Linear and nonlinear regression

HUST Bioinformatics course series

Wei-Hua Chen (CC BY-NC 4.0)

15 November, 2022

section 1: TOC

前情提要

- R basics
- R data wrangler
- R plot
- R string, regular expression
- R parallel computing

本次提要

- linear regression
- nonlinear regression
- modeling and prediction
- **K-fold** & **X times** cross-validation
- external validation

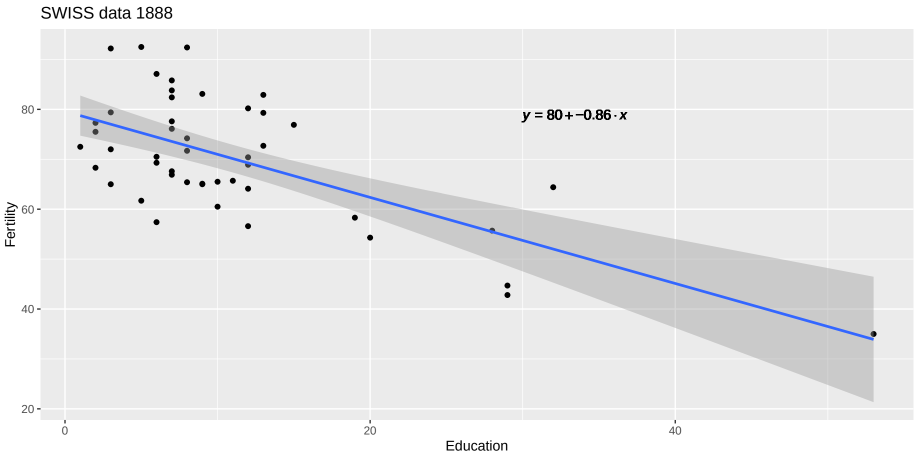
section 2: Linear regression

what is linear regression?

线性回归是利用数理统计中回归分析，来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法

- Y 可以被一个变量 X 解释；一元线性回归
- Y 可以被 X, Z 等多个变量解释；multivariate linear regression

举例


SWISS data 1888
y = 80 + −0.86 · x
Fertility
Education
80
60
40
20
0
20
40

解释

```r
m <- lm(Fertility ~ Education, data = swiss);
summary(m);
```

```
##
## Call:
## lm(formula = Fertility ~ Education, data = swiss)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.036  -6.711  -1.011   9.526  19.689
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  79.6101     2.1041  37.836  < 2e-16 ***
## Education    -0.8624     0.1448  -5.954 3.66e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.446 on 45 degrees of freedom
## Multiple R-squared:  0.4406, Adjusted R-squared:  0.4282
## F-statistic: 35.45 on 1 and 45 DF,  p-value: 3.659e-07
```

lm(

$$targetvariable$$

$\sim$

$$predictorvariables$$

得到 Coefficients

```
coef( m );
```

```
## (Intercept)   Education
##  79.6100585  -0.8623503
```

other useful functions

```
# Other useful functions
coefficients(m) # model coefficients
confint(m, level=0.95) # CIs for model parameters
fitted(m) # predicted values
residuals(m) # residuals
anova(m) # anova table
vcov(m) # covariance matrix for model parameters
influence(m) # regression diagnostics
```

R-squred a.k.a R2 是怎么来的？

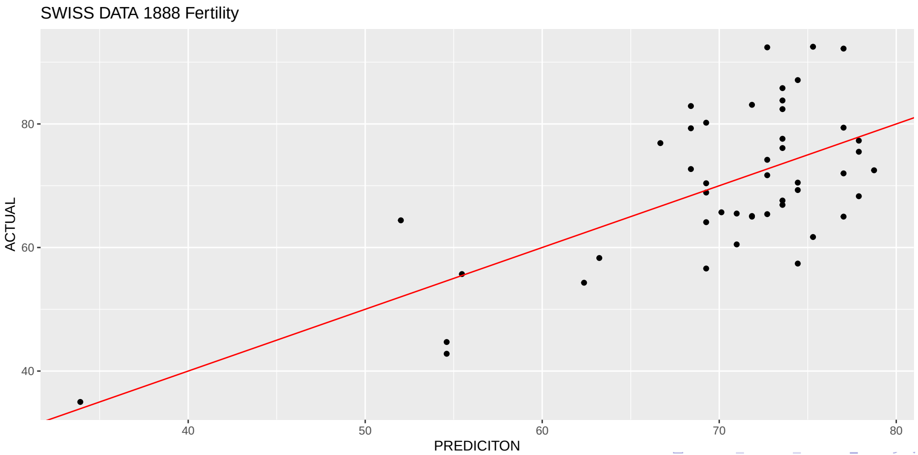
```
library(magrittr);
library(caret);
predictions <- m %>% predict( swiss ); ## or use fitted(m) instead ...

# Model performance
data.frame(
  RMSE = RMSE(predictions, swiss$Fertility),
  R2 = R2(predictions, swiss$Fertility)
)
```

```
##       RMSE        R2
## 1 9.242865 0.4406156
```

