Assmt5

Apurva Hari

July 20, 2016

```
#installing the MAss package
#install.packages("MASS")

#loading the package MASS
library(MASS)

#loading the dataset Boston
data("Boston")

#checking data type of the dataset Boston to verify if it is a data frame
class(Boston)
```

```
## [1] "data.frame"
```

```
#Finding the column names for Boston and siplaying them
col_names_Boston <- colnames(Boston)
col_names_Boston
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```
#lapply(Boston,class)  #Finding the class of each column in Boston
typeof(Boston$crim)
```

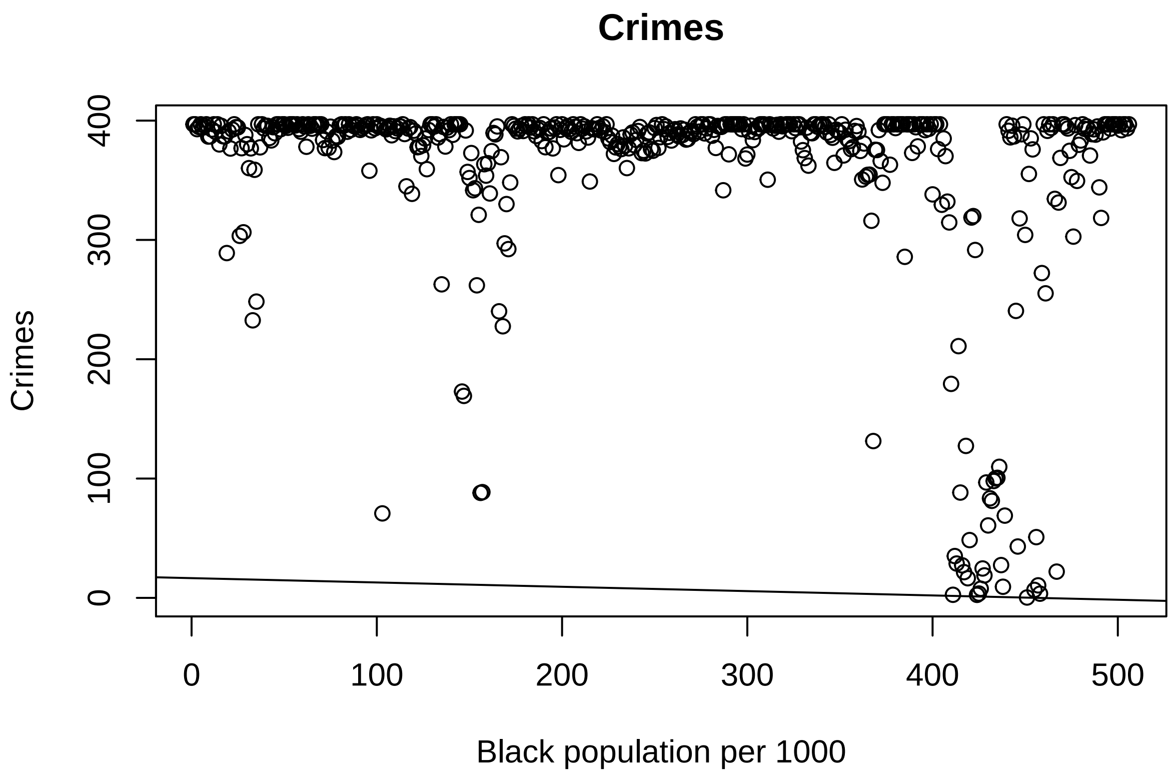
```
## [1] "double"
```

```
# Fitting the data with linear regression. Using the parameter black to predict the number of crimes pe
fit <- lm(formula = crim ~ black, data = Boston )

#Plotting the fitted data
plot( Boston$black, Boston$crib, main ="Crimes",xlab = "Black population per 1000",ylab ="Crimes")

#Getting the summary of the fitted linear model to understand some parameters of the model including -
summary(fit)
```

```
##
## Call:
## lm(formula = crim ~ black, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

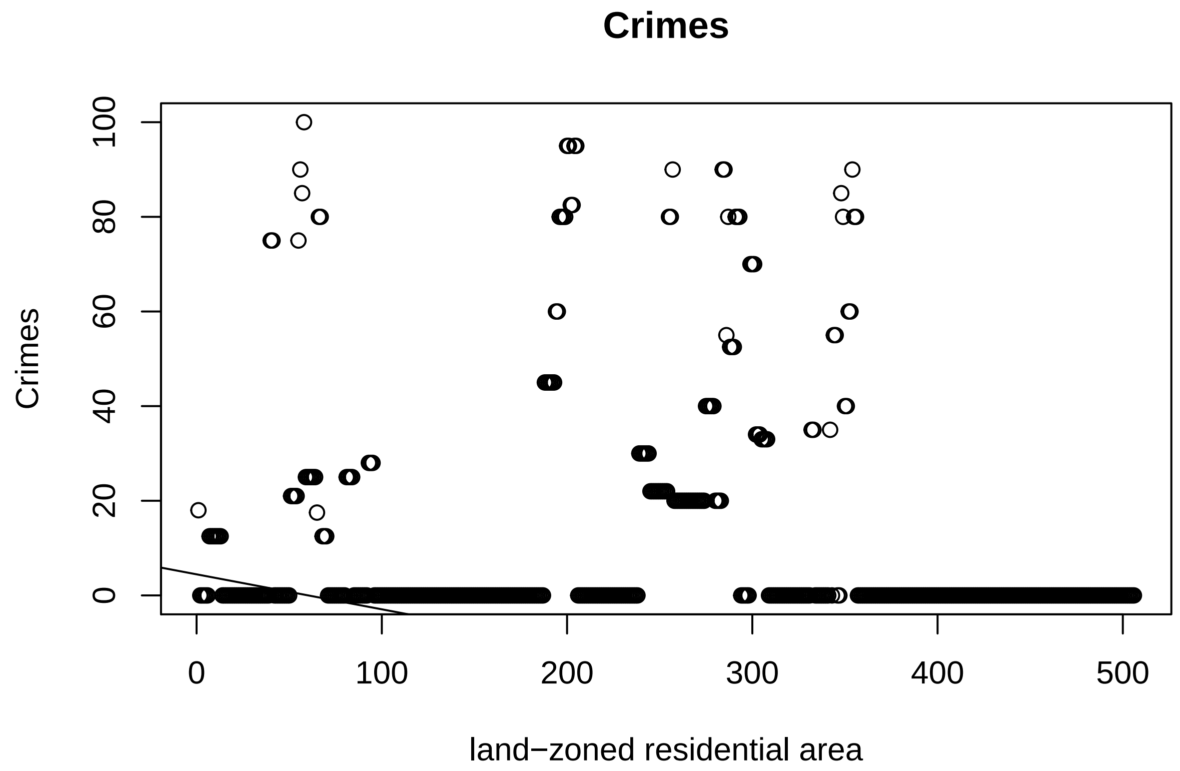
```
abline(fit) #Adding a straight line to the plot to see how the model fits
```



