Project2.R

Lenovo

2022-10-21

#Project1  
#Shimpli Borkar  
#house price prediction  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(ISLR2)  
library(stargazer)

##   
## Please cite as:   
##   
## Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.  
## R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(leaps)  
library(Amelia)

## Loading required package: Rcpp  
## ##   
## ## Amelia II: Multiple Imputation  
## ## (Version 1.8.0, built: 2021-05-26)  
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell  
## ## Refer to http://gking.harvard.edu/amelia/ for more information  
## ##

library(ggplot2)  
library(lmtest)

## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(Hmisc)

## Loading required package: survival  
##   
## Attaching package: 'survival'  
##   
## The following object is masked from 'package:caret':  
##   
## cluster  
##   
## Loading required package: Formula  
##   
## Attaching package: 'Hmisc'  
##   
## The following objects are masked from 'package:dplyr':  
##   
## src, summarize  
##   
## The following objects are masked from 'package:base':  
##   
## format.pval, units

library(tseries)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

library(caTools)  
library(broom)  
  
hodata<-read\_csv("C:\\Users\\Lenovo\\Downloads\\house.csv")

## Rows: 506 Columns: 13  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (13): crim, zn, indus, chas, nox, rm, age, dis, rad, tax, ptratio, lstat...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

summary(hodata)

## crim zn indus chas   
## Min. : 0.00632 Min. : 0.00 Min. : 0.46 Min. :0.00000   
## 1st Qu.: 0.08205 1st Qu.: 0.00 1st Qu.: 5.19 1st Qu.:0.00000   
## Median : 0.25651 Median : 0.00 Median : 9.69 Median :0.00000   
## Mean : 3.61352 Mean : 11.36 Mean :11.14 Mean :0.06917   
## 3rd Qu.: 3.67708 3rd Qu.: 12.50 3rd Qu.:18.10 3rd Qu.:0.00000   
## Max. :88.97620 Max. :100.00 Max. :27.74 Max. :1.00000   
## nox rm age dis   
## Min. :0.3850 Min. :3.561 Min. : 2.90 Min. : 1.130   
## 1st Qu.:0.4490 1st Qu.:5.886 1st Qu.: 45.02 1st Qu.: 2.100   
## Median :0.5380 Median :6.208 Median : 77.50 Median : 3.207   
## Mean :0.5547 Mean :6.285 Mean : 68.57 Mean : 3.795   
## 3rd Qu.:0.6240 3rd Qu.:6.623 3rd Qu.: 94.08 3rd Qu.: 5.188   
## Max. :0.8710 Max. :8.780 Max. :100.00 Max. :12.127   
## rad tax ptratio lstat   
## Min. : 1.000 Min. :187.0 Min. :12.60 Min. : 1.73   
## 1st Qu.: 4.000 1st Qu.:279.0 1st Qu.:17.40 1st Qu.: 6.95   
## Median : 5.000 Median :330.0 Median :19.05 Median :11.36   
## Mean : 9.549 Mean :408.2 Mean :18.46 Mean :12.65   
## 3rd Qu.:24.000 3rd Qu.:666.0 3rd Qu.:20.20 3rd Qu.:16.95   
## Max. :24.000 Max. :711.0 Max. :22.00 Max. :37.97   
## medv   
## Min. : 5.00   
## 1st Qu.:17.02   
## Median :21.20   
## Mean :22.53   
## 3rd Qu.:25.00   
## Max. :50.00

view(hodata)  
head(hodata)

## # A tibble: 6 × 13  
## crim zn indus chas nox rm age dis rad tax ptratio lstat  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.00632 18 2.31 0 0.538 6.58 65.2 4.09 1 296 15.3 4.98  
## 2 0.0273 0 7.07 0 0.469 6.42 78.9 4.97 2 242 17.8 9.14  
## 3 0.0273 0 7.07 0 0.469 7.18 61.1 4.97 2 242 17.8 4.03  
## 4 0.0324 0 2.18 0 0.458 7.00 45.8 6.06 3 222 18.7 2.94  
## 5 0.0690 0 2.18 0 0.458 7.15 54.2 6.06 3 222 18.7 5.33  
## 6 0.0298 0 2.18 0 0.458 6.43 58.7 6.06 3 222 18.7 5.21  
## # … with 1 more variable: medv <dbl>

summary(hodata)