RMSE mean squared error, the smaller the better **R2** R, higher the better * F-statistic: Higher the better

R-squred a.k.a R2 是怎么来的？cont.

```
data.frame( PREDICITON = predictions, ACTUAL = swiss$Fertility ) %>%
  ggplot(aes( PREDICITON, ACTUAL )) + geom_point() +
  ggtitle( "SWISS DATA 1888 Fertility" ) +
  geom_abline(intercept = 0, slope = 1, colour = "red")
```



Multivariate linear modeling

datarium package

```
install.packages("datarium");
```

```
library(datarium);
head(marketing);
```

```
##   youtube facebook newspaper sales
## 1  276.12    45.36     83.04 26.52
## 2   53.40    47.16     54.12 12.48
## 3   20.64    55.08     83.16 11.16
## 4  181.80    49.56     70.20 22.20
## 5  216.96    12.96     70.08 15.48
## 6   10.44    58.68     90.00  8.64
```

问题：广告投放在哪里对销售有帮助??

multivariate linear modeling , cont.

```
m1 <- lm( sales ~ youtube + facebook + newspaper, data = marketing );
m2 <- lm( sales ~ youtube, data = marketing);
m3 <- lm( sales ~ facebook, data = marketing);

summary(m1);
```

```
##
## Call:
## lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5932  -1.0690   0.2902   1.4272   3.3951
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.526667   0.374290   9.422   <2e-16 ***
## youtube      0.045765   0.001395  32.809   <2e-16 ***
## facebook     0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.023 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

facebook vs. youtube

```
coef( m1 );
```

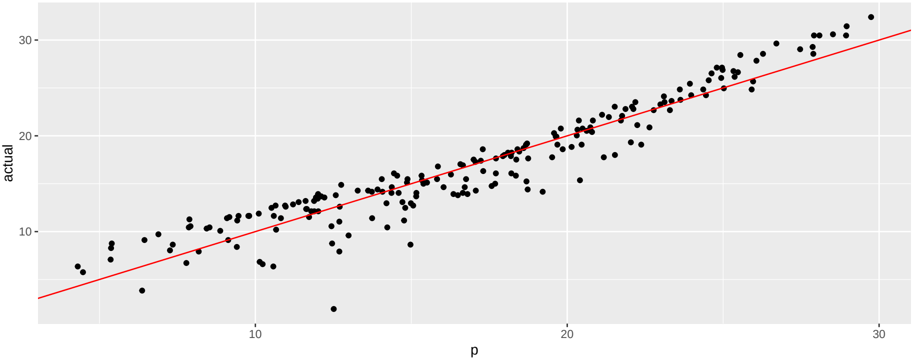
```
##   (Intercept)      youtube     facebook    newspaper
##   3.526667243  0.045764645  0.188530017 -0.001037493
```

```
data.frame( YOUTUBE = summary(m2)$r.squared, FACEBOOK = summary(m3)$r.squared);
```

```
##     YOUTUBE  FACEBOOK
## 1 0.6118751 0.3320325
```

predicted vs. actual

```r
predicted.m1 <- m1 %>% predict( marketing );
data.frame(p = predicted.m1, actual = marketing$sales ) %>%
  ggplot( aes(x = p, y = actual) ) + geom_point() +
  geom_abline(intercept = 0, slope = 1, colour = "red");
```



performance evaluation

```r
# Model performance
data.frame(
  RMSE = RMSE(predicted.m1, marketing$sales ),
  R2 = R2(predicted.m1, marketing$sales )
)
```

```
##       RMSE        R2
## 1 2.002284 0.8972106
```

get rid of newspaper

```
m4 <- lm( sales ~ youtube + facebook, data = marketing );
anova(m1, m4);
```

```
## Analysis of Variance Table
##
## Model 1: sales ~ youtube + facebook + newspaper
## Model 2: sales ~ youtube + facebook
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    196 801.83
## 2    197 801.96 -1  -0.12775 0.0312 0.8599
```

```
data.frame(
  with_newspapers = summary(m1)$r.squared,
  without_newspapers = summary(m4)$r.squared
)
```

```
##   with_newspapers without_newspapers
## 1       0.8972106          0.8971943
```

relative importance analysis

```r
library(relaimpo);
calc.relimp( sales ~ youtube + facebook + newspaper, data = marketing );
```

