

Væñ æ) Á àãáÁã \ Á æ] Á[åã *

0E Áã Áãáã

0E0E1

åããÁ [~ ! & ^ •

- 0E1 0E0E年公民科學數據eBird資料庫
- 行政院農業委員會特有生物研究保育中心所提供的台灣陸域環境因子

contents

- ebird sighters spatial distribution
- mapping for elevation
- mapping by poultry farm distribution
- Taiwan ebird data screening
- bird species total counts distribution
- bird species presence counts distribution
- wild bird distribution spatial probability predicted by ZIP/ZINB modeling
- ZIP(zero-inflated poisson model)
- ZINB(zero-inflated negative binomial model)
- ZIP-Logistic-Kendall
- propensity score matching (PSM) and Bootstrapping
- wild bird risk mapping
- t_sne_cluster

ebird sighters spatial distribution

- 所需資料:
 - 臺灣陸域環境因子資料
 - 候鳥及水鳥觀測者數量資料

資料讀取

```
tw.map = readOGR('C:/Users/user/Desktop/R/data/G3km_3826_environment_dataset/G3km_3826_environmental_dataset.shp', layer = 'G3km_3826_environmental_dataset') #臺灣地圖網格shp檔
```

```
## OGR data source with driver: ESRI Shapefile
```

```
## Source: "C:\Users\user\Desktop\R\data\G3km_3826_environment_dataset\G3km_3826_environmental_dataset.shp", layer: "G3km_3826_environmental_dataset"
```

```
## with 4762 features
```

```
## It has 112 fields
```

```
## Integer64 fields read as strings: id1 FID_G3km_3 id_2 ELEmax ELEmin ELERA PROAD
```

```
observer <- read.csv("C:/Users/user/Desktop/R/data/觀測者.csv") #在各網格中觀測者有無觀測
```

資料整理

```
taiwan_map = fortify(tw.map)
```

```
names(taiwan_map)[1] <- "longitude" #欄位命名
```

```
names(taiwan_map)[2] <- "latitude" #欄位命名
```

```
observer$id = 1:nrow(observer) - 1
```

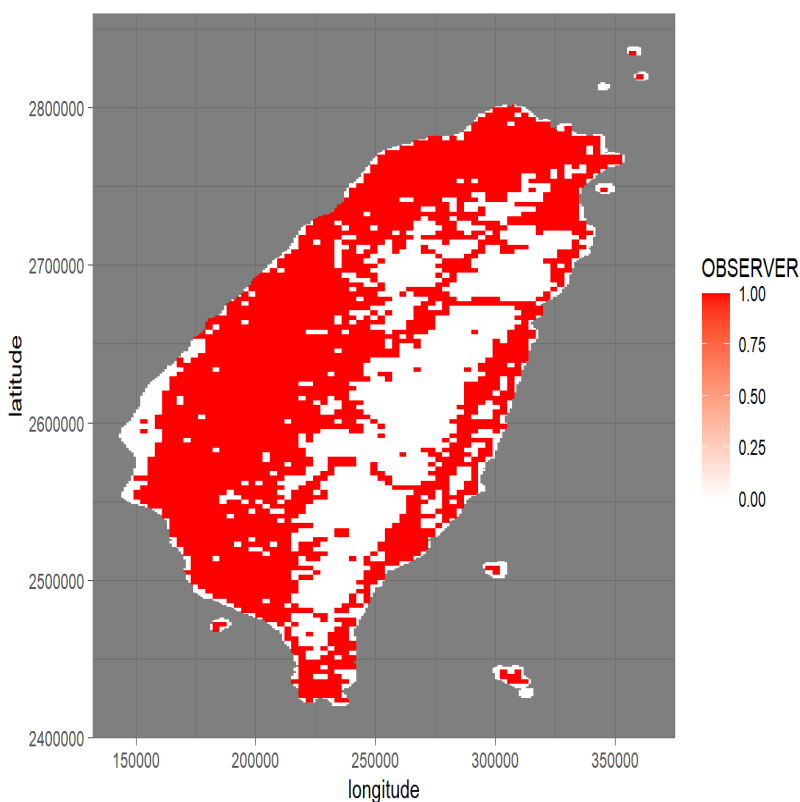
```
tw1 = merge(taiwan_map,observer, by = "id",all.x=T) #地圖與資料合併
```

畫圖

```
Birdmap1 = ggplot() + geom_polygon(data = tw1, aes(x = longitude ,y = latitude,group = group, fill = OBSERVER))
```

```
Birdmap1 + scale_fill_gradient(low = "white", high = "red") + theme_dark() + ggtitle('觀測者分布圖')
```

