R 语言入门

李昕垚

2016 年 8 月 12 日

R 的前世今生

R 的历史

1. R 是什么

- 一门语言
- 一个开源环境
- 用于统计计算和统计制图

2. R 的历史

- 源于贝尔实验室的 S 语言
- 诞生于奥克兰大学
- 开发者是 Ross Ihaka 和 Robert Gentleman，所以命名为 “R”

R 的现状

- 维护：R cores 修修补补；RStudio 维护轮子工程 (RMySQL、knitr、dplyr、tidyr、stringr、readr、rmarkdown、devtools、shiny、ggplot2)；全世界志愿者贡献自己的包
- 使用者：
  1. 学术界
  2. 医药界
  3. Google、facebook 等科技公司
- 统计之都极大的推动了 R 语言在中国的发展
- 上了大数据的车，和 spark、hadoop 紧密结合

R 的优势与短板

1. 优势

- 统计学家开发，能快速搭建模型原型
- 扩展性强，大多数开源框架都支持 R
- 开发者众多，可以得到最新颖的算法
- 计算速度快，略逊色于 C，可媲美 MATLAB
- 火热程度超过 MATLAB，2016 年 8 月 TIOBE 编程语言排行榜位列 17。MATLAB、SAS 为 18、21
- 社区强大、文档很多

