Full Data

Claudia Shrefler

6/6/2019

#Data

```
my.install("class")
my.install("caret")
my.install("ggplot2")
my.install("MASS")
my.install("klaR")
my.install("rpart")
my.install("rpart.plot")
my.install("ggmap")
my.install("rgdal")
```

##Food Environment Atlas Data

Source: https://www.ers.usda.gov/data-products/food-environment-atlas/data-access-and-documentation-downloads/

Loaded in each sheet of the excel formatted data set as separate .csv files.

```
fea.access <- read.csv("FEA_ACCESS.csv")
fea.stores <- read.csv("FEA_STORES.csv")
fea.restaurants <- read.csv("FEA_RESTAURANTS.csv")
fea.assistance <- read.csv("FEA_ASSISTANCE.csv")
fea.insecurity <- read.csv("FEA_INSECURITY.csv")
fea.prices.taxes <- read.csv("FEA_PRICES_TAXES.csv")
fea.local <- read.csv("FEA_LOCAL.csv")
fea.health <- read.csv("FEA_HEALTH.csv")
fea.socioeconomic <- read.csv("FEA_SOCIOECONOMIC.csv")
```

Subsetted the data sets to only PA variables.

```
fea.access.pa <- fea.access[fea.access$State == "PA",]
fea.stores.pa <- fea.stores[fea.stores$State == "PA",]
fea.restaurants.pa <- fea.restaurants[fea.restaurants$State == "PA",]
fea.assistance.pa <- fea.assistance[fea.assistance$State == "PA",]
fea.insecurity.pa <- fea.insecurity[fea.insecurity$State == "PA",]
fea.prices.taxes.pa <- fea.prices.taxes[fea.prices.taxes$State == "PA",]
fea.local.pa <- fea.local[fea.local$State == "PA",]
fea.health.pa <- fea.health[fea.health$State == "PA",]
fea.socioeconomic.pa <- fea.socioeconomic[fea.socioeconomic$State == "PA",]
```

Changed values of -9999 as described in the documentation to NA.

##American Commuity Survey Data

Source: https://www.census.gov/acs/www/data/data-tables-and-tools/

Read in data set. Changed values of (X) described in documentation to NA.

##Census Data

Source: https://www.census.gov/data.html

Read in data set. Changed values of (X) described in documentation to NA.

##Feeding America Data

Source: https://www.feedingamerica.org/research/map-the-meal-gap/by-county

Read in data set and subset it to just PA.

```
feedAmerica <- read.csv("2017 County.csv")
feedAmerica.pa <- feedAmerica[feedAmerica$State == "PA",]
```

Modify variables that included percent signs to get rid of percent using gsub() and put them in decimal form by dividing by 100. Modify columns to get rid of dollar signs and commas. Use as.numeric() to put vectors back into numeric form since gsub() returns character vectors.

```
feedAmerica.pa$X2017.Food.Insecurity.Rate <- (as.numeric(gsub("%", "", feedAmerica.pa$X2017.Food.Insecu
feedAmerica.pa$Low.Threshold.in.state <- (as.numeric(gsub("%", "", feedAmerica.pa$Low.Threshold.in.stat
feedAmerica.pa$High.Threshold.in.state <- (as.numeric(gsub("%", "", feedAmerica.pa$High.Threshold.in.st
feedAmerica.pa$X..FI.â...Low.Threshold <- (as.numeric(gsub("%", "", feedAmerica.pa$X..FI.â...Low.Thresh
feedAmerica.pa$X..FI.Btwn.Thresholds <- (as.numeric(gsub("%", "", feedAmerica.pa$X..FI.Btwn.Thresholds)
feedAmerica.pa$X..FI...High.Threshold <- (as.numeric(gsub("%", "", feedAmerica.pa$X..FI...High.Threshol
feedAmerica.pa$X2017.Child.food.insecurity.rate <- (as.numeric(gsub("%", "", feedAmerica.pa$X2017.Child
feedAmerica.pa$X..food.insecure.children.in.HH.w..HH.incomes.below.185.FPL.in.2017 <- (as.numeric(gsub(
feedAmerica.pa$X..food.insecure.children.in.HH.w..HH.incomes.above.185.FPL.in.2017 <- (as.numeric(gsub(

feedAmerica.pa$X..of.Food.Insecure.Persons.in.2017 <- as.numeric(gsub(",", "", feedAmerica.pa$X..of.Foo
feedAmerica.pa$X..of.Food.Insecure.Children.in.2017 <- as.numeric(gsub(",", "", feedAmerica.pa$X..of.Fo
feedAmerica.pa$X2017.Cost.Per.Meal <- as.numeric(gsub("\\$", "", feedAmerica.pa$X2017.Cost.Per.Meal))
feedAmerica.pa$X2017.Weighted.Annual.Food.Budget.Shortfall <- as.numeric(gsub("[\\$,]", "", feedAmerica
```

##PA Legislature County Profile Data

Source: https://www.rural.palegislature.us/county_profiles.cfm

Read in data set from PA legislature. Transpose the data set to select variables more easily and convert back into a data frame.

```
original <- read.csv("CountyProfile.csv")
countyProfile <- data.frame(t(original))
```

Pick education data by selecting correct column and eliminating first two rows (variable names and PA total), and sort into my own variables. Clean the data to get rid of percent signs and divide by 100 to put into decimal form.

```
noHSDiploma <- as.vector(countyProfile[-c(1,2),209])
HSDiploma <- as.vector(countyProfile[-c(1,2),210])
someCollege <- as.vector(countyProfile[-c(1,2),211])
assocDeg <- as.vector(countyProfile[-c(1,2), 212])
bachDeg <- as.vector(countyProfile[-c(1,2),213])

noHSDiploma <- (as.numeric(as.character(gsub("%", "", noHSDiploma))))/100
HSDiploma <- (as.numeric(as.character(gsub("%","", HSDiploma))))/100
someCollege <- (as.numeric(as.character(gsub("%", "", someCollege))))/100
assocDeg <- (as.numeric(as.character(gsub("%","", assocDeg))))/100
bachDeg <- (as.numeric(as.character(gsub("%", "", bachDeg))))/100
```

Pick population column and get rid of commas.

```
population <- as.vector(countyProfile[-c(1,2),10])
population <- (as.numeric(as.character(gsub(",", "", population))))
```

Get median household income data from 2016 and eliminate dollar signs and commas.

```
medHouseIncome <- as.vector(countyProfile[-c(1,2),143])
medHouseIncome <- (as.numeric(as.character(gsub("[\\$,]", "", medHouseIncome))))
```

Get average commuting time to work in minutes in 2016.

```
aveCommuteTime <- as.numeric(as.vector(countyProfile[-c(1,2),323]))
```

Get number of licensed drivers in 2016.

```
licensedDrivers <- as.vector(countyProfile[-c(1,2),436])
licensedDrivers <- (as.numeric(as.character(gsub(",", "", licensedDrivers))))
```

Get number of registered vehicles and number of vehicles per 1000 residents in 2017.

```
regVehicles <- as.vector(countyProfile[-c(1,2),441])
regVehicles <- (as.numeric(as.character(gsub(",", "", regVehicles))))

pth_regVehicles <- as.vector(countyProfile[-c(1,2),442])
pth_regVehicles <- (as.numeric(as.character(gsub(",", "", pth_regVehicles))))
```

Get number of farms, the percent of land in farms, and the total market value of agricultural products sold in 2012.

```
numFarms <- as.vector(countyProfile[-c(1,2),448])
numFarms <- (as.numeric(as.character(gsub(",", "", numFarms))))

landInFarms <- as.vector(countyProfile[-c(1,2),454])
landInFarms <- (as.numeric(as.character(gsub("%", "", landInFarms))))/100

agProdValue <- as.vector(countyProfile[-c(1,2),468])
agProdValue <- (as.numeric(as.character(gsub("[\\$,]", "", agProdValue))))
```

Get acres of preserved farm land in 2017.

```
presAcres <- as.vector(countyProfile[-c(1,2),479])
presAcres <- (as.numeric(as.character(gsub(",", "", presAcres))))
```

Make temporary data frame out of County Profile variables to add to full data set.

```
temp <- data.frame(noHSDiploma)
temp <- cbind(temp, HSDiploma)
temp <- cbind(temp, assocDeg)
temp <- cbind(temp, bachDeg)
```

```
temp <- cbind(temp, population)
temp <- cbind(temp, medHouseIncome)
temp <- cbind(temp, aveCommuteTime)
temp <- cbind(temp, licensedDrivers)
temp <- cbind(temp, regVehicles)
temp <- cbind(temp, pth_regVehicles)
temp <- cbind(temp, numFarms)
temp <- cbind(temp, landInFarms)
temp <- cbind(temp, agProdValue)
temp <- cbind(temp, presAcres)
```

##US Dept Labor Bureau and Labor Statistics Data

Source: https://www.bls.gov/data/#unemployment

Reading in unemployment rate data.

```
unemployRate <- read.csv("Unemployment Rate.csv")
unemployRate <- unemployRate[,2]
```

##Full Data Set

Make new factor variable that designates county urban-rural classification using NCHS classification scheme found in Flynt and Daepp Int J Health Geogr (2015) 14:25 DOI 10.1186/s12942-015-0017-5.

```
urban.rural.class <- factor(levels = c("Noncore", "Micropolitan", "Small Metro", "Medium Metro", "Large
for (i in 1:length(census$HD01)) {
    if (census$HD01[i] < 10000) {
        urban.rural.class[i] = "Noncore"
    } else if (census$HD01[i] >= 10000 & census$HD01[i] < 49999) {
        urban.rural.class[i] = "Micropolitan"
    } else if (census$HD01[i] >= 50000 & census$HD01[i] < 250000) {
        urban.rural.class[i] = "Small Metro"
    } else if (census$HD01[i] >= 250000 & census$HD01[i] < 999999) {
        urban.rural.class[i] = "Medium Metro"
    } else {
        urban.rural.class[i] = "Large Metro"
    }
}
```

Load in shape files for PA counties. Source: https://www.pasda.psu.edu/uci/DataSummary.aspx?dataset=24

```
library(rgdal)
```

```
## Loading required package: sp
```

```
## rgdal: version: 1.4-4, (SVN revision 833)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20
##  Path to GDAL shared files: C:/Users/Claudia/Documents/R/win-library/3.6/rgdal/gdal
##  GDAL binary built with GEOS: TRUE
##  Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]
##  Path to PROJ.4 shared files: C:/Users/Claudia/Documents/R/win-library/3.6/rgdal/proj
##  Linking to sp version: 1.3-1
```

```
PaCounty <- readOGR(dsn=path.expand("~/BU/Research/Full Data/drive-download-20190705T172729Z-001"), lay
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\Claudia\Documents\BU\Research\Full Data\drive-download-20190705T172729Z-001", laye
## with 67 features
## It has 19 fields
## Integer64 fields read as strings:  COUNTY_N_1
```

coord_fixed() scales the x and y axes so if we change the outer dimensions of the plot the aspect ratio remains unchanged.

```
county_map <- map_data(PaCounty)
```

```
## Warning in SpatialPolygons2map(database, namefield = namefield): database
## does not (uniquely) contain the field 'name'.
```

```
county_data <- PaCounty@data
county_map$region <- factor(county_map$region)
```

