##### Chapter 6: Regression Methods ——————-

#### Part 1: Linear Regression ——————-

## Example: Predicting Medical Expenses —-

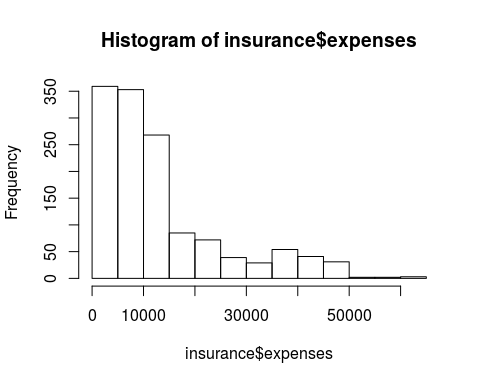
## Step 2: Exploring and preparing the data ----  
insurance <- read.csv("insurance.csv", stringsAsFactors = TRUE)  
str(insurance)

## 'data.frame': 1338 obs. of 7 variables:  
## $ age : int 19 18 28 33 32 31 46 37 37 60 ...  
## $ sex : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...  
## $ bmi : num 27.9 33.8 33 22.7 28.9 25.7 33.4 27.7 29.8 25.8 ...  
## $ children: int 0 1 3 0 0 0 1 3 2 0 ...  
## $ smoker : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...  
## $ region : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...  
## $ expenses: num 16885 1726 4449 21984 3867 ...

# summarize the charges variable  
summary(insurance$expenses)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1122 4740 9382 13270 16640 63770

# histogram of insurance charges  
hist(insurance$expenses)



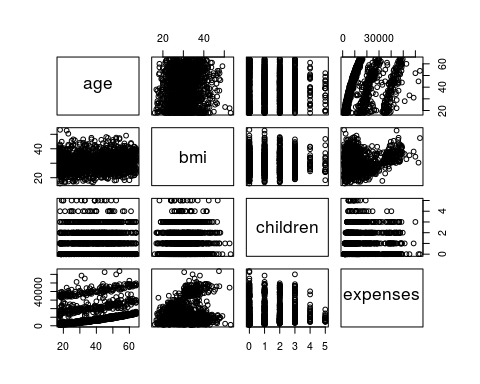
# table of region  
table(insurance$region)

##   
## northeast northwest southeast southwest   
## 324 325 364 325

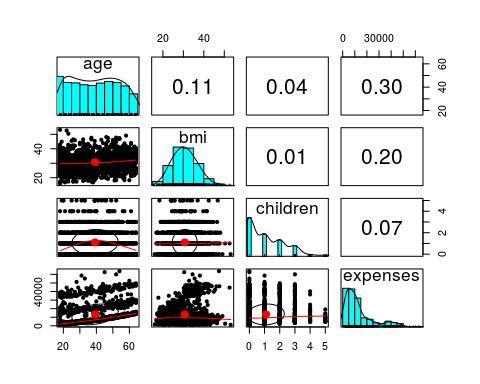
# exploring relationships among features: correlation matrix  
cor(insurance[c("age", "bmi", "children", "expenses")])

## age bmi children expenses  
## age 1.0000000 0.10934101 0.04246900 0.29900819  
## bmi 0.1093410 1.00000000 0.01264471 0.19857626  
## children 0.0424690 0.01264471 1.00000000 0.06799823  
## expenses 0.2990082 0.19857626 0.06799823 1.00000000

# visualing relationships among features: scatterplot matrix  
pairs(insurance[c("age", "bmi", "children", "expenses")])  
  
# more informative scatterplot matrix  
library(psych)



pairs.panels(insurance[c("age", "bmi", "children", "expenses")])