## crim zn indus chas   
## Min. : 0.00632 Min. : 0.00 Min. : 0.46 Min. :0.00000   
## 1st Qu.: 0.08205 1st Qu.: 0.00 1st Qu.: 5.19 1st Qu.:0.00000   
## Median : 0.25651 Median : 0.00 Median : 9.69 Median :0.00000   
## Mean : 3.61352 Mean : 11.36 Mean :11.14 Mean :0.06917   
## 3rd Qu.: 3.67708 3rd Qu.: 12.50 3rd Qu.:18.10 3rd Qu.:0.00000   
## Max. :88.97620 Max. :100.00 Max. :27.74 Max. :1.00000   
## nox rm age dis   
## Min. :0.3850 Min. :3.561 Min. : 2.90 Min. : 1.130   
## 1st Qu.:0.4490 1st Qu.:5.886 1st Qu.: 45.02 1st Qu.: 2.100   
## Median :0.5380 Median :6.208 Median : 77.50 Median : 3.207   
## Mean :0.5547 Mean :6.285 Mean : 68.57 Mean : 3.795   
## 3rd Qu.:0.6240 3rd Qu.:6.623 3rd Qu.: 94.08 3rd Qu.: 5.188   
## Max. :0.8710 Max. :8.780 Max. :100.00 Max. :12.127   
## rad tax ptratio lstat   
## Min. : 1.000 Min. :187.0 Min. :12.60 Min. : 1.73   
## 1st Qu.: 4.000 1st Qu.:279.0 1st Qu.:17.40 1st Qu.: 6.95   
## Median : 5.000 Median :330.0 Median :19.05 Median :11.36   
## Mean : 9.549 Mean :408.2 Mean :18.46 Mean :12.65   
## 3rd Qu.:24.000 3rd Qu.:666.0 3rd Qu.:20.20 3rd Qu.:16.95   
## Max. :24.000 Max. :711.0 Max. :22.00 Max. :37.97   
## medv   
## Min. : 5.00   
## 1st Qu.:17.02   
## Median :21.20   
## Mean :22.53   
## 3rd Qu.:25.00   
## Max. :50.00

dim(hodata)

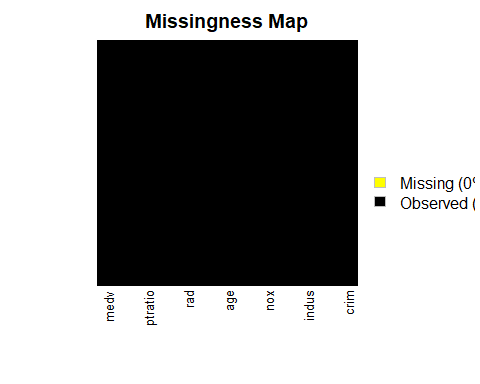
## [1] 506 13

sapply(hodata,class)

## crim zn indus chas nox rm age dis   
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"   
## rad tax ptratio lstat medv   
## "numeric" "numeric" "numeric" "numeric" "numeric"

houseframe<- data.frame(hodata)  
missmap(hodata,col=c('yellow','black'),y.at=1,y.labels='',legend=TRUE)

## Warning: Unknown or uninitialised column: `arguments`.  
## Unknown or uninitialised column: `arguments`.



colSums(is.na(houseframe))

## crim zn indus chas nox rm age dis rad tax   
## 0 0 0 0 0 0 0 0 0 0   
## ptratio lstat medv   
## 0 0 0

cor(hodata$medv, hodata$chas)

## [1] 0.1752602

cor(hodata$medv, hodata$crim)

## [1] -0.3883046

cor(hodata$medv, hodata$zn)

## [1] 0.3604453

cor(hodata$medv, hodata$nox)

## [1] -0.4273208

cor(hodata$medv, hodata$rm)

## [1] 0.6953599

cor(hodata$medv, hodata$dis)

## [1] 0.2499287

cor(hodata$medv, hodata$ptratio)

## [1] -0.5077867

cor(hodata$medv, hodata$lstat)

## [1] -0.7376627

cor(hodata$medv, hodata$indus)

## [1] -0.4837252

cor(hodata$medv, hodata$age)

## [1] -0.3769546

cor(hodata$medv, hodata$tax)

