final

```
setwd("C:/Users/ShinJiyoon/OneDrive/바탕 화면/uga fall 2024/STAT4230/final")
```

Problem 1.

When participants are unable to eat for long periods of time, they must receive parenteral nutrition (fed intravenously). Such patients often show increased calcium loss in their urine, sometimes losing more calcium than they receive in their intravenous fluids. Calcium loss may contribute to bone loss as the body draws calcium from the bones to maintain calcium level in the blood within normal ranges. Potential confounding variables include glomerular filtration rate, dietary sodium, and dietary protein. These data are stored in file P1.txt, and are in 5 columns: column 1) Urinary Calcium y, column 2) Dietary Calcium x_1, column 3) Filtration Rate x_2, column 4) Urinary Sodium x_3, column 5) Dietary Protein x_4. Consider independent variables x_1, x_2, x_3 and x_4 in an all-possible-regressions selection procedure.

a. How many models for $E(y)$ are possible in total?

```
2**4 - 1
```

```
## [1] 15
```

b. For each case in part (a), use a statistical software package to find the maximum R^2, minimum C_p, and minimum $PRESS$.

```
data_p1 <- read.table("P1.txt", header=TRUE, sep=" ")
head(data_p1)
```

	Y <int>	X1 <int>	X2 <int>	X3 <int>	X4 <int>
1	220	554	63	54	80
2	182	303	43	99	48
3	166	287	45	14	53
4	162	519	58	55	70
5	137	249	53	37	82
6	136	142	70	37	36

6 rows

```
variables <- c("X1", "X2", "X3", "X4")
combinations <- unlist(lapply(1:length(variables),
                              function(x) combn(variables, x, simplify=FALSE)), recursive=FALSE)
combinations
```

```
## [[1]]
## [1] "X1"
##
## [[2]]
## [1] "X2"
##
## [[3]]
## [1] "X3"
##
## [[4]]
## [1] "X4"
##
## [[5]]
## [1] "X1" "X2"
##
## [[6]]
## [1] "X1" "X3"
##
## [[7]]
## [1] "X1" "X4"
##
## [[8]]
## [1] "X2" "X3"
##
## [[9]]
## [1] "X2" "X4"
##
## [[10]]
## [1] "X3" "X4"
##
## [[11]]
## [1] "X1" "X2" "X3"
##
## [[12]]
## [1] "X1" "X2" "X4"
##
## [[13]]
## [1] "X1" "X3" "X4"
##
## [[14]]
## [1] "X2" "X3" "X4"
##
## [[15]]
## [1] "X1" "X2" "X3" "X4"
```

```
library(DAAG)

results <- data.frame()

for (combo in combinations) {
  formula <- as.formula(paste("Y ~ ", paste(combo, collapse="+")))

  model <- lm(formula, data=data_p1)

  r_squared <- summary(model)$r.squared
  MSE <- summary(model)$sigma^2
  n <- length(data_p1$Y)
  p <- length(coef(model)) - 1
  Cp <- MSE*model$df.residual / (summary(lm(Y~X1+X2+X3+X4,data=data_p1))$sigma^2) - n + 2*(p+1)
  press <- press(model)

  results <- rbind(results,
                   data.frame(
                     variables = paste(combo, collapse=", "),
                     rSquared = r_squared,
                     Cp = Cp,
                     press = press
                   ))
}

results
```

