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.

```r
#import data
data <- read.csv("data_task1.csv")
placebo <- data$placebo
interve <- data$intervention
#Shapiro-Wilk test
shap_pla <- shapiro.test(placebo)
shap_int <- shapiro.test(interve)
print(shap_pla)
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.

```r
library(ggplot2)
ggplot(data, aes(sample = placebo)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(title = "QQ plot for placebo",
       x = "norm Quantiles",
       y = "Sample Quantiles") +
  theme_minimal()

```

```
ggplot(data, aes(sample = interve)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(title = "QQ plot for intervention",
       x = "norm Quantiles",
       y = "Sample Quantiles") +
  theme_minimal()
```

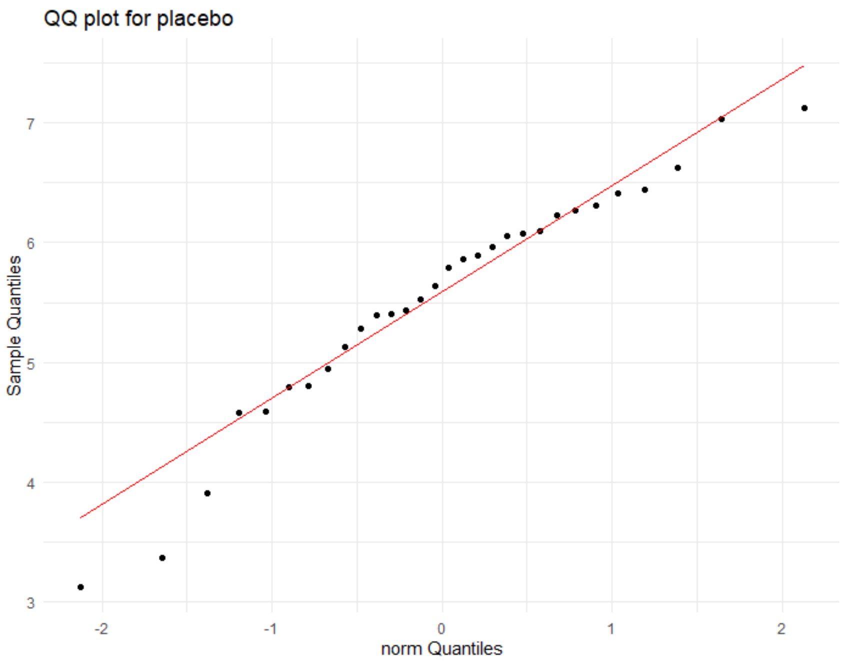

Figure1.1

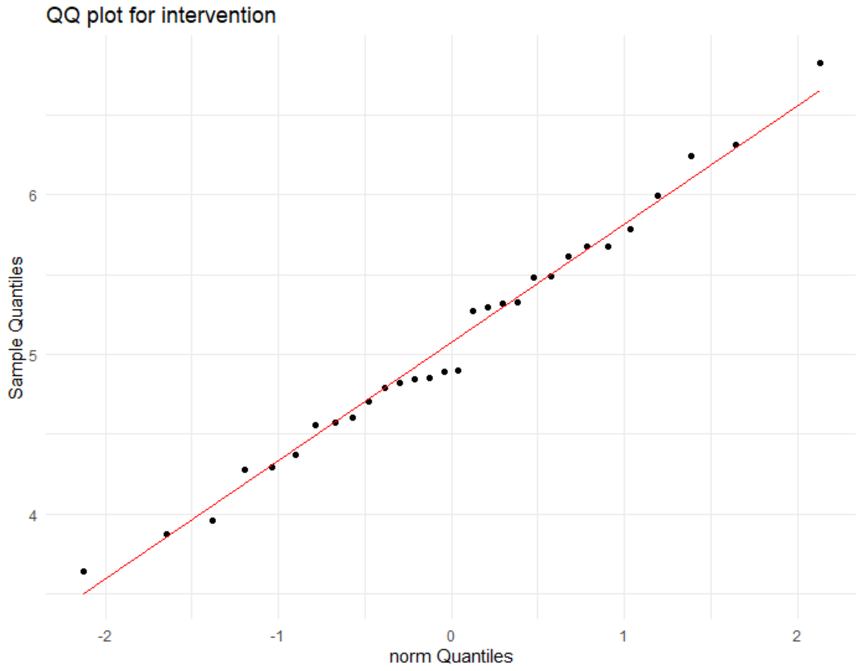

Figure1.2

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.