## [1] -0.4685359

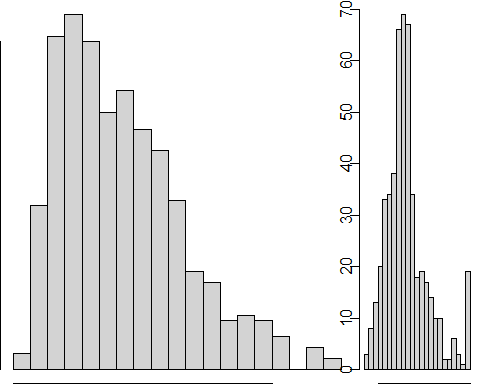
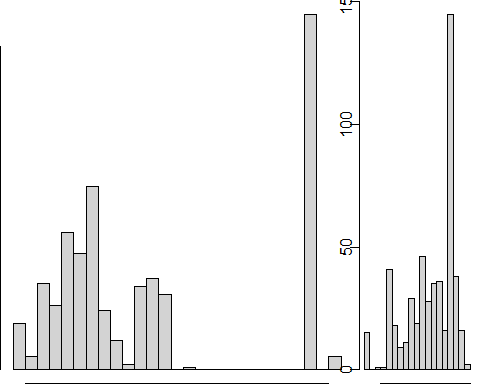
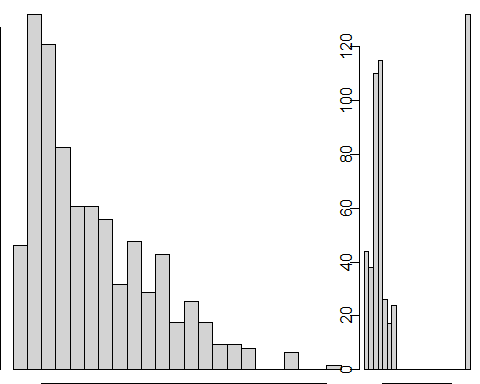
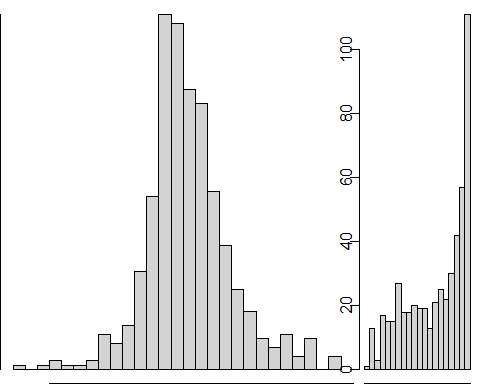
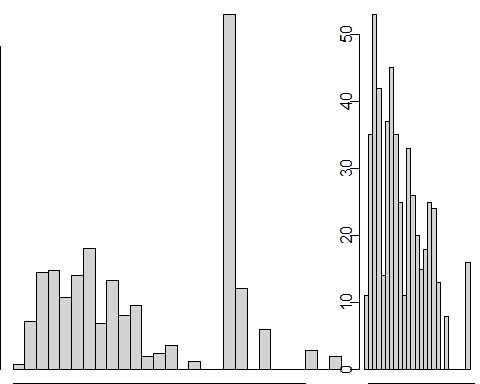
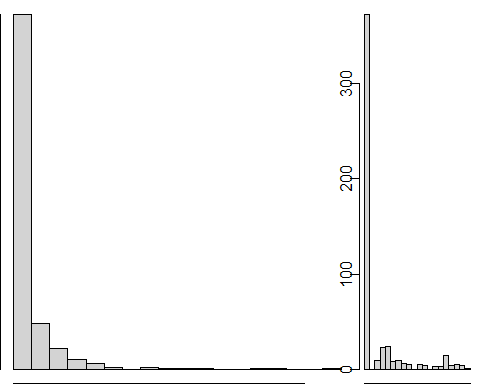
cor(hodata$medv, hodata$rad)

## [1] -0.3816262

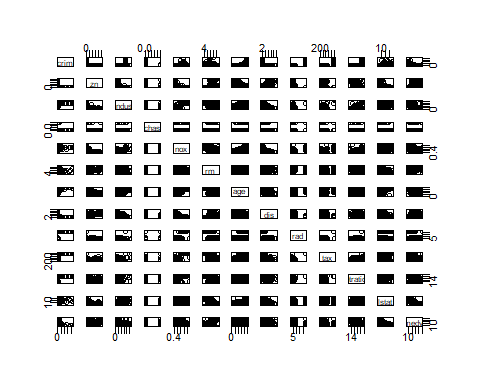
rcorr(as.matrix(houseframe),type='spearman')

