目录

实际数据分析报告

——以 BostonHousing 为例

第 18 组 肖扬 黄诗婕 韩明浩 叶冷竹

一、包的调用与数据导入

```
library(mlbench) # 调用"mlbench"包
library(car) # 调用 car 包
library(MASS) # 调用“MASS”包
library(corpcor) # 调用“corpcor”包
library(corrplot) # 调用“corrplot”包
library(lars) # 调用 lars 包

data("BostonHousing") # 载入"BostonHousing"数据
```

二、描述性统计

1、变量定义：

CRIM: 城镇人均犯罪率

ZN: 占地面积超过 25,000 平方英尺的住宅用地比例

INDUS: 城镇非零售商用土地的比例

CHAS: 查理斯河空变量（如果边界是河流则为 1；否则为 0）

NOX: 一氧化氮浓度

RM: 住宅平均房间数

AGE: 1940 年以前建成的自用房屋比例

DIS: 与波士顿五个就业中心的加权距离

RAD: 辐射性公路的接近指数

TAX: 每一万美元的全值财产税率

PRTATIO: 城镇中的教师学生比例

B: 1000(Bk-0.63)^ 2，其中 Bk 指代城镇中黑人的比例

LSTAT: 人口中地位低下者的比例

MEDV：自住房的中位数房价，以千美元计

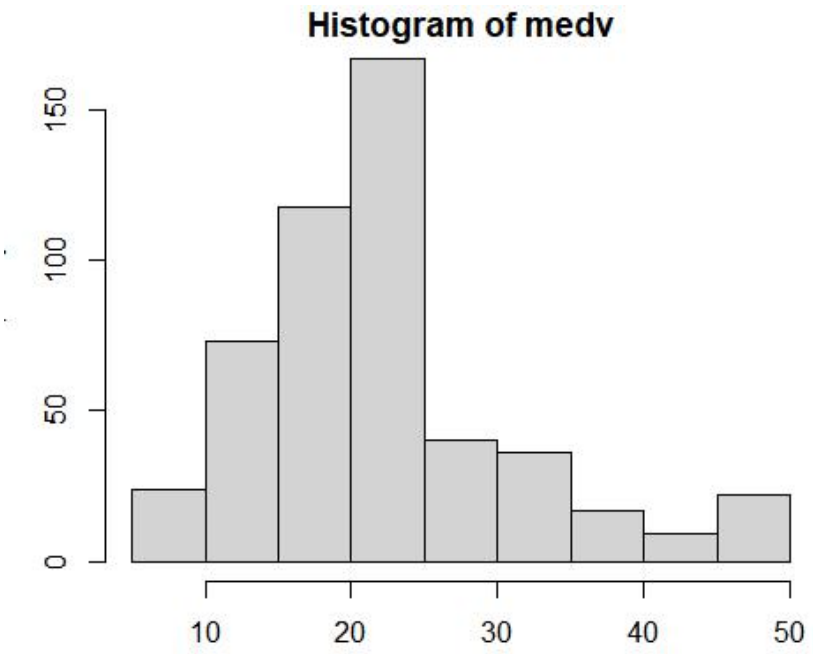
2、均值、中位数、极值与四分位点

```
data = BostonHousing[,-c(4)] # 删除“chas”变量
summary(data) # 计算各变量基本统计量
```

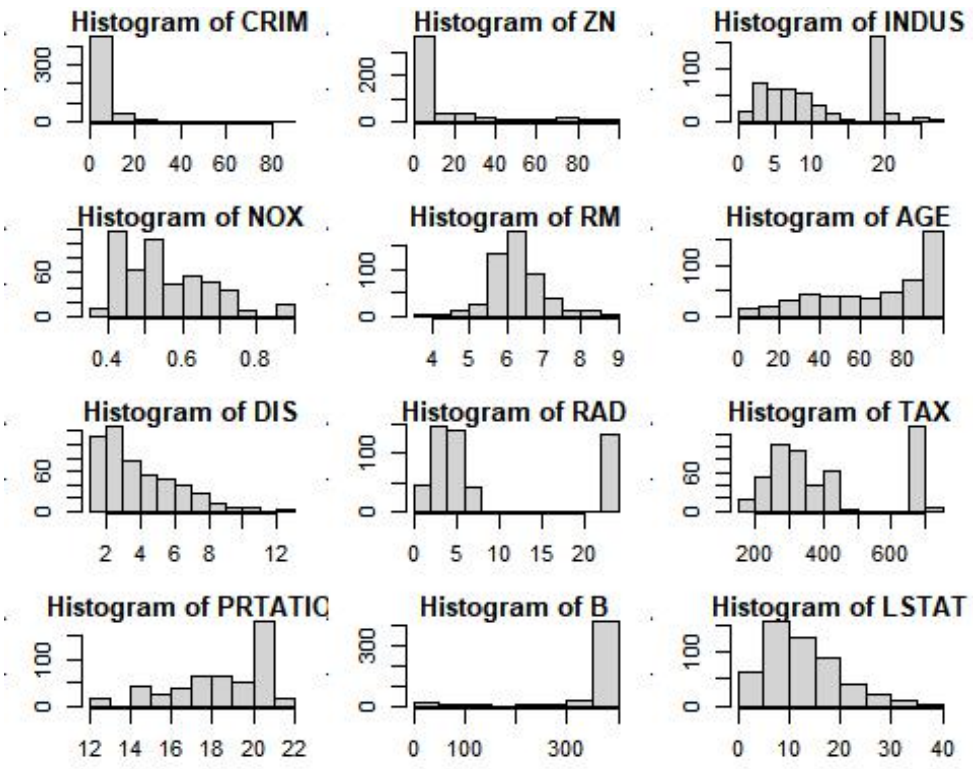
```
##       crim                zn              indus            nox        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.3850  
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.4490  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.5380  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.5547  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.6240  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :0.8710  
##        rm              age              dis              rad        
##  Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000  
##  1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000  
##  Median :6.208   Median : 77.50   Median : 3.207   Median : 5.000  
##  Mean   :6.285   Mean   : 68.57   Mean   : 3.795   Mean   : 9.549  
##  3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000  
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000  
##       tax             ptratio            b              lstat      
##  Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73  
##  1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95  
##  Median :330.0   Median :19.05   Median :391.44   Median :11.36  
##  Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65  
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95  
##  Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97  
##       medv      
##  Min.   : 5.00  
##  1st Qu.:17.02  
##  Median :21.20  
##  Mean   :22.53  
##  3rd Qu.:25.00  
##  Max.   :50.00
```

3、绘制直方图

```
x = data[,c(1:12)] # 将自变量数据赋值给x
y = data[,c(13)] # 将因变量数据赋值给y

title = c("CRIM","ZN","INDUS","NOX","RM","AGE","DIS",
          "RAD","TAX","PRTATIO","B","LSTAT") # 提取自变量名称

par(mfrow=c(1,1)) # 设置绘图布局
par(mar = c(3,3,1,1)) # 设置绘图边距
hist(y,main='Histogram of medv')# 绘制因变量直方图
```