```
#Linear fit for predicting the per capita crime rate per town using proportion of residential land zone

fit_zn <- lm(formula = crim ~ zn, data = Boston )
plot( Boston$zn, Boston$crib, main ="Crimes",xlab = "land-zoned residential area",ylab ="Crimes")

summary(fit_zn)
```

```
##
## Call:
## lm(formula = crim ~ zn, data = Boston)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

```
abline(fit_zn)
```

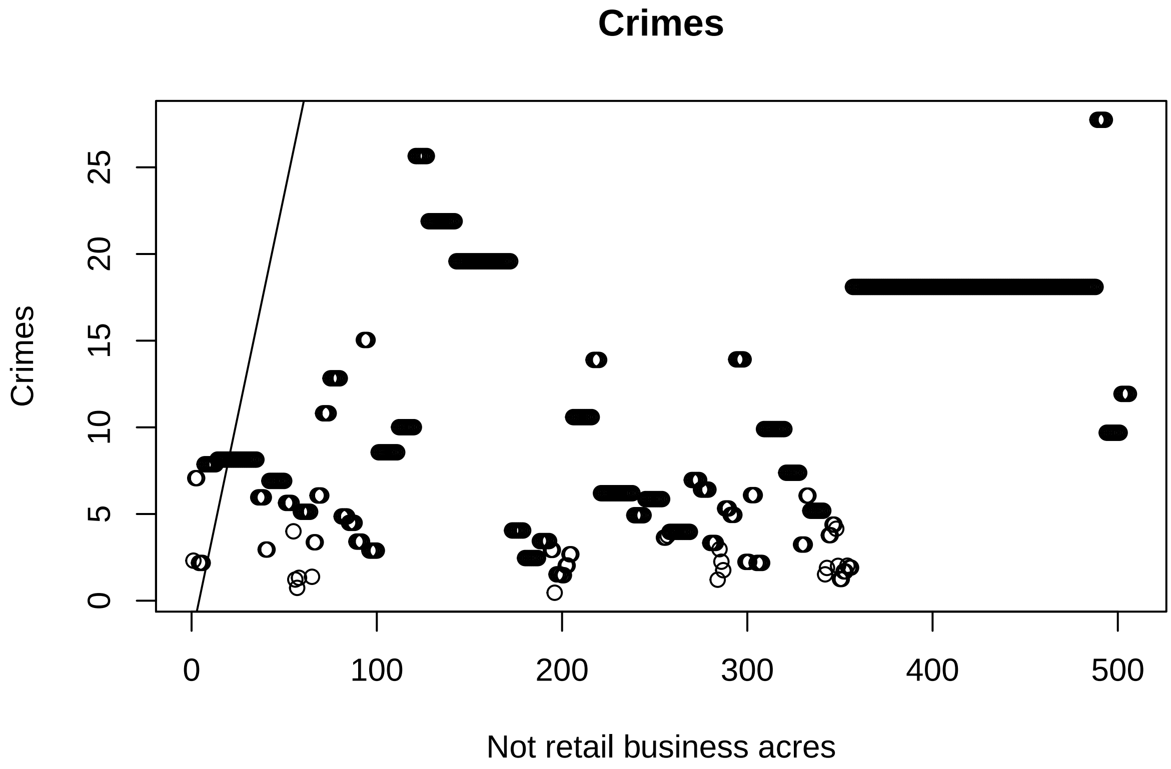


```
#Linear fit for predicting the per capita crime rate per town using proportion of non-retail business a

fit_indus <- lm(formula = crim ~ indus, data = Boston )
plot( Boston$indus, Boston$crib, main ="Crimes",xlab = "Not retail business acres",ylab ="Crimes")

summary(fit_indus)
```

```
##
## Call:
## lm(formula = crim ~ indus, data = Boston)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
abline(fit_indus)
```

