Transformations

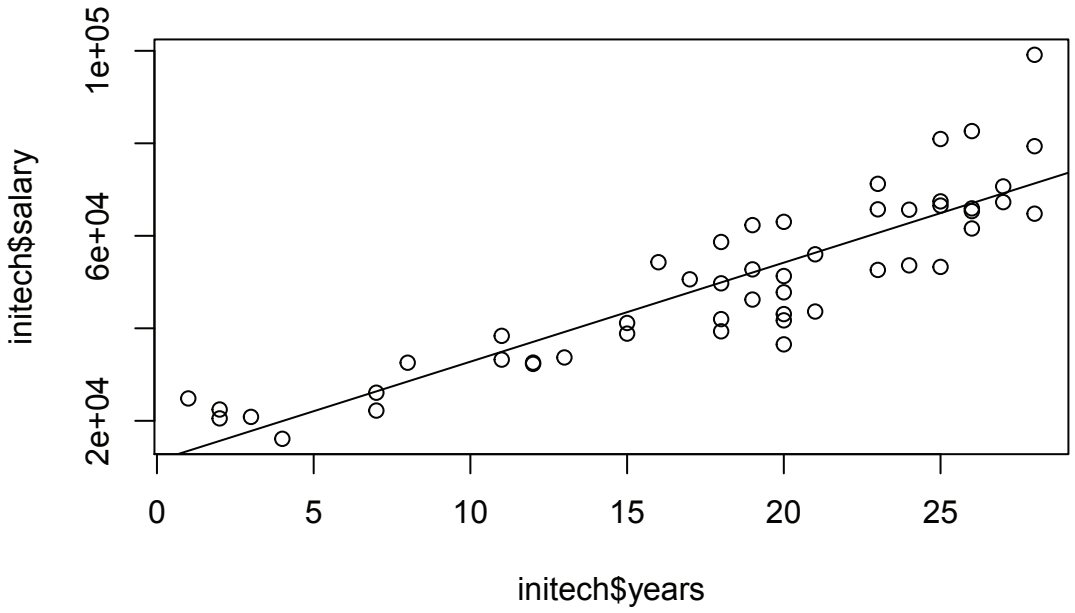
Let's look at some data from *Initech*, where we will try to model `salary` as a function of `years` of experience.

```
salary <- c(26075,79370,65726,41983,62308,41154,53610,33697,22444,32562,43076,
            56000,58667,22210,20521,49727,33233,43628,16105,65644,63022,47780,
            38853,66537,67447,64785,61581,70678,51301,39346,24833,65929,41721,
            82641,99139,52624,50594,53272,65343,46216,54288,20844,32586,71235,
            36530,52745,67282,80931,32303,38371)

years <- c(7,28,23,18,19,15,24,13, 2, 8,20,21,18, 7, 2,18,11,21, 4,24,20,20,15,
           25,25,28,26,27,20,18, 1,26,20,26,28,23,17,25,26,19,16, 3,12,23,20,
           19,27,25,12,11)

initech <- data.frame(years, salary)
head(initech)
```

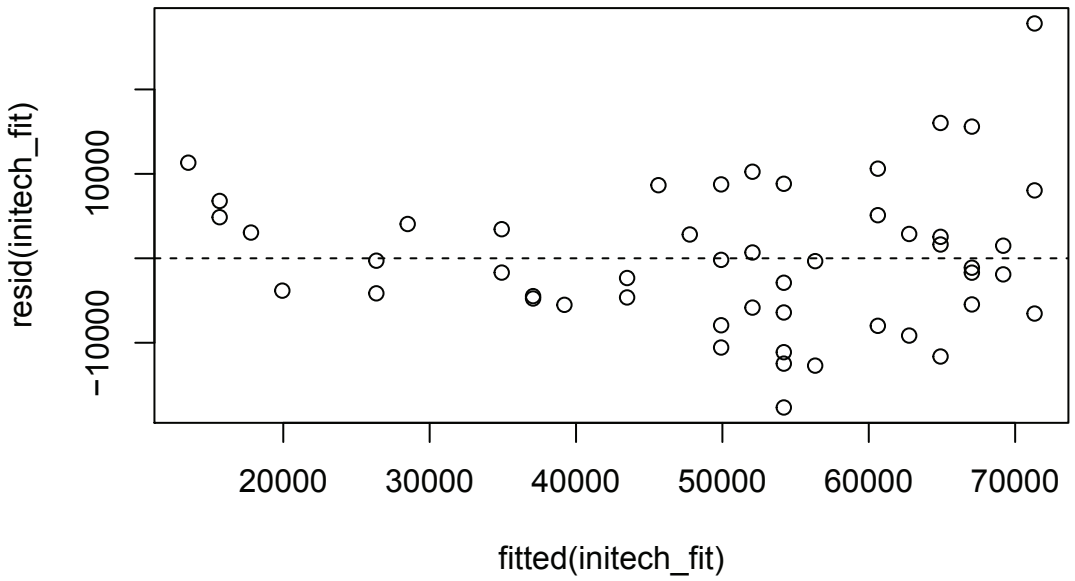
```
##   years salary
## 1     7  26075
## 2    28  79370
## 3    23  65726
## 4    18  41983
## 5    19  62308
## 6    15  41154
```

