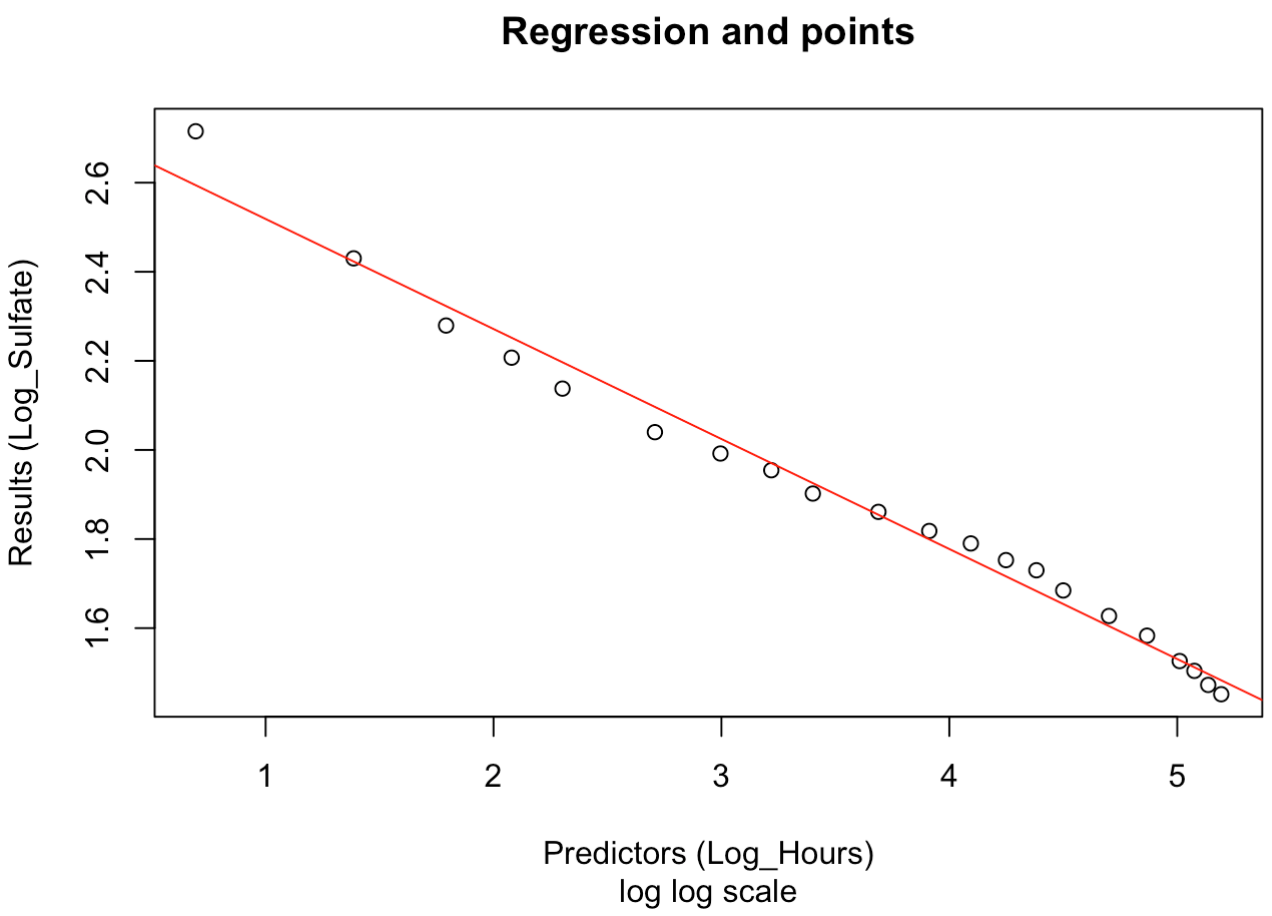
CS 498 AML Homework 4

Question 7.9

```
# preparing data
brunhild <- read_csv("./q1.csv")
```

(a)

```
# Making log-log transformed model
brunhild.lm_log = lm(log(Sulfate) ~ 1 + log(Hours), data=brunhild)
plot(log(brunhild$Hours),
     log(brunhild$Sulfate),
     main="Regression and points",
     sub="log log scale",
     xlab="Predictors (Log_Hours)",
     ylab="Results (Log_Sulfate)")
abline(brunhild.lm_log, col="red")
```

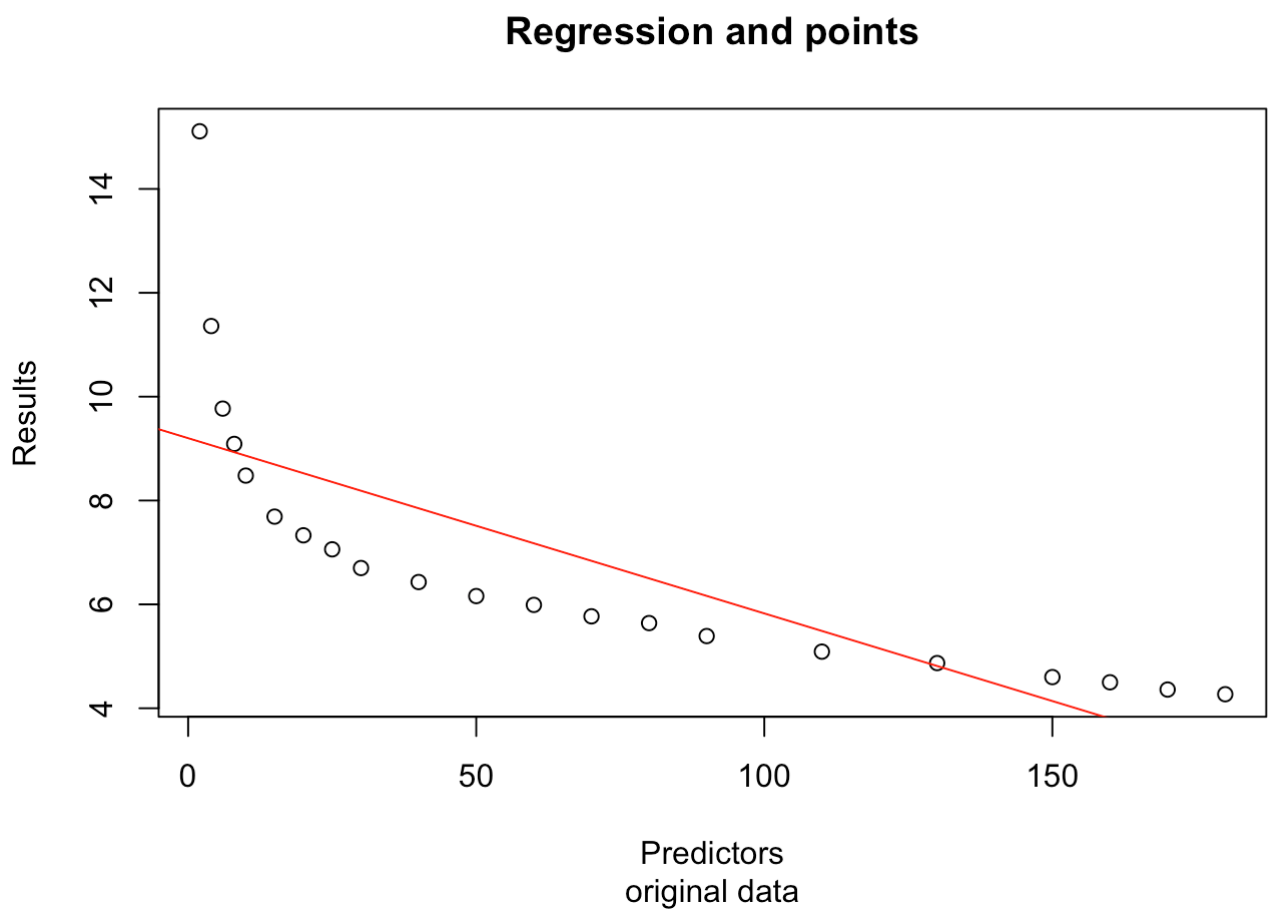



```
summary(brunhild.lm_log)
```

```
##
## Call:
## lm(formula = log(Sulfate) ~ 1 + log(Hours), data = brunhild)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05928 -0.03132 -0.00192  0.02268  0.12076
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.76584    0.02768    99.9   <2e-16 ***
## log(Hours)  -0.24705    0.00724   -34.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0435 on 19 degrees of freedom
## Multiple R-squared:  0.984,  Adjusted R-squared:  0.983
## F-statistic: 1.16e+03 on 1 and 19 DF,  p-value: <2e-16
```

(b)

```
# Making untransforned model
brunhild.lm = lm(Sulfate ~ 1 + Hours, data=brunhild)
plot(brunhild$Hours,
     brunhild$Sulfate,
     main="Regression and points",
     sub="original data",
     xlab="Predictors",
     ylab="Results")
abline(brunhild.lm, col="red")
```




```
summary(brunhild.lm)
```

