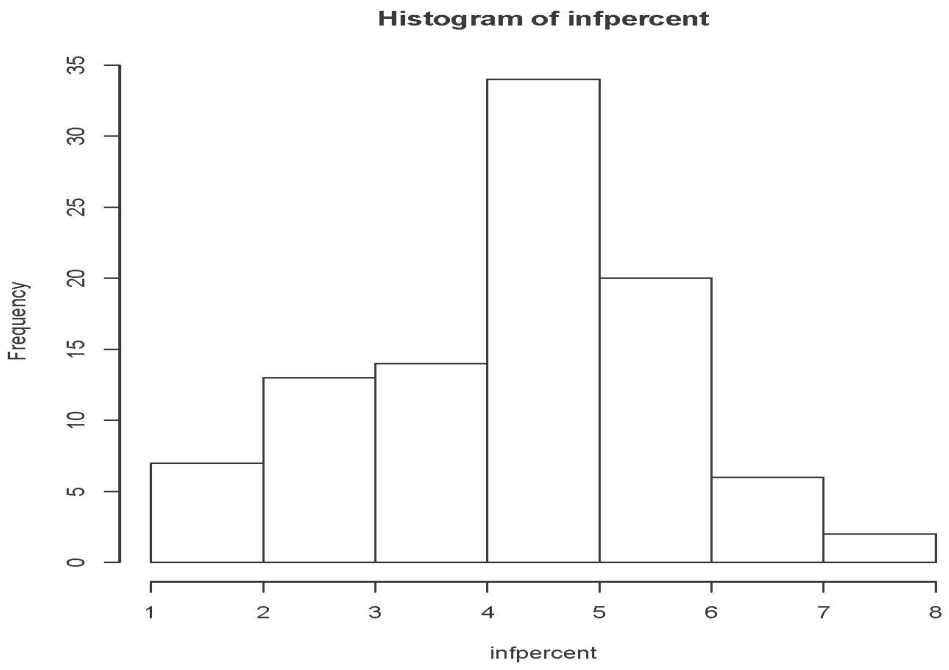
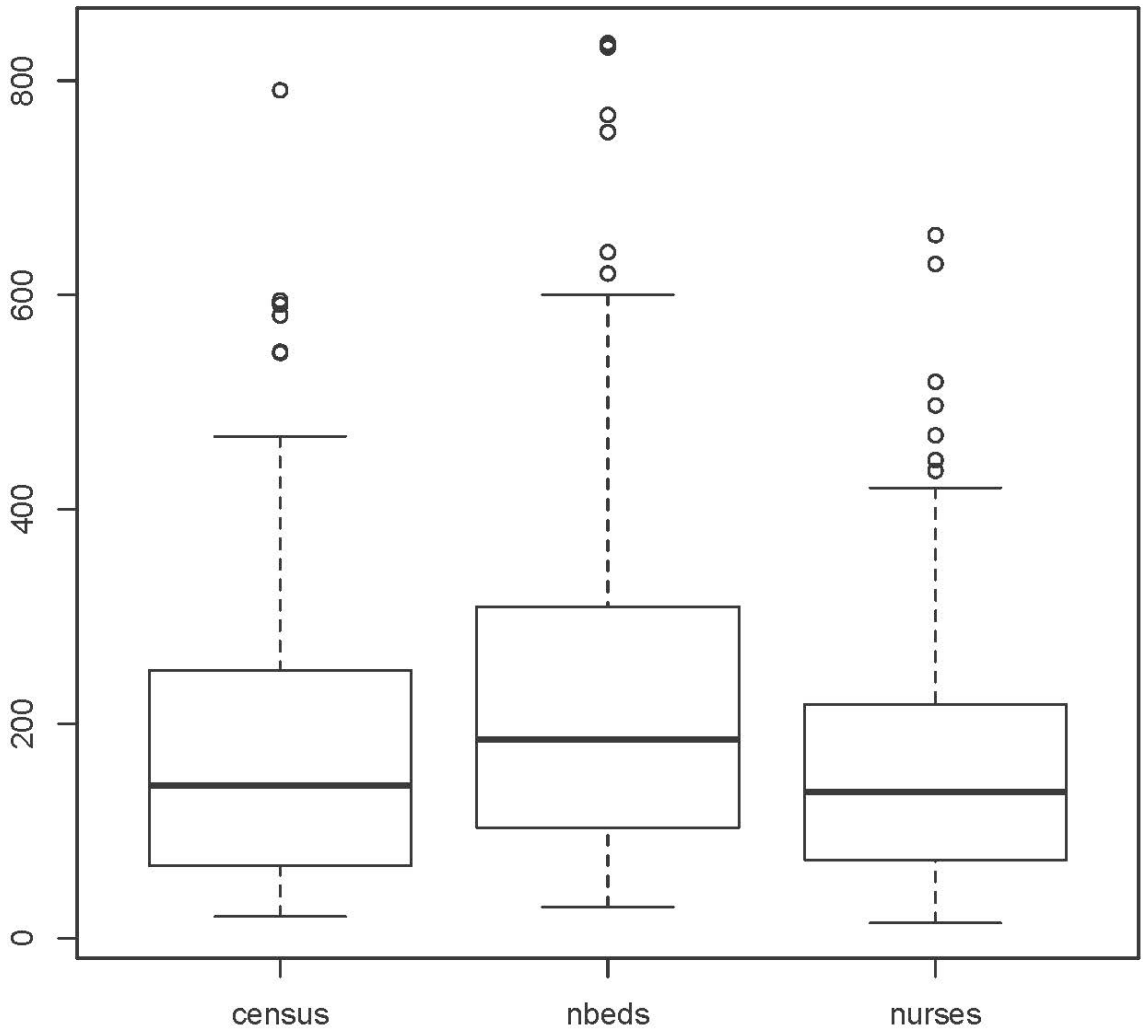
Analysis of the SENIC data with R

```
> senic =
read.table("http://www.utstat.toronto.edu/~brunner/data/legal/openSENIC.data.txt")

> head(senic)
     region mdschl census nbeds nurses lngstay  age xratio culratio infpercent
1 Northeast     No    237   298    115   12.01 52.8   96.9      10.8        4.8
2 Northeast    Yes    144   184    151   10.05 52.0   87.5      36.7        4.5
3 Northeast     No    127   165    158    9.36 54.1   90.6      18.3        4.8
4 Northeast    Yes    240   270    198    9.78 52.3   95.9      17.6        5.0
5       West     No     51    76     79    6.70 48.6   80.8      13.0        4.5
6      South     No     59    95     56    8.93 56.0   72.5       6.2        2.0
> summary(senic)
          region     mdschl       census           nbeds           nurses
 NorthCentral:31   No  :80   Min.   : 20.00   Min.   : 29.0   Min.   : 14.0
 Northeast   :24   Yes :16   1st Qu.: 67.75   1st Qu.:104.5   1st Qu.: 73.0
 South       :30   NA's: 4   Median :142.00   Median :185.0   Median :136.0
 West        :15             Mean   :192.41   Mean   :252.9   Mean   :175.4
                             3rd Qu.:249.00   3rd Qu.:307.5   3rd Qu.:218.0
                             Max.   :791.00   Max.   :835.0   Max.   :656.0
