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')
par(mar = rep(2, 4))
```

The CSV data contains **30 variables** (one which is just an index) with **3047 records**. This seems like an adequate number of observations to glean associations between variables.

```
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)
```

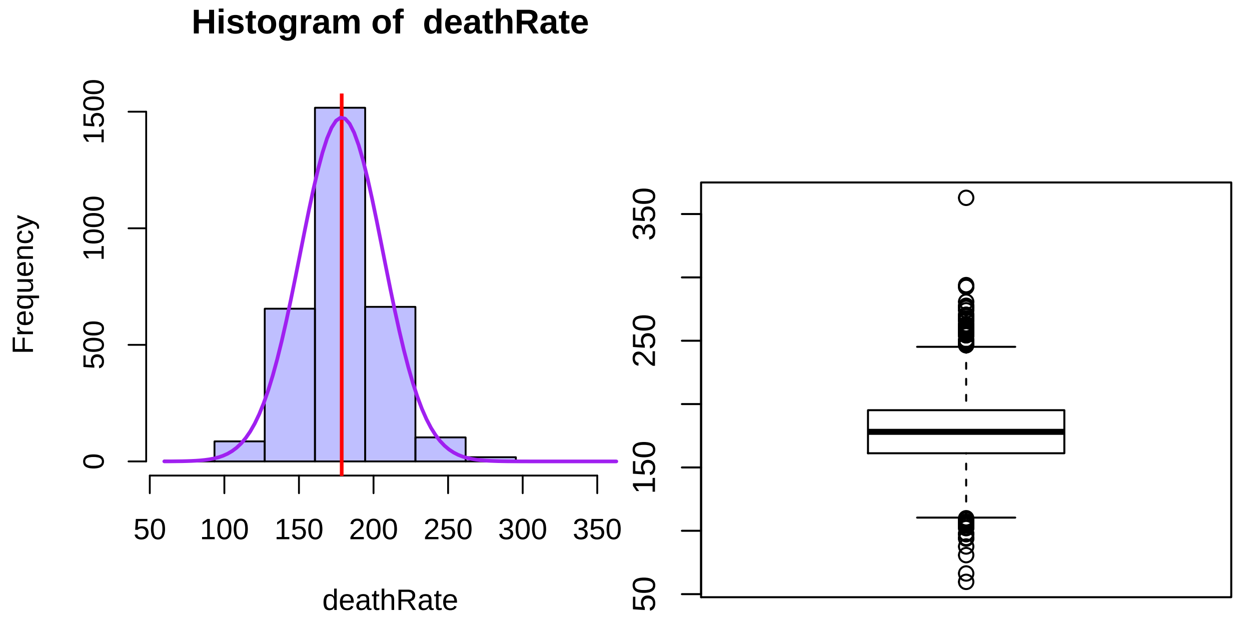
```
## [1] 3047
```

```
# convenient wrapper function for a prettier histogram
histWithNorm <- function(vec, name) {
  # calculate the breaks for the histogram
  vecMin <- min(vec, na.rm=TRUE)
  vecMax <- max(vec, na.rm=TRUE)
  breaks <- seq(vecMin, vecMax, length.out=10)
  vecHist <- hist(vec, col=rgb(0,0,1,1/4), breaks=breaks, main=paste("Histogram of ", name), xlab=name)

  # add a red line down the mean
  vecMean <- mean(vec, na.rm=TRUE)
  abline(v = vecMean, col="red", lwd=2)

  # plot a normal distribution over the histogram to visually compare the distributions
  vecSd <- sd(vec, na.rm=TRUE)
  # create the domain. span 6 sd's centered at the mean
  x <- seq(vecMean - 3 * vecSd, vecMean + 3 * vecSd, length.out=100)
  # get the width between each break
  histWidth <- breaks[2] - breaks[1]
  # calculate the area of the histogram and use it as the scale factor
  scaleFactor <- sum(histWidth * vecHist$counts)
  curve(dnorm(x, mean=vecMean, sd=vecSd) * scaleFactor, add=TRUE, col="purple", lwd=2)
}
```

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



Anomalous Data

We'll count the number of vector elements that violate some constraints to check for anomalies. If entries are missing or clearly erroneous, we can remove them from the data set before calculating the correlation.