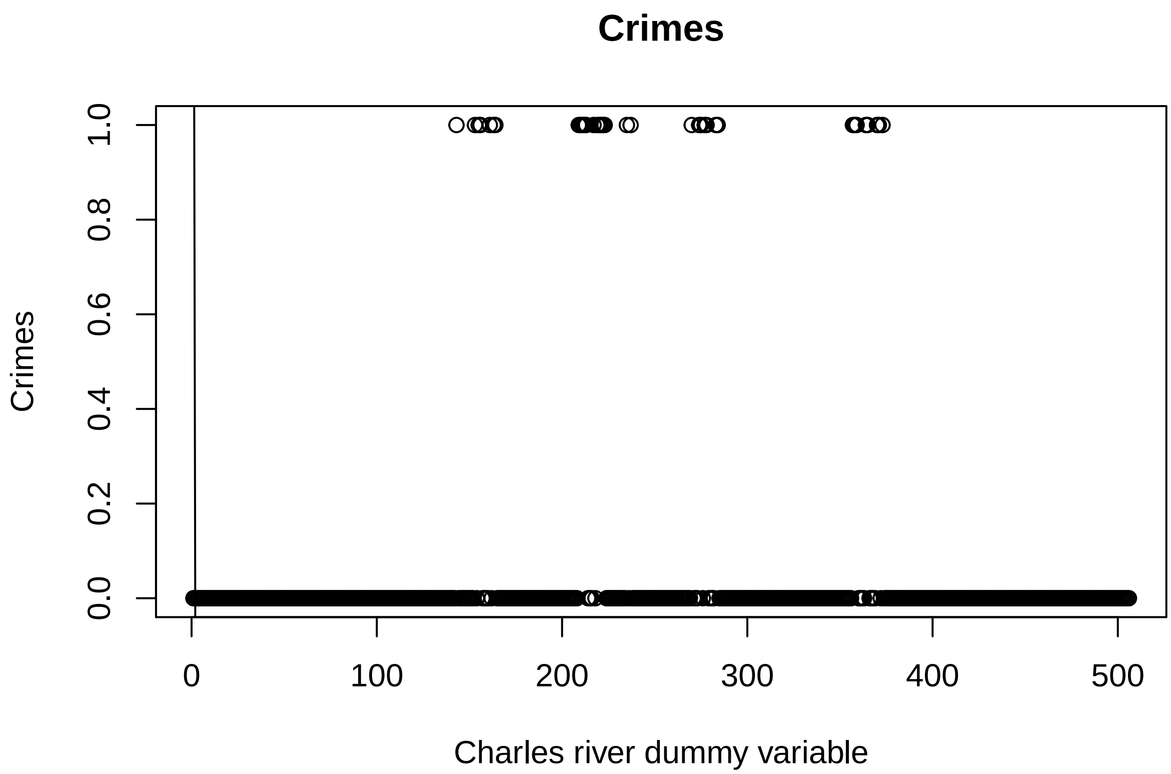


```
#Linear fit for predicting the per capita crime rate per town using Charles River dummy variable(chas)

fit_chas <- lm(formula = crim ~ chas, data = Boston )
plot( Boston$chas, Boston$crib, main ="Crimes",xlab = "Charles river dummy variable",ylab ="Crimes")

summary(fit_chas)
```

```
##
```

```
## Call:
## lm(formula = crim ~ chas, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

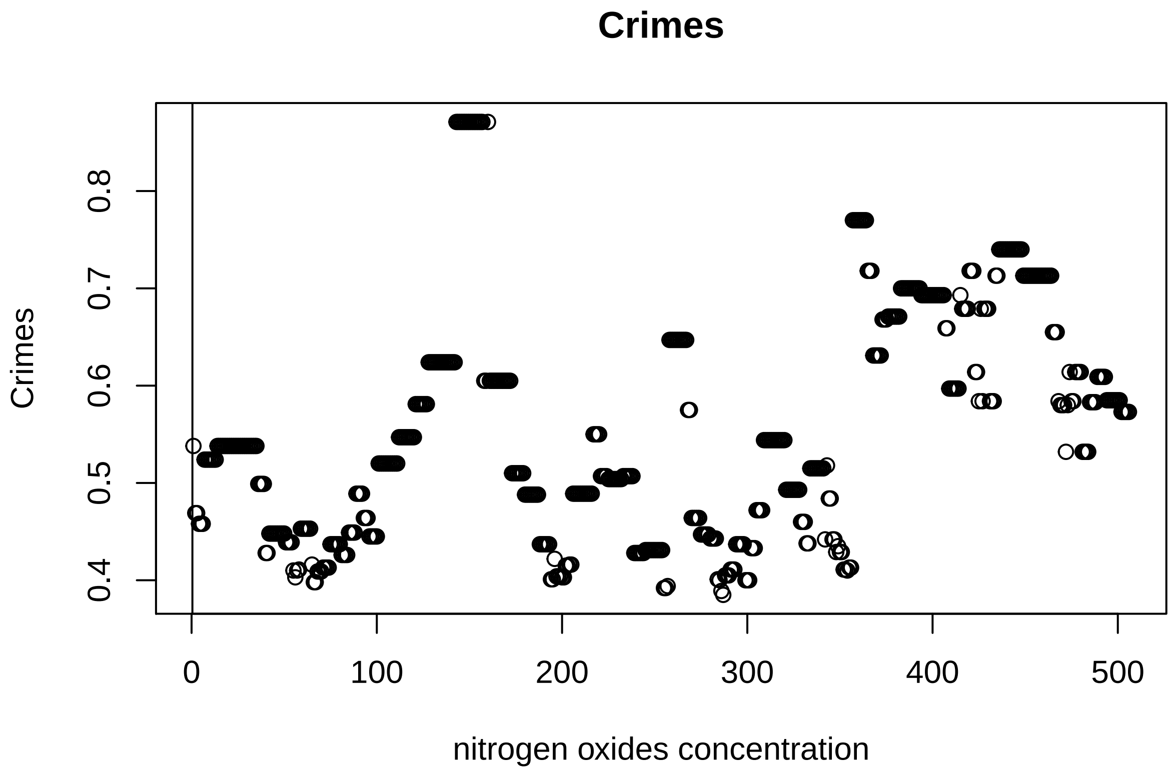
```
abline(fit_chas)
```



```
#Linear fit for predicting the per capita crime rates using nitrogen oxides concentration (nox)

fit_nox <- lm(formula = crim ~ nox, data = Boston )
plot( Boston$nox, Boston$crib, main ="Crimes",xlab = "nitrogen oxides concentration",ylab ="Crimes")

summary(fit_nox)
```

```
##
## Call:
## lm(formula = crim ~ nox, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

