Exploratory Data Analysis with **ModelMap**

Elizabeth A. Freeman, Tracey S. Frescino, Gretchen G. Moisen

July 2, 2016

Abstract

The **ModelMap** package (Freeman, 2009) for R (R Development Core Team, 2008) now includes tools for exploratory data analysis. The `model.explore` function offers several graphical tools for exploring predictor correlation, and relationships between the available training data set and the raster files of predictor data over which the map will be made. It will identify regions of the map where the predictor lies outside the range of the training data, and show the distribution of training data over the range of each predictor.

1 Introduction

2 Exploratory Data Analysis

The `model.explore` function offers several graphical tools for exploring predictor correlation, and relationships between the training data and the area to be mapped. The function identifies regions of the map where predictors fall outside the range of the training data, and shows the distribution of training data and the map pixels over the range of each predictor.

The `model.explore` function can be used for both continuous and factored predictors, and for binary, categorical, and continuous responses.

2.1 Example dataset

The data set is from a pilot study in Nevada launched in 2004 involving acquisition and photo-interpretation of large-scale aerial photography, the Nevada Photo-Based Inventory Pilot (NPIP) (Frescino et al., 2009). The data files for these examples are included in the **ModelMap** package installation in the R library directory. The data sets are under the 'external' then under 'vignetteexamples'.

The predictor data set consists of 6 predictor variables: 5 continuous variables, and 1 categorical variable (Table 1). The predictor layers are 250-meter resolution, pixel-based raster layers including Moderate Resolution Imaging Spectro-radiometer (MODIS) satellite imagery (Justice et al., 2002), a Landsat Thematic Mapper-based, thematic layer of predicted land cover, National Land Cover Data (NLCD) (Homer et al., 2004), and a topographic layer of elevation from the National Elevation Data (Gesch et al., 2002).

The continuous response variables are percent cover of Pinyon and Sage. The binary response variables are presence of Pinyon and Sage. The categorical response variable is the vegetation category: TREE, SHRUB, OTHERVEG, and NONVEG.

The MODIS data included 250-meter, 16-day, cloud-free, composites of MODIS imagery for April 6, 2005: visible-red (RED) and near-infrared (NIR) bands and 2 vegetation indices, normalized difference vegetation index (NDVI) and enhanced vegetation index (EVI) (Huete et al., 2002). The land cover and topographic layers were 30-meter products re-sampled to 250 meter using majority and mean summaries, respectively.

Name	Type	Description
ELEV250	Continuous	90m NED elevation (ft) resampled to 250m, average of 49 points
NLCD01_250	Categorical	National Land Cover Dataset 2001 resampled to 250m - min. value of 49 points
EVI2005097	Continuous	MODIS Enhanced vegetation index
NDV2005097	Continuous	MODIS Normalized difference vegetation index
NIR2005097	Continuous	MODIS Band 2 (Near Infrared)
RED2005097	Continuous	MODIS Band 1 (Red)

Table 1: Predictor variables

The rectangular subset of Nevada chosen for these maps was deliberately selected to lie along the diagonal edge of the study region to illustrate how **ModelMap** handles unsampled regions of a rectangle (Figure 2).

2.2 Spatial Raster Layers

The **ModelMap** uses the **raster** package to read spatial rasters into R. The data for predictive mapping in **ModelMap** should be in the form of pixel-based raster layers representing the predictors in the model. The layers must be a file type recognizable by the **raster** package, for example ERDAS Imagine image (single or multi-band) raster data formats, having continuous or categorical data values. If there is more than one raster layer, the layers must have the same extent, projection, and pixel size.

2.3 Raster Look Up Table

As with the `model.mapmake` function, the Raster Look Up Table (`rastLUTfn`) provides the link between the spatial rasters for map production and the column names of the Training and Test data sets. The Raster Look Up Table can be given as an R data frame specified by the argument `rastLUTfn` or read in from a CSV file specified by `rastLUTfn`.

The `rastLUTfn` must include 3 columns: (1) the full path and base names of the raster file or files; (2) the column headers from the Training and Test data sets for each predictor; (3) the layer (band) number for each predictor. The names (column 2) must match not only the column headers in Training and Test data sets (`qdata.trainfn` and `qdata.testfn`), but also the predictor names in the arguments `predList` and `predFactor`, and the predictor names in `model.obj`.

In a windows environment, the function `build.rastLUT()` may be used to help build the look-up-table with the aid of GUI prompts.

2.4 Device Types for Graphical Output

These are complex graphics, and can be slow to produce, particularly for large maps. If the `"default"` on-screen graphics device is used, it is very important not to move or close the window till `model.explore()` is completely finished running. If you attempt to close or move or use the scroll wheel while the mouse is hovering over the the graphics device before the function is completed, there is a risk of crashing the entire R session, and loosing work.

Therefore `device.type="default"` is disabled unless `allow.default.graphics` is set to `TRUE`. If you decide to enable `device.type="default"` then it is recommended that you save all work first.

Otherwise, it is safer to have the output sent directly to a file, by setting `device.type="jpeg"`, `"pdf"`, `"png"`, etc...

3 Examples

3.1 Set up

After installing the **ModelMap** package, find the sample data sets from the R installation and copy them to your working directory. The data consists of five files and is located in the vignette directory of **ModelMap**, for example, in `C:\R\R-2.15.0\library\ModelMap\vignettes`.

There are 5 files:

VModelMapData.csv
VModelMapData_LUT.csv
VModelMapData_dem_ELEVM_250.img
VModelMapData_modis_STK2005097.img
VModelMapData_nlcd_NLCD01_250.img

Load the **ModelMap** package.

```
R> library("ModelMap")
```

Next define some of the arguments.

Define training and test data file names. Note that the arguments `qdata.trainfn` and `qdata.testfn` will accept either character strings giving the file names of CSV files of data, or the data itself in the form of a data frame.

```
R> qdatafn <- "VModelMapData.csv"
R> qdata.trainfn <- "VModelMapData_TRAIN.csv"
R> qdata.testfn  <- "VModelMapData_TEST.csv"
```

Define the output folder.

```
R> folder <- getwd()
```

Split the data into training and test sets. In example 1, an independent test set is used for model validation diagnostics. The function `get.test()` randomly divides the original data into training and test sets. This function writes the training and test sets to the folder specified by `folder`, under the file names specified by `qdata.trainfn` and `qdata.testfn`. If the arguments `qdata.trainfn` and `qdata.testfn` are not included, file names will be generated by appending `"_train"` and `"_test"` to `qdatafn`.

```
R> get.test(       proportion.test=0.2,
                  qdatafn=qdatafn,
                  seed=42,
                  folder=folder,
                  qdata.trainfn=qdata.trainfn,
                  qdata.testfn=qdata.testfn)
```

Define the predictors and define which predictors are categorical. Example 1 uses five continuous predictors: the four predictor layers from the MODIS imagery plus the topographic elevation layer. As none of the chosen predictors are categorical set `predFactor` to `FALSE`.

```
R> predList <- c( "ELEV250",
                 "NLCD01_250",
                 "EVI2005097",
                 "NDV2005097",
                 "NIR2005097",
                 "RED2005097")
R> predFactor <- c("NLCD01_250")
```