# set up trainning and test data sets  
  
indx <- sample(1:nrow(insurance), as.integer(0.9\*nrow(insurance)))  
indx

## [1] 322 1142 615 1012 962 397 1212 683 71 552 1230 551 724  
## [14] 326 106 560 224 932 74 138 1210 184 1307 391 1239 1100  
## [27] 197 1093 298 427 607 1092 550 604 847 530 914 489 1033  
## [40] 1299 228 311 780 280 832 881 1052 800 64 1061 616 1035  
## [53] 1113 866 766 1257 387 1081 416 937 869 916 287 547 1261  
## [66] 838 309 1328 764 589 202 430 401 1169 1292 460 1184 196  
## [79] 403 741 626 639 1130 394 1236 698 1077 27 999 60 426  
## [92] 1110 297 160 726 459 966 613 1197 1057 471 614 1034 139  
## [105] 781 163 266 627 1332 1145 87 1006 1005 1161 1022 572 512  
## [118] 2 187 1287 749 975 236 339 564 1075 259 1286 362 422  
## [131] 770 1265 132 752 3 740 1248 219 549 16 767 439 1201  
## [144] 980 712 562 380 958 1159 1044 302 888 856 44 158 476  
## [157] 811 1050 1290 319 623 907 535 659 301 243 1187 644 263  
## [170] 678 103 206 986 650 1223 580 666 217 709 776 913 865  
## [183] 834 696 983 1252 43 1107 1138 215 1124 12 205 1134 1141  
## [196] 78 115 773 689 921 601 82 130 411 1191 688 31 829  
## [209] 567 1213 19 1266 442 723 413 787 969 561 1171 291 978  
## [222] 1089 1317 1281 1238 685 38 1082 188 1249 920 1200 107 264  
## [235] 177 516 451 984 1115 1293 797 718 356 199 1073 176 875  
## [248] 656 1306 968 254 10 756 501 612 1104 509 33 1211 1259  
## [261] 521 1181 85 76 1097 716 1243 587 200 370 1160 1101 286  
## [274] 602 972 1029 926 495 514 406 997 124 1027 1047 556 824  
## [287] 1042 428 582 679 360 24 152 317 610 1016 1133 491 1037  
## [300] 314 153 144 372 1335 541 924 874 792 478 746 425 1180  
## [313] 814 412 453 996 815 594 653 334 789 893 456 532 1178  
## [326] 1327 998 374 655 861 316 649 889 533 915 447 603 42  
## [339] 149 1 361 1310 664 1329 1085 273 1011 729 1246 400 1111  
## [352] 692 525 96 835 221 118 111 609 804 57 93 1219 61  
## [365] 667 885 270 1194 899 1275 238 15 785 1325 934 796 1108  
## [378] 1167 332 578 240 1140 65 414 1036 183 727 1154 1291 851  
## [391] 1324 307 1024 720 600 750 373 1336 354 643 977 385 48  
## [404] 905 522 761 275 34 544 70 919 1054 77 369 531 1226  
## [417] 497 128 104 1068 285 1096 1285 156 167 557 662 218 161  
## [430] 630 1220 1019 923 631 306 418 771 429 1224 554 192 598  
## [443] 812 136 953 819 850 506 1053 108 1202 137 431 424 39  
## [456] 1198 1330 682 897 1098 169 213 350 437 474 738 323 658  
## [469] 1010 1245 995 563 882 409 1227 1225 475 695 485 1267 755  
## [482] 1043 722 378 1144 446 1158 28 1297 279 1095 399 959 1125  
## [495] 338 1086 1196 1122 1309 1094 1186 1269 342 1102 1174 822 640  
## [508] 462 69 191 721 1099 642 405 247 204 982 591 14 1301  
## [521] 282 680 159 538 1203 415 1013 1302 559 245 961 708 952  
## [534] 269 1137 142 858 808 1170 232 668 848 367 292 276 493  
## [547] 393 1106 1188 1079 711 635 452 943 1129 1067 520 629 420  
## [560] 271 49 1164 769 195 1166 585 310 759 565 841 36 1277  
## [573] 165 931 1031 220 468 1240 903 260 946 148 194 1313 955  
## [586] 436 1126 537 321 1204 402 333 1274 17 929 1303 146 745  
## [599] 621 329 579 895 1074 164 810 901 900 985 928 500 948  
## [612] 122 129 503 917 671 1320 1233 1193 234 744 681 799 715  
## [625] 1255 274 783 382 472 343 45 801 171 507 376 660 702  
## [638] 669 1218 502 1337 1244 229 1305 798 1264 448 973 855 150  
## [651] 1117 788 490 151 872 80 827 384 918 735 663 272 852  
## [664] 62 687 84 1023 390 381 793 768 1315 133 355 120 1007  
## [677] 902 1175 388 300 9 714 651 964 242 896 754 945 1083  
## [690] 1179 320 1002 707 699 1258 1298 227 4 249 435 278 732  
## [703] 1251 620 844 226 546 1263 864 1039 444 967 870 359 248  
## [716] 878 1268 288 849 892 622 225 839 337 1283 701 806 590  
## [729] 331 1114 940 454 283 1105 51 1146 933 1119 1206 68 910  
## [742] 496 469 1028 725 119 862 1030 592 1153 1278 857 778 1018  
## [755] 545 67 1183 734 1116 214 231 1071 7 180 597 351 348  
## [768] 1294 1295 313 636 760 782 665 576 32 449 1065 772 1131  
## [781] 358 1222 265 1331 1151 684 1189 1064 419 1232 293 20 168  
## [794] 92 1048 845 1128 596 443 1260 81 5 1237 1147 1072 1066  
## [807] 377 346 433 190 938 817 654 854 941 303 1056 233 645  
## [820] 951 398 719 1026 879 676 989 363 510 499 748 661 8  
## [833] 504 100 1334 466 1132 595 193 853 1020 56 483 786 577  
## [846] 37 634 1192 1088 911 41 739 779 268 58 859 960 694  
## [859] 463 389 949 1155 253 1163 909 487 258 255 257 775 1205  
## [872] 568 1304 543 473 117 908 818 1059 344 1136 1209 965 987  
## [885] 505 957 1338 289 618 1195 134 1271 252 599 1176 345 677  
## [898] 686 833 127 588 94 109 212 528 569 1055 185 113 1084  
## [911] 693 178 251 1284 267 1228 438 112 1139 1041 574 216 25  
## [924] 123 335 624 352 305 581 570 83 239 294 657 1199 1234  
## [937] 1069 647 846 963 1312 1049 633 347 930 837 299 1288 11  
## [950] 477 823 481 304 608 730 742 790 611 1322 1221 1004 1216  
## [963] 494 922 284 1148 1262 706 186 386 330 486 795 423 542  
## [976] 1021 166 174 88 1157 1040 947 704 421 1051 1091 583 571  
## [989] 203 207 867 737 1168 876 1149 632 904 1045 1003 942 743  
## [1002] 868 763 467 1009 877 637 173 1120 1123 803 728 341 172  
## [1015] 116 1103 1207 1070 690 988 1080 617 450 408 1279 524 73  
## [1028] 1185 1215 836 141 1229 50 575 534 230 758 753 66 993  
## [1041] 511 125 464 891 1156 246 536 105 52 281 492 777 539  
## [1054] 816 235 1177 328 365 791 548 894 605 971 1311 237 198  
## [1067] 956 484 1217 1143 573 593 784 675 927 23 765 1247 871  
## [1080] 482 1063 1062 110 145 880 90 991 898 417 802 340 295  
## [1093] 162 638 1308 1253 566 994 1087 434 315 990 954 648 527  
## [1106] 625 1270 1242 809 1282 1165 906 887 912 101 1008 805 392  
## [1119] 498 40 379 79 98 457 404 820 21 140 26 517 1272  
## [1132] 1289 170 1152 553 1014 1076 1135 395 529 432 296 353 1321  
## [1145] 705 950 445 222 368 1172 1300 371 441 97 256 182 1273  
## [1158] 1316 175 717 1319 519 513 480 1058 181 201 55 35 731  
## [1171] 1025 22 121 606 1001 673 1127 584 821 223 974 1078 318  
## [1184] 53 336 324 1250 886 674 515 131 1280 826 1060 470 1256  
## [1197] 143 828 672 1162 825 208 794 710

