# Stat131A-Project

AUTHOR
Colin Asbill

## Background

### Question 1

Bililary Cholangitis is an autoimmune disease that impacts the liver, by causing the bile ducts to swell and eventually be destroyed, this can lead to cirrhosis and liver failure. Biliary Cholangitis primarily affects women. About 58 out of every 100,000 US women and about 15 out of every 100,000 US men are affected by Biliary Cholangitis. Risk factors include infections such as UTIs(uniary tract infection), chronic cigarette smoking, exposure to toxic chemicals and having a family member with the disease all lead to an increased chance of getting biliary cholangitis.

### Question 2

Survival analysis is a part of statistics that studies how long it takes an event to occur, it is useful in studying diseases and estimate the likelihood of death a patient has with a certain illness. The outcome variable of interest in survival analysis is time until an event occurs, this causes the distribution to be typically non normal, with events skewed towards the beginning of the diagnosis. Because survival analysis uses censoring where a subset of the study group will have unknown survival times because of the study ends, before a patient experiences the relevant outcome or if follow up isn't possible with that patient anymore. This analysis could be helpful in the real world because it can measure the impact of the D-penicillamine drug compared to a placebo. This data could help a company decide if a drug is worth scaling up for mass production if it is an effective drug. Sending an ineffective drug for mass production costs a lot of money and time that could be allocallated towards producing an effective drug that could help people suffering with billary cholangitis.

## First Steps

### Question 1

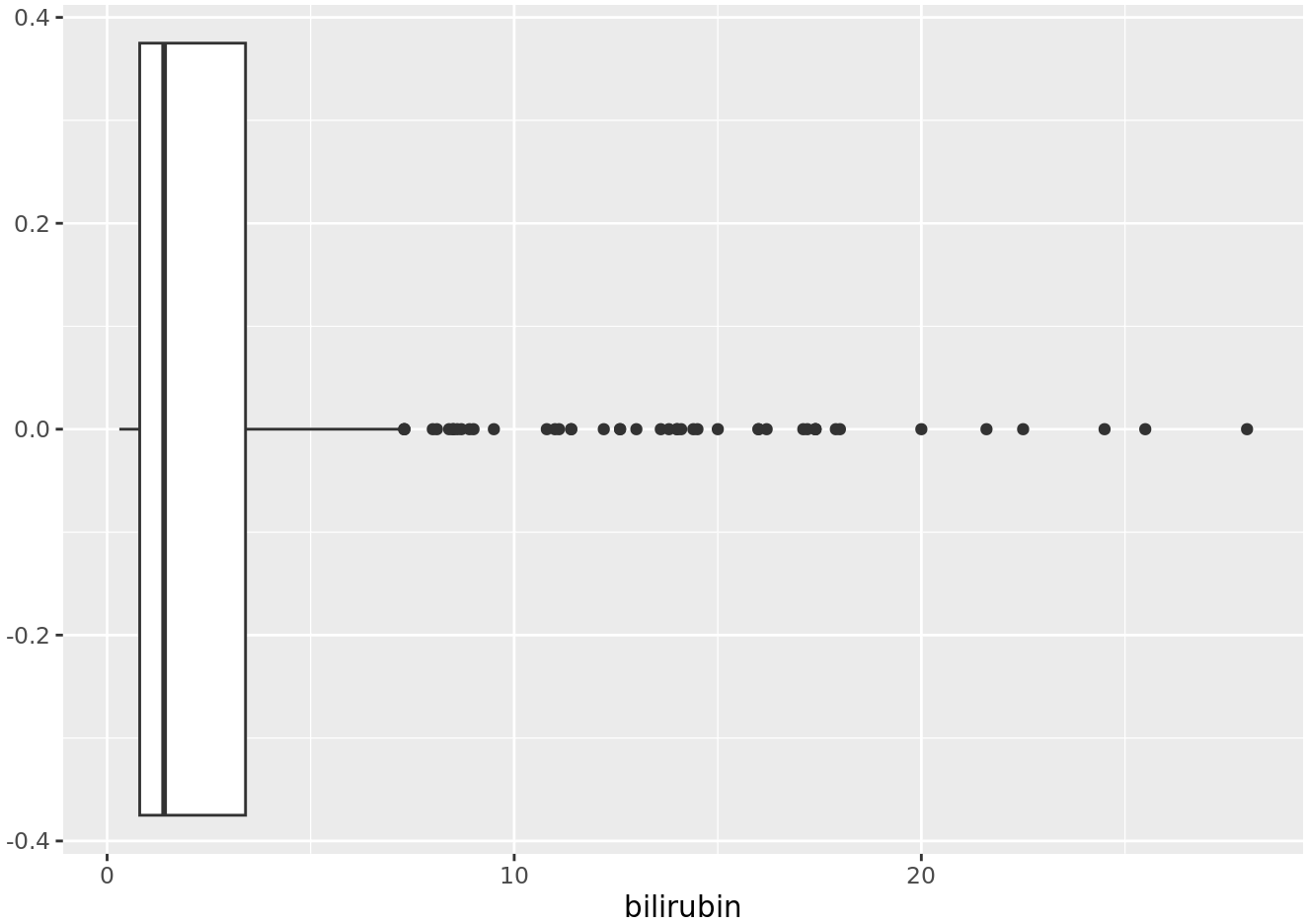
```
library(tidyverse)
library(broom)
library(GGally)
library(ggplot2)
library(leaps)
library(rpart)
library(rpart.plot)
library(randomForest)
cho <- read.csv("cholangitis.csv")
cho_dict <- read.csv("cholangitis_dictionary.csv")
```

```
cho <- cho %>%
  replace_na(list(x = 0, y = "Unknown"))
```

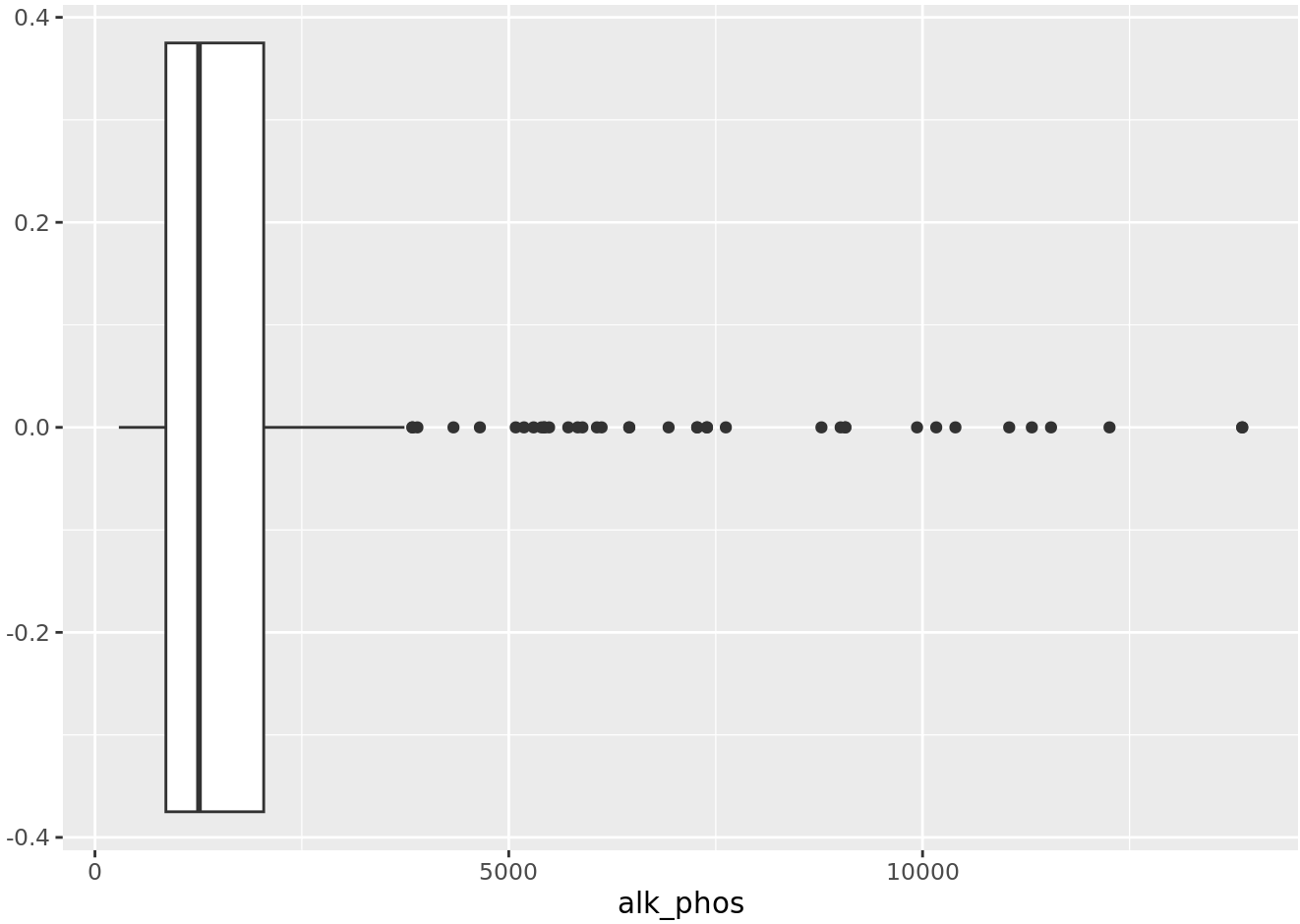
```
cho$status <- factor(cho$status, levels = c("C", "D", "CL"))
cho$stage <- factor(cho$stage, levels = c("1", "2", "3", "4"))
cho$drug <-factor(cho$drug, levels = c("Placebo", "D-penicillamine", "Unknown"))
cho$sex <- factor(cho$sex, levels = c("F", "M"))
cho$ascites <-factor(cho$ascites, levels = c("Y", "N"))
cho$hepatomegaly <- factor(cho$hepatomegaly, levels = c("Y", "N"))
cho$spiders <- factor(cho$spiders, levels = c("Y", "N"))
cho$edema <- factor(cho$edema, levels = c("Y", "N", "S"))
```

## Question 2

```
cho %>%
  ggplot(aes(x = bilirubin )) +
  geom_boxplot()
```

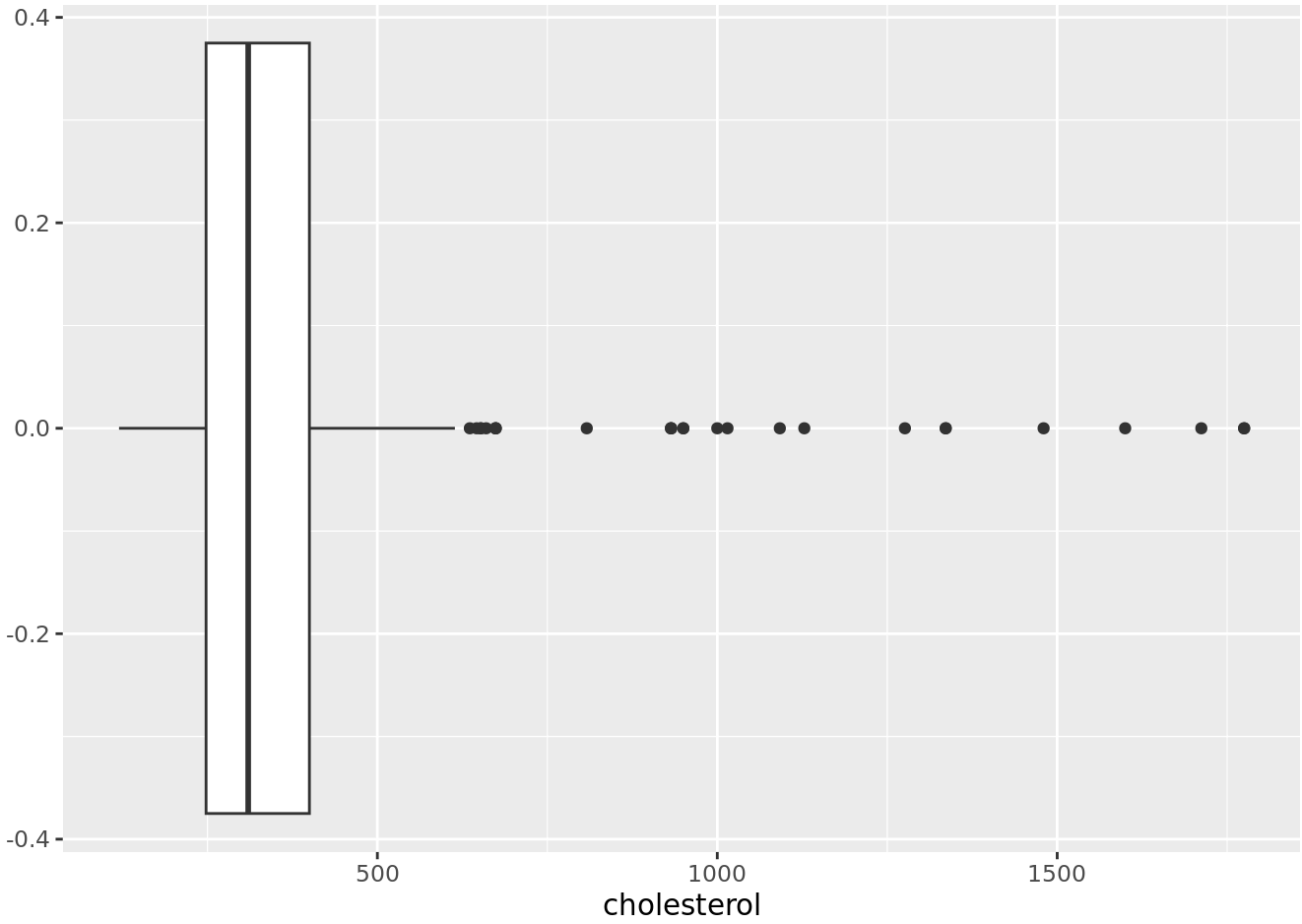



```
cho %>%
  ggplot(aes(x = alk_phos)) +
  geom_boxplot()
```