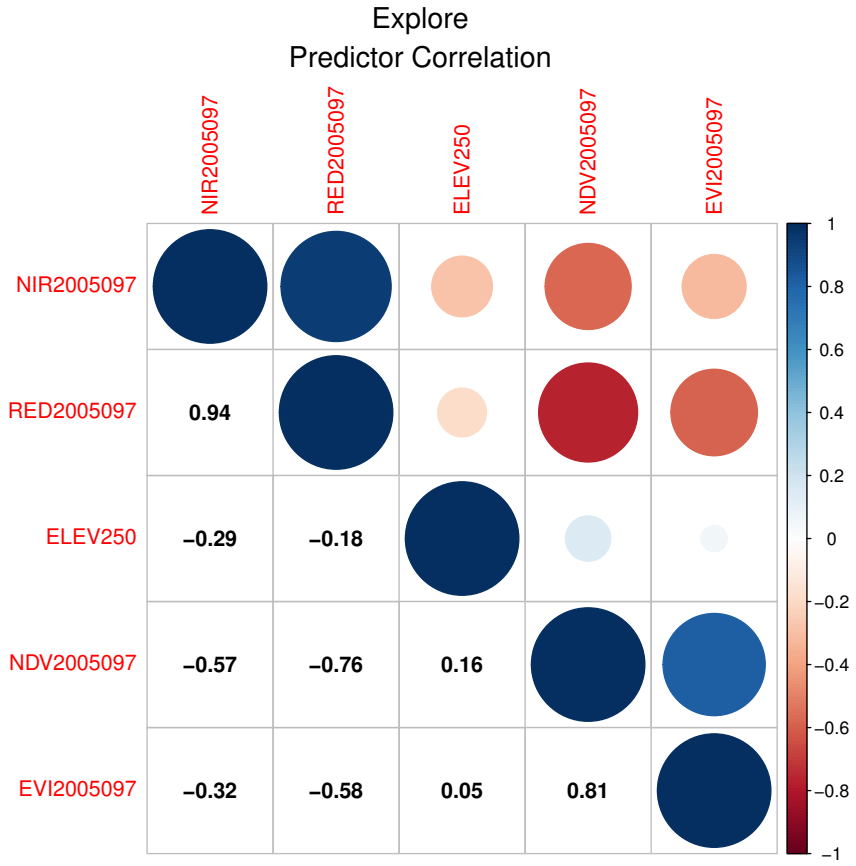

Figure 1: Correlation of continuous predictor variables.

Define the column that contains unique identifiers for each data point. These identifiers will be used to label the output file of observed and predicted values when running model validation.

```
R> unique.rowname <- "ID"
```

Define raster look up table.

```
R> rastLUTfn      <- "VModelMapData_LUT.csv"
R> rastLUTfn      <- read.table( rastLUTfn,
                                 header=FALSE,
                                 sep=",",
                                 stringsAsFactors=FALSE)
R> rastLUTfn[,1] <- paste(folder,rastLUTfn[,1],sep="/")
```

4 Predictor Correlation

The `model.explore` function uses the **corrplot** package (Wei, 2013) to produce a graphical correlation matrix for all the continuous (non-factored) predictor variables (Figure 1).

5 Example 1 - Continuous Response

We will start by looking at Pinyon Percent Cover as a response value.

To produce exploratory plots for all predictor variables and a given response variable call `model.explore`.

The argument `col.ramp` allows you to set the color ramp used to map continuous predictors, while the argument `col.cat` allows you to set the colors used to map the categorical predictors. Note that you can only specify one color ramp and one set of color categories each time you run `model.explore`. However, you can run `model.explore` several times on different subsets of your predictor variables. For example, we could run it once with `col.ramp=terrain.colors(101)` and `predList="ELEV250"`, then run it again with a different color ramp and for the remote sensing predictor variables.

```
R> model.explore( qdata.trainfn=qdata.trainfn,
                  folder=folder,
                  predList=predList,
                  predFactor=predFactor,

                  OUTPUTfn="PinyonCover",

                  response.name="PINYON",
                  response.type="continuous",

                  unique.rowname=unique.rowname,

                  device.type=c("png"),
                  #cex=1.2,

                  # Raster arguments
                  rastLUTfn=rastLUTfn,
                  na.value=-9999,

                  # colors for continuous predictors
                  col.ramp=terrain.colors(101),

                  #colors for categorical predictors
                  col.cat=c("wheat1","springgreen2","darkolivegreen4",
                            "darkolivegreen2","yellow","thistle2",
                            "brown2","brown4")
 )
```

5.1 Example 1 - Continuous Predictor - Continuous Response

As an example of a continuous spatial predictor raster, lets look at elevation. The sub-region of the study area used in this vignette contains part of a small mountain range surrounded by plains, and lies along the diagonal southeast edge of Nevada.

It is possible to produce a basic map of elevation (Figure 2) from the raster package alone:

```
R> elevfn <- paste(folder,"/VModelMapData_dem_ELEVM_250.img",sep="")
R> mapgrid <- raster(elevfn)

R> opar <- par(mar=c(4,4,3,6),xpd=NA,mgp=c(3, 2, .3))
R> col.ramp<-terrain.colors(101)
R> zlim <- c(1500,maxValue(mapgrid))
```

```
R> legend.label<-rev(pretty(zlim,n=5))
R> legend.colors<-col.ramp[trunc((legend.label/max(legend.label))*100)+1]
R> legend.label<-paste(legend.label,"m",sep="")
R> legend.label<-paste((7:3)*500,"m")
R> legend.colors<-col.ramp[c(100,75,50,25,1)]
R> image( mapgrid,
        col = col.ramp,
        xlab="", ylab="",
        zlim=zlim,
        asp=1, bty="n", main="")
R> legend( x=xmax(mapgrid),y=ymax(mapgrid),
         legend=legend.label,
         fill=legend.colors,
         bty="n",
         cex=1.2)
R> mtext("Elevation of Study Region",side=3,line=1,cex=1.5)
R> par(opar)
```

However, this map doesn't tell us anything about how this predictor raster relates to the elevations of the training data.

The `model.explore` functions illustrates graphically the relationships between the training data and the predictor rasters. The map produced by the `model.explore` function, masks out all pixels with NA values in light gray, and predictor values outside the range of the training data in black. In this vignette, the light gray triangle at the souteast edge of all the maps represents a map region that lies outside of Nevada, and therefore no data was collected.

In the exploratory plot for Pinyon Percent Cover as a function of Elevation (Figure 3) the black regions near the center of the map are where the mountain top reaches elevations higher than any of the training data, therefore any model predictions for this region will be extrapolations. The ranges given in the text below the map indicate the range of the training data and that of the predictor raster. Here you can see that the area covered by the raster ranges from 1560m to 3461m, while the highest training plot was only at 3084m.

