SOSC 5340 Tutorial 2

GLM and Machine Learning Basics

Yabin YIN
HKUST

Mar, 2021

Set working directory to the current directory

Remark: Need to save current R file before using *getActiveDocumentContext*

R Packages

R packages for multinomial logit regression:

- *mlogit*: https://cran.r-project.org/web/packages/mlogit
- *mnlogit*: https://cran.r-project.org/web/packages/mnlogit
- *multinom*: https://cran.r-project.org/web/packages/nnet
- Read the *reference manual* and *vignettes*.
- We will focus on the *multinom* function from *nnet* package. Please try other packages yourself.

R packages for ordinal logit regression:

- *polr*: https://rdrr.io/cran/MASS/man/polr.html
- *oglmx*: https://cran.r-project.org/web/packages/oglmx

R packages for LASSO and Ridge:

- *glmnet*: https://cran.r-project.org/web/packages/glmnet

R packages for Tree and Forests:

- *tree*: https://cran.r-project.org/web/packages/tree
- *randomForest*: https://cran.r-project.org/web/packages/randomForest

R packages for imputing missing values:

- *mice*: https://cran.r-project.org/web/packages/mice
- Other packages: https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/

Multinomial Logit Regression

Empirical example:

Entering high school students make program choices among **general program**, **vocational program** and **academic program**. Their choice might be modeled using their **writing score** and their **social economic status**.

```r
# require the packages
library(foreign)
library(tinytex)

# load the data
hsbdemo <- read.dta("https://stats.idre.ucla.edu/stat/data/hsbdemo.dta")
```

The data set contains variables on 200 students. The outcome variable is **prog**, program type. The independent variables are social economic status, **ses**, a three-level categorical variable and writing score, **write**, a continuous variable. Let's start with getting some descriptive statistics of the variables of interest.

```r
## ses by program types
with(hsbdemo, table(ses, prog))
```

```
##         prog
## ses      general academic vocation
##   low         16       19       12
##   middle      20       44       31
##   high         9       42        7
```

```r
## avg.writing score by program types
with(hsbdemo, do.call(rbind, tapply(write, prog, function(x) c(M = mean(x), SD = sd(x)))))
```

```
##                 M       SD
## general  51.33333 9.397775
## academic 56.25714 7.943343
## vocation 46.76000 9.318754
```

Let's use the `multinom` function from the `nnet` package to estimate a multinomial logistic regression model.

```r
## library the package
library(nnet)
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```r
## set reference group: academic program
hsbdemo$prog2 <- relevel(hsbdemo$prog, ref = "academic")

## run multinomial logit regression
mlogit1 <- multinom(prog2 ~ ses + write, data = hsbdemo)
```

```
## # weights:  15 (8 variable)
## initial  value 219.722458
## iter  10 value 179.982880
## final  value 179.981726
## converged
```

```r
summary(mlogit1)
```

```
## Call:
## multinom(formula = prog2 ~ ses + write, data = hsbdemo)
##
## Coefficients:
```

```
##          (Intercept)  sesmiddle    seshigh      write
## general     2.852198 -0.5332810 -1.1628226 -0.0579287
## vocation    5.218260  0.2913859 -0.9826649 -0.1136037
##
## Std. Errors:
##          (Intercept) sesmiddle   seshigh      write
## general     1.166441 0.4437323 0.5142196 0.02141097
## vocation    1.163552 0.4763739 0.5955665 0.02221996
##
## Residual Deviance: 359.9635
## AIC: 375.9635
```

```
## Wald-tests (here z-tests)
z_score <- summary(mlogit1)$coefficients/summary(mlogit1)$standard.errors
z_score
```

```
##          (Intercept)  sesmiddle   seshigh     write
## general     2.445214 -1.2018081 -2.261334 -2.705562
## vocation    4.484769  0.6116747 -1.649967 -5.112689
```

```
## p-values (2-tailed)
p_value <- (1 - pnorm(abs(z_score), 0, 1)) * 2
p_value
```

```
##           (Intercept) sesmiddle    seshigh        write
## general  0.0144766100 0.2294379 0.02373856 6.818902e-03
## vocation 0.0000072993 0.5407530 0.09894976 3.176045e-07
```

```
# report the results
stargazer(mlogit1, type='text', no.space = T)
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                        general       vocation
##                          (1)            (2)
## ------------------------------------------------
## sesmiddle               -0.533          0.291
##                        (0.444)        (0.476)
## seshigh                -1.163**        -0.983*
##                        (0.514)        (0.596)
## write                 -0.058***      -0.114***
##                        (0.021)        (0.022)
## Constant               2.852**        5.218***
##                        (1.166)        (1.164)
## ------------------------------------------------
## Akaike Inf. Crit.      375.963        375.963
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

If we consider our coefficients from the first column to be b_1 and our coefficients from the second column to be b_2, we can write our model equations:

$$ln\frac{P(prog = general)}{P(prog = academic)} = b_{10} + b_{11}(ses = 2) + b_{12}(ses = 3) + b_{13}write$$

$$ln\frac{P(prog = vocation)}{P(prog = academic)} = b_{20} + b_{21}(ses = 2) + b_{22}(ses = 3) + b_{23}write$$

Interpretation:

Continuous Variable (write)

b_{13} A one-unit increase in the variable write is associated with the decrease in the **log odds** of choosing general program rather than academic program in the amount of 0.058. Or you can say, a 1-unit increase in the variable write will decrease the **odds** of being in general program rather than academic program to exp(-0.058)=0.94.

b_{23} A one-unit increase in the variable write is associated with the decrease in the **log odds** of being in vocation program vs. academic program. in the amount of 0.1136. A 1-unit increase in the variable write will decrease the **odds** of being in vocation program rather than academic program by 1-exp(-0.1136)=0.107.

Categorical Variable (ses)

b_{11} The **log odds** of choosing general program rather than academic program for middle SES students is 0.533 lower than that of low SES students. Or, the **odds ratio** of choosing general program rather than academic program for middle SES students is exp(-0.533)=0.586 of that figure for lower SES students.

b_{21} The **log odds** of choosing vocation program rather than academic program for middle SES students is 0.291 higher than that of low SES students. Or, the **odds ratio** of choosing vocation program rather than academic program for middle SES students is exp(0.291)=1.33 times of that figure for lower SES students.

Probability:

If we want to examine the changes in predicted probability associated with one of our two variables, we can create small dataframe varying one variable while holding the other constant.

We will do this holding write at its mean and examining the predicted probabilities for each level of ses.

```
dses <- data.frame(ses = c("low", "middle", "high"), write = mean(hsbdemo$write))
predict(mlogit1, newdata = dses, "probs")
```

```
##    academic   general  vocation
## 1 0.4396845 0.3581917 0.2021238
## 2 0.4777488 0.2283353 0.2939159
## 3 0.7009007 0.1784939 0.1206054
```

Ordinal Logit Regression

Example:

To understand the working of Ordinal Logistic Regression, we'll consider a study from *World Values Surveys*, which looks at factors that influence people's perception of the government's efforts to reduce poverty.

Our objective is to predict an individual's perception about government's effort to reduce poverty based on factors like individual's country, gender, age etc. In the given case study, individual's perception can take the following three values - **Too Little**, **About Right**, **Too Much**.

