8.

（a）

i. 预测变量(horsepower)与响应变量(mpg)的关系：

- horsepower系数的p值： 7.031989e-81 （远小于0.05，统计显著）

- 整体模型F检验p值： NA （远小于0.05，模型整体显著）

结论：horsepower与mpg存在显著的线性关系。

ii. 关系的强弱（由R²衡量）：

- Multiple R-squared： 0.6059483 （表示mpg约 60.59 %的变异可由horsepower解释）

结论：线性关系较强（R²接近0.84）。

iii. 相关性方向：

- horsepower的系数估计： -0.1578447 （符号为负）

结论：horsepower与mpg呈负相关（马力越大，油耗越低）

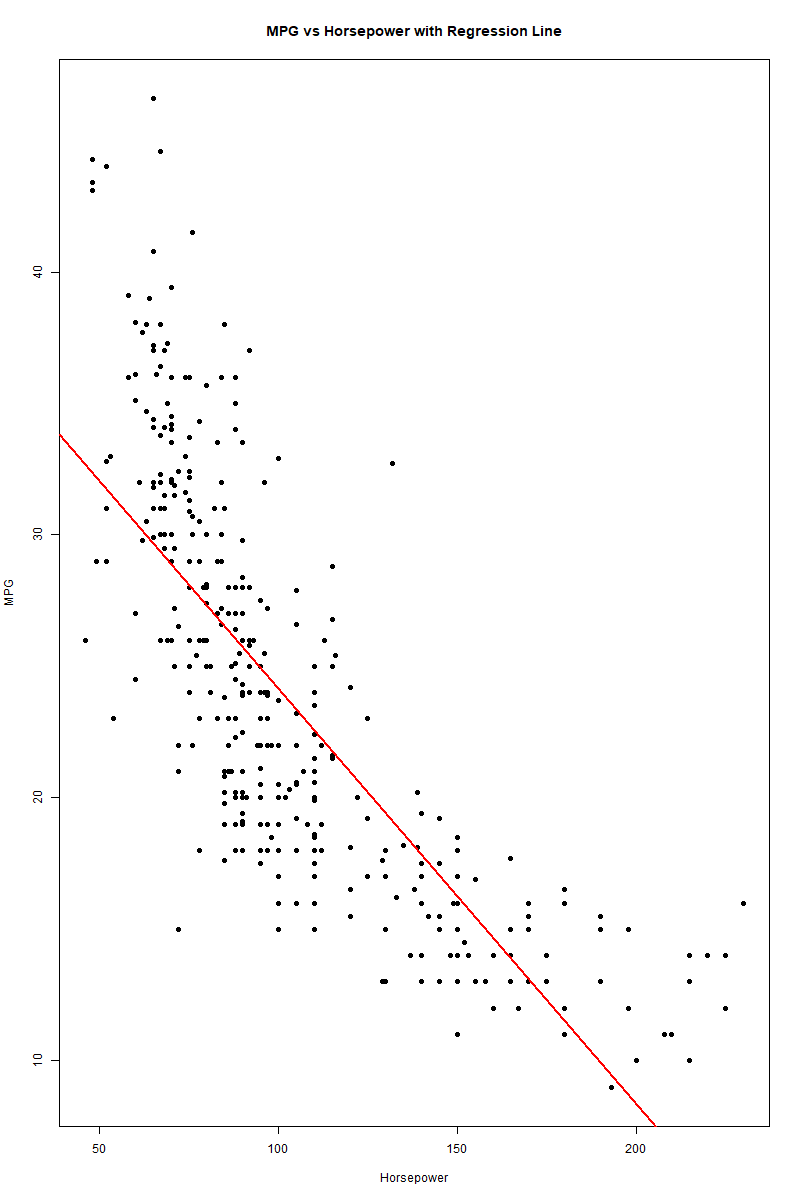
iv. 当horsepower=98时的预测结果：

- 预测值(fit)： 24.46708