```
library(pwr)
# Assume Cohen's d = 0.5, alpha = 0.05
effect_size <- 0.5
alpha <- 0.05
sample_sizes <- seq(10, 200, by = 10)
powers <- sapply(sample_sizes, function(n) {
  pwr.t.test(n = n, d = effect_size, sig.level = alpha, type = "two.sample", alternative =
"two.sided")$power})
plot(sample_sizes, powers, type = "b", xlab = "Sample Size (per group)", ylab = "Power",
     main = "Power vs. Sample Size")
```

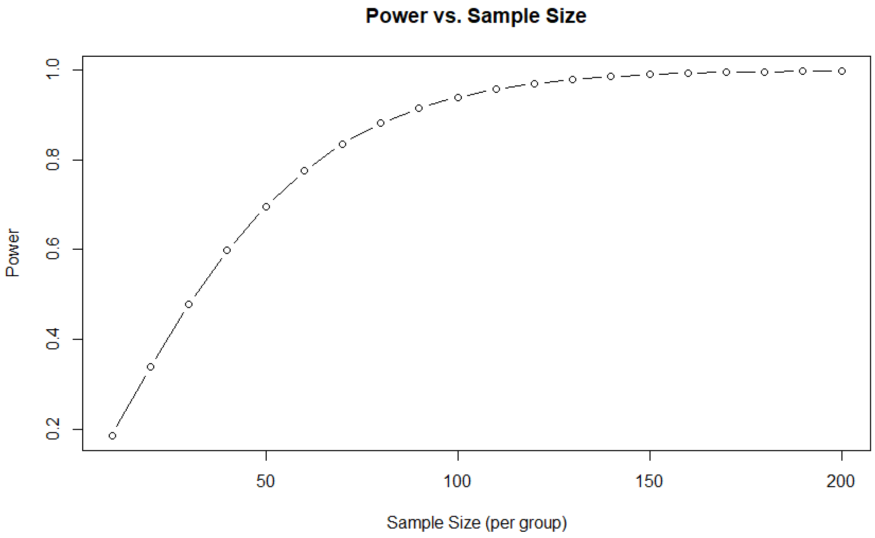

Figure2.1

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))
```

```r
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()
```

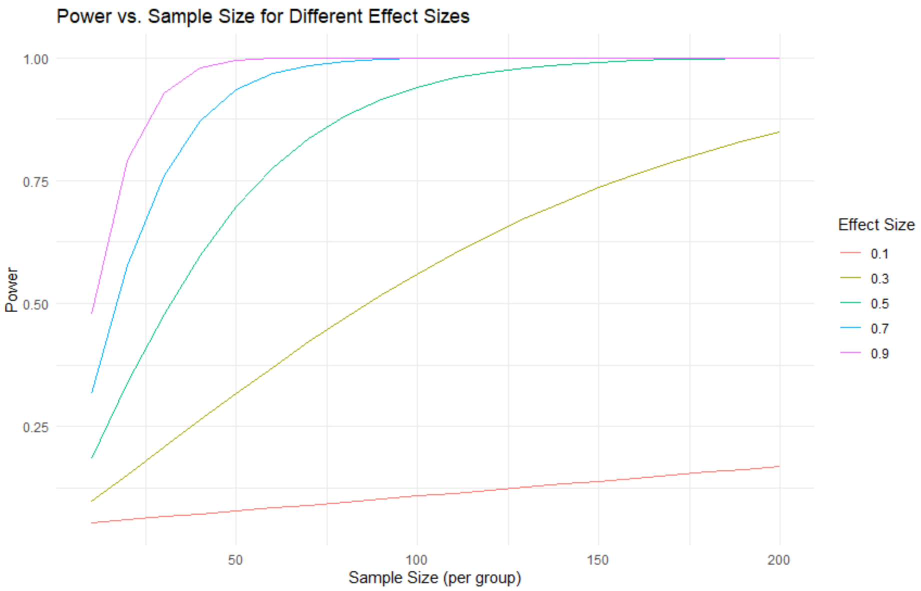

Figure2.2

Task 3

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

Null Hypothesis (H_0): There are no differences between the means of the individual groups in the population.