```
## library packages
library(carData)

## load the data
data(WVS)
head(WVS)
```

```
##       poverty religion degree country age gender
```

```
## 1  Too Little      yes     no     USA  44   male
## 2 About Right      yes     no     USA  40 female
## 3  Too Little      yes     no     USA  36 female
## 4    Too Much      yes    yes     USA  25 female
## 5  Too Little      yes    yes     USA  39   male
## 6 About Right      yes     no     USA  80 female
```

- *religion*: member of a religion -no or yes
- *degree*: held a university degree -no or yes
- *country*: Australia, Norway, Sweden or the USA
- *age*: age (years)
- *gender*: male or female

We'll now fit the Ordinal Logit Regression model using `polr` function from the `MASS` package.

```
## library packages
library(MASS)

## fit ologit model
ologit1 <- polr(poverty ~ religion+degree+country+age+gender, data = WVS, Hess = TRUE)
summary(ologit1)
```

```
## Call:
## polr(formula = poverty ~ religion + degree + country + age +
##     gender, data = WVS, Hess = TRUE)
##
## Coefficients:
##                   Value Std. Error t value
## religionyes     0.17973   0.077346   2.324
## degreeyes       0.14092   0.066193   2.129
## countryNorway -0.32235   0.073766  -4.370
## countrySweden -0.60330   0.079494  -7.589
## countryUSA      0.61777   0.070665   8.742
## age             0.01114   0.001561   7.139
## gendermale      0.17637   0.052972   3.329
##
## Intercepts:
##                          Value   Std. Error t value
## Too Little|About Right  0.7298  0.1041      7.0128
## About Right|Too Much    2.5325  0.1103     22.9496
##
## Residual Deviance: 10402.59
## AIC: 10420.59
```

```
## p_value
p_value2 <- pnorm(abs(coef(summary(ologit1))[,'t value']),lower.tail = FALSE)* 2
summary_table <- cbind(coef(summary(ologit1)), 'pval'=round(p_value2,3))
summary_table
```

```
##                              Value  Std. Error   t value  pval
## religionyes             0.17973194 0.077346042  2.323738 0.020
## degreeyes               0.14091745 0.066193109  2.128884 0.033
## countryNorway          -0.32235359 0.073766034 -4.369946 0.000
## countrySweden          -0.60329785 0.079493909 -7.589234 0.000
## countryUSA              0.61777260 0.070664761  8.742301 0.000
## age                     0.01114091 0.001560585  7.138933 0.000
## gendermale              0.17636863 0.052972253  3.329453 0.001
```

```
## Too Little|About Right  0.72976353 0.104061643  7.012800 0.000
## About Right|Too Much    2.53247870 0.110349780 22.949558 0.000
```

Interpretation

Let J be the total number of categories of the dependent variable *(poverty)*.

- *j=1* refers to "Too Little"
- *j=2* refers to "About Right"
- *j=3* refers to "Too Much"

$$Logit(Y \leq 1|X_i) = \alpha_1 + \sum \beta_i X_i;\ P(Y \leq 1|X_i) = Logit^{-1}(\alpha_1 + \sum \beta_i X_i)$$

$$Logit(Y \leq 2|X_i) = \alpha_2 + \sum \beta_i X_i;\ P(Y \leq 2|X_i) = Logit^{-1}(\alpha_2 + \sum \beta_i X_i)$$

$$P(1 < Y \leq 2|X_i) = P(Y \leq 2|X_i) - P(Y \leq 1|X_i) = Logit^{-1}(\alpha_2 + \sum \beta_i X_i) - Logit^{-1}(\alpha_1 + \sum \beta_i X_i)$$

Coefficients:

gender: The log odds (odds ratio) of having a positive perception about government's efforts to reduce poverty of male is 0.176 higher (1.19 times) compared with female.

age: With one unit increase in age, the log odds of having a positive perception about government's efforts to reduce poverty increases by 0.011.

Intercepts:

- Mathematically, the intercept **'Too Little | About Right'** corresponds to **Logit(P(Y <= 1))**. It can be interpreted as the log odds of believing that the government is doing **'Too Little'** versus believing that the government is doing **'About Right' or 'Too Much'** is of 0.72 higher.
- Similarly, the intercept **'About Right | Too Much'** corresponds to **Logit[P(Y <= 2)]**. It can be interpreted as the log of odds of believing that the government is doing **'Too Little' or 'About Right'** versus believing that the government is doing **'Too Much'** is of 2.53 times.

Probability

- The probability corresponding to **Too Little** perception will be calculated as:

  $Logit[P(Y \leq 1)] = 0.7298 - [(0.17973 * 1) + (0.14092 * 0) + (-0.32235 * 1) + (0.01114 * 30) + (0.17637 * 1)] Logit[P(Y \leq 1)] = 0.$

- Similarly, the probability corresponding to **About Right** perception will be calculated as:

  $Logit[P(Y \leq 2)] = 2.5325 - [(0.17973 * 1) + (0.14092 * 0) + (-0.32235 * 1) + (0.01114 * 30) + (0.17637 * 1)] Logit[P(Y \leq 2)] = 2.$

- The probability corresponding to **Too Much** perception will be calculated as:

$$P(Y = 3) = 1 - P(Y \leq 2) = 0.103$$

Computation in R

Now, let's use `predict` function in R to complete above mathematical calculation.

```
pred <- round(predict(ologit1,WVS,type = "probs"), 3)
head(pred)
```

```
##   Too Little About Right Too Much
## 1      0.324       0.420    0.256
## 2      0.374       0.410    0.216
## 3      0.385       0.407    0.209
## 4      0.381       0.408    0.212
## 5      0.306       0.422    0.272
## 6      0.277       0.422    0.301
```

```
new_data <- data.frame("religion"="yes","degree"="no","country"="Norway","age"=30,
                       "gender"="male")
round(predict(ologit1, new_data, type = "probs"), 3)
```

```
##  Too Little About Right    Too Much
##       0.589       0.308       0.103
```

LASSO

We use `glmnet` function to fit LASSO and Ridge model. Let's first review the estimation:

$$\hat{\beta}_{LASSO} = argmin_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$\hat{\beta}_{Ridge} = argmin_{\beta} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

The tuning parameter λ controls the overall strength of the penalty.

The **Ridge** penalty shrinks the coefficients of correlated predictors towards each other while the **LASSO** tends to pick one of them and discard the others.

Example: LASSO

