Cancer Mortality Exploration

Andrew Carlson, Brandon Cummings, Tako Hisada

Background

In this lab, imagine that your team is hired by a health government agency. They would like to understand factors that predict cancer mortality rates, with the ultimate aim of identifying communities for social interventions, and of understanding which interventions are likely to have the most impact. Your team was hired to perform an exploratory analysis to help the agency address their goals.

Data

You are given a dataset for a selection of US counties, **"cancer.csv"**. The dependent (or target) variable in this data is named "deathRate".

The labels of some of the variables are listed below; the rest of the variables should be self-explanatory.

avgAnnCount:	"2009-2013 mean incidences per county"
povertyPercent:	"Percent of population below poverty line"
popEst2015:	"Estimated population by county 2015"
PctPrivateCoverage:	"Percentage of the population with private insurance coverage"
PctPublicCoverage:	"Percentage of the population with public insurance coverage"

Objective

Perform an exploratory analysis to understand how county-level characteristics are related to cancer mortality.

```
setwd('~/Documents/MIDS/W203/hw/Lab_1/Cancer_EDA')
Cancer = read.csv('cancer.csv')
```

The CSV data contains **30 variables** (one which is just an index) with **3047 records** .

```
colnames(Cancer)
```

```
##  [1] "X"                     "avgAnnCount"          "medIncome"
##  [4] "popEst2015"            "povertyPercent"       "binnedInc"
##  [7] "MedianAge"             "MedianAgeMale"        "MedianAgeFemale"
## [10] "Geography"             "AvgHouseholdSize"     "PercentMarried"
## [13] "PctNoHS18_24"          "PctHS18_24"           "PctSomeCol18_24"
## [16] "PctBachDeg18_24"       "PctHS25_Over"         "PctBachDeg25_Over"
## [19] "PctEmployed16_Over"    "PctUnemployed16_Over" "PctPrivateCoverage"
## [22] "PctEmpPrivCoverage"    "PctPublicCoverage"    "PctWhite"
## [25] "PctBlack"              "PctAsian"             "PctOtherRace"
## [28] "PctMarriedHouseholds" "BirthRate"            "deathRate"
```

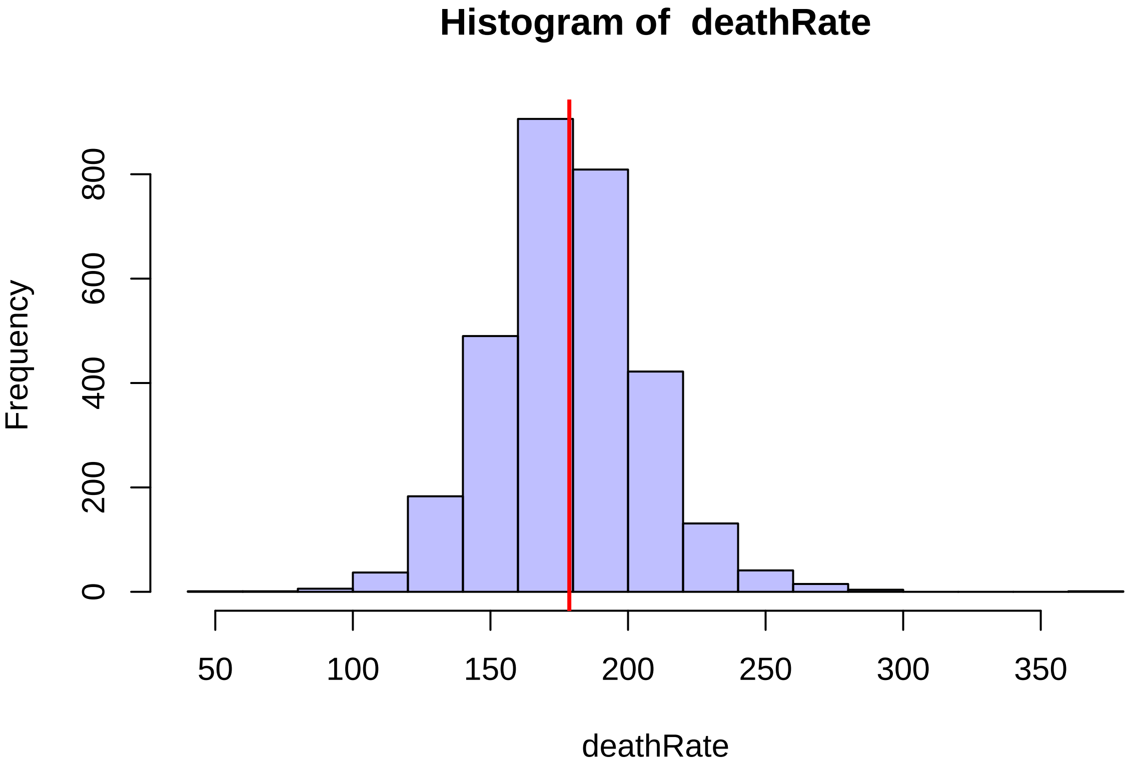
```
nrow(Cancer)
```

```
## [1] 3047
```

```
# convenient wrapper function for a prettier histogram
histWithMean <- function(vec, name) {
  hist(vec, col=rgb(0,0,1,1/4), main=paste("Histogram of ", name), xlab=name)
  # add a red line down the mean
  abline(v = mean(vec, na.rm=TRUE), col="red", lwd=2)
}
```