                                                              NA's   :3
    lngstay           age             xratio          culratio        infpercent
 Min.   : 6.700   Min.   :38.80   Min.   : 39.60   Min.   : 1.600   Min.   :1.300
 1st Qu.: 8.325   1st Qu.:51.00   1st Qu.: 70.55   1st Qu.: 8.325   1st Qu.:3.400
 Median : 9.515   Median :53.20   Median : 82.15   Median :13.050   Median :4.450
 Mean   : 9.692   Mean   :53.21   Mean   : 81.62   Mean   :15.630   Mean   :4.314
 3rd Qu.:10.723   3rd Qu.:56.02   3rd Qu.: 94.55   3rd Qu.:19.400   3rd Qu.:5.225
 Max.   :19.560   Max.   :65.90   Max.   :133.50   Max.   :60.500   Max.   :7.800
                                                                    NA's   :4      >
attach(senic) # Make variable names accessible
>
> # Histograms and boxplots of quantitative variables
> hist(infpercent)
```

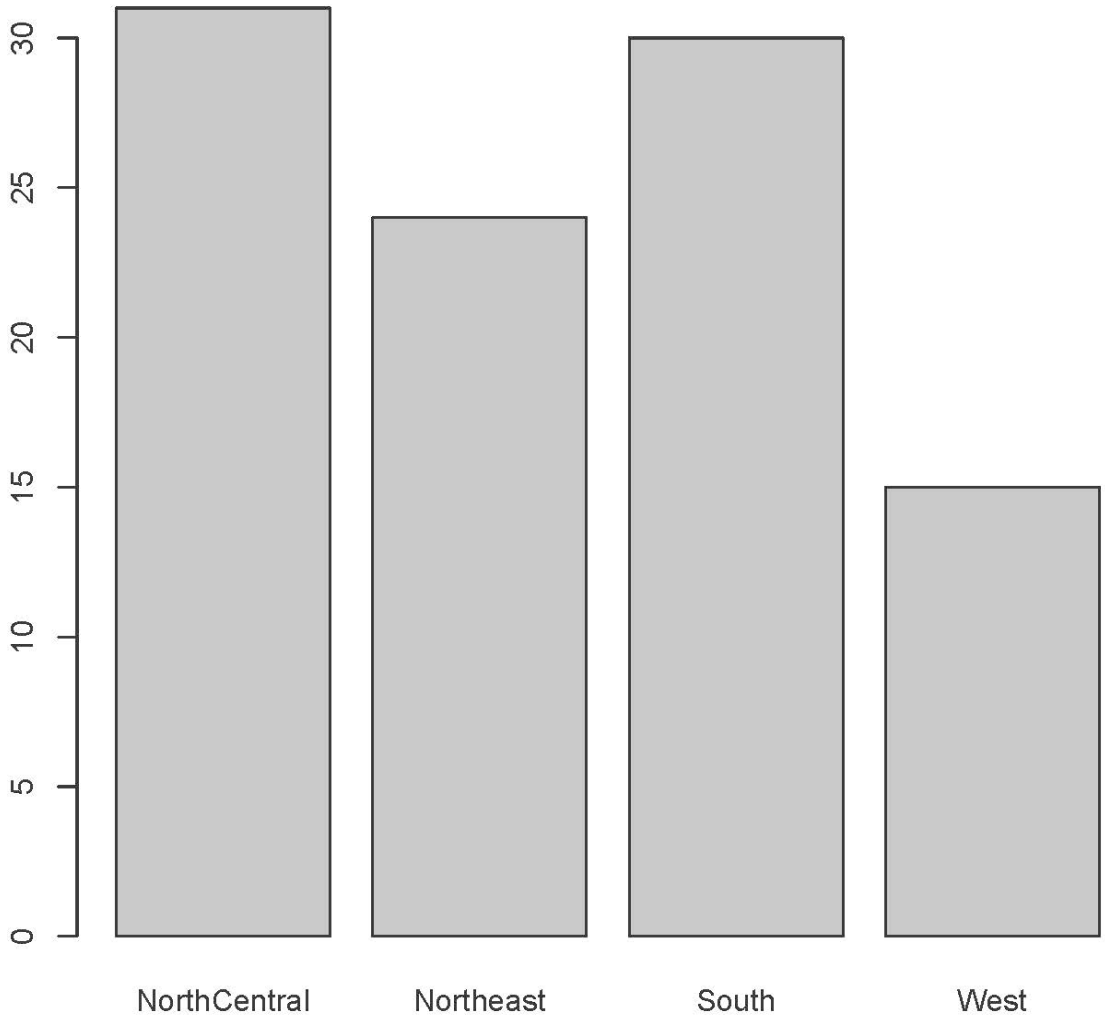


```
> # Histograms and boxplots of quantitative variables
> hist(infpercent)
>
> hist(census)
> hist(nbeds)
> hist(nurses)
> hist(lngstay)
> hist(age)
> hist(xratio)
> hist(culratio)
>
> boxplot(cbind(census,nbeds,nurses))
>
```



```
> boxplot(lngstay)
> boxplot(age)
> boxplot(cbind(xratio,culratio))
> boxplot(infpercent)
>
> # Suppose I just want a table of means, standard deviations and sample sizes.
