R Notebook

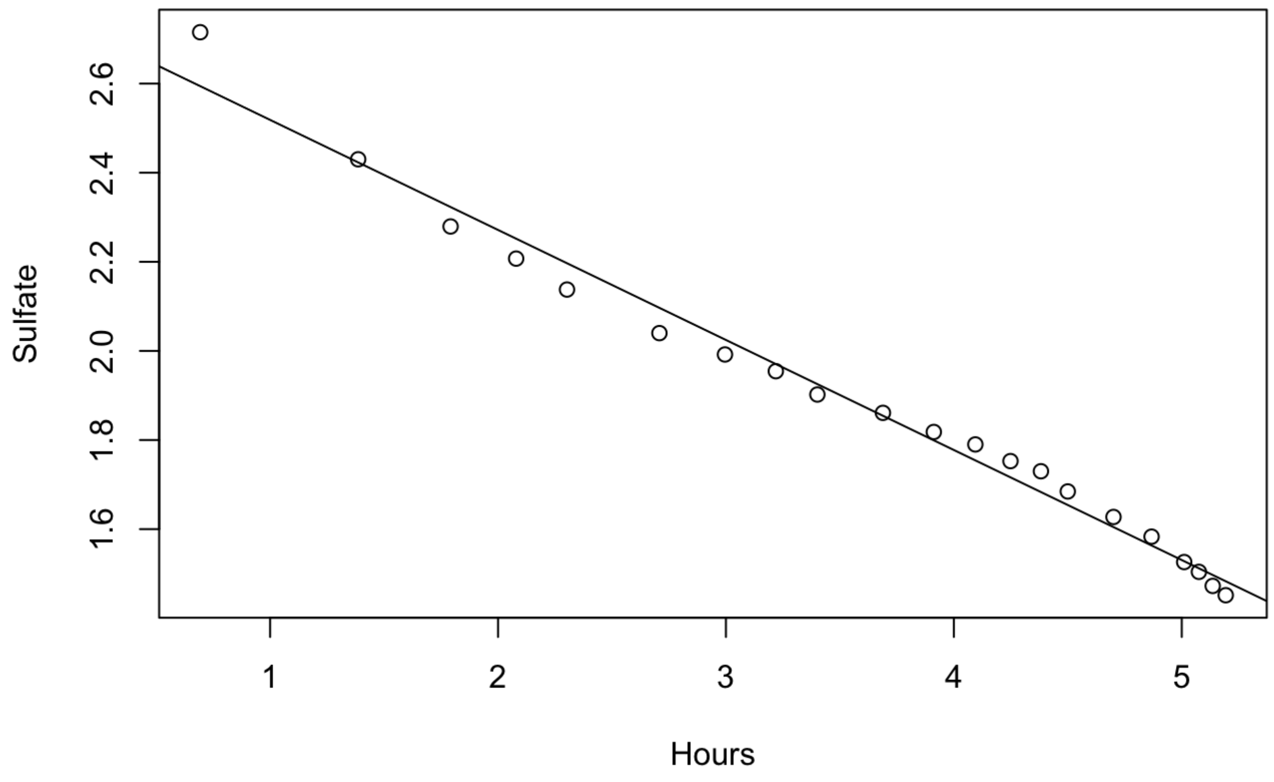
```
library(glmnet)
library(RCurl)
```

7.9.

- Data http://www.statsci.org/data/general/brunhild.html (http://www.statsci.org/data/general/brunhild.html),
- About Data: dataset measures the concentration of a sulfate in the blood of a baboon named Brunhilda as a function of time.

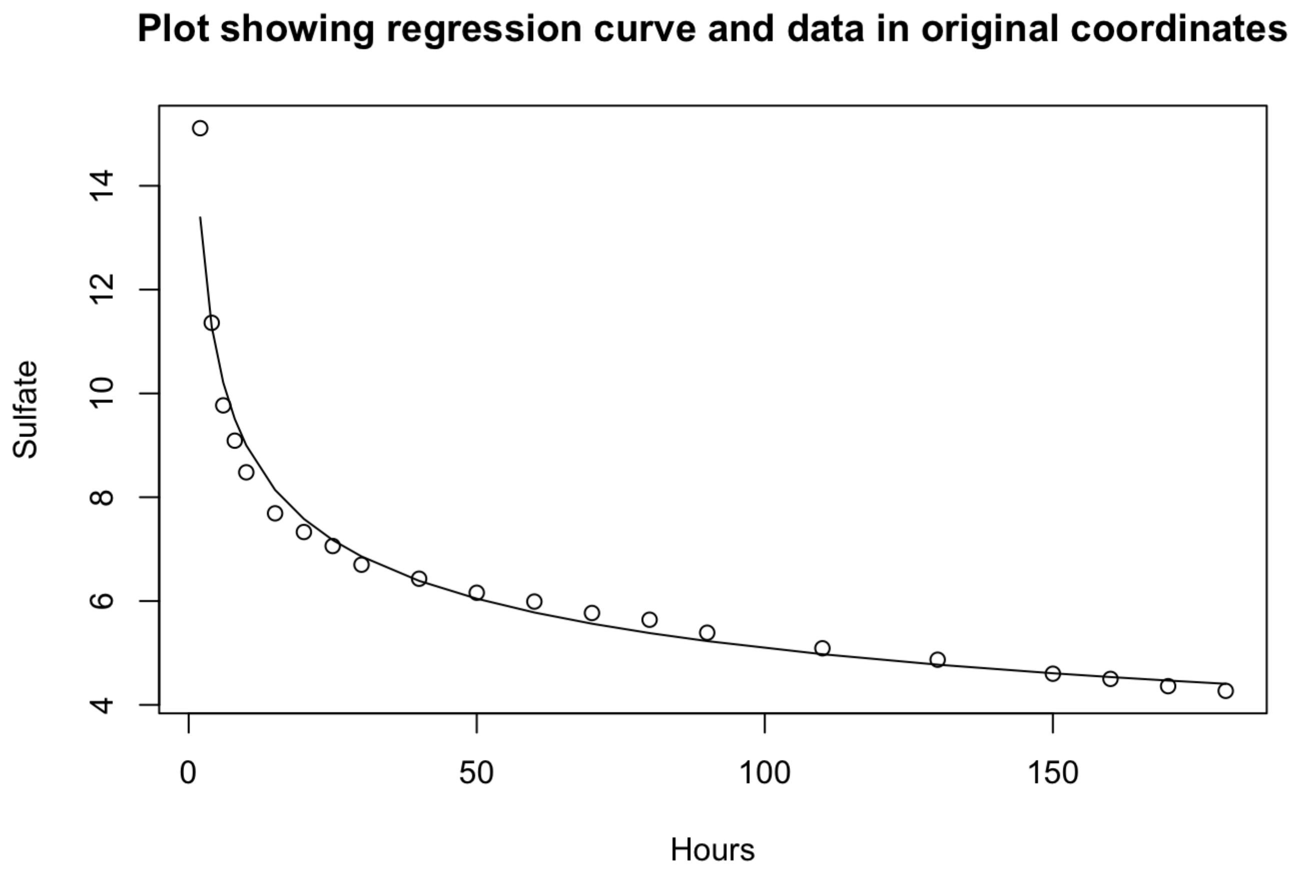
(a) Prepare a plot showing (a) the data points and (b) the regression line in log-log coordinates.

```
url <- getURL("http://www.statsci.org/data/general/brunhild.txt")
data <- read.table(text = url, header = TRUE, stringsAsFactors = FALSE)
plot(log(data) )
abline(lm(formula = Sulfate~Hours, data = log(data)))
```



