ARUNA VIJAYAN

1.Download the dataset BOSTON.csv

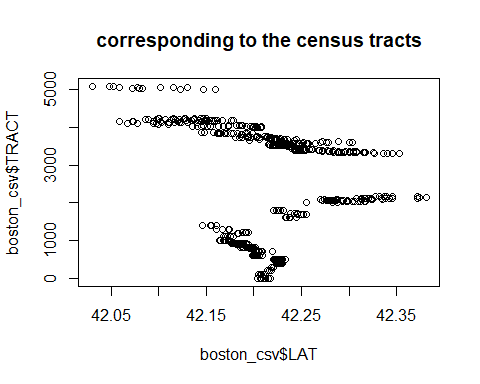
library(readr)  
boston\_csv <- read\_csv("C:/Users/christ/Downloads/boston.csv.txt")

## Parsed with column specification:  
## cols(  
## TOWN = col\_character(),  
## TRACT = col\_double(),  
## LON = col\_double(),  
## LAT = col\_double(),  
## MEDV = col\_double(),  
## CRIM = col\_double(),  
## ZN = col\_double(),  
## INDUS = col\_double(),  
## CHAS = col\_double(),  
## NOX = col\_double(),  
## RM = col\_double(),  
## AGE = col\_double(),  
## DIS = col\_double(),  
## RAD = col\_double(),  
## TAX = col\_double(),  
## PTRATIO = col\_double()  
## )

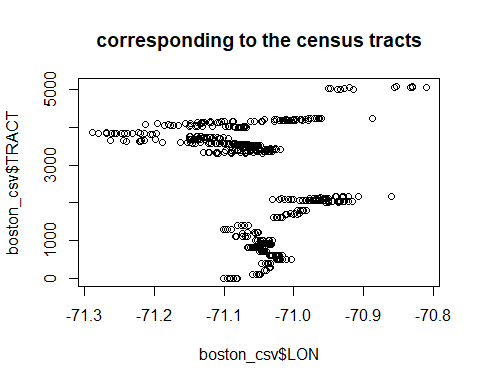
View(boston\_csv)

1. Using the plot commands, plot the latitude and longitude of each of our census tracts

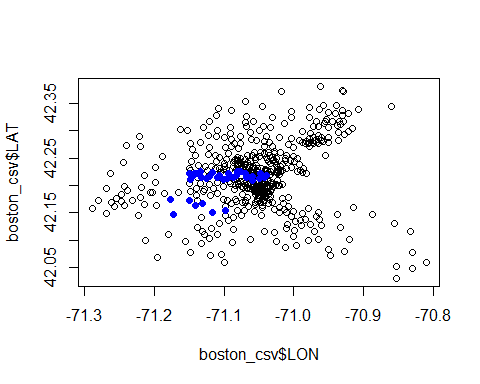
plot(boston\_csv$LAT,boston\_csv$TRACT,main = "corresponding to the census tracts")



plot(boston\_csv$LON,boston\_csv$TRACT,main = "corresponding to the census tracts")

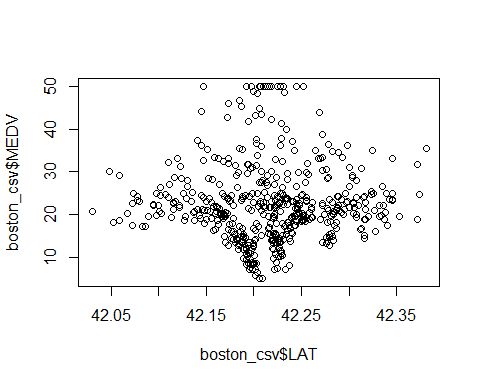
 3. Show all the points that lie along the Charles River in a blue colour.

plot(boston\_csv$LON,boston\_csv$LAT)  
points(boston\_csv$LON[boston\_csv$CHAS==1], boston\_csv$LAT[boston\_csv$CHAS==1],col="blue", pch=19)

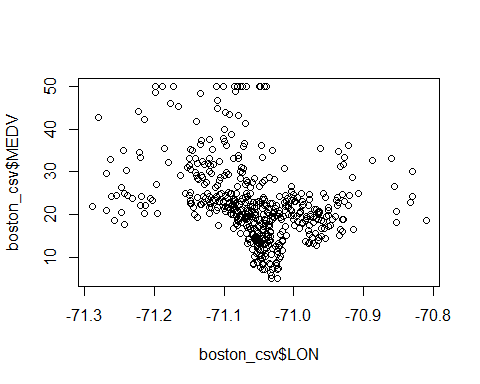


6.Apply Linear Regression by plotting the relationship between latitude and house prices and the longitude and the house prices.