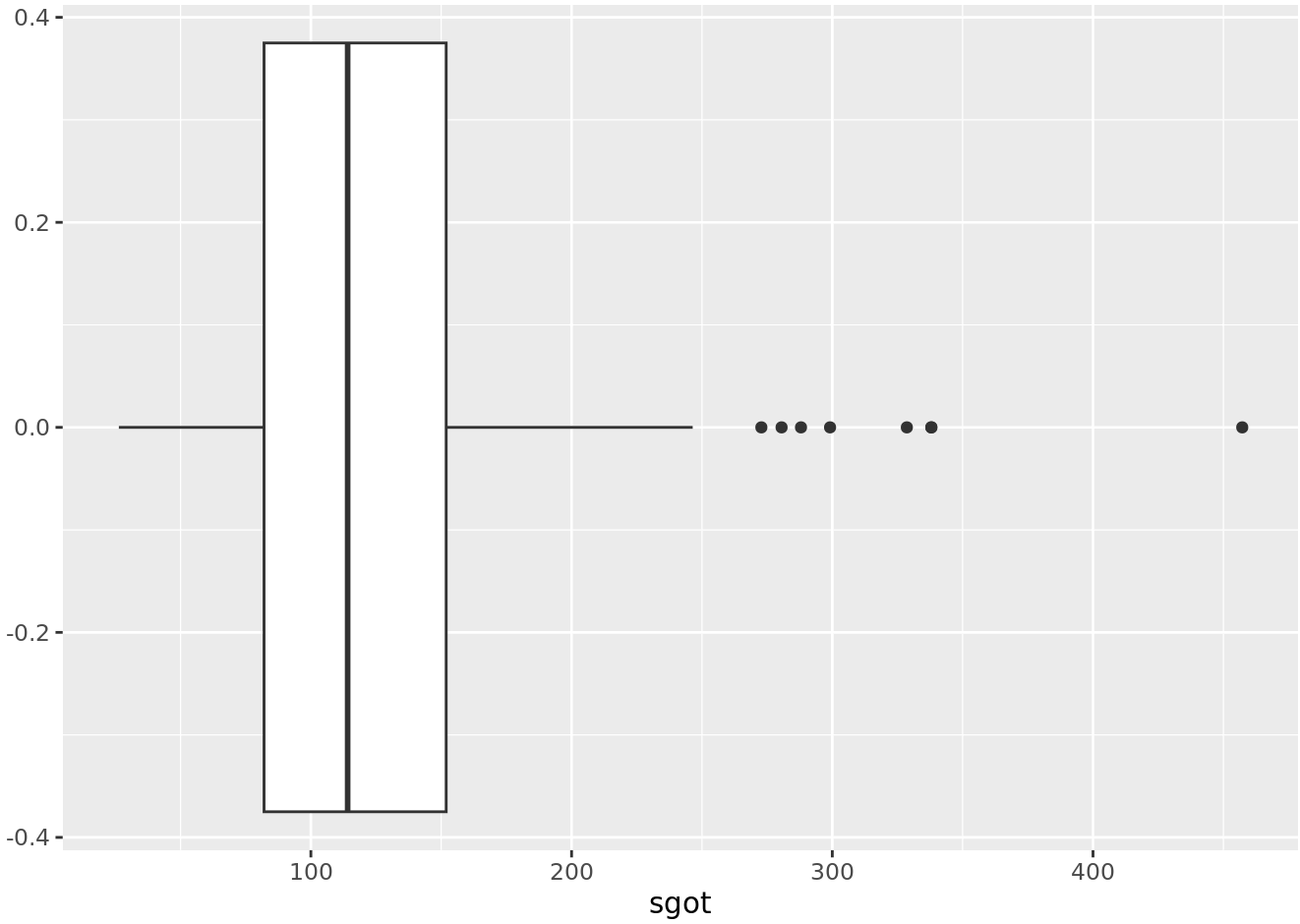



```
cho %>%
  ggplot(aes(x = cholesterol )) +
  geom_boxplot()
```

```
Warning: Removed 5 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```



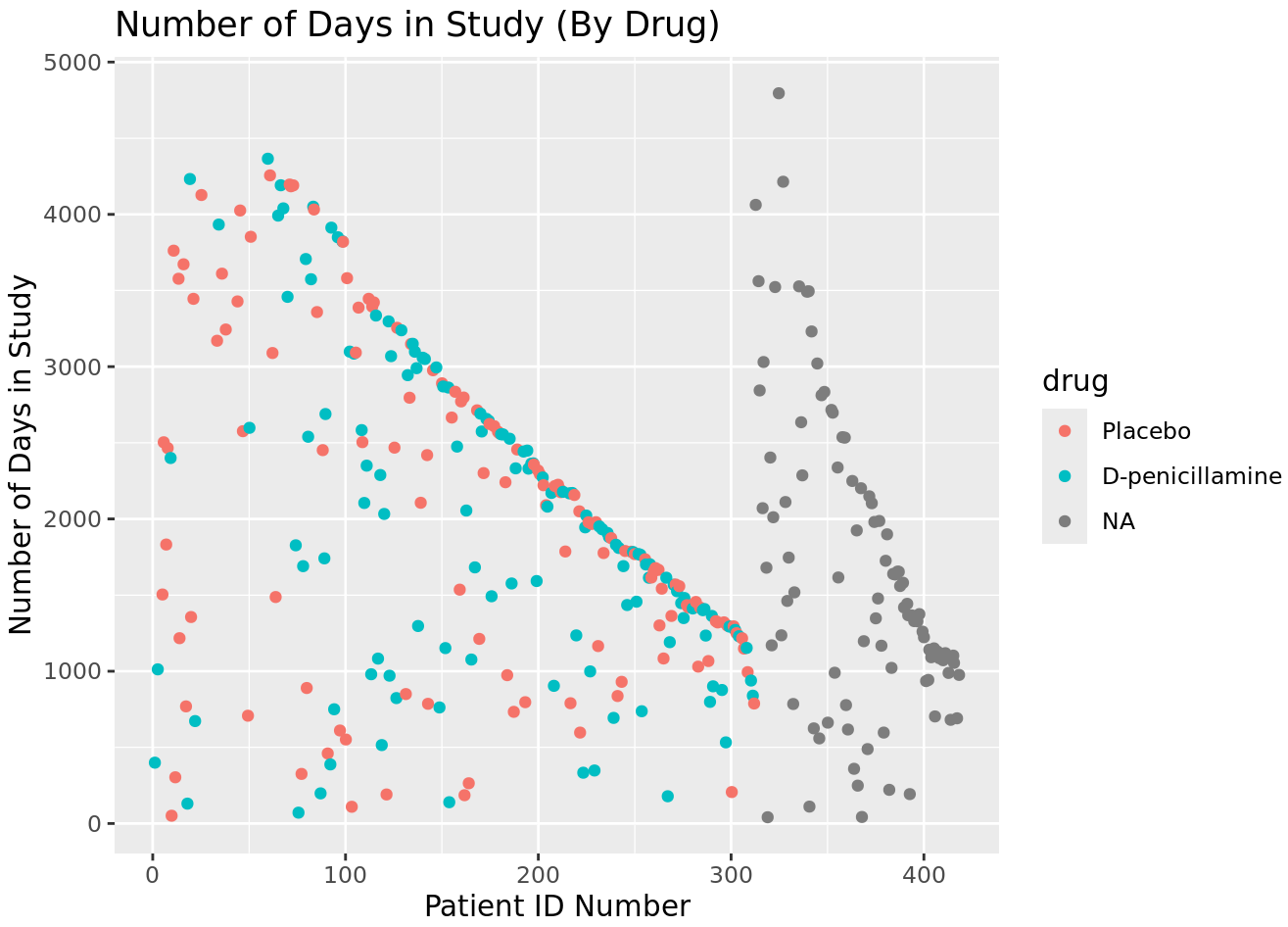

```
cho %>%
  ggplot(aes(x = sgot)) +
  geom_boxplot()
```




```
cho_filtered <- cho %>%
  filter(bilirubin < 15, alk_phos < 5000, cholesterol < 1000, sgot < 300)
```

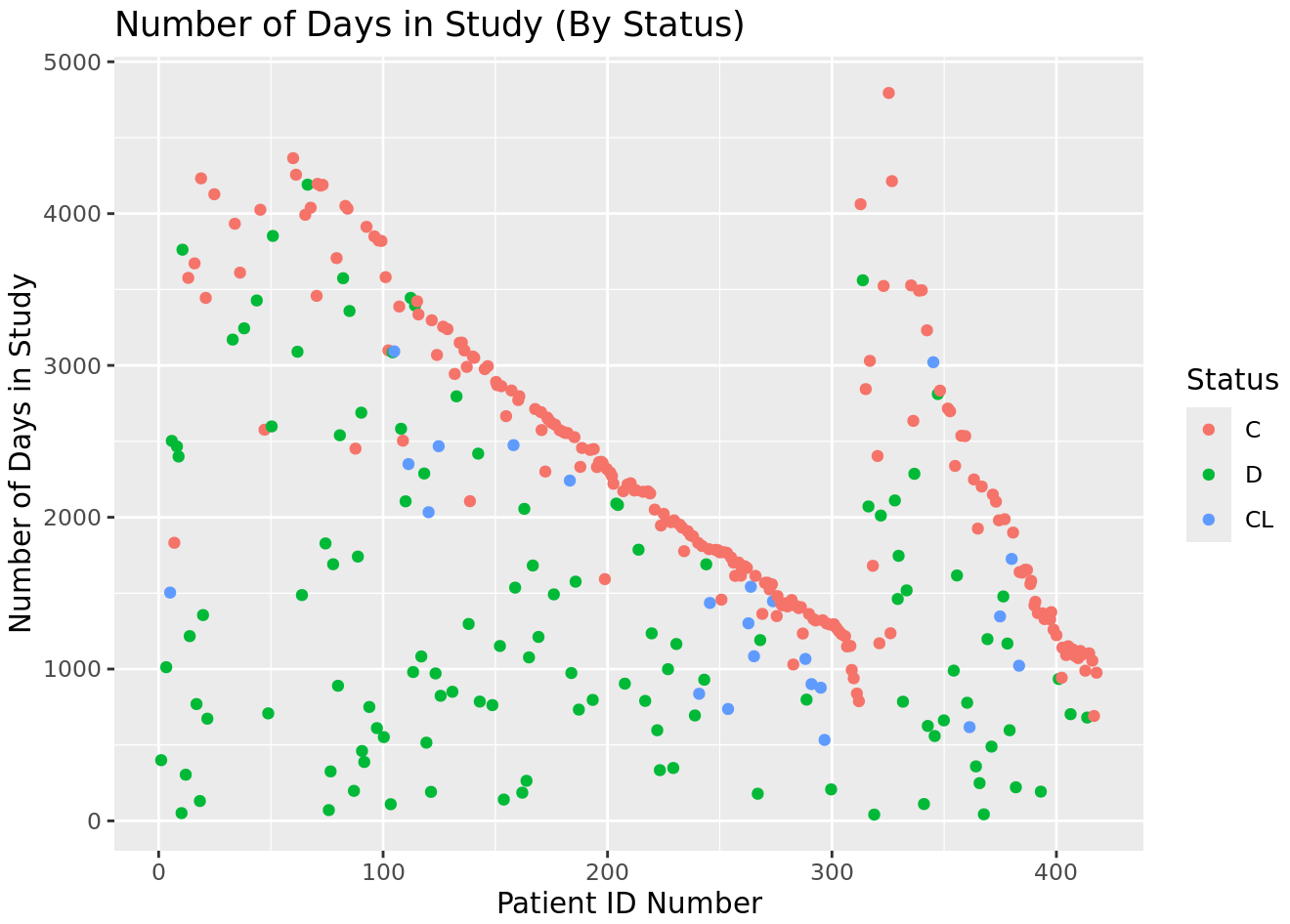
Created a bunch of boxplots to find outliers in the dataset and then filtered them out, only kept the boxplots of variables I filtered.

