Homework 5

CS 498, Spring 2018, Xiaoming Ji

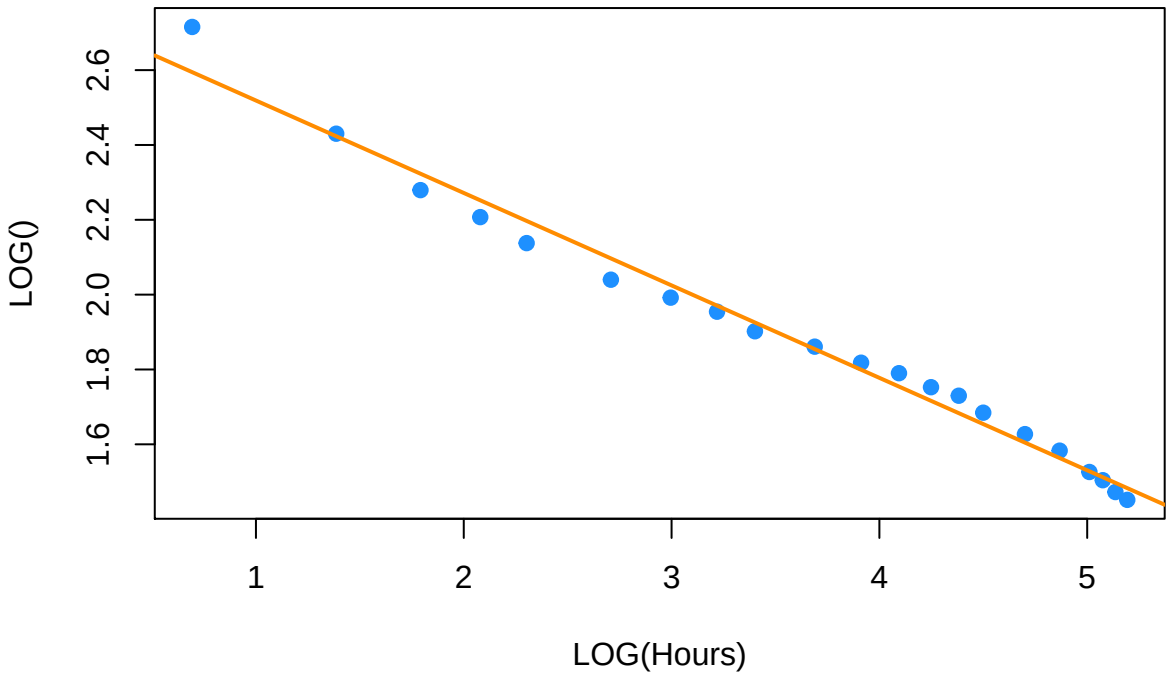
Problem 1

At http://www.statsci.org/data/general/brunhild.html, you will find a dataset that measures the concentration of a sulfate in the blood of a baboon named Brunhilda as a function of time. Build a linear regression of the log of the concentration against the log of time.

(a)

Prepare a plot showing (a) the data points and (b) the regression line in log-log coordinates.

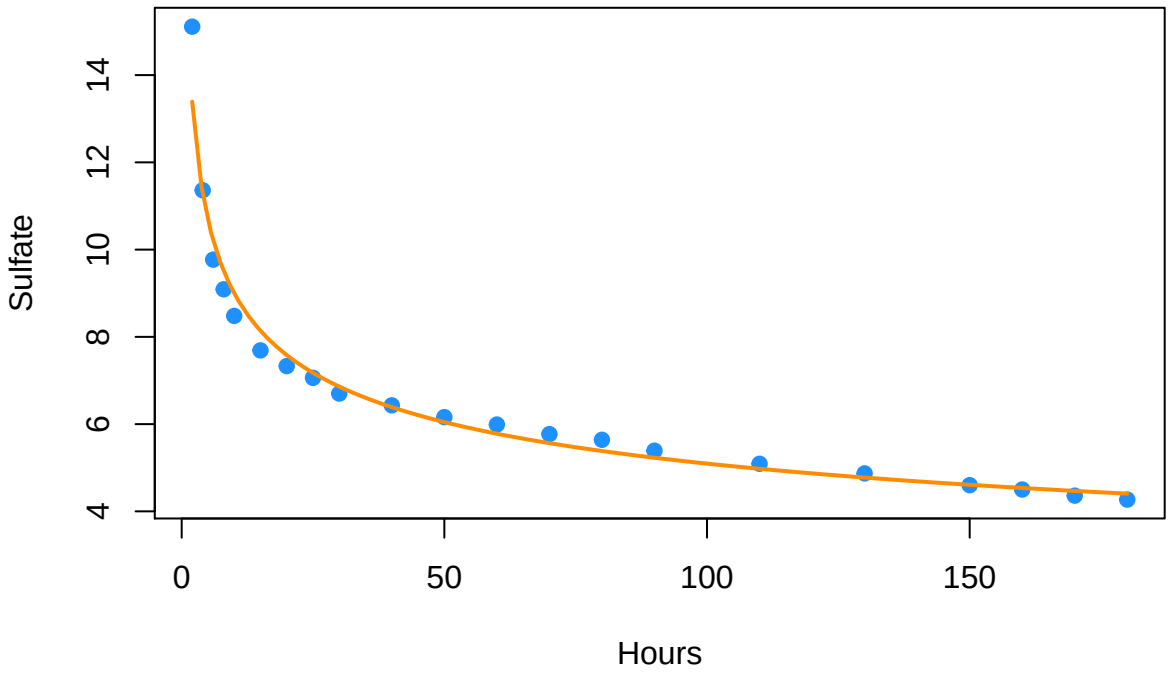
```
brunhild_df = read.delim("brunhild.txt")
model_a = lm(log(Sulfate) ~ log(Hours), data = brunhild_df)
plot(log(brunhild_df$Hours), log(brunhild_df$Sulfate), xlab = "LOG(Hours)", ylab = "LOG()",
     col = "dodgerblue", pch = 20, cex = 1.5)
abline(model_a, col = "darkorange", lwd = 2)
```



(b)

Prepare a plot showing (a) the data points and (b) the regression curve in the original coordinates.

