EX4.1

Jiaqi Wang

2025-10-21

Q1

1-1

```
B0 <- 2
B1 <- 0.5
sigma <- 1
n <- 100

set.seed(1)

datal <- data.frame(
  X = 1:100
) %>%
  mutate(
    e = rnorm(n, mean = 0, sd = sigma),
    Y = B0 + B1 * X + e
  )

head(datal)
```

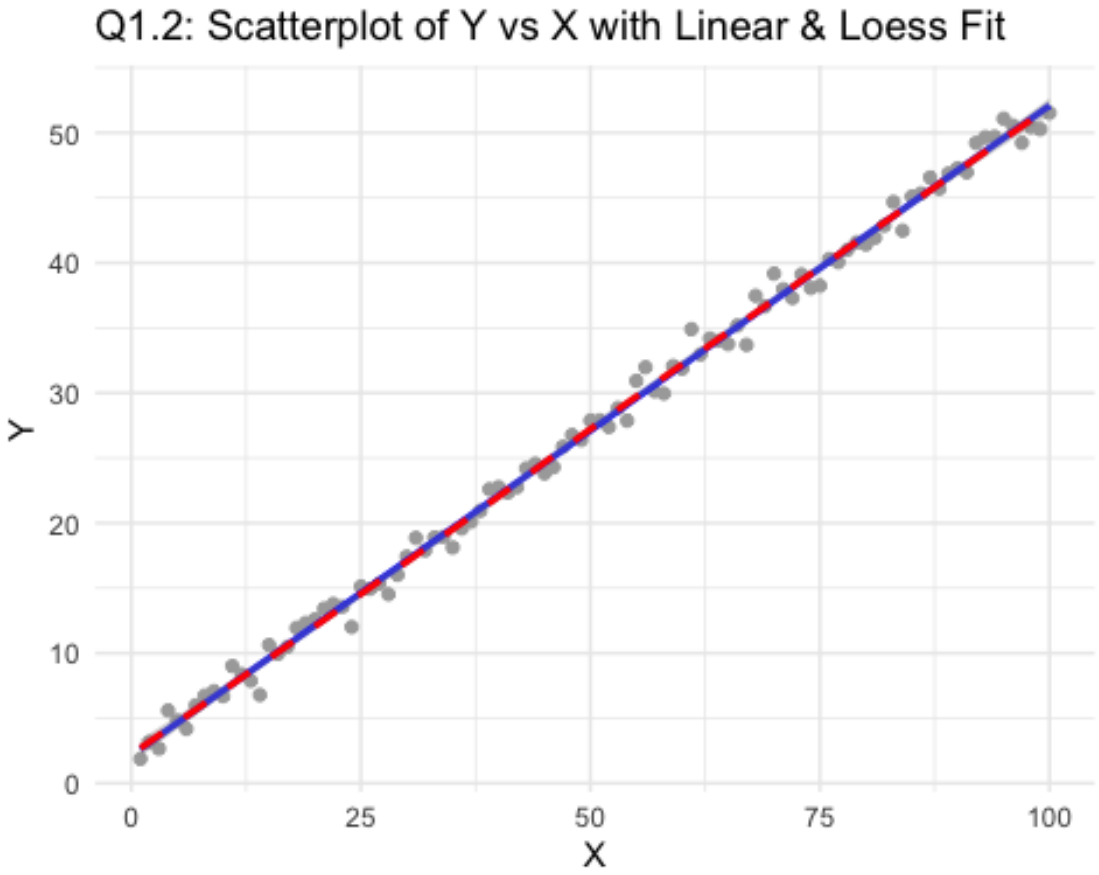
```
##   X          e        Y
## 1 1 -0.6264538 1.873546
## 2 2  0.1836433 3.183643
## 3 3 -0.8356286 2.664371
## 4 4  1.5952808 5.595281
## 5 5  0.3295078 4.829508
## 6 6 -0.8204684 4.179532
```

1-2

Scatterplot + fitted regression line:

```
library(ggplot2)

ggplot(datal, aes(x = X, y = Y)) +
  geom_point(color = "darkgray") +
  geom_smooth(method = "lm", se = TRUE, color = "blue", linewidth = 1) +
  geom_smooth(method = "loess", se = TRUE, color = "red", linetype =
"dashed") +
  labs(
    title = "Q1.2: Scatterplot of Y vs X with Linear & Loess Fit",
```

```
    x = "X",
    y = "Y"
  ) +
  theme_minimal()

## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

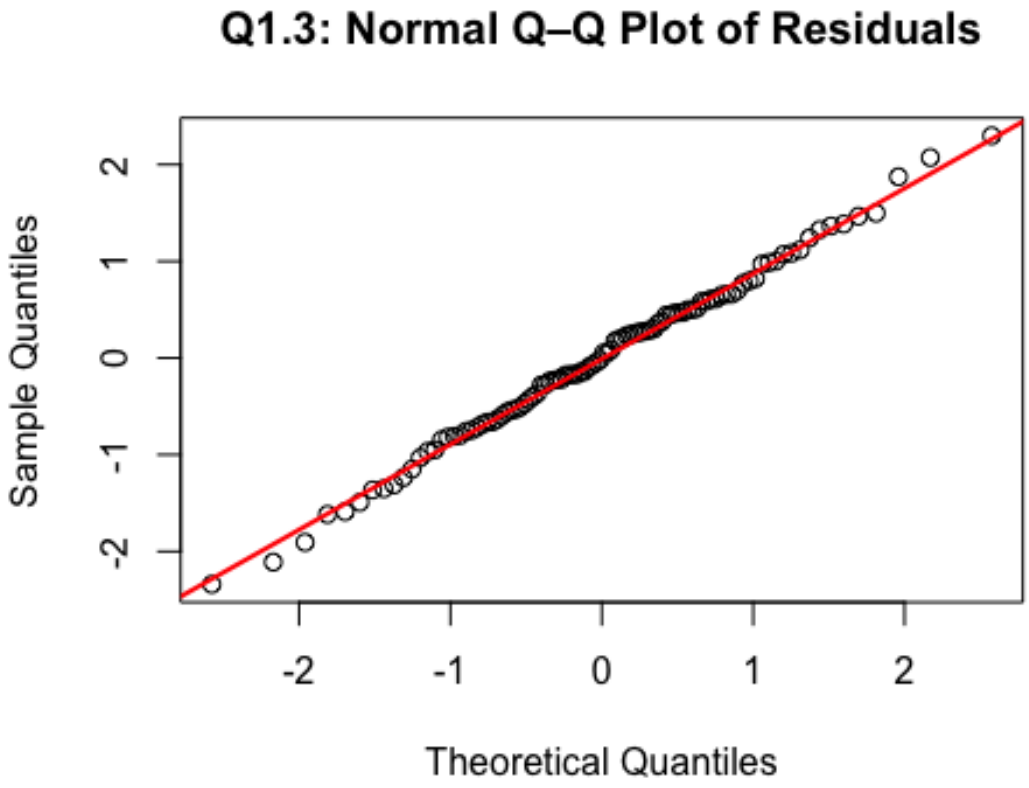


• The gray dots lie very close to a straight upward-sloping line. • The blue line (fitted regression line) passes almost perfectly through the middle of the cloud. • There's only tiny random variation around the line — no curve or pattern. The data were simulated from Y = 2 + 0.5X + ε with $\varepsilon \sim$ N(0,1). The scatterplot shows a clear positive linear trend, and the best-fitting regression line describes the data very well. The fit appears very strong, with only minor random scatter around the line.

1-3

```
# Q1.3: Q-Q plot of residuals ----------------------------
# 1. Fit the model
fit <- lm(Y ~ X, data = data1)