## crim zn indus chas nox rm age dis rad tax ptratio  
## crim 1.00 -0.57 0.74 0.04 0.82 -0.31 0.70 -0.74 0.73 0.73 0.47  
## zn -0.57 1.00 -0.64 -0.04 -0.63 0.36 -0.54 0.61 -0.28 -0.37 -0.45  
## indus 0.74 -0.64 1.00 0.09 0.79 -0.42 0.68 -0.76 0.46 0.66 0.43  
## chas 0.04 -0.04 0.09 1.00 0.07 0.06 0.07 -0.08 0.02 -0.04 -0.14  
## nox 0.82 -0.63 0.79 0.07 1.00 -0.31 0.80 -0.88 0.59 0.65 0.39  
## rm -0.31 0.36 -0.42 0.06 -0.31 1.00 -0.28 0.26 -0.11 -0.27 -0.31  
## age 0.70 -0.54 0.68 0.07 0.80 -0.28 1.00 -0.80 0.42 0.53 0.36  
## dis -0.74 0.61 -0.76 -0.08 -0.88 0.26 -0.80 1.00 -0.50 -0.57 -0.32  
## rad 0.73 -0.28 0.46 0.02 0.59 -0.11 0.42 -0.50 1.00 0.70 0.32  
## tax 0.73 -0.37 0.66 -0.04 0.65 -0.27 0.53 -0.57 0.70 1.00 0.45  
## ptratio 0.47 -0.45 0.43 -0.14 0.39 -0.31 0.36 -0.32 0.32 0.45 1.00  
## lstat 0.63 -0.49 0.64 -0.05 0.64 -0.64 0.66 -0.56 0.39 0.53 0.47  
## medv -0.56 0.44 -0.58 0.14 -0.56 0.63 -0.55 0.45 -0.35 -0.56 -0.56  
## lstat medv  
## crim 0.63 -0.56  
## zn -0.49 0.44  
## indus 0.64 -0.58  
## chas -0.05 0.14  
## nox 0.64 -0.56  
## rm -0.64 0.63  
## age 0.66 -0.55  
## dis -0.56 0.45  
## rad 0.39 -0.35  
## tax 0.53 -0.56  
## ptratio 0.47 -0.56  
## lstat 1.00 -0.85  
## medv -0.85 1.00  
##   
## n= 506   
##   
##   
## P  
## crim zn indus chas nox rm age dis rad tax   
## crim 0.0000 0.0000 0.3511 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## zn 0.0000 0.0000 0.3465 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## indus 0.0000 0.0000 0.0434 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## chas 0.3511 0.3465 0.0434 0.1242 0.1866 0.1278 0.0713 0.5812 0.3179  
## nox 0.0000 0.0000 0.0000 0.1242 0.0000 0.0000 0.0000 0.0000 0.0000  
## rm 0.0000 0.0000 0.0000 0.1866 0.0000 0.0000 0.0000 0.0156 0.0000  
## age 0.0000 0.0000 0.0000 0.1278 0.0000 0.0000 0.0000 0.0000 0.0000  
## dis 0.0000 0.0000 0.0000 0.0713 0.0000 0.0000 0.0000 0.0000 0.0000  
## rad 0.0000 0.0000 0.0000 0.5812 0.0000 0.0156 0.0000 0.0000 0.0000  
## tax 0.0000 0.0000 0.0000 0.3179 0.0000 0.0000 0.0000 0.0000 0.0000   
## ptratio 0.0000 0.0000 0.0000 0.0022 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## lstat 0.0000 0.0000 0.0000 0.2561 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## medv 0.0000 0.0000 0.0000 0.0015 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## ptratio lstat medv   
## crim 0.0000 0.0000 0.0000  
## zn 0.0000 0.0000 0.0000  
## indus 0.0000 0.0000 0.0000  
## chas 0.0022 0.2561 0.0015  
## nox 0.0000 0.0000 0.0000  
## rm 0.0000 0.0000 0.0000  
## age 0.0000 0.0000 0.0000  
## dis 0.0000 0.0000 0.0000  
## rad 0.0000 0.0000 0.0000  
## tax 0.0000 0.0000 0.0000  
## ptratio 0.0000 0.0000  
## lstat 0.0000 0.0000  
## medv 0.0000 0.0000

#checking normality  
hist.data.frame(hodata)



#checking linearity  
plot(hodata)



split = sample.split(hodata$medv, SplitRatio = 0.8)  
training\_set = subset(hodata, split == TRUE)  
test\_set = subset(hodata, split == FALSE)  
view(training\_set)  
view(test\_set)  
dim((training\_set))

## [1] 434 13

dim(test\_set)

## [1] 72 13

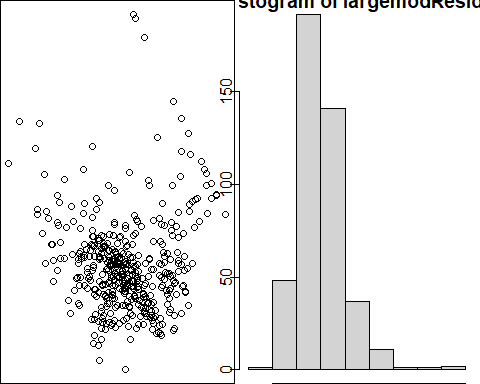
largemod<-lm(medv ~ crim+zn+chas+nox+rm+dis+ptratio+lstat+indus+age+tax+rad ,data = training\_set)  
summary(largemod)

