Given a highly correlated dataset (generated by an unknown model).
How to select a possible good working model?

First we can load the data and see the correlation matrix:

(1) write.csv(mydata, file = "/Users/Penny/Desktop/study /Fall 2016/stat bigdata/hw/mydata.csv")
(2) sample correlation between the response (V1) and each of the covariates

```
> cor(mydata)[1,]
       V1                V2                V3                V4                V5                V6
 1.00000000 -0.35631353 -0.04243960 -0.04011802  0.43695150 -0.46365281
>
```

the correlation matrix :

```
> cor(mydata)
           V1          V2          V3          V4          V5          V6
V1  1.00000000 -0.35631353 -0.04243960 -0.04011802  0.43695150 -0.46365281
V2 -0.35631353  1.00000000 -0.05026962 -0.06689517  0.36128863  0.97089447
V3 -0.04243960 -0.05026962  1.00000000  0.88357693 -0.05573148 -0.05789913
V4 -0.04011802 -0.06689517  0.88357693  1.00000000 -0.06154981 -0.07853427
V5  0.43695150  0.36128863 -0.05573148 -0.06154981  1.00000000  0.33051479
V6 -0.46365281  0.97089447 -0.05789913 -0.07853427  0.33051479  1.00000000
```

scatter plot: V2, V6 are highly correlated; V3, V4 are highly correlated. V2, V3, V6 are not linear to V1

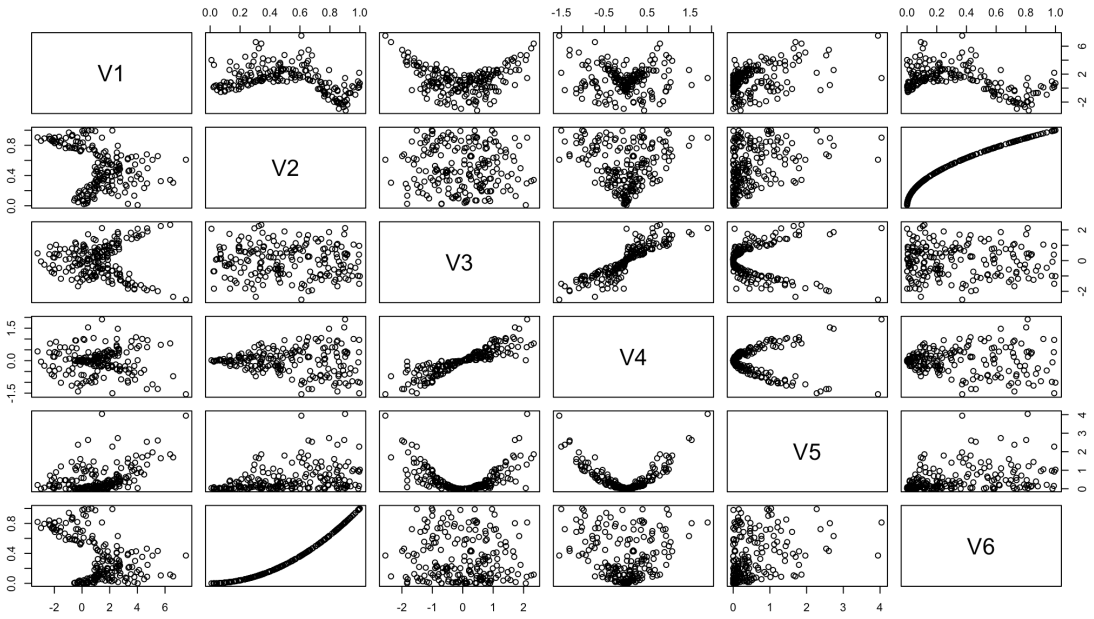


Figure 1 scatter plots

To avoid over-fitting, using cross validation by selecting and validate the model.
(3)fit the model by training data set (80% of the original data)
initial model : $V1 = \beta_1 + \beta_2 * V2 + \beta_3 * V3 + \beta_4 * V4 + \beta_5 * V5 + \beta_6 * V6$
Test $H0$: $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 0$

HA: $not\ all\ of\ the\ coefficents$ ($\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6$) are zeros.

The fitted model is:

$$V1 = 0.07165 + 8.81774 * V2 + 0.26085 * V3 - 0.69342 * V4 + 1.59246 * V5 - 12.48026 * V6 \quad (1)$$

```
> fit.initial<-glm(V1~.,data = mydata.train)
> summary(fit.initial)

Call:
glm(formula = V1 ~ ., data = mydata.train)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-3.1742  -0.6960  -0.0690   0.6134   3.4048

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.07165    0.29824   0.240   0.8105
V2           8.81774    1.31529   6.704 3.62e-10 ***
V3           0.26085    0.17752   1.469   0.1438
V4          -0.69342    0.30659  -2.262   0.0251 *
V5           1.59246    0.12365  12.878  < 2e-16 ***
V6         -12.48026    1.21726 -10.253  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 1.109352)

    Null deviance: 567.41  on 159  degrees of freedom
Residual deviance: 170.84  on 154  degrees of freedom
AIC: 478.55

Number of Fisher Scoring iterations: 2
```

model checking:

for the training data: model is not suitable, the assumption of i.i.d normal error is violated.

(a) residual plot: the residuals are concentrated when the fitted values are between 0 and 3. The residuals have bigger value when the value of fitted y value increase. It is not randomly nor evenly distributed around 0.

(b)QQ plot: the residuals departure away from normal distribution.

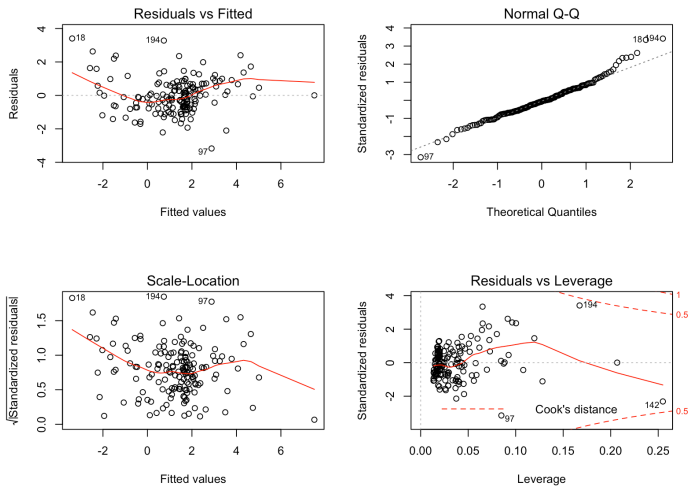


Figure 2 residual plots for initial fitting for the training data set

Fit the test data: most points have test errors greater then -2.The residuals are not evenly distributed since as ytest.hat increase, residual value decrease. The QQ-plot shows that test errors do not follow a normal distribution.

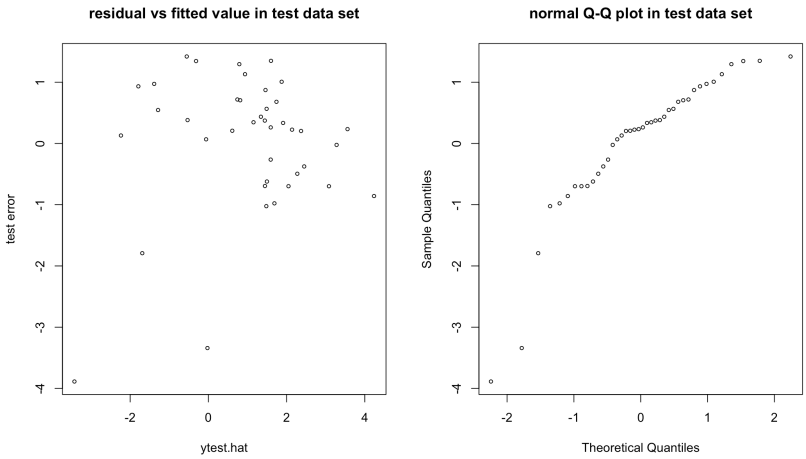


Figure 3 residual plots for testing data set for the initial model

(4)If we fit the initial model to the whole data set:
We have AIC value= 598.13
> fit.initial_prime<-glm(V1~.,data = mydata)
> summary(fit.initial_prime)

Call:
glm(formula = V1 ~ ., data = mydata)

Deviance Residuals:

```
   Min      1Q  Median      3Q     Max
-3.1103  -0.6338  -0.1098   0.5803   3.7018

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.11085    0.25378   0.437    0.663
V2           8.26752    1.14630   7.212  1.2e-11 ***
V3           0.03651    0.15698   0.233    0.816
V4          -0.27470    0.27790  -0.988    0.324
V5           1.64357    0.11685  14.066  < 2e-16 ***
V6         -11.83925    1.06907 -11.074  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 1.119893)

    Null deviance: 704.39  on 199  degrees of freedom
Residual deviance: 217.26  on 194  degrees of freedom
AIC: 598.13

Number of Fisher Scoring iterations: 2
```

In step 3, the coefficient of V3 is not significant, and the coefficient of V4 is mildly significant. To improve the model from the initial fitting, we use lasso to select important covariates

use Lasso to fit the model:

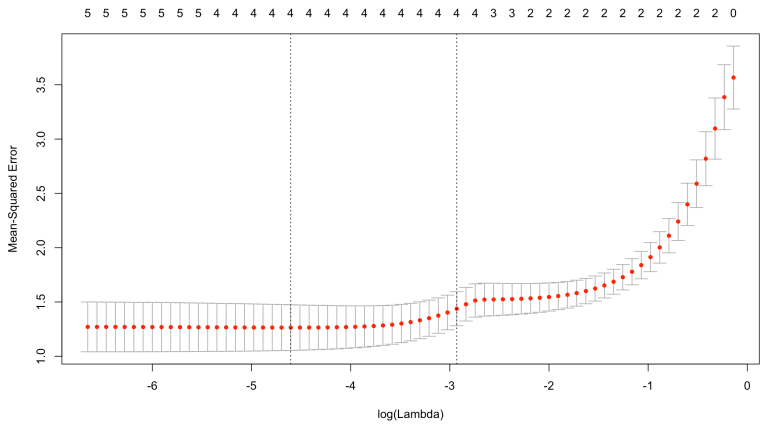


Figure 4 the mean squared error vs log (lambda) for Lasso regression to select the best lambda.

selecting best lambda by 10-fold cross validation:

```
> bestlambda
[1] 0.0100044
```

fit the model, lasso method selected V2, V4, V5, V6 in the model

```
> best.model<-glmnet(x,y,family = "gaussian",lambda = bestlambda)
> predict(best.model,s=bestlambda,type = "coefficients")
6 x 1 sparse Matrix of class "dgCMatrix"
                    1
(Intercept)  0.357739
V2           6.978522
```

V3 .
V4 -0.192811
V5 1.645650
V6 -10.621249

From Lasso , the selected variables are V2, V4, V5, V6. It excludes V3, which is not significant in the initial model.