# 2. Extract residuals
res <- residuals(fit)
```

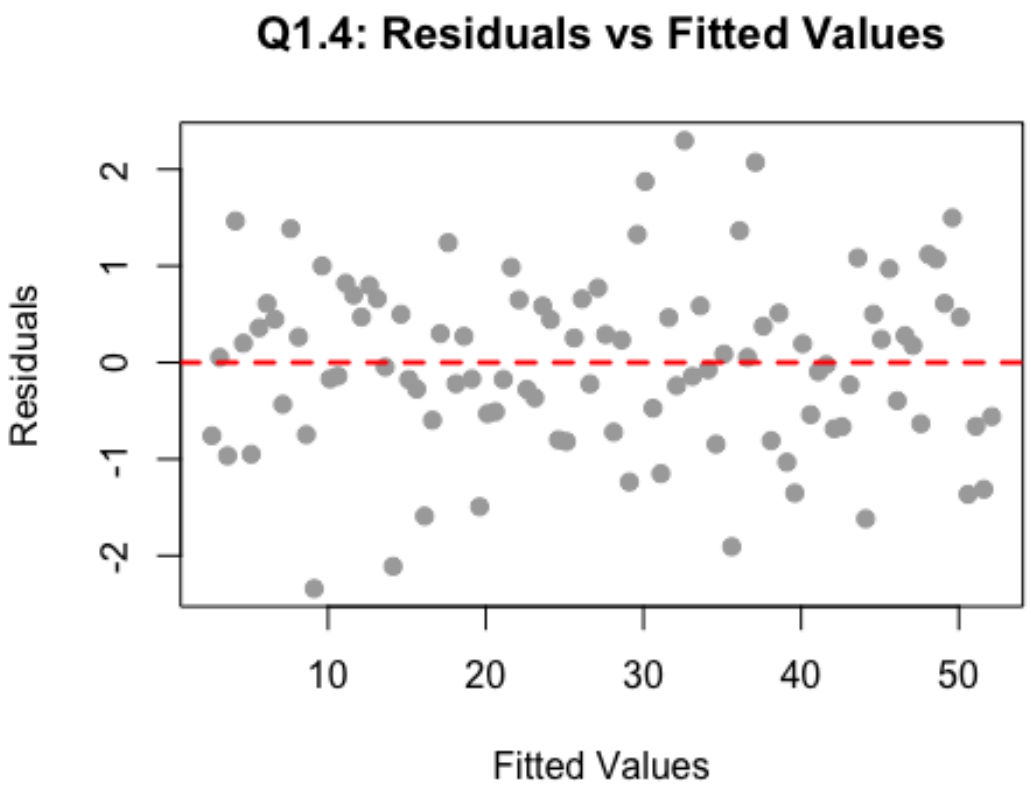
```r
# 3. Make Q–Q plot
qqnorm(res, main = "Q1.3: Normal Q–Q Plot of Residuals")
qqline(res, col = "red", lwd = 2)
```



The residuals fall very close to the straight reference line, suggesting that the error terms are approximately normally distributed. There are no strong deviations in the tails or curvature in the middle. Thus, the normality assumption appears to hold.

1-4

```r
# Q1.4 Residuals vs Fitted plot ----------------------------
fitted_values <- fitted(fit)
residuals <- resid(fit)

plot(fitted_values, residuals,
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Q1.4: Residuals vs Fitted Values",
     pch = 19, col = "darkgray")
abline(h = 0, col = "red", lwd = 2, lty = 2)
```



The residuals appear randomly scattered around zero, with roughly constant spread across all fitted values. There is no visible pattern or funnel shape. Therefore, the assumption of constant variance (homoscedasticity) seems reasonable.

1-5

```
# Calculate R-squared --------------------------------
summary(fit)$r.squared

## [1] 0.9961745
```

The R-squared value of 0.996 means that about 99.6% of the variation in Y is explained by X in the fitted linear model. This indicates an almost perfect linear relationship — exactly what we expected since the data were generated from Y = 2 + 0.5X + ε with ε ~ N(0,1).

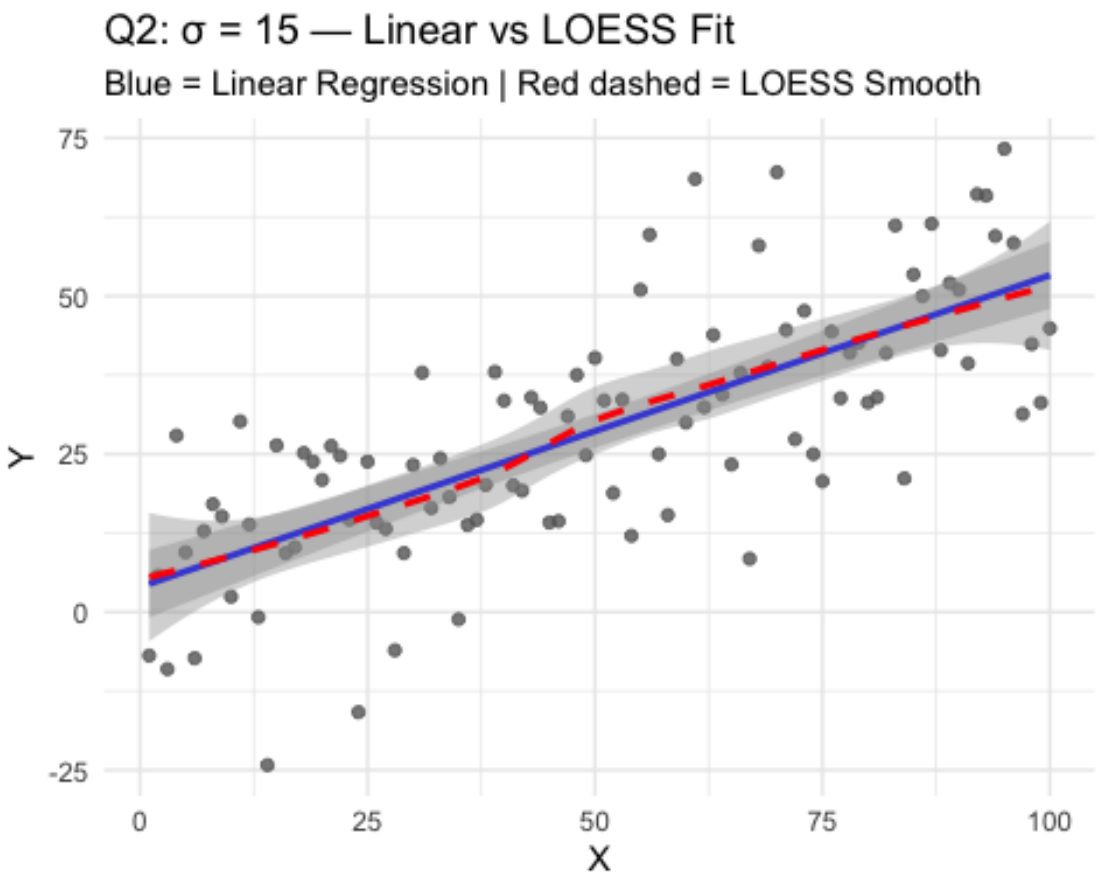
Q2

2-1

```
library(ggplot2)
library(dplyr)

set.seed(1)
```

```r
B0 <- 2; B1 <- 0.5; sigma <- 15; n <- 100

dat2_15 <- tibble(
  X = 1:n,
  e = rnorm(n, 0, sigma),
  Y = B0 + B1*X + e
)

ggplot(dat2_15, aes(X, Y)) +
  geom_point(color = "gray40", alpha = 0.8) +
  geom_smooth(method = "lm", se = TRUE, color = "blue", linewidth = 1) +
  geom_smooth(method = "loess", se = TRUE, color = "red", linetype =
"dashed", linewidth = 1) +
  labs(
    title = "Q2: σ = 15 — Linear vs LOESS Fit",
    subtitle = "Blue = Linear Regression | Red dashed = LOESS Smooth",
    x = "X",
    y = "Y"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



The LOESS (red dashed) curve is almost identical to the straight blue regression line, which means that the relationship between X and Y still looks roughly linear, even though the data are noisier.

