Question 1

(1) For the group with RUNX1=0, the distribution of AUC values is summarized in Table1.1, with Minimum AUC equals to 0.0, 1st Quartile (25th percentile) equals to 164.5, Median (50th percentile) equals to 206.9, Mean equals to 198.9, 3rd Quartile (75th percentile) equals to 243.3, Maximum AUC equals to 286.3.

For the group with RUNX1=1, the distribution of AUC values is summarized in Table 1.2, with Minimum AUC equals to 38.37, 1st Quartile (25th percentile) equals to 194.29, Median (50th percentile) equals to 222.59, Mean equals to 211.37, 3rd Quartile (75th percentile) equals to 250.36, Maximum AUC equals to 271.59.

Table1.1 Group with RUNX1=0 - AUC Distribution Summary

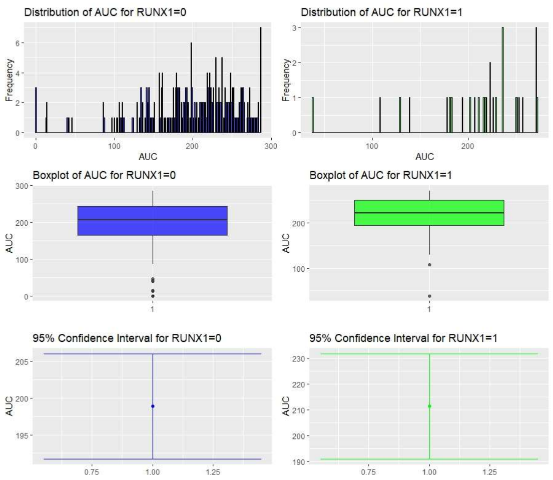
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0	164.5	206.9	198.9	243.3	286.3

Table1.2 Group with RUNX1=1 - AUC Distribution Summary

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
38.37	194.29	222.59	211.37	250.36	271.59

(2) With central limit and normal distribution assumptions, 95% Confidence Interval for Group with RUNX1=0 equals to 191.6842 206.0469, while 95% Confidence Interval for Group with RUNX1=1 equals to 191.0485 231.6917.

The box plot, histogram and 95% confidence interval of the distribution of AUC can be seen in Figure 1.



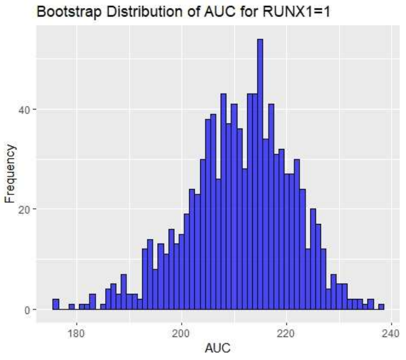


Figure 1 **Figure 2**

(3) Using the central limit theorem and the normal distribution requires the following assumptions or premises to be met: ①Observations should be independent of each other; ② The central limit theorem usually requires that the sample size is large enough to ensure that the distribution of the sample mean is close to normal distribution. When the sample size is greater than 30, the central limit theorem comes into effect. ③The variable should be continuous. Therefore, one problem is that, for RUNX1, the sample size is too small, only 29, which may affect the inference of the confidence interval. We can use Bootstrap to increase sample size(Figure 2).

Question 2

(1) Apply Pearson correlation to calculate the correlation coefficient between genes and AUC. Correct the P value using the Benjamini-Hochberg method to obtain the adjusted P value to reduce the false

discovery. With the threshold of adjusted P value less than 0.05, 2138 genes were found to have significant correlation coefficients with AUC. However, for the value of the correlation coefficient, the correlation between genes and AUC is not strong. The highest correlation coefficient is between ST6GALNAC3 and AUC, with a value of 0.41.

(2) Null Hypothesis (H0): There is no significant correlation between the expression of the gene and the drug response.

(3) When the p-value exceeds 0.2, the distribution can be approximated as uniform. Conversely, the distribution is more concentrated in regions where the p-value is relatively small (P < 0.1). Therefore, a left-skewed distribution with a concentration of low p-values suggests a higher prevalence of significant gene-drug associations.Notably, after adjustment, the frequency of adjusted p-values in the lower range (P < 0.1) is lower than the unadjusted p-values. This suggests that the Benjamini-Hochberg method has influenced the distribution, leading to a more conservative estimation of significant gene-drug associations(Figure3 & Figure4).