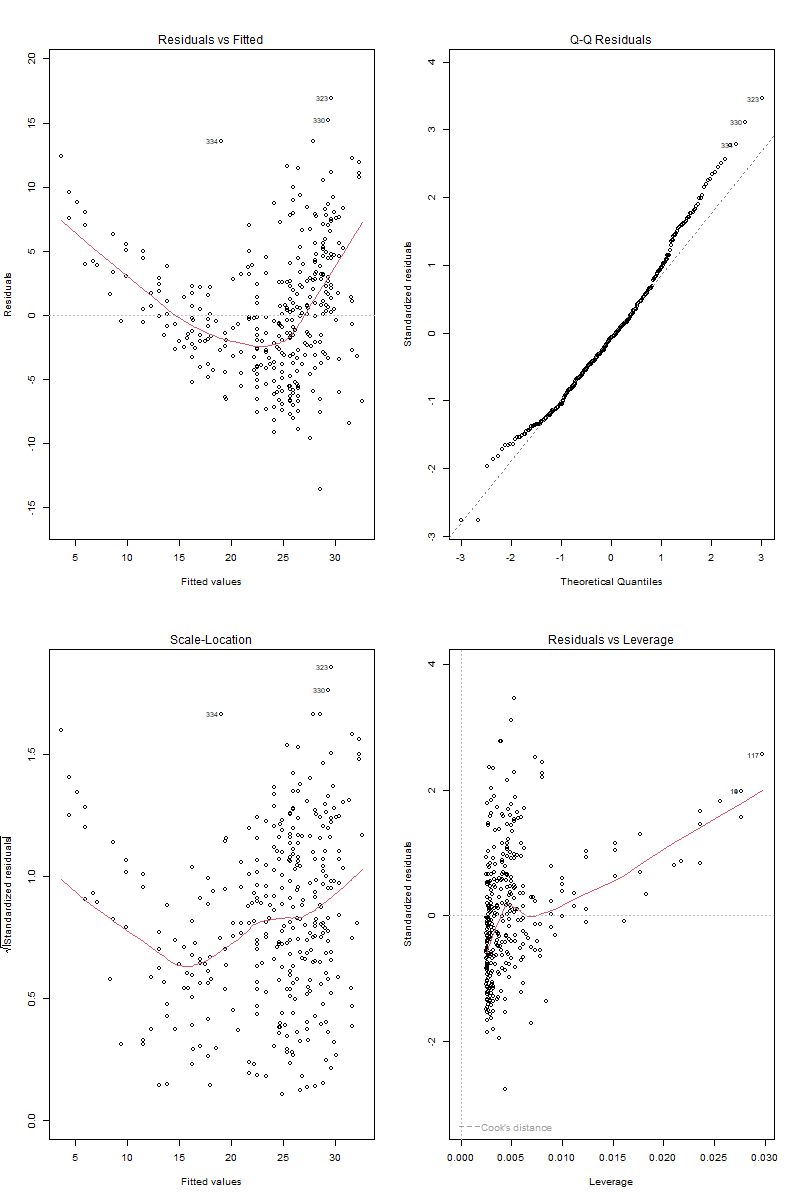
- 95%置信区间（均值mpg的区间）：[ 23.97308 , 24.96108 ]

- 95%预测区间（单个mpg观测的区间）：[ 14.8094 , 34.12476 ]

（b）



（c）



1. Residuals vs Fitted（残差vs拟合值）：

- 残差存在轻微曲线趋势，提示简单线性拟合可能不足，mpg与horsepower可能存在非线性关系（如二次关系）。

2. Normal Q-Q（正态QQ图）：

- 部分点偏离对角线，说明残差分布存在厚尾现象，不完全服从正态分布。

3. Scale-Location（方差稳定性）：

- 残差的离散程度随拟合值变化有轻微趋势，提示可能存在轻度异方差（方差非齐性）。

4. Residuals vs Leverage（杠杆值与残差）：

- 未发现明显的高杠杆点或强影响异常值，但需关注个别远离中心的点（其残差和杠杆值较大时可能影响模型）。

11.