The regions of the map that lie outside the range of the training data will be different for each predictor variable. It is up to the user to determine if in the case of their particular model it is worth collecting additional training data to improve predictions in these regions. If it is known that a particular species does not occur above or below a given elevation, then collecting additional training data may not be worth while. For example, if one is interested in mapping Pinyon Pine, high elevation training data is less important than if one is interested in Bristlecone Pine.

The right half of the exploratory graphic has four smaller figures.

The graphic on the upper left is a scatter plot of the response variable (Pinyon percent cover) as a function of the predictor variable (elevation). The red line is a basic Generalized Additive Model (GAM) of the relationship of Response variable to this predictor.

The upper right graphic has box plots of the predictor variable over the training data and the raster pixels. Here you can see that the subsection of the study area used for this vignette is missing the lower elevations from the training data. This is due to the fact that the training data was collected over the entire state of Nevada, while we are using a raster of a tiny section of the state for the vignette.

The graphic on the bottom left is a histogram of the number of training plots by elevation, while the bottom right is a similar histogram of number of map pixels by elevation. The colors in these histograms correspond to the map colors. Again, you can see that the raster we are using is missing any low elevation pixels.

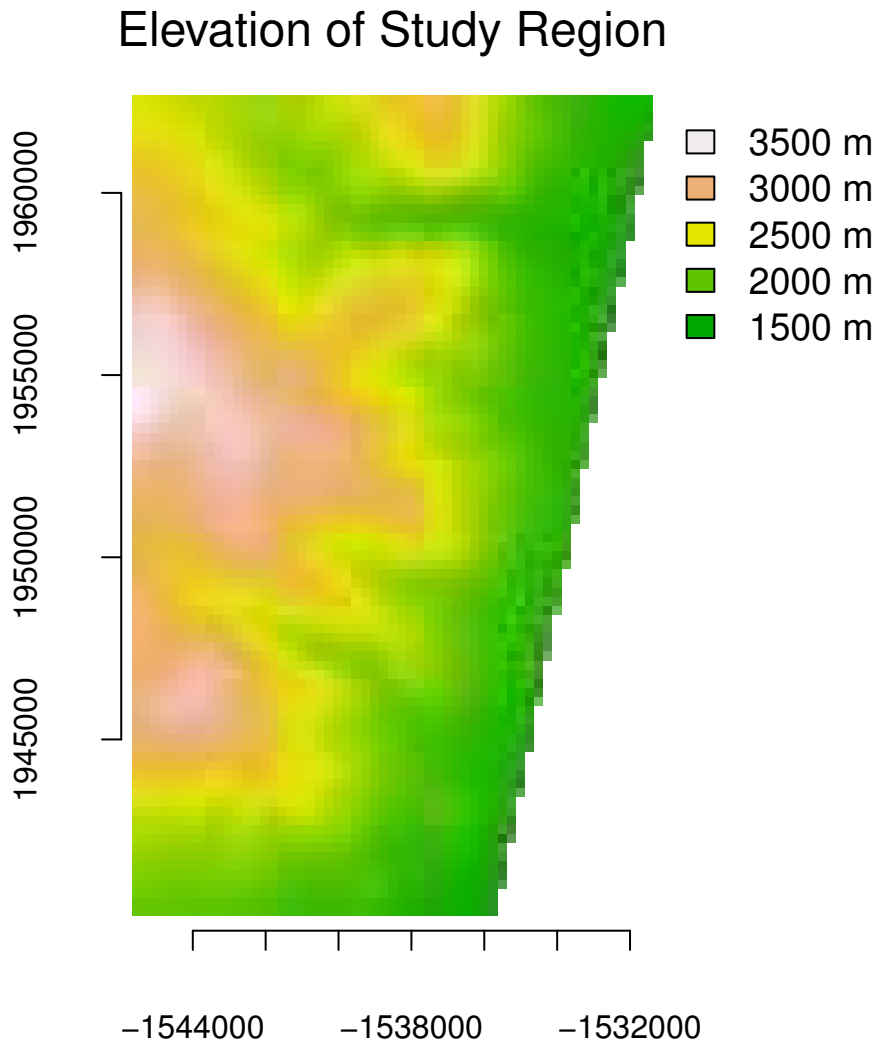

Figure 2: Elevation of study region. Projection: Universal Transverse Mercator (UTM) Zone 11, Datum: NAD83

5.2 Example 1 - Categorical Predictor - Continuous Response

Next we will look at an example of an exploratory plot for a categorical predictor while keeping Pinyon Percent Cover as our continuous response. In this case, we will look at the National Land Cover Data (NLCD) as a predictor (Figure 4).

With categorical predictors, the map masks out categories not found in the training data, and indicates these categories in the map legend with asterisks. In this case, no training data was collected in NLCD classes 41 (deciduous forests) and 43 (mixed forests). The user can then decide to either collect additional training data in these classes, collapse classes (for example, combine deciduous and mixed forests with evergreen forests to create a single "Forest" class), or to leave the original classes, and accept that these pixels will be predicted as "NA".

With categorical predictors, the right side of the exploratory plot gives three bar charts. The top one shows the mean response value for each category in the training data. Here we can see that the only NLCD class present in the training data with any appreciable Pinyon cover is class 42 (evergreen forest).

The middle graphic is a histogram of the number of training plots per NLCD class, while the bottom graphic shows the number of map pixels per NLCD class.