We first fit a simple linear model.

```
initech_fit <- lm(salary ~ years, data = initech)
#plot(initech_fit)
#summary(initech_fit)
plot(initech$years, initech$salary)
abline(initech_fit)
```




```
plot(fitted(initech_fit), resid(initech_fit))
abline(h = 0, lty = 2)
```



From the fitted versus residuals plot it appears there is non-constant variance. Specifically, the variance

increases as the fitted value increases. (Recall the fitted value is our estimate of the mean at a particular value of x.)

Under our usual assumptions,

$$\epsilon_i \sim N(0, \sigma^2)$$

and thus,

$$Var[Y] = \sigma^2$$

However, here we see that the variance is a function of the mean,

$$Var[Y] = h(\mu)$$

We would like to find some function of Y, $g(Y)$ such that,

$$Var[g(Y)] = c$$

where c is a constant that does not depend on μ. A transformation that accomplishes this is called a **variance stabilizaing transformation.**

A common VST when we see increasing variance in a fitted versus residuals plot is $\log(Y)$. Also, if the values of a variable range over more than one order of magnitude and the variable is *strictly positive*, then replacing the variable by its logarithm is likely to be helpful.

(A reminder, that for our purposes, log and ln are both the natural log. `R` uses `log` to mean the natural log, unless a different base is specified.)

We will now use a log transformed response for the *Initech* data,

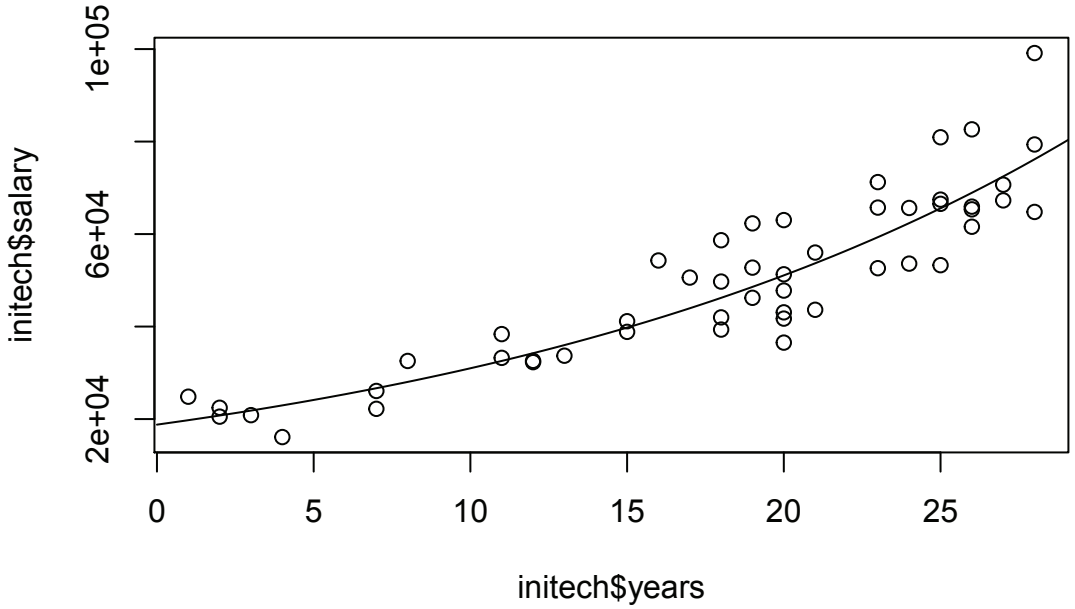
$$\log(y_i) = \beta_0 + \beta_1 x_i + \epsilon_i$$