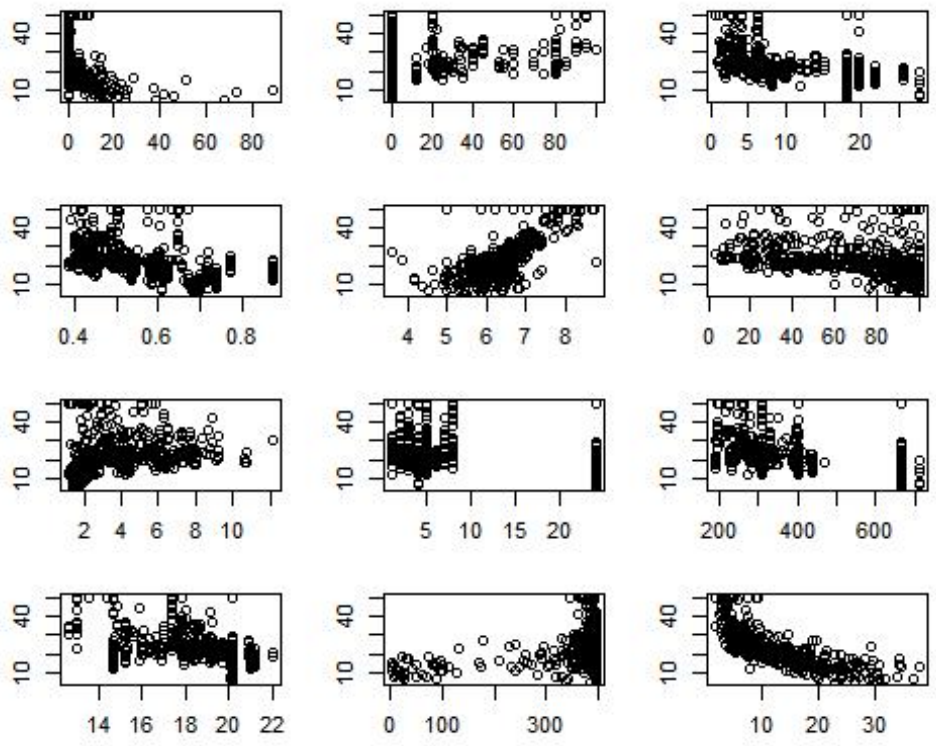



```
par(mfrow=c(4,3)) # 设置绘图布局
for(i in 1:12){
  hist(x[,i], xlab = paste0(title[i]),main=paste('Histogram of',title
[i]))
} # 绘制各自变量直方图
```



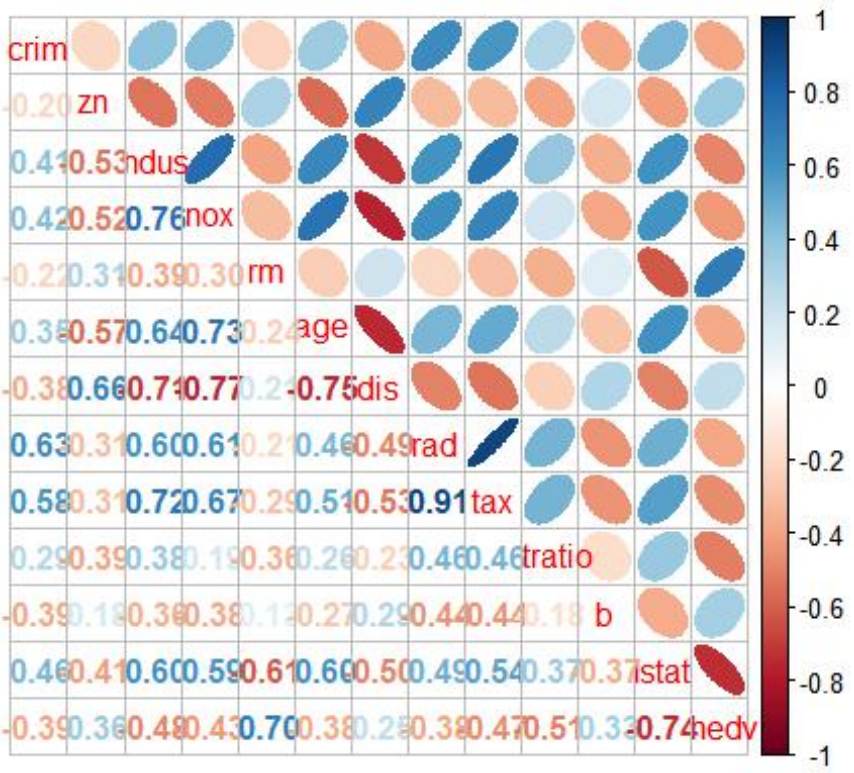
4、绘制散点图

```
par(mfrow=c(4,3)) # 设置绘图布局
par(mar = c(3,3,1,1)) # 设置绘图边距
for(i in 1:12){
  plot(x[,i], y, xlab = paste0(title[i]),ylab = "medv")
} # 绘制因变量与各自变量的一维散点图
```



5、绘制相关系数矩阵图

```
mycor <- cor(data) # 计算数据相关系数
par(mfrow=c(1,1)) # 设置绘图布局
corrplot.mixed(mycor, upper = "ellipse") # 绘制相关系数矩阵图
```



从图中可以看出因变量与各自变量之间的相关性。如图，rm 与 medv 的正相关性最强，为 0.7；lstat 与 medv 的负相关性最强，为-0.74。对此，我们的解释是，每个住宅的平均房间数越高，说明住宅平均占地面积更大，那么房价也就更高。人口中较低地位的百分比越低，说明该地穷人越多，偏贫民窟的可能性更大，贫民窟一般房价更低。

此外，各个自变量之间也有一定相关关系，相关系数绝对值超过 0.7 的有：indus 与 nox，indus 与 dis，nox 与 age，nox 与 dis，age 与 dis，rad 与 tax。对此，我们的解释是，每镇非零售业亩的工业用地比例越高，说明工业化程度更高，那么一氧化氮排放量将会更高；为方便就业，与波士顿五个就业中心的加权距离也会更低。一般来说，一个地区的发展历史越久，该地区被污染的可能性也就较大，所以一氧化氮排放量也就可能更高。而与就业中心的距离越近，说明该地区工厂分布更密集，污染程度也就更高。更容易到达公路的地区，经济会更发达，那么税收也会更多。

三、基础模型构建

```
data.1 = data.frame(scale(data)) # 对数据进行标准化

fit.1 = lm(medv~.-1,data=data.1) # 拟合线性模型
summary(fit.1) # 输出拟合结果

##
## Call:
## lm(formula = medv ~ . - 1, data = data.1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.45663 -0.30556 -0.07019  0.20812  2.86780
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## crim    -0.10581    0.03094  -3.420 0.000678 ***
## zn       0.11932    0.03508   3.401 0.000725 ***
## indus    0.03007    0.04598   0.654 0.513462
## nox     -0.21881    0.04847  -4.514 7.96e-06 ***
## rm       0.29416    0.03216   9.147  < 2e-16 ***
## age      0.00852    0.04069   0.209 0.834241
## dis     -0.34008    0.04602  -7.391 6.29e-13 ***
## rad      0.31083    0.06293   4.939 1.08e-06 ***
## tax     -0.25208    0.06894  -3.657 0.000283 ***
## ptratio -0.23327    0.03090  -7.549 2.13e-13 ***
## b        0.09670    0.02683   3.604 0.000346 ***
## lstat   -0.41474    0.03961 -10.470  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.52 on 494 degrees of freedom
## Multiple R-squared:  0.7355, Adjusted R-squared:  0.7291
## F-statistic: 114.5 on 12 and 494 DF,  p-value: < 2.2e-16
```

从上面基础模型的输出结果可以看出，存在不显著的自变量，故先进行变量选择

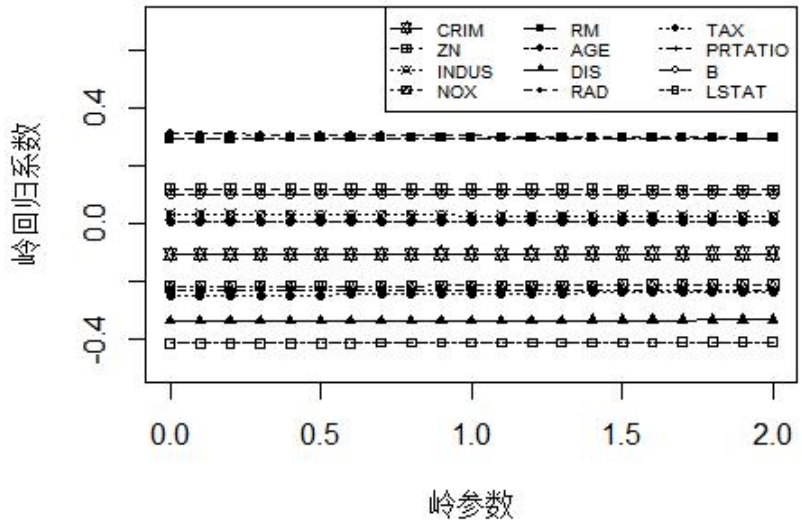
四、变量选择

```
vif(fit.1) # 检验多重共线性

## Warning in vif.default(fit.1): No intercept: vifs may not be sensible.
##     crim       zn    indus      nox       rm      age      dis      rad
## 1.787705 2.298257 3.949246 4.388775 1.931865 3.092832 3.954961 7.3978
44
##      tax  ptratio        b    lstat
## 8.876233 1.783302 1.344971 2.931101
```

