Lab4_Regression_1_a24kimwu

2025-09-28

Load dataset

```
# First, check if the package is already installed
if (!("mlbench" %in% rownames(installed.packages()))){
# If not, install
install.packages("mlbench")
}
# And load
library(mlbench)
```

```
library(Metrics)
```

```
data(BostonHousing2)
x <- BostonHousing2
x <- x[,-5] # Remove old target variable. Our target is cmedv.
```

```
?BostonHousing
```

```
## starting httpd help server ... done
```

```
head(x)
```

```
##         town tract      lon     lat cmedv    crim zn indus chas   nox    rm
## 1     Nahant  2011 -70.9550 42.2550  24.0 0.00632 18  2.31    0 0.538 6.575
## 2 Swampscott  2021 -70.9500 42.2875  21.6 0.02731  0  7.07    0 0.469 6.421
## 3 Swampscott  2022 -70.9360 42.2830  34.7 0.02729  0  7.07    0 0.469 7.185
## 4 Marblehead  2031 -70.9280 42.2930  33.4 0.03237  0  2.18    0 0.458 6.998
## 5 Marblehead  2032 -70.9220 42.2980  36.2 0.06905  0  2.18    0 0.458 7.147
## 6 Marblehead  2033 -70.9165 42.3040  28.7 0.02985  0  2.18    0 0.458 6.430
##    age    dis rad tax ptratio      b lstat
## 1 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 58.7 6.0622   3 222    18.7 394.12  5.21
```

Introduction on regression models with R

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

```
class(x[,1])
```

```
## [1] "factor"
```

```
lapply(x, class)
```

```
## $town
## [1] "factor"
##
## $tract
## [1] "integer"
##
## $lon
## [1] "numeric"
##
## $lat
## [1] "numeric"
##
## $cmedv
## [1] "numeric"
##
## $crim
## [1] "numeric"
##
## $zn
## [1] "numeric"
##
## $indus
## [1] "numeric"
##
## $chas
## [1] "factor"
##
## $nox
## [1] "numeric"
##
## $rm
## [1] "numeric"
##
## $age
## [1] "numeric"
##
## $dis
## [1] "numeric"
##
## $rad
## [1] "integer"
##
## $tax
## [1] "integer"
##
## $ptratio
## [1] "numeric"
##
## $b
## [1] "numeric"
##
## $lstat
```

```
## [1] "numeric"
```

```
idxNum <- unlist(lapply(x,class)) == "numeric"
idxNum
```

```
##    town   tract     lon     lat   cmedv    crim      zn   indus    chas     nox
##   FALSE   FALSE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE   FALSE    TRUE
##      rm     age     dis     rad     tax ptratio       b   lstat
##    TRUE    TRUE    TRUE   FALSE   FALSE    TRUE    TRUE    TRUE
```

Data exploration

```
cor(x[,idxNum])
```

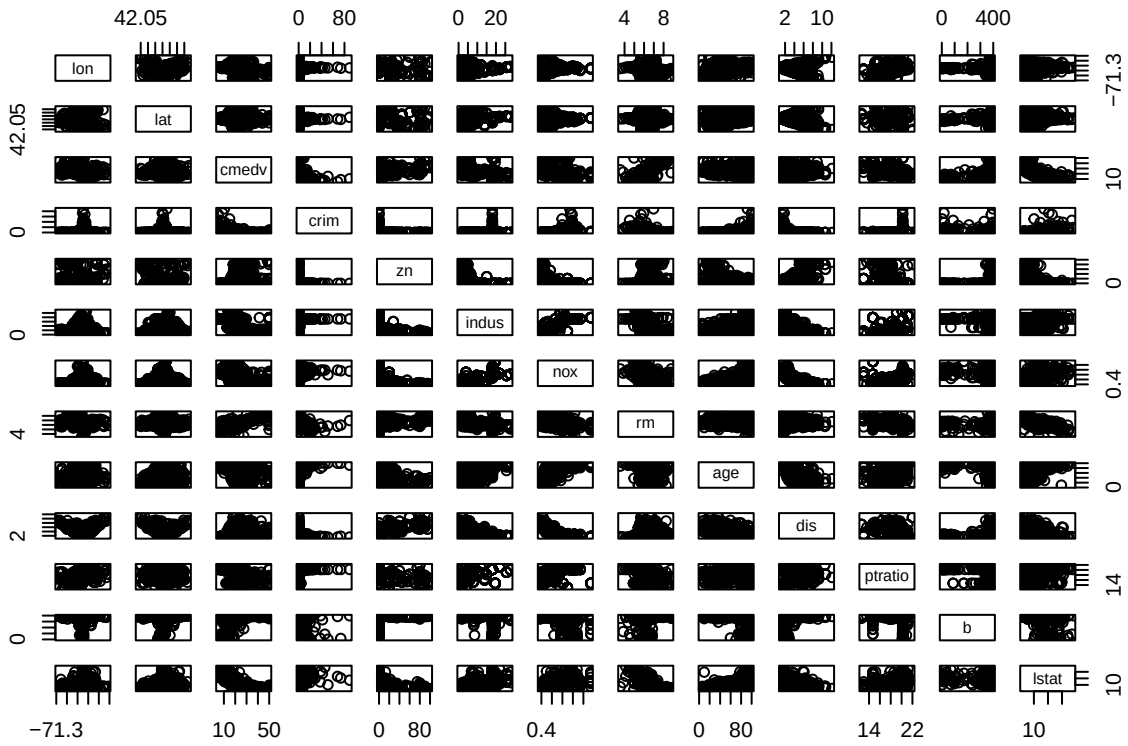
```
##                lon          lat        cmedv        crim         zn
## lon      1.00000000  0.143053589 -0.322946685  0.06510061 -0.2180811
## lat      0.14305359  1.000000000  0.006825792 -0.08429296 -0.1296674
## cmedv   -0.32294669  0.006825792  1.000000000 -0.38958244  0.3603862
## crim     0.06510061 -0.084292955 -0.389582441  1.00000000 -0.2004692
## zn      -0.21808107 -0.129667394  0.360386177 -0.20046922  1.0000000
## indus    0.06270245 -0.041093480 -0.484754379  0.40658341 -0.5338282
## nox      0.16087125 -0.068600401 -0.429300219  0.42097171 -0.5166037
## rm      -0.25711003 -0.069316987  0.696303794 -0.21924670  0.3119906
## age      0.20473895  0.079035217 -0.377998896  0.35273425 -0.5695373
## dis     -0.01124313 -0.082980855  0.249314834 -0.37967009  0.6644082
## ptratio  0.31260219 -0.004527081 -0.505654619  0.28994558 -0.3916785
## b       -0.01829986  0.105253702  0.334860832 -0.38506394  0.1755203
## lstat    0.19562959  0.045659550 -0.740835993  0.45562148 -0.4129946
##               indus        nox          rm         age         dis      ptratio
## lon      0.06270245  0.1608713 -0.25711003  0.20473895 -0.01124313  0.312602186
## lat     -0.04109348 -0.0686004 -0.06931699  0.07903522 -0.08298085 -0.004527081
## cmedv   -0.48475438 -0.4293002  0.69630379 -0.37799890  0.24931483 -0.505654619
## crim     0.40658341  0.4209717 -0.21924670  0.35273425 -0.37967009  0.289945579
## zn      -0.53382819 -0.5166037  0.31199059 -0.56953734  0.66440822 -0.391678548
## indus    1.00000000  0.7636514 -0.39167585  0.64477851 -0.70802699  0.383247556
## nox      0.76365145  1.0000000 -0.30218819  0.73147010 -0.76923011  0.188932677
## rm      -0.39167585 -0.3021882  1.00000000 -0.24026493  0.20524621 -0.355501495
## age      0.64477851  0.7314701 -0.24026493  1.00000000 -0.74788054  0.261515012
## dis     -0.70802699 -0.7692301  0.20524621 -0.74788054  1.00000000 -0.232470542
## ptratio  0.38324756  0.1889327 -0.35550149  0.26151501 -0.23247054  1.000000000
## b       -0.35697654 -0.3800506  0.12806864 -0.27353398  0.29151167 -0.177383302
## lstat    0.60379972  0.5908789 -0.61380827  0.60233853 -0.49699583  0.374044317
##                  b       lstat
## lon      -0.01829986  0.19562959
## lat       0.10525370  0.04565955
## cmedv     0.33486083 -0.74083599
## crim     -0.38506394  0.45562148
## zn        0.17552032 -0.41299457
## indus    -0.35697654  0.60379972
## nox      -0.38005064  0.59087892
## rm        0.12806864 -0.61380827
## age      -0.27353398  0.60233853
## dis       0.29151167 -0.49699583
## ptratio -0.17738330  0.37404432
```

```
## b         1.00000000 -0.36608690
## lstat    -0.36608690  1.00000000
```

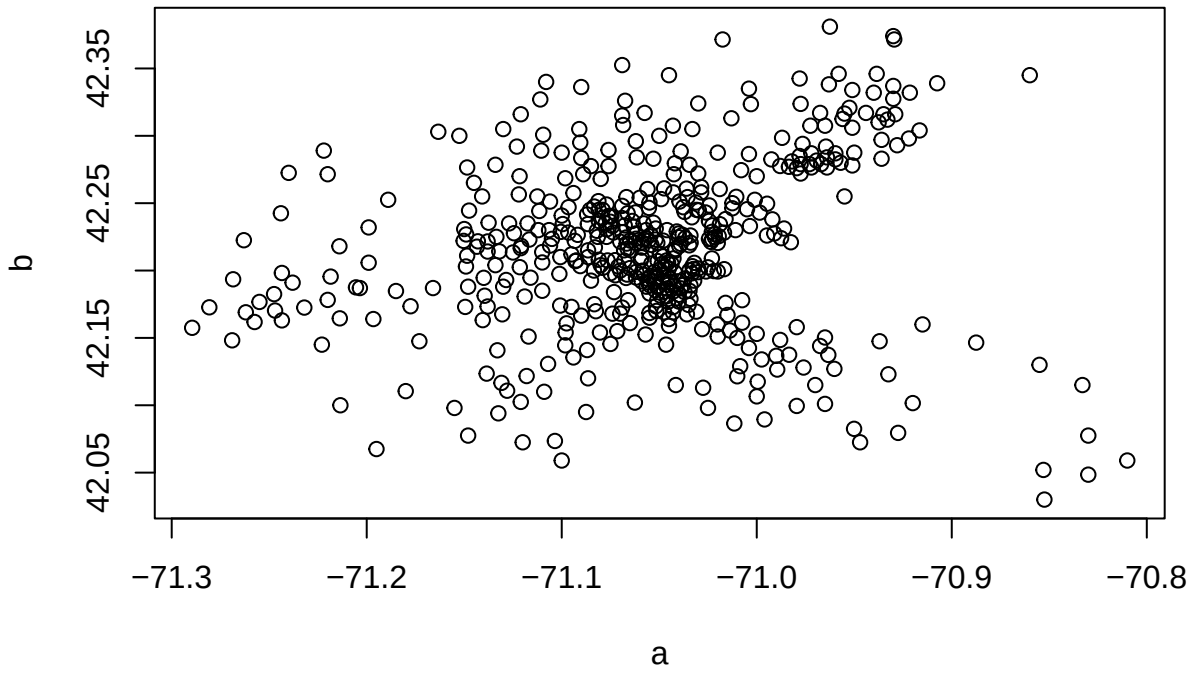
```
a <- x[,3] # Store third column of x as variable a
b <- x[,4] # Store fourth column of x as variable b
cor(a,b) # correlation between the 3rd and 4th column
```

```
## [1] 0.1430536
```

