

kum_glm

jiaxins

2025-12-21

```
# install.packages('topmodels', repos = c('https://R-Forge.R-project.org', 'http://cran.at.r-project.org'))
library(ggplot2)
library(gamair)
library(dplyr)
library(mgcv)
library(tidymv)
library(statmod)
library(rTPC)
library(nls.multstart)
library(broom)
library(tidyverse)
library(pracma)
library(qgam)
library(lme4)
library(lmerTest)
library(MASS)      #for negative bin regression
library(pscl)      #for zero inflated models & predprob
library(reshape2)  #convert wide to tall
# library(topmodels)#rootograms
library(AER)
library(caTools)
library(randomForest)
library(caret)
library(e1071)
library(gridExtra)
library(pROC)
library(pdp)
library(plotly)
library(car)
library(effects)
library(metR)
library(scico)
library(akima)

theme_format <- theme_bw()+
  theme(axis.text.x = element_text(vjust=0.5,size=16, colour = "black"))+
  theme(axis.text.y = element_text(size=16, colour = "black"))+
  theme(axis.title.x = element_text(size=16, colour = "black"))+
  theme(axis.title.y = element_text(size=16, colour = "black"))+
  #panel.background = element_rect(fill="white"),
  theme(axis.ticks = element_line(colour="black"))+
  theme(panel.grid.minor=element_blank())+
```

```

theme(panel.grid.major=element_blank())+
  theme(strip.text = element_text(size = 14))+
  theme(legend.text = element_text(size = 16),
    legend.title = element_text(size = 16))

```

read data

```

df_species <- read.csv('all_urchin_kelp_mhw_level_v3.csv') #

# df_species <- df_species %>% filter(location %in% c('Jervis Bay', 'Batemans'))

data <- df_species[c("location", "site_name", "longitude", "latitude", "survey_mean",
                     "number", "summer_temp", "summer_inten", "summer_dhd50)] # "mean_imax"
data$location <- as.factor(data$location)
data$site_name <- as.factor(data$site_name)
head(data)

##   location           site_name longitude latitude
## 1 Batemans Belowla Island South West     150.39 -35.554
## 2 Batemans Belowla Island South West     150.39 -35.554
## 3 Batemans Belowla Island South West     150.39 -35.554
## 4 Batemans     Brush Island Mid North    150.42 -35.526
## 5 Batemans     Brush Island Mid North    150.42 -35.526
## 6 Batemans     Brush Island Mid North    150.42 -35.526
##   survey_mean number summer_temp summer_inten summer_dhd50
## 1          0.0    6.930      21.587       1.234     22.491
## 2          0.0    2.935      21.742       2.884     78.853
## 3          0.4    3.520      22.209       2.904     39.467
## 4          0.0    5.000      21.587       1.234     22.491
## 5          0.0    2.195      21.742       2.884     78.853
## 6          0.0    3.715      22.209       2.904     39.467

```

convert to binomial

```

data$collapse <- ifelse(data$survey_mean > 10, 0, 1)
# Convert collapse to a factor
data$collapse <- as.factor(data$collapse)
data$collapse <- factor(data$collapse, levels = c(0, 1), labels = c("class_0", "class_1"))
head(data)

```

```

##   location           site_name longitude latitude
## 1 Batemans Belowla Island South West     150.39 -35.554
## 2 Batemans Belowla Island South West     150.39 -35.554
## 3 Batemans Belowla Island South West     150.39 -35.554
## 4 Batemans     Brush Island Mid North    150.42 -35.526
## 5 Batemans     Brush Island Mid North    150.42 -35.526
## 6 Batemans     Brush Island Mid North    150.42 -35.526
##   survey_mean number summer_temp summer_inten summer_dhd50
## 1          0.0    6.930      21.587       1.234     22.491

```

```

## 2      0.0  2.935    21.742     2.884    78.853
## 3      0.4  3.520    22.209     2.904    39.467
## 4      0.0  5.000    21.587     1.234    22.491
## 5      0.0  2.195    21.742     2.884    78.853
## 6      0.0  3.715    22.209     2.904    39.467
##   collapse
## 1  class_1
## 2  class_1
## 3  class_1
## 4  class_1
## 5  class_1
## 6  class_1

```

===== logistic binomial model

```

# ## Variance Inflation Factor (VIF) to detect multicollinearity!
# vif(model)

model <- glm(collapse ~ ., data = data[, c(6:10)], family = binomial)
summary(model)