insurance\_train <- insurance[indx,]  
insurance\_test <- insurance[-indx,]

## Step 3: Training a model on the data ----  
ins\_model <- lm(expenses ~ age + children + bmi + sex + smoker + region,  
 data = insurance\_train)  
ins\_model <- lm(expenses ~ ., data = insurance\_train) # this is equivalent to above  
  
# see the estimated beta coefficients  
ins\_model

##   
## Call:  
## lm(formula = expenses ~ ., data = insurance\_train)  
##   
## Coefficients:  
## (Intercept) age sexmale bmi   
## -11683.2 252.8 -175.6 340.1   
## children smokeryes regionnorthwest regionsoutheast   
## 424.7 23722.5 -477.1 -1036.1   
## regionsouthwest   
## -1110.7

## Step 4: Evaluating model performance ----  
# see more detail about the estimated beta coefficients  
summary(ins\_model)

##   
## Call:  
## lm(formula = expenses ~ ., data = insurance\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11101 -2874 -1013 1351 30052   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -11683.18 1042.58 -11.206 < 2e-16 \*\*\*  
## age 252.76 12.55 20.134 < 2e-16 \*\*\*  
## sexmale -175.65 351.85 -0.499 0.61772   
## bmi 340.09 30.34 11.210 < 2e-16 \*\*\*  
## children 424.74 145.41 2.921 0.00356 \*\*   
## smokeryes 23722.55 435.90 54.422 < 2e-16 \*\*\*  
## regionnorthwest -477.09 498.17 -0.958 0.33842   
## regionsoutheast -1036.08 505.58 -2.049 0.04065 \*   
## regionsouthwest -1110.68 503.81 -2.205 0.02767 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6064 on 1195 degrees of freedom  
## Multiple R-squared: 0.7484, Adjusted R-squared: 0.7467   
## F-statistic: 444.3 on 8 and 1195 DF, p-value: < 2.2e-16

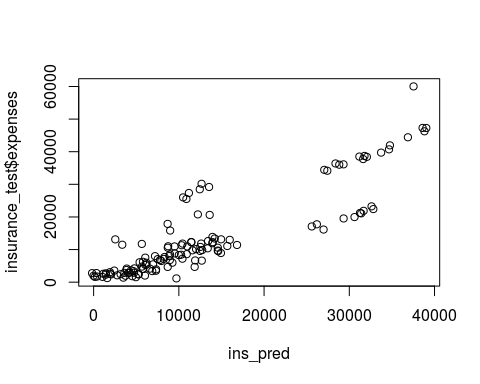
# generate predictions for the testing dataset  
ins\_pred <- predict(ins\_model, insurance\_test)  
  
# compare the distribution of predicted values vs. actual values  
summary(ins\_pred)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -180.1 5460.0 9856.2 12815.8 14569.3 39036.5

summary(insurance\_test)

## age sex bmi children smoker   
## Min. :18.00 female:66 Min. :17.40 Min. :0.000 no :107   
## 1st Qu.:23.25 male :68 1st Qu.:25.82 1st Qu.:0.000 yes: 27   
## Median :38.50 Median :30.30 Median :1.000   
## Mean :37.98 Mean :30.82 Mean :1.067   
## 3rd Qu.:51.00 3rd Qu.:34.40 3rd Qu.:2.000   
## Max. :64.00 Max. :53.10 Max. :5.000   
## region expenses   
## northeast:28 Min. : 1163   
## northwest:27 1st Qu.: 3962   
## southeast:43 Median : 9224   
## southwest:36 Mean :13293   
## 3rd Qu.:17583   
## Max. :60021

# plot  
plot(ins\_pred, insurance\_test$expenses)



# compare the correlation  
cor(ins\_pred, insurance\_test$expenses)

## [1] 0.8801707

# function to calculate the mean absolute error  
MAE <- function(actual, predicted) {  
 mean(abs(actual - predicted))   
}