```
## Response variable: sales
## Total response variance: 39.19947
## Analysis based on 200 observations
##
## 3 Regressors:
## youtube facebook newspaper
## Proportion of variance explained by model: 89.72%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##                  lmg
## youtube   0.58527298
## facebook  0.28878652
## newspaper 0.02315114
##
## Average coefficients for different model sizes:
##
##                   1X        2Xs          3Xs
## youtube   0.04753664 0.04632801  0.045764645
## facebook  0.20249578 0.19351941  0.188530017
## newspaper 0.05469310 0.02543180 -0.001037493
```

interactions

interactions 考虑因素之间的依赖关系或互作关系，比如，在一平台上投放广告会促进另一个平台上广告的效果，因为两个平台的用户可能是重叠的。他们在两个平台都看到广告时，更可能购买产品。

```
m5 <- lm( sales ~ youtube + facebook + youtube:facebook, data = marketing );
anova(m4, m5);
```

```
## Analysis of Variance Table
##
## Model 1: sales ~ youtube + facebook
## Model 2: sales ~ youtube + facebook + youtube:facebook
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    197 801.96
## 2    196 251.26  1     550.7 429.59 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
data.frame(
  no_interactions = summary(m4)$r.squared,
  with_interactions = summary(m5)$r.squared
)
```

```
##   no_interactions with_interactions
## 1       0.8971943         0.9677905
```

interactions, cont.

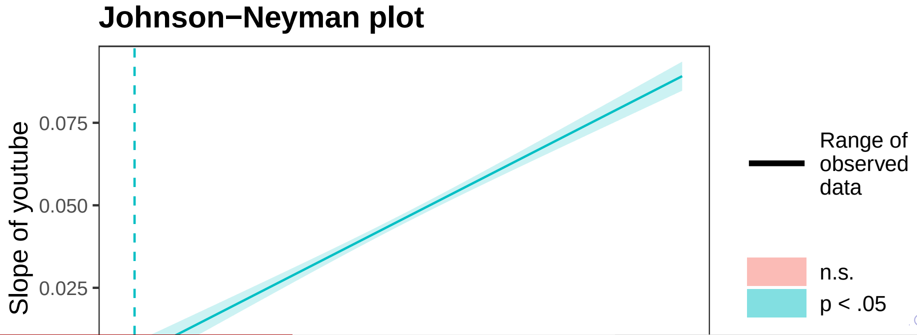
```r
## m5 <- lm( sales ~ youtube + facebook + youtube:facebook, data = marketing );

## 上面的 m5 可以直接写为:
m6 <- lm( sales ~ youtube*facebook, data = marketing );
summary(m6);
```

```
##
## Call:
## lm(formula = sales ~ youtube * facebook, data = marketing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6039 -0.4833  0.2197  0.7137  1.8295
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      8.100e+00  2.974e-01  27.233   <2e-16 ***
## youtube          1.910e-02  1.504e-03  12.699   <2e-16 ***
## facebook         2.886e-02  8.905e-03   3.241   0.0014 **
## youtube:facebook 9.054e-04  4.368e-05  20.727   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.132 on 196 degrees of freedom
## Multiple R-squared:  0.9678, Adjusted R-squared:  0.9673
## F-statistic:  1963 on 3 and 196 DF,  p-value: < 2.2e-16
```

visualize interactions

```
## install.packages("interactions"); 如需要，请安装这个包
library(interactions); ## 装入
sim_slopes(m6, pred = youtube, modx = facebook, jnplot = TRUE)
```

```
## JOHNSON-NEYMAN INTERVAL
##
## When facebook is OUTSIDE the interval [-26.70, -16.41], the slope of
## youtube is p < .05.
##
## Note: The range of observed values of facebook is [0.00, 59.52]
```



relative importance analysis including interactions

```
library(relaimpo);
calc.relimp( sales ~ youtube*facebook, data = marketing );
```

```
## Response variable: sales
## Total response variance: 39.19947
## Analysis based on 200 observations
##
## 3 Regressors:
## youtube facebook youtube:facebook
## Proportion of variance explained by model: 96.78%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##                         lmg
## youtube          0.58851843
## facebook         0.30867583
## youtube:facebook 0.07059629
##
## Average coefficients for different model sizes:
##
##                          1X        2Xs          3Xs
## youtube          0.04753664 0.04575482 0.0191010738
## facebook         0.20249578 0.18799423 0.0288603399
## youtube:facebook        NaN        NaN 0.0009054122
```

assumptions of `linear regression`

重要信息

1. 任何检验都有基本的假设
2. 将检验应用于不符合假设的数据是统计学最大的滥用

assumptions

1. Linearity: The relationship between X and the mean of Y is linear.
2. Homoscedasticity: The variance of residual is the same for any value of X.
3. Independence: Observations are independent of each other.
4. Normality: For any fixed value of X, Y is normally distributed.

glm vs. lm

```
lm(formula, data, …)
glm(formula, family=gaussian, data, …)
```

glm:

1. 当 `family=gaussian` 时，二者是一样的。

```r
library(texreg);
m.lm <- lm(am ~ disp + hp, data=mtcars);
m.glm <- glm(am ~ disp + hp, data=mtcars);
screenreg(l = list(m.lm, m.glm))
```