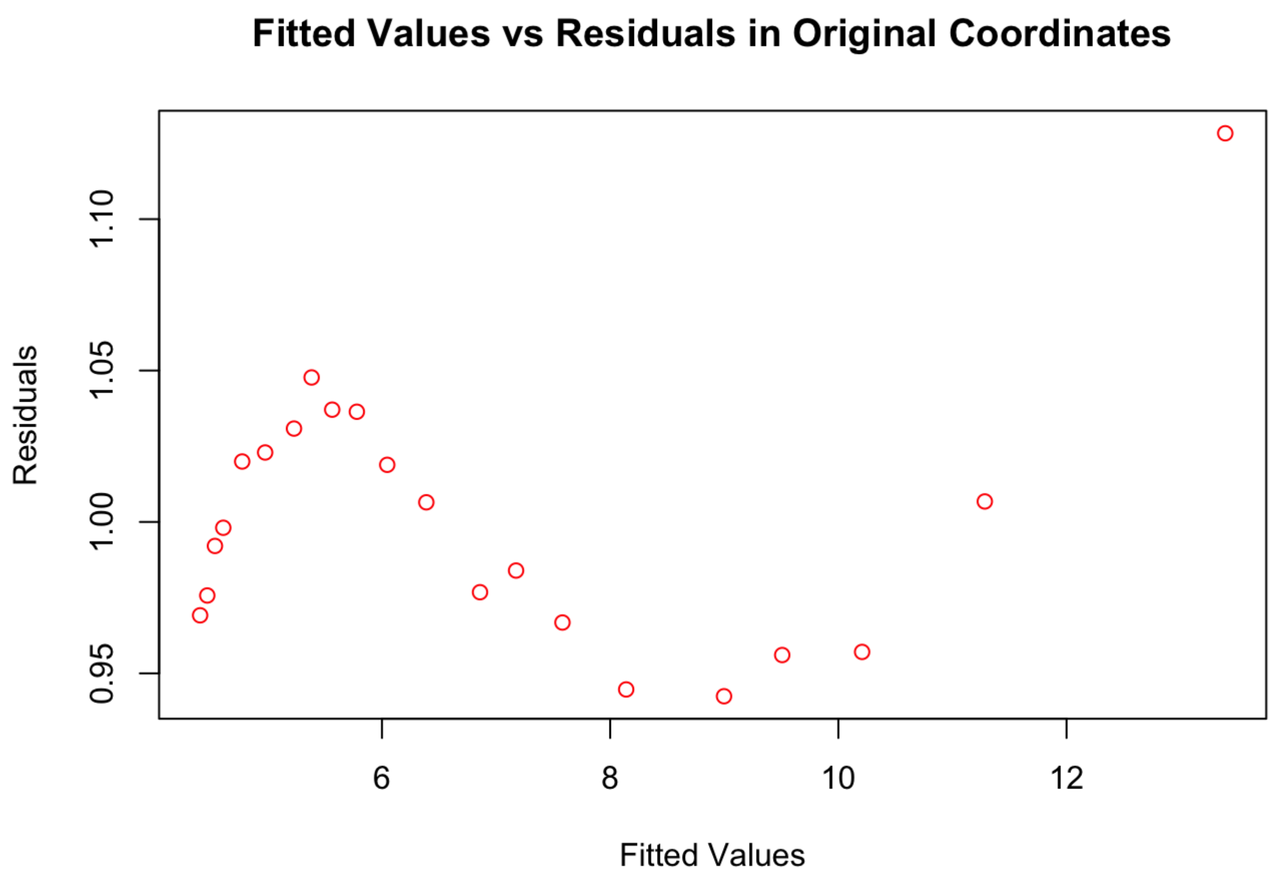
(b) Prepare a plot showing (a) the data points and (b) the regression curve in the original coordinates.

```
model <- lm(formula = Sulfate~Hours, data = log(data))
fitted <- exp(as.numeric(model$fitted.values))

plot(data, main ="Plot showing regression curve and data in original coordinates")
lines(data$Hours, fitted)
```

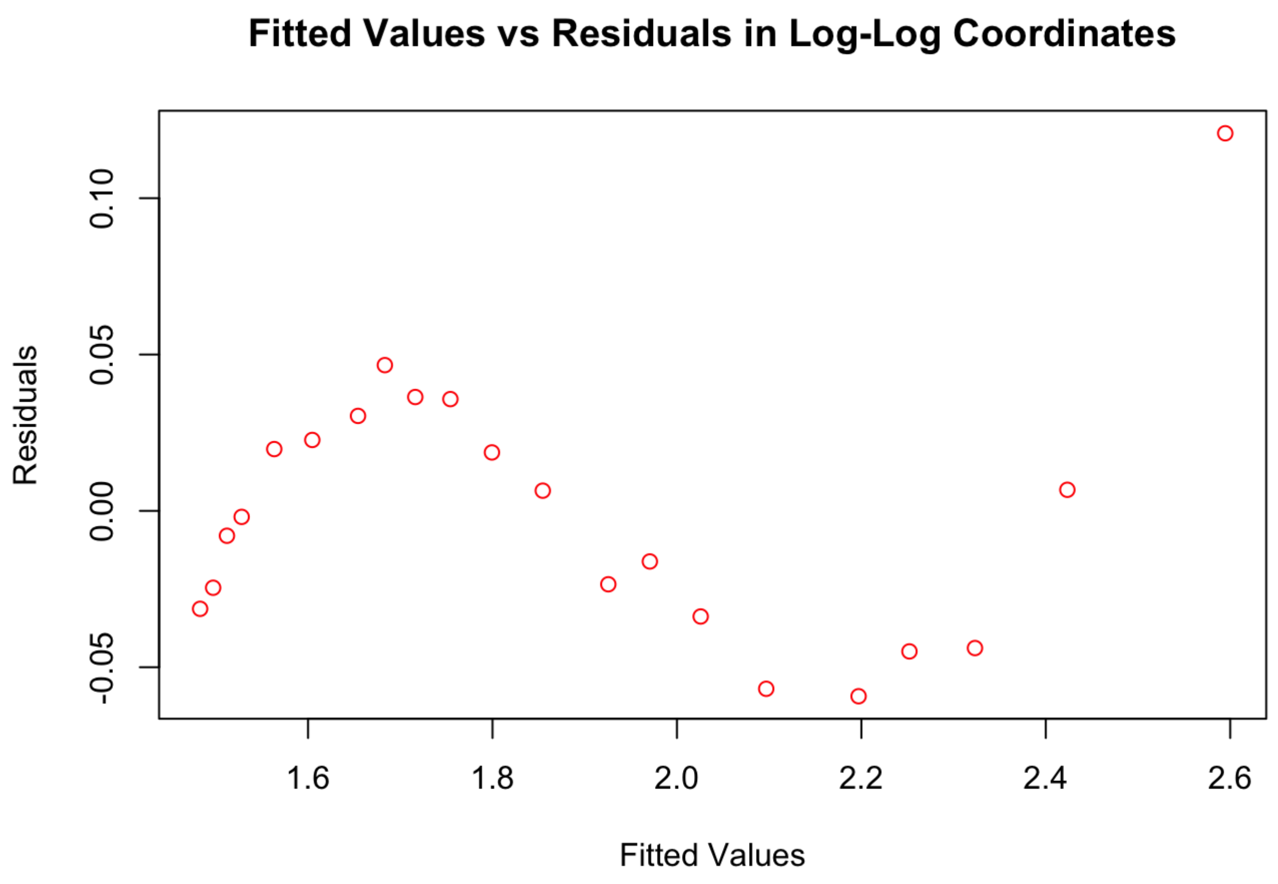


(c) Plot the residual against the fitted values in log-log and in original coordinates.

```
##Original coordinates
plot(x= fitted, y= exp(model$residuals), main = 'Fitted Values vs Residuals in Original
 Coordinates', xlab ='Fitted Values', ylab='Residuals', col='red')
```



```
##Log log coordinates
plot(x= model$fitted.values, y= model$residuals, main = 'Fitted Values vs Residuals in L
og-Log Coordinates', xlab ='Fitted Values', ylab='Residuals', col='red')
```



