Balcewicz HW 3

Katie Balcewicz

1/30/2018

Univariate Assignment

Read in tree data, metadata can be found in: ./data/tree_metadata.txt

```
trees = read.csv("https://raw.githubusercontent.com/dmcglinn/quant_methods/gh-pages/data/treedata_subse
                 header = TRUE)
acer = subset(trees, species == "Acer rubrum")
abies = subset(trees, species == "Abies fraseri")
```

1. Exploratory analysis

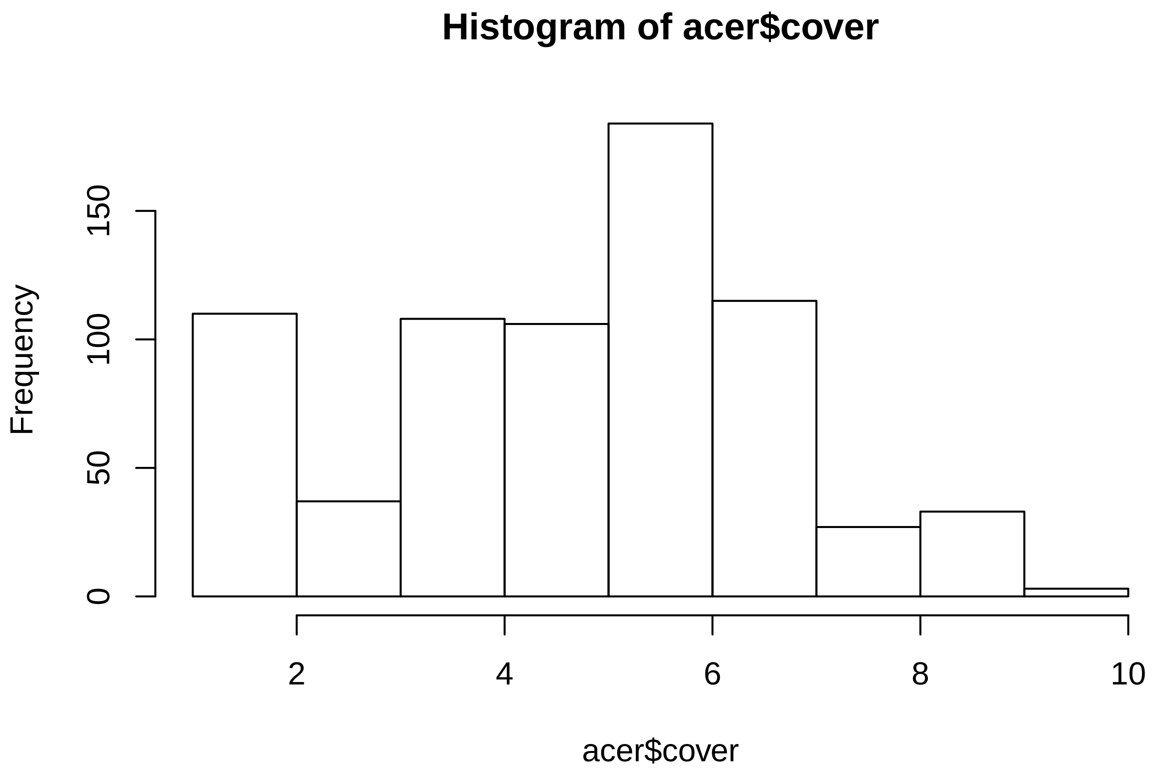
```
str(acer)
```

```
## 'data.frame':    723 obs. of  9 variables:
##  $ plotID    : Factor w/ 734 levels "ATBN-01-0303",..: 1 2 3 4 5 6 8 9 10 18 ...
##  $ spcode    : Factor w/ 52 levels "ABIEFRA","ACERNEG",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ species   : Factor w/ 51 levels "Abies fraseri",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ cover     : int  6 7 5 7 5 4 2 7 4 7 ...
##  $ elev      : num  896 947 1027 450 477 ...
##  $ tci       : num  4.71 4.45 6.15 4.13 5.59 ...
##  $ streamdist: num  197 125 175 202 134 ...
##  $ disturb   : Factor w/ 4 levels "CORPLOG","LT-SEL",..: 1 1 1 2 2 2 1 4 2 1 ...
##  $ beers     : num  1.991 0.817 0.586 0.86 0.101 ...
```

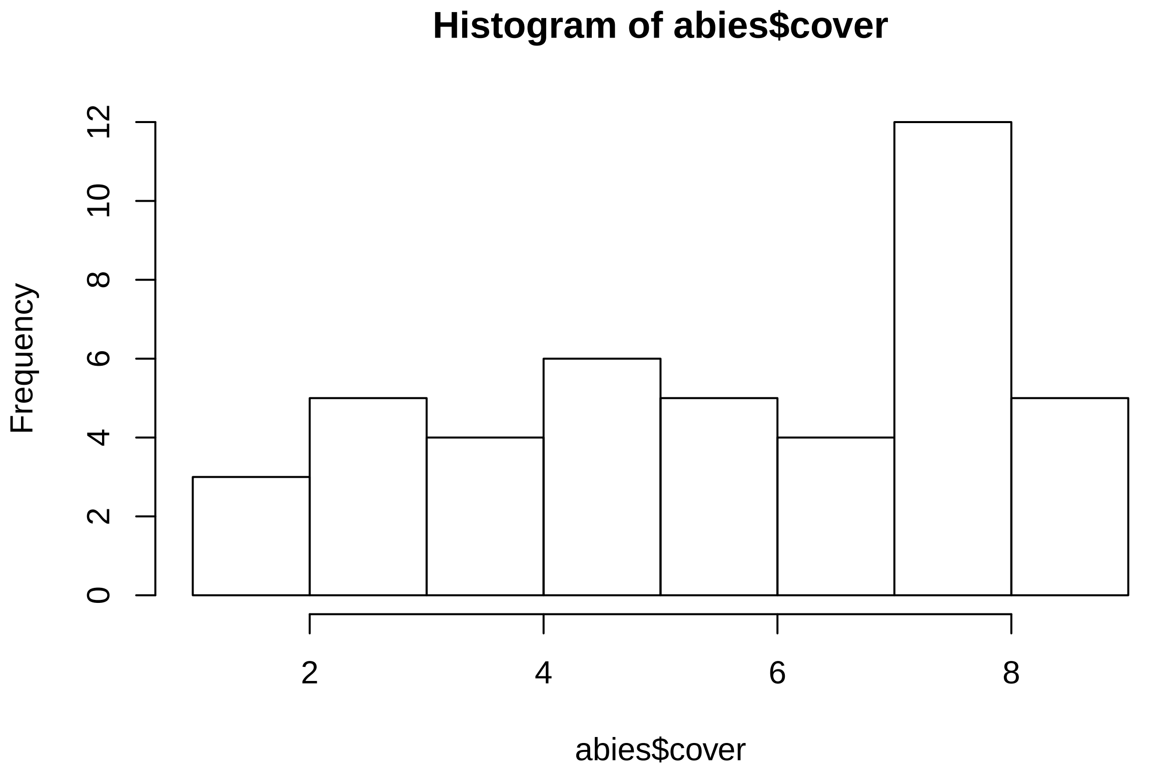
```
str(abies)
```

```
## 'data.frame':    44 obs. of  9 variables:
##  $ plotID    : Factor w/ 734 levels "ATBN-01-0303",..: 20 53 54 56 109 188 452 471 471 471 ...
##  $ spcode    : Factor w/ 52 levels "ABIEFRA","ACERNEG",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ species   : Factor w/ 51 levels "Abies fraseri",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ cover     : int  1 8 3 3 5 2 4 8 8 5 ...
##  $ elev      : num  1660 1712 1722 1754 1570 ...
##  $ tci       : num  5.7 3.82 3.89 3.15 11.85 ...
##  $ streamdist: num  491 454 453 492 0 ...
##  $ disturb   : Factor w/ 4 levels "CORPLOG","LT-SEL",..: 1 4 2 3 2 4 4 4 4 4 ...
##  $ beers     : num  0.224 0.834 1.333 1.471 0.496 ...
```

