# Final Project: Data Analysis on Cholangitis

Morgan Rhee

December 15, 2021

## Visualization

### 1. Importing the Data

To analyze the data, we must first import the data and take a look at the data given. After looking at the data, a total of 106 patients did not give their consent to the randomization process, as well as a group of 25 patients who received a liver transplant. However, we will still include the 19 who gave consent and had a liver transplant due to the nature of its influence to other variables. Thus, the group of patients that denied consent received neither the drug nor the placebo (drug == "NA"), indicating that we should remove these 106 individuals, as well as additional NA values that will affect the analysis and regressions we will perform..

```
cholang_dat <- read.csv("cholangitis.csv")
cholang_dat <- na.omit(subset(cholang_dat, drug != "NA"))
```

By looking at the first 6 columns, we can see that there is a total of 20 variables, which range from being either continuous or categorical variables. Thus, we should convert all categorical variables to factors in order to effectively perform statistical modeling.

```
cholang_dat$status <- factor(cholang_dat$status)
cholang_dat$drug <- factor(cholang_dat$drug)
cholang_dat$sex <- factor(cholang_dat$sex)
cholang_dat$ascites <- factor(cholang_dat$ascites)
cholang_dat$hepatomegaly <- factor(cholang_dat$hepatomegaly)
cholang_dat$spiders <- factor(cholang_dat$spiders)
cholang_dat$edema <- factor(cholang_dat$edema)
cholang_dat$stage <- factor(cholang_dat$stage)
head(cholang_dat)
```

```
##   id n_days status           drug   age sex ascites hepatomegaly spiders edema
## 1  1    400      D D-penicillamine 21464   F       Y            Y       Y     Y
## 2  2   4500      C D-penicillamine 20617   F       N            Y       Y     N
## 3  3   1012      D D-penicillamine 25594   M       N            N       N     S
## 4  4   1925      D D-penicillamine 19994   F       N            Y       Y     S
## 5  5   1504     CL         Placebo 13918   F       N            Y       Y     N
## 6  6   2503      D         Placebo 24201   F       N            Y       N     N
##   bilirubin cholesterol albumin copper alk_phos   sgot tryglicerides platelets
## 1      14.5         261    2.60    156   1718.0 137.95           172       190
## 2       1.1         302    4.14     54   7394.8 113.52            88       221
## 3       1.4         176    3.48    210    516.0  96.10            55       151
## 4       1.8         244    2.54     64   6121.8  60.63            92       183
```

```
## 5       3.4         279    3.53    143    671.0 113.15            72       136
## 6       0.8         248    3.98     50    944.0  93.00            63       361
##   prothrombin stage
## 1         12.2     4
## 2         10.6     3
## 3         12.0     4
## 4         10.3     4
## 5         10.9     3
## 6         11.0     3
```

### 2. Basic Exploratory Data Analysis

We will first investigate the data, performing a general summary and overview of the entire data set using some statistical functions: dim(), str() and summary().

```
dim(cholang_dat)
```

```
## [1] 307  20
```

```
str(cholang_dat)
```

```
## 'data.frame':    307 obs. of  20 variables:
##  $ id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ n_days       : int  400 4500 1012 1925 1504 2503 1832 2466 2400 51 ...
##  $ status       : Factor w/ 3 levels "C","CL","D": 3 1 3 3 2 3 1 3 3 3 ...
##  $ drug         : Factor w/ 2 levels "D-penicillamine",..: 1 1 1 1 2 2 2 2 1 2 ...
##  $ age          : int  21464 20617 25594 19994 13918 24201 20284 19379 15526 25772 ...
##  $ sex          : Factor w/ 2 levels "F","M": 1 1 2 1 1 1 1 1 1 1 ...
##  $ ascites      : Factor w/ 2 levels "N","Y": 2 1 1 1 1 1 1 1 1 2 ...
##  $ hepatomegaly : Factor w/ 2 levels "N","Y": 2 2 1 2 2 2 2 1 1 1 ...
##  $ spiders      : Factor w/ 2 levels "N","Y": 2 2 1 2 2 1 1 1 2 2 ...
##  $ edema        : Factor w/ 3 levels "N","S","Y": 3 1 2 2 1 1 1 1 1 3 ...
##  $ bilirubin    : num  14.5 1.1 1.4 1.8 3.4 0.8 1 0.3 3.2 12.6 ...
##  $ cholesterol  : int  261 302 176 244 279 248 322 280 562 200 ...
##  $ albumin      : num  2.6 4.14 3.48 2.54 3.53 3.98 4.09 4 3.08 2.74 ...
##  $ copper       : int  156 54 210 64 143 50 52 52 79 140 ...
##  $ alk_phos     : num  1718 7395 516 6122 671 ...
##  $ sgot         : num  137.9 113.5 96.1 60.6 113.2 ...
##  $ tryglicerides: int  172 88 55 92 72 63 213 189 88 143 ...
##  $ platelets    : int  190 221 151 183 136 361 204 373 251 302 ...
##  $ prothrombin  : num  12.2 10.6 12 10.3 10.9 11 9.7 11 11 11.5 ...
##  $ stage        : Factor w/ 4 levels "1","2","3","4": 4 3 4 4 3 3 3 3 2 4 ...
##  - attr(*, "na.action")= 'omit' Named int [1:5] 41 106 146 178 190
##   ..- attr(*, "names")= chr [1:5] "41" "106" "146" "178" ...
```

```
summary(cholang_dat)
```

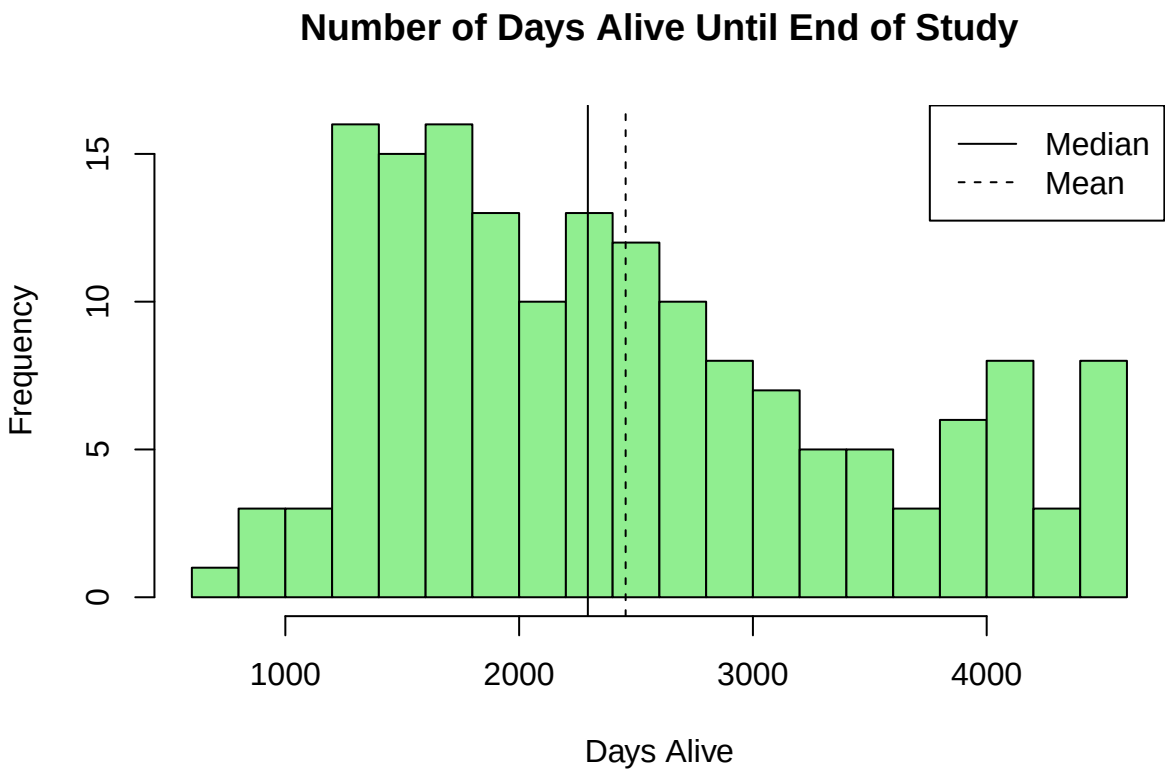
```
##        id            n_days     status                drug          age
##  Min.   :  1.0   Min.   :  41   C :165   D-penicillamine:154   Min.   : 9598
##  1st Qu.: 78.5   1st Qu.:1180   CL: 19   Placebo        :153   1st Qu.:15494
##  Median :157.0   Median :1831   D :123                         Median :18176
```

```
##  Mean   :156.9   Mean   :1999                                    Mean   :18257
##  3rd Qu.:235.5   3rd Qu.:2702                                    3rd Qu.:20696
##  Max.   :312.0   Max.   :4556                                    Max.   :28650
##  sex     ascites hepatomegaly spiders edema     bilirubin       cholesterol
##  F:271   N:284   N:149        N:218   N:259   Min.   : 0.300   Min.   : 120.0
##  M: 36   Y: 23   Y:158        Y: 89   S: 28   1st Qu.: 0.800   1st Qu.: 248.0
##                                       Y: 20   Median : 1.300   Median : 309.0
##                                               Mean   : 3.267   Mean   : 367.4
##                                               3rd Qu.: 3.450   3rd Qu.: 399.5
##                                               Max.   :28.000   Max.   :1775.0
##     albumin          copper          alk_phos          sgot
##  Min.   :1.960   Min.   :  4.00   Min.   :  289   Min.   : 26.35
##  1st Qu.:3.310   1st Qu.: 41.00   1st Qu.:  867   1st Qu.: 80.60
##  Median :3.550   Median : 73.00   Median : 1260   Median :114.70
##  Mean   :3.515   Mean   : 98.06   Mean   : 1995   Mean   :122.48
##  3rd Qu.:3.790   3rd Qu.:123.50   3rd Qu.: 2002   3rd Qu.:151.90
##  Max.   :4.640   Max.   :588.00   Max.   :13862   Max.   :457.25
##  tryglicerides     platelets      prothrombin    stage
##  Min.   : 33.0   Min.   : 62.0   Min.   : 9.00   1: 16
##  1st Qu.: 85.0   1st Qu.:200.0   1st Qu.:10.00   2: 65
##  Median :110.0   Median :258.0   Median :10.60   3:119
##  Mean   :124.4   Mean   :262.3   Mean   :10.73   4:107
##  3rd Qu.:151.5   3rd Qu.:323.0   3rd Qu.:11.10
##  Max.   :598.0   Max.   :563.0   Max.   :17.10
```

Looking at the general summary, we can notice a couple of things from several variables: age, sex, and stage. With age, we can see that the age range of patients vary quite a bit, the youngest being 9598 days old (~ 26 years) and the oldest being 28650 days old (~ 78 years). Furthermore, out of the 307 patients, 271 of the patients were female, indicating a potential biased study against sex due to the much smaller pool of male individuals. Finally, in regards to the histologic stage of disease, a larger number of patients were in the latter portion, having a more serious stage of 3 or 4.