可以看到，部分变量 VIF 值接近 10，说明存在一定的多重共线性

1、岭回归

```
fit.ridge = lm.ridge(medv~.-1,data=data.1,lambda=seq(0,2,0.1)) # 岭回归
beta = coef(fit.ridge) # 将所有不同岭参数对应的回归系数赋给 beta
# 绘制岭迹图
k = fit.ridge$lambda # 将所有岭参数赋给 k
plot(k,k,type='n',xlab='岭参数',ylab='岭回归系数',
     ylim=c(-0.5,0.7)) # 创建没有任何点和线的图形区域
linetype = c(1:12)
char = c(11:22)
for(i in 1:12){
  lines(k,beta[,i],type='o',lty=linetype[i],pch=char[i],cex=0.75)
} # 画岭迹线
legend('topright',legend=title,cex=0.6,lty=linetype,pch=char,ncol=3) #
添加图例
```



上述岭迹图显示各自变量岭回归系数稳定，故不易通过岭回归直接筛选变量

2、逐步回归法

```
fit.2 = step(fit.1,direction="both") # 逐步回归法筛选自变量
```

```
## Start:  AIC=-649.97
## medv ~ (crim + zn + indus + nox + rm + age + dis + rad + tax +
##     ptratio + b + lstat) - 1
##
##           Df Sum of Sq    RSS     AIC
## - age      1    0.0119 133.58 -651.92
## - indus    1    0.1156 133.68 -651.53
## <none>                 133.56 -649.97
## - zn       1    3.1283 136.69 -640.25
## - crim     1    3.1628 136.73 -640.13
## - b        1    3.5109 137.07 -638.84
## - tax      1    3.6151 137.18 -638.46
## - nox      1    5.5093 139.07 -631.52
## - rad      1    6.5950 140.16 -627.58
## - dis      1   14.7678 148.33 -598.90
## - ptratio  1   15.4088 148.97 -596.72
## - rm       1   22.6195 156.18 -572.81
## - lstat    1   29.6362 163.20 -550.57
##
## Step:  AIC=-651.92
## medv ~ crim + zn + indus + nox + rm + dis + rad + tax + ptratio +
##     b + lstat - 1
##
##           Df Sum of Sq    RSS     AIC
## - indus    1     0.116 133.69 -653.48
## <none>                 133.58 -651.92
## + age      1     0.012 133.56 -649.97
## - zn       1     3.127 136.70 -642.21
## - crim     1     3.162 136.74 -642.08
## - b        1     3.554 137.13 -640.64
## - tax      1     3.607 137.18 -640.44
## - nox      1     5.787 139.36 -632.46
## - rad      1     6.589 140.16 -629.56
## - ptratio  1    15.430 149.01 -598.61
## - dis      1    16.418 149.99 -595.26
## - rm       1    23.876 157.45 -570.71
## - lstat    1    33.035 166.61 -542.10
##
## Step:  AIC=-653.48
## medv ~ crim + zn + nox + rm + dis + rad + tax + ptratio + b +
##     lstat - 1
##
##           Df Sum of Sq    RSS     AIC
## <none>                 133.69 -653.48
## + indus    1     0.116 133.58 -651.92
## + age      1     0.012 133.68 -651.53
```

```
## - zn       1     3.034 136.73 -644.13
## - crim     1     3.220 136.91 -643.44
## - b        1     3.521 137.21 -642.33
## - tax      1     3.772 137.46 -641.41
## - nox      1     5.793 139.49 -634.02
## - rad      1     6.609 140.30 -631.07
## - ptratio  1    15.334 149.03 -600.54
## - dis      1    17.962 151.65 -591.70
## - rm       1    23.770 157.46 -572.68
## - lstat    1    32.930 166.62 -544.07
```

从最终输出结果可以看到，逐步回归建议删除“indus”和“age”两个变量，此时模型的 AIC 值达到最小

3、Lasso 回归

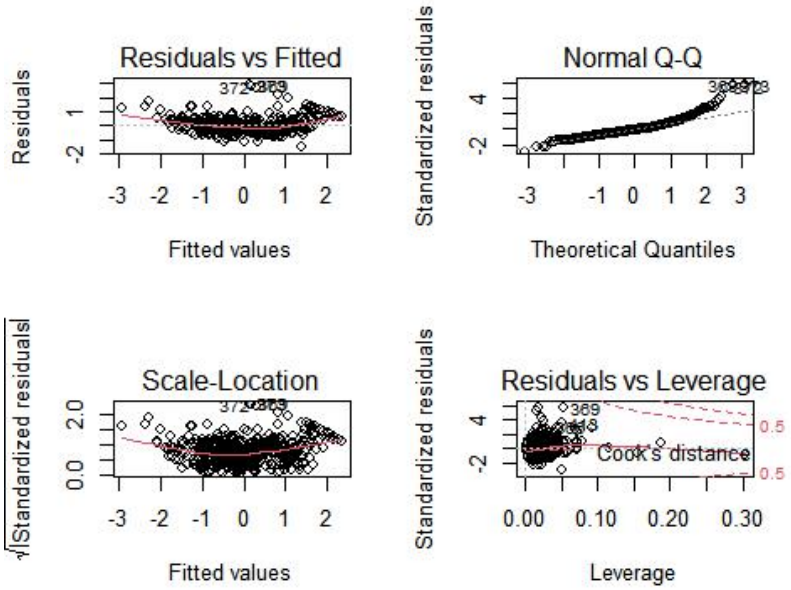
```
fit.lar = lars(as.matrix(x),as.matrix(y),type="lasso") # Lasso 回归

summary(fit.lar) # 输出 cp 值
```

```
## LARS/LASSO
## Call: lars(x = as.matrix(x), y = as.matrix(y), type = "lasso")
##    Df   Rss       Cp
## 0   1 42716 1360.011
## 1   2 36326 1083.143
## 2   3 21335  431.009
## 3   4 14960  154.804
## 4   5 13867  109.115
## 5   6 13720  104.718
## 6   7 13262   86.725
## 7   8 12658   62.354
## 8   9 12186   43.749
## 9  10 12091   41.625
## 10 11 11328   10.335
## 11 12 11310   11.514
## 12 13 11298   13.000
fit.lar # 查看应删除变量
##
## Call:
## lars(x = as.matrix(x), y = as.matrix(y), type = "lasso")
## R-squared: 0.736
## Sequence of LASSO moves:
##      lstat rm ptratio  b crim dis nox zn rad tax indus age
## Var     12  5      10 11    1   7   4  2   8   9     3   6
## Step     1  2       3  4    5   6   7  8   9  10    11  12
```