```
##
## =====================================
##                  Model 1     Model 2
## -------------------------------------
## (Intercept)       0.76 ***    0.76 ***
##                  (0.16)      (0.16)
## disp             -0.00 ***   -0.00 ***
##                  (0.00)      (0.00)
## hp                0.00 *      0.00 *
##                  (0.00)      (0.00)
## -------------------------------------
## R^2               0.48
## Adj. R^2          0.45
## Num. obs.        32          32
## AIC                          32.13
```

glm 还可用于其它类型数据的分析

1. Logistic regression (family=binomial)

预测的结果（Y）是 binary 的分类，比如 Yes, No，且只能有两个值；

```
dat <- iris %>% filter( Species %in% c("setosa", "virginica") );
bm <- glm( Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
           data = dat, family = binomial );

data.frame( predicted = bm %>% predict( dat, type = "response" ),
            original = dat$Species ) %>% sample_n(6) %>% arrange( original );
```

```
##       predicted  original
## 25 5.282004e-11    setosa
## 33 2.220446e-16    setosa
## 30 4.976501e-12    setosa
## 3  1.369834e-13    setosa
## 85 1.000000e+00 virginica
## 67 1.000000e+00 virginica
```

注意：

`predict(., type = "response")` 的意义是什么 ??

glm 的 Poisson regression (family=poisson)

Poisson regression is a special type of regression in which the response variable consists of **count data**.

Asumptions:

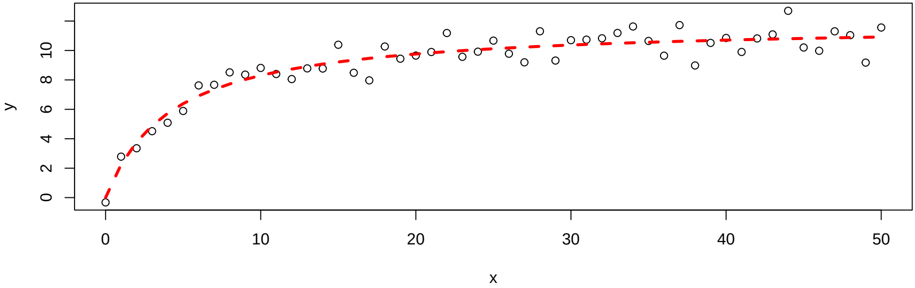
1. The response variable consists of count data.
2. Observations are independent.
3. The mean and variance of the model are equal.
4. The distribution of counts follows a Poisson distribution.

section 3: Non-linear regression (`nls`)

一元 nls

什么是 nls ? to predict **a target variable** using **a non-linear function** consisting of **parameters** and **one or more independent variables**.

non-linear least squares



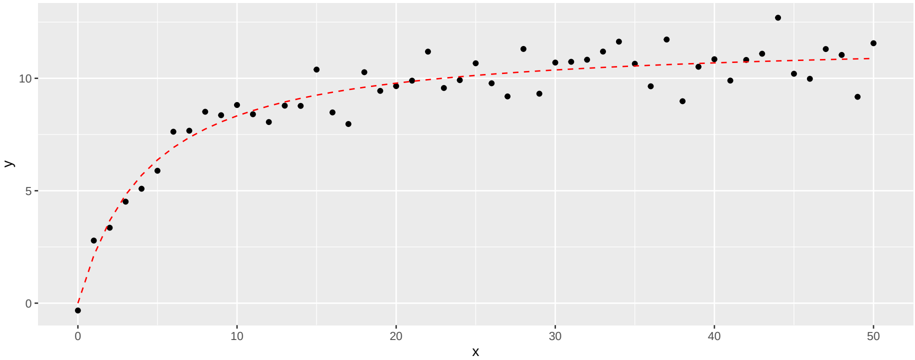
non-linear least squares

```
## 1. generate data
set.seed(20160227)
x<-seq(0,50,1)
y<-((runif(1,10,20)*x)/(runif(1,0,10)+x))+rnorm(51,0,1)

## 2. fit model using nls
m<-nls(y~a*x/(b+x), start = list(a=0.1, b=0.1));

## 3. show how good is the fitting ...
#get some estimation of goodness of fit
plot(x,y)
lines(x,predict(m),lty=2,col="red",lwd=3)
```

1. `nls( equation, data = data, start = ... )`
2. `y~a*x/(b+x)`

non-linear least squares, using drc