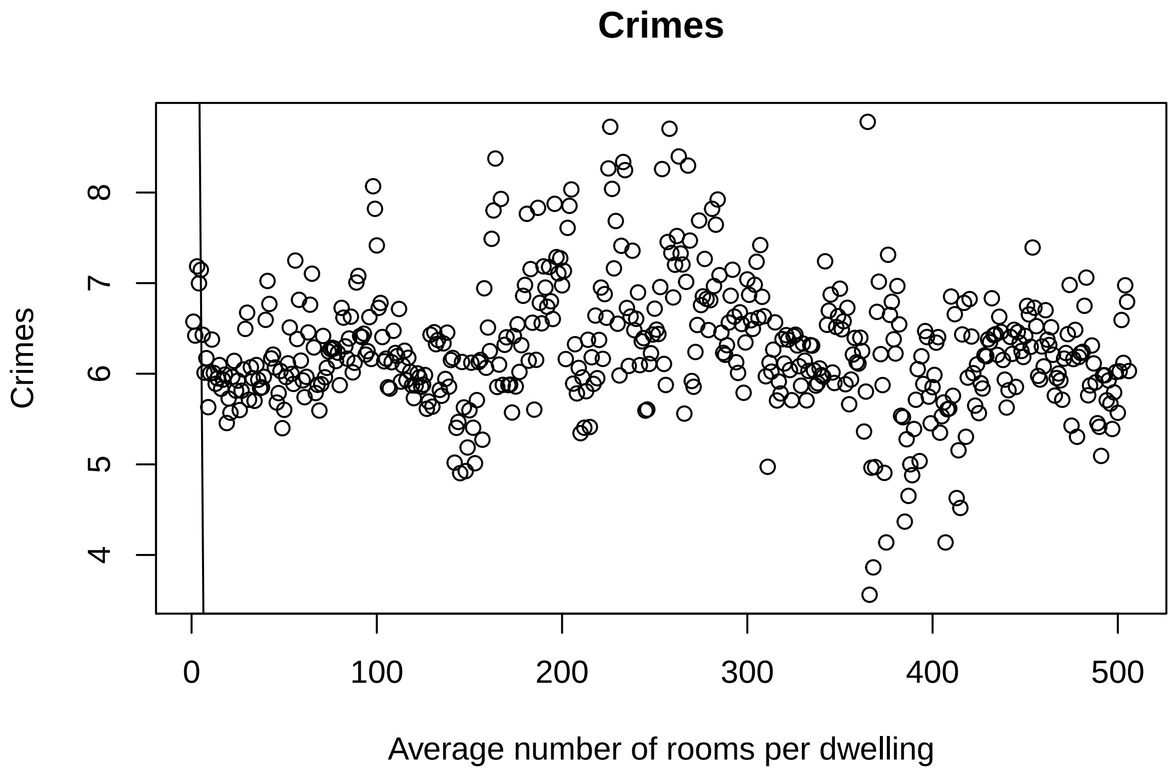
```
abline(fit_nox)
```



```r
#Linear fit for predicting the per capita crime rate per town using the average number of rooms per dwe

fit_rm <- lm(formula = crim ~ rm, data = Boston )
plot( Boston$rm, Boston$crib, main ="Crimes",xlab = "Average number of rooms per dwelling", ylab ="Crim

summary(fit_rm)
```

```
##
## Call:
## lm(formula = crim ~ rm, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

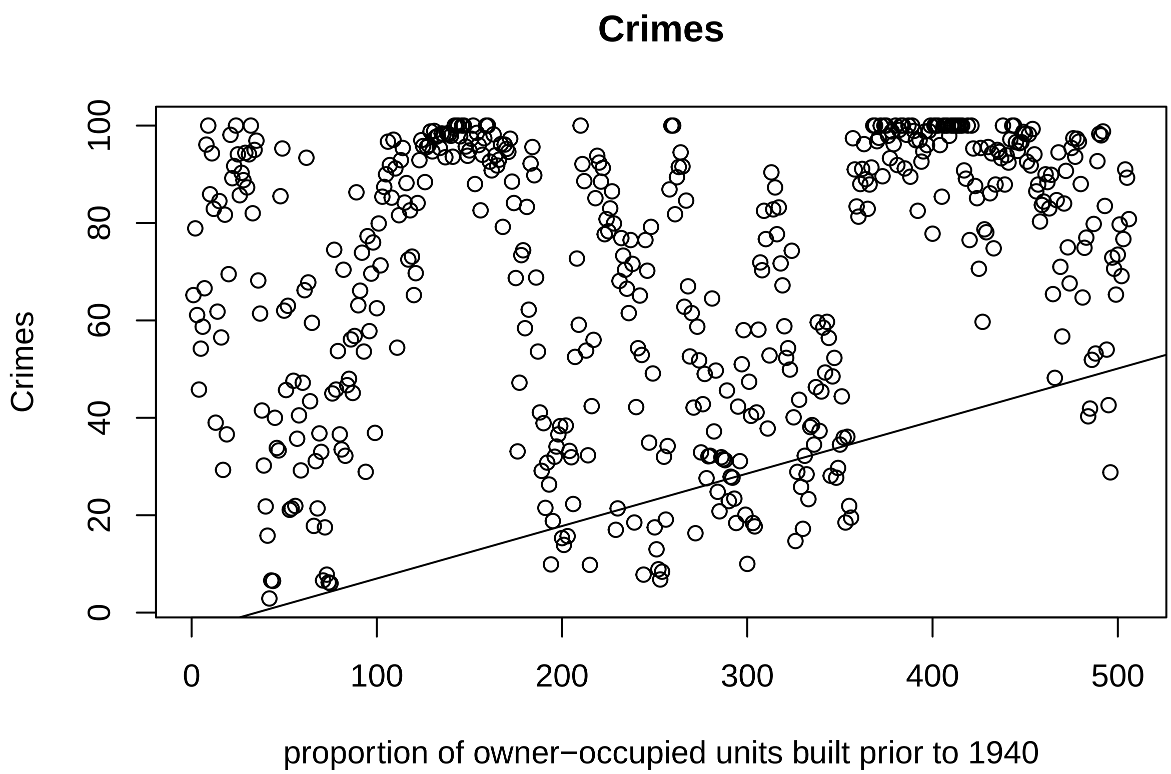
```r
abline(fit_rm)
```



```
#Linear fit for predicting the per capita crime rate per town using proportion of owner-occupied units

fit_age <- lm(formula = crim ~ age, data = Boston )
plot( Boston$age, Boston$crib, main ="Crimes",xlab = "proportion of owner-occupied units built prior to

summary(fit_age)
```

```
##
## Call:
## lm(formula = crim ~ age, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