综合上述输出结果可以发现，Lasso 回归在第十步时模型 Cp 值最小， 而第十一和十二步的变量分别为“indus”和“age”，故同样建议删除这两个变量

4、删除“indus”和“age”后的新模型

```
data.2 = data.1[,-c(3,6)] # 删除“indus”和“age”两个自变量
fit.2 = lm(medv~.-1,data=data.2) # 拟合新的线性模型
summary(fit.2) # 输出新的拟合结果

##
## Call:
## lm(formula = medv ~ . - 1, data = data.2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.45389 -0.30382 -0.05989  0.20596  2.87027
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## crim    -0.10667    0.03086  -3.456 0.000595 ***
## zn       0.11600    0.03457   3.355 0.000854 ***
## nox     -0.20750    0.04476  -4.636 4.55e-06 ***
## rm       0.29371    0.03128   9.391  < 2e-16 ***
## dis     -0.34941    0.04280  -8.163 2.71e-15 ***
## rad      0.29873    0.06033   4.952 1.01e-06 ***
## tax     -0.23226    0.06208  -3.741 0.000205 ***
## ptratio -0.23032    0.03054  -7.542 2.22e-13 ***
## b        0.09658    0.02672   3.614 0.000332 ***
## lstat   -0.41004    0.03710 -11.053  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5192 on 496 degrees of freedom
## Multiple R-squared:  0.7353, Adjusted R-squared:  0.7299
## F-statistic: 137.8 on 10 and 496 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2)) # 设置绘图布局
plot(fit.2) # 绘制回归诊断结果图
```

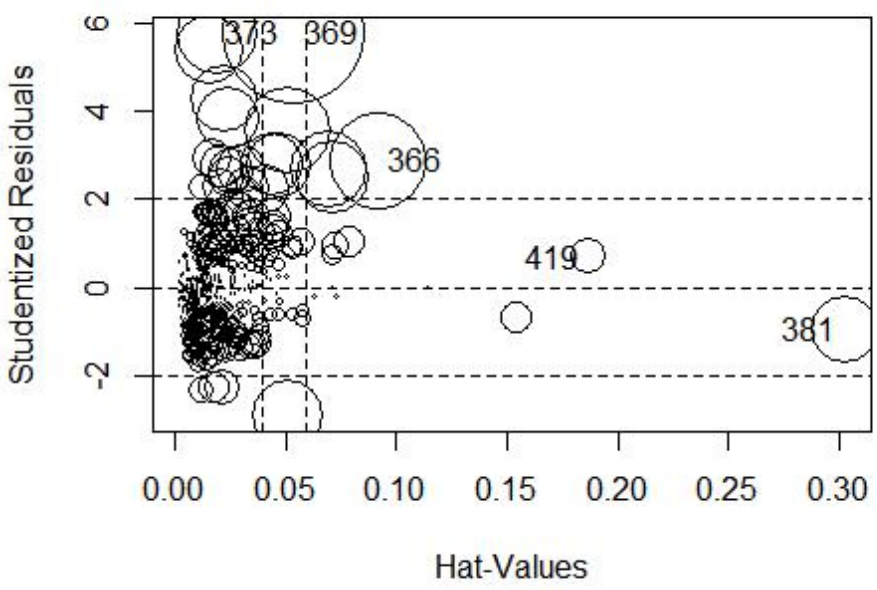


上述输出结果显示，删除两个不显著变量后，R 方和调整后 R 方均变化不大，但 F 值有较明显的提升，说明整体模型更加显著

不过，诊断图显示模型残差尚不符合正态性假设且存在异方差性， 亦可能存在异常值点，故考虑进一步优化数据和模型

五、异常值处理

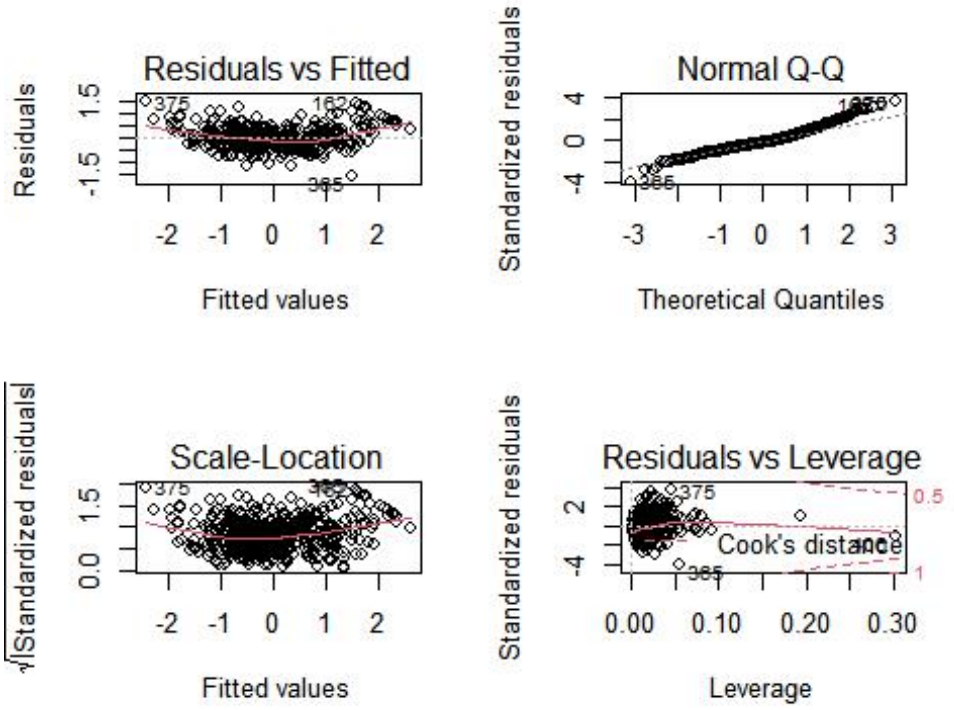
```
par(mfrow = c(1,1)) # 设置绘图模式
influencePlot(fit.2) # 呈现异常值点
```




```
##        StudRes        Hat      CookD
## 366  2.8632912 0.09161548 0.08150281
## 369  5.7675096 0.05383540 0.17770857
## 373  5.7583815 0.01846919 0.05859518
## 381 -0.9503182 0.30277544 0.03922572
## 419  0.6871445 0.18702355 0.01087369

# 计算删除学生化残差、杠杆值、库克距离，并整理为数据框abnormal_test_df
abnormal_test_df = data.frame(rstudent(fit.2),hatvalues(fit.2),cooks.di
stance(fit.2))
# 分别根据删除学生化残差的绝对值、杠杆值、库克距离排序,展示前数列
head(abnormal_test_df[order(-abs(abnormal_test_df$rstudent.fit.2.)),])

##     rstudent.fit.2. hatvalues.fit.2. cooks.distance.fit.2.
## 369        5.767510       0.05383540            0.17770857
## 373        5.758382       0.01846919            0.05859518
## 372        5.382262       0.01519719            0.04231751
## 370        4.254720       0.02274597            0.04073018
## 371        3.816851       0.02441435            0.03548701
## 413        3.554349       0.05138456            0.06686426
```

```
head(abnormal_test_df[order(-abnormal_test_df$hatvalues.fit.2.),])
##     rstudent.fit.2. hatvalues.fit.2. cooks.distance.fit.2.
## 381    -0.950318205      0.30277544         3.922572e-02
## 419     0.687144507      0.18702355         1.087369e-02
## 406    -0.708240344      0.15440876         9.168734e-03
## 411     0.003016869      0.11541199         1.189869e-07
## 366     2.863291240      0.09161548         8.150281e-02
## 491     1.031285231      0.07896278         9.116900e-03
head(abnormal_test_df[order(-abnormal_test_df$cooks.distance.fit.2.),])
##     rstudent.fit.2. hatvalues.fit.2. cooks.distance.fit.2.
## 369       5.767510      0.05383540           0.17770857
## 366       2.863291      0.09161548           0.08150281
## 413       3.554349      0.05138456           0.06686426
## 373       5.758382      0.01846919           0.05859518
## 368       2.678373      0.06937834           0.05282269
## 415       2.505076      0.07065501           0.04720772

# 假定不存在登记误差和测量误差，综合考量上述结果，在模型中删除 366、368、369、
370、371、372、373、381、413、415、419 次观测并重新拟合
abnormal_index = c(369,373,372,370,371,381,419,413,366,368,415)
fit.3 = lm(medv~.-1,data=data.2[-abnormal_index,])
summary(fit.3) # 输出新的拟合结果

## Call:
## lm(formula = medv ~ . - 1, data = data.2[-abnormal_index, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5897 -0.2955 -0.0904  0.1603  1.4801
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## crim    -0.08464    0.03503  -2.416 0.016044 *
## zn       0.07906    0.02809   2.815 0.005076 **
## nox     -0.16384    0.03651  -4.488 9.01e-06 ***
## rm       0.43382    0.02798  15.503  < 2e-16 ***
## dis     -0.24289    0.03533  -6.875 1.91e-11 ***
## rad      0.18141    0.05073   3.576 0.000384 ***
## tax     -0.22744    0.05019  -4.531 7.38e-06 ***
## ptratio -0.22283    0.02471  -9.020  < 2e-16 ***
## b        0.11847    0.02234   5.304 1.73e-07 ***
## lstat   -0.25145    0.03275  -7.678 8.98e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4187 on 485 degrees of freedom
## Multiple R-squared:  0.8123, Adjusted R-squared:  0.8084
## F-statistic: 209.9 on 10 and 485 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2)) # 设置绘图布局
plot(fit.3) # 绘制回归诊断结果图
```




```
#重新计算删除学生化残差、杠杆值、库克距离，并整理为数据框 abnormal_verify_df
abnormal_verify_df = data.frame(rstudent(fit.3),hatvalues(fit.3),cooks.
distance(fit.3))
head(abnormal_verify_df[order(-abs(abnormal_verify_df$rstudent.fit.
3.)),])
##     rstudent.fit.3. hatvalues.fit.3. cooks.distance.fit.3.
## 365       -3.966042       0.05582706            0.09026399
## 375        3.665115       0.04605991            0.06323891
## 162        3.455562       0.02280575            0.02725290
## 187        3.200012       0.01817577            0.01860227
## 167        3.009111       0.02670729            0.02444046
## 163        2.975846       0.02602162            0.02328241
```