Combine data sets into one fullData set. We will make region into a factor for the purpose of inner joining fullData with the shape files to make a together data set. We also add an SES index variable using the sum of the scaled variables: percent of adults with less than a HS degree, percent of households headed by single females, percent of nonwhite county residents, and poverty rate. We then make the together data set by inner joining county_map and fullData by region.

```
fullData <- NULL
together <- NULL
fullData <- fea.access.pa[,1:3]

tempRegion <- vector()
for (i in 1:nrow(fullData)) {
  number <- match(tolower(fullData$County[i]), tolower(county_data$COUNTY_NAM))
  tempRegion[i] <- number
}
region <- factor(tempRegion, labels = 0:66)
fullData <- cbind(fullData, region)
fullData <- cbind(fullData, urban.rural.class)
fullData <- cbind(fullData, census[,8:14])
fullData <- cbind(fullData, acs[,222])
fullData <- cbind(fullData, fea.access.pa[,4:44])
fullData <- cbind(fullData, fea.assistance.pa[,c(4,5,6,10,11,12,26,27,28,29,36,37,38,39,40,41)])
fullData <- cbind(fullData, fea.health.pa[,c(4,5,6,7,9,10,11,12,13,14)])
fullData <- cbind(fullData, fea.local.pa[,4:50])
fullData <- cbind(fullData, fea.restaurants.pa[,4:15])
fullData <- cbind(fullData, fea.socioeconomic.pa[,4:18])
fullData <- cbind(fullData, fea.stores.pa[,4:39])
fullData <- cbind(fullData, feedAmerica.pa[,4:18])
fullData <- cbind(fullData, temp)
fullData <- cbind(fullData, unemployRate)
sesData <- fullData[,c(13,140,149,206)]
pct_nonWhite <- 100-sesData$PCT_NHWHITE10
```

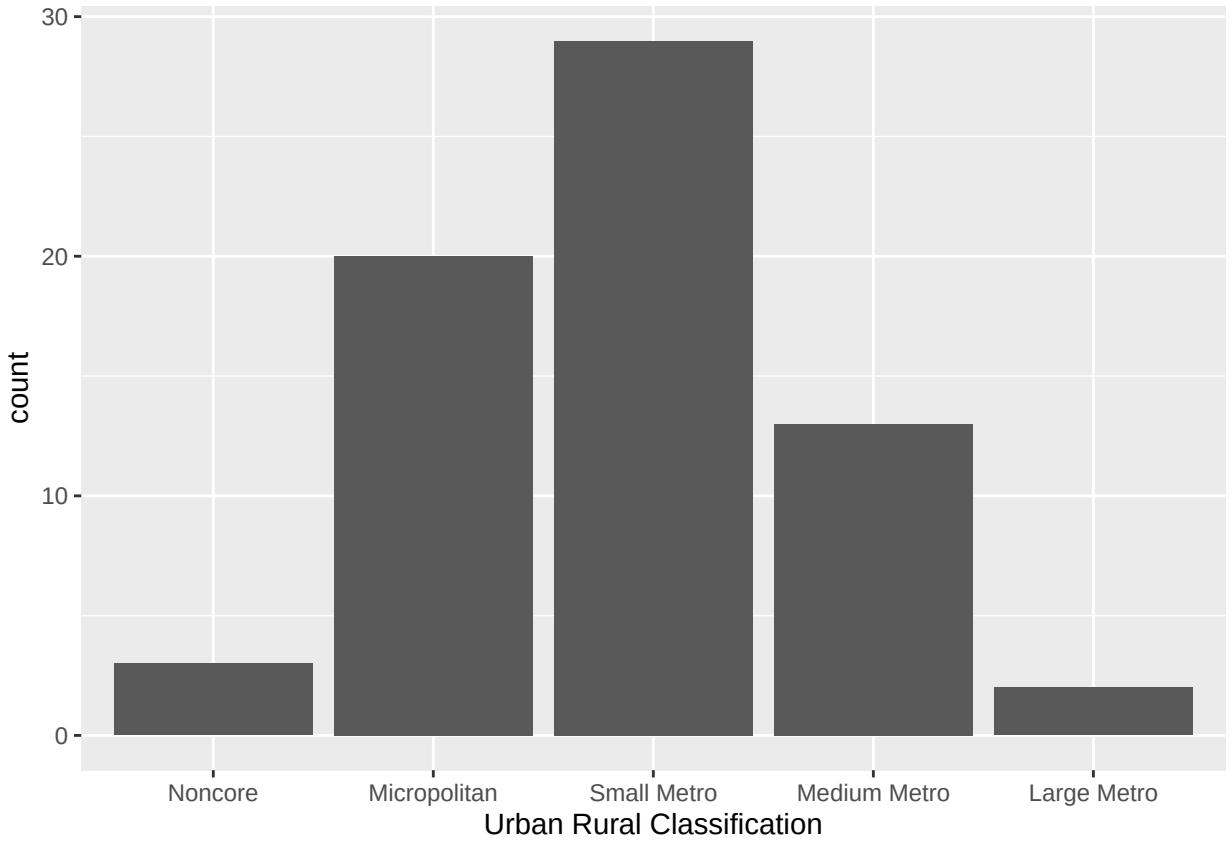
```r
sesData <- cbind(sesData,pct_nonWhite)
sesData <- sesData[,-2]
sesData <- scale(sesData)
SES <- vector()
for (i in 1:nrow(sesData)) {
  SES[i] <- sum(sesData[i,c(1,2,3,4)])
}
fullData <- cbind(fullData, SES)

together <- inner_join(county_map, fullData, by = "region")
```

```
## Warning: Column `region` joining factors with different levels, coercing to
## character vector
```

#Univariate EDA

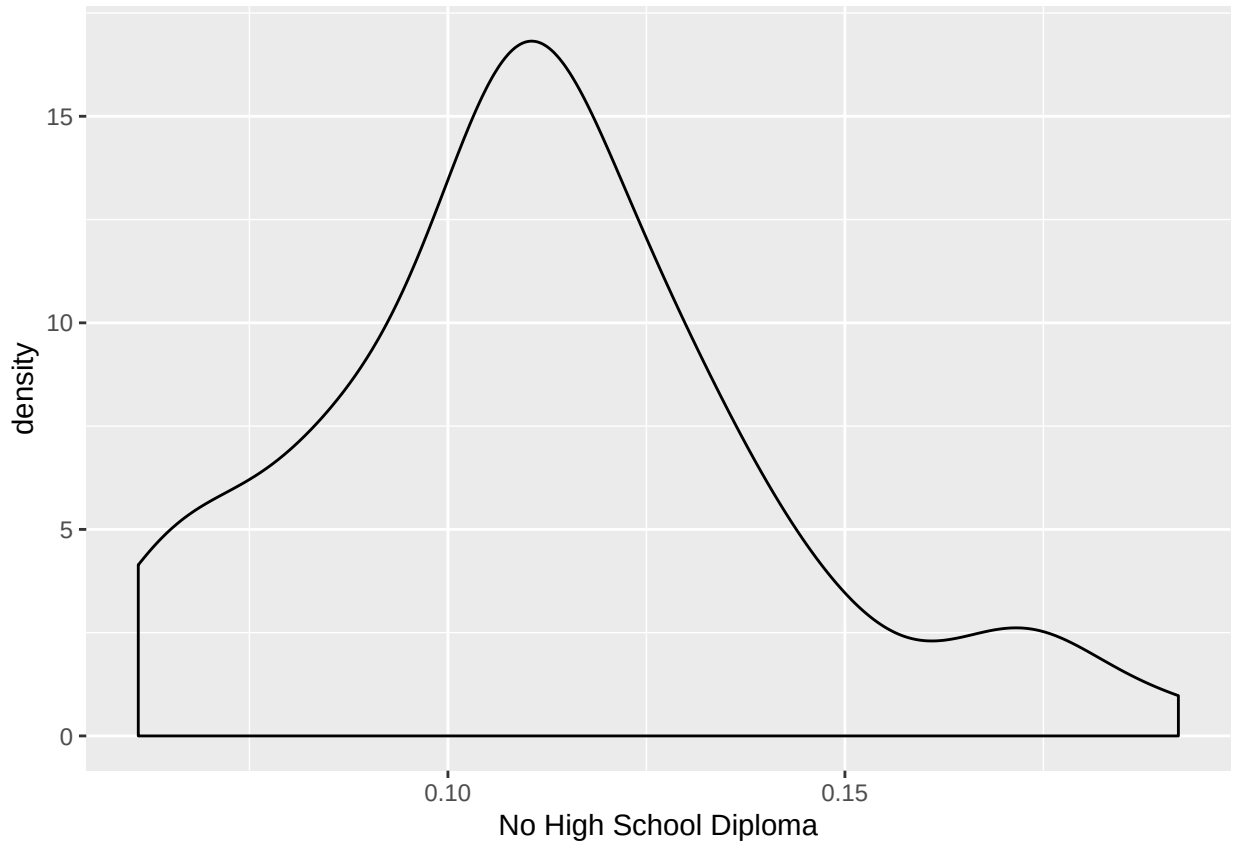
###Urban/Rural Classification Barchart showing the distribution of urban/rural counties in PA. We see that small metro contains the highest number of counties, while large metro contains the smallest number (Allegheny and Philadelphia), with noncore following as the next smallest.

```r
ggplot(fullData) + geom_bar(aes(x= urban.rural.class)) + xlab("Urban Rural Classification")
```

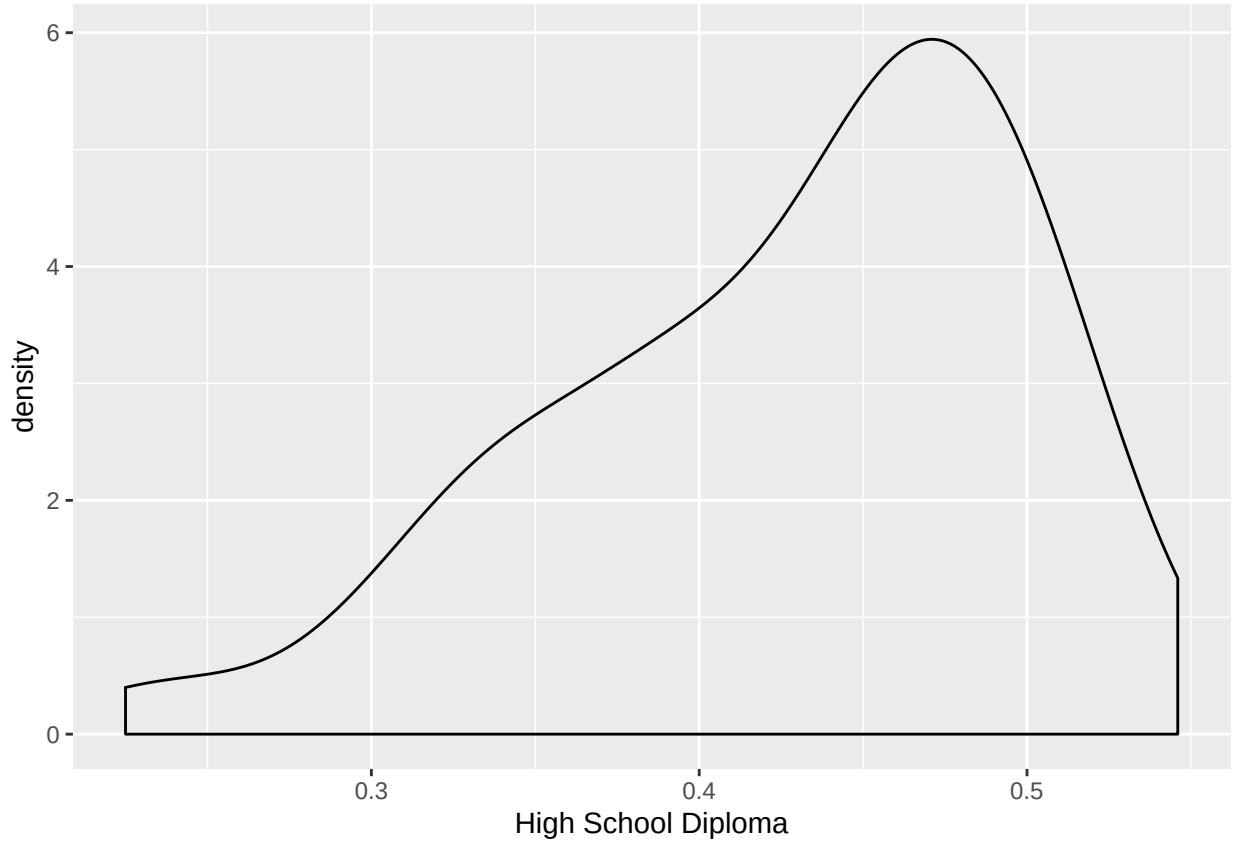


###Education Level Density plots showing education levels.