The dependant variable for this analysis is `deathRate`, which is assumed to be the death rate from cancer.

```
histWithMean(deathRate, "deathRate")
```



Unclean Data

```
# function that counts the number of elements in a vector that satisfy the predicate
# convenient for checking certain sanity bounds and counting how many are out of the bounds
count.by <- function(vec, predicate) {
  yes <- 0
  no <- 0
  for (n in vec) {
    if (predicate(n)) {
      yes <- yes + 1
    } else {
      no <- no + 1
    }
  }
  return(c(yes, no))
}
```

61 of the `AvgHouseholdSize` entries are less than 1. This is probably a coding error. A mean less than 1 for a set of integers is only possible if some values are 0 or negative. These values are nonsensical for a household size.

```
count.by(AvgHouseholdSize, function(num) num < 1)
```

```
## [1]   61 2986
```

30 of the `MedianAge` entries are greater than 200. This seems flagrantly improbable.

```
count.by(MedianAge, function(num) num > 200)
```

```
## [1]   30 3017
```

152 of the `PctEmployed16_Over` entries are `NA`.

```
count.by(PctEmployed16_Over, is.na)
```

```
## [1]  152 2895
```

2285 of the `PctSomeCol18_24` entries are `NA`.

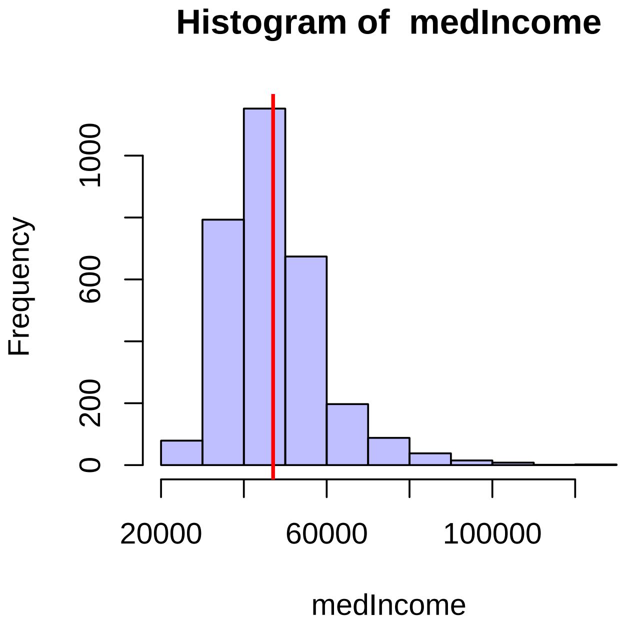
```
count.by(PctSomeCol18_24, is.na)
```

