Rules of Thumb for Species Occupancy Models

Mark Logie, Tom August & Michael Pocock

21 February, 2019

Contents

Built with 3.4.1

1 Introduction

The aim of this document is to start development for rules of thumb around occupancy models. Which species can we make reliable models for? How many records do we need? How large an area? how many visits? Having a greater understanding of how these variables effect the quality of the models produced will allow us to develop rules of thumb for which species can be modelled and at which scale. This will also help us to identify key data gaps.

2 Metrics

To develop these rules of thumb we need two sets of metrics.

2.1 Data quality metrics

The metrics describe the data that we have for a species and can be used to predict the quality of the model:

- The 50th/90th percentile of the number of records of the focal species per year [`median/P90`]
- The 50th/90th percentile of the number of visits to a site each year, for sites where the species has been observed in that year (i.e. including visits where the focal species was not recorded) [`visits_median/P90`]
- The proportion of all site:year combinations, where the focal species is observed, that have > 1 visit [`prop_repeats_grp`]
- Considering all the lists where the focal species was recorded, the proportion of lists that had length 1 (i.e. records only of the focal species) [`prop_list_one`]
- Considering all visits for the 'taxonomic' group in the dataset, the proportion of all visits that did not record the focal species [`prop_abs`]
- Considering the visits for the 'taxonomic' group where the focal species was not observed, this is the proportion of visits with list length > 1 [`prop_abs_list`]*

*Note: this variable is excluded in the final decision trees, see Examining the Classification Variables: Section 5'

2.2 Model quality metrics

These metrics measure the quality of the model produced and are what we will use to develop the rules of thumb:

- The precision of the trend where `Precision <- 1/(sd(long_term_gr)^2)` and `long_term_gr` is a vector of annual growth rates where the `length()` of `long_term_gr` is equal to the number of model iterations [`precision_growth_rate`]
- Average precision of the year estimates where precision is defined in the same way as above [`mean_year_precision`]
- Proportion of years with converged Rhat [`PropYrConverged`]

3 Data

The data are in a few different locations and in most cases we need to go to more than one data source to get both sets of metrics. There is also a difference it the location of outputs, some are on Charlie's Cirrus folder, some her JASMIN folder and some are on LOTUS still.

3.1 Read in CIRRUS outputs

First, the CIRRUS outputs were loaded up, processed and saved. These were the model output data, including information on years which converged and numbers of records.

```
model_data <- read.csv('Results/metrics/model_data.csv')
head(model_data)
```

```
##           speciesName FirstYrConverged LastYrConverged PropYrConverged
## 1   FORMICA aquilonia            FALSE           FALSE               0
## 2 FORMICA cunicularia             TRUE            TRUE               1
## 3     FORMICA exsecta             TRUE            TRUE               1
## 4       FORMICA fusca             TRUE            TRUE               1
## 5      FORMICA lemani             TRUE            TRUE               1
## 6    FORMICA lugubris             TRUE            TRUE               1
##    mean_means median_means mean_medians median_medians
## 1 0.028790722  0.027866278  0.023195764    0.022140221
## 2 0.178260764  0.190840130  0.176301763    0.189831898
## 3 0.005626754  0.005036095  0.004832814    0.004510045
## 4 0.420225682  0.414878580  0.418160565    0.414514145
## 5 0.125723942  0.111931324  0.124597629    0.111521115
## 6 0.026551388  0.023028410  0.026153027    0.022550226
```

3.2 Posteriors

We can also get data on the trend estimate from the 1000 posteriors that Charlie Outhwaite calculated. This is for all species groups regardless of whether they were run on Cirrus or JASMIN. We have to remove some years full of NAs. This is where the model is run to a year where there is no data. As a consequence some of the 'final 10 years' runs will actually be for a shorter period.

```
write.csv(master, file = 'Results/metrics/ALL_posteriorLM.csv',
          row.names = FALSE)
```

3.3 Raw data files

There are some additional stats we can get from the raw data file. Again, these are for all groups not just CIRRUS/JASMIN runs. Given the memory requirements needed to reformat the plant and moth data they have been removed them from this step and will be incorporated after producing the classification trees.

```
write.csv(master, file = 'Results/metrics/ALL_rawMetrics.csv',
          row.names = FALSE)
```

4 Classification of Models

4.1 Create metrics

We can do some simple decision tree statistics to develop rules of thumb once we have decided what constitutes a 'bad' model. Here we choose some metrics that make for a bad model and use the `rpart` package to create a decision tree.

First, combine all the data

```
RM <- read.csv('Results/metrics/ALL_rawMetrics.csv')
LM <- read.csv('Results/metrics/ALL_posteriorLM.csv')
MM <- read.csv('Results/metrics/model_data.csv')

# Remove repeat variables from model data.  The mean_means variable is the mean
#  occupancy for the model across all years
MM <- MM[,c('speciesName','FirstYrConverged',
            'LastYrConverged','PropYrConverged',
            'mean_means')]

# These merges drop species wit no model data
trendsData <- merge(x = LM, y = RM,
                    by.x = 'species',
                    by.y = 'species')
trendsData <- merge(x = MM, y = trendsData,
                    by.x= 'speciesName', by.y = 'species')
head(trendsData[c('speciesName','PropYrConverged','mean_means','avVisitPerYear',
                  'Spnvisits','SpvisitPerYear','Totalnvisits','median','P90',
                  'prop_repeats_grp','visits_median','visits_P90','prop_of_years',
                  'prop_abs_list','prop_abs','Taxa','Taxa_Root')])
```

```
##                         speciesName PropYrConverged mean_means
## 1              Abrothallus bertianus              NA 0.13435213
## 2    Abrothallus bertianus_last10yrs              NA 0.19201512
## 3           Abrothallus caerulescens              NA 0.18045817
## 4 Abrothallus caerulescens_last10yrs              NA 0.12971493
## 5              Abrothallus cetrariae              NA 0.09710464
## 6    Abrothallus cetrariae_last10yrs              NA 0.12577490
##   avVisitPerYear Spnvisits SpvisitPerYear Totalnvisits median P90
## 1       1.846154        24      0.6000000        15545      1   3
## 2       2.000000        14      1.5555556         5061      2   3
## 3       1.000000         3      0.7500000        15545      1   1
## 4       1.000000         3      0.7500000         8380      1   1
## 5       1.000000         4      0.1025641        15545      1   1
## 6       1.000000         2      0.6666667         5061      1   1
##   prop_repeats_grp visits_median visits_P90 prop_of_years prop_abs_list
## 1        0.3333333           1.0        4.0    0.28260870     0.8302944
## 2        0.4285714           1.0        3.7    0.63636364     0.8080048
## 3        0.0000000           1.0        1.0    0.06521739     0.8305237
## 4        0.0000000           1.0        1.0    0.15789474     0.8235645
## 5        0.2500000           1.0        3.1    0.08695652     0.8305128
## 6        0.5000000           2.5        3.7    0.18181818     0.8084602
##    prop_abs             Taxa Taxa_Root
## 1 0.9984561          Lichens   Lichens
## 2 0.9972337 last10yrsLichens   Lichens
## 3 0.9998070          Lichens   Lichens
## 4 0.9996420 last10yrsLichens   Lichens
## 5 0.9997427          Lichens   Lichens
## 6 0.9996048 last10yrsLichens   Lichens
```

```
nrow(trendsData)
```

```
## [1] 20070
```

```
# Assign a category to identify those models which are just from last 10 yrs
trendsData$last10yr <- substring(trendsData$Taxa,1,8) == "last10yr"
```

There are some species with no data in this dataset. We will remove these species.

```
sum(trendsData$P90==0)
```

```
## [1] 658
```

```
trendsData <- trendsData[trendsData$P90>0,]
```

4.2 Consultation on model outputs

Next, define which of the species have 'good' models and which are 'bad'.

This was done by consultation with 3 experts on the models, testing them on 100 models which were sampled to provide a spread of examples across the dataset, with a cluster of models focused around the range in which there was likely to be controversial.

Below are graphs showing the distribution of `precision_growth_rate` (Figure 1) and `mean_year_precision` (Figure 2).

```
# Lets have a look at the distribution
quantile(trendsData$precision_growth_rate)
```

```
##           0%          25%          50%          75%         100%
## 4.259870e-04 2.960039e-02 9.274431e-02 3.223825e-01 2.702034e+02
```

```
quantile(trendsData$mean_year_precision)
```

```
##           0%          25%          50%          75%         100%
## 4.907666e+00 2.901984e+01 2.456900e+02 1.704495e+03 1.378302e+07
```

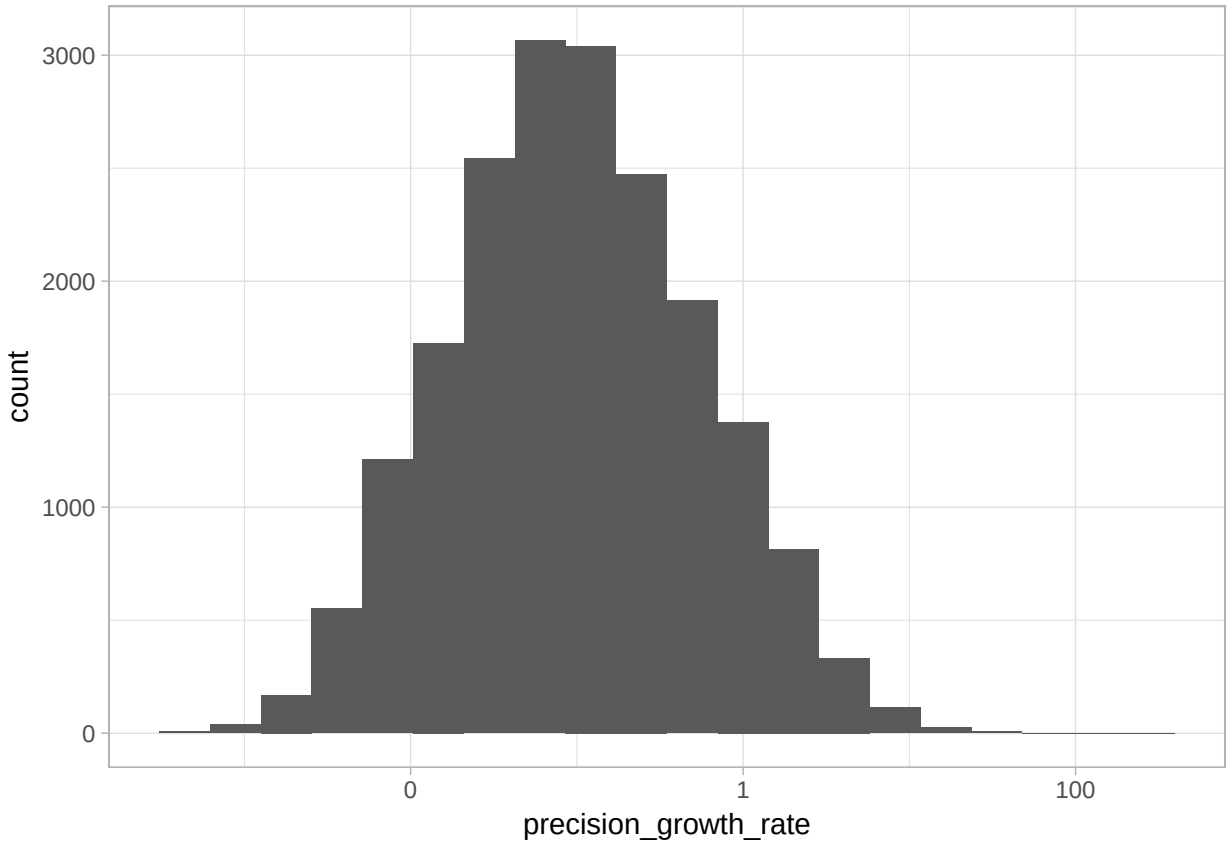

Figure 1: Histogram of the growth rate precision for all models

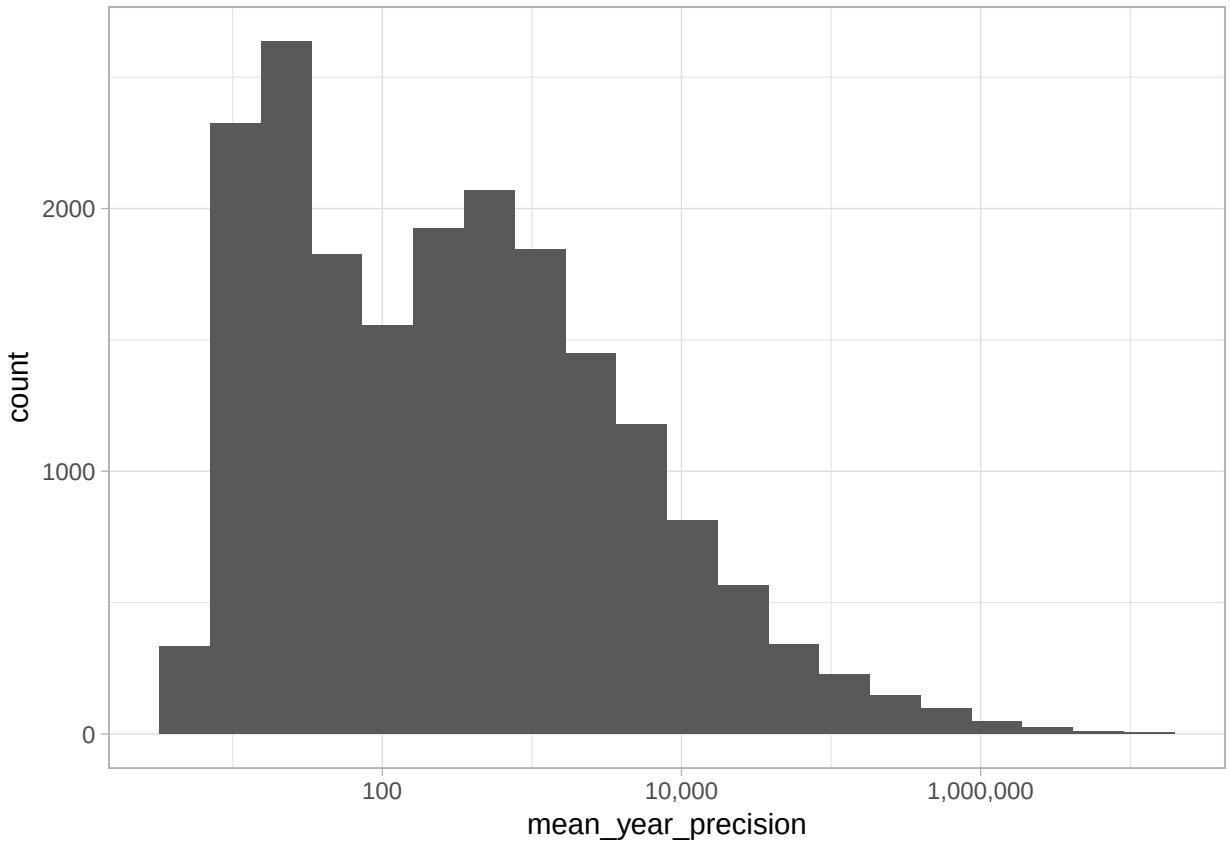

Figure 2: Histogram of the mean year precision for all models

Below are the results of the consultation. Figure 3 shows the score given to each model where the score is the number of experts who thought the model was of good quality. The models not used in the assessment are shown in the background in grey

```
# Lets look at the results of the consultation with model experts
fsdf <- read.csv(file = 'Results/Consultation/fsdf_full.csv')
```

```
## Warning: Removed 4005 rows containing missing values (geom_point).
```

The graphs (Figures 3 and 4) show the results of the consultation with experts on the model. There was unanimous agreement on 80% of the models about which were 'good' and 'bad'. The models chosen for this consultation were deliberately difficult to tell apart, so a 80% unanimous agreeement is considered very good.