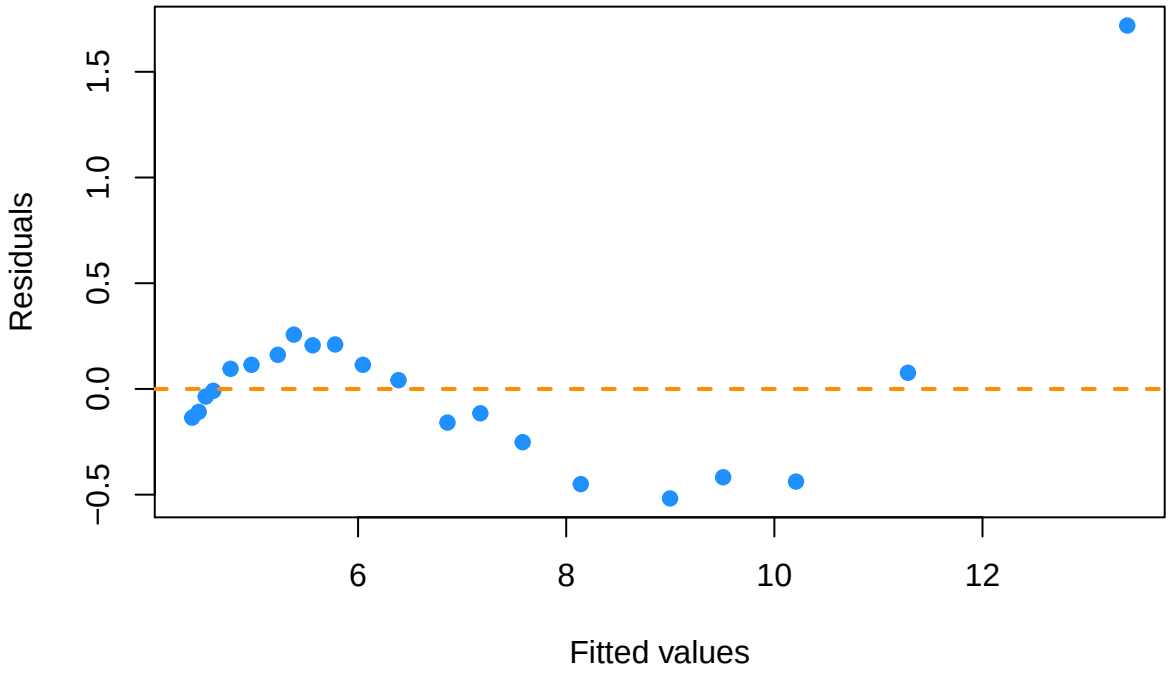
```
plot(brunhild_df, col = "dodgerblue", pch = 20, cex = 1.5)
curve(exp(model_a$coefficients[1] + log(x) * model_a$coefficients[2]),
      from = min(brunhild_df$Hours), to = max(brunhild_df$Hours), add = TRUE,
      col = "darkorange", lwd = 2)
```



(c)

Plot the residual against the fitted values in log-log and in original coordinates.

```
fit_values = exp(model_a$coefficients[1] + log(brunhild_df$Hours) *
                   model_a$coefficients[2])
res_values = brunhild_df$Sulfate - fit_values
plot(fit_values, res_values, col = "dodgerblue",
pch = 20, cex = 1.5, xlab = "Fitted values", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```



(d)

Use your plots to explain whether your regression is good or bad and why.

Answer: If we just eye-exam plot (a) and (b), it seems the model fits the data very well. However, good model also need to follow linearity and constant variance assumptions which is exposed by plot (c).

- To check linearity assumptions, we need exam at any fitted value , the mean of the residuals should be roughly 0. This is not the case for most points in this plot.
- To check constant variance assumption in this plot, we need exam at every fitted value, the spread of the residuals should be roughly the same. This is not the case in this plot and we see the magnitude of residuals increase when fitted value increase.

For these reasons, we believe this is a **BAD** regression.

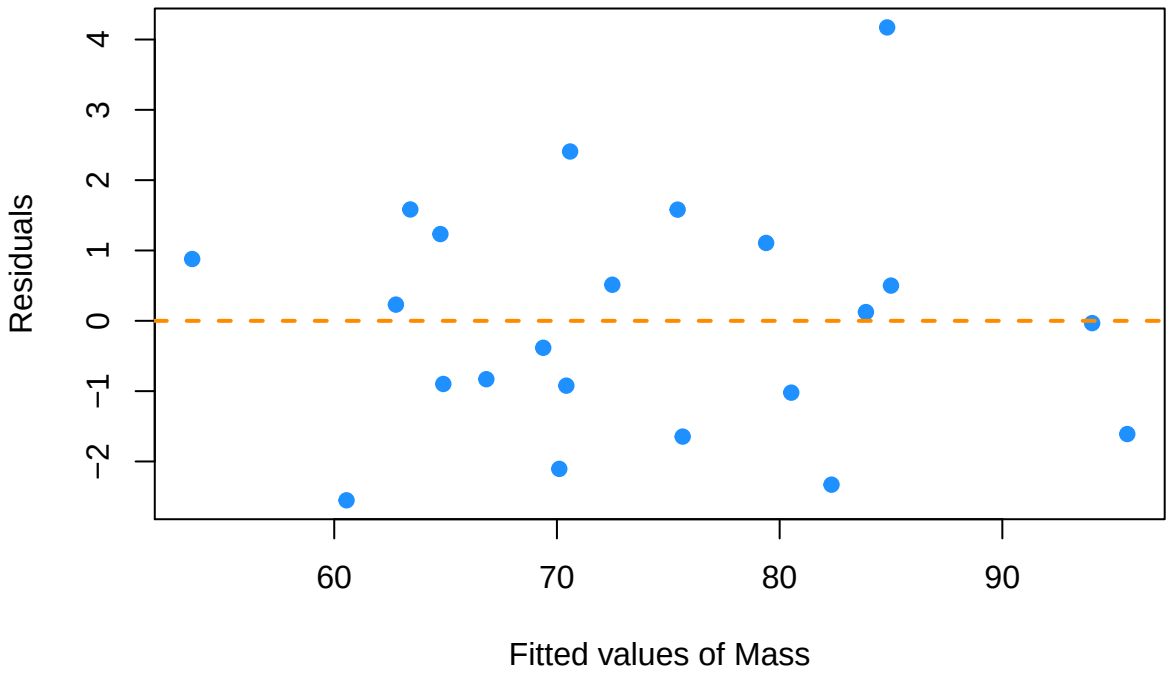
Problem 2

At http://www.statsci.org/data/oz/physical.html, you will find a dataset of measurements by M. Larner, made in 1996. These measurements include body mass, and various diameters. Build a linear regression of predicting the body mass from these diameters.

(a)

Plot the residual against the fitted values for your regression.

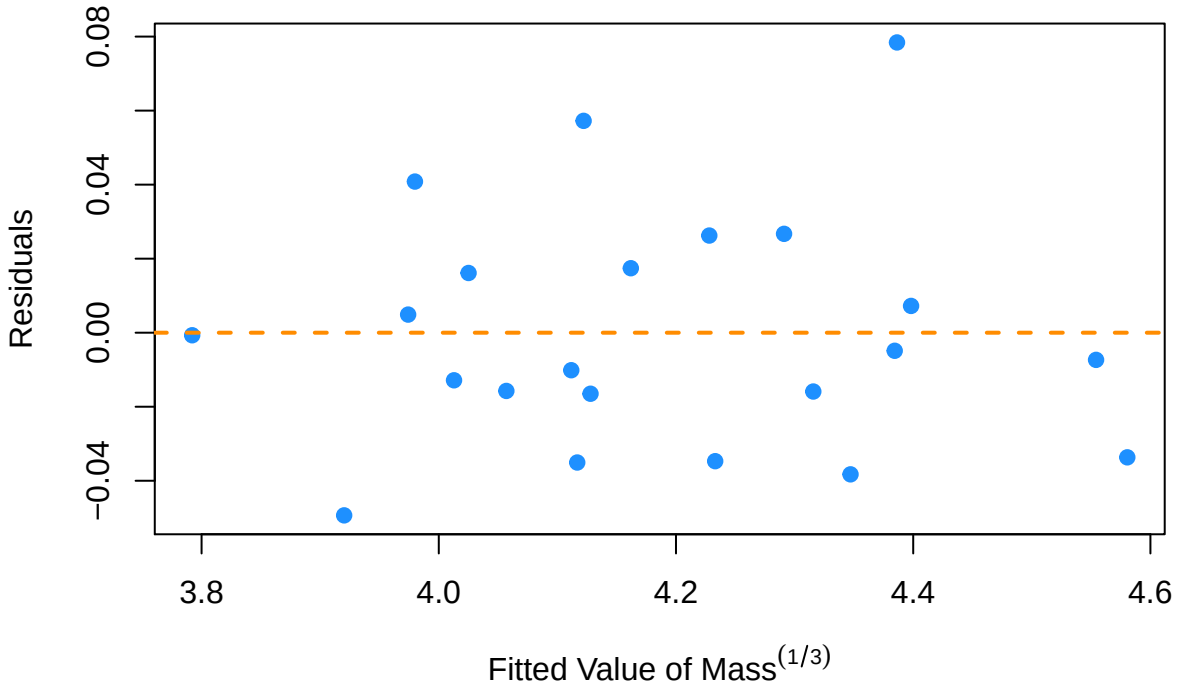
```
physical_df = read.delim("physical.txt")
model_a = lm(Mass ~ ., data = physical_df)
plot(model_a$fitted.values, model_a$residuals, col = "dodgerblue",
     pch = 20, cex = 1.5, xlab = "Fitted values of Mass", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```