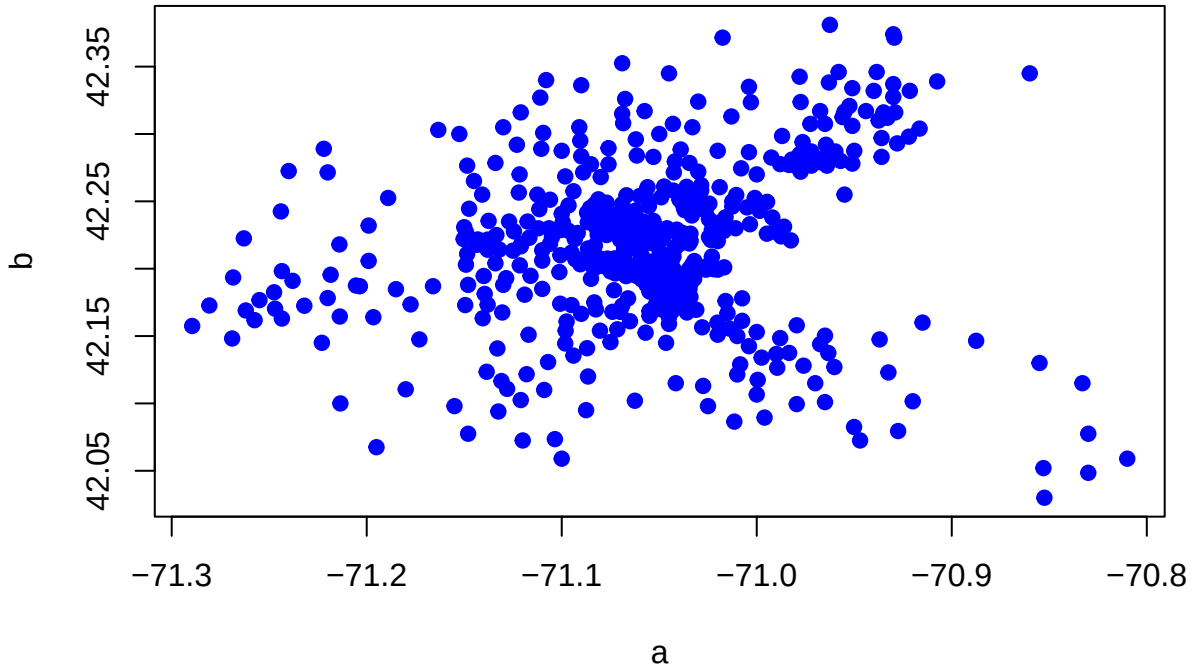
```
plot(x[,idxNum])
```



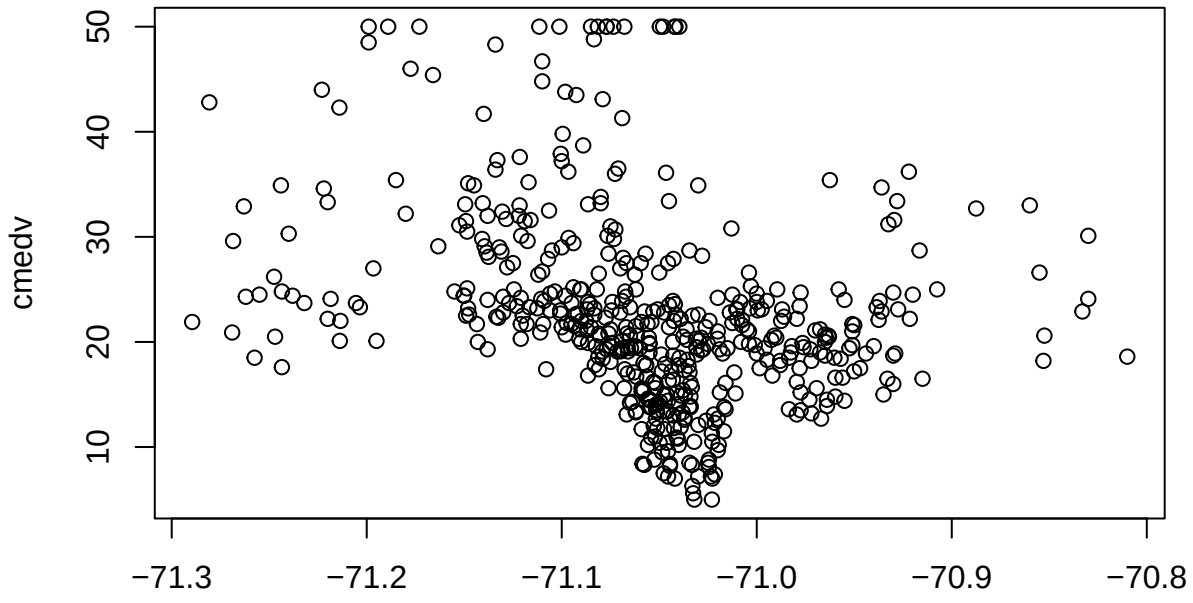

```
plot(a,b)
```

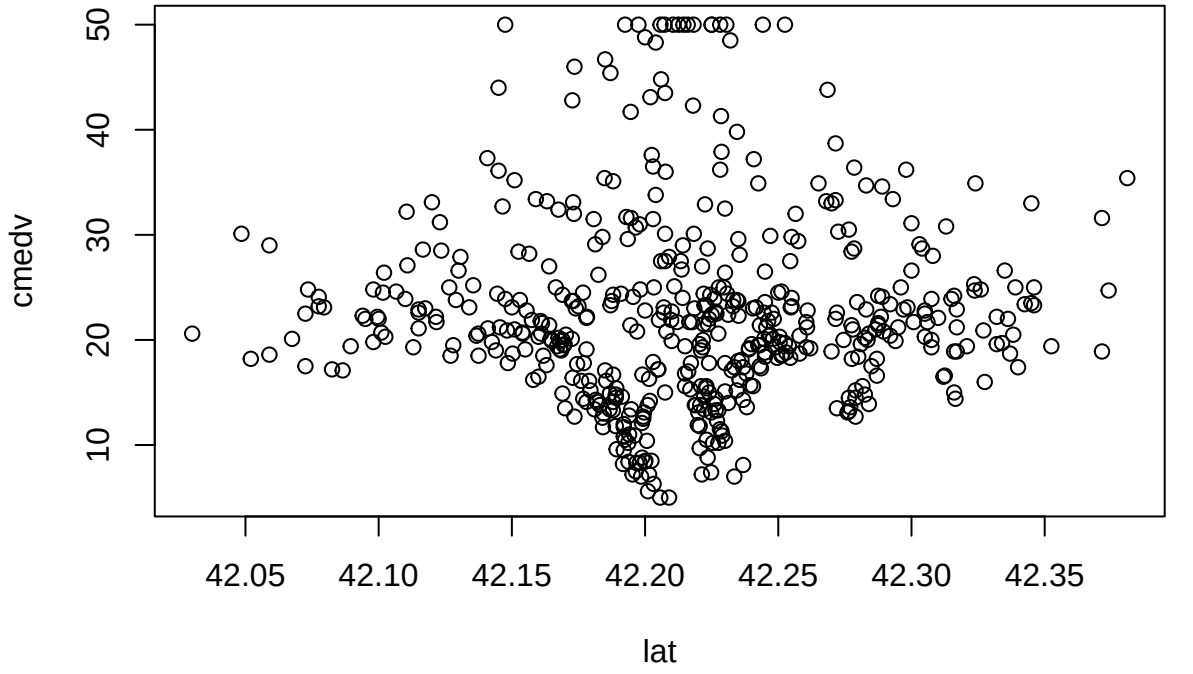


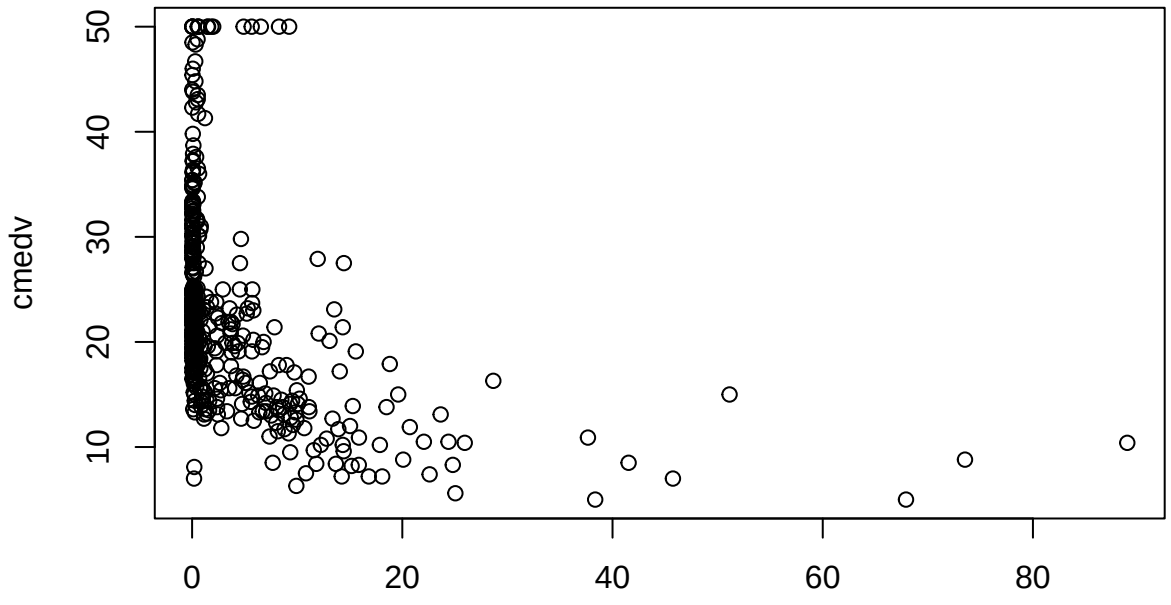

```
plot(a,b,pch=19,col="blue")
```

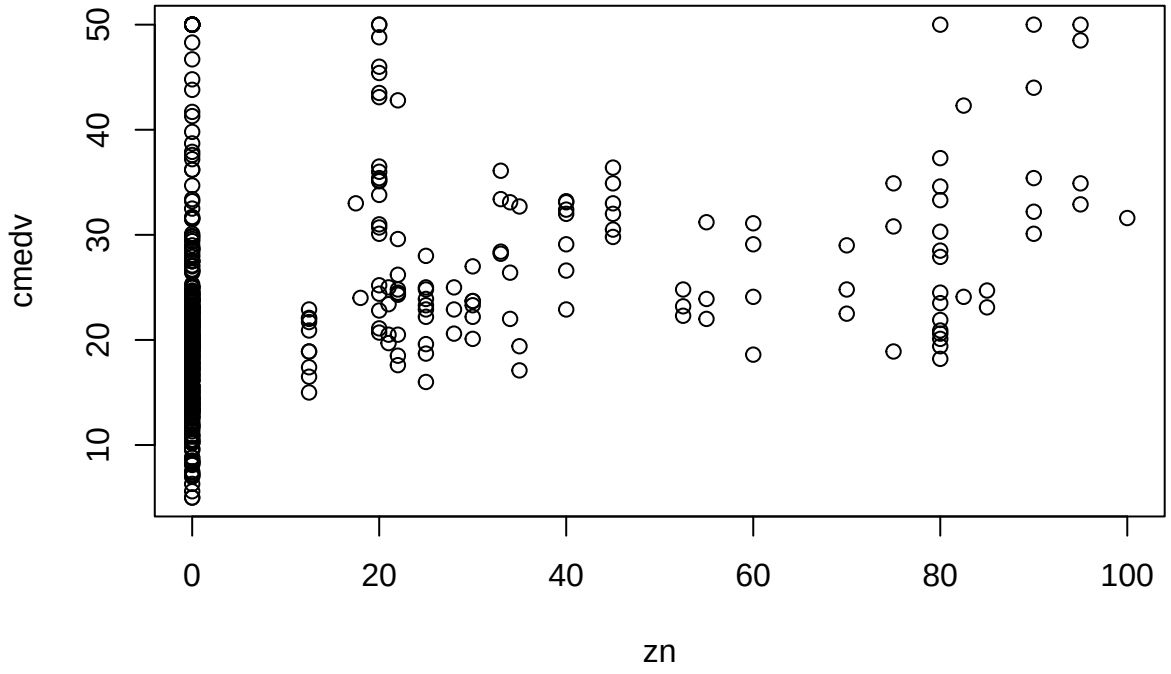


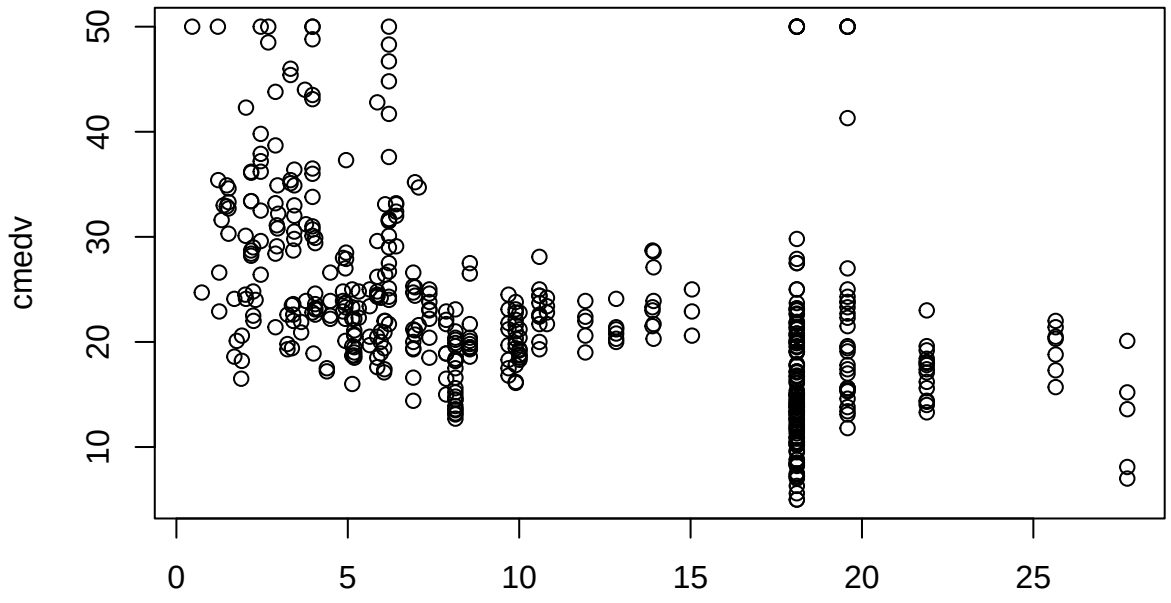

```
tempY <- x[,5] # Store only target
tempX <- (x[,idxNum])[,-3] # Store only numerical inputs
# the [,-3] removes the third column, that is cmedv
# Now we produce all the plots
for (i in 1:ncol(tempX)){ # Iterate for each column in tempX
 plot(tempX[,i],tempY,xlab=colnames(tempX)[i],ylab="cmedv")
}
```

