Lab4_Transformations

Nahom Ayele & Nick Huo

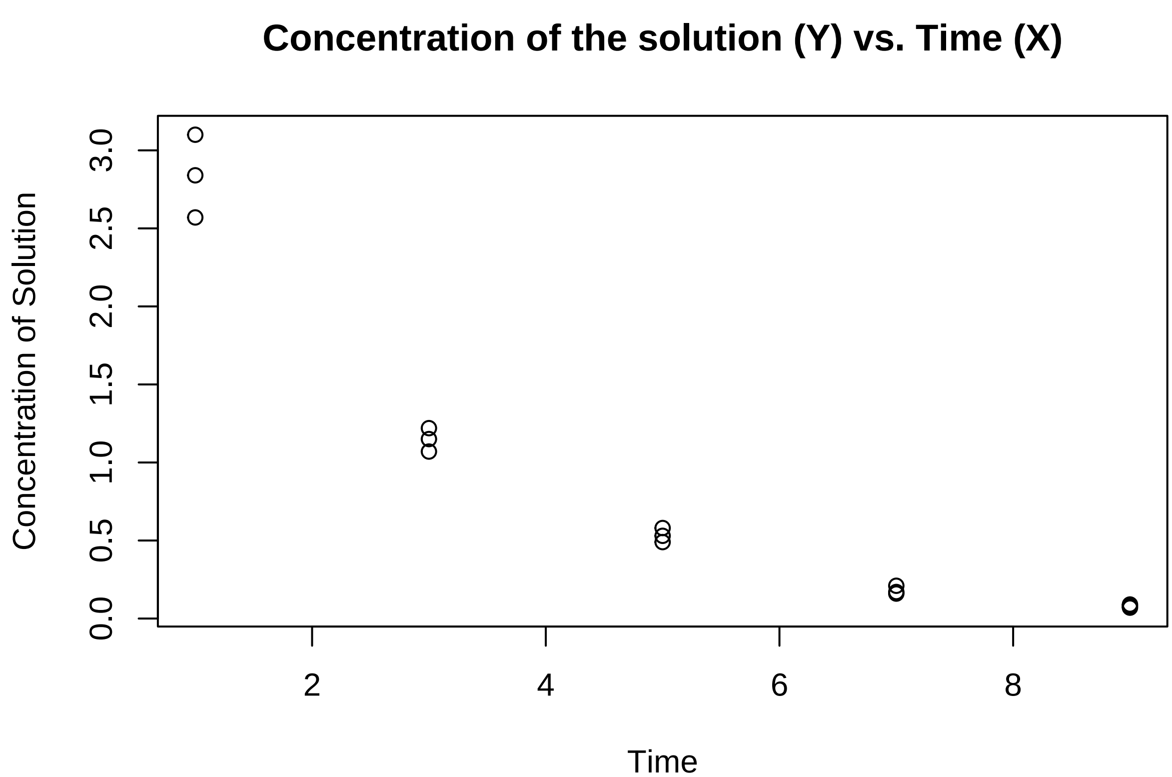
2022-09-30

```r
# Import Data set
sc <- read.csv("./SolutionConcentration.csv")
x <- sc$x
y <- sc$y
```

Question 1

Part (1)

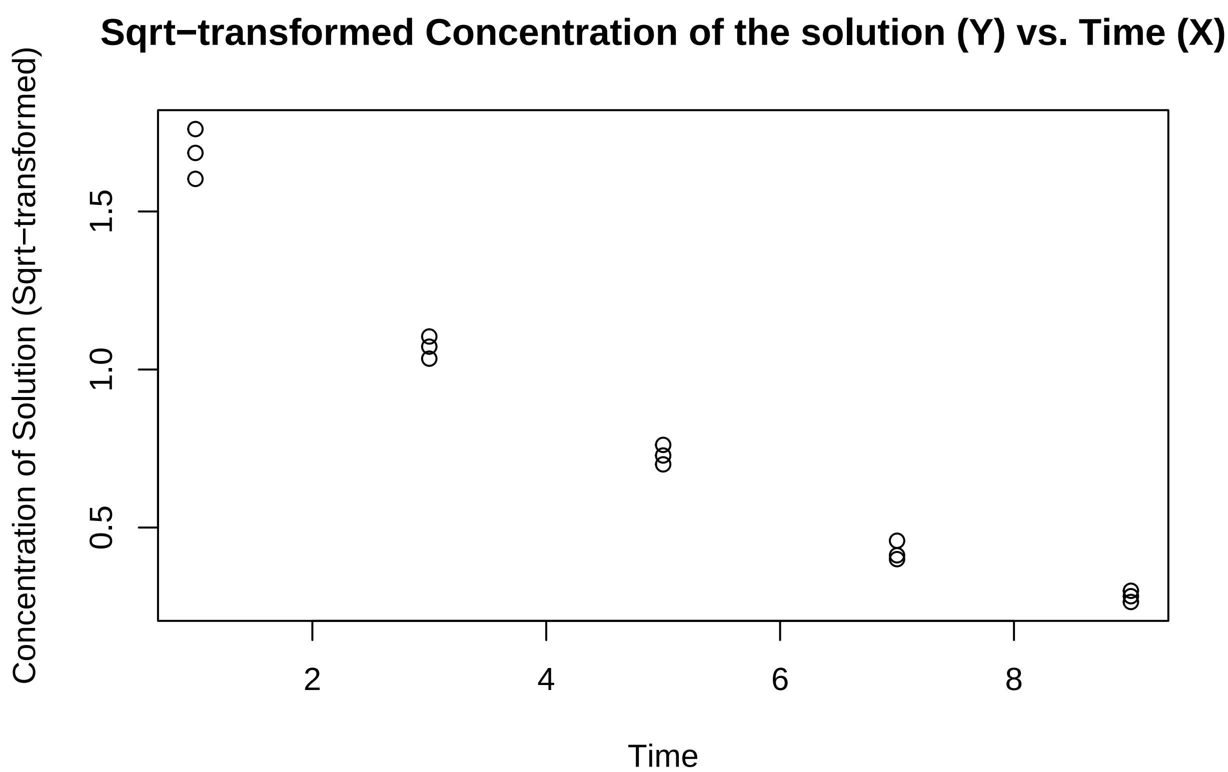
```r
plot(x, y, main='Concentration of the solution (Y) vs. Time (X)',
     xlab='Time', ylab='Concentration of Solution')
```



