Science Discovery
2020; 8(3): 52-63
http://www.sciencepublishinggroup.com/j/sd
doi: 10.11648/j.sd.20200803.12
ISSN: 2331-0642 (Print); ISSN: 2331-0650 (Online)

Linear Regression Model of House Price in Boston

Niu Yilin

School of Mathematics and Statistics, Lanzhou University, Lanzhou, China

Email address:
1727472402@qq.com

To cite this article:
Niu Yilin. Linear Regression Model of House Price in Boston. *Science Discovery.* Vol. 8, No. 3, 2020, pp. 52-63.
doi: 10.11648/j.sd.20200803.12

Received: May 23, 2020; **Accepted**: June 23, 2020; **Published**: June 29, 2020

Abstract: The change of house price is a common phenomenon. People are eager to grasp the law of house price to become the winner of real estate investment. This paper uses Boston house price data to explore the relationship between Boston house price and which independent variables. This paper uses linear regression model to construct the relationship between housing prices and crime rate in Boston. First, the classical linear model is adopted. Then we do the collinearity test, removing the lever point and other operations, the residual of the model still does not conform to the normal distribution, so the classical linear model cannot describe the data very well. Then, we add the quadratic term and the cross term, and use the method of stepwise regression to get the optimal regression autoregressive quantum set. After removing the leverage point and significance test, we found that the residual distribution was approximately normal. It shows that the improved model has well described the law of data. Finally, according to the data, the main conclusions are as follows: house price and tax rate, index close to the highway and index close to the city center are inversely correlated, which is positively correlated with the number of rooms, the proportion of teachers and students, and whether it is close to the Charles River. In addition, the concentration of nitric oxide, the proportion of low-end population and crime rate also have a certain relationship with housing prices.

Keywords: Changes in House Prices, Linear Regression, Stepwise Regression, Positive Correlation, Negative Correlation

波士顿房价的线性回归模型

牛艺霖

兰州大学数学与统计学院，兰州，中国

邮箱
1727472402@qq.com

摘要：房价的变动是一个常见的现象。人们也渴望掌握房价的规律来成为不动产投资的赢家。本文采用波士顿房价的数据，探讨波士顿房价与哪些自变量有关系。本文采用了线性回归模型来构建波士顿房价与犯罪率等因素的关系。首先，经典的线性模型被采用。再做了共线性检验，去除杠杆点等操作之后，模型的残差还是不符合正态分布，因此经典的线性模型并不能很好的刻画数据。随后，我们加入了二次项和交叉项，采用逐步回归的方法得到最优的回归自变量子集。在经过去除杠杆点和显著性检验之后，我们惊喜的发现残差大致呈正态分布。说明改进的模型已经比较好地刻画数据地规律。最后，根据数据得到主要的结论：房价和税率，靠近公路指数以及靠近市中心指数成反相关，与房间个数，师生比例以及是否临近查理斯河呈正相关。此外，一氧化氮浓度，低端人口比例以及犯罪率与房价也有一定的关系。

关键词：房价变动线性，回归逐步回归，正相关，负相关

1．前言

房价一直是人们津津乐道的话题。其中一个重要原因就是房价的价格变动和生活中很多要素相关，如经济发展水平和当地犯罪率。另外，作为一项投资行为，不动产的市场规律也是一个热点话题。本文关注波士顿房价的规律，使用著名的波士顿房价数据集,[1]旨在探讨波士顿不同区域的房价与一些区位要素以及房子本身特点的关系。在这个数据集中，共有503条观测记录，每一条记录代表波士顿某一个区域的平均房价和相关的区位因素。由此，因变量是*波士顿某一个区域的平均房价（PRICE）*，自变量有13个，包括：

CRIM：城镇人均犯罪率。

ZN：住宅用地超过25000 sq.ft.的比例。

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年之前建成的自用房屋比例。

DIS：到波士顿五个中心区域的加权距离。

RAD：辐射性公路的接近指数。

TAX：每 10000 美元的全值财产税率。

PTRATIO：城镇师生比例。

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。

为了简单明了的反映房价和其他区位因素的关系，我们采用线性回归模型以及其衍生模型。线性模型的发展可以追溯到高斯和马可夫的经典模型$Y = X\beta + \varepsilon$，$\varepsilon \sim N(0, \sigma^2 I)$。这个模型在一些比较简单的情形可以清楚地反应自变量和因变量的线性关系但是大多数情况下要求误差ε_i在每个数据点都独立是不现实的。因此，统计学家开始把模型假设放宽：$Y = X\beta + \varepsilon,\ \varepsilon \sim N(0, \Sigma)$，来刻画误差之间的关系。[2]然而，现实中事物的联系往往不仅仅局限在线性关系，更多地是一种非线性关系。幸运的是，线性模型在加入非线性项之后非常容易就划归成了新的线性模型，这为线性模型的推广提供了极大的便利。另一方面，计算回归的算法也在不断发展。[3]人们发现最初的最小二乘法非常容易受到异常点的影响。[4]于是，LAD方法[5]和分位数回归[6，7，8]等方法相继问世。另外，BIC和AIC两个指标的广泛使用解决了平衡参数个数和精度之间的平衡，可以用以计算最优的回归变量子集。

本文的安排如下：首先，在第一部分展示基本的数据统计性描述。接着，在第二部分，我们尝试用最经典的线性回归模型建立模型。基于模型的统计检验和残差分析，我们做出剔除异常点，Box-Cox变换[9]和岭回归[10]等改进。接着，在第三部分，我们考虑加入二次项计算最优回归子集。最后，在第四部分，我们讨论模型带来的结论和问题。