2. 短板

- 不适合大规模高性能计算

使用 R

安装 R

R 官网https://cran.r-project.org/

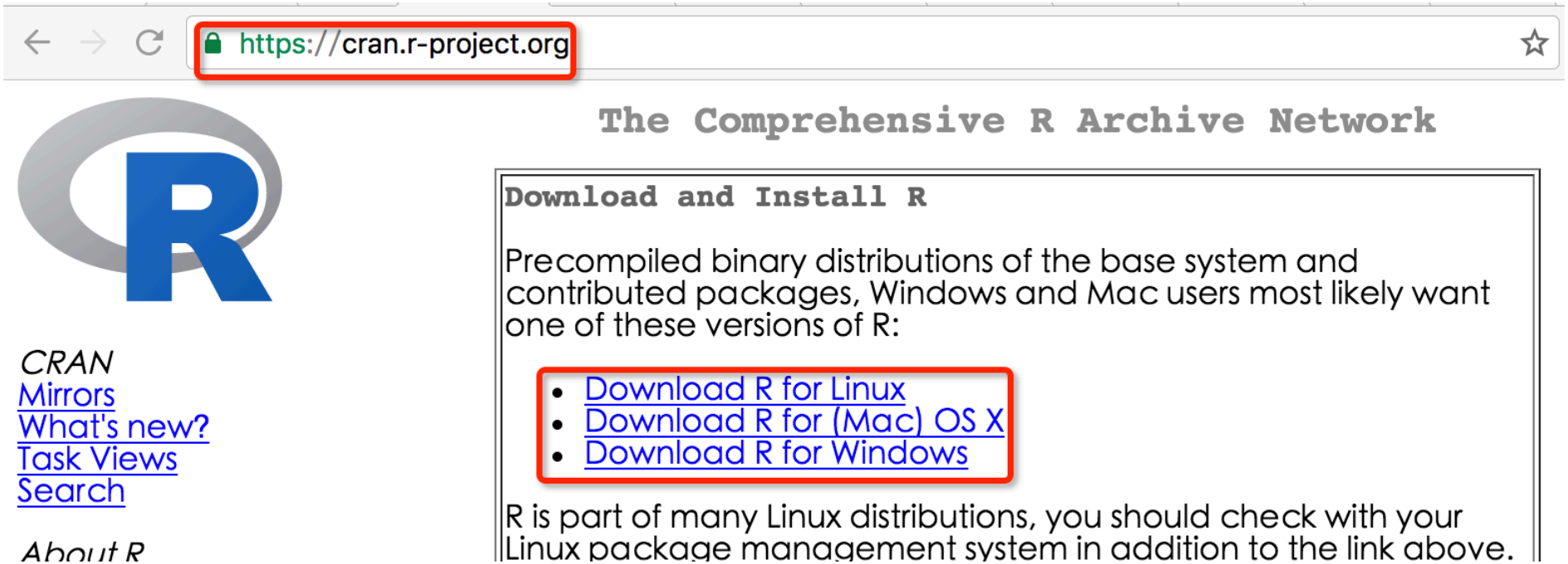
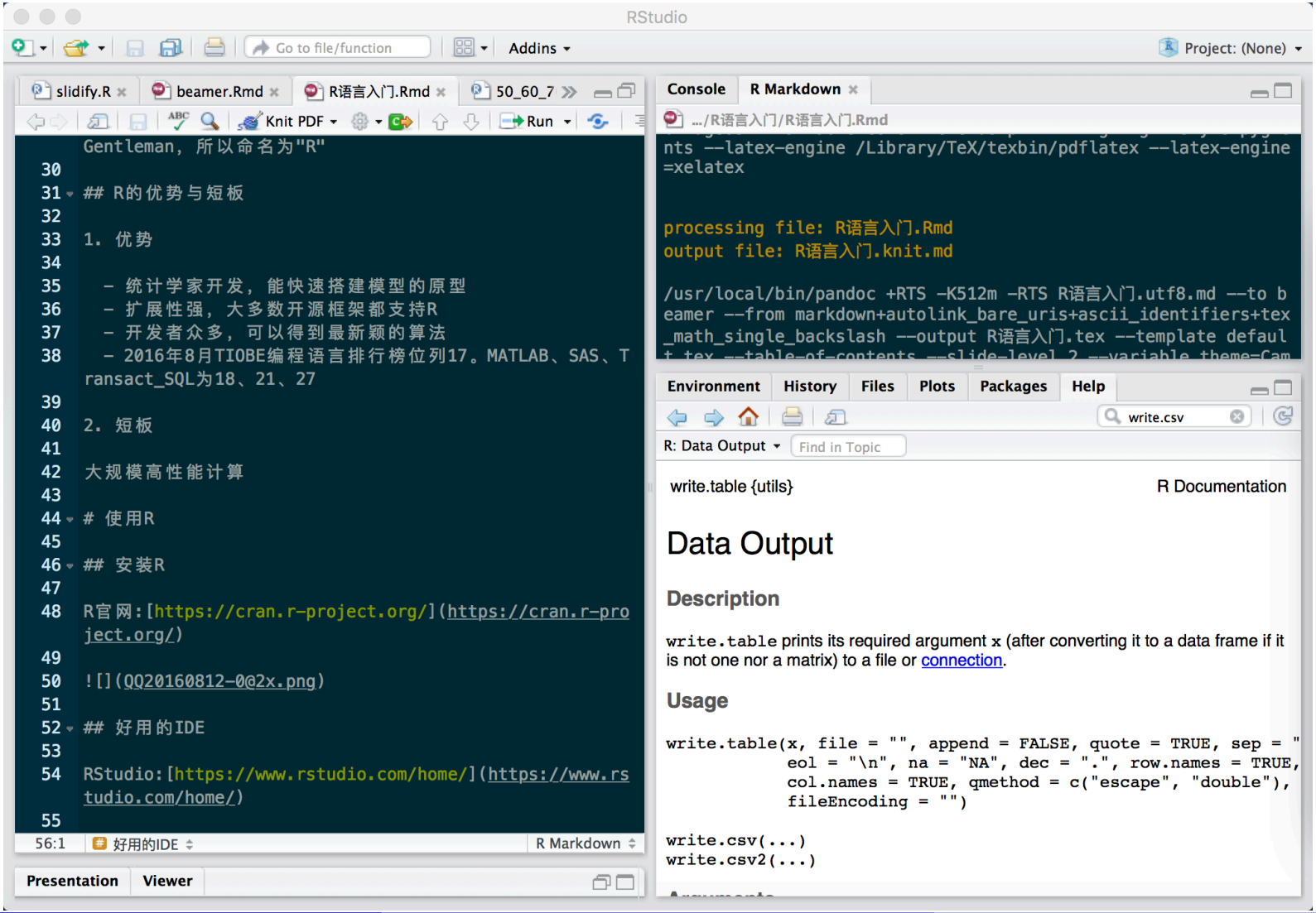


