Problem Set 3

Math 430, Winter 2017

Problem 1: Complete Chapter 3 problem #8 from the textbook.

Note: Multiple answers were accepted here, especially for transformations. What I present below is simply one or two possible outcomes.

Load the data files directly from the book's website:

```
diamond <- read.table("http://www.stat.tamu.edu/~sheather/book/docs/datasets/diamonds.txt", header = TRU
str(diamond)
```

```
## 'data.frame':    49 obs. of  2 variables:
##  $ Size : num  0.17 0.16 0.17 0.18 0.25 0.16 0.15 0.19 0.21 0.15 ...
##  $ Price: int  355 328 350 325 642 342 322 485 483 323 ...
```

Part 1

(a)

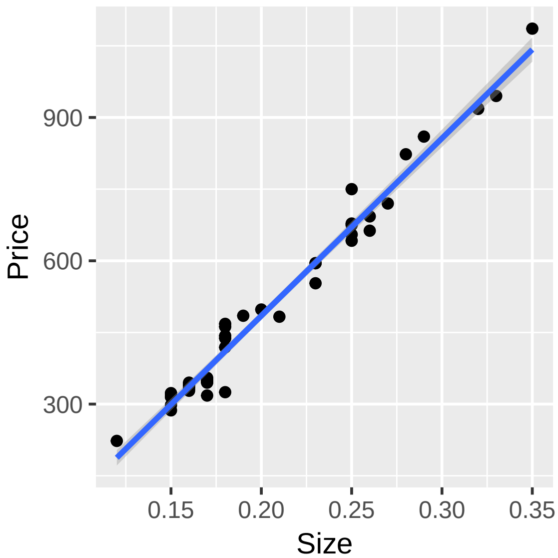
Since we wish to predict price from size, we will use size as the predictor and price as the response. In R, we fit this model using the below code:

```
diamond_lm <- lm(Price ~ Size, data = diamond)
summary(diamond_lm)
```