2．数据统计性描述

表1展示了13个自变量和因变量的平均值，标准差，分位数和最值。值得注意的是，这13个自变量中有一个是0-1变量---CHAS，表示是否临近查理斯河。其余的12个自变量的分布都是典型的厚尾分布[11]。而因变量PRICE的分布比较均匀，大致符合正态分布。这为使用回归模型带来了极大的便利，因为线性模型的正态性假设。

表1 某区域房价的自变量和因变量的平均值，标准差，分位数和最值。

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
count	506	506	506	506	506	506	506
mean	3.593761	11.36364	11.13678	0.06917	0.5546	6.284634	68.5749
std	8.596783	23.32245	6.860353	0.253994	0.115878	0.702617	28.14886
min	0.00632	0	0.46	0	0.385	3.561	2.9
25%	0.082045	0	5.19	0	0.449	5.8855	45.025
50%	0.25651	0	9.69	0	0.538	6.2085	77.5
75%	3.647423	12.5	18.1	0	0.624	6.6235	94.075
max	88.9762	100	27.74	1	0.871	8.78	100

表1 继续。

	DIS	RAD	TAX	PIRATIO	B	LSTAT	PRICE
count	506	506	506	506	506	506	506
mean	3.795043	9.549407	408.2372	18.45553	356.674	12.65306	22.53281
std	2.10571	8.707259	168.5371	2.164946	91.29486	7.141062	9.197104
min	1.1296	1	187	12.6	0.32	1.73	5
25%	2.100175	4	279	17.4	375.3775	6.95	17.025
50%	3.20745	5	330	19.05	391.44	11.36	21.2
75%	5.188425	24	666	20.2	396.225	16.955	25
max	12.1265	24	711	22	396.9	37.97	50

图1展示了13个自变量和因变量的两量之间的散点图。计算PRICE与各个因变量的相关系数，因变量PRICE线性相关程度比较强的变量是平均房间数RM和低端人口占比LSTAT. 而且有多组变量的相关系数比较高，这暗示着数据矩阵X 可能有强烈的多重共线性。

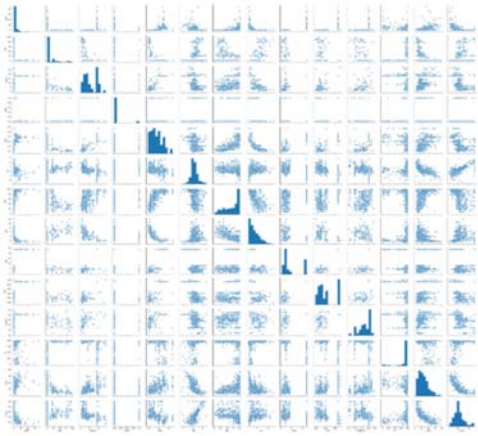


图1 13个自变量和因变量的两量之间的散点图.

3．经典线性回归

考虑经典的线性回归模型，

$$Y = X\beta + \varepsilon,\ \varepsilon \sim N(0, \sigma^2 I)$$

先使用OLS回归，计算得到此时的$\hat{\beta} = X(X^T X)^{-1}Y$。图2展示了使用python statsmodels.api里的OLS包得到的模型概述。可以看出：

$R^2 = 0.741$ 和$R^2_{adjusted} = 0.734$可以看出模型不是可解释性不是非常强，只能解释数据变异73.4%的信息

D-W 统计量等于1.078，不趋于2，意味着残差并不是独立的。

```
In [165]: X=A.iloc[:,:-1]
          Y=A['PRICE']
          X = sm.add_constant(X) # adding a constant

          model = sm.OLS(Y, X).fit()
          predictions = model.predict(X)

          print_model = model.summary()
          print(print_model)
```

OLS Regression Results

Dep. Variable:	PRICE	R-squared:	0.741
Model:	OLS	Adj. R-squared:	0.734
Method:	Least Squares	F-statistic:	108.1
Date:	Thu, 02 Apr 2020	Prob (F-statistic):	6.95e-135
Time:	17:37:32	Log-Likelihood:	-1498.8
No. Observations:	506	AIC:	3026.
Df Residuals:	492	BIC:	3085.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	36.4911	5.104	7.149	0.000	26.462	46.520
CRIM	-0.1072	0.033	-3.276	0.001	-0.171	-0.043
ZN	0.0464	0.014	3.380	0.001	0.019	0.073
INDUS	0.0209	0.061	0.339	0.735	-0.100	0.142
CHAS	2.6886	0.862	3.120	0.002	0.996	4.381
NOX	-17.7958	3.821	-4.658	0.000	-25.302	-10.289
RM	3.8048	0.418	9.102	0.000	2.983	4.626
AGE	0.0008	0.013	0.057	0.955	-0.025	0.027
DIS	-1.4758	0.199	-7.398	0.000	-1.868	-1.084
RAD	0.3057	0.066	4.608	0.000	0.175	0.436
TAX	-0.0123	0.004	-3.278	0.001	-0.020	-0.005
PTRATIO	-0.9535	0.131	-7.287	0.000	-1.211	-0.696
B	0.0094	0.003	3.500	0.001	0.004	0.015
LSTAT	-0.5255	0.051	-10.366	0.000	-0.625	-0.426

Omnibus:	178.029	Durbin-Watson:	1.078
Prob(Omnibus):	0.000	Jarque-Bera (JB):	782.015
Skew:	1.521	Prob(JB):	1.54e-170
Kurtosis:	8.276	Cond. No.	1.51e+04