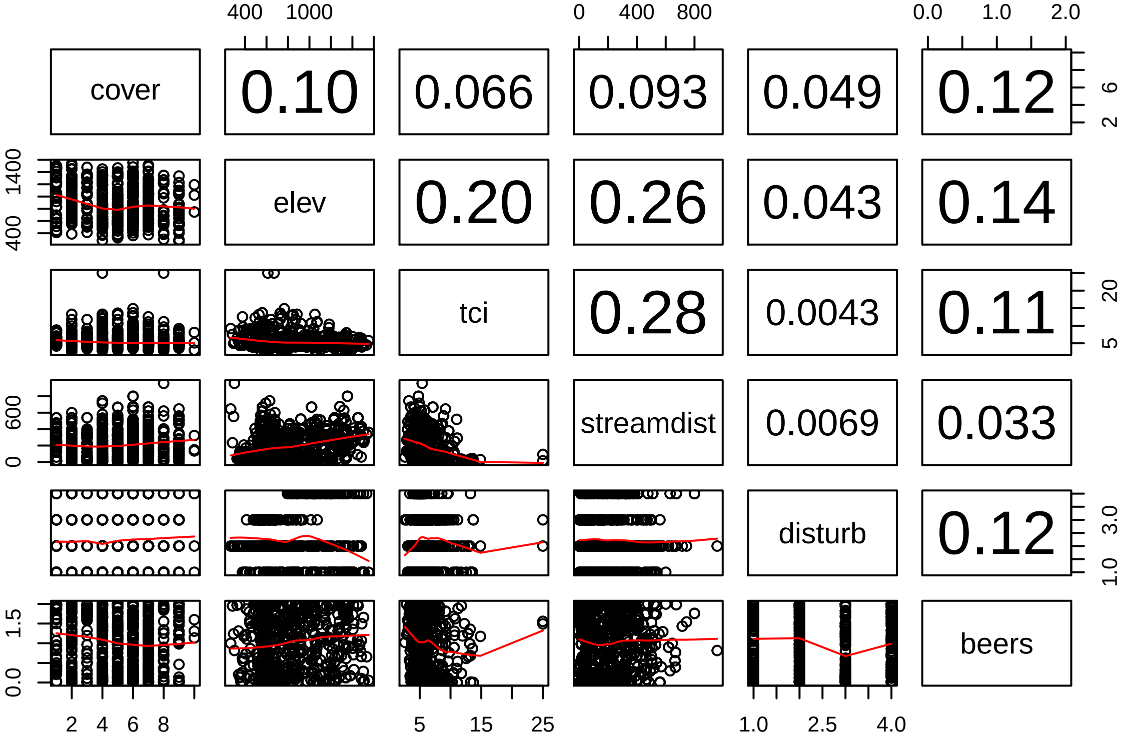
```
hist(acer$cover)
```



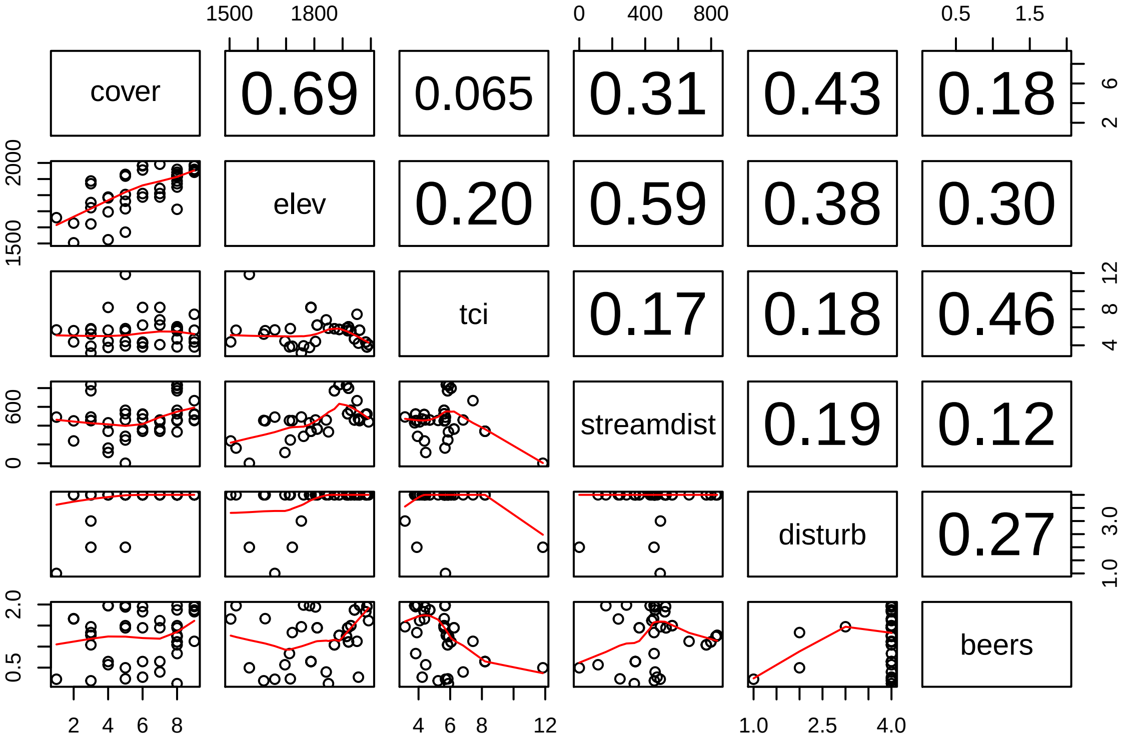
```
hist(abies$cover)
```



```
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor=3, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor))
        cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor)
}
pairs(acer[,c("cover", "elev", "tci", "streamdist", "disturb", "beers")],
      lower.panel = panel.smooth, upper.panel = panel.cor)
```



```
pairs(abies[,c("cover", "elev", "tci", "streamdist", "disturb", "beers")],
      lower.panel = panel.smooth, upper.panel = panel.cor)
```



```
lm.acer = lm(cover ~ elev + tci + streamdist + disturb + beers, data = acer)
summary(lm.acer)
```

```
##
## Call:
## lm(formula = cover ~ elev + tci + streamdist + disturb + beers,
##     data = acer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7073 -1.2446  0.3409  1.3575  5.2732
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.3502303  0.4564973  13.911  < 2e-16 ***
## elev          -0.0010108  0.0003161  -3.197  0.00145 **
## tci           -0.0627613  0.0351922  -1.783  0.07495 .
## streamdist     0.0012895  0.0004756   2.712  0.00686 **
## disturbLT-SEL  0.0829610  0.2166747   0.383  0.70192
## disturbSETTLE -0.1044556  0.2804213  -0.372  0.70963
## disturbVIRGIN  0.3088364  0.2518161   1.226  0.22044
## beers         -0.3269597  0.1089662  -3.001  0.00279 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.989 on 715 degrees of freedom
## Multiple R-squared:  0.04493,    Adjusted R-squared:  0.03558
## F-statistic: 4.805 on 7 and 715 DF,  p-value: 2.669e-05
```

```
lm.abies = lm(cover ~ elev + tci + streamdist + disturb + beers, data = abies)
summary(lm.abies)
```

```
##
## Call:
## lm(formula = cover ~ elev + tci + streamdist + disturb + beers,
##     data = abies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4630 -0.6472  0.0788  1.0872  3.8017
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -20.561173   4.271449  -4.814 2.65e-05 ***