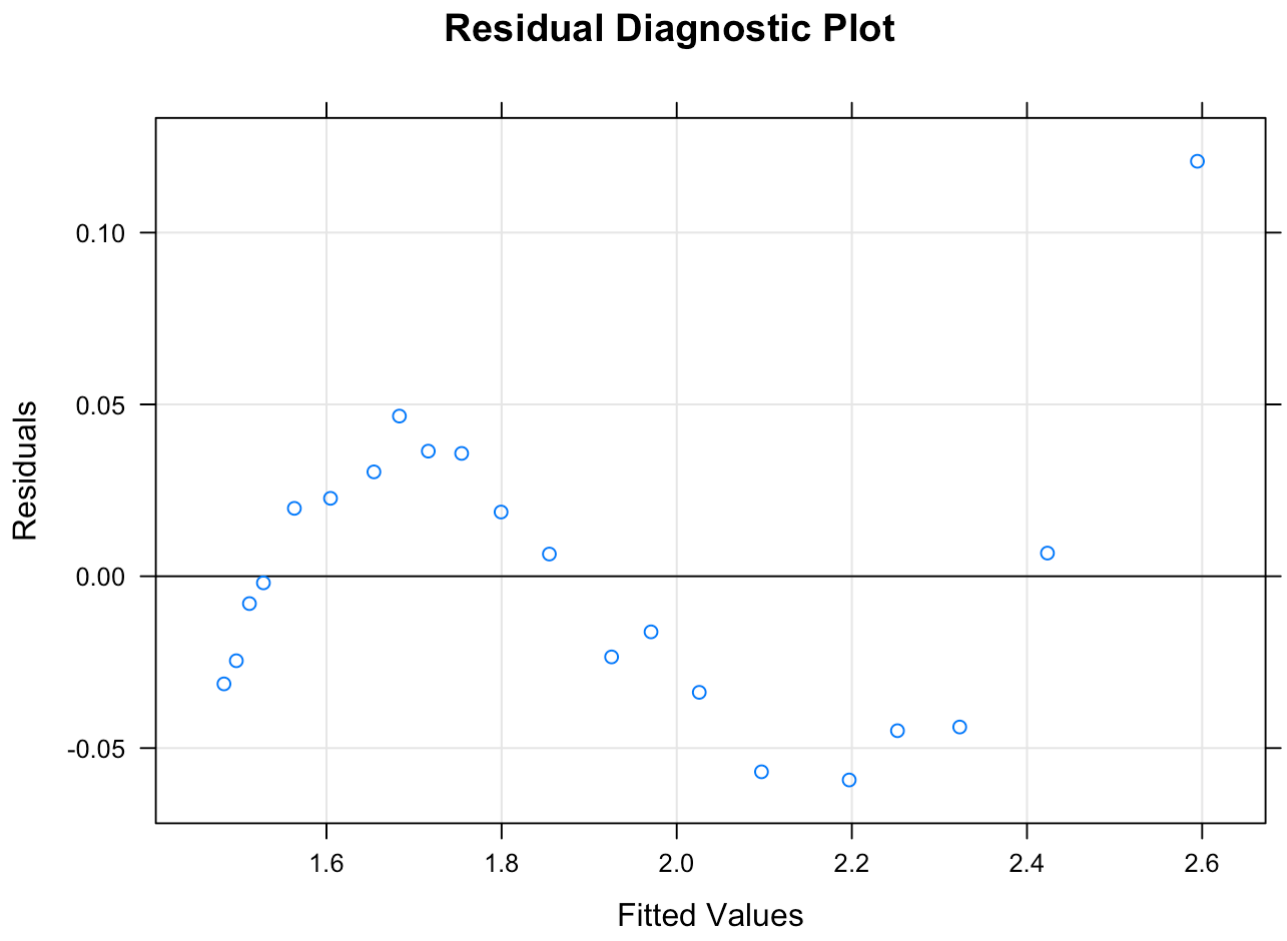
```
##
## Call:
## lm(formula = Sulfate ~ 1 + Hours, data = brunhild)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.490 -1.187 -0.399  0.699  5.975
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.2029     0.5814   15.83  2.1e-12 ***
## Hours        -0.0338     0.0065   -5.19  5.2e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.76 on 19 degrees of freedom
## Multiple R-squared:  0.587,  Adjusted R-squared:  0.565
## F-statistic:   27 on 1 and 19 DF,  p-value: 0.0000519
```

Rather than stopping here we perform some investigations using residual diagnostics to determine whether the various assumptions that underpin linear regression are reasonable for our data or if there is evidence to suggest that additional variables are required in the model or some other alterations to identify a better description of the variables that determine how weight changes.

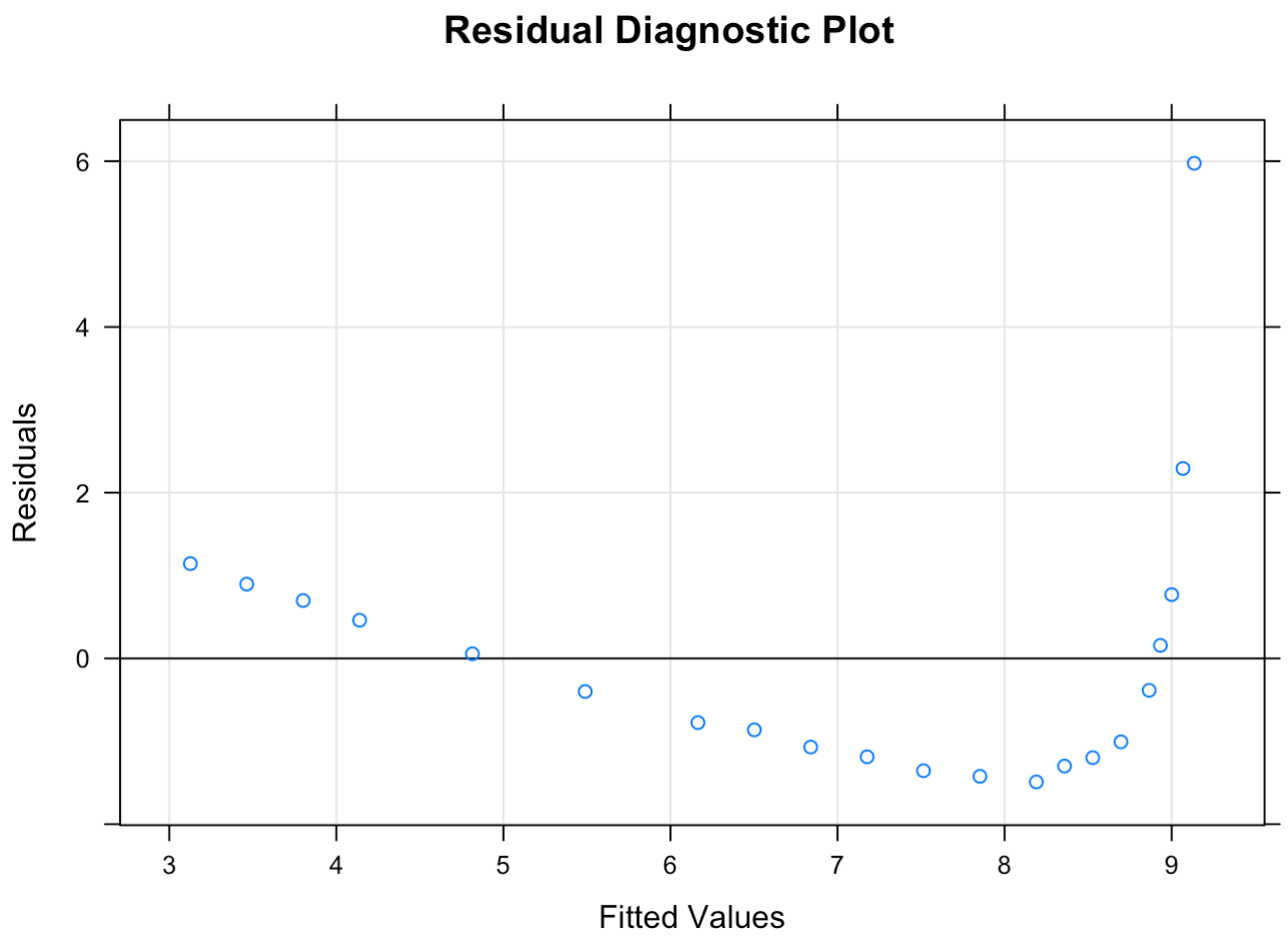
(c)

```
# Comparing residuals of the two models

# residual against fitted values in log-log coordinates
xyplot(resid(brunhild.lm_log) ~ fitted(brunhild.lm_log),
       xlab = "Fitted Values",
       ylab = "Residuals",
       main = "Residual Diagnostic Plot",
       panel = function(x, y, ...)
       {
         panel.grid(h = -1, v = -1)
         panel.abline(h = 0)
         panel.xyplot(x, y, ...)
       }
)
```




```
# residual against fitted values in original coordinates
xyplot(resid(brunhild.lm) ~ fitted(brunhild.lm),
       xlab = "Fitted Values",
       ylab = "Residuals",
       main = "Residual Diagnostic Plot",
       panel = function(x, y, ...)
       {
         panel.grid(h = -1, v = -1)
         panel.abline(h = 0)
         panel.xyplot(x, y, ...)
       }
)
```




```
#two windows will open, please look at both
```

(d)

It is clear that a non-linear regression technique was crucial here to allow a linear regression model to be computed given the "exponentiality" of the data. By doing the nonlinear transformation changes (log-log scale), we have effectively increased the linear relationships between variables and, thus, changed the correlation between variables.

The range of residuals is much smaller in the the log-log scale model which indicates that it is better. Also the residuals in the original model-data model clearly resemble a pattern. The residuals of the log-log look closer to a normal distribution. This also indicates that the log-log model is better.

It is worth mentioning that thanks to the previously observed "summary", we can clearly notice that our Rsquared has effectively increased from 0.565 to 0.983, suggesting that our log scale transformation has indeed been successful and presents itself as a better model.

A final, crucial observation is that athough our scale is now much much smaller for the residual diagnostic plot concerning the log transofrmation, we can still see a clear pattern, if this were to be taken in full consideration, we would have to point out that non-random patterns in a residual plot suggest a departure from linearity in the data being plotted. Hence, exploring some other non linear transformations can be considered although the current transformed model suggests itself as being sufficiently good.

