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.

```
#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.

```
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()
```

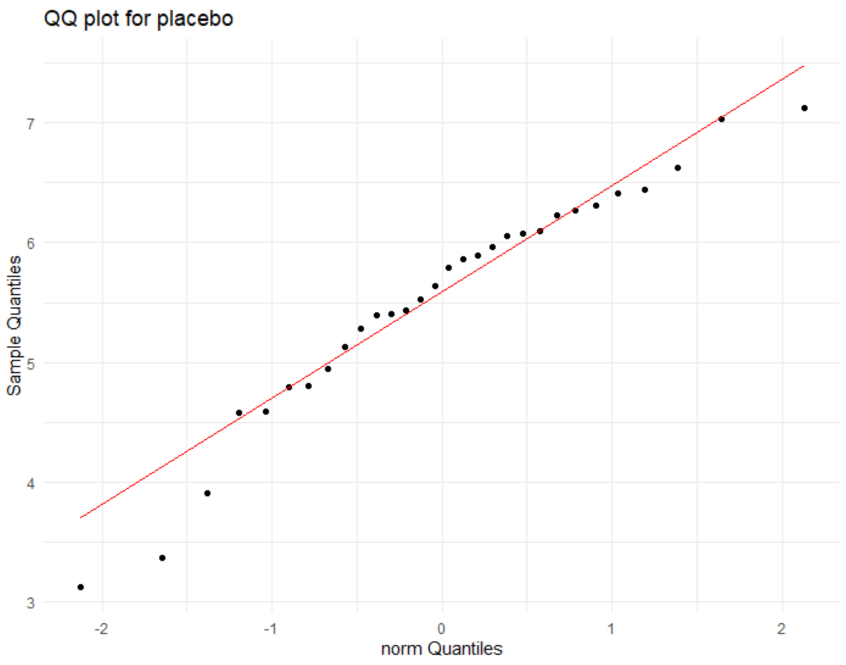

Figure1.1

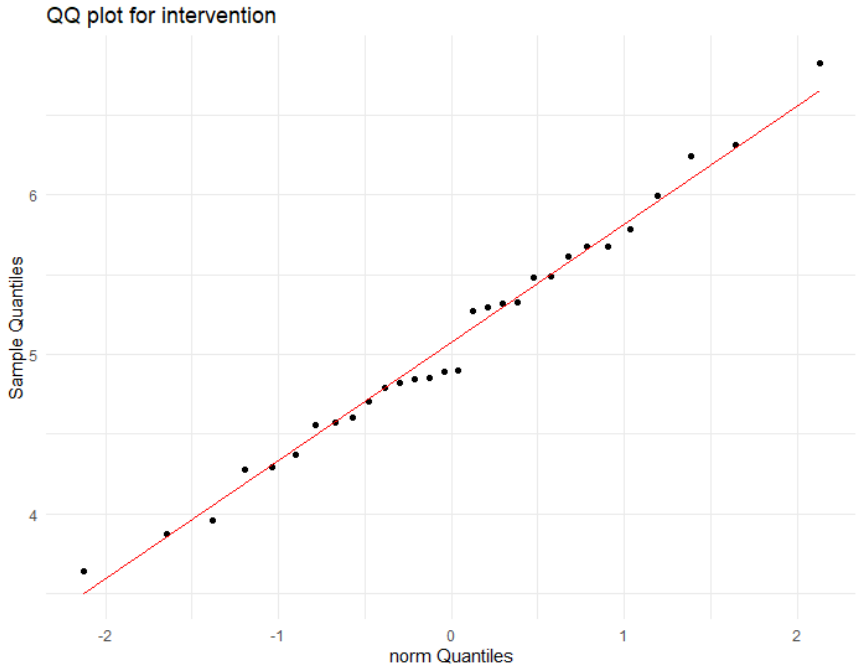

Figure1.2

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.