多次尝试后我们发现仍然存在异常值，虽然 R 方和 F 值有较显著提升， 但除了正态性有些许改善外，其余问题并没有得到解决。 我们认为这很可能因为普通最小二乘并不适合该数据集， 故接下来采用多元加权最小二乘和 Box-Cox 变化处理异方差性

六、异方差性处理

1、多元加权最小二乘

```
#Spearman 相关系数的计算
data.3 = data.2[-abnormal_index,]
e2 = resid(fit.3) #计算新回归中残差
spearman_result = list() #新建一个列表用于储存检验结果
cor.spearman = vector() #新建一个向量，用于储存每个检验的p 值
abse2 = abs(e2) #取残差的绝对值
for(i in 1:10){
  spearman_result[[i]] = cor.test(data.3[,i],abse2,method = "spearman")
  cor.spearman[i] = cor.test(data.3[,i],abse2,method = "spearman")$p.va
lue
} #使用Spearman 相关系数对自变量和残差绝对值之间相关性进行检验
spearman_result #输出Spearman 相关系数计算结果


## [[1]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 19300067, p-value = 0.3152
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.04523555
##
## [[2]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 18726229, p-value = 0.1018
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.07362302
##
## [[3]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 18790658, p-value = 0.1176
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.07043573
```

```
## [[4]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 16314524, p-value = 1.545e-05
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.1929288
##
## [[5]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 22962934, p-value = 0.002434
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.1359646
##
## [[6]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 18069445, p-value = 0.0182
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.1061138
##
## [[7]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 18633664, p-value = 0.08218
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.07820214
##
## [[8]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 22378884, p-value = 0.01717
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.107072
##
```

```
## [[9]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 21027239, p-value = 0.372
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.04020676
##
## [[10]]
##  Spearman's rank correlation rho
##
## data:  data.3[, i] and abse2
## S = 21186906, p-value = 0.2854
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.04810544
```

```
cor.spearman #输出对每个自变量进行Spearman 检验的p 值
```

```
##  [1] 3.151872e-01 1.018211e-01 1.175645e-01 1.544702e-05 2.433717e-03
##  [6] 1.819699e-02 8.218409e-02 1.717069e-02 3.720482e-01 2.854343e-01
```

```
names(data.3)[cor.spearman<0.05]#输出小于0.5 的变量名
## [1] "rm"      "dis"     "rad"     "ptratio"
```

```
which.min(cor.spearman)#第四个自变量"rm"的p 值最小，即等级相关系数最大
## [1] 4
```

```
#多元加权最小二乘
data.4 = BostonHousing[-abnormal_index,-c(3,4,7)]
#剔除原始数据中的"chas""age""indus",得到未标准化的数据data.4
s = seq(-2,2,0.5) #产生数列:-2,-1.5,-1,...,1.5,2
logLik.list1 = list() #新建一个列表，储存不同权函数下的对数似然函数值
result.w.list1 = list() #新建一个列表，储存不同权函数下的回归模型结果
for(i in 1 : length(s)){
  w = data.4[,4] ^ (-s[i]) #计算不同权函数下的权重值
  result.w = lm(medv ~ . ,weights = w,data.3) #用加权最小二乘拟合线性模型
  logLik.list1[[i]] = logLik(result.w) #储存对数似然函数值
  result.w.list1[[i]] = summary(result.w) #储存回归模型结果
}
logLik.list1 #输出不同权函数下的对数似然函数值
## [[1]]
## 'log Lik.' -285.6582 (df=12)
```

```
## [[2]]
## 'log Lik.' -278.8132 (df=12)
## [[3]]
## 'log Lik.' -272.8888 (df=12)
## [[4]]
## 'log Lik.' -267.8266 (df=12)
## [[5]]
## 'log Lik.' -263.5645 (df=12)
## [[6]]
## 'log Lik.' -260.0387 (df=12)
## [[7]]
## 'log Lik.' -257.1877 (df=12)
## [[8]]
## 'log Lik.' -254.9542 (df=12)
## [[9]]
## 'log Lik.' -253.2886 (df=12)
```

```
m = 0.5*which.max(logLik.list1)-2.5 #计算最优权函数中的参数m
m #输出参数m，发现为2
```

```
## [1] 2
```