图2 使用OLS包得到的模型概述。

JB统计量很大，P（JB）很小，这说明残差并不是正态分布的。

自变量AGE 和INDUS 的p值很大，说明这两个变量的系数可以是0（这需要做F检验，看两个系数是否能同时为0。

4．模型诊断与改进

4.1．检验共线性与岭回归的比较

首先，我们检验模型的共线性。依次计算每个变量的VIF。从图3.b我们可以看出VIF最大值是7.21没有超过10，说明因变量之间没有很强的共线性。我们在这里保留所有的变量。但是，自变量中最高的VIF达到7.210，不是很强的共线性可能对结果有影响。考虑到这一点，我们对该数据集做岭回归以避免共线性的影响。图4.a展示了岭迹图，最优$\lambda = 1$。图4.b对比了最优λ值的参数估计和最小二乘估计。可以看出，与最小二乘估计相比，除了RM和AGE这两个变量之外，其余的系数缩小。但系数的改变并不明显，这也验证了共线性的影响不是很强。

```
TAX        7.210320
RAD        6.332047
DIS        2.828715
NOX        2.629255
LSTAT      2.368614
ZN         2.235368
CRIM       1.762718
B          1.226203
PTRATIO    1.133744
CHAS       1.059419
RM         0.884926
dtype: float64
```

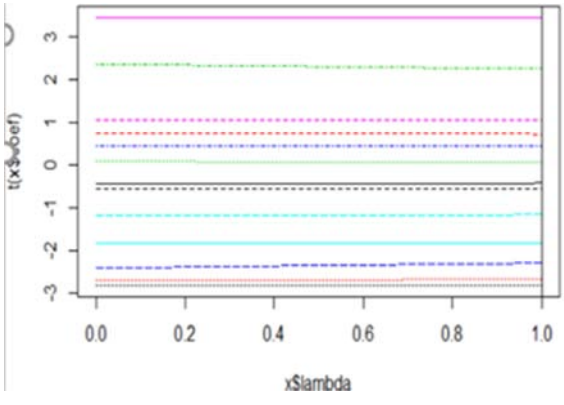
(a)

```
In [173]:
def VIF(data1):
    col=list(data1.columns)
    VIF_store=pd.Series([1.0]*len(col),index=col)
    for i in col:
        #print(col)
        data1_dropped=(data1.drop([i],axis="columns")).copy()
        SSE,SST,H=SSE_and_SST(data1_dropped,data1[i])
        r_2=(SST-SSE)/SST
        VIF_store[i]=1/(1-r_2)
    VIF=VIF_store.sort_values(ascending=False)
    print(VIF)
    if(VIF[0]>7):
        col_dropped=VIF.index[0]
        return True,col_dropped,(data1.drop([VIF.index[0]],axis="columns")).copy()
    else:
        print("THe max VIF is ",VIF[0])
        return False,"",data1
#yn,col_dropped,data1=VIF(A_reduced)
def literation_to_drop_by_VIF(data1,target,significance_level):
    yn,col_dropped,data_dropped=VIF(data1)
    print(data1.head(2),data_dropped.head(2))
    if(yn):
        X_r= sm.add_constant(data_dropped)
        X_f=sm.add_constant(data1)
        df=X_f.shape[0]-X_f.shape[1]
        SSE_r,SST_r,H_r=SSE_and_SST(X_r,target)
        SSE_f,SST_f,H_f=SSE_and_SST(X_f,target)
        if(F_test_reduce(SSE_f,SSE_r,1,df,significance_level)):
            #print("00000000\n")
            literation_to_drop_by_VIF(data_dropped,target,significance_level)

        else:
            return data1
    else:
        #print("bbb",data1)
        return data1
A_new=literation_to_drop_by_VIF((A_reduced.iloc[:,:-1]).copy(),target,0.05)
A_new["PRICE"]=A_reduced["PRICE"]
```

(b)

图3 （a）求VIF值的代码,（b）每个变量的VIF值。



(a)

```
> lm.ridge(PRICE~.,data=d,lambda=1)
                    CRIM           ZN        INDUS         CHAS          NOX           RM          AGE
 20.85244980  -0.07852136   0.03107167   0.00710557   2.00759423 -10.77590421   5.35027445  -0.02040578
         DIS          RAD          TAX      PTRATIO            B        LSTAT
 -1.33337100   0.25932295  -0.01393641  -0.86307571   0.01214596  -0.42533947
> lm(PRICE~.,data=d)

Call:
lm(formula = PRICE ~ ., data = d)

Coefficients:
(Intercept)         CRIM           ZN        INDUS         CHAS          NOX           RM          AGE
   21.20113     -0.08090      0.03180      0.01255      1.97335    -10.99169      5.34393     -0.02034
        DIS          RAD          TAX      PTRATIO            B        LSTAT
   -1.34553      0.27242     -0.01464     -0.86565      0.01218     -0.42602

>
```

(b)

图4 （a）数据集的岭迹图，(b) 最优λ值的参数估计和最小二乘估计。

4.2．检测异常观测

4.2.1．首先，我们需要去除模型中的杠杆点

380	0.303508
418	0.190031
405	0.155130
410	0.124730
365	0.098574
155	0.085319
490	0.082051
367	0.080428
492	0.077108
364	0.076686
491	0.076460
152	0.076135
489	0.075708
353	0.075579
488	0.074213
414	0.074114
214	0.073353
126	0.069988
142	0.069306
123	0.067948
368	0.066380
156	0.065870
163	0.064243
283	0.062858
154	0.061313
120	0.060986
124	0.060922
122	0.060618
125	0.060371
121	0.060134
	...
230	0.009825
327	0.009785
324	0.009773
323	0.009739
36	0.009687
301	0.009068
215	0.009056
235	0.008944
272	0.008937
312	0.008584