Following this, plotting a histogram for the number of days from registration to being alive is plotted below.

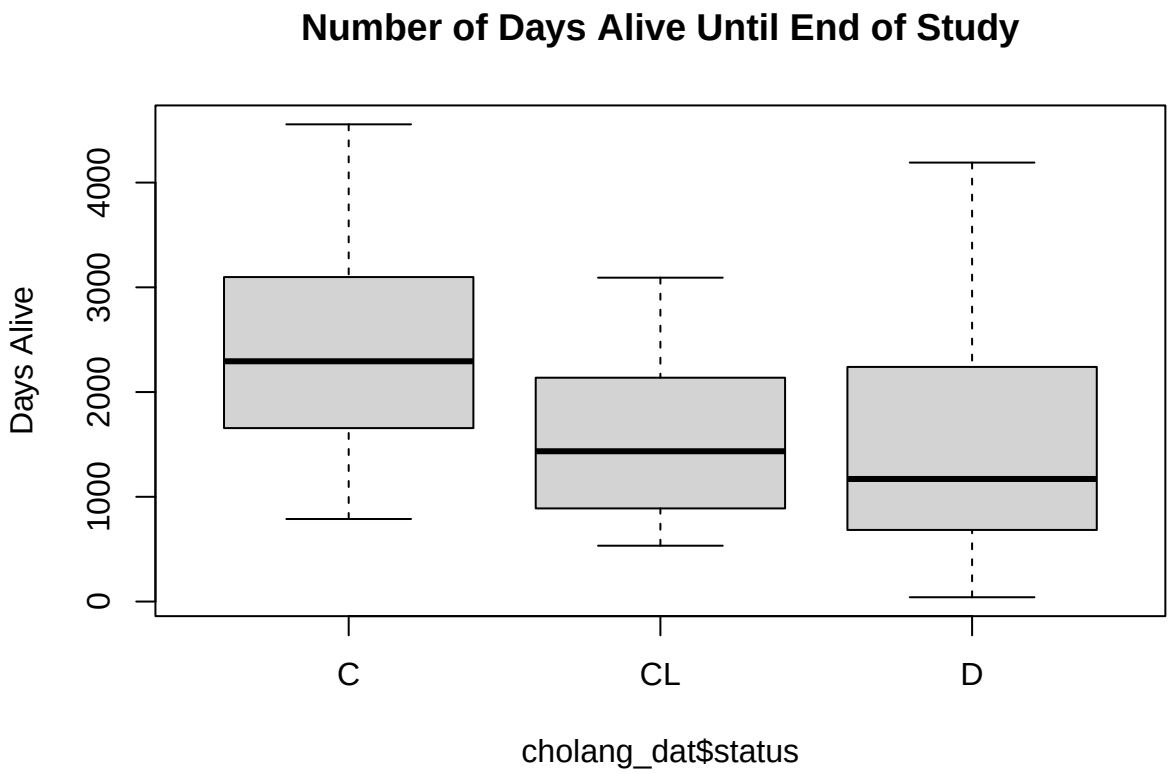
```
cholang_alive <- subset(cholang_dat, status == "C")
hist(cholang_alive$n_days, main = "Number of Days Alive Until End of Study", xlab = "Days Alive", breaks
abline(v = mean(cholang_alive$n_days), lty = "dashed")
abline(v = median(cholang_alive$n_days))
legend("topright", legend = c("Median", "Mean"), lty = c("solid", "dashed"))
```




While the histogram does not follow a normal histogram, we can say that it is roughly centered around the middle, with lower bars on the ends. Seeing as how the median was to the left of the mean, we can come to the conclusion that the data is skewed more to the right with a rather heavier tail. Especially because we have values spreading out to lower than 1000 days alive to as large as about 5000 days alive (given that the mean is about 2500), the histogram is spread relatively far from the mean.

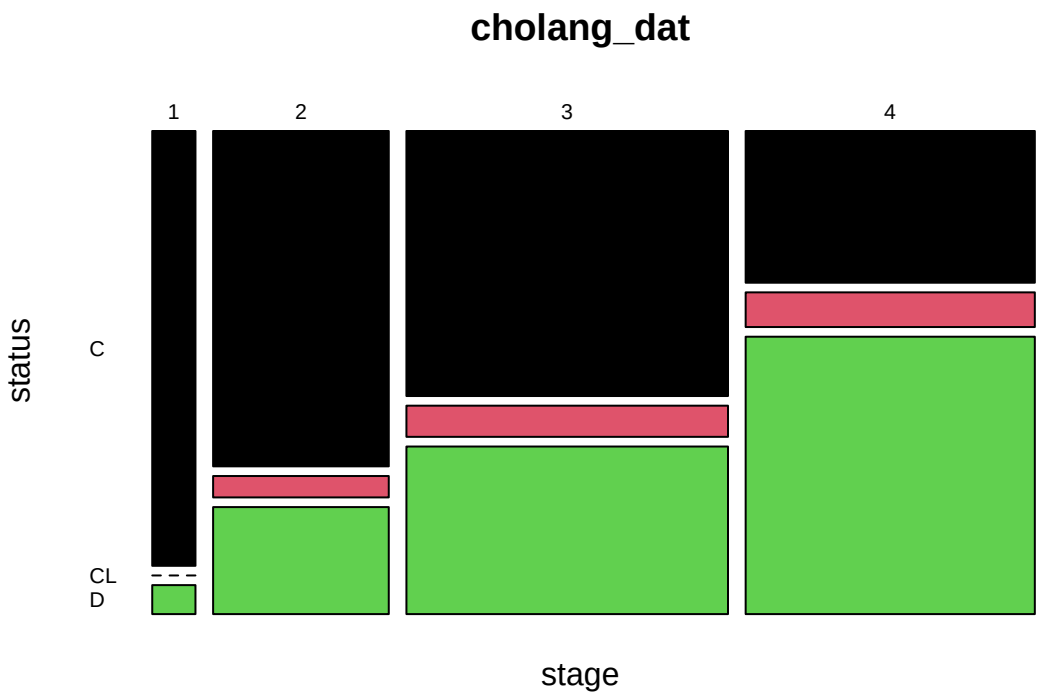
Plotting the same data as a boxplot (with the different statuses) gives us a different approach to visualize such data. However, a key point to see here is that there are no outliers in this plot, indicating that all plots in the this respective data set are within 1.5 of the IQR.

```
par(mfrow = c(1, 1))
boxplot(cholang_dat$n_days ~ cholang_dat$status, main = "Number of Days Alive Until End of Study", breal
```



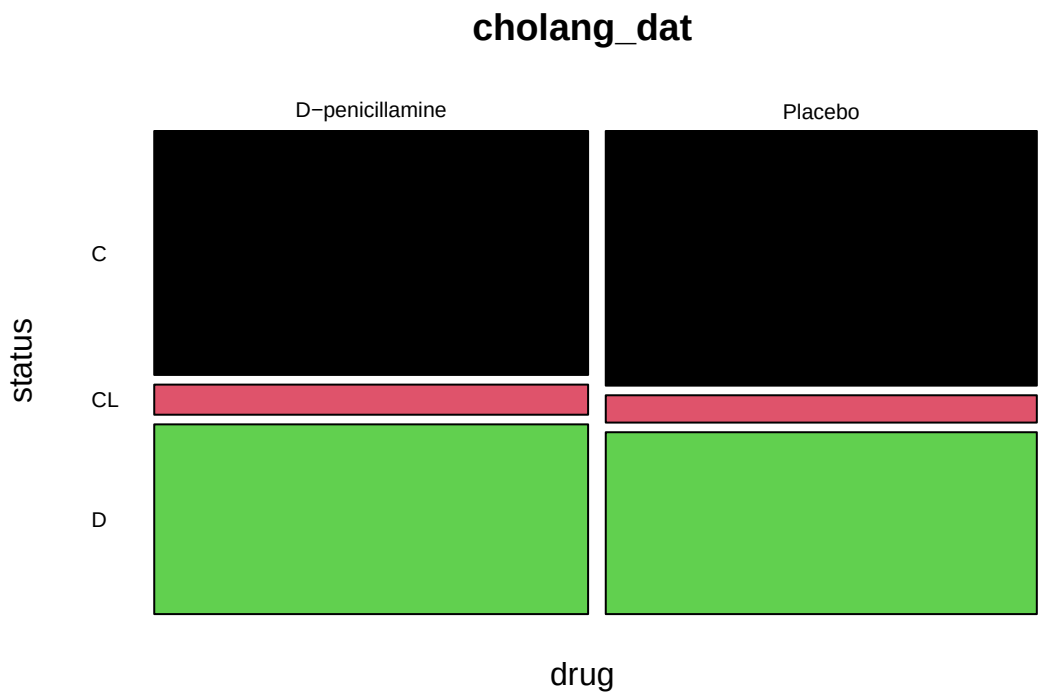

To better analyze two variables in specific, a mosaic plot was used to analyze their relating percentages. In particular, the relationship between status and stage, as well as drug and status were analyzed to not only determine if the drug had a positive effect but also to see the range of the patients in the study.

```
mosaicplot(~ stage + status, data = cholang_dat, las = 1, col = palette())
```




We can see that most admitted patients had a more sever stage of cholangitis (indicated by the wider bars of stage 3 and 4 proportional to the total plot), the percentage of patients surviving at the end of the study decreasing as the stage increased. This was the opposite for patients who died, as the percentage increased as the stage increased.