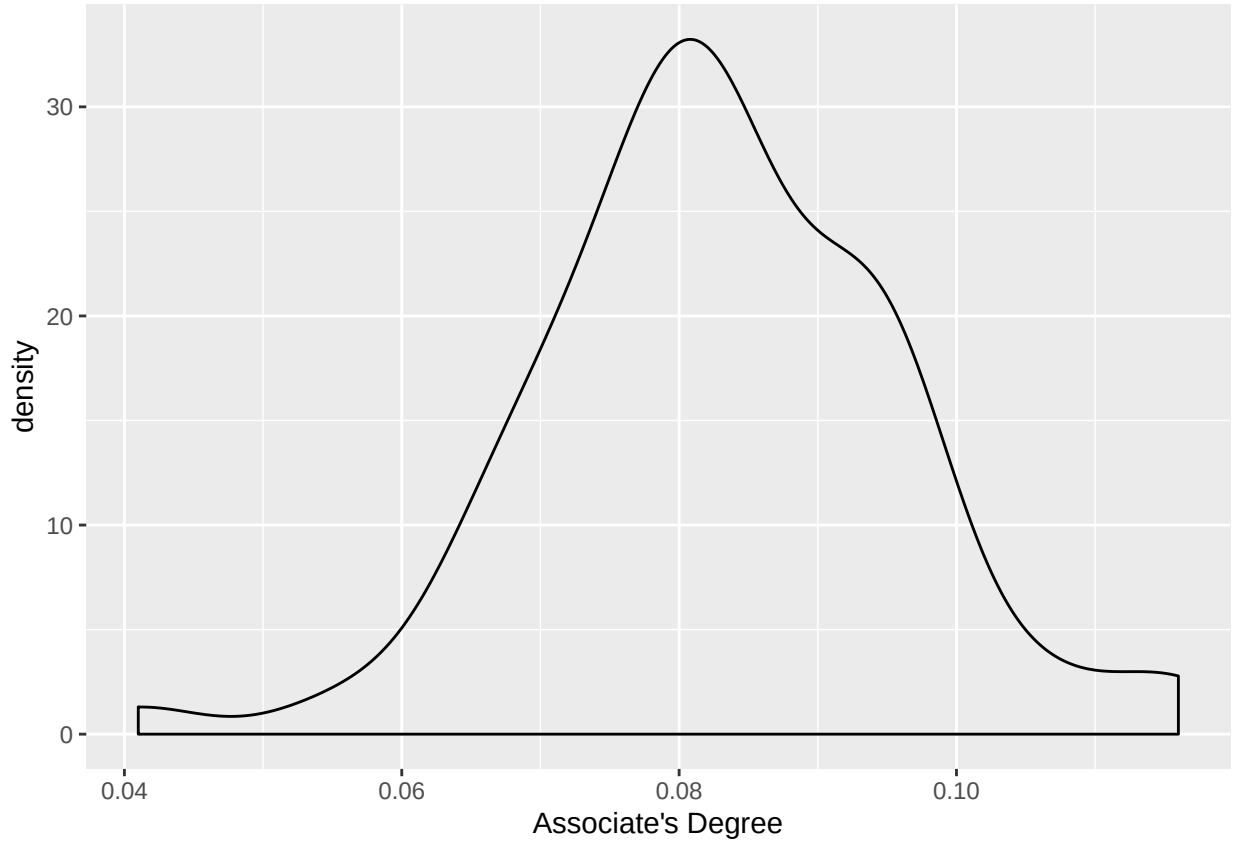
```
ggplot(fullData) + geom_density(aes(x=noHSDiploma)) + xlab("No High School Diploma")
```



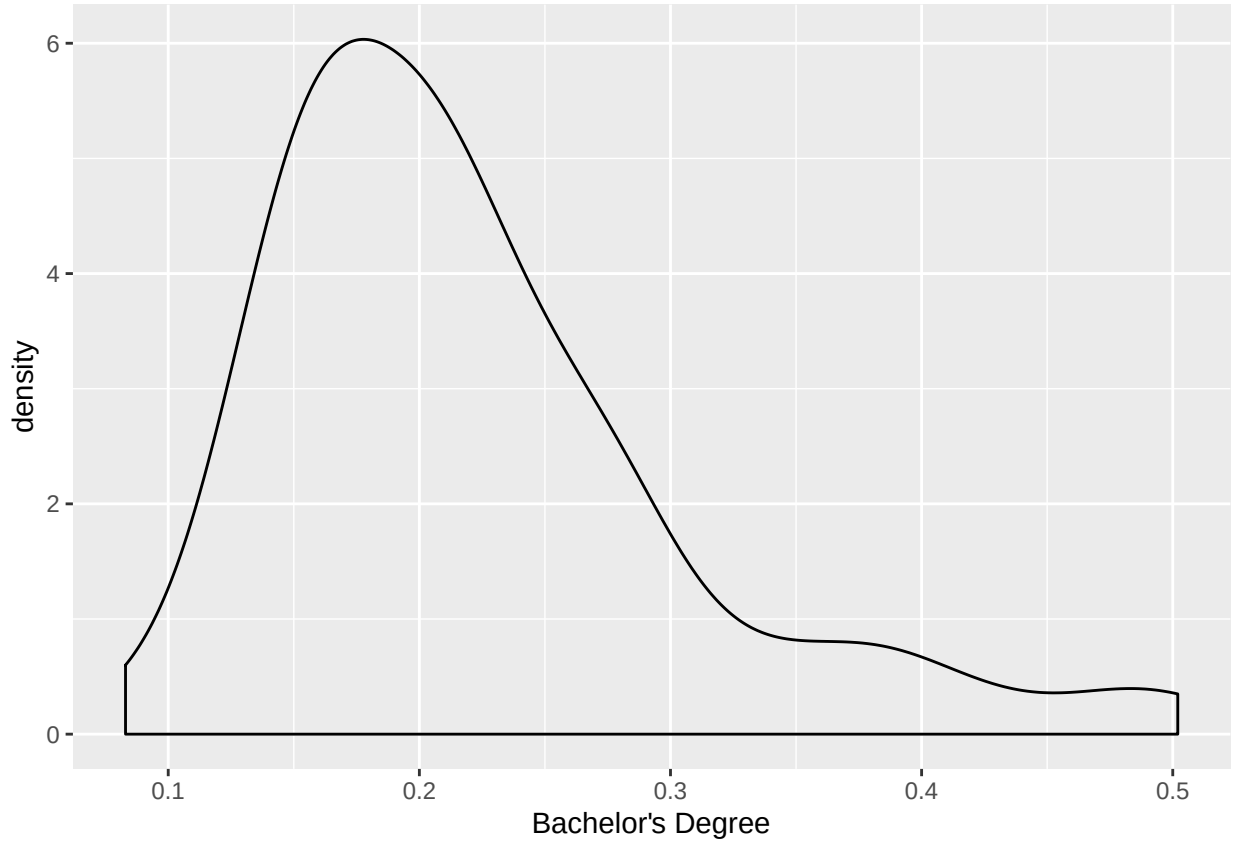

```
ggplot(fullData) + geom_density(aes(x=HSDiploma)) + xlab("High School Diploma")
```




```
ggplot(fullData) + geom_density(aes(x=assocDeg)) + xlab("Associate's Degree")
```

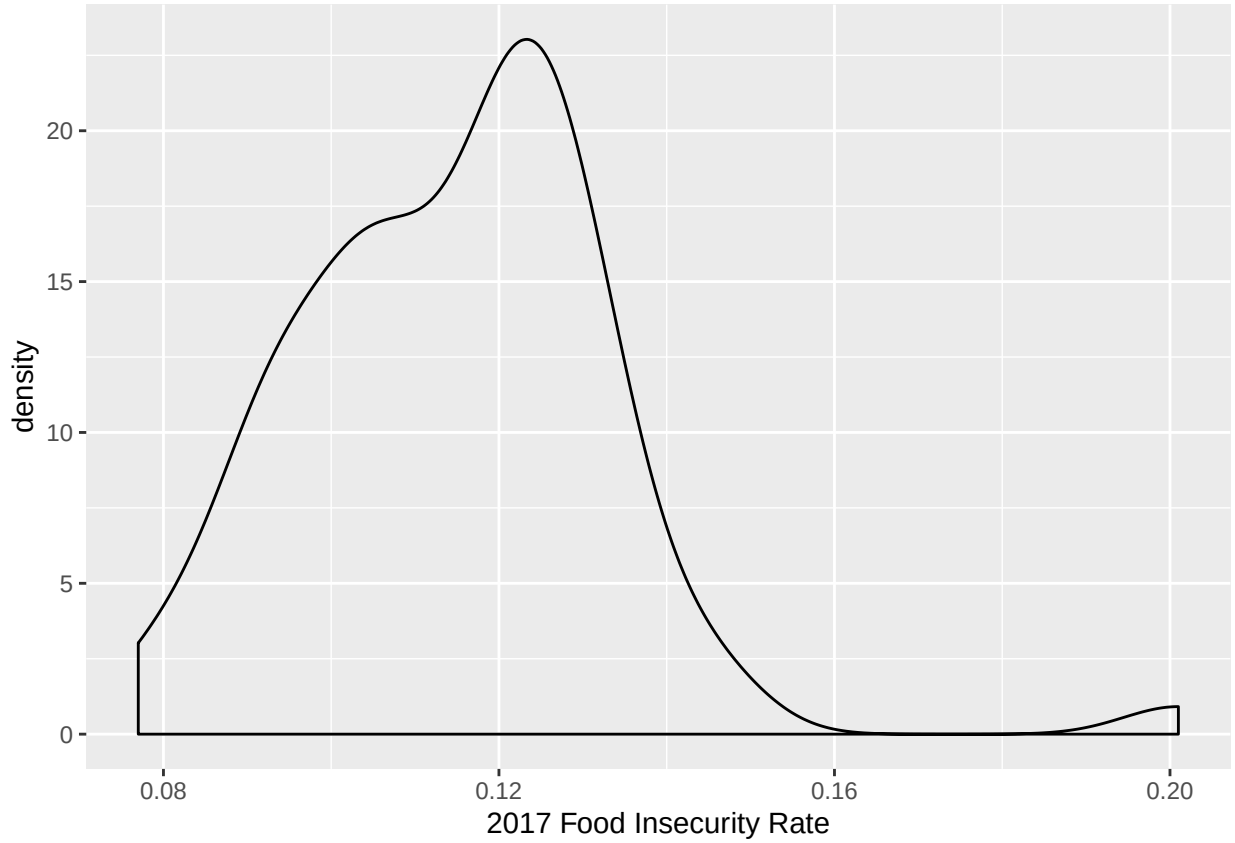



```
ggplot(fullData) + geom_density(aes(x=bachDeg)) + xlab("Bachelor's Degree")
```

