Exercise 1: Use the Highway data involved in probem 10.2. Use the response $log(rate \times len)$ and treat lwid as the focal regressor. Test the significance of lwid in explaining the response. Use quidelines form Section 2 in the lecture videos to d determine which of the other regressors (adt, trks, lane, acpt, sigs, itg, slim, shld, and htype) to test lwid in the presence of . Assume that scientific considerations dictate that acpt and slim be included in the model that test lwid. Interpret the results of your test.

Solution:
Given that our model must included acpt and slim, let's first test the significance of lwid in the presence of those predictors as the first guidelines in section 2 states. To do so we can simply fit the model , and the model summary will give us a significance test for the lwid predictor. Doing so we get a p-value of .060234 and lwid is insignificant on the $\alpha = .05$ level.
Following the section two guidelines, we now want to test lwid in the presence of moderately to low correlated predictors. Testing the predictors we get that they all exhibit low correlation with lwid, so now we test the significance of lwid in the model that includes all other predictors. Doing so we get a p-value of 0.11658 so lwid is not a significant predictor. Since there were now high correlation predictors we would stop here and likely conclude tht lwid should not be included in the model.

Code:

```
> df <- Highway
> Step1Model <- lm(log(I(rate*len)) ~ acpt + slim + lwid, data = df)

Call:
lm(formula = log(I(rate * len)) ~ acpt + slim + lwid, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.07165 -0.23204 -0.09719  0.35883  1.00122

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.826810   2.231625   3.955 0.000355 ***
acpt         0.002199   0.011378   0.193 0.847848
slim        -0.022473   0.018200  -1.235 0.225123
lwid        -0.332115   0.171027  -1.942 0.060234 .

Residual standard error: 0.478 on 35 degrees of freedom
Multiple R-squared:  0.1872,    Adjusted R-squared:  0.1176
F-statistic: 2.688 on 3 and 35 DF,  p-value: 0.06135

-----------------------------------------------------------------
## Testing Correlations
> dfQuantPred = subset(df, select = -c(htype, rate, len))
> round(cor(dfQuantPred), 2)
      adt  trks  lane  acpt  sigs   itg  slim  lwid  shld
```

```
adt    1.00 -0.10  0.82 -0.22  0.15  0.90  0.24  0.13  0.46
trks  -0.10  1.00 -0.15 -0.36 -0.45 -0.07  0.30 -0.16  0.01
lane   0.82 -0.15  1.00 -0.21  0.25  0.70  0.26  0.10  0.48
acpt  -0.22 -0.36 -0.21  1.00  0.50 -0.20 -0.68 -0.04 -0.42
sigs   0.15 -0.45  0.25  0.50  1.00  0.07 -0.41  0.04 -0.13
itg    0.90 -0.07  0.70 -0.20  0.07  1.00  0.24  0.10  0.38
slim   0.24  0.30  0.26 -0.68 -0.41  0.24  1.00  0.10  0.69
lwid   0.13 -0.16  0.10 -0.04  0.04  0.10  0.10  1.00 -0.04
shld   0.46  0.01  0.48 -0.42 -0.13  0.38  0.69 -0.04  1.00

## Testing categoreical data
> cor.test(df$lwid, unclass(df$htype))

        Pearson's product-moment correlation

data:  df$lwid and unclass(df$htype)
t = -1.2197, df = 37, p-value = 0.2303
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4822064  0.1267803
sample estimates:
       cor
-0.1966011

___________________________________________________________

## Fitting model and testing significance.
> Step2Model <- lm(log(I(rate*len)) ~. , data = df)
> summary(Step2Model)

Call:
lm(formula = log(I(rate * len)) ~ ., data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.77134 -0.27683 -0.04212  0.24758  0.83292

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.732944   2.647258   3.299  0.00282 **
adt          0.008077   0.013375   0.604  0.55116
trks         0.004388   0.043395   0.101  0.92023
lane        -0.057867   0.111439  -0.519  0.60796
acpt         0.001278   0.016804   0.076  0.93994
sigs         0.101129   0.207317   0.488  0.62978
itg         -0.566489   0.504865  -1.122  0.27209
slim         0.003586   0.031910   0.112  0.91140
lwid        -0.369487   0.227606  -1.623  0.11658
shld        -0.076419   0.063168  -1.210  0.23725
htypefai     0.095900   0.681898   0.141  0.88924
htypepa     -0.332987   0.436656  -0.763  0.45257
htypema     -0.292539   0.385382  -0.759  0.45463

Residual standard error: 0.4734 on 26 degrees of freedom
Multiple R-squared:  0.4078,    Adjusted R-squared:  0.1344
```

```
F-statistic: 1.492 on 12 and 26 DF,  p-value: 0.1901
```

Exercise 2: Using these 'data' with a response Y and three regressors X_1, X_2 and X_3 from Mantel, apply the forward selection and backward elimination algorithms, using AIC as a criterion function. Also, find AIC and BIC for all possible models and compare results. Which appear to be the active regressors.