Plotting these scores against proportion of years which converged, mean year precision and growth rate precision show that these parameters are correlated with quality, but it is not obvious how to perform this split.

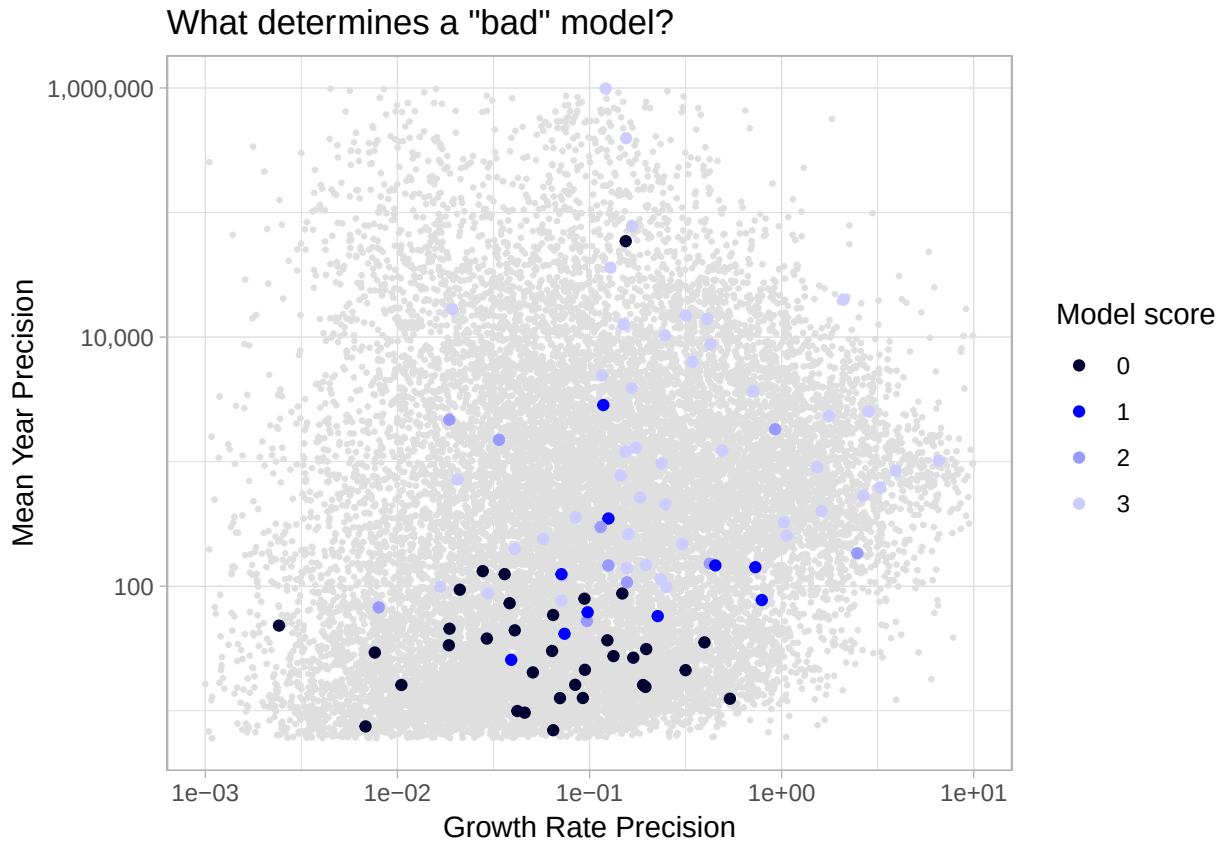

Figure 3: Graph showing results of consulation with model experts, plotting mean year precision vs growth rate precision

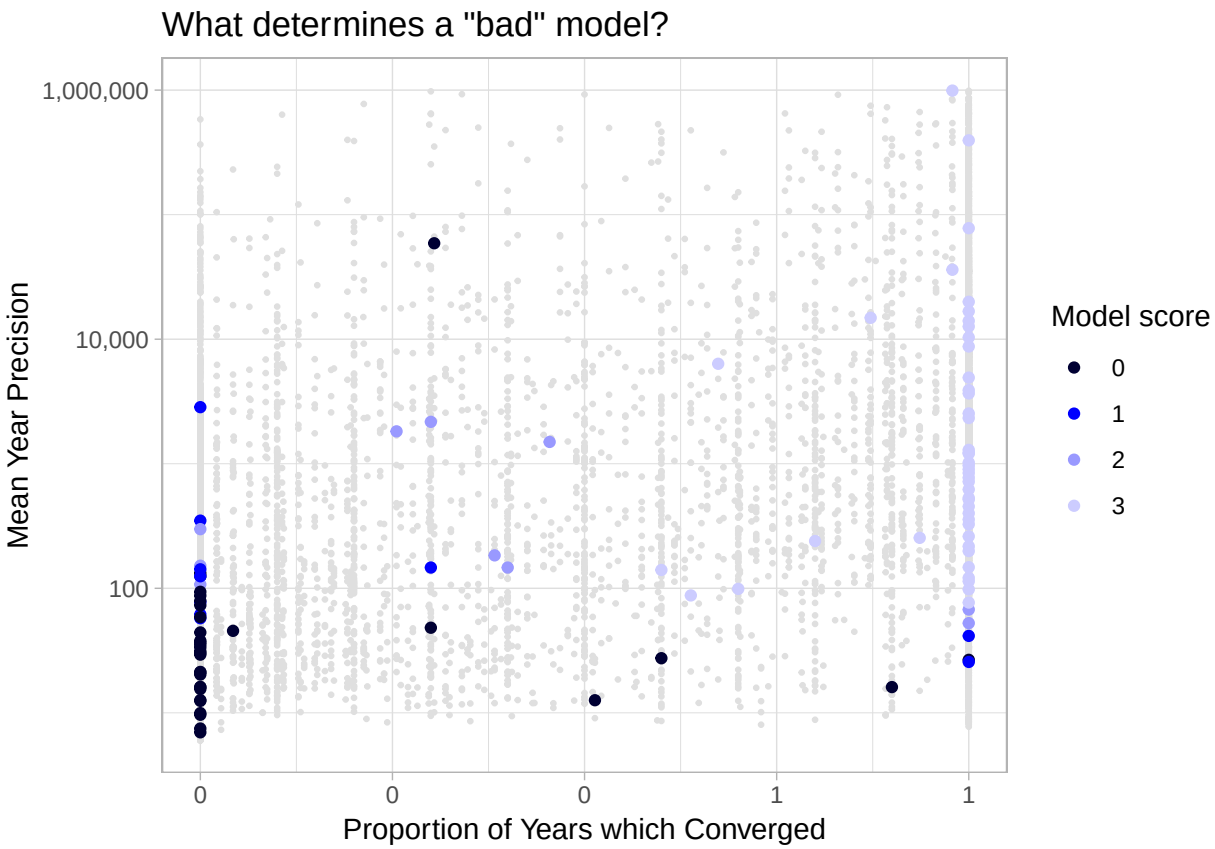

Figure 4: Graph showing results of consulation with model experts, plotting mean year precision vs the proportion of years which converged. Coloured points show the score assigned to models: light blue represents a score of 3 or unanimously a good model; black shows a score of 0, or unanimously a bad model. Light grey points in the background show all models which were not part of the consultation

4.3 Classification tree

To decide which models would be classified as good and bad, a decision tree was created to automatically classify the 100 models tested on. The models with a score of 2 or 3 were classified as good, while the models with a score of 0 or 1 were classified as bad.

Two decision trees are shown below, one using the proportion of years converged (Figure 6), and the other not (Figure 5), for comparison.

```
fsdf$good <- rep('bad',nrow(fsdf))
fsdf$good[fsdf$score>=2] <- 'good'
fsdf$good <- as.factor(fsdf$good)

fit_fsdf <- rpart(good ~ mean_year_precision + precision_growth_rate,
                  method = 'class',
                  data = fsdf)

rpart.plot(fit_fsdf, extra = 108, type = 3, clip.right.labs = FALSE)
```

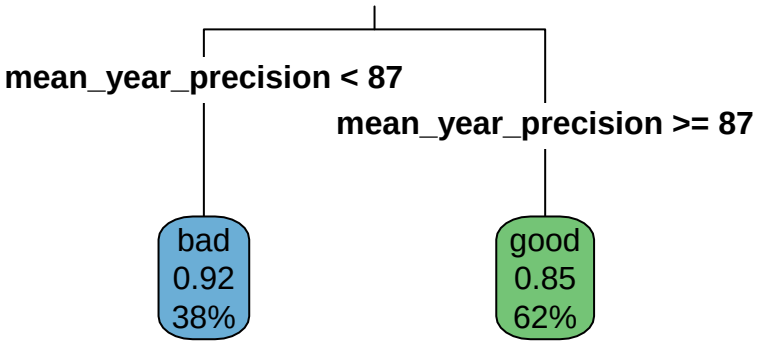

Figure 5: Decision tree for deciding which models are good or bad based on precision only

```
fit_fsdf <- rpart(good ~ mean_year_precision + precision_growth_rate + PropYrConverged,
                  method = 'class',
                  data = fsdf)
```

```
rpart.plot(fit_fsdf, extra = 108, type = 4, clip.right.labs = FALSE)
```

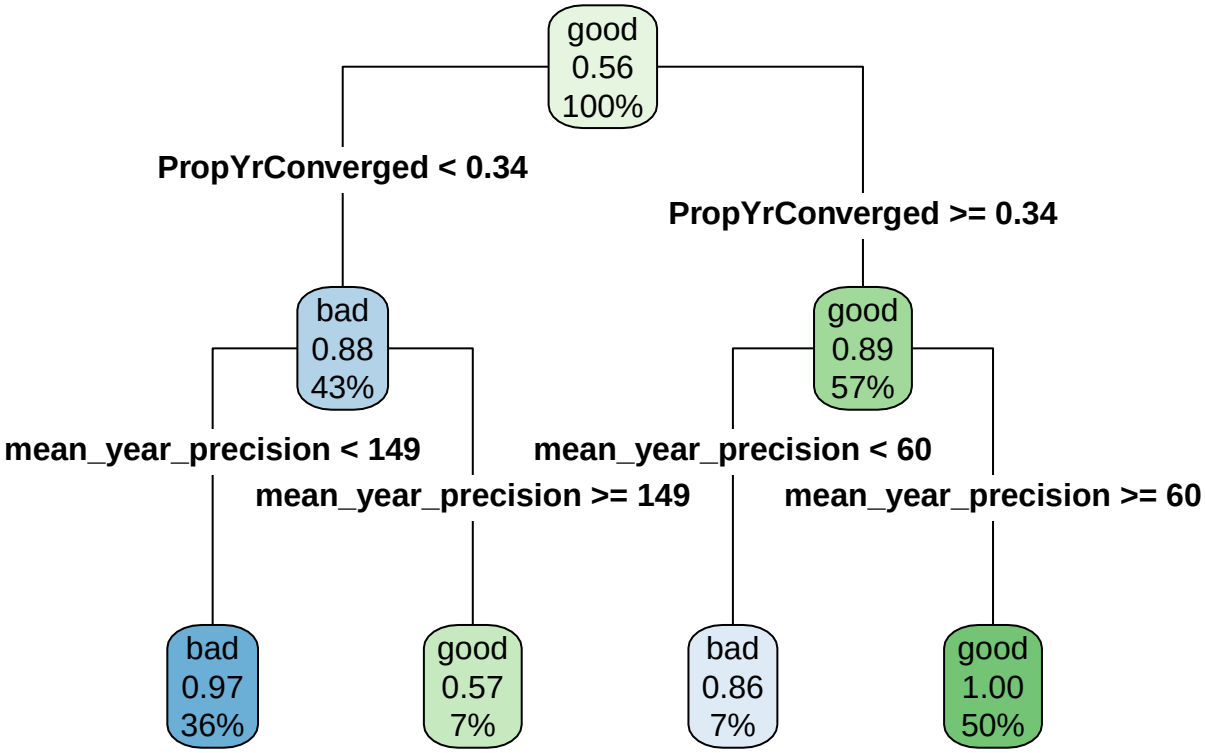

Figure 6: Decision tree for deciding which models are good or bad based on precision and convergence

```
## Warning: Removed 4005 rows containing missing values (geom_point).
```

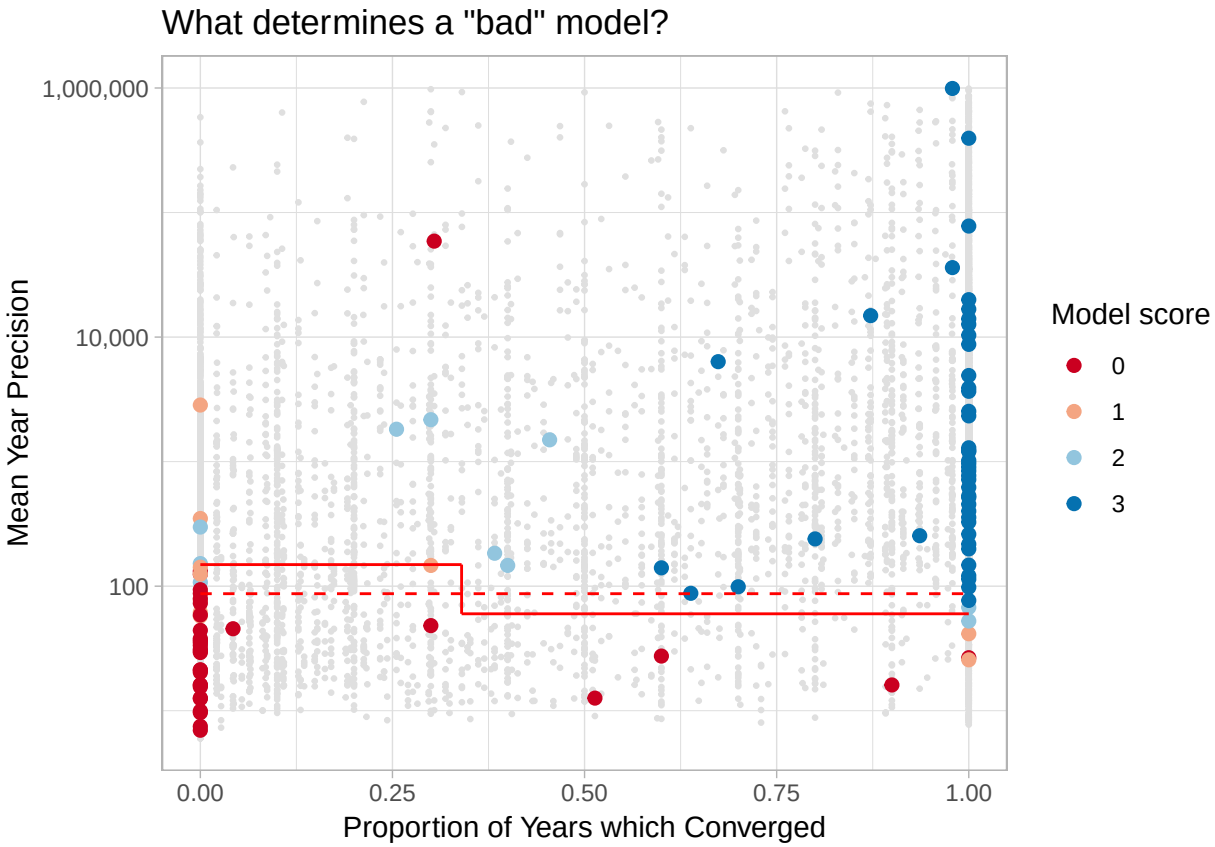

Figure 7: Graph showing results of consulation with model experts, including the results of the decision tree. Light grey points in the background show all models which were not part of the consultation

The success of the decision tree for classifying bad models is:

- Using precision only: 92% for bad models, 85% for good models (Figure 5)
- Using precision and convergences: 95% for bad models, 95% for good models (Figure 6)

Using both metrics is clearly better for splitting good and bad models, therefore this split will be used. The split is shown in Figure 7:

- If the proportion of years converged is < 0.3437, the model will be classified as 'bad' if the mean year precision is < 149.3
- If the proportion of years converged is $>= 0.3437$, the model will be classified as 'bad' if the mean year precision is < 60.2

5 Examining the classification variables

5.1 Taxonomic independence

Before running the classifiers, it is important to determine whether or not any of the variables are highly taxa dependent. If they are, they are much less generalisable across data sets.