```
cho_filtered %>%
  ggplot(aes(x = id, y = n_days)) +
  geom_jitter(aes(color = drug)) +
  labs(title = "Number of Days in Study (By Drug)", x = "Patient ID Number", y = "Number of Days
```



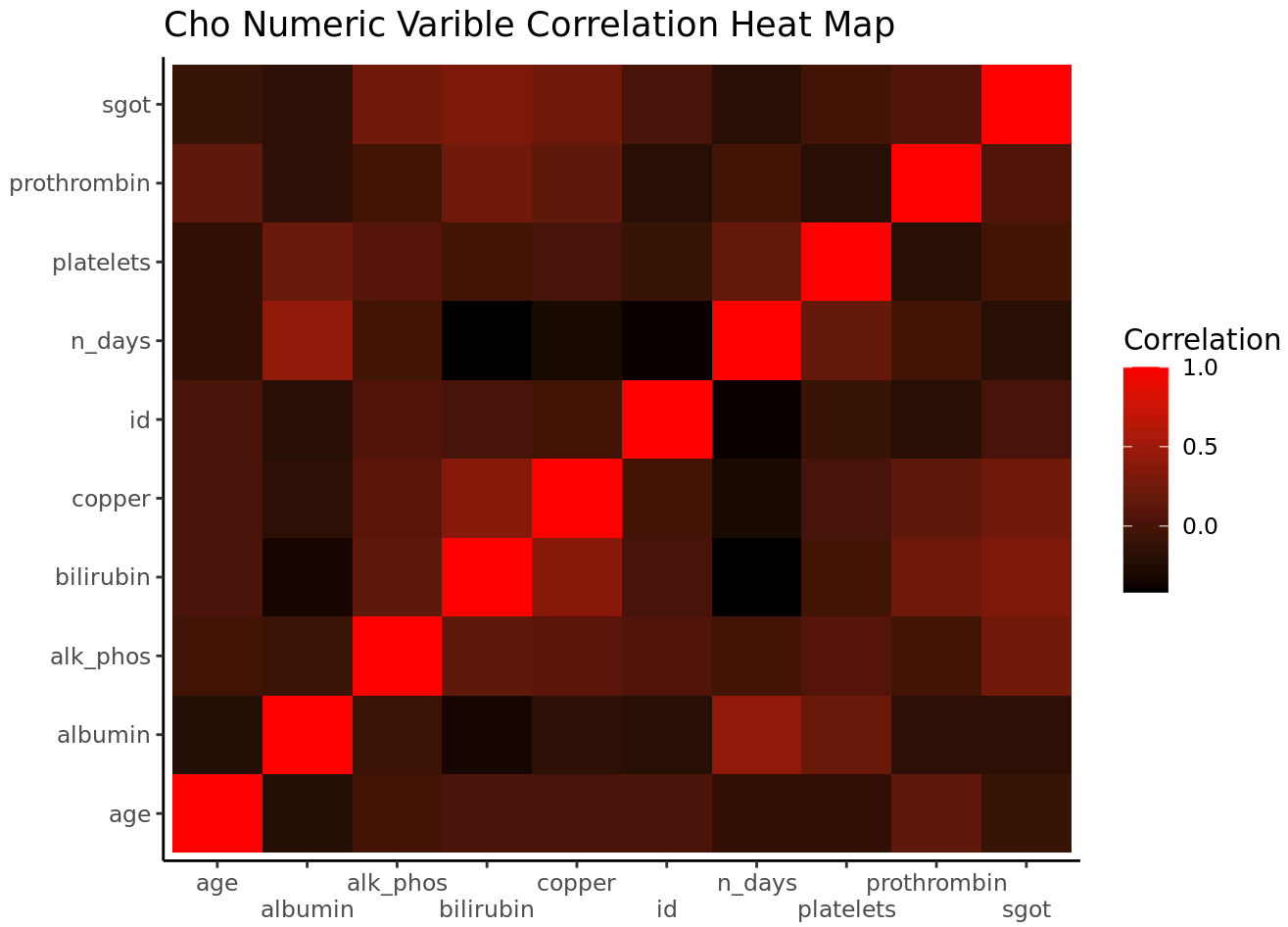

Based on this graph its hard to determine if the drug has an effect on the number of days a patient lasts in the study, but we can see that all the "NA" patients had IDs greater than 300 and if we look in the data table we can see that patients 313-418 are NA and did not receive placebo or "D-penicillamine". It might be a good idea to filter out this NA data by filtering out patients 313-418 to measure the impact of the drug on n_days.

```
cho_filtered %>%
  ggplot(aes(x = id, y = n_days)) +
  geom_jitter(aes(color = status)) +
  labs(title = "Number of Days in Study (By Status)", x = "Patient ID Number", y = "Number of Days
```




In this graph we can see a similar sloping line as in the first graph where patients are exiting the study this sloping line starts around Patient 100 and ends just past patient 300. One explanation could be these patients entered the study later than earlier patients, so they are all exiting the study because the study ended causing this sloping line of patients that were not dead at the end of the study.

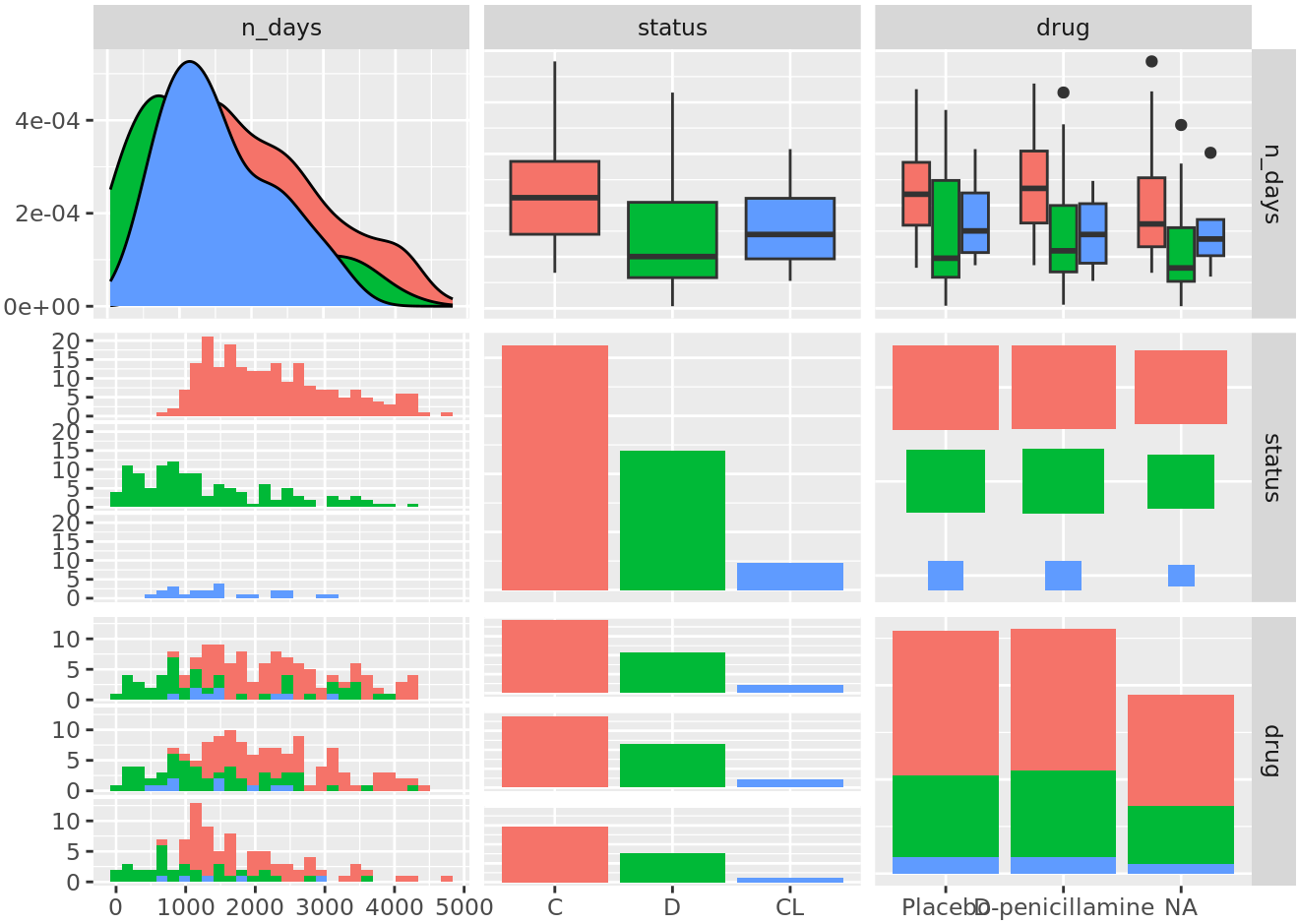
```
cho_num <- cho_filtered %>%
  select(id, n_days, age, bilirubin, albumin, copper, alk_phos, sgot, platelets, prothrombin)
as.data.frame(cor(cho_num)) %>%
  rownames_to_column("Variables_1") %>%
  pivot_longer(-c(Variables_1), names_to = "Variables_2", values_to = "Correlation") %>%
  ggplot(mapping = aes(x = Variables_1, y = Variables_2)) +
    geom_tile(aes(fill = Correlation)) +
    scale_fill_gradient(low = "black", high = "red")+
    scale_x_discrete(guide = guide_axis(n.dodge = 2)) +
  theme_classic() +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank()) +
  labs(title = "Cho Numeric Varible Correlation Heat Map")
```




This is a correlation heat map of all the numerical variables, we don't any extremely strong correlations, but we do see that copper, bilirubin and alk_phos have some correlation and more importantly that n_days has correlation with albumin, alk_phos and platelets.

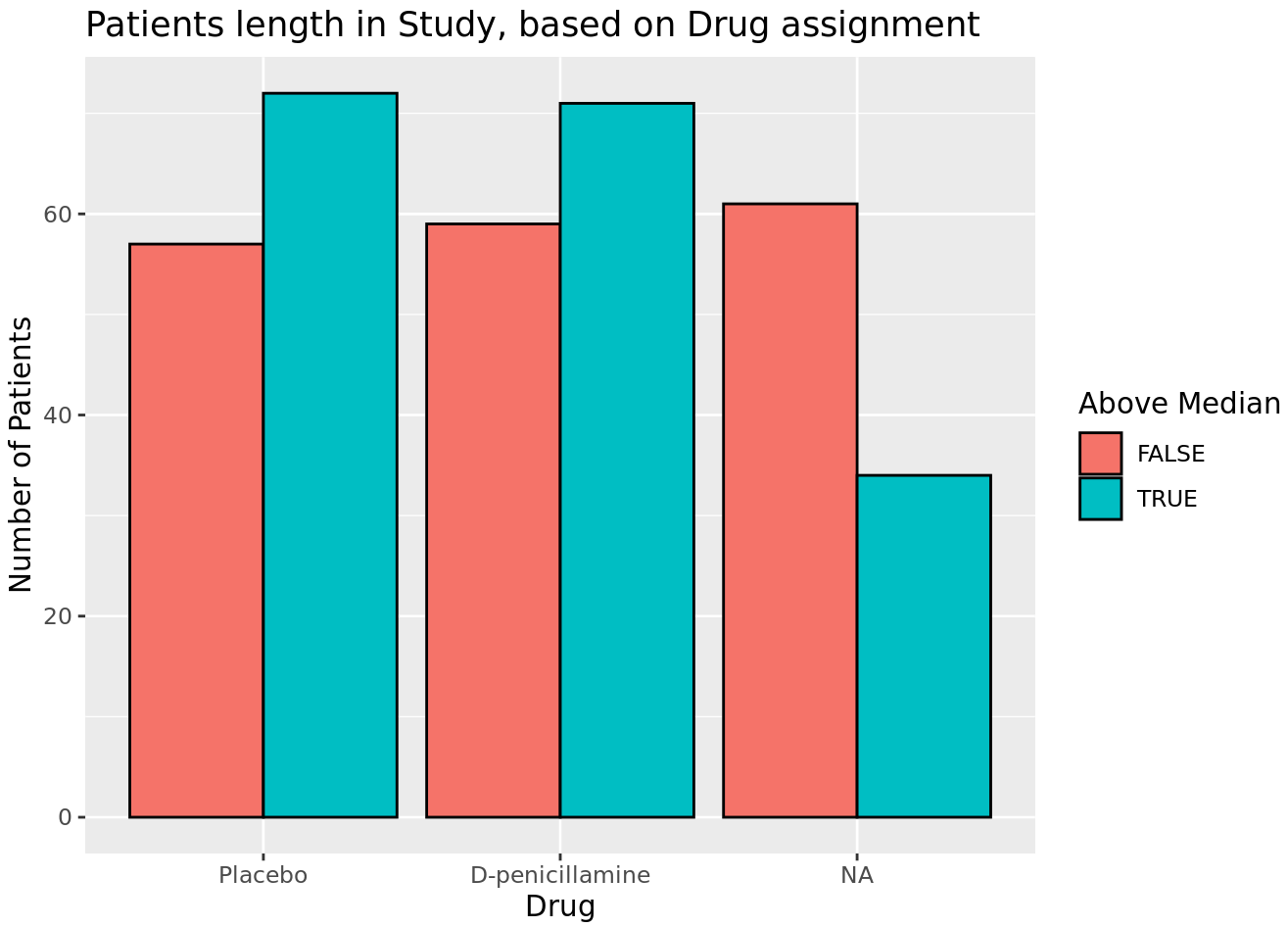
```
cho_filtered %>%
  select(n_days, status, drug) %>%
  ggpairs(aes(color = status))
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



