SDA- Assignment 7

Jose Chacon (2699643)

7.1: Photo Count vs. Observer Count

In the first exercise of this assignment, we are given a data frame containing the exact number of geese in a photo as well as the estimate of the number of geese given by two experienced observers. We are asked to perform some statistical analyses regarding these variables.

b) Linear regression

After creating some scatter plots of the data points, we suspect that there could be a linear relationship between the real number of geese and the estimation of an observer. Note: the two observers are going to be tested and analyzed separately. Therefore, we perform a linear regression for each observer. Additionally, an hypothesis test was performed to test if $\beta_1 = 0$ against $\beta_1 ! = 0$ with significance level 0.05.

We denoted as model 1 (model 2) to the model based on the observations of the observer 1 (observer 2). The results include the intercept, the β_1, the p-value of the F-test, conclusion of the test and the multiple R-squared value. Note that we can include more statistics, but for now these are the most relevant. The results of each model are the following:

Stats	**Values**
p-value	1.54×10^{-14}
Reject H_0?	True
Mult. R-squared	0.75
Residual St. Error	44.41

Figure 1: Model 1 Statistics

Stats	**Values**
p-value	2.2×10^{-16}
Reject H_0?	True
Mult. R-squared	0.85
Residual St. Error	33.87

Figure 2: Model 2 Statistics

From these results, we can start to understand the relationship between the actual number of geese and the estimates from the observers. We know that the 85% of the model 2 can be explained with the observer 2 variable (in model 1 is 75%, which is significantly lower). In both cases, we reject the null hypothesis that $\beta_1 = 0$. Additionally, the coefficients for each model were computed. So the regression formula for these models are:

- Model 1: $Y = 26.65 + 0.88x(+error)$
- Model 2: $Y = 16.16 + 0.77x(+error)$

c) Residuals against Y

In the second exercise, we would like to investigate the residuals of the model. For each model, we have plotted the residuals against Y. The resulting plots are presented below.

These plots can give us insights about the model's assumptions. It can help us understand at least two main assumptions: the linearity of the model and the homoscedasticity of the model.

The linearity is reflected in a plot (residuals vs y), if the plots shows a random scatter of points around a horizontal line with no discernible pattern. On the other hand, when there is a clear pattern or curve, it indicates that the model is not linear. For model 1, we cannot conclude that the linearity assumption is met, because an increasing trend is detectable in the plot. On the other hand, model 2 has more spread and more points across the horizontal line. Whether model 2 follows the linearity assumption or not is debatable. However, an improvement from model 1 is obvious.

Furthermore, this kind of plot can help us see if homoscedasticity assumption is met. This would be the case when the spread of the points is not the same across the predicted values. Again, in model 1,

we can see that that is not the case. We can then conclude that there is no homoscedasticity in model 1. However, in model 2, we can see that the spread is much more consistent (not perfect, but much better).

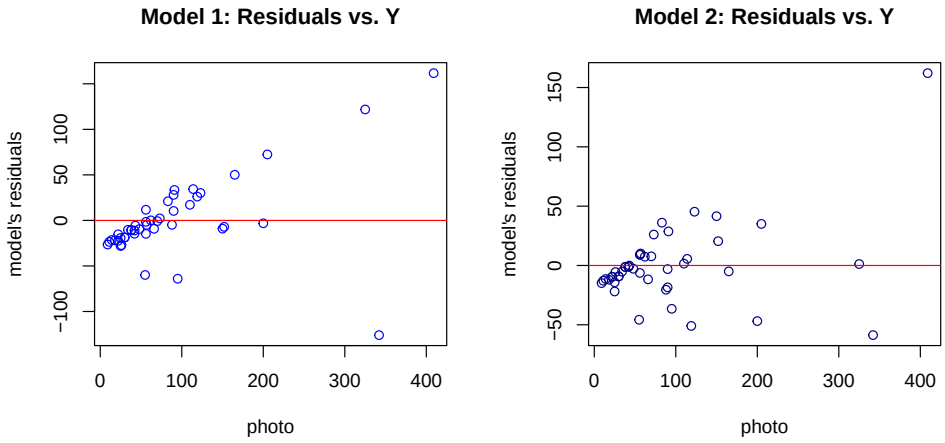

Figure 3: Residuals vs. Y-values

d) Normality of the errors

The next step of the analysis is to investigate graphically the normality of the errors. One of the most common and very successful method is to create a normal QQ plot of the residuals of the model. If the errors of the model are indeed normal, then a straight diagonal line should be visible. The QQ-plot of each model is shown below.

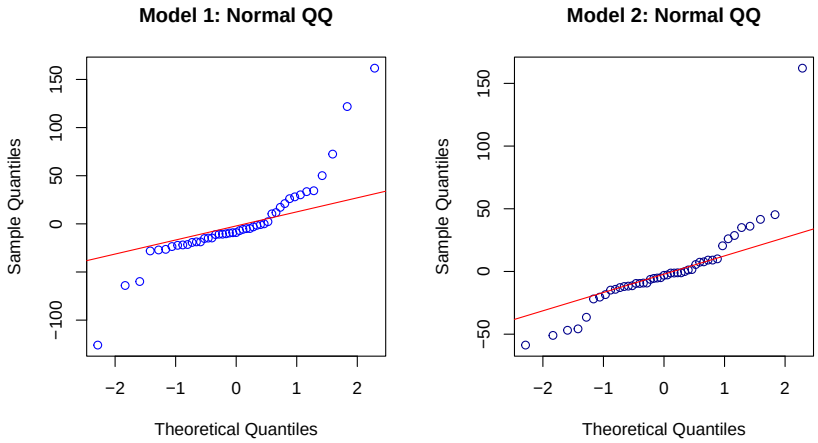

Figure 4: QQ-plots for Model 1 and Model 2

Additionally, we have used the *lm.norm.test* function (bootstrap method in combination with the Kolmogorov Smirnov test) to test the normality of the errors. After the analysis of the 2 bootstrap samples (with B=1000 observations), we conclude that for the two model there is no evidence against normality using this method.

e) Log-Transformation of the data

For this part, we will repeat the previously described procedure/analysis to the logarithmic transformation of the data points. We start by log-transforming all the data points given in order to create the new models. The model created with the log-data of the observer 1 is denoted as model 3; and consequently the model created with the log-data of observer 2 is denoted as model 4.