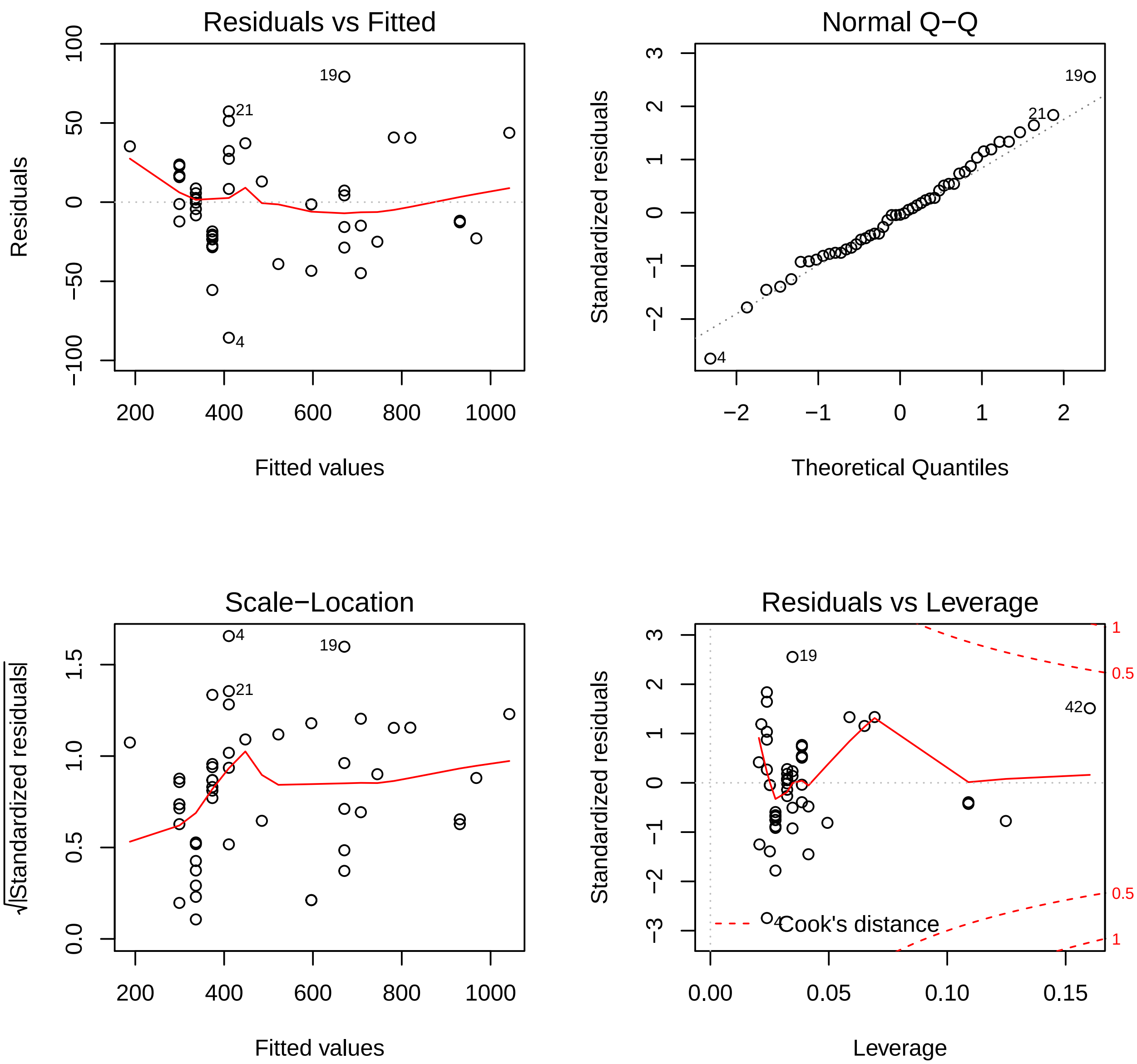
```
##
## Call:
## lm(formula = Price ~ Size, data = diamond)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -85.654 -21.503  -1.203  16.797  79.295
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -258.05      16.94  -15.23   <2e-16 ***
## Size         3715.02      80.41   46.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.6 on 47 degrees of freedom
## Multiple R-squared:  0.9785, Adjusted R-squared:  0.978
## F-statistic:  2135 on 1 and 47 DF,  p-value: < 2.2e-16
```

(b)

```