```
library(drc); ## Analysis of Dose-Response Curves
m13 <- drm( y ~ x, fct = LL.3() ); ## here LL.3() is a least square function
data.frame( x = x, y = y, fitted = predict(m13) ) %>% ggplot( aes(x=x,y=y) ) +
  geom_point() + geom_line( aes(x = x, y = fitted), colour = "red", linetype = 2 );
```



non-linear functions

Polynomials

- Linear equation
- Quadratic polynomial

Concave/Convex curves (no inflection)

- Exponential equation
- Asymptotic equation
- Negative exponential equation
- Power curve equation
- Logarithmic equation
- Rectangular hyperbola

Sygmoidal curves

- Logistic equation
- Gompertz equation
- Log-logistic equation (Hill equation)

example: Exponential equation

Exponential decay

y = a * (exp(k * X) * k)

install aomisc package:

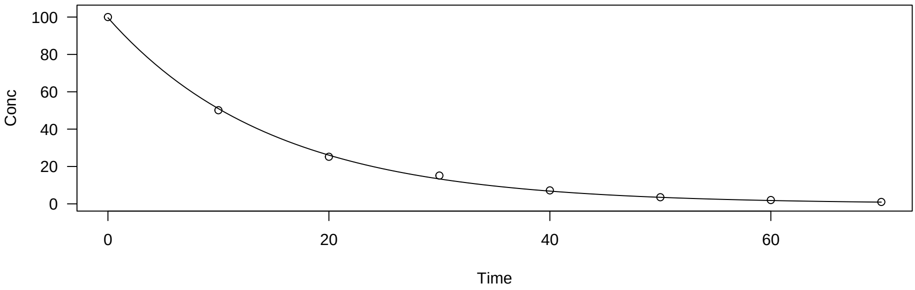
```
install.packages("devtools");
devtools::install_github("onofriAndreaPG/aomisc");
```

```
library(aomisc);
data(degradation);
m14 <- drm(Conc ~ Time, fct = DRC.expoDecay(),
             data = degradation)
summary(m14)
```

```
##
## Model fitted: Exponential Decay Model (2 parms)
##
## Parameter estimates:
##
##                    Estimate Std. Error t-value   p-value
## init:(Intercept) 99.6349312  1.4646680  68.026 < 2.2e-16 ***
## k:(Intercept)     0.0670391  0.0019089  35.120 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

plot Exponential decay

```
plot(m14, log="");
```



Power curve

a * (X^(b - 1) * b)

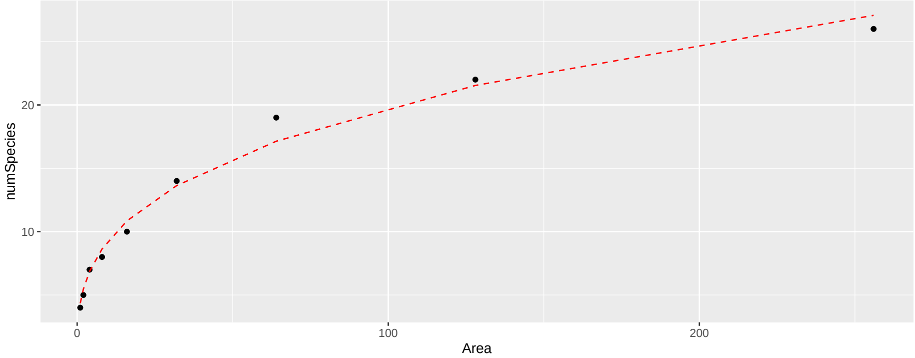
```
data(speciesArea)
m15 <- drm(numSpecies ~ Area, fct = DRC.powerCurve(),
             data = speciesArea)
summary(m15)
```

```
##
## Model fitted: Power curve (Freundlich equation) (2 parms)
##
## Parameter estimates:
##
##                 Estimate Std. Error t-value   p-value
## a:(Intercept) 4.348404   0.337197  12.896 3.917e-06 ***
## b:(Intercept) 0.329770   0.016723  19.719 2.155e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error:
##
##  0.9588598 (7 degrees of freedom)
```

重点：1. DRC.powerCurve 作为建模的参数，来自 library(aomisc)

Power curve 结果作图

```
speciesArea %>% mutate( fitted = predict( m15, speciesArea ) ) %>%
  ggplot( aes(x= Area, y= numSpecies) ) +
  geom_point() + geom_line( aes(x = Area, y = fitted), colour = "red", linetype = 2 );
```



how good is the model?

```
R2(  speciesArea$numSpecies, predict( m15, speciesArea ) );
```

```
## [1] 0.9874392
```

```
## compare with a linear model
m16 <- lm( numSpecies ~ Area, data = speciesArea );
R2(  speciesArea$numSpecies, predict( m16, speciesArea ) );
```

