Variable Selection in Low and High-Dimensional Data Analysis

May 15, 2016

Workshop Description

1. Petty 219 from 9:00AM-12:30PM
2. Presentation I (9:00AM-10:00PM)
3. Presentation II (10:20AM-11:20AM)
4. Presentation III (11:40AM-12:30PM)

Outline

1. Presentation I
   - R introduction
   - Linear regression model
2. Presentation II
   - Model selection in regression
   - Variable selection (VS) in linear regression
   - Techniques and tools for VS in low-dimensional linear regression
   - Case studies
3. Presentation III
   - High-dimensional data
   - Variable selection (VS) in high-dimensional (HD) data
   - Techniques and tools for VS in HD linear regression
   - Case studies

Presentation I

- R introduction
- Linear regression model

R introduction

- Environment for statistical computing and graphics
- Free software
- Associated with simple programming language
- R website: `www.r-project.org`
- RStudio website: `www.rstudio.com`
- Versions of R exist of Windows, MacOS, Linux and various other Unix flavors

R Function Libraries

- Implement many common statistical procedures
- Provide excellent graphics functionality
- A convenient starting point for many data analysis projects

R Programming Language

- Interpreted language
- To start, we will review
  - Syntax and common constructs
  - Function definitions
  - Commonly used functions

Interactive R

- R defaults to an interactive mode
- A prompt ">" is presented to users
- Each input expression is evaluated ···
- ··· and a result returned

R Help System

- R has a built-in help system with useful information and examples
- `help()` or `?` provides general help
- `help(plot)` will explain the `plot` function
- `help.search("histogram")` will search for topics that include the word histogram
- `example(plot)` will provide examples for the plot function

Managing Workspaces

- As you generate functions and variables, these are added to your current workspace
- Use `ls()` to list workspace contents and `rm()` to delete variables or functions
- When you quit, with the `q()` function, you can save the current workspace for later use

R as a Calculator

```
> 1 + 1 # Simple Arithmetic
[1] 2
> 2 + 3 * 4 # Operator precedence
[1] 14
> 3 ^ 2 # Exponentiation
[1] 9
> exp(1) # Basic mathematical functions are available
[1] 2.718282
> sqrt(10)
[1] 3.162278
> pi # The constant pi is predefined
[1] 3.141593
> 2*pi*6378 # Circumference of earth at equator (in km)
[1] 40074.16
```

Variables in R

- Numeric
- Logical (T or F)
- Strings, Sequences of characters
- Type determined automatically when variable is created with “$< -$” operator

R as a smart calculator

```
> x <- 1 # Can define variables
> y <- 3 # using "<-" operator to set values
> z <- 4
> x * y * z
[1] 12
> X * Y * Z # Variable names are case sensitive
Error: Object "X" not found
> This.Year <- 2004 # Variable names can include period
> This.Year
[1] 2004
```

R does a lot more

- Definitely not just a calculator
- R thrives on vectors
- R has many built-in statistical and graphing functions

R vectors

- A series of numbers
- Created with
  - `c()` to concatenate elements or sub-vectors
  - `rep()` to repeat elements or patterns
  - `seq()` or m:n to generate sequences
- Most mathematical functions and operators can be applied to vectors
  - Without loops!

Defining Vectors

```
> rep(1,10) # repeats the number 1, 10 times
[1] 1 1 1 1 1 1 1 1 1 1
> seq(2,6) # sequence of integers between 2 and 6
[1] 2 3 4 5 6 # equivalent to 2:6
> seq(4,20,by=4) # Every 4th integer between 4 and 20
[1] 4 8 12 16 20
> x <- c(2,0,0,4) # Creates vector with elements 2,0,0,4
> y <- c(1,9,9,9)
> x + y # Sums elements of two vectors
[1] 3 9 9 13
> x * 4 # Multiplies elements
[1] 8 0 0 16
> sqrt(x) # Function applies to each element
[1] 1.41 0.00 0.00 2.00 # Returns vector
```

Accessing Vector Elements

- Use the [] operator to select elements
- To select specific elements:
  - Use index or vector of indexes to identify them
- To exclude specific elements:
  - Negate index or vector of indexes
- Alternative:
  - Use vector of T and F values to select subset of elements

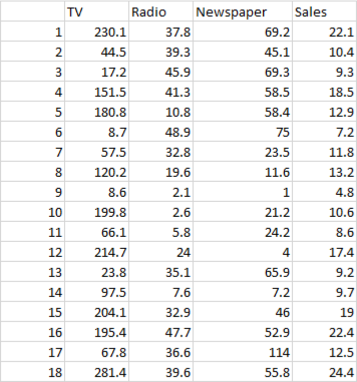
Accessing Vector Elements

```
> x <- c(2,0,0,4)
> x[1] # Select the first element, equivalent to x[c(1)]
[1] 2
> x[-1] # Exclude the first element
[1] 0 0 4
> x[1] <- 3 ; x
[1] 3 0 0 4
> x[-1] = 5 ; x
[1] 3 5 5 5
> y < 9 # Compares each element, returns result as vector
[1] TRUE FALSE FALSE FALSE
> y[4] = 1
> y < 9
[1] TRUE FALSE FALSE TRUE
> y[y<9] = 2 # Edits elements marked as TRUE in index vector
> y
[1] 2 9 9 2
```

Data Frames

- Group a collection of related vectors
- Most of the time, when data is loaded, it will be organized in a data frame
- Let's look at an Advertising dataset

Advertising example



	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
6	8.7	48.9	75	7.2
7	57.5	32.8	23.5	11.8
8	120.2	19.6	11.6	13.2
9	8.6	2.1	1	4.8
10	199.8	2.6	21.2	10.6
11	66.1	5.8	24.2	8.6
12	214.7	24	4	17.4
13	23.8	35.1	65.9	9.2
14	97.5	7.6	7.2	9.7
15	204.1	32.9	46	19
16	195.4	47.7	52.9	22.4
17	67.8	36.6	114	12.5
18	281.4	39.6	55.8	24.4

```
>setwd("C:\\Users\\x_gao2\\Dropbox\\Teaching\\QMS\\Dataset")
>Advertising = read.csv("Advertising.csv",row.names=1)
```

