HW4 Challenge (Regularization)

Written by 郑丽珊&宋悦溪

本文基于 China General Soical Survey (CGSS)数据库，利用 2015 年中国综合社会调查的数据选取“个人对政府是否应该干预生育孩子的态度”作为被解释变量，并选取 50 个相关的解释变量，试图探究个人对政府是否应该干预人们的生育计划的态度的影响因素。

一、 选题意义

新中国成立以来，我国的生育政策在实践中不断调整、完善。19 世纪 70 年代，我国政府以强硬的态度执行“计划生育”，通过法律的影响直接决定了人们的生育计划；2015 年，十八届五中全会公布“二胎政策”，以温和的方式从外部环境来促进人口发展。而作为法律的服从者，人们是否愿意接受政府对生育计划的干涉？若不愿意，又是哪部分的人群会反对政府的干涉？深入研究人们对政府干预行为的态度的影响因素不仅具有重要的学术价值，而且对正确认识政府在生育政策上的态度转变，为今后我国生育政策调整、所提供的生育环境提供更为切合实际的决策支持具有重大意义，同时政府可以根据支持人群的特征情况制定相关的决策，如教育、社会公平、社会保障、地方经济等方面营造社会氛围，为人口发展提供良好的外部环境。

本文利用 2015 年中国综合社会调查数据，以个人对政府干预的看法为被解释变量，并选取 50 个相关的解释变量，探究人们对政府干预的看法的影响因素。

二、 变量解释

经过删除缺失值，我们最终得到 2817 个观测值。

变量名称	变量含义	变量分类情况
Y	您是否同意以下说法-生多少孩子是个人的事，政府不应该干涉	同意赋值为 1，不同意赋值为 0
X1	受访者来源类型	城市赋值为 1，农村赋值为 0
A	采访地点的地级市编码	共计 89 个地级市编码（1-89），设置 88 个虚拟变量
X3	性别	男性赋值为 0，女性赋值为 1
X4	年龄（根据出省年份进行推算）	连续变量
X5	民族	汉族赋值为 0，少数民族赋值为 1
X6	是否有宗教信仰	是赋值为 1，否赋值为 0
B	教育程度	共计 3 个类别：高中及以下、大学、研究生及以上，设定 2 个虚拟变量
X8	去年全年收入	连续变量
X9	是否递交过入党申请书	是赋值为 1，否赋值为 0
C	政治面貌	共计 4 个类别：群众、共青团员、中共党员、民主党派，设定 3 个虚拟变量
X11	现在住的这座住房的套内建筑面积	连续变量
X12	目前的身体健康状况	健康赋值为 1，不健康赋值为 0
D	过去一年社交频率	共计 5 个类别：从不、很少、有时、经常、非常频繁，设定 4 个虚拟变量
X14	是否感到幸福	是赋值为 1，否赋值为 0
X15	如果没有政策限制希望生几个孩子	连续变量
X16	目前处于哪个等级	连续变量
X17	您认为您 10 年后将会在哪个等级上	连续变量
X18	在您 14 岁时，您的家庭处在哪个等级上	连续变量
X19	是否购买城市基本医疗保险/新型农村合作医疗保险/公费医疗	是赋值为 1，否赋值为 0
X20	是否购买城市/农村基本养老保险	是赋值为 1，否赋值为 0
X21	是否购买商业性医疗保险	是赋值为 1，否赋值为 0
X22	是否购买商业性养老保险	是赋值为 1，否赋值为 0
X23	去年全年家庭总收入	连续变量
X24	目前住在一起的通常有几人（包括本人）	连续变量
E	您家的家庭经济状况在所在地属于哪一档	共计 5 个类别：远低于平均水平、低于平均水平、平均水平、高于平均水平、远高于平均水平，设定 4 个虚拟变量
X26	拥有几处住房	连续变量
X27	是否拥有家用汽车	是赋值为 1，否赋值为 0
X28	是否从事投资活动	是赋值为 1，否赋值为 0
F	婚姻状况	共计 4 个类别：初婚 1 配偶、再婚 1 配