## elev            0.012370   0.002523   4.903 2.02e-05 ***
## tci             0.287641   0.193467   1.487   0.1458
## streamdist     -0.001266   0.001585  -0.799   0.4296
## disturbLT-SEL   2.188367   2.097905   1.043   0.3038
## disturbSETTLE   1.527604   2.341471   0.652   0.5183
## disturbVIRGIN   3.025596   1.735921   1.743   0.0899 .
## beers           0.037551   0.500269   0.075   0.9406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.601 on 36 degrees of freedom
## Multiple R-squared:  0.5824, Adjusted R-squared:  0.5011
## F-statistic: 7.171 on 7 and 36 DF,  p-value: 2.215e-05
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.4.3
```

```
Anova(lm.acer, type = 3)
```

```
## Anova Table (Type III tests)
##
## Response: cover
##              Sum Sq  Df  F value    Pr(>F)
## (Intercept)  765.43   1 193.5096 < 2.2e-16 ***
## elev          40.44   1  10.2233  0.001448 **
## tci           12.58   1   3.1805  0.074947 .
## streamdist    29.09   1   7.3531  0.006856 **
## disturb        9.45   3   0.7962  0.496166
## beers         35.61   1   9.0034  0.002789 **
## Residuals   2828.21 715
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(lm.abies, type = 3)
```

```
## Anova Table (Type III tests)
##
## Response: cover
##             Sum Sq Df F value    Pr(>F)
## (Intercept) 59.401  1 23.1710 2.652e-05 ***
## elev        61.618  1 24.0358 2.022e-05 ***
## tci          5.667  1  2.2105    0.1458
## streamdist   1.636  1  0.6382    0.4296
## disturb     10.089  3  1.3118    0.2855
## beers        0.014  1  0.0056    0.9406
## Residuals   92.289 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-values generate by the Anova function are the same as those generated by the lm function.

```
library(MASS)
step.acer =  stepAIC(lm.acer)
```

```
## Start:  AIC=1002.17
## cover ~ elev + tci + streamdist + disturb + beers
##
##              Df Sum of Sq    RSS     AIC
## - disturb     3     9.449 2837.7  998.58
## <none>                    2828.2 1002.17
## - tci         1    12.581 2840.8 1003.37
## - streamdist  1    29.085 2857.3 1007.56
## - beers       1    35.613 2863.8 1009.21
## - elev        1    40.439 2868.7 1010.43
##
## Step:  AIC=998.58
## cover ~ elev + tci + streamdist + beers
##
##              Df Sum of Sq    RSS     AIC
## <none>                    2837.7  998.58
## - tci         1    14.370 2852.0 1000.23
## - streamdist  1    31.491 2869.2 1004.56
## - beers       1    35.515 2873.2 1005.57
## - elev        1    45.778 2883.4 1008.15
```