This was done for all variables and the results from two of the variables are shown below:

- First is the violin plot for prop_repeats_grp: the proportion of site:year combinations for the species of interest which have more than one visit (Figure 8).
- Second is the violin plot for prop_abs_list: the proportion of data for the taxonomic group *not including the species of interest* which have a list length >1 (Figure 9).

The `prop_repeats_grp` data shows a broad range of results from 0 to 1 for all taxonomic groups. By contrast, the `prop_abs_list` data shows very clear division between taxonomic groups. Therefore, any partitioning on the basis of the `prop_abs_list` variable is taxonomically biased. For this reason, `prop_abs_list` will not be used for producing decision trees.

```
# Subset data to remove last 10 yr data, as it just makes the plot messier
td_taxa <- trendsData[as.character(trendsData$Taxa)==
                      as.character(trendsData$Taxa_Root),]
```

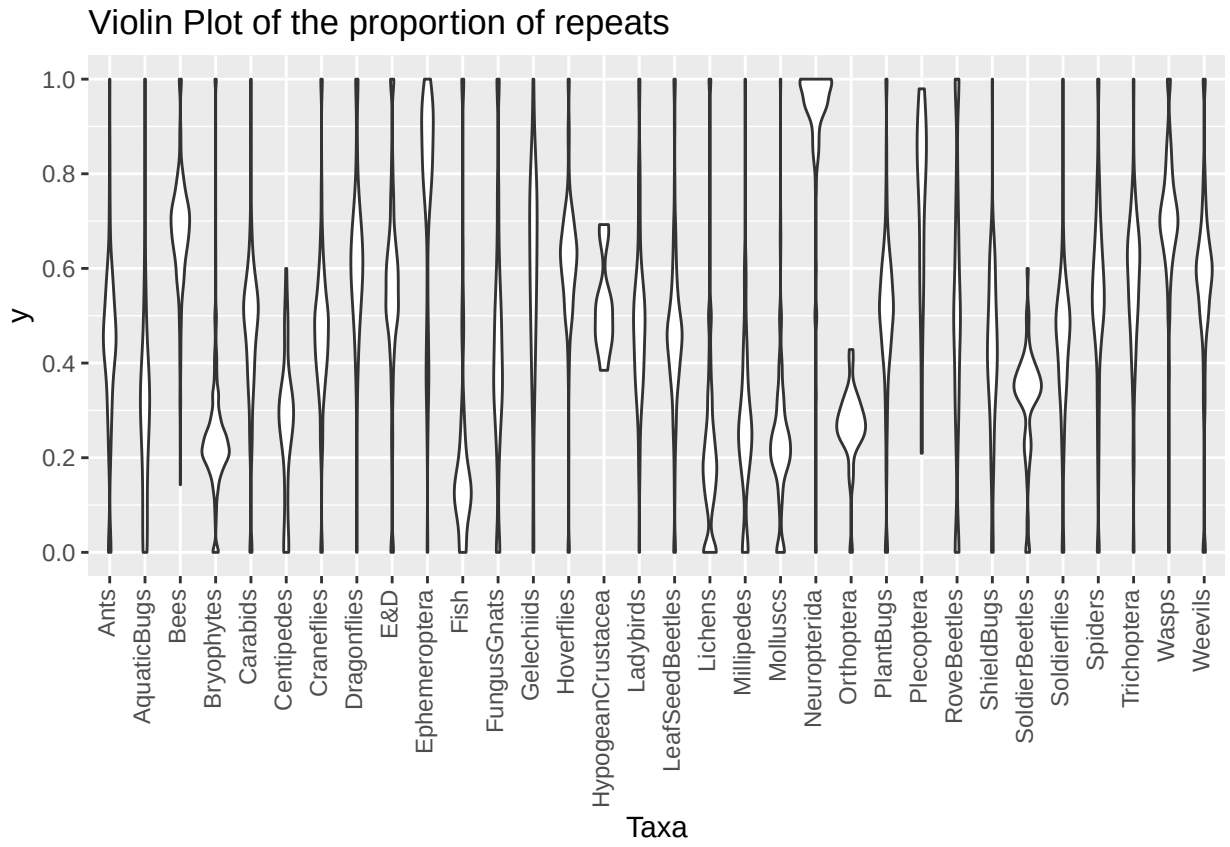

Figure 8: Violin plot for the prop_repeats_grp variable, defined as the proportion of all site:year combinations, where the focal species is observed, that have more than 1 visit for that taxonomic group

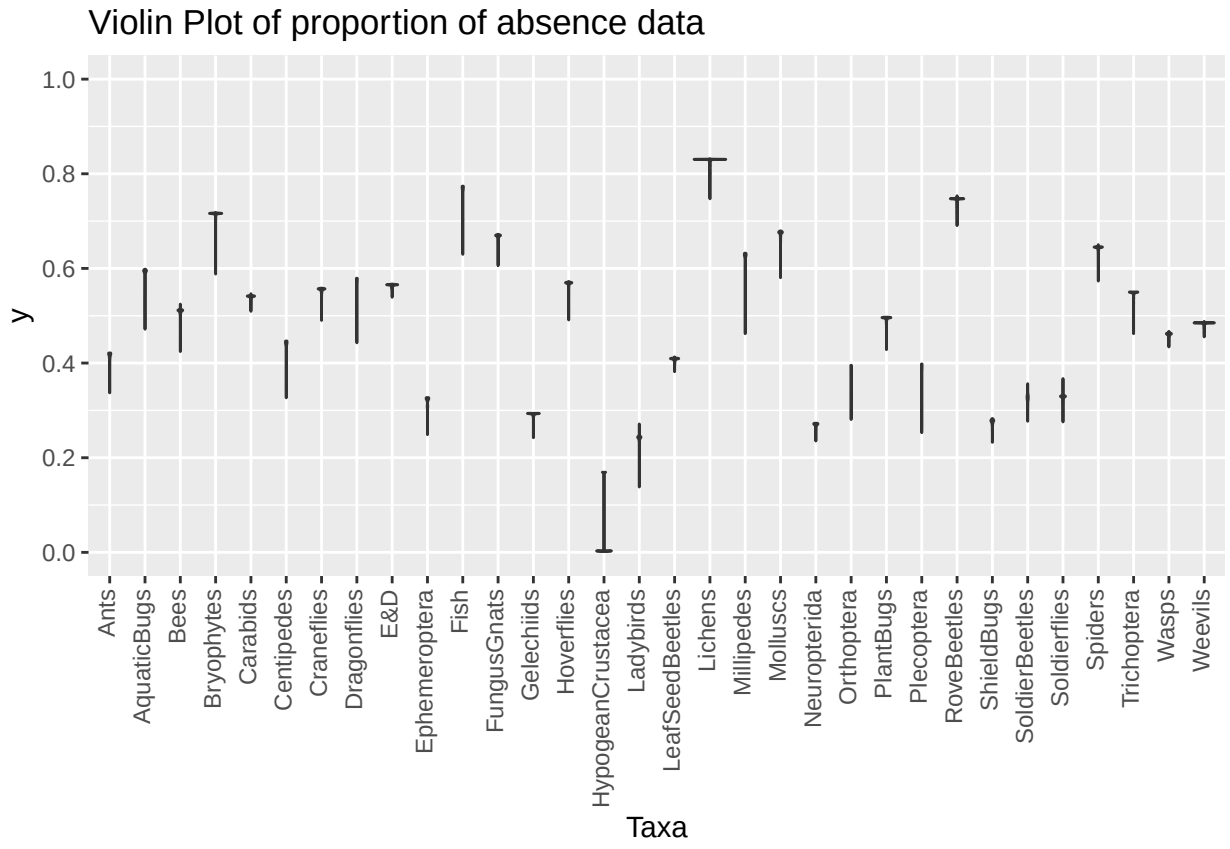

Figure 9: Violin plot for the prop_abs_list variable, defined as the proportion of records with list length greater than 1, for all records where the focal species was not observed

5.2 Correlation between variables

To examine the between all the variables, 2 pairs plots are produced below:

- Linear plots (Figure 10)
- Plots with `Spnvisits`, `P90` and `visits_P90` log transformed (Figure 11)

As can be seen, P90 is strongly correlated with number of records (`Spnvisits`), as expected. `Prop_abs` is negatively correlated with number of records, as the fewer the records for a species, the higher the proportion of taxonomic data without the species of interest.

Apart from number of records, which is not included in the classification tree, the variables show little cross correlation, suggesting they are a good choice for building a decision tree.

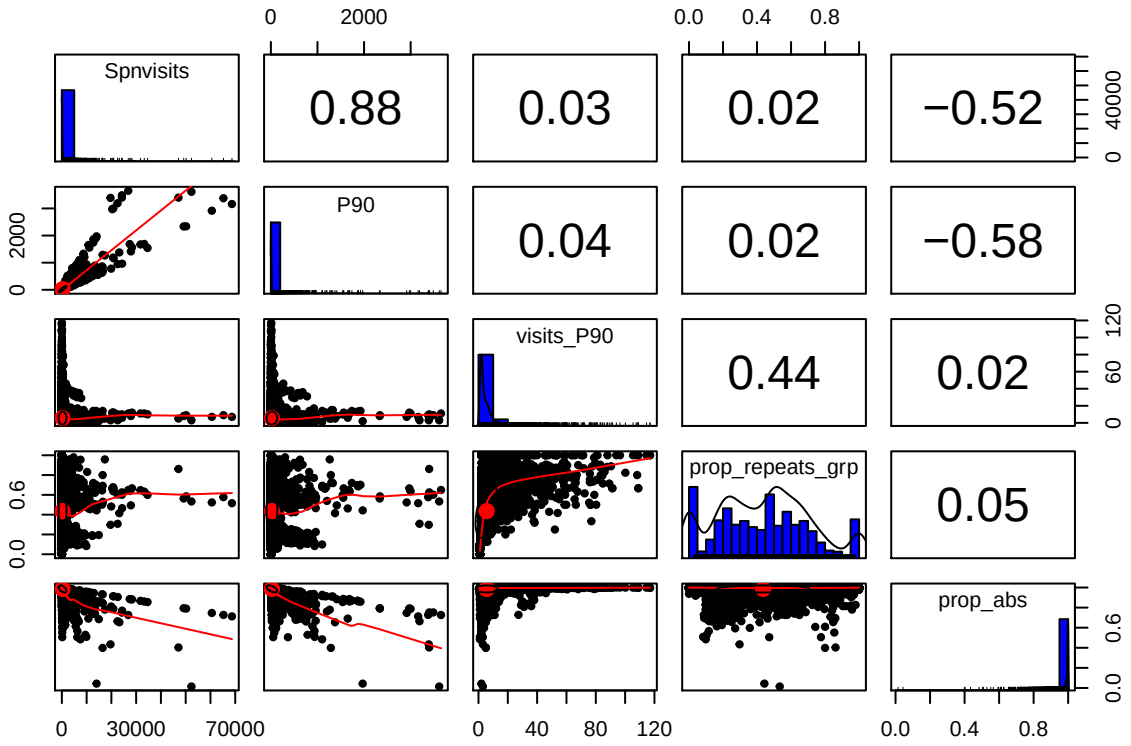

Figure 10: Pairs plot for all variables used to construct the classification tree, plus the Spnvisits variable, defined as the total number of records for each species of interest

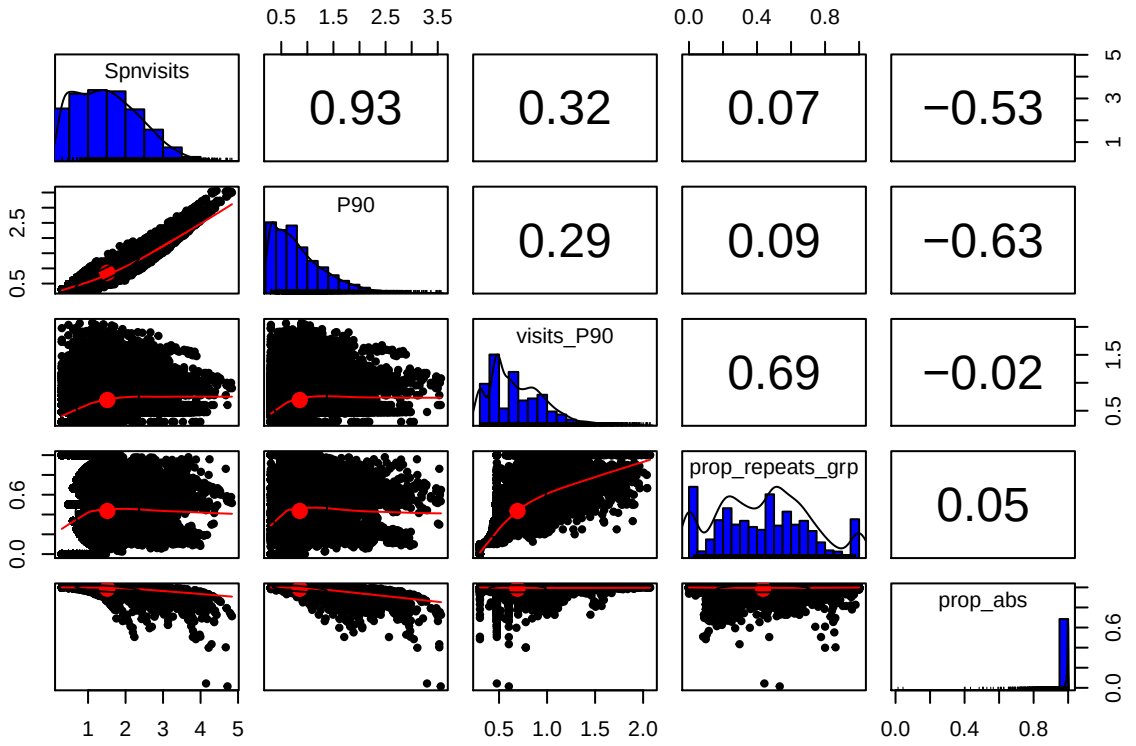

Figure 11: Pairs plot for all variables used to construct the classification tree, plus the Spnvisits variable, defined as the total number of records for each species of interest. Spnvisits, P90 and visits_P90 have been log transformed for this version.

5.3 Input data requirement

In order to be included in the dataset, one preliminary filtering step was carried out to remove some of the data. This step was to remove all records from visits to sites that never saw a repeat visit in any other years, for that taxonomic group. This step removed 33% of all records across our database.