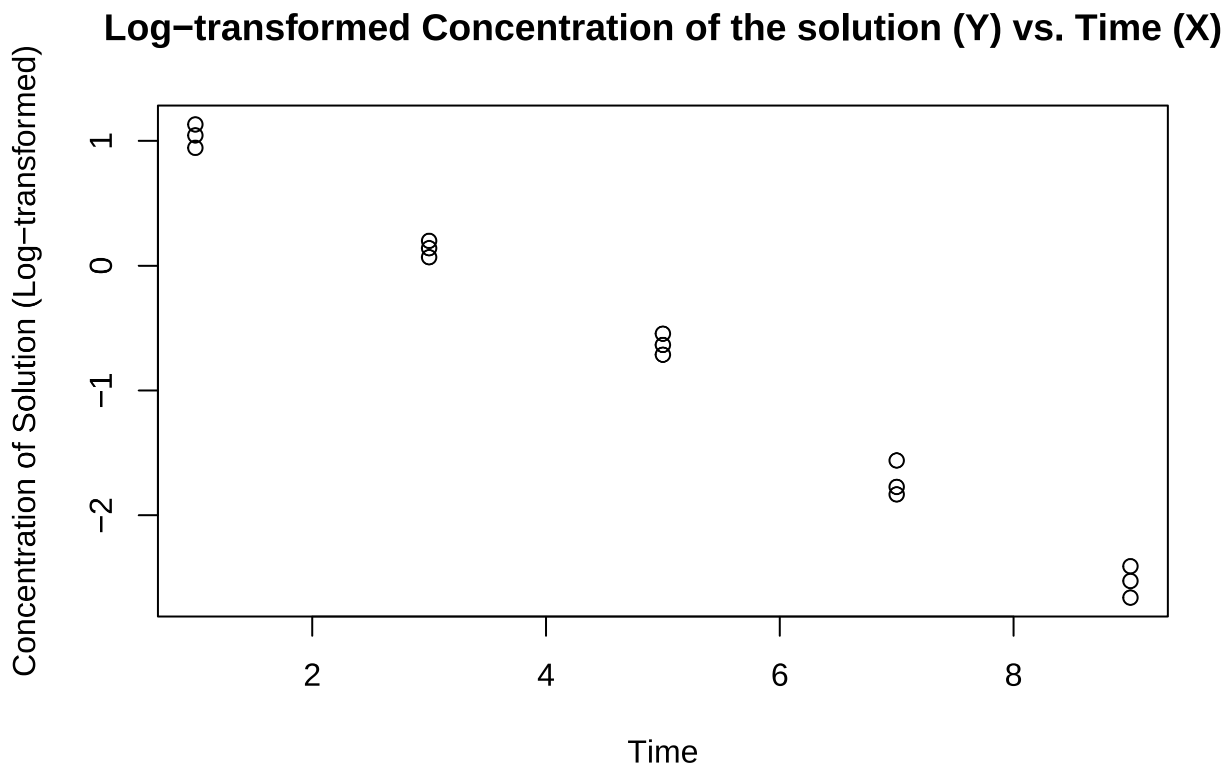
We observe that the relationship appears to be a exponential decay function. As time progresses, the concentration of the solution is exponentially decaying.