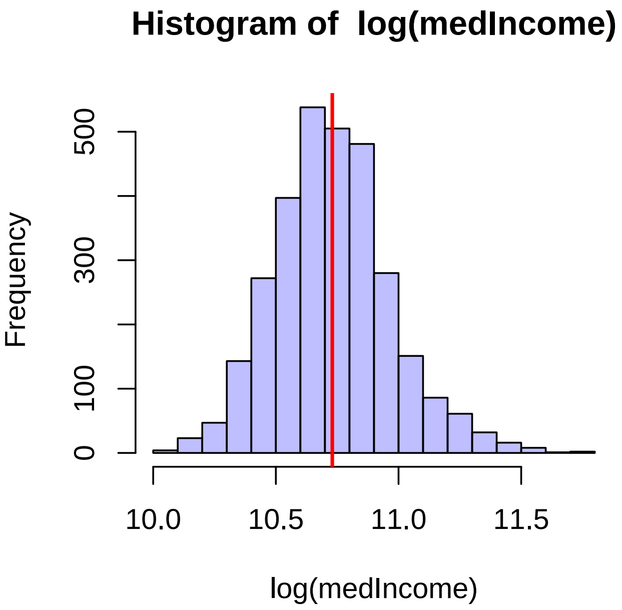
```
## [1] 2285  762
```

Correlated Variables

Here are some histograms of the variables that turned out to be related to `deathRate`.

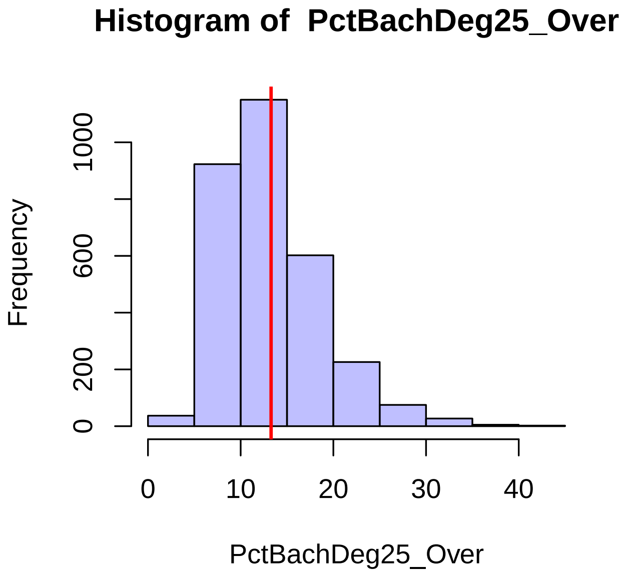
`medIncome` looks like a positively skewed distribution. In fact, in some populations it may look more like a power law distribution than a normal [link]. If we plot `log(medIncome)`, it *looks* closer to a normal distribution. We can check this transformation for correlation with `deathRate` in addition to the plain `medIncome` variable.

