Regression modelling in R

Rizny Mubarak

Contents

1. Lab Experiment

Load dataset

```
if (!("mlbench" %in% rownames(installed.packages()))){
  install.packages("mlbench")
}
library(mlbench)
```

```
data(BostonHousing2)
x <- BostonHousing2
x <- x[,-5]
```

- Check first 5 rows of data

```
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
```

- Convert to Dataframe

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

- Check the class of all columns

```
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"
```

- Get all the numeric columns indexs

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

Data exploration

- Find Correlations

```
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
```

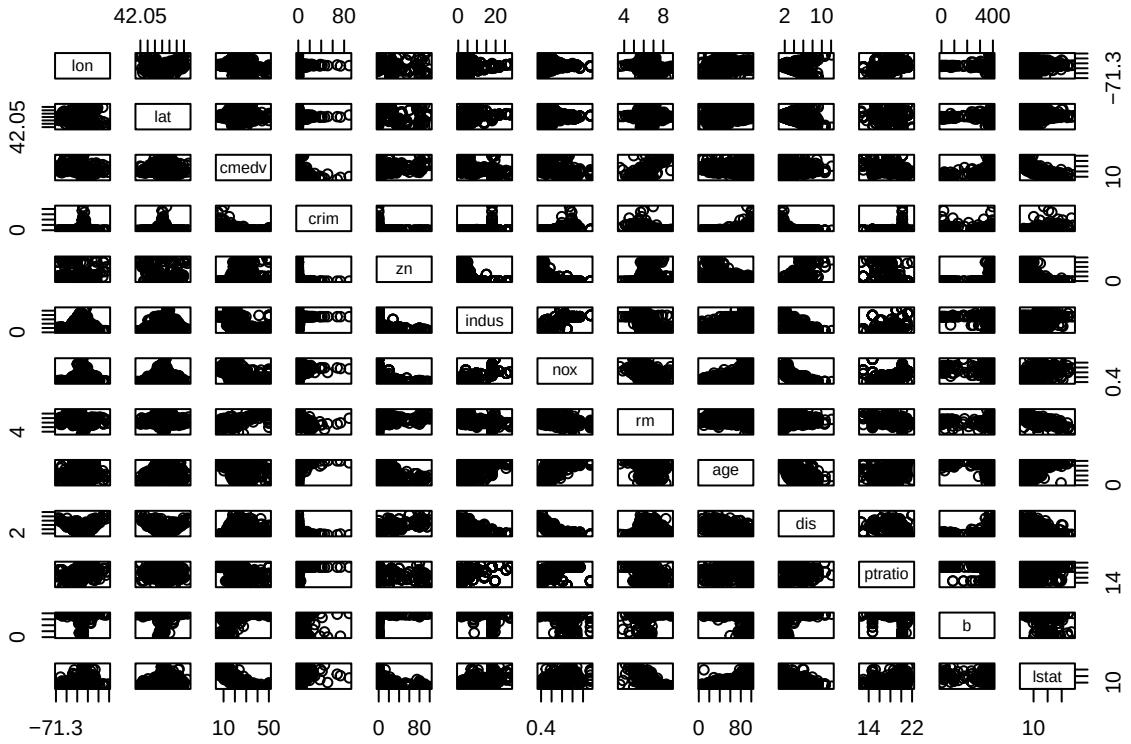
```
## rm      -0.25711003 -0.069316987  0.696303794 -0.21924670  0.3119906
## age      0.20473895  0.079035217 -0.377998896  0.35273425 -0.5695373
## dis     -0.01124313 -0.082980855  0.249314834 -0.37967009  0.6644082
## ptratio  0.31260219 -0.004527081 -0.505654619  0.28994558 -0.3916785
## b       -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]
b <- x[,4]
cor(a,b)
```