ggplot(data = diamond, aes(x = Size, y = Price)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Price
900
600
300
0.15
0.20
0.25
0.30
0.35
Size



Potential weaknesses of this model include:

- There may be slight curvature in the residual plot. This puts linearity into question.
- The scale-location plot exhibits a positive association, indicating potentially increasing residual variance as for larger fitted values.

Part 2

(a)

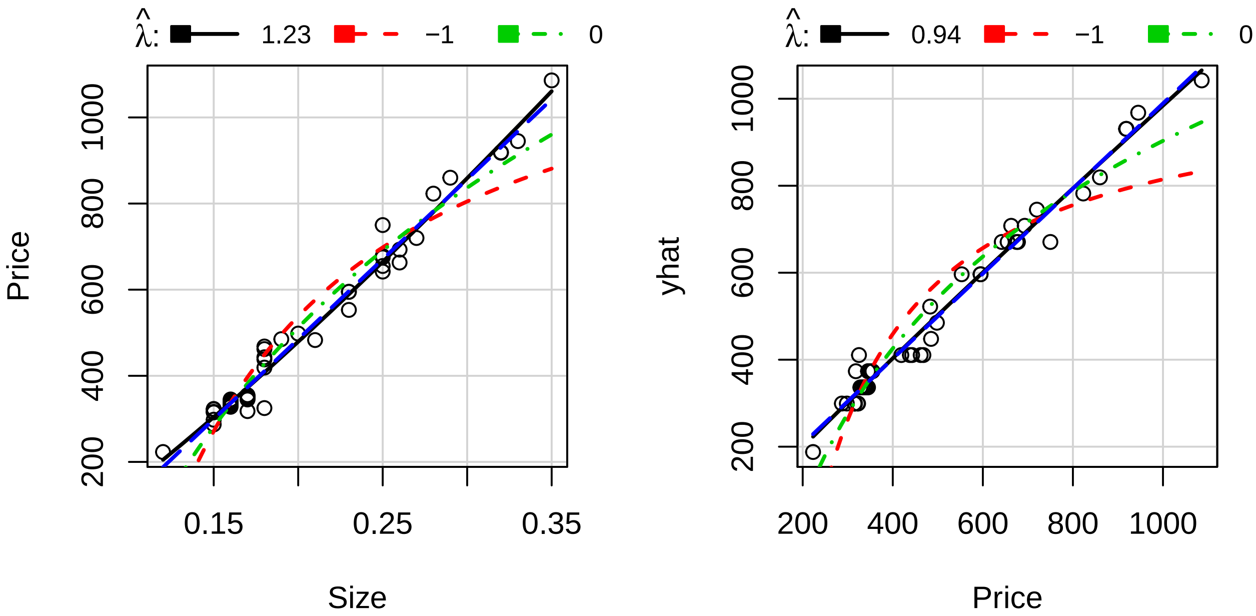
Using the graphical approach to transformations we find:

```r
library(car)
par(mfrow = c(1,2))
invTranPlot(Price ~ Size, data = diamond)
```

```
##      lambda       RSS
```

```
## 1  1.229014  45244.55
## 2 -1.000000 207151.70
## 3  0.000000  94429.08
## 4  1.000000  46929.15
```

```r
invResPlot(diamond_lm)
```




```
##       lambda       RSS
## 1  0.9376257  45670.12
## 2 -1.0000000 272143.61
## 3  0.0000000 101071.53
## 4  1.0000000  45918.17
```

Using inverse transformation and response plots there is little evidence of a need to transform either variable. In both situations, possible transformations to improve the model are close to $\lambda = 1$.

If you utilize the Box-Cox approach to explore transformations, then you ued code similar to the below:

The Box-Cox approach leads to the following model

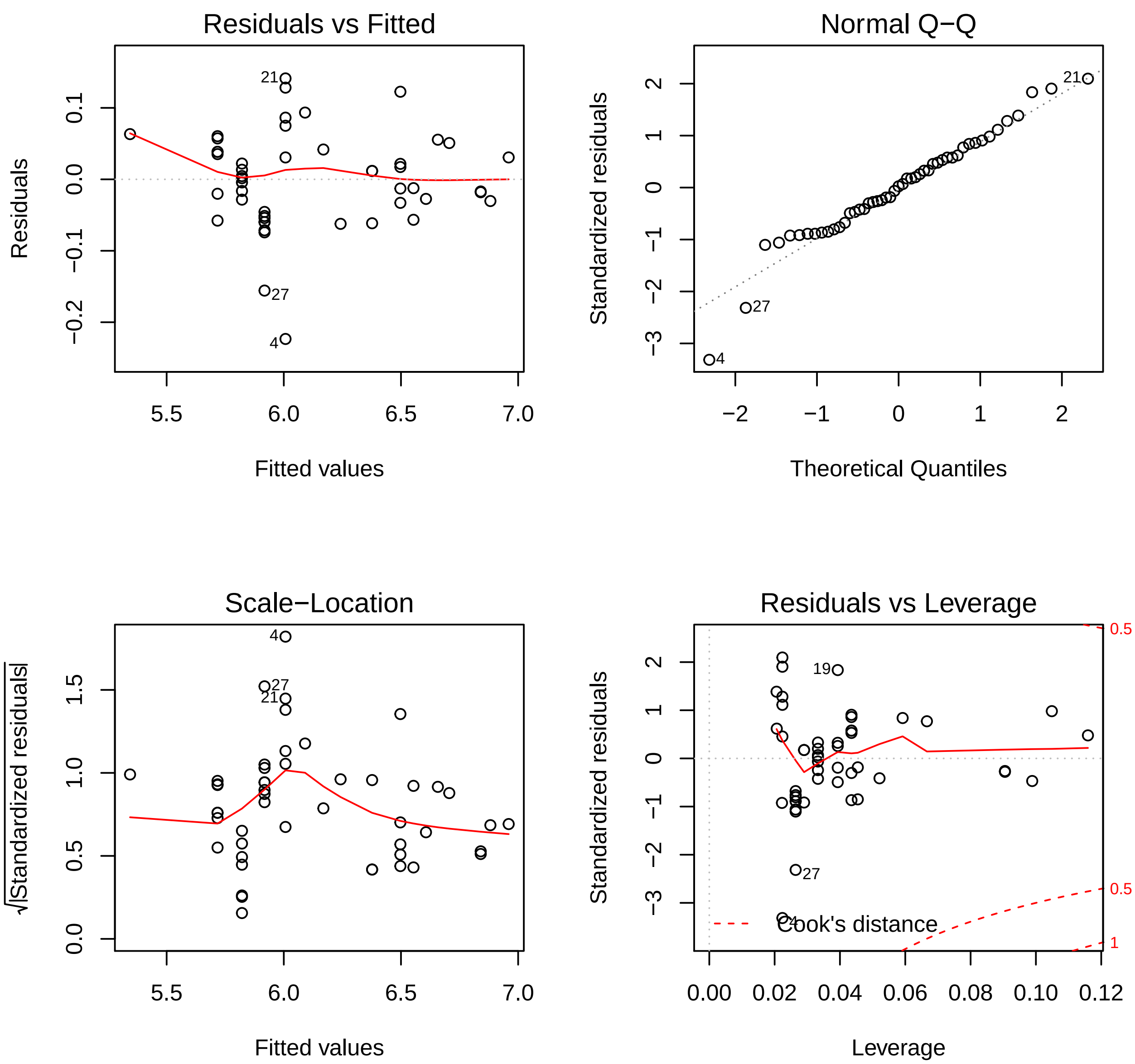
```r
bc_mod <- lm(log(Price) ~ I(Size^-0.25), data = diamond)
summary(bc_mod)
```

```
##
## Call:
## lm(formula = log(Price) ~ I(Size^-0.25), data = diamond)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.223411 -0.045628  0.001625  0.038482  0.141232
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    12.2252     0.1546   79.09   <2e-16 ***
## I(Size^-0.25)  -4.0501     0.1025  -39.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.06816 on 47 degrees of freedom
## Multiple R-squared:  0.9708, Adjusted R-squared:  0.9702
## F-statistic:  1563 on 1 and 47 DF,  p-value: < 2.2e-16
```

(b)

Assuming that you transformed the model, you would produce a new set of residual plots to investigate potential violations of the SLR assumptions.



There is little to pick on in these residual plots. One potential weaknesses of this model includes:

- Observation 4 is a somewhat large outlier that may need to be investigated, however it's Cook's distance value does not cause any alarm.

Part 3

If we are simply interested in explaining the relationship between the two variables using a linear equation, then we would pick the untransformed model, as it has a slightly higher R^2 value:

```
library(broom)
glance(diamond_lm)
```

```
##   r.squared adj.r.squared    sigma statistic      p.value df    logLik
## 1 0.9784573      0.977999 31.59893  2134.716 7.955338e-41  2 -237.7101
##        AIC      BIC deviance df.residual
## 1 481.4201 487.0956 46929.15          47
```

```
glance(bc_mod)
```

```
##   r.squared adj.r.squared      sigma statistic      p.value df   logLik
## 1 0.9708011     0.9701799 0.06815617  1562.651 1.013575e-37  2 63.10472
##         AIC      BIC  deviance df.residual
## 1 -120.2094 -114.534 0.2183274          47
```

If we are more interested in predicting the price of a diamond from it's size, then we should consider using cross validation to assess the predictive accuracy of the models.