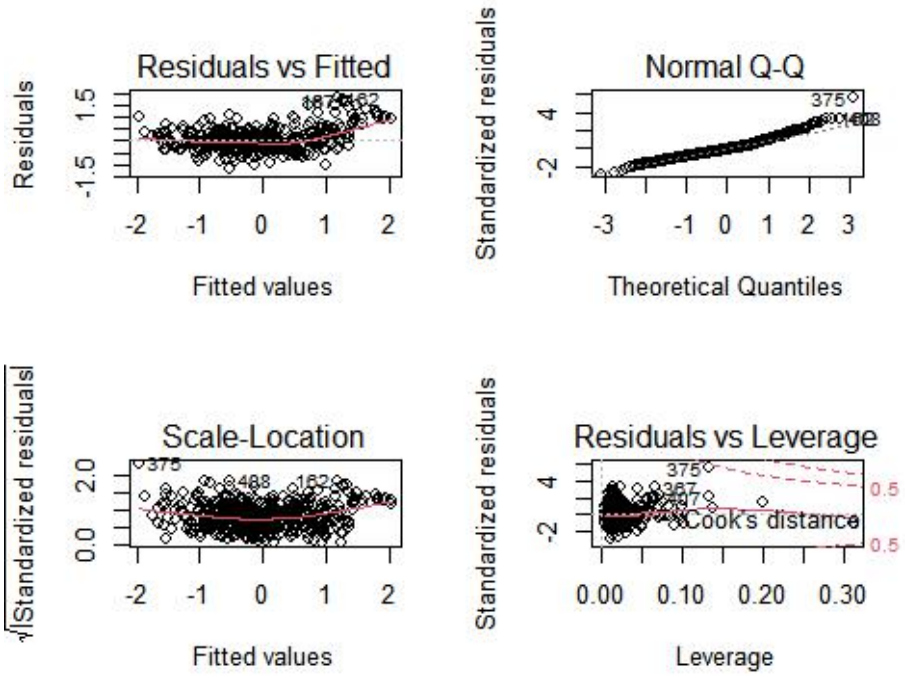
```
result.w.list1[which.max(logLik.list1)] #输出对应对数函数最大值的模型
```

```
## [[1]]
##
## Call:
## lm(formula = medv ~ ., data = data.3, weights = w)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.182538 -0.041426 -0.008642  0.030593  0.296255
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.04633    0.01851  -2.503 0.012659 *
## crim        -0.10199    0.03151  -3.237 0.001290 **
## zn           0.09944    0.02872   3.462 0.000583 ***
## nox         -0.17843    0.03393  -5.259 2.18e-07 ***
## rm           0.35691    0.02793  12.781  < 2e-16 ***
## dis         -0.25712    0.03468  -7.415 5.50e-13 ***
## rad          0.21635    0.04732   4.572 6.13e-06 ***
## tax         -0.23447    0.04749  -4.938 1.09e-06 ***
## ptratio     -0.21746    0.02402  -9.053  < 2e-16 ***
## b            0.12143    0.02115   5.742 1.65e-08 ***
## lstat       -0.23765    0.02984  -7.963 1.21e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06513 on 484 degrees of freedom
```

```
## Multiple R-squared:  0.7878, Adjusted R-squared:  0.7834
## F-statistic: 179.7 on 10 and 484 DF,  p-value: < 2.2e-16

#因为m 的值刚好在取值范围的边界上，故改变参数范围
s = seq(2, 5, 0.5) # 产生数列:2,2.5,3,...,5
logLik.list2 = list() # 新建一个列表，用于储存不同权函数下的对数似然函数值
result.w.list2 = list() # 新建一个列表，用于储存不同权函数下的回归模型结果
for(i in 1 : length(s)){
  w = data.4[,4] ^ (-s[i]) # 计算不同权函数下的权重值
  result.w = lm(medv ~ . ,weights = w,data.3) # 用加权最小二乘拟合线性模型
  logLik.list2[[i]] = logLik(result.w) # 储存对数似然函数值
  result.w.list2[[i]] = summary(result.w) # 储存回归模型结果
}
logLik.list2 # 输出不同权函数下的对数似然函数值
## [[1]]
## 'log Lik.' -253.2886 (df=12)
## [[2]]
## 'log Lik.' -252.1513 (df=12)
## [[3]]
## 'log Lik.' -251.5146 (df=12)
## [[4]]
## 'log Lik.' -251.365 (df=12)
## [[5]]
## 'log Lik.' -251.7038 (df=12)
## [[6]]
## 'log Lik.' -252.5476 (df=12)
## [[7]]
## 'log Lik.' -253.9275 (df=12)

m = 0.5*which.max(logLik.list2) + 1.5 # 计算最优权函数中的参数m
m # 输出m，发现为3.5
## [1] 3.5

result.w.list2[which.max(logLik.list2)] # 输出对应对数函数最大值的模型

## [[1]]
##
## Call:
## lm(formula = medv ~ ., data = data.3, weights = w)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.045543 -0.010228 -0.002037  0.008372  0.083914
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.05911    0.01876  -3.151 0.001728 **
## crim        -0.11354    0.02937  -3.866 0.000126 ***
```

```
## zn           0.11447    0.02930   3.907 0.000107 ***
## nox         -0.19315    0.03224  -5.991 4.08e-09 ***
## rm           0.29104    0.02750  10.584  < 2e-16 ***
## dis         -0.26810    0.03436  -7.803 3.75e-14 ***
## rad          0.24062    0.04500   5.347 1.38e-07 ***
## tax         -0.23936    0.04568  -5.240 2.40e-07 ***
## ptratio     -0.21258    0.02367  -8.983  < 2e-16 ***
## b            0.11970    0.02048   5.846 9.29e-09 ***
## lstat       -0.23148    0.02786  -8.310 9.75e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01638 on 484 degrees of freedom
## Multiple R-squared:  0.7695, Adjusted R-squared:  0.7647
## F-statistic: 161.5 on 10 and 484 DF,  p-value: < 2.2e-16
```

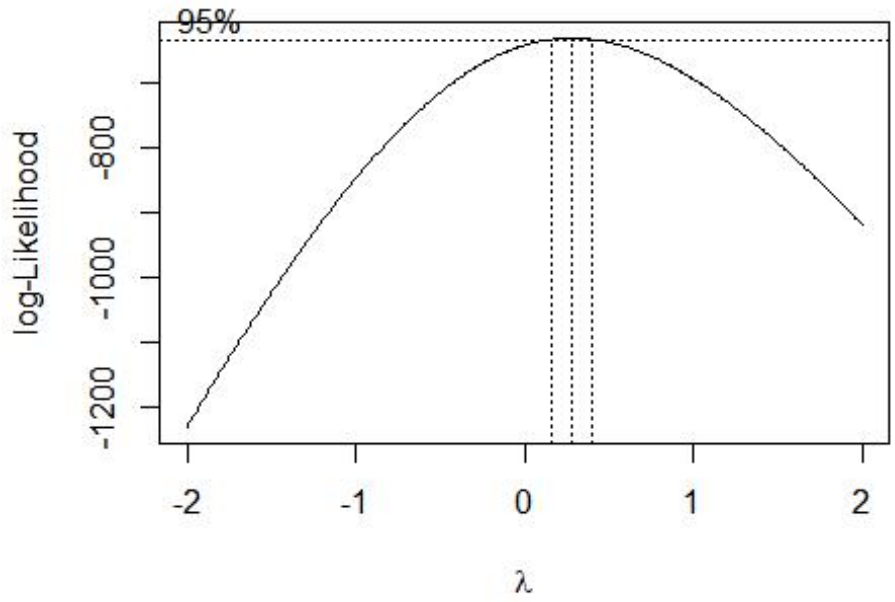
```
# 绘制加权最小二乘诊断图
fit.w = lm(medv~.,weights=data.4[,4]^(-m),data.3) # 将加权最小二乘结果储存在fit.w中
par(mfrow = c(2,2)) # 设置绘图布局
plot(fit.w) # 绘制模型诊断图
```



从上述输出结果可以看到，经过加权最小二乘处理后， 虽然异方差性得到一定改善，但正态性假设仍不满足，且 R 方反而下降至 0.7695，说明拟合效果反而更差，故不选择采用该加权最小二乘模型

