Department of Environment and Geography

University of York

Assessment Submission Cover Sheet 2018/19

This cover sheet should be the first page of your assessment

Exam Number: Y3864317	**Module Code:** ENV00049M
Module Title: Research skills and statistical methods	**Assessment Deadline:** 22nd February 2019

I confirm that I have

- **conformed with University regulations on academic integrity**
- **please insert word count** 1991
- **not written my name anywhere in the assessment**
- **checked that I am submitting the correct and final version of my coursework**
- **saved my assessment in the correct format**
- **formatted my assessment in line with departmental guidelines**
  - I have ensured my work is compatible with black and white printing
  - I have ensured the compatibility of any graphs or charts included in my submission
  - I have used 12pt font (preferably Arial or similar)
  - I have ensured all pages are clearly numbered using the system 1 of 6, 2 of 6 etc. Page 1 will be your Assessment Submission Cover Sheet
  - I have ensured my examination number is on every page of the assessment

PLEASE TICK BOX TO CONFIRM

☑

Please note: if you have any questions please refer to the FAQs available on the VLE (Student Admin Hub).

1. METHODS:

1.1 Introduction to Data

The aim of the project was to evaluate the implications of human activities on grouper fish communities for the use in conservation management. This was achieved by using R studio for exploration and statistical methods to analyse a data set containing the response variable (Number of grouper fish species), and 15 predictor variables (**Table 1**).

The data set had no errors so didn't need manipulating before being imported to R studio. The imported data had predictor variables, watersports and fishing, that were defined as integers, however needed to be defined as factors due to the data being categorical.

1.2 Transforming Data

Some data needed transforming before it was modelled, as if there's too much skew or variance in the data set them problems arise later in testing (Choi, 2016). This included plotting the predictor variables and acknowledging if any variables were skewed through the use of histograms (Zurr *et al*., 2010), boxplots and Q-Q plots. Several predictor variables were found to have significant skew, curvature or heteroscedasticity and were removed using a variety of transformations (**Table 1**). It's often been found that one single transformation doesn't always remove skew, curvature or heteroscedasticity, so a second transformation has to be applied (O'Hara & Kotze, 2010). This was the case with Moray Eels when removing curvature and Coral Simpson when removing skew (**Table 1**).

Table 1: Predictor variable descriptions and the transformations that took place.

Variable	Explanation	Skew	Curvature	Heteroscedasticity
Reef Size	Surface area of the reef (km-sq)	Log	N/A	N/A
Fishing	Presence/absence of fishing	N/A	N/A	N/A
Mean Isolation	Mean distance to all other reefs in the system (km)	Square Root	N/A	N/A
Nearest larger reef	Distance to the nearest largest reef (km)	Log	N/A	N/A
Watersport	Presence/absence of watersports around the edge of each reef	N/A	N/A	N/A
Scuba	Proportion of year which scuba tours visit the reef	N/A	N/A	N/A
Tmin	Minimum mean daily temperature (°C)	N/A	N/A	N/A
Tmax	Maximum mean daily temperature (°C)		Log	
Trange	Range in annual temperature (°C)	Square root	N/A	N/A
NPPmin	Minimum net primary productivity averaged over 20 years (grams of carbon per m-sq per yr).	N/A	N/A	N/A
NPPmax	Maximum net primary productivity averaged over 20 years (grams of carbon per m-sq per yr)	Square root	N/A	N/A
NPPrange	Range in annual productivity (grams of carbon per m-sq per yr)	N/A	N/A	N/A
Moray Eels	Density of moray eels - main competitor for food (individuals per hectare)	N/A	To the sixth power	Square root
Reef Sharks	Density of grey reef sharks- main grouper predator (individuals per km-sq)	N/A	N/A	N/A
Coral (Simpson)	Simpson index of the diversity of coral species	Square root	Square root	Squared

1.3 Intercorrelation

Using the transformed variable data, intercorrelation was acknowledged. This was achieved by the process of elimination based on VIF score. If VIF was greater than 2 then the predictor variable was removed (Zurr *et al.*,2010) until all the variables left had VIF scores lower than 2. During this process of elimination care was taken to ensure that predictor variables which had a strong correlation with the response variable were not removed, and instead the predictor variables which had a strong correlation between one another, a high VIF and little correlation with the response variable were removed first. Using this method, five predictor variables had a high correlation with the response variable (**Table 2**). These five predictor variables were used to create models, which represent predictor variables that could potentially impact on the grouper species if they occur together (**Table 2**). Each model contains at least one predictor variable that has a strong correlation with the response variable, with each variable in the models having a VIF under 2.

Table 2: Predictor variables with a high correlation to the response variable and the predictor variables used in each individual model.

High Correlation	Predictor variable	Model 1	model 2	model 3	model 4	model 5	Model 6
■	Reef size		■				
■	Fishing				■		
	Mean isolation	■	■	■	■	■	
■	Nearest Larger reef	■		■	■	■	■
	Watersports	■					
■	Scuba	■	■	■	■	■	■
	Tmin						
	Tmax					■	
	Trange						
	NPPmin						
	NPPmax						
	NPPrange						
	Moray eels						■
■	Reef sharks	■	■	■	■	■	■
	Coral (simpson)			■			

1.4 Multivariate Generalised Linear Model

The six models created were ran through a multivariate generalised linear model (GLM), in order to test the analysis of deviance, which looks at whether or not the removal of certain variables is the reason for deviance. GLM was used as the data we were using was count data and therefore had a Poisson distribution. To back up results, models were run through backward-forward stepwise reduction (AIC) stage so that the minimum adequate model was produced. To achieve this, it looks at the closeness of the estimated model to the true model. The model with the best fit to the true model, is the minimum adequate model as it has the best quality compared to other models (Akaike, 1987: Gu *et al.*, 2018). The final step includes plotting each individual predictor variable used in the models by using visreg, which observes how the predictor variable is likely to effect the response variable (Breheny & Burchett, 2017).

2. RESULTS:

2.1 Analysis of Deviance

Analysis of deviance was tested for each model created, as this allows us to determine whether the removal of certain predictor variables has caused a significant decrease in the deviance. Deviance of a model is determined using the following equation:

$$Deviance\ (\%) = 1 - \left(\frac{Residual\ deviance}{Null\ deviance}\right) x\ 100$$

The equation above determines the percentage of deviance that can be explained and the significance of the model. From this equation the higher the percentage the better the model due to deviance being explained as well as having more significance. Taking this into account it can be determined that model 2 is most significant as it has a deviance of 75.45%, with model 4 having the lowest deviance of 26.3%, suggesting that it is not a suitable model to be used in terms of grouper fish ecology. Further deviance results for each model are displayed in **Table 3**.

Table 3: Deviance of each model. The lower the deviance the less significant the model and the less deviance that can be explained.

Model	Analysis of deviance (%)
1	50.4
2	75.45
3	47
4	26.3
5	38.9
6	48.9

2.2 Akaike Information Criteria (AIC)

To support analysis of deviance and ensure that the minimum adequate model was established AIC was used. The AIC scores aims to evaluate each individual model produced and compare to one another. As the minimum adequate model is one which is the best fit to the true model it will have a lower AIC score. Therefore, the minimum adequate model produced from this data set is model 2 with an AIC score of 66.9. The higher the AIC score the poorer the fit to the true model. This suggests that model 4 is a bad fit to the data set due to an AIC score of 70.5. Further AIC scores for each model can be seen in **Table 4**.

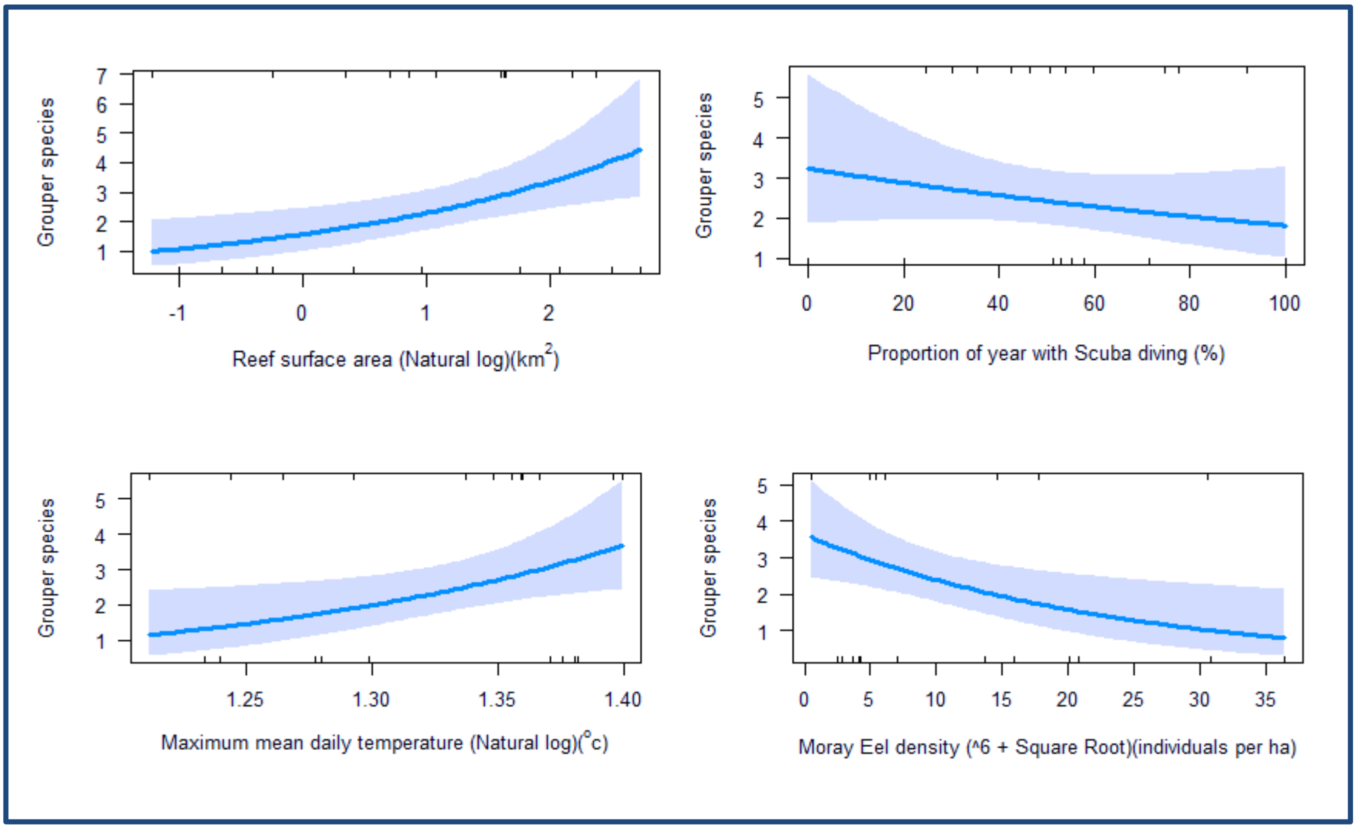
Table 4: AIC scores established for each model. The lower the AIC score the better quality the model is.