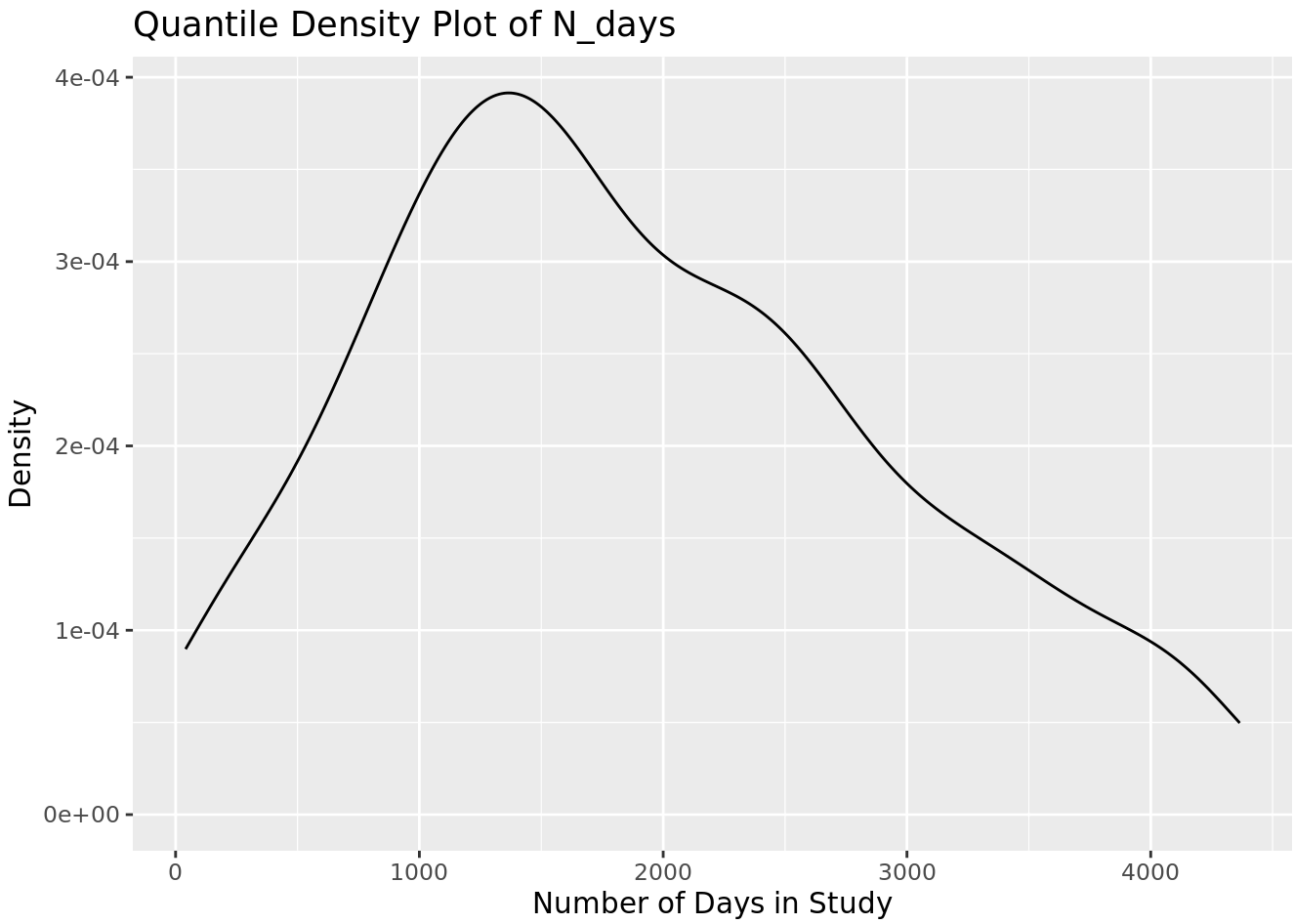

Based on this pairs plot we can see that patients that take the drug seem to take slightly longer to die compared to patients that received the placebo, we can also see that most deaths occur within the first 2000 days of the study.

```
cho_med <- cho_filtered%>%
  mutate(above_median = n_days > median(n_days))
cho_med %>%
  ggplot(aes(x = drug, fill = above_median)) +
  geom_bar(color = "black", position = "dodge") +
  labs(title = "Patients length in Study, based on Drug assignment", x = "Drug", y = "Number of Pa
```




This bar chart shows us the numbers of Patients that survived above the median number of days in the study categorized by Drug given. Based on these graphs we can see that slightly more D-penicillamine patients survived past the median than Placebo patients, but both last longer than NA patients who mostly stayed less than the median number of days in the study.

```
Q1 <- quantile(cho_filtered$n_days, 0.25)
Q3 <- quantile(cho_filtered$n_days, 0.75)
IQR <- Q3 - Q1

lower <- Q1 - 1.5*IQR
upper <- Q3 + 1.5*IQR

cho_quant <- cho_filtered[cho_filtered$n_days >= lower & cho_filtered$n_days <= upper,]
ggplot(data = cho_quant, aes(x = n_days)) +
  geom_density() +
  labs(title = "Quantile Density Plot of N_days",
         x= "Number of Days in Study",
       y = "Density")
```




Density is not normally distributed, is concentrated between 0 and 3000, peaking around 1400 days.

# Explanatory Modeling

## Question 3

```
cho_filtered <- cho_filtered %>%
  drop_na()
lm1 <- lm(n_days ~. -id,  data = cho_filtered)
summary(lm1)
```

```
Call:
lm(formula = n_days ~ . - id, data = cho_filtered)

Residuals:
     Min       1Q   Median       3Q      Max
-2409.79  -598.74   -31.66   554.74  2300.43

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        -1.992e+03  1.095e+03  -1.818 0.070308 .
```

```
statusD             -5.078e+02  1.514e+02  -3.353 0.000932 ***
statusCL            -5.858e+02  2.271e+02  -2.579 0.010503 *
drugD-penicillamine  3.278e+00  1.111e+02   0.029 0.976497
age                 -6.172e-03  1.663e-02  -0.371 0.710961
sexM                 7.846e+01  1.891e+02   0.415 0.678601
ascitesN            -5.772e+01  3.067e+02  -0.188 0.850907
hepatomegalyN       -1.934e-01  1.290e+02  -0.001 0.998805
spidersN             3.066e+01  1.386e+02   0.221 0.825152
edemaN               3.964e+02  3.312e+02   1.197 0.232641
edemaS               1.164e+02  3.497e+02   0.333 0.739521
bilirubin           -7.784e+01  2.872e+01  -2.710 0.007213 **
cholesterol          1.324e-02  4.967e-01   0.027 0.978759
albumin              5.552e+02  1.618e+02   3.432 0.000707 ***
copper              -1.994e+00  8.213e-01  -2.428 0.015941 *
alk_phos             1.287e-01  9.409e-02   1.368 0.172513
sgot                 3.519e-01  1.401e+00   0.251 0.801939
tryglicerides        1.528e+00  1.138e+00   1.343 0.180685
platelets            7.756e-01  6.735e-01   1.151 0.250706
prothrombin          2.086e+02  6.802e+01   3.067 0.002415 **
stage2              -3.487e+02  2.738e+02  -1.274 0.204043
stage3              -3.987e+02  2.710e+02  -1.471 0.142513
stage4              -6.055e+02  2.931e+02  -2.066 0.039940 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 849.4 on 236 degrees of freedom
Multiple R-squared:  0.4092,    Adjusted R-squared:  0.3541
F-statistic:  7.43 on 22 and 236 DF,  p-value: < 2.2e-16
```

Firstly I filtered out outliers and the patients that had drug status as NA, so that we could measure the impact of D-pencillamine on n_days. I also changed all the categorical variables into factors as instructed, so the first level will be included in the baseline and therefore is missing from the regression table. We can see that statusD and albumin are statistically significant to the 1% level, bilirubin and prothrombin are significant to the 5% level, statusCL, copper and stage 4 are significant to the 10% level. Now I'm gonna look at the p.values and cooks distance to eliminate some high influence points and run a new regression with the cleaned data.

```
lm1 %>%
  glance() %>%
  select(df, df.residual, statistic, p.value)
```

```
# A tibble: 1 × 4
     df df.residual statistic  p.value
  <dbl>       <int>     <dbl>    <dbl>
1    22         236      7.43 3.62e-17
```

```
lm1 %>%
  tidy() %>%
  select(term, estimate, std.error, statistic, p.value)
```

