Predicting Heart Disease Mortality

David Edelman

April 2, 2019

Executive Summary

Heart disease is one of the leading causes of death in the United States. This document presents an analysis of county-by-county heart disease mortality rates (per 100,000 individuals) and uses various demographic statistics to build a model that predicts the heart disease mortality rate for that county

The given data set, containing known heart disease mortality rates, comprises 3198 observations, representing 1599 counties and data collected over two separate years. Both categorical and numeric data are present in the feature set, and after some initial data exploration, several other features were calculated. A predictive regression model was then created using select features in order to predict the heart disease mortality rate.

While building and tuning the predictive model, the following features were deemed significant to the predictive power of the model:

Area Information

- Area RUCC (Rural-Urban Continuum Codes) – classifies each county into one of 9 mutually exclusive categories that identifies (a) the population size, (b) the degree of urbanization and (c) proximity to a metropolitan area.
  - While the RUCC itself did now show any strong correlation to heart disease mortality rate, by combining one or more RUCC values into one of 5 groups based solely on population size, the data showed 3 population groups that skewed below the mean heart disease mortality rate and 1 that skewed above the mean.

Economic Factors

- Economic Typology – classifies each county into one of 6 mutually exclusive categories of economic dependence; two categories show a distribution skewed below the mean heart disease mortality rate and two categories are skewed above (source: USDA Economic Research Service).
- Percent of Civilian Labor – the annualized percent of the county's population classified as civilian; analysis shows a moderate negative correlation to heart disease mortality rate (source: Bureau of Labor Statistics)

Demographics

- Percent of non-Hispanic African Americans – analysis shows the strongest positive correlation among the 5 race/ethnicity groups (source: US Census Population Estimates)
- Percent of adults with less than a high-school diploma; percent of adults with a Bachelor's degree (or higher) – the former shows a strong positive correlation, while the latter shows an equally strong negative correlation (source: US Census Population Estimates)
- Death Rate per 1000 individuals - analysis shows a high-moderate correlation (source: US Census Population Estimates)

Health Factors

- Air Pollution Particulate Matter – measured in µg/m3; the average concentration of fine particulate matter as measured over the year. Lower concentrations skewed below the population mean while higher concentrations skewed above (source: CDC WONDER).
- Percent of physical inactivity – percent of adult population that self-identifies as physically inactive (source: National Center for Chronic Disease Prevention and Health Promotion); physical inactivity showed a moderately strong correlation to heart disease mortality rate
- Percent of adult obesity; percent of diabetes – each of the two factors individually showed a strong correlation with heart disease mortality rate, but with a thought experiment and some data transformation into combined factors, an even stronger correlation was found (sources NCCDPHP; NCCDPHP, Division of Diabetes Translation)

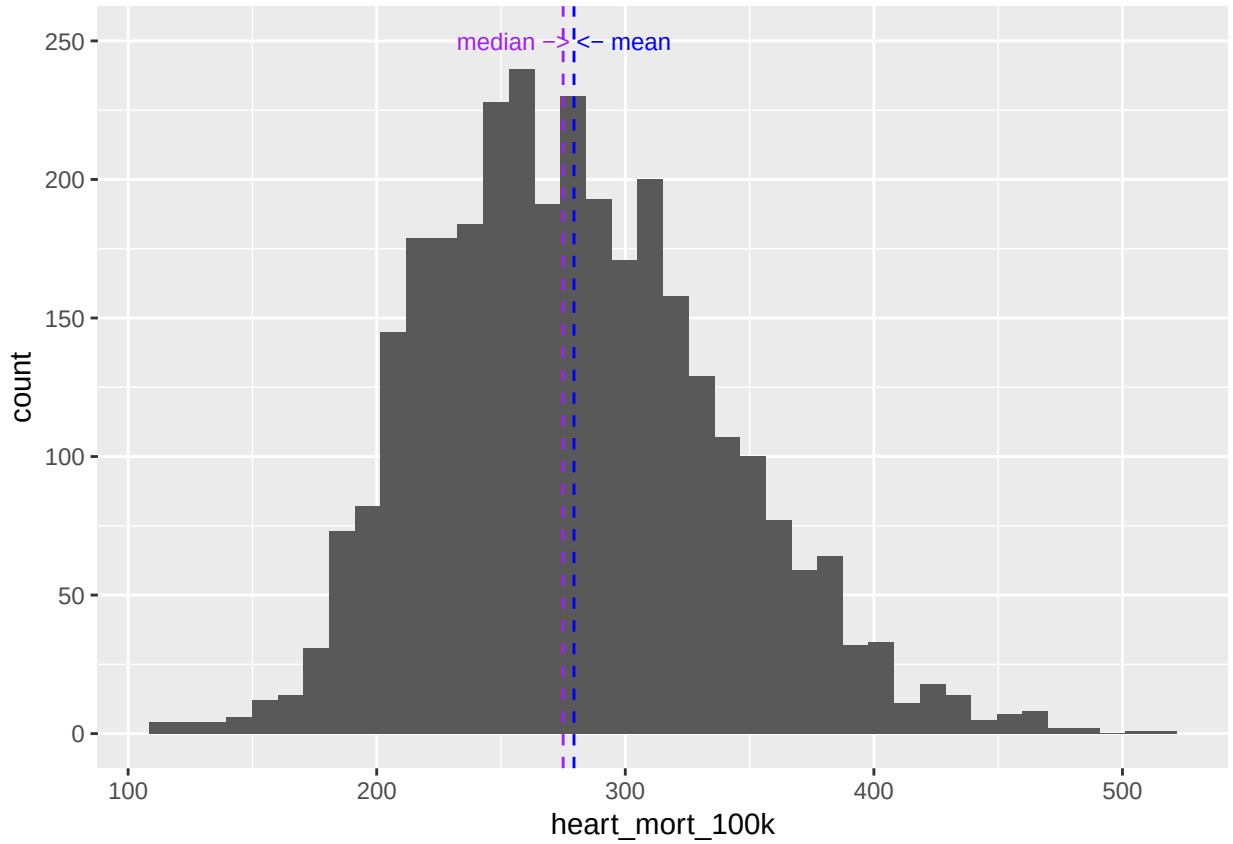
Data Exploration and Initial Analysis

Heart Disease Mortality Rate

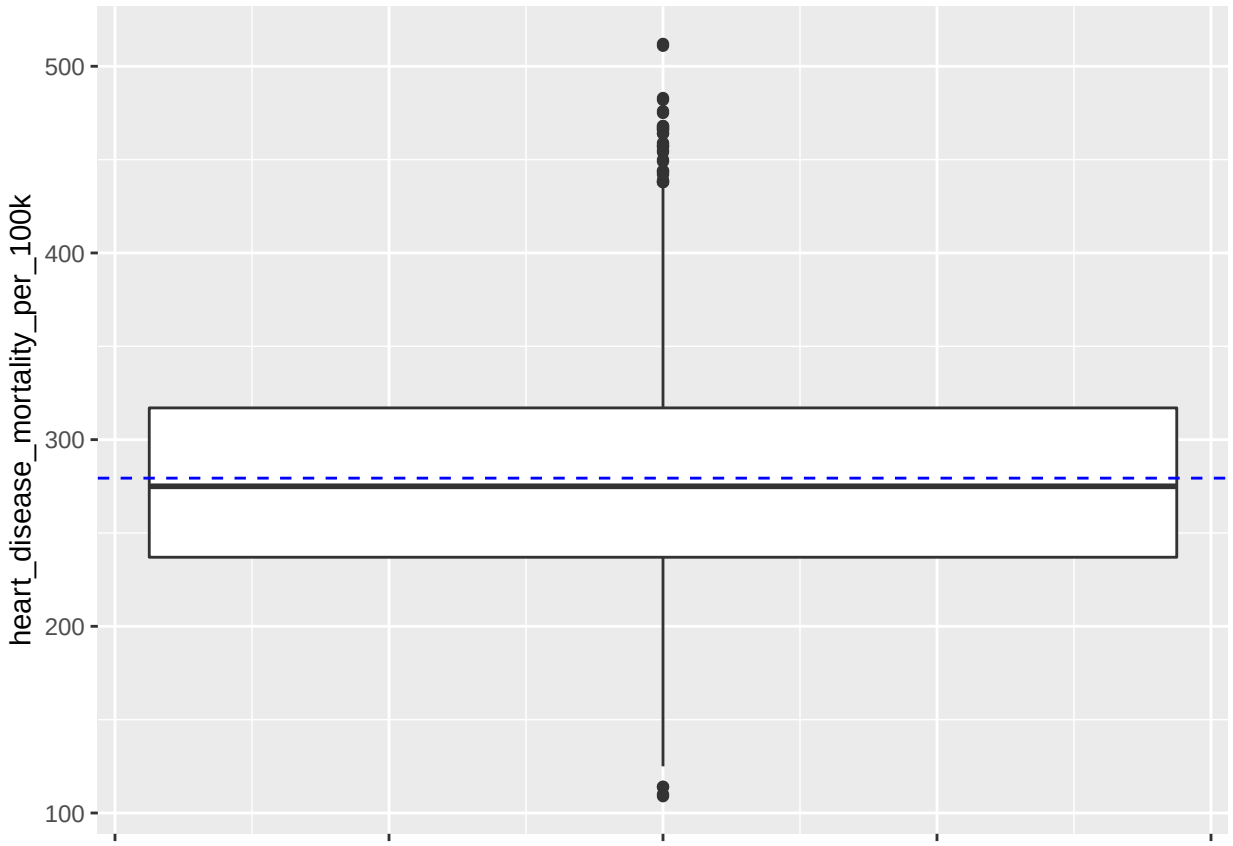
Descriptive Statistics

Heart Disease Mortality Rate (herein shown with a unit of "deaths per 100,000 population") in the given data set showed a mean of 279.4, median of 275.0, and standard deviation of 59.0 with a range of 109.0 – 512.0, across 3198 observations. A histogram with 40 equal bins (calculated based on the minimum and maximum values) and a box plot both show that heart disease mortality rate is approximately normal (skewed very slightly positive), with only approximately 10-15 outliers (values outside of the Inner Quartile Range of 237 – 317).

```
#Histogram & boxplot
train_data %>% ggplot(aes(x=heart_disease_mortality_per_100k))+geom_histogram(bins=40)+
  xlab("heart_mort_100k")+
  geom_vline(linetype = "dashed", color = "purple", xintercept = hd_med)+
  geom_vline(linetype = "dashed", color = "blue", xintercept = hd_mean)+
  annotate("text", x=hd_mean+20, y=250, label="<- mean", size=3, color = "blue")+
  annotate("text", x=hd_med-20, y=250, label="median ->", size=3, color = "purple")
```




```
train_data %>% ggplot(aes(y=heart_disease_mortality_per_100k))+geom_boxplot()+
  geom_hline(linetype = "dashed", color = "blue", yintercept = hd_mean)+
  theme(axis.text.x=element_blank())
```



Initial Analysis of Predictors

Predictor Types

The data set includes 32 values for each rows (ignoring "row_id"), 29 of which are numeric

```
#Numeric columns
num_cols <- train_data %>%
  select(-c(row_id, heart_disease_mortality_per_100k)) %>%
  .[sapply(., is.numeric)] %>%
  colnames()

#Categorical columns
cat_cols <- colnames(train_data)[sapply(train_data, is.factor)]

length(num_cols)
```

```
## [1] 29
```

```
length(cat_cols)
```

```
## [1] 4
```

(the column "yr" will also be ignored as it has 2 values, "a" and "b", which are merely placeholders)

Missing feature values

In a data set of this size, missing data can adversely affect the predictive value of individual features, so we look for any features that have a significant percentage of missing data

```
#What percentage of rows have NA values for all features?
NAs <- data.frame(
  features = colnames(train_data %>% select(-heart_disease_mortality_per_100k,
                                            -yr,
                                            -row_id)),
  NA_pct =
      as.numeric(format(sapply(train_data %>%
                      select(-heart_disease_mortality_per_100k,
                             -yr, -row_id),
             function(x){mean(ifelse(is.na(x), 1, 0))},
             simplify=TRUE)*100, digits=3) )
)

#Show NA percentage from highest to lowest
NAs %>% arrange(desc(NA_pct))
```

```
##                                          features  NA_pct
## 1                       health__homicides_per_100k 61.5072
## 2                     health__pct_excessive_drinking 30.5816
## 3                          health__pct_adult_smoking 14.5091
## 4       health__motor_vehicle_crash_deaths_per_100k 13.0394
## 5                            health__pop_per_dentist  7.6298
## 6             health__pop_per_primary_care_physician  7.1920
## 7                          health__pct_low_birthweight  5.6911
## 8           health__air_pollution_particulate_matter  0.8755
## 9                           econ__pct_uninsured_adults  0.0625
## 10                        econ__pct_uninsured_children  0.0625
## 11                                 demo__pct_female  0.0625
## 12                  demo__pct_below_18_years_of_age  0.0625
## 13                demo__pct_aged_65_years_and_older  0.0625
## 14                                 demo__pct_hispanic  0.0625
## 15          demo__pct_non_hispanic_african_american  0.0625
## 16                       demo__pct_non_hispanic_white  0.0625
## 17      demo__pct_american_indian_or_alaskan_native  0.0625
## 18                                   demo__pct_asian  0.0625
## 19                          health__pct_adult_obesity  0.0625
## 20                              health__pct_diabetes  0.0625
## 21                  health__pct_physical_inactivity  0.0625
## 22                                       area__rucc  0.0000
## 23                             area__urban_influence  0.0000
## 24                           econ__economic_typology  0.0000
## 25                           econ__pct_civilian_labor  0.0000
## 26                             econ__pct_unemployment  0.0000
## 27 demo__pct_adults_less_than_a_high_school_diploma  0.0000
## 28        demo__pct_adults_with_high_school_diploma  0.0000
## 29               demo__pct_adults_with_some_college  0.0000
## 30             demo__pct_adults_bachelors_or_higher  0.0000
## 31                           demo__birth_rate_per_1k  0.0000
## 32                           demo__death_rate_per_1k  0.0000
```

We will remove predictors from the data set that have more than 10% missing values. This turns out to be

- health__homicides_per_100k

- health___pct_excessive_drinking
- health___pct_adult_smoking
- health___motor_vehicle_crash_deaths_per_100k

```
#Remove high NA features
high_NA <- NAs %>%
  filter(NA_pct > 10) %>%
  .$features %>%
  as.character()

train_data <- train_data %>% select(-high_NA)
```

12 predictors have 0.0625% missing values, which equates to 2 rows. If it is the same two rows that are missing all of the predictors, we will just remove those rows from the data set.

```
#Which features have minimum number of NAs (> 0)
min_missing <- min(NAs %>% filter(NA_pct > 0) %>% select(NA_pct))
NA2 <- sapply(train_data %>%
        select(as.vector(NAs$features[NAs$NA_pct == min_missing])),
       function(x){which(is.na(x))}, simplify=TRUE)
t(NA2) %>% grid.table()
```

econ__pct_uninsured_adults	1090	1251
econ__pct_uninsured_children	1090	1251
demo__pct_female	1090	1251
demo__pct_below_18_years_of_age	1090	1251
demo__pct_aged_65_years_and_older	1090	1251
demo__pct_hispanic	1090	1251
demo__pct_non_hispanic_african_american	1090	1251
demo__pct_non_hispanic_white	1090	1251
demo__pct_american_indian_or_alaskan_native	1090	1251
demo__pct_asian	1090	1251
health__pct_adult_obesity	1090	1251
health__pct_diabetes	1090	1251
health__pct_physical_inactivity	1090	1251

We see that index 1090 and 1251 are missing all of the selected predictors, so we will remove them from the data set

```
#Remove the rows with 12 NA predictors
train_data <- train_data[-t(NA2)[1,c(1:2)], ]
```

For the remaining NA values, we will decide how to handle after some exploratory data analysis of the features.