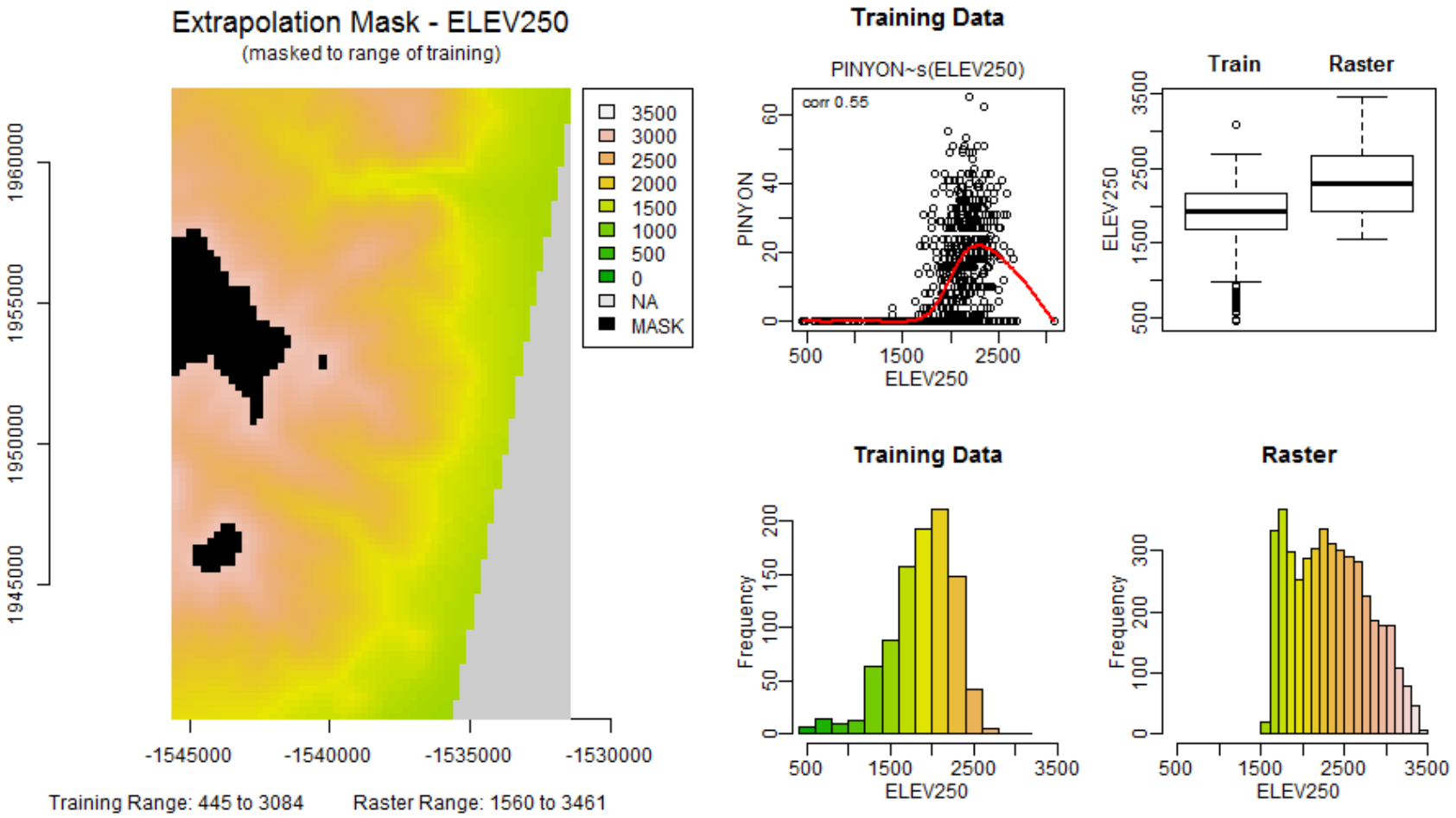

Figure 3: Exploratory plot for continuous predictor (elevation) and continuous response (Percent Cover of Pinyon). Light gray pixels in the map represent NA values in the predictors, while black pixels have values outside the range of the training data. The light gray triangle at the southeast edge represents a map region that lies outside of Nevada, and therefore no data was collected, while the black regions near the center of the map are where the mountain top reaches elevations higher than any of the training data, therefore any model predictions for this region will be extrapolations.

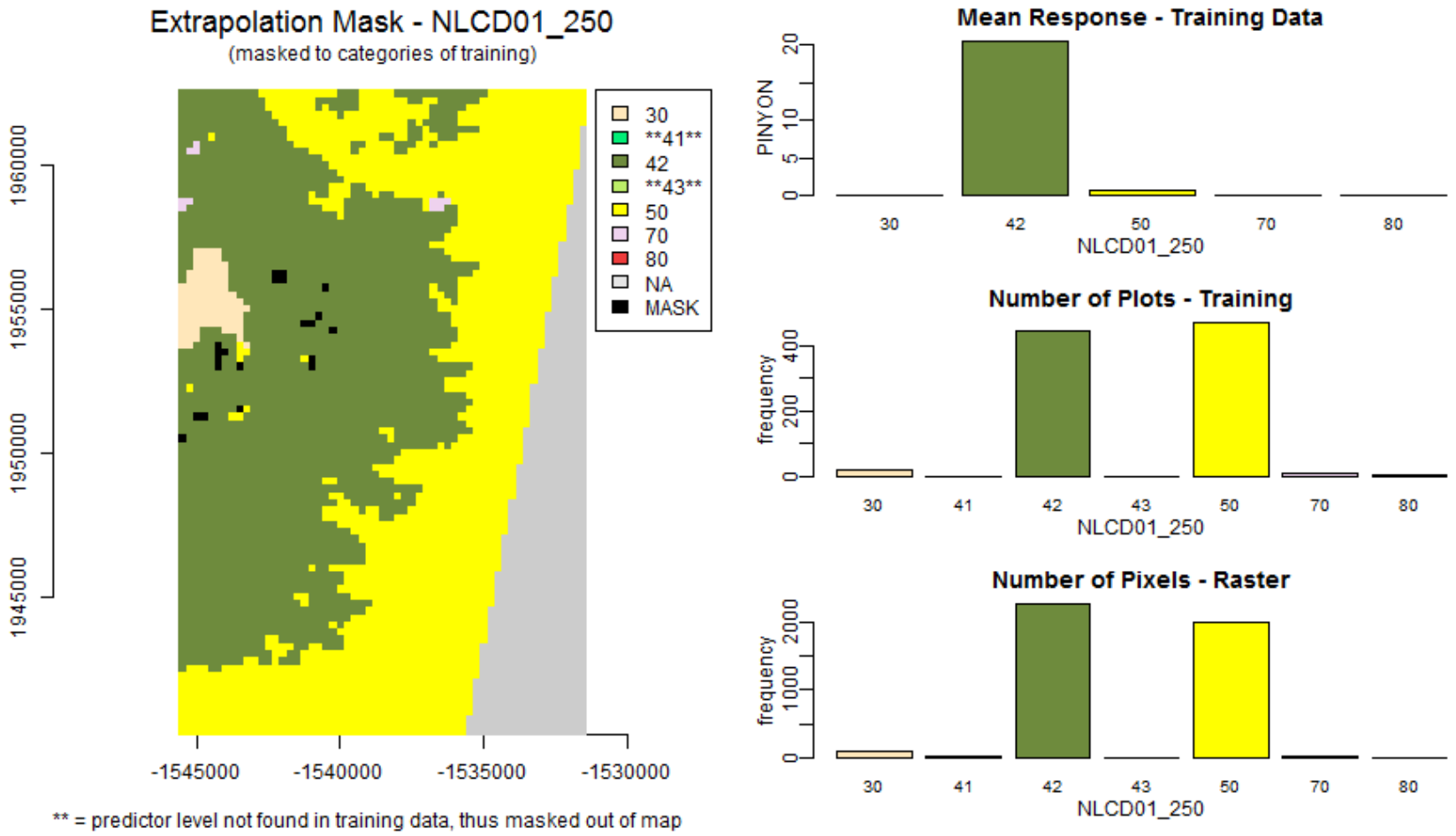

Figure 4: Exploratory plot for factored (categorical) predictor NLCD and continuous response (Percent Cover of Pinyon). Light gray pixels in the map represent NA values in the predictors, while black pixels have predictor categories not found in the training data. Categories not found in the training data are also marked with asterisks in the map legend. Here NLCD classes 41 (deciduous forest) and 43 (mixed forest) were not included in the training data, and are therefor masked out of the exploratory map.