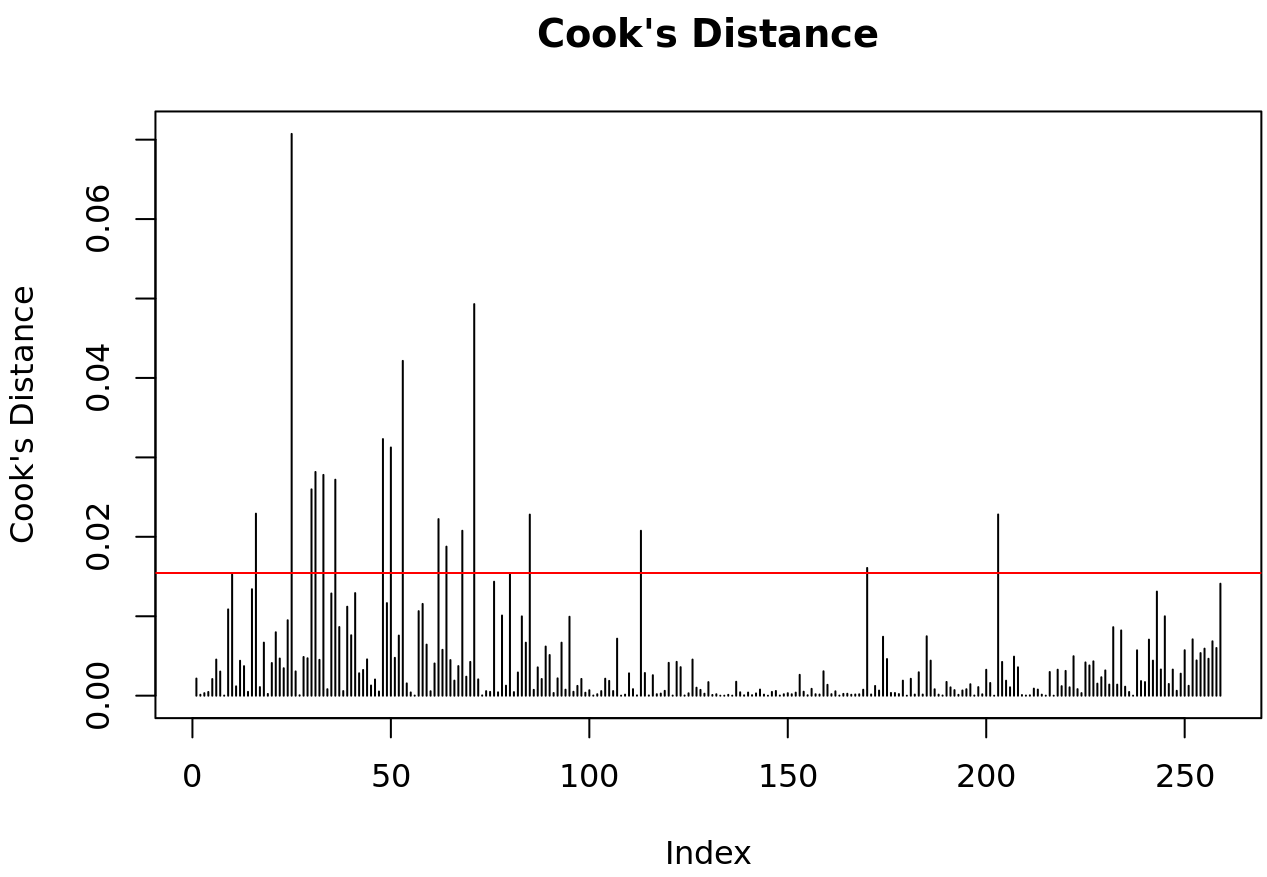
```
# A tibble: 23 × 5
   term                  estimate std.error statistic  p.value
   <chr>                    <dbl>     <dbl>     <dbl>    <dbl>
 1 (Intercept)         -1992.      1095.      -1.82    0.0703
 2 statusD              -508.       151.      -3.35    0.000932
 3 statusCL             -586.       227.      -2.58    0.0105
 4 drugD-penicillamine     3.28     111.       0.0295  0.976
 5 age                    -0.00617    0.0166  -0.371   0.711
 6 sexM                   78.5      189.       0.415   0.679
 7 ascitesN              -57.7      307.      -0.188   0.851
 8 hepatomegalyN          -0.193    129.      -0.00150 0.999
 9 spidersN               30.7      139.       0.221   0.825
10 edemaN                396.       331.       1.20    0.233
# ℹ 13 more rows
```

```
confint(lm1, level = 0.95)
```

```
                            2.5 %        97.5 %
(Intercept)         -4.149999e+03  166.41502089
statusD             -8.061071e+02 -209.39680885
statusCL            -1.033140e+03 -138.37778015
drugD-penicillamine -2.156908e+02  222.24667097
age                 -3.894203e-02    0.02659902
sexM                -2.940973e+02  451.01481603
ascitesN            -6.620115e+02  546.57546920
hepatomegalyN       -2.542874e+02  253.90057093
spidersN            -2.424079e+02  303.71878384
edemaN              -2.561790e+02 1048.92541924
edemaS              -5.724820e+02  805.28042390
bilirubin           -1.344259e+02  -21.26305542
cholesterol         -9.653584e-01    0.99183792
albumin              2.365336e+02  873.95929135
copper              -3.612136e+00   -0.37593946
alk_phos            -5.661622e-02    0.31409767
sgot                -2.409065e+00    3.11294496
tryglicerides       -7.141702e-01    3.77056114
platelets           -5.513685e-01    2.10250951
prothrombin          7.460875e+01  342.61303283
stage2              -8.881459e+02  190.68488966
stage3              -9.325641e+02  135.12595914
stage4              -1.183022e+03  -28.05849756
```

The p-value is close to zero, so we can conclude that at least one coefficient is non zero. The confidence intervals are pretty wide which introduces a lot of variance into the regression making it hard to find the impact of each variable, lets eliminate some high influence points to improve this.

```
cooksd <- cooks.distance(lm1)
plot(cooksd, type="h", main="Cook's Distance", ylab="Cook's Distance")
abline(h=4/(nrow(cho_filtered)), col="red")
```



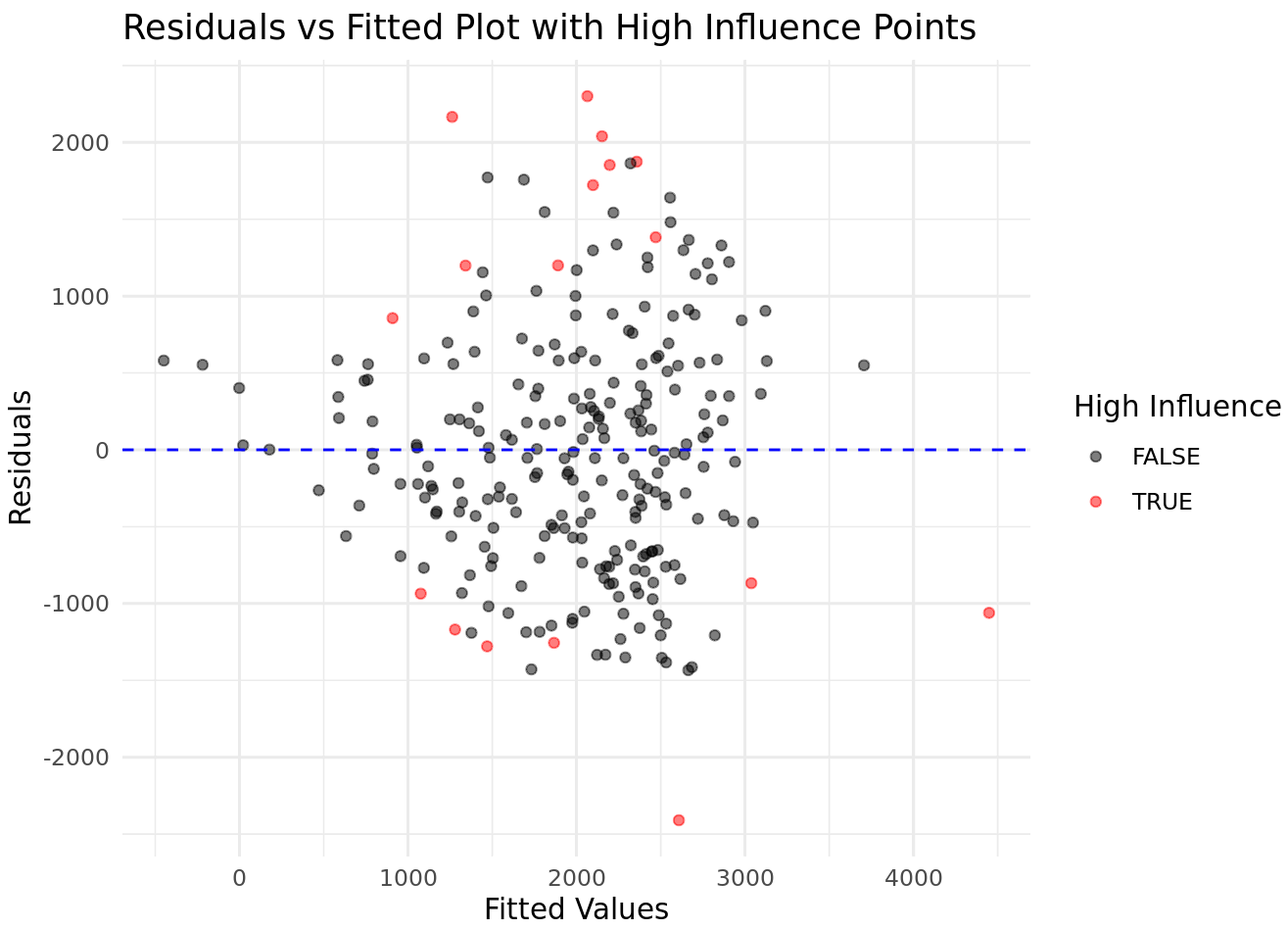

```
cho_filtered %>%
  mutate(cooksd = cooks.distance(lm1)) %>%
  filter(cooksd > 0.02) %>%
  select(id, n_days, status, drug)
```

```
    id n_days status            drug
1   19   4232      C D-penicillamine
2   44   3428      D         Placebo
3   51   3853      D         Placebo
4   60   4365      C D-penicillamine
5   62   3090      D         Placebo
6   66   4191      D D-penicillamine
7   81   2540      D D-penicillamine
8   83   4050      C D-penicillamine
9   87    198      D D-penicillamine
10  97    611      D         Placebo
11 103    110      D         Placebo
12 107   3388      C         Placebo
13 121    191      D         Placebo
14 154    140      D D-penicillamine
15 253   1765      C D-penicillamine
```

Most of the values with high cooks distance are between 19 and 97 as the Patient ID number, is there something different about the first 100 patients compared to the rest? Now I will use the cooks distance to

make a graph of the high influence points in a fitted vs residuals plot.

```
threshold <- 4 / length(cooksd)

outliers <- cooksd > threshold

residuals_df <- data.frame(
  FittedValues = fitted(lm1),
  Residuals = residuals(lm1),
  HighInfluence = outliers
)

ggplot(residuals_df, aes(x = FittedValues, y = Residuals)) +
  geom_point(aes(color = HighInfluence), alpha = 0.5) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "blue") +
  scale_color_manual(values = c("black", "red")) +
  labs(title = "Residuals vs Fitted Plot with High Influence Points",
       x = "Fitted Values",
       y = "Residuals",
       color = "High Influence") +
  theme_minimal()
```




This plot identifies some high influence points that could be impacting the data, so I'm gonna remove these points and run a new regression with the cleaned data to try and improve the regression.

```
cho_cleaned <- cho[!outliers,]
cho_cleaned <- cho_cleaned %>%
  drop_na()
lm2 <- lm(n_days ~. -id, data = cho_cleaned)
summary(lm1)
```

```
Call:
lm(formula = n_days ~ . - id, data = cho_filtered)

Residuals:
     Min       1Q   Median       3Q      Max 
-2409.79  -598.74   -31.66   554.74  2300.43 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)         -1.992e+03  1.095e+03  -1.818 0.070308 .  
statusD             -5.078e+02  1.514e+02  -3.353 0.000932 ***
statusCL            -5.858e+02  2.271e+02  -2.579 0.010503 *  
drugD-penicillamine  3.278e+00  1.111e+02   0.029 0.976497    
age                 -6.172e-03  1.663e-02  -0.371 0.710961    
sexM                 7.846e+01  1.891e+02   0.415 0.678601    
ascitesN            -5.772e+01  3.067e+02  -0.188 0.850907    
hepatomegalyN       -1.934e-01  1.290e+02  -0.001 0.998805    
spidersN             3.066e+01  1.386e+02   0.221 0.825152    
edemaN               3.964e+02  3.312e+02   1.197 0.232641    
edemaS               1.164e+02  3.497e+02   0.333 0.739521    
bilirubin           -7.784e+01  2.872e+01  -2.710 0.007213 ** 
cholesterol          1.324e-02  4.967e-01   0.027 0.978759    
albumin              5.552e+02  1.618e+02   3.432 0.000707 ***
copper              -1.994e+00  8.213e-01  -2.428 0.015941 *  
alk_phos             1.287e-01  9.409e-02   1.368 0.172513    
sgot                 3.519e-01  1.401e+00   0.251 0.801939    
tryglicerides        1.528e+00  1.138e+00   1.343 0.180685    
platelets            7.756e-01  6.735e-01   1.151 0.250706    
prothrombin          2.086e+02  6.802e+01   3.067 0.002415 ** 
stage2              -3.487e+02  2.738e+02  -1.274 0.204043    
stage3              -3.987e+02  2.710e+02  -1.471 0.142513    
stage4              -6.055e+02  2.931e+02  -2.066 0.039940 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 849.4 on 236 degrees of freedom
Multiple R-squared:  0.4092,    Adjusted R-squared:  0.3541 
F-statistic:  7.43 on 22 and 236 DF,  p-value: < 2.2e-16
```

```
summary(lm2)
```

