Data Analysis Learning

stars 1 commits 73/year build passing license GPL-3.0

关于 CWorld 学习 Analysis Learning 一些笔记和代码。该课程使用 R 语言进行数据分析。

Get started

Hint

点击侧栏的目录或下滑以阅览更多章节。 当然，你也可以下载 `PDF 版本` 的笔记。它来自 Github Actions 的自动构建，并时刻保持最新。

Development

如果你对该项目有兴趣，请前往 Github 了解更多。

Contributions

由于作者只是个正在浅学 Database 的初学者，所以笔记难免存在明显纰漏，还请读者们多多海涵。此外，也欢迎诸位使用 PR 或 Issues 来改善它们。

Thanks

一些电子教材对作者学习上帮助颇多，没有这些资料，就没有这部笔记。在此对这些教材的原作者深表感谢。读者若对此项目笔记抱有疑惑，也可以仔细阅读以下教材以作弥补。

- STATS 201 : Data Analysis
- STATS201 book SWU 2023

Skip to main content

Table of Contents

At the beginning

章节

学习提要

本门课程主要研究：线性回归模型、常见问题的解决方法

分数分布

Skip to main content

平时分数	期末测验
20% 作业 +20% 课堂	60% 期末考试

环境搭建

本课程使用工具：R Language（交互式、开放、免费）

1. 安装 R Studio
2. 安装 R Tools
3. 安装 RMarkdown 库

1. Getting Started with Regression

1.1. 什么是线性回归

线性样本回归分析：

$$\hat{y_0} = a_i + b_i x$$

原则：残差平方和最小

怎么算 a_i 和 b_i：

$$\begin{cases} b = \frac{\sum_{i=1}^{n}(x-x_i)(y-y_i)}{\sum_{i=1}^{n}(x-\bar{x})^2} \\ a = \bar{y} - b\bar{x} \end{cases}$$

1.2. 线性回归的残差与模型误差分析

残差表示预测值与真实值的差值，有正负号，一般使用 ε 表示。

$$y_i = ax_i + b + \varepsilon$$

Skip to main content

且 ε 的值符合正态分布：$\varepsilon \sim N(0,\sigma^2)$

误差：

$$\begin{aligned}
Y-\hat{Y} &= Y-\bar{Y}-\hat{Y}+\bar{Y} \\
&= (Y-\bar{Y})-(\hat{Y}-\bar{Y}) \\
Y-\bar{Y} &= (Y-\hat{Y})+(\hat{Y}-\bar{Y})
\end{aligned}$$

其中 $Y-\bar{Y}$ 称为总体差异，$Y-\hat{Y}$ 称为随机变量，$\hat{Y}-\bar{Y}$ 称为可以用自变量 x 进行解释的差异。

于是，我们有：

$$\begin{array}{ccccc}
\sum Y-\bar{Y} = & \sum Y-\hat{Y} & + & \sum \hat{Y}-\bar{Y} \\
SST = & SSE & + & SSR \\
df=n-1 & df=n-2 & & df=1
\end{array}$$

并且有：

$$\begin{cases}
MST & = \frac{SST}{df} \\
MSE & = \frac{SSE}{df} \\
MSR & = \frac{SSR}{df}
\end{cases}$$

2. Basics of Simple Linear Regression

本课程前置需要装的包：

```
require(s20x)
```

▶ Show code cell output

2.1. 分析数据过程

Skip to main content

读取数据表格，`header=TRUE` 表示第一行是表头，`sep=","` 表示分隔符是逗号。

```r
course.df <- read.table("../data/STATS20x.txt", header = TRUE, sep = "\t")
head(course.df) # 看前面大约10行的内容
dim(course.df) # 看有多少行、多少列
course.df$Exam[1:20] # 看前20行的Exam列
```

A data.frame: 6 × 15

	Grade	Pass	Exam	Degree	Gender	Attend	Assign	Test	B	C	MC	Colc
	<chr>	<chr>	<int>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<int>	<int>	<int>	<ch
1	C	Yes	42	BSc	Male	Yes	17.2	9.1	5	13	12	Bl
2	B	Yes	58	BCom	Female	Yes	17.2	13.6	12	12	17	Yell
3	A	Yes	81	Other	Female	Yes	17.2	14.5	14	17	25	Bl
4	A	Yes	86	Other	Female	Yes	19.6	19.1	15	17	27	Yell
5	D	No	35	Other	Male	No	8.0	8.2	4	1	15	Bl
6	A	Yes	72	BCom	Female	Yes	18.4	12.7	15	17	20	Bl

```
146 · 15
42 · 58 · 81 · 86 · 35 · 72 · 42 · 25 · 36 · 48 · 29 · 54 · 49 · 52 · 28 · 34 · 51 · 81 · 80 · 41
```

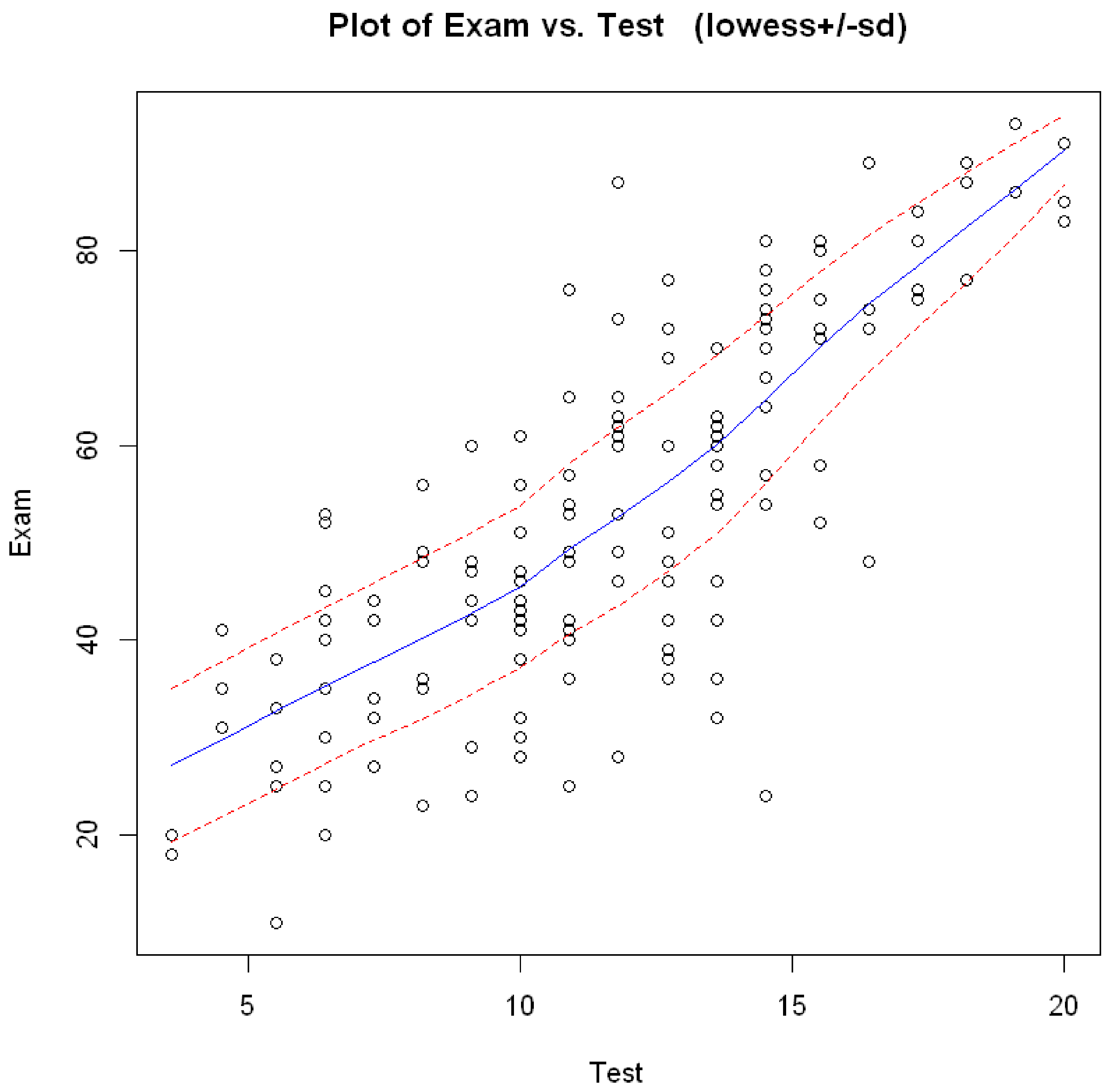
2.1.2. 绘图观测数据

对数据进行绘图分析，着重分析 `Exam` 和 `Test` 两个变量之间的关系。

首先应当粗略查看两者的关系，如线性、二次、曲线、正弦等

```r
library(s20x)
trendscatter(Exam ~ Test, data = course.df)
```

Skip to main content



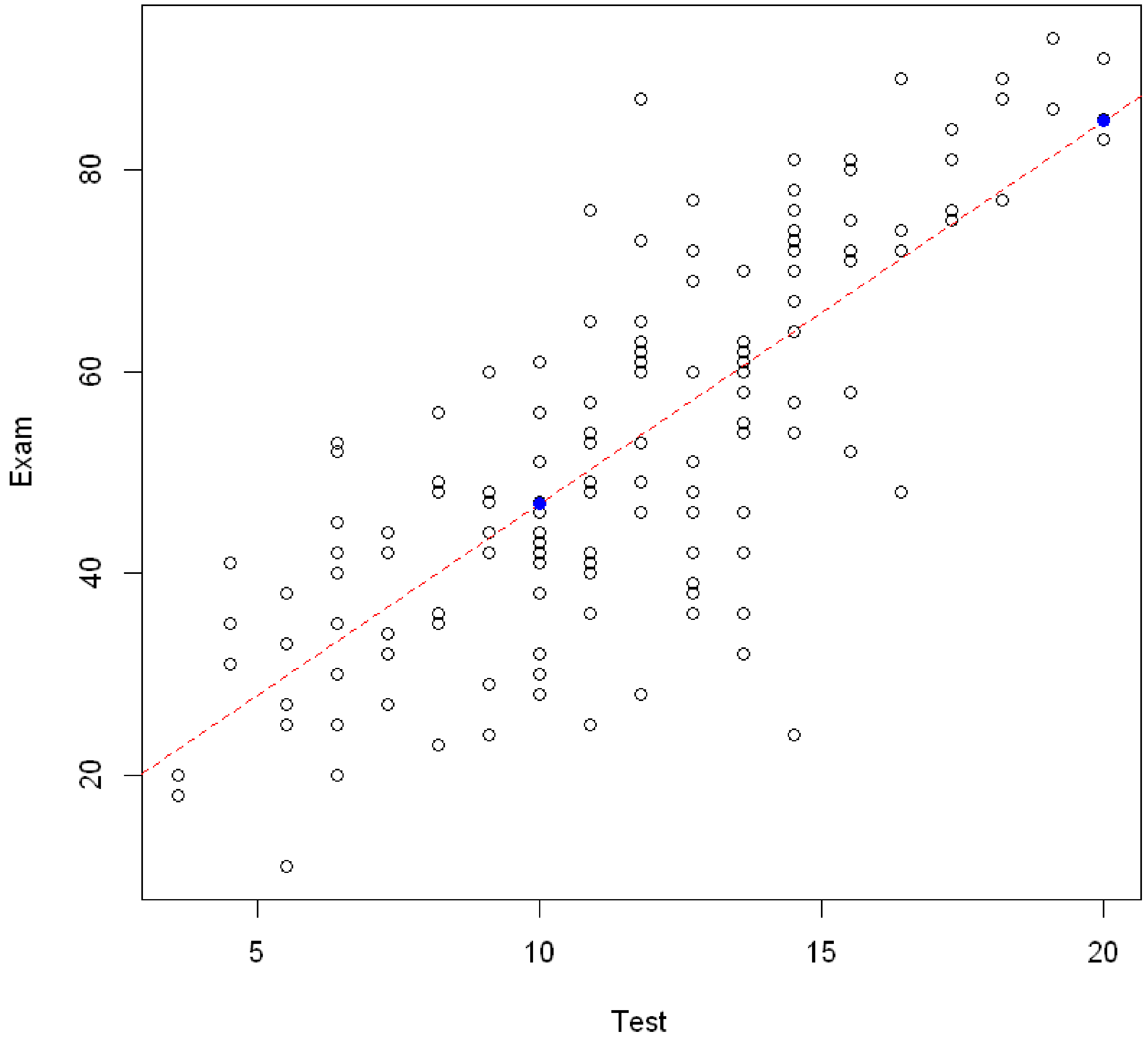
2.1.3. 进行初步拟合

可以看到整体大致呈线性关系，故我们采用线性回归模型。

Skip to main content

```r
plot(Exam ~ Test, data = course.df)
# 绘制回归直线
examtest.fit <- lm(Exam ~ Test, data = course.df)
# lty = 2 表示虚线，col = "red" 表示红色
abline(examtest.fit, lty = 2, col = "red")

points(
    0,
    predict(examtest.fit, newdata = data.frame(Test = 0)),
    col = "blue",
    pch = 19
)
points(10, predict(examtest.fit,
    newdata = data.frame(Test = 10)
), col = "blue", pch = 19)
points(20, predict(examtest.fit,
    newdata = data.frame(Test = 20)
), col = "blue", pch = 19)
```

Skip to main content



```
summary(examtest.fit)
```

Skip to main content

```
Call:
lm(formula = Exam ~ Test, data = course.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-39.980  -6.471   0.826   8.575  33.242 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   9.0845     3.2204   2.821  0.00547 ** 
Test          3.7859     0.2647  14.301  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.05 on 144 degrees of freedom
Multiple R-squared:  0.5868,	Adjusted R-squared:  0.5839 
F-statistic: 204.5 on 1 and 144 DF,  p-value: < 2.2e-16
```

其中：

- Call：表示回归方程，指明了自变量和因变量
- Risiduals：残差，指明了残差的分布，如最大、最小、中值等
- Coefficients：系数，此处即 a_i 和 b_i 的值
- Residual standard error：残差标准差，即残差的标准差
- Multiple R-squared：多元 R^2 值
- Adjusted R-squared：调整后的 R^2 值
- F-statistic：F 统计量，即 F 统计量。F 统计量的分子是回归平方和，分母是残差平方和。F 统计量的值越大，说明回归平方和越大，即回归模型的拟合效果越好。F 统计量的值越小，说明回归平方和越小，即回归模型的拟合效果越差。p-value 则相反。

2.2. 分析数据是否可以接受

2.2.1. 残差观测

针对指定行分析预测值和残差：

```
data.frame(course.df$Test[1], course.df$Exam[1]) # 原第一行
# 按照 tidyverse 的风格，也可以使用 dplyr 包的 select 函数来选择列
# dplyr::select(course.df[1, ], Exam, Test)
fitted(examtest.fit)[1] # 拟合值
```

Skip to main content

A data.frame: 1 × 2

course.df.Test.1.	course.df.Exam.1.
<dbl>	**<int>**
9.1	42

1: 43.5363712056028

1: -1.53637120560281

检验上，一个成功的拟合模型的残差应当有：

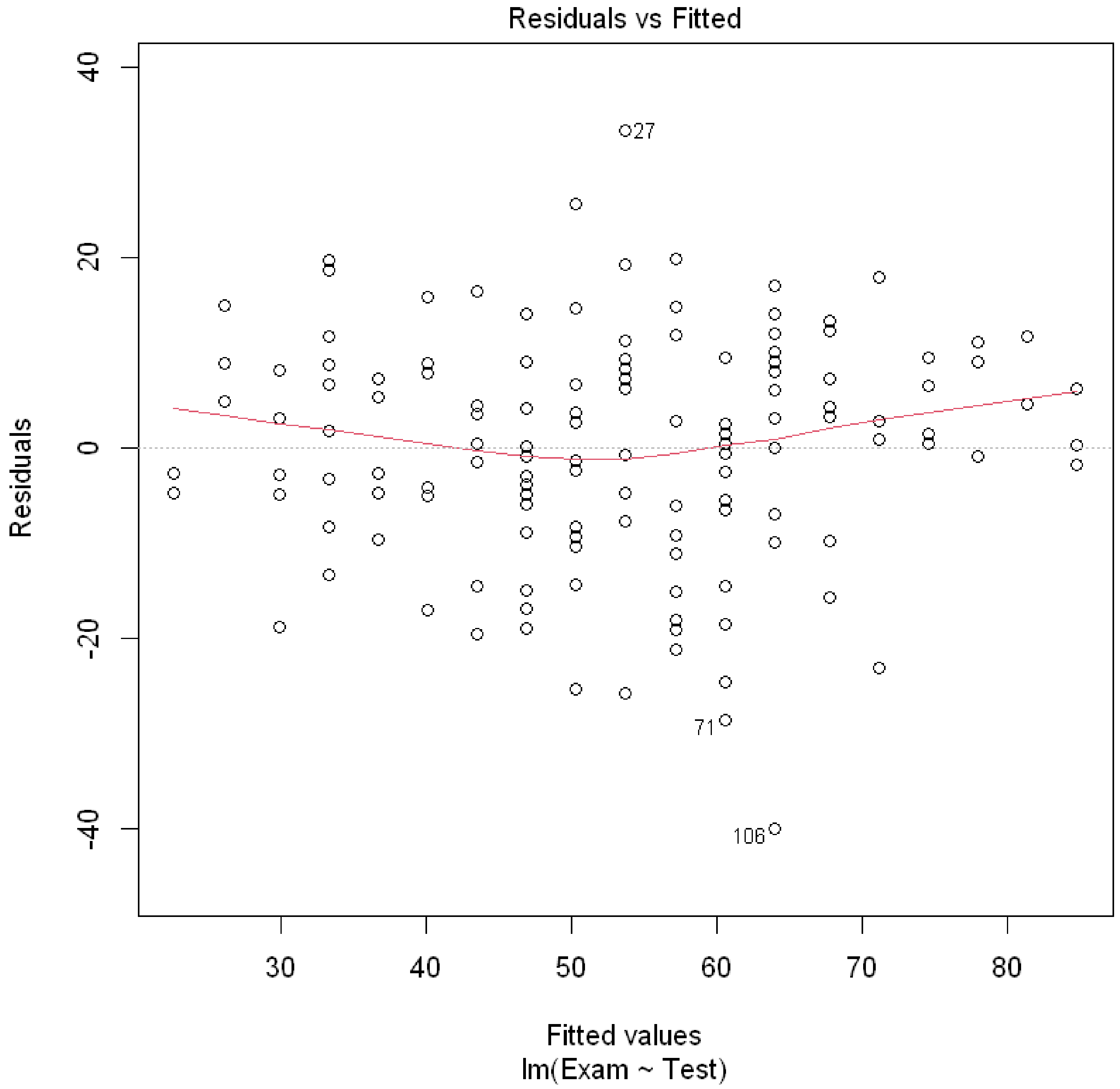
1. 残差均值接近于 0
2. 残差满足正态分布
3. 没有或排除了异常点

2.2.1.1. 残差均值接近于 0

分析残差，看是否符合均值等于0

```r
# 其中 which = 1 表示残差直方图（histogram of residuals），
# which = 2 表示残差QQ图（qqplot，即 normal quantile-quantile-plot），
# which = 3 表示残差标准化图
plot(examtest.fit, which = 1)
```

Skip to main content



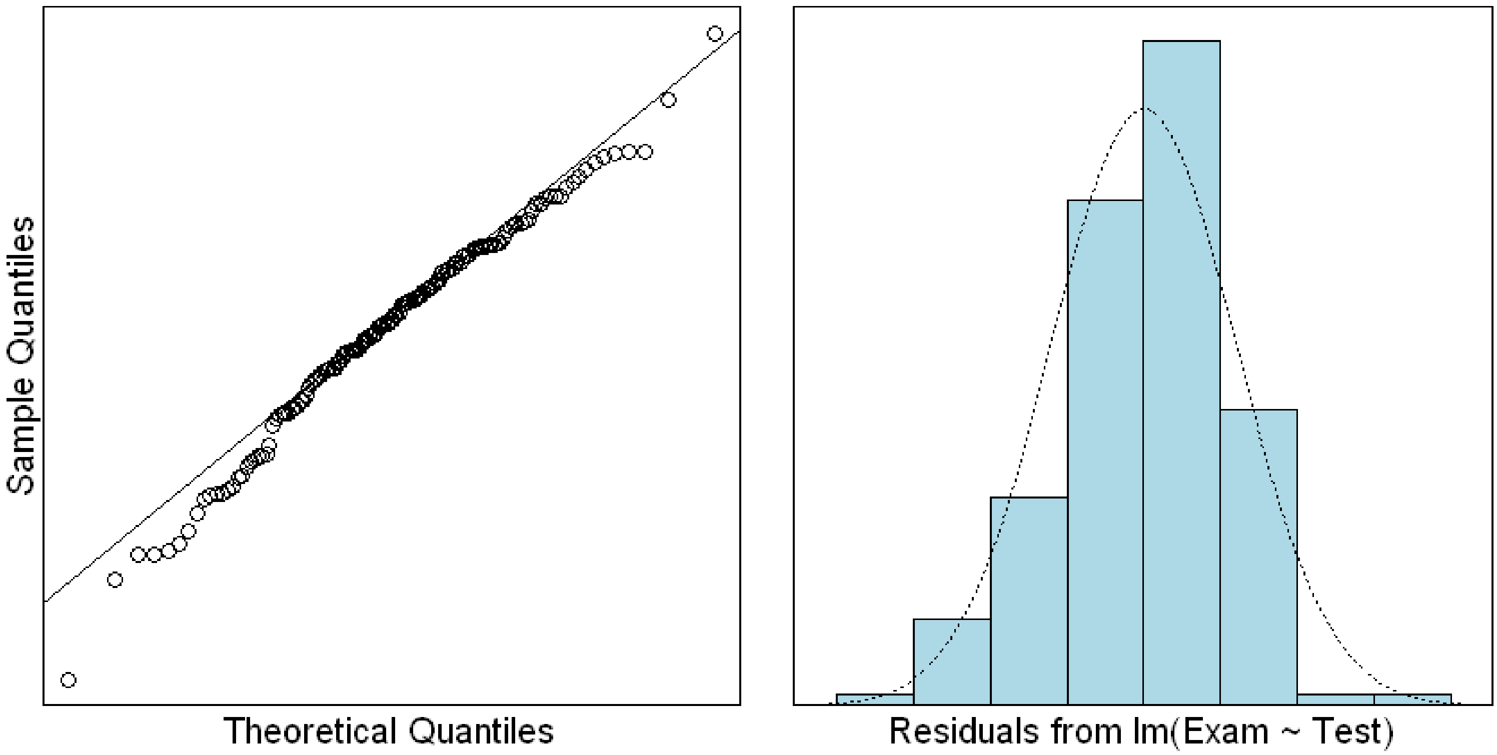
2.2.1.2. 残差满足正态分布

残差在分布上在符合正态同分布：iid – independence（并且这是根据学生在考试中应该相互独立的表现）。残差应该有大致恒定的散布。这其实是 Equality Of Variance (EOV，方差相等) 原则。

检查残差是否满足正态分布：

Skip to main content

```
normcheck(examtest.fit)
```



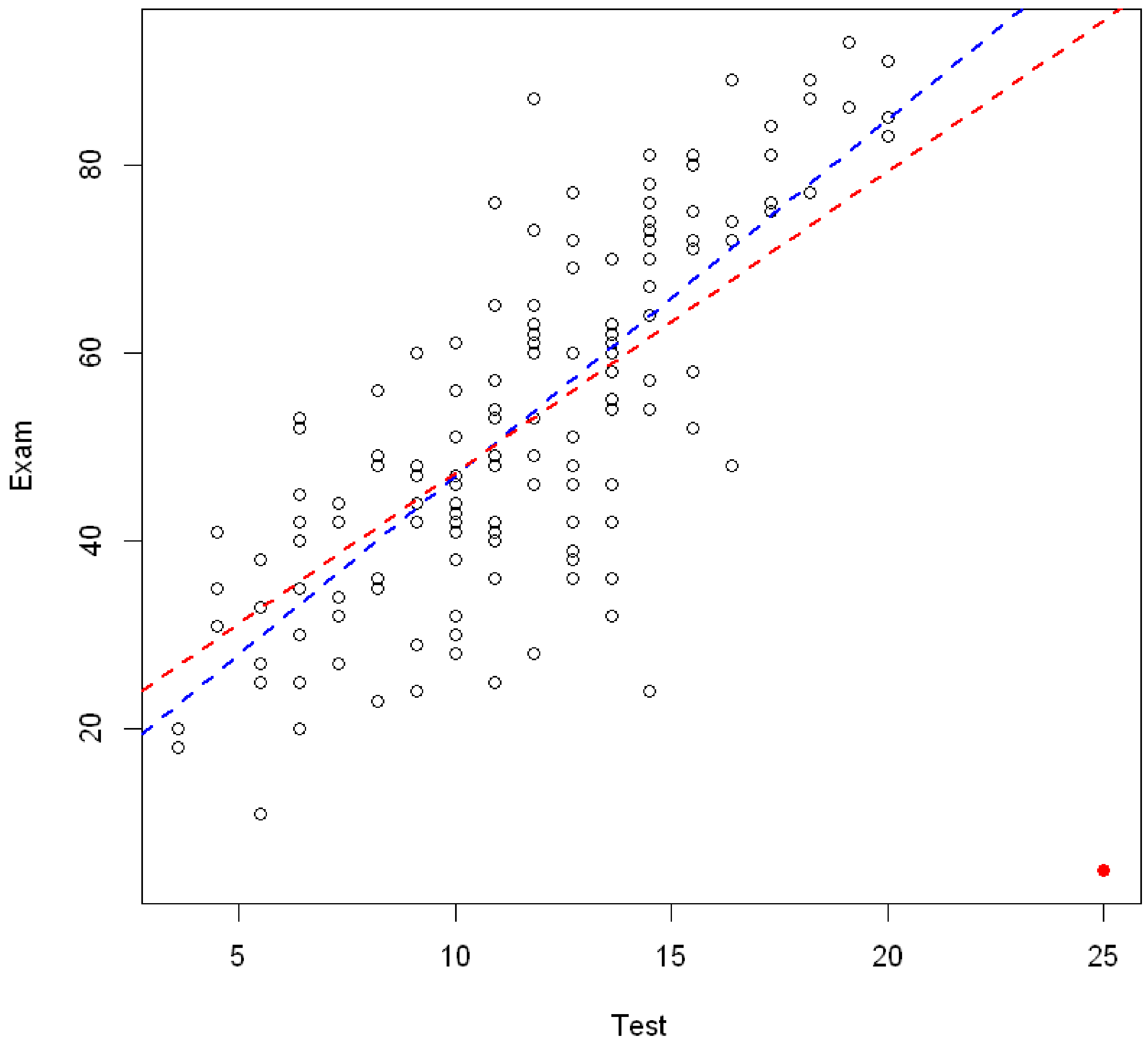
Skip to main content

```r
# 创造一个包含异常点的数据集并验证异常点对回归直线的影响
n <- nrow(course.df)
# 复制一数据集的最后一行
course2.df <- course.df[c(1:n, n), ]
# 修改新数据集的最后一行的 Test 和 Exam 列的值，故意创造一个差异极大的观测值
course2.df[n + 1, c("Test", "Exam")] <- c(25, 5)
# 画出散点图
plot(Exam ~ Test, data = course2.df)
## 并标记我们创建的新的观测点
points(25, 5, pch = 19, col = "red")

# 如果有的观测值是异常值，那么回归直线就会受到影响
examtest2.fit <- lm(Exam ~ Test, data = course2.df)
summary(examtest2.fit)

# 或者直接画图验证该点造成的影响
abline(examtest.fit, lty = 2, lwd = 2, col = "blue")
abline(examtest2.fit, lty = 2, lwd = 2, col = "red")
```

Skip to main content

```
Call:
lm(formula = Exam ~ Test, data = course2.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-90.251  -6.846   2.638   9.456  33.996 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  15.2374     3.7172   4.099 6.88e-05 ***
Test          3.2006     0.3023  10.588  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.34 on 145 degrees of freedom
Multiple R-squared:  0.436,	Adjusted R-squared:  0.4322 
F-statistic: 112.1 on 1 and 145 DF,  p-value: < 2.2e-16
```

Skip to main content

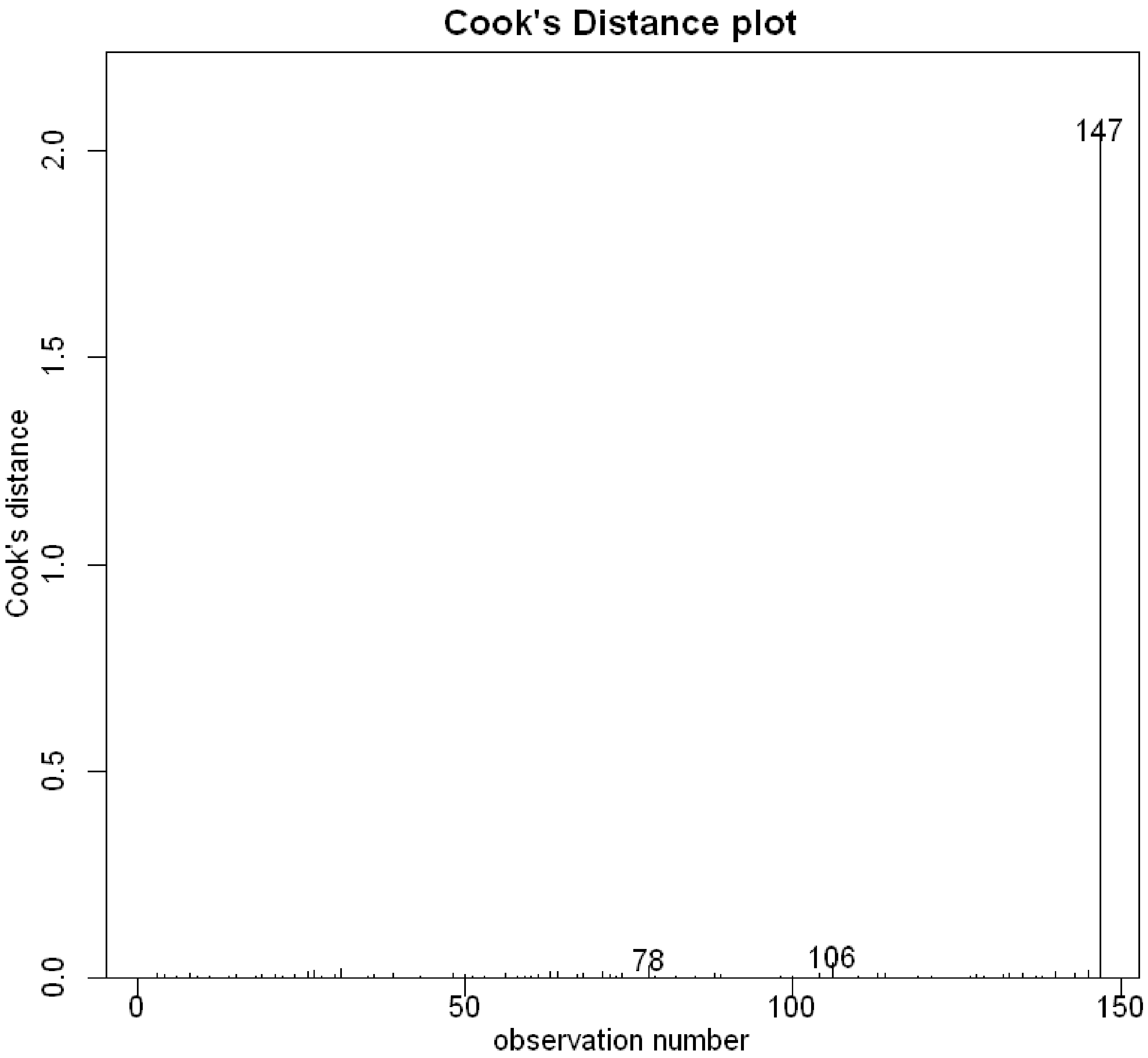


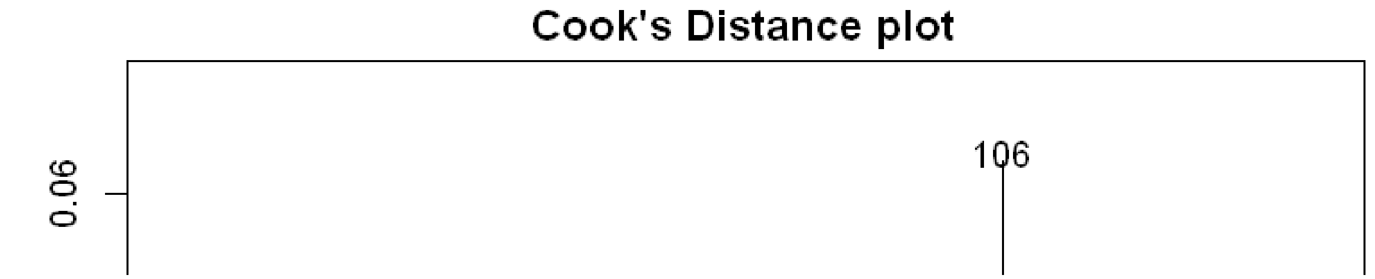
对其进行观测值差异分析：

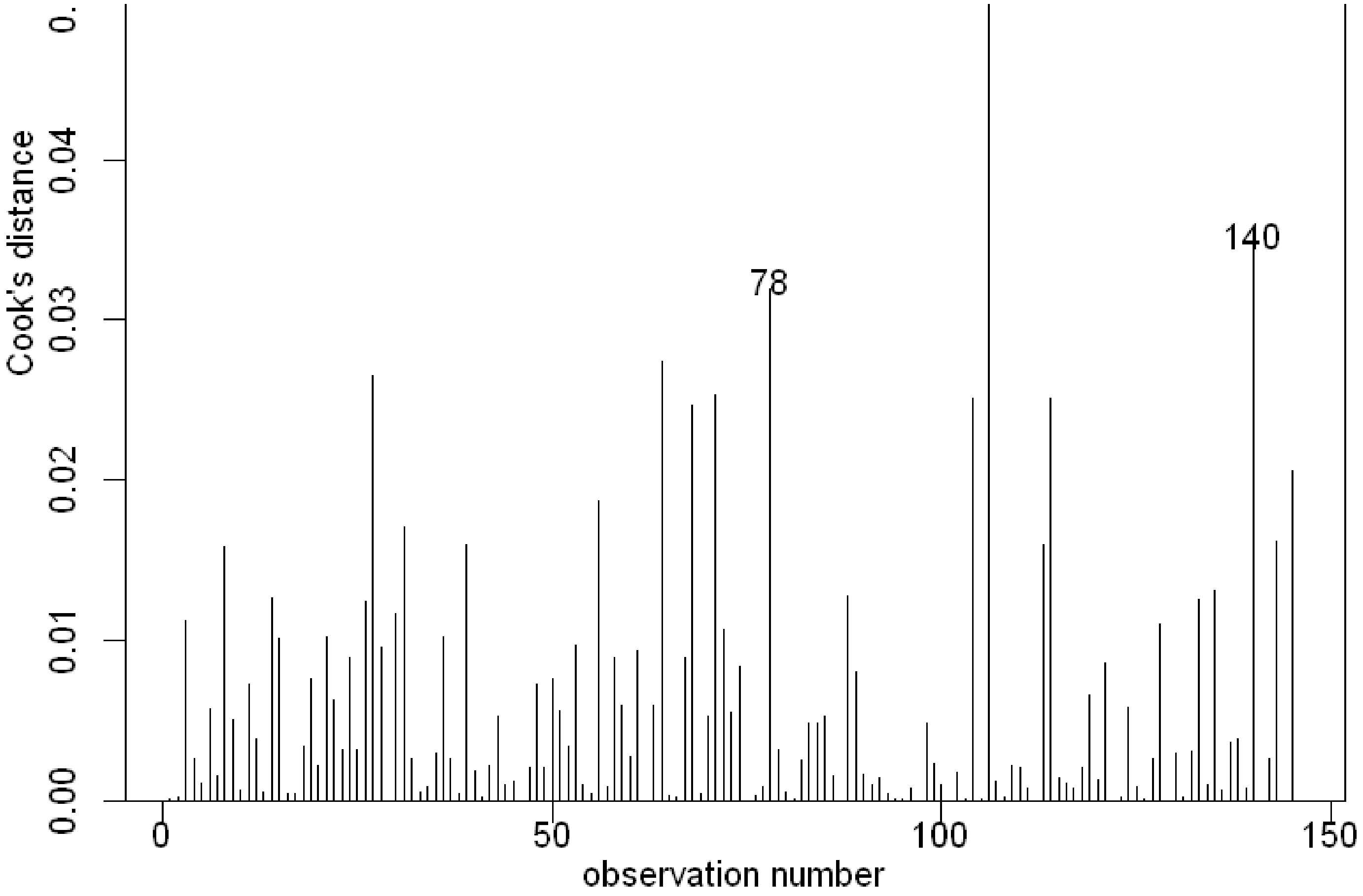
```
# 画出异常值的影响
cooks20x(examtest2.fit)
# 对比原来的值影响
cooks20x(examtest.fit)
```

Skip to main content

Skip to main content


Cook's Distance plot
Cook's distance
0.0
0.5
1.0
1.5
2.0
147
78
106
0
50
100
150
observation number


Cook's Distance plot
0.06
106



2.2.2. R 方观测

R Squared 即 R 平方，是回归平方和与总平方和的比值，即 $R^2 = \frac{SSR}{SST}$ ，其中 SSR 为回归平方和，SST 为总平方和。R 平方的值越大，说明回归平方和越大，即回归模型的拟合效果越好。R 平方的值越小，说明回归平方和越小，即回归模型的拟合效果越差。

SSR 即回归平方和，是因变量的预测值与因变量的均值之差的平方和，即 $SSR = \sum_{i=1}^{n}(y_i - \bar{y})^2$ ，其中 y_i 为第 i 个观测值，$\bar{y}$ 为因变量的均值。下面将简要介绍 SSR 的计算方法。

```
# 消除一次项
examnull.fit <- lm(Exam ~ 1, data = course.df)
summary(examnull.fit)
# 对比之前的 Summary
summary(examtest.fit)
```

Skip to main content

```
Call:
lm(formula = Exam ~ 1, data = course.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-41.877 -12.877  -1.377  15.623  40.123 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   52.877      1.546   34.21   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.68 on 145 degrees of freedom
```

```
Call:
lm(formula = Exam ~ Test, data = course.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-39.980  -6.471   0.826   8.575  33.242 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   9.0845     3.2204   2.821  0.00547 ** 
Test          3.7859     0.2647  14.301  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.05 on 144 degrees of freedom
Multiple R-squared:  0.5868,	Adjusted R-squared:  0.5839 
F-statistic: 204.5 on 1 and 144 DF,  p-value: < 2.2e-16
```

此时我们可以得到 SS（Null）的值 18.68，以及 SS（Test）的值 12.05。

R 方的值即 1 - SS（Null）/SS（Test）的值，即 0.5868。

置信区间：$[a_i - 2SE(a_i), a_i + 2SE(a_i)]$，即 $[a_i - 2\sqrt{Var(a_i)}, a_i + 2\sqrt{Var(a_i)}]$，其中 $Var(a_i)$ 为 a_i 的方差。

2.2.3. 每一个拟合值的 T 检验

> 知道看什么，什么意思，怎么看

Skip to main content

```r
summary(examtest.fit)
```

```

Call:
lm(formula = Exam ~ Test, data = course.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-39.980  -6.471   0.826   8.575  33.242 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   9.0845     3.2204   2.821  0.00547 ** 
Test          3.7859     0.2647  14.301  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.05 on 144 degrees of freedom
Multiple R-squared:  0.5868,	Adjusted R-squared:  0.5839 
F-statistic: 204.5 on 1 and 144 DF,  p-value: < 2.2e-16
```

可以看出 Test 行的 Pr（P-value）的值小于 2.2x10^-16，远小于 0.05，故拒绝原假设，即拟合值的系（旁边的3颗*也表示可信度极高，即该斜率的线性拟合极好）

- 零假设 H_0：Test 和 Exam 之间的线性关系系数为 0（没有线性关系），即 即 a_i 的系数为 0
- 备择假设 H_1：Test 和 Exam 之间的线性关系系数不为 0（有线性关系），即 即 a_i 的系数不为 0

我们对于斜率的置信程度，是由标准误差决定的，即 $SE(a_i)$，即 $SE(a_i)=\sqrt{\frac{SSE}{n-2}}$，其中 SSE 为残差平方和，即 $SSE=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$，其中 $\hat{y}_i$ 为第 i 个观测值的预测值，即 $\hat{y}_i=a_i+b_ix_i$，x_i 为第 i 个观测值的自变量值。此处的 $se(a)$ 为 0.2647。于是我们有：

$$\frac{3.7859-0}{0.2647}=14.34$$

此结果表示偏离此结果的标准差，这个数字越大，代表我们对于斜率的置信程度越高。

2.3. 利用分析结果做预测

2.3.1. 拟合值的置信区间

Skip to main content

```r
confint(examtest.fit)
# Intercept 即截距，Test 即斜率
# 也可以自己修改置信水平
confint(examtest.fit, level = 0.99)
```

A matrix: 2 × 2 of type dbl

	2.5 %	**97.5 %**
(Intercept)	2.719020	15.449907
Test	3.262659	4.309189

A matrix: 2 × 2 of type dbl

	0.5 %	**99.5 %**
(Intercept)	0.6778171	17.491110
Test	3.0948635	4.476984

2.3.2. 预测

1. 准确预测值
2. 预测的均值范围
3. 预测每一个个体的取值范围

区间估计和点估计的区别：

- 区间估计：给出一个区间，表示参数的可能取值范围
- 点估计：给出一个点，表示参数的可能取值

```r
# 区间估计
preds.df <- data.frame(Test = seq(0, 20, by = 10))
predict(examtest.fit, newdata = preds.df, interval = "confidence")
# 点估计
predict(examtest.fit, newdata = preds.df, interval = "prediction")
```

Skip to main content

A matrix: 3 × 3 of type dbl

	fit	lwr	upr
1	9.084463	2.71902	15.44991
2	46.943703	44.80912	49.07828
3	84.802942	79.97021	89.63568

A matrix: 3 × 3 of type dbl

	fit	lwr	upr
1	9.084463	-15.56475	33.73368
2	46.943703	23.03510	70.85231
3	84.802942	60.50438	109.10151

其中：

- 区间估计表格的 [2,2:3] 表示所有半期考试10分，期末考试的分数的均值的范围
- 区间估计表格的 [2,2:3] 表示所有半期考试10分个体的分数的范围，落在这个范围即为正常值

2.4. 总结

遇到此类问题，通用思路（适用于分析x和y两个未知数的某种关系）：

- 绘制数据散点图并简要查看自变量与因变量之间是哪种关系（如果有关系），最好是能够通过工具分析（也可能会有一份研究意图的声明可以被指导）。提出适当的研究方式。在上边的例子中，我们就决定采用了线性模型：

$$y = \beta_0 + \beta_1 x_i + \varepsilon_i, \varepsilon_i \sim N(0, \sigma^2)(where \beta_1 > 0)$$

- 使用 `lm` 函数进行模型拟合。
- 检查我们提出的假设进行合适方式的验证。
    - Independence OK? (how were the data collected?)
    - EOV Okay? Using `plot(examtest.fit, which = 1)`.
    - Normality Okay? Using `normcheck`.

  If these are okay then go to next step

Skip to main content

- 尝试适时删除任何不重要的解释变量（后面会讲）。如果能删除，请检查新的研究方式。
- 确保个别要点不会产生过分的不适当的影响，并尝试删除/纠正它们。Using `cooks20x`.
- 做出结论/预测，讨论极限，并回答相关的研究问题。

注意：在上述步骤中，在对当前步骤满意之前，切记不要匆忙进行下一步。

3. The null model

本课程前置需要装的包：

```
require(s20x)
require(bootstrap)
```

▶ Show code cell output

3.1. Revisiting the null model 回顾零模型

本节同样以 Stats20x 的学生考试成绩为例：

```
Stats20x.df <- read.table("../data/STATS20x.txt", header = TRUE, sep = "\t")
```

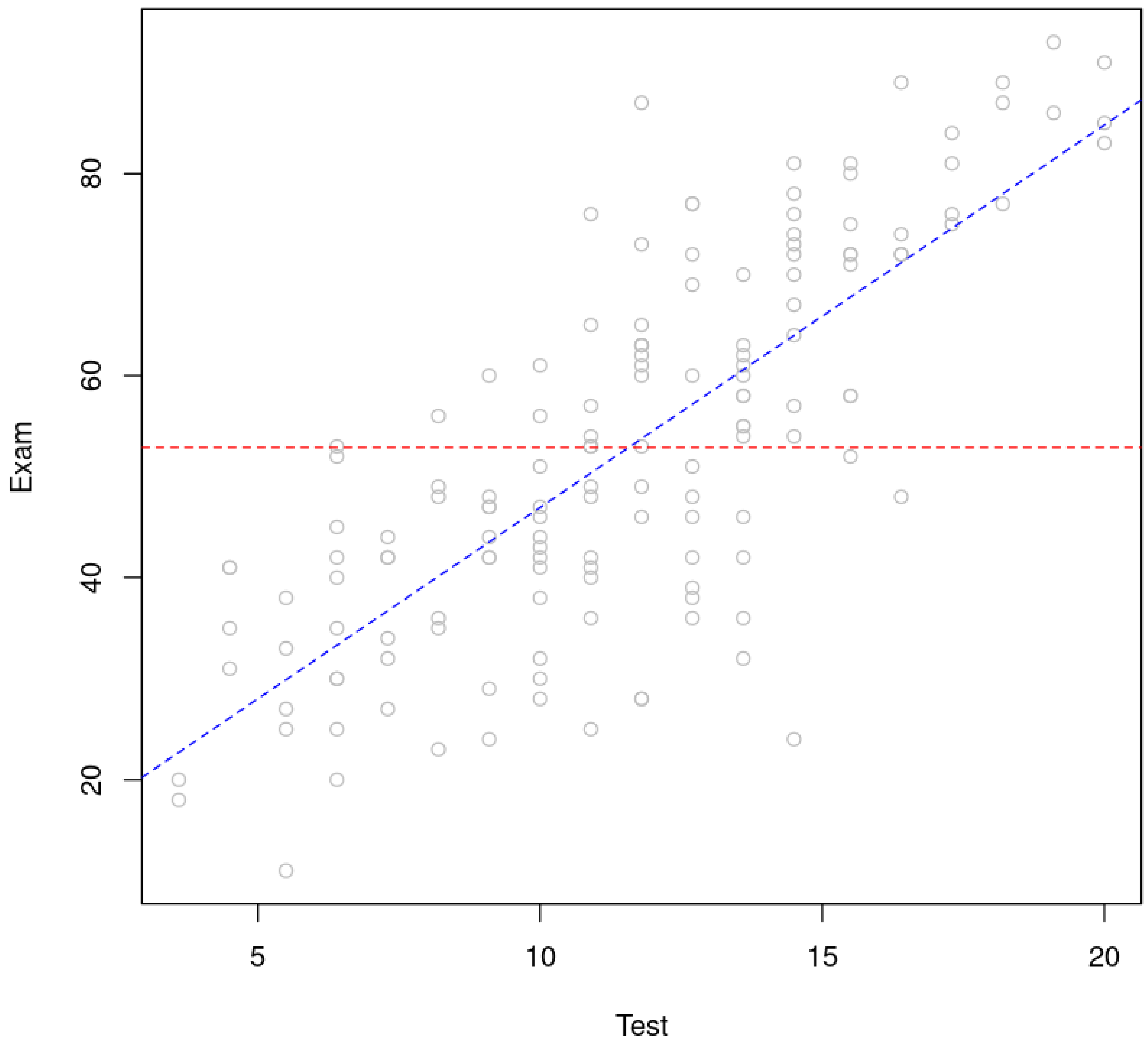
零模型就是把线性模型中的斜率去掉，或斜率指定常数，从而排除其影响单独分析截距。本节将重点讲述零模型的最大作用：T检验。

> 一文详解t检验 - 知乎
>
> t检验（t test）又称学生t检验（Student t-test）可以说是统计推断中非常常见的一种检验方法，用于统计量服从正态分布，但方差未知的情况。
>
> t检验的前提是要求样本服从正态分布或近似正态分布，不然可以利用一些变换（取对数、开根号、倒数等等）试图将其转化为服从正态分布是数据，如若还是不满足正态分布，只能利用非参数检验方法。不过当样本量大于30的时候，可以认为数据近似正态分布。
>
> t检验最常见的四个用途：
>
> - 单样本均值检验（One-sample t-test）用于检验“总体方差未知、正态数据或近似正态的”单样本

> - 两独立样本均值检验（Independent two-sample t-test）用于检验两对 “独立的，正态数据或近似正态的” 样本的均值是否相等，这里可根据总体方差是否相等分类讨论。
> - 配对样本均值检验（Dependent t-test for paired samples）用于检验一对配对样本的均值的差，是否等于某一个值
> - 回归系数的显著性检验（t-test for regression coefficient significance）用于检验回归模型的解释变量，对被解释变量是否有显著影响

```
# 建立回归模型
examtest.fit <- lm(Exam ~ Test, data = Stats20x.df)
examtest.fit2 <- lm(Exam ~ 1, data = Stats20x.df)

# 绘图
plot(Exam ~ Test, data = Stats20x.df, col = "grey")
abline(examtest.fit, col = "blue", lty = 2)
abline(examtest.fit2, col = "red", lty = 2)
```

Skip to main content

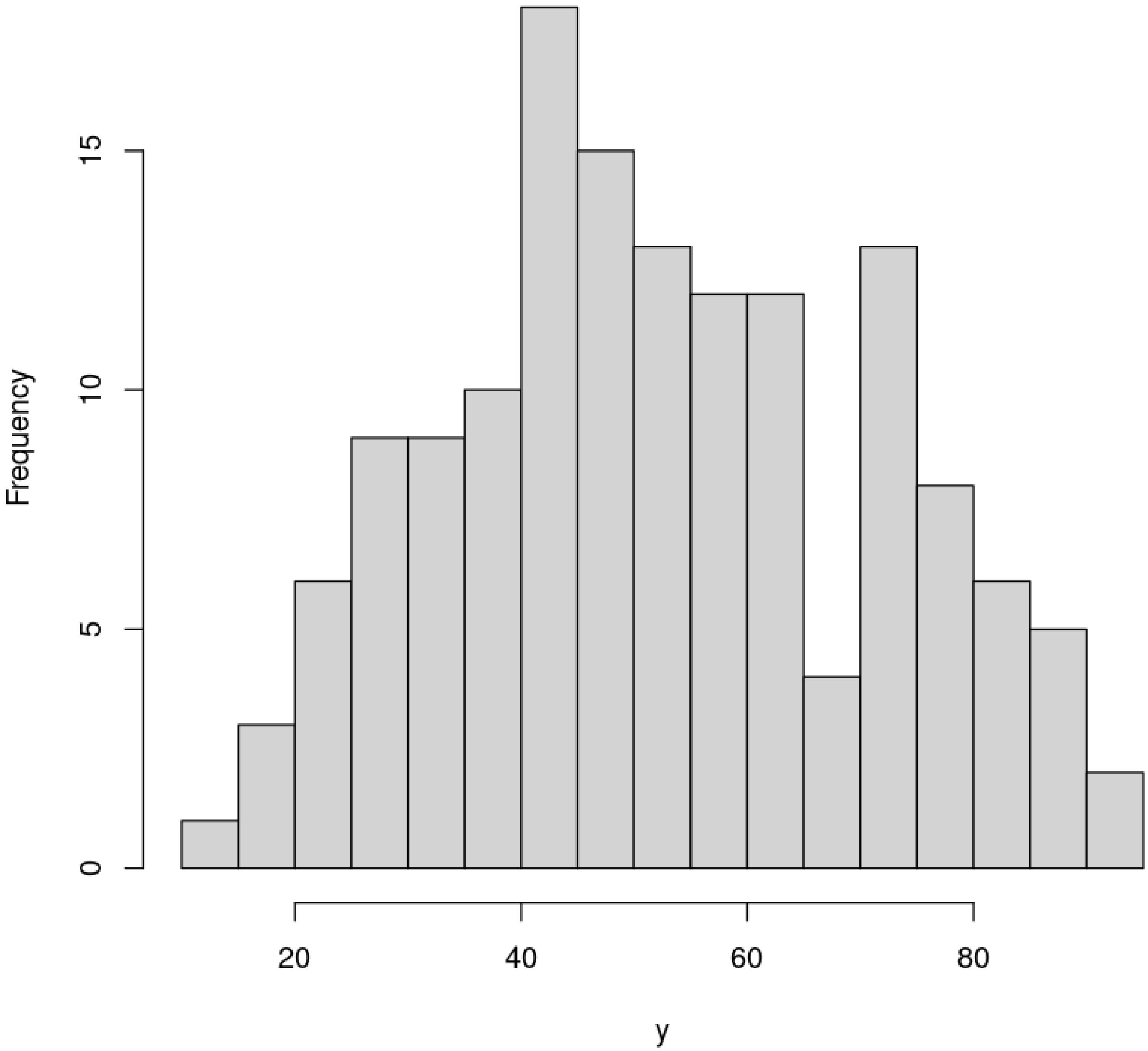


推断总体均值：

To save some typing we’ll let y be the vector Stats20x.df$Exam of exam scores.

```
y <- Stats20x.df$Exam
hist(y, breaks = 20, main = "") # Use main to suppress plot title
```

Skip to main content



继续使用零模型做线性回归，使其更关注于y值的置信关系与p检验。

```
null.fit <- lm(y ~ 1)
# Only give coefficients from summary 将系数板块单独提取出做展示
coef(summary(null.fit))
# 获得该零模型的对应置信区间
confint(null.fit)
```

Skip to main content

A matrix: 1 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	52.87671	1.545802	34.20666	2.632011e-71

A matrix: 1 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	49.8215	55.93193

Conclusion:

- The near zero $Pr(> |t|)$ p-value totally rejects(拒绝) the null hypothesis(零假设) that $H0 : \mu \equiv \beta 0 = 0.$
- The 95% confidence interval(置信区间) for μ is 49.82 to 55.93.

3.2. Revisiting the t-test

$$T = \frac{\bar{y} - \mu}{\frac{s}{\sqrt{n}}} \sim t_{n-1}$$

其中 $\bar{y}$ 为样本均值，s 为样本标准差。

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

```r
n <- length(y) # 146 students
tstat <- (mean(y) - 0) / (sd(y) / sqrt(n))
tstat
```

34.2066579217089

Skip to main content

```
## t-multiplier
tmult <- qt(1 - .05 / 2, df = n - 1)
## We want the upper 97.5% (or 1-.05/2) bound of the CI
## NOTE: mean = sample mean; sd = standard deviation; sqrt = square root
mean(y) - tmult * sd(y) / sqrt(n)

## Upper bound of CI 置信区间上限
mean(y) + tmult * sd(y) / sqrt(n)
## Or if we want both the lower and upper bounds of the CI in one statement
## 置信区间下限
mean(y) + c(-1, 1) * tmult * sd(y) / sqrt(n)
```

49.8214976403875

55.9319270171467

49.8214976403875 · 55.9319270171467

零模型就是单样本T检验。

手动随机抽样检验我们的结果：

```
## Resampling the exam marks, N times with replacement:
N <- 10000 # The number of bootstrap resamples we want
# The new sample means are stored in ybar
ybar <- rep(NA, N) ## A vector of length N to store our resampled means

## A loop - allows us to do something N (10,000) times
for (i in 1:N) {
    ## Take the average of this sample (below) from a sample of size n = 146 from y - w
    ybar[i] <- mean(sample(y, n, replace = T))
}
mean(ybar)
```

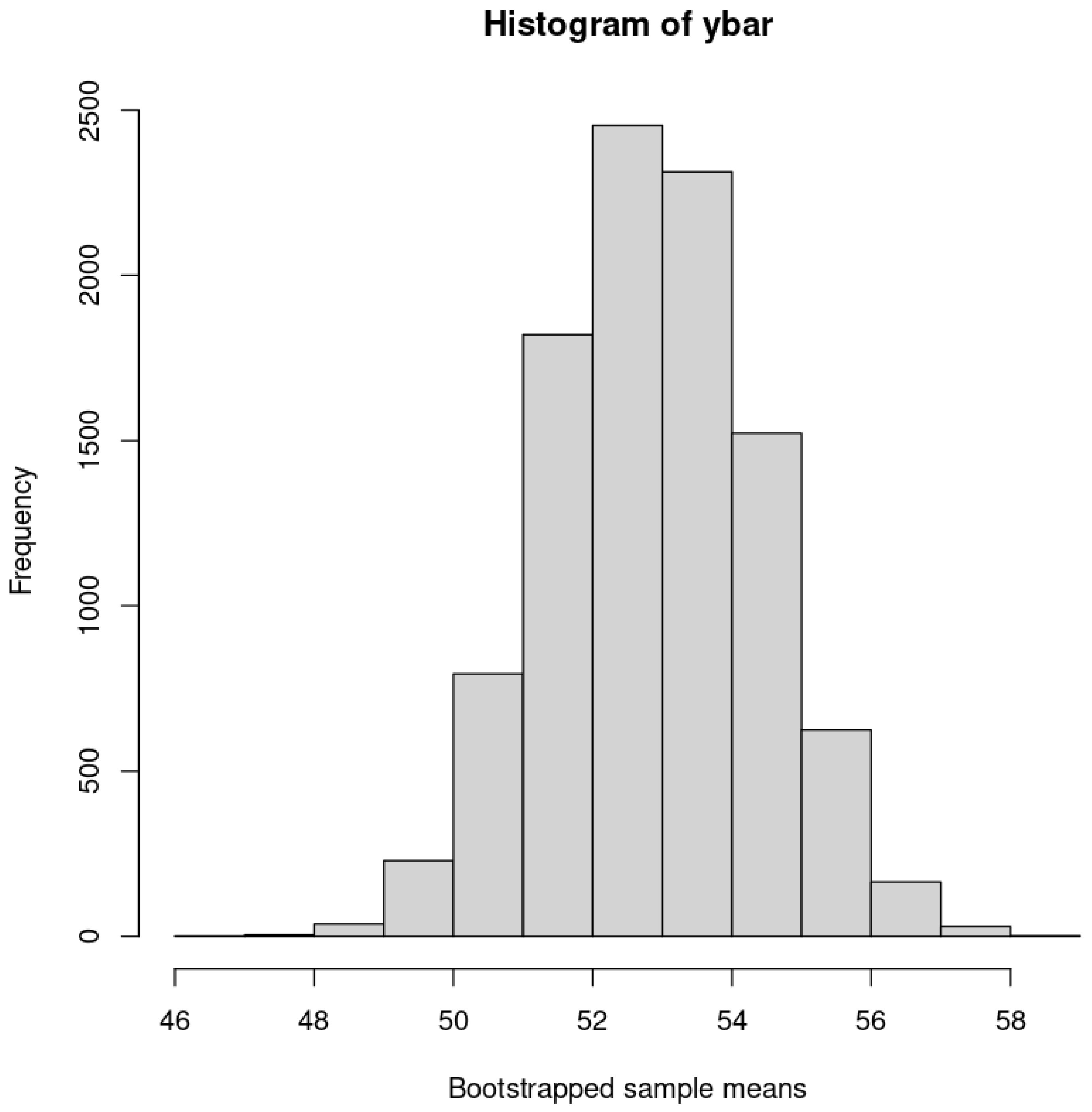
52.8874787671233

```
library(bootstrap)
ybar <- bootstrap(Stats20x.df$Exam, 10000, mean)$thetastar
mean(ybar)
```

52.8795863013699

```
## Histogram of these 10,000 bootstrap means
hist(ybar, xlab = "Bootstrapped sample means")
```

Skip to main content



3.3. The paired t-test

For a meaningful comparison, We will need to make them have the same scale, so we multiply the test mark by 5 so that it is also out of 100.

