HW_3_Team_GA

2022-09-23

Use the following libraries in order to write code and execute output.

library(ggplot2) library(tidyverse)

Show and use R coding to answer the following questions. (Use Tidyverse methods to generate graphs and plots)

1) Explore the Midwest data frame.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
data("midwest")
```

a) How many rows, columns, and variables are in the Midwest data frame?

```
glimpse(midwest)
```

```
## Rows: 437
## Columns: 28
## $ PID                 <int> 561, 562, 563, 564, 565, 566, 567, 568, 569, 570,~
## $ county              <chr> "ADAMS", "ALEXANDER", "BOND", "BOONE", "BROWN", "~
## $ state               <chr> "IL", "IL", "IL", "IL", "IL", "IL", "IL", "IL", "~
## $ area                <dbl> 0.052, 0.014, 0.022, 0.017, 0.018, 0.050, 0.017, ~
## $ poptotal            <int> 66090, 10626, 14991, 30806, 5836, 35688, 5322, 16~
## $ popdensity          <dbl> 1270.9615, 759.0000, 681.4091, 1812.1176, 324.222~
## $ popwhite            <int> 63917, 7054, 14477, 29344, 5264, 35157, 5298, 165~
## $ popblack            <int> 1702, 3496, 429, 127, 547, 50, 1, 111, 16, 16559,~
## $ popamerindian       <int> 98, 19, 35, 46, 14, 65, 8, 30, 8, 331, 51, 26, 17~
## $ popasian            <int> 249, 48, 16, 150, 5, 195, 15, 61, 23, 8033, 89, 3~
## $ popother            <int> 124, 9, 34, 1139, 6, 221, 0, 84, 6, 1596, 20, 7, ~
## $ percwhite           <dbl> 96.71206, 66.38434, 96.57128, 95.25417, 90.19877,~
## $ percblack           <dbl> 2.57527614, 32.90043290, 2.86171703, 0.41225735, ~
## $ percamerindan       <dbl> 0.14828264, 0.17880670, 0.23347342, 0.14932156, 0~
## $ percasian           <dbl> 0.37675897, 0.45172219, 0.10673071, 0.48691813, 0~
## $ percother           <dbl> 0.18762294, 0.08469791, 0.22680275, 3.69733169, 0~
## $ popadults           <int> 43298, 6724, 9669, 19272, 3979, 23444, 3583, 1132~
## $ perchsd             <dbl> 75.10740, 59.72635, 69.33499, 75.47219, 68.86152,~
## $ percollege          <dbl> 19.63139, 11.24331, 17.03382, 17.27895, 14.47600,~
```

```
## $ percprof              <dbl> 4.355859, 2.870315, 4.488572, 4.197800, 3.367680,~
## $ poppovertyknown       <int> 63628, 10529, 14235, 30337, 4815, 35107, 5241, 16~
## $ percpovertyknown      <dbl> 96.27478, 99.08714, 94.95697, 98.47757, 82.50514,~
## $ percbelowpoverty      <dbl> 13.151443, 32.244278, 12.068844, 7.209019, 13.520~
## $ percchildbelowpovert <dbl> 18.011717, 45.826514, 14.036061, 11.179536, 13.02~
## $ percadultpoverty      <dbl> 11.009776, 27.385647, 10.852090, 5.536013, 11.143~
## $ percelderlypoverty    <dbl> 12.443812, 25.228976, 12.697410, 6.217047, 19.200~
## $ inmetro               <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0~
## $ category              <chr> "AAR", "LHR", "AAR", "ALU", "AAR", "AAR", "LAR", ~
```

```
?midwest
```

Rows: 437 Columns: 28

b) Name three categorical variables in the data frame.

```
split(names(midwest),sapply(midwest, function(x) paste(class(x), collapse=" ")))
```

```
## $character
## [1] "county"   "state"    "category"
##
## $integer
##  [1] "PID"             "poptotal"        "popwhite"        "popblack"
##  [5] "popamerindian"   "popasian"        "popother"        "popadults"
##  [9] "poppovertyknown" "inmetro"
##
## $numeric
##  [1] "area"                 "popdensity"           "percwhite"
##  [4] "percblack"            "percamerindan"        "percasian"
##  [7] "percother"            "perchsd"              "percollege"
## [10] "percprof"             "percpovertyknown"     "percbelowpoverty"
## [13] "percchildbelowpovert" "percadultpoverty"     "percelderlypoverty"
```

Categorical variable: 1. “county”: County name
2. “state”: State to which county belongs to. 3. “category”: Miscellaneous 4. “inmetro”: County considered in a metro area

c) Give a description for the variable percollege.

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
describe(midwest$percollege)
```

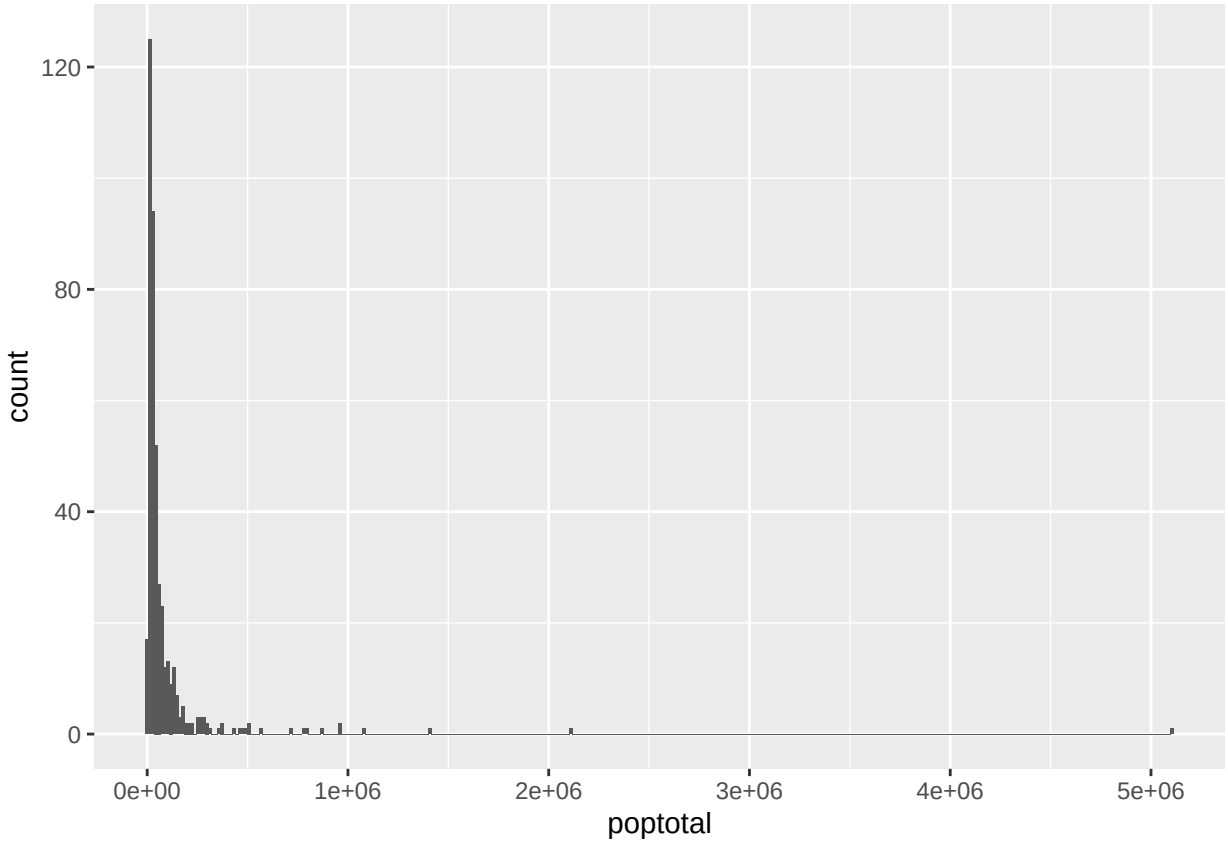
```
##    vars   n  mean   sd median trimmed  mad  min   max range skew kurtosis  se
## X1    1 437 18.27 6.26   16.8   17.42 4.54 7.34 48.08 40.74 1.56     3.08 0.3
```