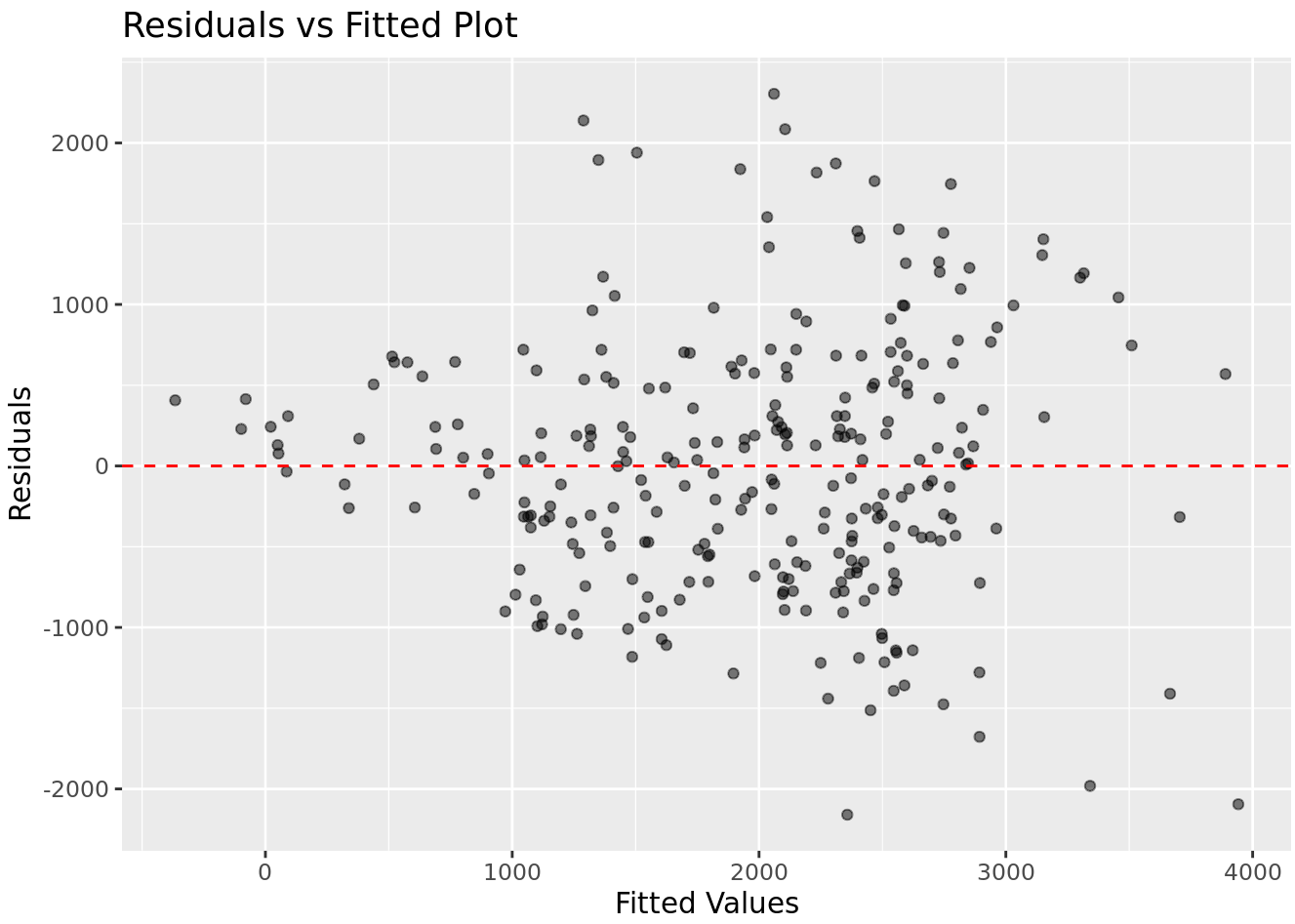
```
Call:
lm(formula = n_days ~ . - id, data = cho_cleaned)

Residuals:
     Min       1Q   Median       3Q      Max 
-2159.65  -539.68     8.34   514.19  2304.33 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)         -998.38631  964.88061  -1.035 0.301766    
statusD             -637.98478  135.27334  -4.716 3.94e-06 ***
statusCL            -634.04695  218.74681  -2.899 0.004071 ** 
drugD-penicillamine   32.46150  103.47196   0.314 0.753985    
age                   -0.00521    0.01533  -0.340 0.734188    
sexM                 206.38705  176.36986   1.170 0.243002    
ascitesN             227.85005  263.98096   0.863 0.388867    
hepatomegalyN         34.98434  121.77250   0.287 0.774119    
spidersN               1.23816  124.97060   0.010 0.992103    
edemaN               191.92093  280.56329   0.684 0.494554    
edemaS               122.71609  290.14982   0.423 0.672691    
bilirubin            -46.64807   16.53524  -2.821 0.005158 ** 
cholesterol           -0.07490    0.25834  -0.290 0.772105    
albumin              553.31390  146.95818   3.765 0.000206 ***
copper                -1.69089    0.71926  -2.351 0.019483 *  
alk_phos               0.13380    0.02561   5.224 3.61e-07 ***
sgot                   0.41140    1.04931   0.392 0.695334    
tryglicerides          1.25493    0.88880   1.412 0.159172    
platelets              0.29948    0.58631   0.511 0.609938    
prothrombin          110.49431   61.13512   1.807 0.071868 .  
stage2              -217.57303  246.28813  -0.883 0.377838    
stage3              -399.38948  241.20043  -1.656 0.098970 .  
stage4              -579.00203  260.75265  -2.221 0.027254 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 827.8 on 258 degrees of freedom
Multiple R-squared:  0.4849,    Adjusted R-squared:  0.441 
F-statistic: 11.04 on 22 and 258 DF,  p-value: < 2.2e-16
```

From this regression we can see that the degrees of freedom have increased from 236 to 258 and the residual standard error has decreased from 849 to 827. We can also see that the statistical significance of the variables has changed with alk_phos becoming significant to the 1% level now. This regression looks more accurate than the first one so removing the high influence points seems to have helped. The estimates have less variance in their numbers no longer needing scientific notion in the second regression.

## Question 4

Now I will check that this cleaned data passes the assumptions using several different diagnostic plots to check the Linearity, Homosexuality and Normality of the error terms.

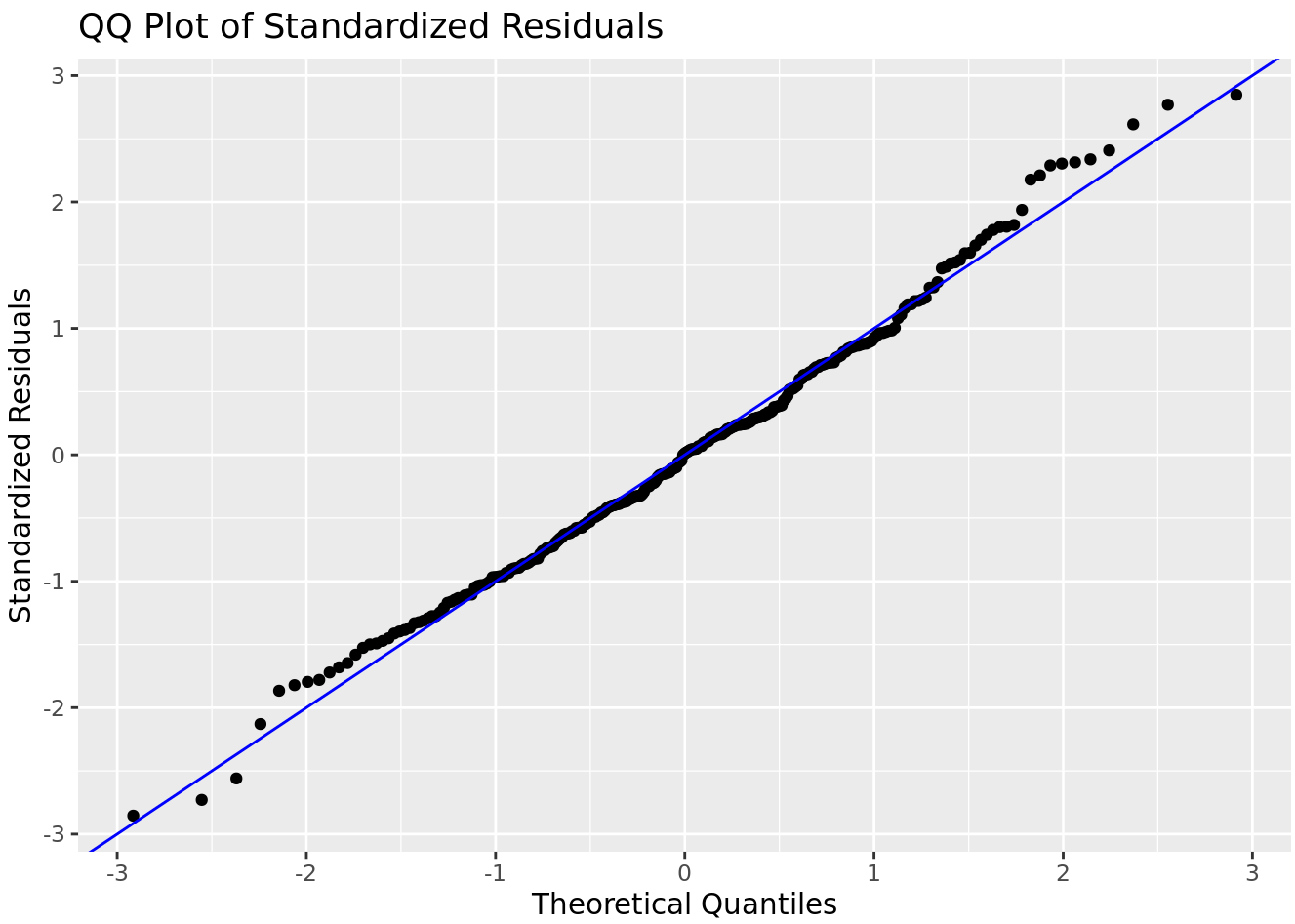
```
cho_cleaned$residuals <- residuals(lm2)
cho_cleaned$fitted_values <- fitted(lm2)
cho_cleaned %>%
  ggplot(aes(x = fitted_values, y = residuals)) +
  geom_point(alpha = 0.5) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residuals vs Fitted Plot",
       x = "Fitted Values",
       y = "Residuals")
```




Residuals seem to be symmetrically distributed around the center line, but the linear relationship is valid, so linearity is good. The variance seems to increase with the fitted values which violates homoscedasticity.

```
standard_resids <- rstandard(lm2)

qq_data <- qqnorm(standard_resids, plot.it = FALSE)

qq_df <- data.frame(Theoretical = qq_data$x, StandardizedResiduals = qq_data$y)

ggplot(qq_df, aes(x = Theoretical, y = StandardizedResiduals)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "blue") +
  labs(title = "QQ Plot of Standardized Residuals",
```

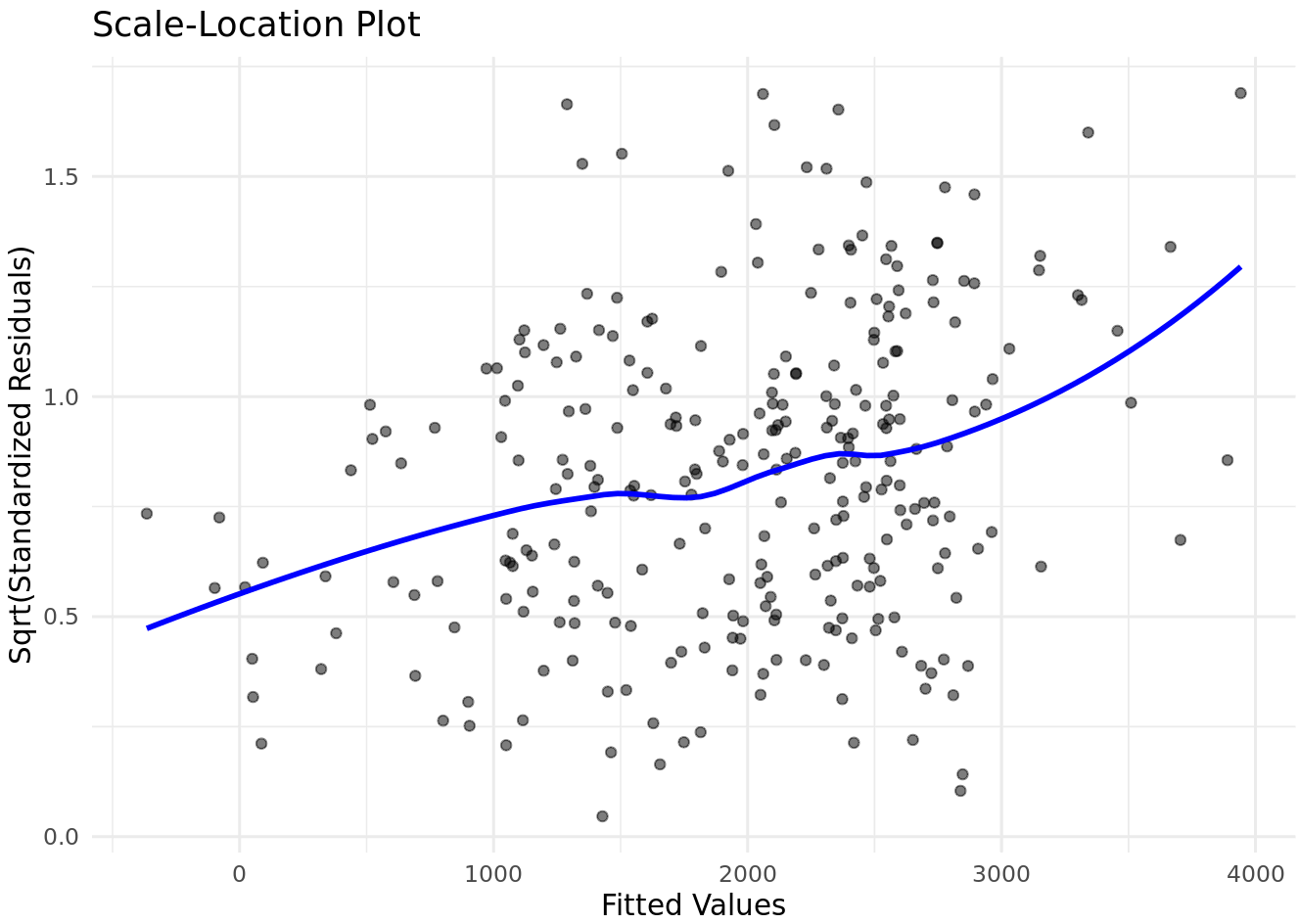
```
       x = "Theoretical Quantiles",
       y = "Standardized Residuals")
```