1. Scatter plot

First, we start by plotting the new set of points in a graph. The result is two scatter plots. In these plots, we can see that there is a linear relationship between the x and y values in both plots. Both plots are displayed in the appendix.

From these plots, we can see that most point do follow the straight diagonal line it should follow when normal. However, due to the variations of many of the points in both QQ-plots it is not clear that the

2. Linear Regression

Now that the relationship between the variables is clear, we formally create the models. The most relevant information about the models can be found on the tables below.

Stats	Values
p-value	$< 2.2 \times 10^{-16}$
Reject H_0?	True
Mult. R-squared	0.87
Residual St. Error	0.33

Figure 5: Model 3 Statistics

Stats	Values
p-value	$< 2.2 \times 10^{-16}$
Reject H_0?	True
Mult. R-squared	0.91
Residual St. Error	0.28

Figure 6: Model 4 Statistics

When comparing the results of these two models, we can see again that the model based on the second observer is better in terms of R^2. While with model 3 has a high R^2 equal to 0.87, the R^2 of model 4 is even higher (0.91).

The resulting formulas for each model are:

- Model 3: $Y = 0.65 + 0.9x + e$
- Model 4: $Y = 0.57 + 0.87x + e$

3. Residuals against Y

The plot of the residuals against the y-values of each model can be found below. We would like to use this plot to evaluate the model's assumption in regard with our model. We see a clear improvement in the two aspects mentioned before: the linearity and the homoscedasticity. None of the two plots show a clear trend or curve. This is a strong indication of linearity on the model.

Furthermore, the spread of the points in both plots are wider than in model 1 and model 2. We have to mention that the spread in model 4 is more evenly distributed than in model 3.

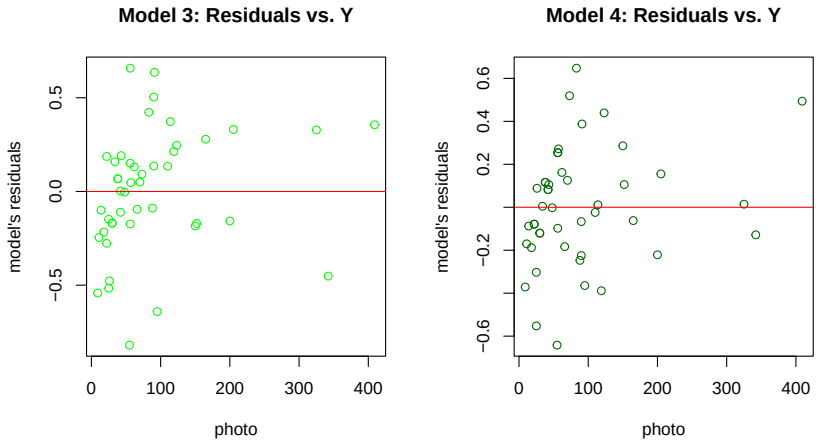

Figure 7: Residuals vs. Y-values

4. Normality of the errors

We investigate the normality of the errors. In these cases, we can see a much better approximation to the normality assumptions. In both cases, we can see that the vast majority of the points are in a diagonal straight line. Therefore, in this cases, there is an strong indication of normality in the errors.

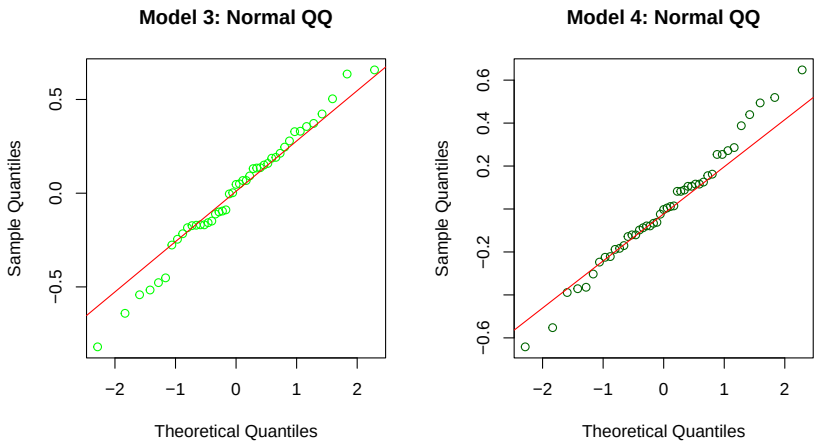

Figure 8: QQ-plots for Model 3 and Model 4

Furthermore, we performed the bootstrap test (calculated with help of the lm.norm.test function). We then confirm that there is no doubt of normality of the residuals.

f) Comparison of Models

When comparing the sets of models, we would preferred to use the log-transformed models to the original models. This is because the log-transformed models follow the model's assumptions (linearity and homoscedasticity), while in the first two these assumptions might not be met. Therefore, we trust the models with the transformed data more than the other two models.

Furthermore, from the two observers, we have concluded that the model that comes from the observer 2 (in both cases: original data and transformed data) is slightly better. We can measure this by comparing the R^2 of the models. We then find out that the best performing model in terms of R^2 is the model 4 with 0.91. This means that 91% of the model can be explained only with that dependent variable. This is the highest of the four model and thus the most preferable model.

For every model, we have created a plot using the computed coefficients against the data points. This help us understand the accuracy of our model in a visual manner. Through these plots, which are displayed below, we can see the improvement of the models when log-transforming the data.