Setting Up Data Sets

- Load from a text file using `read.table()`
- Parameters `header`, `sep`, and `na.strings` control useful options
- `read.csv()` and `read.delim()` have useful defaults for comma or tab delimited files
- Create from scratch using `data.frame()`
    - Example:

```
data.frame(height=c(150,160),
weight=(65,72))
```

Blood Pressure Data Set

-
  ```
  HEIGHT WEIGHT WAIST HIP BPSYS BPDIA
  172 72 87 94 127.5 80
  166 91 109 107 172.5 100
  174 80 95 101 123 64
  176 79 93 100 117 76
  166 55 70 94 100 60
  163 76 96 99 160 87.5
  ...
  ```
- Read into R using:

  ```
  bp <-
  read.table("bp.txt",header=T,na.strings=c("x"))
  ```

Accessing Data Frames

- Multiple ways to retrieve columns
- The following all retrieve weight data:
  - `> bp["WEIGHT"]`
  - `> bp[,2]`
  - `>bp$WEIGHT`
- The following excludes weight data:
  - `>bp[,-2]`

Basic Utility Functions

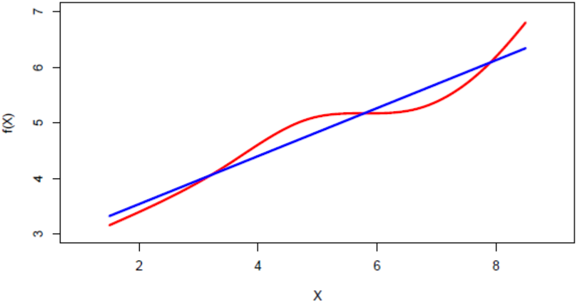
- `length()` returns the number of elements
- `mean()` returns the sample mean
- `median()` returns the sample mean
- `range()` returns the largest and smallest values
- `unique()` removes duplicate elements
- `summary()` calculates descriptive statistics
- `diff()` takes difference between consecutive elements
- `rev()` reverses elements

Linear regression model

- Linear regression is a simple approach to supervised learning
- One variable of interest: Dependent variable, Y
- A few variables may influence Y: Predictors, X_1, $X_2, \cdots, X_p$
- Assume the dependence of Y on predictors to be linear

Linear regression model

- True regression functions are never linear!



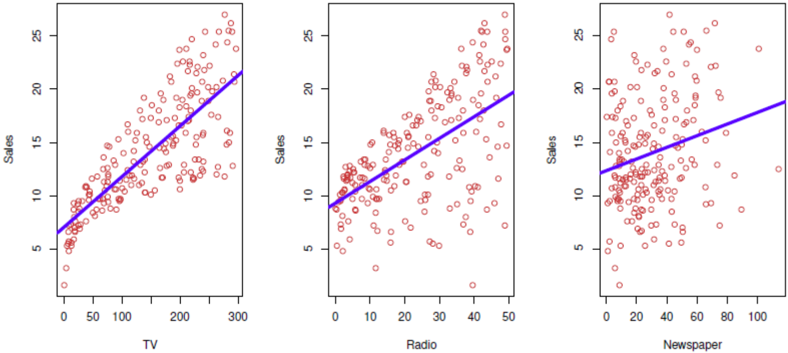
- Although it may seem overly simplistic, linear regression is extremely useful both conceptually and practically.

Linear regression for the advertising data

Consider the advertising data shown on the next slide. Questions we might ask:

- Which media contribute to sales?
- How accurately can we predict future sales?
- Is the relationship linear?

Advertising data



- Sales (in thousands of units) for a particular product as a function of advertising budgets (in thousands of dollars) for TV, radio, and newspaper media.

Advertising data

```
> summary(Advertising)
       TV             Radio          Newspaper          Sales
 Min.   :  0.70   Min.   : 0.000   Min.   :  0.30   Min.   : 1.6
 1st Qu.: 74.38   1st Qu.: 9.975   1st Qu.: 12.75   1st Qu.:10.3
 Median :149.75   Median :22.900   Median : 25.75   Median :12.9
 Mean   :147.04   Mean   :23.264   Mean   : 30.55   Mean   :14.0
 3rd Qu.:218.82   3rd Qu.:36.525   3rd Qu.: 45.10   3rd Qu.:17.4
 Max.   :296.40   Max.   :49.600   Max.   :114.00   Max.   :27.0
```

Simple linear regression using a single predictor X

- We assume a model

  $$Y = \beta_0 + \beta_1 X + \varepsilon$$

  where β_0 and β_1 are two unknown constants that represent the intercept and slope, also known as coefficients or parameters, and ε is the error term.
- Given some estimates $\widehat{\beta}_0$ and $\widehat{\beta}_1$ for the model coefficients, we predict future sales using

  $$\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x$$

  where $\widehat{y}$ indicates a prediction of Y on the basis of $X = x$. The hat symbol denotes an estimated value.

Estimation of the parameters by least squares

- Let $\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$ be the prediction for Y based on the ith value of X.
- $e_i = y_i - \widehat{y}_i$ represents the ith residual.
- Define the residual sum of squares (RSS) as

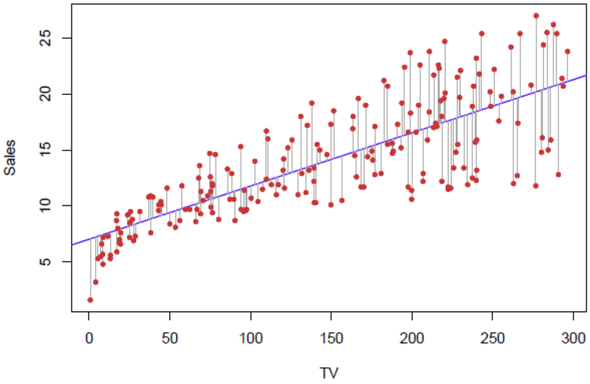
$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2$$

  or equivalently as

$$RSS = (y_1 - \widehat{\beta}_0 - \widehat{\beta}_1 x_1)^2 + (y_2 - \widehat{\beta}_0 - \widehat{\beta}_1 x_2)^2 + \cdots + (y_n - \widehat{\beta}_0 - \widehat{\beta}_1 x_n)^2$$