> # Programming it is a tedious chore.
> # install.packages("tables") # Only need to do this once
> library(tables) # Loads the package -- must do this every time.
Loading required package: Hmisc
Loading required package: lattice
Loading required package: survival
Loading required package: Formula
Loading required package: ggplot2

This data.table install has not detected OpenMP support. It will work but slower in
single threaded mode.


Attaching package: 'Hmisc'

The following objects are masked from 'package:base':

    format.pval, round.POSIXt, trunc.POSIXt, units

>
> # In the tabular syntax, + means stick it together
> # Row descriptions are on the left of ~
> tabular( mean + sd ~ census+nbeds+nurses+lngstay+age+xratio+culratio+infpercent )

      census nbeds nurses lngstay age    xratio culratio infpercent
 mean 192.4  252.9 NA     9.692   53.207 81.62  15.63    NA
 sd   158.2  198.0 NA     1.991    4.444 19.86  10.47    NA
>
> # Want to base mean and sd on non-missing cases, and display number of non-missing
cases.
> # Listwise deletion would be another option. Use data = na.omit(senic)
>
> Mean = function(x) mean(x, na.rm=T) # New function Mean omits NA values
> SD  = function(x) sd(x, na.rm=T)
> N = function(x) sum(!is.na(x)) # T and F are coerced to 1 and 0
>
> tabular( Mean+SD+N ~ census+nbeds+nurses+lngstay+age+xratio+culratio+infpercent )

      census nbeds nurses lngstay age     xratio culratio infpercent
 Mean 192.4  252.9 175.4    9.692  53.207  81.62  15.63     4.314
 SD   158.2  198.0 138.1    1.991   4.444  19.86  10.47     1.352
 N    100.0  100.0  97.0  100.000 100.000 100.00 100.00   96.000
>
> # If you don't like the decimal places on N
> tabular( Mean+SD+(Format(digits=0)*N) ~ census+nbeds+nurses +
lngstay+age+xratio+culratio+infpercent )

      census nbeds nurses lngstay age    xratio culratio infpercent
 Mean 192.4  252.9 175.4  9.692   53.207 81.62  15.63    4.314
 SD   158.2  198.0 138.1  1.991    4.444 19.86  10.47    1.352
 N    100    100    97    100     100    100    100       96
>
```

```
> # Frequency distributions and bar graphs of categorical variables
> table(region)
region
NorthCentral    Northeast        South         West
          31           24           30           15
> # There are 100 hospitals, but in general you would want proportions as well as
> # frequency counts.
> regionfreq = table(region)
> regionprop = prop.table(regionfreq) # Proportions
> rbind(regionfreq,regionprop)
           NorthCentral Northeast South  West
regionfreq        31.00     24.00  30.0 15.00
regionprop         0.31      0.24   0.3  0.15
> barplot(regionfreq) # Bar plot for factors, histogram for numeric variables
```



```
> mdschlfreq = table(mdschl)
> mdschlprop =  prop.table(mdschlfreq)
> rbind(mdschlfreq,mdschlprop) # Note table automatically excludes na
                   No        Yes
mdschlfreq 80.0000000 16.0000000
mdschlprop  0.8333333  0.1666667
>
> # That was fairly ugly.
> mdschlfreq; round(mdschlprop,2) # Display as two separate tables may be preferable
mdschl
 No Yes
 80  16
mdschl
  No  Yes
0.83 0.17
>
> # To include missing values in the table
> table(mdschl,useNA='always')
mdschl
  No  Yes <NA>
  80   16    4
> prop.table(table(mdschl,useNA='always'))
mdschl
  No  Yes <NA>
0.80 0.16 0.04

> # Relationship between region and medical school affiliation
> twoway = table(mdschl,region); twoway
      region
mdschl NorthCentral Northeast South West
   No            24        19    24   13
   Yes            6         5     3    2
> prop.table(twoway,margin=2) # Proportions of column (2nd dimension) totals
      region
mdschl NorthCentral Northeast     South      West
   No     0.8000000 0.7916667 0.8888889 0.8666667
   Yes    0.2000000 0.2083333 0.1111111 0.1333333
> chisq.test(twoway) # Pearson chi-squared test of independence

      Pearson's Chi-squared test

data:  twoway
X-squared = 1.26, df = 3, p-value = 0.7387

Warning message:
In chisq.test(twoway) : Chi-squared approximation may be incorrect
> # The warning seems to be about expected frequencies less than 5
> chisq.test(twoway)$expected # Look at (estimated) expected frequencies
      region
mdschl NorthCentral Northeast South West
   No            25        20  22.5 12.5
   Yes            5         4   4.5  2.5
Warning message:
In chisq.test(twoway) : Chi-squared approximation may be incorrect
```

```
> # T-test: Less risk at Hospitals with Med School Affiliation?
> t.test(infpercent ~ mdschl, var.equal = T)

      Two Sample t-test

data:  infpercent by mdschl
t = -2.542, df = 91, p-value = 0.01271
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.7215138 -0.2112167
sample estimates:
 mean in group No mean in group Yes
         4.205063          5.171429

> tabular(mdschl ~ infpercent*(Mean+SD+N))

        infpercent
 mdschl Mean       SD    N
 No     4.205      1.331 79
 Yes    5.171      1.182 14
>
> # Regional differences in average infection risk?