```r
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")
```

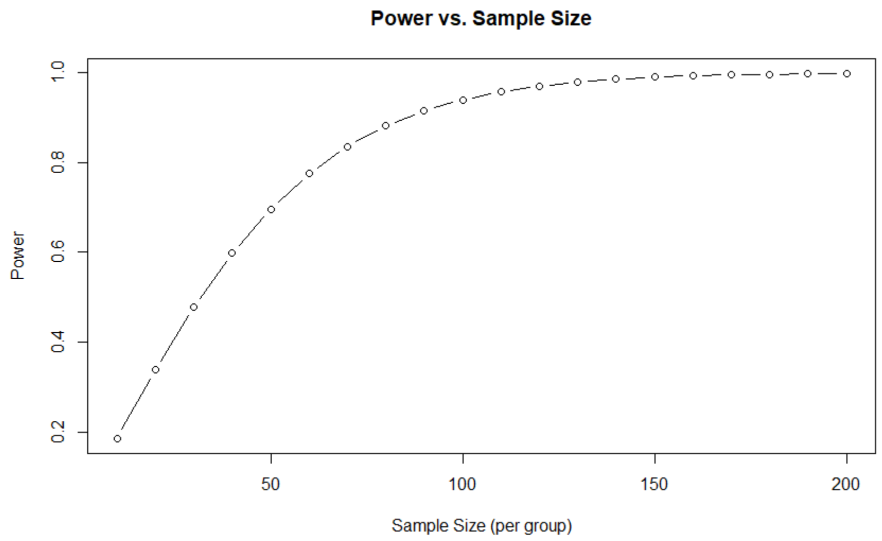

Figure2.1

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.

```r
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()
```

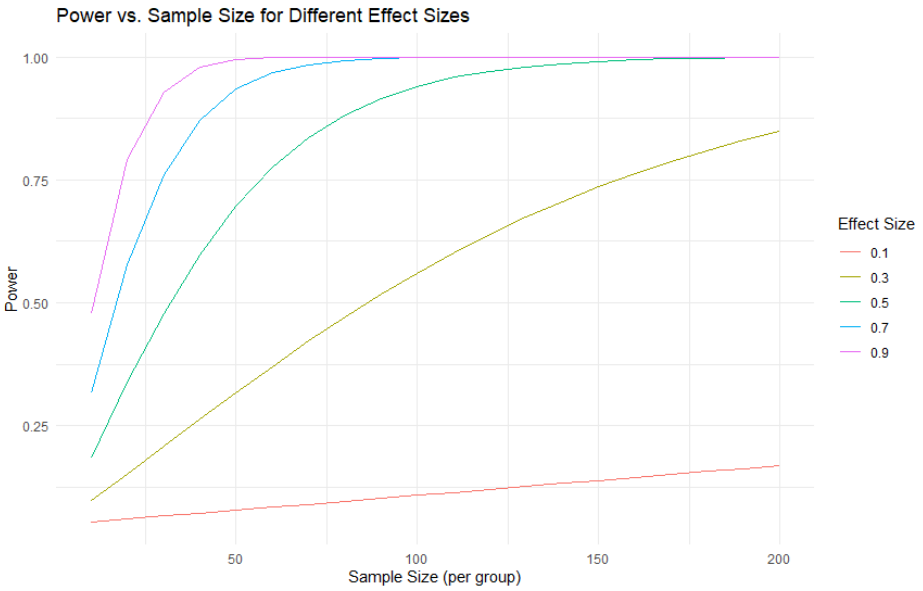

Figure2.2

Task 3

Just like the Task1, we should first test whether these samples are normally distributed by using Shapiro-Wilk test.

```
data <- read.csv("data_task3_crp.csv")
plac <- data$crp_placebo
int1 <- data$crp_intervention_1
int2 <- data$crp_intervention_2
#Shapiro-Wilk test
shap_plac <- shapiro.test(plac)
shap_int1 <- shapiro.test(int1)
shap_int2 <- shapiro.test(int2)
print(shap_plac)
print(shap_int1)
print(shap_int2)
```

```
Output:
> print(shap_plac)
 Shapiro-Wilk normality test
data:  plac
W = 0.84466, p-value = 0.0004803
> print(shap_int1)
```

```
Shapiro-Wilk normality test
data:  int1
W = 0.74996, p-value = 9.092e-06
> print(shap_int2)
Shapiro-Wilk normality test
data:  int2
W = 0.71482, p-value = 2.583e-06
```

With the result of the Shapiro-Wilk test, we can say that none of the samples is normally distributed. So we cannot use the ANOVA test, which assumes the samples are normally distributed. Instead, we use the Kruskal-Wallis H test to detect whether there is a significant difference among the three samples.