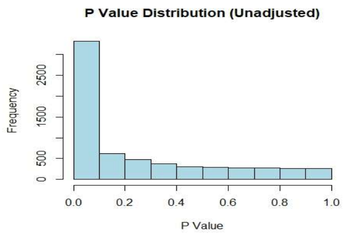


Figure 3

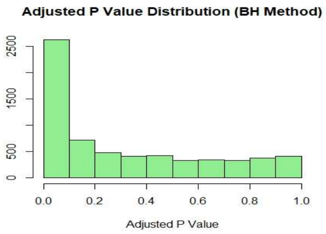
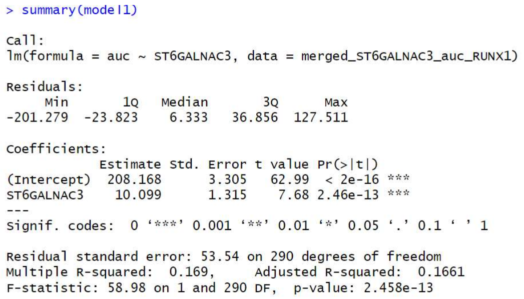


Figure 4

(4) In a linear regression model of genes ST6GALNAC3 and RUNX1 on AUC, the function is **AUC** = 10.040***ST6GALNAC3**+4.921***RUNX1**+207.633. Specific statistical data are in Figure 6.

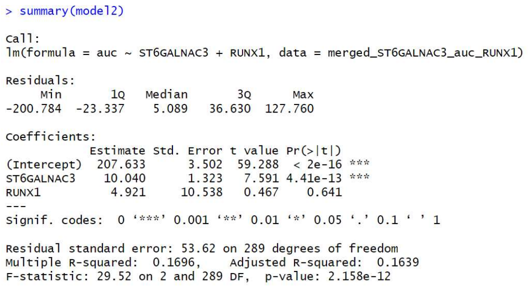
To explore the impact of RUNX1, another linear regression model containing only gene ST6GALNAC3 and AUC was also included. Figure 5 indicates a significant positive correlation between ST6GALNAC3 and AUC (coefficient = 10.099, p-value < 0.001). The high F-statistic and its associated low p-value suggest overall model significance. Figure 6 indicates that the introduction of RUNX1 neither shows a significant association with AUC (p-value > 0.05) nor affects the estimate of ST6GALNAC3. Consider that one of the requirements to become a confounder is an association between the outcome and the confounder. Therefore, according to the current sample data, **RUNX1 cannot be considered a confounder for ST6GALNAC3 or other genes**. (Model solely considering RUNX1 reveals a non-significant association with AUC (p-value > 0.05) and low R-squared value.)

However, considering that the sample size of RUNX1=1 is small, a more accurate study requires a larger sample.



```
> summary(model1)

Call:
lm(formula = auc ~ ST6GALNAC3, data = merged_ST6GALNAC3_auc_RUNX1)

Residuals:
     Min       1Q   Median       3Q      Max 
-201.279  -23.823    6.333   36.856  127.511 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  208.168      3.305   62.99  < 2e-16 ***
ST6GALNAC3    10.099      1.315    7.68 2.46e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.54 on 290 degrees of freedom
Multiple R-squared:  0.169,	Adjusted R-squared:  0.1661 
F-statistic: 58.98 on 1 and 290 DF,  p-value: 2.458e-13
```

Figure 5



```
> summary(model2)

Call:
lm(formula = auc ~ ST6GALNAC3 + RUNX1, data = merged_ST6GALNAC3_auc_RUNX1)

Residuals:
     Min       1Q   Median       3Q      Max 
-200.784  -23.337    5.089   36.630  127.760 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  207.633      3.502  59.288  < 2e-16 ***
ST6GALNAC3    10.040      1.323   7.591 4.41e-13 ***
RUNX1          4.921     10.538   0.467    0.641    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 53.62 on 289 degrees of freedom
Multiple R-squared:  0.1696,	Adjusted R-squared:  0.1639 
F-statistic: 29.52 on 2 and 289 DF,  p-value: 2.158e-12
```

Figure 6

Question 3

(1) Employ hierarchical clustering via ward.D2 and the Euclidean distance to measure the similarities among groups. Ward.D2 is a linkage method of clustering algorithm for calculating the distance between two clusters. It determines whether to merge two clusters based on an increase in variance. The method aims to minimize the total variance within the merged clusters. Ward.D2 method is commonly used in hierarchical clustering to obtain more balanced and compact clusters. Euclidean distance is a method of calculating the straight-line distance between two points.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Select clusters at level 5 of the dendrogram. The resulting clusters and their corresponding sample counts are as follows: Cluster 1: 74 samples; Cluster 2: 70 samples; Cluster 3: 71 samples; Cluster 4: 65 samples; Cluster 5: 12 samples(Figure 7).

(2) The RUNX1 mutation status is related to the clustering status to a certain extent, and the samples with RUNX1 of 1 are those labelled in orange below the column clusters in Figure 8, which are mainly distributed within 2 of the 5 clusters, and many of them are clustered close to each other.

