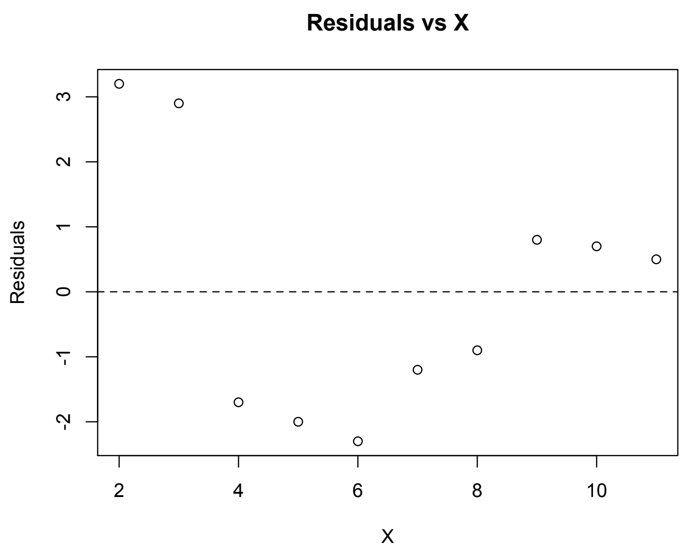
# STATS 203 - HW #3 SOLUTION

COURTESY TO PATRICK LEAHY

1. R code:

```
> elec.resid = read.table("http://www-stat.stanford.edu/~nzhang/203_web/
   Data/ElectricityConsumption.txt", header=T)
> plot(elec.resid, main="Residuals vs X")
> abline(a=0, b=0, lty="dashed")
```




As was the case the bacteria data we looked at in class, there is a clear pattern to the distribution of the residuals, which are positive for more extreme values of X and negative for values closer to the median. A nonlinear transformation such as quadratic or logarithmic might alleviate the problem. Also it seems that the residuals have a heteroscedasticity problem. The absolute value of residuals decreases as X increases. If this is the case, a nonlinear transformation might not be helpful. Instead we may use WLS to fix the problem.

2. (RABE 7.4) R code is given below. Note that, as in the book, we omit Alaska from the data.

```
> edu.data = read.table("http://www-stat.stanford.edu/~nzhang/203_web/
```

```
   Data/EducationExpenditure.txt", header=T)
> edu.data = edu.data[-49,]     # remove Alaska
> attach(edu.data)
> Region = factor(Region)
> edu.lm = lm(Y~X1+X2+X3+Region)
> summary(edu.lm)

Call:
lm(formula = Y ~ X1 + X2 + X3 + Region)

Residuals:
    Min      1Q  Median      3Q 
-74.539 -20.940  -2.867  18.556 
    Max 
 86.766 

Coefficients:
              Estimate Std. Error
(Intercept) -168.03880  147.90029
X1             0.04363    0.01413
X2             0.65703    0.36647
X3             0.04806    0.05278
Region2       -4.15441   16.47796
Region3      -12.40588   16.51665
Region4       17.32351   17.50721
            t value Pr(>|t|)   
(Intercept)  -1.136  0.26233   
X1            3.088  0.00357 **
X2            1.793  0.08020 . 
X3            0.910  0.36779   
Region2      -0.252  0.80218   
Region3      -0.751  0.45677   
Region4       0.990  0.32808   
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 35.45 on 42 degrees of freedom
Multiple R-squared: 0.5396, Adjusted R-squared: 0.4738
F-statistic: 8.204 on 6 and 42 DF,  p-value: 6.709e-06
```

The simple linear regression model for these data is

$$Y = -168.0388 + 0.0436X_1 + 0.6570X_2 + 0.0481X_3 - 4.1544I_2 - 12.4059I_3 + 17.3235I_4,$$

where $I_i = 1$ if the state is in region $i$ and 0 otherwise. The weighted-least-squares model found in Section 7.4 is

$$Y_{WLS} = -316.024 + 0.062X_1 + 0.874X_2 - 0.029X_3.$$

The simple OLS with region indicator ($R^2 = 0.5396$, $\hat{\sigma} = 35.45$) has a higher $R^2$ value and a lower residual standard error than WLS ($R^2 = 0.477$, $\hat{\sigma} = 36.52$), so with respect to these indicators it fits the data better.

