```
library(funModeling) #used to plot categorical variables an their frequencies

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units

## funModeling v.1.9.4 :)
## Examples and tutorials at livebook.datascienceheroes.com
##  / Now in Spanish: librovivodecienciadedatos.ai

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.1 ──

## ✓ tibble  3.1.6     ✓ dplyr   1.0.8
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1
## ✓ purrr   0.3.4

## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x dplyr::src()       masks Hmisc::src()
## x dplyr::summarize() masks Hmisc::summarize()

library(ggplot2)
library(caret)

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(randomForest)

## randomForest 4.7-1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

library(broom)
library(ggcorrplot)
library(nnet) # multinational logistic regression
library(yardstick)

## For binary classification, the first factor level is assumed to be the
event.
## Use the argument `event_level = "second"` to alter this as needed.

##
## Attaching package: 'yardstick'

## The following objects are masked from 'package:caret':
##
##     precision, recall, sensitivity, specificity

## The following object is masked from 'package:readr':
##
##     spec

library(MASS)

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

library(HSAUR2)

## Loading required package: tools

library(ISLR)
library(tictoc)
```

```r
#the libraries used

#read in data
house.data = read.csv("../data/house-data.csv")

#initial statistics on dataset
attributes(house.data)
```

```
## $names
##  [1] "Id"            "LotFrontage"   "LotArea"       "Street"
##  [5] "Alley"         "Utilities"     "LotConfig"     "Neighborhood"
##  [9] "Condition1"    "Condition2"    "BldgType"      "HouseStyle"
## [13] "OverallQual"   "OverallCond"   "YearBuilt"     "RoofStyle"
## [17] "RoofMatl"      "Exterior1st"   "MasVnrArea"    "ExterQual"
## [21] "ExterCond"     "Foundation"    "BsmtQual"      "BsmtCond"
## [25] "TotalBsmtSF"   "Heating"       "X1stFlrSF"     "X2ndFlrSF"
## [29] "LowQualFinSF"  "GrLivArea"     "FullBath"      "BedroomAbvGr"
## [33] "KitchenAbvGr"  "KitchenQual"   "TotRmsAbvGrd"  "Functional"
## [37] "Fireplaces"    "GarageType"    "GarageArea"    "GarageCond"
## [41] "PavedDrive"    "PoolArea"      "PoolQC"        "Fence"
## [45] "MiscFeature"   "MiscVal"       "MoSold"        "YrSold"
## [49] "SaleType"      "SaleCondition" "SalePrice"
##
## $class
## [1] "data.frame"
##
## $row.names
##    [1]    1    2    3    4    5    6    7    8    9   10   11   12   13
14
##   [15]   15   16   17   18   19   20   21   22   23   24   25   26   27
28
##   [29]   29   30   31   32   33   34   35   36   37   38   39   40   41
42
##   [43]   43   44   45   46   47   48   49   50   51   52   53   54   55
56
##   [57]   57   58   59   60   61   62   63   64   65   66   67   68   69
70
##   [71]   71   72   73   74   75   76   77   78   79   80   81   82   83
84
##   [85]   85   86   87   88   89   90   91   92   93   94   95   96   97
98
##   [99]   99  100  101  102  103  104  105  106  107  108  109  110  111
112
##  [113]  113  114  115  116  117  118  119  120  121  122  123  124  125
126
##  [127]  127  128  129  130  131  132  133  134  135  136  137  138  139
140
##  [141]  141  142  143  144  145  146  147  148  149  150  151  152  153
154
##  [155]  155  156  157  158  159  160  161  162  163  164  165  166  167
```

```
168
##  [169]  169  170  171  172  173  174  175  176  177  178  179  180  181
182
##  [183]  183  184  185  186  187  188  189  190  191  192  193  194  195
196
##  [197]  197  198  199  200  201  202  203  204  205  206  207  208  209
210
##  [211]  211  212  213  214  215  216  217  218  219  220  221  222  223
224
##  [225]  225  226  227  228  229  230  231  232  233  234  235  236  237
238
##  [239]  239  240  241  242  243  244  245  246  247  248  249  250  251
252
##  [253]  253  254  255  256  257  258  259  260  261  262  263  264  265
266
##  [267]  267  268  269  270  271  272  273  274  275  276  277  278  279
280
##  [281]  281  282  283  284  285  286  287  288  289  290  291  292  293
294
##  [295]  295  296  297  298  299  300  301  302  303  304  305  306  307
308
##  [309]  309  310  311  312  313  314  315  316  317  318  319  320  321
322
##  [323]  323  324  325  326  327  328  329  330  331  332  333  334  335
336
##  [337]  337  338  339  340  341  342  343  344  345  346  347  348  349
350
##  [351]  351  352  353  354  355  356  357  358  359  360  361  362  363
364
##  [365]  365  366  367  368  369  370  371  372  373  374  375  376  377
378
##  [379]  379  380  381  382  383  384  385  386  387  388  389  390  391
392
##  [393]  393  394  395  396  397  398  399  400  401  402  403  404  405
406
##  [407]  407  408  409  410  411  412  413  414  415  416  417  418  419
420
##  [421]  421  422  423  424  425  426  427  428  429  430  431  432  433
434
##  [435]  435  436  437  438  439  440  441  442  443  444  445  446  447
448
##  [449]  449  450  451  452  453  454  455  456  457  458  459  460  461
462
##  [463]  463  464  465  466  467  468  469  470  471  472  473  474  475
476
##  [477]  477  478  479  480  481  482  483  484  485  486  487  488  489
490
##  [491]  491  492  493  494  495  496  497  498  499  500  501  502  503
504
##  [505]  505  506  507  508  509  510  511  512  513  514  515  516  517
```

```
518
##  [519]  519  520  521  522  523  524  525  526  527  528  529  530  531
532
##  [533]  533  534  535  536  537  538  539  540  541  542  543  544  545
546
##  [547]  547  548  549  550  551  552  553  554  555  556  557  558  559
560
##  [561]  561  562  563  564  565  566  567  568  569  570  571  572  573
574
##  [575]  575  576  577  578  579  580  581  582  583  584  585  586  587
588
##  [589]  589  590  591  592  593  594  595  596  597  598  599  600  601
602
##  [603]  603  604  605  606  607  608  609  610  611  612  613  614  615
616
##  [617]  617  618  619  620  621  622  623  624  625  626  627  628  629
630
##  [631]  631  632  633  634  635  636  637  638  639  640  641  642  643
644
##  [645]  645  646  647  648  649  650  651  652  653  654  655  656  657
658
##  [659]  659  660  661  662  663  664  665  666  667  668  669  670  671
672
##  [673]  673  674  675  676  677  678  679  680  681  682  683  684  685
686
##  [687]  687  688  689  690  691  692  693  694  695  696  697  698  699
700
##  [701]  701  702  703  704  705  706  707  708  709  710  711  712  713
714
##  [715]  715  716  717  718  719  720  721  722  723  724  725  726  727
728
##  [729]  729  730  731  732  733  734  735  736  737  738  739  740  741
742
##  [743]  743  744  745  746  747  748  749  750  751  752  753  754  755
756
##  [757]  757  758  759  760  761  762  763  764  765  766  767  768  769
770
##  [771]  771  772  773  774  775  776  777  778  779  780  781  782  783
784
##  [785]  785  786  787  788  789  790  791  792  793  794  795  796  797
798
##  [799]  799  800  801  802  803  804  805  806  807  808  809  810  811
812
##  [813]  813  814  815  816  817  818  819  820  821  822  823  824  825
826
##  [827]  827  828  829  830  831  832  833  834  835  836  837  838  839
840
##  [841]  841  842  843  844  845  846  847  848  849  850  851  852  853
854
##  [855]  855  856  857  858  859  860  861  862  863  864  865  866  867
```

```
868
##  [869]  869  870  871  872  873  874  875  876  877  878  879  880  881
882
##  [883]  883  884  885  886  887  888  889  890  891  892  893  894  895
896
##  [897]  897  898  899  900  901  902  903  904  905  906  907  908  909
910
##  [911]  911  912  913  914  915  916  917  918  919  920  921  922  923
924
##  [925]  925  926  927  928  929  930  931  932  933  934  935  936  937
938
##  [939]  939  940  941  942  943  944  945  946  947  948  949  950  951
952
##  [953]  953  954  955  956  957  958  959  960  961  962  963  964  965
966
##  [967]  967  968  969  970  971  972  973  974  975  976  977  978  979
980
##  [981]  981  982  983  984  985  986  987  988  989  990  991  992  993
994
##  [995]  995  996  997  998  999 1000 1001 1002 1003 1004 1005 1006 1007
1008
## [1009] 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021
1022
## [1023] 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035
1036
## [1037] 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049
1050
## [1051] 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063
1064
## [1065] 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077
1078
## [1079] 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091
1092
## [1093] 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105
1106
## [1107] 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119
1120
## [1121] 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133
1134
## [1135] 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147
1148
## [1149] 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161
1162
## [1163] 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175
1176
## [1177] 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189
1190
## [1191] 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203
1204
## [1205] 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217
```

```
1218
## [1219] 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231
1232
## [1233] 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245
1246
## [1247] 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259
1260
## [1261] 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273
1274
## [1275] 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287
1288
## [1289] 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301
1302
## [1303] 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315
1316
## [1317] 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329
1330
## [1331] 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343
1344
## [1345] 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357
1358
## [1359] 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371
1372
## [1373] 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385
1386
## [1387] 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399
1400
## [1401] 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413
1414
## [1415] 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427
1428
## [1429] 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441
1442
## [1443] 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455
1456
## [1457] 1457 1458 1459 1460
```

```
dim(house.data)
```

```
## [1] 1460   51
```

```
#having a look at the variables
str(house.data)
```

```
## 'data.frame':    1460 obs. of  51 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120
7420 ...
##  $ Street       : chr  "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley        : chr  NA NA NA NA ...
```

```
##  $ Utilities    : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ LotConfig    : chr  "Inside" "FR2" "Inside" "Corner" ...
##  $ Neighborhood : chr  "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
##  $ Condition1   : chr  "Norm" "Feedr" "Norm" "Norm" ...
##  $ Condition2   : chr  "Norm" "Norm" "Norm" "Norm" ...
##  $ BldgType     : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ HouseStyle   : chr  "2Story" "1Story" "2Story" "2Story" ...
##  $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond  : int  5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939
...
##  $ RoofStyle    : chr  "Gable" "Gable" "Gable" "Gable" ...
##  $ RoofMatl     : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior1st  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
##  $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : chr  "Gd" "TA" "Gd" "TA" ...
##  $ ExterCond    : chr  "TA" "TA" "TA" "TA" ...
##  $ Foundation   : chr  "PConc" "CBlock" "PConc" "BrkTil" ...
##  $ BsmtQual     : chr  "Gd" "Gd" "Gd" "TA" ...
##  $ BsmtCond     : chr  "TA" "TA" "TA" "Gd" ...
##  $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ Heating      : chr  "GasA" "GasA" "GasA" "GasA" ...
##  $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077
...
##  $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : chr  "Gd" "TA" "Gd" "Gd" ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : chr  "Typ" "Typ" "Typ" "Typ" ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ GarageType   : chr  "Attchd" "Attchd" "Attchd" "Detchd" ...
##  $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageCond   : chr  "TA" "TA" "TA" "TA" ...
##  $ PavedDrive   : chr  "Y" "Y" "Y" "Y" ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : chr  NA NA NA NA ...
##  $ Fence        : chr  NA NA NA NA ...
##  $ MiscFeature  : chr  NA NA NA NA ...
##  $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008
...
##  $ SaleType     : chr  "WD" "WD" "WD" "WD" ...
##  $ SaleCondition: chr  "Normal" "Normal" "Normal" "Abnorml" ...
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000
200000 129900 118000 ...
```

```
summary(house.data)

##        Id          LotFrontage        LotArea          Street
##  Min.   :   1.0   Min.   : 21.00   Min.   :  1300   Length:1460
##  1st Qu.: 365.8   1st Qu.: 59.00   1st Qu.:  7554   Class :character
##  Median : 730.5   Median : 69.00   Median :  9478   Mode  :character
##  Mean   : 730.5   Mean   : 70.05   Mean   : 10517
##  3rd Qu.:1095.2   3rd Qu.: 80.00   3rd Qu.: 11602
##  Max.   :1460.0   Max.   :313.00   Max.   :215245
##                   NA's   :259
##     Alley            Utilities          LotConfig         Neighborhood
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   Condition1         Condition2          BldgType          HouseStyle
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   OverallQual      OverallCond      YearBuilt     RoofStyle
##  Min.   : 1.000   Min.   :1.000   Min.   :1872   Length:1460
##  1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954   Class :character
##  Median : 6.000   Median :5.000   Median :1973   Mode  :character
##  Mean   : 6.099   Mean   :5.575   Mean   :1971
##  3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2000
##  Max.   :10.000   Max.   :9.000   Max.   :2010
##
##    RoofMatl         Exterior1st          MasVnrArea      ExterQual
##  Length:1460        Length:1460        Min.   :   0.0   Length:1460
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median :   0.0   Mode  :character
##                                        Mean   : 103.7
##                                        3rd Qu.: 166.0
##                                        Max.   :1600.0
##                                        NA's   :8
##   ExterCond          Foundation          BsmtQual           BsmtCond
##  Length:1460        Length:1460        Length:1460        Length:1460
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

```
##   TotalBsmtSF       Heating             X1stFlrSF      X2ndFlrSF
##  Min.   :   0.0   Length:1460        Min.   : 334   Min.   :   0
##  1st Qu.: 795.8   Class :character   1st Qu.: 882   1st Qu.:   0
##  Median : 991.5   Mode  :character   Median :1087   Median :   0
##  Mean   :1057.4                      Mean   :1163   Mean   : 347
##  3rd Qu.:1298.2                      3rd Qu.:1391   3rd Qu.: 728
##  Max.   :6110.0                      Max.   :4692   Max.   :2065
##
##   LowQualFinSF       GrLivArea       FullBath      BedroomAbvGr
##  Min.   :  0.000   Min.   : 334   Min.   :0.000   Min.   :0.000
##  1st Qu.:  0.000   1st Qu.:1130   1st Qu.:1.000   1st Qu.:2.000
##  Median :  0.000   Median :1464   Median :2.000   Median :3.000
##  Mean   :  5.845   Mean   :1515   Mean   :1.565   Mean   :2.866
##  3rd Qu.:  0.000   3rd Qu.:1777   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :572.000   Max.   :5642   Max.   :3.000   Max.   :8.000
##
##   KitchenAbvGr   KitchenQual         TotRmsAbvGrd     Functional
##  Min.   :0.000   Length:1460        Min.   : 2.000   Length:1460
##  1st Qu.:1.000   Class :character   1st Qu.: 5.000   Class :character
##  Median :1.000   Mode  :character   Median : 6.000   Mode  :character
##  Mean   :1.047                      Mean   : 6.518
##  3rd Qu.:1.000                      3rd Qu.: 7.000
##  Max.   :3.000                      Max.   :14.000
##
##    Fireplaces     GarageType          GarageArea      GarageCond
##  Min.   :0.000   Length:1460        Min.   :   0.0   Length:1460
##  1st Qu.:0.000   Class :character   1st Qu.: 334.5   Class :character
##  Median :1.000   Mode  :character   Median : 480.0   Mode  :character
##  Mean   :0.613                      Mean   : 473.0
##  3rd Qu.:1.000                      3rd Qu.: 576.0
##  Max.   :3.000                      Max.   :1418.0
##
##   PavedDrive           PoolArea          PoolQC             Fence
##  Length:1460        Min.   :  0.000   Length:1460        Length:1460
##  Class :character   1st Qu.:  0.000   Class :character   Class :character
##  Mode  :character   Median :  0.000   Mode  :character   Mode  :character
##                     Mean   :  2.759
##                     3rd Qu.:  0.000
##                     Max.   :738.000
##
##  MiscFeature           MiscVal             MoSold           YrSold
##  Length:1460        Min.   :    0.00   Min.   : 1.000   Min.   :2006
##  Class :character   1st Qu.:    0.00   1st Qu.: 5.000   1st Qu.:2007
##  Mode  :character   Median :    0.00   Median : 6.000   Median :2008
##                     Mean   :   43.49   Mean   : 6.322   Mean   :2008
##                     3rd Qu.:    0.00   3rd Qu.: 8.000   3rd Qu.:2009
##                     Max.   :15500.00   Max.   :12.000   Max.   :2010
##
##    SaleType         SaleCondition        SalePrice
##  Length:1460        Length:1460        Min.   : 34900
```

```
##  Class :character   Class :character   1st Qu.:129975
##  Mode  :character   Mode  :character   Median :163000
##                                        Mean   :180921
##                                        3rd Qu.:214000
##                                        Max.   :755000
##
```

```
#having a breif look at the number of levels per variable
sapply(house.data,function(x) length(unique(x)))
```

```
##            Id   LotFrontage       LotArea        Street         Alley
##          1460           111          1073             2             3
##     Utilities     LotConfig  Neighborhood    Condition1    Condition2
##             2             5            25             9             8
##      BldgType    HouseStyle   OverallQual   OverallCond     YearBuilt
##             5             8            10             9           112
##     RoofStyle      RoofMatl   Exterior1st    MasVnrArea     ExterQual
##             6             8            15           328             4
##     ExterCond    Foundation      BsmtQual      BsmtCond   TotalBsmtSF
##             5             6             5             5           721
##       Heating     X1stFlrSF     X2ndFlrSF  LowQualFinSF     GrLivArea
##             6           753           417            24           861
##      FullBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##             4             8             4             4            12
##    Functional    Fireplaces    GarageType    GarageArea    GarageCond
##             7             4             7           441             6
##    PavedDrive      PoolArea        PoolQC         Fence   MiscFeature
##             3             8             4             5             5
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##            21            12             5             9             6
##     SalePrice
##           663
```

```
#Going though the variables for cleaning----

#firstly, we have alot of character columns which we need to convert to
factors for analysis
#we also have some categorical variables which we would want to make a table
with (to view obs per level)
#and we want to check some of the categorical variables for correct
representation

#We should remove the Id column since this won't provide us with any
analytical insight outside of the data set
#(ID is unique per row whilst not being a continuous variable)
house.data$Id = NULL

#this is a continuous variable which we need to check since it has missing
values
```

```
summary(house.data$LotFrontage) #we have NA's which in this context, can be
treated as 0
```

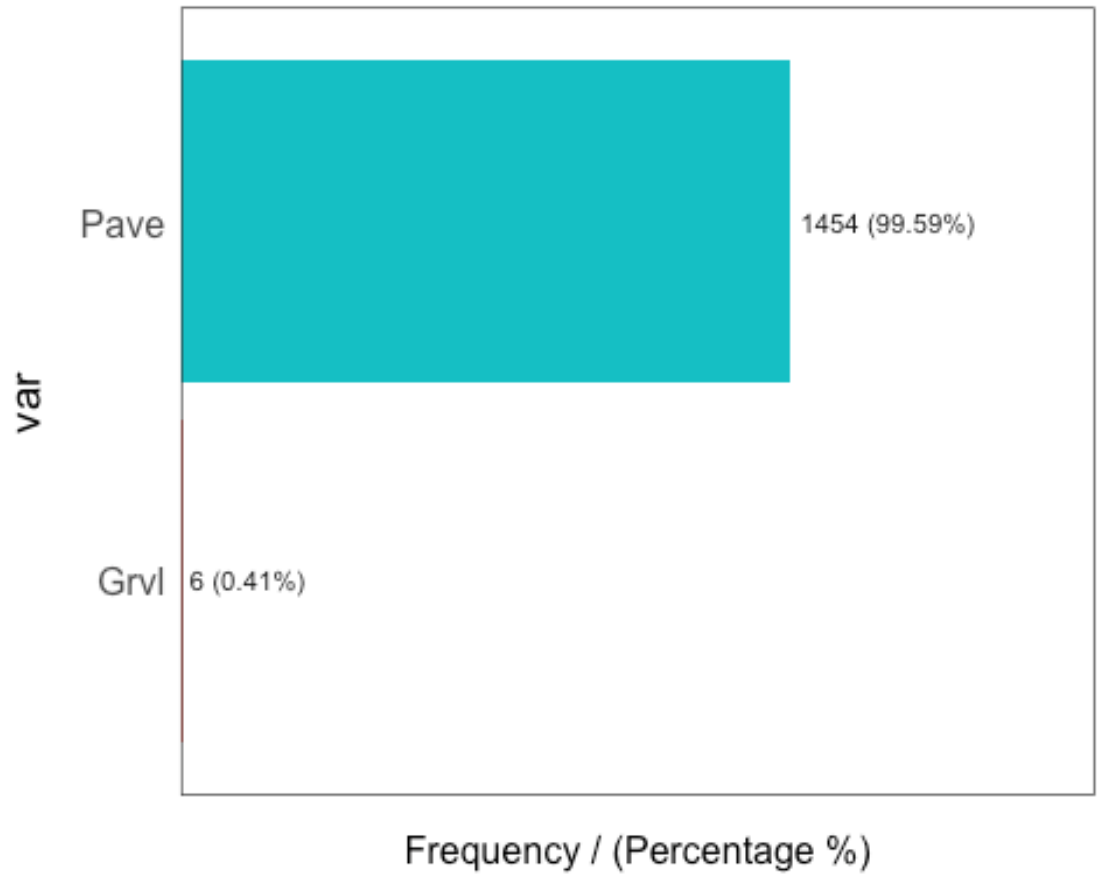
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   21.00   59.00   69.00   70.05   80.00  313.00     259
```

```
house.data$LotFrontage[is.na(house.data$LotFrontage)] = 0

house.data$Street = as.factor(house.data$Street)
summary(house.data$Street)
```

```
## Grvl Pave
##    6 1454
```

```
freq(house.data$Street) #Nearly all 'Pave', the variable isn't useful for
modelling, should be removed
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



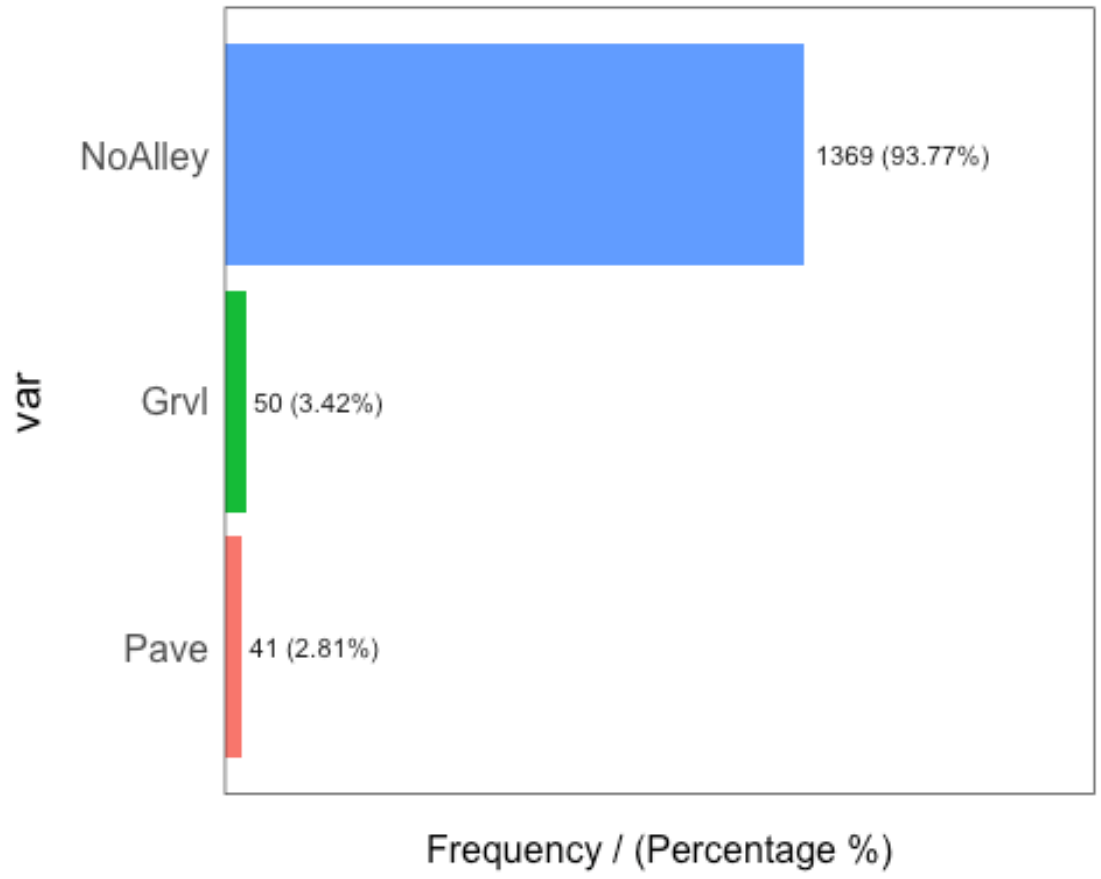

```
##    var frequency percentage cumulative_perc
## 1 Pave      1454      99.59           99.59
## 2 Grvl         6       0.41          100.00
```

```
house.data$Street = NULL

house.data$Alley[is.na(house.data$Alley)] = 'NoAlley'
house.data$Alley = as.factor(house.data$Alley) #NA in this column actually
means no alley access
summary(house.data$Alley)
```

```
##    Grvl NoAlley    Pave
##      50    1369      41
```

```
freq(house.data$Alley)
```

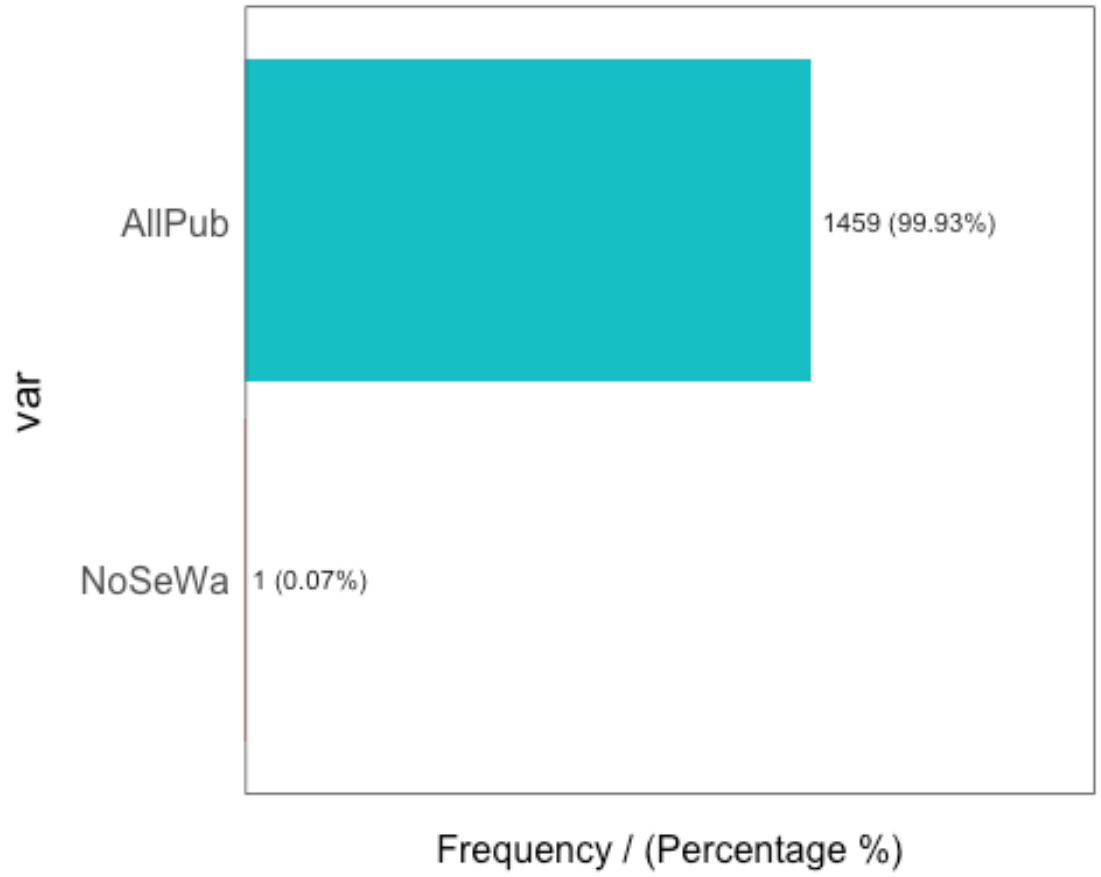
```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##       var frequency percentage cumulative_perc
## 1 NoAlley      1369      93.77           93.77
## 2    Grvl        50       3.42           97.19
## 3    Pave        41       2.81          100.00
```

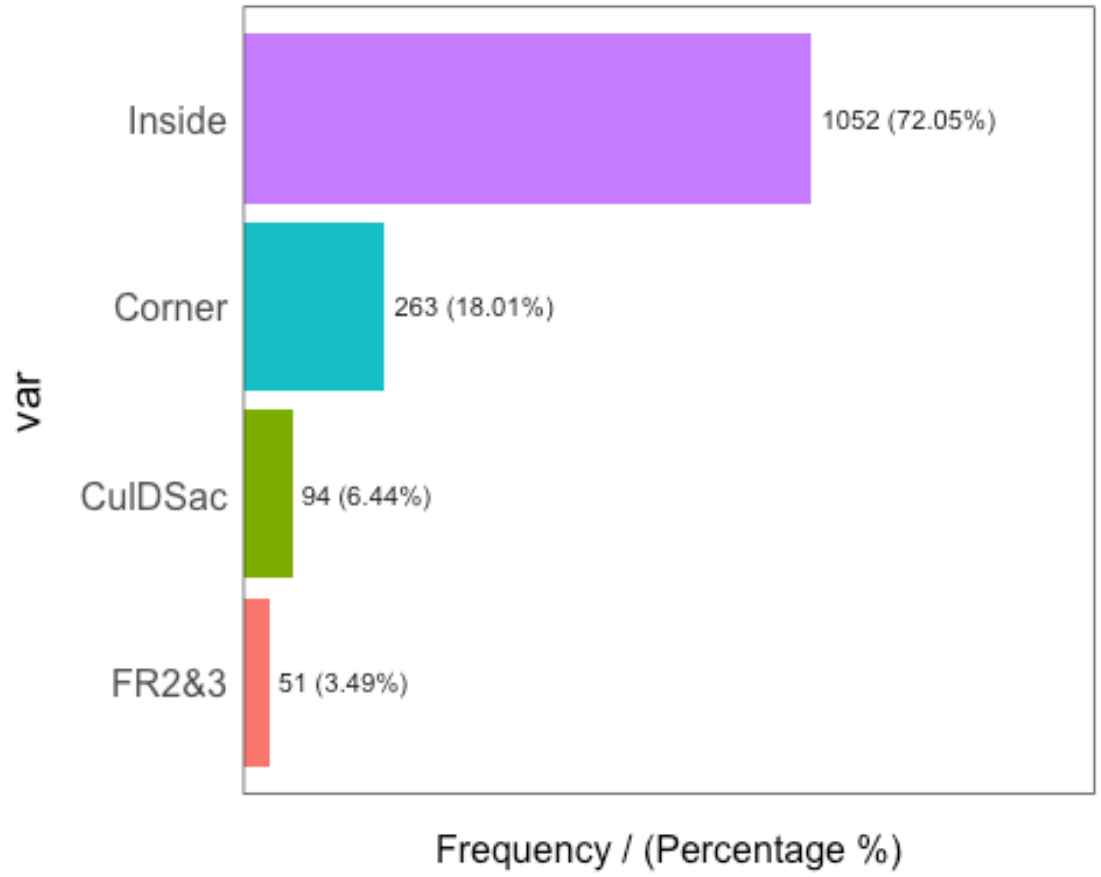
```
house.data$Utilities = as.factor(house.data$Utilities)
summary(house.data$Utilities) #only 1 NoSewa observation, not useful for
modelling, we remove
```

```
## AllPub NoSeWa
##   1459      1

freq(house.data$Utilities)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



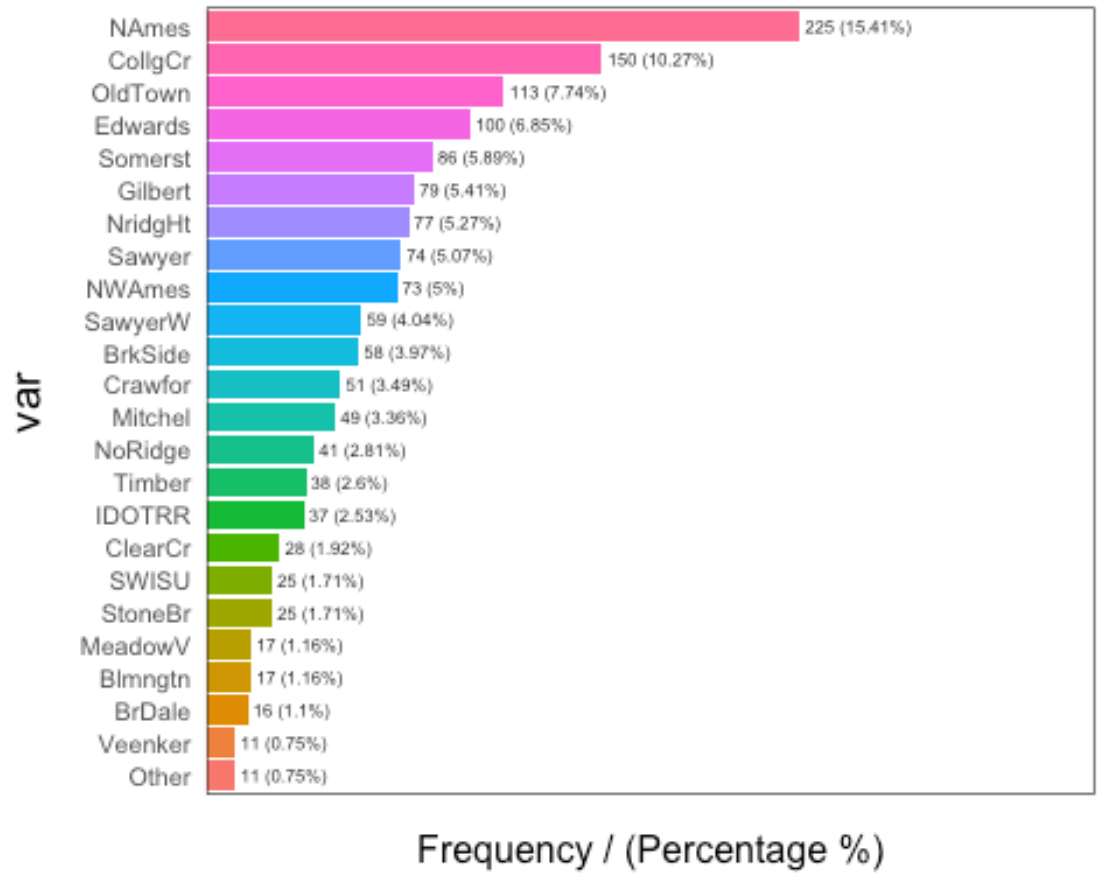

```
##      var frequency percentage cumulative_perc
## 1 AllPub      1459      99.93           99.93
## 2 NoSeWa         1       0.07          100.00

house.data$Utilities = NULL

house.data$LotConfig = as.factor(house.data$LotConfig)
summary(house.data$LotConfig) #FR3 has low observation, we should cobine with
FR2

##  Corner CulDSac     FR2     FR3  Inside
##     263      94      47       4    1052

levels(house.data$LotConfig) = c("Corner","CulDSac","FR2&3","FR2&3","Inside")
summary(house.data$LotConfig)
```

```
##  Corner CulDSac   FR2&3  Inside
##     263      94      51    1052

freq(house.data$LotConfig)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



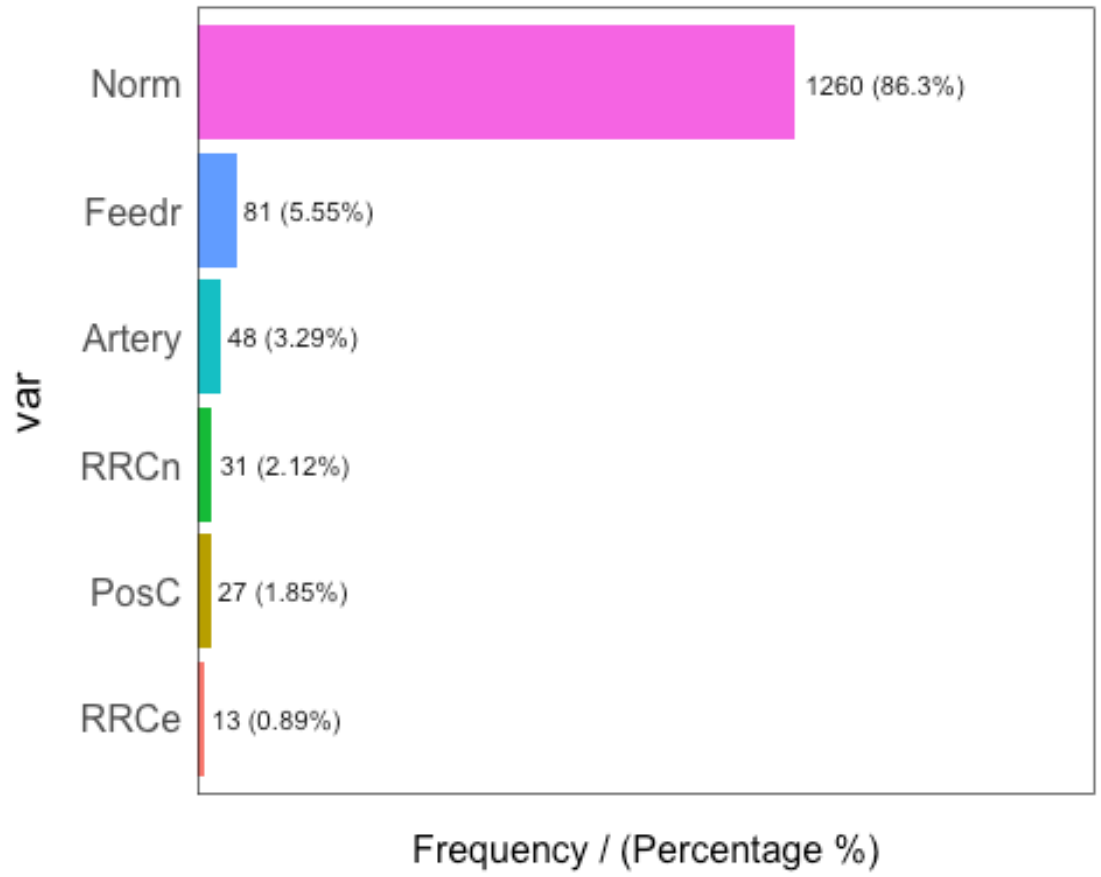