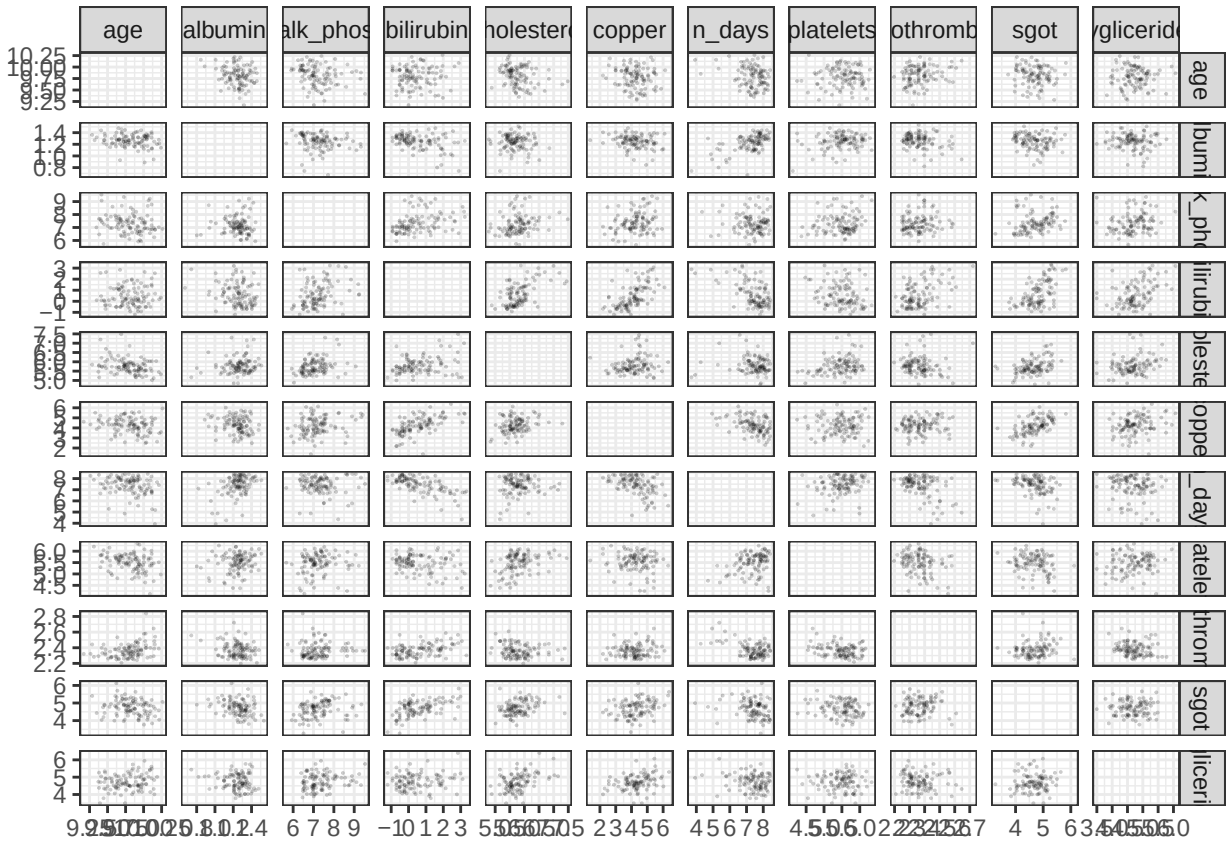
```
mosaicplot(~ drug + status, data = cholang_dat, las = 1, col = palette())
```




Since the number of patients who received the drug and the placebo were almost identical, we can see that there is not as much diversity in terms of the bar widths. Even with the statuses, we see that a similarity in percentages across the different statuses. What is interesting to note is that there were slightly more patients who received the placebo and survived than patients who received the actual drug and survived.

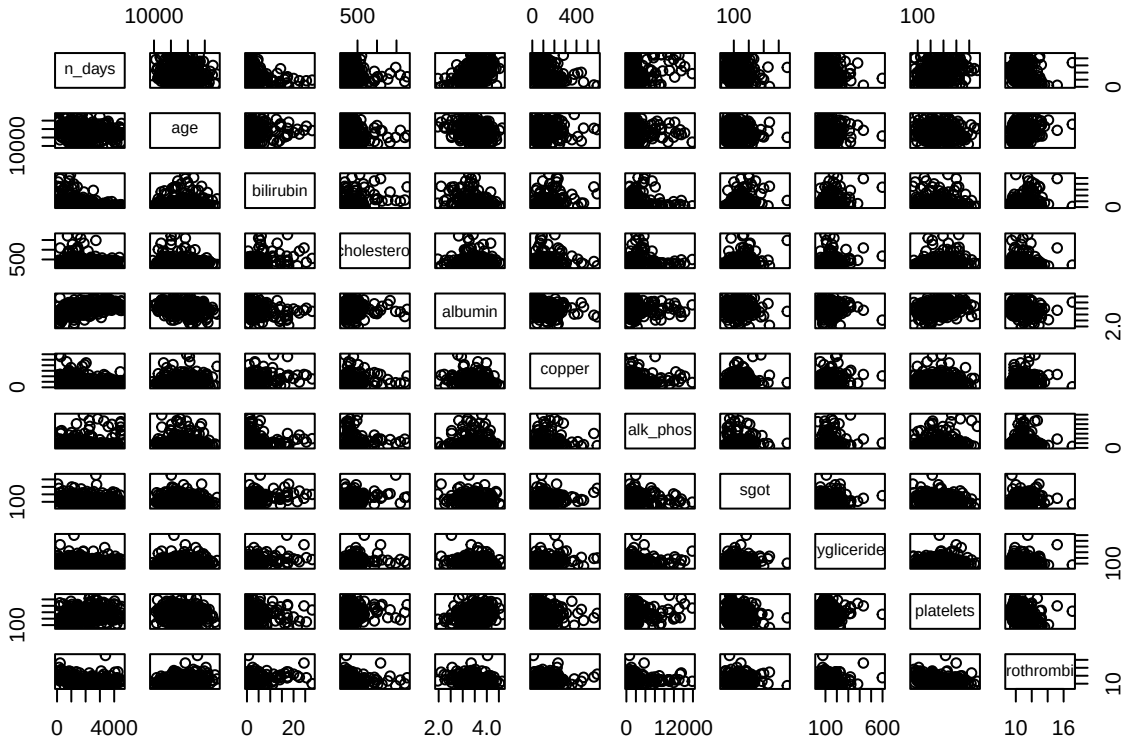
Following the analysis of simply two variables (status and stage, status and drug), demonstrating the relationship of all variables is crucial in exploratory data analysis. A scatter plot between the response variable (n-days) and the continuous variables are plotted below. The log of the data is taken in order to remove any skewness and/or clustering of data that prevents us from analyzing the data properly.

```
new_cholang <- cholang_dat[, -c(1, 3,7:10)]
new_cholang %>% select(where(is.numeric)) -> cholang.select
cholang.log <- log(cholang.select)
apply(cbind(combn(1:11, 2),
combn(11:1, 2)), 2, function(x) {
pair <- cholang.log[,x]
pair$x.name <- names(pair)[1]
pair$y.name <- names(pair)[2]
names(pair)[1:2] <- c("x", "y")
pair[sample(1:nrow(pair), 100),]
}
) %>% do.call(rbind, .) -> cholang.pairs
cholang.pairs %>% ggplot(
aes(x = x, y = y)
) + geom_point(size = 0.01, shape = 1, alpha = 0.2) +
facet_grid(y.name ~ x.name, scales = "free") +
theme_bw() + theme(axis.title = element_blank())
```



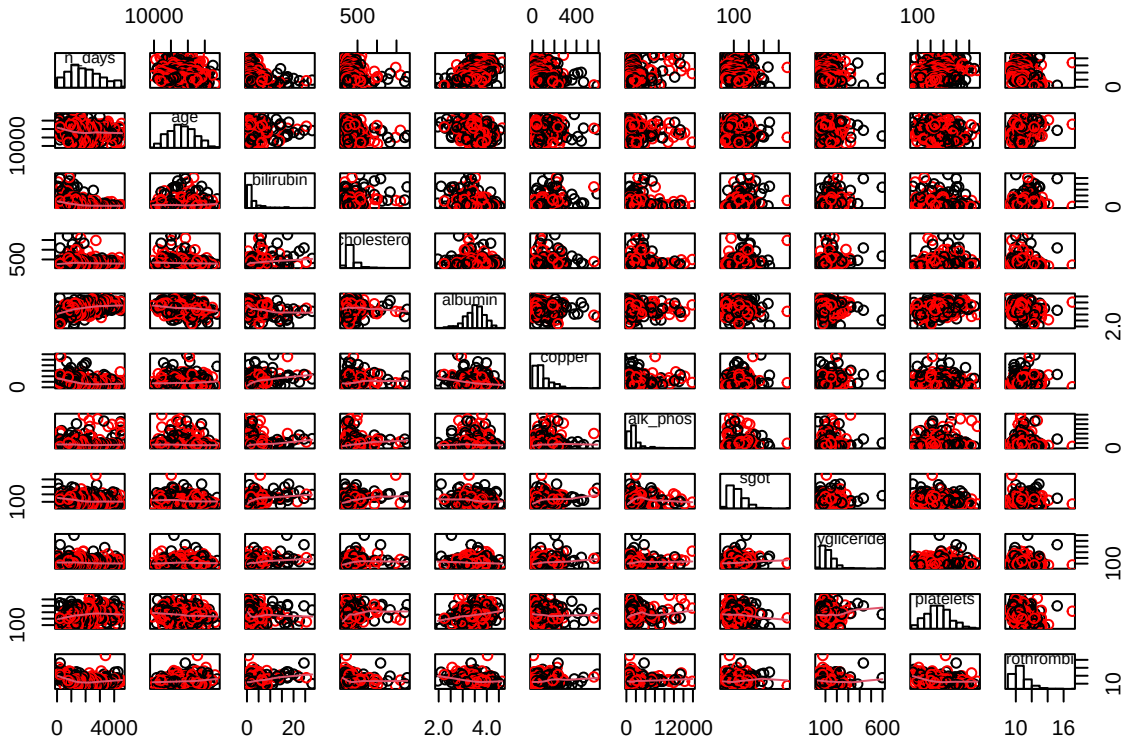

In addition, A pairs plot is plotted below, removing the variables that are in factors since pairs plot only takes in a matrix of numeric values.

```
smallScores <- cholang_dat[,-c(1,3,4,6:10, 20)]
pairs(smallScores)
```