- The least squares approach chooses $\widehat{\beta}_0$ and $\widehat{\beta}_1$ to minimize the RSS

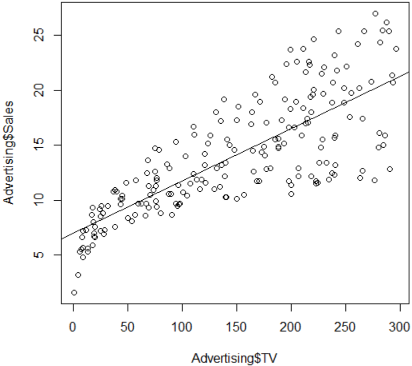
Advertising data



- The least squares fit for the regression of *sales* onto TV.
- A linear fit captures the essence of the relationship in this case.
- Although the linear fit is somewhat deficient in the left of the plot.

Advertising data

```
> # Fit the simple linear regression model with predictor TV and
> mylm = lm(Sales~TV,data=Advertising)
> plot(Advertising$TV,Advertising$Sales)
> abline(mylm)
> # abline draws a line, when given an intercept and slope, or a
```



Using R fit simple linear regression

```
> summary(mylm) # displays basic summaries of the lm
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 7.032594   0.457843   15.36   <2e-16 ***
TV          0.047537   0.002691   17.67   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Multiple linear regression

- Here our model is

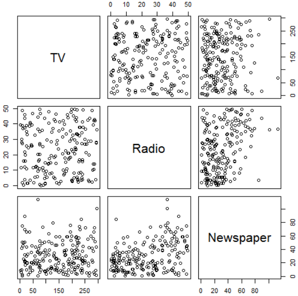
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

- We interpret β_j as the average effect on Y of a one unit increase in X_j , holding all other predictors fixed.
- Ideally, p predictors are independent.
- In the advertising example, the model becomes

$$\text{sales} = \beta_0 + \beta_1 \times \text{TV} + \beta_2 \times \text{radio} + \beta_3 \times \text{newspaper} + \varepsilon$$

Multiple predictors

```
> pairs(Advertising[,1:3])  # look at the X variables
> cor(Advertising[,1:3])
                  TV      Radio  Newspaper
TV        1.00000000 0.05480866 0.05664787
Radio     0.05480866 1.00000000 0.35410375
Newspaper 0.05664787 0.35410375 1.00000000
```



Using R to implement a multiple linear regression fit

```
> mylm2 = lm(Sales~TV+Radio+Newspaper,data=Advertising)
> summary(mylm2)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.938889   0.311908   9.422   <2e-16 ***
TV           0.045765   0.001395  32.809   <2e-16 ***
Radio        0.188530   0.008611  21.893   <2e-16 ***
Newspaper   -0.001037   0.005871  -0.177     0.86
```

Potential problems in multiple linear regression

1. Non-linearity of the response-predictor relationships.
2. Correlation of error terms
3. Non-constant variance of error terms
4. Outliers
5. High-leverage (influential) points
6. Collinearity (correlated predictors)
7. Overfitting and underfitting

Presentation II

- Model selection in regression
- Variable selection (VS) in linear regression
- Techniques and tools for VS in low-dimensional linear regression
- Case studies

Results for advertising data

	Coefficient	Std. Error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

Correlations:

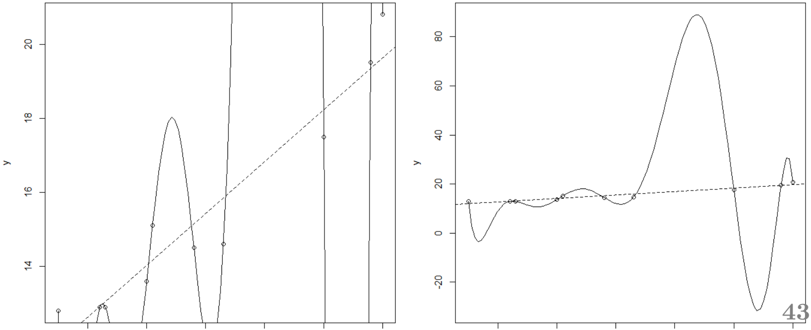
	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

Overfitting and model selection

Least squares is good for model fitting, but useless for model selection. Why?

- A bigger model always has a smaller residual sum of squares, just because a minimum taken over a larger set is smaller.
- Thus least squares, taken as a criterion for model selection says "always choose the biggest model."

Figure : Some regression data. With fitted linear regression function (dashed line) and ninth degree polynomial regression function (solid curve).



Why model selection?

- A lower degree polynomial will have less variance.
- A higher degree will have less bias.
- The bias-variance trade-off will be worse for either of the above.
- We want the model with the smallest MSE, the model which makes the optimal bias-variance trade-off.

Three model selection criteria

- Mallows' Cp

$$C_p = \frac{SSE_p}{\widehat{\sigma}^2} + 2p - n = (k-p)(F_{k-p,n-k} - 1) + p$$

- Akaike information criterion (AIC)

$$AIC = -2 \cdot \log - \text{likelihood} + 2p$$

- Bayes information criterion (BIC)

$$-2 \cdot \log - \text{likelihood} + p \log(n)$$

- In a linear regression model,

$$\log - \text{likelihood} = \text{constant} - (n/2)\log(SSE_p).$$

  Then

$$AIC = \text{constant} + n\log(SSE_p) + 2p$$

$$BIC = \text{constant} + n\log(SSE_p) + p\log(n)$$

How to select the best model?

- Find the model minimizing either of the above three criteria

```
> ?AIC
starting httpd help server ... done
> ?BIC
> out <- lm(y ~ poly(x, 8))
> AIC(out)
[1] 23.09325
> BIC(out)
[1] 26.1191
> AIC(out,k=log(n))
[1] 26.1191
```

Model selection for the sim.txt data