We think that the most appropriate transformation would be a squared root or a log transform, because we need to bring the high Y values at small X down and not change the low Y values as much.

```r
## plotting Log transform
plot(x, sqrt(y), main='Sqrt-transformed Concentration of the solution (Y) vs. Time (X)',
     xlab='Time', ylab='Concentration of Solution (Sqrt-transformed)')
```



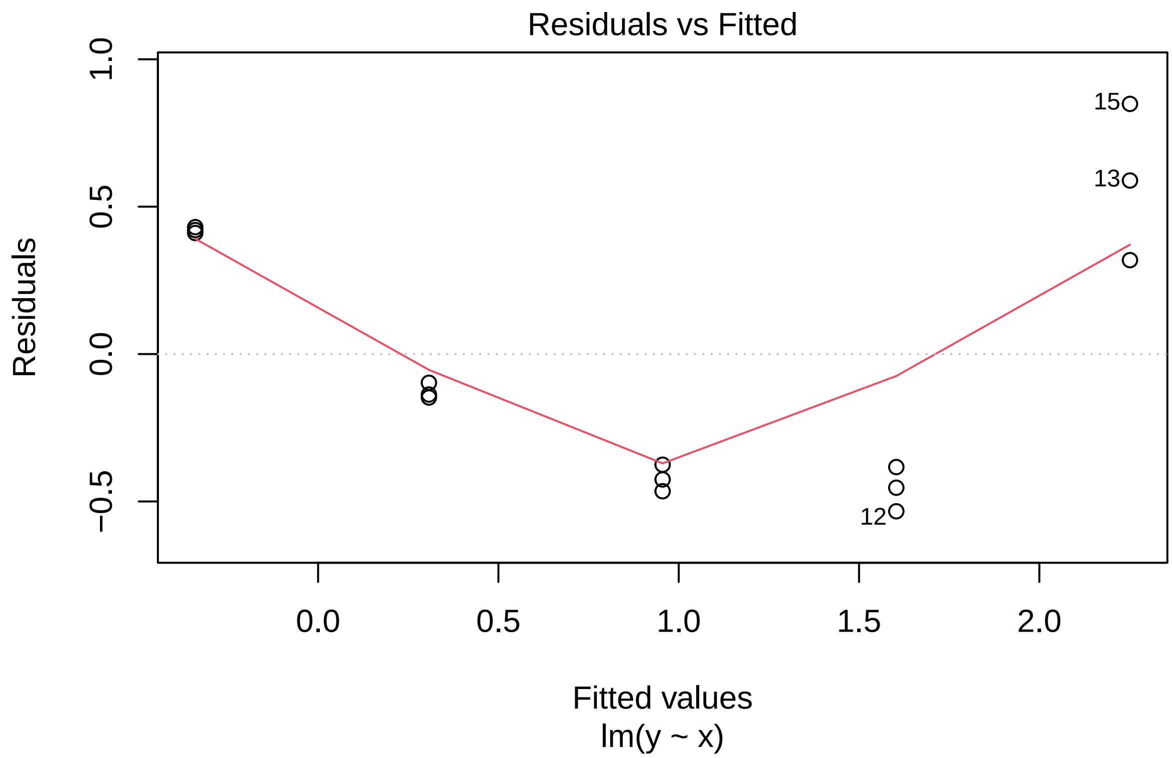

```r
plot(x, log(y), main='Log-transformed Concentration of the solution (Y) vs. Time (X)',
     xlab='Time', ylab='Concentration of Solution (Log-transformed)')
```