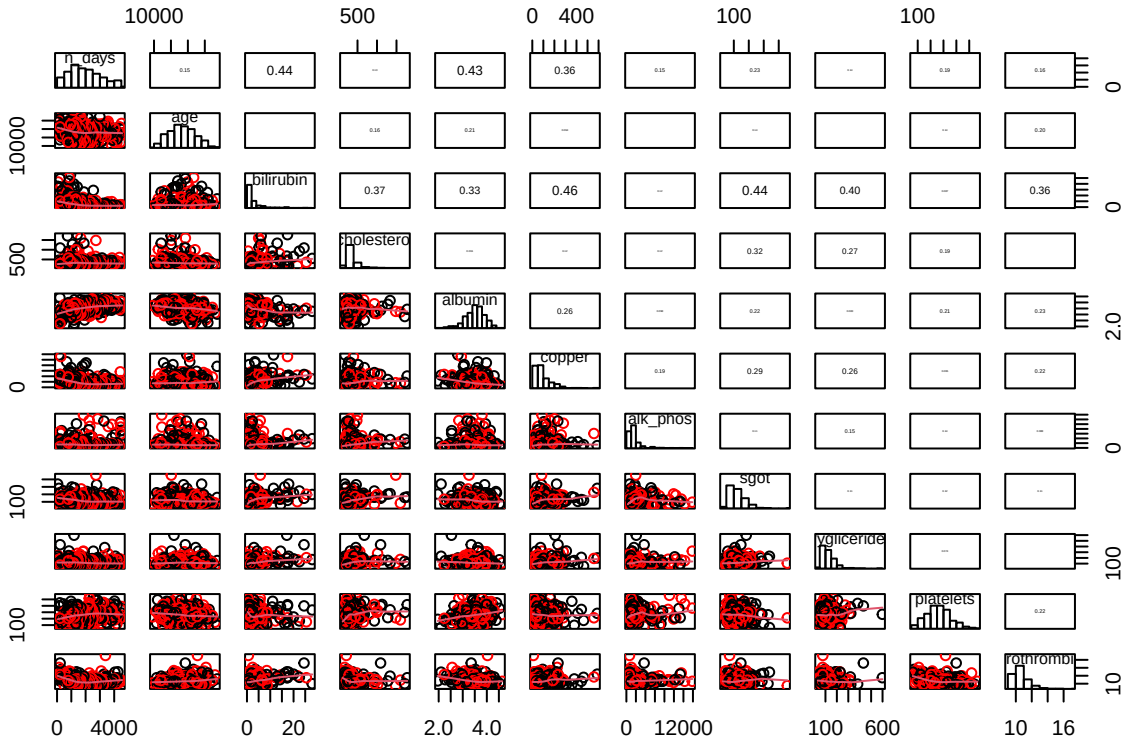

We can see that the plot is not only quite hard to read due to the large number of variables, but also very clustered in various regions. There are no variables that demonstrate a strong linear relationship in the two plots above, with many groups of points covering the majority of the plot and several variables demonstrating heteroskedasticity. We improve this plot below with inputted histograms along the diagonal.

```
panel.hist <- function(x, ...) {
    usr <- par("usr")
    on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5))
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks
    nB <- length(breaks)
    y <- h$counts
    y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y)
}
pairs(smallScores, lower.panel = panel.smooth, col = c("red",
    "black"), diag.panel = panel.hist)
```



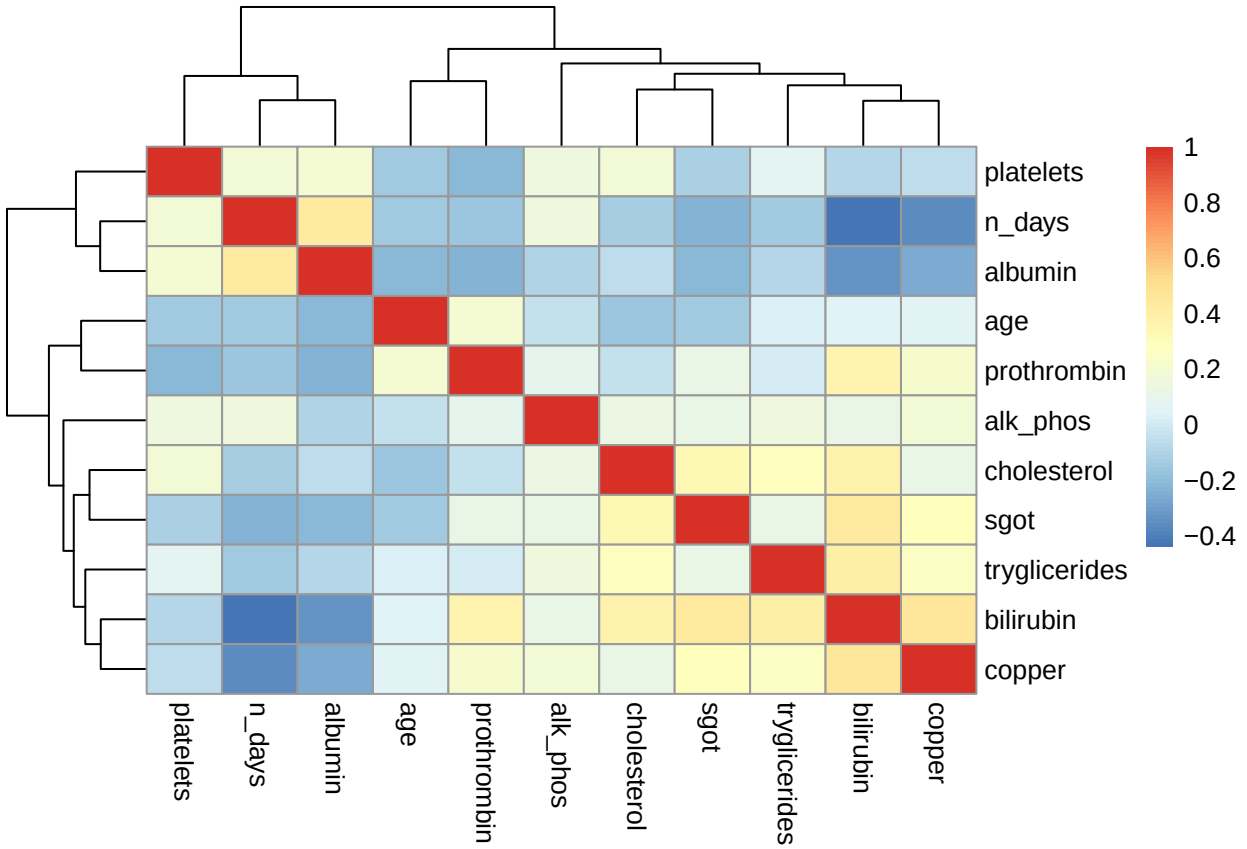

Finally, because there are too many points on the plot (to the point where correlation and proper analysis is difficult to see), the sample correlation is plotted instead in an attempt to better understand and see if high correlations exist between variables.

```
panel.cor <- function(x, y, digits = 2, prefix = "",
    cex.cor, ...) {
    usr <- par("usr")
    on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y, use = "pairwise.complete.obs"))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if (missing(cex.cor))
        cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}
pairs(smallScores, lower.panel = panel.smooth, upper.panel = panel.cor,
    col = c("red", "black"), diag.panel = panel.hist)
```




We can see that while the pairs plot did help in demonstrating if a strong relationship existed between two variables, the abundant number of variables made it hard to clearly see such relationship. Utilizing a heatmap to represent the large data (only with numeric values) with colors might be more efficient.

```
cor_dat <- cor(cholang_dat[,-c(1,3,4,6:10,20)], use="complete.obs")
pheatmap(cor_dat)
```



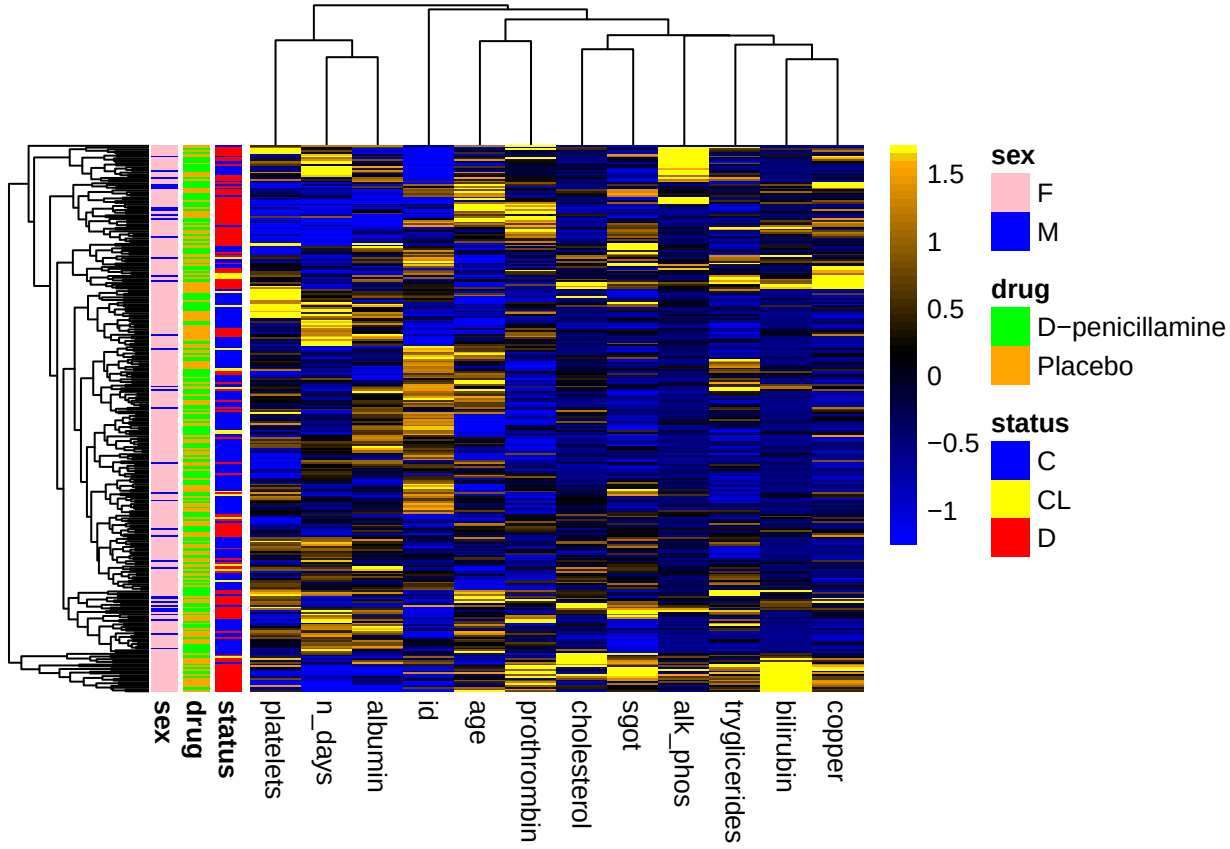

Utilizing a heat map over a pairs plot seems to be more effective in demonstrating correlations. From the plotted data above, we can see that there are stronger correlations near the right bottom, for most of the levels of different substances that are in the body. While they may seem to be correlated in high levels, we cannot forget the fact that they are all in different modes of measurement, ranging from [mg/dl] to [U/liter] to [ug/day]. Regardless, the plotting of the heat map makes for a **better representation** of seeing what variables are most correlated to each other.