- For instance, if a dragonfly was recorded in one site in 1990, if there was one more record of any dragonfly at that site in any year except 1990, both records would be included.
- If there was another record for another dragonfly to the same site in 1990, but no other records for that site, both of these records would be removed from the database.

This step was carried out as it was carried out on the data prior to running the models. Failing to do this step would make the results invalid. This means that any decision tree below has the pre-requisite that all records in the database must meet this requirement. To show the effect of this step, a plot showing the proportion of records removed versus the number of records in each group is shown in Figure 12. A graph showing the proportion of records for each taxonomic group is shown in Figure 13.

Tables are below to show all taxonomic groups with more than 400,000 records removed (Table 1), and all groups with more than 50% of records removed (Table 2).

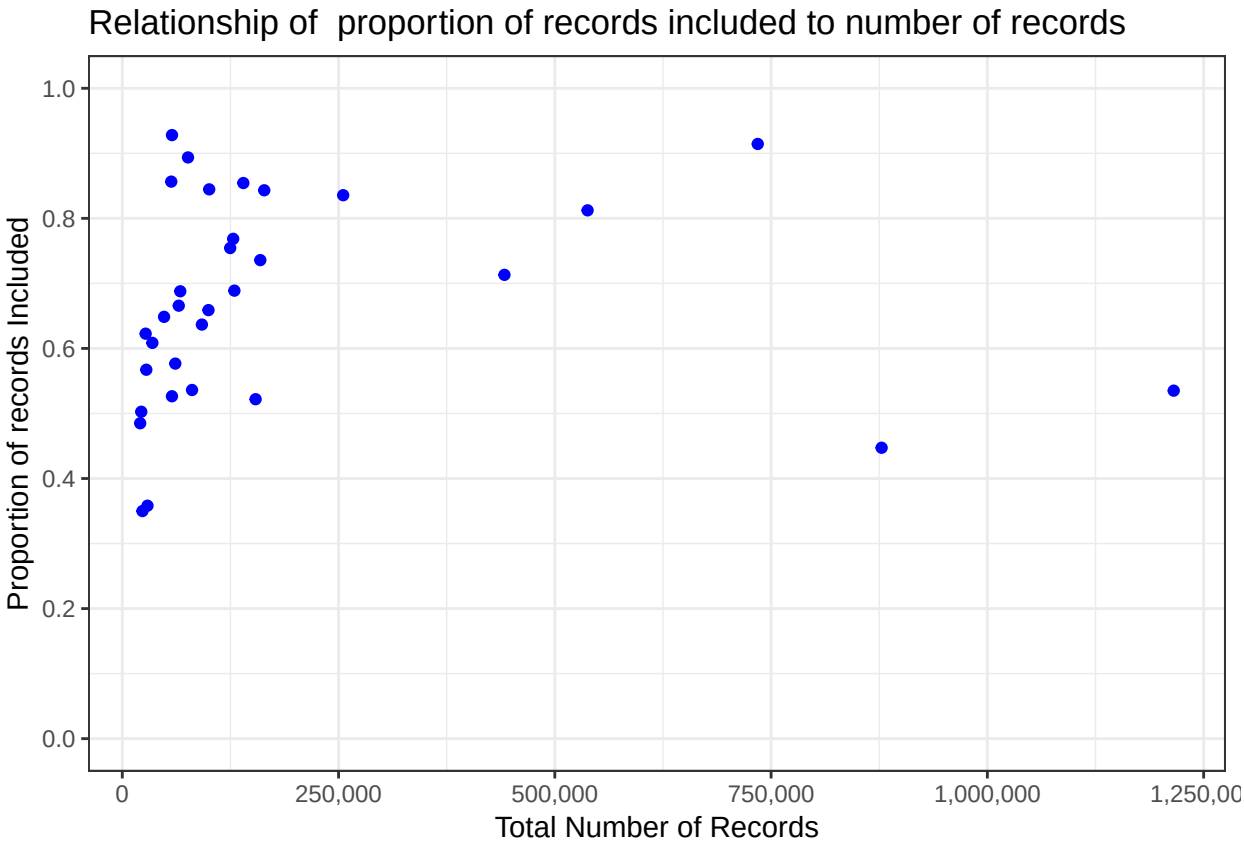

Figure 12: Plot showing proportion of records included vs total number of records. As can be seen, most taxonomic group retain at least half of their records. Those which retain fewer than 50% of their recdords are shown below

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

Overall, we found that 0.33 of all records were removed, varying from 0.07 to 0.65 across taxonomic groups.

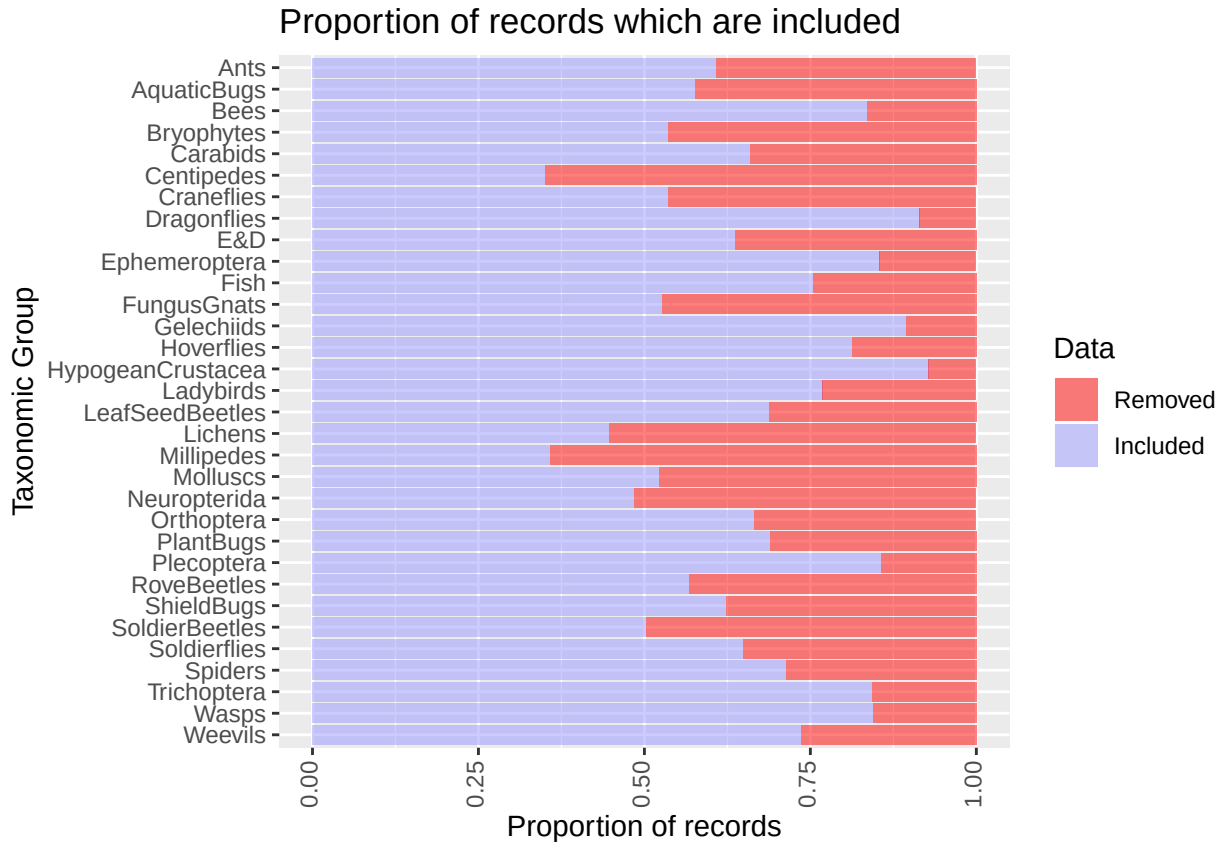

Figure 13: Graph showing the proportion of records which are included in the models. Those which are not are excluded or to sites which only received visit(s) in one year within the taxonomic group

Table 1: List of taxa with more than 400,000 records removed

taxa	RecordsIncluded	RecordsExcluded	TotalRecords	ProportionIncluded
Bryophytes	650324	565077	1215401	0.5350695
Lichens	392598	485143	877741	0.4472823

Table 2: List of taxa with more than half of records removed

taxa	RecordsIncluded	RecordsExcluded	TotalRecords	ProportionIncluded
Centipedes	8159	15156	23315	0.3499464
Lichens	392598	485143	877741	0.4472823
Millipedes	10429	18689	29118	0.3581633
Neuropterida	10045	10668	20713	0.4849611

5.4 Splitting the output data into good and bad models

Using the decision tree above for splitting output models into good and bad, the model outputs can be classified as good or bad. For some of the species we do not have the proportion of years converged. For these we use the simple decision tree: mean_year_precision >= 87.

The data below is also split by those with precision > 2126 and those below. This is a high level precision cut-off from a separate consultation, and represents a gold standard model.

```
trendsData$model_good <- trendsData$model_good_hi <- rep('bad',nrow(trendsData))
trendsData$model_good[(trendsData$PropYrConverged < 0.3437 &
                  trendsData$mean_year_precision >= 149.3) |
                  (trendsData$PropYrConverged >= 0.3437 &
                   trendsData$mean_year_precision >= 60.02) |
                  (is.na(trendsData$PropYrConverged) &
                   trendsData$mean_year_precision >= 87)] <- 'good'
trendsData$model_good <- factor(trendsData$model_good, levels = c('good','bad'))

trendsData$model_good_hi[trendsData$mean_year_precision > 2126] <- 'good'
trendsData$model_good_hi <- factor(trendsData$model_good_hi, levels = c('good','bad'))

trendsData$model_good_combined <- ifelse(trendsData$model_good_hi=='good','good_hi',
                                  ifelse(trendsData$model_good=='good','good','bad'))

trendsData$model_good_combined <- factor(trendsData$model_good_combined,
                                         levels = c('good_hi','good','bad'))
```

5.5 Effect of rarity of species on precision

For species which have a very low occupancy, the absolute precision on the occupancy estimate can be very high even if there is very little data. Conversely, a common or high occupancy species is is likely to have a low precision estimate even with a decent number of records. This is due to the fact that uncertainty on a high occupancy number equates to a greater absolute range in occupancy estimate when compared to the same relative uncertainty on a lower occupancy number.

This effect is demonstrated in Figure 14, showing mean occupancy vs precision.

After discussion, it was decided to drop all species with an occupancy of <0.05 from the data before running the rules of thumb classification tree. By doing so, roughly 20% of all species are dropped. These species are all rare species, and generally easy to model with relatively few records owing to their rarity i.e. even if very

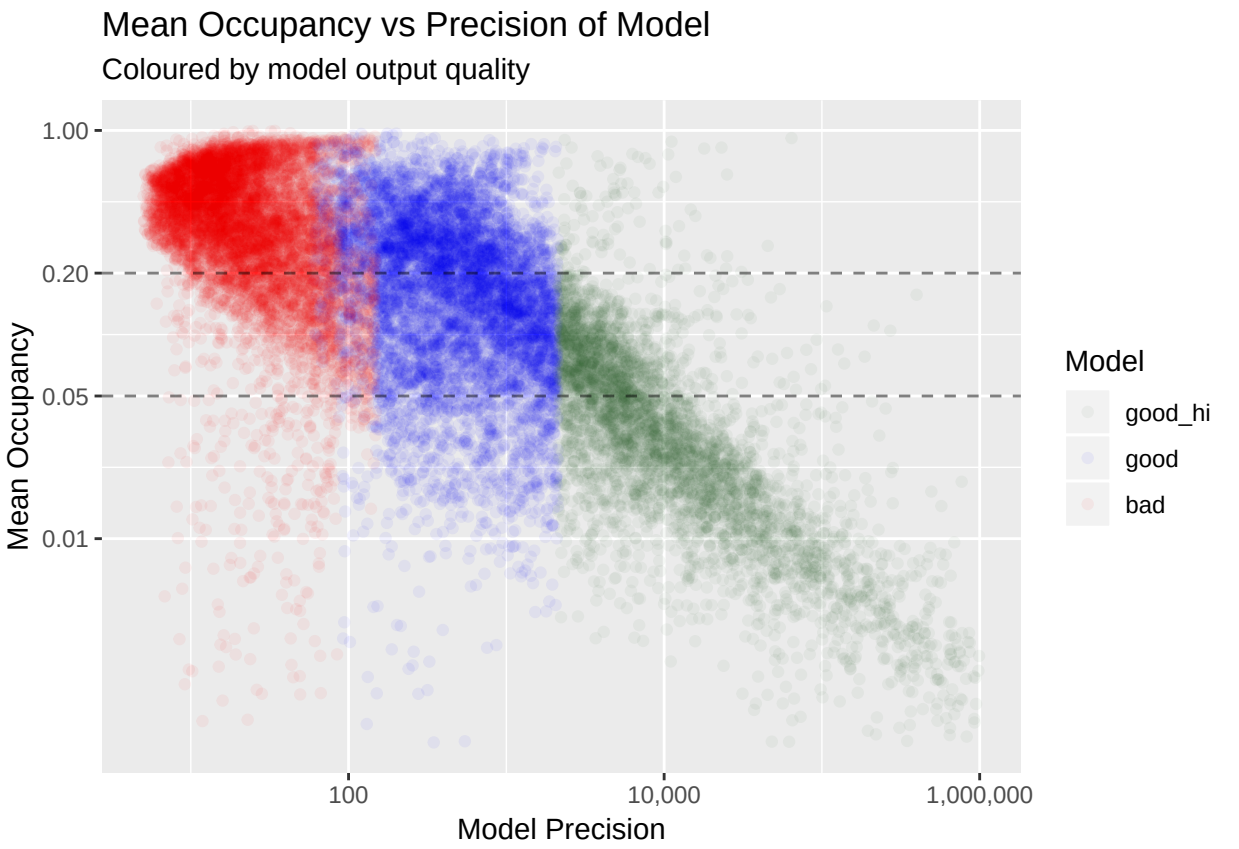

Figure 14: Plot of precision of trend vs mean occupancy for all species in the database. As can be seen, there is a clear negative relationship, indicating that it is easier to obtain a higher precision when a species is rare. There is a cloud of data points also at <100 precision and occupancy of .2 to .8. This cloud is interpreted as all species for which there is almost no data, so the model just has a guess

little data the model can be confident that the species is rare. By dropping these rare species, the resulting rules of thumb will be more applicable to common species.

```
trendsData <- trendsData %>% filter(mean_means > 0.05)
```

6 Classifying the data

6.1 Equally weighted decision tree

Using the split in the models between 'good' and 'bad', you can create a classification tree that attempts to use the variables we have extracted to partition the data to the best of its ability using a decision tree.

For this decision tree the evtree function is being used. This function creates a globally optimised decision tree, rather than a greedy decision tree. This results in more accuracy though the processing time is longer.

In this case, the badweight is set to 4, as we are using the high precision cutoff. This is due to the fact that there are ~4 times as many bad models as good ones using this cutoff. An equally weighted classifier would inevitably tend to be more specific and less sensitive given this imbalance. The weighting is therefore added to counter this imbalance.