percollege: Percent college educated.

Use the Midwest data frame for problems 2,3,4,5,6,7, and 9

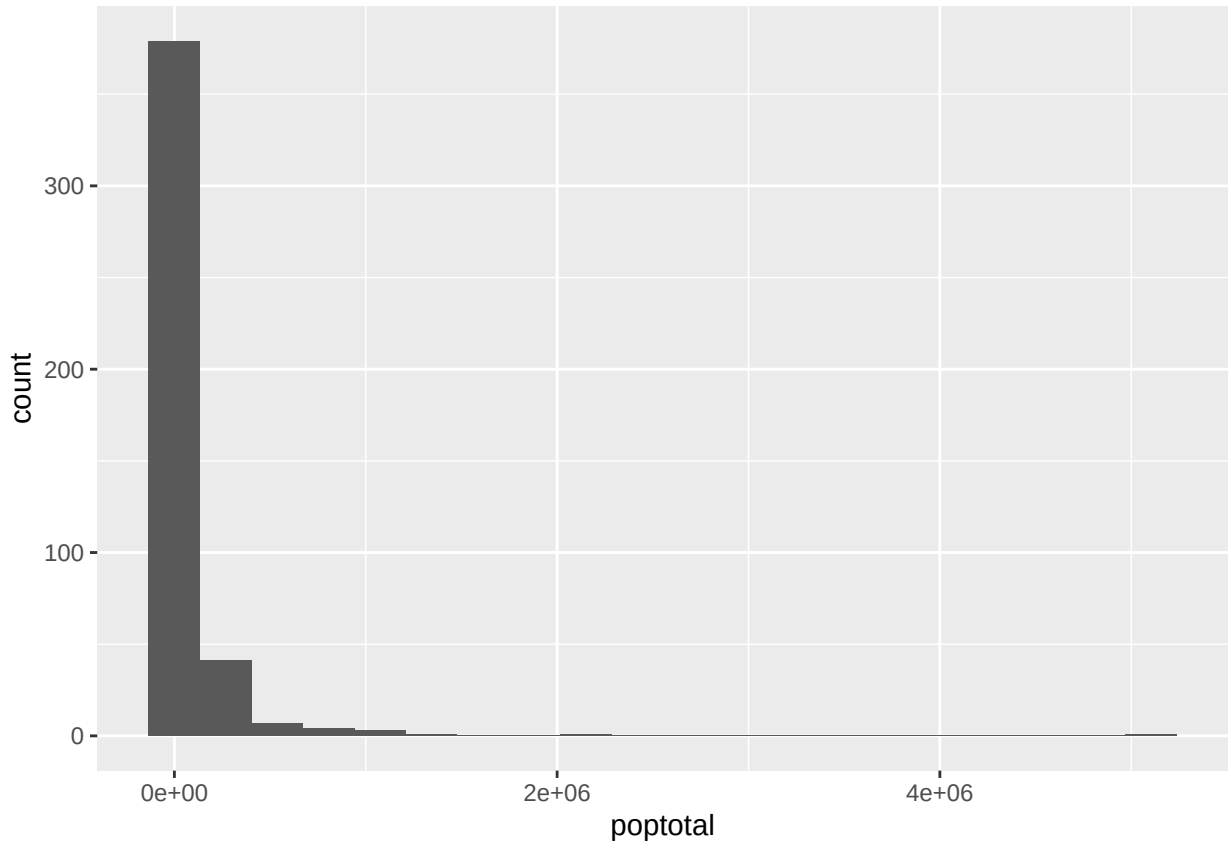
2) Write R code to produce a histogram for the variable poptotal.

```
library(ggplot2)
bw <- 2 * IQR(midwest$poptotal) / length(midwest$poptotal)^(1/3)  # Freedman-Diaconis rule
ggplot(midwest, aes(x=poptotal))+
  geom_histogram(binwidth= bw, bins = sqrt(437))
```



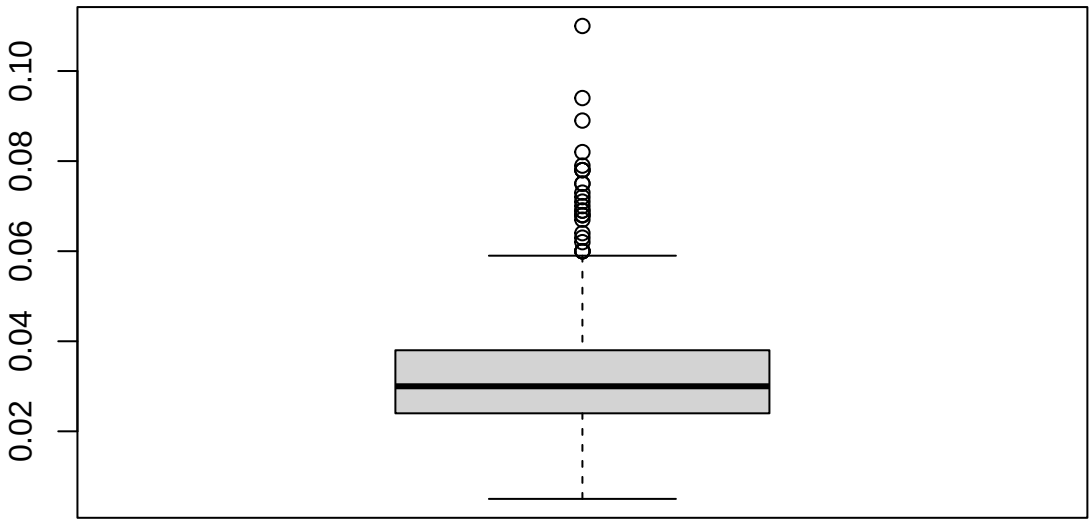

```
# Based on the rule,
#it isn't much clear; if we set binwidth as such.
```

```
library(ggplot2)
library(tidyverse)
ggplot(midwest, aes(x=poptotal))+
  geom_histogram(bins = sqrt(437)) #Better look without the binwidth set.
```