Note, if we rescale the data from a log scale back to the original scale of the data, we now have

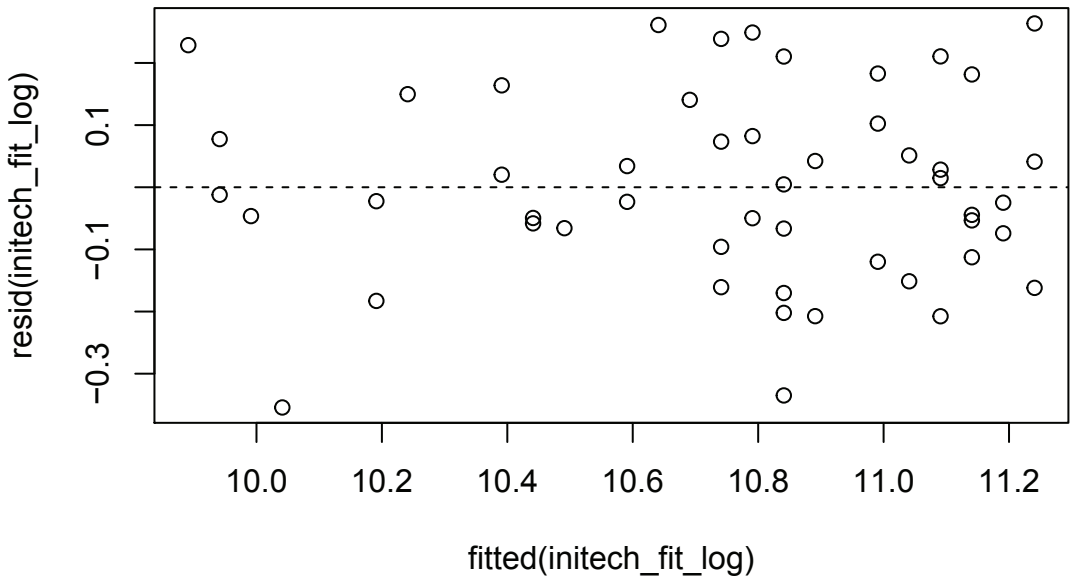
$$y_i = \exp(\beta_0 + \beta_1 x) \exp(\epsilon_i)$$

```
initech_fit_log <- lm(log(salary) ~ years, data = initech)
plot(initech$years, log(initech$salary)) #plot on log scale
abline(initech_fit_log)
```


```
plot(initech$years, initech$salary) #plot on data scale
curve(exp(initech_fit_log$coef[1] + initech_fit_log$coef[2] * x), 0, 30, add = T)
```



```r
plot(fitted(initech_fit_log), resid(initech_fit_log))
abline(h = 0, lty = 2)
```



Here we see this model fits much better, and it does not appear to violate the constant variance assumption.

```r
sum((initech$salary - fitted(initech_fit))^2)
```

```
## [1] 3585052519
```

```r
sum((initech$salary - exp(fitted(initech_fit_log)))^2)
```

```
## [1] 3100401284
```

```r
summary(initech_fit_log)
```

```
##
## Call:
## lm(formula = log(salary) ~ years, data = initech)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35435 -0.09045 -0.01726  0.09740  0.26357
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9.841325   0.056355  174.63   <2e-16 ***
## years       0.049978   0.002868   17.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1541 on 48 degrees of freedom
## Multiple R-squared:  0.8635, Adjusted R-squared:  0.8607
## F-statistic: 303.6 on 1 and 48 DF,  p-value: < 2.2e-16
```

$$\log(\hat{y}) = \hat{\beta}_0 + \hat{\beta}_1 x = 9.84 + 0.05x$$

Note, if we rescale the data from a log scale back to the original scale of the data, we now have

$$\hat{y} = \exp(\hat{\beta}_0)\exp(\hat{\beta}_1 x) = \exp(9.84)\exp(0.05x)$$

We see that for every one additional year of experience, salary increases $\exp(0.05) = 1.051$ times. (Multiply.)