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

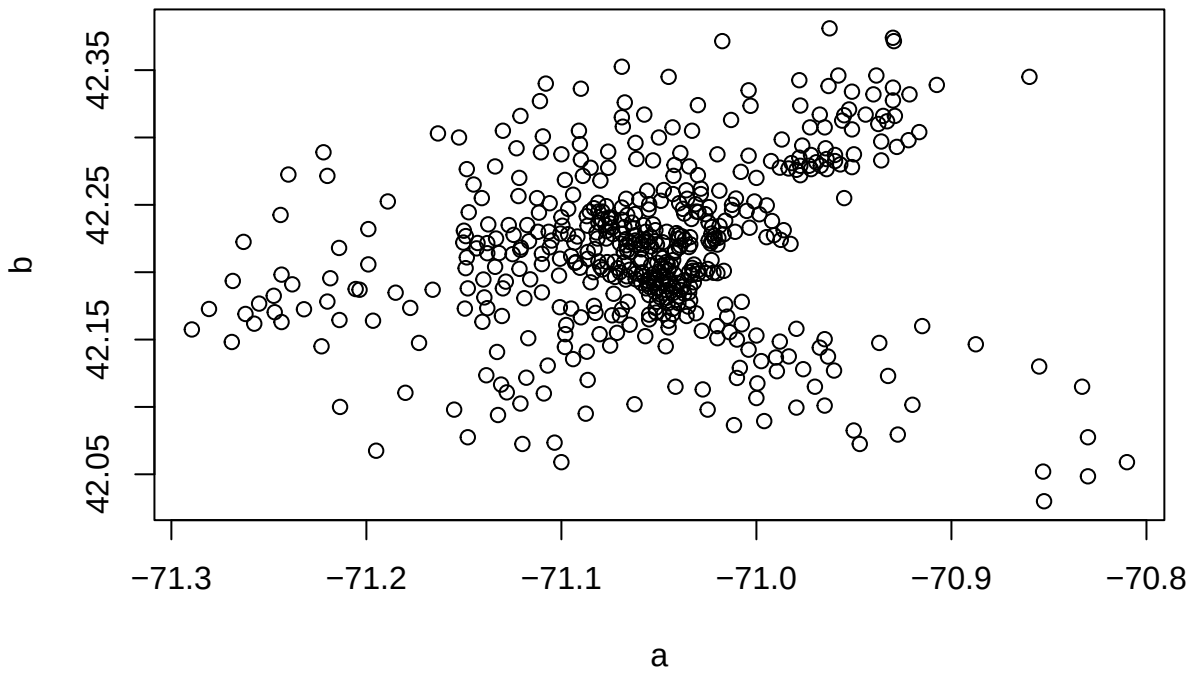
- Plot correlations

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



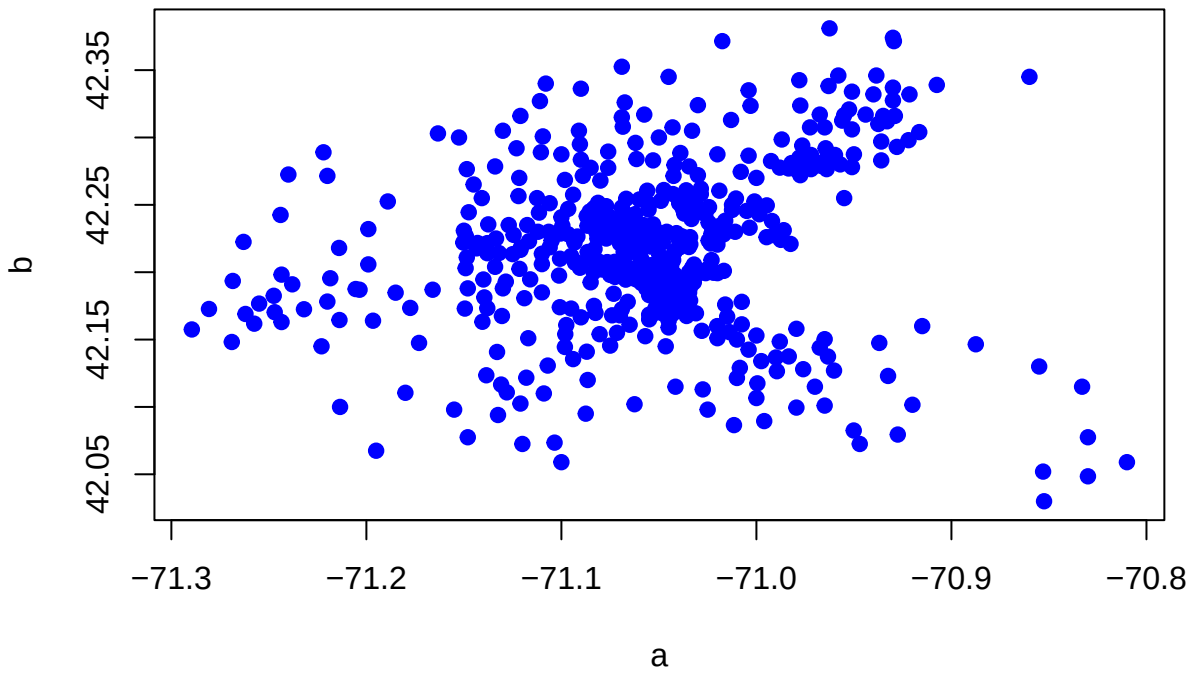
- Scatter plot with two variables

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



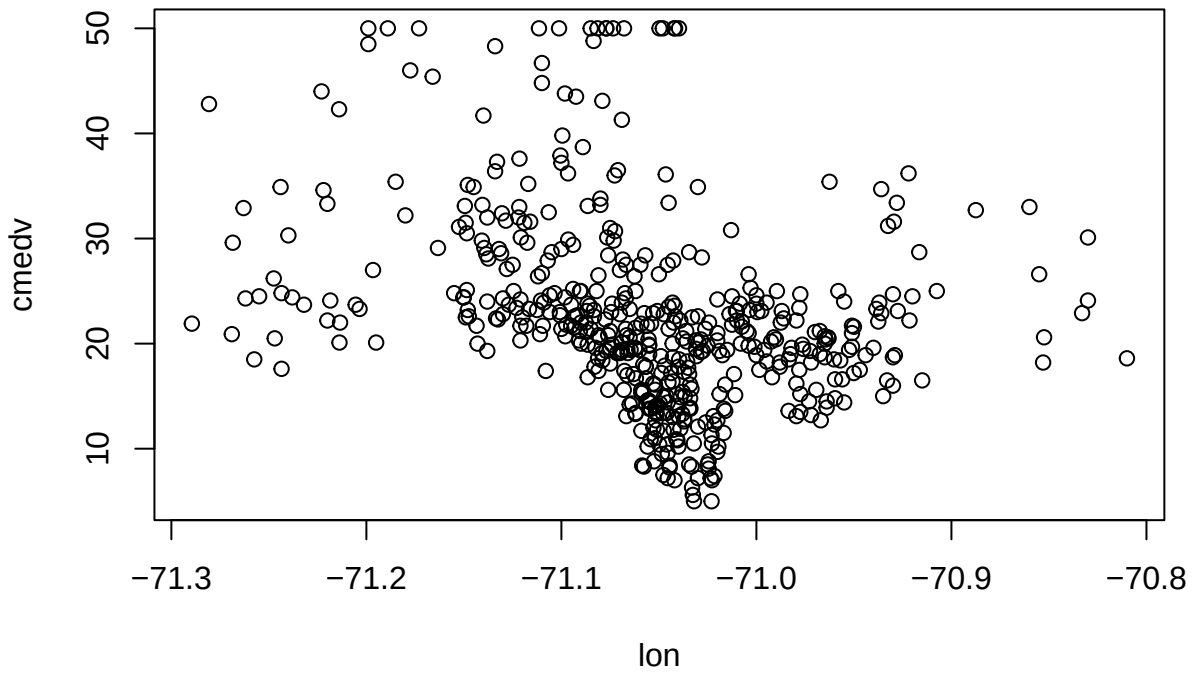
- Scatter plot with colors

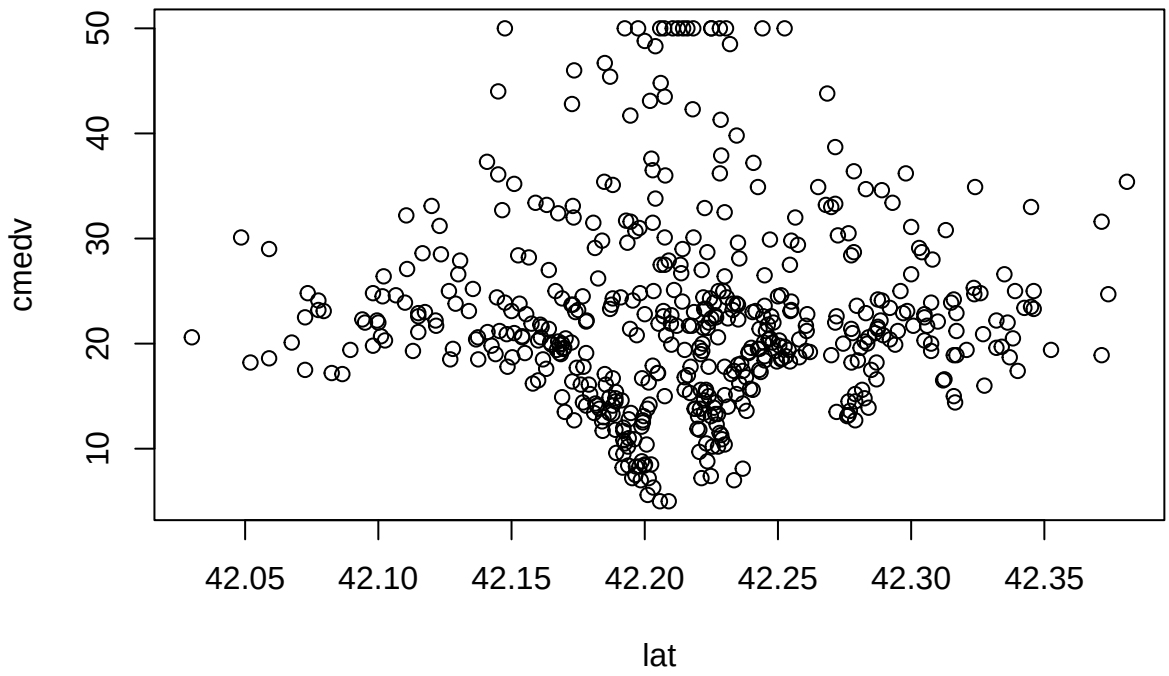
```
plot(a,b,pch=19,col="blue")
```

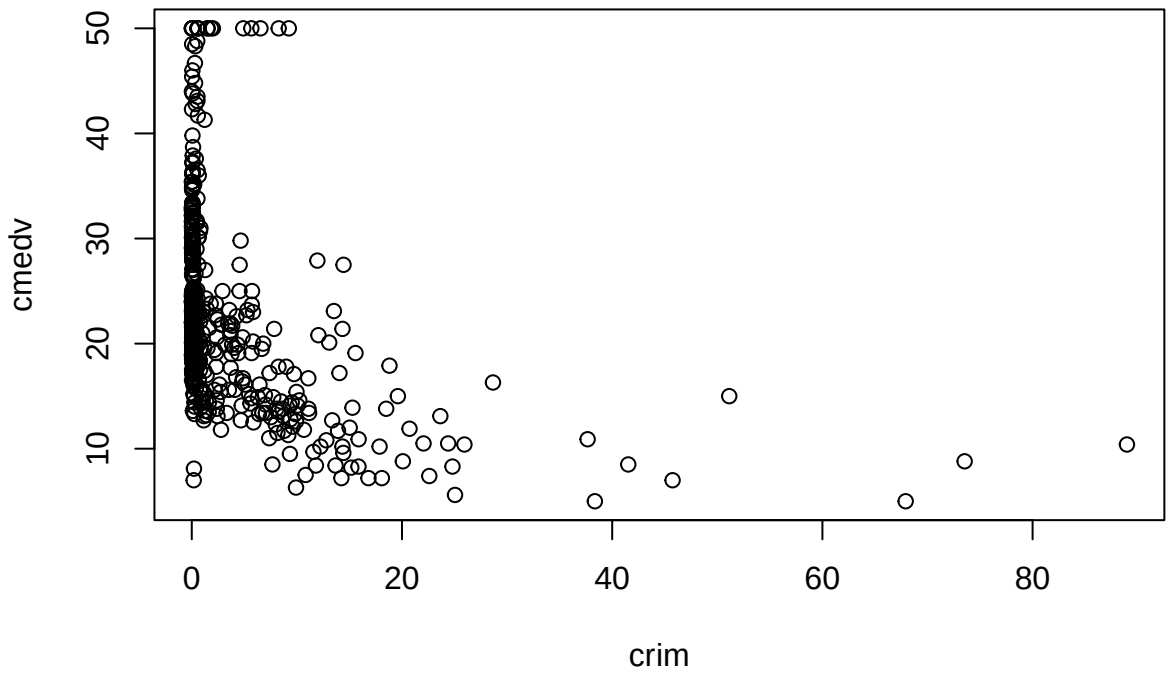


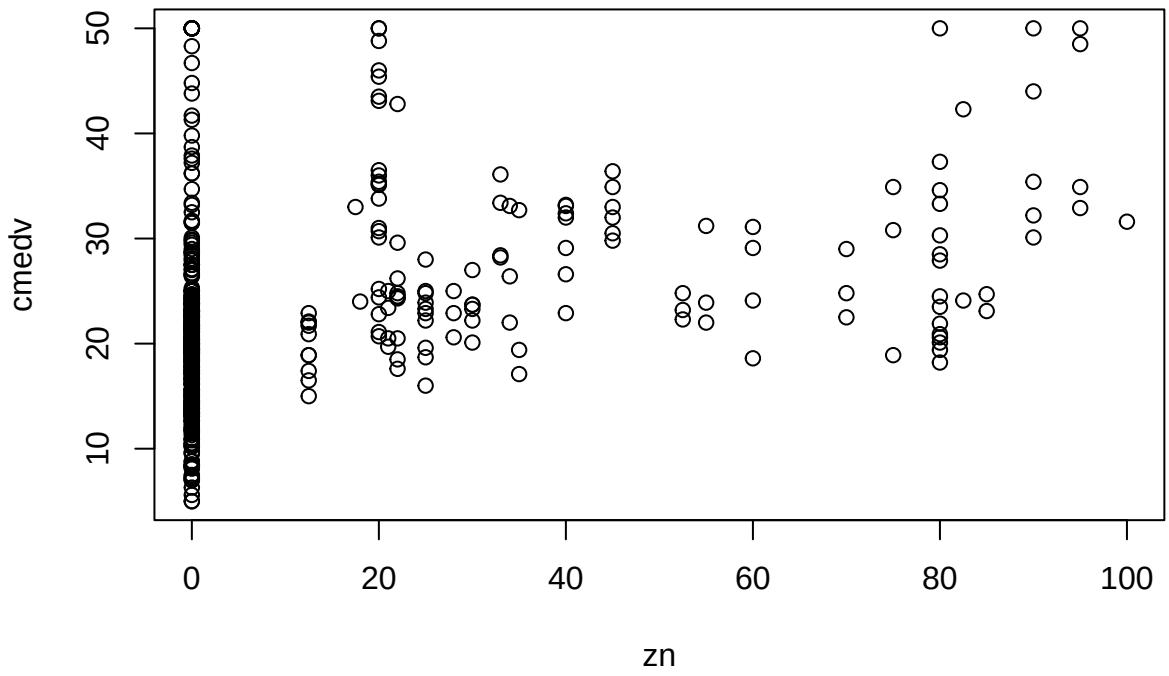
- Plot scatter plot targets vs input variables

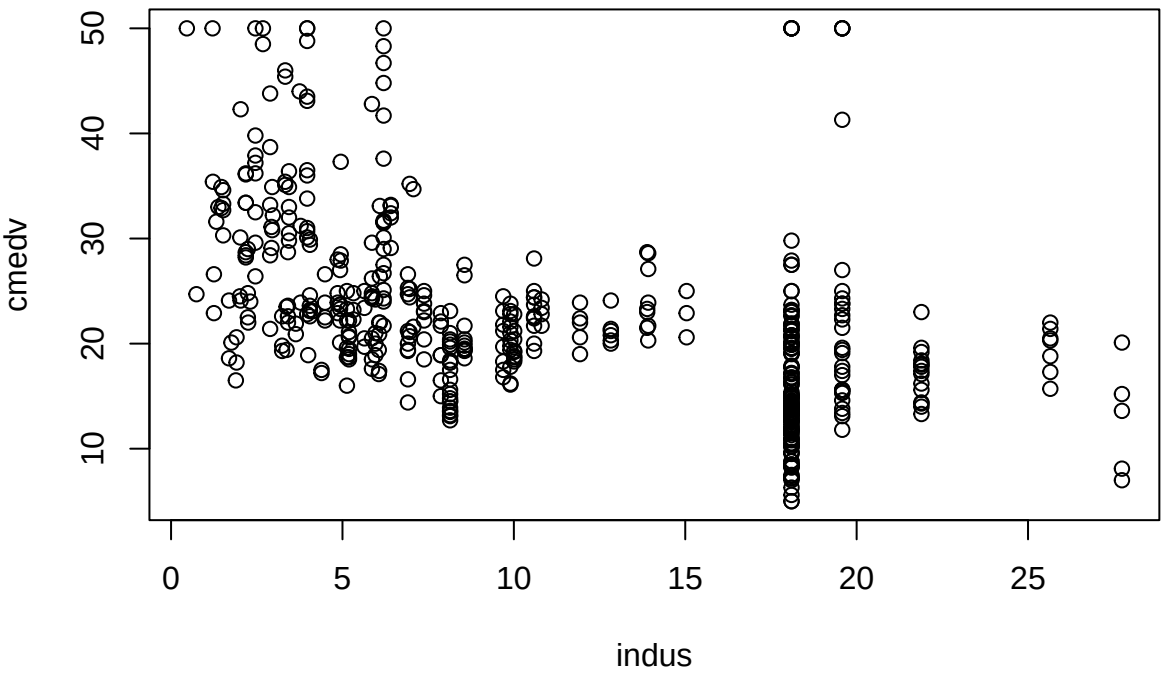
```
tempY <- x[,5] # Starget cmedv
tempX <- (x[,idxNum])[,-3] # numerical inputs

# 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")
}
```