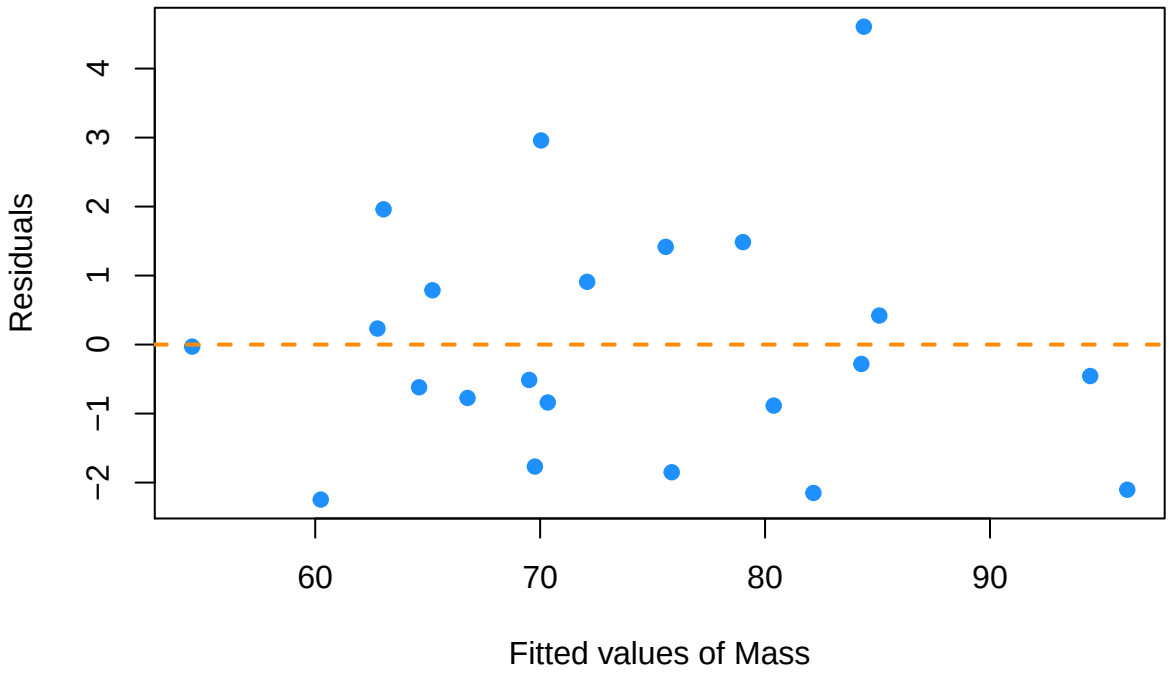
(b)

Now regress the cube root of mass against these diameters. Plot the residual against the fitted values in both these cube root coordinates and in the original coordinates.

```
model_b = lm(Mass ^ (1/3) ~ ., data = physical_df)
plot(model_b$fitted.values, model_b$residuals, col = "dodgerblue",
     pch = 20, cex = 1.5, xlab = expression(paste("Fitted Value of ",Mass^(1/3))),
     ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```




```
plot(model_b$fitted.values ^ 3, physical_df$Mass - model_b$fitted.values ^ 3 ,
     col = "dodgerblue", pch = 20, cex = 1.5, xlab = "Fitted values of Mass",
     ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```