```
# 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
```

```
cleanAvgHouseholdSize <- AvgHouseholdSize >= 1
```

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

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

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

```
cleanMedianAge <- MedianAge < 200
```

152 of the PctEmployed16_Over entries are NA.

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

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

```
cleanPctEmployed16_Over <- !is.na(PctEmployed16_Over)
```

2285 of the PctSomeCol18_24 entries are NA. This is most of the rows, but there are still enough for a meaningful association.

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

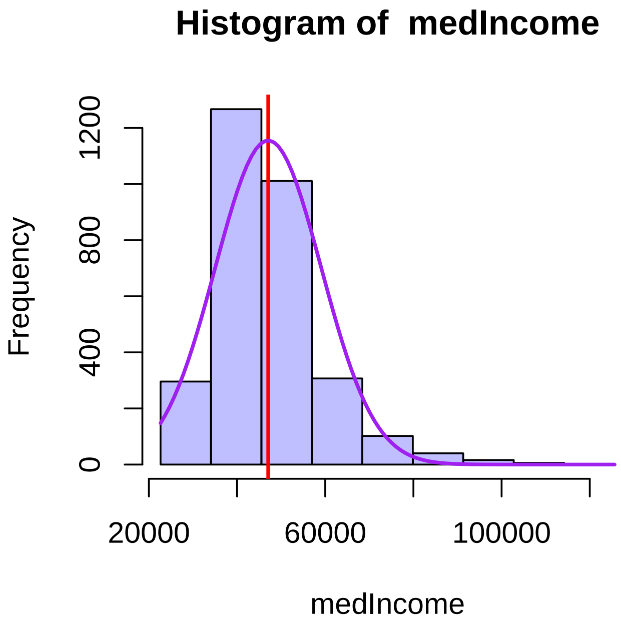
```
## [1] 2285  762
```

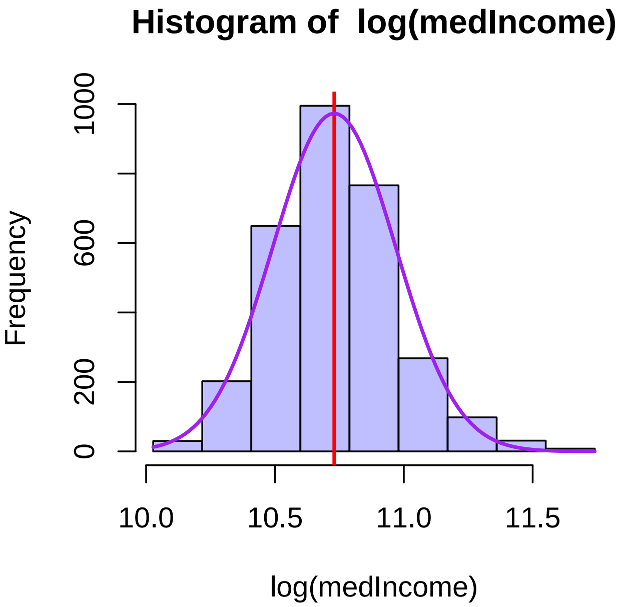
```
cleanPctSomeCol18_24 <- !is.na(PctSomeCol18_24)
```

Key Variables

Here are some histograms of the variables that turned out to be related to `deathRate`. How we determined this in the **correlated variables** section.

