Transformations

Math 430, Winter 2017

What do we do if our assumptions are violated?

1. Change your assumptions (hard, need more stats)
2. Transform y, x, or both

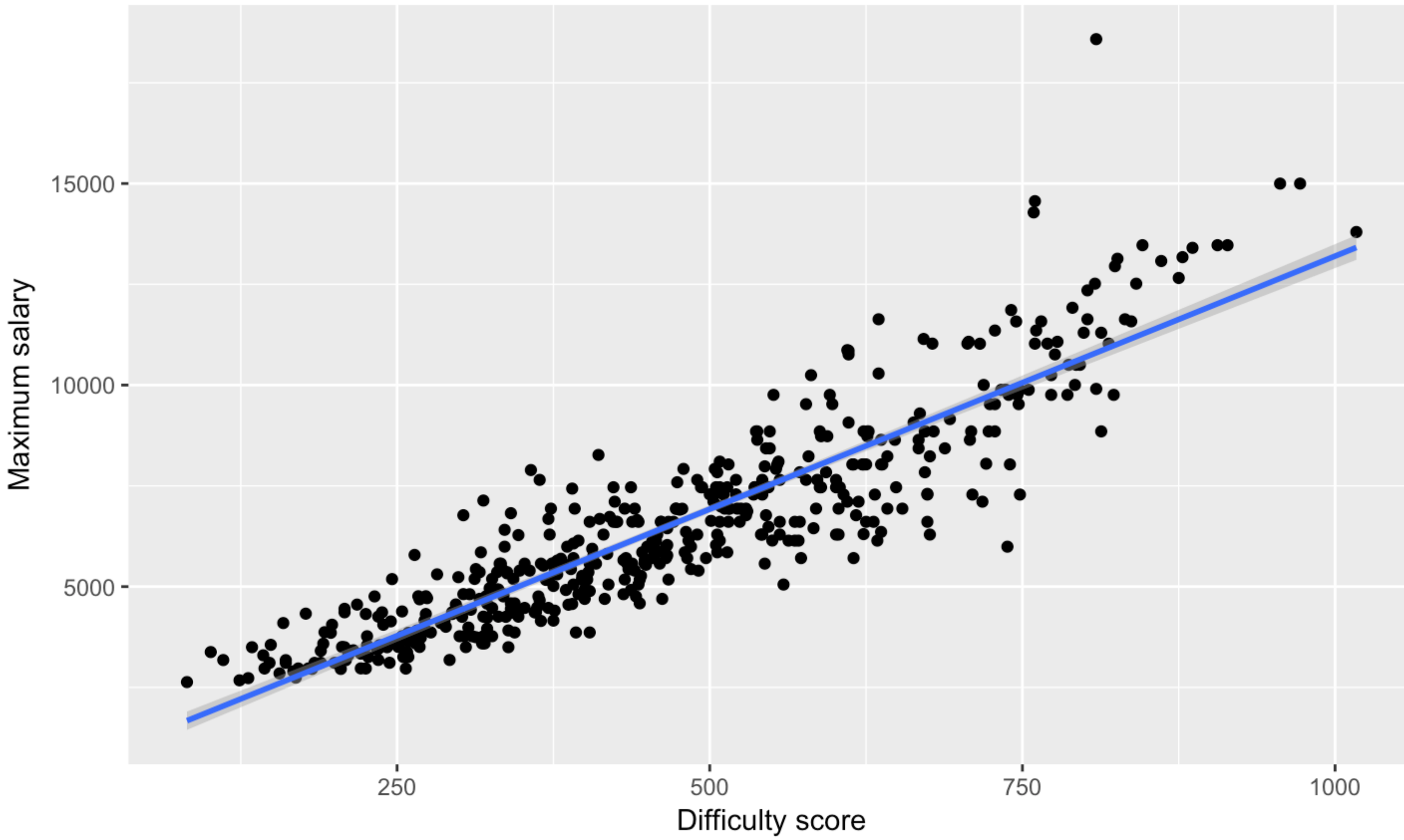
Transformations can…

- address non-linear patterns (i.e. linear on transformed scale)
- stabilize variance
- correct skew
- minimize the effects of outliers
- estimate percentage effects

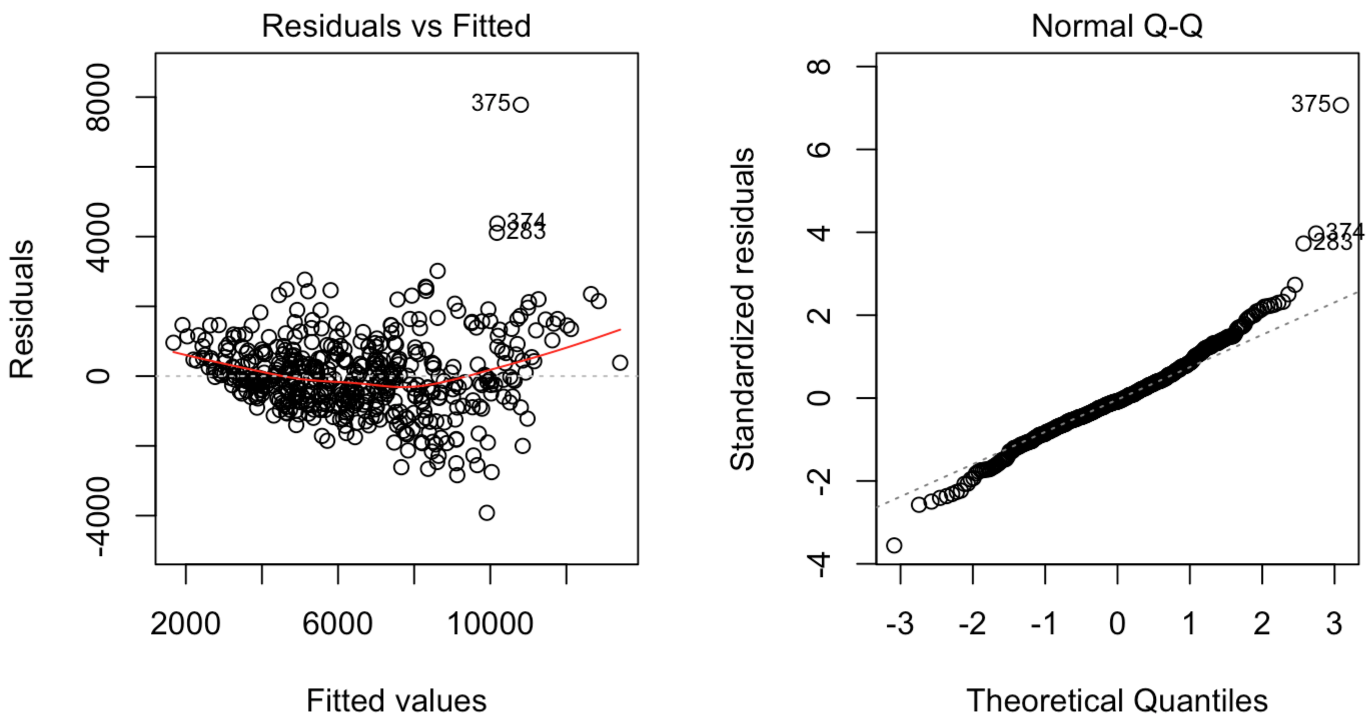
Transforming the response

Example

How is job difficulty related to salary?

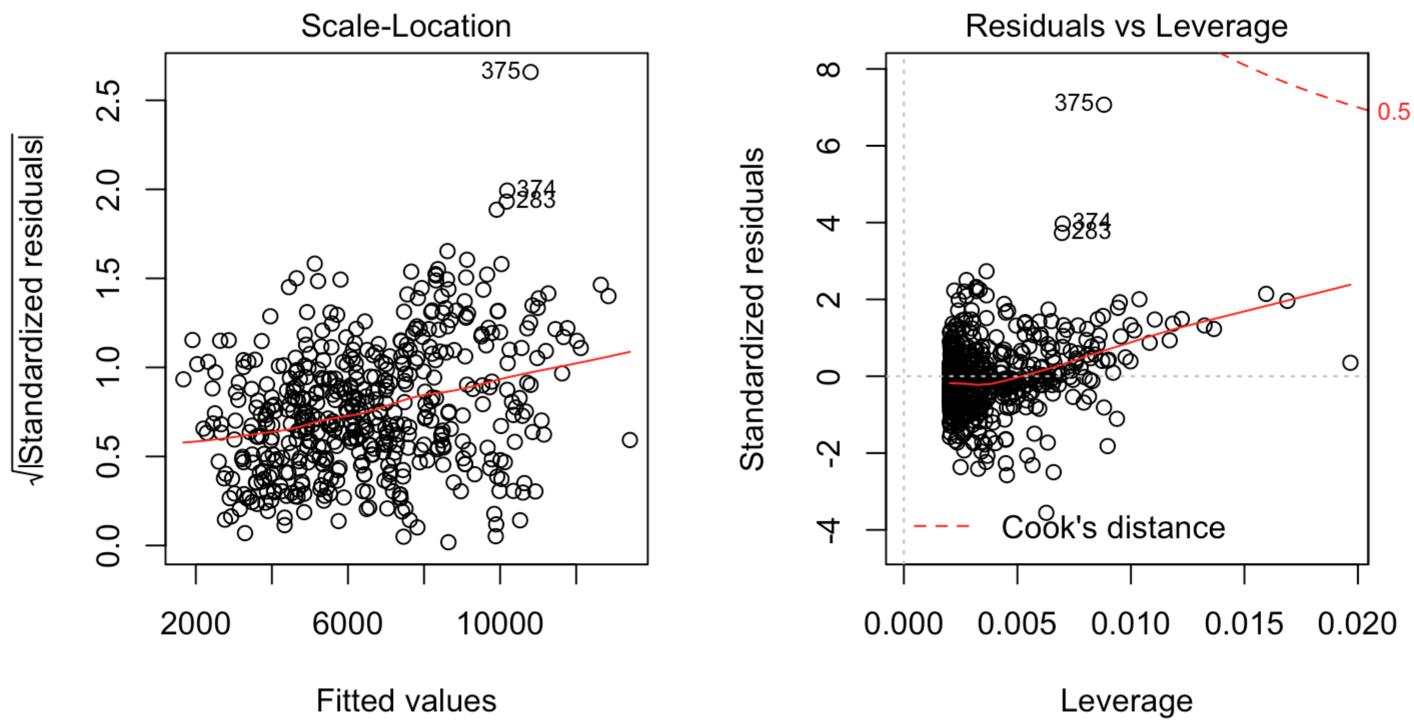


Linearity and normality



Mean function may be nonlinear. Residuals have slightly heavy-tailed distribution.