		偶、同居、分居未离婚，设定 3 个虚拟变量
G	与同龄人相比，您本人的社会经济地位怎样	共计 3 个类别：差于同龄人、差不多、优于同龄人，设定 2 个虚拟变量
X31	您认为您的年收入达到多少元，您才会比较满意	连续变量
X32	是否赞同：只要孩子够努力、够聪明，都能有同样的升学机会	是赋值为 1，否赋值为 0
X33	是否赞同：在我们这个社会，工人和农民的后代与其他人的后代一样，拥有同样多的机会	是赋值为 1，否赋值为 0
X34	对于公共教育服务的总体满意度	连续变量
X35	对于医疗卫生公共服务的总体满意度	连续变量
X36	对于基本住房保障公共服务的总体满意度	连续变量
X37	对于社会管理公共服务的总体满意度	连续变量
X38	对下列公共服务其他各领域的满意度-劳动就业	连续变量
X39	对下列公共服务其他各领域的满意度-社会保障	连续变量
X40	对下列公共服务其他各领域的满意度-低保，灾害，流浪乞讨，残疾，孤儿救助等	连续变量
X41	对下列公共服务其他各领域的满意度-公共文化与体育	连续变量
X42	对下列公共服务其他各领域的满意度-城乡基础设施	连续变量
X43	配偶或同居伴侣去年全年的收入	连续变量
H	配偶或同居伴侣教育程度	共计 3 个类别：高中及以下、大学、研究生及以上，设定 2 个虚拟变量
X45	父亲教育程度	大学赋值为 1，高中及以下赋值为 0
I	父亲政治面貌	共计 4 个类别：群众、共青团员、中共党员、民主党派，设定 3 个虚拟变量
J	您 14 岁时父亲的就业状况	共计 6 个类别：去世、丧失劳动力、离退休、无业；全职务农；半农半工；受雇于人；个体；自由职业者，设定 5 个虚拟变量
X48	母亲教育程度	大学赋值为 1，高中及以下赋值为 0
K	母亲政治面貌	共计 4 个类别：群众、共青团员、中共党员、民主党派，设定 3 个虚拟变量
L	您 14 岁时母亲的就业状况	共计 6 个类别：去世、丧失劳动力、离退休、无业；全职务农；半农半工；受雇于人；个体；自由职业者，设定 5 个虚拟变量

三、 模型结果

1. Logistic Regression: Full Model

从回归结果来看，显著的解释变量有 X14（当前生活是否感到幸福）、X24（目前包括本人通常有几人一起居住）、X48（是否有从事投资活动）、A（被采访者来源地点）和 B（本人受教育程度）。

2. Logistic Regression: Lasso

Lasso 模型结果显示，被保留的解释变量有 X18（14 岁家庭处于的等级）、X24（目前包括本人通常有几人一起居住）、X36（对于基本住房保障公共服务的总体满意度）、X48（母亲受教育程度）、A（被采访者来源地点）和 F（当前婚姻状况）。