```

```

##
## Call:
## glm(formula = collapse ~ ., family = binomial, data = data[, ,
##       c(6:10)])
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 16.81529   5.60867   3.00  0.00272 **
## number      0.76975   0.08720   8.83  < 2e-16 ***
## summer_temp -0.86245   0.26150  -3.30  0.00097 ***
## summer_inten  0.37538   0.20181   1.86  0.06288 .
## summer_dhd50  0.01417   0.00656   2.16  0.03083 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 789.34 on 569 degrees of freedom
## Residual deviance: 641.31 on 565 degrees of freedom
## AIC: 651.3
##
## Number of Fisher Scoring iterations: 5

```

```
vif(model)
```

```

##      number  summer_temp  summer_inten  summer_dhd50
##      1.0039      2.3329      3.2237      3.4615

```

```
pR2(model)
```

```
## fitting null model for pseudo-r2

##      11h     11hNull          G2    McFadden      r2ML
## -320.65568 -394.66923  148.02709     0.18753     0.22871
##      r2CU
##     0.30510
```

The model's interpretation of *summer_temp*'s role is negative. This pattern may suggest underlying interaction between *summer_temp* and *summer_inten* and urchin density *number*.

The relationship between *summer_temp/inten* and *collapse* may not be linear. If the model assumes a linear relationship, it could misrepresent this dynamic. E.g.,

- Low temperatures/intensities might have little effect on collapse
- Moderate increases/intensities might support growth
- High temperatures/intensities might cause collapse

However, All VIFs < 5 (even well below 10), multicollinearity is not severe.

McFadden pseudo-R² 0.19, which is reasonable explanatory power.

Model selection with interaction terms

```
glm_interaction <- glm(collapse ~ number + summer_temp*summer_inten + summer_dhd50,
                        family = binomial, data = data)
summary(glm_interaction)
```

```
##
## Call:
## glm(formula = collapse ~ number + summer_temp * summer_inten +
##       summer_dhd50, family = binomial, data = data)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -14.86353   10.93134  -1.36  0.17392
## number                      0.77963    0.08848   8.81  < 2e-16
## summer_temp                  0.59108    0.50329   1.17  0.24022
## summer_inten                19.96275    5.86189   3.41  0.00066
## summer_dhd50                 0.01712    0.00679   2.52  0.01173
## summer_temp:summer_inten   -0.89311    0.26696  -3.35  0.00082
##
## (Intercept)
## number                   ***
## summer_temp
## summer_inten            ***
## summer_dhd50             *
## summer_temp:summer_inten ***
##
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 789.34  on 569  degrees of freedom
## Residual deviance: 629.84  on 564  degrees of freedom
## AIC: 641.8
##
## Number of Fisher Scoring iterations: 5

```

```
vif(glm_interaction)
```

```

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

```

```

##             number          summer_temp
##           1.0111            8.4418
##      summer_inten        summer_dhd50
##           2652.7856          3.5407
## summer_temp:summer_inten
##           2862.5445

```

```
pR2(glm_interaction)
```

```
## fitting null model for pseudo-r2
```

```

##      1lh    1lhNull       G2   McFadden      r2ML
## -314.91926 -394.66923  159.49993    0.20207    0.24408
##      r2CU
##      0.32561

```

With the interaction of *summer_temp* and *summer_inten*:

- * Main effect of *summer_temp* is now positive but not significant (Estimate = 0.591, p = 0.24).
- * Main effect of *summer_inten* is very large and highly significant.
- * Interaction term *summer_temp:summer_inten* is negative and significant.

This means the effect of *summer_temp* on collapse now depends on the level of *summer_inten* –

- At low intensity, temperature increases collapse probability.
- At high intensity, the interaction dominates, and increasing temperature decreases collapse probability (conditional on intensity).

However, there is extremely high multicollinearity.

So potential interactions with other key variables – Urchin density?

```

glm_interaction1 <- glm(collapse ~ number*summer_temp +summer_inten + summer_dhd50,
                        family = binomial, data = data)
summary(glm_interaction1)

```

```

##
## Call:
##
```

```

## glm(formula = collapse ~ number * summer_temp + summer_inten +
##      summer_dhd50, family = binomial, data = data)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)            30.55151   7.48205   4.08  4.4e-05 ***
## number                 -8.72245   3.28382  -2.66   0.0079 **
## summer_temp             -1.48624   0.34578  -4.30  1.7e-05 ***
## summer_inten            0.36396   0.20823   1.75   0.0805 .
## summer_dhd50             0.01495   0.00674   2.22   0.0264 *
## number:summer_temp     0.43160   0.14991   2.88   0.0040 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 789.34 on 569 degrees of freedom
## Residual deviance: 633.27 on 564 degrees of freedom
## AIC: 645.3
##
## Number of Fisher Scoring iterations: 5

