Modern Regression

A Predictive Model for Coral Bleaching Risk with Random Forest

Michael Harrison Lee

Thursday, April 12, 2018

Problem:

Coral reefs are one of the most biodiverse ecosystems on the planet, providing a habitat for millions of fish and other aquatic species. Many people depend on reefs, especially local communities which thrive off of subsistence fishing. Due to worsening environmental stressors such as pollution and global warming, many coral reefs undergo "bleaching," a process where corals turn white, die, and drastically disrupt local ecosystems. Both anthropogenic and natural influences are to blame, many of which are out of our control. However, if could we could find a way to identify at-risk reefs in advance, then perhaps we could take preventative measures. Is it possible to predict coral bleaching before it happens?

The Reef Check Coral Reef Monitoring Program:

Reef Check was started in 1996 to monitor coral reefs through the use of a single, standardized surveying methodology. The project has grown to include thousands of scientists and local volunteers who are committed to stopping the decline of coral reef ecosystems around the world, and it is now the most widely used coral reef monitoring protocol. Three kinds of data are collected: 1) features of the reef and its surroundings, 2) bioindicator organisms in the general environment, and 3) bioindicator organisms on the ocean floor.

The Reef Check dataset we will be examining contains information of the first type on 12,392 different reefs surveyed from 1997-2017. The response is "Bleaching," a binary variable indicating whether or not bleaching has been observed at the reef. There are 11 predictor variables (Ocean, Year, Depth, Storms, HumanImpact, Siltation, Dynamite, Poison, Sewage, Industrial, Commercial) all of which describe features of the surrounding marine environment.

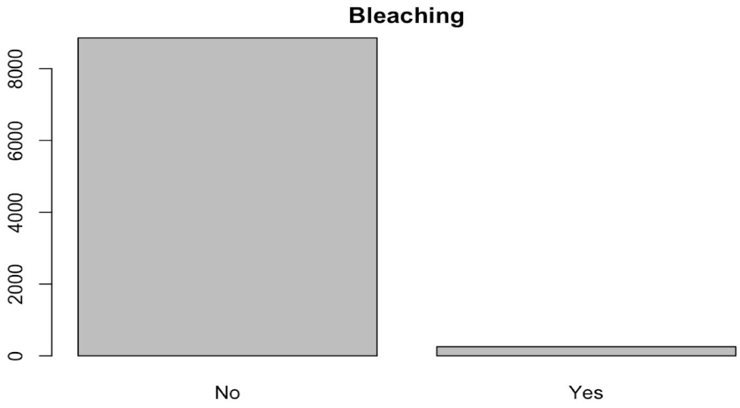
Data Cleaning and Preparation:

Upon "naked-eye" examination of the data, it is apparent that there are many missing values for factor variables. Due to the large number of observations, we feel comfortable removing rows with unknown values in order to train an honest classifier which can be used for future prediction. However, according to the project documentation, many of the missing values actually correspond to the factor level "none," in which case it should not be removed. To sort this out, we examined the unique levels for each factor and changed the whitespace to "unknown" if there was already an "unknown" factor level. If an "unknown" level did not exist, blank values were re-coded to the lowest level in the factor (e.g. "none" or "never").

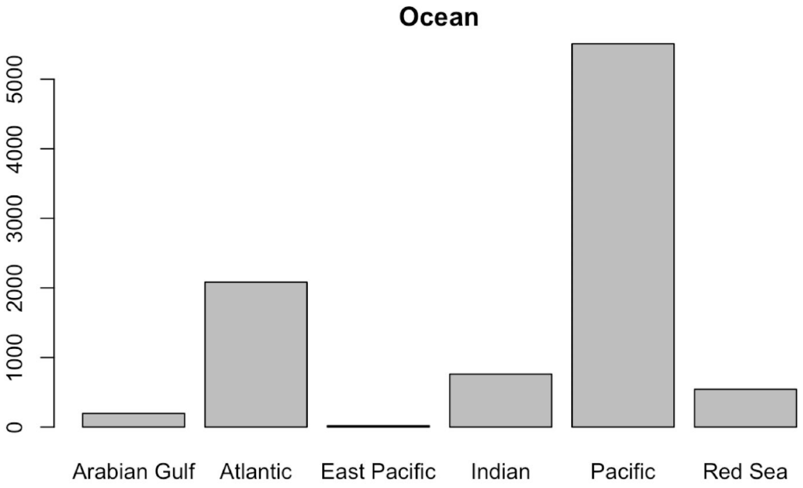
After recoding missing values, all rows with “unknown” data were removed. Fortunately, the dataset is still of sufficient size (>8000 observations, ~12000 originally), and an acceptable ratio of positive to negative response cases has been preserved. There were also some strange factor levels which either did not make sense or were clearly data entry errors; these were changed (Siltation = “Occasionally” > “occasionally”) or deleted (Sewage = “k”, Dynamite = “prior”) on a case-by-case basis.

Univariate Statistics:

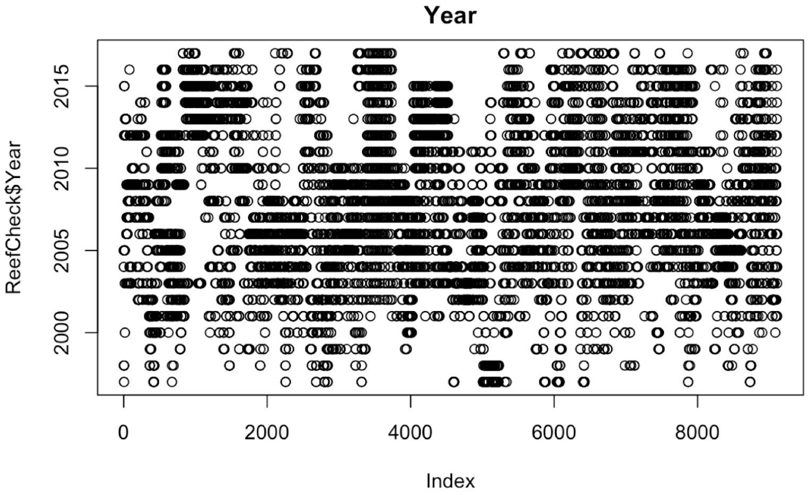
Let’s take a look at the distribution of observations among each of our variables, starting with the response.



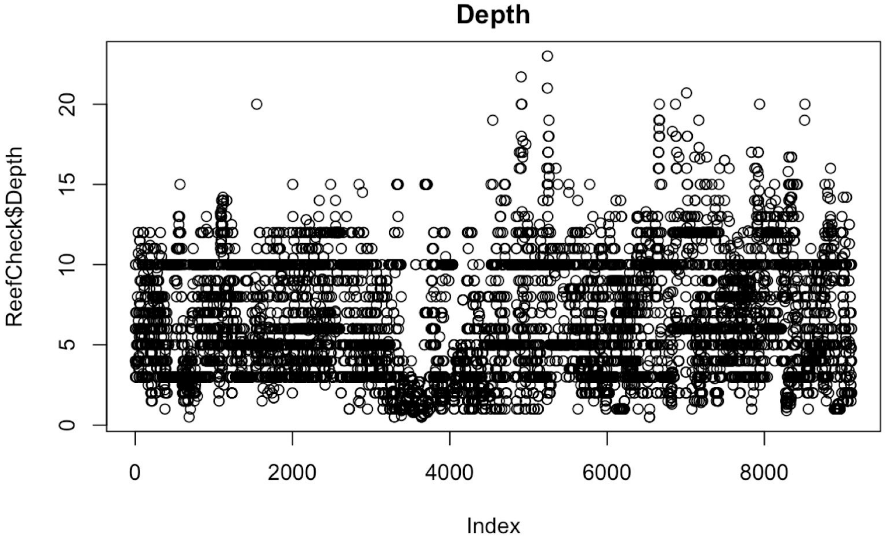
The vast majority of coral reefs experience no bleaching.



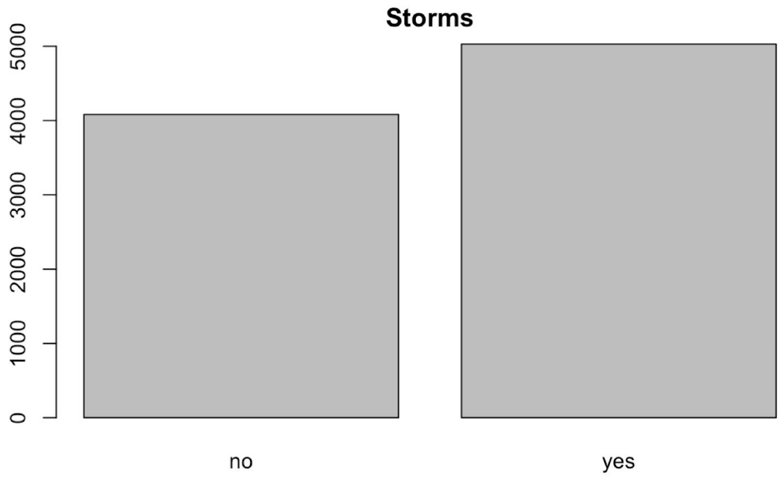
Most of our observations come from the Pacific and Atlantic Oceans, which may make our model slightly more suitable for predictions in those regions.