```
##       var frequency percentage cumulative_perc
## 1  Inside      1052      72.05           72.05
## 2  Corner       263      18.01           90.06
## 3 CulDSac        94       6.44           96.50
## 4   FR2&3        51       3.49          100.00

house.data$Neighborhood = as.factor(house.data$Neighborhood)
summary(house.data$Neighborhood) #We should merge Blueste and NPkVill into an
'Other' column

## Blmngtn Blueste  BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert
IDOTRR
##      17       2      16      58      28     150      51     100      79
37
## MeadowV Mitchel   NAmes NoRidge NPkVill NridgHt  NWAmes OldTown  Sawyer
```

```
SawyerW
##      17      49     225      41       9      77      73     113      74
59
## Somerst StoneBr   SWISU  Timber Veenker
##      86      25      25      38      11

levels(house.data$Neighborhood) =
c("Blmngtn","Other","BrDale","BrkSide","ClearCr","CollgCr","Crawfor","Edwards
","Gilbert","IDOTRR","MeadowV","Mitchel","NAmes","NoRidge","Other","NridgHt",
"NWAmes","OldTown","Sawyer","SawyerW","Somerst","StoneBr","SWISU","Timber","V
eenker")
summary(house.data$Neighborhood)

## Blmngtn   Other  BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert
IDOTRR
##      17      11      16      58      28     150      51     100      79
37
## MeadowV Mitchel   NAmes NoRidge NridgHt  NWAmes OldTown  Sawyer SawyerW
Somerst
##      17      49     225      41      77      73     113      74      59
86
## StoneBr   SWISU  Timber Veenker
##      25      25      38      11

freq(house.data$Neighborhood)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##         var frequency percentage cumulative_perc
## 1     NAmes       225      15.41           15.41
## 2   CollgCr       150      10.27           25.68
## 3   OldTown       113       7.74           33.42
## 4   Edwards       100       6.85           40.27
## 5   Somerst        86       5.89           46.16
## 6   Gilbert        79       5.41           51.57
## 7   NridgHt        77       5.27           56.84
## 8    Sawyer        74       5.07           61.91
## 9    NWAmes        73       5.00           66.91
## 10 SawyerW        59       4.04           70.95
## 11 BrkSide        58       3.97           74.92
## 12 Crawfor        51       3.49           78.41
## 13 Mitchel        49       3.36           81.77
## 14 NoRidge        41       2.81           84.58
## 15  Timber        38       2.60           87.18
## 16  IDOTRR        37       2.53           89.71
## 17 ClearCr        28       1.92           91.63
## 18 StoneBr        25       1.71           93.34
## 19   SWISU        25       1.71           95.05
## 20 Blmngtn        17       1.16           96.21
## 21 MeadowV        17       1.16           97.37
## 22  BrDale        16       1.10           98.47
```

```
## 23   Other        11       0.75           99.22
## 24 Veenker        11       0.75          100.00

house.data$Condition1 = as.factor(house.data$Condition1)
summary(house.data$Condition1) #some levels have less than 10 observations,
we should merge in the following way

## Artery  Feedr   Norm   PosA   PosN   RRAe   RRAn   RRNe   RRNn
##     48     81   1260      8     19     11     26      2      5

#PosA & PosN; Near of adjacent to positive off-site feature--park; PosC
#RRAe & RRNe; within 200' or adjecent to East-West Rail; RRCe
#RRAn & RRNn; within 200' or adjecent to North-South Rail; RRCn
levels(house.data$Condition1) =
c("Artery","Feedr","Norm","PosC","PosC","RRCe","RRCn","RRCe","RRCn")
summary(house.data$Condition1)

## Artery  Feedr   Norm   PosC   RRCe   RRCn
##     48     81   1260     27     13     31

freq(house.data$Condition1)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##      var frequency percentage cumulative_perc
## 1   Norm      1260      86.30           86.30
## 2  Feedr        81       5.55           91.85
## 3 Artery        48       3.29           95.14
## 4   RRCn        31       2.12           97.26
## 5   PosC        27       1.85           99.11
## 6   RRCe        13       0.89          100.00
```

```
house.data$Condition2 = as.factor(house.data$Condition2)
summary(house.data$Condition2) #all levels except Norm have low obs, turn
into binary column around 'Norm'
```

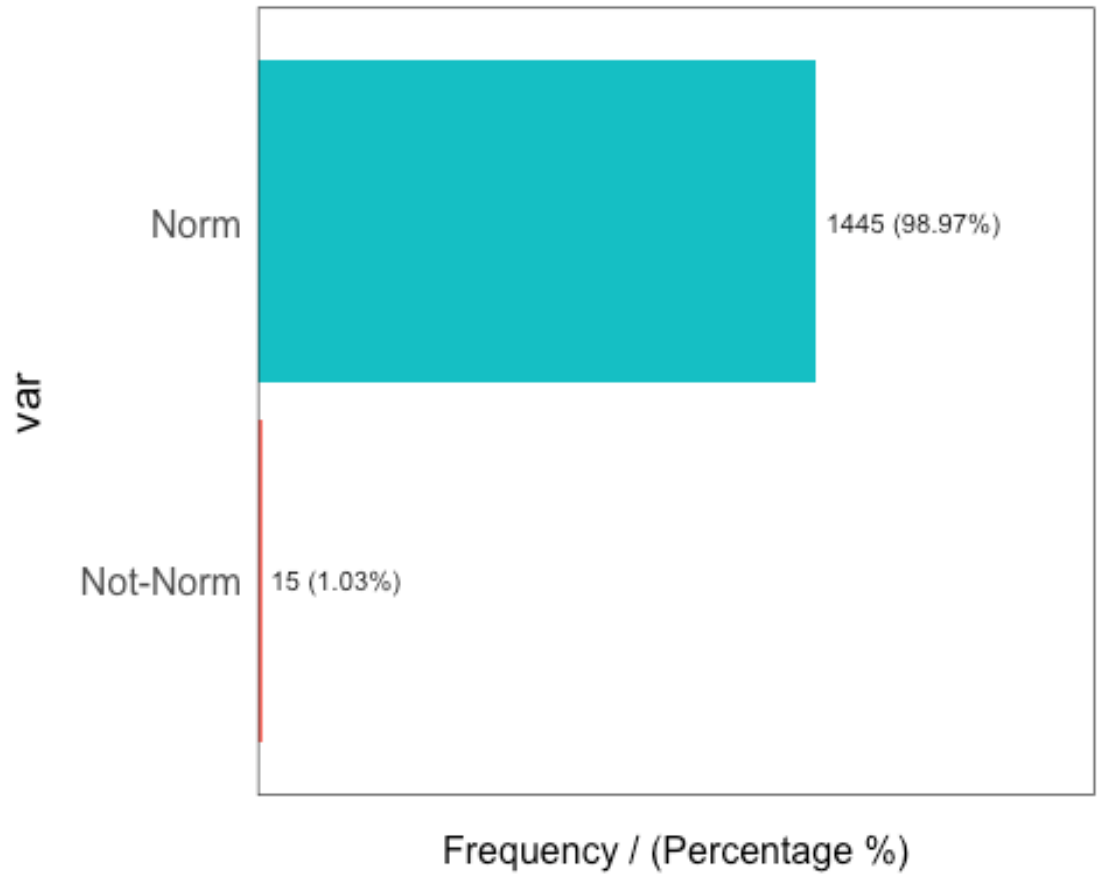
```
## Artery  Feedr   Norm   PosA   PosN   RRAe   RRAn   RRNn
##      2      6   1445      1      2      1      1      2
```

```
levels(house.data$Condition2) = c("Not-Norm","Not-Norm","Norm","Not-
Norm","Not-Norm","Not-Norm","Not-Norm","Not-Norm","Not-Norm")
summary(house.data$Condition2)
```

```
## Not-Norm     Norm
##       15     1445
```

```
freq(house.data$Condition2)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



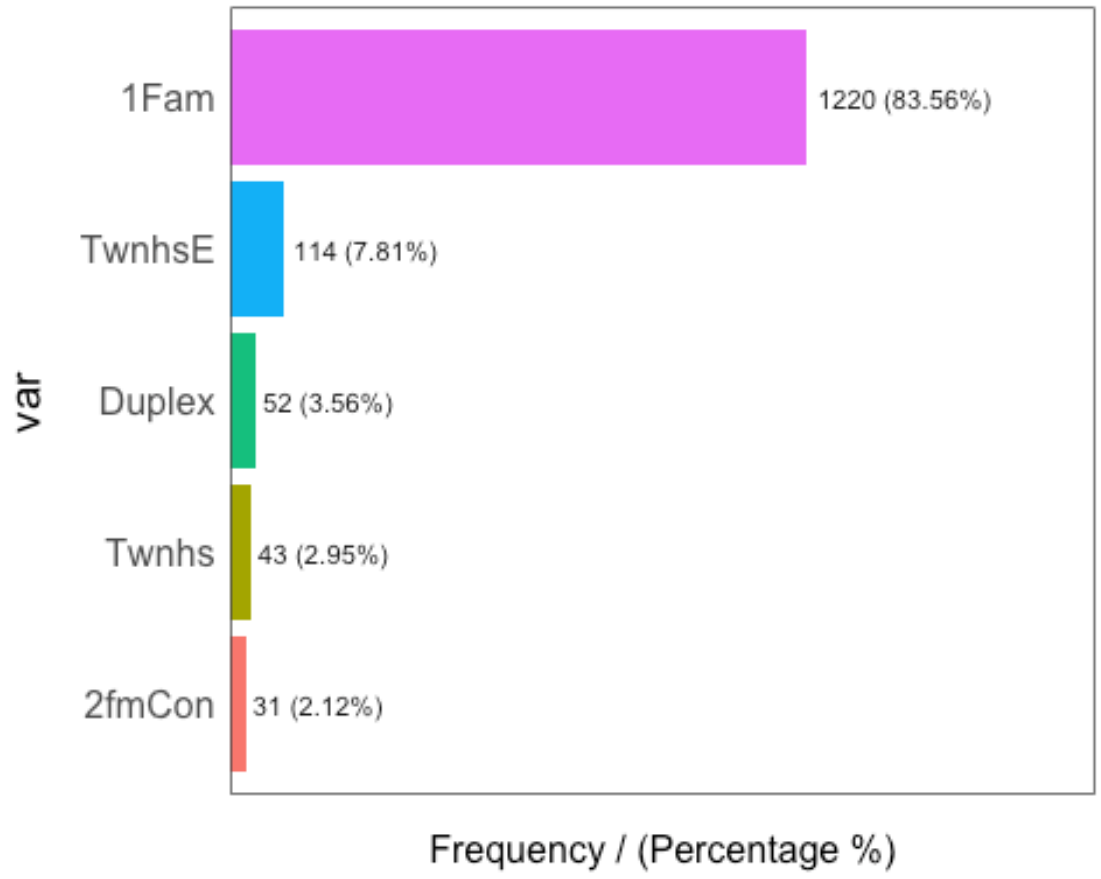

```
##        var frequency percentage cumulative_perc
## 1     Norm      1445      98.97           98.97
## 2 Not-Norm        15       1.03          100.00
```

```
house.data$BldgType = as.factor(house.data$BldgType)
summary(house.data$BldgType) #a useful variable! No NA's and good
representation per level
```

```
##   1Fam 2fmCon Duplex  Twnhs TwnhsE
##   1220     31     52     43    114
```

```
freq(house.data$BldgType)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##      var frequency percentage cumulative_perc
## 1   1Fam      1220      83.56           83.56
## 2 TwnhsE       114       7.81           91.37
## 3 Duplex        52       3.56           94.93
## 4  Twnhs        43       2.95           97.88
## 5 2fmCon        31       2.12          100.00
```

```
house.data$HouseStyle = as.factor(house.data$HouseStyle)
summary(house.data$HouseStyle) #we merge '2.5Fin' and '2.5Unf' into one level for '2.5All'
```

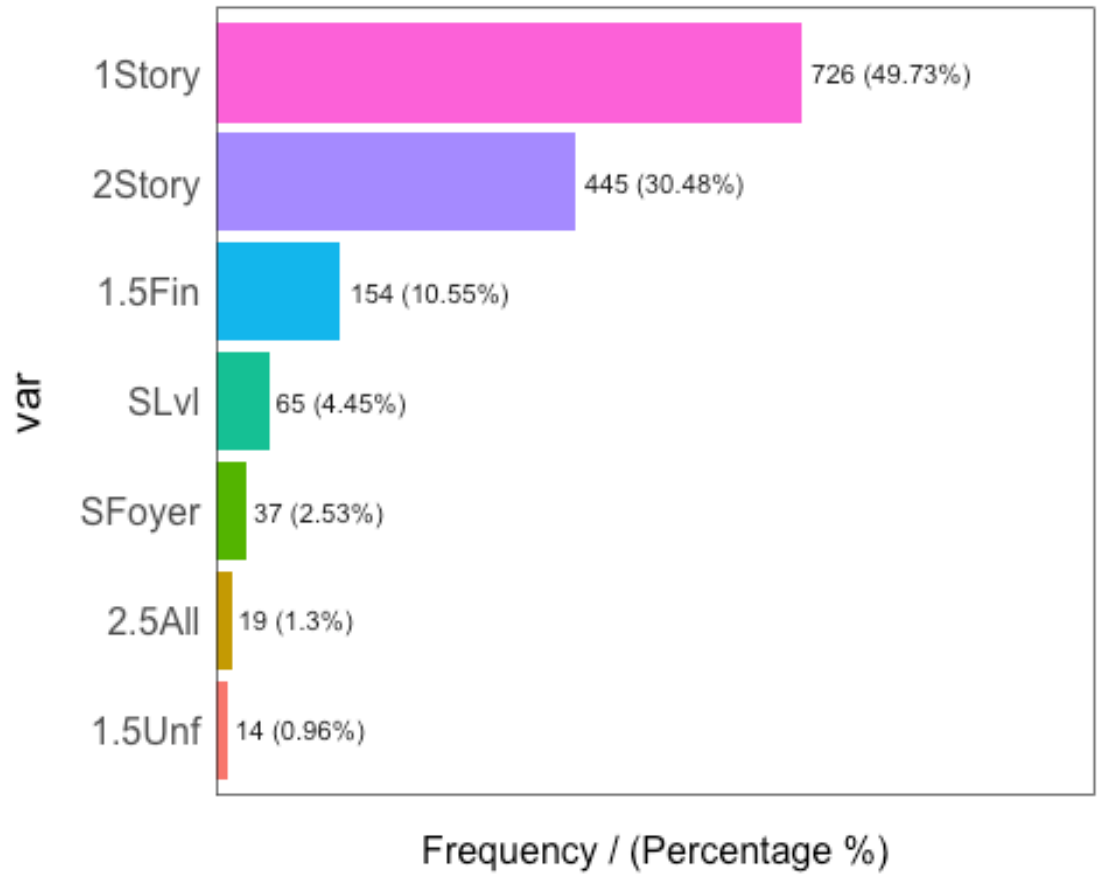
```
## 1.5Fin 1.5Unf 1Story 2.5Fin 2.5Unf 2Story SFoyer   SLvl
##    154     14    726      8     11    445     37     65
```

```
levels(house.data$HouseStyle) =
c("1.5Fin","1.5Unf","1Story","2.5All","2.5All","2Story","SFoyer","SLvl")
summary(house.data$HouseStyle)
```

```
## 1.5Fin 1.5Unf 1Story 2.5All 2Story SFoyer   SLvl
##    154     14    726     19    445     37     65
```

```
freq(house.data$HouseStyle)
```

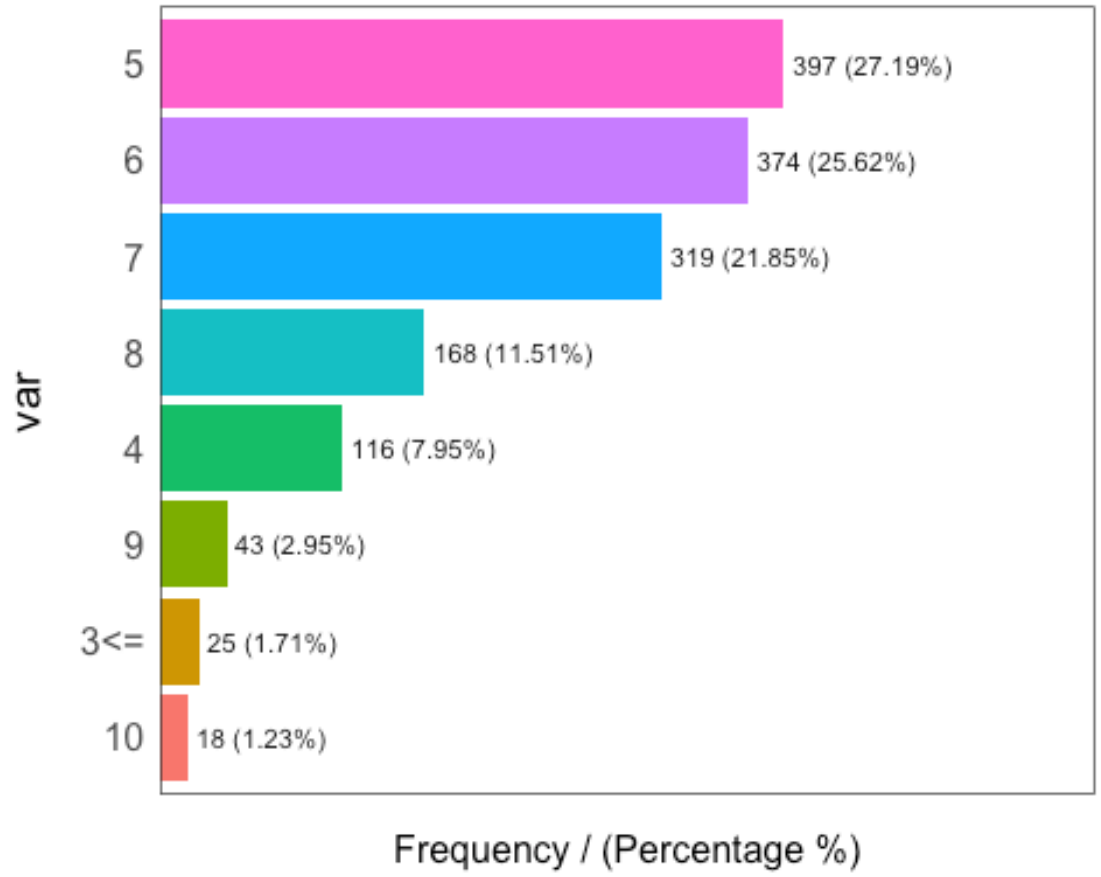
```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##      var frequency percentage cumulative_perc
## 1 1Story       726      49.73           49.73
## 2 2Story       445      30.48           80.21
## 3 1.5Fin       154      10.55           90.76
## 4   SLvl        65       4.45           95.21
## 5 SFoyer        37       2.53           97.74
## 6 2.5All        19       1.30           99.04
## 7 1.5Unf        14       0.96          100.00
```

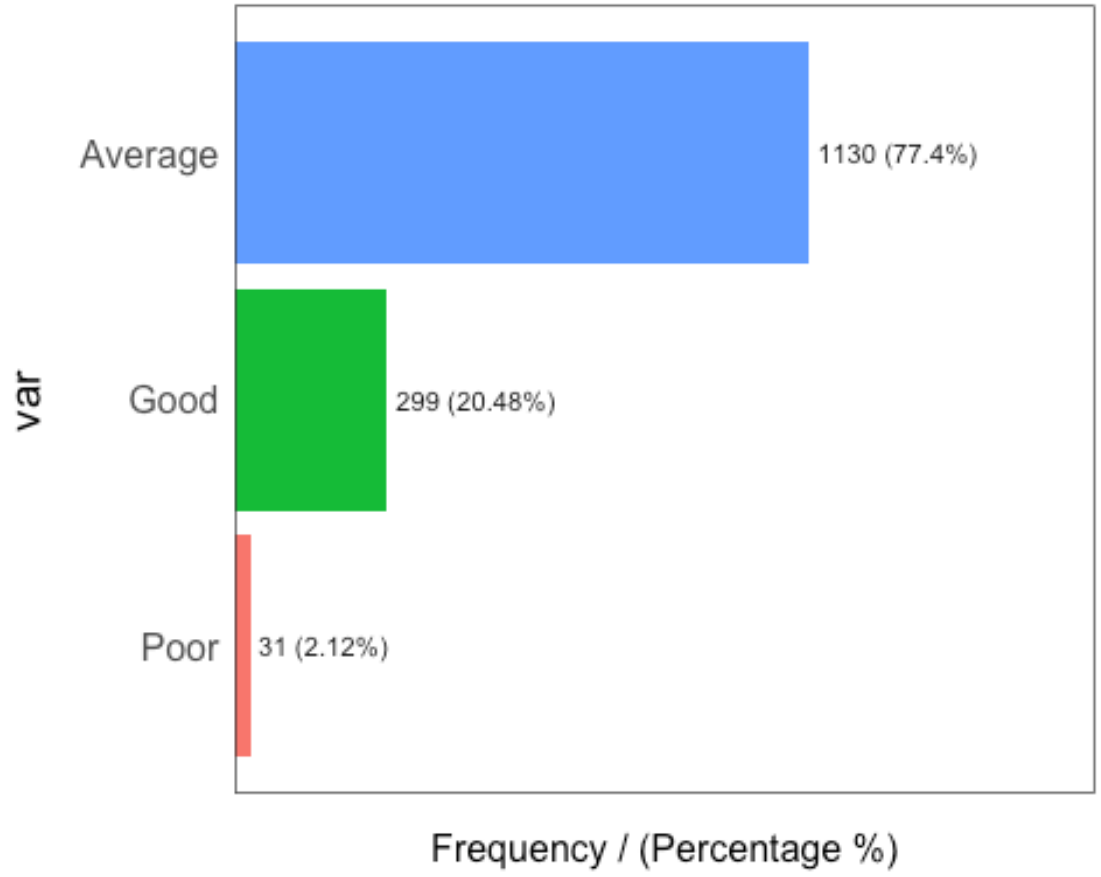
```
###Make this continous###
house.data$OverallQual = as.factor(house.data$OverallQual)
summary(house.data$OverallQual) #We should merge 1,2 into 3 and make new
level 3 or lower

##   1   2   3   4   5   6   7   8   9  10
##   2   3  20 116 397 374 319 168  43  18

levels(house.data$OverallQual) =
c("3<=","3<=","3<=","4","5","6","7","8","9","10")
summary(house.data$OverallQual)
```

```
## 3<=   4   5   6   7   8   9  10
##  25 116 397 374 319 168  43  18

freq(house.data$OverallQual)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



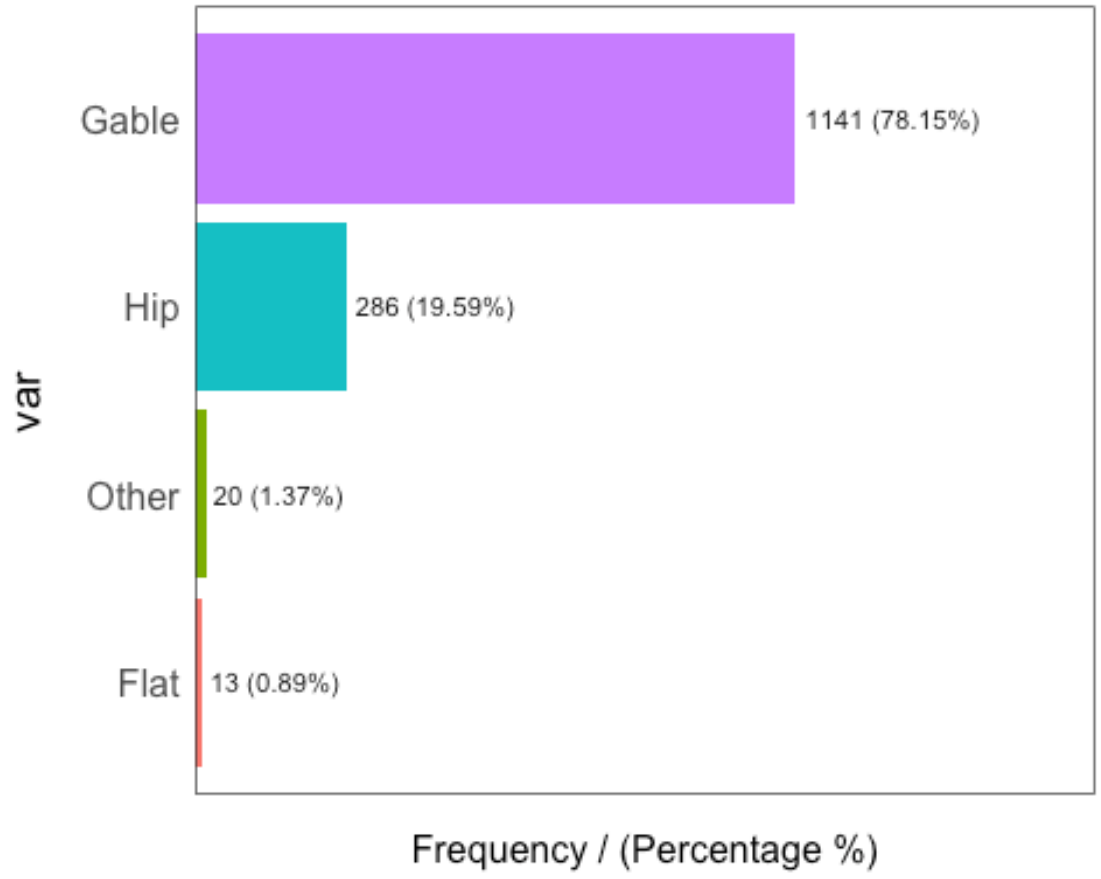

```
##   var frequency percentage cumulative_perc
## 1   5       397      27.19           27.19
## 2   6       374      25.62           52.81
## 3   7       319      21.85           74.66
## 4   8       168      11.51           86.17
## 5   4       116       7.95           94.12
## 6   9        43       2.95           97.07
## 7 3<=        25       1.71           98.78
## 8  10        18       1.23          100.00

#Here we have OverallCond, we can convert in prep for question 2
house.data$OverallCond = as.factor(ifelse(house.data$OverallCond >= 1 &
house.data$OverallCond <= 3, 'Poor',
                                          ifelse(house.data$OverallCond >= 4
```

```
& house.data$OverallCond <= 6, 'Average', 'Good')))
summary(house.data$OverallCond)

## Average    Good    Poor
##    1130     299      31

freq(house.data$OverallCond)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##       var frequency percentage cumulative_perc
## 1 Average      1130      77.40           77.40
## 2    Good       299      20.48           97.88
## 3    Poor        31       2.12          100.00

house.data$RoofStyle = as.factor(house.data$RoofStyle)
summary(house.data$RoofStyle) #some levels have less than 10 observations, we
should make a category of 'other'

##    Flat   Gable Gambrel     Hip Mansard    Shed
##      13    1141      11     286       7       2
```

```
levels(house.data$RoofStyle) =
c("Flat","Gable","Other","Hip","Other","Other")
summary(house.data$RoofStyle)

##  Flat Gable Other   Hip
##    13  1141    20   286

freq(house.data$RoofStyle)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##     var frequency percentage cumulative_perc
## 1 Gable      1141      78.15           78.15
## 2   Hip       286      19.59           97.74
## 3 Other        20       1.37           99.11
## 4  Flat        13       0.89          100.00

house.data$RoofMatl = as.factor(house.data$RoofMatl)
summary(house.data$RoofMatl) #need to consider merging, turn into CompShg &
not CompShg variable
```

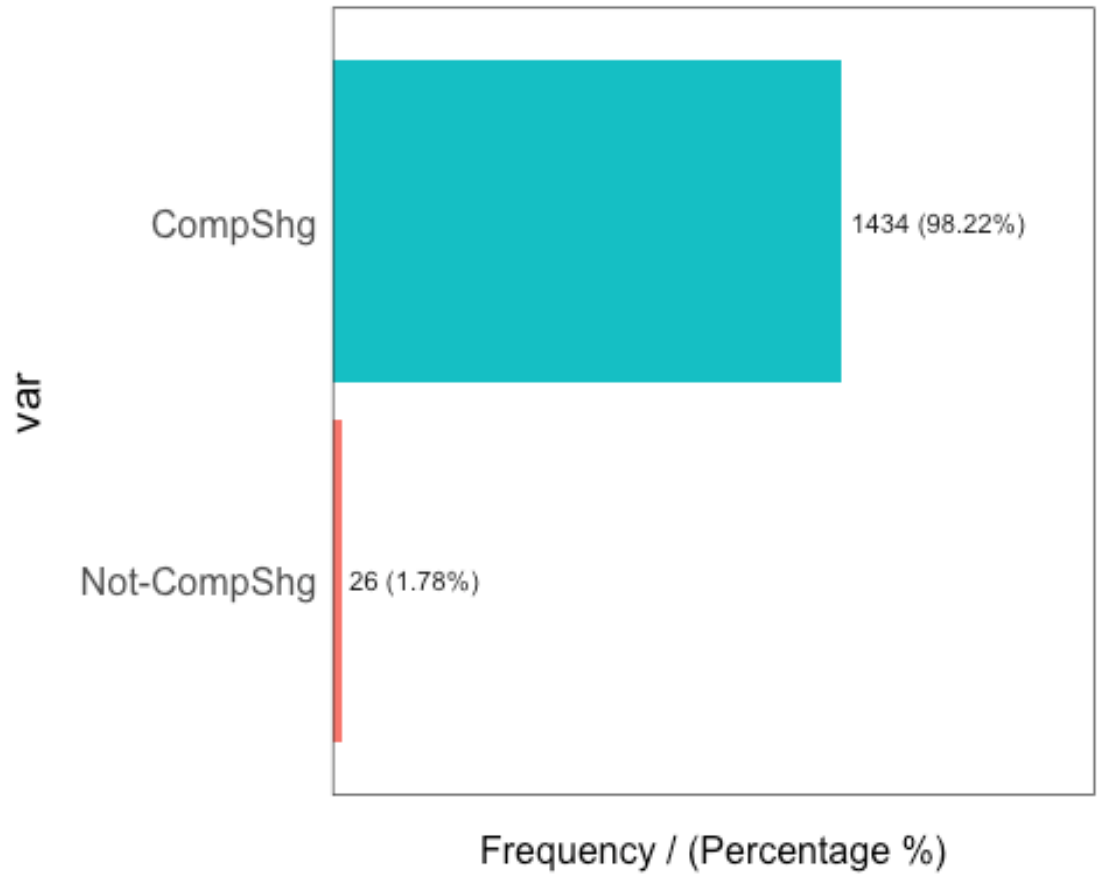
```
## ClyTile CompShg Membran   Metal    Roll Tar&Grv WdShake WdShngl
##       1    1434       1       1       1      11       5       6
```

```
levels(house.data$RoofMatl) = c("Not-CompShg","CompShg","Not-CompShg","Not-CompShg","Not-CompShg","Not-CompShg","Not-CompShg","Not-CompShg")
summary(house.data$RoofMatl)
```

```
## Not-CompShg     CompShg
##          26        1434
```

```
freq(house.data$RoofMatl)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##           var frequency percentage cumulative_perc
## 1     CompShg      1434      98.22           98.22
## 2 Not-CompShg        26       1.78          100.00
```

```
house.data$Exterior1st = as.factor(house.data$Exterior1st)
summary(house.data$Exterior1st) #some levels have less than 10 observations, need to merge in the following way
```

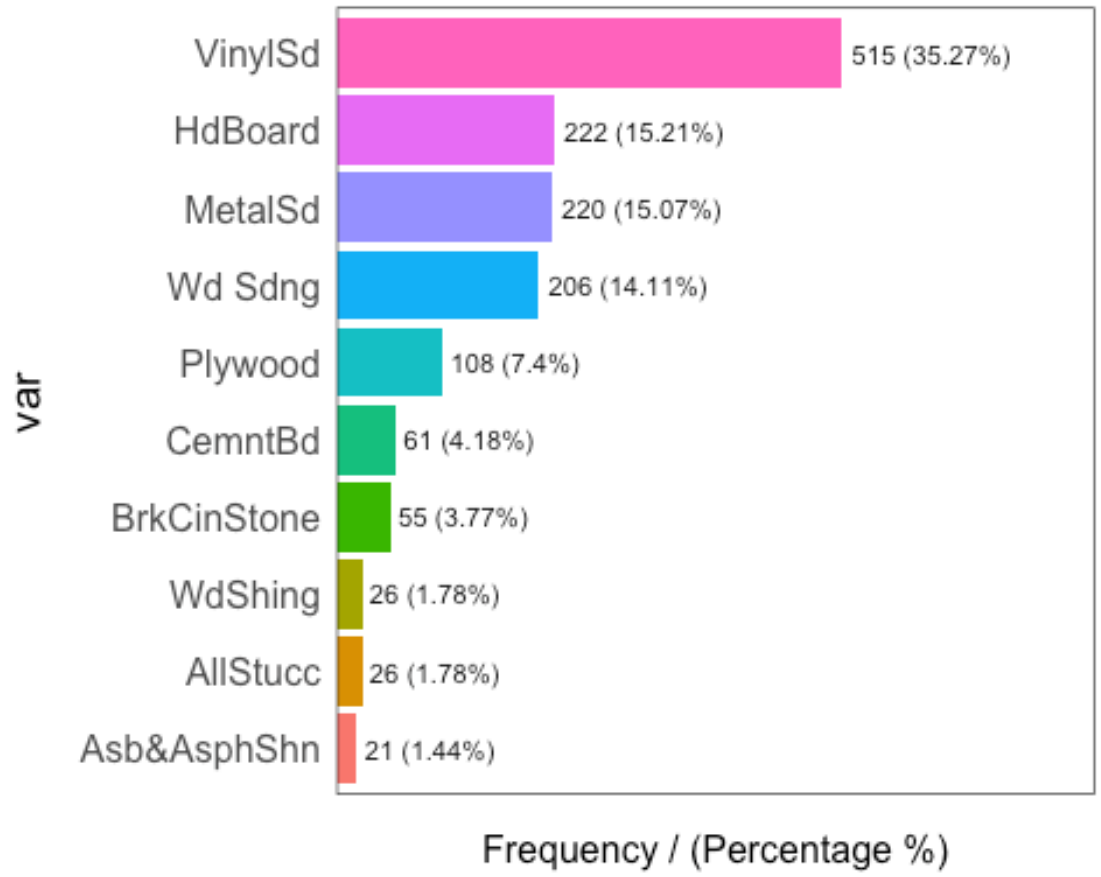
```
## AsbShng AsphShn BrkComm BrkFace  CBlock CemntBd HdBoard ImStucc MetalSd
Plywood
##      20       1       2      50       1      61     222       1     220
108
##   Stone  Stucco VinylSd Wd Sdng WdShing
##       2      25     515     206      26
```

```
#AsbShng & AsphShn; Asbestos or Asphalt Shingles; Asb&AsphShn
#BrkComm, BrkFace, CBlock & Stone; Brick, Cinder or Stone; BrkCinStone
#ImStucc & Stucco; Imitation Stucco and Stucco: AllStucc
levels(house.data$Exterior1st) =
c("Asb&AsphShn","Asb&AsphShn","BrkCinStone","BrkCinStone","BrkCinStone","Cemn
tBd","HdBoard","AllStucc","MetalSd","Plywood","BrkCinStone","AllStucc","Vinyl
Sd","Wd Sdng","WdShing")
summary(house.data$Exterior1st)
```

```
## Asb&AsphShn BrkCinStone     CemntBd     HdBoard    AllStucc     MetalSd
##          21          55          61         222          26         220
##     Plywood     VinylSd     Wd Sdng     WdShing
##         108         515         206          26
```

```
freq(house.data$Exterior1st)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