图5 每个变量的杠杆值。

根据H矩阵算出每个变量的杠杆值，图5可以看出相对于平均值（2*12/503=0.044）确实有很多大于0.1的杠杆点。

4.2.2．去除了杠杆点，我们考虑学生化残差来找出异常值

图6展示了Bonferroni双侧检验的临界值|-3.90|和所有超出临界值的数据点，可以看出第336和第335个数据点是异常值。对于这些异常值，我们无法判定是否是由于数据输入的输入错误还是自然发生的。所以我们暂时保留它们。

```
> d=read.csv("D:/Boston.csv",head=T)
> PRICE.lm=lm(PRICE~.,data=d)
> stud=rstudent(PRICE.lm)
> stud[which.max(abs(stud))]
     336
7.108767
> qt(0.05/(456*2),443)
[1] -3.903311
> stud(which(abs(stud)>3.903))
Error in stud(which(abs(stud) > 3.903)) : 没有"stud"这个函数
> stud[abs(stud)>3.903]
     335      336
6.961932 7.108767
```

图6 Bonferroni双侧检验的临界值和所有超出临界值的数据点。

4.2.3．最后我们考虑有影响的观测点

图7计算了cook统计量并从大到小排序，结果显示第335，334，337，139，131和142观测点的cook统计量比较大。我们将这些点去除。

```
In [28]: import pandas as pd
d=pd.read_csv("D:\Boston.csv")
fit=sm.formula.ols("PRICE~CRIM+ZN+INDUS+CHAS+NOX+RM+AGE+DIS+RAD+TAX+PTRATIO+B+LSTAT",data=d).fit()
outliers=fit.get_influence()
cook=outliers.cooks_distance[0]
cook=pd.Series(cook)
cook.sort_values(ascending=False)

Out[28]: 335    2.679718e-01
334    9.925955e-02
337    6.526455e-02
139    4.258937e-02
131    4.237777e-02
142    3.891719e-02
338    3.109538e-02
371    2.884962e-02
162    1.791818e-02
333    1.744932e-02
336    1.744446e-02
137    1.613875e-02
61     1.581450e-02
367    1.564850e-02
366    1.539618e-02
202    1.504983e-02
362    1.331127e-02
199    1.325439e-02
314    1.242505e-02
207    1.141169e-02
455    1.099208e-02
346    1.081277e-02
```

图7 计算cook统计量并从大到小排序。

将所有的不正常的点全部去除，我们重新做OLS回归。图8展示了重新做最小二乘回归的结果。可以看出去掉这些点之后的模型R^2和

$R^2_{adjusted}$比之前的模型提高了不少，AIC和BIC也有明显的提高。

OLS Regression Results

Dep. Variable:	PRICE	R-squared:	0.783
Model:	OLS	Adj. R-squared:	0.777
Method:	Least Squares	F-statistic:	122.9
Date:	Thu, 02 Apr 2020	Prob (F-statistic):	1.30e-137
Time:	17:52:53	Log-Likelihood:	-1287.4
No. Observations:	456	AIC:	2603.
Df Residuals:	442	BIC:	2661.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	21.2011	5.187	4.088	0.000	11.007	31.395
CRIM	-0.0809	0.065	-1.251	0.212	-0.208	0.046
ZN	0.0318	0.013	2.472	0.014	0.007	0.057
INDUS	0.0126	0.067	0.186	0.852	-0.120	0.145
CHAS	1.9733	0.916	2.155	0.032	0.174	3.773
NOX	-10.9917	4.103	-2.679	0.008	-19.055	-2.929
RM	5.3439	0.448	11.926	0.000	4.463	6.225
AGE	-0.0203	0.013	-1.606	0.109	-0.045	0.005
DIS	-1.3455	0.189	-7.119	0.000	-1.717	-0.974
RAD	0.2724	0.077	3.523	0.000	0.120	0.424
TAX	-0.0146	0.004	-3.285	0.001	-0.023	-0.006
PTRATIO	-0.8657	0.122	-7.093	0.000	-1.105	-0.626
B	0.0122	0.003	4.627	0.000	0.007	0.017
LSTAT	-0.4260	0.059	-7.201	0.000	-0.542	-0.310

Omnibus:	195.412	Durbin-Watson:	1.117
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1460.062
Skew:	1.673	Prob(JB):	0.00
Kurtosis:	11.103	Cond. No.	1.66e+04

图8 重新做最小二乘回归的结果.

最后，我们重新发现去掉杠杆点的模型中,p值大于0.05的自变量有三个：CRIM，INDUS和AGE。我们需要做F检验，看这个三个自变量前系数是否能够同时为0. 在图9中可以看出，事实上，假设通过检验，可以同时去掉这三个变量。这个结果也印证了模型不具有较强的共线性。

```
In [169]: #test for the validity of removing some variables from the previous model
SSE_reduced, SST_reduced, H_reduced=SSE_and_SST(X_reduced, target)
SSE_full, SST_full, H=SSE_and_SST(X, target)

def F_test_reduce(SSE_full, SSE_reduced, m, df, significance_level):#m, n-k-1
    F=((SSE_reduced-SSE_full)/m)/(SSE_full/df)
    if(F>stats.f.isf(significance_level, m, df)):
        return False
    else:
        return True
F_test_reduce(SSE_full, SSE_reduced, 2, 494, 0.05)

SSR/SST=  [[0.74054535]]
SSR/SST=  [[0.74060774]]

Out[169]: True
```

