Introduction and Discussion

- 本文对于随机前沿算法的描述与复现参考自：

  Aigner, D.J., Lovell, C.K. and Schmidt, P., 1977. Formulation and estimation of stochastic. *Review of Economics and Statistics*, *80*(3), pp.454-465.

1. 问题背景

生产函数理论定义了在给定一组输入和固定技术条件下能够获得的最大产出量。而对于实际的统计实践中，想要对所谓"生成前沿"——即最大产出量——进行估计时，区别于传统的最小二乘法或极大使然估计。具体地，我们有

$$y_i = f(x_i;\beta) + \varepsilon_i, i = 1, \dots, n, \tag{1}$$

其中y_i代表从x_i（一组非随机的输入）能得到的最大产出，β是待估计的未知参数向量，ε_i是扰动项。

由于我们想要得到的估计$\hat{y_i} = f(x_i, \hat{\beta})$是生产"前沿"——最大边界，则其必然大于我们已经观测到的值。这导致了估计过程中的重要约束：

$$\hat{y_i} - y_i = f(x_i, \hat{\beta}) - y_i = -\hat{\varepsilon_i} \geq 0$$

在传统的正态回归假设下，$\varepsilon_i \sim n(0, \sigma_i^2)$，有$E(\varepsilon_i) = 0$，显然不满足假设。因此，无法使用通常的定理来判断参数估计的渐近分布。

2. 随机前沿

回到方程（1）中给出的模型，但采用以下误差结构

$$\varepsilon_i = v_i + u_i, i = 1, \dots, N. \tag{2}$$

误差成分v_i代表对称扰动：假设$\{v_i\}$是独立同分布的，服从$n(0, \sigma_v^2)$。误差成分u_i假设与v_i独立分布，并满足$u_i \leq 0$。特别假设u_i来源于在零以上截断的$n(0, \sigma_u^2)$分布的情况。然而，其他单侧分布也是可行的。

为简化估计的讨论，考虑一个线性模型。因此，以矩阵形式重写（2）

$$y = X\beta + s, \tag{3}$$

其中$s = v + u$。

对称正态随机变量和截断正态随机变量之和的累积分布函数的显式表达式是由M.A. Weinstein（1964）首次推导出来。ε的密度函数的推导在这里不与展开。结果是

$$f(\varepsilon) = \frac{2}{\sigma} f^* \left(\frac{\varepsilon}{\sigma}\right)[1 - F^*(\varepsilon\lambda\sigma^{-1})], -\infty \leq \varepsilon \leq +\infty \tag{4}$$

其中$\sigma^2=\sigma_v^2+\sigma_u^2,\lambda=\frac{\sigma_u}{\sigma_v}$，$f^*(\cdot)$和$F^*(\cdot)$分别是标准正态的概率密度和累积分布函数。这个概率密度函数在0为中心是不对称的，其均值和方差由下式给出：

$$\begin{aligned} E(\varepsilon)=E(u)&=-\frac{\sqrt{2}\sigma_v}{\sqrt{\Pi}},\\ V(\varepsilon)=V(v)+V(u)&=(\frac{\Pi-2}{\Pi})\sigma_u^2+\sigma_v^2, \end{aligned} \tag{5}$$

估计问题是通过假设有n个观测值的随机样本，然后形成对数似然函数，

$$lnL(y|\beta,\lambda,\sigma^2)=n\times ln\left(\frac{\sqrt{2}}{\sqrt{\Pi}}\right)+n\times ln\sigma^{-1}+\sum_{i=1}^{n}ln[1-F^*(\varepsilon_i\lambda\sigma^{-1})]-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\varepsilon_i^2, \tag{6}$$