Model	Akaike information criteria (AIC)
1	66.9
2	65.9
3	67.5
4	70.5
5	68.6
6	67.1

2.3 Predictor Variable Correlation

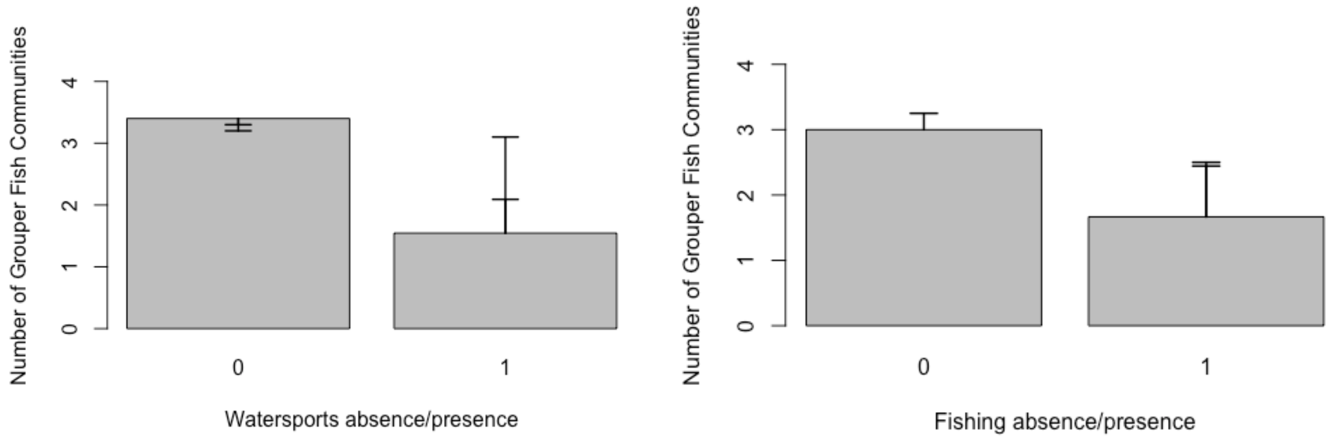
Only certain predictor variables were used in the modelling process, due to their correlation with the response variable. To understand the correlation between predictor variables and the grouper species, visreg graphs with a 95% confidence interval were created. By using visreg graphs the strength of the relationship between the predicator variable and groupers species can be determined. From analysing visreg graphs the four predicator variables which have a strong correlation with groupers species but also have minimal outlying data are displayed in **Figure 1**.

Figure 2: The graphs of the important predictor variables for grouper fish richness: Reef surface area, Proportion of the year with scuba, Maximum mean daily temperature and Moray eel density. The blue line represents the logistic regression line, whilst the shaded blue represents the 95%confidence interval.



As well as the four variables in **Figure 1**, bar graphs have been created due to the nature of data for watersports and fishing. As seen in the bar graphs **(Figure 2)** when fishing and watersports were absent in the reef, the number of groupers present was greater than if fishing and watersports occurred.

Figure 3: Bar graphs representing the spread of groupers when fishing or watersports are absent or



3. DISCUSSION:

3.1 Implications of Findings

Through the use of modelling, it can be said that five predictor variables have a high correlation with the number of grouper species present (**Table 2**). As well as these five predictor variables its also worth paying attention to the other predictor variables used in the modelling as when these variables occur simultaneously it may be problematic for the grouper population. These variables should be taken into account when managing coral reefs, with particular attention being turned to human activities (Hughes *et al*., 2017).

Analysis of data reveals that human activities could have a particular influence on coral reefs, as the number of grouper species present when fishing and watersports occurred was much lower than when no human activities took place. This can also include scuba, which had a negative correlation with groupers, as the more that scuba took place the less groupers were present in that reef (**Figure 1**).

Previous research into coral reef conservation and grouper fish ecology has discovered that Groupers are effected by human activities such as fishing, which is known to remove some species selectively, with groupers often been observed as the first species to disappear (Jackson *et al.,*2001; Pandolfi *et al*., 2003). Due to pressures that coral reefs and grouper fish species are going to face in the future, conservationists need to apply management techniques that will use current knowledge such as information gathered and the future factors that will influence ecosystems to implement and adapt management techniques that will sustain the ecosystem (Mcleod *et al*., 2019). Its important that for conservation of coral reefs to work, interactions between humans, ecosystems and economic systems should be taken into full consideration (Ando &

Mallory, 2012). Actions that could be implemented include rebuilding fisheries, which control access and harvest (McClanahan *et al.,* 2016) or by establishing that some areas of the reef are protected so that human activity stops in that area (Mellin *et al*., 2016).

3.2 Limitations & Future Research

Although the current data set found significant predictor variables that should be considered in relation to grouper fish conservation and coral reef management, limitations did occur. Firstly, the transformations may not be correct, due to the data being count data. It has been acknowledged by researchers that transformations perform badly on count data, so it may be suitable to look for models that are created for counts to avoid transformations (O'Hara & Kotze, 2010).

The modelling process used may also be questioned. Although the models created in this project contained 4 or 5 predictor variables, so the methods used were appropriate, it is worth considering what approach should be taken if only one variable ended up in the GLM. If this was to happen it would be worth considering a univariate model as this allows for regression analysis and analysis of variance for the single predictor variable (IBM, n.d.). univariate models may be useful if this was to occur in future research as it allows for the quality and magnitude of the relationship between the response variable and the predictor variable (Slinker & Glantz, 2008).

Finally, for future projects, it would be beneficial to have more data available. Limited data may have effected the models, which could result in important predictor variables being missed. To improve the data set in the future it would be worth collect integer data for watersports and fishing, rather than just presence and absence, as this would allow for conservationists to see how often a certain human activity was occurring, which would benefit the management of coral reefs.

Although limitations have been discovered with this project, it could also be said that the models created did meet the aim in determining whether human activities were implicating on coral reefs and the groupers fish species. Its therefore been a useful project in determining which variables should be considered in future management of coral reefs and conservation of the grouper fish species.

Word Count: 1991

REFERENCES:

Akaike, H. (1987). Factor analysis and AIC. *Psychometrika*, **52**, pp.317–332

Ando, A.W. and Mallory, M.L. (2012) Optimal portfolio design to reduce climate-related conservation uncertainty in the prairie pot- hole region. *Proceedings of the National Academy of Sciences of the United States of America,***109**, pp.6484–6489. Doi: https://doi.org/10.1073/pnas.1114653109

Breheny, P. and Burchett, W. (2017) Visualization of regression models using Visreg. *The R Journal*, **9**, pp.56-71. Doi: https://doi.org/10.32614/RJ-2017-046

Choi, S. W. (2016), Life is lognormal! What to do when your data does not follow a normal distribution. *Anaesthesia*, **71**, pp.1363-1366. Doi: https://doi.org/10.1111/anae.13666

Gu, Y., Wei, H.L. and Balikhin, M.M. (2018) Nonlinear predictive model selection and model averaging using information criteria. *Systems Science & Control Engineering*, **6**, pp.319-328. Doi: https://doi.org/10.1080/21642583.2018.1496042

Hughes, T.P., Barnes, M.L., Bellwood, D.R., Cinner, J.E., Cumming ,G.S., Jackson, J.B.C., Kleypas, J., van de Leemput, I.A., Lough, J.M., Morrison, T.H., Palumbi, S.R., van Nes, E.H. and Scheffer, M. (2017) Coral reefs in the Anthropocene. *Nature* **546,** pp.82–90. Doi: https://doi.org/10.1038/nature22901