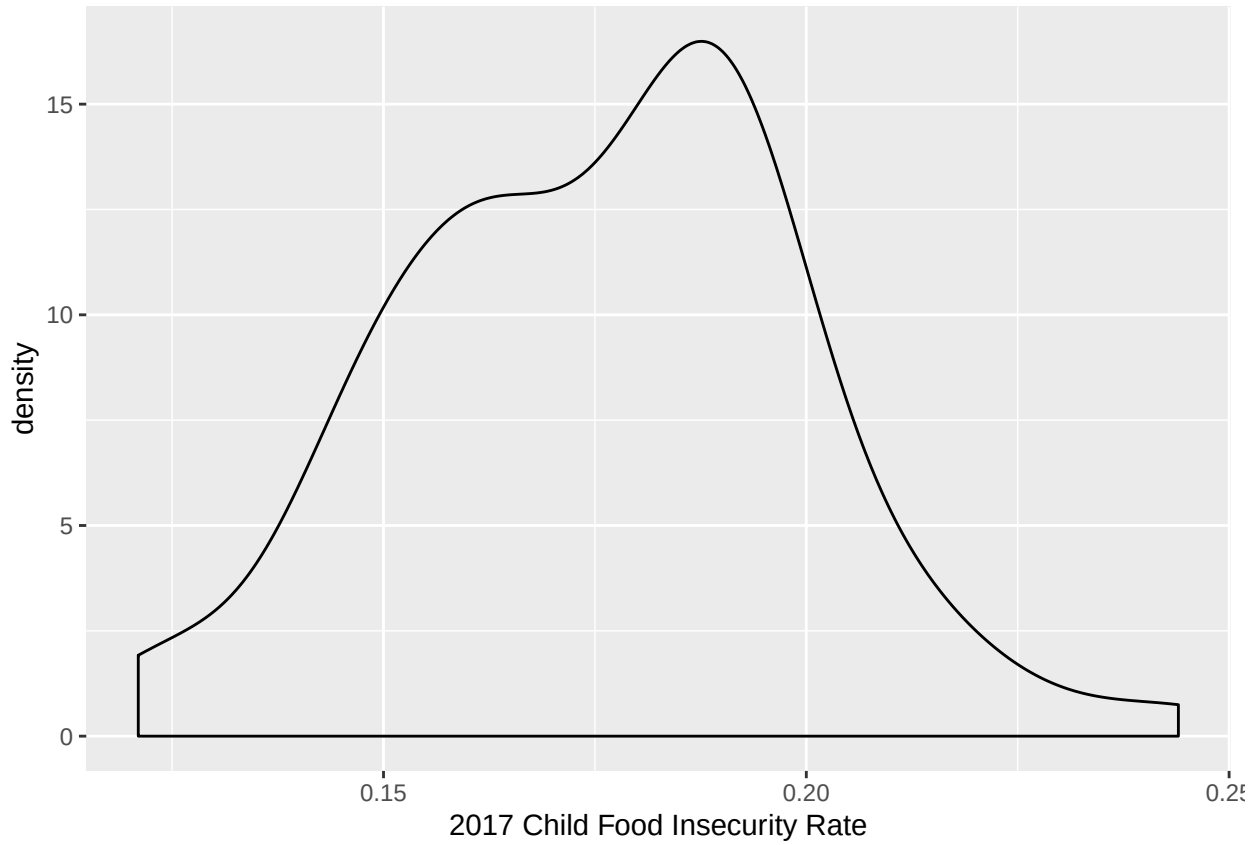


###Food Insecurity Rates Density plots examining variables associated with Food Insecurity rate.

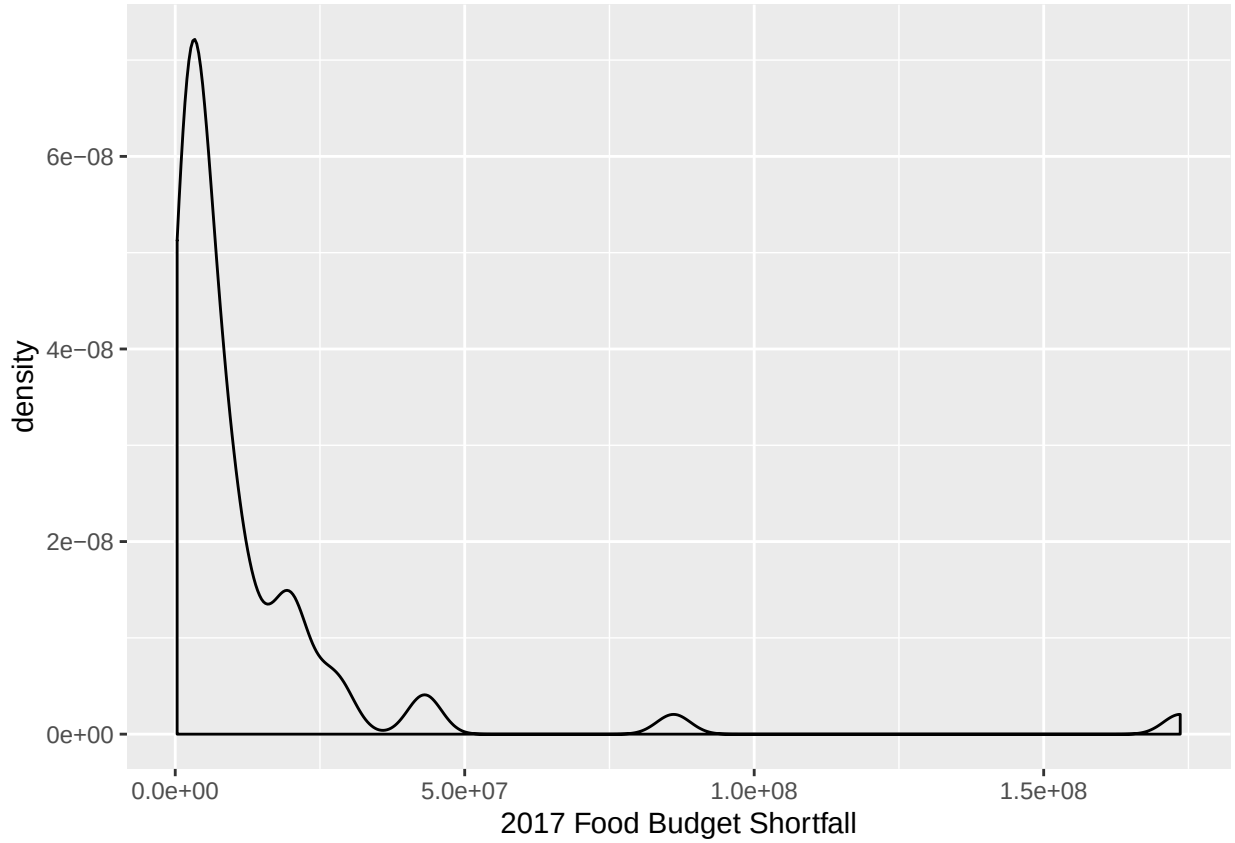
```
ggplot(fullData) + geom_density(aes(x=X2017.Food.Insecurity.Rate)) + xlab("2017 Food Insecurity Rate")
```



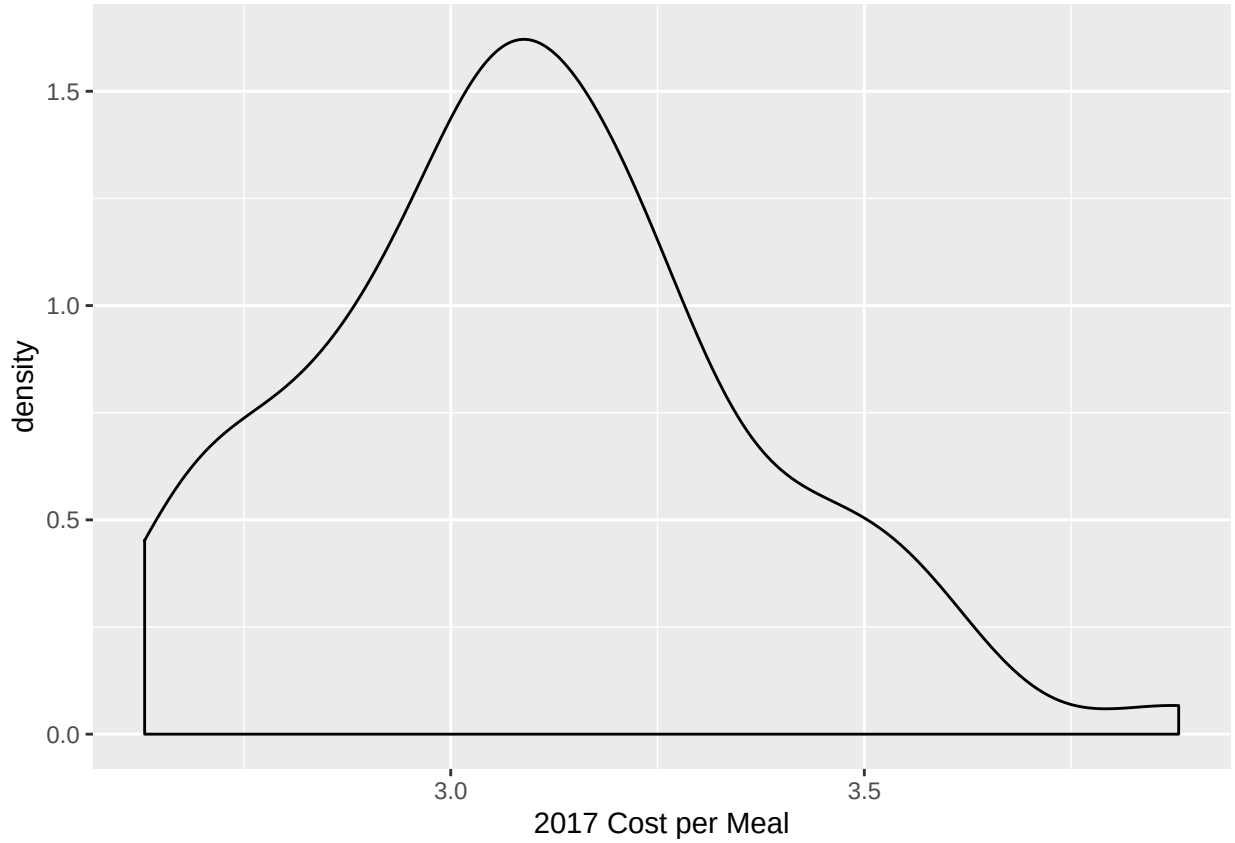

```
ggplot(fullData) + geom_density(aes(x=X2017.Child.food.insecurity.rate)) + xlab("2017 Child Food Insecu
```




```
ggplot(fullData) + geom_density(aes(x=X2017.Weighted.Annual.Food.Budget.Shortfall)) + xlab("2017 Food Bu
```

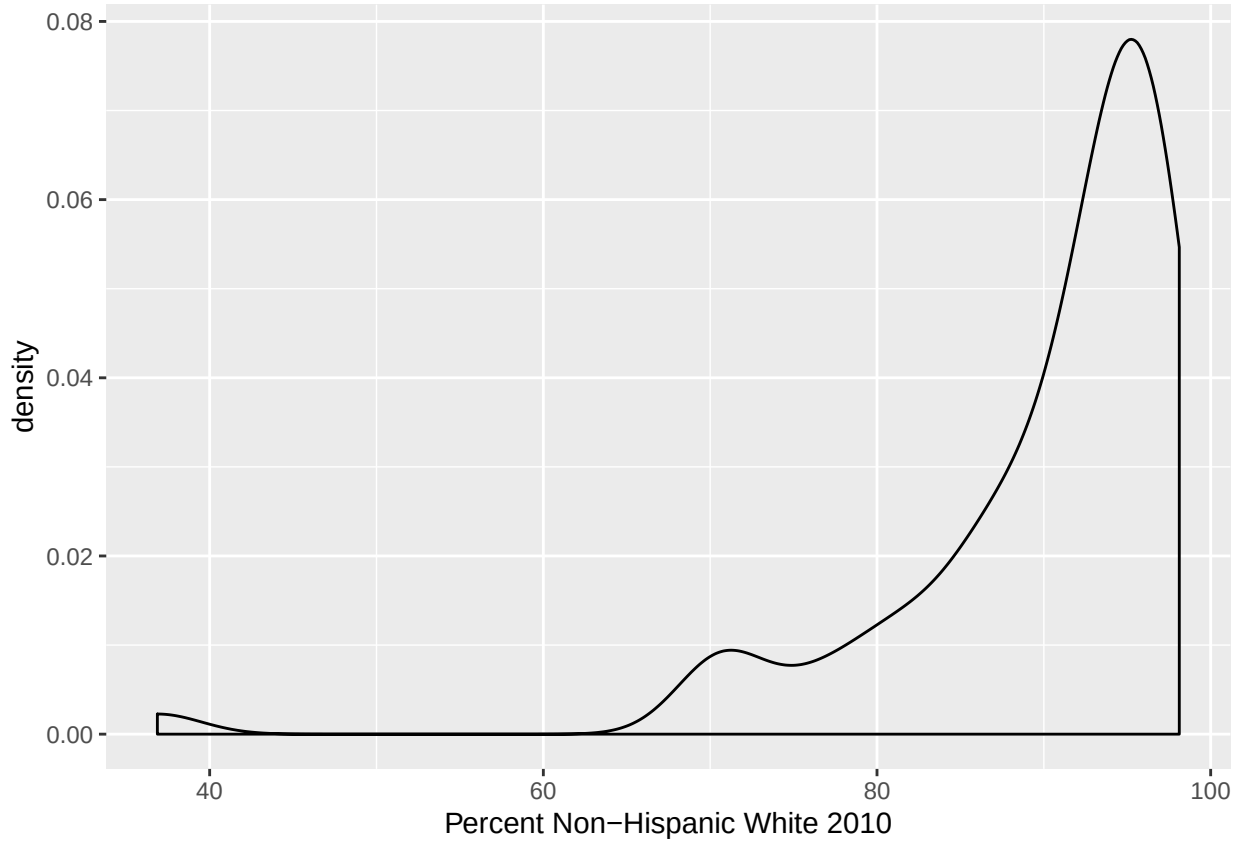



```
ggplot(fullData) + geom_density(aes(x=X2017.Cost.Per.Meal)) + xlab("2017 Cost per Meal")
```