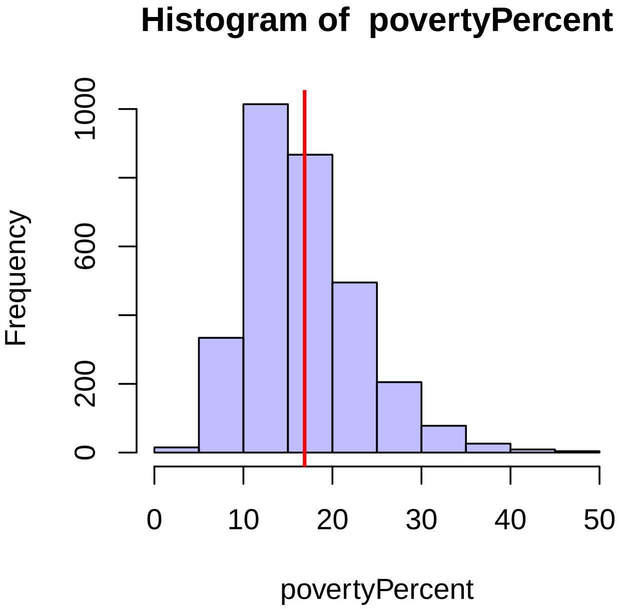


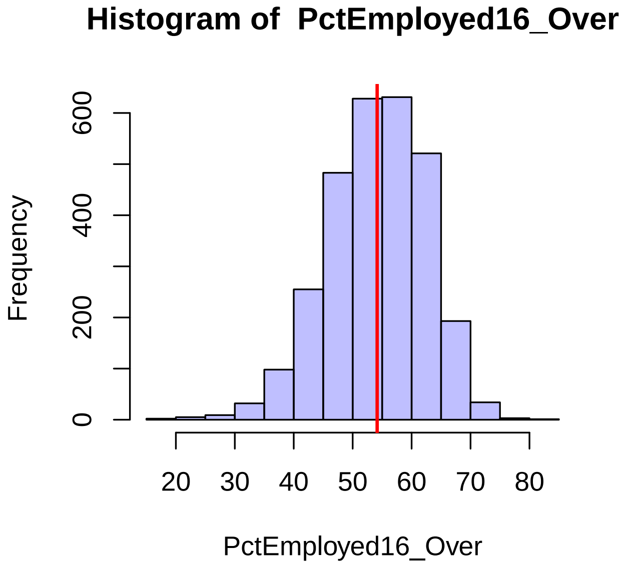


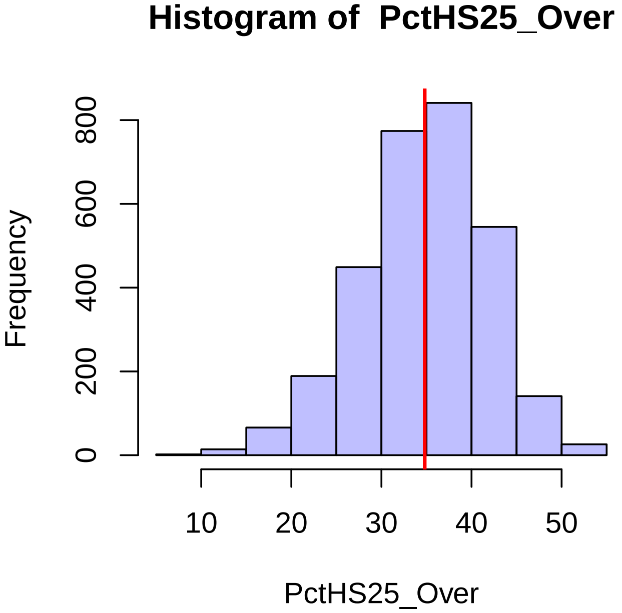
```
Cancer$logMedIncome <- log(medIncome)
```

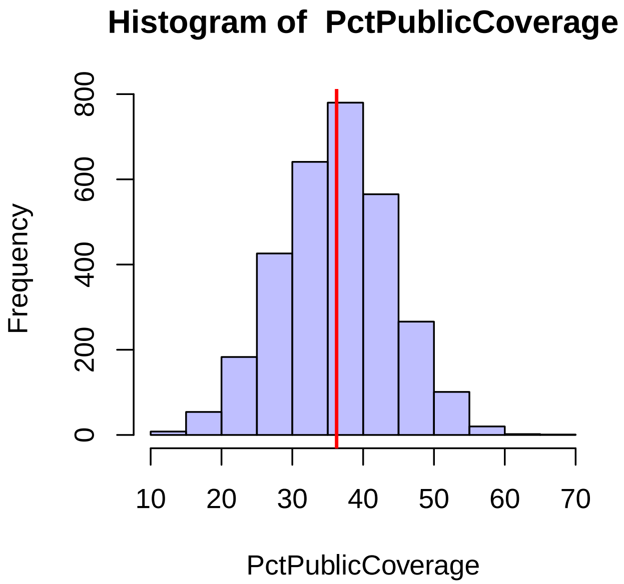
The rest look like clean, valid data. There are no obvious transformations to apply.