求偏导后，我们有

$$\frac{\partial \ln L}{\partial \sigma^2}=-\frac{n}{2\sigma^2}+\frac{1}{2\sigma^4}\sum_{i=1}^{n}(y_i-\beta^Tx_i)^2+\frac{\lambda}{2\sigma^3}\sum_{i=1}^{Nn}\frac{f_i^*}{(1-F_i^*)}(y_i-\beta^Tx_i), \tag{7}$$

$$\frac{\partial \ln L}{\partial \lambda}=-\frac{1}{\sigma}\sum_{i=1}^{n}\frac{f_i^*}{(1-F_i^*)}(y_i-\beta^Tx_i), \tag{8}$$

$$\frac{\partial \ln L}{\partial \beta}=\frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i-\beta^Tx_i)x_i+\frac{\lambda}{\sigma}\sum_{i=1}^{n}\frac{f_i^*}{(1-F_i^*)}x_i, \tag{9}$$

其中 x_i 是 X 中第 i 行的元素组成的 $k\times 1$ 向量，f_i^* 和 F_i^* 分别是在 $(y_i-\beta^Tx_i)\lambda\sigma^{-1}$ 处评估的标准正态分布的概率密度和累积分布函数。

给定(8)，我们知道在最优情况下有

$$\sum_{i=1}^{n}\frac{f_i^*}{(1-F_i^*)}(y_i-\beta^Tx_i)=0,$$

通过将这个结果插入(7)，最大似然估计量 β^T 通过以下方式确定

$$-\frac{n}{2\sigma^2}+\frac{1}{2\sigma^4}\sum_{i=1}^{n}(y_i-\beta^Tx_i)=0, \tag{10}$$

这导出了

$$\hat{\sigma}^2=\frac{1}{n}\sum_{i=1}^{n}(y_i-\beta^Tx_i)^2, \tag{11}$$

这是回归模型中残差方差的常用最大似然估计器的基础。但是，β 的确定并不独立于其他方程中的 β^T。

3. 随机前沿算法实现

根据作者描述，由于除了对应于 X 中一列全为1的系数外，其他维度的 β_i 通过最小二乘法估计是无偏且一致的，从而我们据此简单设计如下梯度下降算法：

> Step 1. 初始值设定
>
> 1. 计算无约束最小二乘估计得到初始β
> 2. 通过$\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^T x_i)^2$，计算初始$\hat{\sigma^2}$
> 3. 利用$\frac{\partial \ln L}{\partial \lambda} = 0$的公式计算$\lambda$
> 4. 设置合适的步长
>
> Step 2. 利用梯度更新$\hat{\beta}$
>
> 1. 利用初始值只计算梯度$\frac{\partial \ln L}{\partial \beta}$
> 2. 利用梯度更新$\hat{\beta}$
> 3. 重新利用$\frac{\partial \ln L}{\partial \lambda} = 0$和$\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta^T x_i)^2$公式计算$\lambda$和$\hat{\sigma^2}$并更新
>
> Step 3. 检查收敛或最大迭代次数