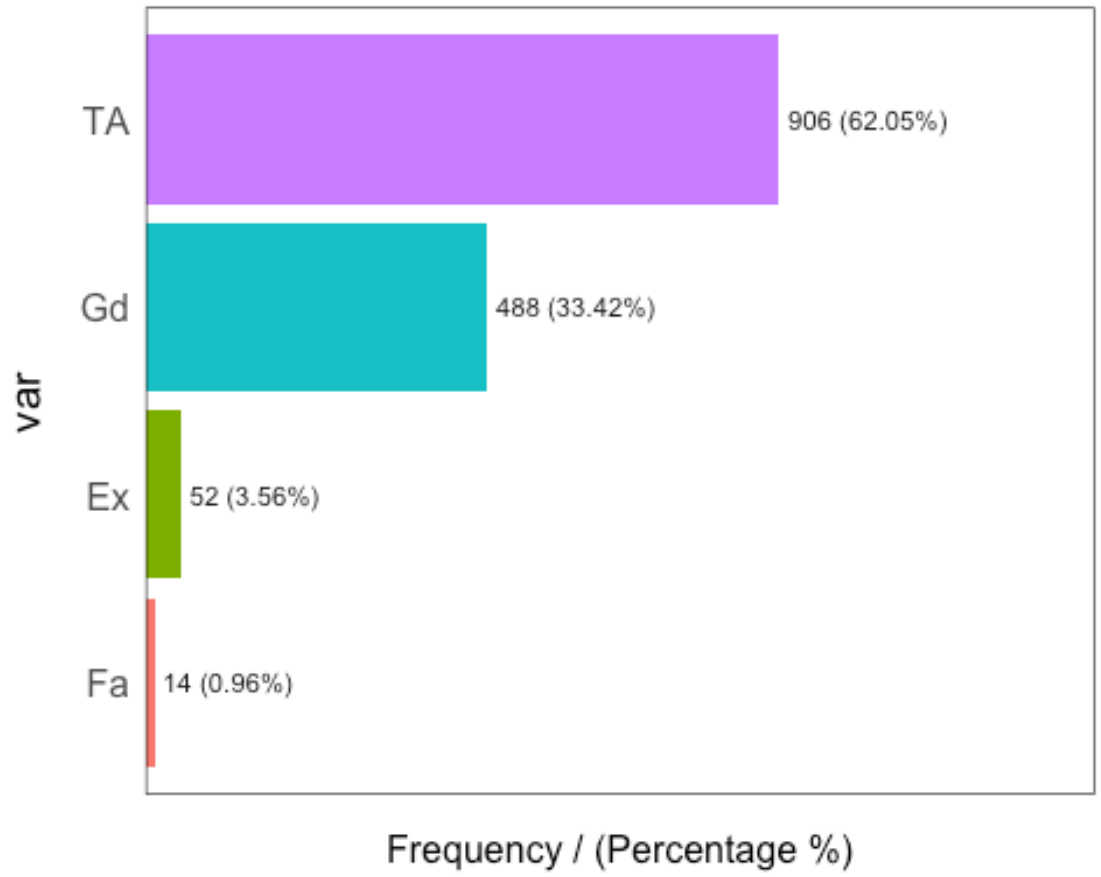

```
##            var frequency percentage cumulative_perc
## 1      VinylSd       515      35.27           35.27
## 2      HdBoard       222      15.21           50.48
## 3      MetalSd       220      15.07           65.55
## 4      Wd Sdng       206      14.11           79.66
## 5      Plywood       108       7.40           87.06
## 6      CemntBd        61       4.18           91.24
## 7  BrkCinStone        55       3.77           95.01
## 8     AllStucc        26       1.78           96.79
## 9      WdShing        26       1.78           98.57
## 10 Asb&AsphShn        21       1.44          100.00
```

```
house.data$ExterQual = as.factor(house.data$ExterQual)
summary(house.data$ExterQual) #a useful variable! No NA's and decent
representation per level
```

```
##  Ex  Fa  Gd  TA
##  52  14 488 906
```

```
freq(house.data$ExterQual)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##   var frequency percentage cumulative_perc
## 1  TA       906      62.05           62.05
## 2  Gd       488      33.42           95.47
## 3  Ex        52       3.56           99.03
## 4  Fa        14       0.96          100.00
```

```
house.data$ExterCond = as.factor(house.data$ExterCond)
summary(house.data$ExterCond) #some levels have less than 10 observations,
merge in the following way (maybe don't werge?)
```

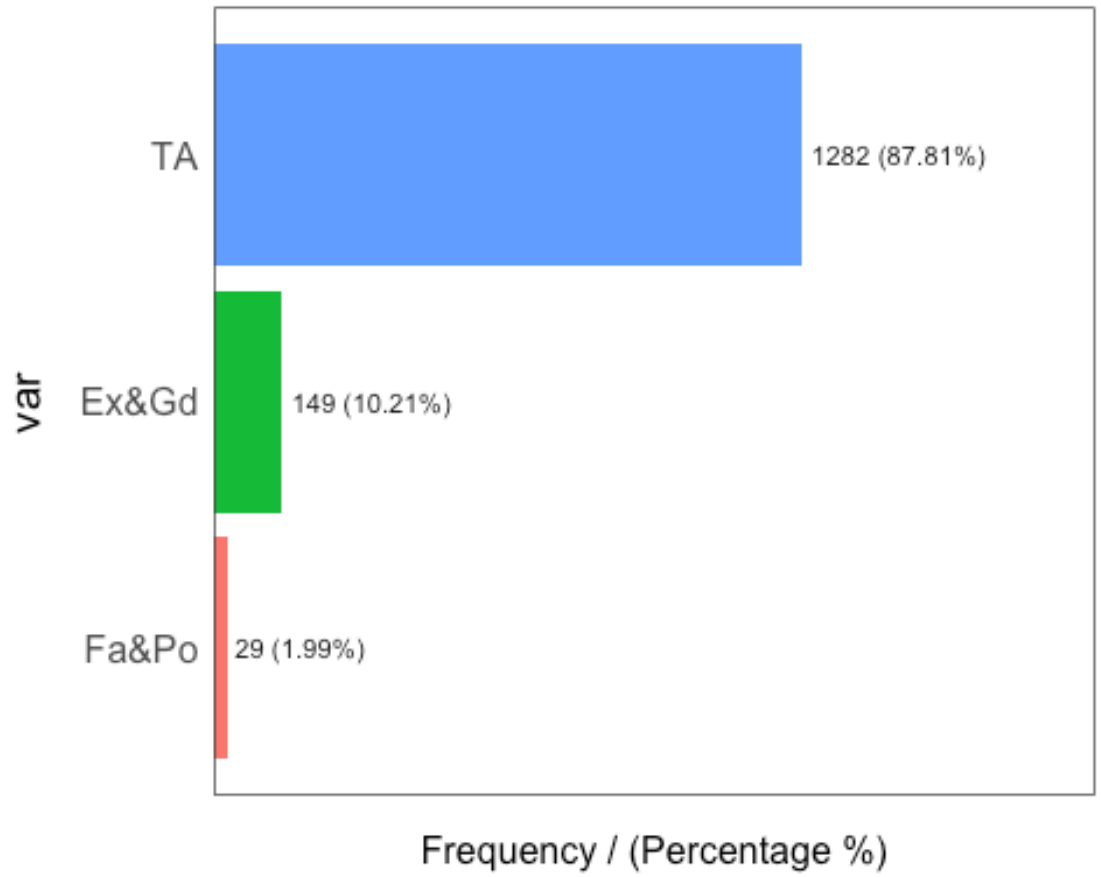
```
##   Ex   Fa   Gd   Po   TA
##    3   28  146    1 1282
```

```
#Ex & Gd; Good and Above; Ex&Gd
#Po & Fa; Fair and worse; Fa&Po
levels(house.data$ExterCond) = c("Ex&Gd","Fa&Po","Ex&Gd","Fa&Po","TA")
summary(house.data$ExterCond)
```

```
## Ex&Gd Fa&Po    TA
##   149    29  1282
```

```
freq(house.data$ExterCond)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##     var frequency percentage cumulative_perc
## 1    TA      1282      87.81           87.81
## 2 Ex&Gd       149      10.21           98.02
## 3 Fa&Po        29       1.99          100.00
```

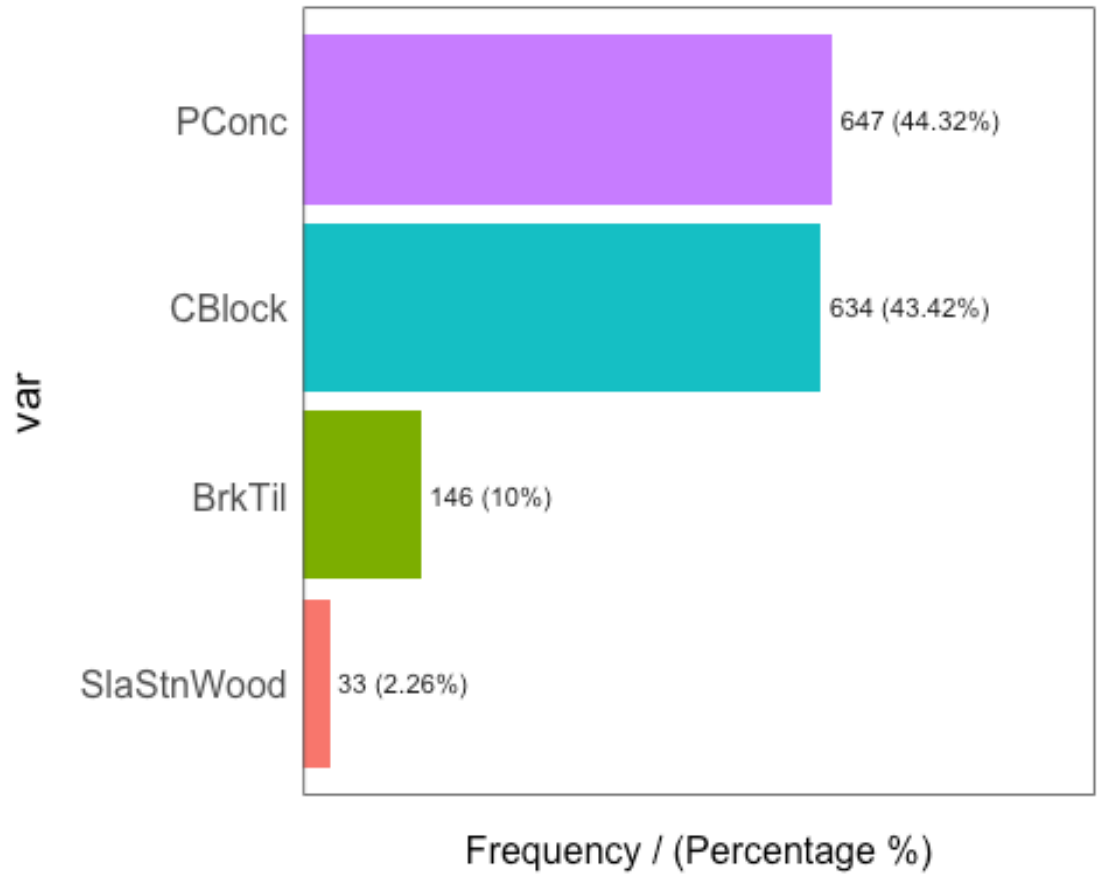
```
house.data$Foundation = as.factor(house.data$Foundation)
summary(house.data$Foundation) #some levels have less than 10 observations,
use the following merge
```

```
## BrkTil CBlock  PConc   Slab  Stone   Wood
##    146    634    647     24      6      3
```

```
#Slab, Stone & Wood; Slab, Stone Or Wood Foundation; SlaStnWood
levels(house.data$Foundation) =
c("BrkTil","CBlock","PConc","SlaStnWood","SlaStnWood","SlaStnWood")
summary(house.data$Foundation)
```

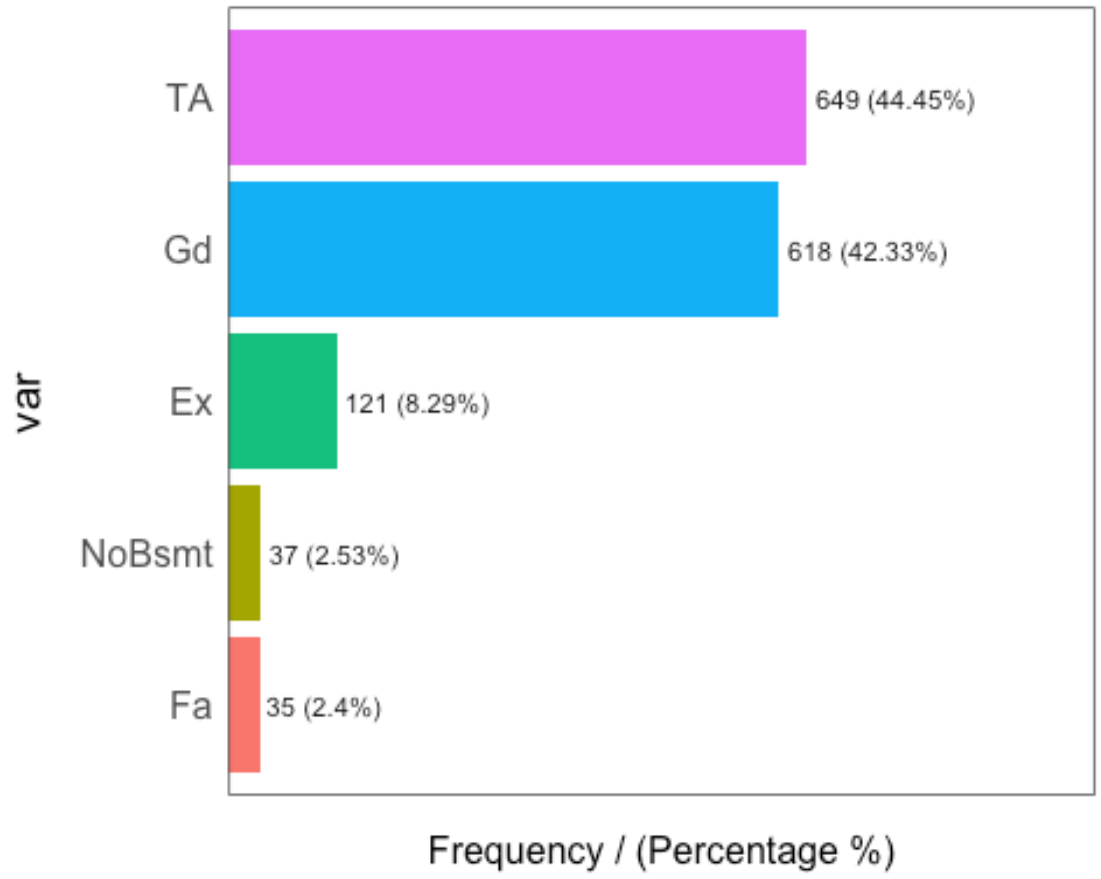
```
##     BrkTil     CBlock      PConc SlaStnWood
##        146        634        647         33
```

```
freq(house.data$Foundation)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##          var frequency percentage cumulative_perc
## 1      PConc       647      44.32           44.32
## 2     CBlock       634      43.42           87.74
## 3     BrkTil       146      10.00           97.74
## 4 SlaStnWood        33       2.26          100.00

house.data$BsmtQual[is.na(house.data$BsmtQual)] = 'NoBsmt' #NA values
actually mean no basement, not missing data
house.data$BsmtQual = as.factor(house.data$BsmtQual)
summary(house.data$BsmtQual) #a useful variable! good representation per
level

##     Ex     Fa     Gd NoBsmt     TA
##    121     35    618     37    649

freq(house.data$BsmtQual)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##      var frequency percentage cumulative_perc
## 1     TA       649      44.45           44.45
## 2     Gd       618      42.33           86.78
## 3     Ex       121       8.29           95.07
## 4 NoBsmt        37       2.53           97.60
## 5     Fa        35       2.40          100.00
```

```
house.data$BsmtCond[is.na(house.data$BsmtCond)] = 'NoBsmt' #NA values
actually mean no basement, not missing data
house.data$BsmtCond = as.factor(house.data$BsmtCond)
summary(house.data$BsmtCond) #'Po' has low observation, merge like the
following:
```

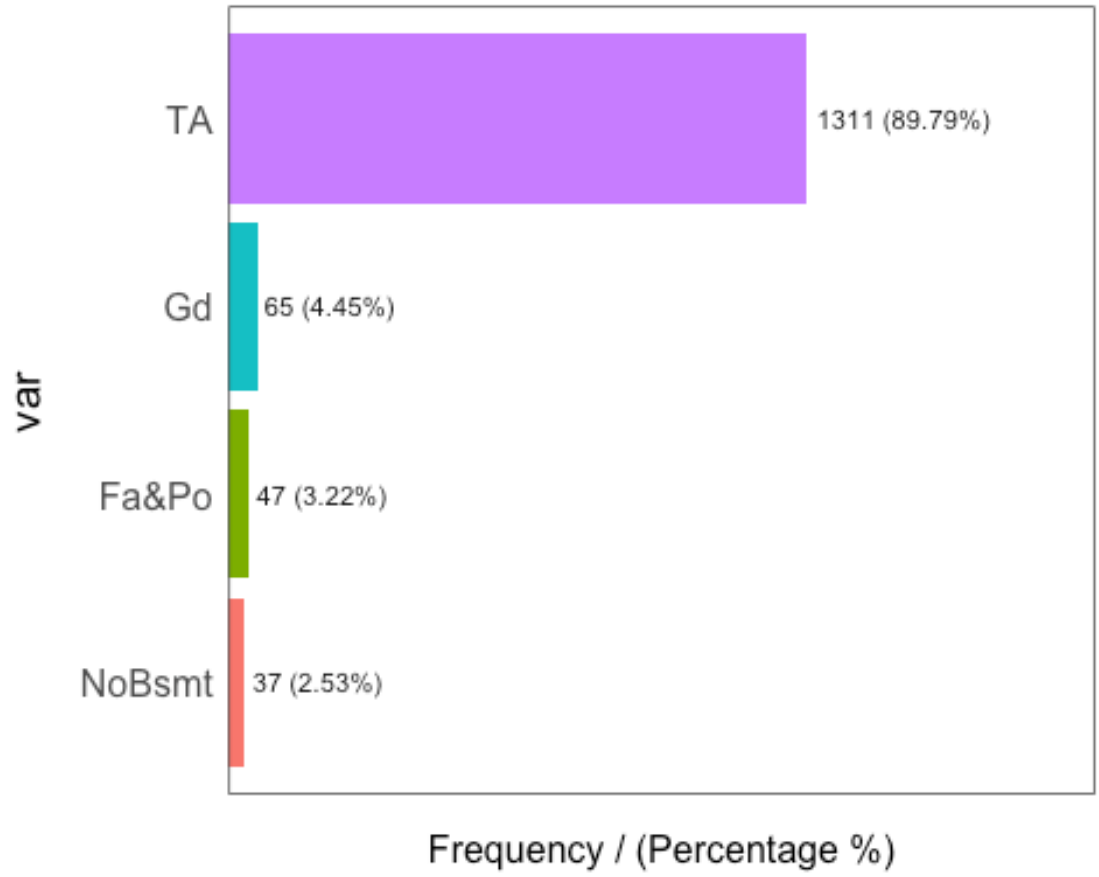
```
##     Fa     Gd NoBsmt     Po     TA
##     45     65     37      2   1311
```

```
#Fa & Po; Fair and Poor BsmtCond; Fa&Po
levels(house.data$BsmtCond) = c("Fa&Po","Gd","NoBsmt","Fa&Po","TA" )
summary(house.data$BsmtCond)
```

```
##  Fa&Po     Gd NoBsmt     TA
##     47     65     37   1311
```

```
freq(house.data$BsmtCond)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



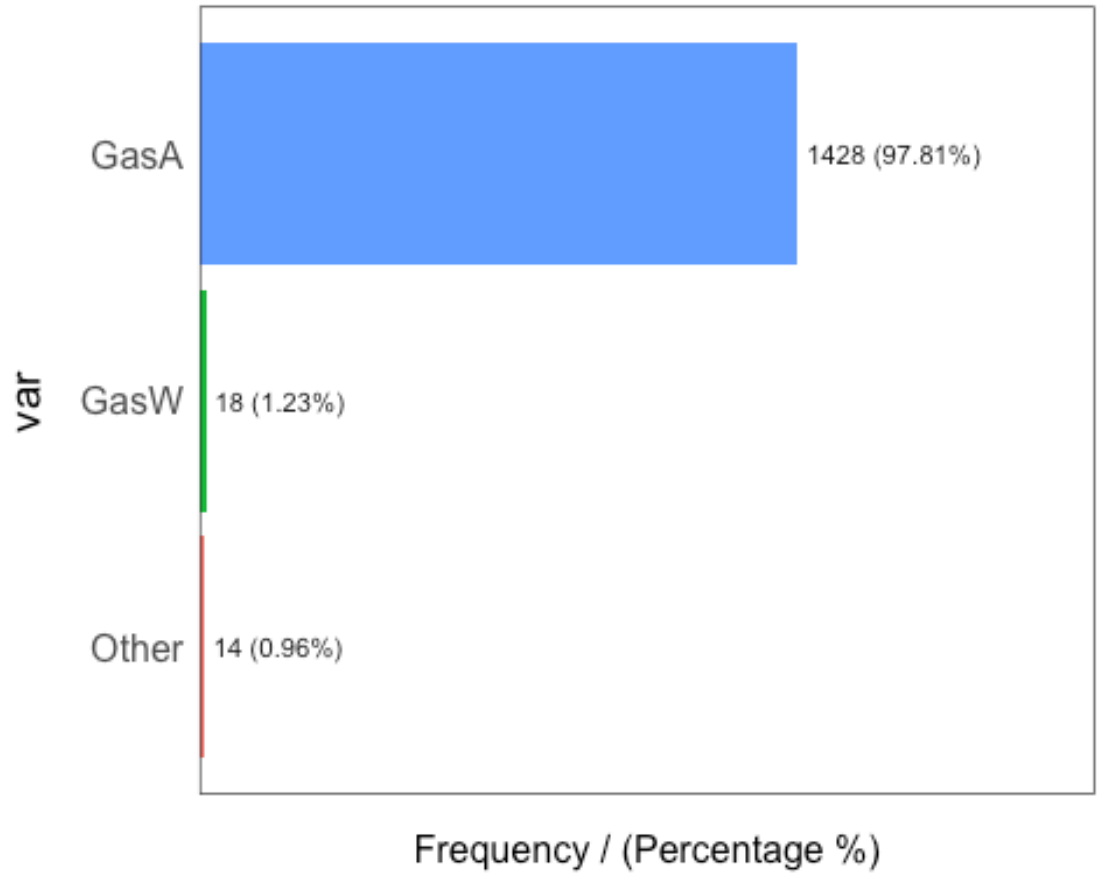

```
##      var frequency percentage cumulative_perc
## 1     TA      1311      89.79           89.79
## 2     Gd        65       4.45           94.24
## 3  Fa&Po        47       3.22           97.46
## 4 NoBsmt        37       2.53          100.00
```

```
house.data$Heating = as.factor(house.data$Heating)
summary(house.data$Heating) #we should put all other levels in 'Other'
```

```
## Floor  GasA  GasW  Grav  OthW  Wall
##     1  1428    18     7     2     4
```

```
levels(house.data$Heating) = c("Other","GasA","GasW","Other","Other","Other")
summary(house.data$Heating)
```

```
## Other  GasA  GasW
##    14  1428    18

freq(house.data$Heating)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##     var frequency percentage cumulative_perc
## 1  GasA      1428      97.81           97.81
## 2  GasW        18       1.23           99.04
## 3 Other        14       0.96          100.00

table(house.data$LowQualFinSF)

##
##    0   53   80  120  144  156  205  232  234  360  371  384  390  392  397
420
## 1434    1    3    1    1    1    1    1    1    2    1    1    1    1    1
1
##  473  479  481  513  514  515  528  572
##    1    1    1    1    1    1    1    1
```

```r
table(house.data$FullBath) #Since 0 and 1 are very different, we should keep separate
```

```
##
##   0   1   2   3
##   9 650 768  33
```

```r
###Make this continous###
house.data$BedroomAbvGr = as.factor(house.data$BedroomAbvGr)
summary(house.data$BedroomAbvGr) #merge some values together (5,6 7 as 5 or above), 0 is very different to 1, we should keep separate
```

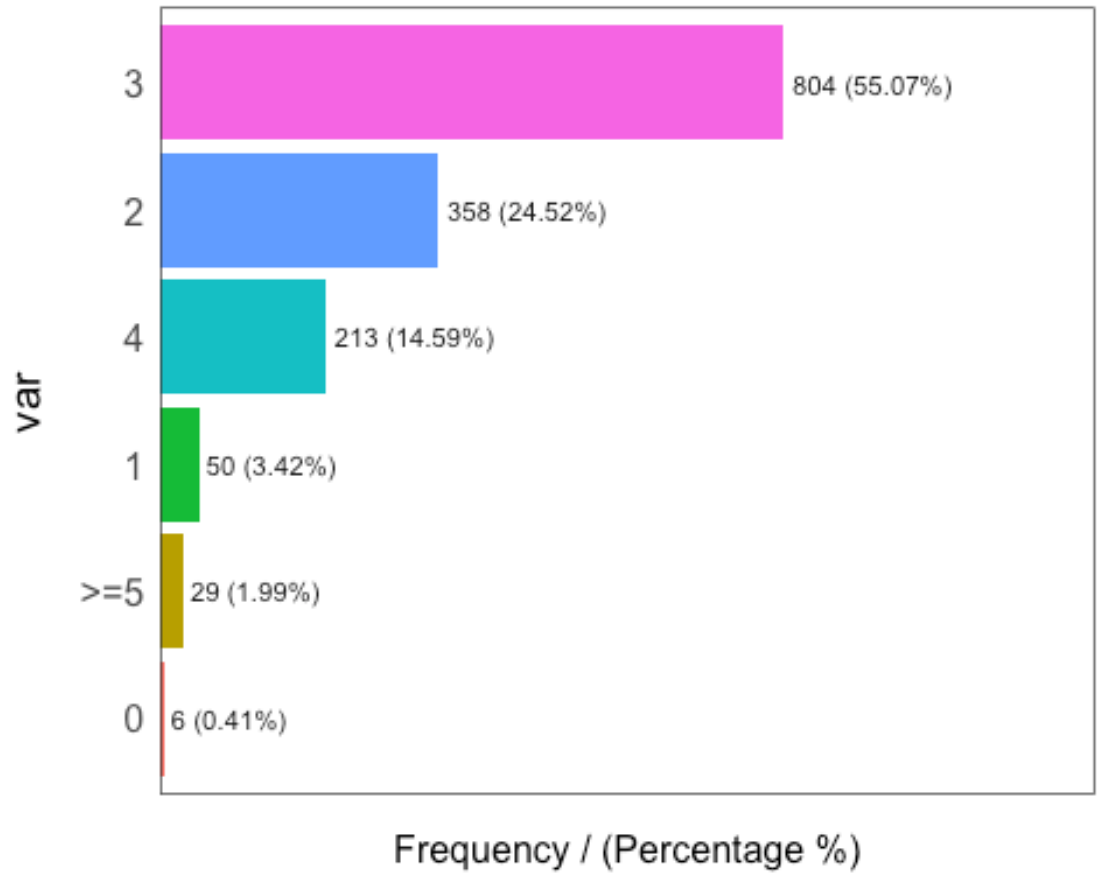
```
##   0   1   2   3   4   5   6   8
##   6  50 358 804 213  21   7   1
```

```r
levels(house.data$BedroomAbvGr) = c("0","1","2","3","4",">=5",">=5",">=5")
summary(house.data$BedroomAbvGr)
```

```
##   0   1   2   3   4 >=5
##   6  50 358 804 213  29
```

```r
freq(house.data$BedroomAbvGr)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##   var frequency percentage cumulative_perc
## 1   3       804      55.07           55.07
## 2   2       358      24.52           79.59
## 3   4       213      14.59           94.18
## 4   1        50       3.42           97.60
## 5 >=5        29       1.99           99.59
## 6   0         6       0.41          100.00
```

```
table(house.data$KitchenAbvGr) #should consider removing 0 and 3 from
dataset, too little observation for data
```

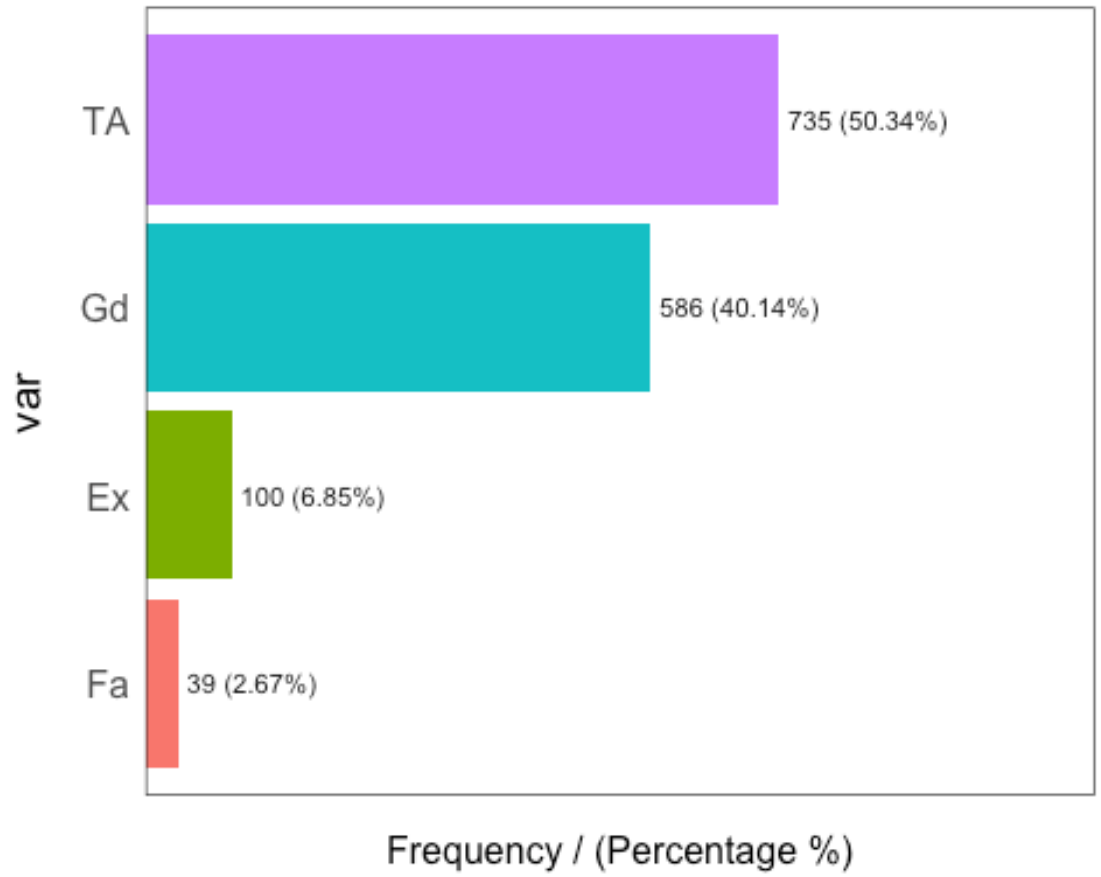
```
##
##    0    1    2    3
##    1 1392   65    2
```

```
house.data$KitchenQual = as.factor(house.data$KitchenQual)
summary(house.data$KitchenQual) #a useful variable! No NA's and good
representation per level
```

```
##  Ex  Fa  Gd  TA
## 100  39 586 735
```

```
freq(house.data$KitchenQual)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



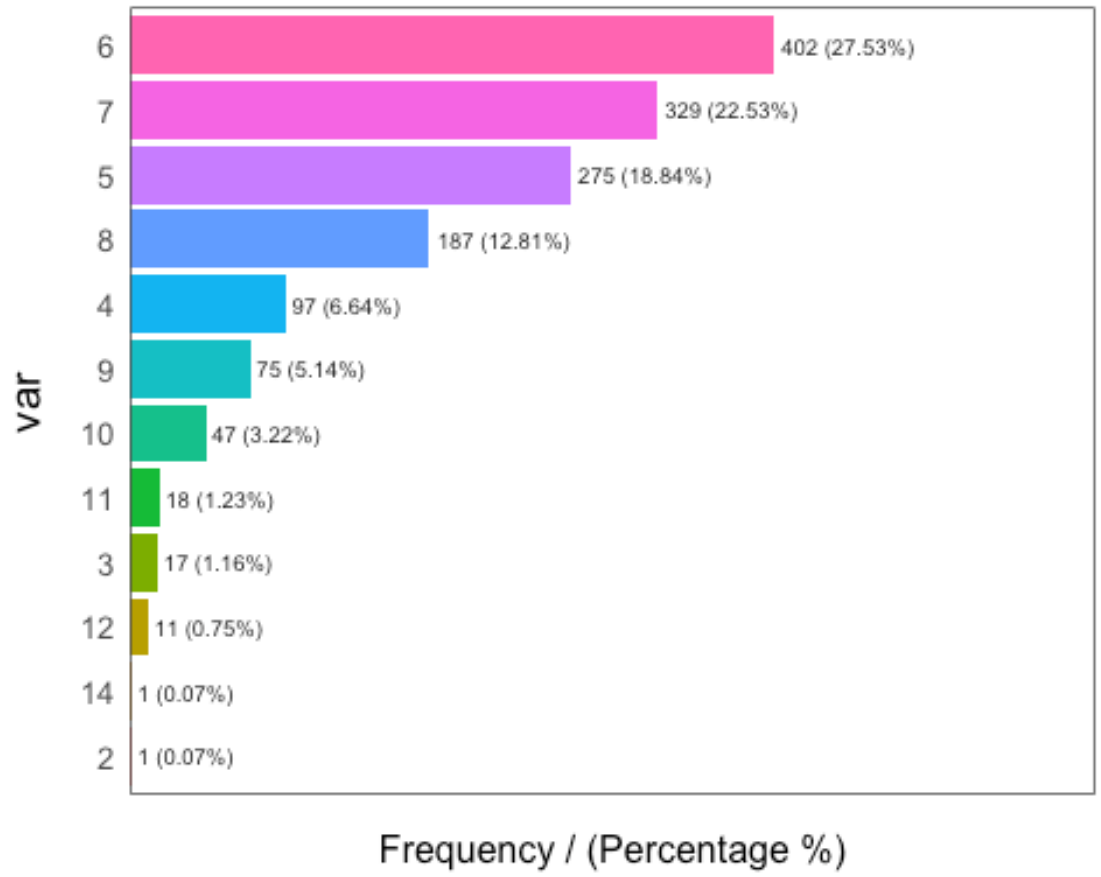

```
##   var frequency percentage cumulative_perc
## 1  TA       735      50.34           50.34
## 2  Gd       586      40.14           90.48
## 3  Ex       100       6.85           97.33
## 4  Fa        39       2.67          100.00
```

```
#We decided to keep TotRmsAbvGrd as continuous since we are interested in
keeping as much observations as possible
summary(house.data$TotRmsAbvGrd)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   5.000   6.000   6.518   7.000  14.000
```

```
freq(house.data$TotRmsAbvGrd)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##    var frequency percentage cumulative_perc
## 1    6       402      27.53           27.53
## 2    7       329      22.53           50.06
## 3    5       275      18.84           68.90
## 4    8       187      12.81           81.71
## 5    4        97       6.64           88.35
## 6    9        75       5.14           93.49
## 7   10        47       3.22           96.71
## 8   11        18       1.23           97.94
## 9    3        17       1.16           99.10
## 10  12        11       0.75           99.85
## 11   2         1       0.07           99.92
## 12  14         1       0.07          100.00
```

```
house.data$Functional = as.factor(house.data$Functional)
summary(house.data$Functional) #we can merge the levels like the following:
```

```
## Maj1 Maj2 Min1 Min2  Mod  Sev  Typ
##   14    5   31   34   15    1 1360
```

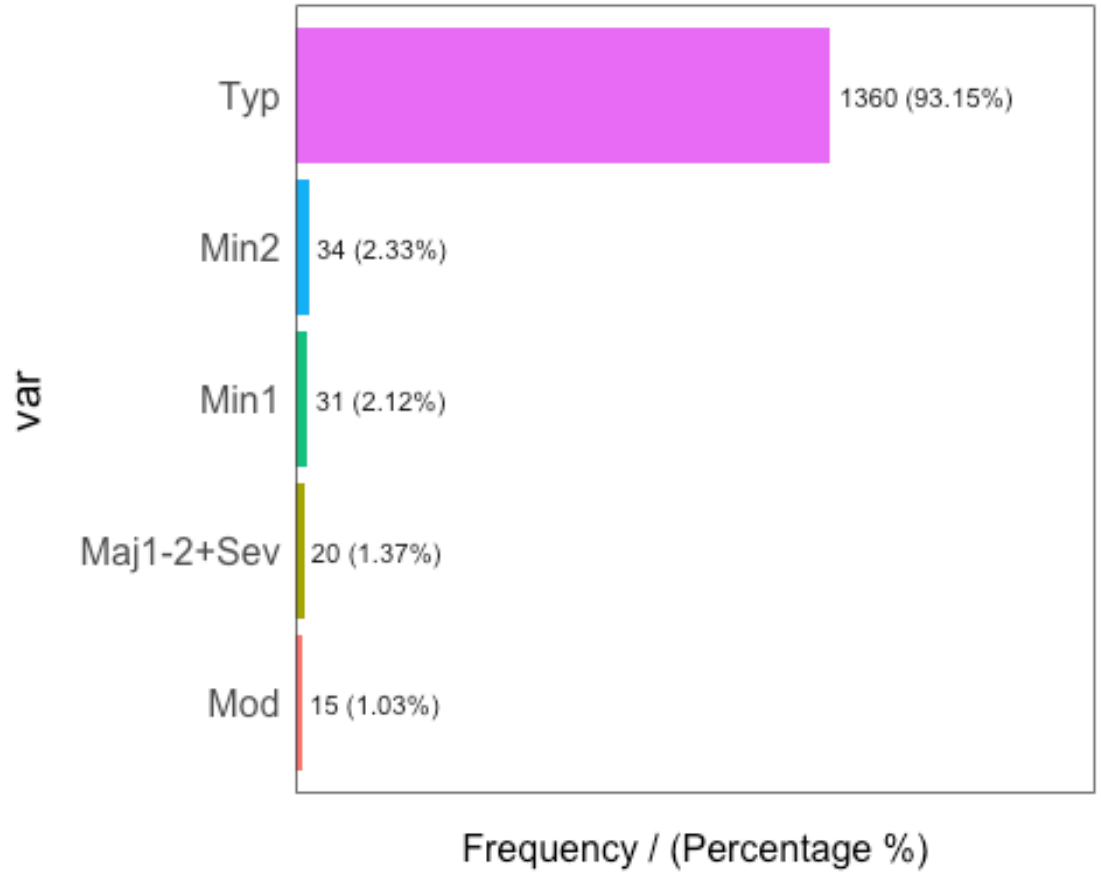
```
#Maj1, Maj2 & Sev; 1 or more major deductions or severley damaged; Maj1-
2+Sev
levels(house.data$Functional) = c("Maj1-2+Sev","Maj1-
```

```
2+Sev","Min1","Min2","Mod","Maj1-2+Sev","Typ")
summary(house.data$Functional)
```

```
## Maj1-2+Sev       Min1       Min2        Mod        Typ
##         20         31         34         15       1360
```

```
freq(house.data$Functional) #consider a 'Typ' not 'Typ' column?
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##          var frequency percentage cumulative_perc
## 1        Typ      1360      93.15           93.15
## 2       Min2        34       2.33           95.48
## 3       Min1        31       2.12           97.60
## 4 Maj1-2+Sev        20       1.37           98.97
## 5        Mod        15       1.03          100.00
```

```
###Make this continous###
house.data$Fireplaces = as.factor(house.data$Fireplaces)
summary(house.data$Fireplaces)
```