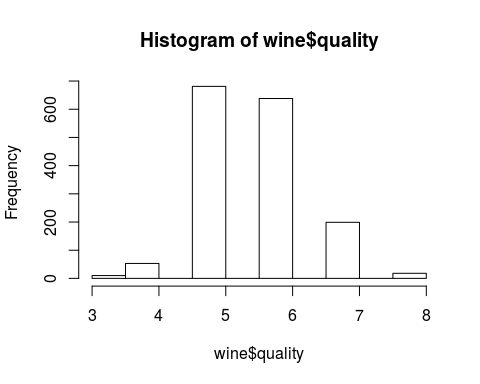
## Step 5: Improving model performance ----  
  
# add a higher-order "age" term  
insurance\_train$age2 <- insurance\_train$age^2  
  
# add an indicator for BMI >= 30  
insurance\_train$bmi30 <- ifelse(insurance\_train$bmi >= 30, 1, 0)  
  
# create final model  
ins\_model2 <- lm(expenses ~ age + age2 + children + bmi + sex +  
 bmi30\*smoker + region, data = insurance\_train)  
  
summary(ins\_model2)

##   
## Call:  
## lm(formula = expenses ~ age + age2 + children + bmi + sex + bmi30 \*   
## smoker + region, data = insurance\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17318.4 -1617.8 -1222.1 -712.5 24118.4   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 352.6880 1437.7018 0.245 0.806256   
## age -36.3585 63.0413 -0.577 0.564223   
## age2 3.7008 0.7844 4.718 2.66e-06 \*\*\*  
## children 678.1171 111.8317 6.064 1.78e-09 \*\*\*  
## bmi 120.5369 36.4932 3.303 0.000985 \*\*\*  
## sexmale -563.7237 258.8252 -2.178 0.029601 \*   
## bmi30 -1047.7185 446.7742 -2.345 0.019187 \*   
## smokeryes 13508.3587 460.5749 29.329 < 2e-16 \*\*\*  
## regionnorthwest -327.3447 366.0803 -0.894 0.371402   
## regionsoutheast -821.5959 372.3773 -2.206 0.027549 \*   
## regionsouthwest -1210.2123 370.2812 -3.268 0.001113 \*\*   
## bmi30:smokeryes 19805.7543 638.2719 31.030 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4456 on 1192 degrees of freedom  
## Multiple R-squared: 0.8645, Adjusted R-squared: 0.8632   
## F-statistic: 691.3 on 11 and 1192 DF, p-value: < 2.2e-16

#### Part 2: Regression Trees and Model Trees -------------------  
  
## Understanding regression trees and model trees ----  
  
## Example: Estimating Wine Quality ----  
## Step 2: Exploring and preparing the data ----  
wine <- read.csv("redwines.csv")  
  
# examine the wine data  
str(wine)

## 'data.frame': 1599 obs. of 12 variables:  
## $ fixed.acidity : num 6.5 9.1 6.9 7.3 12.5 5.4 10.4 7.9 7.3 9.5 ...  
## $ volatile.acidity : num 0.9 0.22 0.52 0.59 0.28 0.74 0.28 0.4 0.39 0.37 ...  
## $ citric.acid : num 0 0.24 0.25 0.26 0.54 0.09 0.54 0.3 0.31 0.52 ...  
## $ residual.sugar : num 1.6 2.1 2.6 2 2.3 1.7 2.7 1.8 2.4 2 ...  
## $ chlorides : num 0.052 0.078 0.081 0.08 0.082 0.089 0.105 0.157 0.074 0.088 ...  
## $ free.sulfur.dioxide : num 9 1 10 17 12 16 5 2 9 12 ...  
## $ total.sulfur.dioxide: num 17 28 37 104 29 26 19 45 46 51 ...  
## $ density : num 0.995 0.999 0.997 0.996 1 ...  
## $ pH : num 3.5 3.41 3.46 3.28 3.11 3.67 3.25 3.31 3.41 3.29 ...  
## $ sulphates : num 0.63 0.87 0.5 0.52 1.36 0.56 0.63 0.91 0.54 0.58 ...  
## $ alcohol : num 10.9 10.3 11 9.9 9.8 11.6 9.5 9.5 9.4 11.1 ...  
## $ quality : int 6 6 5 5 7 6 5 6 6 6 ...

# the distribution of quality ratings  
hist(wine$quality)



# summary statistics of the wine data  
summary(wine)

## fixed.acidity volatile.acidity citric.acid residual.sugar   
## Min. : 4.60 Min. :0.1200 Min. :0.000 Min. : 0.900   
## 1st Qu.: 7.10 1st Qu.:0.3900 1st Qu.:0.090 1st Qu.: 1.900   
## Median : 7.90 Median :0.5200 Median :0.260 Median : 2.200   
## Mean : 8.32 Mean :0.5278 Mean :0.271 Mean : 2.539   
## 3rd Qu.: 9.20 3rd Qu.:0.6400 3rd Qu.:0.420 3rd Qu.: 2.600   
## Max. :15.90 Max. :1.5800 Max. :1.000 Max. :15.500   
## chlorides free.sulfur.dioxide total.sulfur.dioxide  
## Min. :0.01200 Min. : 1.00 Min. : 6.00   
## 1st Qu.:0.07000 1st Qu.: 7.00 1st Qu.: 22.00   
## Median :0.07900 Median :14.00 Median : 38.00   
## Mean :0.08747 Mean :15.87 Mean : 46.47   
## 3rd Qu.:0.09000 3rd Qu.:21.00 3rd Qu.: 62.00   
## Max. :0.61100 Max. :72.00 Max. :289.00   
## density pH sulphates alcohol   
## Min. :0.9901 Min. :2.740 Min. :0.3300 Min. : 8.40   
## 1st Qu.:0.9956 1st Qu.:3.210 1st Qu.:0.5500 1st Qu.: 9.50   
## Median :0.9968 Median :3.310 Median :0.6200 Median :10.20   
## Mean :0.9967 Mean :3.311 Mean :0.6581 Mean :10.42   
## 3rd Qu.:0.9978 3rd Qu.:3.400 3rd Qu.:0.7300 3rd Qu.:11.10   
## Max. :1.0037 Max. :4.010 Max. :2.0000 Max. :14.90   
## quality   
## Min. :3.000   
## 1st Qu.:5.000   
## Median :6.000   
## Mean :5.636   
## 3rd Qu.:6.000   
## Max. :8.000