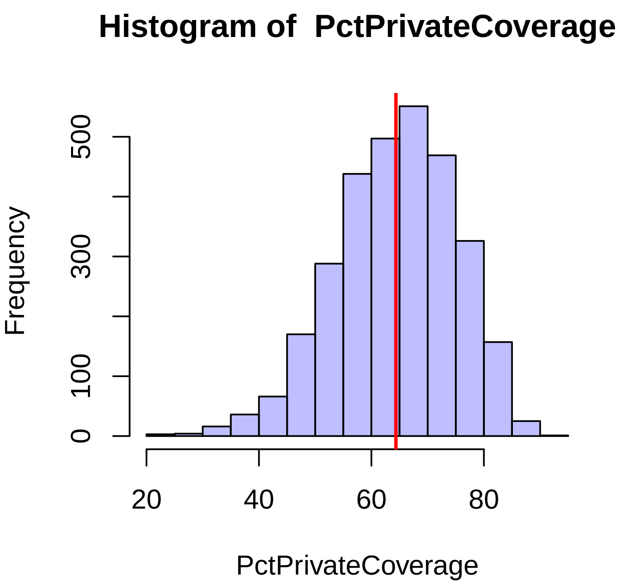

Histogram of PctBachDeg25_Over
Frequency
0
200
600
1000
0
10
20
30
40
PctBachDeg25_Over

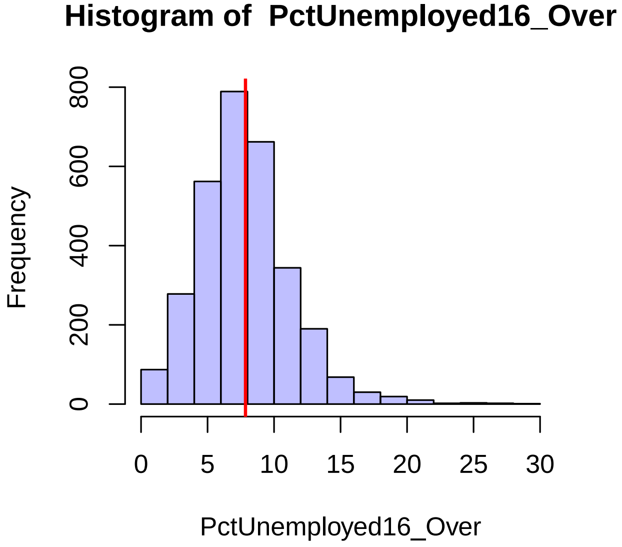

Histogram of povertyPercent
Frequency
0
200
600
1000
0
10
20
30
40
50
povertyPercent


Histogram of PctEmployed16_Over
Frequency
0
200
400
600
20
30
40
50
60
70
80
PctEmployed16_Over


Histogram of PctHS25_Over
Frequency
0
200
400
600
800
10
20
30
40
50
PctHS25_Over







Finding strongest correlations

The numeric variables were taken. The correlation with each numeric variable was calculated and sorted by descending absolute value.

```
# get just the numeric columns
numericColumns <- sapply(Cancer, is.numeric)
NumericCancer <- Cancer[, numericColumns]
# get each correlations with each column
correlations <- apply(NumericCancer, 2, function(col) cor(col, deathRate))
correlations <- correlations[!is.na(correlations)]
```

Now we have a vector of all the correlations. We just filtered out the `NA`s, which includes `PctEmployed16_Over` because some of the entries were `NA`. We'll have to add it back manually after dealing with the `NA`s.

```
# clean the NAs out of PctEmployed16_Over and calculate correlation
cleanPctEmployed16_Over <- !is.na(PctEmployed16_Over)
corPctEmployed16_Over <- cor(PctEmployed16_Over[cleanPctEmployed16_Over], deathRate[cleanPctEmployed16_
# append it to the vector of correlations and name the entry
```

```r
correlations <- c(correlations, corPctEmployed16_Over)
names(correlations)[length(correlations)] <- "PctEmployed16_Over"
```