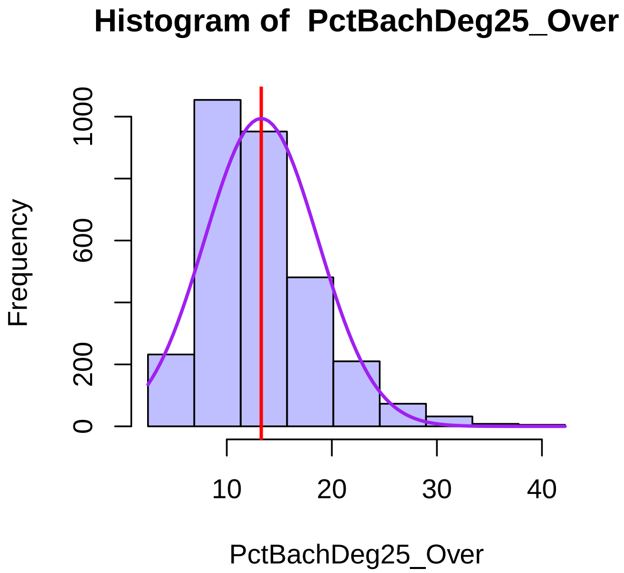
`medIncome` looks like a positively skewed distribution. In fact, in some populations it may look more like a power law distribution than a normal [link]. If we plot `log(medIncome)`, it *looks* closer to a normal distribution. We can check this transformation for correlation with `deathRate` in addition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

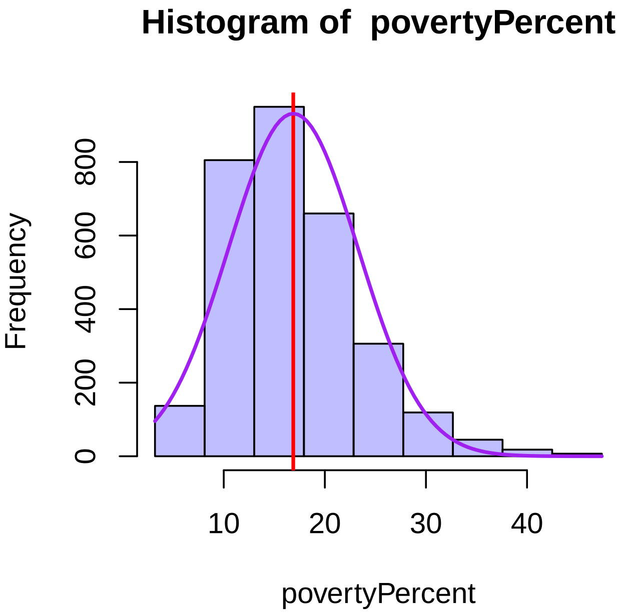




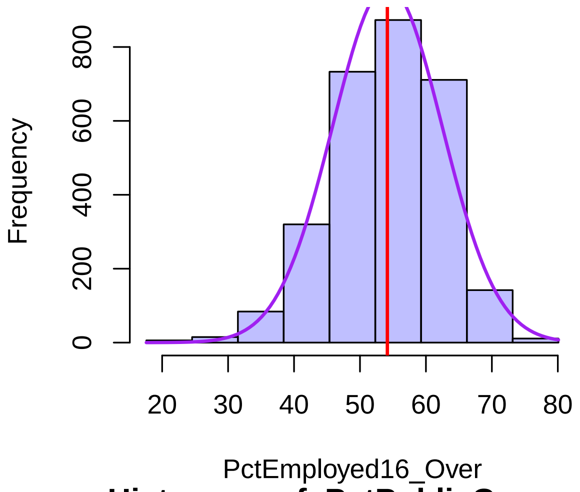
```r
Cancer$logMedIncome <- log(medIncome)
```

The rest look like clean, valid, approximately normally-distributed variables. There are no obvious transformations to apply.

