Surgical Unit -Optimal Model Selection

```
library(tidyverse)
library(cowplot)
library(ggResidpanel)
```

Read the dataset

```
surgery <- read.csv(file.choose())
```

View the data

```
head(surgery)
```

```
##   blood prog enzyme liver age gender a.mod a.heavy survival
## 1   6.7   62     81  2.59  50      0     1       0      695
## 2   5.1   59     66  1.70  39      0     0       0      403
## 3   7.4   57     83  2.16  55      0     0       0      710
## 4   6.5   73     41  2.01  48      0     0       0      349
## 5   7.8   65    115  4.30  45      0     0       1     2343
## 6   5.8   38     72  1.42  65      1     1       0      348
```

Fit the linear model

```
fit.surgery <- lm(survival ~ ., data = surgery)
fit.surgery
```

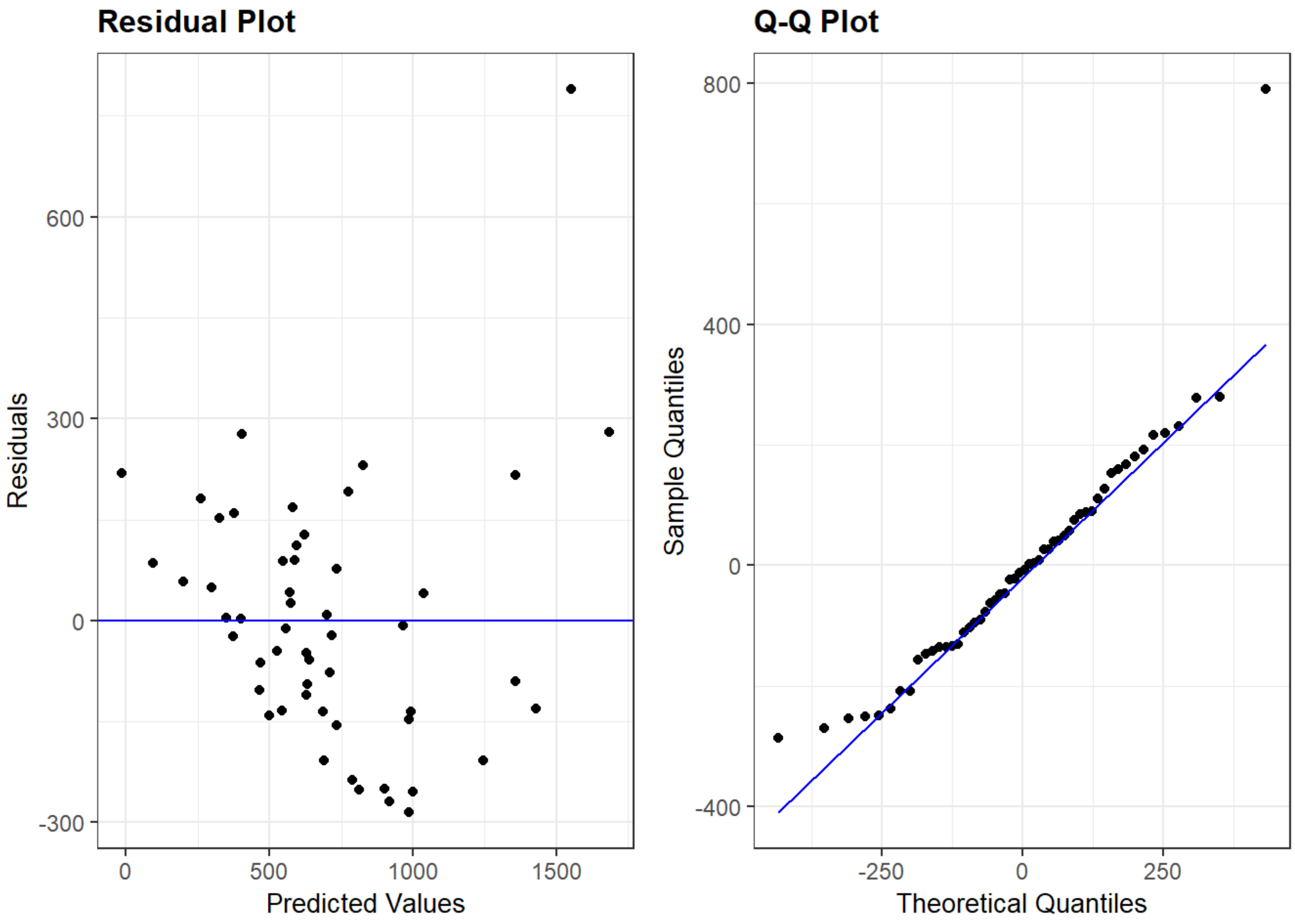
```
## 
## Call:
## lm(formula = survival ~ ., data = surgery)
## 
## Coefficients:
## (Intercept)        blood         prog       enzyme        liver          age  
##   -1148.823       62.390        8.973        9.888       50.413       -0.951  
##      gender        a.mod      a.heavy  
##      15.874        7.713      320.697  
```

MODEL DIAGNOSTICS.