3. KNN

KNN 模型结果显示，最优的 neighbour 个数应选取 16-18 个。

四、 模型比较

模型	Logistic Regression: Full Model	Lasso	KNN
失误率	0.456352	0.4655784	0.4471256

通过比较可以看出，KNN 的失误率最低，但无解释性；相对于 Lasso，logistic 的失误率较低。

附件：

代码及结果展示：

1. Logistic Regression: Full Model

（1）fit on training data

```
###################################
# Logistic Regression: Full Model #
###################################
# Fit on training data
logitfit <- glm(Y ~.,dat_new.tr,family='binomial')
summary(logitfit)
```

```
Call:
glm(formula = Y ~ ., family = "binomial", data = dat_new.tr)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3755  -1.0529  -0.5236   1.1033   2.2027

Coefficients: (2 not defined because of singularities)
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.136e+00  1.935e+00   0.587  0.55719
X1          -1.118e-01  1.566e-01  -0.714  0.47506
X3           7.398e-02  1.279e-01   0.578  0.56298
X4          -2.052e-03  5.826e-03  -0.352  0.72466
X5          -5.045e-02  3.398e-01  -0.148  0.88197
X6           1.078e-01  2.125e-01   0.507  0.61192
X8           1.238e-06  1.115e-06   1.110  0.26704
X9           1.765e-01  3.151e-01   0.560  0.57546
X11         -6.541e-04  7.294e-04  -0.897  0.36987
X12          2.861e-02  1.723e-01   0.166  0.86813
X14         -5.077e-01  2.427e-01  -2.092  0.03645 *
X15          7.027e-02  7.057e-02   0.996  0.31940
X16          2.900e-02  6.659e-02   0.435  0.66322
X17          5.366e-03  5.248e-02   0.102  0.91855
X18          6.597e-02  4.161e-02   1.585  0.11292
X19         -2.348e-01  2.687e-01  -0.874  0.38216
X20          2.101e-01  1.578e-01   1.331  0.18304
X21         -3.028e-01  2.951e-01  -1.026  0.30486
X22          3.375e-01  3.227e-01   1.046  0.29569
X23          4.714e-07  5.039e-07   0.936  0.34949
X24          9.036e-02  4.936e-02   1.831  0.06714 .
X26          1.487e-01  1.120e-01   1.327  0.18440
X27         -2.699e-01  1.657e-01  -1.629  0.10341
X28         -4.983e-01  1.177e+00  -0.423  0.67207
X31         -1.011e-08  2.977e-08  -0.339  0.73428
```