```
## library packages
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-1
```

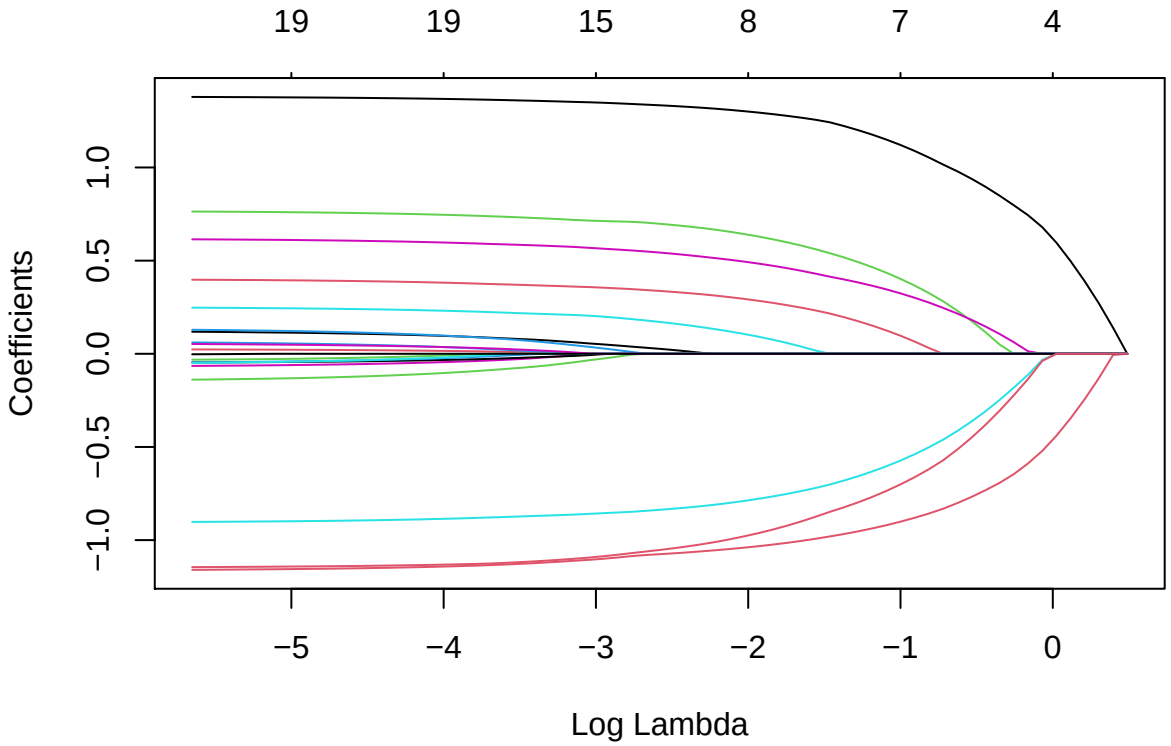
```
## load data
data(QuickStartExample) # The command loads an input matrix x and a response vector y

## fit LASSO model
lasso <- glmnet(x, y, alpha = 1)
```

Notes: **alpha**=1 (default) specifies the model to be LASSO, and **alpha**=0 tells R to fit a Ridge model. We will focus on LASSO in this tutorial, try to fit a Ridge model by yourself.

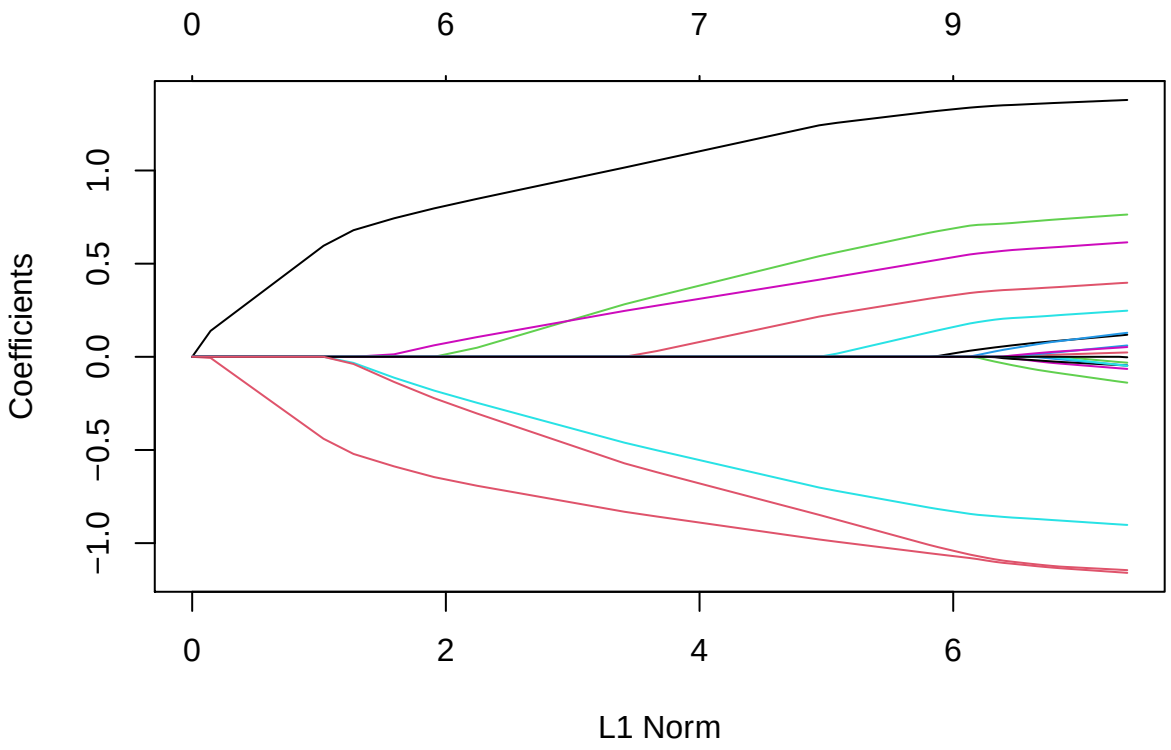
Describe the model

"lasso" is an object of class `glmnet` that contains all the relevant information of the fitted model for further use. We do not encourage users to extract the components directly. Instead, various methods are provided for the object such as `plot`, `print`, `coef` and `predict` that enable us to execute those tasks more elegantly.

```r
## visualize the coefficients
### x-axis1: lambda
plot(lasso, xvar = 'lambda')
```




```r
### x-axis2: L1 Norm
plot(lasso)
```



- **Y-axis:** Regularized coefficients for each variable (ie. coefficients after penalization is applied).
- **X-axis1:** Logarithm of the penalization parameter Lambda (λ). The higher value of lambda indicates more regularization (ie. reduction of the coefficient magnitude, or shrinkage).
- **X-axis2:** L1 Norm ($\sum_{j=1}^{p} |\beta_j|$). The sum of absolute values of estimated coefficients. L1 Norm is small when λ is large.
- **Curve:** Change in the predictor coefficients as the penalty term increases.
- **Numbers on top:** The number of variables in the regression model.
- **Log Lambda = 0** corresponds to "no regularization" (ie. regular linear model with minimum residual sum of squares).

```
print(lasso)
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 1)
##
##    Df  %Dev  Lambda
## 1   0  0.00 1.63100
## 2   2  5.53 1.48600
## 3   2 14.59 1.35400
## 4   2 22.11 1.23400
## 5   2 28.36 1.12400
## 6   2 33.54 1.02400
## 7   4 39.04 0.93320
## 8   5 45.60 0.85030
## 9   5 51.54 0.77470
## 10  6 57.35 0.70590
## 11  6 62.55 0.64320
## 12  6 66.87 0.58610
## 13  6 70.46 0.53400
## 14  6 73.44 0.48660
## 15  7 76.21 0.44330
## 16  7 78.57 0.40400
## 17  7 80.53 0.36810
## 18  7 82.15 0.33540
## 19  7 83.50 0.30560
## 20  7 84.62 0.27840
## 21  7 85.55 0.25370
## 22  7 86.33 0.23120
## 23  8 87.06 0.21060
## 24  8 87.69 0.19190
## 25  8 88.21 0.17490
## 26  8 88.65 0.15930
## 27  8 89.01 0.14520
## 28  8 89.31 0.13230
## 29  8 89.56 0.12050
## 30  8 89.76 0.10980
## 31  9 89.94 0.10010
## 32  9 90.10 0.09117
## 33  9 90.23 0.08307
## 34  9 90.34 0.07569
## 35 10 90.43 0.06897
## 36 11 90.53 0.06284
## 37 11 90.62 0.05726
## 38 12 90.70 0.05217
## 39 15 90.78 0.04754
```