plot(boston\_csv$LAT, boston\_csv$MEDV)



plot(boston\_csv$LON, boston\_csv$MEDV)



linearmodel=lm(MEDV~LON+LAT,data = boston\_csv)  
linearmodel

##   
## Call:  
## lm(formula = MEDV ~ LON + LAT, data = boston\_csv)  
##   
## Coefficients:  
## (Intercept) LON LAT   
## -3178.472 -40.268 8.046

summary(linearmodel)

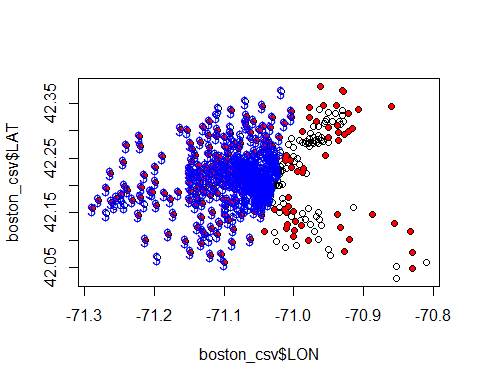
##   
## Call:  
## lm(formula = MEDV ~ LON + LAT, data = boston\_csv)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16.460 -5.590 -1.299 3.695 28.129   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3178.472 484.937 -6.554 1.39e-10 \*\*\*  
## LON -40.268 5.184 -7.768 4.50e-14 \*\*\*  
## LAT 8.046 6.327 1.272 0.204   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 8.693 on 503 degrees of freedom  
## Multiple R-squared: 0.1072, Adjusted R-squared: 0.1036   
## F-statistic: 30.19 on 2 and 503 DF, p-value: 4.159e-13

visualizing the linear regression

plot(boston\_csv$LON, boston\_csv$LAT)  
points(boston\_csv$LON[boston\_csv$MEDV>=21.2], boston\_csv$LAT[boston\_csv$MEDV>=21.2], col="red",pch=20)  
linearmodel$fitted.values