Distributions Across Categorical Features

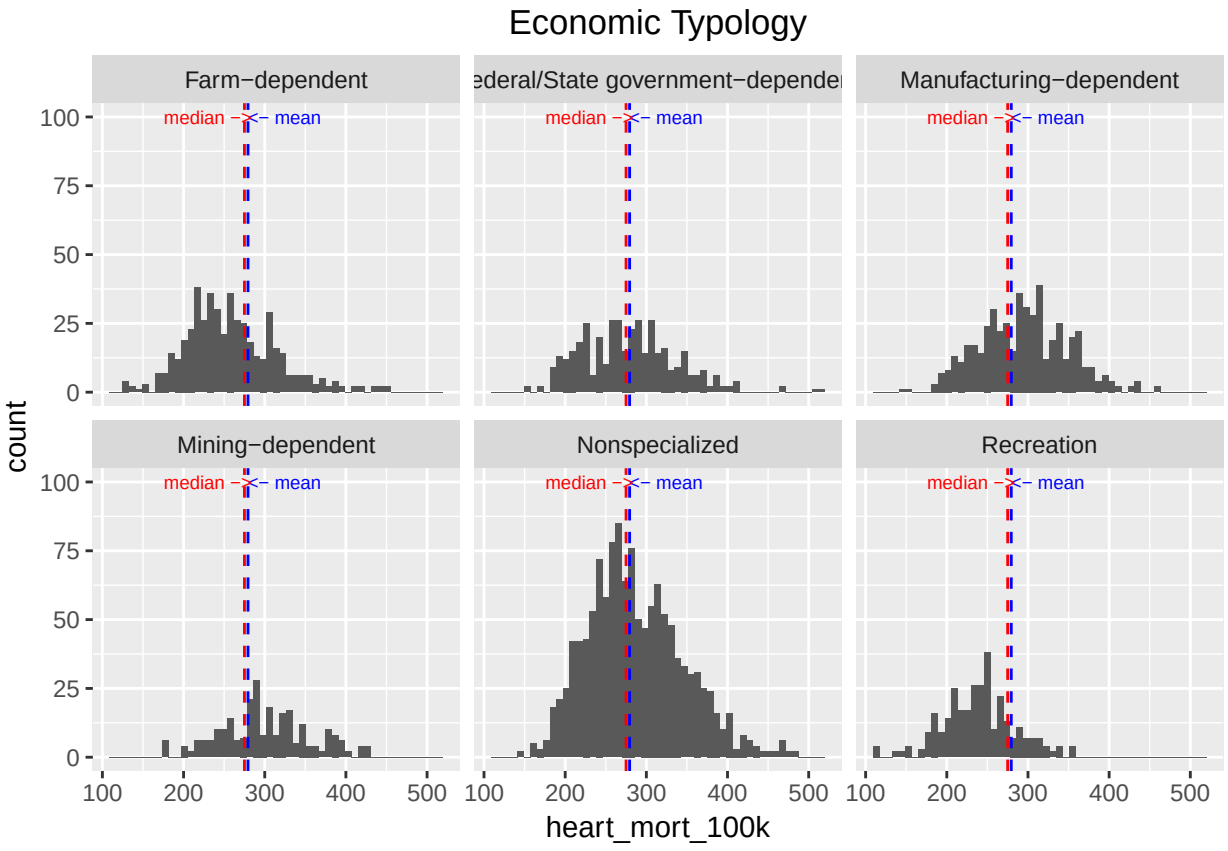
Economic Typology

The 6 economic typologies used by the USDA Economic Research Service to assign to each county are

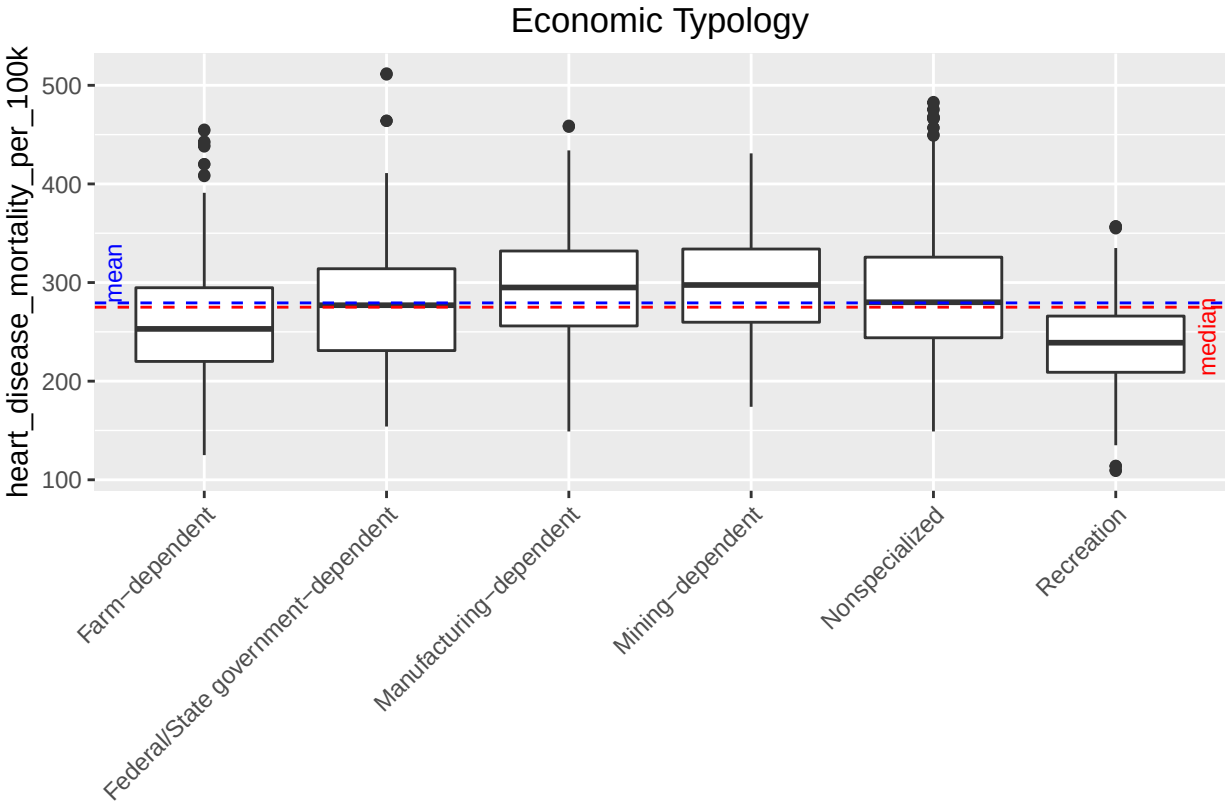
- Farm-dependent
- Federal/State government-dependent
- Manufacturing-dependent
- Mining-dependent
- Non-specialized
- Recreation

The typologies speak to the main type of industries of a particular county. So while we don't have the specific industries in each county (knowing, intellectually, that certain industries have been linked in the past to specific conditions, including heart disease), we can use the typology as a good representative proxy for specific industries. To identify any potential relationship between typology and heart disease mortality, we use histograms and box plots to examine each typology for an abnormal or skewed distribution.

```
#Histograms for economic type
train_data %>% ggplot(aes(x=heart_disease_mortality_per_100k))+
               xlab("heart_mort_100k")+
               geom_histogram(binwidth = (high_mort-low_mort)/50)+
               facet_wrap(~ econ__economic_typology, ncol = 3)+
               geom_vline(linetype="dashed", xintercept = hd_mean, color = "blue")+
               geom_vline(linetype="dashed", xintercept = hd_med, color = "red")+
               ggtitle("Economic Typology")+theme(plot.title = element_text(hjust=0.5))+
               annotate("text", x=hd_mean+45, y=100,
                        label="<- mean", size=2.5, color = "blue")+
               annotate("text", x=hd_med-45, y=100,
                        label="median ->", size=2.5, color = "red")
```



```r
#Box plot by economic type
train_data %>% ggplot(aes(x=econ__economic_typology, y=heart_disease_mortality_per_100k))+
        geom_boxplot() +
        geom_hline(linetype="dashed", yintercept = hd_mean, color = "blue")+xlab("")+
        geom_hline(linetype="dashed", yintercept = hd_med, color = "red")+
        theme(axis.text.x = element_text(angle = 45, hjust=1, vjust=1)) +
        ggtitle("Economic Typology")+theme(plot.title = element_text(hjust=0.5))+
        annotate("text", x=.5, y=hd_mean+30,
                 label="mean", size=3, color = "blue", angle=90)+
        annotate("text", x=6.5, y=hd_med-30,
                 label="median", size=3, color = "red", angle=90)
```



The histograms show that all of the typologies appear to be approximately normal. They also show that the mean from Recreation and from Farm-dependent (to a lesser degree) to be lower than the population mean. None of the typologies appear to have significantly higher means than the population average. However, the box plots give a better picture of the distributions, showing that indeed Recreation and Farm-Dependent have their median value below the population average, while Manufacturing-dependent and Mining-dependent have medians well above the population average (Non-specialized shows a very slight variation from the population mean and median). The same results are shown below:

```r
#Show category mean/median and comparison to population
bind_rows(train_data %>% select(heart_disease_mortality_per_100k,
                      econ__economic_typology) %>%
  group_by(econ__economic_typology) %>%
  summarize(mn = mean(heart_disease_mortality_per_100k),
            md = median(heart_disease_mortality_per_100k)) %>%
  mutate(MeanAboveBelow = ifelse(round(mn,1) < round(hd_mean,1),
                                 "Below",
                                ifelse(round(mn,1) == round(hd_mean,1),
```

```
                              "Equal", "Above")),
        MedianAboveBelow = ifelse(round(md,1) < round(hd_med,1),
                                  "Below",
                                 ifelse(round(md,1) == round(hd_med,1),
                                  "Equal", "Above")))%>%
  select(econ__economic_typology, mn,
         MeanAboveBelow, md, MedianAboveBelow),
  data.frame(econ__economic_typology = "Total Population Values",
             mn= hd_mean,
             MeanAboveBelow = "_",
             md = hd_med,
             MedianAboveBelow = "_")
)
```

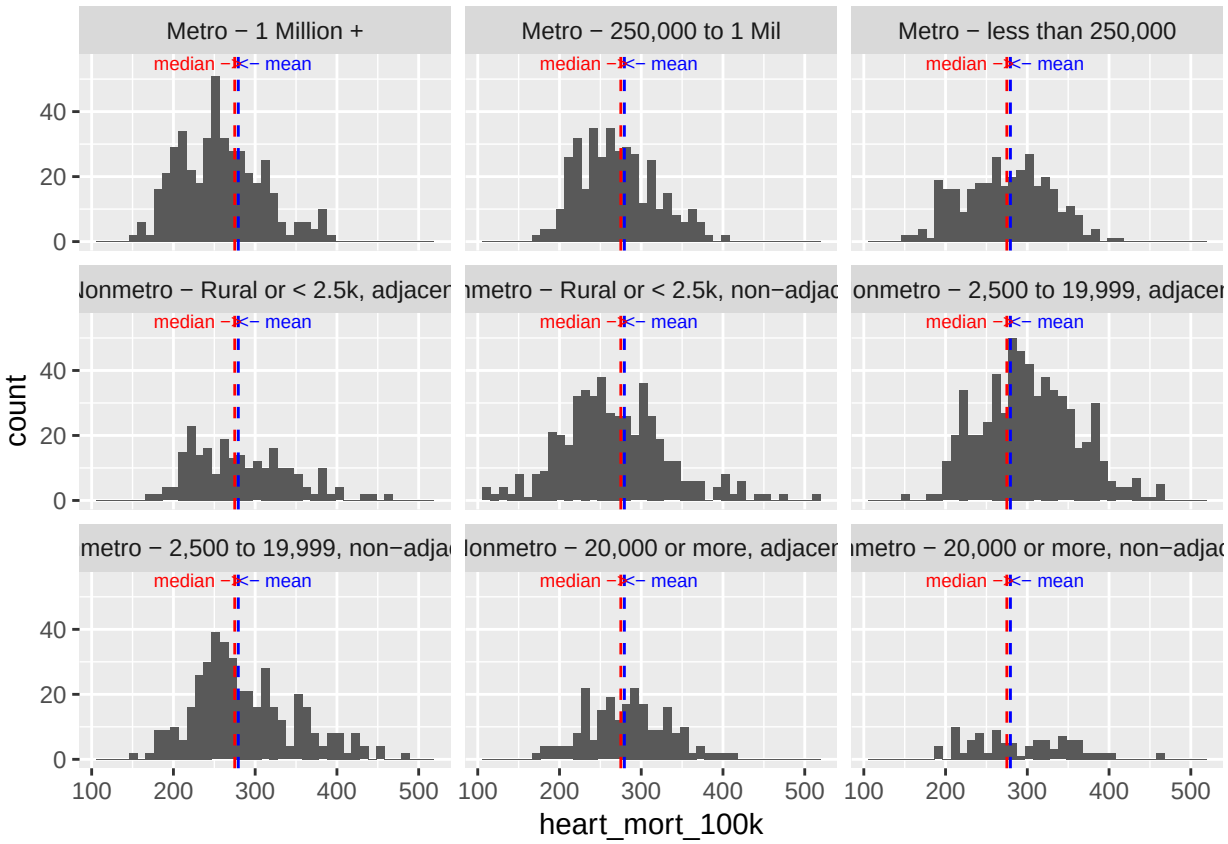
```
## # A tibble: 7 x 5
##   econ__economic_typology          mn MeanAboveBelow    md MedianAboveBelow
##   <chr>                         <dbl> <chr>          <dbl> <chr>
## 1 Farm-dependent                 259. Below           253  Below
## 2 Federal/State government-dep~  278. Below           277  Above
## 3 Manufacturing-dependent        295. Above           295  Above
## 4 Mining-dependent               301. Above           298. Above
## 5 Nonspecialized                 287. Above           280  Above
## 6 Recreation                     239. Below           239  Below
## 7 Total Population Values        279. _               275  _
```

Due to heart disease mortality rates being distributed differently across the economic typologies, this feature is a good candidate for use in a predictive model.