Skip to main content

```
Stats20x.df$Test2 <- 5 * Stats20x.df$Test
## Check that it worked
Stats20x.df[1:3, c("Exam", "Test", "Test2")]
```

A data.frame: 3 × 3

	Exam	Test	Test2
	<int>	<dbl>	<dbl>
1	42	9.1	45.5
2	58	13.6	68.0
3	81	14.5	72.5

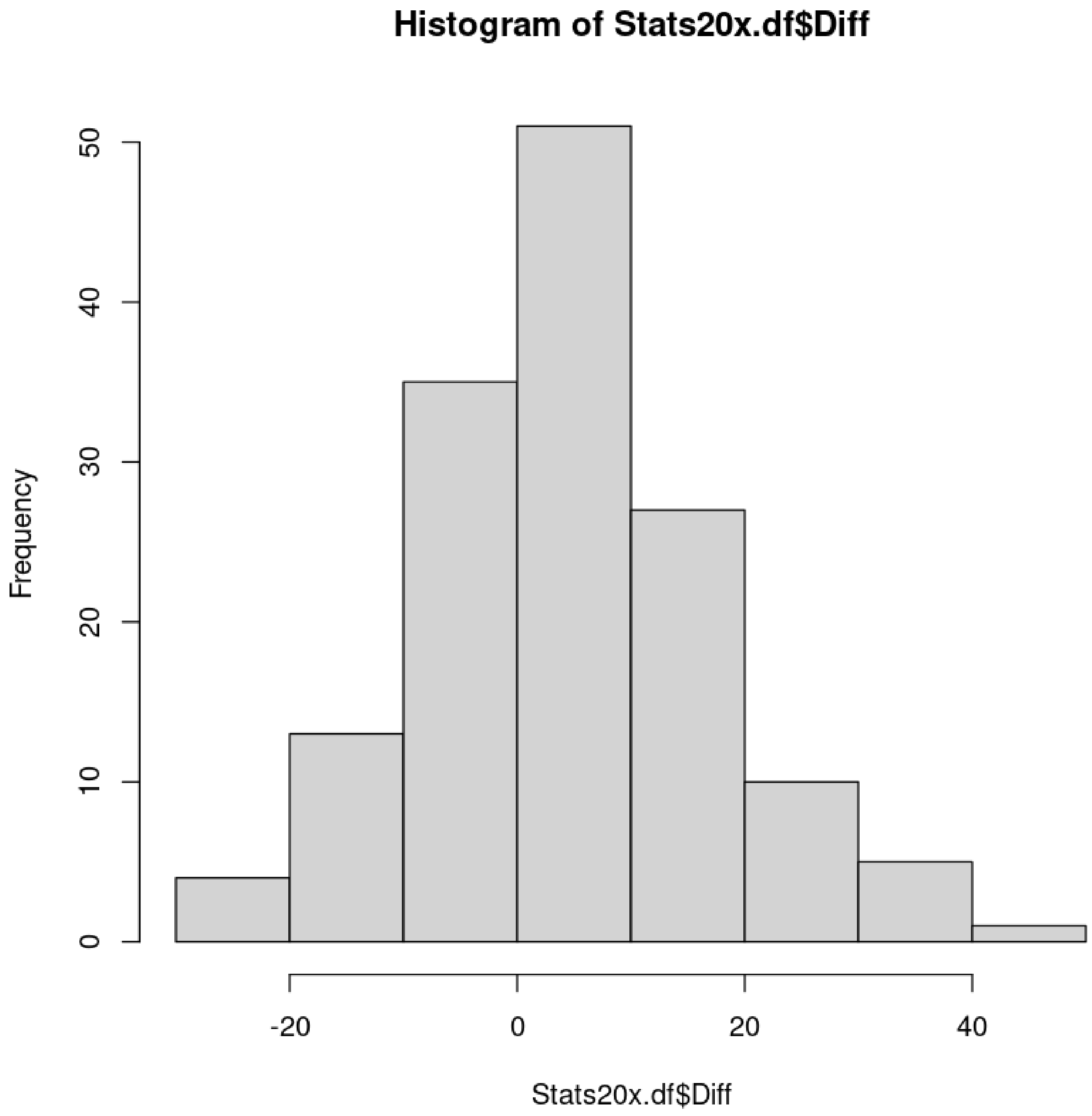
```
Stats20x.df$Diff <- Stats20x.df$Test2 - Stats20x.df$Exam
## Check the first 5 measurements
Stats20x.df[1:5, c("Test2", "Exam", "Diff")]
```

A data.frame: 5 × 3

	Test2	Exam	Diff
	<dbl>	<int>	<dbl>
1	45.5	42	3.5
2	68.0	58	10.0
3	72.5	81	-8.5
4	95.5	86	9.5
5	41.0	35	6.0

```
hist(Stats20x.df$Diff)
```

Skip to main content



4. Fitting curves with the linear model

本节需要的包：

```r
require(s20x)
```

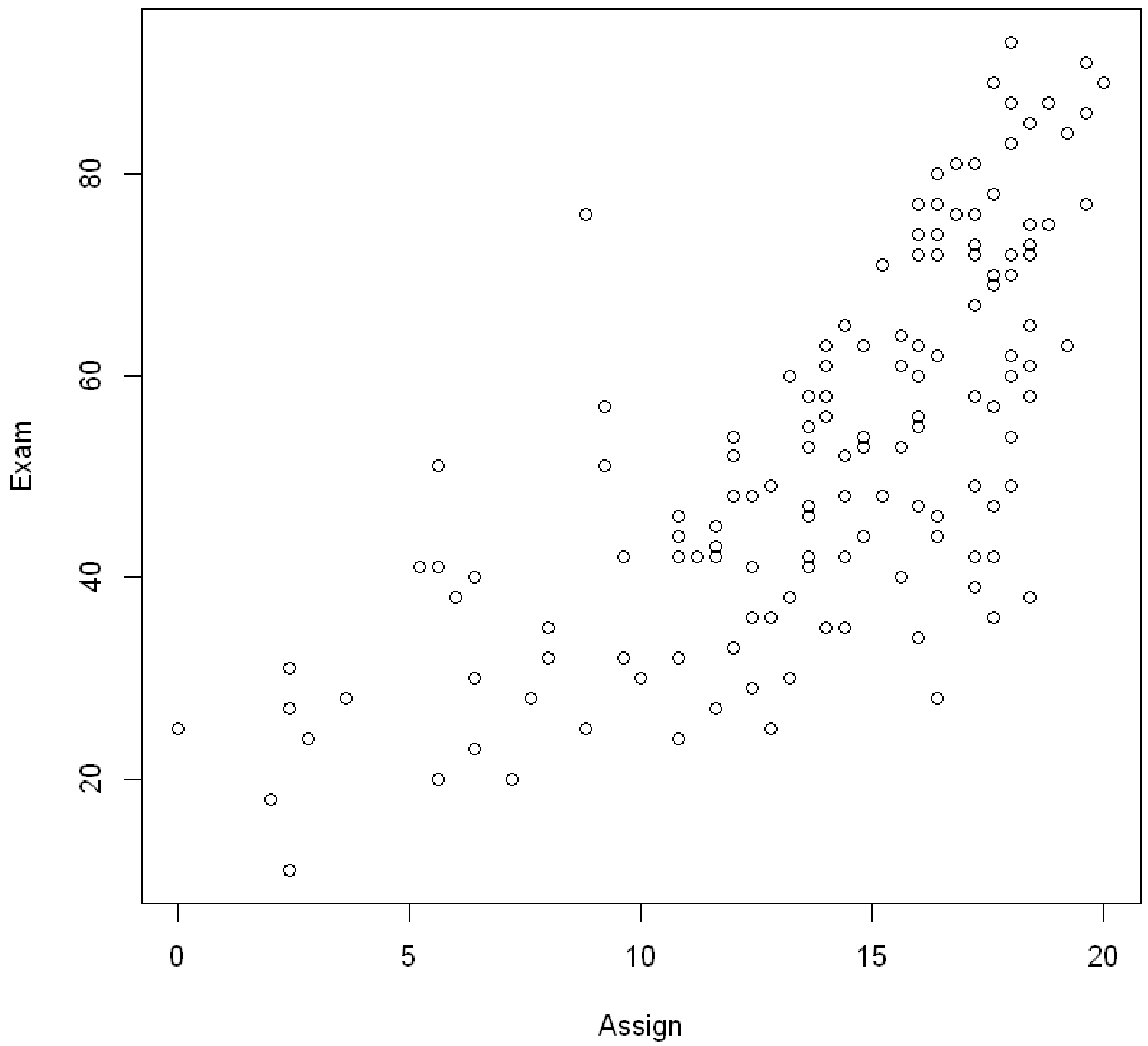
Skip to main content

▶ Show code cell output

4.1. Identifying a curved relationship 初步探究曲线关系

```
## Load the s20x library into our R session
library(s20x)
## Importing data into R
Stats20x.df <- read.table("../data/STATS20x.txt", header = T)
## Examine the data
plot(Exam ~ Assign, data = Stats20x.df)
```

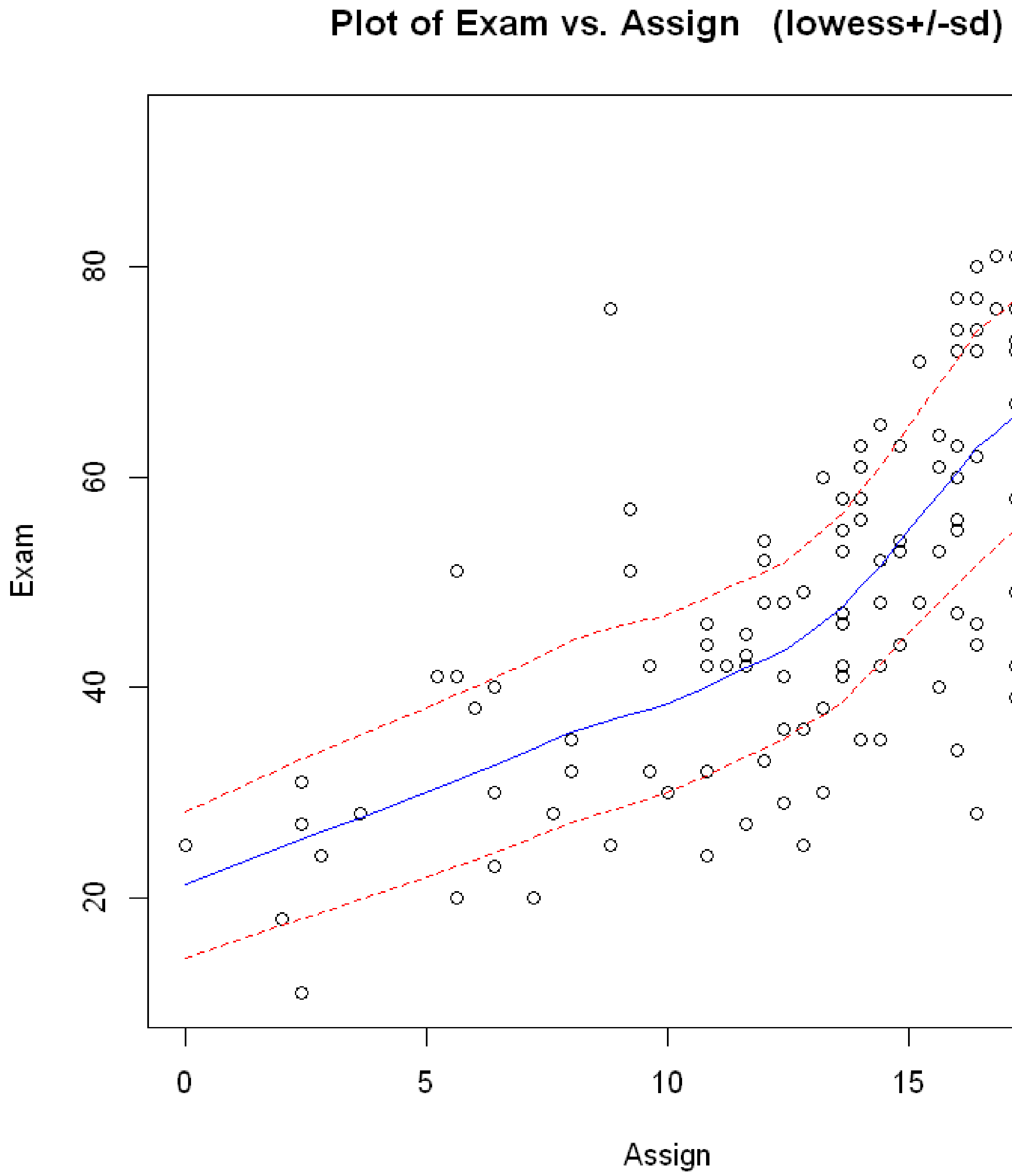
Skip to main content



Hmmm, not quite a straight line – could be some curvature. Maybe will paint a clearer picture. 不是一条很直的线--可能是一些曲率。也许会描绘出一幅更清晰的图景。

```
trendscatter(Exam ~ Assign, data = Stats20x.df)
```

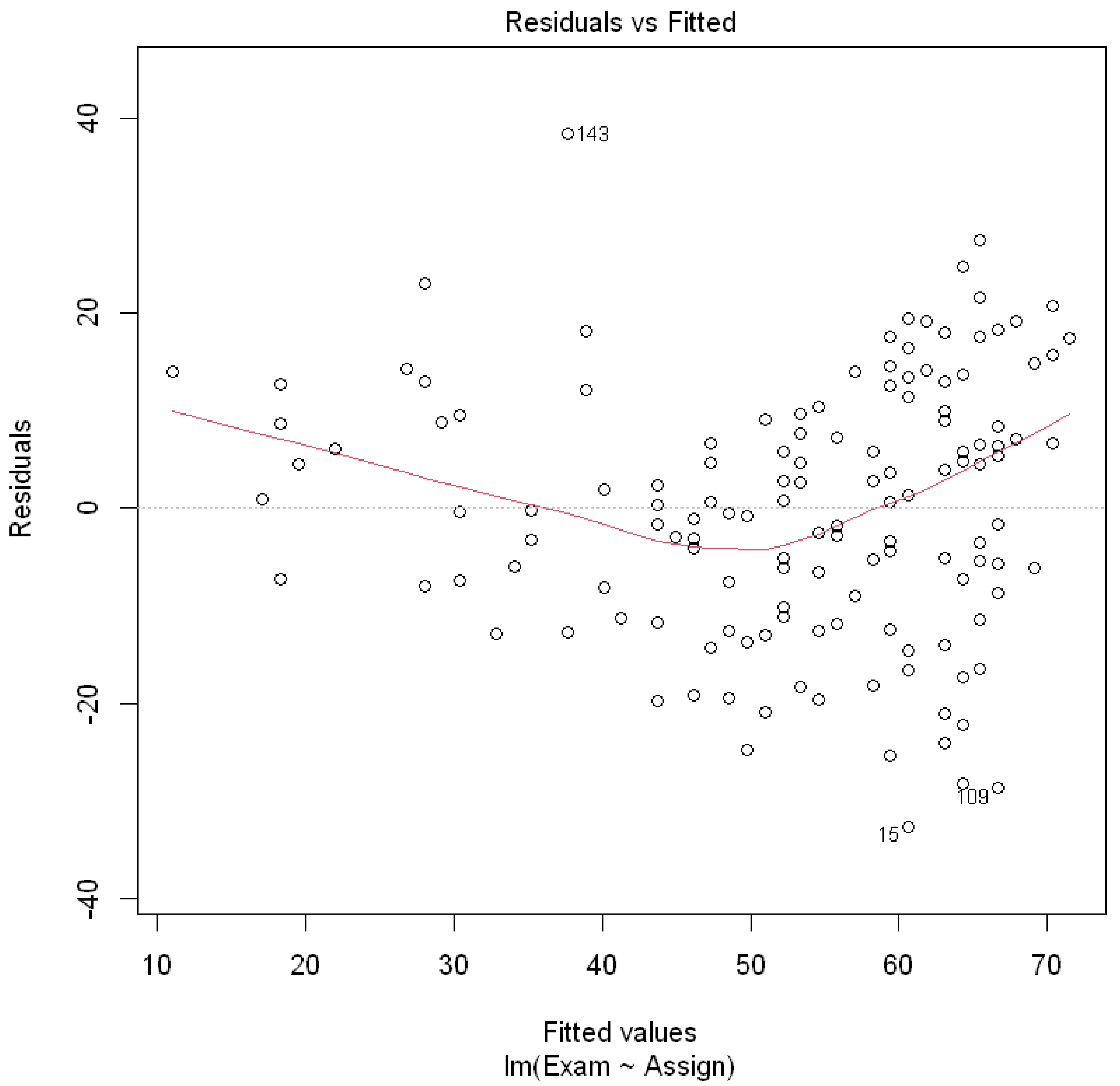
Skip to main content



Let’s fit a simple linear model to these data and see if it works out or not.

```
examassign.fit <- lm(Exam ~ Assign, data = Stats20x.df)
plot(examassign.fit, which = 1)
```

Skip to main content



The assumption of identical distribution with expected value of 0 looks to be questionable here. There tend to be more negative residuals in the middle, but more positive residuals at the extremes of the fitted values. Potential solution – add a quadratic (squared term) for.

假设相同的分布与预期值 0 看起来可疑的。会有更多负面的残差在中间,但更积极的残差的极端值。潜在的解决方案应该是：添加一个二次项(平方项)。

Skip to main content

4.2. Fitting a quadratic model 拟合二次模型

The standard notation for a quadratic curve is:

$$y = ax^2 + bx + c$$

Here we will use different notation: $\beta_0 = c$, $\beta_1 = b$ and $\beta_2 = a$ and use the quadratic curve to describe the expected value of our dependent variable y. That is, we will use the following notation:

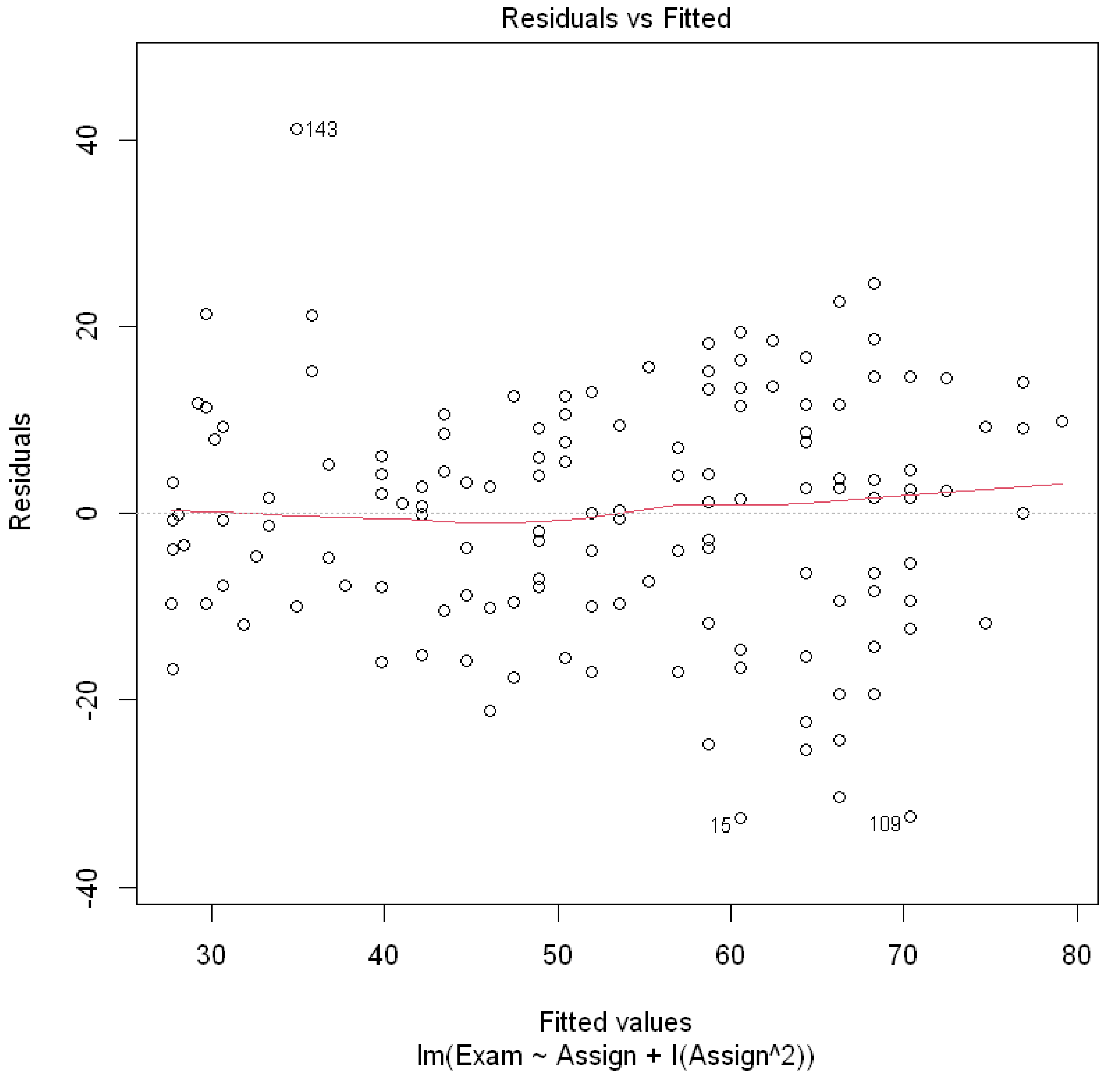
$$E[Y|x] = \beta_0 + \beta_1 x + \beta_2 x^2$$

> If $\beta_2 > 0$, then the quadratic has slope that increases with increasing x(斜率随着 x 增大而增大). If $\beta_2 < 0$, then the quadratic has slope that decreases with increasing x. If $\beta_2 = 0$, then the quadratic(该“二次曲线”) has a constant slope(倾斜直线的外观).

让我们回到之前的学生数据集。我们将使用一个新的变量 x^2 来拟合一个二次模型：

```
examassign.fit2 <- lm(Exam ~ Assign + I(Assign^2), data = Stats20x.df)
plot(examassign.fit2, which = 1)
```

Skip to main content



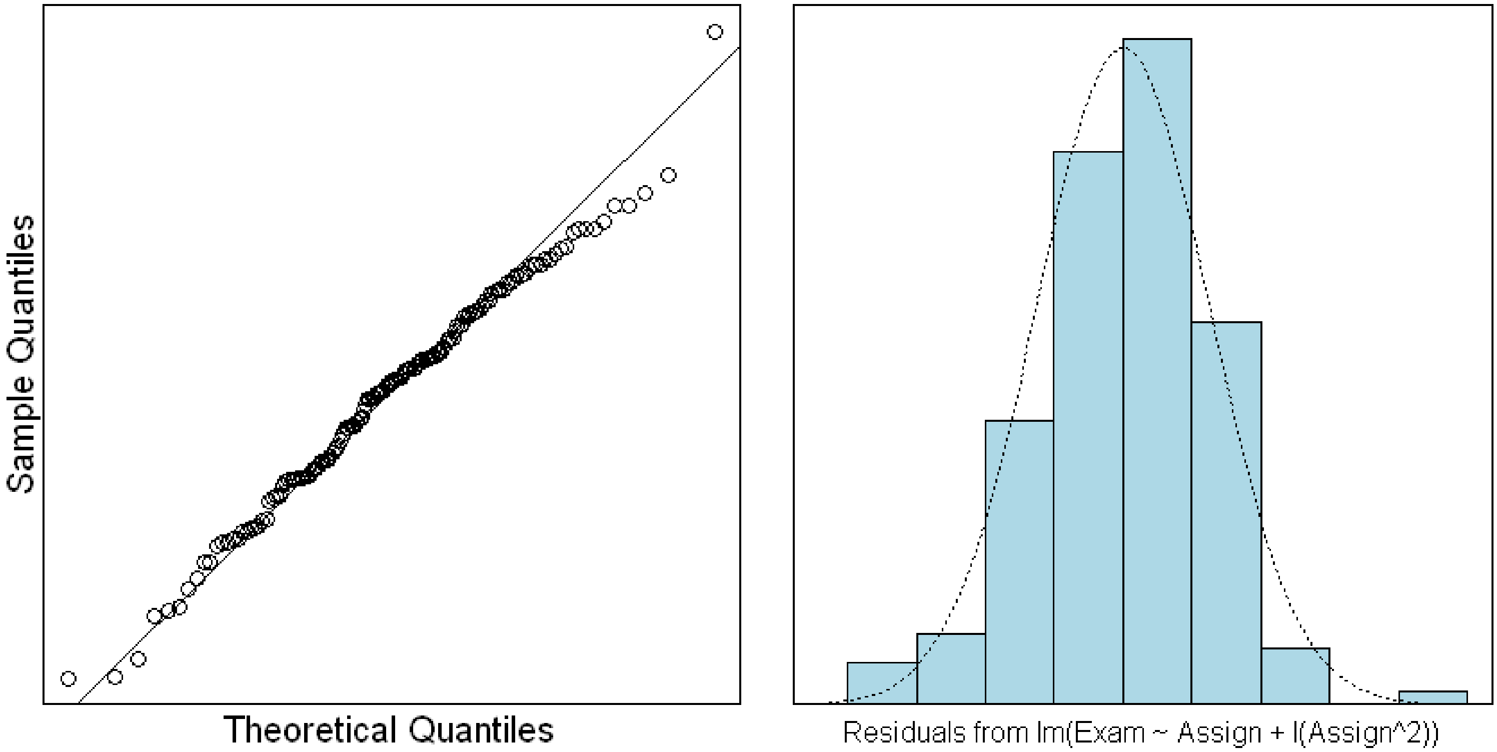
That is looking much better.

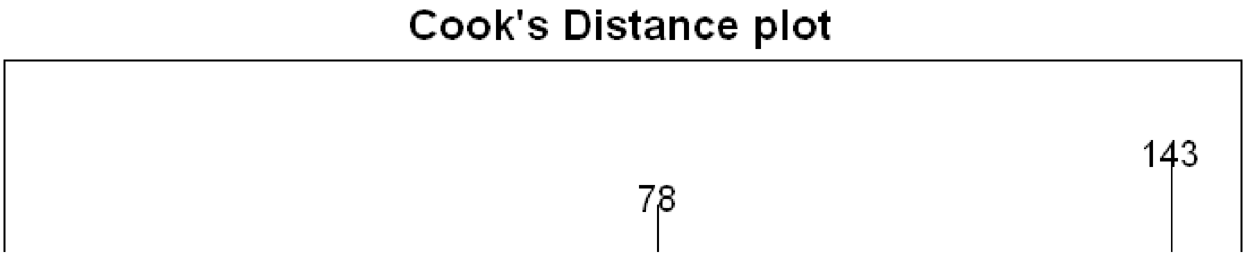
接下来我们会进行“三步走”中的后两步：

```
normcheck(examassign.fit2)
cooks20x(examassign.fit2)
```

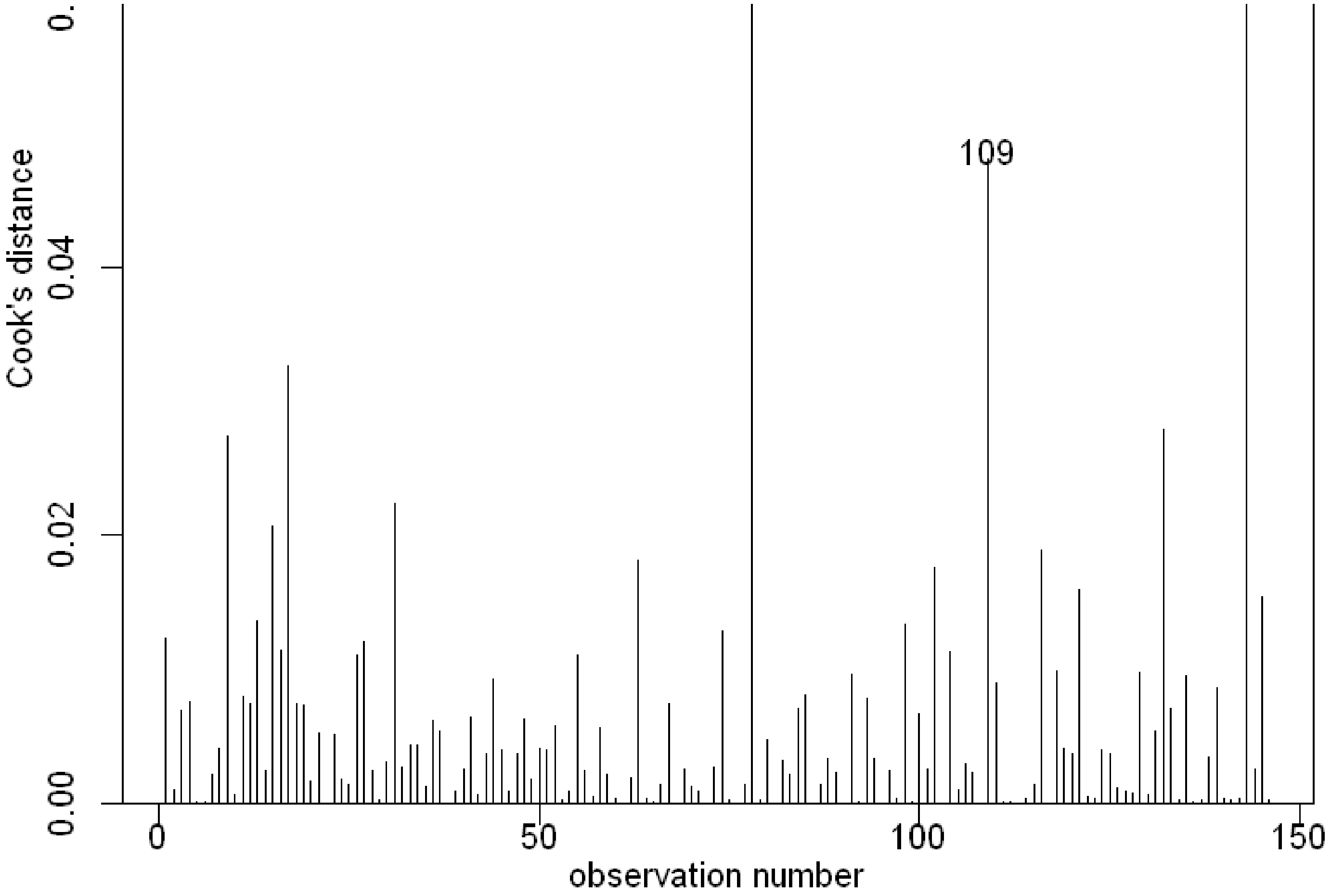
Skip to main content

Skip to main content





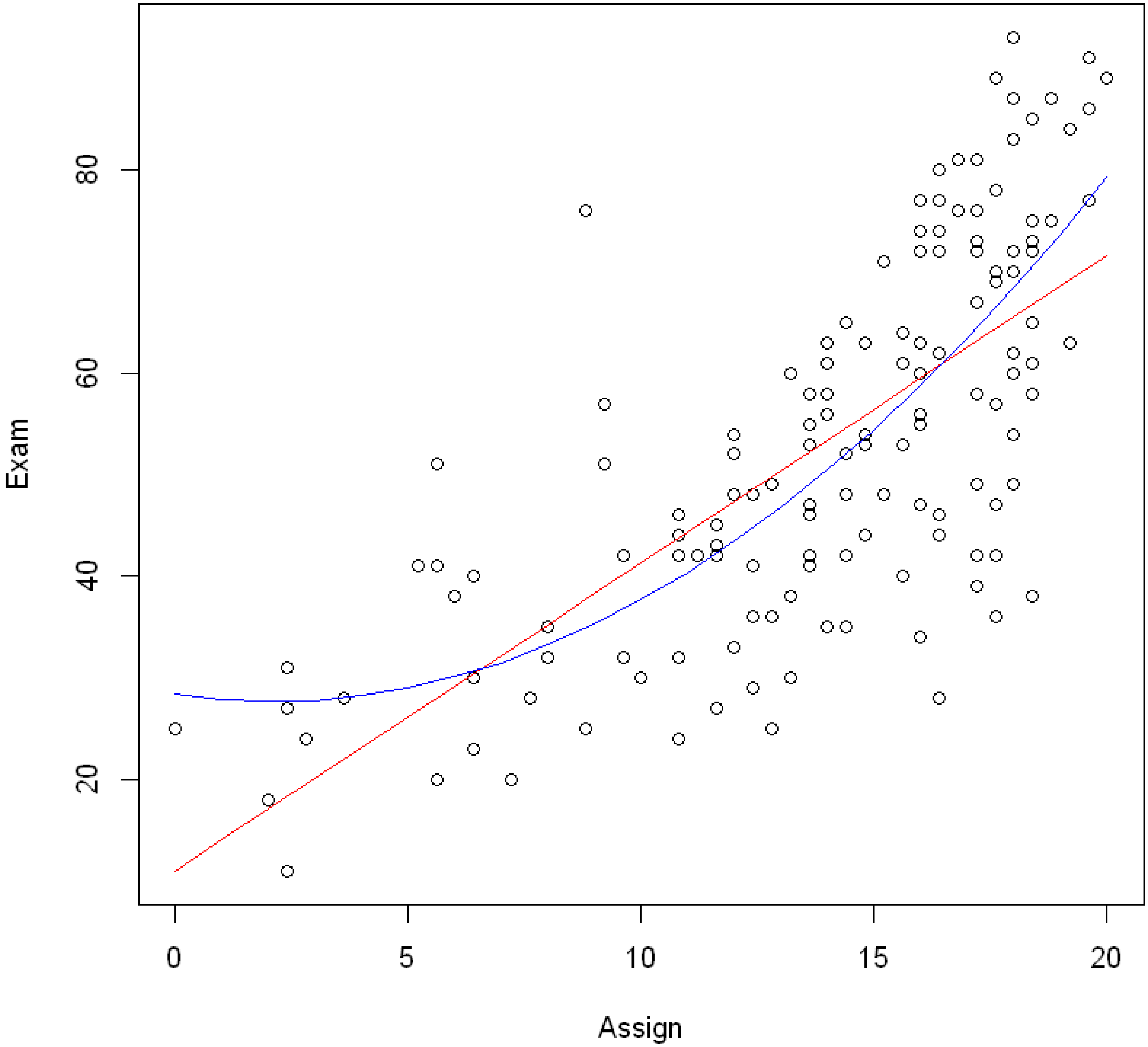
Skip to main content



符合正态分布、方差齐性。我们可以尝试对照一下原来的模型和我们的新模型：

```
plot(Exam ~ Assign, data = Stats20x.df)
x <- 0:20 # Assignment values at which to predict exam mark
## Plot model 1
lines(x, predict(examassign.fit, data.frame(Assign = x)), col = "red")
## Plot model 2
lines(x, predict(examassign.fit2, data.frame(Assign = x)), col = "blue")
```

Skip to main content



```
summary(examassign.fit2)
```

Skip to main content

```
Call:
lm(formula = Exam ~ Assign + I(Assign^2), data = Stats20x.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-32.541  -9.149   1.273   9.087  41.116 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 28.41396    5.99081   4.743 5.05e-06 ***
Assign      -0.68172    1.07242  -0.636 0.525999    
I(Assign^2)  0.16102    0.04545   3.542 0.000536 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.65 on 143 degrees of freedom
Multiple R-squared:  0.5477,	Adjusted R-squared:  0.5414 
F-statistic: 86.59 on 2 and 143 DF,  p-value: < 2.2e-16
```

Note that the coefficient $\beta_2 > 0$ associated with the term $I(Assign)^2$ indicates an increase that starts slowly and ‘accelerates’(加速) as Assign increases.

5. Linear models with a categorical (factor) explanatory variable

本节需要的包：

```
require(s20x)
```

▶ Show code cell output

5.1. Using categorical variables as explanatory variables by using indicator variables

使用指标变量将分类变量用作解释变量

Skip to main content

```
library(s20x)
## Importing data into R
Stats20x.df <- read.table("../data/STATS20x.txt", header = T)
## Change Attend from a character variable to a factor variable
Stats20x.df$Attend <- as.factor(Stats20x.df$Attend)
## Examine the data
Stats20x.df$Attend[1:20]
```

Yes · Yes · Yes · Yes · No · Yes · Yes · No · Yes · Yes · No · Yes · No · No · No · Yes · Yes · No · Yes · Yes

▶ **Levels**:

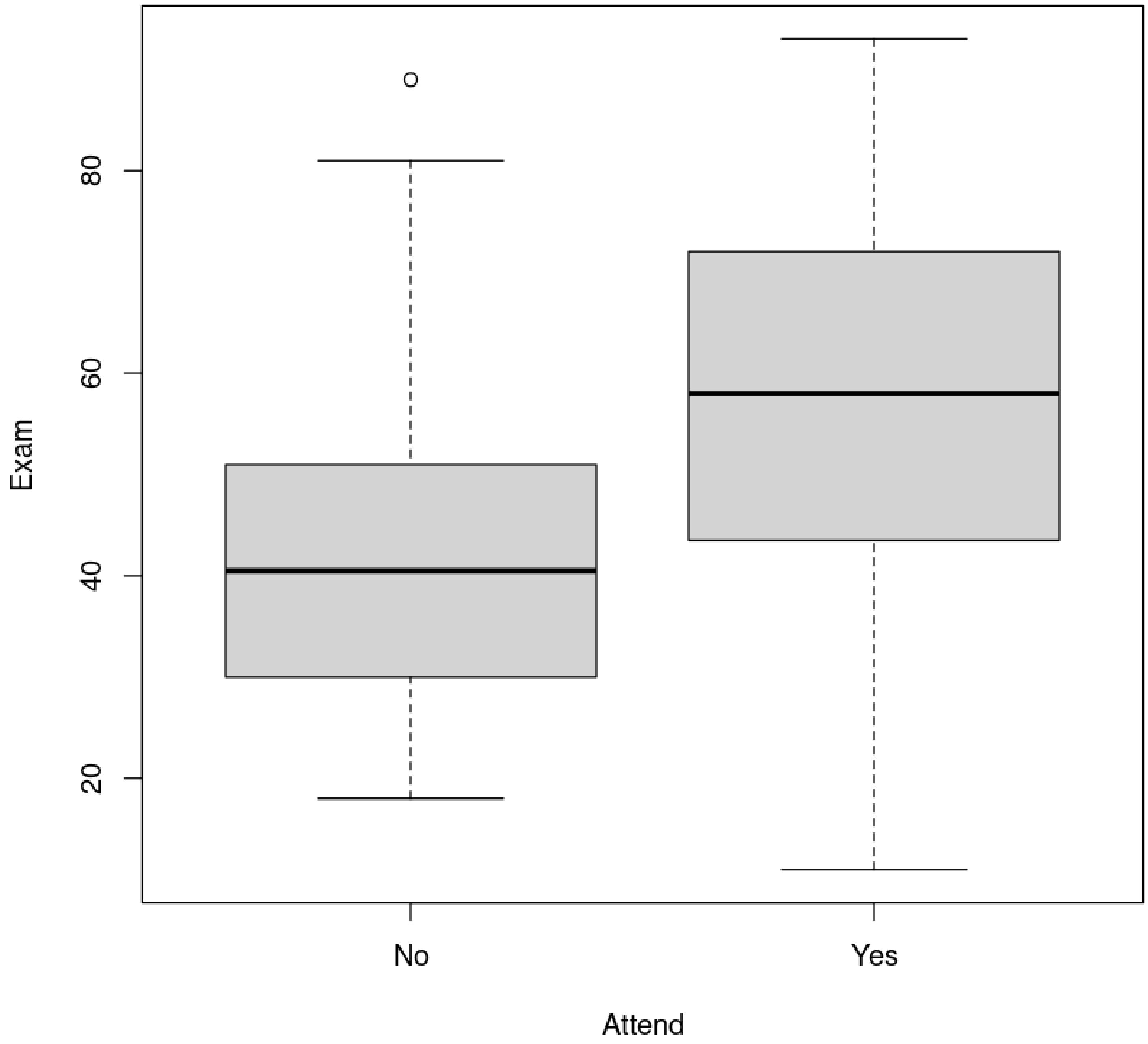
简要分析数据集，确保有你需要的可能的关系：

```
summaryStats(Stats20x.df$Exam, Stats20x.df$Attend)
plot(Exam ~ Attend, data = Stats20x.df)
```

Skip to main content

缺勤的确会让学生成绩变低，在数据分布上的确有一定的关系。

```
    Sample Size     Mean Median  Std Dev Midspread
No           46 42.21739   40.5 16.34206     20.50
Yes         100 57.78000   58.0 17.67757     28.25
```



缺勤的确会让学生成绩变低，在数据分布上的确有一定的关系。

为了在后面进行更好的分析，我们将缺勤的 Yes 和 No 转换为 1 和 0：

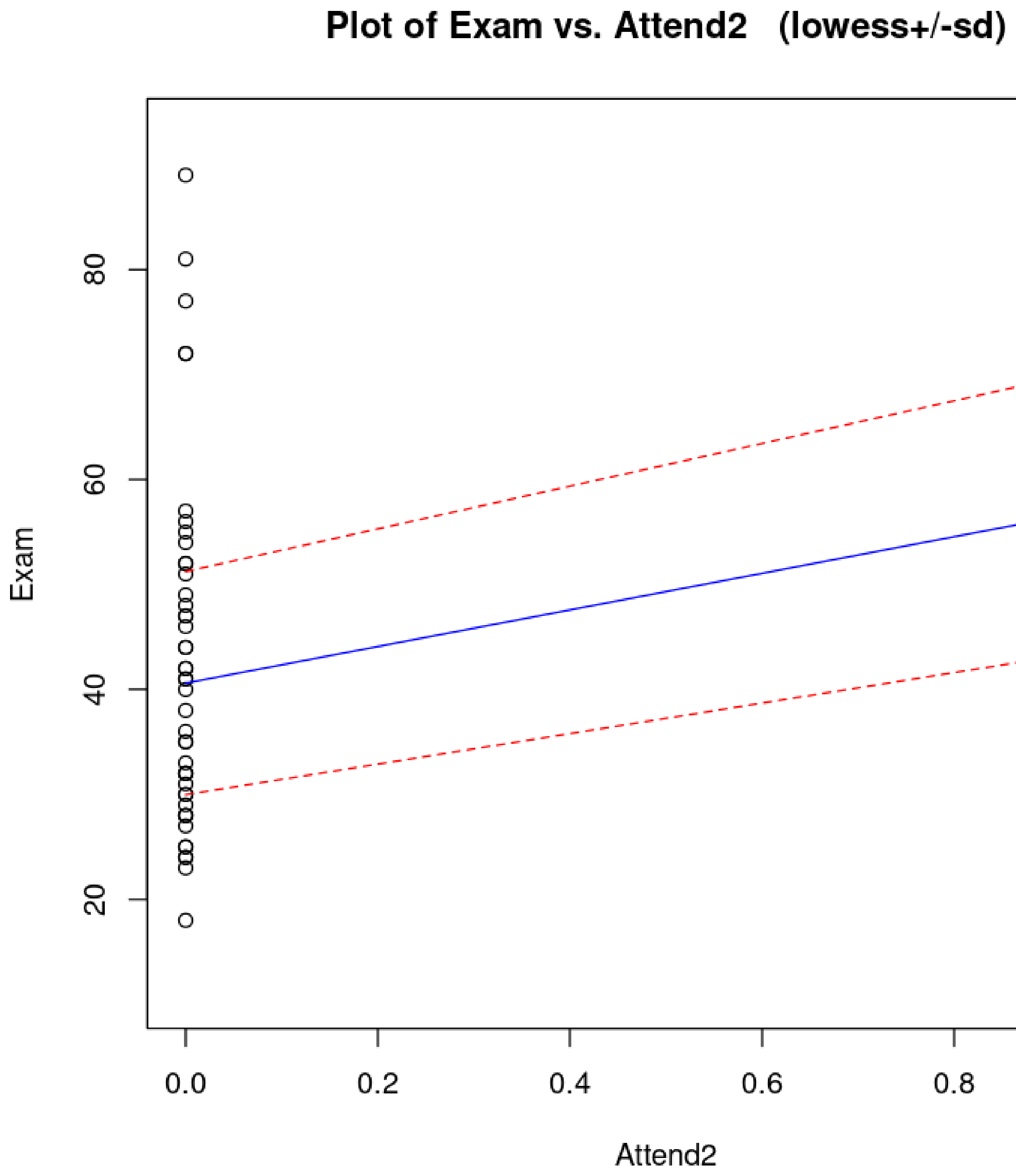
Skip to main content

```
# Make a new variable Attend2 which is 1 if Attend = "Yes" and 0 otherwise

# Note how we use two equal signs, ==, to test equality
Stats20x.df$Attend2 <- as.numeric(Stats20x.df$Attend == "Yes")
with(Stats20x.df, table(Attend, Attend2))
```

```
      Attend2
Attend   0   1
   No   46   0
   Yes   0 100
```

```
trendscatter(Exam ~ Attend2, data = Stats20x.df)
```

Skip to main content



The linear model for the expected value of is

$$E[Exam|Attend2] = \beta_0 + \beta_1 Attend2$$

其中 β_0 是截距，即所有缺勤的均值；β_1 是考试成绩和缺勤的关系，由缺勤和出勤的成绩关系共同决定。

Skip to main content

```
examattend2.fit <- lm(Exam ~ Attend2, data = Stats20x.df)
summary(examattend2.fit)
```

```

Call:
lm(formula = Exam ~ Attend2, data = Stats20x.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-46.780 -13.108  -0.217  12.642  46.783 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   42.217      2.547  16.578  < 2e-16 ***
Attend2       15.563      3.077   5.058 1.27e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.27 on 144 degrees of freedom
Multiple R-squared:  0.1508,	Adjusted R-squared:  0.145 
F-statistic: 25.58 on 1 and 144 DF,  p-value: 1.271e-06
```

上述拟合代表 x 为 Attend2（0 和 1） 时，y 的期望值，即考试成绩的期望值。

但注意事实上，直接使用 lm() 函数进行拟合，也能得出正确的结果，因为 lm() 函数会自动将分类变量转换为指标变量（AttendYes）：

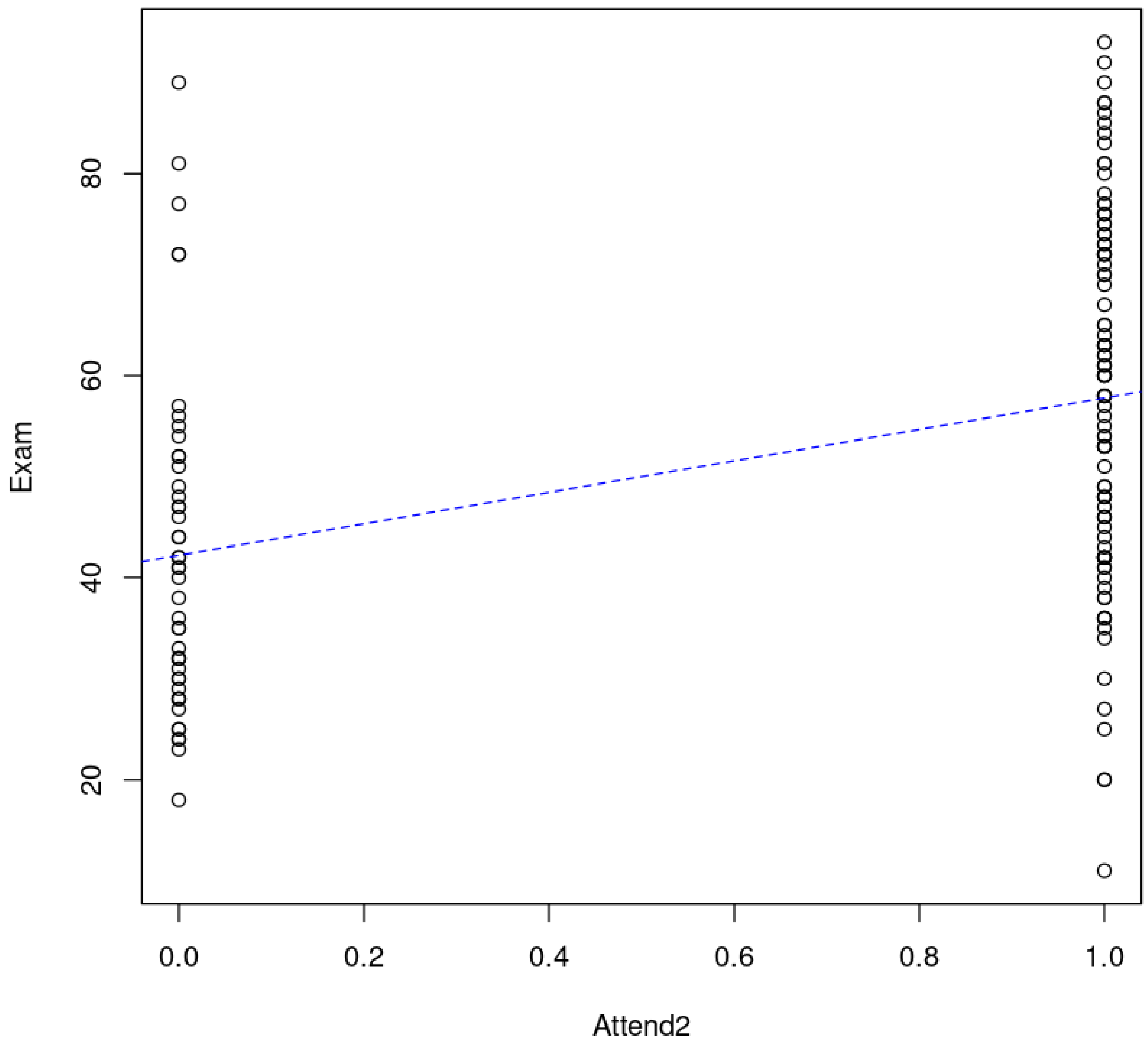
```
examattend.fit <- lm(Exam ~ Attend, data = Stats20x.df)
summary(examattend.fit)
```

```

Call:
lm(formula = Exam ~ Attend, data = Stats20x.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-46.780 -13.108  -0.217  12.642  46.783 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   42.217      2.547  16.578  < 2e-16 ***
AttendYes     15.563      3.077   5.058 1.27e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.27 on 144 degrees of freedom
Multiple R-squared:  0.1508,	Adjusted R-squared:  0.145 
```

Skip to main content

让我们将拟合模型可视化。在这里，我们将使用虚拟变量拟合我们的模型得到的"最佳"估计直线绘制出来。

```
plot(Exam ~ Attend2, data = Stats20x.df)
## Add the lm estimated line to this plot where a=intercept, b=slope
abline(coef(examattend.fit), lty = 2, col = "blue")
```

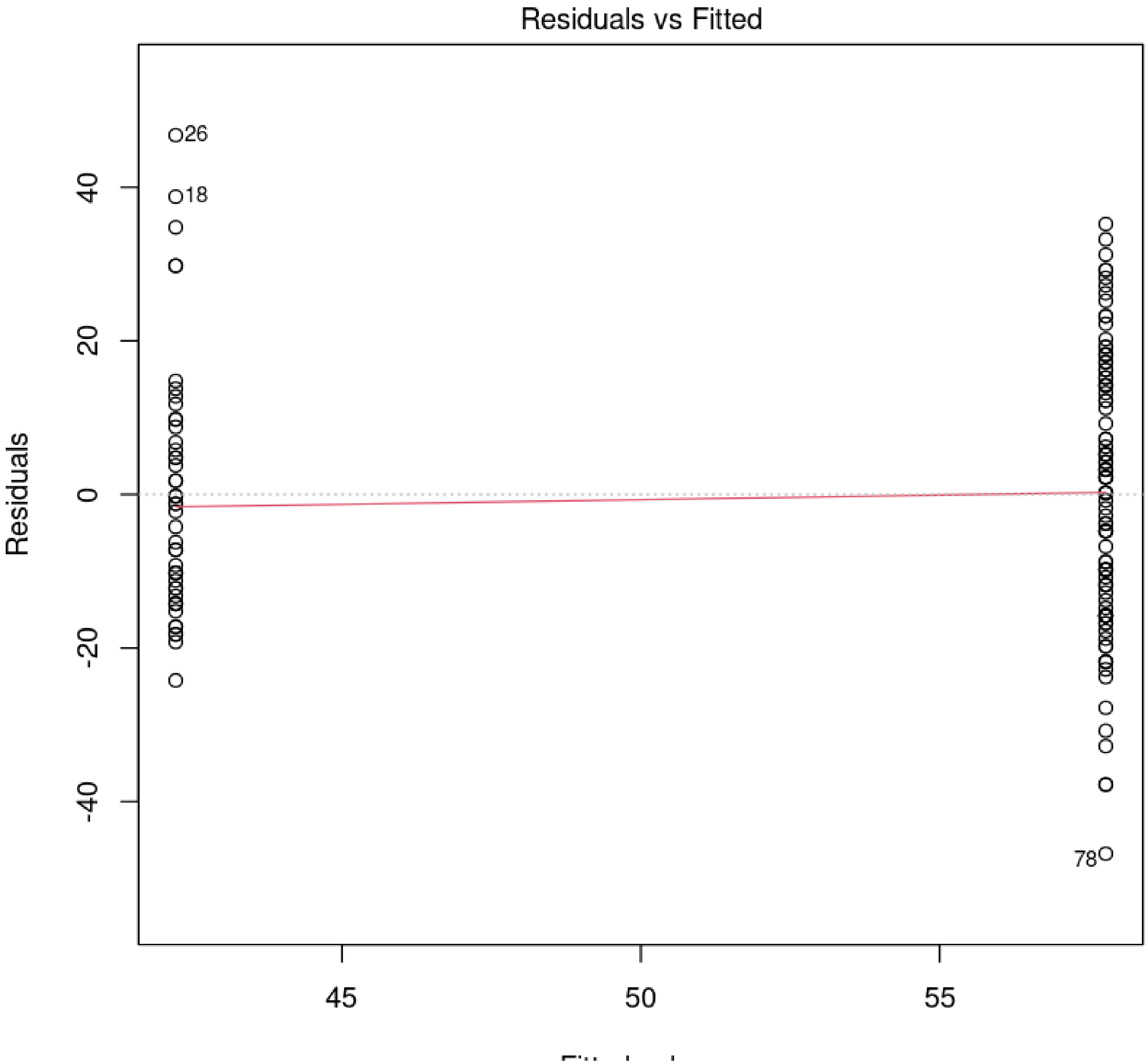


Skip to main content

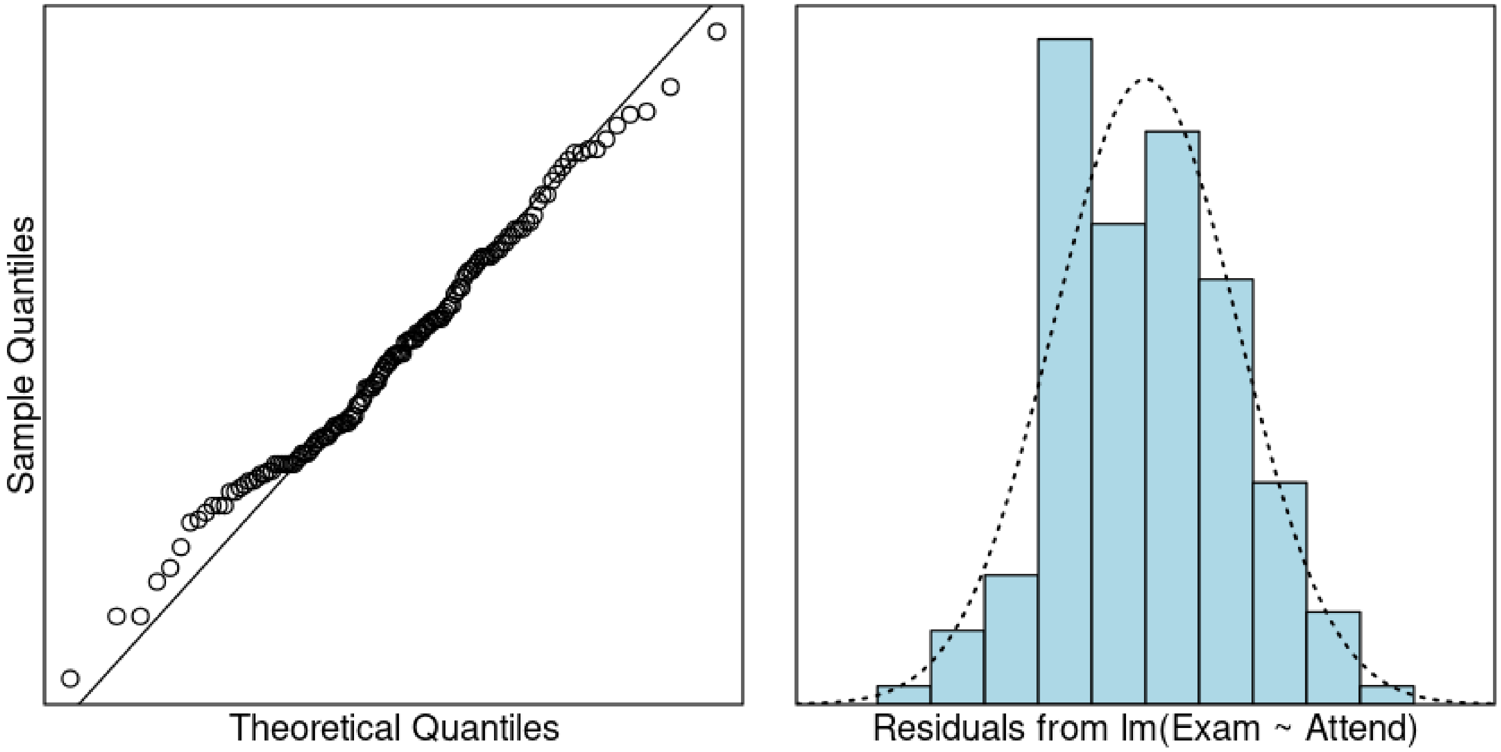
1. 残差均值接近于 0
2. 残差满足正态分布
3. 没有或排除了异常点

```
plot(examattend.fit, which = 1)
normcheck(examattend.fit)
cooks20x(examattend.fit)
```

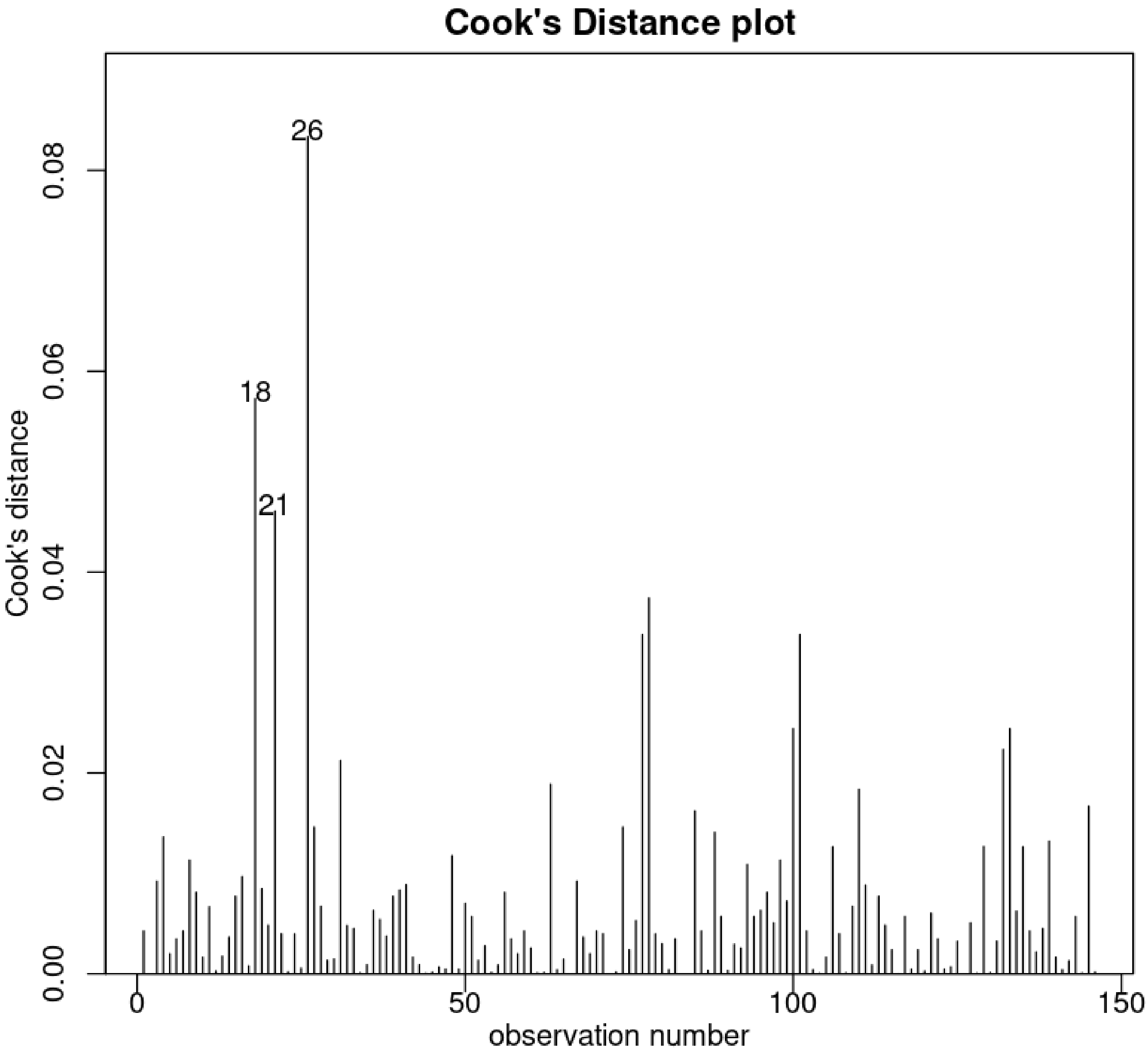
Skip to main content



Skip to main content



Skip to main content



```
## Create data frame of values of interest: Attend=="Yes" and "No"
## Make sure that the names of vars are exactly the same as in the data frame
preds.df <- data.frame(Attend = c("No", "Yes"))
predict(examattend.fit, preds.df, interval = "confidence")
predict(examattend.fit, preds.df, interval = "prediction")
```

Skip to main content

A matrix: 2 × 3 of type dbl

	fit	lwr	upr
1	42.21739	37.18401	47.25077
2	57.78000	54.36619	61.19381

A matrix: 2 × 3 of type dbl

	fit	lwr	upr
1	42.21739	7.710259	76.72452
2	57.78000	23.471673	92.08833

再次强调：“confidence”是代表均值预测范围，而“prediction”是代表个体预测范围。

6. Multiplicative linear models

本节需要的包：

```r
require(s20x)
```

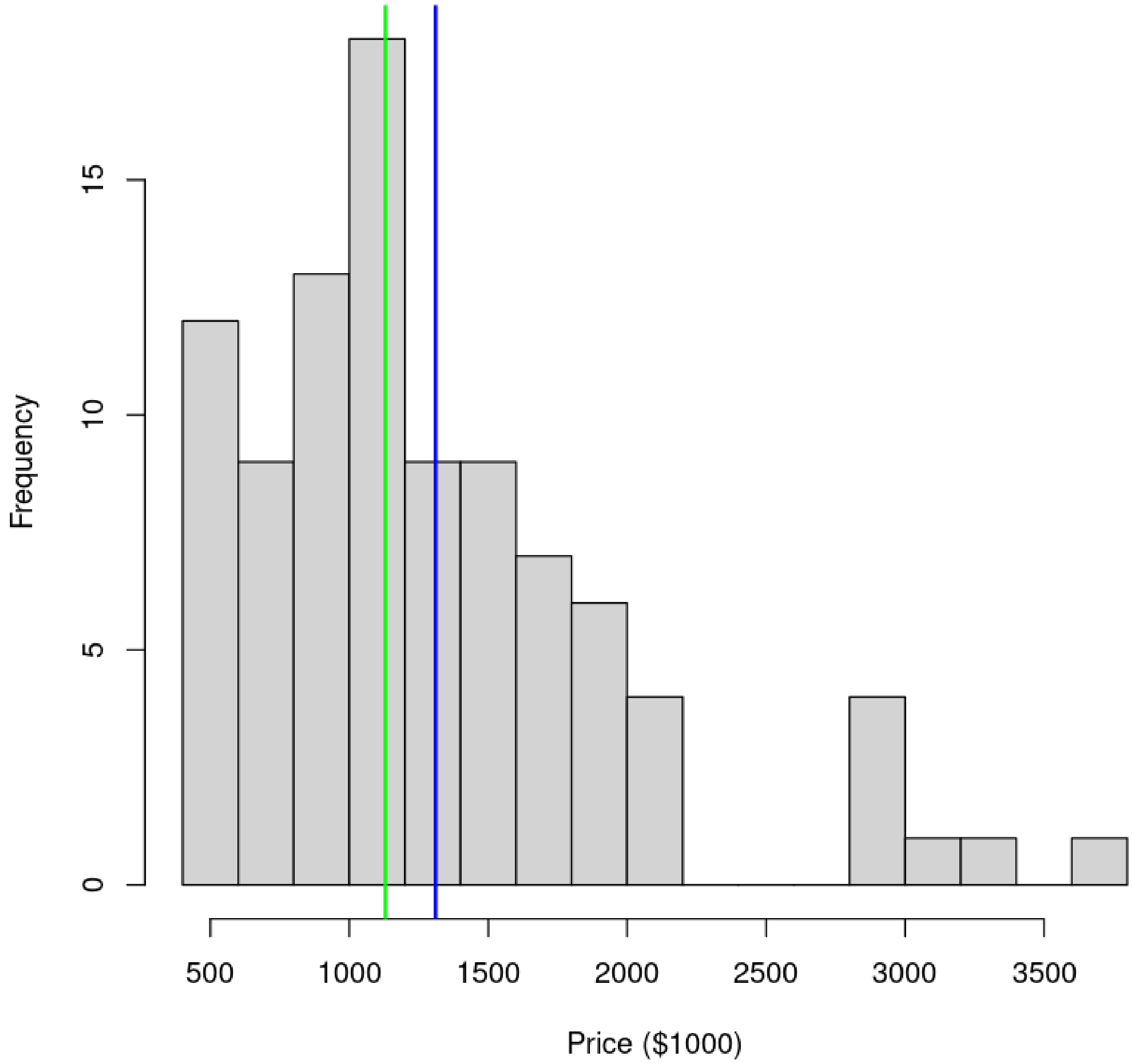
▶ Show code cell output

6.1. Mean versus median – which to use?

接下来我们会看到一个关于房价的数据集：典型的奥克兰郊区房价。

```r
library(s20x)
Houses.df <- read.table("../data/AkldHousePrices.txt", header = T)

hist(Houses.df$price, breaks = 20, main = "", xlab = "Price ($1000)")
abline(
    v = c(mean(Houses.df$price), median(Houses.df$price)),
    col = c("blue", "green"), lwd = 2
)
# 中位值为绿色，均值为蓝色
```

Skip to main content



这个数据就是典型的右偏：中位值比均值更小，因为右偏的分布有更多的更“离谱”的大值，整体却更偏向于小值

```
summary(Houses.df$price)
```

Skip to main content

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  450.0   832.5  1130.0  1310.1  1597.5  3710.0 
```

This type of right-skew distribution is very common when it comes to things involving money ($$$), resources, growth, salary, age, advantage and energy, to name but a few. 这种类型的右偏分布是非常常见的东西涉及金钱（￥￥￥）时，资源，经济增长，工资，年龄和能源优势，等等，不一而足。

Here is the bootstrap 95% CI for the expected price, along with output from the null model. 从数据中抽取了 1000 次，然后看抽取的数值在 5% 和 95% 之间的数值的分布情况，就是这个 95% 的置信区间。

```
bootstrappedMeanPrices <- replicate(
    1000,
    mean(sample(Houses.df$price, size = nrow(Houses.df), replace = T))
)

# 95% 置信区间
quantile(bootstrappedMeanPrices, c(.025, .975))

HousesNull.fit <- lm(price ~ 1, data = Houses.df)
summary(HousesNull.fit)
confint(HousesNull.fit)
```

2.5%: 1184.07180851064 **97.5%:** 1448.93882978723

```
Call:
lm(formula = price ~ 1, data = Houses.df)

Residuals:
   Min     1Q Median     3Q    Max 
-860.1 -477.6 -180.1  287.4 2399.9 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   1310.1       70.1   18.69   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 679.7 on 93 degrees of freedom
```

A matrix: 1 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	1170.899	1449.313

Skip to main content

上述的操作说明其实这个数据是满足中心极限定理的。

To estimate the median sale price of the entire suburb the natural estimate is the median of our sample 估计整个郊区的平均销售价格自然估计的样本中位数：

```
median(Houses.df$price)
```

1130

and we can use a bootstrap to get a 95% CI for the suburb median 我们可以使用一个引导郊区的 95% 中值可信区间值：

```
bootstrappedMedianPrices <- replicate(
    1000, median(sample(Houses.df$price, size = nrow(Houses.df), replace = T))
)
quantile(bootstrappedMedianPrices, c(.025, .975))
```

2.5%: 1045 **97.5%:** 1320

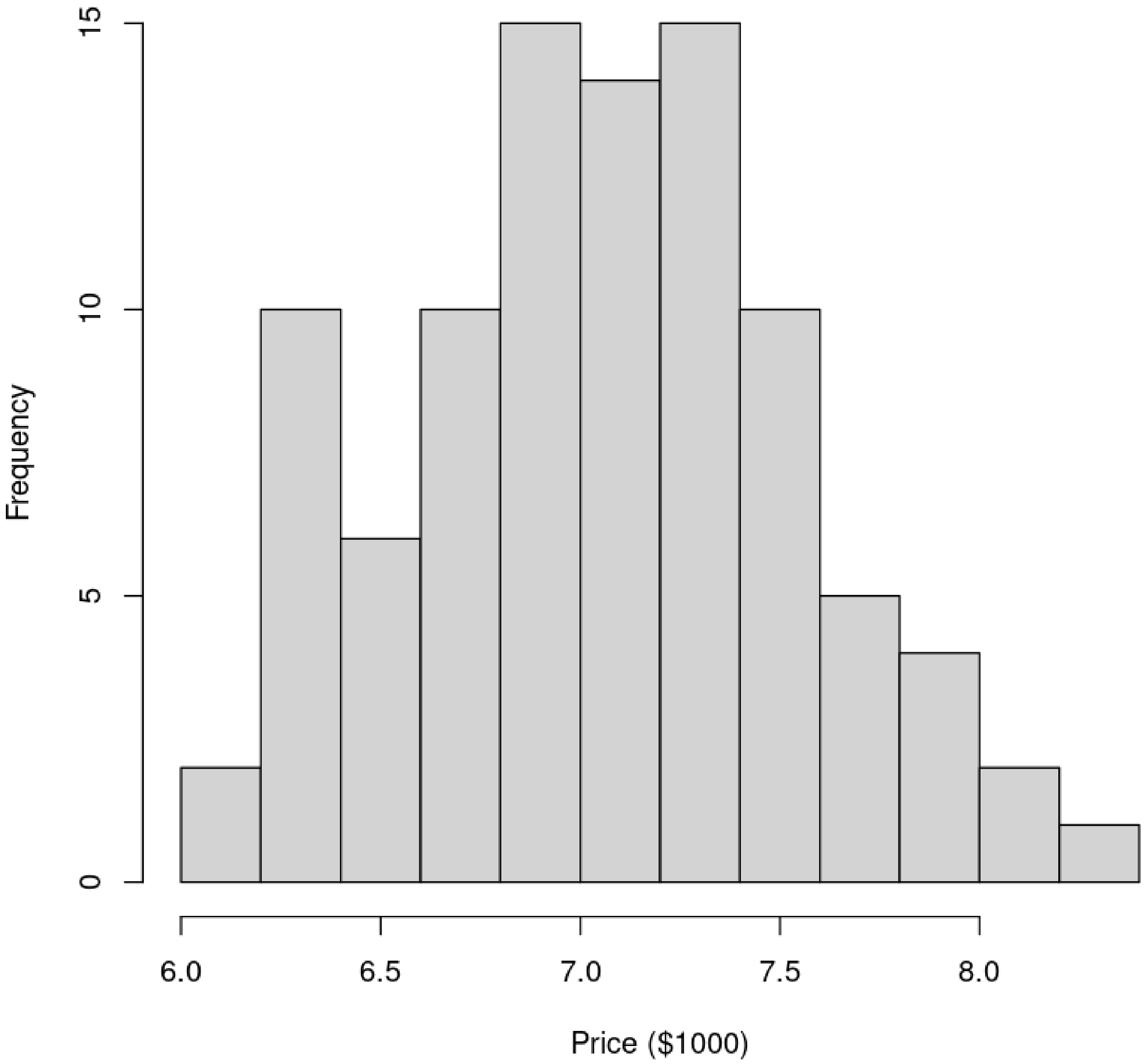
房价在上面的例子中,我们正在与 iid（Independent and Identically Distributed Data，独立和同分布数据）数据,所以很自然的使用示例值来估计人口值。在下一节中我们将看到，线性模型框架还可以用来进行推理所提供的值，记录反应变量为正态分布。这种方法的优点是它也适用于更一般的情况下我们有解释变量可能与响应变量联系在一起。我们也会看到登录响应数据结果拟合线性模型解释变量的影响作用在中位数用乘法。

6.2. Transforming the response variable using the log function

Let’s consider making a transformation of the prices. In particular(特别是), the log transformation. Here is the histogram of `log(price)`.