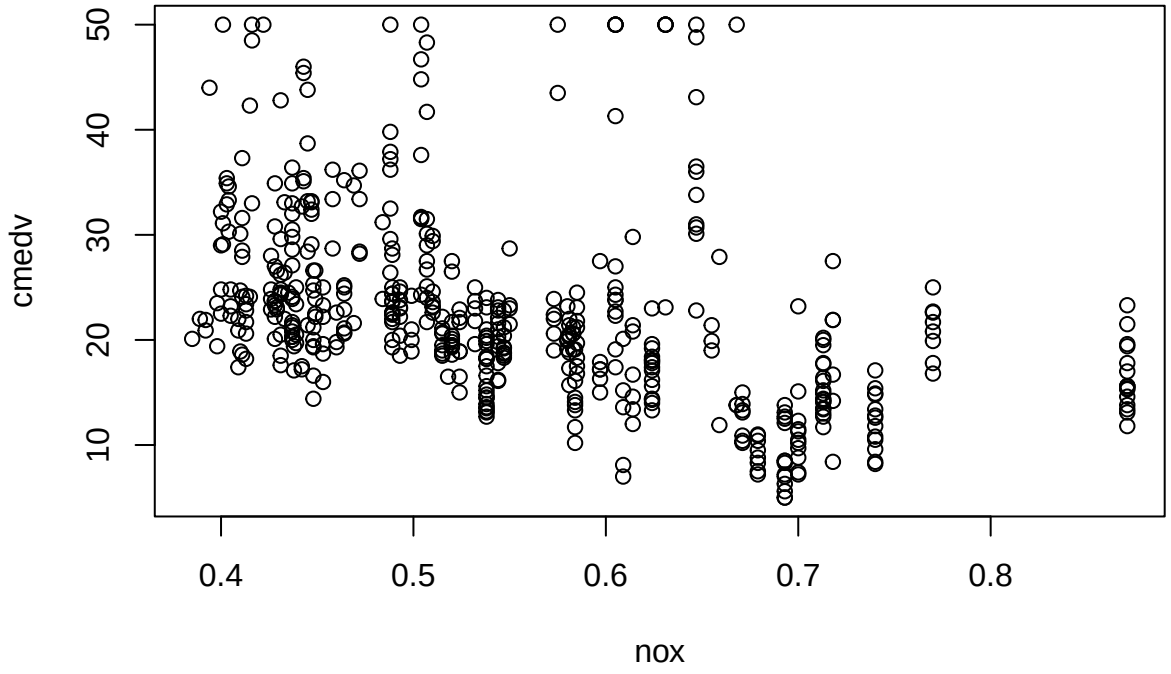

cmedv
10
20
30
40
50
−71.3
−71.2
−71.1
−71.0
−70.9
−70.8
lon

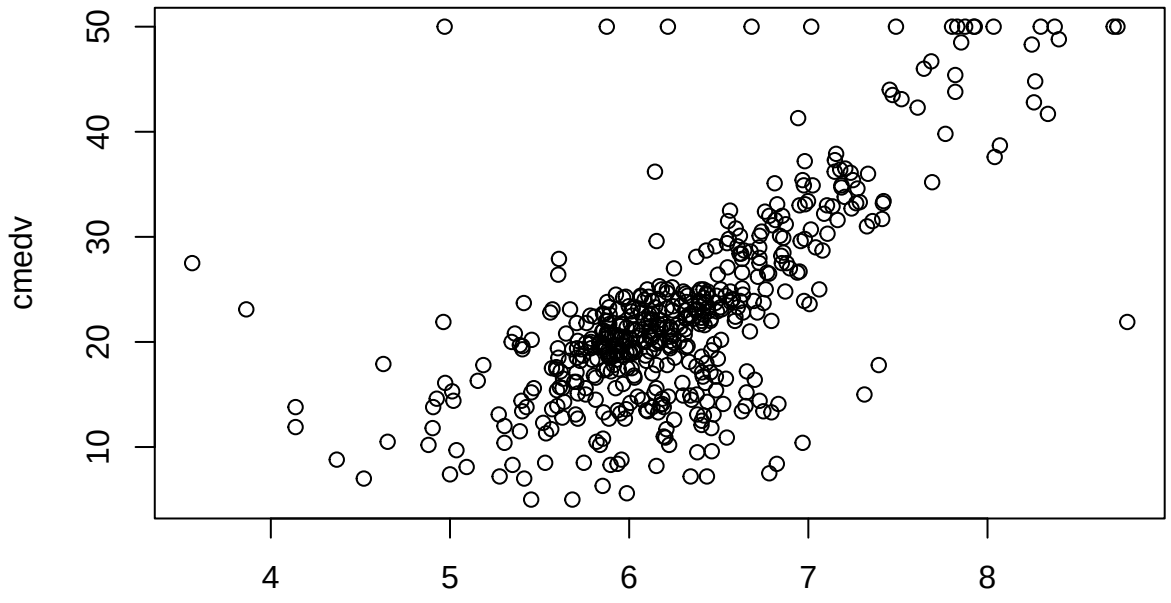

cmedv
10
20
30
40
50
42.05
42.10
42.15
42.20
42.25
42.30
42.35
lat

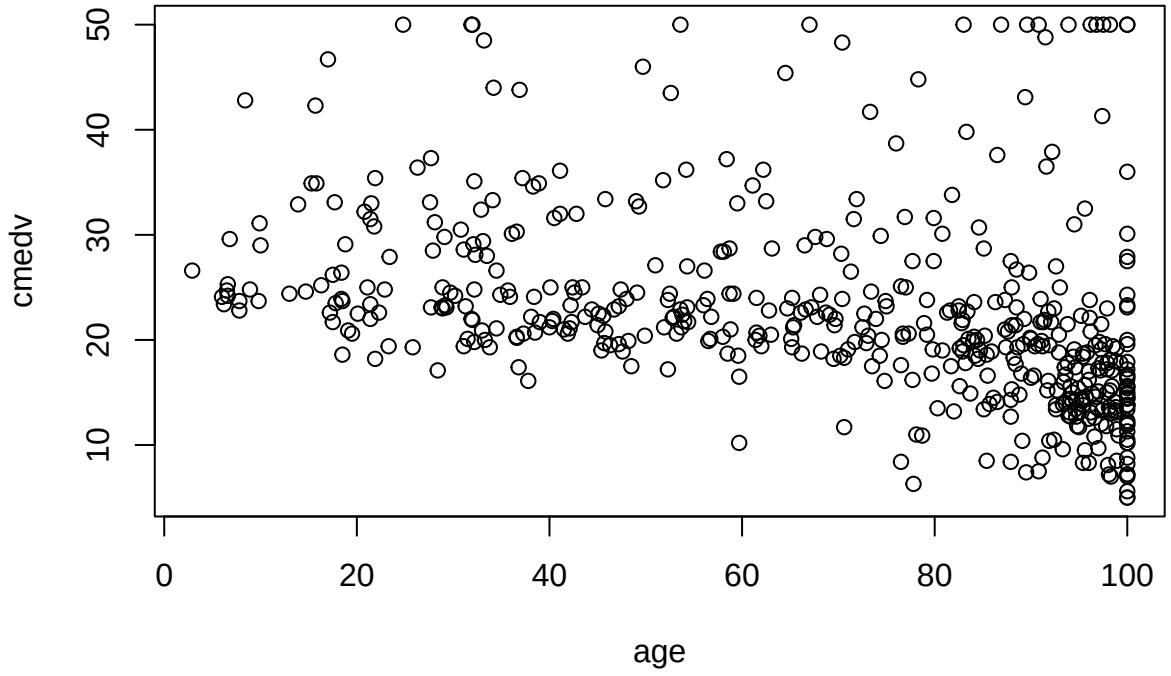

cmedv
10
20
30
40
50
0
20
40
60
80
crim

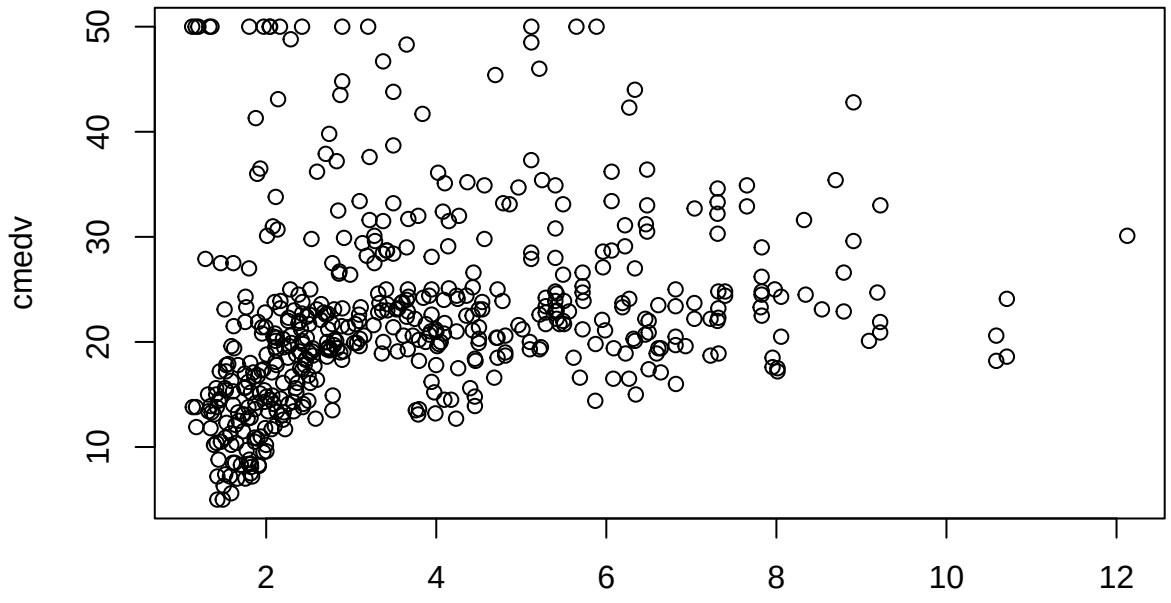

cmedv
10
20
30
40
50
0
20
40
60
80
100
zn

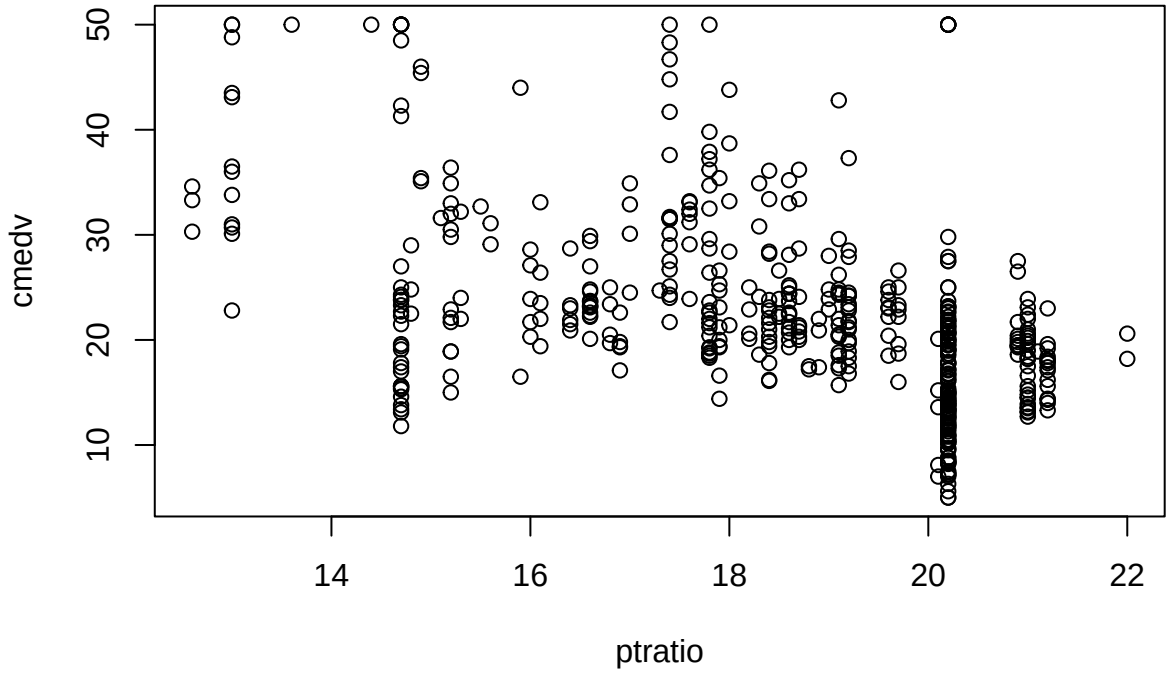

cmedv
10
20
30
40
50
0
5
10
15
20
25
indus

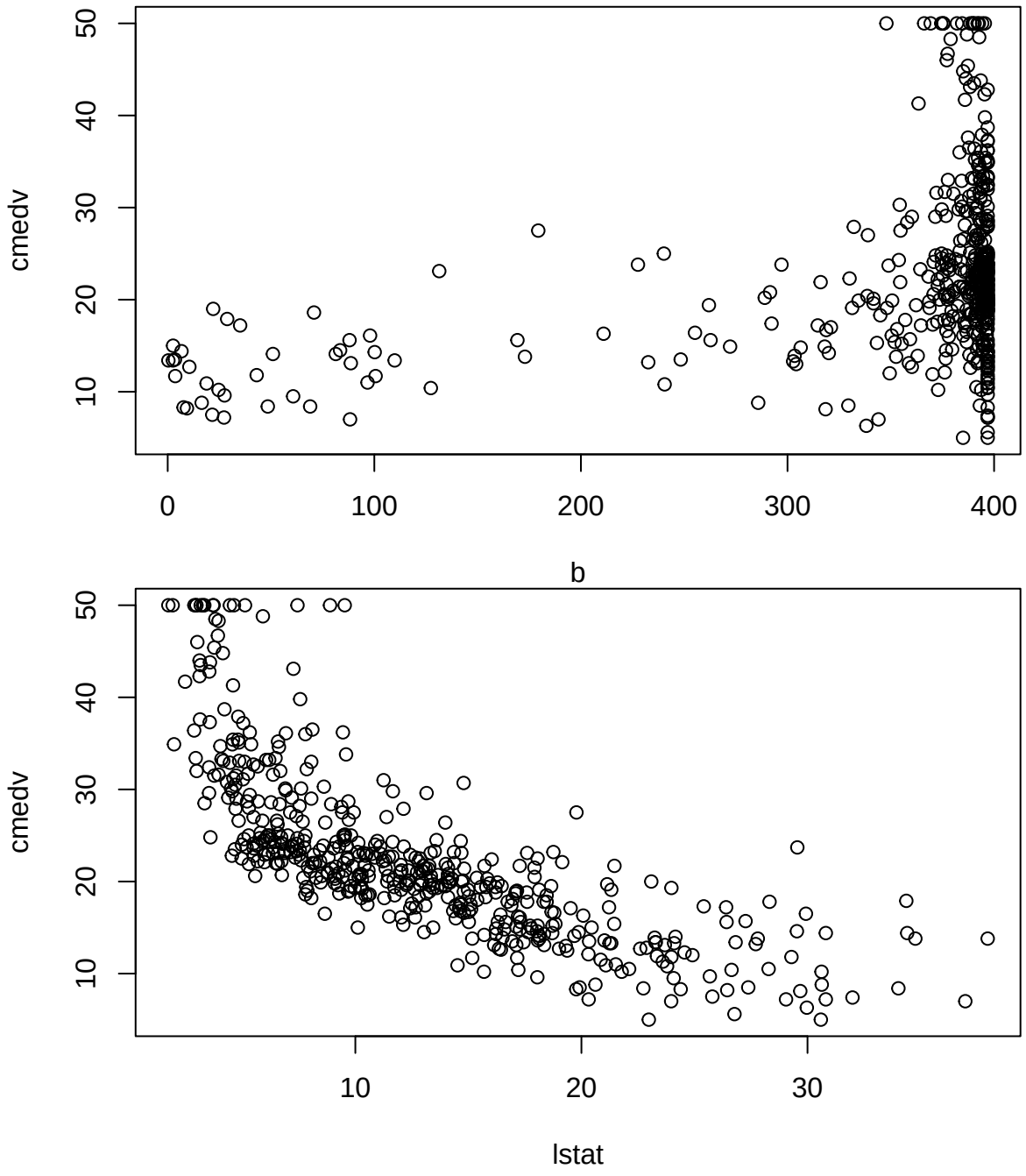

cmedv
10
20
30
40
50
0.4
0.5
0.6
0.7
0.8
nox


cmedv
10
20
30
40
50
4
5
6
7
8
rm


cmedv
10
20
30
40
50
0
20
40
60
80
100
age


cmedv
10
20
30
40
50
2
4
6
8
10
12
dis


cmedv
10
20
30
40
50
14
16
18
20
22
ptratio



Building a regression model

```
# Regression sales on price
fit1 <- lm(cmedv ~ lstat, data=x)
fit1
```

```
##
## Call:
## lm(formula = cmedv ~ lstat, data = x)
##
## Coefficients:
```

```
## (Intercept)        lstat
##     34.5820      -0.9526
```

```
summary(fit1)
```

```
##
## Call:
## lm(formula = cmedv ~ lstat, data = x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.951 -3.997 -1.325  2.086 24.496
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.58200    0.55882   61.88   <2e-16 ***
## lstat       -0.95259    0.03847  -24.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.174 on 504 degrees of freedom
## Multiple R-squared:  0.5488, Adjusted R-squared:  0.5479
## F-statistic: 613.1 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
sum1 <- summary(fit1)
```

```
names(sum1)
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
# Get p-values and evaluate with a = 0.05
sum1$coefficients
```

```
##               Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 34.5820044 0.55881606  61.88441 1.264146e-237
## lstat       -0.9525876 0.03847103 -24.76116  3.731519e-89
```

```
pval <- sum1$coefficients[,4]
pval <= 0.05
```

```
## (Intercept)       lstat
##        TRUE        TRUE
```

```
fit1r2 <- sum1$r.squared
fit1r2
```

```
## [1] 0.548838
```

```
fit1aic <- AIC(fit1)
fit1aic
```

```
## [1] 3282.096
```

```
# Regress sales on revenue
fit2 <- lm(cmedv~lat,data=x) # Fit the regression model
sum2 <- summary(fit2) # Get summary statistics
sum2
```

```
##
## Call:
## lm(formula = cmedv ~ lat, data = x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.521  -5.508  -1.355   2.478  27.541
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -20.302    279.495  -0.073    0.942
## lat            1.015      6.621   0.153    0.878
##
## Residual standard error: 9.191 on 504 degrees of freedom
## Multiple R-squared:  4.659e-05,  Adjusted R-squared:  -0.001937
## F-statistic: 0.02348 on 1 and 504 DF,  p-value: 0.8783
```

```
# Get R^2 and AIC
fit2r2 <- sum2$r.squared
fit2aic <- AIC(fit2)
```

```
# Test coefficients
sum2$coefficients[,4] <= 0.05
```

```
## (Intercept)         lat
##       FALSE       FALSE
```

```
sum2$coefficients
```

```
##               Estimate Std. Error     t value  Pr(>|t|)
## (Intercept) -20.301550  279.49509 -0.07263652 0.9421242
## lat           1.014543    6.62052  0.15324222 0.8782686
```

```
 r2 <- c(fit1r2,fit2r2)
 r2
```

```
## [1] 5.488380e-01 4.659144e-05
```

```
round(r2,2)
```

```
## [1] 0.55 0.00
```

```
 names(r2) <- c("lstat","lat")
 round(r2,3)
```

```
## lstat   lat
## 0.549 0.000
```

```
aic <- c(fit1aic,fit2aic)
names(aic) <- names(r2)
round(aic,3)
```

```
##    lstat      lat
## 3282.096 3684.813
```

```
 # Regress sales on revenue
 fit3 <- lm(cmedv~town,data=x) # Fit the regression model
 sum3 <- summary(fit3) # Get summary statistics
 sum3
```

```
##
## Call:
## lm(formula = cmedv ~ town, data = x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2250  -2.2458  -0.3083   1.7409  26.3500
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   25.2000     2.0376  12.367  < 2e-16 ***
## townAshland                   -3.8000     4.3225  -0.879 0.379840
## townBedford                    4.9000     4.3225   1.134 0.257612
## townBelmont                   11.0000     2.7901   3.942 9.47e-05 ***
## townBeverly                   -4.4000     2.9993  -1.467 0.143133
## townBoston Allston-Brighton   -4.2875     2.7901  -1.537 0.125139
## townBoston Back Bay            7.2000     2.9993   2.401 0.016810 *
## townBoston Beacon Hill        24.8000     3.7202   6.666 8.40e-11 ***
## townBoston Charlestown       -12.5500     2.9993  -4.184 3.49e-05 ***
## townBoston Dorchester         -7.4545     2.6065  -2.860 0.004452 **
## townBoston Downtown           -6.3375     2.7901  -2.271 0.023635 *
## townBoston East Boston       -13.6917     2.5640  -5.340 1.54e-07 ***
## townBoston Forest Hills       -7.8286     2.8816  -2.717 0.006870 **
## townBoston Hyde Park          -4.8250     3.3790  -1.428 0.154066
## townBoston Mattapan           -5.0333     2.9993  -1.678 0.094069 .
## townBoston North End         -11.4000     4.3225  -2.637 0.008669 **
## townBoston Roxbury           -13.5842     2.3836  -5.699 2.29e-08 ***
## townBoston Savin Hill        -11.9652     2.3271  -5.142 4.21e-07 ***
## townBoston South Boston      -16.0769     2.5274  -6.361 5.31e-10 ***
## townBoston West Roxbury       -1.8250     3.3790  -0.540 0.589420
## townBraintree                 -2.6250     2.7901  -0.941 0.347348
## townBrookline                 12.8250     2.5640   5.002 8.40e-07 ***
## townBurlington                -2.1750     3.3790  -0.644 0.520140
## townCambridge                 -1.5500     2.2629  -0.685 0.493750
## townCanton                     1.9667     3.7202   0.529 0.597332
## townChelsea                  -12.4000     3.1567  -3.928 0.000100 ***
## townCohasset                   7.5000     5.7633   1.301 0.193864
## townConcord                    7.5333     3.7202   2.025 0.043509 *
## townDanvers                   -3.0500     3.3790  -0.903 0.367247
## townDedham                     0.1200     3.1567   0.038 0.969694
## townDover                     24.8000     5.7633   4.303 2.10e-05 ***
## townDuxbury                    4.9000     5.7633   0.850 0.395699
## townEverett                   -5.7714     2.8816  -2.003 0.045847 *
## townFramingham                 0.1200     2.6567   0.045 0.963995
## townHamilton                  -0.5000     5.7633  -0.087 0.930907
## townHanover                   -2.1000     5.7633  -0.364 0.715763
## townHingham                    2.3500     4.3225   0.544 0.586959
## townHolbrook                  -6.9500     4.3225  -1.608 0.108623
## townHull                      -8.7000     5.7633  -1.510 0.131918
## townLexington                  7.5667     2.9993   2.523 0.012016 *
## townLincoln                   24.8000     5.7633   4.303 2.10e-05 ***
## townLynn                      -8.4864     2.3394  -3.628 0.000322 ***
## townLynnfield                  7.6500     4.3225   1.770 0.077492 .
## townMalden                    -5.5111     2.7168  -2.029 0.043148 *
```

```
## townManchester                7.8000     5.7633   1.353 0.176667
## townMarblehead                7.5667     3.7202   2.034 0.042593 *
## townMarshfield               -3.8500     4.3225  -0.891 0.373608
## townMedfield                  7.0000     5.7633   1.215 0.225215
## townMedford                  -4.0818     2.6065  -1.566 0.118114
## townMelrose                  -0.9750     3.3790  -0.289 0.773074
## townMiddleton                -6.3000     5.7633  -1.093 0.274972
## townMillis                   -3.2000     5.7633  -0.555 0.579030
## townMilton                    6.3250     3.3790   1.872 0.061933 .
## townNahant                   -1.2000     5.7633  -0.208 0.835163
## townNatick                   -1.8667     2.9993  -0.622 0.534043
## townNeedham                   6.7600     3.1567   2.141 0.032817 *
## townNewton                    8.3444     2.4014   3.475 0.000565 ***
## townNorfolk                  -5.1000     5.7633  -0.885 0.376716
## townNorth Reading            -3.7500     4.3225  -0.868 0.386136
## townNorwell                  -0.7000     5.7633  -0.121 0.903387
## townNorwood                  -0.8800     3.1567  -0.279 0.780557
## townPeabody                  -4.3667     2.7168  -1.607 0.108759
## townPembroke                 -5.8000     4.3225  -1.342 0.180386
## townQuincy                   -4.8667     2.5640  -1.898 0.058377 .
## townRandolph                 -4.6333     3.7202  -1.245 0.213666
## townReading                  -0.3000     3.3790  -0.089 0.929297
## townRevere                   -4.8375     2.7901  -1.734 0.083700 .
## townRockland                 -7.8500     4.3225  -1.816 0.070078 .
## townSalem                    -5.7714     2.8816  -2.003 0.045847 *
## townSargus                   -4.1750     3.3790  -1.236 0.217320
## townScituate                 -0.4500     4.3225  -0.104 0.917134
## townSharon                    0.2333     3.7202   0.063 0.950019
## townSherborn                 18.8000     5.7633   3.262 0.001198 **
## townSomerville               -8.1067     2.4677  -3.285 0.001106 **
## townStoneham                 -2.3667     3.7202  -0.636 0.525016
## townSudbury                   8.7000     4.3225   2.013 0.044788 *
## townSwampscott                2.9500     4.3225   0.682 0.495315
## townTopsfield                10.2000     5.7633   1.770 0.077492 .
## townWakefield                -1.4000     3.3790  -0.414 0.678853
## townWalpole                  -1.7667     3.7202  -0.475 0.635117
## townWaltham                  -2.1000     2.6065  -0.806 0.420897
## townWatertown                -1.0750     3.3790  -0.318 0.750539
## townWayland                   8.0000     4.3225   1.851 0.064910 .
## townWellesley                15.2750     3.3790   4.521 8.06e-06 ***
## townWenham                    6.4000     5.7633   1.110 0.267436
## townWeston                   24.0500     4.3225   5.564 4.74e-08 ***
## townWestwood                  6.0333     3.7202   1.622 0.105610
## townWeymouth                 -5.1625     2.7901  -1.850 0.064987 .
## townWilmington               -5.1000     3.7202  -1.371 0.171148
## townWinchester                7.9000     3.1567   2.503 0.012712 *
## townWinthrop                 -3.6200     3.1567  -1.147 0.252136
## townWoburn                   -3.9000     2.9993  -1.300 0.194220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.391 on 414 degrees of freedom
## Multiple R-squared:  0.7174, Adjusted R-squared:  0.6553
## F-statistic: 11.55 on 91 and 414 DF,  p-value: < 2.2e-16
```

```r
# Get R^2 and AIC
fit3r2 <- sum3$r.squared
fit3aic <- AIC(fit3)
```

```r
# Model with intercept
lm(cmedv ~ lstat, data=x)
```

```
##
## Call:
## lm(formula = cmedv ~ lstat, data = x)
##
## Coefficients:
## (Intercept)        lstat
##     34.5820      -0.9526
```

```r
# Model without intercept
lm(cmedv ~ 0 + lstat, data=x)
```

```
##
## Call:
## lm(formula = cmedv ~ 0 + lstat, data = x)
##
## Coefficients:
## lstat
## 1.121
```

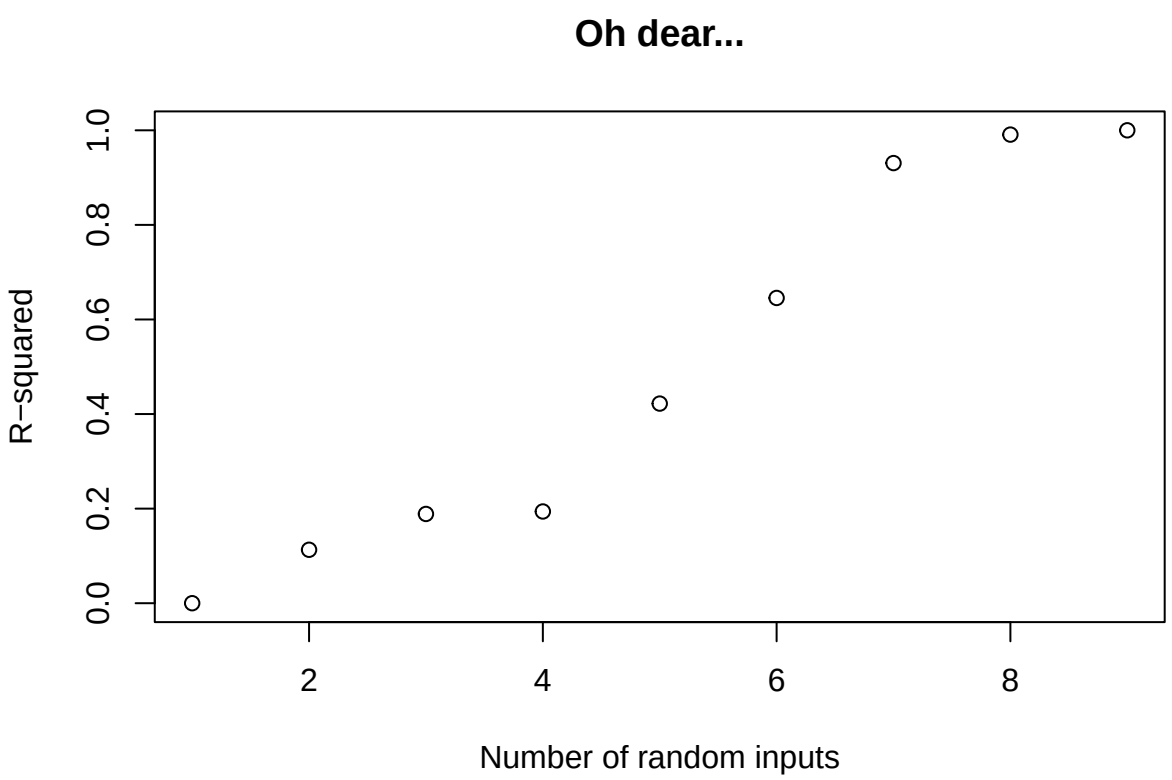
Multiple regression

```r
fit4 <- lm(cmedv~lstat+lat,data=x)
sum4 <- summary(fit4)
sum4
```

```
##
## Call:
## lm(formula = cmedv ~ lstat + lat, data = x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.903 -4.155 -1.386  2.062 24.491
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -221.00354  187.75285  -1.177    0.240
## lstat         -0.95498    0.03848 -24.818   <2e-16 ***
## lat            6.05489    4.44789   1.361    0.174
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.168 on 503 degrees of freedom
## Multiple R-squared:  0.5505, Adjusted R-squared:  0.5487
## F-statistic:   308 on 2 and 503 DF,  p-value: < 2.2e-16
```

```r
# Create a variable yy that includes the first 10 values of
# cmedv, our target variable
yy <- x[,colnames(x)=="cmedv"]
```

```
yy <- yy[1:10]
# Now create a matrix with 9 columns of random data
#Thefunctionrunif()createsrandom drawsfromauniform
#distirbution
xx <-matrix(runif(90),ncol=9)#Draw 100valuesandputthem
#ina matrixwith10columns
#Loopforallregressionsfrom1to9inputs
ftemp<-list() #Pre-allocatealist tosavetheresults
for(i in 1:9){
ftemp[[i]]<-lm(yy~xx[,1:i])
}
#GetR-squared fromallmodels
r2temp<-unlist(lapply(ftemp,function(x){summary(x)$r.squared}))
plot(1:9,r2temp,xlab="Number of random inputs",ylab="R-squared",main="Oh dear...")
```




```
sapply(ftemp,AIC)
```

```
## [1] 72.00283 72.80390 73.91138 75.84629 74.51927 71.63233 57.30468 38.94336
## [9]     -Inf
```

```
yy-ftemp[[9]]$fitted.values
```

```
##  1  2  3  4  5  6  7  8  9 10
##  0  0  0  0  0  0  0  0  0  0
```

```
ftemp[[10]] <- lm(yy~1) # This means just fit a constant
sapply(ftemp,AIC)
```

```
##  [1] 72.00283 72.80390 73.91138 75.84629 74.51927 71.63233 57.30468 38.94336
##  [9]     -Inf 70.00286
```

```r
unlist(lapply(ftemp,function(x){summary(x)$r.squared}))
```

```
##  [1] 2.410150e-06 1.129869e-01 1.887246e-01 1.939878e-01 4.221027e-01
##  [6] 6.455021e-01 9.307352e-01 9.909587e-01 1.000000e+00 0.000000e+00
```