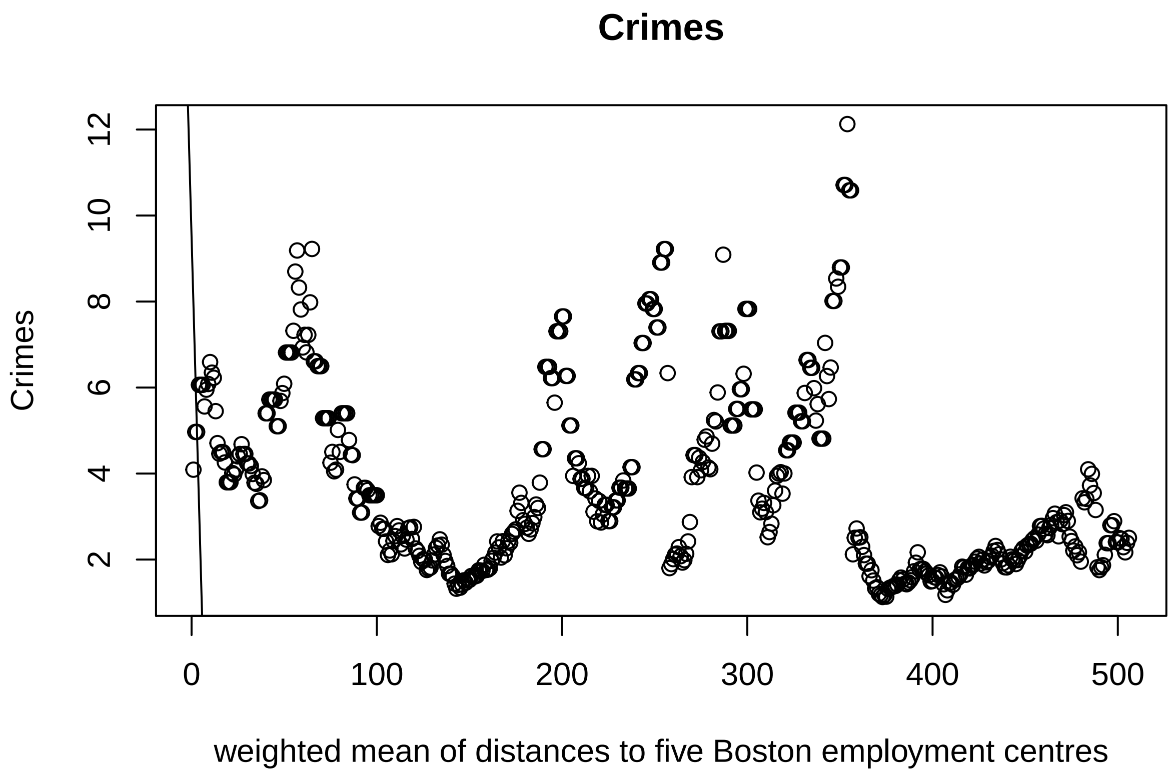
```r
abline(fit_age)
```



```r
#Linear fit for predicting the per capita crime rate per town using the weighted mean of distances to f

fit_dis <- lm(formula = crim ~ dis, data = Boston )
plot( Boston$dis, Boston$crib, main ="Crimes",xlab = "weighted mean of distances to five Boston employme

summary(fit_dis)
```

```
##
## Call:
## lm(formula = crim ~ dis, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

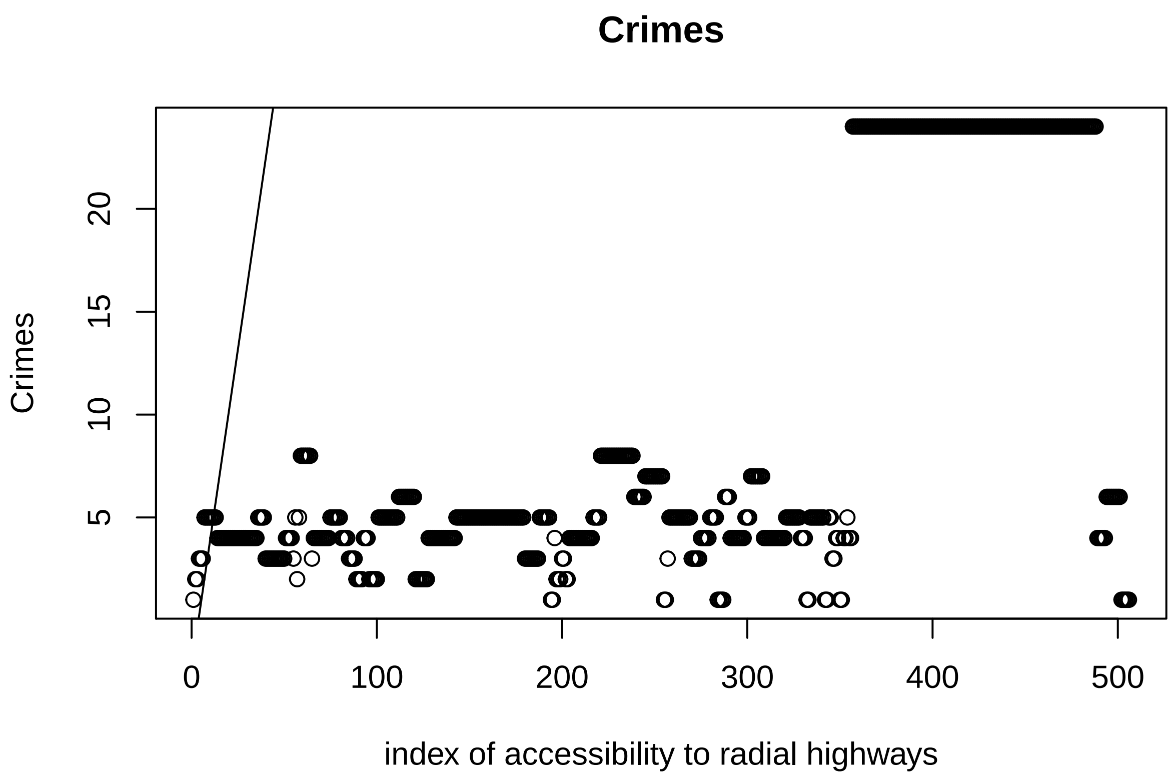
```
abline(fit_dis)
```



```
#Linear fit for predicting the per capita crime rate per town using the index of accessibility to radia

fit_rad <- lm(formula = crim ~ rad, data = Boston )
plot( Boston$rad, Boston$crib, main ="Crimes",xlab = "index of accessibility to radial highways",ylab =

summary(fit_rad)
```

```
##
## Call:
## lm(formula = crim ~ rad, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