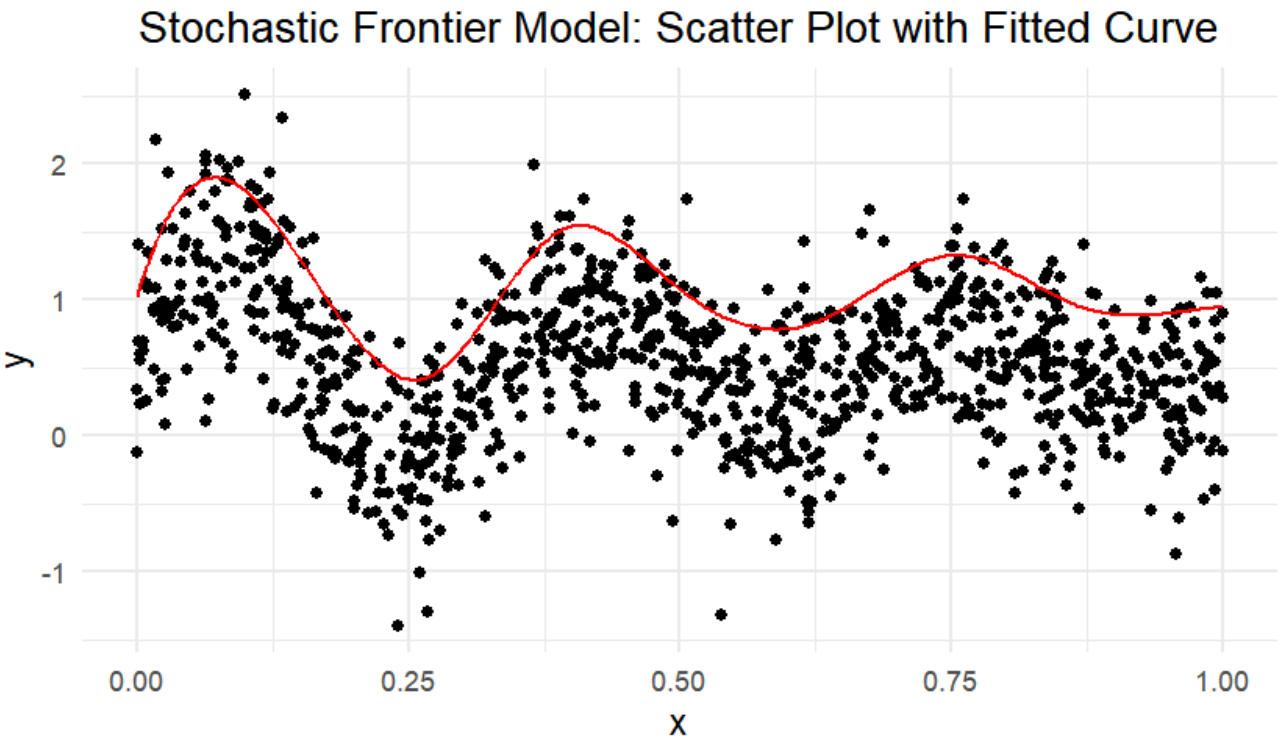
以下是上述算法的R语言实现：

```r
if (!requireNamespace("MASS", quietly = TRUE)) install.packages("MASS")
library(MASS)
library(splines)
library(ggplot2)

# 加载数据
data <- read.csv("data.txt", header = FALSE, sep = " ")
names(data) <- c("x", "y")

x <- data$x
y <- data$y

# 定义B样条基函数
p <- 10  # 基函数的数量
Phi <- bs(x, df = p)
X <- cbind(1, Phi)  # 添加截距项

# Step 1: 初始值设定
beta_hat <- lm(y ~ X - 1)$coefficients  # 无截距的线性模型
sigma_hat_squared <- sum((y - X %*% beta_hat)^2) / length(y)

# 使用pnorm和dnorm获取F*和f*的函数
get_lambda <- function(y, X, beta, sigma_squared) {
```

```r
    epsilon ← (y - X %*% beta) / sqrt(sigma_squared)
    lambda ← sum(dnorm(epsilon) / (1 - pnorm(epsilon))) / sum((y - X %*% beta)^2)
    return(lambda)
}

lambda_hat ← get_lambda(y, X, beta_hat, sigma_hat_squared)

# 设置步长，这可能需要根据实际情况调整
alpha ← 0.0002

# Step 2: 利用梯度更新beta_hat中的截距项
convergence ← FALSE
max_iter ← 50000  # 设置最大迭代次数
iter ← 0

while(!convergence && iter < max_iter) {
    iter ← iter + 1

    # 计算梯度
    epsilon ← (y - X %*% beta_hat) / sqrt(sigma_hat_squared)
    gradient ← t(X) %*% (y - X %*% beta_hat) / sigma_hat_squared +
        lambda_hat / sqrt(sigma_hat_squared) * t(X) %*% (dnorm(epsilon) / (1 - pnorm(epsilon)))

    # 只更新beta_0（截距项）
    beta_hat[1] ← beta_hat[1] + alpha * gradient[1]

    # 重新计算sigma_hat_squared和lambda_hat
    sigma_hat_squared ← sum((y - X %*% beta_hat)^2) / length(y)
    lambda_hat ← get_lambda(y, X, beta_hat, sigma_hat_squared)

    # 检查收敛条件
    if(abs(gradient[1]) < 1e-6) {
        convergence ← TRUE
        cat("Convergence reached at iteration", iter, "\n")
    }
}

# Step 3: 检查收敛或最大迭代次数
if (convergence) {
    cat("Algorithm converged in", iter, "iterations.\n")
} else {
    cat("Algorithm did not converge in the maximum number of iterations.\n")
}
```

```r

# 输出结果
results ← list(beta_hat = beta_hat, sigma_hat_squared =
sigma_hat_squared, lambda_hat = lambda_hat)

# 计算拟合值，注意我们在这里使用完整的X（包括截距项和基函数）
fitted_values ← X %*% beta_hat

# 准备绘图数据
plot_data ← data.frame(x = x, y = y, fitted = fitted_values)

# 绘制散点图和拟合曲线
ggplot(plot_data, aes(x = x, y = y)) +
    geom_point() +
    geom_line(aes(y = fitted), color = 'red') +
    theme_minimal() +
    ggtitle("Stochastic Frontier Model: Scatter Plot with Fitted Curve") +
    theme(plot.title = element_text(hjust = 0.5))  # 居中标题
```

