Death by Chocolate? No, Death by County. Modelling County-Level Mortality Rates

Introduction

Mortality rates from many causes of death are known to vary across counties. We wanted to explore what other factors about counties are correlated to county-level mortality rates from various causes to better understand what some causes of the relative geographic prevalence of different causes of death may be. Our analysis and visualizations display actual variation in causes of death between counties, and show both the power and the shortcomings of using statistical modelling techniques to estimate causes of death based on other factors about counties.

Motivation/Purpose

It is understood that mortality rates from different causes vary between regions of the United States and even between counties in the same state, but those differences and potential reasons for them have not been systematically explored.

Background

Data Sources

The first data source we used was a compilation of county-level data from a variety of sources compiled by Github user Deleetdk. Most of the data sources included are those from a paper published by Emil O. W. Kirkegaard in the journal *Open Quantitative Sociology and Political Science* in 2016, "Inequality across US counties: an S factor analysis". This data source was, very luckily for us, an RData file on GitHub that was very easy to download.

The demographic data (such as racial composition, gender, and age) come from the United States Census Bureau's annual American Community Survey through the American FactFinder. This is also the source for some economic indicators (educational attainment, sector composition of employment). Another data source is the American Human Development Index of the Measure of America, which contains health, education, and income indicators. Many health indicators come from the Robert Wood Johnson Foundation's County Health Rankings & Roadmaps. Finally, the county-level electoral data comes from the failing New York Times.

The first step of data wrangling with this dataset was selecting the variables we wanted from it. We then removed the state-level rows, which all have FIPS codes of X000, so that the data was only counties.

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
load("USA_county_data.RData")
countydata <- USA_county_data %>% select(fips, name_16, rep16_frac,
    dem16_frac, rep16_frac2, dem16_frac2, statecode_prev, rep12_frac,
    dem12_frac, rep12_frac2, dem12_frac2, rep08_frac, dem08_frac,
    rep08_frac2, dem08_frac2, Less.Than.High.School, At.Least.High.School.Diploma,
    At.Least.Bachelor.s.Degree, Graduate.Degree, School.Enrollment,
    Median.Earnings.2010.dollars, White.not.Latino.Population,
    African.American.Population, Native.American.Population,
    Asian.American.Population, Latino.Population, Children.Under.6.Living.in.Poverty,
    Adults.65.and.Older.Living.in.Poverty, Total.Population,
    Preschool.Enrollment.Ratio.enrolled.ages.3.and.4, Poverty.Rate.below.federal.poverty.threshold,
    Child.Poverty.living.in.families.below.the.poverty.line,
    Management.professional.and.related.occupations, Service.occupations,
    Sales.and.office.occupations, Farming.fishing.and.forestry.occupations,
    Construction.extraction.maintenance.and.repair.occupations,
    Production.transportation.and.material.moving.occupations,
    State, median_age, Poor.physical.health.days, Poor.mental.health.days,
    Low.birthweight, Teen.births, Children.in.single.parent.households,
    Adult.smoking, Adult.obesity, Diabetes, Sexually.transmitted.infections,
    HIV.prevalence.rate, Uninsured, Unemployment)

countydata <- countydata %>% filter(!fips %in% seq(0, 1e+05,
    by = 1000)) %>% arrange(fips)
```

The key dataset for this analysis is a set of estimates of "annual mortality rates by US county for 21

mutually exclusive causes of death from 1980 through 2014" constructed by Laura Dwyer-Lindgren, Amelia Bertozzi-Villa, Rebecca Stubbs, et al. from the National Center for Health Statistics' National Vital Statistics System and published in the Journal of the American Medical Association. We downloaded the dataset itself from Kaggle, which it was provided to by the Institute for Health Metrics and Evaluation.

The causes of death (and our short names for them) are:

HIV: HIV-AIDS and Tuberculosis

INF: Diarrhea, lower respiratory, and other common infectious diseases

TROP: Neglected tropical diseases and malaria

MAT: Maternal disorders

NEON: Neonatal disorders

NUT: Nutritional deficiencies

OTHC: Other communicable, maternal, neonatal, and nutritional diseases

NEOP: Neoplasms

CHRON: Chronic respiratory diseases

CIRR: Cirrhosis and other chronic liver diseases

DIG: Digestive diseases

NEUR: Neurological disorders

MENSUB: Mental and substance use disorders

DIAB: Diabetes, urogenital, blood, and endocrine diseases

MUSC: Musculoskeletal disorders

OTHN: Other non-communicable diseases

TRAN: Transport injuries

UNIN: Unintentional injuries

SELF: Self-harm and interpersonal violence

WAR: Forces of nature, war, and legal intervention

This dataset was an Excel file with estimates with estimated mortality rates for each county for each of 20 causes of mortality for each of many years (1980, 1985, 1990, 1995, 2000, 2005, 2010, and 2014). The first thing we had to do was change each estimate from a midpoint and a confidence interval to just the midpoint. We also renamed the sheets and columns to make them easier to work with. Then, because each cause of death was its own sheet, we had to make a function to read in all of the sheets to one dataframe in RStudio.

```
library(readxl)
read_excel_allsheets <- function(filename) {
    sheets <- readxl::excel_sheets(filename)
    x <- lapply(sheets, function(X) readxl::read_excel(filename,
        sheet = X))
    names(x) <- sheets
    x
}
mortalityrates <- read_excel_allsheets("MortalityRatesClean.xlsx")
```

```
## Warning in strptime(x, format, tz = tz): unknown timezone 'zone/tz/2017c.
## 1.0/zoneinfo/America/Los_Angeles'
```

```
mortalityrates <- as.data.frame(mortalityrates)
```

Because each cause of death was its own sheet, when we imported the data there were 20 columns of the FIPS codes and 20 columns of location (the county names). As a result, we selected one of the location and FIPS and all of the mortality rate estimates to keep.

```
require(dplyr)
mortalityrates <- rename(mortalityrates, fips = HIV.FIPS)
mortalityrates <- rename(mortalityrates, location = HIV.Location)
mortalityrates <- mortalityrates %>% select(location, fips, HIV.HIV80,
    HIV.HIV85, HIV.HIV90, HIV.HIV95, HIV.HIV00, HIV.HIV05, HIV.HIV10,
    HIV.HIV14, INF.INF80, INF.INF85, INF.INF90, INF.INF95, INF.INF00,
    INF.INF05, INF.INF10, INF.INF14, TROP.TROP80, TROP.TROP85,
    TROP.TROP90, TROP.TROP95, TROP.TROP00, TROP.TROP05, TROP.TROP10,
    TROP.TROP14, MAT.MAT80, MAT.MAT85, MAT.MAT90, MAT.MAT95,
    MAT.MAT00, MAT.MAT05, MAT.MAT10, MAT.MAT14, NEON.NEON80,
    NEON.NEON85, NEON.NEON90, NEON.NEON95, NEON.NEON00, NEON.NEON05,
    NEON.NEON10, NEON.NEON14, NUT.NUT80, NUT.NUT85, NUT.NUT90,
    NUT.NUT95, NUT.NUT00, NUT.NUT05, NUT.NUT10, NUT.NUT14, OTHC.OTHC80,
    OTHC.OTHC85, OTHC.OTHC90, OTHC.OTHC95, OTHC.OTHC00, OTHC.OTHC05,
    OTHC.OTHC10, OTHC.OTHC14, NEOP.NEOP80, NEOP.NEOP85, NEOP.NEOP90,
    NEOP.NEOP95, NEOP.NEOP00, NEOP.NEOP05, NEOP.NEOP10, NEOP.NEOP14,
    CHRON.CHRON80, CHRON.CHRON85, CHRON.CHRON90, CHRON.CHRON95,
    CHRON.CHRON00, CHRON.CHRON05, CHRON.CHRON10, CHRON.CHRON14,
    CIRR.CIRR80, CIRR.CIRR85, CIRR.CIRR90, CIRR.CIRR95, CIRR.CIRR00,
    CIRR.CIRR05, CIRR.CIRR10, CIRR.CIRR14, DIG.DIG80, DIG.DIG85,
    DIG.DIG90, DIG.DIG95, DIG.DIG00, DIG.DIG05, DIG.DIG10, DIG.DIG14,
    NEUR.NEUR80, NEUR.NEUR85, NEUR.NEUR90, NEUR.NEUR95, NEUR.NEUR00,
    NEUR.NEUR05, NEUR.NEUR10, NEUR.NEUR14, MENSUB.MENSUB80, MENSUB.MENSUB85,
    MENSUB.MENSUB90, MENSUB.MENSUB95, MENSUB.MENSUB00, MENSUB.MENSUB05,
    MENSUB.MENSUB10, MENSUB.MENSUB14, DIAB.DIAB80, DIAB.DIAB85,
    DIAB.DIAB90, DIAB.DIAB95, DIAB.DIAB00, DIAB.DIAB05, DIAB.DIAB10,
    DIAB.DIAB14, MUSC.MUSC80, MUSC.MUSC85, MUSC.MUSC90, MUSC.MUSC95,
    MUSC.MUSC00, MUSC.MUSC05, MUSC.MUSC10, MUSC.MUSC14, OTHN.OTHN80,
    OTHN.OTHN85, OTHN.OTHN90, OTHN.OTHN95, OTHN.OTHN00, OTHN.OTHN05,
    OTHN.OTHN10, OTHN.OTHN14, TRAN.TRAN80, TRAN.TRAN85, TRAN.TRAN90,
    TRAN.TRAN95, TRAN.TRAN00, TRAN.TRAN05, TRAN.TRAN10, TRAN.TRAN14,
    UNIN.UNIN80, UNIN.UNIN85, UNIN.UNIN90, UNIN.UNIN95, UNIN.UNIN00,
    UNIN.UNIN05, UNIN.UNIN10, UNIN.UNIN14, SELF.SELF80, SELF.SELF85,
    SELF.SELF90, SELF.SELF95, SELF.SELF00, SELF.SELF05, SELF.SELF10,
    SELF.SELF14, WAR.WAR80, WAR.WAR85, WAR.WAR90, WAR.WAR95,
    WAR.WAR00, WAR.WAR05, WAR.WAR10, WAR.WAR14)
```

Similar to with the other dataset, there were state rows as well as county rows. However, in this dataset the states had low FIPS codes (below 100) instead of FIPS codes of multiples of 1000. Additionally, the mortality rates dataset had counties in Puerto Rico (FIPS codes in the 72000s), but we removed these because they did not have data in the Deleetdk data.