```
## 40 16 90.86 0.04331
## 41 16 90.93 0.03947
## 42 16 90.98 0.03596
## 43 17 91.03 0.03277
## 44 17 91.07 0.02985
## 45 18 91.11 0.02720
## 46 18 91.14 0.02479
## 47 19 91.17 0.02258
## 48 19 91.20 0.02058
## 49 19 91.22 0.01875
## 50 19 91.24 0.01708
## 51 19 91.25 0.01557
## 52 19 91.26 0.01418
## 53 19 91.27 0.01292
## 54 19 91.28 0.01178
## 55 19 91.29 0.01073
## 56 19 91.29 0.00978
## 57 19 91.30 0.00891
## 58 19 91.30 0.00812
## 59 19 91.31 0.00739
## 60 19 91.31 0.00674
## 61 19 91.31 0.00614
## 62 20 91.31 0.00559
## 63 20 91.31 0.00510
## 64 20 91.31 0.00464
## 65 20 91.32 0.00423
## 66 20 91.32 0.00386
## 67 20 91.32 0.00351
```

- **DF:** the number of nonzero coefficients.
- **%dev:** the percent (of null) deviance explained.
- λ**:** the value of λ.

We can obtain the actual coefficients at one or more λ within the range of the sequence

```
coef(lasso,s=0.1)
```

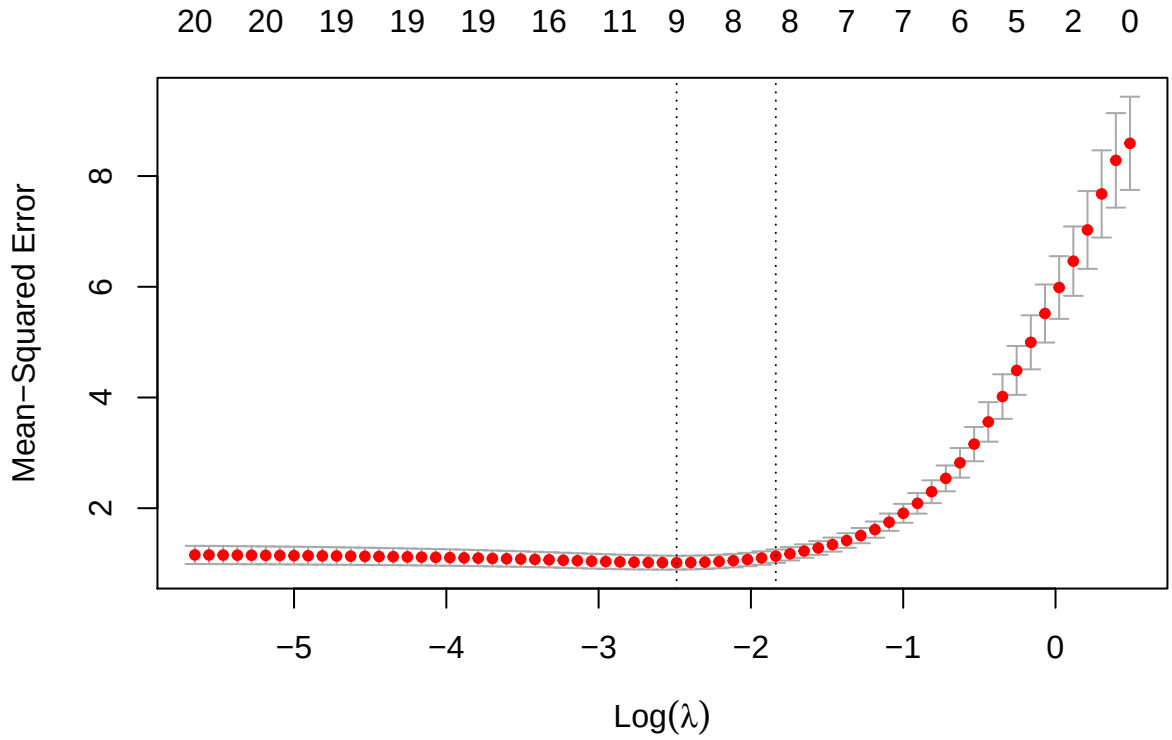
```
## 21 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  0.150928072
## V1           1.320597195
## V2           .
## V3           0.675110234
## V4           .
## V5          -0.817411518
## V6           0.521436671
## V7           0.004829335
## V8           0.319415917
## V9           .
## V10          .
## V11          0.142498519
## V12          .
## V13          .
## V14         -1.059978702
## V15          .
## V16          .
```

```
## V17           .
## V18           .
## V19           .
## V20          -1.021873704
```

Cross Validation

The function `glmnet` returns a sequence of models for the users to choose from. In many cases, users may prefer the software to select one of them. **Cross-validation** is perhaps the simplest and most widely used method for that task.

```
## cross validation using `cv.glmnet`
cv.lasso <- cv.glmnet(x, y, alpha = 1)
## plot
plot(cv.lasso)
```



It includes the cross-validation curve (red dotted line), and upper and lower standard deviation curves along the λ sequence (error bars). Two selected λ's are indicated by the vertical dotted lines (see below).

- The left line: **log(lambda.min)**. `lambda.min` is the minimum value of lambda that results in the **smallest cross-validation error**. This is calculated by dividing the dataset in 10 subsets, followed by the calculation of fit in 9/10 of the subsets and testing the predicted model on the remaining 1/10.
- The right line: **log(lambda.1se)**. `lambda.1se` is the largest value of lambda (i.e. more regularized) within the 1 standard error of the `lambda.min`. This `lambda.1se` value corresponds to a higher level of penalization (i.e. more regularized model) and can be chosen for a simpler model in predictions (less impact from from coefficients).

```
## View the selected lambda's and the corresponding coefficients.
cv.lasso$lambda.min
```

```
## [1] 0.08307327
```

```
cv.lasso$lambda.1se
```

```
## [1] 0.1593271
```

```
coef(cv.lasso, s = "lambda.min")
```

```
## 21 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept)  0.14936467
## V1           1.32975267
## V2           .
## V3           0.69096092
## V4           .
## V5          -0.83122558
## V6           0.53669611
## V7           0.02005438
## V8           0.33193760
## V9           .
## V10          .
## V11          0.16239419
## V12          .
## V13          .
## V14         -1.07081121
## V15          .
## V16          .
## V17          .
## V18          .
## V19          .
## V20         -1.04340741
```

```
## Prediction
predict(cv.lasso, newx = x[1:5,], s = "lambda.min")
```

```
##               1
## [1,] -1.3647490
## [2,]  2.5686013
## [3,]  0.5705879
## [4,]  1.9682289
## [5,]  1.4964211
```

Notes: `newx` is for the new input matrix; `s` is the value(s) of λ at which predictions are made.

Decision Tree and Forests

Decision Trees

We will use `tree` package in R. As for data, we use **Carseats** dataset from `ISLR` package.