##   
## Call:  
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + ptratio +   
## lstat + indus + age + tax + rad, data = training\_set)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.6983 -3.0338 -0.5558 1.9759 25.4706   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 39.807641 5.364936 7.420 6.53e-13 \*\*\*  
## crim -0.120170 0.033719 -3.564 0.000407 \*\*\*  
## zn 0.048427 0.014659 3.304 0.001036 \*\*   
## chas 3.189485 0.957329 3.332 0.000940 \*\*\*  
## nox -15.874876 4.301163 -3.691 0.000253 \*\*\*  
## rm 3.546953 0.451691 7.853 3.42e-14 \*\*\*  
## dis -1.396266 0.217456 -6.421 3.66e-10 \*\*\*  
## ptratio -0.886724 0.141964 -6.246 1.03e-09 \*\*\*  
## lstat -0.588639 0.054910 -10.720 < 2e-16 \*\*\*  
## indus -0.005455 0.065610 -0.083 0.933772   
## age 0.005420 0.014915 0.363 0.716493   
## tax -0.012506 0.004088 -3.059 0.002360 \*\*   
## rad 0.303176 0.072390 4.188 3.43e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.829 on 421 degrees of freedom  
## Multiple R-squared: 0.7455, Adjusted R-squared: 0.7383   
## F-statistic: 102.8 on 12 and 421 DF, p-value: < 2.2e-16

glance(largemod)

## # A tibble: 1 × 12  
## r.squared adj.r.squ…¹ sigma stati…² p.value df logLik AIC BIC devia…³  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.746 0.738 4.83 103. 6.72e-117 12 -1293. 2613. 2670. 9819.  
## # … with 2 more variables: df.residual <int>, nobs <int>, and abbreviated  
## # variable names ¹​adj.r.squared, ²​statistic, ³​deviance

largemodResids <- largemod$residuals  
largemodFitted <- largemod$fitted.values  
plot(largemodFitted,largemodResids)  
hist(largemodResids)



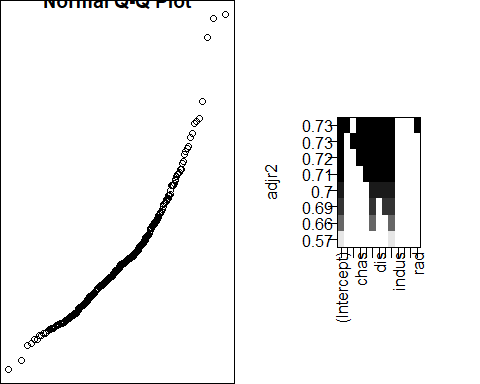
qqnorm(largemodResids)  
  
stargazer(largemod, type="text")

##   
## ===============================================  
## Dependent variable:   
## ---------------------------  
## medv   
## -----------------------------------------------  
## crim -0.120\*\*\*   
## (0.034)   
##   
## zn 0.048\*\*\*   
## (0.015)   
##   
## chas 3.189\*\*\*   
## (0.957)   
##   
## nox -15.875\*\*\*   
## (4.301)   
##   
## rm 3.547\*\*\*   
## (0.452)   
##   
## dis -1.396\*\*\*   
## (0.217)   
##   
## ptratio -0.887\*\*\*   
## (0.142)   
##   
## lstat -0.589\*\*\*   
## (0.055)   
##   
## indus -0.005   
## (0.066)   
##   
## age 0.005   
## (0.015)   
##   
## tax -0.013\*\*\*   
## (0.004)   
##   
## rad 0.303\*\*\*   
## (0.072)   
##   
## Constant 39.808\*\*\*   
## (5.365)   
##   
## -----------------------------------------------  
## Observations 434   
## R2 0.746   
## Adjusted R2 0.738   
## Residual Std. Error 4.829 (df = 421)   
## F Statistic 102.774\*\*\* (df = 12; 421)   
## ===============================================  
## Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

subsetmod <- regsubsets(medv ~ crim+zn+chas+nox+rm+dis+ptratio+lstat+indus+age+tax+rad, data = training\_set)  
summary(subsetmod)

## Subset selection object  
## Call: regsubsets.formula(medv ~ crim + zn + chas + nox + rm + dis +   
## ptratio + lstat + indus + age + tax + rad, data = training\_set)  
## 12 Variables (and intercept)  
## Forced in Forced out  
## crim FALSE FALSE  
## zn FALSE FALSE  
## chas FALSE FALSE  
## nox FALSE FALSE  
## rm FALSE FALSE  
## dis FALSE FALSE  
## ptratio FALSE FALSE  
## lstat FALSE FALSE  
## indus FALSE FALSE  
## age FALSE FALSE  
## tax FALSE FALSE  
## rad FALSE FALSE  
## 1 subsets of each size up to 8  
## Selection Algorithm: exhaustive  
## crim zn chas nox rm dis ptratio lstat indus age tax rad  
## 1 ( 1 ) " " " " " " " " " " " " " " "\*" " " " " " " " "  
## 2 ( 1 ) " " " " " " " " "\*" " " " " "\*" " " " " " " " "  
## 3 ( 1 ) " " " " " " " " "\*" " " "\*" "\*" " " " " " " " "  
## 4 ( 1 ) " " " " " " " " "\*" "\*" "\*" "\*" " " " " " " " "  
## 5 ( 1 ) " " " " " " "\*" "\*" "\*" "\*" "\*" " " " " " " " "  
## 6 ( 1 ) " " " " "\*" "\*" "\*" "\*" "\*" "\*" " " " " " " " "  
## 7 ( 1 ) " " "\*" "\*" "\*" "\*" "\*" "\*" "\*" " " " " " " " "  
## 8 ( 1 ) "\*" " " "\*" "\*" "\*" "\*" "\*" "\*" " " " " " " "\*"

plot(subsetmod, scale = "adjr2" )



reducedmod<- lm(medv ~ crim+zn+chas+nox+rm+dis+ptratio+lstat,data = training\_set)  
summary(reducedmod)