（a）输出结果：

Call:

lm(formula = y ~ x + 0)

Residuals:

Min 1Q Median 3Q Max

-1.9154 -0.6472 -0.1771 0.5056 2.3109

Coefficients:

Estimate Std. Error t value Pr(>|t|)

x 1.9939 0.1065 18.73 <2e-16 \*\*\*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9586 on 99 degrees of freedom

Multiple R-squared: 0.7798, Adjusted R-squared: 0.7776

1. statistic: 350.7 on 1 and 99 DF, p-value: < 2.2e-16

**系数估计**

斜率估计值: 1.9939，非常接近真实值2

标准误差: 0.1065，相对较小

t值: 18.73，高度显著

p值: <2e-16，极其显著

**模型拟合度**

Multiple R-squared: 0.7798

表示约78%的y变异可由x解释

Adjusted R-squared: 0.7776

考虑了模型复杂度的调整R²

**残差分**析

残差范围: -1.9154 到 2.3109

中位数接近0，分布相对对称

（b）输出结果：

Call:

lm(formula = x ~ y + 0)

Residuals:

Min 1Q Median 3Q Max

-0.8699 -0.2368 0.1030 0.2858 0.8938

Coefficients:

Estimate Std. Error t value Pr(>|t|)

y 0.39111 0.02089 18.73 <2e-16 \*\*\*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4246 on 99 degrees of freedom

Multiple R-squared: 0.7798, Adjusted R-squared: 0.7776

1. statistic: 350.7 on 1 and 99 DF, p-value: < 2.2e-16

1. 斜率关系

原模型斜率: 1.9939 ≈ 2 (真实值)

反向模型斜率: 0.39111 ≈ 1/2 = 0.5

两者并不互为倒数，这是回归不对称性的体现

2. R²相同

两个模型的 Multiple R-squared 完全相同(0.7798)，这是因为:

R² = Cor(x,y)²

相关系数在两个方向上是相同的

3. 统计显著性

两个模型的t值相同(18.73)

p值相同(<2e-16)

F统计量相同

（c）这种现象反映了线性回归的不对称性:

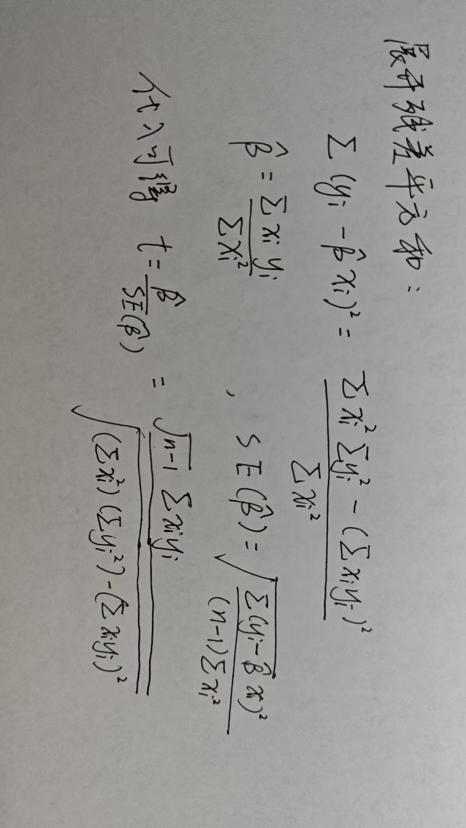
lm(y ~ x + 0) 最小化的是y方向的误差

lm(x ~ y + 0) 最小化的是x方向的误差

因此得到的斜率不同，但解释力(R²)相同

这说明在简单线性回归中，因变量和自变量的角色是不可互换的，除非数据完美线性相关(R²=1)。

（d）



代码验证：> # 正确的手动t统计量计算

> beta\_hat <- sum(x \* y) / sum(x^2)

> residuals <- y - beta\_hat \* x

> sigma\_squared <- sum(residuals^2) / (length(x) - 1)

> se\_beta <- sqrt(sigma\_squared / sum(x^2))

