AML_HW5

Vishal Dalmiya (Dalmiya2); Himanshu Shah (Hs8); Deepak Nagarajan (deepakn2)

Mar 1, 2018

Problem 7.9

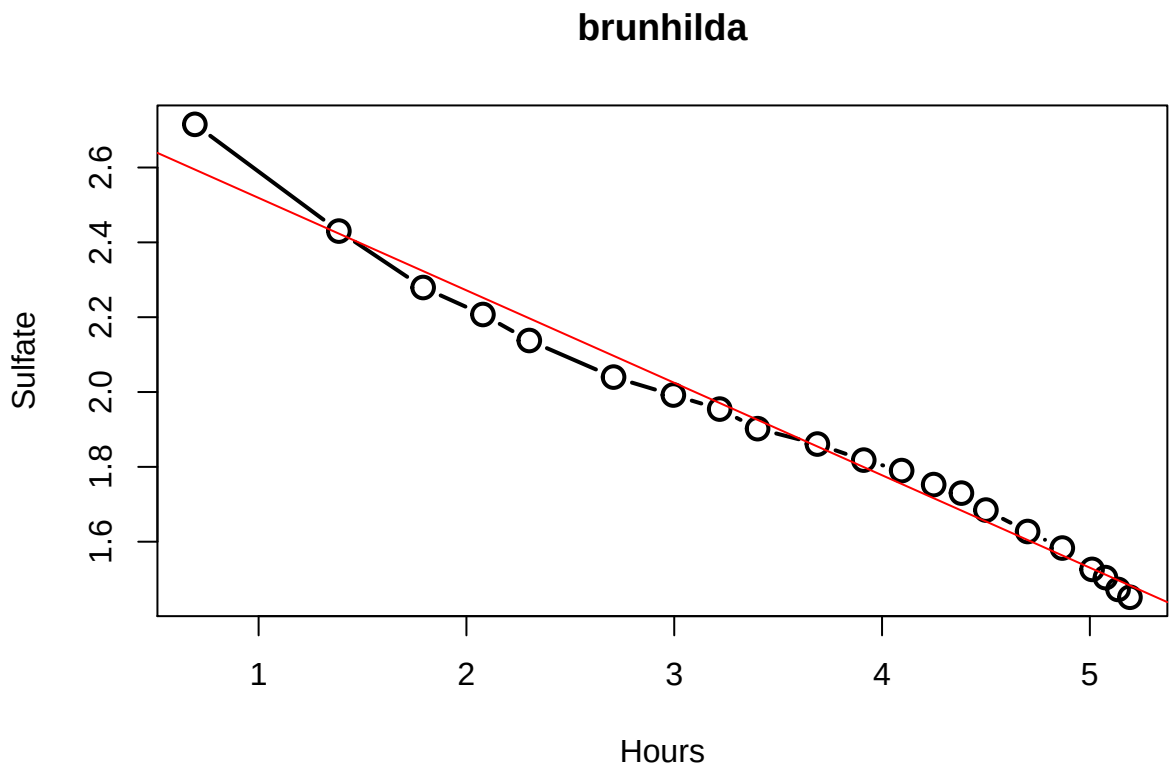
At http://www.statsci.org/data/general/brunhild.html, you will find a datasetthat measures the concentration of a sulfate in the blood of a baboon named Brunhilda as a function of time. Build a linear regression of the log of the concentration against the log of time.

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

```
library(readr)

brunhilda = read_delim("brunhild.txt",
                       "\t",
                       escape_double = FALSE,
                       trim_ws = TRUE)
```

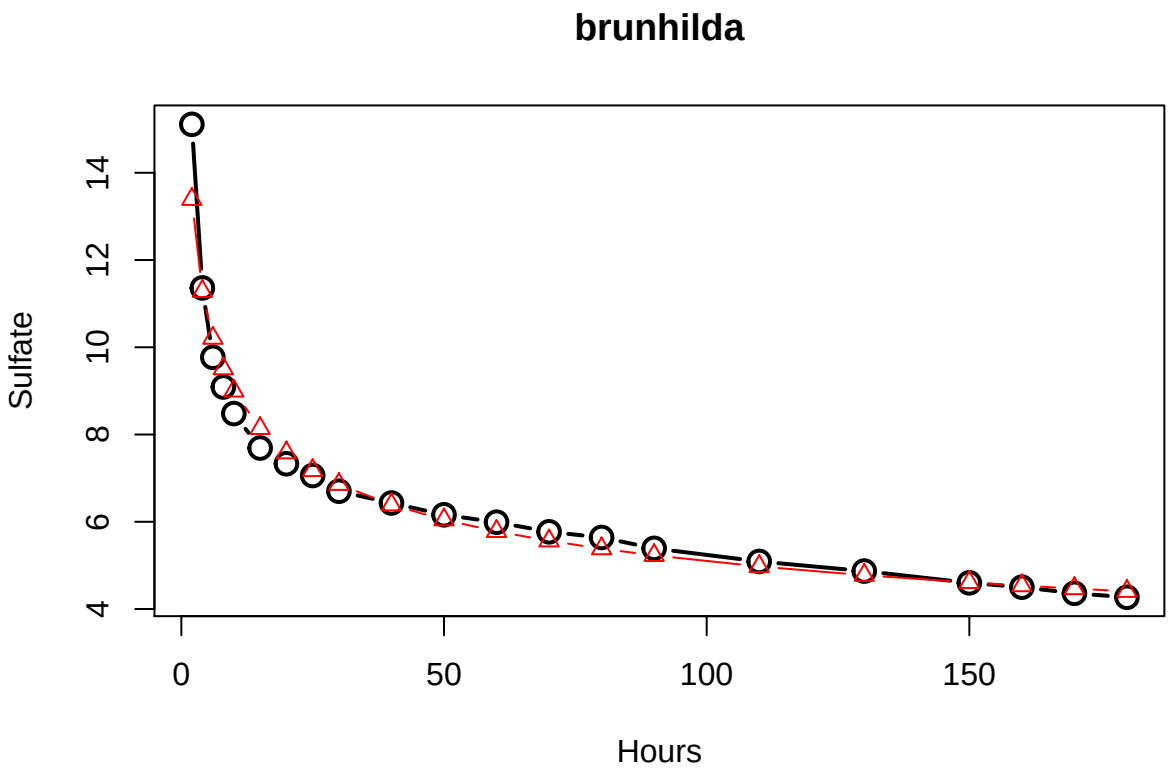
```
## Parsed with column specification:
## cols(
##   Hours = col_integer(),
##   Sulfate = col_double()
## )
```

```
brunhilda_model_log = lm(log(Sulfate) ~ log(Hours), data = brunhilda)

plot(
  log(brunhilda$Sulfate) ~ log(brunhilda$Hours),
  xlab = "Hours",
  ylab = "Sulfate",
  main = "brunhilda",
  col  = 1,
  cex  = 1.5,
  lwd  = 2,
  pch  = 1,
  type = "b"
)

abline(brunhilda_model_log, col = "red")
```



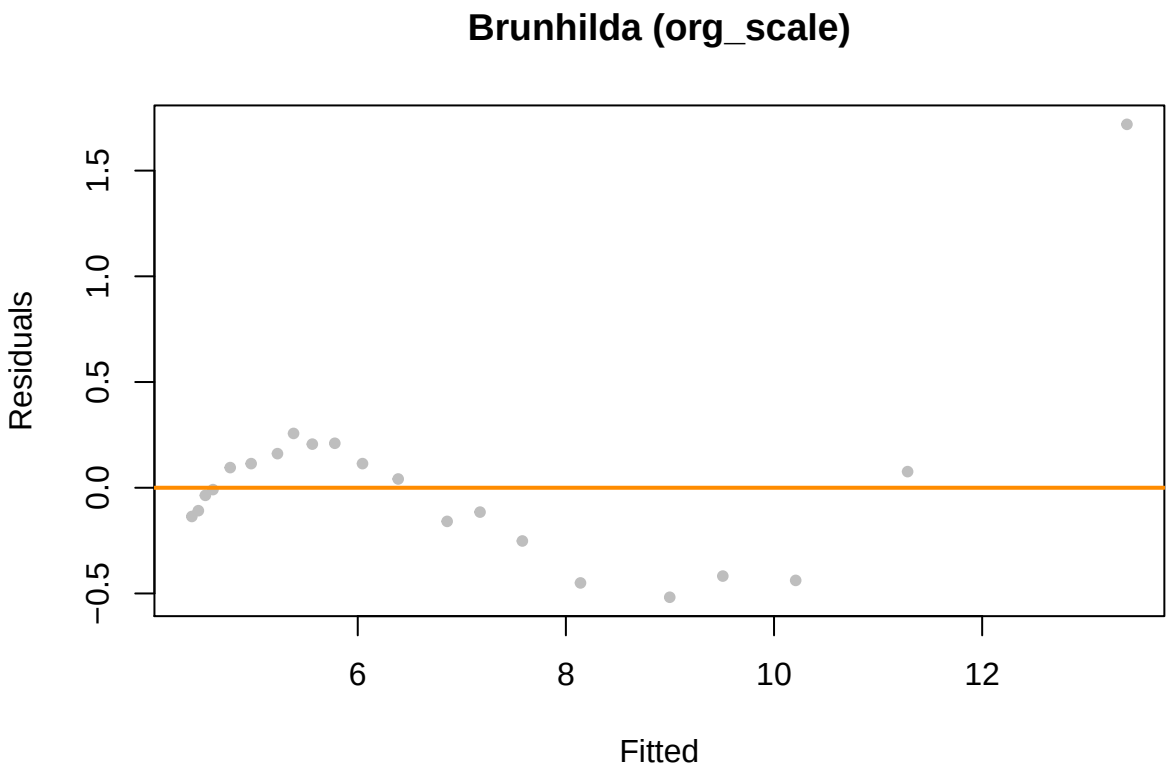
(b) Prepare a plot showing (a) the data points and (b) the regression curve in the original coordinate

```
# fitted values in original coordinates
# plot in original coordinates
plot(
  brunhilda$Sulfate ~ brunhilda$Hours,
  xlab = "Hours",
  ylab = "Sulfate",
  main = "brunhilda",
  col  = 1,
  cex  = 1.5,
  lwd  = 2,
  pch  = 1,
  type = "b"
)

lines(exp(brunhilda_model_log$fitted.values) ~ brunhilda$Hours,
      col = 2,
      type = "b",
      lwd = 1,
      pch = 2)
```

