Assignment #3

Sarah Kate Shore

February 2, 2016

```r
trees = read.csv('http://dmcglinn.github.io/quant_methods/data/treedata_subset.csv')
head(trees)
```

```
##         plotID  spcode       species cover elev       tci streamdist
## 1 ATBN-01-0403 ABIEFRA Abies fraseri     1 1660  5.701460      490.9
## 2 ATBN-01-0532 ABIEFRA Abies fraseri     8 1712  3.823586      454.0
## 3 ATBN-01-0533 ABIEFRA Abies fraseri     3 1722  3.893762      453.4
## 4 ATBN-01-0536 ABIEFRA Abies fraseri     3 1754  3.145527      492.5
## 5 FRID-01-0003 ABIEFRA Abies fraseri     5 1570 11.850000        0.0
## 6 PITT-01-0045 ABIEFRA Abies fraseri     2 1504  4.373741      237.1
##   disturb     beers
## 1 CORPLOG 0.2244286
## 2  VIRGIN 0.8340878
## 3  LT-SEL 1.3332586
## 4  SETTLE 1.4712484
## 5  LT-SEL 0.4961189
## 6  VIRGIN 1.6558421
```

```r
acer = subset(trees, subset= species == 'Acer rubrum',
              select = c('cover', 'tci', 'elev', 'beers', 'streamdist',
                         'disturb'))
abies = subset(trees, subset= species == 'Abies fraseri', select = c('cover', 'tci', 'elev', 'beers', '

mod_gen = lm(cover ~ . , data = acer)
mod_spe = lm(cover ~ . , data = abies)

library(car)
Anova(mod_gen, 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 ***
## tci           12.58   1   3.1805  0.074947 .
## elev          40.44   1  10.2233  0.001448 **
## beers         35.61   1   9.0034  0.002789 **
## streamdist    29.09   1   7.3531  0.006856 **
## disturb        9.45   3   0.7962  0.496166
## Residuals   2828.21 715
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(mod_spe, 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 ***
## tci          5.667  1  2.2105    0.1458
## elev        61.618  1 24.0358 2.022e-05 ***
## beers        0.014  1  0.0056    0.9406
## streamdist   1.636  1  0.6382    0.4296
## disturb     10.089  3  1.3118    0.2855
## Residuals   92.289 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mod_gen)
```

```
##
## Call:
## lm(formula = cover ~ ., 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 ***
## tci           -0.0627613  0.0351922  -1.783  0.07495 .
## elev          -0.0010108  0.0003161  -3.197  0.00145 **
## beers         -0.3269597  0.1089662  -3.001  0.00279 **
## 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
## ---
## 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
```

```r
summary(mod_spe)
```

```
##
## Call:
## lm(formula = cover ~ ., 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 ***
## tci             0.287641   0.193467   1.487   0.1458
## elev            0.012370   0.002523   4.903 2.02e-05 ***
## beers           0.037551   0.500269   0.075   0.9406
## 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 .
## ---
## 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
```

For each predictor variable, the Anova and summary p values are the same, although the Anova gives up to six decimal points and the summary value gives up to five.

