HW4

Qiuying Li UNI ql2280

10/3/2017

# Investigate whether there is any multicollinearity, and suggest remedial measures if appropriate.

library("MASS", lib.loc="/Library/Frameworks/R.framework/Versions/3.3/Resources/library")  
data = Boston  
multi\_lm = lm(medv~crim+zn+indus+nox+rm+age+tax,data)  
summary(multi\_lm)

##   
## Call:  
## lm(formula = medv ~ crim + zn + indus + nox + rm + age + tax,   
## data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16.625 -3.161 -0.833 2.089 41.042   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -19.615259 3.221482 -6.089 2.27e-09 \*\*\*  
## crim -0.132538 0.038482 -3.444 0.000621 \*\*\*  
## zn 0.022103 0.014823 1.491 0.136547   
## indus -0.014980 0.072282 -0.207 0.835909   
## nox 0.010643 4.230468 0.003 0.997994   
## rm 7.606508 0.418424 18.179 < 2e-16 \*\*\*  
## age -0.023198 0.014893 -1.558 0.119964   
## tax -0.009006 0.002662 -3.384 0.000772 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.989 on 498 degrees of freedom  
## Multiple R-squared: 0.5818, Adjusted R-squared: 0.576   
## F-statistic: 98.99 on 7 and 498 DF, p-value: < 2.2e-16

anov = anova(multi\_lm)  
ss = anov$`Sum Sq`  
VIF = 1/(1-ss[-length(ss)]/sum(ss))  
anov$VIF = c(VIF," ")  
anov

## Analysis of Variance Table  
##   
## Response: medv  
## Df Sum Sq Mean Sq F value Pr(>F) VIF  
## crim 1 6440.8 6440.8 179.5726 0.00000 1.1776  
## zn 1 3554.3 3554.3 99.0969 0.00000 1.0908  
## indus 1 2551.2 2551.2 71.1299 0.00000 1.0635  
## nox 1 28.7 28.7 0.7991 0.37180 1.0007  
## rm 1 11794.6 11794.6 328.8410 0.00000 1.3814  
## age 1 74.1 74.1 2.0656 0.15128 1.0017  
## tax 1 410.6 410.6 11.4491 0.00077 1.0097  
## Residuals 498 17861.9 35.9

VIF\_bar = mean(VIF);VIF\_bar

## [1] 1.103626

names<-c("crim","zn","indus","nox","rm","age","tax")  
explanatory<-as.matrix(Boston[names])  
dependent<-as.matrix(Boston["medv"])  
corr\_mat<-cor(explanatory)  
eigen\_values<-eigen(corr\_mat)$values  
con\_number<-max(eigen\_values)/min(eigen\_values);con\_number

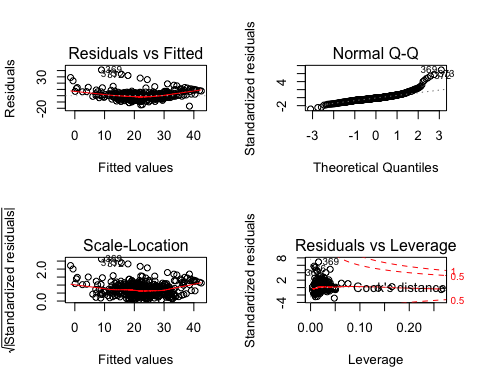
## [1] 19.45283

library(pls)

##   
## Attaching package: 'pls'

## The following object is masked from 'package:stats':  
##   
## loadings

data1 = as.data.frame(Boston)  
lm = lm(medv~crim+zn+indus+nox+rm+age+tax,data=Boston)  
par(mfrow = c(2,2))  
plot(lm)



predict(lm)