```
step.abies = stepAIC(lm.abies)
```

```
## Start:  AIC=48.59
## cover ~ elev + tci + streamdist + disturb + beers
##
##              Df Sum of Sq     RSS    AIC
## - beers       1     0.014  92.304 46.599
## - disturb     3    10.089 102.379 47.157
## - streamdist  1     1.636  93.926 47.366
## <none>                     92.289 48.593
## - tci         1     5.667  97.956 49.215
## - elev        1    61.618 153.908 69.095
##
## Step:  AIC=46.6
## cover ~ elev + tci + streamdist + disturb
##
##              Df Sum of Sq     RSS    AIC
## - streamdist  1     1.665  93.969 45.386
## - disturb     3    10.679 102.983 45.417
## <none>                     92.304 46.599
## - tci         1     6.745  99.049 47.703
## - elev        1    64.662 156.966 67.961
##
## Step:  AIC=45.39
## cover ~ elev + tci + disturb
##
##           Df Sum of Sq     RSS    AIC
## - disturb  3    12.021 105.990 44.683
## <none>                  93.969 45.386
## - tci      1     6.807 100.776 46.463
## - elev     1    78.687 172.656 70.153
##
## Step:  AIC=44.68
## cover ~ elev + tci
##
##        Df Sum of Sq    RSS    AIC
## <none>              105.99 44.683
## - tci   1     9.239 115.23 46.360
## - elev  1   114.046 220.04 74.822
```

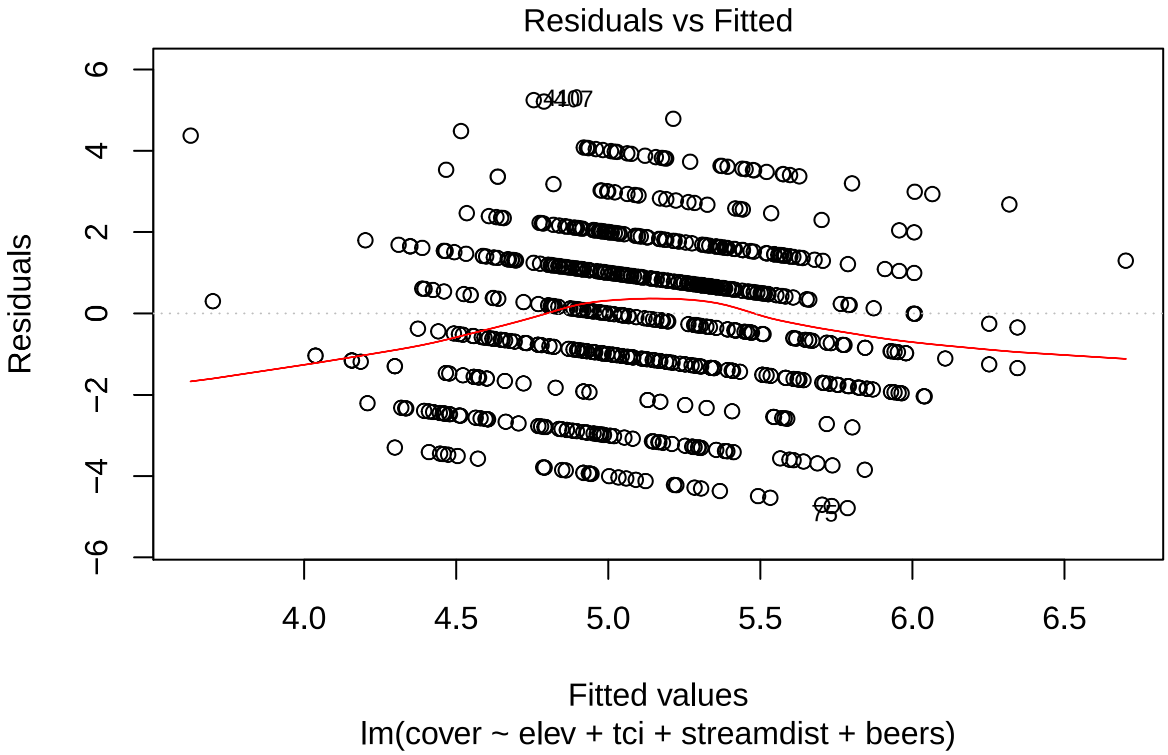
```
summary(step.acer)
```

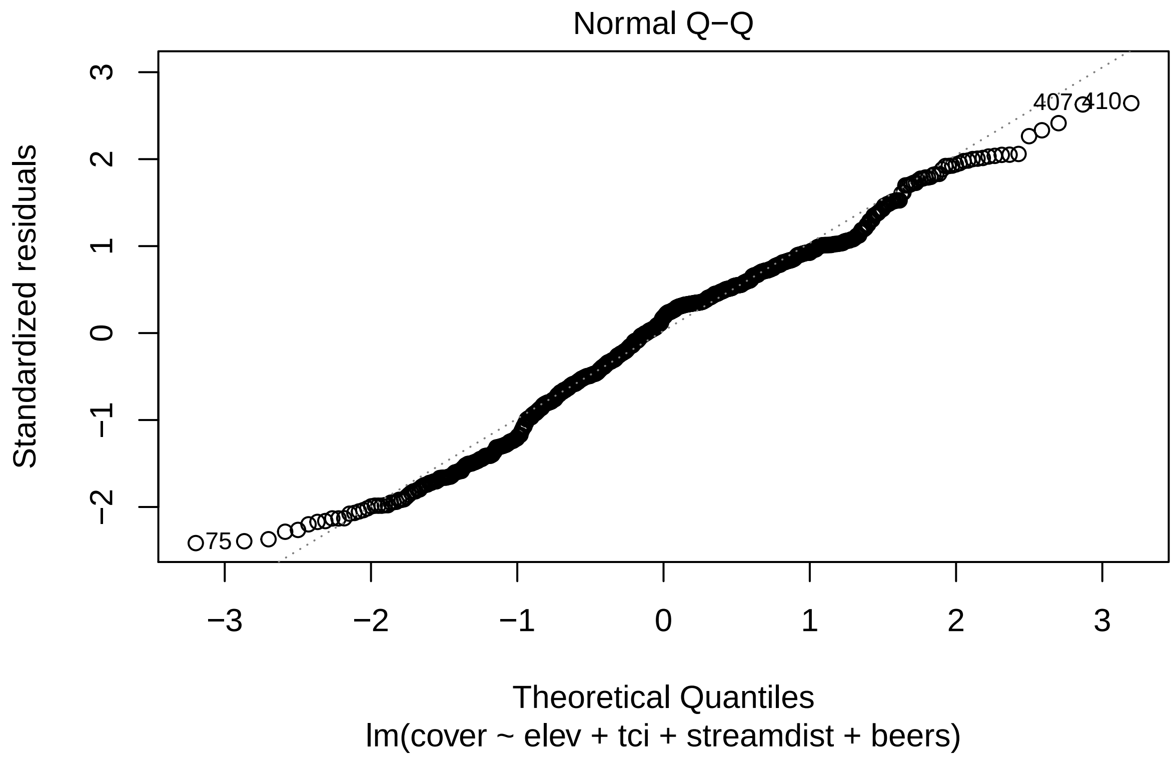
```