Check for Linearity, Normality and Constant Variance

using the Residual plot (for linearity and constant variance) and Q-Q Plot (for normality)

```
plot1 <- resid_panel(fit.surgery, plots = c('resid', 'qq'))
plot1
```



Results

The Linearity Assumption was not viloated:

The residuals on the “Residual Plot” are not so far away from zero,

by standardization, since majority of our values are between 2 and -2

i.e. 200 and -200, then we are convinced there is a linear relationship

between the predictors and the response variable.

The Normality Assumption was not violated:

Majority of the residuals/errors of the model seems to lie well on the

45 degree line on the “Q-Q Plot”, although with very few outliers.

Thus, we would conclude that the assumption holds true.

The Constant Variance assumption was violated:

From the “Residual Plot”, the residuals do no have a constant varaince

across the predicted values (x -axis).

Transformation of Unequal Variances

We transform the response variable.

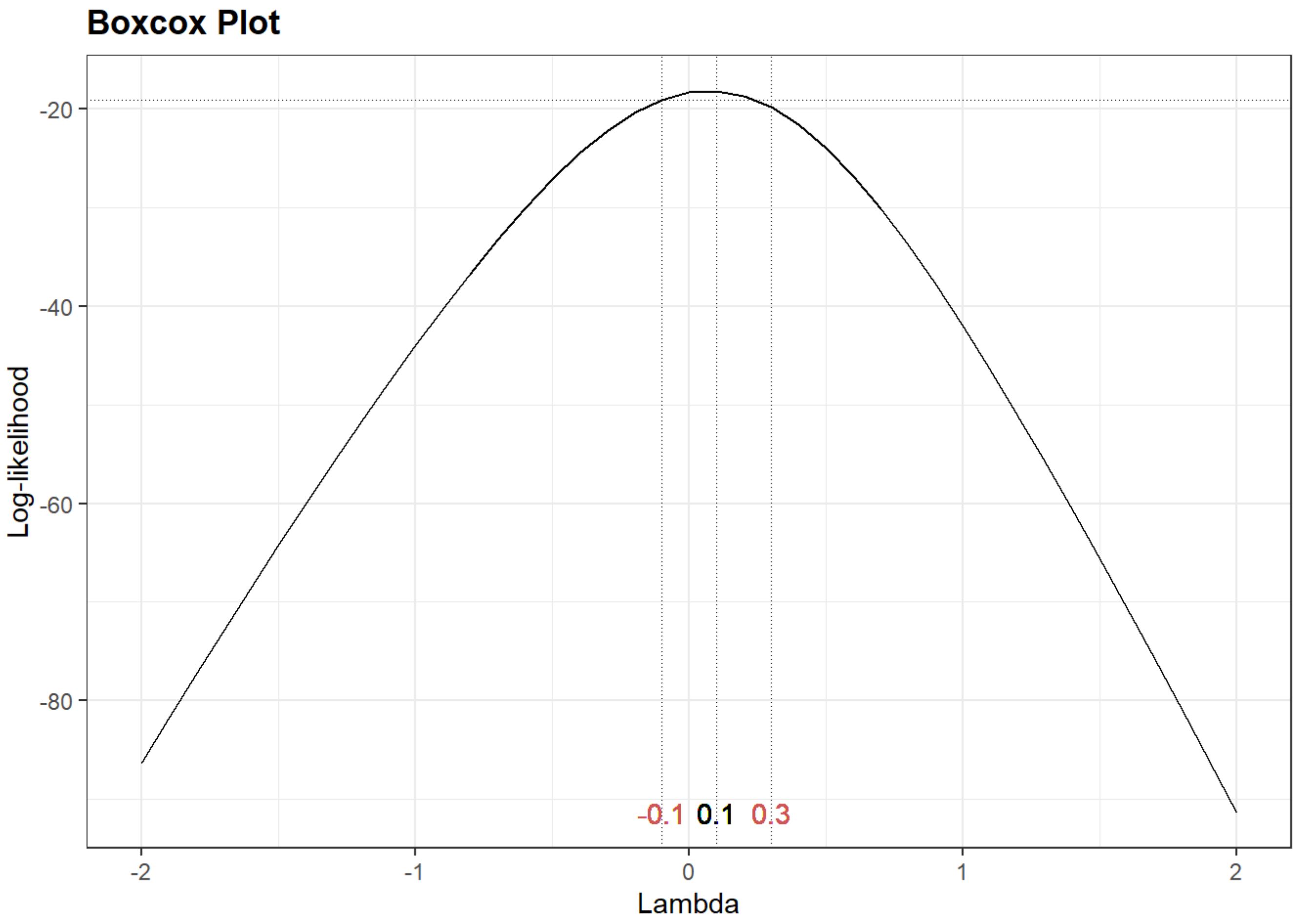
This is because, we transform Y (response variable) when assumptions of