## 1 2 3 4 5 6   
## 26.5873967 25.1118284 31.3361260 30.5212170 31.4548628 25.9018013   
## 7 8 9 10 11 12   
## 21.9214974 22.4467438 18.2323121 21.4020907 24.0371861 21.5166845   
## 13 14 15 16 17 18   
## 21.6254315 21.2377088 21.8281862 20.4862233 21.8289308 21.0674680   
## 19 20 21 22 23 24   
## 18.0493650 19.3577111 17.4303152 20.6943296 21.9322652 19.2695260   
## 25 26 27 28 29 30   
## 20.2822828 17.9930645 19.5364968 21.3135942 24.6156270 26.1114848   
## 31 32 33 34 35 36   
## 18.6268720 21.1910641 20.6762229 18.5119444 21.4113257 21.3269091   
## 37 38 39 40 41 42   
## 20.7804465 21.3128364 22.4447485 29.3888872 32.7904767 29.5994968   
## 43 44 45 46 47 48   
## 24.9402897 25.2597157 23.4073218 20.6009716 21.4014022 22.0410373   
## 49 50 51 52 53 54   
## 17.0107307 19.3318617 22.8663428 23.6272202 27.6100339 23.7014381   
## 55 56 57 58 59 60   
## 21.4442163 34.9545028 27.1593752 31.1786835 24.3517380 22.2827389   
## 61 62 63 64 65 66   
## 20.4210661 21.4985798 25.8277784 28.7192477 31.4637818 26.4990012   
## 67 68 69 70 71 72   
## 22.3633417 21.6743351 19.1465474 21.4491714 26.1265347 22.3957659   
## 73 74 75 76 77 78   
## 23.4207584 24.8132957 24.1787871 23.3708115 22.6323781 22.2427699   
## 79 80 81 82 83 84   
## 22.7633622 20.4332836 28.7247692 27.0468010 25.5227622 24.1596571   
## 85 86 87 88 89 90   
## 25.5755544 27.2199239 22.7978233 23.3300960 29.1965657 30.2829002   
## 91 92 93 94 95 96   
## 25.1786172 24.9073921 26.1038051 24.9214502 24.0858381 26.8966317   
## 97 98 99 100 101 102   
## 23.1096174 37.4583880 36.4765693 32.8114323 26.0995089 26.7143137   
## 103 104 105 106 107 108   
## 23.5120085 21.4293415 21.6067663 19.0486088 19.0407478 21.4149732   
## 109 110 111 112 113 114   
## 23.7787939 22.0340935 22.6497915 25.5214111 19.1558797 20.4463511   
## 115 116 117 118 119 120   
## 21.9489999 19.3726390 21.6285299 20.2126510 19.3023652 18.4112316   
## 121 122 123 124 125 126   
## 21.3377340 22.0226036 21.4885538 20.5871527 20.7968173 21.7730302   
## 127 128 129 130 131 132   
## 18.7398406 17.1703594 22.7102236 16.6921000 22.9113405 21.8220795   
## 133 134 135 136 137 138   
## 22.2472454 18.1561512 17.5064682 21.9631222 19.1139598 22.8908872   
## 139 140 141 142 143 144   
## 18.3680305 20.5722416 20.8805567 11.7692711 14.8092966 15.2008234   
## 145 146 147 148 149 150   
## 11.1288223 20.4640408 16.6753008 11.4069768 13.4337324 16.4810792   
## 151 152 153 154 155 156   
## 20.5616140 15.0587500 12.4043785 17.3270028 20.6771905 20.8818568   
## 157 158 159 160 161 162   
## 14.0679191 26.8587983 20.1117695 23.4809981 21.6922270 31.1332778   
## 163 164 165 166 167 168   
## 33.2933568 37.7933535 18.5701916 20.3308181 34.2823967 19.0960275   
## 169 170 171 172 173 174   
## 21.9997744 22.6322272 18.8021435 18.6308905 17.9757318 24.5039725   
## 175 176 177 178 179 180   
## 20.6253640 26.6792438 22.3506555 23.9889069 28.1096430 30.3459349   
## 181 182 183 184 185 186   
## 35.7383450 23.8972735 30.8885803 26.3054551 19.1476148 23.8137405   
## 187 188 189 190 191 192   
## 36.9306592 28.3715889 26.9246583 31.4873910 30.1101865 28.2847251   
## 193 194 195 196 197 198   
## 31.7260494 30.7764243 29.0800788 39.0111647 33.8038222 32.3757878   
## 199 200 201 202 203 204   
## 33.6078284 31.5427032 32.7940400 25.0241213 36.5666182 39.3905806   
## 205 206 207 208 209 210   
## 40.7995053 22.0111937 24.6070862 20.0051989 22.4735096 16.0082890   
## 211 212 213 214 215 216   
## 20.9117871 16.7370786 20.6308079 25.4602212 18.6373546 23.7502453   
## 217 218 219 220 221 222   
## 21.1788644 26.2358613 20.7725666 24.0145183 28.3047831 22.2469305   
## 223 224 225 226 227 228   
## 27.9724669 25.9164291 38.5495954 41.9039074 36.6314468 30.1096374   
## 229 230 231 232 233 234   
## 35.5621241 26.8152418 21.3759631 32.0665478 39.1712242 38.5862280   
## 235 236 237 238 239 240   
## 27.0917681 22.3551604 26.1275392 31.7723001 27.1344368 27.5341240   
## 241 242 243 244 245 246   
## 29.4641704 23.1141637 25.3981166 26.7073003 18.5570861 18.7964660   
## 247 248 249 250 251 252   
## 23.4217438 23.3106319 25.5877012 28.4851188 26.8390899 26.5517024   
## 253 254 255 256 257 258   
## 30.5656749 40.3942292 24.9775933 