Question 7.10

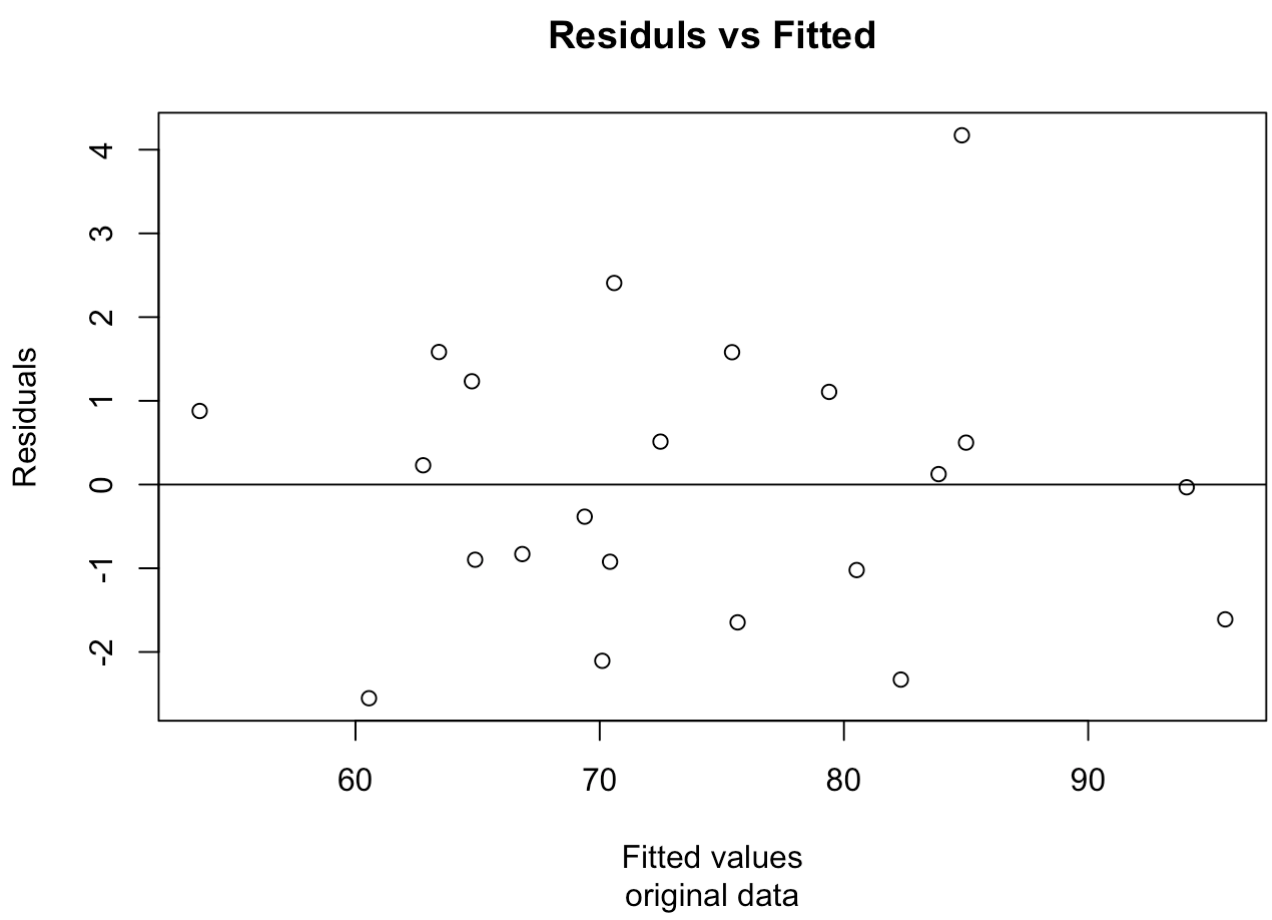
```
# preparing data
physical <- read_csv("./q2.csv")
```

(a)

```
# Making original model
physical.lm <- lm(Mass ~ 1 + Fore + Bicep + Chest + Neck + Shoulder + Waist + Height + Calf + Thig
h + Head, data=physical)
physical.scale = c(min(physical.lm$residuals), max(physical.lm$residuals))
summary(physical.lm)
```

```
##
## Call:
## lm(formula = Mass ~ 1 + Fore + Bicep + Chest + Neck + Shoulder +
##     Waist + Height + Calf + Thigh + Head, data = physical)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.552 -0.996  0.046  1.050  4.172
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -69.5171    29.0374   -2.39  0.03561 *
## Fore          1.7818     0.8547    2.08  0.06120 .
## Bicep         0.1551     0.4853    0.32  0.75527
## Chest         0.1891     0.2258    0.84  0.42013
## Neck         -0.4818     0.7207   -0.67  0.51754
## Shoulder     -0.0293     0.2394   -0.12  0.90477
## Waist         0.6614     0.1165    5.68  0.00014 ***
## Height        0.3178     0.1304    2.44  0.03294 *
## Calf          0.4459     0.4125    1.08  0.30286
## Thigh         0.2972     0.3051    0.97  0.35092
## Head         -0.9196     0.5201   -1.77  0.10474
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.29 on 11 degrees of freedom
## Multiple R-squared:  0.977,  Adjusted R-squared:  0.956
## F-statistic: 47.2 on 10 and 11 DF,  p-value: 1.41e-07
```

```
plot(physical.lm$fitted,
     physical.lm$residuals,
     main="Residuls vs Fitted",
     sub="original data",
     xlab="Fitted values",
     ylab="Residuals",
   abline(h = 0))
```



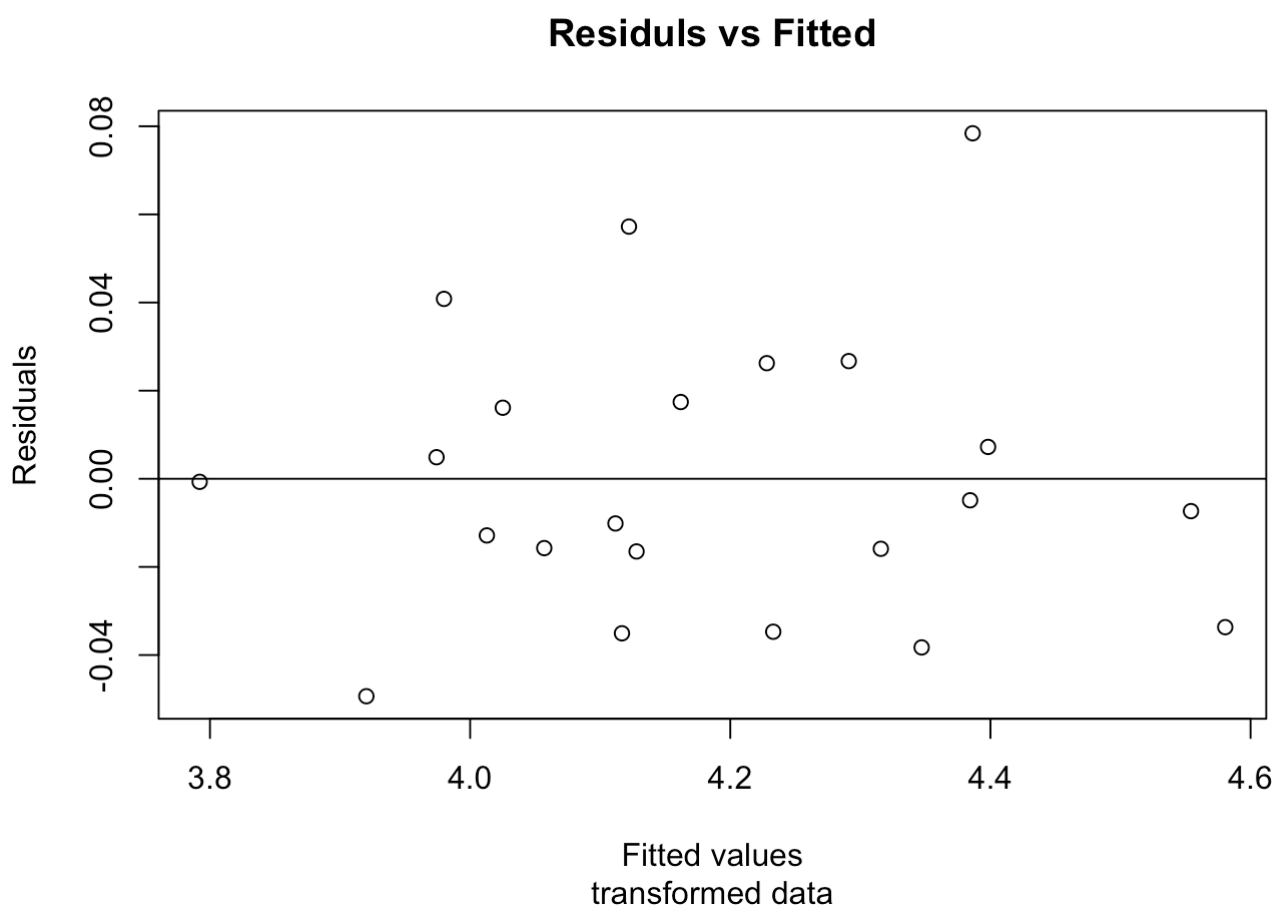
(b)

