R基础统计及线性模型

陈华珊 (中国社科院社会发展战略研究院)

基础统计

◆ 描述统计

❖ 向量

`mean(x)`	Mean of x
`median(x)`	Median of x
`var(x)`	Variance of x
`sd(x)`	Standard deviation of x
`cov(x,y)`	Covariance of x and y
`cor(x,y)`	Correlation of x and y
`min(x)`	Minimum of x
`max(x)`	Maximum of x
`range(x)`	Range of x
`quantile(x)`	Quantiles of x for the given probabilities

```
x <- rnorm(10)
y <- runif(10)
mean(x)
```

```
## [1] -0.2526202
cor(x,y)
## [1] 0.2570403
```

❖ data.frame的变量（列）

summary()	对于每列，显示基本信息
apply()	对于每列，执行一个函数
tapply()	拆分为多个子集，执行一个函数，返回数组
by()	类似于 tapply()``，返回by` 类
ave()	类似于 `tapply()``, 返回向量
aggregate()	类似于 `tapply()``, 返回 dataframe

❖ 示例

```
# Basic summary statistics for each column
summary(gss)

# Variance of first two columns
apply(gss[,1:2], 2, var)

# Mean family income by region
tapply(gss$a8a, gss$s41, FUN=mean) # Return a vector
by(gss$a8a, gss$s41, FUN=mean) # Return a by object,

# extract elements using []
ave(gss$a8a, gss$s41, FUN=mean) # Return a vector, same

# length as first argument
aggregate(gss$a8a, list(gss$s41) , FUN=mean) # Return a dataframe, subset
```

```
# variable needs to be a list
# Mean inc by region and area
with(gss, tapply(a8a, list(s41, s5a), FUN=mean))
with(gss, by(a8a, list(s41, s5a), FUN=mean))
with(gss, ave(a8a, list(s41, s5a), FUN=mean))
with(gss, aggregate(a8a, list(s41, s5a), FUN=mean))
```

❖ 使用第三方包

- `Hmisc::describe()` 返回观测数量、缺失值和唯一值的数目、平均值、分位数，以及五个最大值和最小值;
- `pastecs::stat.desc()` 返回所有值、空值、缺失值的数量，最小值、最大值、值域、标准差、均值，均值95%置信区间等;
- `psych::describe()`
- `doBy::summaryBy()`

◆ 频次表/列联表

❖ 相关函数

`table(var1, var2, ..., varN)`	使用N个类别变量（因子）创建一个N维列联表
`xtabs(formula, data)`	根据一个公式和一个矩阵/数据框创建一个N维列联表
`prop.table(table, margins)`	百分比列联表，行/列
`margin.table(table, margins)`	边际求和
`addmargins(table, margins)`	包含边际求和
`ftable(table)`	“平铺式”列联表

❖ 示例

VCD::Arthritis数据集：关于一项风湿性关节炎新疗法的双盲临床试验结果。

```
library(vcd)
mytable = with(Arthritis, table(Improved))
mytable
## Improved
##   None   Some Marked
##     42     14     28

# 转化为百分比
prop.table(mytable)
## Improved
##      None      Some    Marked
## 0.5000000 0.1666667 0.3333333

# 二维列联表
mytable = xtabs(~Treatment + Improved, data = Arthritis)
```

```
mytable
##            Improved
## Treatment None Some Marked
##   Placebo   29    7      7
##   Treated   13    7     21
margin.table(mytable, 1)
## Treatment
## Placebo Treated
##      43      41
prop.table(mytable, 1)
##            Improved
## Treatment       None      Some    Marked
##   Placebo 0.6744186 0.1627907 0.1627907
##   Treated 0.3170732 0.1707317 0.5121951
addmargins(mytable)
##            Improved
## Treatment None Some Marked Sum
##   Placebo   29    7      7  43
```

```
##   Treated   13    7      21  41
##   Sum       42   14      28  84
addmargins(prop.table(mytable))
##            Improved
## Treatment        None       Some     Marked        Sum
##   Placebo 0.34523810 0.08333333 0.08333333 0.51190476
##   Treated 0.15476190 0.08333333 0.25000000 0.48809524
##   Sum     0.50000000 0.16666667 0.33333333 1.00000000
###useNA = 'ifany'
```

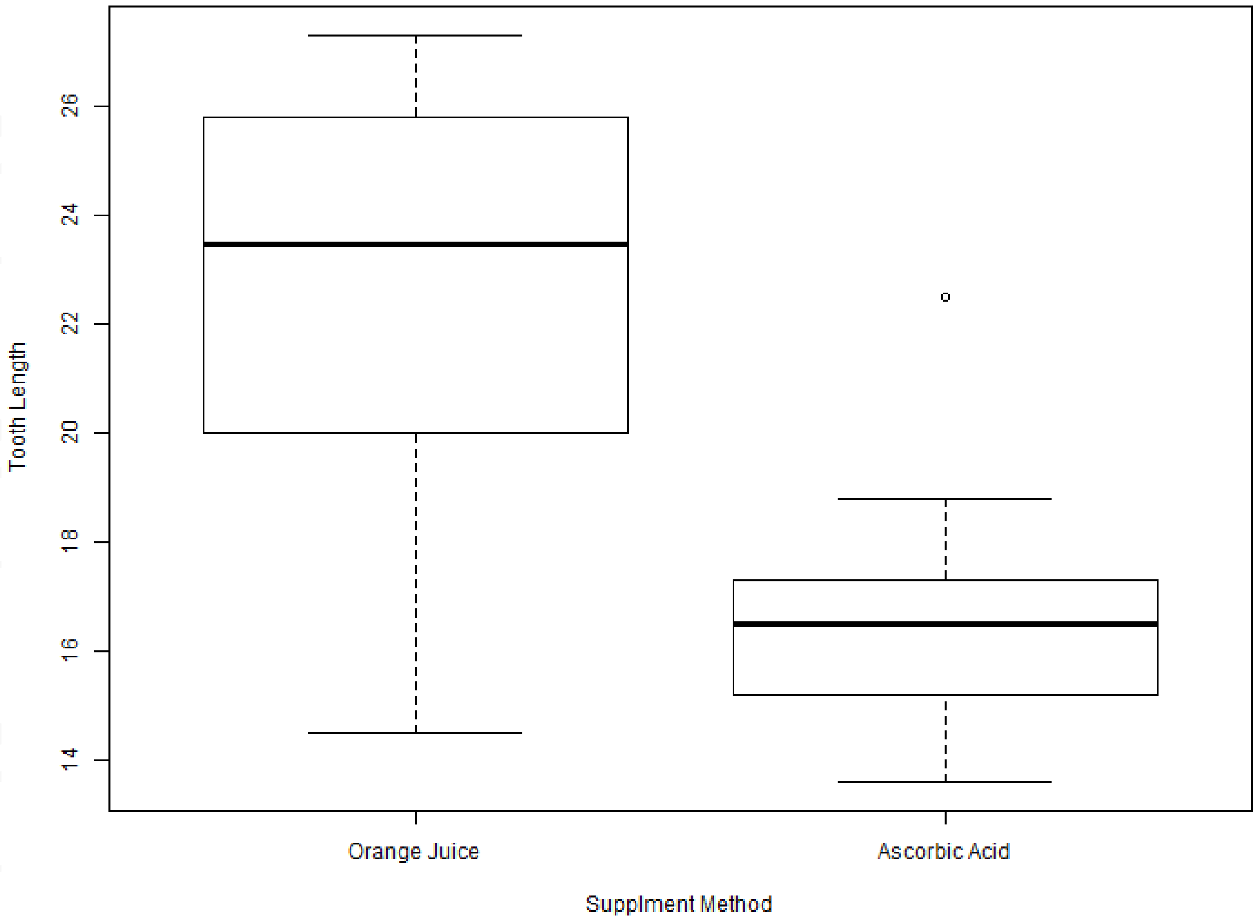
◆ 均值比较

t – t **–test**

```
t.test()
```

❖ 示例

```
# Only focus on the 1mg dose
Tooth.1mg <- subset(ToothGrowth, dose==1)
# From the boxplot orange juice leads to longer teeth
with(Tooth.1mg,boxplot(len~supp, names=c("Orange Juice","Ascorbic Acid"),
                       xlab="Supplment Method", ylab="Tooth Length"))
```



Tooth Length
14
16
18
20
22
24
26
Orange Juice
Ascorbic Acid
Suppliment Method

```
# Two-sample t-test
tt <- t.test(len~supp, data=Tooth.1mg, alternative="two.sided",
             var.equal=FALSE, conf.level=.95)
# Extract results
# See the documentation for a description of the values returned
names(tt)
## [1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
## [6] "null.value"  "alternative" "method"      "data.name"
tt$p.value
## [1] 0.001038376
tt$conf.int
## [1] 2.802148 9.057852
## attr(,"conf.level")
## [1] 0.95
```