For these models, testing was done to compare variables. In the end, the accuracy on the model was roughly the same if only P90 or prop_abs were used, or if all metrics were used. Therefore, just P90 and prop_abs were used to allow easy comparison of decision trees.

```
# Perform the fit
goodweight <- 1
badweight  <- 4
set.seed(1)
ev <- evtree(model_good_hi ~ P90 + prop_abs,
             data = trendsData,
             weights = ifelse(trendsData$model_good_hi == 'bad',
                              goodweight, badweight),
             control = evtree.control(maxdepth = 2))
```

```
load(file = 'ev_trees_hi.Rdata')
ev <- ev_trees$ev
ev
```

```
##
## Model formula:
## model_good_hi ~ P90 + prop_abs
##
## Fitted party:
## [1] root
## |   [2] prop_abs < 0.96925
## |   |   [3] P90 < 453.8: bad (n = 2610, err = 27.7%)
## |   |   [4] P90 >= 453.8: good (n = 351, err = 14.5%)
## |   [5] prop_abs >= 0.96925
## |   |   [6] P90 < 9.6: bad (n = 10762, err = 9.6%)
## |   |   [7] P90 >= 9.6: good (n = 5923, err = 36.7%)
##
## Number of inner nodes:    3
## Number of terminal nodes: 4
```

```
plot(ev)
```

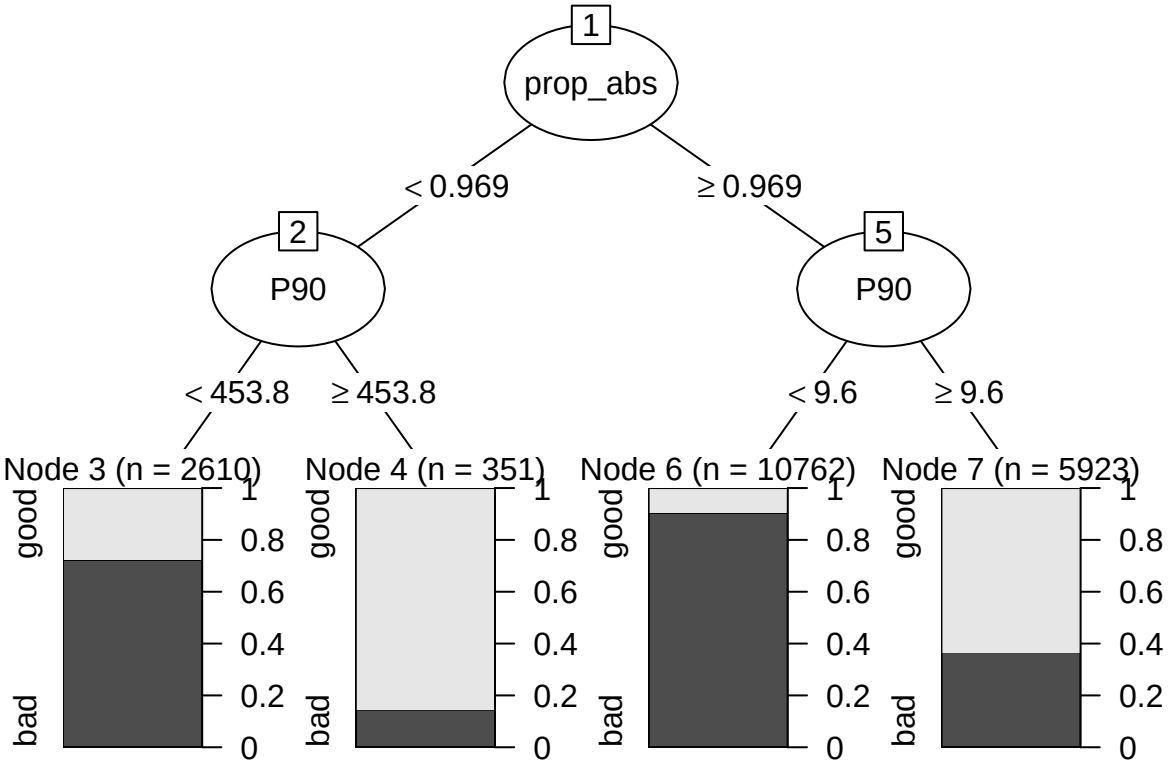

Figure 15: Decision tree for rules of thumb

The decision tree is shown in Figure 15. The first node in this decision tree is `prop_abs` >= 0.969. To meet this requirement, at least 96.9% of the data for the taxonomic group needs to come from visits where species other than the species of interest were observed. This is equivalent to 31 additional visits for every 1 visit in which species of interest was observed.

The second node is `P90`. This is the 90th percentile of number of records of the species of interest for years which have records.

To be classisified as worth running a model:

After removing all data from sites which never saw a repeat visit in subsequent year for any species within the taxonomic group (see *Input Data Requirement*):

If:

- `prop_abs` >= 0.969 and `P90` >= 7.6 OR
- `prop_abs` < 0.9697 and `P90` >= 453.8

The data will probably produce an acceptable model.

```
TP <- sum(predict(ev)==trendsData$model_good_hi & predict(ev)=='good')
FP <- sum(predict(ev)!=trendsData$model_good_hi & predict(ev)=='good')
TN <- sum(predict(ev)==trendsData$model_good_hi & predict(ev)=='bad')
FN <- sum(predict(ev)!=trendsData$model_good_hi & predict(ev)=='bad')
```

This decision tree has a positive precision (TP/(TP+FP)) of 31.3%, meaning 31.3% of the time, if the decision tree predicts the data will be able to produce a good model, it will.

This decision tree has a negative precision (TN/(TN+FN)) of 96.4%, meaning 96.4% of the time, if the decision tree predicts the data will *not* be able to produce a good model, it will indeed not produce a good model.

The recall/sensitivity/true positive rate (TP/(TP+FN)) is 69.7%, meaning 69.7% of all datasets which would produce a good model are classified as good datasets.

The specificity/true negative rate (TN/(TN+FP)) is 83.9%, meaning 83.9% of all datasets which would *not* produce a good model are classified as bad.

A graphic way of considering the decision tree is in Figure 16. This graph has the decision tree cutoffs shown in a black line, where any data below the line are predicted as not producing a good model and any above the line are predicted as having enough data. Points coloured blue did in fact produce a good model, while those below the line did not:

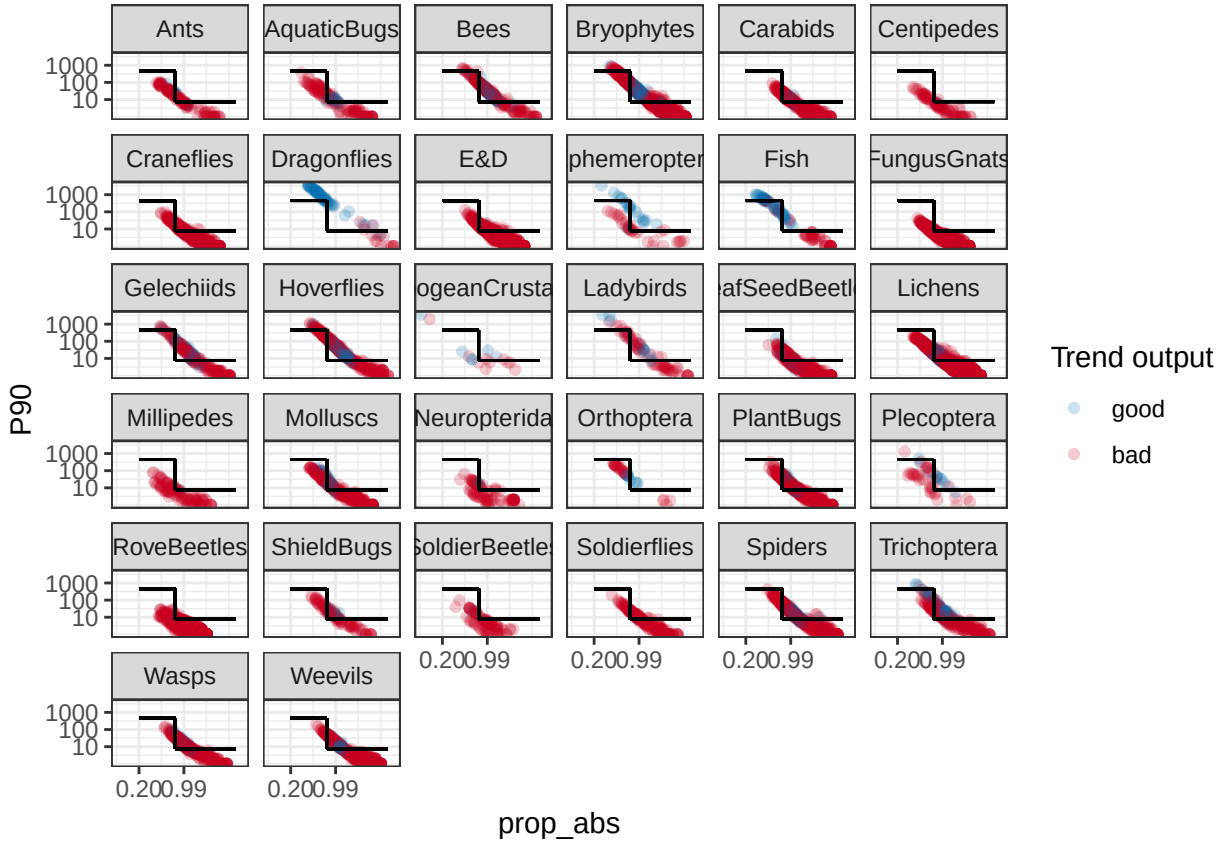

Figure 16: The results of the decision tree with P90 on a log-scale, and prop_abs on a logit scale, by taxonomic group

6.2 10:1 Good:bad tree - Aspirational Target

The decision tree is computed again twice more. In these models, the penalty for misclassifying good models as bad, or vice versa, is varied.

The first one below (Figure 17) is with a 10:1 good:bad ratio i.e. it is 10x more important to correctly classify good models as good. This weighting makes the decision tree much more specific, at the expense of sensitivity.

By using a 10:1 good:bad ratio, models which are classified as good are very likely to be good. As such, this decision tree can be considered an aspirational target.

Note that in the decision tree the values for n are not correct. This is due to the method evtree uses for weighting decisions: by replicating data of one output, in this case it replicates the 'bad' output 10x, such that assigning each of these outputs as bad becomes 10x more signigicant.

```
# produce a decision tree for goodweight = 2
goodweight <- 2
badweight  <- 1
set.seed(1)
ev_aspire <- evtree(model_good_hi ~ P90 + prop_abs,
                    data = trendsData,
                    weights = ifelse(trendsData$model_good_hi == 'bad',
                                     goodweight, badweight),
                    control = evtree.control(maxdepth = 2))
```

```
ev_aspire <- ev_trees$ev_aspire
ev_aspire
```

```
##
## Model formula:
## model_good_hi ~ P90 + prop_abs
##
## Fitted party:
## [1] root
## |   [2] P90 < 821.8: bad (n = 29075, err = 4.8%)
## |   [3] P90 >= 821.8: good (n = 60, err = 23.3%)
##
## Number of inner nodes:    1
## Number of terminal nodes: 2
```

```
plot(ev_aspire)
```

```
TP_aspire <-
  sum(predict(ev_aspire)==trendsData$model_good_hi & predict(ev_aspire)=='good')
FP_aspire <-
  sum(predict(ev_aspire)!=trendsData$model_good_hi & predict(ev_aspire)=='good')
TN_aspire <-
  sum(predict(ev_aspire)==trendsData$model_good_hi & predict(ev_aspire)=='bad')
FN_aspire <-
  sum(predict(ev_aspire)!=trendsData$model_good_hi & predict(ev_aspire)=='bad')
```

To be classified as worth running a model:

After removing all data from sites which never saw a repeat visit in subsequent year for any species within the taxonomic group (see *Input Data Requirement*):

If `P90` >= 821.8.

The data will probably produce an acceptable model.

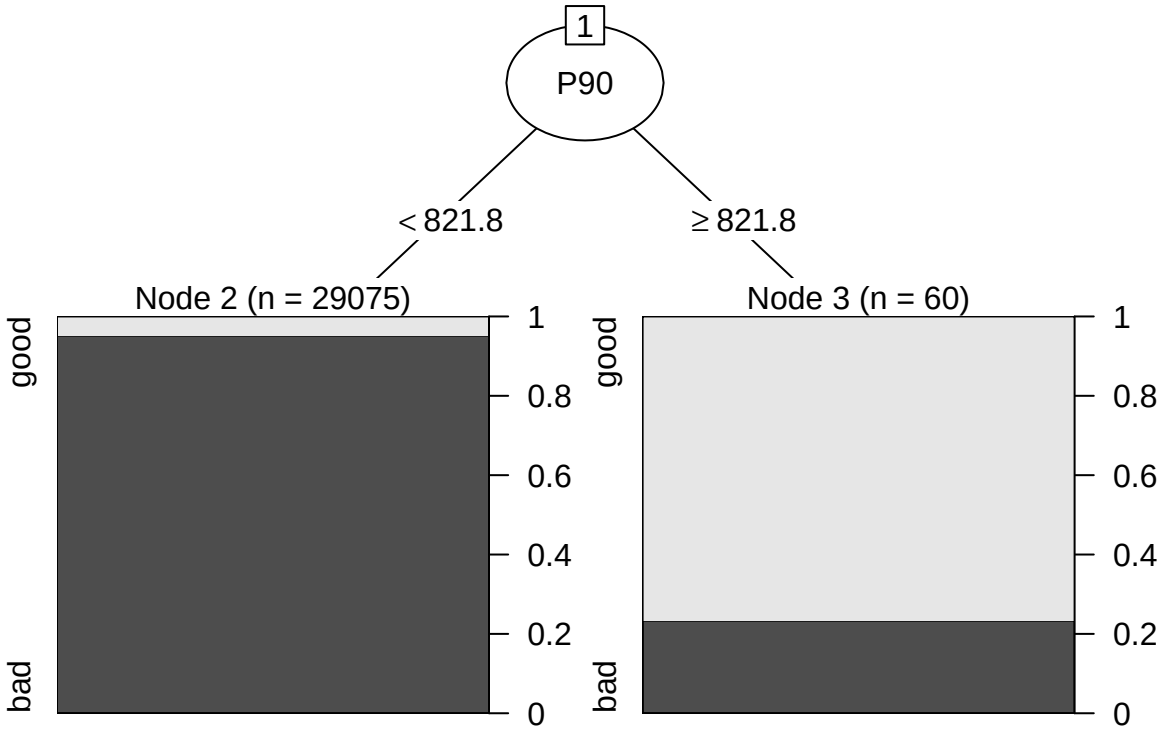

Figure 17: Decision tree for aspirational targets

The recall/sensitivity/true positive rate (TP/(TP+FN)) is 3.2%, meaning 3.2% of all datasets which would produce a good model are classified as good datasets (compare with 31.3% for the first decision tree).

The specificity/true negative rate (TN/(TN+FP)) is 99.9%, meaning 99.9% of all datasets which would NOT produce a good model are classified as bad (compare with 83.9% for the first decision tree).