觀測者分布圖



mapping for elevation

- 所需資料:
 - 臺灣陸域環境因子資料

```
library(dplyr) ##更有效率的資料處理
library(rgdal) ##基本地圖資料處理
library(ggplot2) ##視覺化分析
library(caTools) ##資料集分割函式
library(glmnet) ##廣義線性模型
library(caret) ##對數據進行預處理包括特徵篩選和缺失值填充等，還能進行交叉驗證和各種模型的grid調參
```

```
# 資料讀取
tw.map = st_read("E:/G3km_3826_environment_dataset/G3km_3826_environmental_dataset.shp")
# 開始資料挑選
ele <- as.data.frame(matrix(nrow = 4762, ncol = 113)) #將資料型態轉換
ele[is.na(ele)] = 0 #處理遺漏值
ELE <- filter(ele , ELE >0) #挑選海拔高度高於0(平面)
# 匯出檔案
write.csv(ELE, 'C:/Users/user/Desktop/ELE.csv')
```

再運用QGIS畫圖

mapping by poultry farm distribution

- 所需資料:
 - 臺灣陸域環境因子資料
 - 禽場資料

```
library(rgdal) ##基本地圖資料處理
library(ggplot2) ##視覺化分析
library(dplyr) ##更有效率的資料處理
```

```
# 資料讀取
```

```
tw.map = readOGR('C:/Users/user/Desktop/R/data/G3km_3826_environment_dataset/G3km_3826_environmental_dataset.shp', layer = 'G3km_3826_environmental_dataset')
```

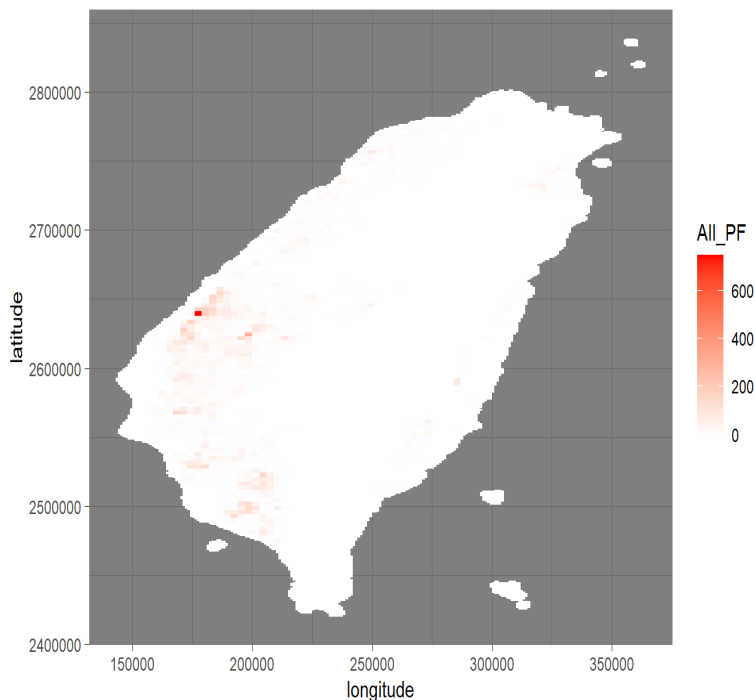
```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\user\Desktop\R\data\G3km_3826_environment_dataset\G3km_3826_environmental_dataset.shp", layer: "G3km_3826_environmental_dataset"
## with 4762 features
## It has 112 fields
## Integer64 fields read as strings: id1 FID_G3km_3 id_2 ELEmax ELEmin ELERA PROAD
```

```
pf <- read.csv("C:/Users/user/Desktop/R/data2/禽場.csv")
```

```
# 資料整理
```

```
taiwan_map = fortify(tw.map)
names(taiwan_map)[1] <- "longitude"
names(taiwan_map)[2] <- "latitude"
pf$id = 1:nrow(pf) - 1
tw1 = merge(taiwan_map, pf, by = "id", all.x=T)
# 畫圖
Birdmap1 = ggplot() + geom_polygon(data = tw1, aes(x = longitude ,y = latitude, group = group, fill = All_PF))
Birdmap1 + scale_fill_gradient(low = "white", high = "red") + theme_dark() + ggtitle('禽場數分布')
```