I.B.M. Kowledge Centre (n.d.) GLM Univariate analysis. [Available at: https://www.ibm.com/support/knowledgecenter/en/SSLVMB_23.0.0/spss/base/idh_glmu.html]

Jackson, J.B.C., Kirby, M.X., Berger, W.H., Bjorndal, K.A., Botsford, L.W., Bourque, B.J., Bradbury, R.H., Cooke, R., Erlandson, J., Estes, J.A., Hughes, T.P., Kidwell,S., Lange, C.B., Lenihan, S.H., Pandolfi, J.M., Peterson, C.H., Steneck, R.S., Tegner, M.J. and Warner, R.R. (2001) Historical overfishing and the recent collapse of coastal ecosystems. *Science*, **293**, pp.629–638. Doi: https://doi.org/10.1126/science.1059199

McClanahan, T., Muthiga, N.A. and Abunge, C.A. (2016) Establishment of Community Managed Fisheries' Closures in Kenya: Early Evolution of the *Tengefu* Movement. *Coastal Management*, **44**, pp.1-20. Doi: https://doi.org/10.1080/08920753.2016.1116667

Mcleod, E., Anthony, K.R.N., Mumby, P.J., Maynard, J., Beeden, R., Graham, N.A.J., Heron, S.F., Hoegh-Guldberg, O., Jupiter, S., MacGowen, P., Mangubhai, S., Marshall, N., Marshall, P.A., McClanahan, T.R., Mcleod, K., Nyström, M., Obura, D., Parker, B., Possingham, H.P., Salm, R.V. and Tamelander, J. (2019) The future of resilience-based management in coral reef ecosystems. *Journal of Environmental Management*, **233**, pp.291-301. Doi: https://doi.org/10.1016/j.jenvman.2018.11.034

Mellin, C., MacNeil, M. A., Cheal, A. J., Emslie, M. J. and Caley, M. J. (2016) Marine protected areas increase resilience among coral reef communities. *Ecology Letters*, **19**, pp.629-637. Doi: https://doi.org/10.1111/ele.12598

O'Hara, R. B. and Kotze, D. J. (2010), Do not log-transform count data. *Methods in Ecology and Evolution*, **1**, pp.118-122. Doi: https://doi.org/10.1111/j.2041-210X.2010.00021.x

Pandolfi, J.M., Bradbury, R.H., Sala, E, Hughes, T.P Bjorndal, K.A., Cooke, R. G., McArdle, D., McClenachan, L., Newman, M.J.H., Paredes, G., Warner, R.R. and Jackson, J.B.C. (2003) Global trajectories of the long-term decline of coral reef ecosystems. *Science*, **301**, pp.955–958. Doi: https://doi.org/10.1126/science.1085706

Slinker, B.K. and Glantz, S.A. (2008) Multiple linear regression. *Circulation*, **117**, pp.1732–1737. Doi: https://doi.org/10.1161/CIRCULATIONAHA.106.654376

Symonds, M. R. E., and Moussalli, A. (2011). A brief guide to model selection, multimodel inference and model averaging in behavioural ecology using Akaike's information criterion. *Behavioural Ecology and Sociobiology*, **65**, pp13-21. Doi: https://doi.org/10.1007/s00265-010-1037-6

Zuur, A. F., Ieno, E. N. and Elphick, C. S. (2010), A protocol for data exploration to avoid common statistical problems. *Methods in Ecology and Evolution*, **1**, pp.3-14. Doi: https://doi.org/10.1111/j.2041-210X.2009.00001.x