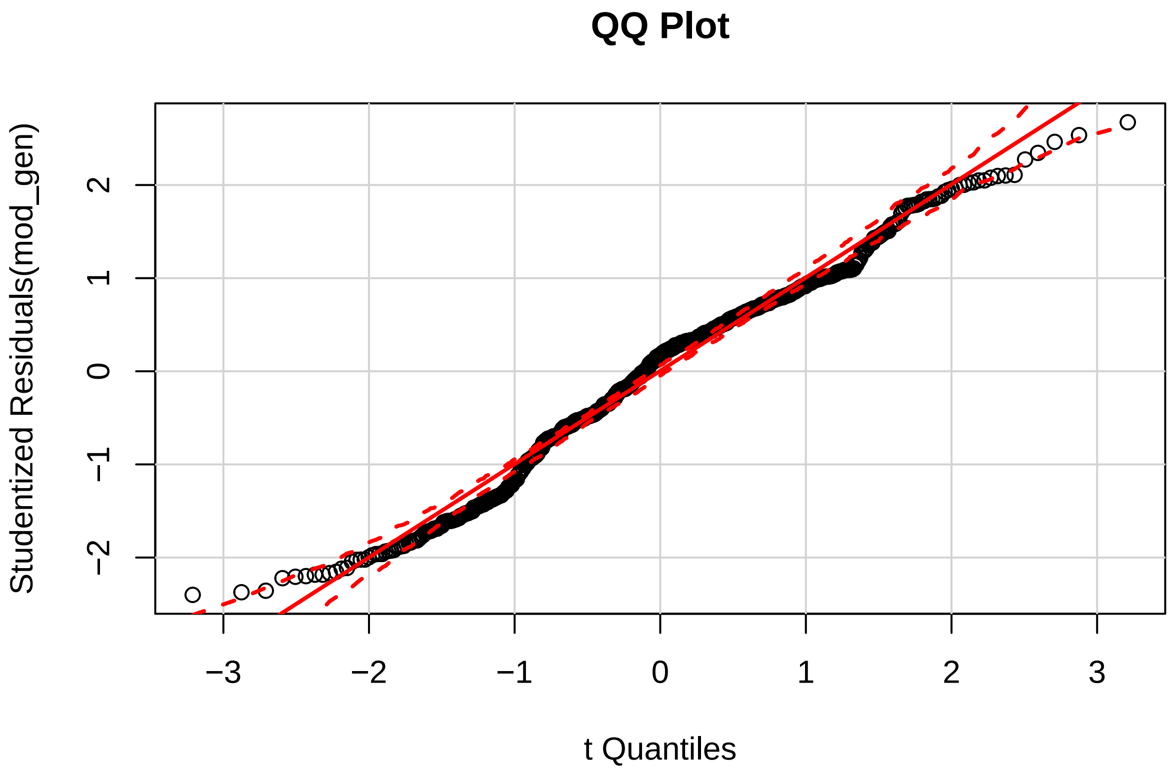
Model comments: The exploratory model for mod_gen (Acer rubrum) doesn't seem to explain cover very well. Although three variables (elev, beers, streamdist) were flagged as significant according to the p value, there is an adjusted R-squared value of only 0.04. Therefore, the model does not fit the data that well, although p-values are low.

For mod_spe (Abies fraseri), cover is explained a little better by one significant variable (elev) with p values less than 0.0001. Also, the adjusted R-squared for mod_spe is 0.5, which isn't great, but it's better than mod_gen.

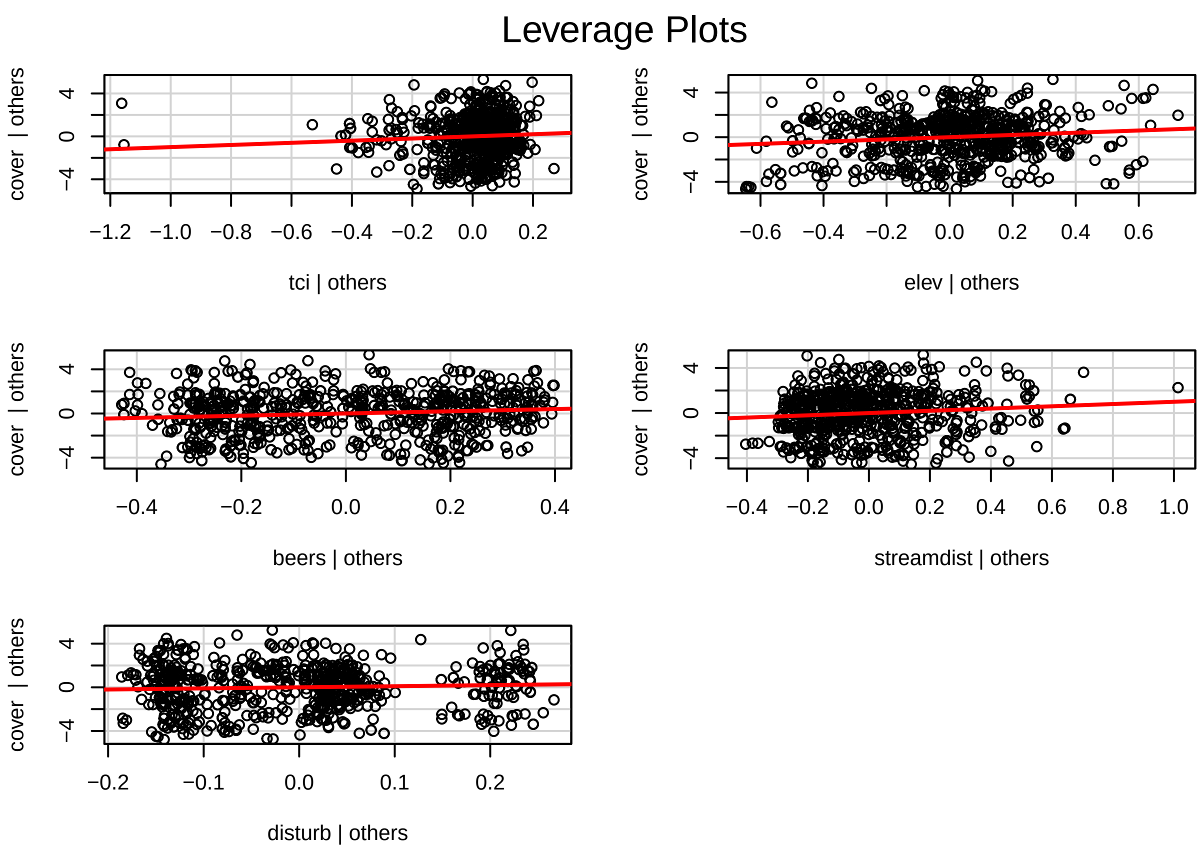
Variance can be explained for Abies fraseri better than for Acer rubrum, due to a maxium Anova F value of 24 for Abies fraseri compared to maximum Anova F value of 10 for Acer rubrum.