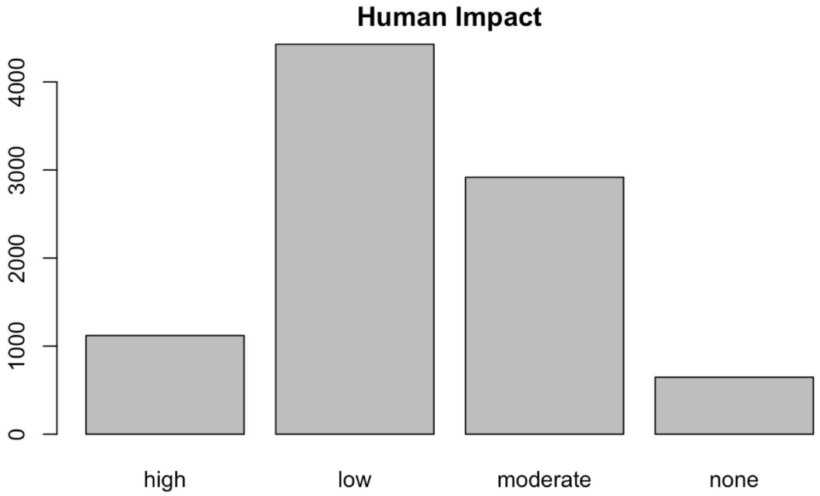
Observations are more or less evenly distributed over time, which bodes well for our inference.



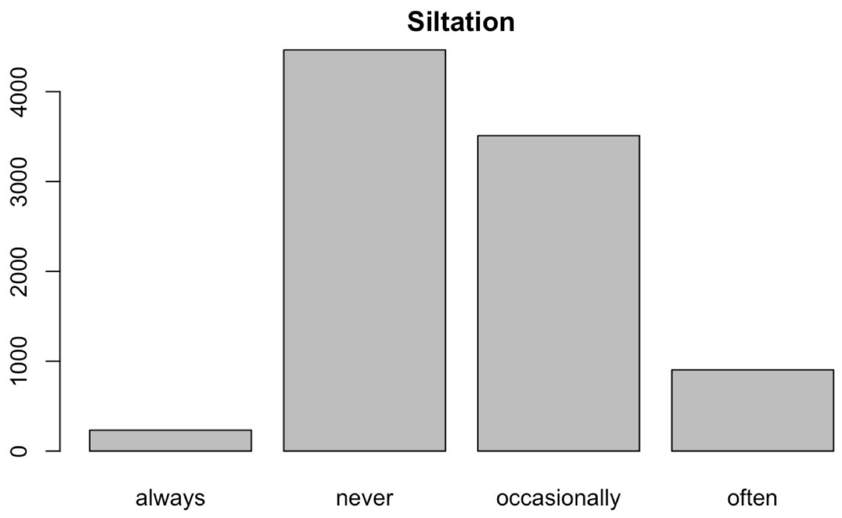
Most of the reefs in our dataset tend to be anywhere from 2.5-12.5 meters deep.



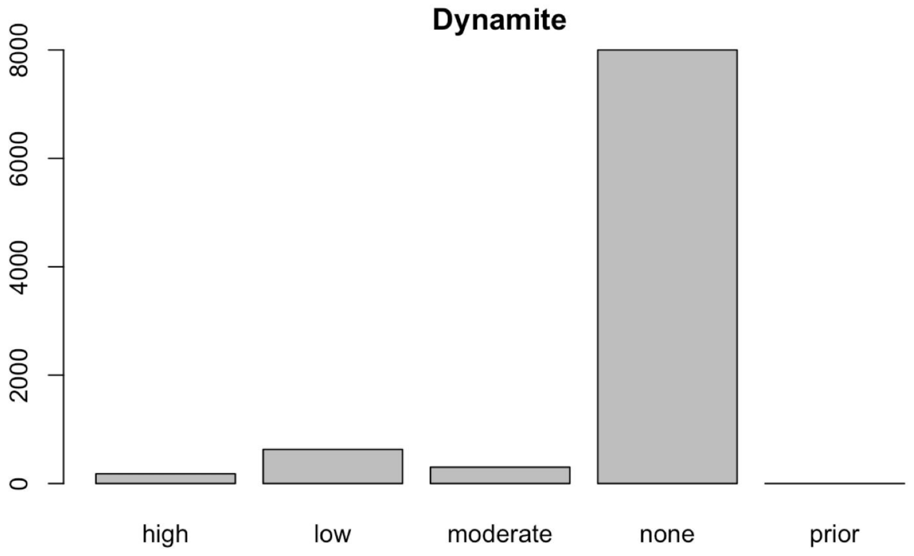
Most of our sites have seen a major storm in the past year; however, a similar amount have not.



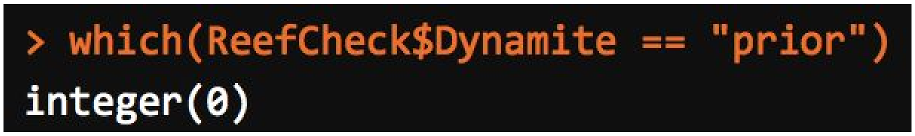
Very few reefs in our dataset have remained entirely unaffected by human activities.



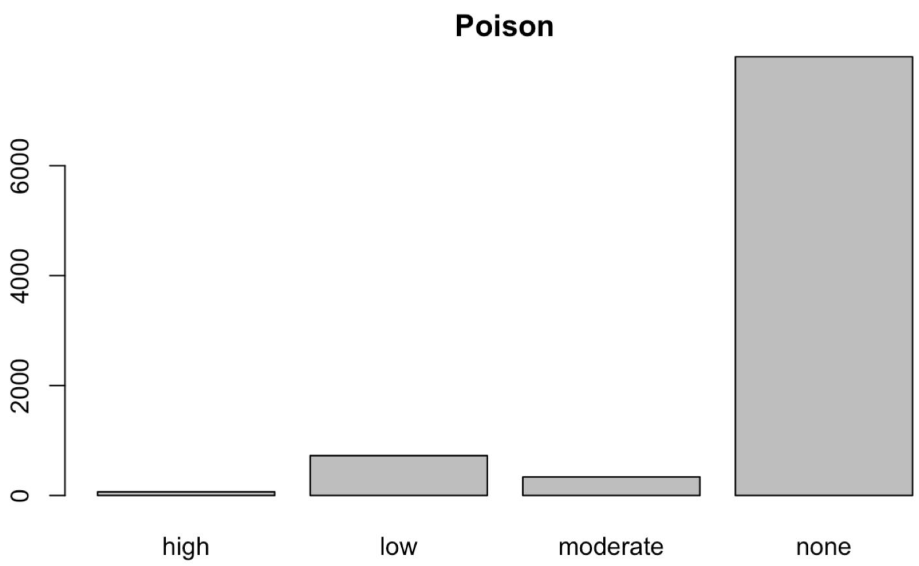
Most of our sites either "never" or "occasionally" have silt.



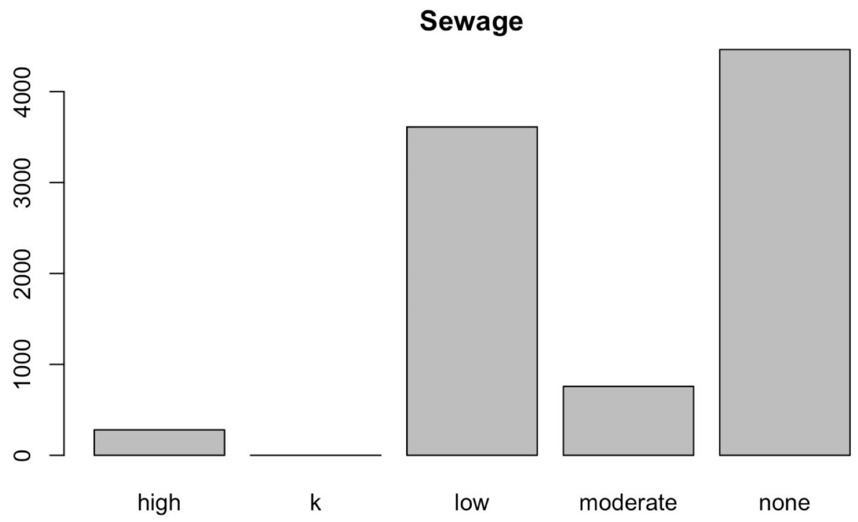
Most sites have never been fished with dynamite. Ignore the "prior" level. It was removed from the dataset but still appears as a level in the plot:



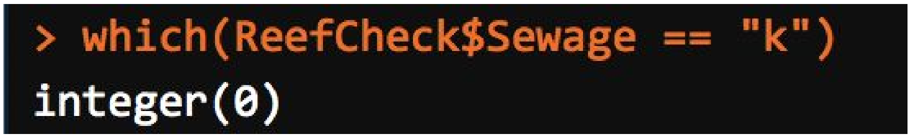
```
> which(ReefCheck$Dynamite == "prior")
integer(0)
```



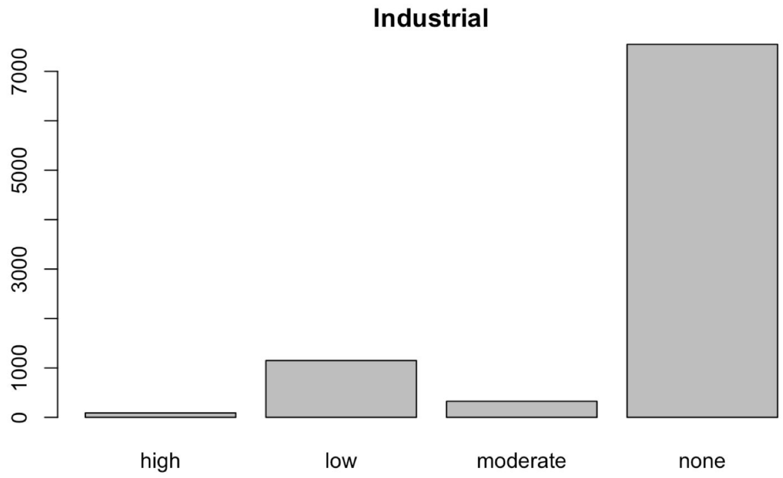
Most of the reefs in our dataset have not been fished with poison.