> t\_manual <- beta\_hat / se\_beta

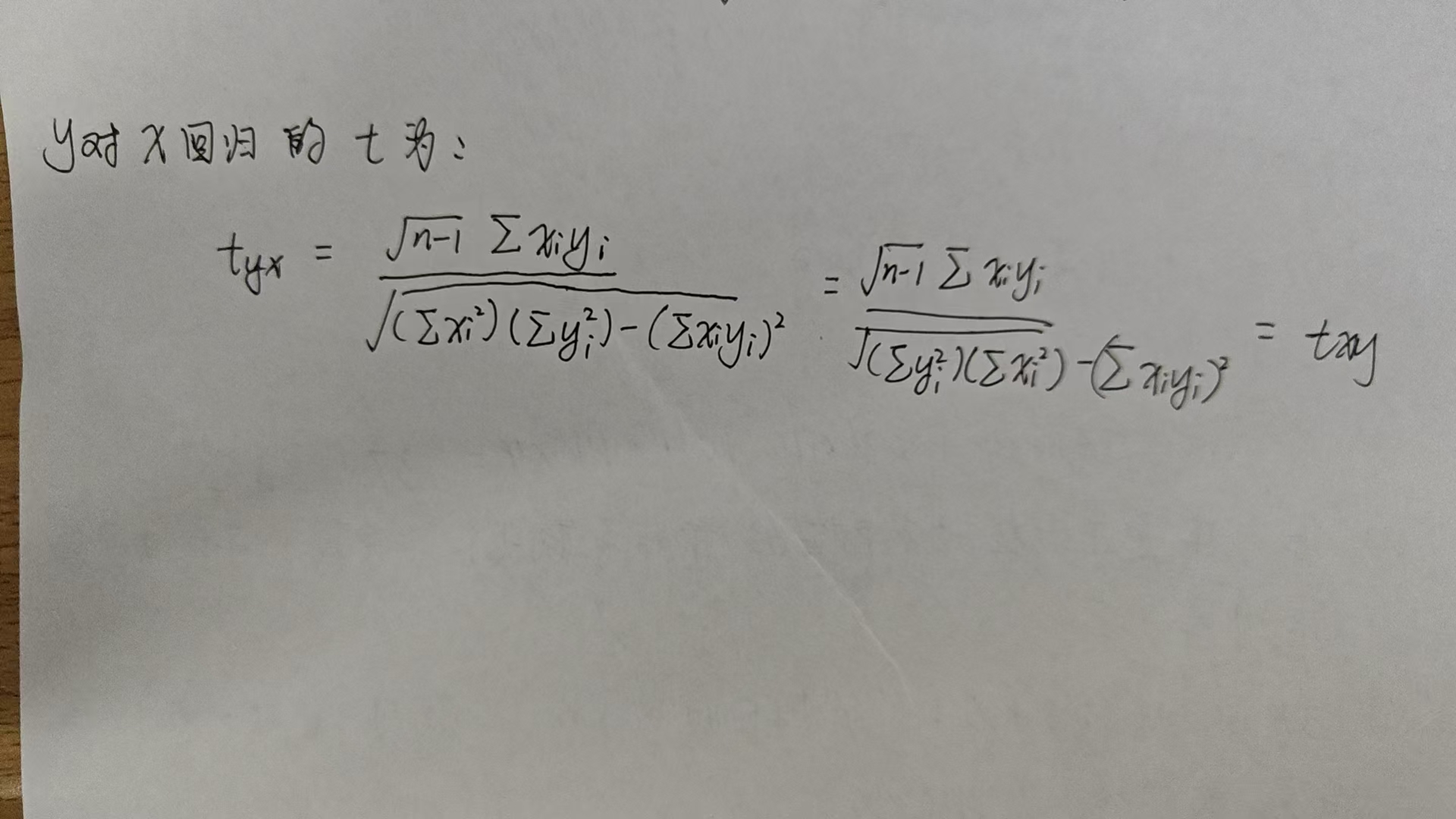
>

> # 使用all.equal进行比较

> all.equal(t\_manual, summary(fit)$coefficients[1, 3])

1. TRUE

（e）



（f）

> fit\_yx\_int <- lm(y ~ x)

> fit\_xy\_int <- lm(x ~ y)

> t\_yx\_int <- coef(summary(fit\_yx\_int))[2, 3]

> t\_xy\_int <- coef(summary(fit\_xy\_int))[2, 3]

> t\_yx\_int == t\_xy\_int # 结果为TRUE

1. TRUE

13.

# (a) 生成x：100个N(0,1)观测

x <- rnorm(100)

# (b) 生成eps：100个N(0, 0.25)观测（方差0.25 → 标准差0.5）

eps <- rnorm(100, mean = 0, sd = sqrt(0.25))

# (c) 生成y：根据模型 Y = -1 + 0.5X + ε

y <- -1 + 0.5 \* x + eps

# 向量y的长度与模型参数

length(y) # 输出100

cat("真实模型参数：β₀ =", -1, "，β₁ =", 0.5, "\n")

