Tutorial 5: Regression Lab 1

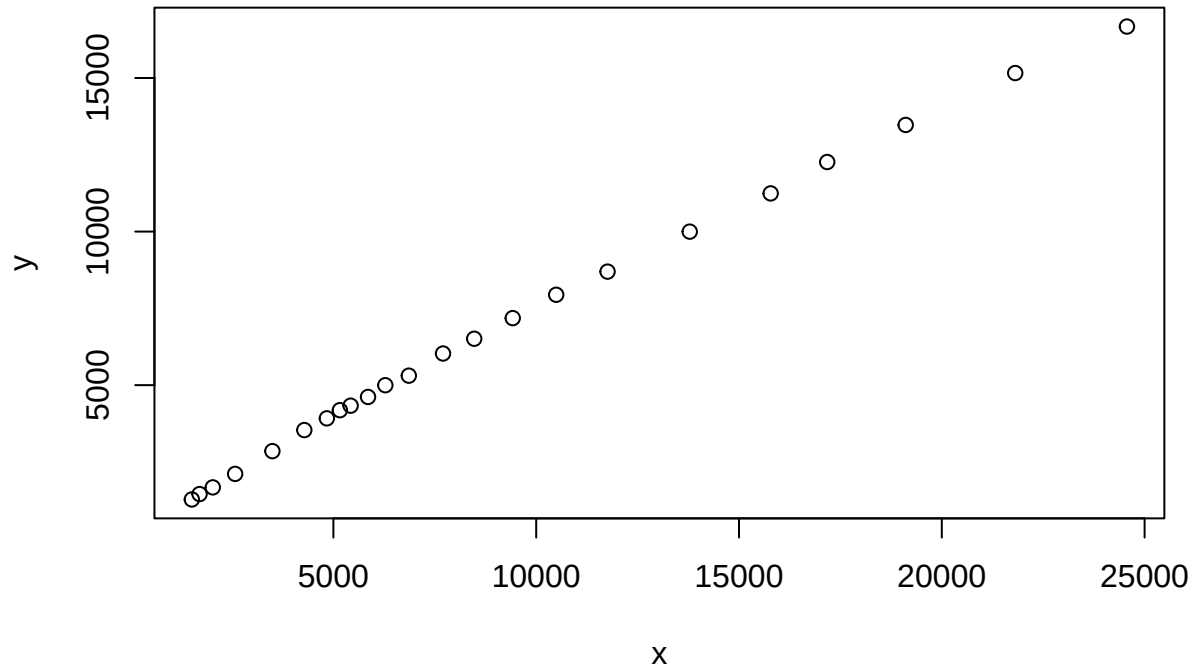
一个书上的例子

1. 数据准备

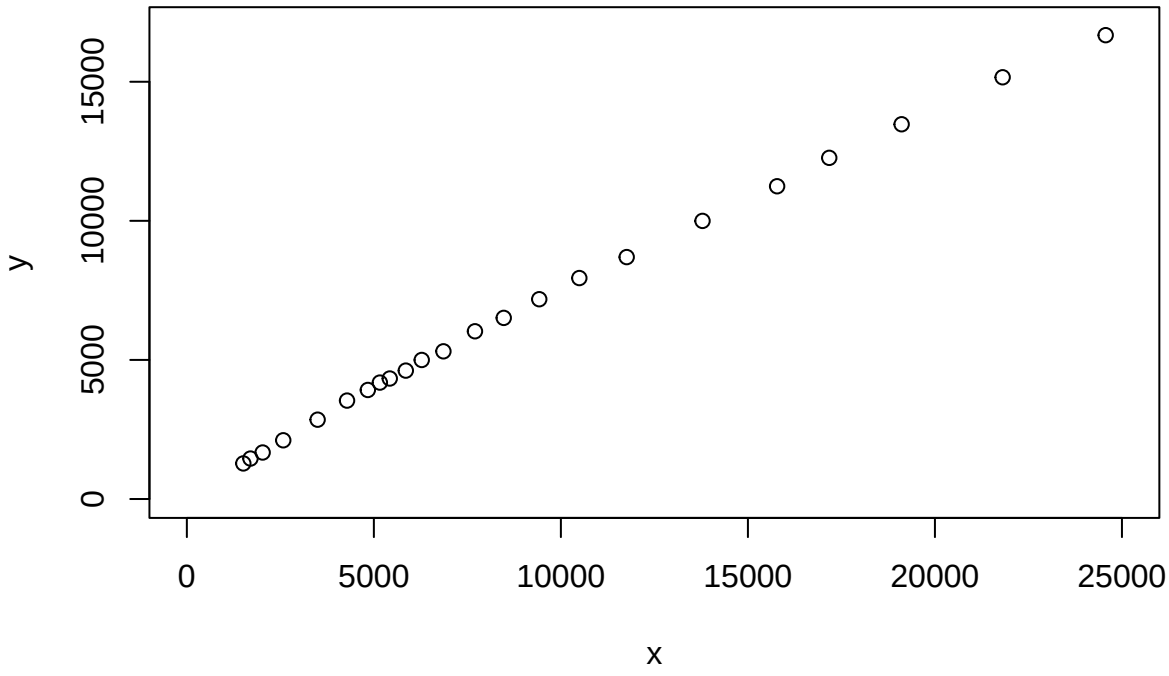
```
y <- c(1278.89, 1453.8, 1671.7, 2110.8, 2851.3, 3537.57, 3919.5, 4185.6, 4331.6, 4615.9, 4998, 530
x <- c(1510.16, 1700.6, 2026.6, 2577.4, 3496.2, 4282.95, 4838.9, 5160.3, 5425.1, 5854, 6279.98, 68
#y<-scale(y)
#x<-scale(x)
```