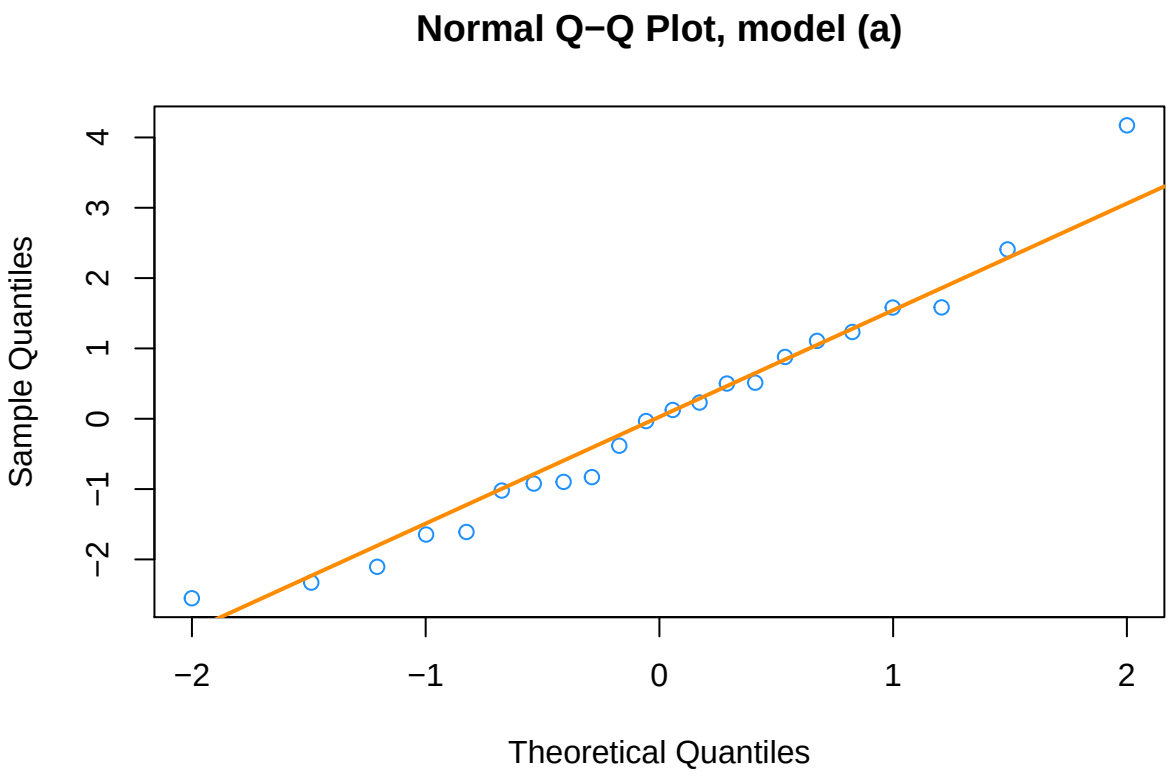
(c)

Use your plots to explain which regression is better.

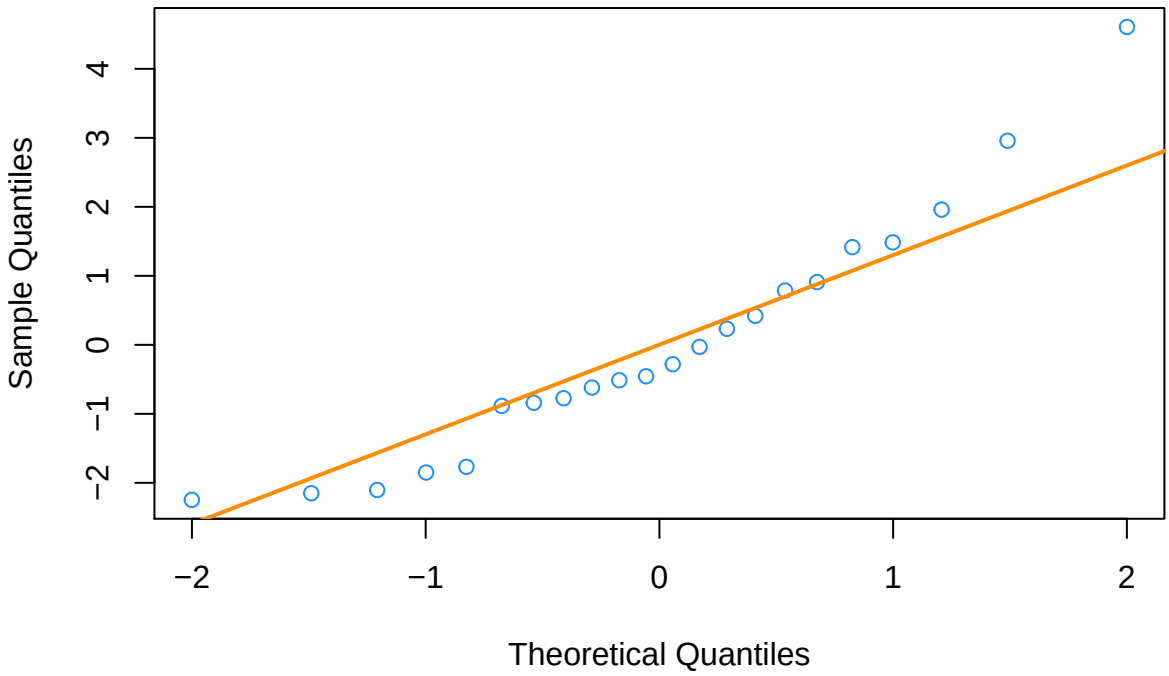
Answer: We don't see much different in these 2 plots in terms of linearity and variance. As a matter of fact, (a) has slightly smaller residual standard deviation (1.6550594 vs. 1.7371571). Since model (a) is simpler, model **(a)** is better.

As another investigation, if we make Q-Q plots of these 2 regressions.

```
qqnorm(model_a$residuals, main = "Normal Q-Q Plot, model (a)", col = "dodgerblue")
qqline(model_a$residuals, col = "darkorange", lwd = 2)
```




```
qqnorm(physical_df$Mass - model_b$fitted.values ^ 3, main = "Normal Q-Q Plot, model (b)",
       col = "dodgerblue")
qqline(physical_df$Mass - model_b$fitted.values ^ 3, col = "darkorange", lwd = 2)
```



- Model (a) show better normality of errors and thus is better than (b).