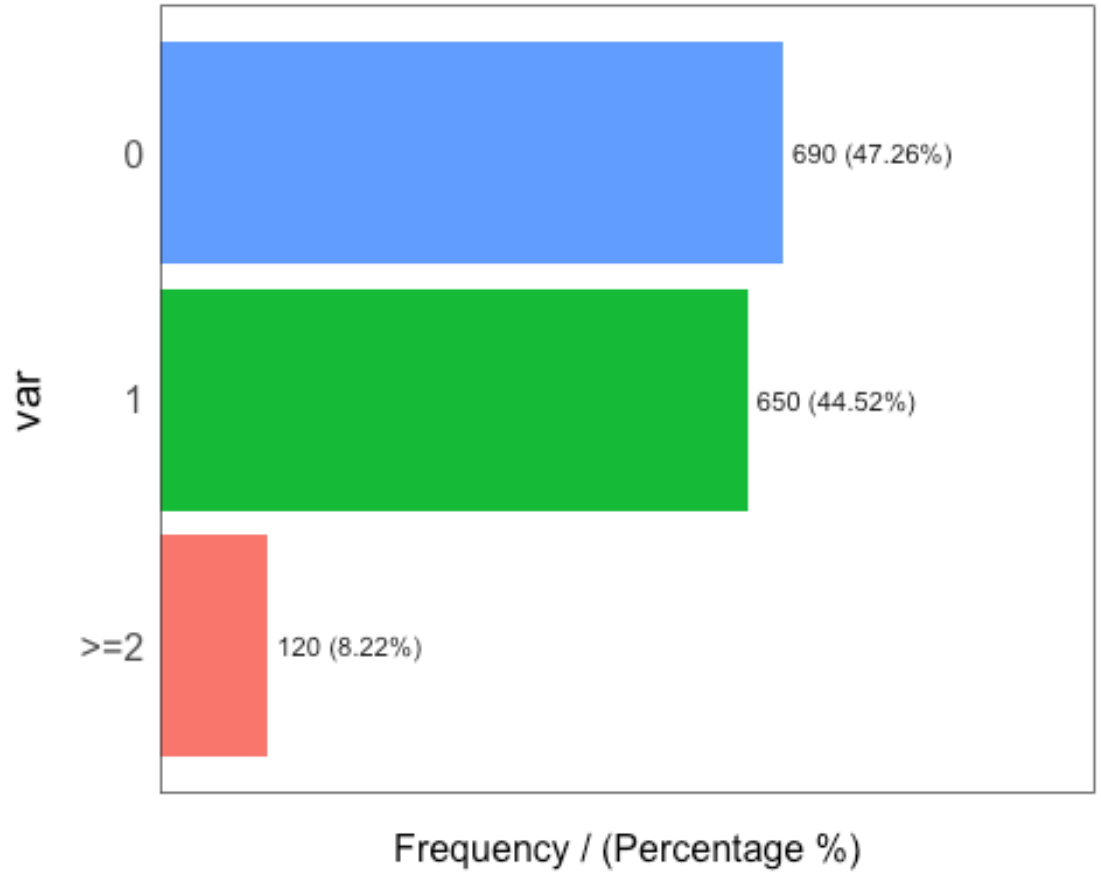
```
##   0   1   2   3
## 690 650 115   5
```

```
levels(house.data$Fireplaces) = c("0","1",">=2",">=2")
summary(house.data$Fireplaces)
```

```
##   0   1 >=2
## 690 650 120
```

```
freq(house.data$Fireplaces)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##   var frequency percentage cumulative_perc
## 1   0       690      47.26           47.26
## 2   1       650      44.52           91.78
## 3 >=2       120       8.22          100.00
```

```
house.data$GarageType[is.na(house.data$GarageType)] = 'NoGarage'
house.data$GarageType = as.factor(house.data$GarageType)
summary(house.data$GarageType) #some levels have less than 10 observations,
we should make a merge such as
```

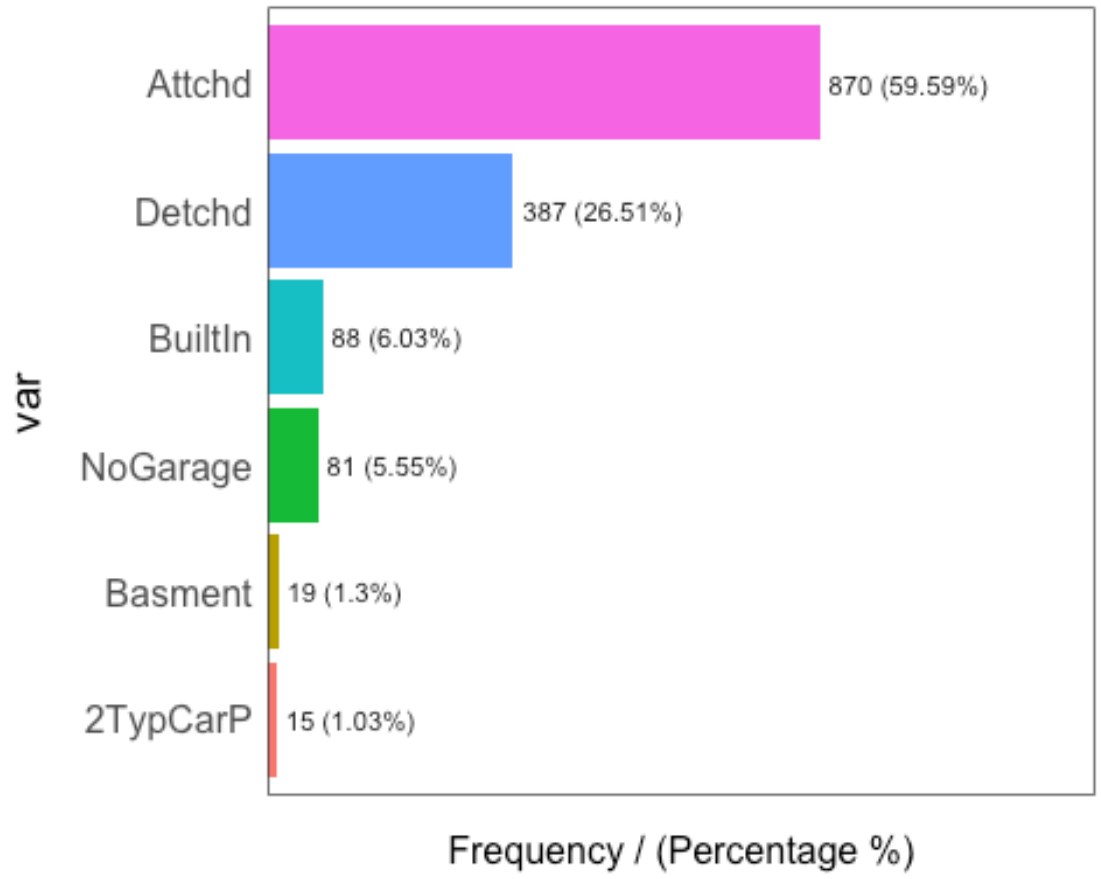
```
##   2Types   Attchd  Basment  BuiltIn  CarPort   Detchd NoGarage
##        6      870       19       88        9      387       81
```

```
#2Types & CarPort; 2 Types of garage or a cart port; 2TypCarP
levels(house.data$GarageType) =
c("2TypCarP","Attchd","Basment","BuiltIn","2TypCarP","Detchd","NoGarage")
summary(house.data$GarageType)
```

```
## 2TypCarP   Attchd  Basment  BuiltIn   Detchd NoGarage
##       15      870       19       88      387       81
```

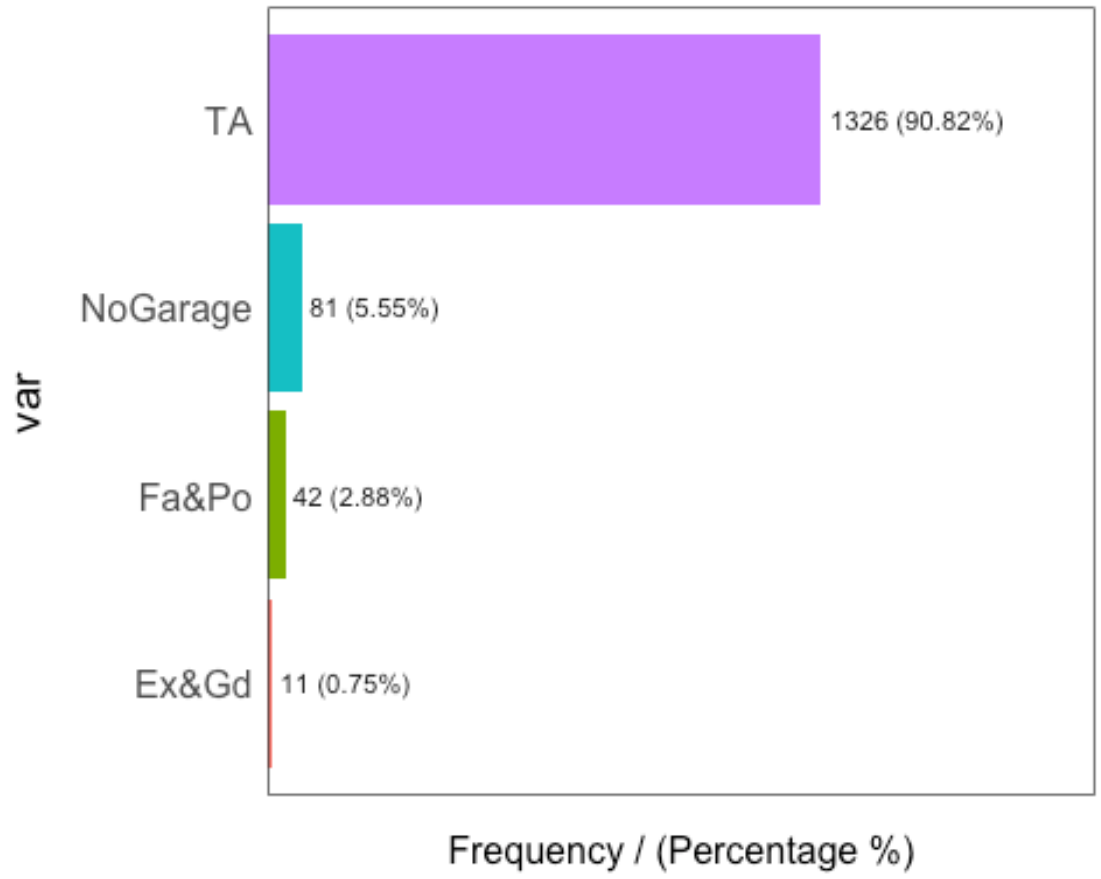
```
freq(house.data$GarageType)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##        var frequency percentage cumulative_perc
## 1   Attchd       870      59.59           59.59
## 2   Detchd       387      26.51           86.10
## 3  BuiltIn        88       6.03           92.13
## 4 NoGarage        81       5.55           97.68
## 5  Basment        19       1.30           98.98
## 6 2TypCarP        15       1.03          100.00
```

```
house.data$GarageCond[is.na(house.data$GarageCond)] = 'NoGarage'
house.data$GarageCond = as.factor(house.data$GarageCond)
summary(house.data$GarageCond) #We merge into Ex&Gd and Fa&Po

##       Ex       Fa       Gd NoGarage       Po       TA
##        2       35        9       81        7     1326

levels(house.data$GarageCond) =
c("Ex&Gd","Fa&Po","Ex&Gd","NoGarage","Fa&Po","TA")
summary(house.data$GarageCond)

##    Ex&Gd    Fa&Po NoGarage       TA
##       11       42       81     1326

freq(house.data$GarageCond)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##        var frequency percentage cumulative_perc
## 1       TA      1326      90.82           90.82
## 2 NoGarage        81       5.55           96.37
```

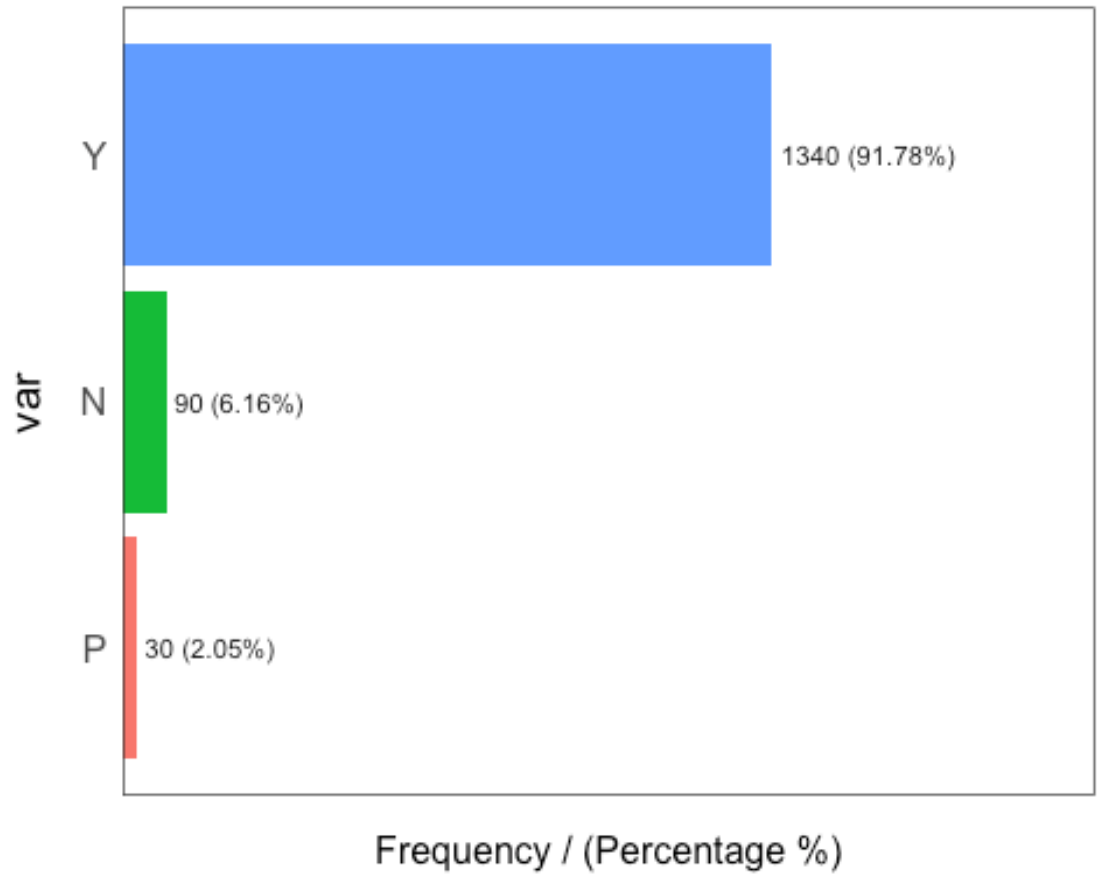
```
## 3    Fa&Po        42       2.88           99.25
## 4    Ex&Gd        11       0.75          100.00
```

```
house.data$PavedDrive = as.factor(house.data$PavedDrive)
summary(house.data$PavedDrive) #a clean variable! good representation per
level
```

```
##    N    P    Y
##   90   30 1340
```

```
freq(house.data$PavedDrive)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##   var frequency percentage cumulative_perc
## 1   Y      1340      91.78           91.78
## 2   N        90       6.16           97.94
## 3   P        30       2.05          100.00
```

```
table(house.data$PoolArea) #due to low amount of info, this variable isn't
good for modelling
```

```
##
##    0  480  512  519  555  576  648  738
## 1453    1    1    1    1    1    1    1
```

```
house.data$PoolArea = NULL
```

```
house.data$PoolQC = as.factor(house.data$PoolQC)
summary(house.data$PoolQC) #poor variable, consider removing for modelling
```

```
##   Ex   Fa   Gd NA's
##    2    2    3 1453
```

```
house.data$PoolQC = NULL
```

```
summary(house.data$MasVnrArea) #this variable has some NA, it's safe to
assume we can recode as 0 since NA in this context means no MasVnrArea
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0     0.0     0.0   103.7   166.0  1600.0       8
```

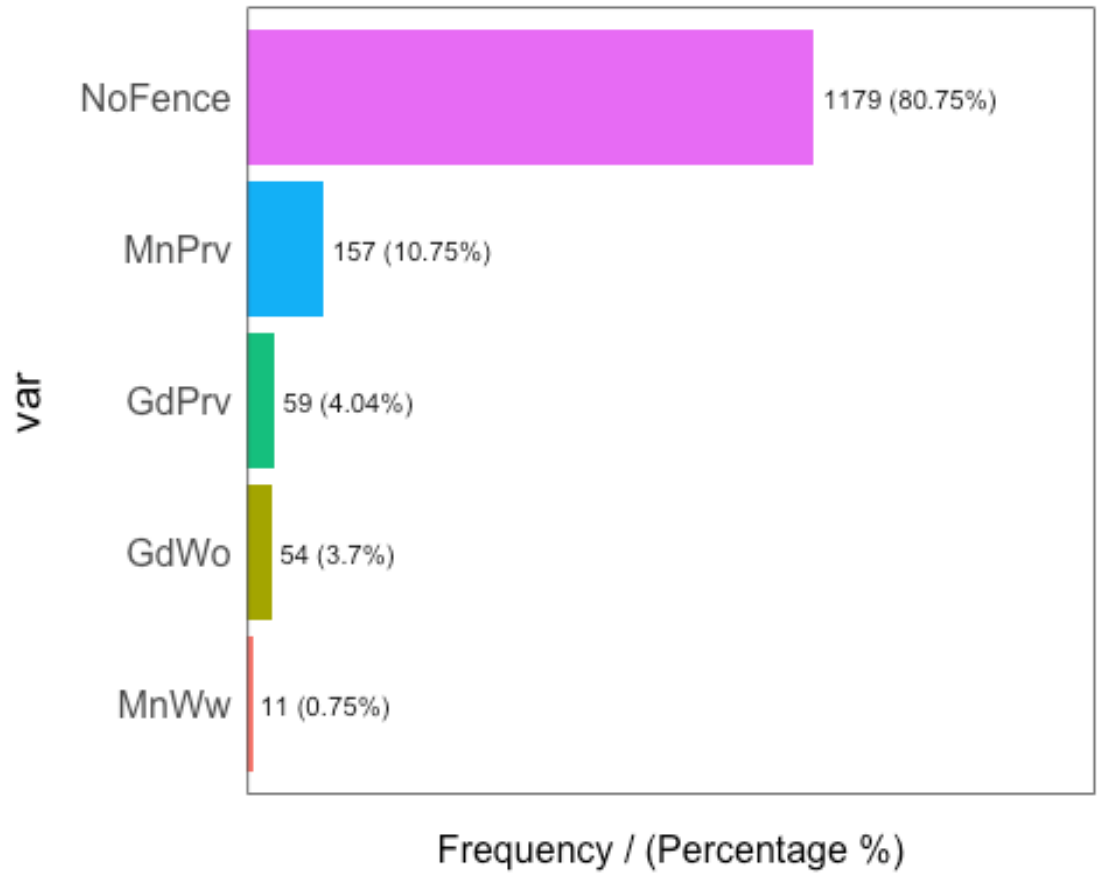
```
house.data$MasVnrArea[is.na(house.data$MasVnrArea)] = 0
```

```
house.data$Fence[is.na(house.data$Fence)] = 'NoFence'
house.data$Fence = as.factor(house.data$Fence)
summary(house.data$Fence) #good variable, decent amount of observations per
level
```

```
##   GdPrv    GdWo   MnPrv    MnWw NoFence
##      59      54     157      11    1179
```

```
freq(house.data$Fence)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##       var frequency percentage cumulative_perc
## 1 NoFence      1179      80.75           80.75
## 2   MnPrv       157      10.75           91.50
## 3   GdPrv        59       4.04           95.54
## 4    GdWo        54       3.70           99.24
## 5    MnWw        11       0.75          100.00
```

```
house.data$MiscFeature[is.na(house.data$MiscFeature)] = 'NoMiscF'
house.data$MiscFeature = as.factor(house.data$MiscFeature)
summary(house.data$MiscFeature) #turn this into a MiscFeature, no-MiscFeature
variable
```

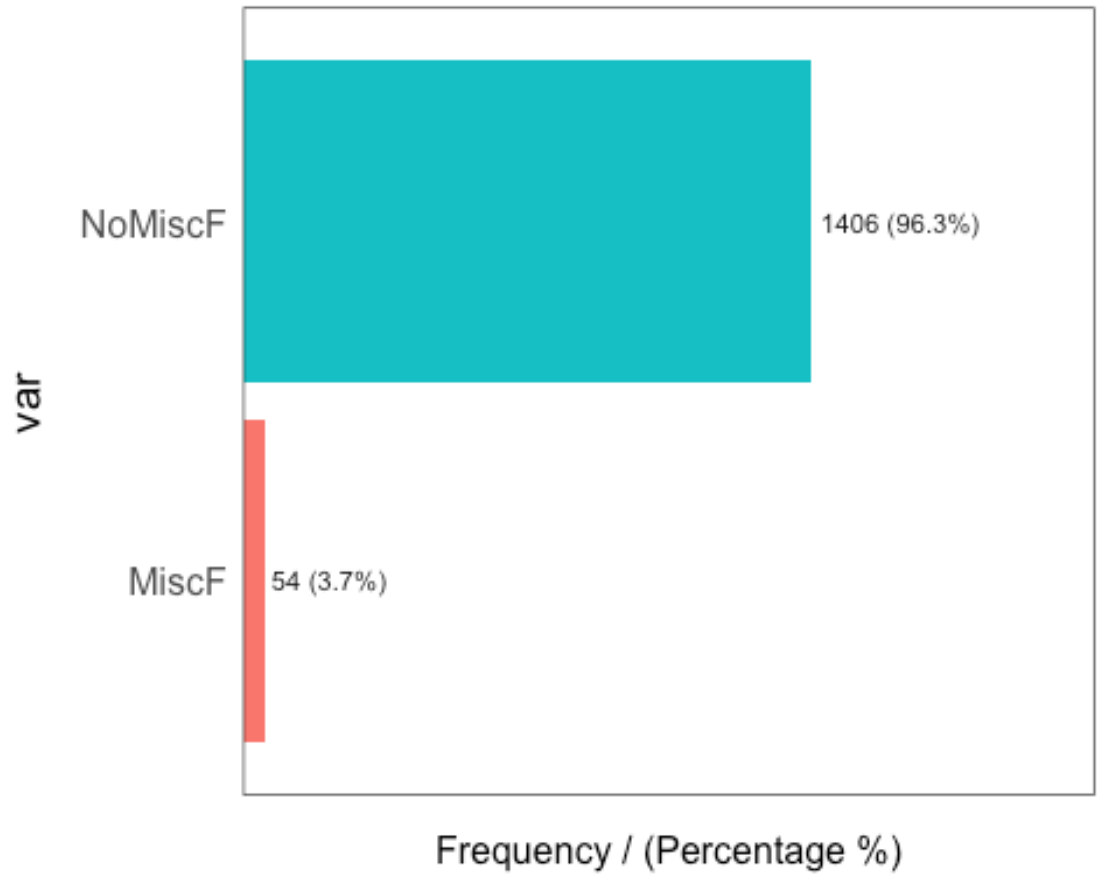
```
##    Gar2 NoMiscF    Othr    Shed    TenC
##       2    1406       2      49       1
```

```
levels(house.data$MiscFeature) = c("MiscF","NoMiscF","MiscF","MiscF","MiscF")
summary(house.data$MiscFeature)
```

```
##   MiscF NoMiscF
##      54    1406
```

```
freq(house.data$MiscFeature)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```

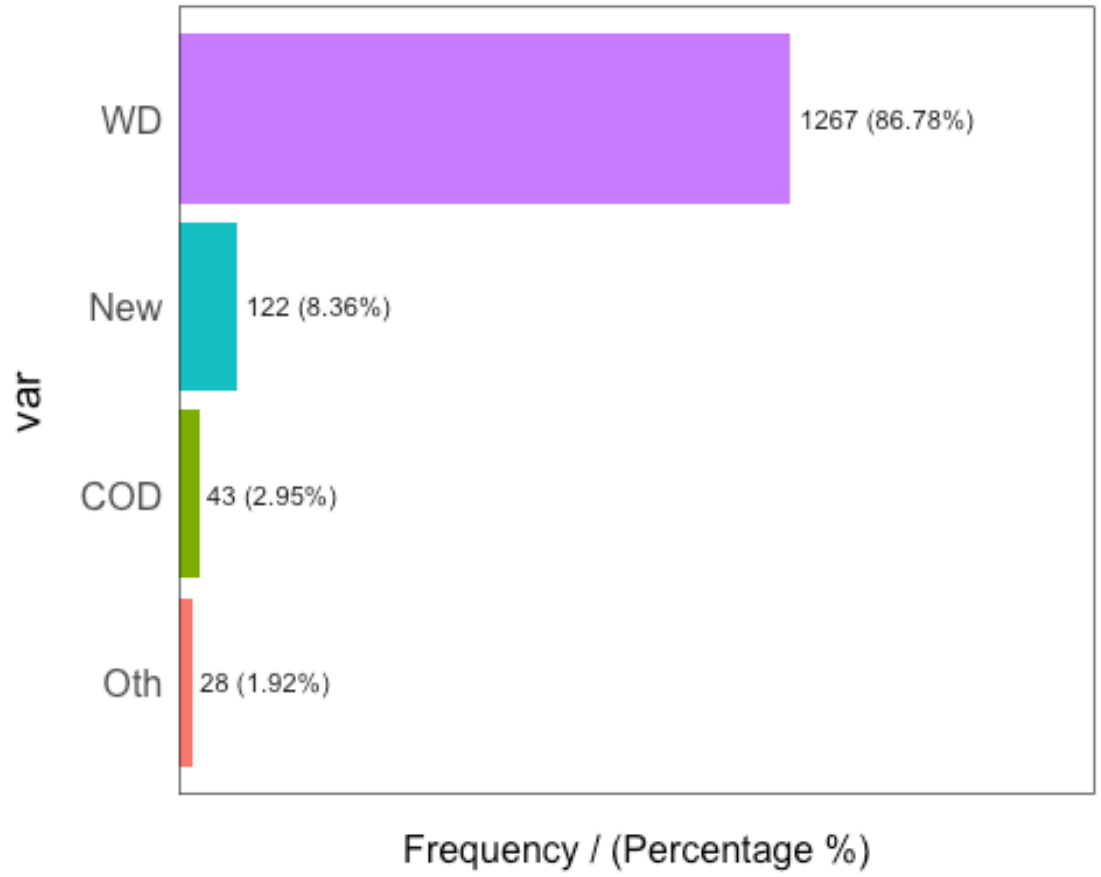



```
##       var frequency percentage cumulative_perc
## 1 NoMiscF      1406       96.3            96.3
## 2   MiscF        54        3.7           100.0

table(house.data$MoSold) #decent spread, worth keeping varible!

##
##   1   2   3   4   5   6   7   8   9  10  11  12
##  58  52 106 141 204 253 234 122  63  89  79  59

house.data$SaleType = as.factor(house.data$SaleType)
summary(house.data$SaleType) #Since we have an other category already, we can
just merge all other low levels into the other

##   COD   Con ConLD ConLI ConLw   CWD   New   Oth    WD
##    43     2     9     5     5     4   122     3  1267

levels(house.data$SaleType) =
c("COD","Oth","Oth","Oth","Oth","Oth","New","Oth","WD")
summary(house.data$SaleType)
```

```
##  COD  Oth  New   WD
##   43   28  122 1267
```

```
freq(house.data$SaleType)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##   var frequency percentage cumulative_perc
## 1  WD      1267      86.78           86.78
## 2 New       122       8.36           95.14
## 3 COD        43       2.95           98.09
## 4 Oth        28       1.92          100.00
```

```
house.data$SaleCondition = as.factor(house.data$SaleCondition)
summary(house.data$SaleCondition) #AdjLand low obs, we can merge:
```

```
## Abnorml AdjLand  Alloca  Family  Normal Partial
##     101       4      12      20    1198     125
```

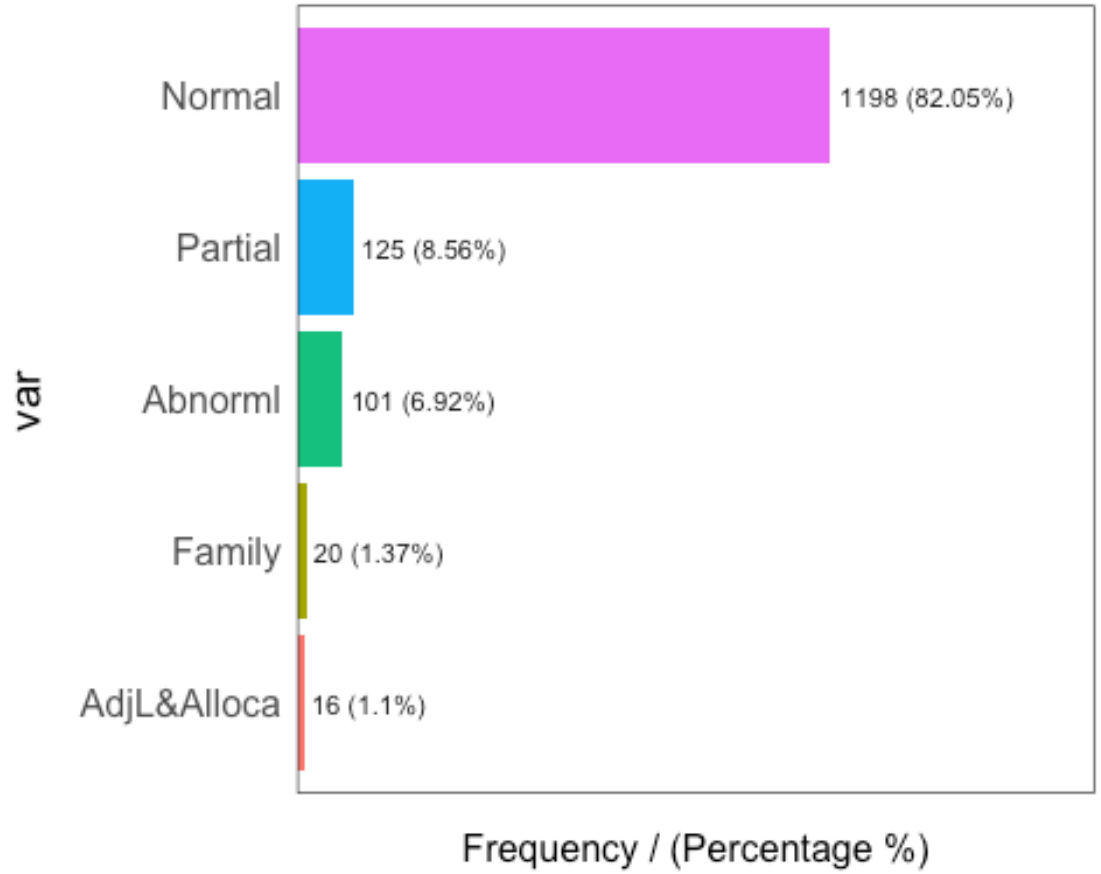
```
#AdjLand & Alloca; Adjoing Land and Two linked properties; AdjL&Alloca
levels(house.data$SaleCondition) =
```

```
c("Abnorml","AdjL&Alloca","AdjL&Alloca","Family","Normal","Partial")
summary(house.data$SaleCondition)
```

```
##     Abnorml AdjL&Alloca      Family      Normal     Partial
##         101          16          20        1198         125
```

```
freq(house.data$SaleCondition)
```

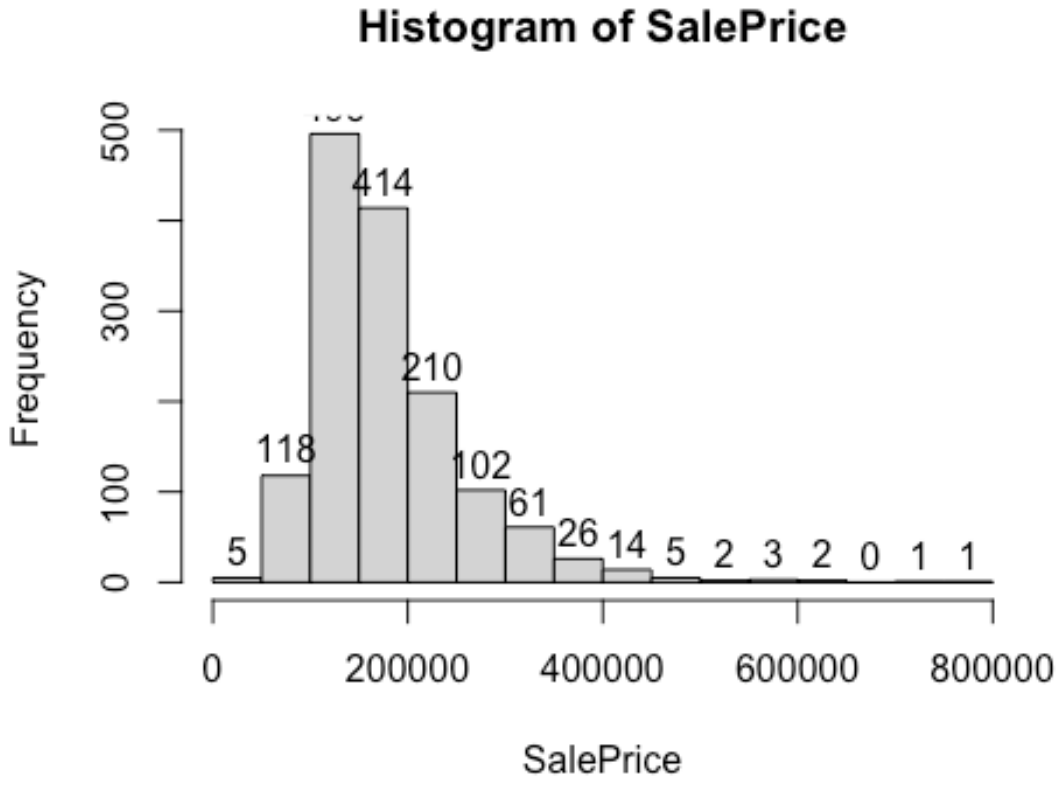
```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```




```
##           var frequency percentage cumulative_perc
## 1      Normal      1198      82.05           82.05
## 2     Partial       125       8.56           90.61
## 3     Abnorml       101       6.92           97.53
## 4      Family        20       1.37           98.90
## 5 AdjL&Alloca        16       1.10          100.00
```

```
#creating a histogram of SalePrice ----

#removal of scientific notation
options(scipen=999)
```

```r
#making histogram
hist(house.data$SalePrice,
     main = "Histogram of SalePrice",
     xlab = "SalePrice",
     labels = T
)
```




```r
#a tabled version of the histogram with 10 bins instead of 16 in the
histogram
table(cut(house.data$SalePrice,breaks=10))

##
## (3.42e+04,1.07e+05] (1.07e+05,1.79e+05] (1.79e+05,2.51e+05]
(2.51e+05,3.23e+05]
##                 148                 723                 373
135
## (3.23e+05,3.95e+05] (3.95e+05,4.67e+05] (4.67e+05,5.39e+05]
(5.39e+05,6.11e+05]
##                  51                  19                   4
3
## (6.11e+05,6.83e+05] (6.83e+05,7.56e+05]
##                   2                   2
```