23.5138210 36.0280995 42.5067357   
## 259 260 261 262 263 264   
## 31.7674324 28.0335461 31.2171435 33.4528964 40.0845479 31.8279438   
## 265 266 267 268 269 270   
## 31.0113035 19.1310522 29.6820092 39.8761593 33.9246421 22.3109913   
## 271 272 273 274 275 276   
## 22.2465786 25.7841481 27.0735932 35.9897440 29.5241775 30.0194945   
## 277 278 279 280 281 282   
## 33.0160089 30.1637288 27.4402461 29.8770199 36.8182290 30.9706186   
## 283 284 285 286 287 288   
## 35.8270380 40.2660602 33.1983089 27.2123897 26.6155004 25.3286091   
## 289 290 291 292 293 294   
## 25.8022160 28.2308239 31.4131502 33.5950471 29.7602876 23.7454205   
## 295 296 297 298 299 300   
## 22.2935101 27.6358075 26.2029496 20.2556526 26.4669732 31.9966012   
## 301 302 303 304 305 306   
## 29.8374093 27.2716667 27.0518092 30.7714444 33.1671353 28.0567835   
## 307 308 309 310 311 312   
## 33.8496024 29.5468105 25.9944231 21.1048859 14.1053892 22.7418413   
## 313 314 315 316 317 318   
## 21.1866314 23.2102978 25.3822362 19.0634528 20.5170479 19.7894467   
## 319 320 321 322 323 324   
## 24.4369602 22.5759369 25.3386729 24.9100975 22.4415162 19.3514292   
## 325 326 327 328 329 330   
## 25.5150217 26.2457314 24.9963540 22.9194103 20.4962350 24.2326215   
## 331 332 333 334 335 336   
## 22.4499122 21.0716738 23.6641619 25.4472616 25.3941260 23.4100454   
## 337 338 339 340 341 342   
## 21.8591245 21.7488731 23.5133025 22.7596140 22.3255440 32.5159318   
## 343 344 345 346 347 348   
## 24.9197465 27.8381766 29.8479652 21.7698871 20.7953390 27.9637628   
## 349 350 351 352 353 354   
## 29.3840605 30.2150939 26.5657227 27.1884067 22.3064629 31.0008992   
## 355 356 357 358 359 360   
## 21.6827067 23.8065316 17.9253710 20.1461831 18.1047417 18.1640068   
## 361 362 363 364 365 366   
## 20.1468505 19.0502306 12.1907587 15.6401303 38.5248123 -1.4328957   
## 367 368 369 370 371 372   
## 9.2640760 -0.6057346 8.9576792 21.9595731 24.3611550 17.8609560   
## 373 374 375 376 377 378   
## 15.6366950 7.6482084 0.8269374 24.8791114 20.5078124 22.2074255   
## 379 380 381 382 383 384   
## 17.2863395 16.7702369 13.2002431 19.5044671 12.6952661 12.7318845   
## 385 386 387 388 389 390   
## 2.5706176 9.7586797 3.9556293 7.0844040 7.0232636 11.7474480   
## 391 392 393 394 395 396   
## 14.4060927 17.5346821 8.6447221 17.9362023 14.9349709 19.8974431   
## 397 398 399 400 401 402   
## 19.8373999 14.5264783 8.1983956 15.5171509 14.0237883 18.1644330   
## 403 404 405 406 407 408   
## 19.2434735 9.2960202 8.7092649 6.0288620 0.5328713 12.8761514   
## 409 410 411 412 413 414   
## 13.5953460 22.0083380 8.8153804 20.5764408 4.5119127 7.2157277   
## 415 416 417 418 419 420   
## 0.1138877 18.3464152 22.1678578 8.9626958 7.3689180 22.6898404   
## 421 422 423 424 425 426   
## 19.0992788 16.6664169 13.4547173 17.6360715 13.6489782 14.6556295   
## 427 428 429 430 431 432   
## 15.5128745 14.4810750 18.4417607 19.1968777 19.2850668 22.5758924   
## 433 434 435 436 437 438   
## 20.4043537 20.2998836 17.2965402 20.8733251 19.1934606 16.5873715   
## 439 440 441 442 443 444   
## 15.4161423 13.5024051 13.3120455 19.3071137 18.3575585 19.8109517   
## 445 446 447 448 449 450   
## 14.7142706 19.6403375 19.2866658 18.1154272 17.6433558 19.6563223   
## 451 452 453 454 455 456   
## 22.4210407 21.7453166 19.2171703 26.9613730 21.8559414 21.1191746   
## 457 458 459 460 461 462   
## 16.9217968 16.3257478 19.0826740 17.5190557 22.3688068 20.0821215   
## 463 464 465 466 467 468   
## 19.3660525 20.8073691 18.7952369 16.3910250 16.9313261 17.0053708   
## 469 470 471 472 473 474   
## 15.4865205 14.5294901 18.5061138 18.8629178 20.8720468 25.0315028   
## 475 476 477 478 479 480   
## 12.1215640 17.8863195 20.6257716 10.2187303 17.5688951 17.5619027   
## 481 482 483 484 485 486   
## 19.3282873 22.9711267 25.2849894 16.6415539 17.4923812 20.4432216   
## 487 488 489 490 491 492   
## 18.0224381 17.1632058 12.8878740 12.4493965 10.0114739 16.7761897   
## 493 494 495 496 497 498   
## 17.1303781 18.8591521 20.7753500 19.1615457 15.9940356 19.1232326   
## 499 500 501 502 503 504   
## 20.9615239 17.3564957 20.6902797 26.2919258 22.5200442 28.6974321   
## 505 506   
## 27.3460122 21.7400634