```
mortalityrates <- mortalityrates %>% filter(fips > 100) %>% filter(fips <
    70000) %>% arrange(fips)
```

At this point, we combined the two datasets.

```
require(dplyr)
countyrates <- inner_join(countydata, mortalityrates, by = c("fips"))
```

When we thought we needed the latitude and longitude of each county, we downloaded a file that contained that information and renamed and selected the variables for them.

```
require(dplyr)
`2015_Gaz_counties_national` <- read.delim("2015_Gaz_counties_national.txt")
`2015_Gaz_counties_national` <- rename(`2015_Gaz_counties_national`,
    fips = GEOID)
`2015_Gaz_counties_national` <- rename(`2015_Gaz_counties_national`,
    latitude = INTPTLAT)
`2015_Gaz_counties_national` <- rename(`2015_Gaz_counties_national`,
    longitude = INTPTLONG)
`2015_Gaz_counties_national` <- `2015_Gaz_counties_national` %>%
    select(fips, latitude, longitude)
```

We then merged that dataset into our main dataframe, countyrates.

```
countyrates <- left_join(countyrates, `2015_Gaz_counties_national`,
    by = c("fips"))
```

After combining the datasets, we noticed there were several observations with missing data for some variables. In particular, all of Alaska was missing most of the Deleetdk data and a few stray observations were missing the education data and a number of other variables. We removed those observations.

```
countyratesnogaps <- countyrates %>% filter(fips > 100) %>% filter(fips <
    70000) %>% filter(!is.na(rep16_frac)) %>% filter(!is.na(At.Least.High.School.Diploma)) %>%
    arrange(fips)
```

We also chose to remove some of the variables that had a lot of missing data by selecting the others.

Finally, we formally created the dataframe we would use of all the county observations with tidy data for all of our variables of interest.

```
countyratesfullcases <- countyratesnogaps[complete.cases(countyratesnogaps),
    ]
```

Model Creation/Statistical Computation

We created two models for each of the twenty causes of death - a regression tree and a k nearest neighbors model.

```
require(caret)
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
require(rpart)
```

```
## Loading required package: rpart
```

```
require(rpart.plot)
```

```
## Loading required package: rpart.plot
```

```
library(tree)
```

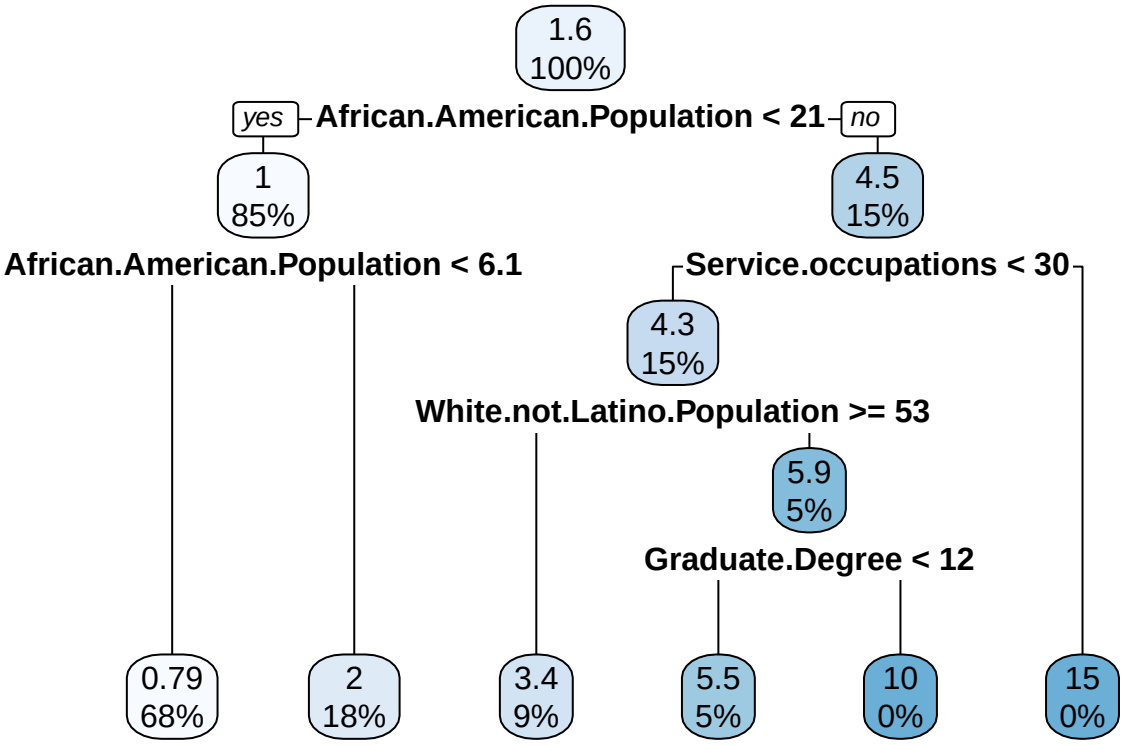
To give a sense for our model-building process, we have included the R code for the model creation for the cause of death, HIV-AIDS and tuberculosis. All of the other models follow the same format, and only the regression trees and the mean square errors from the k-nearest neighbors model are shown for them.

HIV: HIV-AIDS and Tuberculosis

Regression tree and mean square error

```
fips <- countyratesfullcases$fips
latitude <- countyratesfullcases$latitude
longitude <- countyratesfullcases$longitude
```

```
require(dplyr)
set.seed(1)
fitControl <- trainControl(method = "cv")
tr.HIV14 <- train(HIV.HIV14 ~ rep16_frac + dem16_frac + Less.Than.High.School +
    At.Least.High.School.Diploma + At.Least.Bachelor.s.Degree +
    Graduate.Degree + School.Enrollment + Median.Earnings.2010.dollars +
    White.not.Latino.Population + African.American.Population +
    Native.American.Population + Asian.American.Population +
    Latino.Population + Adults.65.and.Older.Living.in.Poverty +
    Total.Population + Poverty.Rate.below.federal.poverty.threshold +
    Child.Poverty.living.in.families.below.the.poverty.line +
    Management.professional.and.related.occupations + Service.occupations +
    Sales.and.office.occupations + Farming.fishing.and.forestry.occupations +
    Construction.extraction.maintenance.and.repair.occupations +
    Production.transportation.and.material.moving.occupations +
    median_age + Children.in.single.parent.households + Adult.obesity +
    Diabetes + Uninsured + Unemployment, data = countyratesfullcases,
    method = "rpart2", trControl = fitControl, tuneGrid = data.frame(maxdepth = 1:20))
rpart.plot(tr.HIV14$finalModel)
```




```
mortrate <- countyratesfullcases$HIV.HIV14
cause <- rep("HIV", 3110)
cause <- as.data.frame(cause)
model <- rep("real", 3110)
model <- as.data.frame(model)
HIVreal <- cbind(fips, cause, model, mortrate)
```

```
realHIV14 <- countyratesfullcases$HIV.HIV14
preds.tr.HIV14 <- predict(tr.HIV14$finalModel, newdata = countyratesfullcases)
HIV14 <- cbind.data.frame(preds.tr.HIV14, realHIV14)
HIV14 <- HIV14 %>% mutate(HIVtreediff = ((preds.tr.HIV14 - realHIV14)/realHIV14))
mortrate <- HIV14$HIVtreediff
cause <- rep("HIV", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
HIVtree <- cbind(fips, cause, model, mortrate)