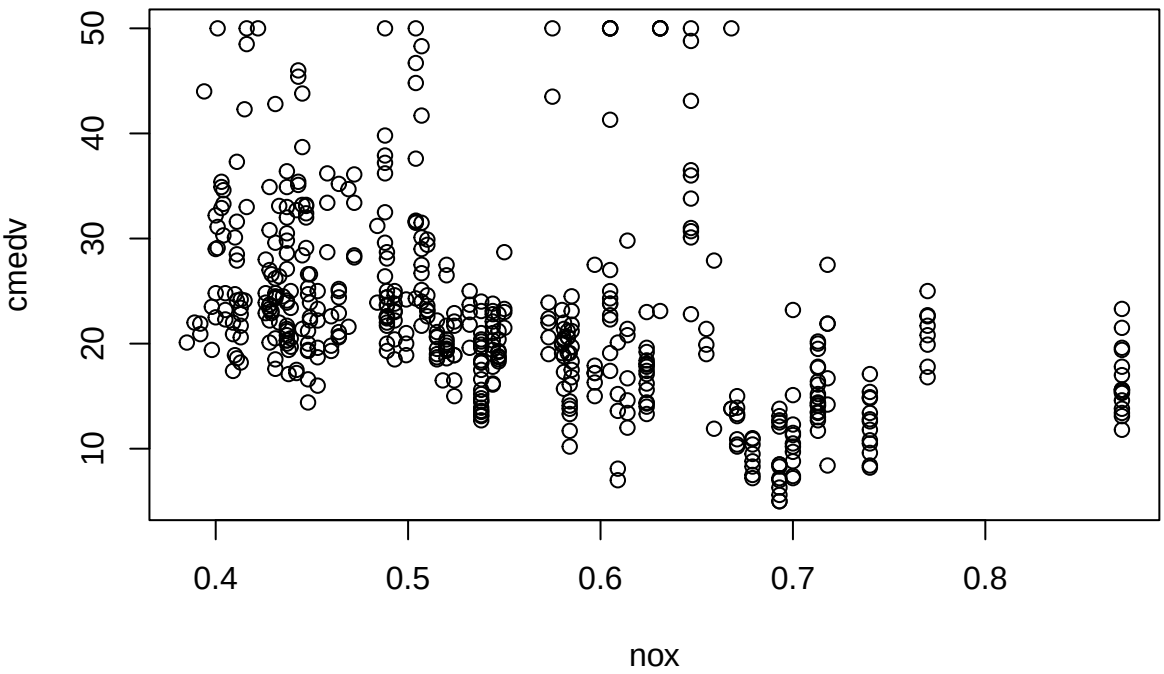

cmedv
10
20
30
40
50
−71.3
−71.2
−71.1
−71.0
−70.9
−70.8
lon

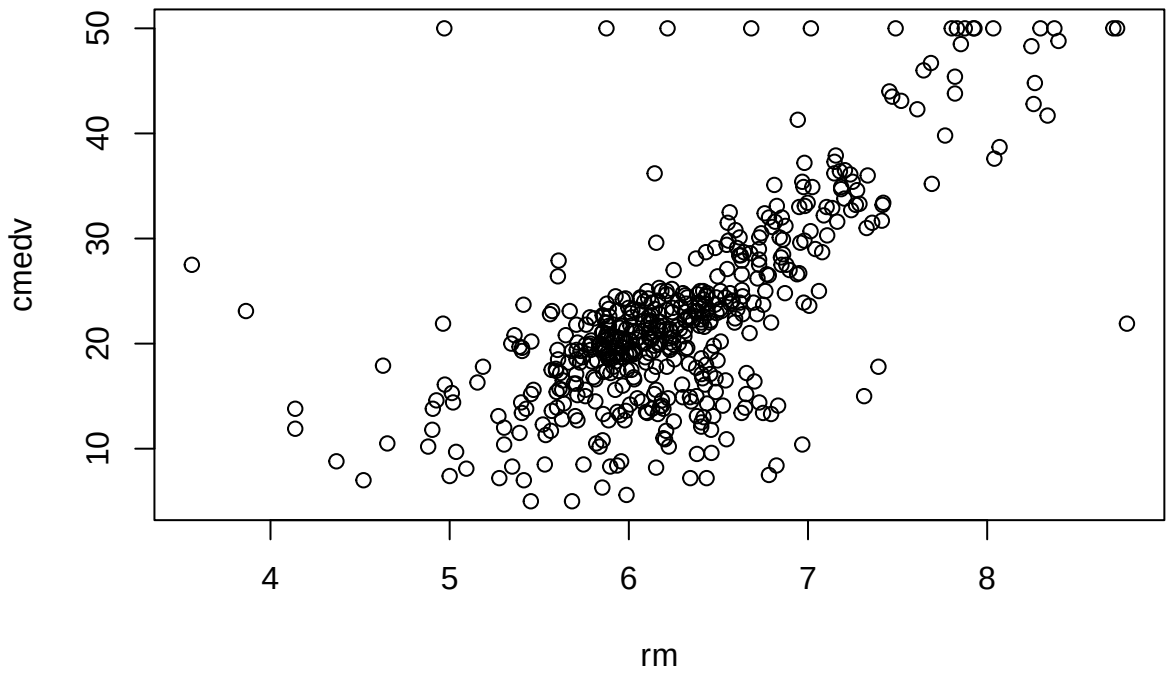

cmedv
10
20
30
40
50
42.05
42.10
42.15
42.20
42.25
42.30
42.35
lat

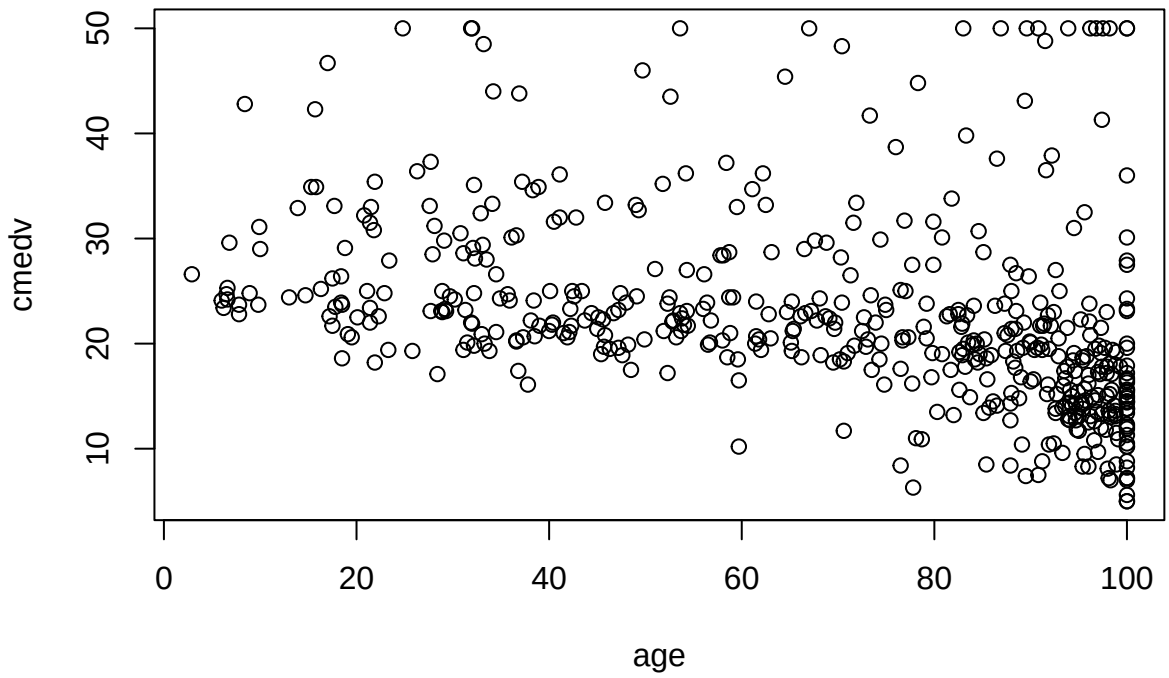

cmedv
10
20
30
40
50
0
20
40
60
80
crim

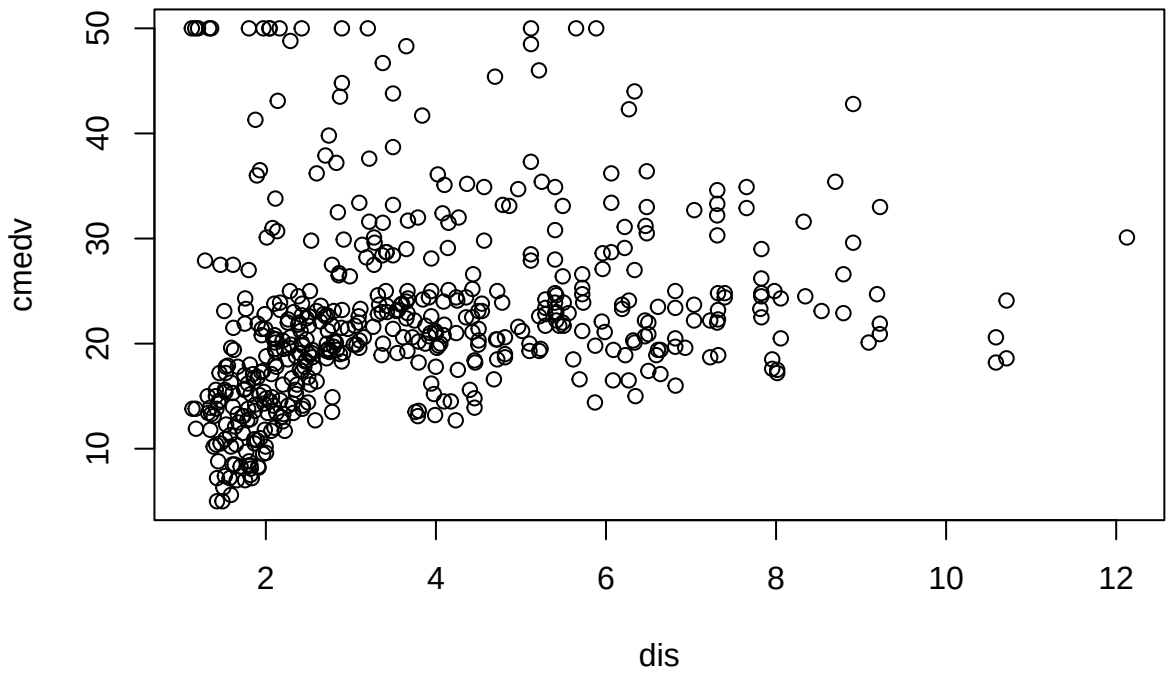

cmedv
10
20
30
40
50
0
20
40
60
80
100
zn

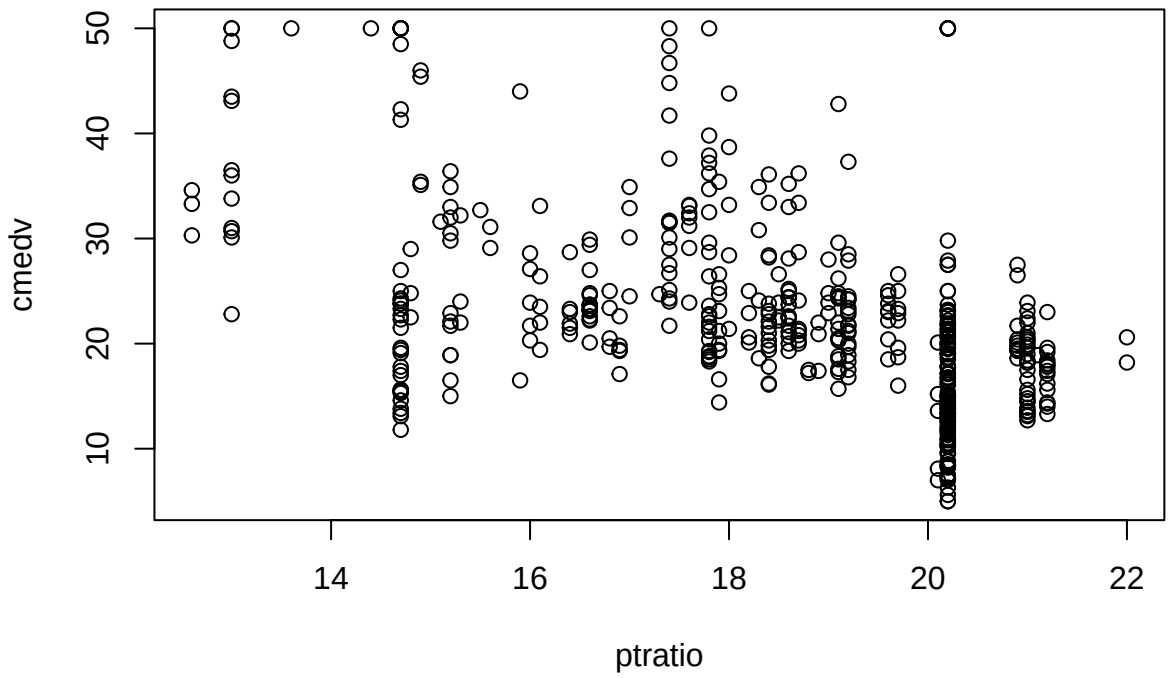

cmedv
indus
10
20
30
40
50
0
5
10
15
20
25

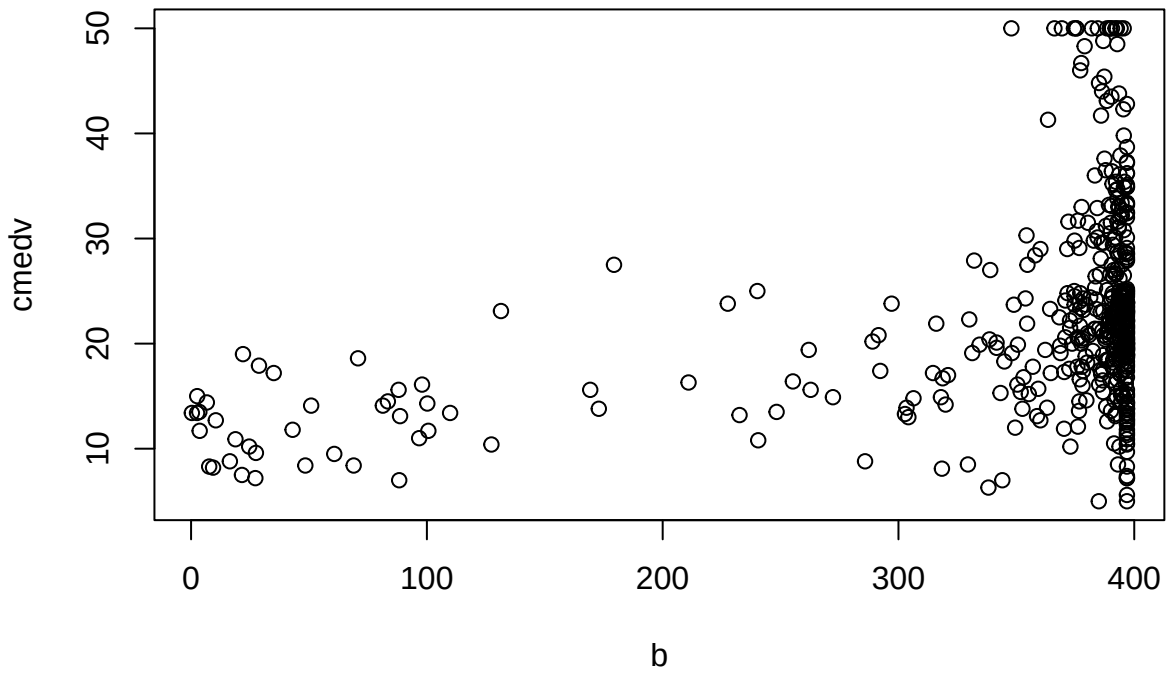

cmedv
10
20
30
40
50
0.4
0.5
0.6
0.7
0.8
nox

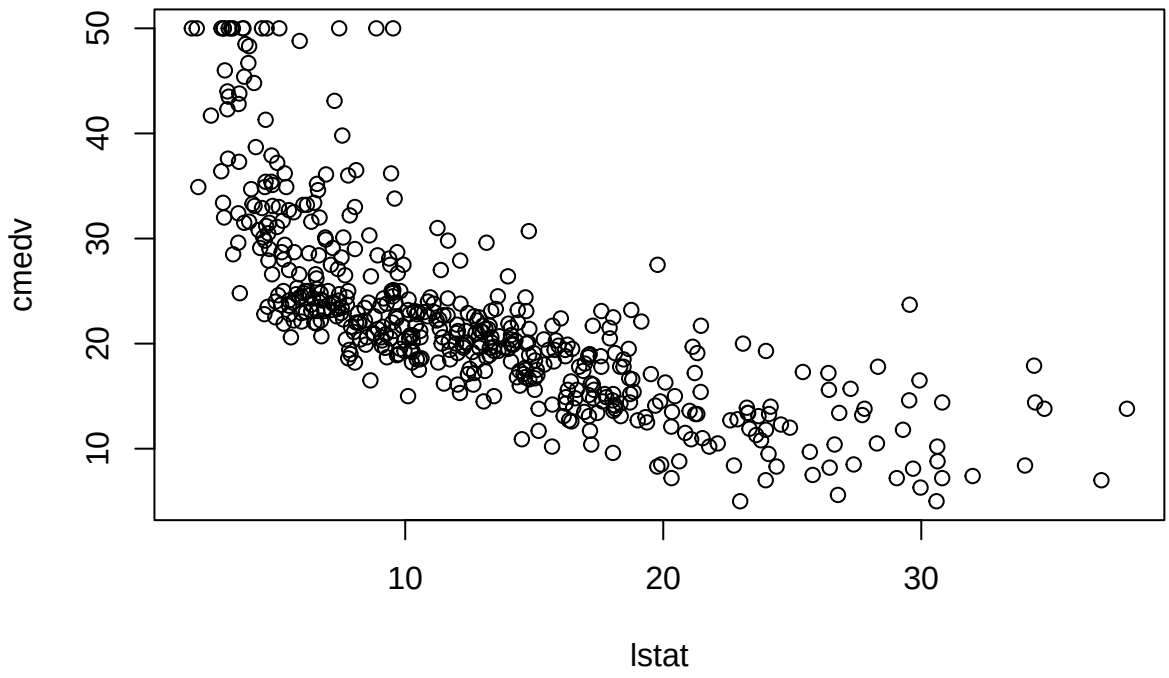

cmedv
10
20
30
40
50
4
5
6
7
8
rm


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


cmedv
10
20
30
40
50
0
100
200
300
400
b



- Observation
  - some nice linear relations;
  - lots of noise;
  - some potentially problematic variables (non-linear relations).

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 of fit1

```
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
```

- Store summary of fit1 to sum1

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

```
names(sum1)
```

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

For future reference: within a list you can save any type of variables. These could be vectors, arrays, lists, data.frames, etc. For example, we extract the p-values and test them at 5% significance.

- Fit1 coeffients

```
# print coeffients
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
```

- Evaluate P-Value

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

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

- Store fit1 R^2

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

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

- Store AIC

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

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

- Sales on Revenue

```
# 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
```

- Fit2 Coefficients

```
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
```

- R^2 Fit1 vs Fit2

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

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

- R^2 value rounding with different decimal places

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

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

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

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

- AIC Comparison

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

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

- Sales on Town

```
# Regress sales on town
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
```

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

- Sales on Price with and without Intercept

```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
```