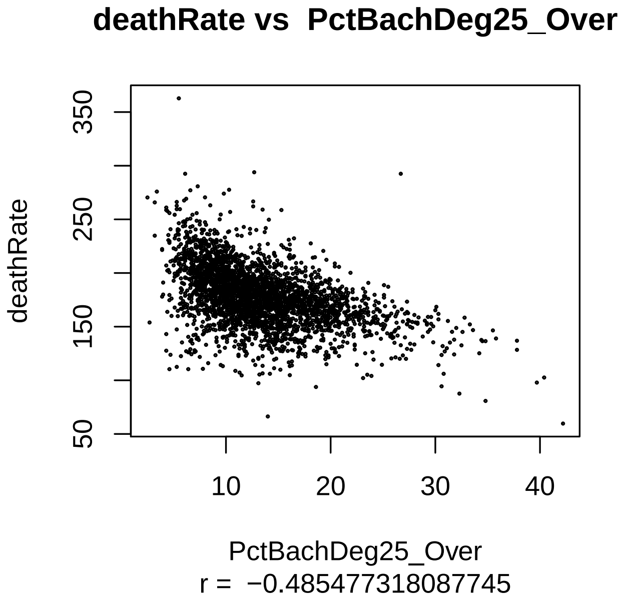
Now we can focus on the correlations that are significant.

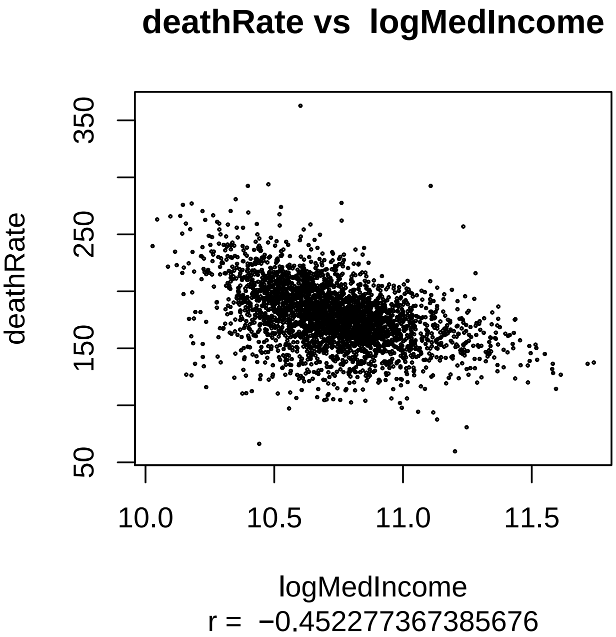
```r
# sort them
correlations <- correlations[order(abs(correlations), decreasing=TRUE)]
correlations <- correlations[2:length(correlations)]
correlations <- correlations[abs(correlations) >= 0.3]
correlations
```

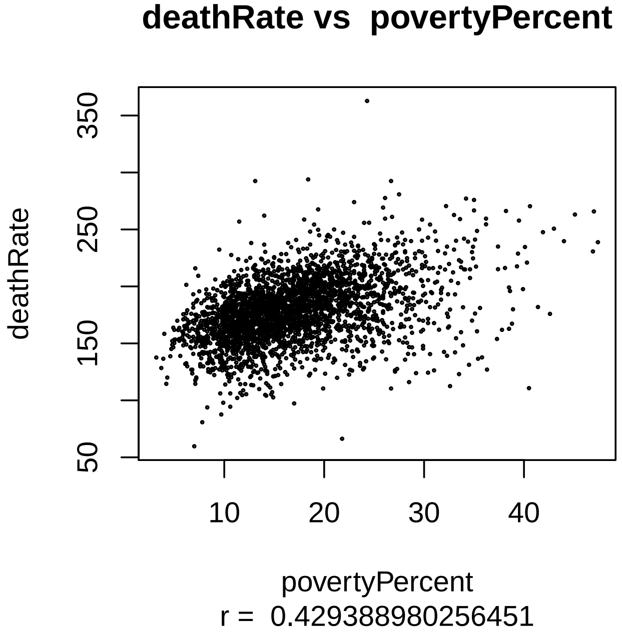
```
##    PctBachDeg25_Over         logMedIncome       povertyPercent 
##           -0.4854773           -0.4522774            0.4293890 
##            medIncome   PctEmployed16_Over          PctHS25_Over 
##           -0.4286149           -0.4120458            0.4045891 
##    PctPublicCoverage   PctPrivateCoverage PctUnemployed16_Over 
##            0.4045717           -0.3860655            0.3784124
```

