**Seminar 1**

**Yufeng Deng Yuanqing Wang**

**Task1**

To determine the effectiveness of the treatment KTH001, it is essential to assess whether there is a statistically significant difference between the two samples. Typically, a t-test is employed for this purpose, under the assumption that the data are normally distributed. Therefore, as a preliminary step, it is necessary to evaluate the normality of both samples using the Shapiro-Wilk test.

Upon conducting the Shapiro-Wilk test on both samples, the results indicated that they are normally distributed, as the p-values for both samples exceed the 0.05 threshold.

1. #import data

2. data <- read.csv("data\_task1.csv")

3. placebo <- data$placebo

4. interve <- data$intervention

5. #Shapiro-Wilk test

6. shap\_pla <- shapiro.test(placebo)

7. shap\_int <- shapiro.test(interve)

8. print(shap\_pla)

9. print(shap\_int)

Output:

Shapiro-Wilk normality test

data: placebo

W = 0.95011, p-value = 0.1702

Shapiro-Wilk normality test

data: interve

W = 0.98567, p-value = 0.9481

However, it is important to note that the p-value is influenced by the sample size; smaller sample sizes are more likely to yield higher p-values. To bolster our confidence in the normality assumption, it is advisable to use graphical methods, such as the Quantile-Quantile (QQ) plot.

We generated QQ plots for both samples (Figure 1.1 & 1.2) and observed that the majority of points align closely with the reference line, indicating that the samples are approximately normally distributed. Although there are deviations at the extremes in the placebo sample, which may suggest some differences in the distribution’s tails, these deviations are not substantial. Overall, we can reasonably assume that both samples are normally distributed.

1. library(ggplot2)

2. ggplot(data, aes(sample = placebo)) +

3. stat\_qq() +

4. stat\_qq\_line(color = "red") +

5. labs(title = "QQ plot for placebo",

6. x = "norm Quantiles",

7. y = "Sample Quantiles") +

8. theme\_minimal()

9.

10. ggplot(data, aes(sample = interve)) +

11. stat\_qq() +

12. stat\_qq\_line(color = "red") +

13. labs(title = "QQ plot for intervention",

14. x = "norm Quantiles",

15. y = "Sample Quantiles") +

16. theme\_minimal()