Observe that the second model, which uses $0 + \ldots$ in the formula, removes the intercept.

Multiple regression

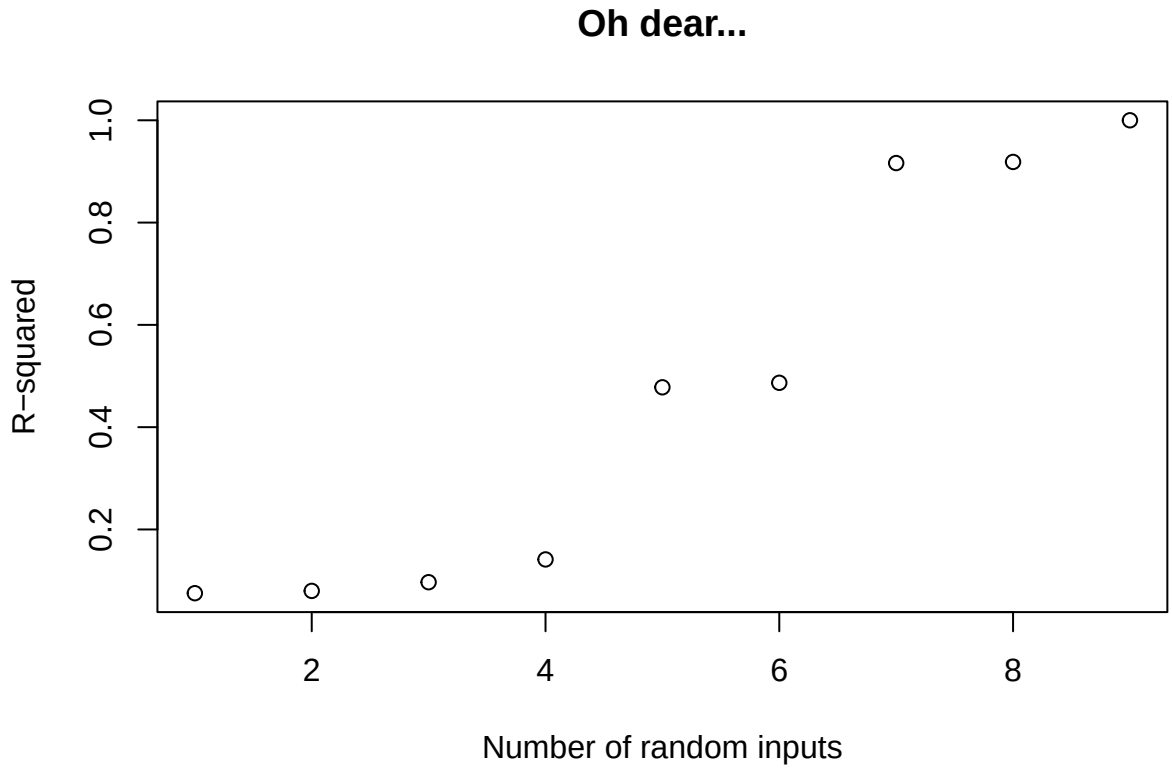
- Sales on Price + Revenue

```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
```

- Sales vs Random inputs - Demonstrate R^2 variations

```
# 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
# The function runif() creates random draws from a uniform
# distirbution
xx <- matrix(runif(90),ncol=9) # Draw 100 values and put them
                               # in a matrix with 10 columns
# Loop for all regressions from 1 to 9 inputs
ftemp <- list() # Pre-allocate a list to save the results
for (i in 1:9){
  ftemp[[i]] <- lm(yy ~xx[,1:i])
}
# Get R-squared from all models
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...")
```



- AIC vlues

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

```
## [1] 71.21855 73.17003 74.98343 76.47833 73.50224 75.33250 59.18422 60.91937
## [9]     -Inf
```

- Error fitted values

```
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
```

- Model with just a constant

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

```
##  [1] 71.21855 73.17003 74.98343 76.47833 73.50224 75.33250 59.18422 60.91937
##  [9]     -Inf 70.00286
```

- Check R^2

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

```
##  [1] 0.07543409 0.07990893 0.09691848 0.14140089 0.47798636 0.48677241
##  [7] 0.91641268 0.91859746 1.00000000 0.00000000
```

Variable selection

- cmedv vs all variables

```
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
```

- cmedv with only intercept - fitmin

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

- Bidirectional model combining fitmin and fit5(all)

```r
fit6 <- step(fitmin,direction="both",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 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
```

- Fit6 summary

```
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
```

- Forward directional model combining fitmin and fit5(all)

```
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 of Forward directional model fit7

```
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
```

- Backward directional model combining fitmin and fit5(all)

```
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 backward directional model

```
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
```

- Compare fit1, fit2, fit5, fit6, fit7 and fit8

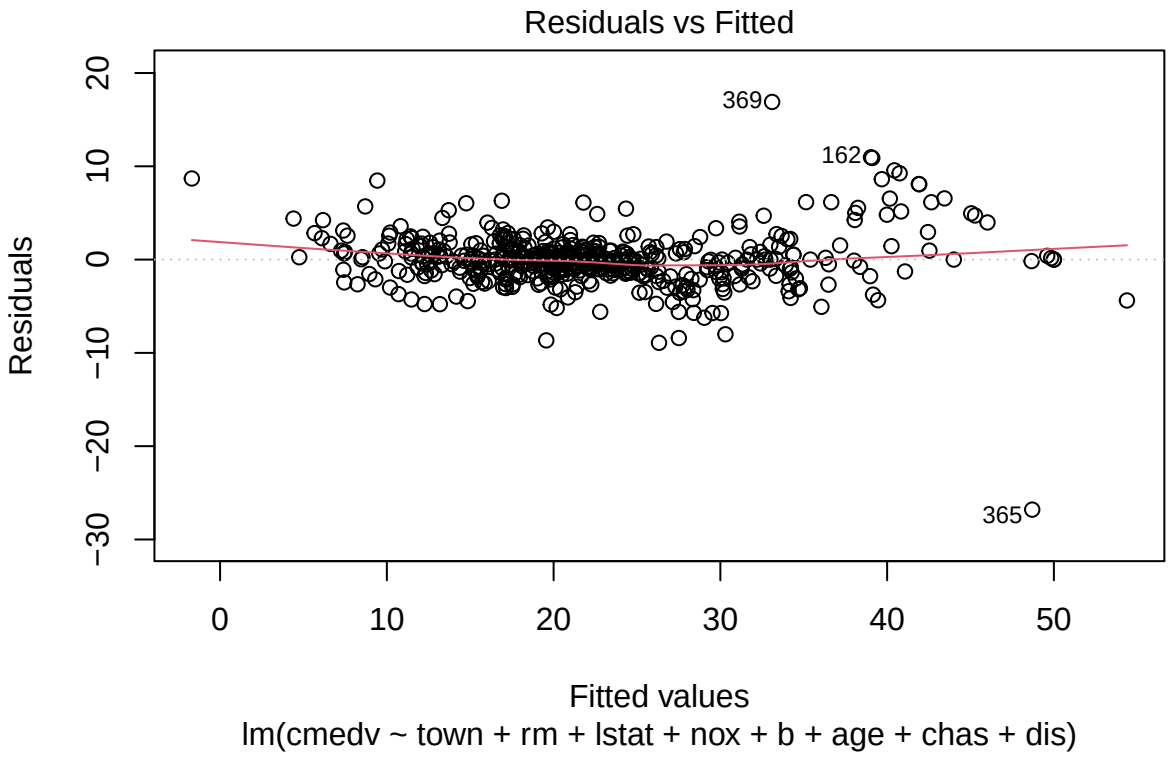
```
aic <- c(AIC(fit1),AIC(fit2),AIC(fit5),AIC(fit6),AIC(fit7),AIC(fit8))
names(aic) <- c(formula(fit1),formula(fit2),"Full model","Stepwise","Forward","Backward")
round(aic,4)
```

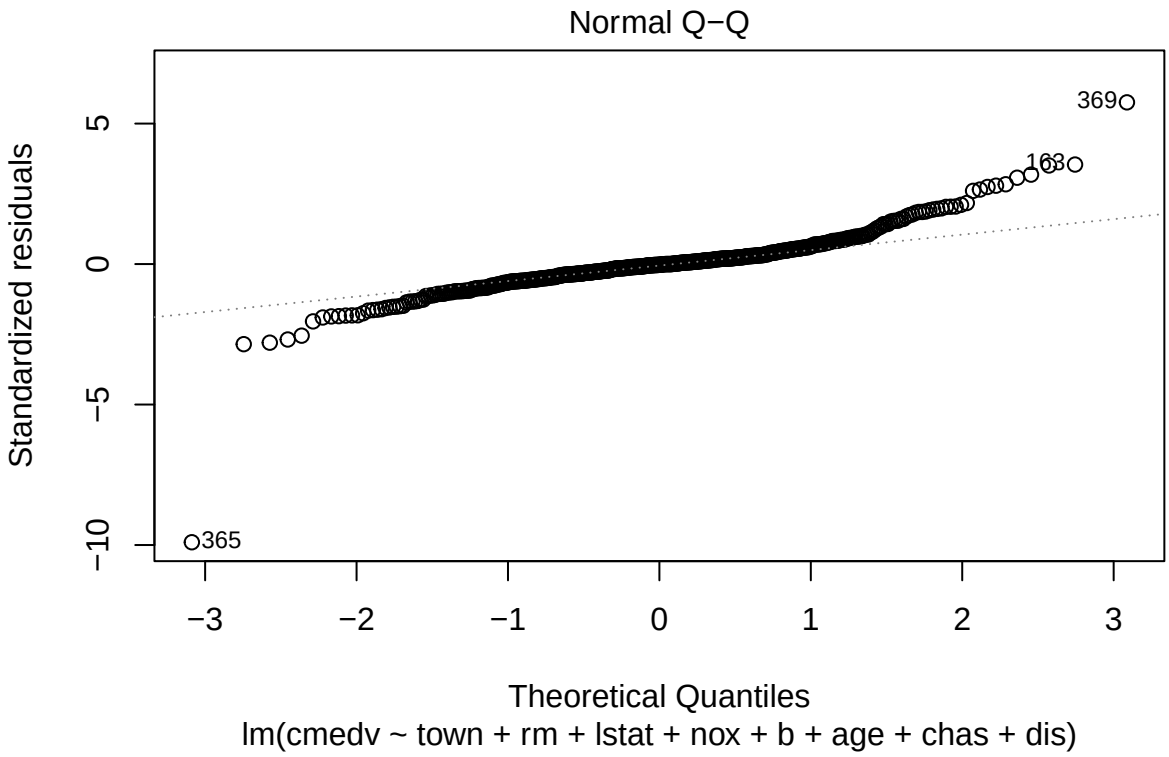
```
## cmedv ~ lstat   cmedv ~ lat    Full model      Stepwise       Forward
##      3282.096      3684.813      2726.360      2721.155      2721.155
##      Backward
##      2721.155
```