2. 基本绘图

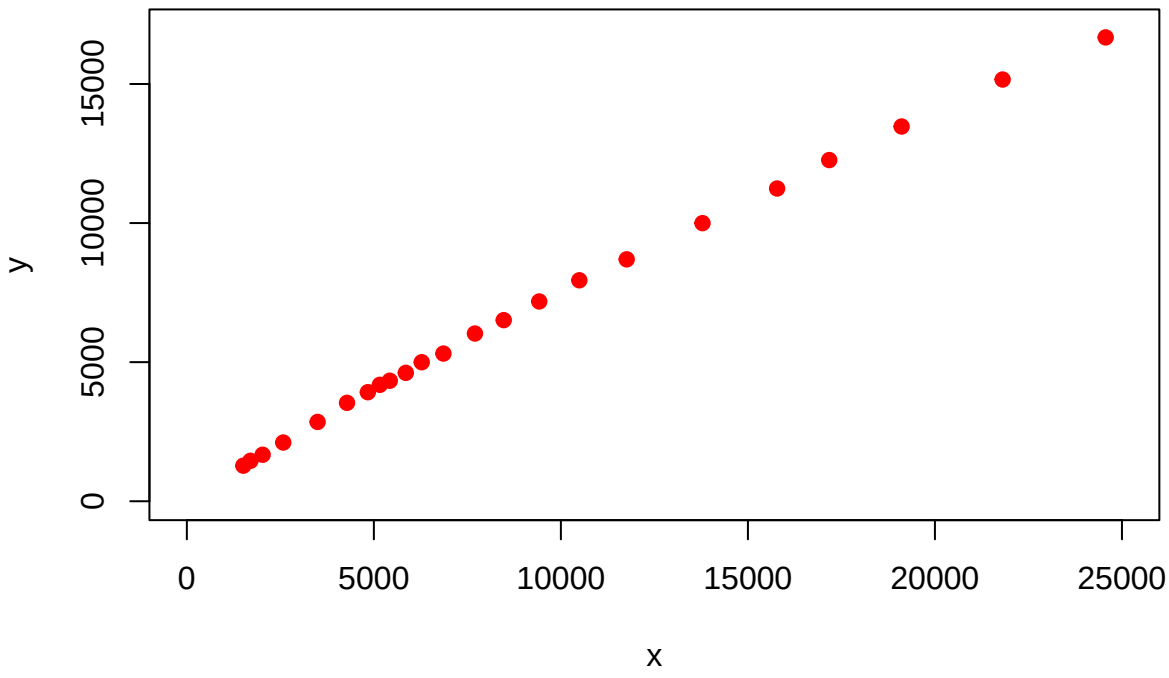
```
plot(x,y)
```



```r
#plot(x,y, xlab = " 给 x 轴标签",ylab = " 给 y 轴标签")
plot(x,y, xlim=c(0,25000),ylim=c(0,17000))
```




```r
plot(x,y, xlim=c(0,25000),ylim=c(0,17000),pch=19,col="red")
```




```
#pch: 点的类型
#col: 点的颜色
#bg: 填充色
#lty: 线型 1-6
```