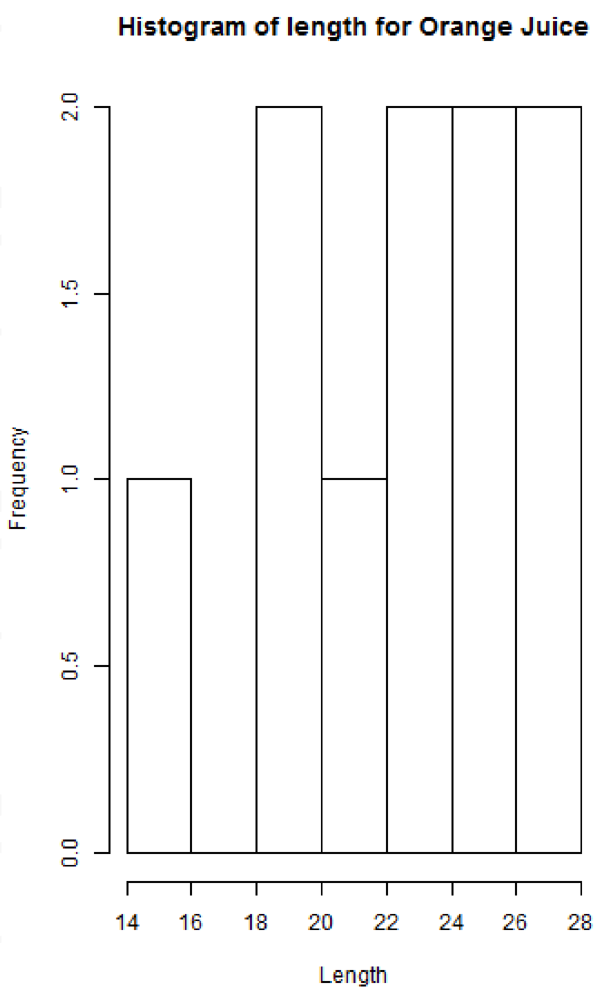
◆ 方差比较

对于来自正态总体的两个样本，可用 F-test 进行方差检验

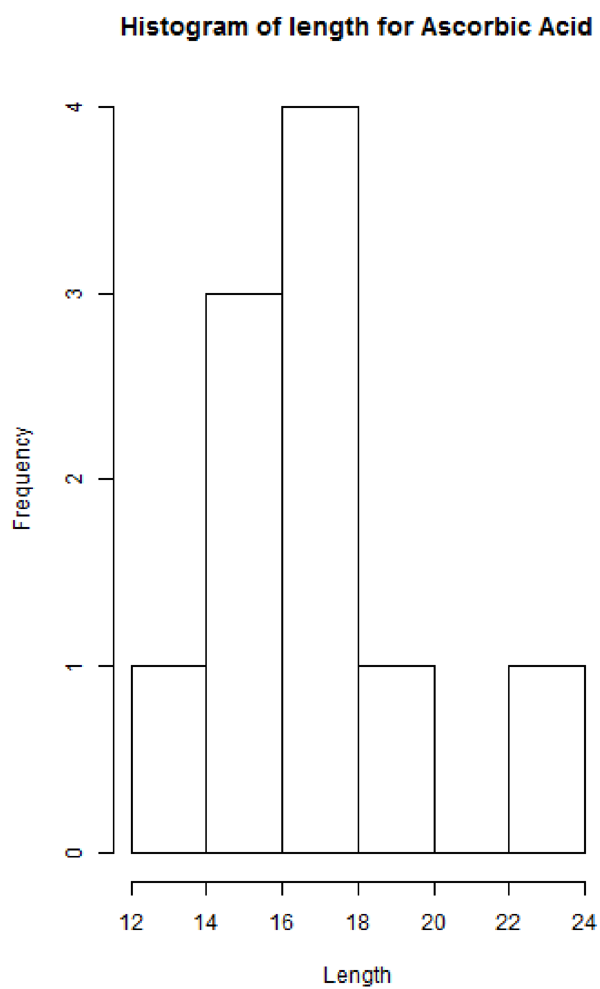
```
`var.test()`
```

❖ 示例

```r
# ToothGrowth dataset where dose=1mg
Tooth.1mg <- subset(ToothGrowth, dose==1)
# Sample Variances
with(Tooth.1mg, tapply(len, supp, var))
##        OJ        VC
## 15.295556  6.326778
# From histograms the normality assumption of the F-test is not reasonable
par(mfrow=c(1,2))
with(Tooth.1mg, hist(len[supp=="OJ"],
                     main="Histogram of length for Orange Juice", xlab="Length"))
with(Tooth.1mg, hist(len[supp=="VC"],
                     main="Histogram of length for Ascorbic Acid", xlab="Length"))
```



Histogram of length for Orange Juice
Frequency
0.0
0.5
1.0
1.5
2.0
14
16
18
20
22
24
26
28
Length



Histogram of length for Ascorbic Acid
Frequency
0
1
2
3
4
12
14
16
18
20
22
24
Length

```r
# Formula interface
var.test(len ~ supp, data=Tooth.1mg, alternative="two.sided")
##
##  F test to compare two variances
##
## data:  len by supp
## F = 2.4176, num df = 9, denom df = 9, p-value = 0.2046
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.6004952 9.7332038
## sample estimates:
## ratio of variances
##            2.41759
```

线性模型

◆ lm()

```
fit1 = lm(formula, data)
```

◆ 公式

❖ 公式构成

- 变量名　　公式所包含的变量名称，无需加引号
- 波浪号（~）　　表示响应变量与控制变量的关系。
- 加号（+）　　变量之间的线性关系。
- 零（0）　　表示去除截距项。
- -　　从等式中移除某个变量

-
    - 数据集中除因变量以外的所有变量。

❖ 公式构成

-

| 条件依赖。如 `y~year|sex`

-

Identity function 变量表达式。例如 `a+b` 表示变量 `a` 和 `b` 同时出现在公式中。`I(a+b)` 表示公式中包含的是一个新变量 `= a + b`。

-

: 自变量交互项

-

星号 (*)　自变量所有可能交互项。例如: `y ~ a * b` 等价于 `y ~ a + b + I(a * b)`, `y~(a+b)*w` 等价于 `y ~ a + b + w + I(a*w) + I(b*w)`。

-

^n　所有主效应及n阶交互项。例如 `y ~ (x + z + w)^2` 等价于y ~ x + z + w + x:z + x:w + z:w`

-

其它函数　例如:

```
y~log(u)+sin(v)+w
```

❖ 公式创建

```
sample.formula<-as.formula(y~x1+x2+x3)
class(sample.formula)
## [1] "formula"
typeof(sample.formula)
## [1] "language"

sample.formula2<-as.formula('y ~ x1 + x2 + x3')
class(sample.formula2)
## [1] "formula"
typeof(sample.formula2)
## [1] "language"
```

◆ 公式示例

Model	Interpretation
y ~ 1 y ~ a y ~ -1 + a y ~ a + b y ~ a+b+c+a:b y ~ a*b y ~ a + b + a:b y ~ factor(a) y ~ (a+b+c)^2 y ~ I(a^2) log(y) ~ y ~ .	仅有截距项Just the intercept 一个主效应去除截距项两个主效应三个主效应，一个 a 和 b 的交互效应 a和b的主效应及其交互效应

◆ 简单线性回归

❖ 一元线性回归

```
# 萼片长度 ~ 萼片宽度
out <- lm(Sepal.Length ~ Sepal.Width, iris)
summary(out)
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5561 -0.6333 -0.1120  0.5579  2.2226
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5262     0.4789   13.63   <2e-16 ***
## Sepal.Width  -0.2234     0.1551   -1.44    0.152
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8251 on 148 degrees of freedom
## Multiple R-squared:  0.01382,    Adjusted R-squared:  0.007159
## F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

❖ 提取回归模型参数

```
summary(out)$r.squared
## [1] 0.01382265
coeffs = coefficients(out); coeffs
## (Intercept) Sepal.Width
##   6.5262226  -0.2233611
```

```
# 方差协方差矩阵
vcov(out)
##             (Intercept) Sepal.Width
## (Intercept)  0.22934170 -0.07352916
## Sepal.Width -0.07352916  0.02405009
# 估计系数的置信区间
confint(out)
##                 2.5 %     97.5 %
## (Intercept)  5.579865 7.47258038
## Sepal.Width -0.529820 0.08309785
```

❖ 估计因变量

```
fitted(out)
##        1        2        3        4        5        6        7        8
## 5.744459 5.856139 5.811467 5.833803 5.722123 5.655114 5.766795 5.766795
##        9       10       11       12       13       14       15       16
## 5.878475 5.833803 5.699787 5.766795 5.856139 5.856139 5.632778 5.543434
##       17       18       19       20       21       22       23       24
## 5.655114 5.744459 5.677451 5.677451 5.766795 5.699787 5.722123 5.789131
##       25       26       27       28       29       30       31       32
```

```
## 5.766795 5.856139 5.766795 5.744459 5.766795 5.811467 5.833803
5.766795
##       33       34       35       36       37       38       39
40
## 5.610442 5.588106 5.833803 5.811467 5.744459 5.722123 5.856139
5.766795
##       41       42       43       44       45       46       47
48
## 5.744459 6.012492 5.811467 5.744459 5.677451 5.856139 5.677451
5.811467
##       49       50       51       52       53       54       55
56
## 5.699787 5.789131 5.811467 5.811467 5.833803 6.012492 5.900812
5.900812
##       57       58       59       60       61       62       63
64
## 5.789131 5.990156 5.878475 5.923148 6.079500 5.856139 6.034828
5.878475
```

```
##       65       66       67       68       69       70       71
72
## 5.878475 5.833803 5.856139 5.923148 6.034828 5.967820 5.811467
5.900812
##       73       74       75       76       77       78       79
80
## 5.967820 5.900812 5.878475 5.856139 5.900812 5.856139 5.878475
5.945484
##       81       82       83       84       85       86       87
88
## 5.990156 5.990156 5.923148 5.923148 5.856139 5.766795 5.833803
6.012492
##       89       90       91       92       93       94       95
96
## 5.856139 5.967820 5.945484 5.856139 5.945484 6.012492 5.923148
5.856139
##       97       98       99      100      101      102      103
104
```

