 95465 3138 196  
## 5 2013 ATHENS OH 0.03 59549 1985. 56163 1678 167  
## 6 2014 AUGLAIZE OH 0.024 44585 1858. 44225 66 50  
## 7 2015 BELMONT OH 0.031 71074 2293. 69520 1308 81  
## 8 2016 BROWN OH 0.028 34966 1249. 34487 406 28  
## 9 2017 BUTLER OH 0.028 291479 10410. 274892 13134 379  
## 10 2018 CARROLL OH 0.024 26521 1105. 26254 135 65  
## # … with 78 more rows, and 19 more variables: popasian <int>, popother <int>,  
## # percwhite <dbl>, percblack <dbl>, percamerindan <dbl>, percasian <dbl>,  
## # percother <dbl>, popadults <int>, perchsd <dbl>, percollege <dbl>,  
## # percprof <dbl>, poppovertyknown <int>, percpovertyknown <dbl>,  
## # percbelowpoverty <dbl>, percchildbelowpovert <dbl>, percadultpoverty <dbl>,  
## # percelderlypoverty <dbl>, inmetro <int>, category <chr>

midwest1 <- filter(midwest, state == "OH")  
midwest1

## # A tibble: 88 × 28  
## PID county state area poptotal popdensity popwhite popblack popamerindian  
## <int> <chr> <chr> <dbl> <int> <dbl> <int> <int> <int>  
## 1 2009 ADAMS OH 0.035 25371 725. 25212 47 67  
## 2 2010 ALLEN OH 0.024 109755 4573. 96177 12313 202  
## 3 2011 ASHLAND OH 0.025 47507 1900. 46686 460 49  
## 4 2012 ASHTABULA OH 0.041 99821 2435. 95465 3138 196  
## 5 2013 ATHENS OH 0.03 59549 1985. 56163 1678 167  
## 6 2014 AUGLAIZE OH 0.024 44585 1858. 44225 66 50  
## 7 2015 BELMONT OH 0.031 71074 2293. 69520 1308 81  
## 8 2016 BROWN OH 0.028 34966 1249. 34487 406 28  
## 9 2017 BUTLER OH 0.028 291479 10410. 274892 13134 379  
## 10 2018 CARROLL OH 0.024 26521 1105. 26254 135 65  
## # … with 78 more rows, and 19 more variables: popasian <int>, popother <int>,  
## # percwhite <dbl>, percblack <dbl>, percamerindan <dbl>, percasian <dbl>,  
## # percother <dbl>, popadults <int>, perchsd <dbl>, percollege <dbl>,  
## # percprof <dbl>, poppovertyknown <int>, percpovertyknown <dbl>,  
## # percbelowpoverty <dbl>, percchildbelowpovert <dbl>, percadultpoverty <dbl>,  
## # percelderlypoverty <dbl>, inmetro <int>, category <chr>

## 2

Using the midwest data frame, produce a data table that shows white population that is greater than 50,000 but less than 90,000 for the state of Indiana (IN)

midwest%>%  
 filter(state == "IN", popwhite >50000, popwhite <90000)%>%  
print

## # A tibble: 10 × 28  
## PID county state area poptotal popdensity popwhite popblack popamerindian  
## <int> <chr> <chr> <dbl> <int> <dbl> <int> <int> <int>  
## 1 665 BARTHOLOMEW IN 0.022 63657 2894. 61774 1005 97  
## 2 672 CLARK IN 0.022 87777 3990. 82289 4703 192  
## 3 684 FLOYD IN 0.009 64404 7156 61415 2642 92  
## 4 689 GRANT IN 0.024 74169 3090. 67817 5047 298  
## 5 694 HENDRICKS IN 0.024 75717 3155. 74519 685 157  
## 6 696 HOWARD IN 0.016 80827 5052. 75420 4398 226  
## 7 703 JOHNSON IN 0.018 88109 4895. 86455 845 139  
## 8 705 KOSCIUSKO IN 0.032 65294 2040. 64058 309 118  
## 9 717 MORGAN IN 0.024 55920 2330 55635 9 137  
## 10 751 WAYNE IN 0.024 71951 2998. 67532 3795 153  
## # … with 19 more variables: popasian <int>, popother <int>, percwhite <dbl>,  
## # percblack <dbl>, percamerindan <dbl>, percasian <dbl>, percother <dbl>,  
## # popadults <int>, perchsd <dbl>, percollege <dbl>, percprof <dbl>,  
## # poppovertyknown <int>, percpovertyknown <dbl>, percbelowpoverty <dbl>,  
## # percchildbelowpovert <dbl>, percadultpoverty <dbl>,  
## # percelderlypoverty <dbl>, inmetro <int>, category <chr>

midwest2 <- filter(midwest,state == "IN", popwhite >50000, popwhite <90000)  
midwest2