```
## library packages
library(tree)
library(ISLR)

## load data
data(Carseats, package="ISLR")
head(Carseats, 5)
```

```
##   Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
```

```
## 1  9.50       138     73          11        276   120       Bad  42        17
## 2 11.22       111     48          16        260    83      Good  65        10
## 3 10.06       113     35          10        269    80    Medium  59        12
## 4  7.40       117    100           4        466    97    Medium  55        14
## 5  4.15       141     64           3        340   128       Bad  38        13
##   Urban  US
## 1   Yes Yes
## 2   Yes Yes
## 3   Yes Yes
## 4   Yes Yes
## 5   Yes  No
```

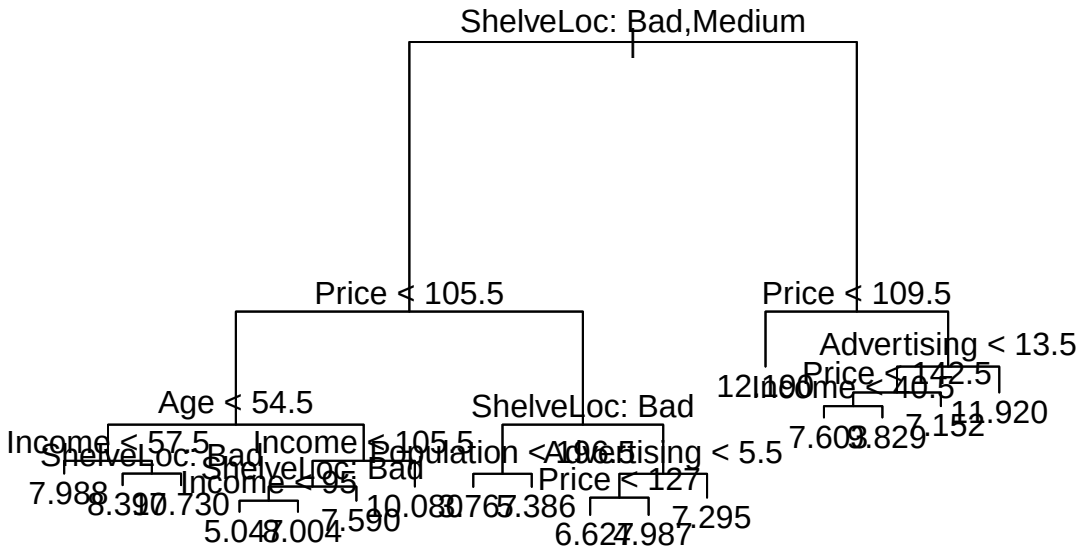
Description of the data:

- **Sales:** unit sales in thousands;
- **CompPrice:** price charged by competitor at each location;
- **Income:** community income level in 1000s of dollars;
- **Advertising:** local ad budget at each location in 1000s of dollars;
- **Population:** regional pop in thousands;
- **Price:** price for car seats at each site;
- **ShelveLoc:** Bad, Good or Medium indicates quality of shelving location;
- **Age:** age level of the population;
- **Education:** education level at location;
- **Urban:** Yes/No;
- **US:** Yes/No.

```
## fit the tree
tree.carseats <- tree(Sales ~ .-Sales, data=Carseats)

## see the tree
summary(tree.carseats)
```

```
##
## Regression tree:
## tree(formula = Sales ~ . - Sales, data = Carseats)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "Income"      "Population"
## [6] "Advertising"
## Number of terminal nodes:  17
## Residual mean deviance:  2.878 = 1102 / 383
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.98700 -1.23000 -0.06125  0.00000  1.22500  4.75400
```

```
plot(tree.carseats)
text(tree.carseats, pretty = 0)
```




```
tree.carseats
```

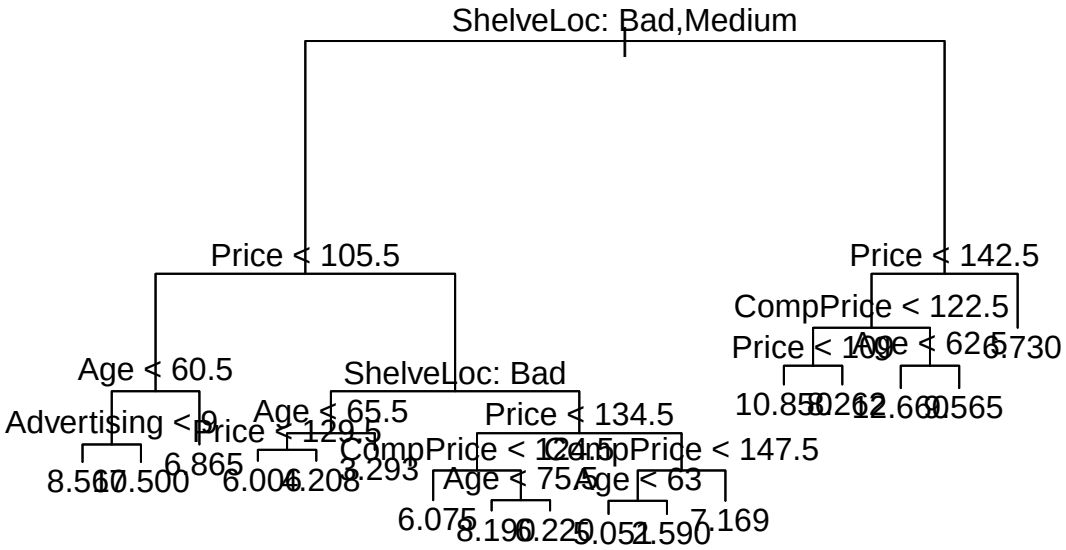
```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 400 3182.0000  7.496
##    2) ShelveLoc: Bad,Medium 315 1860.0000  6.763
##      4) Price < 105.5 108  568.6000  8.189
##        8) Age < 54.5 43  158.7000  9.413
##         16) Income < 57.5 13   19.2400  7.988 *
##         17) Income > 57.5 30  101.6000 10.030
##           34) ShelveLoc: Bad 9   22.7600  8.397 *
##           35) ShelveLoc: Medium 21   44.5300 10.730 *
##        9) Age > 54.5 65  303.1000  7.380
##         18) Income < 105.5 56  203.0000  6.946
##           36) ShelveLoc: Bad 20   76.9600  5.786
##             72) Income < 95 15   43.8100  5.047 *
##             73) Income > 95 5    0.3687  8.004 *
##           37) ShelveLoc: Medium 36   84.2400  7.590 *
##         19) Income > 105.5 9   23.8200 10.080 *
##      5) Price > 105.5 207  956.6000  6.019
##       10) ShelveLoc: Bad 61  240.8000  4.722
##         20) Population < 196.5 25   88.2300  3.767 *
##         21) Population > 196.5 36  113.9000  5.386 *
##       11) ShelveLoc: Medium 146  570.4000  6.560
##         22) Advertising < 5.5 77  280.1000  5.902
##           44) Price < 127 43   95.5200  6.627 *
##           45) Price > 127 34  133.5000  4.987 *
##         23) Advertising > 5.5 69  219.8000  7.295 *
##    3) ShelveLoc: Good 85  525.5000 10.210
##      6) Price < 109.5 28   85.5800 12.190 *
##      7) Price > 109.5 57  277.3000  9.244
##       14) Advertising < 13.5 48  185.4000  8.743
##         28) Price < 142.5 36  108.3000  9.272
##           56) Income < 40.5 9    9.8280  7.603 *
##           57) Income > 40.5 27   65.0600  9.829 *
##         29) Price > 142.5 12   36.6500  7.152 *
```

```
##       15) Advertising > 13.5 9   15.2700 11.920 *
```

Use Cross-Validation to Prune a Tree

Let's create a training set and a test by splitting the **Carseats** dataframe into 250 training and 150 test samples.