The new model fitted is :
$V1 = 0.1326 + 8.6038 * V2 - 0.2920 * V3 + 1.5954 * V5 - 12.2971 * V6$ (2)

```
> fit2<-glm(V1~V2+V4+V5+V6,data = mydata.train)
> summary(fit2)

Call:
glm(formula = V1 ~ V2 + V4 + V5 + V6, data = mydata.train)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-3.0580  -0.6808  -0.0833   0.6203   3.7549

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1326     0.2964   0.447   0.6553
V2            8.6038     1.3121   6.557 7.72e-10 ***
V4           -0.2920     0.1397  -2.091   0.0382 *
V5            1.5954     0.1241  12.856  < 2e-16 ***
V6          -12.2971     1.2154 -10.118  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 1.117647)

    Null deviance: 567.41  on 159  degrees of freedom
Residual deviance: 173.24  on 155  degrees of freedom
AIC: 478.78

Number of Fisher Scoring iterations: 2
```

The residual plots show some improvement. The QQ-plot indicated that the residuals are more close to normal distribution. Most of residuals are more close to zero. However, this is still not a good fit since the residuals are not evenly distributed, and departure from the normal distribution.

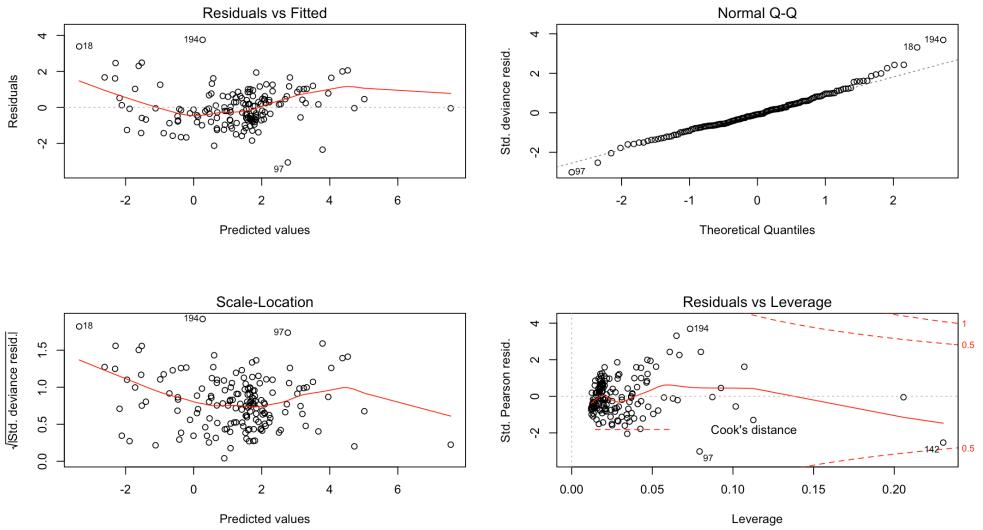


Figure 5 Residual plots for model (2). Linear regression to V1 with selected covariates V2,V4,V5,V6.

Model improvement:

(5)Improve model in step (4)

Since lasso has a nice property for selecting important covariates from potential variables, we consider it as the best linear model. To improve model (2), we can try to consider higher order linear model.

Here we consider polynomial regression. To fit a polynomial regression model, we need to consider two questions: (a)which covariates should we select (b) which degree (d)of polynomial functions should we fit.

(5.1) To answer (a), if it is possible, we should consider all combinations of the 5 covariates (V2,V3,V4,V5,V6), which is 32 (2^5). It is not only tedious and also computation intense. One possible solution is to consider the covariates are important in linear regression models.

We use subsets model selection method to pick up the covariates for different linear model:

Subset selection object
Call: regsubsets.formula(V1 ~ ., data = mydata)
5 Variables (and intercept)
Forced in Forced out
V2 FALSE FALSE
V3 FALSE FALSE
V4 FALSE FALSE
V5 FALSE FALSE
V6 FALSE FALSE
1 subsets of each size up to 5
Selection Algorithm: exhaustive
V2 V3 V4 V5 V6
1 (1) " " " " " " " " "*"
2 (1) " " " " " " "*" "*"
3 (1) "*" " " " " "*" "*"
4 (1) "*" " " "*" "*" "*"

5 (1) "*" "*" "*" "*" "*"

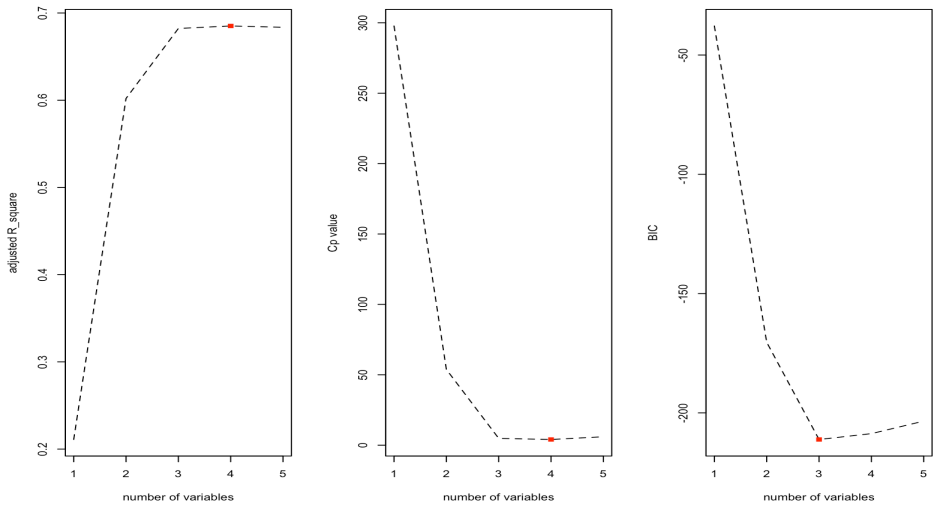


Figure 6 The above graphs show how the adjusted R^2, Cp value or BIC would change with the number of covariates we selected. The red dots in each graph indicate the number of covariates selected by each standard. For R^2 and Cp value standards, the best model contains four covariates V2, V4,V5, V6. For BIC, the selected best model contains three covariates V2,V5 and V6.

The results are consistent with the Lasso selection. If we select the number of variables by adjusted R^2 or Cp value, covariates V2, V4, V5, V6 were into the best linear model. If we select the variables by BIC, covariates V2, V5, V6 were selected.

(5.2) Use the covariates selected in (5.1) to fit polynomial regression model with different degrees d, choose the best d for each model by cross validation.

The models we are going to fitted are as followings:

# of covariates (k)	Covariates	d=1,2,3,4,5
k=1	V2	glm(V1~poly(V2,degree=d))
	V3	glm(V1~poly(V3,degree=d))
	V4	glm(V1~poly(V4, degree=d))
	V5	glm(V1~poly(V5, degree=d))
	V6	glm(V1~poly(V6, degree=d))
k=2	V5, V6	glm(V1~poly(V5,V6, degree=d))
k=3	V2,V5,V6	glm(V1~poly(V2,V5,V6, degree=d))
k=4	V2,V4,V5,V6	glm(V1~poly(V2,V4,V5,V6, degree=d))

. step 1:
k=1:
The cross-validation error for different d for when the number of covariates k=1 is:

```
> cv.error5
       V2       V3       V4       V5       V6
d=1 3.887266 4.819655 4.848086 5.070601 5.569903
d=2 3.523360 5.241781 5.206824 5.427859 5.325759
d=3 3.527035 3.682356 3.795410 3.796144 3.851775
d=4 4.253861 4.442541 4.229779 4.189918 4.266452
```

d=5 4.366427 4.754526 5.231141 5.599553 5.399338

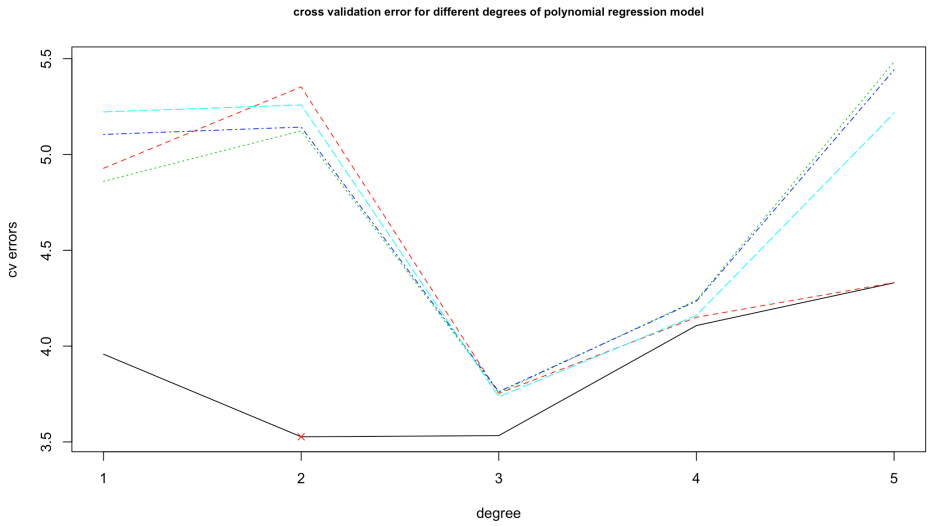


Figure 7 The cross validation errors for different models. The red-cross (x) indicates the smallest cross validation error (3.527035), when covariates V2 were fitted to a polynomial regression model with d=2.

So the fitted model is : $V_1 = 1.2034 - 9.4567 * V_2 - 13.0437 * {V_2}^2$ (3)

```
> poly_v2<-glm(V1~poly(V2, degree=2), data=mydata)
> summary(poly_v2)

Call:
glm(formula = V1 ~ poly(V2, degree = 2), data = mydata)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.7669  -1.0963  -0.3136   0.5768   5.6581

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           1.2034     0.1063  11.325  < 2e-16 ***
poly(V2, degree = 2)1  -9.4567     1.5027  -6.293 1.97e-09 ***
poly(V2, degree = 2)2 -13.0437     1.5027  -8.680 1.49e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 2.257992)

    Null deviance: 704.39  on 199  degrees of freedom
Residual deviance: 444.82  on 197  degrees of freedom
AIC: 735.45

Number of Fisher Scoring iterations: 2
```

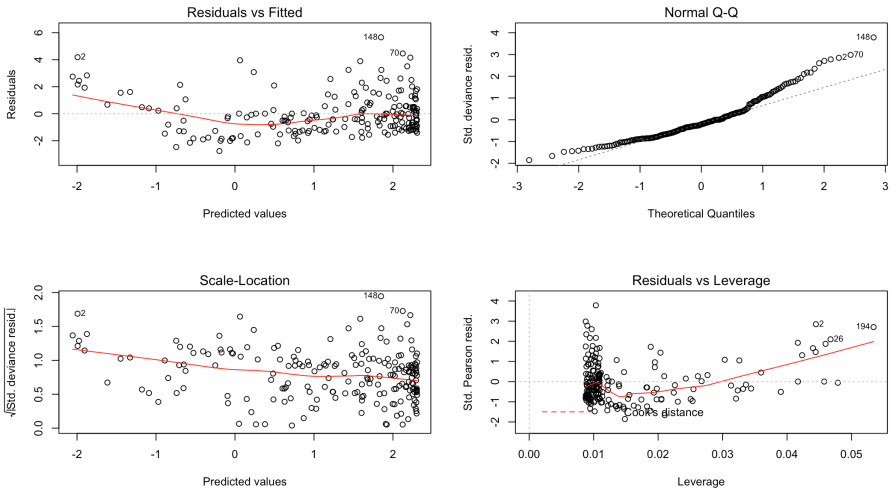


Figure 8 Residual plots for model (3). The QQ-plot indicates violation of normality assumption of the errors.