variables <chr>	rSquared <dbl>	Cp <dbl>	press <dbl>
X1	0.5751128	12.306937	47131.43
X2	0.1683081	46.111274	92735.31
X3	0.2434601	39.866345	87649.98
X4	0.4023837	26.660242	67576.82
X1, X2	0.6601511	7.240492	39682.93
X1, X3	0.7150501	2.678540	34862.36
X1, X4	0.5805672	13.853696	50911.57
X2, X3	0.2617366	40.347624	94956.96
X2, X4	0.4819479	22.048679	61643.83
X3, X4	0.5498467	16.406484	57761.55


```
row_index_r2_max = which.max(results$rSquared)
row_index_cp_min = which.min(results$Cp)
row_index_press_min = which.min(results$press)

print(paste("maximum R squared: (", results$variables[row_index_r2_max], ") ", results$rSquared[row_index_r2_ma
x]))
```

```
## [1] "maximum R squared: ( X1, X2, X3, X4 )  0.735249810495487"
```

```
print(paste("minimum Cp:        (", results$variables[row_index_cp_min], ") ", results$Cp[row_index_cp_min]))
```

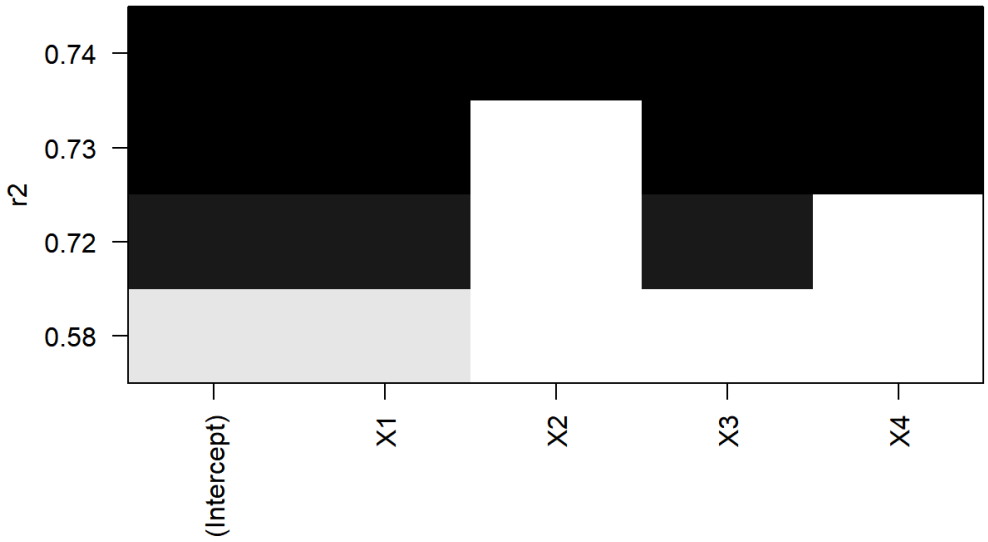
```
## [1] "minimum Cp:        ( X1, X3 )  2.67854029702545"
```

```
print(paste("minimum PRESS:     (", results$variables[row_index_press_min], ") ", results$press[row_index_press_mi
n]))
```

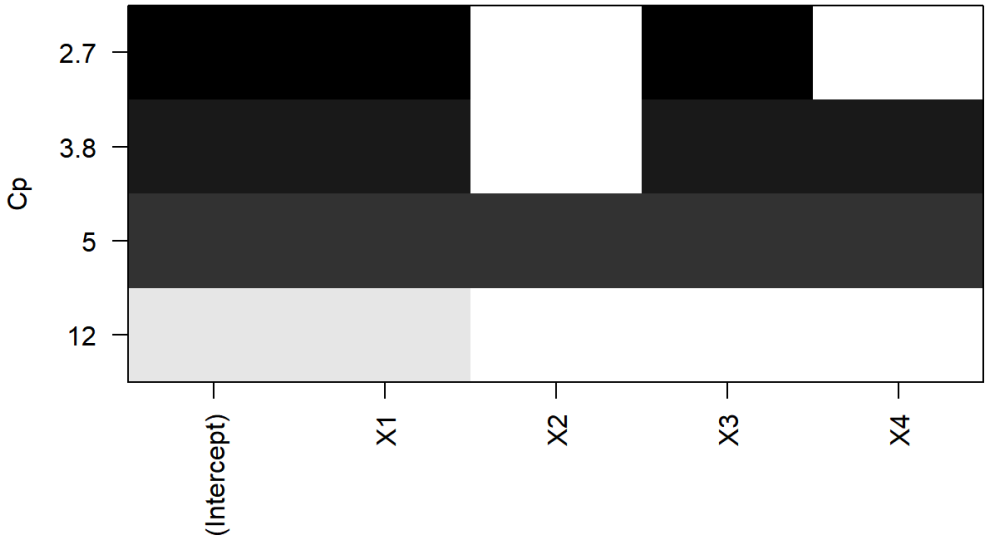
```
## [1] "minimum PRESS:     ( X1, X3 )  34862.3563215795"
```