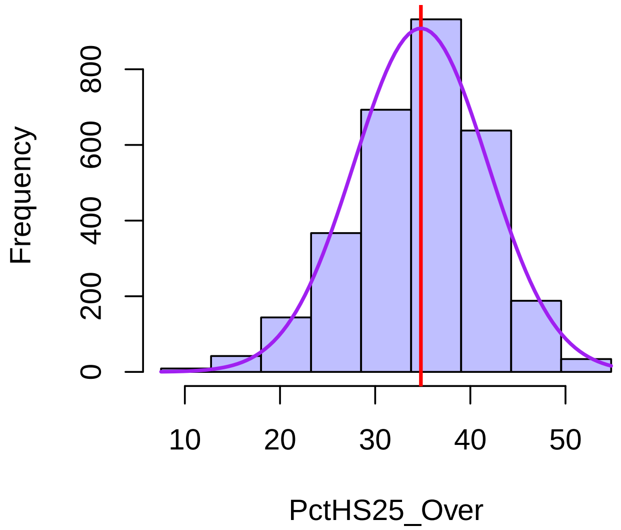




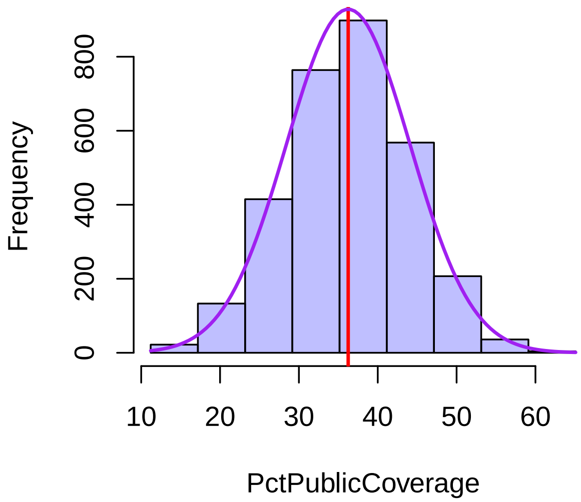
Histogram of PctEmployed16_Over



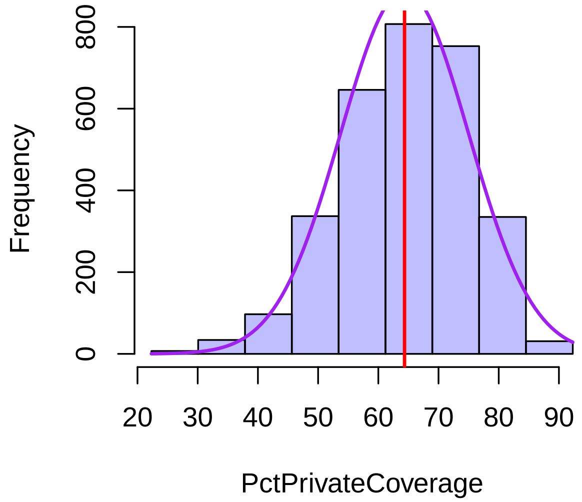
Histogram of PctHS25_Over



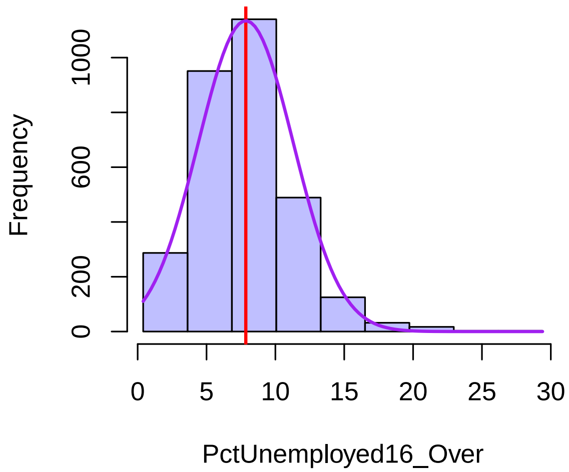
Histogram of PctPublicCoverage



Histogram of PctPrivateCoverage



Histogram of PctUnemployed16_Over



Correlated Variables

The numeric variables were taken. The correlation with each numeric variable was calculated. Some of these variables had NAs, so those will get removed.

