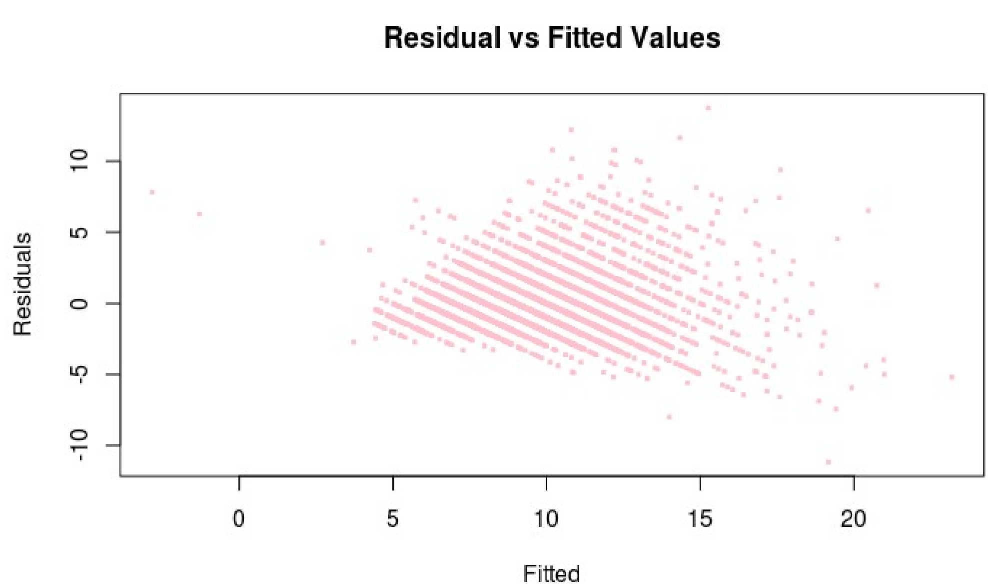
a. Build a linear regression predicting the age from the measurements, ig- noring gender. Plot the residual against the fitted values.