treeHIV14.mse <- mean((preds.tr.HIV14 - realHIV14)^2)
treeHIV14.mse
```

```
## [1] 2.003797
```

k-Nearest Neighbors mean square error

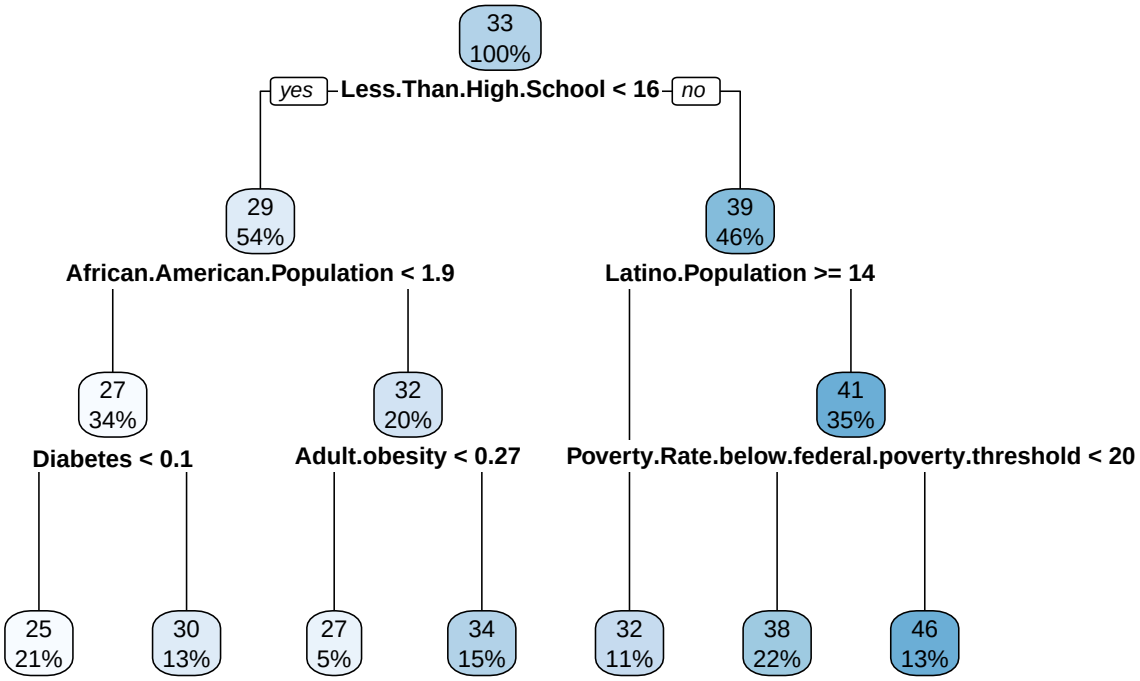
```
set.seed(4747)
fitControl <- trainControl(method = "none")
knn.tr.HIV14 <- train(HIV.HIV14 ~ rep16_frac + dem16_frac + Less.Than.High.School +
    At.Least.High.School.Diploma + At.Least.Bachelor.s.Degree +
    Graduate.Degree + School.Enrollment + Median.Earnings.2010.dollars +
    White.not.Latino.Population + African.American.Population +
    Native.American.Population + Asian.American.Population +
```

```
    Latino.Population + Adults.65.and.Older.Living.in.Poverty +
    Total.Population + Poverty.Rate.below.federal.poverty.threshold +
    Child.Poverty.living.in.families.below.the.poverty.line +
    Management.professional.and.related.occupations + Service.occupations +
    Sales.and.office.occupations + Farming.fishing.and.forestry.occupations +
    Construction.extraction.maintenance.and.repair.occupations +
    Production.transportation.and.material.moving.occupations +
    median_age + Children.in.single.parent.households + Adult.obesity +
    Diabetes + Uninsured + Unemployment, data = countyratesfullcases,
    method = "knn", trControl = fitControl, tuneGrid = data.frame(k = 5))

preds.knn.tr.HIV14 <- predict(knn.tr.HIV14, newdata = countyratesfullcases)
realknnHIV14 <- countyratesfullcases$HIV.HIV14
knnHIV14 <- cbind.data.frame(preds.knn.tr.HIV14, realknnHIV14)
knnHIV14 <- knnHIV14 %>% mutate(HIVknndiff = ((preds.knn.tr.HIV14 -
    realknnHIV14)/realknnHIV14))
mortrate <- knnHIV14$HIVknndiff
cause <- rep("HIV", 3110)
cause <- as.data.frame(cause)
model <- rep("knn", 3110)
model <- as.data.frame(model)
HIVknn <- cbind(fips, cause, model, mortrate)

knnHIV14.mse <- mean((preds.knn.tr.HIV14 - realknnHIV14)^2)
knnHIV14.mse
```

```
## [1] 3.017696
```

INF: Diarrhea, lower respiratory, and other common infectious diseases

Regression tree and mean square error




```
realINF14 <- countyratesfullcases$INF.INF14
preds.tr.INF14 <- predict(tr.INF14$finalModel, newdata = countyratesfullcases)
INF14 <- cbind.data.frame(preds.tr.INF14, realINF14)
INF14 <- INF14 %>% mutate(INFtreediff = ((preds.tr.INF14 - realINF14)/realINF14))
mortrate <- INF14$INFtreediff
cause <- rep("INF", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
INFtree <- cbind(fips, cause, model, mortrate)

treeINF14.mse <- mean((preds.tr.INF14 - realINF14)^2)
treeINF14.mse
```

```
## [1] 66.67221
```

k-Nearest Neighbors mean square error

```
## [1] 79.45923
```

TROP: Neglected tropical diseases and malaria

Regression tree and mean square error

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```

```
realTROP14 <- countyratesfullcases$TROP.TROP14
preds.tr.TROP14 <- predict(tr.TROP14$finalModel, newdata = countyratesfullcases)
TROP14 <- cbind.data.frame(preds.tr.TROP14, realTROP14)
TROP14 <- TROP14 %>% mutate(TROPtreediff = ((preds.tr.TROP14 -
    realTROP14)/realTROP14))
mortrate <- TROP14$TROPtreediff
cause <- rep("TROP", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
TROPtree <- cbind(fips, cause, model, mortrate)

treeTROP14.mse <- mean((preds.tr.TROP14 - realTROP14)^2)
treeTROP14.mse
```

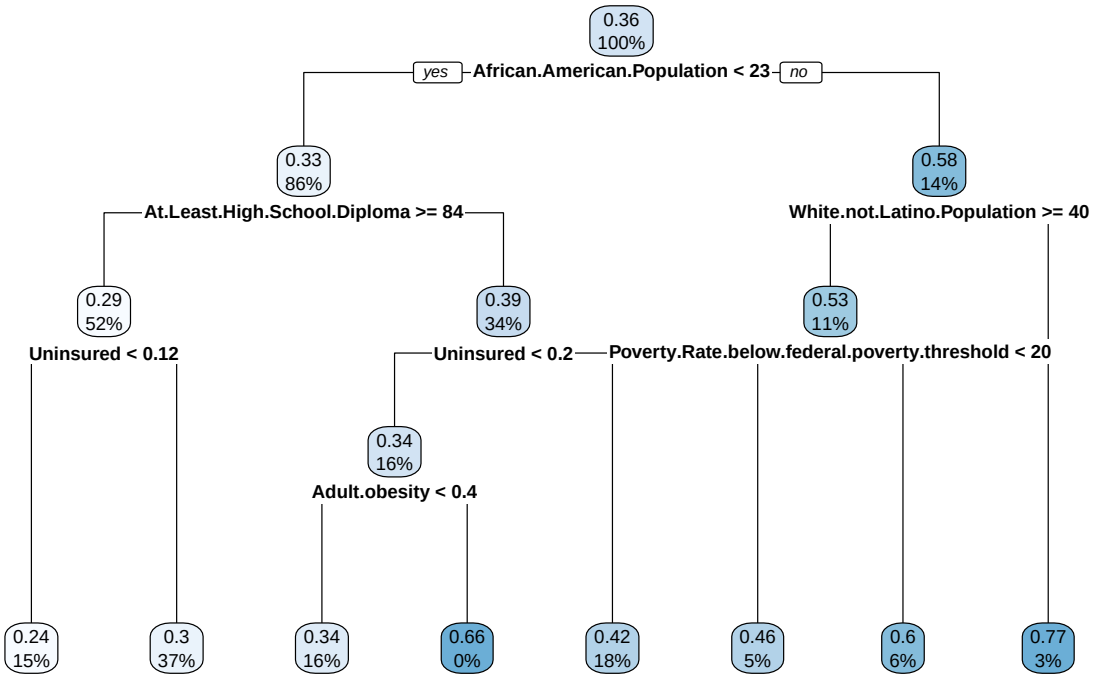
```
## [1] 0.001189699
```

k-Nearest Neighbors mean square error

```
## [1] 0.001668828
```

MAT: Maternal disorders

Regression tree and mean square error




```
realMAT14 <- countyratesfullcases$MAT.MAT14
preds.tr.MAT14 <- predict(tr.MAT14$finalModel, newdata = countyratesfullcases)
MAT14 <- cbind.data.frame(preds.tr.MAT14, realMAT14)
MAT14 <- MAT14 %>% mutate(MATtreediff = ((preds.tr.MAT14 - realMAT14)/realMAT14))
mortrate <- MAT14$MATtreediff
cause <- rep("MAT", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
MATtree <- cbind(fips, cause, model, mortrate)

treeMAT14.mse <- mean((preds.tr.MAT14 - realMAT14)^2)
treeMAT14.mse
```

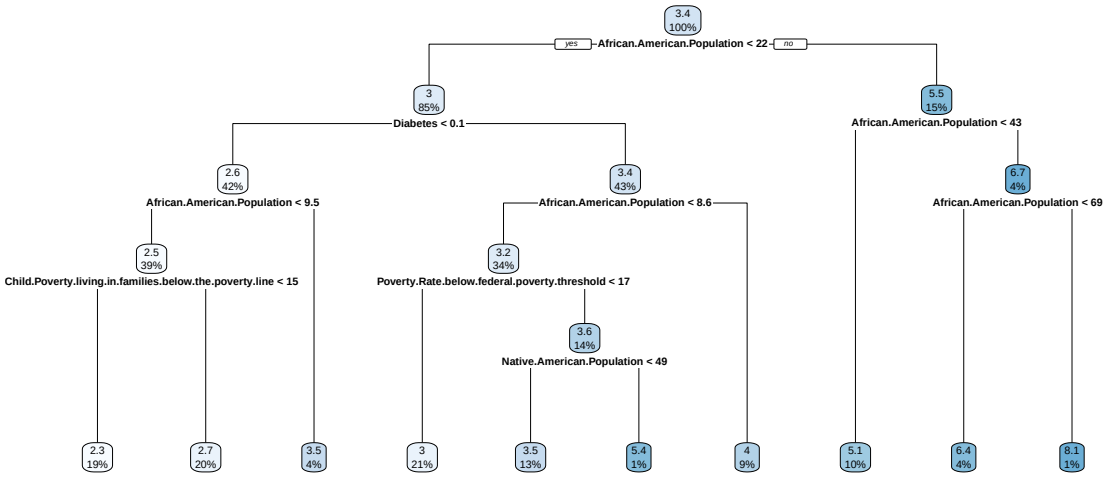
```
## [1] 0.009208227
```

k-Nearest Neighbors mean square error