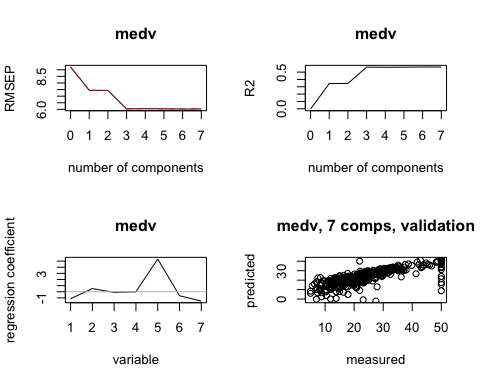
pcr\_model <- pcr(medv~crim+zn+indus+nox+rm+age+tax,data=Boston, scale = TRUE, validation = "CV")  
summary(pcr\_model)

## Data: X dimension: 506 7   
## Y dimension: 506 1  
## Fit method: svdpc  
## Number of components considered: 7  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## CV 9.206 7.445 7.433 6.038 6.052 6.041 6.029  
## adjCV 9.206 7.444 7.438 6.034 6.048 6.037 6.025  
## 7 comps  
## CV 6.032  
## adjCV 6.027  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps  
## X 55.53 69.11 81.10 88.61 94.00 97.15 100.00  
## medv 34.72 34.87 57.42 57.45 57.75 58.03 58.18

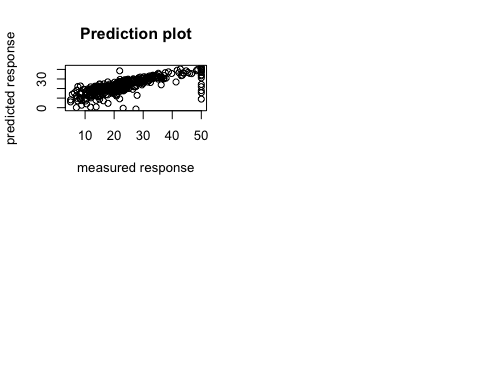
pcr\_pred <- predict(pcr\_model, Boston, ncomp = 3);pcr\_pred