We can use a nested F-test to test the hypothesis $H_0 : I_2 = I_3 = I_4 = 0$ against $H_a$: the regressions vary by region:

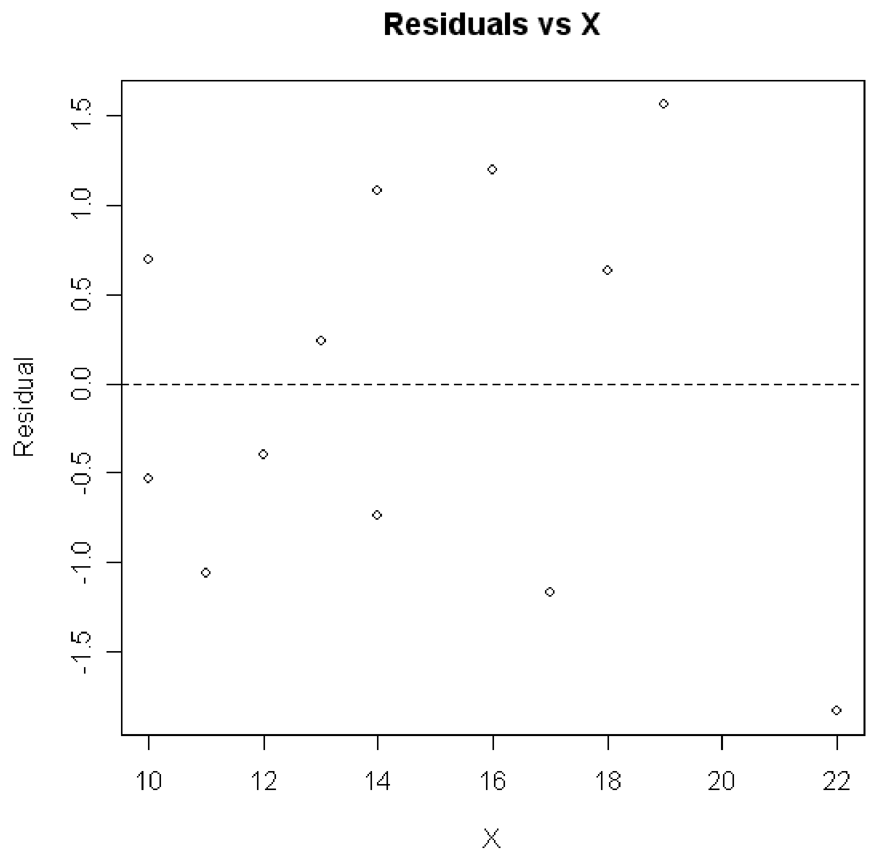
```
> anova(edu.lm, lm(Y~X1+X2+X3))
Analysis of Variance Table

Model 1: Y ~ X1 + X2 + X3 + Region
Model 2: Y ~ X1 + X2 + X3
  Res.Df   RSS Df Sum of Sq      F
1     42 52782
2     45 57700 -3     -4918 1.3045
  Pr(>F)
1
2 0.2856
```

The test produces a F-statistic of 1.3045 and a corresponding p-value of 0.2856, which is not large enough to reject the null hypothesis at a significance level of even 10%. We conclude that the regressions do not vary significantly by region.

3. (a) R code:

```
> cal.data = read.table("http://www-stat.stanford.edu/~nzhang/
   203_web/Data/ComputerAssistedLearning.txt", header=T)
> attach(cal.data)
> cal.lm = lm(Y~X)
> cal.lm

Call:
```