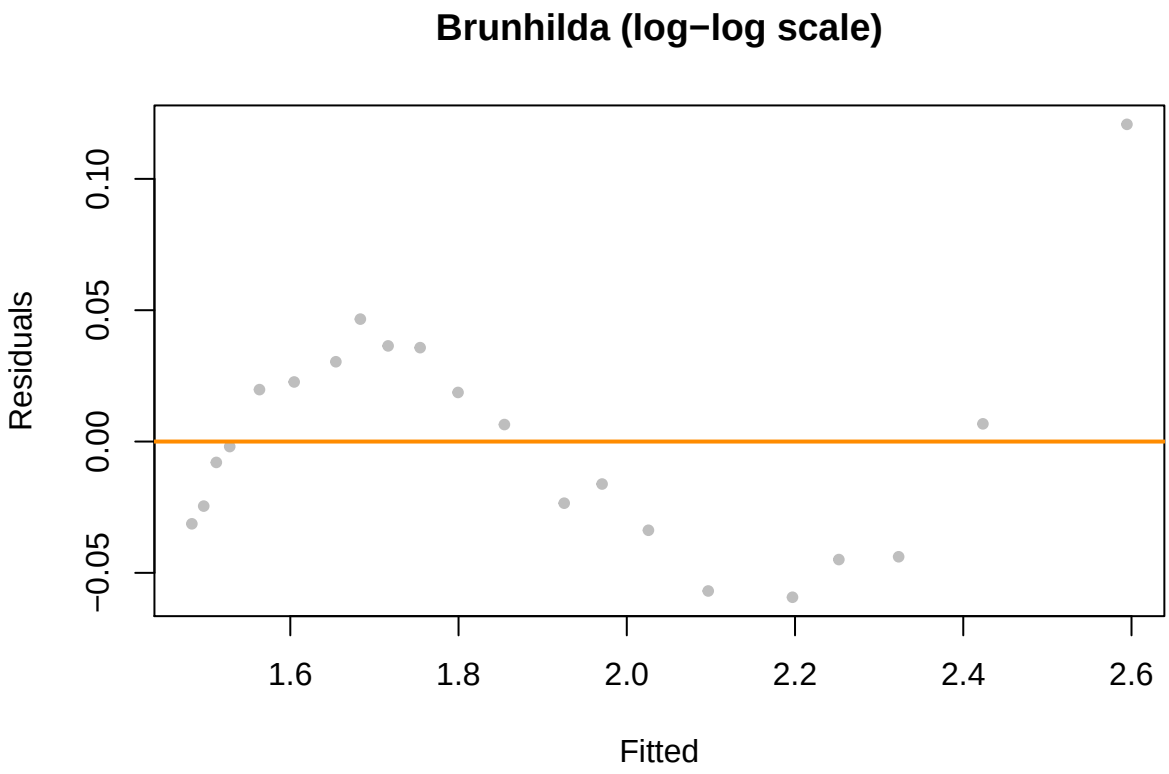


(c) Plot the residual against the fitted values in log-log and in original coordinates.

```
# fitted vs residuals in original coordinates
res_orgscale = brunhilda$Sulfate - exp(brunhilda_model_log$fitted.values)

plot(
  exp(brunhilda_model_log$fitted.values),
  res_orgscale,
  col = "grey",
  pch = 20,
  main = "Brunhilda (org_scale)",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```




```
# fitted vs residulas in log-log
plot(
  fitted(brunhilda_model_log),
  resid(brunhilda_model_log),
  col = "grey",
  pch = 20,
  main = "Brunhilda (log-log scale)",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```



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

The plot in **part b)** indicates that the regression fits the actual data really well. However the residual vs fitted plot shows that the residual is not entirely random and is a sin curve and kind of predictable given the fitted values. So it seems there could be a better regression out there which fits the data well and where the residual vs fitted plot is random.

Problem 7.10

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.

```
library(readr)
physical <- read_delim("physical.txt",
                       "\t", escape_double = FALSE, trim_ws = TRUE)
```

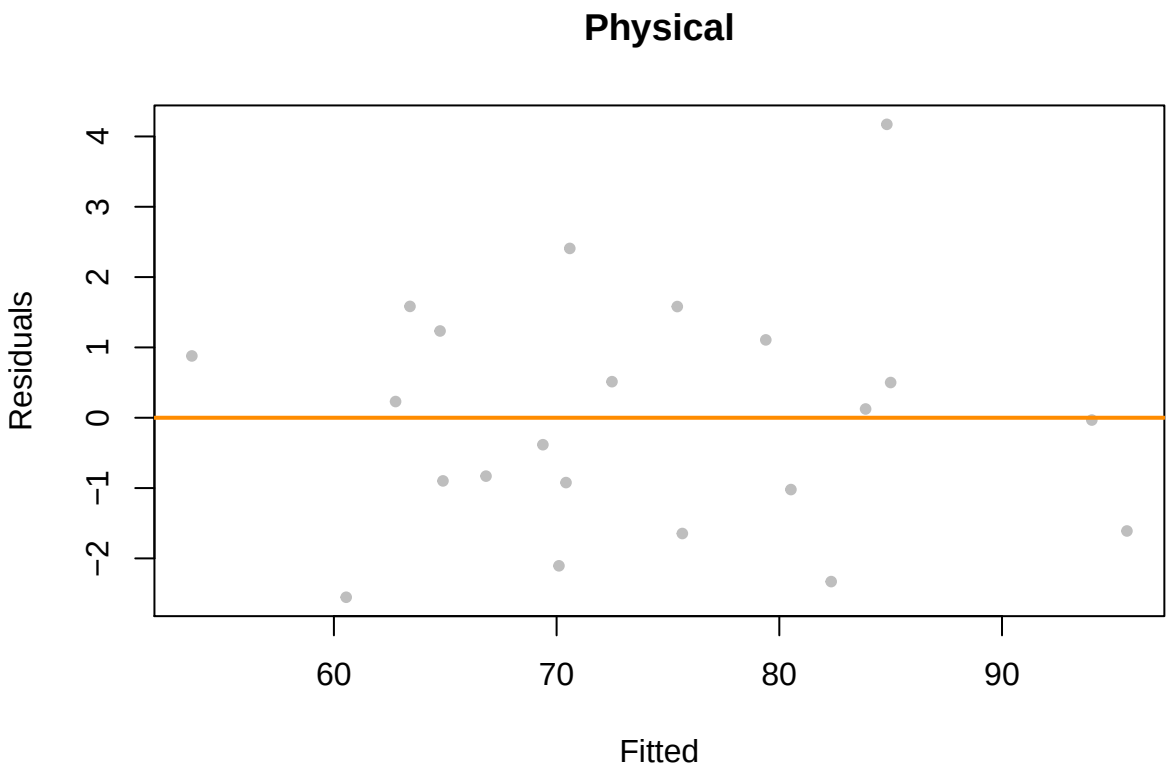
```
## Parsed with column specification:
## cols(
##   Mass = col_double(),
##   Fore = col_double(),
##   Bicep = col_double(),
```

```
##   Chest = col_double(),
##   Neck = col_double(),
##   Shoulder = col_double(),
##   Waist = col_double(),
##   Height = col_double(),
##   Calf = col_double(),
##   Thigh = col_double(),
##   Head = col_double()
## )
```

```
colnames(physical)
```

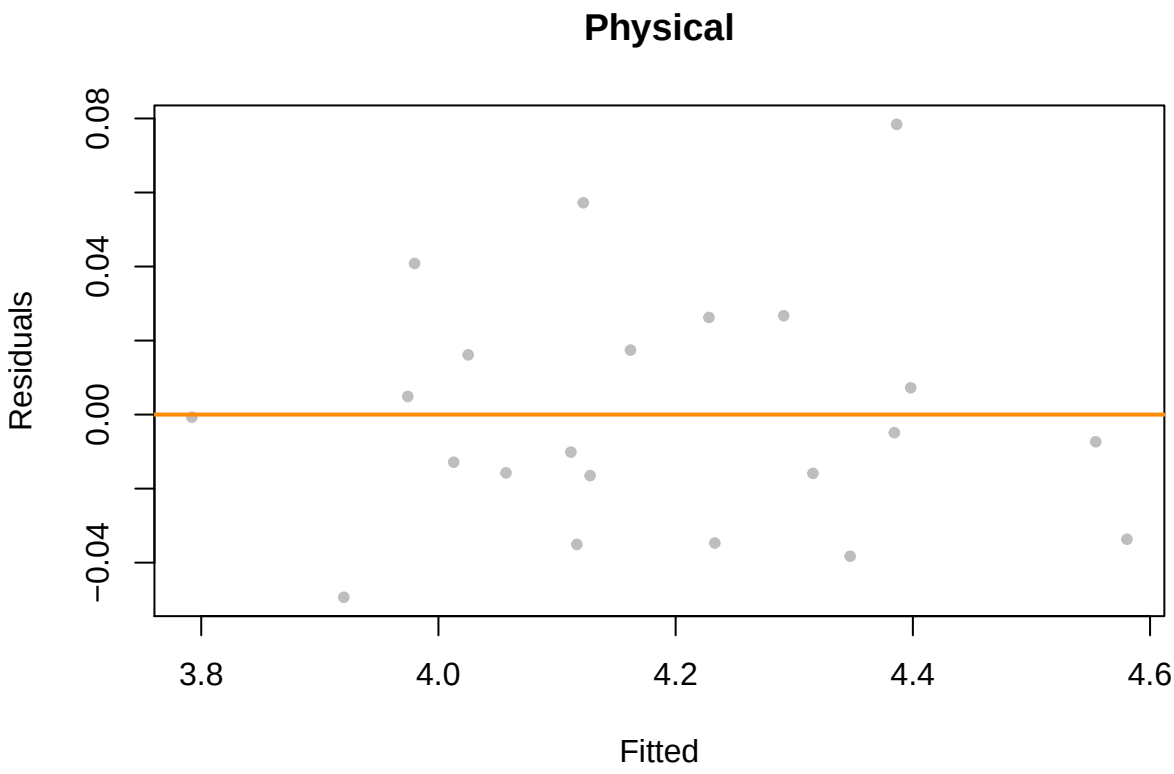
```
##  [1] "Mass"     "Fore"     "Bicep"    "Chest"    "Neck"     "Shoulder"
##  [7] "Waist"    "Height"   "Calf"     "Thigh"    "Head"
```

```
physical_model = lm(Mass ~ ., data = physical)

# fitted vs residuals
plot(
  fitted(physical_model),
  resid(physical_model),
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```

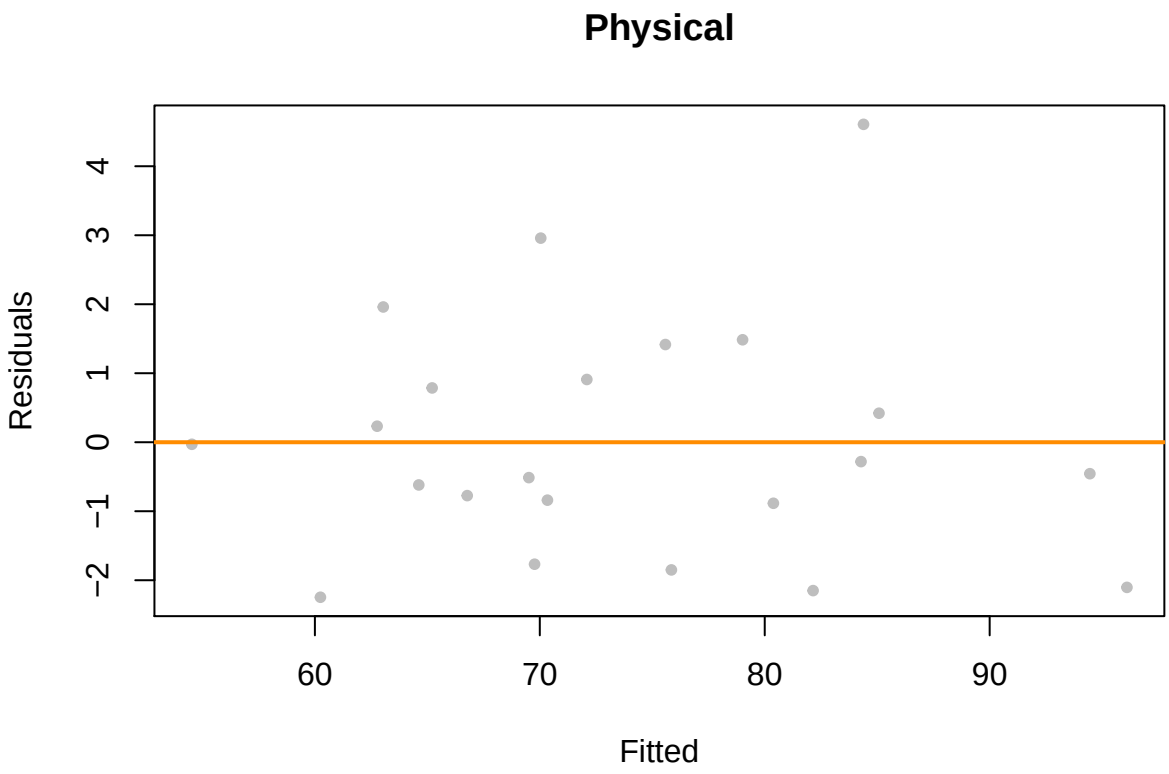


(b) Now regress the cube root of mass against these diameters. Plot the residual against the fitted values in both these cube root coordinates and in the original coordinates.