APPENDIX A

FULLY FUNCTIONING R CODE THAT WAS USED TO COMPLETE DATA ANALYSIS

```
#IMPORT DATA:
fish <- read.table(file.choose(), header=T, sep = "\t")
#CHECK OBJECT MADE:
ls()
#SHOW DATA:
fish
#SHOW HOW DATA VARIABLES ARE DEFINED:
str(fish)
#SET DATA VARIABLES AS ORDERED FACTOR:
fish[, 3] <- factor(fish[, 3], ordered=TRUE)
fish[, 6] <- factor(fish[, 6], ordered=TRUE)
#CHECK NEW DEFINED DATA:
str(fish)
#LOOK AT DATA AGAIN:
fish
#ATTACH DATA TO DATAFRAME:
attach(fish)
#DISPLAYS SUMMARY DATA:
summary(fish)

#############CHECKING FOR ERRORS###################
#SHOWS SIMPLE BASIC CLEVELAND PLOT TO ALLOW US TO SEE ANY OUTLIERS
plot(Groupers)
plot(Reef.size)
plot(Fishing)
plot(Mean.Isolation)
plot(Nearest.Larger.reef)
plot(Watersports)
plot(Scuba)
plot(Tmin)
plot(Tmax)
plot(Tmin)
plot(Tmax)
plot(Trange)
plot(NPPmin)
plot(NPPmax)
plot(NPPrange)
plot(Moray.eels)
plot(Reef.sharks)
plot(Coral..Simpson.)

############CHECKING DATA SPREAD##########

#PLOT SCATTERPLOTS OF ALL VARIABLES:
plot(fish)
#Plot histogram of each individual predictor variable before further analysis.
#CALLED FOR THE "MASS" PACKAGE THAT MAKES HISTOGRAMS.
Library (MASS)

#REEF.SIZE:
hist(Reef.size)
#MEAN.ISOLATION:
hist(Mean.Isolation)
#NEAREST.LARGER.REEF:
hist(Nearest.Larger.reef)
#SCUBA:
hist(Scuba)
#TMIN:
hist(Tmin)
#TMAX:
hist(Tmax)
#TRANGE:
hist(Trange)
#NPPMIN:
hist(NPPmin)
#NNPMAX:
hist(NPPmax)
#NPPRANGE:
hist(NPPrange)
#MORAY.EELS:
hist(Moray.eels)
```

```
#REEF.SHARK:
hist(Reef.sharks)
#CORAl..SIMPSON:
hist(Coral..simpson.)


######### CHECK FOR DATA EVENESS, SPREAD AND NORMALITY OF THE DATA SET. THIS CAN BE DONE BY THE
FOLLOWING CODE. IT IS EASIEST TO VIEW ALL FOUR PLOTS FOR EACH VARIABLE AT ONE TIME RATHER THAN
INDIVIDUALLY. PLOTTING AND VIEWING ALL FOUR PLOTS TOGETHER ###########
# CODE FOR PLOTTING THE FOUR MAIN TYPES OF ANALYSIS USED EARLIER ON:
# TELL R YOU NEED A PACKAGE
require(MASS)
# SPLIT THE GRAPHIC WINDOW INTO 4 QUARTERS:
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
# PLOT THE TRUE HISTOGRAM, WITH AXIS LABELLED:
truehist(Reef.size, nbins = "FD", col = "white", main = "Histogram", xlab = "Reef size (km2)", ylab
= "Frequency", prob = FALSE)
box()
# PLOT KERNEL DENSITY, WITH AXIS LABBELED AND RUG:
plot(density(Reef.size), xlab = "Reef Size (km^2)", main = "Kernel Density")
rug(Reef.size)
# PLOT Q-Q PLOT WITH Q-Q LINE:
qqnorm(Reef.size)
qqline(Reef.size)
# PLOT BOXPLOT, WITH AXIS LABELLED:
boxplot(Reef.size, ylab = "Reef Size (km^2)", main = "Boxplot")
# THEN THIS PUTS A HEADER OVER THE FOUR PLOTS JUST REATED FOR THE ONE VARIABLE:
mtext("Data Exploration Plots for Variable Reef Size", font = 2, outer = TRUE)

#Mean.Isolation:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(Mean.Isolation, nbins = "FD", col = "white", main = "Histogram", xlab = "Mean Isolation
(km)", ylab = "Frequency", prob = FALSE)
box()
plot(density(Mean.Isolation), xlab = "Mean Isolation (km)", main = "Kernel Density")
rug(Mean.Isolation)
qqnorm(Mean.Isolation)
qqline(Mean.Isolation)
boxplot(Mean.Isolation, ylab = "Mean Isolation (km)", main = "Boxplot")
mtext("Data Exploration Plots for Variable Mean Isolation", font = 2, outer = TRUE)

#Nearest.Larger.reef:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(Nearest.Larger.reef, nbins = "FD", col = "white", main = "Histogram", xlab = "Nearest
larger reef (km)", ylab = "Frequency", prob = FALSE)
box()
plot(density(Nearest.Larger.reef), xlab = "Nearest larger reef (km)", main = "Kernel Density")
rug(Nearest.Larger.reef)
qqnorm(Nearest.Larger.reef)
qqline(Nearest.Larger.reef)
boxplot(Nearest.Larger.reef, ylab = "Nearest larger reef (km)", main = "Boxplot")
mtext("Data Exploration Plots for Variable Nearest larger reef", font = 2, outer = TRUE)

#Scuba:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(Scuba, nbins = "FD", col = "white", main = "Histogram", xlab = "Scuba (days)", ylab =
"Frequency", prob = FALSE)
box()
plot(density(Scuba), xlab = "Scuba (days)", main = "Kernel Density")
rug(Scuba)
qqnorm(Scuba)
qqline(Scuba)
boxplot(Scuba, ylab = "Scuba (days)", main = "Boxplot")
mtext("Data Exploration Plots for Variable Scuba", font = 2, outer = TRUE)

# Tmin:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
```

```
truehist(Tmin, nbins = "FD", col = "white", main = "Histogram", xlab = "Tmin (*c)", ylab =
"Frequency", prob = FALSE)
box()
plot(density(Tmin), xlab = "Tmin (*c)", main = "Kernel Density")
rug(Tmin)
qqnorm(Tmin)
qqline(Tmin)
boxplot(Tmin, ylab = "Tmin (*c)", main = "Boxplot")
mtext("Data Exploration Plots for Variable Tmin", font = 2, outer = TRUE)

# Tmax:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(Tmax, nbins = "FD", col = "white", main = "Histogram", xlab = "Tmax (*c)", ylab =
"Frequency", prob = FALSE)
box()
plot(density(Tmax), xlab = "Tmax (*c)", main = "Kernel Density")
rug(Tmax)
qqnorm(Tmax)
qqline(Tmax)
boxplot(Tmax, ylab = "Tmax (*c)", main = "Boxplot")
mtext("Data Exploration Plots for Variable Tmax", font = 2, outer = TRUE)

# Trange:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(Trange, nbins = "FD", col = "white", main = "Histogram", xlab = "Trange (*c)", ylab =
"Frequency", prob = FALSE)
box()
plot(density(Trange), xlab = "Trange (*c)", main = "Kernel Density")
rug(Trange)
qqnorm(Trange)
qqline(Trange)
boxplot(Trange, ylab = "Trange (*c)", main = "Boxplot")
mtext("Data Exploration Plots for Variable Trange", font = 2, outer = TRUE)

# NPPmin:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(NPPmin, nbins = "FD", col = "white", main = "Histogram", xlab = "NPPmin (g C m-sq yr)",
ylab = "Frequency", prob = FALSE)
box()
plot(density(NPPmin), xlab = "Trange (g C m-sq yr)", main = "Kernel Density")
rug(NPPmin)
qqnorm(NPPmin)
qqline(NPPmin)
boxplot(NPPmin, ylab = "NPPmin (g C m-sq yr)", main = "Boxplot")
mtext("Data Exploration Plots for Variable NPPmin", font = 2, outer = TRUE)

# NPPmax:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(NPPmax, nbins = "FD", col = "white", main = "Histogram", xlab = "NPPmax (g C m-sq yr)",
ylab = "Frequency", prob = FALSE)
box()
plot(density(NPPmax), xlab = "NPPmax (g C m-sq yr)", main = "Kernel Density")
rug(NPPmax)
qqnorm(NPPmax)
qqline(NPPmax)
boxplot(NPPmax, ylab = "NPPmax (g C m-sq yr)", main = "Boxplot")
mtext("Data Exploration Plots for Variable NPPmax", font = 2, outer = TRUE)

# NPPrange:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(NPPrange, nbins = "FD", col = "white", main = "Histogram", xlab = "NPPrange (g C m-sq yr)",
ylab = "Frequency", prob = FALSE)
box()
plot(density(NPPrange), xlab = "NPPrange (g C m-sq yr)", main = "Kernel Density")
rug(NPPrange)
qqnorm(NPPrange)
```

```
qqline(NPPrange)
boxplot(NPPrange, ylab = "NPPrange (g C m-sq yr)", main = "Boxplot")
mtext("Data Exploration Plots for Variable NPPrange", font = 2, outer = TRUE)

# Moray.eels:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(Moray.eels, nbins = "FD", col = "white", main = "Histogram", xlab = "Moray Eels (individual
per ha)", ylab = "Frequency", prob = FALSE)
box()
plot(density(Moray.eels), xlab = "Moray Eels (individual per ha)", main = "Kernel Density")
rug(Moray.eels)
qqnorm(Moray.eels)
qqline(Moray.eels)
boxplot(Moray.eels, ylab = "Moray Eels (individual per ha)", main = "Boxplot")
mtext("Data Exploration Plots for Variable Moray Eels", font = 2, outer = TRUE)

# Reef.sharks:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(Reef.sharks, nbins = "FD", col = "white", main = "Histogram", xlab = "Reef Sharks
(individual per km-sq)", ylab = "Frequency", prob = FALSE)
box()
plot(density(Reef.sharks), xlab = "Reef Sharks (individual per km-sq)", main = "Kernel Density")
rug(Reef.sharks)
qqnorm(Reef.sharks)
qqline(Reef.sharks)
boxplot(Reef.sharks, ylab = "Reef Sharks (individual per km-sq)", main = "Boxplot")
mtext("Data Exploration Plots for Variable Reef Sharks", font = 2, outer = TRUE)

# Coral..simpson:
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(Coral..simpson., nbins = "FD", col = "white", main = "Histogram", xlab = "Coral Simpson
index", ylab = "Frequency", prob = FALSE)
box()
plot(density(Coral..simpson.), xlab = "Coral Simpson index", main = "Kernel Density")
rug(Coral..simpson.)
qqnorm(Coral..simpson.)
qqline(Coral..simpson.)
boxplot(Coral..simpson., ylab = "Coral Simpson (individual per km-sq)", main = "Boxplot")
mtext("Data Exploration Plots for Variable Coral Simpson", font = 2, outer = TRUE)



########## TRANSFORMATIONS ##########
# LOOKING AT TRYING TO REMOVE SKEW:

#NOTE: Removed +1 from calculation arguments if the data does not have any 0s.