图 1:

好用的 IDE

RStudio 官网https://www.rstudio.com/home/



入门方式

- 《R 语言实战》
- 统计之都论坛
- 雪晴数据网
- R 语言中文论坛群
- 谢老大博客`yihui.name`
- github 上与作者交流
- 学习路线

1. 基本数据结构 (dataframe)
2. 基本数据操作 (筛选、删除、运算、类型转换)
3. 基本文件操作 (读入数据、输出数据)
4. 向量化批处理 (apply)
5. 基本绘图操作 (graphics、lattice、ggplot2、grid)
6. 统计学习和机器学习 (分类、回归)
7. 注重代码质量和效率问题
8. 维护自己写的 package

R 语言概览

一切皆对象

向量 vector

```
# c 赋值
x = c(1,2,3)
x
```

```
## [1] 1 2 3
```

```
y = rep(1,3)
y
```

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

```
z = seq(1,20,2)
z
```

```
##  [1]  1  3  5  7  9 11 13 15 17 19
```

一切皆对象

列表 list

```
l = list(seq(1,5,1),seq(1,4,2))
l
```

```
## [[1]]
## [1] 1 2 3 4 5
##
## [[2]]
## [1] 1 3
```

```
l[[1]]
```

```
## [1] 1 2 3 4 5
```

一切皆对象

矩阵 matrix

```
m = matrix(data = 1:12,nrow = 4,ncol = 3,
           dimnames = list(c("r1","r2","r3","r4"),
                           c("c1","c2","c3")))
m
```

```
##    c1 c2 c3
## r1  1  5  9
## r2  2  6 10
## r3  3  7 11
## r4  4  8 12
```

一切皆对象

数组 array

```
a = array(data = 1:12,dim = c(2,3,2))
a
```

```
## , , 1
##
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
##
## , , 2
##
##      [,1] [,2] [,3]
## [1,]    7    9   11
## [2,]    8   10   12
```