```
# reform data
crp_data <- data.frame(
  group = factor(rep(c("Placebo", "Intervention_1", "Intervention_2"), each = 30)),
  crp = c(plac, int1, int2)
)

# Kruskal-Wallis H test
kruskal.test(crp ~ group, data = crp_data)
```

```
Output:
Kruskal-Wallis rank sum test
data:  crp by group
Kruskal-Wallis chi-squared = 9.9986, df = 2, p-value = 0.006743
```

Because the p-value is lower than 0.05, it indicates a significant difference among the three samples. Hence, we performed pairwise comparisons using the Mann-Whitney U test to identify which groups differ significantly.

```
pairwise.wilcox.test(crp_data$crp, crp_data$group, p.adjust.method = "bonferroni")
```

```
Output:
Pairwise comparisons using Wilcoxon rank sum exact test


data:  crp_data$crp and crp_data$group


                    Intervention_1    Intervention_2
Intervention_2      0.7254                  -
Placebo             0.2718                0.0026


P value adjustment method: bonferroni
```

We analyzed the results and concluded that there is a significant difference between the Placebo and Intervention_2 groups (p = 0.0026). No significant differences were observed between the Placebo and Intervention_1 groups, nor between the Intervention_1 and Intervention_2 groups after applying the Bonferroni correction. Furthermore, since the mean CRP value in the Intervention_2 group is lower than in the Placebo group, we conclude that the drug KTH002 has a measurable positive effect by reducing inflammation.

```
> mean(plac)
[1] 72.5312
> mean(int2)
[1] 24.84123
```

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.

```
# 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)
```

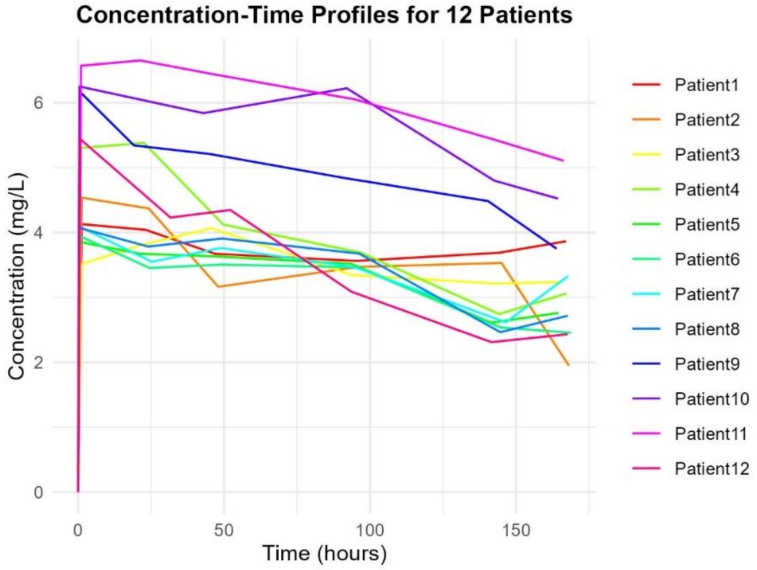

Figure4.1

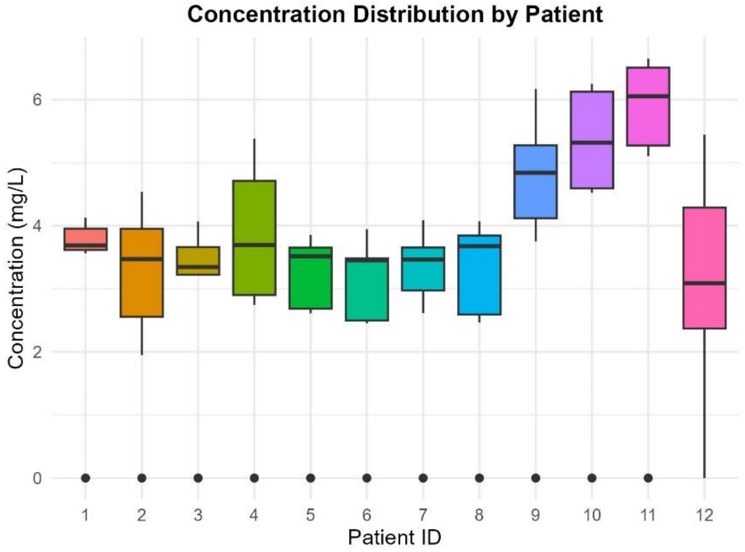

Figure4.2

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,

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

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

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

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