```
physical_model_cube_root = lm((Mass) ^ (1 / 3) ~ ., data = physical)

org_fitted = (physical_model_cube_root$fitted.values) ^ 3
org_resid = physical$Mass - org_fitted

# fitted vs residuals cube root co-ordinates
plot(
  fitted(physical_model_cube_root),
  resid(physical_model_cube_root),
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```

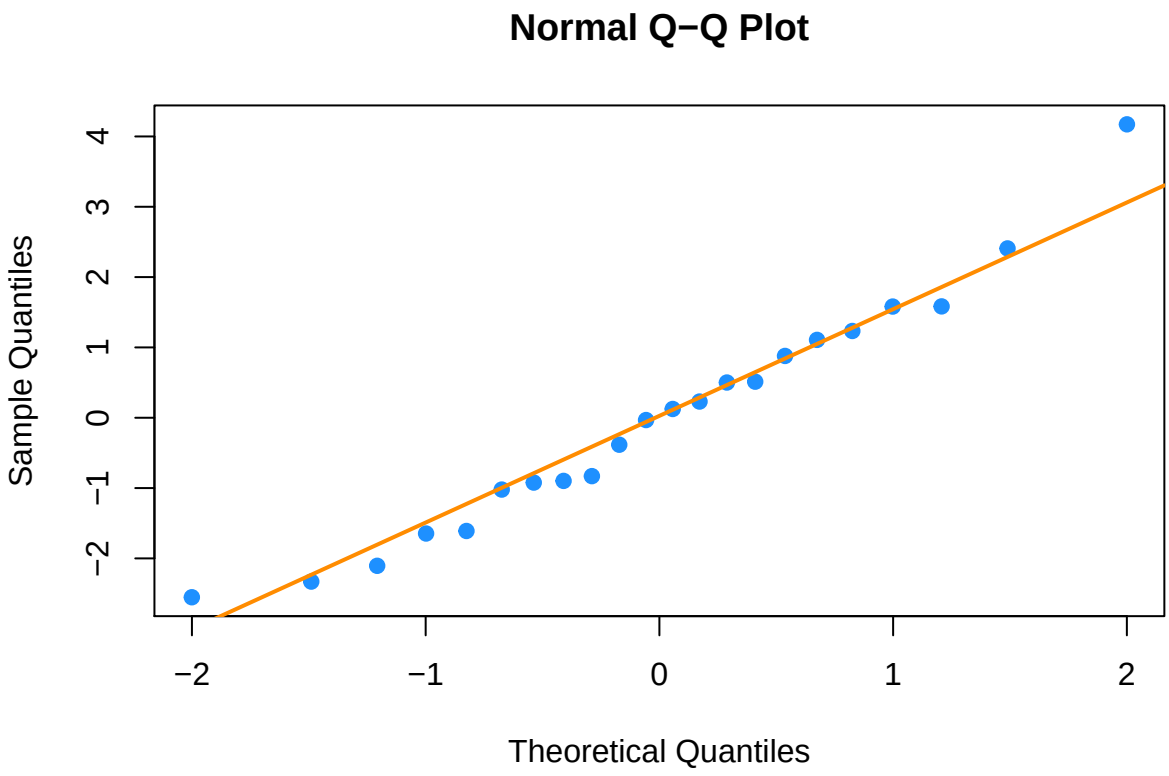



```r
# fitted vs residuals original co-ordinates
plot(
  org_fitted,
  org_resid,
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```

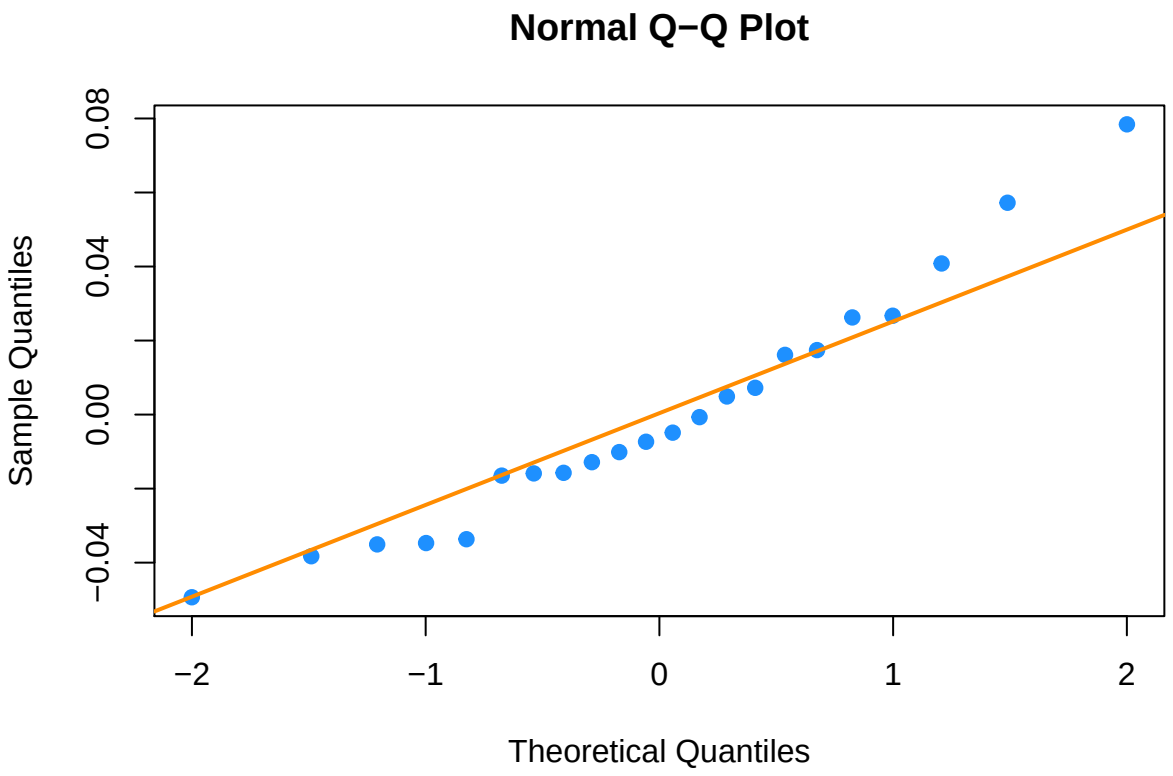


(c) Use your plots to explain which regression is better.