## 1 2 3 4 5 6 7 8   
## 18.75633 18.81648 18.21651 17.97483 17.77344 17.60024 18.32916 18.49416   
## 9 10 11 12 13 14 15 16   
## 18.32904 18.20015 18.44176 18.81222 19.00560 19.43658 19.69836 19.93589   
## 17 18 19 20 21 22 23 24   
## 20.39492 19.92388 20.48766 20.26703 20.08099 20.05277 19.85547 19.67426   
## 25 26 27 28 29 30 31 32   
## 19.54619 19.35208 19.20306 19.16685 19.03801 18.78031 19.43265 19.29173   
## 33 34 35 36 37 38 39 40   
## 19.61388 19.91187 19.79915 20.68910 21.04745 20.98293 21.63527 21.55856   
## 41 42 43 44 45 46 47 48   
## 22.33163 21.37316 21.24036 20.21512 19.75853 19.88344 19.58142 19.25924   
## 49 50 51 52 53 54 55 56   
## 19.25115 19.18508 19.23088 19.74790 19.60931 20.38652 22.21046 20.07213   
## 57 58 59 60 61 62 63 64   
## 18.70708 18.66683 18.82408 18.77184 18.40938 18.33295 18.02687 17.51943   
## 65 66 67 68 69 70 71 72   
## 15.65495 23.10462 24.13141 24.84590 25.60131 25.61752 25.93170 25.34631   
## 73 74 75 76 77 78 79 80   
## 26.20561 25.81913 25.29164 24.85675 24.42188 23.90640 24.53454 24.63514   
## 81 82 83 84 85 86 87 88   
## 23.75323 23.82969 23.85779 23.38269 22.72237 22.94390 22.29957 22.40842   
## 89 90 91 92 93 94 95 96   
## 22.45762 22.14675 21.91325 22.41258 22.92793 23.39494 23.27825 23.80985   
## 97 98 99 100 101 102 103 104   
## 24.17226 24.28506 24.63140 23.89449 23.25432 23.74970 23.81005 23.62081   
## 105 106 107 108 109 110 111 112   
## 23.29062 23.13361 22.97737 22.72687 22.74697 22.97647 22.86770 22.52944   
## 113 114 115 116 117 118 119 120   
## 22.52548 22.31604 22.13247 21.99392 22.06474 21.77238 21.74020 21.37777   
## 121 122 123 124 125 126 127 128   
## 21.49303 21.78858 22.02217 21.96582 21.72822 21.52770 21.78464 22.54575   
## 129 130 131 132 133 134 135 136   
## 22.74307 22.99111 23.22626 23.46786 23.45175 23.66916 23.51215 23.43729   
## 137 138 139 140 141 142 143 144   
## 23.12160 22.92429 22.84380 22.65451 22.46283 22.52568 22.23739 22.40088   
## 145 146 147 148 149 150 151 152   
## 22.34452 22.45326 22.67069 22.53941 22.60221 22.64652 22.81242 22.79553   
## 153 154 155 156 157 158 159 160   
## 22.63851 22.84389 22.92200 22.98483 22.97271 23.13372 23.02100 22.92276   
## 161 162 163 164 165 166 167 168   
## 23.11686 23.31252 23.42771 23.60892 24.03575 23.66526 23.47196 23.89072   
## 169 170 171 172 173 174 175 176   
## 23.37207 23.52911 23.68372 23.62894 23.96395 23.93091 23.86243 24.37376   
## 177 178 179 180 181 182 183 184   
## 25.09862 24.80630 24.38995 24.48101 24.41018 24.24272 24.40379 24.65586   
## 185 186 187 188 189 190 191 192   
## 24.87733 25.13904 24.97145 25.49323 26.23821 26.48860 25.58575 26.72127   
## 193 194 195 196 197 198 199 200   
## 26.15347 27.07142 27.53046 28.15904 29.56023 30.37365 29.78157 30.29335   
## 201 202 203 204 205 206 207 208   
## 30.89753 28.88834 28.88817 28.39678 28.18598 26.42353 26.41397 26.34152   
## 209 210 211 212 213 214 215 216   
## 26.35525 26.05888 26.04683 25.85673 25.70288 25.94843 25.52159 25.36218   
## 217 218 219 220 221 222 223 224   
## 25.00625 24.54316 24.05995 24.54320 24.66565 25.13194 25.26726 25.13756   
## 225 226 227 228 229 230 231 232   
## 24.60370 24.18094 25.03862 24.75439 24.43473 24.76255 25.26424 25.24407   
## 233 234 235 236 237 238 239 240   
## 25.72003 25.55405 25.55558 25.79638 26.19426 26.15406 28.30049 28.21593   
## 241 242 243 244 245 246 247 248   
## 27.75302 28.44966 28.80955 29.24691 29.63356 30.18767 30.42681 29.83324   
## 249 250 251 252 253 254 255 256   
## 30.20688 29.95153 29.63737 29.91678 30.88568 31.20960 31.44166 30.54133   
## 257 258 259 