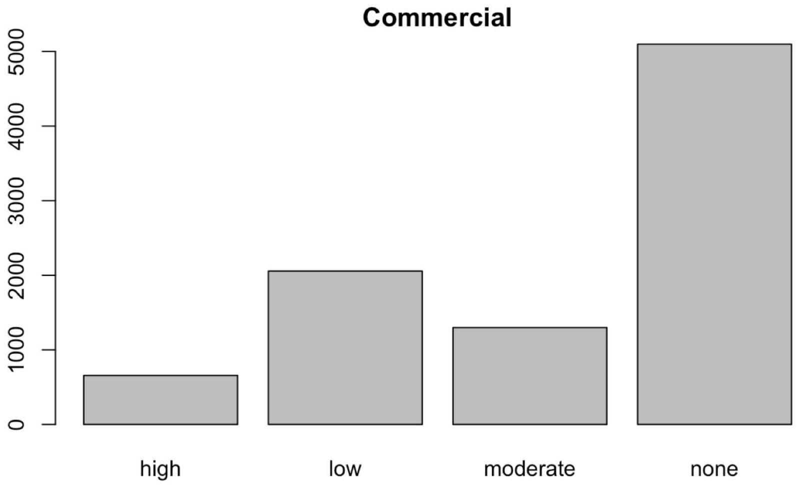
However, a significant portion of reefs have been exposed to moderate or low levels of sewage. Once again, please ignore the "k" level. It was removed from the dataset completely still appears in the plot:



```
> which(ReefCheck$Sewage == "k")
integer(0)
```



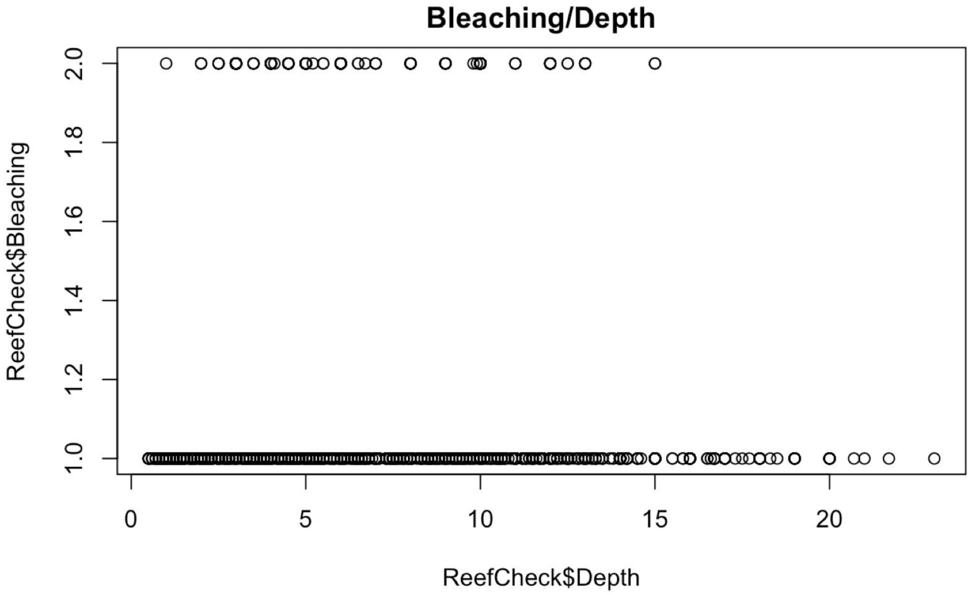
Most of the corals in our dataset have been exposed to little or no industrial pollution.



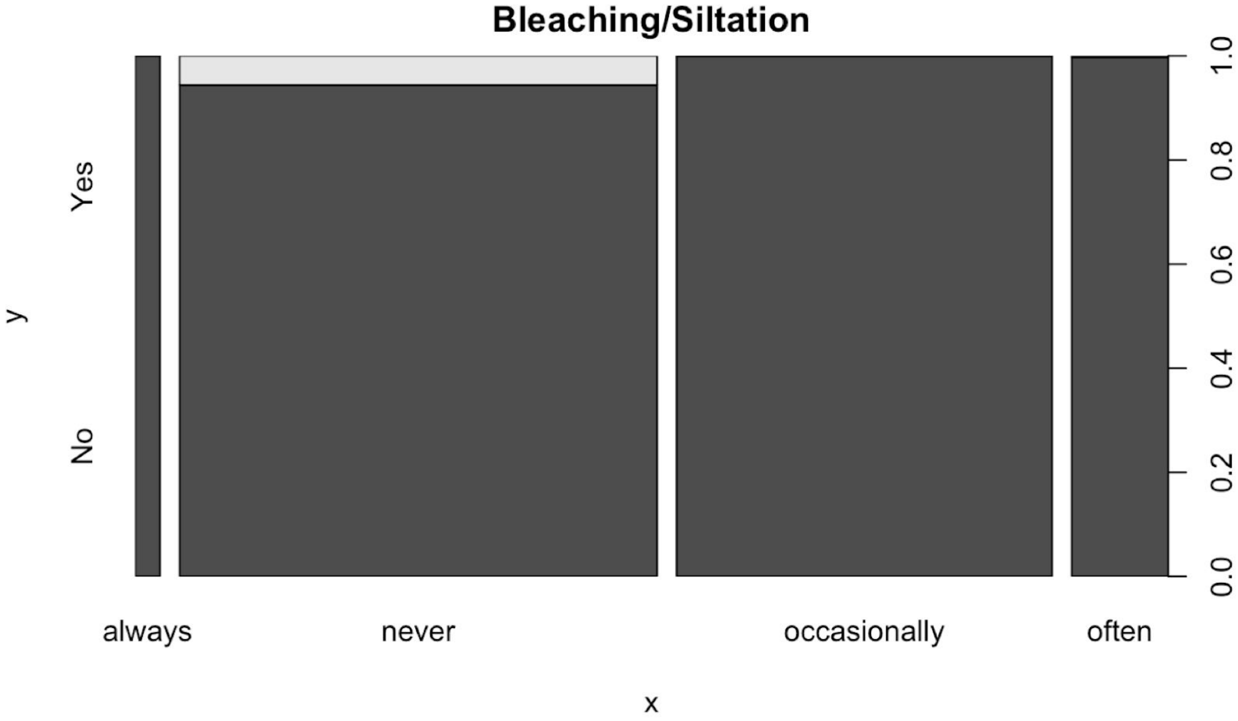
Most reefs have not been commercially fished or have been fished very little. There are only a few hundred reefs in our dataset that have been heavily fished.

Bivariate Statistics:

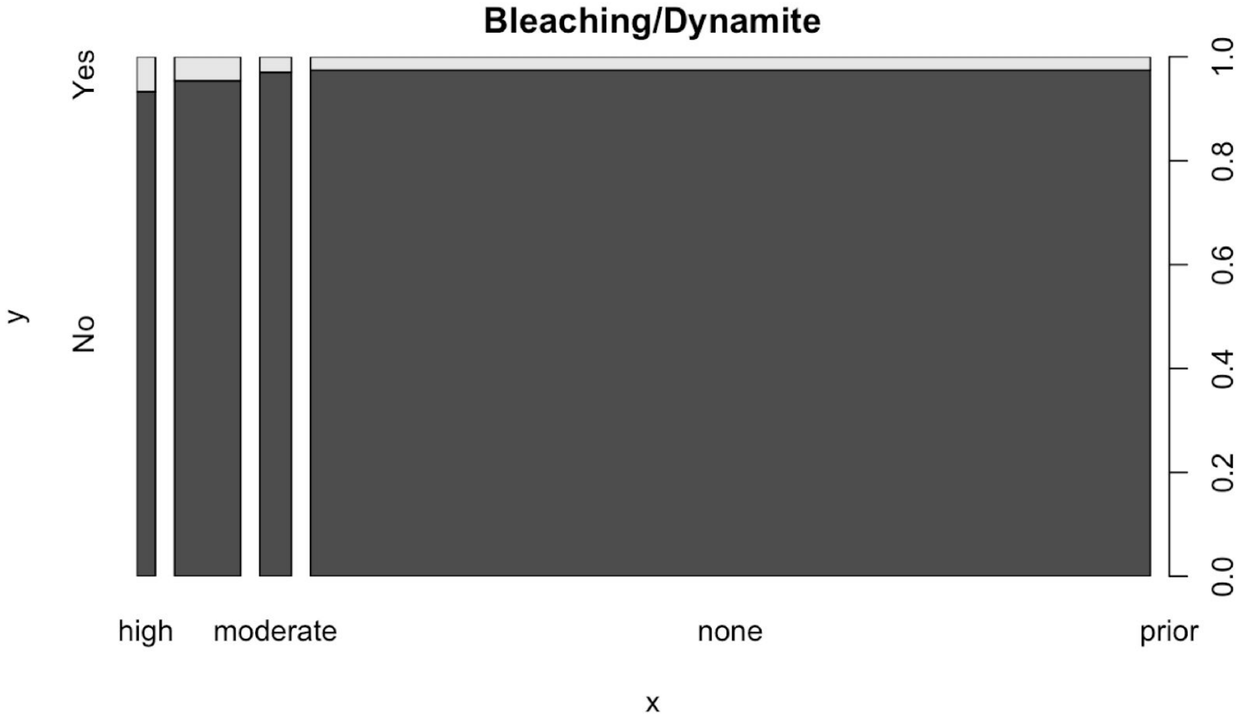
Before proceeding, here are a few interesting relationships between some of our predictors and the response.