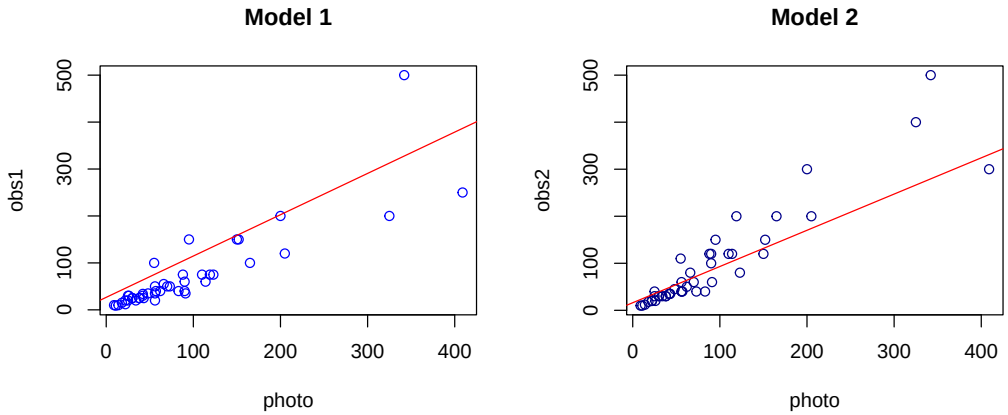

Figure 9: Results when fitting the regression Lines: Models 1 and 2

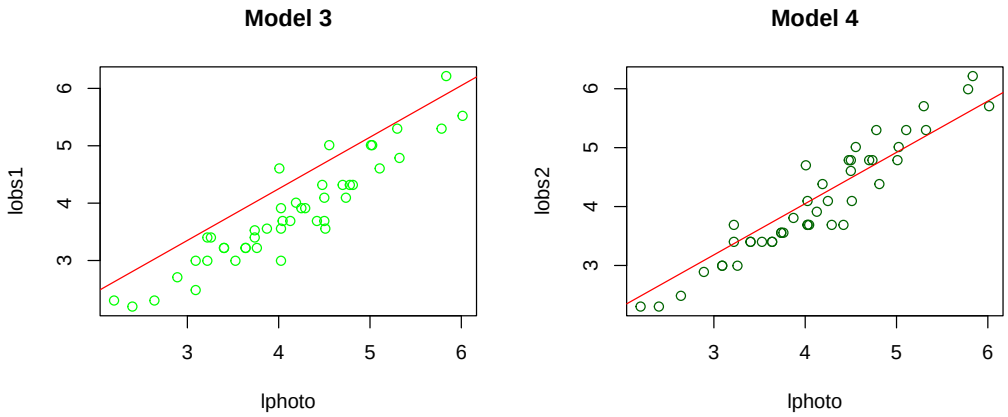

Figure 10: Results when fitting the regression Lines: Models 3 and 4

g) Conclusion

Now, it is possible to answer the question: How well does the photo count reflect the observer counts of the number of geese? As we have previously seen, the results of each model are different. When using the model 4, we have a very good approximation of the number of geese. We conclude then that the photo count reflect the observer count of the number of geese.

7.2: Steam

In this exercise, we are given a table with data about a steam engine that produces glycerine. We have a column names Steam and 9 other variables that possibly explain the relation with the amount of steam used. The objective of this exercise is to create a multiple linear regression model with the steam as the response variable.

Multiple Linear Regression Model

To build a multiple linear regression model, there are several methods. We have used the step up method. The model starts empty and we add variables to the model if there is an improvement in the R^2 and if it passes the t-test. The variables that can be selected are: fatty acid, glycerine, wind mph, calendar days, operating days, freezing days, temperature, wind2, and startups.

The first step of the algorithm is to test these variables individually. In this first iteration, we realized that the temperature variable is the most relevant in terms of R^2. Therefore, we add it to the model. The second step is to test the model with 2 response variables, the temperature plus another one. When performing this iteration, we realized that there are 3 possible variables to add (Fatty acid, glycerine and operating days. Because the results of the 3 models is very similar we decided to create 3 branched to test their performance when adding more variables. For the next iteration in each branch, we test the models Steam Temperature + (Fatty Acid or Glycerine or Operating days)+ third variable.

The result of third iteration is the model that include the following response variable: temperature, operating days and calendar days. We test again to see if there is an improvement in the R^2 and if the t-test is low enough to consider adding the variable. After the corresponding iteration, we realized that even though a fourth variable may improve the R^2, it does not pass the t-test. Therefore, we have decided not to add a fourth variable. The resulting model is: Steam Temperature + Operating days + calendar days.

Formula: $Y = -2.97 - 0.07 * (Temperature) + 0.19 * (OperatingDays) + 0.4 * (Calendarday) + error$
Other relevant statistics of this model are: the R^2 which is 0.99 and the residual standard error, which is 0.59.

Diagnostics tools used: We used two diagnostic tools to set up the model: the added variable plot and the residual vs Y-value plot. Each tool gives us insights about the model, and help us determine if it is appropriate for the given data. We will list the insights gained with each method:

- Added variable plot: is used to assess the relationship between the predictor and a response variable while controlling the effects on other variables. Interpreting the results: When plotting the added-

variable plot in respect with the temperature, we find that the points create a diagonal line with negative slope. This indicates a negative relationship. On the other hand, when plotting this graph in respect to the calendar days, we can clearly see that some x-values have multiple y-values. This indicates that there is some degree of heterogeneity or variation in the relationship between steam and calendar days. The Added-variable plots for the whole model are displayed in the appendix.

- Furthermore, we used the residual against the predictor variable to double check if adding calendar days to the model would make sense or not. We conclude that because the overall spread is kind of constant and adding the variable wont affect this but it will increase the R^2-value. Therefore, we used this tool to confirm our previous decision to add the calendar days variable to the model. We have also used this plot to detect outliers, and we found at least one, that will be investigated later on. These two plots can be also be found in the appendix.

Influence Points and Collinearity

The next step is to investigate about possible influence points and collinearity. First, we investigate the presence of leverage points in the model. We use the hatvalues function and the rule of thumb of $2*(p+1)/n$ to check for these points. We found leverage points in the observation number 7, 15 and 19.

Now we have to investigate the effect of these leverage points in the model. To do this, we use the cooks distance method in R, to check if these points are indeed influence point. If a specific observation has a cooks distance around or greater than 1, then we have an influence point. By this threshold, we do not have any influence points in this model. The largest cooks distance is 0.21 by observation number 14. Another used threshold is $4/n$, which in this case is 0.16. When using this threshold, we could consider observation number 14 as a influence point. We will investigate this further.

To test the effects of this single observation to the whole model, we remove it from the data set. Then we calibrate parameters and estimate the coefficients of the model. There is a slightly improvement of the model in terms of R^2 from 0.88 to 0.89. We have plotted the cooks distance in respect with each observation. This can be found in the appendix.

Next, we investigate collinearity between the response variables. First, by plotting a scatterplot of each pair of observations (these can be found in the appendix). We can graphically confirm that the response variable do not have a linear relationship. To collaborate this in a numerical manner, we compute the correlation matrix. The highest correlation value (in absolute value) is 0.21, which is not high enough to be considered relevant. Finally, we double check this result using the condition indices function in R. Because the largest value is less than 30, we backed-up our conclusion that there is no collinearity between response variables.

Normality assumption for the residuals

To investigate the normality assumption of residuals of the model, we have used some techniques. First, we examine the histogram of the residuals to see if there is the usual bell-shaped histogram. In this case, we do find the a bell-shaped histogram, but it is left-skewed.