```
hist(log(Houses.df$price), breaks = 12, main = "", xlab = "Price ($1000)")
```

Skip to main content



This looks reasonably(合理的，相当的) close to normal, so if we fit a linear model to these data then all inferences(推论) will be valid(有效的).

```
LoggedPriceNull.fit <- lm(log(price) ~ 1, data = Houses.df) # 取对数
# log函数可以带底数参数 base；默认底数为 e（这里就是默认底数）
coef(summary(LoggedPriceNull.fit)) # 估计系数
confint(LoggedPriceNull.fit) # 置信区间
```

Skip to main content

A matrix: 1 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	7.060405	0.04974049	141.9448	1.628721e-110

A matrix: 1 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	6.96163	7.15918

这很有趣，但记录房价不意味着很多人希望买一栋房子。推理需要 back-transformed 价格（新西兰元）。

Since we’ve used the log transformation(转换), the back-transformation(回转) is the exponential(指数的) function `exp()`

```
exp(confint(LoggedPriceNull.fit))
# exp函数同样可以带底数参数 base；同上。
```

A matrix: 1 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	1055.353	1285.856

上述计算置信区间是完全不同于我们的平均房价郊区。上面的原因是因为算的房价中值。明白这是为什么，让我们看看会发生什么当我们变换摘要统计信息使用和功能：

```
# Summaries of price
summary(Houses.df$price)
# Summaries of log(price)
summary(log(Houses.df$price))
# Back-transformed summaries of log(price)
exp(summary(log(Houses.df$price)))
```

Skip to main content

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  450.0   832.5  1130.0  1310.1  1597.5  3710.0 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  6.109   6.724   7.030   7.060   7.376   8.219 
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  450.0   832.5  1130.0  1164.9  1597.0  3710.0 
```

Our back-transformed estimate(估计) ($exp(\hat{\beta}_0)$) and 95% CI(Confidence interval, 置信区间) for the median suburb sale price(郊区销售价格) are:

```
exp(coef(LoggedPriceNull.fit)) # 估计系数（Intercept）
exp(confint(LoggedPriceNull.fit)) # Intercept 的置信区间
```

(Intercept): 1164.91688878205

A matrix: 1 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	1055.353	1285.856

6.3. The log function turns multiplicative effects in to additive effects

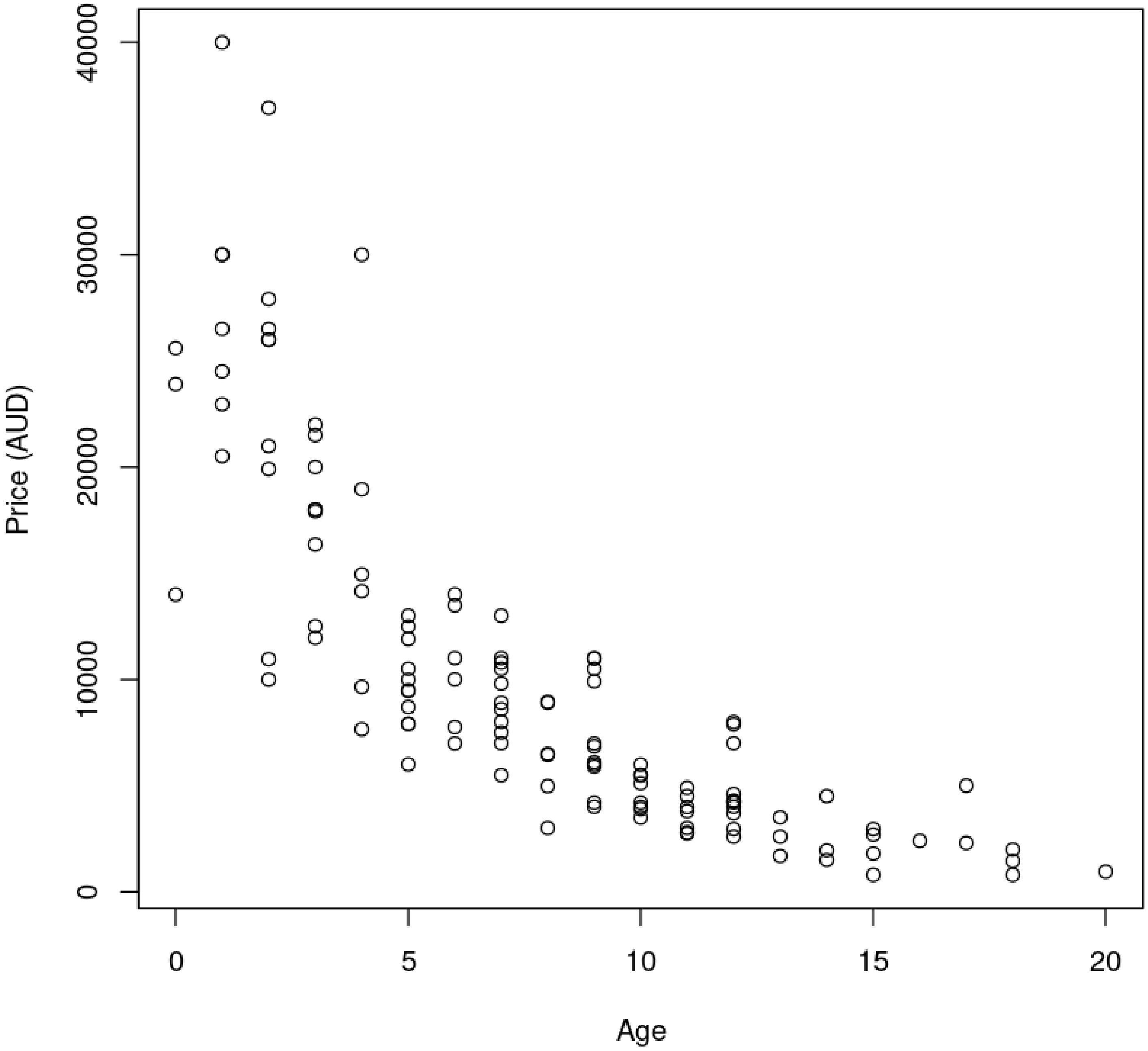
6.4. Example 1: Multiplicative simple linear regression model

乘法简单线性回归模型是一个线性模型，其中响应变量是一个或多个自变量的乘积。这是一个非常简单的模型，但是它是一个很好的起点，因为它可以用来解释一些非常有趣的现象。

```
Mazda.df <- read.table("../data/mazda.txt", header = T)
Mazda.df$age <- 91 - Mazda.df$year # Create the age variable
plot(price ~ age, data = Mazda.df, xlab = "Age", ylab = "Price (AUD)")

trendscatter(
    price ~ age,
    data = Mazda.df, xlab = "Age", ylab = "Price (AUD)"
)
```

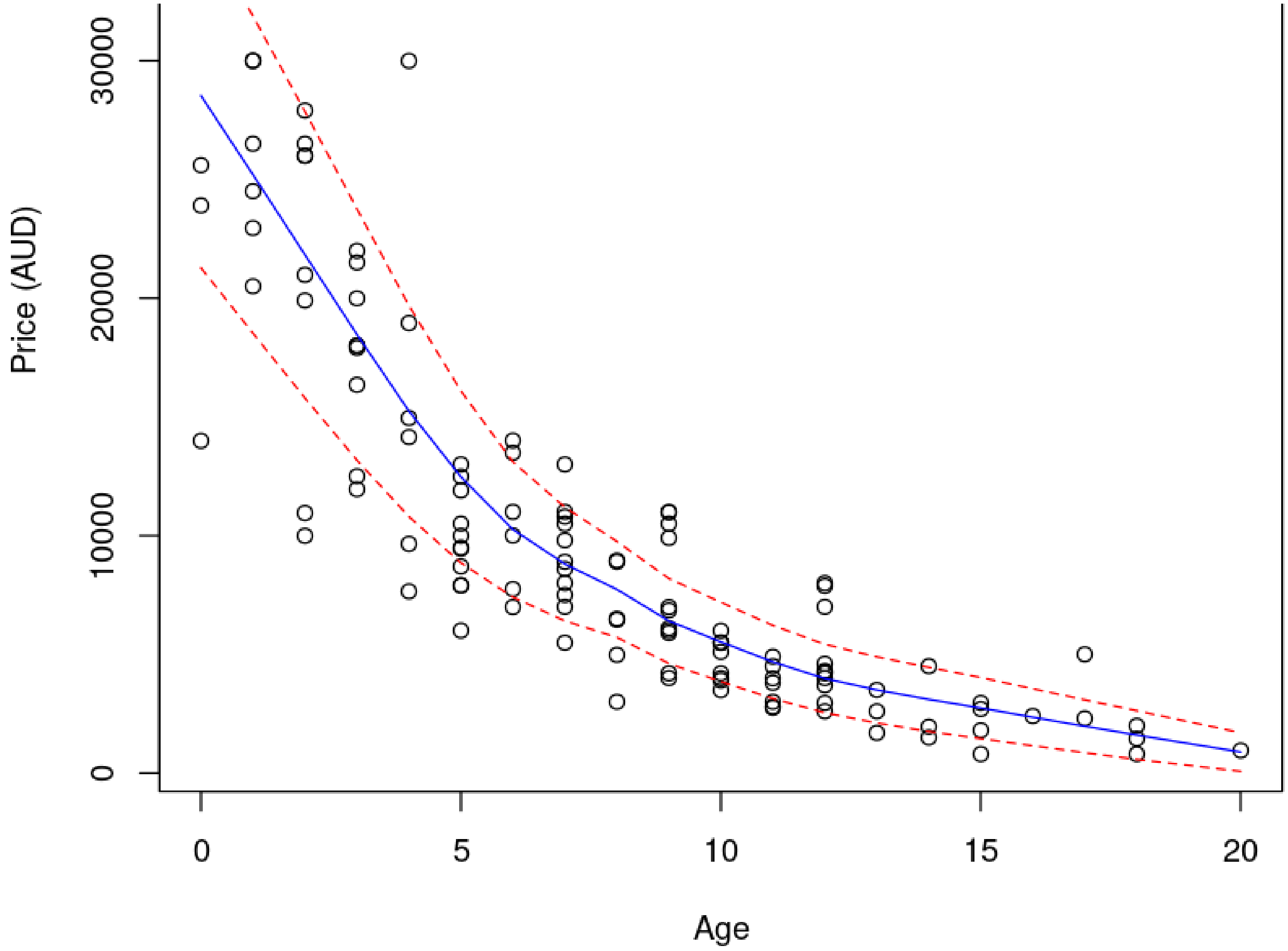
Skip to main content

Skip to main content



Plot of Price (AUD) vs. Age (lowess+/-sd)

Skip to main content

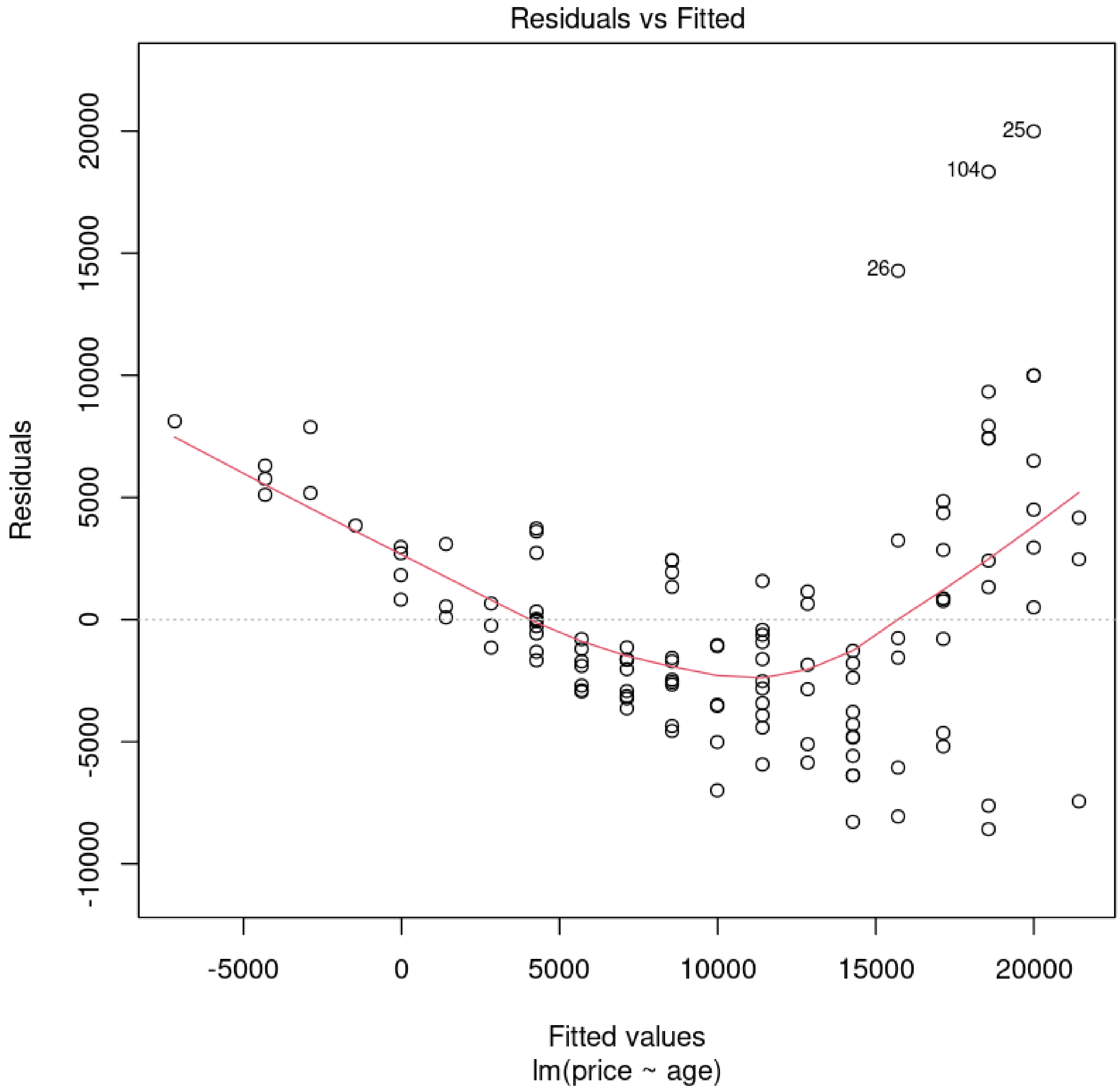


`trendscatter()` 函数给出的蓝线代表均值，红色线代表均值区间。

趋势是减少（指数），以及减少散射这些都是一个潜在的乘法模型的典型症状。假 Assuming would be na ̈ıve in this case. Let us be na ̈ıve and see where it takes us.让我们适应一个线性模型，看看剩余情节告诉我们什么。

```
PriceAge.fit <- lm(price ~ age, data = Mazda.df)
plot(PriceAge.fit, which = 1)
```

Skip to main content

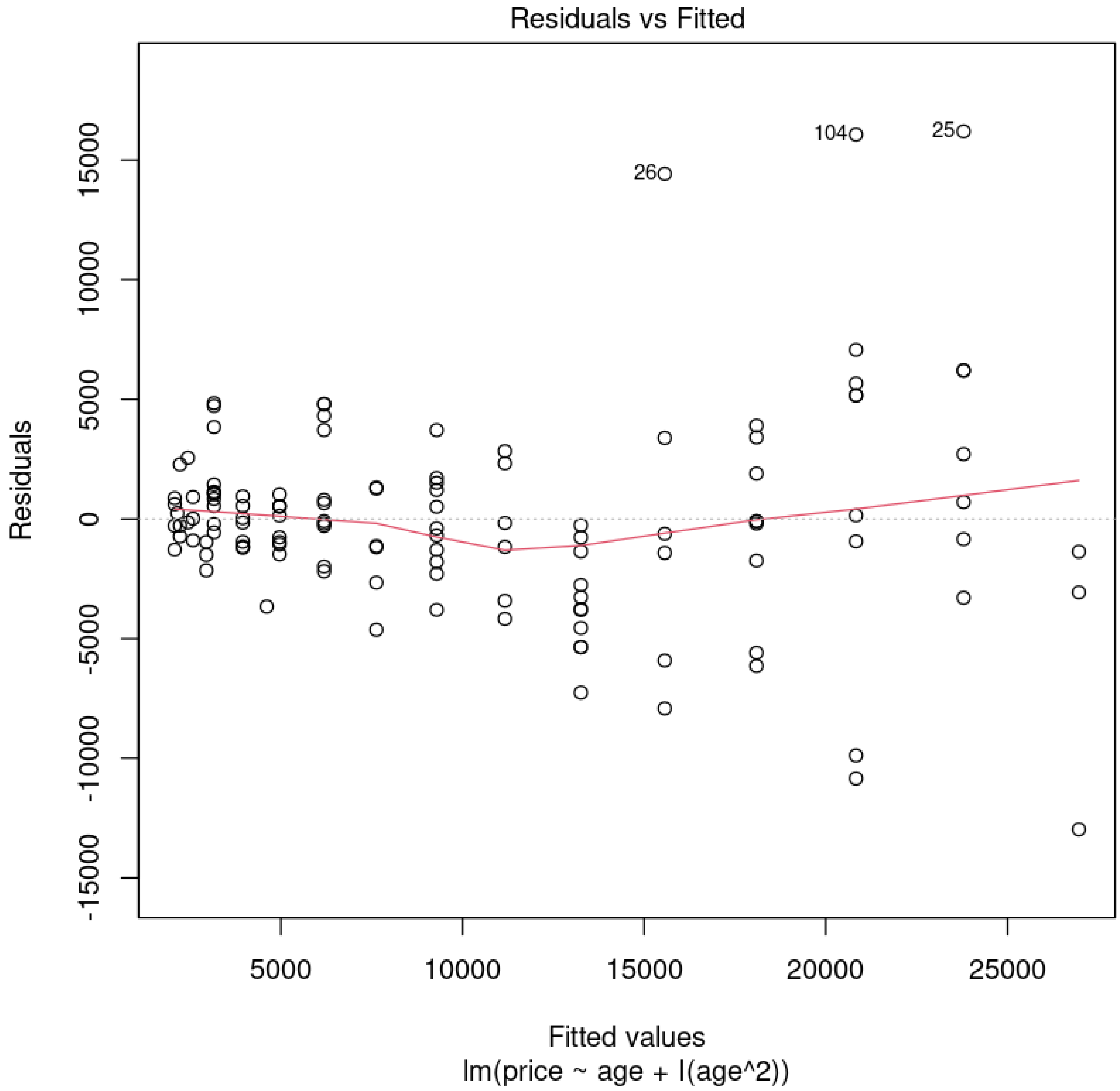


非线性下降趋势和不恒定散射已经变得更加明显。

Na¨ıve price vs age models... 适应价格与年龄模型...

```
PriceAge.fit2 <- lm(price ~ age + I(age^2), data = Mazda.df)
plot(PriceAge.fit2, which = 1)
```

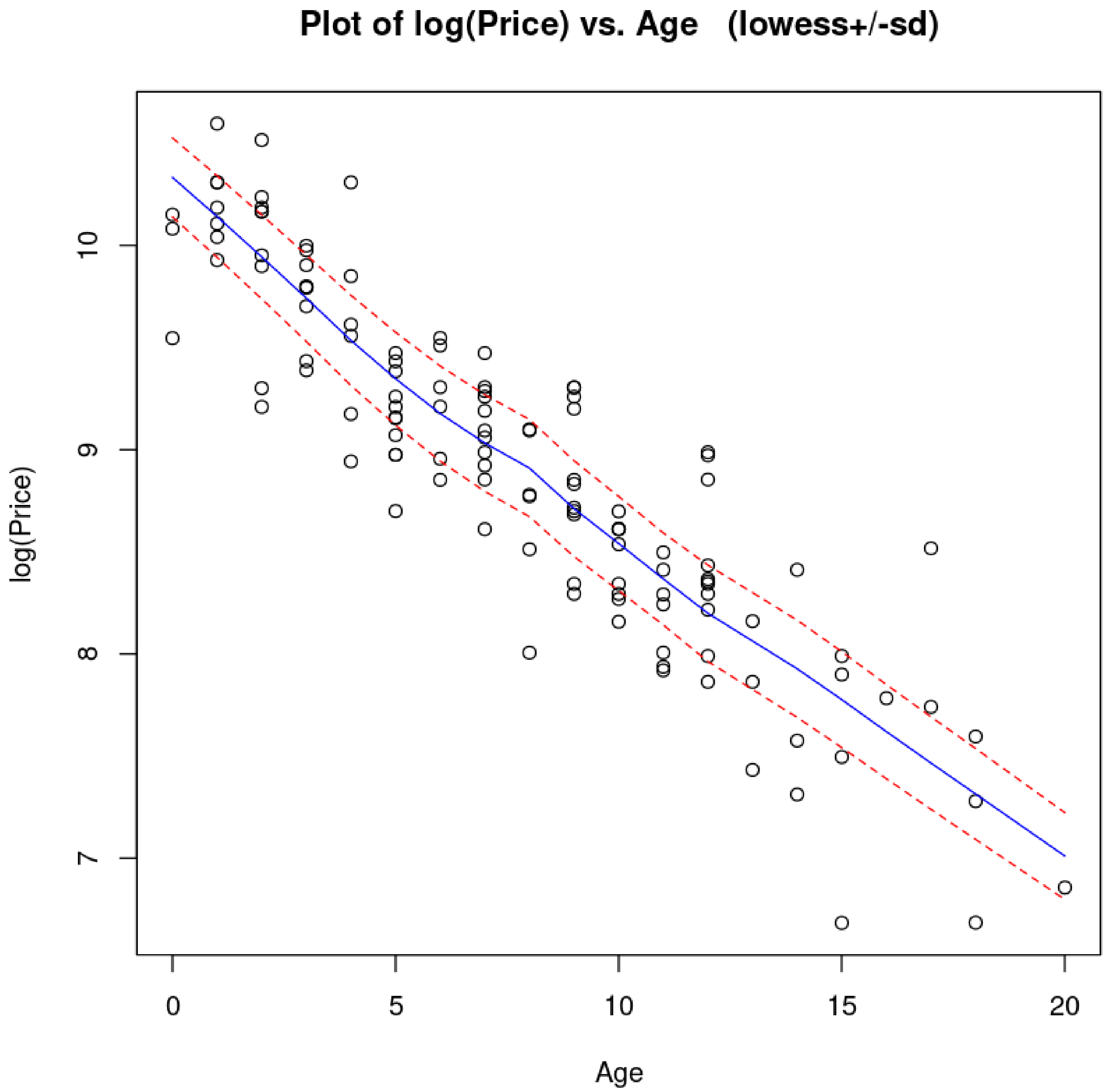
Skip to main content



We have eliminated trend from these residuals but the assumption is still violated. 我们从这些残差但假设消除趋势仍然是违反了。Let us ‘tear up’ this approach and take logs of price.

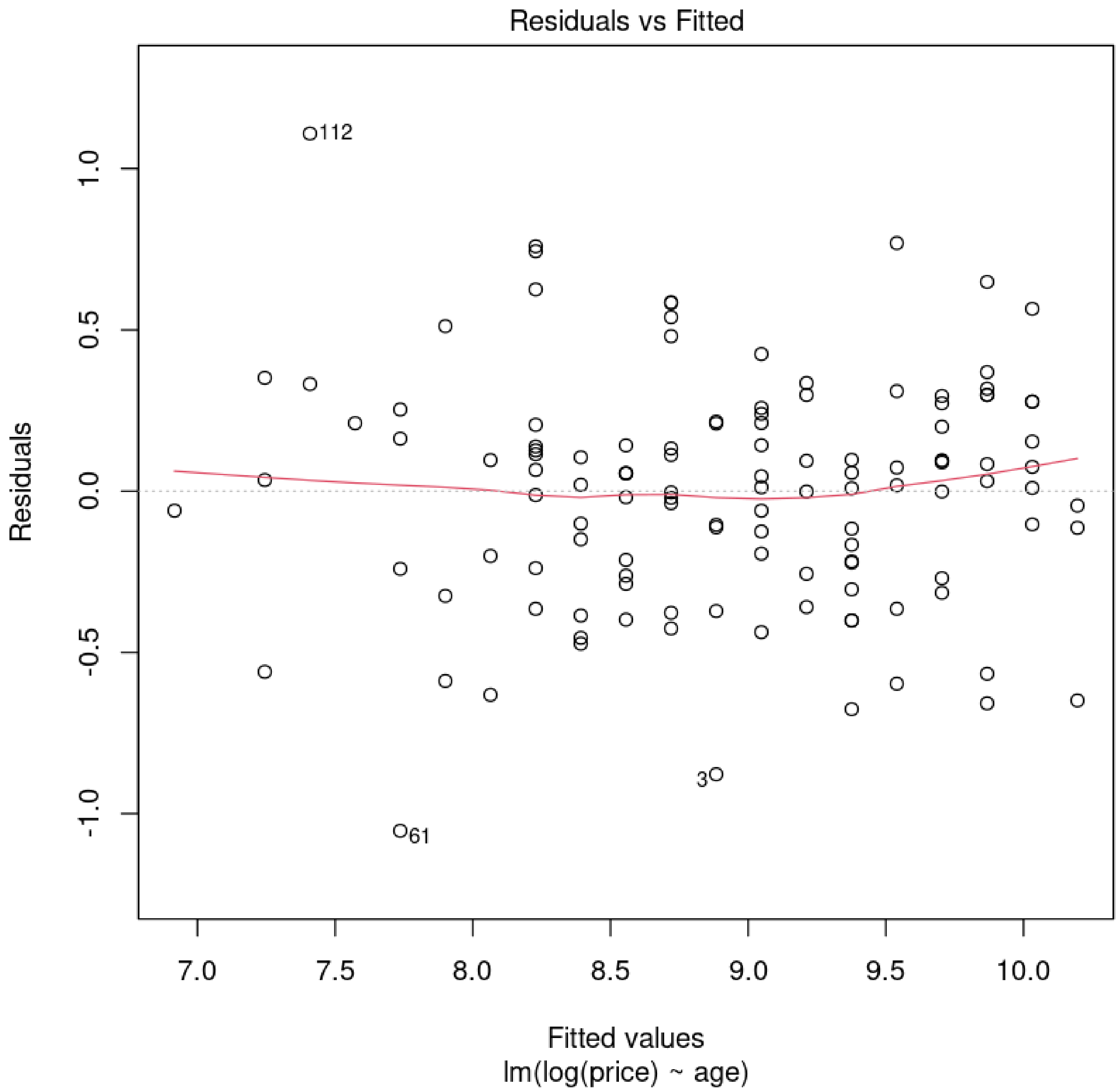
```
# log(Price)
trendscatter(
    log(price) ~ age,
    data = Mazda.df, xlab = "Age", ylab = "log(Price)"
```

Skip to main content



```
LogPriceAge.fit <- lm(log(price) ~ age, data = Mazda.df)
plot(LogPriceAge.fit, which = 1)
```

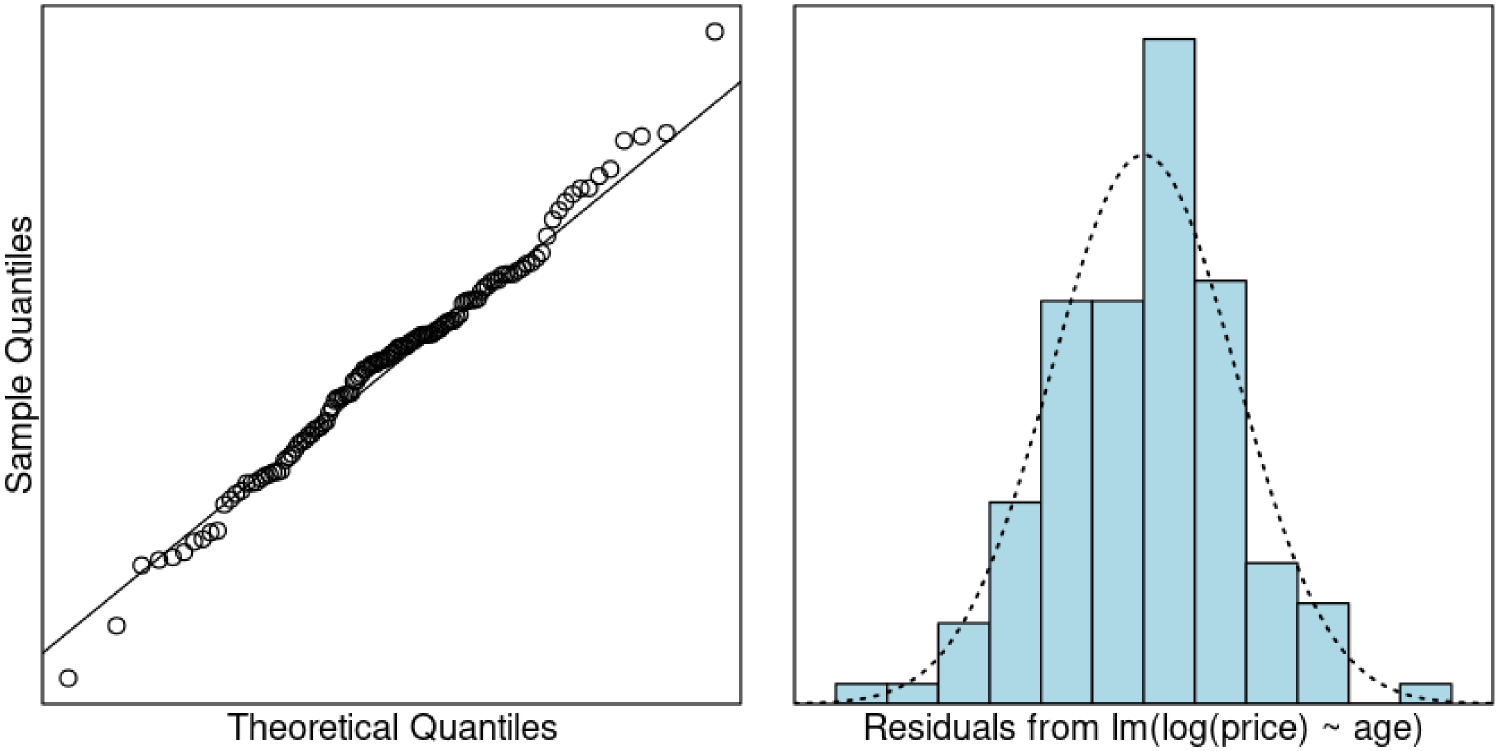
Skip to main content

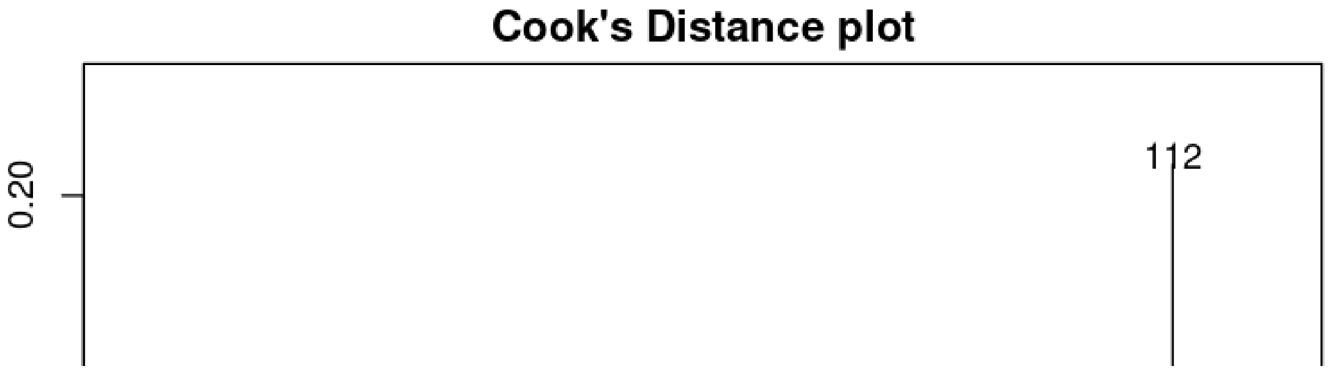


```
# Check for normality of the residuals.
normcheck(LogPriceAge.fit)
# Check for unduly influential data points.
cooks20x(LogPriceAge.fit)
```

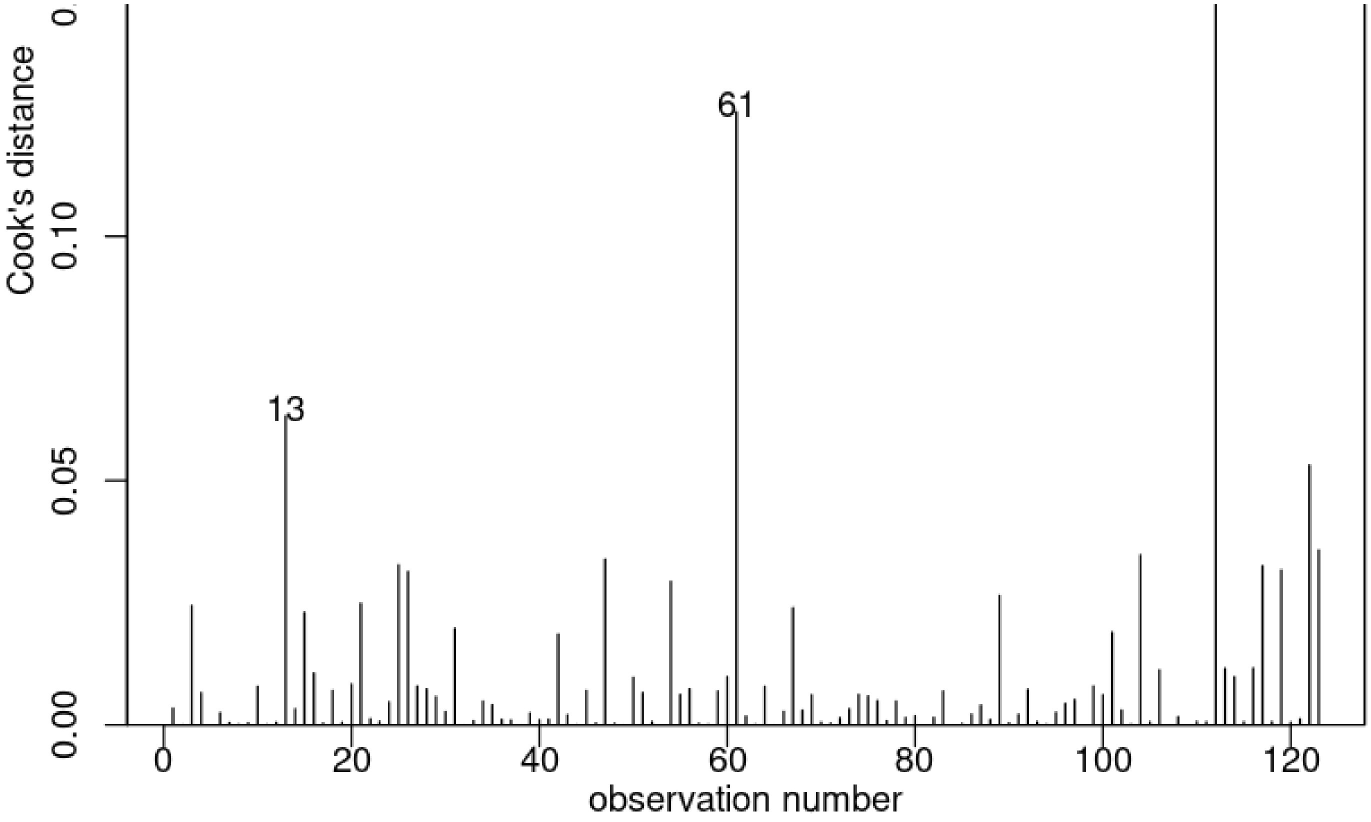
Skip to main content

Skip to main content





Skip to main content



```
summary(LogPriceAge.fit)
confint(LogPriceAge.fit)
```

Skip to main content

```
Call:
lm(formula = log(price) ~ age, data = Mazda.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.0531 -0.2398  0.0311  0.2110  1.1085 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10.195210   0.063602   160.3   <2e-16 ***
age         -0.163915   0.007034   -23.3   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3615 on 121 degrees of freedom
Multiple R-squared:  0.8178,	Adjusted R-squared:  0.8163 
F-statistic: 543.1 on 1 and 121 DF,  p-value: < 2.2e-16
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	10.0692935	10.3211263
age	-0.1778406	-0.1499902

我们可以获得置信区间的中间价格的一辆新车回转换得到的中值，就像我们前面讨论的零模型。

```
exp(confint(LogPriceAge.fit))
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	2.360688e+04	3.036744e+04
age	8.370758e-01	8.607164e-01

```
100 * (exp(confint(LogPriceAge.fit)[2, ]) - 1)
```

2.5 %: -16.2924152317926 **97.5 %:** -13.9283629045699

This says that our 95% CI for the annual depreciation in median price of Mazda cars is between $100\% \times (1 - 0.861) = 13.9\%$ and $100\% \times (1 - 0.837) = 16.3\%$

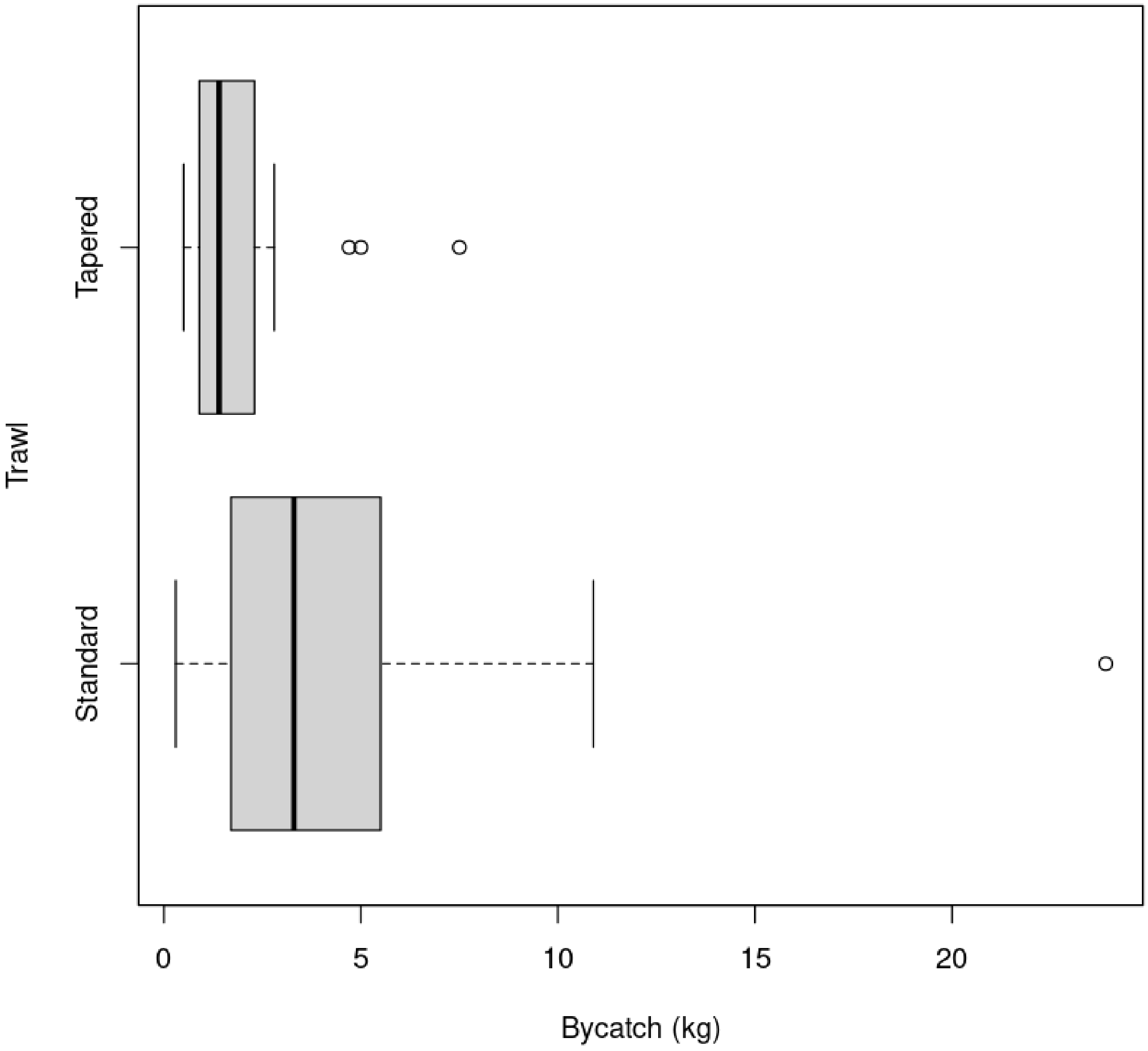
Skip to main content

6.5. Example 2: Multiplicative model with categorical explanatory variable

```r
Bycatch.df <- read.table("../data/Bycatch.txt", header = T)
boxplot(Bycatch ~ Trawl, data = Bycatch.df, horizontal = T, xlab = "Bycatch (kg)")
summaryStats(Bycatch ~ Trawl, data = Bycatch.df)
```

Skip to main content

```
         Sample Size  Mean Median  Std Dev Midspread
Standard          25 4.600    3.3 4.983138       3.8
Tapered           25 1.924    1.4 1.643999       1.4
```

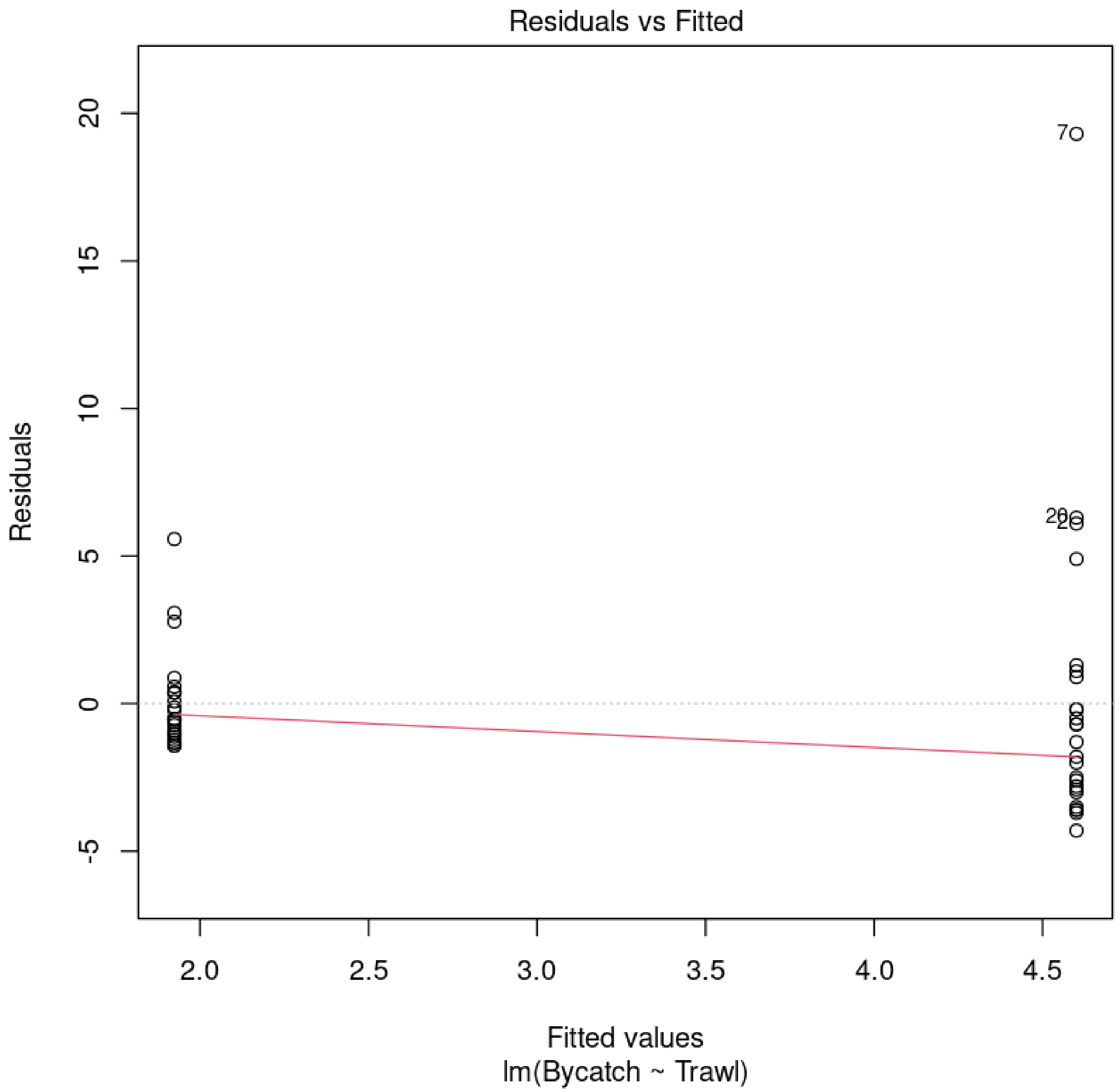


这似乎证实了我们的直觉，这些数据应该仿照对数尺度。线性模型的拟合效果真的很差...

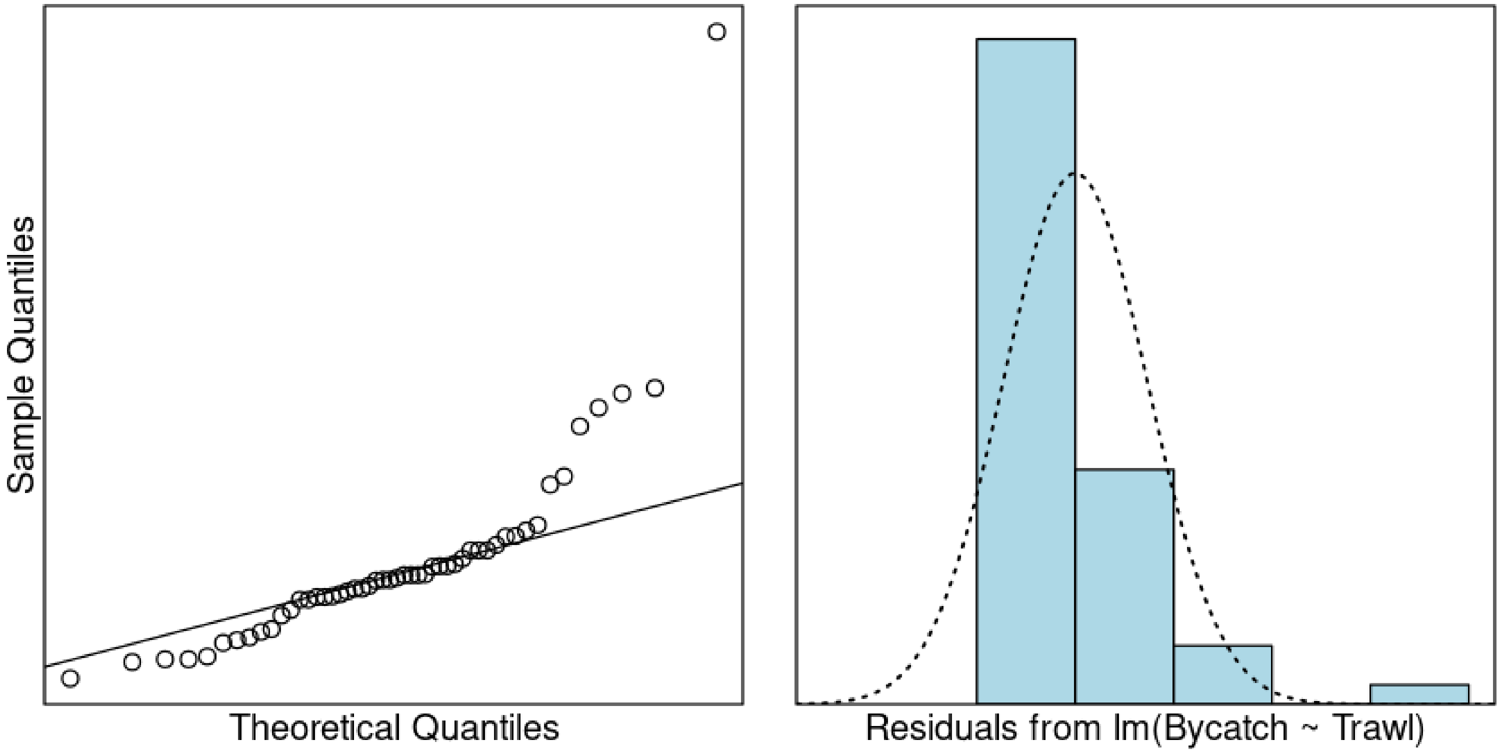
Skip to main content

```r
Trawl.lm <- lm(Bycatch ~ Trawl, data = Bycatch.df)
plot(Trawl.lm, which = 1)
normcheck(Trawl.lm)
```

Skip to main content



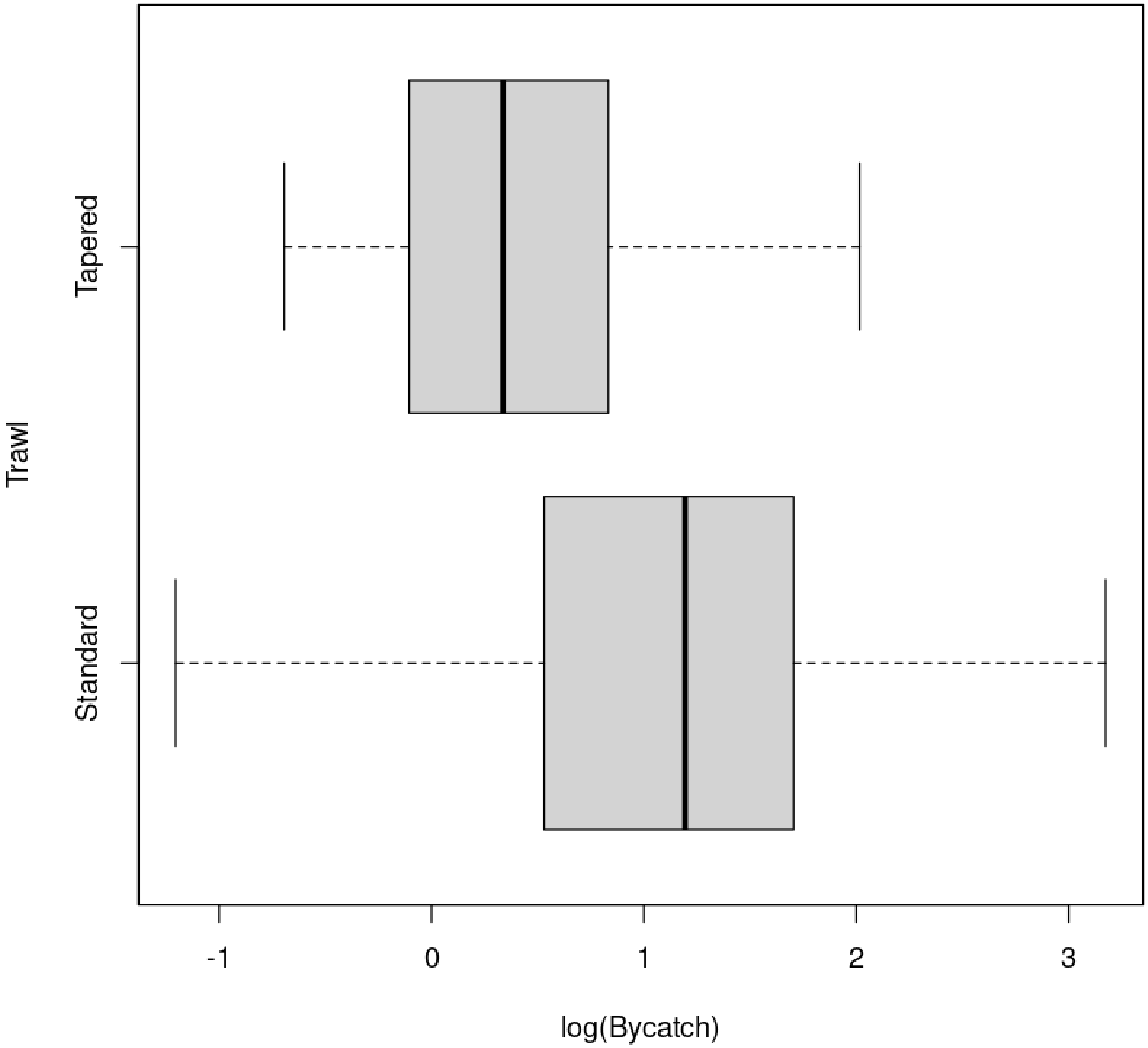
Skip to main content



Multiplicative model with categorical explanatory variable 乘法模型分类解释变量

```
boxplot(log(Bycatch) ~ Trawl, data = Bycatch.df, horizontal = T, xlab = "log(Bycatch)")
```

Skip to main content



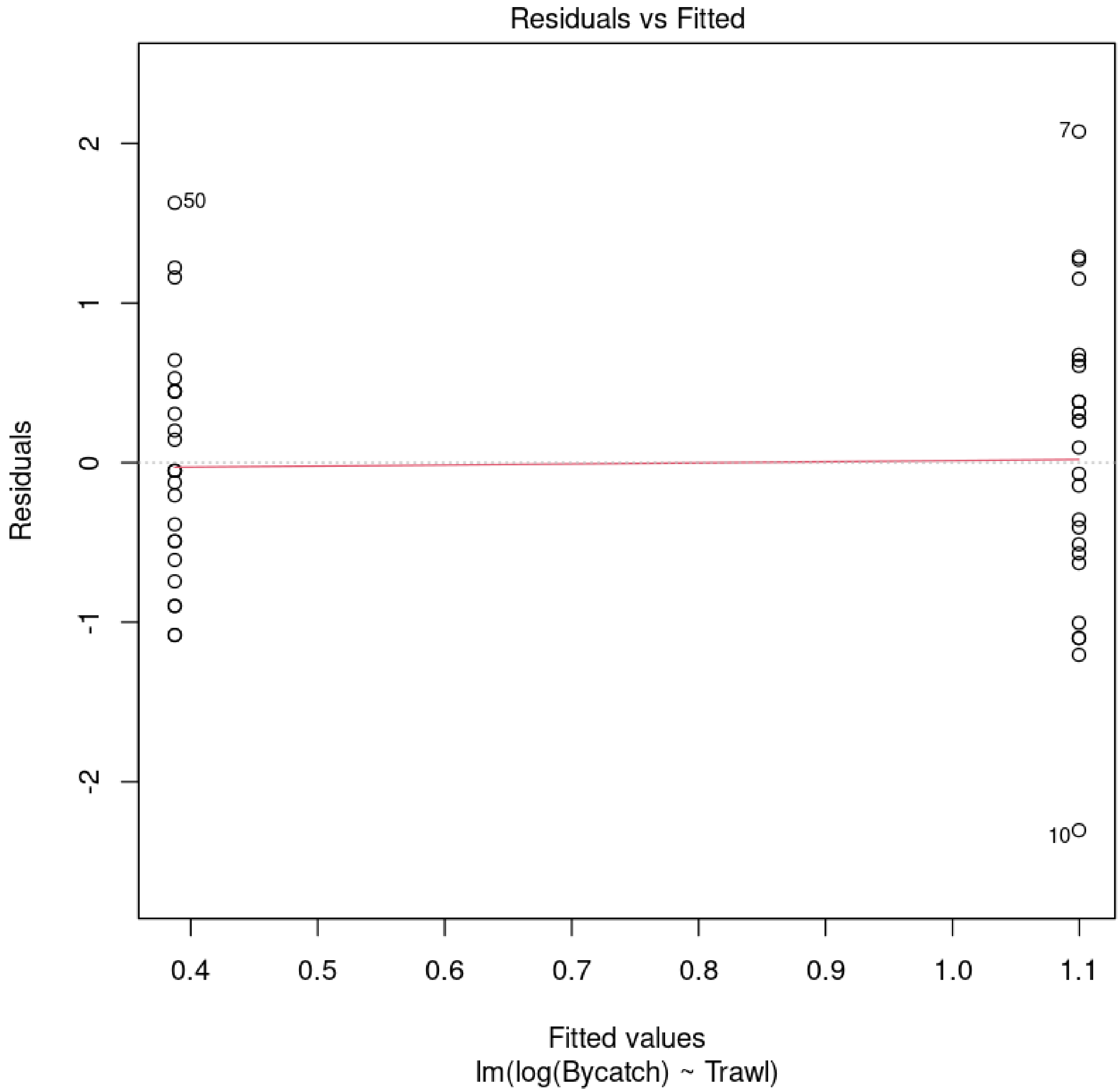
Looking much better.