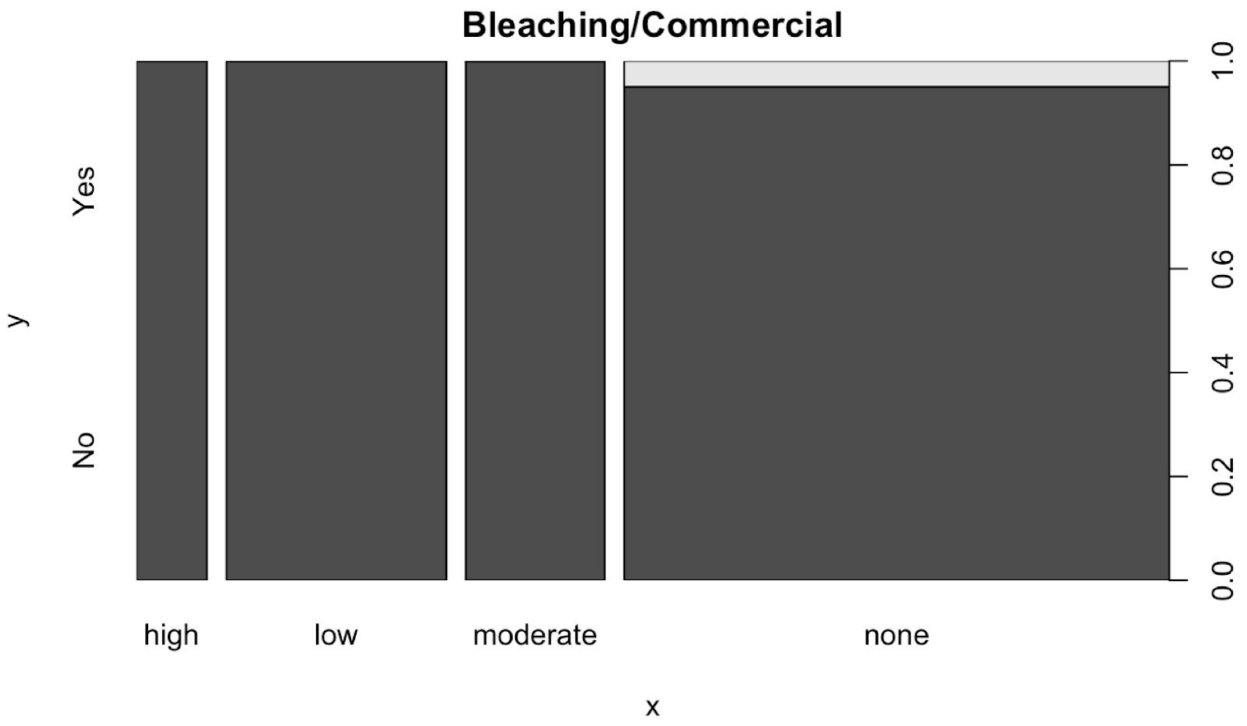
Bleaching tends to occur at shallower depths.



Reefs which never have silt bleach more often.



Sites with moderate and high Dynamite usage tend to have more bleaching.



Surprisingly, reefs with no commercial fishing tend to bleach more often.

Core Assumptions:

We will be attempting to predict coral reef bleachings in the future, so it is necessary to define the underlying assumptions of our model and stress its limitations. To justify this type of statistical inference, the data upon which the algorithm is trained must be representative of the underlying joint probability distribution for the natural phenomenon. Given that the data were collected over two decades from many of the world's oceans, this is not a hard sell; the diverse geographies, long time horizon, and sheer number of observations are reasons enough to believe that we have adequately captured a "snapshot" of the natural process. Admittedly, there is a bias towards reefs in the Atlantic and Pacific oceans, so our fitted model may have more predictive relevance in those areas. Likewise, environmental conditions may be very different now than they were 50 or even 20 years ago. However, we have good reason to believe that nature's "mechanisms" have remained unchanged over time. That is, the biophysical and biochemical laws which underpin ecological processes such as coral bleaching do not change. Under this core assumption, we can proceed with a "level 2" analysis.

Tuning Objectives:

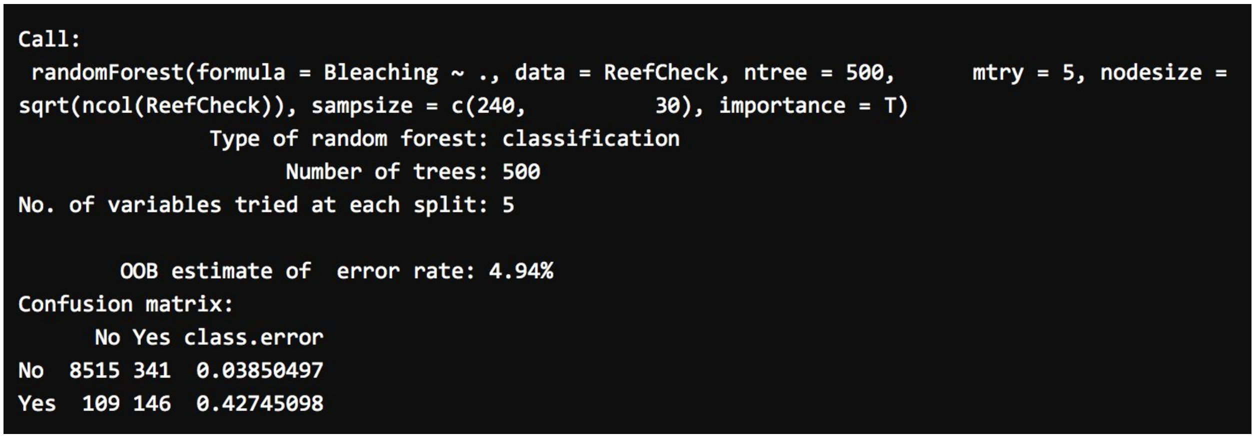
Page 45 of the NOAA Reef Check Manual warns of embracing null hypotheses of "no change expected" for reefs. Enacting a MPA (Marine Protected Area) frequently restricts all human activities in the site to research, which can have short-term economic consequences for local communities. However, the long term economic consequences of failing to take action may be much worse; a dead reef cannot be fished, nor do bleached corals attract snorkeling tourists. Of course, it would be impractical to make the test overly sensitive, as MPA program managers have limited resources and cannot tend to every reef at once.

Considering these factors, we will be imposing a cost ratio of 1:3, where a false negative (no bleaching predicted, bleaching observed) is considered to be 3 times as costly as a false positive (bleaching predicted, no bleaching preserved).

Algorithm Training and Performance:

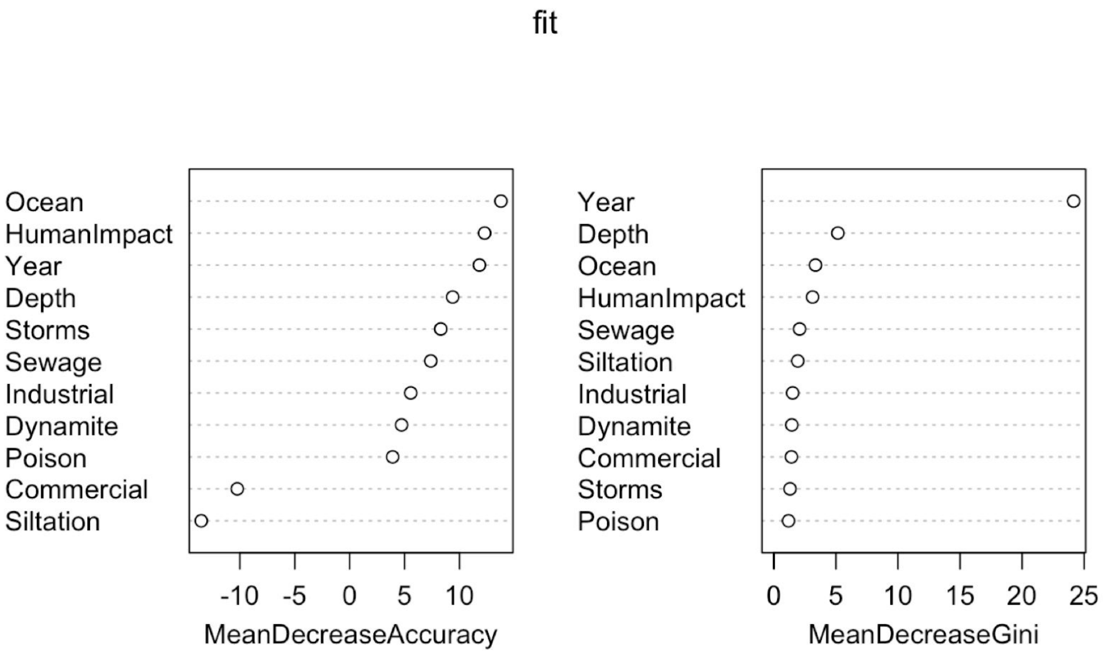
Random forest and other bagging algorithms have an interesting feature that allows them to avoid data snooping during the training process. While growing trees over random re-samplings of data, random forest leaves roughly ⅓ of that data untouched by the procedure. This is known as the out of bag (OOB) sample, and can be used as a test set to evaluate model fit and performance. Performance estimates using OOB data are built into the randomForest package in R, which we will be using. In other words, no training, evaluation, and test datasets will be needed for this analysis.