```
qqnorm(resid(physical_model), col = "dodgerblue", pch = 20, cex = 1.5)
qqline(resid(physical_model), col = "darkorange", lwd = 2)
```




```
qqnorm(resid(physical_model_cube_root), col = "dodgerblue", pch = 20, cex = 1.5)
qqline(resid(physical_model_cube_root), col = "darkorange", lwd = 2)
```




```
# R^2 values for the 2 models
c(summary(physical_model)$r.squared,
  summary(physical_model_cube_root)$r.squared)
```

```
## [1] 0.9772107 0.9758476
```

- The fitted vs residuals plot didn't show much difference as to which model is better.
- As per the results from Q-Q plot, the model without the cube root seems to be better, as you can see that the one with the cube root seems to form a heavier tail (comparatively) and also the distribution is slightly off from the line.
- Also looking at the R^2 values between the two model, the R^2 value is pretty comparable.
- Based on above we should chose the model which is simple (without cube root)

Problem 7.11

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.

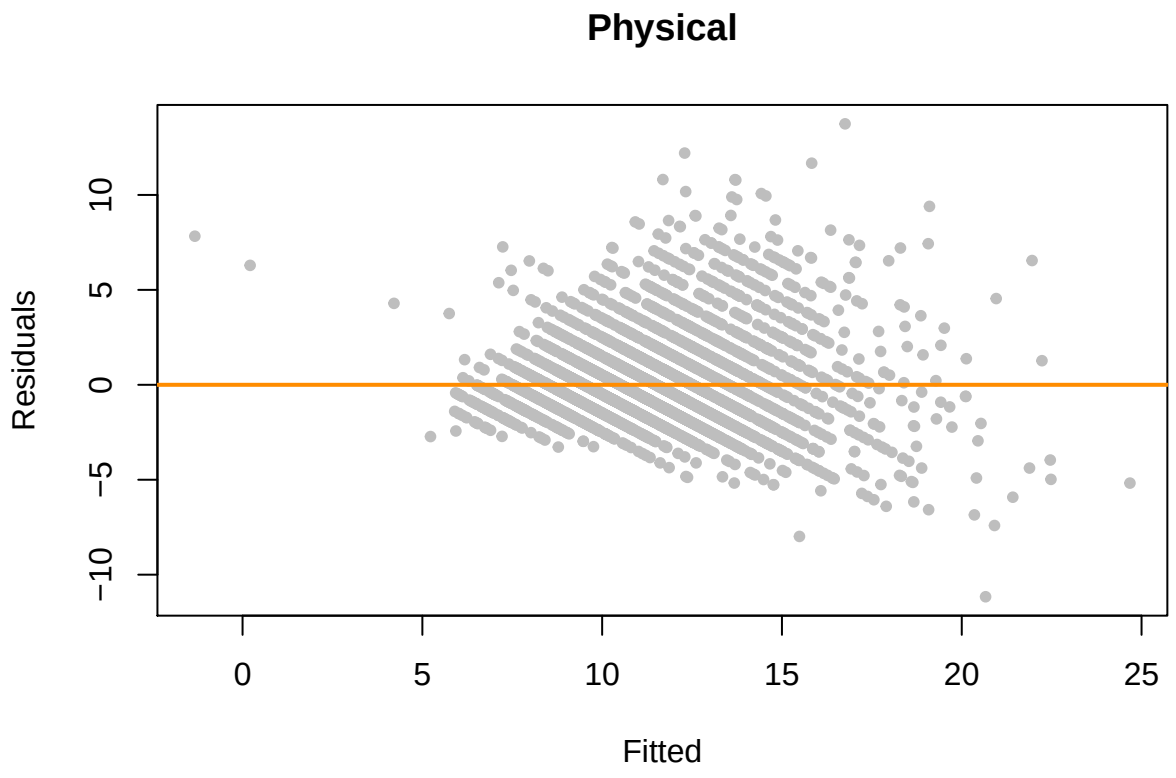
(a) Build a linear regression predicting the age from the measurements, ignoring gender. Plot the residual against the fitted values.

```
library(readr)
abalone <- read_csv("abalone.data",
                    col_names = FALSE)
```

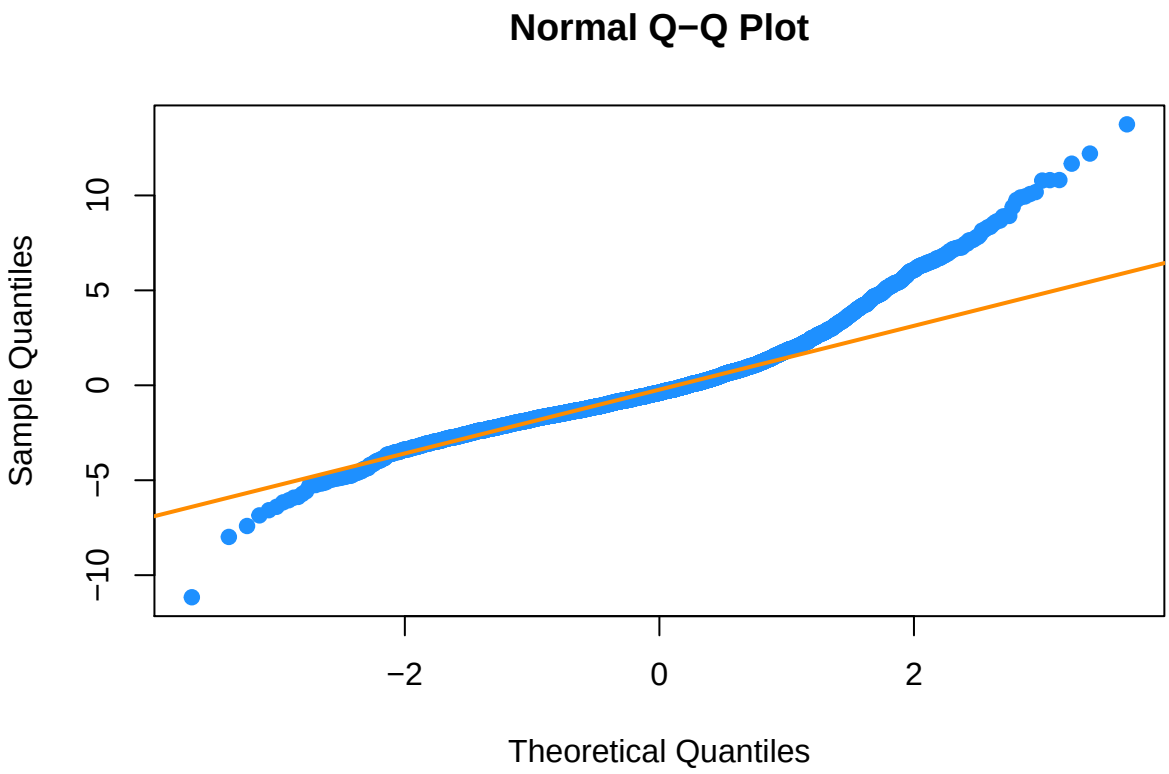
```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_double(),
##   X3 = col_double(),
##   X4 = col_double(),
##   X5 = col_double(),
##   X6 = col_double(),
##   X7 = col_double(),
##   X8 = col_double(),
##   X9 = col_integer()
## )
```

```
colnames(abalone) = c(
  "Sex",
  "Length",
  "Diameter",
  "Height",
  "Whole_Weight",
  "Shucked_Weight",
  "Viscera_Weight",
  "Shell_Weight",
  "Age"
)

abalone$Age = as.numeric(abalone$Age + 1.5)

abalone$Sex[abalone$Sex == "F"] = 1
abalone$Sex[abalone$Sex == "I"] = 0
abalone$Sex[abalone$Sex == "M"] = -1
abalone$Sex = as.integer(abalone$Sex)

abalone_model_1 = lm(Age ~ . - Sex, data = abalone)

# fitted vs residuals
plot(
  fitted(abalone_model_1),
  resid(abalone_model_1),
  col = "grey",
  pch = 20,
```

```
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```

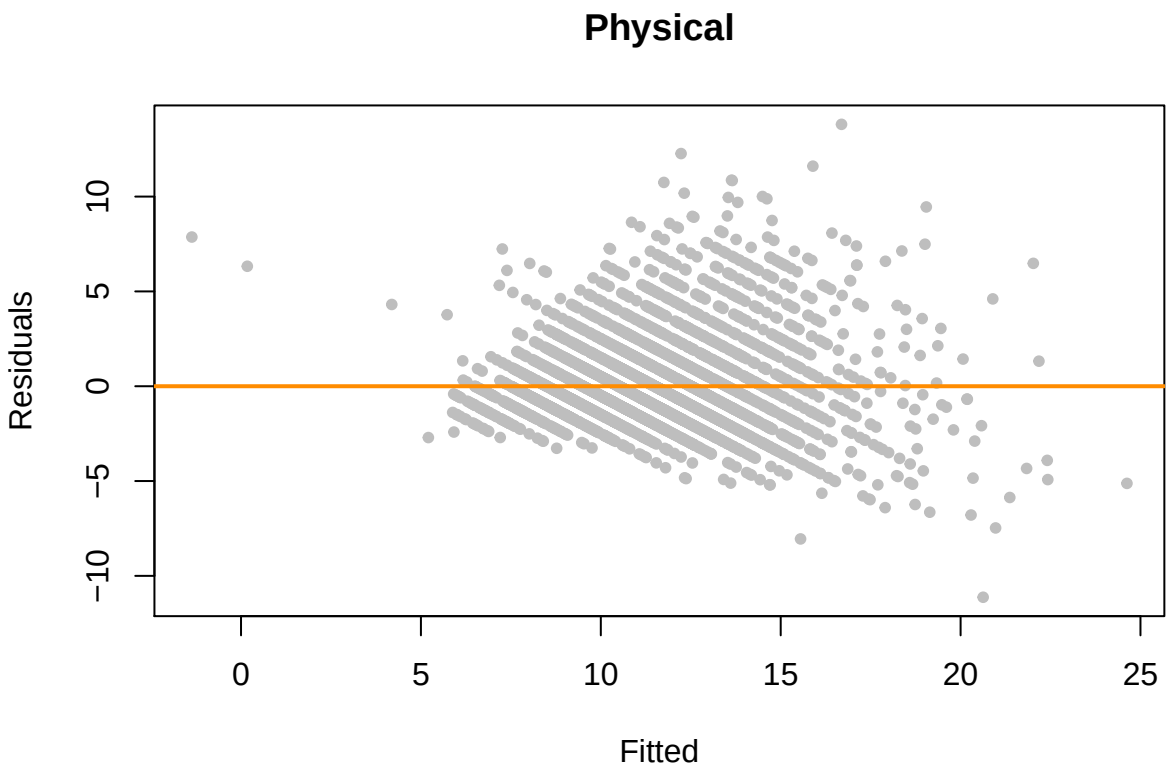



```
qqnorm(resid(abalone_model_1), col = "dodgerblue", pch = 20, cex = 1.5)
qqline(resid(abalone_model_1), col = "darkorange", lwd = 2)
```