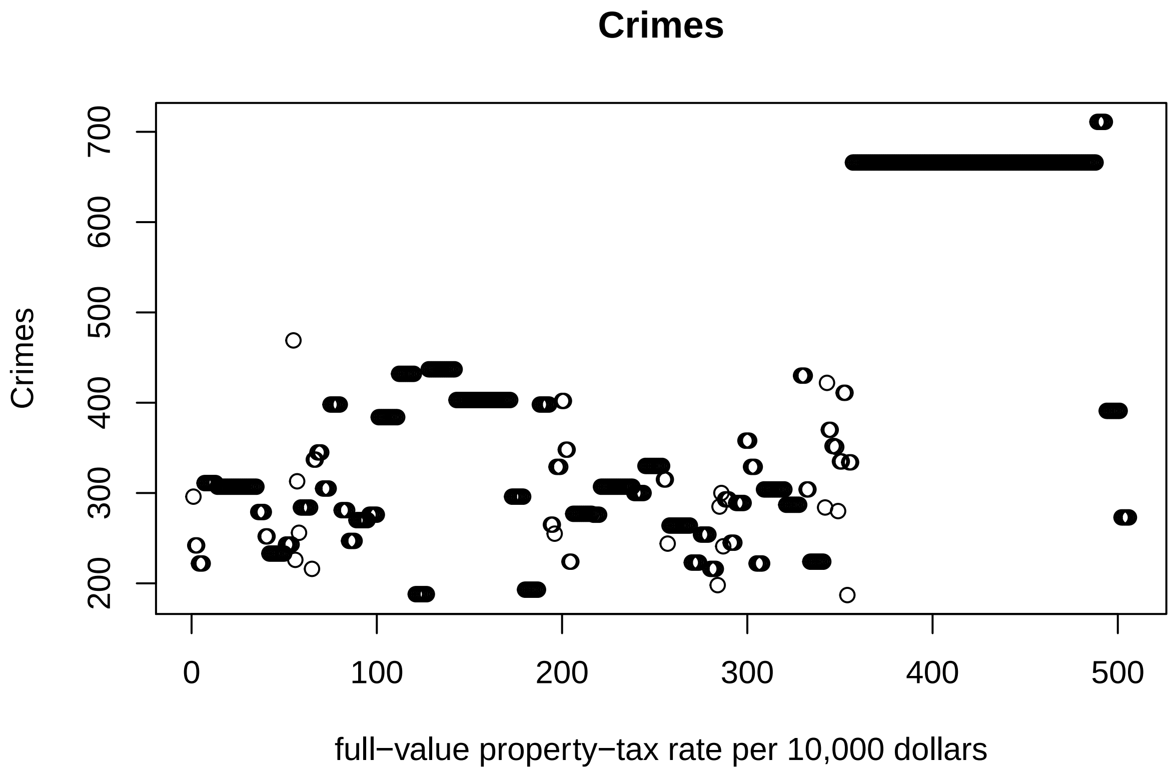
```
abline(fit_rad)
```



```
#Linear fit for predicting the per capita crime rate per town using the full-value property-tax rate pe

fit_tax <- lm(formula = crim ~ tax, data = Boston )
plot( Boston$tax, Boston$crib, main ="Crimes",xlab = "full-value property-tax rate per 10,000 dollars",

summary(fit_tax)
```

```
##
## Call:
## lm(formula = crim ~ tax, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
abline(fit_tax)
```

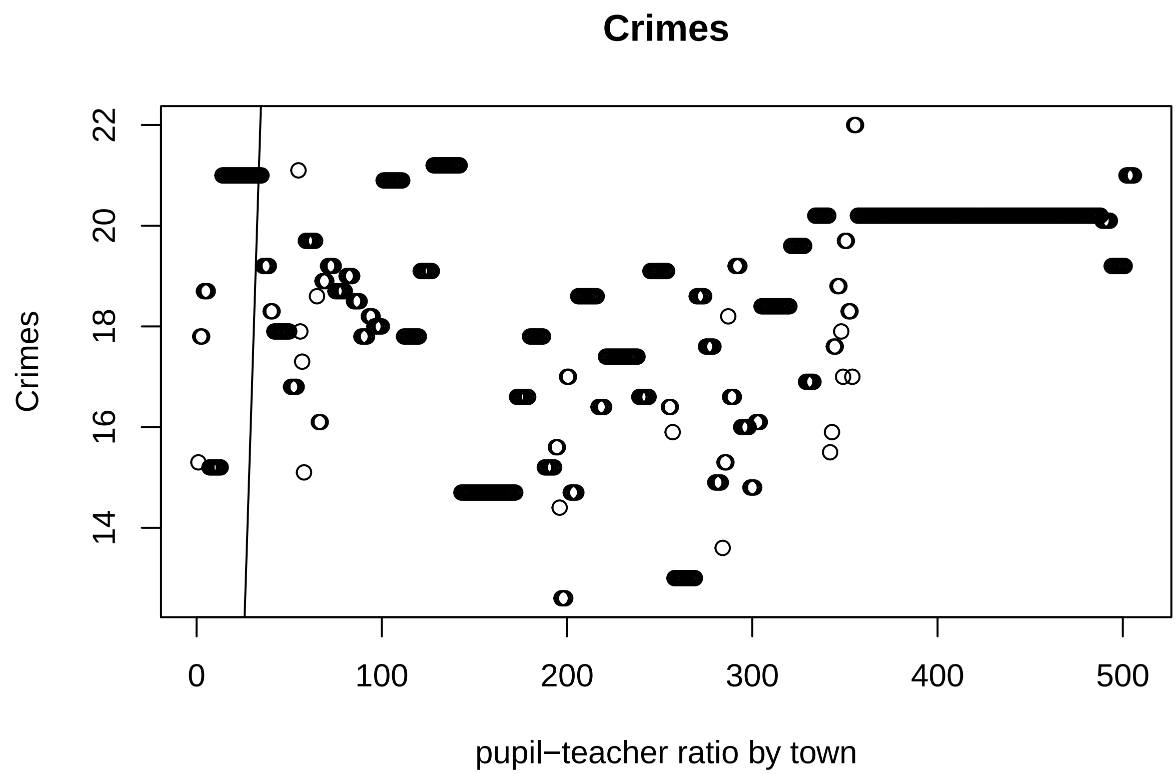


```
#Linear fit for predicting the per capita crime rate per town using pupil-teacher ratio by town

fit_ptratio <- lm(formula = crim ~ ptratio, data = Boston )
plot( Boston$ptratio, Boston$crib, main ="Crimes",xlab = "pupil-teacher ratio by town",ylab ="Crimes")

summary(fit_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