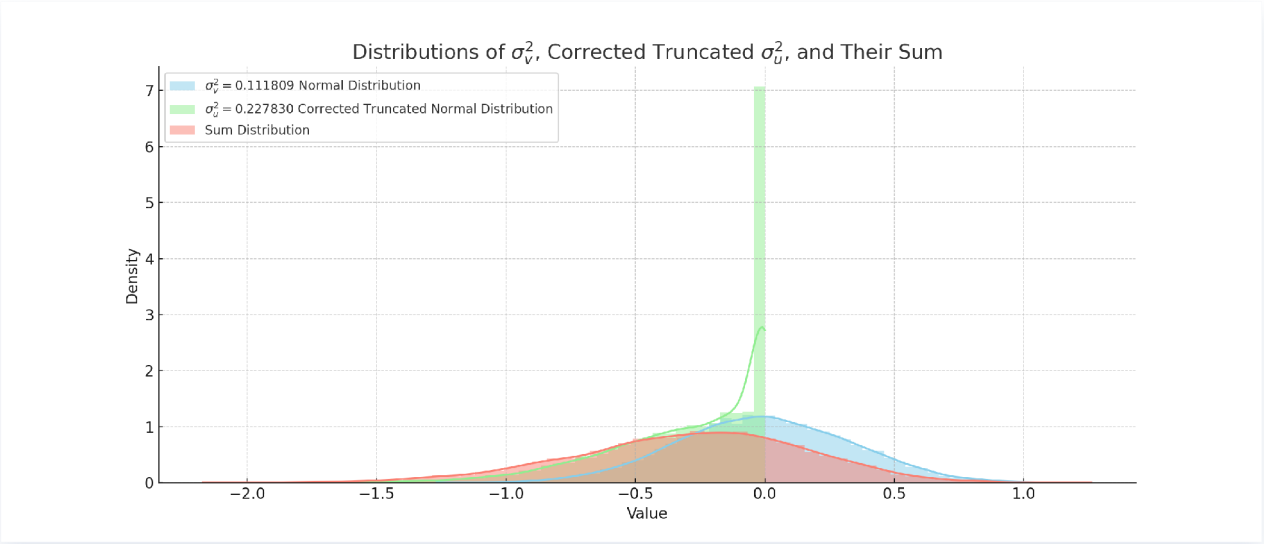
对于示例数据data.txt，随机前沿模型的到如下结果：



可以看到我们的估计——所谓"前沿"——也就是图片中红色数据点形成的曲线，基本上靠近数据点的上边界。解释了这种估计下的“前沿”确实是“随机”的。另外在这种方法下，我们还得到了扰动项部分的方差的估计

$$\sigma_v^2 = 0.111809$$
$$\sigma_u^2 = 0.227830$$

据此，我们通过采样近似出出扰动项ε_i的分布图:



可以看到扰动项ε_i的分布（红色区域）相较于0均值的分布（蓝色区域）向左偏移，使其大部分的概率密度在小于0的区域，与散点及估计曲线图中，大部分数据位于红色拟合曲线下方（对应残差——估计得到的"前沿"减去观测值——大于0）对应。

4. 随机前沿模型的问题

对于这种误差先验的设定，作者如是说：

> 这种框架背后的经济逻辑是，生产过程受到两种经济上可区分的随机扰动，它们有不同的特征。我们相信，**尽管我们的解释显然是新的**，但文献中有充分的先例支持这种观点。[1] 并且从**实用的角度**来看，这种区分极大地促进了前沿的估计和解释。非正扰动u_i反映了每个公司的产出必须位于其前沿$f(x_i;\beta)+v_i$上或以下。任何这样的偏差都是公司可控因素的结果，如**技术和经济效率低下、生产者及其员工的意愿和努力，以及可能的其他因素因素，如缺陷和损坏的产品**。但是，前沿本身可以在公司之间或对于同一公司随时间随机变化。在这种解释下，前沿是随机的，随机扰动$v_i \geq or \leq 0$是由于有利或不利的外部事件造成的，例如**运气、气候、地形和机器性能**。对y的**测量误差**构成了$v_i \geq or \leq 0$的另一个来源。

显然，随机前沿的模型的误差设定动机是便于极大似然估计，而非源于经济理论（尽管作者尽力解释，但仍然显得牵强）。为"边界"设定"随机"显然也不符合我们得直觉。另外，注意到我们的两个误差来源的方差是估计的！这意味着不论结果如何，我们必须接受这种随机的设置，即使他看起来有些违背现实（尽管我们可以在估计前直接设定，但是指定先验更难不是吗）。

5. 约束最小二乘估计

回到我们的问题设置

$$y_i - \hat{y}_i = y_i - f(x_i, \hat{\beta}) = \hat{\varepsilon}_i \geq 0 \tag{12}$$

同样将$f(x_i, \hat{\beta})$设置为线性，我们将其重写为一个优化问题：

最小化目标函数：

$$min = \sum_{i=1}^{n} \|y_i - X_i^T \beta\|_2^2, \text{，对于 } i = 1, \ldots, n \tag{13}$$

受到约束：

$$\text{subject to, } X_i^T \beta - y_i \geq 0 \text{，对于 } i = 1, \ldots, n \tag{14}$$

类似的带有约束最小二乘估计，可以使用更现代的内点法（1984，Karmarkar）计算。

具体地，其中我们定义 $\left(\hat{\beta}_0(r), \ldots, \hat{\beta}_p(r)\right)$ 以最小化

$$\sum_{i=1}^{n} \left(y_i - X_i^T \beta\right)^2 - r \sum_{i=1}^{n} \ln\left(X_i^T \beta - y_i\right) \tag{15}$$

我们为其推导一个迭代加权最小二乘算法：

$$\hat{\beta}_k = (X^T W(\hat{\beta}_{k-1}) X)^{-1} X^T W(\hat{\beta})_{k-1} Z(\hat{\beta}_{k-1}) \tag{16}$$

其中，$W(\beta)$是对角矩阵，其对角元素为：

$$w_{ii} = -1 - \frac{r}{\left(y_i - X_i^T \beta\right)^2} \tag{17}$$

$Z(\hat{\beta}_{k-1})$为：

$$Z(\hat{\beta}_{k-1}) = X\hat{\beta}_{k-1} + X(X^T W(\hat{\beta}_{k-1}) X)^{-1} S(\hat{\beta}_{k-1}) \tag{18}$$