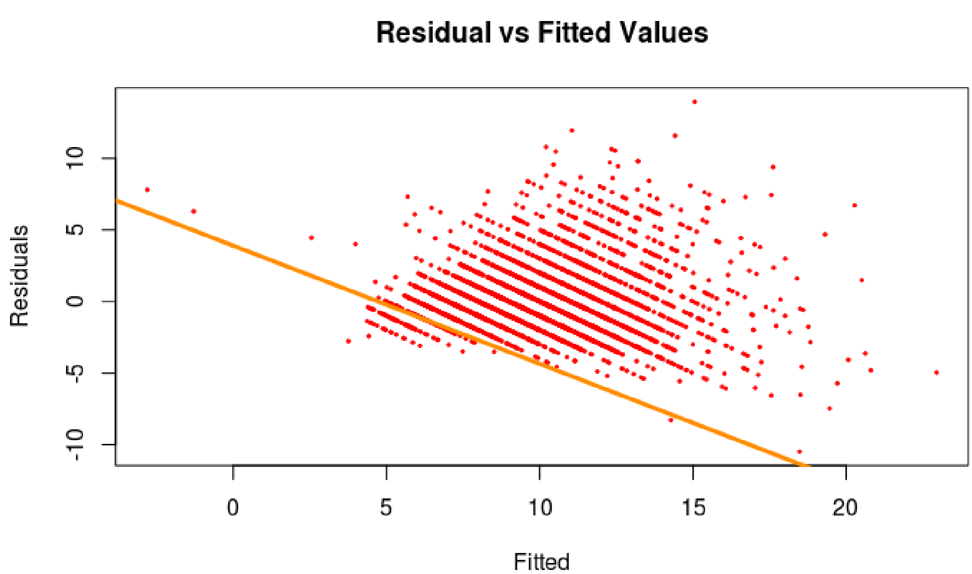
In R, the lm(), or “linear model,” function can be used to create a simple regression model Since we are trying to predict age based on measurements (while ignoring gender), it is important that we only consider columns B to H (which correspond to measurments length, diameter, height, and multiple weights) as well as column I which gives us the age. We can assume here than a bigger, meatier plant will be older than a smaller one (however, that is for our graph to portray). Our predictors would be columns B to H.

```
> size_to_age_model = lm(abalone_data$V9 ~ abalone_data$V2 + abalone_data$V3 + abalone_data$V4 +
abalone_data$V5 + abalone_data$V6 + abalone_data$V7 + abalone_data$V8, data = abalone_data)
> summary(size_to_age_model) #Let's just check out the summary
Call:
lm(formula = abalone_data$V9 ~ abalone_data$V2 + abalone_data$V3 +
     abalone_data$V4 + abalone_data$V5 + abalone_data$V6 + abalone_data$V7 +
     abalone_data$V8, data = abalone_data)

Residuals:
     Min       1Q   Median       3Q      Max
-11.1632  -1.3613  -0.3885   0.9054  13.7440

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          2.9852     0.2691  11.092  < 2e-16 ***
abalone_data$V2     -1.5719     1.8248  -0.861    0.389
abalone_data$V3     13.3609     2.2371   5.972 2.53e-09 ***
abalone_data$V4     11.8261     1.5481   7.639 2.70e-14 ***
abalone_data$V5      9.2474     0.7326  12.622  < 2e-16 ***
abalone_data$V6    -20.2139     0.8233 -24.552  < 2e-16 ***
abalone_data$V7     -9.8297     1.3040  -7.538 5.82e-14 ***
abalone_data$V8      8.5762     1.1367   7.545 5.54e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.218 on 4169 degrees of freedom
Multiple R-squared:  0.5276,    Adjusted R-squared:  0.5268
F-statistic: 665.2 on 7 and 4169 DF,  p-value: < 2.2e-16
> abalone.res = resid(size_to_age_model) #Let us get the residuals
> abalone.predict = predict(size_to_age_model, data.frame(abalone_data[c(1:7)]))
> plot(abalone.predict, abalone.res,pch = 14, cex = .3, col = "pink",xlab="Fitted",
ylab="Residuals", main="Residual vs Fitted Values") #Plot
```



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.