6.3 1:10 Good:bad tree - bare minimum to try running a model

The decision tree below (Figure 18) is with a 1:10 good:bad ratio i.e. it is 10x more important to correctly classify bad models as bad This means that any models which are classified as bad are very likely to be bad, while a model classified as good may or may not be actually good. As such, this decision tree can be considered a bare minimum threshold.

```
# produce a decision tree for badweight = 40
goodweight <- 1
badweight  <- 40
set.seed(1)
ev_bm <- evtree(model_good_hi ~ P90 + prop_abs,
                data = trendsData,
                weights = ifelse(trendsData$model_good_hi == 'bad',
                                 goodweight, badweight),
                control = evtree.control(maxdepth = 2))
ev_trees <- list(ev,ev_aspire,ev_bm)
names(ev_trees) <- c('ev','ev_aspire','ev_bm')
save(x = ev_trees, file = 'ev_trees_hi.Rdata')
```

```
ev_bm <- ev_trees$ev_bm
ev_bm
```

```
##
## Model formula:
## model_good_hi ~ P90 + prop_abs
##
## Fitted party:
## [1] root
## |   [2] prop_abs < 0.9143
## |   |   [3] P90 < 304.9: bad (n = 640, err = 25.0%)
## |   |   [4] P90 >= 304.9: good (n = 2821, err = 3.6%)
## |   [5] prop_abs >= 0.9143
## |   |   [6] P90 < 5: bad (n = 10476, err = 21.4%)
## |   |   [7] P90 >= 5: good (n = 57945, err = 8.7%)
##
## Number of inner nodes:    3
## Number of terminal nodes: 4
```

```
plot(ev_bm)
```

```
TP_bm <- sum(predict(ev_bm)==trendsData$model_good_hi & predict(ev_bm)=='good')
FP_bm <- sum(predict(ev_bm)!=trendsData$model_good_hi & predict(ev_bm)=='good')
TN_bm <- sum(predict(ev_bm)==trendsData$model_good_hi & predict(ev_bm)=='bad')
FN_bm <- sum(predict(ev_bm)!=trendsData$model_good_hi & predict(ev_bm)=='bad')
```

The sensitivity is 95.9%, while the specificity is 63%.

As can be seen in the sensitivity and specificity scores above, this model creates a large number of false positives but a low number of false negatives. Therefore, if a dataset does not meet these requirements, it

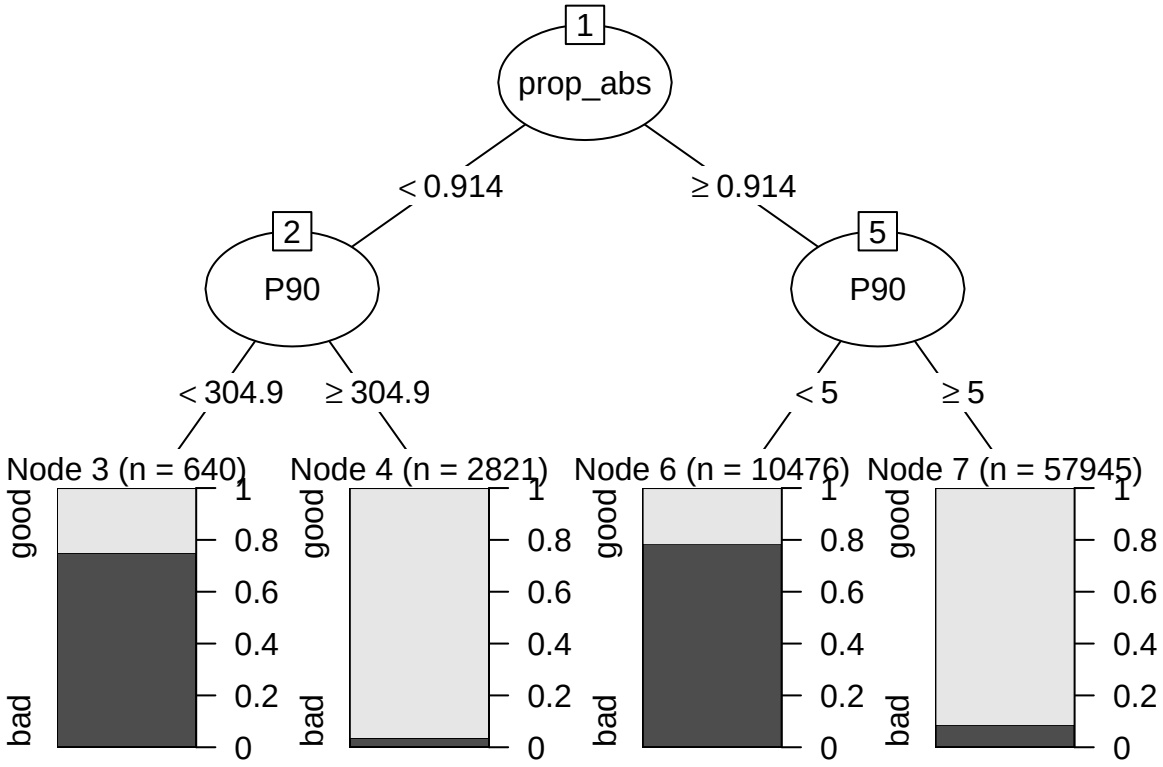

Figure 18: Decision tree for bare minimum targets

Table 3: Sensitivity and Specificity for the decision trees

Decision_Tree	Sensitivity	Specificity
Base	69.7	83.9
Aspirational	3.2	99.9
Bare Minimum	95.9	63.0

will probably not result in a good model.

A table of the sensitivity and specificity scores for these decision trees is shown in Table 3.

All these decision trees can be plotted on the same graph (Figure 19). This is shown below. Data points represent models, while the lines show the decision trees in graphical form. All data points below a given line are predicted to not produce a good model on the basis of that decision tree.

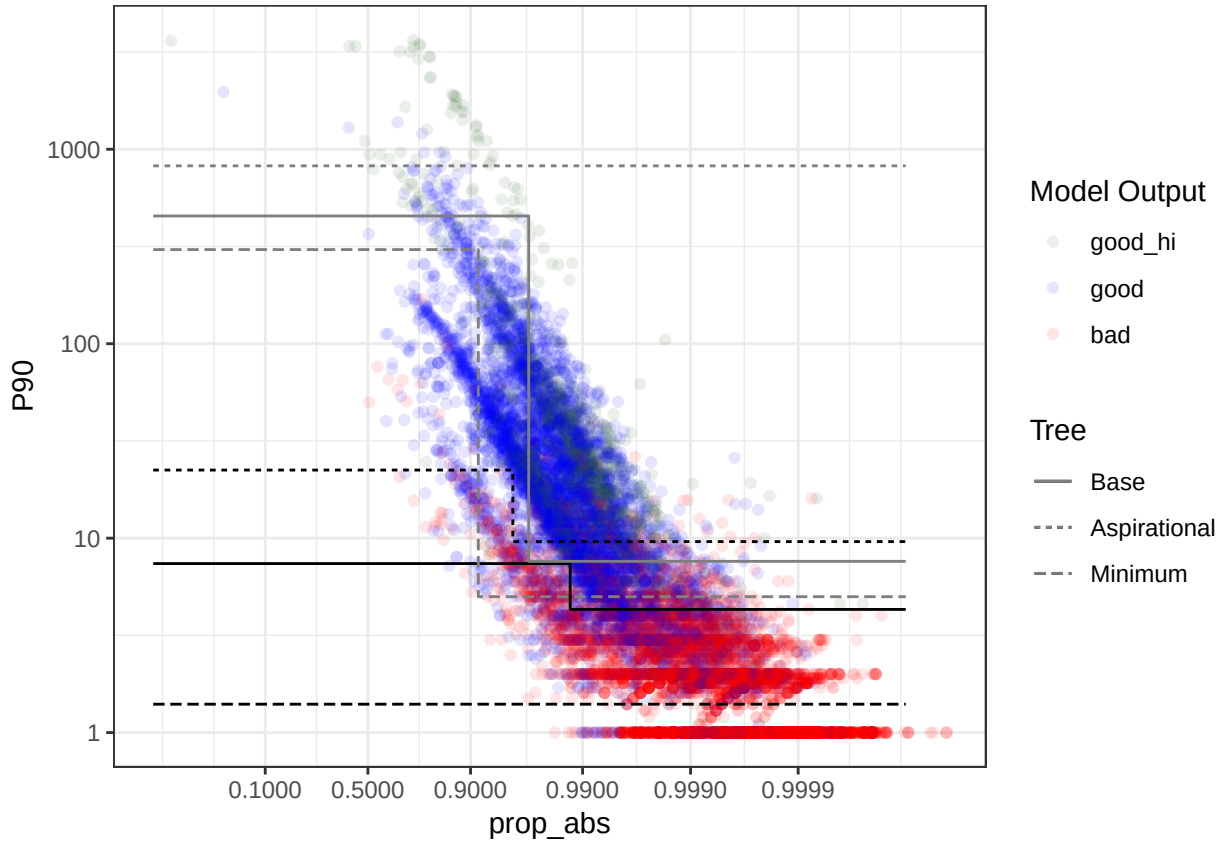

Figure 19: The results of all the decision trees with P90 on a log-scale, and prop_abs on a logit scale. Green are models which pass the high precision threshold, blue pass the consultation threshold, red pass neither. Grey lines are the high precision decision trees, black are the consultation threshold decision trees.

As can be seen, the aspirational tree line is the hardest to get above (it is highly specific), but accordingly there are a large number of blue and green (successful) models below the line.

Similarly, the bare minimum tree line is very easy to get above (it is highly sensitive), but there are a lot of red (unsuccessful) models above the line.

The Base model is a good balance between these two.

6.4 Applying the decision tree to all the data

In order to apply the decision trees to the data, including species and datasets for which we do not currently have a model, the function below creates a new parameter `data_good` which returns `bad` if the data are likely to produce a bad model, and `good` if the data are likely to produce a good model

```
calc_bad <- function(df){
  df$Node1 <- df$prop_abs>=.969
  df$Node2 <- df$P90>=7.6
  df$Node3 <- df$P90>=453.8
  df$data_good <- rep('bad',length(df$species))
  df$data_good[(df$Node1&df$Node2)|(df$Node3)] <- 'good'
  df$data_good <- factor(df$data_good,levels=c('good','bad'))

  if(!is.null(df$habitat)){
    df$habitat <- as.character(df$habitat)
  }
  if(!is.null(df$region)){
    df$region <- as.character(df$region)
  }
  if(!is.null(df$code)){
    df$code <- as.character(df$code)
  }
  return(df)
}

trendsData <- calc_bad(trendsData)
```

7 More investigation about the parameters in the decision tree

7.1 The relationship between P90 and P50

One of the possible concerns in applying this more widely is that while we found that P90 was an important parameter in the decision tree, it could also be strongly correlated with median number of visits per year. We note that we have good prior reasons for thinking that P90 is useful (because it allows good estimation of detection), but here we test the relationship between the two. They are plotted together in Figure 20.

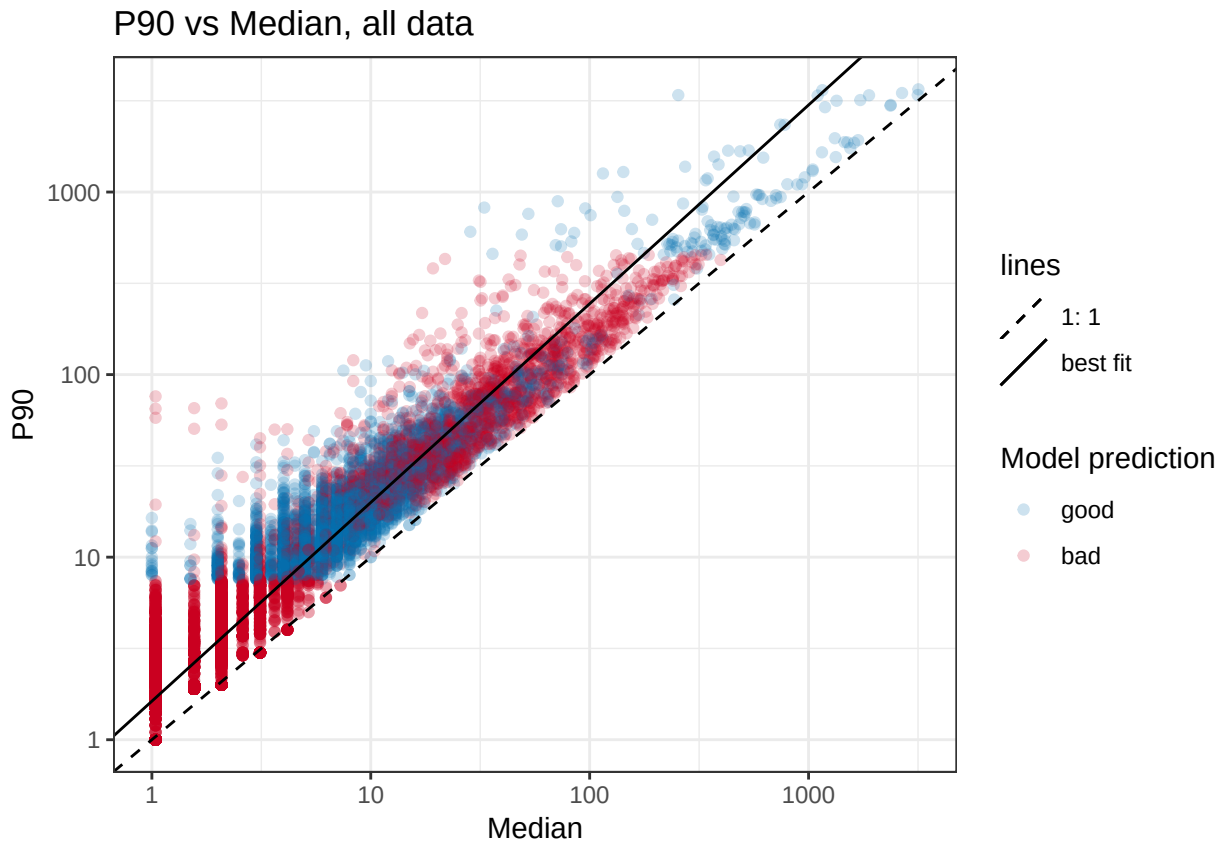

Figure 20: This graph shows that there is a linear trend of the P50 against the P90, so although the P90 represents the best years in the dataset (and this is indeed a better predictor of the adequacy of the data than P50) this is broadly related to the median years. Over the range of the bulk of these data (i.e. up to median = 100) the P90 is two to three times greater than the median

7.2 The relationship between precision and P90

We have used a decision tree to model when the data are minimal adequate. However, does more data automatically mean better trends?