## 1. Reef.size
# Run seperately!
windows()
plot(density(sqrt(Reef.size)), xlab = "Square Root of Reef Size", main = "Kernel Density")
windows()
plot(density(log(Reef.size)), xlab = "Natural Log of Reef Size", main = "Kernel Density")
windows()
plot(density(log10(Reef.size)), xlab = "Log10 of Reef Size", main = "Kernel Density")
windows()
plot(density(Reef.size^2), xlab = "Reef Size Squared", main = "Kernel Density")
windows()
plot(density(Reef.size^3), xlab = "Reef Size Squared", main = "Kernel Density")

# Natural Log
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(log(Reef.size), nbins = "FD", col = "white", main = "Histogram", xlab = "Natural Log of
Reef Size", ylab = "Frequency", prob = FALSE)
box()
plot(density(log(Reef.size)), xlab = "Natural Log of Reef Size", main = "Kernel Density")
```

```
rug(log(Reef.size))
qqnorm(log(Reef.size))
qqline(log(Reef.size))
boxplot(log(Reef.size), ylab = "Natural Log of Reef Size", main = "Boxplot")
mtext("Data Exploration Plots for Variable Natural Log of Reef Size", font = 2, outer = TRUE)



## 2. Mean.Isolation
# Run seperately!
windows()
plot(density(sqrt(Mean.Isolation)), xlab = "Square Root of Mean Isolation", main = "Kernel Density")
windows()
plot(density(log(Mean.Isolation)), xlab = "Natural Log of Mean Isolation", main = "Kernel Density")
windows()
plot(density(log10(Mean.Isolation)), xlab = "Log10 of Mean Isolation", main = "Kernel Density")
windows()
plot(density(Mean.Isolation^2), xlab = "Mean Isolation Squared", main = "Kernel Density")
windows()
plot(density(Mean.Isolation^3), xlab = "Mean Isolation Squared", main = "Kernel Density")


# Square root
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(sqrt(Mean.Isolation), nbins = "FD", col = "white", main = "Histogram", xlab = "Square Root
of Mean Isolation", ylab = "Frequency", prob = FALSE)
box()
plot(density(sqrt(Mean.Isolation)), xlab = "Square Root of Mean Isolation", main = "Kernel Density")
rug(sqrt(Mean.Isolation))
qqnorm(sqrt(Mean.Isolation))
qqline(sqrt(Mean.Isolation))
boxplot(sqrt(Mean.Isolation), ylab = "Square Root of Mean Isolation", main = "Boxplot")
mtext("Data Exploration Plots for Variable Square Root of Mean Isolation", font = 2, outer = TRUE)


## 3. Nearest.Larger.reef
# Run seperately!
# NOTE: added +1
windows()
plot(density(sqrt(Nearest.Larger.reef)), xlab = "Square Root of Nearest larger reef", main = "Kernel
Density")
windows()
plot(density(log(Nearest.Larger.reef+1)), xlab = "Natural Log of Nearest larger reef", main =
"Kernel Density")
windows()
plot(density(log10(Nearest.Larger.reef+1)), xlab = "Log10 of Nearest larger reef", main = "Kernel
Density")
windows()
plot(density(Nearest.Larger.reef^2), xlab = "Nearest larger reef Squared", main = "Kernel Density")
windows()
plot(density(Nearest.Larger.reef^3), xlab = "Nearest larger reef Squared", main = "Kernel Density")

# Natural Log
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(log(Nearest.Larger.reef+1), nbins = "FD", col = "white", main = "Histogram", xlab =
"Natural Log of Nearest larger reef", ylab = "Frequency", prob = FALSE)
box()
plot(density(log(Nearest.Larger.reef+1)), xlab = "Natural Log of Nearest larger reef", main =
"Kernel Density")
rug(log(Nearest.Larger.reef+1))
qqnorm(log(Nearest.Larger.reef+1))
qqline(log(Nearest.Larger.reef+1))
boxplot(log(Nearest.Larger.reef+1), ylab = "Natural Log of Nearest larger reef", main = "Boxplot")
mtext("Data Exploration Plots for Variable Natural Log of Nearest larger reef", font = 2, outer =
TRUE)

## 4. Scuba
# Run seperately!
# NOTE: added +1
windows()
plot(density(sqrt(Scuba)), xlab = "Square Root of Scuba", main = "Kernel Density")
windows()
```

```
plot(density(log(Scuba+1)), xlab = "Natural Log of Scuba", main = "Kernel Density")
windows()
plot(density(log10(Scuba+1)), xlab = "Log10 of Scuba", main = "Kernel Density")
windows()
plot(density(Scuba^2), xlab = "Scuba Squared", main = "Kernel Density")
windows()
plot(density(Scuba^3), xlab = "Scuba Squared", main = "Kernel Density")

##No skew needs to be removed


## 5. Tmin
# Run seperately!
windows()
plot(density(sqrt(Tmin)), xlab = "Square Root of Tmin", main = "Kernel Density")
windows()
plot(density(log(Tmin)), xlab = "Natural Log of Tmin", main = "Kernel Density")
windows()
plot(density(log10(Tmin)), xlab = "Log10 of Tmin", main = "Kernel Density")
windows()
plot(density(Tmin^2), xlab = "Tmin Squared", main = "Kernel Density")
windows()
plot(density(Tmin^3), xlab = "Tmin Squared", main = "Kernel Density")

##No skew needs to be removed

## 6. Tmax
# Run seperately!
windows()
plot(density(sqrt(Tmax)), xlab = "Square Root of Tmax", main = "Kernel Density")
windows()
plot(density(log(Tmax)), xlab = "Natural Log of Tmax", main = "Kernel Density")
windows()
plot(density(log10(Tmax)), xlab = "Log10 of Tmax", main = "Kernel Density")
windows()
plot(density(Tmax^2), xlab = "Tmax Squared", main = "Kernel Density")
windows()
plot(density(Tmax^3), xlab = "Tmax Squared", main = "Kernel Density")

##No skew needs to be removed

## 7. Trange
# Run seperately!
windows()
plot(density(sqrt(Trange)), xlab = "Square Root of Trange", main = "Kernel Density")
windows()
plot(density(log(Trange)), xlab = "Natural Log of Trange", main = "Kernel Density")
windows()
plot(density(log10(Trange)), xlab = "Log10 of Trange", main = "Kernel Density")
windows()
plot(density(Trange^2), xlab = "Trange Squared", main = "Kernel Density")
windows()
plot(density(Trange^3), xlab = "Trange Squared", main = "Kernel Density")


# Square root
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(sqrt(Trange), nbins = "FD", col = "white", main = "Histogram", xlab = "Square Root of
Trange", ylab = "Frequency", prob = FALSE)
box()
plot(density(sqrt(Trange)), xlab = "Square Root of Trange", main = "Kernel Density")
rug(sqrt(Trange))
qqnorm(sqrt(Trange))
qqline(sqrt(Trange))
boxplot(sqrt(Trange), ylab = "Square Root of Trange", main = "Boxplot")
mtext("Data Exploration Plots for Variable Square Root of Trange", font = 2, outer = TRUE)


## 8. NPPmin
# Run seperately!
windows()
plot(density(sqrt(NPPmin)), xlab = "Square Root of NPPmin", main = "Kernel Density")
windows()
plot(density(log(NPPmin)), xlab = "Natural Log of NPPmin", main = "Kernel Density")
```

```
windows()
plot(density(log10(NPPmin)), xlab = "Log10 of NPPmin", main = "Kernel Density")
windows()
plot(density(NPPmin^2), xlab = "NPPmin Squared", main = "Kernel Density")
windows()
plot(density(NPPmin^3), xlab = "NPPmin Squared", main = "Kernel Density")

##No skew needs to be removed


## 9. NPPmax
# Run seperately!
windows()
plot(density(sqrt(NPPmax)), xlab = "Square Root of NPPmax", main = "Kernel Density")
windows()
plot(density(log(NPPmax)), xlab = "Natural Log of NPPmax", main = "Kernel Density")
windows()
plot(density(log10(NPPmax)), xlab = "Log10 of NPPmax", main = "Kernel Density")
windows()
plot(density(NPPmax^2), xlab = "NPPmax Squared", main = "Kernel Density")
windows()
plot(density(NPPmax^3), xlab = "NPPmax Squared", main = "Kernel Density")

# Square root
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(sqrt(NPPmax), nbins = "FD", col = "white", main = "Histogram", xlab = "Square Root of
NPPmax", ylab = "Frequency", prob = FALSE)
box()
plot(density(sqrt(NPPmax)), xlab = "Square Root of NPPmax", main = "Kernel Density")
rug(sqrt(NPPmax))
qqnorm(sqrt(NPPmax))
qqline(sqrt(NPPmax))
boxplot(sqrt(NPPmax), ylab = "Square Root of NPPmax", main = "Boxplot")
mtext("Data Exploration Plots for Variable Square Root of NPPmax", font = 2, outer = TRUE)



## 10. NPPrange
# Run seperately!
windows()
plot(density(sqrt(NPPrange)), xlab = "Square Root of NPPrange", main = "Kernel Density")
windows()
plot(density(log(NPPrange)), xlab = "Natural Log of NPPrange", main = "Kernel Density")
windows()
plot(density(log10(NPPrange)), xlab = "Log10 of NPPrange", main = "Kernel Density")
windows()
plot(density(NPPrange^2), xlab = "NPPrange Squared", main = "Kernel Density")
windows()
plot(density(NPPrange^3), xlab = "NPPrange Squared", main = "Kernel Density")

##No skew needs to be removed


## 11. Moray.eels
# Run seperately!
windows()
plot(density(sqrt(Moray.eels)), xlab = "Square Root of Moray Eels", main = "Kernel Density")
windows()
plot(density(log(Moray.eels)), xlab = "Natural Log of Moray Eels", main = "Kernel Density")
windows()
plot(density(log10(Moray.eels)), xlab = "Log10 of Moray Eels", main = "Kernel Density")
windows()