Second, we plot the QQ-plot against the normal distribution. If there is a diagonal straight line in the QQ-plot that means that we have a location scale family member of the normal distribution. We find that even though most of the points do follow the desired pattern, there are some outliers that deviate from the diagonal straight line. Finally, we test the residuals with the Shapiro test, and we cannot reject the hypothesis of normality. Therefore, we could conclude that even if there are some minor deviations from the ideal confirmation, the normality assumption holds for this model.

Conclusion on Model

There are several factor to take into account when assessing if a particular model is appropriate for the data. First, it should follow the model's assumptions. In this case, the model follows them. Secondly, we have to test the model against the true y values. For this point, we have created the a plot (see below). We can conclude that this is a very good model for the data. There is a linear relationship between the predictive variable and the response variable used. Furthermore, because the R^2 is very high (=0.88), the model is very intuitive. We can explain almost 90% of the model only with the response variables. These factors make the model appropitate for the data.

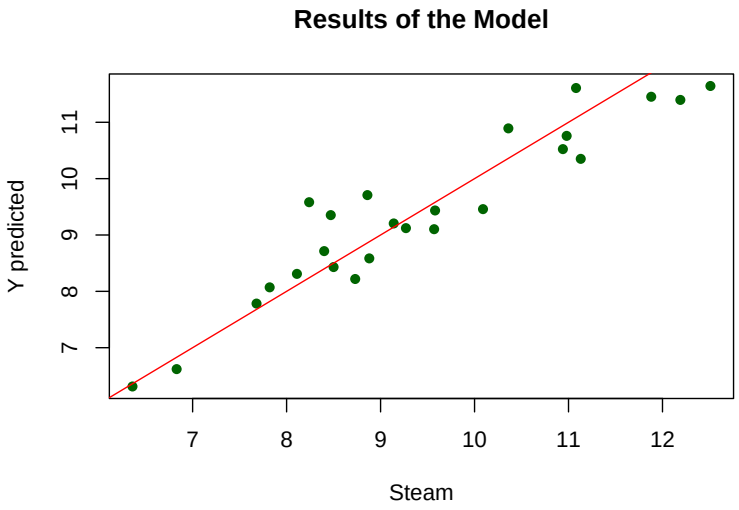

Figure 11: Result of the model

7.3: Expenses on Criminal Activities

The last part of the assignment consists on performing a regression analysis to understand the factors that affect the expenditures on criminal activities. We use the variable expend as the response variable and we have to select between bad, crime, lawyers, employ and pop as the independent variables. We have divided the analysis in the following sections:

Variable Selection Diagnostics

Just as in the previous exercise, we start by selecting the variables to use in the model. To determine this, we use the step-up method. This starts with the empty model and it adds more variables to it when some conditions are met. The first iteration of the method, results in adding the variable *employ*. The second iteration resulted in the addition of the variable *lawyers*. The third iteration did not lead to the addition of a third variable in the model. For now, our model is composed by the variables employ and lawyers.

Leverage Points and Influence Points

Another important aspect of a regression analysis is to investigate the leverage points (potential) and check if these are influence points. After detecting two leverage points in the observation number 5 and number 8, we use cook's distance to asses if these points are influence points. With a cook's distance of 5.47 and 6.38, for observation number 5 and 8, respectively, we confirmed that these observations are influence points. The following plot reflects the cook's distance per observation.

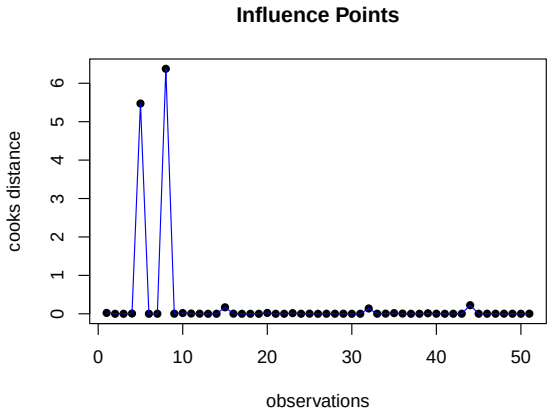

Figure 12: Influence Points in the model

We have tested the model by removing only observation 5, only observation 8 and removing both at the same time. When removing only one observation (does not matter which), we get a $R^2 = 0.969$, which is an improvement from the original model. However, the improvement that we got by removing both observations is even better. We have obtained an R^2 of 0.972.

Multi-collinearity problems

The next step is to investigate if there is collinearity between variables. When there is high multi-collinearity in the model, this can lead to inflated standard errors, making it difficult to determine the statistical significance of the variables. In general, it can lead to a reduced predictive power for the model.

In this exercise, we can measure the collinearity with different techniques. First, we computed a correlation matrix involving the variables in the model. We can rapidly see that there is a very high correlation between the two independent variables in the model (0.97). We also see a high correlation between each individual variable with respect to the predictive variable (around 0.97). This is the first indication of collinearity.

We have used 3 more methods that assess the collinearity between variables. In each of the test, the results is positive in regard with collinearity. The methods used were variance inflation, condition indices, and variance decomposition. We can conclude that there is collinearty between the variables *employ* and *lawyers*.

There are many ways to lead with collinearity. In this case, one of the easiest way is to remove a variable from the model. We could for instance remove the variable *lawyer*; this would lead to a R^2 of 0.96, which is only 0.01 less than the R^2 with both variables.

Investigation of the Residuals

Lastly, it is very important to check whether the model's assumptions are met. Investigating the residuals, is a way to check if the conditions are met for the model to be as effective as possible. The residuals should follow a normal distribution with mean equal to 0.

When plotting the histogram of the residuals of the model (after already removing the influence points), we can see a bell-shaped histogram. Furthermore, when plotting the Normal QQ-plot, we see that most points lie on the normal qq-line. However, it has heavy tails in both ends that deviate from the straight diagonal line. These deviations can be problematic to the normality assumption. These plots can be found in the appendix.

Lastly, we plotted the residuals vs the y-values to check the homoscedasticity assumption in the model. We can see that the spread of the point in respect to the y-axis get larger as x gets larger. This is also a problem to the model's assumptions because it suggests heteroscedasticity. We can also see one outlier in this plot.