##3. 数据探索

```
# 描述性统计表
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1510    4561    6860    9134   12773   24565
```

```
sd(x)
```

```
## [1] 6654.942
```

```
length(x)
```

```
## [1] 23
```

```
sort(x)
```

```
##  [1]  1510.16  1700.60  2026.60  2577.40  3496.20  4282.95  4838.90  5160.30
```

```
##  [9]  5425.10  5854.00  6279.98  6859.60  7702.80  8472.20  9421.60 10493.00
## [17] 11759.50 13785.80 15780.76 17174.65 19109.40 21809.80 24564.70
```

```
which(x>=1500)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
```

```
# 相关性分析
cor(x,y)
```

```
## [1] 0.9989422
```

##4. 线性回归拟合

```
# 最小二乘法的拟合结果
myfit <- lm(y~x)
coefficients(myfit)
```

```
## (Intercept)           x
## 609.2183015   0.6731794
```

```
confint(myfit)
```

```
##                   2.5 %      97.5 %
## (Intercept) 451.4973964 766.9392067
## x             0.6591171   0.6872417
```

```
fitted(myfit)
```

```
##         1         2         3         4         5         6         7         8
##  1625.827  1754.027  1973.484  2344.271  2962.788  3492.412  3866.666  4083.026
##         9        10        11        12        13        14        15        16
##  4261.284  4550.011  4836.772  5226.960  5794.585  6312.529  6951.645  7672.890
##        17        18        19        20        21        22        23
##  8525.472  9889.535 11232.501 12170.839 13473.273 15291.127 17145.669
```

```
residuals(myfit)# 残差
```

```
##           1           2           3           4           5           6
## -346.936916 -300.227202 -301.783689 -233.470907 -111.488147   45.157954
##           7           8           9          10          11          12
##   52.833862  102.574000   70.316093   65.889445  161.228481   82.050233
##          13          14          15          16          17          18
##  235.335356  198.411120  230.454590  269.990173  171.078453  107.935019
##          19          20          21          22          23
##   10.349027   93.710982   -1.822877 -130.236550 -471.348500
```

```
anova(myfit)# 方差分析
```

```
## Analysis of Variance Table
##
## Response: y
##           Df    Sum Sq   Mean Sq F value    Pr(>F)
## x          1 441542935 441542935  9910.9 < 2.2e-16 ***
## Residuals 21    935573     44551
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(myfit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -471.35 -120.86   65.89  134.58  269.99
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.092e+02  7.584e+01   8.033 7.71e-08 ***
## x           6.732e-01  6.762e-03  99.554  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 211.1 on 21 degrees of freedom
## Multiple R-squared:  0.9979, Adjusted R-squared:  0.9978
## F-statistic:  9911 on 1 and 21 DF,  p-value: < 2.2e-16
```

##5. 残差分析正态性：当预测变量值固定时，因变量成正态分布，则残差值（预测与真实的差值）也应该是一个均值为 0 的正态分布。“正态 Q-Q 图”（Normal Q-Q，右上）是在正态分布对应的值下，标准化残差的概图。若满足正态假设，那么图上的点应该落在呈 45 度角的直线上；若不是如此，那么就违反了正态性的假设。

独立性：你无法从这些图中分辨出因变量值是否相互独立，只能从收集的数据中来验证。比如，没有任何先验的理由去相信一位女性的体重会影响另外一位女性的体重。假若你发现数据是从一个家庭抽样得来的，那么可能必须要调整模型独立性的假设。

线性：若因变量与自变量线性相关，那么残差值与预测（拟合）值就没有任何系统关联。换句话说，除了白

噪声，模型应该包含数据中所有的系统方差。在“残差图与拟合图”（Residuals vs Fitted，左上）中可以清楚地看到一个曲线关系，这暗示着你可能需要对回归模型加上一个二次项。