wine\_train <- wine[1:999, ]  
wine\_test <- wine[1000:1599, ]

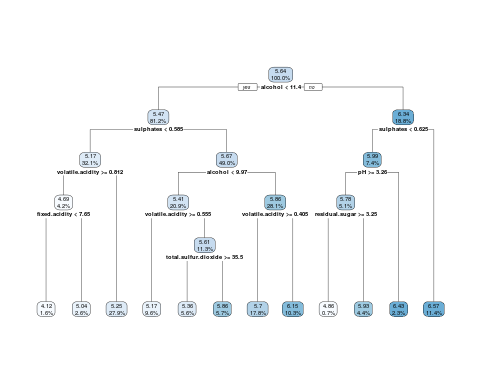
## Step 3: Training a model on the data ----  
# regression tree using rpart  
library(rpart)  
m.rpart <- rpart(quality ~ ., data = wine\_train)  
  
# get basic information about the tree  
m.rpart

## n= 999   
##   
## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 999 655.099100 5.636637   
## 2) alcohol< 11.45 811 434.180000 5.473490   
## 4) sulphates< 0.585 321 118.230500 5.174455   
## 8) volatile.acidity>=0.8125 42 26.976190 4.690476   
## 16) fixed.acidity< 7.65 16 9.750000 4.125000 \*  
## 17) fixed.acidity>=7.65 26 8.961538 5.038462 \*  
## 9) volatile.acidity< 0.8125 279 79.935480 5.247312 \*  
## 5) sulphates>=0.585 490 268.440800 5.669388   
## 10) alcohol< 9.975 209 98.430620 5.406699   
## 20) volatile.acidity>=0.555 96 31.333330 5.166667 \*  
## 21) volatile.acidity< 0.555 113 56.867260 5.610619   
## 42) total.sulfur.dioxide>=35.5 56 24.857140 5.357143 \*  
## 43) total.sulfur.dioxide< 35.5 57 24.877190 5.859649 \*  
## 11) alcohol>=9.975 281 144.861200 5.864769   
## 22) volatile.acidity>=0.405 178 77.219100 5.702247 \*  
## 23) volatile.acidity< 0.405 103 54.815530 6.145631 \*  
## 3) alcohol>=11.45 188 106.212800 6.340426   
## 6) sulphates< 0.625 74 34.986490 5.986486   
## 12) pH>=3.265 51 20.627450 5.784314   
## 24) residual.sugar>=3.25 7 4.857143 4.857143 \*  
## 25) residual.sugar< 3.25 44 8.795455 5.931818 \*  
## 13) pH< 3.265 23 7.652174 6.434783 \*  
## 7) sulphates>=0.625 114 55.938600 6.570175 \*

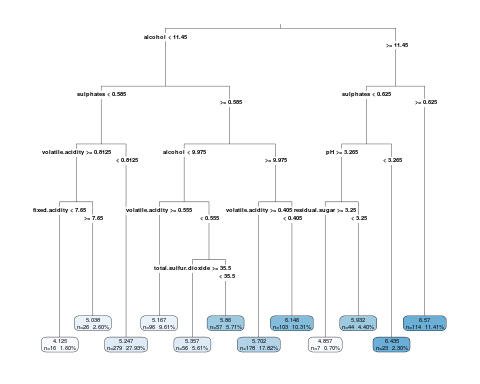
# get more detailed information about the tree  
summary(m.rpart)