Constant variance and influence



Variance seems to be increasing. No troubling influential values.

Power transformations

Let, U be a strictly positive variable, then the *power family of transformations* is defined by

$$\psi(U, \lambda) = U^{\lambda}$$

- Try values in the range [-1, 1] and see how they help problems, {-1, -1/2, 0, 1/3, 1/2, 1} is recommended

- Sometimes expansions to the range [-2, 2] is necessary

Rules of thumb

- **log rule:** if values range over more than 1 order of magnitude and are strictly positive, then the natural log is likely helpful
- **range rule:** if the range is considerably less than 1 order of magnitude, then transformations are unlikely to help
- **square roots** are useful for count data

Estimating λ

Approach:

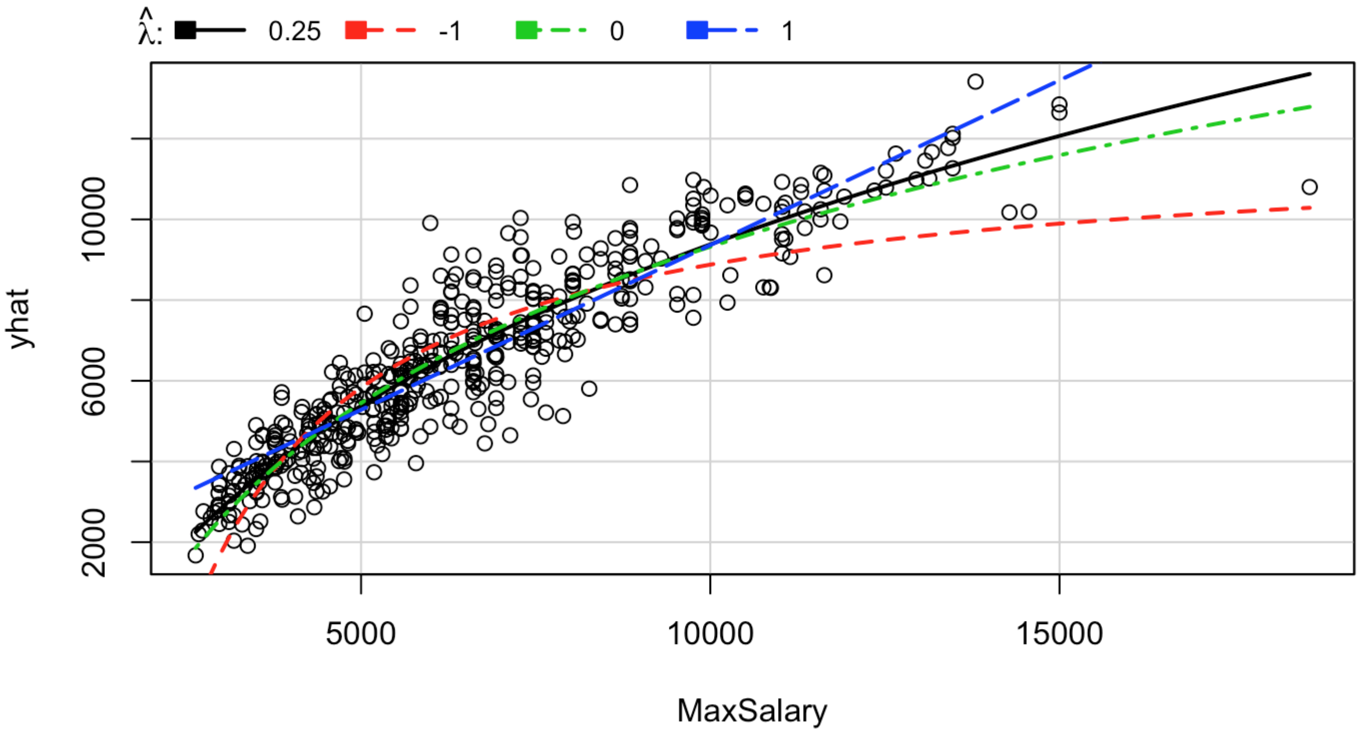
- Minimize RSS($\hat{\lambda}$)
- Use an inverse response plot

Application:

Use the `invResPlot` function found in the `car` package

invRes Plot

```
invResPlot(salary_mod) # prints lambda and RSS in console
```



Transforming the linear model

Approach 1: Create a new column in the data frame

```
library(dplyr)
salarygov <- mutate(salarygov, lmaxsalary = log(MaxSalary))
log_mod1 <- lm(lmaxsalary ~ Score, data = salarygov)
tidy(log_mod1)
```

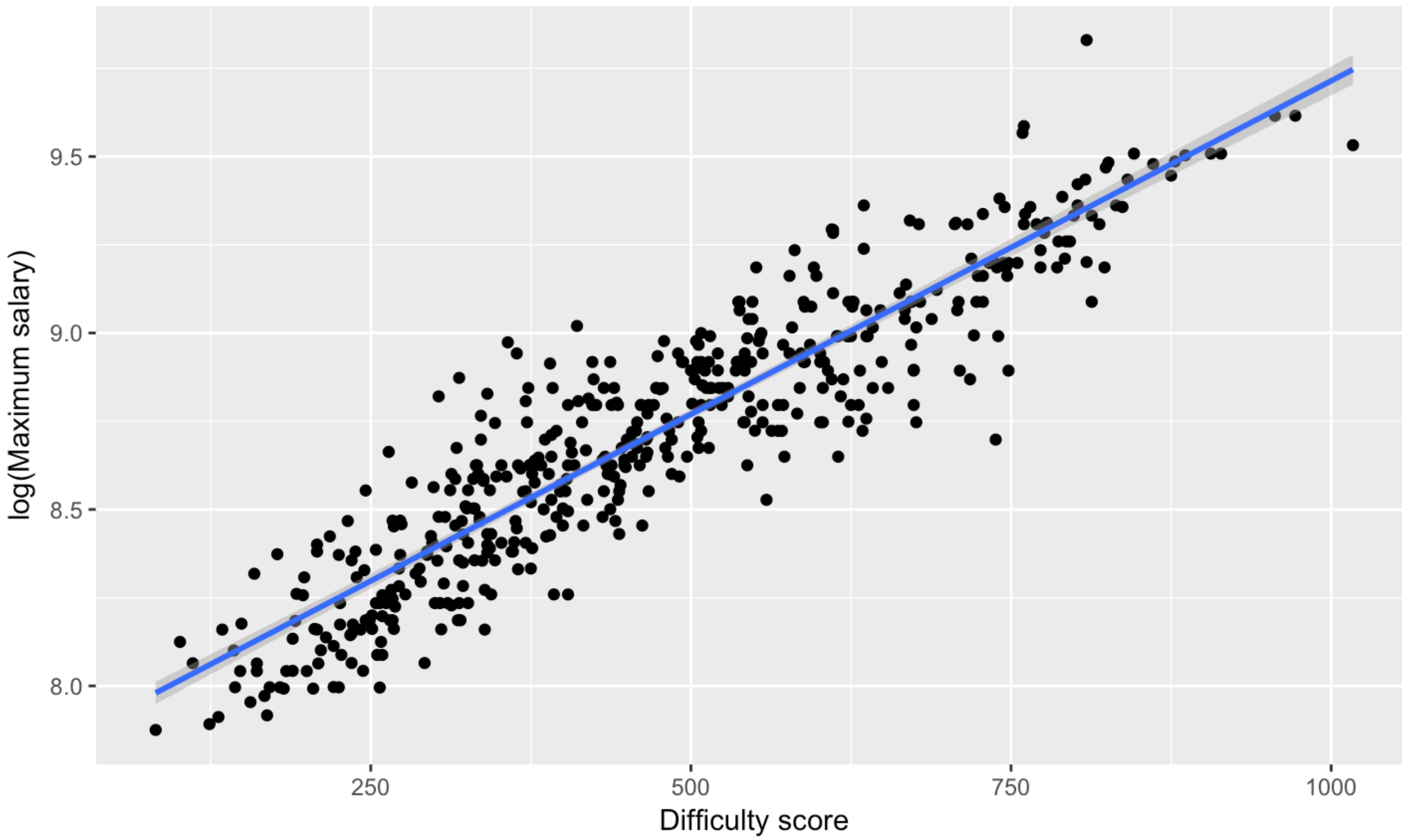
```
##          term    estimate    std.error statistic       p.value
## 1 (Intercept) 7.825242067 1.864010e-02 419.80678  0.000000e+00
## 2       Score 0.001889238 3.695849e-05  51.11783 3.553568e-199
```