3) Write r code to produce a boxplot for the variable area, and then use your box plot to find Q1, Q2, and Q3

```
boxplot(midwest$area)
```



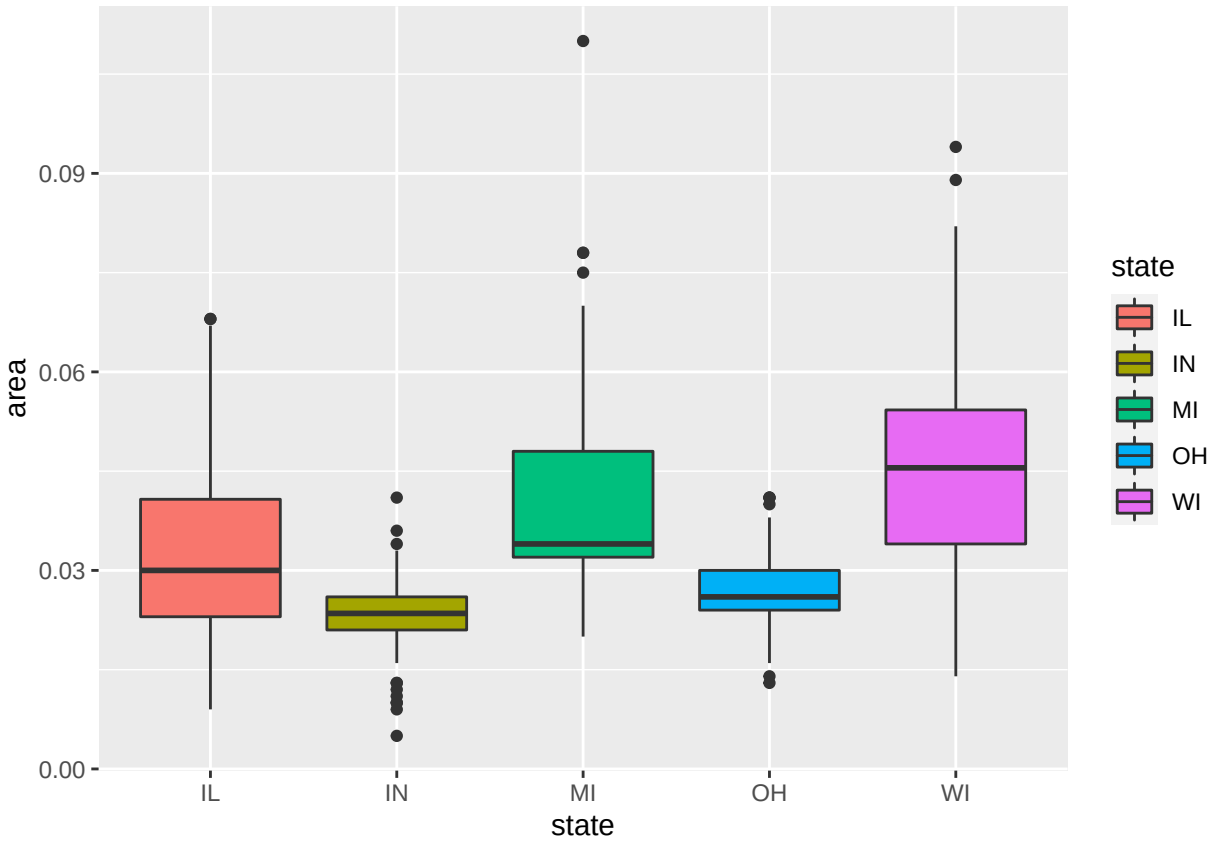

```
library(mosaic)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Attaching package: 'mosaic'
## The following object is masked from 'package:Matrix':
##
##     mean
## The following objects are masked from 'package:psych':
##
##     logit, rescale
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
## The following object is masked from 'package:purrr':
##
##     cross
## The following object is masked from 'package:ggplot2':
##
##     stat
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
favstats(midwest$area)
##    min    Q1 median    Q3  max       mean         sd   n missing
##  0.005 0.024   0.03 0.038 0.11 0.03316934 0.01467878 437       0
```