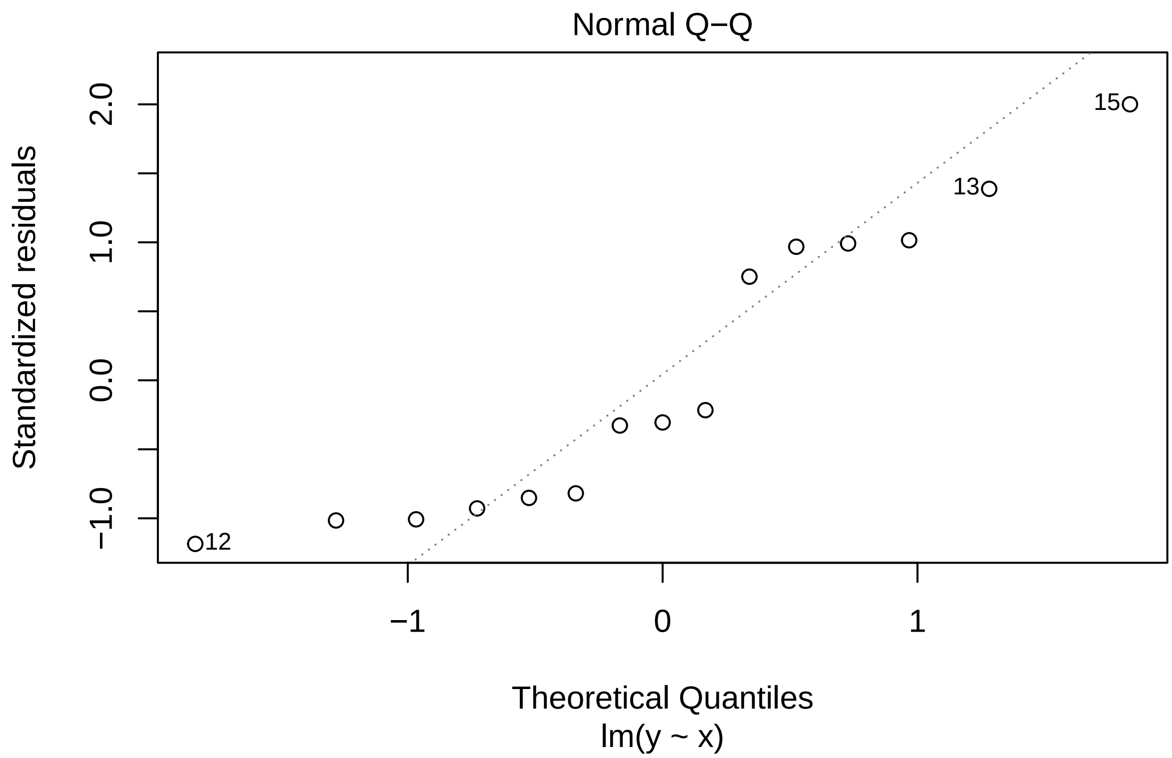


It appears that the log transformation is better as it makes the relationship between time (X) and log-concentration ($\ln[Y]$) quite linear.

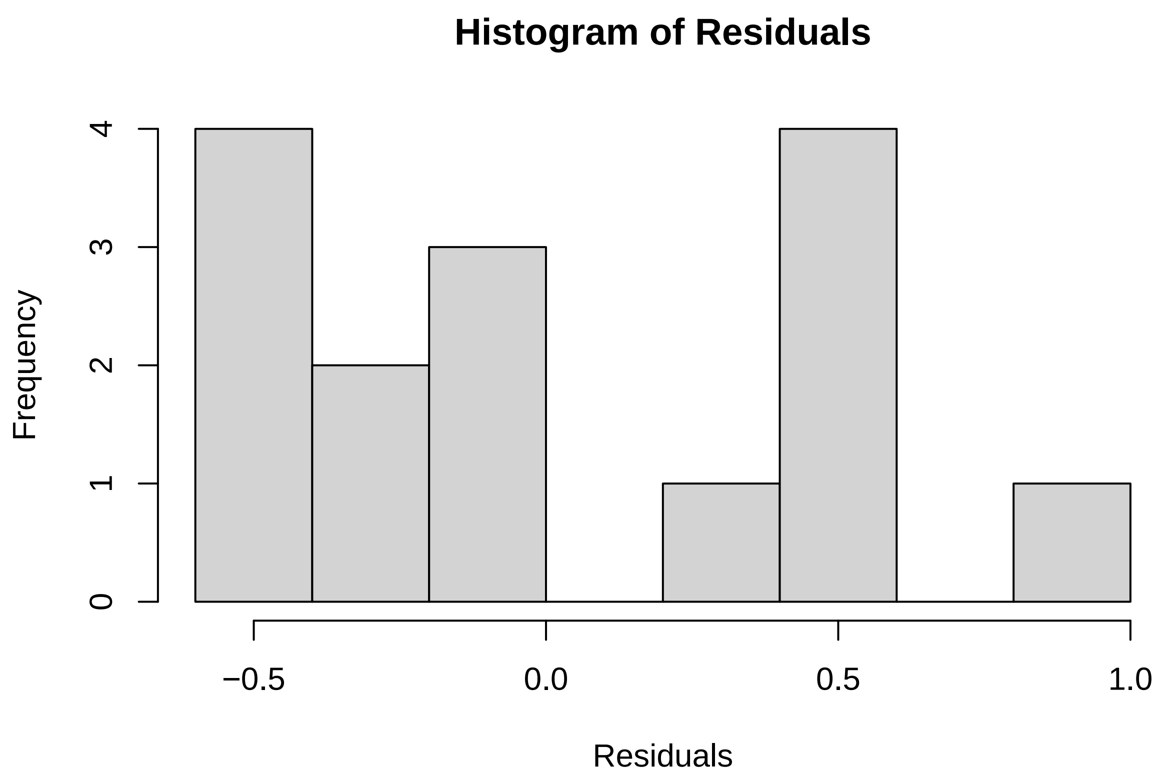
Part (2)

```
og_model <- lm(y ~ x)
plot(og_model, which=c(1,2))
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
−0.5
15
13
12
0.0
0.5
1.0
1.5
2.0
Fitted values
lm(y ~ x)




```
hist(resid(og_model), main='Histogram of Residuals', xlab='Residuals')
```



CONCLUSIONS FROM CONDITIONS??

Part (3)

```
std_obs_wi <- function(y, lbd) {
    n <- length(y)
    k2 <- prod(y)^(1/n) # geometric mean of the Yi observations

    if (lbd==0) {
        return (k2*log(y))
    } else {
        k1 <- 1/(lbd*k2^(lbd-1))
        return (k1*((y^lbd)-1))
    }
}
```

Part (4)