c. Plot each of the quantities R^2, C_p, and $PRESS$ in part (b) against p, the number of independent variables in the subset model.

```
library(leaps)
subsets <- regsubsets(Y ~ X1+X2+X3+X4, data=data_p1)
plot(subsets, scale="r2")
```




```
plot(subsets, scale="Cp")
```



d. Based on the plots in part (c), which variables would you select under criterions R^2, C_p, and $PRESS$, respectively?

- The model with all 4 variables X_1, X_2, X_3, X_4 achieves the highest R^2.
- The model with all 4 variables X_1, X_2, X_3, X_4 does not have the lowest C_p, but it is the closest to $p+1$, indicating that it may be the most appropriate model.
- The model with 2 variables X_1, X_3 achieves the lowest $PRESS$.

e. Based on the best variables you choose under criterion R^2, fit a multiple linear model and plot the summary table.

```
model_p1 <- lm(Y ~ X1+X2+X3+X4, data=data_p1)
summary(model_p1)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4, data = data_p1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.553 -25.722  -4.973  13.928  73.640
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.7316    21.8887  -0.262 0.795872
## X1            0.3450     0.0889   3.881 0.000806 ***
## X2            0.4265     0.4872   0.876 0.390740
## X3            0.4980     0.2213   2.250 0.034753 *
## X4           -0.5113     0.4831  -1.058 0.301391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.19 on 22 degrees of freedom
## Multiple R-squared:  0.7352, Adjusted R-squared:  0.6871
## F-statistic: 15.27 on 4 and 22 DF,  p-value: 4.071e-06
```

Problem 2.

It has been suggested that central nervous system malformations in newborns may be related to the hardness of water supplies. Data on the number of central nervous system malformations per 1000 births, and water hardness (in ppm) were collected from 20 geographic regions. These data are stored in P2.txt with two columns: column 1) malformation rate y (per 1000 birhts), and column 2) hardness x (ppm).

a. Fit the first-order model to the data.

```
data_p2 <- read.table("P2.txt", header=TRUE, sep=" ")
data_p2
```

	Y <dbl>	X <int>
1	7.2	50
2	6.3	160
3	8.1	25
4	12.5	50
5	11.2	15
6	15.0	45
7	9.3	75
8	6.5	60
9	9.4	100
10	8.0	100

1-10 of 20 rows Previous **1** 2 Next

```
model_p2 <- lm(Y~X, data=data_p2)
summary(model_p2)
```