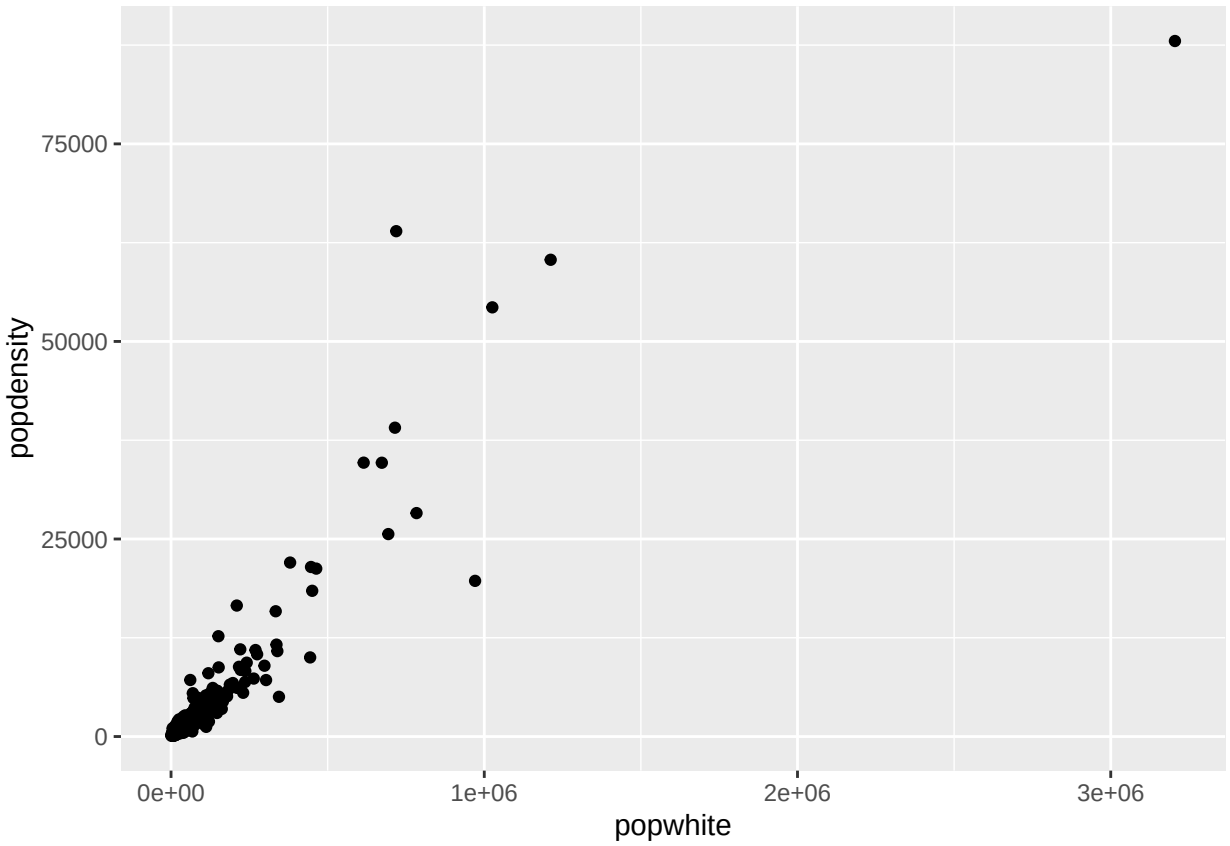
4) Write r code to produce side by side boxplots for the quantitative variable area with respect to the categorical variable state.

```
library(ggplot2)
midwest$state <-factor(midwest$state) # converts state to a categorical variable
my.area_state <<-ggplot(data=midwest, aes(y=area, x=state, fill=state ) ) # Creates boxplots
my.area_state <- my.area_state + geom_boxplot()
my.area_state
```



5) Write r code to produce a scatter plot for the variables popdensity and popwhite. Let popdensity be the independent variable x and popwhite be dependent variable y.

```
plot_scatter <- ggplot(midwest) + geom_point(aes(y = popdensity , x = popwhite))
plot_scatter
```

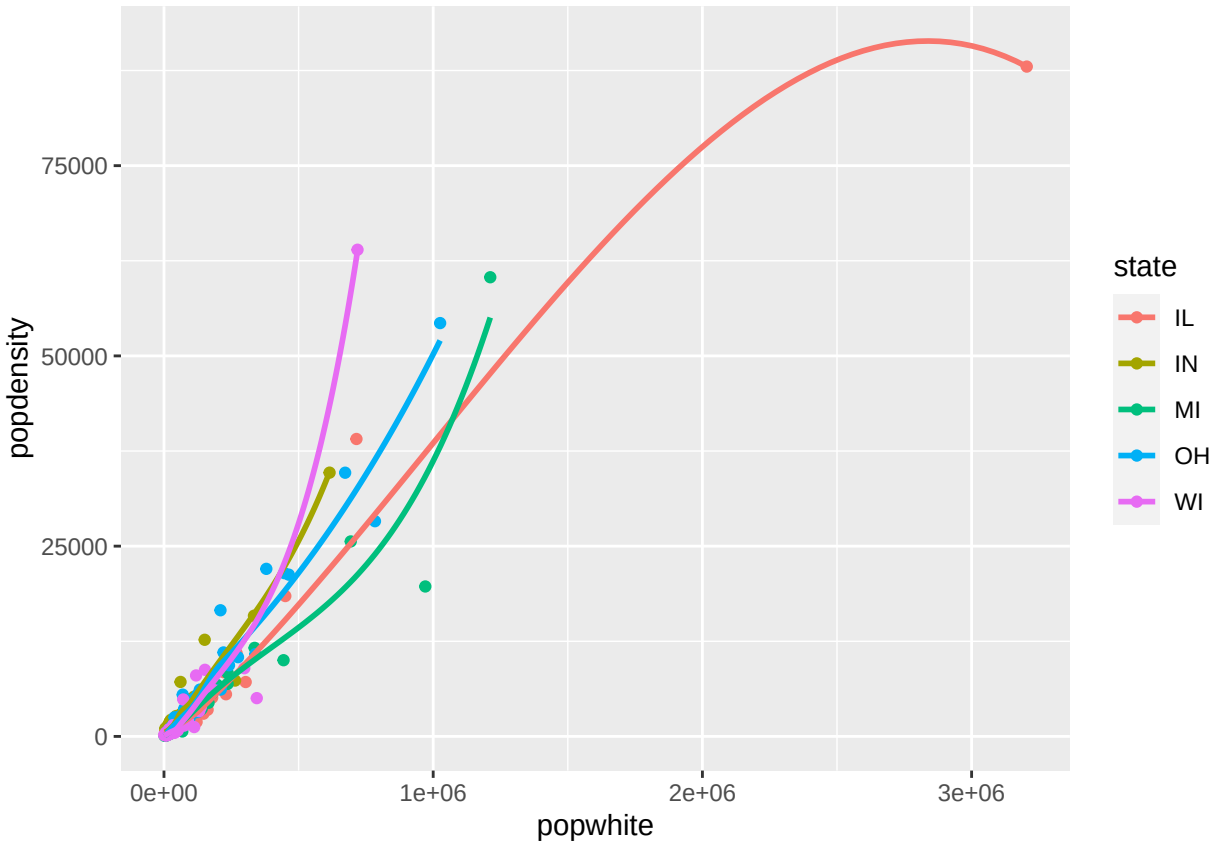


6) Write r code that will produce smooth lines plots and scatter plots on the same axis system for popwhite and popdensity with respect to the categorical variable state.

```
library(ggplot2)
plot_scatter_smooth <- ggplot(midwest,aes(y = popdensity , x = popwhite, color = state)) +
  geom_point() +
  geom_smooth(se = FALSE) #geom_smooth left to default which set ` using method = 'loess' and formula '

plot_scatter_smooth
```