Population Spread

The individual values for "Area_RUCC" have populations ranging from < 100 to > 600, and histograms show that many distributions do not appear to be approximately normal.

```
#Rename original levels to show on plots
levels(train_data$area__rucc) <- c("Metro - 1 Million +",
                        "Metro - 250,000 to 1 Mil",
                        "Metro - less than 250,000",
                        "Nonmetro - Rural or < 2.5k, adjacent",
                        "Nonmetro - Rural or < 2.5k, non-adjacent",
                        "Nonmetro - 2,500 to 19,999, adjacent",
                        "Nonmetro - 2,500 to 19,999, non-adjacent",
                        "Nonmetro - 20,000 or more, adjacent",
                        "Nonmetro - 20,000 or more, non-adjacent")

#Histrograms of area__rucc
train_data %>% ggplot(aes(x=heart_disease_mortality_per_100k))+
  xlab("heart_mort_100k")+
  geom_histogram(binwidth = (high_mort-low_mort)/40)+
  facet_wrap(~ area__rucc, ncol = 3)+
  geom_vline(linetype="dashed", xintercept = hd_mean, color = "blue")+
  geom_vline(linetype="dashed", xintercept = hd_med, color = "red")+
  annotate("text", x=hd_mean+45, y=55, label="<- mean", size=2.5, color = "blue")+
  annotate("text", x=hd_med-45, y=55, label="median ->", size=2.5, color = "red")
```



However, we can transform them into population groups as a proxy for Area_RUCC. The transformation is

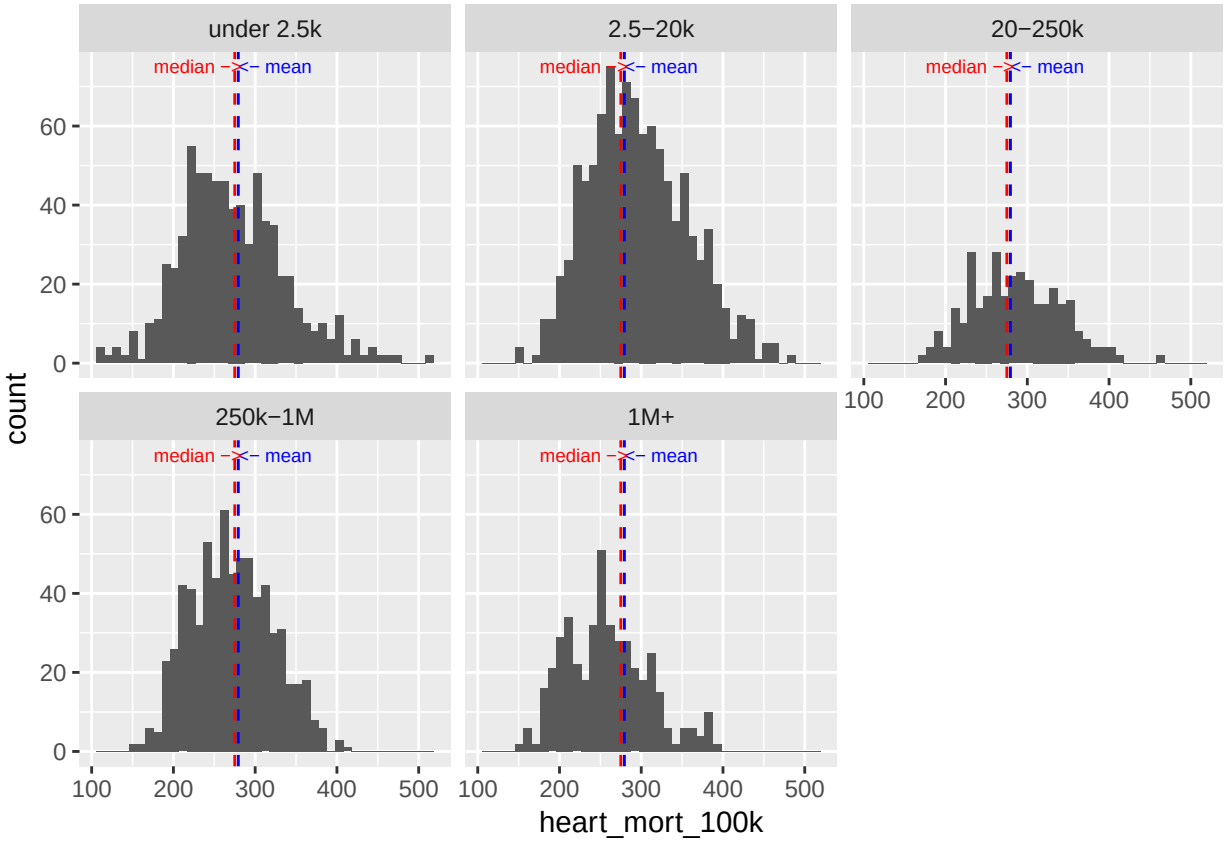
Area_RUCC	Population
Metro − 1 Million +	1M+
Metro − 250,000 to 1 Mil	250k−1M
Metro − less than 250,000	20−250K
Nonmetro − Rural or < 2.5k, adjacent	under 2,500
Nonmetro − Rural or < 2.5k, non−adjacent	under 2,500
Nonmetro − 2,500 to 19,999, adjacent	2.5−20K
Nonmetro − 2,500 to 19,999, non−adjacent	2.5−20k
Nonmetro − 20,000 or more, adjacent	20−250K
Nonmetro − 20,000 or more, non−adjacent	20−250K

```
#Add "population" column using mapping
#Note - 2 different patterns map to 20-250k
train_data <- train_data %>%
  mutate(population = ifelse(like(train_data$area__rucc, "19,999"), "2.5-20k",
                      ifelse(like(train_data$area__rucc, "2.5k"), "under 2.5k",
                      ifelse(like(train_data$area__rucc, "20,000"), "20-250k",
                      ifelse(like(train_data$area__rucc, "fewer than 250,000"), "20-250k",
                      ifelse(like(train_data$area__rucc, "250,000"), "250k-1M",
```

```
                          "1M+"))))))

#Force population into a factor
train_data$population <- factor(train_data$population,
                                levels=c("under 2.5k",
                                         "2.5-20k",
                                         "20-250k",
                                         "250k-1M", "1M+"))
```

Looking at the histograms we now see a more even spread of data and more normal distributions.

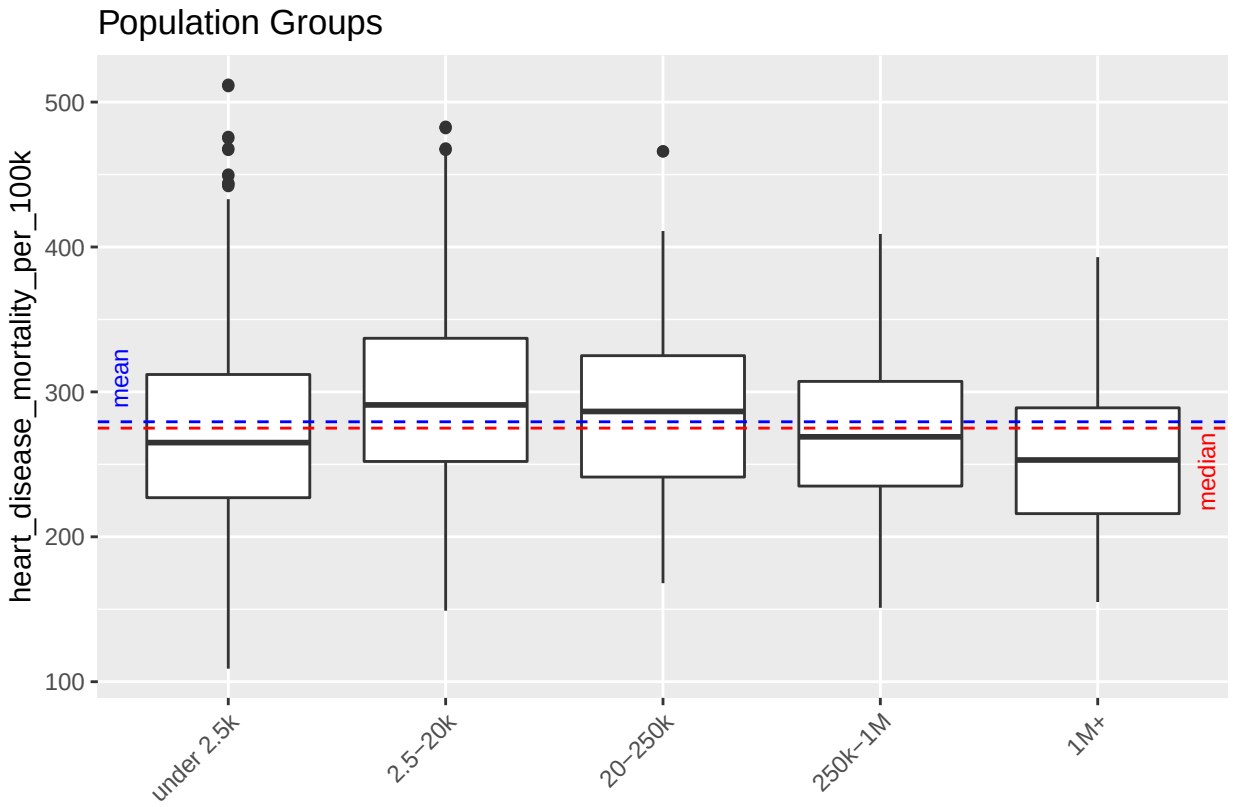
```
#Histograms by population groups
train_data %>% ggplot(aes(x=heart_disease_mortality_per_100k))+
  xlab("heart_mort_100k")+
  geom_histogram(binwidth = (high_mort-low_mort)/40)+
  facet_wrap(~ population, ncol = 3)+
  geom_vline(linetype="dashed", xintercept = hd_mean, color = "blue")+
  geom_vline(linetype="dashed", xintercept = hd_med, color = "red")+
  annotate("text", x=hd_mean+45, y=75, label="<- mean", size=2.5, color = "blue")+
  annotate("text", x=hd_med-45, y=75, label="median ->", size=2.5, color = "red")
```



The "2.5-20K" group appears to be skewed somewhat positive, while the "1M+" group is skewed negative, even though the peak is below the population mean and median. Box plots show the distributions more cleanly.

```
#Box plot by population group
train_data %>%
  ggplot(aes(x=population, y=heart_disease_mortality_per_100k))+
```

```
  xlab("")+
  geom_boxplot() +
  geom_hline(linetype="dashed", yintercept = hd_mean, color = "blue")+
  geom_hline(linetype="dashed", yintercept = hd_med, color = "red")+
  theme(axis.text.x = element_text(angle = 45, hjust=1, vjust=1))+
  ggtitle("Population Groups")+
  annotate("text", x=.5, y=hd_mean+30, label="mean", size=3, color = "blue", angle=90)+
  annotate("text", x=5.5, y=hd_med-30, label="median", size=3, color = "red", angle=90)
```



Here we can more clearly see that the median for "under 2.5K", "250K-1M" and "1M+" are all below the population mean and median. The group "2.5-20K" has a median well above the population mean and median, while the "20-250K" group median is only slightly higher than the population.

```
bind_rows(train_data %>% select(heart_disease_mortality_per_100k,
                        population) %>%
  group_by(population) %>%
  summarize(mn = mean(heart_disease_mortality_per_100k),
            md = median(heart_disease_mortality_per_100k)) %>%
  mutate(MeanAboveBelow = ifelse(round(mn,1) < hd_mean,
                                 "Below",
                                 ifelse(round(mn,1) == hd_mean,
                                        "Equal", "Above")),
         MedianAboveBelow = ifelse(round(md,1) < hd_med,
                                   "Below",
                                   ifelse(round(md,1) == hd_med,
                                          "Equal", "Above"))) %>%
  select(population, mn, MeanAboveBelow, md, MedianAboveBelow),
  data.frame(population = "Total Population Values",
             mn= hd_mean,
```

```
                MeanAboveBelow = "_",
                md = hd_med,
                MedianAboveBelow = "_")
) %>% as.data.frame()
```

```
##                 population       mn MeanAboveBelow    md MedianAboveBelow
## 1               under 2.5k 272.8028          Below 265.0            Below
## 2                  2.5-20k 296.5380          Above 291.0            Above
## 3                  20-250k 286.0807          Above 286.5            Above
## 4                  250k-1M 271.4783          Below 269.0            Below
## 5                      1M+ 257.8005          Below 253.0            Below
## 6 Total Population Values 279.3693              _ 275.0                _
```

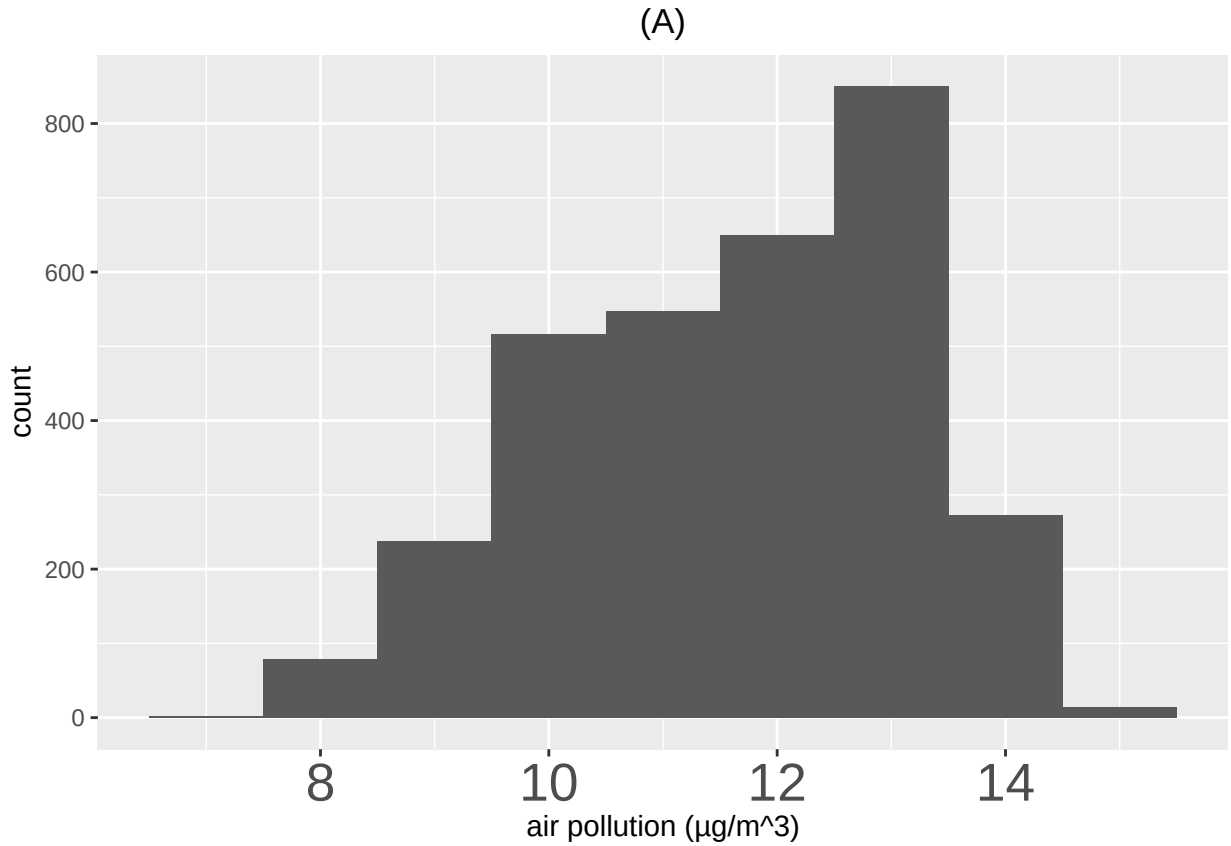
With population groups showing a different distribution for heart mortality rates, this calculated feature is another good candidate for use in a predictive model.