```
## sampling for training set
set.seed(333)
train <- sample(1:nrow(Carseats), 250)

## refit the tree using training set
train.carseats <- tree(Sales ~.-Sales, Carseats, subset = train)
plot(train.carseats)
text(train.carseats, pretty = 0)
```

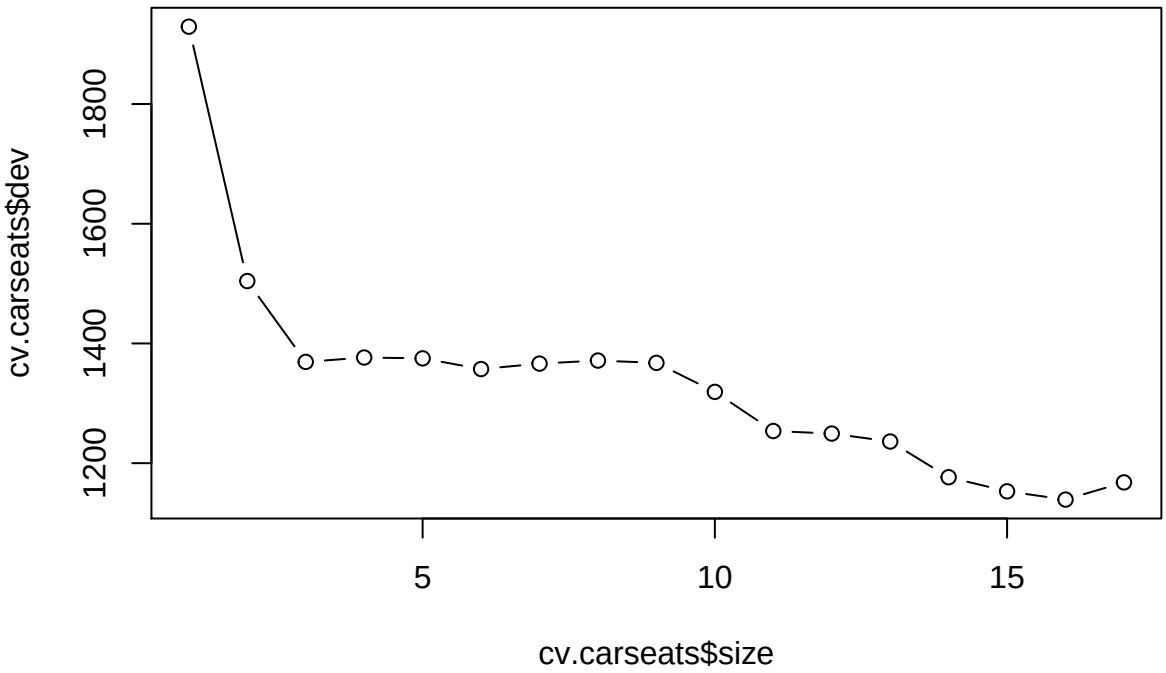



```
## predict on the test set
tree.pred <- predict(train.carseats, newdata = Carseats[-train,])

## evaluate the error, estimate MSE
y = Carseats[-train, "Sales"]
print(mean((y-tree.pred)^2))
```

```
## [1] 4.874415
```

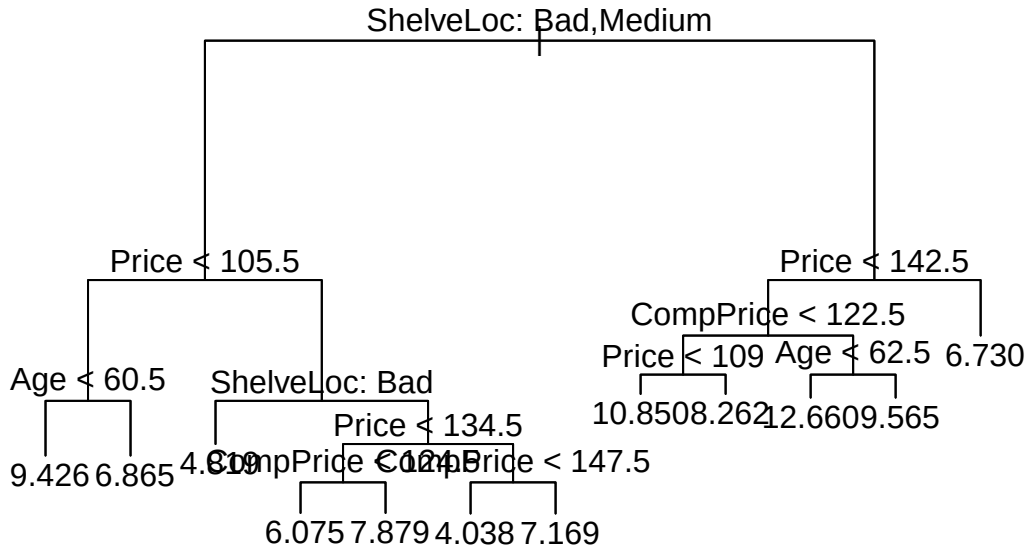
Next, let's use cross-validation to **prune** the tree optimally. We now prune the tree using cross-validation with the `cv.tree()` function.

```
## cross validation
cv.carseats <- cv.tree(train.carseats)
plot(cv.carseats$size, cv.carseats$dev, type='b')
```



The cross-validation results suggests that the most complex tree is the best one. We can try pruning this tree to keep 12 terminal nodes, so let's prune the tree to **size=12**.

```
## prune the tree
prune.carseats <- prune.tree(train.carseats, best = 12)
plot(prune.carseats)
text(prune.carseats, pretty=0)
```



It's a bit shallower than previous trees, and you can actually read the labels. Let's evaluate it on the test dataset again.

```
## predict on test set
tree.pred.cv <- predict(prune.carseats, newdata = Carseats[-train,])

## evaluate the error, estimate MSE
y <- Carseats[-train, "Sales"]
```

```
print(mean((y-tree.pred.cv)^2))
```

```
## [1] 5.225646
```

Seems like the MSE increase a little bit, but it gives you a simpler tree.

Bagging

In bagging, we use bootstrapping to generate B separate training sets and train a tree on each of them. The predictions are then averaged. For each training set, the data not selected is known as the out-of-bag sample, and is used to evaluate the prediction. The average prediction is given by:

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

Bagging overcomes a shortcoming of single decision trees that can give different tree structures when built on different samples of the input data. We use the `randomForest` package to realize that in R.