```
## 5.878475 5.878475 5.967820 5.900812 5.789131 5.923148 5.856139 5.878475
##      105      106      107      108      109      110      111      112
## 5.856139 5.856139 5.967820 5.878475 5.967820 5.722123 5.811467 5.923148
##      113      114      115      116      117      118      119      120
## 5.856139 5.967820 5.900812 5.811467 5.856139 5.677451 5.945484 6.034828
##      121      122      123      124      125      126      127      128
## 5.811467 5.900812 5.900812 5.923148 5.789131 5.811467 5.900812 5.856139
##      129      130      131      132      133      134      135      136
## 5.900812 5.856139 5.900812 5.677451 5.900812 5.900812 5.945484 5.856139
```

```
##      137      138      139      140      141      142      143
144
## 5.766795 5.833803 5.856139 5.833803 5.833803 5.833803 5.923148
5.811467
##      145      146      147      148      149      150
## 5.789131 5.856139 5.967820 5.856139 5.766795 5.856139
###predict(out)
```

❖ 估计因变量

```
# 置信区间估计
predict(out, iris, interval="confidence")
##          fit      lwr      upr
## 1   5.744459 5.554388 5.934529
## 2   5.856139 5.721856 5.990423
## 3   5.811467 5.671342 5.951592
## 4   5.833803 5.700034 5.967573
## 5   5.722123 5.509095 5.935150
## 6   5.655114 5.364576 5.945653
## 7   5.766795 5.597233 5.936357
## 8   5.766795 5.597233 5.936357
## 9   5.878475 5.736884 6.020067
## 10  5.833803 5.700034 5.967573
## 11  5.699787 5.462062 5.937511
## 12  5.766795 5.597233 5.936357
## 13  5.856139 5.721856 5.990423
```

```
## 14   5.856139 5.721856 5.990423
## 15   5.632778 5.314690 5.950866
## 16   5.543434 5.110961 5.975907
## 17   5.655114 5.364576 5.945653
## 18   5.744459 5.554388 5.934529
## 19   5.677451 5.413777 5.941124
## 20   5.677451 5.413777 5.941124
## 21   5.766795 5.597233 5.936357
## 22   5.699787 5.462062 5.937511
## 23   5.722123 5.509095 5.935150
## 24   5.789131 5.636639 5.941623
## 25   5.766795 5.597233 5.936357
## 26   5.856139 5.721856 5.990423
## 27   5.766795 5.597233 5.936357
## 28   5.744459 5.554388 5.934529
## 29   5.766795 5.597233 5.936357
## 30   5.811467 5.671342 5.951592
## 31   5.833803 5.700034 5.967573
```

```
## 32  5.766795 5.597233 5.936357
## 33  5.610442 5.264284 5.956601
## 34  5.588106 5.213473 5.962739
## 35  5.833803 5.700034 5.967573
## 36  5.811467 5.671342 5.951592
## 37  5.744459 5.554388 5.934529
## 38  5.722123 5.509095 5.935150
## 39  5.856139 5.721856 5.990423
## 40  5.766795 5.597233 5.936357
## 41  5.744459 5.554388 5.934529
## 42  6.012492 5.744929 6.280055
## 43  5.811467 5.671342 5.951592
## 44  5.744459 5.554388 5.934529
## 45  5.677451 5.413777 5.941124
## 46  5.856139 5.721856 5.990423
## 47  5.677451 5.413777 5.941124
## 48  5.811467 5.671342 5.951592
## 49  5.699787 5.462062 5.937511
```

```
## 50  5.789131 5.636639 5.941623
## 51  5.811467 5.671342 5.951592
## 52  5.811467 5.671342 5.951592
## 53  5.833803 5.700034 5.967573
## 54  6.012492 5.744929 6.280055
## 55  5.900812 5.746078 6.055546
## 56  5.900812 5.746078 6.055546
## 57  5.789131 5.636639 5.941623
## 58  5.990156 5.748694 6.231618
## 59  5.878475 5.736884 6.020067
## 60  5.923148 5.750766 6.095529
## 61  6.079500 5.729189 6.429812
## 62  5.856139 5.721856 5.990423
## 63  6.034828 5.740287 6.329369
## 64  5.878475 5.736884 6.020067
## 65  5.878475 5.736884 6.020067
## 66  5.833803 5.700034 5.967573
## 67  5.856139 5.721856 5.990423
```

```
## 68  5.923148 5.750766 6.095529
## 69  6.034828 5.740287 6.329369
## 70  5.967820 5.751265 6.184375
## 71  5.811467 5.671342 5.951592
## 72  5.900812 5.746078 6.055546
## 73  5.967820 5.751265 6.184375
## 74  5.900812 5.746078 6.055546
## 75  5.878475 5.736884 6.020067
## 76  5.856139 5.721856 5.990423
## 77  5.900812 5.746078 6.055546
## 78  5.856139 5.721856 5.990423
## 79  5.878475 5.736884 6.020067
## 80  5.945484 5.752180 6.138788
## 81  5.990156 5.748694 6.231618
## 82  5.990156 5.748694 6.231618
## 83  5.923148 5.750766 6.095529
## 84  5.923148 5.750766 6.095529
## 85  5.856139 5.721856 5.990423
```

```
## 86  5.766795 5.597233 5.936357
## 87  5.833803 5.700034 5.967573
## 88  6.012492 5.744929 6.280055
## 89  5.856139 5.721856 5.990423
## 90  5.967820 5.751265 6.184375
## 91  5.945484 5.752180 6.138788
## 92  5.856139 5.721856 5.990423
## 93  5.945484 5.752180 6.138788
## 94  6.012492 5.744929 6.280055
## 95  5.923148 5.750766 6.095529
## 96  5.856139 5.721856 5.990423
## 97  5.878475 5.736884 6.020067
## 98  5.878475 5.736884 6.020067
## 99  5.967820 5.751265 6.184375
## 100 5.900812 5.746078 6.055546
## 101 5.789131 5.636639 5.941623
## 102 5.923148 5.750766 6.095529
## 103 5.856139 5.721856 5.990423
```

```
## 104 5.878475 5.736884 6.020067
## 105 5.856139 5.721856 5.990423
## 106 5.856139 5.721856 5.990423
## 107 5.967820 5.751265 6.184375
## 108 5.878475 5.736884 6.020067
## 109 5.967820 5.751265 6.184375
## 110 5.722123 5.509095 5.935150
## 111 5.811467 5.671342 5.951592
## 112 5.923148 5.750766 6.095529
## 113 5.856139 5.721856 5.990423
## 114 5.967820 5.751265 6.184375
## 115 5.900812 5.746078 6.055546
## 116 5.811467 5.671342 5.951592
## 117 5.856139 5.721856 5.990423
## 118 5.677451 5.413777 5.941124
## 119 5.945484 5.752180 6.138788
## 120 6.034828 5.740287 6.329369
## 121 5.811467 5.671342 5.951592
```

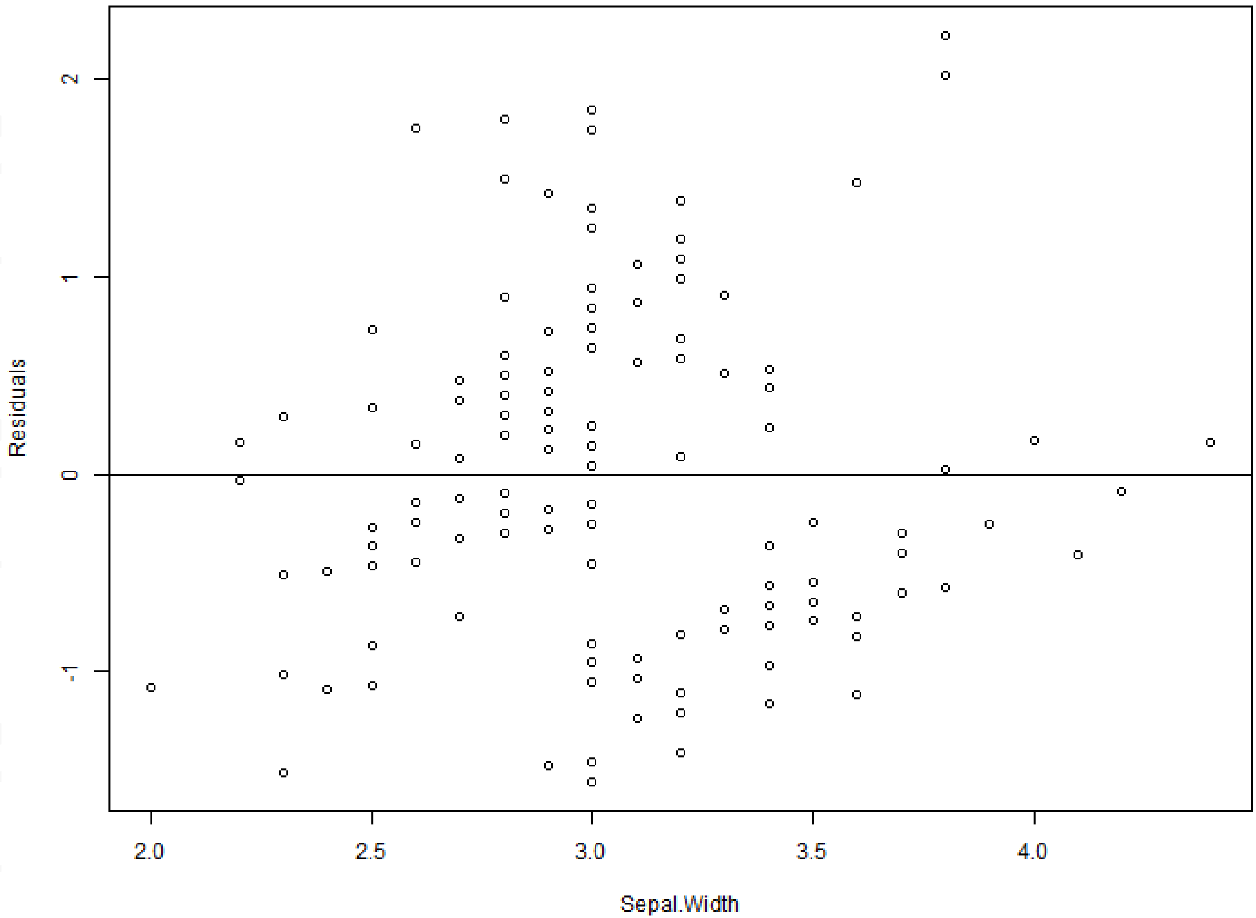
```
## 122 5.900812 5.746078 6.055546
## 123 5.900812 5.746078 6.055546
## 124 5.923148 5.750766 6.095529
## 125 5.789131 5.636639 5.941623
## 126 5.811467 5.671342 5.951592
## 127 5.900812 5.746078 6.055546
## 128 5.856139 5.721856 5.990423
## 129 5.900812 5.746078 6.055546
## 130 5.856139 5.721856 5.990423
## 131 5.900812 5.746078 6.055546
## 132 5.677451 5.413777 5.941124
## 133 5.900812 5.746078 6.055546
## 134 5.900812 5.746078 6.055546
## 135 5.945484 5.752180 6.138788
## 136 5.856139 5.721856 5.990423
## 137 5.766795 5.597233 5.936357
## 138 5.833803 5.700034 5.967573
## 139 5.856139 5.721856 5.990423
```