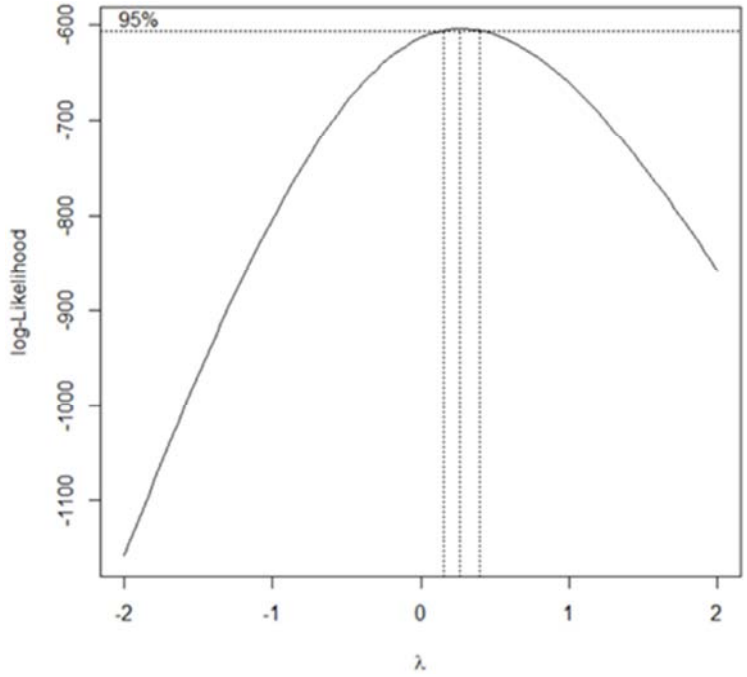
图9 对p值大于0.05的三个自变量做F检验。

4.3．Box-cox变换

从因变量PRICE的直方图可以看出，PRICE与正态分布有些相似,且PRICE>0. 我们考虑用Box-cox变换，即令 $y^* = \begin{cases} \frac{y^\lambda - 1}{\lambda} \ if \ \lambda \neq 0 \\ lny \ if \ \lambda = 0 \end{cases}$. 图10.a展示了对数似然随$\lambda$的变化图。可以看到$\lambda$的95%置信区间在[0.1，0.25]左右，为了便于解释，我们取λ=0.25。即因变量变为$PRICE^* = \sqrt[4]{PRICE}$ 我

们对$PRICE^*$做OLS回归，图10.b展示了回归后的结果。可　以看出模型此时的R^2已经提高到了0.79. 目前的模型为；

$$\sqrt[4]{PRICE} = 2.1642 + 0.0897 * RM - 0.0140 * DIS + 0.0038 * RAD - 0.003 * TAX - 0.019 * PTRATIO + 0.0003 * B - 0.016 * LASTT$$



(a)

```
> data_reduced=read.csv("D:/Boston_reduced_changed.csv",head=T)
> summary(lm(PRICE~.,data=data_reduced))

Call:
lm(formula = PRICE ~ ., data = data_reduced)

Residuals:
     Min       1Q   Median       3Q      Max
-0.37011 -0.04906 -0.00425  0.04268  0.52198

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.162e+00  1.036e-01  20.869  < 2e-16 ***
RM           8.970e-02  9.607e-03   9.337  < 2e-16 ***
DIS         -1.407e-02  2.818e-03  -4.994 8.50e-07 ***
RAD          3.829e-03  1.548e-03   2.474 0.013737 *
TAX         -3.241e-04  8.374e-05  -3.870 0.000125 ***
PTRATIO     -1.889e-02  2.561e-03  -7.375 7.99e-13 ***
B            3.144e-04  6.003e-05   5.237 2.51e-07 ***
LSTAT       -1.575e-02  1.160e-03 -13.576  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09515 on 448 degrees of freedom
Multiple R-squared:  0.7916,    Adjusted R-squared:  0.7883
F-statistic: 243.1 on 7 and 448 DF,  p-value: < 2.2e-16
```

(b)

图10 （a） 对数似然随$\boldsymbol{\lambda}$的变化图,(b) 对$\boldsymbol{PRICE^*}$做OLS回归后的结果。

4.4．残差分析

接着，我们来进行残差分析。图11.a画出了标准化的残差-因变量均值散点图。可以看出，残差并不对称地分布在直线两侧。再结合残差的QQ图和直方图（图11.c），我们可以判定残差并不服从正态分布。