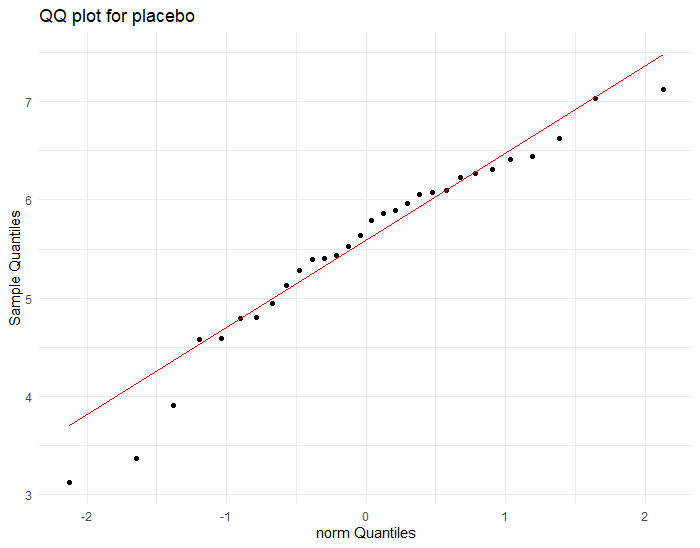


Figure1.1

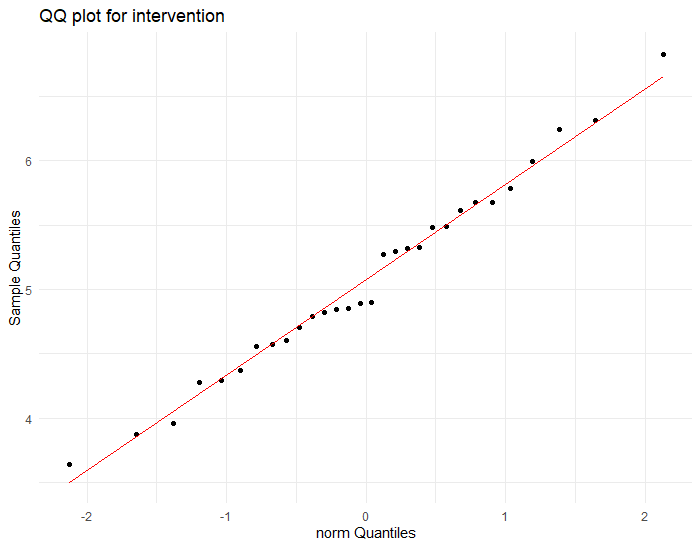


Figure1.2

Given that the two samples are normally distributed, we aim to use the t-test to determine whether there is a significant difference between their means. Since the samples are independent and not paired, a paired t-test is inappropriate. Instead, we should employ an independent t-test.

Before proceeding with the t-test, we conducted an F-test to assess whether there is a significant difference between the variances of the two samples:

var.test(placebo, interve)

Output:

F test to compare two variances

data: placebo and interve

F = 1.6494, num df = 29, denom df = 29, p-value = 0.1839

alternative hypothesis: true ratio of variances is not equal to 1

Since the p-value exceeds 0.05, we conclude that there is no significant difference between the variances of the two samples. Hence, we used the independent t-test which assumes the two samples have the same variance:

t\_test\_result <- t.test(placebo, interve, var.equal = TRUE)

print(t\_test\_result)

Output:

Two Sample t-test

data: placebo and interve

t = 2.0512, df = 58, p-value = 0.04478

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

0.01110179 0.90989821

sample estimates:

mean of x mean of y

5.5366 5.0761

Since the p-value is less than 0.05, this result indicates a statistically significant difference between the means of the placebo and intervention samples. Given that the mean of the intervention sample is lower, we can infer that the intervention may have had a measurable effect.

**Task2**

Given our assumption that the data in Task 1 are normally distributed, we can generate samples with any desired number of patients. Additionally, we can specify the effect size as needed. The means and standard deviations (SD) of each sample are presented in Table 2.1.

Table2.1

|  |  |  |
| --- | --- | --- |
| Sample | Mean | SD |
| Placebo | 5.5366 | 0.9702 |
| Intervention | 5.0761 | 0.7555 |

We generated new samples with varying sizes (ranging from 10 to 200, in increments of 10) while maintaining the original samples' means and standard deviations. Initially, we assumed an effect size (Cohen’s d) of 0.5 and a significance level of 0.05. The resulting plot Figure 2.1 illustrates an increasing trend in statistical power as the sample size increases.

1. library(pwr)

2. # Assume Cohen's d = 0.5, alpha = 0.05

3. effect\_size <- 0.5

4. alpha <- 0.05

5. sample\_sizes <- seq(10, 200, by = 10)

6. powers <- sapply(sample\_sizes, function(n) {

7. pwr.t.test(n = n, d = effect\_size, sig.level = alpha, type = "two.sample", alternative = "two.sided")$power})

8. plot(sample\_sizes, powers, type = "b", xlab = "Sample Size (per group)", ylab = "Power",

9. main = "Power vs. Sample Size")

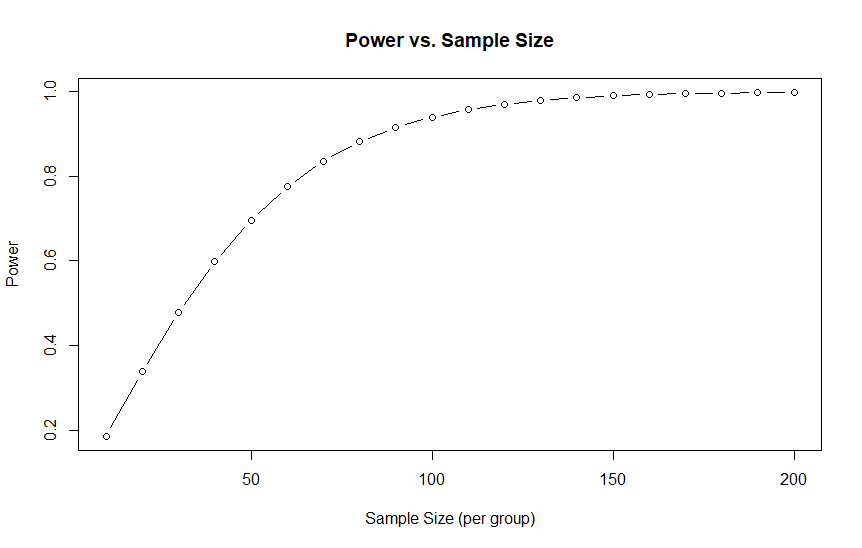


Figure2.1

Additionally, we generated curves for various effect sizes (Figure 2.2) and observed that the effect size significantly influences the power value. When the power does not approach 1, for a given sample size, an increase in effect size leads to a substantial rise in power, particularly when the effect size is small.