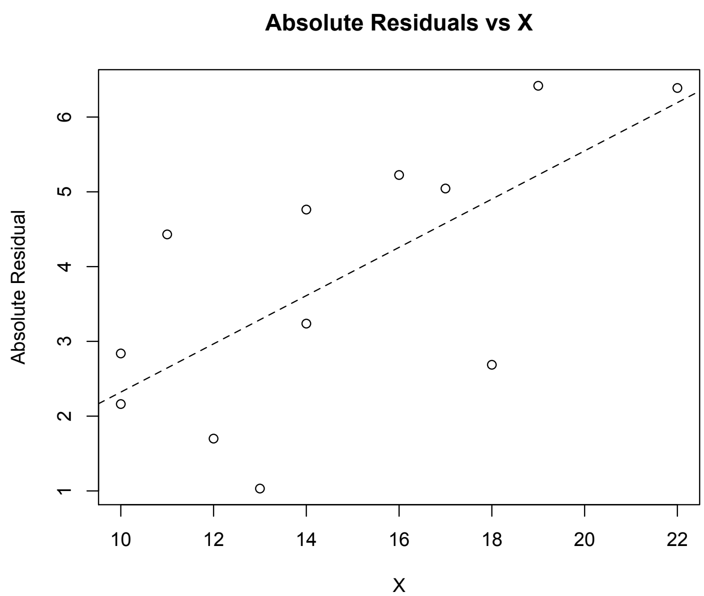
```
lm(formula = Y ~ X)

Coefficients:
(Intercept)            X
     19.473        3.269

> plot(X, resid(cal.lm), main="Residuals vs X", ylab="Residual")
> abline(0, 0, lty="dashed")
> plot(X, abs(resid(cal.lm)), main="Abs Residuals vs X", ylab="Abs
Residual")
```




The plot has a slight rightward-opening fan shape, which suggests that the variance of the error terms increases with $X$. This lead us to infer that the variance is not constant and that the variance increases as X increases.

(b) R code:

```
> plot(X, abs(rstandard(cal.lm)), main="Absolute Residuals vs X",
   ylab="Absolute Residual")
> abline(lm(abs(rstandard(cal.lm))~X), lty="dashed")
```




We can see even more clearly from this plot that the standard deviation of the error terms increases with $X$.

(c) R code:

```
> cal.resid=abs(rstandard(cal.lm))
> resid.lm = lm(cal.resid~X)
> W=1/(fitted(resid.lm))^2
```

As the weight decrease with X, $X = 10$ (case 4,7) has the largest weight and $X = 22$(case 3) has the smallest weight.

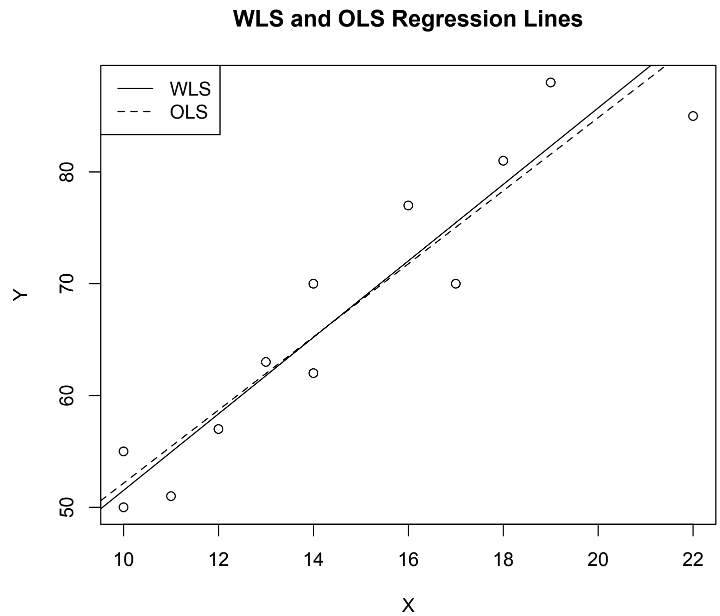
```
> cal.wls1 = lm(Y~X, weights=W)
> summary(cal.wls1)

Call: lm(formula = Y ~ X, weights = W)

Residuals:
    Min      1Q  Median      3Q     Max
-6.5860 -3.9583 -0.1942  4.4173  6.9095
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  17.2704     4.8020   3.597  0.00488 **
          X 3.4233 0.3774  9.070 3.86e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.903 on 10 degrees of freedom Multiple
R-Squared: 0.8916,     Adjusted R-squared: 0.8808 F-statistic: 82.27
on 1 and 10 DF,  p-value: 3.859e-06
```

The weighted least squares estimates of the parameters are $b_{w0} = 17.27$ and $b_{w1} = 3.4$.