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

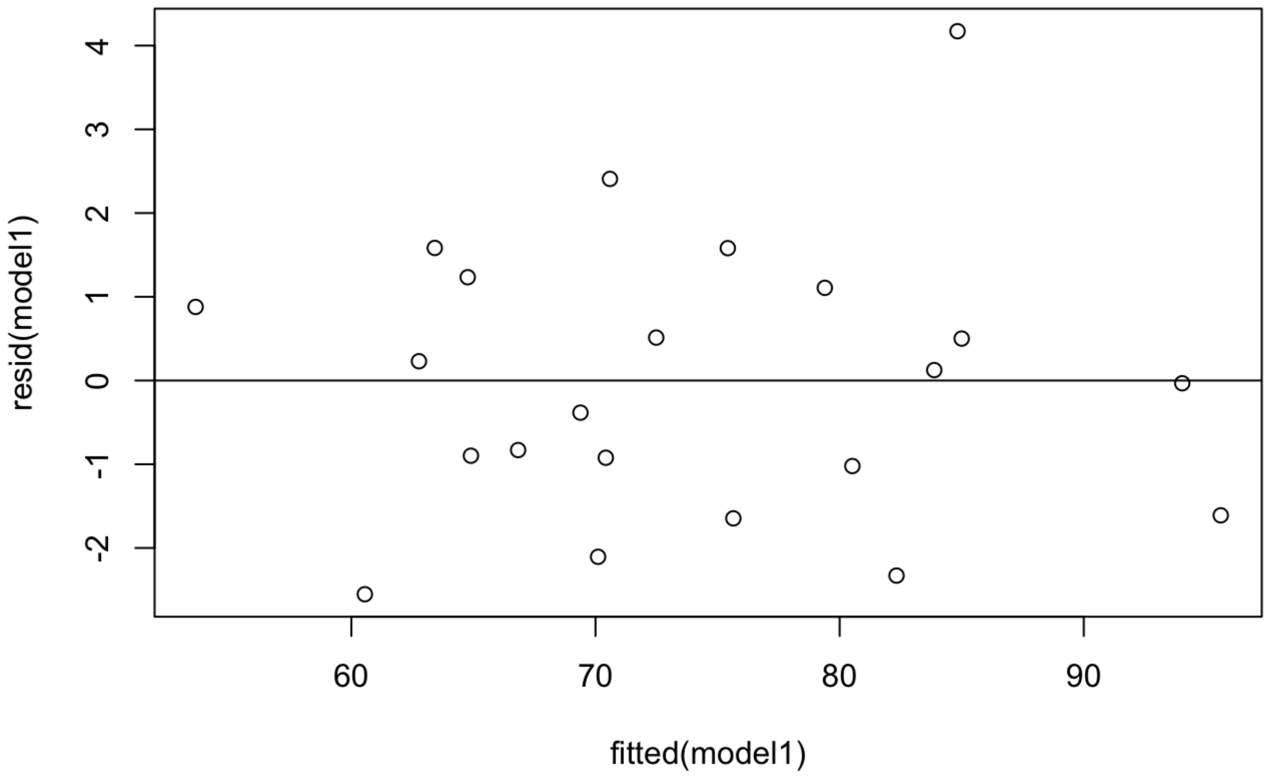
- Regression fits data very well by looking at plot of Part A and B. However, by analyzing residuals against fitted value we foudn our that this regression is not good as we can see randomness and part C plots doesnt follow linearity and constant vaiance assumptions.

7.10

- Data: http://www.statsci.org/data/oz/physical.html (http://www.statsci.org/data/oz/physical.html)
- About data: body measurement dataset

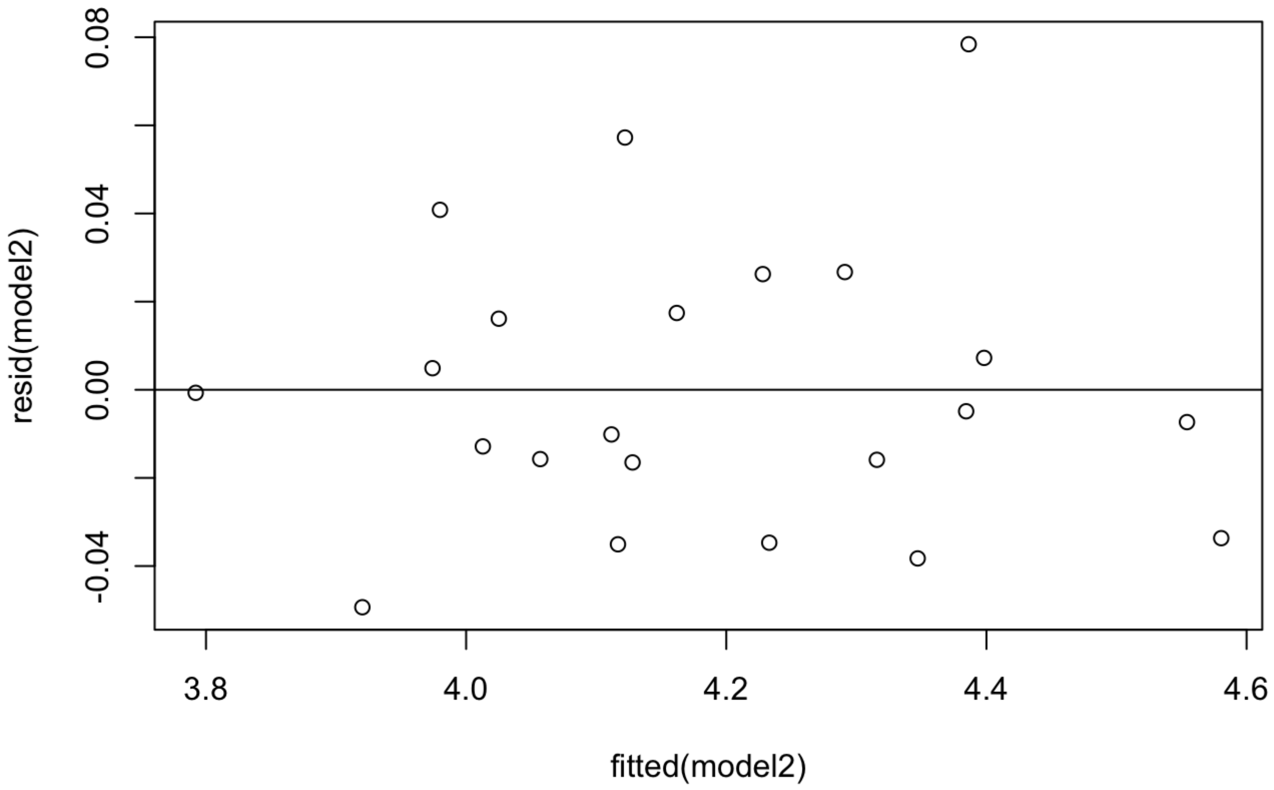
(a) Plot the residual against the fitted values for your regression.

```
url <- getURL("http://www.statsci.org/data/oz/physical.txt")
dat <- read.table(text = url, header = TRUE)
model1 <- lm(Mass ~., data = dat)
plot(fitted(model1), resid(model1))
abline(h = 0)
```