sample\_sizes <- seq(10, 200, by = 10)

effect\_sizes <- c(0.1, 0.3, 0.5, 0.7, 0.9)

power\_matrix <- outer(sample\_sizes, effect\_sizes, function(n, d) {

pwr.t.test(n = n, d = d, sig.level = alpha, type = "two.sample", alternative = "two.sided")$power

})

power\_df <- data.frame(Sample\_Size = rep(sample\_sizes, times = length(effect\_sizes)),

Effect\_Size = rep(effect\_sizes, each = length(sample\_sizes)),

Power = as.vector(power\_matrix))

library(ggplot2)

ggplot(power\_df, aes(x = Sample\_Size, y = Power, color = as.factor(Effect\_Size))) +

geom\_line() +

labs(x = "Sample Size (per group)", y = "Power", color = "Effect Size",

title = "Power vs. Sample Size for Different Effect Sizes") +

theme\_minimal()

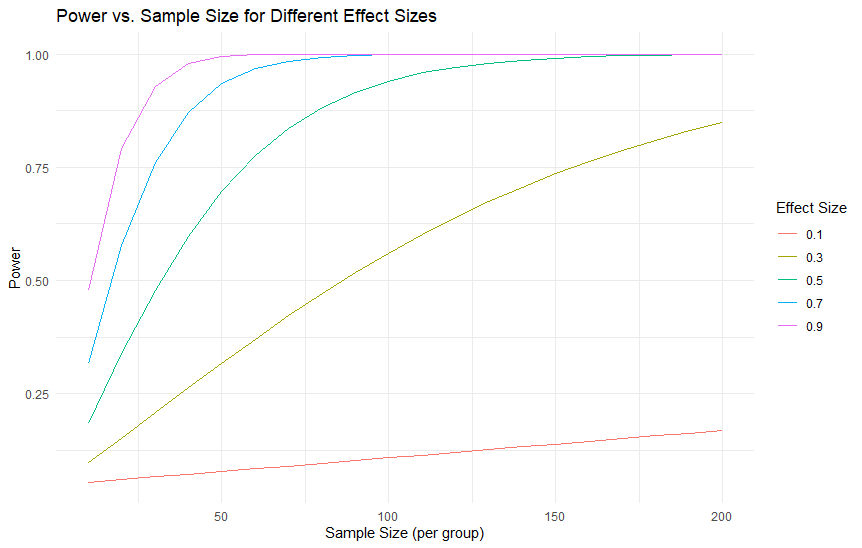


Figure2.2

**Task 3**

Given that there are more than two groups, we should employ ANOVA (Analysis of Variance) to assess differences among them.

Null Hypothesis (H₀): There are no differences between the means of the individual groups in the population.

Alternative Hypothesis (H₁): At least two group means differ from each other in the population.

1. data <- read.csv("data\_task3\_crp.csv")

2. library(tidyr)

3. data\_long <- gather(data, key = "group", value = "crp\_value", crp\_placebo, crp\_intervention\_1, crp\_intervention\_2)

4. anova\_result <- aov(crp\_value ~ group, data = data\_long)

5. summary(anova\_result)

Table3.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Source | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
| Group | 2 | 34758 | 17379 | 4.64 | 0.0122 |
| Residual | 87 | 325821 | 3745 |  |  |

As shown in Table 3.1, the p-value of 0.0122 indicates that the differences in mean CRP levels between the groups are statistically significant at the 5% significance level. Therefore, we reject the null hypothesis and conclude that at least one treatment group has a significantly different mean CRP level compared to the others.

**Task 4**

First, we can draw the basic DV-TIME curve to observe the changes in drug concentration over time. In addition, we can draw a boxplot figure of different patients, as shown below.