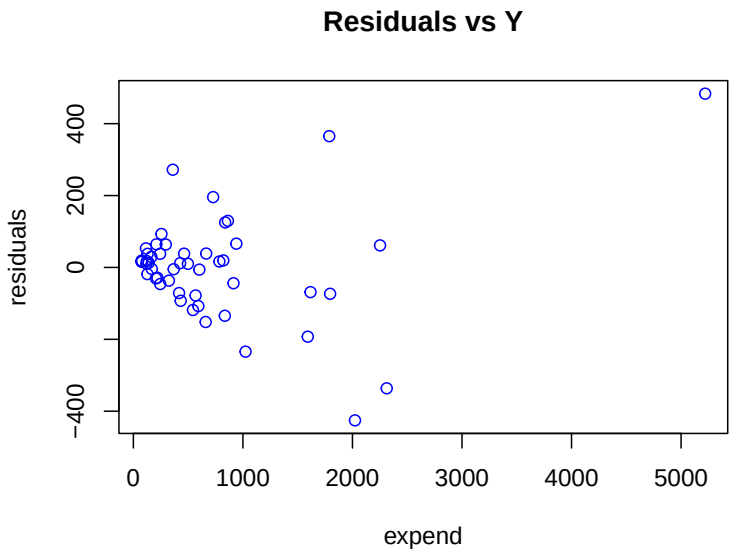

Figure 13: Residuals vs. Y

Appendix

7.1: Supporting Plots

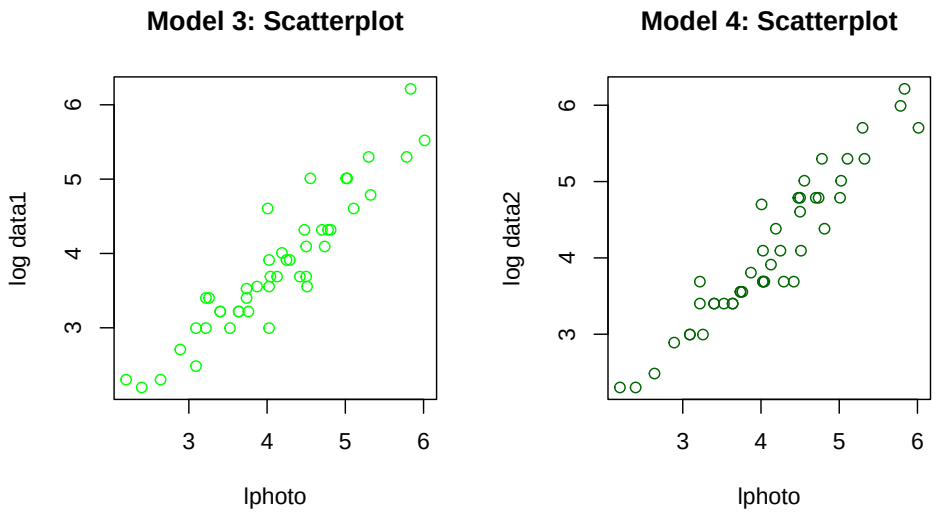

Figure 14: Scatter plot for Model 3 and Model 4

7.1: Code

```
source('functions_Ch8.txt')
sample = read.table('geese.txt', header=T)
photo = sample$photo
obs1 = sample$observer1
obs2 = sample$observer.2

par(mfrow = c(1,2))
plot(photo, obs1)
plot(photo, obs2)

lm1 = lm(photo ~ obs1)
summary(lm1)
# intercept=26.65 and b1=0.88
# F-Test: p-value= 1.54e-14
# Multiple R-squared= 0.75

lm2 = lm(photo ~ obs2)
summary(lm2)
# intercept=16.16 and b1= 0.77
# F-Test: p-value= <2.2e-16
# Multiple R-squared= 0.85

par(mfrow = c(1,2))
plot(photo,lm1$residuals, ylab='model\'s residuals', col='blue', main='Model 1: Residuals
vs. Y')
abline(h=0, col='red')
plot(photo,lm2$residuals, ylab='model\'s residuals', col='darkblue', main='Model 2: Residuals
vs. Y')
abline(h=0, col='red')

#RSE = summary(obs1_lm)[[6]] #residual standard error
```

```
#D = ks.test(residuals(lm1), pnorm, 0., RSE)[[1]]

set.seed(10101101)
hist(lm.norm.test(obs1, photo, 1000))
hist(lm.norm.test(obs2, photo, 1000))

par(mfrow = c(1,2), pty='s')
qqnorm(lm1$residuals, col='blue', main='Model 1: Normal QQ')
qqline(lm2$residuals, col='red')
qqnorm(lm2$residuals, col='darkblue', main='Model 2: Normal QQ')
qqline(lm2$residuals, col='red')

lphoto = log(photo)
lobs1 = log(obs1)
lobs2 = log(obs2)

par(mfrow = c(1,2))
plot(lphoto, lobs1, col='green', main='Model 3: Scatterplot', ylab='log data1')
plot(lphoto, lobs2, col='darkgreen',  main='Model 4: Scatterplot', ylab='log data2')

lm3 = lm(lphoto ~lobs1)
summary(lm3)
# intercept=0.65 and b1=0.90
# F-test p-value= <2.2e-16
# RSE = 0.87

lm4 = lm(lphoto ~lobs2)
summary(lm4)
# intercept=0.57 and b1=0.87
# F-test p-value= <2.2e-16
# RSE = 0.87

par(mfrow = c(1,2))
plot(photo,lm3$residuals, ylab='model\'s residuals', col='green', main='Model 3: Residuals
vs. Y')
abline(h=0, col='red')
plot(photo,lm4$residuals, ylab='model\'s residuals', col='darkgreen', main='Model 4: Residuals
vs. Y')
abline(h=0, col='red')

qqnorm(lm3$residuals, col='green', main='Model 3: Normal QQ')
qqline(lm3$residuals, col='red')
qqnorm(lm4$residuals, col='darkgreen', main='Model 4: Normal QQ')
qqline(lm4$residuals, col='red')

set.seed(10101101)
BS =lm.norm.test(lobs1,lphoto, 1000) #f(x,y) -> y~x
lm.norm.test(lobs2, lphoto, 1000)
par(mfrow = c(1,1), pty='m')
hist(lm.norm.test(lphoto, lobs1, 1000), main='Histogram of the Bootstap Sample')
hist(lm.norm.test(lphoto, lobs2, 1000), main='Histogram of the Bootstap Sample')
# QQ-plot of the residuals
par(mfrow = c(1,1), pty='s')
qqplot(lm3$residuals,lm3$residuals)

par(mfrow = c(1,2), pty='m')
```

```
plot(photo, obs1, col='blue', main ='Model 1')
abline(a=26.650, b=.88, col='red')
plot(photo, obs2, col='darkblue',main ='Model 2')
abline(a=16.16, b=.77, col='red')
plot(lphoto, lobs1, col='green',main ='Model 3')
abline(a=0.65, b=.9, col='red')
plot(lphoto, lobs2, col='darkgreen',main ='Model 4')
abline(a=0.57, b=.87, col='red')
```