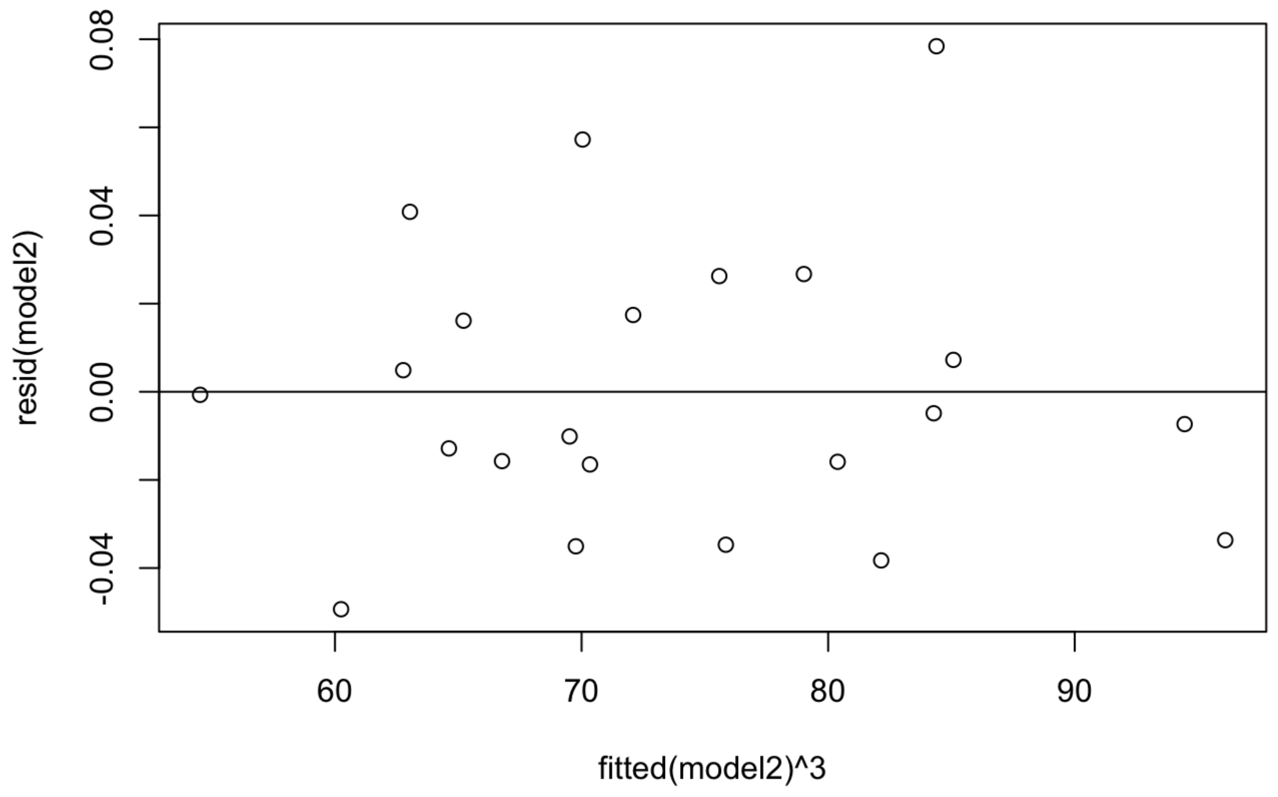


(b) Now regress the cube root of mass against these diameters. Plot theresidual against the fitted values in both these cube root coordinates and in the original coordinates.

```
# (b)Now regress the cube root of mass against these diameters.
model2 <- lm(Mass^(1/3) ~., data = dat)

# Plot the residual against the fitted values (cube root coordinates)
plot(fitted(model2), resid(model2))
abline(h = 0)
```



```
# Plot the residual against the fitted values (original coordinates)
plot(fitted(model2)^3, resid(model2))
abline(h = 0)
```



(c) Use your plots to explain which regression is better.

- In these plots, each point is one male subject, where the prediction made by the model is on the x-axis, and the accuracy of the prediction is on the y-axis. The distance from the line at 0 is how bad the prediction was for that value.0 values for the residual means the predict was exactly correct. Ideally, the plot of the residuals should satisfy the following requirements.

1. they're symmetrically distributed.
2. they're clustered around zero of the y-axis
3. in general, there aren't clear patterns. There are no big differences among these three plots. These are symmetrically distributed and no clear patterns. However, the residuals of regression which dependent value is the cube root of mass look slightly larger. It means the model which dependent value is the mass(model1) is better.