```
n <- length(y)
deg <- seq(1, 8, 1)
cp <- length(deg)
aic <- length(deg)
bic <- length(deg)
out.big <- lm(y ~ poly(x, 2))
sigsqhat.big <- summary(out.big)$sigma^2
for (i in seq(along = deg)) {
 k <- deg[i]
 out <- lm(y ~ poly(x, k))
 aic[i] <- AIC(out)
 bic[i] <- AIC(out, k = log(n))
 cp[i] <- sum(out$residuals^2)/sigsqhat.big + 2 * out$rank - n
}
foo <- cbind(deg,  cp, aic, bic)
dimnames(foo) <- list(NULL, c("degree", "Cp", "AIC", "BIC"))
```

Output for the sim.txt data

```
>foo
     degree       Cp      AIC      BIC
[1,]      1 8.241667 28.82298 29.73074
[2,]      2 3.000000 23.72036 24.93070
[3,]      3 3.105552 22.56455 24.07747
[4,]      4 4.632212 23.59160 25.40712
[5,]      5 5.689259 23.31552 25.43361
[6,]      6 6.575868 21.72313 24.14381
[7,]      7 8.425122 23.12008 25.84334
[8,]      8 9.980202 23.09325 26.11910
```

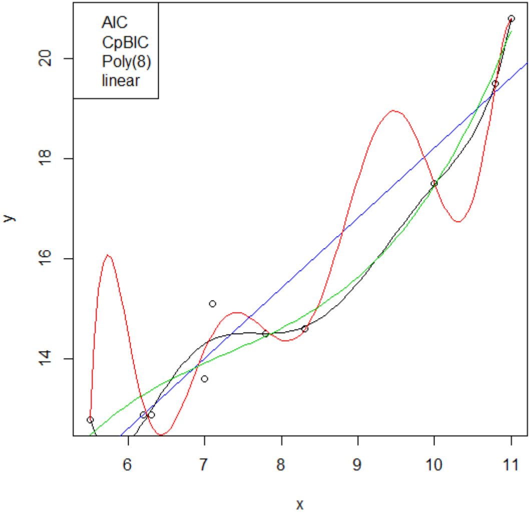
Which model do you choose using different criteria?

Plot the fit

```
par(mfrow=c(1,1))
plot(x, y)
out.poly.CPorBIC<- lm(y ~ poly(x, 3))
out.poly.AIC <- lm(y ~ poly(x, 6))
out.poly.big <- lm(y ~ poly(x, 8))
out.linear <- lm(y ~ x)
curve(predict(out.poly.AIC, data.frame(x = x)), add = TRUE,
         col=1)
curve(predict(out.poly.big, data.frame(x = x)), add = TRUE,
        col=2)
curve(predict(out.poly.CPorBIC, data.frame(x = x)), add = TRUE,
        col=3)
abline(out.linear, col=4)
legend(’topleft’,c(’AIC’, ’CpBIC’,’Poly(8)’,’linear’),
        text.col=c(1,2,3,4))
```

Plot the fit

Figure : Four curve fitting



From model selection to variable selection

- Model selection becomes variable selection under the assumption of linear relationship
- Too many predictors–overfitting; too less predictors–underfitting
- Important question to ask when multiple predictors exist:
  - Do all the predictors help to explain Y , or is only a subset of the predictors useful?
  - If only a subset is useful, how to decide the subset? how to select the important predictors?

Deciding on the important variables

Important question to ask when multiple predictors exist:

- The most direct approach is called all subsets or best subsets regression: we compute the least squares fit for all possible subsets and then choose between them based on some criterion that balances training error with model size.
- However we often can't examine all possible models, since they are 2^p of them; for example when $p = 40$ there are over a billion models!
- Instead we need an automated approach that searches through a subset of them. We discuss two commonly use approaches next.

Different variable selection criteria

- Another approach is to compare the Adjusted R^2 (the best model maximizing Adjusted R^2)
- Three previous mentioned criteria: AIC, BIC, Cp (the best model minimizing the criteria)
- Another approach using Cross-validation (CV) (the best model minimizing CV)

Perform variable selection using AIC or BIC

The R function `step()` can be used to perform variable selection using AIC or BIC.

- Forward selection
- Backward selection
- Stepwise selection–both directions

Forward selection

- Begin with the null model–a model that contains an intercept but no predictors.
- Fit p simple linear regressions and add to the null model the variable that results in the lowest AIC or BIC.
- Add to that model the variable that results in the lowest AIC or BIC amongst all two-variable models.
- Continue until some stopping rule is satisfied, such as AIC or BIC is minimized.

Backward selection

- Start with all variables in the model.
- Remove each variable separately and comparing the resulted AIC or BIC
- The new $(p-1)$-variable model is fit, and the variable with the result in the lowest AIC or BIC is removed.
- Continue until a stopping rule is reached. For instance, we may stop when all remaining variables produced the smallest AIC or BIC.

Select a formula-based model by AIC

```
># Stepwise variable selection
> ? step
step(object, scope, scale = 0,
     direction = c("both", "backward", "forward"),
     trace = 1, keep = NULL, steps = 1000, k = 2, ...)
```

- Backward selection using the `step()` function
- If the `scope` argument is missing the default for direction is “backward”
- If the `scope` argument is not missing, the default for direction is “both”
- Using the AIC criterion (default is $k = 2$)
- If `trace` is positive, all selection steps is given in the output

Final model using backward selection

```
>mylm = lm(Sales~TV+Radio+Newspaper,data=Advertising)
>summary(mylm)
>step(mylm)  # default is backward when just a model is given.