```

```
vif(glm_interaction1)
```

```

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

```

```

##           number       summer_temp       summer_inten
## 1512.9315          3.9997          3.4302
## summer_dhd50 number:summer_temp
##            3.5332         1503.3080

```

```
pR2(glm_interaction1)
```

```

## fitting null model for pseudo-r2

```

```

##      llh    llhNull        G2    McFadden      r2ML
## -316.63263 -394.66923  156.07319    0.19773    0.23953
##      r2CU
##    0.31953

```

There is also large multicollinearity between summer_temp and number. So remove the interaction term summer_temp:number.

```

glm_interaction2 <- glm(collapse ~ summer_temp + summer_inten*number + summer_dhd50,
                        family = binomial, data = data)
summary(glm_interaction2)

```

```

##
## Call:
## glm(formula = collapse ~ summer_temp + summer_inten * number +
## 
```

```

##      summer_dhd50, family = binomial, data = data)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           18.50936   5.68566   3.26  0.00113 **
## summer_temp          -0.91370   0.26413  -3.46  0.00054 ***
## summer_inten         -0.00409   0.25815  -0.02  0.98737
## number                0.34989   0.18104   1.93  0.05328 .
## summer_dhd50          0.01584   0.00680   2.33  0.01985 *
## summer_inten:number   0.26666   0.10792   2.47  0.01348 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 789.34 on 569 degrees of freedom
## Residual deviance: 635.16 on 564 degrees of freedom
## AIC: 647.2
##
## Number of Fisher Scoring iterations: 5

```

```
vif(glm_interaction2)
```

```

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

```

```

##      summer_temp        summer_inten       number
##            2.3900          5.2018          4.5157
##      summer_dhd50 summer_inten:number
##            3.6807          5.6427

```

```
pR2(glm_interaction2)
```

```
## fitting null model for pseudo-r2
```

```

##      llh    llhNull        G2    McFadden      r2ML
## -317.58210 -394.66923  154.17425     0.19532     0.23699
##      r2CU
##     0.31614

```

Large multicollinearity disappears, but ‘summer_temp’ shows negative relationship with kelp collapse, which is not ecological meaningful.

```

glm_interaction3 <- glm(collapse ~ number*summer_inten + summer_dhd50,
                        family = binomial, data = data)
summary(glm_interaction3)

```

```

##
## Call:
## glm(formula = collapse ~ number * summer_inten + summer_dhd50,
##      family = binomial, data = data)

```

```

## 
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.14387   0.34454  -3.32   0.0009 ***
## number                 0.41334   0.18035   2.29   0.0219 *
## summer_inten          0.16908   0.25371  -0.67   0.05052
## summer_dhd50          0.00750   0.00623   1.20   0.2288
## number:summer_inten  0.23859   0.10666   2.24   0.0253 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 789.34 on 569 degrees of freedom
## Residual deviance: 647.60 on 565 degrees of freedom
## AIC: 657.6
##
## Number of Fisher Scoring iterations: 5

```

```
vif(glm_interaction3)
```

```

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

```

```

##             number          summer_inten          summer_dhd50
##             4.4973            5.1929            3.2158
## number:summer_inten
##             5.6461

```

```
pR2(glm_interaction3)
```

```

## fitting null model for pseudo-r2

```

```

##      1lh     1lhNull        G2    McFadden      r2ML
## -323.80138 -394.66923  141.73570     0.17956     0.22015
##      r2CU
##     0.29368

```

Unsignificant term ‘summer_dhd50’.

best fitting model

```

glm_interaction4 <- glm(collapse ~ number*summer_inten,
                        family = binomial, data = data)
summary(glm_interaction4)

```

```

## 
## Call:
## glm(formula = collapse ~ number * summer_inten, family = binomial,
## 
```

```

##      data = data)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.2423    0.3346  -3.71   0.0002 ***
## number                 0.4247    0.1810   2.35   0.0190 *
## summer_inten          0.0412    0.1817   0.23   0.08207 *
## number:summer_inten  0.2307    0.1070   2.16   0.0311 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 789.34 on 569 degrees of freedom
## Residual deviance: 649.06 on 566 degrees of freedom
## AIC: 657.1
##
## Number of Fisher Scoring iterations: 5

```

```
vif(glm_interaction4)
```

```

## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif

```

```

##             number         summer_inten number:summer_inten
##             4.5268            2.6522            5.6223

```

```
pR2(glm_interaction4)
```

```

## fitting null model for pseudo-r2

```

```

##      llh     llhNull        G2    McFadden      r2ML
## -324.53017 -394.66923  140.27811     0.17772    0.21816
##      r2CU
##     0.29102

```

The interaction term ('number':'summer_inten') is significant in both cases ($p < 0.01$).