From the QQ plot we can see that the residuals are normally distributed, with the exception of a few outliers near the tails, but overall seems to be fulfill the normality of error terms requirment.

```
sqrt_std_residuals <- sqrt(abs(rstandard(lm2)))

scale_location_df <- data.frame(
  FittedValues = fitted(lm2),
  SqrtStdResiduals = sqrt_std_residuals
)

ggplot(scale_location_df, aes(x = FittedValues, y = SqrtStdResiduals)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", color = "blue", se = FALSE) +
  labs(title = "Scale-Location Plot",
       x = "Fitted Values",
       y = "Sqrt(Standardized Residuals)") +
  theme_minimal()
```

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




There still seems to be a slight heteroscedasticity/variance problem with the data increasing in variance with the fitted values. So the constant variance for error terms assumption is violated.

## Question 5

```
lm2 %>%
 tidy() %>%
 select(term, estimate, std.error, statistic, p.value)
```

```
# A tibble: 23 × 5
   term                  estimate std.error statistic    p.value
   <chr>                    <dbl>     <dbl>     <dbl>      <dbl>
 1 (Intercept)          -998.       965.      -1.03    0.302
 2 statusD              -638.       135.      -4.72    0.00000394
 3 statusCL             -634.       219.      -2.90    0.00407
 4 drugD-penicillamine    32.5      103.       0.314   0.754
 5 age                    -0.00521    0.0153  -0.340   0.734
 6 sexM                  206.       176.       1.17    0.243
 7 ascitesN              228.       264.       0.863   0.389
 8 hepatomegalyN          35.0      122.       0.287   0.774
 9 spidersN                1.24     125.       0.00991 0.992
10 edemaN                192.       281.       0.684   0.495
# ℹ 13 more rows
```

The p-value of the drug D-penicillamine variable is 0.75, which is close 1, this means that you can not reject the null hypothesis that D-Pencilliamine is not equal to the baseline (Placebo). That means the effect of the drug increasing patient survival by about 32 days could be attributed to randomness as the drug is not statistically different from the effect of the Placebo.

# Predictive Modeling

## Question 6

```
set.seed(1234)
train_pct <- 0.7
indices <- seq(from = 1, to = nrow(cho_cleaned), by =1)
training_indices <- sample(x = indices, replace = FALSE,
                           size = nrow(cho_cleaned)*train_pct)
cho_train <- cho_cleaned %>%
  slice(training_indices)
cho_test <- cho_cleaned %>%
  slice(-training_indices)

predictions <- predict(lm2, cho_test)

mse_test <- mean((cho_test$n_days - predictions)^2)
print(paste("MSE on the test set is:", mse_test))
```

```
[1] "MSE on the test set is: 741855.59204845"
```

0.8 808599.584 0.75 773683.87 0.7 741855.59 The rule of thumb for train/test split seems to be 80/20 train/test, but in the slides and discussion we used a 70/30 split, so I tested a split of 70, 75, and 80 and 0.7 was had the smallest MSE, so I choose 70/30 as the train test split.

## Question 7

```
forward <- regsubsets(n_days ~ . -id -residuals -fitted_values, data = cho_cleaned, method = "for
```

```
Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
force.in, : 1 linear dependencies found

Reordering variables and trying again:
```

```
summary(forward)
```

```
Subset selection object
Call: regsubsets.formula(n_days ~ . - id - residuals - fitted_values,
    data = cho_cleaned, method = "forward")
23 Variables  (and intercept)
```

```
                     Forced in Forced out
statusD                  FALSE      FALSE
statusCL                 FALSE      FALSE
drugD-penicillamine      FALSE      FALSE
age                      FALSE      FALSE
sexM                     FALSE      FALSE
ascitesN                 FALSE      FALSE
hepatomegalyN            FALSE      FALSE
spidersN                 FALSE      FALSE
edemaN                   FALSE      FALSE
edemaS                   FALSE      FALSE
bilirubin                FALSE      FALSE
cholesterol              FALSE      FALSE
albumin                  FALSE      FALSE
copper                   FALSE      FALSE
alk_phos                 FALSE      FALSE
sgot                     FALSE      FALSE
tryglicerides            FALSE      FALSE
platelets                FALSE      FALSE
prothrombin              FALSE      FALSE
stage2                   FALSE      FALSE
stage3                   FALSE      FALSE
stage4                   FALSE      FALSE
drugUnknown              FALSE      FALSE
1 subsets of each size up to 9
Selection Algorithm: forward
          statusD statusCL drugD-penicillamine drugUnknown age sexM ascitesN
1  ( 1 ) " "     " "      " "                  " "         " " " "  " "
2  ( 1 ) " "     " "      " "                  " "         " " " "  " "
3  ( 1 ) " "     " "      " "                  " "         " " " "  " "
4  ( 1 ) "*"     " "      " "                  " "         " " " "  " "
5  ( 1 ) "*"     "*"      " "                  " "         " " " "  " "
6  ( 1 ) "*"     "*"      " "                  " "         " " " "  " "
7  ( 1 ) "*"     "*"      " "                  " "         " " " "  " "
8  ( 1 ) "*"     "*"      " "                  " "         " " " "  " "
9  ( 1 ) "*"     "*"      " "                  " "         " " " "  " "
          hepatomegalyN spidersN edemaN edemaS bilirubin cholesterol albumin
1  ( 1 ) " "           " "      " "    " "    " "       " "         "*"
2  ( 1 ) " "           " "      " "    " "    "*"       " "         "*"
3  ( 1 ) " "           " "      " "    " "    "*"       " "         "*"
4  ( 1 ) " "           " "      " "    " "    "*"       " "         "*"
5  ( 1 ) " "           " "      " "    " "    "*"       " "         "*"
6  ( 1 ) " "           " "      " "    " "    "*"       " "         "*"
7  ( 1 ) " "           " "      " "    " "    "*"       " "         "*"
8  ( 1 ) " "           " "      " "    " "    "*"       " "         "*"
9  ( 1 ) " "           " "      " "    " "    "*"       " "         "*"
          copper alk_phos sgot tryglicerides platelets prothrombin stage2 stage3
1  ( 1 ) " "    " "      " "  " "           " "       " "         " "    " "
2  ( 1 ) " "    " "      " "  " "           " "       " "         " "    " "
3  ( 1 ) " "    "*"      " "  " "           " "       " "         " "    " "
4  ( 1 ) " "    "*"      " "  " "           " "       " "         " "    " "
```

```
5  ( 1 ) " "    "*"      " "  " "          " "       " "         " "    " "
6  ( 1 ) " "    "*"      " "  " "          " "       " "         " "    " "
7  ( 1 ) "*"    "*"      " "  " "          " "       " "         " "    " "
8  ( 1 ) "*"    "*"      " "  " "          " "       " "         " "    "*"
9  ( 1 ) "*"    "*"      " "  " "          " "       "*"         " "    "*"
         stage4
1  ( 1 ) " "
2  ( 1 ) " "
3  ( 1 ) " "
4  ( 1 ) " "
5  ( 1 ) " "
6  ( 1 ) "*"
7  ( 1 ) "*"
8  ( 1 ) "*"
9  ( 1 ) "*"
```

I did a forward stepwise model to find the best variables to use in a regression now I'll make 7 models and test the RMSEs of each to find the best model to use.

```
MLR_model1 <- lm(n_days ~ albumin, data = cho_train)
predictions1 <- predict(MLR_model1, cho_test)

rmse1 <- sqrt(mean((cho_test$n_days - predictions1)^2))
print(paste("RMSE of the first model is:", rmse1))
```

```
[1] "RMSE of the first model is: 964.02762920201"
```

```
MLR_model2 <- lm(n_days ~ albumin + bilirubin, data = cho_train)
predictions2 <- predict(MLR_model2, cho_test)

rmse2 <- sqrt(mean((cho_test$n_days - predictions2)^2))
print(paste("RMSE of the second model is:", rmse2))
```

```
[1] "RMSE of the second model is: 913.402035287601"
```

```
MLR_model3 <- lm(n_days ~ albumin + bilirubin + alk_phos , data = cho_train)
predictions3 <- predict(MLR_model3, cho_test)

rmse3 <- sqrt(mean((cho_test$n_days - predictions3)^2))
print(paste("RMSE of the third model is:", rmse3))
```

```
[1] "RMSE of the third model is: 968.974101652939"
```

```
MLR_model4 <- lm(n_days ~ albumin + bilirubin + alk_phos + status , data = cho_train)
predictions4 <- predict(MLR_model4, cho_test)

rmse4 <- sqrt(mean((cho_test$n_days - predictions4)^2))
print(paste("RMSE of the fourth model is:", rmse4))
```

```
[1] "RMSE of the fourth model is: 951.157037281773"
```

```
MLR_model5 <- lm(n_days ~ albumin + bilirubin + alk_phos + status + stage , data = cho_train)
predictions5 <- predict(MLR_model5, cho_test)

rmse5 <- sqrt(mean((cho_test$n_days - predictions5)^2))
print(paste("RMSE of the fifth model is:", rmse5))
```

```
[1] "RMSE of the fifth model is: 929.614993132088"
```

```
MLR_model6 <- lm(n_days ~ albumin + bilirubin + alk_phos + status + stage + copper , data = cho_t
predictions6 <- predict(MLR_model6, cho_test)

rmse6 <- sqrt(mean((cho_test$n_days - predictions6)^2))
print(paste("RMSE of the sixth model is:", rmse6))
```

```
[1] "RMSE of the sixth model is: 927.813058391884"
```