We can also plot the data so that outliers (if there are any) do not force the range to cover the whole data by changing the bin range to different quantile settings that will cover some of the larger end values.

```
cholScaled <- scale(cholang_dat[, sapply(cholang_dat, is.numeric)])
cho <- cholang_dat[, !sapply(cholang_dat, is.numeric)]
seqPal2 <- colorRampPalette(c("orange", "black", "blue"))(50)
seqPal2 <- (c("yellow", "gold2", seqPal2))
seqPal2 <- rev(seqPal2)
colstatus <- c("Blue", "Yellow", "Red")
names(colstatus) <- c("C","CL","D")
colsex <- c("pink","blue")
names(colsex) <- c("F", "M")
coldrug <- c("green","orange")
names(coldrug) <- c("D-penicillamine", "Placebo")

pheatmap(cholScaled, scale = "column", color = seqPal2,
breaks = seq(quantile(cholScaled, 0.05), quantile(cholScaled,
0.95), length.out = length(seqPal2) + 1), labels_row = rep("",
nrow(cholang_dat)), annotation_colors = list(status = colstatus,
drug = coldrug, sex = colsex),
```

```
annotation_row = cho[, c(1,2,3)])
```

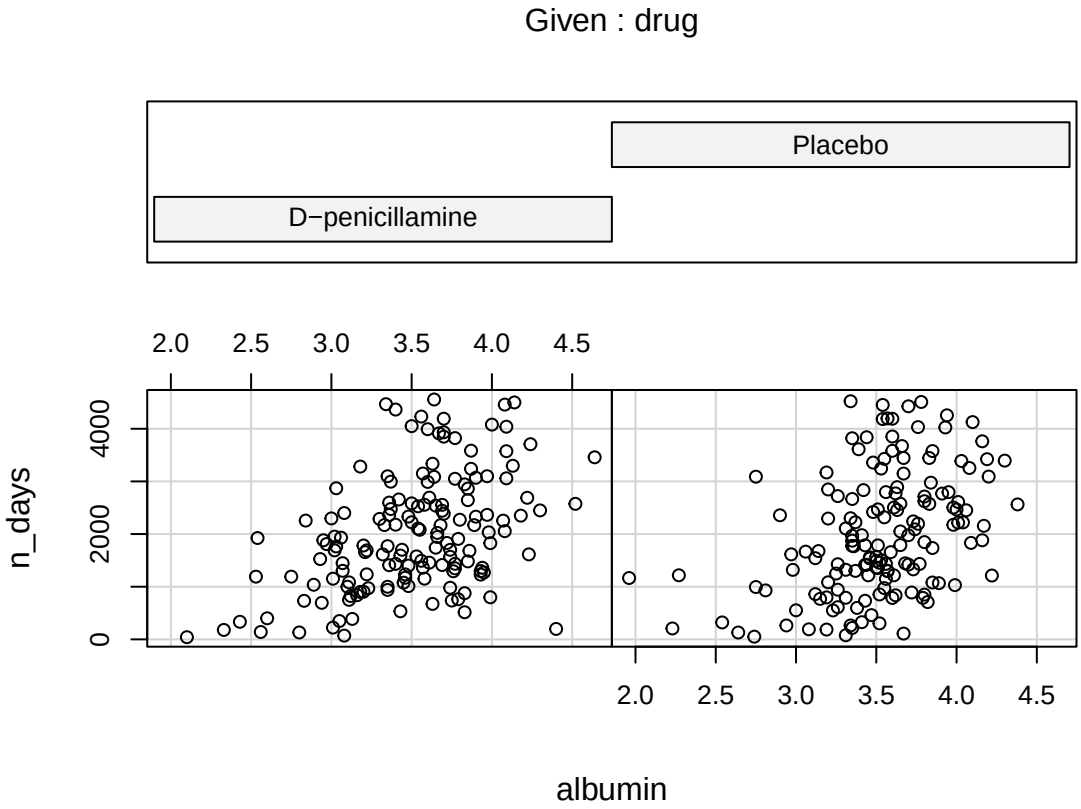



### 3A. Multivariate Regression Analysis

For the multivariate regression, we will analyze the relationship between the response variable— the number of days "n-days"— and several covariates/explanatory variables. We can remove the variable "id" because it will not provide a numeric correlation, given that it is simply a unique identifier for every patient. In addition, status, ascites, hepatomegaly, spiders, edema can be removed since we are looking for a relationship with potential influencing variables of continuous values.

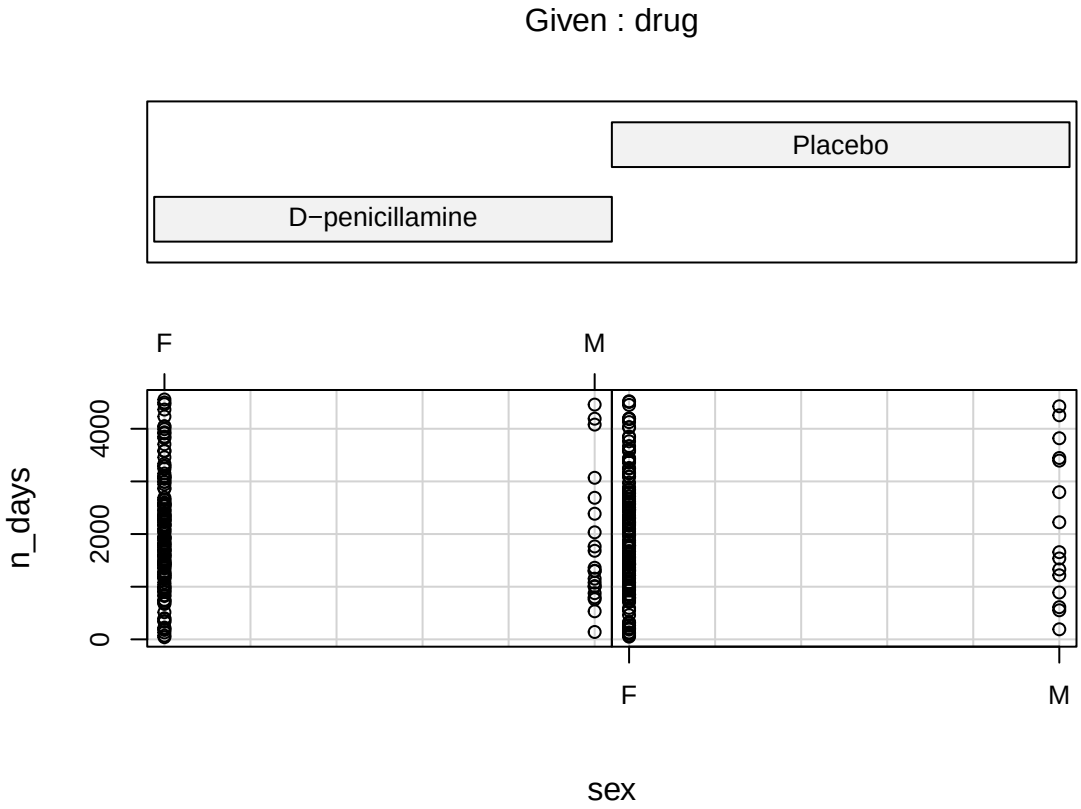
To start, several co-plots were plotted below: relationship of number of days against albumin/sex/stage, conditional on the type of drug given. The three explanatory variables were chosen based on strong or interesting correlations analyzed above.

```
new_cholang <- cholang_dat[, -c(1, 3,7:10)]
coplot(n_days ~ albumin | drug, data = new_cholang)
```



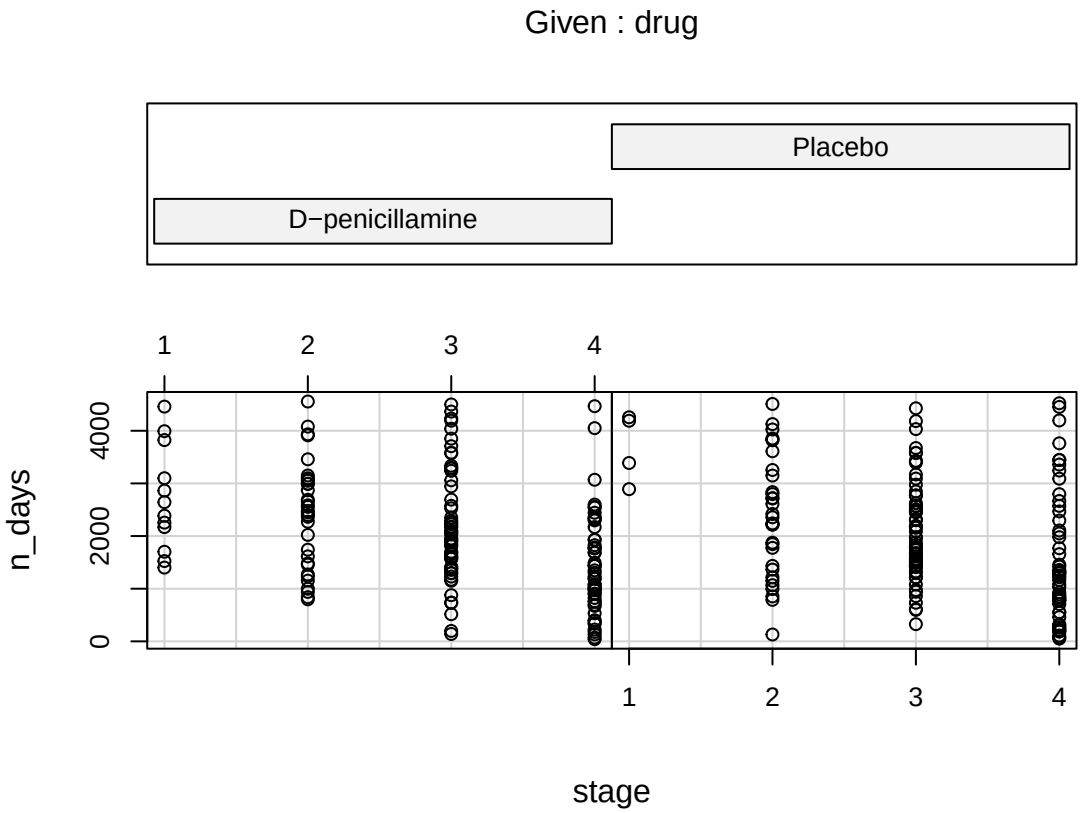

With the protein albumin, both types of drugs injected had a wide yet increasing relationship, as higher levels of albumin indicated generally a trend of longer number of days alive. Both plots demonstrate a heteroskedastic relationship, the clustering of plots spreading out.

```
coplot(n_days ~ sex | drug, data = new_cholang)
```




While there were certainly much more females in the study than males, what is interesting to note about this specific study is that for males, the effect of the drug seemed to quite extreme. While there were more D-penicillamine injected men who passed the 4000 day mark than the placebo male patients, there also respectively seemed to be more male patients who had lower number of days. For women, it was difficult to compare a relationship due to the scattered nature of the plots and the almost uniform range across all number of days from 0 to over 4000.

```
coplot(n_days ~ stage | drug, data = new_cholang)
```




For stage, although the spread of number of days was wide across all stages, we notice that the stage and n_days had a somewhat decreasing, inverse relationship, the number of days decreasing as stage increased for patients. This relationship applicable to both drugs, regardless of what the patient received, and patients with earlier stages of cholangitis has a stronger likelihood of longer days.