```
## 140 5.833803 5.700034 5.967573
## 141 5.833803 5.700034 5.967573
## 142 5.833803 5.700034 5.967573
## 143 5.923148 5.750766 6.095529
## 144 5.811467 5.671342 5.951592
## 145 5.789131 5.636639 5.941623
## 146 5.856139 5.721856 5.990423
## 147 5.967820 5.751265 6.184375
## 148 5.856139 5.721856 5.990423
## 149 5.766795 5.597233 5.936357
## 150 5.856139 5.721856 5.990423
```

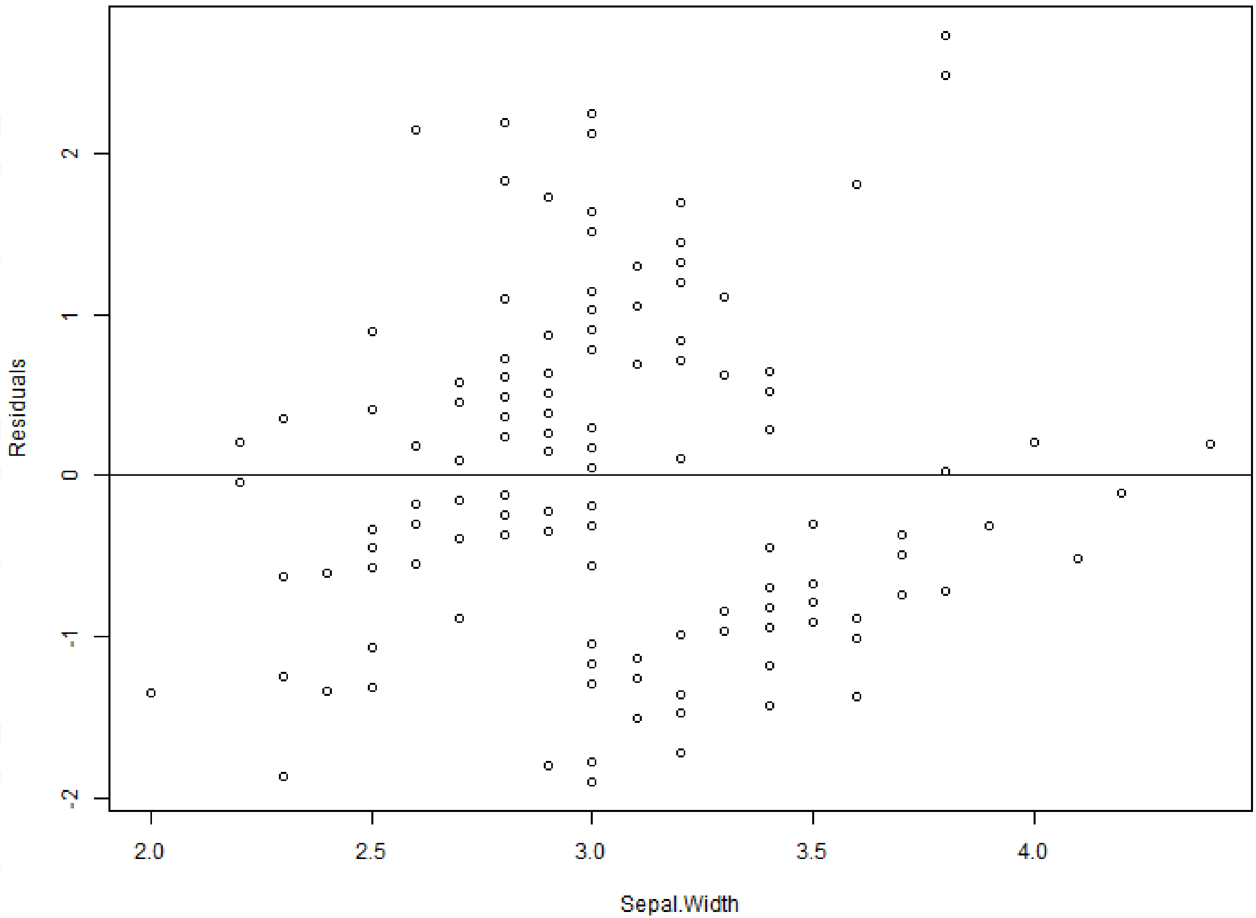
❖ 残差

```r
lm.res = resid(out)
lm.res2 = iris$Sepal.Length - fitted(out)
all.equal(lm.res, lm.res2, check.attributes = F)
## [1] TRUE