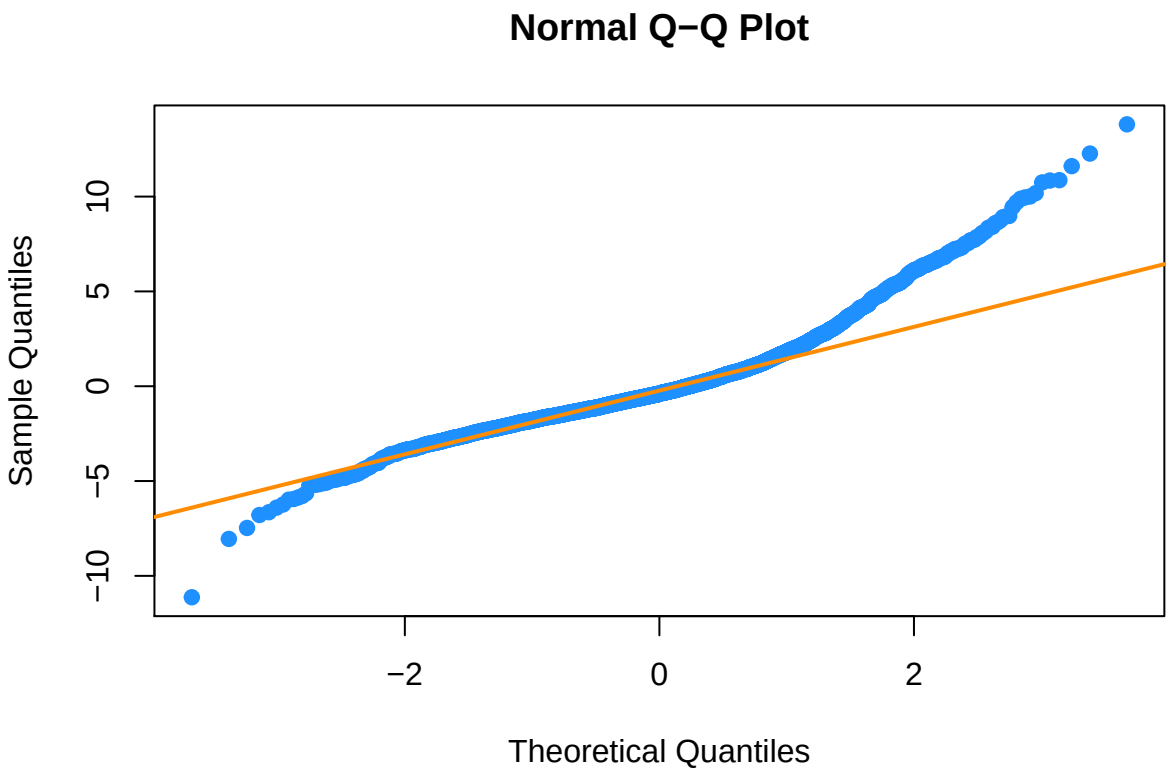


(b) Build a linear regression predicting the age from the measurements, including gender. There are three levels for gender; I'm not sure whether this has to do with abalone biology or difficulty in determining gender. You can represent gender numerically by choosing 1 for one level, 0 for another, and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

```
abalone_model_2 = lm(Age ~ ., data = abalone)

# fitted vs residuals
plot(
  fitted(abalone_model_2),
  resid(abalone_model_2),
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```

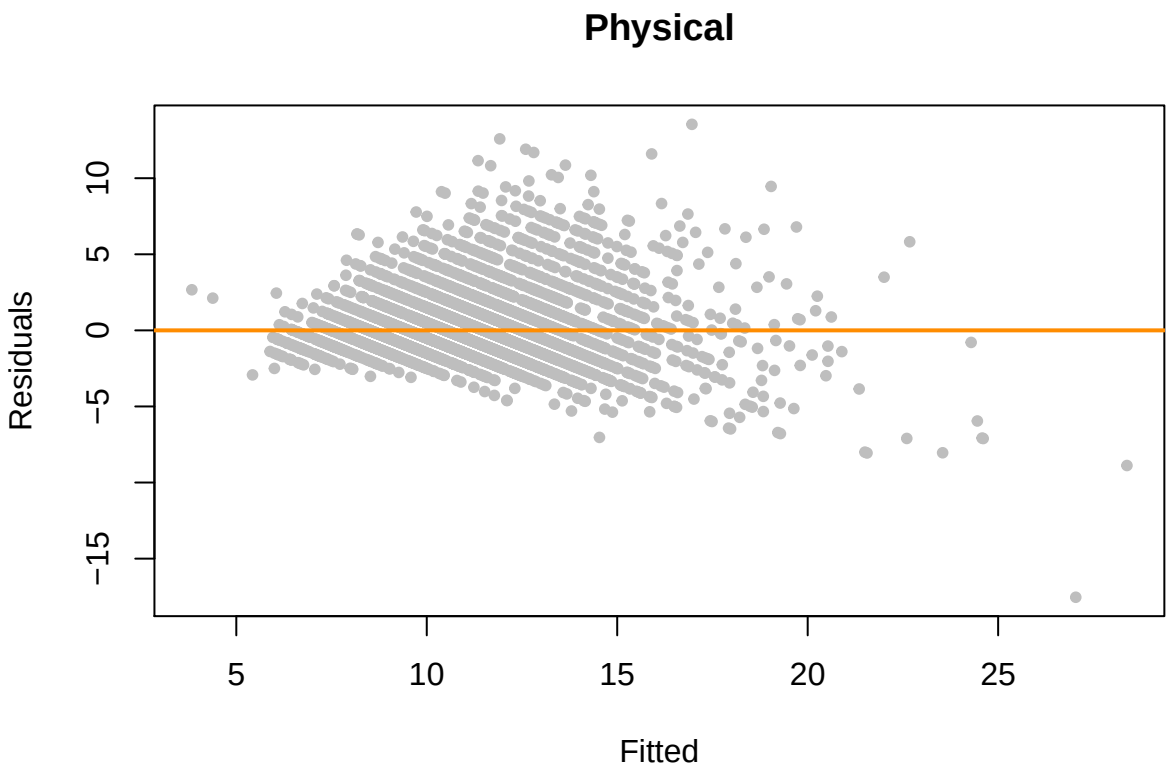



```
qqnorm(resid(abalone_model_2), col = "dodgerblue", pch = 20, cex = 1.5)
qqline(resid(abalone_model_2), col = "darkorange", lwd = 2)
```

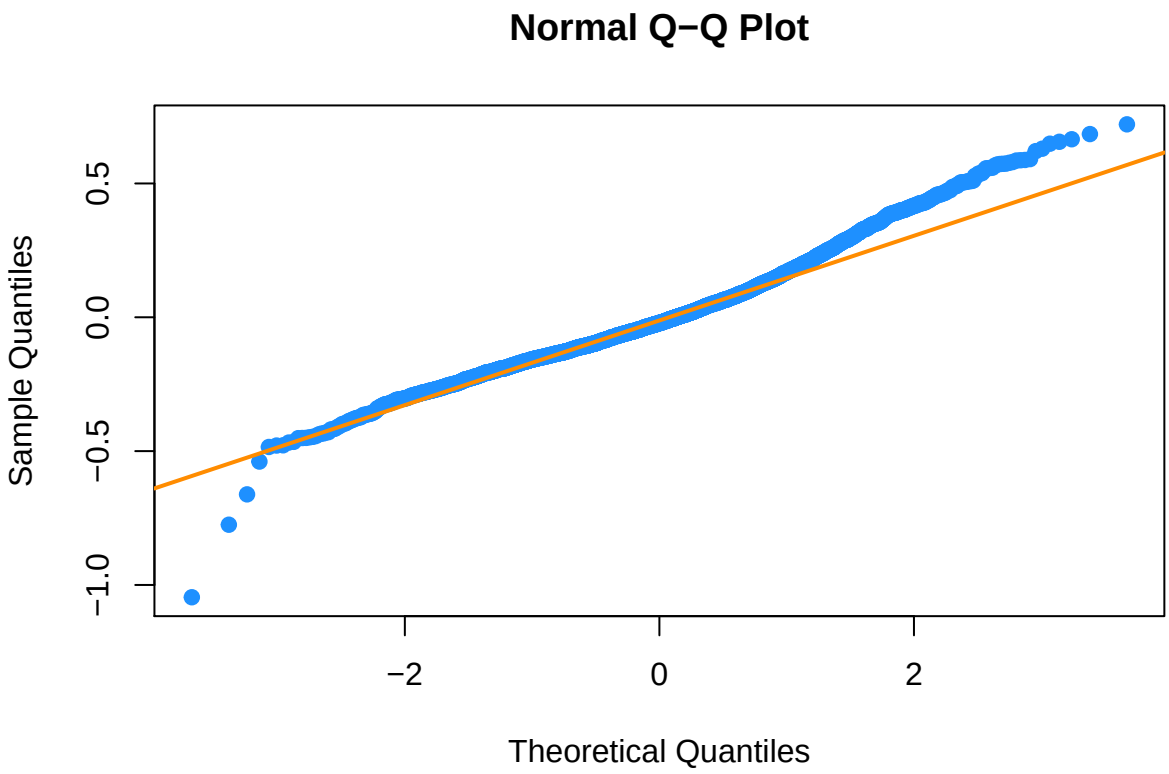


(c) Now build a linear regression predicting the log of age from the measurements,ignoring gender. Plot the residual against the fitted values.

```
abalone_model_3 = lm(log(Age) ~ . - Sex, data = abalone)

# fitted vs residuals
res_new_3 = abalone$Age - exp(fitted(abalone_model_3))

plot(
  exp(fitted(abalone_model_3)),
  res_new_3,
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```