Values of mtry(number of variables to randomly try at each split) from 1 to 11 were cycled through repeatedly to minimize OOB misclassification rate. Number of trees grown was fixed at 500 to achieve as much gain in accuracy as possible without sacrificing computer efficiency. Node size was set to the square root of the number of predictors, and values for "sampsize" were tuned empirically to arrive at a cost ratio close to our target. Our OOB error rate is 4.94%.

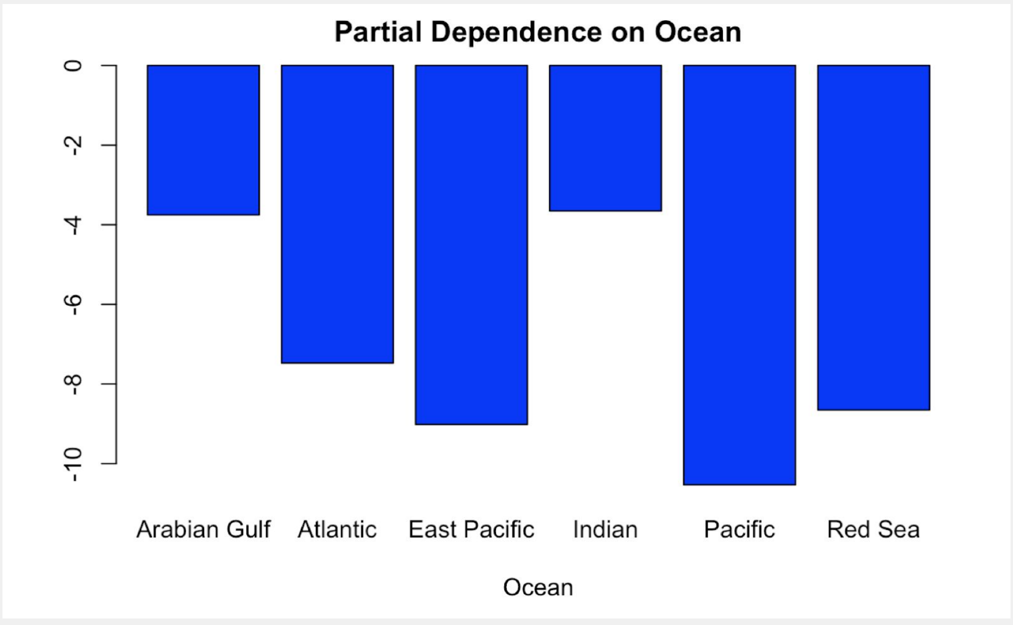


```
Call:
 randomForest(formula = Bleaching ~ ., data = ReefCheck, ntree = 500,      mtry = 5, nodesize =
sqrt(ncol(ReefCheck)), sampsize = c(240,          30), importance = T)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 5

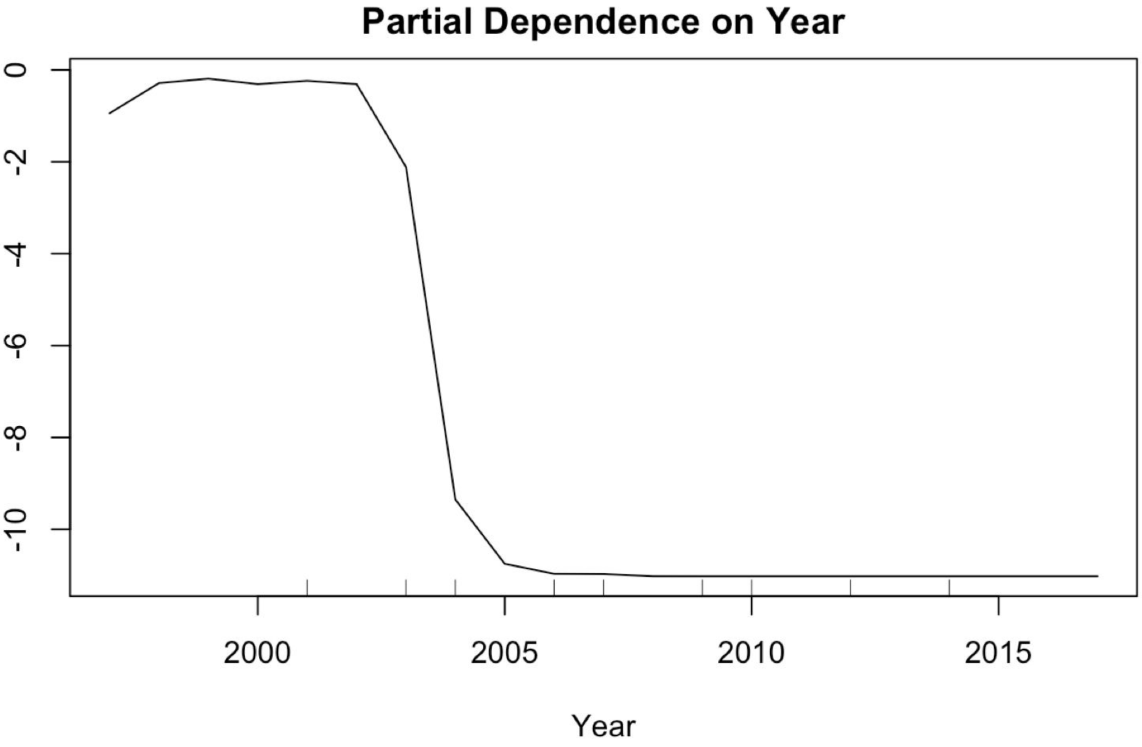
        OOB estimate of  error rate: 4.94%
Confusion matrix:
      No Yes class.error
No  8515 341  0.03850497
Yes  109 146  0.42745098
```



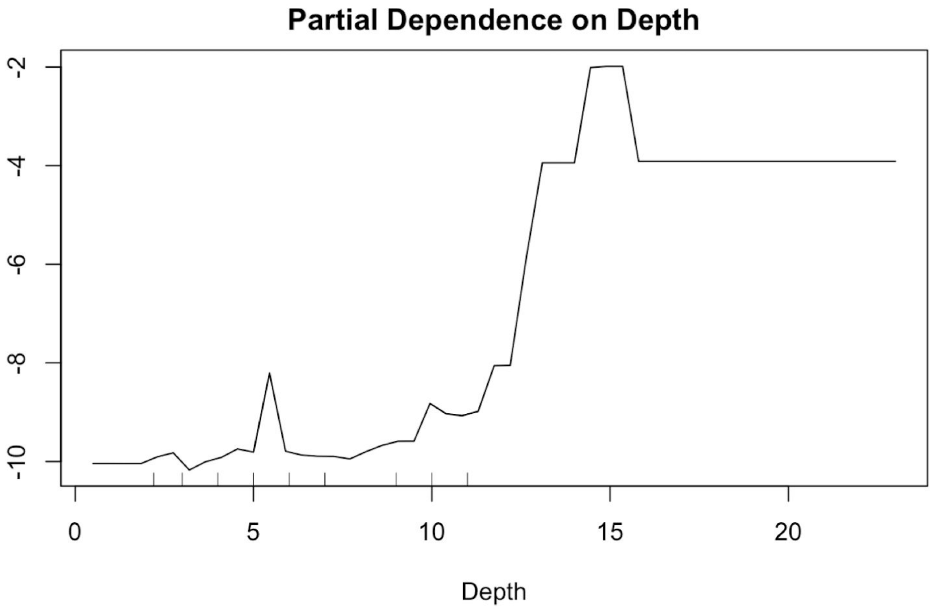
We can see from the variable importance plot that removing anthropogenic factors such as Human Impact, Sewage, Industrial, Dynamite, and Poison seems to significantly increase our OOB error rate. However, it appears that the predictive power of the model is explained by both natural and anthropogenic factors with Ocean, Year, Depth, and Storms also having a heavy impact on accuracy of fit.