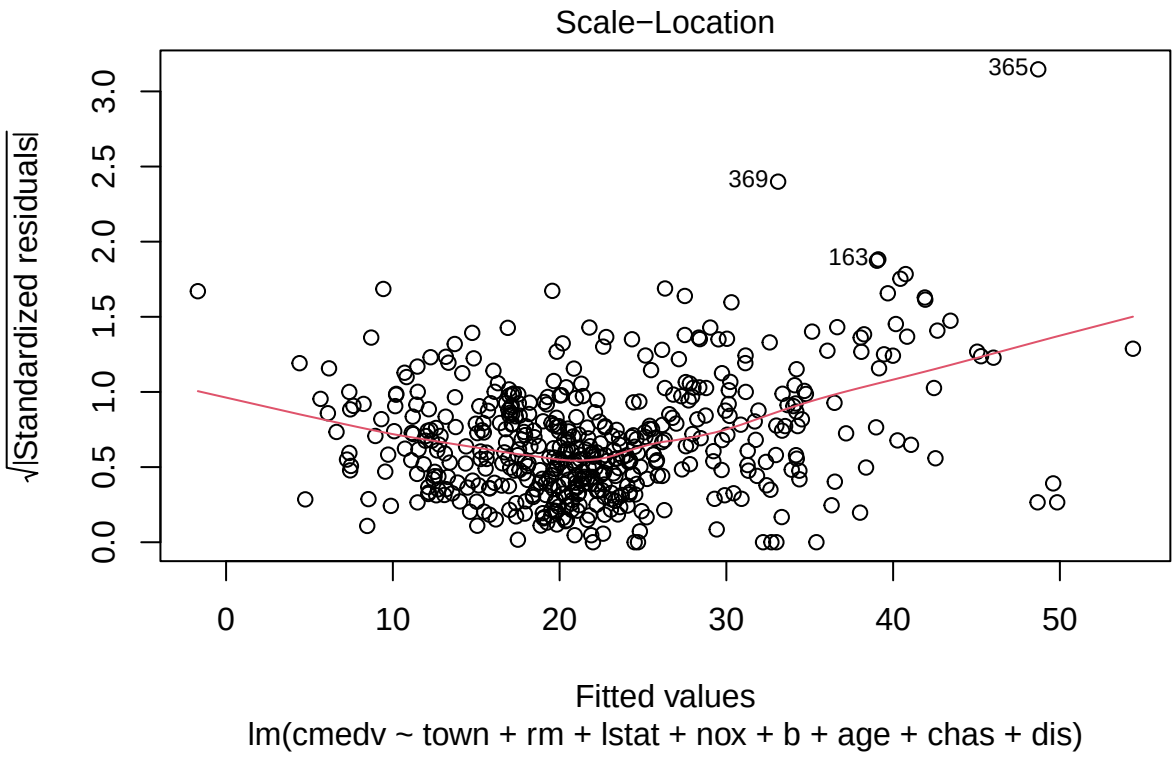
- Summary of Fit6

```
plot(fit6)
```

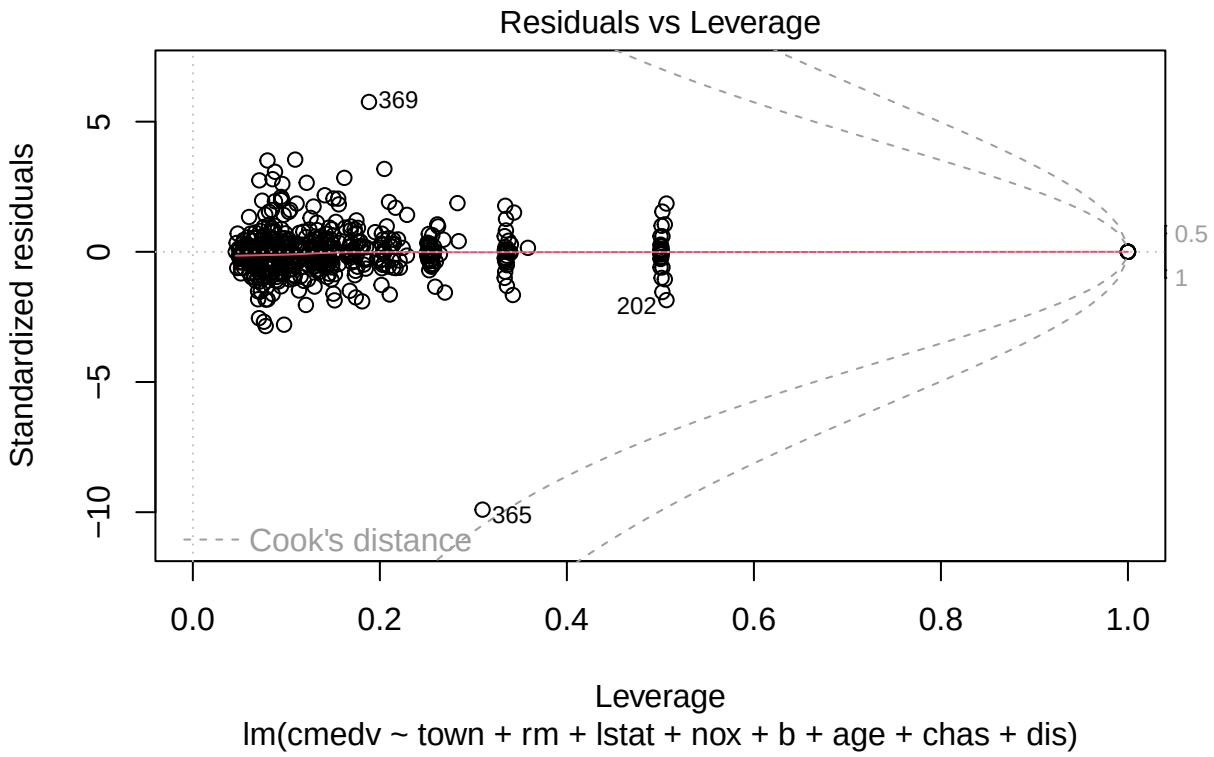
```
## Warning: not plotting observations with leverage one:
##   1, 55, 58, 196, 257, 284, 287, 343, 348, 354
```


Residuals vs Fitted
Residuals
Fitted values
lm(cmedv ~ town + rm + lstat + nox + b + age + chas + dis)
369
162
365
20
10
0
−10
−20
−30
0
10
20
30
40
50


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(cmedv ~ town + rm + lstat + nox + b + age + chas + dis)
369
365
5
0
−5
−10
−3
−2
−1
0
1
2
3




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

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



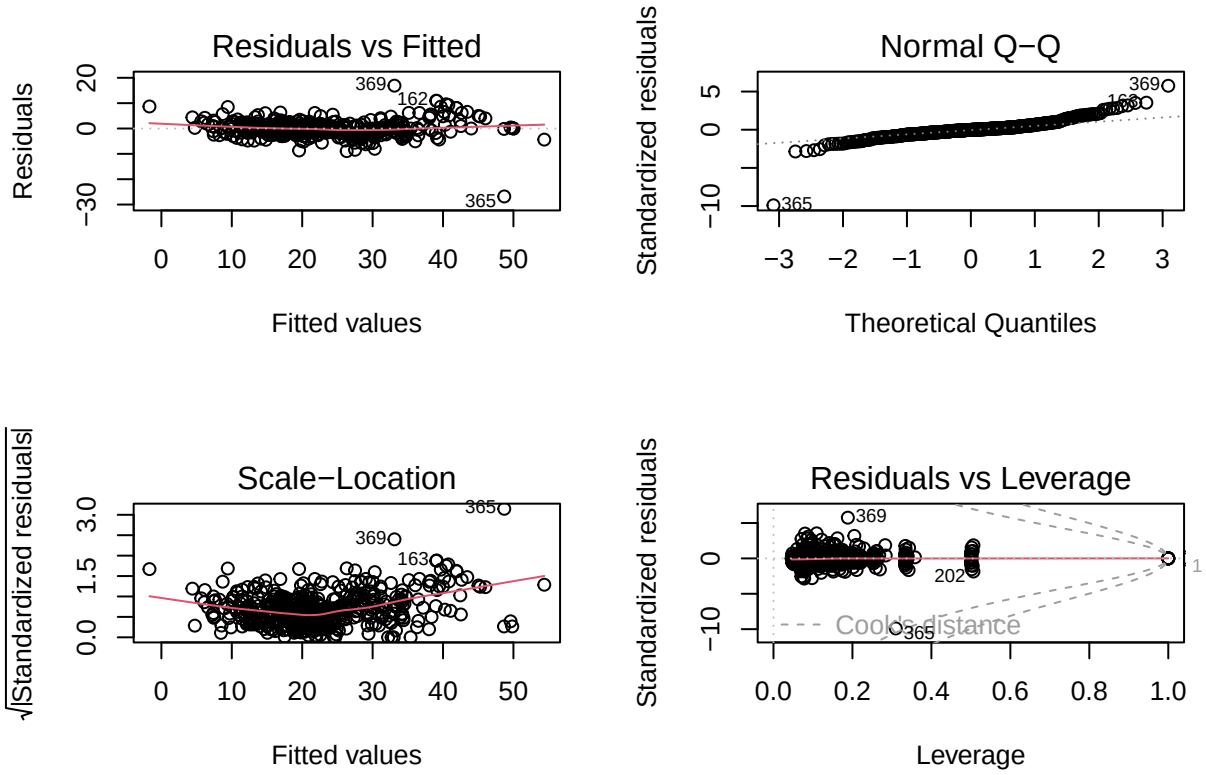
- Fit all the plots for fit6

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

```
## Warning: not plotting observations with leverage one:
##   1, 55, 58, 196, 257, 284, 287, 343, 348, 354
```

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

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

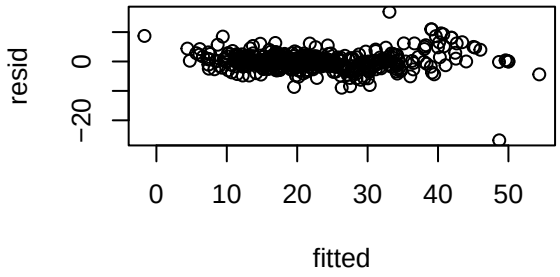


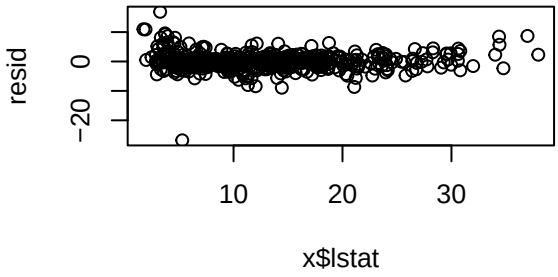

```
par(mfrow=c(1,1))
```

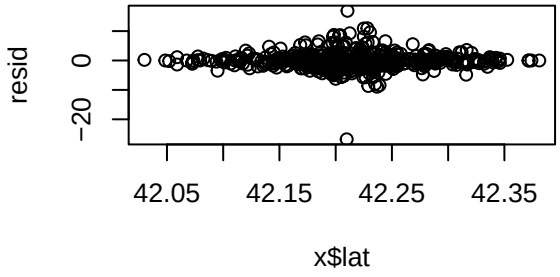
```
resid <- fit6$residuals
fitted <- fit6$fitted.values
```