Air Pollution

Ambient air pollution for the county is presented in the data set as a measure of the concentration of fine particulate matter, as measured in $\mu g/m^3$. These figures were presented as positive integers, so as a feature of the data set it is more akin to a categorical variable. Figure A shows the distribution of values.

Air Pollution has 0.88% NAs in the data set (28 rows), so based on figure A, we should be able to set the NAs to the median value without introducing much bias.

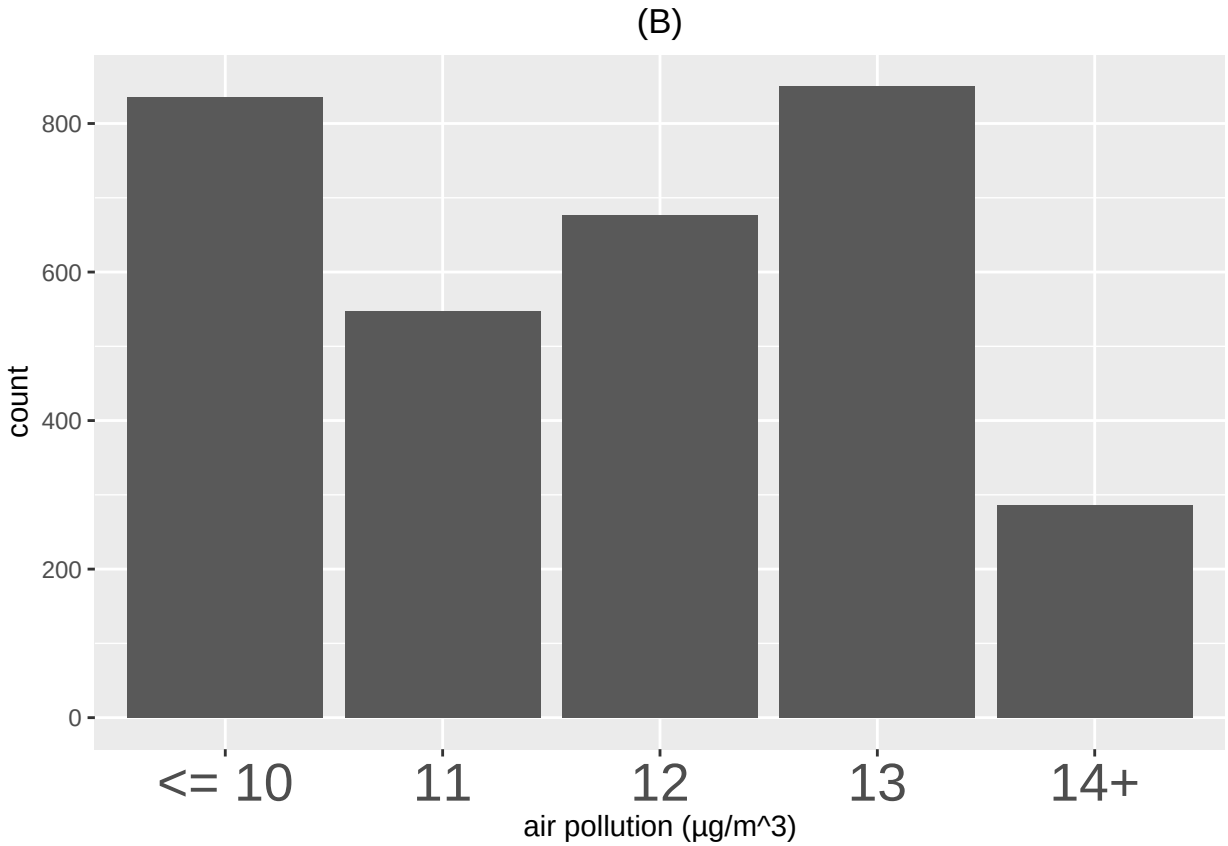
```
#Histogram for air pollution original
train_data %>% filter(!is.na(health__air_pollution_particulate_matter)) %>%
  ggplot(aes(x=health__air_pollution_particulate_matter))+
  geom_histogram(binwidth=1) + xlab("air pollution (µg/m^3)")+
  theme(axis.text.x = element_text(size=20))+
  ggtitle("(A)")+theme(plot.title = element_text(hjust=0.5))
```




```
#Find median, replace NA values with median of population
ap_med <- median(train_data$health__air_pollution_particulate_matter, na.rm=TRUE)
ap_NA <- which(is.na(train_data$health__air_pollution_particulate_matter))

train_data$health__air_pollution_particulate_matter[ap_NA] <- ap_med
```

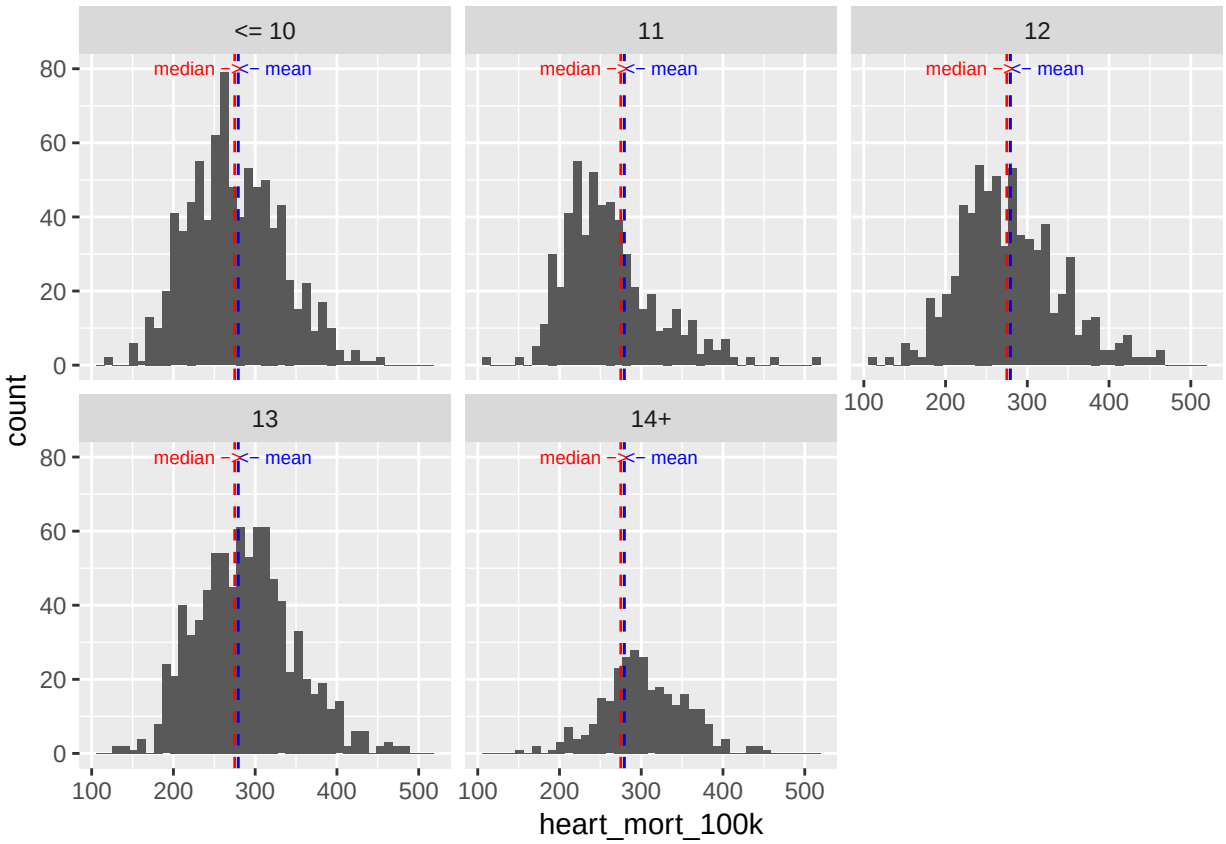
The initial spread of values showed a large discrepancy in population size, especially in the min and max regions so values were grouped into a smaller number of populations that were closer in size (B)

```
#Histogram for air pollution regrouped
cut_labels <- c("<= 10", "11", "12", "13", "14+")
cut_levels <- c(0, 10.9, 11.9, 12.9, 13.9, 100)

train_data$air_pollution <- cut(train_data$health__air_pollution_particulate_matter,
                                cut_levels)
levels(train_data$air_pollution) <- cut_labels

train_data %>%
  ggplot(aes(x=air_pollution))+
  geom_bar() + xlab("air pollution (µg/m^3)")+
  theme(axis.text.x = element_text(size=20))+
  ggtitle("(B)")+theme(plot.title = element_text(hjust=0.5))
```

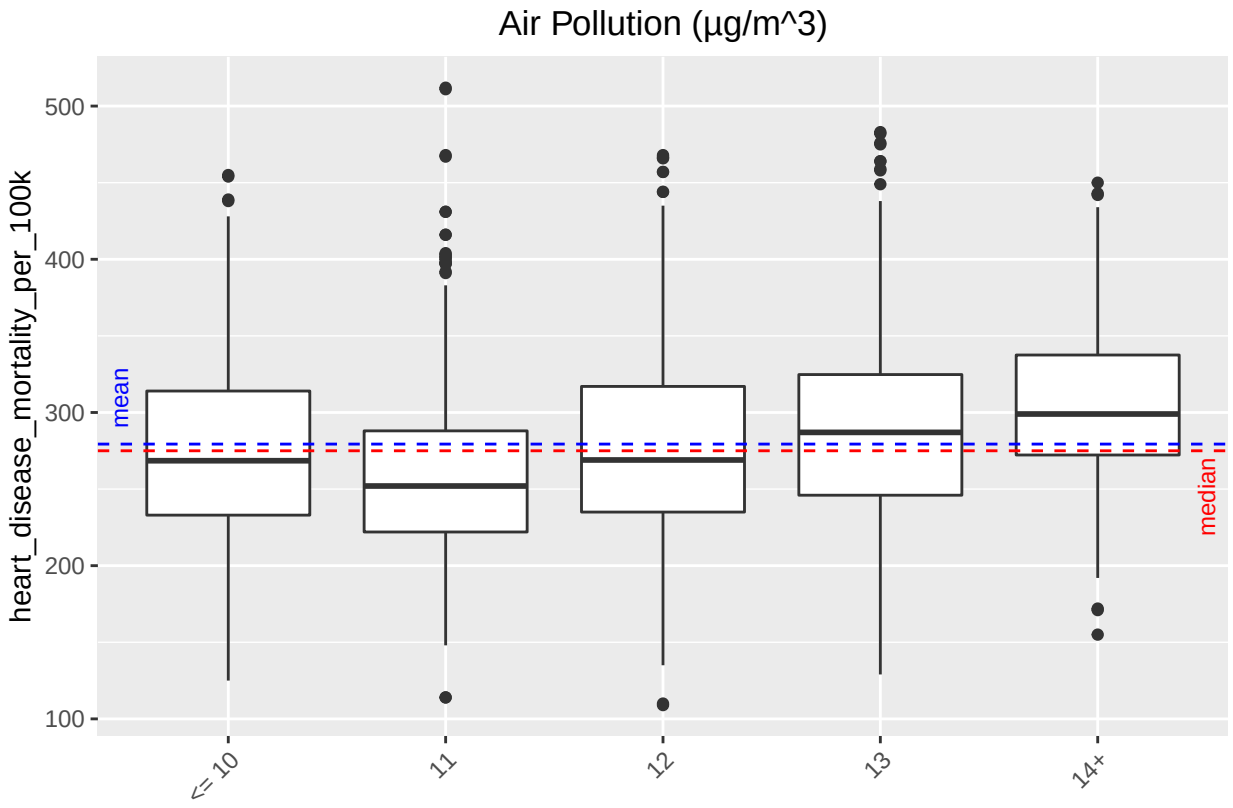


Using the new groups, histograms and boxplots show the spreads to be approximately normal, with some groups skewing below the population mean and some above. The group with 12 µg/m^3 shows a mean just at the population mean, with a median slightly lower.

```
#Histograms by air pollution
train_data %>% ggplot(aes(x=heart_disease_mortality_per_100k))+
  xlab("heart_mort_100k")+
  geom_histogram(binwidth = (high_mort-low_mort)/40)+
  facet_wrap(~ air_pollution, drop=TRUE, ncol = 3)+
  geom_vline(linetype="dashed", xintercept = hd_mean, color = "blue")+
  geom_vline(linetype="dashed", xintercept = hd_med, color = "red")+
  annotate("text", x=hd_mean+45, y=80, label="<- mean", size=2.5, color = "blue")+
  annotate("text", x=hd_med-45, y=80, label="median ->", size=2.5, color = "red")
```




```
#Box plot by air_pollution
train_data %>% ggplot(aes(x=air_pollution, y=heart_disease_mortality_per_100k))+
  xlab("")+
  geom_boxplot()+
  geom_hline(linetype="dashed", yintercept = hd_mean, color = "blue")+
  geom_hline(linetype="dashed", yintercept = hd_med, color = "red")+
  theme(axis.text.x = element_text(angle = 45, hjust=1, vjust=1))+
  ggtitle("Air Pollution (µg/m^3)")+theme(plot.title = element_text(hjust=0.5))+
  annotate("text", x=.5, y=hd_mean+30, label="mean", size=3, color = "blue", angle=90)+
  annotate("text", x=5.5, y=hd_med-30, label="median", size=3, color = "red", angle=90)
```



A statistical summary of the air pollution groups confirms the interpretation of the histogram and box plots, and makes this feature a suitable candidate for a predictive model.

```
bind_rows(train_data %>% select(heart_disease_mortality_per_100k,
                        air_pollution) %>%
  group_by(air_pollution) %>%
  summarize(mn = mean(heart_disease_mortality_per_100k),
            md = median(heart_disease_mortality_per_100k)) %>%
  mutate(MeanAboveBelow = ifelse(round(mn,1) < hd_mean,
                                 "Below",
                                 ifelse(round(mn,1) == hd_mean,
                                        "Equal", "Above")),
         MedianAboveBelow = ifelse(round(md,1) < hd_med,
                                   "Below",
                                   ifelse(round(md,1) == hd_med,
                                          "Equal", "Above"))) %>%
  select(air_pollution, mn, MeanAboveBelow, md, MedianAboveBelow),
  data.frame(air_pollution = "Total Population Values",
             mn= hd_mean,
             MeanAboveBelow = "_",
             md = hd_med,
             MedianAboveBelow = "_")
) %>% as.data.frame()
```

```
##             air_pollution       mn MeanAboveBelow    md MedianAboveBelow
## 1                  <= 10 275.0825          Below 268.5            Below
## 2                     11 261.5985          Below 252.0            Below
## 3                     12 277.8802          Below 269.0            Below
## 4                     13 288.4376          Above 287.0            Above
```

```
## 5                      14+ 303.1608          Above 299.0            Above
## 6 Total Population Values 279.3693              _ 275.0                _
```