```
## [1] 0.01464281
```

NEON: Neonatal disorders

Regression tree and mean square error




```
realNEON14 <- countyratesfullcases$NEON.NEON14
preds.tr.NEON14 <- predict(tr.NEON14$finalModel, newdata = countyratesfullcases)
NEON14 <- cbind.data.frame(preds.tr.NEON14, realNEON14)
NEON14 <- HIV14 %>% mutate(NEONtreediff = ((preds.tr.NEON14 -
    realNEON14)/realNEON14))
mortrate <- NEON14$NEONtreediff
cause <- rep("NEON", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
NEONtree <- cbind(fips, cause, model, mortrate)

treeNEON14.mse <- mean((preds.tr.NEON14 - realNEON14)^2)
treeNEON14.mse
```

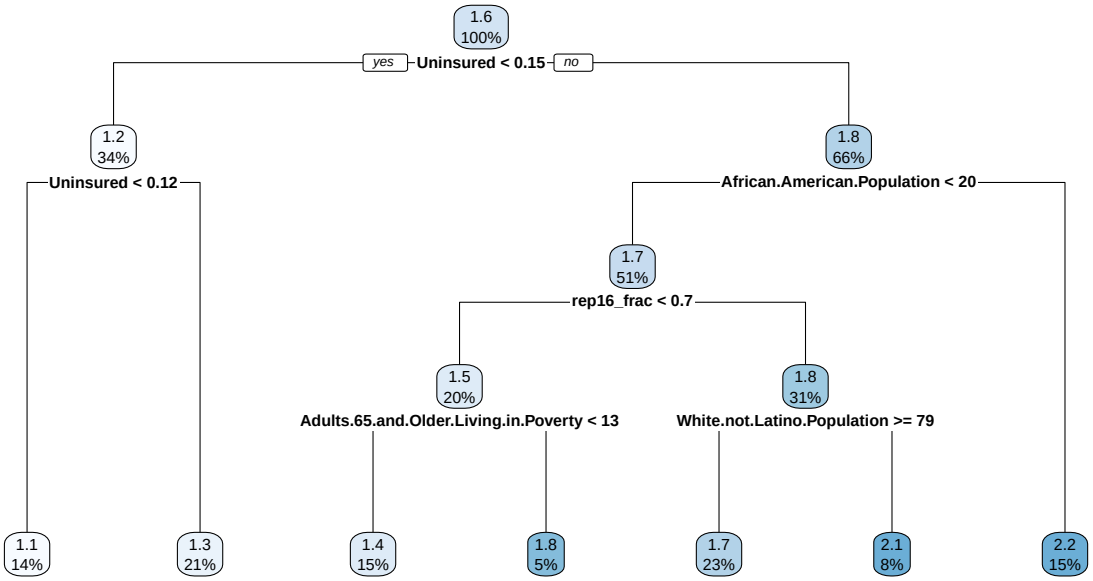
```
## [1] 0.4503517
```

k-Nearest Neighbors mean square error

```
## [1] 1.096217
```

NUT: Nutritional deficiencies

Regression tree and mean square error




```
realNUT14 <- countyratesfullcases$NUT.NUT14
preds.tr.NUT14 <- predict(tr.NUT14$finalModel, newdata = countyratesfullcases)
NUT14 <- cbind.data.frame(preds.tr.NUT14, realNUT14)
NUT14 <- NUT14 %>% mutate(NUTtreediff = ((preds.tr.NUT14 - realNUT14)/realNUT14))
mortrate <- NUT14$NUTtreediff
cause <- rep("NUT", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
NUTtree <- cbind(fips, cause, model, mortrate)

treeNUT14.mse <- mean((preds.tr.NUT14 - realNUT14)^2)
treeNUT14.mse
```

```
## [1] 0.3479154
```

k-Nearest Neighbors mean square error

```
## [1] 0.3380761
```

OTHC: Other communicable, maternal, neonatal, and nutritional diseases

Regression tree and mean square error

```
realOTHC14 <- countyratesfullcases$OTHC.OTHC14
preds.tr.OTHC14 <- predict(tr.OTHC14$finalModel, newdata = countyratesfullcases)
OTHC14 <- cbind.data.frame(preds.tr.OTHC14, realOTHC14)
OTHC14 <- HIV14 %>% mutate(OTHCtreediff = ((preds.tr.OTHC14 -
    realOTHC14)/realOTHC14))
mortrate <- OTHC14$OTHCtreediff
cause <- rep("OTHC", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
OTHCtree <- cbind(fips, cause, model, mortrate)

treeOTHC14.mse <- mean((preds.tr.OTHC14 - realOTHC14)^2)
treeOTHC14.mse
```

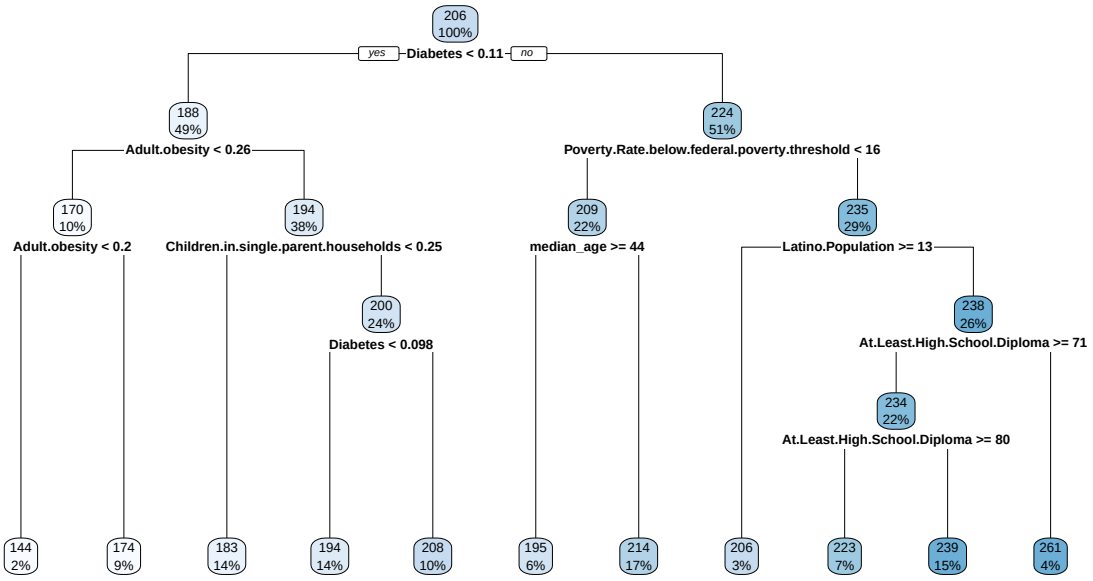
```
## [1] 0.05241569
```

k-Nearest Neighbors mean square error

```
## [1] 0.0618921
```

NEOP: Neoplasms

Regression tree and mean square error




```
realNEOP14 <- countyratesfullcases$NEOP.NEOP14
preds.tr.NEOP14 <- predict(tr.NEOP14$finalModel, newdata = countyratesfullcases)
NEOP14 <- cbind.data.frame(preds.tr.NEOP14, realNEOP14)
NEOP14 <- NEOP14 %>% mutate(NEOPtreediff = ((preds.tr.NEOP14 -
    realNEOP14)/realNEOP14))
mortrate <- NEOP14$NEOPtreediff
cause <- rep("NEOP", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
NEOPtree <- cbind(fips, cause, model, mortrate)

treeNEOP14.mse <- mean((preds.tr.NEOP14 - realNEOP14)^2)
treeNEOP14.mse
```

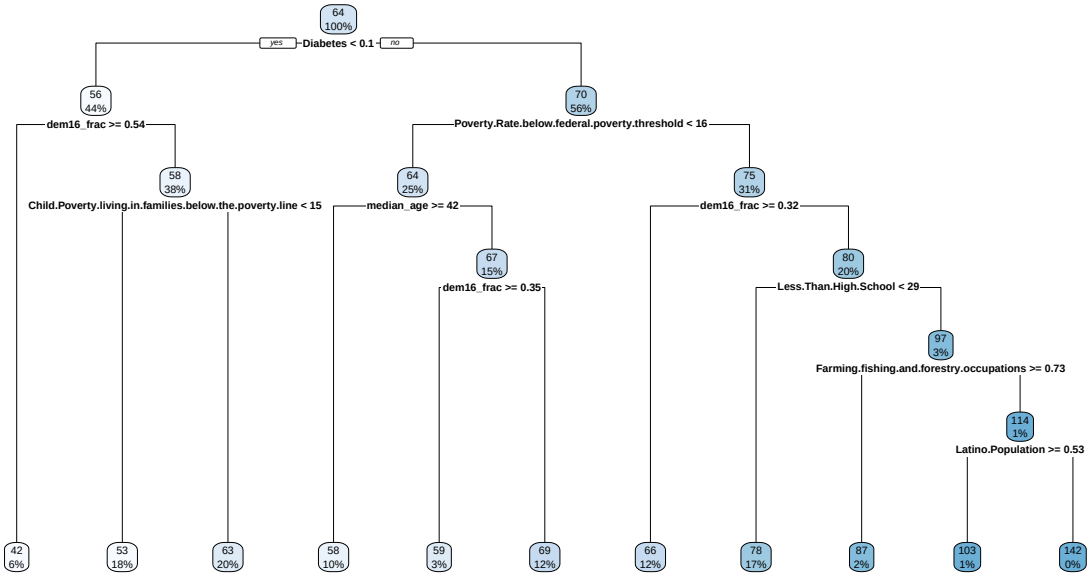
```
## [1] 437.6976
```

k-Nearest Neighbors mean square error