“constant variance” and “normality” are violated. We transform X (predictor)

when “linearity” is violated.

We do this transformation of Y using the Box-Cox Transformation

```
library(lindia)

gg_boxcox(fit.surgery) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold"))
```



Results

The lambda value is 0.1 with a 95% confidence interval of -0.1 and 0.3

Being in practice, it makes more sense to take lambda = 0.0

Since lambda = 0, then we transform Y as " ln(Y) "

Transform Y in the dataset

```
surgery <- mutate(surgery, ln.survival = log(survival))

head(surgery)
```

```
##   blood prog enzyme liver age gender a.mod a.heavy survival ln.survival
## 1   6.7   62     81  2.59  50      0     1       0      695    6.543912
## 2   5.1   59     66  1.70  39      0     0       0      403    5.998937
## 3   7.4   57     83  2.16  55      0     0       0      710    6.565265
## 4   6.5   73     41  2.01  48      0     0       0      349    5.855072
## 5   7.8   65    115  4.30  45      0     0       1     2343    7.759187
## 6   5.8   38     72  1.42  65      1     1       0      348    5.852202
```

Remove the original response variable

```
surgery <- surgery %>% select(-survival)

head(surgery)
```

```
##   blood prog enzyme liver age gender a.mod a.heavy ln.survival
## 1   6.7   62     81  2.59  50      0     1       0    6.543912
## 2   5.1   59     66  1.70  39      0     0       0    5.998937
## 3   7.4   57     83  2.16  55      0     0       0    6.565265
## 4   6.5   73     41  2.01  48      0     0       0    5.855072
## 5   7.8   65    115  4.30  45      0     0       1    7.759187
## 6   5.8   38     72  1.42  65      1     1       0    5.852202
```