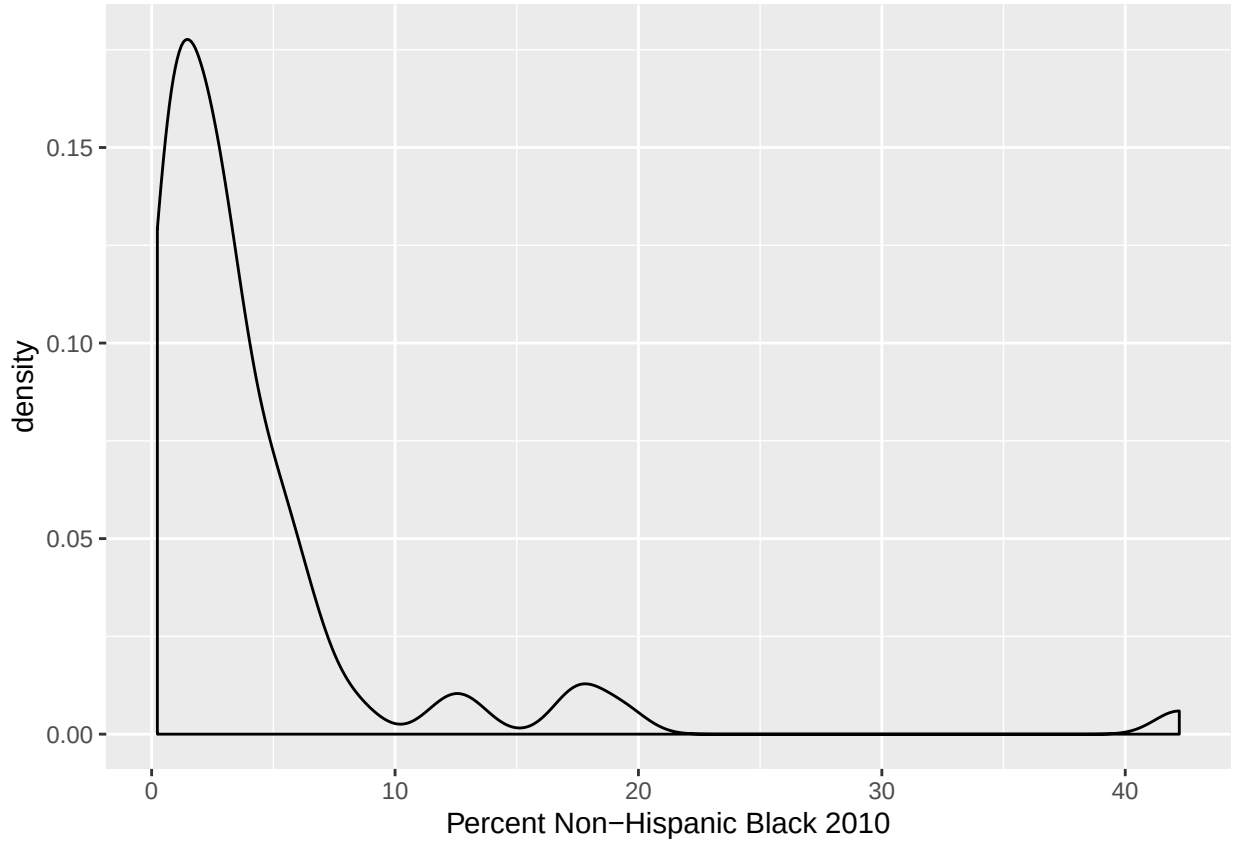


###Demographics Density plots demonstrating demographics.

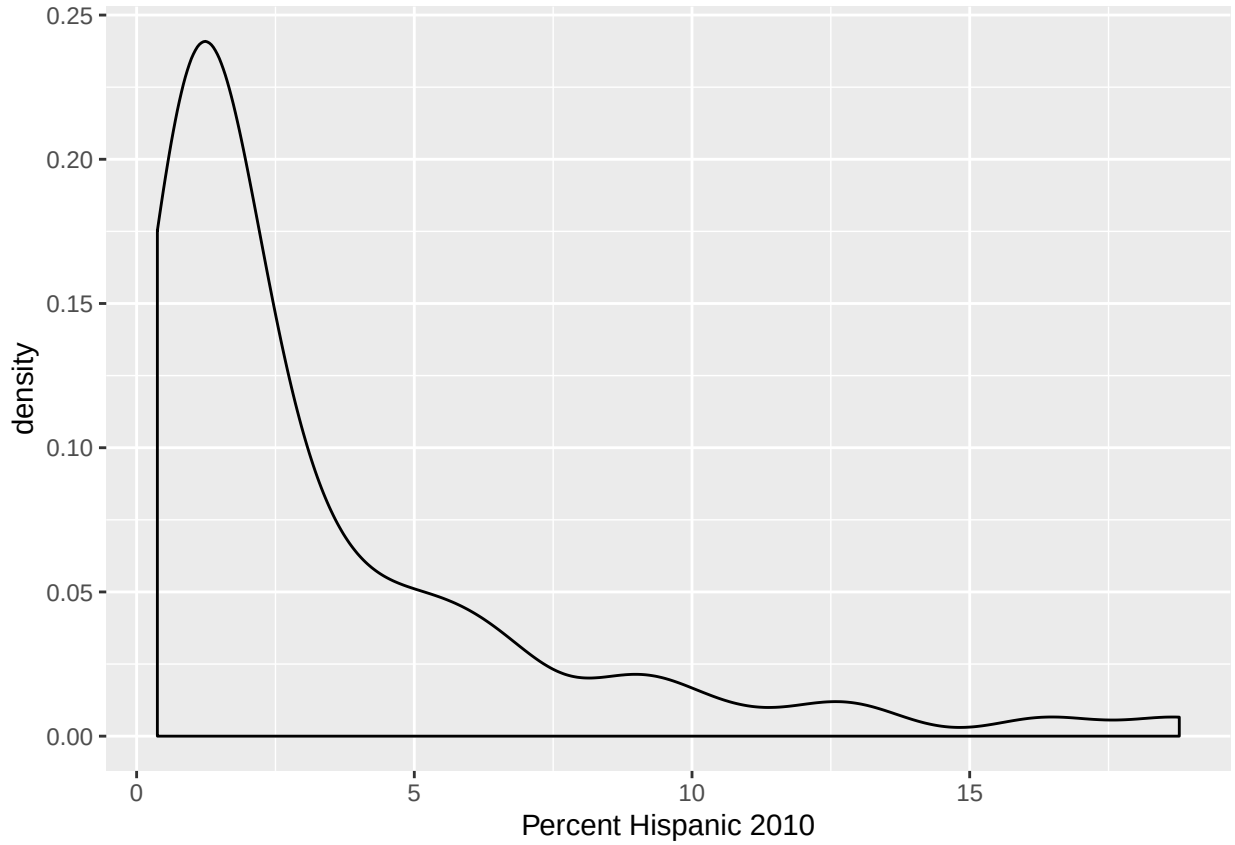
```
ggplot(fullData) + geom_density(aes(x=PCT_NHWHITE10)) + xlab("Percent Non-Hispanic White 2010")
```



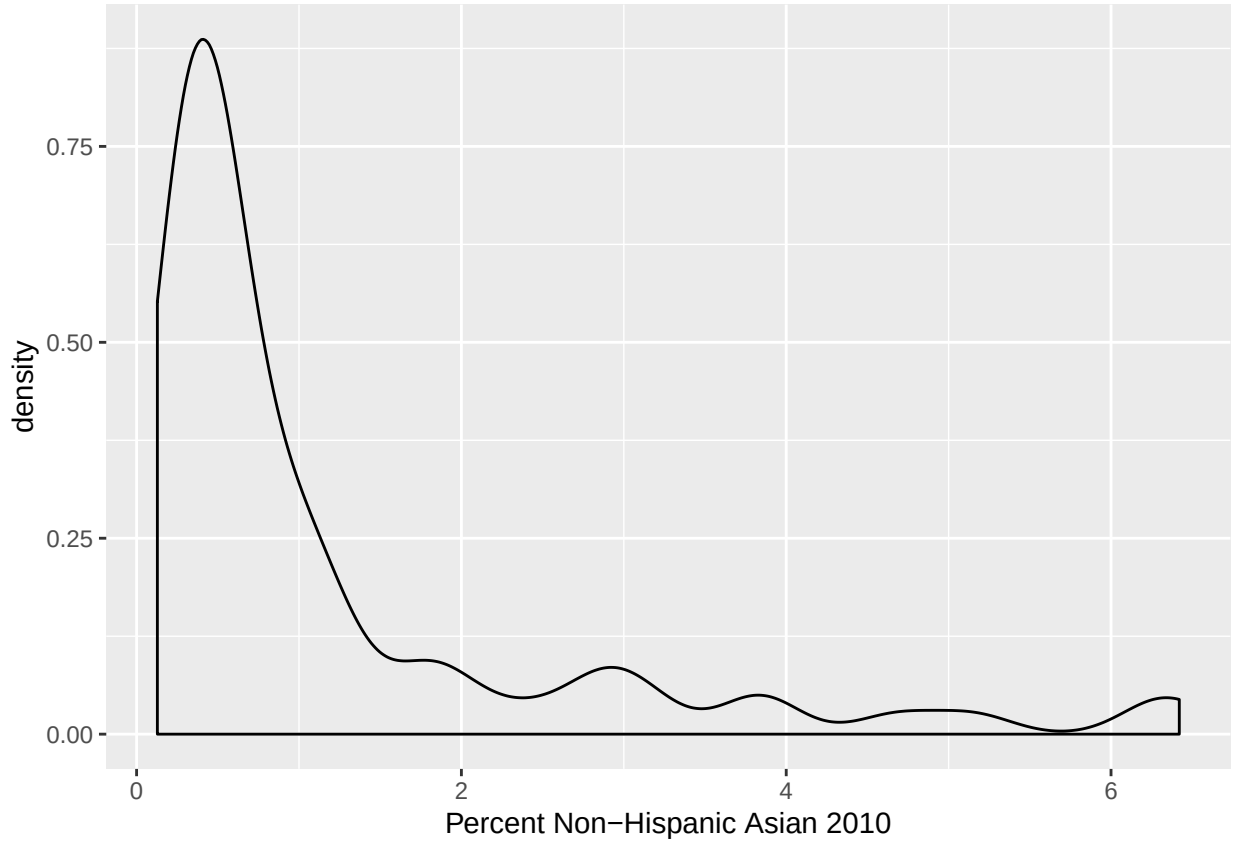

```
ggplot(fullData) + geom_density(aes(x=PCT_NHBLACK10)) + xlab("Percent Non-Hispanic Black 2010")
```