```
## [1] 622.1279
```

CHRON: Chronic respiratory diseases

Regression tree and mean square error




```
realCHRON14 <- countyratesfullcases$CHRON.CHRON14
preds.tr.CHRON14 <- predict(tr.CHRON14$finalModel, newdata = countyratesfullcases)
CHRON14 <- cbind.data.frame(preds.tr.CHRON14, realCHRON14)
CHRON14 <- CHRON14 %>% mutate(CHRONtreediff = ((preds.tr.CHRON14 -
    realCHRON14)/realCHRON14))
mortrate <- CHRON14$CHRONtreediff
cause <- rep("CHRON", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
CHRONtree <- cbind(fips, cause, model, mortrate)

treeCHRON14.mse <- mean((preds.tr.CHRON14 - realCHRON14)^2)
treeCHRON14.mse
```

```
## [1] 158.0894
```

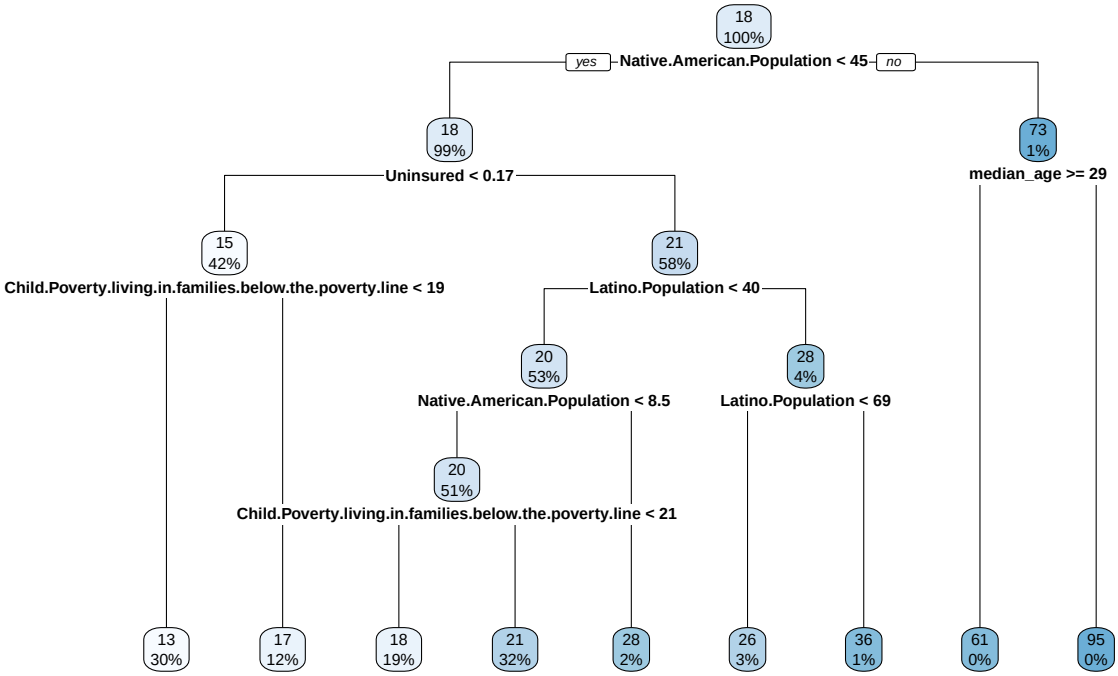
k-Nearest Neighbors mean square error

```
## [1] 189.6379
```

CIRR: Cirrhosis and other chronic liver diseases

Regression tree and mean square error

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```




```
realCIRR14 <- countyratesfullcases$CIRR.CIRR14
preds.tr.CIRR14 <- predict(tr.CIRR14$finalModel, newdata = countyratesfullcases)
CIRR14 <- cbind.data.frame(preds.tr.CIRR14, realCIRR14)
CIRR14 <- CIRR14 %>% mutate(CIRRtreediff = ((preds.tr.CIRR14 -
    realCIRR14)/realCIRR14))
mortrate <- CIRR14$CIRRtreediff
cause <- rep("CIRR", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
CIRRtree <- cbind(fips, cause, model, mortrate)

treeCIRR14.mse <- mean((preds.tr.CIRR14 - realCIRR14)^2)
treeCIRR14.mse
```

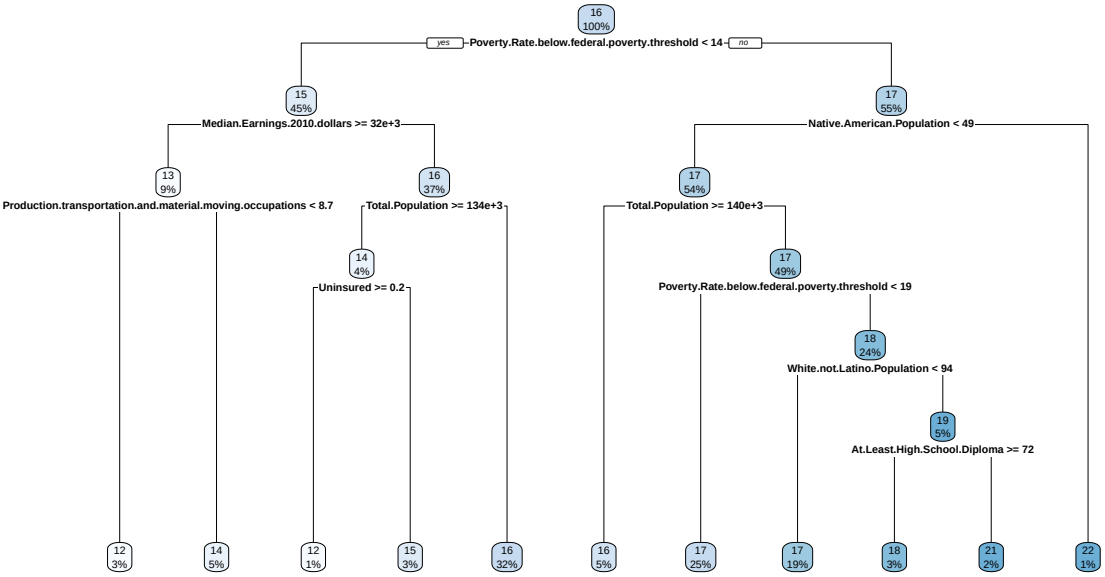
```
## [1] 23.49134
```

k-Nearest Neighbors mean square error

```
## [1] 39.78413
```

DIG: Digestive diseases

Regression tree and mean square error




```
realDIG14 <- countyratesfullcases$DIG.DIG14
preds.tr.DIG14 <- predict(tr.DIG14$finalModel, newdata = countyratesfullcases)
DIG14 <- cbind.data.frame(preds.tr.DIG14, realDIG14)
DIG14 <- DIG14 %>% mutate(DIGtreediff = ((preds.tr.DIG14 - realDIG14)/realDIG14))
mortrate <- DIG14$DIGtreediff
cause <- rep("DIG", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
DIGtree <- cbind(fips, cause, model, mortrate)

treeDIG14.mse <- mean((preds.tr.DIG14 - realDIG14)^2)
treeDIG14.mse
```

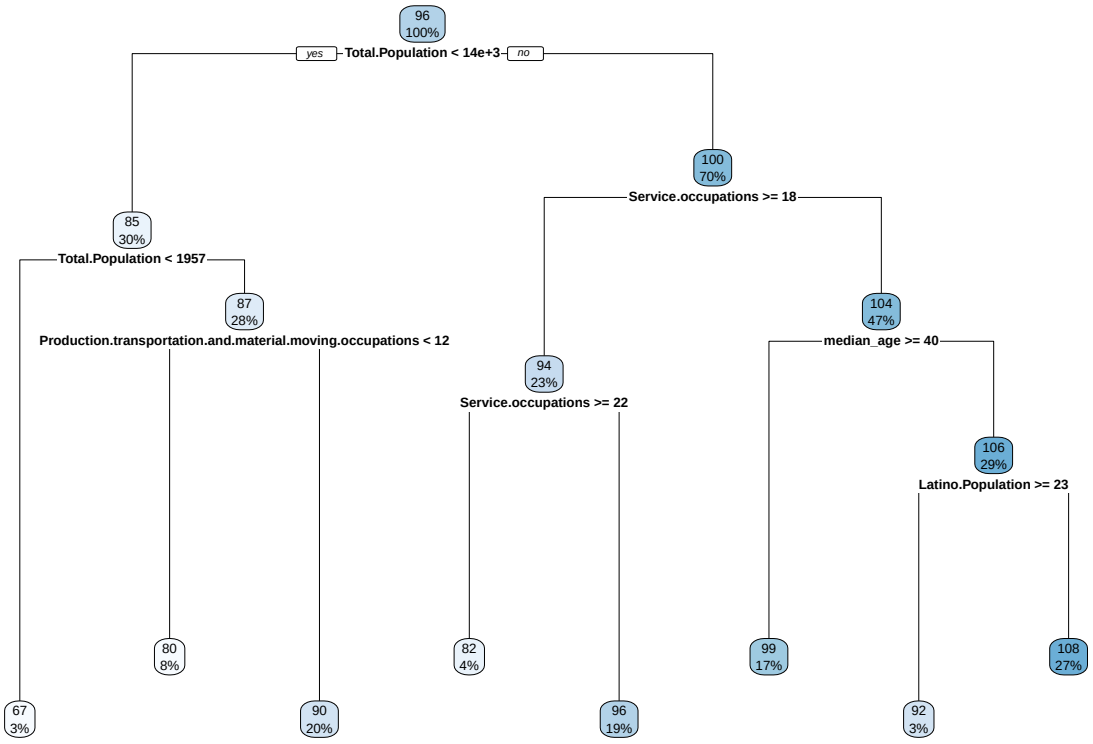
```
## [1] 4.024479
```

k-Nearest Neighbors mean square error