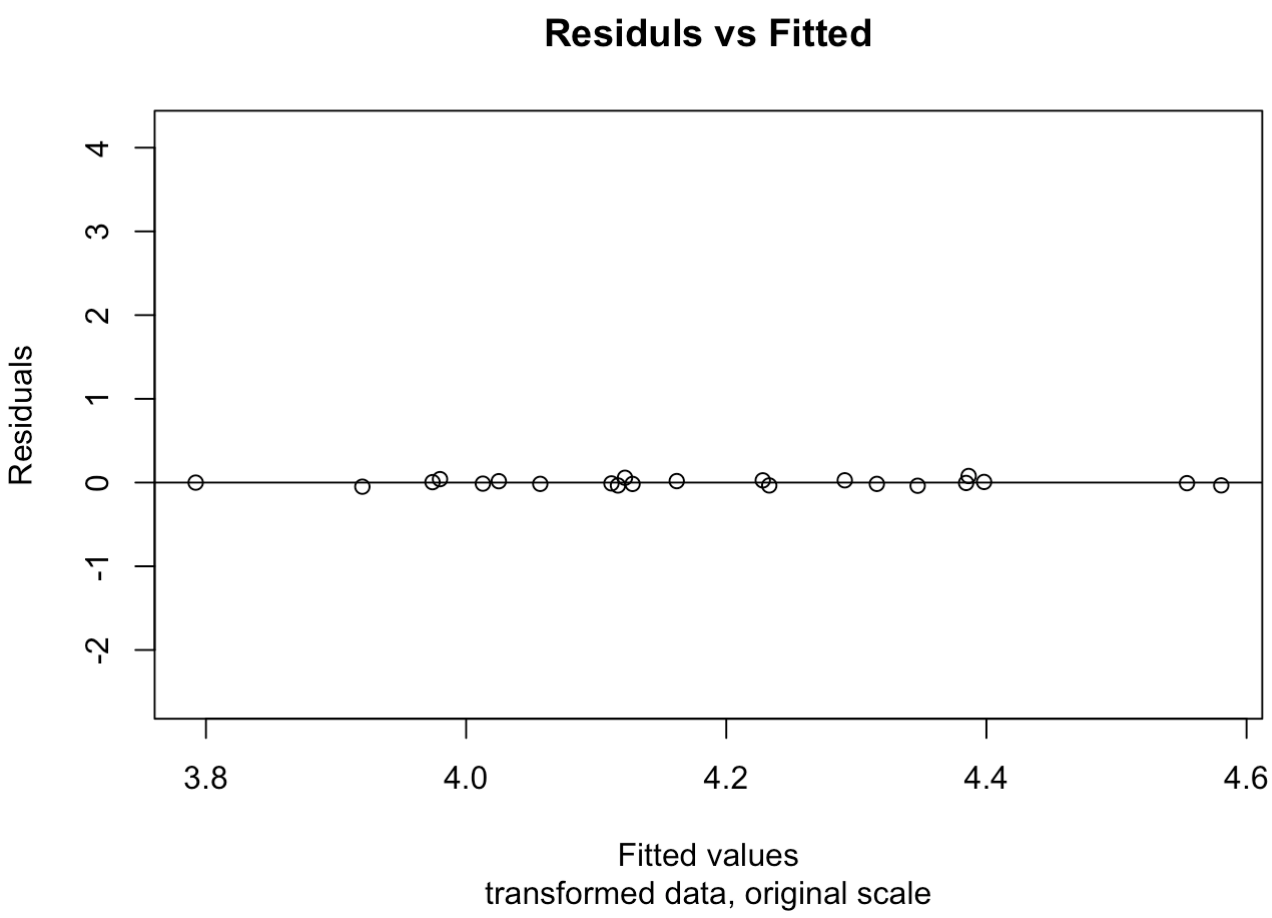
From the residuals vs fitted graph, the errors seem to be about normally shaped with a mean of 0.

```
# Making cube root model
physical.lm_trans <- lm((Mass)^(1/3) ~ 1 + Fore + Bicep + Chest + Neck + Shoulder + Waist + Height
 + Calf + Thigh + Head, data=physical)
summary(physical.lm_trans)
```

```
##
## Call:
## lm(formula = (Mass)^(1/3) ~ 1 + Fore + Bicep + Chest + Neck +
##     Shoulder + Waist + Height + Calf + Thigh + Head, data = physical)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.04935 -0.01634 -0.00611  0.01710  0.07841
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.11923    0.56423    1.98  0.07282 .
## Fore         0.02797    0.01661    1.68  0.12027
## Bicep        0.00414    0.00943    0.44  0.66886
## Chest        0.00105    0.00439    0.24  0.81496
## Neck        -0.00253    0.01400   -0.18  0.85980
## Shoulder     0.00081    0.00465    0.17  0.86494
## Waist        0.01115    0.00226    4.93  0.00045 ***
## Height       0.00577    0.00253    2.28  0.04359 *
## Calf         0.01066    0.00802    1.33  0.21061
## Thigh        0.00792    0.00593    1.34  0.20861
## Head        -0.01245    0.01011   -1.23  0.24358
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0444 on 11 degrees of freedom
## Multiple R-squared:  0.976,  Adjusted R-squared:  0.954
## F-statistic: 44.4 on 10 and 11 DF,  p-value: 1.93e-07
```

```
# Comparing residuals of two models
plot(physical.lm_trans$fitted,
     physical.lm_trans$residuals,
     main="Residuls vs Fitted",
     sub="transformed data",
     xlab="Fitted values",
     ylab="Residuals",
     abline(h = 0))
```





(c)

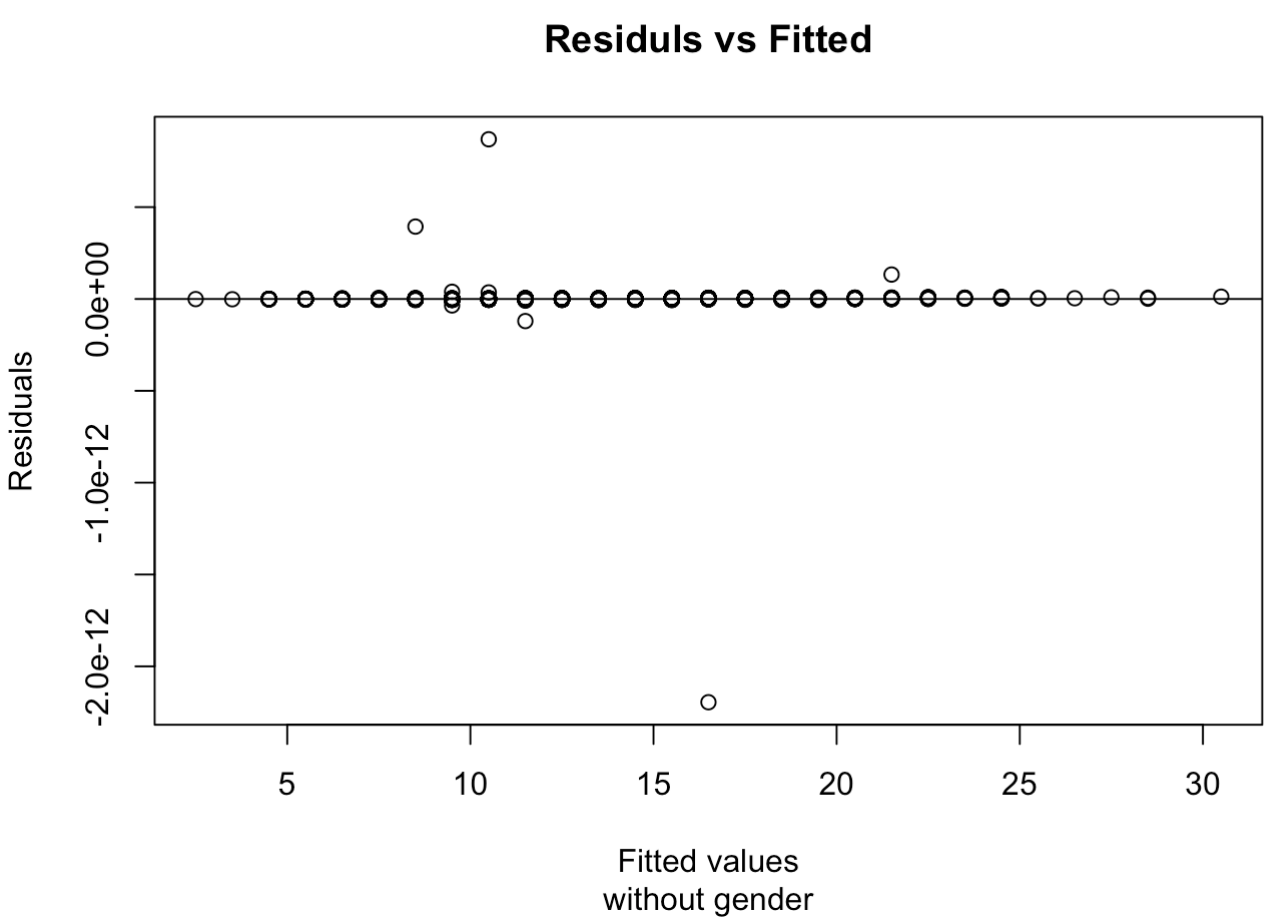
The residuals of the transformed model but on the scale of the orignal model, are really close to 0. Therefore the transformed model is clearly better.

Question 7.11

```
# preparing data
abalone <- read_csv("./q3.csv")
```

(a)