Variable selection

```r
fit5 <- lm(cmedv~.,data=x)
summary(fit5)
```

```
##
## Call:
## lm(formula = cmedv ~ ., data = x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.7502  -1.3118   0.0000   0.9575  16.8532
##
## Coefficients: (5 not defined because of singularities)
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  9.096e+02  1.593e+03   0.571  0.56837
## townAshland                 -1.304e+01  9.063e+00  -1.439  0.15088
## townBedford                 -1.646e+00  3.402e+00  -0.484  0.62879
## townBelmont                  7.250e+00  1.773e+00   4.089 5.24e-05 ***
## townBeverly                  3.927e+01  3.919e+01   1.002  0.31693
## townBoston Allston-Brighton  1.103e+02  9.732e+01   1.134  0.25768
## townBoston Back Bay          1.182e+02  9.474e+01   1.248  0.21292
## townBoston Beacon Hill       1.269e+02  9.208e+01   1.378  0.16904
## townBoston Charlestown       8.492e+01  8.650e+01   0.982  0.32686
## townBoston Dorchester        7.554e+01  7.017e+01   1.077  0.28231
## townBoston Downtown          8.945e+01  7.834e+01   1.142  0.25421
## townBoston East Boston       9.016e+01  8.396e+01   1.074  0.28349
## townBoston Forest Hills      6.701e+01  6.476e+01   1.035  0.30143
## townBoston Hyde Park         6.074e+01  5.954e+01   1.020  0.30824
## townBoston Mattapan          7.080e+01  6.763e+01   1.047  0.29580
## townBoston North End         1.035e+02  8.939e+01   1.158  0.24740
## townBoston Roxbury           7.829e+01  7.542e+01   1.038  0.29988
## townBoston Savin Hill        7.599e+01  7.259e+01   1.047  0.29581
## townBoston South Boston      8.015e+01  8.099e+01   0.990  0.32292
## townBoston West Roxbury      6.296e+01  6.226e+01   1.011  0.31252
## townBraintree               -2.407e+01  1.713e+01  -1.405  0.16066
## townBrookline               -3.265e+00  1.250e+01  -0.261  0.79402
## townBurlington               1.728e+00  7.370e+00   0.234  0.81478
## townCambridge                9.608e+00  1.885e+00   5.096 5.35e-07 ***
## townCanton                  -2.311e+01  1.606e+01  -1.439  0.15100
## townChelsea                  5.448e+01  5.378e+01   1.013  0.31168
## townCohasset                -2.031e+01  1.835e+01  -1.107  0.26906
## townConcord                  2.694e-01  3.848e+00   0.070  0.94421
## townDanvers                  4.045e+01  4.069e+01   0.994  0.32085
## townDedham                  -1.725e+01  1.264e+01  -1.365  0.17292
## townDover                   -3.125e+00  1.376e+01  -0.227  0.82048
## townDuxbury                 -4.445e+01  4.055e+01  -1.096  0.27365
## townEverett                  4.832e+00  4.315e+00   1.120  0.26347
```

```
## townFramingham             -9.815e+00  8.187e+00  -1.199  0.23129
## townHamilton                4.136e+01  4.022e+01   1.028  0.30444
## townHanover                -5.090e+01  3.959e+01  -1.285  0.19939
## townHingham                -4.589e+01  3.923e+01  -1.170  0.24278
## townHolbrook               -2.833e+01  1.763e+01  -1.607  0.10880
## townHull                   -5.219e+01  3.905e+01  -1.337  0.18206
## townLexington               8.985e-01  2.279e+00   0.394  0.69360
## townLincoln                 1.185e+01  4.075e+00   2.907  0.00385 **
## townLynn                    4.185e+01  4.161e+01   1.006  0.31514
## townLynnfield               4.426e+01  4.096e+01   1.080  0.28060
## townMalden                  2.334e+00  4.543e+00   0.514  0.60760
## townManchester              4.648e+01  3.967e+01   1.172  0.24207
## townMarblehead              4.870e+01  4.292e+01   1.135  0.25717
## townMarshfield             -5.122e+01  4.026e+01  -1.272  0.20402
## townMedfield               -1.678e+01  1.404e+01  -1.195  0.23274
## townMedford                 2.829e+00  4.901e+00   0.577  0.56405
## townMelrose                 2.135e+00  6.150e+00   0.347  0.72870
## townMiddleton               4.041e+01  4.050e+01   0.998  0.31895
## townMillis                 -2.365e+01  1.443e+01  -1.639  0.10209
## townMilton                 -1.872e+01  1.638e+01  -1.143  0.25387
## townNahant                  4.289e+01  4.317e+01   0.994  0.32105
## townNatick                 -1.251e+01  7.631e+00  -1.640  0.10186
## townNeedham                -1.399e+01  1.293e+01  -1.082  0.27990
## townNewton                 -1.992e+00  5.145e+00  -0.387  0.69877
## townNorfolk                -2.254e+01  1.495e+01  -1.507  0.13258
## townNorth Reading           3.368e+00  8.704e+00   0.387  0.69903
## townNorwell                -4.996e+01  3.988e+01  -1.253  0.21101
## townNorwood                -2.113e+01  1.555e+01  -1.359  0.17496
## townPeabody                 4.029e+01  4.071e+01   0.990  0.32289
## townPembroke               -5.226e+01  4.079e+01  -1.281  0.20087
## townQuincy                 -2.165e+01  1.675e+01  -1.292  0.19695
## townRandolph               -2.719e+01  1.737e+01  -1.565  0.11827
## townReading                 3.885e+00  7.053e+00   0.551  0.58203
## townRevere                  5.456e+01  5.114e+01   1.067  0.28658
## townRockland               -5.075e+01  3.931e+01  -1.291  0.19739
## townSalem                   4.626e+01  4.242e+01   1.090  0.27615
## townSargus                  4.071e+01  4.097e+01   0.994  0.32098
## townScituate               -4.961e+01  4.008e+01  -1.238  0.21652
## townSharon                 -2.480e+01  1.584e+01  -1.566  0.11814
## townSherborn               -1.335e+00  9.203e+00  -0.145  0.88476
## townSomerville              5.759e-01  2.155e+00   0.267  0.78945
## townStoneham                1.668e+00  5.999e+00   0.278  0.78104
## townSudbury                -1.156e+00  4.702e+00  -0.246  0.80595
## townSwampscott              4.543e+01  4.302e+01   1.056  0.29166
## townTopsfield               4.651e+01  4.030e+01   1.154  0.24911
## townWakefield               3.702e+00  6.658e+00   0.556  0.57853
## townWalpole                -2.277e+01  1.507e+01  -1.511  0.13148
## townWaltham                -2.132e+00  3.807e+00  -0.560  0.57573
## townWatertown              -1.524e+00  4.423e+00  -0.345  0.73060
## townWayland                -1.804e+00  4.417e+00  -0.408  0.68323
## townWellesley              -7.629e+00  1.326e+01  -0.575  0.56528
## townWenham                  4.495e+01  3.982e+01   1.129  0.25958
## townWeston                  7.693e+00  4.371e+00   1.760  0.07915 .
## townWestwood               -2.035e+01  1.529e+01  -1.331  0.18398
```

```
## townWeymouth                -2.547e+01  1.794e+01  -1.420  0.15633
## townWilmington               4.068e+00  8.072e+00   0.504  0.61458
## townWinchester               6.173e+00  5.472e+00   1.128  0.25995
## townWinthrop                 4.775e+01  4.849e+01   0.985  0.32533
## townWoburn                   1.807e+00  6.879e+00   0.263  0.79297
## tract                        2.934e-02  2.744e-02   1.069  0.28562
## lon                          3.643e+00  1.865e+01   0.195  0.84526
## lat                         -1.761e+01  2.228e+01  -0.790  0.42972
## crim                        -8.857e-03  2.612e-02  -0.339  0.73476
## zn                                 NA         NA      NA       NA
## indus                              NA         NA      NA       NA
## chas1                       -1.179e+00  7.527e-01  -1.567  0.11800
## nox                         -2.918e+01  4.909e+00  -5.943 6.07e-09 ***
## rm                           5.144e+00  3.624e-01  14.194  < 2e-16 ***
## age                         -5.959e-02  1.196e-02  -4.980 9.45e-07 ***
## dis                         -8.236e-01  5.396e-01  -1.526  0.12773
## rad                                NA         NA      NA       NA
## tax                                NA         NA      NA       NA
## ptratio                            NA         NA      NA       NA
## b                            1.382e-02  2.613e-03   5.287 2.04e-07 ***
## lstat                       -2.194e-01  4.723e-02  -4.646 4.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.265 on 403 degrees of freedom
## Multiple R-squared:  0.8991, Adjusted R-squared:  0.8735
## F-statistic:  35.2 on 102 and 403 DF,  p-value: < 2.2e-16
```

```
fitmin <- lm(cmedv~1,data=x) # This means use only an intercept.
```

```
fit6 <- step(fitmin,direction="both",scope=formula(fit5)) # fit5 is the complete model
```

```
## Start:  AIC=2244.87
## cmedv ~ 1
##
##           Df Sum of Sq   RSS    AIC
## + town    91   30545.5 12032 1787.4
## + lstat    1   23368.3 19209 1844.1
## + rm       1   20643.3 21934 1911.2
## + ptratio  1   10886.6 31691 2097.4
## + indus    1   10005.2 32573 2111.3
## + tax      1    9484.8 33093 2119.3
## + nox      1    7847.0 34731 2143.8
## + tract    1    7808.7 34769 2144.3
## + crim     1    6462.2 36116 2163.6
## + rad      1    6303.4 36274 2165.8
## + age      1    6083.6 36494 2168.8
## + zn       1    5529.9 37048 2176.5
## + b        1    4774.3 37803 2186.7
## + lon      1    4440.6 38137 2191.1
## + dis      1    2646.5 39931 2214.4
## + chas     1    1313.8 41264 2231.0
## <none>                 42578 2244.9
## + lat      1       2.0 42576 2246.8
##
```

```
## Step:  AIC=1787.42
## cmedv ~ town
##
##         Df Sum of Sq   RSS    AIC
## + rm     1    5704.0  6328 1464.3
## + lstat  1    4752.3  7280 1535.2
## + nox    1    2110.9  9921 1691.8
## + age    1     815.4 11217 1753.9
## + b      1     589.3 11443 1764.0
## + dis    1     471.5 11561 1769.2
## + lon    1     388.1 11644 1772.8
## + crim   1     230.1 11802 1779.7
## + tract  1     193.4 11839 1781.2
## + lat    1      48.1 11984 1787.4
## <none>               12032 1787.4
## + chas   1       6.4 12026 1789.2
## - town  91   30545.5 42578 2244.9
##
## Step:  AIC=1464.28
## cmedv ~ town + rm
##
##         Df Sum of Sq     RSS    AIC
## + lstat  1    1071.2  5257.0 1372.4
## + nox    1     758.1  5570.1 1401.7
## + age    1     565.3  5762.9 1418.9
## + b      1     413.7  5914.5 1432.1
## + tract  1     147.1  6181.1 1454.4
## + chas   1      99.6  6228.6 1458.2
## + lon    1      64.5  6263.7 1461.1
## + dis    1      36.9  6291.3 1463.3
## + lat    1      27.2  6301.1 1464.1
## <none>                6328.2 1464.3
## + crim   1      21.2  6307.0 1464.6
## - rm     1    5704.0 12032.2 1787.4
## - town  91   15606.2 21934.4 1911.2
##
## Step:  AIC=1372.43
## cmedv ~ town + rm + lstat
##
##         Df Sum of Sq     RSS    AIC
## + nox    1     374.3  4882.7 1337.1
## + b      1     218.0  5038.9 1353.0
## + age    1     167.5  5089.4 1358.0
## + tract  1      92.1  5164.9 1365.5
## + chas   1      42.7  5214.3 1370.3
## <none>                5257.0 1372.4
## + crim   1      12.5  5244.5 1373.2
## + lat    1       1.6  5255.4 1374.3
## + lon    1       1.4  5255.5 1374.3
## + dis    1       0.5  5256.5 1374.4
## - lstat  1    1071.2  6328.2 1464.3
## - rm     1    2023.0  7280.0 1535.2
## - town  91    9965.7 15222.7 1728.4
##
```

```
## Step:  AIC=1337.06
## cmedv ~ town + rm + lstat + nox
##
##         Df Sum of Sq     RSS    AIC
## + b      1     250.6  4632.1 1312.4
## + age    1     201.1  4681.6 1317.8
## + chas   1      36.6  4846.1 1335.2
## <none>                4882.7 1337.1
## + tract  1      12.1  4870.6 1337.8
## + crim   1      11.0  4871.7 1337.9
## + dis    1       9.3  4873.4 1338.1
## + lon    1       6.2  4876.5 1338.4
## + lat    1       0.0  4882.7 1339.1
## - nox    1     374.3  5257.0 1372.4
## - lstat  1     687.5  5570.1 1401.7
## - rm     1    1886.9  6769.6 1500.4
## - town  91   10325.1 15207.8 1729.9
##
## Step:  AIC=1312.4
## cmedv ~ town + rm + lstat + nox + b
##
##         Df Sum of Sq     RSS    AIC
## + age    1     263.5  4368.6 1284.8
## + chas   1      45.1  4587.0 1309.5
## <none>                4632.1 1312.4
## + tract  1      12.0  4620.1 1313.1
## + dis    1       5.4  4626.6 1313.8
## + crim   1       2.7  4629.4 1314.1
## + lat    1       0.4  4631.6 1314.3
## + lon    1       0.0  4632.1 1314.4
## - b      1     250.6  4882.7 1337.1
## - nox    1     406.9  5038.9 1353.0
## - lstat  1     521.5  5153.5 1364.4
## - rm     1    1952.4  6584.5 1488.4
## - town  91   10078.3 14710.4 1715.1
##
## Step:  AIC=1284.77
## cmedv ~ town + rm + lstat + nox + b + age
##
##         Df Sum of Sq     RSS    AIC
## + chas   1      28.4  4340.3 1283.5
## + dis    1      19.3  4349.3 1284.5
## <none>                4368.6 1284.8
## + tract  1      10.2  4358.4 1285.6
## + lat    1       7.1  4361.5 1285.9
## + lon    1       6.8  4361.8 1286.0
## + crim   1       1.5  4367.1 1286.6
## - lstat  1     227.5  4596.1 1308.5
## - age    1     263.5  4632.1 1312.4
## - b      1     313.0  4681.6 1317.8
## - nox    1     451.5  4820.1 1332.5
## - rm     1    2130.6  6499.2 1483.8
## - town  91   10299.3 14667.9 1715.6
##
```

```
## Step:  AIC=1283.47
## cmedv ~ town + rm + lstat + nox + b + age + chas
##
##         Df Sum of Sq     RSS    AIC
## + dis    1      19.5  4320.7 1283.2
## <none>                4340.3 1283.5
## + tract  1       9.5  4330.7 1284.4
## + lat    1       6.5  4333.8 1284.7
## - chas   1      28.4  4368.6 1284.8
## + lon    1       4.5  4335.8 1285.0
## + crim   1       1.4  4338.9 1285.3
## - lstat  1     216.7  4557.0 1306.1
## - age    1     246.7  4587.0 1309.5
## - b      1     318.5  4658.8 1317.3
## - nox    1     444.6  4784.9 1330.8
## - rm     1    2158.5  6498.8 1485.7
## - town  91    9828.6 14168.9 1700.1
##
## Step:  AIC=1283.19
## cmedv ~ town + rm + lstat + nox + b + age + chas + dis
##
##         Df Sum of Sq     RSS    AIC
## <none>                4320.7 1283.2
## + tract  1      14.7  4306.1 1283.5
## - dis    1      19.5  4340.3 1283.5
## + lat    1       9.2  4311.6 1284.1
## - chas   1      28.6  4349.3 1284.5
## + crim   1       2.0  4318.7 1285.0
## + lon    1       1.7  4319.0 1285.0
## - lstat  1     224.1  4544.8 1306.8
## - age    1     260.3  4581.0 1310.8
## - b      1     313.6  4634.3 1316.6
## - nox    1     464.1  4784.8 1332.8
## - rm     1    2178.0  6498.7 1487.7
## - town  91    9048.7 13369.4 1672.7
```

```
summary(fit6)
```

```
##
## Call:
## lm(formula = cmedv ~ town + rm + lstat + nox + b + age + chas +
##     dis, data = x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.8015  -1.2470  -0.0179   0.9426  16.8967
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  10.982721   4.230786   2.596 0.009776 **
## townAshland                  -4.394169   4.090273  -1.074 0.283326
## townBedford                  -2.389648   3.090451  -0.773 0.439832
## townBelmont                   7.768241   1.700326   4.569 6.52e-06 ***
## townBeverly                  -3.026896   2.858750  -1.059 0.290311
## townBoston Allston-Brighton   6.846599   2.070443   3.307 0.001027 **
```

```
## townBoston Back Bay          17.823629   2.118133   8.415 6.76e-16 ***
## townBoston Beacon Hill       29.277647   2.537023  11.540  < 2e-16 ***
## townBoston Charlestown       -7.202711   2.100746  -3.429 0.000669 ***
## townBoston Dorchester         2.205005   1.806961   1.220 0.223064
## townBoston Downtown           6.551903   1.949771   3.360 0.000852 ***
## townBoston East Boston        1.004088   1.817590   0.552 0.580958
## townBoston Forest Hills      -1.005363   1.829730  -0.549 0.582991
## townBoston Hyde Park         -0.953817   2.106608  -0.453 0.650953
## townBoston Mattapan           0.006125   1.845075   0.003 0.997353
## townBoston North End          8.640409   2.869271   3.011 0.002763 **
## townBoston Roxbury           -1.305000   1.793701  -0.728 0.467310
## townBoston Savin Hill        -0.323253   1.781587  -0.181 0.856112
## townBoston South Boston      -5.453970   1.800604  -3.029 0.002610 **
## townBoston West Roxbury      -2.065803   2.070342  -0.998 0.318966
## townBraintree                -3.265402   1.958017  -1.668 0.096143 .
## townBrookline                10.818405   1.751904   6.175 1.60e-09 ***
## townBurlington               -6.687729   2.415190  -2.769 0.005879 **
## townCambridge                 9.616578   1.779954   5.403 1.12e-07 ***
## townCanton                   -3.583020   2.599126  -1.379 0.168791
## townChelsea                  -2.433715   2.018035  -1.206 0.228525
## townCohasset                  1.309491   4.024364   0.325 0.745052
## townConcord                   0.121186   3.115536   0.039 0.968991
## townDanvers                  -3.755980   2.817115  -1.333 0.183188
## townDedham                   -2.115347   2.045248  -1.034 0.301622
## townDover                    12.351397   3.868374   3.193 0.001518 **
## townDuxbury                   3.113388   5.763654   0.540 0.589371
## townEverett                   1.134815   1.827390   0.621 0.534945
## townFramingham               -1.818352   3.000577  -0.606 0.544852
## townHamilton                 -2.496000   4.667113  -0.535 0.593076
## townHanover                  -4.937834   4.457404  -1.108 0.268610
## townHingham                  -1.175150   3.042092  -0.386 0.699479
## townHolbrook                 -6.615267   3.221465  -2.053 0.040663 *
## townHull                     -8.182719   3.871335  -2.114 0.035151 *
## townLexington                 0.744518   2.180765   0.341 0.732977
## townLincoln                  12.234225   3.742974   3.269 0.001173 **
## townLynn                     -2.545838   1.600409  -1.591 0.112444
## townLynnfield                -0.174783   2.892320  -0.060 0.951843
## townMalden                   -1.925813   1.664307  -1.157 0.247900
## townManchester                4.225961   4.689359   0.901 0.368025
## townMarblehead                3.130616   2.713477   1.154 0.249289
## townMarshfield               -4.061246   4.685173  -0.867 0.386546
## townMedfield                 -0.626823   4.115188  -0.152 0.879011
## townMedford                  -1.978451   1.604627  -1.233 0.218300
## townMelrose                  -4.082404   2.052222  -1.989 0.047341 *
## townMiddleton                -4.428609   4.147955  -1.068 0.286306
## townMillis                   -7.168981   4.109375  -1.745 0.081819 .
## townMilton                    0.645344   2.070782   0.312 0.755472
## townNahant                   -2.351914   3.537403  -0.665 0.506510
## townNatick                   -4.499124   2.542204  -1.770 0.077514 .
## townNeedham                   0.830392   2.105570   0.394 0.693508
## townNewton                    3.819356   1.507506   2.534 0.011665 *
## townNorfolk                  -5.067131   4.667822  -1.086 0.278323
## townNorth Reading            -6.432322   3.204408  -2.007 0.045375 *
## townNorwell                  -4.045587   4.395341  -0.920 0.357895
```

```
## townNorwood                  -2.503192   2.404581  -1.041 0.298489
## townPeabody                  -3.651868   2.160015  -1.691 0.091666 .
## townPembroke                 -4.127244   4.648317  -0.888 0.375119
## townQuincy                   -1.795939   1.579549  -1.137 0.256209
## townRandolph                 -6.037877   2.589481  -2.332 0.020204 *
## townReading                  -4.028838   2.373799  -1.697 0.090421 .
## townRevere                    0.477635   1.727642   0.276 0.782330
## townRockland                 -5.098421   3.662583  -1.392 0.164674
## townSalem                     0.726120   2.420858   0.300 0.764374
## townSargus                   -3.046968   2.079573  -1.465 0.143641
## townScituate                 -3.552204   3.905851  -0.909 0.363647
## townSharon                   -5.319918   3.295841  -1.614 0.107274
## townSherborn                  8.302901   3.860517   2.151 0.032086 *
## townSomerville               -0.564186   1.636404  -0.345 0.730444
## townStoneham                 -4.613555   2.280837  -2.023 0.043753 *
## townSudbury                   0.554775   3.502390   0.158 0.874221
## townSwampscott               -0.191454   2.794952  -0.068 0.945421
## townTopsfield                 2.202934   4.496742   0.490 0.624472
## townWakefield                -3.412869   2.165458  -1.576 0.115791
## townWalpole                  -4.648651   3.131766  -1.484 0.138489
## townWaltham                   1.518266   1.739308   0.873 0.383225
## townWatertown                 2.995086   2.131103   1.405 0.160661
## townWayland                   0.946968   3.088013   0.307 0.759260
## townWellesley                 7.079814   2.248921   3.148 0.001764 **
## townWenham                    1.545302   4.375677   0.353 0.724153
## townWeston                   10.730237   2.857889   3.755 0.000199 ***
## townWestwood                 -2.292796   2.506633  -0.915 0.360895
## townWeymouth                 -3.716740   2.138492  -1.738 0.082964 .
## townWilmington               -5.335794   2.893106  -1.844 0.065864 .
## townWinchester                0.380010   1.955289   0.194 0.845999
## townWinthrop                 -2.995073   1.959028  -1.529 0.127077
## townWoburn                   -5.909296   1.976387  -2.990 0.002960 **
## rm                            5.141924   0.358986  14.323  < 2e-16 ***
## lstat                        -0.215062   0.046807  -4.595 5.79e-06 ***
## nox                         -30.197923   4.567205  -6.612 1.19e-10 ***
## b                             0.013917   0.002561   5.435 9.47e-08 ***
## age                          -0.057944   0.011702  -4.952 1.08e-06 ***
## chas1                        -1.226567   0.747364  -1.641 0.101530
## dis                          -0.701003   0.516611  -1.357 0.175556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.258 on 407 degrees of freedom
## Multiple R-squared:  0.8985, Adjusted R-squared:  0.8741
## F-statistic: 36.77 on 98 and 407 DF,  p-value: < 2.2e-16
```

```
fit7<-step(fitmin,direction="forward",scope=formula(fit5))
```

```
## Start:  AIC=2244.87
## cmedv ~ 1
##
##           Df Sum of Sq   RSS    AIC
## + town    91   30545.5 12032 1787.4
## + lstat    1   23368.3 19209 1844.1
## + rm       1   20643.3 21934 1911.2
```

```
## + ptratio  1   10886.6 31691 2097.4
## + indus    1   10005.2 32573 2111.3
## + tax      1    9484.8 33093 2119.3
## + nox      1    7847.0 34731 2143.8
## + tract    1    7808.7 34769 2144.3
## + crim     1    6462.2 36116 2163.6
## + rad      1    6303.4 36274 2165.8
## + age      1    6083.6 36494 2168.8
## + zn       1    5529.9 37048 2176.5
## + b        1    4774.3 37803 2186.7
## + lon      1    4440.6 38137 2191.1
## + dis      1    2646.5 39931 2214.4
## + chas     1    1313.8 41264 2231.0
## <none>                  42578 2244.9
## + lat      1       2.0 42576 2246.8
##
## Step:  AIC=1787.42
## cmedv ~ town
##
##         Df Sum of Sq     RSS    AIC
## + rm     1    5704.0  6328.2 1464.3
## + lstat  1    4752.3  7280.0 1535.2
## + nox    1    2110.9  9921.3 1691.8
## + age    1     815.4 11216.8 1753.9
## + b      1     589.3 11442.9 1764.0
## + dis    1     471.5 11560.7 1769.2
## + lon    1     388.1 11644.1 1772.8
## + crim   1     230.1 11802.2 1779.7
## + tract  1     193.4 11838.8 1781.2
## + lat    1      48.1 11984.1 1787.4
## <none>                12032.2 1787.4
## + chas   1       6.4 12025.8 1789.2
##
## Step:  AIC=1464.28
## cmedv ~ town + rm
##
##         Df Sum of Sq    RSS    AIC
## + lstat  1   1071.24 5257.0 1372.4
## + nox    1    758.08 5570.1 1401.7
## + age    1    565.30 5762.9 1418.9
## + b      1    413.68 5914.5 1432.1
## + tract  1    147.13 6181.1 1454.4
## + chas   1     99.58 6228.6 1458.2
## + lon    1     64.51 6263.7 1461.1
## + dis    1     36.88 6291.3 1463.3
## + lat    1     27.15 6301.1 1464.1
## <none>                6328.2 1464.3
## + crim   1     21.22 6307.0 1464.6
##
## Step:  AIC=1372.43
## cmedv ~ town + rm + lstat
##
##         Df Sum of Sq    RSS    AIC
## + nox    1    374.30 4882.7 1337.1
```

```
## + b      1    218.04 5038.9 1353.0
## + age    1    167.54 5089.4 1358.0
## + tract  1     92.12 5164.9 1365.5
## + chas   1     42.69 5214.3 1370.3
## <none>               5257.0 1372.4
## + crim   1     12.49 5244.5 1373.2
## + lat    1      1.62 5255.4 1374.3
## + lon    1      1.43 5255.5 1374.3
## + dis    1      0.45 5256.5 1374.4
##
## Step:  AIC=1337.06
## cmedv ~ town + rm + lstat + nox
##
##         Df Sum of Sq    RSS    AIC
## + b      1   250.600 4632.1 1312.4
## + age    1   201.058 4681.6 1317.8
## + chas   1    36.615 4846.1 1335.2
## <none>               4882.7 1337.1
## + tract  1    12.115 4870.6 1337.8
## + crim   1    11.015 4871.7 1337.9
## + dis    1     9.276 4873.4 1338.1
## + lon    1     6.199 4876.5 1338.4
## + lat    1     0.012 4882.7 1339.1
##
## Step:  AIC=1312.4
## cmedv ~ town + rm + lstat + nox + b
##
##         Df Sum of Sq    RSS    AIC
## + age    1   263.456 4368.6 1284.8
## + chas   1    45.103 4587.0 1309.5
## <none>               4632.1 1312.4
## + tract  1    11.987 4620.1 1313.1
## + dis    1     5.436 4626.6 1313.8
## + crim   1     2.679 4629.4 1314.1
## + lat    1     0.440 4631.6 1314.3
## + lon    1     0.008 4632.1 1314.4
##
## Step:  AIC=1284.77
## cmedv ~ town + rm + lstat + nox + b + age
##
##         Df Sum of Sq    RSS    AIC
## + chas   1   28.3531 4340.3 1283.5
## + dis    1   19.3055 4349.3 1284.5
## <none>               4368.6 1284.8
## + tract  1   10.2494 4358.4 1285.6
## + lat    1    7.1131 4361.5 1285.9
## + lon    1    6.8429 4361.8 1286.0
## + crim   1    1.5124 4367.1 1286.6
##
## Step:  AIC=1283.47
## cmedv ~ town + rm + lstat + nox + b + age + chas
##
##         Df Sum of Sq    RSS    AIC
## + dis    1   19.5468 4320.7 1283.2
```

```
## <none>               4340.3 1283.5
## + tract  1    9.5423 4330.7 1284.4
## + lat    1    6.4590 4333.8 1284.7
## + lon    1    4.5097 4335.8 1285.0
## + crim   1    1.3872 4338.9 1285.3
##
## Step:  AIC=1283.19
## cmedv ~ town + rm + lstat + nox + b + age + chas + dis
##
##         Df Sum of Sq    RSS    AIC
## <none>               4320.7 1283.2
## + tract  1   14.6700 4306.1 1283.5
## + lat    1    9.1782 4311.6 1284.1
## + crim   1    2.0154 4318.7 1285.0
## + lon    1    1.6876 4319.0 1285.0
```

```
summary(fit7)
```

```
##
## Call:
## lm(formula = cmedv ~ town + rm + lstat + nox + b + age + chas +
##     dis, data = x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.8015  -1.2470  -0.0179   0.9426  16.8967
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 10.982721   4.230786   2.596 0.009776 **
## townAshland                 -4.394169   4.090273  -1.074 0.283326
## townBedford                 -2.389648   3.090451  -0.773 0.439832
## townBelmont                  7.768241   1.700326   4.569 6.52e-06 ***
## townBeverly                 -3.026896   2.858750  -1.059 0.290311
## townBoston Allston-Brighton  6.846599   2.070443   3.307 0.001027 **
## townBoston Back Bay         17.823629   2.118133   8.415 6.76e-16 ***
## townBoston Beacon Hill      29.277647   2.537023  11.540  < 2e-16 ***
## townBoston Charlestown      -7.202711   2.100746  -3.429 0.000669 ***
## townBoston Dorchester        2.205005   1.806961   1.220 0.223064
## townBoston Downtown          6.551903   1.949771   3.360 0.000852 ***
## townBoston East Boston       1.004088   1.817590   0.552 0.580958
## townBoston Forest Hills     -1.005363   1.829730  -0.549 0.582991
## townBoston Hyde Park        -0.953817   2.106608  -0.453 0.650953
## townBoston Mattapan          0.006125   1.845075   0.003 0.997353
## townBoston North End         8.640409   2.869271   3.011 0.002763 **
## townBoston Roxbury          -1.305000   1.793701  -0.728 0.467310
## townBoston Savin Hill       -0.323253   1.781587  -0.181 0.856112
## townBoston South Boston     -5.453970   1.800604  -3.029 0.002610 **
## townBoston West Roxbury     -2.065803   2.070342  -0.998 0.318966
## townBraintree               -3.265402   1.958017  -1.668 0.096143 .
## townBrookline               10.818405   1.751904   6.175 1.60e-09 ***
## townBurlington              -6.687729   2.415190  -2.769 0.005879 **
## townCambridge                9.616578   1.779954   5.403 1.12e-07 ***
## townCanton                  -3.583020   2.599126  -1.379 0.168791
## townChelsea                 -2.433715   2.018035  -1.206 0.228525
```

```
## townCohasset                1.309491   4.024364   0.325 0.745052
## townConcord                 0.121186   3.115536   0.039 0.968991
## townDanvers                -3.755980   2.817115  -1.333 0.183188
## townDedham                 -2.115347   2.045248  -1.034 0.301622
## townDover                  12.351397   3.868374   3.193 0.001518 **
## townDuxbury                 3.113388   5.763654   0.540 0.589371
## townEverett                 1.134815   1.827390   0.621 0.534945
## townFramingham             -1.818352   3.000577  -0.606 0.544852
## townHamilton               -2.496000   4.667113  -0.535 0.593076
## townHanover                -4.937834   4.457404  -1.108 0.268610
## townHingham                -1.175150   3.042092  -0.386 0.699479
## townHolbrook               -6.615267   3.221465  -2.053 0.040663 *
## townHull                   -8.182719   3.871335  -2.114 0.035151 *
## townLexington               0.744518   2.180765   0.341 0.732977
## townLincoln                12.234225   3.742974   3.269 0.001173 **
## townLynn                   -2.545838   1.600409  -1.591 0.112444
## townLynnfield              -0.174783   2.892320  -0.060 0.951843
## townMalden                 -1.925813   1.664307  -1.157 0.247900
## townManchester              4.225961   4.689359   0.901 0.368025
## townMarblehead              3.130616   2.713477   1.154 0.249289
## townMarshfield             -4.061246   4.685173  -0.867 0.386546
## townMedfield               -0.626823   4.115188  -0.152 0.879011
## townMedford                -1.978451   1.604627  -1.233 0.218300
## townMelrose                -4.082404   2.052222  -1.989 0.047341 *
## townMiddleton              -4.428609   4.147955  -1.068 0.286306
## townMillis                 -7.168981   4.109375  -1.745 0.081819 .
## townMilton                  0.645344   2.070782   0.312 0.755472
## townNahant                 -2.351914   3.537403  -0.665 0.506510
## townNatick                 -4.499124   2.542204  -1.770 0.077514 .
## townNeedham                 0.830392   2.105570   0.394 0.693508
## townNewton                  3.819356   1.507506   2.534 0.011665 *
## townNorfolk                -5.067131   4.667822  -1.086 0.278323
## townNorth Reading          -6.432322   3.204408  -2.007 0.045375 *
## townNorwell                -4.045587   4.395341  -0.920 0.357895
## townNorwood                -2.503192   2.404581  -1.041 0.298489
## townPeabody                -3.651868   2.160015  -1.691 0.091666 .
## townPembroke               -4.127244   4.648317  -0.888 0.375119
## townQuincy                 -1.795939   1.579549  -1.137 0.256209
## townRandolph               -6.037877   2.589481  -2.332 0.020204 *
## townReading                -4.028838   2.373799  -1.697 0.090421 .
## townRevere                  0.477635   1.727642   0.276 0.782330
## townRockland               -5.098421   3.662583  -1.392 0.164674
## townSalem                   0.726120   2.420858   0.300 0.764374
## townSargus                 -3.046968   2.079573  -1.465 0.143641
## townScituate               -3.552204   3.905851  -0.909 0.363647
## townSharon                 -5.319918   3.295841  -1.614 0.107274
## townSherborn                8.302901   3.860517   2.151 0.032086 *
## townSomerville             -0.564186   1.636404  -0.345 0.730444
## townStoneham               -4.613555   2.280837  -2.023 0.043753 *
## townSudbury                 0.554775   3.502390   0.158 0.874221
## townSwampscott             -0.191454   2.794952  -0.068 0.945421
## townTopsfield               2.202934   4.496742   0.490 0.624472
## townWakefield              -3.412869   2.165458  -1.576 0.115791
## townWalpole                -4.648651   3.131766  -1.484 0.138489
```

```
## townWaltham                   1.518266   1.739308   0.873 0.383225
## townWatertown                 2.995086   2.131103   1.405 0.160661
## townWayland                   0.946968   3.088013   0.307 0.759260
## townWellesley                 7.079814   2.248921   3.148 0.001764 **
## townWenham                    1.545302   4.375677   0.353 0.724153
## townWeston                   10.730237   2.857889   3.755 0.000199 ***
## townWestwood                 -2.292796   2.506633  -0.915 0.360895
## townWeymouth                 -3.716740   2.138492  -1.738 0.082964 .
## townWilmington               -5.335794   2.893106  -1.844 0.065864 .
## townWinchester                0.380010   1.955289   0.194 0.845999
## townWinthrop                 -2.995073   1.959028  -1.529 0.127077
## townWoburn                   -5.909296   1.976387  -2.990 0.002960 **
## rm                            5.141924   0.358986  14.323  < 2e-16 ***
## lstat                        -0.215062   0.046807  -4.595 5.79e-06 ***
## nox                         -30.197923   4.567205  -6.612 1.19e-10 ***
## b                             0.013917   0.002561   5.435 9.47e-08 ***
## age                          -0.057944   0.011702  -4.952 1.08e-06 ***
## chas1                        -1.226567   0.747364  -1.641 0.101530
## dis                          -0.701003   0.516611  -1.357 0.175556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.258 on 407 degrees of freedom
## Multiple R-squared:  0.8985, Adjusted R-squared:  0.8741
## F-statistic: 36.77 on 98 and 407 DF,  p-value: < 2.2e-16
```

```
fit8 <- step(fit5,direction="backward",scope=formula(fit5))
```

```
## Start:  AIC=1288.39
## cmedv ~ town + tract + lon + lat + crim + zn + indus + chas +
##     nox + rm + age + dis + rad + tax + ptratio + b + lstat
##
##
## Step:  AIC=1288.39
## cmedv ~ town + tract + lon + lat + crim + zn + indus + chas +
##     nox + rm + age + dis + rad + tax + b + lstat
##
##
## Step:  AIC=1288.39
## cmedv ~ town + tract + lon + lat + crim + zn + indus + chas +
##     nox + rm + age + dis + rad + b + lstat
##
##
## Step:  AIC=1288.39
## cmedv ~ town + tract + lon + lat + crim + zn + indus + chas +
##     nox + rm + age + dis + b + lstat
##
##
## Step:  AIC=1288.39
## cmedv ~ town + tract + lon + lat + crim + zn + chas + nox + rm +
##     age + dis + b + lstat
##
##
## Step:  AIC=1288.39
## cmedv ~ town + tract + lon + lat + crim + chas + nox + rm + age +
```

```
##     dis + b + lstat
##
##         Df Sum of Sq     RSS    AIC
## - lon    1       0.4  4297.3 1286.4
## - crim   1       1.2  4298.2 1286.5
## - lat    1       6.7  4303.6 1287.2
## - tract  1      12.2  4309.1 1287.8
## <none>                4296.9 1288.4
## - dis    1      24.8  4321.8 1289.3
## - chas   1      26.2  4323.1 1289.5
## - lstat  1     230.1  4527.1 1312.8
## - age    1     264.4  4561.4 1316.6
## - b      1     298.1  4595.0 1320.3
## - nox    1     376.6  4673.5 1328.9
## - rm     1    2148.1  6445.0 1491.5
## - town  91    8482.2 12779.2 1657.9
##
## Step:  AIC=1286.44
## cmedv ~ town + tract + lat + crim + chas + nox + rm + age + dis +
##     b + lstat
##
##         Df Sum of Sq     RSS    AIC
## - crim   1       1.3  4298.6 1284.6
## - lat    1       6.9  4304.2 1285.2
## - tract  1      12.8  4310.2 1286.0
## <none>                4297.3 1286.4
## - chas   1      27.0  4324.4 1287.6
## - dis    1      27.5  4324.8 1287.7
## - lstat  1     230.6  4528.0 1310.9
## - age    1     266.9  4564.2 1314.9
## - b      1     307.5  4604.8 1319.4
## - nox    1     384.6  4682.0 1327.8
## - rm     1    2149.0  6446.3 1489.6
## - town  91    8835.9 13133.2 1669.7
##
## Step:  AIC=1284.59
## cmedv ~ town + tract + lat + chas + nox + rm + age + dis + b +
##     lstat
##
##         Df Sum of Sq     RSS    AIC
## - lat    1       7.4  4306.1 1283.5
## - tract  1      12.9  4311.6 1284.1
## <none>                4298.6 1284.6
## - dis    1      27.0  4325.7 1285.8
## - chas   1      27.1  4325.7 1285.8
## - lstat  1     230.6  4529.2 1309.0
## - age    1     268.2  4566.8 1313.2
## - b      1     316.3  4614.9 1318.5
## - nox    1     385.2  4683.8 1326.0
## - rm     1    2181.9  6480.6 1490.3
## - town  91    8936.9 13235.5 1671.7
##
## Step:  AIC=1283.47
## cmedv ~ town + tract + chas + nox + rm + age + dis + b + lstat
```

```
##
##         Df Sum of Sq     RSS    AIC
## - tract  1      14.7  4320.7 1283.2
## <none>                4306.1 1283.5
## - dis    1      24.7  4330.7 1284.4
## - chas   1      27.7  4333.8 1284.7
## - lstat  1     227.4  4533.5 1307.5
## - age    1     261.0  4567.0 1311.2
## - b      1     312.4  4618.4 1316.9
## - nox    1     377.8  4683.9 1324.0
## - rm     1    2191.5  6497.6 1489.6
## - town  91    8988.4 13294.5 1671.9
##
## Step:  AIC=1283.19
## cmedv ~ town + chas + nox + rm + age + dis + b + lstat
##
##         Df Sum of Sq     RSS    AIC
## <none>                4320.7 1283.2
## - dis    1      19.5  4340.3 1283.5
## - chas   1      28.6  4349.3 1284.5
## - lstat  1     224.1  4544.8 1306.8
## - age    1     260.3  4581.0 1310.8
## - b      1     313.6  4634.3 1316.6
## - nox    1     464.1  4784.8 1332.8
## - rm     1    2178.0  6498.7 1487.7
## - town  91    9048.7 13369.4 1672.7
```

```
summary(fit8)
```

```
##
## Call:
## lm(formula = cmedv ~ town + chas + nox + rm + age + dis + b +
##     lstat, data = x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.8015  -1.2470  -0.0179   0.9426  16.8967
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  10.982721   4.230786   2.596 0.009776 **
## townAshland                  -4.394169   4.090273  -1.074 0.283326
## townBedford                  -2.389648   3.090451  -0.773 0.439832
## townBelmont                   7.768241   1.700326   4.569 6.52e-06 ***
## townBeverly                  -3.026896   2.858750  -1.059 0.290311
## townBoston Allston-Brighton   6.846599   2.070443   3.307 0.001027 **
## townBoston Back Bay          17.823629   2.118133   8.415 6.76e-16 ***
## townBoston Beacon Hill       29.277647   2.537023  11.540  < 2e-16 ***
## townBoston Charlestown       -7.202711   2.100746  -3.429 0.000669 ***
## townBoston Dorchester         2.205005   1.806961   1.220 0.223064
## townBoston Downtown           6.551903   1.949771   3.360 0.000852 ***
## townBoston East Boston        1.004088   1.817590   0.552 0.580958
## townBoston Forest Hills      -1.005363   1.829730  -0.549 0.582991
## townBoston Hyde Park         -0.953817   2.106608  -0.453 0.650953
## townBoston Mattapan           0.006125   1.845075   0.003 0.997353
```

```
## townBoston North End          8.640409   2.869271   3.011 0.002763 **
## townBoston Roxbury           -1.305000   1.793701  -0.728 0.467310
## townBoston Savin Hill        -0.323253   1.781587  -0.181 0.856112
## townBoston South Boston      -5.453970   1.800604  -3.029 0.002610 **
## townBoston West Roxbury      -2.065803   2.070342  -0.998 0.318966
## townBraintree                -3.265402   1.958017  -1.668 0.096143 .
## townBrookline                10.818405   1.751904   6.175 1.60e-09 ***
## townBurlington               -6.687729   2.415190  -2.769 0.005879 **
## townCambridge                 9.616578   1.779954   5.403 1.12e-07 ***
## townCanton                   -3.583020   2.599126  -1.379 0.168791
## townChelsea                  -2.433715   2.018035  -1.206 0.228525
## townCohasset                  1.309491   4.024364   0.325 0.745052
## townConcord                   0.121186   3.115536   0.039 0.968991
## townDanvers                  -3.755980   2.817115  -1.333 0.183188
## townDedham                   -2.115347   2.045248  -1.034 0.301622
## townDover                    12.351397   3.868374   3.193 0.001518 **
## townDuxbury                   3.113388   5.763654   0.540 0.589371
## townEverett                   1.134815   1.827390   0.621 0.534945
## townFramingham               -1.818352   3.000577  -0.606 0.544852
## townHamilton                 -2.496000   4.667113  -0.535 0.593076
## townHanover                  -4.937834   4.457404  -1.108 0.268610
## townHingham                  -1.175150   3.042092  -0.386 0.699479
## townHolbrook                 -6.615267   3.221465  -2.053 0.040663 *
## townHull                     -8.182719   3.871335  -2.114 0.035151 *
## townLexington                 0.744518   2.180765   0.341 0.732977
## townLincoln                  12.234225   3.742974   3.269 0.001173 **
## townLynn                     -2.545838   1.600409  -1.591 0.112444
## townLynnfield                -0.174783   2.892320  -0.060 0.951843
## townMalden                   -1.925813   1.664307  -1.157 0.247900
## townManchester                4.225961   4.689359   0.901 0.368025
## townMarblehead                3.130616   2.713477   1.154 0.249289
## townMarshfield               -4.061246   4.685173  -0.867 0.386546
## townMedfield                 -0.626823   4.115188  -0.152 0.879011
## townMedford                  -1.978451   1.604627  -1.233 0.218300
## townMelrose                  -4.082404   2.052222  -1.989 0.047341 *
## townMiddleton                -4.428609   4.147955  -1.068 0.286306
## townMillis                   -7.168981   4.109375  -1.745 0.081819 .
## townMilton                    0.645344   2.070782   0.312 0.755472
## townNahant                   -2.351914   3.537403  -0.665 0.506510
## townNatick                   -4.499124   2.542204  -1.770 0.077514 .
## townNeedham                   0.830392   2.105570   0.394 0.693508
## townNewton                    3.819356   1.507506   2.534 0.011665 *
## townNorfolk                  -5.067131   4.667822  -1.086 0.278323
## townNorth Reading            -6.432322   3.204408  -2.007 0.045375 *
## townNorwell                  -4.045587   4.395341  -0.920 0.357895
## townNorwood                  -2.503192   2.404581  -1.041 0.298489
## townPeabody                  -3.651868   2.160015  -1.691 0.091666 .
## townPembroke                 -4.127244   4.648317  -0.888 0.375119
## townQuincy                   -1.795939   1.579549  -1.137 0.256209
## townRandolph                 -6.037877   2.589481  -2.332 0.020204 *
## townReading                  -4.028838   2.373799  -1.697 0.090421 .
## townRevere                    0.477635   1.727642   0.276 0.782330
## townRockland                 -5.098421   3.662583  -1.392 0.164674
## townSalem                     0.726120   2.420858   0.300 0.764374
```