Diagnostic plots. . . (help taken from http://www.statmethods.net/stats/rdiagnostics.html for diagnostic ideas)

```
qqPlot(mod_gen, main="QQ Plot") #qq plot for studentized resid
```

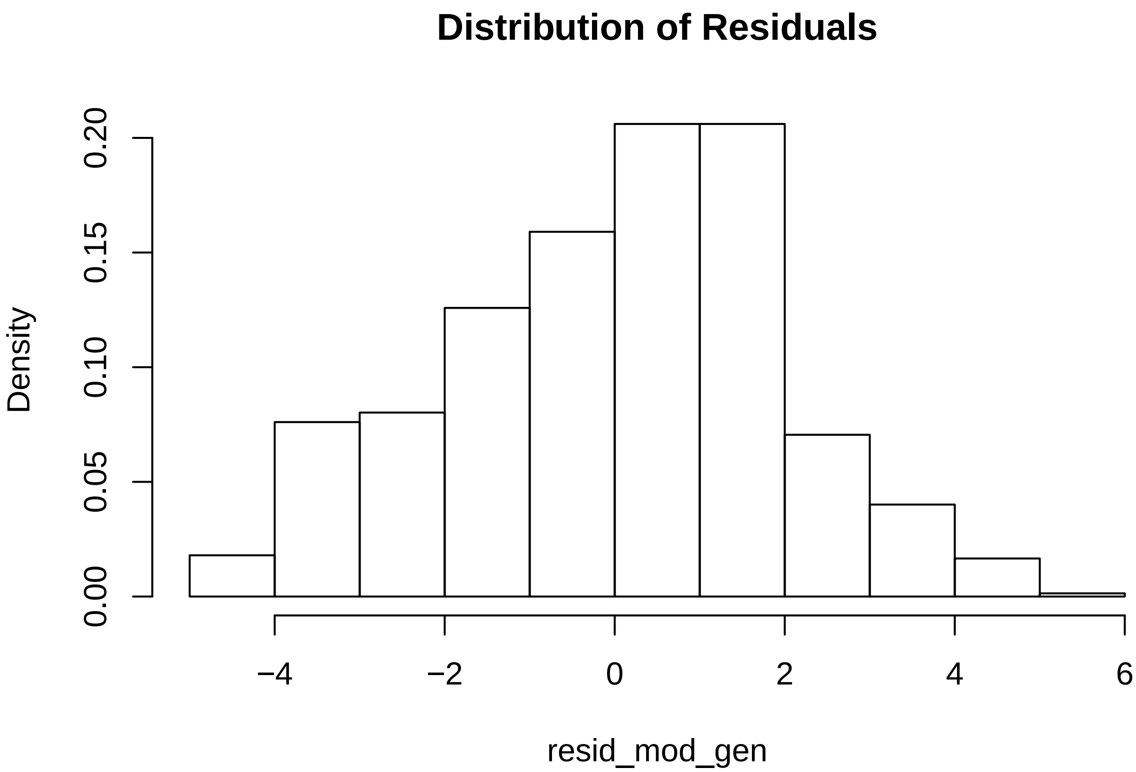



```r
leveragePlots(mod_gen)# leverage plots
```