Correlation Between Numeric Features and Heart Disease Mortality

Handling NAs

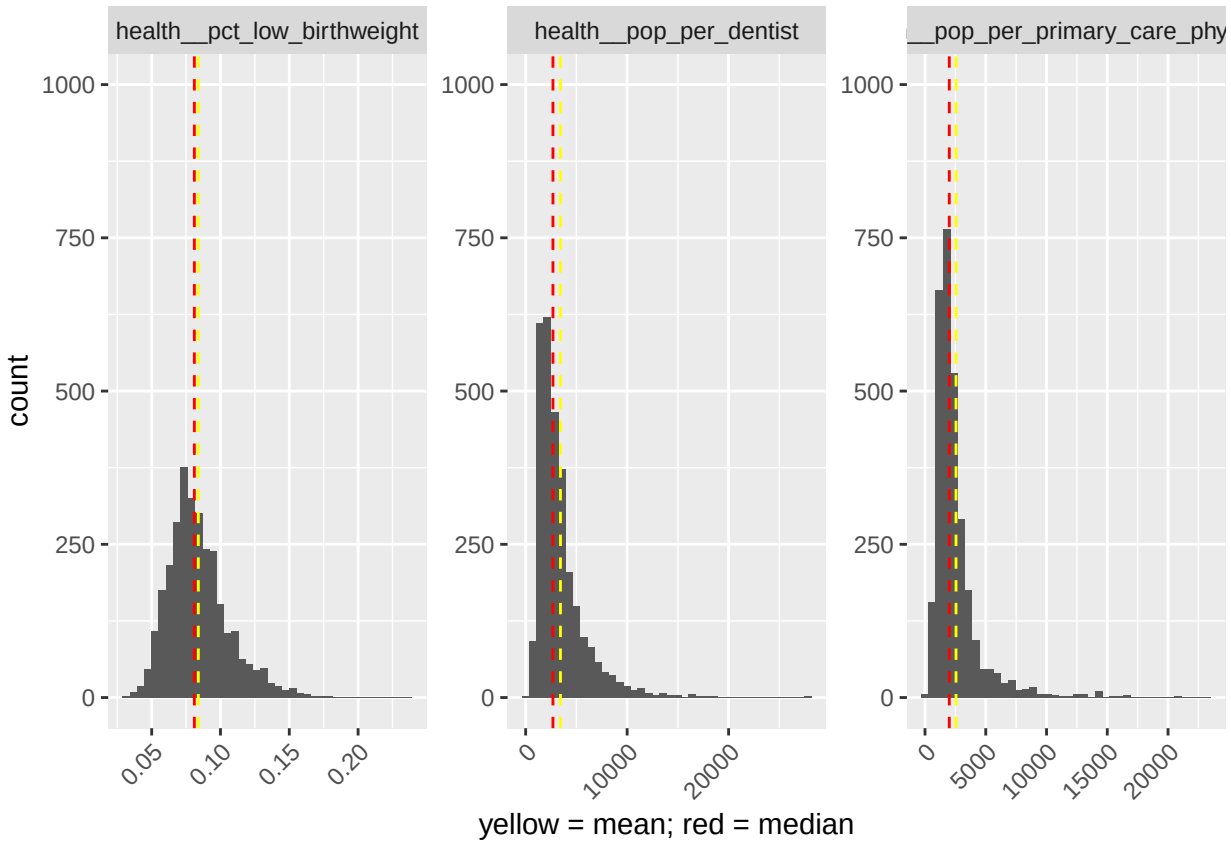
There are 3 features that have not yet been handled that all have an NA percentage between 1% and 10%

```
#Which features have a NA % between 1 and 10
na_mid <- NAs[NAs$NA_pct > 1 & NAs$NA_pct < 10, ]
na_mid %>% arrange(desc(NA_pct))
```

```
##                                features NA_pct
## 1                health__pop_per_dentist 7.6298
## 2 health__pop_per_primary_care_physician 7.1920
## 3            health__pct_low_birthweight 5.6911
```

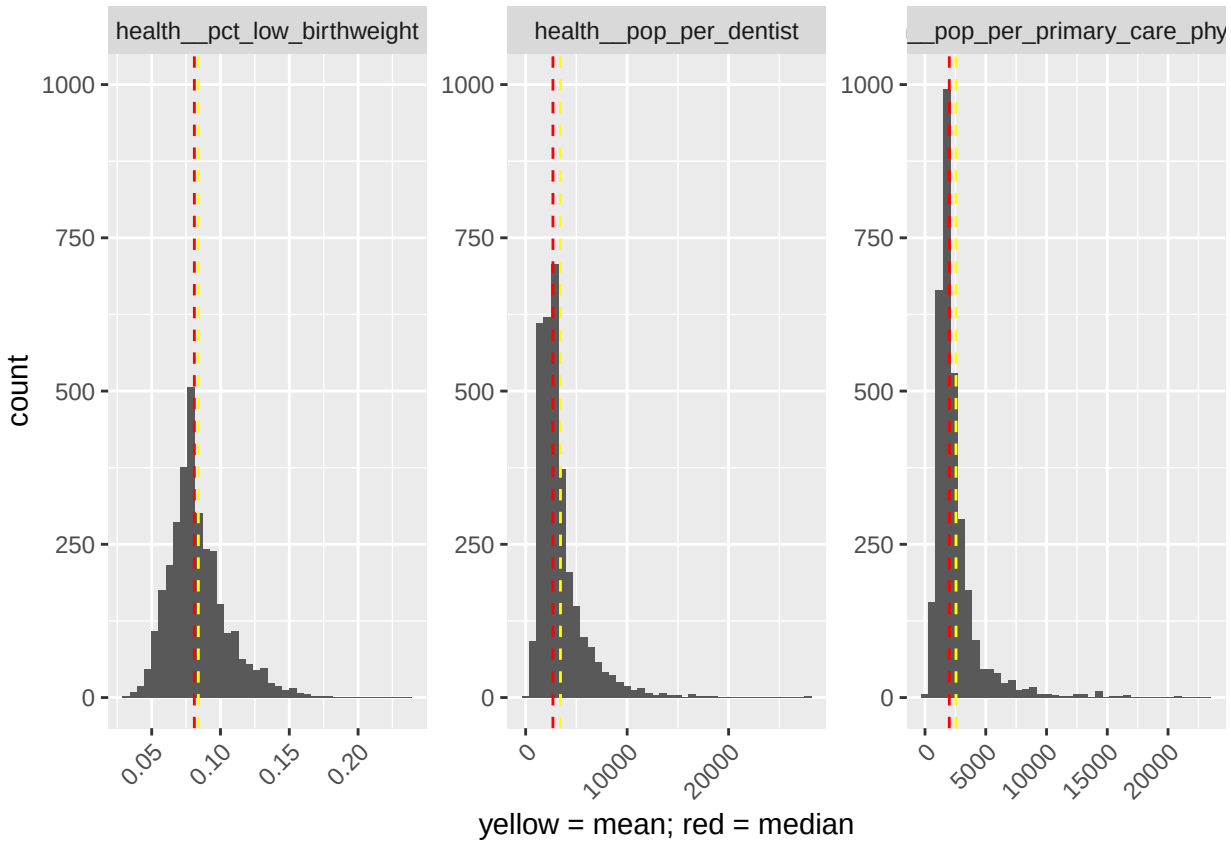
We examine the distributions of these continuous features, showing the mean and median, to determine how best to handle the NAs

```
#Calculate the mean and median of the features in question
feat_means <- train_data %>% select(as.vector(na_mid[,1])) %>%
  gather(feature, value) %>% group_by(feature) %>%
  mutate(mn = mean(value, na.rm=TRUE), md = median(value, na.rm=TRUE))

#Plot histogram, faceted by feature, overlaying the mean and median values of each
train_data %>% select(as.vector(na_mid[,1])) %>%
  gather(feature, value) %>%
  ggplot(aes(x=value))+geom_histogram(bins=40)+
  geom_vline(aes(xintercept=mn), data = feat_means,
             linetype = "dashed", color = "yellow")+
  geom_vline(aes(xintercept=md), data = feat_means,
             linetype = "dashed", color = "red")+
  theme(axis.text.x = element_text(angle = 45, hjust=1, vjust=1))+
  facet_wrap(~ feature, scales="free") +
  xlab("yellow = mean; red = median") + ylim(0, 1000)
```



All 3 features appear approximately normal with a positive skew. The mean and median are different enough that, combined with the skewness, we should be able to replace the NAs with the median value and not introduce too much bias.

```
#Replace the NAs with the median for each feature
mid_cols <- as.vector(na_mid[, 1])

train_data <- train_data %>%
  mutate_at(mid_cols, ~ifelse(is.na(.x), median(.x, na.rm = TRUE), .x))

#Re-plot the distros
train_data %>% select(as.vector(na_mid[,1])) %>%
  gather(feature, value) %>%
  ggplot(aes(x=value))+geom_histogram(bins=40)+
  geom_vline(aes(xintercept=mn), data = feat_means,
             linetype = "dashed", color = "yellow")+
  geom_vline(aes(xintercept=md), data = feat_means,
             linetype = "dashed", color = "red")+
  theme(axis.text.x = element_text(angle = 45, hjust=1, vjust=1))+
  facet_wrap(~ feature, scales="free") +
  xlab("yellow = mean; red = median") + ylim(0, 1000)
```

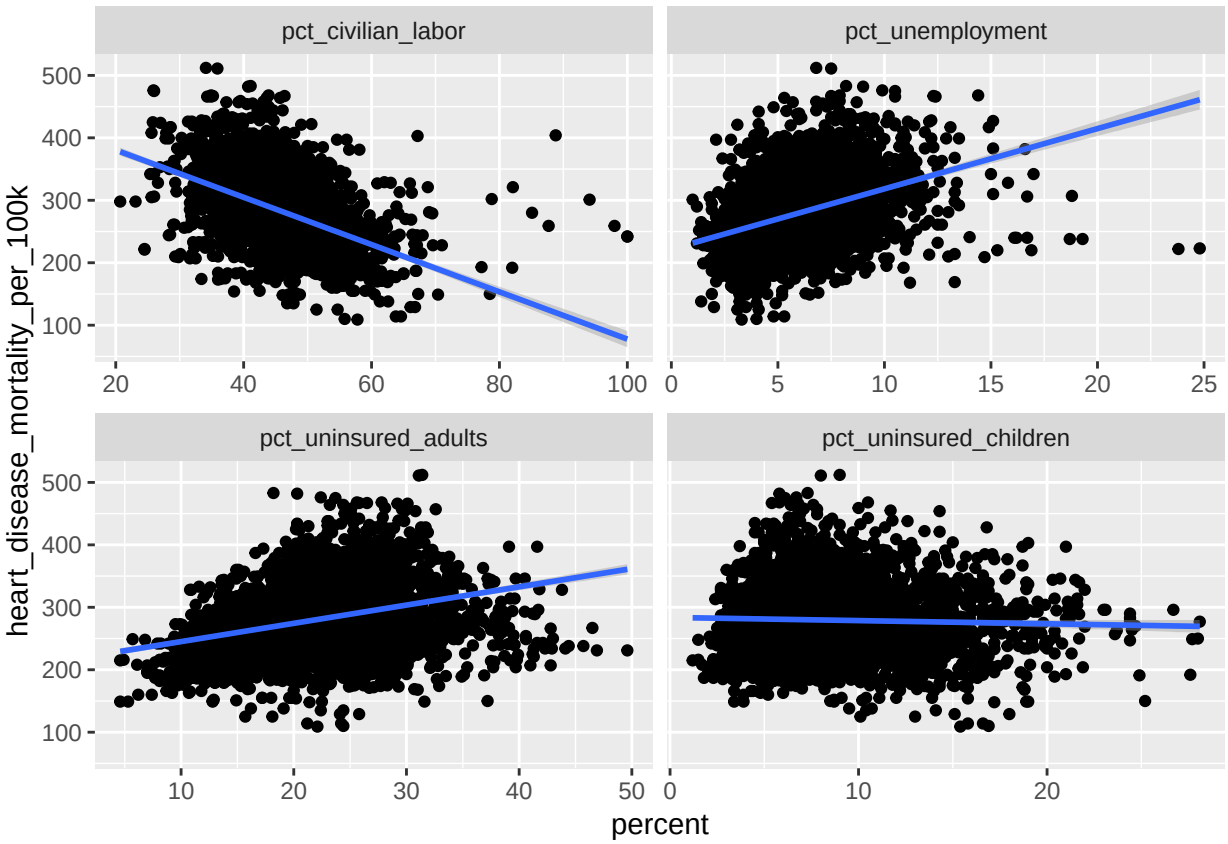


Economic Factors

The remaining economic factors in the dataset (after already looking at typology) are numeric, so a correlation analysis is done to determine if any of these are candidates for use in a predictive model.