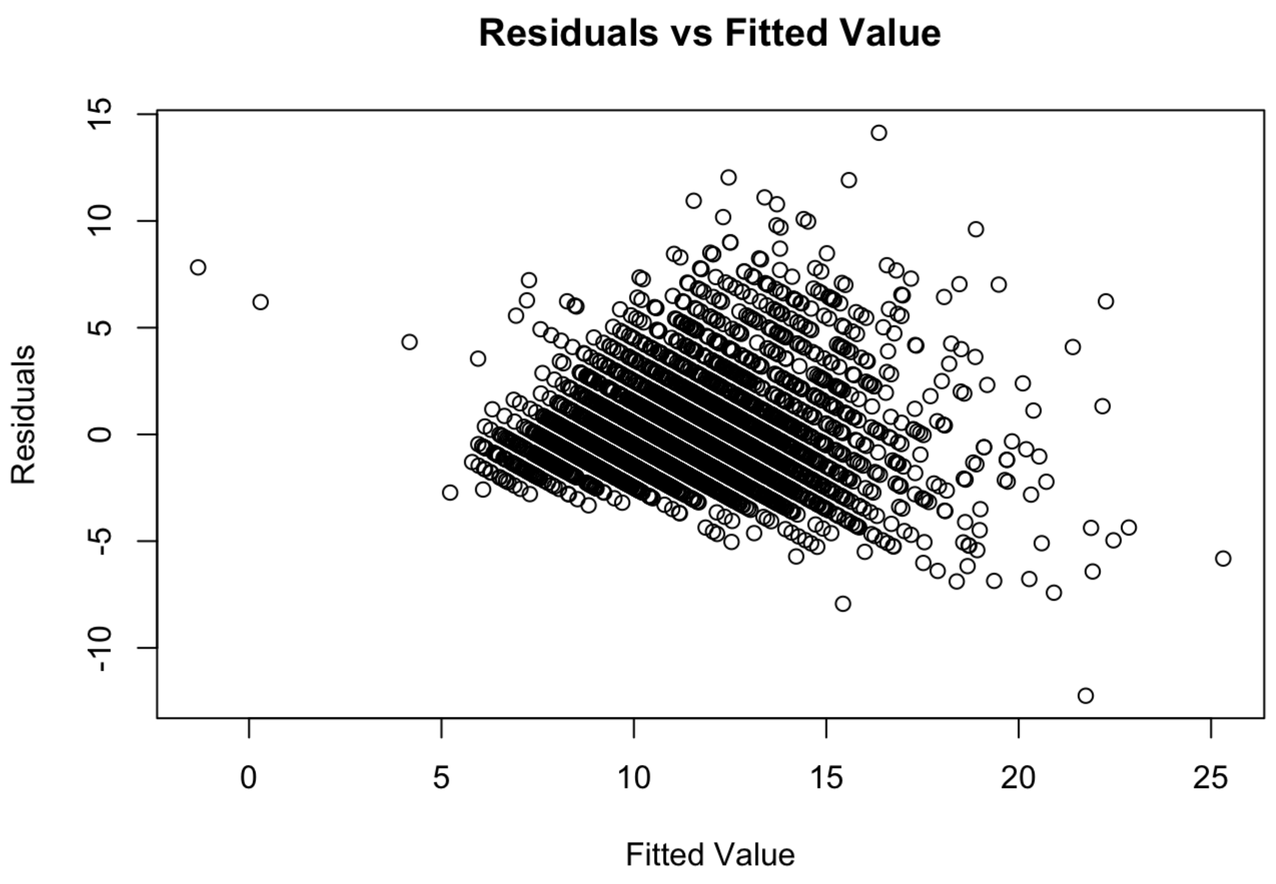
7.11

Data Source: https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/ (https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/)

```
df <- read.csv("abalone.data",col.names =c("Sex","Length","Diameter","Height","WholeWeig
ht","ShuckedWeight","VisceraWeight","ShellWeight","Rings"), stringsAsFactors = FALSE)
#based on data documetation, we can obtian age of Abalone in years by adding 1.5 to the
 ring column.
df$Age <- df$Rings+1.5
```

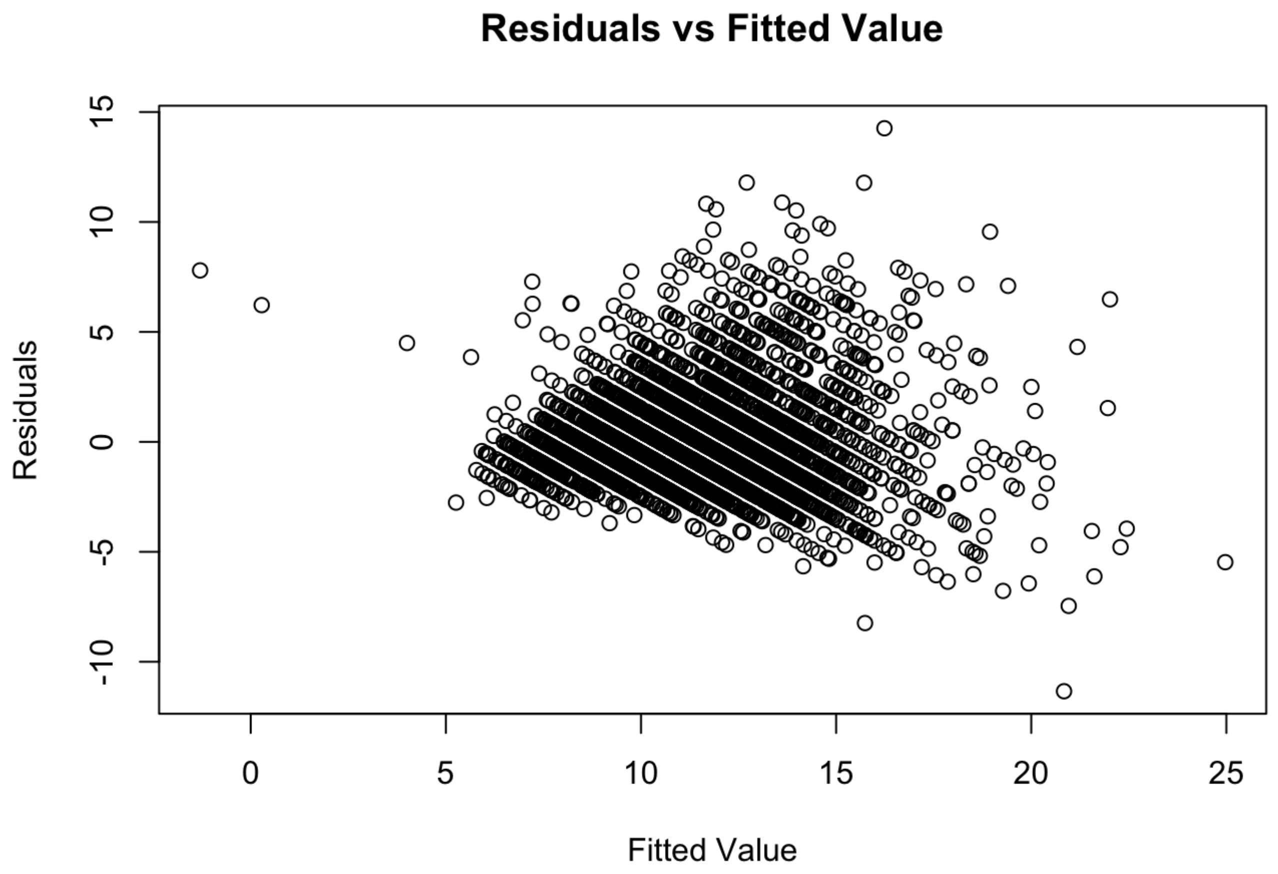
(a) Build a linear regression predicting the age from the measurements, ignoring gender. Plot the residual against the fitted values.

```
# we will be removing Sex, age and Rings from the features (Rings is like a age)
model.fit <- lm(formula = Age~Length+Length+Height+WholeWeight+ShuckedWeight+VisceraWeig
ht+ShellWeight, data = df)
plot(x = model.fit$fitted.values, y =model.fit$residuals,type = "p" ,main = "Residuals v
s Fitted Value",xlab = "Fitted Value", ylab = "Residuals")
```