X32	7.199e-02	1.923e-01	0.374	0.70814
X33	-1.195e-01	1.511e-01	-0.791	0.42882
X34	3.268e-03	5.242e-03	0.623	0.53298
X35	4.309e-03	5.223e-03	0.825	0.40941
X36	-6.432e-03	4.957e-03	-1.298	0.19444
X37	-3.710e-03	6.200e-03	-0.598	0.54955
X38	5.789e-03	5.817e-03	0.995	0.31961
X39	-5.526e-03	6.561e-03	-0.842	0.39960
X40	6.416e-04	4.637e-03	0.138	0.88994
X41	3.968e-04	5.624e-03	0.071	0.94376
X42	8.591e-04	5.462e-03	0.157	0.87502
X43	2.128e-07	4.517e-07	0.471	0.63754
X45	-3.755e-01	4.775e-01	-0.786	0.43170
X48	1.944e+00	9.054e-01	2.148	0.03175 *
A1	-2.058e+00	9.723e-01	-2.116	0.03430 *
A10	-1.306e+00	9.751e-01	-1.339	0.18048
A11	-1.309e+00	9.659e-01	-1.356	0.17520
A12	-2.628e+00	1.090e+00	-2.410	0.01597 *
A13	-3.315e+00	1.136e+00	-2.919	0.00351 **
A14	-1.726e+00	1.022e+00	-1.689	0.09115 .
A15	-1.535e+00	1.649e+00	-0.931	0.35197
A16	-2.133e+00	1.303e+00	-1.638	0.10150
A17	-3.897e+00	1.344e+00	-2.900	0.00373 **
A18	-1.663e+00	8.848e-01	-1.880	0.06015 .
A19	-8.734e-01	1.059e+00	-0.825	0.40965
A2	-2.610e+00	1.136e+00	-2.298	0.02154 *
A20	-3.026e+00	1.153e+00	-2.625	0.00866 **
A21	-1.922e+00	9.471e-01	-2.030	0.04237 *
A22	-2.286e+00	9.682e-01	-2.361	0.01822 *
A23	-1.806e+00	1.016e+00	-1.778	0.07548 .
A24	-2.371e+00	9.315e-01	-2.546	0.01091 *
A25	-1.980e+00	9.092e-01	-2.178	0.02944 *
A26	-2.089e+00	9.979e-01	-2.094	0.03629 *
A27	-1.920e+00	8.866e-01	-2.166	0.03035 *
A28	-2.196e+00	9.747e-01	-2.253	0.02425 *
A29	-2.083e+00	1.066e+00	-1.954	0.05071 .
A3	-1.604e+00	1.009e+00	-1.589	0.11202
A30	-1.806e+00	9.580e-01	-1.886	0.05936 .
A31	-6.191e-01	1.059e+00	-0.585	0.55877
A32	-1.763e+00	9.410e-01	-1.873	0.06101 .
A33	-1.730e+01	8.827e+02	-0.020	0.98437
A34	-9.438e-01	1.119e+00	-0.843	0.39899
A35	-1.730e+00	1.346e+00	-1.285	0.19880
A36	-3.844e+00	1.199e+00	-3.207	0.00134 **

```
A37          1.260e+01  4.981e+02   0.025  0.97982
A39         -5.820e-01  1.167e+00  -0.499  0.61804
A4          -2.912e+00  1.199e+00  -2.428  0.01518 *
A40         -2.051e+00  9.715e-01  -2.111  0.03480 *
A41         -1.871e+00  9.472e-01  -1.975  0.04825 *
A42         -1.822e+00  1.023e+00  -1.782  0.07477 .
A43         -2.031e+00  9.465e-01  -2.146  0.03186 *
A44         -2.030e+00  9.733e-01  -2.086  0.03701 *
A45         -2.607e+00  9.863e-01  -2.643  0.00821 **
A46         -2.283e+00  9.350e-01  -2.442  0.01461 *
A47         -1.614e+01  6.181e+02  -0.026  0.97917
A48         -1.025e+00  1.095e+00  -0.936  0.34945
A49         -1.935e+00  1.094e+00  -1.769  0.07684 .
A5          -1.824e+00  1.681e+00  -1.085  0.27803
A50         -3.405e+00  1.147e+00  -2.969  0.00299 **
A51         -1.204e+00  9.286e-01  -1.296  0.19481
A52         -1.726e+00  8.806e-01  -1.960  0.04996 *
A53         -1.931e+00  1.003e+00  -1.924  0.05429 .
A54         -2.409e+00  9.504e-01  -2.535  0.01125 *
A55         -1.541e+00  9.009e-01  -1.711  0.08712 .
A56         -2.616e+00  1.011e+00  -2.587  0.00969 **
A57         -2.303e+00  9.305e-01  -2.475  0.01334 *
A59         -2.834e+00  1.147e+00  -2.470  0.01351 *
A6          -2.373e+00  9.140e-01  -2.596  0.00943 **
A61         -1.040e+00  1.184e+00  -0.879  0.37958
A62         -2.437e+00  1.085e+00  -2.245  0.02474 *
A63         -1.712e+00  1.061e+00  -1.613  0.10673
A64         -2.351e+00  8.500e-01  -2.766  0.00567 **
A65         -2.554e+00  9.113e-01  -2.802  0.00508 **
A66         -3.027e+00  9.579e-01  -3.160  0.00158 **
A67         -3.317e+00  1.069e+00  -3.102  0.00192 **
A68         -2.003e-01  9.982e-01  -0.201  0.84098
A69         -1.306e+00  9.016e-01  -1.449  0.14747
A7          -1.728e+00  8.627e-01  -2.003  0.04515 *
A70         -2.740e+00  1.001e+00  -2.736  0.00622 **
A71         -2.364e+00  1.052e+00  -2.247  0.02465 *
A72         -2.658e+00  1.016e+00  -2.617  0.00887 **
A74         -1.928e+00  1.086e+00  -1.775  0.07594 .
A75         -2.663e+00  1.140e+00  -2.336  0.01951 *
A76         -2.648e+00  1.177e+00  -2.249  0.02453 *
A77         -1.652e+00  9.601e-01  -1.721  0.08524 .
A78         -2.215e+00  1.018e+00  -2.177  0.02946 *
A79         -1.412e+00  9.699e-01  -1.456  0.14544
A8          -2.117e+00  8.797e-01  -2.406  0.01613 *
```