```
## library packages
library(randomForest)
```

```
## randomForest 4.6-14
```

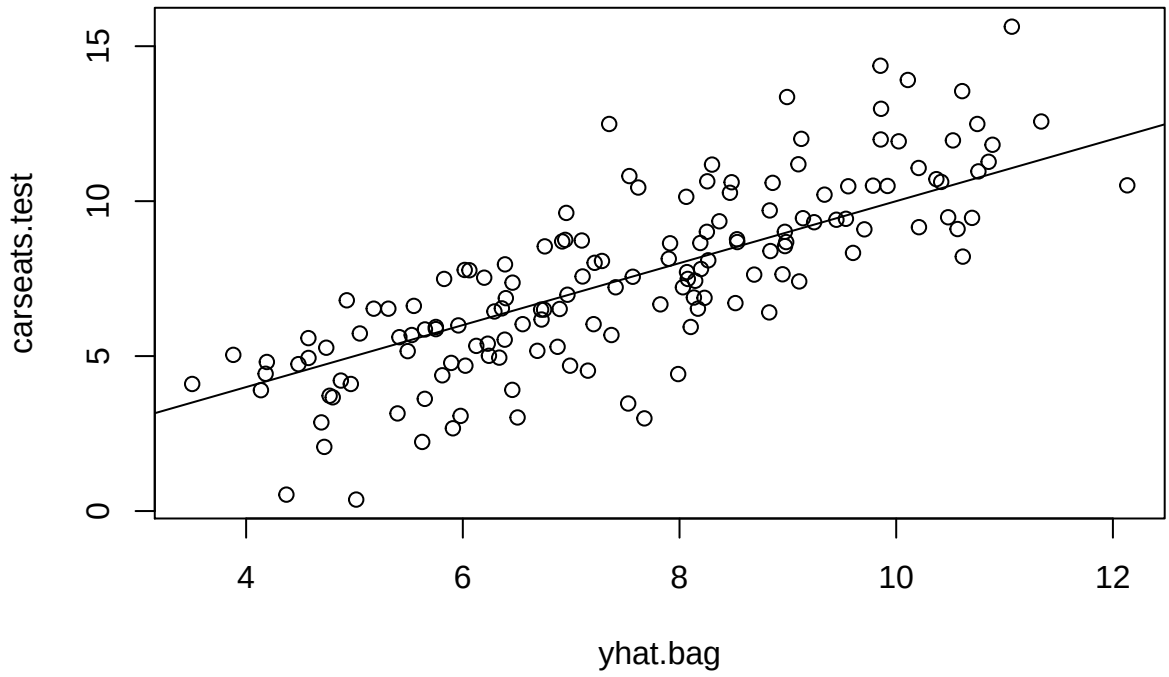
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
## fit the model
bag.carseats <- randomForest(Sales ~.-Sales, data=Carseats, subset=train, mtry=ncol(Carseats)-1, importa
bag.carseats
```

```
##
## Call:
##  randomForest(formula = Sales ~ . - Sales, data = Carseats, mtry = ncol(Carseats) -      1, importan
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 2.547739
##                     % Var explained: 66.67
```

This spawns 500 trees. Let's evaluate the prediction of the bagged model on the test set

```
## predict on test set
yhat.bag <- predict(bag.carseats, newdata=Carseats[-train,])

## plot
carseats.test <- Carseats[-train, "Sales"]
plot(yhat.bag, carseats.test)
abline(0,1)
```




```
## estimate MSE
print(mean((carseats.test-yhat.bag)^2))
```

```
## [1] 3.250705
```

You will find the MSE is largely reduced comparing with a single Tree.

Random Forest

Random forests is similar to **bagging**, except for each tree, a subset $m = \sqrt{p}$ of the total number of predictors are used.

```
## choose m=sqrt(p)
m = round(sqrt(ncol(Carseats)-1))
m
```

```
## [1] 3
```

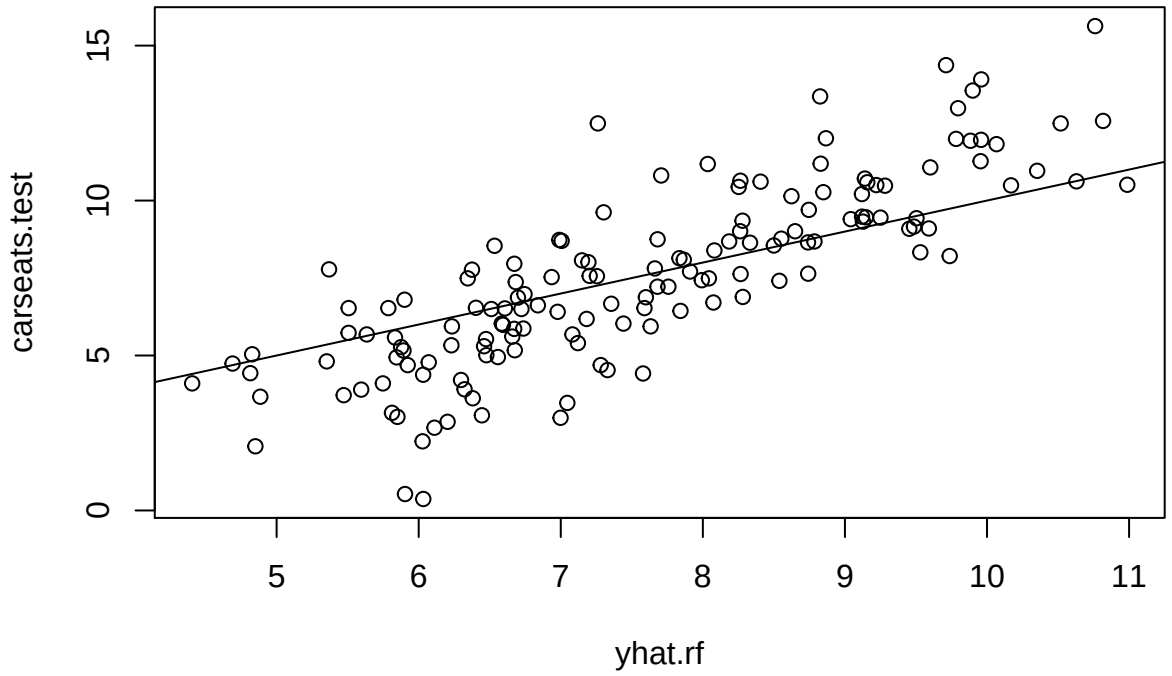
```
## fit a Random Forest model for training set, `mtry=3`
rf.carseats = randomForest(Sales ~ .-Sales, data=Carseats, mtry=m, subset=train, importance=TRUE)
rf.carseats
```

```
##
## Call:
##  randomForest(formula = Sales ~ . - Sales, data = Carseats, mtry = m,      importance = TRUE, subset
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 2.971747
##                     % Var explained: 61.12
```

```
## predict on test set
yhat.rf = predict(rf.carseats, newdata=Carseats[-train, ])
## estimate the MSE
round(mean((carseats.test  - yhat.rf)^2),2)
```

```
## [1] 3.48
```

```
## plot
plot(yhat.rf, carseats.test)
abline(0,1)
```



Multiple Mutation

We will use `mice` function to do multiple imputations when there are lots of missing values.

```
## library packages
library(mice)
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(missForest)
```

```
## Loading required package: foreach
```

```
## Loading required package: itertools
```

```
## Loading required package: iterators
```