Approach 2: Apply the transformation in the `lm` formula

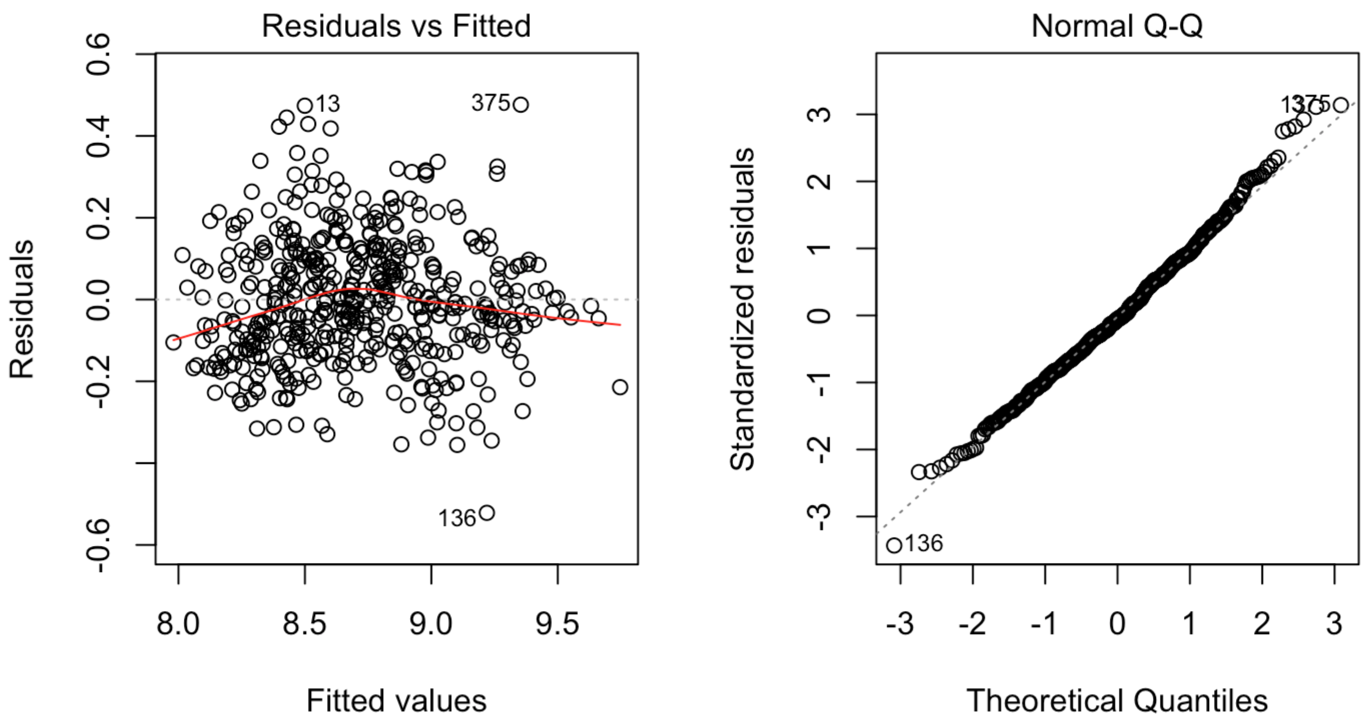
```
log_mod2 <- lm(log(MaxSalary) ~ Score, data = salarygov)
tidy(log_mod2)
```

```
##          term    estimate    std.error statistic       p.value
## 1 (Intercept) 7.825242067 1.864010e-02 419.80678  0.000000e+00
## 2       Score 0.001889238 3.695849e-05  51.11783 3.553568e-199
```

Transformed linear model?

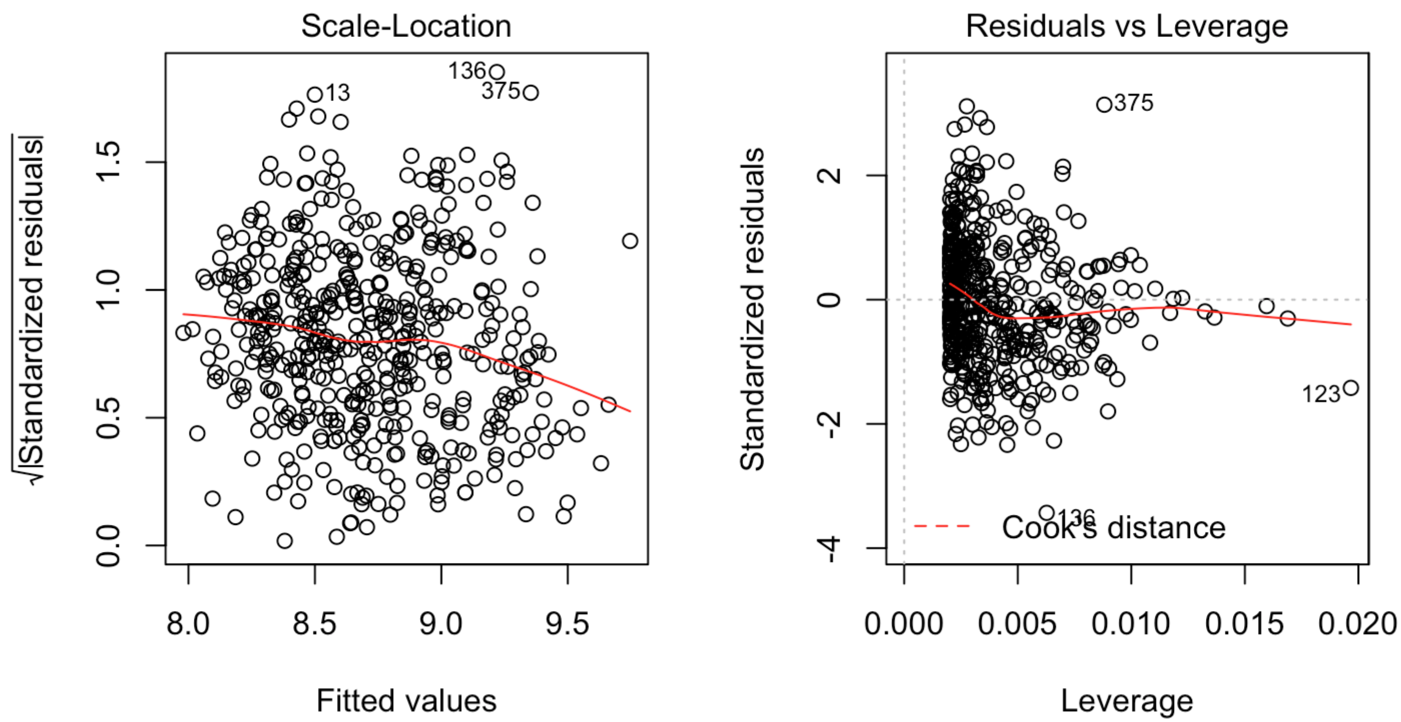


Linearity and normality



The mean function appears to be linear and the residuals are well-approximated by the normal distribution.

Constant variance and influence



There are no influential points and the appears to be constant.

Interpreting the slope?

Making predictions?

We can still use `predict` to make predictions…

```
newdata <- data.frame(Score = 469) # avg. difficulty score
cip <- predict(log_mod2, newdata = newdata, interval = "confidence")
cip
```

```
##        fit      lwr      upr
## 1 8.711295 8.697832 8.724757
```

but we need to back-transform

```
exp(cip)
```

```
##        fit      lwr      upr
## 1 6071.097 5989.913 6153.381
```

Box-Cox transformation

We can automate the selection of the power transformation using the **modified power family** originally defined by Box and Cox (1964)

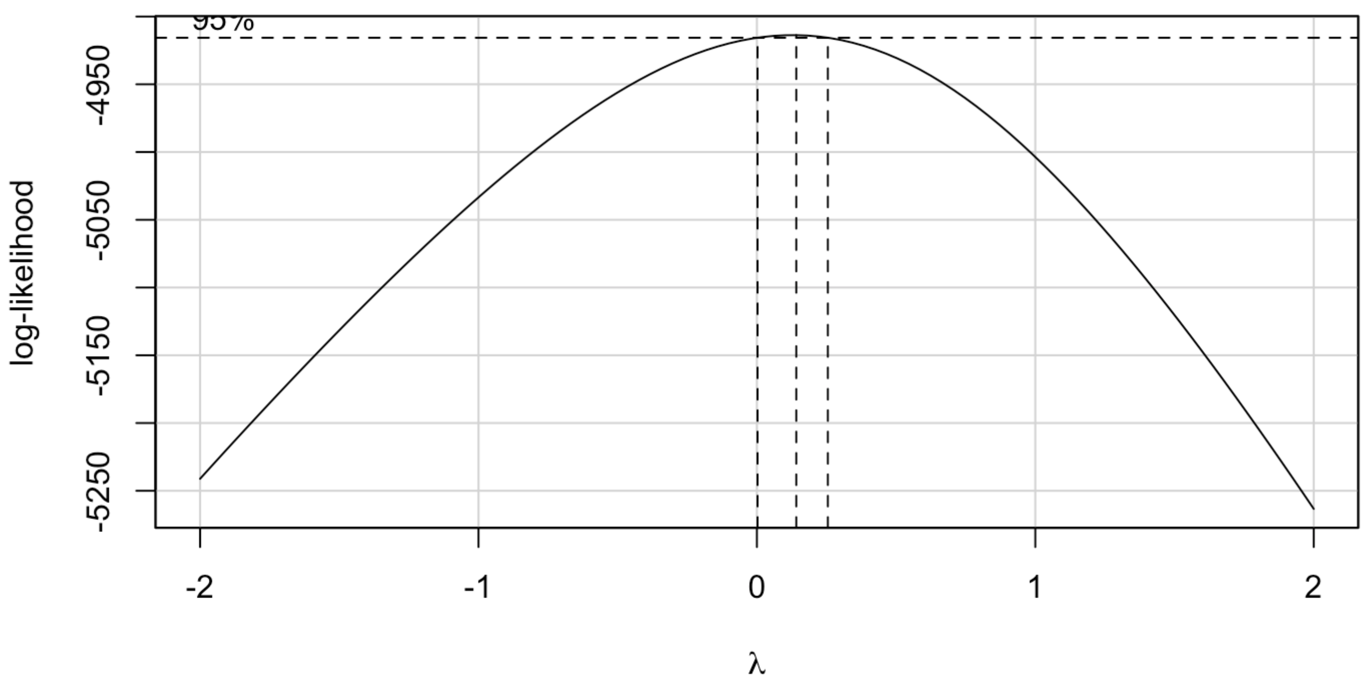
$$\psi_M(Y,\lambda_y) = \begin{cases} \mathrm{gm}(Y)^{1-\lambda_y} \times (Y^{\lambda_y - 1})/\lambda_y & \lambda_y \neq 0 \\ \mathrm{gm}(Y) \times \log(Y) & \lambda_y = 0 \end{cases}$$

where $\mathrm{gm}(Y) = \exp\left(\sum \log(y_i)/n\right)$

- Transforming for normality of residuals
- λ can be estimated via maximum likelihood