Start:  AIC=212.79
Sales ~ TV + Radio + Newspaper
            Df Sum of Sq    RSS    AIC
- Newspaper  1      0.09  556.9 210.82
<none>                    556.8 212.79
- Radio      1   1361.74 1918.6 458.20
- TV         1   3058.01 3614.8 584.90

Step:  AIC=210.82
Sales ~ TV + Radio
        Df Sum of Sq    RSS    AIC
<none>                 556.9 210.82
- Radio  1    1545.6 2102.5 474.52
- TV     1    3061.6 3618.5 583.10

Call:
lm(formula = Sales ~ TV + Radio, data = Advertising)
```

Forward selection for Advertising data

```
> # To do forward stepwise, first fit a "null" model with just a
> nulllm = lm(Sales~1,data=Advertising)
> mylm2 = step(nulllm,scope=list(lower=nulllm,upper=mylm),direct
```

The above command is different in several ways from forward stepwise.

- it specifies a lower and upper bound of models to consider
- the result is assigned to “mylm2”, saving the resultant linear model as a new model.

Output of forward selection for Advertising data

```
Start:  AIC=661.8
Sales ~ 1
            Df Sum of Sq    RSS    AIC
+ TV         1    3314.6 2102.5 474.52
+ Radio      1    1798.7 3618.5 583.10
+ Newspaper  1     282.3 5134.8 653.10
<none>                   5417.1 661.80

Step:  AIC=474.52
Sales ~ TV
            Df Sum of Sq     RSS    AIC
+ Radio      1   1545.62  556.91 210.82
+ Newspaper  1    183.97 1918.56 458.20
<none>                   2102.53 474.52

Step:  AIC=210.82
Sales ~ TV + Radio
            Df Sum of Sq    RSS    AIC
<none>                   556.91 210.82
+ Newspaper  1  0.088717 556.83 212.79
```

Final model of forward selection

```
> summary(mylm2)

Call:
lm(formula = Sales ~ TV + Radio, data = Advertising)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.7977 -0.8752  0.2422  1.1708  2.8328 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.92110    0.29449   9.919   <2e-16 ***
TV           0.04575    0.00139  32.909   <2e-16 ***
Radio        0.18799    0.00804  23.382   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.681 on 197 degrees of freedom
Multiple R-squared:  0.8972,    Adjusted R-squared:  0.8962 
F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
```

Subset selection from both directions

```
> mylm3 <- step(nulllm,scope=list(lower=nulllm,upper=mylm),
        direction='both')
Start:  AIC=661.8
Sales ~ 1
            Df Sum of Sq    RSS    AIC
+ TV         1    3314.6 2102.5 474.52
+ Radio      1    1798.7 3618.5 583.10
+ Newspaper  1     282.3 5134.8 653.10
<none>                   5417.1 661.80

Step:  AIC=474.52
Sales ~ TV
            Df Sum of Sq    RSS    AIC
+ Radio      1    1545.6  556.9 210.82
+ Newspaper  1     184.0 1918.6 458.20
<none>                   2102.5 474.52
- TV         1    3314.6 5417.1 661.80
```

Final model of subset selection from both directions (AIC)

```
Step:  AIC=210.82
Sales ~ TV + Radio
            Df Sum of Sq    RSS    AIC
<none>                    556.9 210.82
+ Newspaper  1      0.09  556.8 212.79
- Radio      1   1545.62 2102.5 474.52
- TV         1   3061.57 3618.5 583.10

> summary(mylm3)

Call:
lm(formula = Sales ~ TV + Radio, data = Advertising)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.92110    0.29449   9.919   <2e-16 ***
TV           0.04575    0.00139  32.909   <2e-16 ***
Radio        0.18799    0.00804  23.382   <2e-16 ***
```

Using BIC in R

- Using both directions with BIC criterion
- Difference from AIC
  - We replace "$k = 2$" by "$k = log(n)$" in BIC
  - We omit the output of all steps using the option `trace=0`

```
> step(nulllm,scope=list(lower=nulllm,upper=mylm),
       k=log(n),trace=0)

Call:
lm(formula = Sales ~ TV + Radio, data = Advertising)

Coefficients:
(Intercept)           TV        Radio
    2.92110      0.04575      0.18799
```

Another approach to implement subset selection

A few steps:

- Install R package `leaps`
- Using R function `regsubsets()`
- Using `plot.regsubsets` to check the output

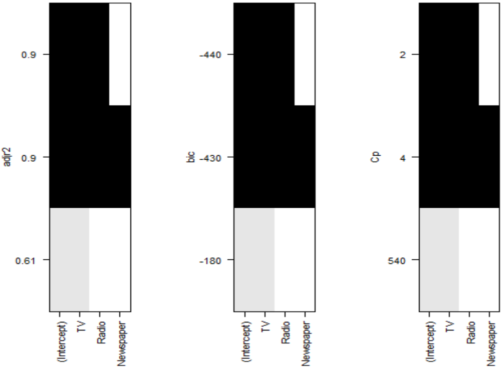
  ```
  plot(x, labels=obj$xnames, main=NULL, scale=c("bic", "Cp",
      "adjr2", "r2"), col=gray(seq(0, 0.9, length = 10)),...)
  ```
- Example

```
>install.packages('leaps')
>library(leaps)
> par(mfrow=c(1,2))
> plot(mylm4, scale="adjr2")
> plot(mylm4, scale="bic")
> plot(mylm4, scale="Cp")
```

leaps output for Advertising data

A few steps:

- Black indicates that a variable is included in the model, while white indicates that they are not.
- The model containing only intercept and TV minimizes the adjusted R^2 and maximize the BIC criteria (left)

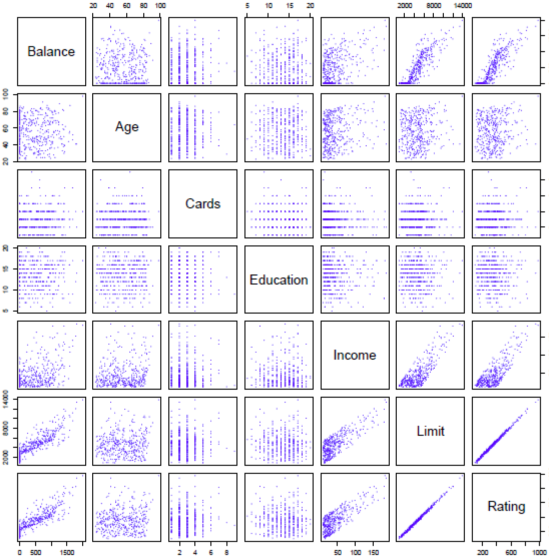


Case study: Credit card data

Investigate effect on credit card balance between different gender, race, student status, marital status, age, education, income, limit, card type

- Some predictors are not quantitative but are qualitative, taking a discrete set of values.
- These are also called categorical predictors or factor variables.
- See for example the scatterplot matrix of the credit card data in the next slide.
  - 7 quantitative variables shown: income, limit, age, cards, education, rating, balance
  - 5 explicit qualitative variables: gender, student (student status), Married (marital status), and Ethnicity (Caucasian, African American (AA) or Asian)

Qualitative Predictors– continued



Descriptive analysis: quantitative data