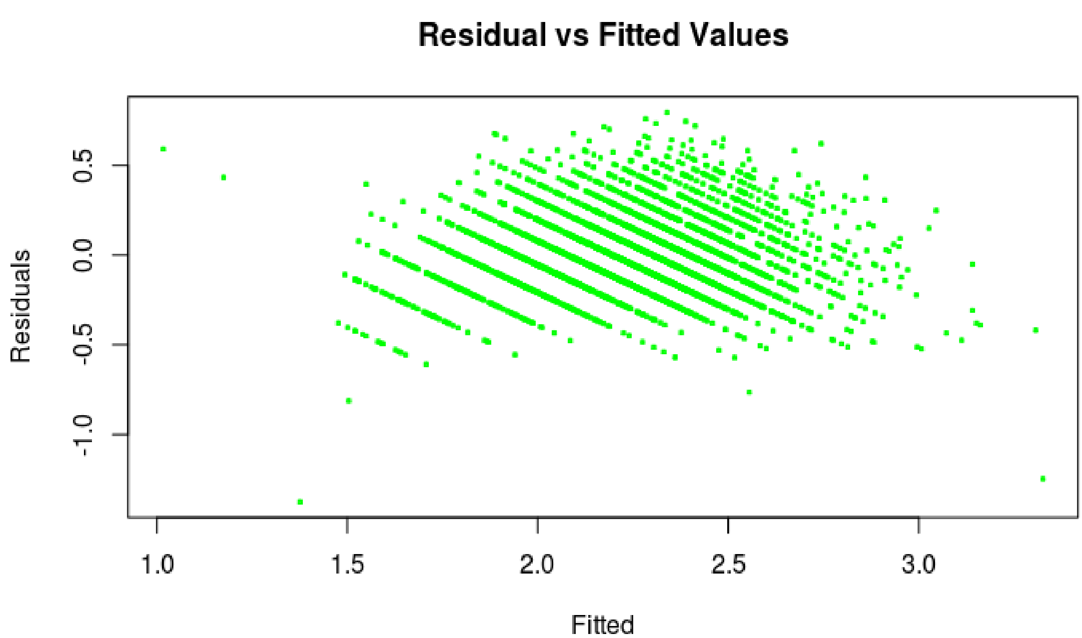
```
> abalone_data <- read.csv("abalone.data", header = FALSE)
> abalone_data$V1 <- factor(abalone_data$V1) ##Using factor to create the categorical gender as numeric
> size_to_age_model_gen = lm(abalone_data$V9 ~ abalone_data$V1 + abalone_data$V2 + abalone_data$V3 + abalone_data$V4 + abalone_data$V5 + abalone_data$V6 + abalone_data$V7 + abalone_data$V8, data = abalone_data)
> summary(size_to_age_model_gen) #Let's just check out the summary
Call:
lm(formula = abalone_data$V9 ~ abalone_data$V1 + abalone_data$V2 +
    abalone_data$V3 + abalone_data$V4 + abalone_data$V5 + abalone_data$V6 +
    abalone_data$V7 + abalone_data$V8, data = abalone_data)

Residuals:
     Min       1Q   Median       3Q      Max
-10.4800  -1.3053  -0.3428   0.8600  13.9426

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         3.89464    0.29157  13.358  < 2e-16 ***
abalone_data$V1I   -0.82488    0.10240  -8.056 1.02e-15 ***
abalone_data$V1M    0.05772    0.08335   0.692    0.489
abalone_data$V2    -0.45834    1.80912  -0.253    0.800
abalone_data$V3    11.07510    2.22728   4.972 6.88e-07 ***
abalone_data$V4    10.76154    1.53620   7.005 2.86e-12 ***
abalone_data$V5     8.97544    0.72540  12.373  < 2e-16 ***
abalone_data$V6   -19.78687    0.81735 -24.209  < 2e-16 ***
abalone_data$V7   -10.58183    1.29375  -8.179 3.76e-16 ***
abalone_data$V8     8.74181    1.12473   7.772 9.64e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.194 on 4167 degrees of freedom
Multiple R-squared:  0.5379,    Adjusted R-squared:  0.5369
F-statistic: 538.9 on 9 and 4167 DF,  p-value: < 2.2e-16
> abalone.res = resid(size_to_age_model_gen) #Resids
> abalone.predict = predict(size_to_age_model_gen, data.frame(abalone_data[c(0:7)])) #Include the first (index[0])
> plot(abalone.predict, abalone.res,pch = 10, cex = .3, col = "red",xlab="Fitted", ylab="Residuals", main="Residual vs Fitted Values")
> abline(size_to_age_model_gen, lwd = 3, col = "darkorange") #Let us check out the fitted line to the scatterplot. It looks good and is consistent to the trend!
```



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

We can use part a again as we are ignoring gender.