Box-Cox transformation

```
boxCox(salary_mod) # to get the plot
```



Box-Cox transformation

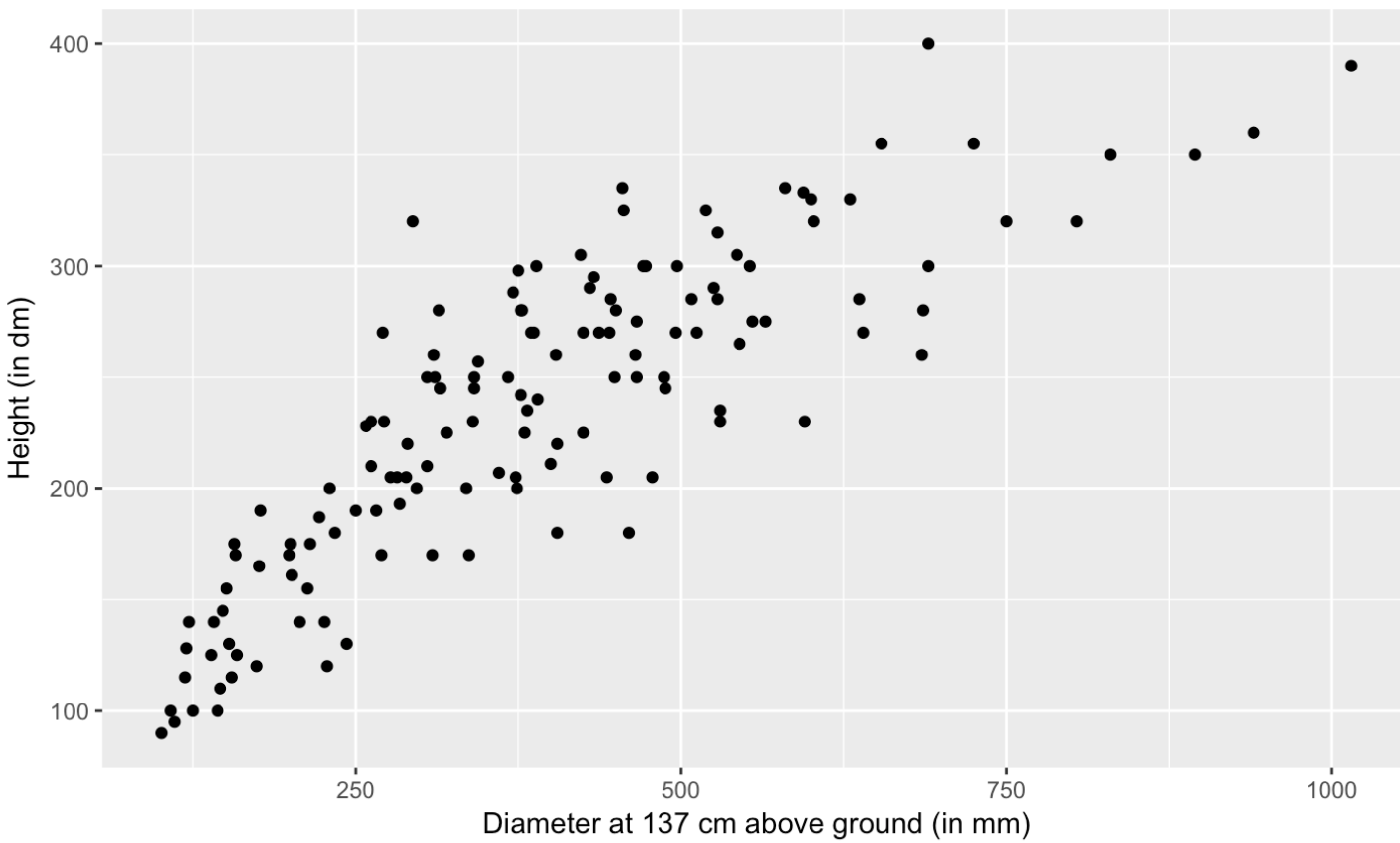
```
summary(powerTransform(salary_mod)) # print the estimate
```

```
## bcPower Transformation to Normality
##
##    Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
## Y1    0.1292   0.0647           0.0025            0.256
##
## Likelihood ratio tests about transformation parameters
##                              LRT df       pval
## LR test, lambda = (0)   3.927686  1 0.04749724
## LR test, lambda = (1) 179.872520  1 0.00000000
```

Transforming the predictor

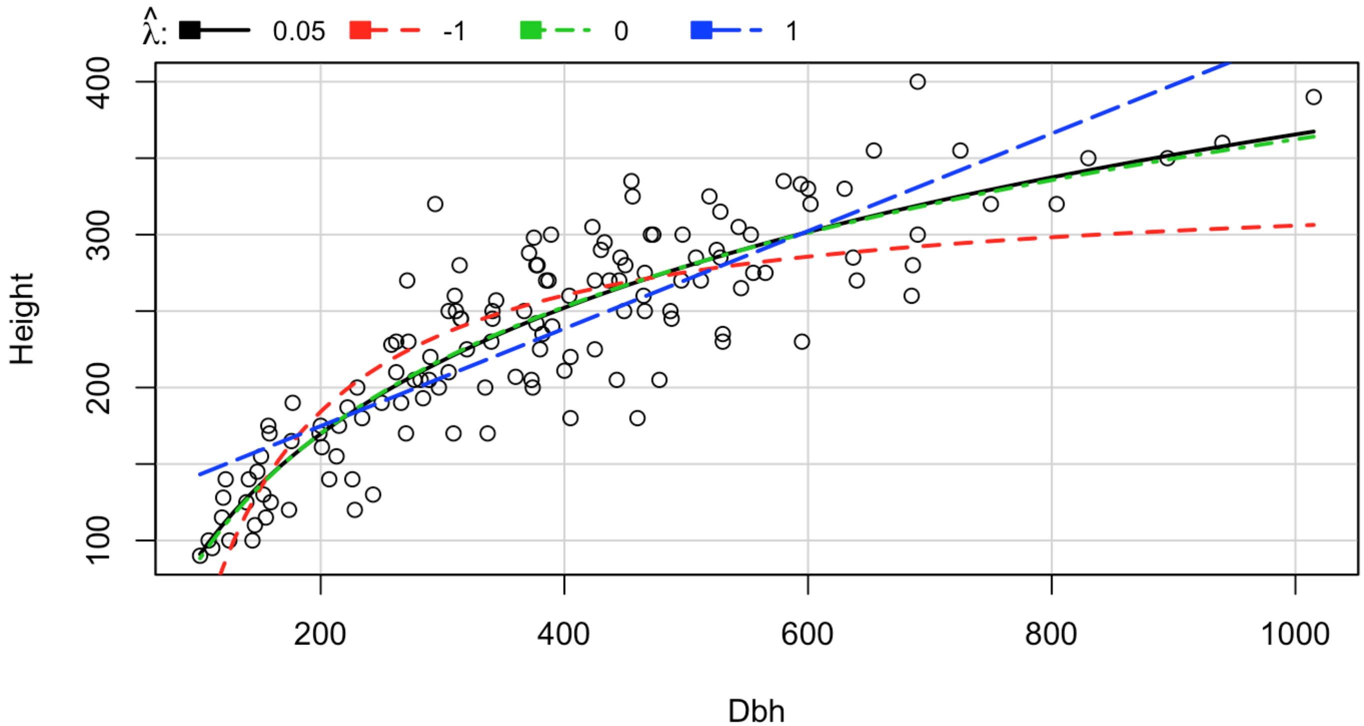
Example

How are tree height and tree diameter related for the Western red cedar?



invTranPlot

```
invTranPlot(Height ~ Dbh, data = ufcwc) # prints lambda and RSS in console
```



Transforming the linear model

Approach 1: Create a new column in the data frame

```
ufcwc <- mutate(ufcwc, ldbh = log(Dbh))
rc_mod1 <- lm(Height ~ ldbh, data = ufcwc)
tidy(rc_mod1)
```

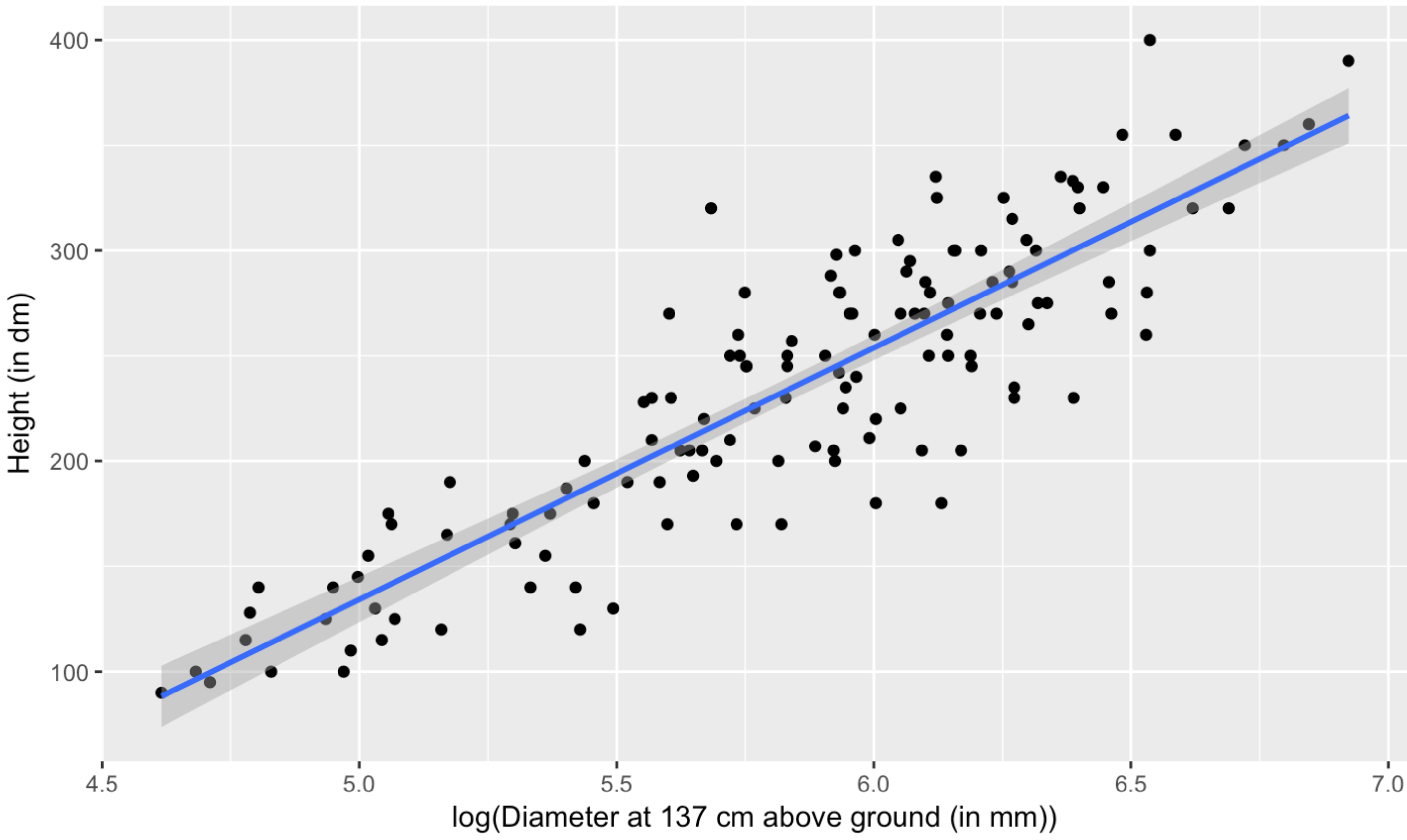
```
##          term  estimate std.error statistic      p.value
## 1 (Intercept) -463.3144 32.437870 -14.28313 6.505273e-29
## 2        ldbh  119.5192  5.531705  21.60621 5.761417e-46
```