```
## [1] 3.739169
```

NEUR: Neurological disorders

Regression tree and mean square error




```
realNEUR14 <- countyratesfullcases$NEUR.NEUR14
preds.tr.NEUR14 <- predict(tr.NEUR14$finalModel, newdata = countyratesfullcases)
NEUR14 <- cbind.data.frame(preds.tr.NEUR14, realNEUR14)
NEUR14 <- NEUR14 %>% mutate(NEURtreediff = ((preds.tr.NEUR14 -
    realNEUR14)/realNEUR14))
mortrate <- NEUR14$NEURtreediff
cause <- rep("NEUR", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
NEURtree <- cbind(fips, cause, model, mortrate)

treeNEUR14.mse <- mean((preds.tr.NEUR14 - realNEUR14)^2)
treeNEUR14.mse
```

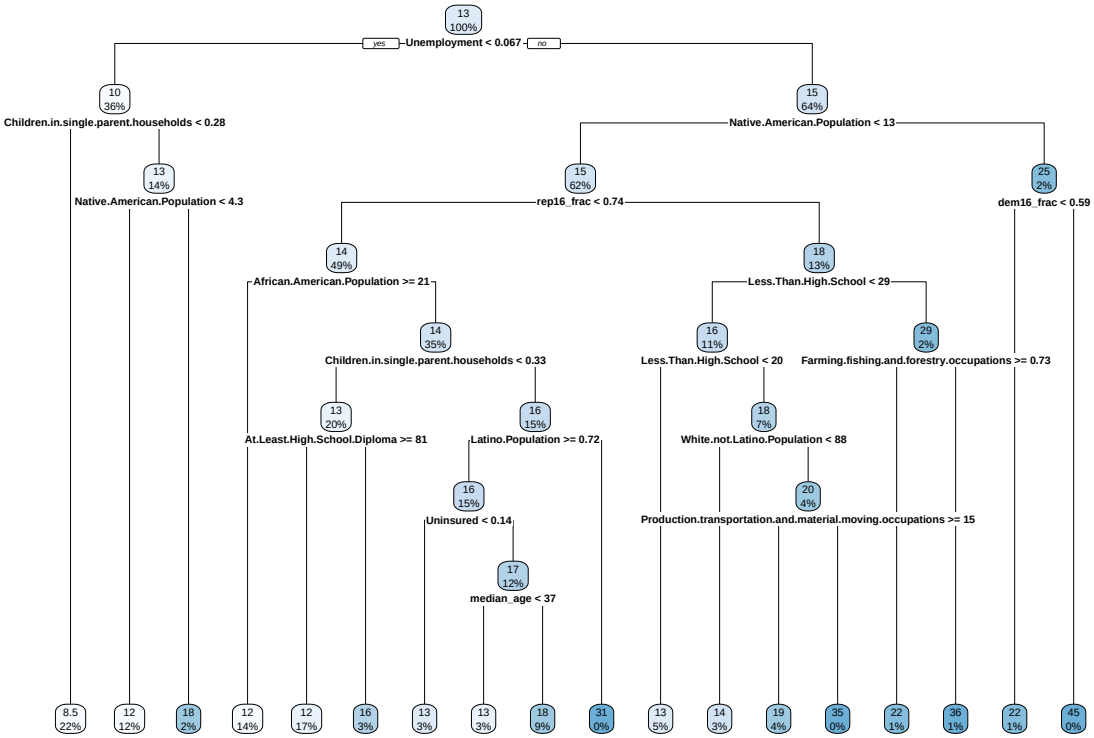
```
## [1] 376.5733
```

k-Nearest Neighbors mean square error

```
## [1] 320.2459
```

MENSUB: Mental and substance use disorders

Regression tree and mean square error




```
realMENSUB14 <- countyratesfullcases$MENSUB.MENSUB14
preds.tr.MENSUB14 <- predict(tr.MENSUB14$finalModel, newdata = countyratesfullcases)
MENSUB14 <- cbind.data.frame(preds.tr.MENSUB14, realMENSUB14)
MENSUB14 <- MENSUB14 %>% mutate(MENSUBtreediff = ((preds.tr.MENSUB14 -
    realMENSUB14)/realMENSUB14))
mortrate <- MENSUB14$MENSUBtreediff
cause <- rep("MENSUB", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
MENSUBtree <- cbind(fips, cause, model, mortrate)

treeMENSUB14.mse <- mean((preds.tr.MENSUB14 - realMENSUB14)^2)
treeMENSUB14.mse
```

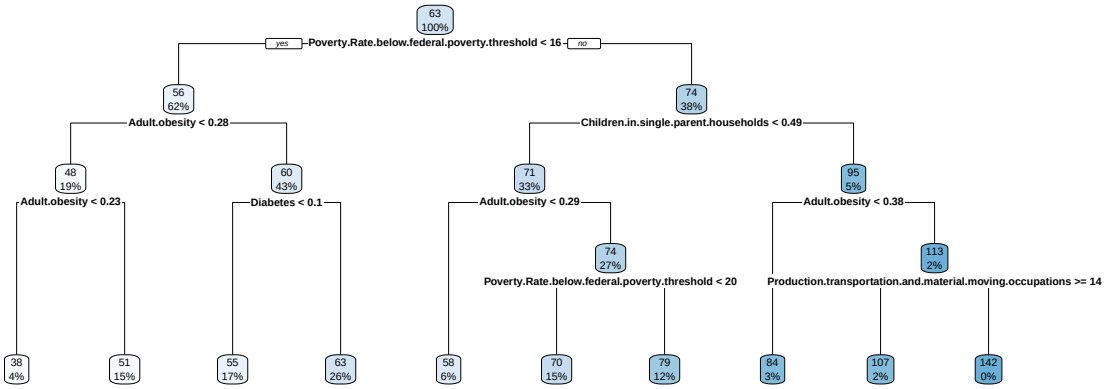
```
## [1] 24.66439
```

k-Nearest Neighbors mean square error

```
## [1] 31.55449
```

DIAB: Diabetes, urogenital, blood, and endocrine diseases

Regression tree and mean square error




```
realDIAB14 <- countyratesfullcases$DIAB.DIAB14
preds.tr.DIAB14 <- predict(tr.DIAB14$finalModel, newdata = countyratesfullcases)
DIAB14 <- cbind.data.frame(preds.tr.DIAB14, realDIAB14)
DIAB14 <- HIV14 %>% mutate(DIABtreediff = ((preds.tr.DIAB14 -
    realDIAB14)/realDIAB14))
mortrate <- DIAB14$HIVtreediff
cause <- rep("DIAB", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
DIABtree <- cbind(fips, cause, model, mortrate)

treeDIAB14.mse <- mean((preds.tr.DIAB14 - realDIAB14)^2)
treeDIAB14.mse
```

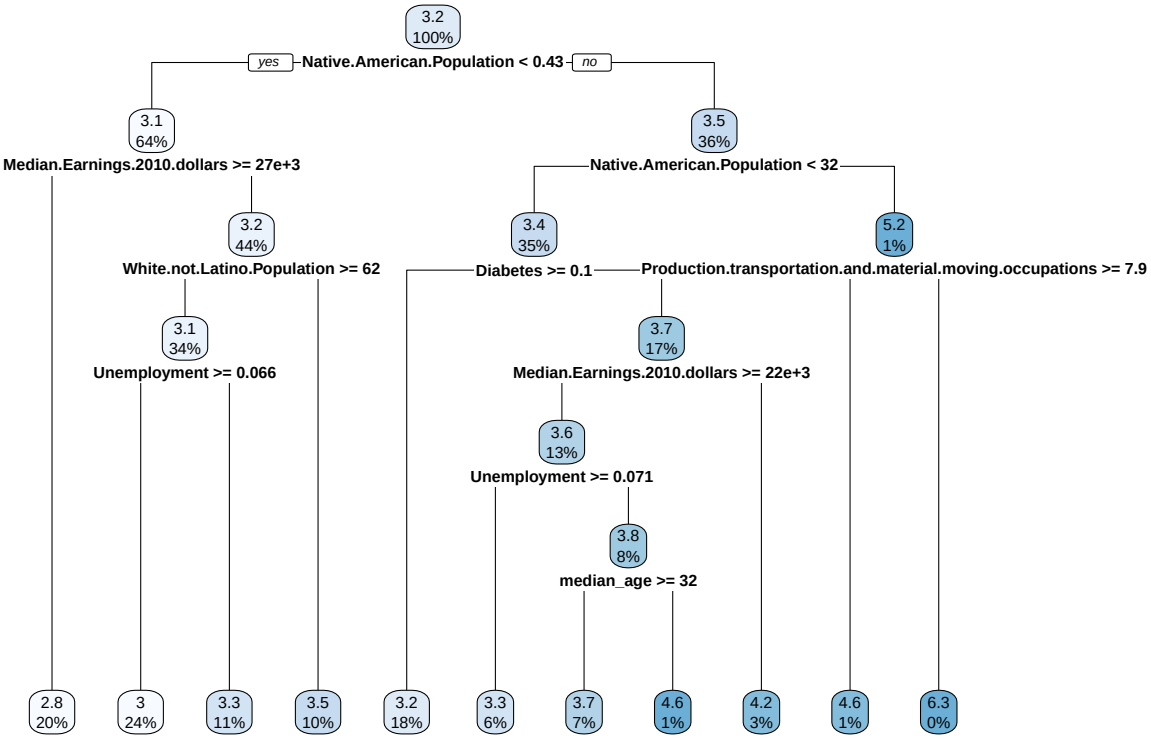
```
## [1] 157.4018
```

k-Nearest Neighbors mean square error