> tabular(region ~ infpercent*(Mean+SD+N))

             infpercent
 region      Mean       SD    N
 NorthCentral 4.359      1.442 27
 Northeast    4.833      1.336 24
 South        3.820      1.354 30
 West         4.387      0.907 15
> summary(aov(infpercent ~ region)) # Analysis of variance (aov is a wrapper for lm)
            Df Sum Sq Mean Sq F value Pr(>F)
region       3  13.93   4.643   2.674 0.0519 .
Residuals   92 159.76   1.737
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
4 observations deleted due to missingness
>
> # Correlation matrix of quantitative variables
> cor(cbind(infpercent,census,nbeds,nurses,lngstay,age,xratio,culratio))
           infpercent      census       nbeds nurses   lngstay          age      xratio
infpercent          1          NA          NA     NA        NA           NA          NA
census             NA  1.00000000  0.98195092     NA 0.4723121 -0.03164390  0.07997783
nbeds              NA  0.98195092  1.00000000     NA 0.3994166 -0.05066108  0.05150114
nurses             NA          NA          NA      1        NA           NA          NA
lngstay            NA  0.47231213  0.39941663     NA 1.0000000  0.19789926  0.38600221
age                NA -0.03164390 -0.05066108     NA 0.1978993  1.00000000 -0.02332373
xratio             NA  0.07997783  0.05150114     NA 0.3860022 -0.02332373  1.00000000
culratio           NA  0.13833513  0.13622372     NA 0.3366701 -0.21189100  0.43978831
            culratio
infpercent         NA
census      0.1383351
nbeds       0.1362237
nurses             NA
lngstay     0.3366701
age        -0.2118910
xratio      0.4397883
culratio    1.0000000
```

```
> cor(cbind(infpercent,census,nbeds,nurses,lngstay,age,xratio,culratio),
use="complete.obs") # Casewise deletion
           infpercent     census      nbeds     nurses   lngstay         age
infpercent 1.00000000 0.39430770 0.35607667 0.42388414 0.5459418  0.03870252
census     0.39430770 1.00000000 0.98259503 0.89934502 0.5097167  0.02877024
nbeds      0.35607667 0.98259503 1.00000000 0.91605300 0.4373542  0.02456440
nurses     0.42388414 0.89934502 0.91605300 1.00000000 0.3787858  0.04033774
lngstay    0.54594178 0.50971675 0.43735421 0.37878576 1.0000000  0.21320119
age        0.03870252 0.02877024 0.02456440 0.04033774 0.2132012  1.00000000
xratio     0.49835433 0.09770767 0.07482915 0.13679080 0.4096145 -0.02461079
culratio   0.60357307 0.16321050 0.12992382 0.22312344 0.3577641 -0.15471397
               xratio   culratio
infpercent  0.49835433  0.6035731
census      0.09770767  0.1632105
nbeds       0.07482915  0.1299238
nurses      0.13679080  0.2231234
lngstay     0.40961455  0.3577641
age        -0.02461079 -0.1547140
xratio      1.00000000  0.5308683
culratio    0.53086826  1.0000000
> cor(cbind(infpercent,census,nbeds,nurses,lngstay,age,xratio,culratio),
use="pairwise.complete.obs") # Pairwise deletion
             infpercent      census       nbeds     nurses   lngstay           age
infpercent 1.0000000000  0.39311005  0.35675311  0.4238841 0.5349122  0.0006074981
census     0.3931100532  1.00000000  0.98195092  0.9085032 0.4723121 -0.0316438998
nbeds      0.3567531115  0.98195092  1.00000000  0.9242085 0.3994166 -0.0506610759
nurses     0.4238841391  0.90850320  0.92420849  1.0000000 0.3481316 -0.0110350991
lngstay    0.5349121983  0.47231213  0.39941663  0.3481316 1.0000000  0.1978992637
age        0.0006074981 -0.03164390 -0.05066108 -0.0110351 0.1978993  1.0000000000
xratio     0.4800060880  0.07997783  0.05150114  0.1166570 0.3860022 -0.0233237345
culratio   0.5921018396  0.13833513  0.13622372  0.2248672 0.3366701 -0.2118910025
               xratio   culratio
infpercent  0.48000609  0.5921018
census      0.07997783  0.1383351
nbeds       0.05150114  0.1362237
nurses      0.11665695  0.2248672
lngstay     0.38600221  0.3366701
age        -0.02332373 -0.2118910
xratio      1.00000000  0.4397883
culratio    0.43978831  1.0000000

> # Simple regression (One explanatory variable)
> nursemodel = lm(infpercent ~ nurses)
> summary(nursemodel)

Call:
lm(formula = infpercent ~ nurses)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3688 -0.8577  0.0475  0.7512  3.8610

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.607787   0.205423  17.563  < 2e-16 ***
nurses      0.004358   0.000976   4.465  2.3e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.23 on 91 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared:  0.1797,  Adjusted R-squared:  0.1707
F-statistic: 19.93 on 1 and 91 DF,  p-value: 2.296e-05
```

```
>
> censusmodel = lm(infpercent ~ census) # Census is number of patients
> summary(censusmodel)

Call:
lm(formula = infpercent ~ census)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5915 -0.8677  0.1160  0.7673  3.7205

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.6746564  0.2000750  18.366  < 2e-16 ***
census      0.0035206  0.0008494   4.145 7.43e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.25 on 94 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.1545,  Adjusted R-squared:  0.1455
F-statistic: 17.18 on 1 and 94 DF,  p-value: 7.434e-05

>
> censusnurses = lm(infpercent ~ census+nurses)
> summary(censusnurses)

Call:
lm(formula = infpercent ~ census + nurses)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3755 -0.8623  0.0726  0.7351  3.8478

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.5989981  0.2083398  17.275   <2e-16 ***
census      0.0006101  0.0019445   0.314    0.754
nurses      0.0037246  0.0022434   1.660    0.100
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.236 on 90 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared:  0.1806,  Adjusted R-squared:  0.1624
F-statistic: 9.916 on 2 and 90 DF,  p-value: 0.0001282

> sizemodel = lm(infpercent ~ census+nbeds+nurses)
> summary(sizemodel)

Call:
lm(formula = infpercent ~ census + nbeds + nurses)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4019 -0.8081 -0.0517  0.6187  3.9318

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.756578   0.209193  17.958  < 2e-16 ***
census       0.011593   0.004419   2.623  0.01025 *
nbeds       -0.010834   0.003947  -2.745  0.00732 **
nurses       0.006308   0.002362   2.671  0.00900 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.193 on 89 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared:  0.2445,  Adjusted R-squared:  0.2191
F-statistic: 9.603 on 3 and 89 DF,  p-value: 1.475e-05

>
> # All the quantitative variables except xratio and culratio
> summary(lm(infpercent ~ census+nbeds+nurses+lngstay+age))

Call:
lm(formula = infpercent ~ census + nbeds + nurses + lngstay +
    age)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3605 -0.7870 -0.0225  0.6774  2.7496

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.195413   1.396121   1.573  0.11946
census       0.001191   0.004575   0.260  0.79519
nbeds       -0.004562   0.003814  -1.196  0.23490
nurses       0.007060   0.002144   3.293  0.00144 **
lngstay      0.335167   0.071793   4.669 1.09e-05 ***
age         -0.026265   0.026270  -1.000  0.32018
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.079 on 87 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared:  0.396,   Adjusted R-squared:  0.3613
F-statistic: 11.41 on 5 and 87 DF,  p-value: 1.761e-08

>
> # This model has all the quantitative variables
> quantmodel = lm(infpercent ~ census+nbeds+nurses+lngstay+age+xratio+culratio)
> summary(quantmodel) # Time to think seriously about correlation and causation.

Call:
lm(formula = infpercent ~ census + nbeds + nurses + lngstay +
    age + xratio + culratio)

Residuals:
     Min       1Q   Median       3Q      Max
-2.11042 -0.69315  0.03675  0.50943  2.15902

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.147865   1.291290   0.115 0.909104
census       0.001966   0.004036   0.487 0.627422
nbeds       -0.002445   0.003377  -0.724 0.471005
nurses       0.003417   0.001997   1.711 0.090658 .
lngstay      0.164179   0.071900   2.283 0.024900 *
age          0.010646   0.024030   0.443 0.658871
xratio       0.010846   0.006222   1.743 0.084918 .
culratio     0.053322   0.013474   3.957 0.000157 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9443 on 85 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared:  0.548,   Adjusted R-squared:  0.5108
F-statistic: 14.72 on 7 and 85 DF,  p-value: 2.012e-12
```

```
> # Test size (census, nbeds and nurses simultaneously)
> # The restricted model rest1 has all the quantitative variables except size.
> rest1 = lm(infpercent ~ lngstay+age+xratio+culratio)
> anova(rest1,quantmodel)
Error in anova.lmlist(object, ...) :
  models were not all fitted to the same size of dataset

> # Excellent! R's anova function saves you from yourself.
> # F = \frac{(SSR_F-SSR_R)/r}{MSE_F} = \frac{(SSE_R-SSE_F)/r}{MSE_F}
```

$$F = \frac{(SSR_F - SSR_R)/r}{MSE_F} = \frac{(SSE_R - SSE_F)/r}{MSE_F}$$

```
>
> # lm omits cases with any NAs. Problems arise when the restricted model has
> # fewer explanatory variables, and there are missing values for those variables.
> # In that case, the restricted model has a larger sample size.
> # Need to omit cases with any NA on any variable in the full model (just nurses in
this case).
>
> # The following would also omit cases there mdschl is missing, which is not what you
want.
> # quant2 = lm(infpercent ~ census+nbeds+nurses+lngstay+age+xratio+culratio, data =
na.omit(senic))
>
> # Make a data frame with just the variables in the full model.
> senic2 = senic[,3:10] # All rows, columns 3-10
> quantmodel2 = lm(infpercent ~ census+nbeds+nurses+lngstay+age+xratio+culratio, data
= na.omit(senic2))
> rest2 = lm(infpercent ~ lngstay+age+xratio+culratio, data = na.omit(senic2))
> anova(rest2,quantmodel2)
Analysis of Variance Table

Model 1: infpercent ~ lngstay + age + xratio + culratio
Model 2: infpercent ~ census + nbeds + nurses + lngstay + age + xratio +
    culratio
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     88 83.200
2     85 75.799  3    7.4003 2.7662 0.04676 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> # General linear test is equivalent; only the full model is used.
> source("http://www.utstat.toronto.edu/~brunner/Rfunctions/ftest.txt")
> # ftest = function(model,L,h=0)
> # General linear test of H0: L beta = h
> # Full model; quantmodel = lm(infpercent ~
census+nbeds+nurses+lngstay+age+xratio+culratio)
> LL = rbind(c(0,1,0,0,0,0,0,0),
+            c(0,0,1,0,0,0,0,0),
+            c(0,0,0,1,0,0,0,0)   )
> ftest(quantmodel,LL)
         F         df1         df2     p-value
 2.76619925  3.00000000 85.00000000  0.04676028
```

```
> # Categorical variables: mdschl and region
>
> # Medical school affiliation
> # Earlier, we used an independent t-test
> t.test(infpercent ~ mdschl, var.equal = T)

      Two Sample t-test

data:  infpercent by mdschl
t = -2.542, df = 91, p-value = 0.01271
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.7215138 -0.2112167
sample estimates:
 mean in group No mean in group Yes
         4.205063          5.171429
>
> 5.171429 - 4.205063  # Difference between means
[1] 0.966366

> summary(lm(infpercent ~ mdschl))

Call:
lm(formula = infpercent ~ mdschl)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9051 -0.8051  0.1949  0.7949  3.5949

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.2051     0.1475  28.509   <2e-16 ***
mdschlYes     0.9664     0.3802   2.542   0.0127 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.311 on 91 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared:  0.0663,  Adjusted R-squared:  0.05604
F-statistic: 6.462 on 1 and 91 DF,  p-value: 0.01271

> is.factor(mdschl)
[1] TRUE
> # mdschl is a "factor" (unordered categorical variable). One of its attributes
> # is a dummy variable coding scheme.