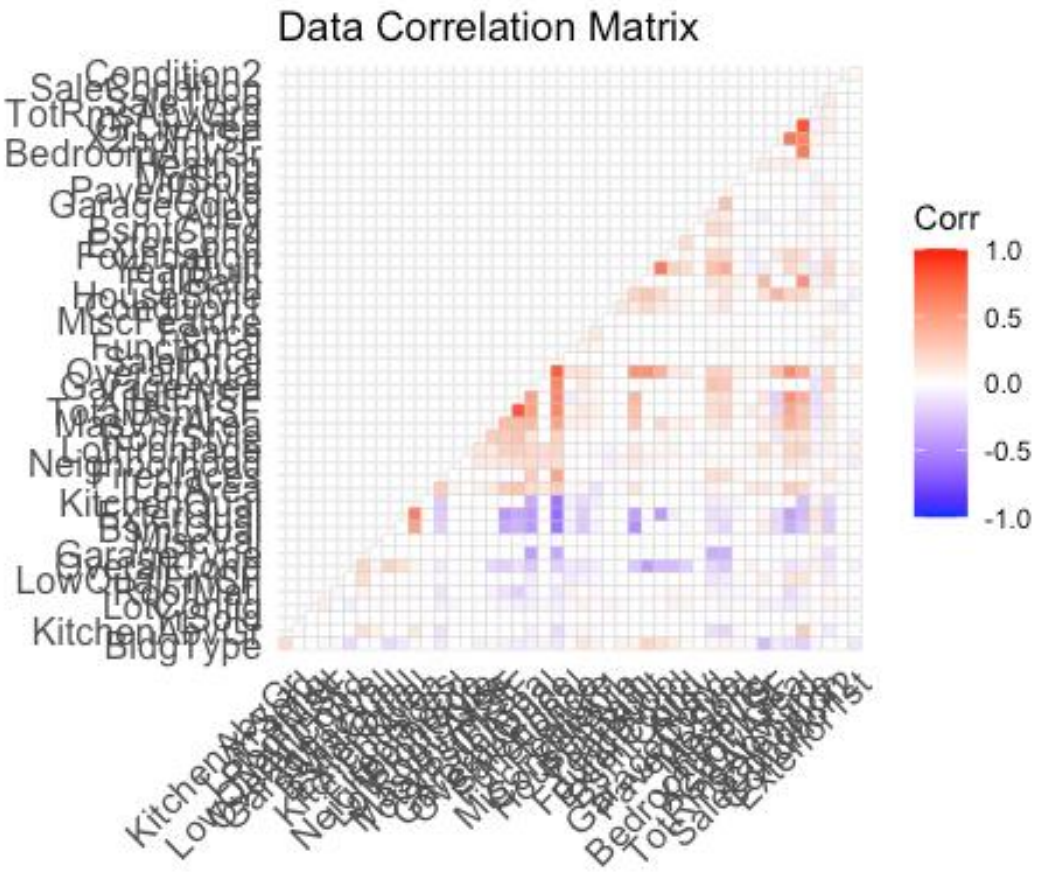
```r
#now data is cleaned, we need to make a dataset which can be used for
correlation checks
#this means tuning all our factors into purely numeric columns
house.data.corr = house.data

#this loops turns all numeric columns into factors
for (i in names(house.data.corr)) {
  if (is.factor(house.data.corr[[i]])) {
    house.data.corr[[i]] = as.numeric(house.data.corr[[i]])
  }
}

#Now lets do some correlation checks on the data

#creating the correlation matrix
summary(house.data.corr)
```

```
##   LotFrontage        LotArea           Alley         LotConfig
##  Min.   :  0.00   Min.   :  1300   Min.   :1.000   Min.   :1.000
##  1st Qu.: 42.00   1st Qu.:  7554   1st Qu.:2.000   1st Qu.:3.000
##  Median : 63.00   Median :  9478   Median :2.000   Median :4.000
##  Mean   : 57.62   Mean   : 10517   Mean   :1.994   Mean   :3.296
##  3rd Qu.: 79.00   3rd Qu.: 11602   3rd Qu.:2.000   3rd Qu.:4.000
##  Max.   :313.00   Max.   :215245   Max.   :3.000   Max.   :4.000
##   Neighborhood     Condition1      Condition2      BldgType
HouseStyle
##  Min.   : 1.00   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.
:1.000
##  1st Qu.: 8.00   1st Qu.:3.000   1st Qu.:2.00   1st Qu.:1.000   1st
Qu.:3.000
##  Median :13.00   Median :3.000   Median :2.00   Median :1.000   Median
:3.000
##  Mean   :12.67   Mean   :2.979   Mean   :1.99   Mean   :1.493   Mean
:3.656
##  3rd Qu.:17.00   3rd Qu.:3.000   3rd Qu.:2.00   3rd Qu.:1.000   3rd
Qu.:5.000
##  Max.   :24.00   Max.   :6.000   Max.   :2.00   Max.   :5.000   Max.
:7.000
##   OverallQual     OverallCond      YearBuilt      RoofStyle
RoofMatl
##  Min.   :1.000   Min.   :1.000   Min.   :1872   Min.   :1.000   Min.
:1.000
##  1st Qu.:3.000   1st Qu.:1.000   1st Qu.:1954   1st Qu.:2.000   1st
Qu.:2.000
##  Median :4.000   Median :1.000   Median :1973   Median :2.000   Median
:2.000
##  Mean   :4.104   Mean   :1.247   Mean   :1971   Mean   :2.397   Mean
:1.982
##  3rd Qu.:5.000   3rd Qu.:1.000   3rd Qu.:2000   3rd Qu.:2.000   3rd
Qu.:2.000
```

```
##  Max.   :8.000   Max.   :3.000   Max.   :2010   Max.   :4.000   Max.
:2.000
##   Exterior1st       MasVnrArea       ExterQual      ExterCond
##  Min.   : 1.000   Min.   :   0.0   Min.   :1.00   Min.   :1.000
##  1st Qu.: 5.000   1st Qu.:   0.0   1st Qu.:3.00   1st Qu.:3.000
##  Median : 8.000   Median :   0.0   Median :4.00   Median :3.000
##  Mean   : 6.604   Mean   : 103.1   Mean   :3.54   Mean   :2.776
##  3rd Qu.: 8.000   3rd Qu.: 164.2   3rd Qu.:4.00   3rd Qu.:3.000
##  Max.   :10.000   Max.   :1600.0   Max.   :4.00   Max.   :3.000
##    Foundation       BsmtQual        BsmtCond      TotalBsmtSF
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :   0.0
##  1st Qu.:2.000   1st Qu.:3.000   1st Qu.:4.000   1st Qu.: 795.8
##  Median :2.000   Median :3.000   Median :4.000   Median : 991.5
##  Mean   :2.388   Mean   :3.725   Mean   :3.789   Mean   :1057.4
##  3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:1298.2
##  Max.   :4.000   Max.   :5.000   Max.   :4.000   Max.   :6110.0
##     Heating        X1stFlrSF      X2ndFlrSF     LowQualFinSF
GrLivArea
##  Min.   :1.000   Min.   : 334   Min.   :   0   Min.   :  0.000   Min.   :
334
##  1st Qu.:2.000   1st Qu.: 882   1st Qu.:   0   1st Qu.:  0.000   1st
Qu.:1130
##  Median :2.000   Median :1087   Median :   0   Median :  0.000   Median
:1464
##  Mean   :2.003   Mean   :1163   Mean   : 347   Mean   :  5.845   Mean
:1515
##  3rd Qu.:2.000   3rd Qu.:1391   3rd Qu.: 728   3rd Qu.:  0.000   3rd
Qu.:1777
##  Max.   :3.000   Max.   :4692   Max.   :2065   Max.   :572.000   Max.
:5642
##     FullBath      BedroomAbvGr   KitchenAbvGr    KitchenQual
TotRmsAbvGrd
##  Min.   :0.000   Min.   :1.00   Min.   :0.000   Min.   :1.00   Min.   :
2.000
##  1st Qu.:1.000   1st Qu.:3.00   1st Qu.:1.000   1st Qu.:3.00   1st Qu.:
5.000
##  Median :2.000   Median :4.00   Median :1.000   Median :4.00   Median :
6.000
##  Mean   :1.565   Mean   :3.86   Mean   :1.047   Mean   :3.34   Mean   :
6.518
##  3rd Qu.:2.000   3rd Qu.:4.00   3rd Qu.:1.000   3rd Qu.:4.00   3rd Qu.:
7.000
##  Max.   :3.000   Max.   :6.00   Max.   :3.000   Max.   :4.00   Max.
:14.000
##    Functional      Fireplaces     GarageType     GarageArea
GarageCond
##  Min.   :1.000   Min.   :1.00   Min.   :1.00   Min.   :   0.0   Min.
:1.000
##  1st Qu.:5.000   1st Qu.:1.00   1st Qu.:2.00   1st Qu.: 334.5   1st
Qu.:4.000
```

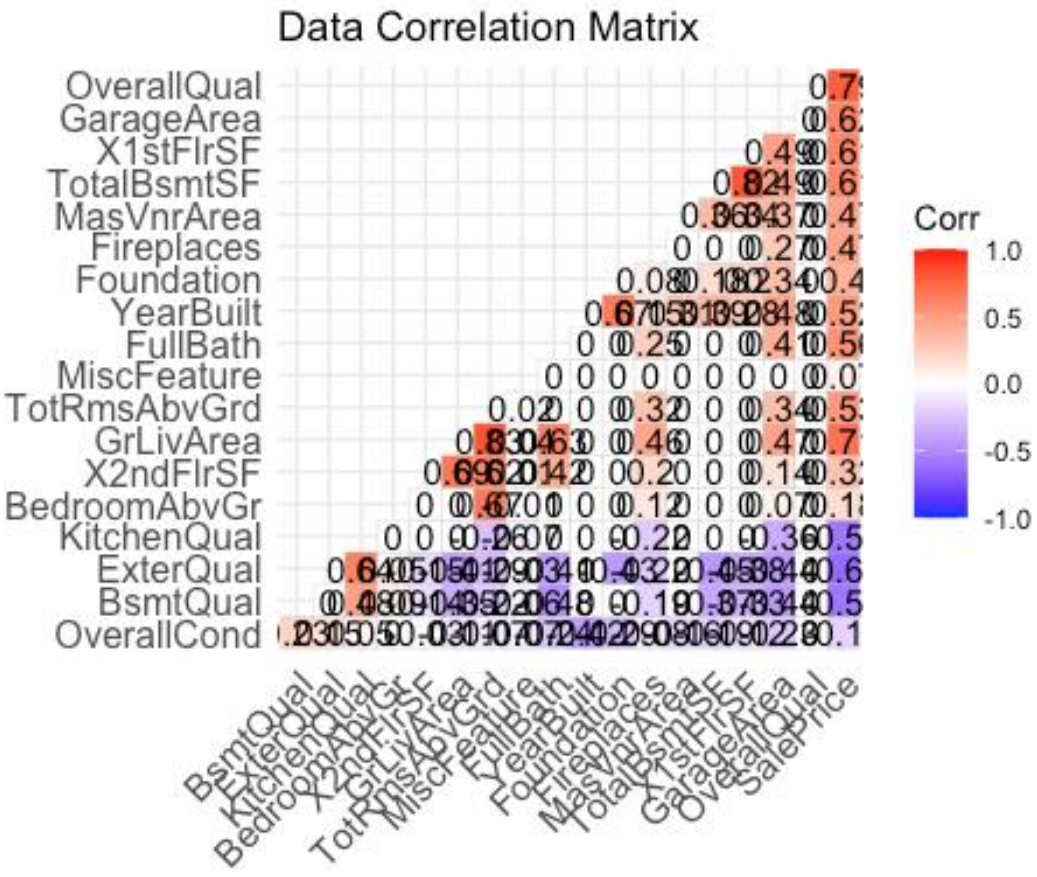
```
##  Median :5.000   Median :2.00   Median :2.00   Median : 480.0   Median
:4.000
##  Mean   :4.825   Mean   :1.61   Mean   :3.14   Mean   : 473.0   Mean
:3.864
##  3rd Qu.:5.000   3rd Qu.:2.00   3rd Qu.:5.00   3rd Qu.: 576.0   3rd
Qu.:4.000
##  Max.   :5.000   Max.   :3.00   Max.   :6.00   Max.   :1418.0   Max.
:4.000
##    PavedDrive        Fence        MiscFeature       MiscVal
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :    0.00
##  1st Qu.:3.000   1st Qu.:5.000   1st Qu.:2.000   1st Qu.:    0.00
##  Median :3.000   Median :5.000   Median :2.000   Median :    0.00
##  Mean   :2.856   Mean   :4.505   Mean   :1.963   Mean   :   43.49
##  3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:    0.00
##  Max.   :3.000   Max.   :5.000   Max.   :2.000   Max.   :15500.00
##      MoSold           YrSold        SaleType    SaleCondition
##  Min.   : 1.000   Min.   :2006   Min.   :1.00   Min.   :1.000
##  1st Qu.: 5.000   1st Qu.:2007   1st Qu.:4.00   1st Qu.:4.000
##  Median : 6.000   Median :2008   Median :4.00   Median :4.000
##  Mean   : 6.322   Mean   :2008   Mean   :3.79   Mean   :3.842
##  3rd Qu.: 8.000   3rd Qu.:2009   3rd Qu.:4.00   3rd Qu.:4.000
##  Max.   :12.000   Max.   :2010   Max.   :4.00   Max.   :5.000
##    SalePrice
##  Min.   : 34900
##  1st Qu.:129975
##  Median :163000
##  Mean   :180921
##  3rd Qu.:214000
##  Max.   :755000
```

```
corr = cor(house.data.corr, use = "complete.obs")
corr[upper.tri(corr)] = 0
diag(corr) = 0

#a plot of the full matrix
ggcorrplot(corr,
           type = "lower",
           lab = F,
           hc.order = T,
           title = "Data Correlation Matrix")
```




```
#it's a bit hard to read the full matrix, so lets start by removing some
variables
#which clearly show no signs of correlation

#sub-setting the data to remove those vars which visibly no correlation
house.data.corr.view  = subset(house.data.corr, select = -c(LotFrontage,
                                                            LotArea,
                                                            Alley,
                                                            LotConfig,
                                                            Neighborhood,
                                                            Condition1,
                                                            Condition2,
                                                            BldgType,
                                                            HouseStyle,
                                                            RoofStyle,
                                                            RoofMatl,
                                                            Exterior1st,
                                                            BsmtCond,
                                                            Heating,
                                                            LowQualFinSF,
                                                            Fence,
                                                            MoSold,
                                                            YrSold,
```

```
                                                    SaleType,
                                                    SaleCondition,
                                                    GarageType,
                                                    MiscVal,
                                                    KitchenAbvGr,
                                                    ExterCond,
                                                    PavedDrive,
                                                    Functional,
                                                    GarageCond))

#move the dependent to the end
#create the new selected correlation matrix
corr = cor(house.data.corr.view, use = "complete.obs")
corr[upper.tri(corr)] = 0
diag(corr) = 0

#a plot of the selected matrix
ggcorrplot(corr,
           type = "lower",
           lab = T,
           hc.order = T,
           title = "Data Correlation Matrix")
```



```
#From the above graph, we can now start to remove highly correlated vars from
the data-set,
#lets mainly those above 0.8;
#GrLivArea & TotRmsAbvGrd; 0.83
#TotalBsmtSF & X1stFlrSF; 0.82

#lets begin our removal process:
#first we will remove GrLivArea since it has the highest correlation coef
(with TotRmsA)
house.data$GrLivArea = NULL

#next we will remove TotalBsmtSF since it highly correlates with X1stFlrSF
and some houses don't have basements
#when they do have 1st floors
house.data$TotalBsmtSF = NULL

#there is no missing data in the dataset since a reason has been given fo
when a value is NA in a column
#therefore our final dataset is:
summary(house.data)
```

```
##   LotFrontage        LotArea           Alley        LotConfig
Neighborhood
##  Min.   :  0.00   Min.   :  1300   Grvl   :  50   Corner : 263   NAmes
:225
##  1st Qu.: 42.00   1st Qu.:  7554   NoAlley:1369   CulDSac:  94
CollgCr:150
##  Median : 63.00   Median :  9478   Pave   :  41   FR2&3  :  51
OldTown:113
##  Mean   : 57.62   Mean   : 10517                  Inside :1052
Edwards:100
##  3rd Qu.: 79.00   3rd Qu.: 11602                                 Somerst:
86
##  Max.   :313.00   Max.   :215245                                 Gilbert:
79
##
(Other):707
##   Condition1      Condition2     BldgType     HouseStyle   OverallQual
##  Artery:  48   Not-Norm:  15   1Fam  :1220   1.5Fin:154   5      :397
##  Feedr :  81   Norm    :1445   2fmCon:  31   1.5Unf: 14   6      :374
##  Norm  :1260                   Duplex:  52   1Story:726   7      :319
##  PosC  :  27                   Twnhs :  43   2.5All: 19   8      :168
##  RRCe  :  13                   TwnhsE: 114   2Story:445   4      :116
##  RRCn  :  31                                 SFoyer: 37   9      : 43
##                                              SLvl  : 65   (Other): 43
##   OverallCond     YearBuilt    RoofStyle           RoofMatl
Exterior1st
##  Average:1130   Min.   :1872   Flat :  13   Not-CompShg:  26   VinylSd:515
##  Good   : 299   1st Qu.:1954   Gable:1141   CompShg    :1434   HdBoard:222
##  Poor   :  31   Median :1973   Other:  20                      MetalSd:220
```

```
##                Mean   :1971   Hip  : 286                     Wd Sdng:206
##                3rd Qu.:2000                                   Plywood:108
##                Max.   :2010                                   CemntBd: 61
##                                                               (Other):128
##    MasVnrArea     ExterQual ExterCond         Foundation    BsmtQual
##  Min.   :   0.0   Ex: 52    Ex&Gd: 149   BrkTil    :146   Ex    :121
##  1st Qu.:   0.0   Fa: 14    Fa&Po:  29   CBlock    :634   Fa    : 35
##  Median :   0.0   Gd:488    TA   :1282   PConc     :647   Gd    :618
##  Mean   : 103.1   TA:906                SlaStnWood: 33   NoBsmt: 37
##  3rd Qu.: 164.2                                          TA    :649
##  Max.   :1600.0
##
##    BsmtCond     Heating       X1stFlrSF      X2ndFlrSF     LowQualFinSF
##  Fa&Po :  47   Other:  14   Min.   : 334   Min.   :   0   Min.   :  0.000
##  Gd    :  65   GasA :1428   1st Qu.: 882   1st Qu.:   0   1st Qu.:  0.000
##  NoBsmt:  37   GasW :  18   Median :1087   Median :   0   Median :  0.000
##  TA    :1311                Mean   :1163   Mean   : 347   Mean   :  5.845
##                             3rd Qu.:1391   3rd Qu.: 728   3rd Qu.:  0.000
##                             Max.   :4692   Max.   :2065   Max.   :572.000
##
##     FullBath     BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##  Min.   :0.000   0  :  6      Min.   :0.000   Ex:100      Min.   : 2.000
##  1st Qu.:1.000   1  : 50      1st Qu.:1.000   Fa: 39      1st Qu.: 5.000
##  Median :2.000   2  :358      Median :1.000   Gd:586      Median : 6.000
##  Mean   :1.565   3  :804      Mean   :1.047   TA:735      Mean   : 6.518
##  3rd Qu.:2.000   4  :213      3rd Qu.:1.000               3rd Qu.: 7.000
##  Max.   :3.000   >=5: 29      Max.   :3.000               Max.   :14.000
##
##       Functional   Fireplaces    GarageType    GarageArea
GarageCond
##  Maj1-2+Sev:  20   0  :690    2TypCarP: 15   Min.   :   0.0   Ex&Gd   :
11
##  Min1      :  31   1  :650    Attchd  :870   1st Qu.: 334.5   Fa&Po   :
42
##  Min2      :  34   >=2:120    Basment : 19   Median : 480.0   NoGarage:
81
##  Mod       :  15              BuiltIn : 88   Mean   : 473.0   TA
:1326
##  Typ       :1360              Detchd  :387   3rd Qu.: 576.0
##                               NoGarage: 81   Max.   :1418.0
##
##  PavedDrive     Fence       MiscFeature      MiscVal             MoSold
##  N:  90     GdPrv  :  59   MiscF  :  54   Min.   :    0.00   Min.   :
1.000
##  P:  30     GdWo   :  54   NoMiscF:1406   1st Qu.:    0.00   1st Qu.:
5.000
##  Y:1340     MnPrv  : 157                  Median :    0.00   Median :
6.000
##             MnWw   :  11                  Mean   :   43.49   Mean   :
6.322
```

```
##             NoFence:1179                  3rd Qu.:    0.00   3rd Qu.:
8.000
##                                           Max.   :15500.00   Max.
:12.000
##
##      YrSold     SaleType       SaleCondition    SalePrice
##  Min.   :2006   COD:  43   Abnorml    : 101   Min.   : 34900
##  1st Qu.:2007   Oth:  28   AdjL&Alloca:  16   1st Qu.:129975
##  Median :2008   New: 122   Family     :  20   Median :163000
##  Mean   :2008   WD :1267   Normal     :1198   Mean   :180921
##  3rd Qu.:2009              Partial    : 125   3rd Qu.:214000
##  Max.   :2010                                 Max.   :755000
##
```

```
dim(house.data)
```

```
## [1] 1460   44
```

```
#lastly, before modelling , we need to scale the continuous variables, can
achieve this using a for loop
for (i in names(house.data)) {
  if (is.numeric(house.data[[i]])) {
    house.data[[i]] = as.vector(scale(house.data[[i]]))
  }
}

#cleaning up the environment for team members going forward
rm(house.data.corr)
rm(house.data.corr.view)
rm(i)
rm(corr)


# Class counts in the OverallCond
# setting the seed before the partition

table(house.data$OverallCond)
```

```
##
## Average    Good    Poor
##    1130     299      31
```

```
set.seed(10)
index <- createDataPartition(house.data$OverallCond, p=0.8, times = 1,
list=F)
ytrain <-unlist(house.data[index,11])
ytest <- unlist(house.data[-index,11])
xtest <- house.data[-index,-11]
xtrain <- house.data[index,-11]
```

```
table(ytest)

## ytest
## Average    Good    Poor
##     226      59       6

table(ytrain)

## ytrain
## Average    Good    Poor
##     904     240      25

# FILIPS garden
================================================================
summary(house.data)

##   LotFrontage         LotArea            Alley        LotConfig
##  Min.   :-1.6623   Min.   :-0.9234   Grvl   :  50   Corner : 263
##  1st Qu.:-0.4507   1st Qu.:-0.2969   NoAlley:1369   CulDSac:  94
##  Median : 0.1551   Median :-0.1040   Pave   :  41   FR2&3  :  51
##  Mean   : 0.0000   Mean   : 0.0000                  Inside :1052
##  3rd Qu.: 0.6167   3rd Qu.: 0.1087
##  Max.   : 7.3671   Max.   :20.5112
##
##   Neighborhood  Condition1      Condition2     BldgType     HouseStyle
##  NAmes  :225   Artery:  48   Not-Norm:  15   1Fam  :1220   1.5Fin:154
##  CollgCr:150   Feedr :  81   Norm    :1445   2fmCon:  31   1.5Unf: 14
##  OldTown:113   Norm  :1260                   Duplex:  52   1Story:726
##  Edwards:100   PosC  :  27                   Twnhs :  43   2.5All: 19
##  Somerst: 86   RRCe  :  13                   TwnhsE: 114   2Story:445
##  Gilbert: 79   RRCn  :  31                                 SFoyer: 37
##  (Other):707                                               SLvl  : 65
##   OverallQual   OverallCond     YearBuilt        RoofStyle
##  5      :397   Average:1130   Min.   :-3.28670   Flat :  13
##  6      :374   Good   : 299   1st Qu.:-0.57173   Gable:1141
##  7      :319   Poor   :  31   Median : 0.05735   Other:  20
##  8      :168                  Mean   : 0.00000   Hip  : 286
##  4      :116                  3rd Qu.: 0.95131
##  9      : 43                  Max.   : 1.28240
##  (Other): 43
##         RoofMatl     Exterior1st    MasVnrArea      ExterQual ExterCond
##  Not-CompShg:  26   VinylSd:515   Min.   :-0.5706   Ex: 52    Ex&Gd: 149
##  CompShg    :1434   HdBoard:222   1st Qu.:-0.5706   Fa: 14    Fa&Po:  29
##                     MetalSd:220   Median :-0.5706   Gd:488    TA   :1282
##                     Wd Sdng:206   Mean   : 0.0000   TA:906
##                     Plywood:108   3rd Qu.: 0.3383
##                     CemntBd: 61   Max.   : 8.2824
##                     (Other):128
##       Foundation    BsmtQual     BsmtCond     Heating       X1stFlrSF
##  BrkTil    :146   Ex    :121   Fa&Po :  47   Other:  14   Min.   :-2.1434
```

```
##  CBlock    :634   Fa    : 35   Gd    :  65   GasA :1428   1st Qu.:-0.7259
##  PConc     :647   Gd    :618   NoBsmt:  37   GasW :  18   Median :-0.1956
##  SlaStnWood: 33   NoBsmt: 37   TA    :1311                Mean   : 0.0000
##                   TA    :649                              3rd Qu.: 0.5914
##                                                           Max.   : 9.1296
##
##    X2ndFlrSF        LowQualFinSF        FullBath       BedroomAbvGr
##  Min.   :-0.7949   Min.   :-0.1202   Min.   :-2.8408   0  :  6
##  1st Qu.:-0.7949   1st Qu.:-0.1202   1st Qu.:-1.0257   1  : 50
##  Median :-0.7949   Median :-0.1202   Median : 0.7895   2  :358
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   3  :804
##  3rd Qu.: 0.8728   3rd Qu.:-0.1202   3rd Qu.: 0.7895   4  :213
##  Max.   : 3.9356   Max.   :11.6438   Max.   : 2.6046   >=5: 29
##
##   KitchenAbvGr     KitchenQual  TotRmsAbvGrd          Functional
Fireplaces
##  Min.   :-4.7499   Ex:100      Min.   :-2.7795   Maj1-2+Sev:  20   0  :690
##  1st Qu.:-0.2114   Fa: 39      1st Qu.:-0.9338   Min1      :  31   1  :650
##  Median :-0.2114   Gd:586      Median :-0.3186   Min2      :  34   >=2:120
##  Mean   : 0.0000   TA:735      Mean   : 0.0000   Mod       :  15
##  3rd Qu.:-0.2114               3rd Qu.: 0.2967   Typ       :1360
##  Max.   : 8.8656               Max.   : 4.6033
##
##     GarageType    GarageArea          GarageCond   PavedDrive     Fence
##  2TypCarP: 15   Min.   :-2.21220   Ex&Gd   :  11   N:  90     GdPrv  :  59
##  Attchd  :870   1st Qu.:-0.64769   Fa&Po   :  42   P:  30     GdWo   :  54
##  Basment : 19   Median : 0.03283   NoGarage:  81   Y:1340     MnPrv  : 157
##  BuiltIn : 88   Mean   : 0.00000   TA      :1326              MnWw   :  11
##  Detchd  :387   3rd Qu.: 0.48184                              NoFence:1179
##  NoGarage: 81   Max.   : 4.42001
##
##   MiscFeature      MiscVal             MoSold            YrSold
##  MiscF  :  54   Min.   :-0.08766   Min.   :-1.9684   Min.   :-1.3672
##  NoMiscF:1406   1st Qu.:-0.08766   1st Qu.:-0.4889   1st Qu.:-0.6142
##                 Median :-0.08766   Median :-0.1191   Median : 0.1387
##                 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##                 3rd Qu.:-0.08766   3rd Qu.: 0.6207   3rd Qu.: 0.8917
##                 Max.   :31.15459   Max.   : 2.1002   Max.   : 1.6446
##
##  SaleType       SaleCondition    SalePrice
##  COD:  43   Abnorml    : 101   Min.   :-1.8381
##  Oth:  28   AdjL&Alloca:  16   1st Qu.:-0.6413
##  New: 122   Family     :  20   Median :-0.2256
##  WD :1267   Normal     :1198   Mean   : 0.0000
##             Partial    : 125   3rd Qu.: 0.4164
##                                Max.   : 7.2263
##

# https://www.analyticsvidhya.com/blog/2016/02/multinomial-ordinal-logistic-
regression/
```

```r
# making a copy of the data not to disturb later processes
mlr_data <- house.data
sapply(mlr_data %>% select_if(is.factor), function(x){length(unique(x))})
```

```
##         Alley     LotConfig  Neighborhood    Condition1    Condition2
##             3             4            24             6             2
##      BldgType    HouseStyle   OverallQual   OverallCond     RoofStyle
##             5             7             8             3             4
##      RoofMatl   Exterior1st     ExterQual     ExterCond    Foundation
##             2            10             4             3             4
##      BsmtQual      BsmtCond       Heating  BedroomAbvGr   KitchenQual
##             5             4             3             6             4
##    Functional    Fireplaces    GarageType    GarageCond    PavedDrive
##             5             3             6             4             3
##         Fence   MiscFeature      SaleType SaleCondition
##             5             2             4             5
```

```r
glimpse(mlr_data)
```

```
## Rows: 1,460
## Columns: 44
## $ LotFrontage   <dbl> 0.21280428, 0.64552608, 0.29934864, 0.06856368,
0.760918…
## $ LotArea       <dbl> -0.20707076, -0.09185490, 0.07345481, -0.09686428,
0.375…
## $ Alley         <fct> NoAlley, NoAlley, NoAlley, NoAlley, NoAlley,
NoAlley, No…
## $ LotConfig     <fct> Inside, FR2&3, Inside, Corner, FR2&3, Inside,
Inside, Co…
## $ Neighborhood  <fct> CollgCr, Veenker, CollgCr, Crawfor, NoRidge,
Mitchel, So…
## $ Condition1    <fct> Norm, Feedr, Norm, Norm, Norm, Norm, Norm, PosC,
Artery,…
## $ Condition2    <fct> Norm, Norm, Norm, Norm, Norm, Norm, Norm, Norm,
Norm, No…
## $ BldgType      <fct> 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam, 1Fam,
1Fam, 2f…
## $ HouseStyle    <fct> 2Story, 1Story, 2Story, 2Story, 2Story, 1.5Fin,
1Story, …
## $ OverallQual   <fct> 7, 6, 7, 7, 8, 5, 8, 7, 7, 5, 5, 9, 5, 7, 6, 7, 6,
4, 5,…
## $ OverallCond   <fct> Average, Good, Average, Average, Average, Average,
Avera…
## $ YearBuilt     <dbl> 1.05063380, 0.15668003, 0.98441500, -1.86299331,
0.95130…
## $ RoofStyle     <fct> Gable, Gable, Gable, Gable, Gable, Gable, Gable,
Gable, …
## $ RoofMatl      <fct> CompShg, CompShg, CompShg, CompShg, CompShg,
CompShg, Co…
```

```
## $ Exterior1st   <fct> VinylSd, MetalSd, VinylSd, Wd Sdng, VinylSd,
VinylSd, Vi…
## $ MasVnrArea    <dbl> 0.5139278, -0.5705546, 0.3258033, -0.5705546,
1.3660211,…
## $ ExterQual     <fct> Gd, TA, Gd, TA, Gd, TA, Gd, TA, TA, TA, TA, Ex, TA,
Gd, …
## $ ExterCond     <fct> TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA,
TA, …
## $ Foundation    <fct> PConc, CBlock, PConc, BrkTil, PConc, SlaStnWood,
PConc, …
## $ BsmtQual      <fct> Gd, Gd, Gd, TA, Gd, Gd, Ex, Gd, TA, TA, TA, Ex, TA,
Gd, …
## $ BsmtCond      <fct> TA, TA, TA, Gd, TA, TA, TA, TA, TA, TA, TA, TA, TA,
TA, …
## $ Heating       <fct> GasA, GasA, GasA, GasA, GasA, GasA, GasA, GasA,
GasA, Ga…
## $ X1stFlrSF     <dbl> -0.79316202, 0.25705235, -0.62761099, -0.52155486, -
0.04…
## $ X2ndFlrSF     <dbl> 1.1614536, -0.7948909, 1.1889432, 0.9369551,
1.6173231, …
## $ LowQualFinSF  <dbl> -0.1202005, -0.1202005, -0.1202005, -0.1202005, -
0.12020…
## $ FullBath      <dbl> 0.789470, 0.789470, 0.789470, -1.025689, 0.789470, -
1.02…
## $ BedroomAbvGr  <fct> 3, 3, 3, 3, 4, 1, 3, 3, 2, 2, 3, 4, 2, 3, 2, 2, 2,
2, 3,…
## $ KitchenAbvGr  <dbl> -0.2113812, -0.2113812, -0.2113812, -0.2113812, -
0.21138…
## $ KitchenQual   <fct> Gd, TA, Gd, Gd, Gd, TA, Gd, TA, TA, TA, TA, Ex, TA,
Gd, …
## $ TotRmsAbvGrd  <dbl> 0.9118973, -0.3185741, -0.3185741, 0.2966616,
1.5271330,…
## $ Functional    <fct> Typ, Typ, Typ, Typ, Typ, Typ, Typ, Typ, Min1, Typ,
Typ, …
## $ Fireplaces    <fct> 0, 1, 1, 1, 1, 0, 1, >=2, >=2, >=2, 0, >=2, 0, 1, 1,
0, …
## $ GarageType    <fct> Attchd, Attchd, Attchd, Detchd, Attchd, Attchd,
Attchd, …
## $ GarageArea    <dbl> 0.35088009, -0.06071021, 0.63150985, 0.79053338,
1.69790…
## $ GarageCond    <fct> TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA, TA,
TA, …
## $ PavedDrive    <fct> Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y,
Y, Y,…
## $ Fence         <fct> NoFence, NoFence, NoFence, NoFence, NoFence, MnPrv,
NoFe…
## $ MiscFeature   <fct> NoMiscF, NoMiscF, NoMiscF, NoMiscF, NoMiscF, MiscF,
NoMi…
## $ MiscVal       <dbl> -0.08765778, -0.08765778, -0.08765778, -0.08765778,
-0.0…
```

```
## $ MoSold        <dbl> -1.5985634, -0.4889425, 0.9905519, -1.5985634,
2.1001728…
## $ YrSold        <dbl> 0.1387300, -0.6142282, 0.1387300, -1.3671863,
0.1387300,…
## $ SaleType      <fct> WD, WD, WD, WD, WD, WD, WD, WD, WD, WD, WD, New, WD,
New…
## $ SaleCondition <fct> Normal, Normal, Normal, Abnorml, Normal, Normal,
Normal,…
## $ SalePrice     <dbl> 0.347154270, 0.007285824, 0.535970074, -0.515104565,
0.8…

table(mlr_data$OverallQual)

##
## 3<=   4   5   6   7   8   9  10
##  25 116 397 374 319 168  43  18

# OverallQual has 8 unique values and natural order, good enough to use as
# numerical variable
mlr_data <- mlr_data %>%
  mutate(OverallQual = as.numeric(OverallQual))

# pick a reference level, Average has the highest frequency
mlr_data$OverallCond <- relevel(mlr_data$OverallCond, ref = "Average")

# test/train split
mlr_train <- mlr_data[index,]
mlr_test <- mlr_data[-index,]

# source https://www.youtube.com/watch?v=QvnsTXfPenU
# Logistic regression predicts a binary variable, a natural extension is
# multinomial logistic regression, which essentially does the same but
ensembles
# more than two classes, both are used for nominal target variables. Simply
# said, it asks the question whether a predictor contributes towards the
final
# outcome or not, and how much to each class of the outcome.

# it would be reasonable to expect that neighborhood would contribute towards
# successfull prediction of overall condition, but many of it's categories
# ended up being insignificant and were polluting the model, which was
causing
# issues given the small pool of data we have available w.r.t. the expanded
# predictors

# full model including all predictors
model_mlr_f <- multinom(OverallCond ~ . -Fireplaces -Heating -GarageType
                        -LotConfig -Alley -LotArea -LotFrontage -PavedDrive
                        -Neighborhood -Exterior1st -HouseStyle,
                        data = mlr_train, trace = F)
```

```r
# null model including only the intercept
model_mlr_n <- multinom( OverallCond ~ 1,
                         data = mlr_train, trace = F)

# output is muted, this takes about 30s to run
tic("fitting mlr model")
model_mlr_aic_b <- stepAIC(model_mlr_f, direction = "backward", trace=FALSE)
model_mlr_aic_f <- stepAIC(model_mlr_n,direction="forward",
scope=list(upper=model_mlr_f,lower=model_mlr_n), trace=FALSE)
toc()

## fitting mlr model: 39.83 sec elapsed

summary(model_mlr_aic_b)

## Call:
## multinom(formula = OverallCond ~ Condition1 + BldgType + YearBuilt +
##     RoofMatl + MasVnrArea + ExterCond + Foundation + BsmtCond +
##     X1stFlrSF + X2ndFlrSF + KitchenQual + YrSold + SaleType +
##     SalePrice, data = mlr_train, trace = F)
##
## Coefficients:
##      (Intercept) Condition1Feedr Condition1Norm Condition1PosC
Condition1RRCe
## Good   -3.559262         1.51866      0.7624463       2.798134
1.68603
## Poor  -28.358508       -24.32421      1.0574248       3.834125      -
12.65033
##      Condition1RRCn BldgType2fmCon BldgTypeDuplex BldgTypeTwnhs
BldgTypeTwnhsE
## Good       1.146416     -0.1887174       -1.70812      0.778546      -
1.094744
## Poor      -9.671501    -26.8421876       -1.95188    -13.272746
1.183295
##      YearBuilt RoofMatlCompShg MasVnrArea ExterCondFa&Po ExterCondTA
## Good -1.843954       1.3742309 -0.4904932     -3.2203298  -1.6678025
## Poor  0.163980      -0.4745349 -1.1769709      0.3352838  -0.4611622
##      FoundationCBlock FoundationPConc FoundationSlaStnWood BsmtCondGd
## Good        1.1166904      -0.5784271             2.355348   1.295385
## Poor       -0.2979999      -0.3059164             2.152081 -16.995094
##      BsmtCondNoBsmt BsmtCondTA X1stFlrSF  X2ndFlrSF KitchenQualFa
KitchenQualGd
## Good     -0.6715086 -0.2673351 -1.119677 -0.7464800    -0.4558117
0.5981127
## Poor     -3.0284084 -1.7626112  1.946634  0.5952757    24.0488471   -
10.1355827
##      KitchenQualTA     YrSold SaleTypeOth SaleTypeNew SaleTypeWD SalePrice
## Good    -0.9763158 0.20130647    1.346024   -34.06493  1.0689306  1.743486
## Poor    23.2662161 0.04670367    3.507165   -12.50486  0.3266993 -4.624657
##
```

```
## Std. Errors:
##      (Intercept)   Condition1Feedr Condition1Norm Condition1PosC
Condition1RRCe
## Good    1.355223 0.608246090985727      0.5167238      0.7665007
1.376937000276
## Poor    1.992124 0.000000000894832      1.3692154      2.0336029
0.000004670251
##      Condition1RRCn    BldgType2fmCon BldgTypeDuplex   BldgTypeTwnhs
## Good  0.91273291353 0.558676445622063       1.134795 0.6444889199866
## Poor  0.00002252099 0.000000001114988       1.303338 0.0000006827822
##      BldgTypeTwnhsE YearBuilt RoofMatlCompShg MasVnrArea ExterCondFa&Po
## Good      0.8212007 0.1847345       0.6657349  0.1493340      0.9278635
## Poor      1.3290735 0.6268734       1.7024927  0.9509848      1.7389735
##      ExterCondTA FoundationCBlock FoundationPConc FoundationSlaStnWood
## Good   0.2636843        0.3273121       0.3939789            0.9702282
## Poor   1.2397904        1.0341649       1.2464851            2.4977211
##           BsmtCondGd BsmtCondNoBsmt BsmtCondTA X1stFlrSF X2ndFlrSF
## Good 0.68233106382325       0.985879  0.5019823 0.1970384 0.1570314
## Poor 0.00000006987932       2.426466  0.7528013 0.4812165 0.5505257
##      KitchenQualFa         KitchenQualGd KitchenQualTA     YrSold
SaleTypeOth
## Good     0.7502436 0.5562199510944234593      0.572509 0.09713391
0.9534337
## Poor     1.1962958 0.0000000000002362962      1.067895 0.26407079
1.8237413
##                 SaleTypeNew SaleTypeWD SalePrice
## Good 0.000000000000008309074  0.6584584 0.2583228
## Poor 0.000000893673253824292  1.2562778 1.0756321
##
## Residual Deviance: 814.8605
## AIC: 938.8605

summary(model_mlr_aic_f)

## Call:
## multinom(formula = OverallCond ~ YearBuilt + ExterCond + KitchenQual +
##     Foundation + SalePrice + X1stFlrSF + X2ndFlrSF + MasVnrArea +
##     Condition1 + BsmtCond + SaleType + BldgType + RoofMatl +
##     YrSold, data = mlr_train, trace = F)
##
## Coefficients:
##      (Intercept) YearBuilt ExterCondFa&Po ExterCondTA KitchenQualFa
## Good   -3.559262 -1.843954     -3.2203298  -1.6678025    -0.4558117
## Poor  -28.358508  0.163980      0.3352838  -0.4611622    24.0488471
##      KitchenQualGd KitchenQualTA FoundationCBlock FoundationPConc
## Good     0.5981127    -0.9763158        1.1166904      -0.5784271
## Poor   -10.1355827    23.2662161       -0.2979999      -0.3059164
##      FoundationSlaStnWood SalePrice X1stFlrSF  X2ndFlrSF MasVnrArea
## Good             2.355348  1.743486 -1.119677 -0.7464800 -0.4904932
## Poor             2.152081 -4.624657  1.946634  0.5952757 -1.1769709
```

```
##      Condition1Feedr Condition1Norm Condition1PosC Condition1RRCe
## Good         1.51866      0.7624463       2.798134        1.68603
## Poor       -24.32421      1.0574248       3.834125      -12.65033
##      Condition1RRCn BsmtCondGd BsmtCondNoBsmt BsmtCondTA SaleTypeOth
## Good       1.146416   1.295385     -0.6715086 -0.2673351    1.346024
## Poor      -9.671501 -16.995094     -3.0284084 -1.7626112    3.507165
##      SaleTypeNew SaleTypeWD BldgType2fmCon BldgTypeDuplex BldgTypeTwnhs
## Good   -34.06493  1.0689306     -0.1887174       -1.70812      0.778546
## Poor   -12.50486  0.3266993    -26.8421876       -1.95188    -13.272746
##      BldgTypeTwnhsE RoofMatlCompShg     YrSold
## Good      -1.094744       1.3742309 0.20130647
## Poor       1.183295      -0.4745349 0.04670367
##
## Std. Errors:
##      (Intercept) YearBuilt ExterCondFa&Po ExterCondTA KitchenQualFa
## Good    1.355223 0.1847345      0.9278635   0.2636843     0.7502436
## Poor    1.992124 0.6268734      1.7389735   1.2397904     1.1962958
##              KitchenQualGd KitchenQualTA FoundationCBlock FoundationPConc
## Good 0.5562199510944163539      0.572509        0.3273121       0.3939789
## Poor 0.0000000000002370231      1.067895        1.0341649       1.2464851
##      FoundationSlaStnWood SalePrice X1stFlrSF X2ndFlrSF MasVnrArea
## Good            0.9702282 0.2583228 0.1970384 0.1570314  0.1493340
## Poor            2.4977211 1.0756321 0.4812165 0.5505257  0.9509848
##         Condition1Feedr Condition1Norm Condition1PosC Condition1RRCe
## Good 0.6082460909857169      0.5167238      0.7665007 1.376937000276
## Poor 0.0000000008948289      1.3692154      2.0336029 0.000004670251
##      Condition1RRCn       BsmtCondGd BsmtCondNoBsmt BsmtCondTA SaleTypeOth
## Good  0.91273291353 0.68233106382325       0.985879  0.5019823   0.9534337
## Poor  0.00002252099 0.00000006987932       2.426466  0.7528013   1.8237413
##                 SaleTypeNew SaleTypeWD    BldgType2fmCon BldgTypeDuplex
## Good 0.0000000000001305159  0.6584584 0.558676445622062       1.134795
## Poor 0.0000089367325365981  1.2562778 0.000000001114989       1.303338
##        BldgTypeTwnhs BldgTypeTwnhsE RoofMatlCompShg     YrSold
## Good 0.6444889199866      0.8212007       0.6657349 0.09713391
## Poor 0.0000006827822      1.3290735       1.7024927 0.26407079
##
## Residual Deviance: 814.8605
## AIC: 938.8605
```

```
data.frame(model_mlr_aic_b = c(model_mlr_aic_b$AIC,
model_mlr_aic_b$deviance),
           model_mlr_aic_f = c(model_mlr_aic_f$AIC,
model_mlr_aic_f$deviance),
           row.names = c("AIC", "DEVIANCE"))
```

```
##           model_mlr_aic_b model_mlr_aic_f
## AIC              938.8605        938.8605
## DEVIANCE         814.8605        814.8605
```