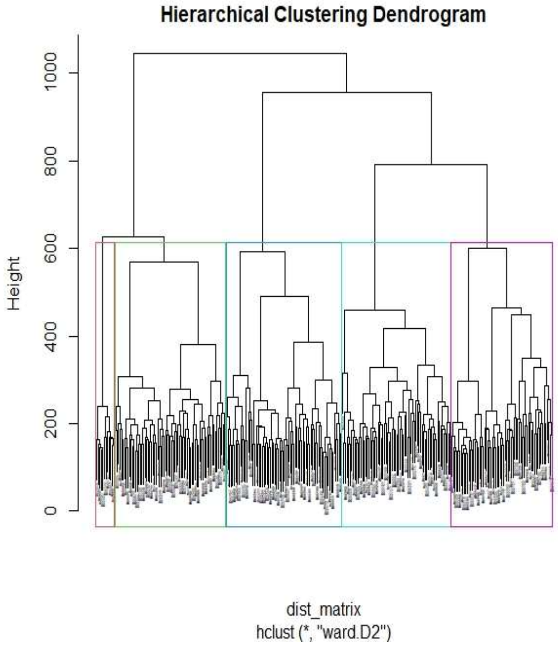


Figure 7

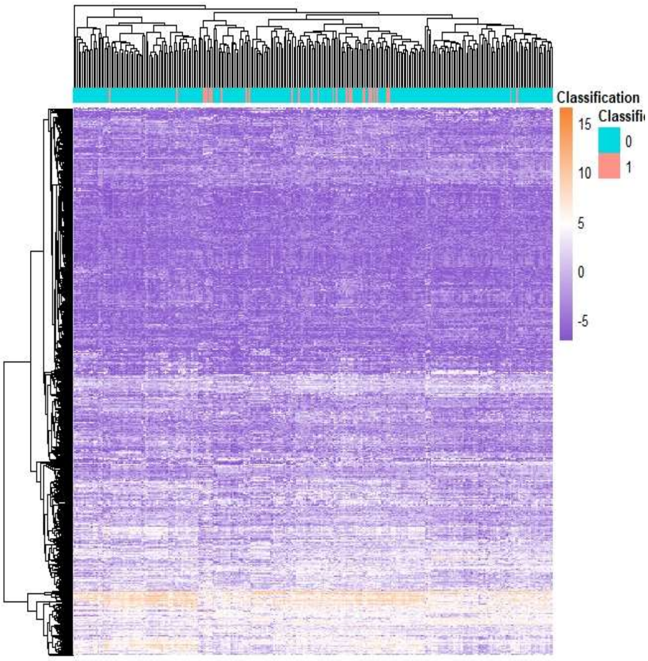
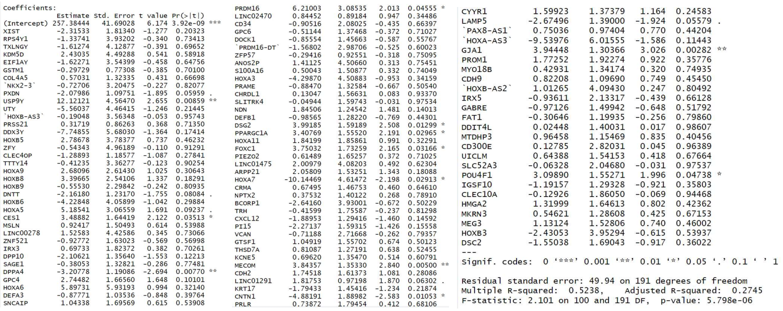