## , , 3 comps  
##   
## medv  
## 1 27.4850989  
## 2 25.7444081  
## 3 31.0813106  
## 4 30.5097427  
## 5 31.6832876  
## 6 26.5384772  
## 7 22.2733316  
## 8 23.7690011  
## 9 19.8793561  
## 10 22.4279159  
## 11 25.2242587  
## 12 22.4367668  
## 13 21.0594954  
## 14 21.2868053  
## 15 22.6159445  
## 16 20.3911333  
## 17 20.7479183  
## 18 21.7936538  
## 19 17.3924781  
## 20 19.7518442  
## 21 18.8706338  
## 22 21.6897937  
## 23 22.9494993  
## 24 20.6930722  
## 25 21.4635617  
## 26 18.9950412  
## 27 20.6250523  
## 28 22.2655157  
## 29 25.6070586  
## 30 26.7910685  
## 31 19.8788569  
## 32 22.5207336  
## 33 21.4217361  
## 34 19.7992880  
## 35 22.6224712  
## 36 21.9573205  
## 37 21.2060895  
## 38 21.0427024  
## 39 21.7397751  
## 40 29.1193067  
## 41 32.1616340  
## 42 27.5147696  
## 43 23.1948603  
## 44 23.4959349  
## 45 22.8598356  
## 46 19.9728511  
## 47 20.7192862  
## 48 23.0902838  
## 49 18.6220551  
## 50 19.7130794  
## 51 22.9081133  
## 52 24.2183679  
## 53 26.6036760  
## 54 22.8853289  
## 55 22.8417474  
## 56 34.7870279  
## 57 28.0589041  
## 58 31.9143211  
## 59 23.9367852  
## 60 22.5712421  
## 61 21.4357974  
## 62 23.3811750  
## 63 26.6479815  
## 64 28.5826533  
## 65 32.1935759  
## 66 26.4843104  
## 67 22.9878588  
## 68 21.1657799  
## 69 19.2729689  
## 70 21.3413916  
## 71 24.0036558  
## 72 20.8115424  
## 73 21.4629785  
## 74 22.7358554  
## 75 21.9899855  
## 76 22.5350072  
## 77 22.8259432  
## 78 21.4860474  
## 79 22.2496794  
## 80 19.4495281  
## 81 28.3576536  
## 82 28.0020865  
## 83 25.2593633  
## 84 24.4483583  
## 85 25.6000245  
## 86 27.4418637  
## 87 22.8523883  
## 88 23.7546551  
## 89 30.4663544  
## 90 30.7198365  
## 91 25.9520310  
## 92 25.9566131  
## 93 24.8803311  
## 94 22.9191688  
## 95 23.7550480  
## 96 27.5163204  
## 97 24.3020545  
## 98 38.2056491  
## 99 35.9502482  
## 100 33.3180012  
## 101 26.7175533  
## 102 27.0143386  
## 103 24.4336807  
## 104 22.5147010  
## 105 22.7727377  
## 106 20.5586210  
## 107 20.3885882  
## 108 22.4279334  
## 109 25.0844499  
## 110 23.2190586  
## 111 22.5668925  
## 112 26.0548544  
## 113 20.3635622  
## 114 21.6774971  
## 115 22.7341534  
## 116 20.4110374  
## 117 22.0338014  
## 118 21.0236818  
## 119 19.8350647  
## 120 18.7182024  
## 121 18.5839402  
## 122 19.7231301  
## 123 19.5103182  
## 124 18.7879562  
## 125 18.9482269  
## 126 19.6287307  
## 127 16.9750472  
## 128 16.5799582  
## 129 21.9581138  
## 130 16.0707775  
## 131 22.1531259  
## 132 21.0610955  
## 133 21.4822300  
## 134 17.4990556  
## 135 16.9710930  
## 136 21.2217911  
## 137 18.3487294  
## 138 22.1165637  
## 139 17.7968254  
## 140 19.8850592  
## 141 20.0377767  
## 142 11.5426688  
## 143 14.0531626  
## 144 14.4153103  
## 145 10.4759201  
## 146 19.4611482  
## 147 15.8502330  
## 148 10.6764038  
## 149 12.5462226  
## 150 15.4844085  
## 151 19.4736759  
## 152 14.3177803  
## 153 11.3861717  
## 154 16.4213809  
## 155 19.5436070  
## 156 19.2556899  
## 157 13.1562508  
## 158 26.1700442  
## 159 19.8198448  
## 160 22.3530545  
## 161 21.0787993  
## 162 30.0213851  
## 163 32.3262419  
## 164 36.4784220  
## 165 18.0594477  
## 166 19.7694965  
## 167 33.1996516  
## 168 18.1423440  
## 169 21.4752792  
## 170 22.0460467  
## 171 18.3903152  
## 172 18.3018962  
## 173 19.7496388  
## 174 25.8293405  
## 175 21.6098724  
## 176 26.1838281  
## 177 22.5303949  
## 178 24.9774934  
## 179 28.9419520  
## 180 30.5753626  
## 181 36.5594463  
## 182 24.5518576  
## 183 32.2330644  
## 184 27.9756819  
## 185 20.9521395  
## 186 24.6950052  
## 187 36.6947824  
## 188 28.8819733  
## 189 27.0960900  
## 190 31.7799208  
## 191 29.8789198  
## 192 28.4516907  
## 193 31.5823688  
## 194 30.0531090  
## 195 28.7353440  
## 196 39.0754594  
## 197 34.2487170  
## 198 32.9707365  
## 199 34.2035123  
## 200 31.7436815  
## 201 32.8903279  
## 202 25.9760965  
## 203 36.2210275  
## 204 39.1700290  
## 205 40.4703964  
## 206 20.3760872  
## 207 23.8700874  
## 208 20.1615102  
## 209 22.0588852  
## 210 17.2672619  
## 211 21.6820947  
## 212 17.5786718  
## 213 20.1210470  
## 214 24.0035955  
## 215 16.7336467  
## 216 22.7123714  
## 217 20.0600232  
## 218 25.8656301  
## 219 20.9470099  
## 220 23.9923105  
## 221 29.3076720  
## 222 23.6226642  
## 223 28.6223541  
## 224 26.7657738  
## 225 38.7446050  
## 226 42.0998877  
## 227 37.1905495  
## 228 30.7460506  
## 229 33.8265205  
## 230 25.6289848  
## 231 22.0147414  
## 232 32.5108352  
## 233 39.1574138  
## 234 38.5050732  
## 235 27.4068643  
## 236 22.7214702  
## 237 26.8234345  
## 238 32.0429227  
## 239 26.3966938  
## 240 27.5775490  
## 241 29.8266746  
## 242 24.1337527  
## 243 25.9008555  
## 244 25.6275172  
## 245 20.0462827  
## 246 20.0622636  
## 247 23.2811267  
## 248 24.6720786  
## 249 25.8290232  
## 250 27.5262659  
## 251 25.8049595  
## 252 25.3913843  
## 253 29.1514965  
## 254 38.5770479  
## 255 