plot(iris$Sepal.Width, lm.res,
  ylab="Residuals", xlab="Sepal.Width",
  main="")
abline(0, 0)   # the horizon
```


Residuals
2
1
0
-1
2.0
2.5
3.0
3.5
4.0
Sepal.Width

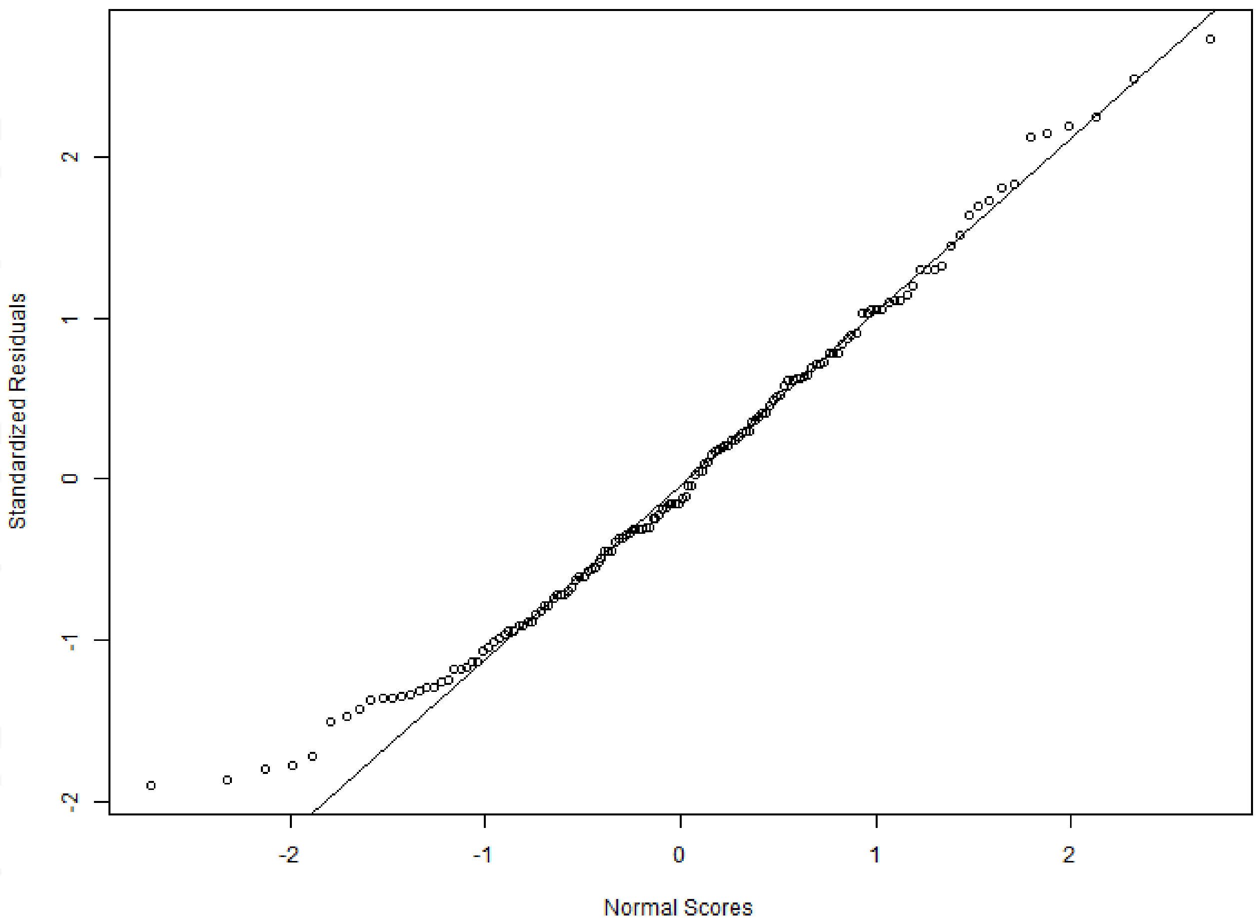
```
# 标准化残差 $res.std = res/ std(res)$
stdres<-rstandard(out)
plot(iris$Sepal.Width, stdres,
  ylab="Residuals", xlab="Sepal.Width",
  main="")
abline(0, 0)
```



Residuals
2
1
0
-1
-2
2.0
2.5
3.0
3.5
4.0
Sepal.Width

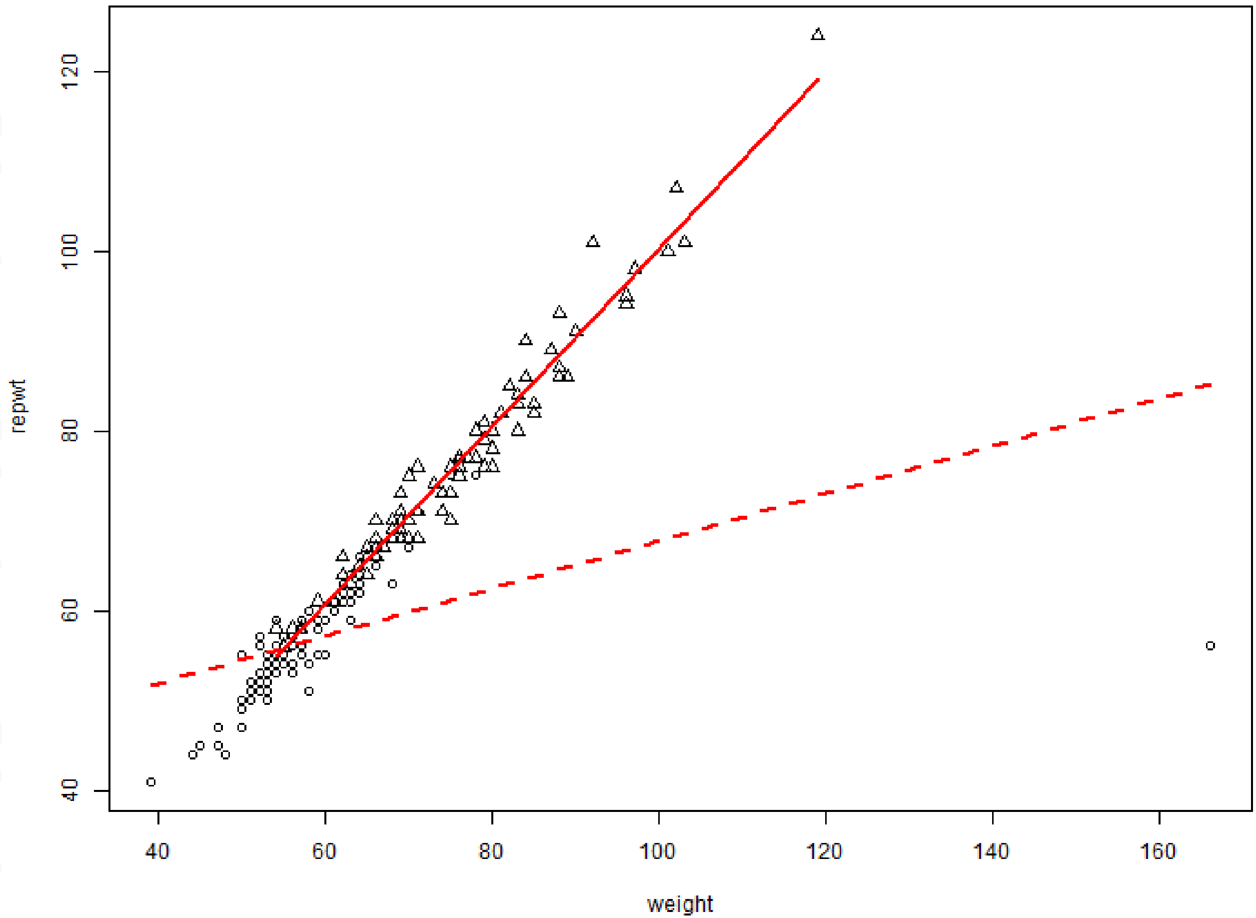
❖ 残差正态概率图

```
qqnorm(stdres,
 ylab="Standardized Residuals",
 xlab="Normal Scores",
 main="")
qqline(stdres)
```


Standardized Residuals
-2
-1
0
1
2
Normal Scores

❖ 使用 car 包制图

```r
library(car)
plot(repwt ~ weight, pch=c(1,2)[sex], data=Davis)
regLine(lm(repwt~weight, subset=sex=="M", data=Davis))
regLine(lm(repwt~weight, subset=sex=="F", data=Davis), lty=2)
```



repwt
weight
40
60
80
100
120
140
160

❖ 异常值

```r
plot(iris$Sepal.Width, iris$Sepal.Length)
identify(iris$Sepal.Width, iris$Sepal.Length)
```

进入交互模式，用鼠标左键逐个点击散点图上的异常值，完成后按 Esc 或鼠标右键退出。R 自动输出被标识为异常值的序号。

❖ 标准化回归系数

- 方式一：对数据进行标准化处理，将数据转化为均值为0，标准差为1，然后再用 lm() 函数拟合。

```
iris_std = as.data.frame(scale(iris[, c('Sepal.Length', 'Sepal.Width')]))
out_std <- lm(Sepal.Length ~ Sepal.Width, iris_std)
summary(out_std)
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width, data = iris_std)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8793 -0.7648 -0.1352  0.6738  2.6840
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -3.759e-16  8.136e-02    0.00    1.000
## Sepal.Width -1.176e-01  8.163e-02   -1.44    0.152
##
## Residual standard error: 0.9964 on 148 degrees of freedom
## Multiple R-squared:  0.01382,    Adjusted R-squared:  0.007159
## F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

❖ 标准化回归系数

- 方式二：在设定回归方程的时候，去掉截距项。

由于少用了一个自由度，除了自变量的回归系数以外，模型的残标准误、回归系数的标准误、t值都会发生变化。

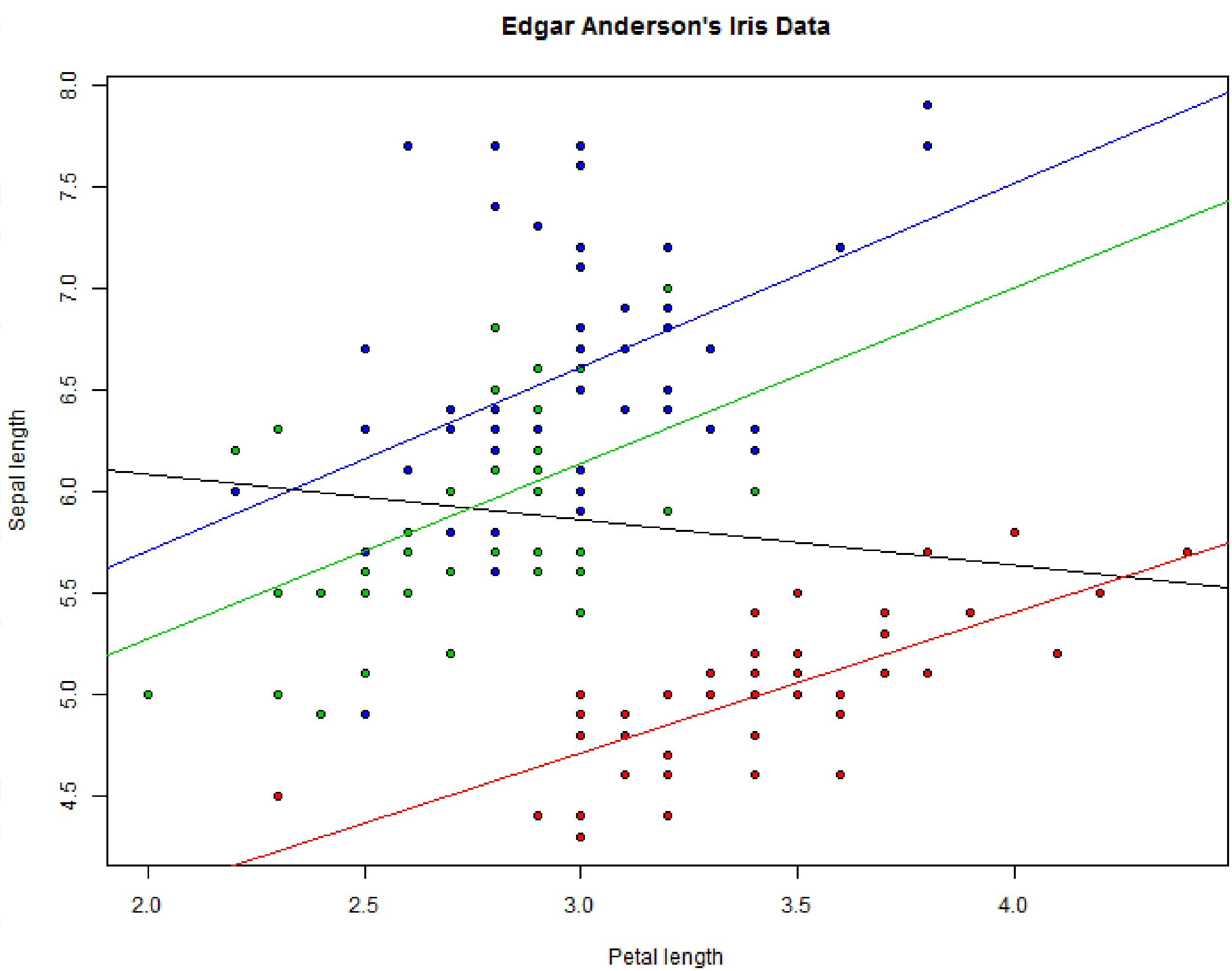
```
lm(y ~ x - 1)
```

```
out3 <- lm(Sepal.Length ~ Sepal.Width -1, iris)
summary(out3)
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width - 1, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5236 -1.0362  0.4823  0.9897  2.8406
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## Sepal.Width  1.86901    0.03265   57.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.235 on 149 degrees of freedom
## Multiple R-squared:  0.9565, Adjusted R-squared:  0.9562
## F-statistic:  3277 on 1 and 149 DF,  p-value: < 2.2e-16
```