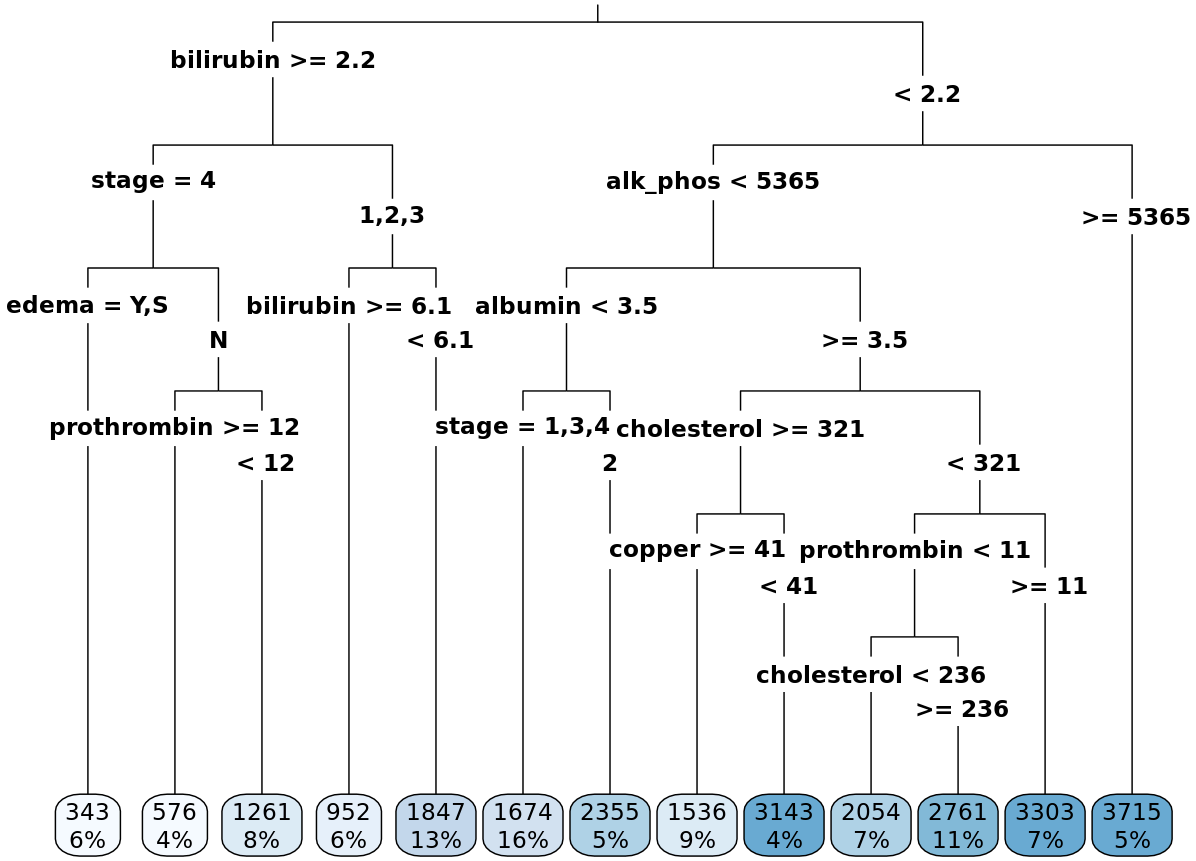
```
MLR_model7 <- lm(n_days ~ albumin + bilirubin + alk_phos + status + stage + copper + prothrombin
predictions7 <- predict(MLR_model7, cho_test)

rmse7 <- sqrt(mean((cho_test$n_days - predictions7)^2))
print(paste("RMSE of the seventh model is:", rmse7))
```

```
[1] "RMSE of the seventh model is: 919.755241317159"
```

We can see that the second model has the best RMSE with the variables albumin and bilirubin. The RMSE increased after including alk_phos then decreased with more variables being added but the second model still remained the best even with the seventh model getting close with 919.75 to the second's 913.4

## Question 8

```
decision_tree <-
  rpart(n_days ~ . -id -residuals -fitted_values, data = cho_train)
rpart.plot(decision_tree, type = 3)
```




Now I'm gonna use printcp to find out more information about the tree and prune as necessary.

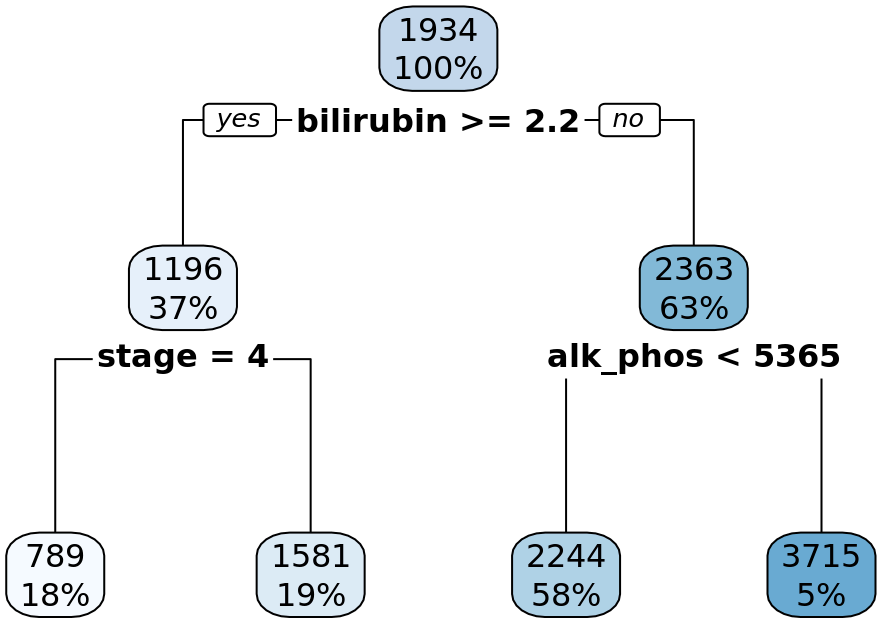
```
printcp(decision_tree)
```

```
Regression tree:
rpart(formula = n_days ~ . - id - residuals - fitted_values,
    data = cho_train)

Variables actually used in tree construction:
[1] albumin     alk_phos    bilirubin   cholesterol copper      edema
[7] prothrombin stage

Root node error: 238187923/196 = 1215245

n= 196

         CP nsplit rel error  xerror     xstd
1  0.260358      0   1.00000 1.00788 0.090044
2  0.083480      1   0.73964 0.79850 0.078302
3  0.047289      2   0.65616 0.88528 0.086444
4  0.045635      3   0.60887 0.87166 0.090810
5  0.026037      6   0.47197 0.84742 0.095314
6  0.026026      7   0.44593 0.89149 0.101642
```

```
7  0.016810      8   0.41991 0.93213 0.106644
8  0.015312      9   0.40310 0.93867 0.108390
9  0.014710     10   0.38778 0.94459 0.109088
10 0.010274     11   0.37307 0.95733 0.110288
11 0.010000     12   0.36280 0.97465 0.112759
```

We can see that the xerror starts increasing again after the 3rd split, so I will set the cp = 0.045635

```
tree2 <- rpart(n_days ~ . -id -residuals -fitted_values, data = cho_train,
                    cp = 0.045635)
rpart.plot(tree2)
```




```
printcp(tree2)
```

```
Regression tree:
rpart(formula = n_days ~ . - id - residuals - fitted_values,
    data = cho_train, cp = 0.045635)

Variables actually used in tree construction:
[1] alk_phos  bilirubin stage

Root node error: 238187923/196 = 1215245
```

```
n= 196

        CP nsplit rel error  xerror     xstd
1 0.260358      0   1.00000 1.00492 0.089524
2 0.083480      1   0.73964 0.80250 0.078483
3 0.047289      2   0.65616 0.86006 0.087587
4 0.045635      3   0.60887 0.89684 0.100867
```

Now the important variables are only alk_phos, bilirubin and stage = 4

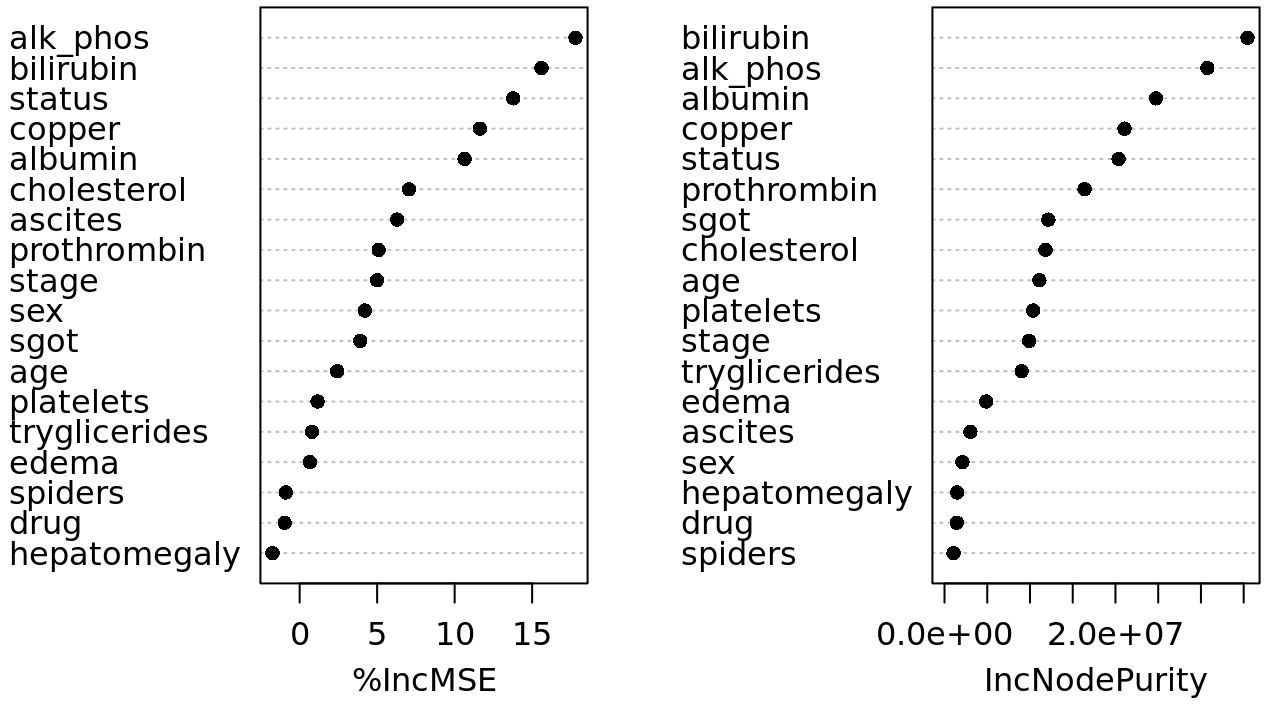
```
predictiontree <- predict(object = tree2, newdata = cho_test)
rmseTree <- sqrt(mean((cho_test$n_days - predictiontree)^2))
print(rmseTree)
```

```
[1] 1022.586
```

```
random_forest <-
  randomForest(n_days ~ . -id -residuals -fitted_values, data = cho_train,
               cutoff = c(0.5, 0.5),
               importance = TRUE)
random_forest
```

```
Call:
 randomForest(formula = n_days ~ . - id - residuals - fitted_values,      data = cho_train,
cutoff = c(0.5, 0.5), importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 6

          Mean of squared residuals: 719961.5
                    % Var explained: 40.76
```

```
varImpPlot(random_forest,
           main = "Variable Importance Plot",
           pch = 16)
```




This chart ranks the variables by how much they increase the MSE and how homogeneous each node is we can see that alk_phos and bilirubin remain important variables.

```
predictionrandom <- predict(object = random_forest, newdata = cho_test)
rmseRandomTree <- sqrt(mean((cho_test$n_days - predictionrandom)^2))
print(rmseRandomTree)
```

```
[1] 856.7104
```

## Question 9

```
final_table <- data.frame(Type = c("Linear Regression", "Regression Trees", "Random Forest"),
                          Metrics = c("Test Prop = 0.7","CP = 0.045635", "Variance Explained: 42.2
                          RMSE = c(rmse2, rmseTree, rmseRandomTree))
final_table
```

```
                Type                   Metrics      RMSE
1 Linear Regression           Test Prop = 0.7  913.4020
2  Regression Trees             CP = 0.045635 1022.5859
3     Random Forest Variance Explained: 42.2%  856.7104
```

I would choose the Linear Regression model out of these three, even though the random forest has a lower RMSE less than half of the Variance is explained, which makes me believe that the Linear Regression model is more robust.

# Next Steps

## Question 1

One aspect I could have really improved on is the EDA and trying to filter the data, I couldn't figure out how to replace the NA values in the dataset, so I just dropped NA which probably isn't the best thing to do to find the best results. I should have started earlier so I would have time to fix this problem in office hours.

## Question 2

Two future work ideas related to this project that could be interesting is analyzing how the study was run to figure out why patients with IDs before 100 had such high influence points and why the patients with IDs after 312 were listed as NA for receiving a drug. Another thing would be if he drug isn't effective, but we can see in the data that Albumin levels and the levels of other chemicals are correlated with surviving more days. It would be interesting to understand why this is the case and if their was a new drug that could increase these levels leading to better patient outcomes.

# Sources

Clark, T. G., Bradburn, M. J., Love, S. B., & Altman, D. G. (2003, July 21). Survival analysis part I: Basic concepts and first analyses. British journal of cancer. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2394262/

Mayo Foundation for Medical Education and Research. (2023, November 14). Primary biliary cholangitis. Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/primary-biliary-cholangitis/symptoms-causes/syc-20376874

U.S. Department of Health and Human Services. (n.d.). Definition & Facts of primary biliary cholangitis (primary biliary cirrhosis) - NIDDK. National Institute of Diabetes and Digestive and Kidney Diseases. https://www.niddk.nih.gov/health-information/liver-disease/primary-biliary-cholangitis/definition-facts