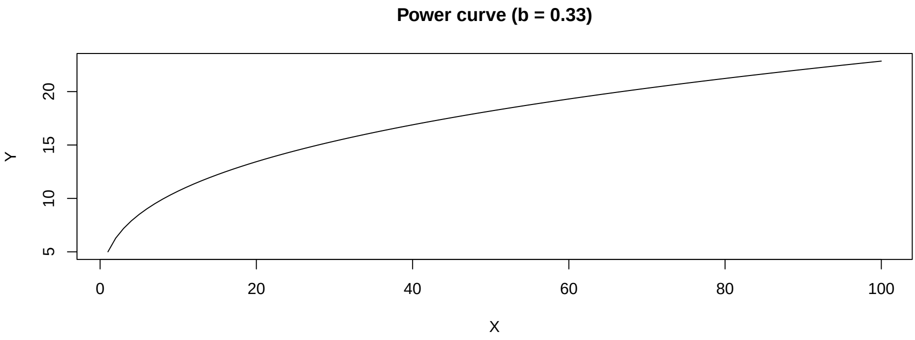
```
## [1] 0.8110041
```

数据生成函数

aomisc 包和 drc 包带了非常多数据生成的函数，其使用示例如下：

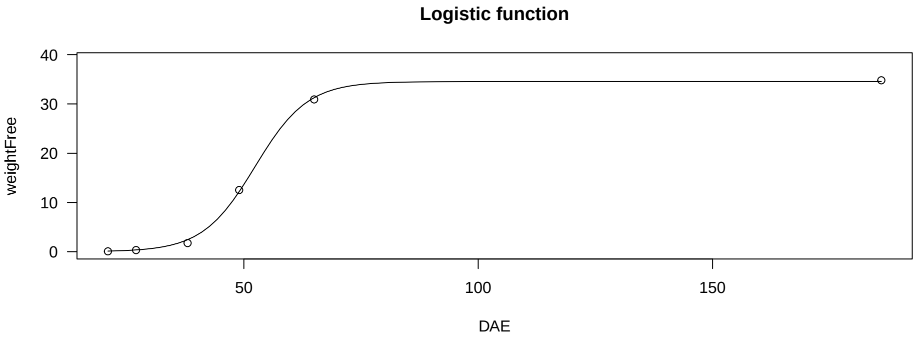
注 a * (X^(b - 1) * b) 需要两个参数，a 和 b

```
plot(powerCurve.fun(1:100, 5, 0.33),
      xlab = "X", ylab = "Y", main = "Power curve (b = 0.33)", type = "l")
```



logistic function

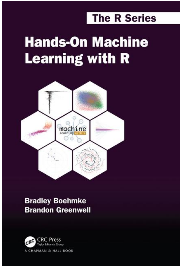
```
data(beetGrowth)
m17 <- drm(weightFree ~ DAE, fct = L.3(), data = beetGrowth)
plot(m17, log="", main = "Logistic function")
```



多元 non-linear regression

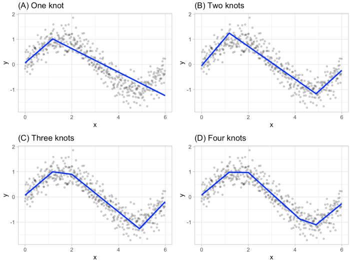
1. `mars` : Multivariate Adaptive Regression Splines (多元自适应回归样条)
2. `machine learning`

推荐一本教程: **`Hands-On Machine Learning with R`**



多元 non-linear regression, cont.

Multivariate adaptive regression splines (MARS) provide a convenient approach to capture the nonlinear relationships in the data by assessing cutpoints (knots). The procedure assesses each data point for each predictor as a knot and creates a linear regression model with the candidate feature(s).



fit a basic MARS model, get data ready...

```r
library(AmesHousing); ## The Ames Iowa Housing Data
ames <- AmesHousing::make_ames();
head(ames);
```

```
## # A tibble: 6 x 81
##   MS_SubC~1 MS_Zo~2 Lot_F~3 Lot_A~4 Street Alley Lot_S~5 Land_~6 Utili~7 Lot_C~8
##   <fct>     <fct>     <dbl>   <int> <fct>  <fct> <fct>   <fct>   <fct>   <fct>
## 1 One_Stor~ Reside~     141   31770 Pave   No_A~ Slight~ Lvl     AllPub  Corner
## 2 One_Stor~ Reside~      80   11622 Pave   No_A~ Regular Lvl     AllPub  Inside
## 3 One_Stor~ Reside~      81   14267 Pave   No_A~ Slight~ Lvl     AllPub  Corner
## 4 One_Stor~ Reside~      93   11160 Pave   No_A~ Regular Lvl     AllPub  Corner
## 5 Two_Stor~ Reside~      74   13830 Pave   No_A~ Slight~ Lvl     AllPub  Inside
## 6 Two_Stor~ Reside~      78    9978 Pave   No_A~ Slight~ Lvl     AllPub  Inside
## # ... with 71 more variables: Land_Slope <fct>, Neighborhood <fct>,
## #   Condition_1 <fct>, Condition_2 <fct>, Bldg_Type <fct>, House_Style <fct>,
## #   Overall_Qual <fct>, Overall_Cond <fct>, Year_Built <int>,
## #   Year_Remod_Add <int>, Roof_Style <fct>, Roof_Matl <fct>,
## #   Exterior_1st <fct>, Exterior_2nd <fct>, Mas_Vnr_Type <fct>,
## #   Mas_Vnr_Area <dbl>, Exter_Qual <fct>, Exter_Cond <fct>, Foundation <fct>,
## #   Bsmt_Qual <fct>, Bsmt_Cond <fct>, Bsmt_Exposure <fct>, ...
```