Approach 2: Apply the transformation in the `lm` formula

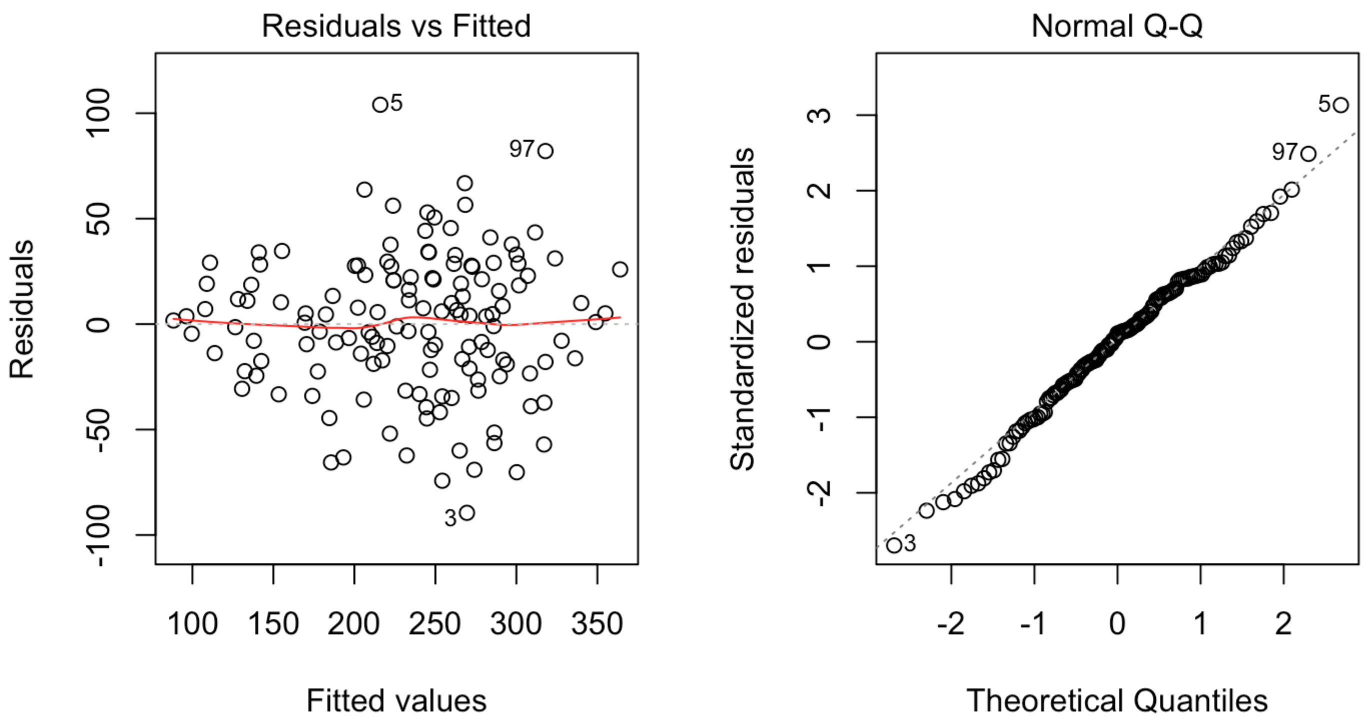
```
rc_mod2 <- lm(Height ~ log(Dbh), data = ufcwc)
tidy(rc_mod2)
```

```
##          term  estimate std.error statistic      p.value
## 1 (Intercept) -463.3144 32.437870 -14.28313 6.505273e-29
## 2    log(Dbh)  119.5192  5.531705  21.60621 5.761417e-46
```

Transformed linear model?

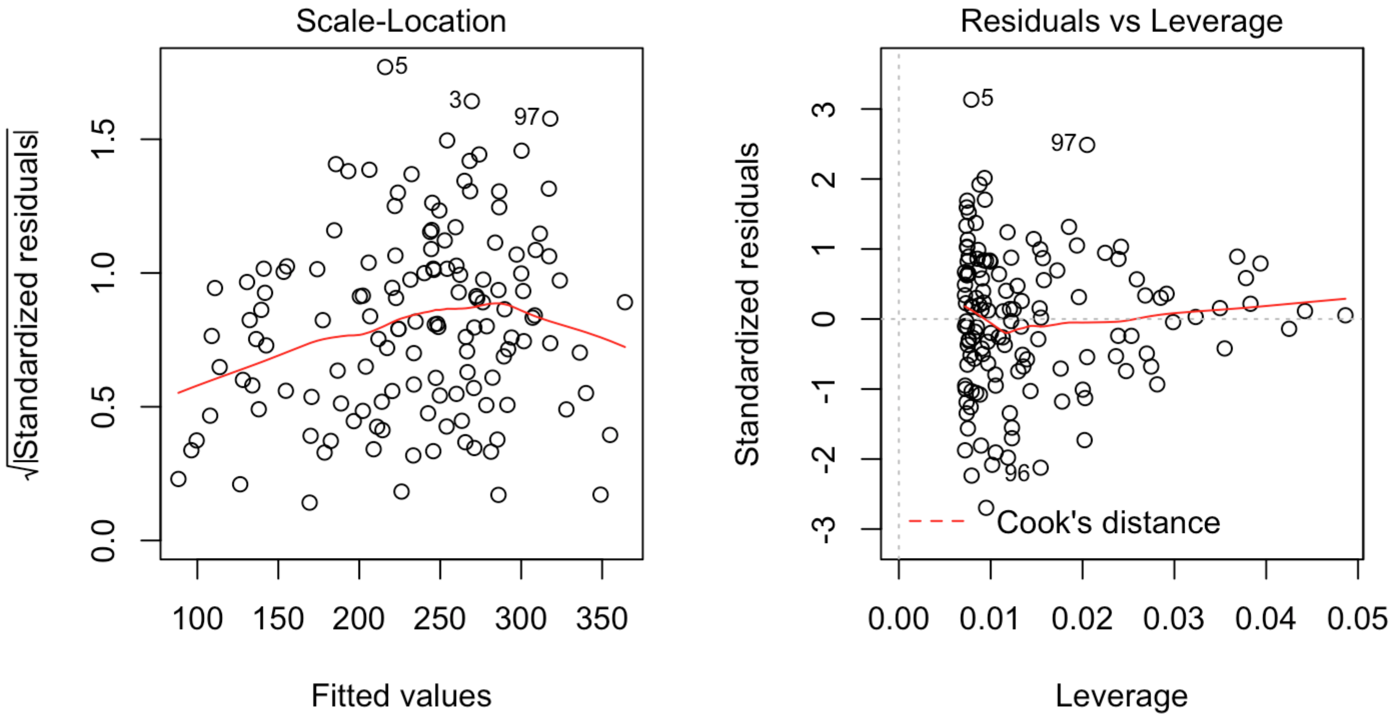


Linearity and normality



The mean function appears to be linear and the residuals are well-approximated by the normal distribution.

Constant variance and influence



There are no influential points and the appears to be roughly constant.

Prediction

If you added a new column to the data frame…

```
newdata <- data.frame(ldbh = log(600))
predict(rc_mod1, newdata = newdata, interval = "prediction")
```

```
##        fit      lwr     upr
## 1 301.2415 234.8101 367.673
```

If you transformed `Dbh` in the model formula…

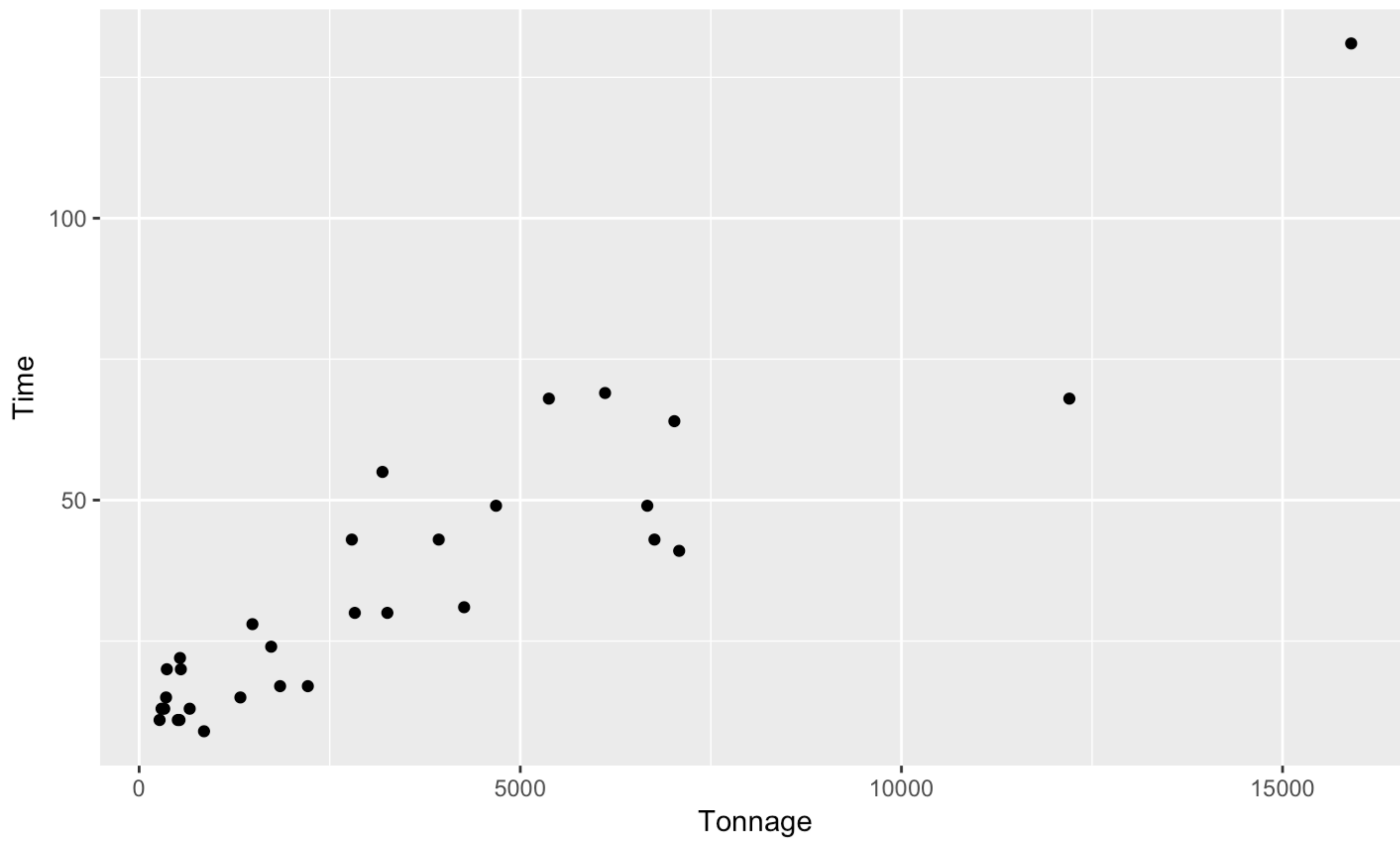
```
newdata2 <- data.frame(Dbh = 600)
predict(rc_mod2, newdata = newdata2, interval = "prediction")
```

```
##        fit      lwr     upr
## 1 301.2415 234.8101 367.673
```