- % Civilian Labor
- % Unemployment
- % Uninsured Adults
- % Uninsured Children

```
cols <- colnames(train_data)[like(colnames(train_data), "econ__pct")]
ncol <- ifelse(length(cols) <= 6, 2, 3)

train_data %>% select(heart_disease_mortality_per_100k, cols) %>%
  gather(data_type, val, -heart_disease_mortality_per_100k) %>%
  mutate(data_type = substr(data_type, 7, 100)) %>%
  mutate(val= val*100) %>%
  ggplot(aes(x=val, y= heart_disease_mortality_per_100k))+geom_point(na.rm = TRUE)+
  facet_wrap(~ data_type, ncol=ncol, scales="free_x")+xlab("percent")+
  geom_smooth(method="lm", na.rm = TRUE)
```




```
data.frame(
  feature=names(train_data[cols]),
  corr = sapply(train_data[cols], function(x) {
    cor(data.frame(train_data$heart_disease_mortality_per_100k, x),
        use="complete.obs")[1,2] })
  ) %>% arrange(desc(abs(corr))) %>% grid.table()
```

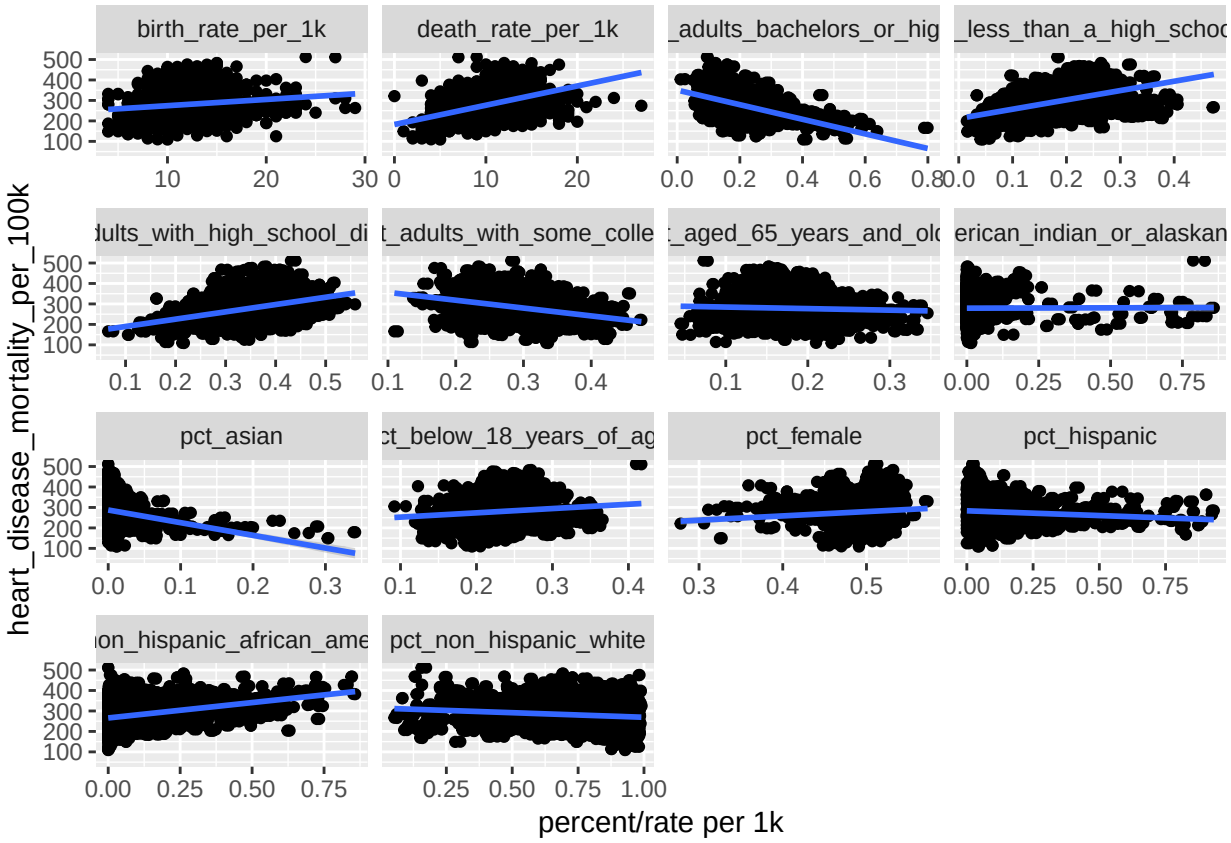
	feature	corr
1	econ__pct_civilian_labor	−0.47550669
2	econ__pct_unemployment	0.37497459
3	econ__pct_uninsured_adults	0.33421710
4	econ__pct_uninsured_children	−0.03448192

As seen above, **"Percent of Civilian Labor"** shows a moderate negative correlation with heart disease mortality (despite the number of outliers appearing to be greater than the other factors) and is a good candidate for use in a predictive model. The remaining factors do not show a strong enough correlation (>= 0.45) to be considered good candidates.

Demographics

The dataset contains 12 demographic features that are reported as percentages of the population and 2 that are given in rates per 1000 people. A correlation analysis of those features is done in a similar manner to the economic factors.

```
cols <- colnames(train_data)[like(colnames(train_data), "demo__")]
ncol <- 4

train_data %>% select(heart_disease_mortality_per_100k, cols) %>%
  gather(data_type, val, -heart_disease_mortality_per_100k) %>%
  mutate(data_type = substr(data_type, 7, 100)) %>%
  ggplot(aes(x=val, y= heart_disease_mortality_per_100k))+geom_point(na.rm = TRUE)+
  facet_wrap(~ data_type, ncol=ncol, scales="free_x")+xlab("percent/rate per 1k")+
  geom_smooth(method="lm")
```




```
data.frame(
  feature=substr(names(train_data[cols]), 7, 100),
  corr = sapply(train_data[cols], function(x) {
    cor(data.frame(train_data$heart_disease_mortality_per_100k, x),
        use="complete.obs")[1,2] })
  ) %>% arrange(desc(abs(corr))) %>% grid.table()
```

	feature	corr
1	pct_adults_bachelors_or_higher	−0.54058907
2	pct_adults_less_than_a_high_school_diploma	0.52657438
3	death_rate_per_1k	0.44340519
4	pct_adults_with_high_school_diploma	0.42771356
5	pct_non_hispanic_african_american	0.37538491
6	pct_adults_with_some_college	−0.34063899
7	pct_asian	−0.26745832
8	pct_non_hispanic_white	−0.15754385
9	birth_rate_per_1k	0.14154707
10	pct_below_18_years_of_age	0.12195605
11	pct_hispanic	−0.11243701
12	pct_female	0.08704009
13	pct_aged_65_years_and_older	−0.05620304
14	pct_american_indian_or_alaskan_native	0.00462565

Analysis of the correlation scatter plots showed that there were several candidates for model features. **"Percent of non-Hispanic African Americans"** was chosen as it had the strongest correlation in the subset of race/ethnicity demographics, even though its actual correlation was only moderate ($< .45$). **"Adults with less than a High School diploma"** and **"Adults with a Bachelor's Degree or higher"** were chosen for their strong positive and negative correlations, respectively. **"Death Rate per 1000"** also showed a high-moderate correlation.

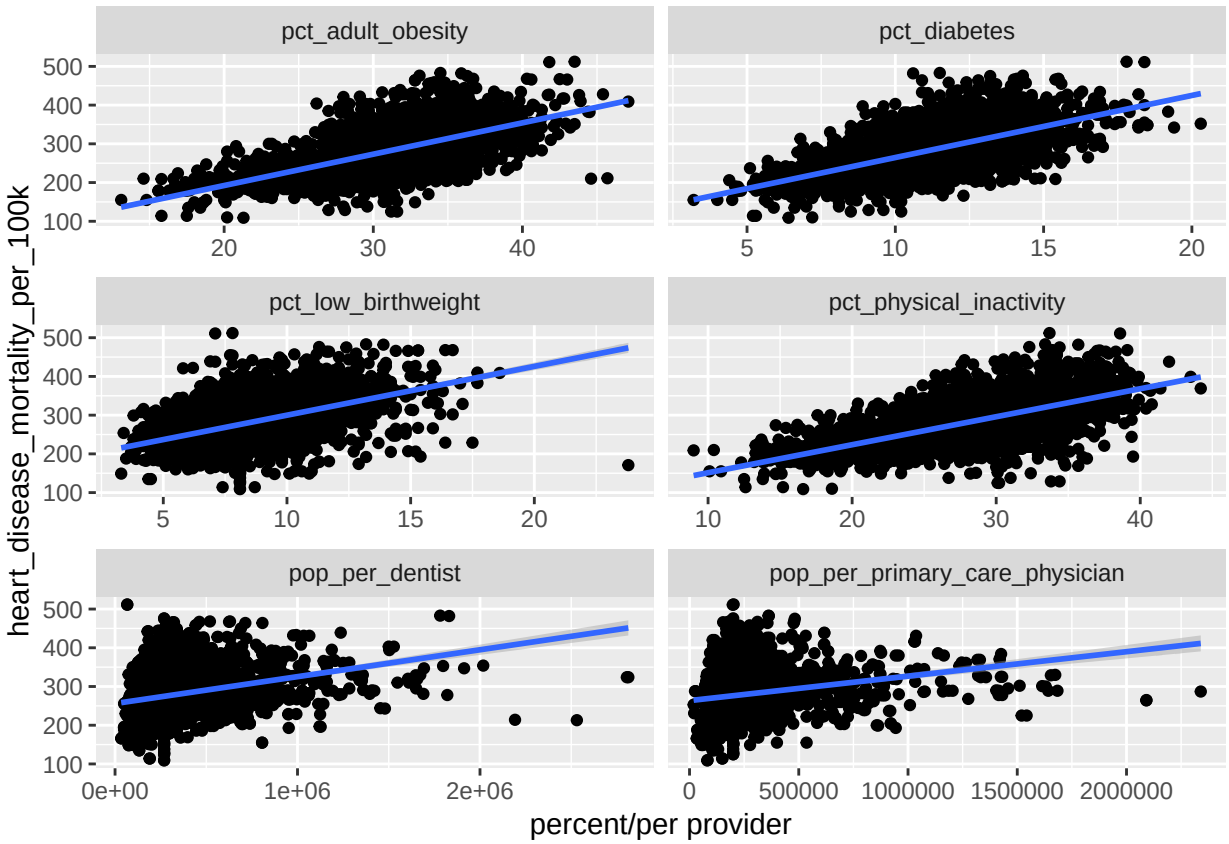
Because of their extremely weak correlations (abs value $< .15$), demographic features related to gender and age were not considered for the model.

Health Statistics

There are 6 remaining health-related factors present in the dataset (after air pollution and the high NA features were removed). All are numerical, 4 are presented as percentage of the population, and 2 are rates related to the number of people per medical professional (primary care doctor, dentist).

A scatter plot and correlation analysis was done for the 6 remaining numerical health-related features:

```
cols <- colnames(train_data)[like(colnames(train_data), "health__")]
cols <- cols[!like(cols, "health__air")]
ncol <- ifelse(length(cols) <= 6, 2, 3)

train_data %>% select(heart_disease_mortality_per_100k, cols) %>%
  gather(data_type, val, -heart_disease_mortality_per_100k) %>%
  mutate(data_type = substr(data_type, 9, 100)) %>%
  mutate(val = val * 100) %>%
  ggplot(aes(x=val, y= heart_disease_mortality_per_100k))+geom_point(na.rm = TRUE)+
  facet_wrap(~ data_type, ncol=ncol, scales="free_x")+xlab("percent/per provider")+
  geom_smooth(method="lm", na.rm = TRUE)
```




```
data.frame(
  feature=names(train_data[cols]),
  corr = sapply(train_data[cols], function(x) {
    cor(data.frame(train_data$heart_disease_mortality_per_100k, x),
        use="complete.obs")[1,2] })
  ) %>% arrange(desc(abs(corr))) %>% grid.table()
```

	feature	corr
1	health__pct_physical_inactivity	0.6503046
2	health__pct_diabetes	0.6317650
3	health__pct_adult_obesity	0.5937750
4	health__pct_low_birthweight	0.4626653
5	health__pop_per_dentist	0.2920029
6	health__pop_per_primary_care_physician	0.2174194

The 2 features related to medical professionals did not show a strong correlation (> 0.45) to heart disease mortality rate, so they were not considered for the predictive model.

"Percent of Physical Inactivity" shows the strongest correlation with heart disease mortality among the health percentages, indicating a strong candidate for the predictive model. Two other percentages (Obesity, Diabetes) also showed strong correlation with heart disease mortality, but some further thought and analysis produced features with even stronger correlations.

Combining Health Percentages

Thinking about the 3 health percentages:

- Percent of adult obesity
- Percent of physical inactivity
- Percent of diabetes

we can say that each of the percentages is the same as the probability of a randomly chosen individual in that county having the specific condition. Let us further assume that each of the events (being obese, being physically inactive, having diabetes) are independent of each other. According to probability theory, the probability of two independent events occurring is simply the product of the two individual probabilities. Therefore

$$\text{P(diabetes and obese)} = \text{P(diabetes)} * \text{P(obese)}$$

$$\text{P(diabetes and physical inactivity)} = \text{P(diabetes)} * \text{P(physical inactivity)}$$

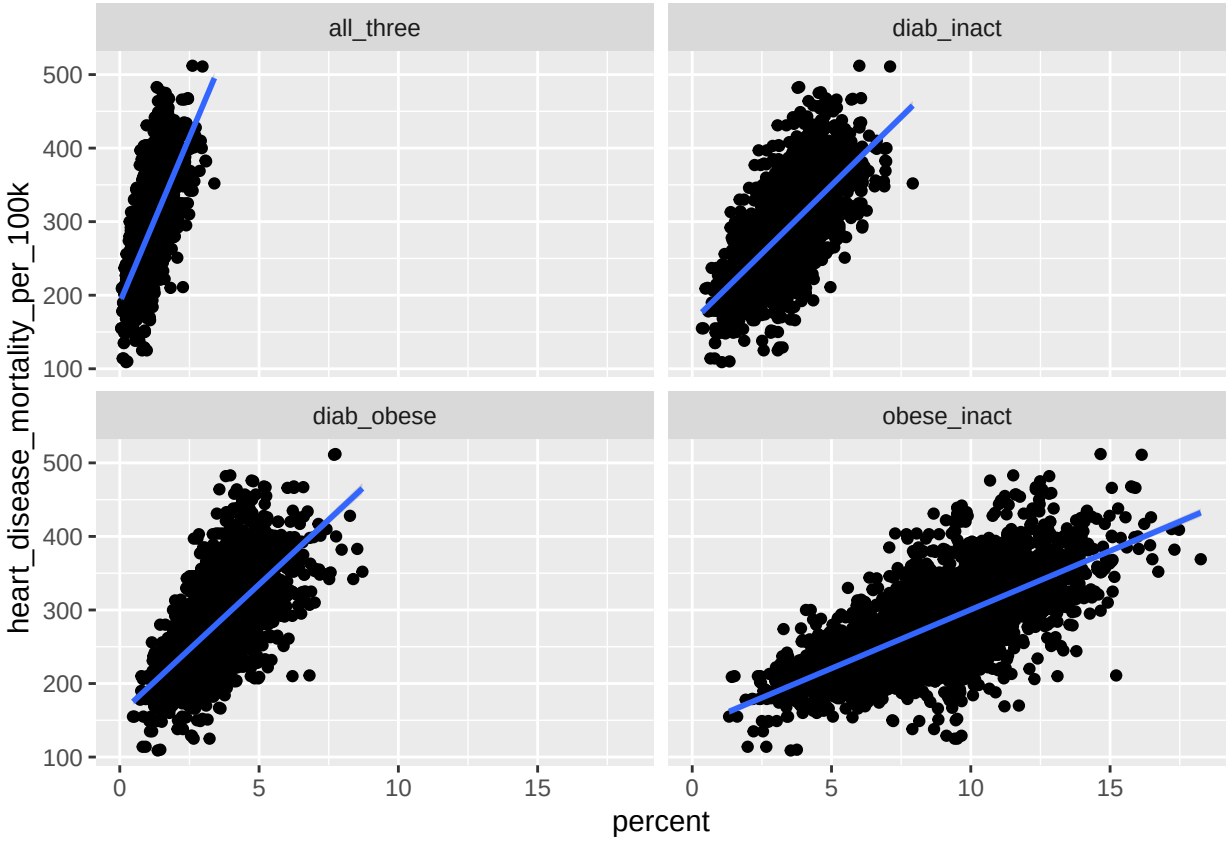
$$\text{P(obese and physical inactivity)} = \text{P(obese)} * \text{P(physical inactivity)}$$

$$\text{P(diabetes and obese and physical inactivity)} = \text{P(diabetes)} * \text{P(obese)} * \text{P(physical inactivity)}$$

By calculating these combined probabilities, we find features that have even stronger correlation that each individual feature

```
#Create new features
train_data <- train_data %>%
  mutate(p_diab_obese = health__pct_adult_obesity * health__pct_diabetes) %>%
  mutate(p_obese_inact = health__pct_adult_obesity * health__pct_physical_inactivity) %>%
  mutate(p_diab_inact = health__pct_diabetes * health__pct_physical_inactivity) %>%
  mutate(p_all_three =
           health__pct_adult_obesity *
           health__pct_diabetes *
           health__pct_physical_inactivity )
```

```
cols <- c("p_diab_obese",
          "p_obese_inact",
          "p_diab_inact",
          "p_all_three")

ncol <- ifelse(length(cols) <= 6, 2, 3)

train_data %>% select(heart_disease_mortality_per_100k, cols) %>%
  gather(data_type, val, -heart_disease_mortality_per_100k) %>%
  mutate(data_type = substr(data_type, 3, 100)) %>%
  mutate(val = val * 100) %>%
  ggplot(aes(x=val, y= heart_disease_mortality_per_100k))+geom_point(na.rm = TRUE)+
  facet_wrap(~ data_type, ncol=ncol, scales="fixed")+xlab("percent")+
  geom_smooth(method="lm", na.rm = TRUE)
```



	feature	corr
1	p_diab_inact	0.7020959
2	p_all_three	0.7008151
3	p_obese_inact	0.6882648
4	p_diab_obese	0.6603029

All of the combinations had a correlation higher than any one individual feature, so we will use these combined probabilities in the predictive model.