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

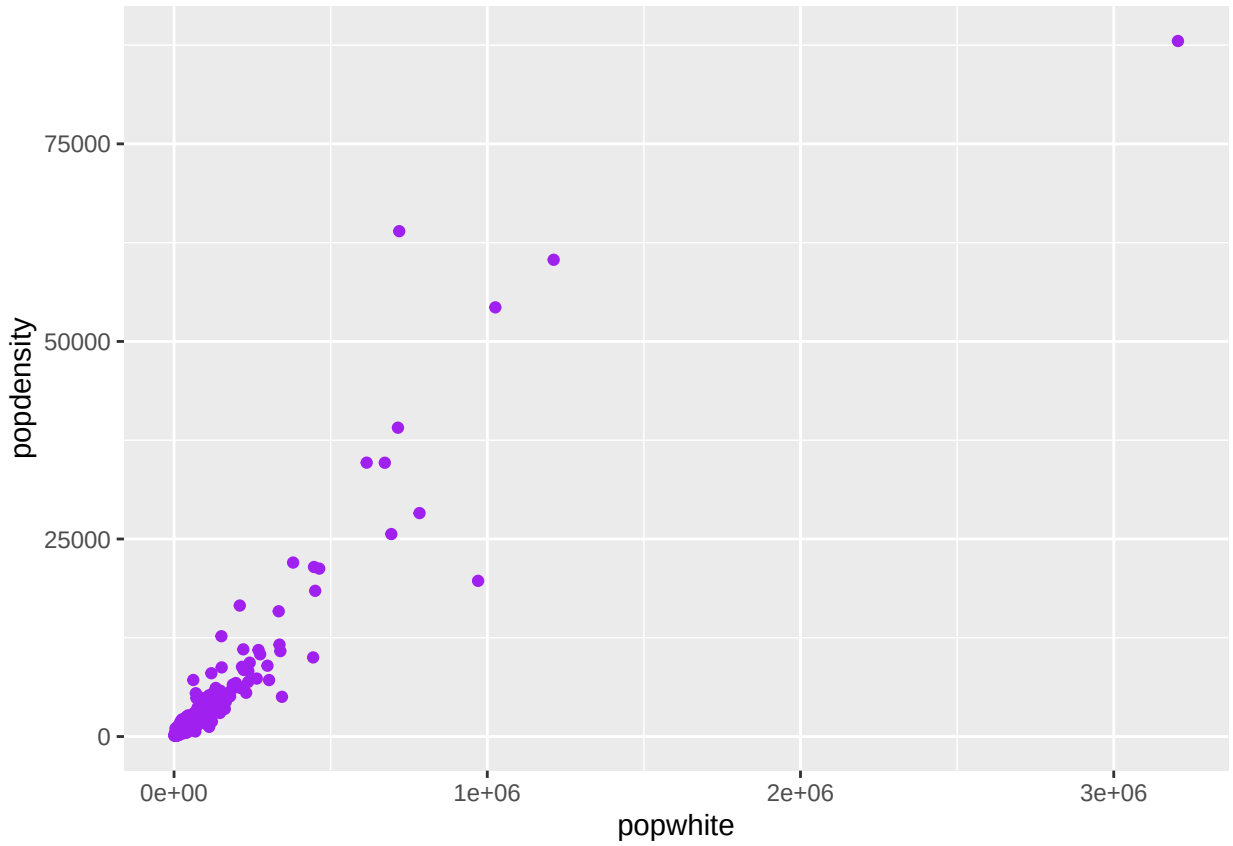



```
#Note: we are not asked to do lm here; just smoothing lines
```

7) Again, using the variables popdensity and popwhite, write r code that will produce the same basic scatter plot, but also make the following changes:

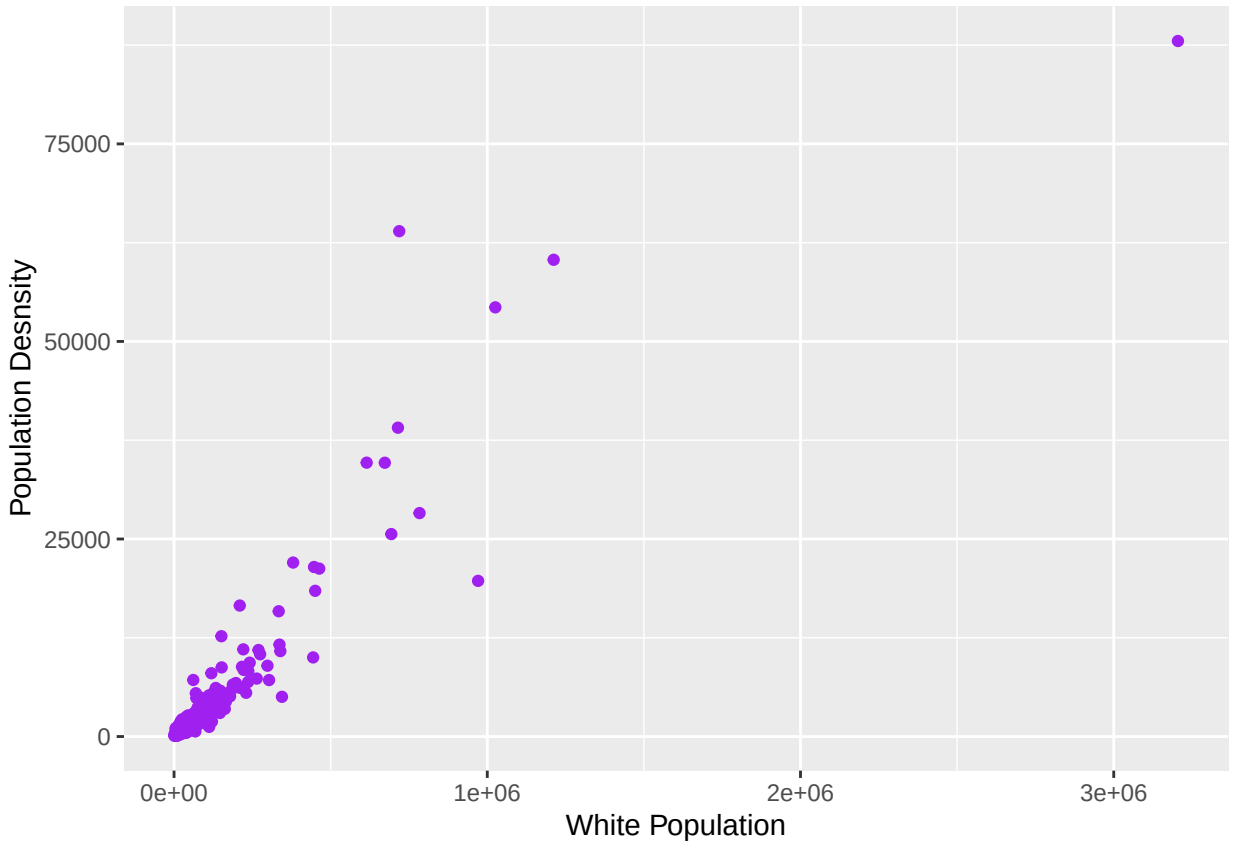
a) Your scatter plot should have purple data points.

```
plot_scatter <- ggplot(midwest, aes(y = popdensity , x = popwhite)) +
  geom_point(color= "purple")
plot_scatter
```



b) The label of the dependent variable should be changed to Population Density and the label of the independent variable should be changed to White Population