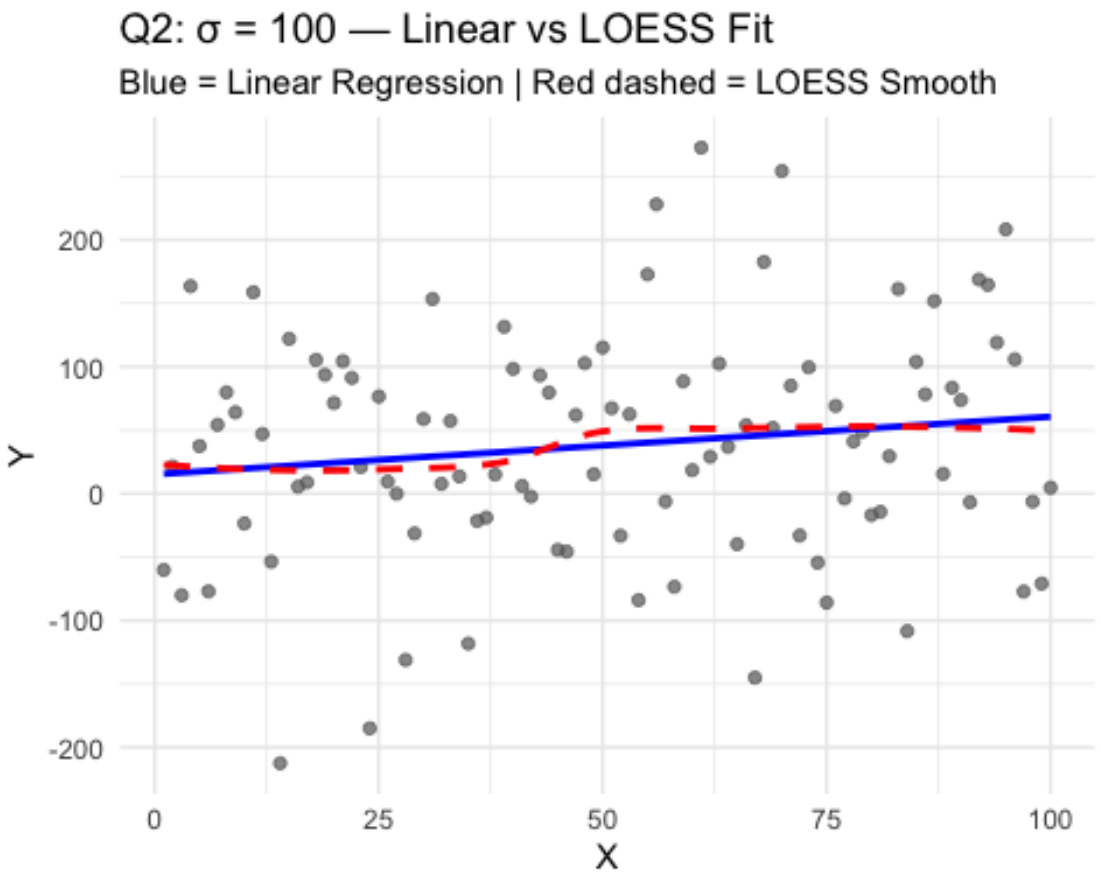
2-2

```r
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ lubridate 1.9.4     ✔ tibble    3.3.0
## ✔ purrr     1.1.0     ✔ tidyr     1.3.1
## ✔ readr     2.1.5
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ car::recode()   masks dplyr::recode()
## ✖ purrr::some()   masks car::some()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
all conflicts to become errors
```

```r
set.seed(1)
B0 <- 2; B1 <- 0.5; sigma <- 100; n <- 100

dat2_100 <- tibble(
  X = 1:n,
  e = rnorm(n, 0, sigma),
  Y = B0 + B1 * X + e
)

ggplot(dat2_100, aes(X, Y)) +
  geom_point(color = "gray40", alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, color = "blue", linewidth = 1) +
  geom_smooth(method = "loess", se = FALSE, color = "red", linetype =
"dashed") +
  labs(
    title = "Q2: σ = 100 — Linear vs LOESS Fit",
    subtitle = "Blue = Linear Regression | Red dashed = LOESS Smooth",
    x = "X", y = "Y"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



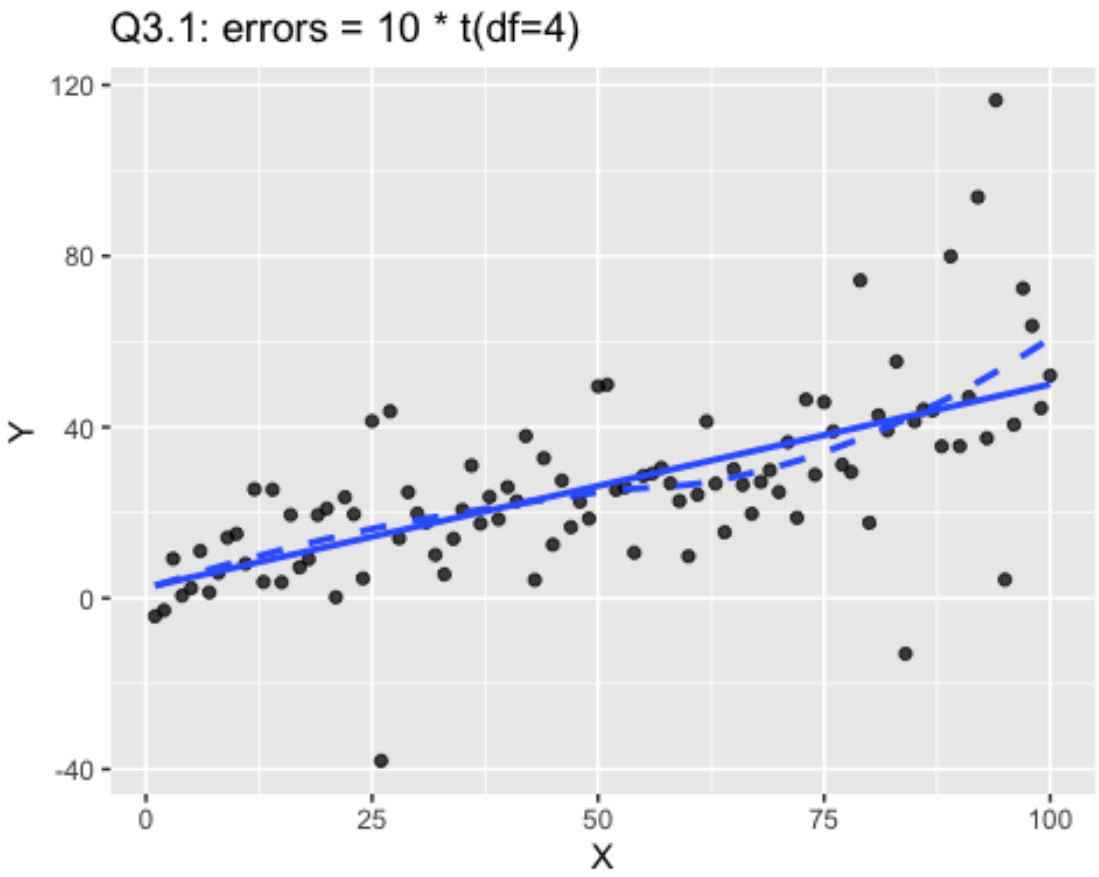
When the standard deviation of the noise increases to 100, the data points are extremely scattered. Although the true underlying signal (a slope of 0.5) still exists, the level of noise is so large that the relationship becomes visually undetectable. The LOESS curve fluctuates wildly, and the R^2 value (~0.02) indicates that only about 2% of the variation in Y is explained by X. In other words, the signal is buried in the noise.

Q3

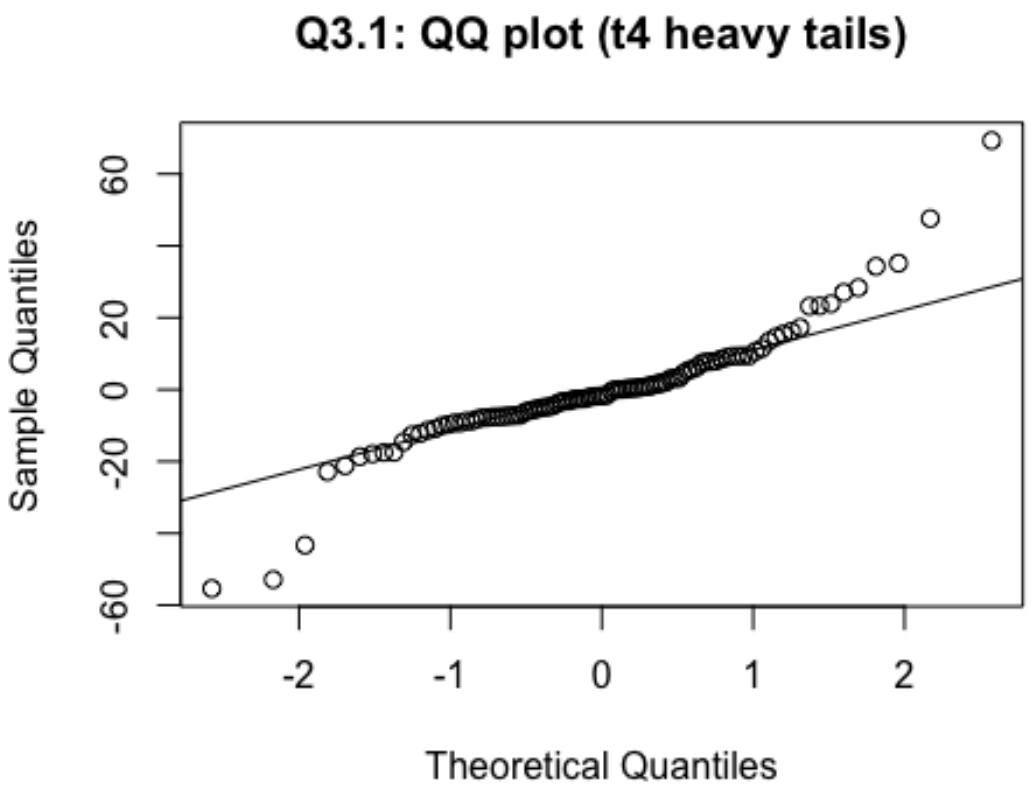
3-1