接下来我们将进行建模 + 验证三部曲。

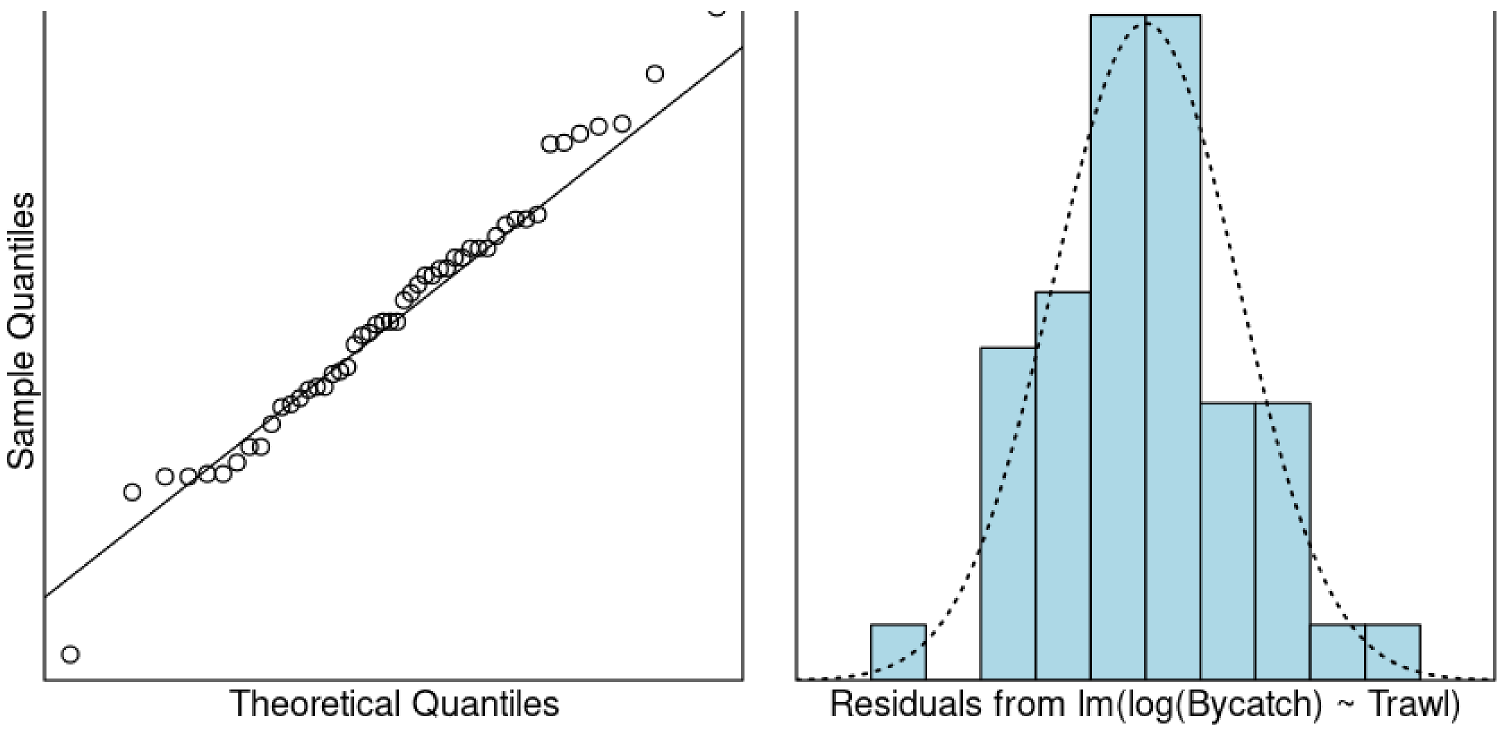
```
Trawl.lmlog <- lm(log(Bycatch) ~ Trawl, data = Bycatch.df)
plot(Trawl.lmlog, which = 1)
normcheck(Trawl.lmlog)
```

Skip to main content

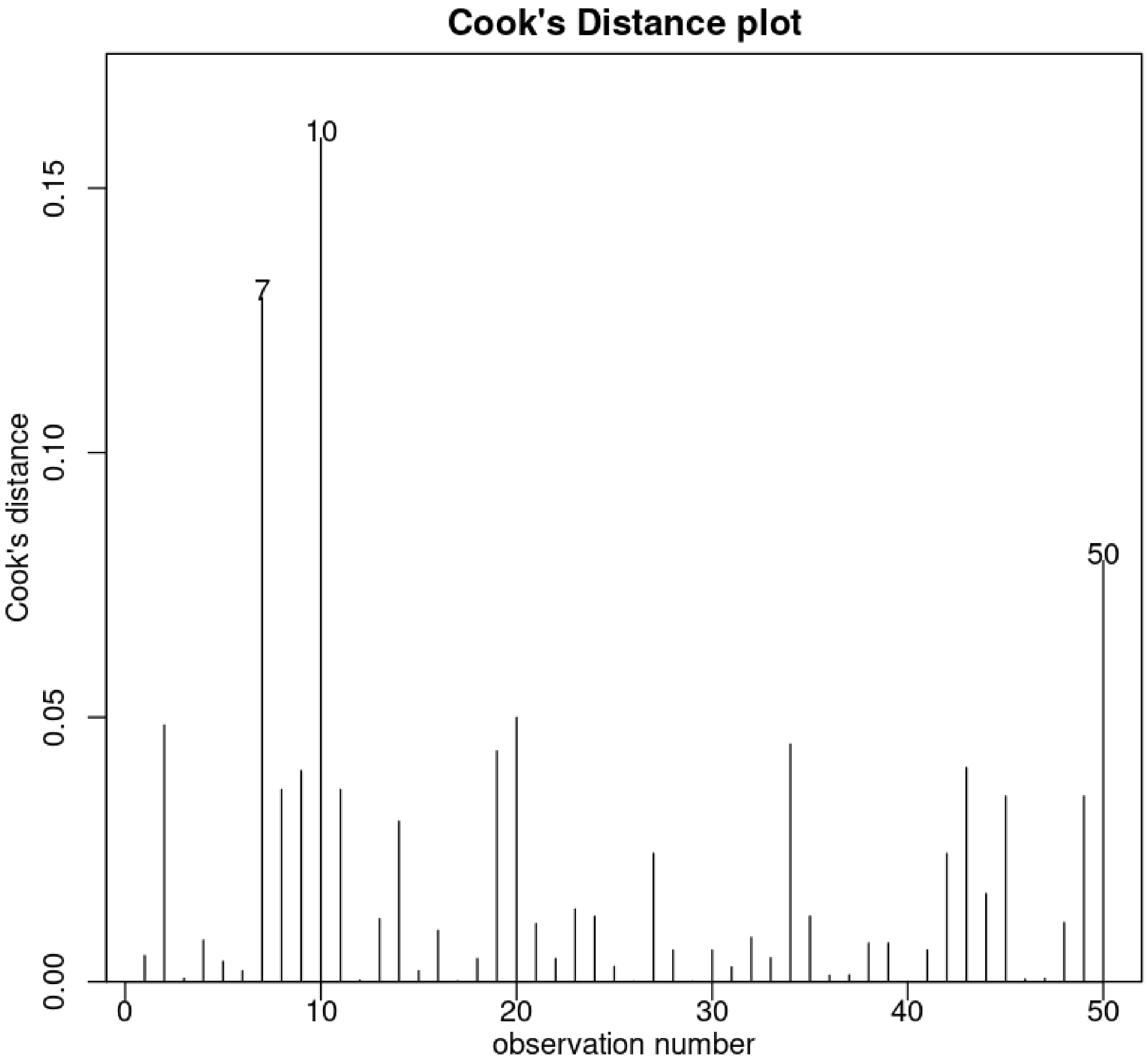
Skip to main content



Skip to main content



Skip to main content



Assumptions are satisfied. We can trust the fitted model.

```
summary(Trawl.lmlog)
```

Skip to main content

```
Call:
lm(formula = log(Bycatch) ~ Trawl, data = Bycatch.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.30353 -0.55464 -0.05088  0.44556  2.07432 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)    1.0996     0.1700   6.469 4.79e-08 ***
TrawlTapered  -0.7122     0.2404  -2.963  0.00473 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8498 on 48 degrees of freedom
Multiple R-squared:  0.1546,	Adjusted R-squared:  0.137 
F-statistic:  8.78 on 1 and 48 DF,  p-value: 0.004728
```

There is a statistically significance effect of trawl type(trawl type 的影响有统计学意义) ($P - value \approx 0.05$). However, our model only explained 15% of the variability in the logged data and will not be very good for prediction. 然而,我们的模型只能解释 15%的变异在记录数据，并不能很好的预测。

```
exp(confint(Trawl.lmlog))
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	2.1336329	4.2261691
TrawlTapered	0.3025531	0.7953873

附模型方程：

$$\log(y) = \beta_0 + \beta_1 \times \mathrm{x} + \epsilon$$

什么时候直接用线性模型，什么时候要取对数？

有明显的正态分布或者线性关系就可以用线性模型，否则就要取对数。当然我们也可以通过“右偏”效果来看取对数的必要性（其中之一：中位值比均值要小一点）。事实上取对数也只是为了更好的拟合模型，是手段而非万能方法。

Skip to main content

7. Power law linear models

本节需要的包：

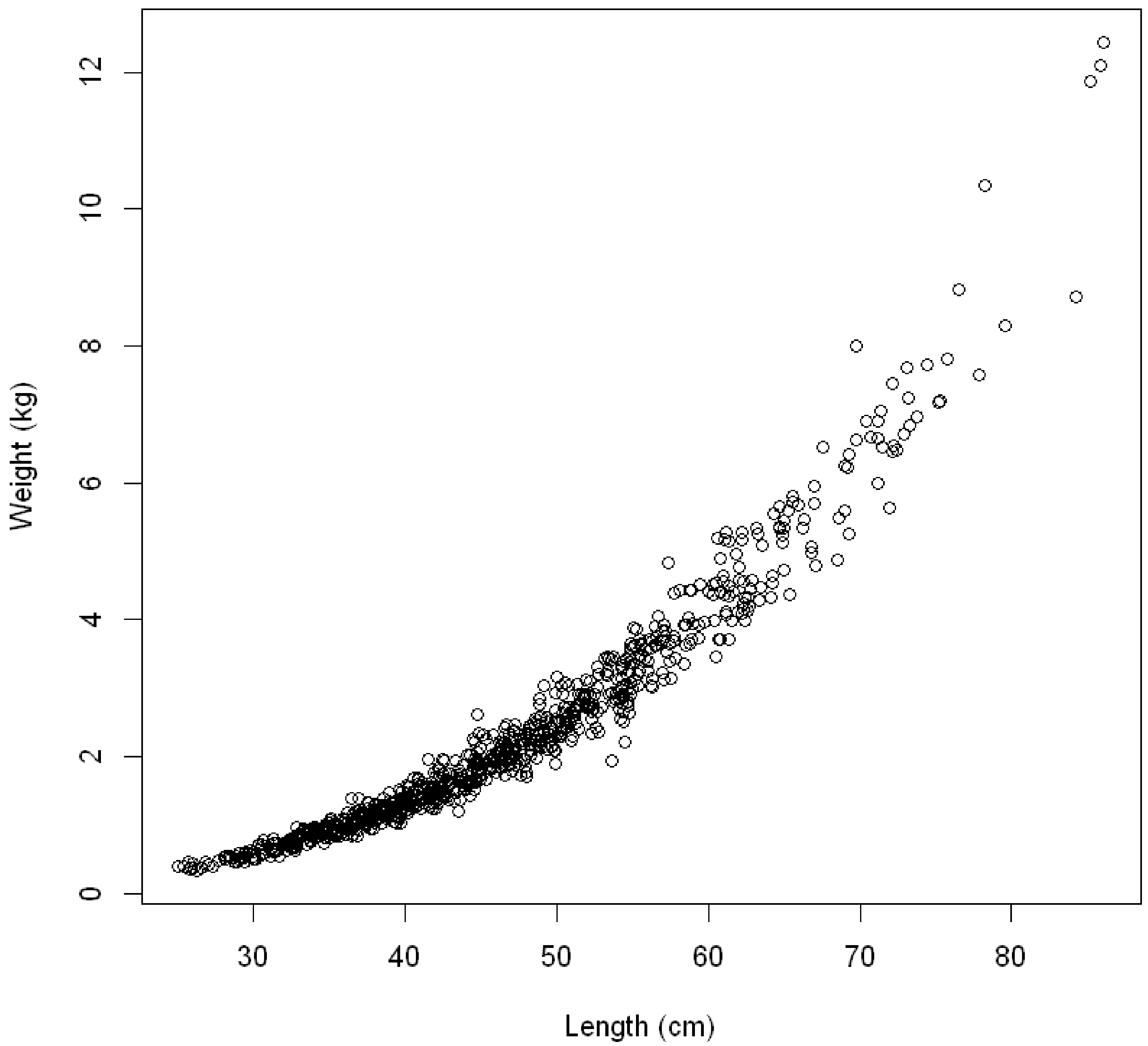
```
require(s20x)
library(s20x)
```

▶ Show code cell output

那些鱼在 Hauraki(墨西哥湾) Gulf 就知道最低法定大小保持 30 厘米鲷鱼（小于 30 厘米的鱼禁止捕捞）。在这里，我们想使用鲷鱼长度解释鲷鱼的重量，特别是我们要估计 30 厘米鲷鱼的重量。

```
Snap.df <- read.table("../data/SnapWgt.txt", header = TRUE)
plot(wgt ~ len, data = Snap.df, xlab = "Length (cm)", ylab = "Weight (kg)")
```

Skip to main content



显然有一个非线性的重量和长度之间的关系。几何告诉我们，如果一个对象的总体规模变化，同时却保持相同的形状（即同样的比率高、深度和长度），那么它的体积会增加长度的 3 次方。

- For a cube with sides of length l, $volume = len^3$.
- For a sphere with radius r, $volume = \frac{4}{3}\pi r^3$.

Skip to main content

$$weight = \alpha \times length^{\beta_1}$$

β_1 可能接近常数，但不一定等于 3。

Taking logs gives

$$\log(weight) = \log(\alpha) + \beta \log(length)$$

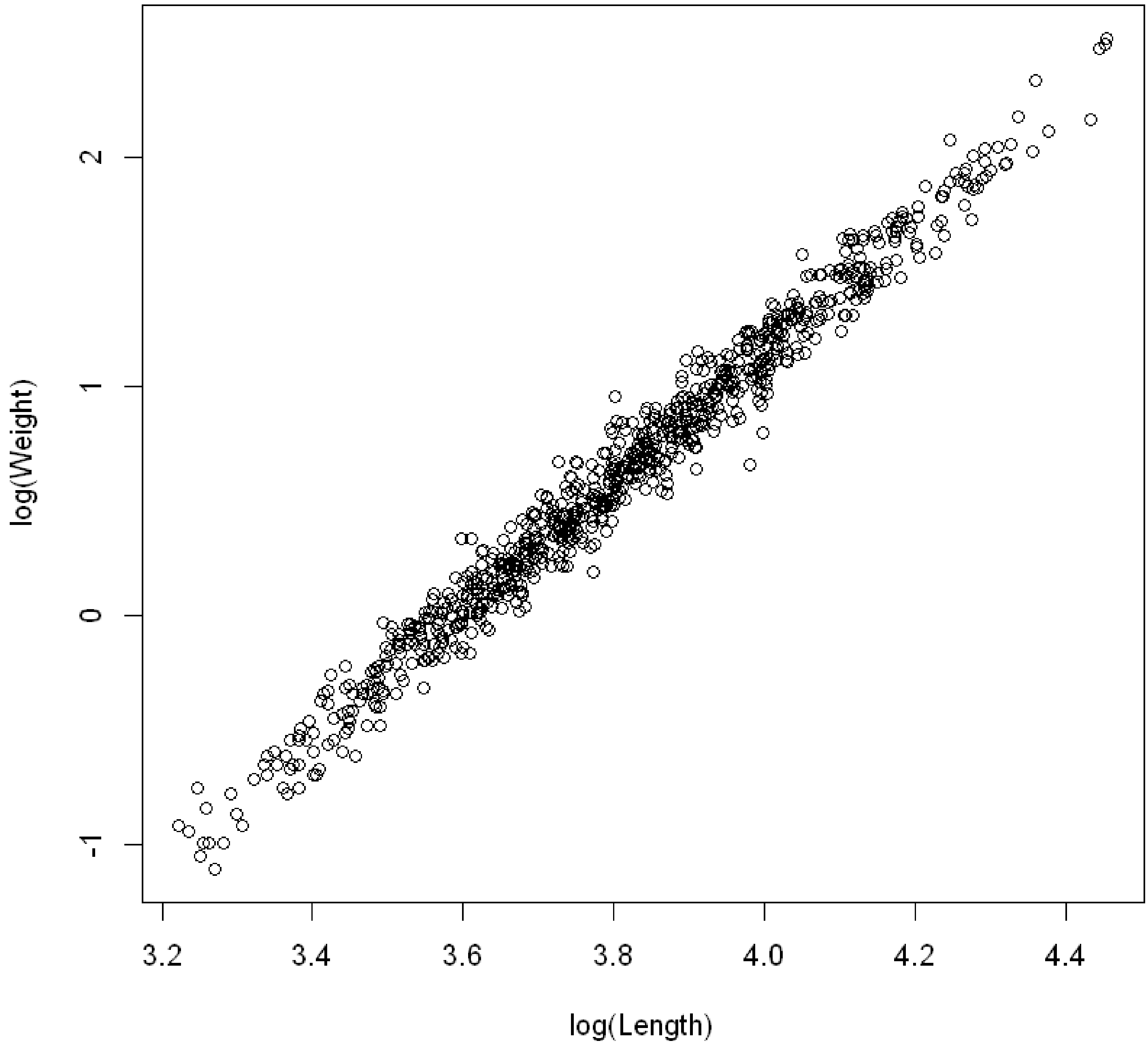
which we can rewrite as:

$$\log(weight) = \beta_0 + \beta_1 \log(length)$$

The above formula should be of very familiar form to you by now. Provided that we make the assumption that $\varepsilon \ N(0, \sigma)$ then this isprecisely the simple linear regression model with response variable $log(weight)$ and explanatory variable $log(len)$.

```
plot(
    log(wgt) ~ log(len),
    data = Snap.df,
    xlab = "log(Length)",
    ylab = "log(Weight)"
)
```

Skip to main content

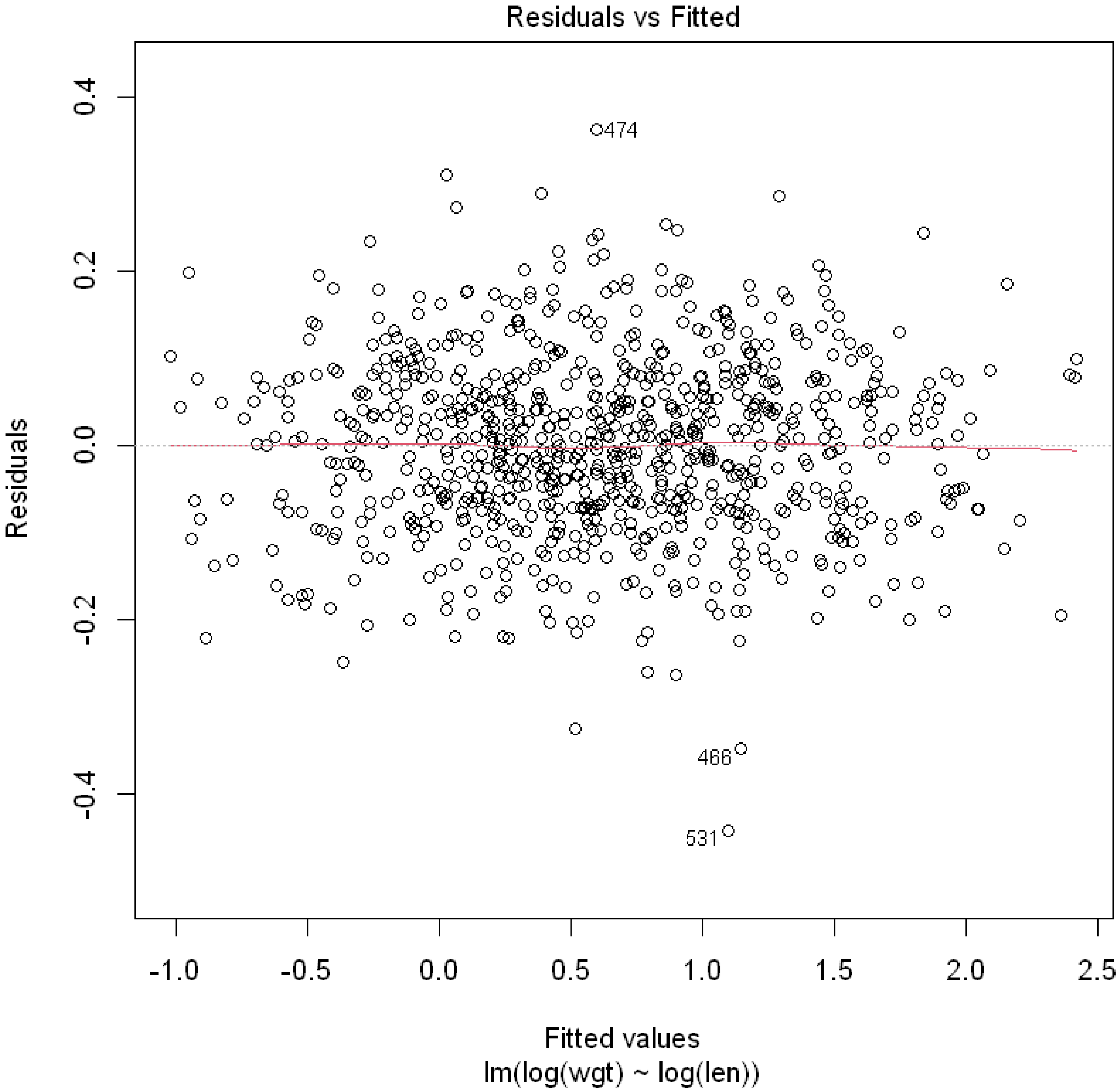


看起来这样取对数后有很好的线性关系。接下来我们将建模 + 检验三步走：

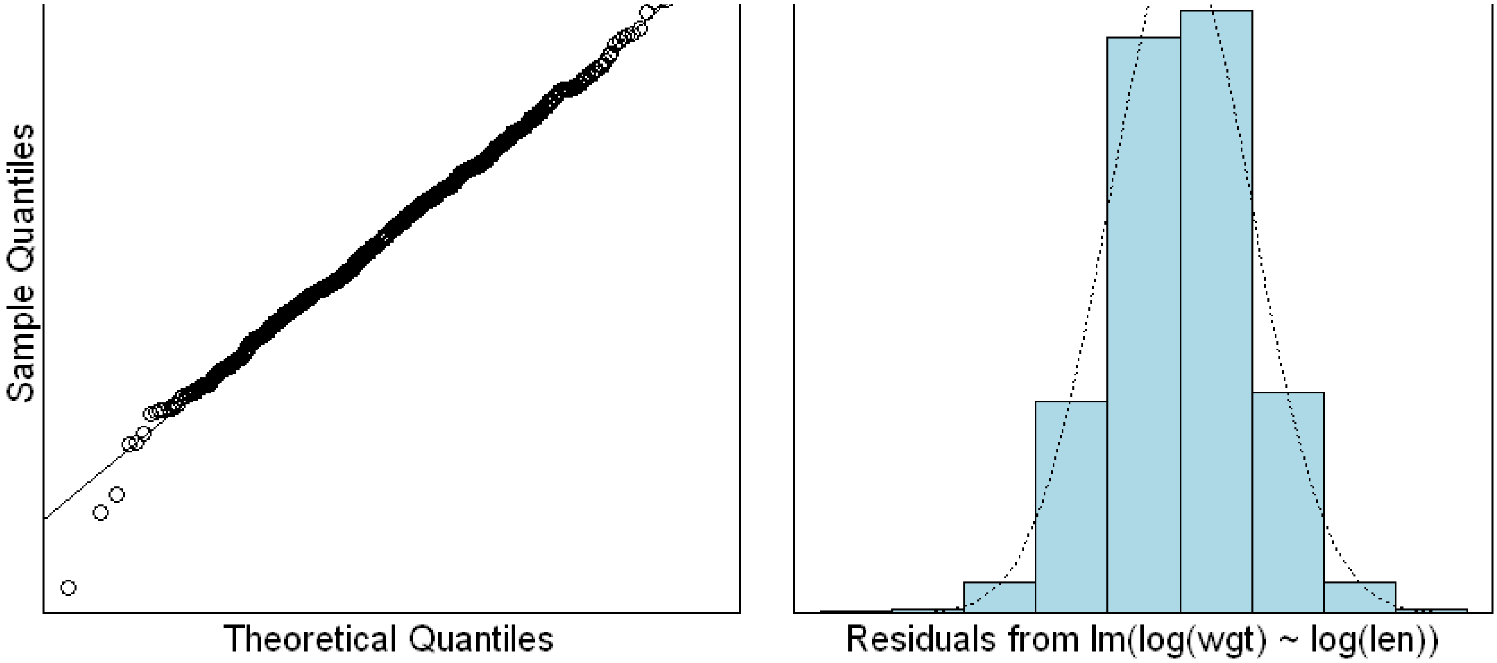
```
Snap.lm <- lm(log(wgt) ~ log(len), data = Snap.df)
plot(Snap.lm, which = 1)
normcheck(Snap.lm)
cooks20x(Snap.lm)
```

Skip to main content

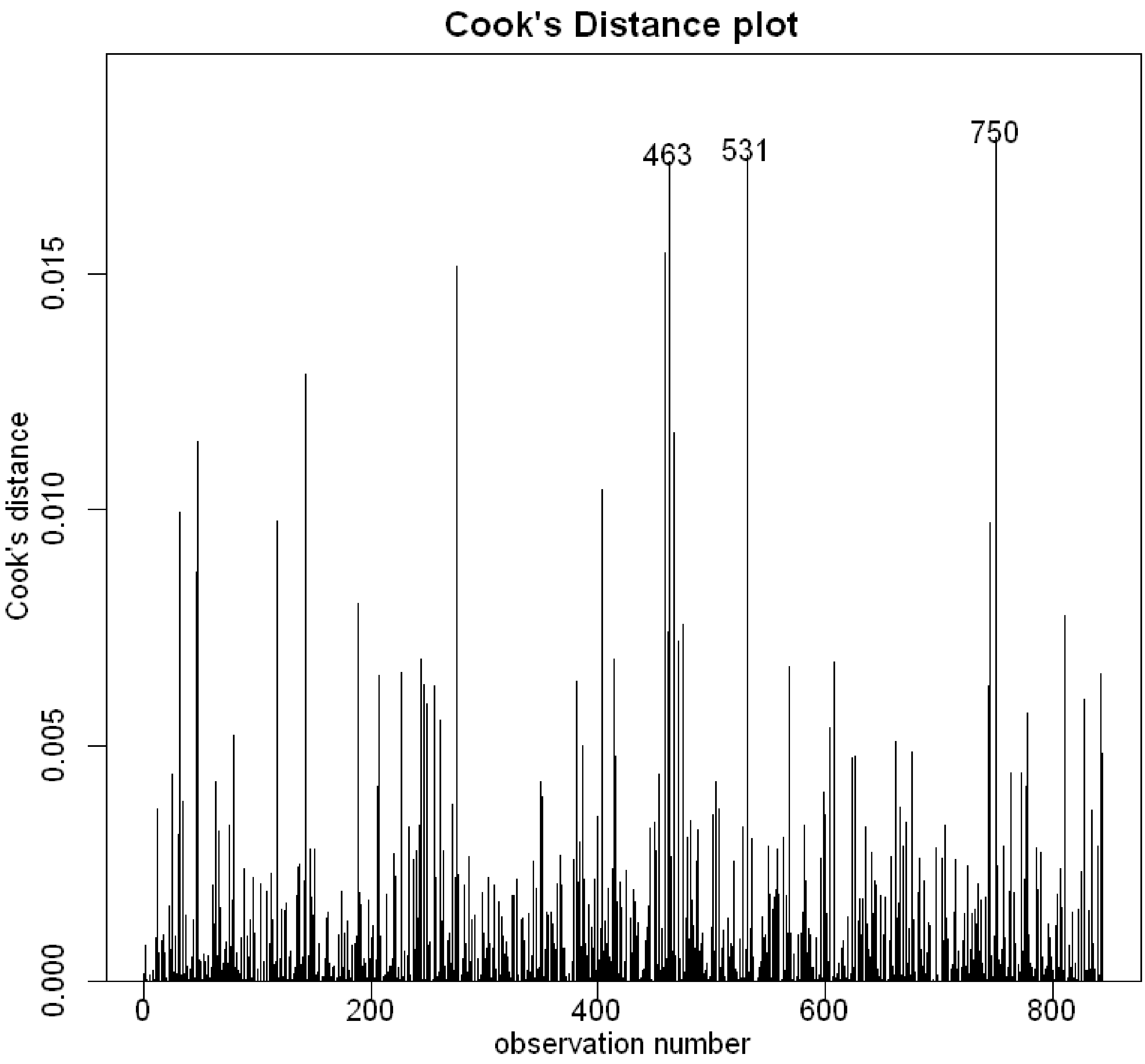
Skip to main content



Skip to main content



Skip to main content

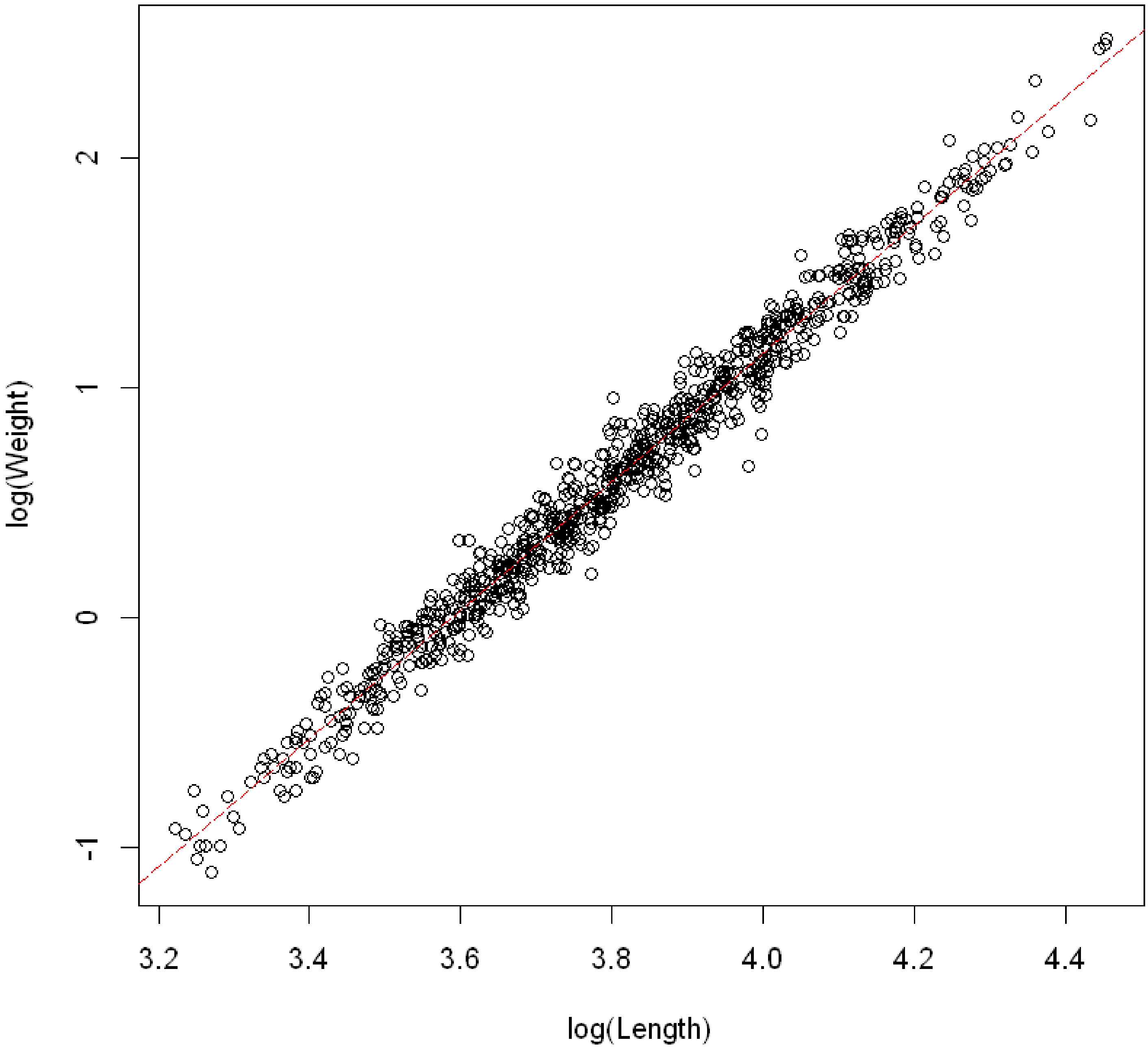


```
summary(Snap.lm)
plot(log(wgt) ~ log(len), data = Snap.df, xlab = "log(Length)", ylab = "log(Weight)")
abline(coef(Snap.lm), lty = 5, col = "red")
```

Skip to main content

```
Call:
lm(formula = log(wgt) ~ log(len), data = Snap.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.44099 -0.06853  0.00234  0.06942  0.36139 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -10.01416    0.05602  -178.7   <2e-16 ***
log(len)      2.79104    0.01469   190.0   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1012 on 842 degrees of freedom
Multiple R-squared:  0.9772,    Adjusted R-squared:  0.9772 
F-statistic: 3.609e+04 on 1 and 842 DF,  p-value: < 2.2e-16
```

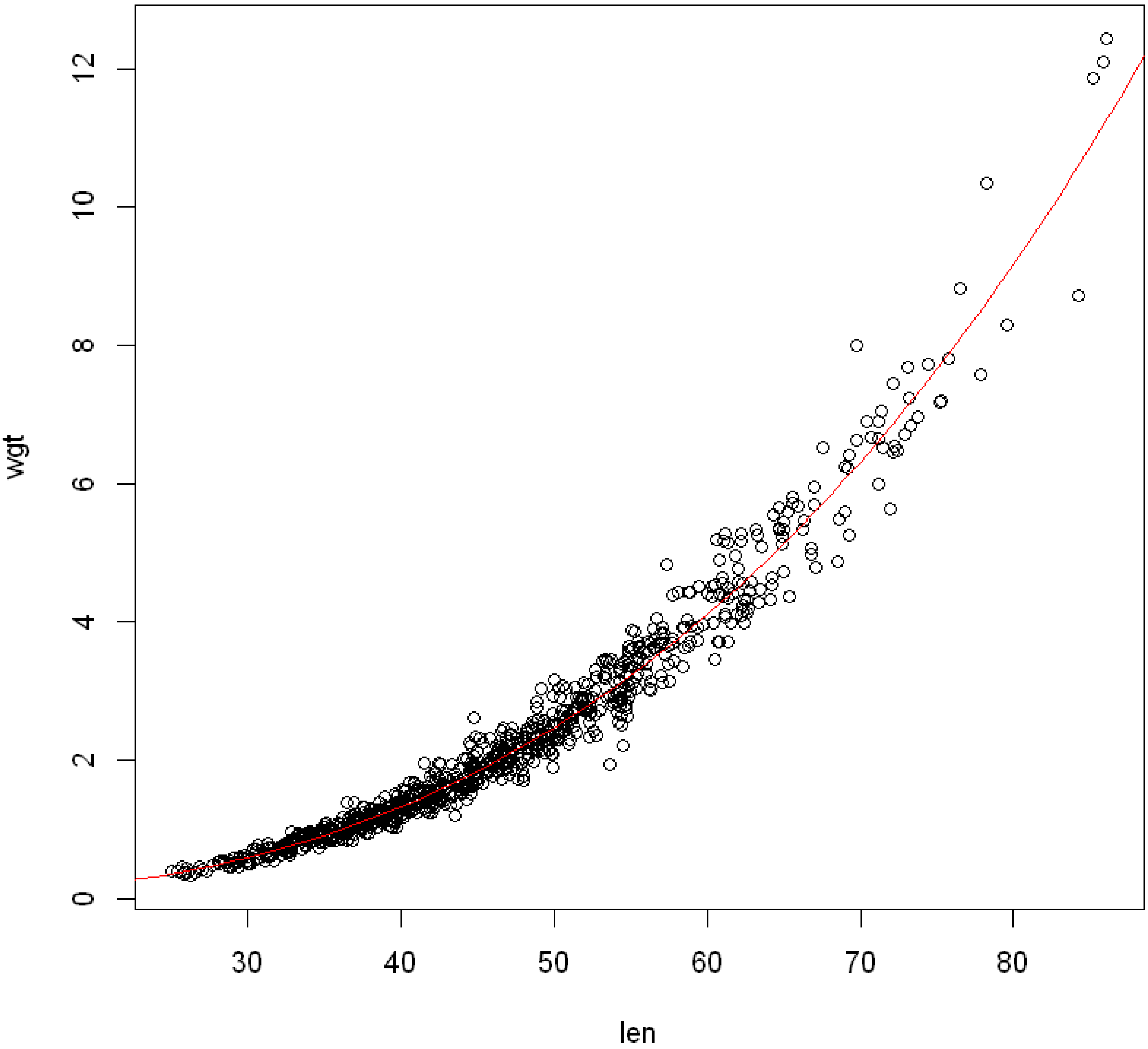
Skip to main content



Let us redo the plot on the raw scale (rather than log scale):

```
plot(wgt ~ len, data = Snap.df)
pred.df <- data.frame(len = 20:90)
Snap.pred <- exp(predict(Snap.lm, pred.df))
lines(pred.df$len, Snap.pred, col = "red")
```

Skip to main content



```
Pred.df <- data.frame(len = 30)
exp(predict(Snap.lm, Pred.df, interval = "confidence"))
exp(predict(Snap.lm, Pred.df, interval = "prediction"))
```

Skip to main content

A matrix: 1 × 3 of type dbl

	fit	lwr	upr
1	0.5937602	0.5857844	0.6018445

A matrix: 1 × 3 of type dbl

	fit	lwr	upr
1	0.5937602	0.4865954	0.7245262

A few slides earlier we deduced(推断) that the power coefficient(系数) should be β1 close to, though not necessarily equal to 3.

Let us examine this formally by testing the null hypothesis $H_0 : \beta_1 = 3$.

Question 1: Is this hypothesis rejected at the 5% level? (Hint: the answer can be worked out from output already seen)

Question 2: What is the P-value for $H_0 : \beta_1 = 3$? (This takes a bit more work)

8. Linear models with both numeric and factor explanatory variables

本节需要的包：

```r
require(s20x)
```

▶ Show code cell output

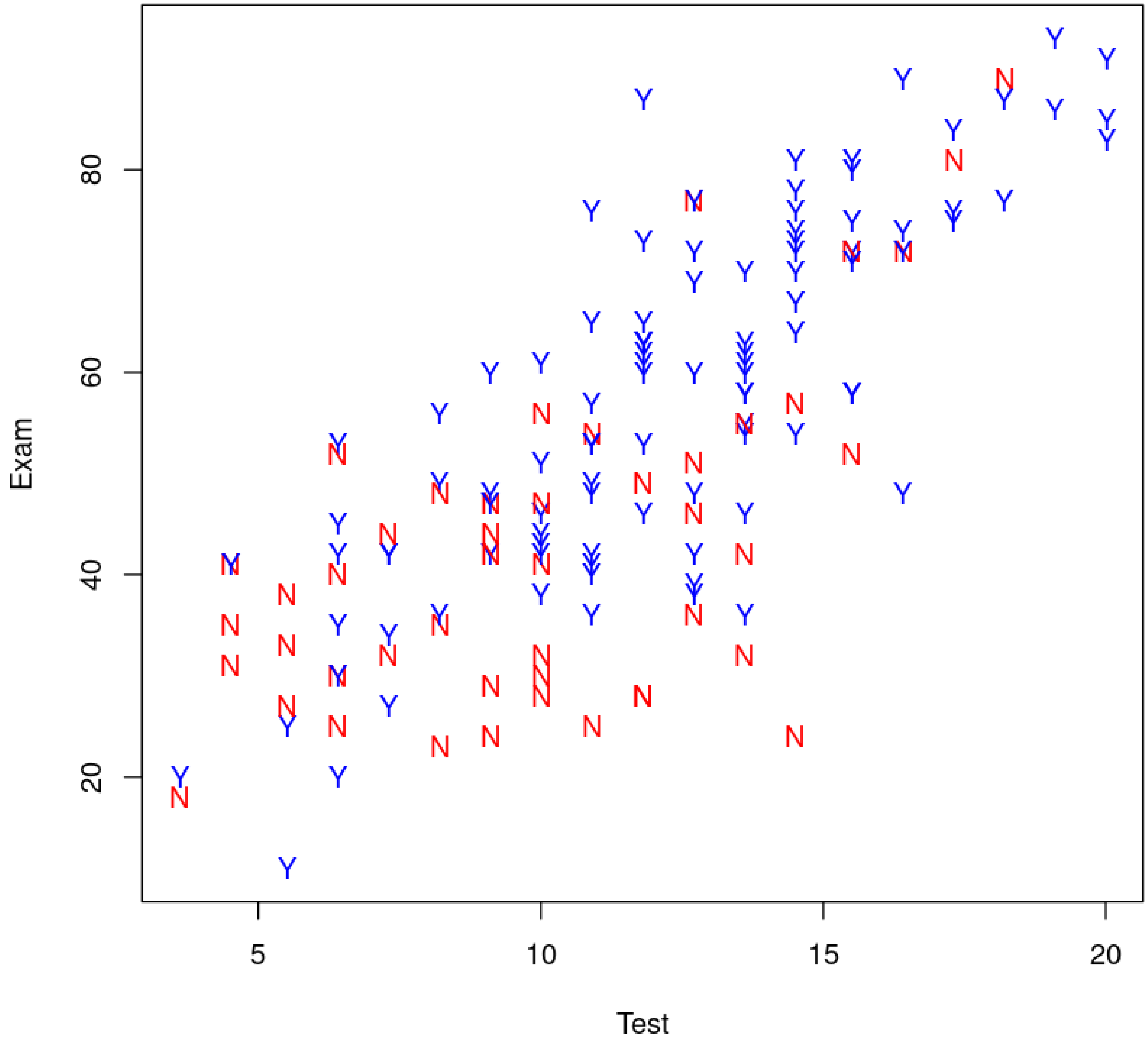
8.1. Example: Using both test score and attendance to explain exam score

示例：使用测试成绩和出勤率解释考试分数

Skip to main content

```r
## Invoke the s20x library
library(s20x)
## Importing data into R
Stats20x.df <- read.table("../data/STATS20x.txt", header = TRUE)
Stats20x.df$Attend <- as.factor(Stats20x.df$Attend)
## Plot blue "Y" for "Yes" (regular attenders), and red "N" for "No"
plot(Exam ~ Test,
    data = Stats20x.df,
    pch = substr(Attend, 1, 1), # "Y" or "N"
    col = ifelse(Attend == "Yes", "blue", "red")
)
```

Skip to main content

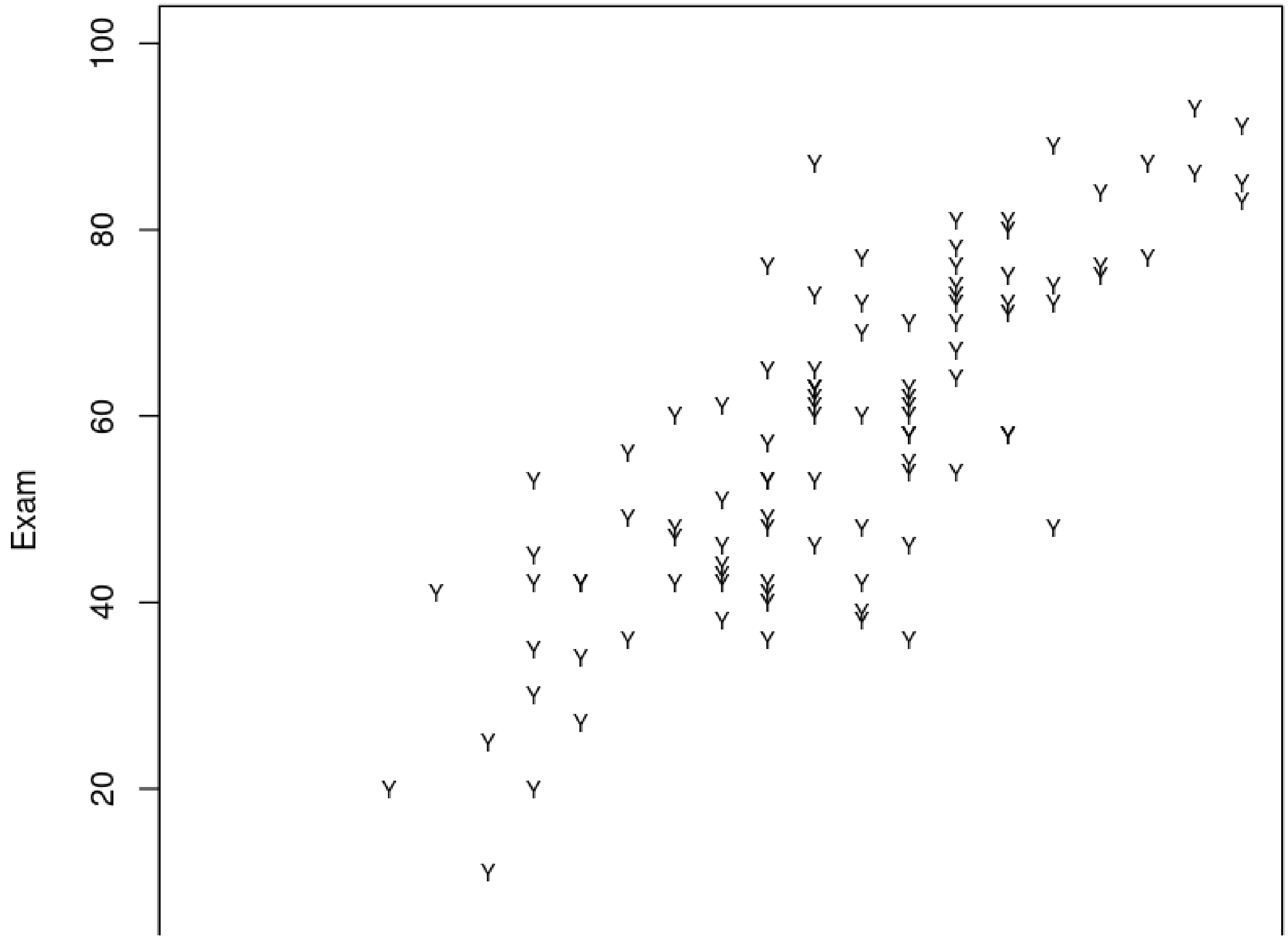


Here is the plot for the regular attenders and the non-attenders.

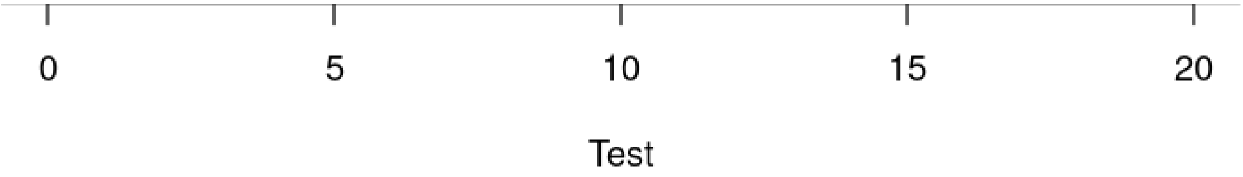
Skip to main content

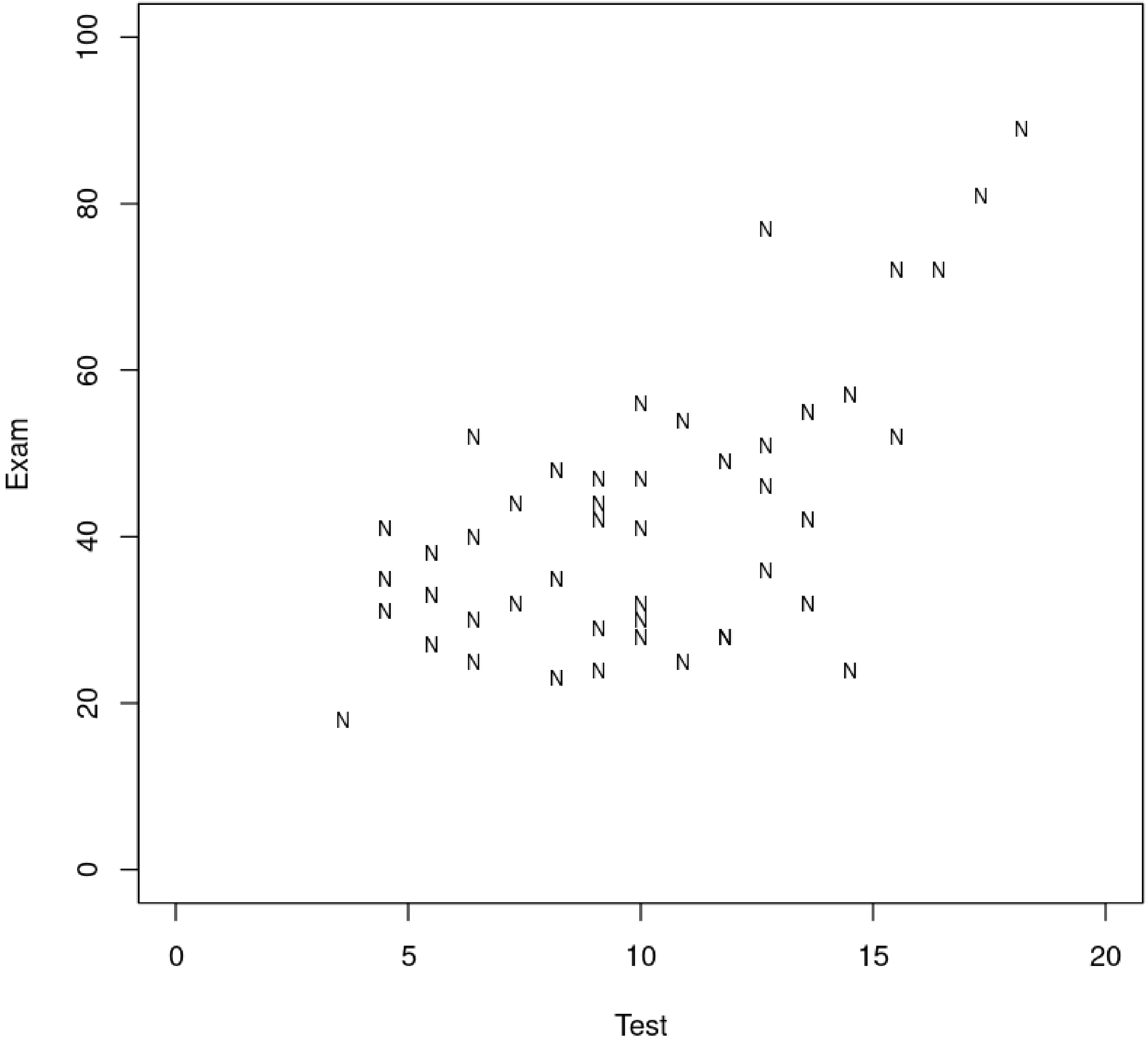
```
Attendees.df <- subset(Stats20x.df, Attend == "Yes")
plot(Exam ~ Test,
    data = Attendees.df,
    xlim = c(0, 20),
    ylim = c(0, 100),
    pch = "Y", cex = 0.7
)
Absentees.df <- subset(Stats20x.df, Attend == "No")
plot(Exam ~ Test,
    data = Absentees.df,
    xlim = c(0, 20),
    ylim = c(0, 100),
    pch = "N",
    cex = 0.7
)
```

Skip to main content



Skip to main content





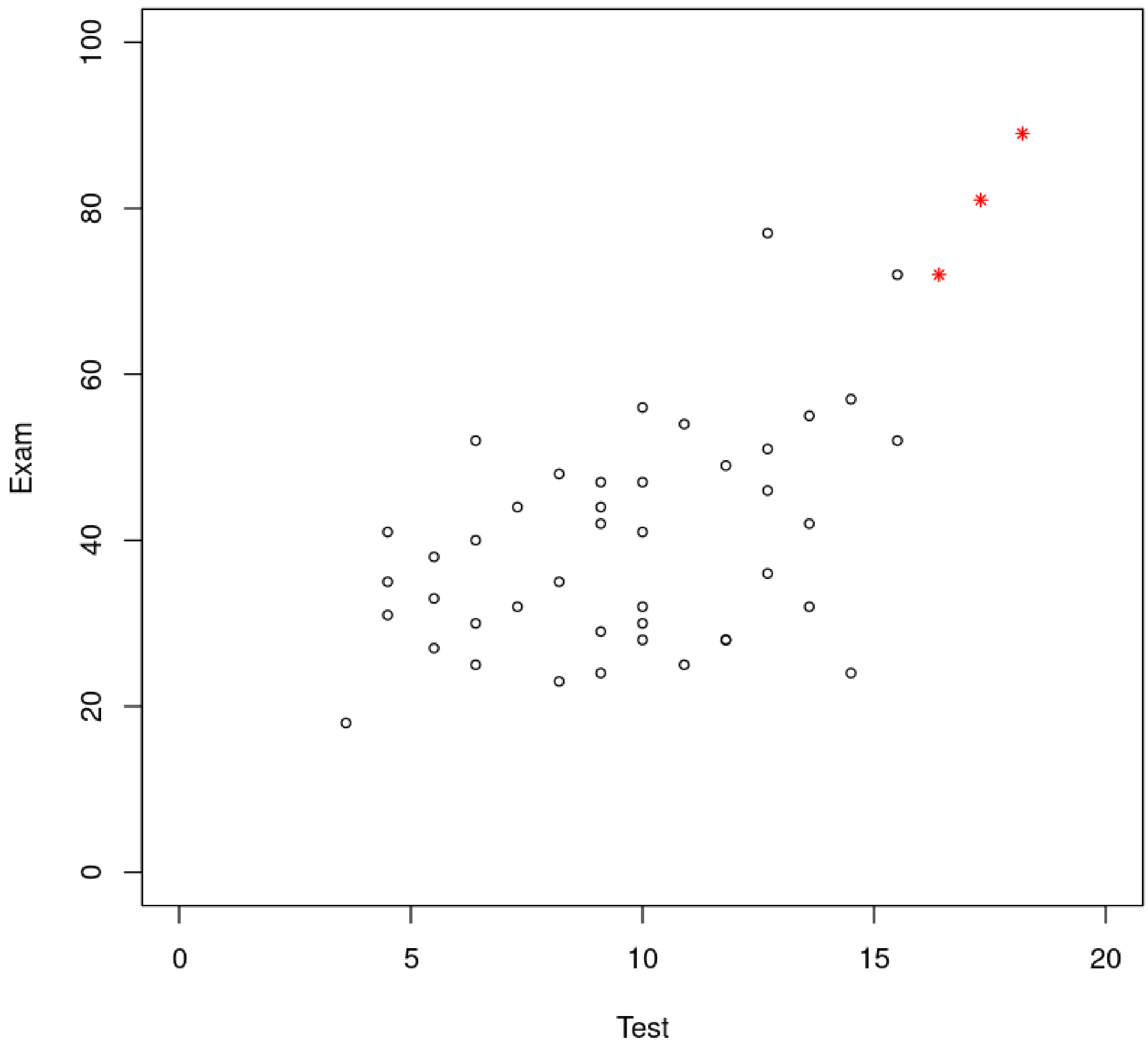
Also, there seems to be some non-attenders who do well in the test and exam so we could (and will) see whether we should include these people. They are identified in red (stars) with the code below. 此外，似

Skip to main content

乎有一些没有参加考试的人在考试和考试中表现很好，所以我们可以(也将)看看是否应该包括这些人。它们用红色(星号)标识，代码如下。

```
Absentees.df <- subset(Stats20x.df, Attend == "No")
plot(Exam ~ Test,
    data = Absentees.df, xlim = c(0, 20), ylim = c(0, 100),
    cex = 0.7, col = ifelse(Absentees.df$Test <= 16, "black", "red"),
    pch = ifelse(Absentees.df$Test <= 16, 1, 8)
)
```

Skip to main content



8.2. Fitting the linear model

看起来我们需要符合两个不同取决于学生是否经常出席者。

We will call our indicator variable for greater convenience of notation: 我们将为了方便表示法而称之为指示变量

Skip to main content

```
## Boolean statement if Attend ="Yes" (TRUE) D=1, othwerwise 0 (FALSE);
Stats20x.df$D <- as.numeric(Stats20x.df$Attend == "Yes")
table(Stats20x.df$Attend, Stats20x.df$D) ## Check it is okay
```

```
       0   1
No    46   0
Yes    0 100
```

相当于说，No 为 0，Yes 为 1。

Our straight line model for the non-attenders (i.e., D = 0) students will be:

$$Exam = \beta_0 + \beta_1 \times Test + \varepsilon \; where \; \varepsilon \sim N(0, \sigma^2)$$

For the non-attenders (D = 0) the slope(斜率) is:

$$\beta_1 + D \times \beta_2 = \beta_1$$

For the attenders (D = 1) the slope is:

$$\beta_1 + D \times \beta_3 = \beta_1 + \beta_3$$

So, our model is:

$$\begin{aligned} \text{Exam} &= \beta_0 + \beta_2 \times D + (\beta_1 + \beta_3 \times D)\text{Test} + \varepsilon \\ &= (\beta_0 + \beta_2 \times D) + (\beta_1 + \beta_3 \times D)\text{Test} + \varepsilon \\ &= \beta_0 + \beta_1 \times \text{Test} + \beta_2 \times D + \beta_3 \times D \times \text{Test} + \varepsilon \end{aligned}$$

建模：

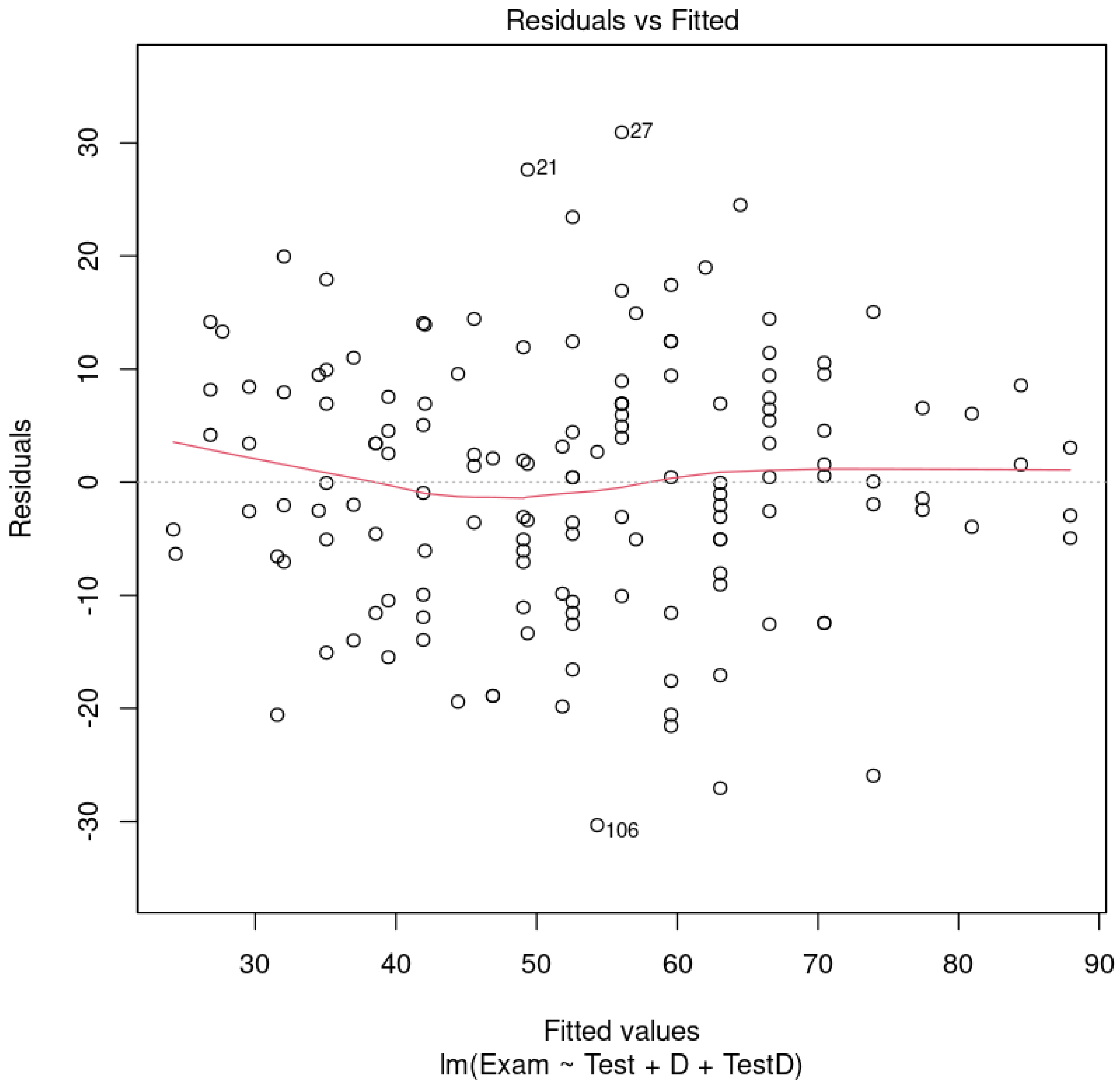
```
Stats20x.df$TestD <- with(Stats20x.df, {
    TestD <- D * Test
})
TestAttend.fit <- lm(Exam ~ Test + D + TestD, data = Stats20x.df)
```

Assumption checks 三步走:

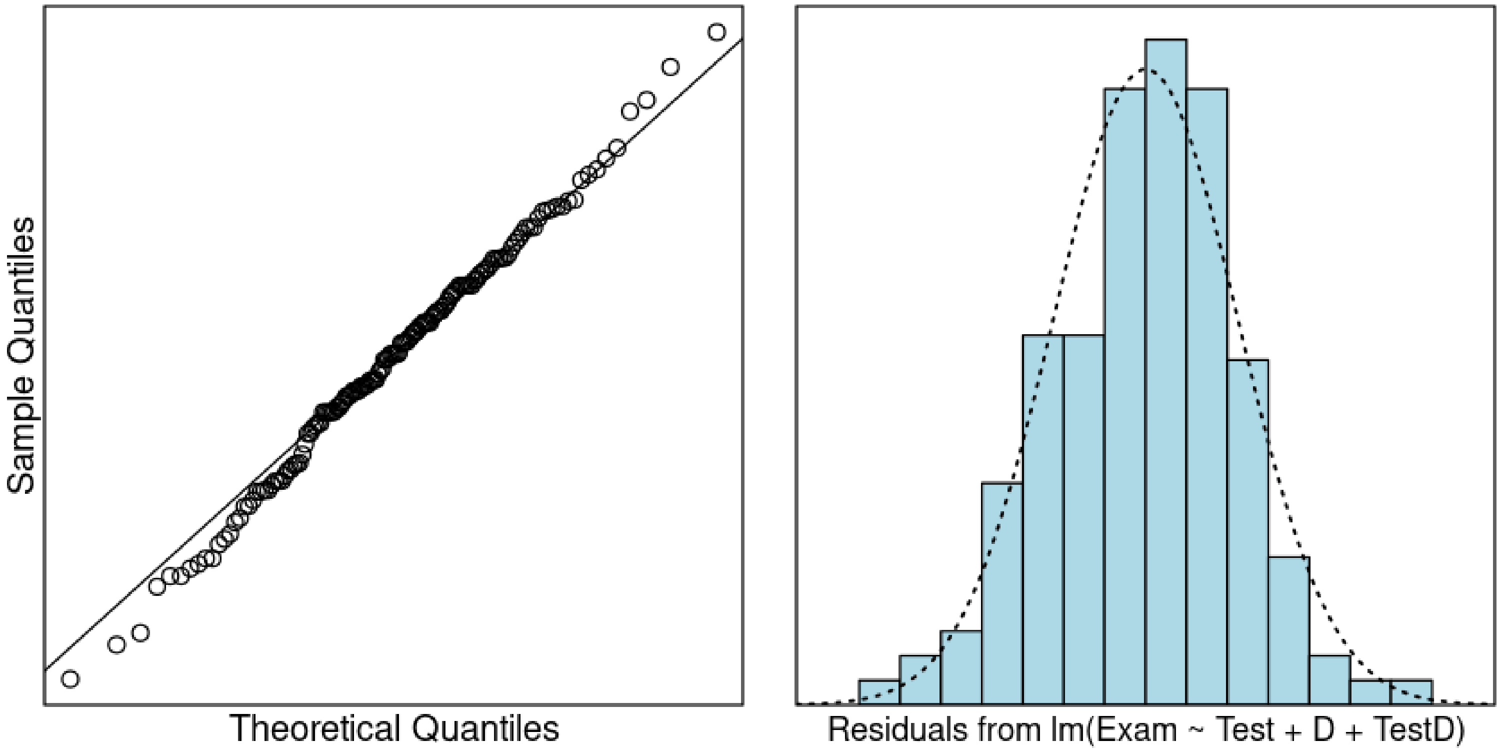
Skip to main content

```r
plot(TestAttend.fit, which = 1)
normcheck(TestAttend.fit)
cooks20x(TestAttend.fit)
```

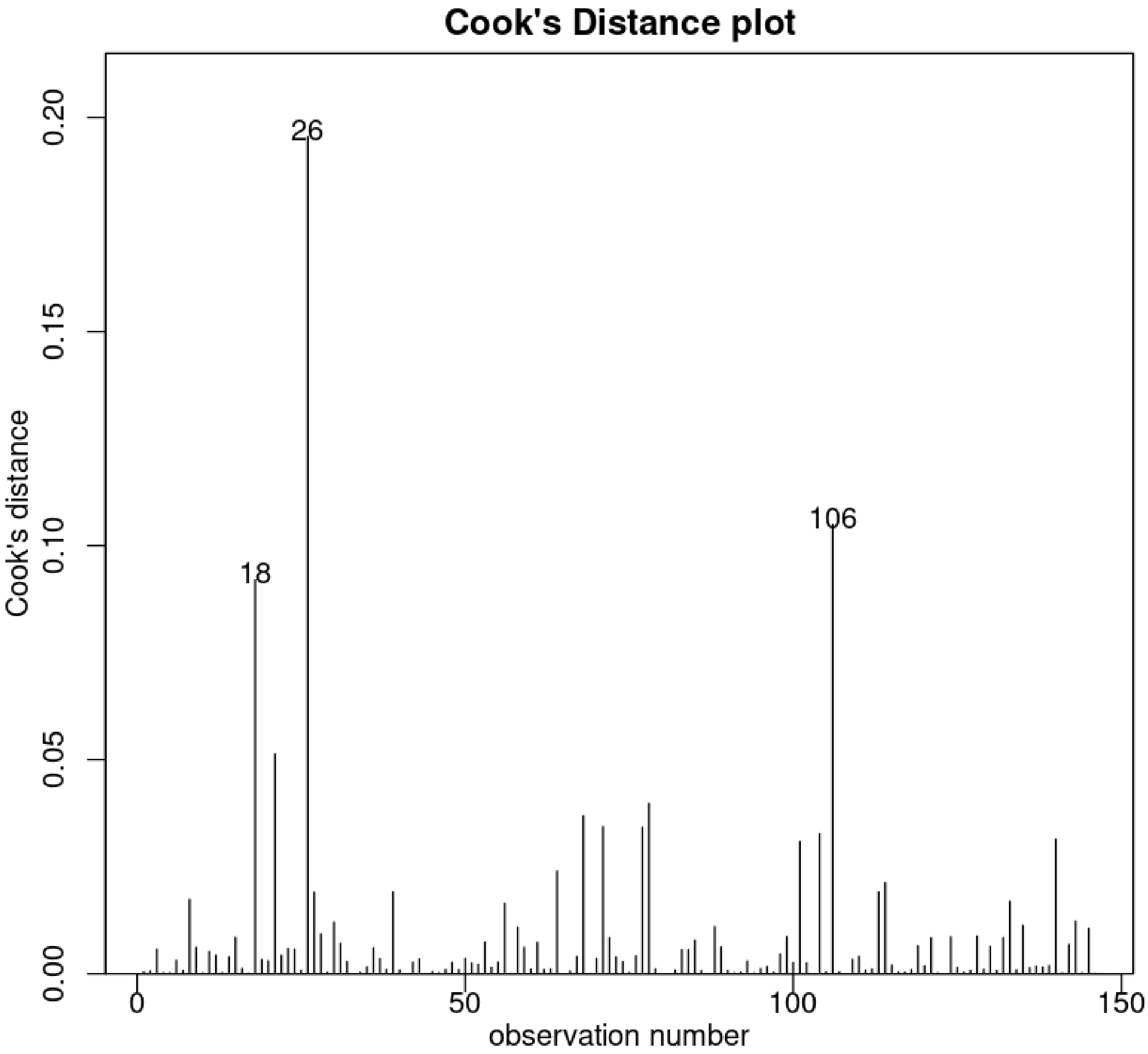
Skip to main content



Skip to main content



Skip to main content



We can now trust the fitted . The summary output is:

```
summary(TestAttend.fit)
```

Skip to main content

```
Call:
lm(formula = Exam ~ Test + D + TestD, data = Stats20x.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-30.3155  -6.5139   0.4383   7.3166  30.9383 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  14.4467     4.9443   2.922  0.00405 ** 
Test          2.7496     0.4603   5.973 1.78e-08 ***
D            -4.2582     6.3723  -0.668  0.50506    
TestD         1.1380     0.5577   2.040  0.04316 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.41 on 142 degrees of freedom
Multiple R-squared:  0.6347,	Adjusted R-squared:  0.627 
F-statistic: 82.25 on 3 and 142 DF,  p-value: < 2.2e-16
```

Note that the above Executive Summary is missing the confidence interval for the effect of test mark on attenders. To obtain this CI we need to change attenders to the baseline level of . 改变了基准线

让我们仔细看看我们刚刚安装模式。我们将会产生一个单独的情节为每个出席者。

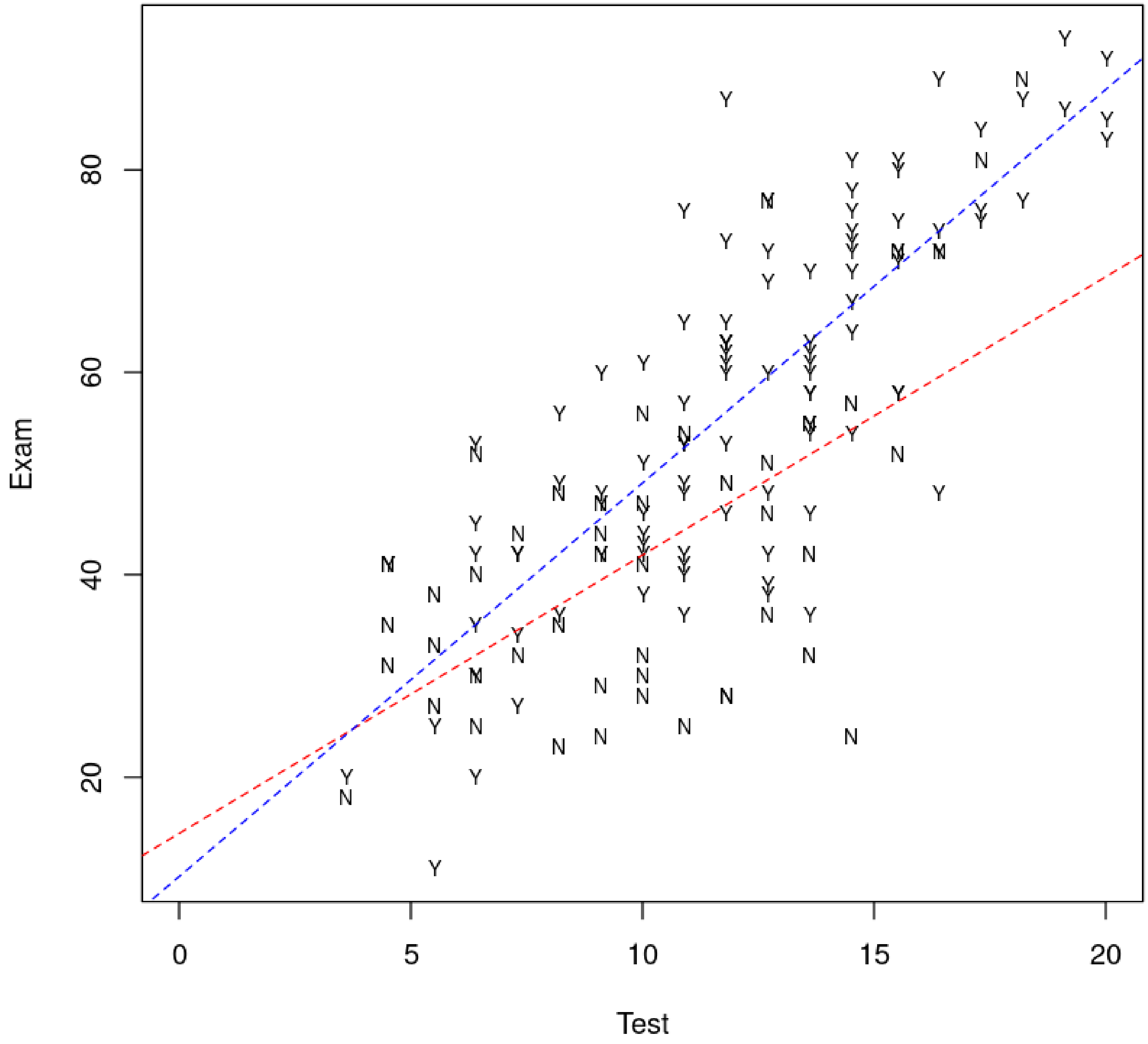
```
coef(TestAttend.fit)[1:2]
# 分别为截距 和 Test 的系数
```

(Intercept): 14.4467499989379 **Test:** 2.74956844608019

将我们的模型放到图上：

```
## Plot these data all together
b <- coef(TestAttend.fit) # easier to work with these terms
plot(Exam ~ Test,
    data = Stats20x.df,
    pch = substr(Attend, 1, 1), # "Y" or "N"
    cex = 0.7, # 缩放，默认为 1
    xlim = c(0, 20) # x 轴范围
)
## Red for "No" and blue for "Yes".
abline(b[1:2], lty = 2, col = "red")
# No 群体：beta0 做截距，beta1 做斜率
abline(b[1] + b[3], b[2] + b[4], lty = 2, col = "blue")
# Yes群体：beta0 + beta2 做截距，beta1 + beta3 做斜率
```

Skip to main content



All that hard work we did with constructing and can be avoided D TestD since will automatically do this for us.

We were interested to see whether the effect of interacts with the Test students variable. Using we simply specify to Attend lm Test * Attend fit the model with interaction. That is,

Skip to main content

```
TestAttend.fit2 = lm(Exam ~ Test * Attend, data = Stats20x.df)
summary(TestAttend.fit2)
```

```
Call:
lm(formula = Exam ~ Test * Attend, data = Stats20x.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-30.3155  -6.5139   0.4383   7.3166  30.9383 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     14.4467     4.9443   2.922  0.00405 ** 
Test             2.7496     0.4603   5.973 1.78e-08 ***
AttendYes       -4.2582     6.3723  -0.668  0.50506    
Test:AttendYes   1.1380     0.5577   2.040  0.04316 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.41 on 142 degrees of freedom
Multiple R-squared:  0.6347,	Adjusted R-squared:  0.627 
F-statistic: 82.25 on 3 and 142 DF,  p-value: < 2.2e-16
```

We have the same outputs, but with slightly different names.

注意: `Test * Attend` 是速记符号。你可以用下边的写法更明确关于模型中各个方面：

```
TestAttend.fit2 <- lm(Exam ~ Test + Attend + Test:Attend, data = Stats20x.df)
```

8.3. Interpretting the fitted model

We see that our intuition was correct. That is, the slope for of Test attenders is greater that for non-attenders. This is because the estimate of the difference in these slopes `TestD`.