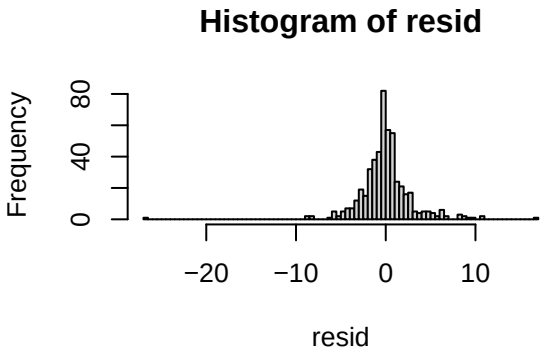
And produce the plots

```
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

- Split data train and test

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

- Fit the model - cmedv ~ lstat + lat

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

```
##
## Call:
## lm(formula = cmedv ~ lstat + lat, data = xTrain)
##
## Coefficients:
## (Intercept)        lstat          lat
##   -240.9153      -0.9647       6.5336
```

- Predict with test data.

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

```
##         5         6        10        13        25        37        38        39
## 30.299340 30.454307 19.062246 20.337865 19.609264 24.280333 26.905584 25.600105
##        40        49        55        59        62        63        66        71
## 31.371705  5.839268 21.643704 29.136936 21.703525 29.170880 31.243129 29.075990
##        74        75        81        82        94       125       134       141
## 28.128118 28.841701 30.403264 28.587099 29.437323 18.135300 20.595324 11.697311
##       146       153       159       164       170       171       180       182
##  8.106039 23.219705 28.757936 31.761467 24.109735 21.120578 30.205387 25.868671
##       183       187       188       195       197       221       230       232
## 30.338569 30.796783 28.725831 31.248487 31.332094 25.523108 31.047556 29.690493
##       235       237       240       244       246       247       248       249
## 27.127802 25.669469 27.606096 29.614438 16.742759 25.761651 24.815077 25.464662
##       254       260       266       272       274       276       277       280
## 31.207525 28.193450 24.719709 28.022095 28.133012 31.751685 28.724021 30.043711
##       296       297       304       309       311       312       317       325
## 28.207320 27.088944 29.589782 30.130432 22.450177 28.888248 16.781275 28.416709
##       339       350       352       354       356       360       363       365
## 26.182256 28.661461 28.704335 29.469930 28.316814 22.623776 25.022804 29.761668
##       369       376       380       381       384       398       407       411
## 31.724614 21.992016 13.969159 18.394242 11.283984 15.599537 12.413895 25.096714
##       413       416       421       424       427       441       443       444
##  1.663323  6.745300 20.304427 12.298792 19.623440 13.422129 18.724489 16.545539
##       456       462       465       482       483       485       486       487
## 17.166958 20.501300 21.812240 27.157681 27.819455 21.631139 24.238217 20.055539
##       490       495       502       504
## 11.895635 22.017182 25.676034 29.529862
```

- Predict training data

```
predict(fitTrain)
```

```
##          1          2          3          4          7          8          9
## 30.3560469 26.5551726 31.4554660 32.5723411 23.4433286 17.0453790  6.6588168
##         11         12         14         15         16         17         18
## 15.8304507 22.7636391 27.4335230 25.4714248 27.2439999 29.0967126 21.2039639
##         19         20         21         22         23         24         26
## 24.0624811 24.4259999 15.0257542 21.9881904 17.2576661 16.1385966 19.4223544
##         27         28         29         30         31         32         33
## 21.0819709 18.6697234 22.9753134 23.7533127 13.5145713 22.7209143  8.5685430
##         34         35         36         41         42         43         44
## 17.5950101 15.6490927 25.9198891 33.7010081 31.0137919 30.0028823 28.4317033
##         45         46         47         48         50         51         52
## 26.3523794 25.6535986 21.8526209 17.3993632 19.9618512 22.7010609 26.6066567
##         53         54         56         57         58         60         61
## 30.6611864 27.5994662 31.3432772 30.3714157 32.1108634 26.7687388 23.0100761
##         64         65         67         68         69         70         72
## 26.5443535 27.9823919 25.9186674 27.8766608 23.0875597 27.1508986 25.9281797
##         73         76         77         78         79         80         83
## 30.1617788 26.7481156 23.7988945 25.4781116 23.5170858 26.7080985 29.1413254
##         84         85         86         87         88         89         90
```

```
## 28.3203983 26.2227805 29.1547487 23.0807695 27.2370068 30.0151206 29.8149907
##         91         92         93         95         96         97         98
## 26.7722584 27.3574679 27.4711924 25.1334682 28.8919777 24.3674636 31.2066813
##         99        100        101        102        103        104        105
## 31.8044983 29.2736779 26.0694448 27.6956275 24.8825388 22.1553553 23.2196556
##        106        107        108        109        110        111        112
## 19.1898003 17.0809942 21.5119566 23.2938724 20.1524742 22.6549659 25.3980238
##        113        114        115        116        117        118        119
## 19.5092286 18.6929471 25.0542275 19.9531507 23.5830526 25.2694972 20.3522571
##        120        121        122        123        124        126        127
## 22.0664897 21.2535967 21.3644421 17.8107172 10.5717803 20.7951475  8.7609230
##        128        129        130        131        132        133        135
## 18.4494704 20.1826909 17.3485416 22.9003804 23.2349172 24.3379592 18.3663724
##        136        137        138        139        140        142        143
## 18.6759284 18.7161707 20.9381291 14.4129283 17.1883476  1.7782734  9.0723671
##        144        145        147        148        149        150        151
##  9.4634800  6.6784136 18.8756800  6.4579890  7.6370548 14.2581143 21.3526909
##        152        154        155        156        157        158        160
## 22.1235033 19.6661336 20.3203331 20.4089643 19.3611511 30.5591464 27.8174359
##        161        162        163        165        166        167        168
## 29.6348674 33.3162717 33.1120684 23.7448375 25.5298669 31.4308101 23.3147484
##        169        172        173        174        175        176        177
## 24.3089052 23.4502681 20.9011130 26.3739678 25.8114726 30.0347305 25.4070582
##        178        179        181        184        185        186        189
## 29.0674425 28.4322310 27.7331438 29.5174073 21.5102715 22.3436529 30.7612273
##        190        191        192        193        194        196        198
## 30.0265027 30.3382845 30.7762859 32.5451346 30.6018214 32.2787905 26.9684682
##        199        200        201        202        203        204        205
## 28.9960551 30.6795578 30.6550052 27.6037479 31.9183225 31.3344917 32.0604975
##        206        207        208        209        210        211        212
## 24.6052722 24.4192909 17.5533262 20.8019967 12.6674879 18.2559958 11.8082380
##        213        214        215        216        217        218        219
## 19.4999376 25.9838958  6.5223243 25.8461096 21.9179527 25.6064313 17.6570253
##        220        222        223        224        225        226        227
## 24.7923443 14.2066524 25.3089107 27.5887115 30.8462631 30.3180174 31.7977580
##        228        229        231        233        234        236        238
## 28.6307663 30.9212957 23.4836482 32.3828549 31.0164919 24.3970048 30.2574805
##        239        241        242        243        245        250        251
## 28.5837246 23.5873159 22.6065732 23.8344468 22.5003013 28.3581138 29.0503611
##        252        253        255        256        257        258        259
## 31.3258947 31.3530699 28.1851276 25.5389289 31.4413725 29.9093359 27.3361599
##        261        262        263        264        265        267        268
## 25.5754984 27.8102176 29.0995160 23.9348699 27.0063904 20.5099779 27.5745001
##        269        270        271        273        275        278        279
## 31.8014843 21.3320768 21.8742054 26.9811217 31.1831973 30.6113613 27.7425030
##        281        282        283        284        285        286        287
## 31.0887172 30.2736298 31.7240505 31.4094707 26.6432148 26.2080207 21.4615187
##        288        289        290        291        292        293        294
## 27.1125551 26.7765899 24.9601182 31.0886636 30.9798098 29.8140455 25.9357060
##        295        298        299        300        301        302        303
## 24.2563671 18.8828723 29.1733192 29.3070006 28.1186661 24.9501646 25.7966131
##        305        306        307        308        310        313        314
## 27.7561606 25.8757320 28.2913995 27.2524675 24.9214300 23.2789134 26.9268840
##        315        316        318        319        320        321        322
```

```
## 25.5413485 23.4323487 19.0477432 24.4798555 22.2000141 27.3911506 27.7421744
##        323        324        326        327        328        329        330
## 26.9591014 23.0668790 29.2899949 28.3289655 21.9493911 24.6143526 27.1646206
##        331        332        333        334        335        336        337
## 25.3652985 22.0680140 26.5253044 28.6647775 27.6423327 26.5182614 24.8763576
##        338        340        341        342        343        344        345
## 24.1366405 25.0387775 25.5140607 29.1551508 26.1948540 27.5313157 29.8505615
##        346        347        348        349        351        353        355
## 23.8095029 21.8103481 27.8781000 28.3794363 28.4766332 26.3646193 26.0680585
##        357        358        359        361        362        364        366
## 17.9343730 22.0776151 23.7913881 27.3074122 21.1770101 20.7755561 27.9812123
##        367        368        370        371        372        373        374
## 21.3289449 21.9870644 31.2842645 32.0395089 25.7117841 26.3532228  1.3930813
##        375        377        378        379        382        383        385
## -1.7070743 12.5508749 14.4835720 12.1089128 14.6542603 12.2212181  5.4052960
##        386        387        388        389        390        391        392
##  5.2172734  7.6697632  4.1017770  5.4280103 14.8702646 18.4907130 16.9185337
##        393        394        395        396        397        399        400
## 10.1610353 20.1727872 19.0233162 18.2824456 16.1098765  5.3491480  5.9093767
##        401        402        403        404        405        406        408
##  8.9827447 15.2077709 15.1997774 15.7154968  8.3877349 12.6690695 23.1577897
##        409        410        412        414        415        417        418
##  9.3651702 15.7581181 14.3610880 15.4393025 -0.8839843  9.8981112  9.1055443
##        419        420        422        423        425        426        428
## 14.9013691 12.8627065 19.6667150 21.1788981 18.1942020 11.2500191 20.7678376
##        429        430        431        432        433        434        435
## 14.0011110 11.5183731 17.7115380 15.7273384 23.1000692 19.0324318 20.0630233
##        436        437        438        439        440        442        445
## 12.2651645 17.3179649  9.2300381  1.9408648 12.6890984 15.8717393 11.7981405
##        446        447        448        449        450        451        452
## 11.5952439 17.5524569 18.8365284 17.1930927 16.0710627 17.8522125 17.5626448
##        453        454        455        457        458        459        460
## 18.0031470 18.5013789 16.5995833 16.2886080 18.2627010 18.9411152 20.4171293
##        461        463        464        466        467        468        469
## 18.7826468 21.1053439 24.6617228 20.9408827 18.0503104 14.0895171 17.1016228
##        470        471        472        473        474        475        476
## 20.3821139 18.8766987 22.1877849 20.7843339 23.4478668 17.1899791 11.4722912
##        477        478        479        480        481        484        488
## 16.7017007 10.7176595 17.3503263 22.1176322 24.2700692 24.4925350 23.5001851
##        489        491        492        493        494        496        497
## 17.6029813  6.4086718 17.6181617 22.2140854 23.5166040 18.1793816 14.7499161
##        498        499        500        501        503        505        506
## 21.5225635 22.6341397 20.5147268 21.2248897 26.2210418 28.7064338 27.3362319
```

Using regression for hypothesis testing

- Three random sample with different mean and sd.

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

- Compare x1 and x2

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

- Build regression model

```
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
```

- Compare x1 and x2 mean values

```
mean(x1)
```

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

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

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

- Regression model with x1 and x3

```
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
```

- Regression with x1,x2, and x3

```
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
```

- Regression model with x1 and x4 (different sample sizes)

```
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
```

Works just fine! We model the means, so the number of observations will only help with better estimates of the coefficients, but not with the design of the comparison itself.

2. Exercises for regression building.

I utilize two distinct samples in order to assess and contrast the adequacy of the model fit as well as the accuracy of the model predictions.

1. Data Processing

```
data(BostonHousing2)
x <- BostonHousing2
x <- x[,-5]
```

- Convert to Data frame and name as df

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

Note: In the laboratory experiment model that we constructed using the stepwise method, we have substantiated that the 'town' variable holds less significance in predicting Boston housing prices, as it led to a notable increase in prediction errors. Therefore, for the upcoming exercise, I will proceed without factoring in the 'town' variable.

- Remove town column

```
# Define the list of columns to keep in the samples (exclude "town")
columns_to_keep <- colnames(x)[!colnames(x) %in% c("town")]
```

- Split data & store in a list

```r
# Set the seed for reproducibility
set.seed(123)

# Define the sample sizes
sample_sizes <- c(50, 100)

# Initialize empty data frames to store the datasets
xTest <- list()
xTrain <- list()

# Loop through each sample size
for (i in 1:length(sample_sizes)) {
  size <- sample_sizes[i]

  # Generate random indices
  idx <- sort(sample(1:nrow(x), size))

  # Create test and train datasets for the current size
  xTest[[i]] <- x[idx, columns_to_keep]
  xTrain[[i]] <- x[-idx, columns_to_keep]
}
```

2. Modeling