# (d) 绘制散点图

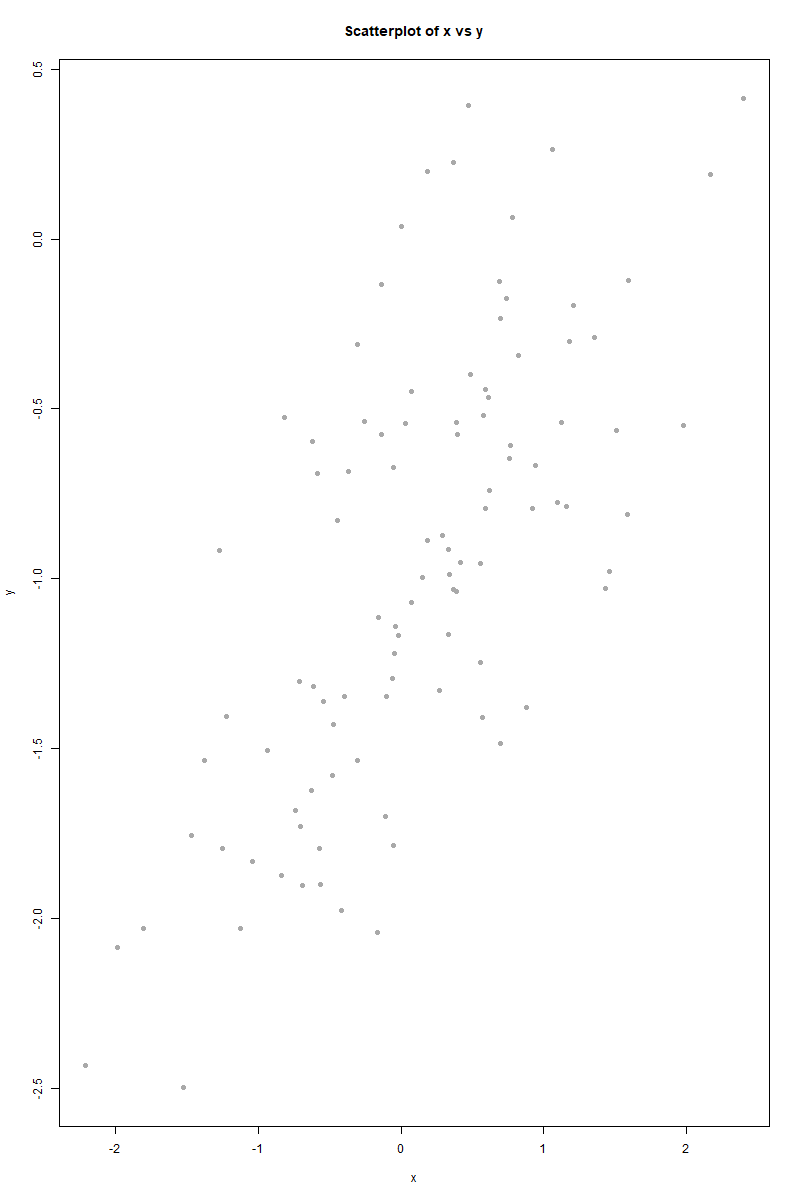
plot(x, y,

    main = "Scatterplot of x vs y",

    xlab = "x", ylab = "y",

    pch = 16, col = "darkgray"

)



散点大致呈线性趋势，但存在随机波动（由误差项eps引入）。这是因为y的核心趋势由-1 + 0.5x决定，而eps的随机性导致点围绕潜在直线分散。

# (e)回归结果：

Call:

lm(formula = y ~ x)

Residuals:

Min 1Q Median 3Q Max

-0.93842 -0.30688 -0.06975 0.26970 1.17309

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1.01885 0.04849 -21.010 < 2e-16 \*\*\*

x 0.49947 0.05386 9.273 4.58e-15 \*\*\*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4814 on 98 degrees of freedom

Multiple R-squared: 0.4674, Adjusted R-squared: 0.4619

1. statistic: 85.99 on 1 and 98 DF, p-value: 4.583e-15

截距估计值为 -1.01885 真实值为-1

斜率估计值为 0.49947 真实值为0.5

# (f) 叠加回归线

plot(x, y,

    main = "Scatterplot with Regression Lines",

    xlab = "x", ylab = "y"

)

# 最小二乘拟合线（红色）

abline(fit, col = "red", lwd = 2)