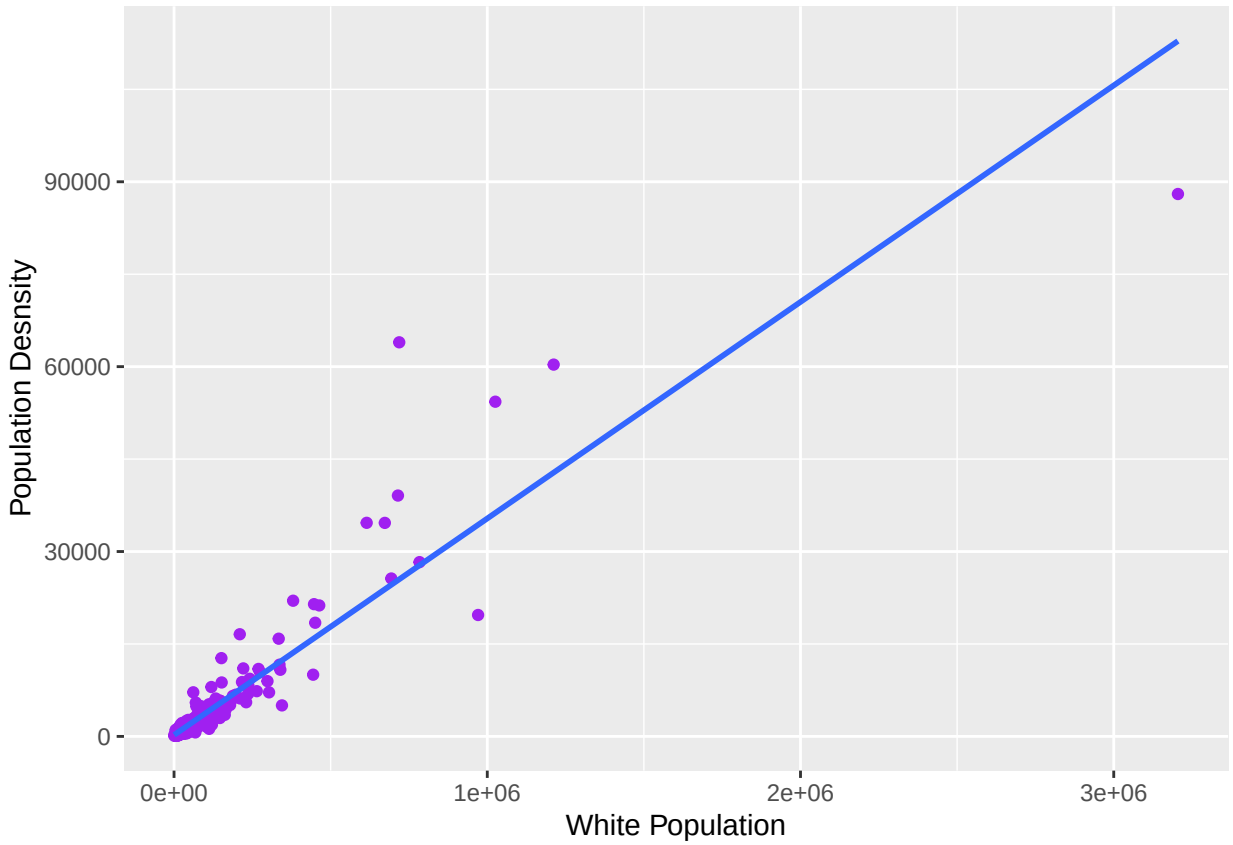
```
plot_scatter_level <- ggplot(midwest, aes(y = popdensity , x = popwhite)) +
  geom_point(color= "purple") +
  xlab("White Population") +
  ylab("Population Desnsity")
plot_scatter_level
```



c) Add a linear regression line to your graph.

```
plot_scatter_level_smoothing <- ggplot(midwest, aes(y = popdensity , x = popwhite)) +
  geom_point(color= "purple") +
  xlab("White Population") +
  ylab("Population Desnsity") +
  geom_smooth(se=FALSE, method = lm)
plot_scatter_level_smoothing
```

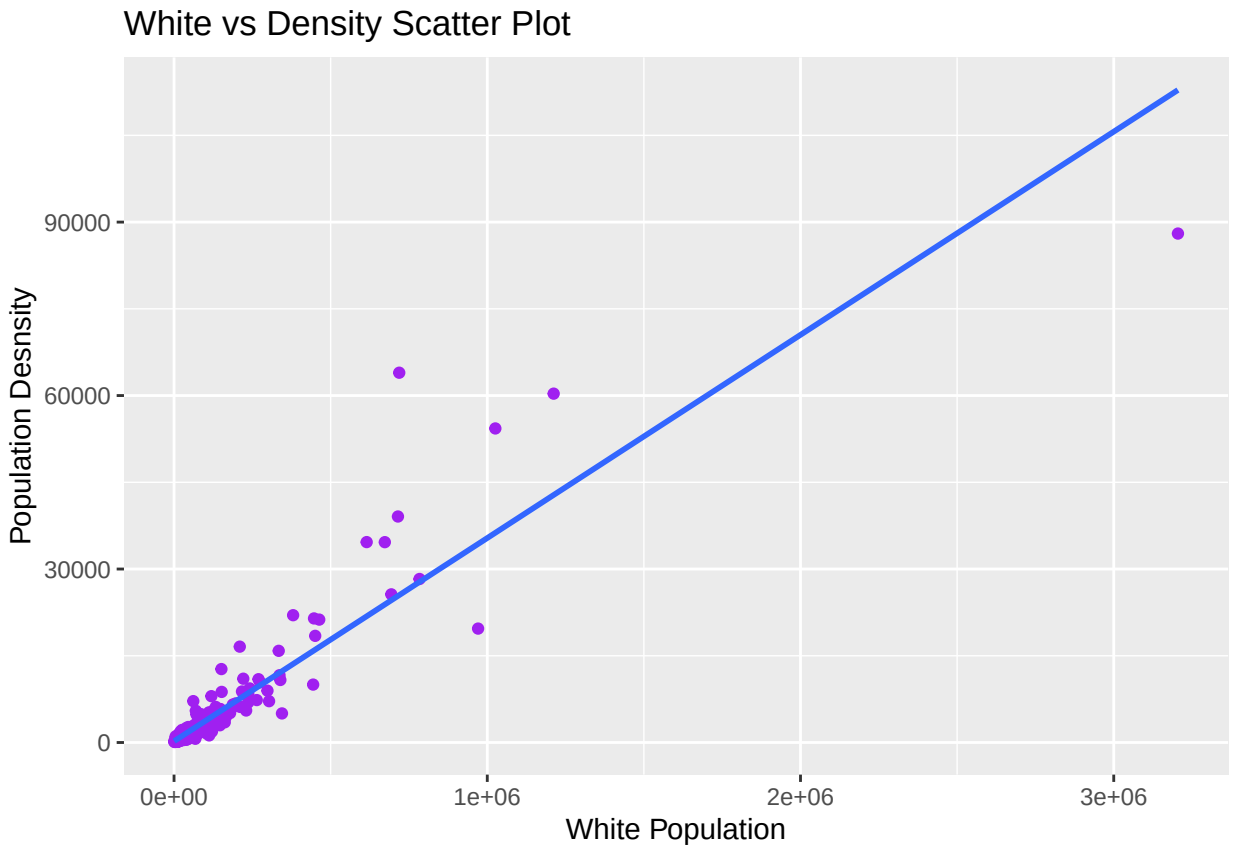
```
## `geom_smooth()` using formula 'y ~ x'
```