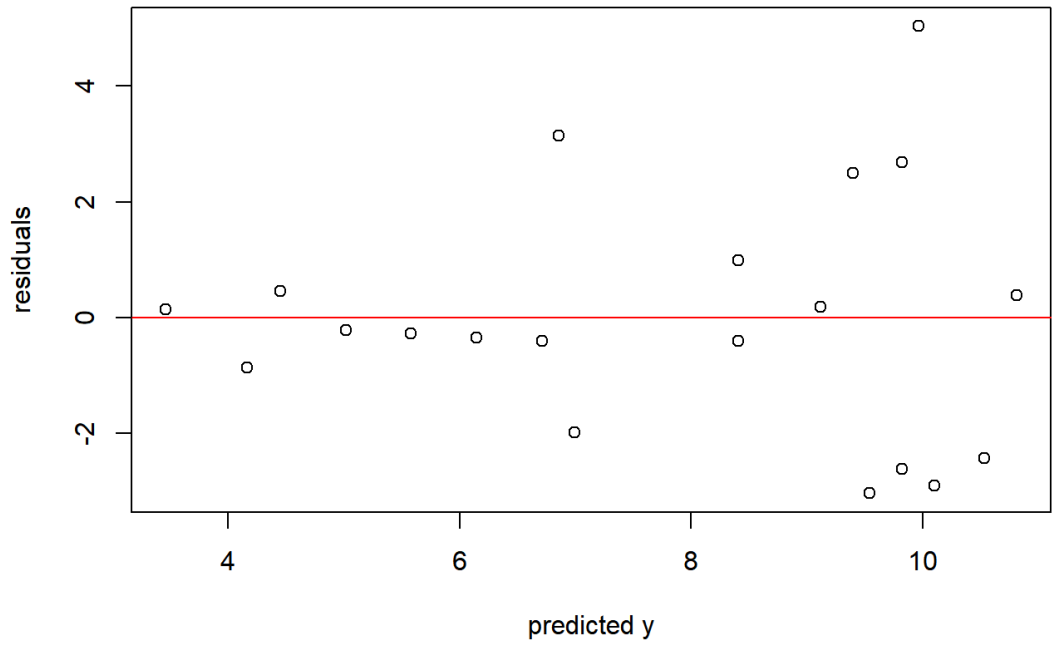
```
##
## Call:
## lm(formula = Y ~ X, data = data_p2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0405 -1.1469 -0.2463  0.5872  5.0351
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.238111   0.889018  12.641 2.17e-10 ***
## X           -0.028294   0.006052  -4.675 0.000188 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.184 on 18 degrees of freedom
## Multiple R-squared:  0.5484, Adjusted R-squared:  0.5233
## F-statistic: 21.86 on 1 and 18 DF,  p-value: 0.0001884
```

b. Calculate the residuals and construct a residual plot versus $\hat{y}$.

```
residuals(model_p2)
```

```
##          1          2          3          4          5          6          7          8          9         10
11
## -2.6234020 -0.4110416 -2.4307567  2.6765980  0.3863015  5.0351270  0.1839526 -3.0404602  0.9913073 -0.4086927 -
1.9939834
##         12         13         14         15         16         17         18         19         20
##  3.1474875 -0.3451579 -0.2792741 -0.8645648  0.4524933  0.1427898 -2.9063439 -0.2133904  2.5010108
```

```
plot(fitted(model_p2), residuals(model_p2), xlab="predicted y", ylab="residuals")
abline(h=0, col="red")
```



c. What does the plot from part (b) suggest about the variance of y? What are three potential solutions you could stabilize the variances?

- The residual plot shows that the variance increases as the $\hat{y}$ increases. This suggests a funnel-shaped pattern, indicating non-constant variance of residuals.
- Three potential solutions to stabilize the variances are:
  - Try transformation on y, starting with a log transformation.
  - If the log transformation does not stabilize the variance, move on to a box-cox-transformation.
  - If these transformations are not working, apply weighted least squares regression or robust regression.

d. Refit the model using all the three variance-stabilizing transformation methods in part (c). Plot the residuals for all the transformed models and compare to the plot obtained in part (b). Get a brief summary.