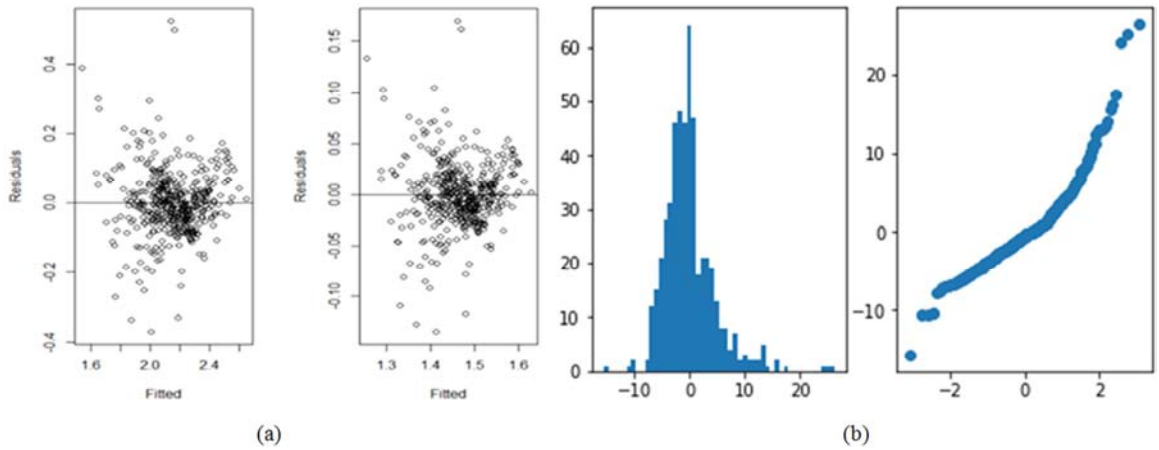


图11 （a）标准化的残差-因变量均值散点图,（b）平方根变换之后的残差图,（c）残差的QQ图和直方图。

4.5．判定是否为异方差和非线性

我们对预测值和残差做一阶线性回归，看是否存在线性关系。从p值可以看出线性趋势并不显著。但这只能说明线性趋势不明显。图12.b展示了D-W统计量的值，p值小于2.2*e-16,这是序列相关性强的证据。但是D-W统计量为0.98，表明一阶的线性关系并不显著可能存在高阶的关系。

```
Call:
lm(formula = sqrt(abs(residuals(PRICE.lm))) ~ fitted(PRICE.lm))

Residuals:
    Min      1Q  Median      3Q     Max
-1.4630 -0.5734 -0.0688  0.4661  3.8316

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.373260   0.110515  12.426   <2e-16 ***
fitted(PRICE.lm) 0.007364   0.004616   1.595    0.111
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7563 on 454 degrees of freedom
Multiple R-squared:  0.005575,  Adjusted R-squared:  0.003384
F-statistic: 2.545 on 1 and 454 DF,  p-value: 0.1113

> dwtest(PRICE~.,data=d)

        Durbin-Watson test

data:  PRICE ~ .
DW = 0.97752, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

图12 （a）展示D-W统计量的值的代码,（b）D-W统计量的值。

4.6．方差稳定变换

基于以上的分析，ε_i的方差可能是异方差且非线性。考虑方差稳定变换，图11.b是平方根变换之后的残差图。从图11.a和图11.b的对比中可以看到方差还是没有达到稳定，残差仍然未对称地分布在直线的两侧。这说明平方根变换并没有太大的作用，问题的根源还是在于自变量没有二次项

5．模型的改进

由于之前的分析可以看出，对于波士顿房价的数据，经典的线性模型确实得到令人满意的模型。于是，我们先考虑加入全部二次项和交叉项，再用逐步回归筛查出最优的的回归变量子集.故目前的模型是

$$y_i^* = \sum_{k\le j} \alpha_{kj} x_{ik} x_{ij} + \sum_{l=1}^{p} \beta_l x_{il} + \varepsilon_i, \varepsilon = 1,2 \dots p$$

逐步回归算法简介如下[12]

计算第零步增广矩阵。第零步增广矩阵是由预测因子和预测对象两两之间的相关系数构成的。

引进因子。在增广矩阵的基础上，计算每个因子的方差贡献，挑选出没有进入方程的因子中方差贡献最大者对应的因子，计算该因子的方差比，查F分布表确定该因子是否引入方程。

剔除因子。计算此时方程中已经引入的因子的方差贡献，挑选出方差贡献最小的因子,计算该因子的方差比，查F分布表确定该因子是否从方程中剔除。

矩阵变换。将第零步矩阵按照引入方程的因子序号进行矩阵变换，变换后的矩阵再次进行引进因子和剔除因子的步骤，直到无因子可以引进，也无因子可以剔除为止，终止逐步回归分析计算

图13展示了逐步回归的代码和结果。注意到模型参数的增加R^2增加但$R^2_{adjusted}$也有很大的提高，达到了0.860，而且两者十分接近，说明多余的变量很少。另外，此时的D-W统计量比之前更接近2，说明此时的残差更接近独立。更可喜的是，AIC和BIC明显减少，说明精度的增加大大压过参数增加的惩罚。

接着我们筛查一遍模型中的异常点（杠杆点），按照上述模型重新建模，发现筛查杠杆点后的模型中又出现了5项的t值大于2（见图14）：$PRATIO, B, LSTAT^2, TAX * PRATIO$ 和 $B * LSTAT$。我们做F检验，检验通过，将这五个变量全部剔除。得到最终模型。

$$\sqrt[4]{PRICE} = -43.0051 + 34.7164 * CHAS + 17.7707 * RM - 0.0972 * TAX + 4.0136 * LSTAT - 0.0507 * DIS^2 - 0.1082 * RAD^2 + 2.1578 * CRIM * CHAS - 7.2173 * CHAS * RM - 0.6689 * RM * LSTAT + 0.0058 * RAD * TAX + 0.0040 * TAX * PTRATIO - 0{,}0011 * TAX * LSTAT - 0.0141 * CRIM * LSTAT - 30.9793 * CHAS * RAD - 0.6003 * CHAS * LSTAT - 0.9181 * NOX * RAD - 0.3182 * RM * PTRATIO$$

图15.b展示了最终结果的总结表。剔除了杠杆点之后$R^2_{adjusted}$增加至0.868。AIC和BIC大幅下降，D-W统计量也接近1.6，表明自变量更加接近独立。图15.a展示了最终模型的残差图.残差大致均匀的分布在0的两侧且标准化残差集中分布在[-2,2]的带型区域。图15.b展示了QQ图和直方图，也反映了此时的残差大致服从正态分布[13]。综上所述，我们的最终模型要优于之前的经典模型。