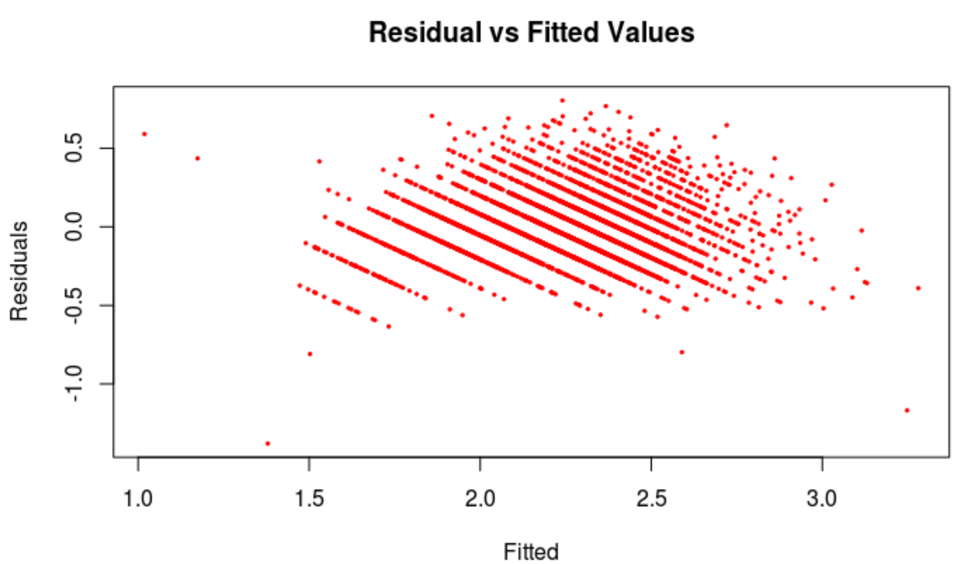
```
> abalone_data <- read.csv("abalone.data", header = FALSE)
> size_to_age_model_log = lm(log(abalone_data$V9) ~ abalone_data$V2 + abalone_data$V3 +
abalone_data$V4 + abalone_data$V5 + abalone_data$V6 + abalone_data$V7 + abalone_data$V8, data =
abalone_data) #Log of Age
> summary(size_to_age_model_log) #Let's just check out the summary
Call:
lm(formula = log(abalone_data$V9) ~ abalone_data$V2 + abalone_data$V3 +
     abalone_data$V4 + abalone_data$V5 + abalone_data$V6 + abalone_data$V7 +
     abalone_data$V8, data = abalone_data)

Residuals:
     Min       1Q  Median       3Q      Max
-1.3759 -0.1373 -0.0223  0.1121  0.7962

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      1.23950    0.02498  49.611  < 2e-16 ***
abalone_data$V2  0.40669    0.16940   2.401   0.0164 *
abalone_data$V3  1.68204    0.20768   8.099 7.20e-16 ***
abalone_data$V4  1.32680    0.14372   9.232  < 2e-16 ***
abalone_data$V5  0.63908    0.06802   9.396  < 2e-16 ***
abalone_data$V6 -1.70429    0.07643 -22.298  < 2e-16 ***
abalone_data$V7 -0.75136    0.12106  -6.207 5.94e-10 ***
abalone_data$V8  0.58793    0.10553   5.571 2.69e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2059 on 4169 degrees of freedom
Multiple R-squared:  0.5855,    Adjusted R-squared:  0.5848
F-statistic: 841.2 on 7 and 4169 DF,  p-value: < 2.2e-16
> abalone.res = resid(size_to_age_model_log) #Let us get the residuals
> abalone.predict = predict(size_to_age_model_log, data.frame(abalone_data[c(1:7)]))
> plot(abalone.predict, abalone.res,pch = 14, cex = .3, col = "green",xlab="Fitted",
ylab="Residuals", main="Residual vs Fitted Values") #Plot
```



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.