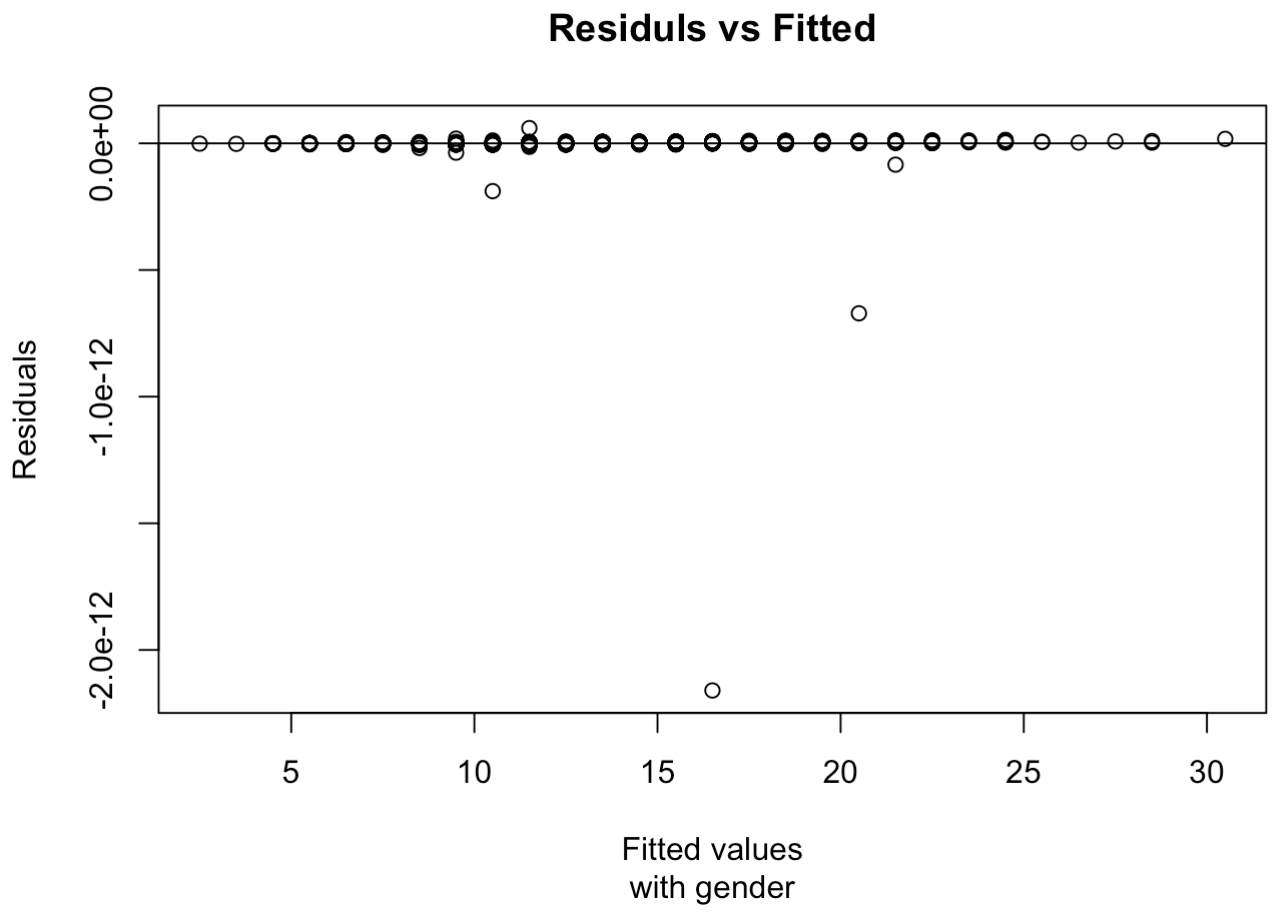
```
abalone.lm <- lm(age ~ 1 + length + diameter + height + whole_weight + shuckled_weight + viscera_w
eight + shell_weight + rings, data = abalone)
plot(abalone.lm$fitted,
     abalone.lm$residuals,
     main="Residuls vs Fitted",
     sub="without gender",
     xlab="Fitted values",
     ylab="Residuals",
     abline(h = 0))
```



(b)

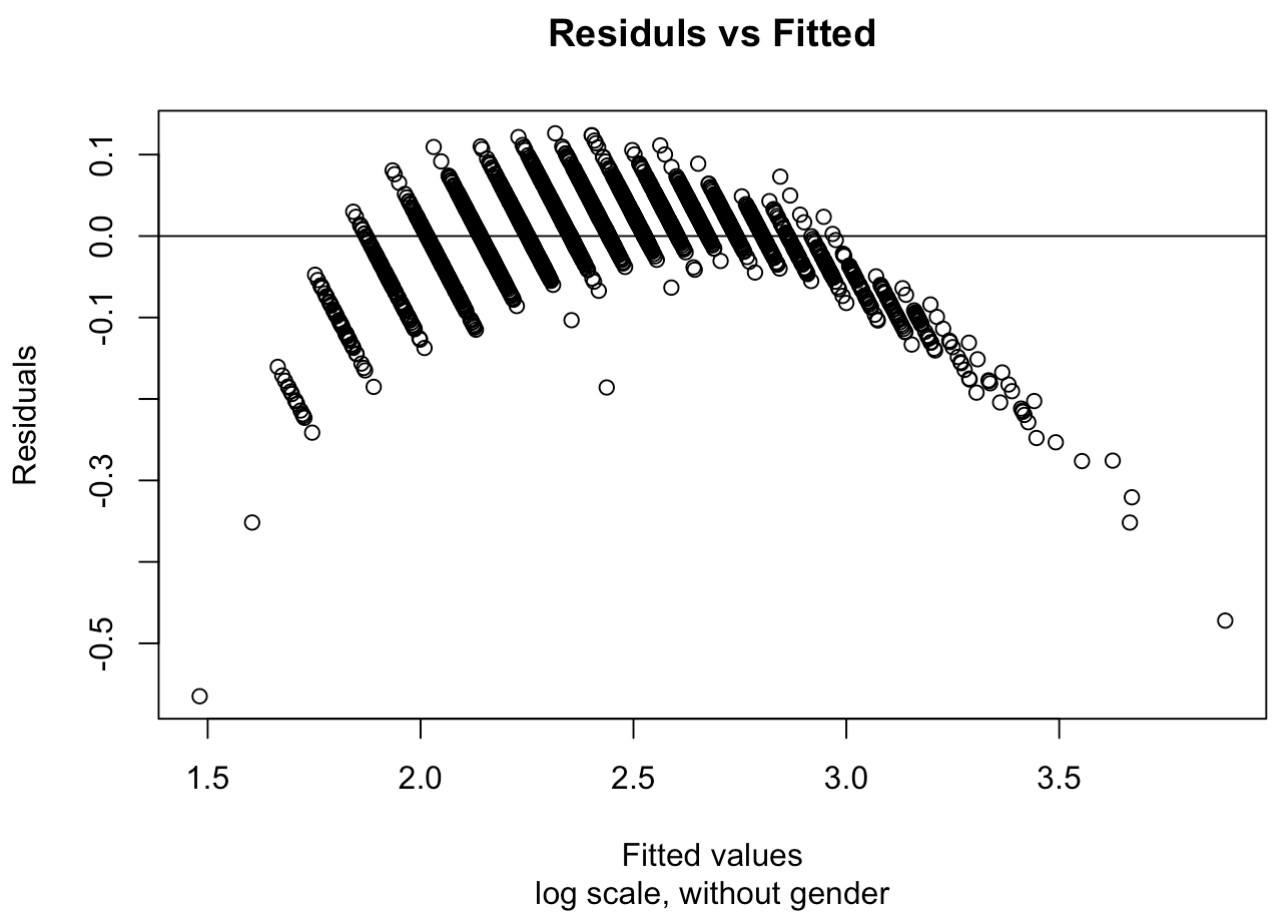
```
abalone.with_gender <- read_csv("./q3.csv")
abalone.with_gender[abalone.with_gender["sex"] == "F",]["sex"]<-"-1"
abalone.with_gender[abalone.with_gender["sex"] == "M",]["sex"]<-"1"
abalone.with_gender[abalone.with_gender["sex"] == "I",]["sex"]<-"0"
abalone.with_gender["sex"] = as.numeric(unlist(abalone.with_gender["sex"]))

abalone.lm_gender <- lm(age ~ 1 + sex + length + diameter + height + whole_weight + shuckled_weigh
t + viscera_weight + shell_weight + rings, data = abalone.with_gender)
plot(abalone.lm_gender$fitted,
     abalone.lm_gender$residuals,
     main="Residuls vs Fitted",
     sub="with gender",
     xlab="Fitted values",
     ylab="Residuals",
     abline(h = 0))
```



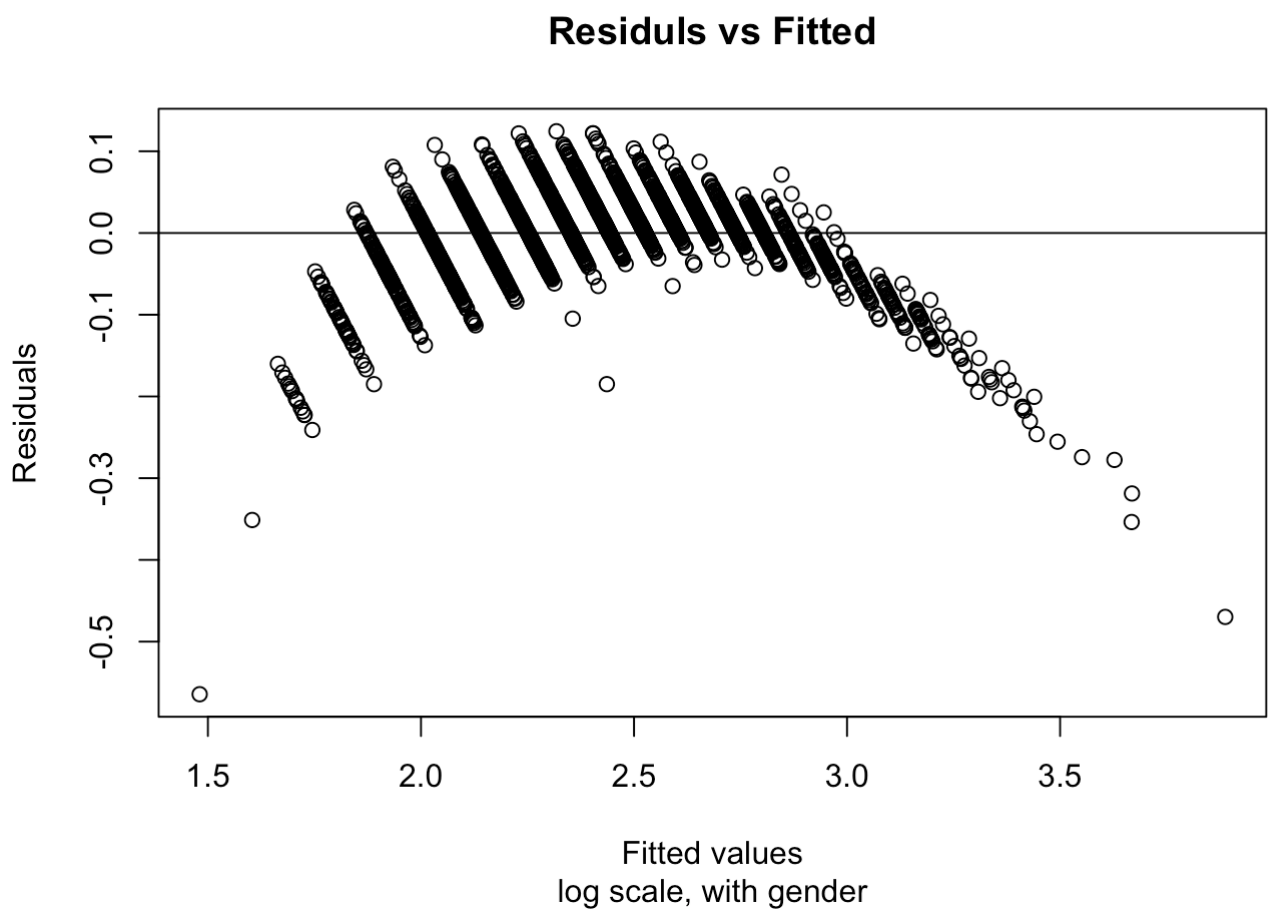
(c)

```
abalone.lm_log <- lm(log(age) ~ 1 + length + diameter + height + whole_weight + shuckled_weight +
viscera_weight + shell_weight + rings, data = abalone)
plot(abalone.lm_log$fitted,
     abalone.lm_log$residuals,
     main="Residuls vs Fitted",
     sub="log scale, without gender",
     xlab="Fitted values",
     ylab="Residuals",
     abline(h = 0))
```