##
## Call:
## lm(formula = cover ~ elev + tci + streamdist + beers, data = acer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7869 -1.2983  0.3618  1.4014  5.2451
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.3218898  0.3604346  17.540  < 2e-16 ***
## elev        -0.0008868  0.0002606  -3.403 0.000703 ***
## tci         -0.0668631  0.0350647  -1.907 0.056939 .
## streamdist   0.0013256  0.0004696   2.823 0.004893 **
## beers       -0.3204370  0.1068951  -2.998 0.002814 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.988 on 718 degrees of freedom
## Multiple R-squared:  0.04174,    Adjusted R-squared:  0.0364
## F-statistic: 7.818 on 4 and 718 DF,  p-value: 3.603e-06
```

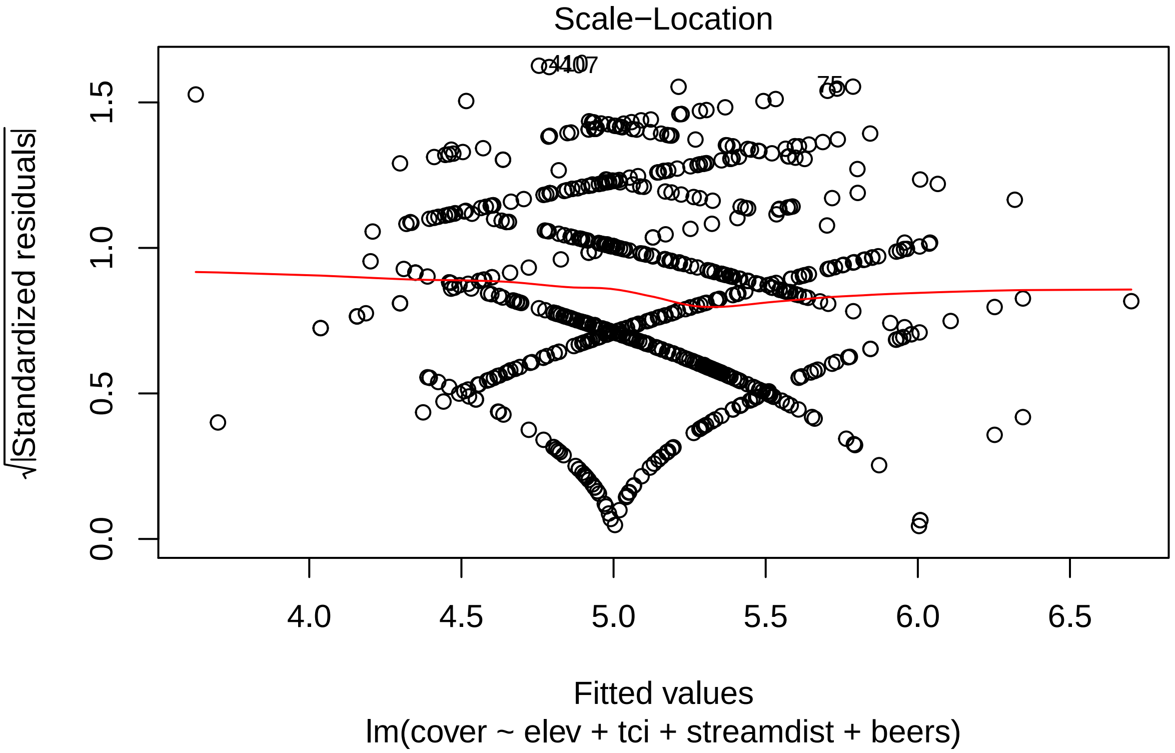
```
summary(step.abies)
```

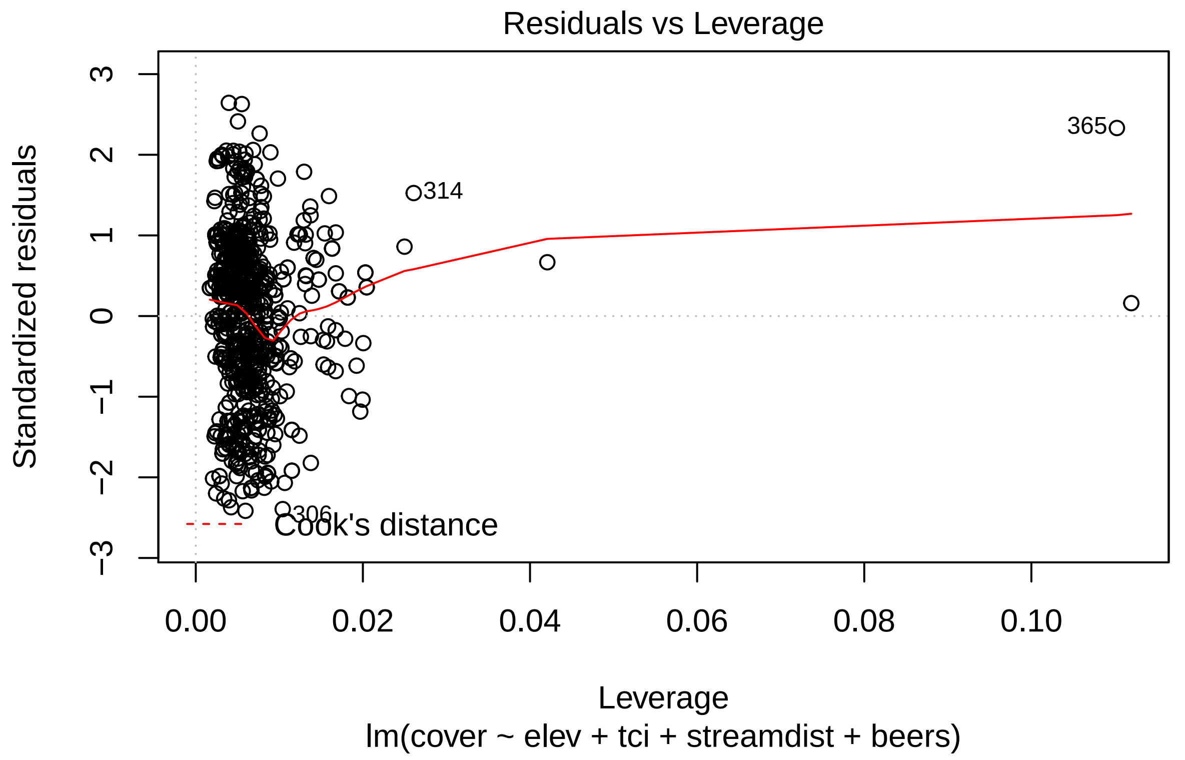
```
##
## Call:
## lm(formula = cover ~ elev + tci, data = abies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7819 -1.1346  0.3731  0.8880  4.0268
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18.789840   3.767380  -4.988 1.17e-05 ***
## elev          0.012616   0.001899   6.642 5.29e-08 ***
## tci           0.304539   0.161094   1.890   0.0658 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.608 on 41 degrees of freedom
## Multiple R-squared:  0.5204, Adjusted R-squared:  0.497
## F-statistic: 22.24 on 2 and 41 DF,  p-value: 2.876e-07
```