(d) The WLS estimate of $\beta_0$ is slightly smaller than the OLS estimate (19.473); the WLS estimate of $\beta_1$ is slightly larger than the OLS estimate (3.269). The two regression lines are shown on the graph below, the code for which is also given.
Note, here we use the standardized residuals $y_i - \hat{y}_i^{(-i)}$ to protect us from possible outliers.

```
> plot(X,Y,main="WLS and OLS Regression Lines")
> abline(cal.wls1)
> abline(cal.lm,lty="dashed")
> legend(x="topleft",legend=c("WLS","OLS"),lty=c("solid","dashed"))
```




(e)

```
resid2.lm = lm(abs(rstandard(cal.wls1))~X)

W2 = 1/(fitted(resid2.lm))^2

cal.wls2 = lm(Y~X, weights=W2) summary(cal.wls2) Call: lm(formula =
Y ~ X, weights = W2)

Residuals:
    Min      1Q  Median      3Q     Max
-7.2350 -3.7895 -0.4768  3.6966  7.6867

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  20.3633     5.7305   3.553  0.00524 **

         X 3.2087     0.3682   8.715 5.52e-06 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 5.118 on 10 degrees of freedom Multiple
R-Squared: 0.8836,     Adjusted R-squared: 0.872 F-statistic: 75.95
on 1 and 10 DF,  p-value: 5.523e-06
```

The estimated regression coefficients are different from the previous iteration, so more iteration may be required here to ensure convergence.

4. R code (including a short function written to generate random normal quantiles):

```
> # randnorm: generates n probabilities (up to the specified number of
> # decimal places) and finds the corresponding normal quantiles
>
> randnorm = function(n, places){
+    values = matrix(nrow=n, ncol=1)
+    for(i in 1:n){
+       q = sample(0:99,1)
+       for(j in 1:(places-2)){
+          q = q+.1^j*sample(0:9,1)
+       }
+       values[i] = qnorm(q/100)
+    }
+    return(values)
+ }
>
> X = matrix(nrow=200,ncol=3)
> X[,1] = randnorm(200,10)
> X[,2] = X1 + 0.001*randnorm(200,10)
> X[,3] = 10*randnorm(200,10)
>
> eigen(cov(X))$vectors[,1]
[1] 0.001995757 0.001991666 0.999996025
```

The principal eigenvector of the covariance matrix is (0.001996, 0.01992, 0.999996), which is very close to the unit vector along the $X_3$ axis, the direction of maximum variation.

```
> eigen(cor(X))$vectors[,1]
[1] 0.70688692 0.70688647 0.02494796
```

The principal eigenvector of the correlation matrix is (0.706887, 0.706887, 0.024948), which reflects the close correlation of $X_1$ and $X_2$ and essentially ignores the uncorrelated component $X_3$.

5. (Note: I broke up the problem into two parts, each implementing a different model selection procedure.)