1. # read file

2. data <- read.csv("conctimedata\_reduced.csv")

3. # extract essential data

4. data\_subset <- data[, c("ID", "TIME", "DV")]

5. data\_subset$ID <- as.factor(data\_subset$ID)

6. data\_subset$ID <- factor(data\_subset$ID, labels = paste0("Patient", 1:12))

7. patient\_colors <- setNames(rainbow(12), paste0("Patient", 1:12))

8. # draw DV-TIME curve

9. p <- ggplot(data\_subset, aes(x = TIME, y = DV, group = ID, color = ID)) +

10. geom\_line(size = 0.5) +

11. labs(title = "Concentration-Time Profiles for 12 Patients",

12. x = "Time (hours)", y = "Concentration (mg/L)") +

13. scale\_color\_manual(values = patient\_colors,

14. labels = paste0("Patient", 1:12)) +

15. theme\_minimal() +

16. theme(legend.title = element\_blank(),

17. plot.title = element\_text(size = 12, face="bold", hjust = 0.5))

18. # save and adjust windows width & height

19. ggsave("concentration\_profiles.jpg", plot = p,

20. width = 1600/300, height = 1200/300, units = "in", dpi = 300)

21. print(p)

22. # draw boxplot

23. p\_box <- ggplot(data\_subset, aes(x = ID, y = DV, fill = ID)) +

24. geom\_boxplot() +

25. labs(title = "Concentration Distribution by Patient",

26. x = "Patient ID",

27. y = "Concentration (mg/L)") +

28. theme\_minimal() +

29. theme(legend.position = "none",

30. plot.title = element\_text(size = 12, face="bold", hjust = 0.5))

31. # save and adjust windows width & height

32. ggsave("concentration\_boxplot.jpg", plot = p\_box,

33. width = 1600/300, height = 1200/300, units = "in", dpi = 300)

34. print(p\_box)