#The QQ plot shows that the variance for mod_gen is homogenous on either side of the mean. However on leverage plot "cover vs. tci" there are two outliers that pull distribution down on the left.

```
resid_mod_gen = residuals(mod_gen)
hist(resid_mod_gen, freq=FALSE, main = "Distribution of Residuals")
```



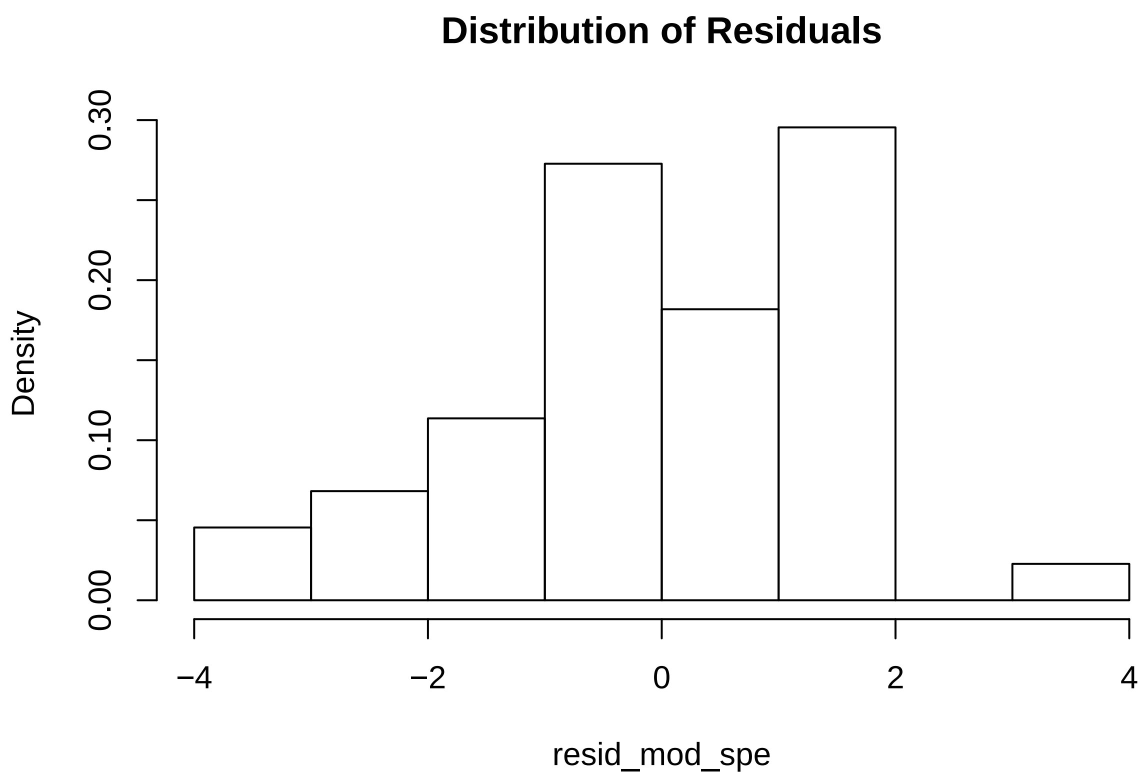
#This piece of code did not go through due to "variable lengths differ", however, plots were generated when code was executed…

qqPlot(mod_spe, main = "QQ Plot") #qq plot for studentized resid

leveragePlots(mod_spe) #leverage plots

This histogram shows that the residual errors for mod_gen are normally distributed for the most part, although there is a little bit of right skewing.

```
resid_mod_spe = residuals(mod_spe)
hist(resid_mod_spe, freq=FALSE, main = "Distribution of Residuals")
```



#This histogram shows that the data for Abies fraseri might not be normally distributed, violating an OLS assumption.