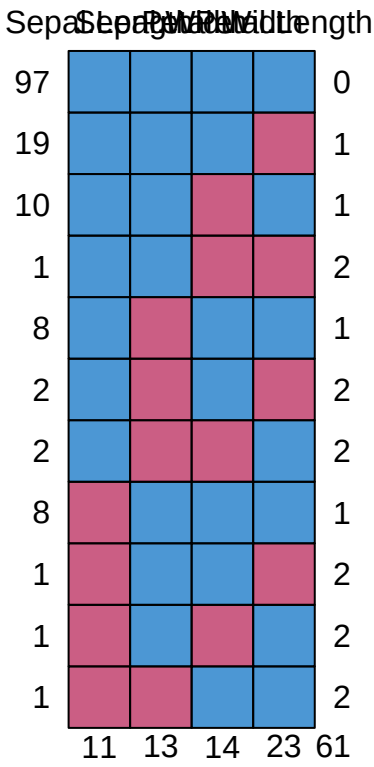
```
## load data
data("iris")
```

Then, let's seed missing values in our data set using `prodNA` function in `missForest` package.

```
## generate 10% missing values at Random
set.seed(333)
iris.mis <- prodNA(iris, noNA = 0.1)
summary(iris.mis)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.400   Median :1.300
##  Mean   :5.839   Mean   :3.053   Mean   :3.787   Mean   :1.177
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.700   Max.   :2.500
##  NA's   :11      NA's   :13      NA's   :23      NA's   :14
##        Species
##  setosa    :46
##  versicolor:45
##  virginica :45
##  NA's      :14
##
##
##
```

```
## remove categorical variables
iris.mis <- subset(iris.mis, select = -c(Species))

## check missing values using `md.pattern()` in `mice` package
md.pattern(iris.mis)
```




```
##    Sepal.Length Sepal.Width Petal.Width Petal.Length
## 97            1           1           1            1  0
```

```
## 19            1            1            1            0  1
## 10            1            1            0            1  1
## 1             1            1            0            0  2
## 8             1            0            1            1  1
## 2             1            0            1            0  2
## 2             1            0            0            1  2
## 8             0            1            1            1  1
## 1             0            1            1            0  2
## 1             0            1            0            1  2
## 1             0            0            1            1  2
##              11           13           14           23 61
```

There are 97 observations with no missing values. There are 19 observations with missing values in Sepal.Length. Similarly, there are 10 missing values with Sepal.Width and so on.

Now, let's use `mice` to impute missing values

```
## imputation
imputed_Data <- mice(iris.mis, m=5, maxit = 50, method = 'pmm', seed = 333)
```

```
##
##  iter imp variable
##   1   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   6   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   6   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   6   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   6   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   6   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   7   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   7   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   7   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   7   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
```

```
##   7   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   8   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   8   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   8   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   8   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   8   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   9   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   9   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   9   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   9   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   9   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  10   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  10   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  10   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  10   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  10   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  11   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  11   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  11   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  11   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  11   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  12   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  12   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  12   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  12   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  12   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  13   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  13   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  13   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  13   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  13   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  14   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  14   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  14   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  14   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  14   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  15   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  15   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  15   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  15   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  15   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  16   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  16   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  16   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  16   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  16   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  17   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  17   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  17   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  17   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  17   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  18   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  18   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##  18   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
```

```
##   18   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   18   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   19   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   19   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   19   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   19   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   19   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   20   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   20   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   20   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   20   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   20   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   21   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   21   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   21   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   21   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   21   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   22   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   22   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   22   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   22   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   22   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   23   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   23   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   23   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   23   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   23   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   24   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   24   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   24   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   24   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   24   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   25   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   25   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   25   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   25   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   25   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   26   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   26   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   26   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   26   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   26   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   27   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   27   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   27   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   27   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   27   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   28   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   28   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   28   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   28   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   28   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   29   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   29   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
```

```
##   29   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   29   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   29   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   30   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   30   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   30   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   30   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   30   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   31   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   31   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   31   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   31   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   31   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   32   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   32   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   32   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   32   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   32   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   33   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   33   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   33   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   33   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   33   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   34   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   34   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   34   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   34   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   34   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   35   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   35   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   35   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   35   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   35   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   36   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   36   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   36   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   36   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   36   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   37   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   37   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   37   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   37   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   37   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   38   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   38   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   38   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   38   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   38   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   39   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   39   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   39   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   39   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   39   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   40   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
```

```
##   40   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   40   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   40   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   40   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   41   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   41   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   41   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   41   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   41   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   42   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   42   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   42   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   42   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   42   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   43   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   43   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   43   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   43   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   43   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   44   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   44   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   44   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   44   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   44   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   45   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   45   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   45   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   45   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   45   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   46   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   46   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   46   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   46   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   46   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   47   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   47   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   47   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   47   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   47   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   48   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   48   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   48   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   48   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   48   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   49   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   49   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   49   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   49   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   49   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   50   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   50   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   50   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   50   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   50   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
```

```
summary(imputed_Data)
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        "pmm"        "pmm"        "pmm"        "pmm"
## PredictorMatrix:
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length            0           1            1           1
## Sepal.Width             1           0            1           1
## Petal.Length            1           1            0           1
## Petal.Width             1           1            1           0
```

Notes: - `m=5` refers to the number of imputed datasets is 5; - `method = 'pmm'` refers to the imputation method is **predictive mean matching**, type `methods(mice)` to check others. - `maxit=50` refers to the no. of iterations taken to impute missing values is 50

```
## check imputed data
imputed_Data$imp$Sepal.Length
```

```
##       1   2   3   4   5
## 2   4.9 4.8 4.8 5.0 4.7
## 34  4.7 5.0 5.0 5.4 4.6
## 40  4.8 4.8 4.8 5.4 5.0
## 45  5.4 5.6 5.2 5.5 5.7
## 58  5.4 5.1 5.0 5.4 5.4
## 67  6.0 6.9 6.6 5.4 6.7
## 105 6.4 6.3 6.2 6.9 6.9
## 113 6.1 6.7 7.7 6.8 6.3
## 117 6.3 6.4 6.9 6.3 6.1
## 123 6.8 7.2 7.2 7.4 7.3
## 134 6.4 7.0 6.4 6.3 6.3
```

```
imputed_Data$imp$Sepal.Width
```

```
##       1   2   3   4   5
## 10  2.8 3.0 2.3 3.2 3.4
## 11  4.1 3.8 3.5 3.8 3.9
## 25  2.8 2.9 3.4 2.7 3.4
## 34  2.8 3.3 3.6 3.4 2.9
## 54  3.1 2.4 2.8 3.0 2.8
## 65  3.3 3.0 2.8 3.0 3.1
## 78  3.4 3.1 3.9 3.0 3.2
## 107 2.7 2.9 2.2 3.0 2.2
## 116 2.3 2.8 3.4 3.1 3.1
## 122 2.8 2.7 2.5 2.7 3.3
## 127 3.4 2.9 3.2 3.3 3.1
## 141 3.0 3.2 3.5 2.9 2.0
## 145 2.8 3.4 3.0 3.1 3.1
```

Now we can get back the completed dataset using the `complete()` function.

```
## complete dataset, check the 1st of the 5 datasets
compdata <- complete(imputed_Data, 1)
summary(compdata)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.400   Median :1.300
##  Mean   :5.832   Mean   :3.051   Mean   :3.767   Mean   :1.198
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.700   Max.   :2.500
```

Our next step is to fit a linear model to the data. There are five imputed datasets, and we need fit a model for each of them and pool the results together.

```
## pooling
modelFit <- with(imputed_Data, lm(Sepal.Width ~ Sepal.Length + Petal.Width))
modelPool <- pool(modelFit)
summary(modelPool)
```

```
##           term   estimate  std.error statistic       df      p.value
## 1  (Intercept)  1.9646646 0.33762974  5.818992 115.1479 5.427619e-08
## 2 Sepal.Length  0.2758349 0.06969046  3.958000 110.8056 1.337965e-04
## 3  Petal.Width -0.4366388 0.07512717 -5.811996 108.6909 6.255818e-08
```