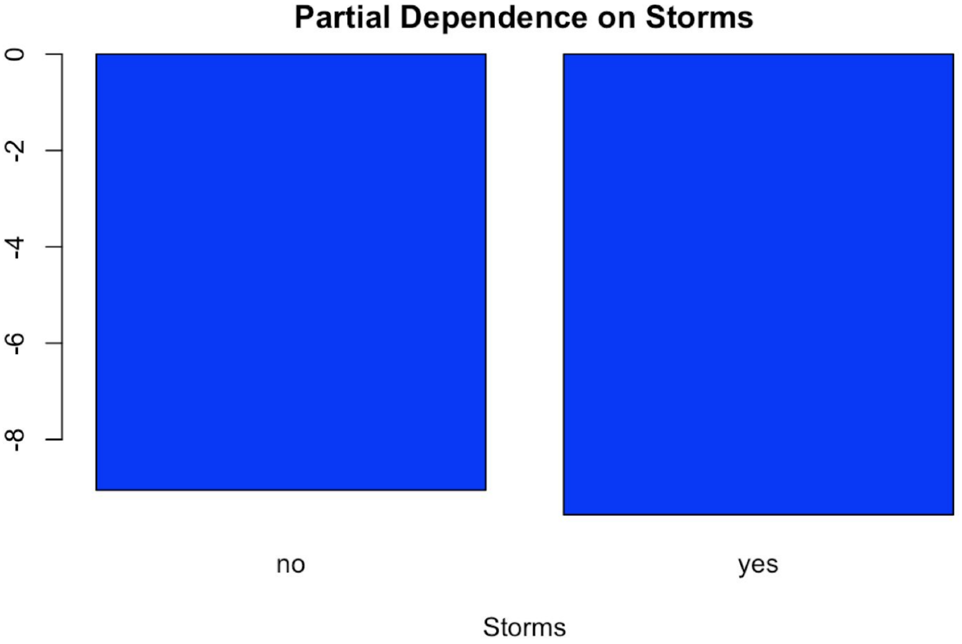
Reefs in the Arabian Gulf and Indian ocean tend to predict higher rates of bleaching, while the Pacific Ocean tends to predict lower rates, holding all else constant.



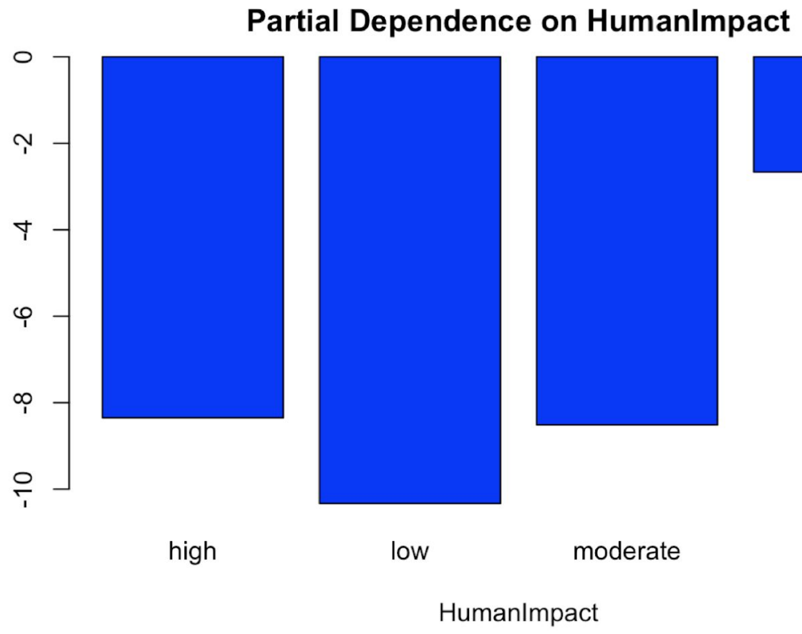
Bleaching predictions appear to depend heavier on Year when the year is before 2005. Years after 2005 tend to predict fewer instances of bleaching.



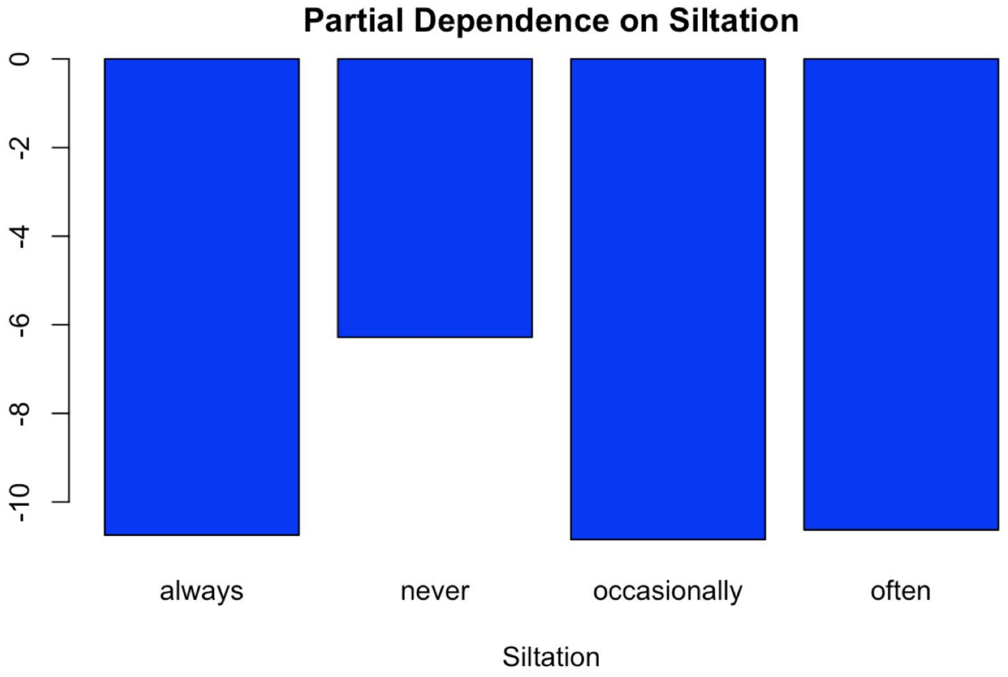
Depths greater than 11 meters tend to predict greater rates of bleaching, holding all else constant.



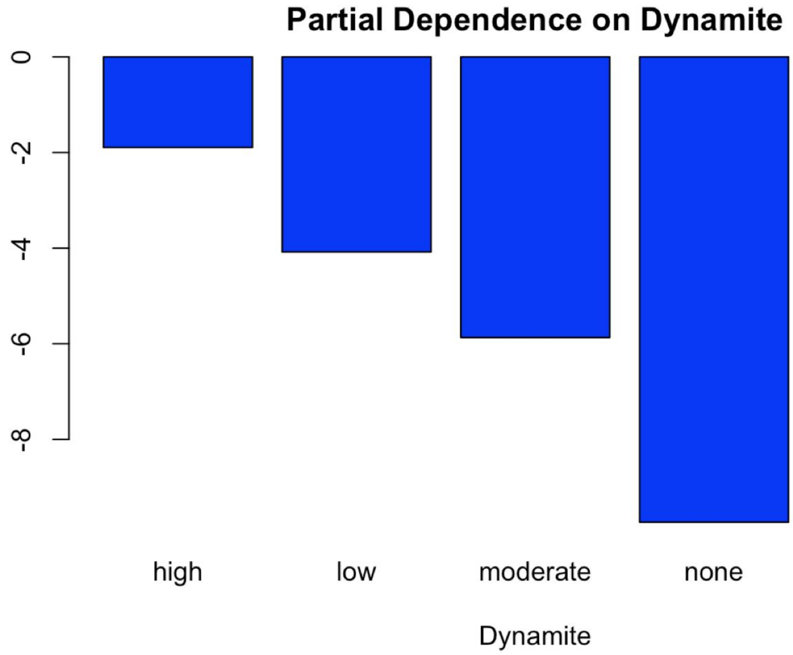
Partial dependence on storms appears to be relatively uniform. There seems to be very little dependence on this variable.



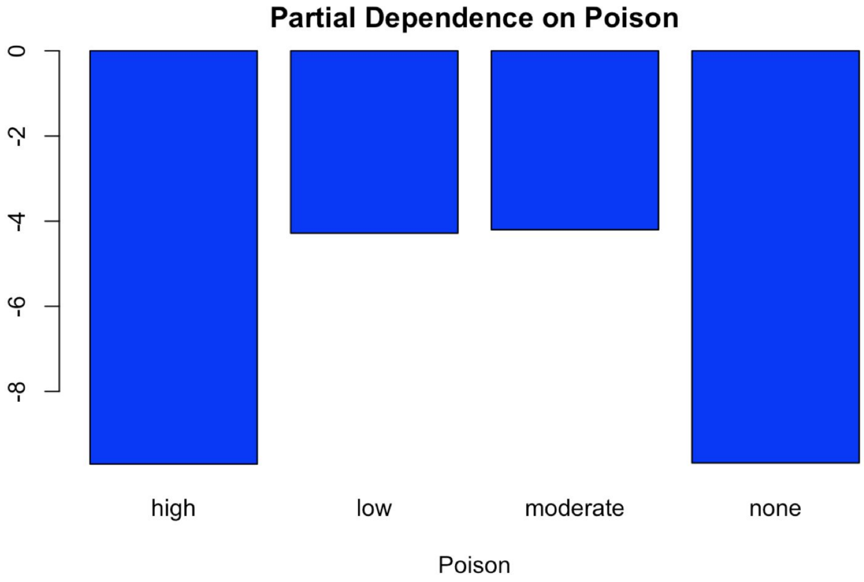
Lack of human impact on the reef tends to predict bleaching most strongly within this factor.