One way to answer this is to model the lower limit of P90 against mean year precision. This can be done using quantile regression, and the resulting plot is shown in Figure 21.

```
trendsData.p90.7_6 <- subset(trendsData, P90>7.6)

mod.limit.of.precision <- rq(
      log10(mean_year_precision) ~
        log10(P90),
      tau = 0.05, data = trendsData.p90.7_6)

# predict the points at P90 = 453.8 and max P90
segment.ys <-
  10^(predict(mod.limit.of.precision,
              newdata = data.frame('P90' = c(7.6,
                                             max(trendsData.p90.7_6$P90)))))
```

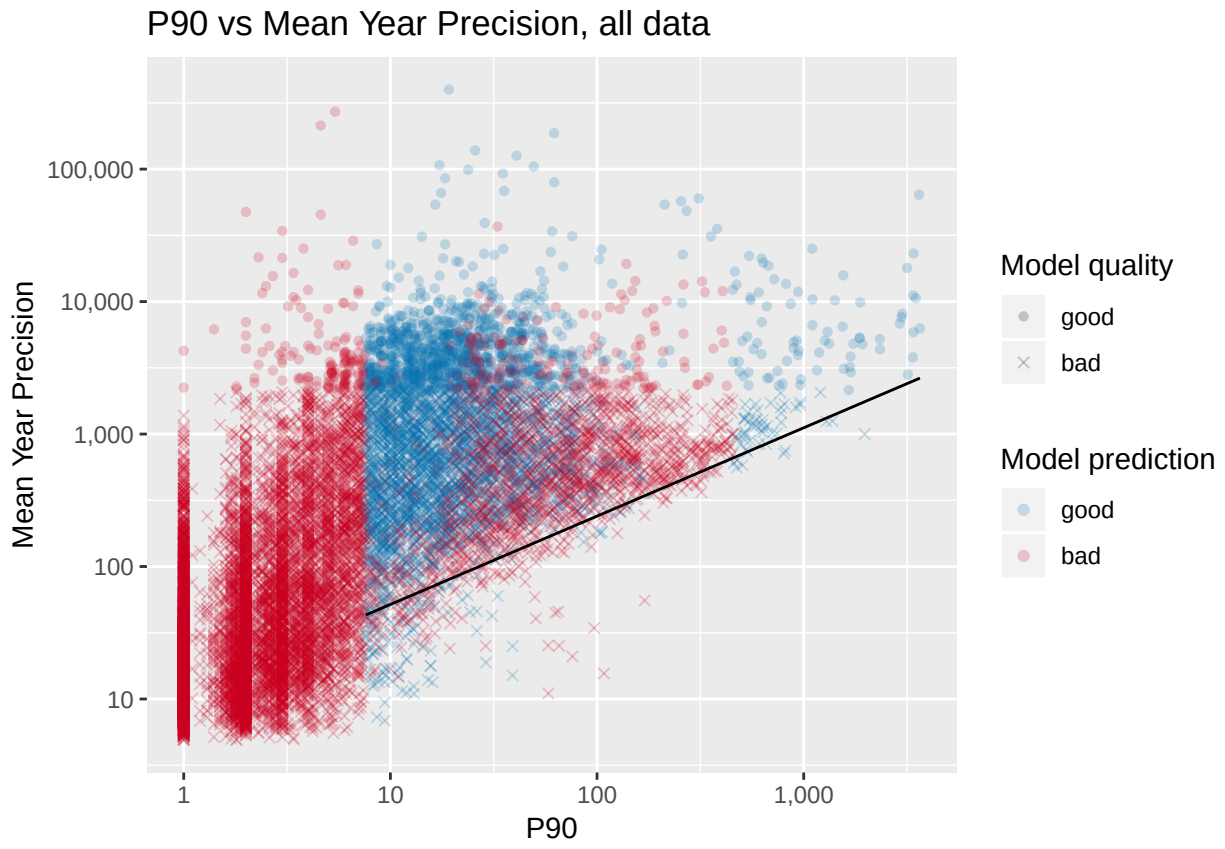

Figure 21: This shows that after a cut-off of P90 > 7.6 (the lower of the P90 cut-offs from the best guess decision tree), the more data there are (represented here by the increase in P90) the better the precision on the resulting occupancy model. The line represents the quantile regression through the 5th percentile. This shows that precision can be high with low P90, but the lowest possible precision is higher with higher P90, i.e. more data

8 How many species can we model?

8.1 For the UK

Based on the work above, we can estimate how many species we can model for taxa across the whole of the UK and across multiple habitat and regional areas.

8.1.1 Extracting butterfly and moth data

Before estimating how many species we can model, the data for the moth and butterfly records was extracted and added to the metric output data table. Just the metrics which came out from the decision trees were included in this extraction, so it ran much faster than the earlier data extraction.

```
write.csv(master, file = 'Results/metrics/ALL_combinedRawMetrics.csv',
          row.names = FALSE)
```

8.1.2 Plotting the data

In order to plot the taxa data, functions were written to plot the graphs when given an input data frame.

To provide a benchmark, below are three plots for all taxa for all UK data.

- Absolute number of species within each group which can be modelled (Figure 22)
- Proportion of species for each group which can be modelled (Figure 23)

```
num_spec(RM) %>% stack_taxa()
```

```
num_spec(RM,proportional=TRUE) %>% stack_taxa()
```

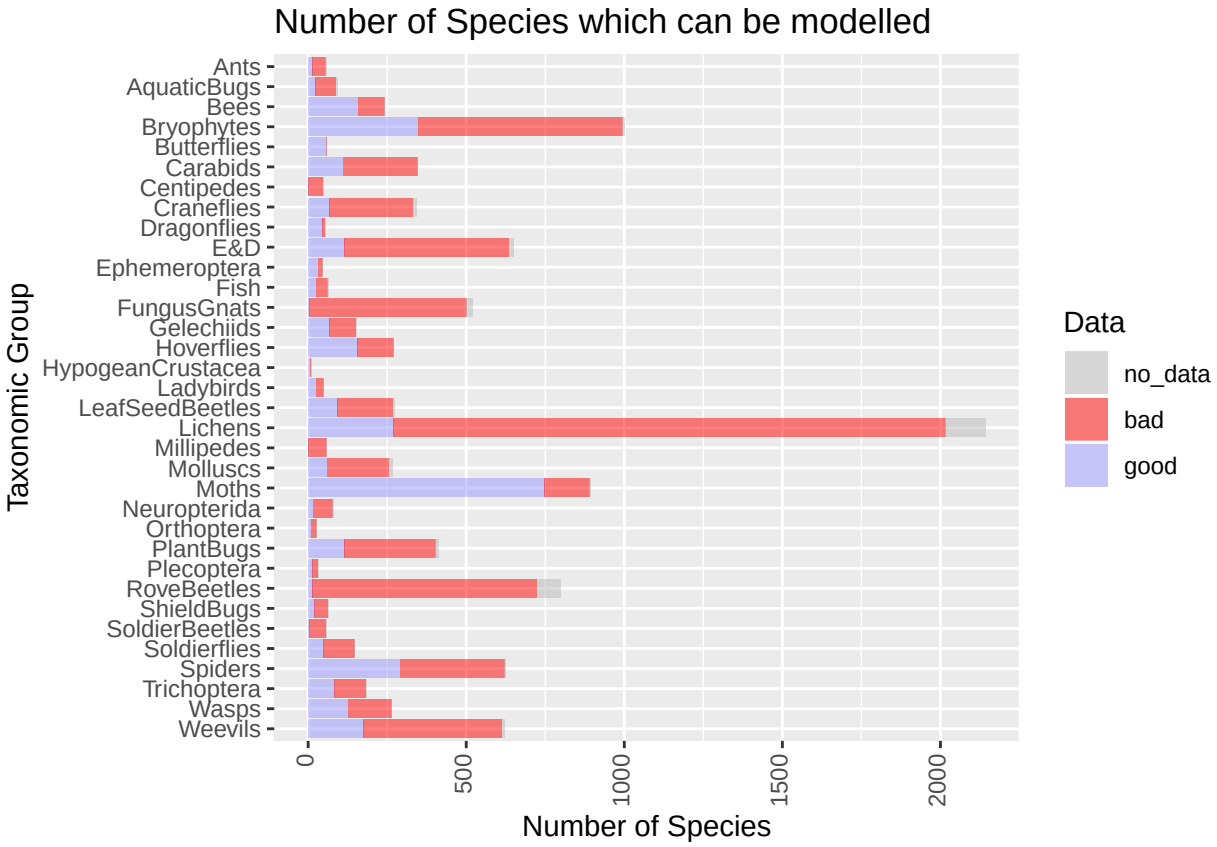

Figure 22: Plot of number of species which can be modelled, split by taxonomic group. Blue shows species which can be modelled, red those which cannot. Grey are those for which there is no data, due to there being no records which meet the requirement of coming from sites with repeat visits in any subsequent years.

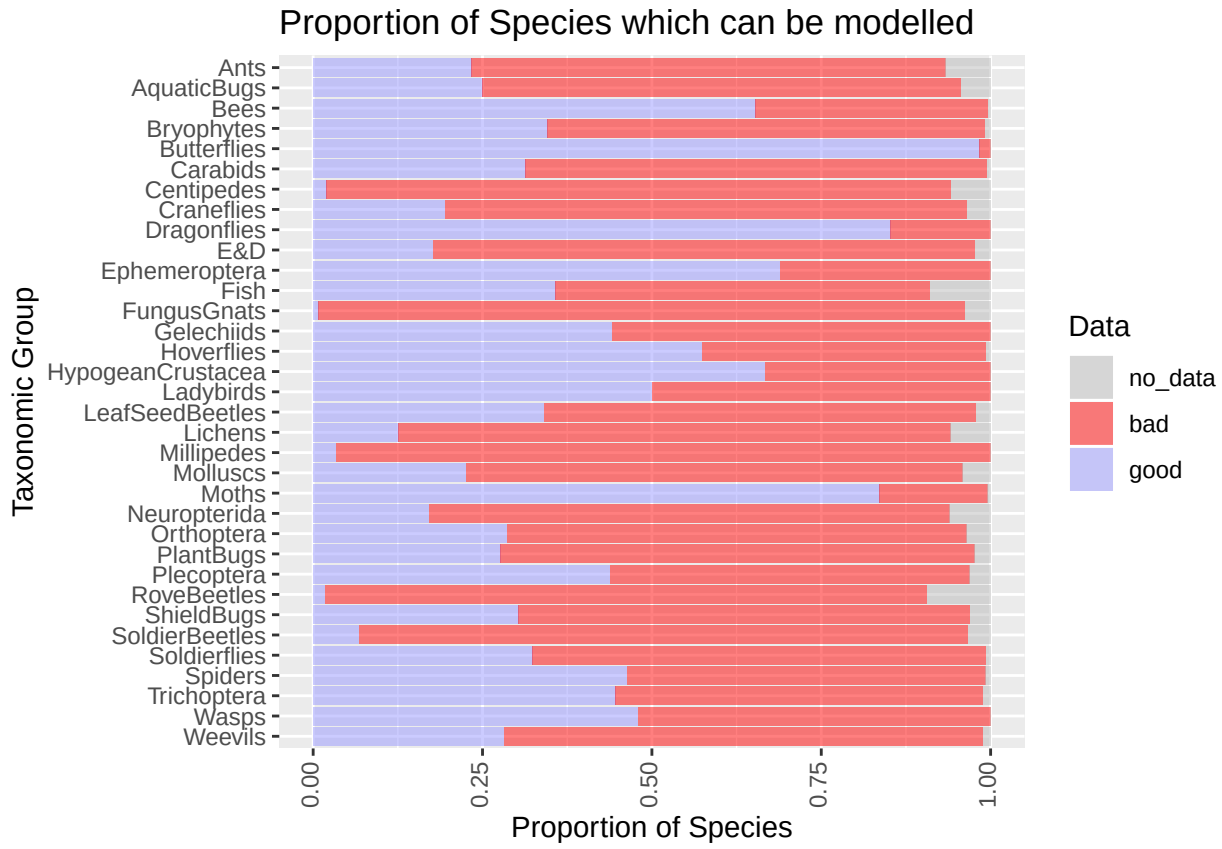

Figure 23: Plot of number of species which can be modelled, as a proportion of the total number of species for each taxonomic group.

Table 4: List of habitats

Habitat	Habitat_code
Broadleaf woodland	BLW
Coniferous woodland	CWL
Arable	A
Improved grassland	IG
Semi-natural grassland	SNG
Mountain, heath, bog	MHB
Coastal	C
Freshwater	FW
Built-up areas and gardens	BU

8.2 By habitat

The possible habitats are shown in Table 4.

To calculate the raw metrics for each habitat, the squares in the UK all need to be assigned to a habitat. To do so, a habitat datafile is read in and the range of percentage cover for each habitat is extracted. This produces a list of coverage for each habitat such as:

- Habitat: BLW; Coverage: 1%, 1%, 2%, 3% ... 95%, 98%, 99%

From this list, the median coverage for each habitat is calculated. Each square in the UK which has above this median coverage is assigned to that habitat. Therefore it is possible for a square to contain multiple habitats or not represent any habitat.

```
write.csv(x = td, file.path('./Habitat/HabMetrics/ALL_habMetrics.csv'))
```

The metrics for butterflies and moths also needed to be calculated, in a separate function. These are formatted differently so only the basic metrics were extracted, to save computational power.

```
write.csv(x = td, file.path('./Habitat/HabMetrics/ALL_combinedHabMetrics.csv'))
```

Using all these data, the proportion of species which can be modelled for each habitat can be assessed.

```
# Read in the habitat raw metrics
RM_hab <-
  read.csv(file = file.path('Habitat/HabMetrics/ALL_combinedHabMetrics.csv'))
```

```
# Calculate which datasets are likely to produce good or bad models
RM_hab <- calc_bad(RM_hab)
```

Below are two sample graphs for two habitats:

- Proportion of species which can be modelled using only Broad-Leaf Woodland data (Figure 24)
- Proportion of species which can be modelled using only Coastal data (Figure 25)

```
# Plot up a sample graph for broad leaf woodland
num_spec(RM_hab,habitat='BLW',proportional=TRUE,absences=TRUE) %>%
  stack_taxa(prefix='B-L Woodland')
```

```
# And for coastal
num_spec(RM_hab,habitat='C',proportional=TRUE,absences=TRUE) %>%
  stack_taxa(prefix='Coastal')
```