# 真实回归线（蓝色，Y = -1 + 0.5X）

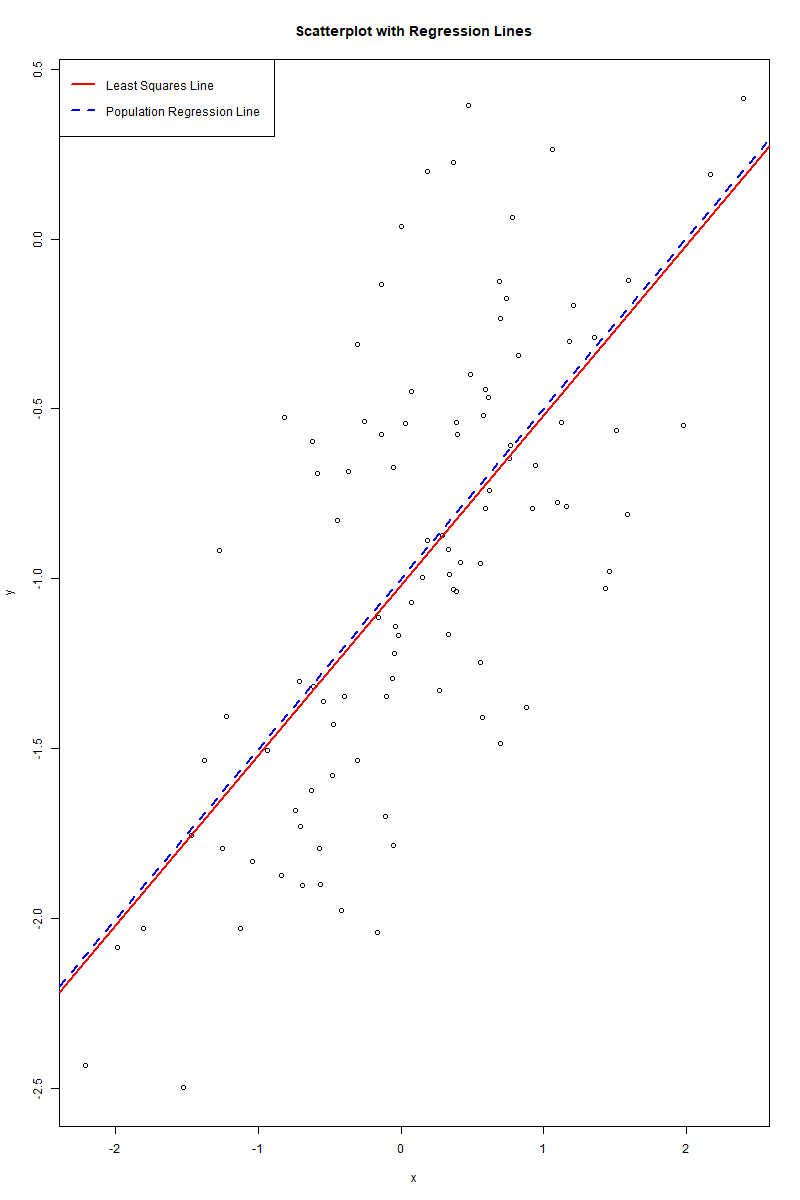
abline(-1, 0.5, col = "blue", lwd = 2, lty = 2)

legend("topleft",

    legend = c("Least Squares Line", "Population Regression Line"),

    col = c("red", "blue"), lty = c(1, 2), lwd = 2

)



# (g) 拟合模型 y ~ x + I(x^2)

fit\_poly <- lm(y ~ x + I(x^2))

summary(fit\_poly)

输出结果：

Call:

lm(formula = y ~ x + I(x^2))

Residuals:

Min 1Q Median 3Q Max

-0.98252 -0.31270 -0.06441 0.29014 1.13500

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -0.97164 0.05883 -16.517 < 2e-16 \*\*\*

x 0.50858 0.05399 9.420 2.4e-15 \*\*\*

I(x^2) -0.05946 0.04238 -1.403 0.164

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.479 on 97 degrees of freedom