25.4310515  
## 256 23.5996057  
## 257 35.9092724  
## 258 42.7954740  
## 259 32.9922185  
## 260 29.4304536  
## 261 31.8549464  
## 262 34.2442382  
## 263 40.6414339  
## 264 32.8619880  
## 265 31.9891335  
## 266 19.6812959  
## 267 30.4814312  
## 268 39.7990226  
## 269 33.6363351  
## 270 22.5903657  
## 271 21.8712440  
## 272 24.3772478  
## 273 27.0387205  
## 274 35.3097020  
## 275 28.8354841  
## 276 29.6414672  
## 277 32.7090168  
## 278 29.2666582  
## 279 26.8202240  
## 280 29.4061790  
## 281 37.1200812  
## 282 30.6206465  
## 283 35.6747732  
## 284 39.8881333  
## 285 32.9639576  
## 286 27.5867120  
## 287 27.0954922  
## 288 25.2279256  
## 289 26.1621451  
## 290 27.7129240  
## 291 31.0085346  
## 292 33.0825217  
## 293 29.2799561  
## 294 21.5557597  
## 295 20.9772266  
## 296 25.6947805  
## 297 25.0005414  
## 298 19.5621695  
## 299 26.6884894  
## 300 31.6226853  
## 301 30.8251546  
## 302 27.1754730  
## 303 26.2225431  
## 304 29.7470857  
## 305 33.0435889  
## 306 28.7423230  
## 307 34.7336343  
## 308 30.5754587  
## 309 26.2341701  
## 310 21.3762785  
## 311 13.3532191  
## 312 22.1248987  
## 313 21.9178299  
## 314 23.5916998  
## 315 25.8139655  
## 316 19.4640387  
## 317 21.0353121  
## 318 19.9542447  
## 319 24.2335080  
## 320 22.1737014  
## 321 25.0655054  
## 322 24.7239579  
## 323 22.2181550  
## 324 20.0947609  
## 325 24.8195205  
## 326 24.6616528  
## 327 23.9473666  
## 328 22.4664292  
## 329 20.6263602  
## 330 23.9003976  
## 331 22.7062943  
## 332 20.7915312  
## 333 23.0927165  
## 334 24.8107506  
## 335 24.7737571  
## 336 22.7460756  
## 337 21.6648614  
## 338 22.0085388  
## 339 22.9391541  
## 340 22.4931932  
## 341 22.5210697  
## 342 33.0446476  
## 343 26.0278083  
## 344 28.7101408  
## 345 29.6723134  
## 346 22.2832243  
## 347 21.4813551  
## 348 28.0840892  
## 349 29.6078317  
## 350 30.5833389  
## 351 27.4356755  
## 352 28.0361386  
## 353 22.7920677  
## 354 31.2572484  
## 355 22.2044976  
## 356 24.1485633  
## 357 17.9979662  
## 358 19.9754935  
## 359 17.7518970  
## 360 17.7513085  
## 361 19.8647864  
## 362 18.9335941  
## 363 12.5645509  
## 364 15.6041038  
## 365 37.3724269  
## 366 -0.5914994  
## 367 9.7433108  
## 368 0.6971441  
## 369 9.9459522  
## 370 22.2295906  
## 371 24.5314600  
## 372 18.3757380  
## 373 15.8225810  
## 374 8.5115616  
## 375 1.8965902  
## 376 24.7459184  
## 377 20.4839095  
## 378 22.3706404  
## 379 17.3865779  
## 380 17.1069380  
## 381 12.3892861  
## 382 19.7139365  
## 383 13.2724080  
## 384 13.3247898  
## 385 3.1580574  
## 386 10.2956110  
## 387 4.7132604  
## 388 7.3698284  
## 389 7.7865309  
## 390 12.3462619  
## 391 14.8356868  
## 392 17.3552386  
## 393 9.2723150  
## 394 18.0479941  
## 395 15.1870127  
## 396 20.1270401  
## 397 20.0168372  
## 398 15.0221536  
## 399 8.5745399  
## 400 15.2224014  
## 401 14.3259315  
## 402 18.4337320  
## 403 19.5308514  
## 404 9.6845679  
## 405 8.5227962  
## 406 6.0729984  
## 407 1.6066640  
## 408 13.5093307  
## 409 14.3493600  
## 410 22.3428628  
## 411 9.2216129  
## 412 20.9825958  
## 413 5.5886243  
## 414 8.0240559  
## 415 0.7544678  
## 416 18.5868790  
## 417 22.0277409  
## 418 9.1528980  
## 419 7.3050839  
## 420 21.9292100  
## 421 19.3076366  
## 422 16.8876421  
## 423 13.7563693  
## 424 17.7337378  
## 425 13.4922480  
## 426 14.9434010  
## 427 14.8519927  
## 428 13.8948773  
## 429 18.0954085  
## 430 19.3774494  
## 431 19.3960922  
## 432 22.7890703  
## 433 20.1124273  
## 434 20.1378424  
## 435 17.3907222  
## 436 20.7604616  
## 437 19.0664923  
## 438 16.7955003  
## 439 15.2918292  
## 440 13.7313550  
## 441 13.3141990  
## 442 19.3751210  
## 443 18.6230734  
## 444 19.9466892  
## 445 14.9286584  
## 446 19.5981646  
## 447 19.3788165  
## 448 18.2151580  
## 449 17.9133820  
## 450 19.8464534  
## 451 22.3032684  
## 452 21.8662895  
## 453 19.2437822  
## 454 26.8381536  
## 455 21.7739751  
## 456 20.8842530  
## 457 16.9287114  
## 458 16.0520758  
## 459 18.8032684  
## 460 17.3492007  
## 461 22.1935448  
## 462 19.9745648  
## 463 19.0660009  
## 464 20.6859283  
## 465 18.0584719  
## 466 15.2529341  
## 467 16.9910621  
## 468 17.5643252  
## 469 15.1697532  
## 470 13.8108066  
## 471 18.6529113  
## 472 19.3465292  
## 473 20.6175468  
## 474 24.2330542  
## 475 12.8828061  
## 476 18.4737544  
## 477 20.9043562  
## 478 10.9533171  
## 479 18.0145920  
## 480 17.6544452  
## 481 18.8870342  
## 482 22.7079263  
## 483 24.9857150  
## 484 15.5446376  
## 485 16.2863834  
## 486 19.4197967  
## 487 18.0225113  
## 488 16.3177841  
## 489 12.2200555  
## 490 11.9902694  
## 491 9.6541875  
## 492 16.1357320  
## 493 15.9572406  
## 494 18.6159493  
## 495 20.0576353  
## 496 18.0539839  
## 497 16.5188893  
## 498 19.4266224  
## 499 21.0018082  
## 500 17.8404595  
## 501 21.2291869  
## 502 25.6113234  
## 503 22.2699202  
## 504 28.6450247  
## 505 27.2977872  
## 506 21.6641918