$S(\hat{\beta}_{k-1})$为：

$$S(\hat{\beta}_{k-1}) = \nabla_\beta L(\beta) = -\sum_{i=1}^{n} \left(y_i - X_i^T \beta\right) X_i - r \sum_{i=1}^{n} \frac{X_i}{(X_i^T \beta - y_i)} \tag{19}$$

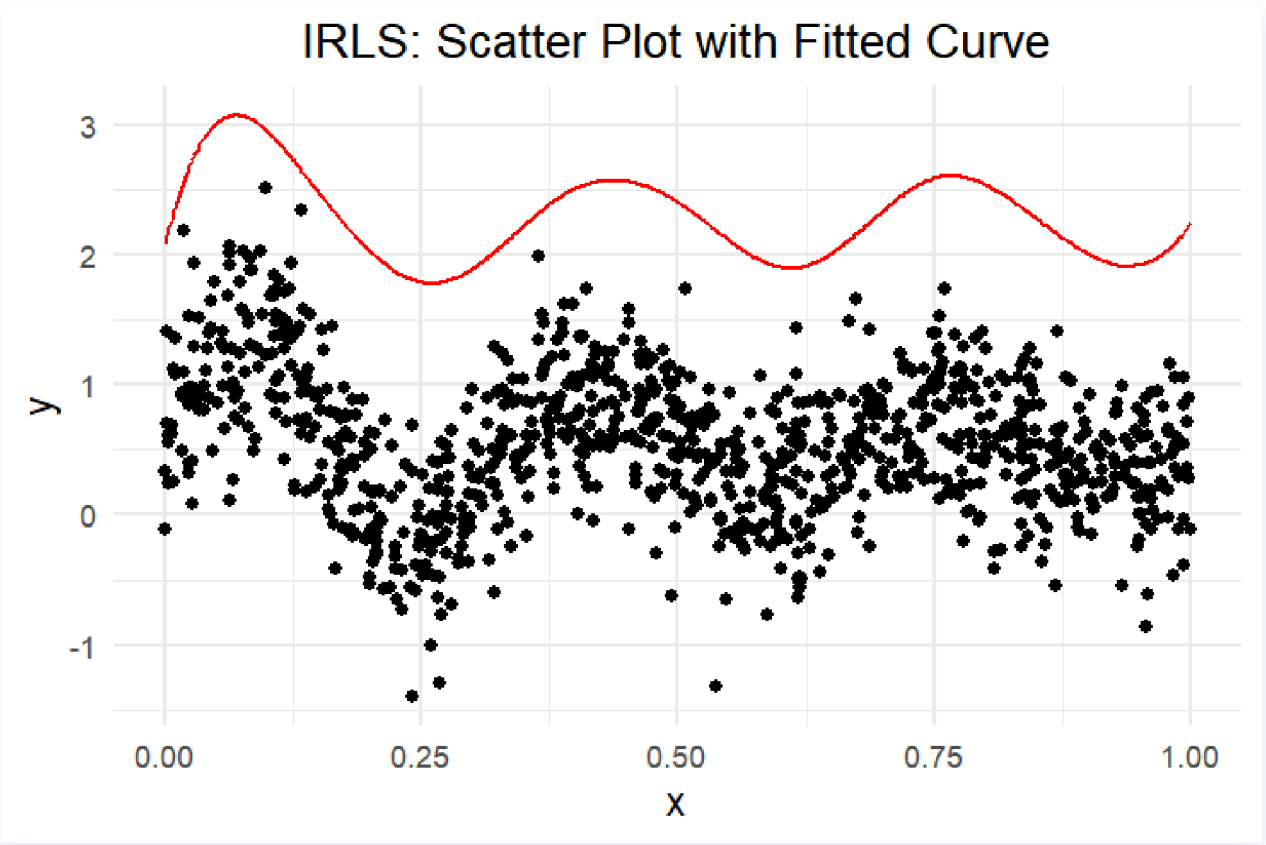
以下是上述算法的R语言实现：

```r
library(splines)
library(ggplot2)

# 加载数据
data <- read.csv("data.txt", header = FALSE, sep = " ")
names(data) <- c("x", "y")
```