6 Example 2 - Binary Response

Next we will look at Sage Presence-Absence as a binary response value, and a remote sensing band as a continuous predictor. We will use `col.ramp=heat.colors(101)` for the map color ramp.

```
R> model.explore( qdata.trainfn=qdata.trainfn,
                  folder=folder,
                  predList=predList,
                  predFactor=predFactor,

                  OUTPUTfn="SagePresence",

                  response.name="SAGE",
                  response.type="binary",

                  unique.rowname=unique.rowname,

                  device.type=c("png"),
                  #cex=1.2,

                  # Raster arguments
                  rastLUTfn=rastLUTfn,
                  na.value=-9999,

                  # colors for continuous predictors
                  col.ramp=heat.colors(101),

                  #colors for categorical predictors
                  col.cat=c("wheat1","springgreen2","darkolivegreen4",
                            "darkolivegreen2","yellow","thistle2",
                            "brown2","brown4")
 )
```

6.1 Example 2 - Continuous Predictor - Binary Response

In the exploratory map (Figure 5) you can see a few small areas on the map where the values of the Near Infra Red (NIR2005097) remote sensing band are higher or lower than the values found in the training data.

On the right side, the graphic on the upper left shows relative proportions of Sage Presence to Sage Absence in the training data as a function of NIR2005097. This shows that for the training data plots with values of NIR2005097 between roughly 3250 and 5250 had the highest probability of sage presence.

The box plots in the upper right show that the median values of NIR2005097 is quite similar between the training data and the raster pixels, though the raster pixels had a slightly greater spread.

Note that the box plot for the training data shows outliers, while the box plot for the raster does not. This is due to the method used to produce the plots, rather than a lack of outliers in the raster. The `boxplot` function in the **raster** package does not offer the option of showing outliers. The **raster** package uses sampling on large rasters when making the box plot, to keep memory usage reasonable, but this means it does not check every single pixel to locate all outliers.

Again, the two bottom graphics are again histograms of plot and raster frequency as a function of the predictor variable. Here we see that training data and raster data are roughly similar

across the range of NIR2005097, with most of the training data and raster pixels lying at values of NIR2005097 close to 2000. Though as seen on the box plot, the raster pixels have a wider range than the training data.

6.2 Example 2 - Categorical Predictor - Binary Response

This data set only has only categorical predictor (NLCD) so the exploratory plot is very similar to that of Example 1 (Figure 6).

The map and the lower two graphics on the right are based purely on the predictor variable, therefore changing in response variable from Pinyon Percent Cover to Sage Presence has no effect on these portions of the graphic.

The change to a binary response variable, however does change the upper right graphic. For binary Presence-Absence response variables, such as Sage presence, the upper graphic shows the relative proportion of presences and absences in each predictor category of the training data. Here we can see that Sage is relatively common in NLCD categories 42 (evergreen forest), 50 (shrub land), and 70 (herbaceous), very rare in category 30 (barren), and almost non-existent in category 80 (cultivated). Categories 41 (deciduous forest) and 43 (mixed forest) are not found in the training data, and are therefore not included in this graphic.

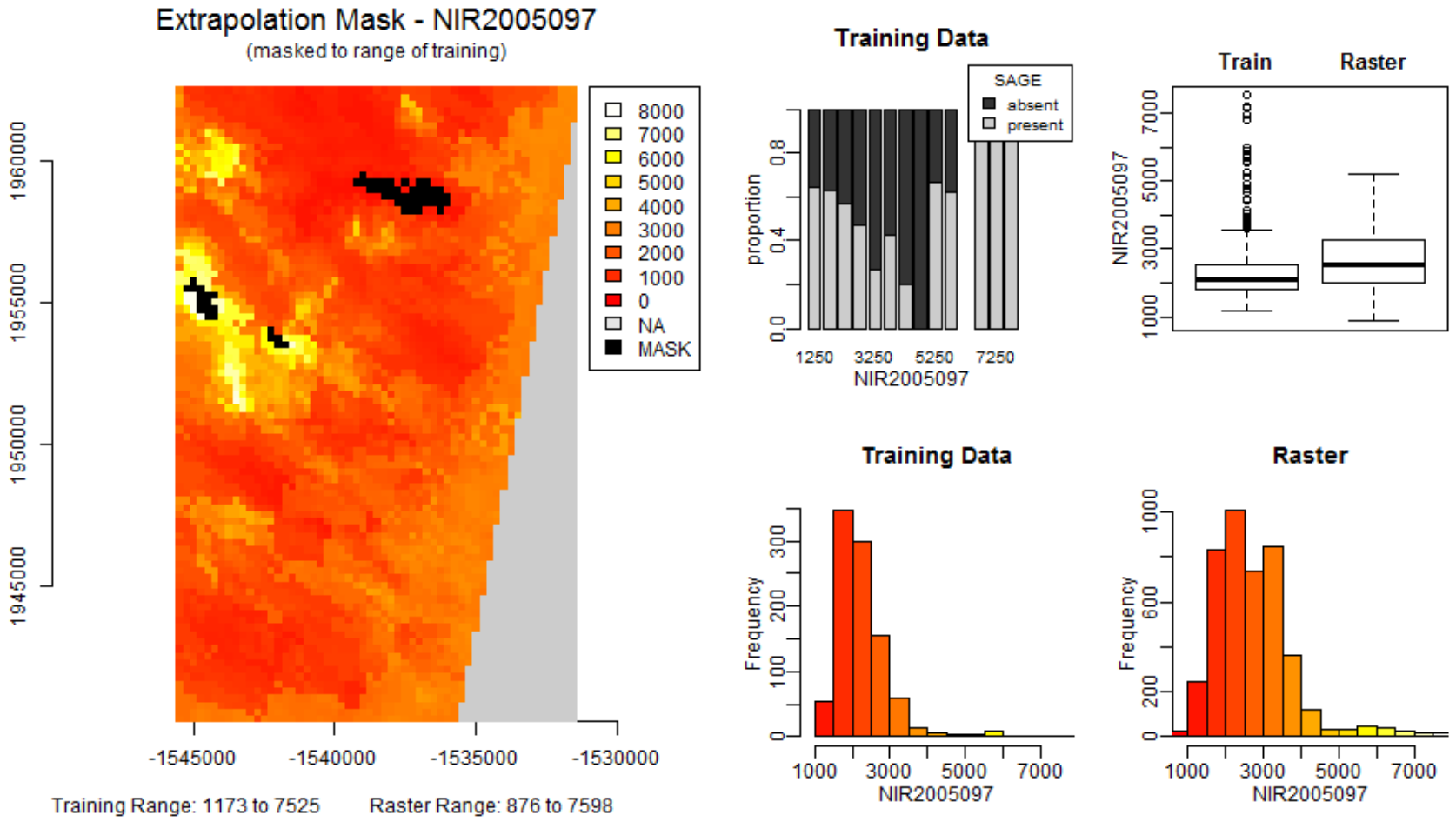

Figure 5: Exploratory plot for continuous predictor (Near Infared - NIR2005097) and binary response (Sage Presence-Absence). Light gray pixels in the map represent NA values in the predictors and black pixels are predictor values outside the range of the training data. The light gray triangle at the southeast edge represents a map region that lies outside of Nevada, and therefore no data was collected, while the black regions are where the map pixels have values higher or lower than that of the training data, therefore any model predictions for this region will be extrapolations.