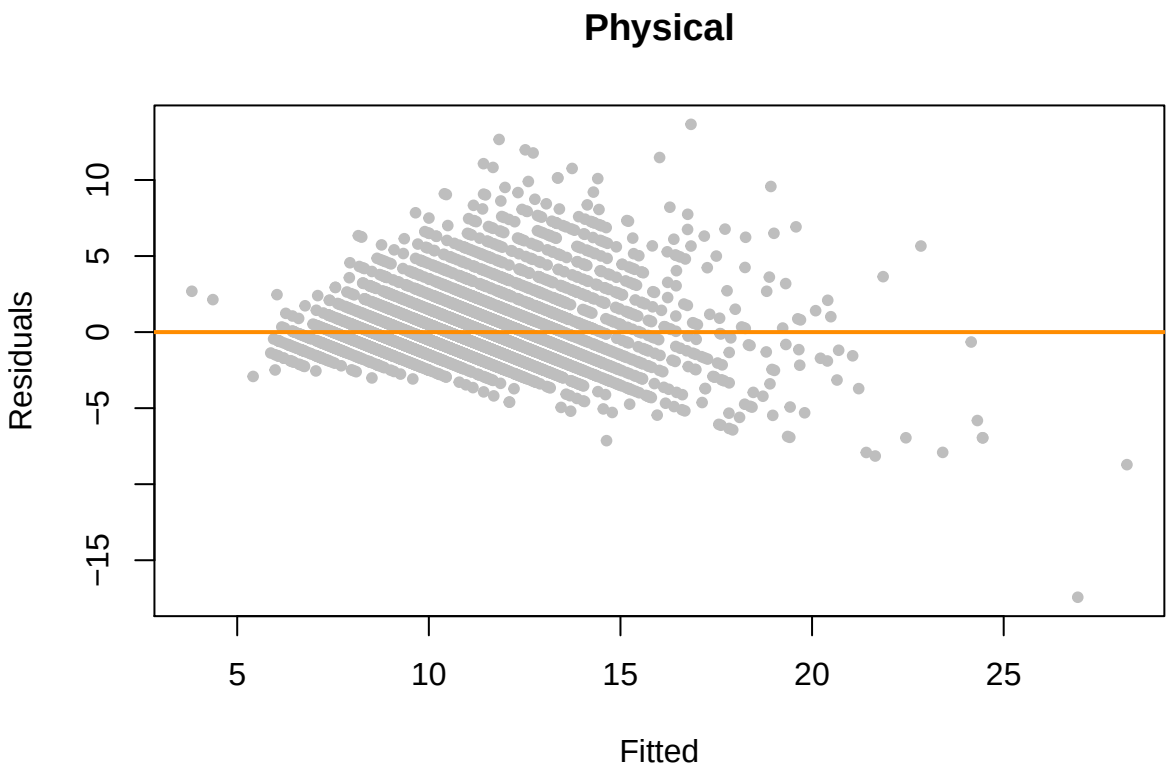



```
qqnorm(resid(abalone_model_3), col = "dodgerblue", pch = 20, cex = 1.5)
qqline(resid(abalone_model_3), col = "darkorange", lwd = 2)
```

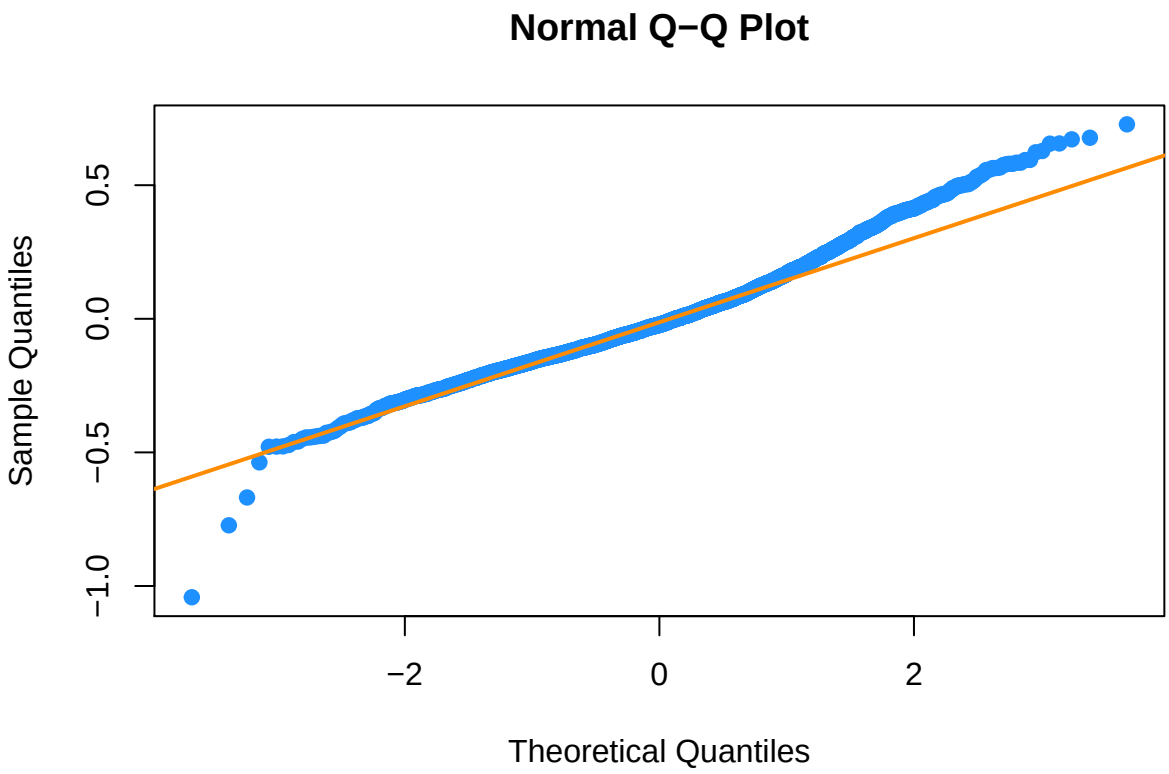


(d) Now build a linear regression predicting the log age from the measurements, including gender, represented as above. Plot the residual against the fitted values.

```
abalone_model_4 = lm(log(Age) ~ ., data = abalone)

res_new_4 = abalone$Age - exp(fitted(abalone_model_4))

plot(
  exp(fitted(abalone_model_4)),
  res_new_4,
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```




```
qqnorm(resid(abalone_model_4), col = "dodgerblue", pch = 20, cex = 1.5)
qqline(resid(abalone_model_4), 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.

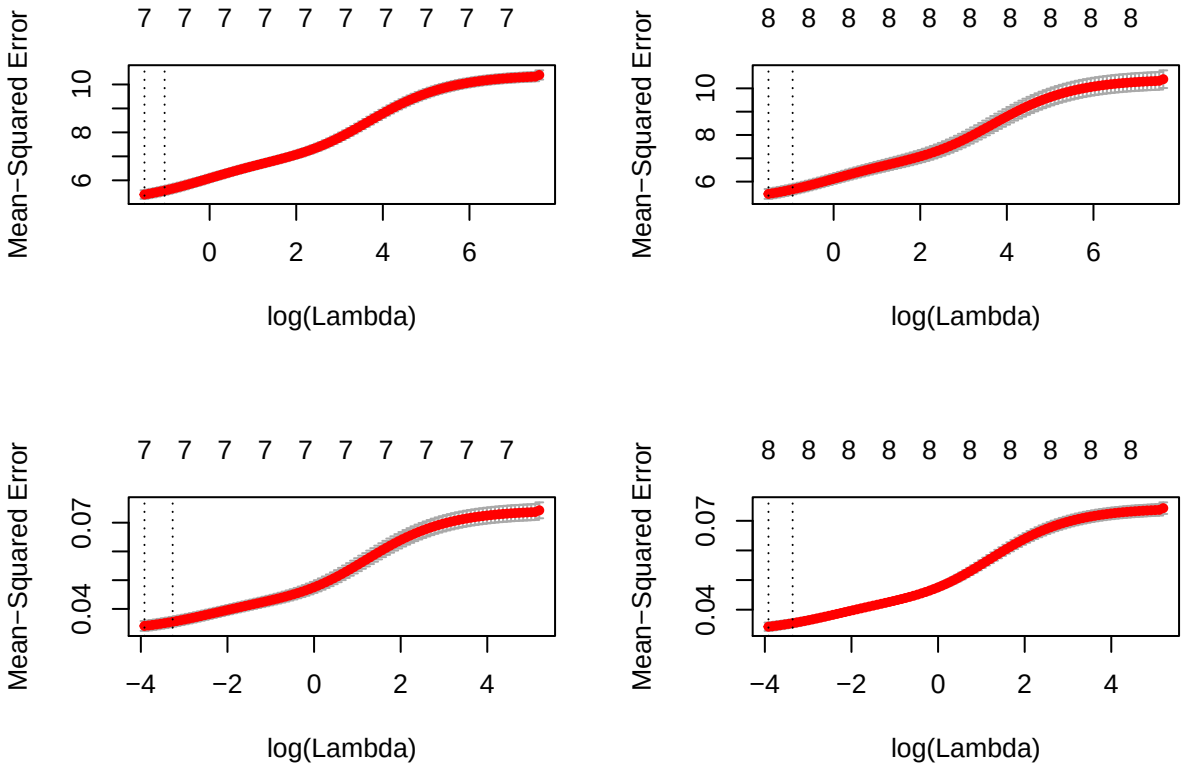
```
# R Squared values for all models
c(
  summary(abalone_model_1)$r.squared,
  summary(abalone_model_2)$r.squared,
  summary(abalone_model_3)$r.squared,
  summary(abalone_model_4)$r.squared
  )
```

```
## [1] 0.5276299 0.5278909 0.5796935 0.5801268
```

- The residual vs fitted plot looks similar for all models.
- The qq plot suggests better normality for the 3rd & 4th model than the rest.
- The R^2 is better for the 3rd & 4th model is better than the rest.
- From looking at the above plots (residual vs fitted & qq-plots) and the R^2 values, the 3rd & 4th model are quite comparable and better than the others. Since 3rd model has one less parameter we chose that.

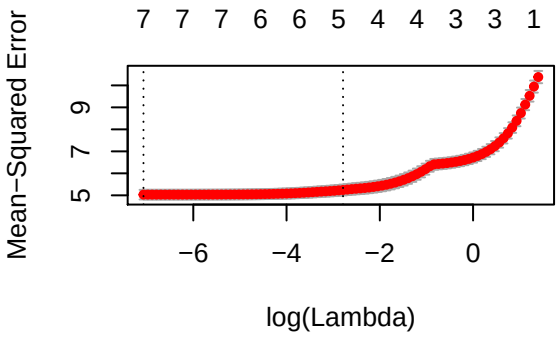
(f) Can you improve these regressions by using a regularizer? Use glmnet to obtain plots of the cross-validated prediction error.