Predictive Modelling

The features chosen for use in the predictive model are

- Economic Typology (categorical)
- Population (categorical)
- Air Pollution concentration (categorical)
- Percent of Civilian Labor (numeric)
- Percent non-Hispanic African American (numeric)
- Death Rate per 1,000 (numeric)
- Percent of adults with less than a High School diploma (numeric)
- Percent of adults with Bachelor's Degree or higher (numeric)
- Percent of adults that are obese and have diabetes (numeric)
- Percent of adults that are obese and are physically inactive (numeric)
- Percent of adults that have diabetes and are physically inactive (numeric)

- Percent of adults that are obese, have diabetes and are physically inactive (numeric)

```
#Remove unused columns from the data set
model_cols <- c("econ__economic_typology",
                "population",
                "air_pollution",
                "econ__pct_civilian_labor",
                "demo__pct_non_hispanic_african_american",
                "demo__death_rate_per_1k",
                "demo__pct_adults_bachelors_or_higher",
                "demo__pct_adults_less_than_a_high_school_diploma",
                "p_diab_obese",
                "p_obese_inact",
                "p_diab_inact",
                "p_all_three")

train_data <- train_data %>% select(model_cols, heart_disease_mortality_per_100k)
```

Since the outcome of the model (Heart Disease Mortality per 100,000 people) needs to be a number (as opposed to a class or category), we will need to use some kind of regression model to get the outcome we desire.

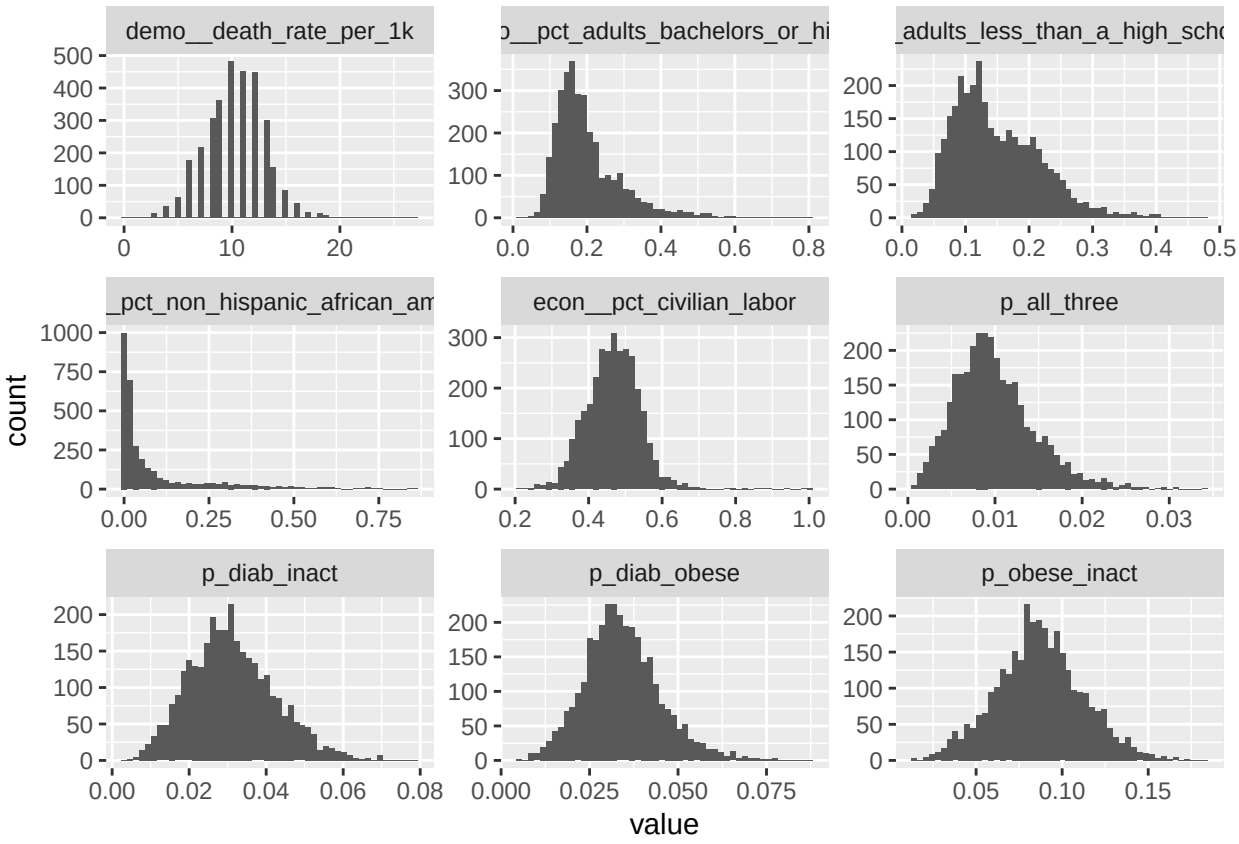
Normalization of Numeric Features

For a regression model, if we have features that are different by orders of magnitude we likely need to do something so that larger values do not dominate the formula. Since most of the numeric features are percentages (i.e. $0 < x < 1$), we will apply a transformation to reduce the relative effect of the "Death Rate per 1,000" feature.

Exploration of feature distributions

We plot histograms of each of the continuous features to get more information about each feature

```
#Find all numeric columns left in the data set
num_cols <- train_data %>%
  select(-heart_disease_mortality_per_100k) %>%
  .[sapply(., is.numeric)] %>%
  colnames()

#Plot histograms
train_data %>% select(num_cols) %>%
  gather(feature, value) %>%
  ggplot(aes(x=value)) + geom_histogram(bins=50) +
  facet_wrap(~ feature, scales="free")
```



The continuous features show an approximate normal distribution (although some are skewed), with the exception of **"Percent non-Hispanic African American"** and **"Percent of adults with less than a High School Diploma"**. With approximate normal features, there are several options to consider for normalization:

- Scaling/z-score
- Cube Root
- Square Root
- Log10
- Natural Log (Log e)

For the logarithmic transformations, the data would have to be completely free of negative and zero values. Looking at the overall range of the data

```
#Find overall range of predictors by taking min and max of individual predictor ranges
ranges <- sapply(train_data %>% select(num_cols), range, simplify=TRUE)
data.frame(MIN = min(ranges[1,]), MAX = max(ranges[2,]))
```

```
##   MIN MAX
## 1   0  27
```

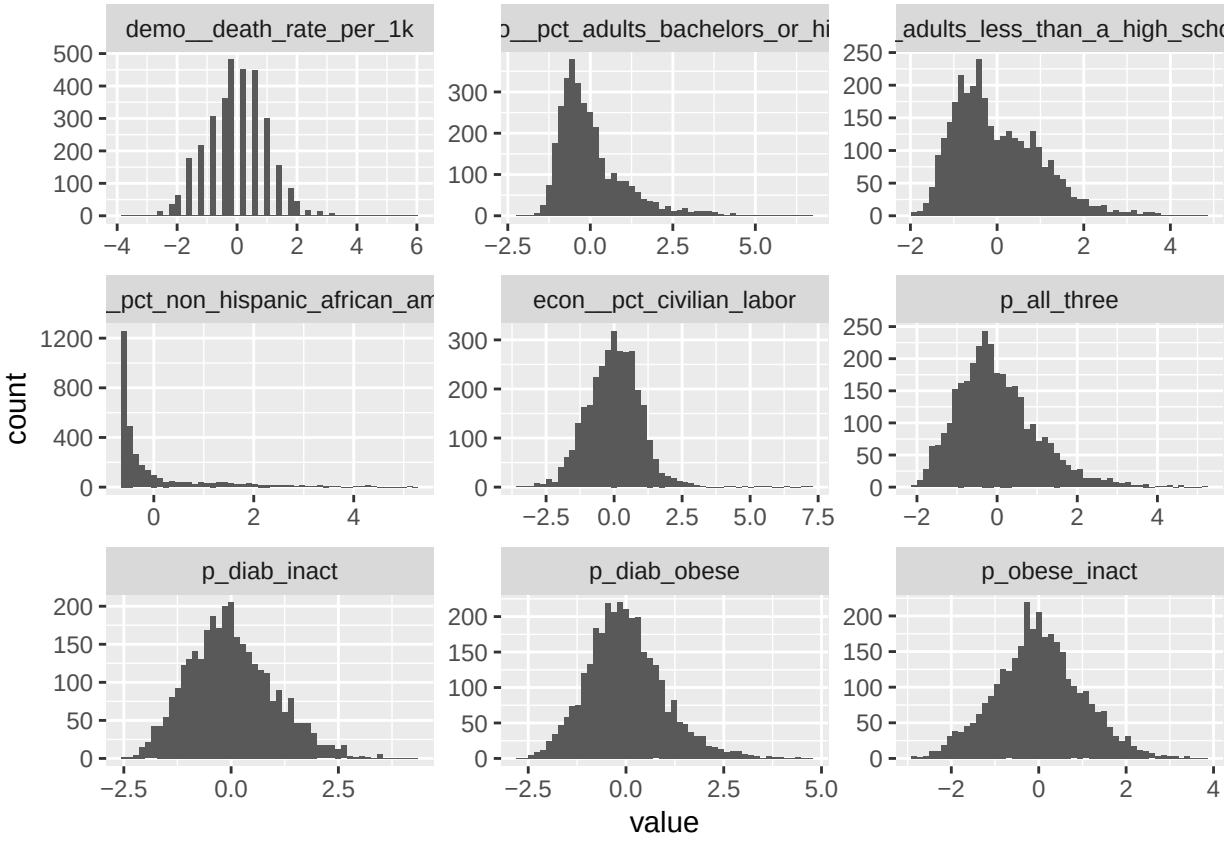
we see that there is at least one feature with a 0 data point. Since we cannot assume that 0 is an invalid data point, we must exclude the logarithmic transformations.

Similary, were there any features with negative values, we could not use the square root transformation, but that is not the case here.

We apply the scale, cube root and square root transformations and look at the resulting features

```
#Apply scale/z-score, plot distributions, show range
data_scale <- scale(train_data[num_cols])
```

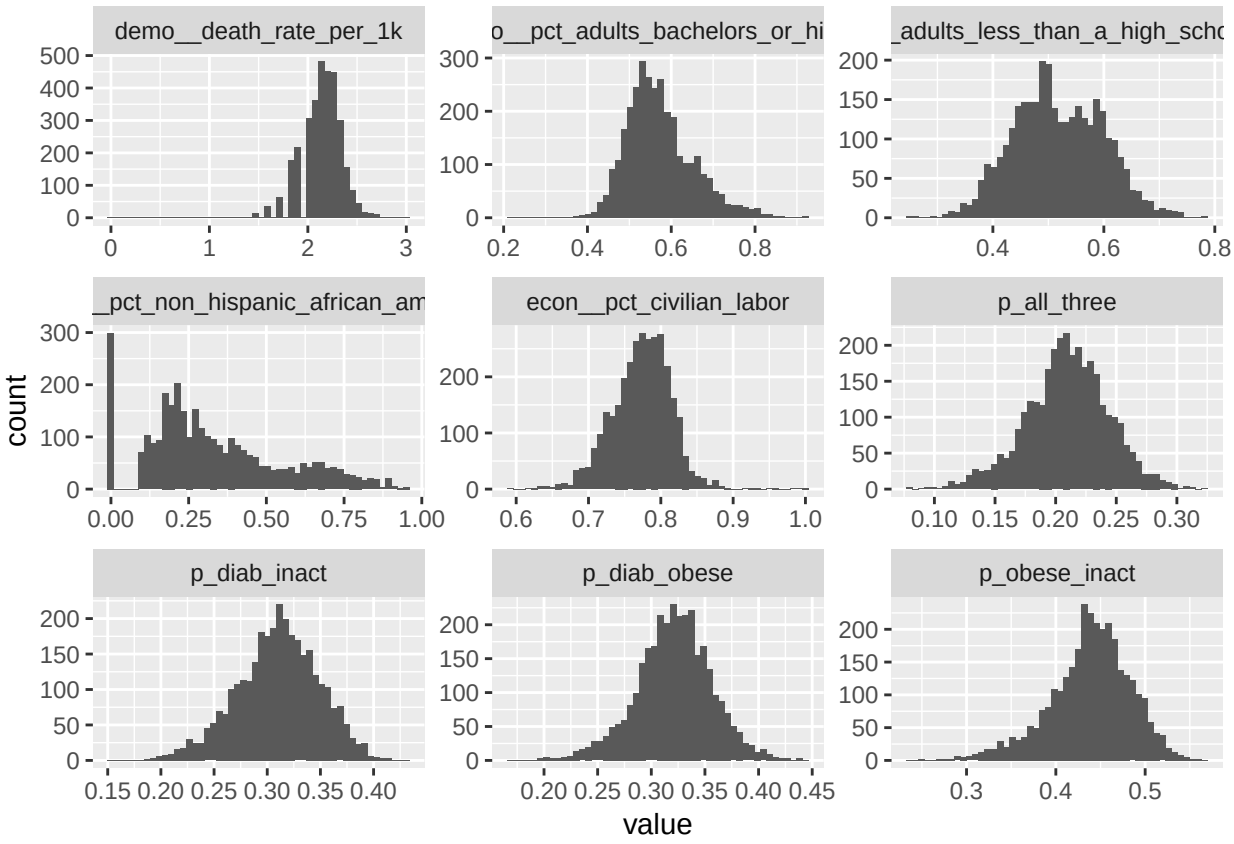
```
as.data.frame(data_scale) %>%
  gather(feature, value) %>%
  ggplot(aes(x=value)) + geom_histogram(bins=50) +
  facet_wrap(~ feature, scales="free")
```




```
ranges <- sapply(data_scale, range, simplify=TRUE)

data.frame(MIN = min(ranges[1,]), MAX = max(ranges[2,]))
```

```
##         MIN      MAX
## 1 -3.711321 7.199119
```