```
## [1] 212.7531
```

MUSC: Musculoskeletal disorders

Regression tree and mean square error




```
realMUSC14 <- countyratesfullcases$MUSC.MUSC14
preds.tr.MUSC14 <- predict(tr.MUSC14$finalModel, newdata = countyratesfullcases)
MUSC14 <- cbind.data.frame(preds.tr.MUSC14, realMUSC14)
MUSC14 <- HIV14 %>% mutate(MUSCtreediff = ((preds.tr.MUSC14 -
    realMUSC14)/realMUSC14))
mortrate <- MUSC14$MUSCtreediff
cause <- rep("MUSC", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
MUSCtree <- cbind(fips, cause, model, mortrate)

treeMUSC14.mse <- mean((preds.tr.MUSC14 - realMUSC14)^2)
treeMUSC14.mse
```

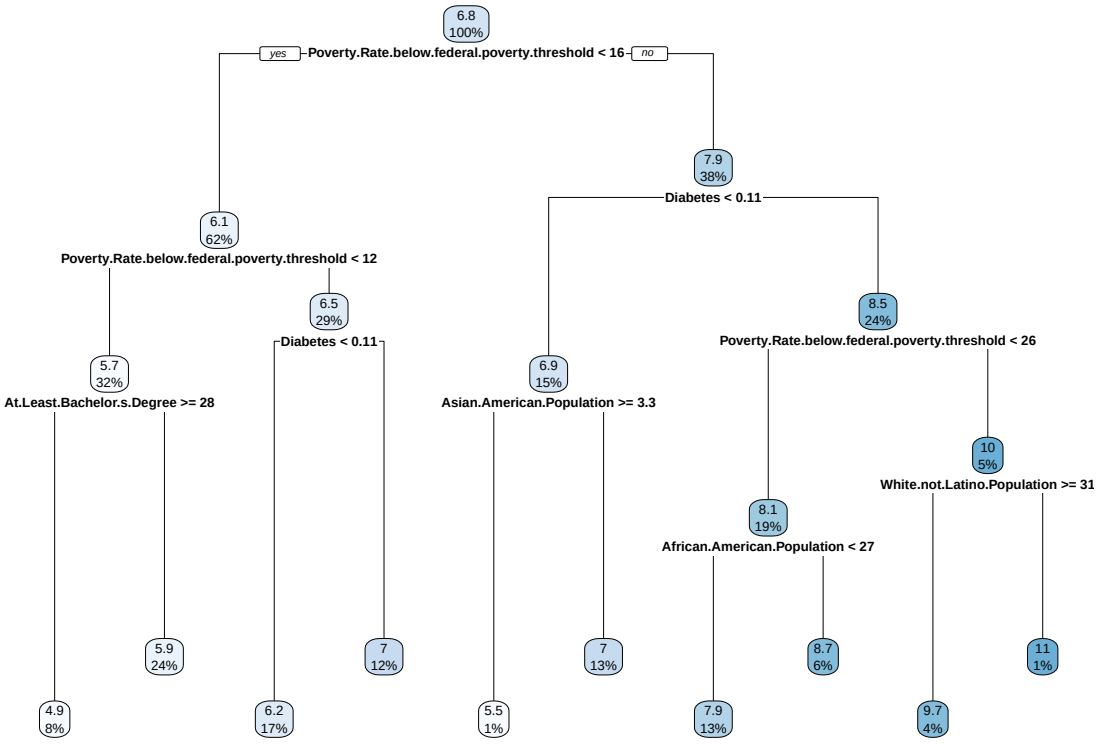
```
## [1] 0.4045811
```

k-Nearest Neighbors mean square error

```
## [1] 0.4037828
```

OTHN: Other non-communicable diseases

Regression tree and mean square error




```
realOTHN14 <- countyratesfullcases$OTHN.OTHN14
preds.tr.OTHN14 <- predict(tr.OTHN14$finalModel, newdata = countyratesfullcases)
OTHN14 <- cbind.data.frame(preds.tr.OTHN14, realOTHN14)
OTHN14 <- OTHN14 %>% mutate(OTHNtreediff = ((preds.tr.OTHN14 -
    realOTHN14)/realOTHN14))
mortrate <- OTHN14$OTHNtreediff
cause <- rep("OTHN", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
OTHNtree <- cbind(fips, cause, model, mortrate)

treeOTHN14.mse <- mean((preds.tr.OTHN14 - realOTHN14)^2)
treeOTHN14.mse
```

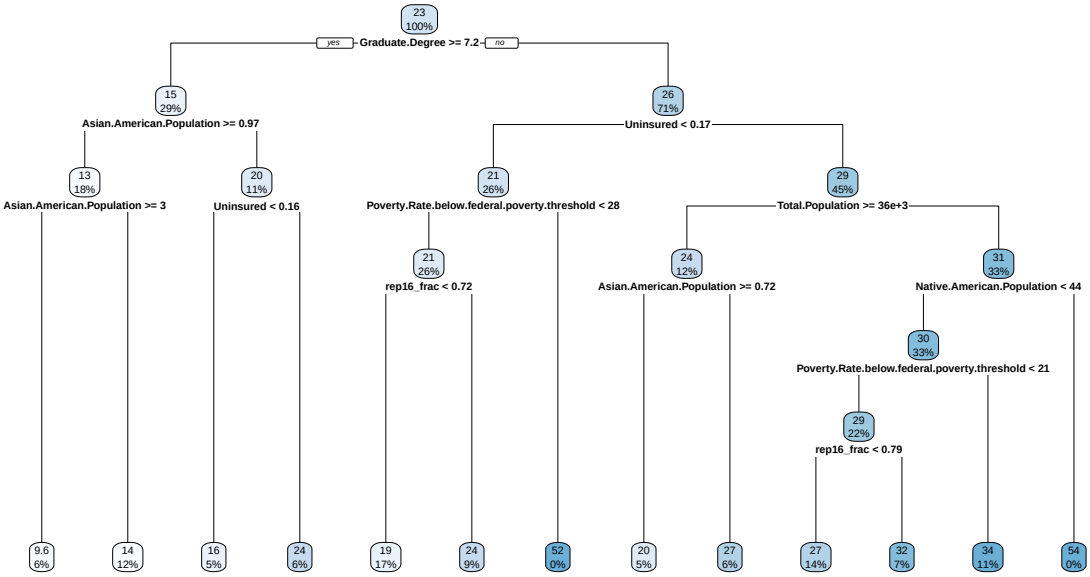
```
## [1] 0.7641577
```

k-Nearest Neighbors mean square error

```
## [1] 1.235623
```

TRAN: Transport injuries

Regression tree and mean square error




```
realTRAN14 <- countyratesfullcases$TRAN.TRAN14
preds.tr.TRAN14 <- predict(tr.TRAN14$finalModel, newdata = countyratesfullcases)
TRAN14 <- cbind.data.frame(preds.tr.TRAN14, realTRAN14)
TRAN14 <- TRAN14 %>% mutate(TRANtreediff = ((preds.tr.TRAN14 -
    realTRAN14)/realTRAN14))
mortrate <- TRAN14$TRANtreediff
cause <- rep("TRAN", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
TRANtree <- cbind(fips, cause, model, mortrate)

treeTRAN14.mse <- mean((preds.tr.TRAN14 - realTRAN14)^2)
treeTRAN14.mse
```

```
## [1] 31.85381
```

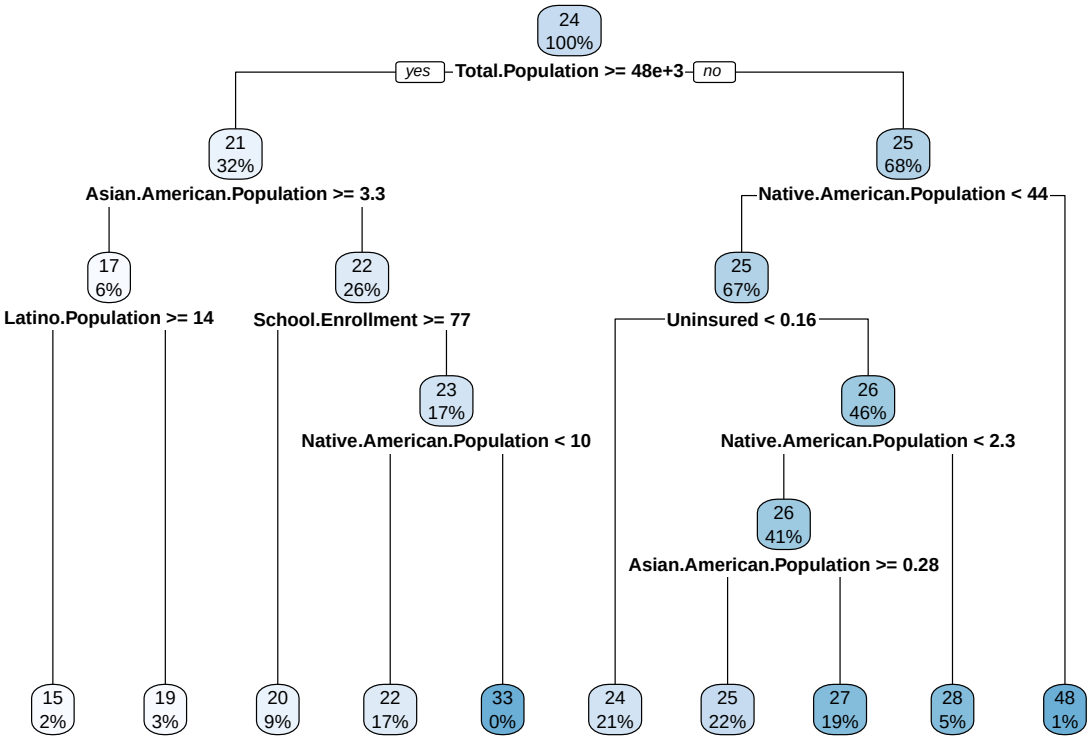
k-Nearest Neighbors mean square error