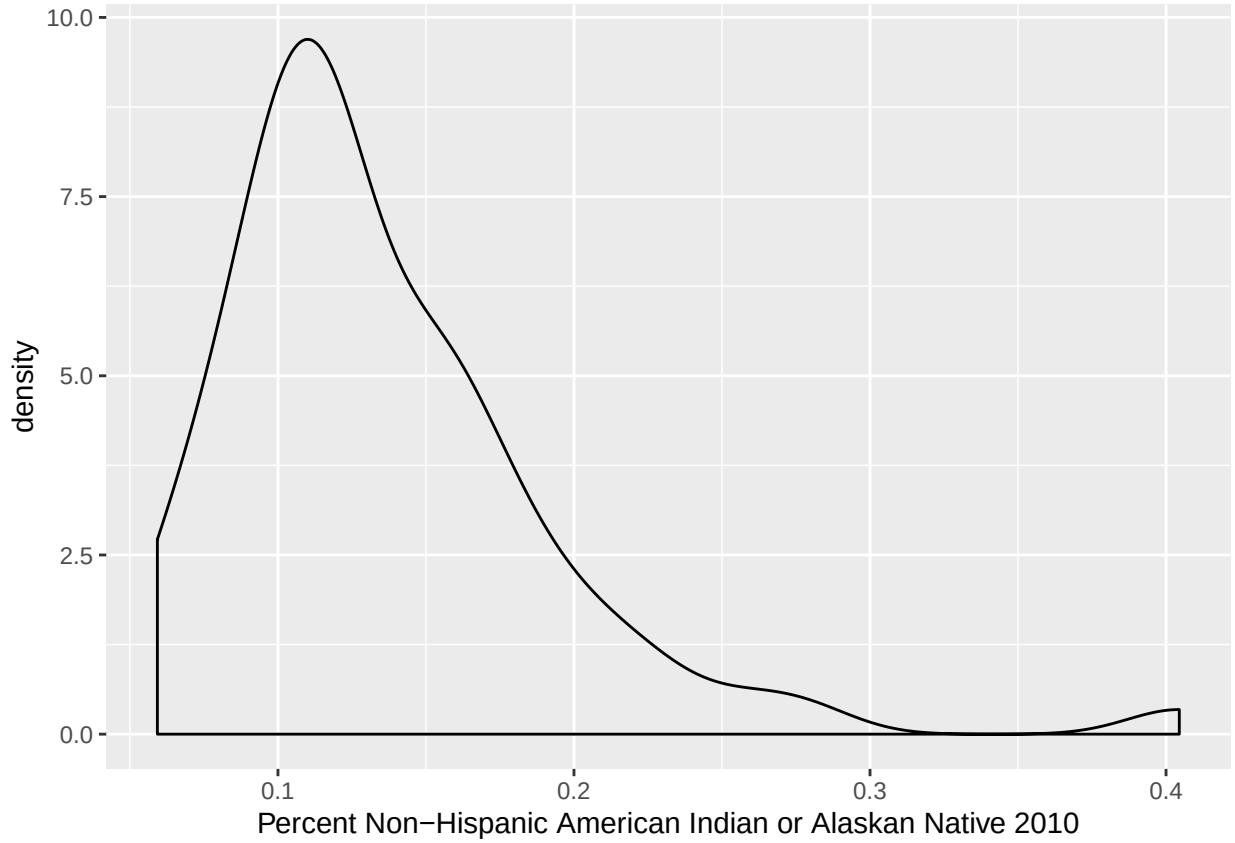

```
ggplot(fullData) + geom_density(aes(x=PCT_HISP10)) + xlab("Percent Hispanic 2010")
```



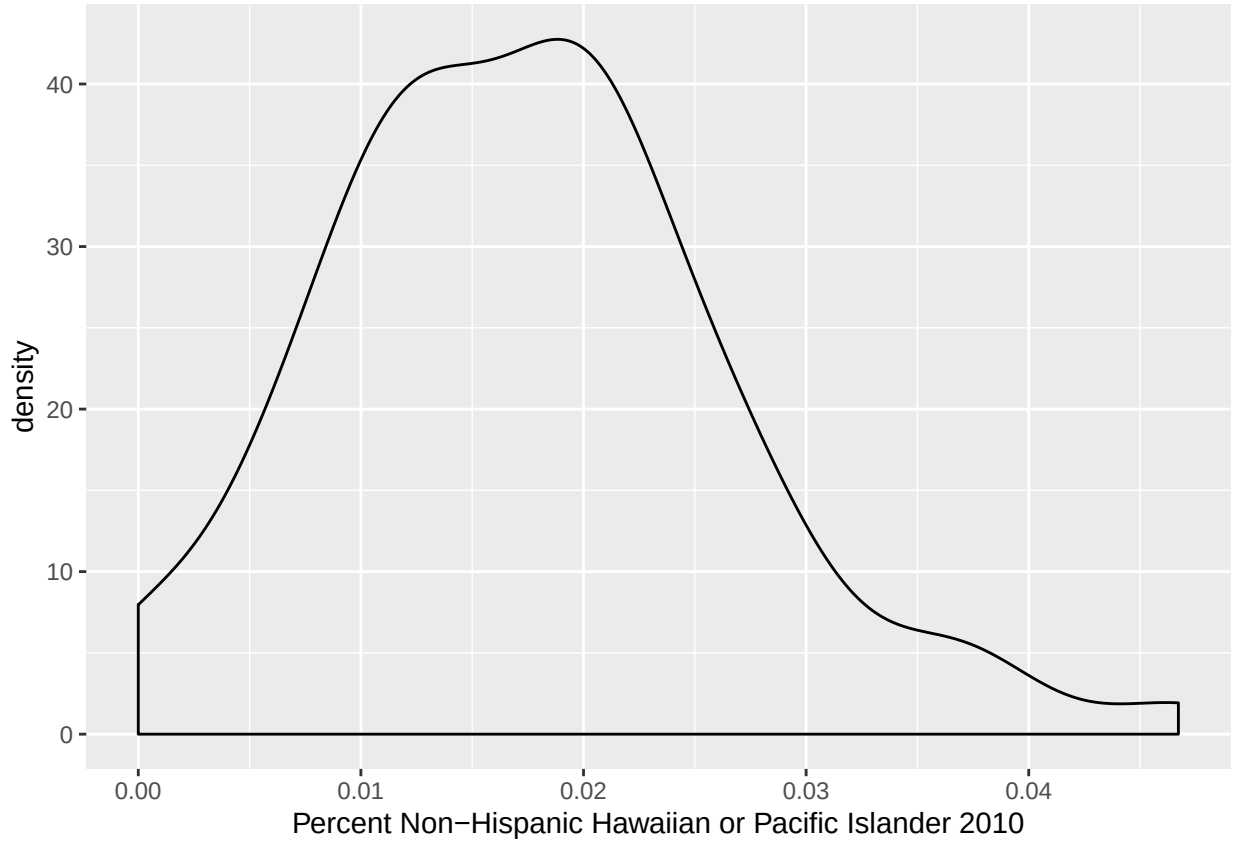

```
ggplot(fullData) + geom_density(aes(x=PCT_NHASIAN10)) + xlab("Percent Non-Hispanic Asian 2010")
```




```
ggplot(fullData) + geom_density(aes(x=PCT_NHNA10)) + xlab("Percent Non-Hispanic American Indian or Alask
```



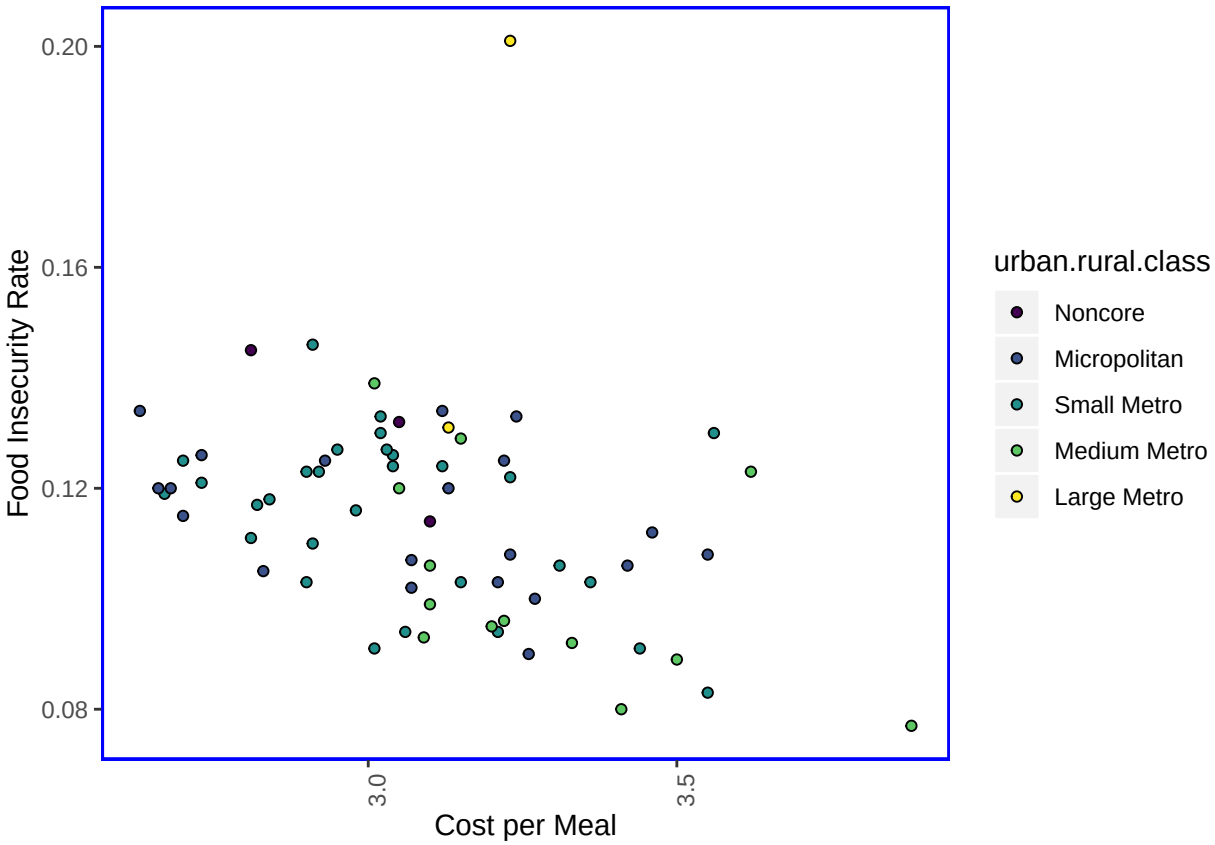

```
ggplot(fullData) + geom_density(aes(x=PCT_NHPI10)) + xlab("Percent Non-Hispanic Hawaiian or Pacific Isla
```