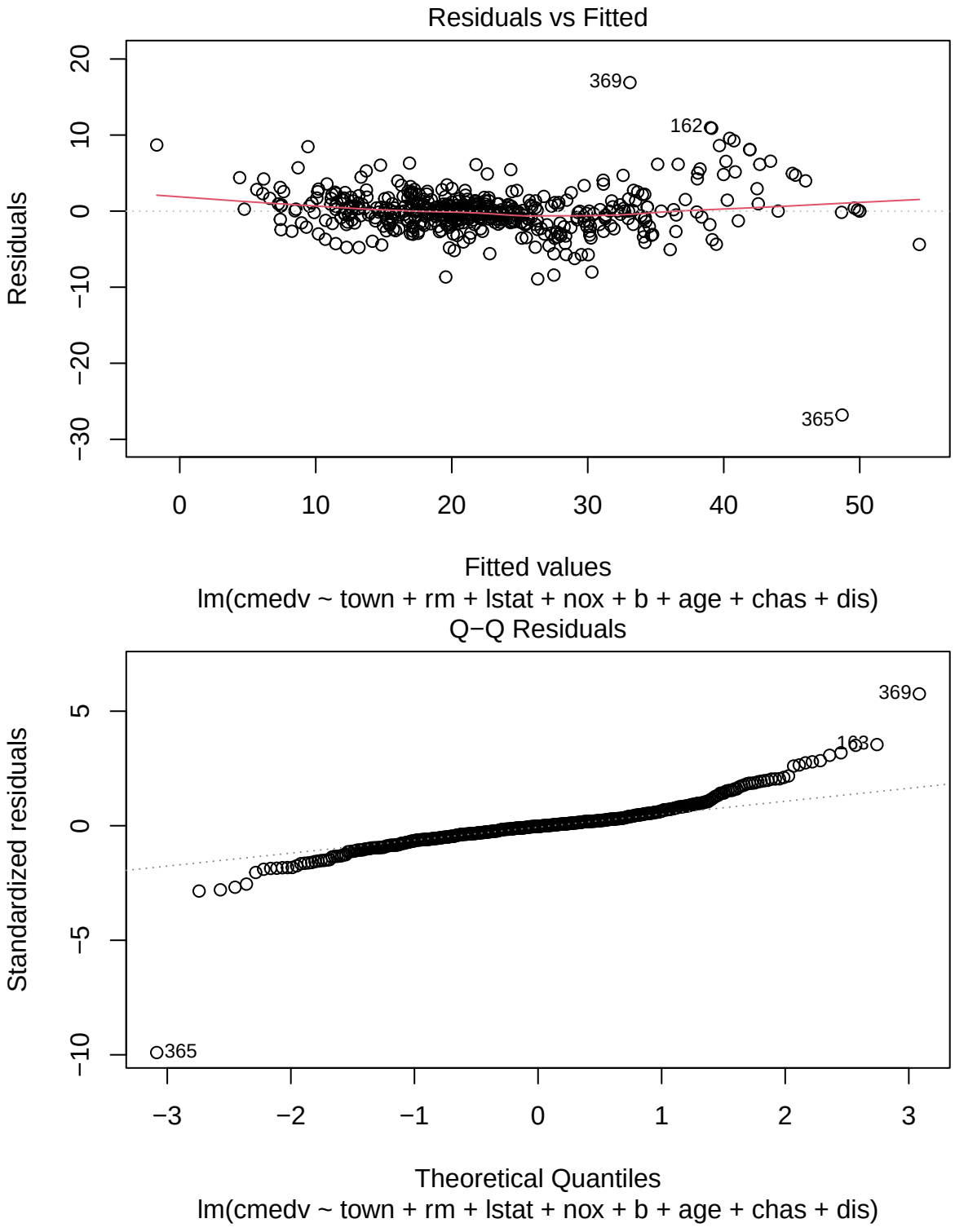
```
## townSargus                  -3.046968   2.079573  -1.465 0.143641
## townScituate                -3.552204   3.905851  -0.909 0.363647
## townSharon                  -5.319918   3.295841  -1.614 0.107274
## townSherborn                 8.302901   3.860517   2.151 0.032086 *
## townSomerville              -0.564186   1.636404  -0.345 0.730444
## townStoneham                -4.613555   2.280837  -2.023 0.043753 *
## townSudbury                  0.554775   3.502390   0.158 0.874221
## townSwampscott              -0.191454   2.794952  -0.068 0.945421
## townTopsfield                2.202934   4.496742   0.490 0.624472
## townWakefield               -3.412869   2.165458  -1.576 0.115791
## townWalpole                 -4.648651   3.131766  -1.484 0.138489
## townWaltham                  1.518266   1.739308   0.873 0.383225
## townWatertown                2.995086   2.131103   1.405 0.160661
## townWayland                  0.946968   3.088013   0.307 0.759260
## townWellesley                7.079814   2.248921   3.148 0.001764 **
## townWenham                   1.545302   4.375677   0.353 0.724153
## townWeston                  10.730237   2.857889   3.755 0.000199 ***
## townWestwood                -2.292796   2.506633  -0.915 0.360895
## townWeymouth                -3.716740   2.138492  -1.738 0.082964 .
## townWilmington              -5.335794   2.893106  -1.844 0.065864 .
## townWinchester               0.380010   1.955289   0.194 0.845999
## townWinthrop                -2.995073   1.959028  -1.529 0.127077
## townWoburn                  -5.909296   1.976387  -2.990 0.002960 **
## chas1                       -1.226567   0.747364  -1.641 0.101530
## nox                        -30.197923   4.567205  -6.612 1.19e-10 ***
## rm                           5.141924   0.358986  14.323  < 2e-16 ***
## age                         -0.057944   0.011702  -4.952 1.08e-06 ***
## dis                         -0.701003   0.516611  -1.357 0.175556
## b                            0.013917   0.002561   5.435 9.47e-08 ***
## lstat                       -0.215062   0.046807  -4.595 5.79e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.258 on 407 degrees of freedom
## Multiple R-squared:  0.8985, Adjusted R-squared:  0.8741
## F-statistic: 36.77 on 98 and 407 DF,  p-value: < 2.2e-16
```