Box-Cox Transformations

The Box-Cox method considers a family of transformations on strictly positive response variables,

$$g_\lambda(y) = \begin{cases} \dfrac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log(y) & \lambda = 0 \end{cases}$$

λ is chosen by numerically by maxmimizing the log-likelihood,

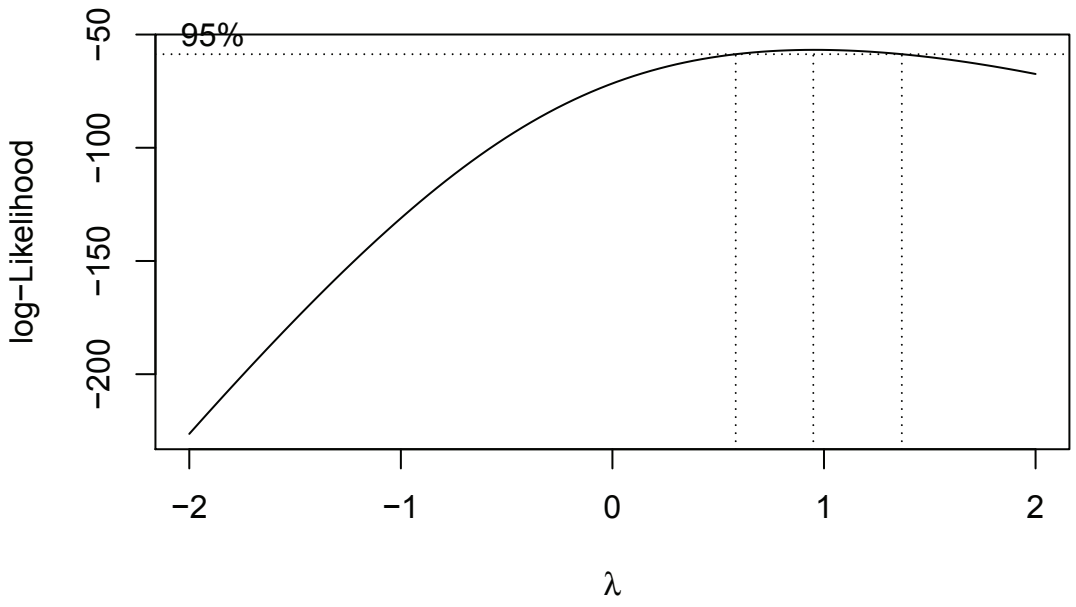
$$\log(L(\lambda)) = -\frac{n}{2}\log(RSS_\lambda/n) + (\lambda - 1)\sum \log(y_i)$$

A $100(1-\alpha)\%$ confidence interval for λ is,

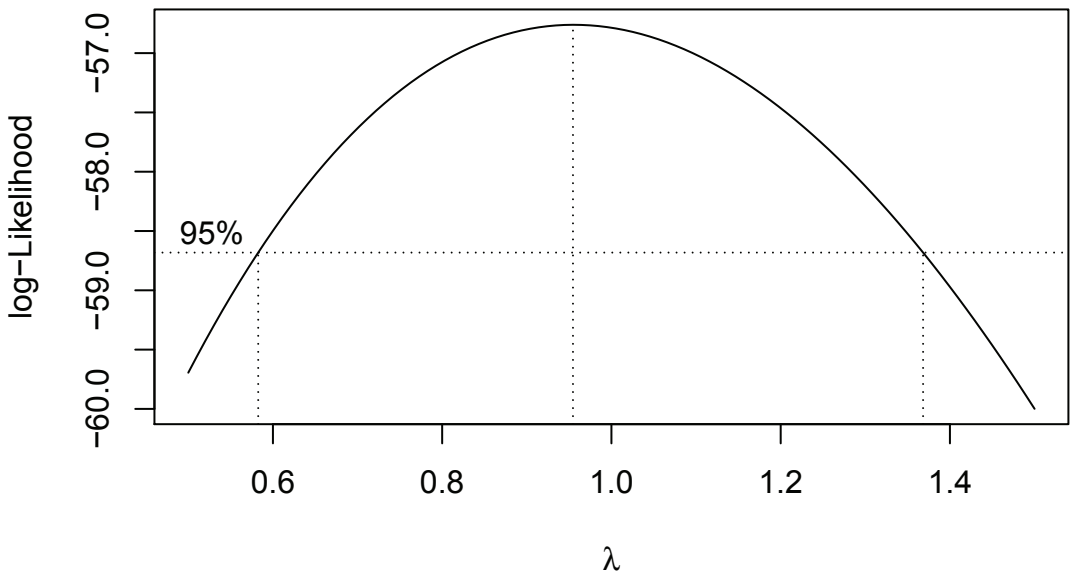
$$\left\{\lambda : L(\lambda) > L(\hat{\lambda}) - \frac{1}{2}\chi^2_{1,\alpha}\right\}$$

