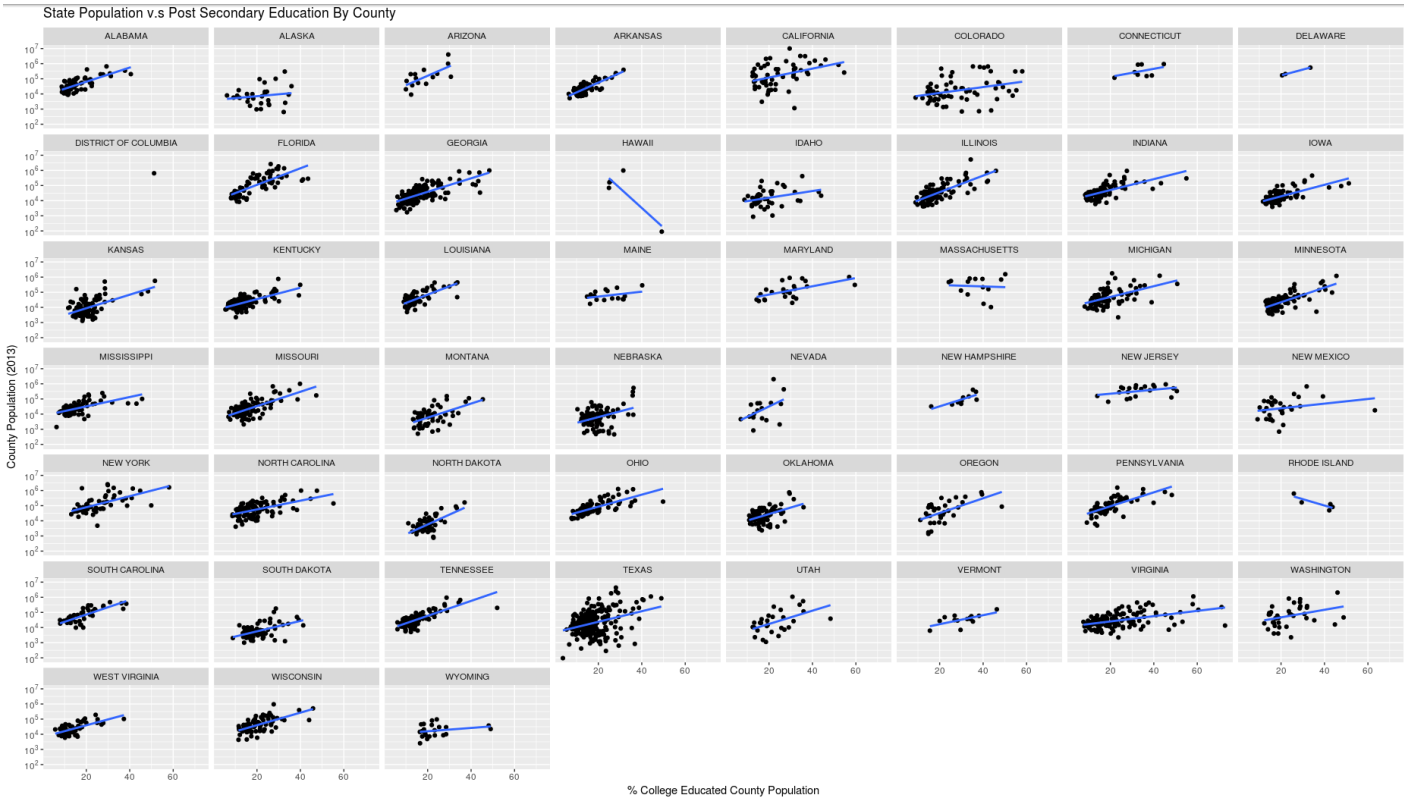
Home Work 6 | Class: Introduction To Data Science | Name: Charles Wallace

Plot Produced:



Code Used To Load Data, Clean Data, and Create Image:

- Loading And Cleaning The FIPS data.

```
1  ##############################################################################
2  ###                ### Cleaning The FIPS Data Set  ###                     ###
3  ###                ###################################                    ###
4  ###                                                                       ###
5  ##############################################################################
6
7  # Reading Lines Out of Fips The Max Width Of The Lines Is 43
8  FIPS <- readLines("FIPS_CountyName.txt")
9  # Figure Out The Max Width of Lines
10 maxWidth <- max(nchar(FIPS))
11 # Load Data Into Data Frame
12 FIPS <- read.fwf("FIPS_CountyName.txt", width = c(5, (maxWidth - 5)), strip.white = TRUE)
13 # Assign Names
14 names(FIPS) <- c("fips", "CountyState")
15 # Split off County And States, Seperating By ","
16 FIPSCsplit <- strsplit(as.character(FIPS$CountyState), ',')
17 # Clean Up Leading White Spaces
18 county <- trimws(lapply(FIPSCsplit, function(x){x[1]}), "l")
19 state <- trimws(lapply(FIPSCsplit, function(x){x[2]}), "l")
20 # Create Clean FIPS Data Frame
21 FIPS <- data.frame(factor(FIPS$fips), county, state)
22 # Assign New Names To Clean FIPS Data Frame
23 names(FIPS) <- c("fips", "county", "state")
24
25
```

Home Work 6 | Class: Introduction To Data Science | Name: Charles Wallace

- *Loading Data Set and Joining with FIPS Data Frame*

```
26  ###############################################################################
27  ###                   ### Joining FIPS And DataSet ###                      ###
28  ###                   ################################                      ###
29  ###                                                                         ###
30  ###############################################################################
31
32  # Use The dplyr Library install.packages(dplyr) If Not Already Isntalled
33  library(dplyr)
34  # Read In The Data Set
35  DataSet <- read.delim("DataSet.txt", sep=",", header = TRUE)
36  # Create Names For Subset of DataSet Used For Modeling
37  modelData <- data.frame(factor(DataSet$fips), DataSet$PST045213, DataSet$EDU685212)
38  # Create Names For Subset of DataSet Used For Modeling
39  names(modelData) <- c("fips", "population", "education")
40  # Perform Left Join To Match fips Identifier And Associated County From FIPS With modelData
41  modelData <- left_join(modelData, FIPS, by="fips")
42  # Re-Assign The Names To Make Things Easier Later
43  names(modelData) <- c("fips", "population", "education", "county", "postal")
44
```

- *Aggregating Postal Code With State Name In Each Row Of Data Set. Removing State Data and Na's*

```
45  ###############################################################################
46  ###           ### Getting Proper Titles And Eliminating NA's ###             ###
47  ###           ################################################              ###
48  ###                                                                         ###
49  ###############################################################################
50
51  # Extract The Postal Codes The Model Data
52  postal <- unique(modelData[!is.na(modelData$postal),]$postal)
53  # Extract The State Names From The Model Data In The Same Order
54  state <- modelData[is.na(modelData$postal),]$county
55  # Create New Data Frame with Matching Postal Code and State Name In Each Row
56  namesAndAbbr <- data.frame(postal, state[2:52])
57  # Name The Columns
58  names(namesAndAbbr) <- c("postal", "state")
59  # Get Rid Of The NAs In The modelData Data Set
60  modelDataNoNA <- na.omit(modelData)
61  # Join modelDataNoNA With The Name Of Each State
62  dataModelFinal <- left_join(modelDataNoNA, namesAndAbbr, by="postal")
63
```

Home Work 6 | Class: Introduction To Data Science | Name: Charles Wallace

- Creating Final Plots of Each State With Points and Linear Models

```
64  ##############################################################################
65  ###                        ### Modeling Data ###                           ###
66  ###                        ####################                            ###
67  ###                                                                        ###
68  ##############################################################################
69
70
71  # Using Scale Library To Scale Y Axis
72  # Install Scales Library install.packages("Scale")
73  library(scales)
74  # Loading ggplot2
75  library(ggplot2)
76  #plotting points and using lm to create smooth line
77  plot <- ggplot(dataModelFinal, aes(education, population)) + geom_point() + geom_smooth(se=FALSE, method = "lm")
78  # Creating Breaks With log10 Scaling On Y Axis To Better Capture Low/High Populations In Counties For Linear Model
79  # Also Adding Respective Labels To Ticks
80  # Used Example From CookBook for R: http://www.cookbook-r.com/Graphs/Axes_(ggplot2)/
81  plot <- plot + scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),labels = trans_format("log10", math_format(10^.x)))
82  # Using facet_wrap To Create Subplot For Each State
83  plot <- plot + facet_wrap(~state)
84  # Creating Titles
85  plot <- plot + ggtitle("State Population v.s Post Secondary Education by County")
86  # Creating Labels
87  plot <- plot + ylab("County Population (2013)") + xlab("% College Educated County Population")
```