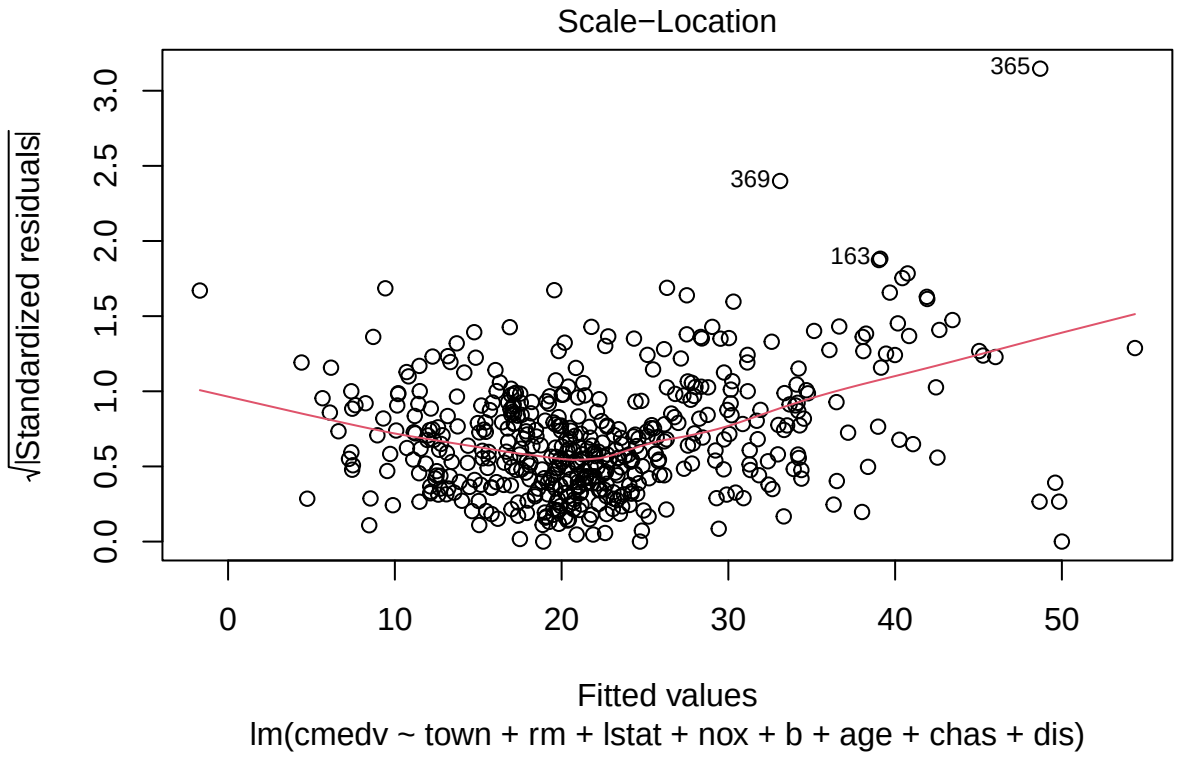
```
aic <- c(AIC(fit1),AIC(fit2),AIC(fit5),AIC(fit6))
names(aic) <- c(formula(fit1),formula(fit2),"Full model","Stepwise")
round(aic,4)
```

```
## cmedv ~ lstat   cmedv ~ lat    Full model      Stepwise
##      3282.096      3684.813      2726.360      2721.155
```

```
plot(fit6)
```