#Bivariate EDA

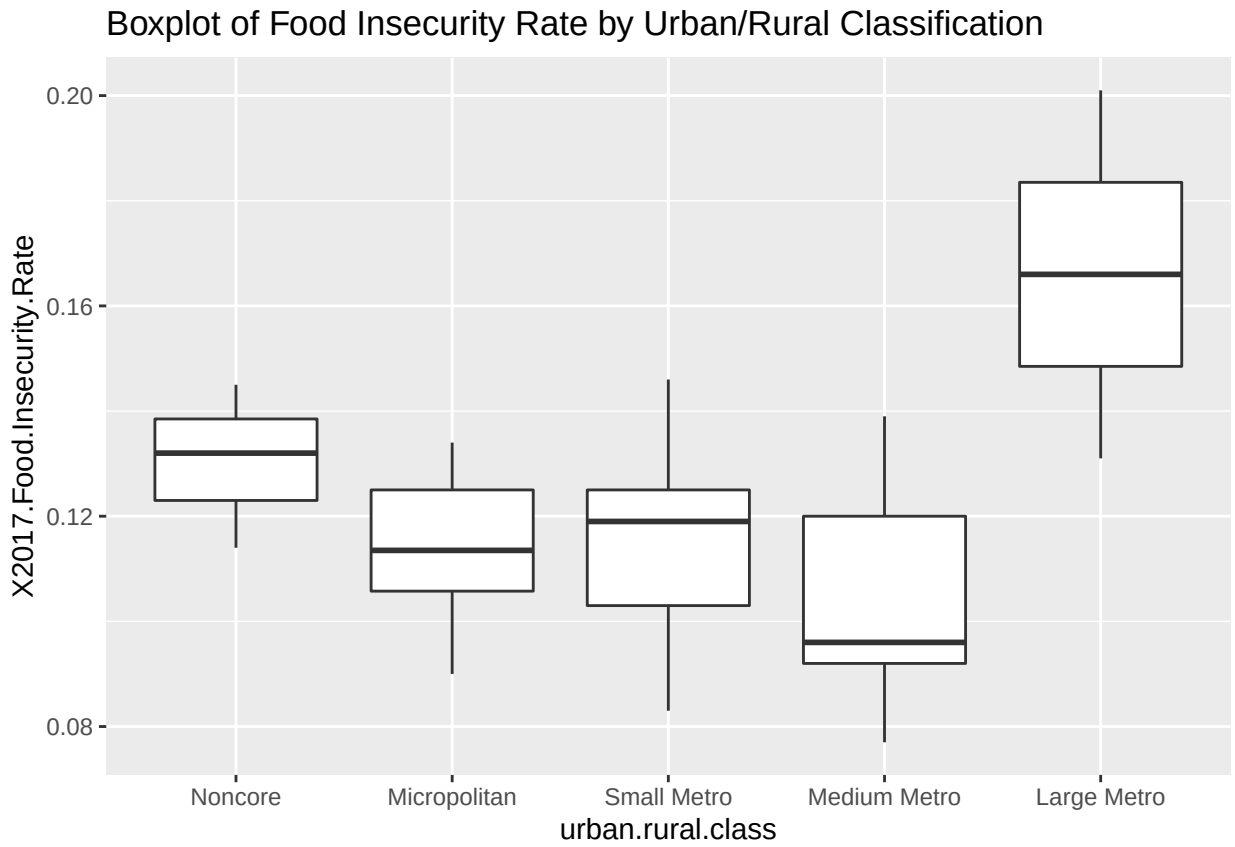
Scatterplot of food insecurity rate vs cost per meal colored by urban/rural classification.

```
ggplot(fullData, aes(x = X2017.Cost.Per.Meal, y = X2017.Food.Insecurity.Rate)) + geom_point(aes(fill=url
```



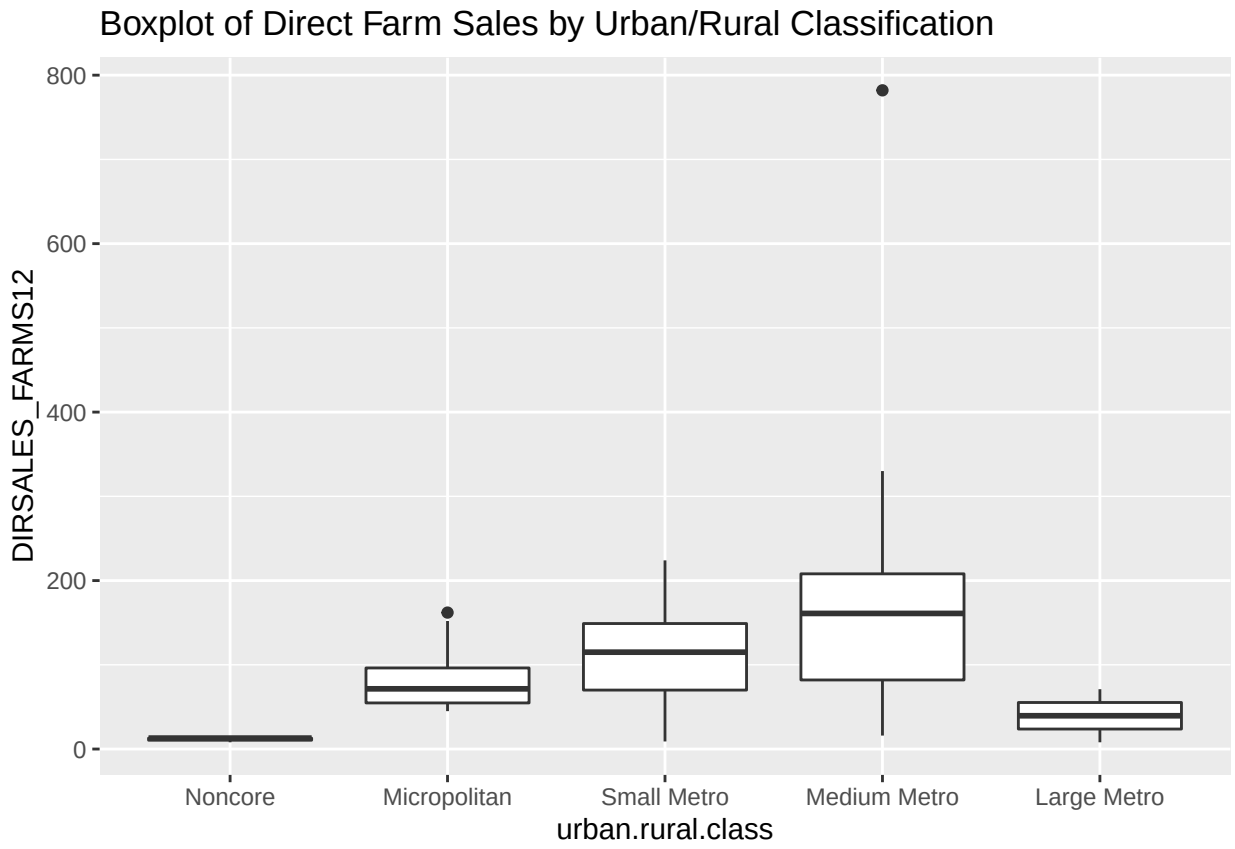
Boxplot of food insecurity rate for each urban/rural classification. We can see that large metro has the highest overall food insecurity rate with its first quartile higher than the maximum of noncore, the second highest food insecurity rate overall. We can see that medium metro has the lowest median food insecurity rate, and that micropolitan and small metro have similar insecurity distributions.

```
ggplot(fullData) + geom_boxplot(aes(x=urban.rural.class, y=X2017.Food.Insecurity.Rate)) + ggtitle("Boxpl
```

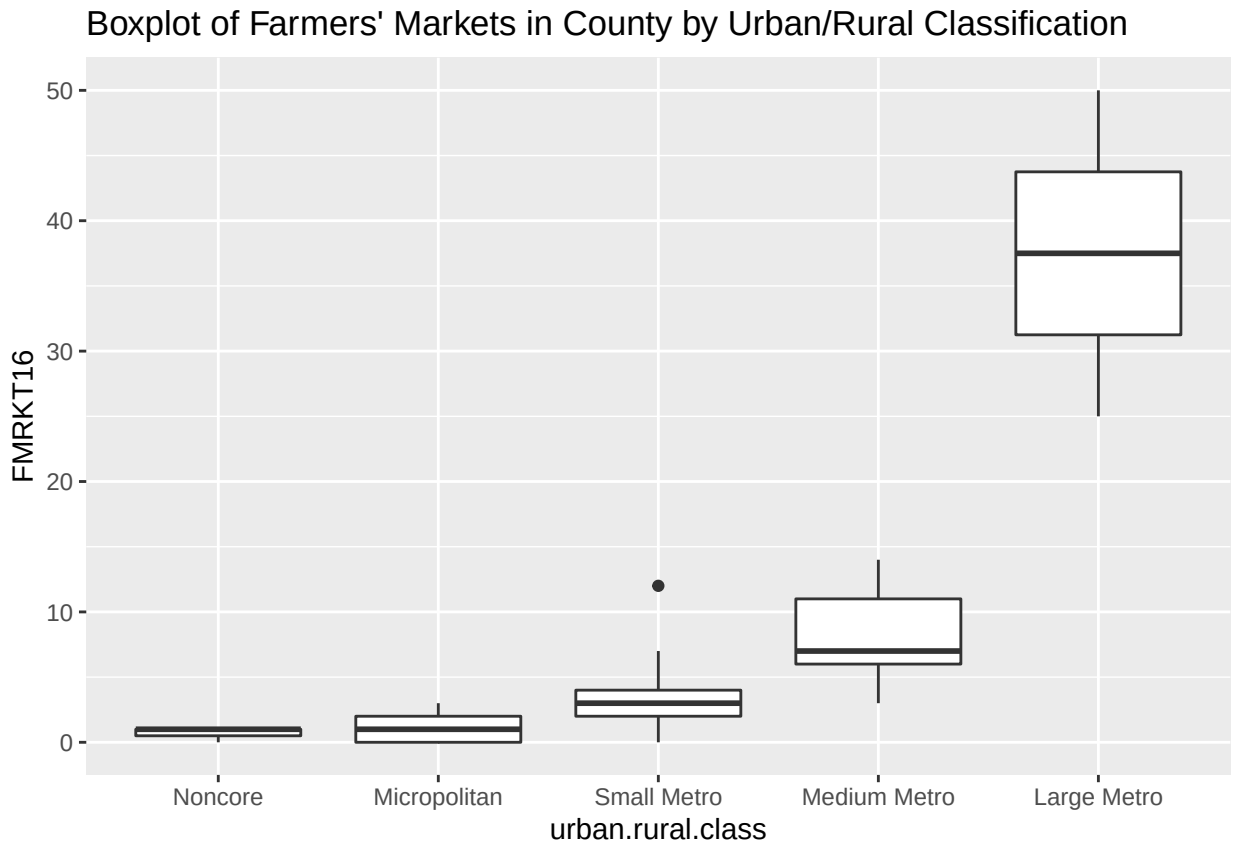


###Farmers' Markets and Urban/Rural classification

```
ggplot(fullData) + geom_boxplot(aes(x=urban.rural.class, y=DIRSALES_FARMS12)) + ggtitle("Boxplot of Dir
```