Transforming both variables

Example

How is the volume of a ship's cargo related to the time required to load and unload the cargo?



Approaches

Approach 1:

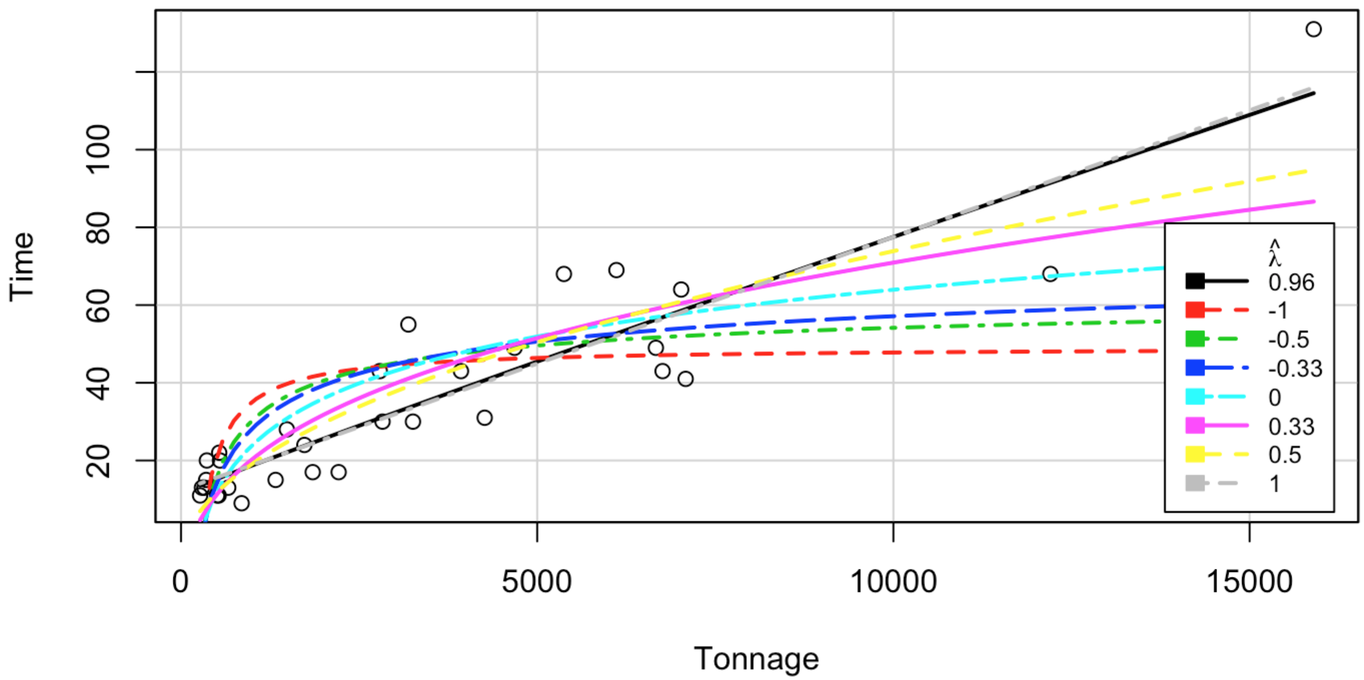
1. Use an inverse transformation plot to choose a transformation for X
2. Transform X, then use an inverse response plot to choose a transformation for Y

Approach 2:

Transform X and Y simultaneously using the Box-Cox procedure

Graphical approach

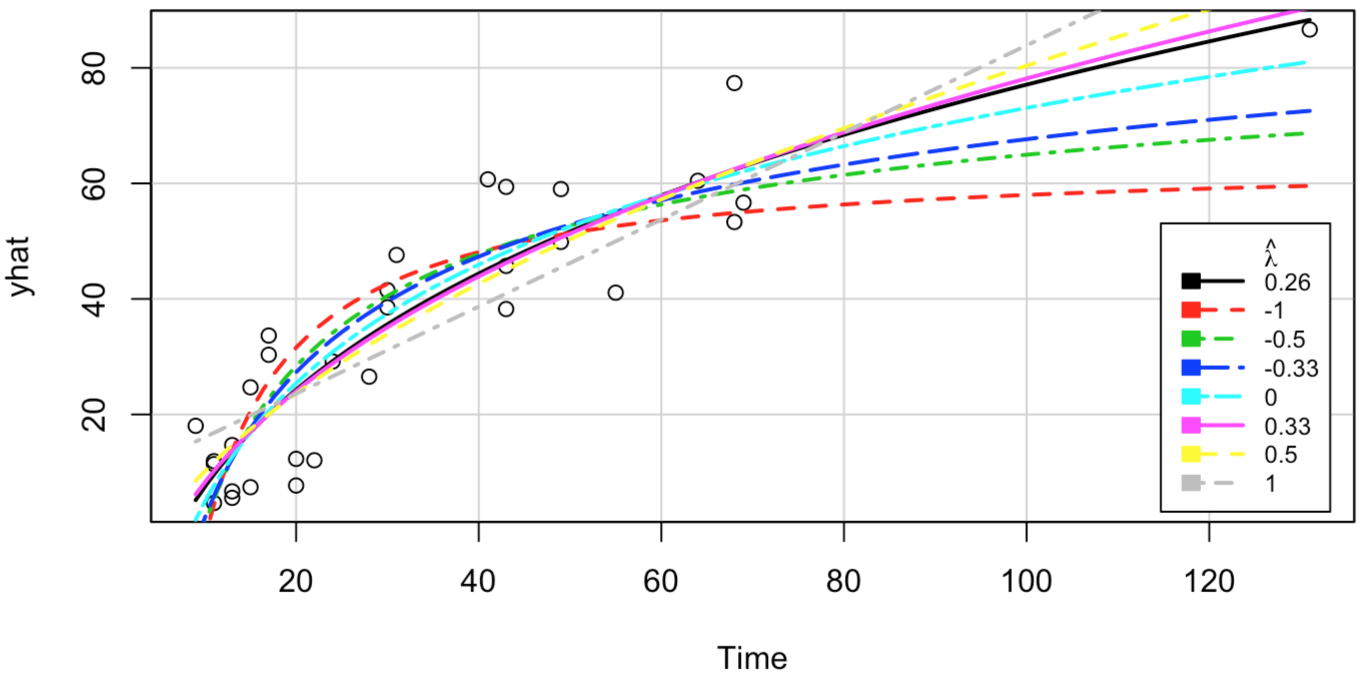
```
invTranPlot(Time ~ Tonnage, data = glakes, lambda = c(-1, -.5, -.33, 0, .33, .5, 1))
```




```
##       lambda       RSS
## 1  0.9591321  3313.093
## 2 -1.0000000 13096.852
```