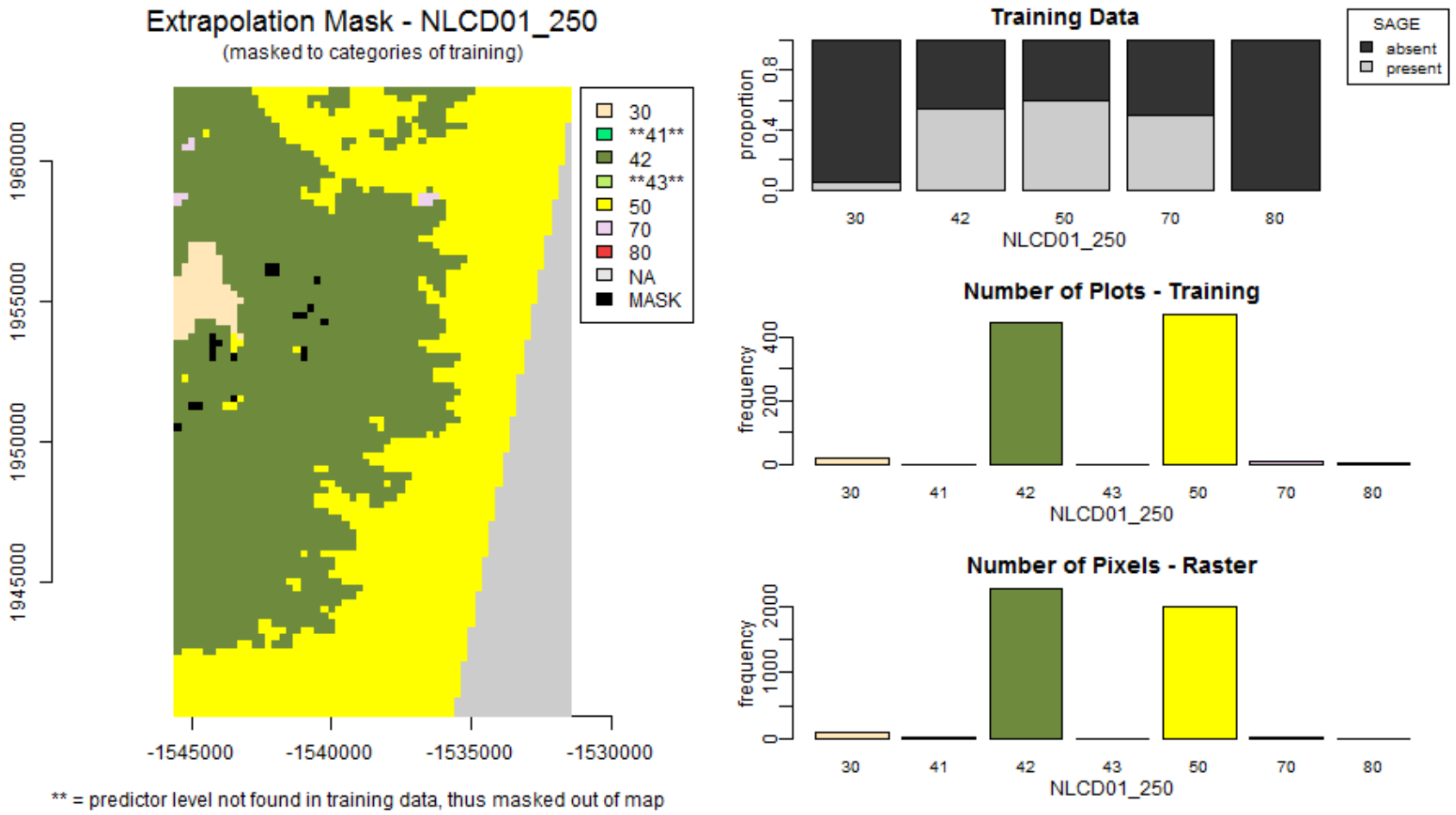

Figure 6: Exploratory plot for factored (categorical) predictor NLCD and binary response (Sage Presence-Absence). Light gray pixels in the map represent NA values in the predictors and black pixels are predictor categories not found in the training the training data. Categories not found in the training data are also marked with asterisks in the map legend. Here NLCD classes 41 (deciduous forest) and 43 (mixed forest) were not included in the training data, and are therefor masked out of the exploratory map.

7 Example 3 - Categorical Response

Example 3 builds a categorical response model for vegetation category. The response variable consists of four categories: TREE, SHRUB, OTHERVEG, and NONVEG.

```
R> model.explore( qdata.trainfn=qdata.trainfn,
                  folder=folder,
                  predList=predList,
                  predFactor=predFactor,

                  OUTPUTfn="VegCat",

                  response.name="VEGCAT",
                  response.type="categorical",

                  unique.rowname=unique.rowname,

                  device.type=c("png"),
                  #cex=1.2,

                  # Raster arguments
                  rastLUTfn=rastLUTfn,
                  na.value=-9999,

                  # colors for continuous predictors
                  col.ramp=heat.colors(101),

                  #colors for categorical predictors
                  col.cat=c("wheat1","springgreen2","darkolivegreen4",
                            "darkolivegreen2","yellow","thistle2",
                            "brown2","brown4")
 )
```

7.1 Example 3 - Continuous Predictor - Categorical Response

In the exploratory map (Figure 7) you can see a few small areas on the map where the values of the Near Infra Red (NIR2005097) remote sensing band are higher or lower than the values found in the training data.

On the right side, the graphic on the upper left shows relative proportions of Sage Presence to Sage Absence in the training data as a function of NIR2005097. This shows that for the training data plots with values of NIR2005097 between roughly 3250 and 5250 had the highest probability of sage presence.

The box plots in the upper right show that the median values of NIR2005097 is quite similar between the training data and the raster pixels, though the raster pixels had a slightly greater spread.

Note again that the box plot for the training data shows outliers, while the boxplot for the raster does not. This is due to the method used to produce the plots, rather than a lack of outliers in the raster.

Again, the two bottom graphics are again histograms of plot and raster frequency as a function of the predictor variable. Here we see that training data and raster data are roughly similar across the range of NIR2005097, with most of the training data and raster pixels lying at values of NIR2005097 close to 2000. Though as seen on the box plot, the raster pixels have a wider range than the training data.

7.2 Example 3 - Categorical Predictor - Categorical Response

This dataset only has only categorical predictor (NLCD) so the exploratory plot is very similar to that of Example 1 and 2 (Figure 8).

The change to a binary response variable, however does change the upper right graphic. The upper right graphic shows the relative proportion of the response categories in each predictor category of the training data. Here we can see that in the training data Vegcat=TREE is predominatly found in NLCD categories 42 (evergreen forest). Vegcat=SHRUB is modtly found in 50 (shrubland), though there are also some SHRUB plots in 70 (herbaceous). Vegcat=OTHERVEG dominates 70 (herbaceous) and 80 (cultivated). Not surprisingly most NLCD category 30 (barren) training plots are Vegcat=NONVEG, but there are also substation proportions of NLCD 42 (evergreen forest) and 50 (shrubland) classified as Vegcat=NONVEG.