7.2: Supporting Plots

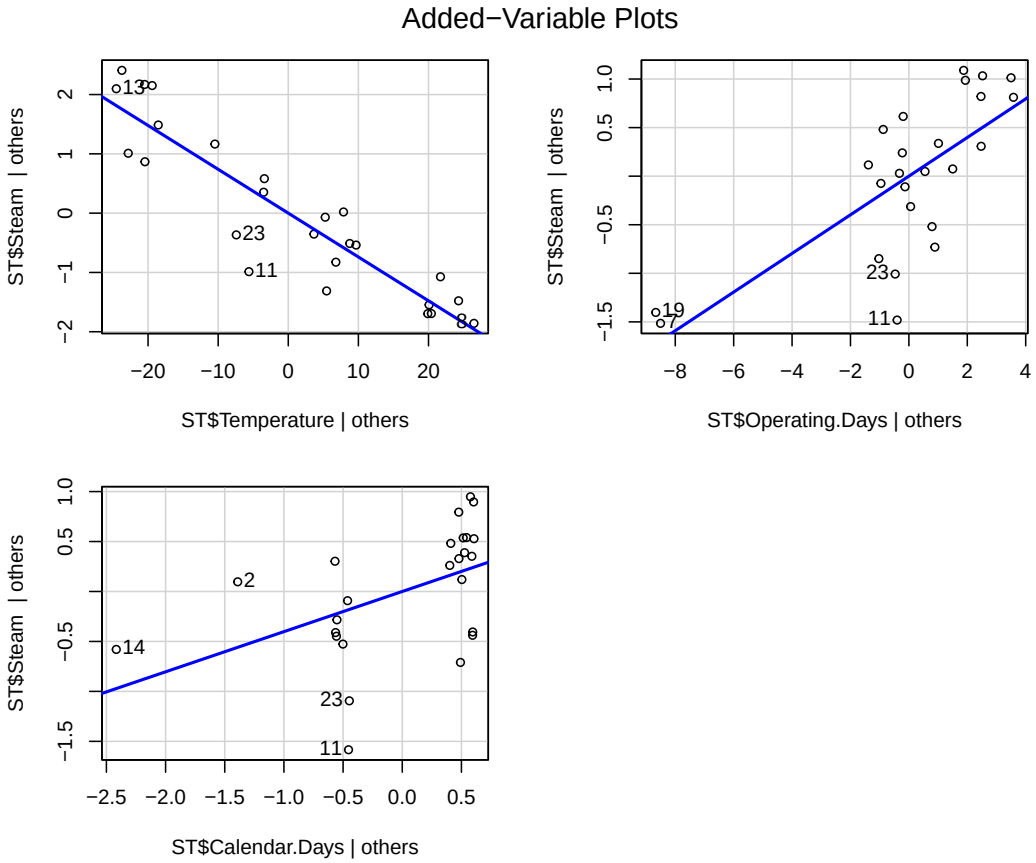

Figure 15: Added Variable Plots

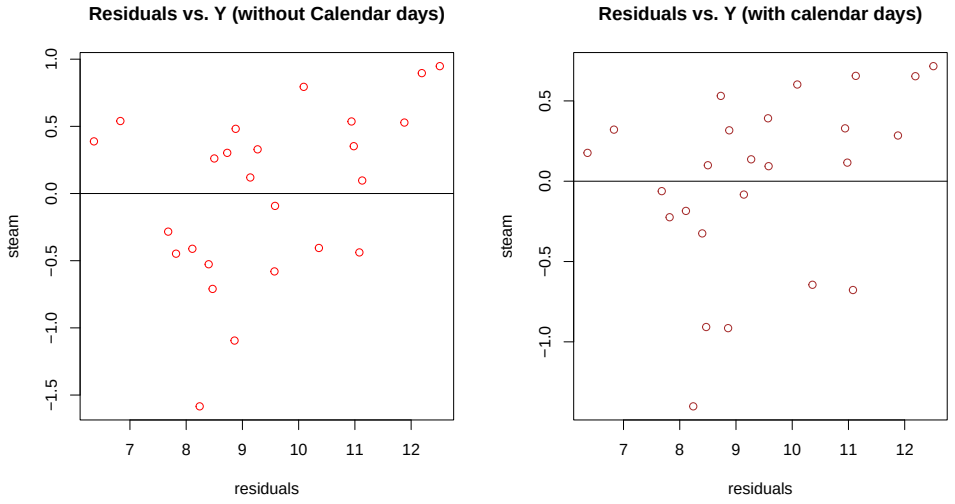

Figure 16: Residuals vs Predictors

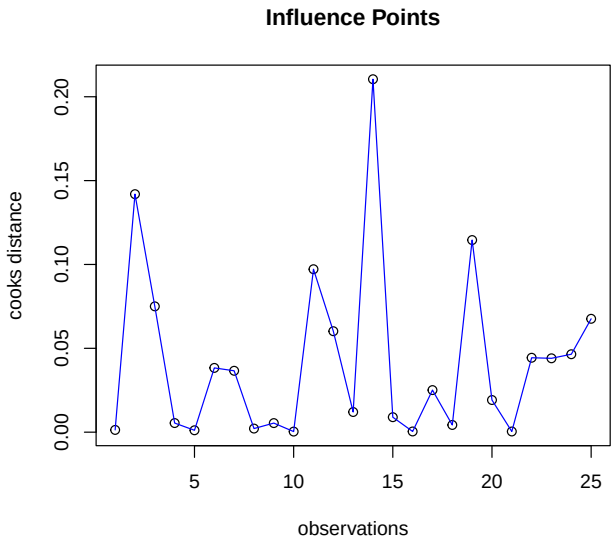

Figure 17: Influence Points in the model

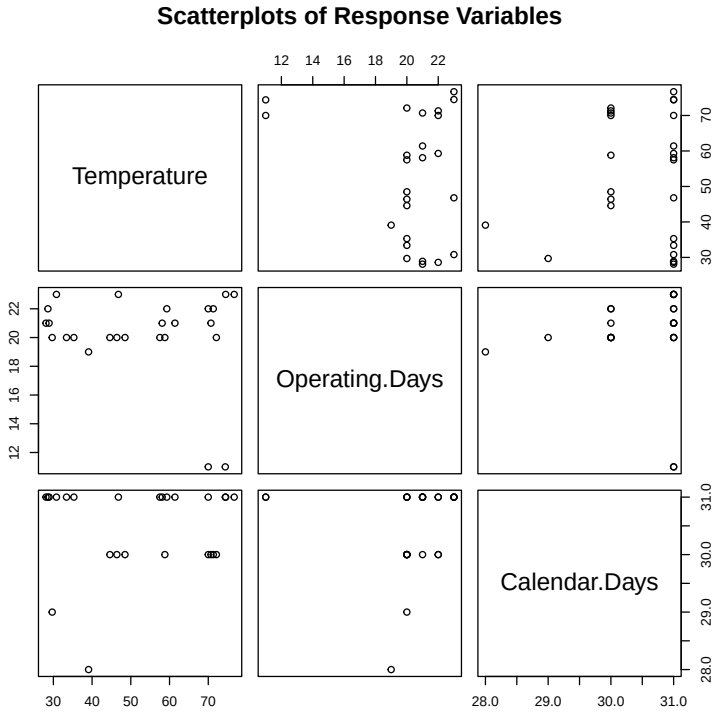

Figure 18: Scatter points between response variables

7.2: Code

```
source('functions_Ch8.txt')
ST = read.table('steamtable.txt', header=T)
pairs(ST)
cor(ST)

summary(lm(ST$Steam~ST$Fatty.Acid)) #R^2=0.147
summary(lm(ST$Steam~ST$Glycerine)) #R^2=0.093
summary(lm(ST$Steam~ST$Wind.Mph)) #R^2=0.225
summary(lm(ST$Steam~ST$Calendar.Days)) #R^2=0.019
summary(lm(ST$Steam~ST$Operating.Days)) #R^2=0.287
summary(lm(ST$Steam~ST$Freezing.Days)) #R^2=0.4104
summary(lm(ST$Steam~ST$Temperature)) #R^2=0.714
```

```
summary(lm(ST$Steam~ST$Wind2)) #R^2=0.156
summary(lm(ST$Steam~ST$Startups)) #R^2=0.146
# According to the step up method, we would add the variable Temperature to the model
# R^2=0.71

#Model 2: Steam ~ Temperature + var.
summary(lm(ST$Steam~ST$Temperature+The ST$Fatty.Acid)) #R^2=0.86 & Pr(>|t|)<5%
summary(lm(ST$Steam~ST$Temperature+ST$Glycerine)) #R^2=0.85 & Pr(>|t|)<5%
summary(lm(ST$Steam~ST$Temperature+ST$Wind.Mph)) #R^2=0.71
summary(lm(ST$Steam~ST$Temperature+ST$Calendar.Days)) #R^2=0.76
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days)) #R^2=0.85 & Pr(>|t|)<5%
summary(lm(ST$Steam~ST$Temperature+ST$Freezing.Days)) #R^2=0.74
summary(lm(ST$Steam~ST$Temperature+ST$Wind2))#R^2=0.72
summary(lm(ST$Steam~ST$Temperature+ST$Startups))#R^2=0.75
# => 3 options to add to the model: Fatty.Acid, Glycerine and Operating.Days

#Model 3.1: Steam ~ Temp + FattyAcid + var
summary(lm(ST$Steam~ST$Temperature+ST$Fatty.Acid+ST$Glycerine)) #0.86
summary(lm(ST$Steam~ST$Temperature+ST$Fatty.Acid+ST$Wind.Mph)) #0.86
summary(lm(ST$Steam~ST$Temperature+ST$Fatty.Acid+ST$Calendar.Days)) #0.864 but p-value>0.05
summary(lm(ST$Steam~ST$Temperature+ST$Fatty.Acid+ST$Operating.Days)) #0.88 but p-value>0.05
summary(lm(ST$Steam~ST$Temperature+ST$Fatty.Acid+ST$Freezing.Days)) #0.86
summary(lm(ST$Steam~ST$Temperature+ST$Fatty.Acid+ST$Wind2))#0.86
summary(lm(ST$Steam~ST$Temperature+ST$Fatty.Acid+ST$Startups)) #0.87 but p-value>0.05

#Model 3.2: Steam ~ Temp + Glycerine + var
summary(lm(ST$Steam~ST$Temperature+ST$Glycerine+ST$Fatty.Acid)) #0.86
summary(lm(ST$Steam~ST$Temperature+ST$Glycerine+ST$Wind.Mph)) #0.85
summary(lm(ST$Steam~ST$Temperature+ST$Glycerine+ST$Calendar.Days)) #0.86
summary(lm(ST$Steam~ST$Temperature+ST$Glycerine+ST$Operating.Days)) #0.86
summary(lm(ST$Steam~ST$Temperature+ST$Glycerine+ST$Freezing.Days)) #0.85
summary(lm(ST$Steam~ST$Temperature+ST$Glycerine+ST$Wind2))#0.85
summary(lm(ST$Steam~ST$Temperature+ST$Glycerine+ST$Startups)) #0.85

#Model 3.3: Steam ~ Temp + Operating Days + var