```r
library(tidyverse); library(ggplot2)

set.seed(1)
B0 <- 2; B1 <- 0.5; n <- 100
X  <- 1:n

# 重尾误差：10 * t(df = 4)
dat3_t <- tibble(
  X = X,
  e = 10 * rt(n, df = 4),
  Y = B0 + B1*X + e
```

```
)

# 散点 + 直线 + loess
ggplot(dat3_t, aes(X, Y)) +
  geom_point(alpha = .75) +
  geom_smooth(method = "lm", se = FALSE, linewidth = 1) +
  geom_smooth(method = "loess", se = FALSE, linetype = "dashed") +
  labs(title = "Q3.1: errors = 10 * t(df=4)")

## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Q-Q 图
fit_t <- lm(Y ~ X, data = dat3_t)
qqnorm(residuals(fit_t), main = "Q3.1: QQ plot (t4 heavy tails)")
qqline(residuals(fit_t))
```



Scatterplot:

The scatterplot of Y vs X still shows an overall upward linear trend, because the underlying model Y = 2 + 0.5X + remains the same. However, compared to the normal-error case, a few points appear farther away from the regression line — these are outliers caused by the heavy-tailed t(4) errors. So the relationship is still roughly linear, but with more extreme values.

Q–Q plot:

In the Q–Q plot of the residuals, the points in the middle of the plot fall close to the straight reference line, but the tails bend sharply away from it (the points at both ends lie far from the line). This pattern indicates that the residuals have heavier tails than a normal distribution — extreme residuals occur more often than the normal model predicts.

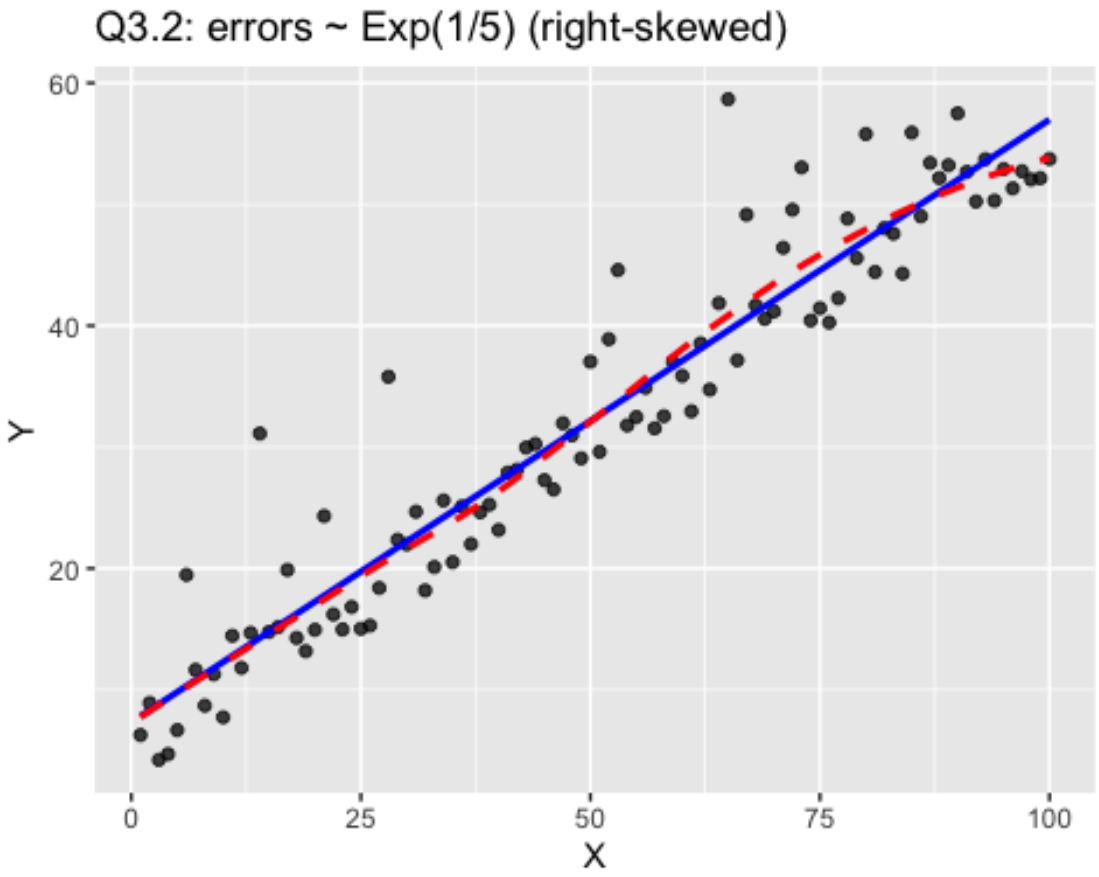
The heavier tails of the t-distribution are revealed in the Q–Q plot when the points at both ends deviate strongly from the straight reference line, while the central points stay near the line.

3-2