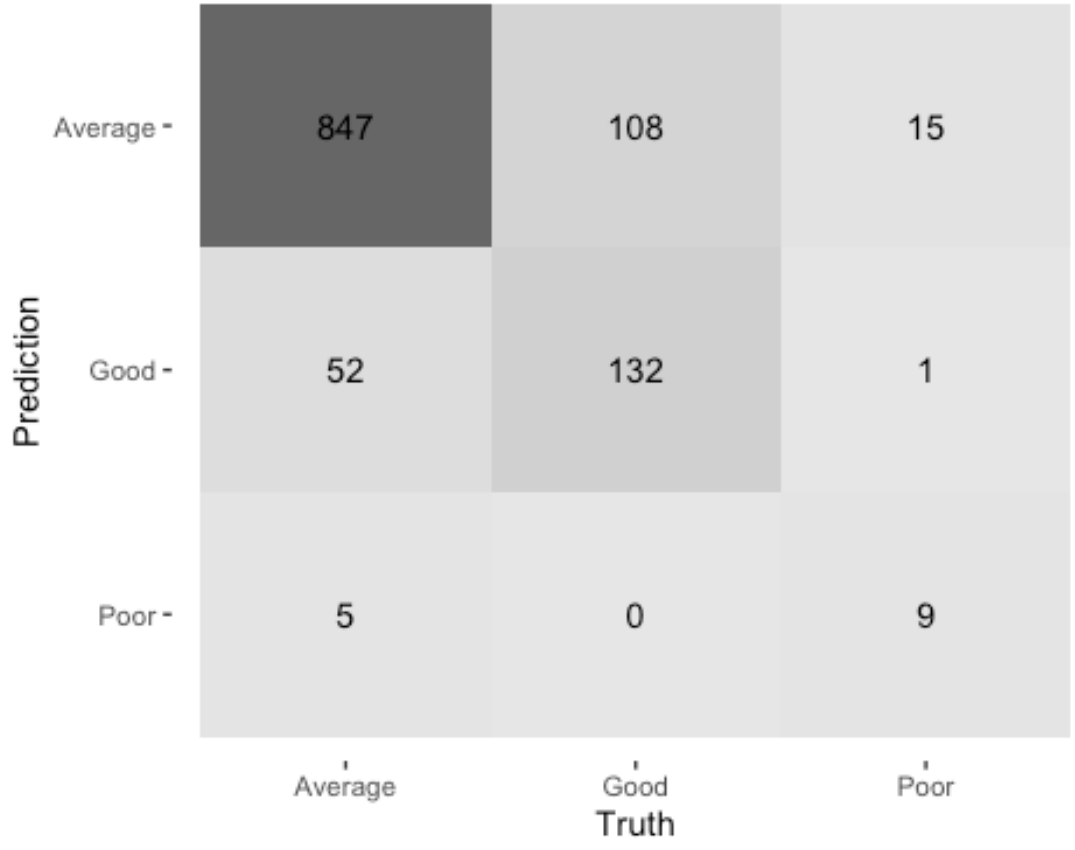
```r
# by working on excluding non-significant predictors, it ended up being the
case
# where forward and backward wrapper methods produced models with the same
# metrics

evaluate_model <- function(model, data){
  predicted <-
    model %>%
    predict(newdata = data, "class")

  correct <- data %>%
    bind_cols(., pred_oc = predicted) %>%
    dplyr::select(OverallCond, pred_oc) %>%
    mutate(correct = OverallCond == pred_oc) %>%
    group_by(correct) %>%
    summarise(count = n())

  print(correct)

  conf_mat_res <- data %>%
    bind_cols(., pred_oc = predicted) %>%
    dplyr::select(OverallCond, pred_oc) %>%
    conf_mat(OverallCond, pred_oc)

  average_tp <- sum(conf_mat_res$table[1,1])
  good_tp <- sum(conf_mat_res$table[2,2])
  poor_tp <- sum(conf_mat_res$table[3,3])

  average_tn <- sum(conf_mat_res$table[-1, -1])
  good_tn <- sum(conf_mat_res$table[-2,-2])
  poor_tn <- sum(conf_mat_res$table[-3,03])

  average_fp <- sum(conf_mat_res$table[1, -1])
  good_fp <- sum(conf_mat_res$table[2,-2])
  poor_fp <- sum(conf_mat_res$table[3,-3])

  average_fn <- sum(conf_mat_res$table[-1,1])
  good_fn <- sum(conf_mat_res$table[-2,2])
  poor_fn <- sum(conf_mat_res$table[-3,3])

  #source: https://towardsdatascience.com/confusion-matrix-for-your-multi-
class-machine-learning-model-ff9aa3bf7826
  precision <- function(tp, fp){tp/(tp+fp)}
  recall <- function(tp, fn){tp/(tp+fn)}
  f1 <- function(tp, fp, fn){2*tp/(2*tp+fp+fn)}

  class_metrics <- data.frame(precision = c(precision(average_tp,
average_fp), precision(good_tp, good_fp), precision(poor_tp, poor_fp)),
                              recall = c(recall(average_tp, average_fn),
```

```
recall(good_tp, good_fn), recall(poor_tp, poor_fn)),
                                f1 = c(f1(average_tp, average_fp, average_fn),
f1(good_tp, good_fp, good_fn), f1(poor_tp, poor_fp, poor_fn)),
                                row.names = c("average", "good", "poor"))

  print(class_metrics)

  # source: https://www.analyticsvidhya.com/blog/2016/02/multinomial-ordinal-
logistic-regression/
  # calculating significance of individual variables
  z <- summary(model)$coefficients/summary(model)$standard.errors
  print('z values:')
  print(z)
  # 2-tailed z test
  print('p values:')
  p <- (1 - pnorm(abs(z), 0, 1))*2
  print(p)

  # ratio of P(choosing one categoty)/P(choosing baseline category) -
relative risk
  # source https://stats.oarc.ucla.edu/r/dae/multinomial-logistic-regression/
  relative_risk <- exp(coef(model))
  print("relative risk")
  print(relative_risk)

  data %>%
    bind_cols(., pred_oc = predicted) %>%
    dplyr::select(OverallCond, pred_oc) %>%
    conf_mat(OverallCond, pred_oc) %>%
    autoplot(type = "heatmap")
}

evaluate_model(model_mlr_aic_f, mlr_train)
```

```
## # A tibble: 2 × 2
##   correct count
##   <lgl>   <int>
## 1 FALSE     181
## 2 TRUE      988
##         precision    recall        f1
## average 0.8731959 0.9369469 0.9039488
## good    0.7135135 0.5500000 0.6211765
## poor    0.6428571 0.3600000 0.4615385
## [1] "z values:"
##       (Intercept) YearBuilt ExterCondFa&Po ExterCondTA KitchenQualFa
## Good    -2.62633 -9.981644     -3.4706935  -6.3249981    -0.6075516
## Poor   -14.23531  0.261584      0.1928056  -0.3719679    20.1027594
##               KitchenQualGd KitchenQualTA FoundationCBlock FoundationPConc
## Good               1.075317     -1.705328        3.4116987      -1.4681677
## Poor -42761999702188.679688     21.786998       -0.2881552      -0.2454233
```

```
##      FoundationSlaStnWood SalePrice X1stFlrSF X2ndFlrSF MasVnrArea
## Good            2.4276228  6.749254 -5.682531 -4.753699  -3.284538
## Poor            0.8616178 -4.299479  4.045235  1.081286  -1.237634
##          Condition1Feedr Condition1Norm Condition1PosC  Condition1RRCe
## Good            2.496785      1.4755396       3.650530        1.224479
## Poor -27183078762.602932      0.7722852       1.885385 -2708704.379214
##      Condition1RRCn       BsmtCondGd BsmtCondNoBsmt BsmtCondTA SaleTypeOth
## Good       1.256026          1.89847     -0.6811268 -0.5325588    1.411764
## Poor -429443.822925 -243206343.11650     -1.2480738 -2.3414029    1.923061
##           SaleTypeNew SaleTypeWD       BldgType2fmCon BldgTypeDuplex
## Good -2610022165593524  1.6233836           -0.3377937      -1.505224
## Poor         -13992657  0.2600534 -24073954878.1987457      -1.497601
##         BldgTypeTwnhs BldgTypeTwnhsE RoofMatlCompShg    YrSold
## Good         1.208005      -1.333102       2.0642314 2.0724632
## Poor -19439208.877614       0.890316      -0.2787295 0.1768604
## [1] "p values:"
##      (Intercept) YearBuilt ExterCondFa&Po        ExterCondTA KitchenQualFa
## Good 0.008631103 0.0000000   0.0005191161 0.0000000002532357     0.5434849
## Poor 0.000000000 0.7936422   0.8471112277 0.7099167799861434     0.0000000
##      KitchenQualGd KitchenQualTA FoundationCBlock FoundationPConc
## Good     0.2822329    0.08813318     0.0006455942       0.1420587
## Poor     0.0000000    0.00000000     0.7732279779       0.8061287
##      FoundationSlaStnWood           SalePrice        X1stFlrSF
X2ndFlrSF
## Good           0.01519814 0.00000000001486078 0.00000001327156
0.00000199728
## Poor           0.38889785 0.00001712001721921 0.00005227066707
0.27956998325
##       MasVnrArea Condition1Feedr Condition1Norm Condition1PosC
Condition1RRCe
## Good 0.001021497      0.01253248      0.1400675   0.0002616999
0.2207716
## Poor 0.215851928      0.00000000      0.4399455   0.0593778330
0.0000000
##      Condition1RRCn BsmtCondGd BsmtCondNoBsmt BsmtCondTA SaleTypeOth
## Good      0.2091065 0.05763415      0.4957913 0.59433901  0.15801935
## Poor      0.0000000 0.00000000      0.2120040 0.01921142  0.05447241
##      SaleTypeNew SaleTypeWD BldgType2fmCon BldgTypeDuplex BldgTypeTwnhs
## Good           0  0.1045074      0.7355187      0.1322665     0.2270452
## Poor           0  0.7948226      0.0000000      0.1342370     0.0000000
##      BldgTypeTwnhsE RoofMatlCompShg     YrSold
## Good      0.1824983      0.03899578 0.03822226
## Poor      0.3732962      0.78045243 0.85961803
## [1] "relative risk"
##                (Intercept) YearBuilt ExterCondFa&Po ExterCondTA
## Good 0.0284598244640643164 0.1581907     0.03994188   0.1886612
## Poor 0.0000000000004831217 1.1781908     1.39833718   0.6305504
##            KitchenQualFa KitchenQualGd       KitchenQualTA
FoundationCBlock
## Good           0.6339332 1.81868320527           0.3766964
```

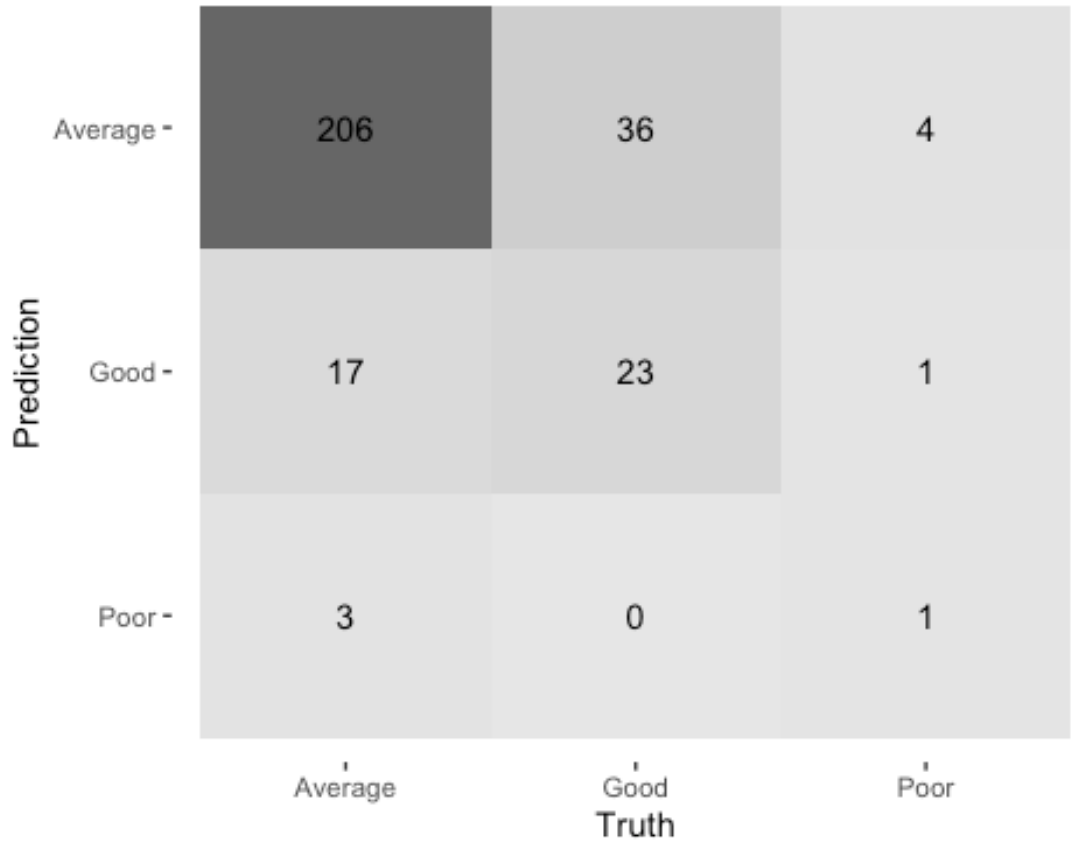
```
3.0547276
## Poor 27815161773.4437065 0.00003964353 12717134985.8223896
0.7423014
##      FoundationPConc FoundationSlaStnWood   SalePrice X1stFlrSF X2ndFlrSF
## Good       0.5607797            10.541797 5.717238631 0.3263853 0.4740322
## Poor       0.7364481             8.602743 0.009807014 7.0050668 1.8135308
##      MasVnrArea     Condition1Feedr Condition1Norm Condition1PosC
## Good  0.6123243 4.56610248435731325       2.143514       16.41399
## Poor  0.3082109 0.00000000002729805       2.878948       46.25294
##      Condition1RRCe Condition1RRCn       BsmtCondGd BsmtCondNoBsmt
BsmtCondTA
## Good   5.3980099047  3.14689549424 3.65240292699834      0.5109372
0.7654165
## Poor   0.0000032065  0.00006305514 0.00000004160298      0.0483926
0.1715962
##      SaleTypeOth             SaleTypeNew SaleTypeWD       BldgType2fmCon
## Good    3.842118 0.00000000000000001606152   2.912263 0.828020496634317982
## Poor   33.353591 0.00000370857488869996   1.386384 0.000000000002200828
##      BldgTypeDuplex  BldgTypeTwnhs BldgTypeTwnhsE RoofMatlCompShg   YrSold
## Good      0.1812061 2.178302793674      0.3346251       3.9520360 1.223000
## Poor      0.1420069 0.000001720757      3.2651166       0.6221743 1.047811
```




```
evaluate_model(model_mlr_aic_f, mlr_test)
```

```
## # A tibble: 2 × 2
##   correct count
##   <lgl>   <int>
## 1 FALSE      61
## 2 TRUE      230
##         precision    recall        f1
## average 0.8373984 0.9115044 0.8728814
## good    0.5609756 0.3898305 0.4600000
## poor    0.2500000 0.1666667 0.2000000
## [1] "z values:"
##      (Intercept) YearBuilt ExterCondFa&Po ExterCondTA KitchenQualFa
## Good    -2.62633 -9.981644     -3.4706935  -6.3249981    -0.6075516
## Poor   -14.23531  0.261584      0.1928056  -0.3719679    20.1027594
##               KitchenQualGd KitchenQualTA FoundationCBlock FoundationPConc
## Good                1.075317     -1.705328        3.4116987      -1.4681677
## Poor -42761999702188.679688     21.786998       -0.2881552      -0.2454233
##      FoundationSlaStnWood SalePrice X1stFlrSF X2ndFlrSF MasVnrArea
## Good            2.4276228  6.749254 -5.682531 -4.753699  -3.284538
## Poor            0.8616178 -4.299479  4.045235  1.081286  -1.237634
##          Condition1Feedr Condition1Norm Condition1PosC  Condition1RRCe
## Good            2.496785      1.4755396       3.650530        1.224479
## Poor -27183078762.602932      0.7722852       1.885385 -2708704.379214
##      Condition1RRCn       BsmtCondGd BsmtCondNoBsmt BsmtCondTA SaleTypeOth
## Good       1.256026          1.89847     -0.6811268 -0.5325588    1.411764
## Poor -429443.822925 -243206343.11650     -1.2480738 -2.3414029    1.923061
##            SaleTypeNew SaleTypeWD       BldgType2fmCon BldgTypeDuplex
## Good -2610022165593524  1.6233836           -0.3377937      -1.505224
## Poor         -13992657  0.2600534 -24073954878.1987457      -1.497601
##         BldgTypeTwnhs BldgTypeTwnhsE RoofMatlCompShg    YrSold
## Good         1.208005      -1.333102       2.0642314 2.0724632
## Poor -19439208.877614       0.890316      -0.2787295 0.1768604
## [1] "p values:"
##      (Intercept) YearBuilt ExterCondFa&Po        ExterCondTA KitchenQualFa
## Good 0.008631103 0.0000000   0.0005191161 0.0000000002532357     0.5434849
## Poor 0.000000000 0.7936422   0.8471112277 0.7099167799861434     0.0000000
##      KitchenQualGd KitchenQualTA FoundationCBlock FoundationPConc
## Good     0.2822329    0.08813318     0.0006455942       0.1420587
## Poor     0.0000000    0.00000000     0.7732279779       0.8061287
##      FoundationSlaStnWood           SalePrice        X1stFlrSF
X2ndFlrSF
## Good           0.01519814 0.00000000001486078 0.00000001327156
0.00000199728
## Poor           0.38889785 0.00001712001721921 0.00005227066707
0.27956998325
##       MasVnrArea Condition1Feedr Condition1Norm Condition1PosC
Condition1RRCe
## Good 0.001021497      0.01253248      0.1400675   0.0002616999
0.2207716
## Poor 0.215851928      0.00000000      0.4399455   0.0593778330
0.0000000
```

```
##      Condition1RRCn BsmtCondGd BsmtCondNoBsmt BsmtCondTA SaleTypeOth
## Good      0.2091065 0.05763415      0.4957913 0.59433901  0.15801935
## Poor      0.0000000 0.00000000      0.2120040 0.01921142  0.05447241
##      SaleTypeNew SaleTypeWD BldgType2fmCon BldgTypeDuplex BldgTypeTwnhs
## Good           0  0.1045074      0.7355187      0.1322665     0.2270452
## Poor           0  0.7948226      0.0000000      0.1342370     0.0000000
##      BldgTypeTwnhsE RoofMatlCompShg     YrSold
## Good      0.1824983      0.03899578 0.03822226
## Poor      0.3732962      0.78045243 0.85961803
## [1] "relative risk"
##                (Intercept) YearBuilt ExterCondFa&Po ExterCondTA
## Good 0.0284598244640643164 0.1581907     0.03994188   0.1886612
## Poor 0.0000000000004831217 1.1781908     1.39833718   0.6305504
##            KitchenQualFa KitchenQualGd       KitchenQualTA
FoundationCBlock
## Good           0.6339332 1.81868320527           0.3766964
3.0547276
## Poor 27815161773.4437065 0.00003964353 12717134985.8223896
0.7423014
##      FoundationPConc FoundationSlaStnWood   SalePrice X1stFlrSF X2ndFlrSF
## Good       0.5607797            10.541797 5.717238631 0.3263853 0.4740322
## Poor       0.7364481             8.602743 0.009807014 7.0050668 1.8135308
##      MasVnrArea     Condition1Feedr Condition1Norm Condition1PosC
## Good  0.6123243 4.56610248435731325       2.143514       16.41399
## Poor  0.3082109 0.00000000002729805       2.878948       46.25294
##      Condition1RRCe Condition1RRCn       BsmtCondGd BsmtCondNoBsmt
BsmtCondTA
## Good   5.3980099047  3.14689549424 3.65240292699834      0.5109372
0.7654165
## Poor   0.0000032065  0.00006305514 0.00000004160298      0.0483926
0.1715962
##      SaleTypeOth             SaleTypeNew SaleTypeWD       BldgType2fmCon
## Good    3.842118 0.000000000000001606152   2.912263 0.828020496634317982
## Poor   33.353591 0.000003708574888699996   1.386384 0.000000000002200828
##      BldgTypeDuplex  BldgTypeTwnhs BldgTypeTwnhsE RoofMatlCompShg   YrSold
## Good      0.1812061 2.178302793674      0.3346251       3.9520360 1.223000
## Poor      0.1420069 0.000001720757      3.2651166       0.6221743 1.047811
```



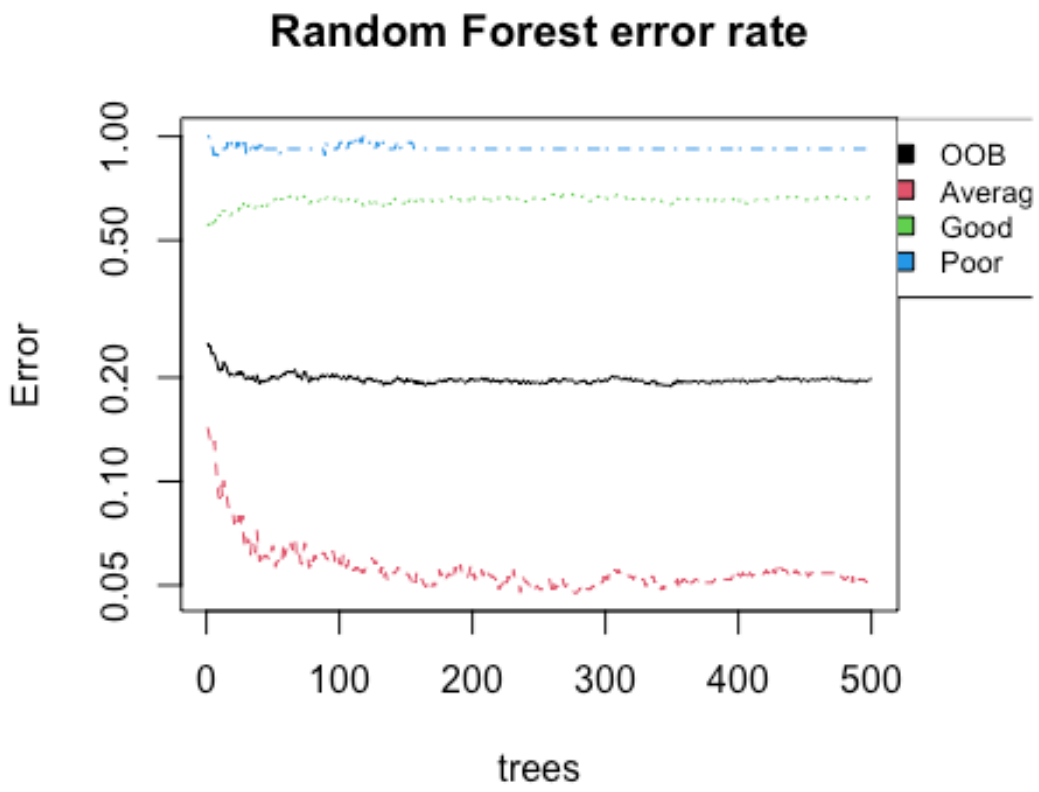

```
# From the output we can see the relation of predictors and the individual
output
# classes C, |C|-1 in total because we have picked the most numerous one
# "Average" as being the reference class.
# For example, the log odds of having Average vs Good house condition seems
to decrease
# by (-)3.4116987 when Foundation becomes CBlock as opposed of BrkTill

# WERONIKA masterplan
============================================================


# Choice of model: random forest
# training the model using 80% of original data
set.seed(123)
table(ytest)

## ytest
## Average    Good    Poor
##     226      59       6

table(ytrain)
```

```
## ytrain
## Average    Good    Poor
##     904     240      25
```

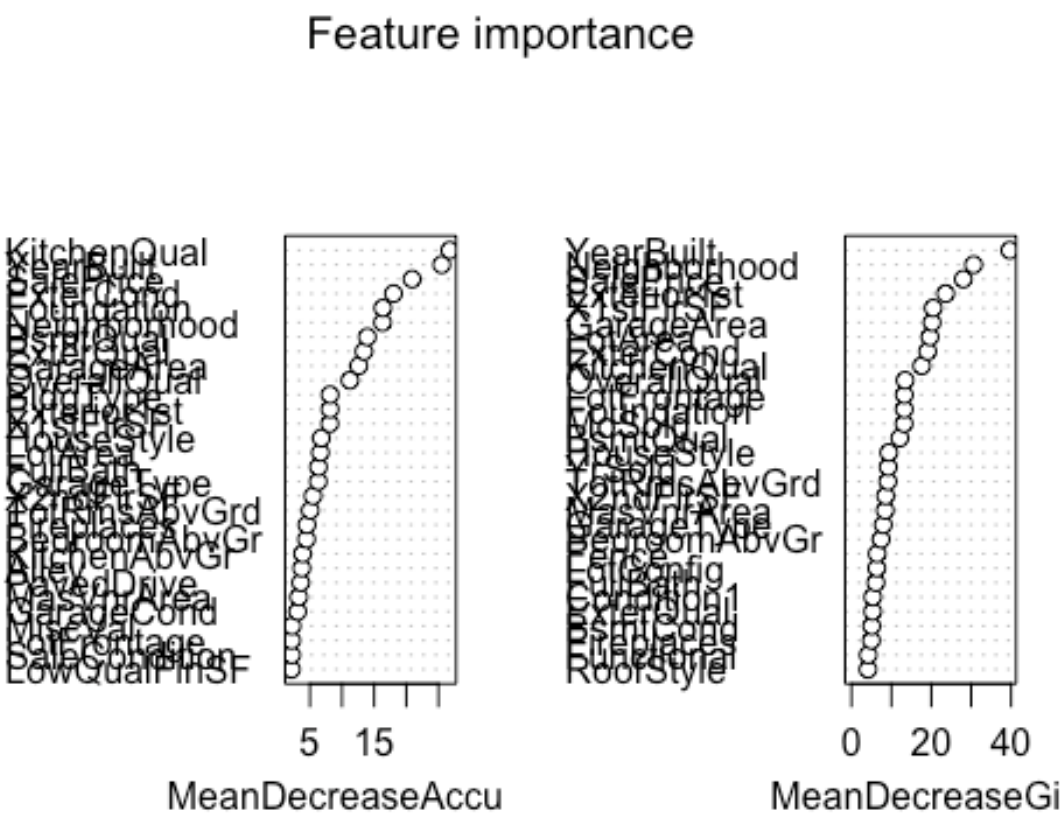
```
rf_train <- randomForest(xtrain, ytrain, method='class', importance = T)
print(rf_train)
```

```
##
## Call:
##  randomForest(x = xtrain, y = ytrain, importance = T, method = "class")
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##         OOB estimate of  error rate: 19.85%
## Confusion matrix:
##         Average Good Poor class.error
## Average     856   48    0  0.05309735
## Good        161   79    0  0.67083333
## Poor         21    2    2  0.92000000
```

```
layout(matrix(c(1,2),nrow=1),
       width=c(4,1))
par(mar=c(5,4,4,0)) #No margin on the right side
plot(rf_train, log="y", main="Random Forest error rate")
par(mar=c(5,0,4,2)) #No margin on the left side
plot(c(0,1),type="n", axes=F, xlab="", ylab="")
legend("top", colnames(rf_train$err.rate),col=1:4,cex=0.8,fill=1:4)
```




```
# Plotting the most significant variables for the accuracy of the model
varImpPlot(rf_train, main="Feature importance")
```




```
# Prediction using 20% of original data
prediction_rf <- predict(rf_train, xtest)


# Evaluating the model using the accuracy metric
confusion_matrix_rf <- confusionMatrix(prediction_rf, ytest)
confusion_matrix_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Average Good Poor
##    Average     215   35    5
##    Good         11   24    1
##    Poor          0    0    0
##
## Overall Statistics
##
##                Accuracy : 0.8213
##                  95% CI : (0.7724, 0.8636)
##     No Information Rate : 0.7766
##     P-Value [Acc > NIR] : 0.0367336
##
```

```
##                   Kappa : 0.3929
##
##  Mcnemar's Test P-Value : 0.0003433
##
## Statistics by Class:
##
##                      Class: Average Class: Good Class: Poor
## Sensitivity                  0.9513     0.40678     0.00000
## Specificity                  0.3846     0.94828     1.00000
## Pos Pred Value               0.8431     0.66667         NaN
## Neg Pred Value               0.6944     0.86275     0.97938
## Prevalence                   0.7766     0.20275     0.02062
## Detection Rate               0.7388     0.08247     0.00000
## Detection Prevalence         0.8763     0.12371     0.00000
## Balanced Accuracy            0.6680     0.67753     0.50000
```

```
#before part3, we clean up the environment
rm(confusion_matrix_rf,
   index,
   mlr_data,
   mlr_test,
   mlr_train,
   rf_train,
   xtest,
   xtrain,
   prediction_rf,
   ytest,
   ytrain)
```

```
summary(house.data)
```

```
##   LotFrontage         LotArea            Alley        LotConfig
##  Min.   :-1.6623   Min.   :-0.9234   Grvl   :  50   Corner : 263
##  1st Qu.:-0.4507   1st Qu.:-0.2969   NoAlley:1369   CulDSac:  94
##  Median : 0.1551   Median :-0.1040   Pave   :  41   FR2&3  :  51
##  Mean   : 0.0000   Mean   : 0.0000                  Inside :1052
##  3rd Qu.: 0.6167   3rd Qu.: 0.1087
##  Max.   : 7.3671   Max.   :20.5112
##
##   Neighborhood  Condition1      Condition2     BldgType     HouseStyle
##  NAmes  :225   Artery:  48   Not-Norm:  15   1Fam  :1220   1.5Fin:154
##  CollgCr:150   Feedr :  81   Norm    :1445   2fmCon:  31   1.5Unf: 14
##  OldTown:113   Norm  :1260                   Duplex:  52   1Story:726
##  Edwards:100   PosC  :  27                   Twnhs :  43   2.5All: 19
##  Somerst: 86   RRCe  :  13                   TwnhsE: 114   2Story:445
##  Gilbert: 79   RRCn  :  31                                 SFoyer: 37
##  (Other):707                                               SLvl  : 65
##   OverallQual   OverallCond     YearBuilt        RoofStyle
```

```
##  5      :397   Average:1130   Min.   :-3.28670   Flat :  13
##  6      :374   Good   : 299   1st Qu.:-0.57173   Gable:1141
##  7      :319   Poor   :  31   Median : 0.05735   Other:  20
##  8      :168                  Mean   : 0.00000   Hip  : 286
##  4      :116                  3rd Qu.: 0.95131
##  9      : 43                  Max.   : 1.28240
##  (Other): 43
##        RoofMatl     Exterior1st    MasVnrArea      ExterQual ExterCond
##  Not-CompShg:  26   VinylSd:515   Min.   :-0.5706   Ex: 52    Ex&Gd: 149
##  CompShg    :1434   HdBoard:222   1st Qu.:-0.5706   Fa: 14    Fa&Po:  29
##                     MetalSd:220   Median :-0.5706   Gd:488    TA   :1282
##                     Wd Sdng:206   Mean   : 0.0000   TA:906
##                     Plywood:108   3rd Qu.: 0.3383
##                     CemntBd: 61   Max.   : 8.2824
##                     (Other):128
##       Foundation    BsmtQual     BsmtCond     Heating       X1stFlrSF
##  BrkTil    :146   Ex    :121   Fa&Po :  47   Other:  14   Min.   :-2.1434
##  CBlock    :634   Fa    : 35   Gd    :  65   GasA :1428   1st Qu.:-0.7259
##  PConc     :647   Gd    :618   NoBsmt:  37   GasW :  18   Median :-0.1956
##  SlaStnWood: 33   NoBsmt: 37   TA    :1311                Mean   : 0.0000
##                   TA    :649                              3rd Qu.: 0.5914
##                                                           Max.   : 9.1296
##
##    X2ndFlrSF        LowQualFinSF        FullBath       BedroomAbvGr
##  Min.   :-0.7949   Min.   :-0.1202   Min.   :-2.8408   0  :  6
##  1st Qu.:-0.7949   1st Qu.:-0.1202   1st Qu.:-1.0257   1  : 50
##  Median :-0.7949   Median :-0.1202   Median : 0.7895   2  :358
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   3  :804
##  3rd Qu.: 0.8728   3rd Qu.:-0.1202   3rd Qu.: 0.7895   4  :213
##  Max.   : 3.9356   Max.   :11.6438   Max.   : 2.6046   >=5: 29
##
##   KitchenAbvGr     KitchenQual  TotRmsAbvGrd          Functional
Fireplaces
##  Min.   :-4.7499   Ex:100      Min.   :-2.7795   Maj1-2+Sev:  20   0  :690
##  1st Qu.:-0.2114   Fa: 39      1st Qu.:-0.9338   Min1      :  31   1  :650
##  Median :-0.2114   Gd:586      Median :-0.3186   Min2      :  34   >=2:120
##  Mean   : 0.0000   TA:735      Mean   : 0.0000   Mod       :  15
##  3rd Qu.:-0.2114               3rd Qu.: 0.2967   Typ       :1360
##  Max.   : 8.8656               Max.   : 4.6033
##
##     GarageType    GarageArea          GarageCond   PavedDrive     Fence
##  2TypCarP: 15   Min.   :-2.21220   Ex&Gd   :  11   N:  90     GdPrv  :  59
##  Attchd  :870   1st Qu.:-0.64769   Fa&Po   :  42   P:  30     GdWo   :  54
##  Basment : 19   Median : 0.03283   NoGarage:  81   Y:1340     MnPrv  : 157
##  BuiltIn : 88   Mean   : 0.00000   TA      :1326              MnWw   :  11
##  Detchd  :387   3rd Qu.: 0.48184                              NoFence:1179
##  NoGarage: 81   Max.   : 4.42001
##
##   MiscFeature      MiscVal             MoSold            YrSold
##  MiscF  :  54   Min.   :-0.08766   Min.   :-1.9684   Min.   :-1.3672
```

```
##  NoMiscF:1406   1st Qu.:-0.08766   1st Qu.:-0.4889   1st Qu.:-0.6142
##                 Median :-0.08766   Median :-0.1191   Median : 0.1387
##                 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##                 3rd Qu.:-0.08766   3rd Qu.: 0.6207   3rd Qu.: 0.8917
##                 Max.   :31.15459   Max.   : 2.1002   Max.   : 1.6446
##
##  SaleType       SaleCondition    SalePrice
##  COD:  43   Abnorml    : 101   Min.   :-1.8381
##  Oth:  28   AdjL&Alloca:  16   1st Qu.:-0.6413
##  New: 122   Family     :  20   Median :-0.2256
##  WD :1267   Normal     :1198   Mean   : 0.0000
##             Partial    : 125   3rd Qu.: 0.4164
##                                Max.   : 7.2263
##
```

```r
# partitioning the model for training and testing set (80/20 split)
set.seed(16)
samp <- createDataPartition(house.data$SalePrice, p = 0.8, list = FALSE)
training <- house.data[samp,]
testing <- house.data[-samp,]
x_test <- testing[,1:43]
y_test <- testing[,44]

set.seed(16)

# linear regression model using all the variables (regression based)
lm_full_model <- train(SalePrice~., data = training, method = "lm")
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

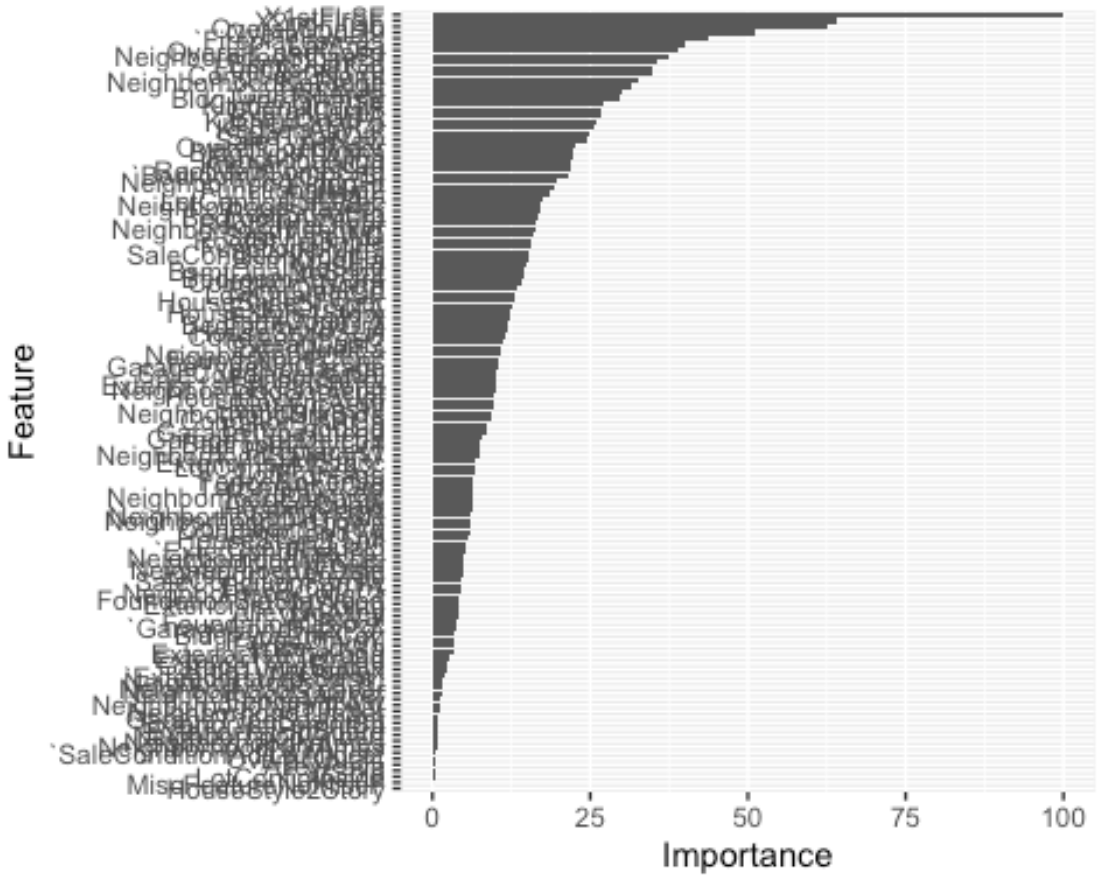
```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