(d)

```
abalone.lm_gender_log <- lm(log(age) ~ 1 + sex + length + diameter + height + whole_weight + shuck
led_weight + viscera_weight + shell_weight + rings, data = abalone.with_gender)
plot(abalone.lm_gender_log$fitted,
     abalone.lm_gender_log$residuals,
     main="Residuls vs Fitted",
     sub="log scale, with gender",
     xlab="Fitted values",
     ylab="Residuals",
     abline(h = 0))
```



(e) From the residuals vs fitted graph, it is clear that using rings as a predictors decreases the range of residuals by several orders of magnitude (both using and excluding gender). Also using gender also eliminates most of the positive residuals without effecting the negative residuals.

However, using an extra predictor for age doesn't really give such a big benefit. With more predictors, there is a higher chance of overfitting and the simpler model (tends) to be less accurate but tends to be better.

We would chose the model without gender predicting the age, NOT the log of age because it minimizes residuals and is less likely to overfit. **(f)**

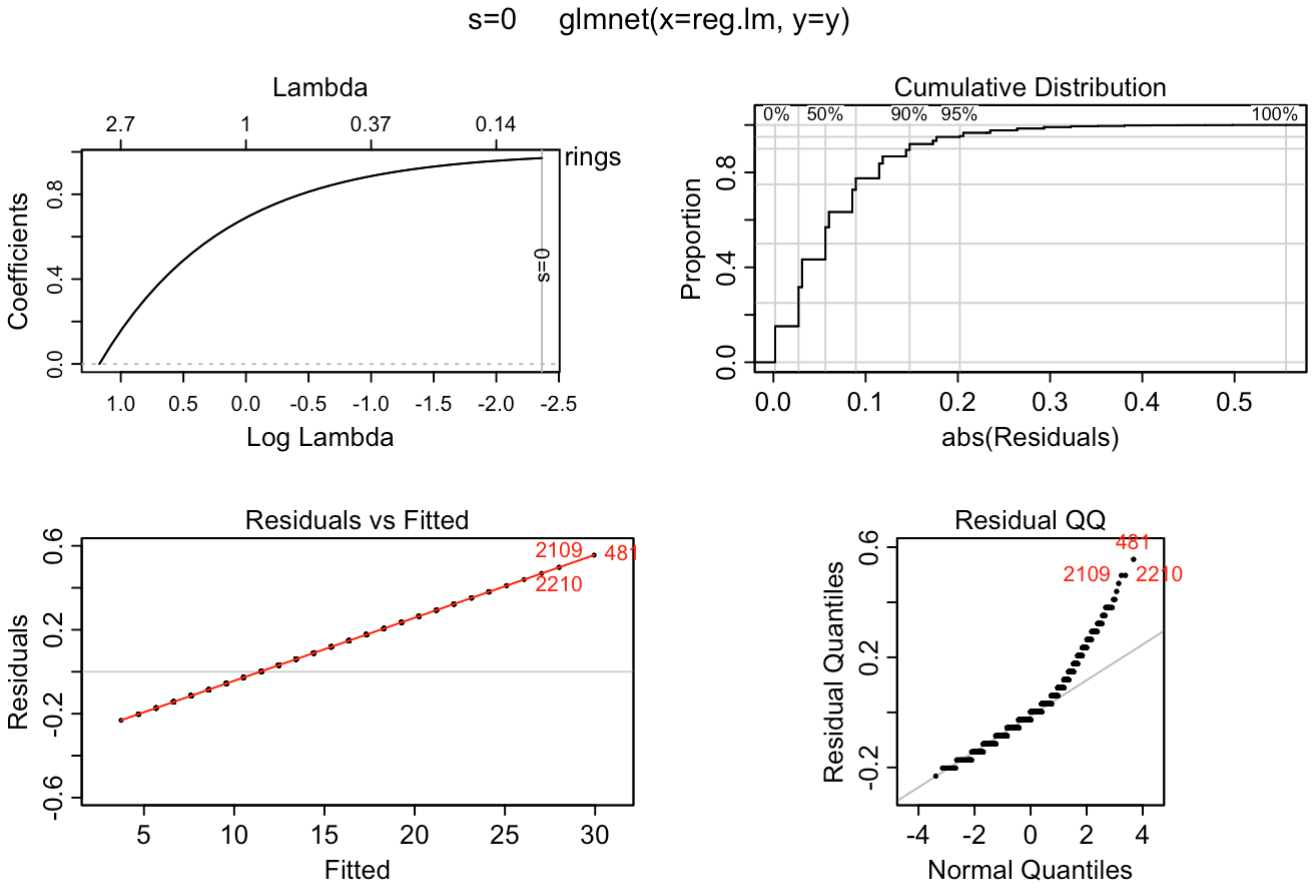
```
library(glmnet)
library(plotmo)
```

```
## Warning: package 'plotmo' was built under R version 3.3.2
```

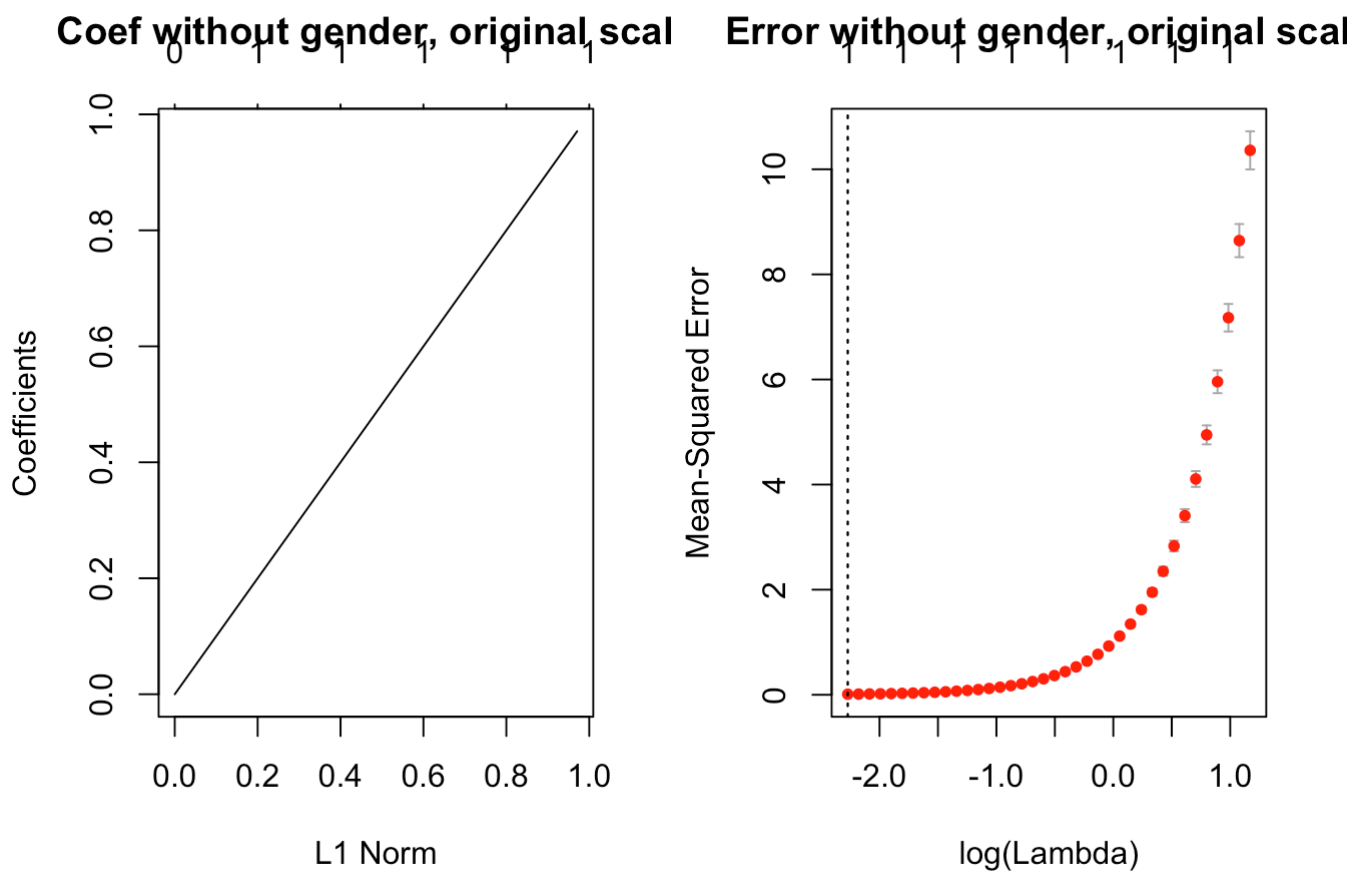
```
## Warning: package 'plotrix' was built under R version 3.3.2
```

```
y <- abalone$age

reg.lm <- as.matrix(data.frame(model.matrix(age ~ 1 + length + diameter + height + whole_weight +
shuckled_weight + viscera_weight + shell_weight + rings, data = abalone)))
g.lm <- glmnet(reg.lm, y)
c.lm <- cv.glmnet(reg.lm, y)
plotres(g.lm)
```