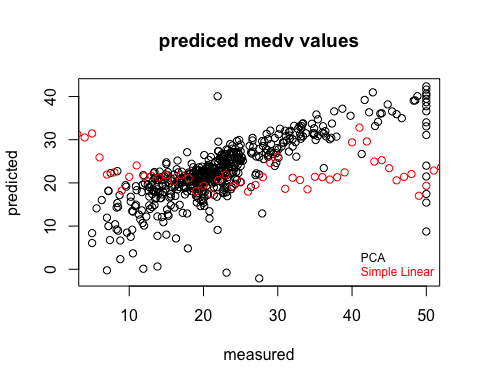
validationplot(pcr\_model)  
validationplot(pcr\_model, val.type = "R2")  
coefplot(pcr\_model)  
predplot(pcr\_model)



predplot(lm)  
  
#predplot(pcr\_model,main = "predictoin values of two models")  
par(mfrow = c(1,1))



predplot(pcr\_model,main = "prediced medv values")  
points(predict(lm), col = "red")  
legend('bottomright',c("PCA", "Simple Linear"), text.col = c("black", "red"),bty='n', cex=.75)



#ridge regression  
  
ridge\_reg<-lm.ridge(medv~crim+zn+indus+nox+rm+age+tax,data=Boston)  
ridge\_coef<-ridge\_reg$coef;ridge\_coef

## crim zn indus nox rm age   
## -1.13890869 0.51499393 -0.10266382 0.00123203 5.33917934 -0.65234909   
## tax   
## -1.51634546