The red line is the predicted value of fitted model given different V2, the grey points are the original data set.

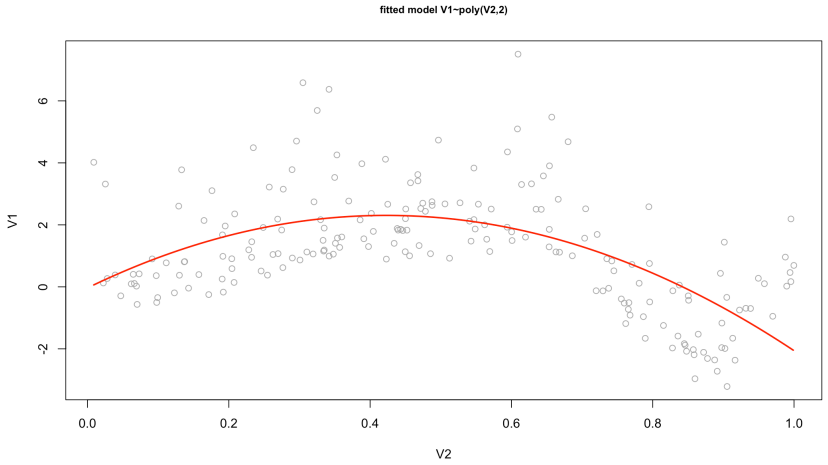


Figure 9 The red line is the predicated value of the model (3), the grey points are the true observations.

step 2:
.k=2

polynomial with two variables V5 V6 selected.

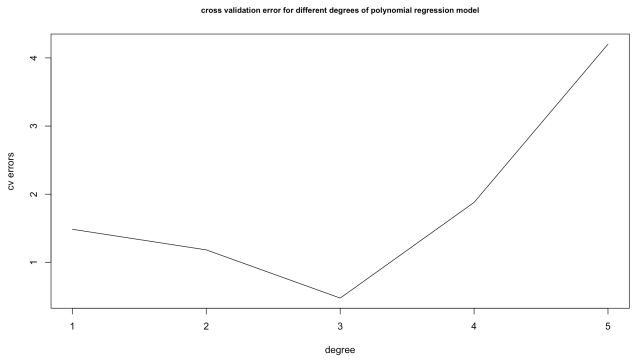


Figure 10 cross validation error is minimized when d=3, when the selected covariates are V5 and V6.

cv.error
[1] 1.4857886 1.1834070 0.4762358 1.8800980 4.2008376

Therefore the fitted model is:
fitted model :

$$V1 = 1.39466 + 20.81718 * V_5 - 1.30966 * {V_5}^2 - 0.15861 * {V_5}^3 - 19.08055 * V_6 - 104.24259 * (V_5 V_6) + 4.87665 * ({V_5}^2 V_6) - 2.03542 * {V_6}^2 + 18.70482 * (V_5 {V_6}^2) + 10.33903 * {V_6}^3 \quad (4)$$

only $V_5, V_6, V_5V_6, {V_6}^2, {V_6}^3$ are significant.

```
> glm2.fit<-glm(V1~poly(V5,V6, degree=3),data = mydata)
> summary(glm2.fit)
Call:
glm(formula = V1 ~ poly(V5, V6, degree = 3), data = mydata)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.4443  -0.4529  -0.0437   0.4153   3.8047

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                    1.39466    0.05748  24.262  < 2e-16 ***
poly(V5, V6, degree = 3)1.0   20.81718    1.26252  16.489  < 2e-16 ***
poly(V5, V6, degree = 3)2.0   -1.30966    1.04655  -1.251  0.21232
poly(V5, V6, degree = 3)3.0   -0.15861    0.71139  -0.223  0.82380
poly(V5, V6, degree = 3)0.1  -19.08055    0.81436 -23.430  < 2e-16 ***
poly(V5, V6, degree = 3)1.1 -104.24259   16.33368  -6.382  1.3e-09 ***
poly(V5, V6, degree = 3)2.1    4.87665   13.07450   0.373  0.70957
poly(V5, V6, degree = 3)0.2   -2.03542    0.73911  -2.754  0.00646 **
poly(V5, V6, degree = 3)1.2   18.70482   13.34901   1.401  0.16278
poly(V5, V6, degree = 3)0.3   10.33903    0.74946  13.795  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.4547515)

    Null deviance: 704.392  on 199  degrees of freedom
Residual deviance:  86.403  on 190  degrees of freedom
AIC: 421.72
Number of Fisher Scoring iterations: 2
```

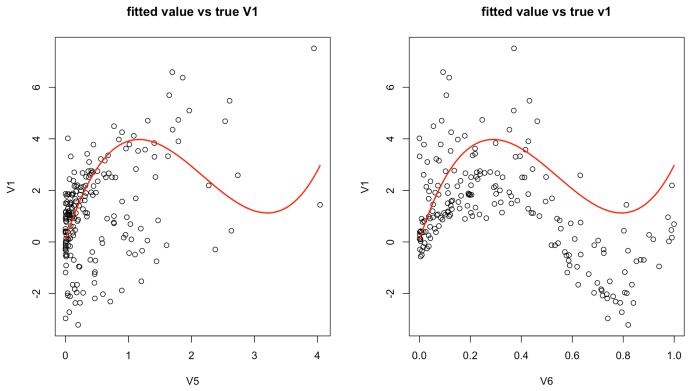


Figure 11 fitted values are the read line, true values are the black dots. The two graphs show how the fitted model captures the variation in the V5 and V6 direction.

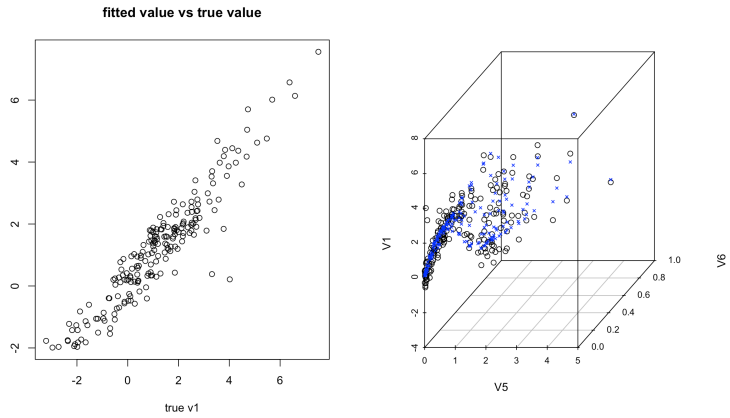


Figure 12 Left graph is the true value vs the fitted value. When model is fitted well, the points would be around the diagonal line of x –axis and y-axis. The left plot is the 3D scatter plot, black dots (o) are the true value , blue-cross (x) are the fitted value.

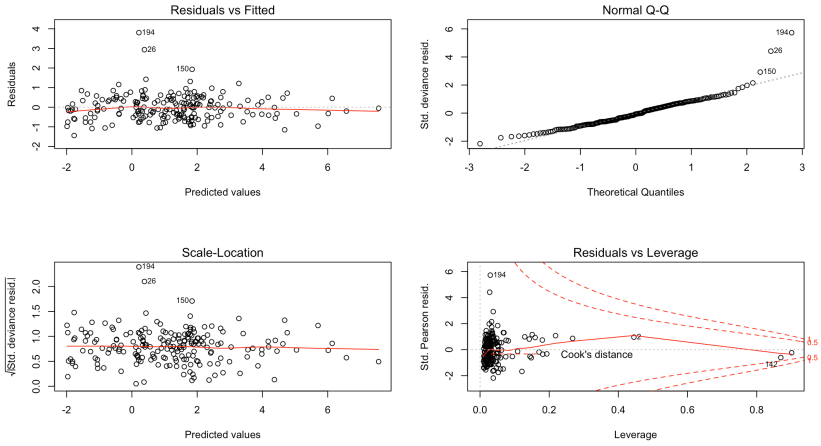


Figure 13 Residual plot is improved a lot. The residuals are basically normal distribute

step 3:
. k=3 covariates V2, V5, V6 being selected.

The result switch between d=2 and d=3. The cv-error difference between d=2 and d=3 are comparable small.
For example:

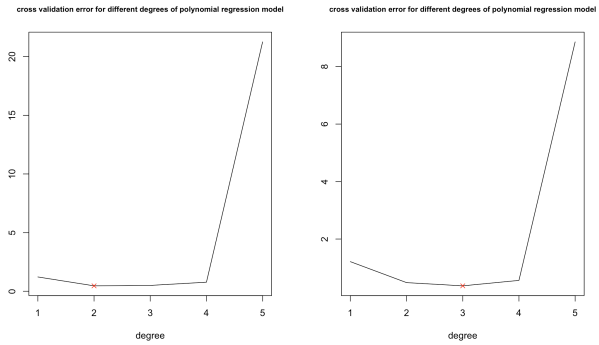


Figure 14 Cross validation error and the corresponding d, when V2, V5 and V6 are fitted to the model.