##   
## Call:  
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + ptratio +   
## lstat, data = training\_set)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.5257 -3.1742 -0.6131 1.9614 26.9884   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 32.99371 5.00124 6.597 1.25e-10 \*\*\*  
## crim -0.07637 0.03073 -2.485 0.013332 \*   
## zn 0.04291 0.01430 3.000 0.002854 \*\*   
## chas 3.55864 0.96442 3.690 0.000253 \*\*\*  
## nox -14.47046 3.59359 -4.027 6.70e-05 \*\*\*  
## rm 3.92750 0.44013 8.923 < 2e-16 \*\*\*  
## dis -1.36501 0.20491 -6.661 8.42e-11 \*\*\*  
## ptratio -0.80509 0.12645 -6.367 5.01e-10 \*\*\*  
## lstat -0.58750 0.05205 -11.287 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.914 on 425 degrees of freedom  
## Multiple R-squared: 0.734, Adjusted R-squared: 0.729   
## F-statistic: 146.6 on 8 and 425 DF, p-value: < 2.2e-16

glance(reducedmod)

## # A tibble: 1 × 12  
## r.squared adj.r.squ…¹ sigma stati…² p.value df logLik AIC BIC devia…³  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.734 0.729 4.91 147. 3.93e-117 8 -1302. 2624. 2665. 10262.  
## # … with 2 more variables: df.residual <int>, nobs <int>, and abbreviated  
## # variable names ¹​adj.r.squared, ²​statistic, ³​deviance

anova(reducedmod,largemod)

## Analysis of Variance Table  
##   
## Model 1: medv ~ crim + zn + chas + nox + rm + dis + ptratio + lstat  
## Model 2: medv ~ crim + zn + chas + nox + rm + dis + ptratio + lstat +   
## indus + age + tax + rad  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 425 10261.9   
## 2 421 9818.7 4 443.17 4.7505 0.0009292 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#pvalue should be less than 0.5 here   
  
interactionmod<-lm(medv ~ (crim+zn+chas+nox+rm+dis+ptratio+lstat)^2 ,data = training\_set)  
summary(interactionmod)

##   
## Call:  
## lm(formula = medv ~ (crim + zn + chas + nox + rm + dis + ptratio +   
## lstat)^2, data = training\_set)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.8889 -2.0090 -0.2607 1.7626 24.4090   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.167e+02 3.842e+01 -3.036 0.002552 \*\*   
## crim -3.610e+00 2.850e+00 -1.266 0.206102   
## zn -1.725e-01 2.889e-01 -0.597 0.550870   
## chas 4.058e+01 1.836e+01 2.210 0.027647 \*   
## nox 2.475e+01 4.090e+01 0.605 0.545380   
## rm 2.781e+01 4.570e+00 6.085 2.75e-09 \*\*\*  
## dis -7.127e+00 3.014e+00 -2.365 0.018518 \*   
## ptratio 5.361e+00 1.736e+00 3.088 0.002157 \*\*   
## lstat 2.087e+00 7.850e-01 2.659 0.008159 \*\*   
## crim:zn -7.027e-02 1.536e-01 -0.458 0.647480   
## crim:chas 3.755e+00 7.204e-01 5.213 2.99e-07 \*\*\*  
## crim:nox -2.527e+00 7.708e-01 -3.279 0.001133 \*\*   
## crim:rm 1.249e-01 4.489e-02 2.781 0.005673 \*\*   
## crim:dis -1.741e-02 7.674e-02 -0.227 0.820596   
## crim:ptratio 2.054e-01 1.400e-01 1.468 0.143008   
## crim:lstat 1.531e-02 5.542e-03 2.763 0.005989 \*\*   
## zn:chas -3.549e-02 6.025e-02 -0.589 0.556154   
## zn:nox 3.381e-01 4.019e-01 0.841 0.400749   
## zn:rm -1.026e-02 2.479e-02 -0.414 0.679097   
## zn:dis 6.211e-03 5.833e-03 1.065 0.287643   
## zn:ptratio 6.598e-03 6.102e-03 1.081 0.280213   
## zn:lstat -8.500e-03 3.859e-03 -2.203 0.028182 \*   
## chas:nox -4.198e+01 9.840e+00 -4.266 2.49e-05 \*\*\*  
## chas:rm -1.310e+00 1.554e+00 -0.843 0.399690   
## chas:dis 2.436e-01 1.426e+00 0.171 0.864452   
## chas:ptratio -7.111e-01 5.812e-01 -1.224 0.221859   
## chas:lstat 6.951e-02 1.850e-01 0.376 0.707273   
## nox:rm -6.662e+00 4.811e+00 -1.385 0.166886   
## nox:dis -9.128e+00 2.564e+00 -3.559 0.000417 \*\*\*  
## nox:ptratio 1.938e+00 1.560e+00 1.242 0.214904   
## nox:lstat -4.712e-02 5.631e-01 -0.084 0.933356   
## rm:dis 9.346e-01 2.994e-01 3.122 0.001928 \*\*   
## rm:ptratio -1.088e+00 1.833e-01 -5.937 6.36e-09 \*\*\*  
## rm:lstat -2.992e-01 4.761e-02 -6.285 8.66e-10 \*\*\*  
## dis:ptratio 1.566e-01 9.570e-02 1.637 0.102527   
## dis:lstat 1.659e-01 3.990e-02 4.157 3.95e-05 \*\*\*  
## ptratio:lstat -7.353e-02 2.325e-02 -3.163 0.001683 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.435 on 397 degrees of freedom  
## Multiple R-squared: 0.8786, Adjusted R-squared: 0.8676   
## F-statistic: 79.79 on 36 and 397 DF, p-value: < 2.2e-16