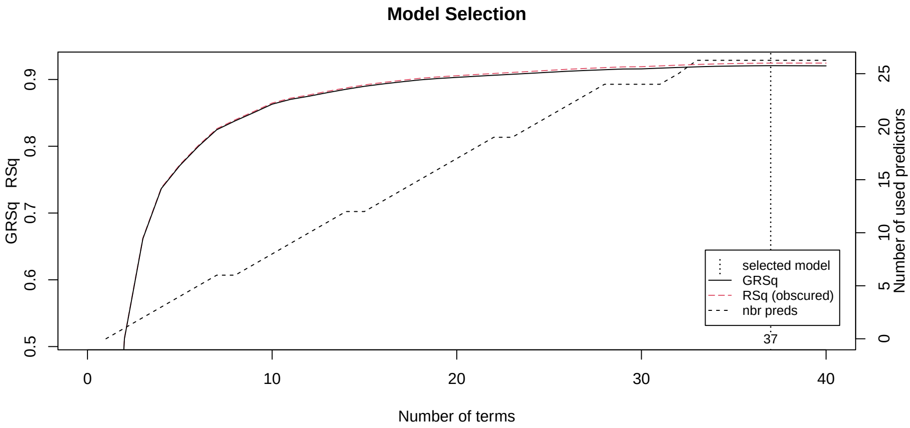
fit a basic MARS model, fit a basic model

```r
library(earth);
mars1 <- earth( Sale_Price ~ .,  data = ames );
print(mars1);
```

```
## Selected 37 of 40 terms, and 26 of 308 predictors
## Termination condition: RSq changed by less than 0.001 at 40 terms
## Importance: Gr_Liv_Area, Year_Built, Total_Bsmt_SF, Overall_QualExcellent, ...
## Number of terms at each degree of interaction: 1 36 (additive model)
## GCV 506531262    RSS 1.411104e+12    GRSq 0.9206569    RSq 0.9245098
```

MARS model, model selection

```
plot(mars1, which = 1);
```



如何解释？

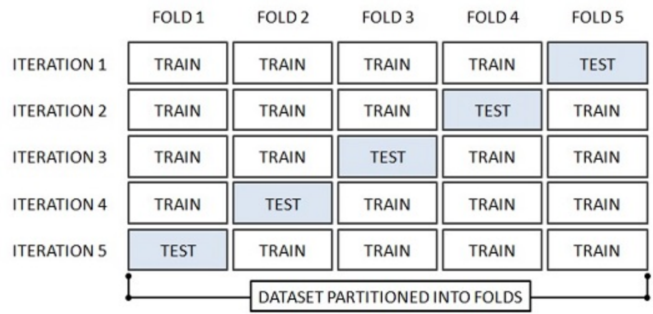
mars model with interactions

```r
mars2 <- earth(Sale_Price ~ .,  data = ames, degree = 2);
summary(mars2) %>% .$coefficients %>% head(10)
```

```
##                                             Sale_Price
## (Intercept)                               3.040042e+05
## h(Gr_Liv_Area-3228)                       1.975129e+02
## h(3228-Gr_Liv_Area)                      -4.415067e+01
## h(Year_Built-2003)                        7.976946e+03
## h(2003-Year_Built)                       -4.970063e+02
## h(Total_Bsmt_SF-2452)                     4.990177e+01
## h(2452-Total_Bsmt_SF)                    -5.482326e+01
## h(Year_Built-2003)*h(2439-Gr_Liv_Area)   -5.515644e+00
## h(2003-Year_Built)*h(Total_Bsmt_SF-1117) -7.288488e-01
## h(2003-Year_Built)*h(1117-Total_Bsmt_SF)  3.682410e-01
```

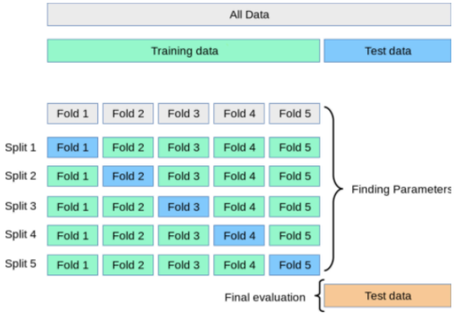
Parameter tuning: cross validation

K fold, N times



注 1. 每个 iteration 为随机 split； 2. 得到 K * N 个模型；

cross validation & additional test



Parameter tuning

```
hyper_grid <- expand.grid(
  degree = 1:3,  ## number of interaction degrees
  nprune = seq(2, 100, length.out = 10) %>% floor() ## number of features to select
)
head(hyper_grid);
```

```
##   degree nprune
## 1      1      2
## 2      2      2
## 3      3      2
## 4      1     12
## 5      2     12
## 6      3     12
```