## # A tibble: 10 × 28  
## PID county state area poptotal popdensity popwhite popblack popamerindian  
## <int> <chr> <chr> <dbl> <int> <dbl> <int> <int> <int>  
## 1 665 BARTHOLOMEW IN 0.022 63657 2894. 61774 1005 97  
## 2 672 CLARK IN 0.022 87777 3990. 82289 4703 192  
## 3 684 FLOYD IN 0.009 64404 7156 61415 2642 92  
## 4 689 GRANT IN 0.024 74169 3090. 67817 5047 298  
## 5 694 HENDRICKS IN 0.024 75717 3155. 74519 685 157  
## 6 696 HOWARD IN 0.016 80827 5052. 75420 4398 226  
## 7 703 JOHNSON IN 0.018 88109 4895. 86455 845 139  
## 8 705 KOSCIUSKO IN 0.032 65294 2040. 64058 309 118  
## 9 717 MORGAN IN 0.024 55920 2330 55635 9 137  
## 10 751 WAYNE IN 0.024 71951 2998. 67532 3795 153  
## # … with 19 more variables: popasian <int>, popother <int>, percwhite <dbl>,  
## # percblack <dbl>, percamerindan <dbl>, percasian <dbl>, percother <dbl>,  
## # popadults <int>, perchsd <dbl>, percollege <dbl>, percprof <dbl>,  
## # poppovertyknown <int>, percpovertyknown <dbl>, percbelowpoverty <dbl>,  
## # percchildbelowpovert <dbl>, percadultpoverty <dbl>,  
## # percelderlypoverty <dbl>, inmetro <int>, category <chr>

## 3

Using the midwest data , produce a data frame (20 observations)that shows only the variables state, county, poptotal , popamerindian, percamerindian for the state of Indiana. Also your data frame should show popamerindian in descending order.Which county in Indiana has the highest number of Native Americans?

Marion

midwest%>%  
 select(state, county, poptotal, popamerindian, percamerindan)%>%  
 filter(state == "IN")%>%  
 arrange(desc(popamerindian))%>%  
print(n=20)

## # A tibble: 92 × 5  
## state county poptotal popamerindian percamerindan  
## <chr> <chr> <int> <int> <dbl>  
## 1 IN MARION 797159 1698 0.213  
## 2 IN ALLEN 300836 892 0.297  
## 3 IN LAKE 475594 865 0.182  
## 4 IN ST JOSEPH 247052 846 0.342  
## 5 IN MIAMI 36897 571 1.55   
## 6 IN ELKHART 156198 453 0.290  
## 7 IN TIPPECANOE 130598 320 0.245  
## 8 IN MADISON 130669 299 0.229  
## 9 IN GRANT 74169 298 0.402  
## 10 IN VIGO 106107 297 0.280  
## 11 IN VANDERBURGH 165058 284 0.172  
## 12 IN DELAWARE 119659 274 0.229  
## 13 IN LA PORTE 107066 259 0.242  
## 14 IN WABASH 35069 259 0.739  
## 15 IN PORTER 128932 243 0.188  
## 16 IN HOWARD 80827 226 0.280  
## 17 IN MONROE 108978 216 0.198  
## 18 IN CLARK 87777 192 0.219  
## 19 IN HAMILTON 108936 163 0.150  
## 20 IN HENDRICKS 75717 157 0.207  
## # … with 72 more rows

## 4

Using the midwest data and dplyr functions, create a data frame for only the state of Michigan (MI) showing those counties that have a known poverty population that is greater than 10,000 and a percentage of professionals that is greater than 10 percent. Only select variables that you need for the data frame, Your output should only have four variables and six (rows) / observations.

midwest%>%  
 select(state, county, poppovertyknown, percprof)%>%  
 filter(state == "MI", poppovertyknown >10000, percprof > 10.)%>%  
print

## # A tibble: 6 × 4  
## state county poppovertyknown percprof  
## <chr> <chr> <int> <dbl>  
## 1 MI INGHAM 261491 12.9  
## 2 MI ISABELLA 48498 10.0  
## 3 MI KALAMAZOO 212670 10.9  
## 4 MI MIDLAND 74135 11.2  
## 5 MI OAKLAND 1070844 11.2  
## 6 MI WASHTENAW 261261 20.8

## 5

Using the midwest data and dplyr commands and functions, write r code that will show the mean of the poverty population for the counties of each state.