which `R` will plot for us to help quickly select an appropriate λ.

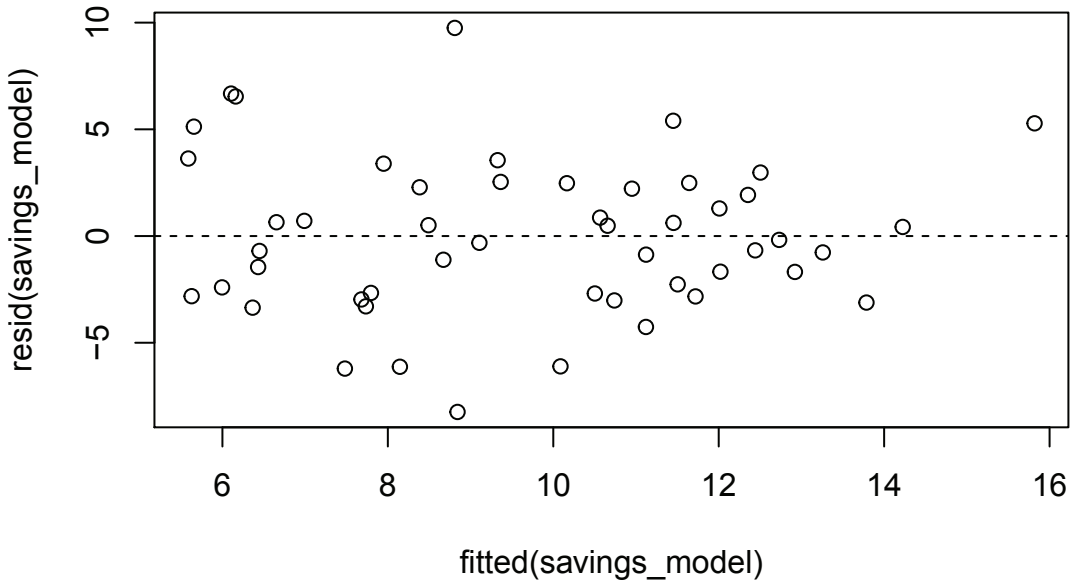
```r
library(MASS)
library(faraway)

data(savings)
savings_model <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data = savings)

boxcox(savings_model, plotit = TRUE)
```




```r
boxcox(savings_model, plotit = TRUE, lambda = seq(0.5,1.5,by = 0.1))
```