```
plot(step.acer)
```

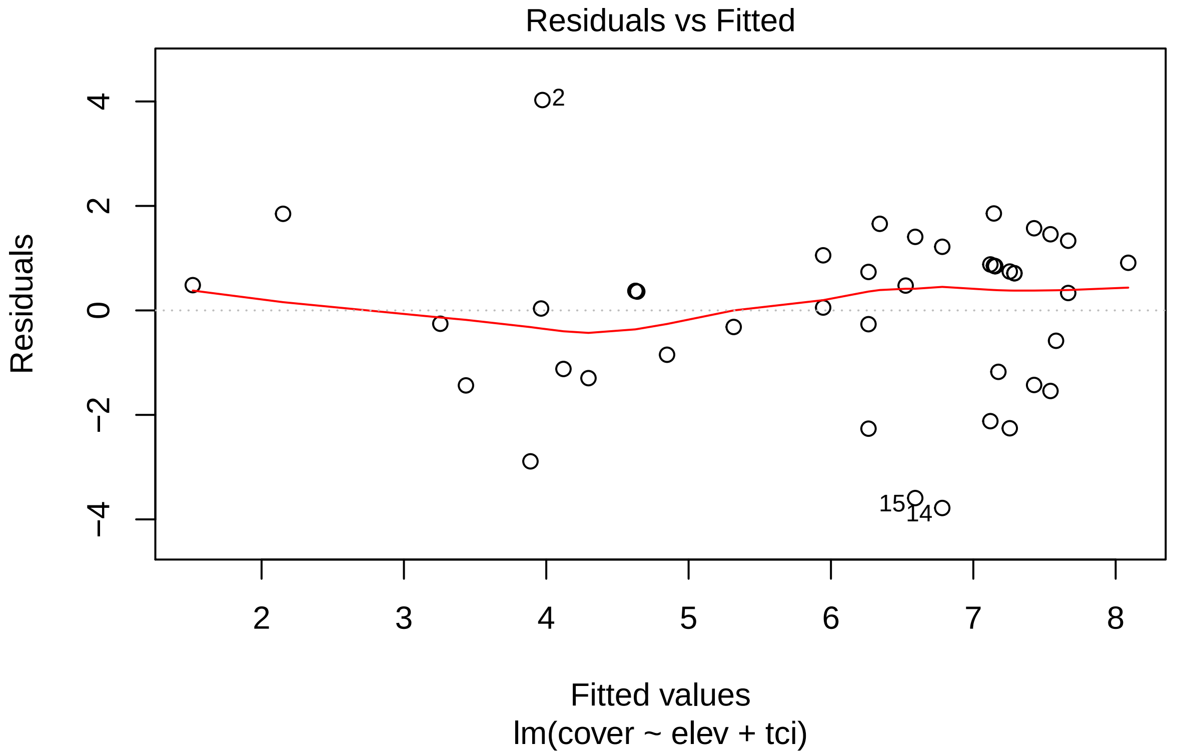


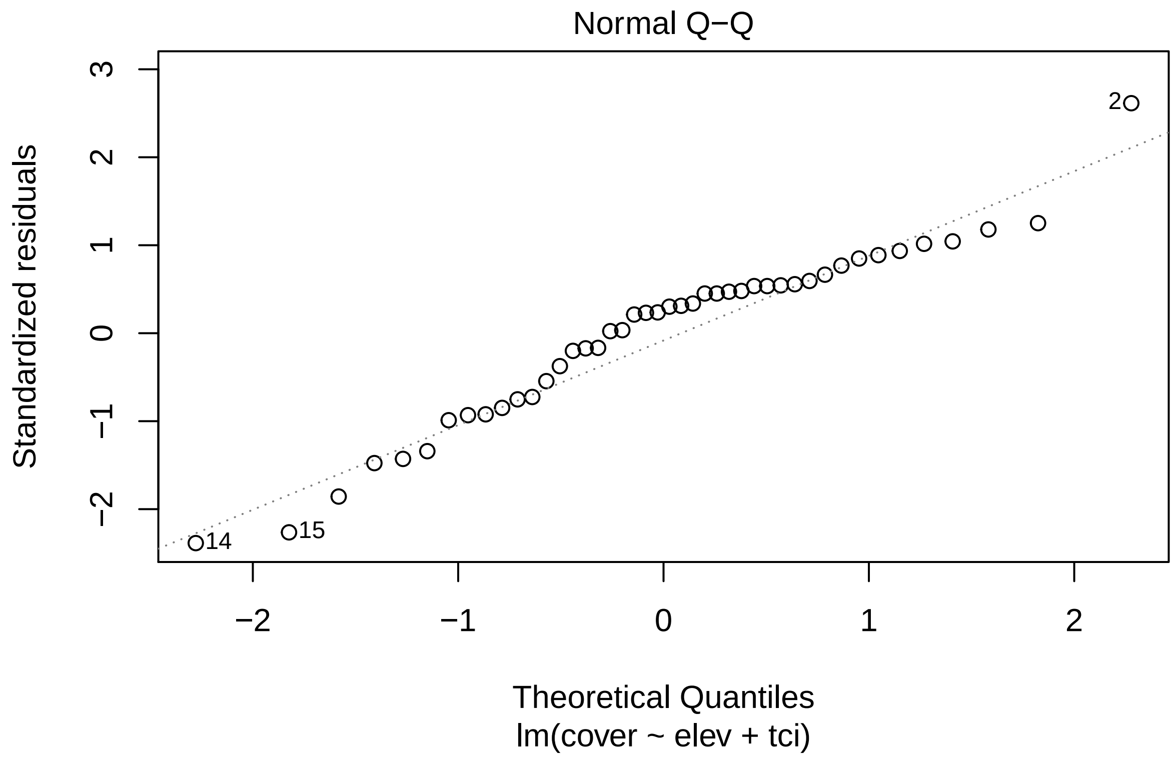

Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
407
410
75
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(cover ~ elev + tci + streamdist + beers)