plot(density(Moray.eels^2), xlab = "Moray Eels Squared", main = "Kernel Density")
windows()
plot(density(Moray.eels^3), xlab = "Moray Eels Squared", main = "Kernel Density")

##No skew needs to be removed



## 12. Reef.sharks
# Run seperately!
windows()
plot(density(sqrt(Reef.sharks)), xlab = "Square Root of Reef Sharks", main = "Kernel Density")
```

```
windows()
plot(density(log(Reef.sharks)), xlab = "Natural Log of Reef Sharks", main = "Kernel Density")
windows()
plot(density(log10(Reef.sharks)), xlab = "Log10 of Reef Sharks", main = "Kernel Density")
windows()
plot(density(Reef.sharks^2), xlab = "Reef Sharks Squared", main = "Kernel Density")
windows()
plot(density(Reef.sharks^3), xlab = "Reef Sharks Squared", main = "Kernel Density")

##No skew needs to be removed


## 13. Coral..simpson.
# Run seperately!
# NOTE: added +1
windows()
plot(density(sqrt(Coral..simpson.)), xlab = "Square Root of Coral Simpson", main = "Kernel Density")
windows()
plot(density(log(Coral..simpson.+1)), xlab = "Natural Log of Coral Simpson", main = "Kernel
Density")
windows()
plot(density(log10(Coral..simpson.+1)), xlab = "Log10 of Coral Simpson", main = "Kernel Density")
windows()
plot(density(Coral..simpson.^2), xlab = "Coral Simpson Squared", main = "Kernel Density")
windows()
plot(density(Coral..simpson.^3), xlab = "Coral Simpson Squared", main = "Kernel Density")


# Square root
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(sqrt(Coral..simpson.), nbins = "FD", col = "white", main = "Histogram", xlab = "Square Root
of Coral Simpson", ylab = "Frequency", prob = FALSE)
box()
plot(density(sqrt(Coral..simpson.)), xlab = "Square Root of Coral Simpson", main = "Kernel Density")
rug(sqrt(Coral..simpson.))
qqnorm(sqrt(Coral..simpson.))
qqline(sqrt(Coral..simpson.))
boxplot(sqrt(Coral..simpson.), ylab = "Square Root of Coral Simpson", main = "Boxplot")
mtext("Data Exploration Plots for Variable Square Root of Coral Simpson", font = 2, outer = TRUE)


##### NOW TRANSFORMATIONS ARE DONE AND CONFIRMED, DATA SHOULD BE TRANSFORMED IN EXCEL #####
#####ONCE TRANSFORMED IN EXCEL DATA NEEDS TO BE ADDED BACK INTO RSTUDIO SO THAT THE CORAL SIMPSON
CAN BE TRANSFORMED AGAIN USING SQRT######
#DETACH PREVIOUS DATA FRAME
detach(fish)
# Import new data saved using the skew transformations:
fish_trans1 <- read.table(file.choose(), header=T, sep = "\t")
#CHECK OBJECT MADE:
ls()
#SHOW DATA:
fish_trans1
#SHOW HOW DATA VARIABLES ARE DEFINED:
str(fish_trans1)
#DEFINE VARIABLES AGAIN:
fish_trans1[, 3] <- factor(fish_trans1[, 3], ordered=TRUE)
fish_trans1[, 6] <- factor(fish_trans1[, 6], ordered=TRUE)
#SHOW HOW DATA VARIABLES ARE DEFINED:
str(fish_trans1)
#ATTACH DATA TO DATAFRAME:
attach(fish_trans1)
#DISPLAYS SUMMARY DATA:
summary(fish_trans1)

# Square root
windows()
require(MASS)
oldpar <- par(mfrow = c(2,2), oma = c(0,0,2,0) + 0.1)
truehist(sqrt(Coral..simpson.), nbins = "FD", col = "white", main = "Histogram", xlab = "Square Root
of Coral Simpson", ylab = "Frequency", prob = FALSE)
box()
plot(density(sqrt(Coral..simpson.)), xlab = "Square Root of Coral Simpson", main = "Kernel Density")
rug(sqrt(Coral..simpson.))
qqnorm(sqrt(Coral..simpson.))
```

```
qqline(sqrt(Coral..simpson.))
boxplot(sqrt(Coral..simpson.), ylab = "Square Root of Coral Simpson", main = "Boxplot")
mtext("Data Exploration Plots for Variable Square Root of Coral Simpson", font = 2, outer = TRUE)




############ REMOVING CURVATURE ##########
# NOW WE WILL FOLLOW A SIMULAR TRANSFORMATION PROCESS TO ASSESS THE CURVATURE OF RELATIONSHIPS
BETWEEN A PREDICTOR VARIABLE AND THE REPSONSE VARIBALE (GROUPERS)
#DETACH PREVIOUS DATA FRAME
detach(fish)
# Import new data saved using the skew transformations:
fish_trans1 <- read.table(file.choose(), header=T, sep = "\t")
#CHECK OBJECT MADE:
ls()
#SHOW DATA:
fish_trans1
#SHOW HOW DATA VARIABLES ARE DEFINED:
str(fish_trans1)
#DEFINE VARIABLES AGAIN:
fish_trans1[, 3] <- factor(fish_trans1[, 3], ordered=TRUE)
fish_trans1[, 6] <- factor(fish_trans1[, 6], ordered=TRUE)
#SHOW HOW DATA VARIABLES ARE DEFINED:
str(fish_trans1)
#ATTACH DATA TO DATAFRAME:
attach(fish_trans1)
#DISPLAYS SUMMARY DATA:
summary(fish_trans1)

# 1. Reef.size VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(Reef.size, Groupers)
# Relationship looks straight. No curve.

# 2. Mean.isolation VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(Mean.isolation, Groupers)
# Relationship does not look straight. Maybe a slight negative association. Curvature is weak so
doesn't need transforming

# 3. Nearest_larger_reef VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(Nearest.Larger.reef, Groupers)
# Shows a positive relationship with some curvature.

# 4. Scuba VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(Scuba, Groupers)
# Shows a negative relationship with slight curvature. Curvature to weak to transform.

# 5. Tmin VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(Tmin_NL, Groupers)
# Shows a positive relationship with slight curvature. Curvature is weak so does not need
transforming

# 6. Tmax VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(Tmax_C, Groupers)
# Shows a positive relationship, quite straight.

# Natural Log
windows()
plot(log(Tmax_C),Groupers)
# made it even straighter

# 7. Trange VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
```

```
plot(Trange, Groupers)
# Shows a negative relationship, however data is too complex to make any serious changes to.

# 8. NPPmin VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(NPPmin, Groupers)
# Shows a positive relationship, with a weak curve.

# 9. NPPmax VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(NPPmax, Groupers)
# Shows a positive relationship, kind of straight, so does not need transforming

# 10. NPPrange VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(NPPrange, Groupers)
# Shows a positive relationship, with a curve that is too weak to transform.

# 11. Moray.eels VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(Morsy.eels, Groupers)
# Shows a negative relationship, only has a very slight curve.

# Power 6
windows()
plot((Moray.eels^6),Groupers)
# BEST

# 12. Reef.Sharks VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(Reef.sharks, Groupers)
# No real relationship or curve, so would be too complex to transform.


# 13. Coral..Simpson VS Groupers
# Look at normal plot to assess if relationship is curved.
windows()
plot(Coral..simpson, Groupers)
# Positive and straight, with a weak curve.

# Square root
windows()
plot(sqrt(Coral..simpson.),Groupers)
# improves curvature



########## HETEROSCEDASTICITY ##########
# WE ARE NOW PLOTTING RESPONSE vs PREDICTOR VARIABLE TO CHECK FOR UNEVEN VARIANCE.
# THIS INCLUDES USING A CALCULATION TO SEE IF THE VARIANCE IS GREATER THAN 4, WHICH ACCORDING TO
ZUUR MEANS IT WOULD BE SERIOUS, AND HENCE WOULD NEED TRANSFORMING.

# First detach old data:
detach (fish_trans1)
# Import new data saved from curvature transformations:
fish_trans2<- read.table(file.choose(), header=T, sep = "\t")

#CHECK OBJECT MADE:
ls()
#SHOW DATA:
fish_trans2
#SHOW HOW DATA VARIABLES ARE DEFINED:
str(fish_trans2)
#DEFINE VARIABLES AGAIN:
fish_trans2 [, 3] <- factor(fish_trans2 [, 3], ordered=TRUE)
fish_trans2 [, 6] <- factor(fish_trans2 [, 6], ordered=TRUE)
#SHOW HOW DATA VARIABLES ARE DEFINED:
str(fish_trans2)
#ATTACH DATA TO DATAFRAME:
attach(fish_trans2)
#DISPLAYS SUMMARY DATA:
```

```
summary(fish_trans2)

# 1. Reef_Size VS Groupers
windows()
plot(Reef.size, Groupers)
# Calculating variance:
var(Reef.size[Groupers<mean(Groupers)])/ var(Reef.size[Groupers>mean(Groupers)])
 =1.879448
# The resulting number is LOWER than 4, so NO transformation needed.

# 2. Fishing VS Groupers
windows()
plot(Fishing, Groupers)
var(Groupers[Fishing==0])/var(Groupers[Fishing==1])
=0.4848485
# Smaller than 4, so no transformation necessary

# 3. Mean.Isolation VS Groupers
windows()
plot(Mean.isolation, Groupers)
# Calculating variance:
> var(Mean.isolation[Groupers<mean(Groupers)])/var(Mean.isolation[Groupers>mean(Groupers)])
=0.5065373
# The resulting number is LESS 4, so there is no need to transform.

# 4. Nearest.Larger.reef VS Groupers
windows()
plot(Nearest.Larger.reef, Groupers)
# Calculating variance:
var(Nearest.Larger.reef[Groupers<mean(Groupers)])/var(Nearest.Larger.reef[Groupers>mean(Groupers)])
=3.055479
# The resulting number is less than 4, so no need to transform.

# 5. Watersports VS Groupers
windows()
plot(Watersports, Groupers)
# Caluclating variance:
var(Groupers[Watersports==0]) / var(Groupers[Watersports==1])