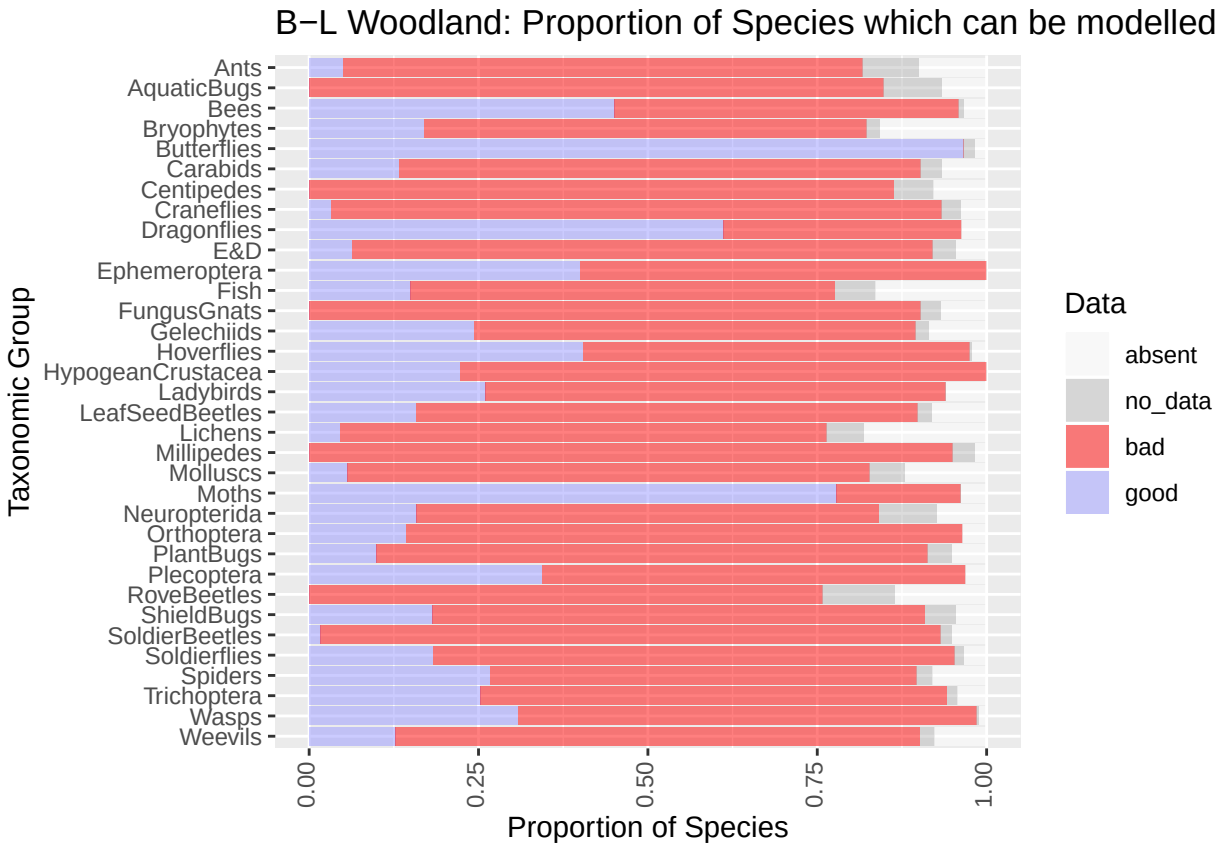

Figure 24: Proportion of species which can be modelled in broad-leaf woodland

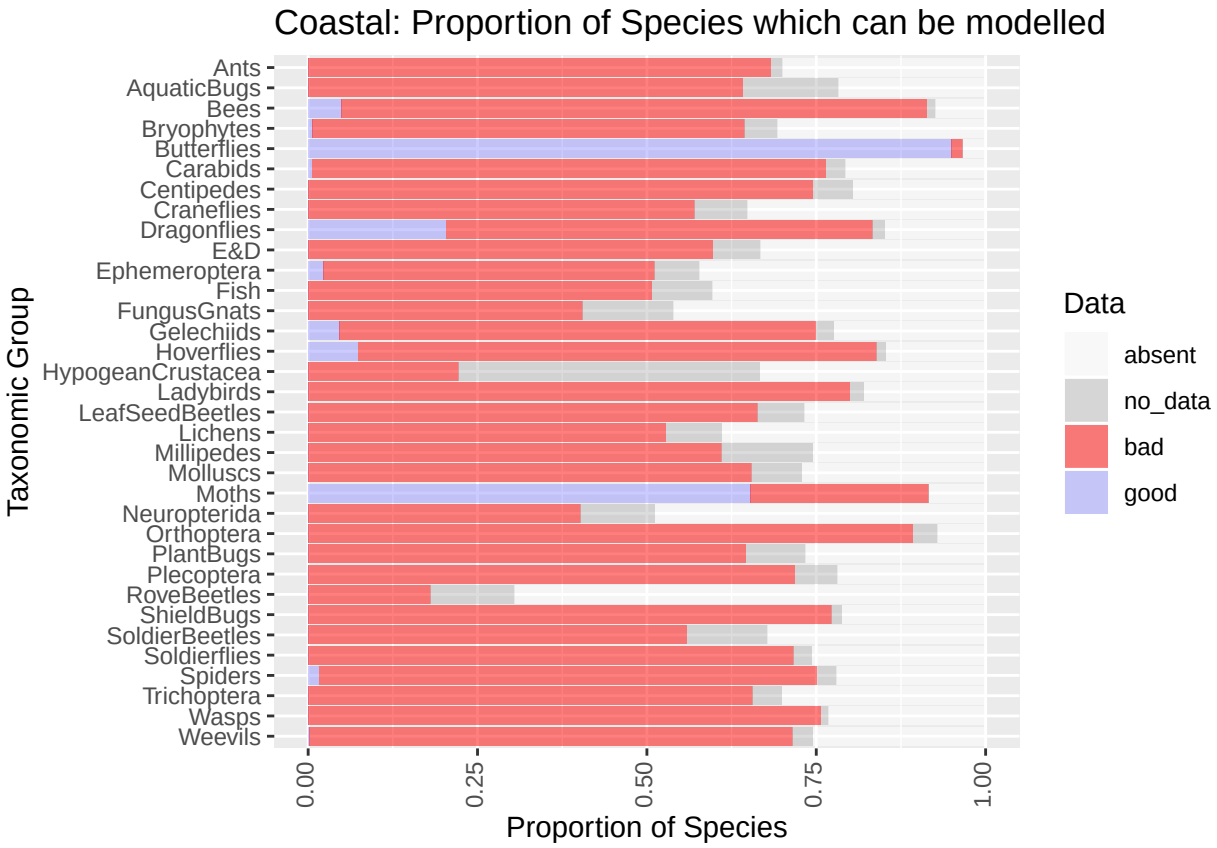

Figure 25: Proportion of species which can be modelled in coastal habitats

Table 5: List of regions

region	code
Northern Ireland	UKN
Scotland	UKM
North East (England)	UKC
North West (England)	UKD
Yorkshire and The Humber	UKE
Wales	UKL
West Midlands (England)	UKG
East Midlands (England)	UKF
South West (England)	UKK
South East (England)	UKJ
East of England	UKH
London	UKI

8.3 By region

The data can also be split by region. Splitting by region is simpler than for habitat as all squares in the UK exist within a region which can be obtained from the NUTS lookup table. The data is then subset by region and the metrics extracted for each of these regions. The available regions are shown in Table 5.

```
write.csv(x = td, file.path('./Region/RegMetrics/ALL_regMetrics.csv'))
```

```
# Read in the regional raw metrics
RM_reg <-
  read.csv(file =  file.path('Region/RegMetrics/ALL_combinedRegMetrics.csv'))
```

Below are two sample graphs for two regions:

- Proportion of species which can be modelled using only Scottish data (Figure 26)
- Proportion of species which can be modelled using only Welsh data (Figure 27)

```
# Plot up a sample graph for Scotland
num_spec(RM_reg,region='Scotland',proportional=T,absences=T) %>%
  stack_taxa(prefix='Scotland')
```

```
# And for Wales
num_spec(RM_reg,region='UKL',proportional=T,absences=T) %>%  stack_taxa(prefix='Wales')
```

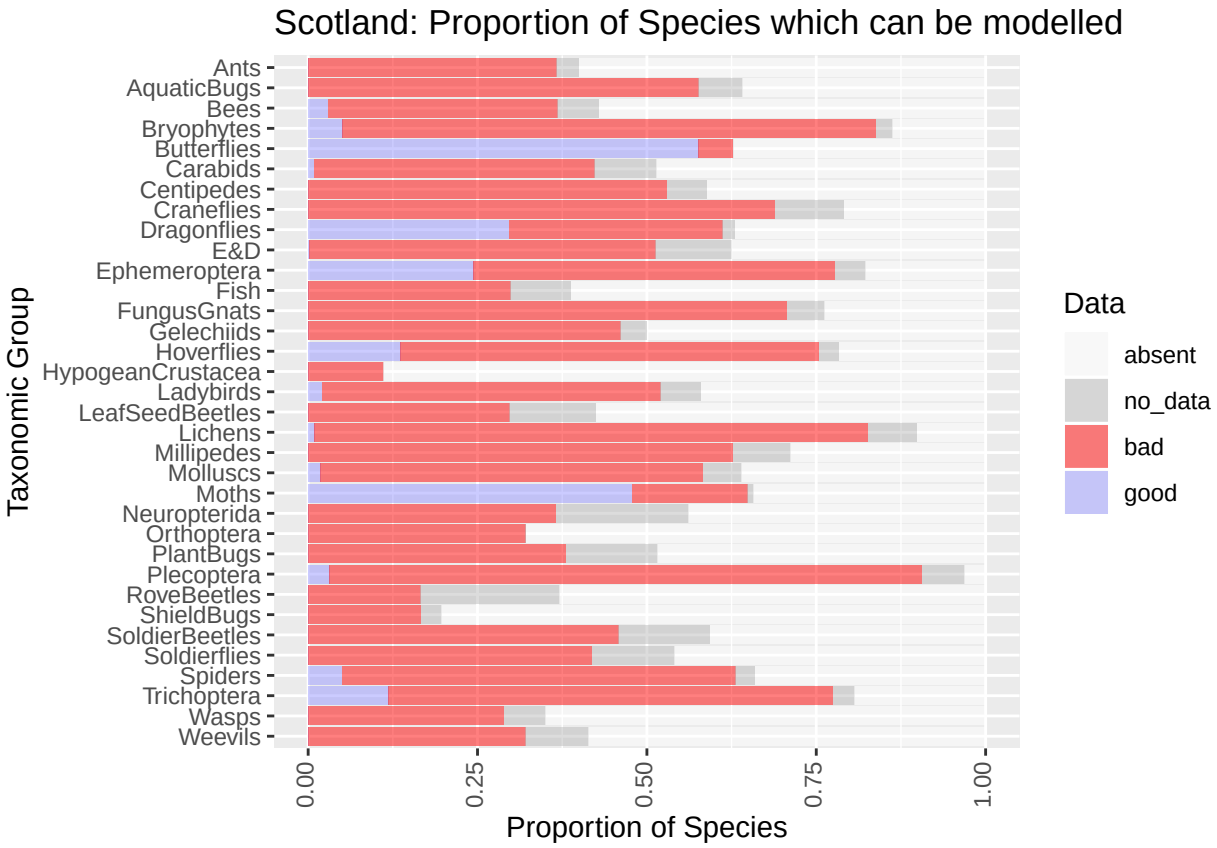

Figure 26: Proportion of species which can be modelled in Scotland

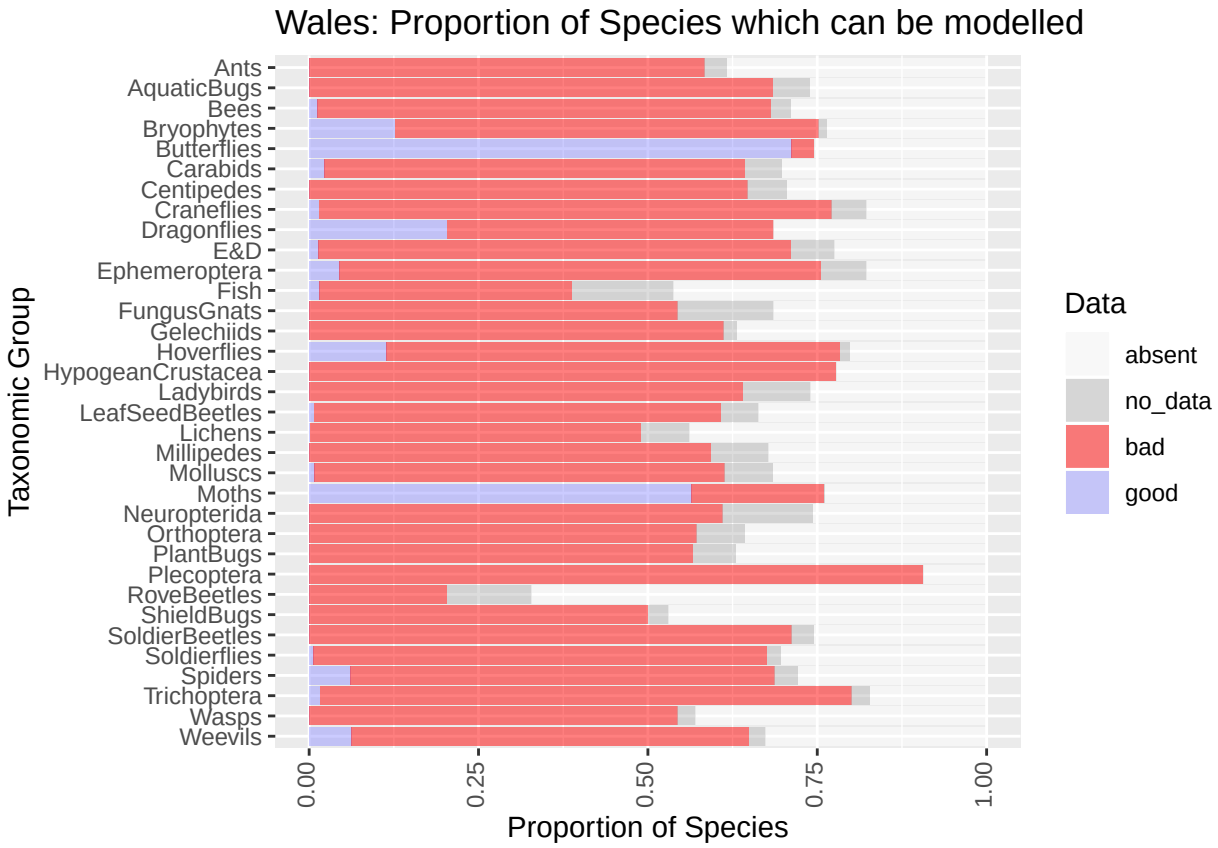

Figure 27: Proportion of species which can be modelled in Wales

9 Conclusions

Previously it was difficult to predict from a dataset whether it was possible to produce an occupancy model with a useful level of precision. This work has demonstrated that the criteria for making such an assessment are relatively simple and easy to apply to new datasets.

With this decision tree (repeated below, Figure 28), it can be estimated how many species we can model, not just in the UK as a whole, but for any region or habitat of interest. The aspirational criteria also represent a target to aim for, to enable the modelling of any species of interest.

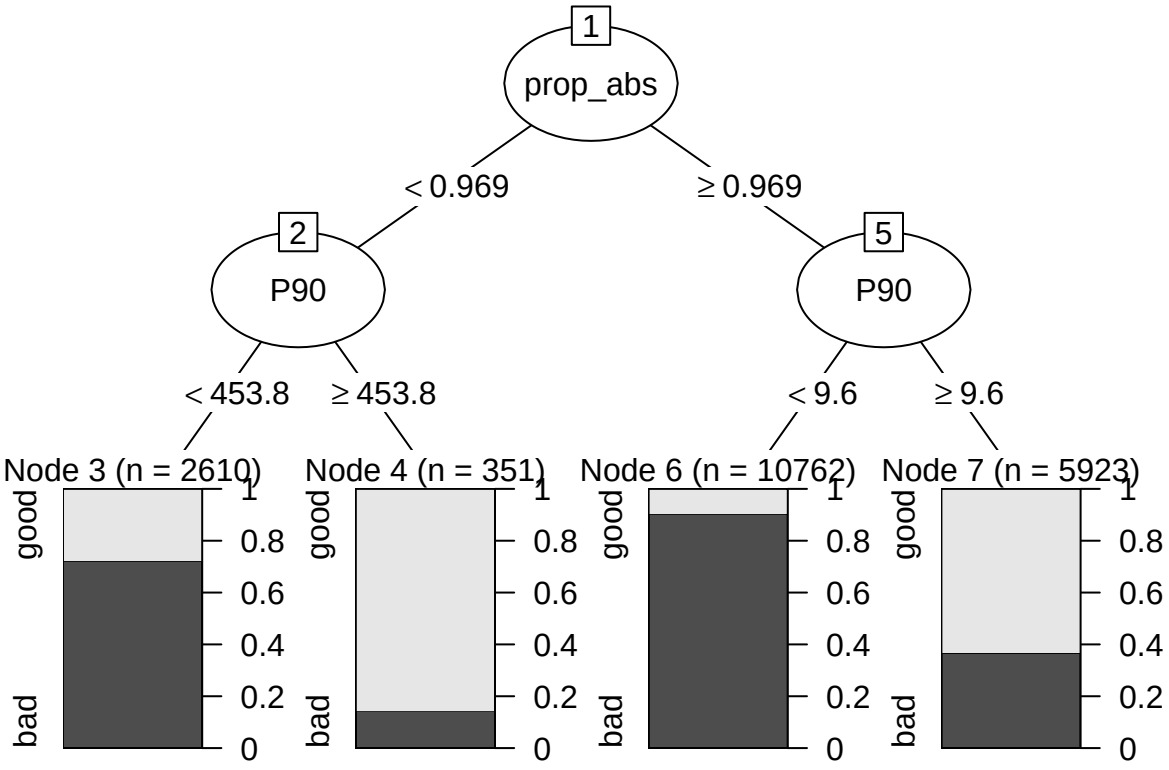

Figure 28: Best fit decision tree for deciding if a dataset is likely to produce an acceptable model