redinmodel<-lm(medv ~ crim+zn+chas+nox+rm+dis+ptratio+lstat+rm\*ptratio+rm\*lstat,data = training\_set)  
coef(redinmodel)

## (Intercept) crim zn chas nox   
## -79.910720274 -0.121234438 0.007800952 3.431178108 -5.554386651   
## rm dis ptratio lstat rm:ptratio   
## 20.042749106 -0.847672840 3.896675452 1.465466283 -0.691379192   
## rm:lstat   
## -0.361945864

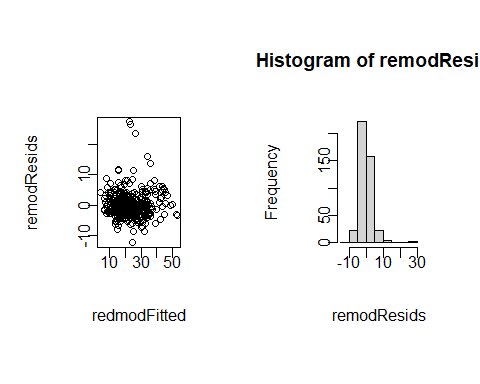
summary(redinmodel)

##   
## Call:  
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + ptratio +   
## lstat + rm \* ptratio + rm \* lstat, data = training\_set)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.3656 -2.1979 -0.5115 1.5146 27.3108   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -79.910720 14.530566 -5.499 6.60e-08 \*\*\*  
## crim -0.121234 0.025550 -4.745 2.86e-06 \*\*\*  
## zn 0.007801 0.012144 0.642 0.5210   
## chas 3.431178 0.798252 4.298 2.14e-05 \*\*\*  
## nox -5.554387 3.030944 -1.833 0.0676 .   
## rm 20.042749 2.147409 9.333 < 2e-16 \*\*\*  
## dis -0.847673 0.174675 -4.853 1.71e-06 \*\*\*  
## ptratio 3.896675 0.798618 4.879 1.51e-06 \*\*\*  
## lstat 1.465466 0.207192 7.073 6.32e-12 \*\*\*  
## rm:ptratio -0.691379 0.124749 -5.542 5.27e-08 \*\*\*  
## rm:lstat -0.361946 0.035326 -10.246 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.053 on 423 degrees of freedom  
## Multiple R-squared: 0.8199, Adjusted R-squared: 0.8157   
## F-statistic: 192.6 on 10 and 423 DF, p-value: < 2.2e-16

anova(redinmodel,interactionmod)

## Analysis of Variance Table  
##   
## Model 1: medv ~ crim + zn + chas + nox + rm + dis + ptratio + lstat +   
## rm \* ptratio + rm \* lstat  
## Model 2: medv ~ (crim + zn + chas + nox + rm + dis + ptratio + lstat)^2  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 423 6947.9   
## 2 397 4684.9 26 2263 7.3756 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

remodResids <- redinmodel$residuals  
redmodFitted <- redinmodel$fitted.values  
plot(redmodFitted,remodResids)  
hist(remodResids)



qqnorm(remodResids)  
  
glance(redinmodel)