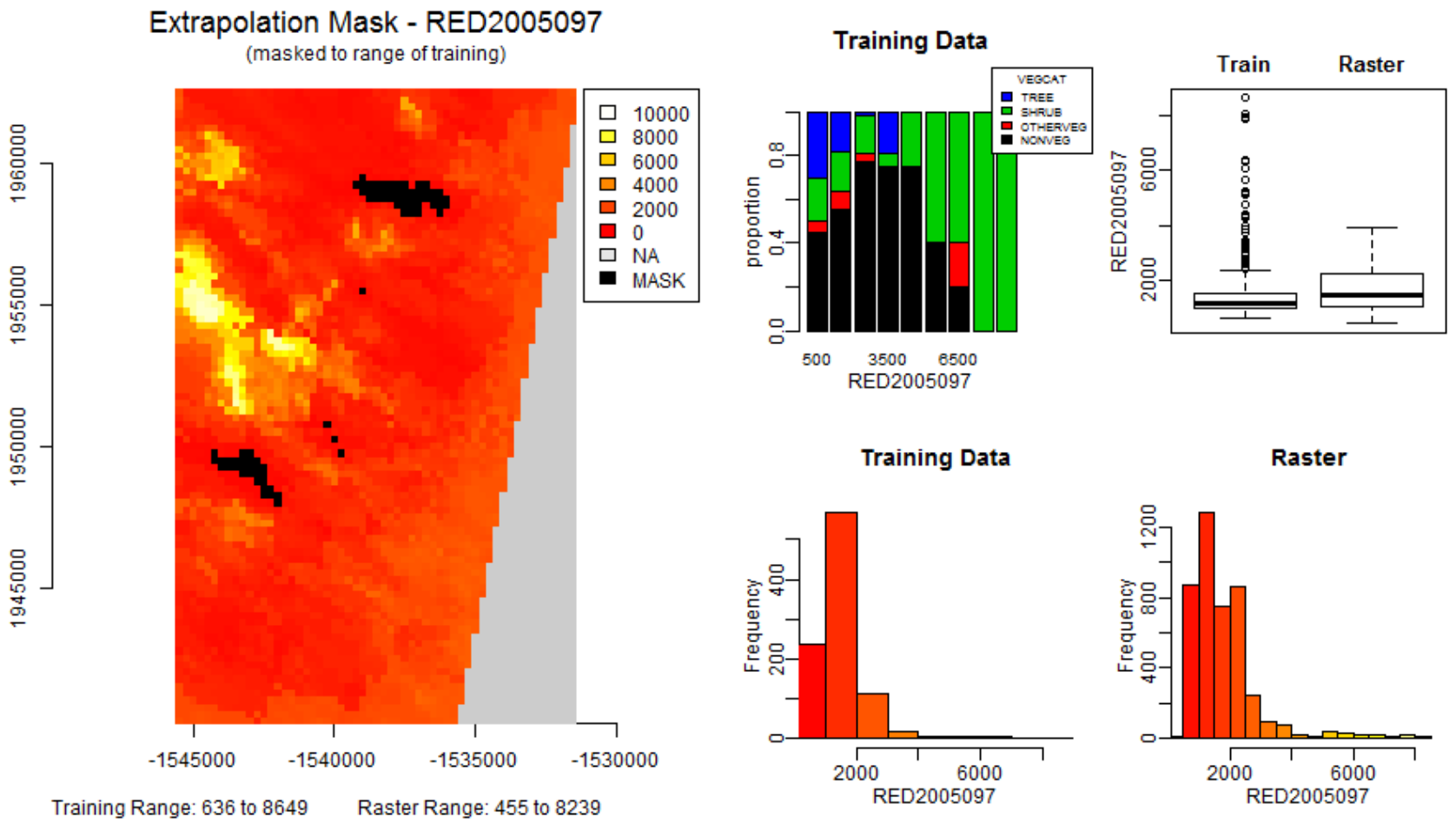

Figure 7: Exploratory plot for continuous predictor (Red band - RED2005097) and categorical response (Vegetation category - VEGCAT). Light gray pixels in the map represent NA values in the predictors and black pixels are predictor values outside the range of the training data. The light gray triangle at the southeast edge represents a map region that lies outside of Nevada, while the black regions where the map pixels have values higher or lower than that of the training data, therefore any model predictions for this region will be extrapolations.