A80	-2.875e+00	9.176e-01	-3.134	0.00173	**
A81	-2.511e+00	9.438e-01	-2.661	0.00780	**
A82	-2.636e+00	9.643e-01	-2.734	0.00626	**
A83	-2.593e+00	9.325e-01	-2.781	0.00542	**
A84	-1.363e+00	9.875e-01	-1.380	0.16765	
A85	-2.019e+00	9.133e-01	-2.210	0.02709	*
A86	4.267e-02	1.386e+00	0.031	0.97545	
A87	-1.986e+00	8.750e-01	-2.270	0.02320	*
A88	-2.068e+00	9.417e-01	-2.196	0.02807	*
A89	-3.932e+00	1.348e+00	-2.918	0.00353	**
A9	NA	NA	NA	NA	
B2	-3.761e-01	2.281e-01	-1.649	0.09924	.
B3	-2.179e-01	7.710e-01	-0.283	0.77745	
C1	5.792e-02	4.005e-01	0.145	0.88501	
C2	-4.801e-02	3.504e-01	-0.137	0.89103	
C3	-1.448e+01	6.239e+02	-0.023	0.98148	
D1	-6.751e-02	2.249e-01	-0.300	0.76405	
D2	-1.394e-01	2.328e-01	-0.599	0.54925	
D3	-9.436e-02	2.372e-01	-0.398	0.69081	
D4	-2.338e-01	3.444e-01	-0.679	0.49728	
`E-2`	2.487e-01	3.228e-01	0.770	0.44101	
E0	1.749e-01	1.569e-01	1.114	0.26512	
E1	1.066e-01	2.666e-01	0.400	0.68919	
E2	-1.961e+00	1.323e+00	-1.482	0.13832	
F0	1.638e+00	1.834e+00	0.893	0.37191	
F1	5.926e-01	1.706e+00	0.347	0.72839	
F2	8.563e-02	1.757e+00	0.049	0.96112	
G0	-5.379e-02	1.591e-01	-0.338	0.73529	
G1	8.579e-02	3.116e-01	0.275	0.78306	
H2	-2.872e-01	2.295e-01	-1.252	0.21070	
H3	-5.559e-01	7.302e-01	-0.761	0.44646	
I1	-4.863e-01	1.305e+00	-0.373	0.70945	
I2	1.807e-01	2.134e-01	0.847	0.39718	
J1	-1.337e-01	2.878e-01	-0.464	0.64230	
J2	-3.746e-01	4.632e-01	-0.809	0.41872	
J3	-1.697e-02	2.925e-01	-0.058	0.95373	
J4	-4.593e-01	4.720e-01	-0.973	0.33042	
J5	-1.259e+00	9.992e-01	-1.260	0.20757	
K1	1.567e+01	8.827e+02	0.018	0.98584	
K2	4.236e-01	4.943e-01	0.857	0.39152	
K3	NA	NA	NA	NA	
L1	6.139e-02	2.050e-01	0.299	0.76464	
L2	5.930e-01	8.234e-01	0.720	0.47140	
L3	3.057e-01	2.589e-01	1.181	0.23760	

```
L4          5.128e-01  4.855e-01   1.056  0.29087
L5          2.454e-01  1.653e+00   0.148  0.88198
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1949.5  on 1407  degrees of freedom
Residual deviance: 1755.4  on 1251  degrees of freedom
AIC: 2069.4

Number of Fisher Scoring iterations: 13
```

(2)Predict on test data

```
# Predict on test data
p = predict(logitfit,dat_new.te,type="response")
logitpred = as.factor(p > 0.5)
table(logitpred,dat_new.te$Y,dnn=c("predicted","true"))
logiterr <- 1-mean(logitpred==dat_new.te$Y) #misclassification error rate
logiterr
```

结果:

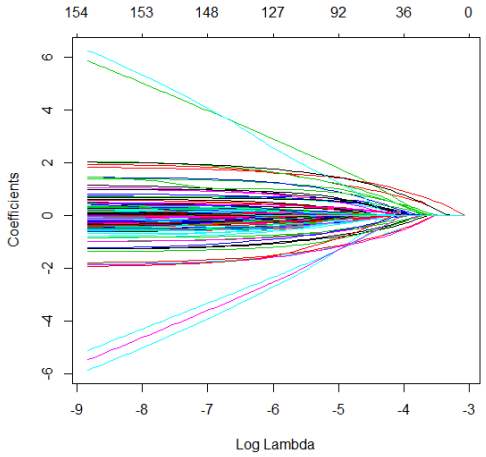
```
          true
predicted FALSE TRUE
    FALSE   444  345
    TRUE    298  32

> logiterr
[1] 0.456352
```