Re-fit the model using the transformed Y

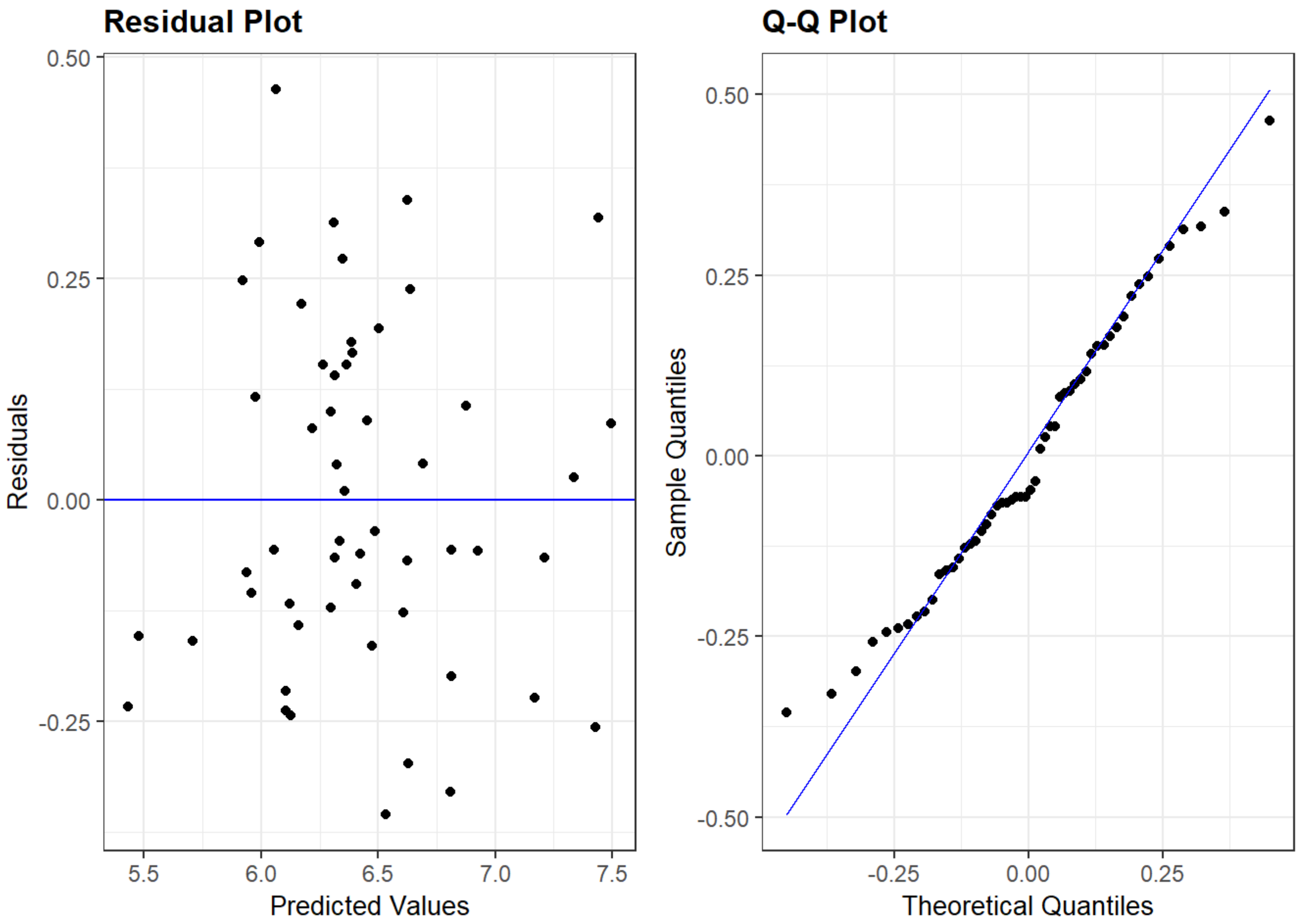
```
# use the new response variable to fit the model

fit2.surgery <- lm(ln.survival ~., surgery)
fit2.surgery
```

```
## 
## Call:
## lm(formula = ln.survival ~ ., data = surgery)
## 
## Coefficients:
## (Intercept)        blood         prog       enzyme        liver          age  
##    