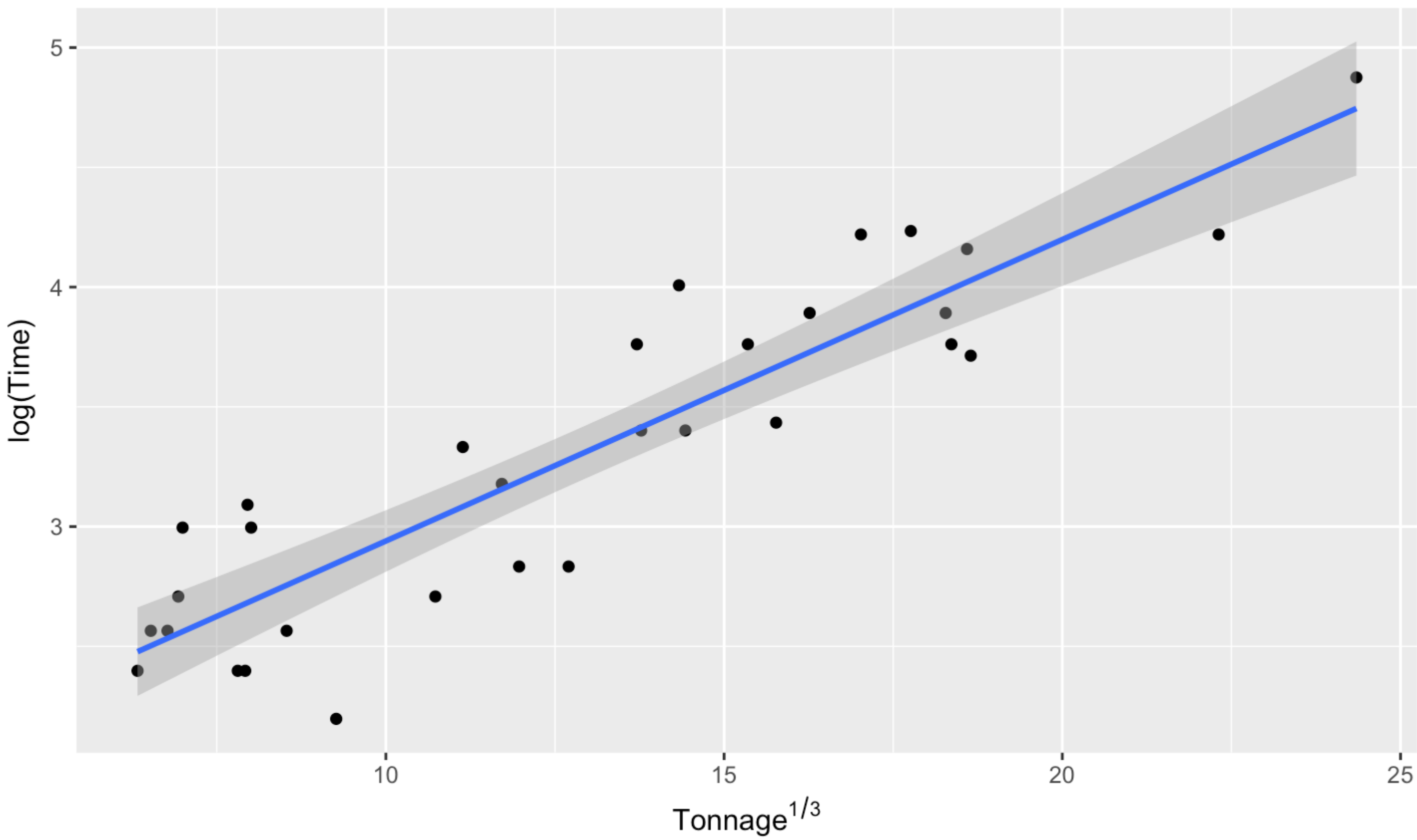
Graphical approach

```
cargo_mod1 <- lm(Time ~ I(Tonnage^.33), data = glakes)
invResPlot(cargo_mod1, lambda = c(-1, -.5, -.33, 0, .33, .5, 1))
```

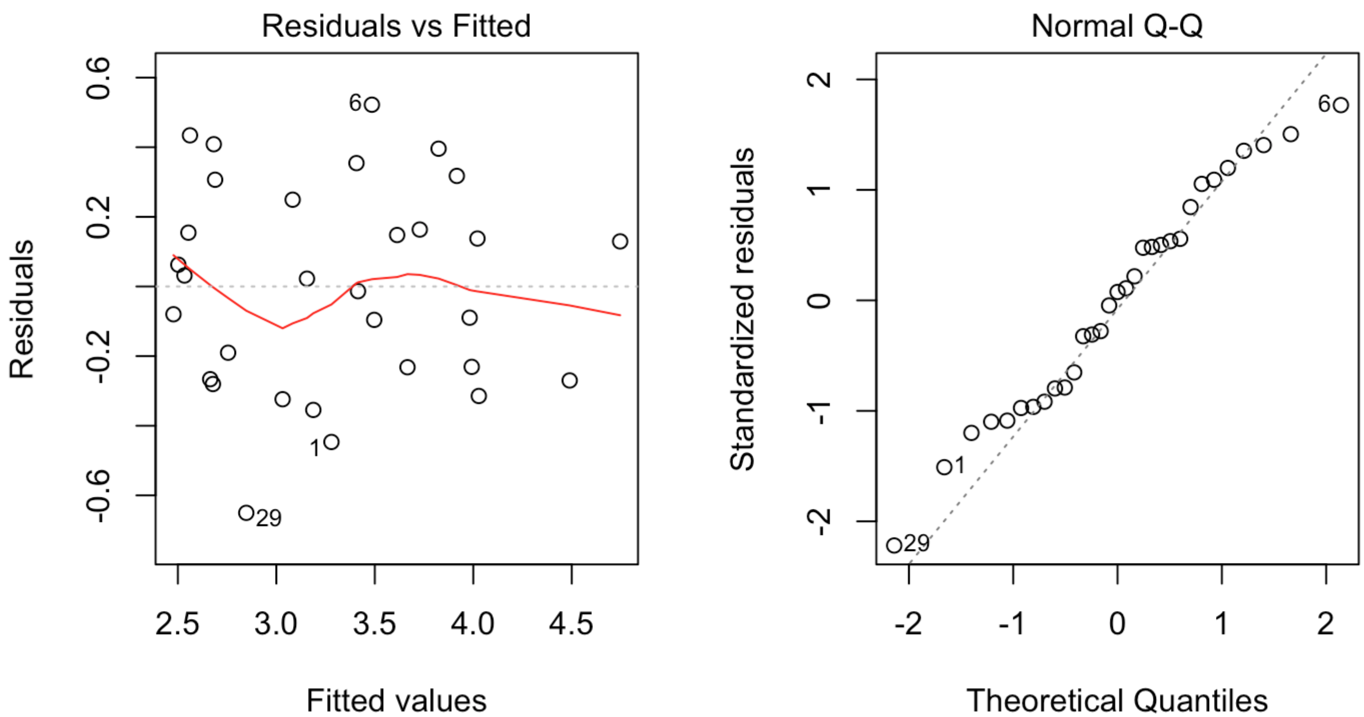


Transformed linear model?

```
cargo_mod2 <- lm(log(Time) ~ I(Tonnage^.33), data = glakes)
```

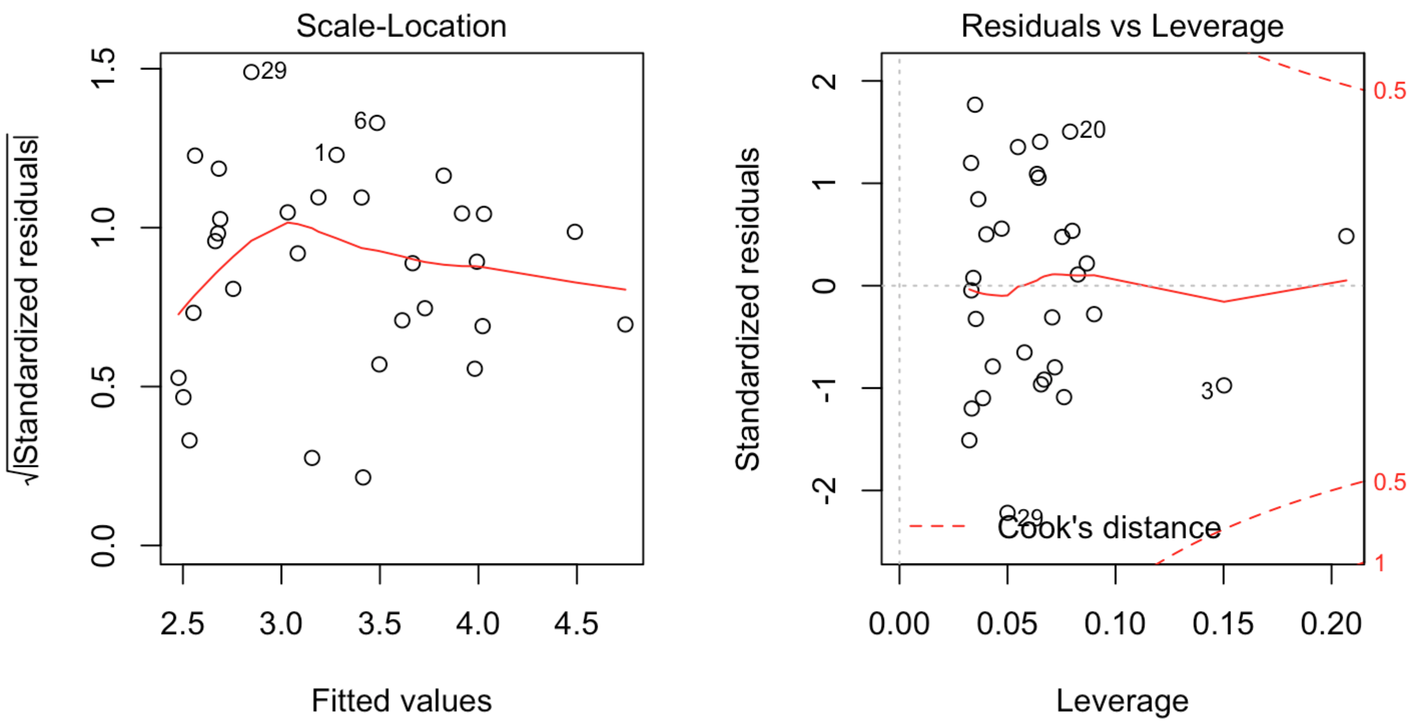


Linearity and normality



The mean function appear to be linear and the residuals are well-approximated by the normal distribution.

Constant variance and influence



There are no influential points and the appears to be roughly constant.