一切皆对象

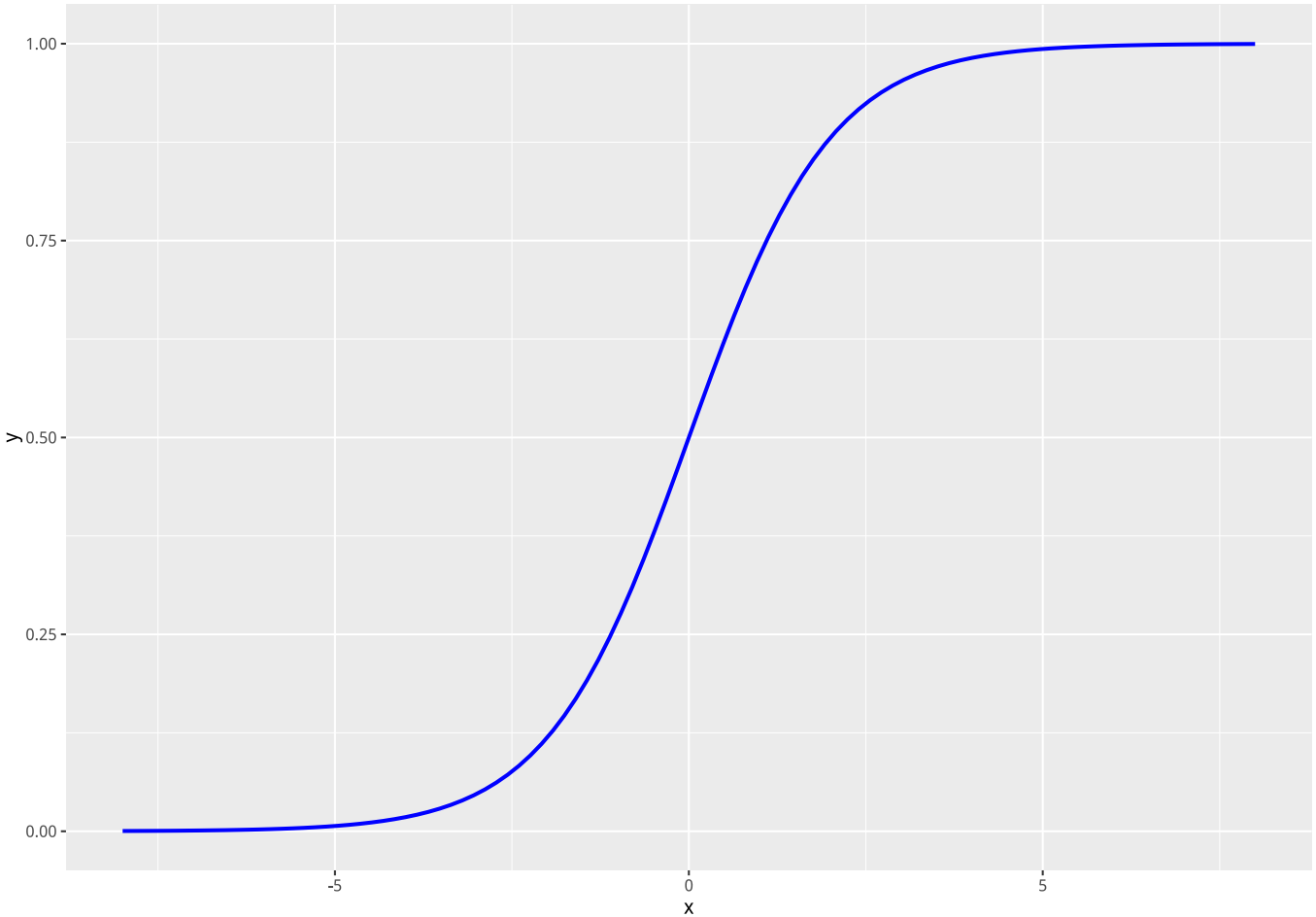
数据框 dataframe

```
x = seq(1,5,1)
y = x
d = data.frame(x,y)
d
```

```
##   x y
## 1 1 1
## 2 2 2
## 3 3 3
## 4 4 4
## 5 5 5
```

统计图形

R 快速绘制 sigmoid 函数



统计图形

R 快速绘制 sigmoid 函数

```
library(ggplot2)
logistic_fun = function(x){1/(1+exp(-x))}
ggplot(data.frame(x = c(-8,8)),aes(x)) +
  stat_function(fun = logistic_fun,
                color = "blue",
                size = 1)
```

统计分析-简单线性回归分析为例

构建回归模型

```
x = seq(1,20,1)
e = rnorm(20,0,1)
y = 2 * x + e
data1 = data.frame(x,y)
model = lm(y~x,data = data1)
```

统计分析-简单线性回归分析为例

模型描述

```
> summary(model)

Call:
lm(formula = y ~ x, data = data1)

Residuals:
     Min       1Q   Median       3Q      Max
-1.76870 -0.55364  0.06412  0.74244  1.10642

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.38148    0.37879  -1.007    0.327
x            1.97856    0.03162  62.571   <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8154 on 18 degrees of freedom
Multiple R-squared:  0.9954,    Adjusted R-squared:  0.9952
F-statistic:  3915 on 1 and 18 DF,  p-value: < 2.2e-16
```