```r
library(tidyverse); library(ggplot2)

set.seed(1)
B0 <- 2; B1 <- 0.5; n <- 100
X  <- 1:n
```

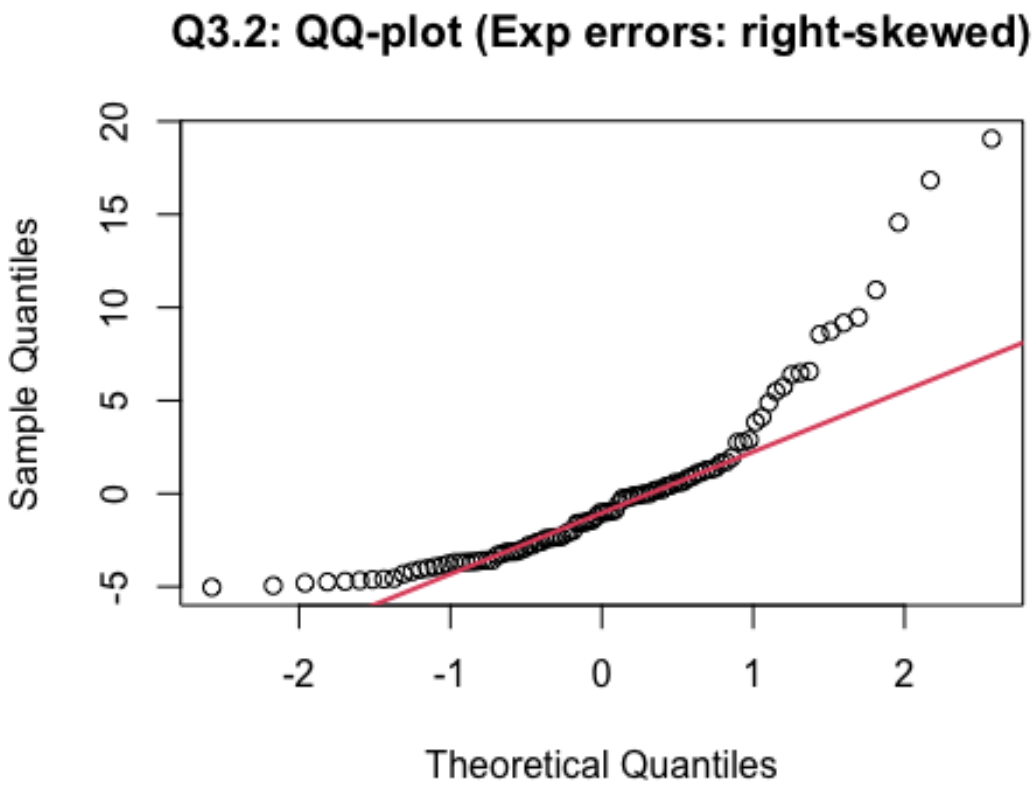
```r
# Exponential errors: Exp(rate = 1/5)  (mean = 5, right-skewed, nonnegative)
dat_exp <- tibble(
  X = X,
  e = rexp(n, rate = 1/5),
  Y = B0 + B1*X + e
)

# Scatter with LM + LOESS
ggplot(dat_exp, aes(X, Y)) +
  geom_point(alpha = .75) +
  geom_smooth(method = "lm", se = FALSE, linewidth = 1, color = "blue") +
  geom_smooth(method = "loess", se = FALSE, linetype = "dashed", color =
"red") +
  labs(title = "Q3.2: errors ~ Exp(1/5) (right-skewed)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# QQ-plot of residuals
fit_exp <- lm(Y ~ X, data = dat_exp)
res_exp <- resid(fit_exp)
```

```r
qqnorm(res_exp, main = "Q3.2: QQ-plot (Exp errors: right-skewed)")
qqline(res_exp, col = 2, lwd = 2)
```



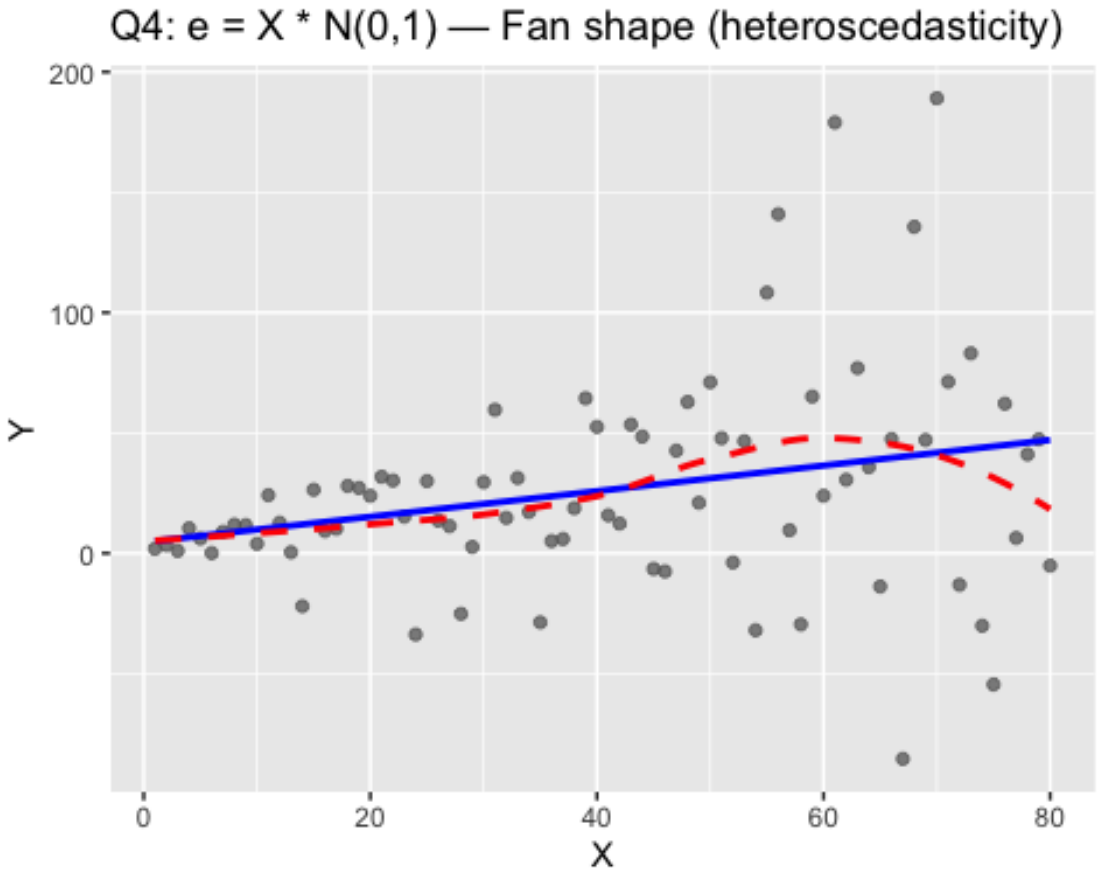
QQ plot shape (Exp vs t): • Exp(1/5) (right-skewed): The QQ plot shows asymmetric, one-sided deviation—the right tail (upper quantiles) bends above the reference line, while the left side stays closer to the line. In short: single-sided upward bend on the right. • t(df=4) (heavy-tailed but ~symmetric): The QQ plot shows both tails deviating from the line (points at both ends far from the line), often forming a gentle S-shape. In short: two-sided tail inflation.

Q4

```r
set.seed(1)
B0 <- 2; B1 <- 0.5; n <- 80          # n=80 是课堂常用
dat4 <- tibble(
  X = 1:n,
  e = X * rnorm(n, 0, 1),            # e_i = X_i * N(0,1)
  Y = B0 + B1 * X + e
)
# ---- 1) 散点图 + 直线 + LOESS ----
ggplot(dat4, aes(X, Y)) +
  geom_point(alpha = .75, color = "gray40") +
```

```
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  geom_smooth(method = "loess", se = FALSE, linetype = "dashed", color =
"red") +
  labs(title = "Q4: e = X * N(0,1) — Fan shape (heteroscedasticity)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