```
# 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 anomalous data, which includes PctEmployed16_Over because some of the entries were NA. We'll have to add it back manually after dealing with the NAs.

```
# clean the  out of PctEmployed16_Over and calculate correlation
corPctEmployed16_Over <- cor(PctEmployed16_Over[cleanPctEmployed16_Over],
                             deathRate[cleanPctEmployed16_Over])
# append it to the vector of correlations and name the entry
correlations <- c(correlations, corPctEmployed16_Over)
names(correlations)[length(correlations)] <- "PctEmployed16_Over"

# add the rest of the cleaned variables
corPctSomeCol18_24 <- cor(PctSomeCol18_24[cleanPctSomeCol18_24],
                          deathRate[cleanPctSomeCol18_24])
correlations <- c(correlations, corPctSomeCol18_24)
names(correlations)[length(correlations)] <- "PctSomeCol18_24"

corAvgHouseholdSize <- cor(AvgHouseholdSize[cleanAvgHouseholdSize],
                           deathRate[cleanAvgHouseholdSize])
correlations <- c(correlations, corAvgHouseholdSize)
names(correlations)[length(correlations)] <- "cleanAvgHouseholdSize"

corMedianAge <- cor(MedianAge[cleanMedianAge],
                    deathRate[cleanMedianAge])
correlations <- c(correlations, corMedianAge)
names(correlations)[length(correlations)] <- "cleanMedianAge"
correlations
```

```