```
coef(TestAttend.fit2)[4]
```

Test:AttendYes: 1.13799041829866

Confidence intervals may be needed for the coefficients:

Skip to main content

A matrix: 4 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	4.67287511	24.220625
Test	1.83956971	3.659567
AttendYes	-16.85506294	8.338572
Test:AttendYes	0.03547053	2.240510

注：Statistical significance (at the 5% level) of a coefficient is NOT equivalent to the (95%) confidence interval containing zero. 统计学意义(5%)的系数并不等同于(95%)置信区间包含零。

Some predictions:

```
predTestAttend.df <- data.frame(
    Test = c(0, 10, 10, 20),
    Attend = factor(c("No", "No", "Yes", "Yes"))
)
predTestAttend.df
```

A data.frame: 4 × 2

Test	Attend
<dbl>	<fct>
0	No
10	No
10	Yes
20	Yes

Let us estimate the expected exam scores for these values of test score and attendance 让我们这些值的估计预期的考试分数测试成绩和出勤:

```
predict(TestAttend.fit2, predTestAttend.df, interval = "confidence") # 区间估计
predict(TestAttend.fit2, predTestAttend.df, interval = "prediction") # 点估计
```

Skip to main content

A matrix: 4 × 3 of type dbl

	fit	lwr	upr
1	14.44675	4.672875	24.22062
2	41.94243	38.616376	45.26849
3	49.06409	46.412194	51.71599
4	87.93968	82.610100	93.26926

A matrix: 4 × 3 of type dbl

	fit	lwr	upr
1	14.44675	-10.13028	39.02378
2	41.94243	19.14848	64.73639
3	49.06409	26.35871	71.76947
4	87.93968	64.76845	111.11092

This is not the best model for predicting individual student exam scores as the intervals are too wide and in some case are meaningless (at the extremes of `Test`). 这不是最好的模型预测个体学生的考试成绩作为间隔太宽,在某些情况下是毫无意义的(在极端的 `Test` 成绩下)。

请注意，上述的执行摘要缺少有关测试成绩对参与者影响的置信区间。要获得此置信区间，我们需要将参与者更改为基线水平，即 Atten。

```r
Stats20x.df$Attend2 <- relevel(Stats20x.df$Attend, ref = "Yes")
TestAttend.fit2b <- lm(Exam ~ Test * Attend2, data = Stats20x.df)
coef(summary(TestAttend.fit2b))
confint(TestAttend.fit2b)
```

Skip to main content

A matrix: 4 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	10.188504	4.0199956	2.5344566	1.234648e-02
Test	3.887559	0.3148792	12.3461898	3.020895e-24
Attend2No	4.258246	6.3722922	0.6682439	5.050626e-01
Test:Attend2No	-1.137990	0.5577265	-2.0404094	4.316270e-02

A matrix: 4 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	2.241733	18.13527583
Test	3.265102	4.51001561
Attend2No	-8.338572	16.85506294
Test:Attend2No	-2.240510	-0.03547053

We estimate that each additional test mark will increase the expected exam mark of an attender by 3.3 to 4.5. 我们估计，每增加一个测试分数，参与者的预期考试成绩将增加3.3到4.5分。

8.4. Assessing influence of the atypical students

在分析的探索性阶段，我们确定了三名可能异常的学生。回想一下，这些是那3个在测试中得分大于16分但没有参加考试的学生。

如果我们有理由认为这些学生“不典型”（我们有吗？），那么在分析时不考虑他们可能是合理的。

```r
## Remove atypical points - Note that ! means ’not’
Subset.df <- subset(Stats20x.df, !(Test > 16 & Attend == "No"))
TestAttend.fit3 <- lm(Exam ~ Test * Attend, data = Subset.df)
summary(TestAttend.fit3)
```

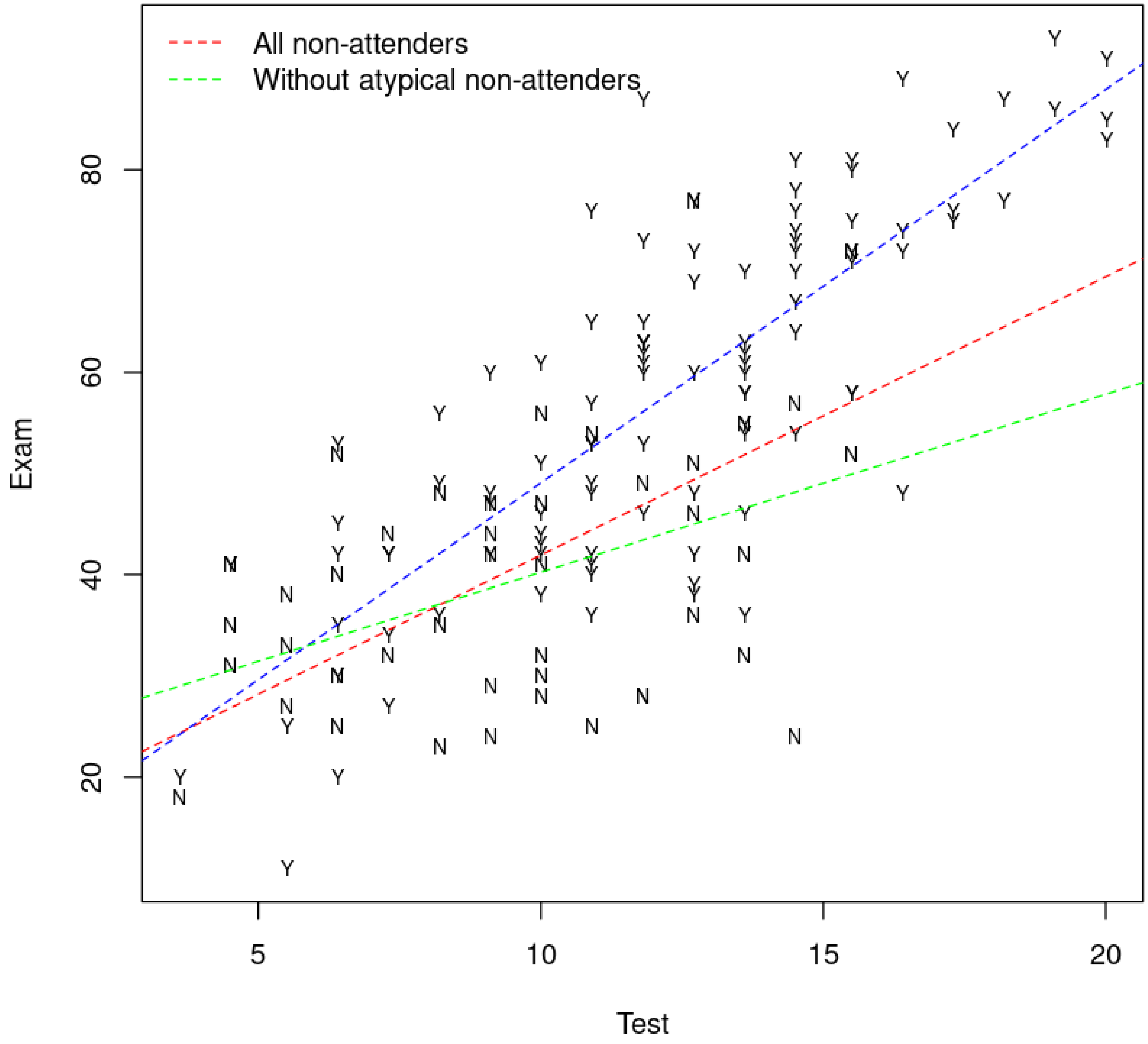
Skip to main content

```
Call:
lm(formula = Exam ~ Test * Attend, data = Subset.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-27.059  -6.817   0.439   6.938  32.008 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     22.6522     5.2776   4.292  3.3e-05 ***
Test             1.7590     0.5213   3.374 0.000960 ***
AttendYes      -12.4637     6.5505  -1.903 0.059146 .  
Test:AttendYes   2.1285     0.6034   3.527 0.000569 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.01 on 139 degrees of freedom
Multiple R-squared:  0.6495,    Adjusted R-squared:  0.6419 
F-statistic: 85.86 on 3 and 139 DF,  p-value: < 2.2e-16
```

R code to generate the plot of the full data with all three of the fitted Rlines:

```r
## Plot these data all together
plot(Exam ~ Test, data = Subset.df, pch = substr(Attend, 1, 1), cex = 0.7)

## Remember that we’ve defined b in Slide 26
## Each abline() will have a different colour
abline(b[1:2], lty = 2, col = "red")
abline(b[1] + b[3], b[2] + b[4], lty = 2, col = "blue")

## The fitted line without the 3 atypical points
b2 <- coef(TestAttend.fit3) ## Easier to work with these terms
abline(b2[1:2], lty = 2, col = "green")

## Add a legend to help us differentiate between the lines for non-attenders
legend("topleft",
    legend = c("All non-attenders", "Without atypical non-attenders"),
    lty = 2,
    col = c("red", "green"),
    bty = "n"
)
```

Skip to main content



9. Linear models with both numeric and factor explanatory variables without interaction

本节需要的包：

Skip to main content

```
require(s20x)
```

▶ Show code cell output

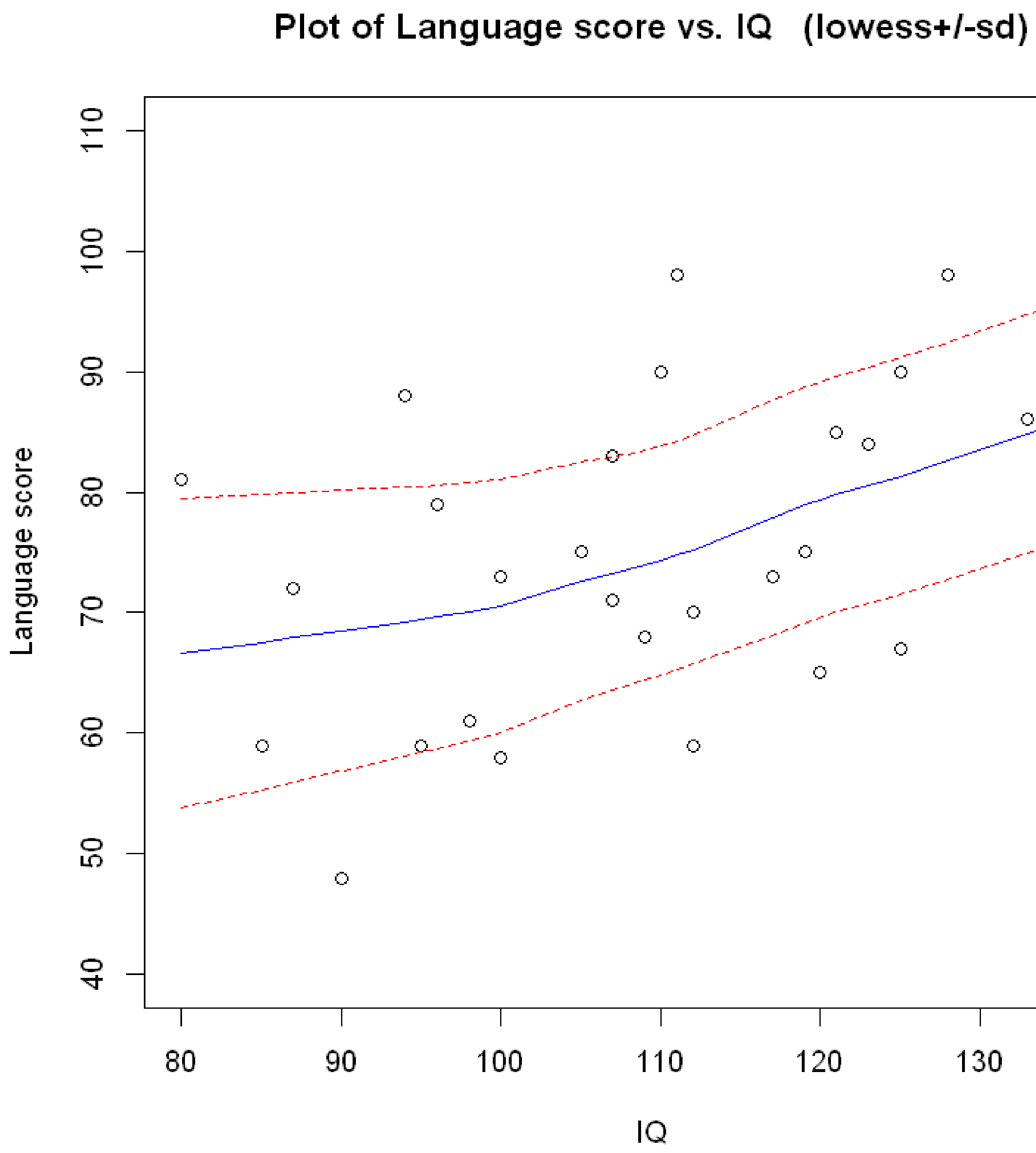
9.1. Using both IQ and teaching method to explain increase in language proficiency

在以下示例中，教育专家对三种不同的教学方法中哪一种最有效地提高语言成绩感兴趣——这是通过智商测量的。为了做到这一点，30名学生被随机分配到三个组中，并使用不同的教学方法进行教学。每个学生的智商在教学计划开始之前进行了测量。这种随机化是为了确保每个组中都代表了一定范围的学生能力。由于学生处于测试环境中，我们可以假设他们的测试成绩互相独立。

As usual, we begin by inspecting the data:

```
## Invoke the s20x library
library(s20x)
## Importing data found in the s20x library into R
data(teach.df)
## Plot the data with trendscatter()
trendscatter(lang ~ IQ,
    f = 0.8, ylim = c(40, 110),
    data = teach.df,
    ylab = "Language score"
)
## Note that f is the proportion of points in the plot which influence the
## smooth at each value. Larger values of f give more smoothness!
```

Skip to main content



嗯，智商与语言成绩呈正相关，但统计显著性并不明显。一个显示教学方法的图表可能更有用。

In dataframe `teach.df` the `method` is recorded as a number, 1, 2 or 3:

```
teach.df$method
class(teach.df$method)
```

Skip to main content

```
1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 3 · 3 · 3 · 3 · 3 · 3 · 3 · 3 · 3 · 3
'integer'
```

然而，这些只是标签，也可以是“A”、“B”或“C”。因此，需要将其强制转换为因子，以使 `method` 不被视为数值变量。

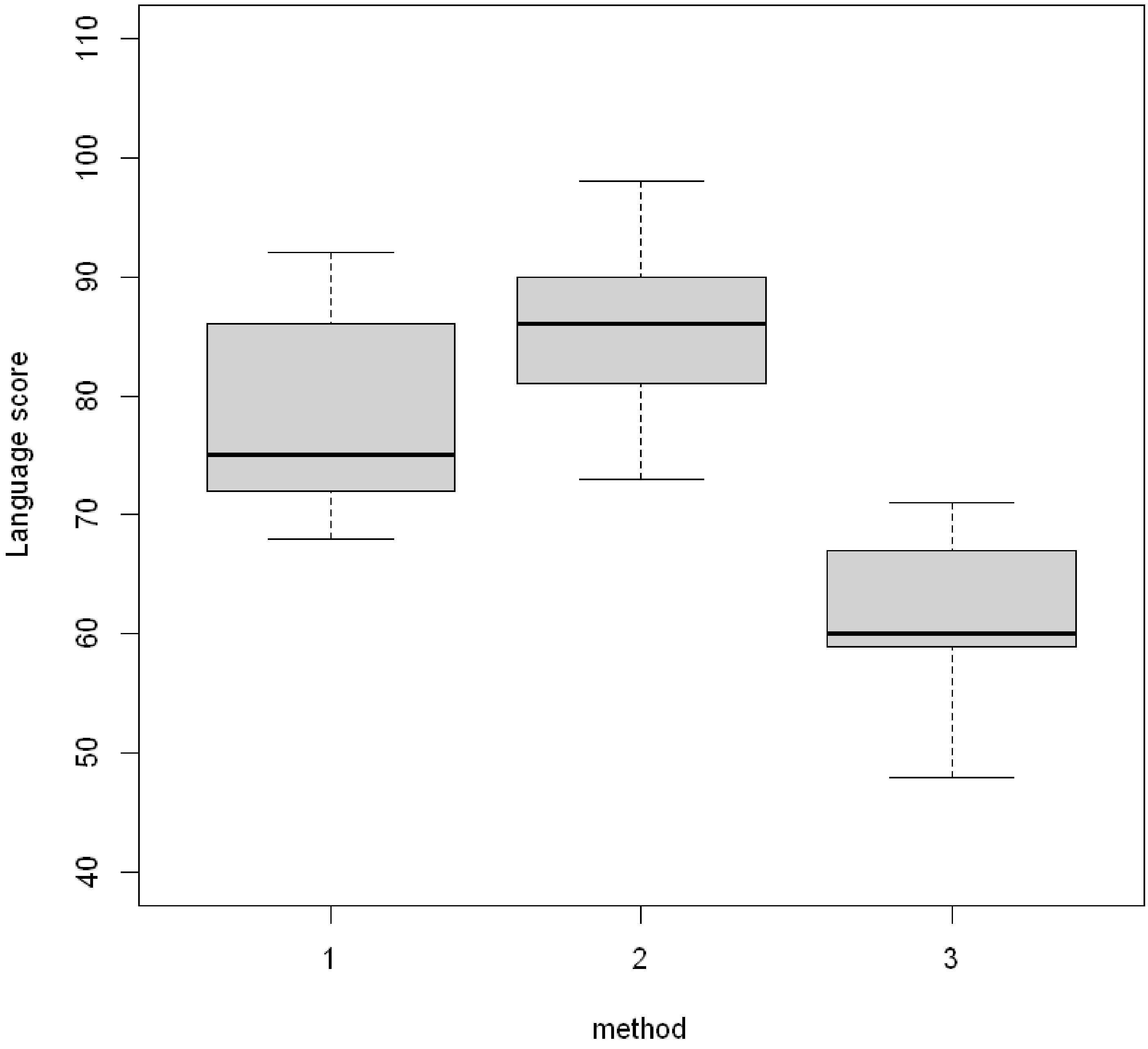
```r
teach.df$method <- factor(teach.df$method)
teach.df$method
class(teach.df$method)
```

```
1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 1 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 2 · 3 · 3 · 3 · 3 · 3 · 3 · 3 · 3 · 3 · 3
▶ Levels:
'factor'
```

Student language score by teaching method and IQ:

```r
plot(lang ~ method, ylim = c(40, 110), data = teach.df, ylab = "Language score")
```

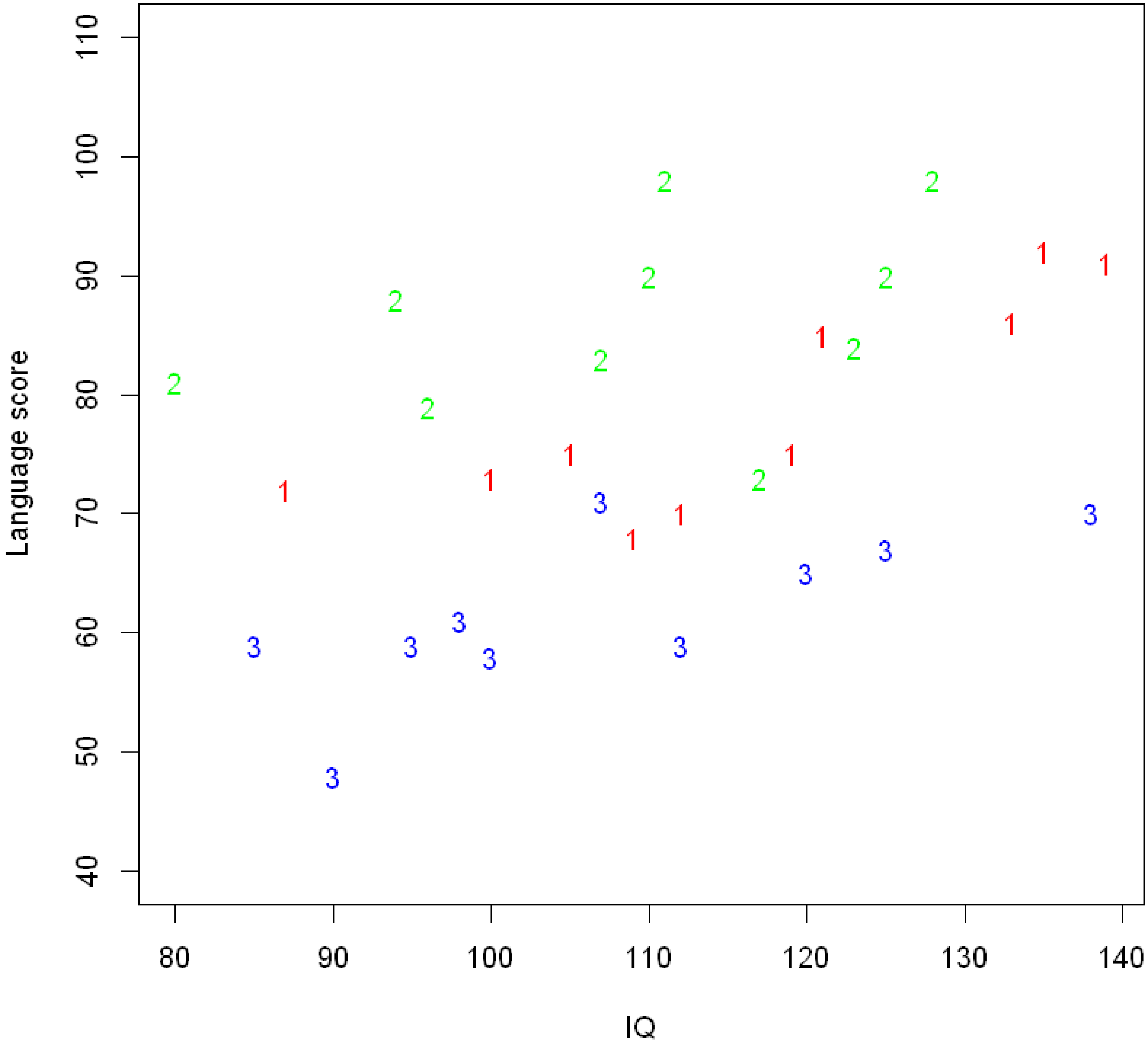
Skip to main content



A more useful plot:

```r
plot(
    lang ~ IQ,
    ylim = c(40, 110),
    pch = as.character(method),
    col = c("red", "green", "blue")[method],
    data = teach.df,
    ylab = "Language score"
)
```

Skip to main content

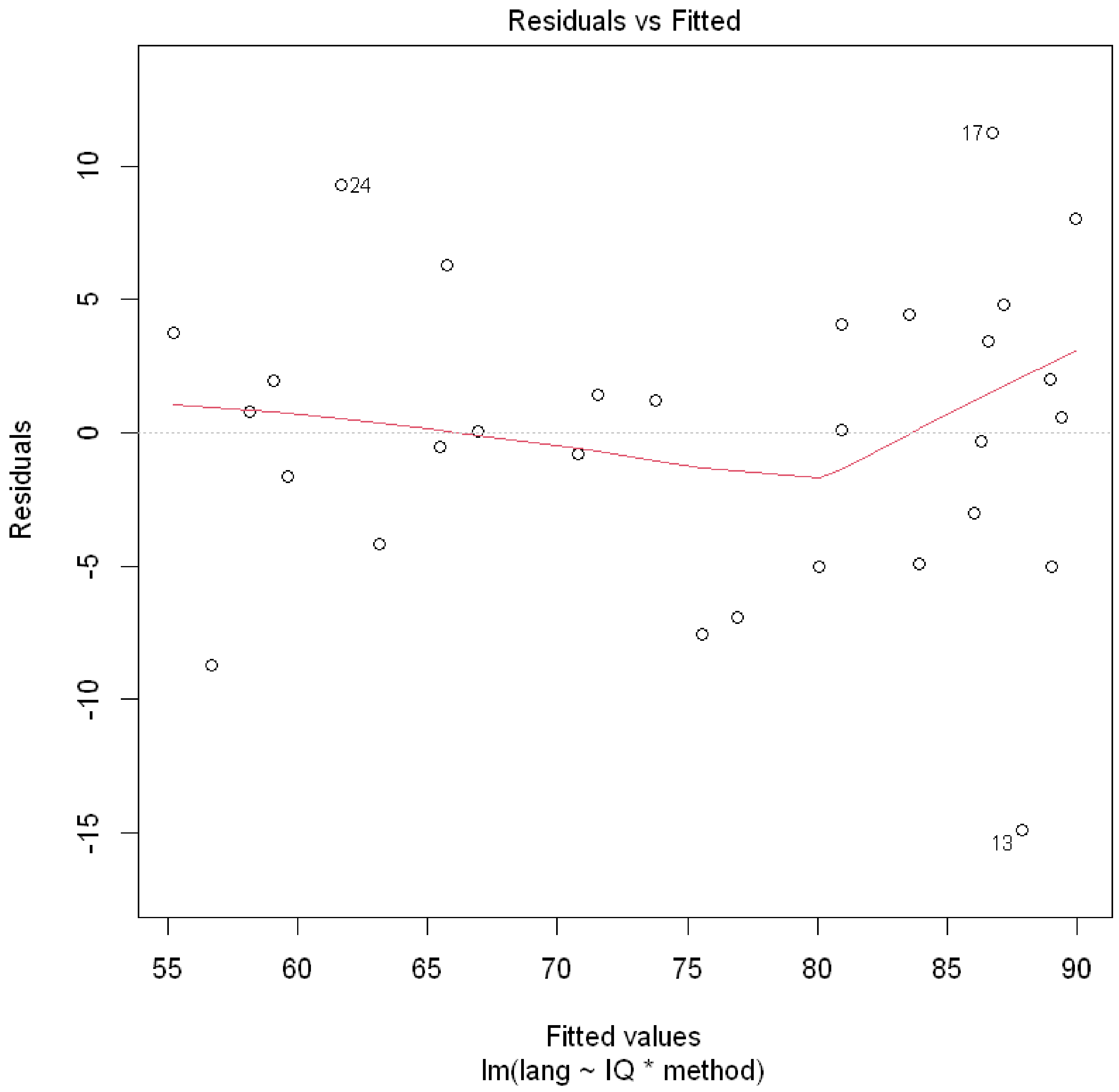


We will fit the model with interaction first, anticipating that the interaction will not be significant.

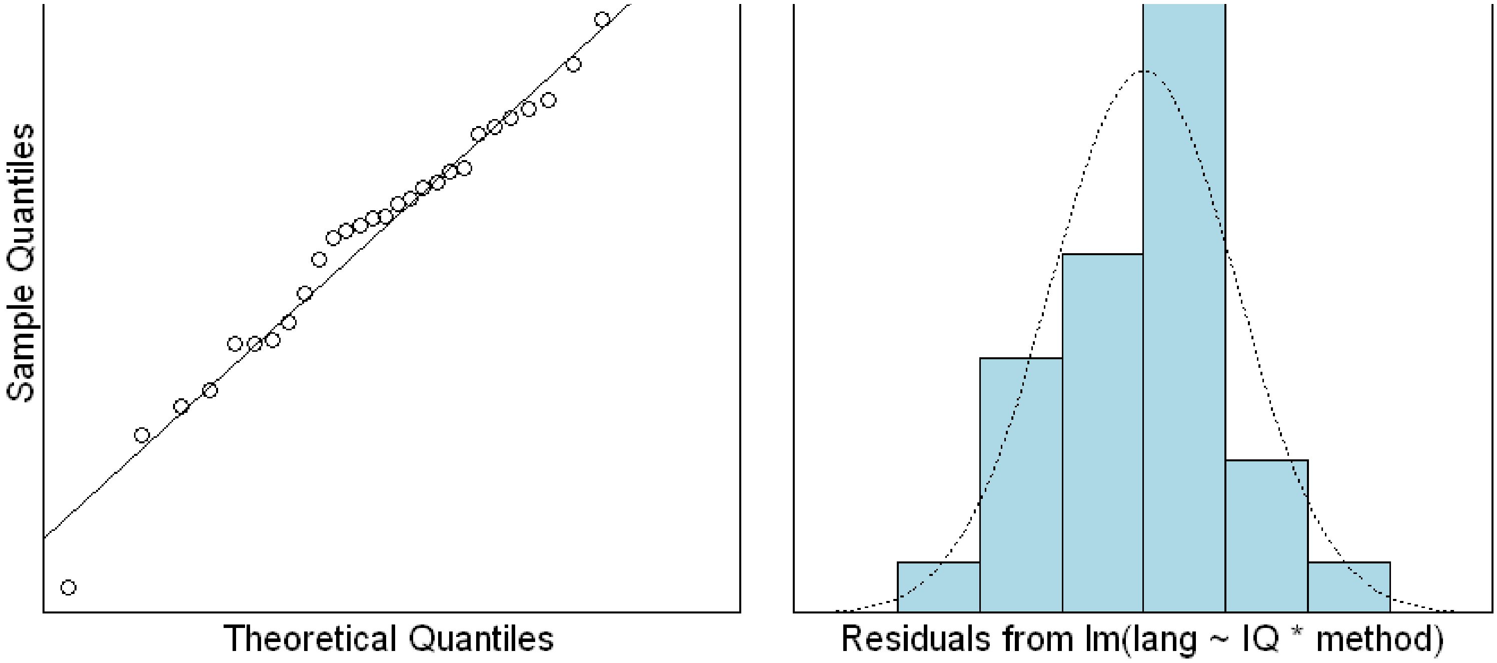
```
TeachIQmethod.fit <- lm(lang ~ IQ * method, data = teach.df)
plot(TeachIQmethod.fit, which = 1)
normcheck(TeachIQmethod.fit)
cooks20x(TeachIQmethod.fit)
```

Skip to main content

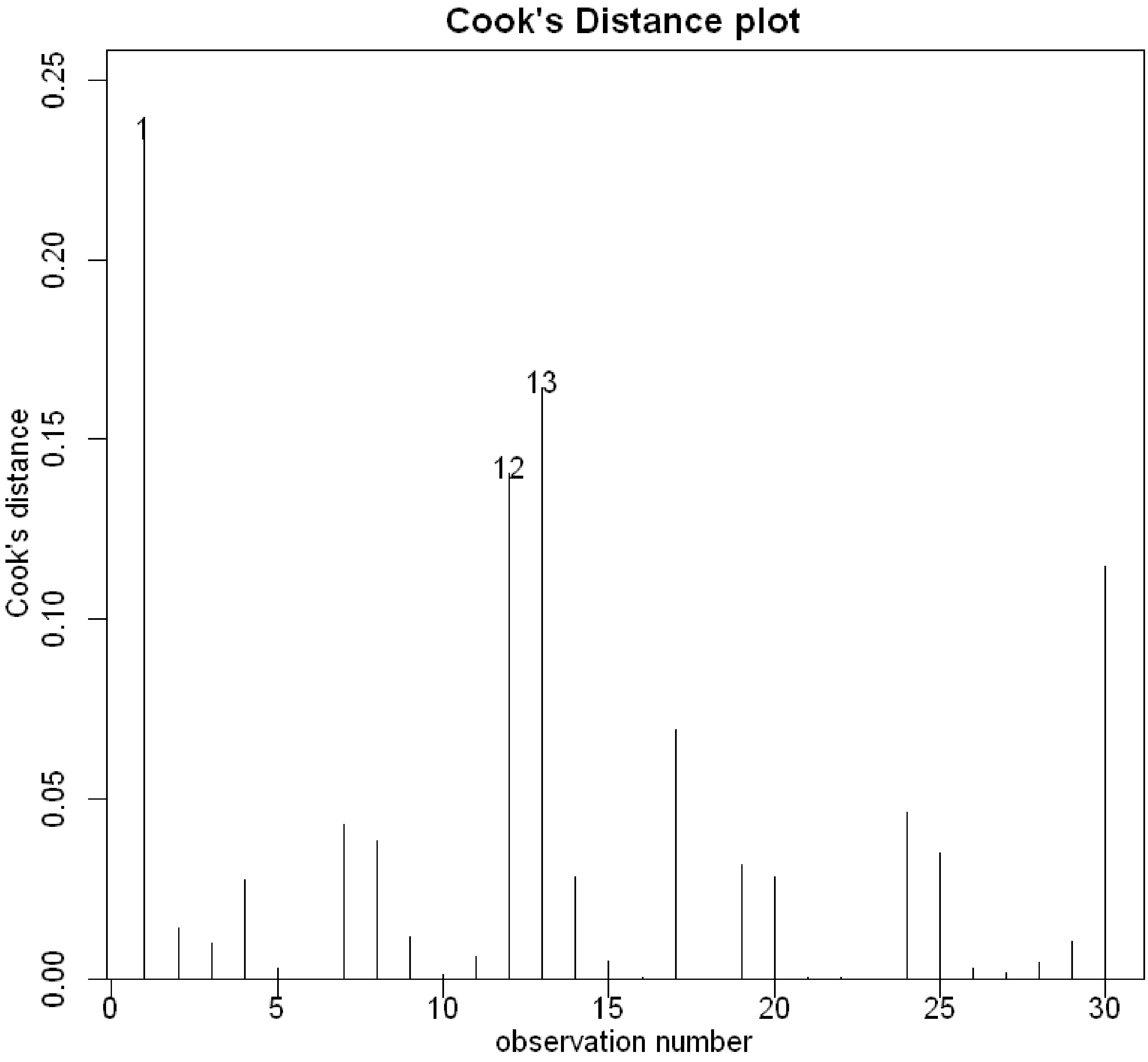
Skip to main content



Skip to main content



Skip to main content



It looks like we can trust the output of the fitted model.

9.2. Model selection using Occam’s razor

Our fitted interaction model is:

Skip to main content

```
summary(TeachIQmethod.fit)
```

```
Call:
lm(formula = lang ~ IQ * method, data = teach.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.8884  -3.8732   0.3435   3.6598  11.2420 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  26.8346    14.5250   1.847  0.07704 . 
IQ            0.4471     0.1241   3.604  0.00142 **
method2      39.0098    20.7473   1.880  0.07227 . 
method3       3.5617    19.7222   0.181  0.85820   
IQ:method2   -0.2587     0.1831  -1.413  0.17042   
IQ:method3   -0.1546     0.1749  -0.883  0.38574   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.199 on 24 degrees of freedom
Multiple R-squared:  0.8121,	Adjusted R-squared:  0.7729 
F-statistic: 20.74 on 5 and 24 DF,  p-value: 5.284e-08
```

在之前的章节中，我们已经看到，如果这样做可以简化拟合模型，我们将删除不重要的项。这是模型选择的非常重要的原则，也是 “Occam’s Razor” 原理的应用，又称为“principle of parsimony”。

该原则指出，在预测能力相等的竞争模型中，应选择参数最少的模型。

在STATS20x中，有时我们称之为 “keep it simple, statistician” 的原则。在本课程中，我们使用的一般模型选择方法是进行假设检验，以确定是否可以从当前模型中删除最复杂的项。

> 当我们在模型公式中使用了“乘上”符号时，表示我们认为它具有交互效应。
>
> 交互效应不一定会带来较好的效果。有时也需要考虑去除交互效应，以获得更好的效果。

anova 函数分析

Sum：来解释的偏差 R 方等于 1-其他偏差/总偏差

```
anova(TeachIQmethod.fit)
```

Skip to main content

A anova: 4 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
IQ	1	1004.41693	1004.41693	26.141645	3.123528e-05
method	2	2901.82976	1450.91488	37.762507	3.866766e-08
IQ:method	2	78.82287	39.41143	1.025749	3.737167e-01
Residuals	24	922.13044	38.42210	NA	NA

Occam’s razor 原理要求我们通过删除交互项来精简我们的模型。为此，我们只需在模型公式中用“+”替换交互项“x”。非交互模型有时被称为加法模型（因为效应“相加”）或“主效应”模型。

```
TeachIQmethod.fit2 <- lm(lang ~ IQ + method, data = teach.df)
summary(TeachIQmethod.fit2)
```

```
Call:
lm(formula = lang ~ IQ + method, data = teach.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.8936  -3.1331  -0.3047   4.1294  11.0003 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  42.08552    8.73921   4.816 5.47e-05 ***
IQ            0.31564    0.07341   4.299 0.000213 ***
method2       9.87793    2.82068   3.502 0.001688 ** 
method3     -14.15922    2.85240  -4.964 3.70e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.205 on 26 degrees of freedom
Multiple R-squared:  0.796,	Adjusted R-squared:  0.7725 
F-statistic: 33.82 on 3 and 26 DF,  p-value: 3.986e-09
```

The equation for the parallel lines (i.e, no-interaction) model is:

$$\text{lang} = \beta_0 + \beta_1 \times \text{IQ} + \beta_2 \times \text{D2} + \beta_3 \times \text{D3} + \varepsilon$$

where, as usual $\varepsilon \overset{iid}{\sim} N(0, \sigma^2)$.

Skip to main content

There are two indicator variables since teaching method has three levels:

- `D2` is an indicator variable whereby: `D2 = 1` if teaching method 2 is taught – otherwise it is 0.
- `D3` is an indicator variable whereby: `D3 = 1` if teaching method 3 is taught – otherwise it is 0.
- Teaching method 1 is the reference/baseline level group.

Let us see if we really do have identical intercepts.

```
anova(TeachIQmethod.fit2)
```

A anova: 3 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
IQ	1	1004.417	1004.4169	26.08997	2.528819e-05
method	2	2901.830	1450.9149	37.68786	2.077362e-08
Residuals	26	1000.953	38.4982	NA	NA

Our preferred model is the no-interaction(没有交互的) model:

```
summary(TeachIQmethod.fit2)
```

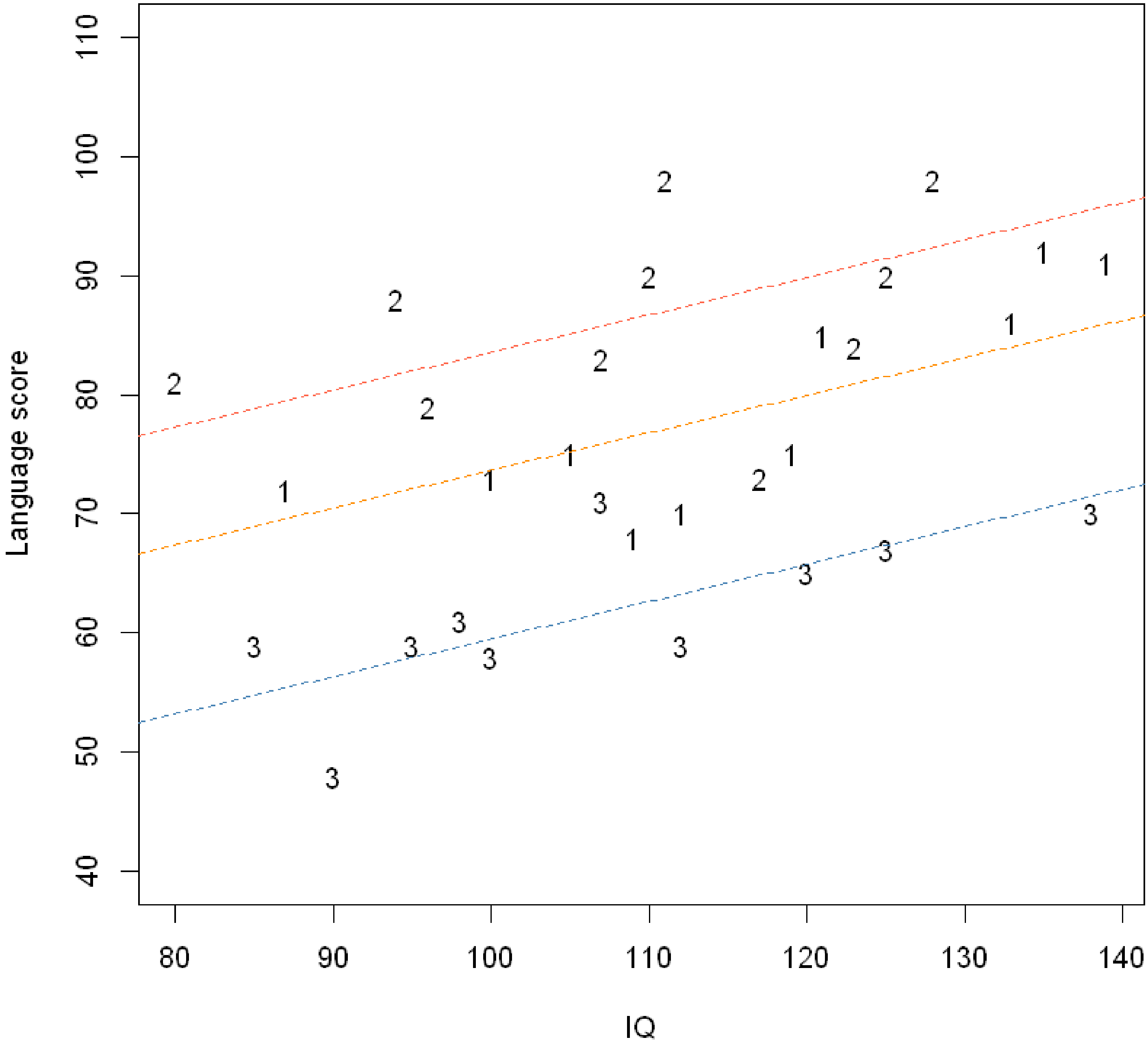
```
Call:
lm(formula = lang ~ IQ + method, data = teach.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.8936  -3.1331  -0.3047   4.1294  11.0003 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  42.08552    8.73921   4.816 5.47e-05 ***
IQ            0.31564    0.07341   4.299 0.000213 ***
method2       9.87793    2.82068   3.502 0.001688 ** 
method3     -14.15922    2.85240  -4.964 3.70e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.205 on 26 degrees of freedom
Multiple R-squared:  0.796,	Adjusted R-squared:  0.7725 
```

Skip to main content

图像：

```r
plot(lang ~ IQ,
    ylim = c(40, 110),
    pch = as.character(method),
    data = teach.df,
    ylab = "Language score"
)
b <- coef(TeachIQmethod.fit2)
abline(b[1], b[2], lty = 2, col = "darkorange")
abline(b[1] + b[3], b[2], lty = 2, col = "tomato")
abline(b[1] + b[4], b[2], lty = 2, col = "steelblue")
```

Skip to main content



We are now able to deduce:

- $\beta_1 > 0$: IQ has a common positive effect on the expected language score of all students
- $\beta_2 > 0$: teaching method 2 is better than teaching method 1 regardless of a student's IQ.
- $\beta_3 < 0$: teaching method 3 is worse than teaching method 1 regardless of a student's IQ.

Skip to main content

9.3. Changing the reference level of teaching method

We need to change this to make method 2 (or alternatively method 3) the baseline. The fitted model will be exactly the same, but the intercept coefficients will change due to the change in reference level.

```
teach.df$method <- relevel(teach.df$method, ref = "2")
TeachIQmethod.fit3 <- lm(lang ~ IQ + method, data = teach.df)
summary(TeachIQmethod.fit3)
```

```
Call:
lm(formula = lang ~ IQ + method, data = teach.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.8936  -3.1331  -0.3047   4.1294  11.0003 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  51.96345    8.24637   6.301 1.14e-06 ***
IQ            0.31564    0.07341   4.299 0.000213 ***
method1      -9.87793    2.82068  -3.502 0.001688 ** 
method3     -24.03715    2.77910  -8.649 3.97e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.205 on 26 degrees of freedom
Multiple R-squared:  0.796,	Adjusted R-squared:  0.7725 
F-statistic: 33.82 on 3 and 26 DF,  p-value: 3.986e-09
```

As the fit2:

```
summary(TeachIQmethod.fit2)
```

Skip to main content

```

Call:
lm(formula = lang ~ IQ + method, data = teach.df)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.8936  -3.1331  -0.3047   4.1294  11.0003 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  42.08552    8.73921   4.816 5.47e-05 ***
IQ            0.31564    0.07341   4.299 0.000213 ***
method2       9.87793    2.82068   3.502 0.001688 ** 
method3     -14.15922    2.85240  -4.964 3.70e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.205 on 26 degrees of freedom
Multiple R-squared:  0.796,	Adjusted R-squared:  0.7725 
F-statistic: 33.82 on 3 and 26 DF,  p-value: 3.986e-09
```

Let us put confidence bounds on our effects.

```r
## Baseline method here is method1.
confint(TeachIQmethod.fit2)
## Baseline method here is method2.
confint(TeachIQmethod.fit3)
```

Skip to main content

A matrix: 4 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	24.1218063	60.0492251
IQ	0.1647361	0.4665482
method2	4.0799363	15.6759248
method3	-20.0224212	-8.2960209

A matrix: 4 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	35.0127936	68.9140989
IQ	0.1647361	0.4665482
method1	-15.6759248	-4.0799363
method3	-29.7496781	-18.3246250

> 备注：有log就是中位值，其他都是讨论的均值

10. Multiple linear regression models

本节需要的包：

```
require(s20x)
```

▶ Show code cell output

10.1. Example: Modelling birth weights using several explanatory variables

我们学习了如何使用线性模型来建模数值和/或因子解释变量的影响。更一般地，原则上我们可以拟合任意数量的解释变量。然而，我们将看到这并不总是一个好主意。需要谨慎处理。举例来说，让我们研究可能解释婴儿出生体重的变量是什么。

Skip to main content

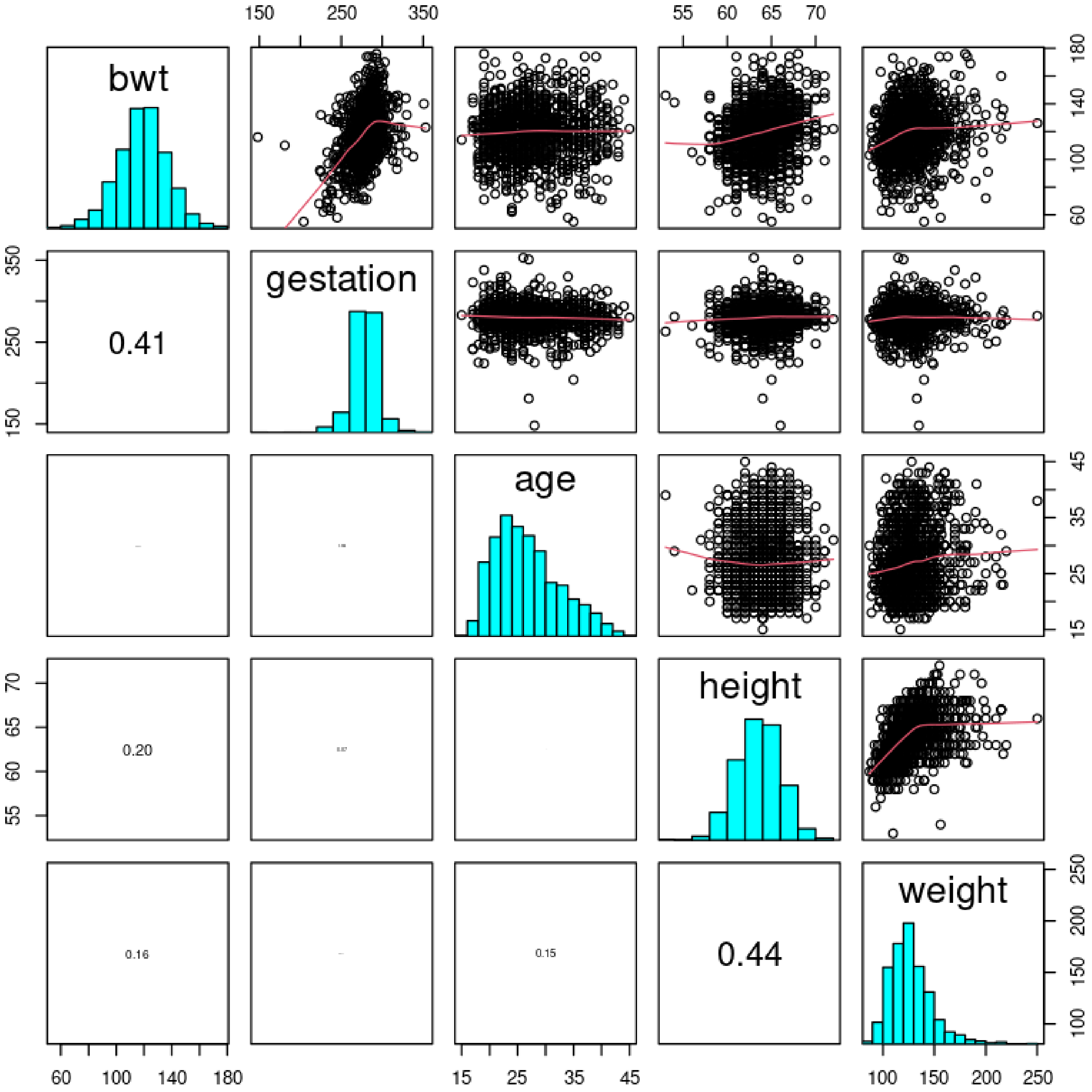
10.2. Exploring relationships between the variables

Let us first inspect the relationships between the numerical explanatory variables and the response variable.

The five variables are in columns 1,2,4,5 and 6 in the data frame Babies.df.

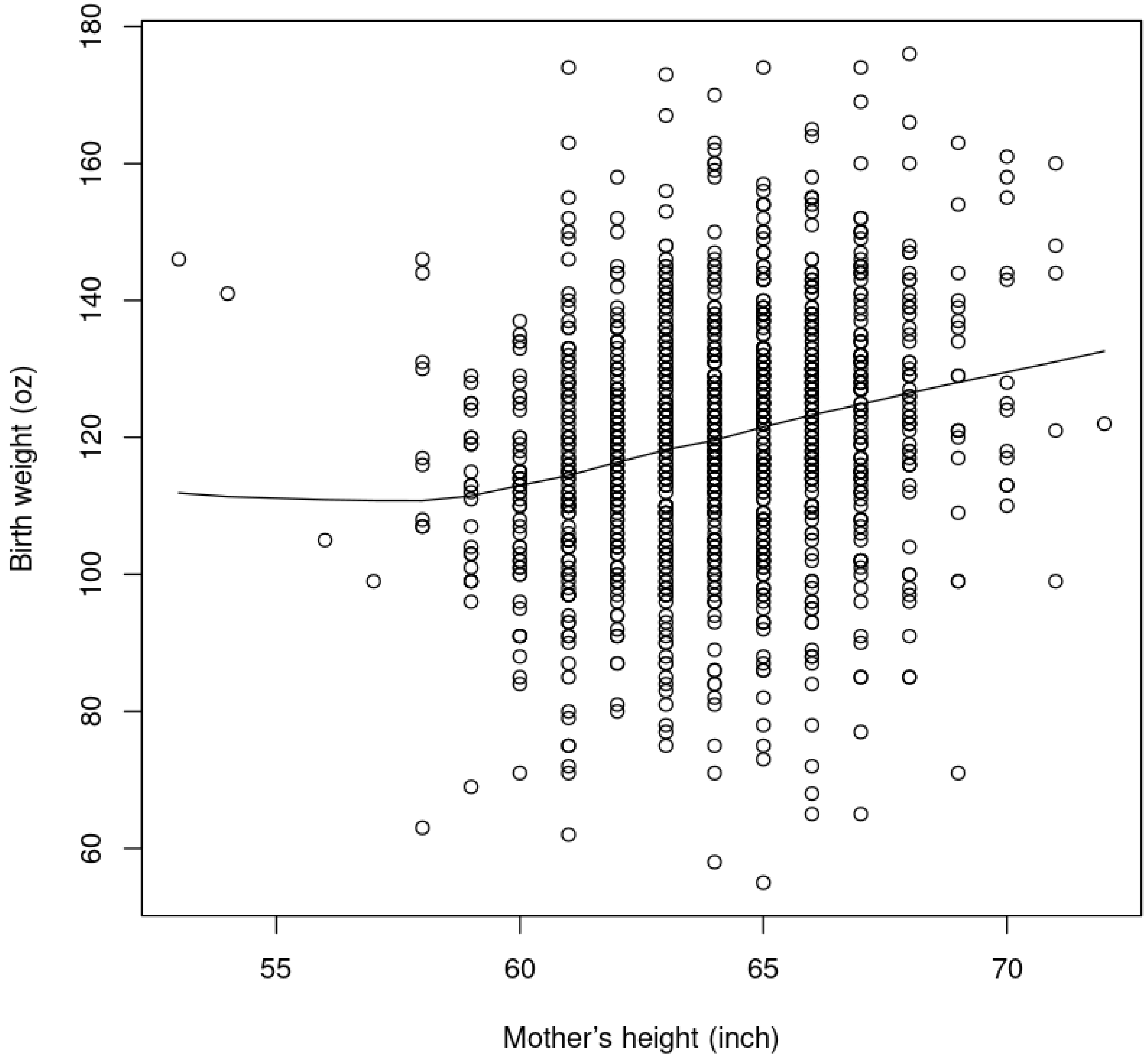
```
## Invoke the s20x library
library(s20x)
## Importing data into R
Babies.df <- read.table("../data/babies_data.txt", header = T)
## Create the pairs plot of the five numeric variables
pairs20x(Babies.df[, c(1, 2, 4, 5, 6)])
```

Skip to main content



```
plot(bwt ~ height,
    data = Babies.df,
    xlab = "Mother’s height (inch)", ylab = "Birth weight (oz)"
)
lines(lowess(Babies.df$height, Babies.df$bwt))
```

Skip to main content



```
summary(lm(bwt ~ height, data = Babies.df))$r.squared
cor(Babies.df$bwt, Babies.df$height)^2 # R 方是差异的平方，所以跟上边那个是一样的
```

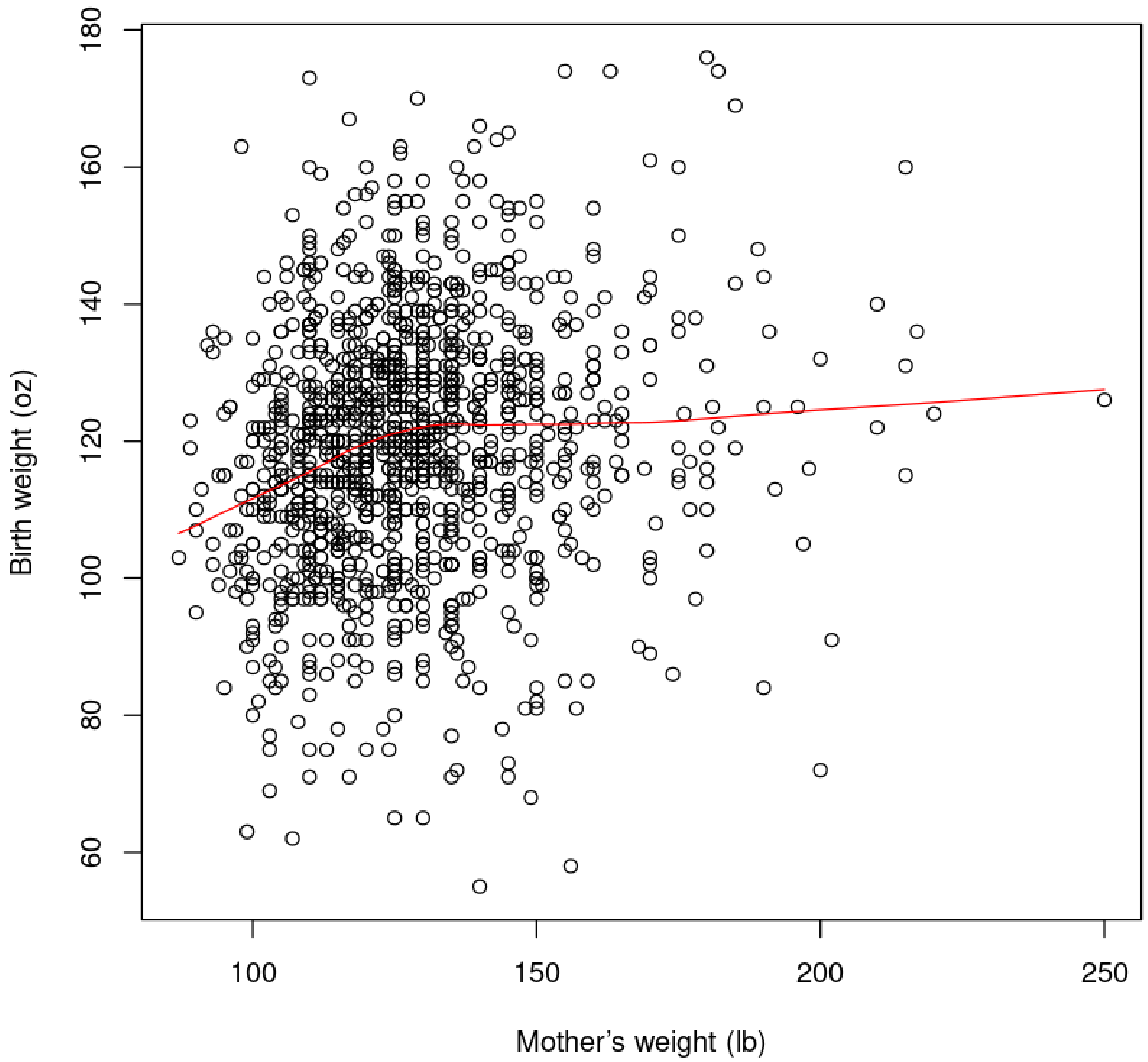
```
0.0414953918045247
0.0414953918045247
```

Skip to main content

Looking at the pairs plot again, we also see a somewhat weak relationship between `bwt` and mother’s `weight`.

```r
plot(bwt ~ weight,
    data = Babies.df,
    xlab = "Mother’s weight (lb)",
    ylab = "Birth weight (oz)"
)
lines(lowess(Babies.df$weight, Babies.df$bwt), col = "red")
```

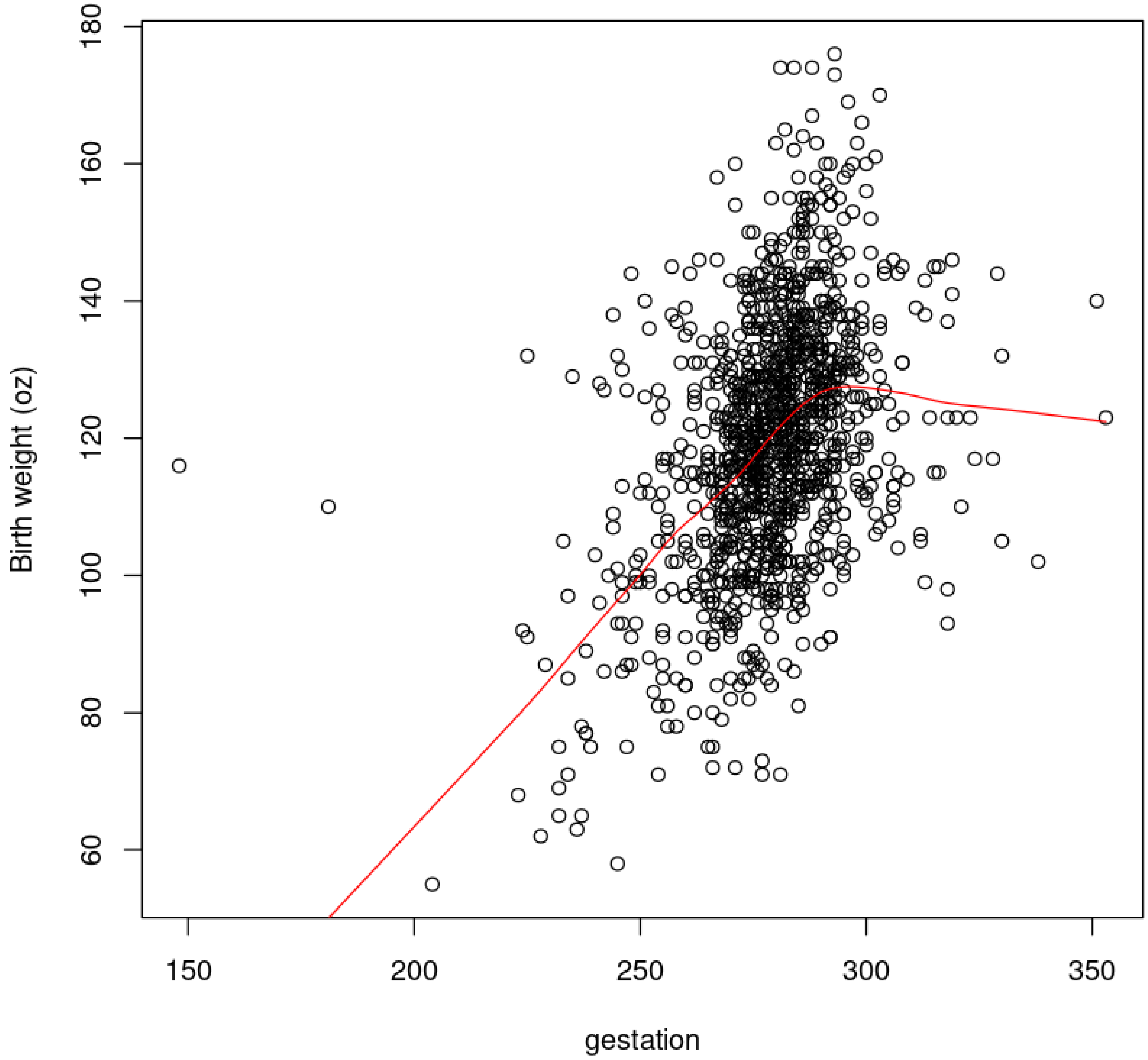
Skip to main content



胎儿孕育时间与其出生体重之间存在更强的关系，这并不令人意外，因为孩子在母亲子宫内的时间越长，孩子就有更多的时间来获得营养和生长。但是，在某个特定的胎龄后，这种关系显然会变得平缓起来 - 有些人称其为“曲棍球杆形状的曲线”。

```
plot(bwt ~ gestation, data = Babies.df, ylab = "Birth weight (oz)")
lines(lowess(Babies.df$gestation, Babies.df$bwt), col = "red")
```

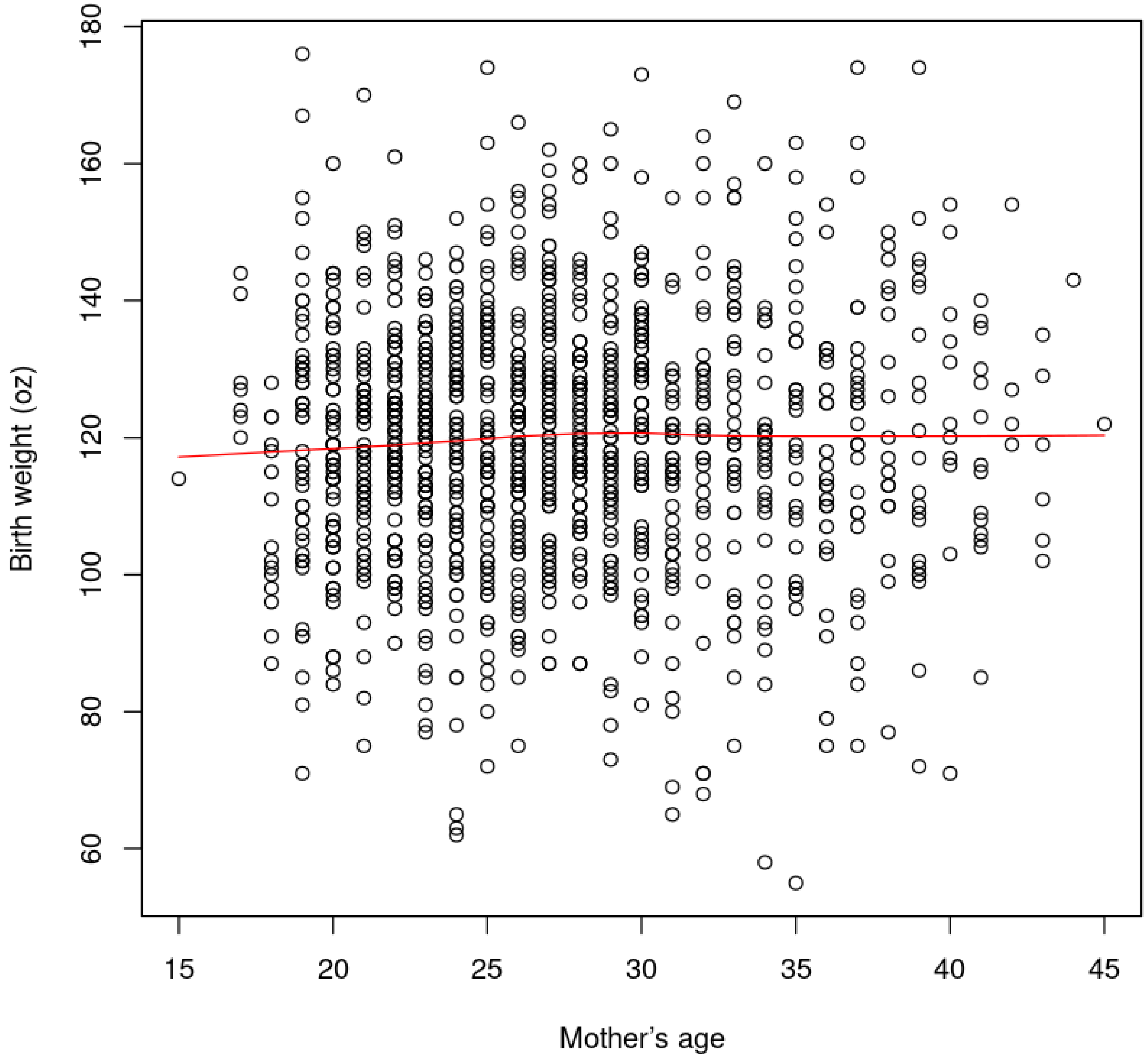
Skip to main content



There does not seem to be any relationship between a mother’s age and her child’s `bwt`.