The final model has the lowest AIC value.

evaluate glm model performance

```

## 10-fold
train_control <- trainControl(method = "cv",
                                number = 10) # , classProbs = TRUE, summaryFunction = twoClassSummary, s

```

```

set.seed(123) # For reproducibility
glm_cv <- train(collapse ~ number*summer_inten, data = data,
                  method = "glm",
                  family = "binomial",
                  trControl = train_control) # ,metric = "ROC"    Use ROC for optimization

# Access accuracy and AUC
mean_accuracy <- glm_cv$results$Accuracy
mean_auc <- glm_cv$results$ROC

cat("Mean Accuracy: ", mean_accuracy, "\n")
cat("Mean AUC: ", mean_auc, "\n")

```

Create a new data frame for glm prediction -partical dependence

```

# Create prediction data
prediction_data <- expand.grid(
  number = seq(min(data$number), max(data$number), length.out = 100),
  summer_inten = seq(0, max(data$summer_inten), length.out = 100),# mean(data$summer_inten),
  summer_dhd50 = mean(data$summer_dhd50, na.rm = TRUE)
)

prediction_data$predicted_prob <- predict(glm_interaction4, #glm_interaction
                                            newdata = prediction_data,
                                            type = "response")
# recover number into numerical for colorbar
prediction_data$number <- as.numeric(prediction_data$number)

=====3d plot: Create a grid for all predictors, including latitude=====

scico_colors <- scico(n = 5, palette = "navia", begin = 0.12, direction = 1)
# barplot(rep(1, 5), col = scico_colors, space = 0, border = NA)

# Perform interpolation on the data to create a smooth grid
interp_data <- with(prediction_data,
                      interp(x = summer_inten, y = predicted_prob, z = number,
                             xo = seq(min(summer_inten),max(summer_inten), length.out = 500),
                             yo = seq(min(predicted_prob), max(predicted_prob), length.out = 500)))

# Convert interpolation data into a dataframe for ggplot
interp_df <- expand.grid(summer_inten = interp_data$x, predicted_prob = interp_data$y)
interp_df$z <- as.vector(interp_data$z)
interp_df <- interp_df[is.finite(interp_df$z), ]
ubreaks <- as.numeric(c(0:11))
my_breaks <- c(0, 0.5, 1, 1.5, 2, 2.5, 3.2, 4.5, 6, 7.5, 9, 10.4)

```

Plot contour

```

p_contour = ggplot(interp_df, aes(summer_inten, predicted_prob)) +
  metR::geom_contour_fill(aes(z = z), breaks = my_breaks) +
  geom_contour(aes(z = z), breaks = my_breaks, colour = "black", linewidth = 0.8) +
  metR::geom_text_contour(
    aes(z = z, label = sprintf("%.1f", ..level..)),
    breaks = my_breaks,
    label.placer = label_placer_fraction(0.5),
    stroke = 0.15, size = 3.5, vjust = 0.1, hjust = 0.2) +
  # metR::geom_contour_fill(aes(z = z), bins = 50) +
  # geom_contour(aes(z = z), colour = 'black', bins = 12.5, linewidth = 0.8) +
  # metR::geom_text_contour(aes(z = z, label = sprintf("%.1f", ..level..)),
  #                         bins = 12.5, label.placer = label_placer_fraction(0.5),
  #                         stroke = 0.15, size = 3.5, vjust = 0.1, hjust = 0.2) +
  # Define the gradient colors
  scale_fill_gradientn(colors = scico_colors, #c("#3d73c2", "#89718e", "#c96d69", "#c0503f")
                        values = c(0, 0.12, 0.3, 0.5, 0.7, 1),      #seq(from = 0,to = 1,length.out = 6)
                        limits = c(0, 10.3),
                        name = expression(paste("Urchin \nDensity/", m^2))) +
  labs(x = expression('Summer Maximum intensity(~degree~'C)'), y = "Probability of Kelp Colonization"),
  scale_y_continuous(expand = c(0, 0)) +
  scale_x_continuous(expand = c(0, 0)) +
  theme_format +
  theme(legend.position = "right", legend.key.height = unit(1.5, "cm"))
p_contour

```