同方差性：若满足不变方差假设，那么在“位置尺度图”（Scale-Location Graph，左下）中，水平线周围的点应该随机分布。

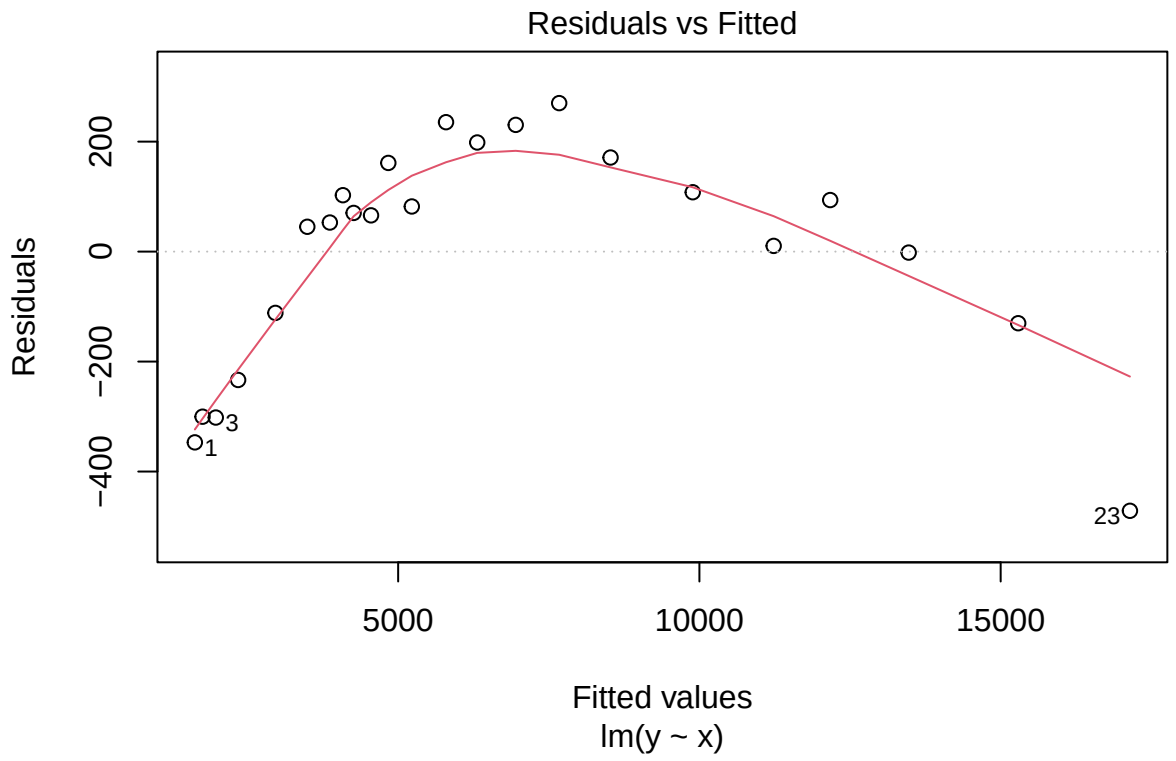
最后一幅“残差与杠杆图”（Residuals vs Leverage）提供了你可能关注的单个观测点的信息。从图形可以鉴别出离群点、高杠杆值点和强影响点。下面来详细介绍。

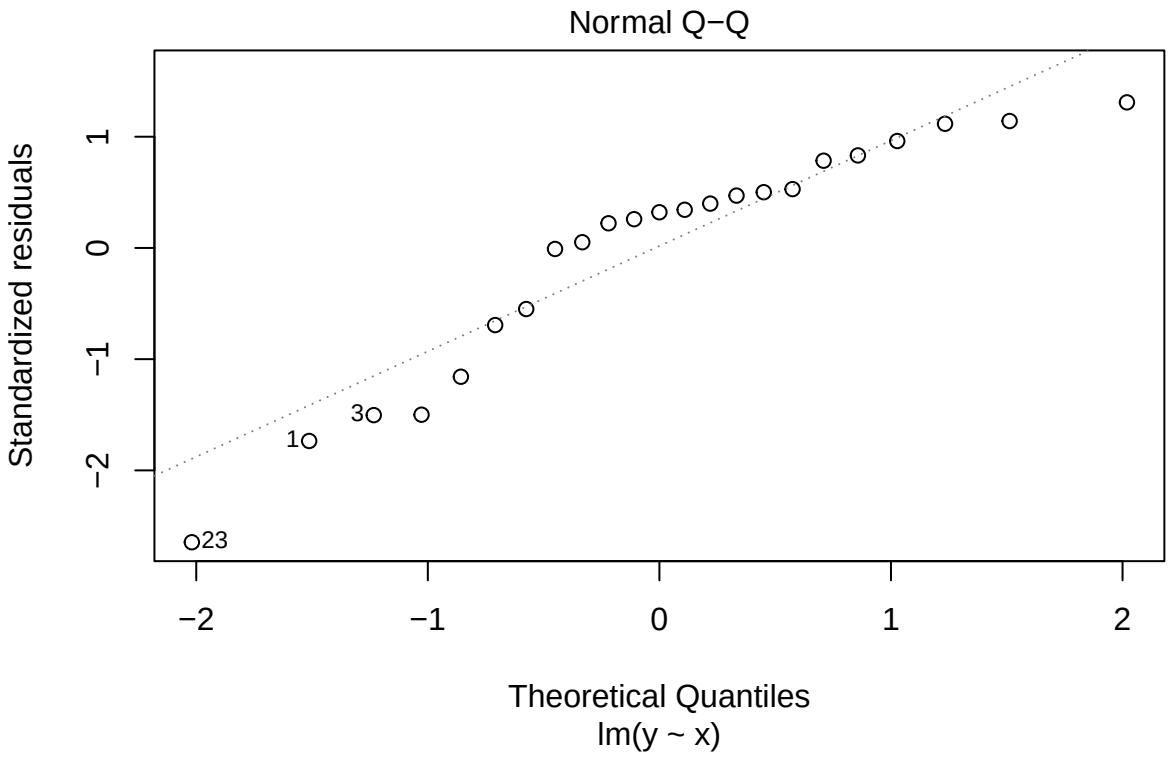
一个观测点是离群点，表明拟合回归模型对其预测效果不佳（产生了巨大的或正或负的残差）