```
plot(bwt ~ age,
     data = Babies.df,
     xlab = "Mother’s age",
     ylab = "Birth weight (oz)"
)
```

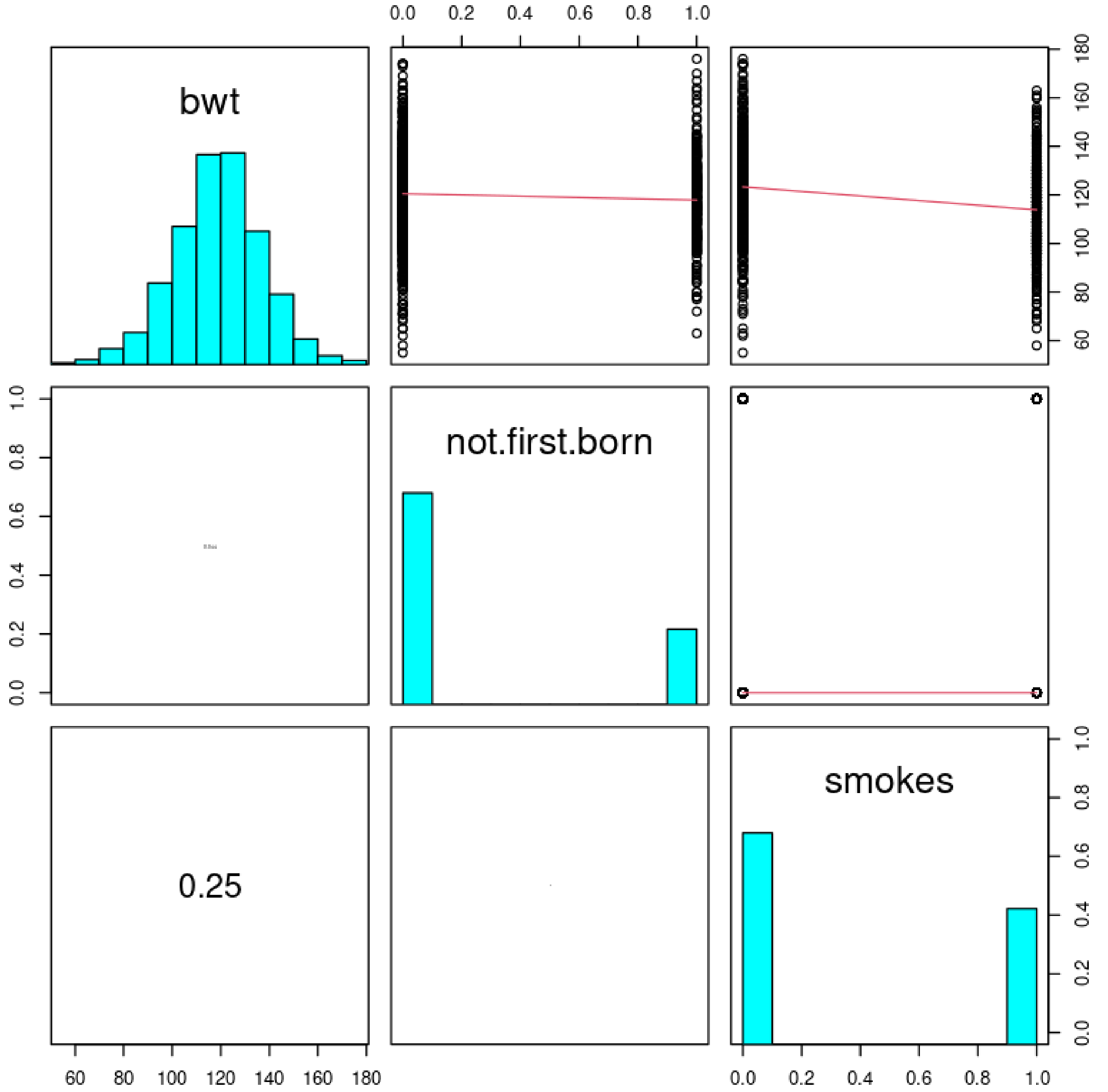
Skip to main content



Note: There seem to be some outlying data points(一些偏远的数据点) in these plots. There does not appear to be much of a relationship between the x variables, except between `height` and `weight`.

```
pairs20x(Babies.df[, c(1, 3, 7)])
```

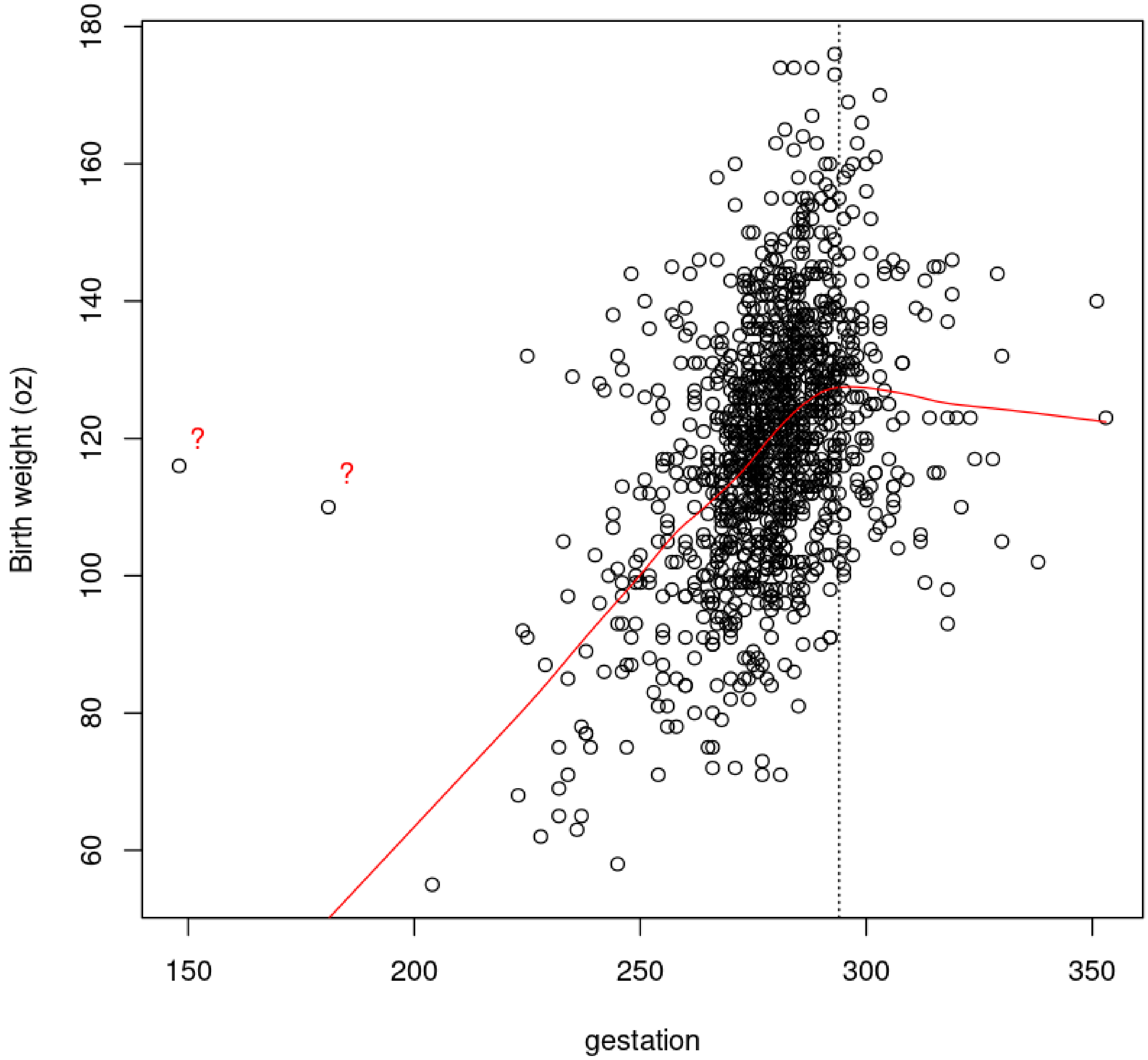
Skip to main content



让我们从理解“解释”的角度开始，因为它是最强大的关系之一。那些不典型的数据点已经用问号标记了。我们稍后会添加其他的解释变量。

```
plot(bwt ~ gestation, data = Babies.df, ylab = "Birth weight (oz)")
lines(lowess(Babies.df$gestation, Babies.df$bwt), col = "red")
text(c(152, 185), c(120, 115), "?", col = "red")
abline(v = 294, lty = 3)
```

Skip to main content



They look extremely implausible as they have typical birth-weight but have a gestational age that is extremely low for these data. 他们看起来非常难以置信,因为他们典型的出生体重但有孕龄,对这些数据是极低的。

```
id <- (Babies.df$gestation < 200)
Babies.df[id, ]
```

Skip to main content

A data.frame: 2 × 7

	bwt	gestation	not.first.born	age	height	weight	smokes
	<int>	<int>	<int>	<int>	<int>	<int>	<int>
239	116	148	0	28	66	135	0
820	110	181	0	27	64	133	0

Relationship between birth weight and gestational age...

For `gestation` ≤ 294 days we'll use the familiar simple linear regression model

$$\mathrm{E}[\mathrm{bwt}] = \beta_0 + \mathrm{gestation} \times \beta_1$$

We'd like to extend this model by adding an extra term so that the slope changes when `gestation` > 294. That is,

$$\mathrm{E}[\mathrm{bwt}] = \beta_0 + \mathrm{gestation} \times \beta_1 + \mathrm{v} \times \beta_2$$

where v is some suitable explanatory variable. What should v be?

- For `gestation` ≤ 294 the extended model is just the simple linear regression model, so that means v = 0 when gestation ≤ 294.
- For `gestation` > 294 we need another slope effect for gestational age. In fact, we need $v =$ `gestation` -294.

Let's create the new explanatory v = 294 that is described gestation above. We'll give it the name because it is the number of days ODdays that the baby is overdue.

```
Babies.df$ODdays <- ifelse(
    Babies.df$gestation < 294,
    0,
    Babies.df$gestation - 294
)
head(Babies.df, 12) # Print first 12 lines of dataframe
```

Skip to main content

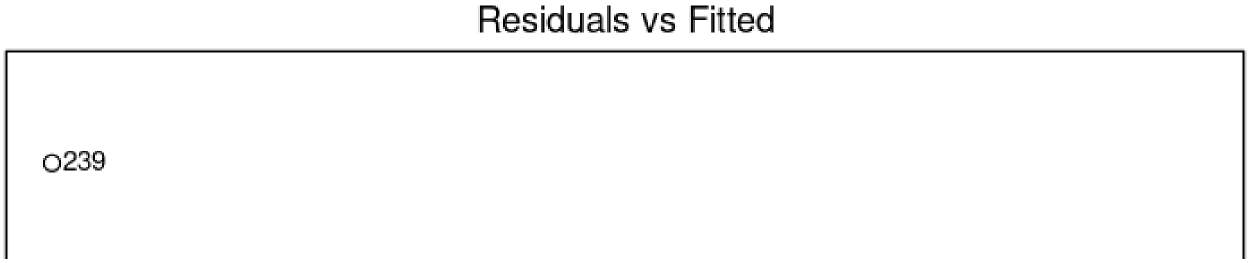
A data.frame: 12 × 8

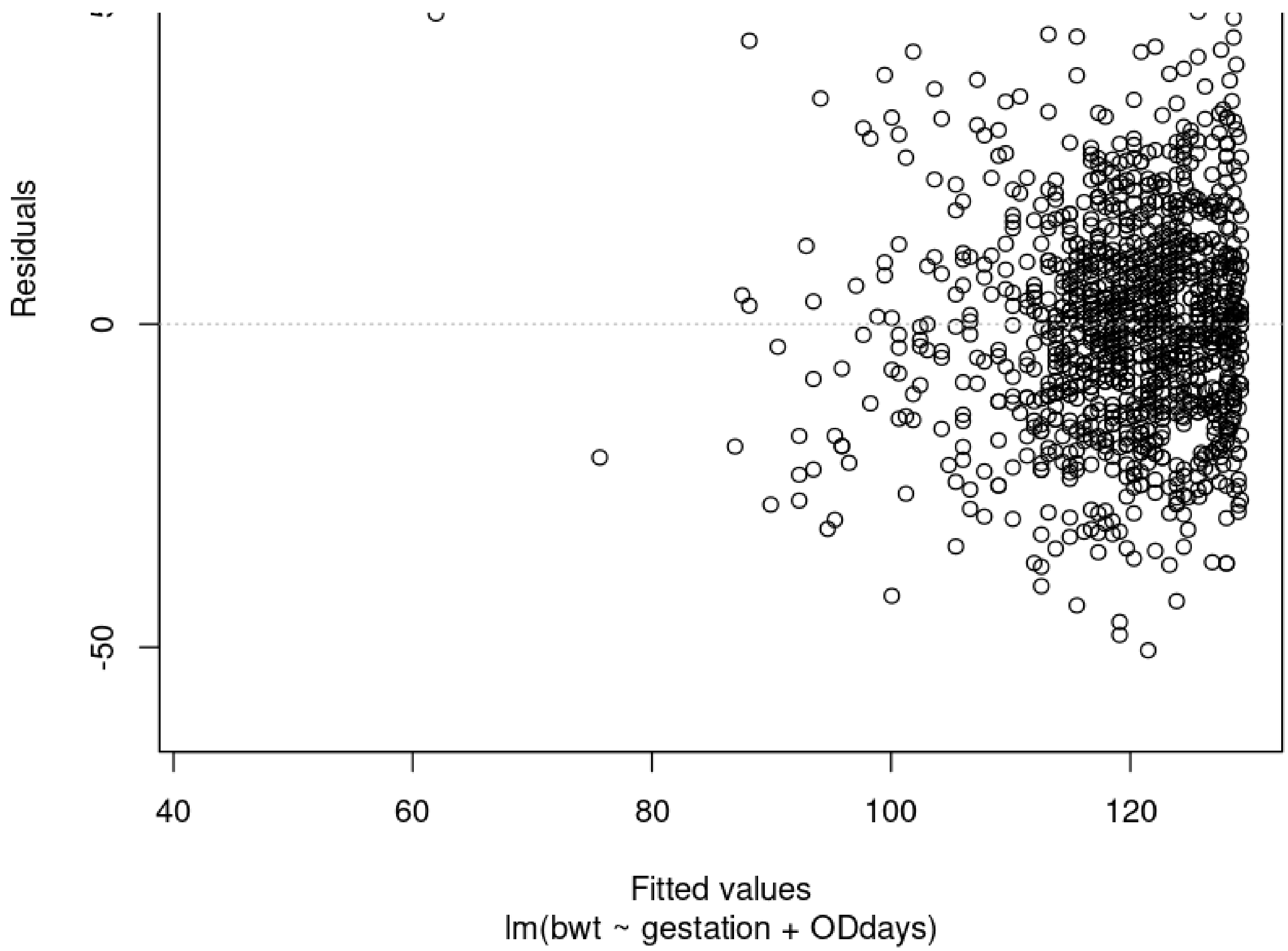
	bwt	gestation	not.first.born	age	height	weight	smokes	ODdays
	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>
1	120	284	0	27	62	100	0	0
2	113	282	0	33	64	135	0	0
3	128	279	0	28	64	115	1	0
4	108	282	0	23	67	125	1	0
5	136	286	0	25	62	93	0	0
6	138	244	0	33	62	178	0	0
7	132	245	0	23	65	140	0	0
8	120	289	0	25	62	125	0	0
9	143	299	0	30	66	136	1	5
10	140	351	0	27	68	120	0	57
11	144	282	0	32	64	124	1	0
12	141	279	0	23	63	128	1	0

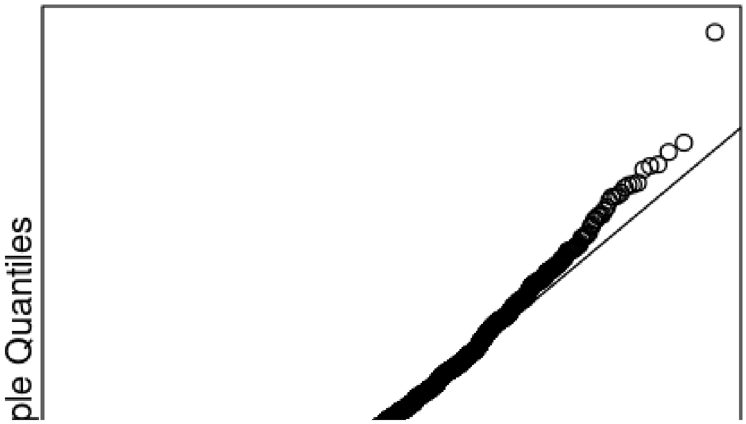
10.3. Fitting the initial model

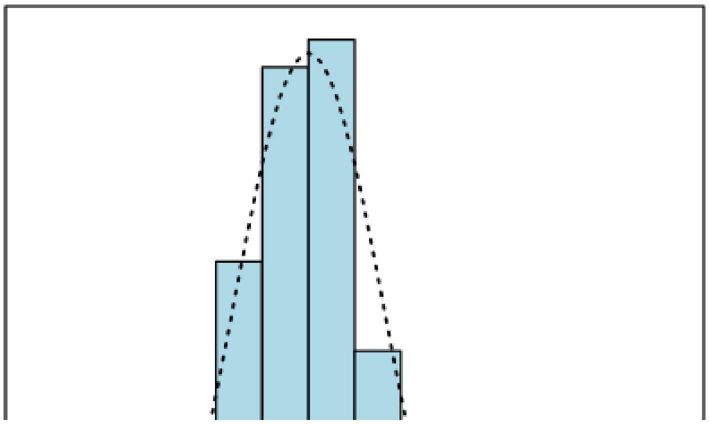
```r
bwt.fit <- lm(bwt ~ gestation + ODdays, data = Babies.df)
plot(bwt.fit, which = 1, add.smooth = FALSE)
normcheck(bwt.fit)
cooks20x(bwt.fit)
```

Skip to main content

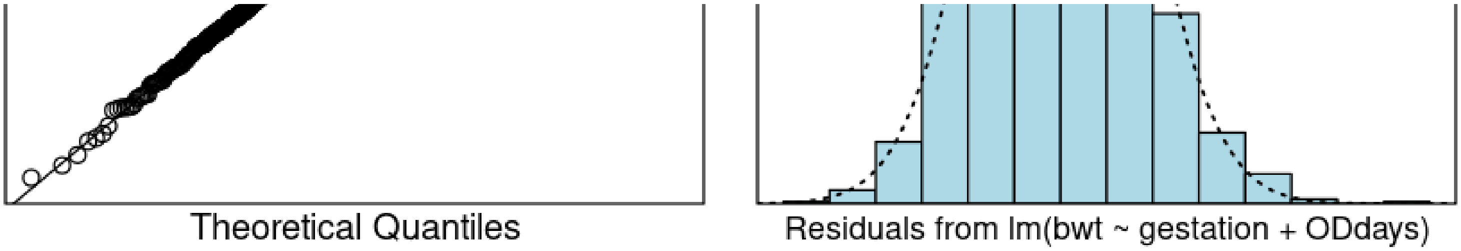

Residuals vs Fitted
239



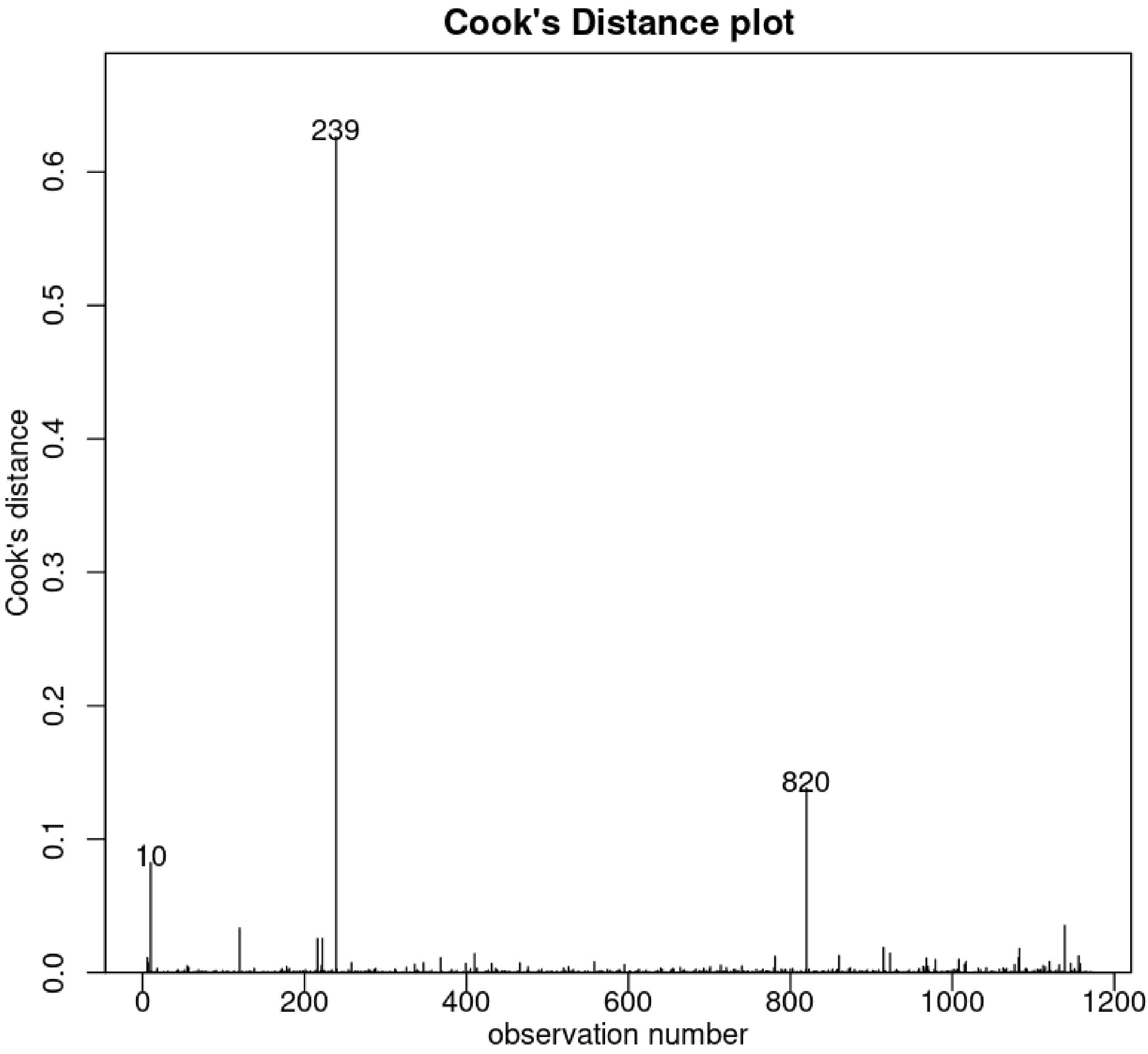




Skip to main content



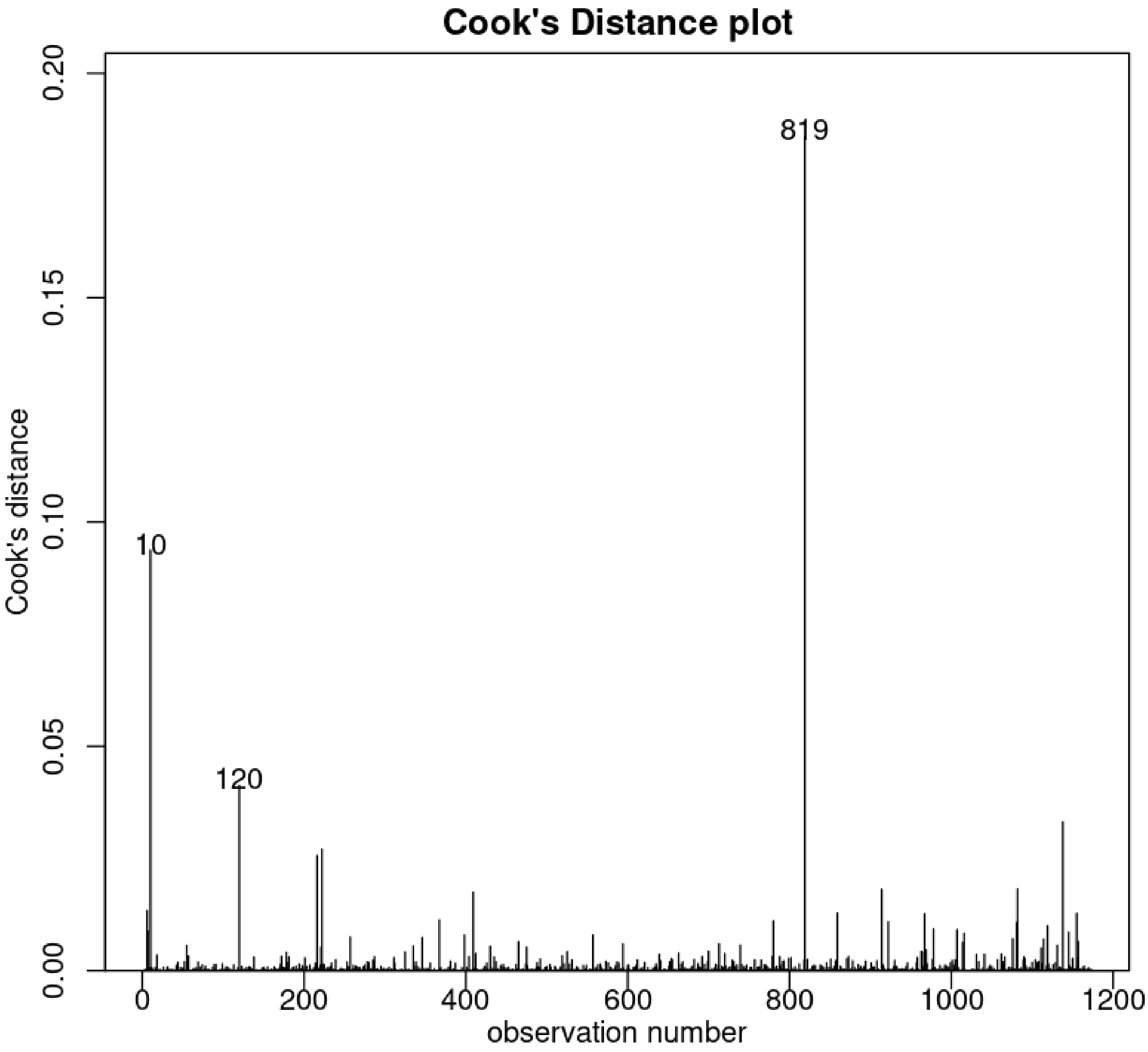
Skip to main content



Let us refit with observation 239 removed.

```
bwt.fit2 <- lm(bwt ~ gestation + ODdays, data = Babies.df[-239, ])
cooks20x(bwt.fit2)
```

Skip to main content

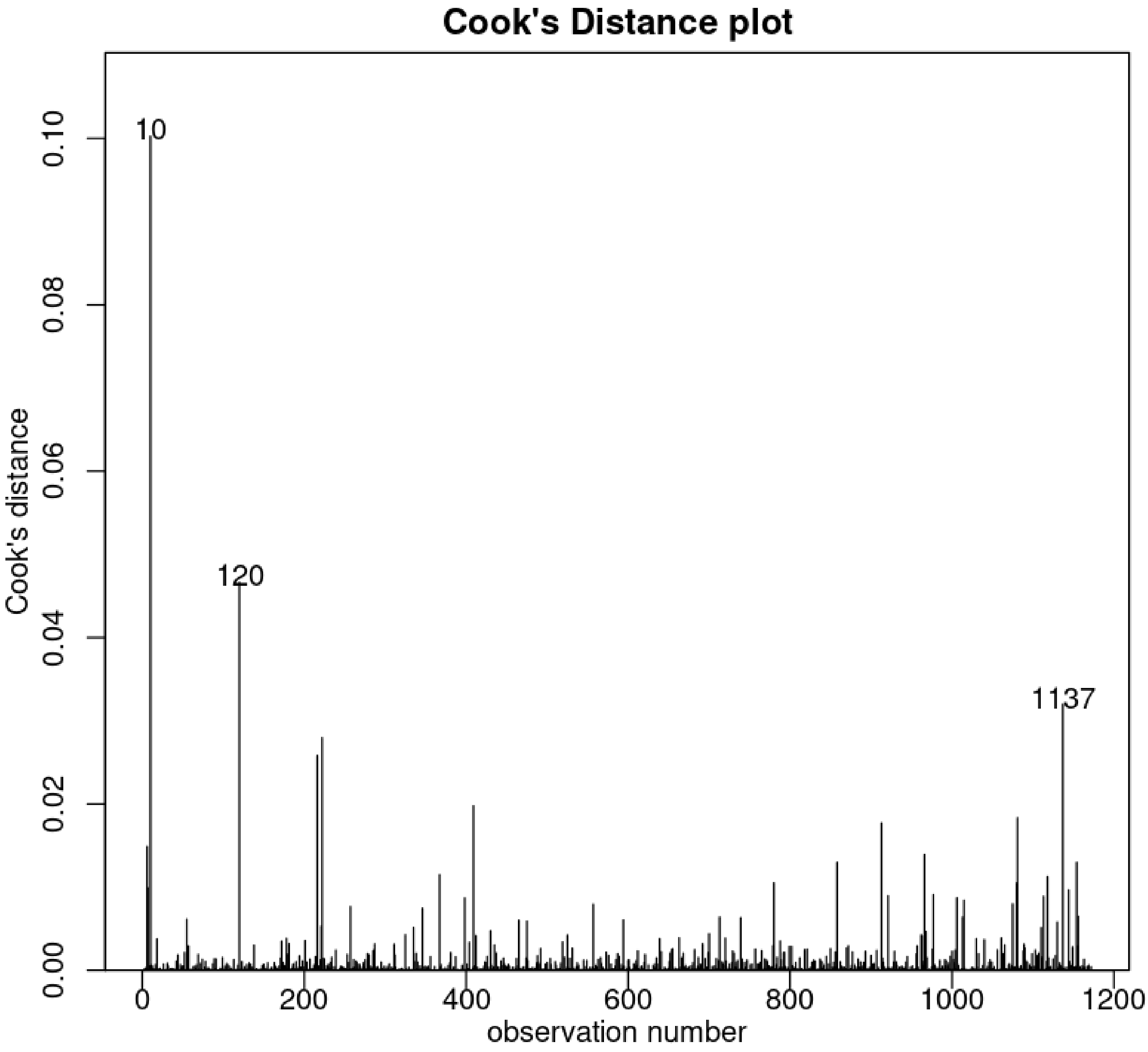


We refit the model using the reduced data.

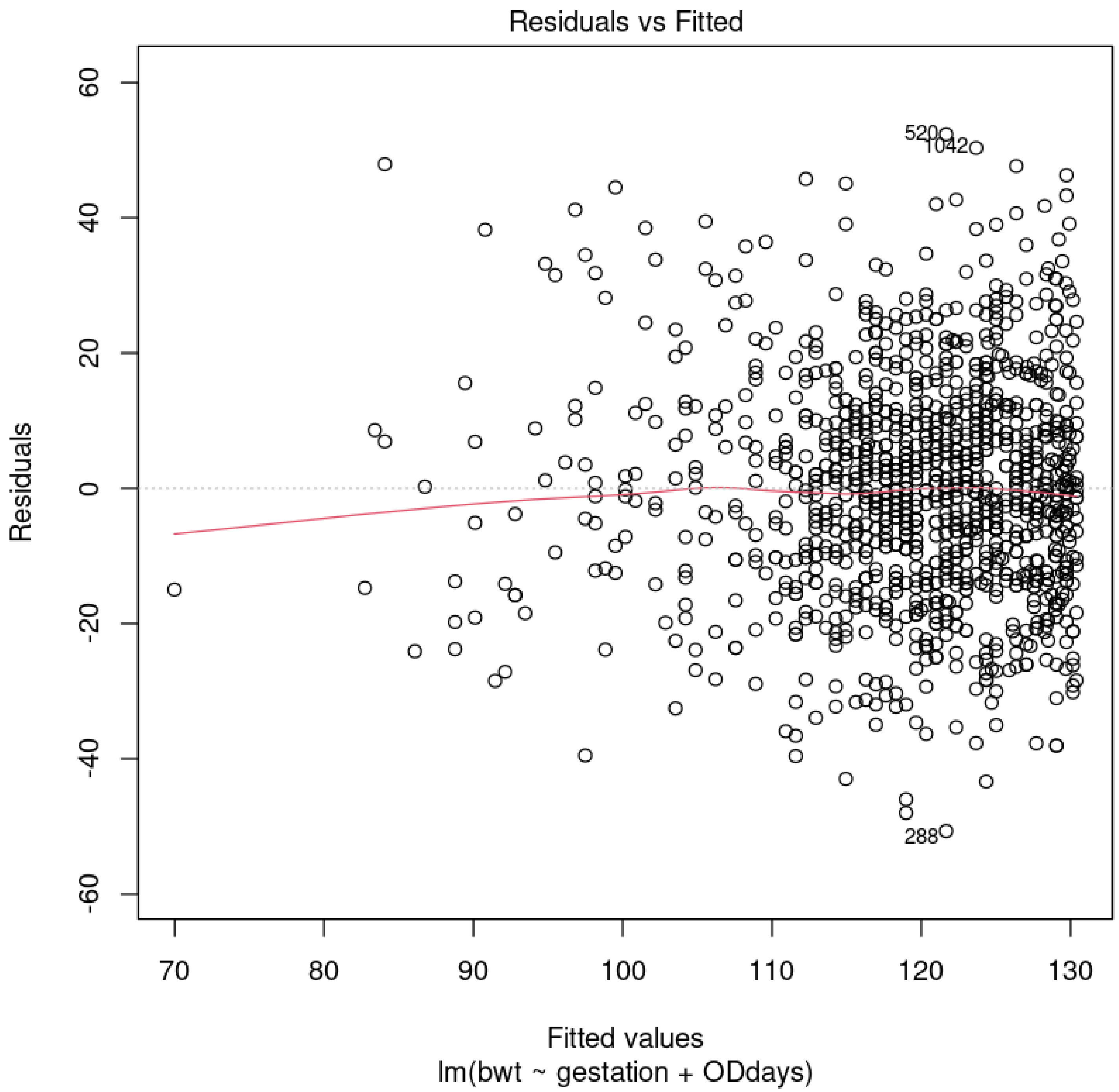
Skip to main content

```r
# This time we demonstrate using the subset argument to remove points
bwt.fit3 <- lm(bwt ~ gestation + ODdays,
    data = Babies.df,
    subset = -c(239, 820)
)
cooks20x(bwt.fit3)
plot(bwt.fit3, which = 1)
```

Skip to main content



Skip to main content

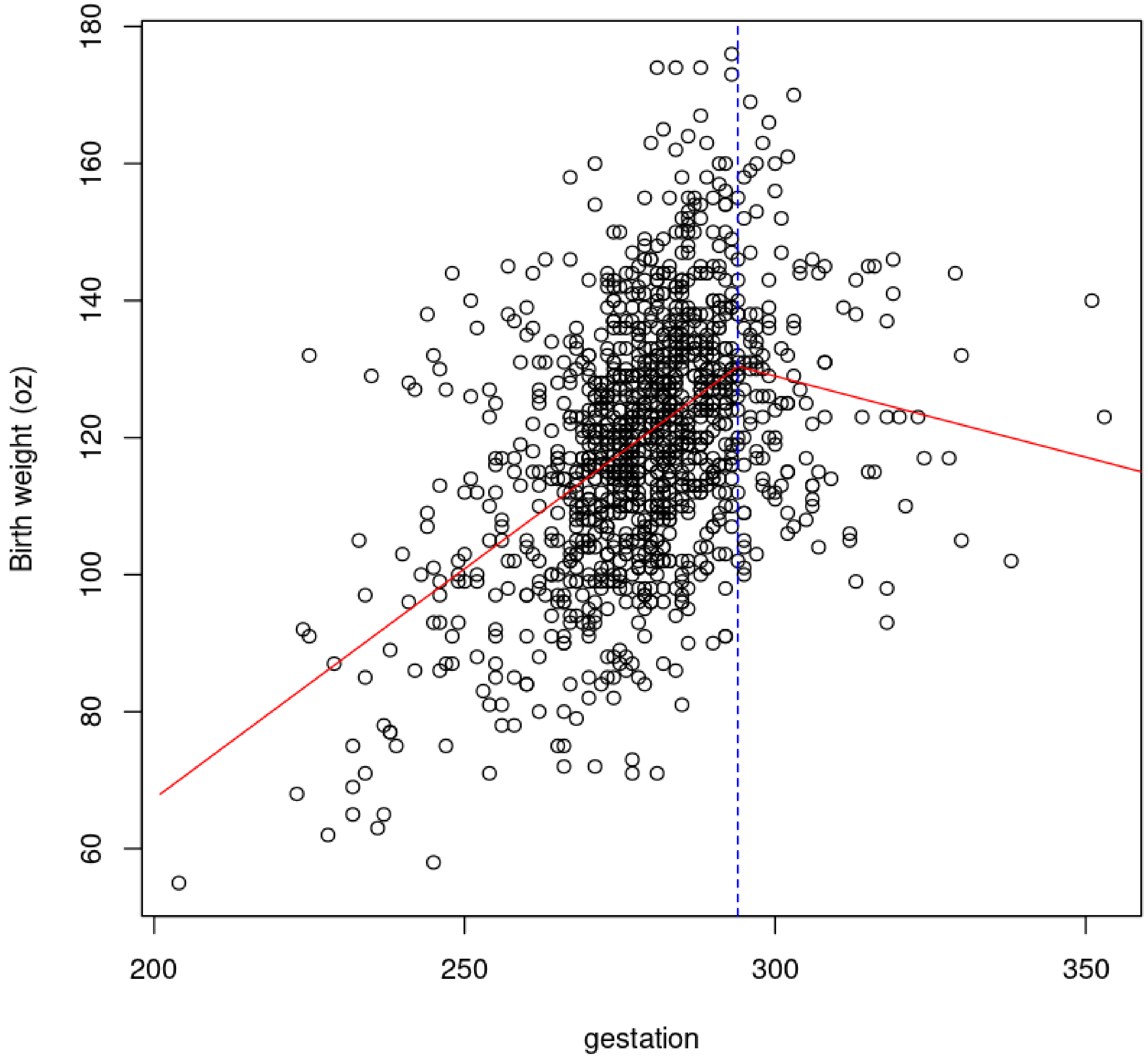


Let’s take a look at our fitted hockey stick model.

Skip to main content

```r
gestation.seq <- 201:360 # Explanatory values at which to get predictions
ODdays.seq <- ifelse(gestation.seq <= 294, 0, gestation.seq - 294)
fit.seq <- predict(bwt.fit3, new = data.frame(
    gestation = gestation.seq,
    ODdays = ODdays.seq
))
plot(bwt ~ gestation,
    data = Babies.df[-c(239, 820), ],
    ylab = "Birth weight (oz)"
)
lines(gestation.seq, fit.seq, col = "red")
abline(v = 294, lty = 2, col = "blue")
```

Skip to main content



模型检查是好的，没有影响力的点依然存在，所以我们可以相信这个。让我们解释输出。

```
summary(bwt.fit3)
```

Skip to main content

```
Call:
lm(formula = bwt ~ gestation + ODdays, data = Babies.df, subset = -c(239, 
    820))

Residuals:
    Min      1Q  Median      3Q     Max 
-50.664 -10.993  -0.308   9.795  52.336 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -66.95336   10.42810   -6.42 1.97e-10 ***
gestation     0.67124    0.03757   17.87  < 2e-16 ***
ODdays       -0.90783    0.11745   -7.73 2.31e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16.23 on 1169 degrees of freedom
Multiple R-squared:  0.2188,	Adjusted R-squared:  0.2174 
F-statistic: 163.7 on 2 and 1169 DF,  p-value: < 2.2e-16
```

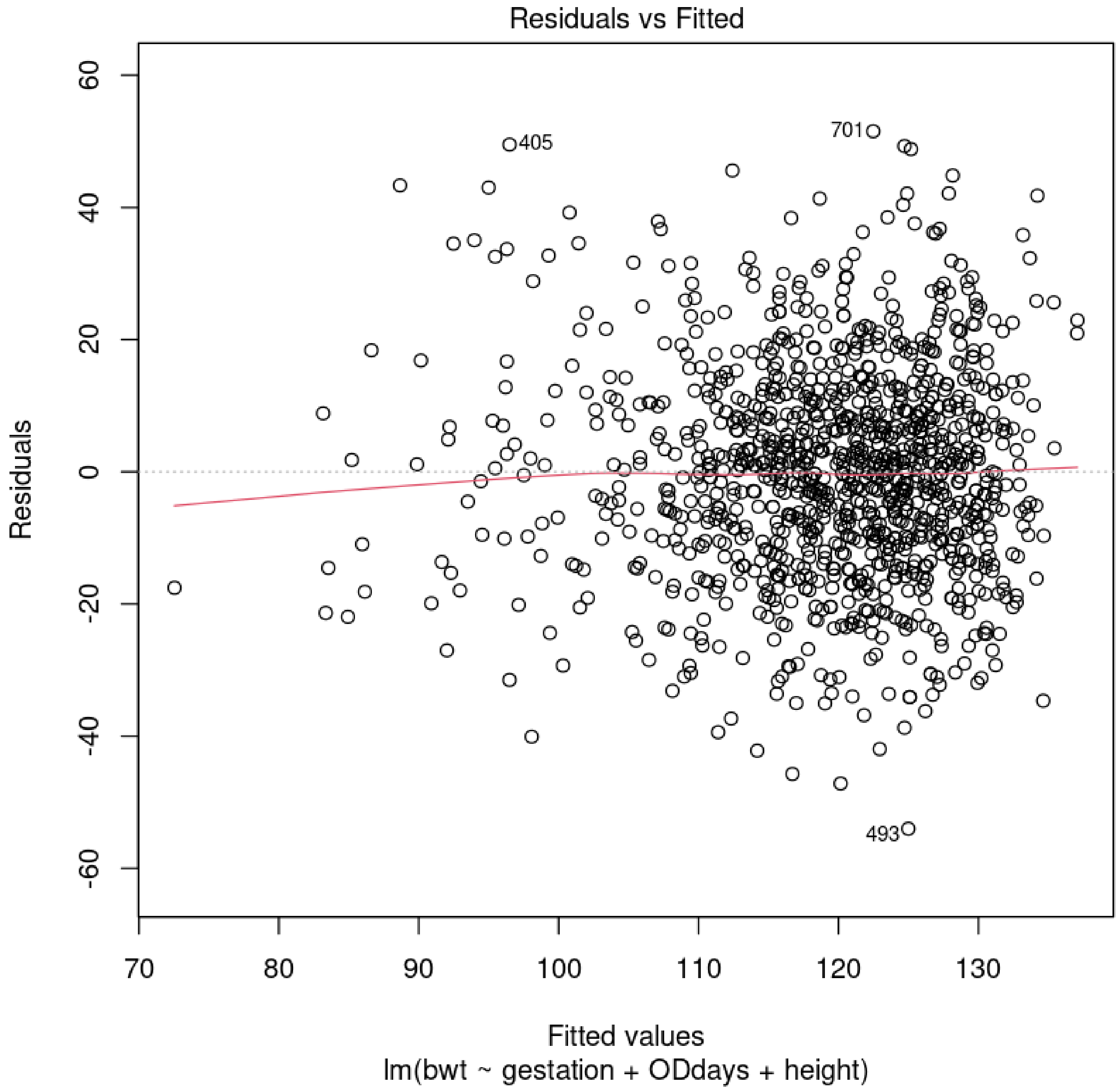
The fitted model is:

$$\mathrm{E}[\mathrm{bwt}] = -66.95 + 0.67 \times \mathrm{gestation} - 0.91 \times \mathrm{ODdays}$$

10.4. Multiple linear regression model: Adding more terms to the model and the peril of multi-collinearity

```r
bwt.fit4 = lm(bwt ~ gestation + ODdays + height,
    data = Babies.df,
    subset = -c(239, 820)
)
plot(bwt.fit4, which = 1)
```

Skip to main content



All seems okay. Let us make sure that this makes sense in terms of output.

```
summary(bwt.fit4)
```

Skip to main content

```
Call:
lm(formula = bwt ~ gestation + ODdays + height, data = Babies.df,
    subset = -c(239, 820))

Residuals:
    Min      1Q  Median      3Q     Max
-53.999 -10.393  -0.050   9.772  51.514

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -139.20571   15.05961  -9.244  < 2e-16 ***
gestation      0.65219    0.03703  17.613  < 2e-16 ***
ODdays        -0.89039    0.11543  -7.714 2.61e-14 ***
height         1.21083    0.18495   6.547 8.79e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.94 on 1168 degrees of freedom
Multiple R-squared:  0.2464,	Adjusted R-squared:  0.2445
F-statistic: 127.3 on 3 and 1168 DF,  p-value: < 2.2e-16
```

Let us add `weight` to the model. We’re going to save some typing and use the `update`` function to update our model.

```
bwt.fit5 <- update(bwt.fit4, ~ . + weight)
summary(bwt.fit5)
```

```
Call:
lm(formula = bwt ~ gestation + ODdays + height + weight, data = Babies.df,
    subset = -c(239, 820))

Residuals:
    Min      1Q  Median      3Q     Max
-53.053 -10.540   0.121  10.076  47.746

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -131.68169   15.14974  -8.692  < 2e-16 ***
gestation      0.65624    0.03688  17.795  < 2e-16 ***
ODdays        -0.90868    0.11502  -7.900 6.41e-15 ***
height         0.90486    0.20453   4.424 1.06e-05 ***
weight         0.08535    0.02485   3.434 0.000615 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.87 on 1167 degrees of freedom
Multiple R-squared:  0.254,	Adjusted R-squared:  0.2514
F-statistic: 99.32 on 4 and 1167 DF,  p-value: < 2.2e-16
```

Skip to main content

11. Linear models with a single factor explanatory variable having three or more levels (One-way analysis of variance)

本节需要的包：

```
require(s20x)
require(dplyr)
require(emmeans)
```

▶ Show code cell output

11.1. Example with a 5-level explanatory factor variable

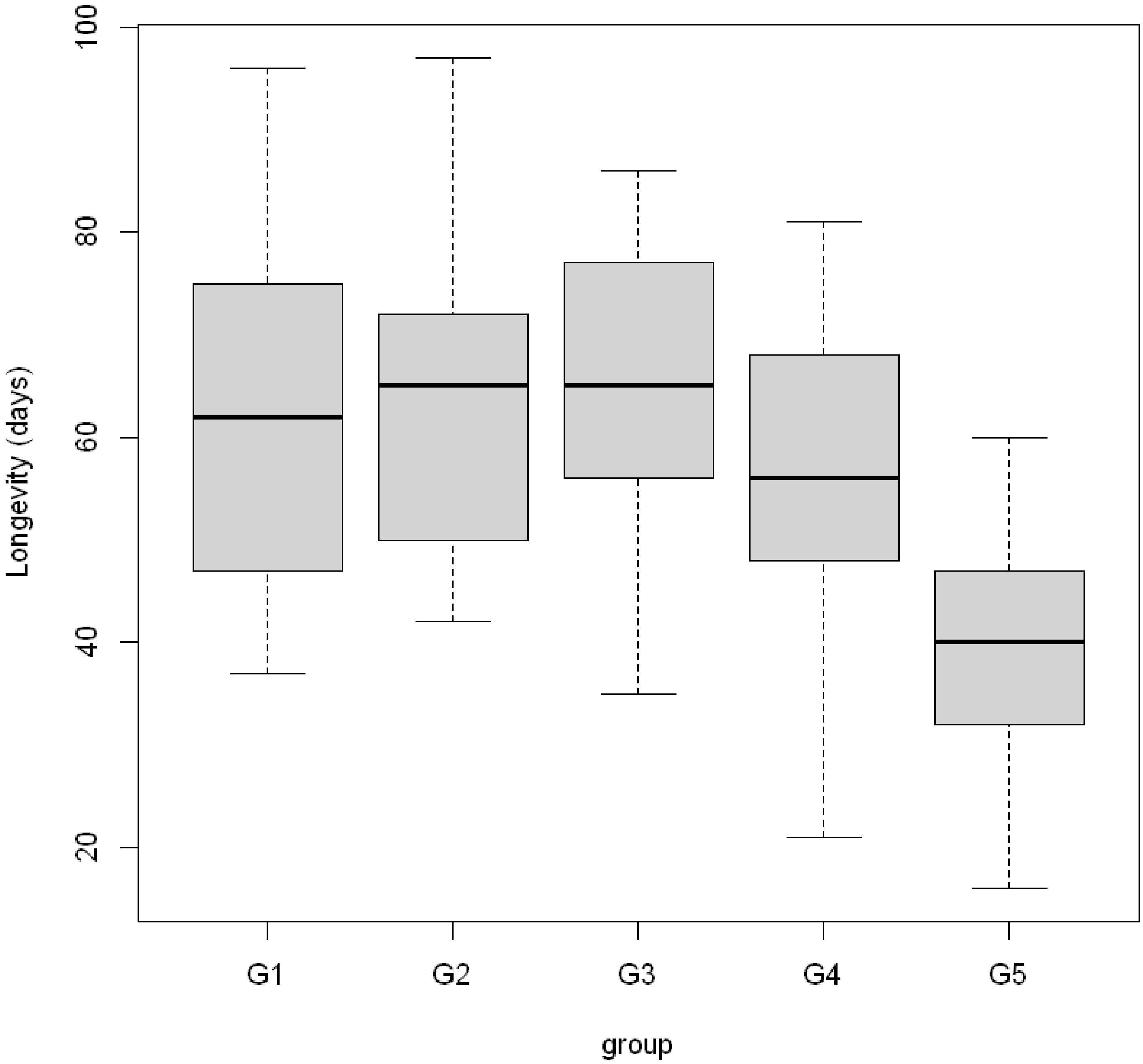
The explanatory factor variable was group, with five levels:

- G1 males living alone,
- G2 males living with one interested female,
- G3 males living with eight interested females,
- G4 males living with one uninterested female, and
- G5 males living with eight uninterested females.

Let us take a look at the data:

```
library(s20x)
Fruitfly.df <- read.csv("../data/Fruitfly.csv", header = T)
Fruitfly.df$group <- factor(Fruitfly.df$group)
boxplot(days ~ group, data = Fruitfly.df, ylab = "Longevity (days)")
```

Skip to main content

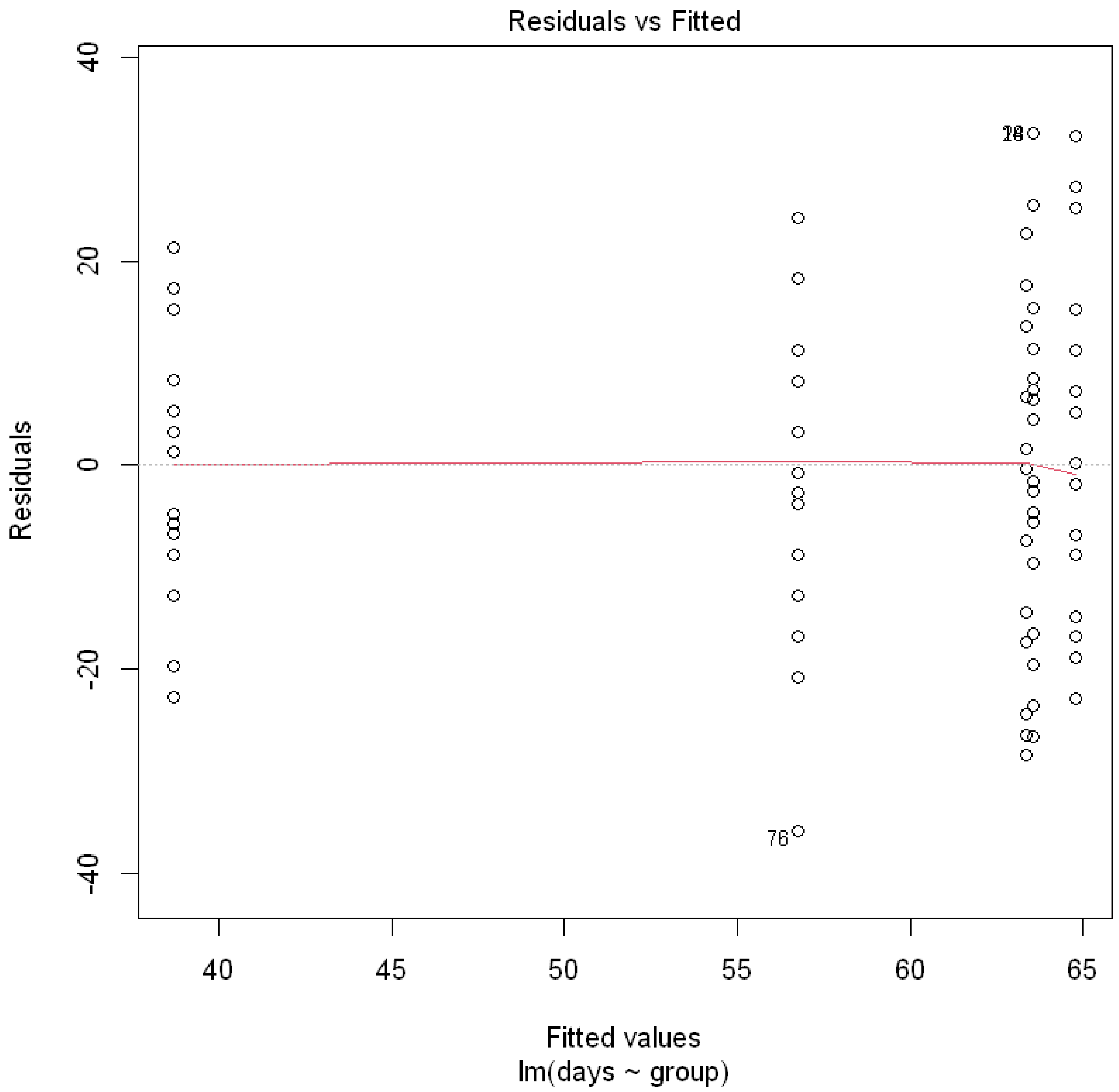


As seen in previous chapters that involved categorical explanatory variables, our model specification uses indicator variables. In this case:

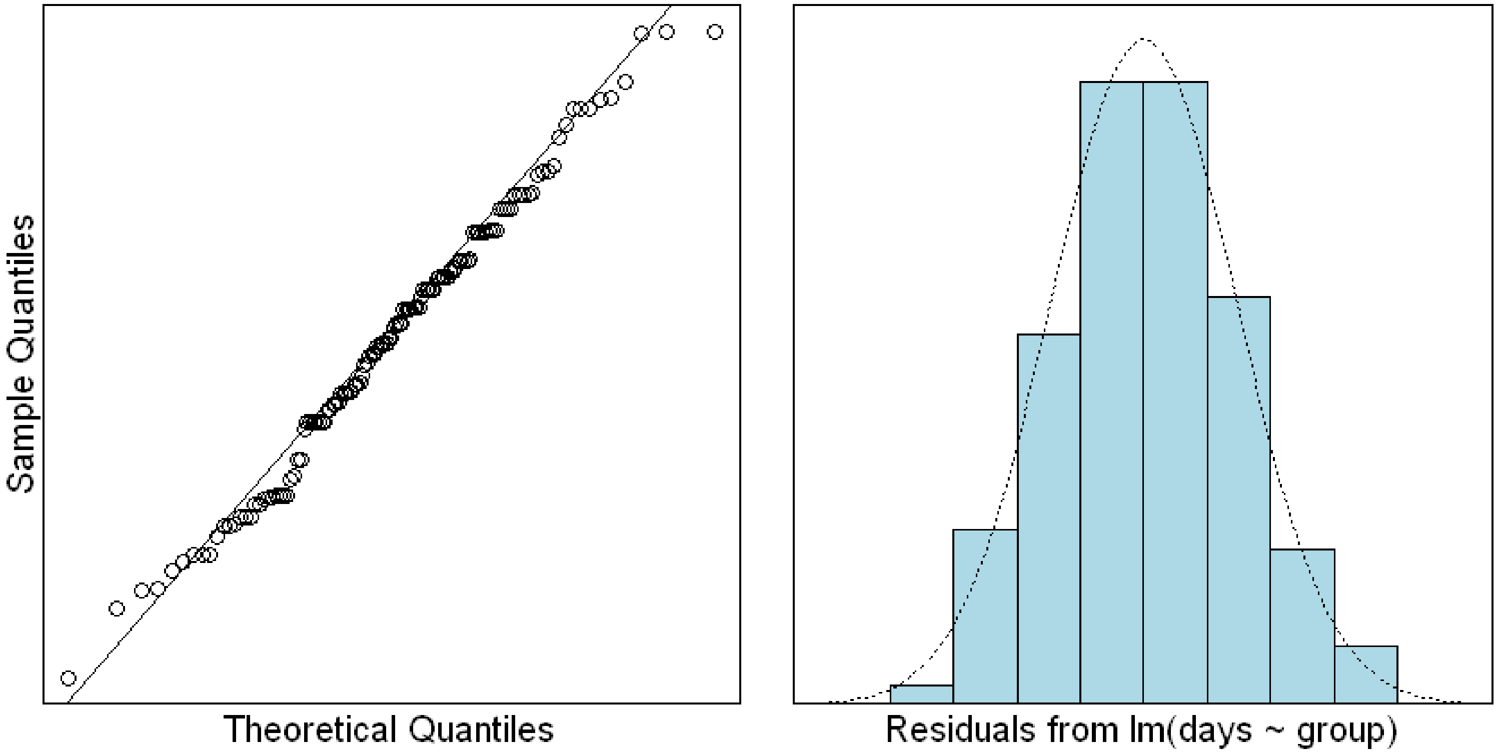
$$\text{days} = \beta_0 + \beta_1 \times \texttt{D2} + \beta_2 \times \texttt{D3} + \beta_3 \times \texttt{D4} + \beta_4 \times \texttt{D5} + \epsilon$$

```r
Fruitfly.fit <- lm(days ~ group, data = Fruitfly.df)
plot(Fruitfly.fit, which = 1)
normcheck(Fruitfly.fit)
cooks20x(Fruitfly.fit)
```

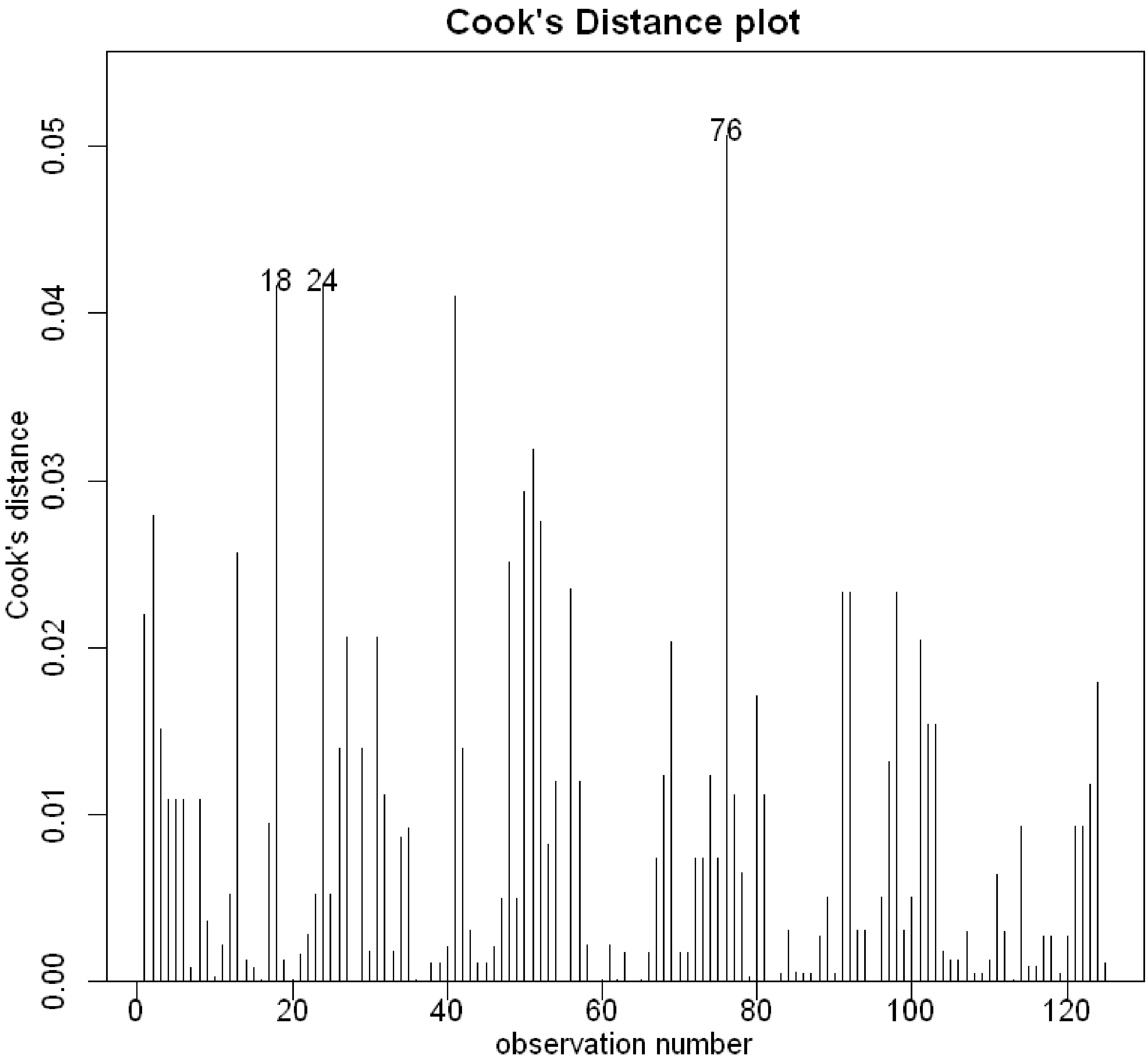
Skip to main content



Skip to main content



Skip to main content



We can trust the fitted model. What can we conclude?

```
anova(Fruitfly.fit)
```

Skip to main content

A anova: 2 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
group	4	11939.28	2984.8200	13.61195	3.515622e-09
Residuals	120	26313.52	219.2793	NA	NA

11.2. Interpreting the output

```
summary(Fruitfly.fit)
```

```
Call:
lm(formula = days ~ group, data = Fruitfly.df)

Residuals:
   Min     1Q Median     3Q    Max 
-35.76  -8.76   0.20  11.20  32.44 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   63.560      2.962  21.461  < 2e-16 ***
groupG2        1.240      4.188   0.296    0.768    
groupG3       -0.200      4.188  -0.048    0.962    
groupG4       -6.800      4.188  -1.624    0.107    
groupG5      -24.840      4.188  -5.931 2.98e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.81 on 120 degrees of freedom
Multiple R-squared:  0.3121,	Adjusted R-squared:  0.2892 
F-statistic: 13.61 on 4 and 120 DF,  p-value: 3.516e-09
```

Some researchers like to examine the group means and their deviations from the overall (or so-called“grand”) mean. 一些研究人员喜欢检查群体平均数及其与总体（或所谓的“大”）平均数的偏差。These deviations are commonly called group “effects”.