260 261 262 263 264   
## 28.66323 22.91885 23.11536 23.26677 23.37955 23.32319 23.48831 23.12993   
## 265 266 267 268 269 270 271 272   
## 23.00909 22.94468 23.01719 23.48836 23.91107 23.70209 23.15454 23.39217   
## 273 274 275 276 277 278 279 280   
## 23.63371 24.44305 25.11943 25.46811 25.48752 25.92879 25.59288 25.98907   
## 281 282 283 284 285 286 287 288   
## 26.70586 27.45326 27.06033 26.66993 26.65411 27.91862 26.91216 25.10000   
## 289 290 291 292 293 294 295 296   
## 24.60860 25.54682 25.08756 25.00528 23.87704 23.79102 24.24758 24.72601   
## 297 298 299 300 301 302 303 304   
## 24.56256 24.21390 23.93226 23.01827 23.27587 22.80456 21.85417 22.96544   
## 305 306 307 308 309 310 311 312   
## 21.55585 22.03901 21.60809 20.90341 20.32355 20.61345 20.57708 20.25089   
## 313 314 315 316 317 318 319 320   
## 20.46440 20.11653 19.74773 18.96246 19.22830 19.82432 20.12628 20.53701   
## 321 322 323 324 325 326 327 328   
## 19.89691 19.49419 19.21390 18.95456 19.11170 19.37440 19.44197 19.89697   
## 329 330 331 332 333 334 335 336   
## 20.53328 21.11313 20.31192 19.67577 19.07574 18.49177 17.91995 18.23393   
## 337 338 339 340 341 342 343 344   
## 18.58014 17.93585 18.14436 18.36659 18.31661 15.16523 16.38123 17.16657   
## 345 346 347 348 349 350 351 352   
## 16.78823 16.96581 17.17110 16.23690 16.11270 13.72375 12.71715 12.29463   
## 353 354 355 356 357 358 359 360   
## 11.34041 12.06130 13.01563 12.81849 23.60656 24.04794 24.26138 23.92718   
## 361 362 363 364 365 366 367 368   
## 23.79674 23.68957 23.45200 23.72980 22.58058 22.58221 22.34222 22.22785   
## 369 370 371 372 373 374 375 376   
## 22.14326 22.22781 21.94108 21.93382 21.87098 21.65754 21.66964 21.79684   
## 377 378 379 380 381 382 383 384   
## 21.70421 21.86931 21.93617 21.95789 22.21961 22.01023 21.28298 21.20890   
## 385 386 387 388 389 390 391 392   
## 21.30154 21.24358 21.22987 21.19120 21.11629 21.02849 20.80619 20.53234   
## 393 394 395 396 397 398 399 400   
## 21.09620 20.80642 20.92563 21.00858 21.16724 21.15191 21.12448 21.48050   
## 401 402 403 404 405 406 407 408   
## 21.44749 21.34601 21.32429 21.44671 21.51596 21.46034 21.85567 21.91777   
## 409 410 411 412 413 414 415 416   
## 21.92585 22.00638 22.04664 22.12558 22.04264 21.95407 21.80510 21.92029   
## 417 418 419 420 421 422 423 424   
## 22.01853 22.13689 22.23113 22.48482 22.60643 22.75782 22.77960 22.57667   
## 425 426 427 428 429 430 431 432   
## 22.43736 22.42283 22.32218 22.27706 22.17560 22.01454 22.24007 22.06290   
## 433 434 435 436 437 438 439 440   
## 22.10238 21.97033 21.91152 21.87123 21.87201 21.85105 21.86795 21.56755   
## 441 442 443 444 445 446 447 448   
## 21.35816 21.39446 21.40651 21.56114 21.69239 21.74879 21.59015 21.48709   
## 449 450 451 452 453 454 455 456   
## 21.59180 21.72467 21.82537 21.61196 21.32605 21.55157 21.77144 22.00257   
## 457 458 459 460 461 462 463 464   
## 21.99455 21.95834 21.78922 21.60801 21.63859 21.33012 21.12881 21.30601   
## 465 466 467 468 469 470 471 472   
## 21.64832 22.05905 22.08721 22.60660 22.68721 22.68315 22.84829 23.20505   
## 473 474 475 476 477 478 479 480   
## 23.27911 22.94481 22.20387 22.08625 22.30451 22.18770 22.30287 22.77962   
## 481 482 483 484 485 486 487 488   
## 23.97576 23.73415 23.48051 23.73667 22.99336 22.70752 22.64302 22.42955   
## 489 490 491 492 493 494 495 496   
## 21.16374 21.31355 21.40855 21.07754 21.68151 21.00096 21.03154 20.98882   
## 497 498 499 500 501 502 503 504   
## 20.58857 20.31154 20.69332 20.41146 20.10948 19.81316 19.98473 20.12569   
## 505 506   
## 19.81563 19.59015