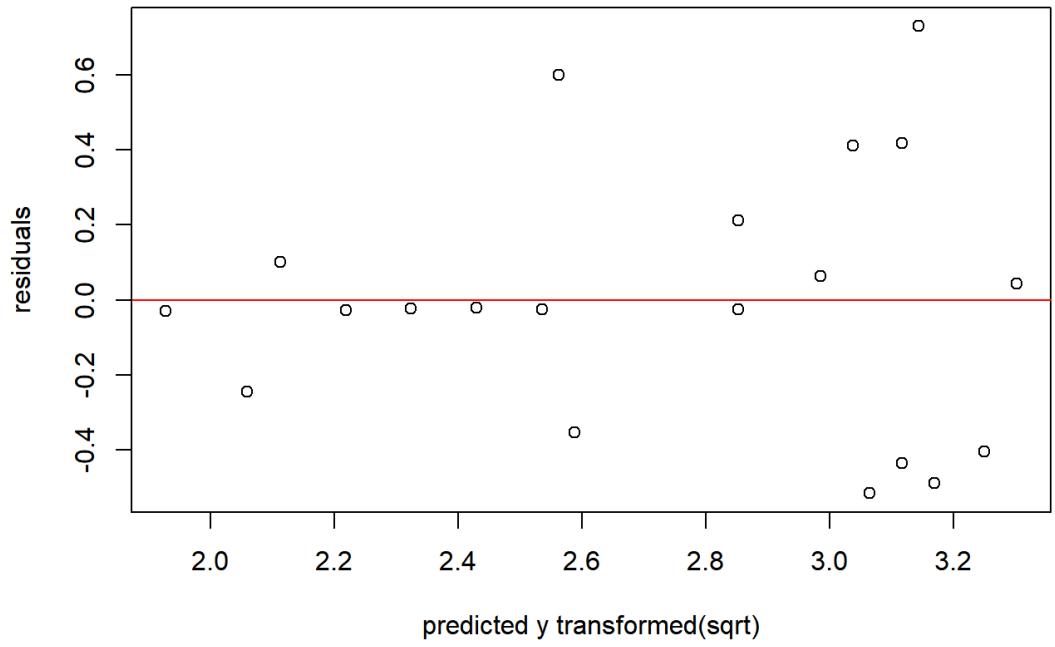
- squared root transformation

```
data_p2$Y_sqrt <- sqrt(data_p2$Y)
data_p2
```

	Y <dbl>	X <int>	Y_sqrt <dbl>
1	7.2	50	2.683282
2	6.3	160	2.509980
3	8.1	25	2.846050
4	12.5	50	3.535534
5	11.2	15	3.346640
6	15.0	45	3.872983
7	9.3	75	3.049590
8	6.5	60	2.549510
9	9.4	100	3.065942
10	8.0	100	2.828427

1-10 of 20 rows Previous **1** 2 Next

```
model_p2_sqrt <- lm(Y_sqrt ~ X, data=data_p2)
plot(fitted(model_p2_sqrt), residuals(model_p2_sqrt), xlab="predicted y transformed(sqrt)", ylab="residuals")
abline(h=0, col="red")
```



Still shows a funnel-shaped pattern. This indicates the square root transformation is not effective.