Figure 8

Question 4

(1) Select the 100 genes with the highest variance as covariates and construct a linear regression model to predict the AUC(Figure 9).



```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept) 257.38444   41.69028   6.174 3.92e-09 ***
XIST         -2.31533    1.81340  -1.277  0.20323    
RPS4Y1       -1.33741    3.93202  -0.340  0.73413    
TXLNGY       -1.61274    4.12877  -0.391  0.69652    
KDM5D         2.43035    4.49288   0.541  0.58918    
EIF1AY       -1.62271    3.54399  -0.458  0.64756    
GSTM1        -0.29729    0.77308  -0.385  0.70100    
COL4A5        0.57031    1.32335   0.431  0.66698    
`NKX2-3`     -0.72706    3.20475  -0.227  0.82077    
PXDN         -2.07986    1.09751  -1.895  0.05959 .  
USP9Y        12.12121    4.56470   2.655  0.00859 ** 
UTY          -5.56037    4.46415  -1.246  0.21445    
`HOXB-AS3`   -0.19048    3.56348  -0.053  0.95743    
PRSS21        0.31719    0.86263   0.368  0.71350    
DDX3Y        -7.74855    5.68030  -1.364  0.17414    
HOXB5         2.78678    3.78377   0.737  0.46232    
ZFY          -0.54343    4.96189  -0.110  0.91291    
CLEC4OP      -1.28893    1.18577  -1.087  0.27841    
TTTY14       -0.41235    3.36277  -0.123  0.90254    
HOXA9         2.68096    2.61430   1.025  0.30643    
HOXB8         3.39665    2.54106   1.337  0.18291    
HOXB9        -0.55530    2.29842  -0.242  0.80935    
DNTT         -2.16180    1.23170  -1.755  0.08084 .  
HOXB6        -4.22848    4.05899  -1.042  0.29884    
HOXA5         5.18541    3.06559   1.691  0.09237 .  
CES1          3.48882    1.64419   2.122  0.03513 *  
MSLN          0.92417    1.50493   0.614  0.53988    
LINC00278     1.52583    4.42586   0.345  0.73066    
ZNF521       -0.92772    1.63023  -0.569  0.56998    
IRX3          0.69733    1.82372   0.382  0.70261    
DPP10        -2.10621    1.35640  -1.553  0.12213    
SAGE1        -0.38053    1.32821  -0.286  0.77481    
DPPA4        -3.20778    1.19086  -2.694  0.00770 ** 
GPC4          2.74482    1.66560   1.648  0.10101    
HOXA6         5.89731    5.93193   0.994  0.32140    
DEFA3        -0.87771    1.03536  -0.848  0.39764    
SNCAIP        1.04338    1.69569   0.615  0.53908    

PRDM16         6.21003    3.08535   2.013  0.04555 *  
LINC02470      0.84452    0.89184   0.947  0.34486    
CD34          -0.90516    2.08025  -0.435  0.66397    
GPC6          -0.51144    1.37468  -0.372  0.71027    
DOCK1         -0.85554    1.45663  -0.587  0.55767    
`PRDM16-DT`   -1.56802    2.98706  -0.525  0.60023    
ZFP57         -0.29416    0.92551  -0.318  0.75095    
ANOS2P         1.41125    4.50660   0.313  0.75451    
S100A16        0.50043    1.50877   0.332  0.74049    
HOXA3         -4.29870    4.50883  -0.953  0.34159    
PRAME         -0.88470    1.32584  -0.667  0.50540    
CHRDL1         0.13047    1.56631   0.083  0.93370    
SLITRK4       -0.04944    1.59743  -0.031  0.97534    
NDN            1.84506    1.24542   1.481  0.14013    
DEFB1         -0.98565    1.28220  -0.769  0.44301    
DSG2           3.99185    1.59189   2.508  0.01299 *  
PPARGC1A       3.40769    1.55520   2.191  0.02965 *  
HOXA11         1.84199    1.85861   0.991  0.32291    
FOXC1          3.75032    1.73259   2.165  0.03166 *  
PIEZO2         0.61489    1.65257   0.372  0.71025    
LINC01475      2.00979    4.08203   0.492  0.62304    
ARPP21         2.05809    1.53251   1.343  0.18088    
HOXA7        -10.14469    4.61472  -2.198  0.02913 *  
CRMA           0.67495    1.46753   0.460  0.64610    
NPTX2          0.37532    1.40122   0.268  0.78910    
BCORP1        -2.64160    3.93001  -0.672  0.50229    
TRH           -0.41599    1.75587  -0.237  0.81298    
CXCL12        -1.88953    1.29416  -1.460  0.14592    
PI15          -2.27137    1.59315  -1.426  0.15558    
VCAN          -0.71188    2.71668  -0.262  0.79357    
GTSF1          1.04919    1.55702   0.674  0.50123    
THSD7A         0.81087    1.27191   0.638  0.52455    
KCNE5          0.69620    1.35470   0.514  0.60791    
MECOM          3.84357    1.35330   2.840  0.00500 ** 
CDH2           1.74518    1.61373   1.081  0.28086    
LINC01291      1.81753    0.97198   1.870  0.06302 .  
KRT17         -1.79433    1.45416  -1.234  0.21874    
CNTN1         -4.88191    1.88982  -2.583  0.01053 *  
PRLR           0.73872    1.79454   0.412  0.68106    

CYYR1          1.59923    1.37379   1.164  0.24583    
LAMP5         -2.67496    1.39000  -1.924  0.05579 .  
`PAX8-AS1`     0.75036    0.97404   0.770  0.44204    
`HOXA-AS3`    -9.53976    6.01555  -1.586  0.11443    
GJA1           3.94448    1.30366   3.026  0.00282 ** 
PROM1          1.77252    1.92274   0.922  0.35776    
MYO18B         0.42931    1.34174   0.320  0.74935    
CDH9           0.82208    1.09690   0.749  0.45450    
`HOXB-AS2`     1.01265    4.09430   0.247  0.80492    
IRX5          -0.93611    2.13317  -0.439  0.66128    
GABRE         -0.97126    1.49942  -0.648  0.51792    
FAT1          -0.30646    1.19935  -0.256  0.79860    
DDIT4L         0.02448    1.40031   0.017  0.98607    
MTDHP3         0.96458    1.15469   0.835  0.40456    
CD300E         0.12785    2.82031   0.045  0.96389    
UICLM          0.64388    1.54153   0.418  0.67664    
SLC52A3       -0.06328    2.04680  -0.031  0.97537    
POU4F1         3.09890    1.55271   1.996  0.04738 *  
IGSF10        -1.19157    1.29328  -0.921  0.35803    
CLEC10A       -0.12926    1.86050  -0.069  0.94468    
HMGA2          1.31999    1.64613   0.802  0.42362    
MKRN3          0.54621    1.28608   0.425  0.67153    
MEG3           1.13124    1.52806   0.740  0.46002    
HOXB3         -2.43053    3.95294  -0.615  0.53937    
DSC2          -1.55038    1.69043  -0.917  0.36022    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 49.94 on 191 degrees of freedom
Multiple R-squared:  0.5238,	Adjusted R-squared:  0.2745 
F-statistic: 2.101 on 100 and 191 DF,  p-value: 5.798e-06
```