```
## [1] 37.92881
```

UNIN: Unintentional injuries

Regression tree and mean square error

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```




```
realUNIN14 <- countyratesfullcases$UNIN.UNIN14
preds.tr.UNIN14 <- predict(tr.UNIN14$finalModel, newdata = countyratesfullcases)
UNIN14 <- cbind.data.frame(preds.tr.UNIN14, realUNIN14)
UNIN14 <- UNIN14 %>% mutate(UNINtreediff = ((preds.tr.UNIN14 -
    realUNIN14)/realUNIN14))
mortrate <- UNIN14$UNINtreediff
cause <- rep("UNIN", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
UNINtree <- cbind(fips, cause, model, mortrate)

treeUNIN14.mse <- mean((preds.tr.UNIN14 - realUNIN14)^2)
treeUNIN14.mse
```

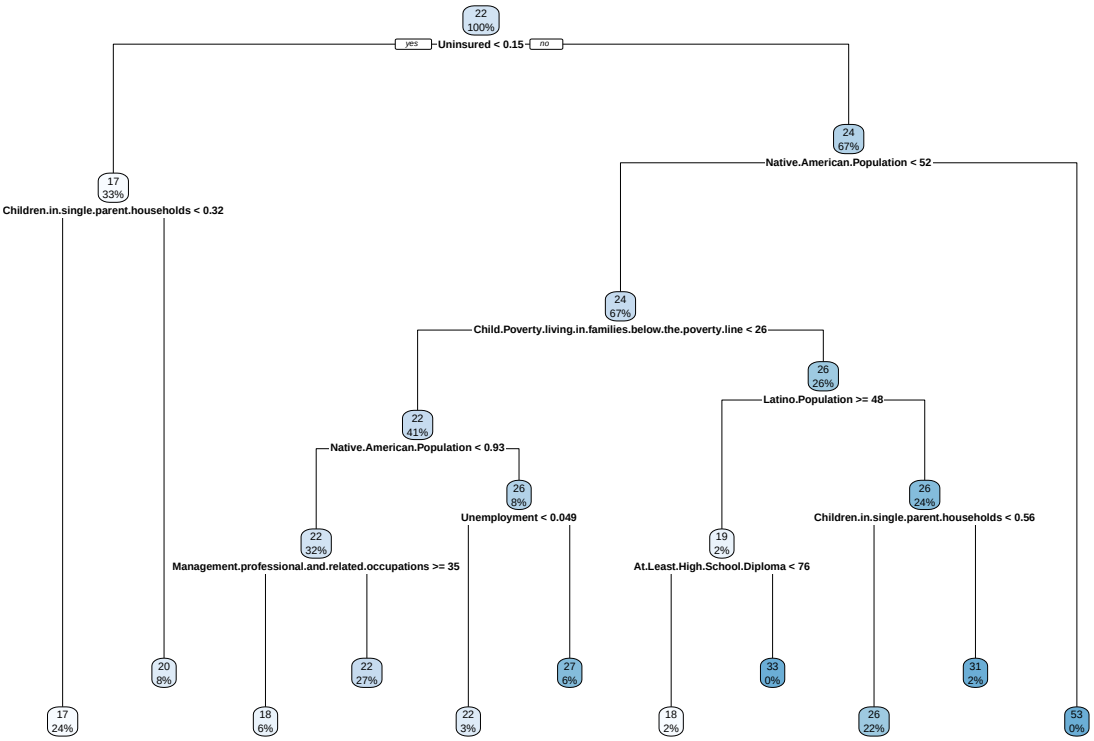
```
## [1] 14.23563
```

k-Nearest Neighbors mean square error

```
## [1] 13.94861
```

SELF: Self-harm and interpersonal violence

Regression tree and mean square error




```
realSELF14 <- countyratesfullcases$SELF.SELF14
preds.tr.SELF14 <- predict(tr.SELF14$finalModel, newdata = countyratesfullcases)
SELF14 <- cbind.data.frame(preds.tr.SELF14, realSELF14)
SELF14 <- SELF14 %>% mutate(SELFtreediff = ((preds.tr.SELF14 -
    realSELF14)/realSELF14))
mortrate <- SELF14$SELFtreediff
cause <- rep("SELF", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
SELFtree <- cbind(fips, cause, model, mortrate)

treeSELF14.mse <- mean((preds.tr.SELF14 - realSELF14)^2)
treeSELF14.mse
```

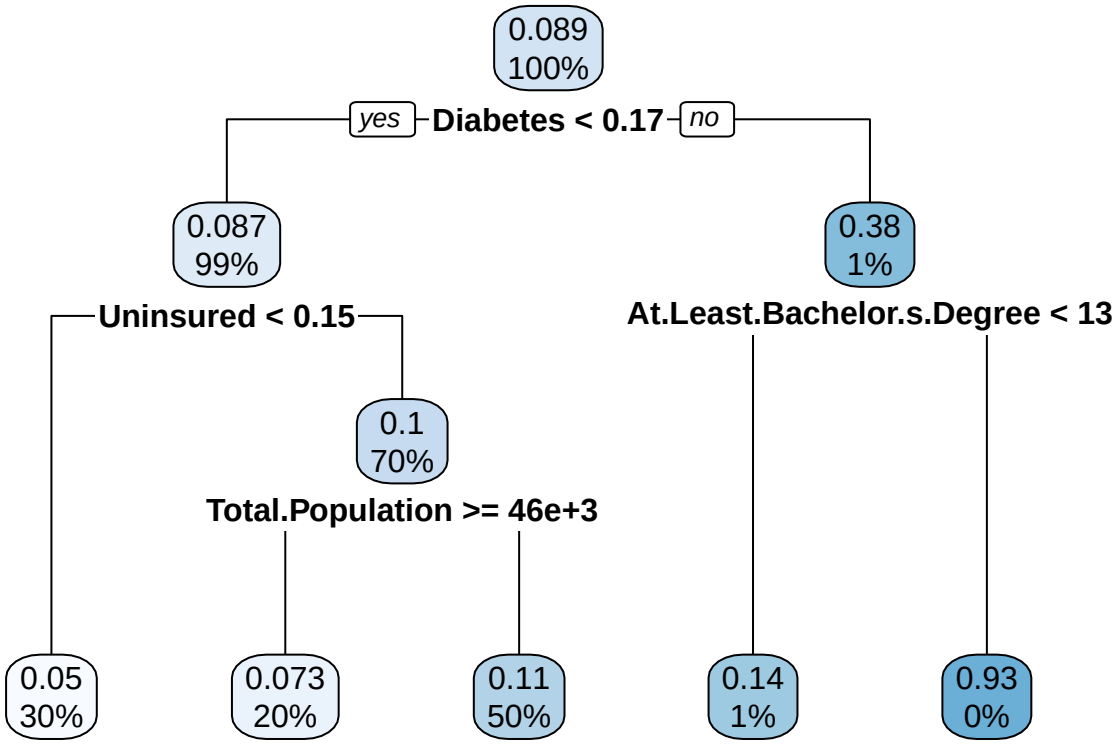
```
## [1] 25.32089
```

k-Nearest Neighbors mean square error

```
## [1] 29.88253
```

WAR: Forces of nature, war, and legal intervention

Regression tree and mean square error




```
realWAR14 <- countyratesfullcases$WAR.WAR14
preds.tr.WAR14 <- predict(tr.WAR14$finalModel, newdata = countyratesfullcases)
WAR14 <- cbind.data.frame(preds.tr.WAR14, realWAR14)
WAR14 <- WAR14 %>% mutate(WARtreediff = ((preds.tr.WAR14 - realWAR14)/realWAR14))
mortrate <- WAR14$WARtreediff
cause <- rep("WAR", 3110)
cause <- as.data.frame(cause)
model <- rep("tree", 3110)
model <- as.data.frame(model)
WARtree <- cbind(fips, cause, model, mortrate)

treeWAR14.mse <- mean((preds.tr.WAR14 - realWAR14)^2)
treeWAR14.mse
```

```
## [1] 0.01477997
```

k-Nearest Neighbors mean square error

```
## [1] 0.01402043
```

Shiny Visualization

The first important step to building our shiny application was to clean the data in a way that would be "function friendly." While researching Shiny and finding ways to build interactive maps of US county data, we discovered a package called choroplethr. From this package we utilize a function called county_choropleth() which builds a choropleth map of the counties in the United States. However, count-choropleth() takes a datafram with specific column names: region and value. Region must be assigned to the standard fips codes for the United States counties without any leading zeros. Value is then the information that will be projected onto the map. By setting number of colors to 1 in this instance we are able to create a continuous scale for

our variable of interest - mortality rates. However, this particular function that county-choropleth() takes means that we need to manipulate our data set in several ways. This required us to rename our fips column to region and our mortality column to value. Most importantly we needed to filter the data so that the values used when rendering our model / this specific instance of our map could be selected by the user. To do this we created two drop down menus in

Limitations