```
sse <- function(x, w) {
  stdmodel <- lm(w~x)
  return (sum((fitted(stdmodel)-w)^2))
}
```

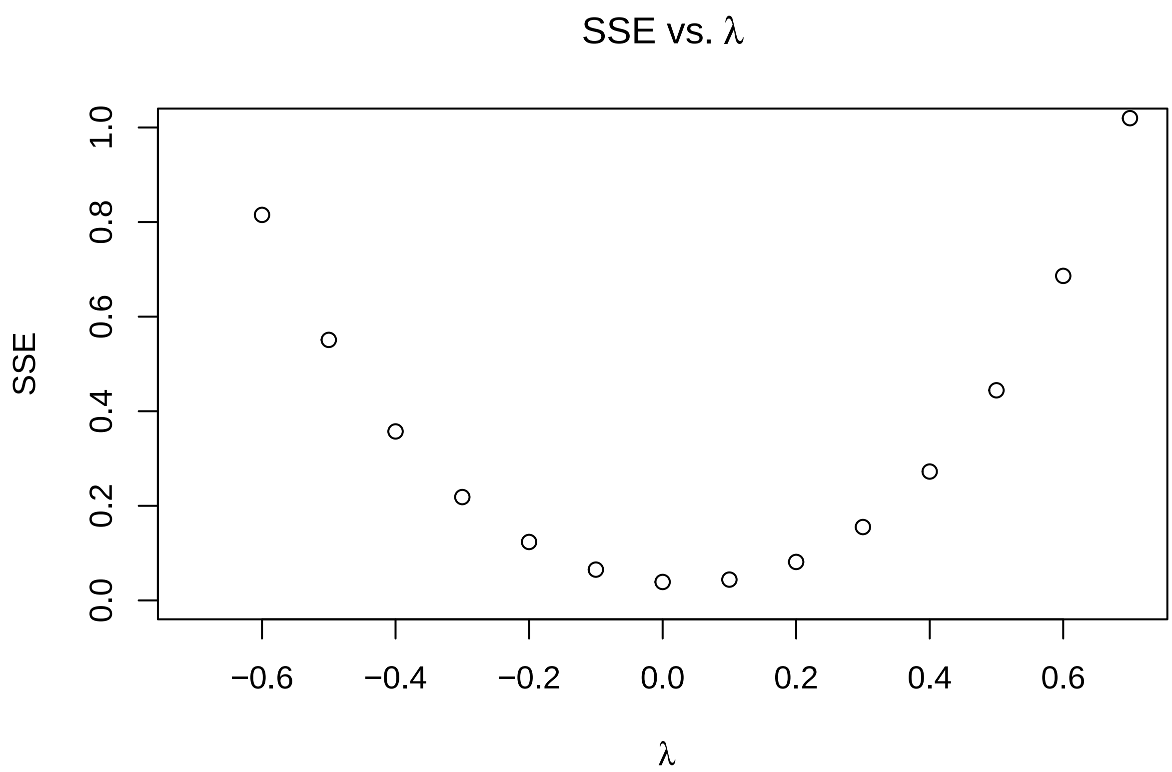
Part (5)

```
lambda_set <- seq(from=-3, to=3, by=.1)
sse_calc <- rep(NA, length(lambda_set))

for(i in 1:length(lambda_set)){
  Wi_calc <- std_obs_wi(y, lambda_set[i])
  sse_calc[i] <- sse(x, Wi_calc)
}
```

Part (6)