= 0.3055556
# Smaller than 4, so no transformation necessary

# 6.   Scuba VS Groupers
windows()
plot(Scuba, Groupers)
# Calculating variance:
var(Scuba[Groupers<mean(Groupers)])/var(Scuba[Groupers>mean(Groupers)])
=1.135642

# 7. Tmin VS Groupers
windows()
plot(Tmin_NL, Groupers)
# Calculating variance:
var(Tmin[Groupers<mean(Groupers)])/var(Tmin[Groupers>mean(Groupers)])
=0.6799132
# The resulting number is LESS than 4, so there is no need to transform.

# 8. Tmax VS Groupers
windows()
plot(Tmax_C, Groupers)
var(Tmax[Groupers<mean(Groupers)])/var(Tmax[Groupers>mean(Groupers)])
=1.393753
# The resulting number is much less than 4, so there is no need to transform.

# 9. Trange VS Groupers
windows()
plot(Trange_Ex, Groupers)
# Calculating variance:
var(Trange[Groupers<mean(Groupers)])/var(Trange[Groupers>mean(Groupers)])
=1.047138
# The resulting number is much less than 4, so there is no need to transform.

# 10. NPPmin VS Groupers
windows()
plot(NPPmin, Groupers)
# Calculating variance:
var(NPPmin[Groupers<mean(Groupers)])/var(NPPmin[Groupers>mean(Groupers)])
```

```
=0.8670191
# The resulting number is much less than 4, so there is no need to transform.

# 11. NPPmax VS Groupers
windows()
plot(NPPmax, Groupers)
# Calculating variance:
var(NPP.max[Groupers<mean(Groupers)])/var(NPP.max[Groupers>mean(Groupers)])
=1.282886
# The resulting number is much less than 4, so there is no need to transform.

# 12. NPPrange VS Groupers
windows()
plot(NPPrange, Groupers)
# Calculating variance:
var(NPPrange[Groupers<mean(Groupers)])/var(NPPrange[Groupers>mean(Groupers)])
=1.148367
# The resulting number is LOWER 4, so there is no need to transform.

# 13. Moray.eels VS Groupers
windows()
plot(Moray.eels, Groupers)
# Calculating variance:
var(Moray.eels[Groupers<mean(Groupers)])/var(Moray.eels[Groupers>mean(Groupers)])
=4.341079
# The resulting number is HIGHER than 4, so needs transforming.
## Transforming Moray.eels:
# Square root
var(sqrt(Moray.eels)[Groupers<mean(Groupers)])/var(sqrt(Moray.eels)[Groupers>mean(Groupers)])
=1.713473
# this transformation removes the heteroscadacity.

# 14. Reef.sharks VS Groupers
windows()
plot(Reef.sharks, Groupers)
# Calculating variance:
var(Reef.Sharks[Groupers<mean(Groupers)])/var(Reef.Sharks[Groupers>mean(Groupers)])
=0.2346609
# The resulting number is LESS than 4, so there is no need to transform.

# 15. Coral..simpson VS Groupers
windows()
plot(Csimp_SQRT, Groupers)
# Calculating variance:
var(Coral..simpson.[Groupers<mean(Groupers)])/var(Coral..simpson.[Groupers>mean(Groupers)])
=11.56337
# The resulting number is much HIGHER than 4 so needs transforming.
#Squared
var(Coral..simpson.[Groupers<mean(Groupers)]^2)/var(Coral..simpson.[Groupers>mean(Groupers)]^2)
=2.306121
# This transformation removes heteroscadacity

########## IDENTIFYING INTERCORRELATION  ###########
# Now we have transformed our data into the correct format for more detailed analysis, we need to
begin investigating the relationship between all of the variables
# At this point it is the relationship between PREDICTOR VARIABLES that is most important
#  WE FIRST NEED TO IMPORT THE DATA WITH UNDEFINED FACTORS, THIS IS BECAUSE CALCULATING THE VIF
SCORE NEEDS ALL TO BE NUMBERS/INTEGERS.
# Import data:
fish_trans3 <- read.table(file.choose(), header=TRUE, row.names = NULL)
# Check it is made:
ls()
# Look at data
fish_trans3
# Check how data is defined:
str(fish_trans3)
# If data needs definfing as numeric instead of factor:
fish_trans3 [, 3] <- as.numeric(fish_trans3 [, 3])
fish_trans3 [, 6] <- as.numeric(fish_trans3 [, 6])
# Attach data:
attach(fish_trans3)

# First, plot a simple scatter-plot matrix of first four variables:
windows()
plot(fish_trans3[,1:4])
# To remove the gaps between plots:
```

```
windows()
plot(fish_trans3[,1:4], gap=0)

# Then can draw a plot that includes kernel density:
plot(fish_trans3 [,1:4], gap = 0, diag.panel = function(x) {
par(new = TRUE)
plot(density(x), ann = FALSE, axes = FALSE)
rug(x)
})

#### The following code works out two important parameters for looking at intercorrelation: ####
# (1) Pearson correlation coefficients (r) (a measure of the correlation between individual
variables)
# (2) Variance Inflation Factor (VIF) scores (a measure of the correlations of each variable to all
of the others)
# And produces a more interesting scatterplot matrix.

panel.cor <- function(x, y, digits=1, prefix="", cex.cor)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r1=cor(x,y,use="pairwise.complete.obs")
    r <- abs(cor(x, y,use="pairwise.complete.obs"))

    txt <- format(c(r1, 0.123456789), digits=digits)[1]
    txt <- paste(prefix, txt, sep="")
    if(missing(cex.cor)) cex <- 0.9/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex * r)
}

panel.smooth2=function (x, y, col = par("col"), bg = NA, pch = par("pch"),
    cex = 1, col.smooth = "red", span = 2/3, iter = 3, ...)
{
    points(x, y, pch = pch, col = col, bg = bg, cex = cex)
    ok <- is.finite(x) & is.finite(y)
    if (any(ok))
        lines(stats::lowess(x[ok], y[ok], f = span, iter = iter),
            col = 1, ...)
}

panel.lines2=function (x, y, col = par("col"), bg = NA, pch = par("pch"),
    cex = 1, ...)
{
    points(x, y, pch = pch, col = col, bg = bg, cex = cex)
    ok <- is.finite(x) & is.finite(y)
    if (any(ok)){
        tmp=lm(y[ok]~x[ok])
        abline(tmp)}
}

panel.hist <- function(x, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5) )
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col="white", ...)
}


#VIF (for this section the data MUST BE DEFINED AS NUMERIC OR INTEGER)
myvif <- function(mod) {
  v <- vcov(mod)
  assign <- attributes(model.matrix(mod))$assign
  if (names(coefficients(mod)[1]) == "(Intercept)") {
    v <- v[-1, -1]
    assign <- assign[-1]
  } else warning("No intercept: vifs may not be sensible.")
  terms <- labels(terms(mod))
  n.terms <- length(terms)
  if (n.terms < 2) stop("The model contains fewer than 2 terms")
  if (length(assign) > dim(v)[1] ) {
     diag(tmp_cor)<-0
     if (any(tmp_cor==1.0)){
        return("Sample size is too small, 100% collinearity is present")
     } else {
        return("Sample size is too small")
     }
  }
  R <- cov2cor(v)
  detR <- det(R)
  result <- matrix(0, n.terms, 3)
  rownames(result) <- terms
  colnames(result) <- c("GVIF", "Df", "GVIF^(1/2Df)")
  for (term in 1:n.terms) {
    subs <- which(assign == term)
    result[term, 1] <- det(as.matrix(R[subs, subs])) * det(as.matrix(R[-subs, -subs])) / detR
    result[term, 2] <- length(subs)
  }
  if (all(result[, 2] == 1)) {
    result <- data.frame(GVIF=result[, 1])
  } else {
    result[, 3] <- result[, 1]^(1/(2 * result[, 2]))
  }
  invisible(result)
}

corvif <- function(dataz) {
    dataz <- as.data.frame(dataz)
    #correlation part
    cat("Correlations of the variables\n\n")
    tmp_cor <- cor(dataz,use="complete.obs")
    print(tmp_cor)

    #vif part
    form    <- formula(paste("fooy ~ ",paste(strsplit(names(dataz)," "),collapse=" + ")))
    dataz   <- data.frame(fooy=1,dataz)
    lm_mod  <- lm(form,dataz)

    cat("\n\nVariance inflation factors\n\n")
    print(myvif(lm_mod))
}

myvif <- function(mod) {
  v <- vcov(mod)
  assign <- attributes(model.matrix(mod))$assign
  if (names(coefficients(mod)[1]) == "(Intercept)") {
    v <- v[-1, -1]
    assign <- assign[-1]
  } else warning("No intercept: vifs may not be sensible.")
  terms <- labels(terms(mod))
  n.terms <- length(terms)
  if (n.terms < 2) stop("The model contains fewer than 2 terms")
  if (length(assign) > dim(v)[1] ) {
     diag(tmp_cor)<-0
     if (any(tmp_cor==1.0)){
        return("Sample size is too small, 100% collinearity is present")
     } else {
        return("Sample size is too small")
     }
  }
  R <- cov2cor(v)
  detR <- det(R)
  result <- matrix(0, n.terms, 3)
  rownames(result) <- terms
  colnames(result) <- c("GVIF", "Df", "GVIF^(1/2Df)")
  for (term in 1:n.terms) {
    subs <- which(assign == term)
    result[term, 1] <- det(as.matrix(R[subs, subs])) * det(as.matrix(R[-subs, -subs])) / detR
    result[term, 2] <- length(subs)
  }
  if (all(result[, 2] == 1)) {
    result <- data.frame(GVIF=result[, 1])
  } else {
    result[, 3] <- result[, 1]^(1/(2 * result[, 2]))
  }
  invisible(result)
}
```

```
corvif <- function(dataz) {
    dataz <- as.data.frame(dataz)
    #correlation part
    cat("Correlations of the variables\n\n")
    tmp_cor <- cor(dataz,use="complete.obs")
    print(tmp_cor)