4.050949     0.068551     0.013459     0.014948     0.007931    -0.003567  
##      gender        a.mod      a.heavy  
##    0.084151     0.057313     0.388190
```

MODEL DIAGNOSTICS 2

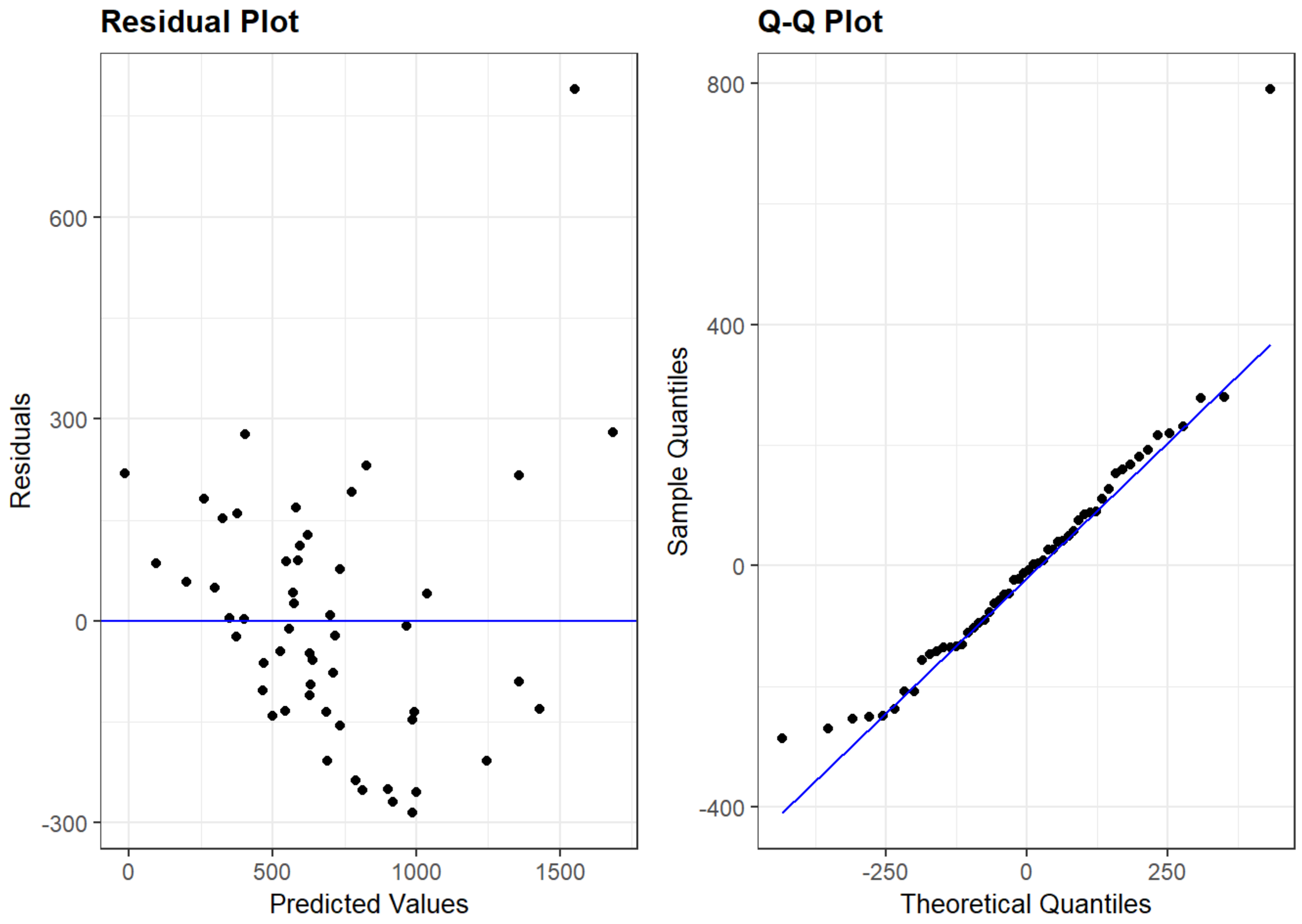
Recheck for Linearity, Normality and Constant Variance assumptions

```
plot2 <- resid_panel(fit2.surgery, plots = c('resid', 'qq'))
plot2
```