Following some initial visualizations, the regression analysis begins by fitting a model between the exploratory and response variables.

```
fit <- lm(n_days ~ ., data = cholang.select)
summary(fit)
```

```
##
## Call:
## lm(formula = n_days ~ ., data = cholang.select)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2914.46  -641.83   -68.04   564.61  2565.74
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.059e+03  9.173e+02  -1.154 0.249282
## age           -1.607e-02  1.457e-02  -1.103 0.270869
## bilirubin     -6.947e+01  1.630e+01  -4.263 2.72e-05 ***
## cholesterol   -2.015e-01  2.658e-01  -0.758 0.449009
## albumin        7.903e+02  1.380e+02   5.728 2.50e-08 ***
## copper        -2.687e+00  7.023e-01  -3.826 0.000159 ***
```

```
## alk_phos       1.241e-01  2.520e-02   4.923 1.42e-06 ***
## sgot           5.674e-02  1.082e+00   0.052 0.958209
## tryglicerides  3.620e-01  9.222e-01   0.393 0.694930
## platelets      8.373e-01  5.898e-01   1.420 0.156773
## prothrombin    5.763e+01  5.874e+01   0.981 0.327292
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 906.8 on 296 degrees of freedom
## Multiple R-squared:  0.3752, Adjusted R-squared:  0.3541
## F-statistic: 17.77 on 10 and 296 DF,  p-value: < 2.2e-16
```

We can see that the significant values are only bilirubin, albumin, copper, and alk_phos.

However, because the values are quite large and ranging diversely, the explanatory variables are all scaled and their respective coefficients from the model are found.

```
scaledchol <- cholang.select
scaledchol[,-1] <- scale(scaledchol[,-1])
ftScale <- lm(n_days ~., data = scaledchol)
summary(ftScale)
```

```
##
## Call:
## lm(formula = n_days ~ ., data = scaledchol)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2914.46  -641.83   -68.04   564.61  2565.74
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1999.068     51.754  38.627  < 2e-16 ***
## age            -61.430     55.687  -1.103 0.270869
## bilirubin     -316.645     74.285  -4.263 2.72e-05 ***
## cholesterol    -46.012     60.696  -0.758 0.449009
## albumin        331.986     57.961   5.728 2.50e-08 ***
## copper        -230.800     60.326  -3.826 0.000159 ***
## alk_phos       267.357     54.313   4.923 1.42e-06 ***
## sgot             3.233     61.652   0.052 0.958209
## tryglicerides   23.105     58.858   0.393 0.694930
## platelets       80.141     56.453   1.420 0.156773
## prothrombin     58.048     59.160   0.981 0.327292
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 906.8 on 296 degrees of freedom
## Multiple R-squared:  0.3752, Adjusted R-squared:  0.3541
## F-statistic: 17.77 on 10 and 296 DF,  p-value: < 2.2e-16
```

```
coef(ftScale)
```

```
##   (Intercept)           age     bilirubin   cholesterol       albumin
```

```
##   1999.068404    -61.430018   -316.644873    -46.012257    331.986122
##        copper      alk_phos          sgot tryglicerides     platelets
##   -230.800082    267.357445      3.233405     23.105217     80.141247
##   prothrombin
##     58.047787
```

The correlation of the variables were also determined below; what we can see is that many of the correlation values are quite low for these values, many in the negative and some of the highest being less than 0.5. Even with these relatively larger correlations, we can see that some of them are not even related to n_days, which is interesting to note as we are using "n_days" as our response variable.

```
cor(scaledchol)
```

```
##                  n_days         age   bilirubin cholesterol     albumin
## n_days         1.0000000 -0.14702221 -0.43987615 -0.13059733  0.43122227
## age           -0.1470222  1.00000000  0.04652375 -0.16344603 -0.21420063
## bilirubin     -0.4398762  0.04652375  1.00000000  0.36734109 -0.33105987
## cholesterol   -0.1305973 -0.16344603  0.36734109  1.00000000 -0.06087991
## albumin        0.4312223 -0.21420063 -0.33105987 -0.06087991  1.00000000
## copper        -0.3600632  0.06395218  0.45719098  0.11955732 -0.25697813
## alk_phos       0.1525268 -0.04533374  0.11623117  0.12325308 -0.09797234
## sgot          -0.2253532 -0.13915733  0.44229073  0.32378736 -0.21994525
## tryglicerides -0.1379835  0.02126683  0.39896247  0.26822709 -0.08255252
## platelets      0.1912610 -0.13898704 -0.08737103  0.19183470  0.20509456
## prothrombin   -0.1635593  0.20413666  0.35853063 -0.04868821 -0.22695563
##                    copper    alk_phos       sgot tryglicerides   platelets
## n_days        -0.36006323  0.15252677 -0.2253532   -0.13798348  0.19126098
## age            0.06395218 -0.04533374 -0.1391573    0.02126683 -0.13898704
## bilirubin      0.45719098  0.11623117  0.4422907    0.39896247 -0.08737103
## cholesterol    0.11955732  0.12325308  0.3237874    0.26822709  0.19183470
## albumin       -0.25697813 -0.09797234 -0.2199453   -0.08255252  0.20509456
## copper         1.00000000  0.18642701  0.2939067    0.25696716 -0.06548098
## alk_phos       0.18642701  1.00000000  0.1121817    0.15207060  0.13691569
## sgot           0.29390667  0.11218166  1.0000000    0.11010193 -0.11730272
## tryglicerides  0.25696716  0.15207060  0.1101019    1.00000000  0.07416257
## platelets     -0.06548098  0.13691569 -0.1173027    0.07416257  1.00000000
## prothrombin    0.21670761  0.08834040  0.1140980    0.01632285 -0.22163583
##                prothrombin
## n_days        -0.16355927
## age            0.20413666
## bilirubin      0.35853063
## cholesterol   -0.04868821
## albumin       -0.22695563
## copper         0.21670761
## alk_phos       0.08834040
## sgot           0.11409800
## tryglicerides  0.01632285
## platelets     -0.22163583
## prothrombin    1.00000000
```

In addition to the fitted models, some statistical information are determined below: fitted values, residuals, RSS, TSS, Rsq to demonstrate how well the model fits with the produced regression as well as to provide a statistical foundation.

```
fitted.values <- ftScale$fitted.values
residuals <- ftScale$residuals
RSS <- sum(residuals^2)
TSS <- sum((cholang.select$n_days - mean(cholang.select$n_days))^2)
Rsq <- 1 - RSS/TSS
#
RSS
```

```
## [1] 243396442
```

```
TSS
```

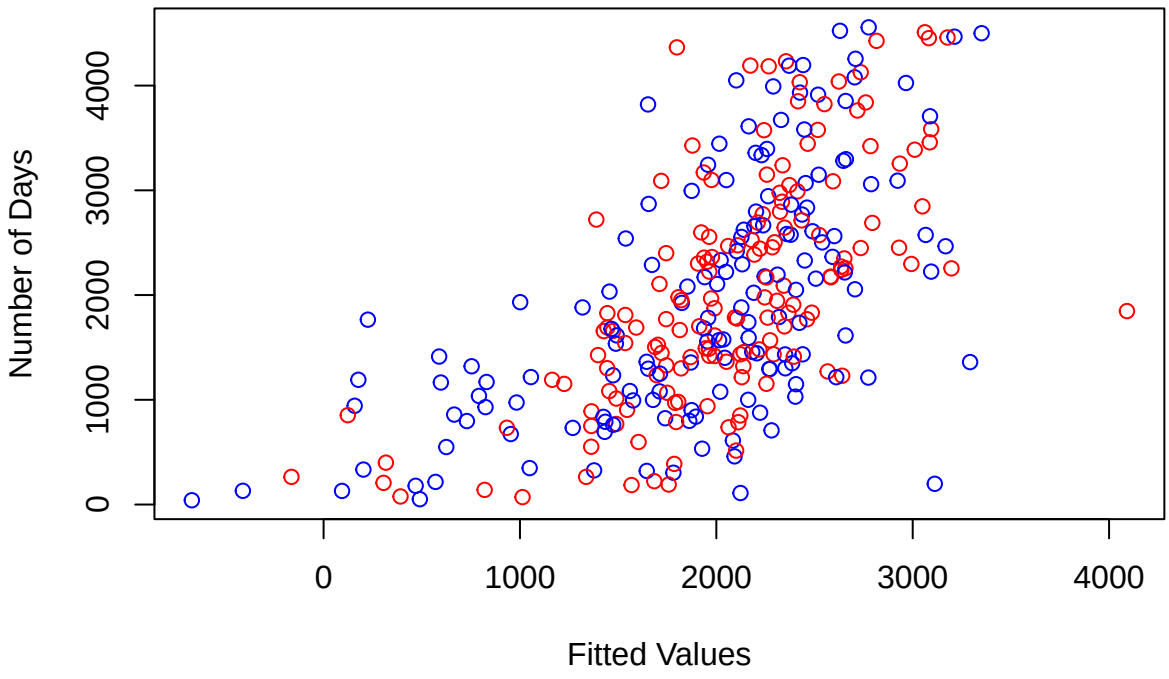
```
## [1] 389535252
```

```
Rsq
```

```
## [1] 0.375162
```

If the regression equation works well, we can expect that when we plot fitted values to the observed values, then the two values would be close to each other. However, given the relatively low correlations, we see that the values to do not fit as close as they could be, and rather demonstrate a general loose correlation. Disregarding some of the outliers, we can not state that the fitted and observed values are not correlated; they are correlated but not as high as other data sets we have previously seen this semester.

```
plot(fitted(fit), cholang.select$n_days, xlab = "Fitted Values", ylab = "Number of Days", col = c("red"
```



### 3B. Variable Selection

In the analysis above, we used all exploratory variables, which could have included unnecessary variables that may have hindered the correlation values and plots. Variable selection will be performed on the cholangitis data set to determine if we can obtain a simpler model that eliminates both the noise and collinearity between variables.

First, regsubsets is used to give the best model with the respective number of k variables we are going to be using from the residual sum of squares.

```
bDays <- regsubsets(n_days ~ ., cholang.select)
summary(bDays)
```

```
## Subset selection object
## Call: regsubsets.formula(n_days ~ ., cholang.select)
## 10 Variables  (and intercept)
##               Forced in Forced out
## age               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
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          age bilirubin cholesterol albumin copper alk_phos sgot tryglicerides
## 1  ( 1 ) " " "*"       " "         " "     " "    " "      " "  " "
## 2  ( 1 ) " " "*"       " "         "*"     " "    " "      " "  " "
## 3  ( 1 ) " " "*"       " "         "*"     " "    "*"      " "  " "
## 4  ( 1 ) " " "*"       " "         "*"     "*"    "*"      " "  " "
## 5  ( 1 ) " " "*"       " "         "*"     "*"    "*"      " "  " "
## 6  ( 1 ) " " "*"       " "         "*"     "*"    "*"      " "  " "
## 7  ( 1 ) "*" "*"       " "         "*"     "*"    "*"      " "  " "
## 8  ( 1 ) "*" "*"       "*"         "*"     "*"    "*"      " "  " "
##          platelets prothrombin
## 1  ( 1 ) " "       " "
## 2  ( 1 ) " "       " "
## 3  ( 1 ) " "       " "
## 4  ( 1 ) " "       " "
## 5  ( 1 ) "*"       " "
## 6  ( 1 ) "*"       "*"
## 7  ( 1 ) "*"       "*"
## 8  ( 1 ) "*"       "*"
```

From this summary, we can interpret the data set as having the best model (of RSS) with bilirubin if the model is with only one explanatory variable. Following, if it is with two, the best are bilirubin and albumin, three with bilirubin, albumin, and alk_phos, and four with bilirubin, albumin, alk_phos, and copper. The summary results in 8 regression models.