Parameter tuning, cont.

```
set.seed(123)  # for reproducibility
cv_mars <- train(
  x = subset(ames, select = -Sale_Price),
  y = ames$Sale_Price,
  method = "earth",
  metric = "RMSE",
  trControl = trainControl(method = "cv", number = 10),
  tuneGrid = hyper_grid
)
```

注 10 折 cross validation

注非常 time consuming … 最好能使用 parallel computing …

Parameter tuning, show results

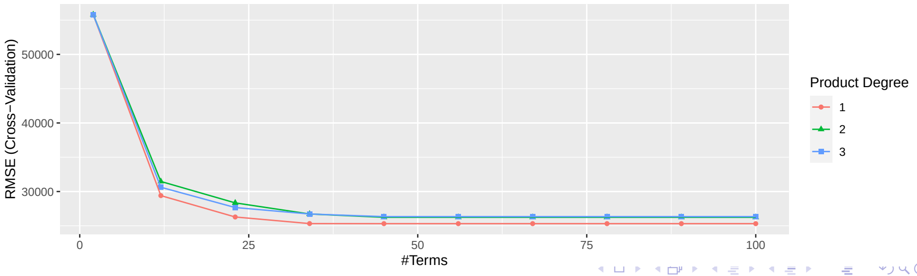
```
cv_mars$bestTune;
```

```
##   nprune degree
## 5     45      1
```

```
cv_mars$results %>%
  filter(nprune == cv_mars$bestTune$nprune, degree == cv_mars$bestTune$degree);
```

```
##   degree nprune     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      1     45 25312.47 0.8968114 16326.29 4233.975 0.04749767 1172.919
```

```
ggplot(cv_mars) ## plot
```



compare with other methods e.g., lm

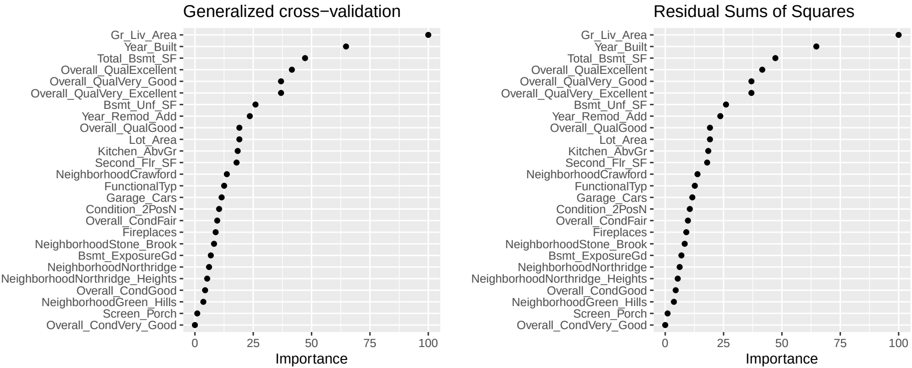
```
cv_lm <- train(
  Sale_Price ~ .,
  data = ames,
  method = "lm",
  trControl = trainControl(method = "repeatedcv", number = 10, repeats = 3)
);
```

```
print(cv_lm);
```

```
## Linear Regression
##
## 2930 samples
##   80 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 2637, 2637, 2637, 2637, 2637, 2638, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   35829.97  0.8050749  16411.61
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

variable importance plot (VIP)

```r
library(vip);
p1 <- vip(cv_mars, num_features = 40, geom = "point", value = "gcv") + ggtitle("Generalized cro
p2 <- vip(cv_mars, num_features = 40, geom = "point", value = "rss") + ggtitle("Residual Sums o

gridExtra::grid.arrange(p1, p2, ncol = 2)
```



mars : final thoughts

advantages

1. First, MARS naturally handles mixed types of predictors (quantitative and qualitative).
2. MARS also requires minimal feature engineering (e.g., feature scaling) and performs automated feature selection.
3. Highly correlated predictors do not impede predictive accuracy as much as they do with OLS models.

shortcomings

1. typically slower to train
2. Also, although correlated predictors do not necessarily impede model performance, they can make model interpretation difficult.

section 4: 小结及作业!

本次小结

linear regression

- lm vs. glm
- 一元
- 多元
- 相关函数
- performance evaluation
- interactions
- visualizations

non-linear regression

- nls
- mars
- earth
- cross validation
- K fold, N times (下次详细讲)

下次预告

- Random Forest
- Support Vector Machine
- Deep learning

作业

- `Exercises and homework` 目录下 `talk11-homework.Rmd` 文件;
- 完成时间：见钉群的要求

important

- all codes are available at Github: `https://github.com/evolgeniusteam/R-for-bioinformatics`