> cv.error_3
[1] 1.2132896 0.4839271 0.3717472 0.5612892 8.8592851
> cv.error_3
[1] 1.2215274 0.4642866 0.5006255 0.7744926 21.2538947

Here we tried to calculate the cross-validation error difference between d=2 and d=3, when doing the cross validation for 100 times:
> summary(dif)
Min. 1st Qu. Median Mean 3rd Qu. Max.
-0.17110 -0.09802 -0.07578 -0.06461 -0.05000 0.23190
> dif
[1] -0.0763457563 -0.0642182029 -0.1375715776 -0.0925236322 -0.0786590535 -0.1147766387
[7] -0.1167520764 0.1533245092 -0.0997149032 -0.0758754985 -0.0107807223 -0.0725347193
[13] -0.1125070165 -0.0378587331 -0.0913101407 -0.0665628480 -0.0700604737 0.1479208394
……
[91] -0.0251916853 -0.0971606431 -0.0608189163 -0.0672024839 -0.0827579030 -0.0763564851
[97] -0.1330267267 -0.1169905493 -0.0756756288 -0.0406785881

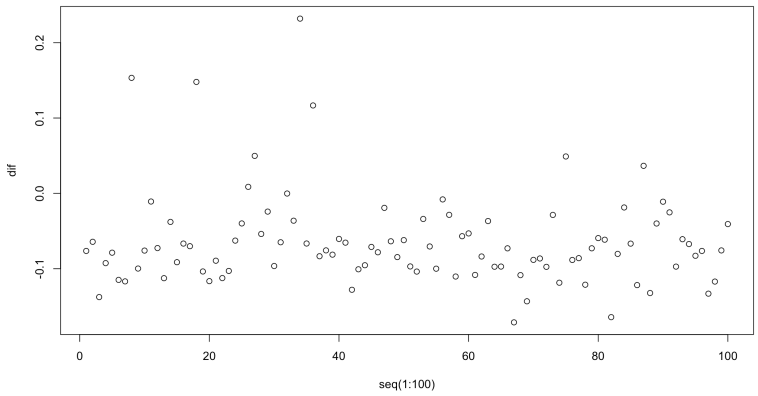
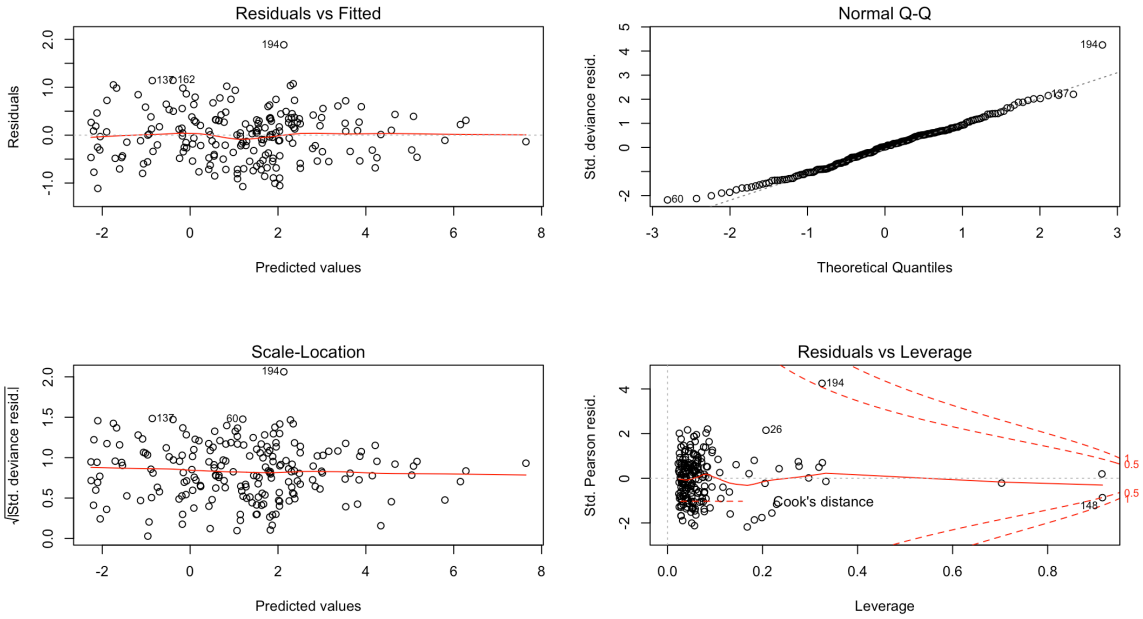


Figure 15 Cross validation error differences between d=2 and d=3. Differences were calculated by fitting 100 times.

Although, when d=3, the cross validation error is smaller, but the difference is quite close to zero. However, d=3 means a complicated model is and the data might have over fitted. Therefore, it is reasonable to consider a polynomial model with d=2 when V2, V5, V6 are selected.

d=3 we fit model (5) which is summarized below:

```
> summary(glm3.fit)

Call:
glm(formula = V1 ~ poly(V2, V5, V6, degree = 3), data = mydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.11267  -0.40728   0.01345   0.32667   1.88565

Coefficients: (4 not defined because of singularities)
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -9.623e+00  4.011e+00  -2.399 0.017443 *
poly(V2, V5, V6, degree = 3)1.0.0 -1.342e+02  3.332e+01  -4.028 8.22e-05 ***
poly(V2, V5, V6, degree = 3)2.0.0  7.680e+01  2.504e+01   3.067 0.002486 **
poly(V2, V5, V6, degree = 3)3.0.0 -1.399e+02  4.010e+01  -3.489 0.000607 ***
poly(V2, V5, V6, degree = 3)0.1.0  2.080e+02  5.936e+01   3.504 0.000575 ***
poly(V2, V5, V6, degree = 3)1.1.0  4.777e+02  2.236e+02   2.136 0.033965 *
poly(V2, V5, V6, degree = 3)2.1.0  1.797e+03  5.316e+02   3.380 0.000886 ***
poly(V2, V5, V6, degree = 3)0.2.0 -2.769e-01  2.683e+00  -0.103 0.917910
poly(V2, V5, V6, degree = 3)1.2.0 -1.878e+01  1.456e+02  -0.129 0.897521
poly(V2, V5, V6, degree = 3)0.3.0 -2.404e-01  6.060e-01  -0.397 0.692117
poly(V2, V5, V6, degree = 3)0.0.1         NA         NA     NA       NA
poly(V2, V5, V6, degree = 3)1.0.1         NA         NA     NA       NA
poly(V2, V5, V6, degree = 3)2.0.1  5.609e+03  1.904e+03   2.946 0.003640 **
poly(V2, V5, V6, degree = 3)0.1.1         NA         NA     NA       NA
poly(V2, V5, V6, degree = 3)1.1.1 -3.607e+04  1.155e+04  -3.123 0.002080 **
poly(V2, V5, V6, degree = 3)0.2.1  1.957e+01  1.070e+02   0.183 0.855045
poly(V2, V5, V6, degree = 3)0.0.2         NA         NA     NA       NA
poly(V2, V5, V6, degree = 3)1.0.2 -4.709e+03  1.675e+03  -2.811 0.005468 **
poly(V2, V5, V6, degree = 3)0.1.2  8.180e+02  2.924e+02   2.798 0.005693 **
poly(V2, V5, V6, degree = 3)0.0.3  1.201e+02  3.815e+01   3.148 0.001921 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.2913669)

    Null deviance: 704.392  on 199  degrees of freedom
Residual deviance:  53.612  on 184  degrees of freedom
AIC: 338.26

Number of Fisher Scoring iterations: 2
```



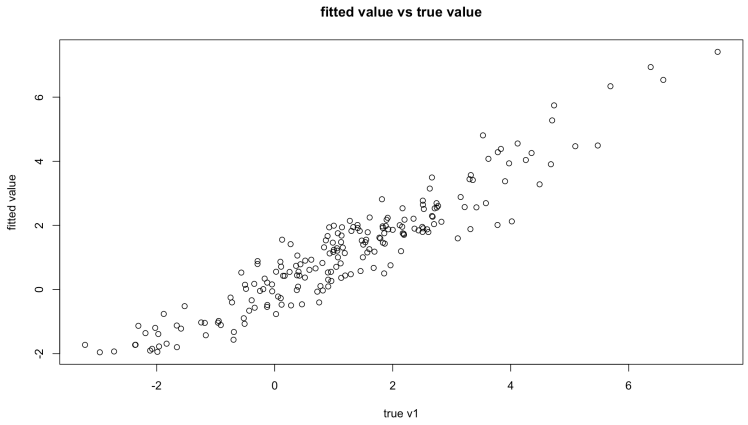
The fitted model is:

$V_1 = 24.4833 + 100.7756 * V_2 + 158.0687 * {V_2}^2 + 24.4888 * V_5 - 346.0536(V_2 * V_5) - 0.9330 * {V_5}^2 - 4727.6798 * (V_2 * V_6) + 178.8877 * (V_5 * V_6) + 150.3101 * {V_6}^2$ (6)

Only ${V_5}^2$ are not significant. V_6 cannot be estimated.

```
> glm3.fit_d2<-glm(V1~poly(V2,V5,V6, degree=2),data = mydata)
> summary(glm3.fit_d2
Call:
```

```
glm(formula = V1 ~ poly(V2, V5, V6, degree = 2), data = mydata)
 Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.48755  -0.47748  -0.00805   0.41280   1.89384

Coefficients: (1 not defined because of singularities)
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      24.4833     1.6187  15.125  < 2e-16 ***
poly(V2, V5, V6, degree = 2)1.0.0   100.7756     8.3362  12.089  < 2e-16 ***
poly(V2, V5, V6, degree = 2)2.0.0   158.0687    12.2369  12.917  < 2e-16 ***
poly(V2, V5, V6, degree = 2)0.1.0    24.4888     1.2510  19.575  < 2e-16 ***
poly(V2, V5, V6, degree = 2)1.1.0  -346.0536    82.4950  -4.195 4.18e-05 ***
poly(V2, V5, V6, degree = 2)0.2.0    -0.9330     0.7219  -1.292  0.19776
poly(V2, V5, V6, degree = 2)0.0.1        NA         NA      NA       NA
poly(V2, V5, V6, degree = 2)1.0.1 -4727.6798   331.7572 -14.250  < 2e-16 ***
poly(V2, V5, V6, degree = 2)0.1.1   178.8877    66.8865   2.674  0.00813 **
poly(V2, V5, V6, degree = 2)0.0.2   150.3101    10.1583  14.797  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for gaussian family taken to be 0.4087035)
Null deviance: 704.392  on 199  degrees of freedom
Residual deviance:  78.062  on 191  degrees of freedom
AIC: 399.41
Number of Fisher Scoring iterations: 2
Number of Fisher Scoring iterations: 2
```