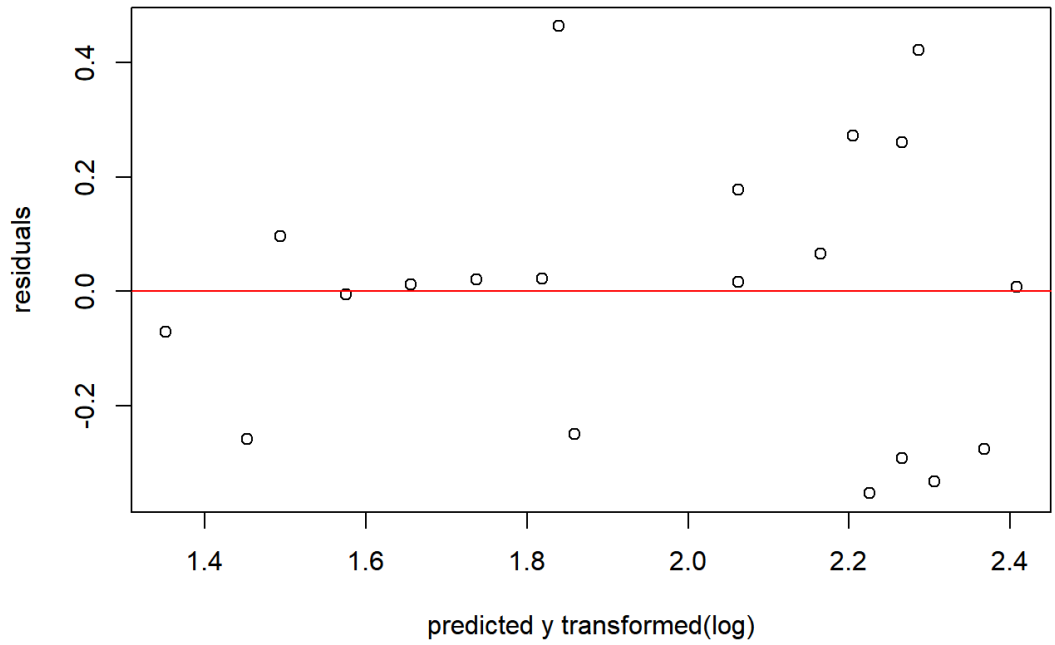
- log transformation

```
data_p2$Y_log <- log(data_p2$Y)
data_p2
```

	Y <dbl>	X <int>	Y_sqrt <dbl>	Y_log <dbl>
1	7.2	50	2.683282	1.974081
2	6.3	160	2.509980	1.840550
3	8.1	25	2.846050	2.091864
4	12.5	50	3.535534	2.525729
5	11.2	15	3.346640	2.415914
6	15.0	45	3.872983	2.708050
7	9.3	75	3.049590	2.230014
8	6.5	60	2.549510	1.871802
9	9.4	100	3.065942	2.240710
10	8.0	100	2.828427	2.079442

1-10 of 20 rows Previous **1** 2 Next

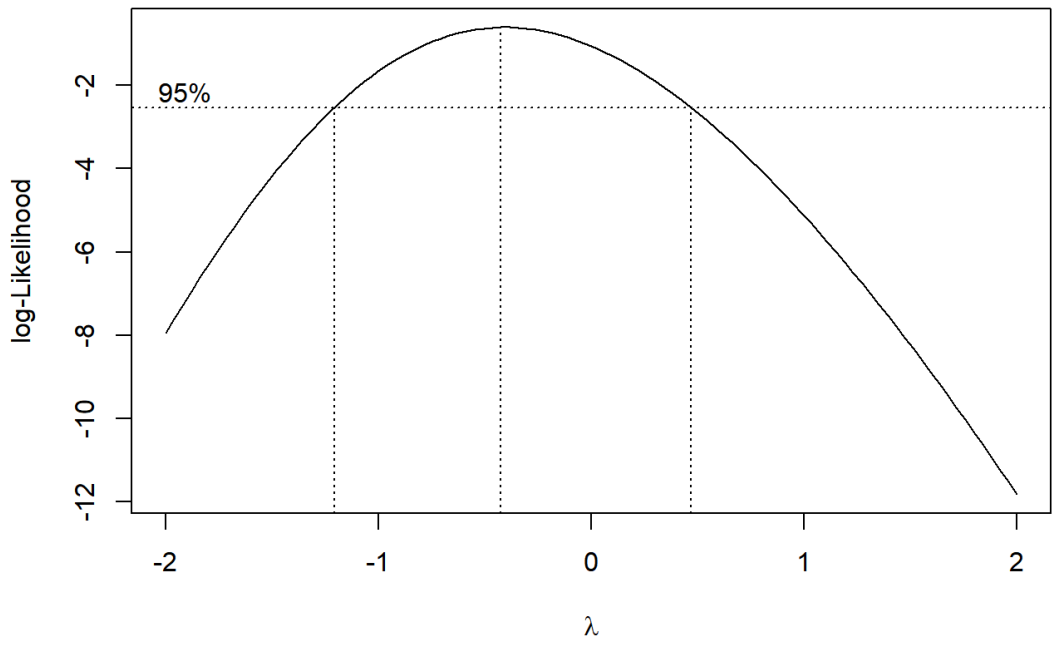
```
model_p2_log <- lm(Y_log ~ X, data=data_p2)
plot(fitted(model_p2_log), residuals(model_p2_log), xlab="predicted y transformed(log)", ylab="residuals")
abline(h=0, col="red")
```