In order to find the best k-sized models, we can use the following function to determine when the AIC and

the CV are lowest. In addition to regression subsets found above, we can use cross validation to find the best model.

```
permutation <- sample(1:nrow(cholang.select))
folds <- cut(1:nrow(cholang.select), breaks = 10, labels = FALSE)
predErrorMat <- matrix(nrow = 10, ncol = nrow(summary(bDays)$which))
for (i in 1:10) {
    testIndexes <- which(folds == i, arr.ind = TRUE)
    testData <- cholang.select[permutation, ][testIndexes, ]
    trainData <- cholang.select[permutation, ][-testIndexes,
        ]
    predError <- apply(summary(bDays)$which[, -1], 1,
        function(x) {
            lmObj <- lm(trainData$n_days ~ ., data = trainData[,
                -1][, x, drop = FALSE])
            testPred <- predict(lmObj, newdata = testData[,
                -1])
            mean((testData$n_days - testPred)^2)
        })
    predErrorMat[i, ] <- predError
}


LOOCV <- function(lm) {
    vals <- residuals(lm)/(1 - lm.influence(lm)$hat)
    sum(vals^2)/length(vals)
}
calculateCriterion <- function(x = NULL, y, dataset, lmObj = NULL) {
    sigma2 = summary(lm(y ~ ., data = dataset))$sigma^2
    if (is.null(lmObj))
        lmObj <- lm(y ~ ., data = dataset[, x, drop = FALSE])
    sumlmObj <- summary(lmObj)
    n <- nrow(dataset)
    p <- sum(x)
    RSS <- sumlmObj$sigma^2 * (n - p - 1)
    c(R2 = sumlmObj$r.squared, R2adj = sumlmObj$adj.r.squared,
        `RSS/n` = RSS/n, LOOCV = LOOCV(lmObj), Cp = RSS/n +
            2 * sigma2 * (p + 1)/n, CpAlt = RSS/sigma2 -
            n + 2 * (p + 1), AIC = AIC(lmObj), BIC = BIC(lmObj))
}
critSeat <- apply(summary(bDays)$which[, -1], 1, calculateCriterion,
    y = cholang.select$n_days, dataset = cholang.select[, -1])
critSeat <- t(critSeat)
critSeat <- cbind(critSeat, CV = colMeans(predErrorMat))
critSeat
```

```
##          R2     R2adj     RSS/n     LOOCV        Cp     CpAlt      AIC      BIC
## 1 0.1934910 0.1908467 1023334.4 1036307.7 1034048.3 79.061657 5125.671 5136.852
## 2 0.2850967 0.2803934  907101.1  921629.4  923171.9 37.665978 5090.657 5105.564
## 3 0.3355911 0.3290128  843031.5  867311.0  864459.2 15.745614 5070.169 5088.804
## 4 0.3665817 0.3581920  803709.4  830941.4  830493.9  3.064684 5057.505 5079.866
## 5 0.3698619 0.3593945  799547.3  832695.0  831688.7  3.510766 5057.911 5083.999
## 6 0.3715537 0.3589847  797400.7  836365.4  834899.0  4.709338 5059.086 5088.901
## 7 0.3737220 0.3590600  794649.3  839700.6  837504.6  5.682130 5060.025 5093.566
```

```
## 8 0.3748365 0.3580536  793235.2  842235.5  841447.4  7.154157 5061.478 5098.746
##          CV
## 1 1045888.3
## 2  929842.9
## 3  874099.4
## 4  840200.7
## 5  839602.2
## 6  845534.0
## 7  846651.7
## 8  848598.3
```

Based on the AIC and CV values, we can see that the best model is that of size 4.

Just in case, we can confirm which is the best model by determining what the best model is will be with a stepwise regression method, where we will add/remove a variable until we do not get any further improvements.

```
stepDay <- step(ftScale, trace = 0, direction = "both")
stepDay
```

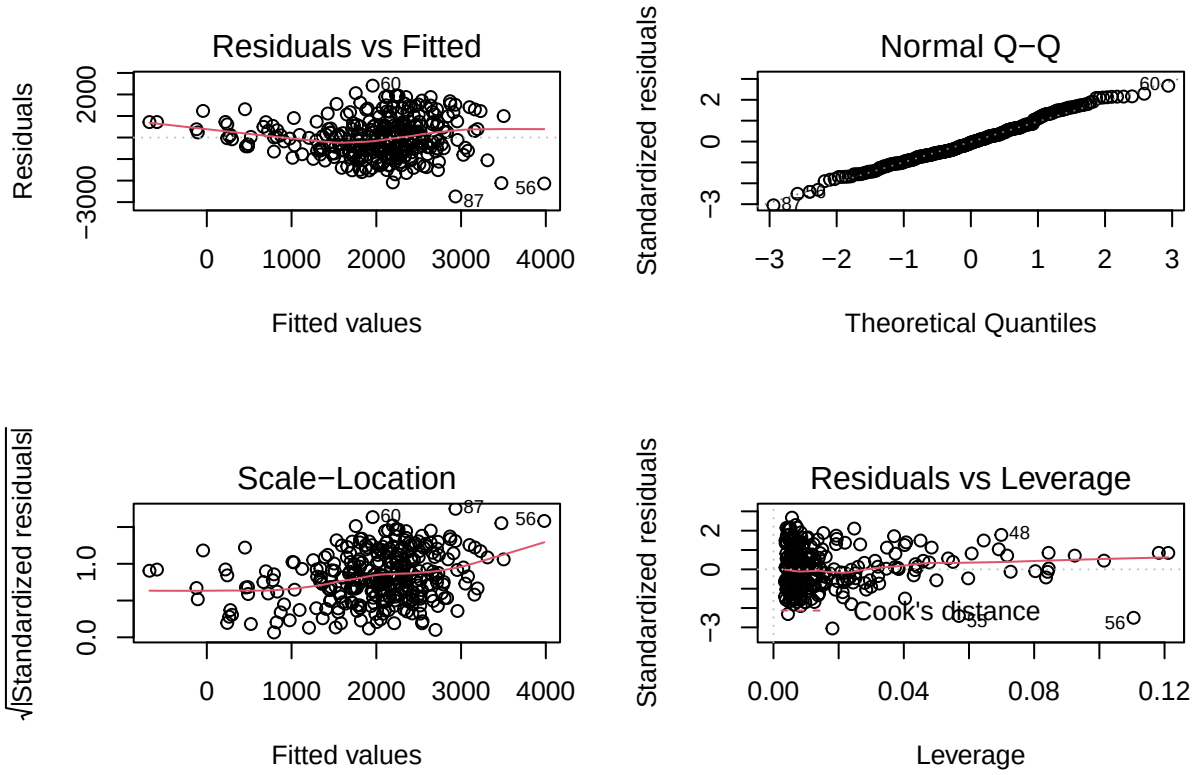
```
##
## Call:
## lm(formula = n_days ~ bilirubin + albumin + copper + alk_phos,
##     data = scaledchol)
##
## Coefficients:
## (Intercept)    bilirubin      albumin       copper     alk_phos
##      1999.1       -308.2        354.0       -227.5        285.0
```

We can see here as well that the best model is the model with size 4, the 4 variables being **bilirubin, albumin, copper and alk_phos.** Therefore, our model is best fitted as N_DAYS = 1999.1 - 308.2 * BILIRUBIN + 354 * ALBUMIN - 227.5 * COPPER + 285 * ALK_PHOS.

### 3C. Regression Diagnostics

Finally, in order to see if any regression assumptions are violated, we will plot the diagnostic plots for the final model lm(formula = n_days ~ bilirubin + albumin + copper + alk_phos, data = scaledchol).

```
par(mfrow = c(2,2))
plot(stepDay)
```




1. Residuals vs Fitted Plot –> Although the correlation between $\hat{y}$ and the residuals should be zero, we see that there is a non-linear relationship between the residuals and the fitted values. In addition, as the fitted values increase numerically, we can also notice that there is heteroscedasticity, indicating that the variance is different across the observations.

2. Scale-Location Plot –> Associated with the Residuals vs Fitted Plot is the Scale-Location plot, where it demonstrates an increasing pattern. Since the scale-location plot plots the square root of the absolute value of the residuals against the fitted values, if we see anything other than a constant pattern, we can conclude that it is heteroscedasticity.

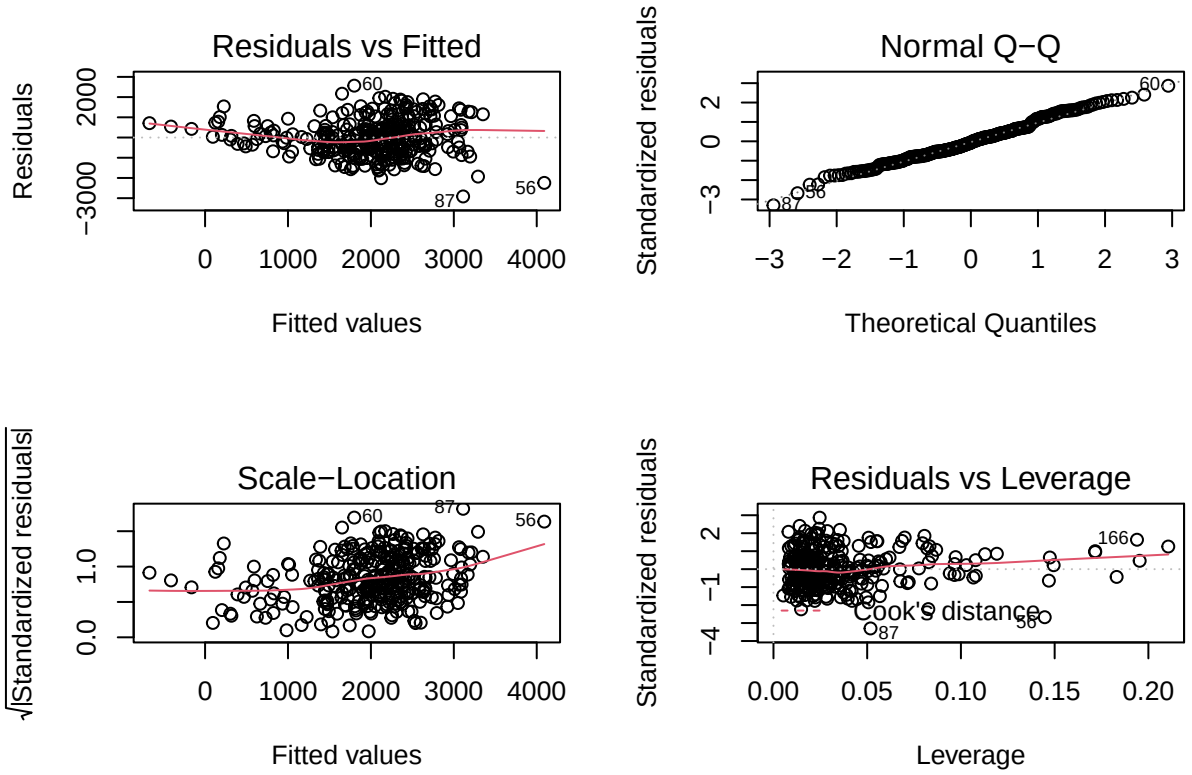
3. Normal Q-Q Plot –> In the plot of the standardized residuals, the plots seem to be located straight along the line, where every point in the final model is a represented as a quantile. Especially from [-2, 2] from the Theoretical Quantiles, we can see that the data set of the final model follows a strong normal distribution.