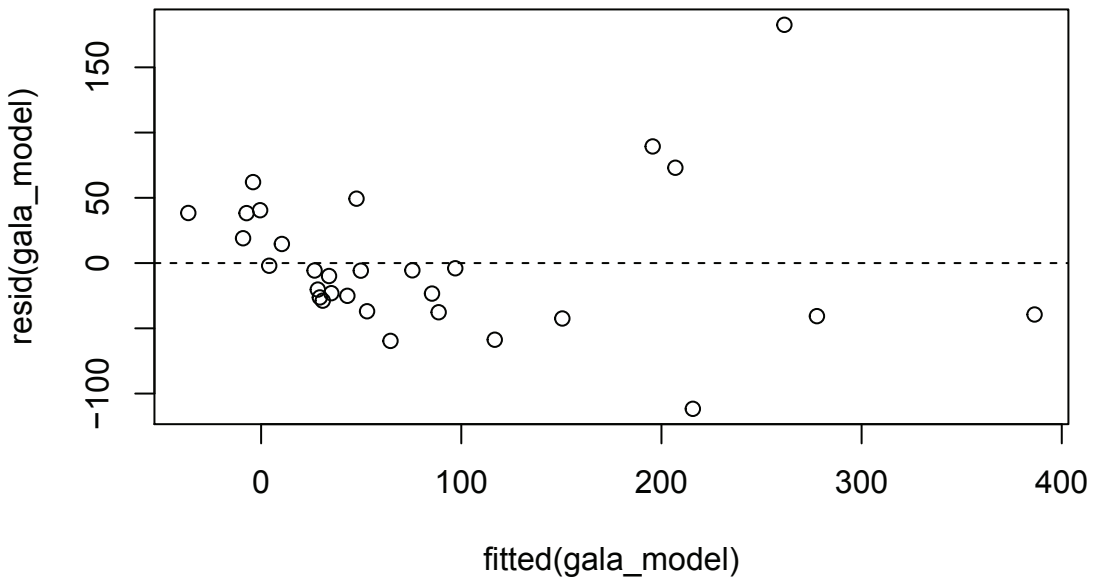

```
#plot(savings_model)
plot(fitted(savings_model), resid(savings_model))
abline(h = 0, lty = 2)
```



```r
summary(savings_model)
```

```
##
## Call:
## lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2422 -2.6857 -0.2488  2.4280  9.7509
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
## pop15       -0.4611931  0.1446422  -3.189 0.002603 **
## pop75       -1.6914977  1.0835989  -1.561 0.125530
## dpi         -0.0003369  0.0009311  -0.362 0.719173
## ddpi         0.4096949  0.1961971   2.088 0.042471 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.803 on 45 degrees of freedom
## Multiple R-squared:  0.3385, Adjusted R-squared:  0.2797
## F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904
```

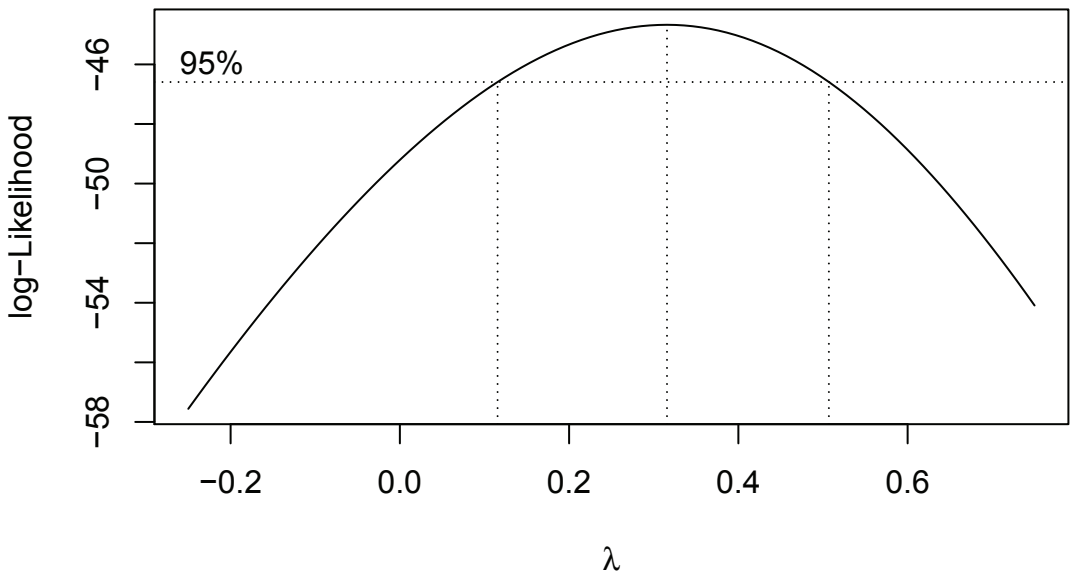
```r
data(gala)
gala_model <- lm(Species ~ Area + Elevation + Nearest + Scruz
                 + Adjacent, data = gala)
plot(fitted(gala_model), resid(gala_model))
abline(h = 0, lty = 2)
```