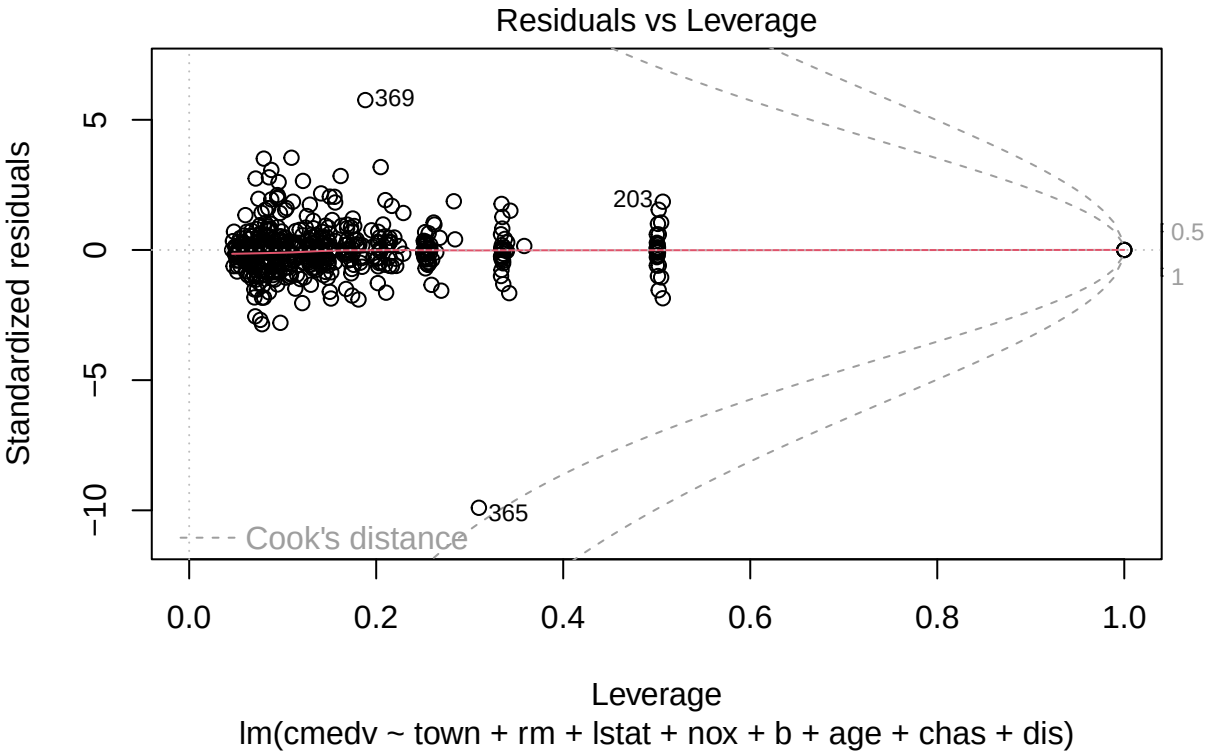
```
## Warning: not plotting observations with leverage one:
##   1, 56, 58, 65, 257, 284, 285, 286, 287, 342, 343, 348, 349, 354
```


Residuals vs Fitted
369
162
365
Residuals
20
10
0
−10
−20
−30
0
10
20
30
40
50
Fitted values
lm(cmedv ~ town + rm + lstat + nox + b + age + chas + dis)
Q−Q Residuals
369
163
365
Standardized residuals
5
0
−5
−10
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(cmedv ~ town + rm + lstat + nox + b + age + chas + dis)




```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

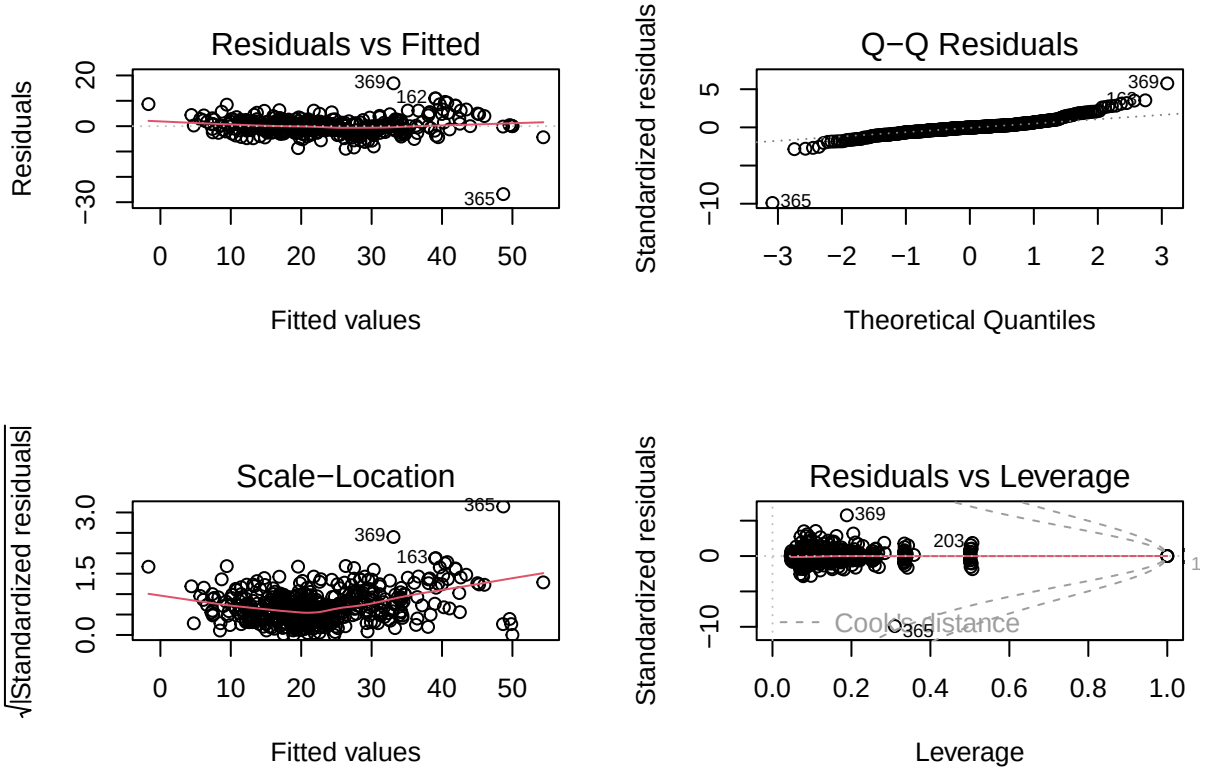
```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```




```
par(mfrow=c(2,2)) # Split into 2 by 2
plot(fit6) # Plot
```

```
## Warning: not plotting observations with leverage one:
##   1, 56, 58, 65, 257, 284, 285, 286, 287, 342, 343, 348, 349, 354
```

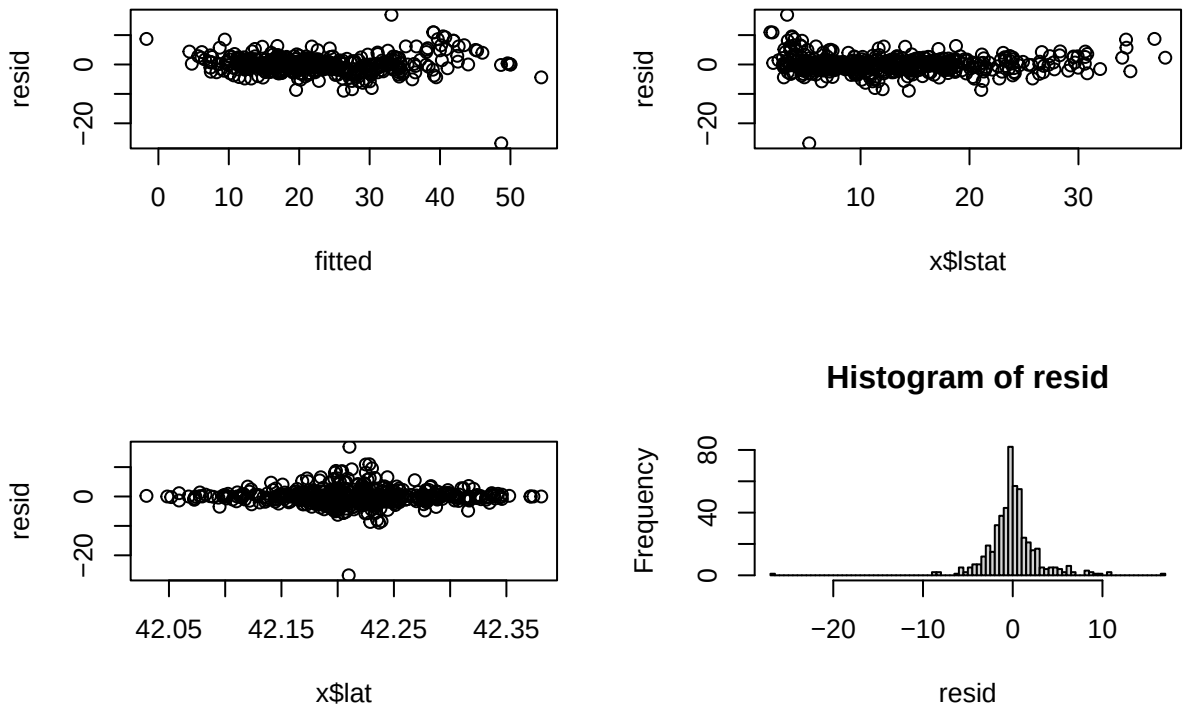
```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```




```
par(mfrow=c(1,1)) #Revert to a single plot. Otherwise it will keep on plotting on a 2 by 2 matrix.
```

```
resid<-fit6$residuals
fitted<-fit6$fitted.values
```

```
par(mfrow=c(2,2))
plot(fitted,resid) #Scatter plot fitted vs. residuals
plot(x$lstat,resid) #Scatter plot lstat vs. residuals
plot(x$lat,resid) #Scatter lat vs. residuals
hist(resid,100) #Historgram of residuals with 100 bins
```




```
par(mfrow=c(1,1))
```

Predicting with regression

```
idx <- sort(sample(1:nrow(x),100))
xTest <- x[idx,]
xTrain <- x[-idx,]
```

```
fitTrain <- lm(cmedv ~ lstat + lat, data=xTrain)
fitTrain
```

```
##
## Call:
## lm(formula = cmedv ~ lstat + lat, data = xTrain)
##
## Coefficients:
## (Intercept)        lstat          lat
##     -14.819       -0.999        1.184
```

```
predict(fitTrain,newdata=xTest)
```

```
##          3         19         23         34         40         46         48
## 31.2297029 23.5767528 16.5496214 16.9156995 30.9755203 25.0848831 16.5093743
##         50         52         55         56         57         62         65
## 19.1166098 25.9004620 20.5752556 30.5665395 29.5992074 20.8827885 27.2871382
##         70         83         84        111        112        116        119
## 26.5265596 28.5933102 27.7934399 22.2420058 25.0797646 19.4776538 19.8714096
##        125        129        130        131        137        142        146
## 17.6482189 19.8241914 16.8792661 22.6161450 18.3139214  0.8131005  7.4076240
##        160        166        168        173        184        196        200
## 27.8011699 25.3921110 23.0703568 20.5313092 29.5185828 32.2525252 30.6522688
```

```
##        215        217        219        221        223        225        232
##  5.6783182 21.6881045 17.2813199 25.4732997 25.2531639 31.0286239 29.9043358
##        238        241        245        246        247        264        266
## 30.4356597 23.7461076 22.6260407 16.6705656 25.9698142 23.9162431 24.7178135
##        268        271        282        288        289        290        299
## 27.7159285 22.1004856 30.5539663 27.8794382 27.4394376 25.5360802 30.0413520
##        302        309        314        319        323        327        332
## 25.5425174 30.5692174 27.2194382 24.7485102 27.3901057 28.9158213 22.6053743
##        333        334        337        338        340        342        343
## 27.2043378 29.3640356 25.2818976 24.5214716 25.3607861 29.6095067 26.4686469
##        344        348        349        351        362        368        371
## 27.9223770 28.6610286 29.0568318 29.0826911 20.9933874 21.8489766 32.2173945
##        373        375        377        383        394        396        397
## 26.3078009 -2.7525852 11.9713645 11.6149212 20.0037067 18.0539942 15.8058836
##        398        399        407        409        412        422        427
## 15.2601036  4.6085647 11.8641218  8.7896475 13.9642428 19.4751829 19.4755803
##        439        451        454        458        472        476        478
##  1.1639858 17.7122860 18.4068506 18.1987602 22.2644629 11.0671018 10.2627655
##        482        497
## 27.3931346 14.1006526
```

```
predict(fitTrain) #No new data input uses the training set
```

```
##          1          2          4          5          6          7          8
## 30.2474912 26.1301302 32.3304582 29.9487639 30.0757498 22.8546630 16.1567628
##          9         10         11         12         13         14         15
##  5.3898860 18.2118232 14.8651653 22.0391864 19.5885938 27.0145814 25.0106554
##         16         17         18         20         21         22         24
## 26.8071595 28.7006031 20.6026865 23.9805409 14.2495475 21.4371114 15.3907787
##         25         26         27         28         29         30         31
## 18.9702862 18.7633380 20.4651947 17.9923297 22.4648996 23.2817130 12.6734923
##         32         33         35         36         37         38         39
## 22.2209941  7.5656297 14.9229477 25.5699442 23.8469993 26.4985768 25.1411178
##         41         42         43         44         45         47         49
## 33.3262128 30.4820931 29.4994417 27.8713047 25.7554752 21.1488138  4.5160932
##         51         53         54         58         59         60         61
## 21.8794986 30.0555590 26.9045567 31.4144310 28.4771353 26.1029100 22.1827521
##         63         64         66         67         68         69         71
## 28.5904258 25.8314792 30.6637662 25.1082051 27.2267665 22.2462451 28.5814675
##         72         73         74         75         76         77         78
## 25.4066192 29.7672434 27.7338631 28.4895521 26.3299306 23.2982165 25.0036261
##         79         80         81         82         85         86         87
## 22.9422047 26.1908150 30.0005667 28.0807821 25.6742944 28.7523297 22.4345662
##         88         89         90         91         92         93         94
## 26.8306161 29.7571430 29.5560398 26.4414448 27.0502441 27.1038233 29.0672733
##         95         96         97         98         99        100        101
## 24.6774318 28.6058033 23.9204823 31.0362634 31.6720721 29.0540938 25.8113286
##        102        103        104        105        106        107        108
## 27.5483322 24.5989830 21.7888257 22.8965340 18.7541509 16.5670463 21.1365135
##        109        110        113        114        115        117        118
## 22.9594349 19.6868521 19.0263263 18.1531256 24.7784481 23.2014034 24.9410886
##        120        121        122        123        124        126        127
## 21.6326144 20.8589245 20.9614302 17.3009366  9.8242542 20.4132053  7.9713628
##        128        132        133        134        135        136        138
## 18.0265793 22.9569900 24.0964449 20.1956749 17.9126203 18.2571788 20.6204326
```

```
##        139        140        141        143        144        145        147
## 13.8912254 16.7513328 11.0584406  8.3904359  8.7909843  5.9208869 18.5488686
##        148        149        150        151        152        153        154
##  5.6831396  6.8940641 13.7560259 21.0994037 21.9149144 23.0696122 19.4009050
##        155        156        157        158        159        161        162
## 20.0716577 20.1701368 19.0571756 30.6057189 28.7628175 29.6882185 33.4625103
##        163        164        165        167        169        170        171
## 33.2689102 31.8703070 23.5603839 31.4971995 24.1076613 23.8902493 20.7853653
##        172        174        175        176        177        178        179
## 23.1855763 26.1796989 25.5832582 29.9008010 25.1226091 28.9342979 28.2999525
##        180        181        182        183        185        186        187
## 30.1707344 27.6457876 25.7502123 30.3761852 21.2268633 22.0619566 30.7641723
##        188        189        190        191        192        193        194
## 28.5509636 30.6670721 29.8498613 30.1453749 30.5626636 32.3832164 30.2508332
##        195        197        198        199        201        202        203
## 30.9037375 31.1661337 26.6418374 28.6493925 30.7384736 27.7294714 32.0718076
##        204        205        206        207        208        209        210
## 31.3890857 32.2871300 24.3509325 24.2348078 17.1471441 20.5380676 12.1161225
##        211        212        213        214        216        218        220
## 17.9255791 11.2268920 19.1729872 25.8287876 25.7296401 25.5048856 24.6897724
##        222        224        226        227        228        229        230
## 13.7384566 27.5866423 30.5290465 32.0334713 28.7974565 31.2235348 31.3782828
##        231        233        234        235        236        237        239
## 23.5047995 32.6834572 31.2160658 27.1322358 24.3049407 25.6399326 28.7889297
##        240        242        243        244        248        249        250
## 27.7793452 22.7277174 23.9227646 29.9399984 24.9823414 25.6192922 28.5832079
##        251        252        253        254        255        256        257
## 29.2526157 31.5688379 31.6232120 31.5885891 28.5436111 25.8552711 31.9853559
##        258        259        260        261        262        263        265
## 30.0511412 27.3842786 28.2731538 25.5816926 27.9069995 29.2532841 27.0690218
##        267        269        270        272        273        274        275
## 20.3779984 32.0094226 21.4665296 28.4975772 27.3693729 28.5259234 31.5924210
##        276        277        278        279        280        281        283
## 32.1487410 29.0698427 30.9695630 27.9524153 30.2980154 31.3857437 32.1190078
##        284        285        286        287        291        292        293
## 31.9383664 27.2092275 26.8171717 22.0833720 31.7401136 31.5308309 30.3800071
##        294        295        296        297        298        300        301
## 26.4793636 24.6750353 28.7948752 27.6691238 19.2177246 30.2551349 28.9436337
##        303        304        305        306        307        308        310
## 26.3799792 30.2074950 28.1691669 26.1800443 28.6452878 27.5833846 25.1499638
##        311        312        313        315        316        317        318
## 22.5015759 29.1573000 23.4099996 25.8309856 23.6191217 16.7846851 19.1651951
##        320        321        322        324        325        326        328
## 22.3820589 27.8804880 28.2160801 23.3550836 28.9564498 29.9718453 22.2871828
##        329        330        331        335        336        339        341
## 25.0943032 27.7240479 25.9556612 28.2968795 27.0547163 26.5817428 25.8177980
##        345        346        347        350        352        353        354
## 30.4607984 24.4868989 22.3608766 29.1903653 29.5277921 27.2081777 30.4824606
##        355        356        357        358        359        360        361
## 26.9401472 29.3916191 17.5953162 21.9148381 23.7006838 22.5083182 27.3790678
##        363        364        365        366        367        369        370
## 24.9870283 20.5497577 29.8842714 28.0533733 21.1775134 31.9130752 31.4459126
##        372        374        376        378        379        380        381
## 25.6558440  0.4465909 21.7592190 13.9699614 11.5203924 13.4316844 18.0000857
```