2、Box-Cox 变换

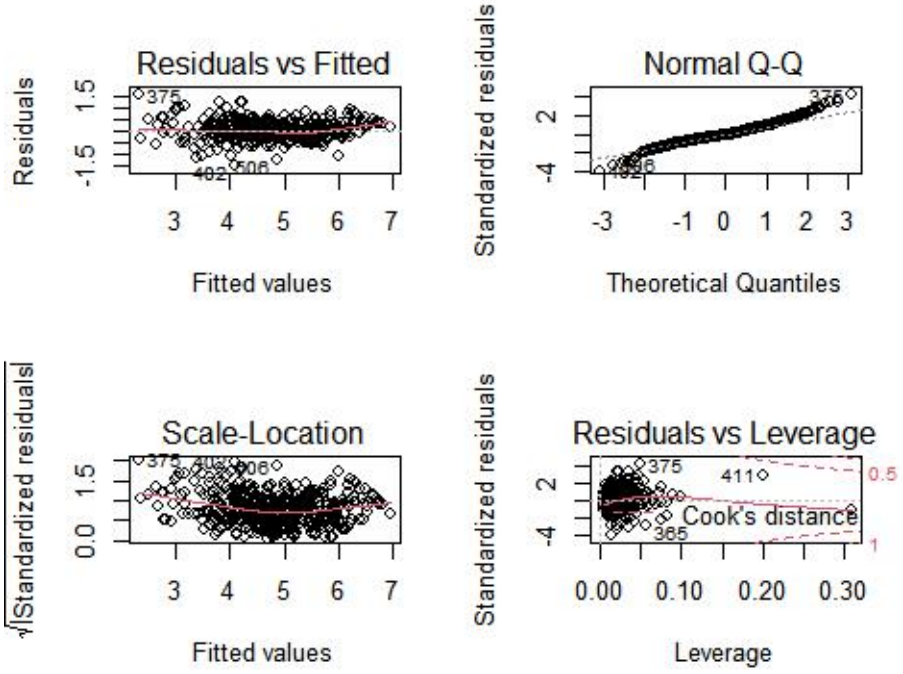
```
# 使响应变量为正
medv = BostonHousing[-abnormal_index,c("medv")] # 提取原因变量 medv
data.5 = cbind(data.3[,c(1:10)],medv) # 合并原因变量 medv

bc.boston = boxcox(medv~., data=data.5, lambda=seq(-2, 2, 0.01))
```




```
#计算不同 Lambda 值对应 BoxCox 变换的似然函数
#Lambda 取值区间为[-2, 2]，步长为 0.01
lambda = bc.boston$x[which.max(bc.boston$y)]
# 选取使似然函数达到最大值的 Lambda 值
lambda # 输出 Lambda，发现为 0.28
## [1] 0.28

medv_bc = (data.5$medv ^ lambda - 1) / lambda
# 计算变换后的 medv 值，记为 medv_bc
fit.3_bc = lm(medv_bc~.-medv,data=data.5) # 以 medv_bc 为因变量拟合模型
summary(fit.3_bc) # 输出拟合结果


##
## Call:
## lm(formula = medv_bc ~ . - medv, data = data.5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.47717 -0.20553 -0.03298  0.20219  1.54535
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.79774    0.01719 279.048  < 2e-16 ***
## crim        -0.21540    0.03193  -6.746 4.35e-11 ***
## zn           0.05051    0.02555   1.977   0.0486 *
## nox         -0.15727    0.03322  -4.735 2.89e-06 ***
## rm           0.29471    0.02548  11.567  < 2e-16 ***
## dis         -0.20667    0.03215  -6.429 3.08e-10 ***
## rad          0.22332    0.04615   4.839 1.76e-06 ***
## tax         -0.22906    0.04566  -5.016 7.40e-07 ***
## ptratio     -0.19900    0.02248  -8.854  < 2e-16 ***
## b            0.11478    0.02032   5.648 2.78e-08 ***
## lstat       -0.33480    0.02983 -11.223  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3809 on 484 degrees of freedom
## Multiple R-squared:  0.8321, Adjusted R-squared:  0.8286
## F-statistic: 239.8 on 10 and 484 DF,  p-value: < 2.2e-16
```

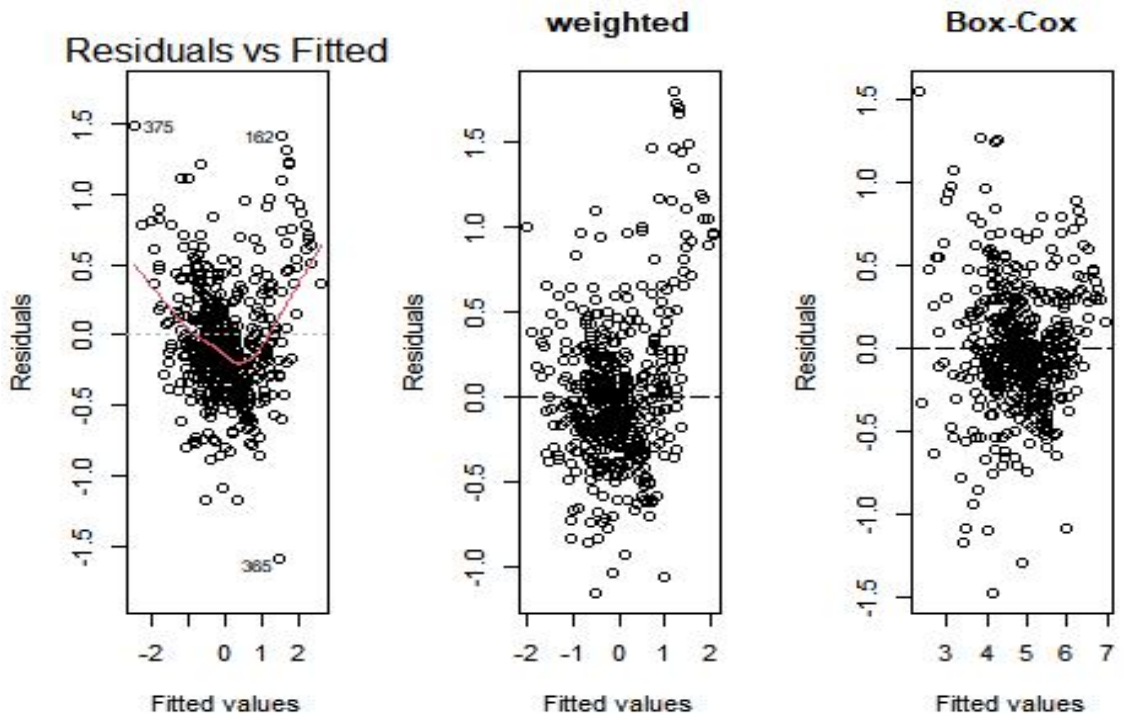
```
par(mfrow = c(2,2)) # 设置绘图布局
plot(fit.3_bc) # 绘制模型诊断图
```



从上述输出结果可以看到，异方差性和正态性都得到一定改善， 且 R 方上升至 0.8321，故我们采用 Box-Cox 变换后的模型

3、残差图对比

```
par(mfrow = c(1,3)) #设置绘图模式
plot(fit.3,which = 1) #普通最小二乘残差图
fit.3_w = result.w.list2[[which.max(logLik.list2)]]
plot(x = as.matrix(cbind(1,data.3[,-c(11)])) %*%
       fit.3_w$coefficients[,1],
     y = data.3$medv - as.matrix(cbind(1,data.3[,-c(11)]))
     %*% fit.3_w$coefficients[,1],
     xlab = "Fitted values",ylab = "Residuals",main="weighted")
      #加权最小二乘残差图
abline(h = c(0),lty = 5) #添加直线y = 0
plot(x = as.matrix(cbind(1,data.5[,-c(11)])) %*%
       fit.3_bc$coefficients,
     y = resid(fit.3_bc),
     xlab = "Fitted values",ylab = "Residuals",main="Box-Cox")
      #进行BOX-COX变换后回归的残差图
abline(h = c(0),lty = 5) #添加直线y = 0
```