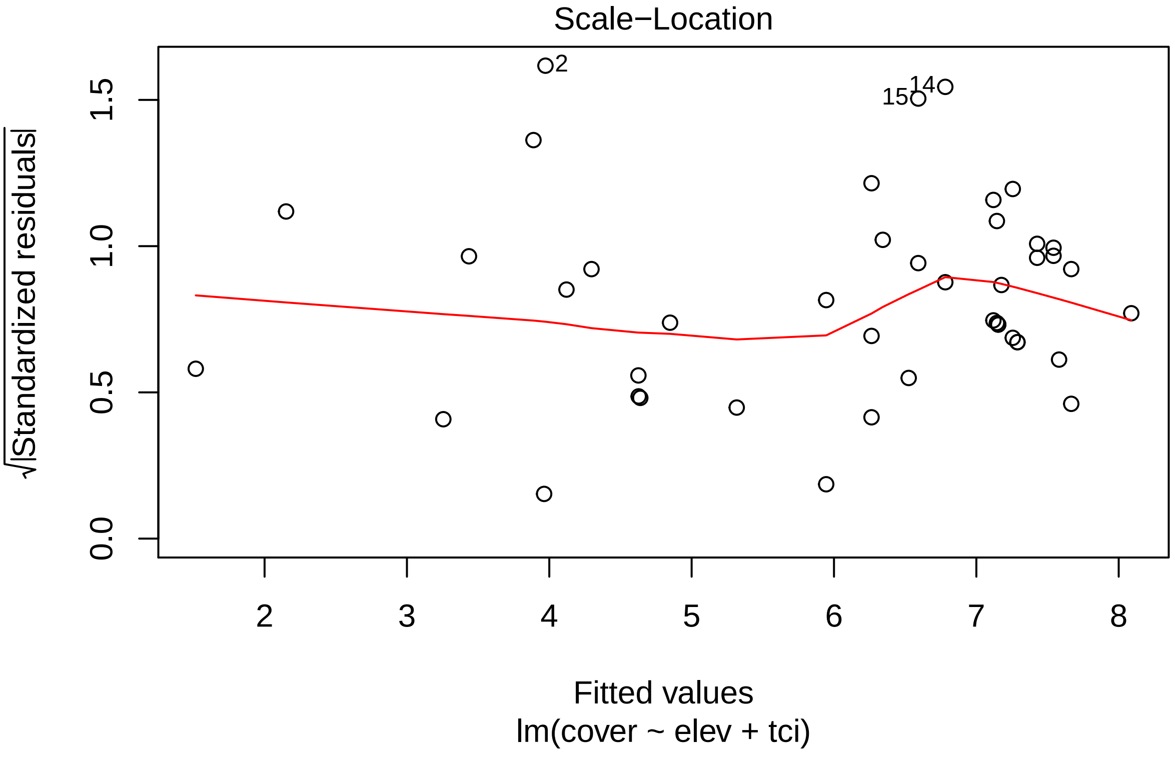

Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
410
407
75
4.0
4.5
5.0
5.5
6.0
6.5
Fitted values
lm(cover ~ elev + tci + streamdist + beers)

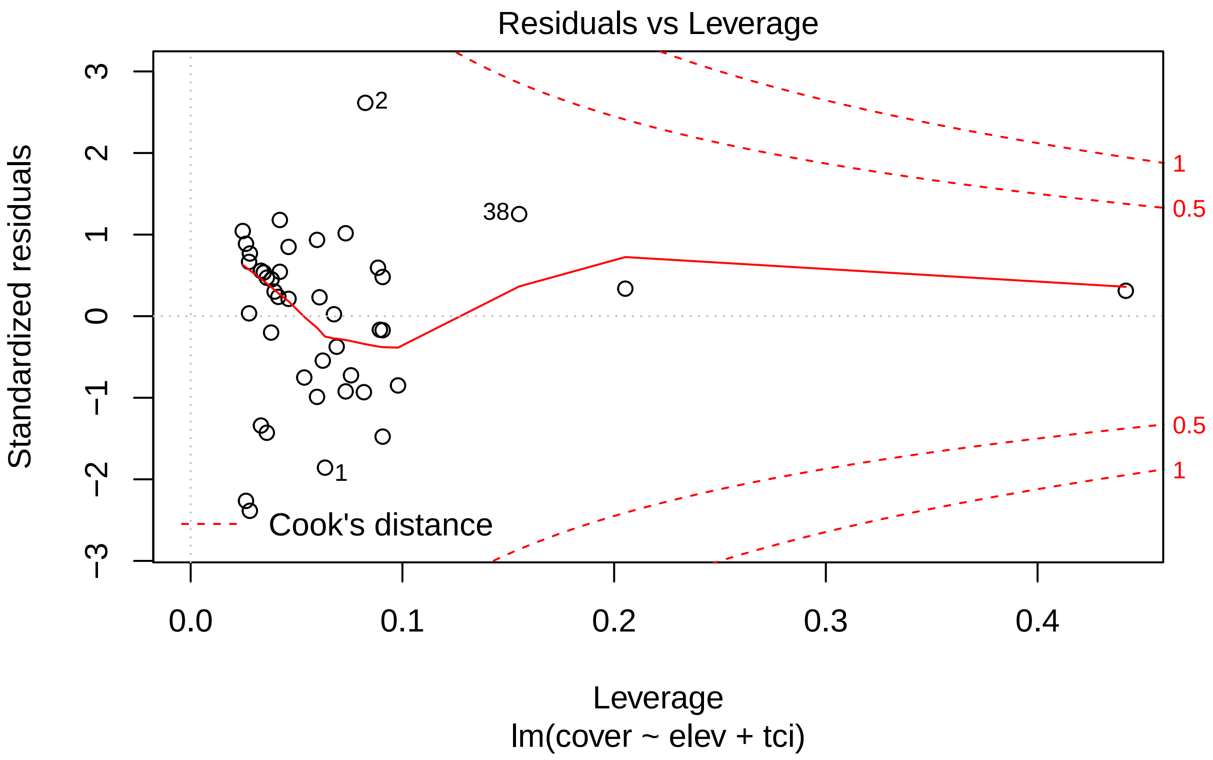