```
> summary(credit[,c(1:5,11)])
     Income           Limit           Rating          Cards      
 Min.   : 10.35   Min.   :  855   Min.   : 93.0   Min.   :1.000  
 1st Qu.: 21.01   1st Qu.: 3088   1st Qu.:247.2   1st Qu.:2.000  
 Median : 33.12   Median : 4622   Median :344.0   Median :3.000  
 Mean   : 45.22   Mean   : 4736   Mean   :354.9   Mean   :2.958  
 3rd Qu.: 57.47   3rd Qu.: 5873   3rd Qu.:437.2   3rd Qu.:4.000  
 Max.   :186.63   Max.   :13913   Max.   :982.0   Max.   :9.000  
      Age           Balance       
 Min.   :23.00   Min.   :   0.00  
 1st Qu.:41.75   1st Qu.:  68.75  
 Median :56.00   Median : 459.50  
 Mean   :55.67   Mean   : 520.01  
 3rd Qu.:70.00   3rd Qu.: 863.00  
 Max.   :98.00   Max.   :1999.00  
```

Descriptive analysis: qualitative data

```
> table(credit$Gender)

  Male Female
   193    207
> table(credit$Student)

 No Yes
360  40
> table(credit$Married)

 No Yes
155 245
> table(credit$Ethnicity)

African American            Asian        Caucasian
              99              102              199
```

Qualitative predictors with more than two levels

With more than two levels, we create additional dummy variables. For example, for the `Ethnicity` variable we create two dummy variables. The first could be

$$x_{i1} = \begin{cases} 1 & \text{if person } i \text{ is Asian} \\ 0 & \text{if person } i \text{ is not Asian} \end{cases}$$

The second could be

$$x_{i2} = \begin{cases} 1 & \text{if person } i \text{ is Caucasian} \\ 0 & \text{if person } i \text{ is not Caucasian} \end{cases}$$

Qualitative predictors with more than two levels

Then both of these variables can be used in the regression equation, in order to obtain the model

$$Y_i = \beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\varepsilon_i = \begin{cases} \beta_0+\beta_1+\varepsilon_i & \text{if person } i \text{ is Caucasian} \\ \beta_0+\beta_2+\varepsilon_i & \text{if person } i \text{ is not Caucasian} \\ \beta_0+\varepsilon_i & \text{if person } i \text{ is not AA} \end{cases}$$

There will always be one fewer dummy variable than the number of levels. The level with no dummy variable– African American in this example–is known as the baseline.

Run a full model analysis

```
> full=lm(Balance~.,data=credit) ## run a full model
> summary(full)
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        -479.20787   35.77394 -13.395  < 2e-16 ***
Income               -7.80310    0.23423 -33.314  < 2e-16 ***
Limit                 0.19091    0.03278   5.824 1.21e-08 ***
Rating                1.13653    0.49089   2.315   0.0211 *
Cards                17.72448    4.34103   4.083 5.40e-05 ***
Age                  -0.61391    0.29399  -2.088   0.0374 *
Education            -1.09886    1.59795  -0.688   0.4921
GenderFemale        -10.65325    9.91400  -1.075   0.2832
StudentYes          425.74736   16.72258  25.459  < 2e-16 ***
MarriedYes           -8.53390   10.36287  -0.824   0.4107
EthnicityAsian       16.80418   14.11906   1.190   0.2347
EthnicityCaucasian   10.10703   12.20992   0.828   0.4083

Residual standard error: 98.79 on 388 degrees of freedom
Multiple R-squared:  0.9551,    Adjusted R-squared:  0.9538
F-statistic: 750.3 on 11 and 388 DF,  p-value: < 2.2e-16
```

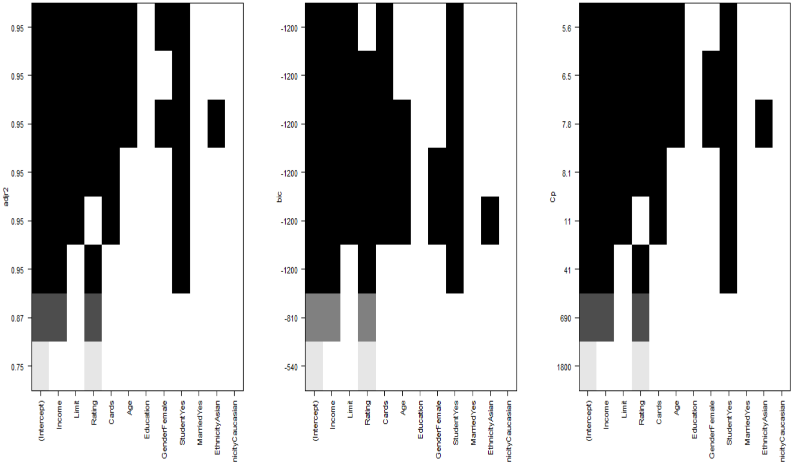
Run a full stepwise search using BIC

```
> null=lm(Balance~1,data=credit)
> #run a full stepwise search using BIC first
> n=dim(credit)[1]
> final.bic=step(null,scope=list(lower=null,upper=full),
                k=log(n),trace=0)#BIC
> summary(final.bic)
Call:
lm(formula = Balance ~ Income + Student + Limit + Cards,
    data = credit)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.997e+02  1.589e+01 -31.449  < 2e-16 ***
Income      -7.839e+00  2.321e-01 -33.780  < 2e-16 ***
StudentYes   4.296e+02  1.661e+01  25.862  < 2e-16 ***
Limit        2.666e-01  3.542e-03  75.271  < 2e-16 ***
Cards        2.318e+01  3.639e+00   6.368 5.32e-10 ***
```

Run a full stepwise search using AIC

```
> final.aic=step(null,scope=list(lower=null,upper=full),
                k=a,trace=0)  #AIC
> summary(final.aic)
lm(formula = Balance ~ Rating + Income + Student + Limit + Cards
    Age, data = credit)