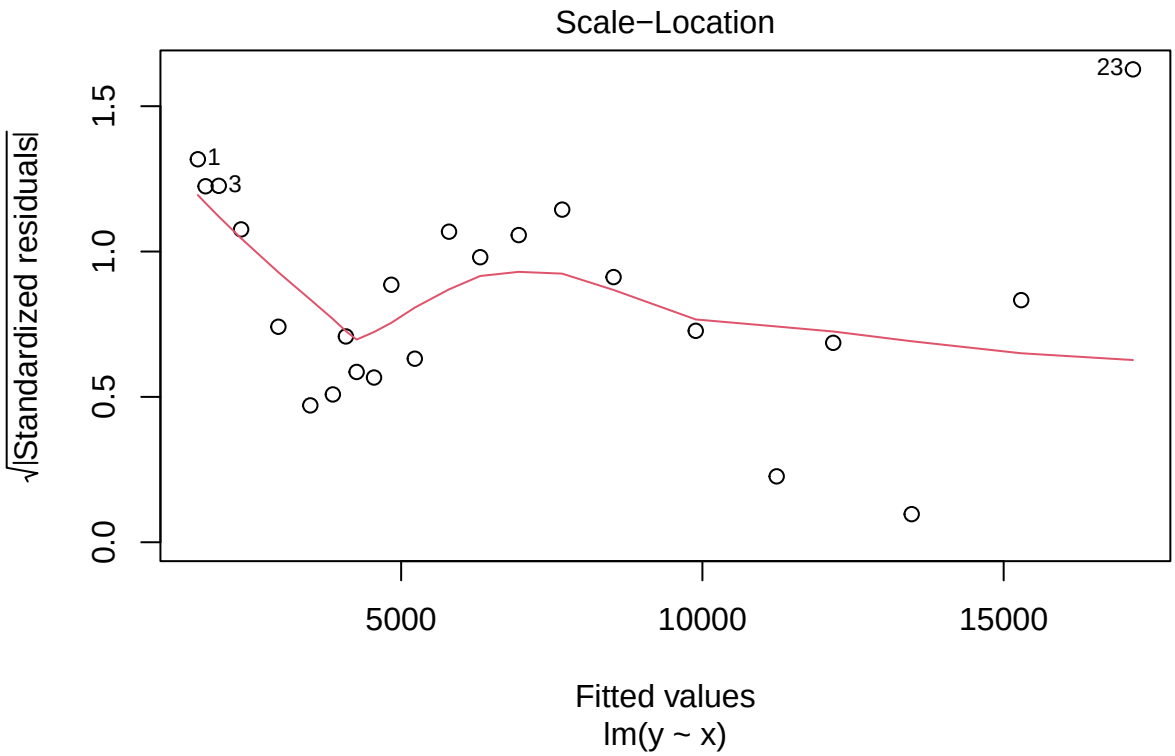
一个观测点有很高的杠杆值，表明它是一个异常的预测变量值的组合。也就是说，在预测变量空间中，它是一个离群点。因变量值不参与计算一个观测点的杠杆值。