以 iris 数据为例，做4个回归模型：第一个模型利用全部数据，另外三个分别用筛选后的数据。

```
plot(iris$Sepal.Width, iris$Sepal.Length, pch=21, bg=c("red","green3","bl
main="Edgar Anderson's Iris Data", xlab="Petal length", ylab="Sepal
length")
abline(lm(Sepal.Length ~ Sepal.Width, data=iris)$coefficients,
col="black")
abline(lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="seto
col="red")
abline(lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="vers
col="green3")
abline(lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="virg
col="blue")
```



Edgar Anderson's Iris Data
Sepal length
Petal length
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
2.0
2.5
3.0
3.5
4.0

◆ 多元线性回归

❖ 示例

```
fit <- lm(len~factor(dose)+supp, data=ToothGrowth)
summary(fit)
##
## Call:
## lm(formula = len ~ factor(dose) + supp, data = ToothGrowth)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.085 -2.751 -0.800  2.446  9.650
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    12.4550     0.9883  12.603  < 2e-16 ***
## factor(dose)1   9.1300     1.2104   7.543 4.38e-10 ***
## factor(dose)2  15.4950     1.2104  12.802  < 2e-16 ***
## suppVC         -3.7000     0.9883  -3.744 0.000429 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.828 on 56 degrees of freedom
## Multiple R-squared:  0.7623, Adjusted R-squared:  0.7496
## F-statistic: 59.88 on 3 and 56 DF,  p-value: < 2.2e-16
```

```
anova(fit)
## Analysis of Variance Table
##
## Response: len
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## factor(dose)  2 2426.43 1213.22  82.811 < 2.2e-16 ***
## supp          1  205.35  205.35  14.017 0.0004293 ***
## Residuals    56  820.43   14.65
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(fit)$r.squared
## [1] 0.7623478
# $r.adj = 1- (1-R^2) (n-1)/(n-p-1)$
summary(fit)$adj.r.squared
## [1] 0.7496165
```

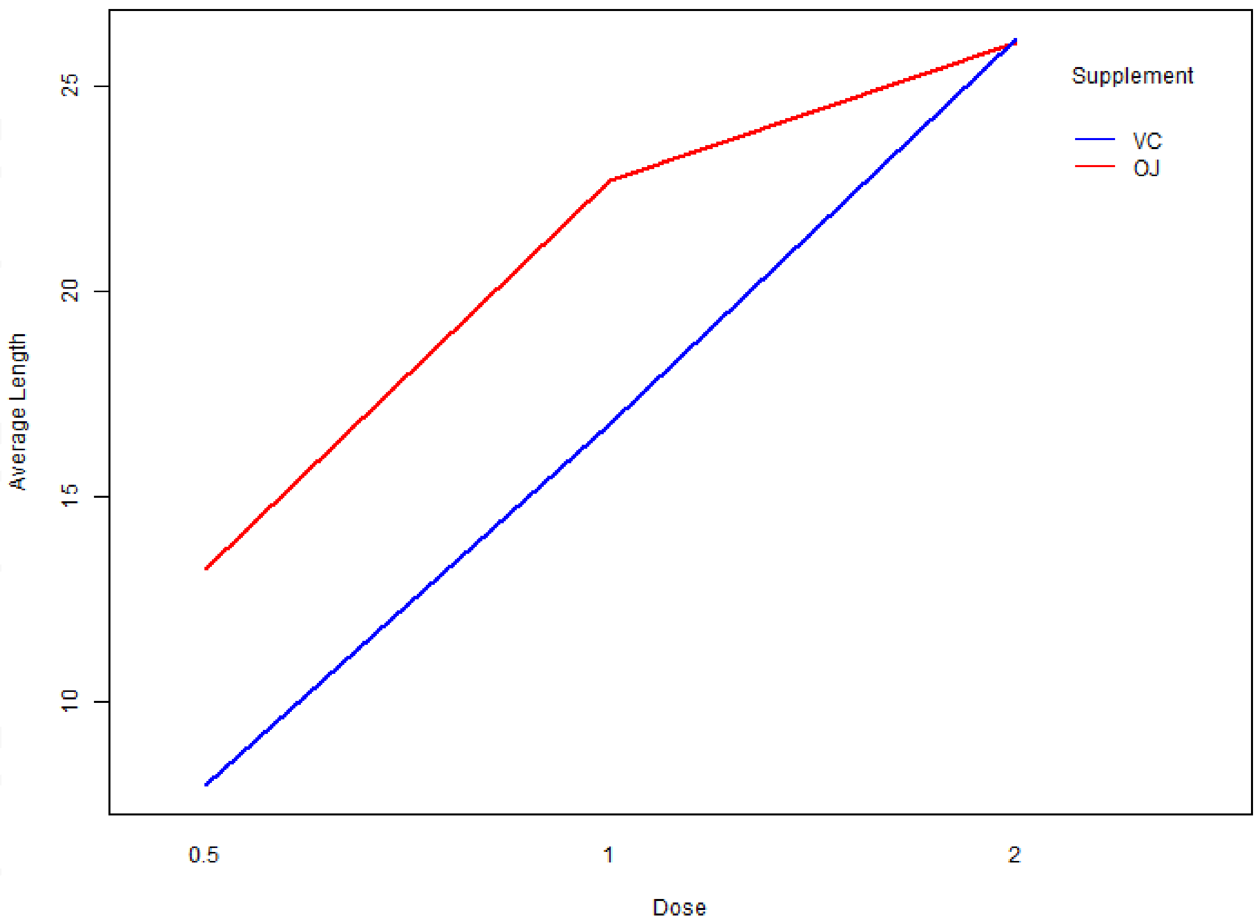
```
ssq = anova(fit)$`Sum Sq`
sum(head(ssq,2))/sum(ssq)
## [1] 0.7623478
```

```
dose.2 <- relevel(factor(ToothGrowth$dose), 3)
fit <- lm(len~dose.2+supp, data=ToothGrowth)
summary(fit)
##
## Call:
## lm(formula = len ~ dose.2 + supp, data = ToothGrowth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -7.085 -2.751 -0.800  2.446  9.650
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.9500     0.9883  28.281  < 2e-16 ***
## dose.20.5   -15.4950     1.2104 -12.802  < 2e-16 ***
## dose.21      -6.3650     1.2104  -5.259 2.35e-06 ***
## suppVC       -3.7000     0.9883  -3.744 0.000429 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.828 on 56 degrees of freedom
## Multiple R-squared:  0.7623, Adjusted R-squared:  0.7496
## F-statistic: 59.88 on 3 and 56 DF,  p-value: < 2.2e-16
```

❖ 画图

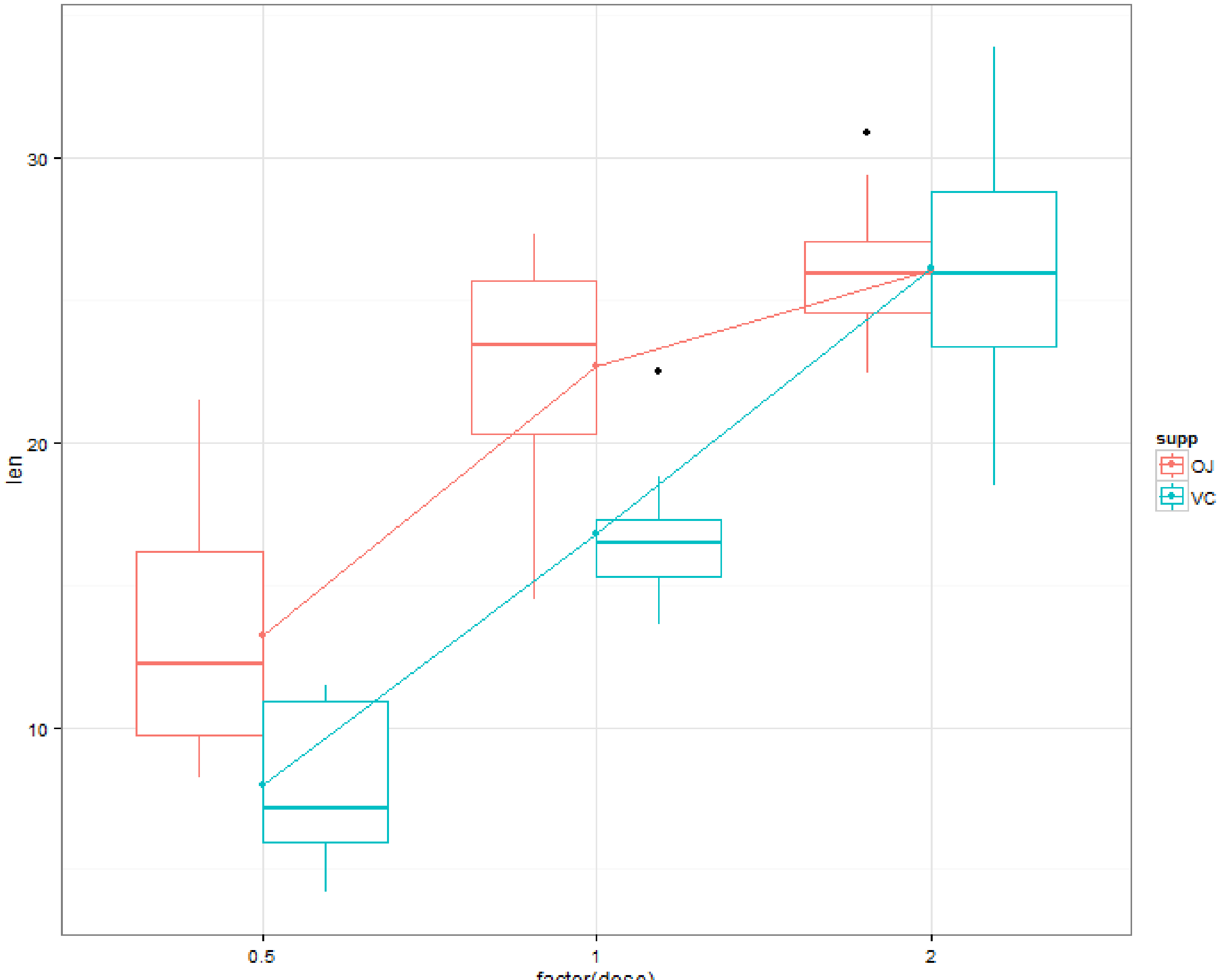
```
with(ToothGrowth,
     interaction.plot(x.factor=dose, trace.factor=supp, response=len,
                      fun=mean,
                      xlab="Dose", ylab="Average Length", trace.label="Su
                      lty=1, lwd=2, col=c("red", "blue")))
```