```r


fit_min <- list()
fit_min[[1]] <- lm(cmedv ~ 1, data=xTrain[[1]])
fit_min[[2]] <- lm(cmedv ~ 1, data=xTrain[[2]])

summary(fit_min[[1]])
```

Min Model - fit_min

```
##
## Call:
## lm(formula = cmedv ~ 1, data = xTrain[[1]])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.671  -5.396  -1.271   2.329  27.329
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.6711     0.4267   53.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.112 on 455 degrees of freedom
```

```r
summary(fit_min[[2]])
```

```
##
## Call:
## lm(formula = cmedv ~ 1, data = xTrain[[2]])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.629  -5.904  -1.679   2.371  27.371
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    22.63       0.48   47.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.672 on 405 degrees of freedom
```

```
fit_full <- list()
fit_full[[1]] <- lm(cmedv ~ ., data=xTrain[[1]])
fit_full[[2]] <- lm(cmedv ~ ., data=xTrain[[2]])

summary(fit_full[[1]])
```

Full model - fit_full

```
##
## Call:
## lm(formula = cmedv ~ ., data = xTrain[[1]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.9040  -2.6613  -0.5688   1.8495  25.2163
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.367e+02  3.358e+02  -0.705 0.481357
## tract       -7.956e-04  4.624e-04  -1.721 0.085993 .
## lon         -5.252e+00  3.593e+00  -1.462 0.144552
## lat         -2.247e+00  5.507e+00  -0.408 0.683397
## crim        -1.185e-01  3.400e-02  -3.485 0.000542 ***
## zn           4.690e-02  1.416e-02   3.312 0.001004 **
## indus        3.239e-03  6.190e-02   0.052 0.958295
## chas1        1.861e+00  9.083e-01   2.048 0.041125 *
## nox         -1.702e+01  4.175e+00  -4.076 5.44e-05 ***
## rm           3.593e+00  4.318e-01   8.321 1.11e-15 ***
## age          9.252e-03  1.375e-02   0.673 0.501402
## dis         -1.434e+00  2.143e-01  -6.690 6.82e-11 ***
## rad          2.029e-01  8.841e-02   2.295 0.022179 *
## tax         -1.314e-02  3.829e-03  -3.432 0.000657 ***
## ptratio     -9.781e-01  1.449e-01  -6.749 4.73e-11 ***
## b            8.276e-03  2.801e-03   2.955 0.003294 **
## lstat       -5.573e-01  5.243e-02 -10.629  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.588 on 439 degrees of freedom
## Multiple R-squared:  0.7554, Adjusted R-squared:  0.7464
## F-statistic: 84.72 on 16 and 439 DF,  p-value: < 2.2e-16
```

```
summary(fit_full[[2]])
```

```
##
## Call:
## lm(formula = cmedv ~ ., data = xTrain[[2]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.2781  -2.7532  -0.6698   1.9767  26.7294
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.294e+02  3.726e+02  -1.153 0.249779
## tract       -5.576e-04  5.037e-04  -1.107 0.269005
## lon         -6.867e+00  4.010e+00  -1.713 0.087555 .
## lat         -5.860e-01  6.188e+00  -0.095 0.924605
## crim        -1.249e-01  4.039e-02  -3.093 0.002127 **
## zn           4.850e-02  1.594e-02   3.043 0.002502 **
## indus        1.763e-02  7.303e-02   0.241 0.809356
## chas1        2.878e+00  1.065e+00   2.702 0.007184 **
## nox         -1.568e+01  4.802e+00  -3.265 0.001193 **
## rm           4.242e+00  5.083e-01   8.344 1.26e-15 ***
## age         -2.190e-03  1.538e-02  -0.142 0.886856
## dis         -1.456e+00  2.397e-01  -6.073 2.99e-09 ***
## rad          2.538e-01  9.543e-02   2.659 0.008152 **
## tax         -1.442e-02  4.336e-03  -3.325 0.000968 ***
## ptratio     -9.001e-01  1.611e-01  -5.587 4.35e-08 ***
## b            1.102e-02  3.090e-03   3.567 0.000406 ***
## lstat       -4.919e-01  6.231e-02  -7.895 2.99e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.881 on 389 degrees of freedom
## Multiple R-squared:  0.7554, Adjusted R-squared:  0.7453
## F-statistic: 75.07 on 16 and 389 DF,  p-value: < 2.2e-16
```

```
fit_best <- list()
fit_best[[1]] <- step(fit_min[[1]], direction = "both", scope = formula(fit_full[[1]]), trace = 0)

fit_best[[2]] <- step(fit_min[[2]], direction = "both", scope = formula(fit_full[[2]]), trace = 0)

summary(fit_best[[1]])
```

Best model - fit_best(direction - both)

```
##
## Call:
## lm(formula = cmedv ~ lstat + rm + ptratio + dis + nox + b + zn +
##     chas + crim + tract + tax + rad, data = xTrain[[1]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.0786  -2.6586  -0.5149   1.9268  25.5253
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.193e+01  5.513e+00   7.605 1.72e-13 ***
## lstat       -5.464e-01  4.879e-02 -11.199  < 2e-16 ***
## rm           3.657e+00  4.170e-01   8.769  < 2e-16 ***
## ptratio     -1.010e+00  1.364e-01  -7.411 6.42e-13 ***
## dis         -1.559e+00  1.889e-01  -8.252 1.80e-15 ***
## nox         -1.781e+01  3.641e+00  -4.892 1.40e-06 ***
## b            8.258e-03  2.787e-03   2.963 0.003212 **
## zn           4.934e-02  1.365e-02   3.614 0.000336 ***
## chas1        2.118e+00  8.898e-01   2.380 0.017726 *
## crim        -1.220e-01  3.385e-02  -3.605 0.000347 ***
## tract       -5.386e-04  3.034e-04  -1.775 0.076594 .
## tax         -1.239e-02  3.460e-03  -3.581 0.000380 ***
## rad          2.370e-01  7.133e-02   3.323 0.000966 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.58 on 443 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.7473
## F-statistic: 113.1 on 12 and 443 DF,  p-value: < 2.2e-16
```

```
summary(fit_best[[2]])
```

```
##
## Call:
## lm(formula = cmedv ~ lstat + rm + ptratio + dis + nox + chas +
##     b + zn + crim + rad + tax + lon + tract, data = xTrain[[2]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.2238  -2.7993  -0.6825   1.9360  26.6773
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.696e+02  2.754e+02  -1.705 0.088913 .
## lstat       -4.944e-01  5.740e-02  -8.613  < 2e-16 ***
## rm           4.213e+00  4.865e-01   8.660  < 2e-16 ***
## ptratio     -8.905e-01  1.544e-01  -5.767 1.63e-08 ***
## dis         -1.451e+00  2.245e-01  -6.466 3.00e-10 ***
## nox         -1.541e+01  4.446e+00  -3.465 0.000588 ***
## chas1        2.895e+00  1.050e+00   2.758 0.006079 **
## b            1.097e-02  3.064e-03   3.580 0.000386 ***
## zn           4.828e-02  1.556e-02   3.103 0.002052 **
## crim        -1.249e-01  4.015e-02  -3.110 0.002011 **
```

```
## rad          2.533e-01  7.861e-02   3.223 0.001377 **
## tax         -1.392e-02  3.797e-03  -3.665 0.000281 ***
## lon         -7.080e+00  3.863e+00  -1.833 0.067635 .
## tract       -5.191e-04  3.465e-04  -1.498 0.134954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.863 on 392 degrees of freedom
## Multiple R-squared:  0.7553, Adjusted R-squared:  0.7472
## F-statistic: 93.08 on 13 and 392 DF,  p-value: < 2.2e-16
```

```r
fit_best_f <- list()
fit_best_f[[1]] <- step(fit_min[[1]], direction = "forward", scope = formula(fit_full[[1]]), trace = 0)

fit_best_f[[2]] <- step(fit_min[[2]], direction = "forward", scope = formula(fit_full[[2]]), trace = 0)

summary(fit_best_f[[1]])
```

Best model - fit_best_f(direction - forward)

```
##
## Call:
## lm(formula = cmedv ~ lstat + rm + ptratio + dis + nox + b + zn +
##     chas + crim + tract + tax + rad, data = xTrain[[1]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.0786  -2.6586  -0.5149   1.9268  25.5253
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.193e+01  5.513e+00   7.605 1.72e-13 ***
## lstat       -5.464e-01  4.879e-02 -11.199  < 2e-16 ***
## rm           3.657e+00  4.170e-01   8.769  < 2e-16 ***
## ptratio     -1.010e+00  1.364e-01  -7.411 6.42e-13 ***
## dis         -1.559e+00  1.889e-01  -8.252 1.80e-15 ***
## nox         -1.781e+01  3.641e+00  -4.892 1.40e-06 ***
## b            8.258e-03  2.787e-03   2.963 0.003212 **
## zn           4.934e-02  1.365e-02   3.614 0.000336 ***
## chas1        2.118e+00  8.898e-01   2.380 0.017726 *
## crim        -1.220e-01  3.385e-02  -3.605 0.000347 ***
## tract       -5.386e-04  3.034e-04  -1.775 0.076594 .
## tax         -1.239e-02  3.460e-03  -3.581 0.000380 ***
## rad          2.370e-01  7.133e-02   3.323 0.000966 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.58 on 443 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.7473
## F-statistic: 113.1 on 12 and 443 DF,  p-value: < 2.2e-16
```

```r
summary(fit_best_f[[2]])
```

```
##
## Call:
## lm(formula = cmedv ~ lstat + rm + ptratio + dis + nox + chas +
##     b + zn + crim + rad + tax + lon + tract, data = xTrain[[2]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.2238  -2.7993  -0.6825   1.9360  26.6773
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.696e+02  2.754e+02  -1.705 0.088913 .
## lstat       -4.944e-01  5.740e-02  -8.613  < 2e-16 ***
## rm           4.213e+00  4.865e-01   8.660  < 2e-16 ***
## ptratio     -8.905e-01  1.544e-01  -5.767 1.63e-08 ***
## dis         -1.451e+00  2.245e-01  -6.466 3.00e-10 ***
## nox         -1.541e+01  4.446e+00  -3.465 0.000588 ***
## chas1        2.895e+00  1.050e+00   2.758 0.006079 **
## b            1.097e-02  3.064e-03   3.580 0.000386 ***
## zn           4.828e-02  1.556e-02   3.103 0.002052 **
## crim        -1.249e-01  4.015e-02  -3.110 0.002011 **
## rad          2.533e-01  7.861e-02   3.223 0.001377 **
## tax         -1.392e-02  3.797e-03  -3.665 0.000281 ***
## lon         -7.080e+00  3.863e+00  -1.833 0.067635 .
## tract       -5.191e-04  3.465e-04  -1.498 0.134954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.863 on 392 degrees of freedom
## Multiple R-squared:  0.7553, Adjusted R-squared:  0.7472
## F-statistic: 93.08 on 13 and 392 DF,  p-value: < 2.2e-16
```

```r
fit_best_b <- list()
fit_best_b[[1]] <- step(fit_full[[1]], direction = "backward", scope = formula(fit_full[[1]]), trace = 0

fit_best_b[[2]] <- step(fit_full[[2]], direction = "backward", scope = formula(fit_full[[2]]), trace = 0

summary(fit_best_b[[1]])
```

Best model - fit_best_b(direction - backward)

```
##
## Call:
## lm(formula = cmedv ~ tract + crim + zn + chas + nox + rm + dis +
##     rad + tax + ptratio + b + lstat, data = xTrain[[1]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -15.0786  -2.6586  -0.5149   1.9268  25.5253
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.193e+01  5.513e+00   7.605 1.72e-13 ***
## tract       -5.386e-04  3.034e-04  -1.775 0.076594 .
## crim        -1.220e-01  3.385e-02  -3.605 0.000347 ***
## zn           4.934e-02  1.365e-02   3.614 0.000336 ***
## chas1        2.118e+00  8.898e-01   2.380 0.017726 *
## nox         -1.781e+01  3.641e+00  -4.892 1.40e-06 ***
## rm           3.657e+00  4.170e-01   8.769  < 2e-16 ***
## dis         -1.559e+00  1.889e-01  -8.252 1.80e-15 ***
## rad          2.370e-01  7.133e-02   3.323 0.000966 ***
## tax         -1.239e-02  3.460e-03  -3.581 0.000380 ***
## ptratio     -1.010e+00  1.364e-01  -7.411 6.42e-13 ***
## b            8.258e-03  2.787e-03   2.963 0.003212 **
## lstat       -5.464e-01  4.879e-02 -11.199  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.58 on 443 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.7473
## F-statistic: 113.1 on 12 and 443 DF,  p-value: < 2.2e-16
```