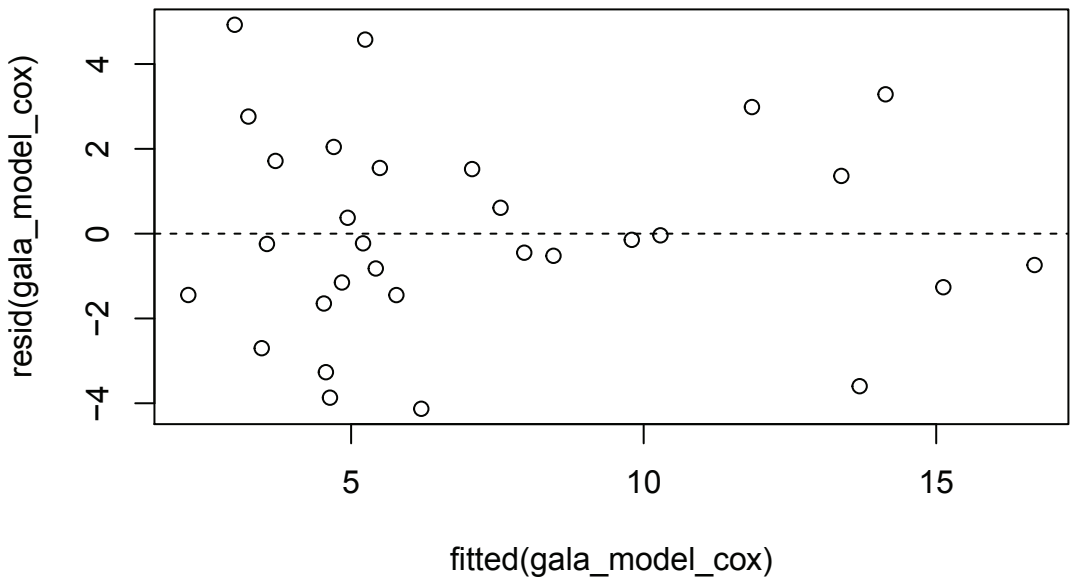

```
summary(gala_model)
```

```
##
## Call:
## lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
##     data = gala)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.679  -34.898   -7.862   33.460  182.584
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.068221  19.154198   0.369 0.715351
## Area        -0.023938   0.022422  -1.068 0.296318
## Elevation    0.319465   0.053663   5.953 3.82e-06 ***
## Nearest      0.009144   1.054136   0.009 0.993151
## Scruz       -0.240524   0.215402  -1.117 0.275208
## Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.98 on 24 degrees of freedom
## Multiple R-squared:  0.7658, Adjusted R-squared:  0.7171
## F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

```
boxcox(gala_model, lambda=seq(-0.25,0.75,by=0.05),plotit=T)
```




```
gala_model_cox <- lm(((Species^0.3)-1)/0.3 ~ Area + Elevation
                     + Nearest + Scruz +  Adjacent, data = gala)
plot(fitted(gala_model_cox), resid(gala_model_cox))
abline(h = 0, lty = 2)
```



```
summary(gala_model_cox)
```

```
##
## Call:
## lm(formula = ((Species^0.3) - 1)/0.3 ~ Area + Elevation + Nearest +
##     Scruz + Adjacent, data = gala)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1301 -1.4007 -0.2357  1.5423  4.9260
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.5618689  0.8144515   4.373 0.000204 ***
## Area        -0.0019671  0.0009534  -2.063 0.050074 .
## Elevation    0.0142730  0.0022818   6.255 1.83e-06 ***
## Nearest      0.0329434  0.0448227   0.735 0.469478
## Scruz       -0.0120948  0.0091591  -1.321 0.199114
## Adjacent    -0.0027477  0.0007526  -3.651 0.001267 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.593 on 24 degrees of freedom
## Multiple R-squared:  0.7457, Adjusted R-squared:  0.6927
## F-statistic: 14.07 on 5 and 24 DF,  p-value: 1.779e-06
```

```
boxcox(initech_fit)
```