> contrasts(mdschl)
    Yes
No    0
Yes   1
```

```
>
> # We can set up our own dummy variables. It's easy to do wrong when
> # there are missing values.
> n = length(infpercent)
> mschool = numeric(n) # Vector of length n, all zeros
> mschool[mdschl=='Yes'] = 1; mschool[is.na(mdschl)] = NA
> # Never believe you did it right. Check it.
> table(mdschl,mschool,useNA='always') # Include missing values in the table.
      mschool
mdschl  0  1 <NA>
  No   80  0    0
  Yes   0 16    0
  <NA>  0  0    4
> summary(lm(infpercent ~ mschool))

Call:
lm(formula = infpercent ~ mschool)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9051 -0.8051  0.1949  0.7949  3.5949

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.2051     0.1475  28.509   <2e-16 ***
mschool       0.9664     0.3802   2.542   0.0127 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.311 on 91 degrees of freedom
  (7 observations deleted due to missingness)
Multiple R-squared:  0.0663,  Adjusted R-squared:  0.05604
F-statistic: 6.462 on 1 and 91 DF,  p-value: 0.01271

> # Region of the U.S.
> summary(lm(infpercent ~ region))

Call:
lm(formula = infpercent ~ region)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0593 -0.9200  0.0537  0.8883  3.4407

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.35926    0.25361  17.189   <2e-16 ***
regionNortheast  0.47407    0.36969   1.282    0.203
regionSouth     -0.53926    0.34957  -1.543    0.126
regionWest       0.02741    0.42437   0.065    0.949
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.318 on 92 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.08019, Adjusted R-squared:  0.0502
F-statistic: 2.674 on 3 and 92 DF,  p-value: 0.0519

> contrasts(region) # Note alphabetical order
             Northeast South West
NorthCentral         0     0    0
Northeast            1     0    0
South                0     1    0
West                 0     0    1
```

```
>
> # Make our own dummy variables
> # Make all 4 dummy variables. Just use 3 if there is an intercept.
> # Region has no missing values, but assume it might.
> nc=ne=s=w = numeric(n) # All zeros
> nc[region=='NorthCentral'] = 1; nc[is.na(region)] = NA
> ne[region=='Northeast'] = 1;    ne[is.na(region)] = NA
> s[region=='South'] = 1;          s[is.na(region)] = NA
> w[region=='West'] = 1;           w[is.na(region)] = NA
>
> # Always check dummy variables
> table(nc,region,useNA='always')
      region
nc     NorthCentral Northeast South West <NA>
  0               0        24    30   15    0
  1              31         0     0    0    0
  <NA>            0         0     0    0    0
> table(ne,region,useNA='always')
> table(s,region,useNA='always')
> table(w,region,useNA='always')
>
> justregion = lm(infpercent ~ nc+ne+w) # South is the reference category
> summary(justregion)

Call:
lm(formula = infpercent ~ nc + ne + w)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0593 -0.9200  0.0537  0.8883  3.4407

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.8200     0.2406  15.877  < 2e-16 ***
nc            0.5393     0.3496   1.543  0.12636
ne            1.0133     0.3609   2.808  0.00609 **
w             0.5667     0.4167   1.360  0.17721
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.318 on 92 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.08019, Adjusted R-squared:  0.0502
F-statistic: 2.674 on 3 and 92 DF,  p-value: 0.0519

> tabular(region ~ infpercent*(Mean+SD+N))

             infpercent
 region      Mean       SD    N
 NorthCentral 4.359      1.442 27
 Northeast    4.833      1.336 24
 South        3.820      1.354 30
 West         4.387      0.907 15
>
```

```
> # Or, let R do the work
> contrasts(region) = contr.treatment(4,base=3) # Third is reference category
> contrasts(region)
              1 2 4
NorthCentral 1 0 0
Northeast    0 1 0
South        0 0 0
West         0 0 1
> summary(lm(infpercent ~ region))

Call:
lm(formula = infpercent ~ region)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0593 -0.9200  0.0537  0.8883  3.4407

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.8200     0.2406  15.877  < 2e-16 ***
region1       0.5393     0.3496   1.543  0.12636
region2       1.0133     0.3609   2.808  0.00609 **
region4       0.5667     0.4167   1.360  0.17721
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.318 on 92 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.08019, Adjusted R-squared:  0.0502
F-statistic: 2.674 on 3 and 92 DF,  p-value: 0.0519

>
> # Nicer labels
> ContrastMatrixSouth = contr.treatment(4,base=3)
> colnames(ContrastMatrixSouth) = c("NC","NE","W")
> contrasts(region) = ContrastMatrixSouth
> contrasts(region)
              NC NE W
NorthCentral  1  0 0
Northeast     0  1 0
South         0  0 0
West          0  0 1
> summary(lm(infpercent ~ region))

Call:
lm(formula = infpercent ~ region)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0593 -0.9200  0.0537  0.8883  3.4407

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.8200     0.2406  15.877  < 2e-16 ***
regionNC      0.5393     0.3496   1.543  0.12636
regionNE      1.0133     0.3609   2.808  0.00609 **
regionW       0.5667     0.4167   1.360  0.17721
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.318 on 92 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.08019, Adjusted R-squared:  0.0502
F-statistic: 2.674 on 3 and 92 DF,  p-value: 0.0519
```

```
> # No intercept
> noint1 = lm(infpercent ~ 0 + nc+ne+s+w); summary(noint1)

Call:
lm(formula = infpercent ~ 0 + nc + ne + s + w)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0593 -0.9200  0.0537  0.8883  3.4407

Coefficients:
   Estimate Std. Error t value Pr(>|t|)
nc   4.3593     0.2536   17.19   <2e-16 ***
ne   4.8333     0.2690   17.97   <2e-16 ***
s    3.8200     0.2406   15.88   <2e-16 ***
w    4.3867     0.3403   12.89   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.318 on 92 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.9185,  Adjusted R-squared:  0.9149
F-statistic: 259.2 on 4 and 92 DF,  p-value: < 2.2e-16

>
> # Test equality of 4 means
> L2 = rbind( c(1,-1, 0, 0),
+             c(0, 1,-1, 0),
+             c(0, 0, 1,-1))
> ftest(noint1,L2) # Compare F = 2.674

        F       df1       df2   p-value
 2.673584  3.000000 92.000000  0.051899
>
> # Test South versus Northeast
> L3 = rbind( c(0, 1,-1, 0))
> ftest(noint1,L3) # Compare F = t^2 = 2.808^2 = 7.88
           F          df1          df2      p-value
 7.884110437  1.000000000 92.000000000  0.006089212
>
> # What will lm do with a factor and no intercept?