Figure 9

For individual BA2409, the observed AUC is 281.36 and the predicted AUC is 218.56. Figure 10 is the plot of the in-sample predictions against the observed values. Where the red line indicates a straight line where the predicted and observed values are equal, the scatter in the graph indicates that the predicted values of the regression equations are all more deviated in the interval of AUC observations in the range of 0-50. The regression equation is more accurate in the 100-300 AUC observations range, and the scatter points are more evenly distributed at both ends of the red line.

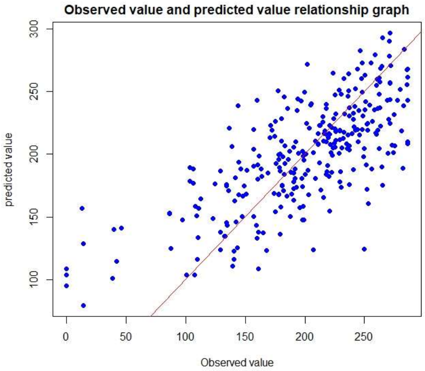


Figure 10

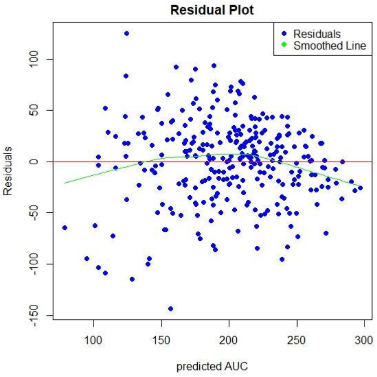


Figure 11

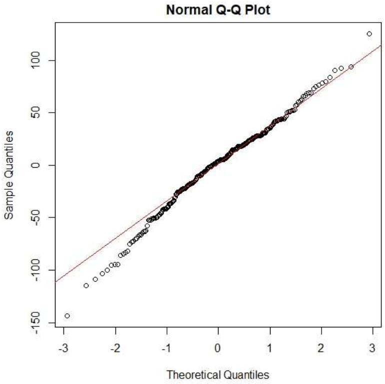


Figure 12

The assumptions are not fully satisfied. The first point is the linear relationship between the independent and dependent variables, which can be seen from the parameters of the regression equation (Figure 9) where the p-values of many of the terms covariates are much larger than 0.05 and are not significant. The second point is the normality of the residuals, which can be seen from the Q-Q plot (Figure 12), where the red straight line indicates the theoretical quantile of the normal distribution. Most points are near the red straight line, indicating that the data roughly satisfy the assumption that the residuals must conform to a normal distribution. The third point is homoscedasticity or the assumption that the random disturbance term is uncorrelated with the explanatory variables; ideally, the residuals should be randomly distributed around 0 as the predicted value increases, but the variance of the residuals on the left side of AUC100 in the residual plot (Figure 11) is greater than the variance of the residuals on the right side of AUC100. The green smoothed curve bends to indicate that some independent variables are not put into the model.

（2）The in-sample Correlation Coefficient is **0.7237513**, which is computed on the dataset used for model training, indicating that the model fits the training data well; the LOOCV Correlation Coefficient is **0.3403362**, which is obtained through a more rigorous assessment that excludes each sample from the training data during model construction and then uses the model to predict the excluded samples. The number shows that the model's performance on new, unseen data could be higher since the model is overly reliant on the specific features of the training data and needs to be generalised to other datasets.

The reason for the difference in the two correlation coefficients is Overfitting. The model learns noise in the training data or specific features of the data that are not generalised to other datasets.

For the two correlation coefficients, the in-sample correlation coefficient is used to assess how well the model fits the training data; the LOOCV correlation coefficient estimates the model's performance on unseen data, which helps assess the model's generalisation performance.