## Call:  
## rpart(formula = quality ~ ., data = wine\_train)  
## n= 999   
##   
## CP nsplit rel error xerror xstd  
## 1 0.17509764 0 1.0000000 1.0007887 0.04937515  
## 2 0.07252136 1 0.8249024 0.8334357 0.04406135  
## 3 0.03838959 2 0.7523810 0.7708389 0.04183796  
## 4 0.02333644 3 0.7139914 0.7453385 0.04164482  
## 5 0.01957959 4 0.6906550 0.7401206 0.04110092  
## 6 0.01727808 5 0.6710754 0.7314128 0.04024081  
## 7 0.01561601 6 0.6537973 0.7219078 0.03920640  
## 8 0.01261588 7 0.6381813 0.7062627 0.03851310  
## 9 0.01088831 8 0.6255654 0.7031333 0.03879462  
## 10 0.01044248 9 0.6146771 0.7006339 0.03775883  
## 11 0.01000000 11 0.5937921 0.6914282 0.03739046  
##   
## Variable importance  
## alcohol sulphates density   
## 33 15 12   
## volatile.acidity fixed.acidity pH   
## 10 7 5   
## citric.acid chlorides total.sulfur.dioxide   
## 4 4 4   
## residual.sugar free.sulfur.dioxide   
## 3 2   
##   
## Node number 1: 999 observations, complexity param=0.1750976  
## mean=5.636637, MSE=0.6557549   
## left son=2 (811 obs) right son=3 (188 obs)  
## Primary splits:  
## alcohol < 11.45 to the left, improve=0.17509760, (0 missing)  
## sulphates < 0.635 to the left, improve=0.12132530, (0 missing)  
## volatile.acidity < 0.555 to the right, improve=0.09945794, (0 missing)  
## citric.acid < 0.305 to the left, improve=0.06347979, (0 missing)  
## density < 0.99539 to the right, improve=0.06318247, (0 missing)  
## Surrogate splits:  
## density < 0.99419 to the right, agree=0.871, adj=0.314, (0 split)  
## fixed.acidity < 5.5 to the right, agree=0.830, adj=0.096, (0 split)  
## chlorides < 0.0525 to the right, agree=0.827, adj=0.080, (0 split)  
## pH < 3.675 to the left, agree=0.820, adj=0.043, (0 split)  
## volatile.acidity < 0.17 to the right, agree=0.814, adj=0.011, (0 split)  
##   
## Node number 2: 811 observations, complexity param=0.07252136  
## mean=5.47349, MSE=0.5353638   
## left son=4 (321 obs) right son=5 (490 obs)  
## Primary splits:  
## sulphates < 0.585 to the left, improve=0.10942160, (0 missing)  
## volatile.acidity < 0.405 to the right, improve=0.08535928, (0 missing)  
## alcohol < 9.975 to the left, improve=0.08041403, (0 missing)  
## citric.acid < 0.305 to the left, improve=0.04217574, (0 missing)  
## pH < 3.475 to the right, improve=0.03267395, (0 missing)  
## Surrogate splits:  
## volatile.acidity < 0.6875 to the right, agree=0.651, adj=0.118, (0 split)  
## total.sulfur.dioxide < 79.5 to the right, agree=0.628, adj=0.059, (0 split)  
## citric.acid < 0.105 to the left, agree=0.625, adj=0.053, (0 split)  
## density < 0.99477 to the left, agree=0.618, adj=0.034, (0 split)  
## residual.sugar < 6.25 to the right, agree=0.613, adj=0.022, (0 split)  
##   
## Node number 3: 188 observations, complexity param=0.02333644  
## mean=6.340426, MSE=0.5649615   
## left son=6 (74 obs) right son=7 (114 obs)  
## Primary splits:  
## sulphates < 0.625 to the left, improve=0.14393450, (0 missing)  
## citric.acid < 0.325 to the left, improve=0.06504058, (0 missing)  
## fixed.acidity < 7.75 to the left, improve=0.06486160, (0 missing)  
## volatile.acidity < 0.575 to the right, improve=0.06009615, (0 missing)  
## pH < 3.275 to the right, improve=0.04505435, (0 missing)  
## Surrogate splits:  
## density < 0.99377 to the left, agree=0.686, adj=0.203, (0 split)  
## fixed.acidity < 7.15 to the left, agree=0.670, adj=0.162, (0 split)  
## residual.sugar < 1.775 to the left, agree=0.665, adj=0.149, (0 split)  
## citric.acid < 0.065 to the left, agree=0.654, adj=0.122, (0 split)  
## total.sulfur.dioxide < 11.5 to the left, agree=0.644, adj=0.095, (0 split)  
##   
## Node number 4: 321 observations, complexity param=0.01727808  
## mean=5.174455, MSE=0.3683194   
## left son=8 (42 obs) right son=9 (279 obs)  
## Primary splits:  
## volatile.acidity < 0.8125 to the right, improve=0.09573547, (0 missing)  
## pH < 3.475 to the right, improve=0.05400178, (0 missing)  
## chlorides < 0.13 to the right, improve=0.04780725, (0 missing)  
## sulphates < 0.525 to the left, improve=0.02548852, (0 missing)  
## total.sulfur.dioxide < 14.5 to the left, improve=0.02015651, (0 missing)  
## Surrogate splits:  
## chlorides < 0.13 to the right, agree=0.879, adj=0.071, (0 split)  
## total.sulfur.dioxide < 144.5 to the right, agree=0.875, adj=0.048, (0 split)  
## residual.sugar < 1.3 to the left, agree=0.872, adj=0.024, (0 split)  
##   
## Node number 5: 490 observations, complexity param=0.03838959  
## mean=5.669388, MSE=0.5478384   
## left son=10 (209 obs) right son=11 (281 obs)  
## Primary splits:  
## alcohol < 9.975 to the left, improve=0.09368540, (0 missing)  
## volatile.acidity < 0.405 to the right, improve=0.09246757, (0 missing)  
## total.sulfur.dioxide < 57.5 to the right, improve=0.06396781, (0 missing)  
## density < 0.995745 to the right, improve=0.04563609, (0 missing)  
## chlorides < 0.0975 to the right, improve=0.03886579, (0 missing)  
## Surrogate splits:  
## chlorides < 0.1045 to the right, agree=0.622, adj=0.115, (0 split)  
## total.sulfur.dioxide < 62.5 to the right, agree=0.618, adj=0.105, (0 split)  
## density < 0.99706 to the right, agree=0.618, adj=0.105, (0 split)  
## sulphates < 0.975 to the right, agree=0.604, adj=0.072, (0 split)  
## pH < 3.015 to the left, agree=0.588, adj=0.033, (0 split)  
##   
## Node number 6: 74 observations, complexity param=0.01044248  
## mean=5.986486, MSE=0.4727904   
## left son=12 (51 obs) right son=13 (23 obs)  
## Primary splits:  
## pH < 3.265 to the right, improve=0.19169860, (0 missing)  
## volatile.acidity < 0.57 to the right, improve=0.13266510, (0 missing)  
## citric.acid < 0.405 to the left, improve=0.12454320, (0 missing)  
## free.sulfur.dioxide < 25 to the left, improve=0.08714755, (0 missing)  
## fixed.acidity < 8.7 to the left, improve=0.08471221, (0 missing)  
## Surrogate splits:  
## citric.acid < 0.335 to the left, agree=0.851, adj=0.522, (0 split)  
## fixed.acidity < 7.8 to the left, agree=0.838, adj=0.478, (0 split)  
## chlorides < 0.0995 to the left, agree=0.770, adj=0.261, (0 split)  
## volatile.acidity < 0.285 to the right, agree=0.757, adj=0.217, (0 split)  
## free.sulfur.dioxide < 34 to the left, agree=0.757, adj=0.217, (0 split)  
##   
## Node number 7: 114 observations  
## mean=6.570175, MSE=0.4906894   
##   
## Node number 8: 42 observations, complexity param=0.01261588  
## mean=4.690476, MSE=0.6422902   
## left son=16 (16 obs) right son=17 (26 obs)  
## Primary splits:  
## fixed.acidity < 7.65 to the left, improve=0.3063684, (0 missing)  
## density < 0.995225 to the left, improve=0.2162401, (0 missing)  
## pH < 3.28 to the right, improve=0.1593116, (0 missing)  
## alcohol < 9.65 to the right, improve=0.1498545, (0 missing)  
## citric.acid < 0.065 to the left, improve=0.1356014, (0 missing)  
## Surrogate splits:  
## pH < 3.375 to the right, agree=0.833, adj=0.563, (0 split)  
## density < 0.995 to the left, agree=0.762, adj=0.375, (0 split)  
## chlorides < 0.089 to the right, agree=0.738, adj=0.313, (0 split)  
## citric.acid < 0.095 to the left, agree=0.714, adj=0.250, (0 split)  
## alcohol < 10.95 to the right, agree=0.690, adj=0.187, (0 split)  
##   
## Node number 9: 279 observations  
## mean=5.247312, MSE=0.2865071   
##   
## Node number 10: 209 observations, complexity param=0.01561601  
## mean=5.406699, MSE=0.4709599   
## left son=20 (96 obs) right son=21 (113 obs)  
## Primary splits:  
## volatile.acidity < 0.555 to the right, improve=0.10393140, (0 missing)  
## fixed.acidity < 11.8 to the left, improve=0.10289890, (0 missing)  
## total.sulfur.dioxide < 46.5 to the right, improve=0.08618566, (0 missing)  
## free.sulfur.dioxide < 22.5 to the right, improve=0.07322011, (0 missing)  
## residual.sugar < 1.55 to the left, improve=0.03927434, (0 missing)  
## Surrogate splits:  
## citric.acid < 0.245 to the left, agree=0.742, adj=0.437, (0 split)  
## density < 0.997145 to the left, agree=0.632, adj=0.198, (0 split)  
## fixed.acidity < 7.15 to the left, agree=0.627, adj=0.188, (0 split)  
## total.sulfur.dioxide < 50.5 to the right, agree=0.622, adj=0.177, (0 split)  
## chlorides < 0.0825 to the right, agree=0.593, adj=0.115, (0 split)  
##   
## Node number 11: 281 observations, complexity param=0.01957959  
## mean=5.864769, MSE=0.5155203   
## left son=22 (178 obs) right son=23 (103 obs)  
## Primary splits:  
## volatile.acidity < 0.405 to the right, improve=0.08854389, (0 missing)  
## sulphates < 0.725 to the left, improve=0.06687032, (0 missing)  
## pH < 3.455 to the right, improve=0.06206956, (0 missing)  
## total.sulfur.dioxide < 83 to the right, improve=0.04949434, (0 missing)  
## citric.acid < 0.655 to the left, improve=0.04127300, (0 missing)  
## Surrogate splits:  
## citric.acid < 0.315 to the left, agree=0.765, adj=0.359, (0 split)  
## sulphates < 0.765 to the left, agree=0.701, adj=0.184, (0 split)  
## free.sulfur.dioxide < 33.5 to the left, agree=0.655, adj=0.058, (0 split)  
## chlorides < 0.0575 to the right, agree=0.651, adj=0.049, (0 split)  
## total.sulfur.dioxide < 10.5 to the right, agree=0.648, adj=0.039, (0 split)  
##   
## Node number 12: 51 observations, complexity param=0.01044248  
## mean=5.784314, MSE=0.4044598   
## left son=24 (7 obs) right son=25 (44 obs)  
## Primary splits:  
## residual.sugar < 3.25 to the right, improve=0.3381345, (0 missing)  
## density < 0.995535 to the right, improve=0.2401095, (0 missing)  
## free.sulfur.dioxide < 11.5 to the left, improve=0.1660851, (0 missing)  
## total.sulfur.dioxide < 25.5 to the left, improve=0.1516160, (0 missing)  
## alcohol < 12.1 to the left, improve=0.1196198, (0 missing)  
## Surrogate splits:  
## density < 0.99563 to the right, agree=0.922, adj=0.429, (0 split)  
## fixed.acidity < 8.35 to the right, agree=0.902, adj=0.286, (0 split)  
## pH < 3.285 to the left, agree=0.882, adj=0.143, (0 split)  
##   
## Node number 13: 23 observations  
## mean=6.434783, MSE=0.3327032   
##   
## Node number 16: 16 observations  
## mean=4.125, MSE=0.609375   
##   
## Node number 17: 26 observations  
## mean=5.038462, MSE=0.3446746   
##   
## Node number 20: 96 observations  
## mean=5.166667, MSE=0.3263889   
##   
## Node number 21: 113 observations, complexity param=0.01088831  
## mean=5.610619, MSE=0.5032501   
## left son=42 (56 obs) right son=43 (57 obs)  
## Primary splits:  
## total.sulfur.dioxide < 35.5 to the right, improve=0.12543110, (0 missing)  
## free.sulfur.dioxide < 14.5 to the right, improve=0.12278370, (0 missing)  
## fixed.acidity < 11.8 to the left, improve=0.11975460, (0 missing)  
## volatile.acidity < 0.315 to the right, improve=0.08661830, (0 missing)  
## alcohol < 9.65 to the left, improve=0.05487065, (0 missing)  
## Surrogate splits:  
## free.sulfur.dioxide < 13.5 to the right, agree=0.912, adj=0.821, (0 split)  
## volatile.acidity < 0.365 to the right, agree=0.690, adj=0.375, (0 split)  
## pH < 3.305 to the right, agree=0.681, adj=0.357, (0 split)  
## residual.sugar < 2.25 to the right, agree=0.655, adj=0.304, (0 split)  
## fixed.acidity < 9.95 to the left, agree=0.646, adj=0.286, (0 split)  
##   
## Node number 22: 178 observations  
## mean=5.702247, MSE=0.4338152   
##   
## Node number 23: 103 observations  
## mean=6.145631, MSE=0.5321897   
##   
## Node number 24: 7 observations  
## mean=4.857143, MSE=0.6938776   
##   
## Node number 25: 44 observations  
## mean=5.931818, MSE=0.1998967   
##   
## Node number 42: 56 observations  
## mean=5.357143, MSE=0.4438776   
##   
## Node number 43: 57 observations  
## mean=5.859649, MSE=0.436442