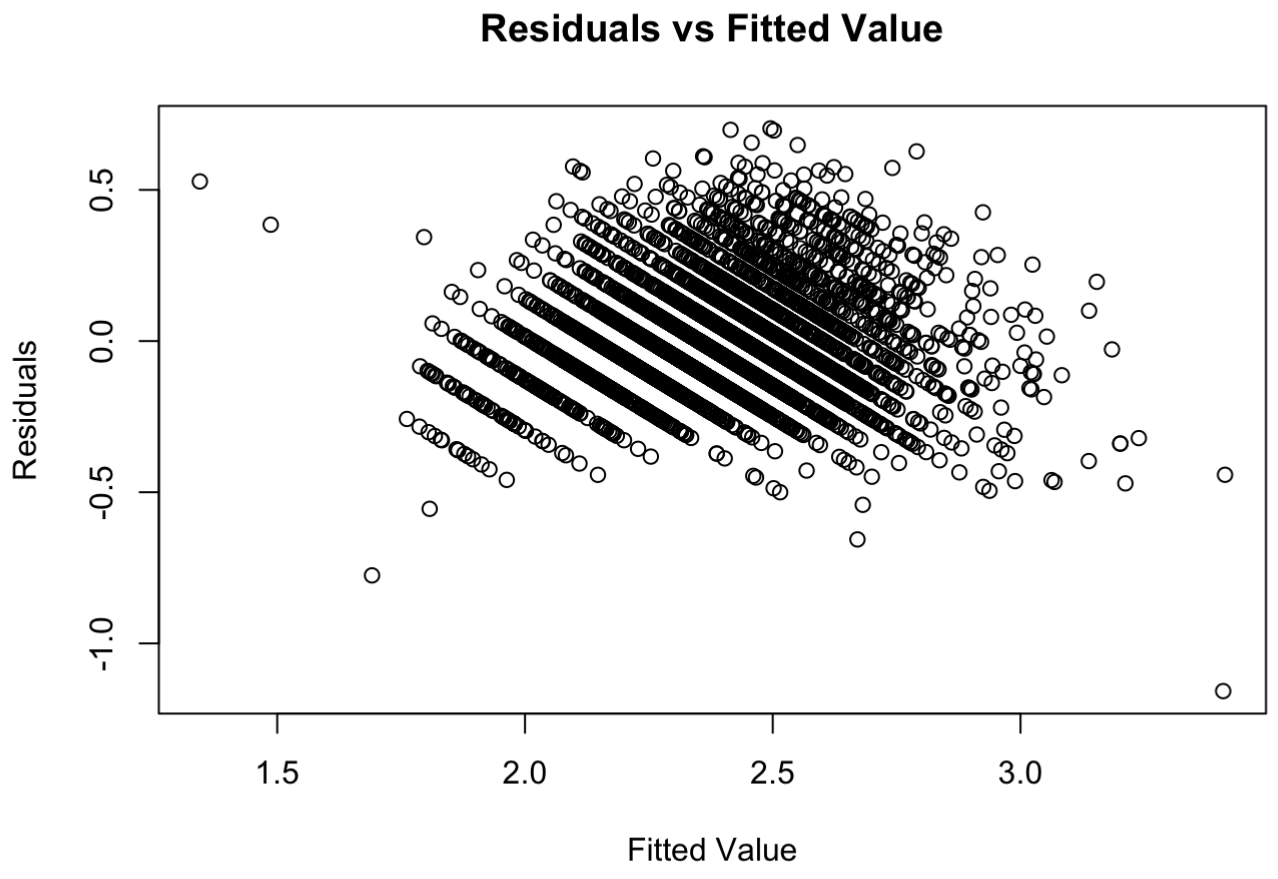
(b) Build a linear regression predicting the age from the measurements, including gender. There are three levels for gender; I'm not sure whetherthis has to do with abalone biology or difficulty in determining gender.You can represent gender numerically by choosing 1 for one level, 0 foranother, and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

```
df$Sex[df$Sex == "F"] =-1
df$Sex[df$Sex == "I"] =0
df$Sex[df$Sex == "M"] =1
model.fit <- lm(formula = Age~Length+Length+Height+WholeWeight+ShuckedWeight+VisceraWeig
ht+ShellWeight+Sex, data = df)
plot(x = model.fit$fitted.values, y =model.fit$residuals,type = "p" ,main = "Residuals v
s Fitted Value",xlab = "Fitted Value", ylab = "Residuals")
```



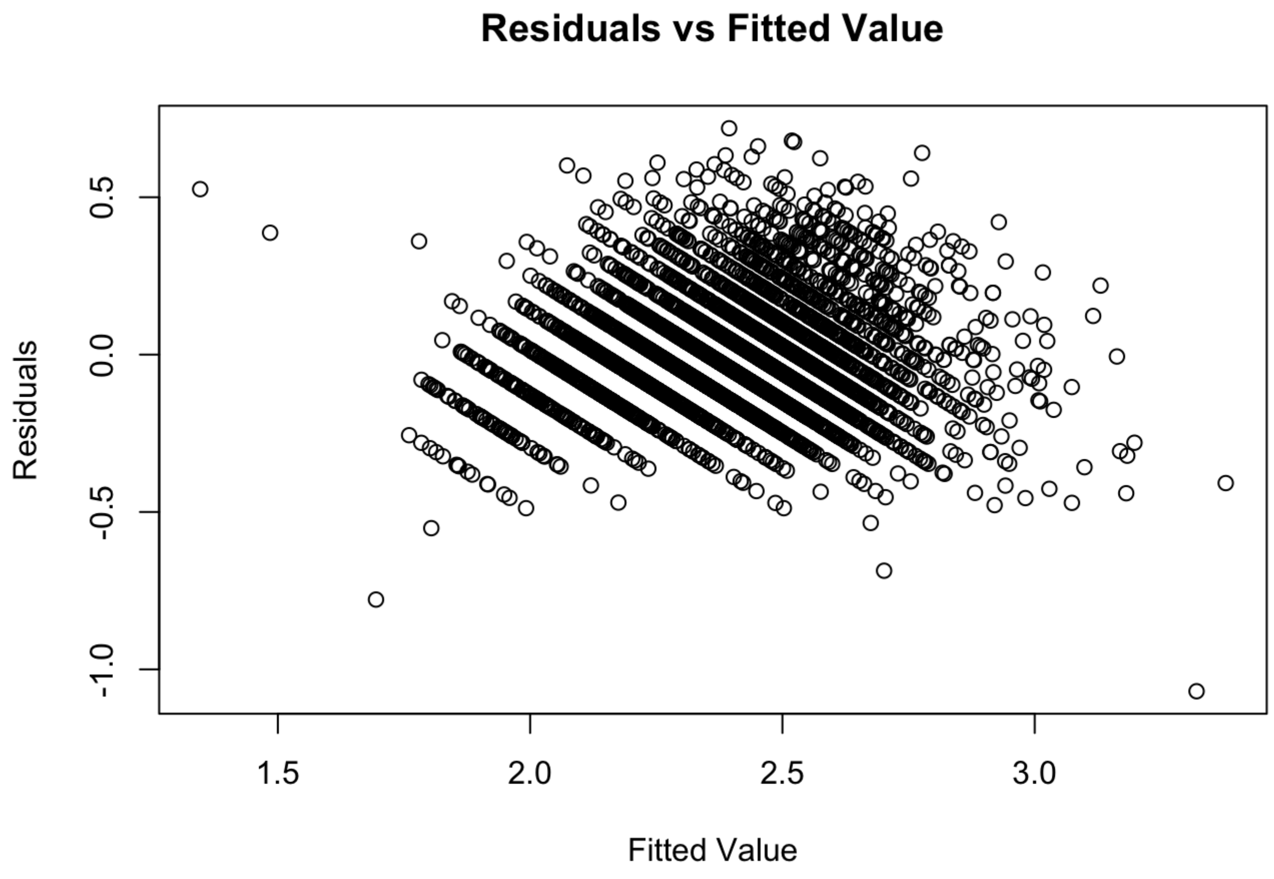
(c) Now build a linear regression predicting the log of age from the measurements, ignoring gender. Plot the residual against the fitted values.

```
dfLog <- df
dfLog$Age <- log(df$Age)

model.fit <- lm(formula = Age~Length+Length+Height+WholeWeight+ShuckedWeight+VisceraWeig
ht+ShellWeight, data = dfLog)

plot(x = model.fit$fitted.values, y =model.fit$residuals,type = "p" ,main = "Residuals v
s Fitted Value",xlab = "Fitted Value", ylab = "Residuals")
```