```
grand.mean <- mean(Fruitfly.df$days)
grand.mean
```

57.44

Skip to main content

The estimated group means are just the sample means within each group. 7 We can quickly obtain these using the incredibly useful `dplyr` package:估计组意味着只是在每组样本均值。我们可以快速获得这些使用非常有用的包:

```
library(dplyr())
Df <- Fruitfly.df |>
    group_by(group) |>
    summarize(group.mean = mean(days)) |>
    data.frame()
```

```
Warning message:
"程辑包'dplyr'是用R版本4.2.3 来建造的"

载入程辑包：'dplyr'


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

The estimated group means are:

```
Df$group.mean
Df$group.mean - grand.mean
```

63.56 · 64.8 · 63.36 · 56.76 · 38.72

6.12 · 7.36 · 5.92 · -0.68 · -18.72

11.3. The multiple comparisons problem

The following R code fits a simple linear regression model to iid (independent and identically distributed) normal data.

```
x <- 1:30 ## Our explanatory variable
y <- rnorm(30) ## y has NO relationship with x
summary(lm(y ~ x))$coef ## Print only the coefficient table
```

Skip to main content

A matrix: 2 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.51763352	0.37980672	-1.3628867	0.1837793
x	0.02053823	0.02139403	0.9599985	0.3452731

如果多次运行此代码，会发现大约5%的时间斜率会满足 p-value < 0.05。

Let’s get simultaneous 95% confidence intervals for all 10 comparisons via the `pairs` and `emmeans` functions of the package.

```
library(emmeans)
Fruitfly.pairs = pairs(emmeans(Fruitfly.fit, ~group, infer = T))
Fruitfly.pairs
```

```
Warning message:
"程辑包'emmeans'是用R版本4.2.3 来建造的"
```

```
 contrast estimate   SE  df t.ratio p.value
 G1 - G2     -1.24 4.19 120  -0.296  0.9983
 G1 - G3      0.20 4.19 120   0.048  1.0000
 G1 - G4      6.80 4.19 120   1.624  0.4854
 G1 - G5     24.84 4.19 120   5.931  <.0001
 G2 - G3      1.44 4.19 120   0.344  0.9970
 G2 - G4      8.04 4.19 120   1.920  0.3127
 G2 - G5     26.08 4.19 120   6.227  <.0001
 G3 - G4      6.60 4.19 120   1.576  0.5158
 G3 - G5     24.64 4.19 120   5.883  <.0001
 G4 - G5     18.04 4.19 120   4.307  0.0003

P value adjustment: tukey method for comparing a family of 5 estimates
```

12. Linear models with two explanatory factor variables (Two-way analysis of variance)

本节需要的包：

```
require(s20x)
```

Skip to main content

▶ Show code cell output

12.1. Example: Using test success and attendance to explain exam score

在第8章中，我们调查了一个学生的考试成绩对考试分数的影响是否取决于他们是否经常出席。我们发现那些经常出席的学生，比不出席者每多得到一分测试分数都有更高的“回报”。

```r
## Importing data into R
Stats20x.df <- read.table("../data/STATS20x.txt", header = TRUE)
Stats20x.df$Attend <- factor(Stats20x.df$Attend)
```

We next transform the numeric `Test` variable into a factor with two levels, `pass` and `nopass`.

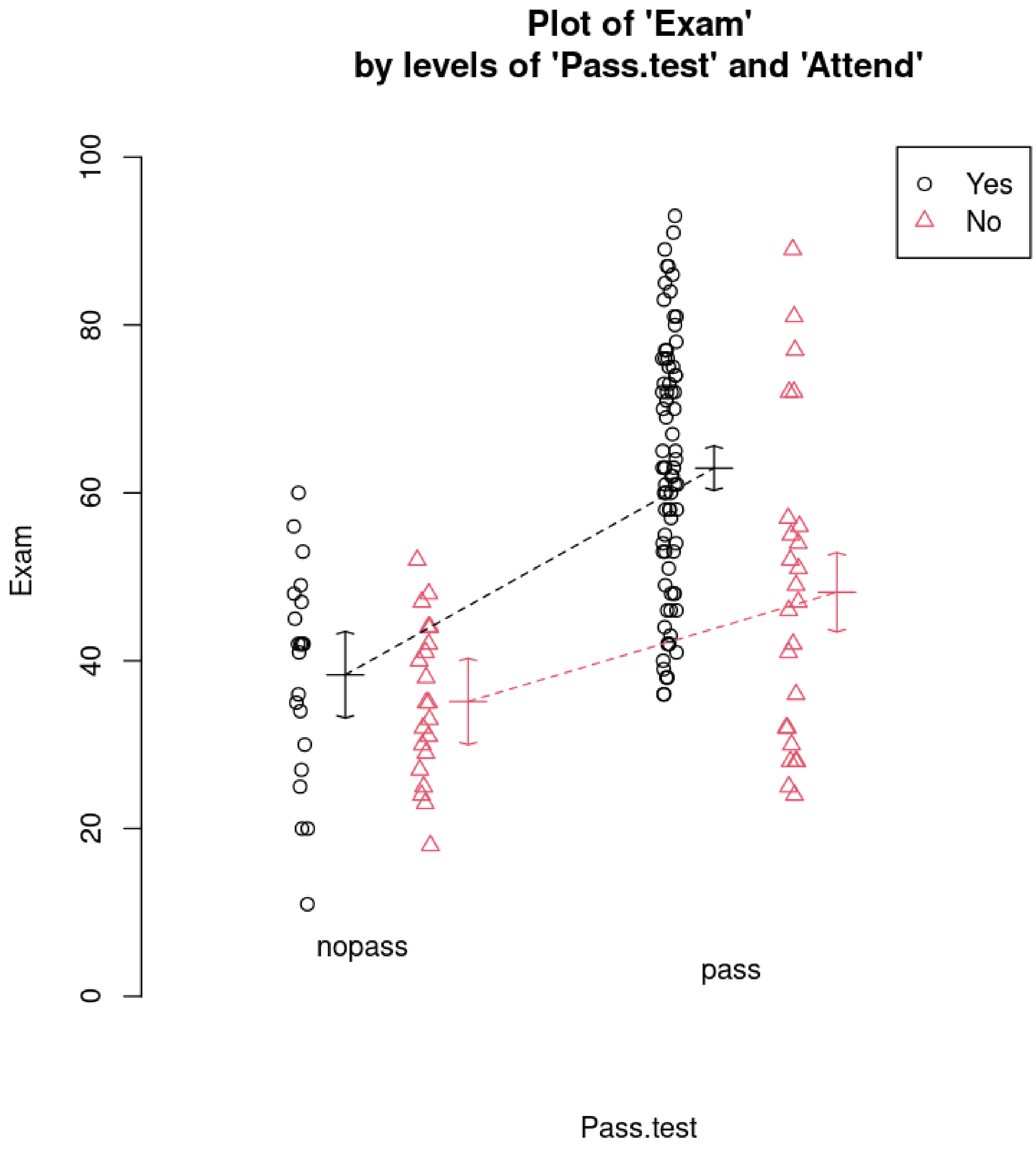
Let us create the new factor variable `Pass.test`:

```r
Stats20x.df$Pass.test <- with(
    Stats20x.df,
    factor(ifelse(Test >= 10, "pass", "nopass"))
)
## Check to see if the call above does what we expect
min <- min(Stats20x.df$Test[Stats20x.df$Pass.test == "pass"])
max <- max(Stats20x.df$Test[Stats20x.df$Pass.test == "nopass"])
cat("min =", min, ", max =", max, "\n")
```

```
min = 10 , max = 9.1
```

```r
interactionPlots(Exam ~ Pass.test + Attend, data = Stats20x.df)
```

Skip to main content

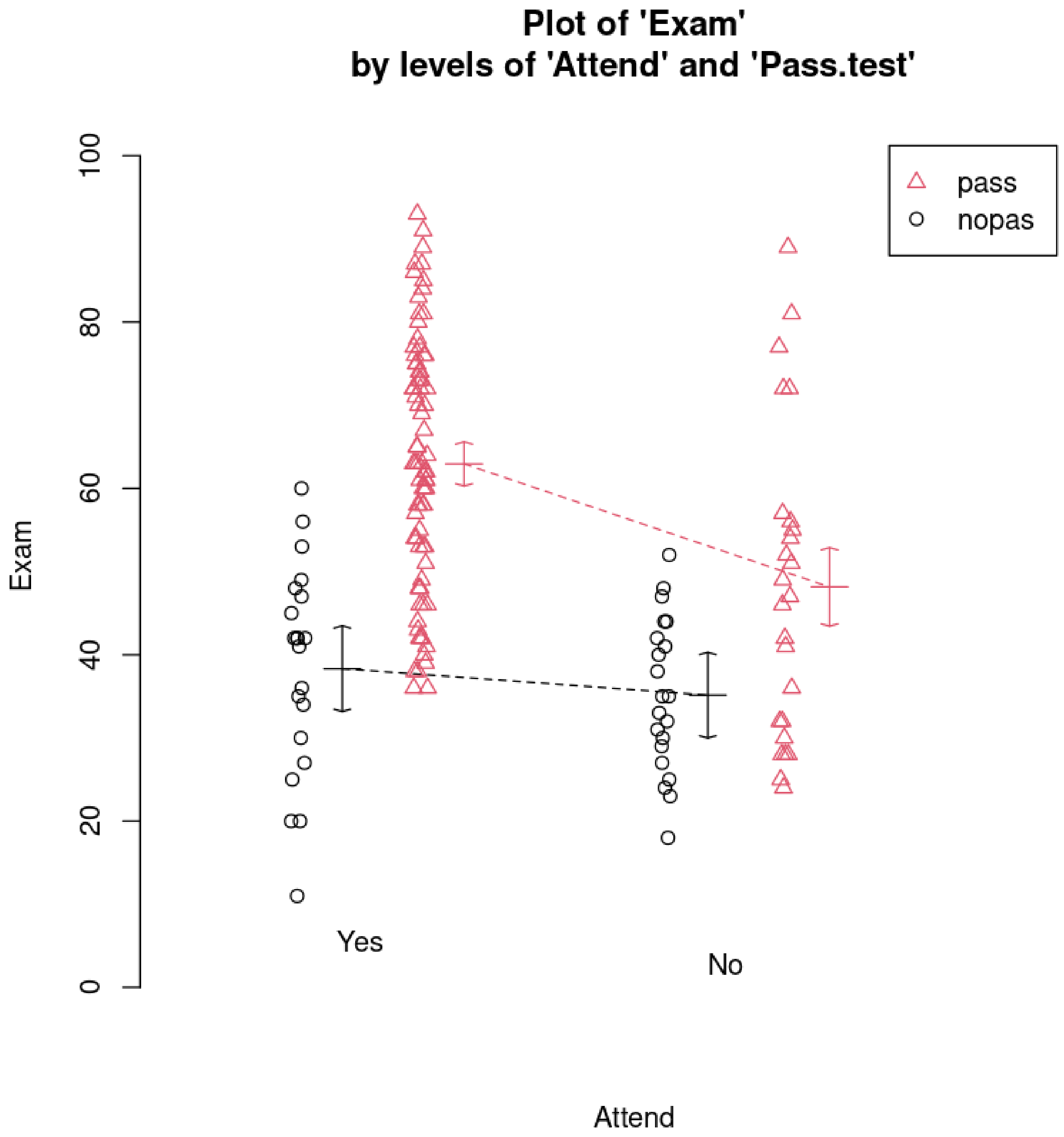


这里我们看到，“平时参与课堂”的通过测试似乎明显比大多数其他学生。请注意,我们没有平行线，从而表明可能有两个因素之间的交互。

如下所示，我们可以重新排列布局的互动情节逆转的顺序给出的解释变量是在右手边的模型公式参数。

```
# 只是交换了下位置，但是图形不一样了
```

Skip to main content



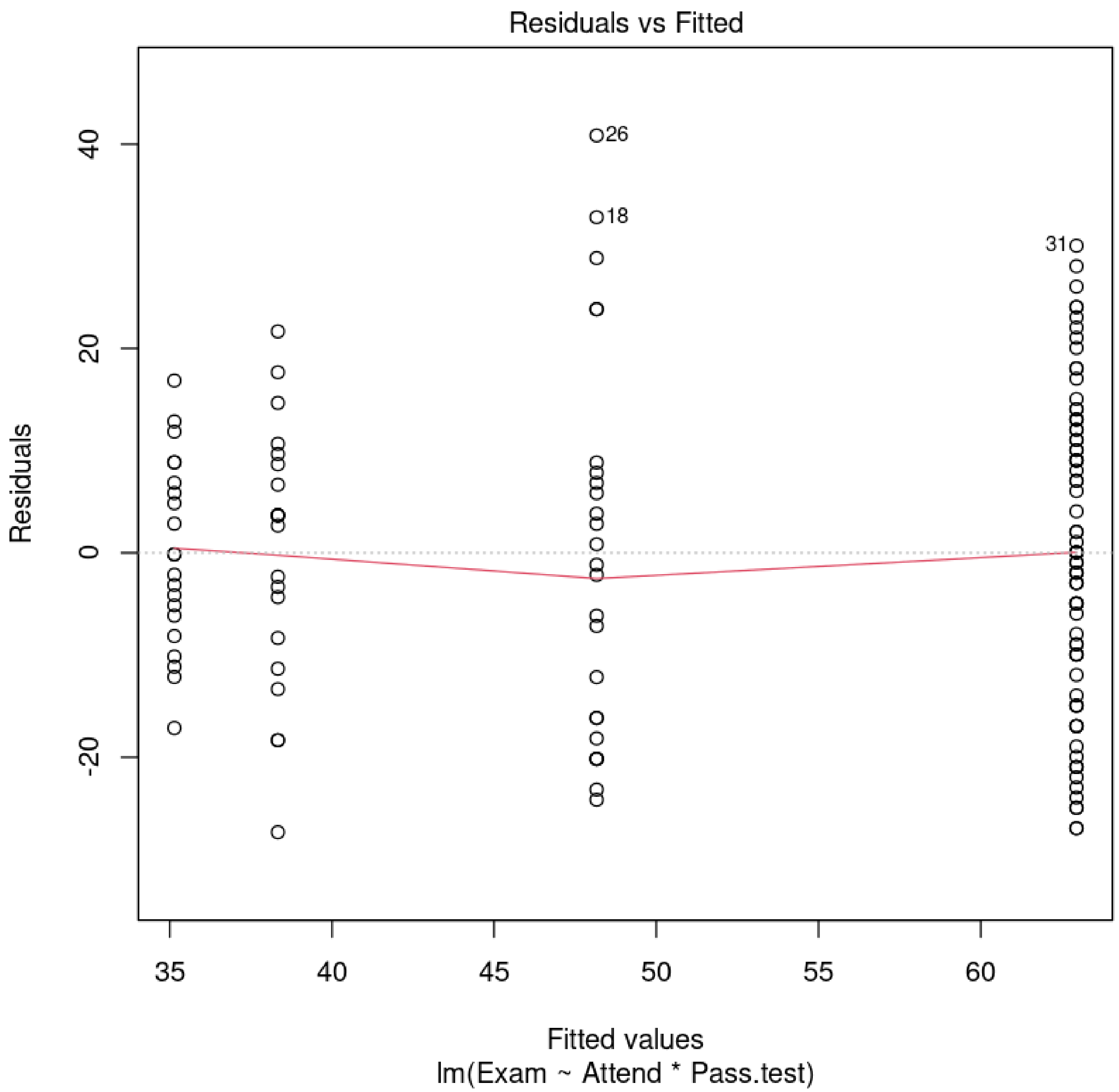
12.2. Fitting the interaction model

Let us fit the model with interaction, and check the assumptions.

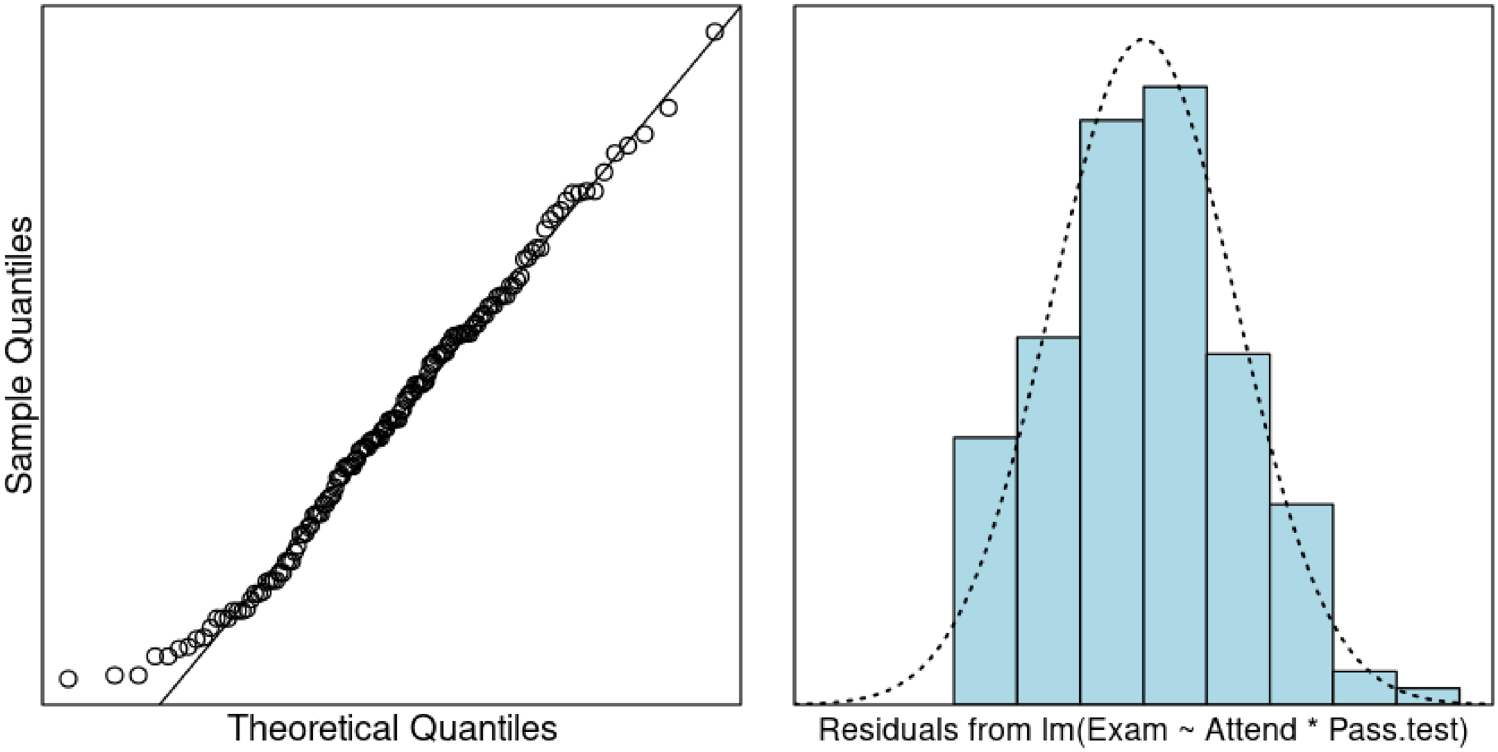
Skip to main content

```
Exam.fit <- lm(Exam ~ Attend * Pass.test, data = Stats20x.df)
# 再来最后一次经典三步走（后面的内容基本就不是标准离散正态分布的数据了）
plot(Exam.fit, which = 1)
normcheck(Exam.fit)
cooks20x(Exam.fit)
```

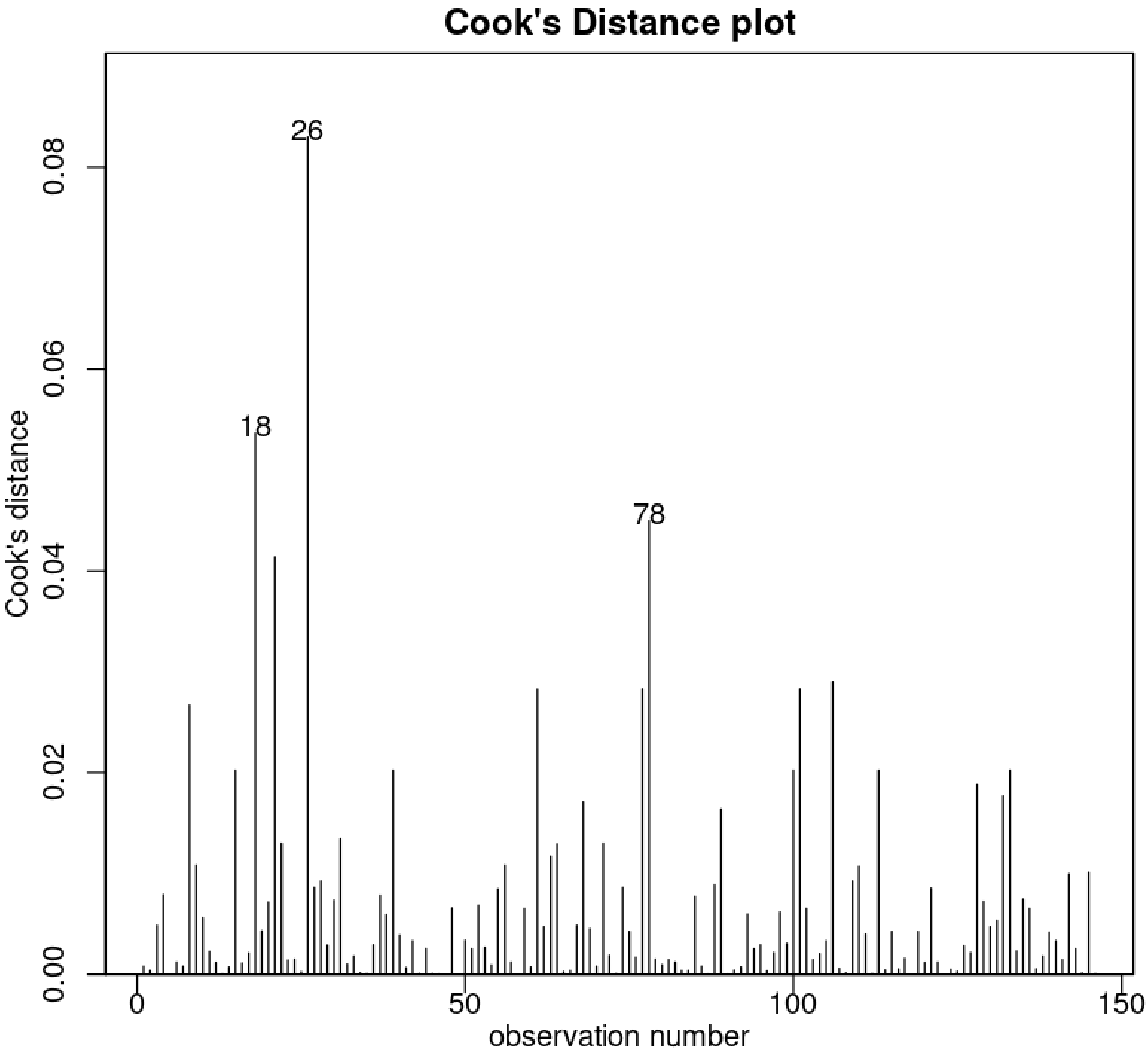
Skip to main content



Skip to main content



Skip to main content



No unduly influential data points.

We conclude that we can trust the output. Let us see what it is telling us. 我们得出结论，我们可以信任的输出。让我们看看它告诉我们什么。

```
anova(Exam.fit)
```

Skip to main content

A anova: 4 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
Attend	1	7630.7947	7630.7947	34.989635	2.364415e-08
Pass.test	1	11076.9380	11076.9380	50.791304	4.763017e-11
Attend:Pass.test	1	909.6526	909.6526	4.171048	4.297087e-02
Residuals	142	30968.3956	218.0873	NA	NA

附加：怎么通过这个算R方？

```
sum(anova(Exam.fit)$"Sum Sq"[1:3]) / sum(anova(Exam.fit)$"Sum Sq")
```

0.387804338286016

Let us investigate what our model tells us in terms of the estimated parameters:

```
summary(Exam.fit)
```

```
Call:
lm(formula = Exam ~ Attend * Pass.test, data = Stats20x.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-27.333 -10.893  -0.046   9.513  40.840 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)               35.143      3.223  10.905  < 2e-16 ***
AttendYes                  3.190      4.557   0.700  0.48504    
Pass.testpass             13.017      4.371   2.978  0.00341 ** 
AttendYes:Pass.testpass   11.599      5.679   2.042  0.04297 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.77 on 142 degrees of freedom
Multiple R-squared:  0.3878,	Adjusted R-squared:  0.3749 
F-statistic: 29.98 on 3 and 142 DF,  p-value: 4.452e-15
```

The formula for the above two-way ANOVA can be written as:

Skip to main content

$$Exam = \beta_0 + \beta_1 \times \text{Attend}_{Yes} + \beta_2 \times \text{Pass.test}_{pass} + \\ \beta_3 \times \text{Attempt}_{yes} \times \text{Pass.test}_{pass} + \varepsilon$$

where and are indicator variables, and $\varepsilon \overset{iid}{\sim} N(0, \sigma^2)$.

如前一章所示，另一个选择是去除基线的和为-1的模型公式。其他的并发症之一是,我们必须使用,而不是在指定的交互项。

12.3. Interpretting the output using pairwise differences

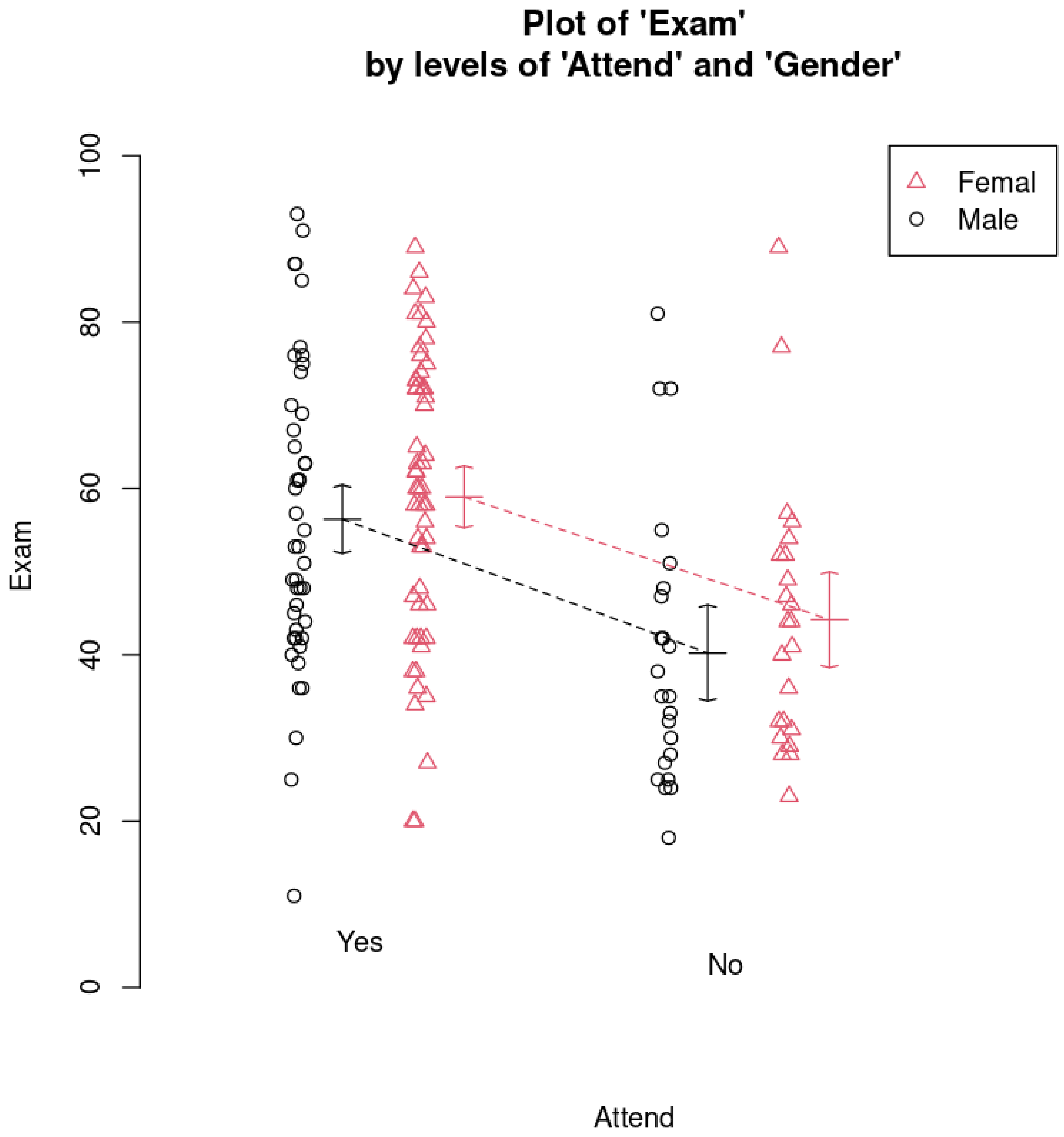
```
library(emmeans)
exam.pairs <- pairs(emmeans(Exam.fit, ~ Attend * Pass.test), infer = TRUE)
exam.pairs
```

```
 contrast                estimate   SE  df lower.CL upper.CL t.ratio p.value
 No nopass - Yes nopass     -3.19 4.56 142    -15.0     8.66  -0.700  0.8969
 No nopass - No pass       -13.02 4.37 142    -24.4    -1.65  -2.978  0.0178
 No nopass - Yes pass      -27.81 3.63 142    -37.2   -18.38  -7.669  <.0001
 Yes nopass - No pass       -9.83 4.37 142    -21.2     1.54  -2.248  0.1155
 Yes nopass - Yes pass     -24.62 3.63 142    -34.0   -15.19  -6.789  <.0001
 No pass - Yes pass        -14.79 3.39 142    -23.6    -5.98  -4.364  0.0001

Confidence level used: 0.95
Conf-level adjustment: tukey method for comparing a family of 4 estimates
P value adjustment: tukey method for comparing a family of 4 estimates
```

12.4. Example 2: Using gender and attendance to explain exam score

```
interactionPlots(Exam ~ Attend + Gender, data = Stats20x.df)
```

Skip to main content

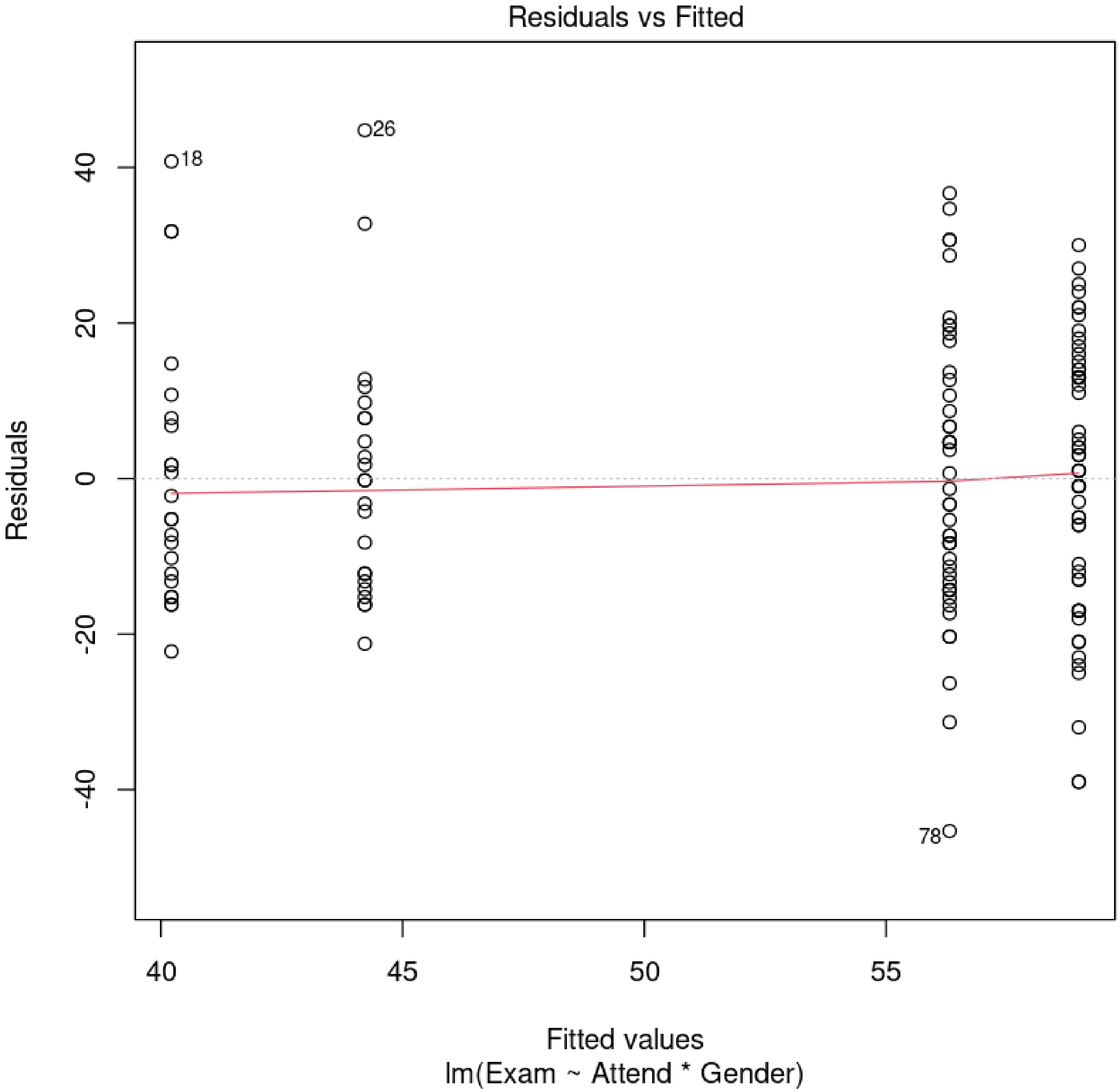


Let us fit an interaction model and check the assumptions.

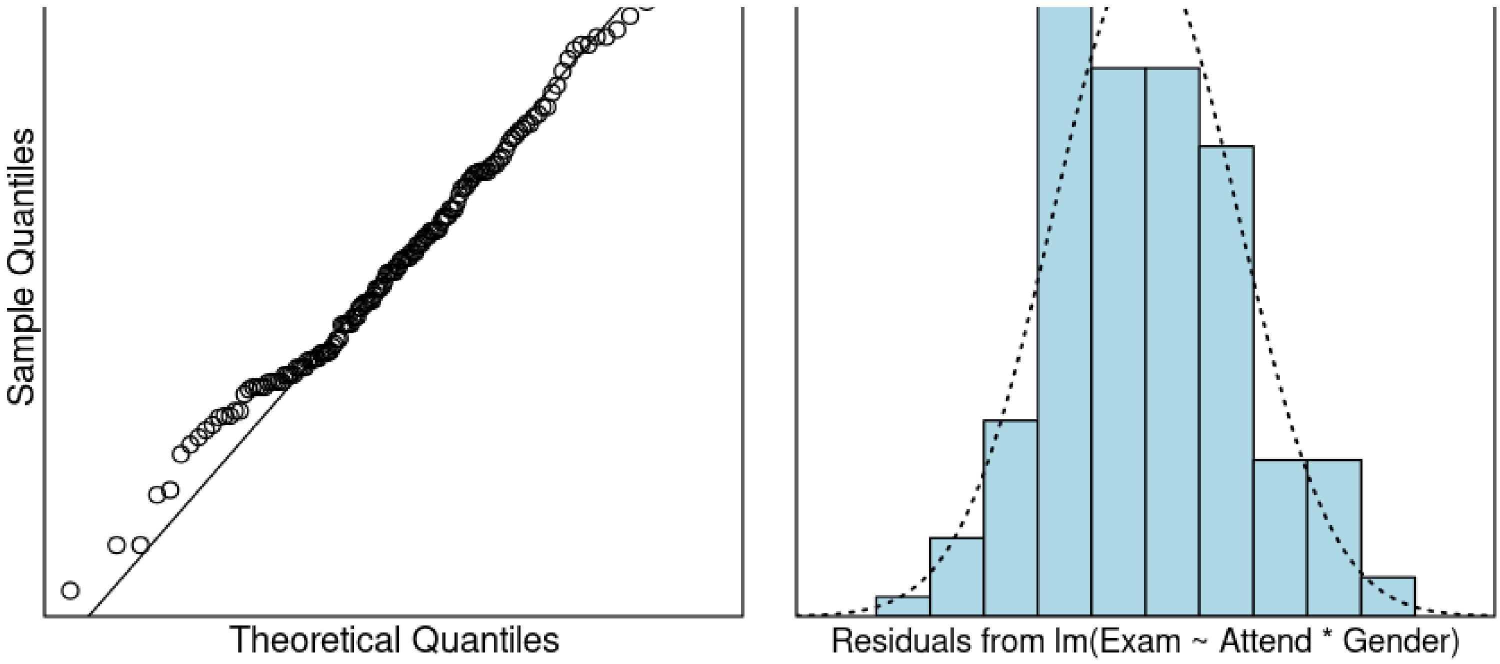
```
Exam.fit2 <- lm(Exam ~ Attend * Gender, data = Stats20x.df)
plot(Exam.fit2, which = 1)
normcheck(Exam.fit2)
cooks20x(Exam.fit2)
```

Skip to main content

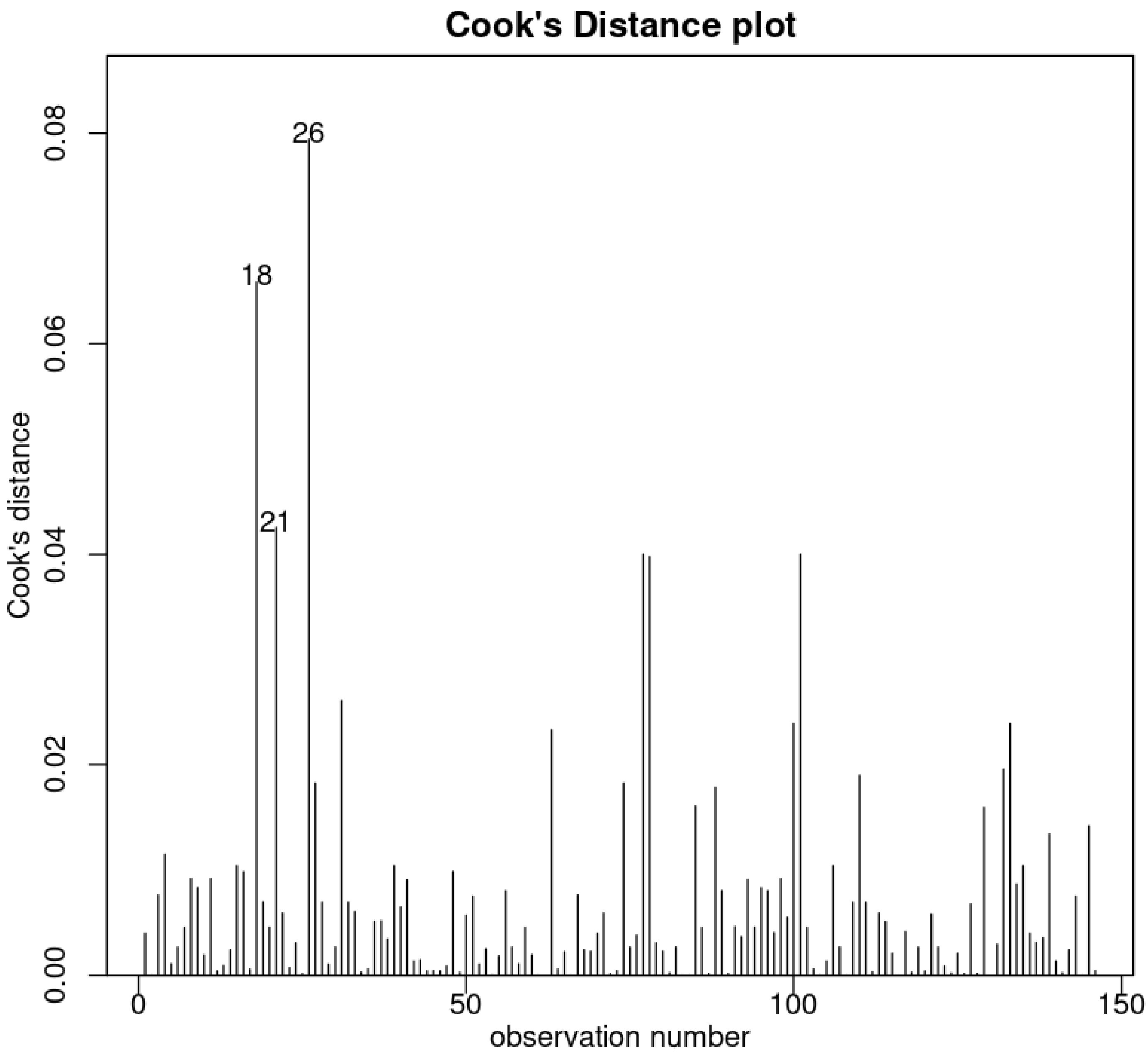
Skip to main content



Skip to main content



Skip to main content



We can trust the model. Lets see what it is telling us.

```
anova(Exam.fit2)
```

Skip to main content

A anova: 4 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
Attend	1	7630.79473	7630.79473	25.43929532	1.371947e-06
Gender	1	346.65959	346.65959	1.15568245	2.841860e-01
Attend:Gender	1	13.87415	13.87415	0.04625319	8.300248e-01
Residuals	142	42594.45235	299.96093	NA	NA

There is definitely no evidence of an interaction, so we'll apply Occam's razor and fit a simpler main-effects model (i.e., no interaction term). 这绝对是没有交互的证据，所以我们运用奥卡姆定律适合简单主效应模型(即没有交互项)。

```
Exam.fit3 <- lm(Exam ~ Attend + Gender, data = Stats20x.df)
anova(Exam.fit3)
```

A anova: 3 × 5

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
Attend	1	7630.7947	7630.7947	25.610103	1.263935e-06
Gender	1	346.6596	346.6596	1.163442	2.825689e-01
Residuals	143	42608.3265	297.9603	NA	NA

We see that the gender is also not significant here, so we again apply Occam's razor and remove this term.

```
Exam.fit4 <- lm(Exam ~ Attend, data = Stats20x.df)
summary(Exam.fit4)
```

Skip to main content

```
Call:
lm(formula = Exam ~ Attend, data = Stats20x.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-46.780 -13.108  -0.217  12.642  46.783 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   42.217      2.547  16.578  < 2e-16 ***
AttendYes     15.563      3.077   5.058 1.27e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.27 on 144 degrees of freedom
Multiple R-squared:  0.1508,	Adjusted R-squared:  0.145 
F-statistic: 25.58 on 1 and 144 DF,  p-value: 1.271e-06
```

13. Modelling count data using the Poisson distribution

本节需要的包：

```
require(s20x)
```

▶ Show code cell output

13.1. The nature of count data

在很多研究中变量将是一个计数。（非负整数）

在统计的说法，一个计数变量是离散变量的一个例子，因为它可以是离散的值分开。(相比之下，一个变量正态分布是一个连续变量，因为如果可以带任何值连续)。

In this course we shall encounter three types of count data:

- Counts of the number of “events” occurring. “事件”发生的次数

  where ideally, the events occur independently of one another and with no specific upper limit on the maximum number. 其中，在理想情况下，事件的发生是相互独立的，并且没有对最大数量的具体上限。

Skip to main content

- Counts of the number of “successes” from a fixed number of trials. 从固定数量的试验中计算出 "成功 "的数量

  E.g., the number of Heads from tossing a coin 10 times. In this case, the response variable y is the proportion of successes.

- Counts of the number of items in a category. 一个类别中的项目数的计数

  E.g., The count of A, B, and C grades in the course.

13.2. Number of R packages submitted to the Comprehensive R Archive Network (CRAN)

In the following example we will model a count variable (number of R packages submitted to the CRAN over the years) using what we know to date: the linear model via the function `lm()`.

我们将把它与一种更普遍、更合适的计数建模方式进行对比：通过函数的广义线性模型（GLM）`glm()`。

```
CRAN.df <- read.table("../data/CRAN.txt", header = TRUE)
CRAN.df
```

Skip to main content

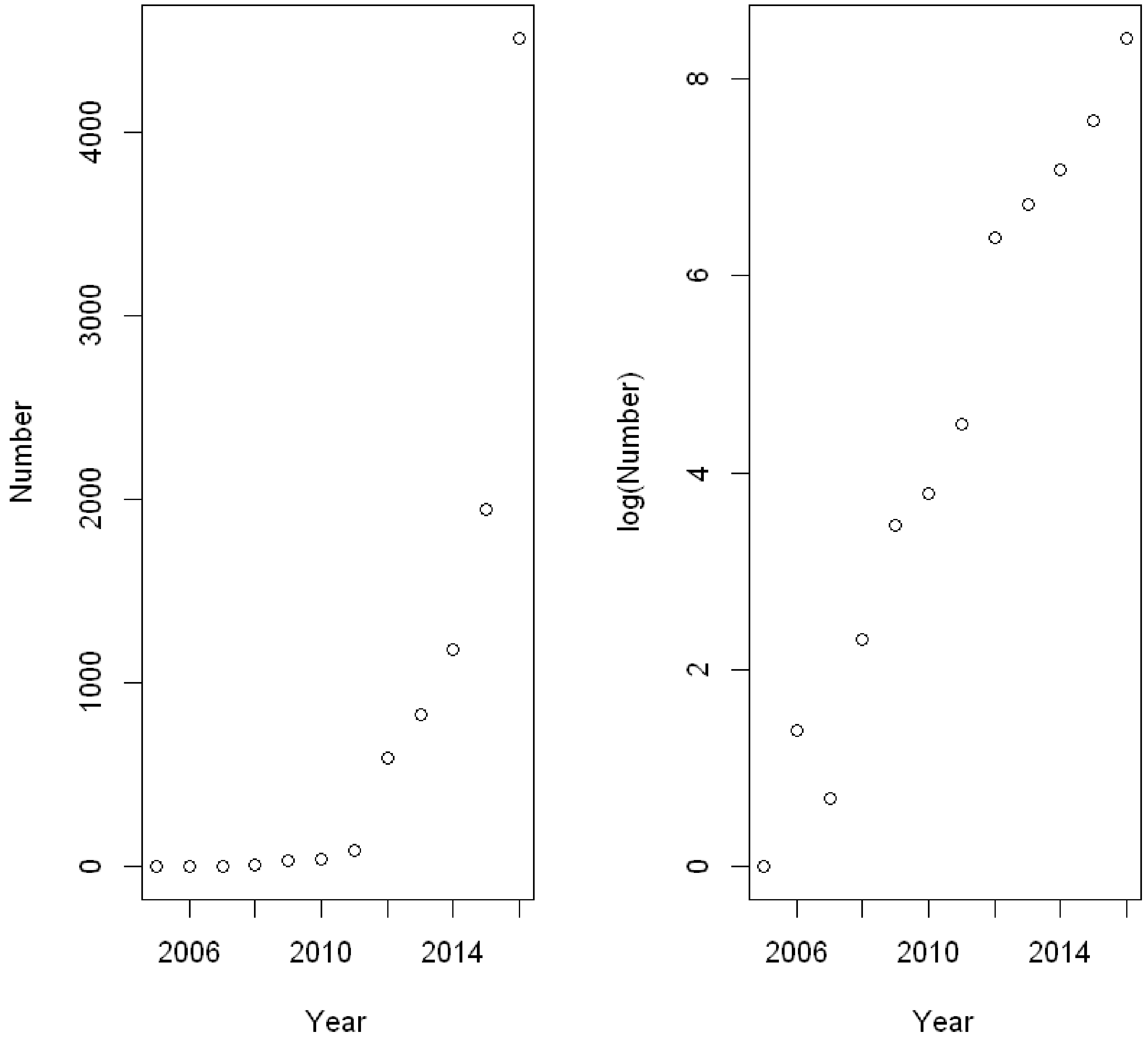
A data.frame: 12 × 2

Year	Number
<int>	<int>
2005	1
2006	4
2007	2
2008	10
2009	32
2010	44
2011	89
2012	594
2013	830
2014	1185
2015	1944
2016	4512

Let's plotting the data

```r
## One-by-two figure layout
par(mfrow = c(1, 2))
## Scatter plot using raw y
plot(Number ~ Year, data = CRAN.df)
## Scatter plot using log y
plot(log(Number) ~ Year, data = CRAN.df)
```

Skip to main content



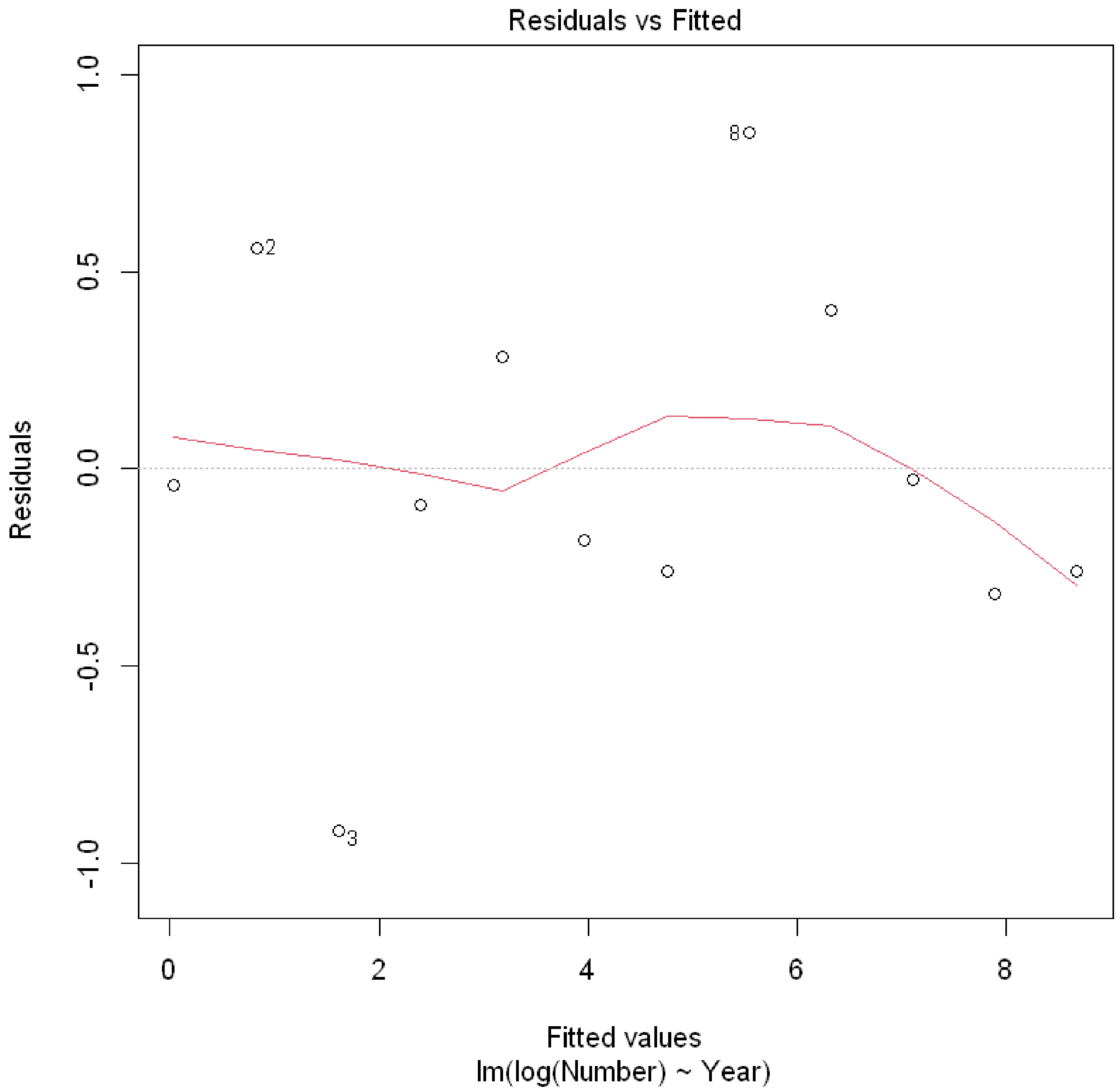
The relationship between year and number of submissions looks reasonably linear on the log scale, so we’ll fit a linear model to `log(Y)`.

注意：凡是做了 log 处理的数据，都是估计的中位数，而不是平均数。

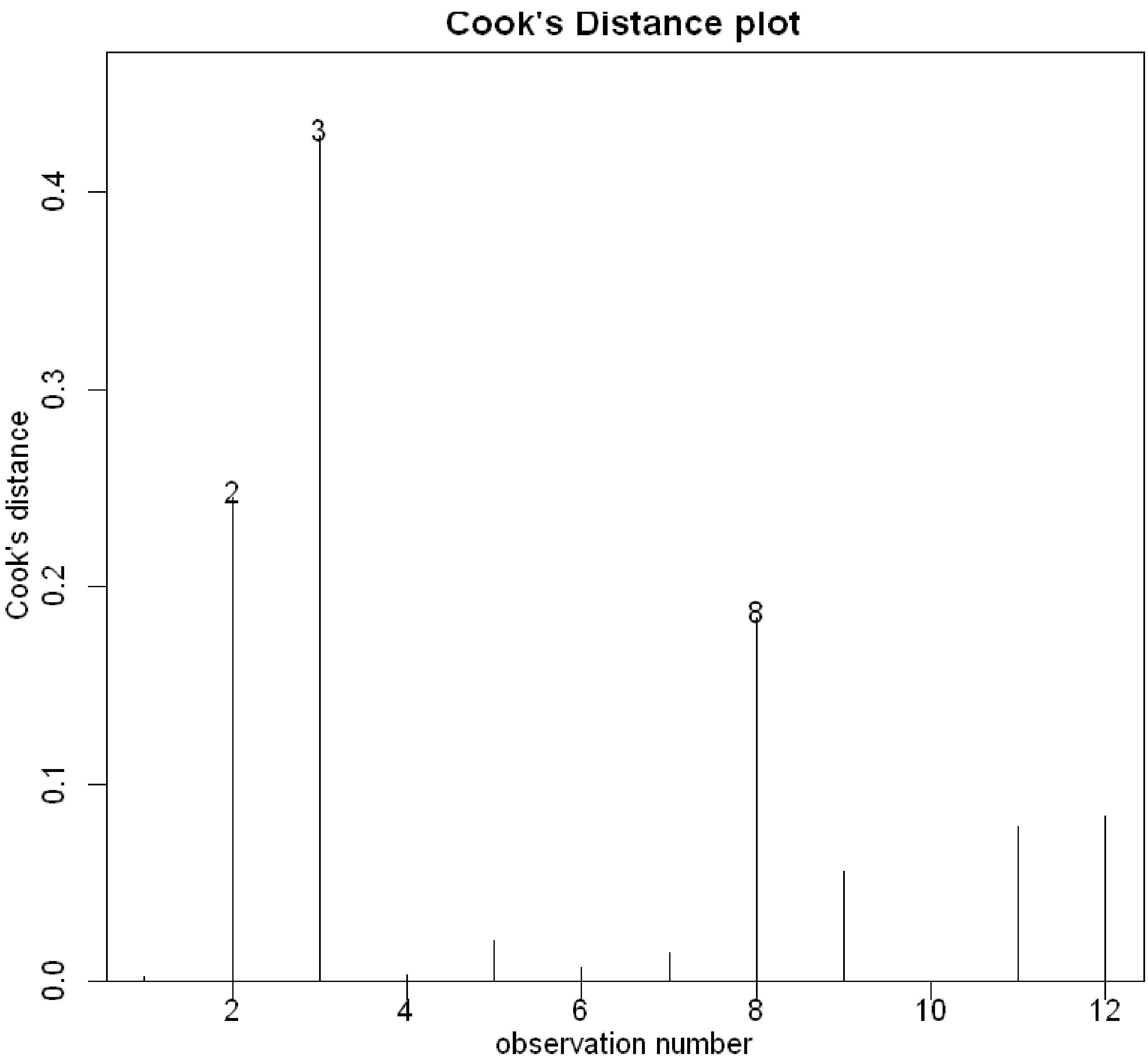
Skip to main content

```r
CRAN.fit <- lm(log(Number) ~ Year, data = CRAN.df)
plot(CRAN.fit, which = 1)
# 还有一句被我吞了，我说的
cooks20x(CRAN.fit)
```

Skip to main content



Skip to main content



The Cook’s distance for observation 3 exceeds our threshold of 0.4. However, with only 12 observations this is perhaps not of great concern. 观察3的库克距离超过了我们的阈值0.4。然而，由于只有12个观测值，这也许并不值得关注。

```
summary(CRAN.fit)
```

Skip to main content

```
Call:
lm(formula = log(Number) ~ Year, data = CRAN.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.9174 -0.2607 -0.0668  0.3143  0.8517 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.574e+03  8.245e+01  -19.09 3.39e-09 ***
Year         7.849e-01  4.101e-02   19.14 3.30e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4904 on 10 degrees of freedom
Multiple R-squared:  0.9734,	Adjusted R-squared:  0.9708 
F-statistic: 366.4 on 1 and 10 DF,  p-value: 3.295e-09
```

Back-transform to get the multiplicative effect of year. 注意这里有写 exp，还原了数据

```
## Estimated annual multiplier
exp(CRAN.fit$coef["Year"])
## Confidence interval
exp(confint(CRAN.fit))
```

Year: 2.19223728471024

A matrix: 2 × 2 of type dbl

	2.5 %	**97.5 %**
(Intercept)	0.00000	0.00000
Year	2.00081	2.40198

So, the Executive Summary would have said that the median annual number of submissions to CRAN multiplies by between 2.00 to 2.40 times each year. In other words, it increases by between 100% and 140% per annum. 因此，执行摘要替代，相较于头一年，每年提交给CRAN的中位数是2.00到2.40倍之间。换句话说，它每年增加 100% 到 140% 之间。

```
predCRAN.df <- data.frame(Year = 2017)
## Prediction on the log scale
pred2017 <- predict(CRAN.fit, predCRAN.df, interval = "confidence")
## Back-transform for the median of the number of submissions in 2017
exp(pred2017)
```

Skip to main content

A matrix: 1 × 3 of type dbl

	fit	lwr	upr
1	12833.05	6550.586	25140.85

So, the Executive Summary would have said that the median of the number of submissions to CRAN in 2017 is between 6550 and 25100.