We can use part b because we are looking at gender

```
> abalone_data <- read.csv("abalone.data", header = FALSE)
> abalone_data$V1 <- factor(abalone_data$V1) ##Using factor to create the categorical gender as numeric
> size_to_age_model_genl = lm(log(abalone_data$V9) ~ abalone_data$V1 + abalone_data$V2 + abalone_data$V3 + abalone_data$V4 + abalone_data$V5 + abalone_data$V6 + abalone_data$V7 + abalone_data$V8, data = abalone_data)
> summary(size_to_age_model_genl) #Let's just check out the summary
Call:
lm(formula = log(abalone_data$V9) ~ abalone_data$V1 + abalone_data$V2 +
     abalone_data$V3 + abalone_data$V4 + abalone_data$V5 + abalone_data$V6 +
     abalone_data$V7 + abalone_data$V8, data = abalone_data)

Residuals:
     Min       1Q   Median       3Q      Max
-1.37909 -0.13172 -0.01587  0.11120  0.80427

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      1.341185   0.026914  49.832  < 2e-16 ***
abalone_data$V1I -0.092485   0.009452  -9.785  < 2e-16 ***
abalone_data$V1M  0.008926   0.007694   1.160  0.24605
abalone_data$V2   0.533049   0.166998   3.192  0.00142 **
abalone_data$V3   1.423575   0.205598   6.924 5.06e-12 ***
abalone_data$V4   1.206625   0.141805   8.509  < 2e-16 ***
abalone_data$V5   0.608252   0.066961   9.084  < 2e-16 ***
abalone_data$V6  -1.657046   0.075449 -21.963  < 2e-16 ***
abalone_data$V7  -0.835499   0.119425  -6.996 3.05e-12 ***
abalone_data$V8   0.606814   0.103823   5.845 5.46e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2025 on 4167 degrees of freedom
Multiple R-squared:  0.5991,    Adjusted R-squared:  0.5982
F-statistic: 691.8 on 9 and 4167 DF,  p-value: < 2.2e-16
> abalone.res = resid(size_to_age_model_genl) #Resids
> abalone.predict = predict(size_to_age_model_genl, data.frame(abalone_data[c(0:7)])) #Include the first (index[0])
> plot(abalone.predict, abalone.res,pch = 10, cex = .3, col = "red",xlab="Fitted", ylab="Residuals", main="Residual vs Fitted Values")
```



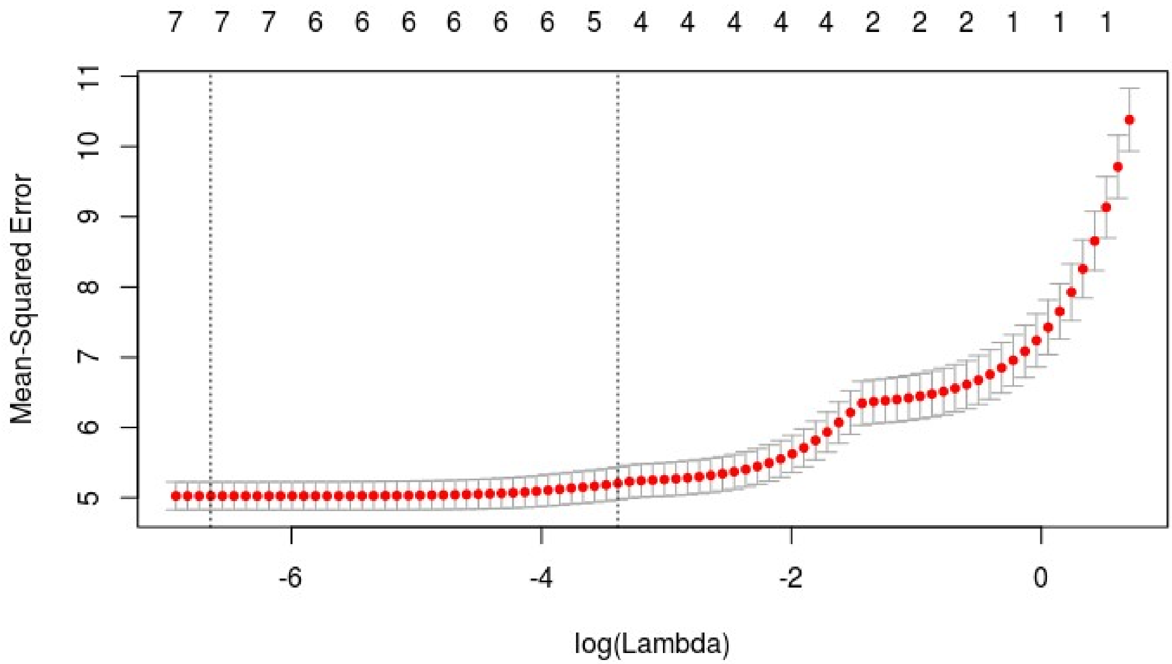
e. It turns out that determining theage of anabalone is possible,but difficult (you section the shell, and count rings). Use your plots to explain which regression you would use to replace this procedure, and why.