   (a) I first use the leaps() function to implement all-subsets regression. The model consists, initially, of the 19 variables. The year 1996 record is not considered in this analysis

```
> library(leaps)
> election.data2 = election.data[-21,]   # remove 1996
> attach(election.data2)
> full = lm(formula = V ~ I + D + W + G + P + N + I:D + I:G + I:P
+ I:N + D:W + D:G + D:P + D:N + W:G + G:P + G:N + P:N + W:N)
> full

Call:
lm(formula = V ~ I + D + W + G + P + N + I:D + I:G + I:P + I:N
+ D:W + D:G + D:P + D:N + W:G + G:P + G:N + P:N + W:N)

Coefficients:
(Intercept)            I            D
  0.4868376   -0.2029945    0.2664057
          W            G            P
 -0.6660692   -0.0184079    0.0006601
          N          I:D          I:G
 -0.0057641    0.0034679    0.0009166
        I:P          I:N          D:W
  0.0111918    0.0279627    0.7721145
        D:G          D:P          D:N
  0.0070711   -0.0241844   -0.0203815
        W:G          G:P          G:N
 -0.0553465    0.0020937    0.0012948
        P:N          W:N
  0.0008779           NA
```

```
> # select variables
> X = model.matrix(full)[,-c(1,20)]
> election.lps=leaps(x=X,y=V,nbest=5,method="Cp")
> election.best = election.lps$which[which((election.lps$Cp ==
  min(election.lps$Cp))),]
> election.best
```

```
    1     2     3     4     5     6
FALSE  TRUE FALSE FALSE FALSE FALSE
    7     8     9     A     B     C
FALSE  TRUE  TRUE FALSE FALSE FALSE
    D     E     F     G     H     I
FALSE FALSE  TRUE FALSE FALSE FALSE

> # calculate parameters
> election.lps.lm = lm(V~D+I:G+I:P+G:P)
> election.lps.lm

Call:
lm(formula = V ~ D + I:G + I:P + G:P)

Coefficients:
(Intercept)            D          I:G
  0.4731217    0.0722815    0.0074844
        I:P          G:P
 -0.0057950   -0.0002734

 > # predict 1996 result
 > predict.lm(election.lps.lm,newdata=election.data[21,])
       21
0.5522903
```

All-subsets regression thus yields the model

$$V = 0.4731 + 0.0723D + 0.0075I : G - 0.0058I : P - 0.0003G : P.$$

For the 1996 election, it predicts that $V = 0.5523$ (actual value was 0.5474).

(b) I now repeat the variable selection process using step-wise selection. The starting point is set to be V on I+D+W+P+N+G

```
step(lm(V ~ I + D + W + G + P + N),scope=list(upper=~I + D + W + G +
P + N + I:D + I:G + I:P+ I:N + D:W + D:G + D:P + D:N + W:G + G:P +
G:N + P:N + W:N, lower=~1), direction="both")

Call: lm(formula = V ~ I + D + G + P + N + I:G + I:N + D:P)

Coefficients: (Intercept)            I            D            G
```

```
P            N
   0.502645    -0.074419     0.084288     0.001657     0.003251    -0.008120
        I:G          I:N          D:P
   0.008924     0.009207    -0.006799
```

The final model chosen by step-wise selection is $V\ I + D + G + P + N + I : G + I : N + D : P$. For the 1996 election, it predicts that $V = 0.5382$ with confidence interval [0.4683118 0.6088935](actual value was 0.5474).

(c) The multi-collinearity has a negative impact on the step-wise selection as well as the subset . From the correlation matrix of predictors, we find that I and D are highly correlated. From the vif we computed for the full model (19 predictors), we find that the multi-collinearity problem exist in this data set. It would be wise to remove some of the predictors based on the vif. For example, remove the predictors with largest vif and refit the model. Stop until all predictors have vif under 10.

```
> vif(full)
          I           D           W           G           P           N
1603.436345  976.516701         Inf  227.799597  250.794629
57.034367
        I:D         I:G         I:P         I:N         D:W         D:G
   3.645741  233.375731  519.110353 1887.037276         Inf  177.525837
        D:P         D:N         W:G         G:P         G:N         P:N
 350.956244 1313.798192         Inf  500.118713   73.094180  305.029093
        W:N
        Inf
```