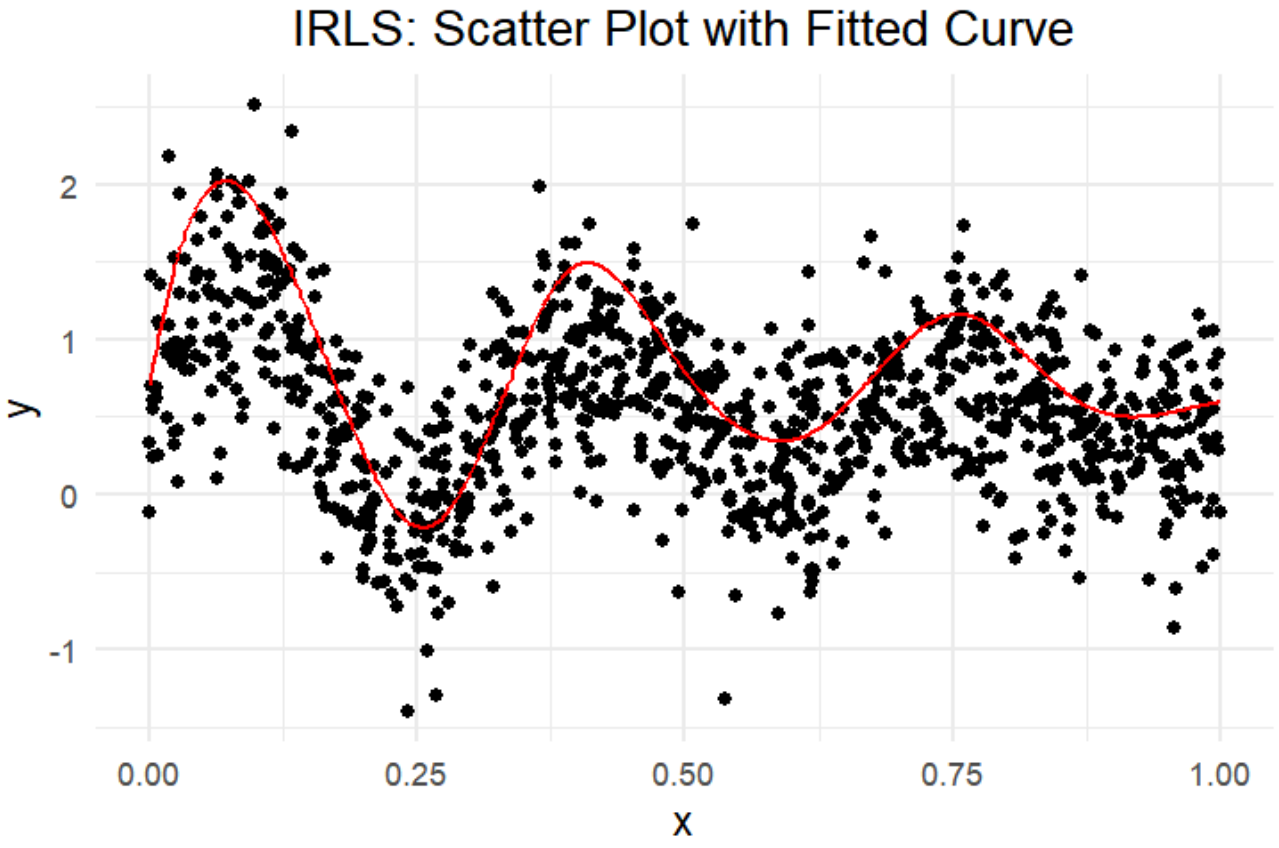
```r

x ← data$x
y ← data$y

# 定义B样条基函数
p ← 10 # 基函数的数量
Phi ← bs(x, df = p)
Phi ← cbind(1, Phi)  # 添加截距项

# 无约束最小二乘法估计作为初始值
beta_hat ← solve(t(Phi) %*% Phi) %*% t(Phi) %*% y
beta_hat ← beta_hat - min(beta_hat)

# 检查约束是否满足
constraints_satisfied ← all(Phi %*% beta_hat -y ⩾ 0)
cat("First Constraints satisfied:", constraints_satisfied, "\n")

# 设置收敛阈值和迭代最大次数
threshold ← 1e-6
max_iter ← 50

# 计算初始r值
u ← y - Phi %*% beta_hat
v ← 1 / (y - Phi %*% beta_hat)
r ← sum(u * (Phi %*% t(Phi)) %*% v) / sum(v * (Phi %*% t(Phi)) %*% v)
r ← ifelse(r > 0, r, 1)
r ← r/2.5

cat("The r_0:", r)


for (iter in 1:max_iter) {
    Phi_beta ← Phi %*% beta_hat
    inv_Phi_beta ← 1 / Phi_beta  # 逐元素求倒数
    residuals ← y - Phi_beta

    # 计算S(beta)
    S_beta ← -t(Phi) %*% residuals - r * colSums(Phi / matrix(Phi_beta - y, nrow = nrow(Phi), ncol = ncol(Phi), byrow = TRUE))

    # 计算W的对角线元素
    W_diag ← -1 - r / (residuals^2)
    W ← diag(as.vector(W_diag))  # 创建对角矩阵W

    # 计算Z(beta)
    Z ← Phi_beta + Phi %*% solve(t(Phi) %*% W %*% Phi) %*% S_beta