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Fatty.Acid)) #0.88 (p-value<5%)
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Wind.Mph)) #0.86
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Calendar.Days)) #0.885 (p-val<5%)
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Glycerine)) #0.86
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Freezing.Days)) #0.86
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Wind2))#0.86
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Startups)) #0.85

#model 4: Steam ~ temperature + operating days + calendar Days + var
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Calendar.Days+ST$Fatty.Acid))#0.893
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Calendar.Days+ST$Wind.Mph))
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Calendar.Days+ST$Glycerine))
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Calendar.Days+ST$Freezing.Days))
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Calendar.Days+ST$Wind2))
summary(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Calendar.Days+ST$Startups))

model = lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Calendar.Days)
model2 = lm(ST$Steam~ST$Temperature+ST$Operating.Days)
summary(model)

avPlots(lm(ST$Steam~ST$Temperature+ST$Operating.Days+ST$Calendar.Days))
```

```
avPlots(model)

par(mfrow=c(2,2))
plot(ST$Steam, ST$Temperature)
plot(ST$Steam, ST$Operating.Days)
plot(ST$Steam, ST$Calendar.Days)

plot(ST$Steam, residuals(model), col='brown', main='Residuals vs. Y (with calendar days)', xlab
abline(h=0)

round(hatvalues(model),3)
round(hatvalues(model),3)>2*4/25

par(mfrow=c(1,1))
round(cooks.distance(model), 2)
plot(1:length(ST$Steam), cooks.distance(model), main='Influence Points', xlab='observations', y
lines(1:length(ST$Steam), cooks.distance(model), col='blue')

ST2 = ST #so we do not mess with the original data
toRemove = c(14)
ST2 = ST2[-toRemove, ]
modelC =lm(ST2$Steam~ ST2$Temperature+ST2$Operating.Days+ST2$Calendar.Days)
summary(modelC) #R^2=0.89, RSE=0.59

respVar = ST[, c(8,6,5)]
pairs(respVar, main='Scatterplots of Response Variables')

cor(respVar) # no indication of high correlation
conditionindices(respVar)

par(mfrow = c(1,2), pty='s')
hist(residuals(model), col='lightblue')
qqnorm(residuals(model))
qqline(residuals(model))
shapiro.test(residuals(model))

par(mfrow = c(2,2), pty='s')
hist(residuals(model), col='lightblue')
qqnorm(residuals(model))
qqline(residuals(model))
hist(residuals(model2), col='blue')
qqnorm(residuals(model2))
qqline(residuals(model2))

Y_pred = -2.97-0.07*(ST$Temperature)+0.19*(ST$Operating.Days)+0.4*(ST$Calendar.Days)

par(mfrow =c(1,1), pty='m')
plot(ST$Steam, Y_pred,pch=16, col='darkgreen', main='Results of the Model',
xlab='Steam', ylab='Y predicted')
abline(a = 0, b = 1, col = "red")
```