Solution:
Given that the data is very small we can just use the dredge() command from the MuMLN package we can quickly compute all possible models and their AIC and BIC (I recognize the point of stepwise regression is to avoid this). Performing forward substitution we get the model $lm(Y \sim X_3)$ which just fits X_3 since, it has an AIC which is lower than the null and the lowest compared to all other single predictor models. The resulting possible models which include X_3 give higher AIC so we stick with $lm(Y \sim X_3)$.

Backward elimination gives us that the model $lm(Y \sim X_1 + x_2)$ is the best. Note that we start with the full model, and removing X_3 gives us the lowest AIC of all models so we stop there and stick with $lm(Y \sim X_1 + x_2)$.

Note that we can also compute the BIC using the dredge() command and we still would get the same models. From our test it seems as though X_1 and X_2 are the most active regressors.

Code:

```
> AllModels <- dredge(lm(Y~., data = df), rank = 'AIC')
Fixed term is "(Intercept)"
> AllModels
Global model call: lm(formula = Y ~ ., data = df)
---
Model selection table
    (Intrc)         X1        X2        X3 df  logLik   AIC  delta weight
4 -1000.0000  1.0000000  1.0000000           4 139.780 -271.6   0.00  0.729
8 -1000.0000  1.0000000  1.0000000 1.330e-14 5 139.789 -269.6   1.98  0.271
5     0.7975                       6.947e-01 3  -4.940   15.9 287.44  0.000
7     0.1187             0.0004441 7.314e-01 4  -4.861   17.7 289.28  0.000
6     0.5663 -0.0004382            7.312e-01 4  -4.863   17.7 289.29  0.000
2     6.6460  0.0036840                      3  -9.702   25.4 296.96  0.000
3    10.3500            -0.0036750           3  -9.721   25.4 297.00  0.000
1     7.8000                                 2 -10.888   25.8 297.34  0.000
Models ranked by AIC(x)


-----------------------------------------------------------------------
> AllModels <- dredge(lm(Y~., data = df), rank = 'BIC')
Fixed term is "(Intercept)"
> AllModels
Global model call: lm(formula = Y ~ ., data = df)
---
Model selection table
    (Intrc)         X1        X2        X3 df  logLik   BIC  delta weight
```

```
4 −1000.0000  1.0000000  1.0000000            4 139.780 −273.1   0.00  0.689
8 −1000.0000  1.0000000  1.0000000 1.330e−14  5 139.789 −271.5   1.59  0.311
5     0.7975                       6.947e−01  3  −4.940   14.7 287.83  0.000
7     0.1187             0.0004441 7.314e−01  4  −4.861   16.2 289.28  0.000
6     0.5663 −0.0004382            7.312e−01  4  −4.863   16.2 289.29  0.000
2     6.6460  0.0036840                       3  −9.702   24.2 297.36  0.000
3    10.3500            −0.0036750            3  −9.721   24.3 297.39  0.000
1     7.8000                                  2 −10.888   25.0 298.12  0.000
Models ranked by BIC(x)
```

Exercise 3: Use the galapagos data described in problem 10.6. Regard NS as the response and Area, Anear, Dist, Dist SC, and Elevation as the possible regressors. Assume Elevation equals 80m for Baltra, 10m for Coamano, 38 m for Daphne Major, 71m for Eden, 23m for Las Plazas, and 28m for Seymour. Fit a linear model with LASSO with three values of λ : .3, .2, and .1. Report the regressors your three models admit and compare their coefficient estimates.

Solution:
Filling the NA values and fitting teh lasso models in r we get,
Code:

```
FillingNA <- c( 80, 10, 38, 71, 23, 28)
> for(i in 1:nrow(df)){
+   count = 1
+   if(is.na(df$Elevation[i]) == TRUE){
+     df$Elevation[i] = FillingNA[count]
+     count = count + 1
+   }
+ }
-----------------------------------------------------------

> X <- model.matrix(lm(NS ~ Area + Anear + Dist +
                            DistSC + Elevation  ,data= df))
> Lasso <- glmnet(X, df$NS, alpha=1, lambda=0.1)
> Lasso$beta
6 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept)   .
Area         -0.02544071
Anear        -0.07584734
Dist         -0.05625126
DistSC       -0.28585350
Elevation     0.31889555
> Lasso <- glmnet(X, df$NS, alpha=1, lambda=0.2)
> Lasso$beta
6 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept)   .
Area         -0.02460915
Anear        -0.07528838
Dist         -0.03957100
DistSC       -0.28577752
Elevation     0.31677628
> Lasso <- glmnet(X, df$NS, alpha=1, lambda=0.3)
> Lasso$beta
6 x 1 sparse Matrix of class "dgCMatrix"
                    s0
(Intercept)   .
Area         -0.02377835
Anear        -0.07472995
Dist         -0.02290415
DistSC       -0.28570047
```

```
Elevation      0.31465877
```

It seems as though all lasso models reported the same regressors for each λ level. There is also very little discrepancy in the size of each regressor coefficient across each model.