##                     X           avgAnnCount             medIncome
##           0.051913500          -0.143531620          -0.428614927
##            popEst2015        povertyPercent             MedianAge
##          -0.120073096           0.429388980           0.004375077
##         MedianAgeMale       MedianAgeFemale      AvgHouseholdSize
##          -0.021929429           0.012048386          -0.036905314
##        PercentMarried          PctNoHS18_24            PctHS18_24
##          -0.266820464           0.088462610           0.261975940
##       PctBachDeg18_24          PctHS25_Over     PctBachDeg25_Over
##          -0.287817410           0.404589076          -0.485477318
##  PctUnemployed16_Over    PctPrivateCoverage    PctEmpPrivCoverage
##           0.378412442          -0.386065507          -0.267399428
##     PctPublicCoverage              PctWhite              PctBlack
##           0.404571656          -0.177399980           0.257023560
##              PctAsian          PctOtherRace  PctMarriedHouseholds
##          -0.186331105          -0.189893571          -0.293325341
##             BirthRate             deathRate          logMedIncome
##          -0.087406970           1.000000000          -0.452277367
##    PctEmployed16_Over       PctSomeCol18_24 cleanAvgHouseholdSize
##          -0.412045764          -0.188687667          -0.034641021
##        cleanMedianAge
##          -0.004288054
```

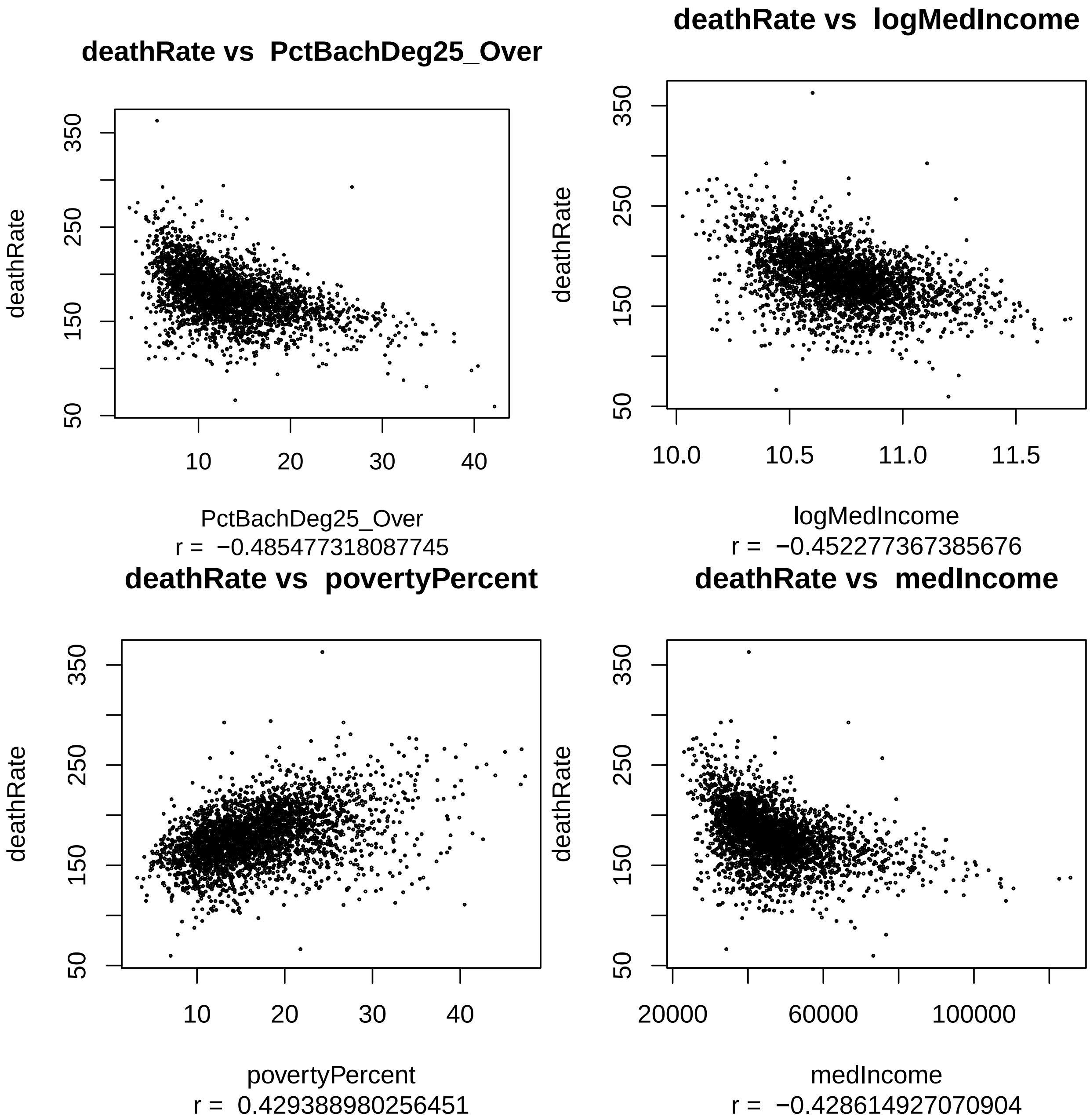
Now we can determine the correlations that are significant. We'll sort these by descending order of absolute value.