> summary(lm(infpercent ~ 0 + region))

Call:
lm(formula = infpercent ~ 0 + region)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0593 -0.9200  0.0537  0.8883  3.4407

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
regionNorthCentral   4.3593     0.2536   17.19   <2e-16 ***
regionNortheast      4.8333     0.2690   17.97   <2e-16 ***
regionSouth          3.8200     0.2406   15.88   <2e-16 ***
regionWest           4.3867     0.3403   12.89   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.318 on 92 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.9185,  Adjusted R-squared:  0.9149
F-statistic: 259.2 on 4 and 92 DF,  p-value: < 2.2e-16
```

```
>
> # Now a full model including the categorical variables.
> # R^2 from quantmodel was 0.548
> # Could do it this way:
> # lm(infpercent ~ census+nbeds+nurses+lngstay+age+xratio+culratio+mdschl+region)
> # But how about
> fullmod = update(quantmodel, . ~ . + mdschl + region)
> summary(fullmod)

Call:
lm(formula = infpercent ~ census + nbeds + nurses + lngstay +
    age + xratio + culratio + mdschl + region)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8306 -0.5273  0.0364  0.4689  1.9483

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.119954   1.311818  -0.091  0.92738
census       0.005745   0.004054   1.417  0.16045
nbeds       -0.005241   0.003352  -1.564  0.12197
nurses       0.004009   0.001990   2.015  0.04738 *
lngstay      0.233959   0.074699   3.132  0.00244 **
age          0.002413   0.023960   0.101  0.92003
xratio       0.008030   0.006116   1.313  0.19303
culratio     0.058690   0.013354   4.395 3.45e-05 ***
mdschlYes   -0.612755   0.364909  -1.679  0.09712 .
regionNC     0.273660   0.266697   1.026  0.30801
regionNE    -0.236151   0.306243  -0.771  0.44296
regionW      0.912365   0.317348   2.875  0.00521 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9027 on 78 degrees of freedom
  (10 observations deleted due to missingness)
Multiple R-squared:  0.6062,  Adjusted R-squared:  0.5507
F-statistic: 10.92 on 11 and 78 DF,  p-value: 6.957e-12

>
>
> # Predict infection percent for a hospital in the South with no medical school
affiliation, 200 patients, 250 beds and 180 nurses, where the mean length of stay is
12 days, mean patient age is 51 years, xratio rate is 70% and the culturing ratio is
20%.
>
> fullmod$coefficients # beta-hat values
 (Intercept)       census        nbeds       nurses      lngstay          age
-0.119954218  0.005744689 -0.005241114  0.004008682  0.233958891  0.002413362
      xratio     culratio    mdschlYes     regionNC     regionNE      regionW
 0.008030024  0.058689788 -0.612754961  0.273660475 -0.236151328  0.912364558
>
> x = c(1,200,250,180,12,51,70,20,0, 0,0,0)
> sum(x*fullmod$coefficients)
[1] 5.106754
>
> # Use the predict function -- takes a data frame as input
> newhosp = data.frame(region='South', mdschl='No', census=200, nbeds=250, nurses=180,
lngstay=12,  age=51, xratio=70, culratio=20)
> predict(fullmod,newdata=newhosp)
       1
5.106754
```

```
>
> # Want to test region, size variables controlling for other variables.