```

```r

    # 更新beta
    beta_new ← solve(t(Phi) %*% W %*% Phi) %*% t(Phi) %*% W %*% Z

    # 检查收敛性
    if (max(abs(beta_new - beta_hat)) < threshold) {
        cat("Convergence achieved after", iter, "iterations.\n")
        break
    }

    # 更新r值
    r ←  r / 1 # 可选的衰减

    # 更新beta_hat
    beta_hat ← beta_new

    # 报告当前迭代信息
    cat("Iteration:", iter, "- r value:", r, "\n")

    # 检查约束是否满足
    constraints_satisfied ← all(Phi %*% beta_hat - y ⩾ 0)
    cat("Constraints satisfied:", constraints_satisfied, "\n")
}

# 输出最终估计
print(beta_hat)

# 绘制结果
fitted_values ← Phi %*% beta_hat

plot_data ← data.frame(x = x, y = y, fitted = fitted_values)

ggplot(plot_data, aes(x = x, y = y)) +
    geom_point() +
    geom_line(aes(y = fitted), color = 'red') +
    theme_minimal() +
    ggtitle("IRLS: Scatter Plot with Fitted Curve") +
    theme(plot.title = element_text(hjust = 0.5))  # 居中标题
```

下图是迭代最小二乘法得到的估计“前沿”的图像。注意我们并没有为约束项的系数r指定衰减（`r←r/1`），从而约束项的障碍函数会非常严格的考虑约束，我们得到了一个真正意义上（而不是概率上）的前沿。



但是如果我们我们为$\{r_i\}$指定一个趋向于0的衰减，意味着随着迭代的进行，约束项的约束力度越弱，算法有可能突破约束，得到类似随机前沿的估计：



1. Marschak和Andrews（1944）建议$v_i + u_i$的总和反映了生产者的“技术效率”和“意愿、努力和运气”。Zellner, Kmenta, 和 Dreze (1966)建议它反映了“诸如天气、机器或劳动性能的不可预测变化等因素”，他们可能是第一批提出随机生产函数的人，尽管他们显然没有考虑到前沿。误差项的其他表述存在：Aigner和Chu（1968）通过技术和经济效率低下以及生产过程中可能由于粗心处理和缺陷或损坏产出导致的纯随机冲击来解释它。Timmer (1971)引用了技术和经济效率低下，以及“变量中的定义和测量问题”。而且，农业经济学家经常引用诸如气候、地形和土壤类型等环境条件在农场之间的变化，作为随机生产函数的指示。↩