points(boston\_csv$LON[linearmodel$fitted.values >= 21.2], boston\_csv$LAT[linearmodel$fitted.values >= 21.2], col="blue", pch="$")



5.Apply Regression Tree to the problem and draw conclusions from it.

#install.packages("rpart.plot")

library(rpart.plot)

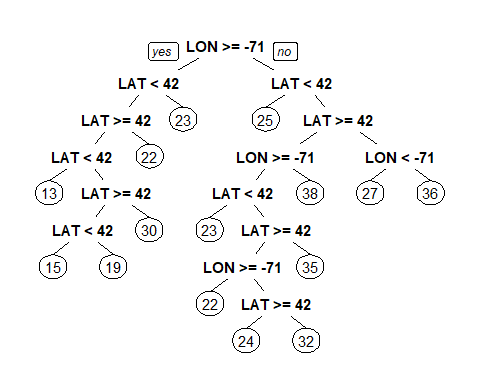
## Loading required package: rpart

latlontree = rpart(MEDV ~ LAT + LON, data=boston\_csv)  
latlontree

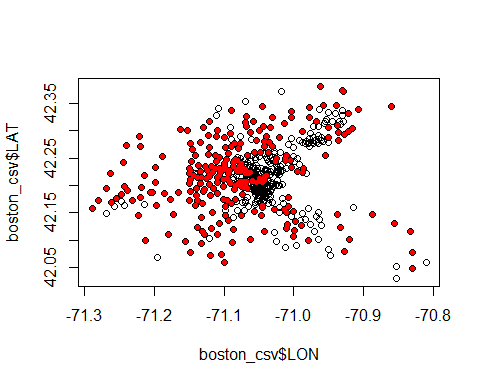
## n= 506   
##   
## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 506 42577.7400 22.52885   
## 2) LON>=-71.0678 309 15730.4600 18.79871   
## 4) LAT< 42.28275 252 12849.1100 17.85040   
## 8) LAT>=42.168 197 10418.1300 16.60457   
## 16) LAT< 42.20585 76 1216.9210 13.31579 \*  
## 17) LAT>=42.20585 121 7862.8730 18.67025   
## 34) LAT>=42.21845 106 2197.7190 17.07075   
## 68) LAT< 42.241 54 1034.6750 14.81667 \*  
## 69) LAT>=42.241 52 603.7531 19.41154 \*  
## 35) LAT< 42.21845 15 3477.5690 29.97333 \*  
## 9) LAT< 42.168 55 1030.0410 22.31273 \*  
## 5) LAT>=42.28275 57 1652.8260 22.99123 \*  
## 3) LON< -71.0678 197 15804.0800 28.37970   
## 6) LAT< 42.1726 46 2027.1460 25.18261 \*  
## 7) LAT>=42.1726 151 13163.5200 29.35364   
## 14) LAT>=42.20785 104 7710.6490 27.28269   
## 28) LON>=-71.17615 96 6248.7900 26.36042   
## 56) LAT< 42.22325 22 306.7382 22.70909 \*  
## 57) LAT>=42.22325 74 5561.5440 27.44595   
## 114) LAT>=42.23065 59 2835.2800 25.60000   
## 228) LON>=-71.0925 30 663.3680 22.38000 \*  
## 229) LON< -71.0925 29 1539.0820 28.93103   
## 458) LAT>=42.283 10 141.9690 23.61000 \*  
## 459) LAT< 42.283 19 964.9611 31.73158 \*  
## 115) LAT< 42.23065 15 1734.4490 34.70667 \*  
## 29) LON< -71.17615 8 400.3200 38.35000 \*  
## 15) LAT< 42.20785 47 4019.8490 33.93617   
## 30) LON< -71.20125 10 328.9840 26.56000 \*  
## 31) LON>=-71.20125 37 2999.7370 35.92973 \*

plottting the tree using the prp command

prp(latlontree)

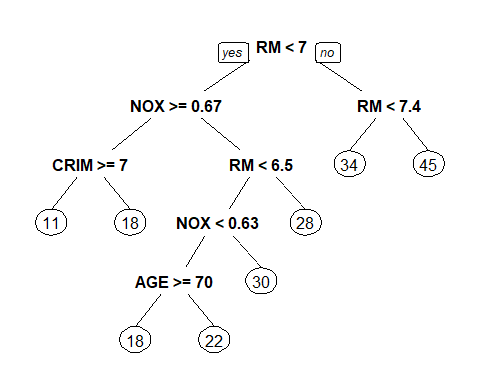
 visualizing the out put

plot(boston\_csv$LON,boston\_csv$LAT)  
points(boston\_csv$LON[boston\_csv$MEDV>=21.2],boston\_csv$LAT[boston\_csv$MEDV>=21.2], col="red", pch=20)



library(caTools)

set.seed(123)  
split = sample.split(boston\_csv$MEDV, SplitRatio = 0.7)  
train = subset(boston\_csv, split==TRUE)  
test = subset(boston\_csv, split==FALSE)  
tree = rpart(MEDV ~ LAT + LON + CRIM + ZN + INDUS + CHAS + NOX + RM + AGE + DIS + RAD + TAX + PTRATIO, data=train)  
prp(tree)



prediction = predict(tree, newdata=test)  
treevisual = sum((prediction - test$MEDV)^2)  
treevisual

## [1] 4328.988

conclusion: The decision tree is working as well but there is a high chance to overfitt.Infact it is the key challenge in case of Decision Trees. If no limit is set, in the worst case, it will end up putting each observation into a leaf node.