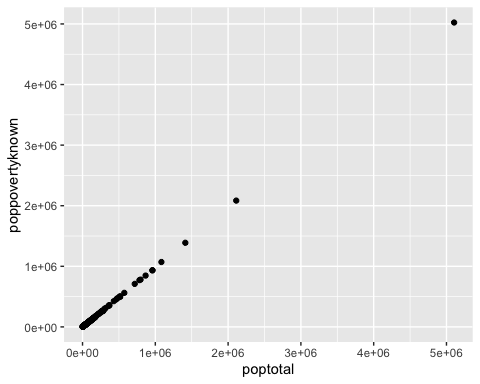
midwest%>%  
 group\_by(state) %>%  
 summarise(MeanPov = mean(poppovertyknown))%>%  
print

## # A tibble: 5 × 2  
## state MeanPov  
## <chr> <dbl>  
## 1 IL 109253.  
## 2 IN 58396.  
## 3 MI 109362.  
## 4 OH 120163.  
## 5 WI 66029.

## 6

Using the midwest data, produce a scatter plot showing a relationship between the variables poppovertyknown and poptotal (Let poptotal = x and poppovertyknown = y).

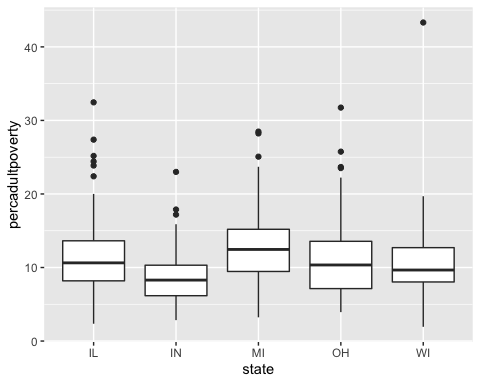
ggplot(data=midwest) +  
 geom\_point(mapping = aes(x = poptotal, y=poppovertyknown))



## 7

Using the midwest data, write r code that will produce the following side by side boxplots.

ggplot(data=midwest) +  
 geom\_boxplot(mapping = aes(x = state, y = percadultpoverty))



## 8

Using the midwest data, write r code that will produce a facet plot that shows scatter plots (red data points) with respect to the levels for the variable state. Also add code that will generate regression lines through your scatter plots that feature x = percollege and y = percprof. Title your facet plot “College/Professional Work Scatter Plots”

ggplot(data=midwest)+  
 geom\_point(mapping = aes(x=percollege, y = percprof, color = "red"))+  
 geom\_smooth(method = lm, mapping = aes(x = percollege, y=percprof))+   
 facet\_grid(~state) + ggtitle("College/Professional Work Scatter Plots")

## `geom\_smooth()` using formula 'y ~ x'

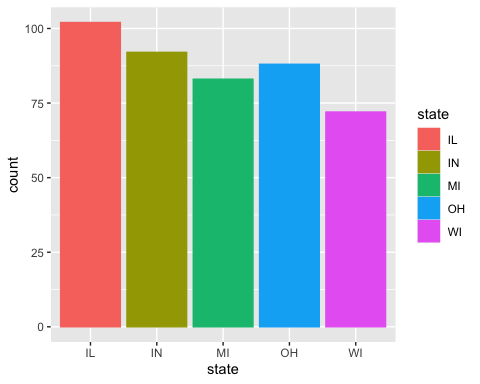


## 9

Using the midwest data frame, create a bar graph that shows the different counts for each state in the data set. Your bars should have different colors. Which state has the highest count?

IL

ggplot(data= midwest) +  
 geom\_bar(mapping = aes(x = state, color = state, fill = state))



## 10

The formula used to find the volume of a cylinder is V = pi times r squared and the formula to find the Surface Area of a cylinder is A = 2(pi times r times h + pi times r squared) Using the formal notation and process for writing a function, as demonstrated in class, to write a function that will calculate the Volume and the Surface Area of a given cylinder. Test your function by calculating answers for r = 5 and h = 10.

V <- function(r)  
{pi\*r^2  
 return(pi\*r^2)  
}  
V(5)

## [1] 78.53982

SA<- function(r,h)  
{2\*((pi\*r\*h)+(pi\*r^2))  
 return(2\*((pi\*r\*h)+(pi\*r^2)))  
}  
SA(5,10)

## [1] 471.2389