Appendix

```
# prepare section ----
# Load data
my_data <- load(file = "D:/desktop/MTLS/KI/biostatistic/assignment8_KW-2449_RUNX1.RData")

# Install and load required packages
# install.packages("dplyr")
# install.packages("tidyr")
# install.packages("gridExtra")
# install.packages("ggplot2")
library(dplyr)
library(tidyr)
library(ggplot2)
library(gridExtra)

# section 1 ----
# Question1.(i)(ii)

# Filter the required data
selected_drug_data <- drug_data %>%
  select(sample, auc)

selected_clinical_data <- clinical_data %>%
  select(sample, RUNX1)

# Based on the "sample column", merge two data frames
merged_clinical_drug_data <- merge(selected_drug_data, selected_clinical_data,
                                   by = "sample")
# Divide into two groups based on the value of the RUNX1

merged_clinical_drug_data_0 <- merged_clinical_drug_data %>%
  filter(RUNX1 == 0)

merged_clinical_drug_data_1 <- merged_clinical_drug_data %>%
  filter(RUNX1 == 1)

# Calculate the distribution and 95% confidence interval of AUC
summary_stats_0 <- summary(merged_clinical_drug_data_0$auc)
summary_stats_1 <- summary(merged_clinical_drug_data_1$auc)

ci_0 <- t.test(merged_clinical_drug_data_0$auc)$conf.int
ci_1 <- t.test(merged_clinical_drug_data_1$auc)$conf.int

# Output results
cat("Group with RUNX1=0 - AUC Distribution Summary:\n")
print(summary_stats_0)

cat("\n95% Confidence Interval for Group with RUNX1=0:\n")
print(ci_0)

cat("\nGroup with RUNX1=1 - AUC Distribution Summary:\n")
print(summary_stats_1)

cat("\n95% Confidence Interval for Group with RUNX1=1:\n")
print(ci_1)

# plot histogram
```

```r
histogram_0 <- ggplot(merged_clinical_drug_data_0, aes(x = auc)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of AUC for RUNX1=0",
       x = "AUC",
       y = "Frequency")

histogram_1 <- ggplot(merged_clinical_drug_data_1, aes(x = auc)) +
  geom_histogram(binwidth = 1, fill = "green", color = "black", alpha = 0.7) +
  labs(title = "Distribution of AUC for RUNX1=1",
       x = "AUC",
       y = "Frequency")

# plot confidence intervals
ci_plot_0 <- ggplot(data = data.frame(ci = ci_0), aes(x = 1, ymin = ci[1], ymax = ci[2])) +
  geom_errorbar(y = mean(merged_clinical_drug_data_0$auc), color = "blue") +
  geom_point(y = mean(merged_clinical_drug_data_0$auc), color = "blue") +
  labs(title = "95% Confidence Interval for RUNX1=0",
       x = "",
       y = "AUC")

ci_plot_1 <- ggplot(data = data.frame(ci = ci_1), aes(x = 1, ymin = ci[1], ymax = ci[2])) +
  geom_errorbar(y = mean(merged_clinical_drug_data_1$auc), color = "green") +
  geom_point(y = mean(merged_clinical_drug_data_1$auc), color = "green") +
  labs(title = "95% Confidence Interval for RUNX1=1",
       x = "",
       y = "AUC")

# plot boxplot
boxplot_0 <- ggplot(merged_clinical_drug_data_0, aes(x = factor(1), y = auc, fill = factor(1))) +
  geom_boxplot(fill = "blue", alpha = 0.7) +
  labs(title = "Boxplot of AUC for RUNX1=0",
       x = "",
       y = "AUC")

boxplot_1 <- ggplot(merged_clinical_drug_data_1, aes(x = factor(1), y = auc, fill = factor(1))) +
  geom_boxplot(fill = "green", alpha = 0.7) +
  labs(title = "Boxplot of AUC for RUNX1=1",
       x = "",
       y = "AUC")

# Arrange graphs via grid.arrange function
grid.arrange(histogram_0, histogram_1, boxplot_0, boxplot_1, ci_plot_0, ci_plot_1, ncol = 2)


# Question1.(iii)
library(boot)
library(dplyr)

# Create a Bootstrap function, which is used to resample with replacement from the sample
bootstrap_function <- function(data, indices) {
  sampled_data <- data[indices, ]
  return(mean(sampled_data$auc))
}

#Set Bootstrap sampling times
num_bootstrap_samples <- 1000

# Bootstrap resampling
set.seed(123)
```

```r
bootstrap_with_RUNX1 <- boot(merged_clinical_drug_data_1, statistic = bootstrap_function, R =
num_bootstrap_samples)

# Draw a histogram of Bootstrap results
histogram_Bootstrap_0 <- ggplot(data.frame(auc = bootstrap_with_RUNX1$t), aes(x = auc)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Bootstrap Distribution of AUC for RUNX1=1",
       x = "AUC",
       y = "Frequency")

grid.arrange(histogram_Bootstrap_0)

# section 2 ----
# Question2.(i)(ii)(iii)

# "selected_drug_data" contains two columns of data, "AUC" and "sample".
# Load necessary libraries

# install.packages("Hmisc")
#install.packages("htmltools", dependencies=TRUE)
library(tidyverse)
library("Hmisc")

# Transpose selected_drug_data
selected_drug_data_transpose<- t(selected_drug_data)

# Use sample as column name
colnames(selected_drug_data_transpose) <- selected_drug_data_transpose[1,]

# Merge the two data sets to get data with "auc" and "gex"
combined_data_auc_gex <- rbind(selected_drug_data_transpose,gex)