Multiple R-squared: 0.4779, Adjusted R-squared: 0.4672

statistic: 44.4 on 2 and 97 DF, p-value: 2.038e-14

结论：

不能，二次项的p值为0.164>0.05不显著，同时R方的提升幅度也极小，因此并未改善拟合度

# (h)

# 生成低噪声数据

eps\_small <- rnorm(100, 0, 0.1) # 方差0.01（原方差0.25→更小）

y\_small <- -1 + 0.5 \* x + eps\_small

# 重复(d)-(f)：散点图+回归线

plot(x, y\_small,

    main = "Low Noise: Scatterplot with Regression Lines",

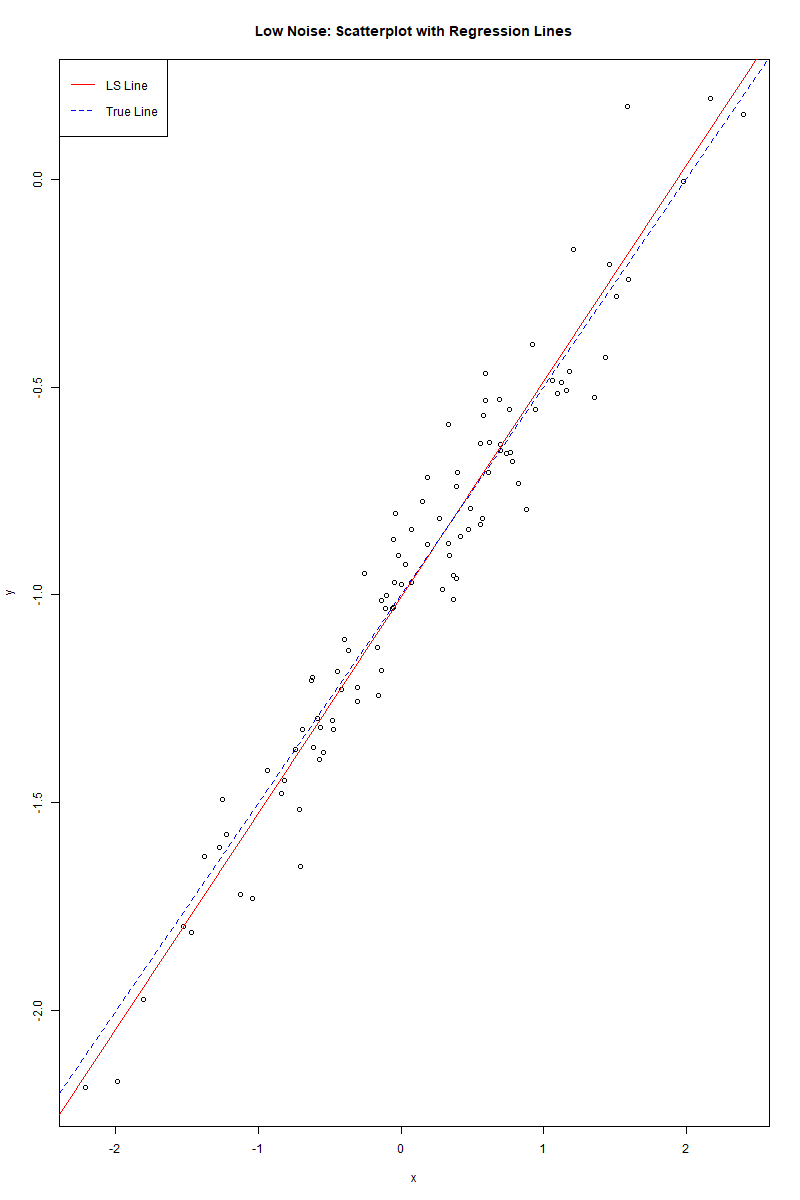
    xlab = "x", ylab = "y"

)

abline(lm(y\_small ~ x), col = "red")

abline(-1, 0.5, col = "blue", lty = 2)

legend("topleft", legend = c("LS Line", "True Line"), col = c("red", "blue"), lty = c(1, 2))