    #vif part
    form    <- formula(paste("fooy ~ ",paste(strsplit(names(dataz)," "),collapse=" + ")))
    dataz   <- data.frame(fooy=1,dataz)
    lm_mod  <- lm(form,dataz)

    cat("\n\nVariance inflation factors\n\n")
    print(myvif(lm_mod))
}

### Following all of this code, we can plot a scatterplot matrix and some correlation statistics
using the following code:
Z <- cbind(fish_trans3)
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)

corvif(Z)

# THIS HAS THEN PRODUCED THREE OUTPUTS FOR THE VARIABLE SSELECTED IN THE CODE:
# 1) Scatterplot matrix including (rounded off) Pearson correlation coefficients # Top plot above
the diagonal
# 2) A correlation matrix of Pearson coefficients # Bottom plot below the diagonals
# 3) VIF scores for each variable # In the R console output

# NOW WE WILL RUN AGAIN WITH ALL VARIABLES INCLUDED.
# FIRST REMOVE THE LAST CORRELATION AND VIF TO AVOID CONFUSION:
rm(Z)
# NOW RUN NEW ONE:
windows()
Z <- cbind(fish_trans3)
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)

corvif(Z)

## Code for Pearson correlation tests between all variables Produces scatterplot, r values and VIF
scores.
# NOTE: The output of this produces a matrix of numbers, with 1s on the diagonal indicating perfect
correlation when correlating each variable correlated against itself, with p values above the
diagonal, and r values below

cor.prob <- function(X, dfr = nrow(X) - 2) {
         R <- cor(X)
         above <- row(R) < col(R)
         r2 <- R[above]^2
         Fstat <- r2 * dfr / (1 - r2)
         R[above] <- 1 - pf(Fstat, 1, dfr)
         R
}
cor.prob(fish_trans3)

# At this point you have the option of using more complex methods to solve intercorrelation. But due
to time constraints I will proceed with the simple removal method.

# First we need the VIf scores and the R values WITHOUT the response variable - Groupers.
# Remove last correlation:
rm(Z)
# To select all data expect Groupers and run the correlation again:
Z <- cbind(fish_trans3[,2:16])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)

corvif(Z)

Based on VIF scores certain variables are to be removed, so that the VIF score can be under 2.

# Removal of Tmin, Tmax, Trange, NPPmin, NPPmax, NPPrange.
rm(Z)
Z <- cbind(fish[,c(2:7, 14:16)])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)
corvif(Z)

#Remove Coral..simpson, Moray.eels and Mean.isolation
rm(Z)
Z <- cbind(fish[,c(2:3,5:7,15)])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)
corvif(Z)

# Remove Watersports
rm(Z)
Z <- cbind(fish[,c(2:3,5,7,15)])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)
corvif(Z)

Model 1 (Mean.isolation, Nearest.Larger.reef, Watersports, Scuba, Reef.sharks):
rm(Z)
Z <- cbind(fish[,c(4:7,15)])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)
corvif(Z)

Model 2 (Reef.size, Means.isolation, Scuba, Reef.sharks):
rm(Z)
Z <- cbind(fish[,c(2,4,7,15)])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)
corvif(Z)

#Model 3 (Mean.isolation, Nearest.Larger.reef, Scuba, Reef.sharks, Coral..simpson):
rm(Z)
Z <- cbind(fish[,c(4,5,7,15,16)])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)
corvif(Z)


#Model 4 (Fishing, Mean.isolation, Nearest.larger.reef, Scuba, Reef.sharks):
rm(Z)
Z <- cbind(fish[,c(3,4,5,7,15)])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)
corvif(Z)


#Model 5 (Mean.isolation, Nearest.Larger.reef, Scuba, Tmax, Reef.sharks):
rm(Z)
Z <- cbind(fish[,c(4,5,7,9,15)])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)
corvif(Z)


#Model 6 (Nearest.Larger.reef, Scuba, Moray.eels, Reef.sharks):
rm(Z)
Z <- cbind(fish[,c(5,7,14,15)])
pairs(Z, lower.panel=panel.smooth,
upper.panel=panel.cor)
corvif(Z)


#######General linearized models#######
fish_trans3 <- read.table(file.choose(), header=T, sep = "\t")
fish_trans3[,3]<-factor(fish_trans3[,3], ordered=TRUE)
fish_trans3[,6]<-factor(fish_trans3[,6], ordered=TRUE)
attach(fish_trans3)
library(mgcv)

#GLM MODEL 1
GLM1<-glm(Groupers~Mean.isolation+Nearest.larger.reef+Watersports+Scuba+Reef.Sharks, poisson)
summary(GLM1)
par(mfrow=c(2,2))
plot(GLM1)
#GLM MODEL 2
```

```
GLM2 <- glm(Groupers~Reef.size+ Mean.isolation+Scuba+Reef.Sharks, poisson)
summary(GLM2)
par(mfrow=c(2,2))
plot(GLM2)
#GLM MODEL 3
GLM3 <- glm(Groupers~Mean.isolation+Nearest.larger.reef+Scuba+Reef.Sharks+Coral..simpson., poisson)
summary(GLM3)
par(mfrow=c(2,2))
plot(GLM3)
#GLM MODEL 4
GLM4 <- glm(Groupers~Fishing+Mean.isolation+Nearest.larger.reef+Scuba+Reef.Sharks, poisson)
summary(GLM4)
par(mfrow=c(2,2))
plot(GLM4)
#GLM MODEL 5
GLM5 <- glm(Groupers~Mean.isolation+Nearest.larger.reef+Scuba+Tmax+Reef.Sharks, poisson)
summary(GLM5)
par(mfrow=c(2,2))
plot(GLM5)
#GLM MODEL 6
GLM6 <- glm(Groupers~Nearest.larger.reef+Scuba+Moray.Eels+Reef.Sharks, poisson)
summary(GLM6)
par(mfrow=c(2,2))
plot(GLM6)

## Stepwise reduction using AIC forward and backward:
# GLM1 -> GLM1.AIC
library(MASS)
GLM1.AIC <- stepAIC(GLM1, dir="both")
summary(GLM1.AIC)
par(mfrow=c(2,2))
plot(GLM1.AIC)
# GLM2 -> GLM2.AIC
library(MASS)
GLM2.AIC <- stepAIC(GLM2, dir="both")
summary(GLM2.AIC)
par(mfrow=c(2,2))
plot(GLM2.AIC)
# GLM3 -> GLM3.AIC
library(MASS)
GLM3.AIC <- stepAIC(GLM3, dir="both")
summary(GLM3.AIC)
par(mfrow=c(2,2))
plot(GLM3.AIC)
# GLM4 -> GLM4.AIC
library(MASS)
GLM4.AIC <- stepAIC(GLM4, dir="both")
summary(GLM4.AIC)
par(mfrow=c(2,2))
plot(GLM4.AIC)
# GLM5 -> GLM5.AIC
library(MASS)
GLM5.AIC <- stepAIC(GLM5, dir="both")
summary(GLM5.AIC)
par(mfrow=c(2,2))
plot(GLM5.AIC)
# GLM6 -> GLM6.AIC
library(MASS)
GLM6.AIC <- stepAIC(GLM6, dir="both")
summary(GLM6.AIC)
par(mfrow=c(2,2))
plot(GLM6.AIC)

# Running anova Chi test to see if deviance reduced by too much from removing the variables. ()GLM
vs GLM.AIC)
anova(GLM1,GLM1.AIC,test="Chi")
anova(GLM2,GLM2.AIC,test="Chi")
anova(GLM3,GLM3.AIC,test="Chi")
anova(GLM4,GLM4.AIC,test="Chi")
anova(GLM5,GLM5.AIC,test="Chi")
anova(GLM6,GLM6.AIC,test="Chi")

install.packages("visreg")
library(visreg)
```

```
# Multi-model inference for GLMs. Using selected models - The "Use your brain" method:
# Looking at univariate models:
# Also calculate deviance of each model:
M1 <- glm(Groupers~Reef.size, poisson)
summary(M1)
visreg(M1, scale='response', xlab=expression(paste("Reef surface area (Natural log)(km"^"2"*")")),
ylab="Grouper species")
# 1-(5.9246/14.9076)x100= 60.26%

M2 <- glm(Groupers~Fishing, poisson)
summary(M2)
visreg(M2, scale='response', xlab="Fishing absence(0)/presence(1))", ylab="Grouper species")
# 1-(10.988/14.908)x100 = 26.30%

M3 <- glm(Groupers~Mean.isolation, poisson)
summary(M3)
visreg(M3, scale='response', xlab="Reef mean isolation (Square root)(km)", ylab="Grouper species")
# 1-(14.875/14.908)x100=  0.22%

M4 <- glm(Groupers~Nearest.Larger.reef, poisson)
summary(M4)
visreg(M4, scale='response', xlab="Proximity of nearest larger reef (Natural log)(km)",
ylab="Grouper species")
# 1-(13.791/14.908 )x100=  7.4%

M5 <- glm(Groupers~Watersports, poisson)
summary(M5)
visreg(M5, scale='response', xlab="Watersports absence(0)/presence(1)", ylab="Grouper species")
# 1-(7.395/14.908)x100= 50.40%

M6 <- glm(Groupers~Scuba, poisson)
summary(M6)
visreg(M6, scale='response', xlab="Proportion of year with Scuba diving (%)", ylab="Grouper
species")
# 1-(13.622/14.908)x100= 8.63%

M7 <- glm(Groupers~Tmax, poisson)
summary(M7)
visreg(M7, scale='response', xlab=expression(paste("Maximum mean daily temperature (Natural
log)("^"o"*"c)")), ylab="Grouper species")
# 1-(9.1127/14.9076)x100= 38.09%

M8 <- glm(Groupers~Moray.Eels, poisson)
summary(M8)
visreg(M8, scale='response', xlab="Moray Eel density (^6 + Square Root)(individuals per ha)",
ylab="Grouper species")
# 1-(7.6184/14.9076)x100= 48.9%

M9 <- glm(Groupers~Reefsharks, poisson)
summary(M9)
visreg(M9, scale='response', xlab=expression(paste("Grey reef shark density (individuals per
km"^"2"*")")), ylab="Grouper species")
# 1-(14.513/14.908)x100= 2.65%

M10 <- glm(Groupers~Coral..simpson, poisson)
summary(M10)
visreg(M10, scale='response', xlab="Coral simpson Index(Square root)", ylab="Grouper species")
# 1-(8.0259/14.9076)x100= 46.16%

library(bbmle)
# Check AIc models:
> AICctab(GLM1.AIC,GLM2.AIC, GLM3.AIC, GLM4.AIC,GLM5.AIC,GLM6.AIC, base = T, weights = T, nobs =
length(Groupers))
# Check all GLM.AIC and M models:
> AICctab(GLM1.AIC,GLM2.AIC,GLM3.AIC,GLM4.AIC,GLM5.AIC, GLM6.AIC, M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,
base = T, weights = T, nobs = length(Groupers))
# Look at the best model:
library(visreg)
visreg(GLM2.AIC)
```