# use the rpart.plot package to create a visualization  
library(rpart.plot)  
  
# a basic decision tree diagram  
rpart.plot(m.rpart, digits = 3)



# a few adjustments to the diagram  
rpart.plot(m.rpart, digits = 4, fallen.leaves = TRUE, type = 3, extra = 101)



## Step 4: Evaluate model performance ----  
  
# generate predictions for the testing dataset  
p.rpart <- predict(m.rpart, wine\_test)  
  
# compare the distribution of predicted values vs. actual values  
summary(p.rpart)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4.125 5.247 5.357 5.584 5.878 6.570

summary(wine\_test$quality)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.000 5.000 6.000 5.635 6.000 8.000

# compare the correlation  
cor(p.rpart, wine\_test$quality)

## [1] 0.5709981

# function to calculate the mean absolute error  
MAE <- function(actual, predicted) {  
 mean(abs(actual - predicted))   
}  
  
# mean absolute error between predicted and actual values  
MAE(p.rpart, wine\_test$quality)

## [1] 0.5273212

# mean absolute error between actual values and mean value  
mean(wine\_train$quality) # result = 5.87

## [1] 5.636637

MAE(5.87, wine\_test$quality)

## [1] 0.6699

## Step 5: Improving model performance ----  
# train a M5' Model Tree  
library(RWeka)  
m.m5p <- M5P(quality ~ ., data = wine\_train)  
  
# display the tree  
m.m5p

## M5 pruned model tree:  
## (using smoothed linear models)  
## LM1 (999/81.523%)  
##   
## LM num: 1  
## quality =   
## -0.9338 \* volatile.acidity   
## - 2.3157 \* chlorides   
## - 0.002 \* total.sulfur.dioxide   
## - 0.4718 \* pH   
## + 0.817 \* sulphates   
## + 0.2872 \* alcohol   
## + 4.4449  
##   
## Number of Rules : 1

# get a summary of the model's performance  
summary(m.m5p)

##   
## === Summary ===  
##   
## Correlation coefficient 0.5791  
## Mean absolute error 0.5134  
## Root mean squared error 0.6602  
## Relative absolute error 75.2486 %  
## Root relative squared error 81.523 %  
## Total Number of Instances 999

# generate predictions for the model  
p.m5p <- predict(m.m5p, wine\_test)  
  
# summary statistics about the predictions  
summary(p.m5p)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4.610 5.258 5.543 5.611 5.966 6.845

# correlation between the predicted and true values  
cor(p.m5p, wine\_test$quality)

## [1] 0.6280751

# mean absolute error of predicted and true values  
# (uses a custom function defined above)  
MAE(wine\_test$quality, p.m5p)

## [1] 0.4920995