              Estimate Std. Error t value Pr(>|t|)
(Intercept) -493.73419   24.82476 -19.889  < 2e-16 ***
Rating         1.09119    0.48480   2.251   0.0250 *
Income        -7.79508    0.23342 -33.395  < 2e-16 ***
StudentYes   425.60994   16.50956  25.780  < 2e-16 ***
Limit          0.19369    0.03238   5.981 4.98e-09 ***
Cards         18.21190    4.31865   4.217 3.08e-05 ***
Age           -0.62406    0.29182  -2.139   0.0331 *
```

leaps output for credit data



Summary on the model selection

- BIC is the most aggressive, choose the smallest model
    - select `Income`, `Student`, `Limit`, `Cards`
- AIC and Cp perform similarly. They select a less aggressive, and larger model than the BIC does
    - select `Rating`, `Income`, `Student`, `Limit`, `Cards`, `Age`
- The adjust R^2 is the most conservative, select a even larger model.
    - select `Rating`, `Income`, `Student`, `Limit`, `Cards`, `Age` and `Gender`

Report your final model

- If we want to use a more parsimony model, we use BIC
- The final model is

  Average Balance=-5+7.84 Income+4.3 I(Student)+2.67 Limit+2.32 Card
- How to interpret your final selected model?
- I would suggest you to check the AIC produced model. Some signs are different from the BIC one.

Presentation III

- High-dimensional data
- Variable selection (VS) in high-dimensional (HD) data
- Techniques and tools for VS in HD linear regression
- Case studies

The era of big data

We are entering a big data era:

- The data can be obtained continuously and stored at a much cheaper cost
- The data can be very high dimensional or unstructured
- the big data trend is likely to maintain or even accelerate
- The massive amounts of high dimensional data bring both opportunities and new challenges to data analysis
- Scientific advances are becoming more and more data-driven
- Valid statistical analysis for Big Data is becoming increasingly important

Big data arise in many fields

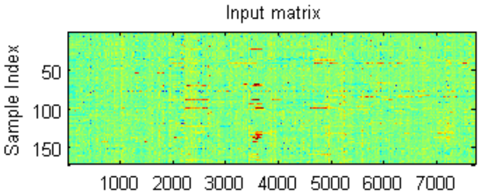
- *Cancer research.* How to find important genes from the massive genomic data at tens of thousands genomic markers from only hundreds or less samples. For example, gene expression level at 21, 944 genes of 59 samples are collected to predict the protein expression on the KRT18 antibody from other gene expression levels (Shankavaram et al., 2007)
- *E-business.* How to predict online sellers' pricing strategies in gray market in e-business? For example, all listings, sellers and transactions of 183 Coach handbag styles on Taobao are collected from May 30th, 2011 to January 23rd, 2012. Month 1 only includes over 7,000 listings and over 200 features in total (Zhao et al. 2015).
- *Social media.* How to detect zombie accounts in online social networks? 10,000 inspected accounts information are collected from SINA WeiBo in late 2014 (Deng et al. 2015).
- *Many many others*$\cdots$

High-dimensional Data

Take a multiple regression model setting as an example

- $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$, $i = 1, \cdots, n$
  - p is huge, for example $p = \exp(n^\alpha)$, $0 < \alpha < 1$, high-dimensionality
  - n is large too, big data in general
  - High correlation among some predictors
- Model assumptions of high-dimensional data analysis
  - The model is sparse: some covariates (x_{ij}s) are noisy covariates
  - The sparsity has certain structures (spatial dependence, group-wise)
  - How to find those truly important features with certain structured sparsity

Bekhouche copy number data



- Heatmap of raw copy number data from Chromosome 17 (7,727 probes from 173 breast tumors)
- Target is to detect both recurrent copy number variations (CNVs) and individual CNVs
- After normalization, copy number data should be around 0 if there is no copy number variation
- Recurrent CNVs means a genome region where CNVs exist for a group of samples

Goal of Big data: Feature selection

What are the goals of analyzing Big Data?

- According to Bickel (2008), in high-dimensional data analysis
  - develop effective methods that can accurately predict the future observations and at the same time
  - to gain insight into the relationship between the features and response for scientific purposes

Penalized regression and Lasso

Lasso stands for least absolute shrinkage and selection operator (Tibshirani 1996)

- The ℓ_1 penalized criterion (LASSO criterion) is to minimize

$$\sum_{i=1}^{n}[y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip})]^2$$

  subject to $|\beta_1| + |\beta_2| + \cdots + |\beta_p| < s$ for some $s > 0$

  - It provide exact 0 estimate for some coefficients
  - It run both variable selection and coefficients estimation simultaneously
  - It is computationally efficient when p is large
- Lasso is a regularization approach

Other alternatives to Lasso

Lasso is a regularization approach providing sparse solution: 0 estimations for some coefficients. It also has some advantages and disadvantages

- When p is not too big, and important variables are not too much correlated with those unimportant ones, Lasso is a good option
- When p is huge, Lasso tends to select a larger model.
- When $p \gg n$, SCAD or MCP or Adaptive Lasso provide a better variable selection result.
- Lasso is a regularization approach

Lasso is a regularization approach providing sparse solution: 0 estimations for some coefficients. It also has some advantages and disadvantages

- When p is not too big, and important variables are not too much correlated with those unimportant ones, Lasso is a good option
- When p is huge, Lasso tends to select a larger model.
- When $p \gg n$, SCAD or MCP or Adaptive Lasso provide a better variable selection result.
- Lasso is a regularization approach

Existing R packages and functions

Existing R packages and functions have been developed very quickly recently. I will introduce three typical ones.

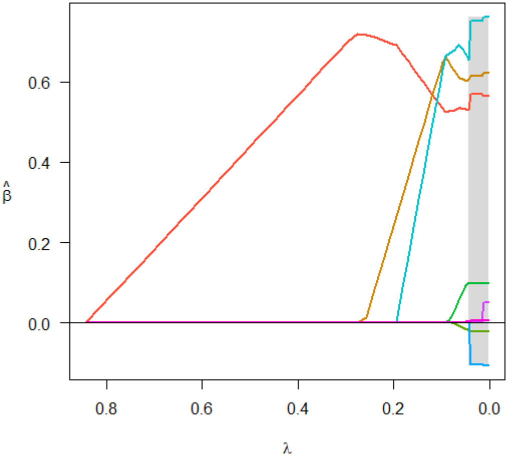
- `ncvreg` package: Regularization Paths for SCAD and MCP Penalized Regression Models
  - It is an efficient algorithms for fitting regularization paths for linear or logistic regression.
  - It also provide penalized regression using MCP or SCAD, and an optional additional L2 penalty.
- `parcor` package: Regularized estimation of partial correlation matrices
  - It provides model selection for lasso, adaptive lasso and Ridge regression based on cross-validation.
- `lassoshooting` package: ℓ_1 regularized regression (Lasso) solver using the Cyclic Coordinate Descent algorithm aka Lasso Shooting
  - Designed for Lasso and Adaptive Lasso

Prostate data

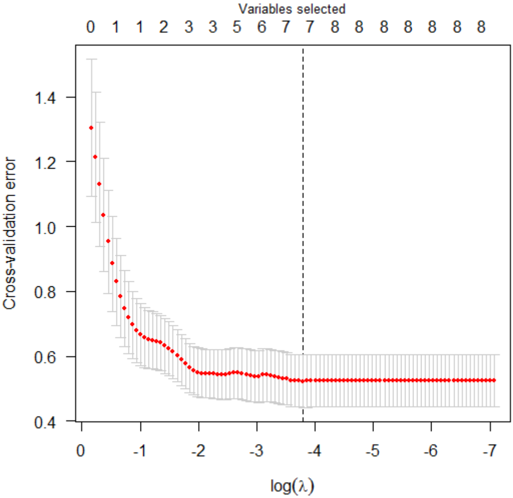
Data from a study by by Stamey et al. (1989) to examine the association between prostate specific antigen (PSA) and several clinical measures that are potentially associated with PSA in men who were about to receive a radical prostatectomy. 97 observations on 9 variables. The variables are as follows:

- lcavol: Log cancer volume
- lweight: Log prostate weight
- age: The mans age
- lbph: Log of the amount of benign hyperplasia
- svi: Seminal vesicle invasion; 1=Yes, 0=No
- lcp: Log of capsular penetration
- gleason: Gleason score
- pgg45: Percent of Gleason scores 4 or 5
- lpsa: Log PSA

Prostate data Lasso solution path



Prostate data Lasso solution cross validation curve



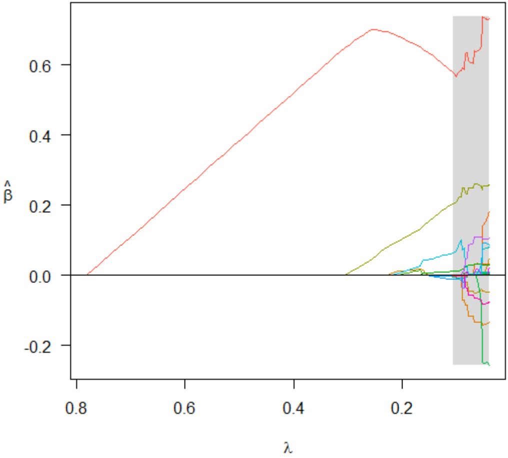
Variable selection result for Prostate data