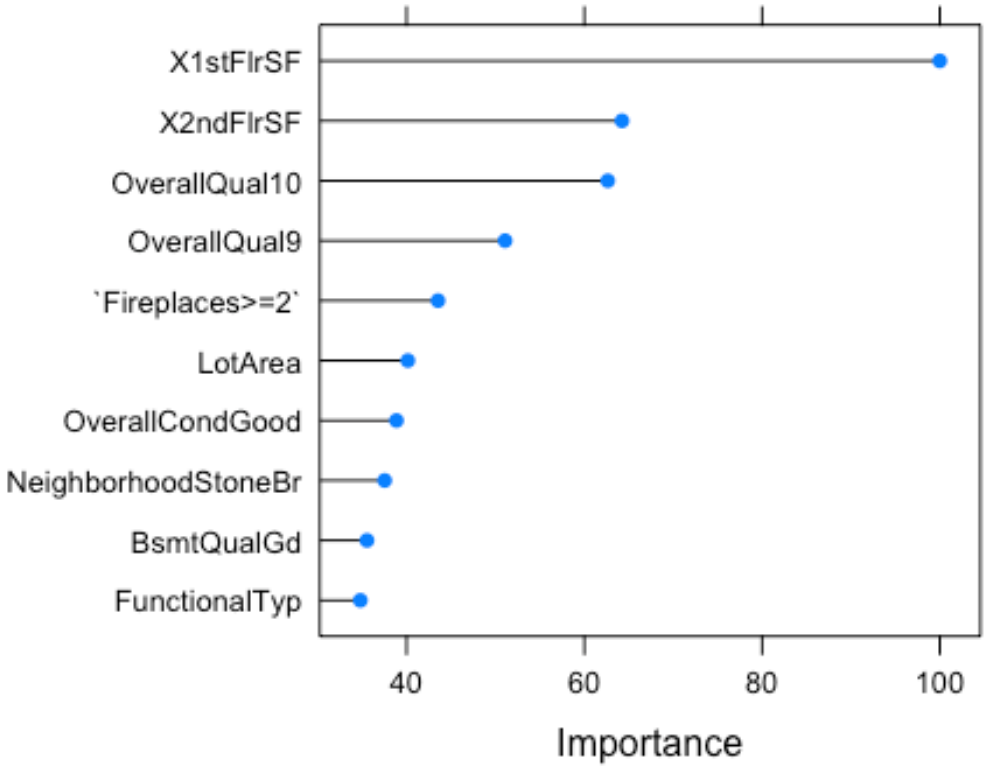
```
# random forest model using all the variables (tree based)
tic("randomForest full model")
rf_full_model <- train(SalePrice~., data = training, method = "rf")
toc()
```

```
## randomForest full model: 433.396 sec elapsed
```

```
# checking most important variables for the linear regression model
lm_imp<- varImp(lm_full_model, xlim = c(50, 100))
ggplot(lm_imp)
```




```
plot(lm_imp, top = 10)
```

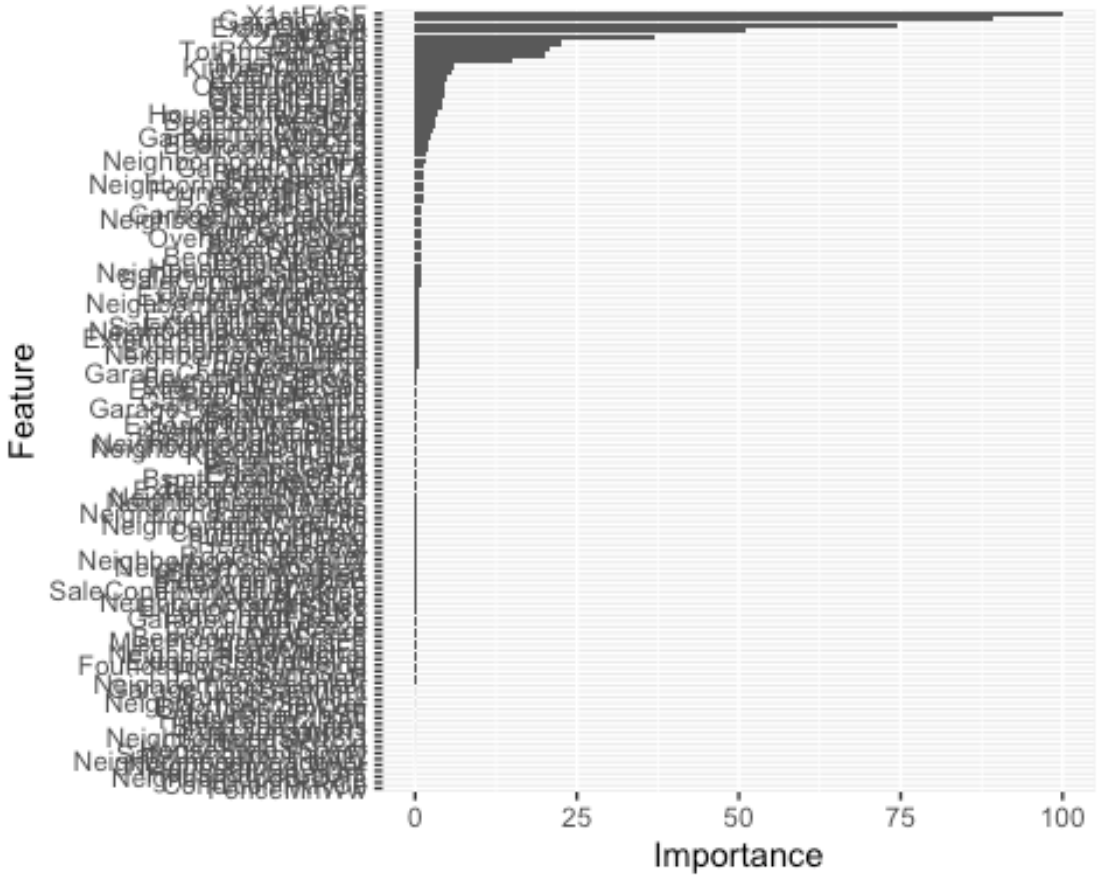



```
lm_imp
```

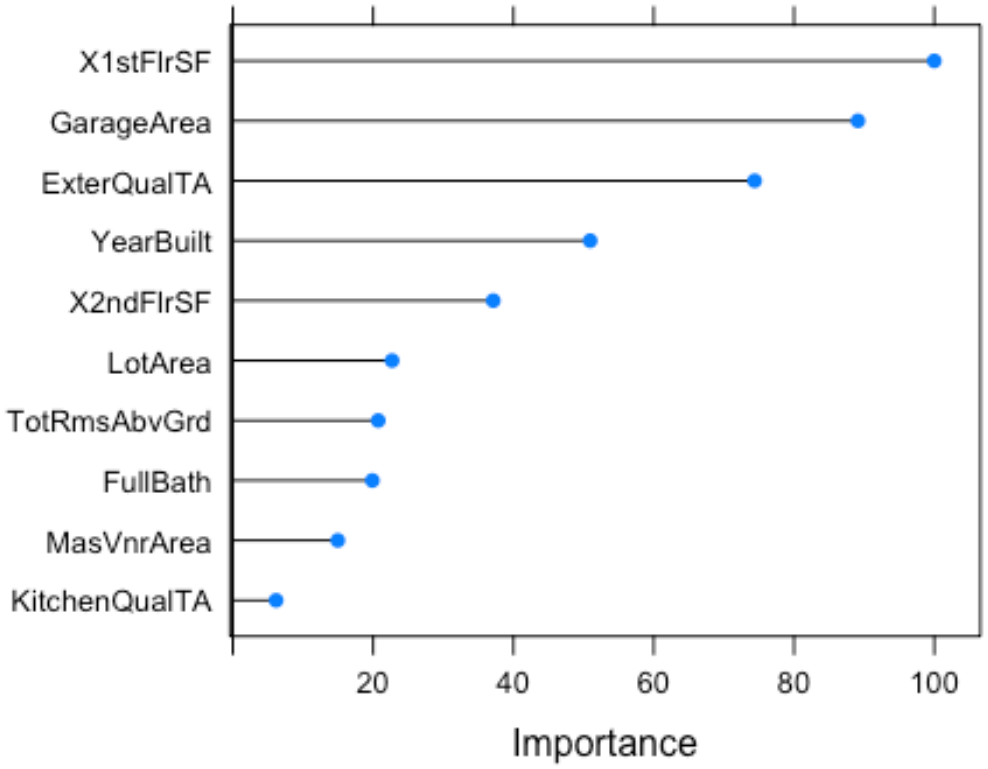
```
## lm variable importance
##
##   only 20 most important variables shown (out of 131)
##
##                       Overall
## X1stFlrSF              100.00
## X2ndFlrSF               64.24
## OverallQual10           62.64
## OverallQual9            51.08
## `Fireplaces>=2`         43.54
## LotArea                 40.14
## OverallCondGood         38.86
## NeighborhoodStoneBr     37.54
## BsmtQualGd              35.54
## FunctionalTyp           34.80
## Condition2Norm          34.68
## NeighborhoodNoRidge     32.48
## YearBuilt               31.34
## GarageArea              30.05
## BldgTypeTwnhsE          29.53
## KitchenQualTA           26.98
```

```
## OverallQual8          26.87
## BsmtQualTA            26.78
## KitchenQualFa         25.92
## KitchenAbvGr          25.63
```

```r
# checking most important variables for the random forest model
rf_imp <- varImp(rf_full_model, xlim = c(20,100))
ggplot(rf_imp)
```




```r
plot(rf_imp, top = 10)
```




```
rf_imp
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 133)
##
##                  Overall
## X1stFlrSF        100.000
## GarageArea        89.115
## ExterQualTA       74.388
## YearBuilt         50.942
## X2ndFlrSF         37.120
## LotArea           22.696
## TotRmsAbvGrd      20.720
## FullBath          19.885
## MasVnrArea        14.959
## KitchenQualTA      6.166
## LotFrontage        5.580
## ExterQualGd        5.039
## OverallQual10      4.582
## OverallQual8       4.479
## OverallQual7       4.390
## OverallQual9       4.364
```

```
## BsmtQualGd         4.243
## HouseStyle2Story   3.446
## BedroomAbvGr4      3.210
## MoSold             3.161

# Now that we know what the most important variables are with respect to the
dataset,
# we can model accordingly to get the most accurate model using both, a
refression
# model and a tree based model.

# linear regression model using the most important variables out of all of
them (regression based)
lm_selected_model <-
train(SalePrice~OverallQual+X2ndFlrSF+OverallCond+X1stFlrSF+Fireplaces+BsmtQu
al+KitchenQual+Neighborhood+BldgType,
                           data = training, method = "lm")

# random forest model using the most important variables out of all of them
(tree based)
tic("randomForest reduced model")
rf_selected_model <-
train(SalePrice~OverallQual+X2ndFlrSF+OverallCond+X1stFlrSF+Fireplaces+BsmtQu
al+KitchenQual+Neighborhood,
                           data = training, method = "rf")
toc()

## randomForest reduced model: 160.629 sec elapsed

# getting the insight about the model
lm_full_model$finalModel

##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Coefficients:
##                (Intercept)                 LotFrontage
##                -1.21848904                  0.01048271
##                    LotArea                AlleyNoAlley
##                 0.06569930                 -0.03496698
##                  AlleyPave            LotConfigCulDSac
##                 0.00640774                  0.11627628
##           `LotConfigFR2&3`              LotConfigInside
##                -0.05388159                 -0.00177122
##          NeighborhoodOther           NeighborhoodBrDale
##                 0.22942259                  0.10262952
##        NeighborhoodBrkSide         NeighborhoodClearCr
##                 0.17549249                  0.03007490
##        NeighborhoodCollgCr         NeighborhoodCrawfor
##                 0.06721669                  0.30268215
```

```
##        NeighborhoodEdwards         NeighborhoodGilbert
##                 -0.10578316                  0.01306791
##         NeighborhoodIDOTRR         NeighborhoodMeadowV
##                 -0.11724321                 -0.02713519
##        NeighborhoodMitchel           NeighborhoodNAmes
##                 -0.08539234                 -0.01306519
##        NeighborhoodNoRidge         NeighborhoodNridgHt
##                  0.58611249                  0.30055149
##         NeighborhoodNWAmes         NeighborhoodOldTown
##                  0.01107498                 -0.10725491
##         NeighborhoodSawyer         NeighborhoodSawyerW
##                  0.02729680                  0.12187396
##        NeighborhoodSomerst         NeighborhoodStoneBr
##                  0.15774218                  0.62845901
##          NeighborhoodSWISU          NeighborhoodTimber
##                 -0.03308060                  0.01817229
##        NeighborhoodVeenker             Condition1Feedr
##                  0.35224339                  0.04982005
##             Condition1Norm              Condition1PosC
##                  0.12008037                 -0.14687009
##             Condition1RRCe              Condition1RRCn
##                 -0.10585436                  0.12422270
##             Condition2Norm              BldgType2fmCon
##                  0.47991102                  0.03956789
##             BldgTypeDuplex              BldgTypeTwnhs
##                 -0.02795407                 -0.24046093
##             BldgTypeTwnhsE            HouseStyle1.5Unf
##                 -0.22990290                  0.15036042
##           HouseStyle1Story            HouseStyle2.5All
##                  0.09633154                 -0.08884982
##           HouseStyle2Story            HouseStyleSFoyer
##                 -0.00009378                  0.15426212
##             HouseStyleSLvl                OverallQual4
##                  0.11411406                  0.00649489
##                OverallQual5                OverallQual6
##                  0.06432839                  0.08182004
##                OverallQual7                OverallQual8
##                  0.15162144                  0.38809296
##                OverallQual9               OverallQual10
##                  0.90497890                  1.32807585
##             OverallCondGood             OverallCondPoor
##                  0.16001782                 -0.23835251
##                   YearBuilt              RoofStyleGable
##                  0.13166433                  0.33960630
##              RoofStyleOther                RoofStyleHip
##                  0.38733336                  0.36883082
##             RoofMatlCompShg      Exterior1stBrkCinStone
##                 -0.33855612                  0.14430391
##          Exterior1stCemntBd          Exterior1stHdBoard
##                  0.01412374                  0.01203358
```

```
##        Exterior1stAllStucc          Exterior1stMetalSd
##                 0.10341468                  0.03364699
##         Exterior1stPlywood          Exterior1stVinylSd
##                 0.04512268                  0.06061531
##       `Exterior1stWd Sdng`          Exterior1stWdShing
##                 0.02348879                 -0.06662740
##                 MasVnrArea                 ExterQualFa
##                 0.01684965                 -0.21669201
##                ExterQualGd                 ExterQualTA
##                -0.13967155                 -0.14276161
##            `ExterCondFa&Po`                 ExterCondTA
##                -0.06279836                 -0.02181666
##           FoundationCBlock             FoundationPConc
##                 0.02490042                  0.07438484
##       FoundationSlaStnWood                  BsmtQualFa
##                -0.05940177                 -0.18968201
##                 BsmtQualGd              BsmtQualNoBsmt
##                -0.24750402                 -0.24736509
##                 BsmtQualTA                  BsmtCondGd
##                -0.22483336                  0.09474171
##              BsmtCondNoBsmt                  BsmtCondTA
##                         NA                  0.10045717
##                HeatingGasA                 HeatingGasW
##                 0.09779944                  0.18906409
##                  X1stFlrSF                   X2ndFlrSF
##                 0.33208519                  0.31851657
##                LowQualFinSF                    FullBath
##                 0.02089621                  0.04320892
##              BedroomAbvGr1               BedroomAbvGr2
##                -0.16671831                 -0.25341773
##              BedroomAbvGr3               BedroomAbvGr4
##                -0.35719548                 -0.31906524
##           `BedroomAbvGr>=5`                KitchenAbvGr
##                -0.53430932                 -0.06746838
##              KitchenQualFa               KitchenQualGd
##                -0.31028943                 -0.16004456
##              KitchenQualTA                TotRmsAbvGrd
##                -0.21541509                  0.02011229
##             FunctionalMin1              FunctionalMin2
##                 0.25026586                  0.30277379
##              FunctionalMod               FunctionalTyp
##                 0.41261212                  0.46267982
##                Fireplaces1             `Fireplaces>=2`
##                 0.02698247                  0.26385627
##            GarageTypeAttchd           GarageTypeBasment
##                 0.12082647                 -0.01913679
##           GarageTypeBuiltIn           GarageTypeDetchd
##                 0.03761026                  0.10857147
##          GarageTypeNoGarage                  GarageArea
##                 0.27090954                  0.06958520
```

```
##          `GarageCondFa&Po`          GarageCondNoGarage
##                 0.08271938                          NA
##               GarageCondTA                PavedDriveP
##                 0.12339961                -0.07263144
##                PavedDriveY                  FenceGdWo
##                -0.02311026                 0.03301777
##                 FenceMnPrv                  FenceMnWw
##                 0.08300029                 0.01985951
##               FenceNoFence          MiscFeatureNoMiscF
##                 0.04725754                -0.00024512
##                    MiscVal                      MoSold
##                 0.00543437                -0.02041155
##                     YrSold                SaleTypeOth
##                 0.00070219                 0.33243748
##                SaleTypeNew                 SaleTypeWD
##                 0.70133036                 0.14192876
## `SaleConditionAdjL&Alloca`         SaleConditionFamily
##                -0.00845555                 0.06533751
##        SaleConditionNormal        SaleConditionPartial
##                 0.09019403                -0.27844387
```

```
lm_selected_model$finalModel
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Coefficients:
##         (Intercept)         OverallQual4         OverallQual5
##           0.6012717           -0.0361491           -0.0089074
##        OverallQual6         OverallQual7         OverallQual8
##           0.0719807            0.2116651            0.4818463
##        OverallQual9        OverallQual10            X2ndFlrSF
##           1.0850967            1.4979491            0.2503064
##     OverallCondGood      OverallCondPoor            X1stFlrSF
##           0.1288509           -0.3920610            0.3513705
##         Fireplaces1      `Fireplaces>=2`           BsmtQualFa
##           0.0234044            0.2962992           -0.5271764
##          BsmtQualGd       BsmtQualNoBsmt           BsmtQualTA
##          -0.3819929           -0.6914294           -0.4742645
##       KitchenQualFa        KitchenQualGd        KitchenQualTA
##          -0.4461709           -0.1918601           -0.3140792
##   NeighborhoodOther   NeighborhoodBrDale  NeighborhoodBrkSide
##           0.0443166           -0.0768416           -0.3161034
## NeighborhoodClearCr  NeighborhoodCollgCr  NeighborhoodCrawfor
##          -0.0233318            0.0006788           -0.0433921
## NeighborhoodEdwards  NeighborhoodGilbert   NeighborhoodIDOTRR
##          -0.3645021           -0.0323329           -0.5259318
## NeighborhoodMeadowV  NeighborhoodMitchel    NeighborhoodNAmes
##          -0.2468946           -0.1933630           -0.2496152
```

```
## NeighborhoodNoRidge  NeighborhoodNridgHt   NeighborhoodNWAmes
##           0.6355882           0.3111419           -0.1787062
## NeighborhoodOldTown   NeighborhoodSawyer  NeighborhoodSawyerW
##          -0.5447687          -0.2168945           -0.0663159
## NeighborhoodSomerst  NeighborhoodStoneBr    NeighborhoodSWISU
##           0.1530134           0.5916758           -0.4984069
##  NeighborhoodTimber  NeighborhoodVeenker       BldgType2fmCon
##          -0.0074779           0.2851517           -0.0953813
##      BldgTypeDuplex        BldgTypeTwnhs       BldgTypeTwnhsE
##          -0.1940818          -0.3757718           -0.2655172

rf_full_model$finalModel

##
## Call:
##  randomForest(x = x, y = y, mtry = min(param$mtry, ncol(x)))
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 67
##
##           Mean of squared residuals: 0.1659169
##                     % Var explained: 84.03

rf_selected_model$finalModel

##
## Call:
##  randomForest(x = x, y = y, mtry = min(param$mtry, ncol(x)))
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 22
##
##           Mean of squared residuals: 0.1601754
##                     % Var explained: 84.58

# evaluating the model based just on the metrics
lm_selected_model$results

##   intercept      RMSE  Rsquared       MAE     RMSESD RsquaredSD      MAESD
## 1      TRUE 0.4023806 0.8538142 0.2566676 0.03746723 0.02571973 0.01022954

rf_selected_model$results

##   mtry      RMSE  Rsquared       MAE     RMSESD RsquaredSD      MAESD
## 1    2 0.5213382 0.8040252 0.3366715 0.04282532 0.01984177 0.01535198
## 2   22 0.4212257 0.8328002 0.2690463 0.04140520 0.02848882 0.01352567
## 3   43 0.4461024 0.8109636 0.2812737 0.04772990 0.03424725 0.01504456

# listing models to compare them agaisnt each other
model_list <- list(lm = lm_selected_model, rf = rf_selected_model)
res <- resamples(model_list)
summary(res)
```

```
##
## Call:
## summary.resamples(object = res)
##
## Models: lm, rf
## Number of resamples: 25
##
## MAE
##         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm 0.2360173 0.2501714 0.2575019 0.2566676 0.2627016 0.2822663    0
## rf 0.2381163 0.2603980 0.2670130 0.2690463 0.2815752 0.2860086    0
##
## RMSE
##         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm 0.3472806 0.3653557 0.4005705 0.4023806 0.4364831 0.4704909    0
## rf 0.3368867 0.3998349 0.4249551 0.4212257 0.4466673 0.5071835    0
##
## Rsquared
##         Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm 0.7964739 0.8291263 0.8663065 0.8538142 0.8753026 0.8841061    0
## rf 0.7759317 0.8149591 0.8365592 0.8328002 0.8507142 0.8841256    0
```

```
compare_models(lm_selected_model, rf_selected_model)
```

```
##
##  One Sample t-test
##
## data:  x
## t = -2.2709, df = 24, p-value = 0.03241
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.035972692 -0.001717528
## sample estimates:
##   mean of x
## -0.01884511
```

```
# making predictions for SalePrice based on the train/test data
pred_lm <- as.data.frame(predict(lm_selected_model, x_test))
names(pred_lm)[1] = "fit"
pred_lm$real = y_test
pred_lm$error = pred_lm$fit - pred_lm$real
print(mean(pred_lm$error^2))
```

```
## [1] 0.2321264
```

```
# [1] 0.1344096

pred_rf <- as.data.frame(predict(rf_selected_model, x_test))
names(pred_rf)[1] = "fit"
pred_rf$real = y_test
```

```r
pred_rf$error = pred_rf$fit - pred_rf$real
print(mean(pred_rf$error^2))
```

```
## [1] 0.1511958
```

```r
# [1] 0.1318999

print("Section 3b")
```

```
## [1] "Section 3b"
```

```r
#Leave-One-Out Cross-Validation

#defines the cross-validation method
cvmethod <- trainControl(method = 'LOOCV')

#creates a linear regression model and performs the LOOCV method
modelloocv1 <- train(SalePrice ~
Fireplaces+TotRmsAbvGrd+Foundation+X1stFlrSF, data=house.data, method= 'lm',
trControl=cvmethod, na.action=na.exclude)
modelloocv2 <- train(SalePrice ~., data=house.data, method= 'lm',
trControl=cvmethod, na.action=na.exclude)
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
```

```
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
```

```
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

#shows the results of the LOOCV
print(modelloocv1)

## Linear Regression
##
## 1460 samples
##    4 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 1459, 1459, 1459, 1459, 1459, 1459, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.6179808  0.6178828  0.4242605
##
## Tuning parameter 'intercept' was held constant at a value of TRUE

print(modelloocv2)
```

```
## Linear Regression
##
## 1460 samples
##   43 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 1459, 1459, 1459, 1459, 1459, 1459, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.4123231  0.8310898  0.2374757
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
#K-Fold Cross Validation

#defines the cross-validation method
cvmethod <- trainControl(method = 'cv', number=10)

#creates a linear regression model and performs the k-fold classification
method
modelkfold1 <- train(SalePrice ~Fireplaces+TotRmsAbvGrd+Foundation+X1stFlrSF,
data=house.data, method= 'lm', trControl=cvmethod, na.action=na.exclude)
modelkfold2 <- train(SalePrice ~., data=house.data, method= 'lm',
trControl=cvmethod, na.action=na.exclude)
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading

## Warning in predict.lm(modelFit, newdata): prediction from a rank-deficient
fit
## may be misleading
```

```
#shows the predictions made across each fold from 1 to 10
modelkfold1$resample
```

```
##         RMSE  Rsquared       MAE Resample
## 1  0.5434546 0.6914243 0.3992270   Fold01
## 2  0.6128202 0.6366057 0.4414733   Fold02
## 3  0.5070511 0.6885089 0.3816738   Fold03
## 4  0.7629111 0.4698470 0.4936693   Fold04
## 5  0.6404822 0.6402016 0.4222857   Fold05
## 6  0.6851551 0.6112270 0.4198590   Fold06
## 7  0.6035207 0.5897320 0.4283801   Fold07
## 8  0.5311594 0.6542722 0.3974995   Fold08
## 9  0.5984467 0.6284004 0.4362167   Fold09
## 10 0.6538886 0.6610170 0.4288158   Fold10
```

```
modelkfold2$resample
```

```
##         RMSE  Rsquared       MAE Resample
## 1  0.3357891 0.8900576 0.2348783   Fold01
## 2  0.3420586 0.8610316 0.2249609   Fold02
## 3  0.3204826 0.9075410 0.2087111   Fold03
## 4  0.5311853 0.8203871 0.2556736   Fold04
## 5  0.3056975 0.8608605 0.2214889   Fold05
## 6  0.3779676 0.8565040 0.2533991   Fold06
## 7  0.6461793 0.6965650 0.2548312   Fold07
## 8  0.3418823 0.8590699 0.2563450   Fold08
## 9  0.3972089 0.8761593 0.2417966   Fold09
## 10 0.3607933 0.8683603 0.2341084   Fold10
```

```
#shows the results of the k-fold classification method
print(modelkfold1)
```

```
## Linear Regression
##
```

```
## 1460 samples
##    4 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1315, 1314, 1314, 1313, 1315, 1314, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   0.613889  0.6271236  0.42491
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
print(modelkfold2)
```

```
## Linear Regression
##
## 1460 samples
##   43 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1315, 1314, 1313, 1314, 1313, 1315, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.3959245  0.8496536  0.2386193
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
# cleaning the environment
rm(lm_imp,
   model_list,
   predictions_rf,
   res,
   rf_imp,
   samp,
   testing,
   training,
   x_test,
   y_test,
   pred_lm
)
```

```
## Warning in rm(lm_imp, model_list, predictions_rf, res, rf_imp, samp,
testing, :
## object 'predictions_rf' not found
```