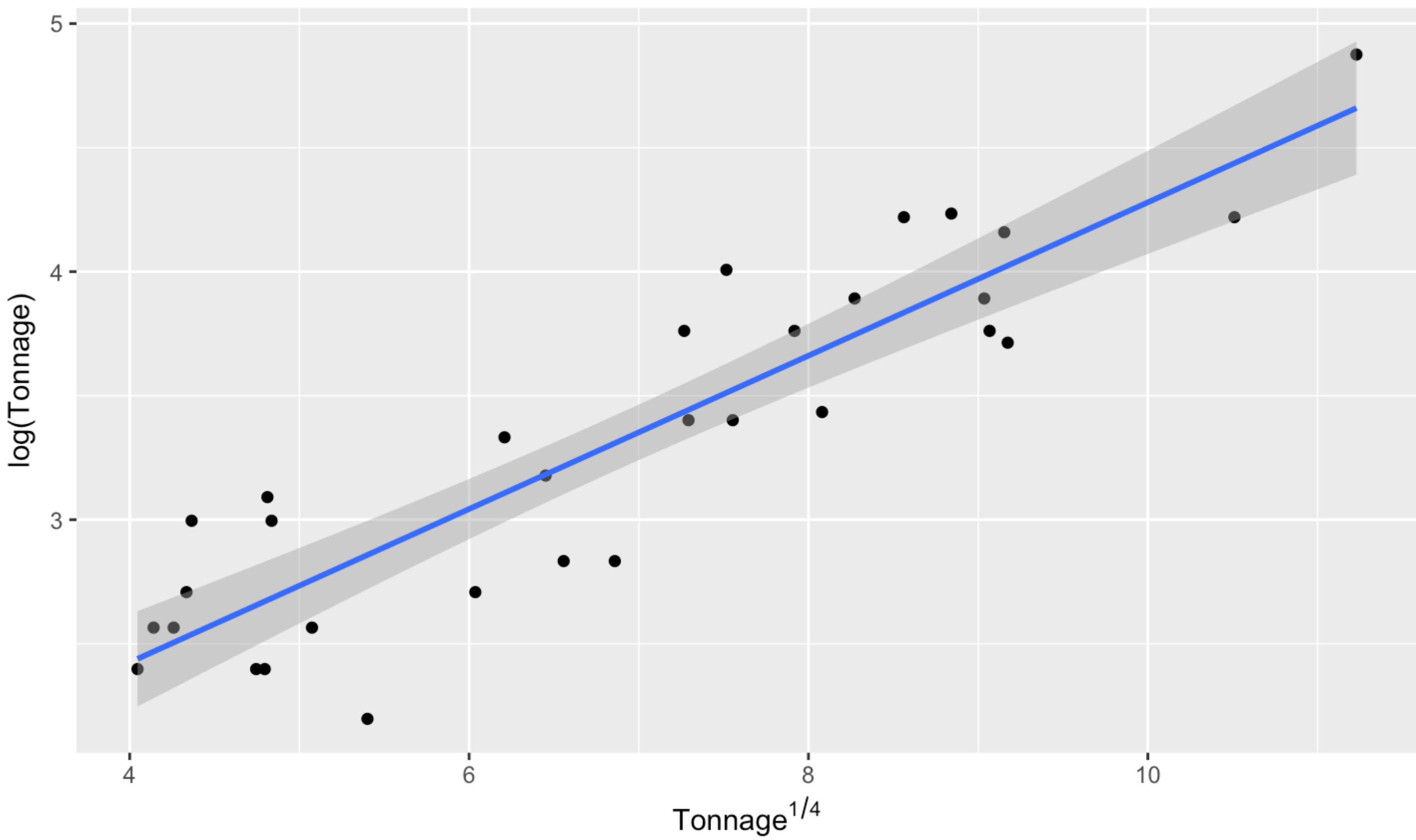
Box-Cox approach

```
summary(powerTransform(cbind(Time, Tonnage) ~ 1, glakes))
```

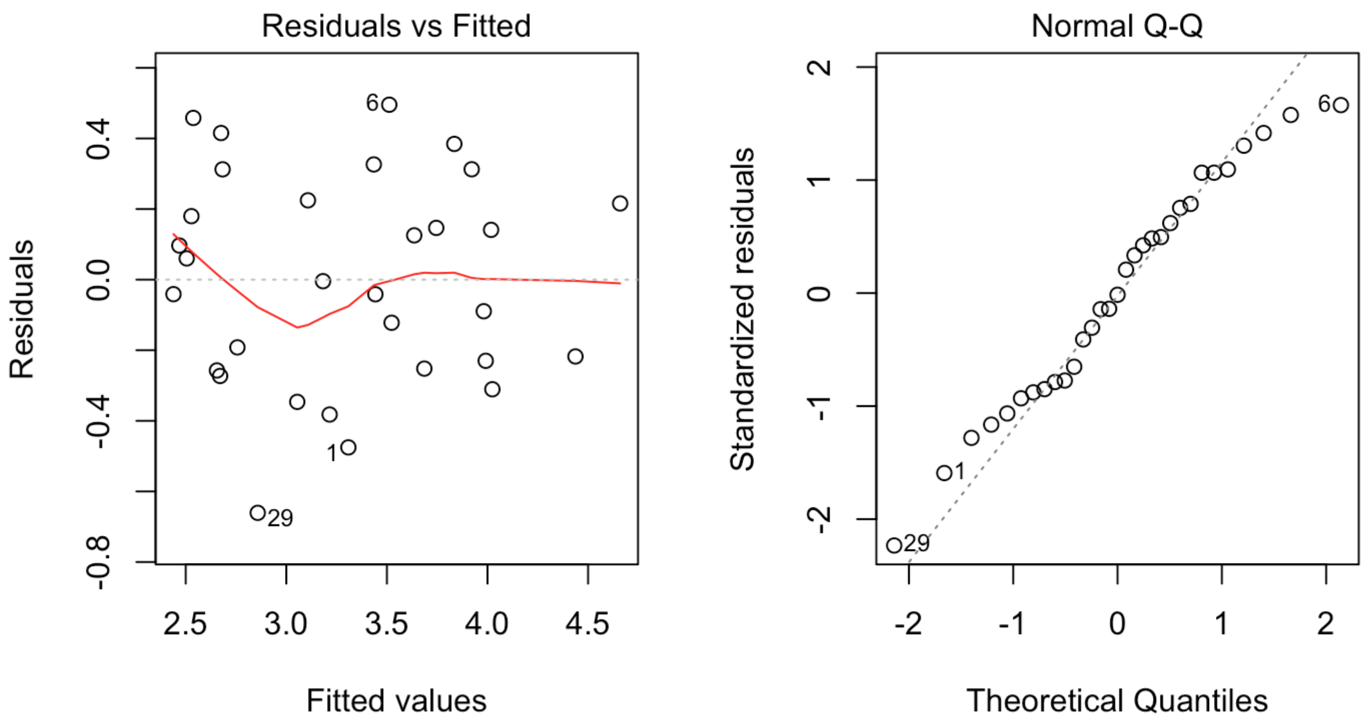
```
## bcPower Transformations to Multinormality
##
##         Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
## Time       0.0228   0.1930          -0.3554           0.4011
## Tonnage    0.2378   0.1237          -0.0046           0.4802
##
## Likelihood ratio tests about transformation parameters
##                              LRT df         pval
## LR test, lambda = (0 0)  3.759605  2 1.526202e-01
## LR test, lambda = (1 1) 45.315290  2 1.445140e-10
```

Transformed linear model?

```
cargo_bcmod <- lm(log(Time) ~ I(Tonnage^.25), data = glakes)
```

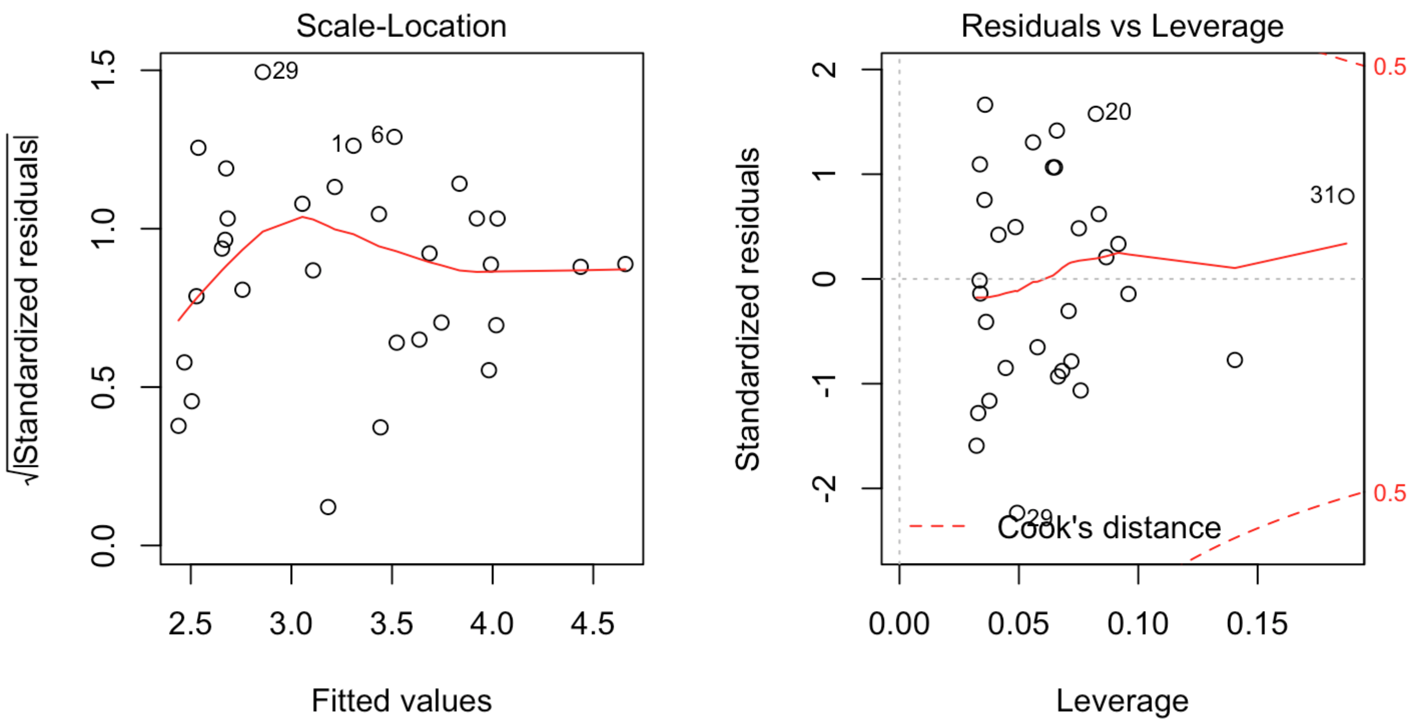


Linearity and normality



The mean function appears to be linear and the residuals are well-approximated by the normal distribution.

Constant variance and influence



There are no influential points and the appears to be roughly constant.

A note on log transformations

If we apply a log transformation to both the response and the predictor, then

$$\%\Delta Y \approx \beta_1 \times \%\Delta x$$

- So, for every 1% increase in x, the model predicts a $\beta_1 \%$ increase in Y
- β_1 needs to be small for this to work out (see p.79 for details)

Issues with Transformations

- You're often guessing
  - Statistics is an art AND a science!
- Changes the interpretation of the parameters
  - need to back-transform to provide interpretable results

- Changes SEs of the parameters
- Not always easy to keep track of all your assumptions