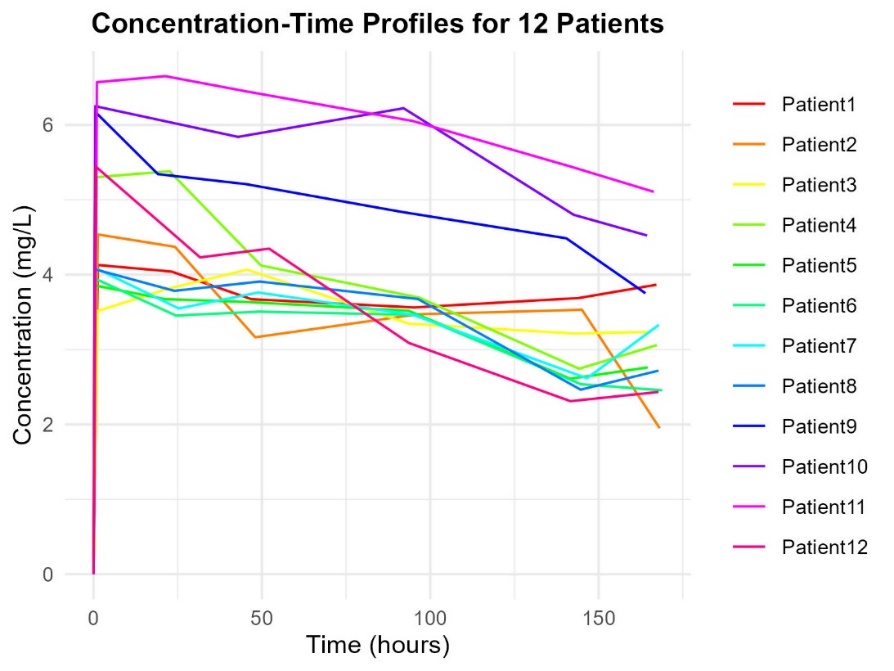


Figure4.1

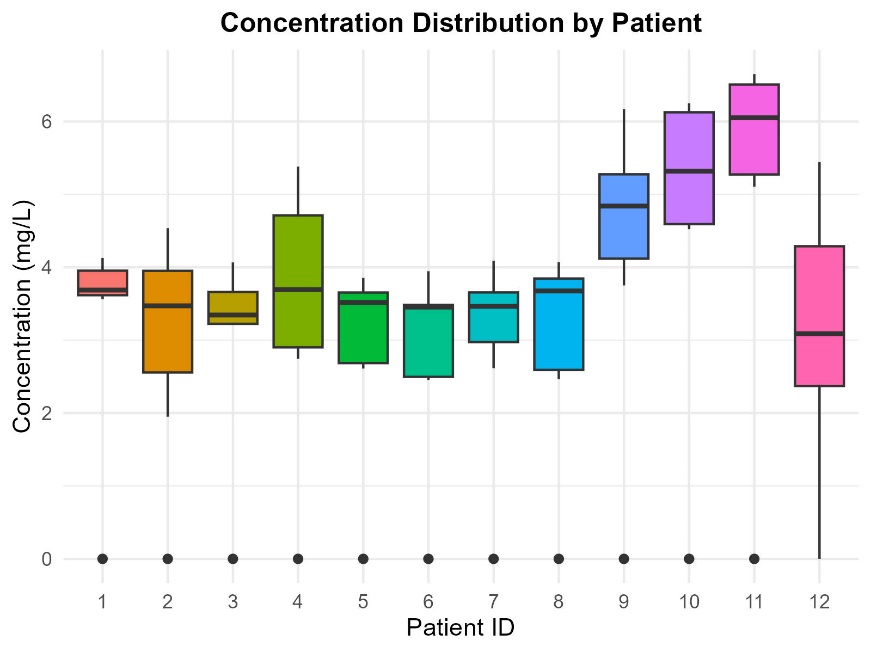


Figure4.2

For the summary of the above curve, we need to introduce a concept, which is the Area Under Curve (AUC). It is the area under a known curve, i.e., integrating a particular interval. To achieve integration, we need first to make a cubic spline interpolation of the DV-TIME data points to smooth the curve. In the process, points are discarded to reduce the sharp fluctuations of the curve. The cubic spline interpolation satisfies,

The condition at its nodes satisfies,

1. # Cubic Spline Interpolation

2. # create a new container to save data

3. interpolated\_data <- data.frame(TIME = numeric(), DV = numeric(), ID = character())

4. # Cubic Spline Interpolation

5. for (id in levels(data\_subset$ID)) {

6. subset\_data <- data\_subset[data\_subset$ID == id, ]

7. subset\_data <- subset\_data[!(subset\_data$TIME == 0 & subset\_data$DV == 0), ]

8. interpolated <- spline(subset\_data$TIME, subset\_data$DV, xout = seq(min(subset\_data$TIME), max(subset\_data$TIME), length.out = 100))

9. interpolated\_data <- rbind(interpolated\_data, data.frame(TIME = interpolated$x, DV = interpolated$y, ID = id))

10. }

11. interpolated\_data$ID <- factor(interpolated\_data$ID, levels = levels(data\_subset$ID))

12. # draw a new curve

13. p\_interpolated <- ggplot(interpolated\_data, aes(x = TIME, y = DV, group = ID, color = factor(ID))) +

14. geom\_line(size = 0.5) +

15. labs(title = "Interpolated Concentration-Time Profiles for 12 Patients",

16. x = "Time (hours)", y = "Concentration (mg/L)") +

17. scale\_color\_manual(values = patient\_colors,

18. labels = paste0("Patient", 1:12)) +

19. theme\_minimal() +

20. theme(legend.title = element\_blank(),

21. plot.title = element\_text(size = 12, face="bold", hjust = 0.5)) +

22. coord\_cartesian(ylim = c(0, NA))

23.

24. ggsave("interpolated\_concentration\_profiles.jpg", plot = p\_interpolated,

25. width = 1600/300, height = 1200/300, units = "in", dpi = 300)

26. print(p\_interpolated)