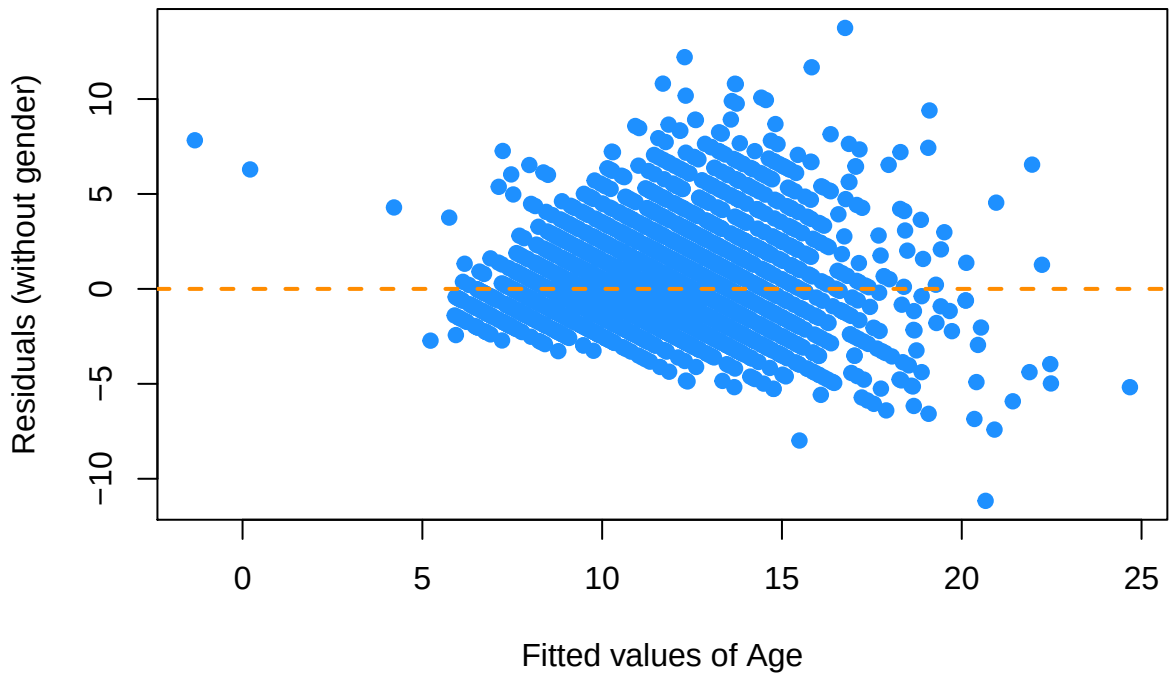
Problem 3

At https://archive.ics.uci.edu/ml/datasets/Abalone, you will find a dataset of measurements by W. J. Nash, T. L. Sellers, S. R. Talbot, A. J. Cawthorn and W. B. Ford, made in 1992. These are a variety of measurements of blacklip abalone (Haliotis rubra; delicious by repute) of various ages and genders.

(a)

Build a linear regression predicting the age from the measurements, ignoring gender. Plot the residual against the fitted values.

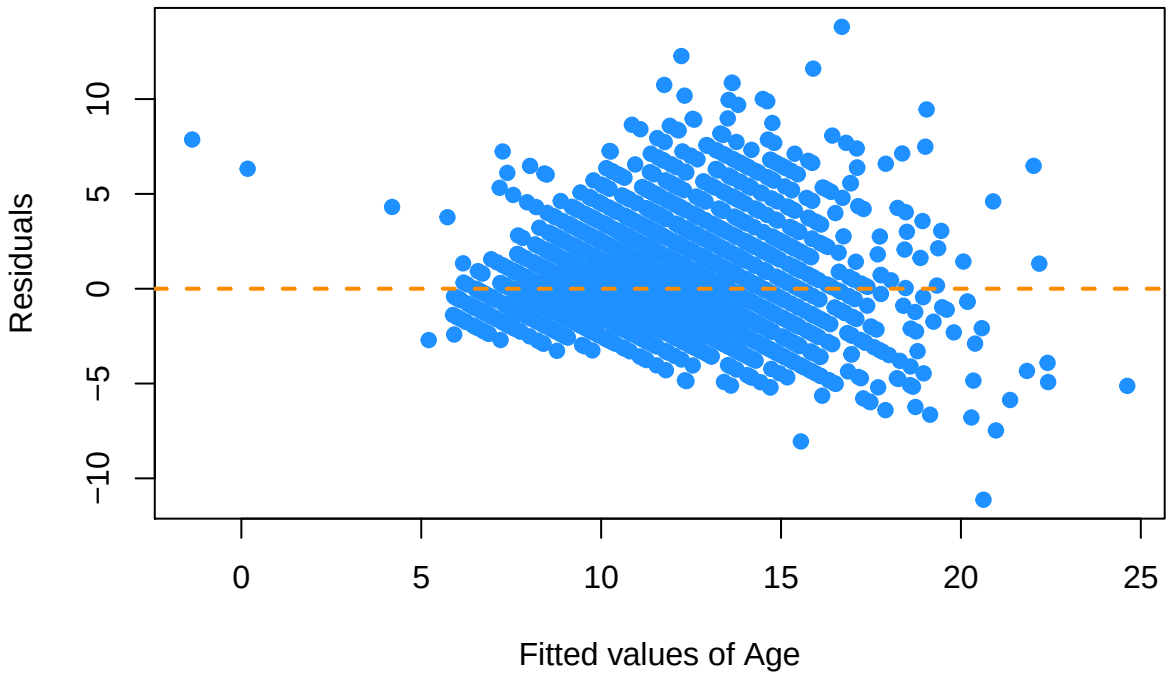
```
abalone_data = read.csv("abalone.data", header=FALSE)
abalone_data$V9 = abalone_data$V9 + 1.5
abalone_data$V1 = as.integer(abalone_data$V1)
abalone_data$V1[abalone_data$V1 == 3] = -1
abalone_data$V1[abalone_data$V1 == 2] = 0

model_a = lm(V9 ~ V2 + V3 + V4 + V5 + V6 + V7 + V8, data = abalone_data)
plot(model_a$fitted.values, model_a$residuals, col = "dodgerblue",
     pch = 20, cex = 1.5, xlab = "Fitted values of Age",
     ylab = "Residuals (without gender)")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```



(b)

Build a linear regression predicting the age from the measurements, including gender. There are three levels for gender; I'm not sure whether this has to do with abalone biology or difficulty in determining gender. You can represent gender numerically by choosing 1 for one level, 0 for another, and -1 for the third. Plot the residual against the fitted values.