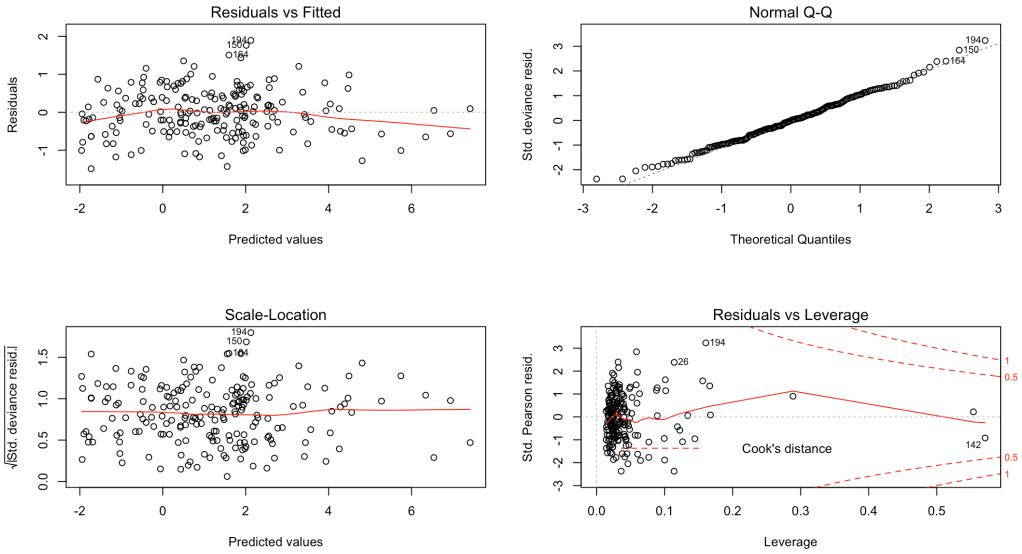
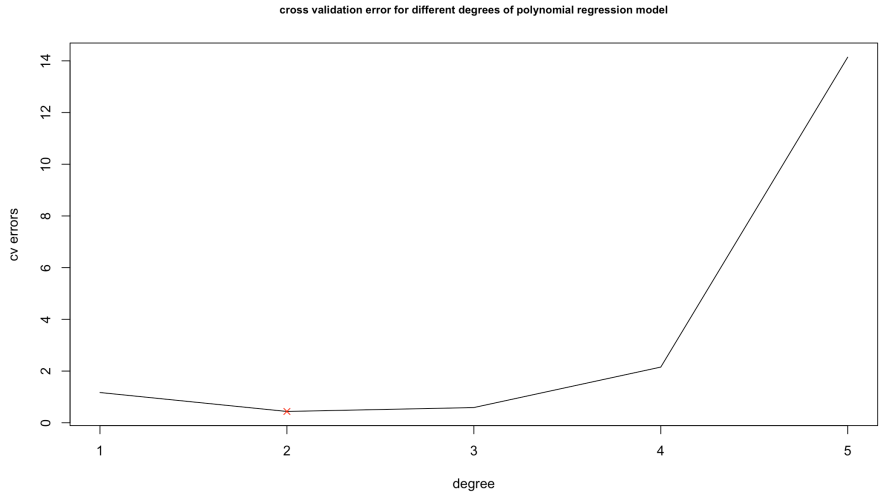


Figure 16 The residual plots indicate that expect for three points (164, 150, 194), the error terms follow a normal distribution.

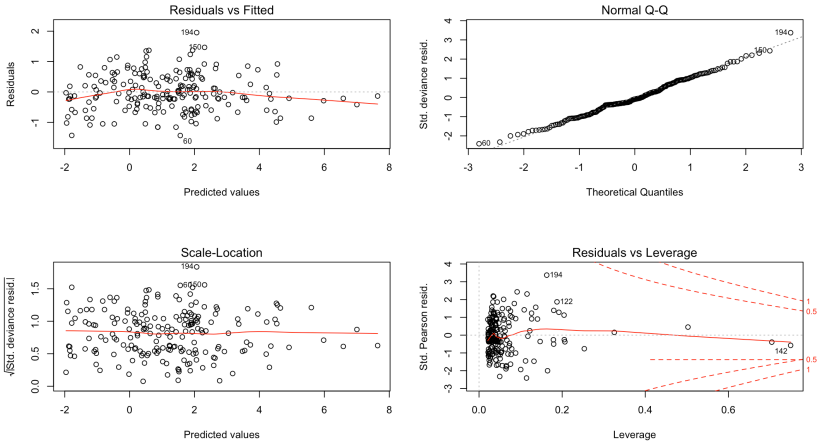
Step 4:
.k=4, the smallest cross validation error is obtained when d=2.

cv.error_4
[1] 1.1707688 0.4380063 0.5891632 2.1552870 14.1413605



```
> glm4.fit<-glm(V1~poly(V2,V4,V5,V6, degree=which.min(cv.error_4)),data = mydata)
> summary(glm4.fit)

Call:
glm(formula = V1 ~ poly(V2, V4, V5, V6, degree = which.min(cv.error_4)),
    data = mydata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4348  -0.4156  -0.0530   0.4522   1.9525

Coefficients: (2 not defined because of singularities)
                                                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                                                      23.5965     1.5995  14.753  < 2e-16 ***
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))1.0.0.0   117.4099     9.7627  12.026  < 2e-16 ***
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))2.0.0.0   156.0734    12.2424  12.749  < 2e-16 ***
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))0.1.0.0     0.7449     1.7791   0.419 0.675911
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))1.1.0.0   158.4675    78.7061   2.013 0.045505 *
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))0.2.0.0   -67.0556    16.9995  -3.945 0.000113 ***
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))0.0.1.0    69.2510    12.2449   5.656 5.74e-08 ***
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))1.0.1.0        NA         NA      NA       NA
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))0.1.1.0    -0.1696     6.7330  -0.025 0.979933
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))0.0.2.0    -0.7379     0.7553  -0.977 0.329871
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))0.0.0.1        NA         NA      NA       NA
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))1.0.0.1 -4670.9458   332.0950 -14.065  < 2e-16
***
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))0.1.0.1  -116.7224    63.8540  -1.828 0.069150 .
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))0.0.1.1   174.8668    72.5894   2.409 0.016967 *
poly(V2, V4, V5, V6, degree = which.min(cv.error_4))0.0.0.2   148.4662    10.1875  14.573  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.4001469)

    Null deviance: 704.392  on 199  degrees of freedom
Residual deviance:  74.827  on 187  degrees of freedom
AIC: 398.95

Number of Fisher Scoring iterations: 2
```

The residual plots are almost the same, no significant improvements of model.



Summary:

linear	V_2, V_3, V_4, V_5, V_6 (initial fitting)	V_2, V_4, V_5, V_6 (Lasso)	V_2, V_5, V_6	V_5, V_6	V_6
AIC	598.13	478.78	597.03	641.09	776.97
poly-nomial	$V_2, V_4, V_5, V_6, d = 2$	$V_2, V_5, V_6, d = 3$	$V_2, V_5, V_6, d = 2$	$V_5, V_6, d = 3$	V2, d=2
AIC	398.95	338.26	399.41	421.72	735.45

As we stated before, Lasso can select the most significant covariates in the linear regression model. We fit linear models (the AIC value see attachment) with 3 covariates (V_2, V_5, V_6), 2 covariates (V_5, V_6) and one covariate (V_6) to support that Lasso is the best among the other linear regression.

For nonlinear regression model, we consider polynomial regression here. For a given number of variables k, we fit a polynomial model with degree d (d from 1 to 5). When k is fixed, the covariates are those selected by subsets selection method in the linear model. We use cross validation to select the best d for a given k (also given covariates) .

To prevent over-fitting, we would like to select a simple model where the AIC value is comparable similar. In step 3, we showed that the cross validation error for $d = 3$ and $d = 2$ is similar. Therefore, we prefer $poly(V_2, V_5, V_6, d = 2)$ to model $poly(V_2, V_5, V_6, d = 3)$.

Fit two best model candidates $poly(V_5, V_6, d = 3)$ and $poly(V_2, V_5, V_6, d = 2)$

By comparing the AIC value, the best model is:

$$V_1 = 0.291 - 13.153 * V_6 - 2.137 * {V_6}^2 + 10.895 * {V_6}^3 + 2.900 * V_5 - 2.458 * (V_5 : V_6) \quad (7)$$

```
> best.fit1<-glm(V1~poly(V6,degree=3)+V5+V5:V6,data = mydata)
> summary(best.fit1)

Call:
glm(formula = V1 ~ poly(V6, degree = 3) + V5 + V5:V6, data = mydata)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.4998  -0.4738  -0.0537   0.3821   3.7515

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)          0.29064    0.06222   4.671 5.59e-06 ***
poly(V6, degree = 3)1 -13.15292    0.88671 -14.833  < 2e-16 ***
poly(V6, degree = 3)2  -2.13691    0.74005  -2.888  0.00432 **
poly(V6, degree = 3)3  10.89504    0.68621  15.877  < 2e-16 ***
V5                 2.89990    0.16366  17.719  < 2e-16 ***
V5:V6               -2.45894    0.28466  -8.638 2.09e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.4615098)

   Null deviance: 704.392  on 199  degrees of freedom
Residual deviance:  89.533  on 194  degrees of freedom
AIC: 420.83

Number of Fisher Scoring iterations: 2


> best.fit2<-glm(V1~poly(V2,degree = 2)+V2:V5+V2:V6+poly(V6,degree=2), data = mydata)
> summary(best.fit2)

Call:
glm(formula = V1 ~ poly(V2, degree = 2) + V2:V5 + V2:V6 + poly(V6,
   degree = 2), data = mydata)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-3.1968  -0.7061  -0.0693   0.5518   3.7351

Coefficients: (1 not defined because of singularities)
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)          66.8413    7.8611   8.503 4.91e-15 ***
poly(V2, degree = 2)1  990.1020   119.1650   8.309 1.65e-14 ***
poly(V2, degree = 2)2  268.7820    34.4478   7.803 3.66e-13 ***
poly(V6, degree = 2)1       NA       NA      NA       NA
poly(V6, degree = 2)2  142.2351    15.9343   8.926 3.35e-16 ***
V2:V5                1.9495    0.1554  12.548  < 2e-16 ***
V2:V6             -266.1068    31.5529  -8.434 7.57e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 1.012223)
```

Null deviance: 704.39 on 199 degrees of freedom
Residual deviance: 196.37 on 194 degrees of freedom
AIC: 577.91

Number of Fisher Scoring iterations: 2

(6) The final model is:

$$V_1 = 0.291 - 13.153 * V_6 - 2.137 * {V_6}^2 + 10.895 * {V_6}^3 + 2.900 * V_5 - 2.458 * (V_5 : V_6) \quad (7)$$
$$(0.06222) \quad (0.88671) \quad (0.74005) \quad (0.68621) \quad (0.16366) \quad (0.28466)$$

95% confidence intervals for coefficients (interval ignored):

```
> ci
                              lower.ci   upper.ci
poly(V6, degree = 3)1 -14.901745 -11.404087
poly(V6, degree = 3)2  -3.596491  -0.677338
poly(V6, degree = 3)3   9.541646  12.248425
V5                      2.577115   3.222678
V5:V6                  -3.020369  -1.897518
```

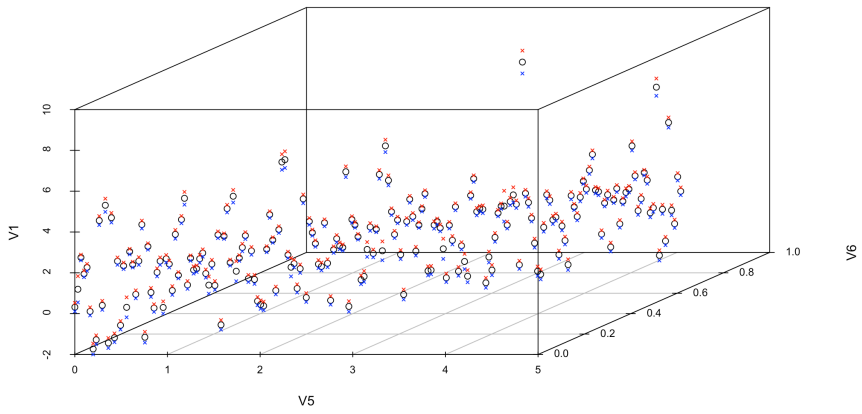


Figure 17 is point confidence bands for each estimated V_1. Black circles are fitted value by model (7). Blue-cross (x) is the lower bound and red-cross (x) is the upper bound for a 95% confidence interval of a given point. The estimated V_1 and the confidence intervals are listed in the appendix.