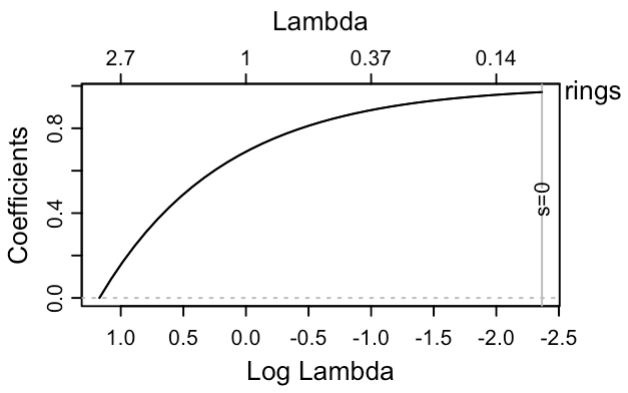


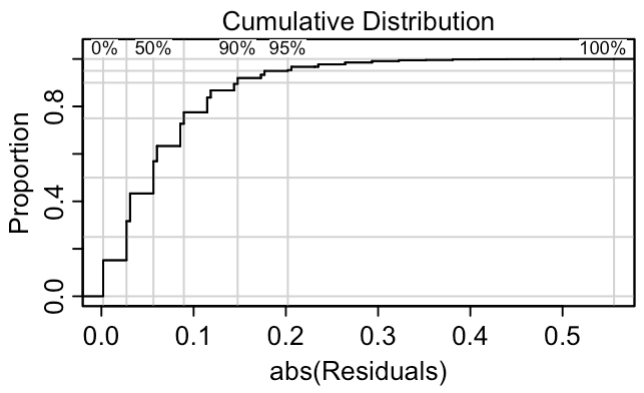

```
par(mfrow=c(1,2))
plot(g.lm, main="Coef without gender, original scale")
plot(c.lm, main="Error without gender, original scale")
```

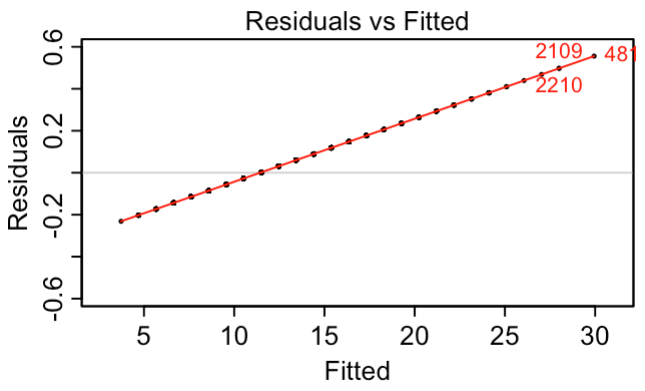


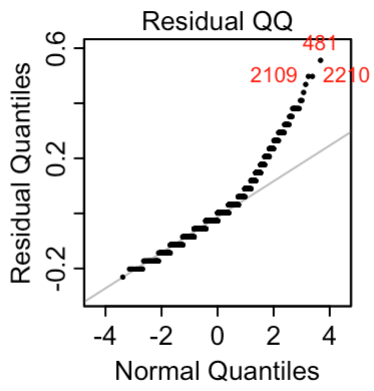

```
reg.gender <- as.matrix(data.frame(model.matrix(age ~ 1 + sex + length + diameter + height + whole
_weight + shuckled_weight + viscera_weight + shell_weight + rings, data = abalone.with_gender)))
g.gender <- glmnet(reg.gender, y)
c.gender <- cv.glmnet(reg.gender, y)
plotres(g.gender)
```

s=0 glmnet(x=reg.gender, y=...

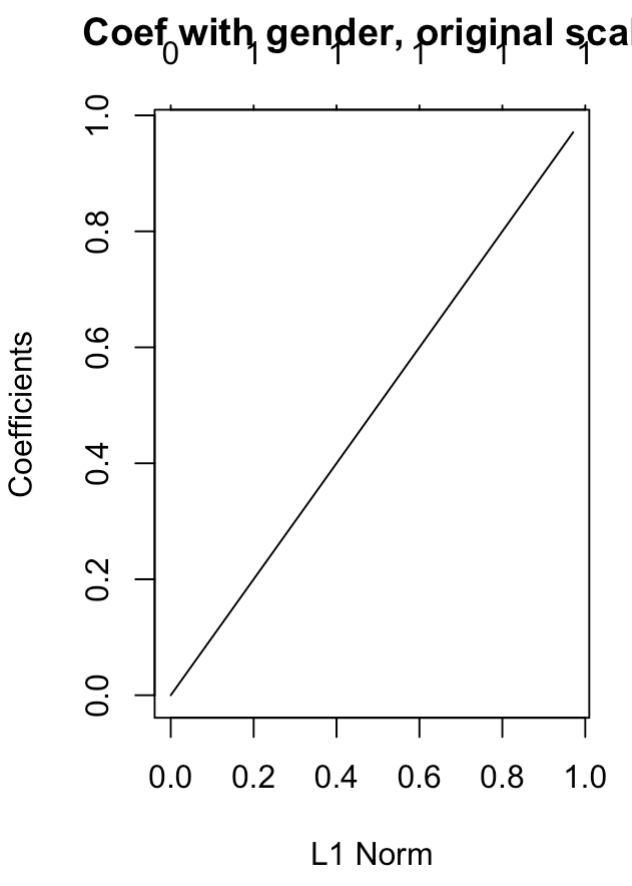


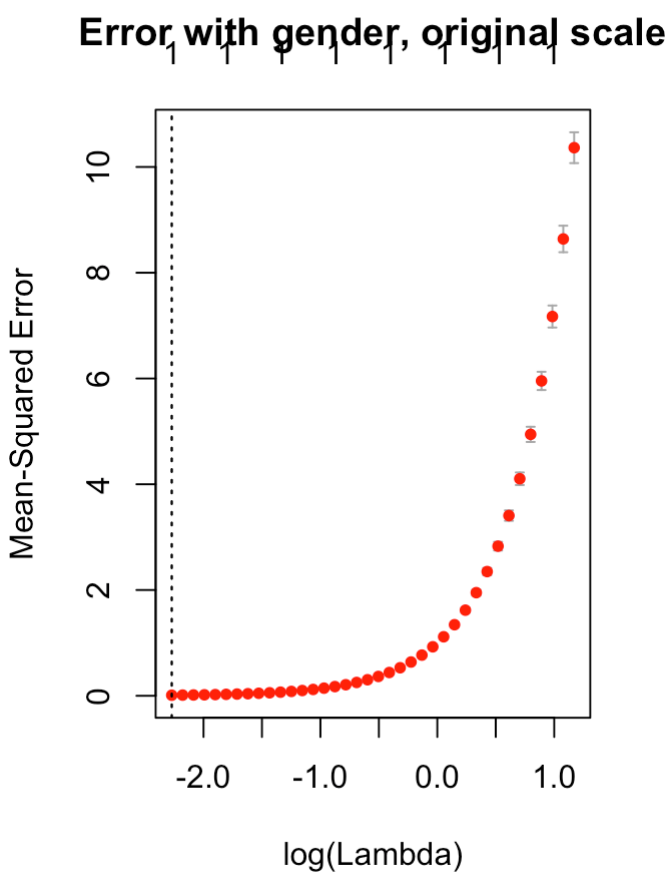




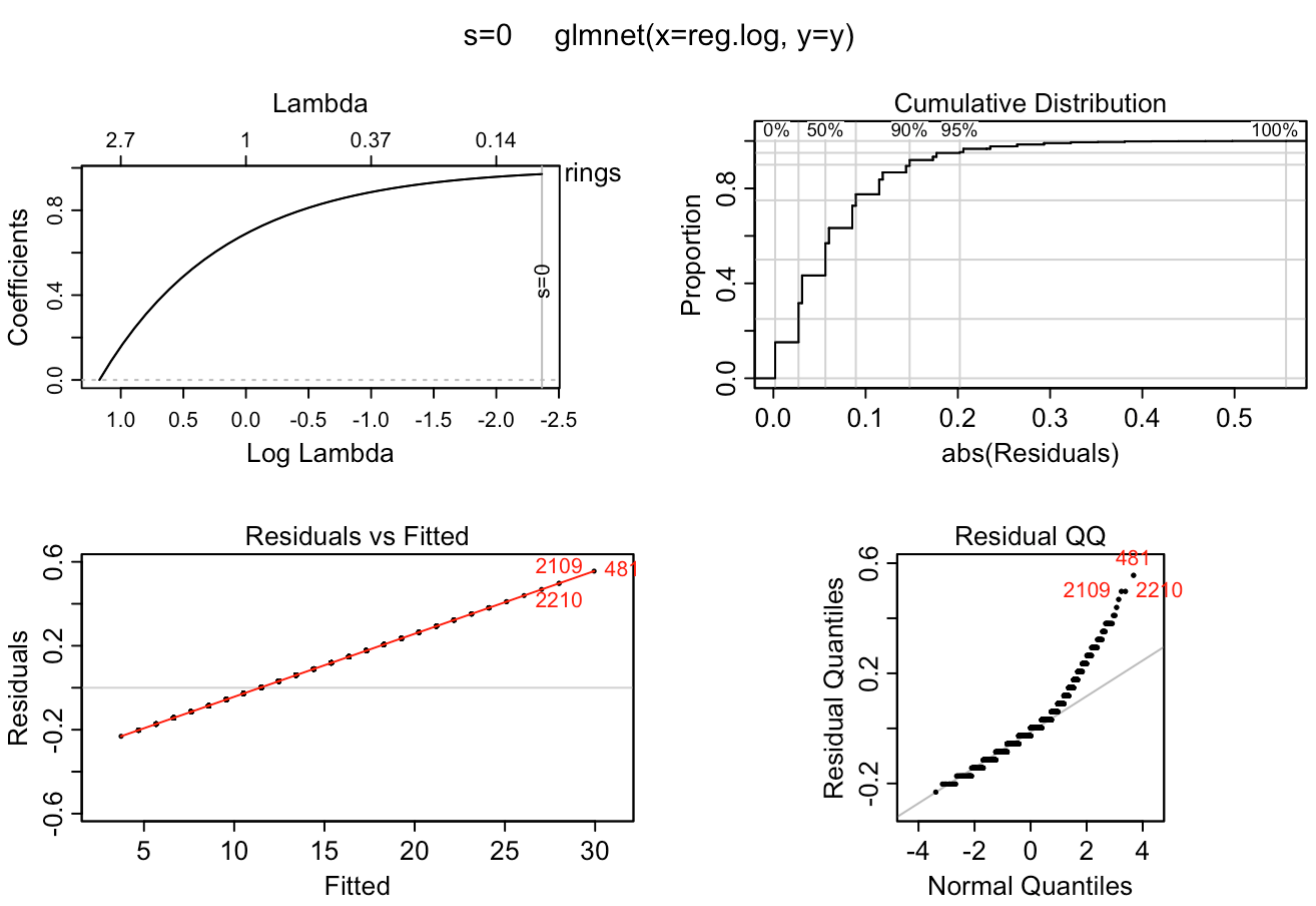



```
plot(g.gender, main="Coef with gender, original scale")
plot(c.gender, main="Error with gender, original scale")
```