7.3: Supporting Plots

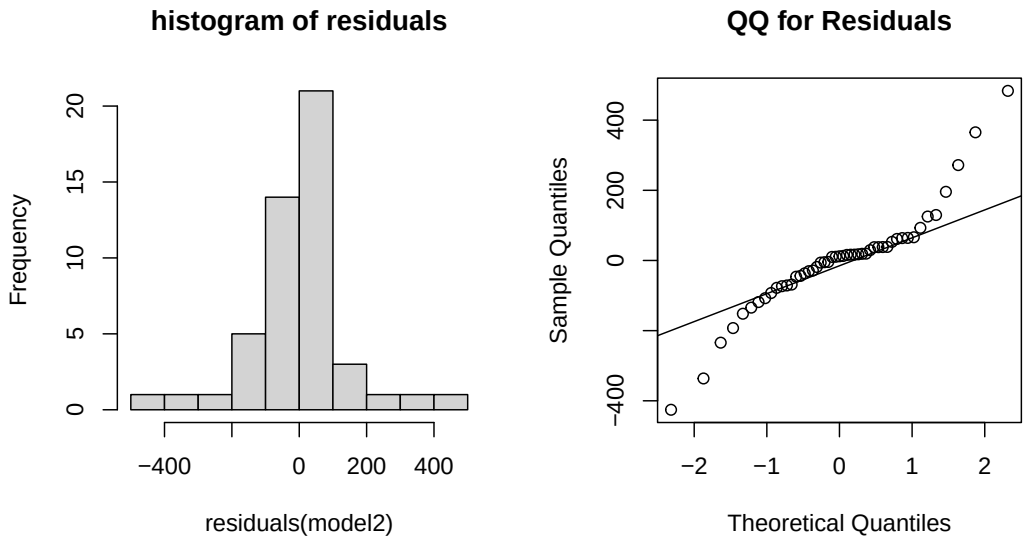

Figure 19: Normality of Residuals

7.3: Code

```
source('functions_Ch8.txt')

EC = read.table('expensescrime.txt', header=TRUE)
expend = EC$expend
bad = EC$bad
crime = EC$crime
lawyers = EC$lawyers
employ = EC$employ
pop =EC$pop

summary(lm(expend ~ bad)) #R=0.7
summary(lm(expend ~ crime)) #R=0.11
summary(lm(expend ~ lawyers)) #0.94
summary(lm(expend ~ employ)) #0.95
summary(lm(expend ~ pop)) #0.91

summary(lm(expend ~ employ+bad)) #0.955
summary(lm(expend ~ employ+crime)) #0.955
summary(lm(expend ~ employ+lawyers)) #0.963 &p-value<5%
summary(lm(expend ~ employ+pop)) #0.954

summary(lm(expend ~ employ+lawyers+bad))
summary(lm(expend ~ employ+lawyers+crime))
summary(lm(expend ~ employ+lawyers+pop))
#none of these perform better than employ+lawyers

# Model chosen: expend ~ employ + lawyers
model = lm(expend~ employ + lawyers)
summary(model)

library(car)
avPlots(model)
```

```
round(hatvalues(model),3)
round(hatvalues(model),3)>2*3/length(expend) #leverage points: 5, 8

round(cooks.distance(model),2)
round(cooks.distance(model),2) > 1 # influence points: 5, 8
plot(1:length(expend), cooks.distance(model), main='Influence Points',pch=16, xlab='observation
lines(1:length(expend), cooks.distance(model), col='blue')

EC2 = EC #so we do not mess with the original data
toRemove = c(5)
EC2 = EC2[-toRemove, ]
modelC =lm(EC2$expend~ EC2$employ+EC2$lawyers)
summary(modelC) #R^2=0.969

toRemove = c(8)
EC2 = EC
EC2 = EC2[-toRemove, ]
modelC2 =lm(EC2$expend~ EC2$employ+EC2$lawyers)
summary(modelC) #R^2=0.969

EC2 = EC
toRemove = c(5,8)
EC2 = EC2[-toRemove, ]
model2 =lm(EC2$expend~ EC2$employ+EC2$lawyers)
summary(modelC) #R^2=0.972

round(cor(EC[c(2,6,5)]),2) # very high between variables
varianceinflation(EC[,c(6,5)]) #larger than 10 are considered problematic
conditionindices(EC[,c(6,5)]) # much larger than 30
round(vardecomposition(EC[,c(6,5)]),3)

plot(EC2$expend, residuals(lm(EC2$expend~EC2$employ)))

hist(residuals(model2), col='lightblue')
qqnorm(residuals(model2))
qqline(residuals(model2))

plot(EC2$expend, residuals(model2))
```