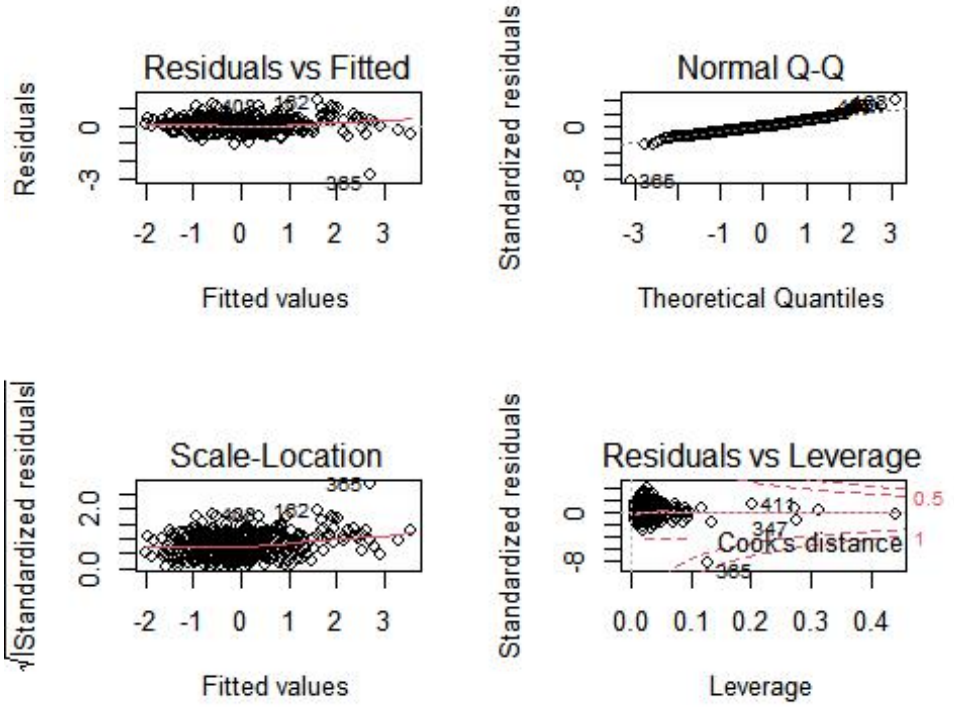


在上述残差图的直接对比中也可以看到， 经过 Box-Cox 变换处理后的残差在 0 上下两侧的分布更加均匀

七、其他改进方向

但是，上述模型在异方差性等方面仍不完美，联想到最初的散点图， 我们发现，因变量 medv 与自变量 rm、lstat 之间可能分别存在 二次项关系和倒数关系，故考虑将这两项纳入模型中

```
fit.4 = lm(medv~.+I(rm^2)+I(1/lstat),data = data.3)
# 加入 rm^2 和 1/lstat 两项后重新拟合模型
summary(fit.4) # 输出拟合结果

##
## Call:
## lm(formula = medv ~ . + I(rm^2) + I(1/lstat), data = data.3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.76928 -0.20518 -0.02475  0.18871  1.39391
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.1716481  0.0189626  -9.052  < 2e-16 ***
## crim        -0.1386785  0.0303758  -4.565 6.34e-06 ***
## zn           0.0689775  0.0241804   2.853 0.004522 **
## nox         -0.1773021  0.0314379  -5.640 2.91e-08 ***
## rm           0.2937532  0.0264542  11.104  < 2e-16 ***
## dis         -0.2142448  0.0307166  -6.975 1.01e-11 ***
## rad          0.1808400  0.0435474   4.153 3.89e-05 ***
## tax         -0.2074135  0.0431345  -4.809 2.04e-06 ***
## ptratio     -0.1837896  0.0214267  -8.578  < 2e-16 ***
## b            0.0716519  0.0195286   3.669 0.000271 ***
## lstat       -0.3266104  0.0288094 -11.337  < 2e-16 ***
## I(rm^2)      0.1386463  0.0107155  12.939  < 2e-16 ***
## I(1/lstat)   0.0009771  0.0004365   2.239 0.025638 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3594 on 482 degrees of freedom
## Multiple R-squared:  0.8625, Adjusted R-squared:  0.859
## F-statistic: 251.9 on 12 and 482 DF,  p-value: < 2.2e-16

par(mfrow = c(2,2)) #设置绘图模式
plot(fit.4) # 绘制诊断图
```

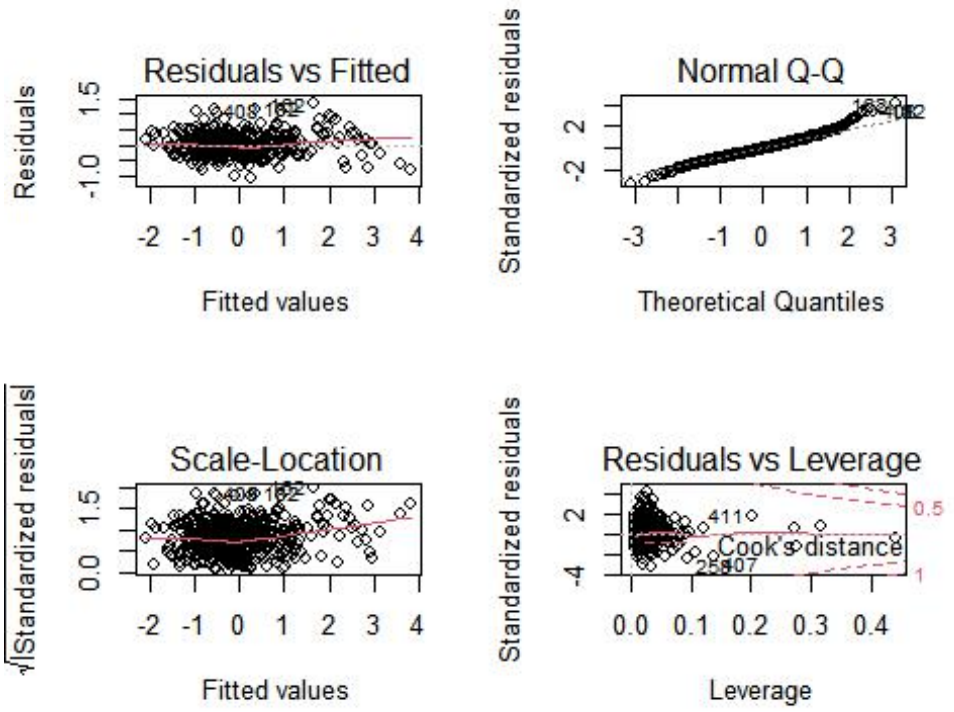


从四个诊断图中都发现，第 365 次观测是一个明显的异常值点，故选择删除

```
fit.5 = lm(medv~.+I(rm^2)+I(1/lstat),data = data.3[-c(365),])
# 删除第 365 次观测
summary(fit.5) # 输出拟合结果
```

```
##
## Call:
## lm(formula = medv ~ . + I(rm^2) + I(1/lstat), data = data.3[-c(365),
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0580 -0.1932 -0.0179  0.1808  1.3317
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.1888694  0.0176975 -10.672  < 2e-16 ***
## crim        -0.1552142  0.0282403  -5.496 6.31e-08 ***
## zn           0.0618209  0.0224461   2.754  0.00611 **
## nox         -0.1580883  0.0292442  -5.406 1.02e-07 ***
## rm           0.3051289  0.0245743  12.417  < 2e-16 ***
## dis         -0.1966887  0.0285635  -6.886 1.80e-11 ***
## rad          0.1800562  0.0403980   4.457 1.03e-05 ***
```

```
## tax         -0.1912553  0.0400561  -4.775 2.39e-06 ***
## ptratio     -0.1644063  0.0199962  -8.222 1.88e-15 ***
## b            0.0711694  0.0181163   3.928 9.80e-05 ***
## lstat       -0.3393566  0.0267642 -12.679  < 2e-16 ***
## I(rm^2)      0.1636538  0.0103306  15.842  < 2e-16 ***
## I(1/lstat)   0.0009684  0.0004049   2.392  0.01715 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3334 on 481 degrees of freedom
## Multiple R-squared:  0.8819, Adjusted R-squared:  0.8789
## F-statistic: 299.3 on 12 and 481 DF,  p-value: < 2.2e-16

par(mfrow = c(2,2)) #设置绘图模式
plot(fit.5) # 绘制诊断图
```



从上述输出结果可以发现，正态性和异方差性得到进一步改善， R 方更是提升到了 0.8819，故我们选择该模型为最终模型。 当然，仍然可以采用 Box-Cox 变换等方式对上述模型做进一步处理，此处就不再加以赘述