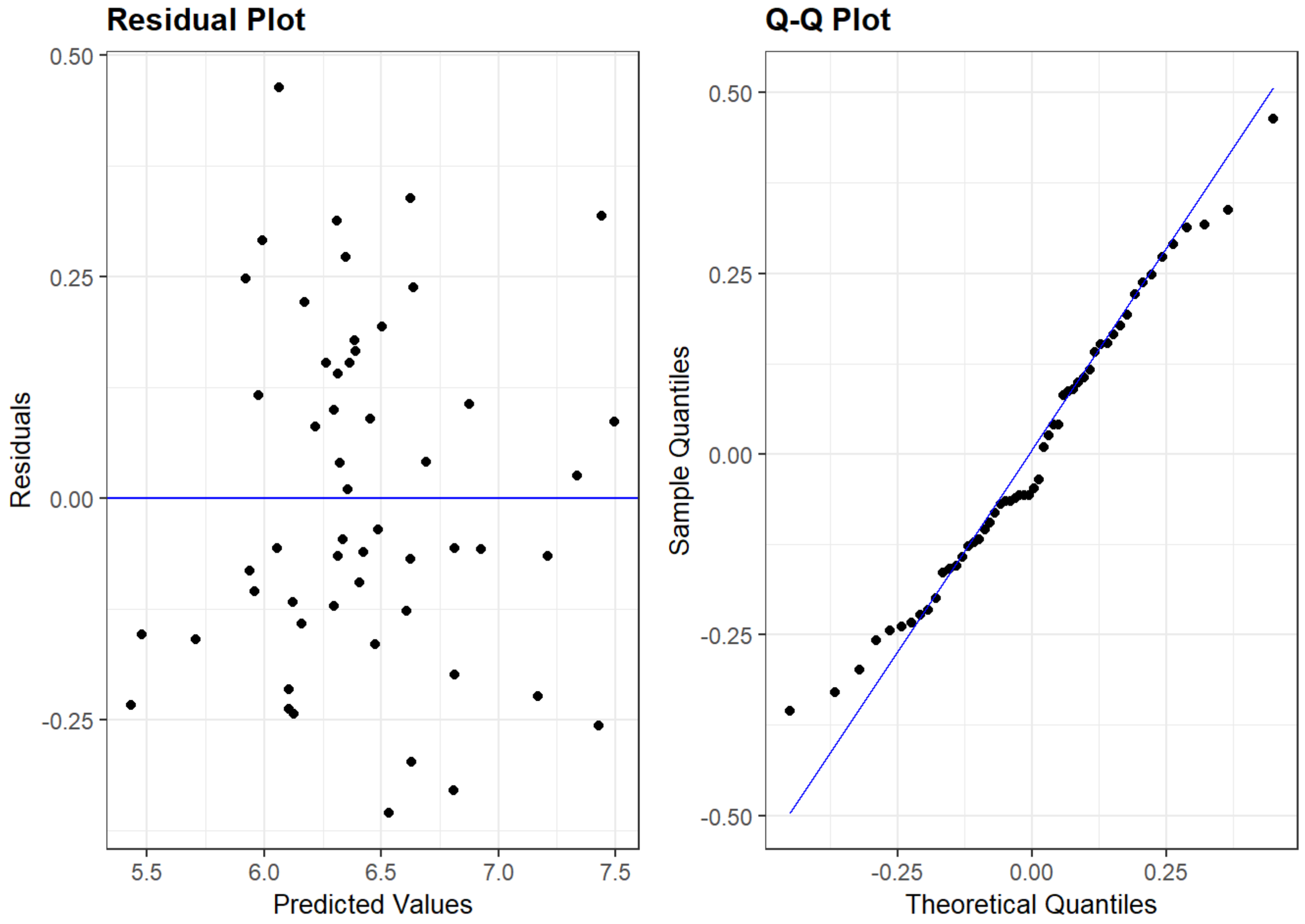


Compare old (before transformation) and new (after transformation)

```
old.par <- par(mfrow=c(2, 1))
plot1
```


Residual Plot
Residuals
600
300
0
-300
0
500
1000
1500
Predicted Values
Q-Q Plot
Sample Quantiles
800
400
0
-400
-250
0
250
Theoretical Quantiles

```
plot2
```


Residual Plot
Residuals
0.50
0.25
0.00
-0.25
5.5
6.0
6.5
7.0
7.5
Predicted Values
Q-Q Plot
Sample Quantiles
0.50
0.25
0.00
-0.25
-0.50
-0.25
0.00
0.25
Theoretical Quantiles

```
par(old.par)
```

Results

1) With this transformation, the variance of the predictor is a little better

i.e. more constant than before the transformation

2) The main outliers on the Q-Q plot has been taken care of.

Our final model is “fit2.surgery”

BEST MODEL SELECTION

Using the “Best Subset Algorithms” method

We have “p-1” predictor variables. So the possible number of models will be

2^(p-1) = 2^(8) = 256.

* where “p” is the number of predictor variables and p-1 is 8 (intercept excluded)

As “p” increases, so does the multiple-coefficient-of-determination (R^2_p)

thus, we should not pick the model with the largest R^2_p.

We need the model with the least R^2_p.

We seek a levelling off point where adding more variables provides little

increase in R^2_p.

```
library(leaps)

select_model <- regsubsets(ln.survival ~ .,
                           data = surgery)

search.models = summary(select_model)
search.models
```

```
## Subset selection object
## Call: regsubsets.formula(ln.survival ~ ., data = surgery)
## 8 Variables  (and intercept)
##         Forced in Forced out
## blood       FALSE      FALSE
## prog        FALSE      FALSE
## enzyme      FALSE      FALSE
## liver       FALSE      FALSE
## age         FALSE      FALSE
## gender      FALSE      FALSE
## a.mod       FALSE      FALSE
## a.heavy     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          blood prog enzyme liver age gender a.mod a.heavy
## 1  ( 1 ) " "   " "  "*"    " "   " " " "    " "   " "    
## 2  ( 1 ) " "   "*"  "*"    " "   " " " "    " "   " "    
## 3  ( 1 ) " "   "*"  "*"    " "   " " " "    " "   "*"    
## 4  ( 1 ) "*"   "*"  "*"    " "   " " " "    " "   "*"    
## 5  ( 1 ) "*"   "*"  "*"    " "   " " "*"    " "   "*"    
## 6  ( 1 ) "*"   "*"  "*"    " "   "*" "*"    " "   "*"    
## 7  ( 1 ) "*"   "*"  "*"    " "   "*" "*"    "*"   "*"    
## 8  ( 1 ) "*"   "*"  "*"    "*"   "*" "*"    "*"   "*"    
```

Results

We already established that there are 256 possible models and

from these 256 models:

* the best model with 1 predictor variable has “enzyme”

* the best model with 2 predictor variables has “prog” and “enzyme”

* the best model with 3 predictor variables has “blood”, “prog” and “enzyme”

* …

* …

* …

* the best model with 7 predictor variables has all predictors except “liver”

* the best model with 8 predictor variables has all predictors

Which is the Best Model to use?

i.e. we have 8 best model, which of them is the best, since we must not use

all the predictor variables.