GLM Poisson models

```
acer = subset(trees, subset= species == 'Acer rubrum',
              select = c('cover', 'tci', 'elev', 'beers', 'streamdist',
                         'disturb'))
abies = subset(trees, subset= species == 'Abies fraseri', select = c('cover', 'tci', 'elev', 'beers', '

glm_gen = glm(cover ~ . , data = acer, family = 'poisson')
glm_spe = glm(cover ~ . , data = abies, family = 'poisson')

library(car)
Anova(glm_gen, type=3)
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: cover
##              LR Chisq Df Pr(>Chisq)
## tci            2.5877  1   0.107699
## elev           7.7744  1   0.005299 **
## beers          6.9611  1   0.008330 **
## streamdist     5.4866  1   0.019163 *
## disturb        1.9033  3   0.592714
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(glm_spe, type=3)
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: cover
##            LR Chisq Df Pr(>Chisq)
## tci          1.1830  1  0.2767545
## elev        11.3450  1  0.0007565 ***
## beers        0.0155  1  0.9008297
## streamdist   0.3059  1  0.5802166
## disturb      3.3953  3  0.3346007
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(glm_gen)
```

```
##
## Call:
## glm(formula = cover ~ ., family = "poisson", data = acer)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4282  -0.5903   0.1391   0.5786   2.1038
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.873e+00  1.023e-01  18.315  < 2e-16 ***
## tci           -1.297e-02  8.159e-03  -1.589  0.11202
## elev          -1.961e-04  7.047e-05  -2.783  0.00538 **
## beers         -6.391e-02  2.423e-02  -2.638  0.00834 **
## streamdist     2.428e-04  1.030e-04   2.357  0.01843 *
## disturbLT-SEL  1.840e-02  4.880e-02   0.377  0.70619
## disturbSETTLE -1.739e-02  6.253e-02  -0.278  0.78099
## disturbVIRGIN  6.311e-02  5.638e-02   1.119  0.26293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 649.34  on 722  degrees of freedom
## Residual deviance: 623.38  on 715  degrees of freedom
## AIC: 3101.8
##
## Number of Fisher Scoring iterations: 4
```

```r
summary(glm_spe)
```

```
##
## Call:
## glm(formula = cover ~ ., family = "poisson", data = abies)
##
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.47931  -0.35524   0.08027   0.36453   1.69535
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -4.1157009  1.5505526  -2.654  0.00795 **
## tci            0.0568868  0.0524222   1.085  0.27785
## elev           0.0023508  0.0007292   3.224  0.00126 **
## beers         -0.0165548  0.1326724  -0.125  0.90070
## streamdist    -0.0002186  0.0003969  -0.551  0.58176
## disturbLT-SEL  1.2440008  1.0827736   1.149  0.25060
## disturbSETTLE  1.0440232  1.1644892   0.897  0.36996
## disturbVIRGIN  1.4002993  1.0171140   1.377  0.16859
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 41.274  on 43  degrees of freedom
## Residual deviance: 16.126  on 36  degrees of freedom
## AIC: 189.3
##
## Number of Fisher Scoring iterations: 4
```

Pseudo R2

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

```
## [1] 0.03997917
```

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

```
## [1] 0.60931
```

The pseudo r2 value for the generalist model was only 0.04, which is a pretty bad fit for the data. However, the psuedo r2 for the specialist model was 0.61, which is much better.

Compare residual sums of squares

```r
anova(mod_gen, glm_gen)
```

```
## Analysis of Variance Table
##
## Model 1: cover ~ tci + elev + beers + streamdist + disturb
## Model 2: cover ~ tci + elev + beers + streamdist + disturb
##   Res.Df     RSS Df Sum of Sq F Pr(>F)
## 1    715 2828.21
## 2    715  623.38  0    2204.8
```

```r
anova(mod_spe, glm_spe)
```

```
## Analysis of Variance Table
##
## Model 1: cover ~ tci + elev + beers + streamdist + disturb
## Model 2: cover ~ tci + elev + beers + streamdist + disturb
##   Res.Df    RSS Df Sum of Sq F Pr(>F)
## 1     36 92.289
## 2     36 16.126  0    76.164
```

In both the generalist and specialist models, the glm Poisson distribution resulted in a much lower residual sum of squares. For the Acer rubrum species, the RSS was over four times greater in the ols model, and in the Abies fraseri species, the RSS was over twice as high in ols.