(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.fit <- lm(formula = Age~Length+Length+Height+WholeWeight+ShuckedWeight+VisceraWeig
ht+ShellWeight+Sex, data = dfLog)

plot(x = model.fit$fitted.values, y =model.fit$residuals,type = "p" ,main = "Residuals v
s Fitted Value",xlab = "Fitted Value", ylab = "Residuals")
```

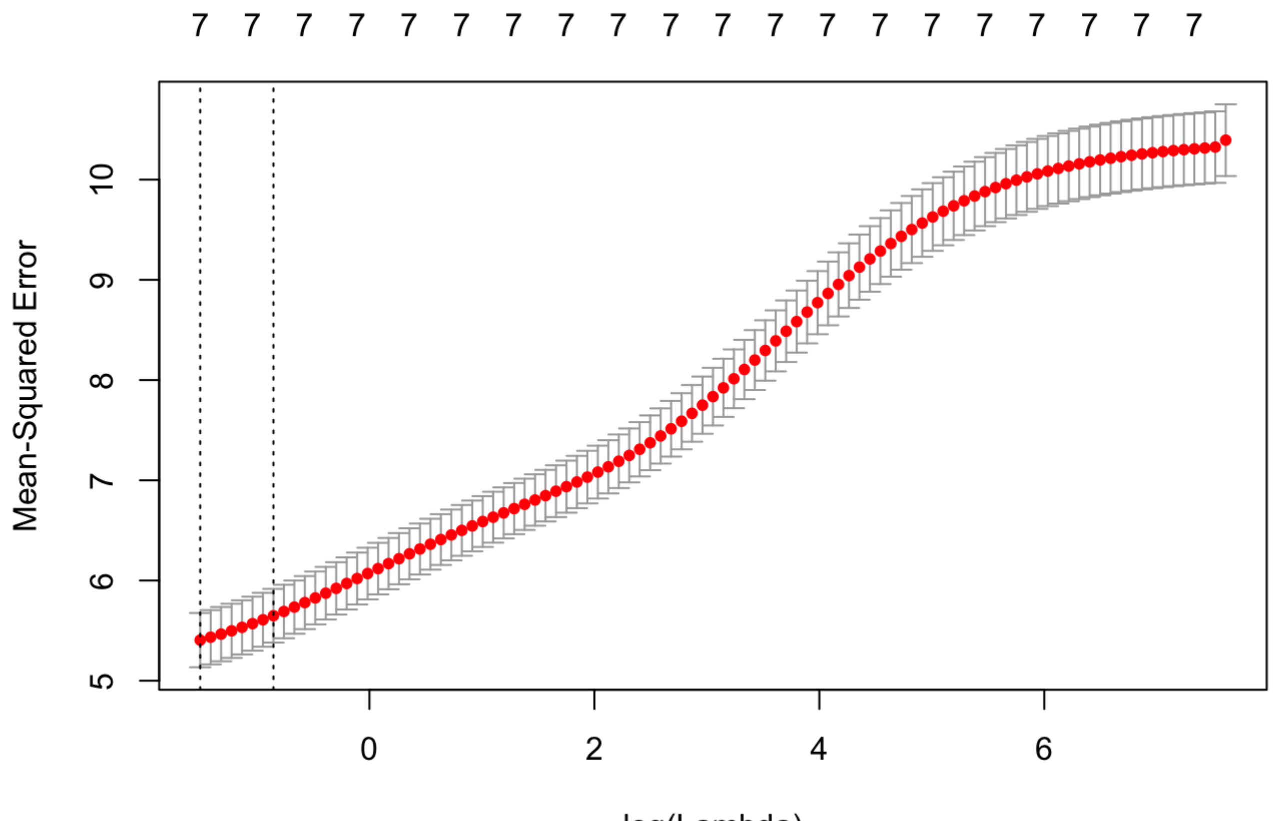


(e) It turns out that determining the age of an abalone is possible, but difficult(you section the shell, and count rings). Use your plots to explain which regression you would use to replace this procedure, and why.

- First of all, its hard to find difference in any improvement by using gender so we can remove model b and d from our analysis. Secondly, plot of part (c) follows linearity assumption better than plot of part a. Therefore, we can select model of part (c) as the best model

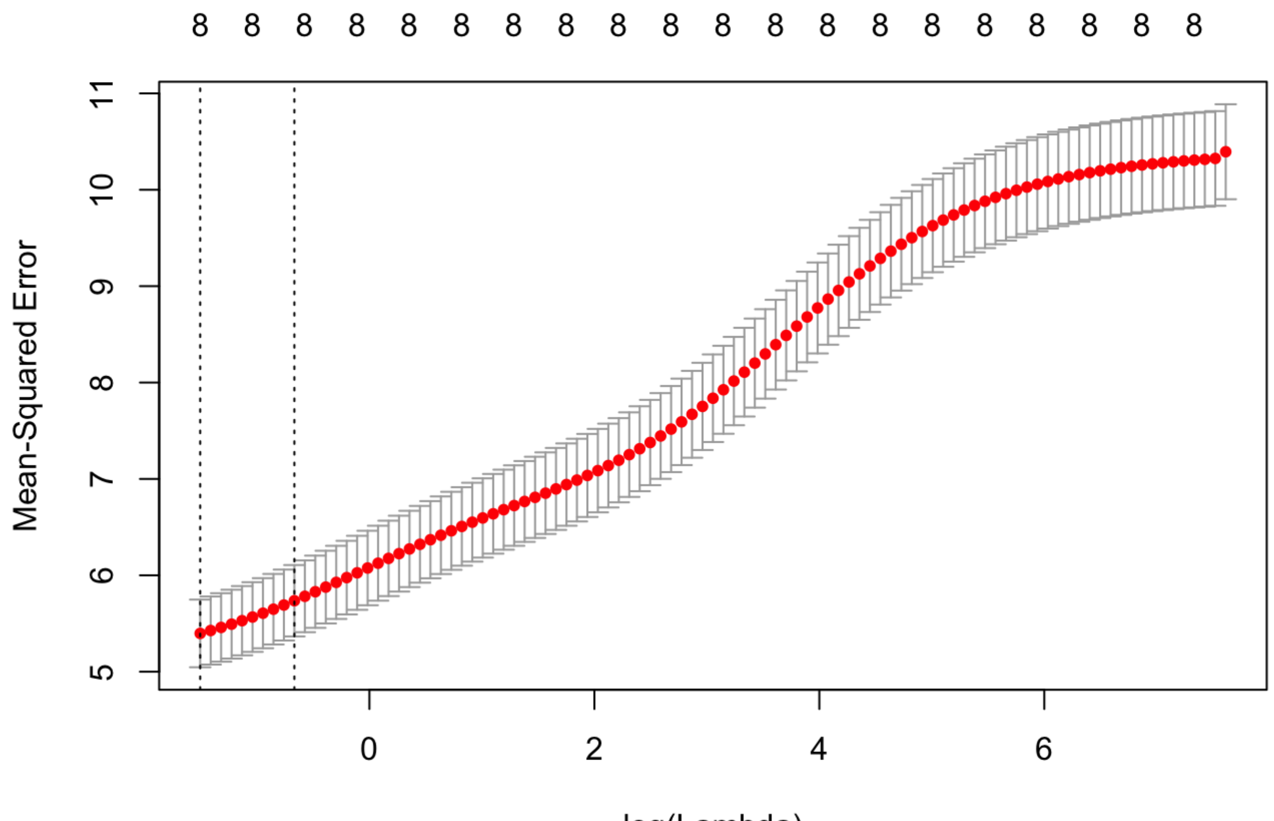
(f) Can you improve these regressions by using a regularizer? Use glmnet toobtain plots of the cross-validated prediction error.