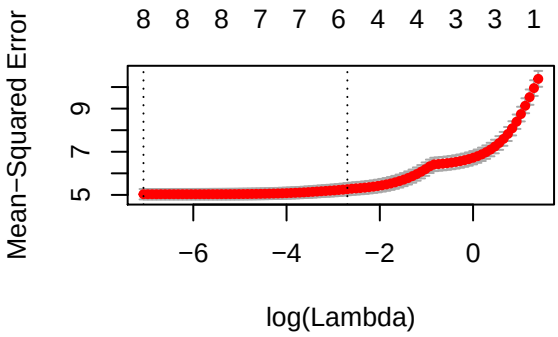
```
# Find cv_fit$cvm min for alpha = 0, 0.5 & 1 and then use the model with min value of cvm
# Use cv_fit$fit.preval to get the fitted value for the min lambda (# I am using cv.glmnet, when you se
# Transform fitted value in original scale
# Plot residual (original scale) vs fitted plot, rmse, r^2 and use that to say regularizatio help

cv_fita = cv.glmnet(as.matrix(abalone[, 2:(ncol(abalone) - 1)]), abalone$Age, alpha = 0, keep = TRUE)
cv_fitb = cv.glmnet(as.matrix(abalone[, 1:(ncol(abalone) - 1)]), abalone$Age, alpha = 0, keep = TRUE)
cv_fitc = cv.glmnet(as.matrix(abalone[, 2:(ncol(abalone) - 1)]), log(abalone$Age), alpha = 0, keep = TRU
cv_fitd = cv.glmnet(as.matrix(abalone[, 1:(ncol(abalone) - 1)]), log(abalone$Age), alpha = 0, keep = TRU

par(mfrow = c(2, 2))
plot(cv_fita)
plot(cv_fitb)
plot(cv_fitc)
plot(cv_fitd)
```

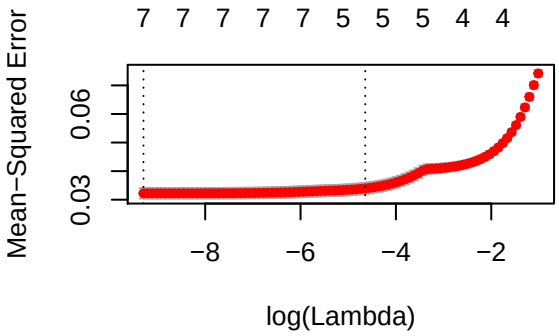


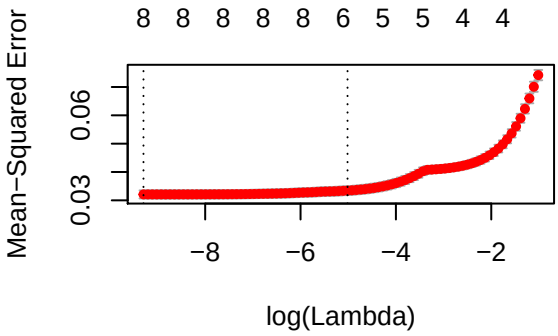

```
cv_fita1 = cv.glmnet(as.matrix(abalone[, 2:(ncol(abalone) - 1)]), abalone$Age, alpha = 0.5, keep = TRUE)
cv_fitb1 = cv.glmnet(as.matrix(abalone[, 1:(ncol(abalone) - 1)]), abalone$Age, alpha = 0.5, keep = TRUE)
cv_fitc1 = cv.glmnet(as.matrix(abalone[, 2:(ncol(abalone) - 1)]), log(abalone$Age), alpha = 0.5, keep =
cv_fitd1 = cv.glmnet(as.matrix(abalone[, 1:(ncol(abalone) - 1)]), log(abalone$Age), alpha = 0.5, keep =

par(mfrow = c(2, 2))
plot(cv_fita1)
```

```
plot(cv_fitb1)
plot(cv_fitc1)
plot(cv_fitd1)
```

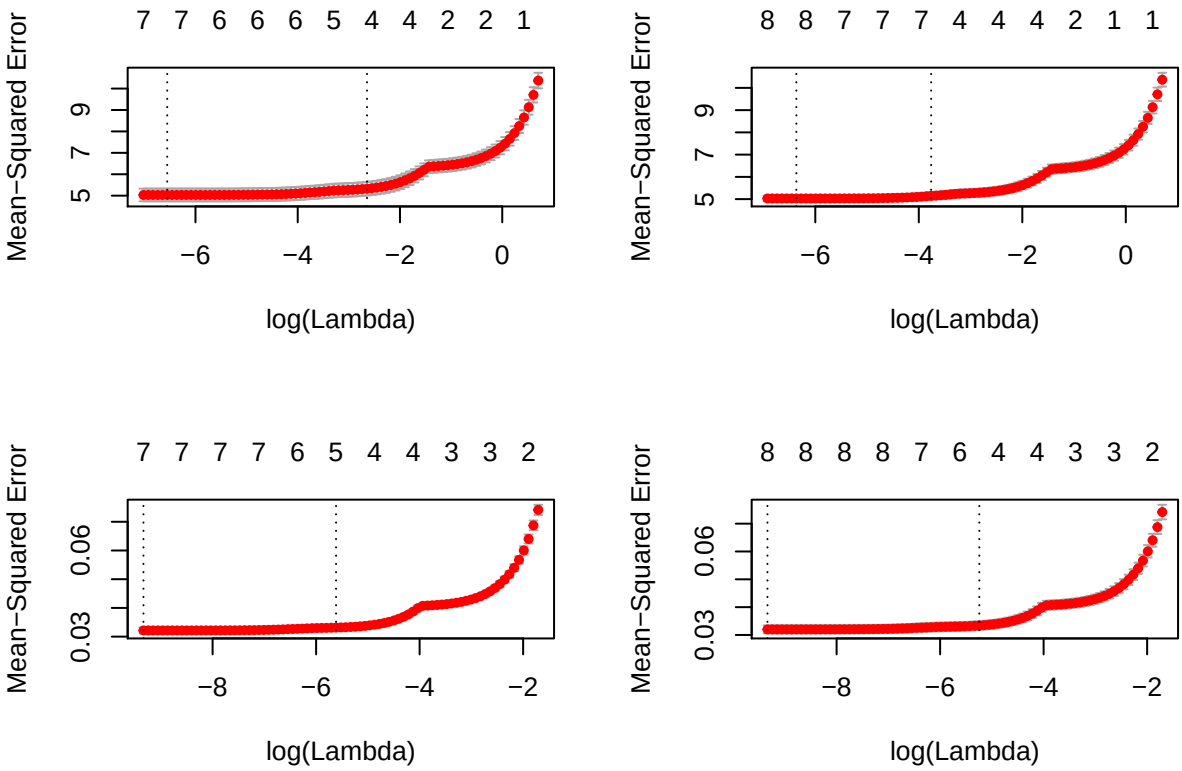









```
cv_fita2 = cv.glmnet(as.matrix(abalone[, 2:(ncol(abalone) - 1)]), abalone$Age, alpha = 1, keep = TRUE)
cv_fitb2 = cv.glmnet(as.matrix(abalone[, 1:(ncol(abalone) - 1)]), abalone$Age, alpha = 1, keep = TRUE)
cv_fitc2 = cv.glmnet(as.matrix(abalone[, 2:(ncol(abalone) - 1)]), log(abalone$Age), alpha = 1, keep = TR
cv_fitd2 = cv.glmnet(as.matrix(abalone[, 1:(ncol(abalone) - 1)]), log(abalone$Age), alpha = 1, keep = TR

par(mfrow = c(2, 2))
plot(cv_fita2)
plot(cv_fitb2)
plot(cv_fitc2)
plot(cv_fitd2)
```



Cross-Validated predicted error for different alphas on all models

	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
Model #1	5.4072665	5.0319142	5.0329787
Model #2	5.4712102	5.0318104	5.0306941
Model #3	0.0340739	0.0322391	0.0321621
Model #4	0.034259	0.0320884	0.0320044

- From the above table its evident that the cross validated predicted error for model #3 & model # 4 are lowest & comparable for values of *alpha = 0.5 & 1*. We chose *model 3 since* since its the simplest with *alpha = 1*

Minimum lambdas & Best R^2 for different alphas on all models

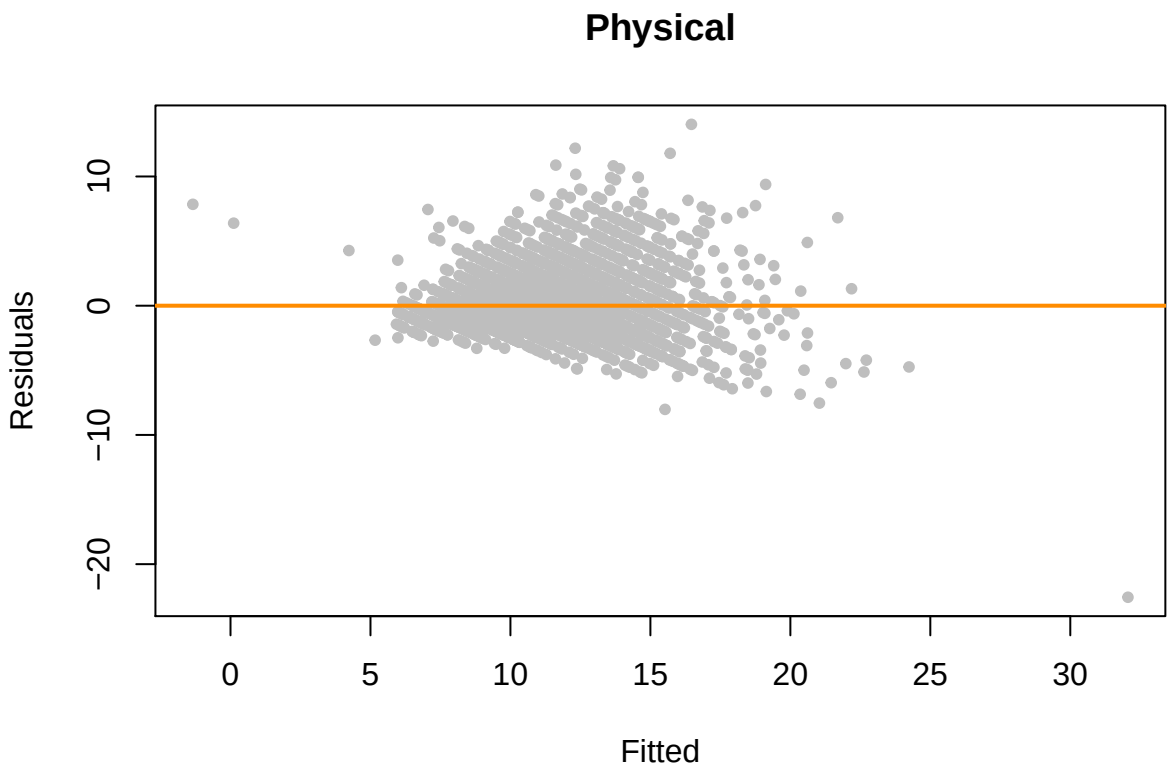
```
fit_modela2 = cv_fita2$fit.preval[, which.min(cv_fita2$lambda)]
fit_modelb2 = cv_fitb2$fit.preval[, which.min(cv_fitb2$lambda)]
fit_modelc2 = exp(cv_fitc2$fit.preval[, which.min(cv_fitc2$lambda)])
fit_modeld2 = exp(cv_fitd2$fit.preval[, which.min(cv_fitd2$lambda)])

res_orgscalea2 = abalone$Age - fit_modela2
res_orgscaleb2 = abalone$Age - fit_modelb2
res_orgscalec2 = abalone$Age - fit_modelc2
```

```
res_orgscaled2 = abalone$Age - fit_modeld2

r2a2 = 1 - min(cv_fita2$cvm) / var(abalone$Age)
r2b2 = 1 - min(cv_fitb2$cvm) / var(abalone$Age)
r2c2 = 1 - min(cv_fitc2$cvm) / var(log(abalone$Age))
r2d2 = 1 - min(cv_fitd2$cvm) / var(log(abalone$Age))
```