Supplement
VC
OJ
Average Length
25
20
15
10
0.5
1
2
Dose

❖ **ggplot** 实现

```r
toothInt = aggregate(len ~ dose + supp, ToothGrowth, mean)

library(ggplot2)
ggplot(ToothGrowth, aes(x = factor(dose), y = len, colour = supp)) +
  geom_boxplot() +
  geom_point(data = toothInt, aes(y = len)) +
  geom_line(data = toothInt, aes(y = len, group = supp)) +
  theme_bw()
```



30
20
10
len
0.5
1
2
factor(dose)
supp
OJ
VC

❖ 嵌套模型检验

什么是嵌套模型?

```
fit.1 = lm(len ~ factor(dose), ToothGrowth)
fit.2 = lm(len ~ factor(dose) + supp, ToothGrowth)
anova(fit.2, fit.1)
## Analysis of Variance Table
##
## Model 1: len ~ factor(dose) + supp
## Model 2: len ~ factor(dose)
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     56  820.43
## 2     57 1025.78 -1   -205.35 14.017 0.0004293 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
drop1(fit.2, ~supp, test="F")
## Single term deletions
##
## Model:
## len ~ factor(dose) + supp
##        Df Sum of Sq     RSS    AIC F value    Pr(>F)
## <none>                820.43 164.93
## supp    1    205.35 1025.78 176.33  14.017 0.0004293 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
library(lmtest)
lrtest(m1, m2)
```

❖ 变量选择

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

```
summary(lm1 <- lm(Fertility ~ ., data = swiss))
##
## Call:
## lm(formula = Fertility ~ ., data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2743  -5.2617   0.5032   4.1198  15.3213
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
## Agriculture      -0.17211    0.07030  -2.448  0.01873 *
```

```
## Examination      -0.25801    0.25388  -1.016  0.31546
## Education        -0.87094    0.18303  -4.758 2.43e-05 ***
## Catholic          0.10412    0.03526   2.953  0.00519 **
## Infant.Mortality  1.07705    0.38172   2.822  0.00734 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.165 on 41 degrees of freedom
## Multiple R-squared:  0.7067, Adjusted R-squared:  0.671
## F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10
slm1 <- step(lm1)
## Start:  AIC=190.69
## Fertility ~ Agriculture + Examination + Education + Catholic +
##     Infant.Mortality
##
##                    Df Sum of Sq    RSS    AIC
## - Examination       1     53.03 2158.1 189.86
## <none>                          2105.0 190.69
```

```
## - Agriculture        1    307.72 2412.8 195.10
## - Infant.Mortality  1    408.75 2513.8 197.03
## - Catholic           1    447.71 2552.8 197.75
## - Education          1   1162.56 3267.6 209.36
##
## Step:  AIC=189.86
## Fertility ~ Agriculture + Education + Catholic + Infant.Mortality
##
##                    Df Sum of Sq    RSS    AIC
## <none>                          2158.1 189.86
## - Agriculture        1    264.18 2422.2 193.29
## - Infant.Mortality  1    409.81 2567.9 196.03
## - Catholic           1    956.57 3114.6 205.10
## - Education          1   2249.97 4408.0 221.43
summary(slm1)
##
## Call:
## lm(formula = Fertility ~ Agriculture + Education + Catholic +
```

```
##     Infant.Mortality, data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6765  -6.0522   0.7514   3.1664  16.1422
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      62.10131    9.60489   6.466 8.49e-08 ***
## Agriculture      -0.15462    0.06819  -2.267  0.02857 *
## Education        -0.98026    0.14814  -6.617 5.14e-08 ***
## Catholic          0.12467    0.02889   4.315 9.50e-05 ***
## Infant.Mortality  1.07844    0.38187   2.824  0.00722 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.168 on 42 degrees of freedom
## Multiple R-squared:  0.6993, Adjusted R-squared:  0.6707
```

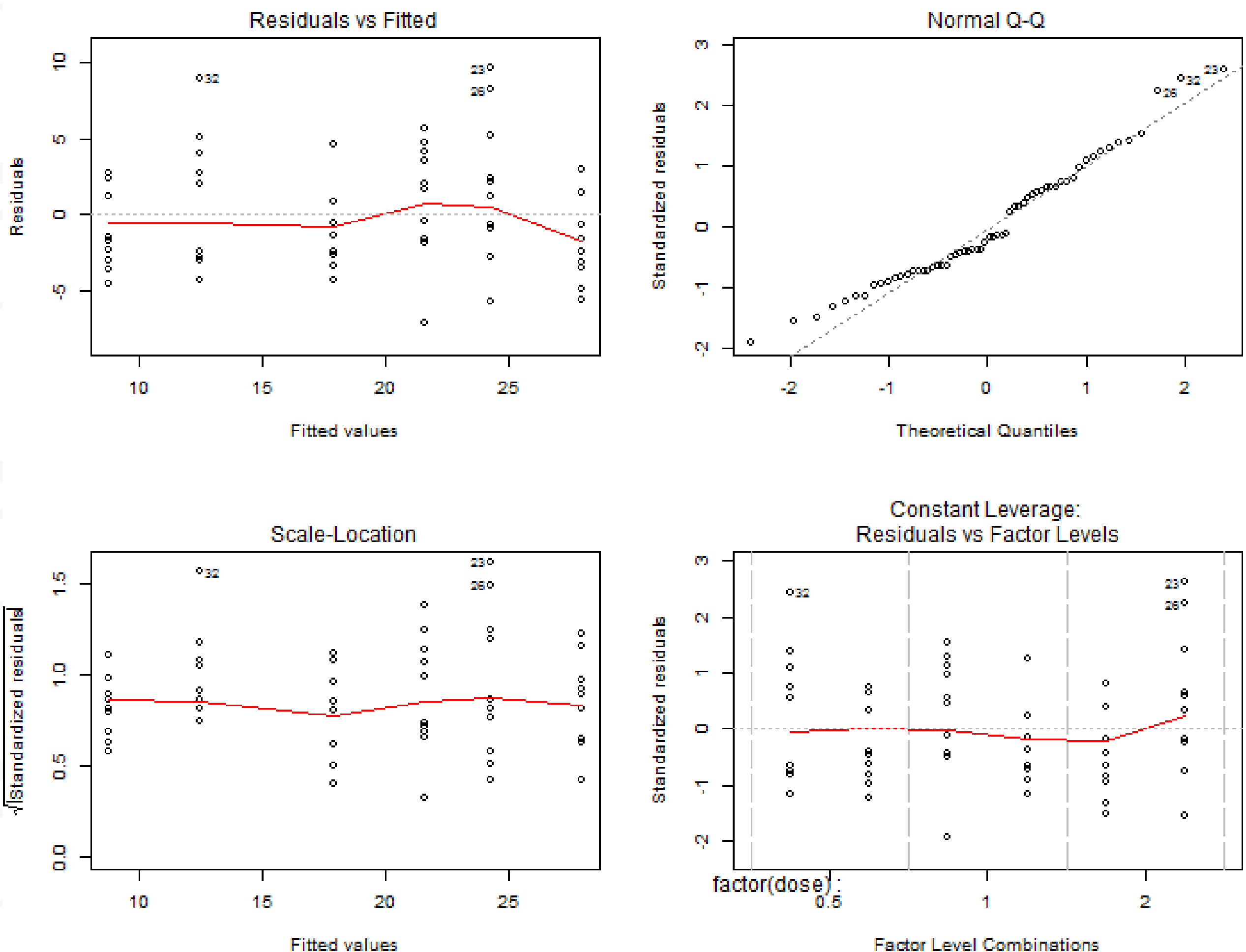
```
## F-statistic: 24.42 on 4 and 42 DF,  p-value: 1.717e-10
slm1$anova
##             Step Df Deviance Resid. Df Resid. Dev      AIC
## 1                NA       NA        41   2105.043 190.6913
## 2 - Examination  1 53.02656        42   2158.069 189.8606
```