> # The following is easier than setting up L matrices for general linear test
> fullmodel = lm(infpercent ~
census+nbeds+nurses+lngstay+age+xratio+culratio+mdschl+region,
+             data = na.omit(senic))

> restregion = lm(infpercent ~ census+nbeds+nurses+lngstay+age+xratio+culratio+mdschl,
+             data = na.omit(senic)) # Restricted model without region

> anova(restregion,fullmodel)
Analysis of Variance Table

Model 1: infpercent ~ census + nbeds + nurses + lngstay + age + xratio +
    culratio + mdschl
Model 2: infpercent ~ census + nbeds + nurses + lngstay + age + xratio +
    culratio + mdschl + region
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     81 73.533
2     78 63.566  3    9.9674 4.0769 0.0096 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

>
> restsize = lm(infpercent ~ lngstay+age+xratio+culratio+mdschl+region,
+             data = na.omit(senic))

> anova(restsize,fullmodel)
Analysis of Variance Table

Model 1: infpercent ~ lngstay + age + xratio + culratio + mdschl + region
Model 2: infpercent ~ census + nbeds + nurses + lngstay + age + xratio +
    culratio + mdschl + region
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     81 72.332
2     78 63.566  3    8.7663 3.5856 0.01741 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

```
> # Reporting sample mean infection rates by region is a bad idea. Compare:
> tabular(region ~ infpercent*(Mean+SD+N))

              infpercent
 region       Mean       SD    N
 NorthCentral 4.359      1.442 27
 Northeast    4.833      1.336 24
 South        3.820      1.354 30
 West         4.387      0.907 15
> summary(fullmodel)

Call:
lm(formula = infpercent ~ census + nbeds + nurses + lngstay +
    age + xratio + culratio + mdschl + region, data = na.omit(senic))

Residuals:
    Min      1Q  Median      3Q     Max
-1.8306 -0.5273  0.0364  0.4689  1.9483

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      0.153706   1.308717   0.117  0.90681
census           0.005745   0.004054   1.417  0.16045
nbeds           -0.005241   0.003352  -1.564  0.12197
nurses           0.004009   0.001990   2.015  0.04738 *
lngstay          0.233959   0.074699   3.132  0.00244 **
age              0.002413   0.023960   0.101  0.92003
xratio           0.008030   0.006116   1.313  0.19303
culratio         0.058690   0.013354   4.395 3.45e-05 ***
mdschlYes       -0.612755   0.364909  -1.679  0.09712 .
regionNortheast -0.509812   0.282081  -1.807  0.07457 .
regionSouth     -0.273660   0.266697  -1.026  0.30801
regionWest       0.638704   0.313793   2.035  0.04520 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9027 on 78 degrees of freedom
Multiple R-squared:  0.6062,  Adjusted R-squared:  0.5507
F-statistic: 10.92 on 11 and 78 DF,  p-value: 6.957e-12

> # Controlling for other variables, it looks like Northeast has a lower rate than
West.

> # Report y-hat values with other x variables held to their sample means. Call these
"corrected" means or something. Copying and pasting all the values would be a drag, so
...
```

```
> # Report y-hat values with other x variables held to their sample means. Call these
"corrected" means or something. Copying and pasting all the values would be a drag, so
...
>
> NorthCentral = data.frame(region='NorthCentral', mdschl='No', census=mean(census),
nbeds=mean(nbeds), nurses=mean(nurses, na.rm=T), lngstay=mean(lngstay),
age=mean(age), xratio=mean(xratio), culratio=mean(culratio))
>
> Northeast =  data.frame(region='Northeast', mdschl='No', census=mean(census),
nbeds=mean(nbeds), nurses=mean(nurses, na.rm=T), lngstay=mean(lngstay),
age=mean(age), xratio=mean(xratio), culratio=mean(culratio))
>
> South =  data.frame(region='South', mdschl='No', census=mean(census),
nbeds=mean(nbeds), nurses=mean(nurses, na.rm=T), lngstay=mean(lngstay),
age=mean(age), xratio=mean(xratio), culratio=mean(culratio))
>
> West =  data.frame(region='West', mdschl='No', census=mean(census),
nbeds=mean(nbeds), nurses=mean(nurses, na.rm=T), lngstay=mean(lngstay),
age=mean(age), xratio=mean(xratio), culratio=mean(culratio))
>
> average = rbind(NorthCentral, Northeast, South, West); average

        region mdschl census  nbeds   nurses lngstay    age xratio culratio
1 NorthCentral     No 192.41 252.88 175.3505  9.6921 53.207  81.62    15.63
2    Northeast     No 192.41 252.88 175.3505  9.6921 53.207  81.62    15.63
3        South     No 192.41 252.88 175.3505  9.6921 53.207  81.62    15.63
4         West     No 192.41 252.88 175.3505  9.6921 53.207  81.62    15.63

>
> predict(fullmodel,average)
       1        2        3        4
4.605286 4.095474 4.331626 5.243990

>
> # West - North central should equal 0.638704, the regression coefficient for the
West dummy variable.
> 5.243990 - 4.605286
[1] 0.638704
```

```
> # As one last peek at missing values, Just do the regression for hospitals without
medical school affiliation, which are in the majority. The issue is that medical
school affiliation is missing sometimes.
>
> table(mdschl,useNA='always') # Want 80 hospitals
mdschl
  No  Yes <NA>
  80   16    4
>
> # A little experiment first
>
> id = 1:8
> x = c(1,2,2,1,2,2,NA,NA)
> M = cbind(id,x); M
     id  x
[1,]  1  1
[2,]  2  2
[3,]  3  2
[4,]  4  1
[5,]  5  2
[6,]  6  2
[7,]  7 NA
[8,]  8 NA
>
> x[x==2]
[1]  2  2  2  2 NA NA
> id[x==2]
[1]  2  3  5  6 NA NA
> two = subset(id,x==2); two
[1] 2 3 5 6
> M[two,]
     id x
[1,]  2 2
[2,]  3 2
[3,]  5 2
[4,]  6 2
>
>
> # Now select the hospitals
> id = 1:n # n = 100 hospitals
> noschool =  subset(id,mdschl=='No'); length(noschool)
[1] 80
> senicnomdschl = senic[noschool,]; dim(senicnomdschl)
[1] 80 10
> head(senicnomdschl)
     region mdschl census nbeds nurses lngstay  age xratio culratio infpercent
1 Northeast     No    237   298    115   12.01 52.8   96.9     10.8        4.8
3 Northeast     No    127   165    158    9.36 54.1   90.6     18.3        4.8
5      West     No     51    76     79    6.70 48.6   80.8     13.0        4.5
6     South     No     59    95     56    8.93 56.0   72.5      6.2        2.0
7     South     No    468   600    497    9.84 62.2   82.3     12.0        4.8
8     South     No    349   477    188    7.91 52.8   79.5     11.9        2.9
>
> summary(lm(infpercent ~ census+nbeds+nurses+lngstay+age+xratio+culratio+region,
+             data = senicnomdschl))
```