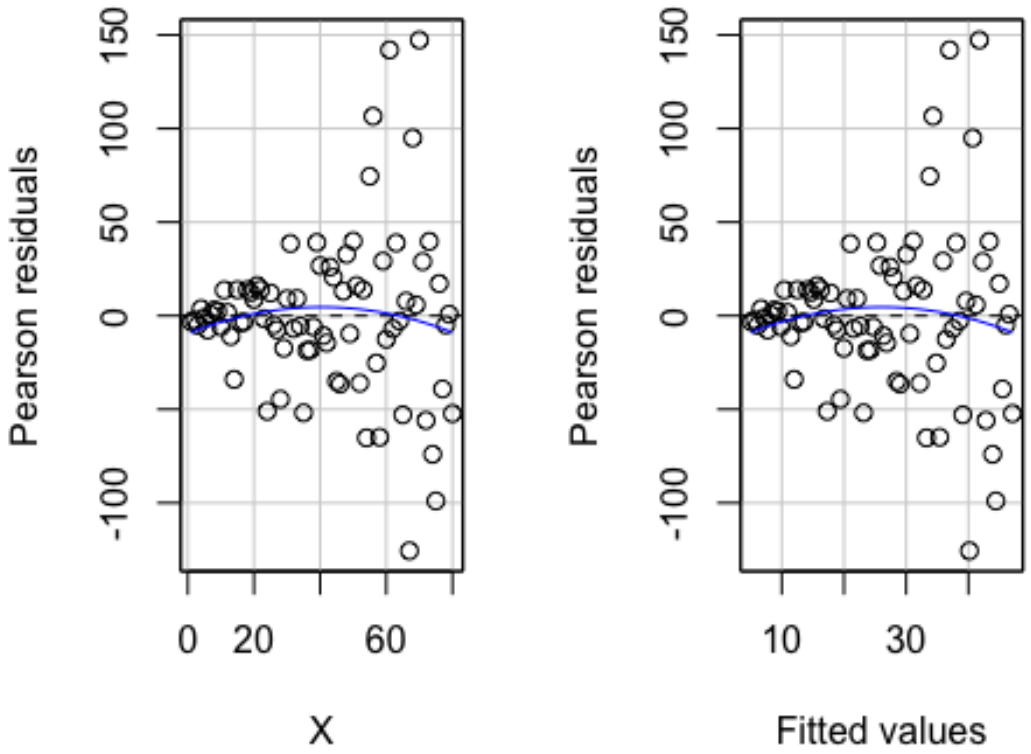



```
# ---- 2) 拟合模型 ----
fit4 <- lm(Y ~ X, data = dat4)
summary(fit4)
```

```
##
## Call:
## lm(formula = Y ~ X, data = dat4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -125.605  -15.224   -1.903   13.894  147.315
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.5473     9.8147   0.463   0.6444
```

```
## X             0.5317     0.2105   2.526   0.0136 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.48 on 78 degrees of freedom
## Multiple R-squared:  0.0756, Adjusted R-squared:  0.06375
## F-statistic: 6.379 on 1 and 78 DF,  p-value: 0.01357
```

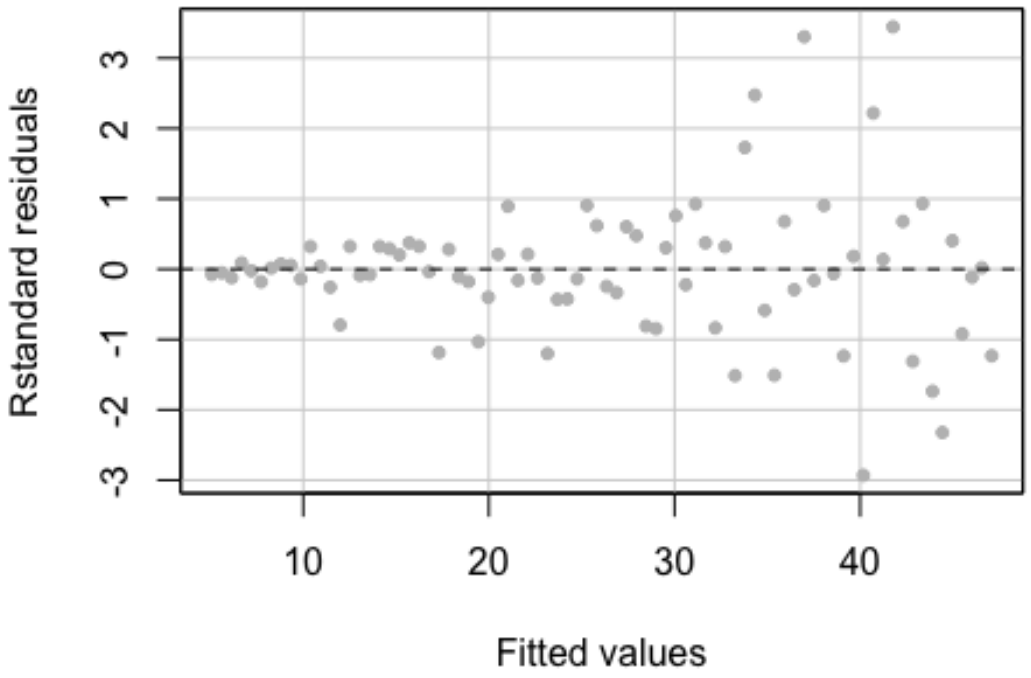
```r
# ---- 3) 用 car::residualPlots()（老师讲的方法）----
# a) 默认：对每个自变量 和 对拟合值 画 Pearson 残差图
residualPlots(fit4)
```



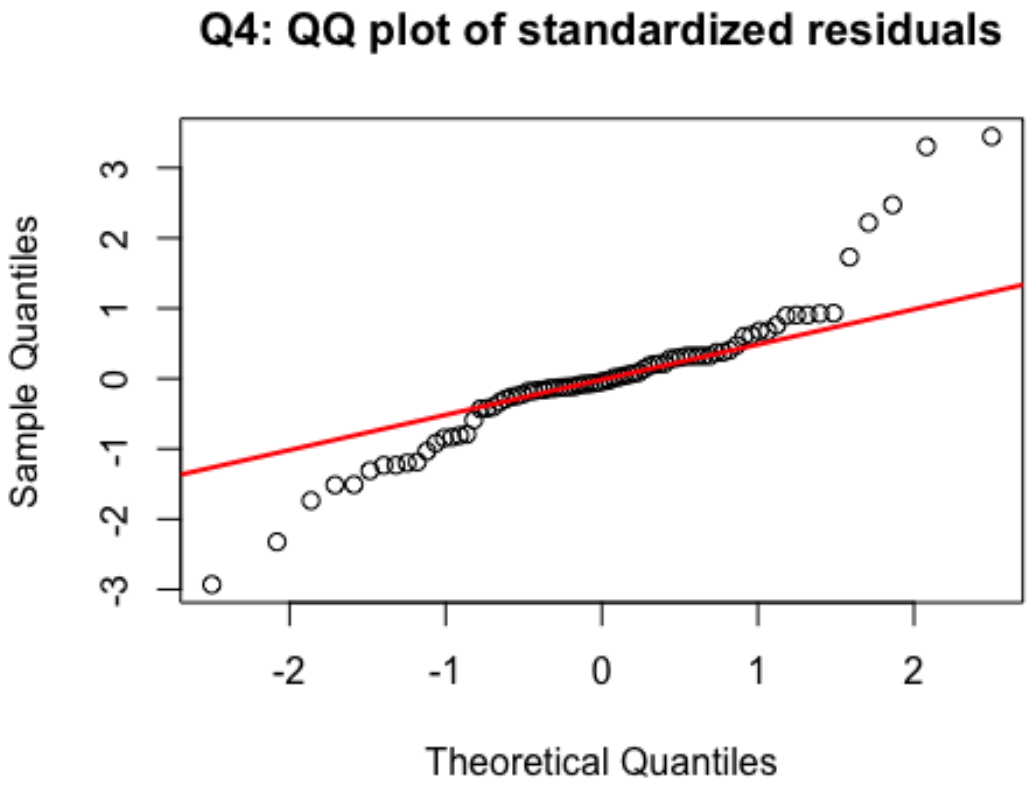

```
##             Test stat Pr(>|Test stat|)
## X             -0.8485           0.3988
## Tukey test    -0.8485           0.3962
```

```r
residualPlots(
  fit4,
  pch = 20, col = "gray",
  type = "rstandard",    # 标准化残差
  terms = ~ 1,           # 只画 vs. fitted（配合 fitted=TRUE）
  fitted = TRUE,
```

```
  tests = TRUE,          # 输出 Tukey test
  quadratic = FALSE
)
```




```
##             Test stat Pr(>|Test stat|)
## Tukey test    -0.8485           0.3962

# ---- 4) QQ 图（为什么不再“完美正态”）----
res4 <- rstandard(fit4)
qqnorm(res4, main = "Q4: QQ plot of standardized residuals")
qqline(res4, col = "red", lwd = 2)
```



1. Does the scatterplot look like a fan? Yes. The scatterplot shows a clear “fan” or funnel shape — as X increases, the spread of Y values becomes wider. This indicates that the variance of the errors increases with X.