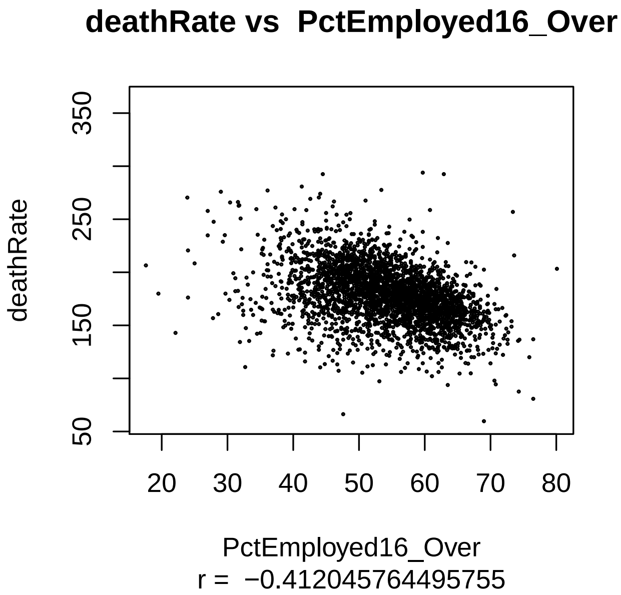
```
# sort them
correlations <- correlations[order(abs(correlations), decreasing=TRUE)]
# remove the cor of deathRate with itself, which is 1, and always the first element after sorting
correlations <- correlations[2:length(correlations)]
correlations <- correlations[abs(correlations) >= 0.3]
correlations
```

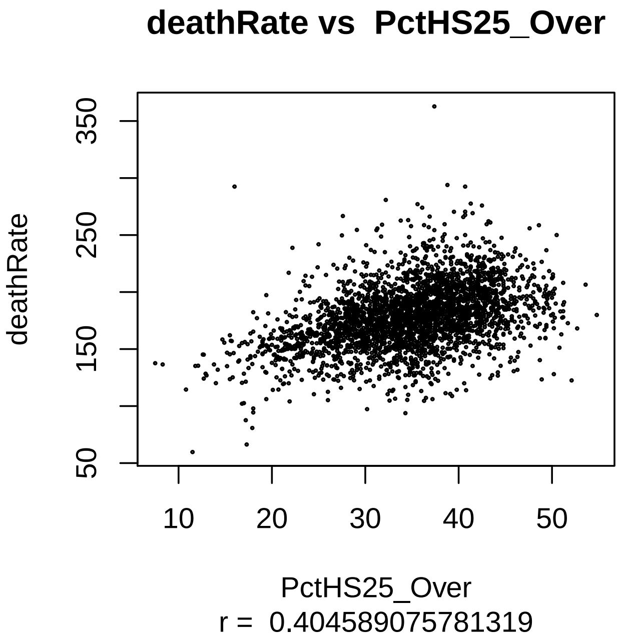
```
##    PctBachDeg25_Over         logMedIncome       povertyPercent
##           -0.4854773           -0.4522774            0.4293890
##            medIncome   PctEmployed16_Over         PctHS25_Over
##           -0.4286149           -0.4120458            0.4045891
##    PctPublicCoverage   PctPrivateCoverage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

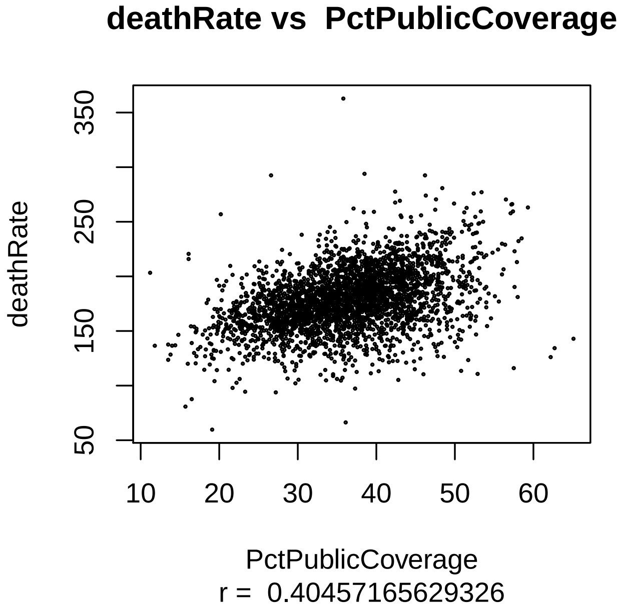
We will consider correlations of `0.3` or stronger a significant association. This includes 9 of the variables, one of which is our transformed log(medianIncome). This actually had stronger correlation with `deathRate` than `medIncome`.