```
plot(step.abies)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(cover ~ elev + tci)
−2
−1
0
1
2
3
14
15
2


Scale−Location
√|Standardized residuals|
Fitted values
lm(cover ~ elev + tci)
0.0
0.5
1.0
1.5
2
3
4
5
6
7
8
15
14



The Acer rubrum model has a multiple r-squared value of 0.04174, indicating that only 4.174% of the variation in cover can be explained by the explanatory variables. The variables that were included in the stepwise regression, and thus the most important variables, are elev, tci, streamdist, and beers. Model diagnostics plots indicate that there are no major violations of the OLS assumptions.

The Abies fraseri model has a multiple r-squared value of 0.5204, indicating that 52.04% of the variation in cover can be explained by the explanatory variables. The variables that were included in the stepwise regression, and thus the most important variables, are elev and tci. Model diagnostics plots indicate that there are no major violatuins of the OLS assumptions.

Between the two models, the variance in Abies fraseri cover is much better explained by the data. This is possibly because it is a habitat specialist and there is less variation in the range of habitats for which the model must make predictions.

2. General Linear Model (GLM) with a Poisson error term

```
acer_glm = glm(cover ~ elev + tci + streamdist + beers , data = acer,
                family = 'poisson')
abies_glm = glm(cover ~ elev + tci, data = abies, family = 'poisson')

pseudo_r2 = function(glm_mod) {
     1 -  glm_mod$deviance / glm_mod$null.deviance
}

acer_r2 = pseudo_r2(acer_glm)
acer_r2
```

```
## [1] 0.03704802
```

```
abies_r2 = pseudo_r2(abies_glm)
abies_r2
```

```
## [1] 0.5140995
```

```
anova(step.acer, acer_glm)
```

```
## Analysis of Variance Table
##
## Model 1: cover ~ elev + tci + streamdist + beers
## Model 2: cover ~ elev + tci + streamdist + beers
##   Res.Df     RSS Df Sum of Sq F Pr(>F)
## 1    718 2837.66
## 2    718  625.28  0    2212.4
```

```
anova(step.abies, abies_glm)
```

```
## Analysis of Variance Table
##
## Model 1: cover ~ elev + tci
## Model 2: cover ~ elev + tci
##   Res.Df     RSS Df Sum of Sq F Pr(>F)
## 1     41 105.990
## 2     41  20.055  0    85.935
```

Changing the error distribution greatly reduced the residual sum of squares errors for both models. For the Acer model, it reduded from 2837.66 to 625.28, a difference of 2212.4. For the Abies model, it reduced from 105.99 to 20.055, a reduction of 85.935.

3. Plain english summary

The cover of Acer rubrum and Abies fraseri trees can be predicted using orinary least squares regression and more accurately predicted using a generalized linear model with a poisson error term, that is, changing the structure of the model so that it better fits the format of the data. The predictions for Abies fraseri, a habitat specialist with less variation in its predictor and response variables, are more accurate than those for Acer rubrum, a habitat generalist with more variation in its predictor and response variables. It is easier to build a model that predicts for a smaller range of data than a wider range.

4. Examine the behavior of the function step()

```
step.abies = stepAIC(lm.abies)
```

```
## Start:  AIC=48.59
## cover ~ elev + tci + streamdist + disturb + beers
##
##              Df Sum of Sq     RSS    AIC
## - beers       1     0.014  92.304 46.599
## - disturb     3    10.089 102.379 47.157
## - streamdist  1     1.636  93.926 47.366
## <none>                     92.289 48.593
## - tci         1     5.667  97.956 49.215
## - elev        1    61.618 153.908 69.095
##
## Step:  AIC=46.6
## cover ~ elev + tci + streamdist + disturb
##
##              Df Sum of Sq     RSS    AIC
## - streamdist  1     1.665  93.969 45.386
## - disturb     3    10.679 102.983 45.417
## <none>                     92.304 46.599
## - tci         1     6.745  99.049 47.703
## - elev        1    64.662 156.966 67.961
##
## Step:  AIC=45.39
## cover ~ elev + tci + disturb
##
##           Df Sum of Sq     RSS    AIC
## - disturb  3    12.021 105.990 44.683
## <none>                  93.969 45.386
## - tci      1     6.807 100.776 46.463
## - elev     1    78.687 172.656 70.153
##
## Step:  AIC=44.68
## cover ~ elev + tci
##
##        Df Sum of Sq    RSS    AIC
## <none>              105.99 44.683
## - tci   1     9.239 115.23 46.360
## - elev  1   114.046 220.04 74.822
```

The step.aic() function starts with the full model and reports the AIC. It then tests each of the models that result from removing a single variable and removing no variables and reports the AIC from each individual model. It chooses the model that had the biggest drop in AIC (lower is better) and repeats. Again, it tests each of the models that result from removing one of the single remaining variables and no variables and chooses the model that has the largest frop in AIC. This repeats until the model that results from removing no variables (the row) is chosen. This is the final model that is returned by the function.