```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)
```

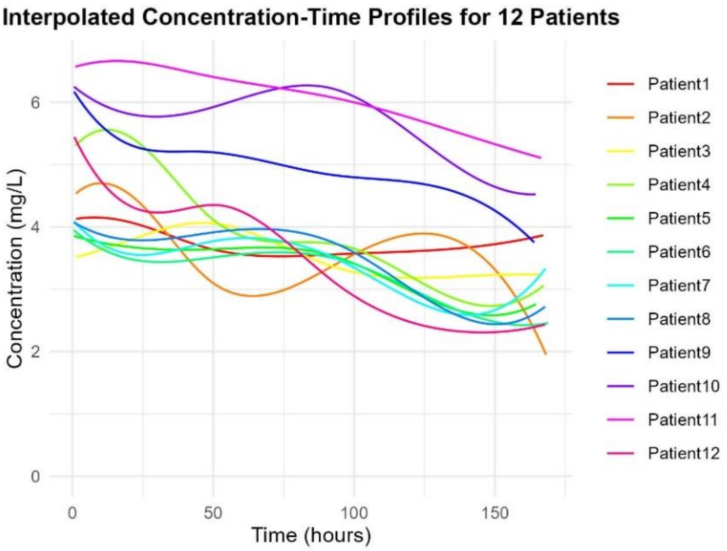

Figure4.3

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.