```
abline(fit_ptratio)
```

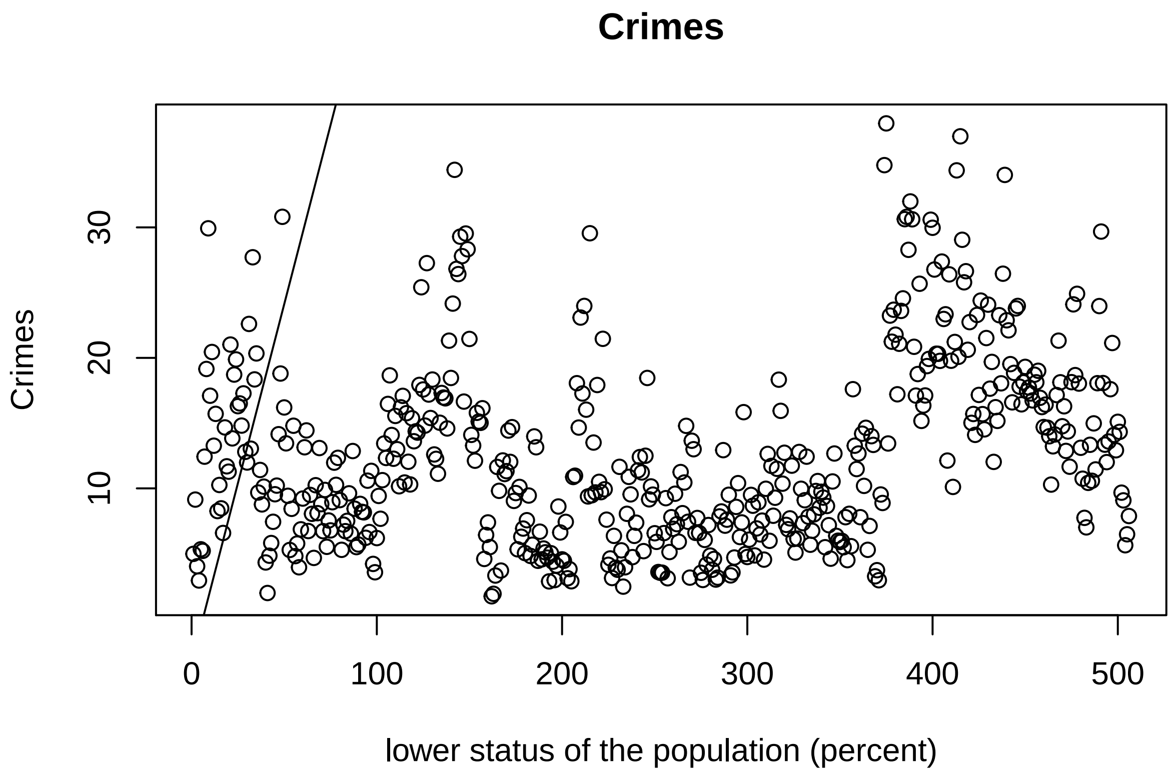


```
#Linear fit for predicting the per capita crime rate per town using the lower status of the population

fit_lstat <- lm(formula = crim ~ lstat, data = Boston )
plot( Boston$lstat, Boston$crib, main ="Crimes",xlab = "lower status of the population (percent)",ylab

summary(fit_lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat, data = Boston)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
abline(fit_lstat)
```



```
#Linear fit for predicting the per capita crime rate per town using the median value of owner-occupied 

fit_medv <- lm(formula = crim ~ medv, data = Boston )
plot( Boston$medv, Boston$crib, main ="Crimes",xlab = "median value of owner-occupied homes in \1000s d

summary(fit_medv)
```

```
##
```

```
## Call:
## lm(formula = crim ~ medv, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
abline(fit_medv)


#Multiple Linear Regression

cor(Boston) #checking for correlation between predictors
```

```
##                crim          zn       indus         chas         nox
## crim     1.00000000 -0.20046922  0.40658341 -0.055891582  0.42097171
## zn      -0.20046922  1.00000000 -0.53382819 -0.042696719 -0.51660371
## indus    0.40658341 -0.53382819  1.00000000  0.062938027  0.76365145
## chas    -0.05589158 -0.04269672  0.06293803  1.000000000  0.09120281
## nox      0.42097171 -0.51660371  0.76365145  0.091202807  1.00000000
## rm      -0.21924670  0.31199059 -0.39167585  0.091251225 -0.30218819
## age      0.35273425 -0.56953734  0.64477851  0.086517774  0.73147010
## dis     -0.37967009  0.66440822 -0.70802699 -0.099175780 -0.76923011
## rad      0.62550515 -0.31194783  0.59512927 -0.007368241  0.61144056
## tax      0.58276431 -0.31456332  0.72076018 -0.035586518  0.66802320
## ptratio  0.28994558 -0.39167855  0.38324756 -0.121515174  0.18893268
## black   -0.38506394  0.17552032 -0.35697654  0.048788485 -0.38005064
## lstat    0.45562148 -0.41299457  0.60379972 -0.053929298  0.59087892
## medv    -0.38830461  0.36044534 -0.48372516  0.175260177 -0.42732077
##                  rm         age         dis          rad         tax
## crim    -0.21924670  0.35273425 -0.37967009  0.625505145  0.58276431
## zn       0.31199059 -0.56953734  0.66440822 -0.311947826 -0.31456332
## indus   -0.39167585  0.64477851 -0.70802699  0.595129275  0.72076018
## chas     0.09125123  0.08651777 -0.09917578 -0.007368241 -0.03558652
## nox     -0.30218819  0.73147010 -0.76923011  0.611440563  0.66802320
## rm       1.00000000 -0.24026493  0.20524621 -0.209846668 -0.29204783
## age     -0.24026493  1.00000000 -0.74788054  0.456022452  0.50645559
## dis      0.20524621 -0.74788054  1.00000000 -0.494587930 -0.53443158
## rad     -0.20984667  0.45602245 -0.49458793  1.000000000  0.91022819
## tax     -0.29204783  0.50645559 -0.53443158  0.910228189  1.00000000
## ptratio -0.35550149  0.26151501 -0.23247054  0.464741179  0.46085304
## black    0.12806864 -0.27353398  0.29151167 -0.444412816 -0.44180801
## lstat   -0.61380827  0.60233853 -0.49699583  0.488676335  0.54399341
## medv     0.69535995 -0.37695457  0.24992873 -0.381626231 -0.46853593
##            ptratio       black      lstat       medv
## crim     0.2899456 -0.38506394  0.4556215 -0.3883046
## zn      -0.3916785  0.17552032 -0.4129946  0.3604453
## indus    0.3832476 -0.35697654  0.6037997 -0.4837252
## chas    -0.1215152  0.04878848 -0.0539293  0.1752602
## nox      0.1889327 -0.38005064  0.5908789 -0.4273208
## rm      -0.3555015  0.12806864 -0.6138083  0.6953599
## age      0.2615150 -0.27353398  0.6023385 -0.3769546
## dis     -0.2324705  0.29151167 -0.4969958  0.2499287
## rad      0.4647412 -0.44441282  0.4886763 -0.3816262
## tax      0.4608530 -0.44180801  0.5439934 -0.4685359
## ptratio  1.0000000 -0.17738330  0.3740443 -0.5077867
## black   -0.1773833  1.00000000 -0.3660869  0.3334608
## lstat    0.3740443 -0.36608690  1.0000000 -0.7376627
## medv    -0.5077867  0.33346082 -0.7376627  1.0000000
```