```
##        382        384        385        386        387        388        389
## 14.1327624 10.6538657  4.5857765  4.4033507  6.9329583  3.2286729  4.5981351
##        390        391        392        393        395        400        401
## 14.3614671 18.0999860 16.4551849  9.5272853 18.8228707  5.2210774  8.4153979
##        402        403        404        405        406        408        410
## 14.8594368 14.8662293 15.4047431  7.8048222 12.2070645 23.0501480 15.4042273
##        411        413        414        415        416        417        418
## 25.0663563  0.8252302 15.0991973 -1.7874951  6.1308038  9.3889726  8.5447945
##        419        420        421        423        424        425        426
## 14.5566570 12.4399563 20.1511885 21.0679021 11.8844652 18.0050336 10.7878127
##        428        429        430        431        432        433        434
## 20.6472553 13.6517519 11.0919374 17.5219597 15.4728206 23.1220995 18.9316610
##        435        436        437        438        440        441        442
## 19.9838110 11.8948527 17.1127240  8.7239466 12.2931088 13.0605642 15.6390982
##        443        444        445        446        447        448        449
## 18.5709121 16.3134037 11.3816481 11.1882848 17.3695039 18.7148411 17.0241586
##        450        452        453        455        456        457        459
## 15.8482966 17.4207989 17.8797479 16.4385791 17.0194216 16.1349703 18.9068676
##        460        461        462        463        464        465        466
## 20.4353412 18.7215574 20.4918048 21.1452250 24.8391651 21.9079433 21.0000354
##        467        468        469        470        471        473        474
## 17.9871933 13.8326041 17.0075789 20.3795460 18.8457432 20.7821076 23.4900724
##        475        477        479        480        481        483        484
## 17.0153529 16.4818130 17.1350726 22.0539538 24.3973118 28.1147085 24.7013602
##        485        486        487        488        489        490        491
## 21.7762203 24.5234005 20.1390407 23.6726246 17.1561445 11.2509748  5.5505796
##        492        493        494        495        496        498        499
## 17.1506547 21.8736436 23.2140831 21.6401596 17.6397263 21.1301947 22.3041620
##        500        501        502        503        504        505        506
## 20.1233762 20.8866867 25.5339845 26.1190141 29.5538058 28.7122753 27.3101192
```

```
#  fitTrain$fitted.values - SAME
```

Using regression for hypothesis testing

```
set.seed(1)
x1 <- rnorm(50,mean=20,sd=10)
x2 <- rnorm(50,mean=30,sd=10)
x3 <- rnorm(50,mean=21,sd=10)
```

```
x1x2 <- c(x1,x2)
id12 <- c(rep(1,length(x1)),rep(2,length(x2))) # rep() repeats a number as many times as the 2nd argume
id12 <- as.factor(id12)
id12
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## Levels: 1 2
```

```
summary(lm(x1x2~id12))
```

```
##
## Call:
## lm(formula = x1x2 ~ id12)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.1515  -6.0801   0.0502   5.7503  22.8429
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.004      1.277  16.453  < 2e-16 ***
## id122         10.169      1.805   5.632  1.7e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.027 on 98 degrees of freedom
## Multiple R-squared:  0.2445, Adjusted R-squared:  0.2368
## F-statistic: 31.72 on 1 and 98 DF,  p-value: 1.702e-07
```

```
mean(x1)
```

```
## [1] 21.00448
```

```
mean(x2)-mean(x1)
```

```
## [1] 10.16878
```

```
x1x3 <- c(x1,x3)
id13 <- c(rep(1,length(x1)),rep(3,length(x3)))
id13 <- as.factor(id13)
summary(lm(x1x3~id13))
```

```
##
## Call:
## lm(formula = x1x3 ~ id13)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.1515  -4.9561  -0.5849   5.5163  22.3965
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.004      1.225  17.146   <2e-16 ***
## id133         -1.529      1.732  -0.883     0.38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.662 on 98 degrees of freedom
## Multiple R-squared:  0.007889,   Adjusted R-squared:  -0.002235
## F-statistic: 0.7792 on 1 and 98 DF,  p-value: 0.3795
```

```
x1x2x3 <- c(x1,x2,x3)
id123 <- c(rep(1,length(x1)),rep(2,length(x1)),rep(3,length(x3)))
id123 <- as.factor(id123)
summary(lm(x1x2x3~id123))
```

```
##
## Call:
## lm(formula = x1x2x3 ~ id123)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.1515  -5.2574  -0.4894   5.3235  22.8429
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.004      1.275  16.471  < 2e-16 ***
## id1232        10.169      1.803   5.638 8.52e-08 ***
## id1233        -1.529      1.803  -0.848    0.398
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.017 on 147 degrees of freedom
## Multiple R-squared:  0.2528, Adjusted R-squared:  0.2426
## F-statistic: 24.86 on 2 and 147 DF,  p-value: 5.005e-10
```

```
x4 <- rnorm(10,mean=30,sd=10) # Only 10 observations
x1x4 <- c(x1,x4)
id14 <- c(rep(1,length(x1)),rep(2,length(x4))) # I can use any numbers I want, they do not mean anythin
id14 <- as.factor(id14)
summary(lm(x1x4~id14))
```

```
##
## Call:
## lm(formula = x1x4 ~ id14)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.1515  -5.3123  -0.4623   6.4409  21.2077
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   21.004      1.240  16.943   <2e-16 ***
## id142          6.481      3.037   2.134   0.0371 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.766 on 58 degrees of freedom
## Multiple R-squared:  0.07281,    Adjusted R-squared:  0.05682
## F-statistic: 4.555 on 1 and 58 DF,  p-value: 0.03707
```

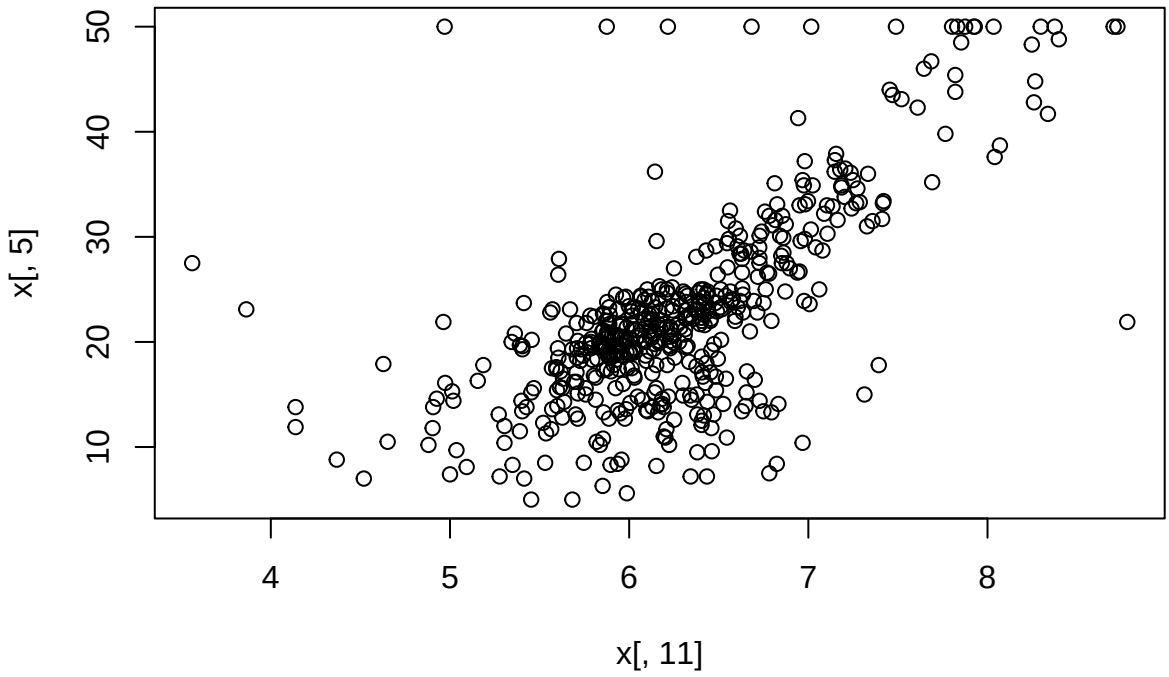
Excercises

For this question I want to focus on a regression model that predicts 'cmedv' with the help of 'rm' and 'lstat'. These two variables have the highest correlation with the target variable and therefore would be a good fit for forecasting.

1. Data exploration and test set creation

```
# model the correlation cmedv - rm

plot(x[,11], x[,5])
```

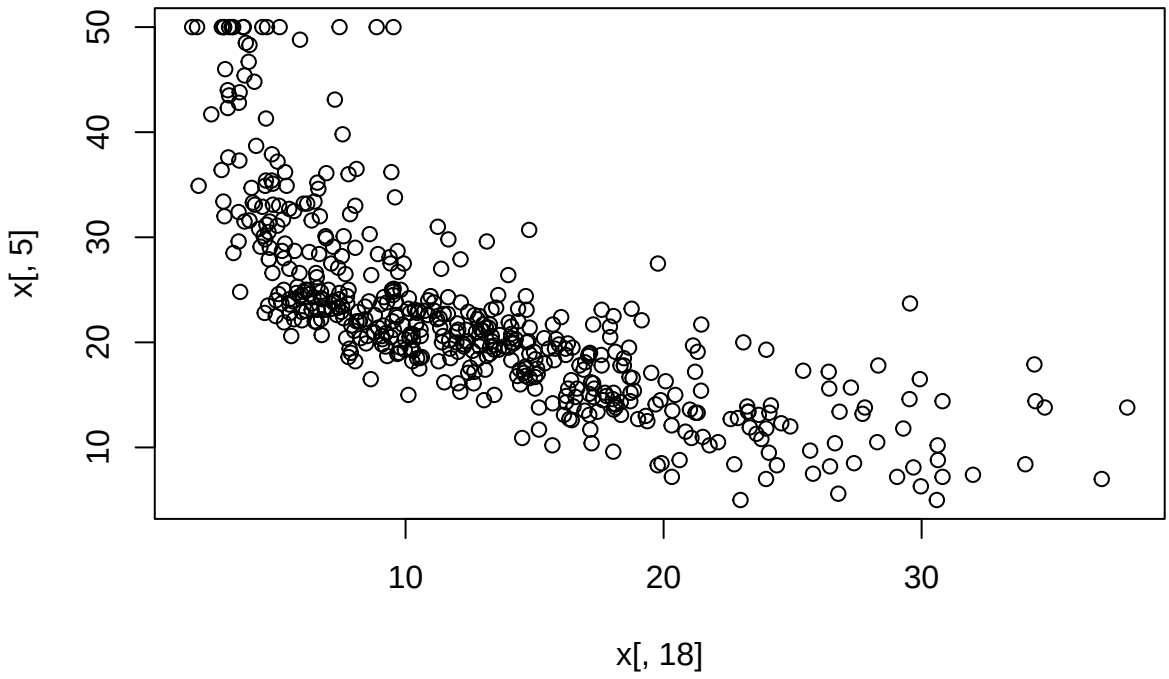



```
cor(x[,11], x[,5])
```

```
## [1] 0.6963038
```

```
# model the correlation cmedv - lstat

plot(x[,18], x[,5])
```




```
cor(x[,18], x[,5])
```

```
## [1] -0.740836
```

```r
# Creating a test set
set.seed(45)
idx_y<- sort(sample(1:nrow(x),50))
xTest_y <- x[idx_y,]
xTrain_y <- x[-idx_y,]
```

2. Model building

```r
fit1_y <- lm(cmedv ~ rm, data=xTrain_y)
sum1_y <- summary(fit1_y)
sum1_y
```

```
##
## Call:
## lm(formula = cmedv ~ rm, data = xTrain_y)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.060  -2.619   0.047   3.051  39.361
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -34.1318     2.8731  -11.88   <2e-16 ***
## rm            9.0082     0.4545   19.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.818 on 454 degrees of freedom
## Multiple R-squared:  0.4639, Adjusted R-squared:  0.4627
## F-statistic: 392.9 on 1 and 454 DF,  p-value: < 2.2e-16
```

```r
fit2_y <- lm(cmedv ~ lstat, data=xTrain_y)
sum2_y <- summary(fit2_y)
sum2_y
```

```
##
## Call:
## lm(formula = cmedv ~ lstat, data = xTrain_y)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.023  -4.072  -1.361   2.233  24.424
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.65620    0.59625   58.12   <2e-16 ***
## lstat       -0.95281    0.04057  -23.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.257 on 454 degrees of freedom
## Multiple R-squared:  0.5485, Adjusted R-squared:  0.5475
## F-statistic: 551.5 on 1 and 454 DF,  p-value: < 2.2e-16
```

```r
fit3_y <- lm(cmedv ~ rm + lstat, data=xTrain_y)
sum3_y <- summary(fit3_y)
sum3_y
```

```
##
## Call:
## lm(formula = cmedv ~ rm + lstat, data = xTrain_y)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.655  -3.578  -1.071   1.977  27.535
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.57832    3.38296   0.171    0.864
## rm           4.84073    0.47442  10.203   <2e-16 ***
## lstat       -0.66633    0.04615 -14.439   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.649 on 453 degrees of freedom
## Multiple R-squared:  0.6329, Adjusted R-squared:  0.6312
## F-statistic: 390.4 on 2 and 453 DF,  p-value: < 2.2e-16
```

```r
fit4_y <- lm(cmedv ~ rm + lstat + nox + ptratio + chas + b + age, data=xTrain_y)
sum4_y <- summary(fit4_y)
sum4_y
```

```
##
## Call:
## lm(formula = cmedv ~ rm + lstat + nox + ptratio + chas + b +
##     age, data = xTrain_y)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.6146  -3.0181  -0.8873   1.6939  29.0511
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.566370   4.782714   3.464 0.000584 ***
## rm           4.347981   0.455276   9.550  < 2e-16 ***
## lstat       -0.558828   0.056048  -9.971  < 2e-16 ***
## nox         -5.956139   3.370929  -1.767 0.077924 .
## ptratio     -0.883592   0.125151  -7.060 6.37e-12 ***
## chas1        3.329805   0.994503   3.348 0.000882 ***
## b            0.008861   0.002901   3.055 0.002386 **
## age          0.028742   0.013663   2.104 0.035962 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.203 on 448 degrees of freedom
## Multiple R-squared:  0.692,  Adjusted R-squared:  0.6872
## F-statistic: 143.8 on 7 and 448 DF,  p-value: < 2.2e-16
```

3. Model evaluation

```
# Get R^2 and AIC
fit1_yr2 <- sum1_y$r.squared
fit1_yaic <- AIC(fit1_y)

fit2_yr2 <- sum2_y$r.squared
fit2_yaic <- AIC(fit2_y)

fit3_yr2 <- sum3_y$r.squared
fit3_yaic <- AIC(fit3_y)

fit4_yr2 <- sum4_y$r.squared
fit4_yaic <- AIC(fit4_y)
```

```
r2_y <- c(fit1_yr2,fit2_yr2, fit3_yr2, fit4_yr2)
r2_y
```

```
## [1] 0.4638976 0.5484862 0.6328622 0.6919850
```

```
names(r2_y) <- c("rm","lstat","rm & lstat", "multiple")
round(r2_y,3)
```

```
##         rm      lstat rm & lstat   multiple
##      0.464      0.548      0.633      0.692
```

Based on the r2 - Score we can see that the multiple regression model performs the best. Better than the singular models and the model based on a combination of the two most influencing features. For the singular models the r2-score hints that the lstat-model performs better than the rm-model.

```
aic_y <- c(fit1_yaic, fit2_yaic, fit3_yaic, fit4_yaic)
names(aic_y) <- names(r2_y)
round(aic_y,3)
```

```
##         rm      lstat rm & lstat   multiple
##   3048.775   2970.471   2878.139   2808.070
```

The AIC score has the same outcomes and agrees that the multiple model performs best.

4. Prediction using the different models

```
pred1 <- predict(fit1_y, newdata=xTest_y)
rmse1 <- rmse(xTest_y$cmedv, pred1)
```

```
pred2 <- predict(fit2_y, newdata=xTest_y)
rmse2 <- rmse(xTest_y$cmedv, pred2)
```

```
pred3 <- predict(fit3_y, newdata=xTest_y)
rmse3 <- rmse(xTest_y$cmedv, pred3)
```

```
pred4 <- predict(fit4_y, newdata=xTest_y)
rmse4 <- rmse(xTest_y$cmedv, pred4)
```

```
rmse_y <- c(rmse1, rmse2, rmse3, rmse4)
names(rmse_y) <- names(r2_y)
round(rmse_y,3)
```

```
##         rm      lstat rm & lstat   multiple
```

```
##      4.075      5.358      3.943      3.402
```

The RMSE score shows that the multiple regression model performed best in the predictions but it is also visible that the singular feature rm model functions better in the prediction than singular feature lstat model.

5. Resampling of the data

```
# Creating a test set
set.seed(200)
idx_y<- sort(sample(1:nrow(x),50))
xTest_y2 <- x[idx_y,]
xTrain_y2 <- x[-idx_y,]
```

```
fit1_y2 <- lm(cmedv ~ rm, data=xTrain_y2)
```

```
fit2_y2 <- lm(cmedv ~ lstat, data=xTrain_y2)
```

```
fit3_y2 <- lm(cmedv ~ rm + lstat, data=xTrain_y2)
```

```
fit4_y2 <- lm(cmedv ~ rm + lstat + nox + ptratio + chas + b + age, data=xTrain_y2)
```

```
# Get R^2 and AIC
sum1_y2 <- summary(fit1_y2)
fit1_y2r2 <- sum1_y2$r.squared
fit1_y2aic <- AIC(fit1_y2)

sum2_y2 <- summary(fit2_y2)
fit2_y2r2 <- sum2_y2$r.squared
fit2_y2aic <- AIC(fit2_y2)

sum3_y2 <- summary(fit3_y2)
fit3_y2r2 <- sum3_y2$r.squared
fit3_y2aic <- AIC(fit3_y2)

sum4_y2 <- summary(fit4_y2)
fit4_y2r2 <- sum4_y2$r.squared
fit4_y2aic <- AIC(fit4_y2)
```

```
r2_y2 <- c(fit1_y2r2,fit2_y2r2, fit3_y2r2, fit4_y2r2)
r2_y2
```

```
## [1] 0.4662829 0.5454583 0.6374662 0.6989702
```

```
names(r2_y2) <- c("rm","lstat","rm & lstat", "multiple")
round(r2_y2,3)
```

```
##         rm      lstat rm & lstat   multiple
##      0.466      0.545      0.637      0.699
```

The r2 scores are very similar to the previous ones where another training test split was used.

```
aic_y2 <- c(fit1_y2aic, fit2_y2aic, fit3_y2aic, fit4_y2aic)
names(aic_y2) <- names(r2_y2)
round(aic_y2,3)
```

```
##         rm      lstat rm & lstat   multiple
##   3025.885   2952.663   2851.528   2776.754
```

The AIC score also remains similar to the previous exercise although all of the models perform better with the second training set.

```
pred12 <- predict(fit1_y2, newdata=xTest_y2)
rmse12 <- rmse(xTest_y2$cmedv, pred12)
```

```
pred22 <- predict(fit2_y2, newdata=xTest_y2)
rmse22 <- rmse(xTest_y2$cmedv, pred22)
```

```
pred32 <- predict(fit3_y2, newdata=xTest_y2)
rmse32 <- rmse(xTest_y2$cmedv, pred32)
```

```
pred42 <- predict(fit4_y2, newdata=xTest_y2)
rmse42 <- rmse(xTest_y2$cmedv, pred42)
```

```
rmse_y2 <- c(rmse12, rmse22, rmse32, rmse42)
names(rmse_y2) <- names(r2_y2)
round(rmse_y2,3)
```

```
##         rm      lstat rm & lstat   multiple
##      6.109      6.506      5.643      5.339
```

The predictions have the same outcome as with the other training set where the multiple regression model performs best in the prediction. There is to mention that the overall prediction is way worse than in the previous train-test-split.