4. Residuals vs Leverage Plot –> The Residuals vs Leverage Plot can help us in detecting outliers and other observations we should be careful of. Since large leverage results can indicate possible outlier points, we can see in our cholangitis plot that i = 47, 54, 55 seem to be potential outlier points.

Finally, since we do not have a particular aspect to determine if there were any violations to independence, we have to believe that there were no violations to independence, assuming that the data was collected in a fair, with-consent manner.

Analysis on the overall data set can also be made to see if any assumptions were violated.

```
par(mfrow = c(2,2))
plot(fit)
```




1. Residuals vs Fitted Plot –> We can see some heteroscedasticity in the data, where variance is clearly unequal across the plot.

2. Scale-Location Plot –> Associated with the Residuals vs Fitted Plot is the Scale-Location plot, where it demonstrates an increasing pattern. Since the scale-location plot plots the square root of the absolute value of the residuals against the fitted values, if we see anything other than a constant pattern, we can conclude that it is heteroscedasticity.

3. Normal Q-Q Plot –> In the plot of the standardized residuals, the plots seem to be located along the line, where every point in the cholangitis data set is a quantile. Following a straight line across most if not all points demonstrates a normal distribution of the dataset.

4. Residuals vs Leverage Plot –> The Residuals vs Leverage Plot can help us in detecting outliers and other observations we should be careful of. Since large leverage results can indicate possible outlier points, we can see in our cholangitis plot that i = 55, 86, 163 seem to be potential outlier points.

### 4. Logistic Regression

For logistic regression, since we are fitting a model for the survival status of a patient at the end of the study, given the explanatory variables, we must filter the given data set differently than what we have done above. In addition, any NA values in the data must be removed to allow for analysis of the fitted model.

```
cholang_dat2 <- read.csv("cholangitis.csv")
cholang_dat2 <- na.omit(subset(cholang_dat2, status != "CL"))
```

Our main variable here is the "status" variable, which is now a binary variable with either "C" denoting that the patient is not dead or "D" denoting that the patient is dead.

We will first fit the logistic model, removing the categorical variables and converting the column "status" to a factor of numeric indicators, where 0 = "D" (Dead) and 1 = "C" (Not Dead).

```
cholang_dat2$status <- as.numeric(factor(cholang_dat2$status, c("C", "D"))) - 1
cholang_logistic <- cholang_dat2[, -c(1,4,6:10,20)]
glmChol <- glm(status ~., family = binomial, data = cholang_logistic)
summary(glmChol)
```

```
##
## Call:
## glm(formula = status ~ ., family = binomial, data = cholang_logistic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5876  -0.6212  -0.3555   0.5496   2.3966
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.316e+01  3.289e+00  -4.000 6.33e-05 ***
## n_days        -7.734e-04  1.916e-04  -4.037 5.41e-05 ***
## age            1.215e-04  4.650e-05   2.612 0.009001 **
## bilirubin      6.608e-02  7.749e-02   0.853 0.393841
## cholesterol    7.354e-04  9.737e-04   0.755 0.450079
## albumin        2.187e-02  4.509e-01   0.049 0.961313
## copper         4.284e-03  2.608e-03   1.642 0.100527
## alk_phos       2.945e-04  8.855e-05   3.325 0.000883 ***
## sgot           6.077e-03  3.191e-03   1.904 0.056876 .
## tryglicerides  4.692e-03  3.285e-03   1.428 0.153150
## platelets     -1.354e-03  1.932e-03  -0.701 0.483422
## prothrombin    9.025e-01  2.273e-01   3.971 7.17e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 393.11  on 287  degrees of freedom
## Residual deviance: 235.95  on 276  degrees of freedom
## AIC: 259.95
##
## Number of Fisher Scoring iterations: 6
```

We can interpret the function above as estimates for our parameters, where the change in log-odds of the event of a level of any variable above increase by one, assuming that all the other variables remain constant. This can be interpreted as being the same as the odds of a patient being alive being multiplied by exp(estimate) when the respective variable increases. For instance, we can see that the estimated coefficient of the variable "bilirubin" is 0.06608, which indicates the change in log-odds of the event of a patient not dying when the level of biliburin increases by one.

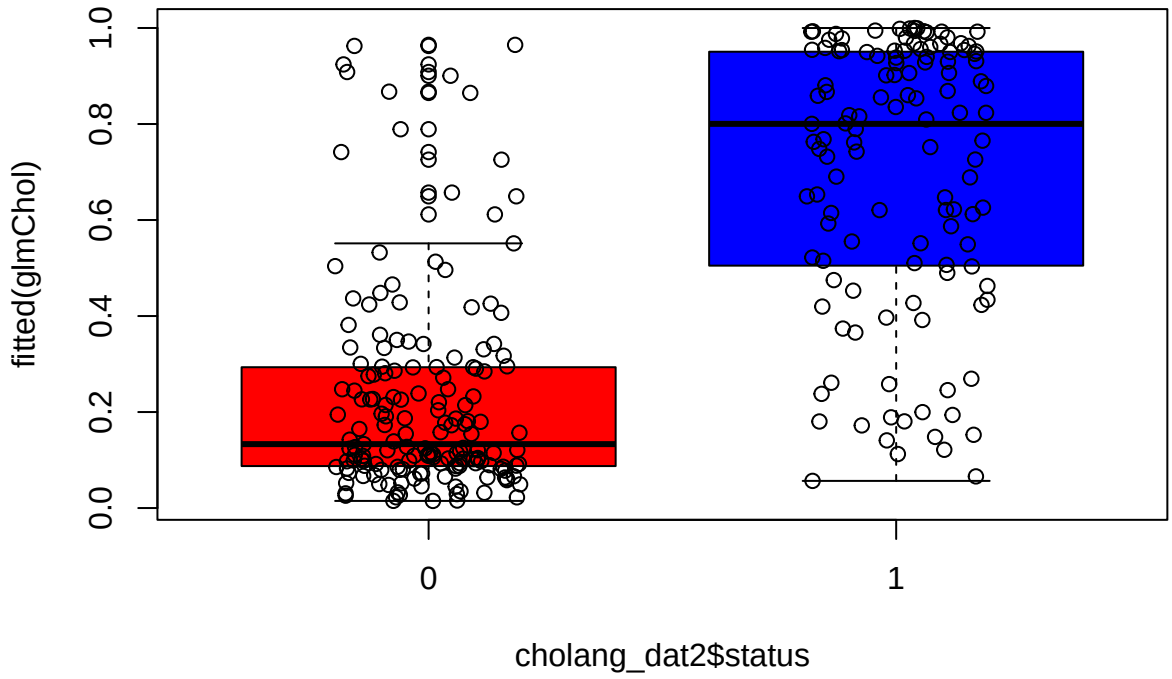
Next, the fitted probabilities in logistic regression for each observations in our sample is found.

```
chol_logisticFit <- fitted(glmChol)
head(chol_logisticFit)
```

```
##         1         2         3         4         6         7
## 0.9805303 0.2749036 0.8792199 0.5151161 0.2610750 0.1964521
```

Using the data above, we can plot the actual observed response values agains the fitted values

```
boxplot(fitted(glmChol) ~ cholang_dat2$status, at = c(0,1), col = c("red", "blue"))
points(x = jitter(cholang_dat2$status), fitted(glmChol))
```




The Residual Deviance (RD) is also calculated below to get a numerical measure of how "good" the fit is. As seen below, the value of RD is 235.9506, which is quite high and demonstrates a lack of fit for our model.

```
deviance(glmChol)
```

```
## [1] 235.9506
```

The deviance definitely comes with degrees of freedom (being n-p-1), we can attempt to use anova to compare the submodels, despite it not giving a significance value for comparing the submodel to the actual larger model at hand. The fitted function "f0" is a model with no variables.

```
f0 <- glm(status ~ 1, family = binomial, data = cholang_logistic)
anova(f0, glmChol)
```

```
## Analysis of Deviance Table
##
## Model 1: status ~ 1
## Model 2: status ~ n_days + age + bilirubin + cholesterol + albumin + copper +
##     alk_phos + sgot + tryglicerides + platelets + prothrombin
##   Resid. Df Resid. Dev Df Deviance
## 1       287     393.11
## 2       276     235.95 11   157.16
```

Instead of Residual Deviance, we can use AIC as a measure of good fit, like we did above in linear regression.

```
AIC(glmChol)
```

```
## [1] 259.9506
```

Again, we use the step function to compare the change in our RD values.

```
step(glmChol, direction = "both", trace = 0)
```

```
##
## Call:  glm(formula = status ~ n_days + age + copper + alk_phos + sgot +
##     tryglicerides + prothrombin, family = binomial, data = cholang_logistic)
##
## Coefficients:
##   (Intercept)         n_days            age         copper       alk_phos
##    -1.420e+01     -8.467e-04      1.164e-04      5.010e-03      2.919e-04
##          sgot  tryglicerides    prothrombin
##     8.191e-03      6.294e-03      9.908e-01
##
## Degrees of Freedom: 287 Total (i.e. Null);  280 Residual
## Null Deviance:       393.1
## Residual Deviance: 238.3     AIC: 254.3
```

We can see that the best model is that of size 7, removing the variables "bilirubin", "cholesterol", "albumin", and "platelets". While there are some limitations to using the step model— such as the nature of the function being to add/remove variables one at a time(leading to miss of possible best model) or the possibility of seeing a "best" model that is not found by this method— we utilize this procedure to see the model with the explanatory variables that have the most correlation to our response variable, which is status.

Therefore, we can finalize the model to be STATUS = -14.20 - 0.0008467 * N_DAYS + 0.0001164 * AGE + 0.00501 * COPPER + 0.0002919 * ALK_PHOS + 0.008191 * SGOT + 0.006294 * TRYGLICERIDES + 0.9908 * PROTHROMBIN.