```
# Split data into training and test sets
index <-  sample(1:nrow(diamond), size = 0.2 * nrow(diamond))
train_data <- diamond[-index,]
test_data <- diamond[index,]
```

```
### Predictive accuracy for diamond_lm
# Fit model to training data
train_diamond_lm <- lm(Price ~ Size, data = train_data)

# Obtain predictions
preds_diamond_lm <- predict(train_diamond_lm, newdata = test_data)

# Calculate metrics
bias_diamond_lm <- mean(test_data$Price - preds_diamond_lm)
pmse_diamond_lm <- mean((test_data$Price - preds_diamond_lm)^2)
rpmse_diamond_lm <- sqrt(pmse_diamond_lm)

# Print the results
c(Bias = bias_diamond_lm, RPMSE = rpmse_diamond_lm)
```

```
##      Bias     RPMSE
##  2.552459 43.403930
```

```
### Predictive accuracy for bc_mod

# Fit model to training data
train_bc_mod <- lm(log(Price) ~ I(Size^-0.25), data = train_data)

# Obtain predictions
preds_bc_mod <- predict(train_bc_mod, newdata = test_data)
preds_orig_bc_mod <- exp(preds_bc_mod)

# Calculate metrics
bias_bc_mod <- mean(test_data$Price - preds_orig_bc_mod)
pmse_bc_mod <- mean((test_data$Price - preds_orig_bc_mod)^2)
rpmse_bc_mod <- sqrt(pmse_bc_mod)
```

```r
# Print the results
c(Bias = bias_bc_mod, RPMSE = rpmse_bc_mod)
```

```
##      Bias     RPMSE
##  2.452201 40.770399
```

The models are quite close in terms of predictive accuarcy, but the transformed model has the slight edge (smaller biar and RPMSE).

Problem 2

The data in the file `baeskel.csv` were collected in the study of the effect of dissolved sulfur on the surface tension of liquid copper (Baes and Kellogg, 1953). The predictor `Sulfur` is the weight percent sulfur, and the response is `Tension`, the decrease in surface tension in dynes per centimeter. Two replicate observations were taken at each value of `Sulfur`. These data were previously discussed by Sclove (1968).

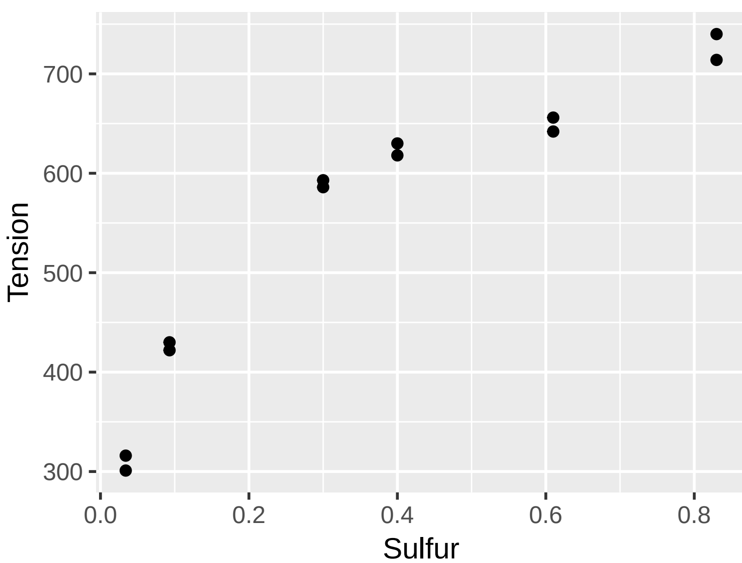
```r
baeskel <- read.csv("https://raw.githubusercontent.com/math430-lu/data/master/baeskel.csv")
```

a.

Create a plot of `Tension` vs. `Sulfur` to verify that a transformation is required to achieve a linear mean function.

From the below plot, it is obvious that there is a nonlinear relationship between tension and sulfur, so a transformation is needed to use simple linear regression.