Alternative Hypothesis (H_1): At least two group means differ from each other in the population.

```r
data <- read.csv("data_task3_crp.csv")
library(tidyr)
data_long <- gather(data, key = "group", value = "crp_value", crp_placebo,
crp_intervention_1, crp_intervention_2)
anova_result <- aov(crp_value ~ group, data = data_long)
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.

```r
# read file
data <- read.csv("conctimedata_reduced.csv")
# extract essential data
data_subset <- data[, c("ID", "TIME", "DV")]
data_subset$ID <- as.factor(data_subset$ID)
data_subset$ID <- factor(data_subset$ID, labels = paste0("Patient", 1:12))
patient_colors <- setNames(rainbow(12), paste0("Patient", 1:12))
# draw DV-TIME curve
p <- ggplot(data_subset, aes(x = TIME, y = DV, group = ID, color = ID)) +
  geom_line(size = 0.5) +
  labs(title = "Concentration-Time Profiles for 12 Patients",
       x = "Time (hours)", y = "Concentration (mg/L)") +
  scale_color_manual(values = patient_colors,
                     labels = paste0("Patient", 1:12)) +
  theme_minimal() +
  theme(legend.title = element_blank(),
        plot.title = element_text(size = 12, face="bold", hjust = 0.5))
# save and adjust windows width & height
ggsave("concentration_profiles.jpg", plot = p,
       width = 1600/300, height = 1200/300, units = "in", dpi = 300)
print(p)
# draw boxplot
p_box <- ggplot(data_subset, aes(x = ID, y = DV, fill = ID)) +
  geom_boxplot() +
  labs(title = "Concentration Distribution by Patient",
       x = "Patient ID",
       y = "Concentration (mg/L)") +
  theme_minimal() +
  theme(legend.position = "none",
        plot.title = element_text(size = 12, face="bold", hjust = 0.5))
# save and adjust windows width & height
ggsave("concentration_boxplot.jpg", plot = p_box,
       width = 1600/300, height = 1200/300, units = "in", dpi = 300)
print(p_box)
```

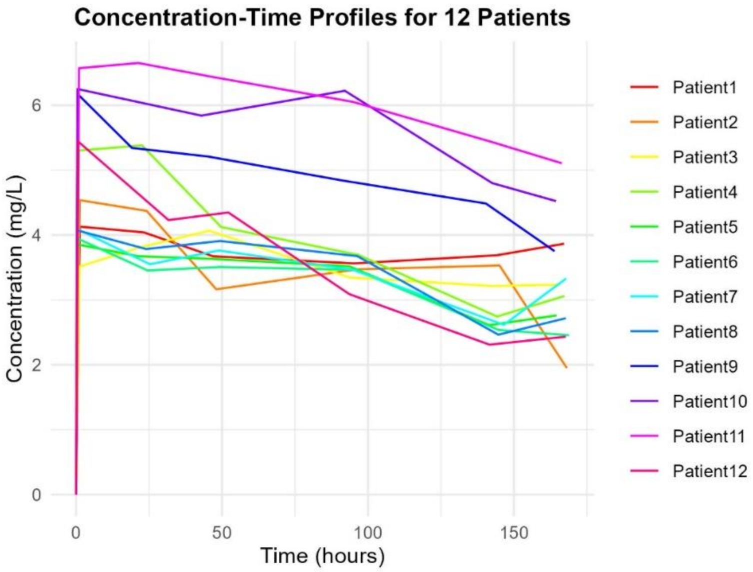

Figure4.1

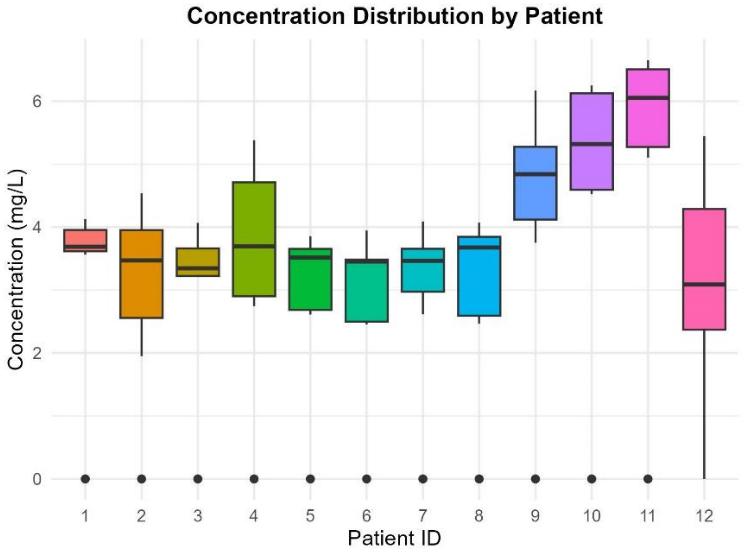

Figure4.2

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, (0,0) points are discarded to reduce the sharp fluctuations of the curve. The cubic spline interpolation $S(x)$ satisfies,

$$\boldsymbol{S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3}$$

The condition at its nodes satisfies,

$$\boldsymbol{S_i(x_i) = y_i \quad S_i(x_{i+1}) = y_{i+1}}$$

$$\boldsymbol{S_i(x_{i+1}) = S_{i+1}(x_{i+1})}$$

$$\boldsymbol{S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})}$$

$$\boldsymbol{S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})}$$