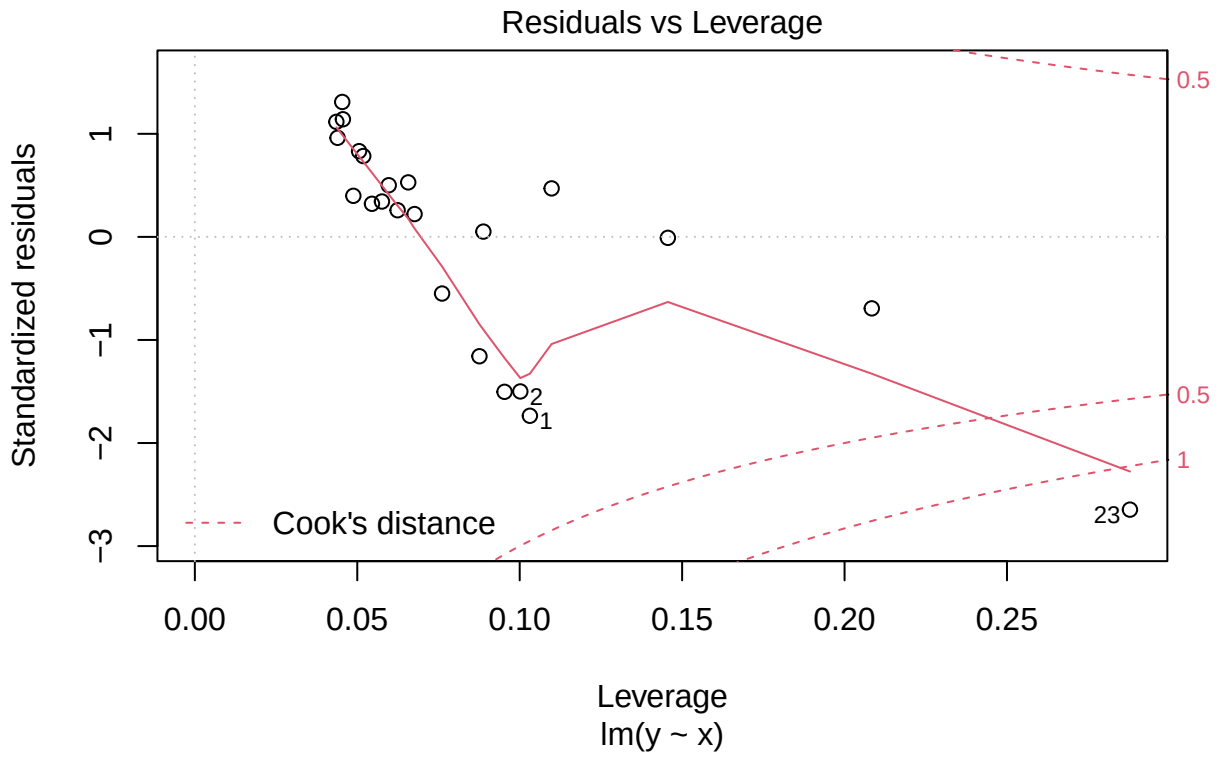
一个观测点是强影响点（influential observation），表明它对模型参数的估计产生的影响过大，非常不成比例。强影响点可以通过 Cook 距离即 Cook‘s D 统计量来鉴别。

```
plot(myfit)
```




Normal Q−Q
Standardized residuals
−2
−1
0
1
1
3
23
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
1
3
23
5000
10000
15000
Fitted values
lm(y ~ x)