```
In [214]: def stageWise(xArr, yArr, step=0.01, numIt=100):
              xMat = mat(xArr)
              xMat = regularize(xMat)
              yMat = mat(yArr).T
              yMean = mean(yMat)
              yMat = yMat - yMean
              N, n = shape(xMat)
              returnMat = zeros((numIt, n))
              ws = zeros((n, 1))
              wsTest = ws.copy()
              weMax = ws.copy()
              for ii in range(numIt):
                  print(ws.T)
                  lowestErr = inf
                  for jj in range(n):
                      for sign in [-1, 1]:
                          wsTest = ws.copy()
                          wsTest[jj] += step * sign
                          yTest = xMat * wsTest
                          rssE = rssError(yMat.A, yTest.A)
                          if rssE < lowestErr:
                              lowestErr = rssE
                              wsMax = wsTest
                  ws = wsMax.copy()
                  returnMat[ii, :] = ws.T
              return returnMat
          OLS_summary(stagewise(X,Y,0.001,91))
```

(a)

OLS Regression Results

Dep. Variable:	PRICE	R-squared:	0.866
Model:	OLS	Adj. R-squared:	0.860
Method:	Least Squares	F-statistic:	142.2
Date:	Thu, 02 Apr 2020	Prob (F-statistic):	5.84e-195
Time:	17:37:33	Log-Likelihood:	-1331.3
No. Observations:	506	AIC:	2709.
Df Residuals:	483	BIC:	2806.
Df Model:	22		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-68.3759	17.903	-3.819	0.000	-103.554	-33.198
CHAS	26.8036	4.592	5.837	0.000	17.781	35.826
RM	20.9205	1.924	10.872	0.000	17.140	24.702
TAX	-0.0976	0.025	-3.866	0.000	-0.147	-0.048
PTRATIO	2.6367	0.953	2.766	0.006	0.764	4.510
B	0.0492	0.012	4.087	0.000	0.026	0.073
LSTAT	1.6139	0.445	3.630	0.000	0.740	2.488
CRIM*CHAS	1.8472	0.333	5.550	0.000	1.193	2.501
CHAS*CHAS	26.8036	4.592	5.837	0.000	17.781	35.826
CHAS*RM	-5.2952	1.108	-4.778	0.000	-7.473	-3.118
RM*LSTAT	-0.3158	0.046	-6.876	0.000	-0.406	-0.226
DIS*DIS	-0.0465	0.009	-5.213	0.000	-0.064	-0.029
RAD*RAD	-0.1043	0.026	-4.053	0.000	-0.155	-0.054
RAD*TAX	0.0052	0.001	4.583	0.000	0.003	0.007
LSTAT*LSTAT	0.0197	0.004	4.457	0.000	0.011	0.028
TAX*PTRATIO	0.0045	0.001	3.476	0.001	0.002	0.007
TAX*LSTAT	-0.0014	0.000	-7.161	0.000	-0.002	-0.001
CRIM*LSTAT	-0.0052	0.001	-4.469	0.000	-0.008	-0.003
CHAS*NOX	-28.2264	5.918	-4.770	0.000	-39.855	-16.598
CHAS*LSTAT	-0.3514	0.146	-2.413	0.016	-0.638	-0.065
NOX*RAD	-0.5831	0.202	-2.894	0.004	-0.979	-0.187
RM*PTRATIO	-0.7482	0.109	-6.856	0.000	-0.963	-0.534
B*B	-6.593e-05	2.16e-05	-3.058	0.002	-0.000	-2.36e-05
B*LSTAT	-0.0009	0.000	-2.805	0.005	-0.002	-0.000

Omnibus:	185.816	Durbin-Watson:	1.388
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1409.549
Skew:	1.395	Prob(JB):	8.32e-307
Kurtosis:	10.686	Cond. No.	4.77e+19

(b)

图13 （a）逐步回归的代码,（b）逐步回归的结果。

OLS Regression Results

Dep. Variable:	PRICE	R-squared:	0.873
Model:	OLS	Adj. R-squared:	0.868
Method:	Least Squares	F-statistic:	163.4
Date:	Thu, 02 Apr 2020	Prob (F-statistic):	5.48e-169