```r
ggplot(data = baeskel, aes(x = Sulfur, y = Tension)) +
  geom_point()
```



b.

Set $\lambda = -1$, and fit the mean function

$$E(Tension|Sulfur) = \beta_0 + \beta_1 Sulfur^{\lambda}$$

using the `lm` function; that is, fit the regression model with `Tension` as the response and 1/`Sulfur` as the regressor. Add a line for the fitted values from this fit to the plot you created in part a. Repeat for $\lambda = 0, 1$, and so in the end you will have three lines on your plot. Which of these three choices of λ gives the fitted values that match the data most closely?

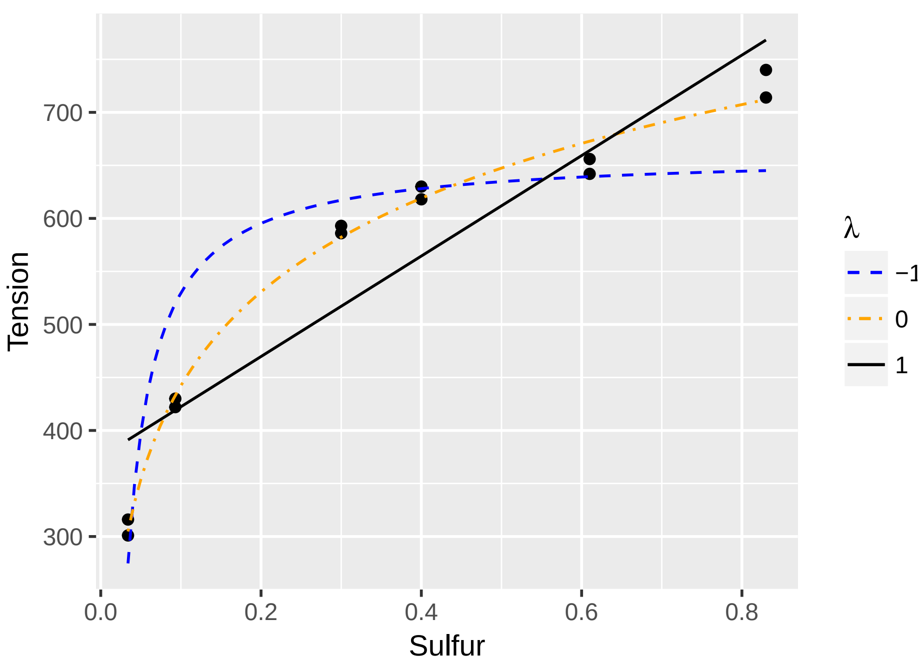
```r
# Fitting the three regression models
invx <- lm(Tension ~ I(Sulfur^-1), data = baeskel)
logx <- lm(Tension ~ log(Sulfur), data = baeskel)
rawx <- lm(Tension ~ Sulfur, data = baeskel)

# Load broom
library(broom)

# Set up data frame for predictions
newdata <- data.frame(Sulfur = seq(min(baeskel$Sulfur), max(baeskel$Sulfur), length.out = 100))

# Get a data frame with x and y coordinates for each line
invx_data <- augment(invx, newdata = newdata)
logx_data <- augment(logx, newdata = newdata)
rawx_data <- augment(rawx, newdata = newdata)