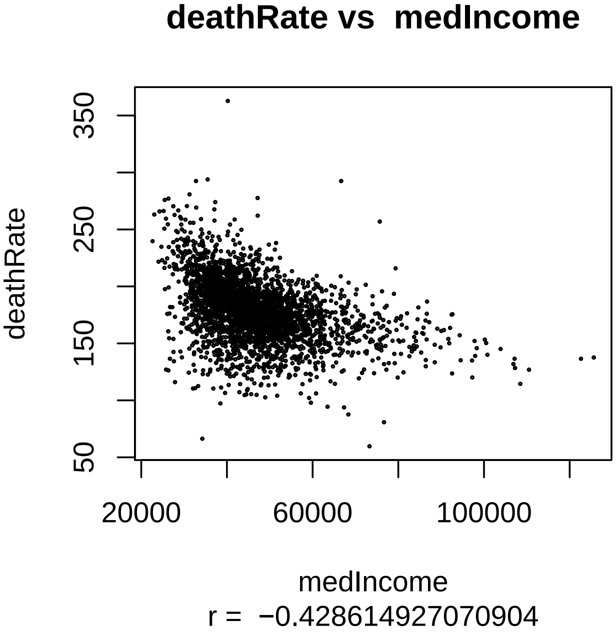
We will consider correlations of `0.3` or greater a significant association. This includes 9 of the variables, one of which is our transformed log(medianIncome). This actually had stronger correlation with `deathRate` than `medIncome`.

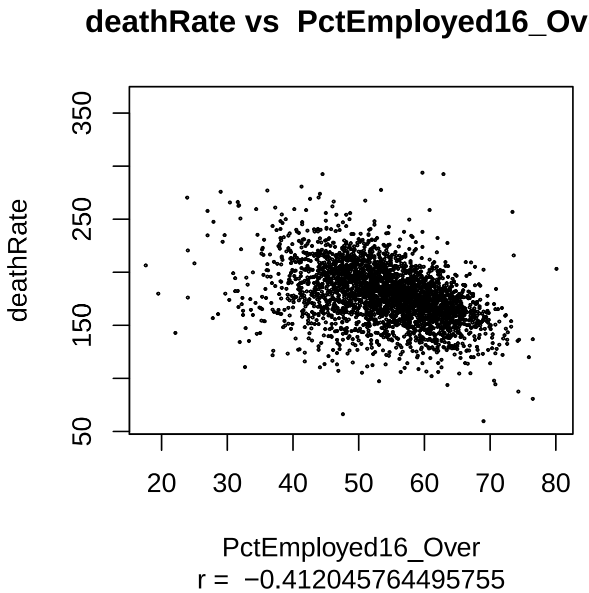
Plot deathRate with all variables with at least a weak correlation