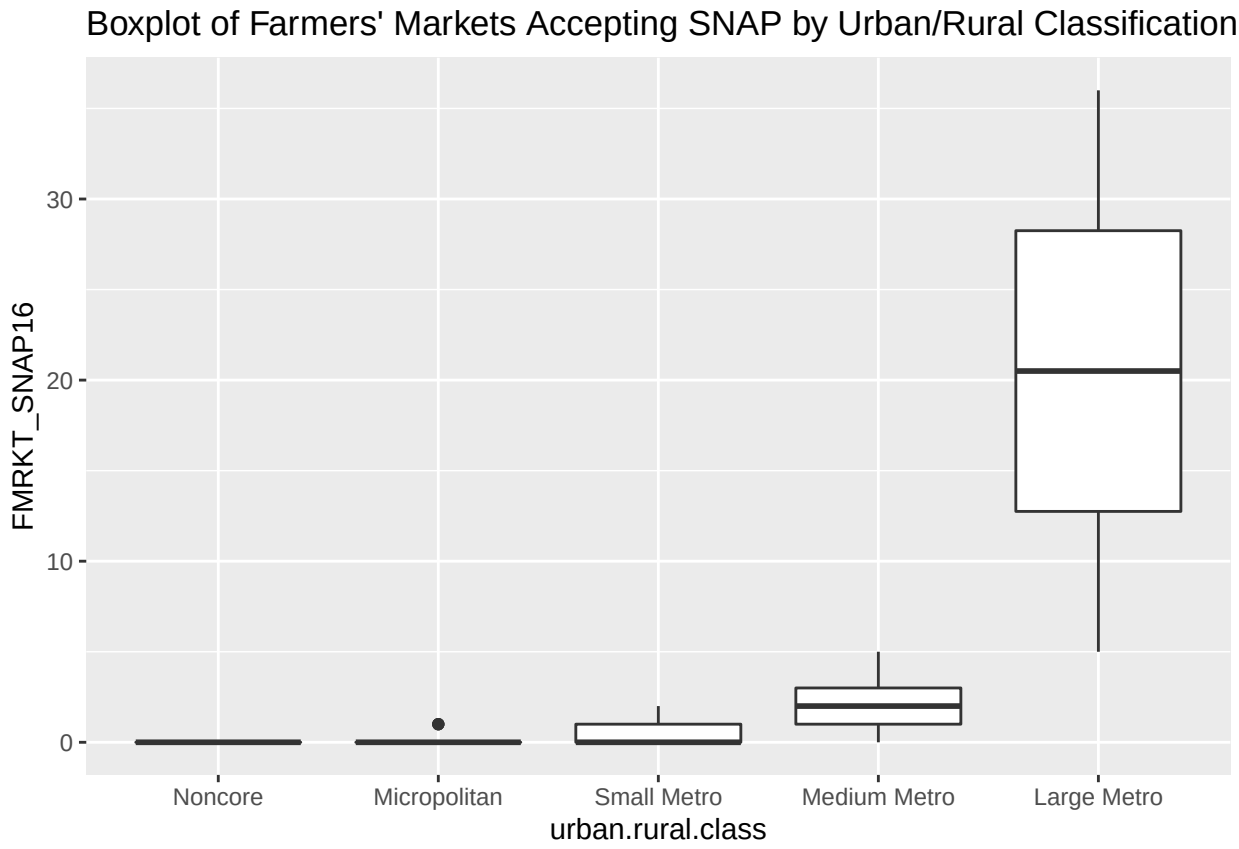



```
ggplot(fullData) + geom_boxplot(aes(x=urban.rural.class, y=FMRKT16)) + ggtitle("Boxplot of Farmers' Mar
```




```
ggplot(fullData) + geom_boxplot(aes(x=urban.rural.class, y=FMRKT_SNAP16)) + ggtitle("Boxplot of Farmers
```

```
## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
```




```
# FEA variables vs Feeding America food insecurity

cor(fullData$X2017.Food.Insecurity.Rate, fullData$LACCESS_POP15)
```

```
## [1] -0.2597942
```

```
cor(fullData$X2017.Food.Insecurity.Rate, fullData$PCT_FREE_LUNCH14)
```

```
## [1] 0.7375572
```

```
cor(fullData$X2017.Food.Insecurity.Rate, fullData$PCT_DIABETES_ADULTS13)
```

```
## [1] 0.3430861
```

```
cor(fullData$X2017.Food.Insecurity.Rate, fullData$PCT_OBESE_ADULTS13)
```

```
## [1] 0.3051705
```

```
cor(fullData$X2017.Food.Insecurity.Rate, fullData$DIRSALES_FARMS12)
```

```
## [1] -0.3457519
```

```
cor(fullData$X2017.Food.Insecurity.Rate, fullData$PCT_LOCLFARM12)
```

```
## [1] 0.2771358
```

```
cor(fullData$X2017.Food.Insecurity.Rate, fullData$FMRKT16)
```

```
## [1] 0.3810322
```

```
cor(fullData$X2017.Food.Insecurity.Rate, fullData$PCT_NHWHITE10)
```

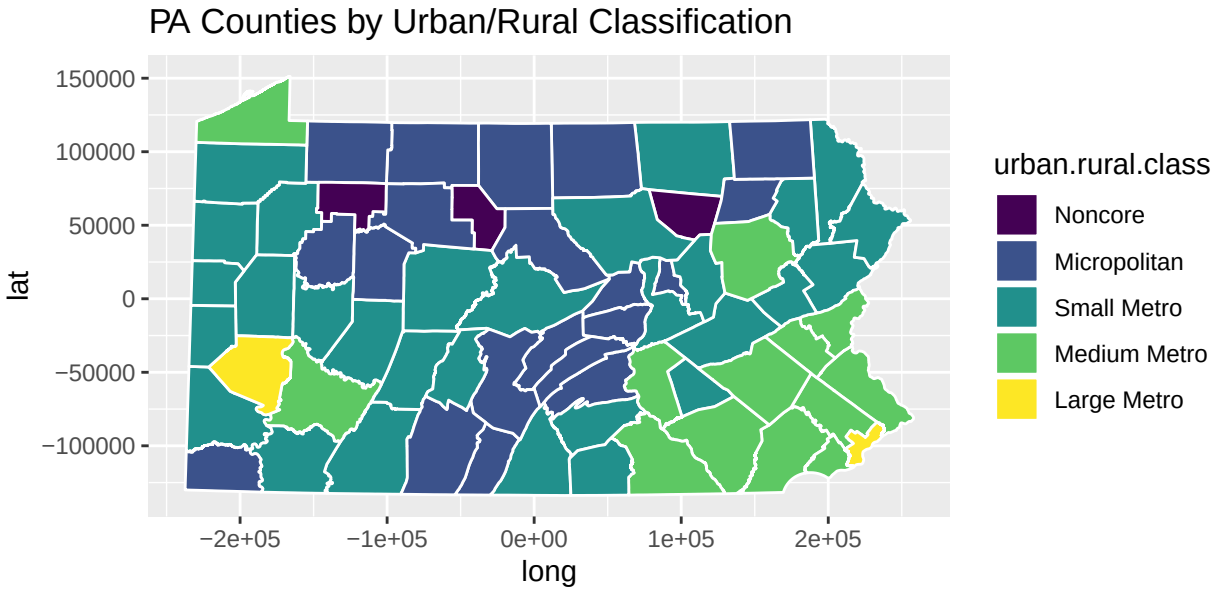
```
## [1] -0.2467364
```

```
cor(fullData$X2017.Food.Insecurity.Rate, fullData$PCT_NHBLACK10)
```

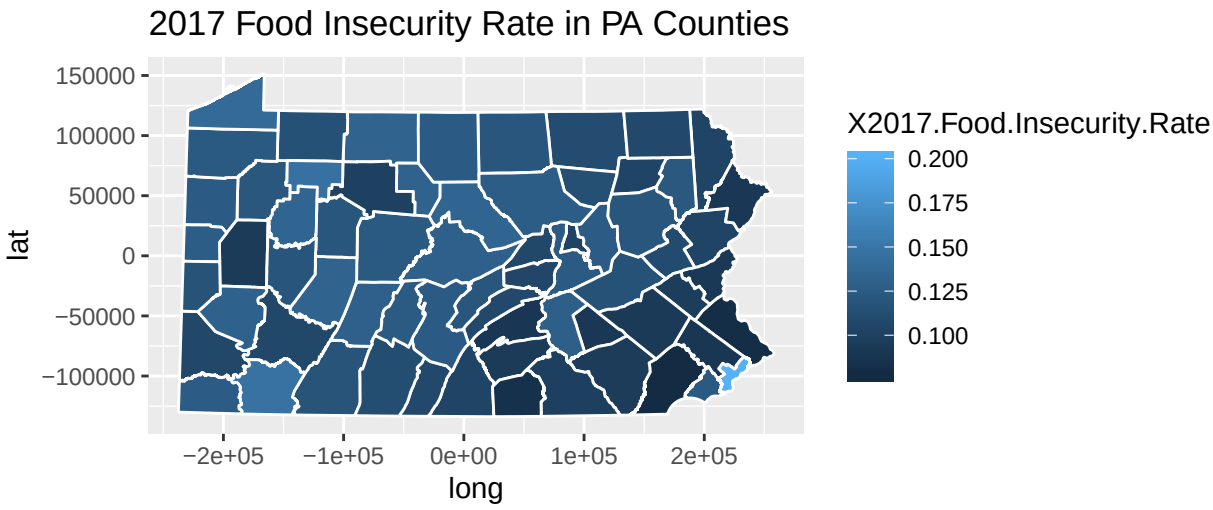
```
## [1] 0.5149624
```

#Mapping Variables

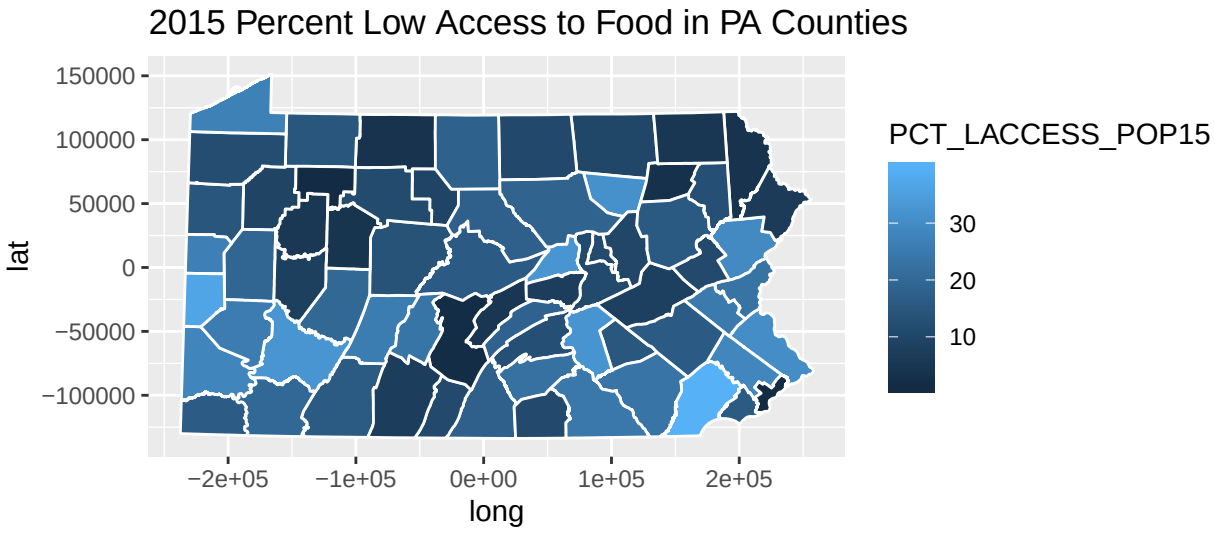
```
ggplot() + geom_polygon(data = together, aes(x = long, y = lat, fill = urban.rural.class, group=group),
```



```
#X2017.Food.Insecurity.Rate: the percentage of the population that experienced food insecurity at some
#Hint: Higher value = higher food insecurity
ggplot() + geom_polygon(data = together, aes(x = long, y = lat, fill = X2017.Food.Insecurity.Rate, grou
```




```
#PCT_LACCESS_POP15: Percentage of people in a county living more than 1 mile from a supermarket, superc
#Hint: Higher value = lower food access
ggplot() + geom_polygon(data = together, aes(x = long, y = lat, fill = PCT_LACCESS_POP15, group=group),
```



I can see that Philadelphia county has the highest rate of food insecurity by the first map, which is interesting because the second map shows Philadelphis having the best access to food. Looking at the 3 noncore regions, I can see that they each have a middle range (.125-.175) food insecurity rate, but one has relatively good food access, one has medium food access, and the other has low food access. In fact, the county with the lowest food insecurity of the noncore counties has the lowest access to food, and vice versa. This demonstrates that proximity to a grocery store is not a direct indication of food insecurity. In fact, by looking at the maps and the correlation, there appears to be a negative relationship between the two.