- Appendix.

By subsets selection, the best model selected by BIC contains three variables V2,V5,V6

```
> fit.subsets_3v<-glm(V1~V2+V5+V6,data = mydata)
> summary(fit.subsets_3v)

Call:
glm(formula = V1 ~ V2 + V5 + V6, data = mydata)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-3.1978  -0.6488  -0.0577   0.6110   3.7632

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1205     0.2542   0.474    0.636
V2            8.1731     1.1473   7.124 1.95e-11 ***
V5            1.6522     0.1170  14.125  < 2e-16 ***
V6          -11.7246     1.0690 -10.968  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 1.124678)

    Null deviance: 704.39  on 199  degrees of freedom
Residual deviance: 220.44  on 196  degrees of freedom
AIC: 597.03

Number of Fisher Scoring iterations: 2
```

fit a linear regression model with two variables V5 and V6

```
> fit.subsets_2v<-glm(V1~V5+V6, data=mydata)
> summary(fit.subsets_2v)

Call:
```

```
glm(formula = V1 ~ V5 + V6, data = mydata)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-4.0249  -0.9152   0.0937   0.9477   2.7664

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.7266     0.1315   13.13   <2e-16 ***
V5            1.8012     0.1288   13.98   <2e-16 ***
V6           -4.3561     0.3024  -14.41   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 1.408701)

    Null deviance: 704.39  on 199  degrees of freedom
Residual deviance: 277.51  on 197  degrees of freedom
AIC: 641.09

Number of Fisher Scoring iterations: 2
```

fit linear with V6:

```
> fit.subsets_1v<-glm(V1~V6, data=mydata)
> summary(fit.subsets_1v)

Call:
glm(formula = V1 ~ V6, data = mydata)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.9835  -1.3223  -0.1270   0.9829   6.4121

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.1935     0.1790  12.253  < 2e-16 ***
V6           -2.9587     0.4018  -7.363 4.71e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 2.792757)

    Null deviance: 704.39  on 199  degrees of freedom
Residual deviance: 552.97  on 198  degrees of freedom
AIC: 776.97

Number of Fisher Scoring iterations: 2
```

. confidence intervals for estimated V1 by model 7:

```
> ci_v1
```

	v5.grid	v6.grid	lower ci	upper ci	fitted value
1	0.0001370915	7.876911e-05	0.11030976	0.52477707	0.31754342
2	0.0204427789	5.100731e-03	0.52669300	1.81168185	1.16918743
3	0.0407484663	1.012269e-02	2.56085578	2.83894049	2.69989813
4	0.0610541536	1.514466e-02	1.71142991	2.03466520	1.87304756
5	0.0813598410	2.016662e-02	1.99492130	2.33613830	2.16552980
6	0.1016655283	2.518858e-02	-0.21446354	0.18450395	-0.01497979
7	0.1219712157	3.021054e-02	-2.14861819	-1.61556859	-1.88209339
8	0.1422769030	3.523251e-02	-1.67314099	-1.23920989	-1.45617544
9	0.1625825904	4.025447e-02	4.13966876	4.58005806	4.35986341
10	0.1828882778	4.527643e-02	-0.04014014	0.40252682	0.18119334
11	0.2031939651	5.029839e-02	4.73477483	5.38380948	5.05929216
12	0.2234996525	5.532035e-02	-1.96056518	-1.46876554	-1.71466536
13	0.2438053398	6.034232e-02	4.16204356	4.63340342	4.39772349
14	0.2641110272	6.536428e-02	-1.72331541	-1.29262506	-1.50797023
15	0.2844167146	7.038624e-02	2.03922729	2.40247900	2.22085314
16	0.3047224019	7.540820e-02	-1.15600516	-0.73866580	-0.94733548
17	0.3250280893	8.043017e-02	1.78229560	2.09971722	1.94100641
18	0.3453337766	8.545213e-02	-0.60763188	0.37215672	-0.11773758
19	0.3656394640	9.047409e-02	2.45088345	2.72286399	2.58687372
20	0.3859451513	9.549605e-02	1.78375736	2.06576848	1.92476292
21	0.4062508387	1.005180e-01	0.24905418	0.64000457	0.44452938
22	0.4265565261	1.055400e-01	1.86057960	2.24094203	2.05076082
23	0.4468622134	1.105619e-01	3.61778710	4.00583951	3.81181330
24	0.4671679008	1.155839e-01	-1.99582488	-1.46828353	-1.73205420
25	0.4874735881	1.206059e-01	2.53199054	2.84309906	2.68754480
26	0.5077792755	1.256278e-01	0.19072573	0.63156988	0.41114780
27	0.5280849629	1.306498e-01	-0.56530474	-0.19210652	-0.37870563
28	0.5483906502	1.356718e-01	1.14786441	1.54043874	1.34415157
29	0.5686963376	1.406937e-01	1.68905582	2.05204510	1.87055046
30	0.5890020249	1.457157e-01	-0.72251036	-0.12890301	-0.42570668
31	0.6093077123	1.507376e-01	1.75500118	2.11807735	1.93653927
32	0.6296133996	1.557596e-01	1.48291208	1.82260367	1.65275788
33	0.6499190870	1.607816e-01	0.11525343	0.54439336	0.32982340
34	0.6702247744	1.658035e-01	2.83989006	3.27309795	3.05649401
35	0.6905304617	1.708255e-01	0.87453903	1.18330413	1.02892158
36	0.7108361491	1.758474e-01	3.55027823	3.90787298	3.72907561
37	0.7311418364	1.808694e-01	4.43491452	5.05998278	4.74744865
38	0.7514475238	1.858914e-01	0.28448820	0.66950035	0.47699428
39	0.7717532111	1.909133e-01	1.63397388	1.89884040	1.76640714
40	0.7920588985	1.959353e-01	0.98171505	1.33923313	1.16047409
41	0.8123645859	2.009573e-01	1.03630006	1.35846695	1.19738350
42	0.8326702732	2.059792e-01	1.46105224	1.85252043	1.65678634
43	0.8529759606	2.110012e-01	1.73679813	2.09514308	1.91597060
44	0.8732816479	2.160231e-01	0.68352080	1.08568595	0.88460337
45	0.8935873353	2.210451e-01	-0.12430021	0.71294319	0.29432149
46	0.9138930227	2.260671e-01	1.09620400	1.50517388	1.30068894
47	0.9341987100	2.310890e-01	0.01010313	0.44315042	0.22662677
48	0.9545043974	2.361110e-01	2.52027001	2.84916946	2.68471974
49	0.9748100847	2.411330e-01	-1.98795233	-1.51982276	-1.75388755
50	0.9951157721	2.461549e-01	2.43338974	2.69974831	2.56656902
51	1.0154214594	2.511769e-01	3.70275070	4.07602142	3.88938606
52	1.0357271468	2.561988e-01	1.07262114	1.36475587	1.21868850
53	1.0560328342	2.612208e-01	4.13828701	4.75481993	4.44655347
54	1.0763385215	2.662428e-01	0.25593892	1.23738753	0.74666322

55	1.0966442089	2.712647e-01	1.22998864	1.51246107	1.37122486
56	1.1169498962	2.762867e-01	1.67415541	2.04275311	1.85845426
57	1.1372555836	2.813087e-01	2.24953632	2.58421745	2.41687688
58	1.1575612710	2.863306e-01	0.11854999	0.46484303	0.29169651
59	1.1778669583	2.913526e-01	1.42872443	1.80872634	1.61872538
60	1.1981726457	2.963745e-01	-0.02724859	0.41819622	0.19547381
61	1.2184783330	3.013965e-01	-1.09372650	-0.69130443	-0.89251547
62	1.2387840204	3.064185e-01	-1.29740285	-0.88061162	-1.08900723
63	1.2590897077	3.114404e-01	-1.38981457	-0.96550660	-1.17766058
64	1.2793953951	3.164624e-01	1.42742013	1.72335714	1.57538864
65	1.2997010825	3.214844e-01	3.08464786	3.40892249	3.24678518
66	1.3200067698	3.265063e-01	1.83966247	2.10903082	1.97434665
67	1.3403124572	3.315283e-01	-0.71322854	-0.33308286	-0.52315570
68	1.3606181445	3.365502e-01	2.27088836	2.59892933	2.43490885
69	1.3809238319	3.415722e-01	5.34099438	6.10238104	5.72168771
70	1.4012295193	3.465942e-01	5.40514072	6.21937493	5.81225782
71	1.4215352066	3.516161e-01	0.96425496	1.28010372	1.12217934
72	1.4418408940	3.566381e-01	0.04240797	0.94340249	0.49290523
73	1.4621465813	3.616601e-01	0.46149890	0.82641959	0.64395924
74	1.4824522687	3.666820e-01	-0.79976189	-0.39637918	-0.59807053
75	1.5027579560	3.717040e-01	0.13829303	0.55094002	0.34461652
76	1.5230636434	3.767259e-01	3.52642229	3.95405776	3.74024003
77	1.5433693308	3.817479e-01	-1.33584215	-0.90419909	-1.12002062
78	1.5636750181	3.867699e-01	2.42011432	2.74990083	2.58500757
79	1.5839807055	3.917918e-01	1.85838981	2.16023494	2.00931238
80	1.6042863928	3.968138e-01	1.29202114	1.60811620	1.45006867
81	1.6245920802	4.018358e-01	0.21984783	0.61615189	0.41799986
82	1.6448977675	4.068577e-01	0.08644379	0.47222545	0.27933462
83	1.6652034549	4.118797e-01	2.22358234	2.56140969	2.39249602
84	1.6855091423	4.169016e-01	0.19242207	0.57960956	0.38601582
85	1.7058148296	4.219236e-01	-1.66762412	-1.24850930	-1.45806671
86	1.7261205170	4.269456e-01	0.84273681	1.16127074	1.00200377
87	1.7464262043	4.319675e-01	1.33927067	1.68931800	1.51429434
88	1.7667318917	4.369895e-01	0.97030487	1.29641613	1.13336050
89	1.7870375791	4.420114e-01	0.87764710	1.18509479	1.03137094
90	1.8073432664	4.470334e-01	4.45498360	4.98038364	4.71768362
91	1.8276489538	4.520554e-01	-2.16301990	-1.65057948	-1.90679969
92	1.8479546411	4.570773e-01	2.17177984	2.48567672	2.32872828
93	1.8682603285	4.620993e-01	1.90042725	2.17359114	2.03700920
94	1.8885660158	4.671213e-01	1.29237875	1.57013364	1.43125620
95	1.9088717032	4.721432e-01	-0.91688579	-0.48356303	-0.70022441
96	1.9291773906	4.771652e-01	-0.82945529	-0.29615800	-0.56280664
97	1.9494830779	4.821871e-01	0.37288774	1.09645954	0.73467364
98	1.9697887653	4.872091e-01	1.62202493	1.99792405	1.80997449
99	1.9900944526	4.922311e-01	0.29811278	0.67935516	0.48873397
100	2.0104001400	4.972530e-01	1.50542649	1.79377442	1.64960046
101	2.0307058274	5.022750e-01	4.09749283	4.52109221	4.30929252
102	2.0510115147	5.072970e-01	0.08012592	1.00323177	0.54167884
103	2.0713172021	5.123189e-01	5.35425499	5.96140772	5.65783135
104	2.0916228894	5.173409e-01	3.69895757	4.18718220	3.94306988
105	2.1119285768	5.223628e-01	2.21747989	2.54930460	2.38339224
106	2.1322342641	5.273848e-01	1.06780862	1.41871877	1.24326370
107	2.1525399515	5.324068e-01	1.76705333	2.08869401	1.92787367
108	2.1728456389	5.374287e-01	-0.01487405	0.42931850	0.20722222
109	2.1931513262	5.424507e-01	-2.01838930	-1.52980802	-1.77409866