```
summary(fit_best_b[[2]])
```

```
##
## Call:
## lm(formula = cmedv ~ tract + lon + crim + zn + chas + nox + rm +
##     dis + rad + tax + ptratio + b + lstat, data = xTrain[[2]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.2238  -2.7993  -0.6825   1.9360  26.6773
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.696e+02  2.754e+02  -1.705 0.088913 .
## tract       -5.191e-04  3.465e-04  -1.498 0.134954
## lon         -7.080e+00  3.863e+00  -1.833 0.067635 .
## crim        -1.249e-01  4.015e-02  -3.110 0.002011 **
## zn           4.828e-02  1.556e-02   3.103 0.002052 **
## chas1        2.895e+00  1.050e+00   2.758 0.006079 **
## nox         -1.541e+01  4.446e+00  -3.465 0.000588 ***
## rm           4.213e+00  4.865e-01   8.660  < 2e-16 ***
## dis         -1.451e+00  2.245e-01  -6.466 3.00e-10 ***
## rad          2.533e-01  7.861e-02   3.223 0.001377 **
## tax         -1.392e-02  3.797e-03  -3.665 0.000281 ***
## ptratio     -8.905e-01  1.544e-01  -5.767 1.63e-08 ***
## b            1.097e-02  3.064e-03   3.580 0.000386 ***
## lstat       -4.944e-01  5.740e-02  -8.613  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.863 on 392 degrees of freedom
## Multiple R-squared:  0.7553, Adjusted R-squared:  0.7472
## F-statistic: 93.08 on 13 and 392 DF,  p-value: < 2.2e-16
```

3. Model Selection

```
aic <- list()
aic[[1]] <- c(AIC(fit_min[[1]]),AIC(fit_full[[1]]),AIC(fit_best[[1]]),AIC(fit_best_f[[1]]),AIC(fit_best

aic[[1]] <- round(aic[[1]],4)

aic[[2]] <- c(AIC(fit_min[[2]]),AIC(fit_full[[2]]),AIC(fit_best[[2]]),AIC(fit_best_f[[2]]),AIC(fit_best

aic[[2]] <- round(aic[[2]],4)

# Combine the list of data frames into a single dataframe
aic_df <- do.call(rbind, aic)


# Add column names to the dataframe
colnames(aic_df) <- c("Min", "Full", "Bi-Direction", "Forward", "Backward")

# Define row names
row_names <- c("Sample 50", "Sample 100")

# Set row names for the dataframe
rownames(aic_df) <- row_names


print(as.data.frame(aic_df))
```

```
##                 Min     Full Bi-Direction  Forward Backward
## Sample 50  3312.214 2702.174     2696.753 2696.753 2696.753
## Sample 100 2997.811 2458.160     2452.249 2452.249 2452.249
```

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.2.3
```

4. Predictions

- Calculate RMSE

```
rmse_min <- list()
rmse_full <- list()
rmse_best <- list()

# Calculate RMSE for each model
rmse_min[[1]] <-
  rmse(predict(fit_min[[1]], newdata = xTest[[1]]), xTest[[1]]$cmedv)
```

```
rmse_min[[2]] <-
  rmse(predict(fit_min[[2]], newdata = xTest[[2]]), xTest[[2]]$cmedv)

rmse_full[[1]] <-
  rmse(predict(fit_full[[1]], newdata = xTest[[1]]), xTest[[1]]$cmedv)
rmse_full[[2]] <-
  rmse(predict(fit_full[[2]], newdata = xTest[[2]]), xTest[[2]]$cmedv)

rmse_best[[1]] <-
  rmse(predict(fit_best[[2]], newdata = xTest[[1]]), xTest[[1]]$cmedv)
rmse_best[[2]] <-
  rmse(predict(fit_best[[2]], newdata = xTest[[2]]), xTest[[2]]$cmedv)

# Combine RMSE values into a matrix
rmse_matrix <- matrix(unlist(c(rmse_min, rmse_full, rmse_best)), nrow = 2, byrow = TRUE)

# Add column and row names
colnames(rmse_matrix) <- c("Min Model", "Full Model", "Best Model")
rownames(rmse_matrix) <- c("Sample 1", "Sample 2")

# Convert the matrix to a dataframe
rmse_df <- as.data.frame(rmse_matrix)
```

```
# Print the dataframe
print(rmse_df)
```

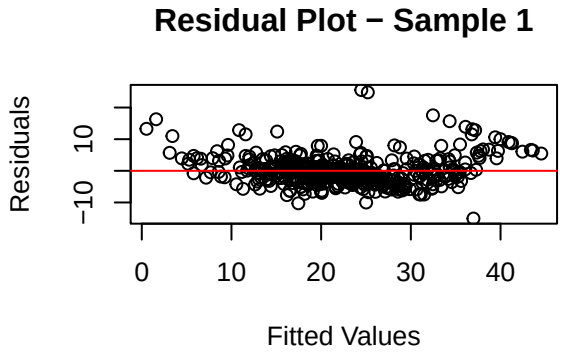
Print and compare results

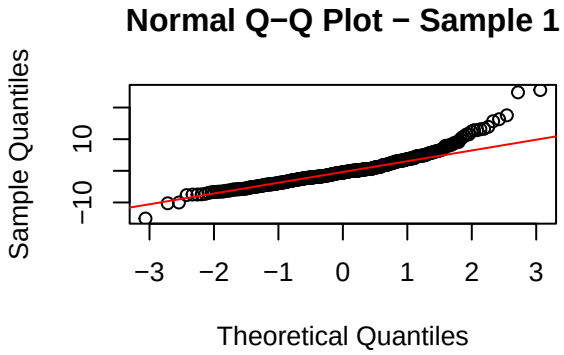
```
##          Min Model Full Model Best Model
## Sample 1  9.808580   6.851717   5.604421
## Sample 2  4.015471   5.446195   4.013075
```

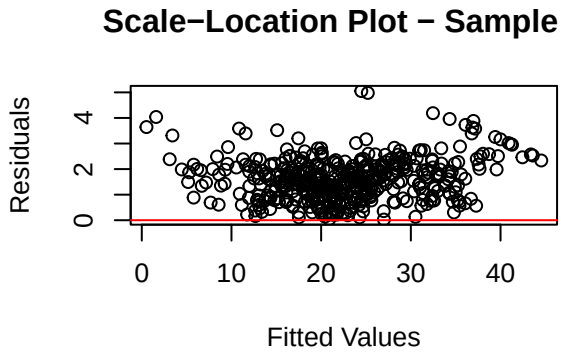
After conducting a comprehensive analysis of the errors, it has been ascertained that the best-performing model consistently exhibits superior predictive capabilities. In order to gain further insights into the predictive accuracy of this model, we have undertaken the task of visualizing the distribution of errors through the creation of residual plots.

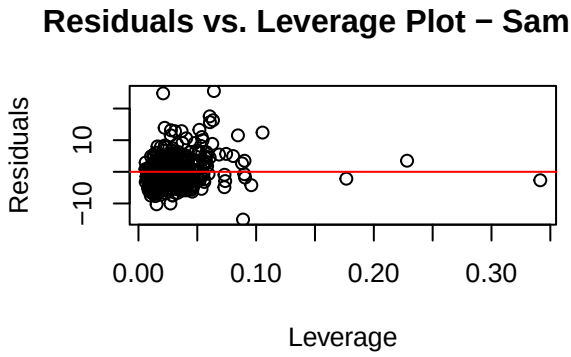
```
# Generate residuals for Sample 1 and Sample 2
residuals_sample1 <- resid(fit_best[[1]])
residuals_sample2 <- resid(fit_best[[2]])

# Create a 2x2 grid for the residual plots
par(mfrow = c(2, 2))

# Residual vs. Fitted Values Plot for Sample 1
plot(predict(fit_best[[1]]), residuals_sample1,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residual Plot - Sample 1")
abline(h = 0, col = "red")
```

```r
# Normal Q-Q Plot for Sample 1
qqnorm(residuals_sample1, main = "Normal Q-Q Plot - Sample 1")
qqline(residuals_sample1, col = "red")

# Scale-Location Plot for Sample 1
sqrt_abs_residuals_sample1 <- sqrt(abs(residuals_sample1))
plot(predict(fit_best[[1]]), sqrt_abs_residuals_sample1,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Scale-Location Plot - Sample 1")
abline(h = 0, col = "red")

# Residuals vs. Leverage Plot for Sample 1
plot(hatvalues(fit_best[[1]]), residuals_sample1,
     xlab = "Leverage", ylab = "Residuals",
     main = "Residuals vs. Leverage Plot - Sample 1")
abline(h = 0, col = "red")
```



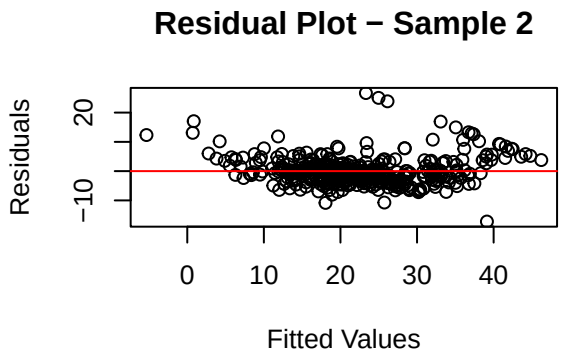


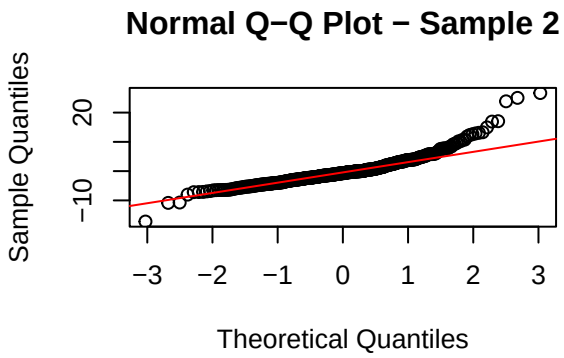


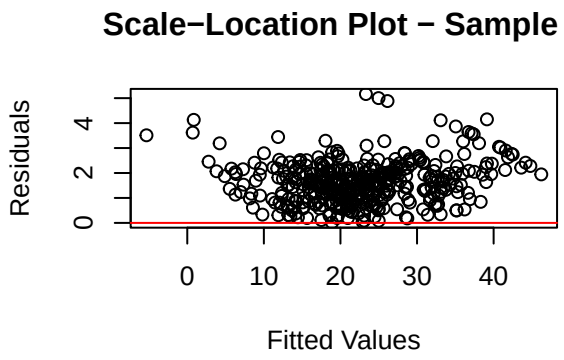


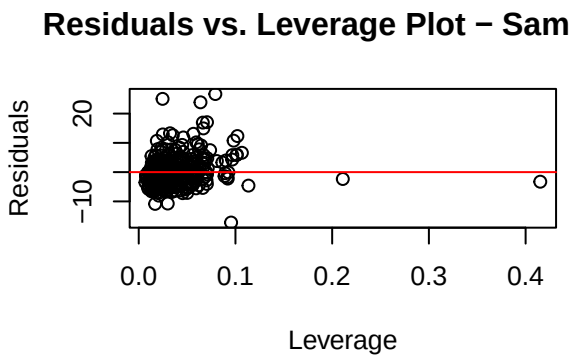

```r
# Residual vs. Fitted Values Plot for Sample 2
plot(predict(fit_best[[2]]), residuals_sample2,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Residual Plot - Sample 2")
abline(h = 0, col = "red")

# Normal Q-Q Plot for Sample 2
qqnorm(residuals_sample2, main = "Normal Q-Q Plot - Sample 2")
```

```r
qqline(residuals_sample2, col = "red")

# Scale-Location Plot for Sample 2
sqrt_abs_residuals_sample2 <- sqrt(abs(residuals_sample2))
plot(predict(fit_best[[2]]), sqrt_abs_residuals_sample2,
     xlab = "Fitted Values", ylab = "Residuals",
     main = "Scale-Location Plot - Sample 2")
abline(h = 0, col = "red")

# Residuals vs. Leverage Plot for Sample 2
plot(hatvalues(fit_best[[2]]), residuals_sample2,
     xlab = "Leverage", ylab = "Residuals",
     main = "Residuals vs. Leverage Plot - Sample 2")
abline(h = 0, col = "red")
```










```r
# Reset the plotting layout
par(mfrow = c(1, 1))
```

5. Conclusion

In summary, we conducted an extensive model analysis, including "Min," "Full," "Bi-Direction," "Forward," and "Backward" models. Upon evaluating the models, we discovered that the AIC values for "Bi-Direction," "Forward," and "Backward" were identical. Consequently, we opted to focus exclusively on the "Bi-Direction" model for further evaluation, alongside the "Min" and "Full" models. After comparing the RMSE values, it became evident that the "Bi-Direction" model consistently exhibited the best predictive performance.

To gain deeper insights, we examined residual plots, which confirmed the well-distributed nature of errors in the "Bi-Direction" model. In conclusion, the "Bi-Direction" model stands out as the optimal choice for our dataset, demonstrating superior accuracy and robustness in predicting outcomes.