```r
model_b = lm(V9 ~ ., data = abalone_data)
plot(model_b$fitted.values, model_b$residuals, col = "dodgerblue",
     pch = 20, cex = 1.5, xlab = "Fitted values of Age", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```

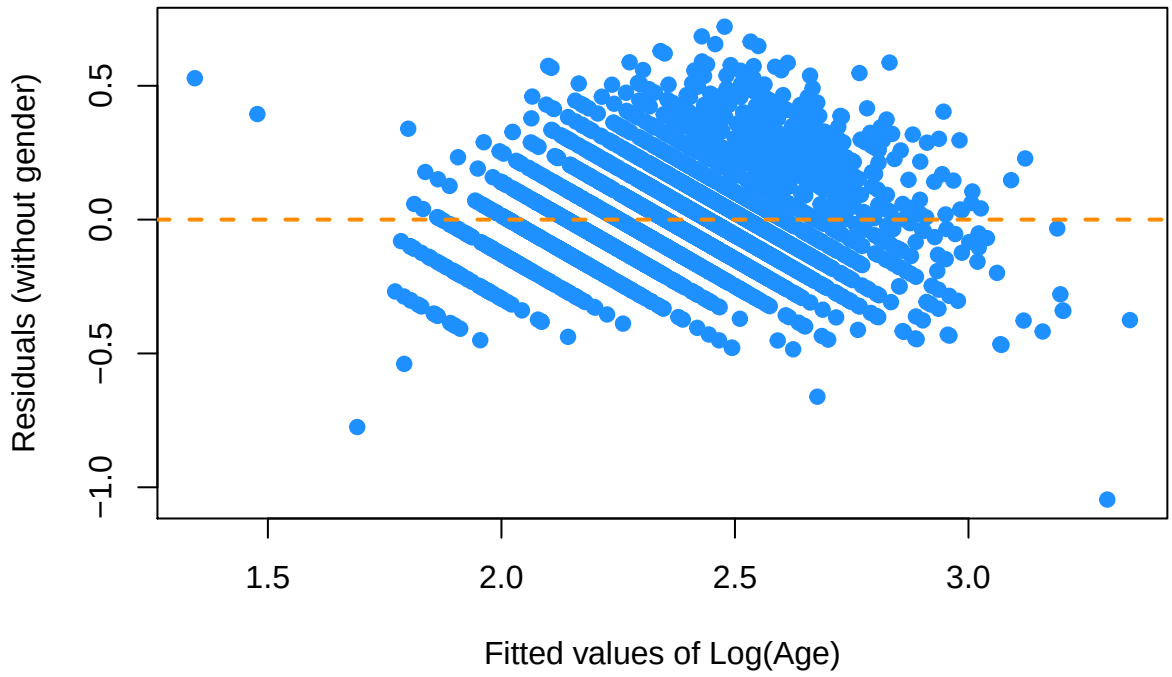


Note: we actually make gender a categorical variable and build the model.

(c)

Now build a linear regression predicting the log of age from the measurements, ignoring gender. Plot the residual against the fitted values.

```r
model_c = lm(log(V9) ~ V2 + V3 + V4 + V5 + V6 + V7 + V8, data = abalone_data)
plot(model_c$fitted.values, model_c$residuals, col = "dodgerblue",
     pch = 20, cex = 1.5, xlab = "Fitted values of Log(Age)",
     ylab = "Residuals (without gender)")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```

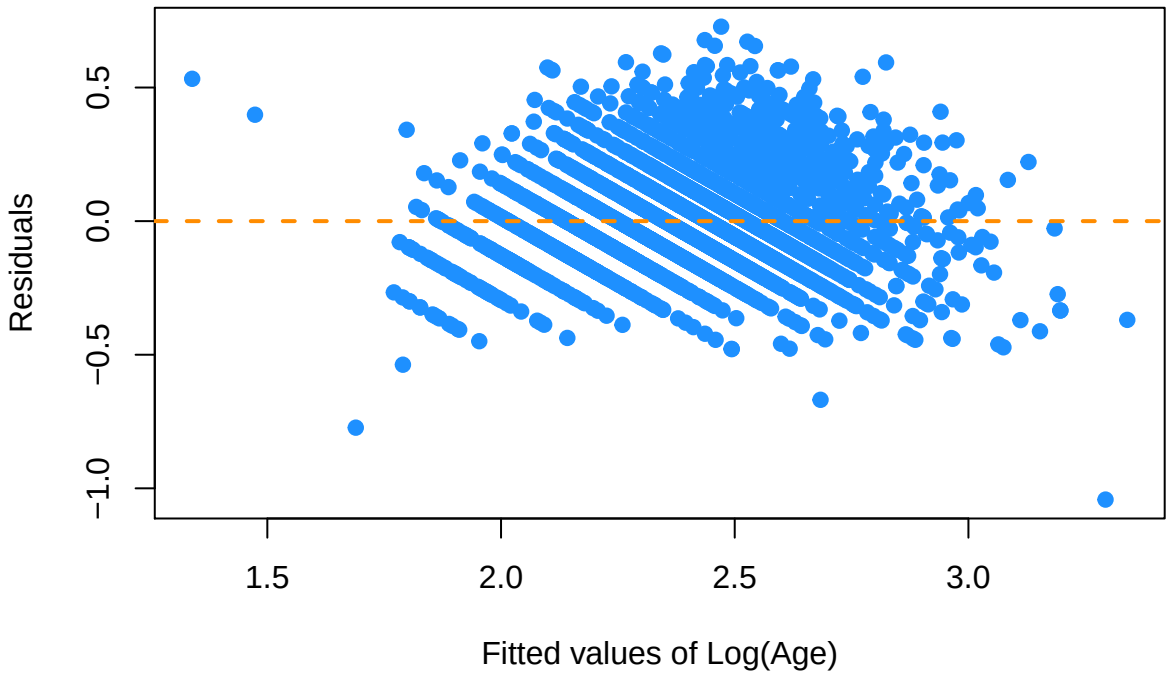



```
# plot(exp(model_c$fitted.values), abalone_data$V9 - exp(model_c$fitted.values),
#      col = "dodgerblue", pch = 20, cex = 1.5, xlab = "Fitted values of Age",
#      ylab = "Residuals (without gender)")
# abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```

(d)

Now build a linear regression predicting the log age from the measurements, including gender, represented as above. Plot the residual against the fitted values.

```
model_d = lm(log(V9) ~ ., data = abalone_data)
plot(model_d$fitted.values, model_d$residuals, col = "dodgerblue",
     pch = 20, cex = 1.5, xlab = "Fitted values of Log(Age)", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```