Time:	18:06:18	Log-Likelihood:	-1072.3
No. Observations:	422	AIC:	2181.
Df Residuals:	404	BIC:	2253.
Df Model:	17		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-43.0051	4.821	-8.920	0.000	-52.483	-33.527
CHAS	34.7164	4.320	8.036	0.000	26.224	43.209
RM	17.7707	0.889	19.988	0.000	16.023	19.518
TAX	-0.0972	0.019	-5.145	0.000	-0.134	-0.060
LSTAT	4.0136	0.346	11.598	0.000	3.333	4.694
CRIM*CHAS	2.1578	0.313	6.902	0.000	1.543	2.772
CHAS*CHAS	34.7164	4.320	8.036	0.000	26.224	43.209
CHAS*RM	-7.2173	1.050	-6.872	0.000	-9.282	-5.153
RM*LSTAT	-0.6689	0.051	-13.215	0.000	-0.768	-0.569
DIS*DIS	-0.0507	0.010	-4.968	0.000	-0.071	-0.031
RAD*RAD	-0.1082	0.027	-3.952	0.000	-0.162	-0.054
RAD*TAX	0.0058	0.001	4.688	0.000	0.003	0.008
TAX*PTRATIO	0.0040	0.001	4.174	0.000	0.002	0.006
TAX*LSTAT	-0.0011	0.000	-3.516	0.000	-0.002	-0.000
CRIM*LSTAT	-0.0141	0.005	-3.112	0.002	-0.023	-0.005
CHAS*NOX	-30.9792	5.493	-5.640	0.000	-41.777	-20.182
CHAS*LSTAT	-0.6003	0.137	-4.374	0.000	-0.870	-0.331
NOX*RAD	-0.9181	0.237	-3.878	0.000	-1.384	-0.453
RM*PTRATIO	-0.3128	0.049	-6.333	0.000	-0.410	-0.216

Omnibus:	177.029	Durbin-Watson:	1.529
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2409.031
Skew:	1.403	Prob(JB):	0.00
Kurtosis:	14.364	Cond. No.	1.98e+18

图14 筛查一遍模型中的异常点（杠杆点）重新建模。

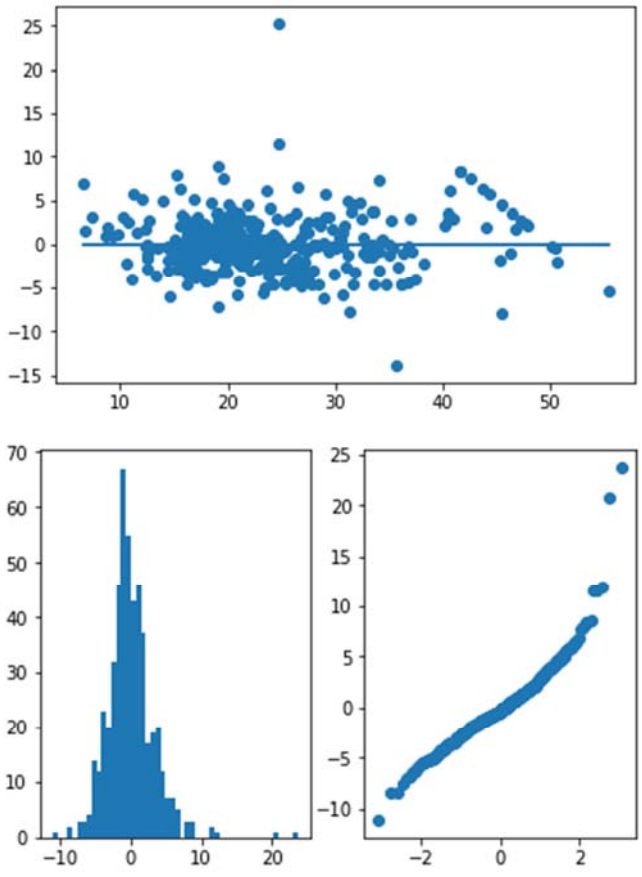


图15 （a）最终模型的残差图,(b)最终模型的结果的QQ图和直方图。

6．结果与讨论

结论：

波士顿各地区的房价PRICE 与 税率TAX 成反比，与住宅平均房价数RM和该地区师生比例PTRATIO 成正比。与0-1变量，是否临近查尔斯河CHAS，成正比。

波士顿各地区的房价PRICE与 接近公路指数RAD的平方和接近街区指数DIS的平方成反比。

其他3个变量：犯罪率CRIM，低端人口比例LSTAT和一氧化氮浓度NOX，以交叉影响的方式与之前提到的6个变量影响房价。

房价 和 大面积房子占比 ZN， 旧房子比例 AGE，非裔美国人比例指数B和非零售土地比例 INDUS无关。

讨论：

值得注意的是模型中有一个0-1变量 CHAS。而在最终的模型中，含有CHAS的交叉项有四项：CHAS*RM，CHAS*CRIM，CHAS*NOX 和CHAS*PTRATIO. 也就是说这些交叉项的含义是 在查尔斯湖畔的RM，CRIM，NOX 和PRATIO指数(如果不临近查尔斯湖畔，那么就等于0)，相当于CHAS这个要素突出了RM，CRIM，NOX 和PRATIO指数是否发生跳跃。这些交叉项组成了新的有特殊意义的项，使得0-1变量不再难以在控制。

另外，有4个变量未出现在模型中，至于它们是否与模型无关，可以用多元方差分析[14]去检验。或者用pearson检验[15]其与PRICE的独立性。

参考文献

[1] Applied Sciences，波士顿房价数据集[Z]。CSDN.2018.

[2] 王桂松，陈敏，陈立萍，线性回归模型[M]。北京：高等教育出版社 2017：235页.

[3] 周新辉,李昱喆,李富有,新冠疫情对中小服务型企业影响评估及对策研究[J]。经济评论，2020，第三期：102-120页.

[4] 李占宏,韩春红,王泽来,在线振动管液体密度计静态压力修正试验分析[J]。实验研究，2020，第六期：29-32页.

[5] 陈璋鑫,宋玉梅,万群, LAD准则下的无线传感器网络节点定位方法[J]。电子科技大学学报，2009，第38卷第一期：43-36页.

[6] 陈晋玲, 基于分位数回归的人力资本结构对产业结构优化升级的影响研究[J]。商业经济，2020，第六期：33-40页.

[7] 江勇杰，自动选择可变系数的分位数回归[D]。西南财经大学，2014.

[8] 李顺毅，房价如何影响消费对经济增长的贡献—基于分位数回归的实证分析[J]。消费经济，2011，03:3－6 +10.

[9] 张继超,朴建也,吴睿,宋新,基于Box-Cox变换的双层联合模型[J]。东北师大学报(自然科学版）,2019,第51卷第4期：36-41页。

[10] 路晨，沈勇，单泽，基于岭回归算法自动优化既有铁路曲线的研究[J]。云南民族大学学报(自然科学版)，2020，第29卷第2期：285-291页。

[11] 黄一凡，孟生旺，基于厚尾分布的非寿险准备金评估模型[J]。系统工程理论与实践，2020，第40卷第1期：42-54页。

[12] 陈慈，张敬磊，王云，盖姣云，基于逐步回归分析和ＢＰ__Ａｄａｂｏｏｓｔ算法的危险驾驶行为辨识[J]。数学的实践与认识，2019，第49卷第14期：200-207页.

[13] 陈子亮,卿清,影响波士顿不同社区房价水平的因素分析[M]。中央财经大学：商界论坛/产业经济，2015：278-279页.

[14] 阿荣高娃,孙根年,乔少辉,王翠平,内蒙古A级景区客流量估算模型-5个单因素方差分析与多元回归建模[J]。干旱区资源与环境，2019，第33卷第12期：193-200页. 胡添翼1，杨光1，陈 波1， 2，俞 扬1，陶 园1。

[15] 胡添翼.杨光,陈波,俞扬,陶园,基于Pearson相关性检验的ARIMA坡位移监测模型[J]。水利水电技术2016,第47卷第1期:71-75页。