```
features <- subset(names(df),!(names(df) %in% c("Sex","Rings","Age")))
outcome <- c("Age")
#7.11a
model.fit <- cv.glmnet(x= data.matrix(df[,features]), y=data.matrix(df[,outcome]),alpha
= 0)
plot(model.fit, sub = "7.11a using regularizer")
```



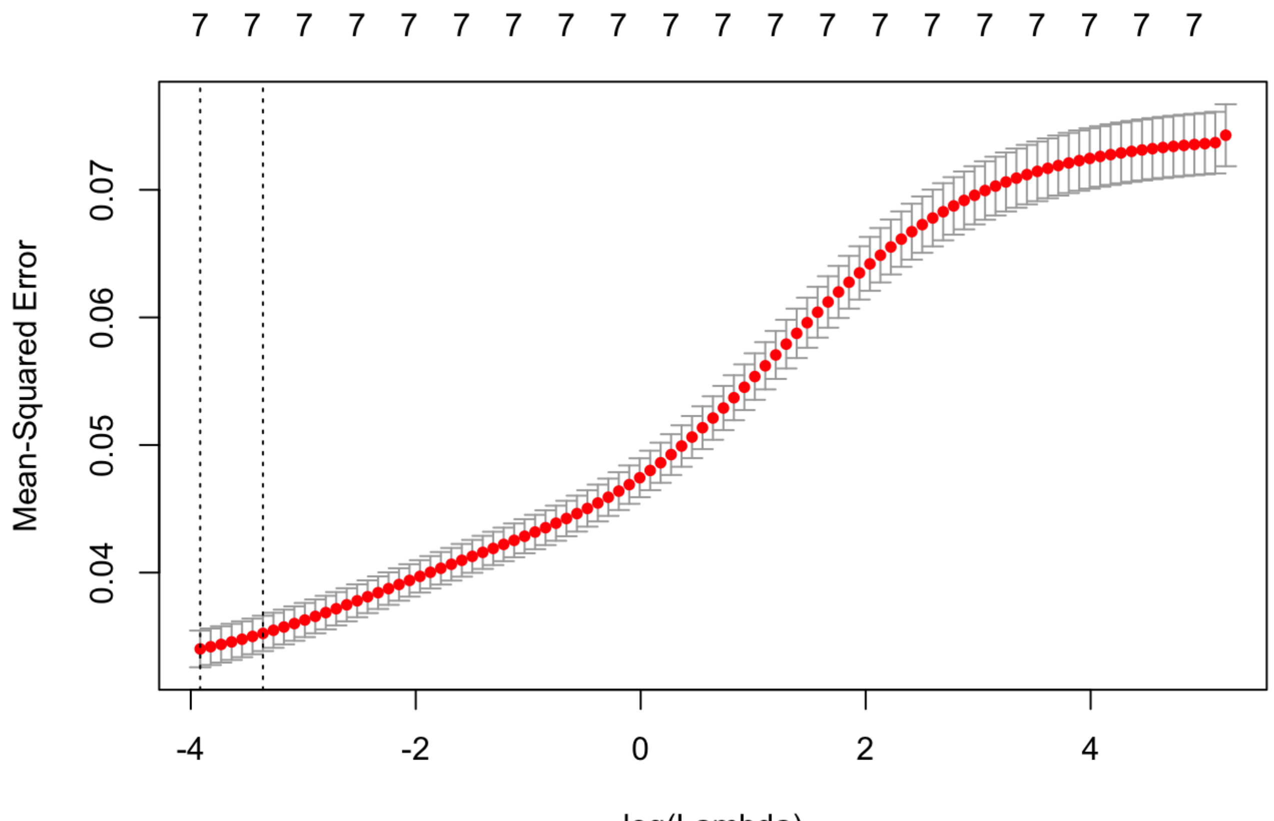
7.11a using regularizer

```
#7.11b
#including sex
features <- subset(names(df),!(names(df) %in% c("Rings","Age")))
model.fit <- cv.glmnet(x= data.matrix(df[,features]), y=data.matrix(df[,outcome]),alpha
= 0)
plot(model.fit, sub = "7.11b using regularizer")
```



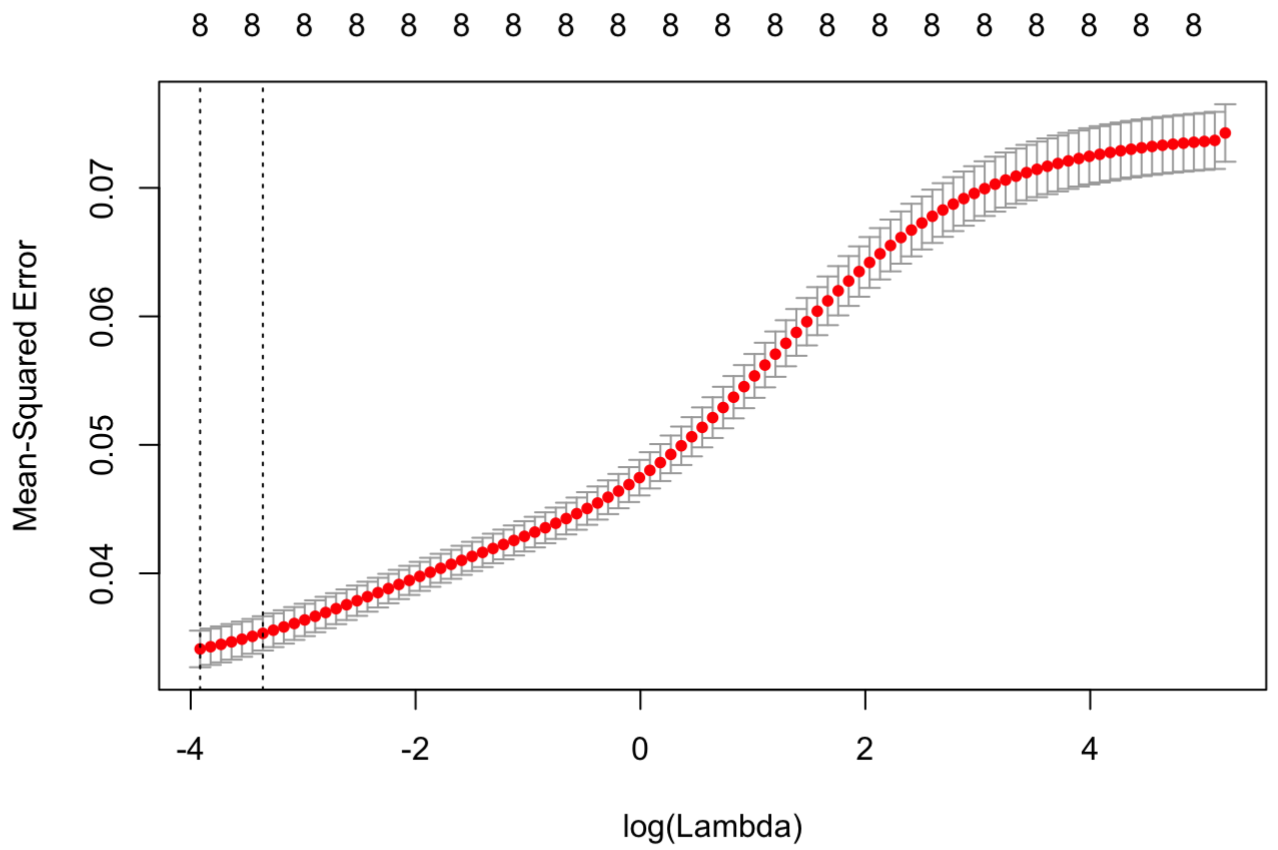
7.11b using regularizer

```
#7.11c
# now log the age and ignore gender
features <- subset(names(df),!(names(df) %in% c("Sex","Rings","Age")))
model.fit <- cv.glmnet(x= data.matrix(dfLog[,features]), y=data.matrix(dfLog[,outcome]),
alpha = 0)
plot(model.fit, sub = "7.11c using regularizer")
```



7.11c using regularizer

```
#7.11d
#now log the age and include gender
features <- subset(names(df),!(names(df) %in% c("Rings","Age")))
model.fit <- cv.glmnet(x= data.matrix(dfLog[,features]), y=data.matrix(dfLog[,outcome]),
alpha = 0)
plot(model.fit, sub = "7.11d using regularizer")
```



7.11d using regularizer

- From the above analysis of using regularizer, we found that both models can be improved using regularizer