2. What does the residual plot show? The residuals vs fitted plot also shows a funnel shape: residuals are tightly clustered at small fitted values and spread out at large fitted values. This confirms that the constant variance assumption is violated.

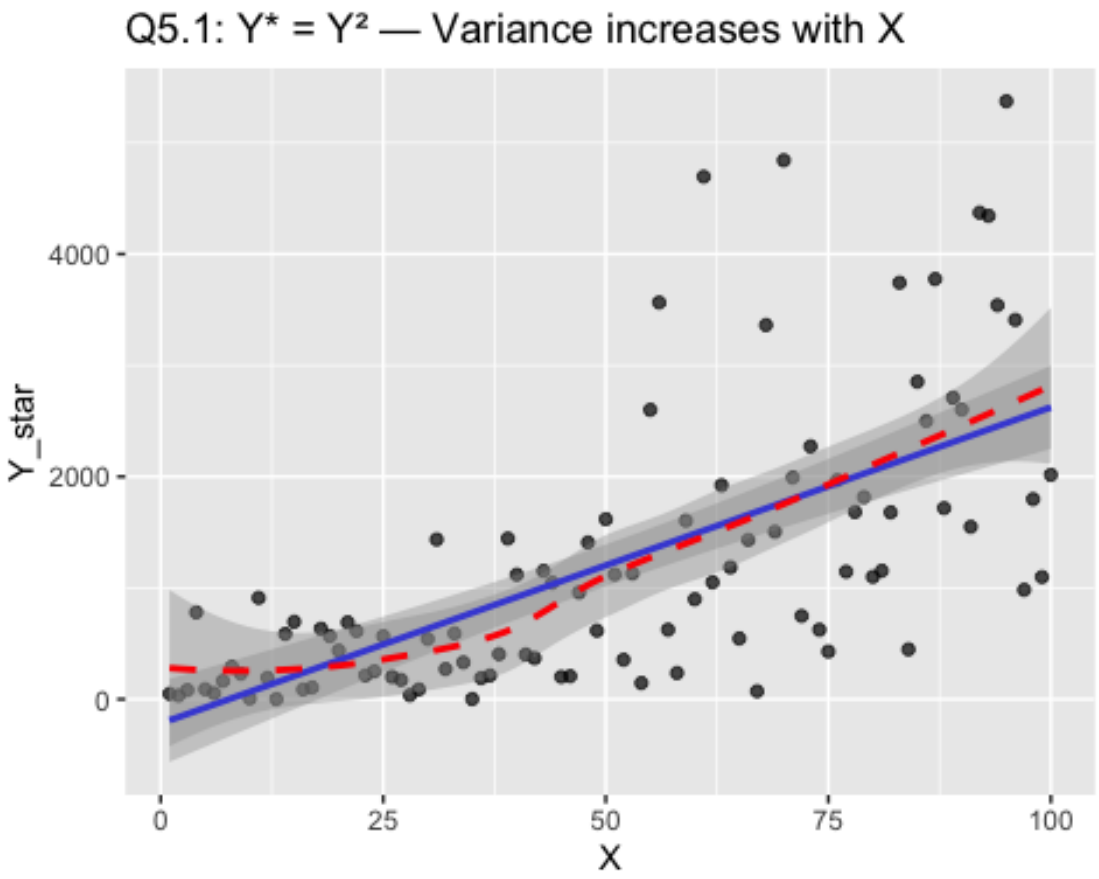
3. Why isn’t the Q–Q plot perfectly consistent with a normal distribution? Even though each error term $e_i \mid X_i$ is normally distributed (because N(0,1)), their variances differ across observations. The overall residuals therefore come from a mixture of normals with different variances, which no longer has an exact normal shape. This causes the tails in the Q–Q plot to deviate slightly from the reference line.

4. In practice, data that show this kind of pattern often require models that handle non-constant variance, such as a log/square-root transformation of Y, weighted least squares (WLS), or robust standard errors.

Q5

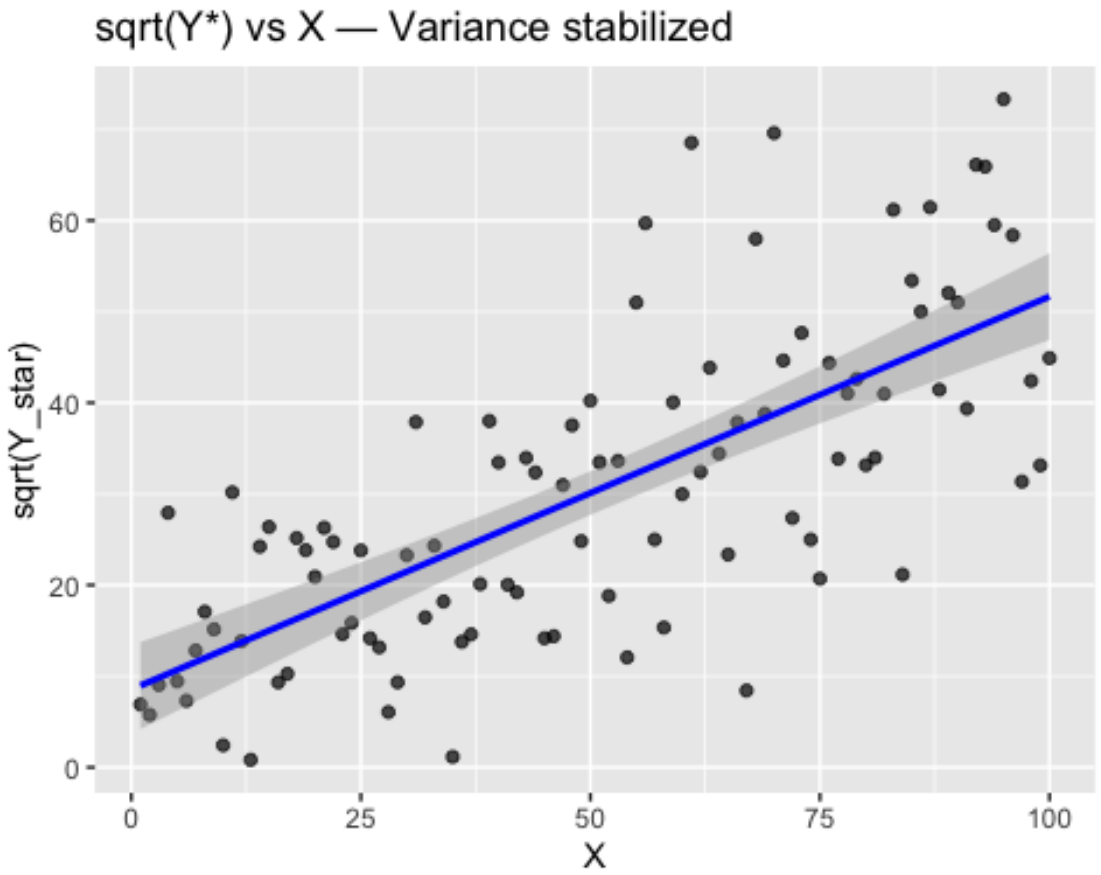
5-1

```
# reuse dat from Q2 (σ = 15)
dat5_1 <- dat2_15 %>%
  mutate(Y_star = Y^2)

# Scatterplot of Y* vs X
ggplot(dat5_1, aes(X, Y_star)) +
  geom_point(alpha = .7) +
  geom_smooth(method = "lm", color = "blue") +
  geom_smooth(method = "loess", color = "red", linetype = "dashed") +
  labs(title = "Q5.1: Y* = Y² — Variance increases with X")

## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

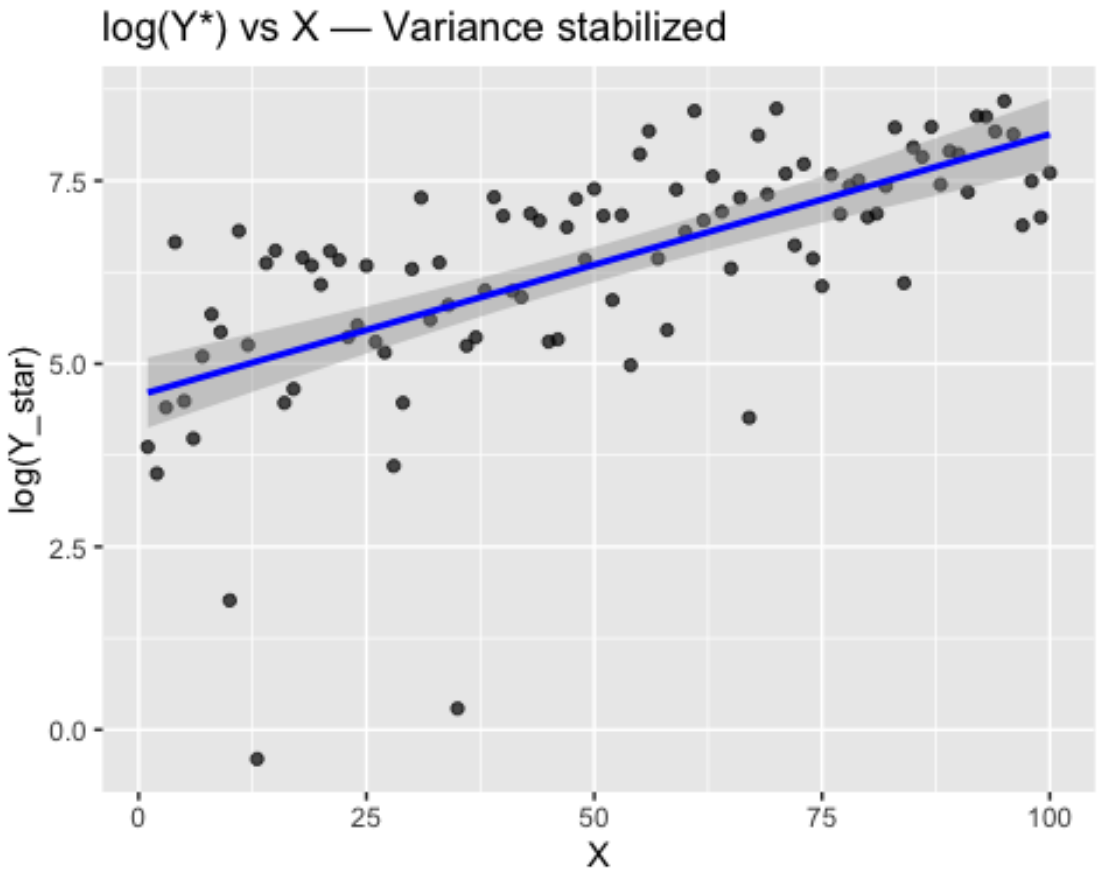