It demonstrates a more stabilized variance compared to the origin data but still shows a slightly funnel-shaped pattern.

- box-cox transformation

```
# find out optimal lambda
library(MASS)
boxcox_result <- boxcox(model_p2, lambda=seq(-2, 2, by=0.1))
```

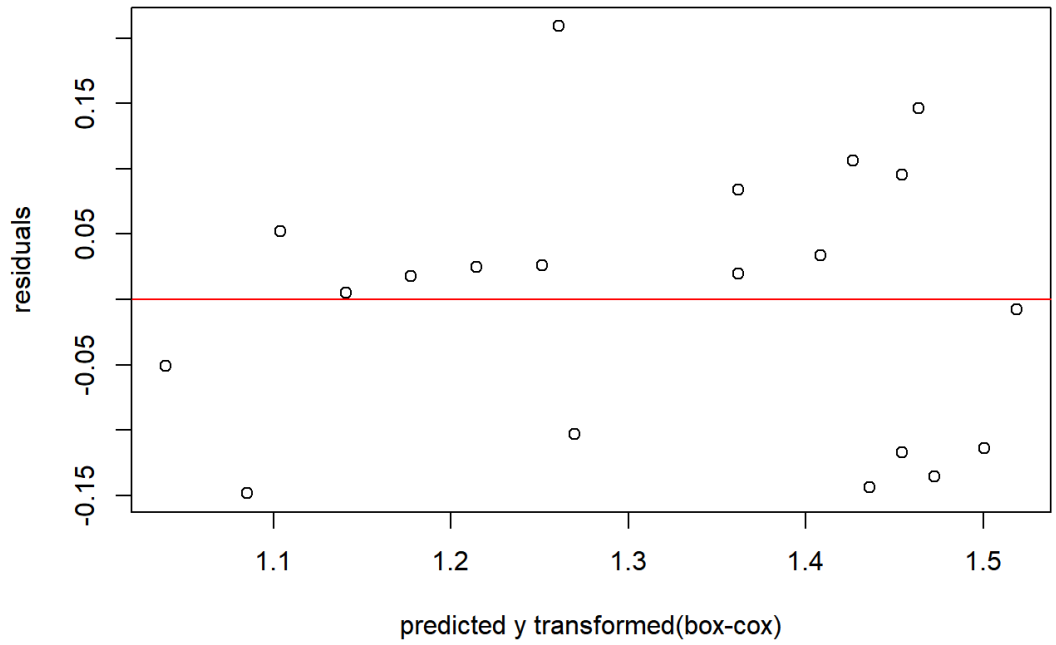



```
(optimal_lambda <- boxcox_result$x[which.max(boxcox_result$y)])
```

```
## [1] -0.4242424
```

```
if (optimal_lambda == 0) {
  data_p2$Y_boxcox <- log(data_p2$Y)
} else {
  data_p2$Y_boxcox <- (data_p2$Y ** optimal_lambda - 1) / optimal_lambda
}
```

```
model_p2_boxcox <- lm(Y_boxcox ~ X, data=data_p2)
plot(fitted(model_p2_boxcox), residuals(model_p2_boxcox), xlab="predicted y transformed(box-cox)", ylab="residual
s")
abline(h=0, col="red")
```



This demonstrates a more stabilized variance compared to the log transformation, suggesting an improved model fit.