R-squared is a statistical measure of how close the data are to the fitted regression line. Generally, the higher the R-squared....the better the model fits the data! If you see above, I used summary(lm) to display the R^2 values. We will choose the graph with the largest R^2 value to explain which regression we would use to give us the best result. In this scenario we would use plot 4 (look below). This would intuitively also make the most sense as well.

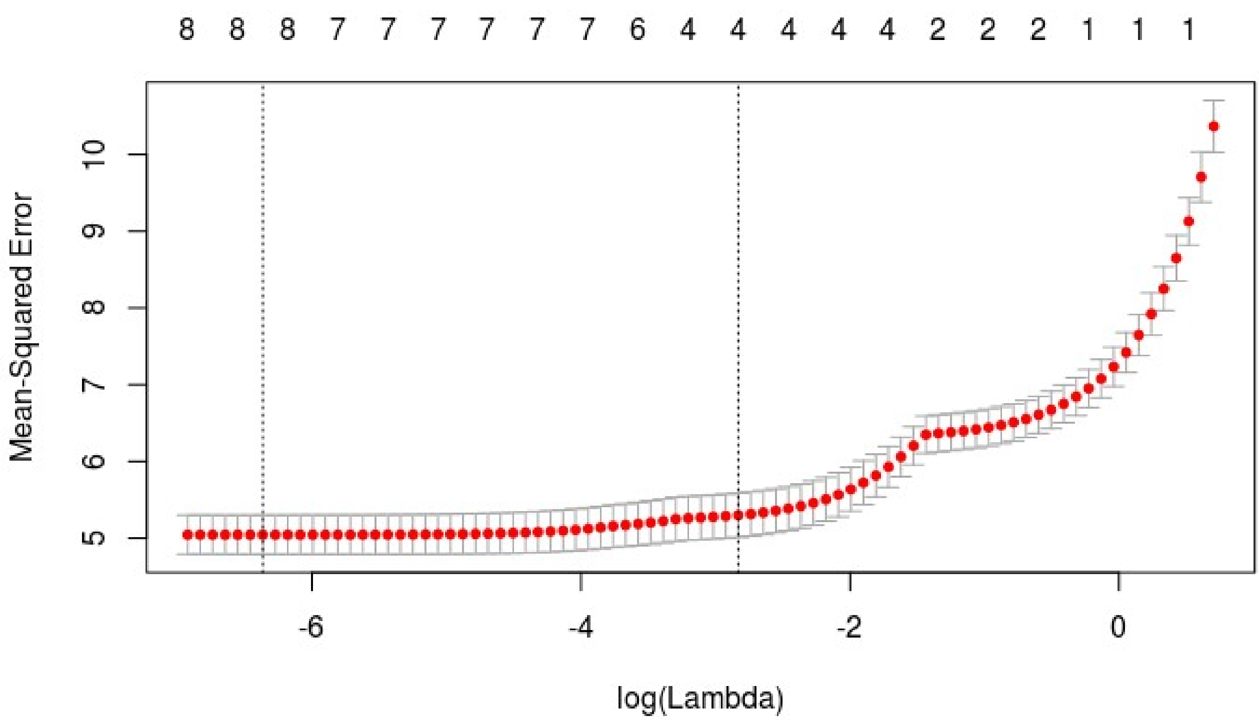
Graph 1 R^2: 0.5268
Graph 2 R^2: 0.5369
Graph 3 R^2: 0.5848
Graph 4 R^2: 0.5982