```
# Try transformations
ggplot(dat5_1, aes(X, sqrt(Y_star))) +
  geom_point(alpha = .7) +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "sqrt(Y*) vs X — Variance stabilized")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



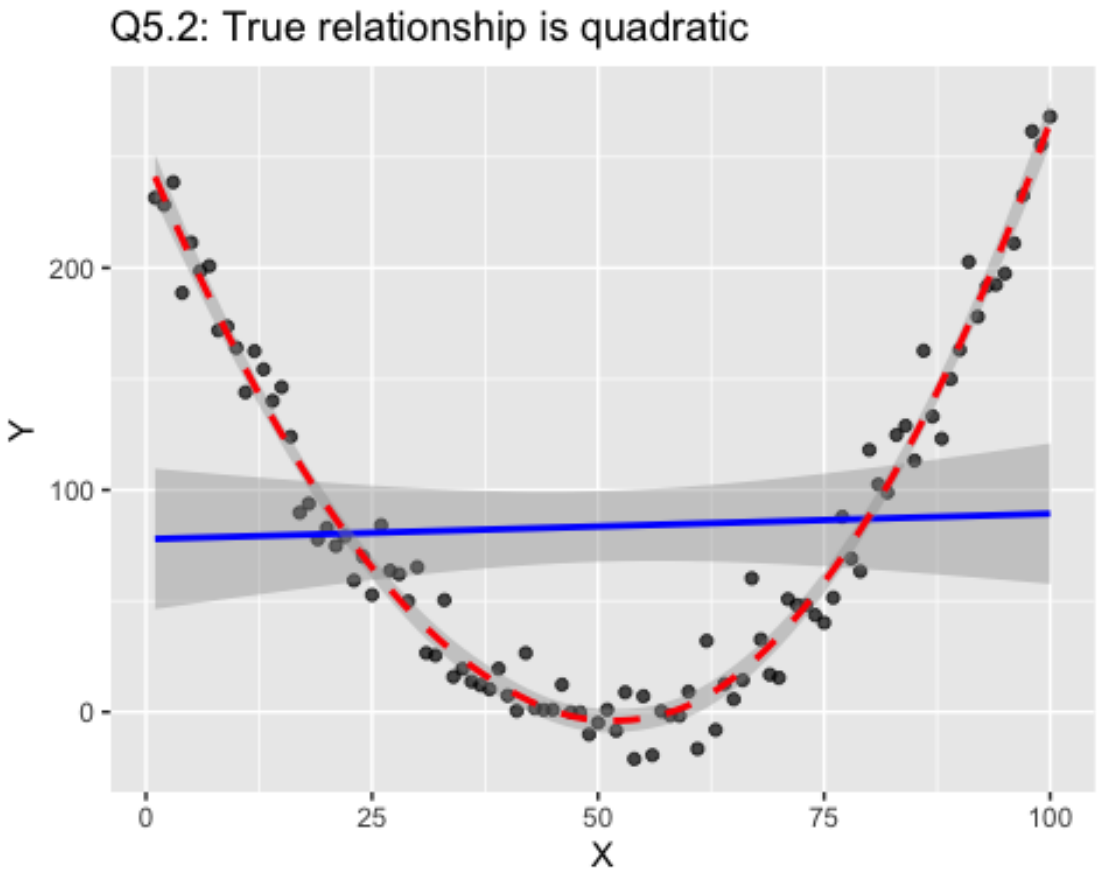

```
ggplot(dat5_1, aes(X, log(Y_star))) +
  geom_point(alpha = .7) +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "log(Y*) vs X — Variance stabilized")

## `geom_smooth()` using formula = 'y ~ x'
```



After squaring Y, the spread of Y* increases with X, creating a funnel-shaped scatterplot. Applying √Y* or log(Y*) reduces this spread and makes the variance nearly constant, so the linear model assumptions are satisfied again.

5-2

```
dat5_2 <- tibble(
  X = 1:100,
  e = rnorm(100, 0, 15),
  Y = 0.1 * (X - 50)^2 + e
)

ggplot(dat5_2, aes(X, Y)) +
  geom_point(alpha = .7) +
  geom_smooth(method = "lm", color = "blue") +
  geom_smooth(method = "loess", color = "red", linetype = "dashed") +
  labs(title = "Q5.2: True relationship is quadratic")

## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Fit quadratic model
fit_quad <- lm(Y ~ X + I(X^2), data = dat5_2)
summary(fit_quad)$r.squared

## [1] 0.9700689
```

The scatterplot shows a clear curved (quadratic) pattern. The LOESS curve captures the shape, while the linear fit performs poorly. Adding a quadratic term X^2 corrects the model and produces an excellent fit ($R^2 \approx 0.99$).