❖ 模型诊断

诊断函数	说明
fitted.values()	
fitted( )	
residuals( )	
rstandard( )	
rstudent( )	
qqnorm() qqline()	
plot.lm()	

❖ 模型诊断示例

```r
fit <- lm(len~factor(dose)+supp, data=ToothGrowth)
par(mfrow=c(2,2))
plot(fit)
```



Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
factor(dose) :
Factor Level Combinations

❖ 影响点判断

```
# dfbetas, dffits, covratio and cooks.distance
inf.temp <- influence.measures(fit)
inf.pts <- which(apply(inf.temp$is.inf, 1, any))
ToothGrowth[inf.pts,]
##     len supp dose
## 23 33.9   VC  2.0
## 32 21.5   OJ  0.5
```

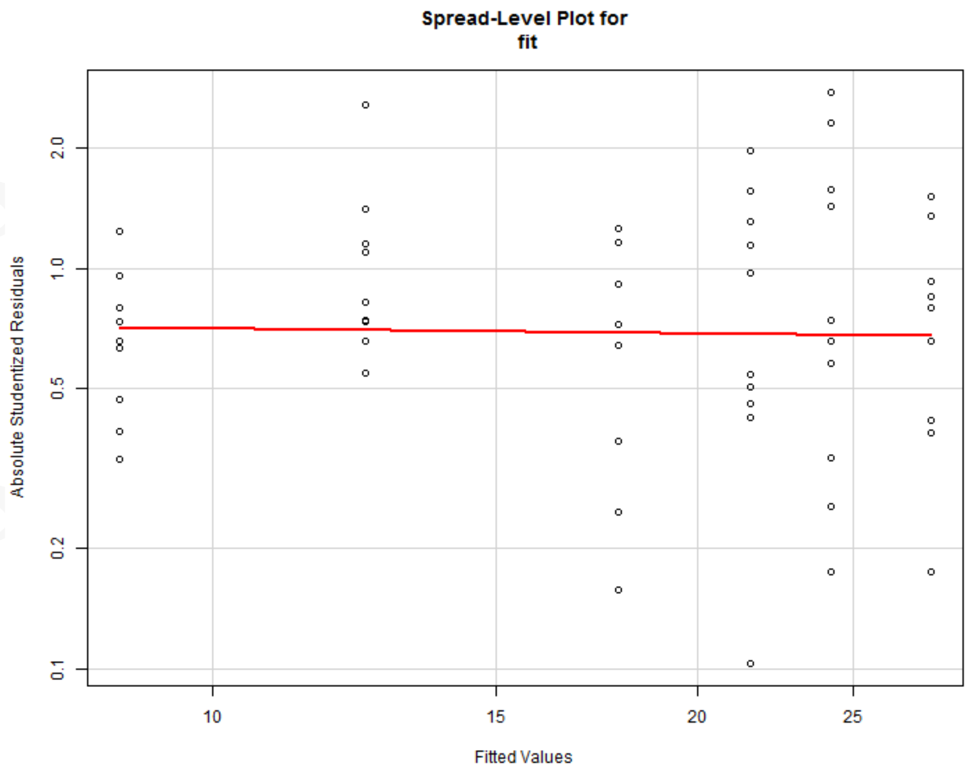
```r
lm.inf.coef <- lm.influence(fit)$coefficients
lm.inf.pts <- apply(lm.inf.coef, 2, FUN=function(x) which.max(abs(x)))
# Get the five points that cause the greatest change in the estimates
lm.inf.pts.top5 <- apply(lm.inf.coef, 2, FUN=function(x)
  names(rev(sort(abs(x)))[1:5]))
lm.inf.pts.top5
##      (Intercept) factor(dose)1 factor(dose)2 suppVC
## [1,] "32"        "32"          "23"          "23"
## [2,] "33"        "49"          "32"          "32"
## [3,] "37"        "50"          "26"          "26"
## [4,] "39"        "33"          "22"          "49"
## [5,] "38"        "44"          "53"          "22"
```

❖ 方差同质性检验

- car:
  - ncv.test(): 生成一个计分检验，零假设为误差方差守恒，备择假设则为误差方差随拟合值变化而变化。
  - spreadLevelPlot(): 包含最佳拟合曲线的散点图。

```
library(car)
ncvTest(fit)
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.4405518    Df = 1     p = 0.5068562
```

```
1 spreadLevelPlot(fit)
```



```
##
## Suggested power transformation:  1.037044
```

◆ 广义线性模型

- Generalized Linear Models are fit using the function `glm()`.

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

- `family` 参数设定分布类型和链接函数

```
binomial(link = "logit")
##
## Family: binomial
## Link function: logit
gaussian(link = "identity")
##
## Family: gaussian
## Link function: identity
poisson(link = "log")
##
## Family: poisson
## Link function: log
```

`summary.glm()`	Summarize the model fit
`anova.glm()`	Analysis of deviance table
`confint.glm()`	Confidence interval for model parameters
`predict.glm()`	Obtain predicted values
`influence.measures()`	Measures of influence
`step()`	Step-wise selection using AIC
`drop1()`	Test parameter using deviance

❖ 示例

```
y <- ifelse(ToothGrowth[,1]>20, 1, 0)
# Fit logistic model
fit <- glm(y~supp+factor(dose), family="binomial", data=ToothGrowth)
summary(fit)
##
## Call:
## glm(formula = y ~ supp + factor(dose)    , family = "binomial",
data = ToothGrowth)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.16659  -0.43759  -0.09337   0.44842   2.18796
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.247      1.051  -2.138  0.03253 *
```

```
## suppVC          -3.187      1.218  -2.617  0.00887 **
## factor(dose)1    3.136      1.237   2.534  0.01128 *
## factor(dose)2    7.680      1.848   4.156 3.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 82.911  on 59  degrees of freedom
## Residual deviance: 31.913  on 56  degrees of freedom
## AIC: 39.913
##
## Number of Fisher Scoring iterations: 6
```

```
anova(fit, test="Chisq") # Compare reduction in deviance, sequentially
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: y
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           59     82.911
## supp          1    4.339       58     78.572   0.03724 *
## factor(dose)  2   46.658       56     31.913 7.384e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
exp(coef(fit)) # Exponentiate coefficients
##   (Intercept)        suppVC factor(dose)1 factor(dose)2
##  1.057677e-01  4.129916e-02  2.300199e+01  2.164477e+03
exp(confint(fit)) # 95% CI for Exponentiated coefficients
##                      2.5 %       97.5 %
## (Intercept)   5.751916e-03 5.559953e-01
## suppVC        1.843523e-03 3.157606e-01
## factor(dose)1 2.832232e+00 5.260241e+02
## factor(dose)2 1.010501e+02 1.701531e+05
drop1(fit, ~.     , test="Chisq") # Compare reduction in deviance, marginally
## Single term deletions
##
## Model:
## y ~ supp + factor(dose)
##               Df Deviance    AIC    LRT  Pr(>Chi)
## <none>            31.913 39.913
## supp           1   42.802 48.802 10.888 0.0009678 ***
```

```
## factor(dose)  2   78.572 82.572 46.658 7.384e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
confint(fit) # Confidence interval for the parameters
##                     2.5 %     97.5 %
## (Intercept)   -5.158222 -0.5869954
## suppVC        -6.296077 -1.1527709
## factor(dose)1  1.041065  6.2653470
## factor(dose)2  4.615616 12.0444538
```

◆ 回归表格输出

```r
fit.1 = lm(len ~ factor(dose), ToothGrowth)
fit.2 = lm(len ~ factor(dose) + supp, ToothGrowth)

library(stargazer)
stargazer(fit.1, fit.2, fit, title="Regression Results",
          type = 'html', style = 'asr')
```

Regression Results len y OLS logistic (1) (2) (3) factor(dose)1 9.130*** 9.130*** 3.136* factor(dose)2 15.495*** 15.495*** 7.680*** suppVC -3.700*** -3.187** Constant 10.605*** 12.455*** -2.247* N 60 60 60 R2 0.703 0.762 Adjusted R2 0.692 0.750 Log Likelihood -15.957 Residual Std. Error 4.242 (df = 57) 3.828 (df = 56) F Statistic 67.416*** (df = 2; 57) 59.880*** (df = 3; 56) AIC 39.913 *p < .05;* ***p < .01;*** p < .001

◆ stargazer

Table 7: Mining Sector Expansion and Local Energy Prices

	(1) Average All Use Gas	(2) Industrial Gas Use	(3) Electricity All Use
Instrumental Variables:			
Mining Sector Share	-2.409*	-6.067**	-0.188*
	(1.262)	(2.453)	(0.113)
Reduced Form:			
Shale x Post 2008	-0.022**	-0.055***	-0.002*
	(0.011)	(0.019)	(0.001)
Ordinary Least Squares:			
Mining Sector Share	-0.059	-0.033	-0.003
	(0.069)	(0.149)	(0.005)
Clusters	337	337	364
Observations	24187	24620	33849
Instrument	15.09	14.99	15.79
R-squared	.953	.898	.885

Notes: All regressions include state-time fixed effects and county fixed effects. Column (1) uses the log of average gas prices in a county, where the consumer, commercial and industrial gas prices are weighted by their national consumption shares. In column (2) I only study the price charged to industrial consumers. Column (3) is the level of average electricity prices, where consumer, commercial and industrial prices are weighted by their respective national consumption shares. Robust standard errors clustered at the workforce investment board area are given in the parentheses with stars indicating *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.