combined_data_auc_gex <-   combined_data_auc_gex[-1, ]

# Transpose the “combined_data_auc_gex” data to match the requirements
combined_data_auc_gex__transpose <- t(combined_data_auc_gex)

# Calculate the correlation coefficient between auc and genes
regression_gene_auc <- rcorr(as.matrix(combined_data_auc_gex__transpose), type = c("pearson"))

# Extract the correlation coefficient and P value
cor_col <- regression_gene_auc$r[,1]
p_value_col <- regression_gene_auc$P[,1]
#head(p_value_col)
#head(cor_col)

# Store correlation coefficient and P value into dataframe
regression_result_df <- data.frame(
  gene = rownames(regression_gene_auc$r)[-1],   # The first row is AUC, so exclude
  cor = cor_col[-1],   # Exclude correlation coefficient with itself
  p_value = p_value_col[-1]   # Exclude and own P-value
)
# p_values is the original P value vector, which needs to be corrected using the BH method
adjusted_p_values <- p.adjust(regression_result_df$p_value, method = "BH")

# Add adjusted P-values to the results data frame
regression_result_df$adjusted_p_value <- adjusted_p_values

# Convert the adjusted P value to numeric type
regression_result_df$adjusted_p_value <- as.numeric(regression_result_df$adjusted_p_value)
```

```
# Print results
# head(regression_result_df)

# Histogram of P values before BH adjustment
hist(regression_result_df$p_value, main = "P Value Distribution (Unadjusted)",
        xlab = "P Value", ylab = "Frequency", col = "lightblue", border = "black")

# Histogram of P values after BH adjustment
hist(regression_result_df$adjusted_p_value, main = "Adjusted P Value Distribution (BH Method)",
        xlab = "Adjusted P Value", ylab = "Frequency", col = "lightgreen", border = "black")

# Screen for genes with high correlation
correlated04_result <- regression_result_df[regression_result_df$cor > 0.4, ]
head(correlated04_result)

# Filter rows with adjusted_p_value less than 0.05
significant005_results <- regression_result_df[regression_result_df$adjusted_p_value < 0.05, ]
head(significant005_results)


# Question2.(iv)

# Filter out the gene expression data of ST6GALNAC3
row_index <- rownames(gex) == "ST6GALNAC3"
gex_ST6GALNAC3 <- gex[row_index,]

# Create a new data frame with two columns: Sample and Value
gex_ST6GALNAC3 <- data.frame(
   sample = colnames(gex),
   ST6GALNAC3 = as.numeric(as.matrix(gex_ST6GALNAC3))
)

# Merge the gene expression data of ST6GALNAC3 with merged_clinical_drug_data
merged_ST6GALNAC3_auc_RUNX1 <- merge(merged_clinical_drug_data, gex_ST6GALNAC3,
                                                        by = "sample")

# Use the lm function to fit the linear regression model
model1 <- lm(auc ~ ST6GALNAC3, data = merged_ST6GALNAC3_auc_RUNX1)
model2 <- lm(auc ~ ST6GALNAC3 + RUNX1, data = merged_ST6GALNAC3_auc_RUNX1)
model3 <- lm(auc ~ RUNX1, data = merged_ST6GALNAC3_auc_RUNX1)

# View summary of regression model
summary(model1)
summary(model2)
summary(model3)



# section 3 ----
# Question3.(i)

# Calculate the distance matrix between samples
dist_matrix <- dist(t(gex), method = "euclidean")

# Perform hierarchical clustering
hclust_result <- hclust(dist_matrix, method = "ward.D2")

# Divide clusters based on hierarchical clustering results and select clusters at a specific level
```

```
clusters <- cutree(hclust_result, k = 5)

# Print the number of samples in each cluster
cluster_counts <- table(clusters)
for (i in seq_along(cluster_counts)) {
  cat("Cluster", i, ":", cluster_counts[i], "\n")
}

# Set graphics parameters and increase the size of the graph
par(mfrow = c(1, 1), mar = c(2,3,2,2) + 0.01)

# Draw hierarchical clustering dendrogram
p   <-   plot(hclust_result, main = "Hierarchical Clustering Dendrogram", cex = 0.2)

# Draw the boundaries of cluster divisions
rect.hclust(hclust_result, k = 5, border = 2:6)


# Question3.(ii)
#BiocManager::install("heatmaps")
library(heatmaps)
library("pheatmap")

# Transpose selected_clinical_data
selected_clinical_data_transpose<- t(selected_clinical_data)

# Use sample as the column name
colnames(selected_clinical_data_transpose) <- selected_clinical_data_transpose[1,]

# Merge two data sets
combined_data_RUNX_gex <- rbind(selected_clinical_data_transpose,gex)
combined_data_RUNX_gex <-   combined_data_RUNX_gex[-1, ]

classification_info <- as.factor(combined_data_RUNX_gex[1, ])

# Creating a new dataframe for annotation
annotation_c <- data.frame(Classification = classification_info)

# Setting row names to match the column names of the original dataframe
rownames(annotation_c) <- colnames(combined_data_RUNX_gex)

pheatmap(gex,
         cluster_rows = T,
         cluster_cols = T,