Summary of results: For the generalist species Acer rubrum, none of the explanitory variables seem to have a strong effect on cover. However, for the specialist species Abies fraseri, elevation could have a slight correlation to tree cover.

function step()

```r
step(mod_gen)
```

```
## Start:  AIC=1002.17
## cover ~ tci + elev + beers + streamdist + disturb
##
##              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 ~ tci + elev + beers + streamdist
##
##              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

##
## Call:
## lm(formula = cover ~ tci + elev + beers + streamdist, data = acer)
##
## Coefficients:
## (Intercept)          tci         elev        beers   streamdist
##   6.3218898   -0.0668631   -0.0008868   -0.3204370    0.0013256
```

```r
step(mod_spe)
```

```
## Start:  AIC=48.59
## cover ~ tci + elev + beers + streamdist + disturb
##
##              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 ~ tci + elev + 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 ~ tci + elev + 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 ~ tci + elev
##
##        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

##
## Call:
## lm(formula = cover ~ tci + elev, data = abies)
##
## Coefficients:
## (Intercept)          tci         elev
##   -18.78984      0.30454      0.01262
```

```r
step(glm_gen)
```

```
## Start:  AIC=3101.77
## cover ~ tci + elev + beers + streamdist + disturb
##
##              Df Deviance    AIC
## - disturb     3   625.28 3097.7
## <none>            623.38 3101.8
## - tci         1   625.97 3102.4
## - streamdist  1   628.87 3105.2
## - beers       1   630.34 3106.7
## - elev        1   631.16 3107.5
##
## Step:  AIC=3097.67
## cover ~ tci + elev + beers + streamdist
##
##              Df Deviance    AIC
## <none>            625.28 3097.7
## - tci         1   628.24 3098.6
## - streamdist  1   631.22 3101.6
## - beers       1   632.24 3102.6
## - elev        1   634.11 3104.5

##
## Call:  glm(formula = cover ~ tci + elev + beers + streamdist, family = "poisson",
##     data = acer)
##
## Coefficients:
## (Intercept)          tci         elev        beers   streamdist
```

```
##   1.8700348   -0.0138226   -0.0001719   -0.0626543    0.0002500
##
## Degrees of Freedom: 722 Total (i.e. Null);  718 Residual
## Null Deviance:       649.3
## Residual Deviance: 625.3     AIC: 3098
```

```r
step(glm_spe)
```

```
## Start:  AIC=189.3
## cover ~ tci + elev + beers + streamdist + disturb
##
##              Df Deviance    AIC
## - disturb     3   19.521 186.70
## - beers       1   16.141 187.32
## - streamdist  1   16.431 187.61
## - tci         1   17.308 188.49
## <none>            16.125 189.30
## - elev        1   27.471 198.65
##
## Step:  AIC=186.7
## cover ~ tci + elev + beers + streamdist
##
##              Df Deviance    AIC
## - beers       1   19.533 184.71
## - streamdist  1   20.014 185.19
## - tci         1   21.459 186.64
## <none>            19.521 186.70
## - elev        1   35.334 200.51
##
## Step:  AIC=184.71
## cover ~ tci + elev + streamdist
##
##              Df Deviance    AIC
## - streamdist  1   20.055 183.23
## <none>            19.533 184.71
## - tci         1   21.731 184.91
## - elev        1   37.364 200.54
##
## Step:  AIC=183.23
## cover ~ tci + elev
##
##        Df Deviance    AIC
## <none>      20.055 183.23
## - tci   1   22.180 183.36
## - elev  1   41.120 202.30
```

```
##
## Call:  glm(formula = cover ~ tci + elev, family = "poisson", data = abies)
##
## Coefficients:
## (Intercept)          tci         elev
##   -3.137624     0.065410     0.002469
##
```

```
## Degrees of Freedom: 43 Total (i.e. Null);  41 Residual
## Null Deviance:       41.27
## Residual Deviance: 20.06     AIC: 183.2
```

According to AIC values, of which I think lower is better, step() has chosen only tci and elev to be model predictors with the lowest AIC for the Abies fraseri (mod_spe and glm_spe) models.