```
Q3_plot <- plot(lambda_set, sse_calc, xlab = expression(~lambda),
               ylab= "SSE", main= expression(paste("SSE vs.",~lambda)),
               xlim=c(-.7,.7), ylim=c(0,1))
```



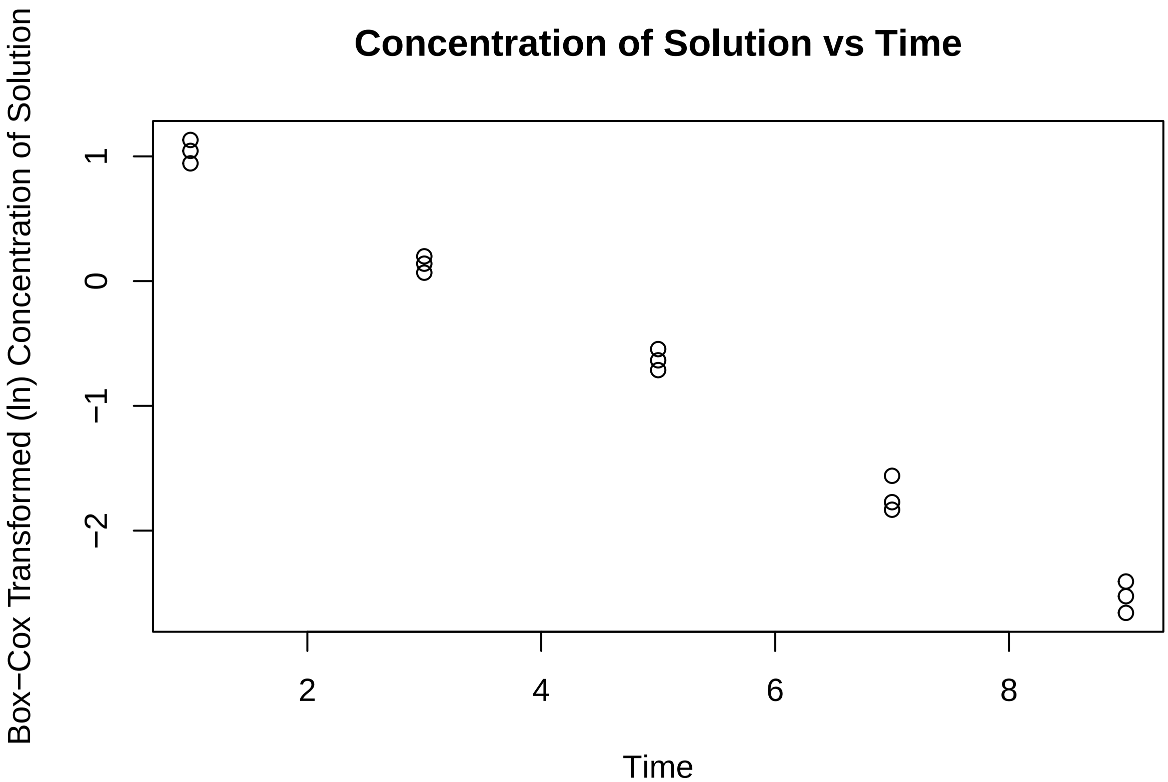

```
## Calculating the min lambda
lambda_set[which.min(sse_calc)]
```

```
## [1] 0
```

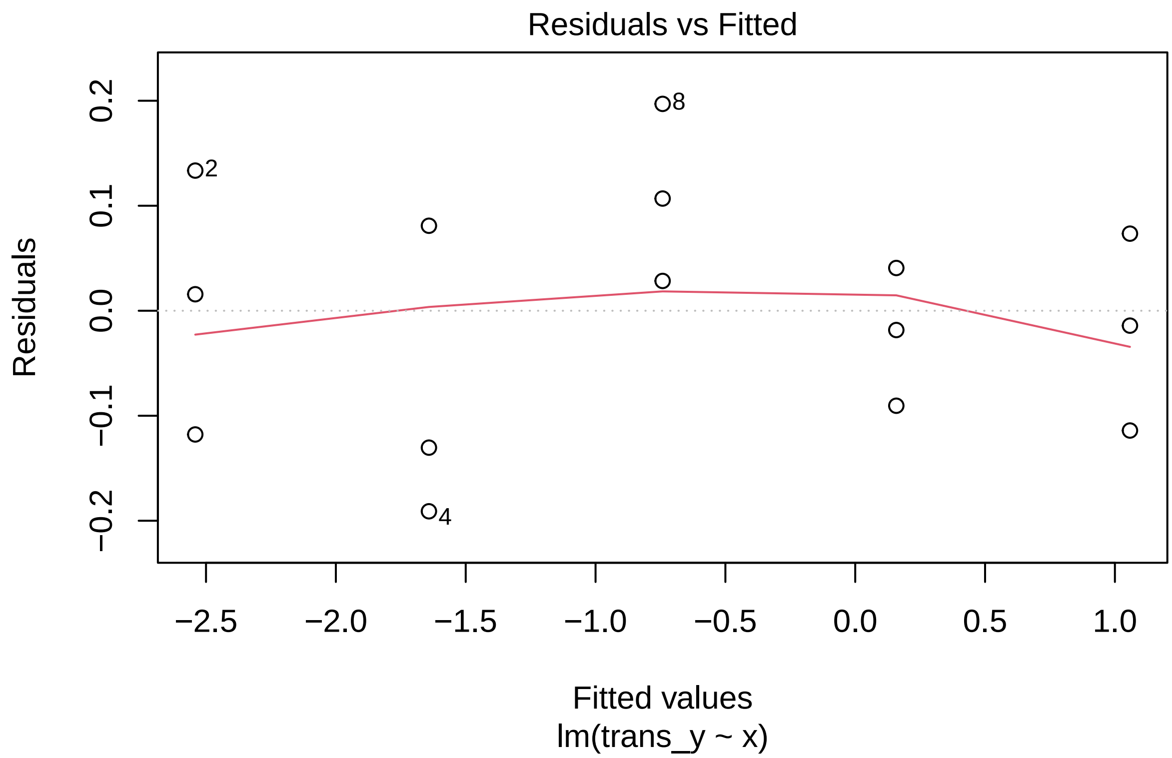
The value of λ that minimizes our SSE is 0.

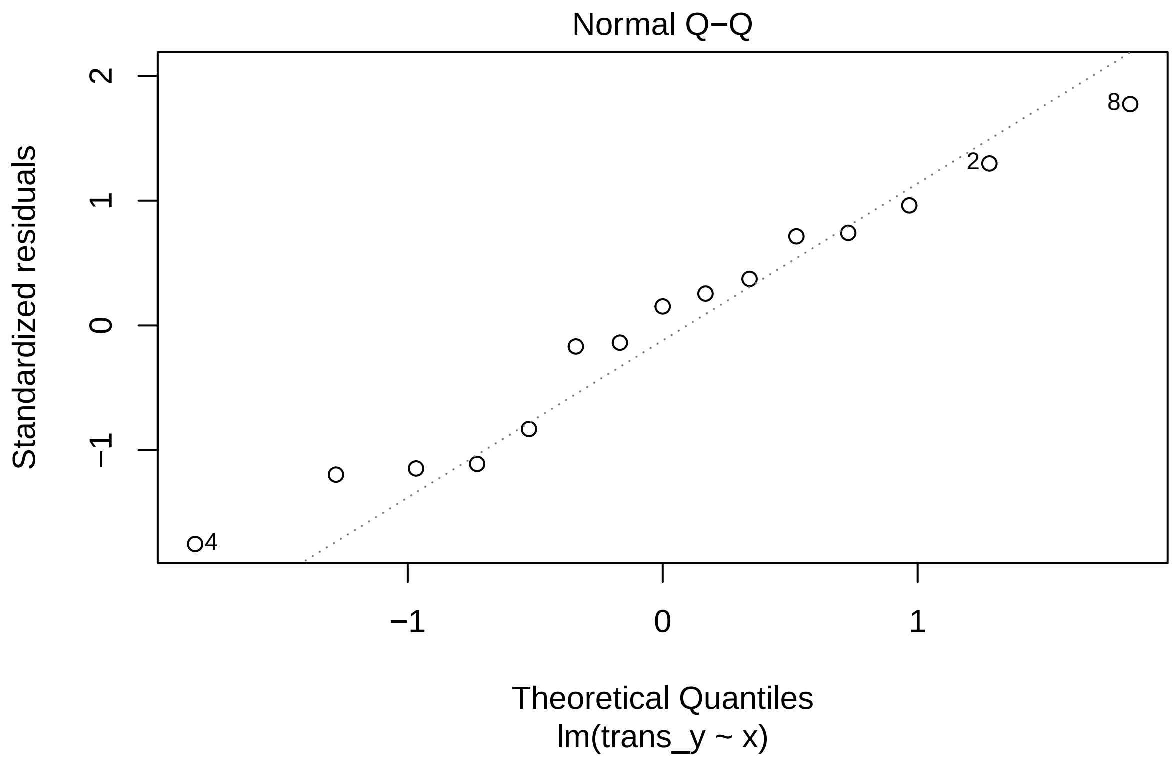
Part (7)