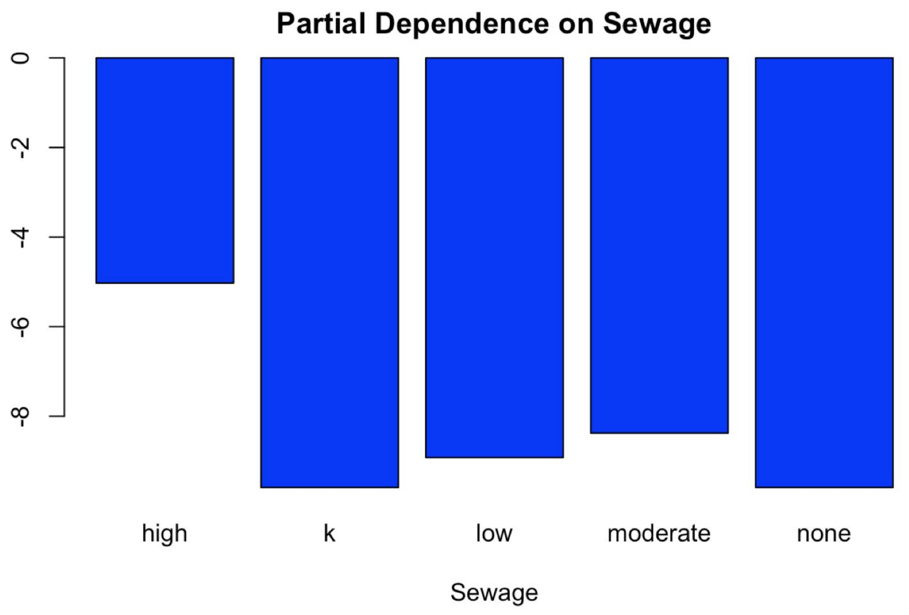
Lack of siltation tends to predict bleaching more strongly.



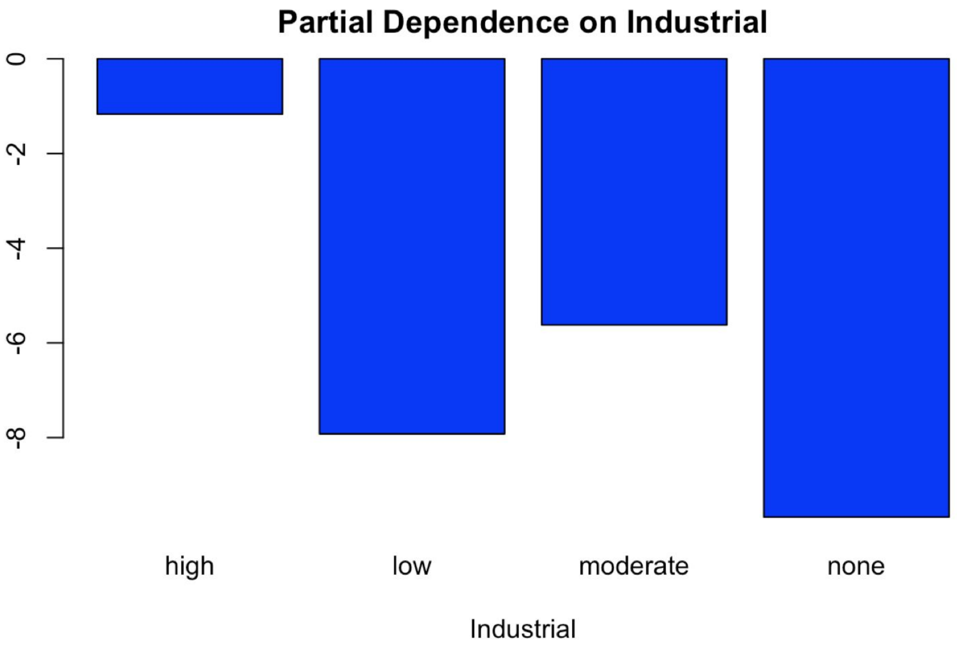
Frequent use of dynamite for fishing very strongly predicts bleaching, holding all else constant.



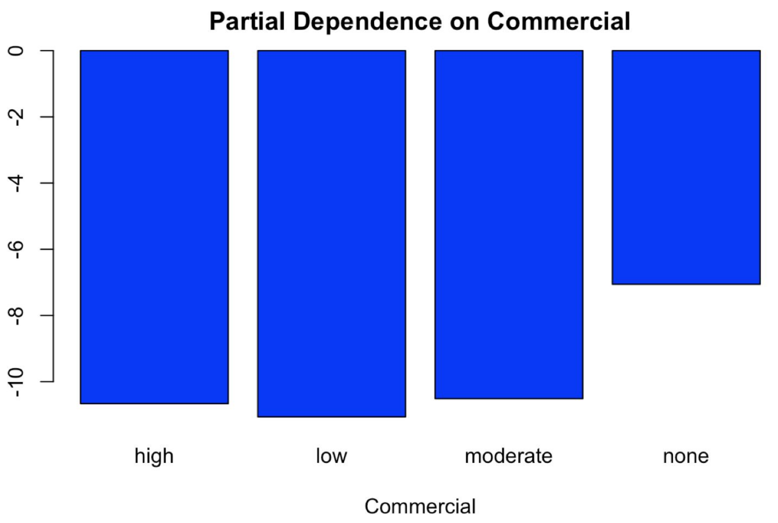
Low and moderate use of poison for fishing tends to predict bleaching more strongly.



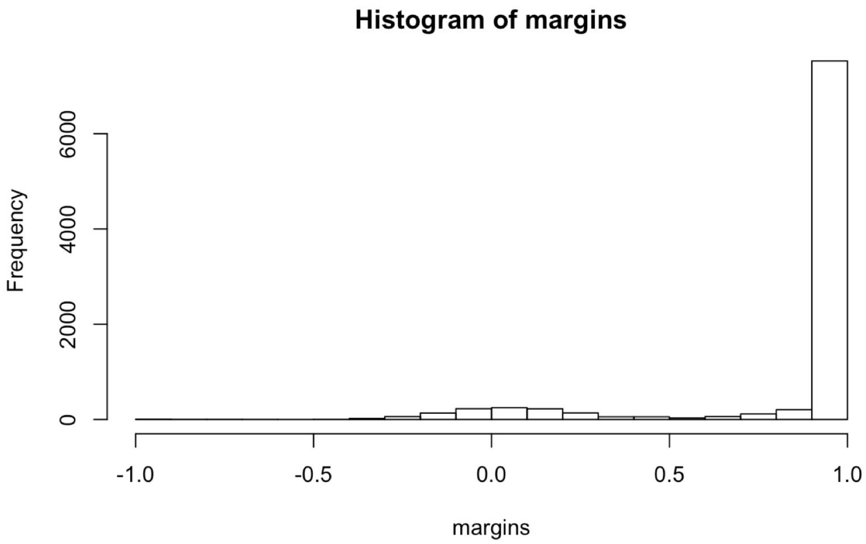
High sewage levels tend to predict bleaching more strongly.

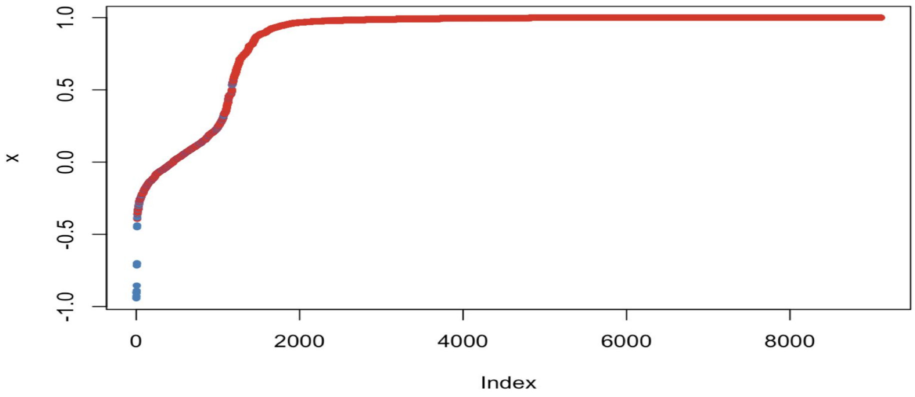


High levels of industrial pollution predict bleaching very strongly.



Zero commercial fishing seems to predict bleaching more strongly, but not by a substantial amount.





Conclusion:

We have found that coral bleaching can be successfully predicted by our model 95.06% of the time; however, this is only in terms of OOB misclassification rate. Sensitivity (true positive rate) is 57.25%, while specificity (true negative rate) sits at 96.14%. Looking back, tuning our algorithm with a more aggressive cost ratio (maybe 1:20) would perhaps have increased its sensitivity enough to be deployable as a stand-alone tool. We do not recommend its current form for the prescription of MPAs.