```r
#Fitting the model using multiple linear regression
multi_fit <- lm(crim ~ rm + zn + indus + chas + nox + age + dis +rad + tax + ptratio +
                  black + lstat + medv,data = Boston)


summary(multi_fit)
```

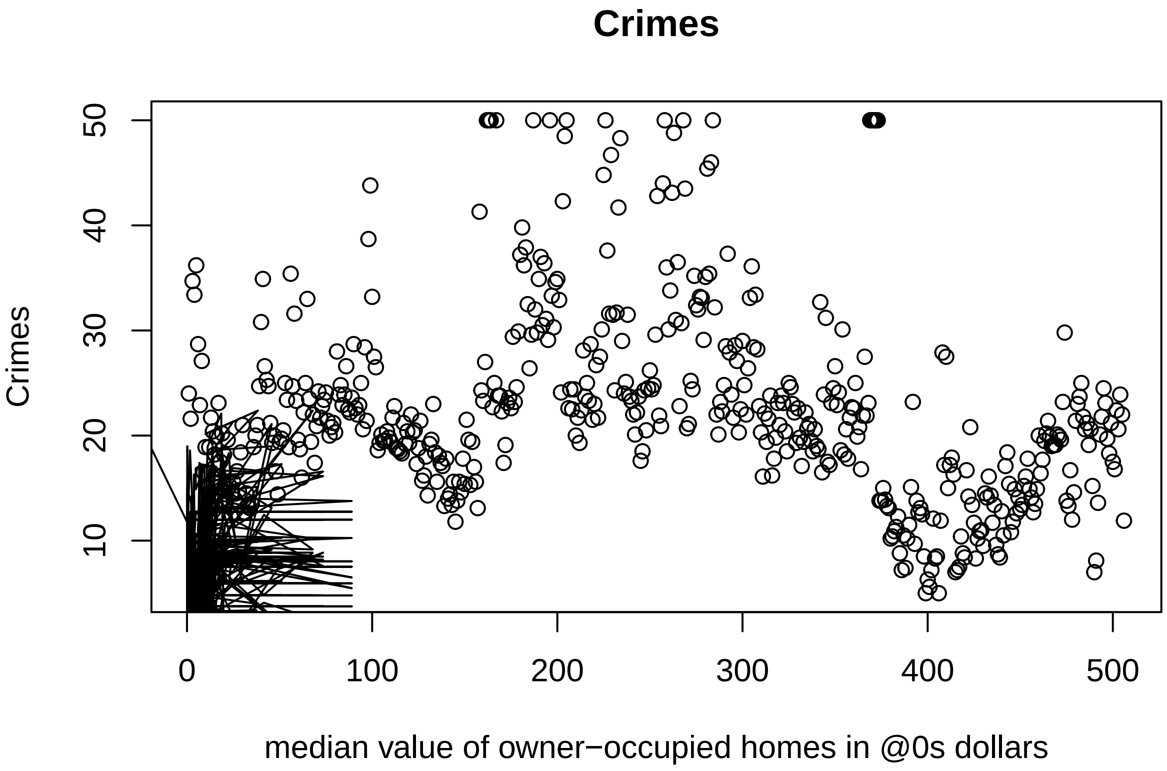
```
##
## Call:
## lm(formula = crim ~ rm + zn + indus + chas + nox + age + dis +
##     rad + tax + ptratio + black + lstat + medv, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## rm            0.430131   0.612830   0.702 0.483089
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
confint(multi_fit) #checking for the confidence interval
```

```
##                     2.5 %        97.5 %
## (Intercept)   2.818109179 31.2483458660
## rm           -0.773956866  1.6342178774
## zn            0.008046562  0.0816638671
## indus        -0.227733150  0.1000235023
## chas         -3.067882868  1.5696156471
## nox         -20.678894713  0.0518248891
## age          -0.033767600  0.0366708869
## dis          -1.540889544 -0.4334619069
## rad           0.415209611  0.7612075719
## tax          -0.013909700  0.0063496670
## ptratio      -0.637417996  0.0952568794
## black        -0.014754837 -0.0003201725
## lstat        -0.022572584  0.2749953365
## medv         -0.317788478 -0.0799851646
```

```
#Non- Linear Regression for each of the predictors to identify the crime rate per town
fit_non_linear <- lm(formula = crim ~ black + I(black ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear))


fit_non_linear_zn <- lm(formula = crim ~ zn + I(zn ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_zn))

fit_non_linear_indus <- lm(formula = crim ~ indus + I(indus ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_indus))

fit_non_linear_chas <- lm(formula = crim ~ chas + I(chas ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_chas))

fit_non_linear_nox <- lm(formula = crim ~ nox + I(nox ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_nox))

fit_non_linear_age <- lm(formula = crim ~ age + I(age ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_age))

fit_non_linear_dis <- lm(formula = crim ~ dis + I(dis ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_dis))

fit_non_linear_rad <- lm(formula = crim ~ rad + I(rad ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_rad))

fit_non_linear_tax <- lm(formula = crim ~ tax + I(tax ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_tax))

fit_non_linear_ptratio <- lm(formula = crim ~ ptratio + I(ptratio ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_ptratio))

fit_non_linear_lstat <- lm(formula = crim ~ lstat + I(lstat ^2), data = Boston)
```

```
lines(Boston$crim,fitted(fit_non_linear_lstat))

fit_non_linear_medv <- lm(formula = crim ~ medv + I(medv ^2), data = Boston)
lines(Boston$crim,fitted(fit_non_linear_medv))
```


Crimes
Crimes
median value of owner−occupied homes in @0s dollars