6. 预测

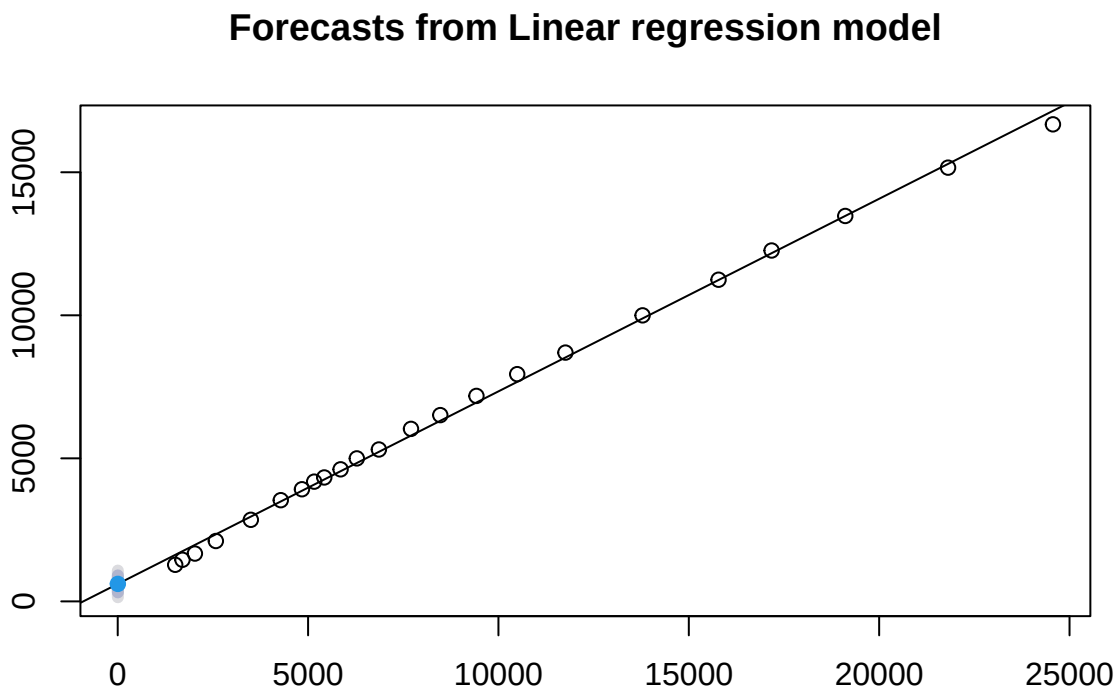
```
library(forecast)
```

```
## Warning: 程辑包'forecast'是用R版本4.1.3 来建造的
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

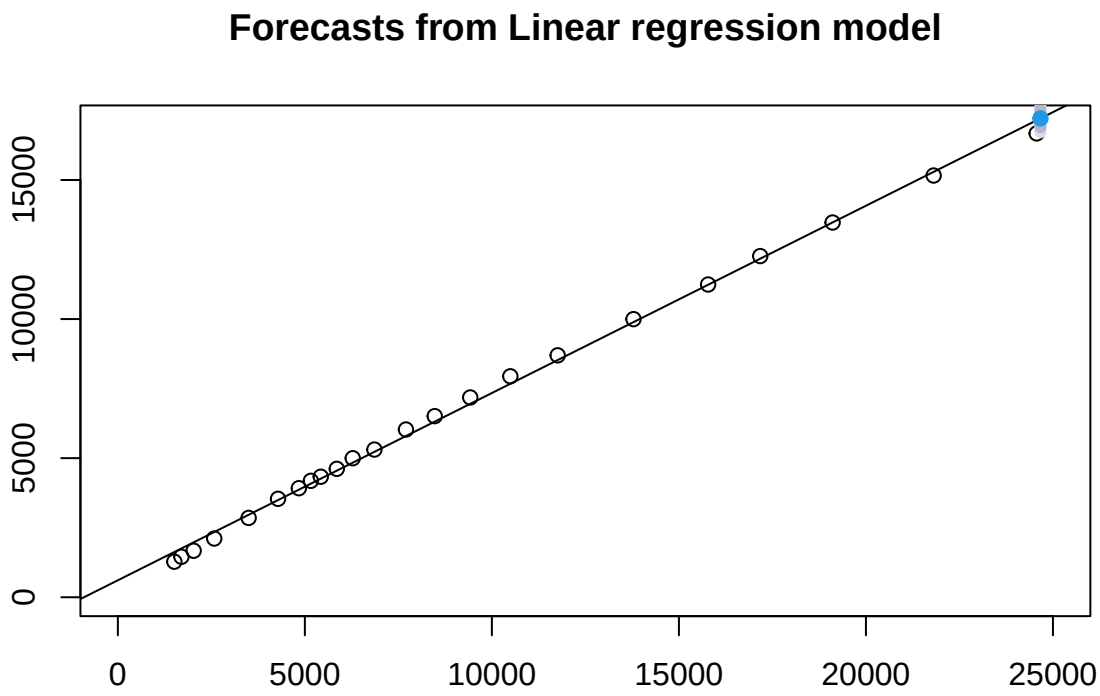
```
plot(forecast(myfit,3))
```