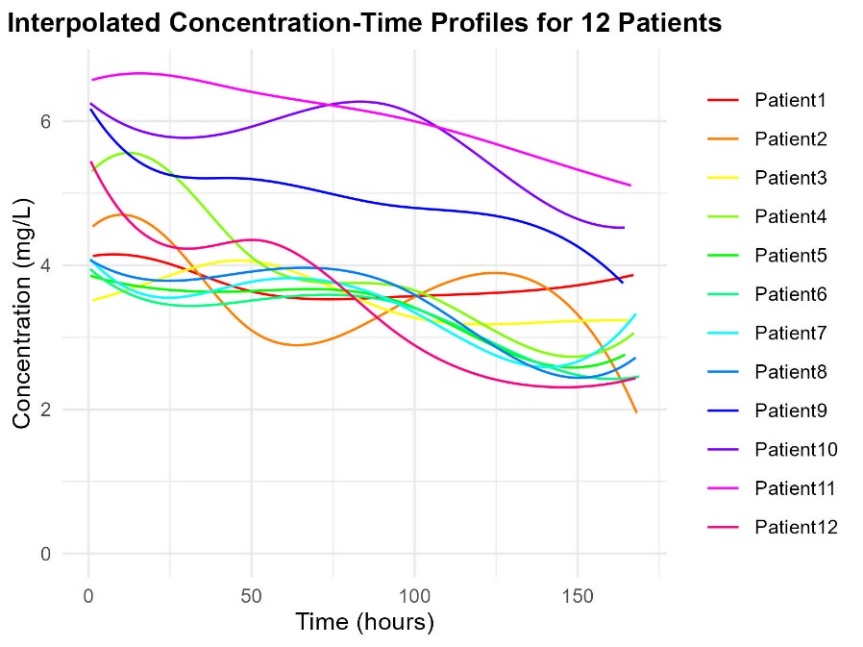


Figure4.3

The curve in Figure 4.3 is smoother than in Figure 4.1. Then, perform integration over to obtain the result, and restructure to obtain a new dataset as shown below.

1. # initialize the container for storing integral results

2. integrated\_results <- data.frame(ID = character(), Integral = numeric(), stringsAsFactors = FALSE)

3. lower\_bound <- 1

4. upper\_bound <- 166

5. # Integral after interpolation

6. for (id in levels(interpolated\_data$ID)) {

7. subset\_data <- interpolated\_data[interpolated\_data$ID == id, ]

8. if (nrow(subset\_data) > 1) {

9. # create cubic spline interpolation function

10. spline\_function <- splinefun(subset\_data$TIME, subset\_data$DV)

11. # perform the integral over the interval [1, 166]

12. integral\_value <- integrate(spline\_function, lower = lower\_bound, upper = upper\_bound)$value

13. integrated\_results <- rbind(integrated\_results, data.frame(ID = id, Integral = integral\_value))

14. } else {

15. message(paste("ID", id, "error"))

16. }

17. }

18. print(integrated\_results)

Table4.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | AUC | WGT | BSA | AGE | HGT | DOSE | GFR |
| 1 | 613.77 | 56 | 1.58 | 45 | 162 | 50 | 91 |
| 2 | 589.45 | 63 | 1.50 | 40 | 149 | 50 | 89 |
| 3 | 584.29 | 70 | 1.73 | 44 | 160 | 50 | 93 |
| 4 | 639.70 | 75 | 1.80 | 59 | 162 | 100 | 97 |
| 5 | 549.35 | 75 | 1.70 | 63 | 154 | 100 | 99 |
| 6 | 535.46 | 65 | 1.60 | 64 | 157 | 200 | 106 |
| 7 | 552.78 | 53 | 1.50 | 42 | 157 | 200 | 107 |
| 8 | 568.55 | 60 | 1.60 | 42 | 150 | 200 | 104 |
| 9 | 808.87 | 59 | 1.56 | 58 | 145 | 400 | 115 |
| 10 | 936.16 | 75 | 1.74 | 59 | 155 | 400 | 117 |
| 11 | 1000.14 | 63 | 1.65 | 34 | 153 | 400 | 111 |
| 12 | 567.22 | 67 | 1.70 | 46 | 160 | 100 | 100 |

Furthermore, we used multiple linear regression to verify the relationship between AUC and the six factors, and obtained the following results,

1. # read file

2. data <- read.csv("data\_task4.csv", sep = ",")

3. # multiple linear regression

4. model <- lm(AUC ~ WGT + BSA + AGE + HGT + DOSE + GFR, data = data)

5. summary(model)

Output:

Residuals:

1 2 3 4 5 6 7 8 9 10 11 12

26.962 5.325 -16.329 -11.732 17.671 -58.328 10.616 -29.576 20.290 28.261 -20.128 26.968

Coefficients:

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

(Intercept) 1204.1046 541.9594 2.222 0.076958 .