#Lasso regression  
library(glmnet)

## Loading required package: Matrix

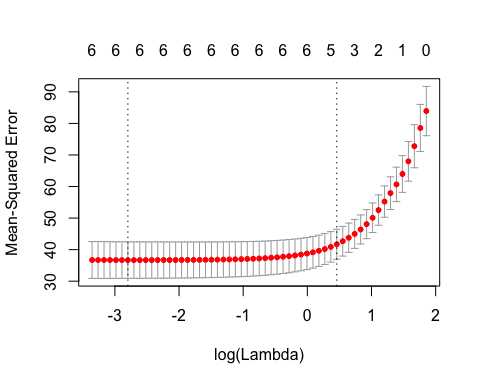
## Loading required package: foreach

## Loaded glmnet 2.0-5

names<-c("crim","zn","indus","nox","rm","age","tax")  
explanatory<-as.matrix(Boston[names])  
dependent<-as.matrix(Boston["medv"])  
lasso\_reg<-glmnet(explanatory,dependent)  
summary(lasso\_reg)

## Length Class Mode   
## a0 58 -none- numeric  
## beta 406 dgCMatrix S4   
## df 58 -none- numeric  
## dim 2 -none- numeric  
## lambda 58 -none- numeric  
## dev.ratio 58 -none- numeric  
## nulldev 1 -none- numeric  
## npasses 1 -none- numeric  
## jerr 1 -none- numeric  
## offset 1 -none- logical  
## call 3 -none- call   
## nobs 1 -none- numeric

cv\_reg<-cv.glmnet(explanatory,dependent)  
plot(cv\_reg)



#choose model coefficient  
lambda<-cv\_reg$lambda.min  
coeff<-coef(cv\_reg,s="lambda.min")  
predict\_lasso<-predict(cv\_reg,newx = explanatory,s="lambda.min")  
MSE\_lasso<-mean((dependent-predict\_lasso)^2);MSE\_lasso

## [1] 35.3082

#stepwise regression  
glm\_model1<-glm(medv~1,data=Boston)  
glm\_model2<-glm(medv~crim+zn+indus+nox+rm+age+tax,data=Boston)  
  
backward<-stepAIC(glm\_model2,direction = "backward",scope=list(  
 upper = glm\_model2,lower = glm\_model1),trace = F)  
summary(backward)

##   
## Call:  
## glm(formula = medv ~ crim + zn + rm + age + tax, data = Boston)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -16.669 -3.167 -0.808 2.075 41.083   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -19.713176 2.862677 -6.886 1.73e-11 \*\*\*  
## crim -0.131852 0.038261 -3.446 0.000617 \*\*\*  
## zn 0.022947 0.014231 1.612 0.107487   
## rm 7.625253 0.408770 18.654 < 2e-16 \*\*\*  
## age -0.024121 0.012709 -1.898 0.058271 .   
## tax -0.009323 0.002139 -4.358 1.59e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 35.72726)  
##   
## Null deviance: 42716 on 505 degrees of freedom  
## Residual deviance: 17864 on 500 degrees of freedom  
## AIC: 3253.3  
##   
## Number of Fisher Scoring iterations: 2

forward<-stepAIC(glm\_model1,direction = "forward",scope=list(  
 upper = glm\_model2,lower = glm\_model1),trace = F)  
summary(forward)

##   
## Call:  
## glm(formula = medv ~ rm + tax + crim + age + zn, data = Boston)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -16.669 -3.167 -0.808 2.075 41.083   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -19.713176 2.862677 -6.886 1.73e-11 \*\*\*  
## rm 7.625253 0.408770 18.654 < 2e-16 \*\*\*  
## tax -0.009323 0.002139 -4.358 1.59e-05 \*\*\*  
## crim -0.131852 0.038261 -3.446 0.000617 \*\*\*  
## age -0.024121 0.012709 -1.898 0.058271 .   
## zn 0.022947 0.014231 1.612 0.107487   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 35.72726)  
##   
## Null deviance: 42716 on 505 degrees of freedom  
## Residual deviance: 17864 on 500 degrees of freedom  
## AIC: 3253.3  
##   
## Number of Fisher Scoring iterations: 2

#calculate MSE  
predict\_back<-predict(backward,newx = explanatory)  
predict\_for<-predict(forward,newx = explanatory)  
MSE\_back<-mean((dependent-predict\_back)^2);MSE\_back

## [1] 35.30361

MSE\_for<-mean((dependent-predict\_for)^2);MSE\_for

## [1] 35.30361

#comparison plot  
par(mfrow = c(1,3))  
plot(predict\_lasso)  
plot(predict\_back)  
plot(predict\_for)