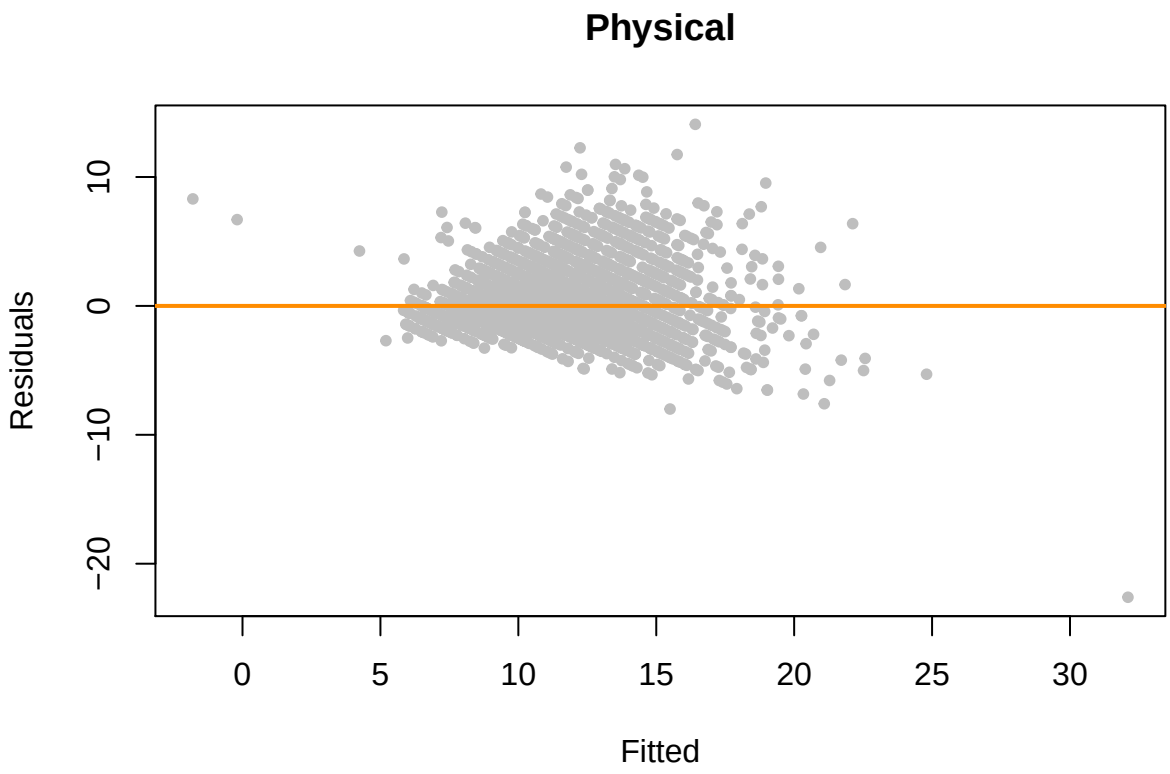
	min λ for α=1	R^2
Model #1	0.0014273	0.5158394
Model #2	0.0017192	0.5160591
Model #3	0.0000882	0.5671298
Model #4	0.0000882	0.5692523

- The regularization value for *model 3* is very low
- The fitted vs residual curve for *model 3* with regularization has some outliers as compared to *model 3* without regularization
- The r.squared value for *model 3* with regularization 0.5671298 is silghtly less to *model 3* without regularization 0.5796935.
- Hence the regularization is not helping much here

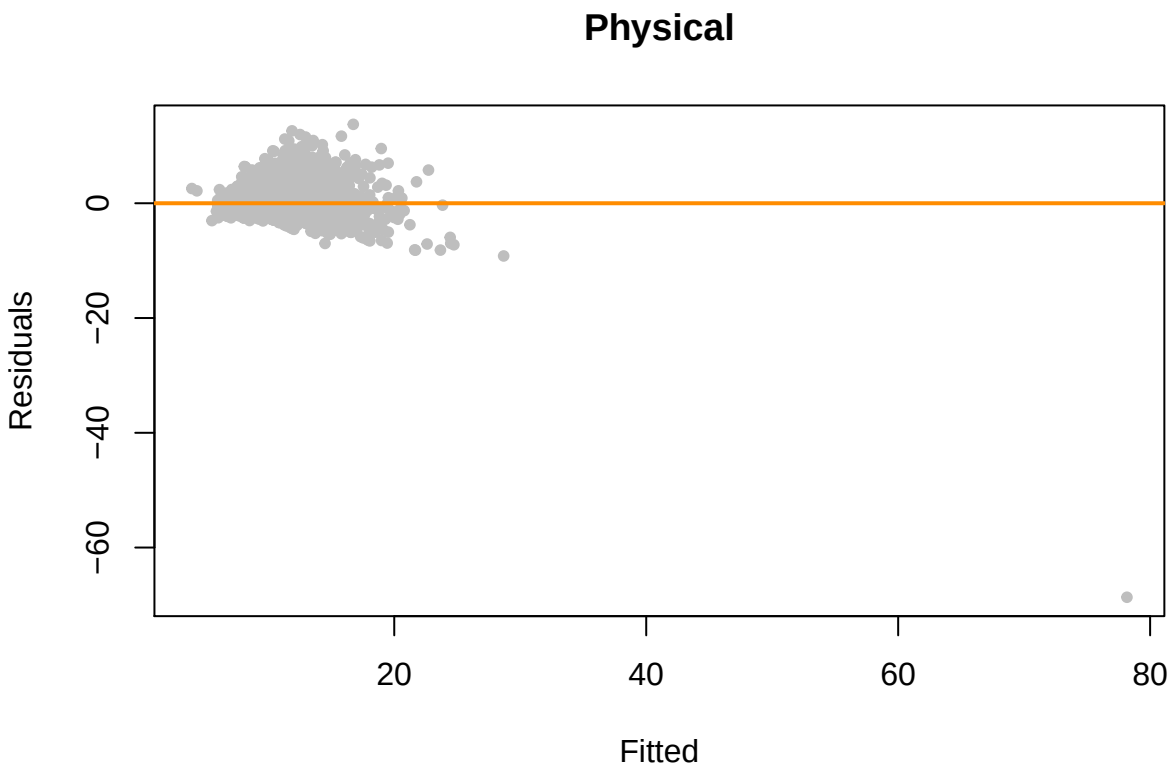
```
plot(
  fit_modela2,
  res_orgscalea2,
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```



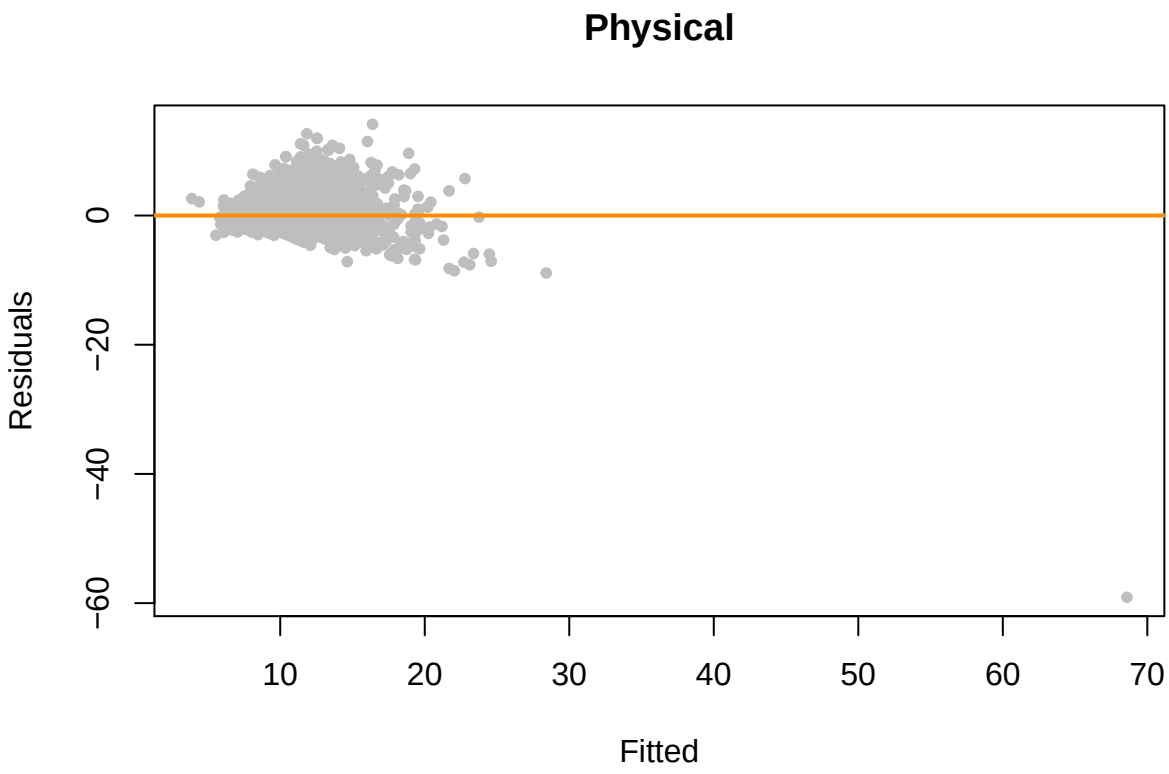

```
plot(
  fit_modelb2,
  res_orgscaleb2,
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```




```
plot(
  fit_modelc2,
  res_orgscalec2,
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```




```
plot(
  fit_modeld2,
  res_orgscaled2,
  col = "grey",
  pch = 20,
  main = "Physical",
  xlab = "Fitted",
  ylab = "Residuals"
)
abline(h = 0, col = "darkorange", lwd = 2)
```


Physical
Residuals
0
−20
−40
−60
10
20
30
40
50
60
70
Fitted