```r
# Cubic Spline Interpolation
# create a new container to save data
interpolated_data <- data.frame(TIME = numeric(), DV = numeric(), ID = character())
# Cubic Spline Interpolation
for (id in levels(data_subset$ID)) {
  subset_data <- data_subset[data_subset$ID == id, ]
  subset_data <- subset_data[!(subset_data$TIME == 0 & subset_data$DV == 0), ]
  interpolated <- spline(subset_data$TIME, subset_data$DV, xout = seq(min(subset_data$TIME), max(subset_data$TIME), length.out = 100))
  interpolated_data <- rbind(interpolated_data, data.frame(TIME = interpolated$x, DV = interpolated$y, ID = id))
}
interpolated_data$ID <- factor(interpolated_data$ID, levels = levels(data_subset$ID))
# draw a new curve
p_interpolated <- ggplot(interpolated_data, aes(x = TIME, y = DV, group = ID, color = factor(ID))) +
  geom_line(size = 0.5) +
  labs(title = "Interpolated Concentration-Time Profiles for 12 Patients",
       x = "Time (hours)", y = "Concentration (mg/L)") +
  scale_color_manual(values = patient_colors,
                     labels = paste0("Patient", 1:12)) +
  theme_minimal() +
  theme(legend.title = element_blank(),
        plot.title = element_text(size = 12, face="bold", hjust = 0.5)) +
  coord_cartesian(ylim = c(0, NA))

ggsave("interpolated_concentration_profiles.jpg", plot = p_interpolated,
       width = 1600/300, height = 1200/300, units = "in", dpi = 300)
print(p_interpolated)
```

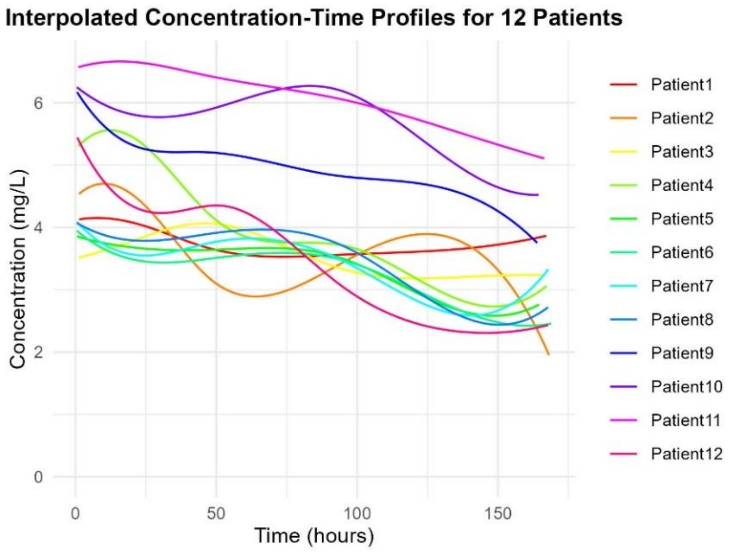

Figure4.3

The curve in Figure 4.3 is smoother than in Figure 4.1. Then, perform integration over $x \in [1, 166]$ to obtain the result, and restructure to obtain a new dataset as shown below.