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

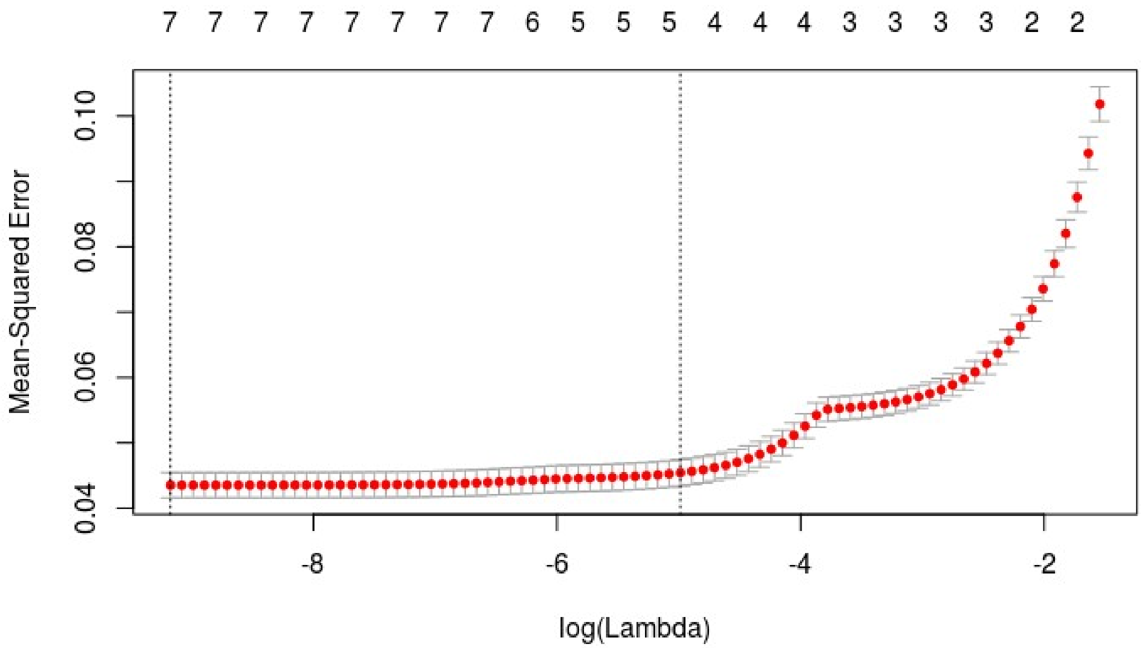
```
> install.packages("glmnet")
> install.packages("plyr")
> library("glmnet")
> library("plyr")

> abalone_data <- read.csv("abalone.data", header = FALSE)

Get x and y for Part (a) dataset
> ax = as.matrix(abalone_data[2:8])
> ay = as.matrix(abalone_data[9])

Get x and y for Part (b) dataset by making gender field numeric
> b_data  = abalone_data
> b_data$num <- mapvalues(b_data$V1, from = c("M", "F", "I"), to = c(1, -1, 0))
> b_data[1] = as.numeric(b_data$num)
> b_data$num <- NULL

> bx = as.matrix(b_data[1:8])
> by = as.matrix(b_data[9])

Get x and y for Part (c) dataset
> cx = as.matrix(abalone_data[2:8])
> cy = as.matrix(log(abalone_data[9]))

Get x and y for Part (d) dataset
> dx = as.matrix(b_data[1:8])
> dy = as.matrix(log(b_data[9]))

Cross validation and plots for each part

> p1 = cv.glmnet(ax,ay)
> plot(p1)
> print(p1$lambda.min)
[1] 0.001300504
```




```
> p2 = cv.glmnet(bx,by)
> plot(p2)
> print(p2$lambda.min)
[1] 0.001719189
```



```
> p3 = cv.glmnet(cx,cy)
> plot(p3)
> print(p3$lambda.min)
[1] 0.000103824
```




```
> plot(p4)
> print(p4$lambda.min)
[1] 0.000103824
```