2. Logistic Regression: Lasso

(1) Fit on training data

```
x <- model.matrix(Y ~.,dat_new.tr)[,-1] #no intercept
y <- dat_new.tr$Y
lassofit.all <- glmnet(x,y,alpha=1,family="binomial")
plot(lassofit.all,xvar="lambda")
```

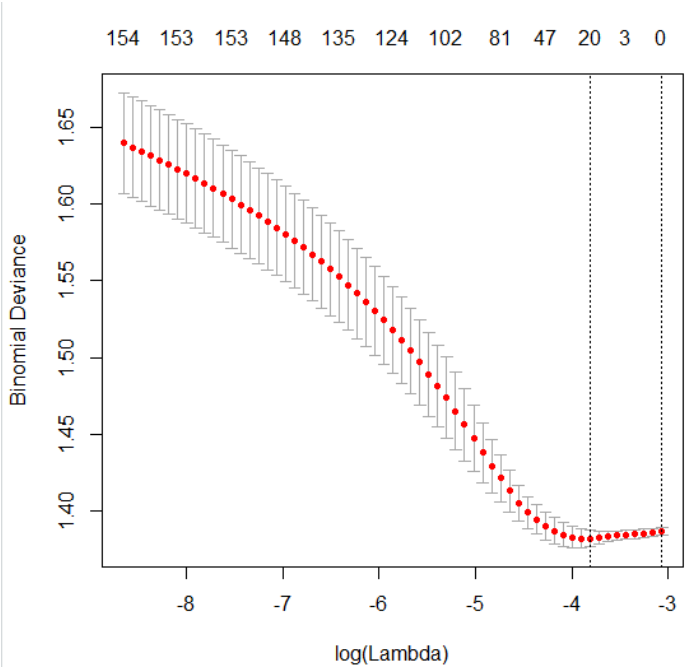
结果:



(2) Cross validation

```
# Cross Validation
cv.lasso <- cv.glmnet(x,y,alpha=1,family="binomi
plot(cv.lasso)
```

结果:



(3) Refit the model using optimal lamda

```
# Refit the model using optimal lambda
lambda.star <- cv.lasso$lambda.min #alternatively, use the 1se rule: lambda.star <- cv.lasso$lambda.1s
lassofit.star <- glmnet(x,y,alpha=1,lambda=lambda.star,family="binomial")
coef(lassofit.star)
```

结果:

```
159 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept) -0.106914227
X1           .
X3           .
X4           .
```

X5	.
X6	.
X8	.
X9	.
X11	.
X12	.
X14	.
X15	.
X16	.
X17	.
X18	0.007773932
X19	.
X20	.
X21	.
X22	.
X23	.
X24	0.019793746
X26	.
X27	.
X28	.
X31	.
X32	.
X33	.
X34	.
X35	.
X36	-0.001199064
X37	.
X38	.
X39	.
X40	.
X41	.
X42	.
X43	.
X45	.
X48	0.542944835
A1	.
A10	0.026562916
A11	.
A12	.
A13	-0.151466000
A14	.
A15	.
A16	.
A17	-0.317723209

A18	.
A19	.
A2	.
A20	.
A21	.
A22	.
A23	.
A24	.
A25	.
A26	.
A27	.
A28	.
A29	.
A3	.
A30	.
A31	0.194699549
A32	.
A33	.
A34	.
A35	.
A36	-0.318558231
A37	0.159963975
A39	.
A4	.
A40	.
A41	.
A42	.
A43	.
A44	.
A45	.
A46	.
A47	.
A48	.
A49	.
A5	.
A50	-0.053847725
A51	.
A52	.
A53	.
A54	.
A55	.
A56	.
A57	.
A59	.