```r
## plotting the Suggested Box-Cox transformation
trans_y <- log(y)
plot(x, trans_y, main = "Concentration of Solution vs Time",
     xlab="Time", ylab="Box-Cox Transformed (ln) Concentration of Solution")
```




```r
bc_model <- lm(trans_y ~ x)
plot(bc_model, which=c(1,2))
```


Residuals vs Fitted
Residuals
0.2
0.1
0.0
−0.1
−0.2
2
8
4
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
Fitted values
lm(trans_y ~ x)




```
hist(resid(bc_model), main='Transformed Model: Histogram of Residuals', xlab='Residuals')
```

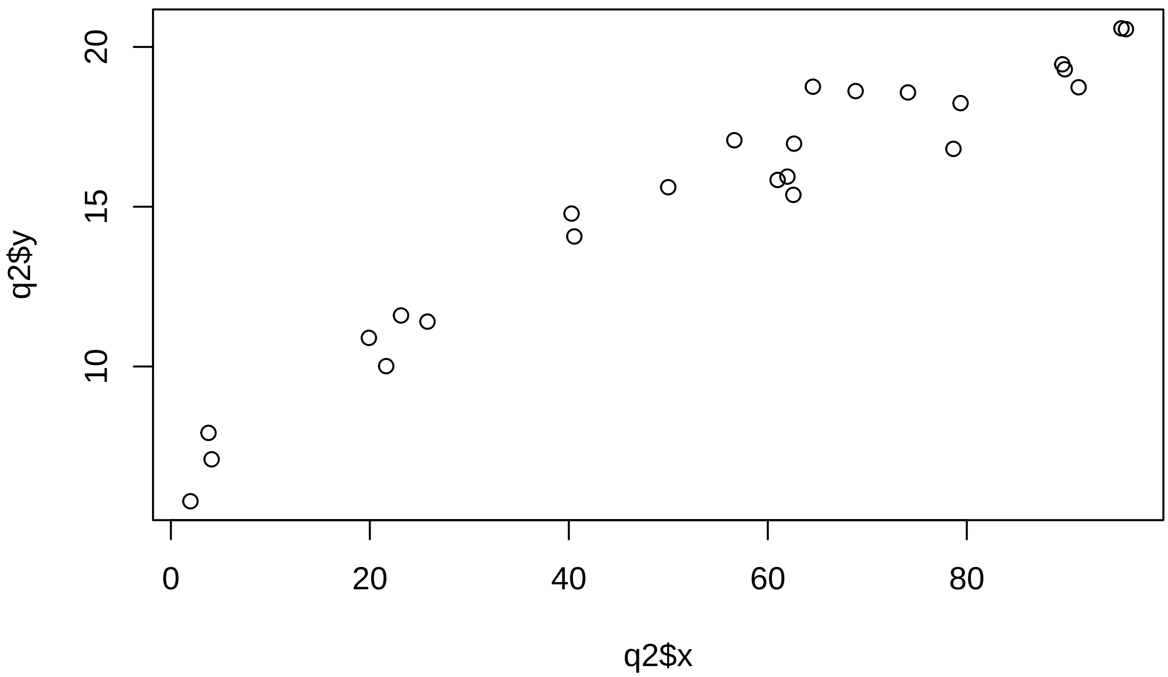
CONDITIONS

Section 2 - Transforming X - The Box-Tidwell Transformation