```
# plot(exp(model_d$fitted.values), abalone_data$V9 - exp(model_d$fitted.values),
#      col = "dodgerblue",pch = 20, cex = 1.5, xlab = "Fitted values of Age",
#      ylab = "Residuals")
# abline(h = 0, lty = 2, col = "darkorange", lwd = 2)
```

(e)

It turns out that determining the age of an abalone is possible, but difficult (you section the shell, and count rings). Use your plots to explain which regression you would use to replace this procedure, and why.

Abswer: The plots of (c) and (d) do look follow linearity assumption better than (a) and (b). In addition, (c) and (d) don't seem have much difference, since (c) has fewer variables and simpler, we would choose **(c)** as the best regression model.

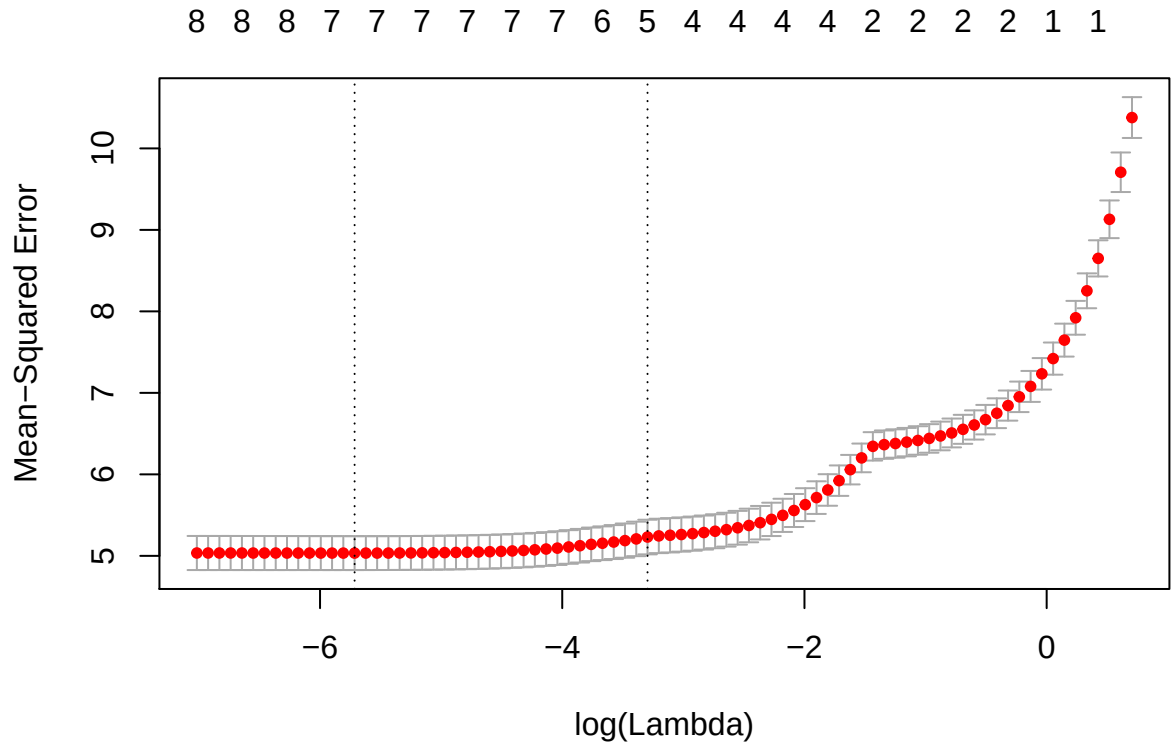
(f)

Can you improve these regressions by using a regularizer? Use glmnet to obtain plots of the cross-validated prediction error.

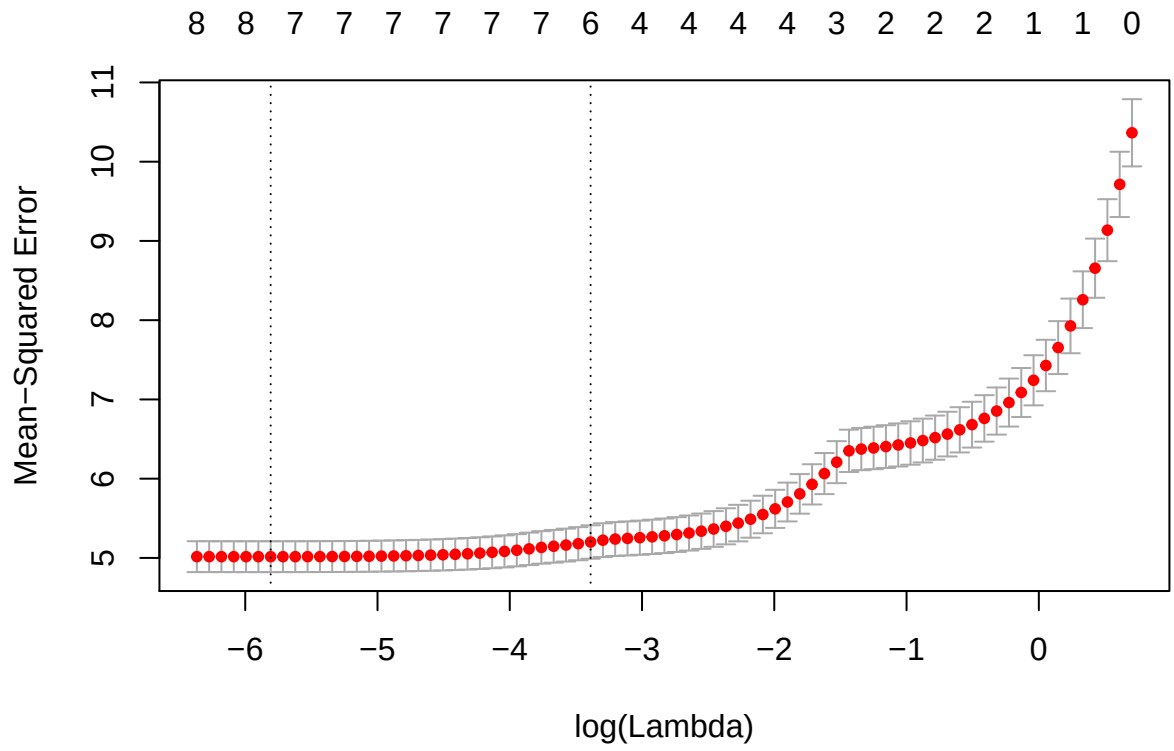
We do glmnet and make plots for the above 4 models.