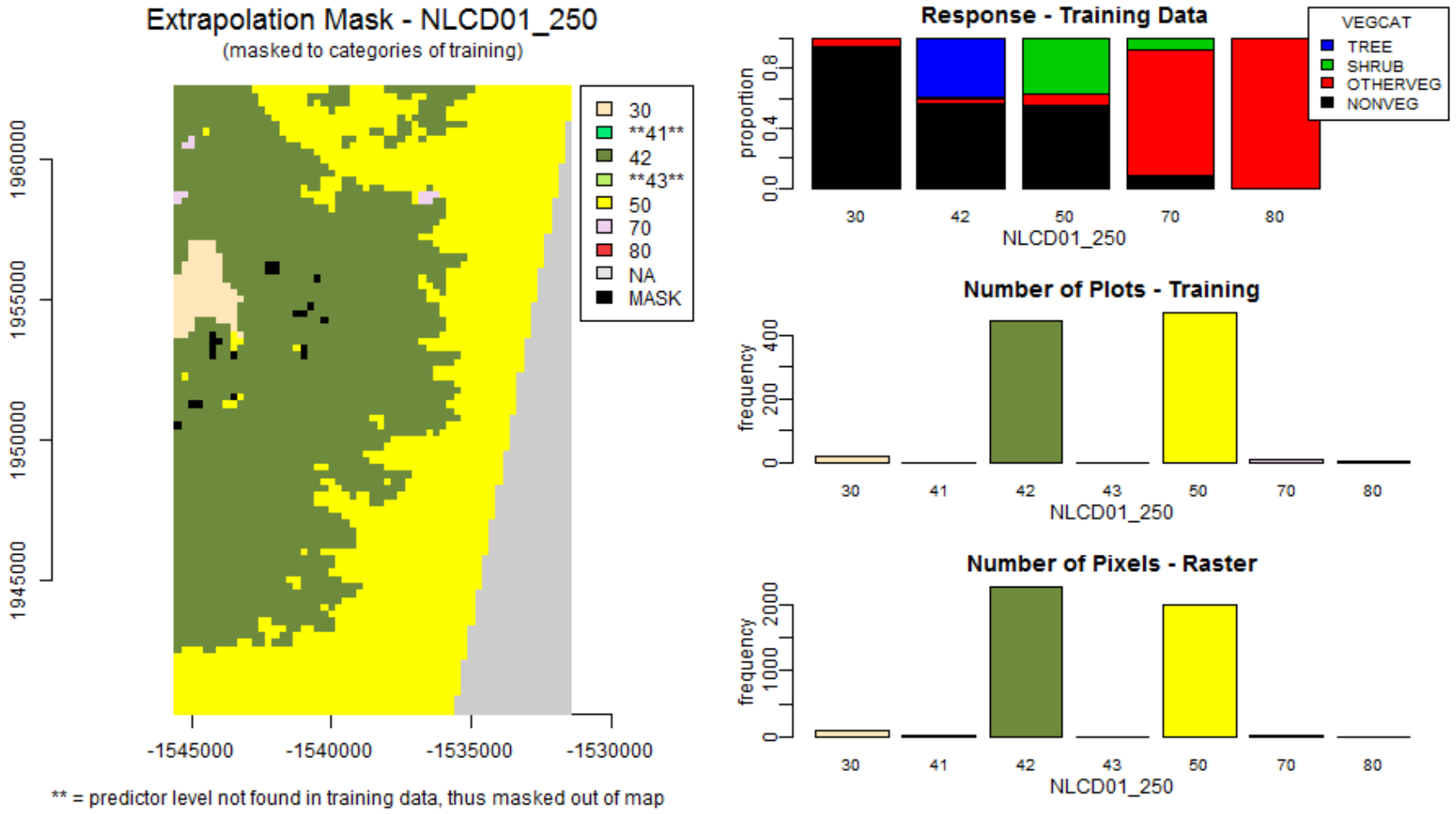

Figure 8: Exploratory plot for factored (categorical) predictor NLCD and categorical response (Vegetation category - VEGCAT). Light gray pixels in the map represent NA values in the predictors and black pixels are predictor categories not found in the training the training data. Categories not found in the training data are also marked with asterisks in the map legend. Here NLCD classes 41 (deciduous forest) and 43 (mixed forest) were not included in the training data, and are therefor masked out of the exploratory map.

8 Overall Mask for All Predictors

The `model.explore` function outputs an overall mask raster showing pixels where any of the predictors had NA values or values lying outside the range of the training data (Figure 9). This can be used on the final prediction map, to mask out regions were the model is extrapolating.

If you wish to save the individual masks for each predictor variable, set argument `create.extrapolation.masks=TRUE` and a multi layer raster will be produces with one layer for each predictor variable in `predList`. The layers in this file will be in the same order as the predictors in `predList`.

References

E. Freeman. ***ModelMap****: An R Package for Modeling and Map production using Random Forest and Stochastic Gradient Boosting*. USDA Forest Service, Rocky Mountain Research Station, 507 25th street, Ogden, UT, USA, 2009. URL `http://CRAN.R-project.org/`. eafreeman@fs.fed.us.

T. S. Frescino, G. G. Moisen, K. A. Megown, V. J. Nelson, Elizabeth, Freeman, P. L. Patterson, M. Finco, K. Brewer, and J. Menlove. Nevada photo-based inventory pilot(npip) photo sampling procedures. Gen. Tech. Rep. RMRSGTR-222, U.S. Departmentof Agriculture, Forest Service, Rocky Mountain Research Station., Fort Collins, CO, 2009.

D. Gesch, M. Oimoen, S. Greenlee, C. Nelson, M. Steuck, and D. Tyler. The national elevation dataset. photogrammetric engineering and remote sensing. *Photogrammetric Engineering and Remote Sensing*, 68:5–11, 2002.

C. Homer, C. Huang, L. Yang, B. Wylie, and M. Coan. Development of a 2001 national land-cover database for the united states. *Photogrammetric Engineering and Remote Sensing*, 70:829–840, 2004.

A. Huete, K. Didan, T. Miura, E. P. Rodriguez, X. Gao, and L. G. Ferreira. Overview of the radiometric and biophysical performance of the modis vegetation indices. *Remote Sensing of Environment*, 83:195–213, 2002.

C. O. Justice, J. R. G. Townshend, E. F. Vermote, E. Masuoka, R. E. Wolfe, N. Saleous, D. P. Roy, and J. T. Morisette. An overview of modis land data processing and product status. *Remote Sensing of Environment*, 83:3–15, 2002.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

T. Wei. *corrplot: Visualization of a correlation matrix*, 2013. URL `http://CRAN.R-project.org/package=corrplot`. R package version 0.73.

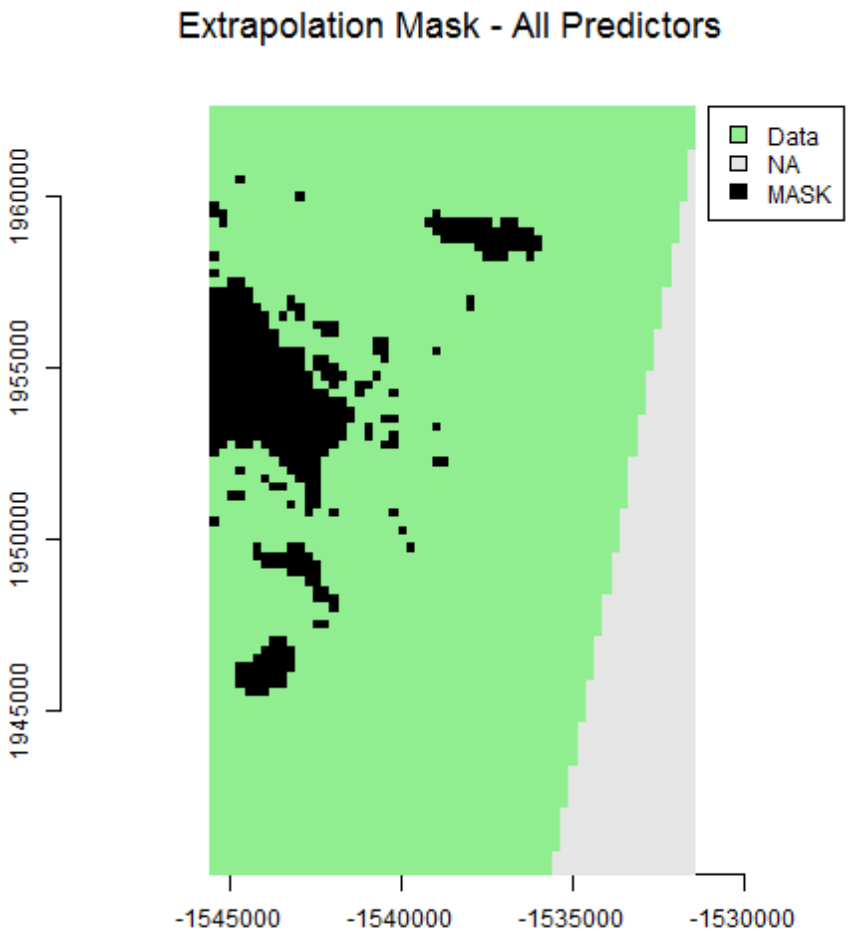

Figure 9: Map showing all pixels where at least one of the predictor variables is NA (light gray) or is outside the range of the training data (black).