```
q2 <- read.csv("./Lab4q2.csv")
```

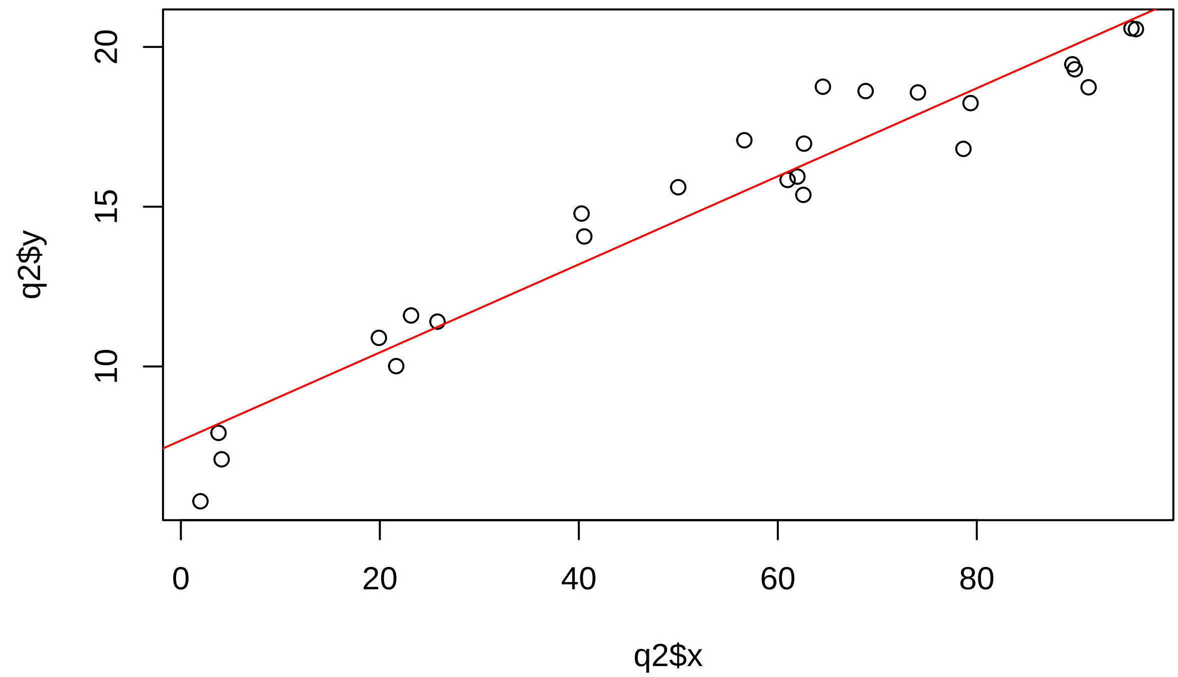
Part (1)

```
#plotting the data
plot(q2$x, q2$y)
```



Part (2)

```
plot(q2$x, q2$y)
abline(lin_model<-lm(q2$y~q2$x), col = "red")
```



Part (3)

```
alpha_bt <- function(xi, yi, aj) {
    x.a <- xi^aj
    b <- x.a*log(xi)
    model1 <- lm(yi~x.a)
    model2 <- lm(yi~x.a+b)
    gamma <- coef(model2)[3]
    new_a <- gamma/coef(model1)[2] + aj
    return(new_a)
}
```

Part (4)

```
a_vec <- NULL
a_vec[1] <- 1
for (i in 1:9) {
    next_a <- alpha_bt(q2$x, q2$y, a_vec[i])
    a_vec[i+1] <- next_a
}

#model_bt <- lm()
```

fit the regression model and conduct a residual analysis