## # A tibble: 1 × 12  
## r.squared adj.r.squ…¹ sigma stati…² p.value df logLik AIC BIC devia…³  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.820 0.816 4.05 193. 1.35e-150 10 -1218. 2459. 2508. 6948.  
## # … with 2 more variables: df.residual <int>, nobs <int>, and abbreviated  
## # variable names ¹​adj.r.squared, ²​statistic, ³​deviance

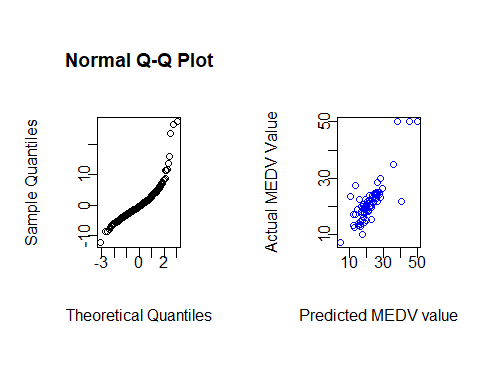
largeCVModel <- train(  
 form =medv ~ crim+zn+chas+nox+rm+dis+ptratio+lstat+rm\*ptratio+rm:lstat,  
 data = training\_set,  
 method = "lm",  
 trControl = trainControl(method = "cv", number = 10)  
 )  
largeCVModel

## Linear Regression   
##   
## 434 samples  
## 8 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 390, 390, 391, 391, 391, 390, ...   
## Resampling results:  
##   
## RMSE Rsquared MAE   
## 3.95452 0.8118935 2.765967  
##   
## Tuning parameter 'intercept' was held constant at a value of TRUE

summary(largeCVModel)

##   
## Call:  
## lm(formula = .outcome ~ ., data = dat)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.3656 -2.1979 -0.5115 1.5146 27.3108   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -79.910720 14.530566 -5.499 6.60e-08 \*\*\*  
## crim -0.121234 0.025550 -4.745 2.86e-06 \*\*\*  
## zn 0.007801 0.012144 0.642 0.5210   
## chas 3.431178 0.798252 4.298 2.14e-05 \*\*\*  
## nox -5.554387 3.030944 -1.833 0.0676 .   
## rm 20.042749 2.147409 9.333 < 2e-16 \*\*\*  
## dis -0.847673 0.174675 -4.853 1.71e-06 \*\*\*  
## ptratio 3.896675 0.798618 4.879 1.51e-06 \*\*\*  
## lstat 1.465466 0.207192 7.073 6.32e-12 \*\*\*  
## `rm:ptratio` -0.691379 0.124749 -5.542 5.27e-08 \*\*\*  
## `rm:lstat` -0.361946 0.035326 -10.246 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.053 on 423 degrees of freedom  
## Multiple R-squared: 0.8199, Adjusted R-squared: 0.8157   
## F-statistic: 192.6 on 10 and 423 DF, p-value: < 2.2e-16

predictionvalue<-predict(largeCVModel,test\_set)  
plot(predictionvalue,test\_set$medv,xlab="Predicted MEDV value", ylab="Actual MEDV Value",col="Blue")



test\_set$predictionvalue<-predict(largeCVModel,test\_set)  
acpred<-data.frame(test\_set$medv,test\_set$predictionvalue)  
names(acpred)<-c("medv","predictionvalue")  
correlation\_accuracy<-cor(acpred)  
correlation\_accuracy

## medv predictionvalue  
## medv 1.0000000 0.8303563  
## predictionvalue 0.8303563 1.0000000

predict(largeCVModel, test\_set, interval="predict")

## 1 2 3 4 5 6 7 8   
## 26.523356 19.087211 21.142424 21.794884 13.150634 19.219573 29.625559 22.035677   
## 9 10 11 12 13 14 15 16   
## 27.662360 24.808982 20.443075 17.738872 26.084213 24.180976 22.478841 25.417278   
## 17 18 19 20 21 22 23 24   
## 23.812682 26.665574 18.820639 26.274162 20.105177 15.378197 17.810026 23.400474   
## 25 26 27 28 29 30 31 32   
## 27.591960 28.595232 49.960489 45.170115 21.764143 38.511327 35.743287 16.333453   
## 33 34 35 36 37 38 39 40   
## 17.507482 20.839921 11.070616 28.188378 20.272559 21.738756 20.012607 25.154115   
## 41 42 43 44 45 46 47 48   
## 24.329773 18.492607 20.561428 25.459600 23.345675 26.085601 25.036134 22.864725   
## 49 50 51 52 53 54 55 56   
## 19.945268 21.668781 26.858496 23.721473 19.038317 19.562398 40.505883 20.143728   
## 57 58 59 60 61 62 63 64   
## 16.748032 12.505119 14.063791 5.303027 18.438230 12.594542 17.696320 14.037335   
## 65 66 67 68 69 70 71 72   
## 17.066648 15.361751 16.603372 14.920570 16.452842 20.497194 17.803380 18.984556