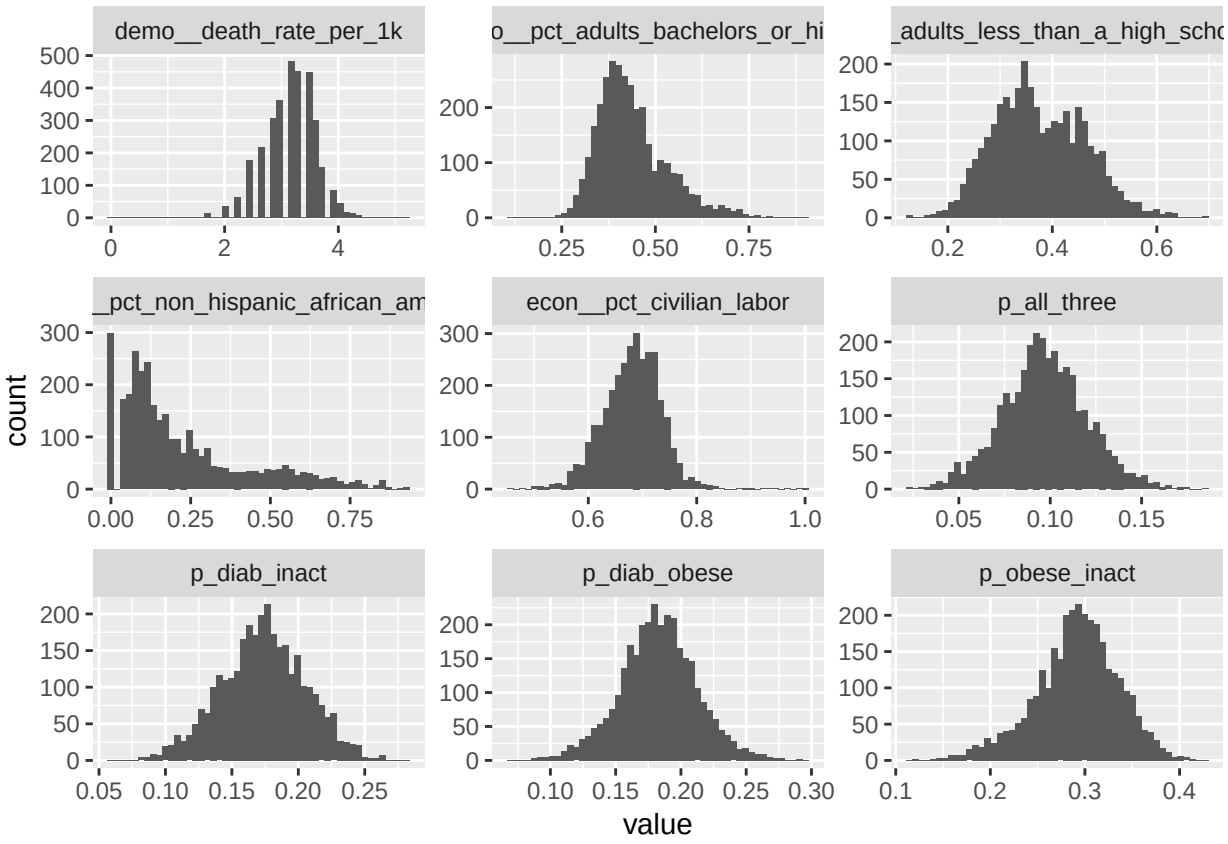
```
#Apply cube root transformation, plot distributions, show range
data_scale <- sapply(train_data[num_cols], function(x) {x^(1/3)}, simplify=TRUE)
as.data.frame(data_scale) %>%
  gather(feature, value) %>%
  ggplot(aes(x=value)) + geom_histogram(bins=50) +
  facet_wrap(~ feature, scales="free")
```




```
ranges <- sapply(data_scale, range, simplify=TRUE)

data.frame(MIN = min(ranges[1,]), MAX = max(ranges[2,]))
```

```
##   MIN MAX
## 1   0   3
```

```
#Apply square root transformation, plot distributions, show range
data_scale <- sapply(train_data[num_cols], sqrt, simplify=TRUE)
as.data.frame(data_scale) %>%
  gather(feature, value) %>%
  ggplot(aes(x=value)) + geom_histogram(bins=50) +
  facet_wrap(~ feature, scales="free")
```




```
ranges <- sapply(data_scale, range, simplify=TRUE)

data.frame(MIN = min(ranges[1,]), MAX = max(ranges[2,]))
```

```
##   MIN      MAX
## 1   0 5.196152
```

While the Scale/Z-Score transformation gives us the largest overall range of the three, the histograms show that all features are now within the same order of magnitude (while cube root and square root still have Death Rate per 1000 above 1 and all percentages below 1). We will apply the Scale transformation to the data set and then create the regression models.

```
#Categorical columns
cat_cols <- colnames(train_data)[sapply(train_data, is.factor)]

#Create new DF with the scaled data
model_data <- cbind(train_data %>% select(heart_disease_mortality_per_100k, cat_cols),
                    sapply(train_data[num_cols], scale, simplify=TRUE))
```

Training and test sets

Best practice is to take a random sampling of the entire data set to use as a "training set", on which to train the model, and then use the remainder as a "test set" for evaluating the robustness of the model. We will split our data 50/50 into the 2 sets:

```
set.seed(999)
test_index <- createDataPartition(model_data$heart_disease_mortality_per_100k,
                                  times = 1, p=0.5, list=FALSE)
```

```
train_set <- model_data[-test_index, ]
test_set <- model_data[test_index, ]
```

Creating and evaluating regression models

Now that the data is prepared, we look at different regression models and determine their predictive power. There are many ways to evaluate the strength of a regression model, but we have chosen to use Root Mean Squared Error:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y)^2}$$

which calculated in R looks like this

```
RMSE <- sqrt(mean((y_i - y)^2))
```

where y_i is the predicted value and y is the true value.

We chose the following regression models to examine:

- Linear Regression (lm)
- Decision/Regression Tree (rpart)
- Random Forest
- randomForest
- Rborist

Linear Regression

First we ran the "lm" linear regression model to get an RMSE for a basic algorithm

```
#Linear regression model
fit <- lm(formula = heart_disease_mortality_per_100k ~ ., data = train_set)
y_hat <- predict(fit, test_set)
sqrt(mean((y_hat-test_set$heart_disease_mortality_per_100k)^2))
```

```
## [1] 36.61253
```

Decision/Regression Tree

The next regression method tried was a regression tree. One method for fitting a model of this kind is to create a model where each data value is its own node then train the model such that each node improves the RMSE by a certain amount at each split. This amount is called "complexity parameter".

We first run the model by training and predciting against the entire data set, then again by training against the train set and predicting against the test set.

Full dataset

```
#Start with complexity parameter = 0 then prune
rt_fit <- rpart(heart_disease_mortality_per_100k ~ ., data = model_data,
                control = rpart.control(cp=0, minsplit=2))

y_hat_rt <- predict(rt_fit)
```

```r
#We know this is going to be 0 as each row is its own node
sqrt(mean((y_hat_rt-model_data$heart_disease_mortality_per_100k)^2))
```

```
## [1] 0
```

```r
#Train the complexity parameter
train_rt <- train(heart_disease_mortality_per_100k ~ .,
                  method = "rpart",
                  tuneGrid = data.frame(cp = seq(0, 0.05, len=100)),
                  data = model_data)

bt <- as.numeric(train_rt$bestTune[which.min(train_rt$bestTune)])
```

```
## [1] "cp=0.00253"
```

```r
#Prune the regression tree using the best cp parameter
rt_pruned <- prune(rt_fit, cp=bt)

y_hat_pruned <- predict(rt_pruned)

sqrt(mean((y_hat_pruned - model_data$heart_disease_mortality_per_100k)^2))
```

```
## [1] 35.4419
```

Train/test sets

```r
#Start with complexity parameter = 0 then prune
rt_fit <- rpart(heart_disease_mortality_per_100k ~ ., data = train_set,
                control = rpart.control(cp=0, minsplit=2))

y_hat_rt <- predict(rt_fit, train_set)

#We know this is going to be 0 as each row is its own node
sqrt(mean((y_hat_rt - train_set$heart_disease_mortality_per_100k)^2))
```

```
## [1] 0
```

```r
#Train the complexity parameter
train_rt <- train(heart_disease_mortality_per_100k ~ .,
                  method = "rpart",
                  tuneGrid = data.frame(cp = seq(0, 0.05, len=100)),
                  data = train_set)
bt_train <- as.numeric(train_rt$bestTune[which.min(train_rt$bestTune)])
```

```
## [1] "cp=0.00707"
```

```r
#Prune the regression tree using the best cp parameter; predict against test set
rt_pruned <- prune(rt_fit, cp=bt_train)

y_hat_pruned <- predict(rt_pruned, test_set)

sqrt(mean((y_hat_pruned - test_set$heart_disease_mortality_per_100k)^2))
```

```
## [1] 40.75769
```

As we can see, the RMSE for the regression tree that trains/fits the entire data set is superior to that of the train/test (and superior to the simple linear regression model), but that is likely due to overfitting.

Random Forest

randomForest/rf

For the first attempt using a random forest prediction algorithm, we will apply the "randomForest" package against the entire dataset.

```
library(randomForest)
train_rf <- randomForest(heart_disease_mortality_per_100k ~ ., data=model_data)

sqrt(mean((train_rf$predicted - model_data$heart_disease_mortality_per_100k)^2))
```

```
## [1] 31.56144
```

This is the best RMSE so far, but since the model was fitted over the entire dataset, it is likely to suffer from overfitting when run against an unknown dataset.

We the run the same package, but this time with our train dataset, then preditcing against the test set.

```
train_rf_full <- randomForest(x=train_set[model_cols],
                              y=train_set$heart_disease_mortality_per_100k,
                              xtest = test_set[model_cols]
                              )
sqrt(mean((train_rf_full$test$predicted - test_set$heart_disease_mortality_per_100k)^2))
```

```
## [1] 33.49754
```

As expected, the RMSE is a little higher than the forest fit against the entire dataset.

The tuning parameter available in the caret package for the randomForest algorithm (method="rf") is the mtry parameter, which specifies how many predictors are randomly selected as candidates at each split of the tree. We will train the rf model over mtry values from 1 thru 11.

```
fit_rf_full <- train(method="rf",
                     x=train_set[model_cols],
                     y=train_set$heart_disease_mortality_per_100k,
                     tuneGrid = data.frame(mtry = seq(1,11)))

y_hat_rf_full <- predict(fit_rf_full$finalModel, test_set)

sqrt(mean((y_hat_rf_full - test_set$heart_disease_mortality_per_100k)^2, na.rm=TRUE))
```

```
## [1] 33.605
```

```
fit_rf_full$bestTune$mtry
```

```
## [1] "mtry=2"
```

The trained randomForest with an mtry parameter of 2 has very little difference from the untrained randomForest (which by default has a mtry parameter of $floor(\frac{ncol(train\_set)}{3}) = 4$)

Rborist

Another random forest algorithm that can be trained with the caret package is Rborist. It has 2 tunable parameters:

- predFixed - the number of predictors to randomly select at each split
- minNodes - the mininmum number of members (size) of each final node

We will just train Rborist against the train and test sets, since we have seen in all cases that we are liklely introducing overfitting when using the entire dataset to predict. We will use ranges of predFixed from 2 thru 5, and minNodes as even numbers from 2 thru 50.

```
library(Rborist)
tGrid = rbind(data.frame(predFixed=2, minNode = seq(2,50, by=2)),
              data.frame(predFixed=3, minNode = seq(2,50, by=2)),
              data.frame(predFixed=4, minNode = seq(2,50, by=2)),
              data.frame(predFixed=5, minNode = seq(2,50, by=2)))

fit_rf <- train(heart_disease_mortality_per_100k ~ .,
                method="Rborist",
                tuneGrid = tGrid,
                data=train_set)

y_hat_rf <- predict(fit_rf, test_set)
sqrt(mean((y_hat_rf-test_set$heart_disease_mortality_per_100k)^2))
```

```
## [1] 34.32222
```

```
fit_rf$bestTune
```

```
## [1] "predFixed=4 minNodes=2"
```

With 4 randomly selected predictors and a minimum node size of 2, we get a RMSE slightly worse than the randomForest algorithm.

Ensemble method

By averaging the predictions of multiple models, we may be able to get better predictive power than any one model. To see if this is the case with our dataset, we will use the results of 4 models:

- linear regression
- Regression Tree (pruned off of train_set)
- randomForest (mtry = 2)
- Rborist (predFixed = 4; minNodes = 2)

```
#Put all predictions together
ens <- data.frame(lm = y_hat, rt = y_hat_pruned, rf = y_hat_rf_full, Rborist = y_hat_rf)

#Get the mean predicted value across 4 models
ens$ensemble <- rowMeans(ens)

#Calculate RMSE
ens_RMSE <- sqrt(mean((ens$ensemble - test_set$heart_disease_mortality_per_100k)^2))
```

```
## [1] 34.93187
```

The results of the 4-model ensemble shows an RMSE lower than the linear regression and regression tree models, but higher than both of the random forest models. We will remove the constituent model with the highest RMSE (regression tree) and run an ensemble for the remaining 3.

```
ens3 <- data.frame(lm = y_hat, rf = y_hat_rf_full, Rborist = y_hat_rf)

ens3$ensemble=rowMeans(ens3)

sqrt(mean((ens3$ensemble - test_set$heart_disease_mortality_per_100k)^2))
```

```
## [1] 34.21603
```

The 3-model ensemble RMSE is still higher than the rf model. One final ensemble to look at would be using just the 2 trained random forest models.

```
ens_rf <- data.frame(rf = y_hat_rf_full, Rborist = y_hat_rf)

ens_rf$ensemble=rowMeans(ens_rf)

ens_rf_RMSE <- sqrt(mean((ens_rf$ensemble - test_set$heart_disease_mortality_per_100k)^2))
```

```
## [1] 33.8959
```

The RMSE for the mean prediction in our random forest models is in between the RMSEs of the two consituent models. This suggests that the nodes within each forest are fairly similar, and our predictive power does not increase by using multiple sets of predictions.

Summary of modelling

method	RMSE	TrainVal
randomForest trained	33.60500	mtry=2
random forest ensemble	33.89590	N/A
3 model (lm, rf, Rbor) ensemble	34.21603	N/A
trained Rborist	34.32222	predFixed=4 minNodes=2
4 model ensemble	34.93187	N/A
lm	36.61253	N/A
Reg Tree − pruned; train/test	40.75769	cp=0.00707

Conclusions

Predicting heart disease mortality rate is a challenging endeavor, as there are many "environmental" (i.e., not necessarily specific to an individual such as genetics) factors that can affect the outcome. Within the given data set, we see that heart disease mortality can be predicted reasonably well using a trained *randomForest* model using the following features:

- Economic Typology
- Population
- Air Pollution concentration
- Percent of Civilian Labor
- Percent non-Hispanic African American
- Death Rate per 1,000
- Percent of adults with less than a High School diploma
- Percent of adults with Bachelor's Degree or higher
- Percent of adults that are obese and have diabetes
- Percent of adults that are obese and are physically inactive
- Percent of adults that have diabetes and are physically inactive
- Percent of adults that are obese, have diabetes and are physically inactive