```
# 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,

```
# 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.

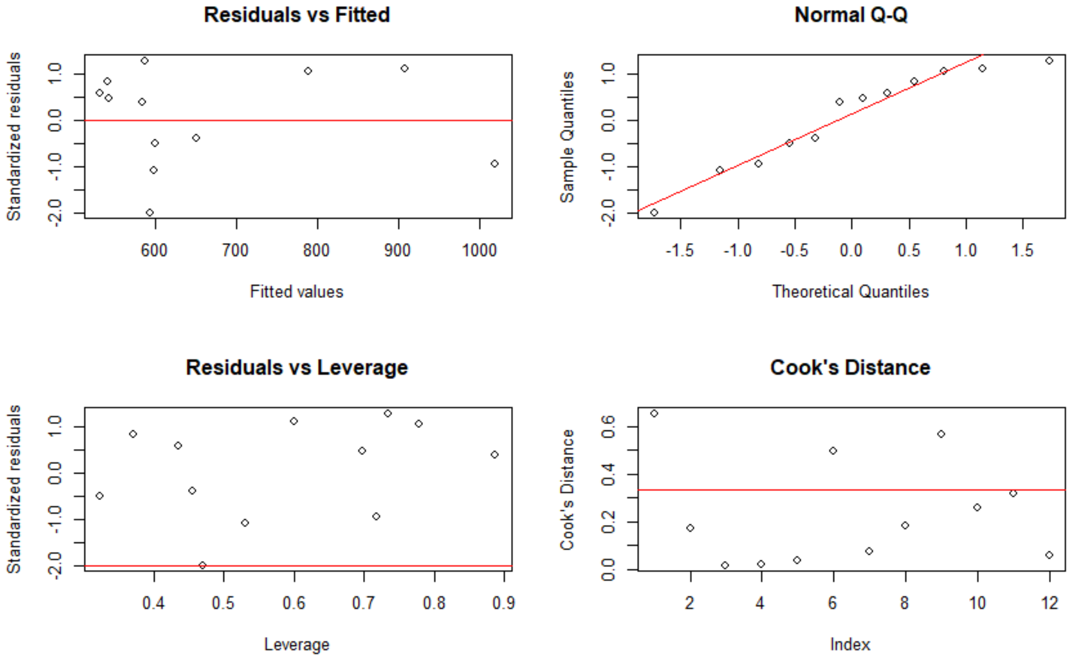

Figure 4.4

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.

The residuals are scattered around the horizontal line at zero, which indicates that the model does not show significant signs of non-linearity. However, we can see a slight deviation at the higher fitted values (above 900), suggesting a potential issue. But the residuals range between approximately -1.5 and 1.2, meaning heteroscedasticity is not a major concern in this model.

2. Normal Q-Q Plot

A Q-Q plot (quantile-quantile plot) is a graphical tool used to compare the distribution of a dataset with a theoretical distribution, usually a normal distribution. It plots the quantiles of the sample data against the quantiles of the theoretical distribution. If the data follows the theoretical distribution closely, the points will approximately form a straight line along the 45-degree reference line. Deviations from this line indicate departures from the theoretical distribution, such as skewness or heavy tails.

The points mostly fall along the 45-degree reference line, which indicates that the residuals follow an approximately normal distribution. There is a slight deviation at both ends, especially below -1.5 and above 1.5 on the theoretical quantile axis. These deviations suggest that some of the extreme values are slightly off from normality, but overall the residuals are close to normal.

3. Residuals vs Leverage Plot

Leverage[1] measures how much influence a data point has on the regression model, particularly in terms of how far the point is from the center of the predictor variables. Leverage values typically range between 0 and 1, high-leverage points have no neighboring points in R^p space, where p is the number of independent variables in a regression model. This makes the fitted model likely to pass close to a high leverage observation[2]. So we can consider that higher leverage means bigger influence.

The plot shows that most points exhibit relatively low leverage, and no points appear to be extreme outliers and most of the residuals fluctuate around 0. This suggests that no individual observation is disproportionately influencing the model's results. The leverage values range from approximately 0.45 to 0.9. While most residuals are between -1.5 and 1, a few points have high leverage values, especially those beyond 0.8. This suggests that these points have a higher influence on the model. However, the absence of a distinct pattern and the moderate residuals indicate that there are no extreme outliers in the data.

4. Cook's distance Plot

Cook's distance[3] is used to evaluate the influence of a data point when performing a least-squares regression analysis. One observation's Cook's distance is defined as the sum of all the changes in the model when this observation is removed from the model[4]. If a data point has a large Cook's distance, it indicates that removing the point would significantly alter the regression model's estimates, making it an influential point.

In this plot, just a few of the observations have Cook's distance values greater than the threshold, which is set at 4 divided by the sample size, 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. In addition, none of the points exceed the critical threshold of 0.5 (red line), suggesting no individual observation has an undue influence on the regression model. The Cook's distance values range from 0 to approximately 0.5, which is below the threshold of 1 typically used to flag influential observations.

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.

Additionally, we can use a correlation matrix to explore the linear relationships between pairs of variables. A correlation matrix is a symmetric matrix used to display the correlation coefficients between multiple variables. The correlation coefficient measures the linear relationship between two variables, typically ranging from -1 to 1. Each element in the correlation matrix represents the correlation between two variables, with the Pearson correlation coefficient being a commonly used measure. The closer the absolute value is to 1, the stronger the linear relationship. As shown in the 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.

References

[1] 'Leverage (statistics)', *Wikipedia*. Oct. 04, 2024. Accessed: Oct. 11, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Leverage_(statistics)&oldid=1249350472#cite_note-1

[2] H. Joe, 'Everitt BS 2002: The Cambridge dictionary of statistics, Cambridge: Cambridge University Press. 420 pp.\pounds 30.00 (US $50.00)(HB). ISBN 05218 1099 X.' 2004.

[3] R. D. Cook, 'Detection of Influential Observation in Linear Regression', *Technometrics*, vol. 19, no. 1, pp. 15–18, Feb. 1977, doi: 10.1080/00401706.1977.10489493.

[4] 'Cook's Distance - MATLAB & Simulink - MathWorks 中国'. Accessed: Oct. 11, 2024. [Online]. Available: https://ww2.mathworks.cn/help/stats/cooks-distance.html