```
library(glmnet)
set.seed(19720816)

x = as.matrix(abalone_data[,-9])
y = abalone_data$V9

model_a_1 = cv.glmnet(x[,-9], y)
model_b_1 = cv.glmnet(x, y)
model_c_1 = cv.glmnet(x[,-9], log(y))
model_d_1 = cv.glmnet(x, log(y))
```

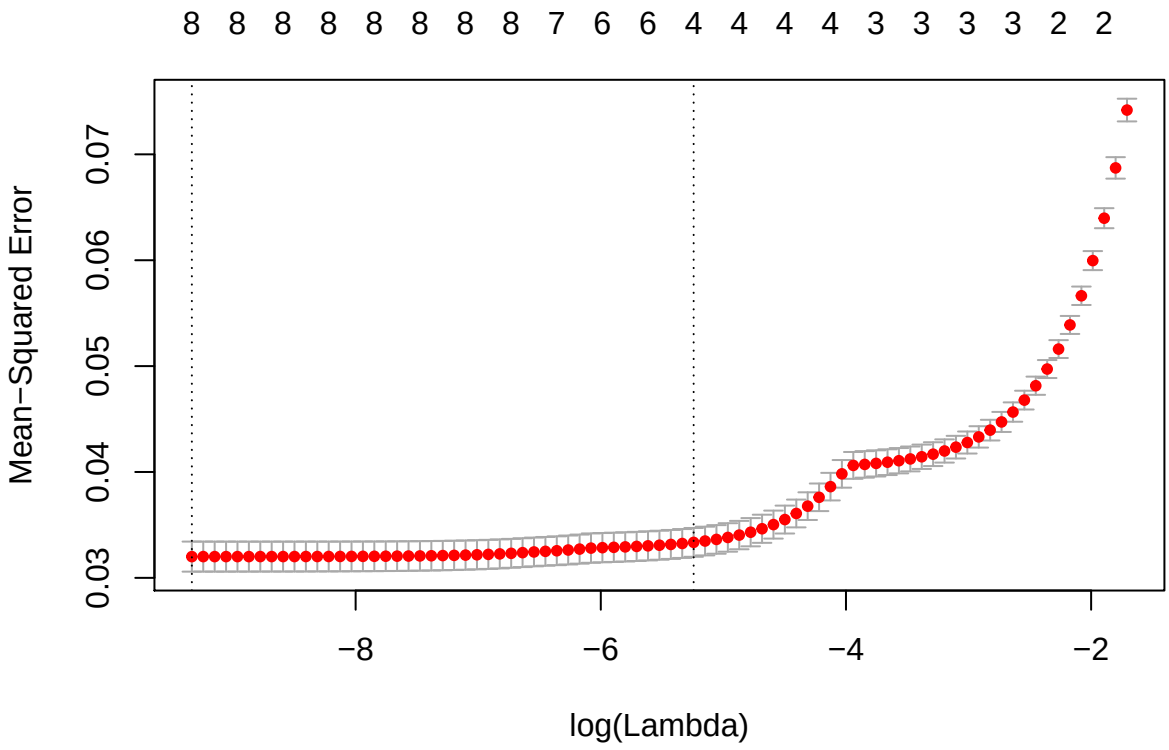
```
plot(model_a_1)
```



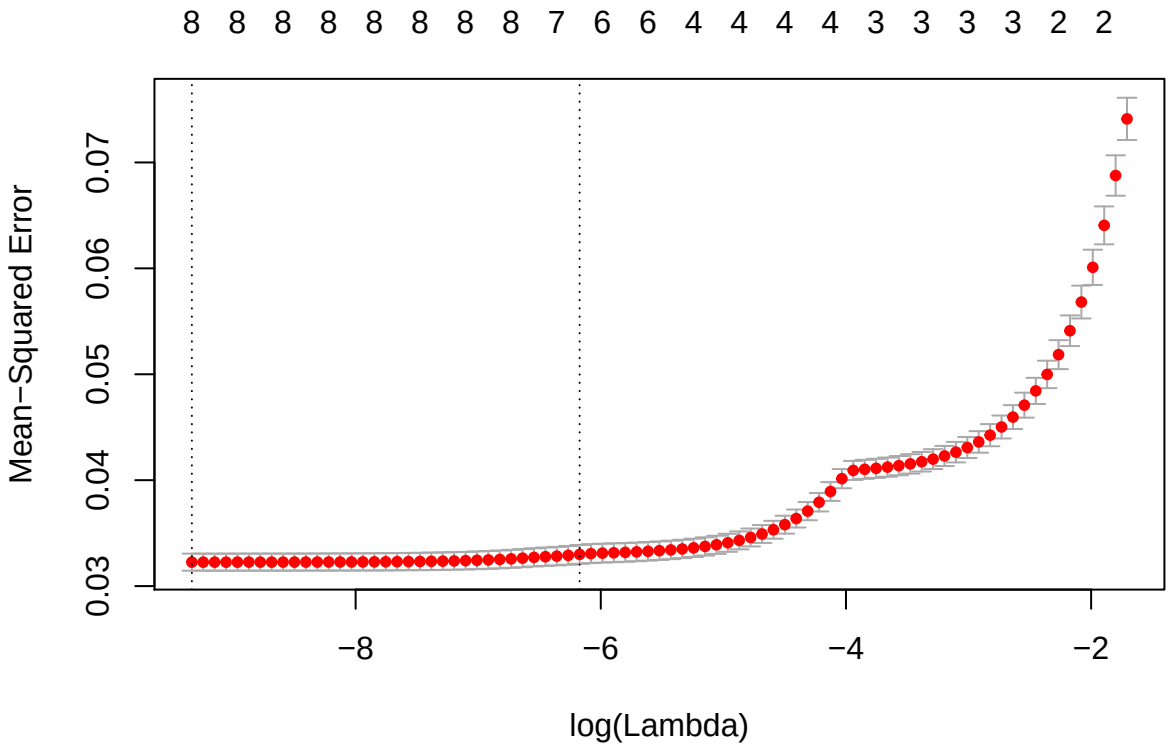

```
plot(model_b_1)
```



```
plot(model_c_1)
```




```
plot(model_d_1)
```



- The experiment shows that model (a) and (d) can be slightly improved by using a regularizer with λ of 0.0032973 and 0.0030043 respectively.