A6	.
A61	.
A62	.
A63	.
A64	.
A65	.
A66	-0.040942685
A67	-0.057072814
A68	0.775456642
A69	0.136878189
A7	.
A70	.
A71	.
A72	.
A74	.
A75	.
A76	.
A77	.
A78	.
A79	.
A8	.
A80	-0.069104969
A81	.
A82	.
A83	.
A84	.
A85	.
A86	0.089479350
A87	.
A88	.
A89	-0.353104543
A9	0.583787964
B2	.
B3	.
C1	.
C2	.
C3	.
D1	.
D2	.
D3	.
D4	.
`E-2`	.
E0	.
E1	.

```
E2          .
F0          0.249417784
F1          .
F2          .
G0          .
G1          .
H2          .
H3          .
I1          .
I2          .
J1          .
J2          .
J3          .
J4          .
J5          .
K1          .
K2          .
K3          .
L1          .
L2          .
L3          .
L4          .
L5          .
```

(4) Prdict on test data

```
# Predict on test data
newx <- model.matrix(Y ~.,dat_new.te)[,-1]
lassopred <- predict(lassofit.star,newx,type="class")
table(lassopred,dat_new.te$Y,dnn=c("predicted","true"))
lassoerr <- 1-mean(lassopred==dat_new.te$Y)
lassoerr
```

结果：

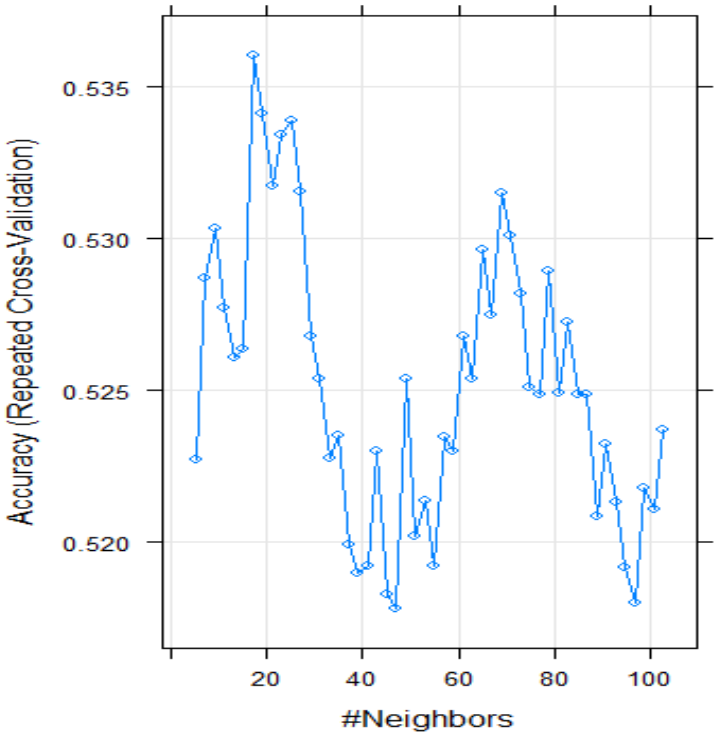
```
> table(lassopred,dat_new.te$Y,dnn=c("predicted","true"))
         true
predicted FALSE TRUE
    FALSE   693  607
    TRUE     49   60
> lassoerr
[1] 0.4655784
```

3.Knn

(1)fit on training data

```
# Fit on training data
knnfit <- train(Y ~.,data=dat_new.tr,method="knn",              #require("caret")
                trControl=trainControl(method="repeatedcv",repeats=3), #use repeated CV to choose K
                preProcess=c("center","scale"),tuneLength = 50)        #center and scale predictors before KNN
plot(knnfit)
```

结果：



(2)predict on test data

```
# Predict on test data
knnpred <- predict(knnfit,dat_new.te)
table(knnpred,dat_new.te$Y,dnn=c("predicted","true"))
knnerr <- 1-mean(knnpred==dat_new.te$Y)
knnerr
```

结果：

```
> table(knnpred,dat_new.te$Y,dnn=c("predicted","true"))
        true
predicted FALSE TRUE
    FALSE   482  370
    TRUE    260  297
> knnerr
[1] 0.4471256
```