```
>data(prostate)
>X <- as.matrix(prostate[,1:8])
>y <- prostate$lpsa
>fit <- ncvreg(X,y)
>plot(fit) ## Lasso solution path

>cvfit <- cv.ncvreg(X,y)
>plot(cvfit) ## Lasso cross validation curve

>fitall <- cvfit$fit
>beta <- fitall$beta[,cvfit$min]
>beta
 (Intercept)       lcavol      lweight          age         lbph
 0.494142558  0.569547314  0.614420958 -0.020913384  0.097352193
svi         lcp      gleason        pgg45
0.752400999 -0.104961106  0.000000000  0.005324464
```

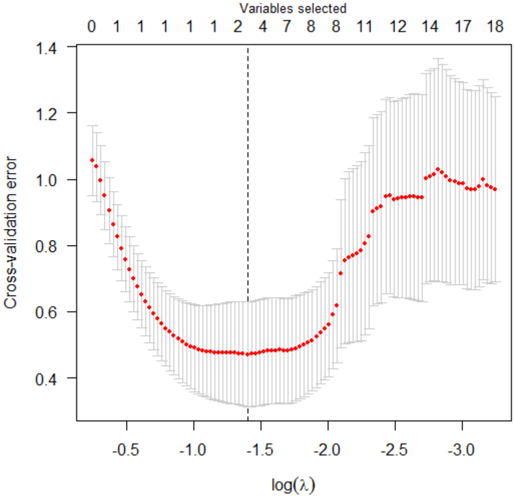
NCI-60 data

- The NCI-60 is a gene expression data set collected from Affymetrix HG-U133A chip and normalized with the GCRMA method.
- The study is to predict the protein expression on the KRT18 antibody from other gene expression levels.
- After removing the missing data, there are $n = 59$ samples with $21,944$ genes in the dataset.
- We preprocess the data by scaling all variables using the MAD with the consistency factor 1.4826.
- The response variable is generated from variables with the largest MAD, which turns out to measure the expression levels of the protein *keratin* 18.
- After initial analysis, we keep 500 genes which are most related to *keratin* 18.

NCI-60 data Lasso solution path



NCI-60 data Lasso solution cross validation curve



Variable selection result for NCI-60 data

```
>load("nci_MAD.rda")
>names(nci_MAD)
[1] "x" "y"
>dim(nci_MAD$x) # [1]  59 500
>length(nci_MAD$y)

>fit_nci <- ncvreg(nci_MAD$x,nci_MAD$y)
>plot(fit_nci) ## Lasso solution path

>cvfit_nci <- cv.ncvreg(nci_MAD$x,nci_MAD$y)
>plot(cvfit_nci) ## Lasso cross validation curve
>fitall_nci <- cvfit_nci$fit
>beta_nci <- fitall_nci$beta[,cvfit_nci$min]
>beta_nci[!beta_nci==0]
(Intercept)        KRT8
  0.1462848   0.6234228
```