However, this does not mean the model is completely useless. In fact, the high specificity of the model may make it more useful for decisions involving reef recovery efforts, after a site has already been discovered as "at risk." If the model is very good at predicting "no bleach," then a manager might use it to determine whether the reef is safe to re-open to the public.

A follow-up study might include re-tuning the random forest with a better cost ratio for the task at hand and taking multiple surveys of the same reefs in this dataset for future analysis; the latter may provide a clear picture of which predictors lead and lag with regard to coral bleaching.

Code Appendix:

~~~
```{r}
#Dear Computer,

install.packages("randomForest") #get randomForest
library(randomForest)
```

```{r}
#get the data
rm(list=ls()) #clear the global environment
load("~/Documents/Coding/Data/ReefCheck974.rdata") #load the data
head(ReefCheck) #observe the rows & data types
ncol(ReefCheck) #12
nrow(ReefCheck) #12392
min(ReefCheck$Year);max(ReefCheck$Year)
head(ReefCheck)
```

```{r}
#clean the data

#ReefCheck$Bleaching
#ReefCheck$Ocean
#ReefCheck$Year
#ReefCheck$Depth
#ReefCheck$Storms
#ReefCheck$HumanImpact
#ReefCheck$Siltation
#ReefCheck$Dynamite
#ReefCheck$Poison
#ReefCheck$Sewage
#ReefCheck$Industrial
#ReefCheck$Commercial

#complete.cases(ReefCheck)
#dataset is read by r as complete despite blank values
#blanks will be assumed to be "none"
~~~

```
#all of the variables will be considered as important
#all of the varialbe types are appropriate, so we will keep them

#given that blank is a legitimate level for a factor, we will recode these values to "none"

any(ReefCheck$Bleaching == "")
unique(ReefCheck$Bleaching) #fine

any(ReefCheck$Ocean == "") #check for missing values, unable to fully clean unknowns
unique(ReefCheck$Ocean) #examine factor values to determine a reasonable recoding for missing values
ReefCheck$Ocean = as.character(ReefCheck$Ocean) #convert to character to replace values with strings
ReefCheck$Ocean[(ReefCheck$Ocean == "")] = "unknown" #change to "unknown"
# assumption is that blank values here correspond to the "unknown" factor level

any(ReefCheck$Year == "")
unique(ReefCheck$Year) #fine, leave as integer

any(ReefCheck$Depth == "") #fine, leave as numerical

any(ReefCheck$Storms == "") #unable to fully clean unknowns
unique(ReefCheck$Storms)
ReefCheck$Storms = as.character(ReefCheck$Storms)
ReefCheck$Storms[(ReefCheck$Storms == "")] = "unknown"
ReefCheck$Storms[(ReefCheck$Storms == "y")] = "yes" #lets also convert the weird y to a yes

any(ReefCheck$HumanImpact == "") #unable to fully clean unknowns
unique(ReefCheck$HumanImpact)
ReefCheck$HumanImpact = as.character(ReefCheck$HumanImpact)
ReefCheck$HumanImpact[(ReefCheck$HumanImpact == "")] = "unknown"

any(ReefCheck$Siltation == "")
unique(ReefCheck$Siltation)
ReefCheck$Siltation = as.character(ReefCheck$Siltation)
ReefCheck$Siltation[(ReefCheck$Siltation == "")] = "never"
ReefCheck$Siltation[(ReefCheck$Siltation == "Occasionally")] = "occasionally"

any(ReefCheck$Dynamite == "")
```

```
unique(ReefCheck$Dynamite)
ReefCheck$Dynamite = as.character(ReefCheck$Dynamite)
ReefCheck$Dynamite[(ReefCheck$Dynamite == "")] = "none"

any(ReefCheck$Poison == "") #unavble to fully clean unknowns
unique(ReefCheck$Poison)
ReefCheck$Poison = as.character(ReefCheck$Poison)
ReefCheck$Poison[(ReefCheck$Poison == "")] = "unknown"

any(ReefCheck$Sewage == "")
unique(ReefCheck$Sewage)
ReefCheck$Sewage = as.character(ReefCheck$Sewage)
ReefCheck$Sewage[(ReefCheck$Sewage == "")] = "none"

any(ReefCheck$Industrial == "")
unique(ReefCheck$Industrial)
ReefCheck$Industrial = as.character(ReefCheck$Industrial)
ReefCheck$Industrial[(ReefCheck$Industrial == "")] = "none"

any(ReefCheck$Commercial == "")
unique(ReefCheck$Commercial)
ReefCheck$Commercial = as.character(ReefCheck$Commercial)
ReefCheck$Commercial[(ReefCheck$Commercial == "")] = "none"

#now let's remove all of the unknowns

unknown.indices = c()
for(i in 1:nrow(ReefCheck)){
  if(any(ReefCheck[i,] == "unknown")){
    unknown.indices = append(unknown.indices, i)
  }
}
length(unknown.indices)
ReefCheck = ReefCheck[-unknown.indices,]

#convert back to factors
ReefCheck$Ocean = as.factor(ReefCheck$Ocean)
ReefCheck$Storms = as.factor(ReefCheck$Storms)
ReefCheck$HumanImpact = as.factor(ReefCheck$HumanImpact)
```

~~~
ReefCheck$Siltation = as.factor(ReefCheck$Siltation)
ReefCheck$Sewage = as.factor(ReefCheck$Sewage)
ReefCheck$Dynamite = as.factor(ReefCheck$Dynamite)
ReefCheck$Poison = as.factor(ReefCheck$Poison)
ReefCheck$Industrial = as.factor(ReefCheck$Industrial)
ReefCheck$Commercial = as.factor(ReefCheck$Commercial)

which(ReefCheck$Sewage == "k") #these are weird so I'm just going to remove them
ReefCheck = ReefCheck[-which(ReefCheck$Sewage == "k"),]

which(ReefCheck$Dynamite == "prior")
ReefCheck = ReefCheck[-which(ReefCheck$Dynamite == "prior"),]

length(which(ReefCheck$Bleaching == "No")) #should we remove the unknowns?
length(which(ReefCheck$Bleaching == "Yes"))

#the resulting proportion of negative to positive response cases is quite acceptable, so we can remove the unknowns

any(ReefCheck == "unknown")#check
any(ReefCheck == "k")
any(ReefCheck == "prior")

#data clean, ready to go!

```

```{r}
#Univariate statistics

plot(ReefCheck$Bleaching, main = "Bleaching")
#most coral reefs are not bleached

plot(ReefCheck$Ocean, main = "Ocean")
#most cases occur in the pacific oceanA

plot(ReefCheck$Year, main = "Year")
plot(ReefCheck$Depth, main = "Depth")
plot(ReefCheck$Storms, main = "Storms")
~~~

~~~~
plot(ReefCheck$HumanImpact, main = "Human Impact")
plot(ReefCheck$Siltation, main = "Siltation")
plot(ReefCheck$Dynamite, main = "Dynamite")
plot(ReefCheck$Poison, main = "Poison")
plot(ReefCheck$Sewage, main = "Sewage")
plot(ReefCheck$Industrial, main = "Industrial" )
plot(ReefCheck$Commercial, main = "Commercial")
```

```{r}
#bivariate statistics
plot(ReefCheck$Ocean, ReefCheck$Bleaching, main = "Bleaching/Ocean")
plot(ReefCheck$Year, ReefCheck$Bleaching, main = "Bleaching/Year")
plot(ReefCheck$Depth, ReefCheck$Bleaching, main = "Bleaching/Depth")
plot(ReefCheck$Storms, ReefCheck$Bleaching, main = "Bleaching/Storms")
plot(ReefCheck$HumanImpact, ReefCheck$Bleaching, main = "Bleaching/Human Impact")
plot(ReefCheck$Siltation, ReefCheck$Bleaching, main = "Bleaching/Siltation")
plot(ReefCheck$Dynamite, ReefCheck$Bleaching, main = "Bleaching/Dynamite")
plot(ReefCheck$Poison, ReefCheck$Bleaching, main = "Bleaching/Poison")
plot(ReefCheck$Sewage, ReefCheck$Bleaching, main = "Bleaching/Sewage")
plot(ReefCheck$Industrial, ReefCheck$Bleaching, main = "Bleaching/Industrial" )
plot(ReefCheck$Commercial, ReefCheck$Bleaching ,main = "Bleaching/Commercial")
```

```{r}
#now let's fit the random forest

#Set ntree sufficiently high to 500
#set nodesize to sqrt(#predictors)
#tune mtry with some sort of procedure
#empirically tune sampsize
#set importance = T

fit <- randomForest(Bleaching ~., data = ReefCheck, ntree = 500, mtry = 5, nodesize =
sqrt(ncol(ReefCheck)), sampsize = c(240,30), importance = T)
fit

varImpPlot(fit)
~~~~

~~~~
partialPlot(fit, pred.data = ReefCheck, x.var = Ocean, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = Year, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = Depth, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = Storms, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = HumanImpact, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = Siltation, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = Dynamite, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = Poison, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = Sewage, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = Industrial, which.class = "Yes",rug = F)
partialPlot(fit, pred.data = ReefCheck, x.var = Commercial, which.class = "Yes",rug = F)

margins = margin(fit)
hist(margins)
plot(margins,sort=T)

```
~~~~