"如果你只有一把锤子，你往往会把每个问题都看成是钉子。"（亚伯拉罕•马斯洛）

【我们将 "锤子" 改为 "线性模型"，将 "钉子" 改为 "正态分布"】

13.3. The Poisson Distribution

The Poisson is a distribution that takes values on the non-negative { 0,1,2,3,... } integers and it has no upper limit. 泊松是一种在非负{0,1,2,3，...}整数上取值的分布，它没有上限。

The probability that the non-negative integer value y will be observed if generated by a Pois(μ) distribution is given by the following formula 如果由POIS(μ)分布生成，则观察到非负整数值y的概率由以下公式给出:

$$\Pr(y) = \frac{\exp(-\mu)\mu^y}{y!}$$

where $y! = \text{factorial(y)} = 1 \times 2 \times \ldots \times \left(y - 1\right) \times y\left(\text{and } 0! = 1\right)$

For y = 12 and μ = 9.61, this could be calculated in using the code:

```
y <- 12
mu <- 9.61
(exp(-mu) * mu^y) / factorial(y)
```

0.0868507787370292

In R, the in-built function `dpois(y, μ)` calculates these Poisson probabilities. E.g., the probability that y = 12 will be observed from a Pois(9 61) distribution is

```
dpois(12, 9.61)
```

Skip to main content

```
0.0868507787370291
```

You can also generate random Poisson values. E.g., here are 20 random values from a Pois(10) distribution,

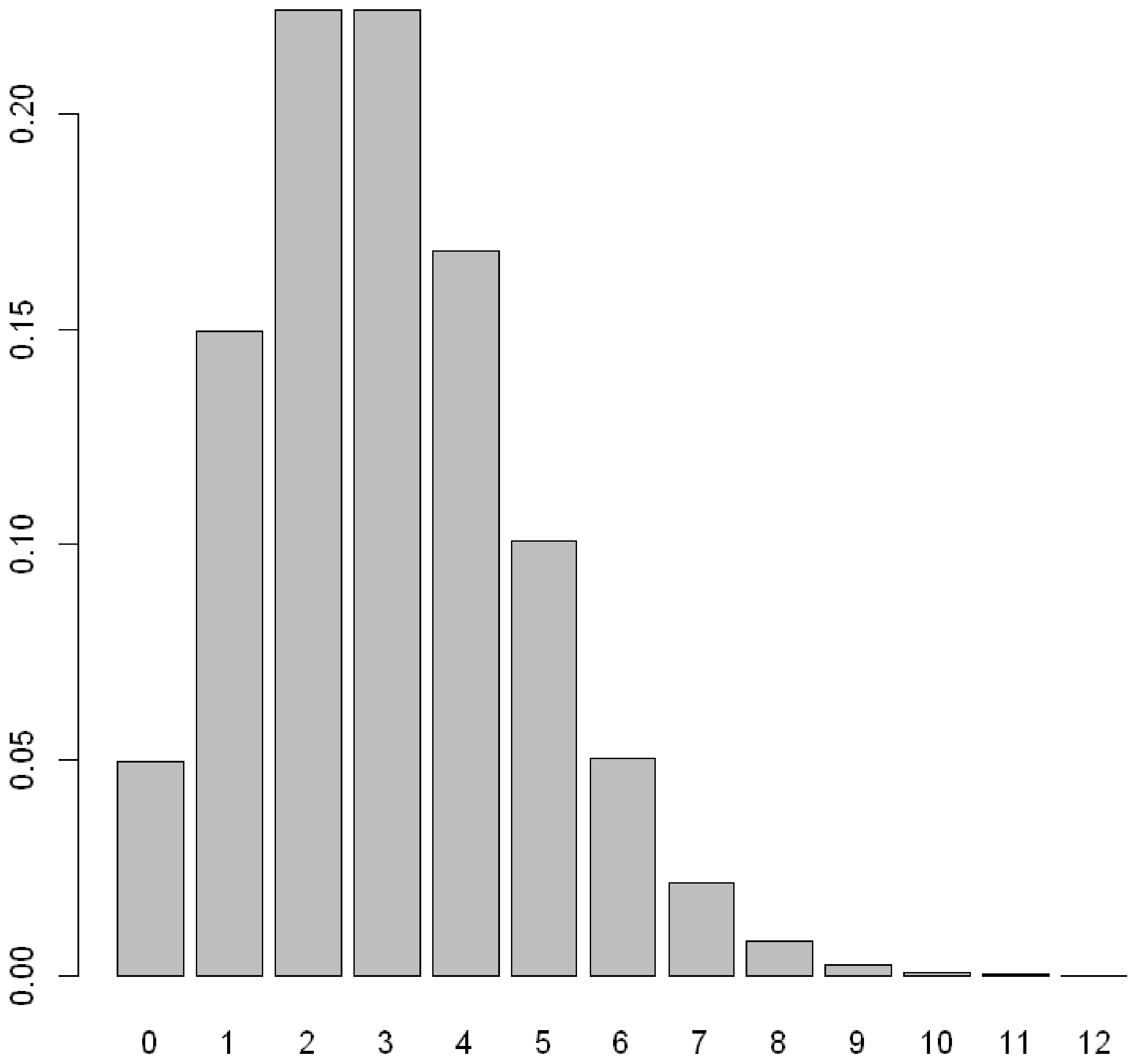
```r
rpois(20, 10)
```

```
8 · 10 · 9 · 16 · 9 · 16 · 10 · 11 · 5 · 15 · 11 · 13 · 7 · 11 · 10 · 6 · 11 · 4 · 8 · 11
```

假设任何一个星期五晚上的实际人数可以用平均数为3的泊松分布（Pois(3)）来描述，那么酒精中毒者人数的分布是怎样的？

```r
barplot(dpois(0:12, 3), names = 0:12)
```

Skip to main content



由图可以看出泊松分布可能会有长尾、右偏效应。

The probabilities for the number of victims from 0 to 20 are:

```
round(dpois(0:20, 3), 6)
```

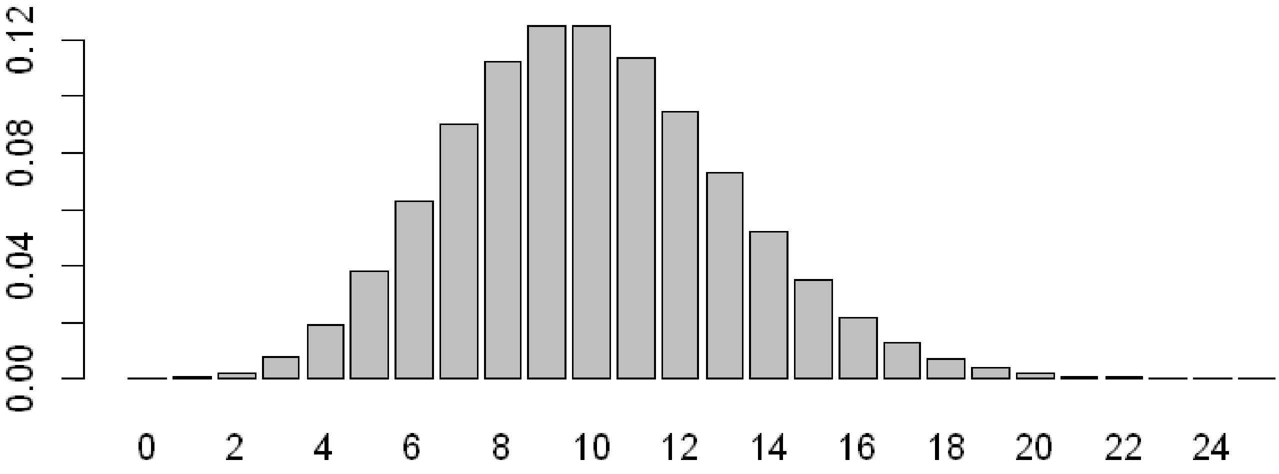
0.049787 · 0.149361 · 0.224042 · 0.224042 · 0.168031 · 0.100819 · 0.050409 · 0.021604 · 0.008102 · 0.002701 · 0.00081 · 0.000221 · 5.5e-05 · 1.3e-05 · 3e-06 · 1e-06 · 0 · 0 · 0 · 0 · 0

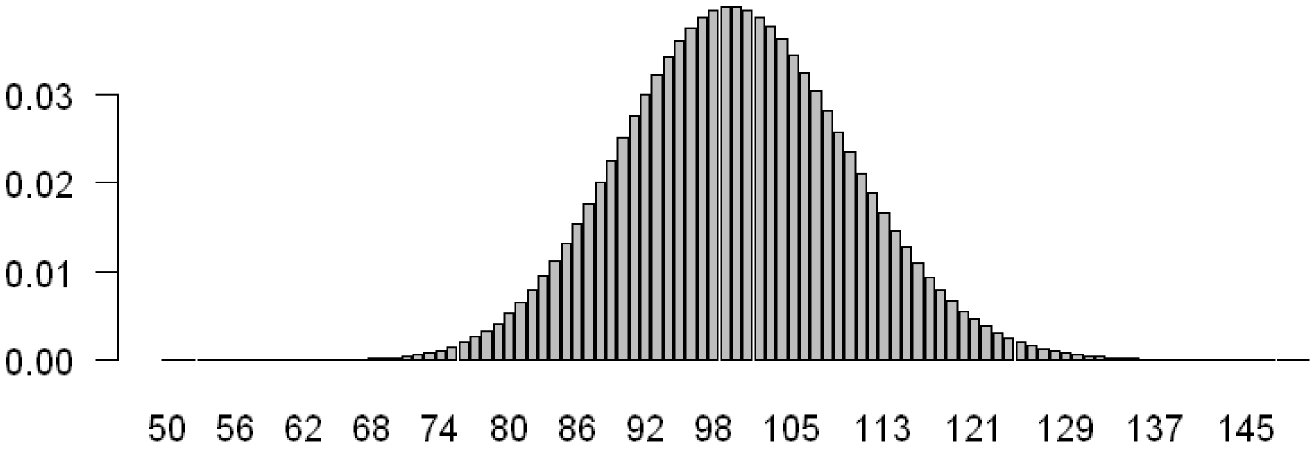
但如果均值很大呢？或者说只要足够大，泊松分布就会变成正态分布。

More Poisson distributions:

```r
par(mfrow = c(2, 1)) ## Two-by-One figure layout
barplot(dpois(0:25, 10), names = 0:25) ## Pois(10)
barplot(dpois(50:150, 100), names = 50:150, las = 1) ## Pois(100)
```

Skip to main content





Variability increases with the mean μ. In fact, the variance of a Pois(μ) is also μ. That is, if Y is Pois(μ) distributed then 如果Y是Pois(μ)分布的，则

$$Var(Y) = E(Y) = \mu$$

The Poisson distribution is right-skewed for small values of μ, but looks very much like a (discretised)

Skip to main content

散的）正态分布。

一般来说我们以 30 为界限，30 往后就差不多，30往内称两者有较大差异。

13.4. The generalized linear model (GLM)

广义线性模型。

$$Y \sim \text{Poisson}(\mu)$$

where the Poisson parameter ($\mu = E[Y]$) changes with respect to an x variable as follows:

$$\mu = \exp(\beta_0 + \beta_1 x)$$

注意：由于变形上是对 x 进行变形而不是 y，所以我们估计的就不是中位值了，而是均值。

The relationship = $\mu = \exp(\beta_0 + \beta_1 x)$ can equivalently be expressed as 等效的表示为

$$\log(\mu) = \log E[Y|x] = \beta_0 + \beta_1 x$$

```r
CRAN.gfit <- glm(Number ~ Year, family = poisson, data = CRAN.df)
summary(CRAN.gfit)
```

Skip to main content

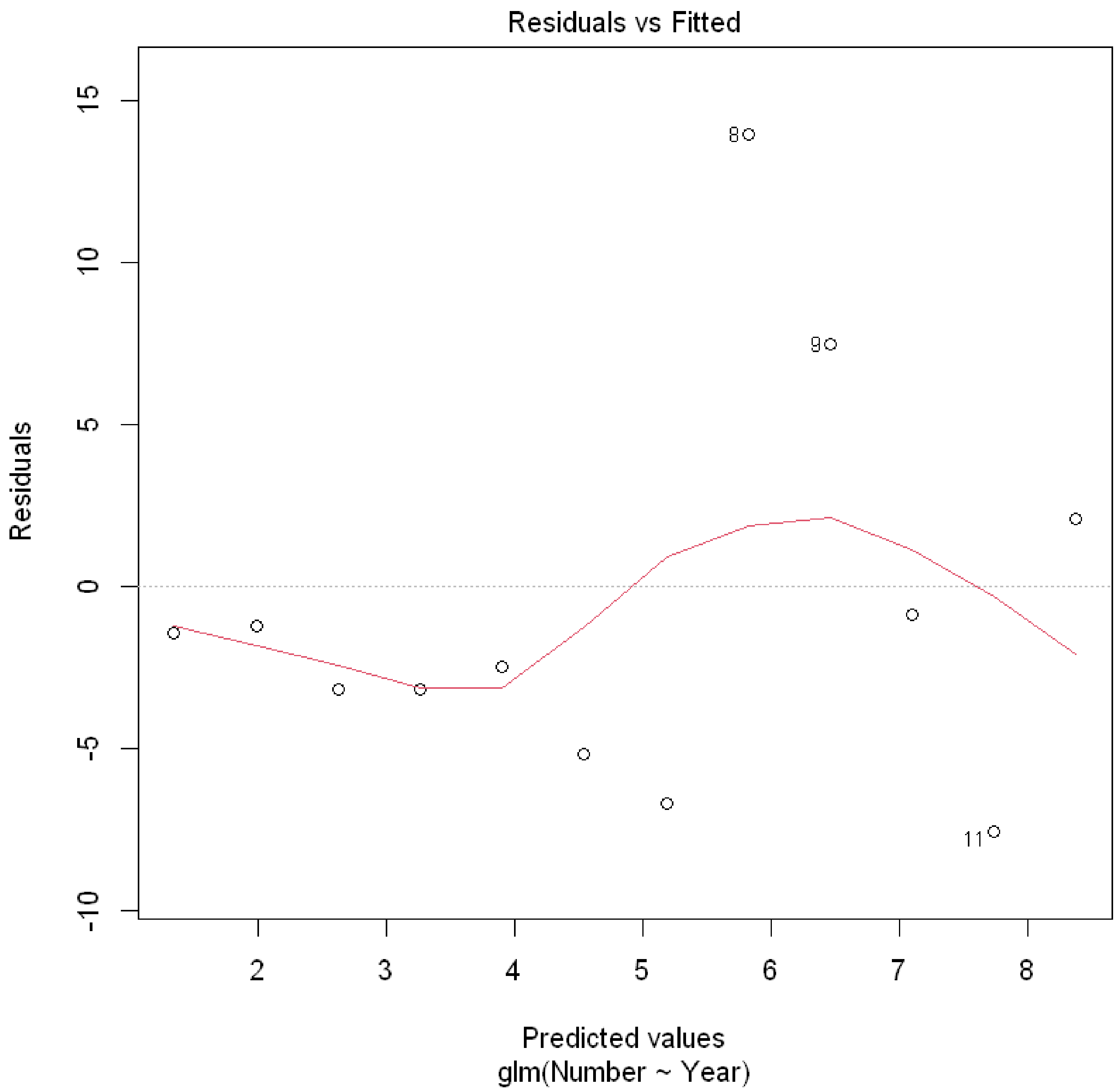
```
Call:
glm(formula = Number ~ Year, family = poisson, data = CRAN.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.7487  -4.4233  -2.1953  -0.1449  12.5634

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.282e+03  1.384e+01  -92.64   <2e-16 ***
Year         6.401e-01  6.868e-03   93.20   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 19374.51  on 11  degrees of freedom
Residual deviance:   402.61  on 10  degrees of freedom
AIC: 481.27

Number of Fisher Scoring iterations: 4
```

坏了，这下 R 方消失了，只有自由度。别慌，后面继续讲。

First, check the residuals to see if they look random.

```r
plot(CRAN.gfit, which = 1)
```

Skip to main content



It is not as easy to check the assumptions of a GLM compared to a linear model.

There is another that is essential: Checking the Poisson assumption that the variances of the counts are equal to their means. 检查泊松假设，即计数的方差与它们的平均值相等。

In this example, the residual deviance is 402.61, with 10 df. The P-value is

Skip to main content

```
1 - pchisq(402.61, 10)
```

0

If this P-value is small then we conclude that our model is not adequate. That is clearly the case with the CRAN data. 如果这个P值很小，那么我们的结论是，我们的模型是不充分的。这显然是CRAN数据的情况。

Without quasi-Poisson correction:

```
CRAN.gfit <- glm(Number ~ Year, family = poisson, data = CRAN.df)
summary(CRAN.gfit)
```

```
Call:
glm(formula = Number ~ Year, family = poisson, data = CRAN.df)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-7.7487  -4.4233  -2.1953  -0.1449  12.5634  

Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -1.282e+03  1.384e+01  -92.64   <2e-16 ***
Year         6.401e-01  6.868e-03   93.20   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 19374.51  on 11  degrees of freedom
Residual deviance:   402.61  on 10  degrees of freedom
AIC: 481.27

Number of Fisher Scoring iterations: 4
```

对比一下使用 “quasipoisson”（卡方泊松，类泊松）的模型：

```
CRAN.quasigfit <- glm(Number ~ Year, family = quasipoisson, data = CRAN.df)
summary(CRAN.quasigfit)
```

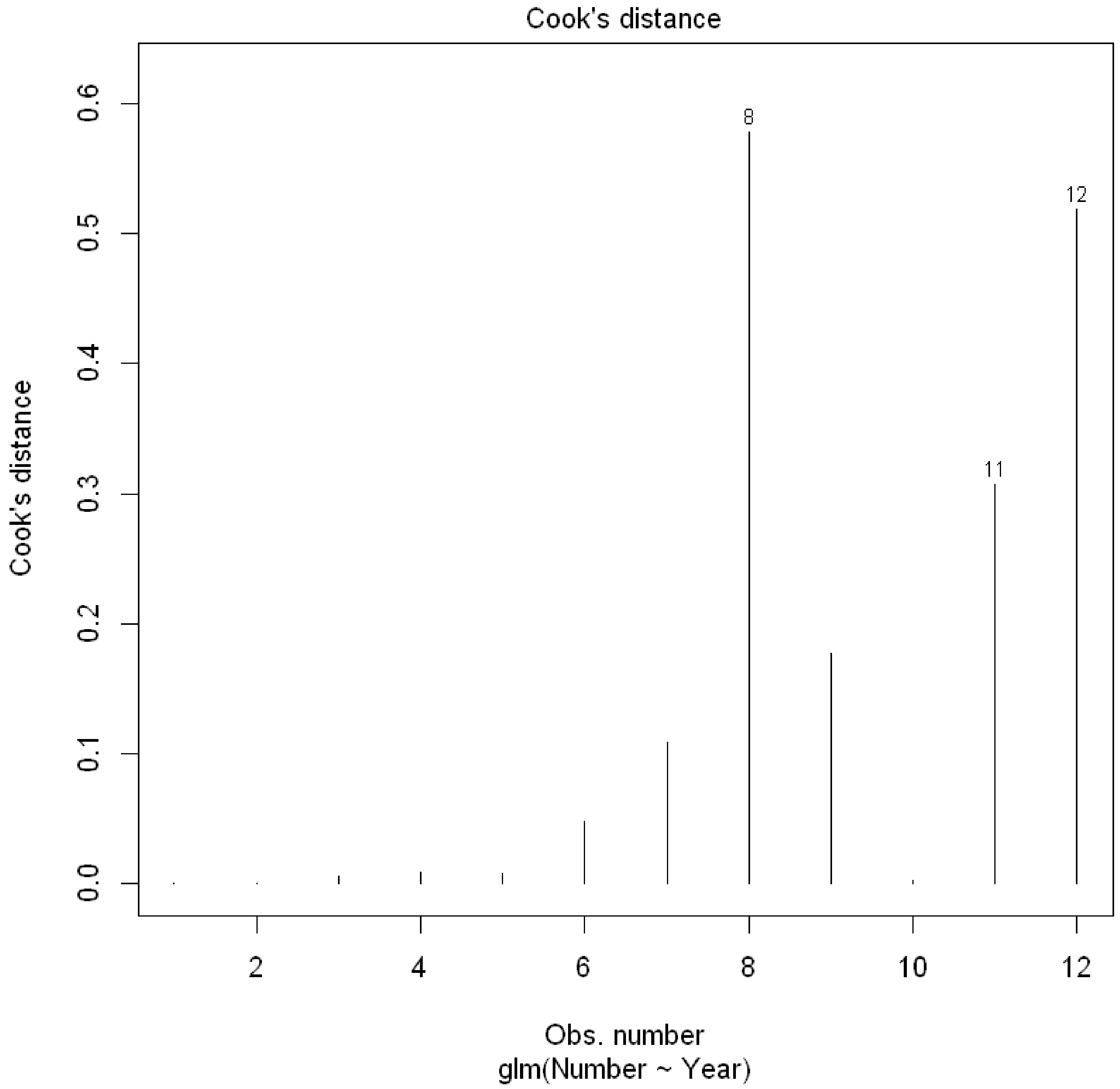
Skip to main content

```
Call:
glm(formula = Number ~ Year, family = quasipoisson, data = CRAN.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.7487  -4.4233  -2.1953  -0.1449  12.5634

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.282e+03  8.889e+01  -14.42 5.09e-08 ***
Year         6.401e-01  4.411e-02   14.51 4.81e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 41.25925)

    Null deviance: 19374.51  on 11  degrees of freedom
Residual deviance:   402.61  on 10  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4
```

Let’s check the influence of the observations.

```
plot(CRAN.quasigfit, which = 4)
```

Skip to main content



我们看到，观察点8和12超过了我们的0.4门槛。然而，必须谨慎地解释这一点。与正常的线性回归不同，在泊松回归中，预计μ值较高的观测值会比μ值较低的观测值有更大的影响。

13.5. Interpretting the GLM output

Recall that our model for the expected number of submissions to the CRAN is

Skip to main content

$$\mu = \exp(\beta_0 + \beta_1 \times Year) = \exp(\beta_0) \times \exp(\beta_1)^{Year}$$

so we need to exponentiate our estimate and its confidence interval. 所以我们需要对我们的估计值及其置信区间进行指数化。

```
## The estimated annual multiplier
exp(CRAN.quasigfit$coef["Year"])

# get confidence intervals for the parameters.
exp(confint(CRAN.quasigfit))
# Sometimes you might see `confint.default()` used instead.
exp(confint.default(CRAN.quasigfit))
```

Year: 1.89661418646857

```
Waiting for profiling to be done...
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	0.000000	0.000000
Year	1.745819	2.075781

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	0.000000	0.000000
Year	1.739517	2.067898

That is, it increases by between 75% and 108% per year.

> 请注意，GLM模型是对均值而不是对中位值进行陈述的。 这是因为GLM没有对y进行转换。

14. Poisson modelling of count data: Two examples

本节需要的包：

Skip to main content

```
require(s20x)
```

▶ Show code cell output

14.1. Example 1: Earthquake frequency

古腾堡•里希特定律说，在一定时期内，预期的地震数量会随着地震的大小而成倍地减少。其公式为：

$$\log_{10} N = a - bM$$

where N is the expected number of earthquakes of magnitude M or more on the Richter scale. Here, a and b are unknown parameters. 其中N是里氏震级M级以上的地震的预期数量。这里，a和b是未知参数。

```
Quakes.df <- read.table("../data/EarthquakeMagnitudes.txt", header = TRUE)
Quakes.df$Locn <- as.factor(Quakes.df$Locn)
# Print first 4 SC observations
subset(Quakes.df, subset = c(Locn == "SC"))[1:4, ]
# Print first 4 WA observations
subset(Quakes.df, subset = c(Locn == "WA"))[1:4, ]
```

Skip to main content

A data.frame: 4 × 3

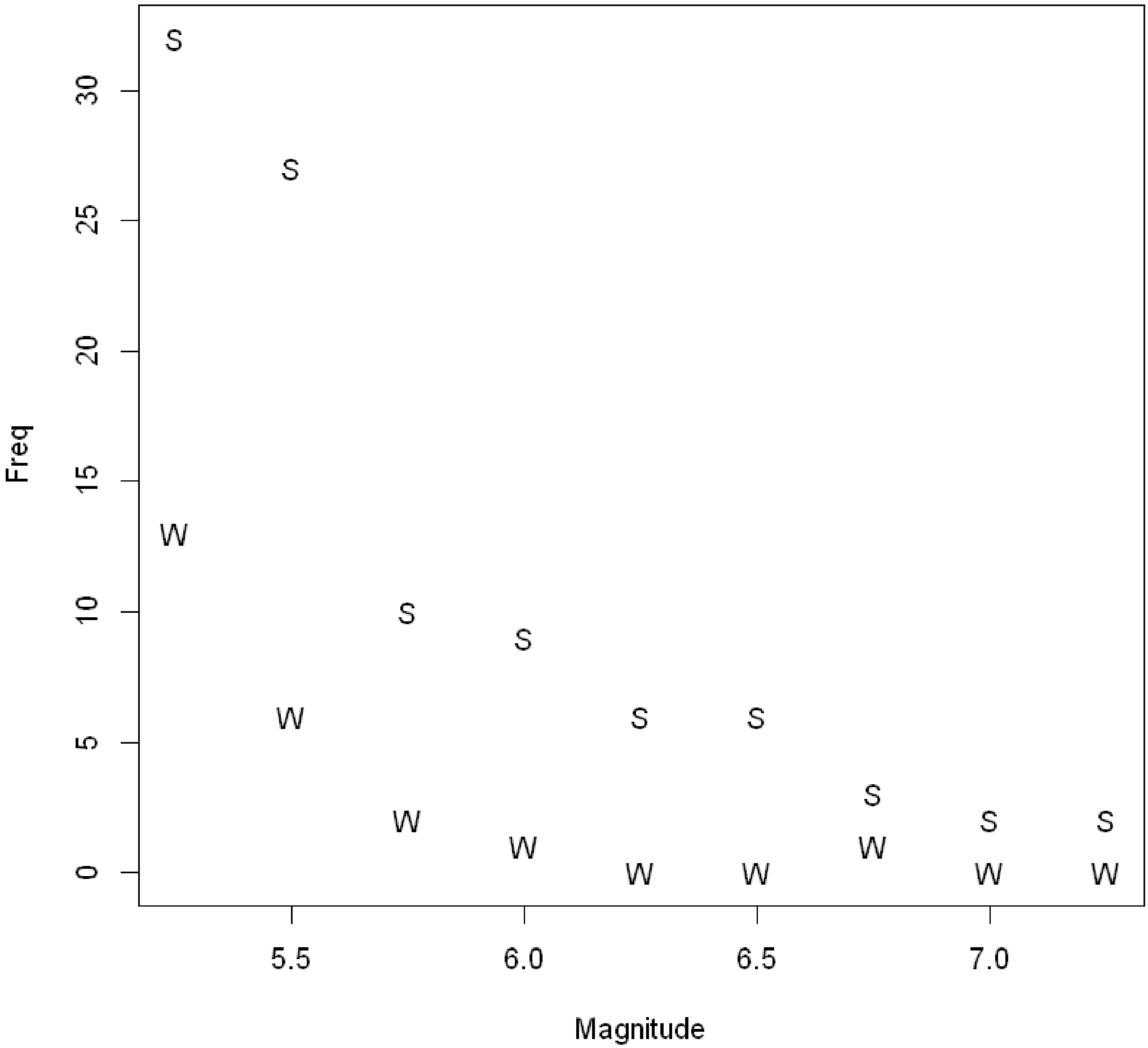
	Locn	Magnitude	Freq
	<fct>	<dbl>	<int>
1	SC	5.25	32
2	SC	5.50	27
3	SC	5.75	10
4	SC	6.00	9

A data.frame: 4 × 3

	Locn	Magnitude	Freq
	<fct>	<dbl>	<int>
10	WA	5.25	13
11	WA	5.50	6
12	WA	5.75	2
13	WA	6.00	1

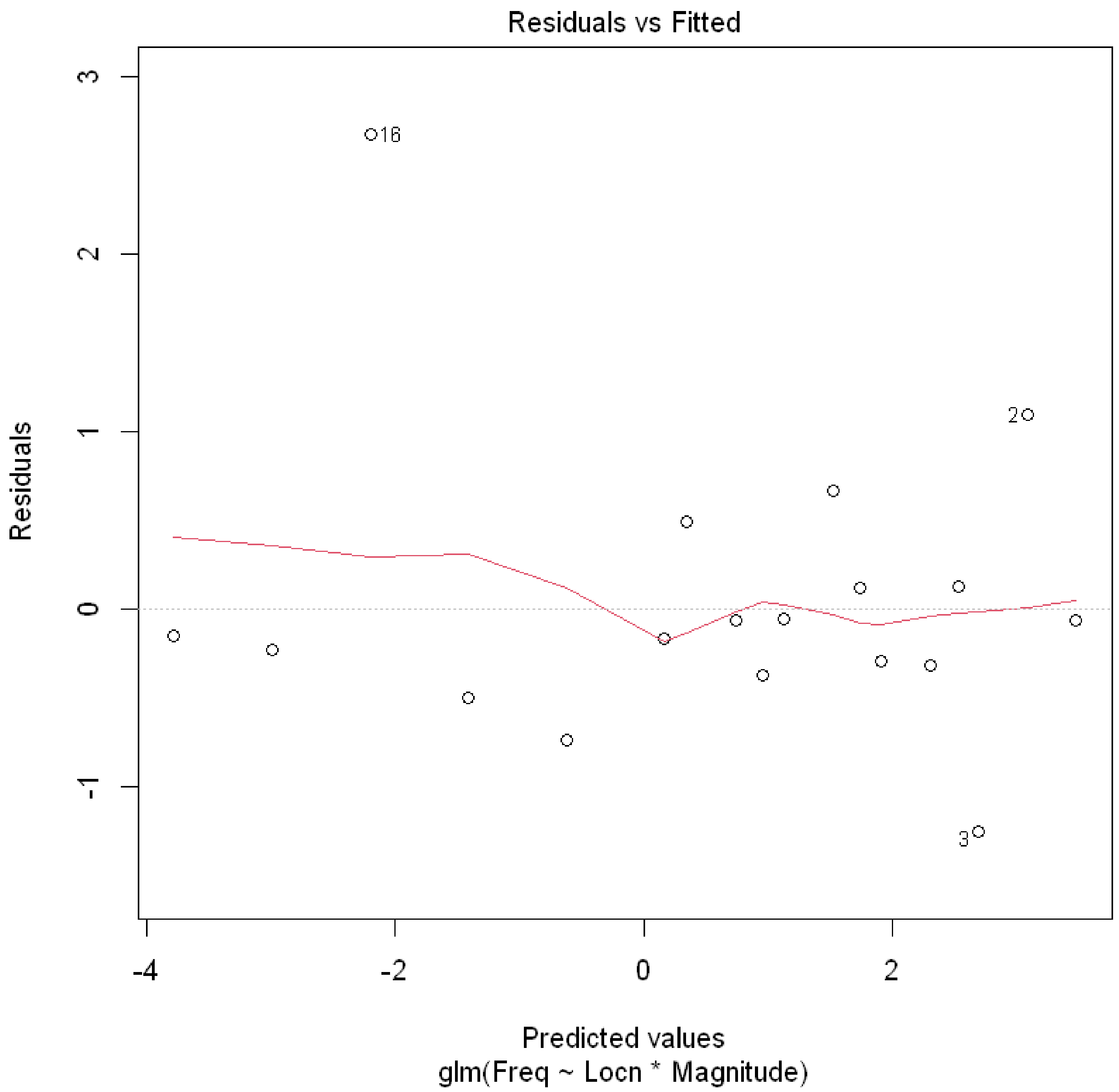
```r
plot(Freq ~ Magnitude, data = Quakes.df, pch = substr(Locn, 1, 1))
```

Skip to main content



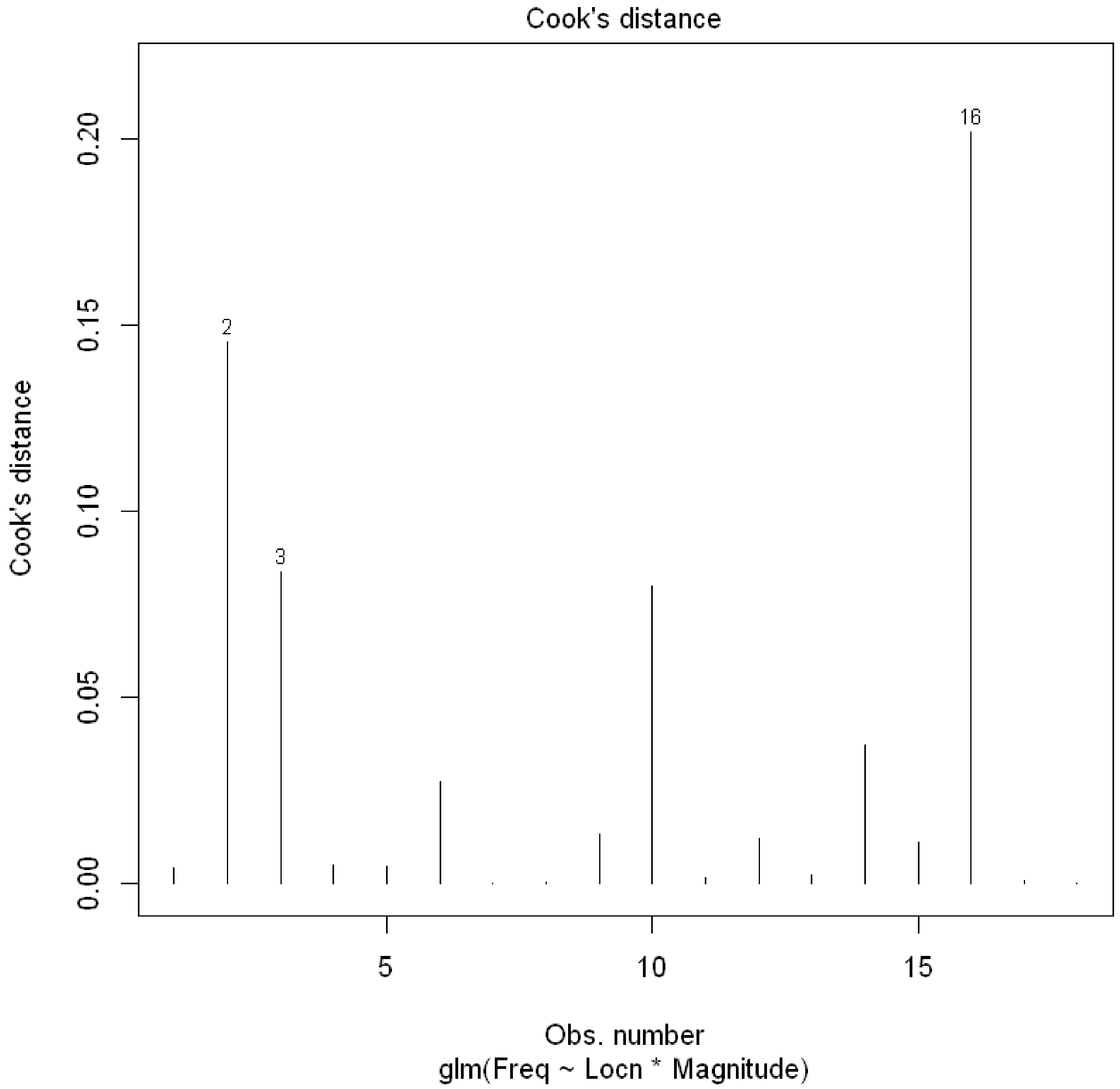
```
Quake.gfit <- glm(
    Freq ~ Locn * Magnitude,
    family = poisson,
    data = Quakes.df
)
plot(Quake.gfit, which = 1)
```

Skip to main content



```
plot(Quake.gfit, which = 4)
```

Skip to main content



```
summary(Quake.gfit)
```

Skip to main content

```
Call:
glm(formula = Freq ~ Locn * Magnitude, family = poisson, data = Quakes.df)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-1.3261  -0.3225  -0.1172   0.1241   1.6190  

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)       11.6923     1.1762   9.941  < 2e-16 ***
LocnWA             7.3923     3.9500   1.871   0.0613 .  
Magnitude         -1.5648     0.2055  -7.616 2.61e-14 ***
LocnWA:Magnitude  -1.5884     0.7199  -2.206   0.0274 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 176.1767  on 17  degrees of freedom
Residual deviance:   8.2295  on 14  degrees of freedom
AIC: 65.11

Number of Fisher Scoring iterations: 5
```

```r
1 - pchisq(8.23, 14)
```

0.877002515280767

```r
Quake.cis <- confint(Quake.gfit)
exp(Quake.cis[3, ])
## To interpret as percentage decreases
100 * (1 - exp(Quake.cis[3, ]))
```

```
Waiting for profiling to be done...
```

2.5 %: 0.137474251413585 **97.5 %:** 0.308243734868518

2.5 %: 86.2525748586415 **97.5 %:** 69.1756265131482

Skip to main content

```
Quakes.df$Locn2 <- factor(Quakes.df$Locn, levels = c("WA", "SC"))

Quake2.gfit <- glm(Freq ~ Locn2 * Magnitude, family = poisson, data = Quakes.df)
(Quake.WA.ci <- exp(confint(Quake2.gfit)[3, ]))

## To interpret as percentage decreases
100 * (1 - Quake.WA.ci)
```

```
Waiting for profiling to be done...
```

2.5 %: 0.00907766110939229 **97.5 %:** 0.140175444866894

2.5 %: 99.0922338890608 **97.5 %:** 85.9824555133106

14.2. Example 2: Snapper counts in and around marine reserves

```
Snap.df <- read.table("../data/SnapperCROPvsHAHEI.txt", header = TRUE)
with(Snap.df, {
    Locn <- as.factor(Locn)
    Reserve <- as.factor(Reserve)
})
Snap.df
```

Skip to main content

A data.frame: 18 × 3

Locn	Reserve	Freq
<chr>	<chr>	<int>
Leigh	N	2
Leigh	N	1
Leigh	N	0
Leigh	Y	5
Leigh	Y	11
Leigh	Y	7
Leigh	Y	8
Leigh	Y	7
Leigh	Y	14
Hahei	N	1
Hahei	N	0
Hahei	N	1
Hahei	N	0
Hahei	Y	3
Hahei	Y	2
Hahei	Y	1
Hahei	Y	5
Hahei	Y	3

15. Modelling proportion data using the binomial distribution

本节需要的包：

```
require(s20x)
```

Skip to main content

15.1. Binary (Bernoulli) data, odds and log-odds

Here we are considering the situation where the response can only take two possible values. These might be coded in the form of:

- Zeros or ones.
- TRUE or FALSE.
- Yes or No.
- Success or Failure.
- Or any other pair of categorical values.

Bernoulli random variables(伯努利随机变量):

If Y is a Bernoulli random variable with parameter p, then Y will take the value 1 with probability p, and the value 0 with probability 1 - p. Since it is a probability, p must be a value that is between 0 and 1, i.e. $p \in [0, 1]$.

很容易表明，伯努利随机变量的平均值为：

$$E(Y) = p$$

而且，方差为

$$Var(Y) = p(1-p)$$

15.1.1. Odds

为了使 glm 对数几率的解释更加容易，我们需要引入 Odds 这个概念，把定义域从 $[0, 1]$ 扩展到 $[0, \infty]$。

Odds 是指事件发生的概率与事件不发生的概率的比值，即：

$$Odds = \frac{p}{1-p}$$

反过来说：

Skip to main content

$$p = \frac{Odds}{1 + Odds}$$

15.1.2. Log-odds

Odds 的对数（简称为 log-odds）为：

$$LogOdds = log(\frac{p}{1-p})$$

此时定义域从 $[0, \infty]$ 变为 $[-\infty, \infty]$。

$$p = \frac{exp(LogOdds)}{1 + exp(LogOdds)}$$

15.2. Modelling log-odds

Why log-odds?

由于我们的模型对或的值没有限制，所以在设置变量为 β_0、β_1 的情况下，$\beta_0 + \beta_1 x$ 可以在实线上取任何值。

That is, $\beta_0 + \beta_1 x \in (-\infty, \infty)$.

$$\text{Log-Odds} = \beta_0 + \beta_1 x$$

Log-Odds can be any real number.

即：

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x$$

其中p是解释变量x的主体 "成功" 的概率。

Skip to main content

$$p = \frac{exp(\beta_0 + \beta_1 x)}{1 + exp(\beta_0 + \beta_1 x)}$$

15.3. Modelling the response when it is binary (ungrouped data) via glm

```
bb.df <- read.csv("../data/basketball.csv")
head(bb.df, 10)
```

A data.frame: 10 × 3

	distance	gender	basket
	<int>	<chr>	<int>
1	3	M	1
2	1	F	1
3	2	M	1
4	3	M	0
5	1	M	1
6	2	F	1
7	2	F	1
8	1	F	1
9	3	F	0
10	1	F	1

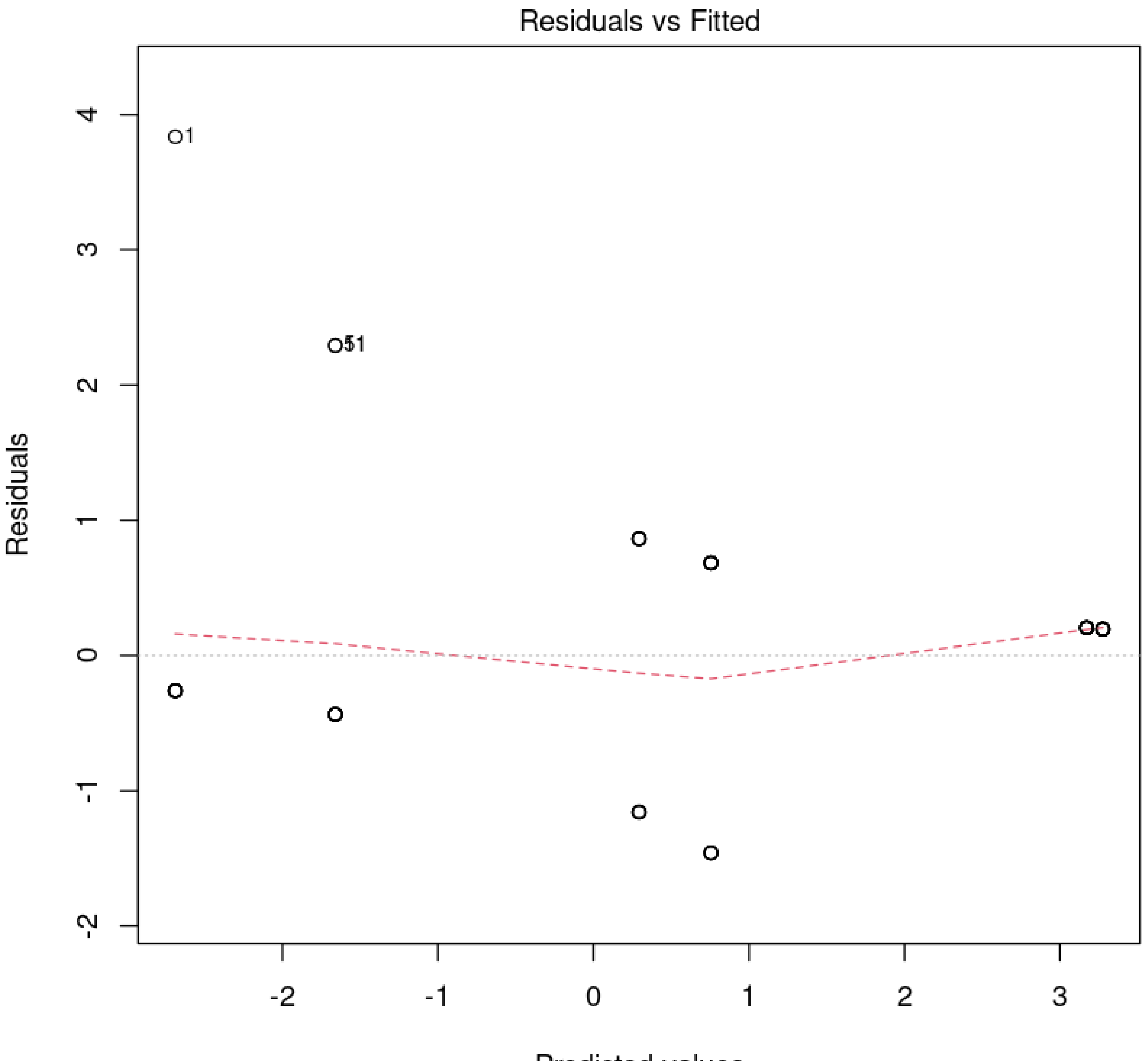
```
success.tbl <- xtabs(basket ~ distance + gender, data = bb.df)
success.tbl
```

```
        gender
distance  F  M
       1 10 10
       2  6  5
```

Skip to main content

```r
bb.fit <- glm(
    basket ~ distance * gender,
    family = binomial, # binomial distribution
    data = bb.df
)
```

```r
plot(bb.fit, which = 1, lty = 2)
```



Skip to main content

```r
summary(bb.fit)
```

```
Call:
glm(formula = basket ~ distance * gender, family = binomial,
    data = bb.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5106  -0.5900   0.2723   0.2866   2.3474

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)        5.5878     1.9050   2.933  0.00335 **
distance          -2.4159     0.8181  -2.953  0.00314 **
genderM            0.6710     2.9235   0.230  0.81847
distance:genderM  -0.5668     1.3213  -0.429  0.66794
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82.108  on 59  degrees of freedom
Residual deviance: 46.202  on 56  degrees of freedom
AIC: 54.202

Number of Fisher Scoring iterations: 5
```

```r
bb.fit1 <- glm(basket ~ distance + gender,
    family = binomial, data = bb.df
)
summary(bb.fit1)
```

Skip to main content

```
Call:
glm(formula = basket ~ distance + gender, family = binomial,
    data = bb.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5382  -0.5411   0.2461   0.3219   2.2283

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   6.1469     1.5242   4.033 5.51e-05 ***
distance     -2.6648     0.6364  -4.188 2.82e-05 ***
genderM      -0.5478     0.7486  -0.732    0.464
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82.108  on 59  degrees of freedom
Residual deviance: 46.392  on 57  degrees of freedom
AIC: 52.392

Number of Fisher Scoring iterations: 5
```

```r
bb.fit2 <- glm(basket ~ distance, family = binomial, data = bb.df)
summary(bb.fit2)
```

```
Call:
glm(formula = basket ~ distance, family = binomial, data = bb.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4118  -0.4818   0.2873   0.2873   2.1029

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   5.7980     1.4038   4.130 3.63e-05 ***
distance     -2.6310     0.6274  -4.193 2.75e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82.108  on 59  degrees of freedom
Residual deviance: 46.937  on 58  degrees of freedom
AIC: 50.937

Number of Fisher Scoring iterations: 5
```

Skip to main content

```
coef(bb.fit2)
```

(Intercept): 5.79796774980361 **distance:** -2.63103340427345

```
exp(coef(bb.fit2))
100 * (1 - exp(coef(bb.fit2)))
```

(Intercept): 329.628990177678 **distance:** 0.0720040145217848

(Intercept): -32862.8990177678 **distance:** 92.7995985478215

```
(bb.ci2 <- confint(bb.fit2))
100 * (1 - exp(bb.ci2))
```

```
Waiting for profiling to be done...
```

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	3.422396	9.037020
distance	-4.103523	-1.568945

A matrix: 2 × 2 of type dbl

	2.5 %	97.5 %
(Intercept)	-2964.27509	-840767.86136
distance	98.34856	79.17351

```
predn.df <- data.frame(distance = 1:3)
bb.logit.pred <- predict(bb.fit2, newdata = predn.df)
bb.logit.pred
```

1: 3.16693434553016 **2:** 0.53590094125671 **3:** -2.09513246301674

Skip to main content

```
1:      0.959570820970203 2:      0.630858358052515 3:      0.109570820976233
```

```
predict(bb.fit2, newdata = predn.df, type = "response")
```

```
1:      0.959570820970203 2:      0.630858358052515 3:      0.109570820976233
```

```
bb.logit.predses <- predict(bb.fit2, newdata = predn.df, se.fit = TRUE)$se.fit
bb.logit.predses
# Lower and upper bounds of CIs for the log-odds
lower = bb.logit.pred - 1.96 * bb.logit.predses
upper = bb.logit.pred + 1.96 * bb.logit.predses
ci = cbind(lower, upper)
plogis(ci)
```

```
1:      0.815101808705842 2:      0.381297733450052 3:      0.643231168633725
```

A matrix: 3 × 2 of type dbl

	lower	upper
1	0.82768876	0.9915452
2	0.44733541	0.7830016
3	0.03370361	0.3027157

```
predictGLM(bb.fit2, newdata = data.frame(distance = 1:3), type = "link")
predictGLM(bb.fit2, newdata = data.frame(distance = 1:3), type = "response")
```

Skip to main content

```
***Estimates and CIs are on the link scale***
```

A matrix: 3 × 3 of type dbl

	fit	lwr	upr
1	3.1669343	1.5693642	4.7645045
2	0.5359009	-0.2114289	1.2832308
3	-2.0951325	-3.3558424	-0.8344225

```
***Estimates and CIs are on the response scale***
```

A matrix: 3 × 3 of type dbl

	fit	lwr	upr
1	0.9595708	0.82769294	0.9915450
2	0.6308584	0.44733881	0.7829992
3	0.1095708	0.03370437	0.3027108

15.4. Modelling the response when it is binomial (grouped binary data) via glm

```r
# Load dplyr package to manipulate data frames
library(dplyr)
bb.grouped.df = bb.df %>%
    group_by(gender, distance) %>%
    summarize(n = n(), propn = sum(basket) / n)
# Change tibble back to a data frame
bb.grouped.df = data.frame(bb.grouped.df)
bb.grouped.df
```

```
Error in library(dplyr): there is no package called ‘dplyr’
Traceback:

1. library(dplyr)
```

这样转换后，虽然结果相同，但我们可以做卡方检验了（手动经历了分组）。

Skip to main content

```
bb.fit3 = glm(propn ~ distance * gender,
    weights = n,
    family = binomial, data = bb.grouped.df
)
summary(bb.fit3)
```

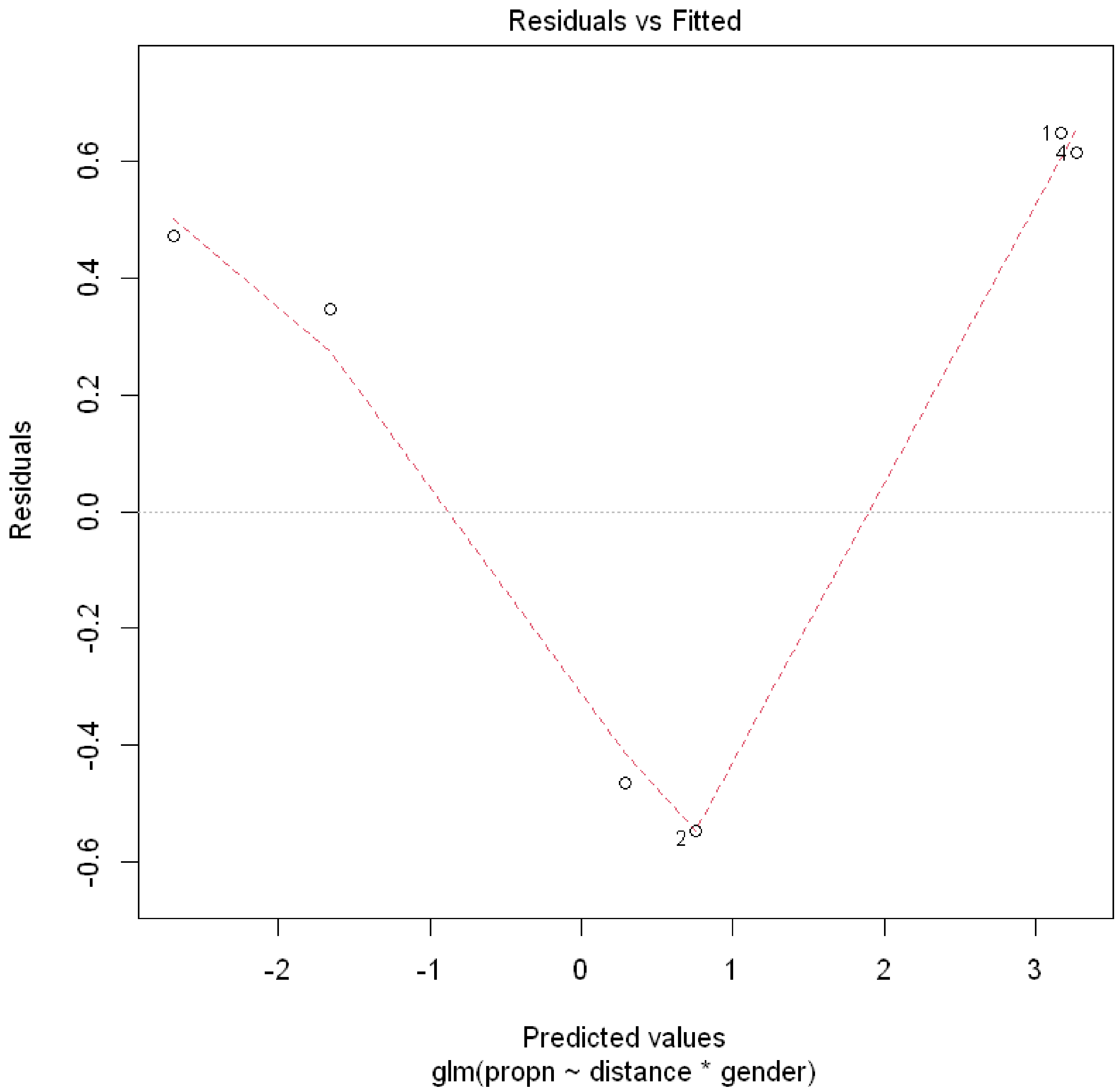
```
Call:
glm(formula = propn ~ distance * gender, family = binomial, data = bb.grouped.df,
    weights = n)

Deviance Residuals:
      1        2        3        4        5        6
 0.9063  -0.5354   0.3367   0.8612  -0.4629   0.4376

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)        5.5878     1.9050   2.933  0.00335 **
distance          -2.4159     0.8181  -2.953  0.00314 **
genderM            0.6710     2.9236   0.230  0.81848
distance:genderM  -0.5668     1.3214  -0.429  0.66795
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 38.2749  on 5  degrees of freedom
Residual deviance:  2.3688  on 2  degrees of freedom
AIC: 20.23

Number of Fisher Scoring iterations: 4
```

```
1 - pchisq(2.3688, 2)
```

0.305929682251732

```
plot(bb.fit3, which = 1, lty = 2)
```

Skip to main content



```
bb.grouped.df <- transform(bb.grouped.df, success = n * propn, fail = n * (1 - propn))
bb.grouped.df
```

Skip to main content

A data.frame: 6 × 6

gender	distance	n	propn	success	fail
<chr>	<int>	<int>	<dbl>	<dbl>	<dbl>
F	1	10	1.0	10	0
F	2	10	0.6	6	4
F	3	10	0.2	2	8
M	1	10	1.0	10	0
M	2	10	0.5	5	5
M	3	10	0.1	1	9

```
bb.fit4 = glm(cbind(success, fail) ~ distance * gender,
    family = binomial,
    data = bb.grouped.df
)
summary(bb.fit4)
```

15.5. Example 1: Space shuttle Challenger accident

The NASA space shuttle Challenger broke up during launch on the cold morning of 28 January 1986. Most of the crew survived the initial break-up, but are believed to have been killed when the crew capsule hit the ocean at high speed. 1986年1月28日的寒冷早晨，美国宇航局的挑战者号航天飞机在发射过程中解体。大多数机组人员在最初的解体过程中幸存下来，但据信在机组人员舱高速撞向海洋时被杀死。

```
Space.df <- read.table("../data/ChallengerShuttle.txt", head = TRUE)
Space.df$Temp
Space.df$Failure
```

66 · 70 · 69 · 68 · 67 · 72 · 73 · 70 · 57 · 63 · 70 · 78 · 67 · 53 · 67 · 75 · 70 · 81 · 76 · 79 · 75 · 76 · 58

0 · 1 · 0 · 0 · 0 · 0 · 0 · 0 · 1 · 1 · 1 · 0 · 0 · 2 · 0 · 0 · 0 · 0 · 0 · 0 · 2 · 0 · 1

```
Space.gfit = glm(cbind(Failure, 6 - Failure) ~ Temp,
    family = binomial,
    data = Space.df
)
```

Skip to main content

```
Call:
glm(formula = cbind(Failure, 6 - Failure) ~ Temp, family = binomial,
    data = Space.df)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.95227  -0.78299  -0.54117  -0.04379   2.65152

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.08498    3.05247   1.666   0.0957 .
Temp        -0.11560    0.04702  -2.458   0.0140 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24.230  on 22  degrees of freedom
Residual deviance: 18.086  on 21  degrees of freedom
AIC: 35.647

Number of Fisher Scoring iterations: 5
```

```r
predictGLM(Space.gfit, newdata = data.frame(Temp = 31), type = "response")
```

```r
6 * predictGLM(Space.gfit, newdata = data.frame(Temp = 31), type = "response")
```

16. Analysis of contingency tables

本节需要的包：

```r
require(s20x)
```

▶ Show code cell output

16.1. Introduction

分类统计表

```r
AP.df <- read.table("../data/AttendPass.txt", header = T)
```

Skip to main content

A data.frame: 8 × 3

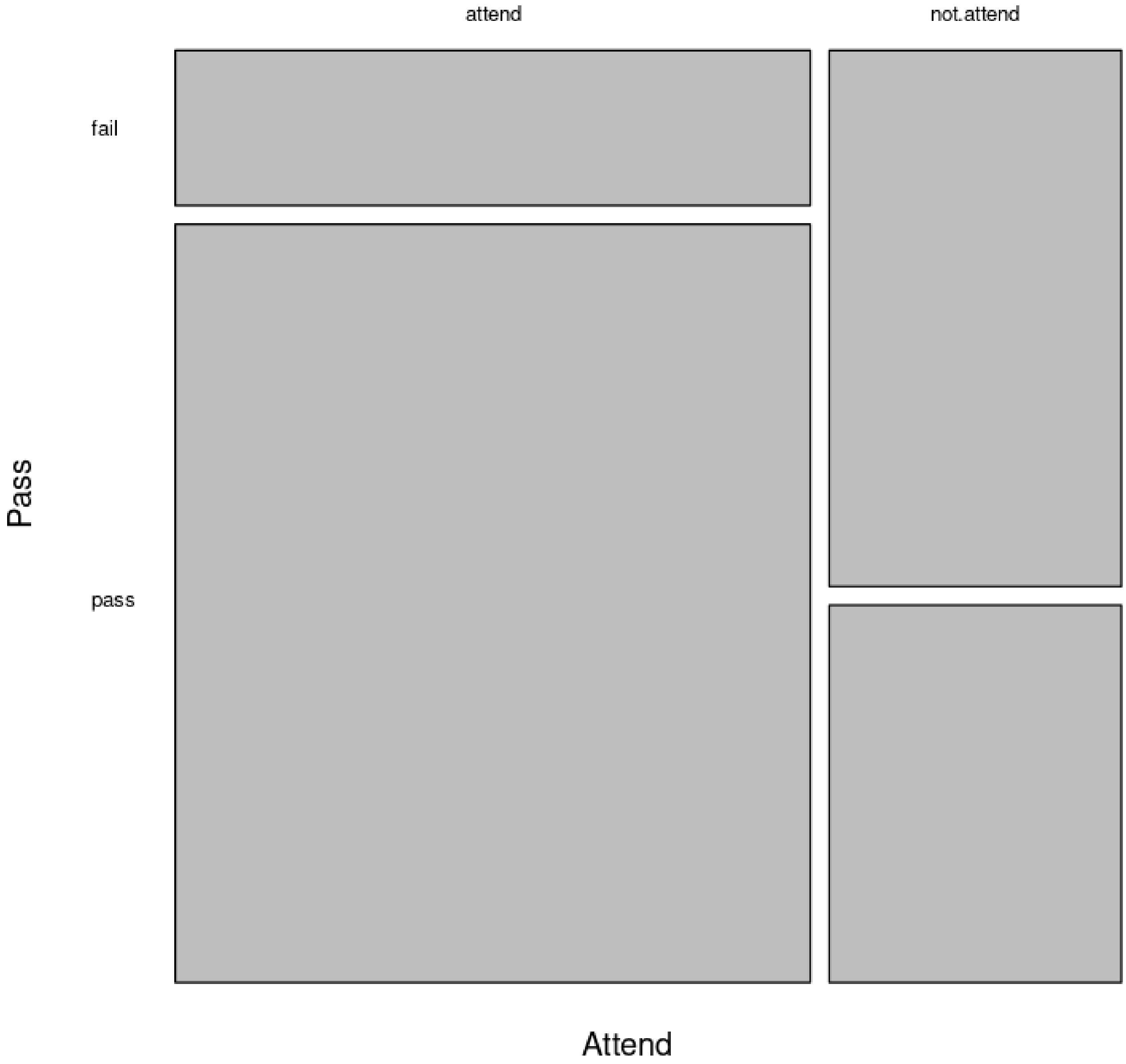
	Subject	Pass	Attend
	<int>	<chr>	<chr>
1	1	pass	attend
2	2	pass	attend
3	3	pass	attend
4	4	pass	attend
5	5	pass	attend
6	6	fail	not.attend
7	7	pass	not.attend
8	8	fail	attend

```
AP.tbl <- with(AP.df, table(Attend, Pass))
AP.tbl
```

```
             Pass
Attend        fail pass
  attend        17   83
  not.attend    27   19
```

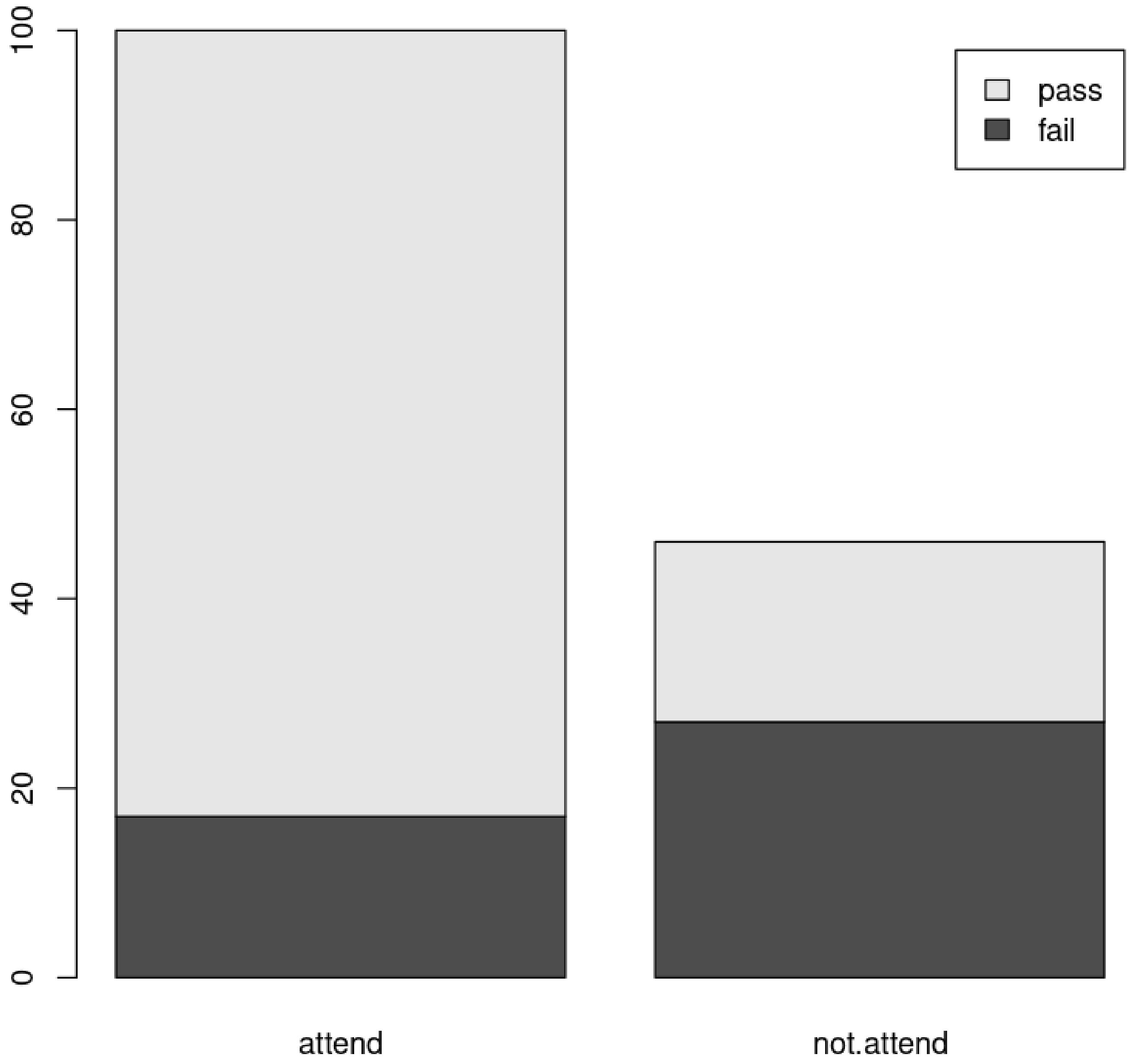
```
plot(AP.tbl, main = "", las = 1)
```

Skip to main content



```r
barplot(t(AP.tbl), legend = T)
```

Skip to main content



16.2. The binomial approach to contingency table analysis

```r
Freqs.df <- data.frame(
    Attend = c("not.attend", "attend"),
    Fail = c(27, 17), Pass = c(19, 83)
)
Freqs.df <- transform(Freqs.df, Attend = factor(Attend))
Freqs.df
```

A data.frame: 2 × 3

Attend	Fail	Pass
<fct>	<dbl>	<dbl>
not.attend	27	19
attend	17	83

```r
AP.binom <- glm(cbind(Pass, Fail) ~ Attend, data = Freqs.df, family = binomial)
summary(AP.binom)
```

```
Call:
glm(formula = cbind(Pass, Fail) ~ Attend, family = binomial,
    data = Freqs.df)

Deviance Residuals:
[1]  0  0

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)        1.5856     0.2662   5.956 2.58e-09 ***
Attendnot.attend  -1.9370     0.4007  -4.834 1.34e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance:  2.5162e+01  on 1  degrees of freedom
Residual deviance: -3.5527e-15  on 0  degrees of freedom
AIC: 12.756

Number of Fisher Scoring iterations: 3
```

```r
exp(confint(AP.binom))[2, ]
```

Skip to main content

```
Waiting for profiling to be done...
```

2.5 %: 0.0642983964738621 **97.5 %:** 0.311134072466483

16.3. The Poisson approach to contingency table analysis

```
library(dplyr)
AP.df <- read.table("../data/AttendPass.txt", header = T)
AP.df <- transform(AP.df, Pass = factor(Pass), Attend = factor(Attend))
Freqs2.df <- AP.df %>%
    group_by(Attend, Pass) %>%
    summarize(freq = n()) %>%
    data.frame()
Freqs2.df
```

```
Error in library(dplyr): there is no package called ‘dplyr’
Traceback:

1. library(dplyr)
```

```
Freqs2.df$Attend <- relevel(Freqs2.df$Attend, ref = "not.attend")
AP.pois <- glm(freq ~ Attend * Pass, family = poisson, data = Freqs2.df)
summary(AP.pois)
```

A matrix: 4 × 4 of type dbl

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.2958369	0.1924501	17.125671	9.550242e-66
Attendattend	-0.4626235	0.3096136	-1.494197	1.351243e-01
Passpass	-0.3513979	0.2994472	-1.173489	2.405999e-01
Attendattend:Passpass	1.9370252	0.4006749	4.834407	1.335434e-06

Attendattend:Passpass: 6.93808049535602

Null deviance 叫做零模型自由度，Residual deviance 叫做残差自由度。

残差自由度等于零时，参数个数等于该数据的行数。由此可以推出原数据有 4 行。

Skip to main content

```r
exp(confint(AP.pois))[4, ]
```

```
Waiting for profiling to be done...
```

2.5 %: 3.21404850350281 **97.5 %:** 15.552487384471

16.4. Equivalence of the binomial and Poisson approaches

```r
Freqs.df
exp(confint(AP.pois))[4, ]
```

A data.frame: 2 × 3

Attend	Fail	Pass
<fct>	<dbl>	<dbl>
not.attend	27	19
attend	17	83

```
Waiting for profiling to be done...
```

2.5 %: 3.21404850350281 **97.5 %:** 15.552487384471

```r
predict(AP.pois, type = "response")
```

1: 17.0000000000001 **2:** 83.0000000000001 **3:** 27 **4:** 19

```r
Freqs.df
coef(summary(AP.pois))
exp(coef(AP.pois))[4]
```

Skip to main content

A data.frame: 2 × 3

Attend	Fail	Pass
<fct>	<dbl>	<dbl>
not.attend	27	19
attend	17	83

A matrix: 4 × 4 of type dbl

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.2958369	0.1924501	17.125671	9.550242e-66
Attendattend	-0.4626235	0.3096136	-1.494197	1.351243e-01
Passpass	-0.3513979	0.2994472	-1.173489	2.405999e-01
Attendattend:Passpass	1.9370252	0.4006749	4.834407	1.335434e-06

Attendattend:Passpass: 6.93808049535602

```r
options(digits = 4)
OR <- 27 * 83 / (17 * 19)
OR
```

At the last

至此，所有内容你都学完了。恭喜！让我们复习一下这些内容。当然接下来的内容也是考试重点考察内容。

考题构造

总构造：5+12 题。其中：

- 选择题 3.5*12
- 综合题 58

> 注：multi-choice 是从多个选一个

Skip to main content

相关模型

lm model (1-12 chapter)

前提假设：正态分布

- 简单线性（Simple Linear Regression）

  Formula: $y = \beta_0 + \beta_1 x_i + \varepsilon_i,\ \varepsilon_i \sim N(0, \sigma^2)$ where $\beta_1 > 0$.

- 零模型（The Null Model）

  Formula: $y = \beta_0 + x$

- 多项式（Curved Model (Like Quadratic Model)）

  Formula: $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon_i$

- 多变量
- 多变量交互
- 指数
- 幂

glm model (13-16 chapter)

前提假设：泊松分布、二项分布

泊松一般是一到二道题

模型研究

1. 对数据绘制图形，评价图形中数据变化的趋势

   什么样的趋势，自变量和因变量的关系

2. 选择合适的模型进行拟合（包括 lm 和 glm）

   要求会写不同模型的代码、拟合函数。注意不要忘了分布的书写

3. 残差三分析：同分布、正态检验、异常值

Skip to main content

- 公式
- t 检验

  t-value 含义：预估变量值 - 假设H_0（假设能接受的该变量值） / 标准差

  由 t-value 和自由度一起分析得到 p-value，从而得出我们能否接受该假设。（对于三十几行以内的数据而言，t 绝对值大于 1.96 就基本等价于 p 小于 0.05，大于95%置信区间）当然零假设本身就落在95%置信区间也可以。

  但是我们的summary里面是设定的零假设为0。

- p 检验

  p-value：大于等于 0.05 则可以接受零假设，小于 0.05 不能接受零假设

- r 方

  误差：模型解释的偏差+观测的误差

  r 方即模型能解释误差的百分比，模型能解释的误差越少，就说明数据越不受控，不能直接拒绝。

- 参数的95%置信区间

  含义：哪一个解释更有用，解释是什么？

- 自由度

  原数据有多少行：残差自由度 + 参数变量个数（零模型自由度加一也行，但一般不用）

5. 预测 predict

   代码里参数有两个，prediction预测个体，confidence预测均值（个体预测范围更大）

   注意有时还要套用exp（看y有没有变形）

6. method and assumption checks：问题是什么，数据什么情况，拟合什么具体模型（解释单变量、多变量在模型里的含义），r方是多少
7. executive summary：解释拟合参数的含义（解释参数都需要变形，每单位参数的变化，导致预测值变化多少、变成原来多少倍），回答题目问题

其他知识点

Skip to main content

什么是 1 - 卡方？

代码中一般是 `1 - pchisq(36.59, 18)` 这样的。

The p-value for testing the null hypothesis that the data are 泊松/二项 distributed, the p-value is below 0.05 (above 0.05), rejecting the null hypothesis.

the null hypothesis is that the binomial variance assumption holds, the residual deviance is chi-squared with 18 degrees of freedom if this assumption is true.

in this case, the p-value is below 0.05, rejecting the null hypothesis.

log 使用条件

- 数值数据（曲线拟合log，使用后变成直线）
- 分类数据（箱线图观察，log后宽度变成一致了就是好的）

公式书写

- 正态分布：$E(Exam_i) = \beta_0 + \beta_1 Test_i + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$
- 泊松分布：$log(Exam_i) = \beta_0 + \beta_1 Test_i + \epsilon$ where ...

License

Note

This website is built using Jupyter Book, a Jupyter static website generator.

Skip to main content