```
## Warning in forecast.lm(myfit, 3): newdata column names not specified, defaulting
## to first variable required.
```




```
plot(forecast(myfit,24666),xlim=c(0,25000),ylim=c(0,17000))
```

```
## Warning in forecast.lm(myfit, 24666): newdata column names not specified,
## defaulting to first variable required.
```




```
predict(myfit)
```

```
##         1         2         3         4         5         6         7         8
##  1625.827  1754.027  1973.484  2344.271  2962.788  3492.412  3866.666  4083.026
##         9        10        11        12        13        14        15        16
##  4261.284  4550.011  4836.772  5226.960  5794.585  6312.529  6951.645  7672.890
##        17        18        19        20        21        22        23
##  8525.472  9889.535 11232.501 12170.839 13473.273 15291.127 17145.669
```

```
predict(myfit,newdata=data.frame(x=c(400,500,600)))
```

```
##         1         2         3
##  878.4901  945.8080 1013.1259
```

7. 练习

7.1 向量基本操作

现有 10 个人的期末考试成绩为：100, 65, 80, 79, 88, 95, 93, 35, 56, 68

1. 创建向量 x 来存储上述数据；

```
x <- c(100, 65, 80, 79, 88, 95, 93, 35, 56, 68)
```

2. 将 x 从小到大排序，并分别找出最大值，最小值、中位数和第三大的元素；

```
x1 <- sort(x, decreasing = FALSE)
print(x1)
```

```
##  [1]  35  56  65  68  79  80  88  93  95 100
```

```
print(c(max(x1), min(x1), median(x1), x1[length(x1)-3+1]))
```

```
## [1] 100.0  35.0  79.5  93.0
```

3. 计算平均值、标准差和方差；

```
meanx <- mean(x)
stdx <- sd(x)
varx <- var(x)
print(c(meanx, stdx, varx))
```

```
## [1]  75.90000  20.16846 406.76667
```

4. 计算及格的人数；

```
pass <- x[x>=60]
cnt <- length(pass)
print(cnt)
```

```
## [1] 8
```

7.2 冬奥会

现在有中国 1992～2018 年在冬奥会上的奖牌统计数据 WinterOlym

1. 创建一个变量 medal 来存储上述数据；

```
medal <-  read.csv("WinterOlym.csv", header = TRUE)
```

2. 应用一元回归方法预测 2022 冬奥会中国总奖牌数目；

```
t <- medal[,c(1,5)]
olst <- lm(Total~Time, data=t)
predictt <- predict(olst, newdata = data.frame(Time = 2022), interval = "confidence")
print(predictt)
```

```
##        fit      lwr      upr
## 1 12.29136 8.013296 16.56943
```

3. 应用一元回归方法分别预测 2022 冬奥会中国金银铜牌数目；

```r
olsg <- lm(Gold~Time, data=medal)
olss <- lm(Silver~Time, data=medal)
olsb <- lm(Bronze~Time, data=medal)
predictg <- predict(olsg, newdata = data.frame(Time = 2022))
predicts <- predict(olss, newdata = data.frame(Time = 2022))
predictb <- predict(olsb, newdata = data.frame(Time = 2022))
print(c(predictg, predicts, predictb))
```

```
##        1        1        1
## 3.652139 4.961259 3.677966
```

4. 结合中国实际获得的奖牌情况，对比分析 2 和 3 的预测结果；

```r
print(predictt - (predictg+ predicts+ predictb))
```

```
##            fit       lwr      upr
## 1 7.105427e-14 -4.278068 4.278068
```

发现中国其实拿了 15 块奖牌，与预测的 12~13 块差别不大；但金牌 9 块与预测的 3.6 差别极大，考虑到只是做了线性回归，而获奖情况不只是往年的成绩就能预测，由此才会产生这种可以理解的差别。

5. 从高斯马尔可夫条件出发，分析 2 中创建的模型好坏。

很难说明往年的奖牌总数数据的误差是零均值，同方差且不相关的。因此建立线性回归模型的前提并不能很好满足，故模型的预测很可能不准确。

8. 参考

R 语言统计-回归篇：回归诊断 https://zhuanlan.zhihu.com/p/341318994 R 语言数据分析: 线性回归 https://zhuanlan.zhihu.com/p/378228742