结果：散点更集中于真实回归线，最小二乘线与真实线几乎重合——噪声减小后，数据更贴近模型，估计精度显著提高。

# (i)

# 生成高噪声数据

eps\_large <- rnorm(100, 0, 1) # 方差1（原方差0.25→更大）

y\_large <- -1 + 0.5 \* x + eps\_large

# 重复(d)-(f)：散点图+回归线

plot(x, y\_large,

    main = "High Noise: Scatterplot with Regression Lines",

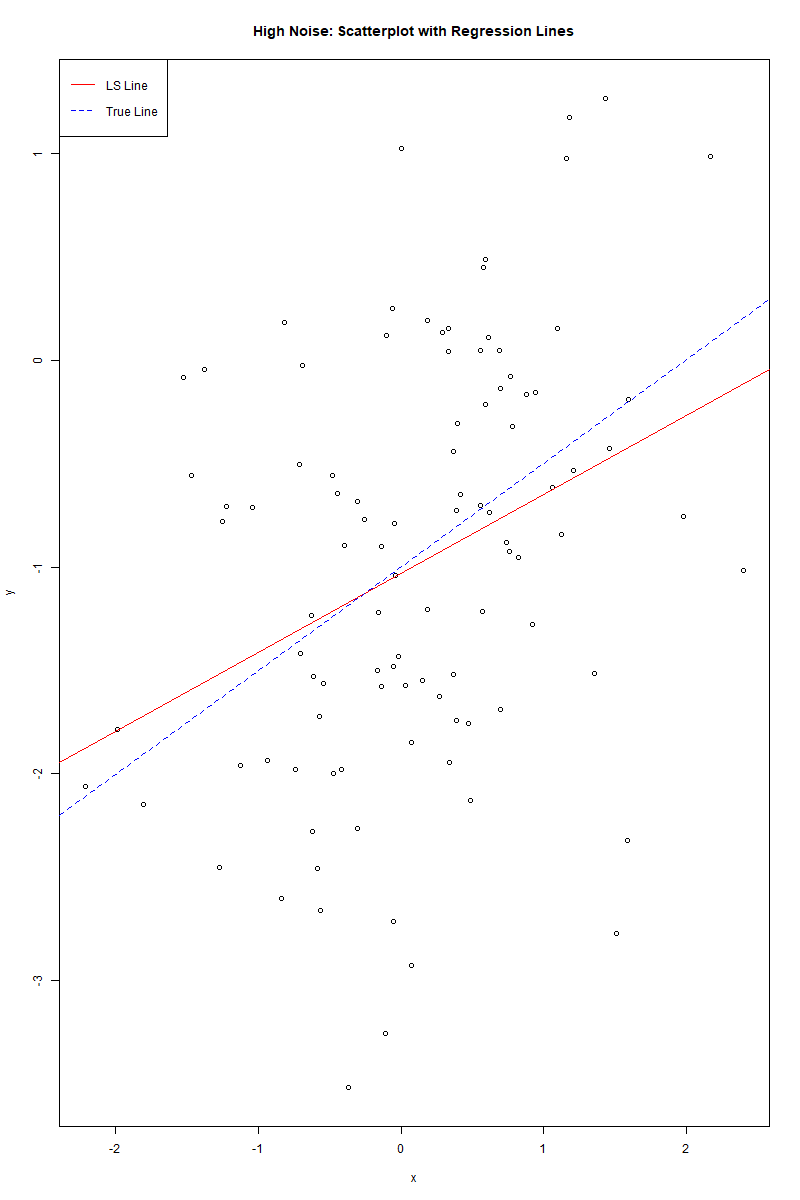
    xlab = "x", ylab = "y"

)

abline(lm(y\_large ~ x), col = "red")

abline(-1, 0.5, col = "blue", lty = 2)

legend("topleft", legend = c("LS Line", "True Line"), col = c("red", "blue"), lty = c(1, 2))



结果：散点更分散，最小二乘线与真实线的视觉偏差增大（但估计仍无偏）——噪声增大后，信号被噪声掩盖，估计波动加剧，模型解释力（如R²）降低。

# (j)

> # 打印置信区间宽度对比

> cat("原始数据置信区间宽度（β₀）：", conf\_int\_orig[1, 2] - conf\_int\_orig[1, 1$

原始数据置信区间宽度（β₀）： 0.1924681

> cat("高噪声数据置信区间宽度（β₀）：", conf\_int\_large[1, 2] - conf\_int\_large[$

高噪声数据置信区间宽度（β₀）： 0.3859185

> cat("低噪声数据置信区间宽度（β₀）：", conf\_int\_small[1, 2] - conf\_int\_small[$

低噪声数据置信区间宽度（β₀）： 0.04649145

结论：

高噪声时，置信区间最宽（噪声大→误差方差大→系数标准误大→区间宽）； 低噪声时，置信区间最窄（噪声小→误差方差小→系数标准误小→区间窄）； 原始数据的置信区间宽度介于两者之间。