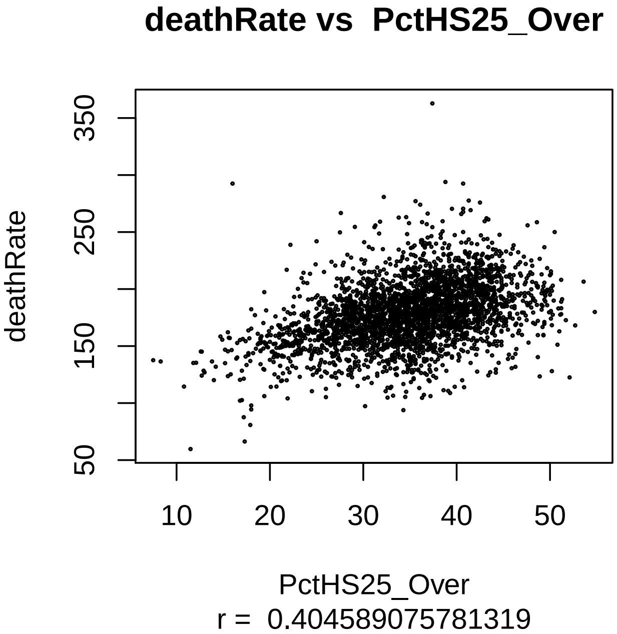






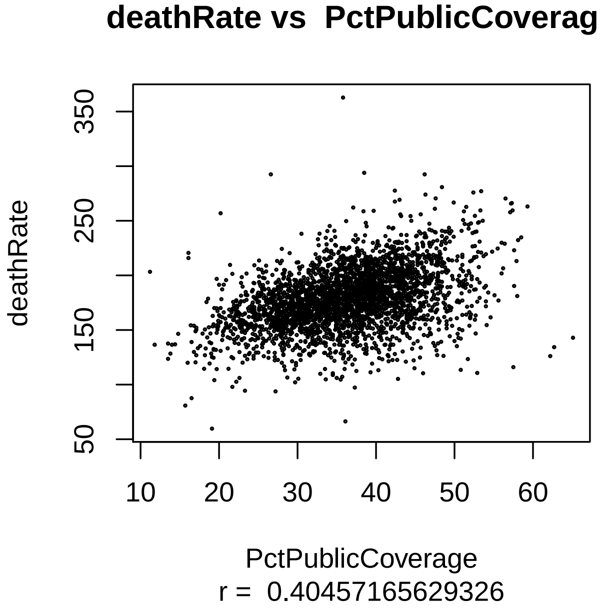
deathRate vs PctEmployed16_Over

r = −0.412045764495755



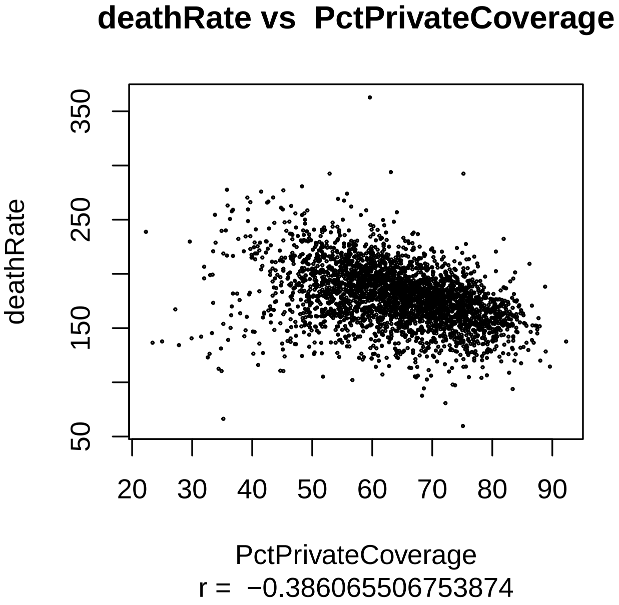
deathRate vs PctHS25_Over

r = 0.404589075781319



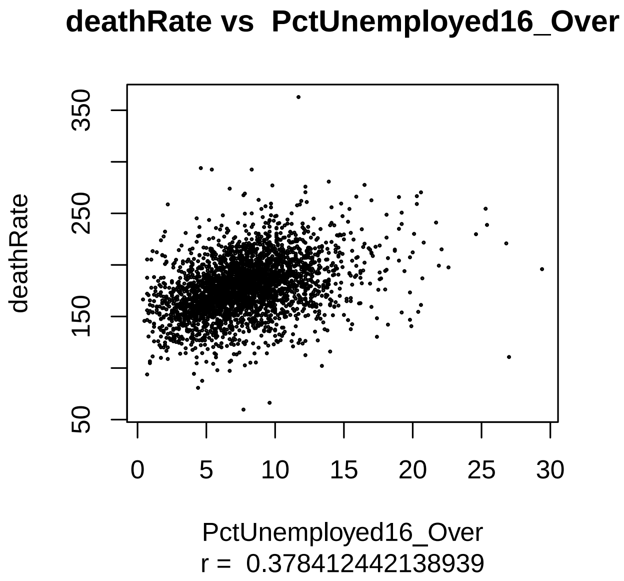
deathRate vs PctPublicCoverage

r = 0.40457165629326



deathRate vs PctPrivateCoverage

r = −0.386065506753874


deathRate vs PctUnemployed16_Over
deathRate
50
150
250
350
0
5
10
15
20
25
30
PctUnemployed16_Over
r = 0.378412442138939