d) Add the following title to your graph; White vs Density Scatter Plot

```
plot_scatter_level_smoothing <- ggplot(midwest, aes(y = popdensity , x = popwhite)) +
  geom_point(color= "purple") +
  xlab("White Population") +
  ylab("Population Desnsity") +
  ggtitle("White vs Density Scatter Plot")+
  geom_smooth(se=FALSE, method = lm)
plot_scatter_level_smoothing
```

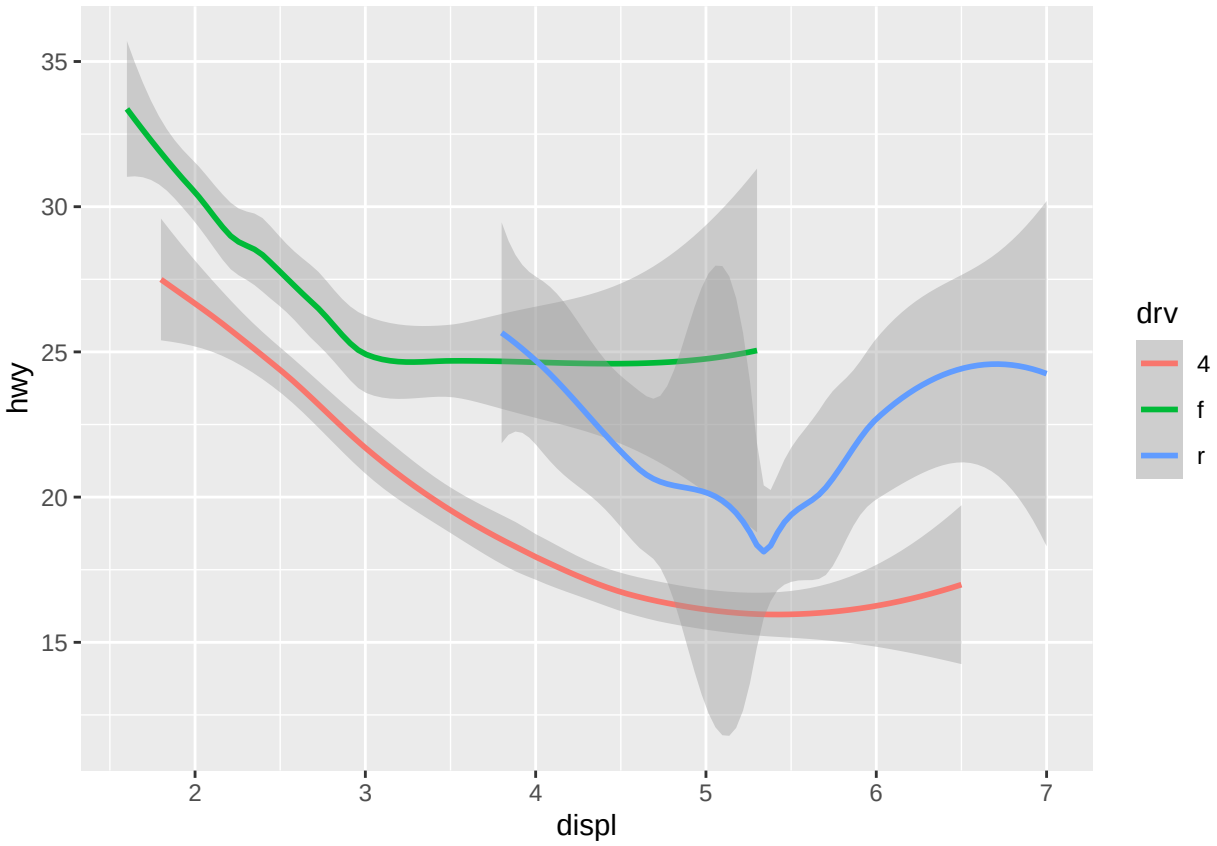
```
## `geom_smooth()` using formula 'y ~ x'
```



8) Write R code that will generate the following graph (use the mpg data frame)