* We use the adjusted R-squared approach (R^2_a,p), since it adjust for us more

paramters to the regression model.

* We want the model with the largest value of adjusted R-square

Note.

What is “R-square”?: This is called the coefficient of determination. It is

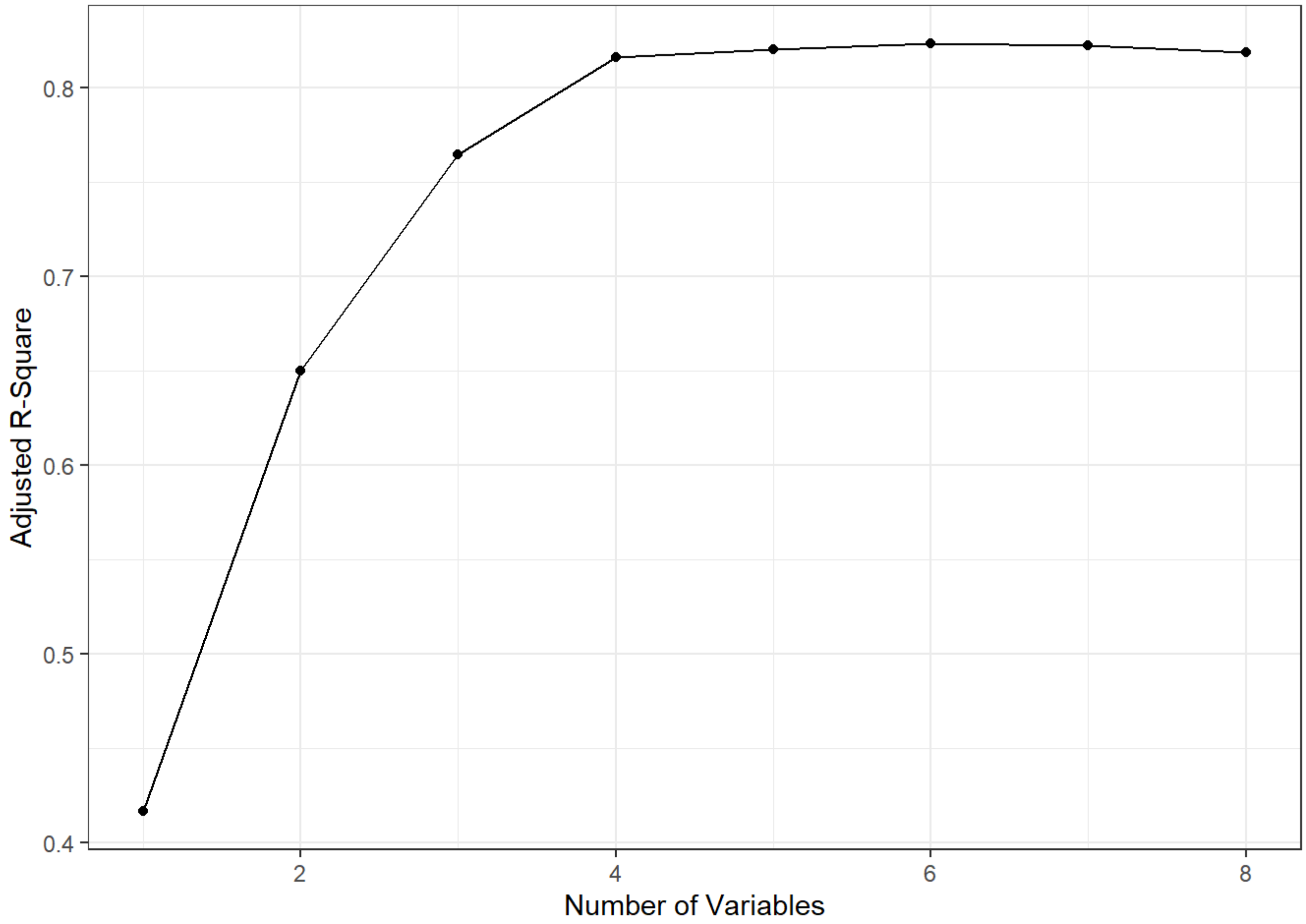
the percentage of observed variability in the Y-variable that is

explained by the model (predictors).