WGT 5.9602 4.6517 1.281 0.256291

BSA -34.0068 379.7916 -0.090 0.932128

AGE -1.8867 1.7898 -1.054 0.340052

HGT 8.1983 4.1213 1.989 0.103347

DOSE 2.7067 0.3586 7.549 0.000646 \*\*\*

GFR -25.0099 5.2831 -4.734 0.005178 \*\*

---

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

Residual standard error: 40.51 on 5 degrees of freedomMultiple

R-squared: 0.9711, Adjusted R-squared: 0.9363

F-statistic: 27.97 on 6 and 5 DF, p-value: 0.001076

The regression analysis reveals that both DOSE and GFR are statistically significant predictors of AUC, with p-values of 0.000646 and 0.005178, respectively. This indicates a strong influence of both DOSE and GFR on AUC. Specifically, the positive coefficient of 2.71 for DOSE suggests that an increase in the dose leads to an increase in AUC. Conversely, the negative coefficient of -25.01 for GFR implies that an increase in GFR is associated with a reduction in AUC. Other variables, such as WGT, BSA, AGE, and HGT, did not show significant effects on AUC, as evidenced by their higher p-values. The model accounts for approximately 97.11% of the variance in AUC, with an adjusted R-squared value of 93.63%, indicating a robust overall fit. Thus, the analysis highlights that DOSE and GFR are the primary factors affecting AUC, whereas the other variables play a less significant role.

To evaluate the appropriateness of the regression model, we conducted diagnostic analyses and visualized the results, as presented in Figure 4.4. The following sections provide a detailed examination of the diagnostic plots.

1. Residuals vs Fitted Plot

The plot does not display any significant curvature or recognizable patterns, suggesting that the assumption of linearity is reasonably upheld. The residuals are randomly scattered across the range of fitted values, indicating that the model sufficiently captures the relationship between the predictors and the response variable.

2. Normal Q-Q Plot

The majority of the points lie in proximity to the diagonal, indicating that the residuals approximate a normal distribution. Although minor deviations are observed at the tails, they are not substantial enough to raise serious concerns regarding the normality assumption.

3. Residuals vs Leverage Plot

The plot shows that most points exhibit relatively low leverage, and no points appear to be extreme outliers. This suggests that no individual observation is disproportionately influencing the model’s results.

4. Cook's Distance Plot

In this plot, none of the observations have Cook's distance values greater than 0.5, implying that no highly influential points are present in the dataset. Consequently, the regression model is not unduly affected by outliers or influential data points.

Overall, the regression model appears to be well-specified, and the assumptions of linear regression are reasonably satisfied. Further adjustments to the model may be unnecessary unless future data indicates otherwise.

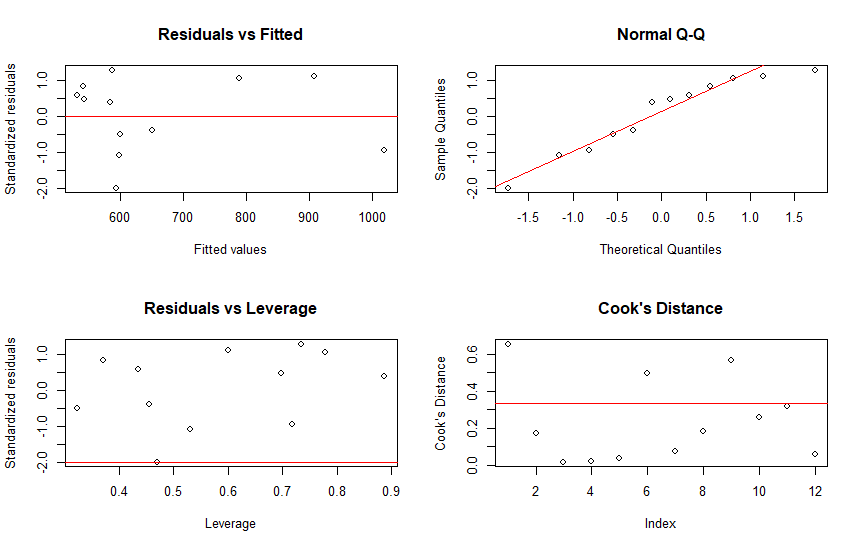


Figure 4.4

Perhaps we can use more flexible and intelligent regression methods, such as deep learning (neural networks). Nonlinear fitting with the sigmoid function might yield more accurate weights and prediction precision. However, due to the small sample size in this case, the deep learning packages provided by PyTorch would easily overfit, causing the fitting results to lose practical significance. In this task, we attempted this method, but we abandoned the results due to overfitting eventually.