         annotation_col =annotation_c, # Sample classification data
         annotation_legend=TRUE, # Display sample classification
         show_rownames = F,
         show_colnames = F,
         scale = "none", # No normalization
         color =colorRampPalette(c("#8854d0", "#ffffff","#fa8231"))(100)
)


# section 4 ----
# Question4.(i)
# Directly use combined_data_auc_gex data in Question2

# Remove the first row (AUC value)
gene_expression_data <- combined_data_auc_gex[-1, ]
```

```r
# Calculate the variance of genes
gene_variances <- apply(gene_expression_data, 1, var)

# Find the 100 genes with the highest variance
top_genes <- names(sort(gene_variances, decreasing = TRUE)[1:100])

# Select these 100 genes in the data frame
selected_genes_data <- combined_data_auc_gex[top_genes,]
selected_genes_data    <- rbind(selected_drug_data_transpose,selected_genes_data)
selected_genes_data <- selected_genes_data[-1,]

# Make sure selected_genes_data is a data frame
selected_genes_data <- t(selected_genes_data)
selected_genes_data <- as.data.frame(selected_genes_data)

# Use lapply to convert each column into a numeric type
#(Note: Not numeric type will cause a lot of trouble)
selected_genes_data[top_genes] <- lapply(selected_genes_data[top_genes], as.numeric)

# Define multiple linear regression model
lm_model <- lm(auc ~ ., data = selected_genes_data)

# Output model summary
summary(lm_model)

combined_data_auc_gex    <- as.data.frame(combined_data_auc_gex)

# Calculate the observed values and predicted values of sample BA2409
sample_index <- which(colnames(combined_data_auc_gex) == "BA2409")
observed_auc <- combined_data_auc_gex[sample_index,1]
predicted_auc <- predict(lm_model, newdata = selected_genes_data[sample_index, , drop = FALSE])

print(paste("Observed AUC:", observed_auc))
print(paste("predictedAUC:", predicted_auc))

# Get the observed values and predicted values of all samples
observed_auc_all <- combined_data_auc_gex[1, ]
predicted_auc_all <- predict(lm_model, newdata = selected_genes_data[,-1])

# Convert observed values and predicted values to numeric types
observed_auc_all <- as.numeric(observed_auc_all)
predicted_auc_all <- as.numeric(predicted_auc_all)

# Draw the relationship between observed values and predicted values of all samples
plot(observed_auc_all, predicted_auc_all, main = "Observed value and predicted value relationship graph",
        xlab = "Observed value", ylab = "predicted value", col = "blue", pch = 16)
# Add a diagonal line to represent perfect fit
abline(0, 1, col = "red")

# Get the residuals of the model
residuals <- residuals(lm_model)

# Draw residual plot
plot(predicted_auc_all, residuals, main = "Residual Plot",
        xlab = "predicted AUC", ylab = "Residuals", col = "blue", pch = 16)
# Add a horizontal line to check the distribution of residuals
abline(h = 0, col = "red")
```

```
# Draw a smooth curve and check the homogeneity of variances of the residuals
lines(lowess(predicted_auc_all, residuals), col = "green")

#Add legend
legend("topright", legend = c("Residuals", "Smoothed Line"), col = c("blue", "green"), pch = 16)

# Draw Q-Q plot
qqnorm(residuals)
qqline(residuals, col = "red")

# Add title and tags
xlabel <- "Theoretical Quantiles"
ylabel <- "Sample Quantiles"
xlab(xlabel)
ylab(ylabel)


# 4.(ii)

# Load required library
library(boot)

# Assuming loocv_model is your regression model
# predicted_auc_all is the predicted values
# observed_auc_all is the observed (actual) values
# selected_genes_data is your dataframe with AUC in the first column and other predictors in the remaining columns

# In-sample correlation coefficient
in_sample_cor <- cor(predicted_auc_all, observed_auc_all)

# Leave-One-Out Cross-Validation
# Initialize an empty vector to store cross-validated predictions
loocv_predictions <- numeric(length(observed_auc_all))

selected_genes_data <- as.data.frame(sapply(selected_genes_data, as.numeric))
str(selected_genes_data)

# Perform leave-one-out cross-validation
for (i in 1:length(observed_auc_all)) {
  # Exclude the i-th observation from the training set
  train_data <- selected_genes_data[-i, ]

  # Fit the model on the training data
  loocv_model <- lm(auc ~ ., data = train_data)

  # Predict on the i-th observation
  loocv_predictions[i] <- predict(loocv_model, newdata = selected_genes_data[i, , drop = FALSE])
}

# Calculate the correlation coefficient for leave-one-out cross-validation
loocv_cor <- cor(loocv_predictions, observed_auc_all)

# Display the results
cat("In-Sample Correlation Coefficient:", in_sample_cor, "\n")
cat("Leave-One-Out Cross-Validation Correlation Coefficient:", loocv_cor, "\n")
```