统计分析-简单线性回归分析为例

回归方程的检验

- t 检验。回归系数的显著性检验，看 P 值
- F 检验。回归方程整体的显著性检验，看 P 值
- R^2 拟合优度检验。回归方程的拟合程度检验，看 R^2

回归方程

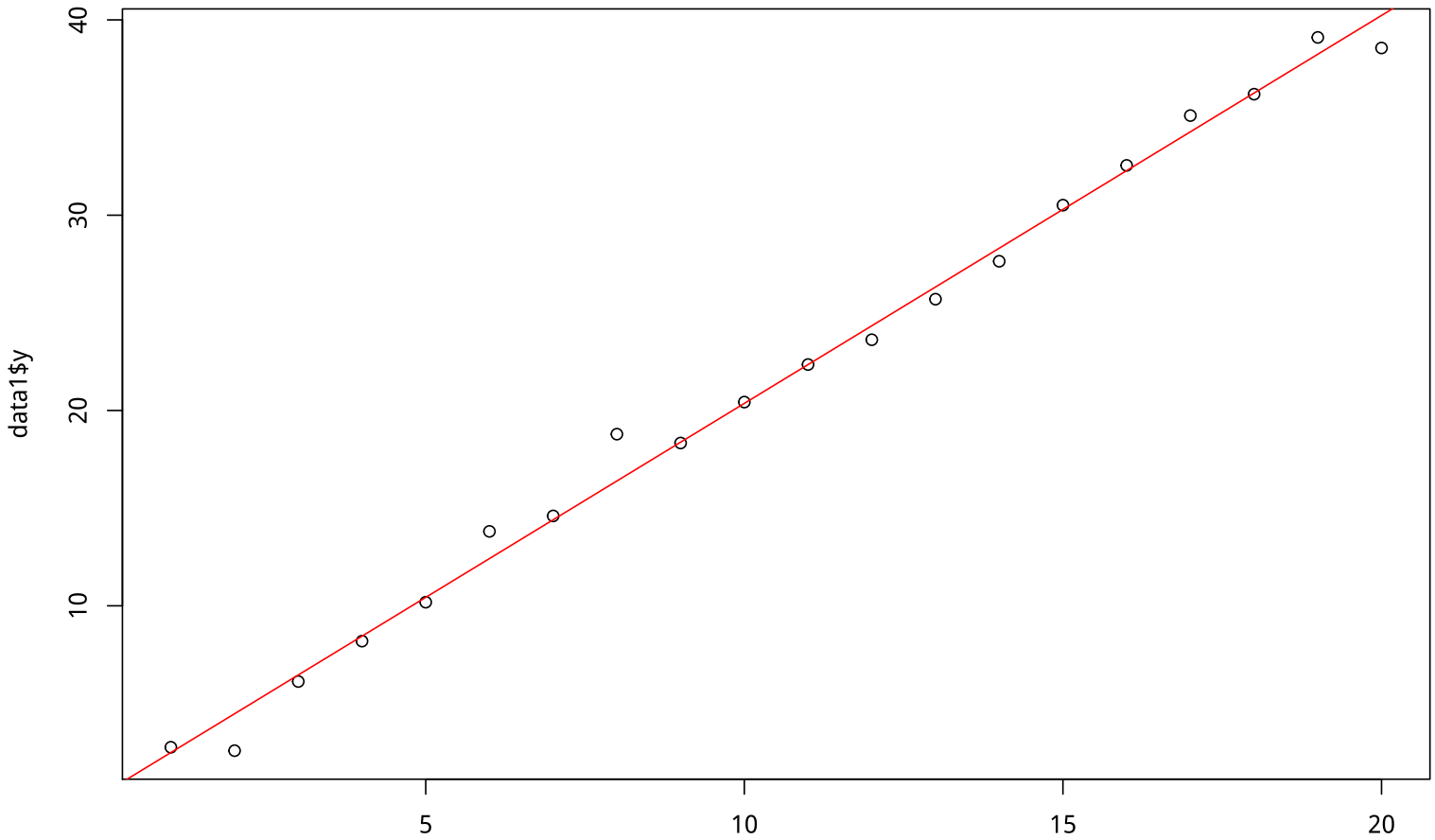
通过检验后，得到的回归方程为：

$$y = 1.97856x - 0.38148$$

统计分析-简单线性回归分析为例

真实值与拟合值

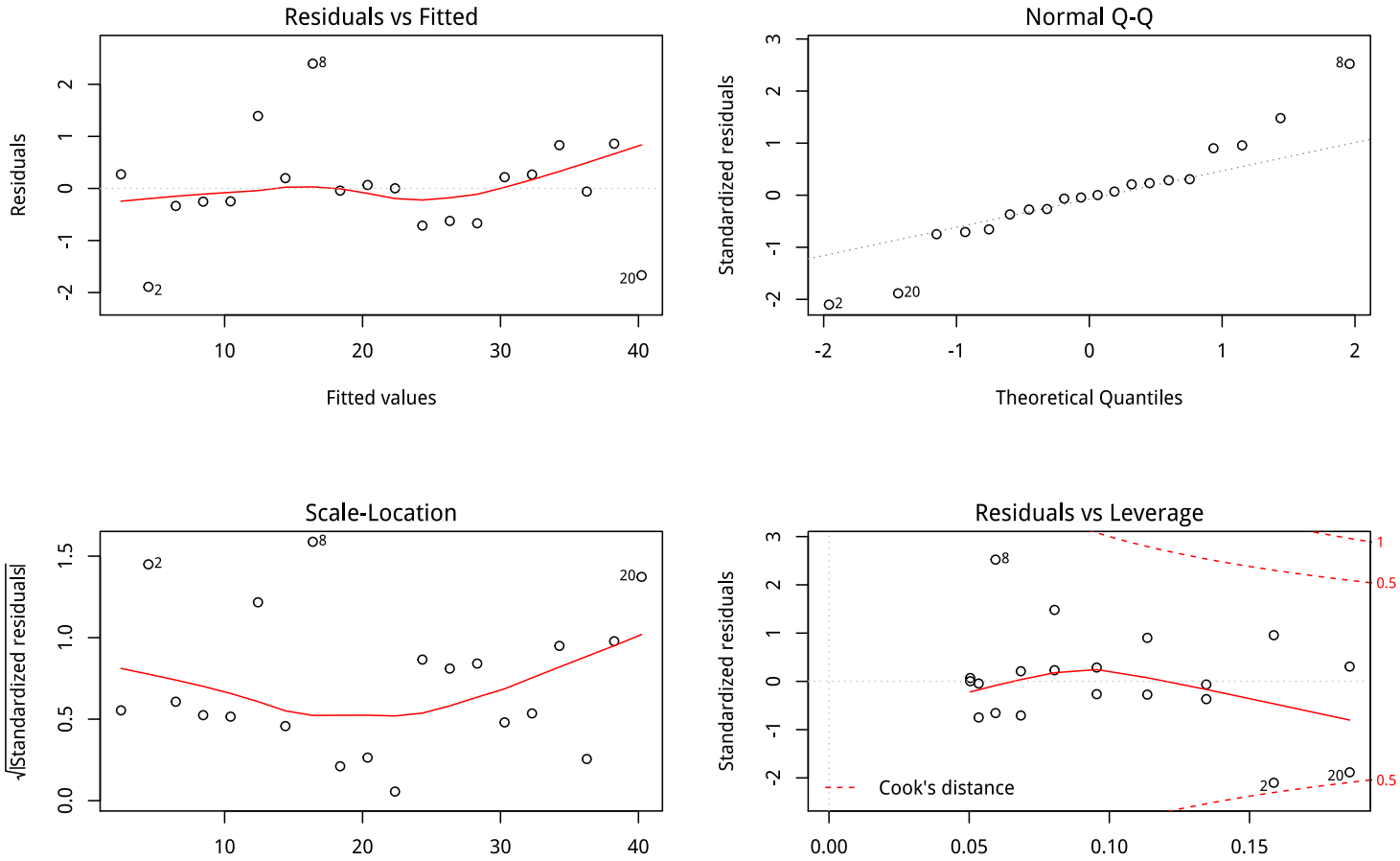
```
plot(data1$x,data1$y,type = "p")
abline(model,col="red",lwd=1)
```



统计分析-简单线性回归分析为例

模型检验

```
par(mfrow=c(2,2))
plot(model)
```



统计分析-简单线性回归分析为例

预测

```
v = 100
predict = as.numeric(model$coefficients) %*% c(1,v)
predict
```

```
##           [,1]
## [1,] 199.1463
```