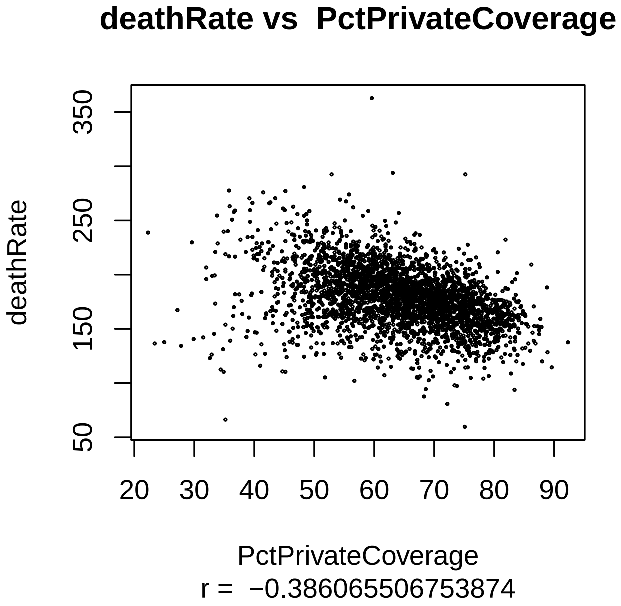
Plot Bivariate Associations

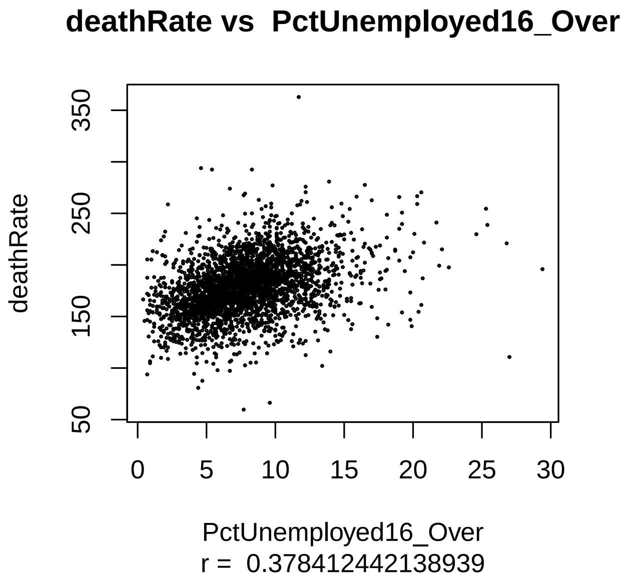



deathRate vs PctEmployed16_Over
deathRate
50
150
250
350
20
30
40
50
60
70
80
PctEmployed16_Over
r = −0.412045764495755


deathRate vs PctHS25_Over
deathRate
50
150
250
350
10
20
30
40
50
PctHS25_Over
r = 0.404589075781319


deathRate vs PctPublicCoverage
deathRate
50
150
250
350
10
20
30
40
50
60
PctPublicCoverage
r = 0.40457165629326


deathRate vs PctPrivateCoverage
deathRate
50
150
250
350
20
30
40
50
60
70
80
90
PctPrivateCoverage
r = −0.386065506753874



These variables have a positive correlation, meaning counties in this set which have higher values are more likely to have a higher `deathRate`: `povertyPercent, PctHS25_Over, PctPublicCoverage, PctUnemployed16_Over`. The rest of the variables are associated with lower `deathRate`: `PctBachDeg25_Over, logMedIncome, medIncome, PctEmployed16_Over, PctPrivateCoverage`.

Confounding Variables

The correlations can be summarized as such: employment, education, and wealth are linked to lower cancer mortality. We can speculate about the underlying causality between some of the variables. For example, suppose the correlation with `PctBachDeg25_Over` is because educated people are aware of cancer causes and choose to avoid those causes. Education also enables people to make more money. `PctBachDeg25_Over` would therefore confound the association between `medIncome` and `deathRate`. However, suppose it's actually income that lowers cancer mortality because people can afford better treatment. In that case, its `medIncome` that confounds the association of `PctBachDeg25_Over` with `deathRate` because the wealthy are more likely to afford tuition for school. A similar statement can be made about the insurance variables.