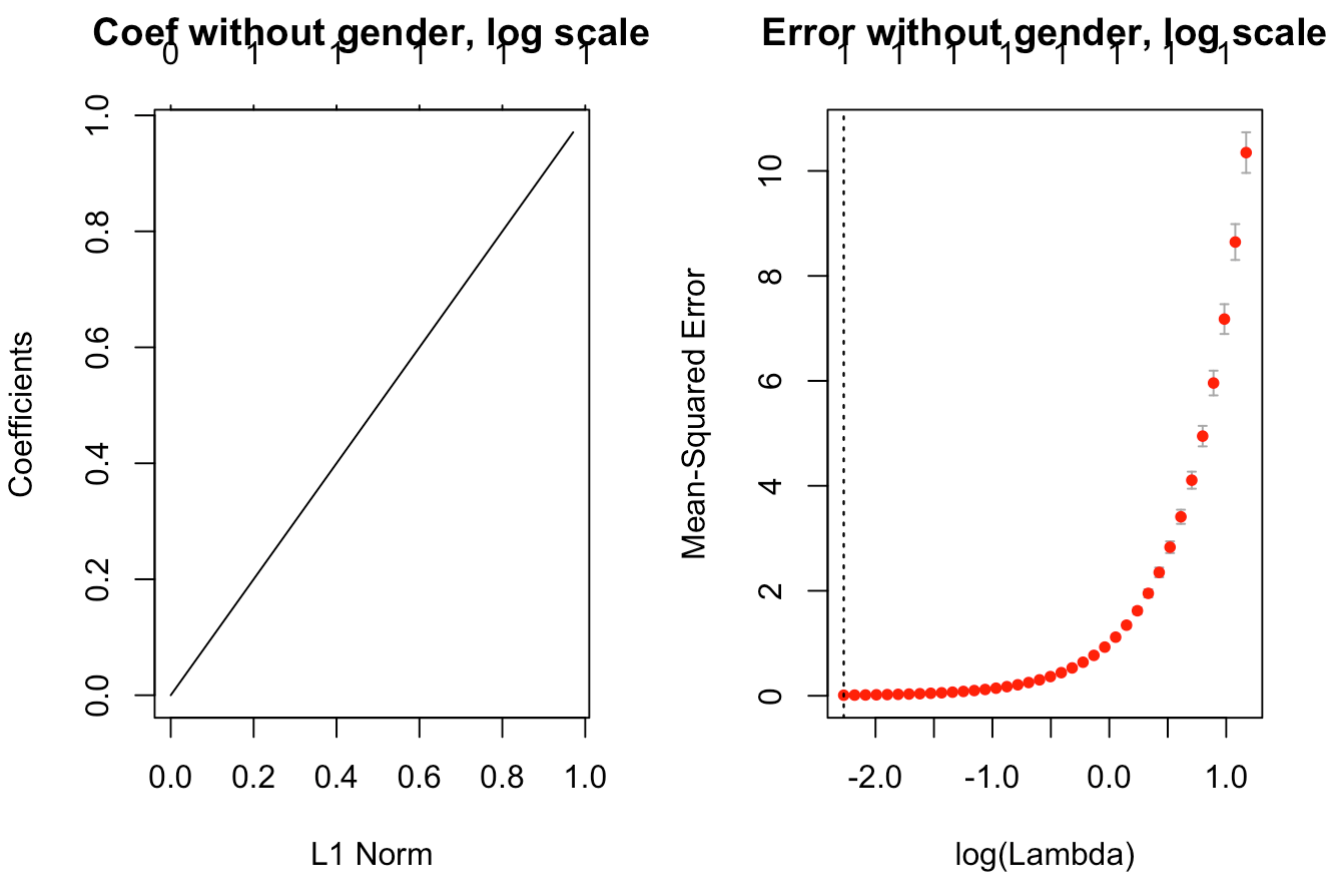




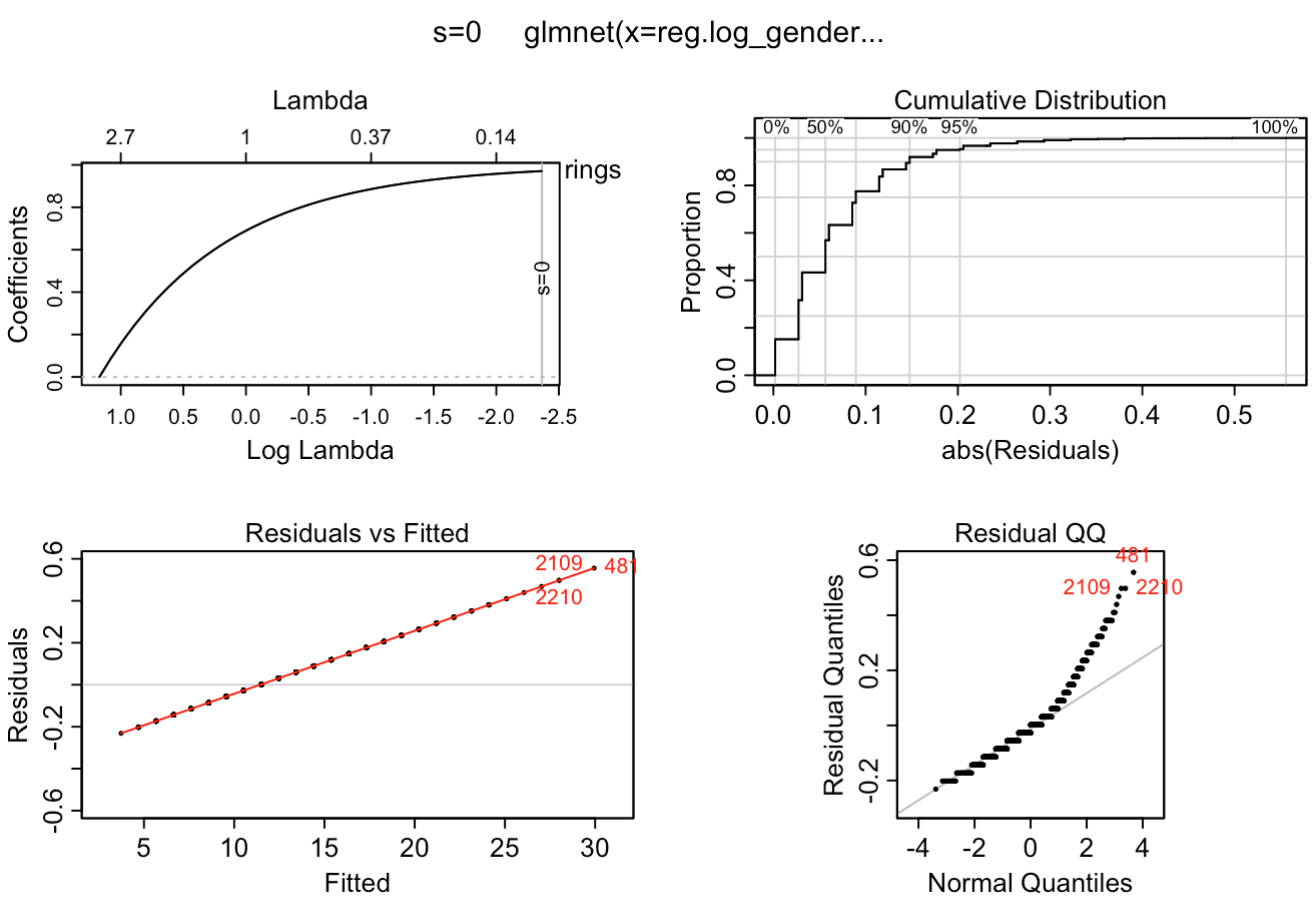

```
reg.log<- as.matrix(data.frame(model.matrix(log(age) ~ 1 + length + diameter + height + whole_weig
ht + shuckled_weight + viscera_weight + shell_weight + rings, data = abalone)))
g.log <- glmnet(reg.log, y)
c.log <- cv.glmnet(reg.log, y)
plotres(g.log)
```




```
plot(g.log, main="Coef without gender, log scale")
plot(c.log, main="Error without gender, log scale")
```




```
reg.log_gender <- as.matrix(data.frame(model.matrix(log(age) ~ 1 + sex + length + diameter + heigh
t + whole_weight + shuckled_weight + viscera_weight + shell_weight + rings, data = abalone.with_ge
nder)))
g.log_gender <- glmnet(reg.log_gender, y)
c.log_gender <- cv.glmnet(reg.log_gender, y)
plotres(g.log_gender)
```




```
plot(g.log_gender, main="Coef with gender, log scale")
plot(c.log_gender, main="Error with gender, log scale")
```