```r
# initialize the container for storing integral results
integrated_results <- data.frame(ID = character(), Integral = numeric(), stringsAsFactors = FALSE)
lower_bound <- 1
upper_bound <- 166
# Integral after interpolation
for (id in levels(interpolated_data$ID)) {
  subset_data <- interpolated_data[interpolated_data$ID == id, ]
  if (nrow(subset_data) > 1) {
    # create cubic spline interpolation function
    spline_function <- splinefun(subset_data$TIME, subset_data$DV)
    # perform the integral over the interval [1, 166]
    integral_value <- integrate(spline_function, lower = lower_bound, upper = upper_bound)$value
    integrated_results <- rbind(integrated_results, data.frame(ID = id, Integral = integral_value))
  } else {
    message(paste("ID", id, "error"))
  }
}
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,

```r
# read file
data <- read.csv("data_task4.csv", sep = ",")
# multiple linear regression
model <- lm(AUC ~ WGT + BSA + AGE + HGT + DOSE + GFR, data = data)
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.

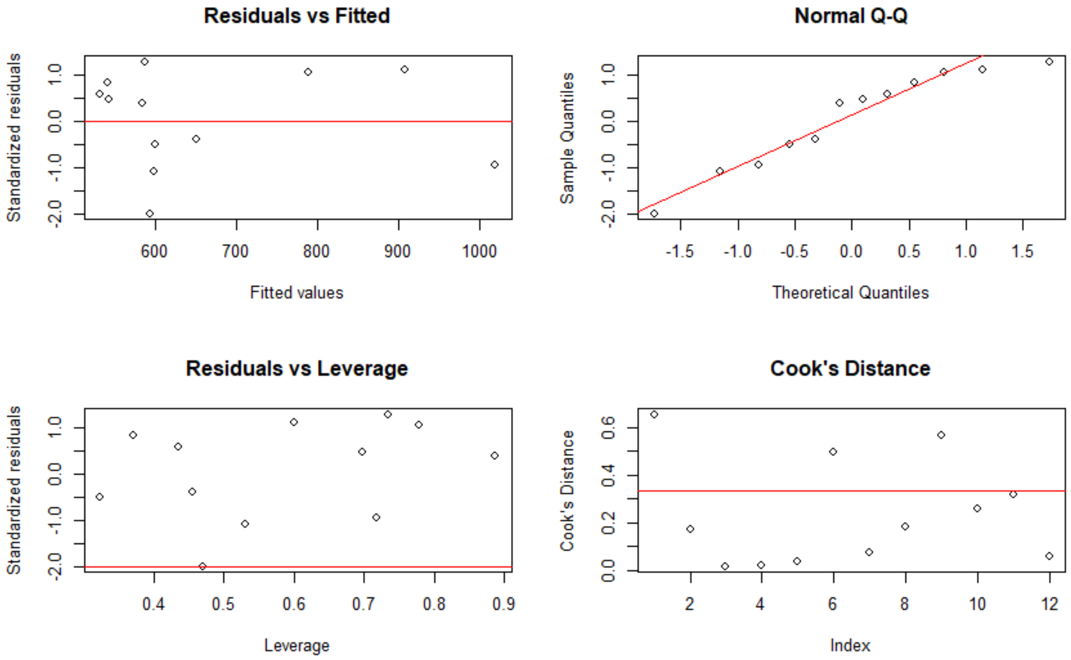

Figure 4.4

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.