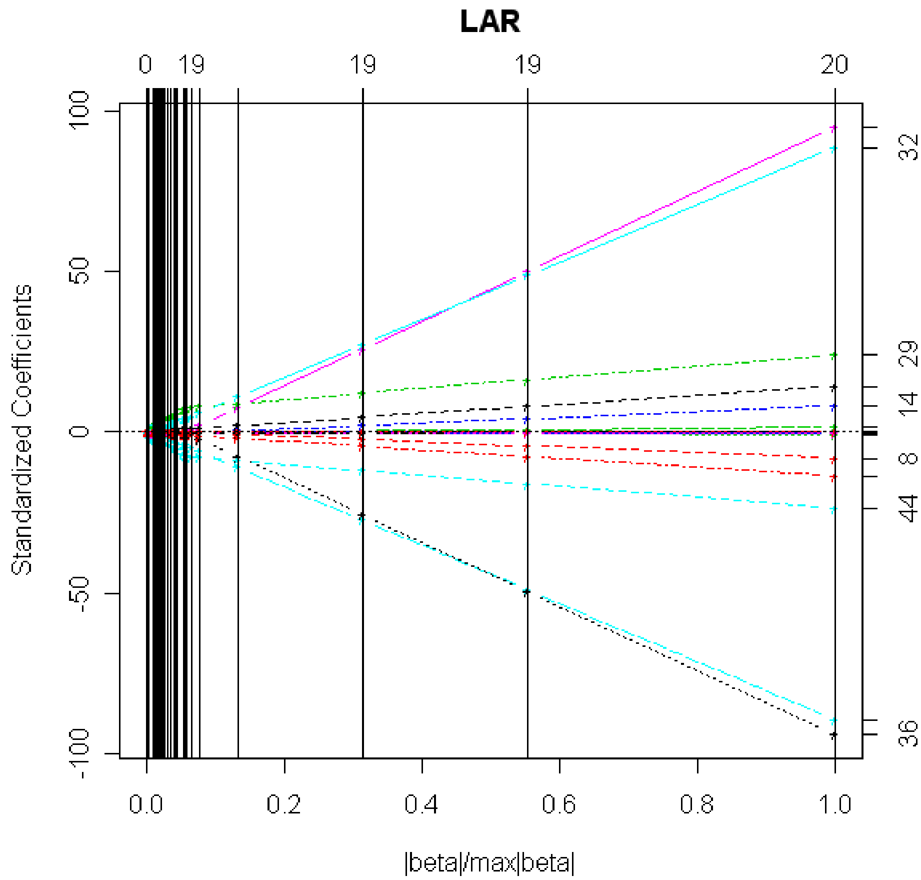
6. (a) The LARS algorithm finds the values of $\beta_0, ..., \beta_p, \lambda$ that minimize the quantity

$$\sum_{i=1}^{n}\left(Y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

(b) The order in which the variables are added can be seen below:

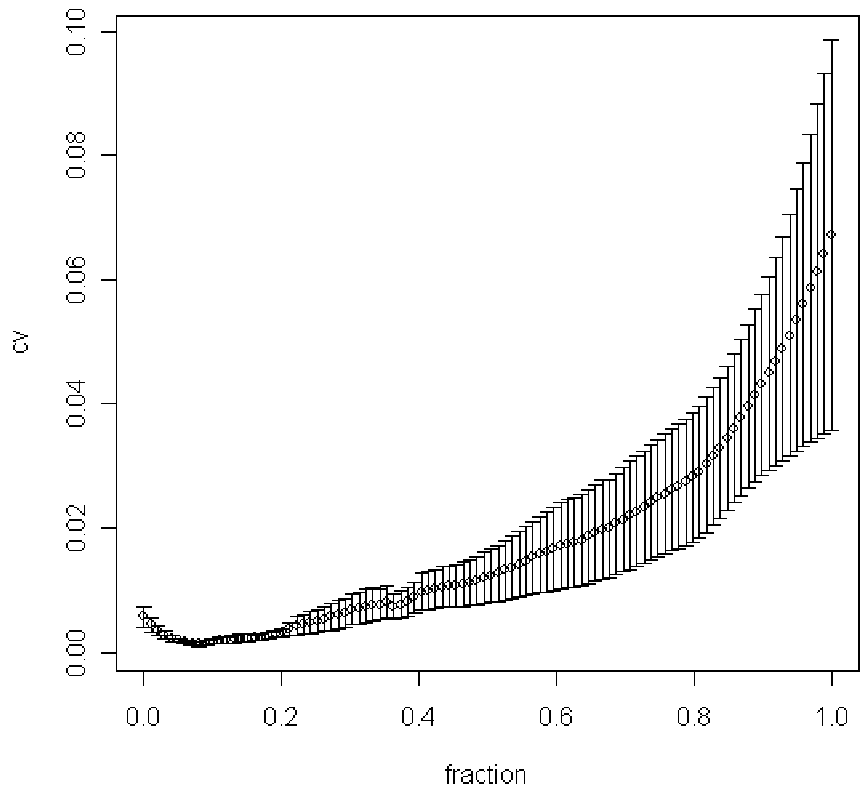
```
library(lars)
X=model.matrix(lm(V~.^3,data=data.frame(election.data)))
election.lars=lars(X,V,type="lar",trace=TRUE)

LARS Step 0 :    1 Variables with Variance < eps; dropped for good
```

```
Computing X'X ..... LARS

Step 1 :    Variable 32    added LARS

Step 2 :    Variable 53 added LARS

Step 3 :    Variable 38    added LARS

Step 4 : Variable 17    added LARS

Step 5 :    Variable 29 added LARS

Step 6 :    Variable 36    added LARS

Step 7 : Variable 2     added LARS

Step 8 :    Variable 44 added LARS

Step 9 :    Variable 8     added LARS

Step 10 : Variable 33 added LARS

Step 11 :   Variable 23 added LARS

Step 12 : Variable 12 added LARS

Step 13 :   Variable 26 added LARS

Step 13 : Variable 51 collinear; dropped for good LARS

Step 14 : Variable 54    added LARS

Step 15 : Variable 56    added LARS

Step 16 : Variable 30    added LARS

Step 17 :   Variable 43 added LARS

Step 18 :   Variable 42 added LARS

Step 19 : Variable 14 added LARS

Step 20 :   Variable 21 added Computing residuals, RSS etc .....
```