110	2.2134570136	5.474727e-01	1.60784505	1.94723012	1.77753759
111	2.2337627009	5.524946e-01	2.67429013	3.00831512	2.84130262
112	2.2540683883	5.575166e-01	1.79968756	2.14138375	1.97053566
113	2.2743740757	5.625385e-01	0.03378331	0.46268203	0.24823267
114	2.2946797630	5.675605e-01	1.36615833	1.64221437	1.50418635
115	2.3149854504	5.725825e-01	2.14198172	2.40133665	2.27165919
116	2.3352911377	5.776044e-01	2.82896543	3.15525795	2.99211169
117	2.3555968251	5.826264e-01	-1.04263178	-0.59158645	-0.81710911
118	2.3759025124	5.876484e-01	-1.00906922	-0.59937961	-0.80422441
119	2.3962081998	5.926703e-01	1.20121030	1.55922985	1.38022008
120	2.4165138872	5.976923e-01	1.38216129	1.67138766	1.52677447
121	2.4368195745	6.027142e-01	1.02095593	1.35070470	1.18583032
122	2.4571252619	6.077362e-01	-0.35701599	0.63385655	0.13842028
123	2.4774309492	6.127582e-01	-1.51577724	-1.10072408	-1.30825066
124	2.4977366366	6.177801e-01	1.08319869	1.40479463	1.24399666
125	2.5180423239	6.228021e-01	0.27728045	0.69015613	0.48371829
126	2.5383480113	6.278241e-01	1.92622052	2.26680129	2.09651090
127	2.5586536987	6.328460e-01	-1.30764960	-0.87615639	-1.09190300
128	2.5789593860	6.378680e-01	-0.03999188	0.31077058	0.13538935
129	2.5992650734	6.428899e-01	-1.03135817	-0.28623905	-0.65879861
130	2.6195707607	6.479119e-01	-1.70040635	-1.10949777	-1.40495206
131	2.6398764481	6.529339e-01	2.36684323	2.68391250	2.52537787
132	2.6601821355	6.579558e-01	3.16664671	3.48577788	3.32621229
133	2.6804878228	6.629778e-01	1.50123904	1.86375569	1.68249737
134	2.7007935102	6.679998e-01	1.67248843	1.93454508	1.80351676
135	2.7210991975	6.730217e-01	1.55891599	1.94008242	1.74949920
136	2.7414048849	6.780437e-01	-2.12181816	-1.63446061	-1.87813938
137	2.7617105722	6.830656e-01	-0.90915795	-0.37353811	-0.64134803
138	2.7820162596	6.880876e-01	-1.55743873	-1.05518034	-1.30630954
139	2.8023219470	6.931096e-01	0.05706970	0.48630711	0.27168841
140	2.8226276343	6.981315e-01	1.27936394	1.59083576	1.43509985
141	2.8429333217	7.031535e-01	1.61536195	1.88271288	1.74903741
142	2.8632390090	7.081755e-01	0.97523299	2.47973820	1.72748560
143	2.8835446964	7.131974e-01	0.60572315	0.93959533	0.77265924
144	2.9038503838	7.182194e-01	1.72298200	2.08090991	1.90194596
145	2.9241560711	7.232413e-01	1.83139126	2.57326543	2.20232835
146	2.9444617585	7.282633e-01	1.53530153	1.90645200	1.72087676
147	2.9647674458	7.332853e-01	-1.47584908	-1.07401172	-1.27493040
148	2.9850731332	7.383072e-01	8.07571667	9.19590810	8.63581239
149	3.0053788205	7.433292e-01	2.01360813	2.34890489	2.18125651
150	3.0256845079	7.483511e-01	1.49413862	1.91693724	1.70553793
151	3.0459901953	7.533731e-01	0.74891078	1.08010885	0.91450981
152	3.0662958826	7.583951e-01	-0.55778065	-0.11578664	-0.33678365
153	3.0866015700	7.634170e-01	-1.97022885	-1.50623951	-1.73823418
154	3.1069072573	7.684390e-01	-2.16240447	-1.66817977	-1.91529212
155	3.1272129447	7.734610e-01	0.14971686	0.57984681	0.36478184
156	3.1475186321	7.784829e-01	1.76177997	2.10924394	1.93551195
157	3.1678243194	7.835049e-01	1.45390019	1.83650201	1.64520110
158	3.1881300068	7.885268e-01	0.44908365	0.85646639	0.65277502
159	3.2084356941	7.935488e-01	0.58640207	0.92899537	0.75769872
160	3.2287413815	7.985708e-01	-1.30766294	-0.90058302	-1.10412298
161	3.2490470688	8.035927e-01	0.06118729	0.49068095	0.27593412
162	3.2693527562	8.086147e-01	-0.65307176	-0.26924154	-0.46115665
163	3.2896584436	8.136367e-01	-1.92779047	-1.40574082	-1.66676565
164	3.3099641309	8.186586e-01	1.50883903	1.87075710	1.68979806

165	3.3302698183	8.236806e-01	0.96041939	1.26282760	1.11162350
166	3.3505755056	8.287025e-01	0.43023015	0.78896826	0.60959920
167	3.3708811930	8.337245e-01	1.33226641	1.74535249	1.53880945
168	3.3911868803	8.387465e-01	2.16212366	2.43097445	2.29654906
169	3.4114925677	8.437684e-01	1.70287768	2.05626445	1.87957106
170	3.4317982551	8.487904e-01	2.60526336	2.96022333	2.78274335
171	3.4521039424	8.538124e-01	3.33827226	3.73074689	3.53450958
172	3.4724096298	8.588343e-01	1.59526637	1.91465109	1.75495873
173	3.4927153171	8.638563e-01	1.48340053	1.83415219	1.65877636
174	3.5130210045	8.688782e-01	-0.63499918	-0.25898870	-0.44699394
175	3.5333266919	8.739002e-01	0.88480019	1.21268057	1.04874038
176	3.5536323792	8.789222e-01	1.22943776	1.63519054	1.43231415
177	3.5739380666	8.839441e-01	-1.37463910	-0.93990764	-1.15727337
178	3.5942437539	8.889661e-01	0.97736206	1.27871956	1.12804081
179	3.6145494413	8.939881e-01	1.49062424	1.82943907	1.66003165
180	3.6348551286	8.990100e-01	-0.28058775	0.08790533	-0.09634121
181	3.6551608160	9.040320e-01	0.83017104	1.14077975	0.98547539
182	3.6754665034	9.090539e-01	1.19395477	1.59706720	1.39551099
183	3.6957721907	9.140759e-01	1.37319516	1.68593329	1.52956422
184	3.7160778781	9.190979e-01	3.39905205	3.84866166	3.62385686
185	3.7363835654	9.241198e-01	1.95720879	2.30925515	2.13323197
186	3.7566892528	9.291418e-01	0.19530742	0.59916445	0.39723593
187	3.7769949402	9.341638e-01	0.77105873	1.14402087	0.95753980
188	3.7973006275	9.391857e-01	2.07923183	2.34866092	2.21394637
189	3.8176063149	9.442077e-01	1.64238124	1.99460881	1.81849503
190	3.8379120022	9.492296e-01	-0.02619721	0.41394103	0.19387191
191	3.8582176896	9.542516e-01	0.22044321	0.60479600	0.41261961
192	3.8785233769	9.592736e-01	5.87858642	6.72430866	6.30144754
193	3.8988290643	9.642955e-01	-2.21810995	-1.71494088	-1.96652541
194	3.9191347517	9.693175e-01	0.04439806	0.48937852	0.26688829
195	3.9394404390	9.743395e-01	-1.51616426	-1.10893253	-1.31254839
196	3.9597461264	9.793614e-01	4.21908764	4.71531111	4.46719938
197	3.9800518137	9.843834e-01	-0.10321839	0.39033592	0.14355877
198	4.0003575011	9.894053e-01	-0.74588873	-0.33993249	-0.54291061
199	4.0206631885	9.944273e-01	1.55199554	1.91420672	1.73310113
200	4.0409688758	9.994493e-01	0.81154491	1.18838959	0.99996725