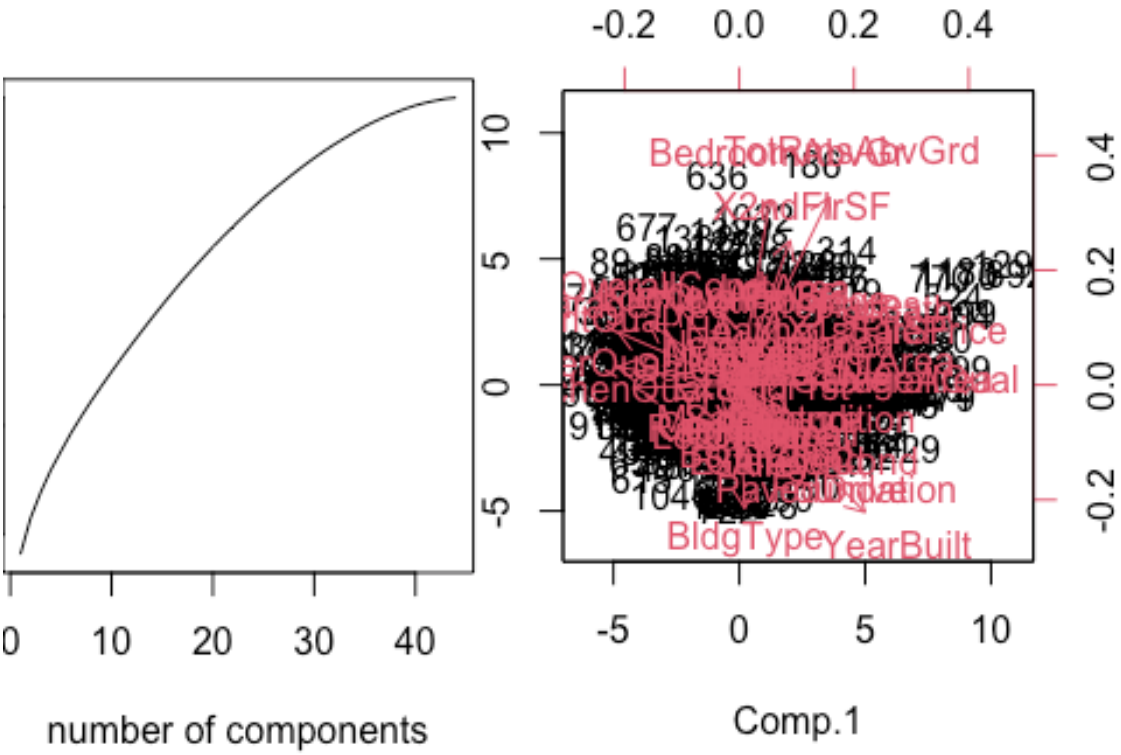
```
#Our question is on if the data naturally falls into clusters based on
neighborhoods
#We will be comparing the cluster results to the clustered form by
```

```r
OverallCond

#first, a numeric version of the data
house.data.num = house.data

#this loops turns all numeric columns into factors
for (i in names(house.data.num)) {
  if (is.factor(house.data.num[[i]])) {
    house.data.num[[i]] = as.numeric(house.data.num[[i]])
  }
}

#We can employ K-Means to see if there are any inherit groups within the data

#lets start by doing a PCA transformation using the correlation matrix over
the covariance
x.pca = princomp(house.data.num, cor = TRUE)
x.pca.test = prcomp(house.data.num)
#summaries of the transformation
s = summary(x.pca)
s

## Importance of components:
##                           Comp.1     Comp.2     Comp.3     Comp.4
Comp.5
## Standard deviation     2.6845363 1.70915894 1.44081492 1.34058226
1.22991498
## Proportion of Variance 0.1637894 0.06639146 0.04718063 0.04084456
0.03437934
## Cumulative Proportion  0.1637894 0.23018090 0.27736152 0.31820609
0.35258543
##                            Comp.6     Comp.7     Comp.8     Comp.9
Comp.10
## Standard deviation     1.21553935 1.14944902 1.12353689 1.09041971
1.07693994
## Proportion of Variance 0.03358036 0.03002802 0.02868944 0.02702307
0.02635908
## Cumulative Proportion  0.38616579 0.41619381 0.44488325 0.47190632
0.49826540
##                           Comp.11    Comp.12    Comp.13    Comp.14
Comp.15
## Standard deviation     1.06186858 1.04424330 1.03597128 1.02830200
1.00867106
## Proportion of Variance 0.02562647 0.02478282 0.02439174 0.02403193
0.02312312
## Cumulative Proportion  0.52389187 0.54867469 0.57306643 0.59709837
0.62022149
##                           Comp.16    Comp.17    Comp.18    Comp.19
Comp.20
## Standard deviation     1.00067281 0.98747569 0.95639199 0.94967739
```

```
0.94131701
## Proportion of Variance 0.02275787 0.02216155 0.02078831 0.02049744
0.02013813
## Cumulative Proportion  0.64297935 0.66514090 0.68592921 0.70642665
0.72656478
##                            Comp.21    Comp.22    Comp.23    Comp.24
Comp.25
## Standard deviation     0.91318126 0.90526202 0.88803789 0.87896330
0.86357840
## Proportion of Variance 0.01895227 0.01862498 0.01792298 0.01755856
0.01694926
## Cumulative Proportion  0.74551705 0.76414204 0.78206502 0.79962358
0.81657284
##                            Comp.26    Comp.27    Comp.28    Comp.29
Comp.30
## Standard deviation     0.82206609 0.81010653 0.80649107 0.79095113
0.77454870
## Proportion of Variance 0.01535892 0.01491529 0.01478245 0.01421827
0.01363467
## Cumulative Proportion  0.83193176 0.84684705 0.86162950 0.87584777
0.88948244
##                            Comp.31    Comp.32    Comp.33    Comp.34
Comp.35
## Standard deviation     0.75578348 0.72841985 0.70453815 0.69659209
0.67990572
## Proportion of Variance 0.01298202 0.01205899 0.01128123 0.01102819
0.01050618
## Cumulative Proportion  0.90246446 0.91452345 0.92580467 0.93683287
0.94733904
##                            Comp.36     Comp.37     Comp.38     Comp.39
## Standard deviation     0.638824505 0.608900912 0.575755705 0.558772535
## Proportion of Variance 0.009274926 0.008426371 0.007533969 0.007096062
## Cumulative Proportion  0.956613970 0.965040341 0.972574309 0.979670372
##                            Comp.40     Comp.41     Comp.42     Comp.43
## Standard deviation     0.517888560 0.465399187 0.414416603 0.384617750
## Proportion of Variance 0.006095649 0.004922646 0.003903207 0.003362064
## Cumulative Proportion  0.985766021 0.990688666 0.994591874 0.997953938
##                            Comp.44
## Standard deviation     0.300044565
## Proportion of Variance 0.002046062
## Cumulative Proportion  1.000000000

s$sdev

##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
Comp.8
## 2.6845363 1.7091589 1.4408149 1.3405823 1.2299150 1.2155393 1.1494490
1.1235369
##    Comp.9   Comp.10   Comp.11   Comp.12   Comp.13   Comp.14   Comp.15
Comp.16
```

```
## 1.0904197 1.0769399 1.0618686 1.0442433 1.0359713 1.0283020 1.0086711
1.0006728
##    Comp.17   Comp.18   Comp.19   Comp.20   Comp.21   Comp.22   Comp.23
Comp.24
## 0.9874757 0.9563920 0.9496774 0.9413170 0.9131813 0.9052620 0.8880379
0.8789633
##    Comp.25   Comp.26   Comp.27   Comp.28   Comp.29   Comp.30   Comp.31
Comp.32
## 0.8635784 0.8220661 0.8101065 0.8064911 0.7909511 0.7745487 0.7557835
0.7284199
##    Comp.33   Comp.34   Comp.35   Comp.36   Comp.37   Comp.38   Comp.39
Comp.40
## 0.7045381 0.6965921 0.6799057 0.6388245 0.6089009 0.5757557 0.5587725
0.5178886
##    Comp.41   Comp.42   Comp.43   Comp.44
## 0.4653992 0.4144166 0.3846177 0.3000446
```

```r
# plot the Cumulative  contribution of components
plot(cumsum(s$sdev^2 / sum(s$sdev^2)), type="l", xlab="number of components",
ylab="cumulative varince")

#biplot of the first two components
biplot(x.pca, scale = 0) # no scaling in the plot
```



```
#looking at the rotations/loadings for the 1st and 2nd PCA's
x.pca$loadings[,1]
```

```
##   LotFrontage       LotArea         Alley     LotConfig  Neighborhood
##    0.08719548    0.07976222    0.05689854   -0.01300138    0.06789463
##    Condition1    Condition2      BldgType    HouseStyle   OverallQual
##    0.06088327    0.01109284    0.01151392    0.10285246    0.31664684
##   OverallCond     YearBuilt     RoofStyle      RoofMatl   Exterior1st
##   -0.13346316    0.27392533    0.08256085   -0.02983052    0.03678531
##    MasVnrArea     ExterQual     ExterCond    Foundation      BsmtQual
##    0.19479819   -0.26584454    0.06672505    0.21805446   -0.26393471
##      BsmtCond       Heating     X1stFlrSF     X2ndFlrSF  LowQualFinSF
##    0.04775350    0.01715294    0.22559360    0.11047447   -0.03194826
##      FullBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##    0.25521365    0.06658092   -0.03972530   -0.22790733    0.19646464
##    Functional    Fireplaces    GarageType    GarageArea    GarageCond
##    0.04472175    0.16867012   -0.19686057    0.26702993    0.12688749
##    PavedDrive         Fence   MiscFeature       MiscVal        MoSold
##    0.12679095    0.08369909    0.03489171   -0.01660648    0.01938814
##        YrSold      SaleType SaleCondition     SalePrice
##   -0.01073624   -0.04251002    0.10559363    0.32865608
```

```
x.pca$loadings[,2]
```

```
##   LotFrontage       LotArea         Alley     LotConfig  Neighborhood
##  0.1448494539  0.1716388928 -0.1429111104 -0.0920078730  0.0676979140
##    Condition1    Condition2      BldgType    HouseStyle   OverallQual
## -0.0808579775 -0.0709089550 -0.2682419729  0.0161113164  0.0112612329
##   OverallCond     YearBuilt     RoofStyle      RoofMatl   Exterior1st
##  0.1727298427 -0.2770347455  0.0539896899 -0.0735124091  0.0008162604
##    MasVnrArea     ExterQual     ExterCond    Foundation      BsmtQual
##  0.0433034611  0.0389093762 -0.1288442312 -0.1811866385  0.1150455340
##      BsmtCond       Heating     X1stFlrSF     X2ndFlrSF  LowQualFinSF
## -0.1186745622  0.0884626130  0.0985754620  0.3140529983  0.1520903823
##      FullBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##  0.1312772285  0.4042153850  0.1566373494 -0.0056532581  0.4084734300
##    Functional    Fireplaces    GarageType    GarageArea    GarageCond
## -0.0828635449  0.1338667737  0.1350606029  0.0043023336 -0.1370273757
##    PavedDrive         Fence   MiscFeature       MiscVal        MoSold
## -0.1817136130 -0.0935721405 -0.0587515796  0.0553283878  0.0422438834
##        YrSold      SaleType SaleCondition     SalePrice
## -0.0247431289  0.0236185427 -0.0559600163  0.0979615136
```

```
#now we can plot the data across the 1st 2 PC's and colour by OverCond and
then neighbourhood
proj = as.data.frame(x.pca$scores)
proj$OverallCond = house.data$OverallCond
proj$Neighborhood = house.data$Neighborhood

#the OverallCond Plot
dev.off()
```

```
## null device
##           1
```

```
ggplot(proj) +
  geom_point(aes(x=proj[,1], y=proj[,2], color=OverallCond)) +
  labs(color = "Overall Condition", x = "PC1", y = "PC2", title = "PCA Plot:
By Overall Condition")
```

```
#The Neighbourhood Plot
ggplot(proj) +
  geom_point(aes(x=proj[,1], y=proj[,2], color=Neighborhood)) +
  labs(color = "Neighborhood", x = "PC1", y = "PC2", title = "PCA Plot: By
Neighborhood")
```

```
#We will use the PC's since it explains our data in better dimessions
#now we can have a look at these groups though a k-means clustering
algorithm, k = the number of levels in OverallCond
#to make results reproducible, we set the random seed so the center points
picked are always the same
set.seed(1)
km.OverallCond = kmeans(x.pca$scores, centers=
length(unique(house.data$OverallCond)) ,1000)
km.OverallCond$cluster
```

```
##    [1] 1 1 1 3 2 1 2 1 3 3 3 2 3 2 3 3 3 1 1 3 2 3 2 1 3 2 3 2 3 3 3 3 1 3
2 2 1
##   [38] 3 3 3 3 3 3 3 3 2 1 1 3 3 1 3 1 2 3 1 1 1 2 3 1 3 1 3 1 2 1 1 3 1 2
3 1 3
##   [75] 3 3 3 3 3 3 1 1 2 3 1 2 1 1 3 3 3 3 3 3 1 1 1 3 3 3 1 1 1 1 3 2 3 3
3 1 3
##  [112] 1 2 1 3 1 3 1 2 1 1 3 3 1 1 3 1 3 1 3 1 1 3 1 3 1 3 1 1 1 3 1 3 1 3
1 3 1
##  [149] 3 3 3 2 1 3 3 3 3 2 1 2 1 2 1 3 3 3 3 2 1 2 3 3 1 3 1 1 1 1 2 3 1 3
3 1 3
##  [186] 2 1 3 3 1 2 3 1 1 3 1 2 3 3 2 1 3 3 1 3 1 3 3 1 3 3 1 1 1 3 3 1 3 1
1 1 1
##  [223] 1 3 2 3 1 3 3 1 3 2 3 3 1 3 1 1 2 3 1 3 3 1 1 1 3 3 1 2 3 1 1 3 3 1
1 1 1
##  [260] 3 3 2 3 3 3 1 1 3 3 3 2 1 2 3 3 3 1 3 2 1 1 1 1 2 1 1 3 3 3 3 1 3 3
1 3 3
##  [297] 3 2 1 3 3 1 1 3 1 2 1 3 3 2 1 3 3 2 3 1 2 1 2 1 2 2 1 3 2 3 1 3 3 3
3 3 2
##  [334] 1 1 1 2 1 1 3 1 3 3 2 3 3 3 1 1 2 2 1 3 3 3 1 1 1 3 2 1 3 1 3 1 3 1
1 3 3
##  [371] 1 3 3 3 1 3 1 2 2 1 3 1 1 3 1 1 3 3 1 2 3 1 3 3 3 3 3 1 3 1 1 1 3 2
1 3 3
##  [408] 3 2 2 3 3 2 3 1 1 3 3 3 3 1 1 3 2 3 3 1 3 1 1 3 3 3 1 3 1 3 3 3 3 2
3 3 1
##  [445] 1 3 1 1 3 3 3 1 1 1 3 3 3 1 3 3 1 3 3 3 3 1 3 3 2 1 1 1 1 2 1 3 1 2
2 3 2
```

```
##  [482] 2 3 1 3 3 3 1 3 3 1 3 1 3 3 3 2 3 3 3 3 1 3 1 1 3 1 1 3 3 3 1 3 3 3
2 1 2
##  [519] 1 3 3 3 3 2 2 1 3 2 3 1 1 3 3 3 1 3 1 3 3 1 2 1 1 1 1 1 3 3 3 1 1 3
2 3 2
##  [556] 3 3 3 1 1 3 3 3 3 1 3 2 1 1 1 3 3 1 1 3 3 3 3 1 3 1 2 3 2 3 2 3 3 1
3 1 2
##  [593] 3 1 3 2 3 1 1 1 2 3 1 1 1 1 3 1 1 3 2 1 1 3 3 3 1 3 2 2 3 1 3 1 1 3
3 3 1
##  [630] 3 3 1 3 3 3 3 3 3 3 2 2 1 2 1 2 3 3 3 1 3 1 3 1 3 2 3 3 3 3 3 3 1 2 3
3 2 1
##  [667] 1 1 3 3 1 3 1 1 3 1 3 3 2 3 1 3 1 2 1 1 1 1 2 1 1 2 2 3 3 1 3 3 3 1
2 3 2
##  [704] 3 1 3 2 1 1 3 3 3 1 3 1 1 3 3 2 3 1 1 3 3 2 3 1 1 3 3 1 1 1 3 3 3 3
1 3 1
##  [741] 3 3 1 1 1 2 1 3 2 3 3 1 1 2 3 1 1 3 1 2 3 3 1 2 1 2 1 3 1 2 3 3 3 3
2 1 1
##  [778] 3 1 1 1 1 1 1 1 3 1 3 1 3 1 1 1 1 1 1 1 3 3 2 3 1 3 1 2 3 2 1 3 3 3 3
1 3 3
##  [815] 3 1 3 2 3 1 1 3 1 3 1 2 3 1 1 1 3 1 1 3 3 3 3 3 1 3 3 3 3 3 3 1 1 3
3 1 1
##  [852] 1 3 3 3 3 3 3 1 1 1 3 3 1 3 1 3 2 3 1 1 3 1 3 3 3 2 3 2 3 3 1 1 1 3 3
1 3 3
##  [889] 2 3 3 1 3 3 1 3 3 1 2 3 3 3 1 2 3 3 2 3 3 1 3 3 3 1 1 3 3 3 1 1 1 3
1 1 1
##  [926] 1 2 1 2 1 1 3 2 1 2 3 1 1 1 1 1 1 3 3 3 3 3 2 1 3 3 3 3 1 3 1 1 3 1
1 3 1
##  [963] 1 2 1 1 3 3 3 3 3 1 1 1 3 1 3 1 3 3 1 2 1 1 1 3 3 2 1 1 2 3 1 1 2 3
3 1 3
## [1000] 1 3 3 1 3 1 1 1 3 2 3 3 3 3 3 3 1 1 1 1 1 1 1 3 1 2 3 3 2 3 3 3 3 1
1 3 3
## [1037] 2 1 3 3 3 3 1 2 2 1 2 3 3 3 1 1 3 3 1 1 1 1 2 3 1 3 3 3 3 1 1 3 1 3
3 1 3
## [1074] 3 1 1 3 3 1 1 3 3 1 3 1 1 3 2 1 1 3 1 3 3 3 1 3 1 3 1 3 3 3 3 3 3 2 1
2 1 2
## [1111] 1 1 3 3 3 2 1 3 3 3 3 1 3 3 1 3 1 2 1 3 3 3 3 1 1 3 3 3 1 3 3 1 2 3
3 3 1
## [1148] 3 3 3 3 3 3 3 3 1 1 1 1 2 1 1 3 3 3 1 1 1 1 1 2 3 3 1 3 3 2 3 3 3 3 1
2 2 3
## [1185] 1 3 3 1 1 1 1 1 3 1 3 1 1 3 1 3 3 1 3 1 3 2 3 1 3 2 1 1 3 3 3 3 1 1
3 3 3
## [1222] 3 3 3 1 3 1 3 2 3 3 3 1 3 3 3 1 1 1 2 1 2 1 2 1 1 1 1 3 3 2 1 3 1 1
3 2 3
## [1259] 1 3 1 3 3 3 1 1 3 2 2 3 1 1 3 3 3 3 3 3 1 1 3 1 1 3 3 3 3 3 3 3 1 2 3 3
1 1 3
## [1296] 3 3 1 2 3 1 3 2 1 1 2 1 3 3 1 2 1 2 2 3 1 2 1 2 3 3 3 1 3 2 3 3 3 3
1 2 3
## [1333] 3 3 3 1 3 3 1 3 3 1 1 3 1 3 1 2 1 3 1 1 3 2 1 1 3 3 1 2 3 2 3 1 1 1
1 1 1
## [1370] 2 3 3 1 2 1 2 3 3 3 1 3 1 3 3 3 3 1 3 2 3 1 3 3 3 1 2 3 3 3 3 3 1 1
1 3 1
```

```
## [1407] 3 3 3 1 1 3 3 2 3 1 3 2 3 1 1 1 1 1 3 3 1 3 3 1 1 1 3 1 3 3 3 2 3 1
3 1 2
## [1444] 3 1 3 3 1 3 3 3 2 1 3 1 1 1 1 3 3

#we can table the results with the OverallCond variable
table(km.OverallCond$cluster,as.matrix(house.data$OverallCond))

##
##     Average Good Poor
##   1     513   48    2
##   2     176   14    1
##   3     441  237   28

#observations per cluster
margin.table(table(km.OverallCond$cluster,as.matrix(house.data$OverallCond)),
margin=1)

##
##   1   2   3
## 563 191 706

#we can plot the boxplots of the first 3 clusters across the SalePrice and
OverallCond (SalePrice is scaled)
par(mfrow = c(3,1))
plot(house.data[km.OverallCond$cluster==1,c(11,44)],pch="x", ylim=c(-3,3))
plot(house.data[km.OverallCond$cluster==2,c(11,44)],col="firebrick", ylim=c(-
3,3))
plot(house.data[km.OverallCond$cluster==3,c(11,44)],col="skyblue", ylim=c(-
3,3))
par(mfrow = c(1,1))

#now we can have a look at these groups though a k-means clustering
algorithm, k = the number of levels in Neighbourhood
set.seed(1)
km.Neighborhood = kmeans(x.pca$scores,
centers=length(unique(house.data$Neighborhood)) ,1000)
km.Neighborhood$cluster

##    [1] 14  3 14 13 16  8 23  8 15 20  6 16  6 23 13 13  8  8  3  6 16 10
23 22
##   [25] 17 23 13 23  3 20 12 13 23  6  2 16 13  6 17 15  3 17  7 13  3  2
23 23
##   [49] 15  3 14 18 13  9  3  3 19 14 16 24 23 12  5 20 14 16 23 23 17  4
23 13
##   [73] 14  3 15 22 24 24 15 10  4  5 23 24  8 16 14 19 18 24 24  6 10 15
14  8
##   [97] 23  6  8  8  3 11 15 23  4 16 10 24 12  6 24 14 16 17  4 19  3 13
16 14
##  [121]  9 24 17  5 17 12 22 12  3  3  4 14 13 23  3  6 13 15 14 14 24 23
13 23
##  [145] 15 22 24 14 24 24 13  2  4  3  7  7 24 11 11 16 13 16 23 24 17 15
```

```
6 16
##  [169] 14  9 18  6  5  3 23  4 14 17  2 12  5  4  6 12 20 18 13 12 15  5
4  4
##  [193] 23 22  6 22  2 18 18  2 13  3 17  5 10 23 17  3 11  7 24 23 11 13
8  3
##  [217] 23 10 17  5 23 14 14 24  2 22 16 22 13  5  6 16 22 13 14 22 23 11
2 24
##  [241] 23 17 24 22 11 23 15  6 14  1  8  5 11  6 17 14 11 23 14 24 13 16
13 18
##  [265] 10 17 14 18  7  7 11  9 16  6 13 17 23 24  2  4 11 19  5 23  5 19
3 24
##  [289] 17 24 14 10  3  4  6 13  3 16  6 17  6 14 23 13  4 23 11 10 24  2
14 17
##  [313] 24  1 10 14  4 11 11  4 16 16  4 17  9 24  5  6  4 12 15  3 23  5
14  1
##  [337]  2 23  8  6 11 12 15  2 22 24 21  3  5  2  2  3  7 17  3 17 23 22
6 16
##  [361] 13 12 14 22 14  7  3  9  3 17 14 12 22  3 14  7 13 11  2 14 24 19
14 12
##  [385] 14  5 12  6 23  2 10 14  8 24 24  3  6  4 12 11  5 23 12 16 14 13
18  4
##  [409] 16 16 17  6 23 10 11 23  6  4 24  3 15  4  6 16  3 10 23  6 23 23
22 12
##  [433] 22 14 22 14 24 24 24  8  2 15 24  5 14  6 23 14 12 17 12  9 11 11
15  6
##  [457] 10  9 10 24 11 17 24  6 24  5  9  3 23 11  5  9 22  2  5 13 23  2
23  6
##  [481]  2  2  4  5 17  3  6  3 15 22 22  3 14 24 24 12  2 12  6  8 22 11
8  3
##  [505] 22 15 11 23 12 13  8  5 13 17  7  2 13 16 14  4 10  3 17 20 16 19
6 16
##  [529] 12 15 11 20 24 12 14 12 14  6  6  8  2 14 23 22 14 11 24 17 20 11
22  6
##  [553] 23 12 11 24  3  3 14  5  6  3 24 12 11 13 16 23 23 13 15  3 11 14
13 24
##  [577] 24  3 19 10  3  2 22 20 24  2 12  6  9 20 14  2 17  5  6 23 10  5
23 22
##  [601] 16 17 11 19 23  4 17 17  4  3 16  8 14 24 22 17 14 13  2 16 10  4
24  5
##  [625]  4  6  8  3  4  9 10  5 17 13  8 18  7 15 12  5  2 11  4  4  2 13
6  6
##  [649]  4 22 11  7 14 10  2 22  3  3  3 17  4 16  6  3  2 14  9 23 13  7
14 17
##  [673]  3  6  3 22 10 12  2  6  5 24  3 23 11  5 11  5 23  5  5 16 16 24
13  3
##  [697] 24 24 17 19  2  6 16 15 23  8  1  5 14  3 12 12  5 24  4  6 10  3
16  3
##  [721] 23  5 17 24  2  8 17 23 15 18  5 11 14  8 13 12 15 14 24 14  7 17
3  3
##  [745]  5 16 14 12 23 12 10 14 23 16 24 19 14 13 19 16  7 24 11 16  5 23
```

```
8  8
##  [769] 23 16 13 24  3  3  2  5 23  6 15  3 23 11 23 17 10  3  8 11 24  4
5 13
##  [793] 11 23  8 11  3  3 16 13  8 24 14 16  3 23  3  3  3 15  6  5  8  8
10 23
##  [817] 13 23  3  5 14 19 14 24 23 20 24 23  9 19 13 19 14  3  3  3  3 22
3 24
##  [841]  4 10  7 15 10  4 11 13  4 17  5  5  4 17  6 17  7 14  3  4 13  6
13  6
##  [865] 23 13 23  8 14 11 24 14 13  8 24 16  3 16  8 17  3 17 14 18  3  5
15 17
##  [889]  9  6  8  6  6  3 15  6 10 15  2  3 24 17 14 23  3  3 23  4  3 14
15 24
##  [913]  8 15 19 22 24  3 11  6 11 15 23  5 17 13 16  4 23 14 23 24  2 23
9 17
##  [937] 23 11 23  4 15 14 15 15 13 24 17 23 14  6 24 13 17  8  9 15 22  6
23  5
##  [961] 17  4 22 23 11 14 17  6  7  6 24  5  5 23 20  5 24 19 24 13 13 16
23 14
##  [985] 15 24 17  2  4 11 16 10  4 14  2 24  6  6  7 23  9 12 23 20  5  3
4 22
## [1009] 23 18  3 15 17  6 24 11 23  5 14  5  3 23  7  5  9 13  3 23  3 22
15 18
## [1033]  4 23 17 12  2 14 22 22  3  6  5 11  9  6 16 24  6 24 23 23 13  6
14  3
## [1057]  5 11 16  7  5  8 15 17 13 14 14  3  4 24  6  4 12 13 23  3  8 13
5  3
## [1081] 17 13 23  8 14  3 22 11 22 19 15 19  3 13  3 23 24  5 24 23 12 24
3  6
## [1105] 22 16 23 14 14 23 14  4 24  3 17  2 14  3 17  3 24 23 13 17 14  7
5 23
## [1129] 14 10  4 24 12 11 14 24  3 12 23 24 13  4 16 24  7 24 23  4 24 17
3  3
## [1153]  3 17  4 17 17  5 23  4 22  6  3  3  3 23 23 14  4 16  3  8 19 18
4 16
## [1177] 13 10 24 12 11  2 16 10  6 17 20 23 14 14 17 19 10  5 13 14 14 24
23  3
## [1201]  8 14 12 23  3 23 24 23 17  2  8 23 12 17  3 24 15 23 12 12  3  6
3  6
## [1225] 14 17 11 13  2  6 21  6 15  6 12 12  5 14  3 23 14 23  3  2  4  4
11 13
## [1249]  7  3 23  5  8  4 14 24  2 24 14  3 14  6  4 12  5  5 15  2  4  3
9  6
## [1273]  6 13 24  9 13 13 14 17 23  7 24 15  4 24  6 17  5 16 13 22 15 11
12  6
## [1297] 17  5  2  6 14  4 16 23 22  2  5 17  3  3 23 23 16 16 24  4 23 19
23  6
## [1321]  6  7 11 24 23 12 24 17  8 14 23 24  7 24 22  3 15  7 14 17 24 23
14  7
## [1345] 14 24  4 23 23  7 15  4 12 16 14  4 13 13  5  2  4 23 17 14 19 11
```

```
14 22
## [1369]  5 23 24  6 14  2 11 23 24 10 22 11 10  4 12 12 24 12  8 10  2 17
23 15
## [1393]  3 15  5 16 12 12  3 24 24 14 23 23 10  5 13 13 24  4 14 17 15 23
12  5
## [1417] 15 16 24 17  4 22  5  9  6  6 14 10 17 17 14 22 10 14  3 17 13  2
3  6
## [1441] 18  5 16  7 23 24 13 14 24 22 15 23 22 24 19 14  3  8  6  3

#we can table the results with the OverallCond variable
table(km.Neighborhood$cluster,as.matrix(house.data$Neighborhood))

##
##      Blmngtn BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert IDOTRR
##   1        0      0       0       2       0       0       0       0      0
##   2        0      0       0       0       0       1       2       1      0
##   3        0      0       1       4       4       4       9       0      2
##   4        0      0       1       6       0      14       2       0      1
##   5       16      0       0       0      10       2       1       0      0
##   6        0      0       0       2       3       4       8       1      0
##   7        0      0       3       0       1       4       7       0      3
##   8        0      0       2       1       2       1       3       3      3
##   9        0      0       0       7       0       0       3       0      0
##   10       0      0       0       0       0       1       5       0      4
##   11       0      0       0       0       0       0       0       0      0
##   12       0      1      11       1       0       1      12       0     11
##   13       0      0       2       0      10       2       5       0      1
##   14       0      0       0       3      47       0       2      63      0
##   15       0      0       0       0       0       3       6       0      0
##   16       0      0       0       0       4       0       0       1      0
##   17       0      0       5       1       6       8       7       0      3
##   18       0      0       1       1       0       1       0       0      2
##   19       0      0       0       0       0       0       0       0      0
##   20       0      0       6       0       0       0       1       0      0
##   21       0      0       0       0       0       0       0       0      0
##   22       0     15       0       0       0       0       7       0      0
##   23       1      0       0       0      56       2       1       9      0
##   24       0      0      26       0       7       3      19       1      7
##
##      MeadowV Mitchel NAmes NoRidge NridgHt NWAmes OldTown Other Sawyer
SawyerW
##   1        0       0     0       0       0      0       0     0      0
0
##   2        0       0     0       4      32      0       0     0      0
0
##   3        0       7    56       0       0     10       3     0     18
5
##   4        1       0    14       1       0     22       2     0      2
1
##   5        0       4     1       0      16      0       0     0      0
```

```
5
##   6        0       2    40       0       0     13       3     0     16
0
##   7        0       0     4       0       0      0       7     0      0
1
##   8        0       4    13       0       0      5       4     0      4
1
##   9        0       0     8       0       0      2       0     0      0
0
##   10       0       0     1       0       0      0      28     0      0
0
##   11       0       0     0       6       1      3       0     0      0
20
##   12       0       0     0       0       0      0      13     0      3
2
##   13       0      10    28       0       0      4       0     0     12
2
##   14       0       1     0       2       0      2       0     0      0
0
##   15       0       7    14       0       0      1      15     0      3
5
##   16       0       0     0      26      18      1       0     0      0
1
##   17       0       6    17       0       0      6      12     0     10
0
##   18       0       0     1       0       0      0       4     0      0
0
##   19       0       0     0       0       0      0       1     0      0
0
##   20       0       0     0       0       1      1       4     0      0
0
##   21       0       0     1       0       0      0       0     0      1
0
##   22      16       0     1       0       0      0       0    11      0
2
##   23       0       5     4       2       9      3       0     0      0
12
##   24       0       3    22       0       0      0      17     0      5
2
##
##      Somerst StoneBr SWISU Timber Veenker
##   1        0       0     0      2       0
##   2        5       4     0      5       0
##   3        0       0     2      1       3
##   4        0       0     4      0       1
##   5        8      12     0      0       3
##   6        0       0     0      1       0
##   7        0       0     0      1       0
##   8        0       0     0      0       0
##   9        0       0     0      2       1
```

```
##   10       0       0     1      0       0
##   11      22       0     0      9       0
##   12       0       0     2      1       0
##   13       0       0     2      1       1
##   14       0       0     0      0       0
##   15       0       0     1      0       0
##   16       5       5     0      3       0
##   17       0       0     1      2       2
##   18       0       0     6      0       0
##   19      23       0     0      0       0
##   20       0       0     0      0       0
##   21       0       0     0      0       0
##   22       0       0     0      0       0
##   23      23       4     0     10       0
##   24       0       0     6      0       0
```

```
#observations per cluster
margin.table(table(km.Neighborhood$cluster,as.matrix(house.data$Neighborhood)
), margin=1)
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
19  20
##   4  54 129  72  78  93  31  46  23  40  61  58  80 120  55  64  86  16
24  13
##  21  22  23  24
##   2  52 141 118
```

```
#we can plot the boxplots of the first 3 clusters across the SalePrice and
Neighborhood (SalePrice is scaled)
plot(house.data[km.Neighborhood$cluster==1,c(5,44)], ylim=c(-3,3), xaxt =
"n", xlab = "")
par(new = T)
plot(house.data[km.Neighborhood$cluster==2,c(5,44)],col="firebrick", ylim=c(-
3,3), xaxt = "n", xlab = "")
par(new = T)
plot(house.data[km.Neighborhood$cluster==3,c(5,44)],col="skyblue", ylim=c(-
3,3), xaxt = "n", xlab = "")
par(new = T)
plot(house.data[km.Neighborhood$cluster==4,c(5,44)],col="khaki", ylim=c(-
3,3), xaxt = "n", xlab = "")
par(new = T)
plot(house.data[km.Neighborhood$cluster==5,c(5,44)],col="green", ylim=c(-
3,3), xaxt = "n", xlab = "")
par(new = T)
plot(house.data[km.Neighborhood$cluster==6,c(5,44)],col="magenta", ylim=c(-
3,3), xaxt = "n", xlab = "")
par(new = T)
plot(house.data[km.Neighborhood$cluster==7,c(5,44)],col="coral4", ylim=c(-
3,3), xaxt = "n", xlab = "")
```

```
par(new = T)
plot(house.data[km.Neighborhood$cluster==8,c(5,44)],col="purple", ylim=c(-
3,3), xaxt = "n", xlab = "")
par(new = T)
plot(house.data[km.Neighborhood$cluster==9,c(5,44)],col="salmon", ylim=c(-
3,3), xaxt = "n", xlab = "")
par(new = T)
plot(house.data[km.Neighborhood$cluster==10,c(5,44)],col="pink", ylim=c(-
3,3), xaxt = "n", xlab = "")

#adding in better x-axis
text(x = 1:length(unique(house.data$Neighborhood)),
     y = par("usr")[3] - 0.1,
     labels = unique(house.data$Neighborhood),
     xpd = NA,
     adj = 1,
     srt = 55
)

#adding in rect to highlight a how the SalePrice varies per a neighbourhood
per cluster
rect(xleft = 7.5, xright = 8.5, ybottom = -2, ytop = 2, border = "red", lwd =
3)

#as a final experiment, lets look at how the OverallCond K-Means cluster
model looks over neighborhood
#(basically K-Means with 3 centers)
plot(house.data[km.OverallCond$cluster==1,c(5,44)], ylim=c(-3,3), xaxt = "n",
xlab = "")
par(new = T)
plot(house.data[km.OverallCond$cluster==2,c(5,44)],col="firebrick", ylim=c(-
3,3), xaxt = "n", xlab = "")
par(new = T)
plot(house.data[km.OverallCond$cluster==3,c(5,44)],col="skyblue", ylim=c(-
3,3), xaxt = "n", xlab = "")

#adding in better x-axis
text(x = 1:length(unique(house.data$Neighborhood)),
     y = par("usr")[3] - 0.1,
     labels = unique(house.data$Neighborhood),
     xpd = NA,
     adj = 1,
     srt = 55
)

#adding in rect to highlight clusters
rect(xleft = 0, xright = 25, ybottom = -0.5, ytop = 1.2, border = "black")
rect(xleft = 4, xright = 24, ybottom = 0, ytop = 3, border = "red")
rect(xleft = 2, xright = 24, ybottom = -1.5, ytop = 0.2, border = "blue")
```

```r
#the above shows us that we seem to have 3 recognizable clusters for
Neighborhoods, nice!

#now we can have a look at hierarchical clustering, first the libraries


#fitting the hierarchical cluster model
set.seed(1)
hclust.house = hclust(dist(x.pca$scores), method="complete") # aggomorative
hierarchical clustering based on complete linkage

#ploting with the OverallCond labels as the labels
plot(hclust.house,labels=(as.character(house.data$OverallCond)),
main="",xlab="complete-linkage",ylab="level")

#ploting squares around the desired clusters (we want the amount of
Conditions as the amount of clusters)
rect.hclust(hclust.house,k=length(unique(house.data$OverallCond)), border =
"red")

#we can get a table of the results as well
qualClus = cutree(hclust.house, length(unique(house.data$OverallCond)))
table(qualClus, house.data$OverallCond)

##
## qualClus Average Good Poor
##        1    1113  297   31
##        2      16    1    0
##        3       1    1    0

#observations per cluster
margin.table(table(qualClus, house.data$OverallCond), margin=1)

## qualClus
##    1    2    3
## 1441   17    2

#ploting with the neighborhood as the labels
plot(hclust.house,labels=(as.character(house.data$Neighborhood)),
main="",xlab="complete-linkage",ylab="level")

#ploting squares around the desired clusters (we want the amount of neighbors
as the amount of clusters)
rect.hclust(hclust.house,k=length(unique(house.data$Neighborhood)), border =
"red")

#we can get a table of the results aswell
neighClus = cutree(hclust.house, length(unique(house.data$Neighborhood)))
table(neighClus, house.data$Neighborhood)
```

```
##
## neighClus Blmngtn Other BrDale BrkSide ClearCr CollgCr Crawfor Edwards
Gilbert
##         1       17     0      0       1       4     118        6        8
71
##         2        0    11     14      27      11      27       28       51
6
##         3        0     0      0       6       0       0        0        0
0
##         4        0     0      0       0       0       1        0        0
0
##         5        0     0      0       0       0       0        2        5
0
##         6        0     0      0       0       0       0        1        1
0
##         7        0     0      2      18       2       4        8       23
1
##         8        0     0      0       1       1       0        1        0
0
##         9        0     0      0       0       7       0        0        2
0
##        10        0     0      0       2       1       0        1        2
0
##        11        0     0      0       0       0       0        0        0
0
##        12        0     0      0       0       0       0        1        0
1
##        13        0     0      0       2       0       0        1        3
0
##        14        0     0      0       0       0       0        0        0
0
##        15        0     0      0       0       0       0        2        0
0
##        16        0     0      0       0       0       0        0        0
0
##        17        0     0      0       0       2       0        0        0
0
##        18        0     0      0       0       0       0        0        0
0
##        19        0     0      0       1       0       0        0        2
0
##        20        0     0      0       0       0       0        0        1
0
##        21        0     0      0       0       0       0        0        1
0
##        22        0     0      0       0       0       0        0        0
0
##        23        0     0      0       0       0       0        0        1
0
##        24        0     0      0       0       0       0        0        0
```

```
0
##
## neighClus IDOTRR MeadowV Mitchel NAmes NoRidge NridgHt NWAmes OldTown
Sawyer
##         1       1       0       8     4      36      72     18       1
0
##         2      14      17      30   184       2       0     48      42
64
##         3       0       0       0     0       0       0      1       3
0
##         4       0       0       0     0       3       4      0       0
0
##         5       0       0       7    11       0       0      1      11
5
##         6       2       0       0     1       0       0      0      19
0
##         7      15       0       4     9       0       0      2      19
4
##         8       1       0       0     2       0       0      0       2
0
##         9       0       0       0     8       0       0      2       0
0
##        10       1       0       0     1       0       0      0       8
0
##        11       1       0       0     0       0       0      0       0
0
##        12       1       0       0     0       0       0      1       0
0
##        13       0       0       0     3       0       0      0       3
0
##        14       0       0       0     0       0       0      0       1
0
##        15       0       0       0     1       0       0      0       0
0
##        16       0       0       0     0       0       0      0       1
0
##        17       0       0       0     0       0       0      0       0
0
##        18       0       0       0     1       0       0      0       0
0
##        19       1       0       0     0       0       0      0       1
0
##        20       0       0       0     0       0       1      0       1
0
##        21       0       0       0     0       0       0      0       0
0
##        22       0       0       0     0       0       0      0       0
1
##        23       0       0       0     0       0       0      0       0
0
```

```
##        24      0       0       0     0       0       0      0       1
0
##
## neighClus SawyerW Somerst StoneBr SWISU Timber Veenker
##         1       34      85      19     0     25        5
##         2       17       1       2    11      6        5
##         3        0       0       0     0      0        0
##         4        0       0       4     0      0        0
##         5        5       0       0     1      0        0
##         6        0       0       0     1      0        0
##         7        2       0       0     6      3        0
##         8        0       0       0     2      0        0
##         9        0       0       0     0      2        1
##         10       0       0       0     0      0        0
##         11       0       0       0     0      0        0
##         12       0       0       0     0      0        0
##         13       0       0       0     0      0        0
##         14       0       0       0     0      0        0
##         15       0       0       0     0      0        0
##         16       0       0       0     4      0        0
##         17       0       0       0     0      2        0
##         18       0       0       0     0      0        0
##         19       1       0       0     0      0        0
##         20       0       0       0     0      0        0
##         21       0       0       0     0      0        0
##         22       0       0       0     0      0        0
##         23       0       0       0     0      0        0
##         24       0       0       0     0      0        0
```

```
#observations per cluster
margin.table(table(neighClus, house.data$Neighborhood), margin=1)
```

```
## neighClus
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
19  20
## 533 618  10  12  48  25 122  10  22  16   1   4  12   1   3   5   4   1
6   3
##  21  22  23  24
##   1   1   1   1
```

```
#as a final experiment, let's loop though 3 - 24 to see what best clusters we
get for neighborhood
####UNCOMMENT THIS SETTING TO SEE ALL THE GRAPHS####
#par(ask = T)
for(i in 3:24){

  #plot graph
  plot(hclust.house,labels=(as.character(house.data$Neighborhood)),
main="",xlab="complete-linkage",ylab="level")
```

```
  #plot the labelled clusters
  rect.hclust(hclust.house,k=i, border = "red")

  #cut the tree at this cluster amount and table the results
  neighClus = cutree(hclust.house, i)
  table(neighClus, house.data$OverallCond)

  #observations per cluster
  margin.table(table(neighClus, house.data$Neighborhood), margin=1)

}
```