# Superimpose the lines on the above plot
ggplot(data = baeskel, aes(x = Sulfur, y = Tension)) +
  geom_point() +
  geom_line(data = invx_data, aes(x = Sulfur, y = .fitted, color = "-1", linetype = "-1")) +
  geom_line(data = logx_data, aes(x = Sulfur, y = .fitted, color = "0", linetype = "0")) +
  geom_line(data = rawx_data, aes(x = Sulfur, y = .fitted, color = "1", linetype = "1")) +
  scale_colour_manual(name = expression(lambda), values = c("blue", "orange", "black")) +
  scale_linetype_manual(name = expression(lambda), values = c(2, 4, 1))
```



From the above plot it is apparent that a log transformation of sulfur (i.e. $\lambda = 0$) gives the fitted values that match the data most closely?

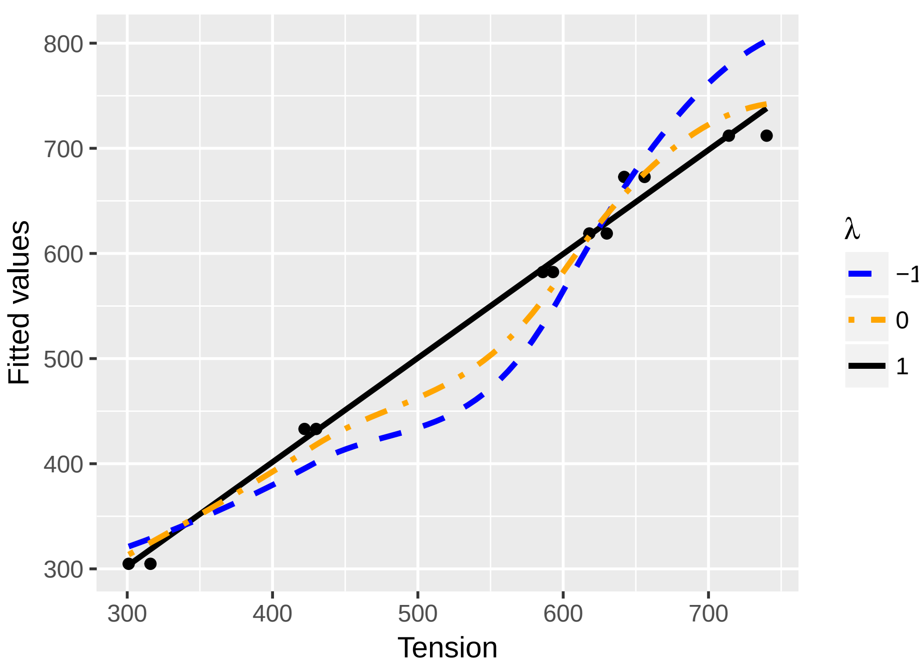
c.

Replace `Sulfur` by `log(Sulfur)`, and consider transforming the response `Tension`. To do this, draw the inverse fitted value plot with the fitted values from the regression `Tension ~ log(Sulfur)` on the vertical axis and `Tension` on the horizontal axis. Repeat the methodology from part b. to decide if further transformation of the response will be helpful.

```
# The the models
invy_logx <- lm(I(Tension^-1) ~ log(Sulfur), data = baeskel)
logy_logx <- lm(log(Tension) ~ log(Sulfur), data = baeskel)

# Augment the data with output from the models
y_logx_data <- augment(logx)
invy_logx_data <- augment(invy_logx)
logy_logx_data <- augment(logy_logx)

# Create the plots
ggplot(data = y_logx_data, aes(x = Tension, y =  .fitted)) +
  geom_point() +
  geom_smooth(aes(color = "1", linetype = "1"), method = "lm", se = FALSE) +
  stat_smooth(data = invy_logx_data, aes(x = baeskel$Tension, y = 1/.fitted, color = "-1", linetype = "-
  geom_smooth(data = logy_logx_data, aes(x = baeskel$Tension, y = exp(.fitted), color = "0", linetype =
  scale_colour_manual(name = expression(lambda), values = c("blue", "orange", "black")) +
  scale_linetype_manual(name = expression(lambda), values = c(2, 4, 1))+
  labs(y = "Fitted values")
```



From the plot, it is apparent that no further transformation is necessary.