R-code:

```
dir<-c("/Users/Penny/Desktop/study /Fall 2016/stat bigdata/hw")
mydata<-read.table(file.path(dir,"mydata.txt"),header = TRUE)
#(1)write as a csv
write.csv(mydata, file = "/Users/Penny/Desktop/study /Fall 2016/stat bigdata/hw/mydata.csv")

#separate the data into two parts : training and test
set.seed(001)
n<-dim(mydata)[1]
train<-sample(seq(1:n),0.8*n)
mydata.train<-mydata[train,]
mydata.test<-mydata[-train,]


#(2)sample correaltion between the response and each of the covariates
cor(mydata)[1,]
```

```
cor(mydata)
pairs(~V1+V2+V3+V4+V5+V6, data = mydata)

#(3)
#fit an initial model to fit the dataset and check the goodness of fit
fit.initial<-glm(V1~.,data = mydata.train)
summary(fit.initial)
par(mfrow=c(2,2))
plot(fit.initial)

#fit the model on the test dataset
dev.off()
ytest.hat<-predict(fit.initial,newdata = mydata.test)
ytest<-mydata.test[,1]
rtest<-ytest.hat-ytest
par(mfrow=c(1,2))
plot(ytest.hat,rtest,ylab = "test error",main = "residual vs fitted value in test data set",cex=0.6)
qqnorm(rtest,main = "normal Q-Q plot in test data set",cex=0.6)
sqrt(sum(rtest^2)/(40-6))

#get the AIC for fit.intial on the whole dataset
fit.initial_prime<-glm(V1~.,data = mydata)
summary(fit.initial_prime)

#(3) refine the model
fit2<-glm(V1~V2+V4+V5+V6,data = mydata.train)
summary(fit2)
par(mfrow=c(2,2))
plot(fit2)
#on test dataset
ytest_hat<-predict(fit2,newdata = mydata.test)
r_test<-ytest_hat-ytest
par(mfrow=c(1,2))
plot(ytest_hat,r_test,ylab = "test error",main = "residual vs fitted value in test data set",cex=0.6)
qqnorm(r_test,main = "normal Q-Q plot in test data set",cex=0.6)
sqrt(sum(rtest^2)/(40-6))


#
#do model selecting by stepwise selecting method
library(MASS)
step <- stepAIC(fit.initial, direction="both")
stepAIC(fit.initial,direction = "backward")
step(fit.initial)


#by subset selection
library(leaps)
fit.subsets<-regsubsets(V1~.,data = mydata)
summary.fit.subsets<-summary(fit.subsets)
summary.fit.subsets
names(summary.fit.subsets)
summary.fit.subsets$rsq
```

```
par(mfrow=c(1,3))
plot(summary.fit.subsets$adjr2,xlab = "number of variables", ylab = "adjusted R_square",type = "l",
lty=2)
which.max(summary.fit.subsets$adjr2)
points(4,summary.fit.subsets$adjr2[4], col="red", cex=1,pch=15)
plot(summary.fit.subsets$cp,xlab = "number of variables", ylab = "Cp value",type = "l",lty=2)
which.min(summary.fit.subsets$cp)
points(4,summary.fit.subsets$cp[4], col="red", cex=1,pch=15)
plot(summary.fit.subsets$bic,xlab = "number of variables", ylab = "BIC",type = "l",lty=2)
which.min(summary.fit.subsets$bic)
points(3,summary.fit1$bic[3], col="red", cex=1, pch=15)
#by bic select three variables v2,v5,v6
fit.subsets_3v<-glm(V1~V2+V5+V6,data = mydata)
summary(fit.subsets_3v)
plot(fit.subsets_3v)
fit.subsets_2v<-glm(V1~V5+V6, data=mydata)
summary(fit.subsets_2v)
plot(fit.subsets_2v)
fit.subsets_1v<-glm(V1~V6, data=mydata)
summary(fit.subsets_1v)
plot(fit.subsets_1v)

fit.foward<-regsubsets(V1~., mydata, method = "forward")
fit.backward<-regsubsets(V1~., mydata, method = "backward")
summary(fit.foward)
summary(fit.backward)

#by lasso
#fit the tuning parameter first by cross validation
library(Matrix)
library(foreach)
library(glmnet)
x<-as.matrix(mydata[,-1],nrow=200,ncol=5)
y<-as.matrix(mydata[,1],nrow=200,ncol=1)
cv.out<-cv.glmnet(x,y,alpha=1,nfolds = 10)
bestlambda<-cv.out$lambda.min
bestlambda
dev.off()
plot(cv.out)
best.model<-glmnet(x,y,family = "gaussian",lambda = bestlambda)
predict(best.model,s=bestlambda,type = "coefficients")



#Fit polynomial models
#since the by subset model selection V2,V4,V5,V6 were selected
#select the best model when k=1, d=1,2,3,4,5 by cross validataion
dev.off()
library(boot)
degree=1:5
cv.error5=matrix(,nrow = 5,5)
set.seed(002)

for(d in degree){
  for (j in 1:5){
```

```
   glm.fit <- glm(mydata[,1]~poly(mydata[,j+1], degree=d),data = mydata)
   cv.error5[j,d] <- cv.glm(mydata,glm.fit,K=5)$delta[1]
  }
}
matplot(degree,cv.error5,type = "l",ylab = "cv errors", main = "cross validation error for different
degrees of polynomial regression model", cex.main=0.8)
row.names(cv.error5)<-c("d=1","d=2","d=3","d=4","d=5")
colnames(cv.error5)<-c("V2","V3","V4","V5","V6")
points(which.min(cv.error5),cv.error5[which.min(cv.error5)],col="red",cex=1,pch = 4)
cv.error5
#min cv.errors were obtained when degree=2, variable =V2
poly_v2<-glm(V1~poly(V2, degree=2), data=mydata)
summary(poly_v2)
par(mfrow=c(2,2))
plot(poly_v2)
v2lim = range(mydata$V2)
v2.grid = seq(from = v2lim[1], to = v2lim[2], length=200)
poly_v2.pred <- predict(poly_v2, list(V2 = v2.grid))
plot(V1 ~V2 , data = mydata, col = "darkgrey",main="fitted model V1~poly(V2,2)",cex=0.8)
lines(v2.grid, poly_v2.pred, col = "red", lwd = 2)



#fit k=2, covariates V5,V6 (since V5 V6 were selected by subsets regression)
set.seed(003)
cv.error<-rep(0,5)
for(d in degree){
  glm2.fit <- glm(V1~poly(V5,V6, degree=d),data = mydata)
  cv.error[d] = cv.glm(mydata,glm2.fit,K=10)$delta[1]
}
plot(degree,cv.error,type = "l",ylab = "cv errors", main = "cross validation error for different degrees
of polynomial regression model", cex.main=0.8)
cv.error
glm2.fit<-glm(V1~poly(V5,V6, degree=3),data = mydata)
summary(glm2.fit)
par(mfrow=c(2,2))
plot(glm2.fit)
#3d scatterplot:
library(plot3D)
library(scatterplot3d)
par(mfrow=c(1,2))
glm2.fit.pred<-predict(glm2.fit,list(V5=mydata$V5,V6=mydata$V6))
plot(mydata$V1,glm2.fit.pred,main = "fitted value vs true value", xlab = "true v1", ylab = "fitted
value")
x<-mydata$V5
y<-mydata$V6
z<-mydata$V1
my.3d<-scatterplot3d(x,y,z,angle = 45,xlab = "V5",ylab = "V6",zlab = "V1")
my.3d$points3d(x,y,glm2.fit.pred,col="blue",cex=0.5,pch=4)

#predition errors:
v5lim<-range(mydata$V5)
v5.grid<- seq(from = v5lim[1], to = v5lim[2], length=200)
v6lim<-range(mydata$V6)
v6.grid<- seq(from = v6lim[1], to = v6lim[2], length=200)
```

```
glm2.fit.pred <- predict(glm2.fit, list(V5 = v5.grid, V6=v6.grid))
par(mfrow=c(1,2))
plot(V1 ~V5,data=mydata,main="fitted value vs true V1")
lines(v5.grid, glm2.fit.pred, col = "red", lwd = 2)
plot(V1~V6,data = mydata,main="fitted value vs true v1")
lines(v6.grid,glm2.fit.pred,col="red",lwd=2)

# fit k=3, with covariables: V2,V5,V6
set.seed(004)
cv.error_3<-rep(0,5)
for(d in degree){
 glm3.fit <- glm(V1~poly(V2,V5,V6, degree=d),data = mydata)
 cv.error_3[d] = cv.glm(mydata,glm3.fit,K=5)$delta[1]
}
plot(degree,cv.error_3,type = "l",ylab = "cv errors", main = "cross validation error for different
degrees of polynomial regression model", cex.main=0.8)
points(which.min(cv.error_3),cv.error_3[which.min(cv.error_3)],col="red",cex=1,pch = 4)
cv.error_3

#the differences of cv_error for d=2 vs d=3
set.seed(005)
cv.error_2v3<-matrix(,ncol = 5,nrow = 100)
for (i in 1:100){
 for(d in degree){
  glm3.fit <- glm(V1~poly(V2,V5,V6, degree=d),data = mydata)
  cv.error_2v3[i,d] = cv.glm(mydata,glm3.fit,K=5)$delta[1]
 }
}
dif<-cv.error_2v3[,3]-cv.error_2v3[,2]
summary(dif)
plot(seq(1:100),dif)

# fit with d=2 vs d=3
glm3.fit<-glm(V1~poly(V2,V5,V6, degree=3),data = mydata)
summary(glm3.fit)
par(mfrow=c(2,2))
plot(glm3.fit)
glm3.fit_d2<-glm(V1~poly(V2,V5,V6, degree=2),data = mydata)
summary(glm3.fit_d2)
par(mfrow=c(2,2))
plot(glm3.fit_d2)

#fitted value vs true value:
glm3.fit_d2.pred<-predict(glm3.fit_d2,list(V2=mydata$V2,V5=mydata$V5,V6=mydata$V6))
plot(mydata$V1,glm3.fit_d2.pred,main = "fitted value vs true value", xlab="true v1", ylab = "fitted
value")

#fit a polynomial model with V2 V4 V5 V6
set.seed(005)
cv.error_4<-rep(0,5)
for(d in degree){
 glm4.fit <- glm(V1~poly(V2,V4,V5,V6, degree=d),data = mydata)
 cv.error_4[d] = cv.glm(mydata,glm4.fit,K=5)$delta[1]
}
```

```
plot(degree,cv.error_4,type = "l",ylab = "cv errors", main = "cross validation error for different
degrees of polynomial regression model", cex.main=0.8)
points(which.min(cv.error_4),cv.error_4[which.min(cv.error_4)],col="red",cex=1,pch = 4)
cv.error_4
glm4.fit<-glm(V1~poly(V2,V4,V5,V6, degree=which.min(cv.error_4)),data = mydata)
summary(glm4.fit)
par(mfrow=c(2,2))
plot(glm4.fit)



#6 final model:
#best model selected by polynomial with degree 3, with variable V5,V6 :
best.fit1<-glm(V1~poly(V6,degree=3)+V5+V5:V6,data = mydata)
summary(best.fit1)
par(mfrow=c(2,2))
plot(best.fit1)


#polynoimal  with degree 2 and variable v2,v5,v6
best.fit2<-glm(V1~poly(V2,degree = 2)+V2:V5+V2:V6+poly(V6,degree=2), data = mydata)
summary(best.fit2)

#comparing the two best model candidates, best.fit1 is the final fit.
#construct confidence intervals for parameters
beta<-best.fit1$coefficients[-1]
t.critical<-qt(0.025,df=200-6,lower.tail = FALSE)
std<-c(0.88671,0.74005,0.68621,0.16366,0.28466)
ci<-cbind(beta-t.critical*std,beta+t.critical*std)
colnames(ci)<-c("lower.ci","upper.ci")
ci

pred.fit<-predict(best.fit1,list=c(V5=v5.grid,V6=v6.grid),se.fit = TRUE)
pred.fit$se.fit
ci_v1<-cbind.data.frame(v5.grid,v6.grid,pred.fit$fit - 1.96*pred.fit$se.fit,pred.fit$fit +
1.96*pred.fit$se.fit,pred.fit$fit)
colnames(ci_v1)<-c("v5.grid","v6.grid","lower ci","upper ci","fitted value")
my.pred.3d<-scatterplot3d(v5.grid,v6.grid,pred.fit$fit,angle = 45,xlab = "V5",ylab = "V6",zlab = "V1")
my.pred.3d$points3d(v5.grid,v6.grid,ci_v1$`lower ci`,col="blue",cex=0.5,pch=4)
my.pred.3d$points3d(v5.grid,v6.grid,ci_v1$`upper ci`,col="red",cex=0.5,pch=4)
```