(c) R code:

```
election.lar.cv=cv.lars(X,V)

election.lar.cv$fraction[which.min(election.lar.cv$cv)]

election.lars$lambda[which.min(election.lar.cv$cv)]
predict.lars(election.lars, s=0.03259, type = "coefficients", mode =
"lambda")
```




The cross validation gives us $\lambda = 0.034$. The coefficient for predictors are as follows. And the final model is $V\ Year + N + I : G + P : N + Year : I : G + Year : P : N + I : G : N$. It is quite different from the model we fit in the previous problem, partly because we are searching models in a larger model spaces.

```
$coefficients
  (Intercept)          Year             I             D             W
 0.000000e+00  4.139045e-04  0.000000e+00  0.000000e+00  0.000000e+00
            G             P             N        Year:I        Year:D
 0.000000e+00  0.000000e+00 -1.077127e-06  0.000000e+00  0.000000e+00
       Year:W        Year:G        Year:P        Year:N           I:D
 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
          I:W           I:G           I:P           I:N           D:W
 0.000000e+00  1.200309e-01  0.000000e+00  0.000000e+00  0.000000e+00
          D:G           D:P           D:N           W:G           W:P
```

```
 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
          W:N           G:P           G:N           P:N      Year:I:D
 0.000000e+00  0.000000e+00  0.000000e+00  1.164033e-02  0.000000e+00
     Year:I:W      Year:I:G      Year:I:P      Year:I:N      Year:D:W
 0.000000e+00 -5.818618e-05  0.000000e+00  0.000000e+00  0.000000e+00
     Year:D:G      Year:D:P      Year:D:N      Year:W:G      Year:W:P
 1.162116e-06  0.000000e+00  2.778501e-06  0.000000e+00  0.000000e+00
     Year:W:N      Year:G:P      Year:G:N      Year:P:N         I:D:W
 0.000000e+00  0.000000e+00  0.000000e+00 -6.113092e-06  0.000000e+00
        I:D:G         I:D:P         I:D:N         I:W:G         I:W:P
 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
        I:W:N         I:G:P         I:G:N         I:P:N         D:W:G
 0.000000e+00  0.000000e+00 -2.322260e-04  0.000000e+00  0.000000e+00
        D:W:P         D:W:N         D:G:P         D:G:N         D:P:N
 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
        W:G:P         W:G:N         W:P:N         G:P:N
 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
```