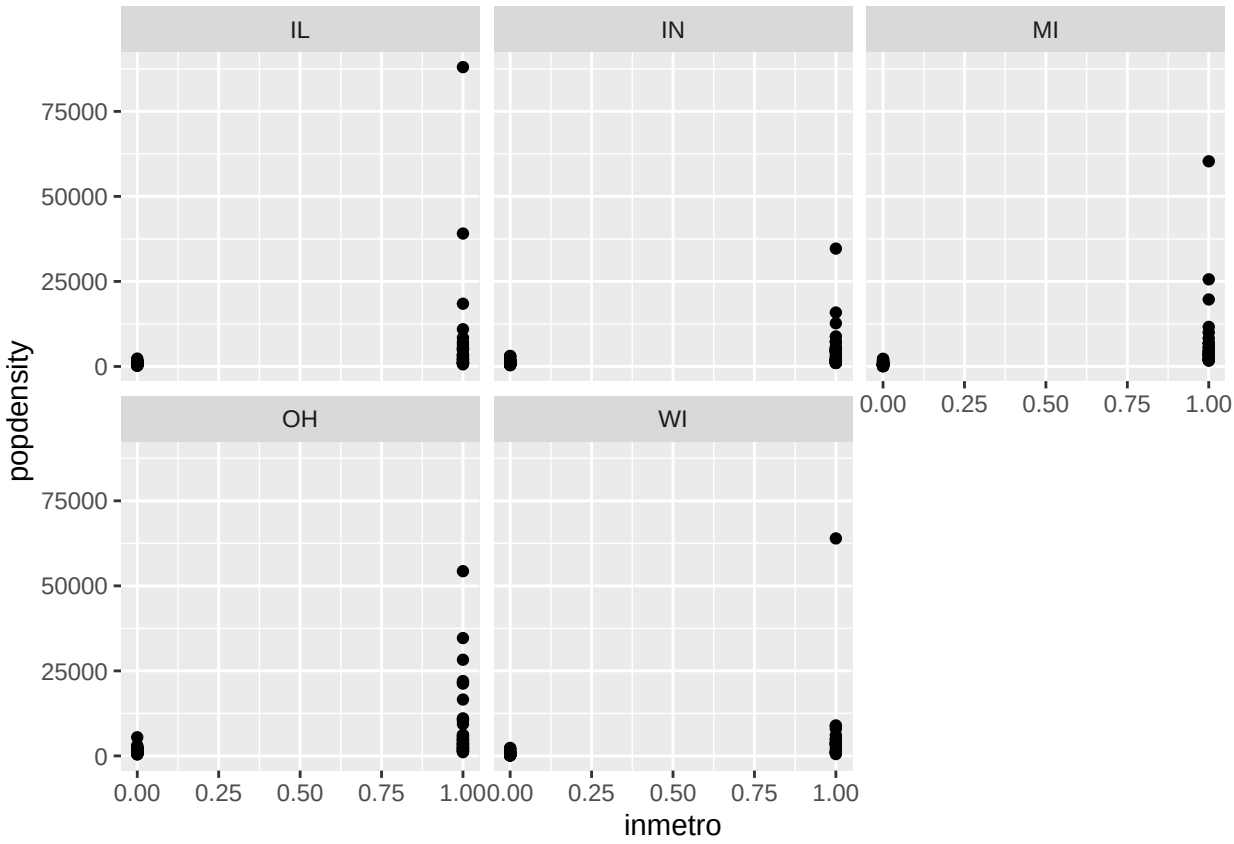
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = drv)) +
geom_smooth(se = TRUE)
```

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



9 Write R code that will produce the following facet plot using the midwest data frame

```
ggplot(data = midwest, mapping = aes(x = inmetro, y = popdensity)) +
  geom_point() +
  facet_wrap( ~ state)
```

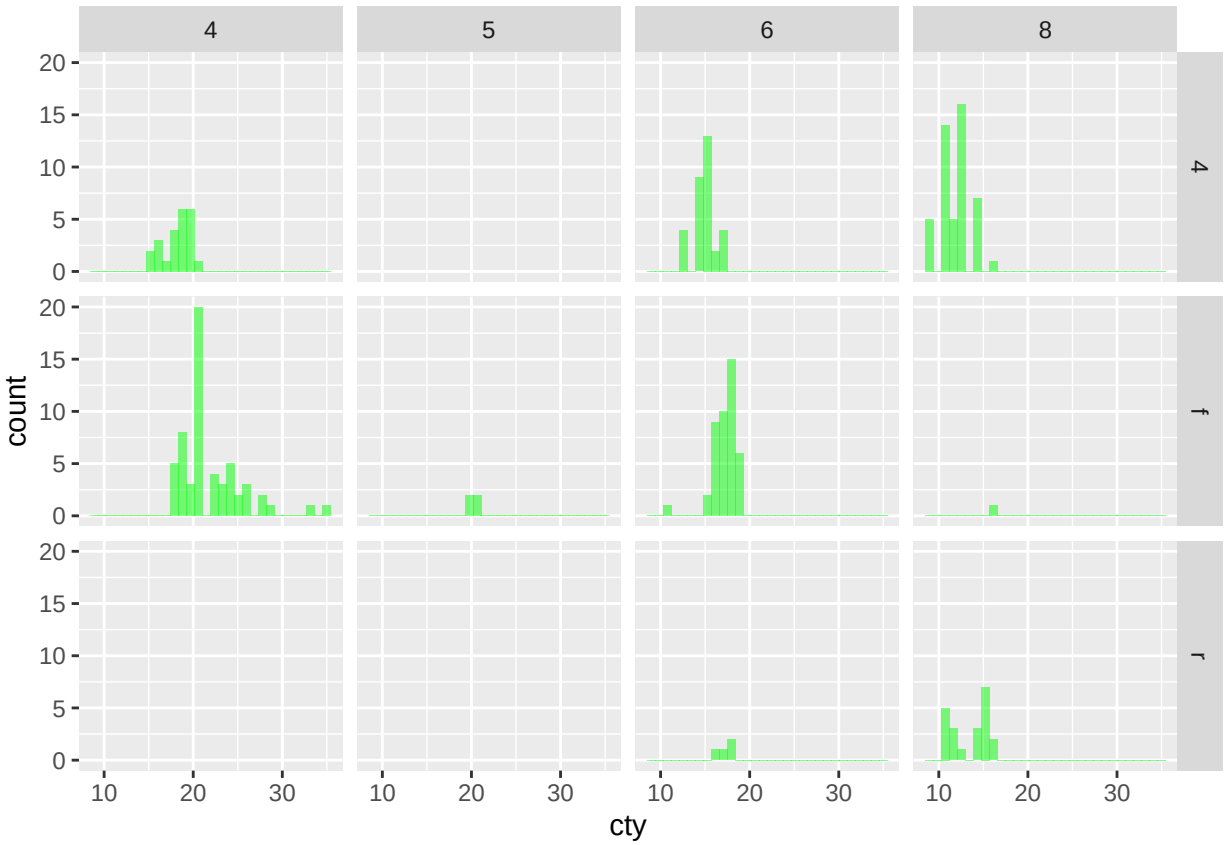


10 Write R code that will produce the following facet plot using the mpg data frame

ggplot(data = mpg, aes(x=cty)) + geom_histogram(position="identity", fill="green", alpha=1, bins = 30) + facet_wrap(drv ~cyl)

```
ggplot(data = mpg, aes(x=cty)) +
  geom_histogram(fill="green", alpha=0.5) +
  facet_grid(drv ~cyl)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Note: the Alpha =0.5 and the bin set to the default is an estimation because its hard to know the density of the color and size of the bin from your graph.