What is “Adjusted R-square”?: Adjusted R-squared adjusts the statistic based

on the number of independent variables in the model.

```
tibble(predictors = seq(1,8),
       y = search.models$adjr2) %>%
                                 ggplot(aes(x = predictors, y = y)) +
                                 geom_point() +
                                 geom_line() +
                                 labs(x = "Number of Variables",
                                      y = "Adjusted R-Square") +
                                 theme_bw()
```



Check the number of variables that has the highest Adjusted R-Square in the graph

```
which.max(search.models$adjr2)
```

```
## [1] 6
```

RESULTS

1) Both with the graph and extra check, we see that the number of variables

as our best model is 6 (without intercept).

2) Hence, by “Best Subset Algorithm” approach, our best model will only need:

blood, prog, enzyme, age, gender, and a.heavy(heavy alcohol users) as predictors

The best model will become:

```
best_model  = lm(formula = ln.survival ~ blood + prog + enzyme + age + gender + a.heavy,
                 data    = surgery)

best_model
```

```
##
## Call:
## lm(formula = ln.survival ~ blood + prog + enzyme + age + gender +
##     a.heavy, data = surgery)
##
## Coefficients:
## (Intercept)        blood         prog       enzyme          age       gender
##    4.054389     0.071524     0.013760     0.015109    -0.003452     0.087227
##     a.heavy
##    0.351069
```

Although our predictor variables here are less than 30, we would like to

also use a method that will suit cases when the variables are 30 or more.

Using “Stepwise Regression Method”

* When the number of variables is 30 or more then “best” subset algorithms

* become very compuationally expensive.

We will employ the Backward Stepwise Search (or simply backward elimination):

* This elimination starts with the full model (all predictors) and then

* removes one variable at a time in the model.

* It compares models using numerical criteria such as “AIC” (Akakie Information Criteria)

Source: Applied Linear Statistical Methods by Michael H. Kutner et al. 5th edition

Backward Elimination:

```
backward_elim <- step(object    = fit2.surgery,
                      direction = "backward")
```

```
## Start:  AIC=-160.78
## ln.survival ~ blood + prog + enzyme + liver + age + gender + 
##     a.mod + a.heavy
## 
##           Df Sum of Sq    RSS     AIC
## - liver    1   0.00126 1.9718 -162.74
## - a.mod    1   0.03159 2.0021 -161.92
## - age      1   0.07359 2.0441 -160.80
## <none>                 1.9705 -160.78
## - gender   1   0.08403 2.0545 -160.52
## - blood    1   0.31845 2.2890 -154.69
## - a.heavy  1   0.84489 2.8154 -143.51
## - prog     1   2.09285 4.0634 -123.70
## - enzyme   1   2.98863 4.9591 -112.94
## 
## Step:  AIC=-162.74
## ln.survival ~ blood + prog + enzyme + age + gender + a.mod + 
##     a.heavy
## 
##           Df Sum of Sq    RSS      AIC
## - a.mod    1    0.0326 2.0043 -163.858
## <none>                 1.9718 -162.743
## - age      1    0.0876 2.0593 -162.396
## - gender   1    0.0969 2.0687 -162.152
## - blood    1    0.6269 2.5987 -149.835
## - a.heavy  1    0.8438 2.8156 -145.506
## - prog     1    2.6755 4.6473 -118.446
## - enzyme   1    5.0934 7.0652  -95.825
## 
## Step:  AIC=-163.86
## ln.survival ~ blood + prog + enzyme + age + gender + a.heavy
## 
##           Df Sum of Sq    RSS      AIC
## <none>                 2.0043 -163.858
## - age      1    0.0769 2.0812 -163.826
## - gender   1    0.0975 2.1018 -163.293
## - blood    1    0.6284 2.6327 -151.133
## - a.heavy  1    0.9011 2.9054 -145.810
## - prog     1    2.7644 4.7688 -119.052
## - enzyme   1    5.0752 7.0795  -97.716
```

RESULT

1) The reduction stopped at 6 variables.

2) This implies that our BEST model is with the following predictors:

blood, prog, enzyme, age, gender and a.heavy

3) These predictors are the same the predictors we got when we used the

Adjusted R-squared criteria. Hence, the same final and best model.

DIAGNOSIS OF BEST MODEL

Check for Multicollinearity of of our final and best model.

Using the Variance Inflation Factor (VIF) = 1/(1 - R_squared)

* VIF is between 1 and infinity.

* VIF > 10 indicated “severe multicollinearity” but as a rule of thumb.

* VIF > 5 is often regarded as “severe multicollinearity”.

```
library(car)

vif(best_model)
```

```
##    blood     prog   enzyme      age   gender  a.heavy
## 1.108821 1.037099 1.076909 1.016471 1.047745 1.114506
```

RESULT

We see that there exist no significant MULTICOLLINEARITY amongst

our predictor variables in our model.