禽場數分布



Taiwan ebird data screening

- 所需資料:
 - 公民科學數據eBird資料庫
- 資料整理限制條件:
 - 調查移動距離小於2km
 - 觀測面積小於100ha且持續時間不超過240min

```
library(tidyr) ##表格長寬轉換
library(data.table) ##利用該package · 在讀取大型資料的速度會比單純用資料框讀取快很多
library(dplyr) ##更有效率的資料處理
```

```
# 資料讀取
eBird.all = fread('C:/Users/user/Desktop/R/data/ebd_TW_201501_202006_re1Jul-2020.txt', encoding
= "UTF-8", sep = "\t") ##fread指令---library(data.table)

# 資料整理
eBird.all$`DURATION MINUTES`[is.na(eBird.all$`DURATION MINUTES`)] = 0
eBird.all$`EFFORT DISTANCE KM`[is.na(eBird.all$`EFFORT DISTANCE KM`)] = 0
eBird.all$`EFFORT AREA HA`[is.na(eBird.all$`EFFORT AREA HA`)] = 0
eBird.all = filter(eBird.all, `DURATION MINUTES` < 240 & `EFFORT DISTANCE KM` < 2 & `EFFORT AREA
HA` < 100)
eBird.select <- eBird.all %>% .[, c(6, 9, 26, 27, 28, 30, 32, 33, 35, 36, 37, 38)]

# 刪除沒有觀察到鳥的資料
eBird.select <- eBird.select[-which(eBird.select$`OBSERVATION COUNT` == "X"),]
eBird.select$year <- format(as.Date(eBird.select$`OBSERVATION DATE`), "%Y")

# 篩選出特定的鳥
Pmbird = filter(eBird.select, `SCIENTIFIC NAME` == 'Anas acuta') ##以鳥的學名

# 資料匯出成csv檔
write.csv(Pmbird, 'C:/Users/user/Desktop/尖尾鴨.csv') ##該檔案年為2015-2020年
```

bird species total counts distribution

- 所需資料:
 - 臺灣環境陸域因子資料
 - 留鳥及候鳥個數資料

```
library(rgdal)
library(ggplot2)
library(dplyr)
```

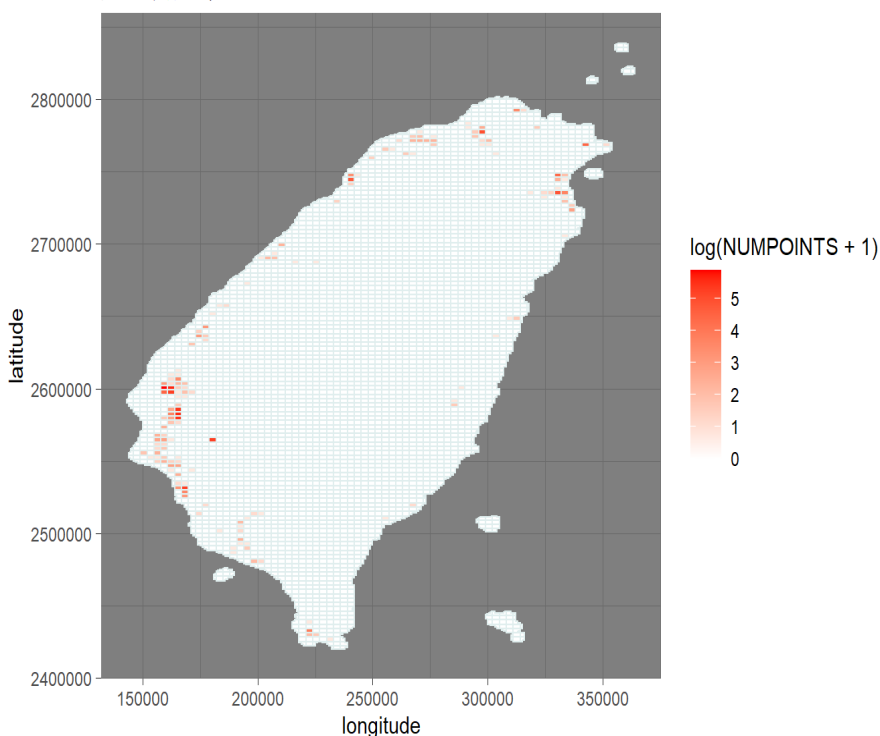
```
# 資料讀取(臺灣環境陸域因子資料)
```

```
tw.map = readOGR('C:/Users/user/Desktop/R/data/G3km_3826_environment_dataset/G3km_3826_environmental_dataset.shp', layer = 'G3km_3826_environmental_dataset')
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\user\Desktop\R\data\G3km_3826_environment_dataset\G3km_3826_environmental_dataset.shp", layer: "G3km_3826_environmental_dataset"
## with 4762 features
## It has 112 fields
## Integer64 fields read as strings: id1 FID_G3km_3 id_2 ELEmax ELEmin ELERA PROAD
```

```
taiwan_map = fortify(tw.map)
names(taiwan_map)[1] <- "longitude"
names(taiwan_map)[2] <- "latitude"
Birdcount1 = read.csv('C:/Users/user/Desktop/R/data/69水鳥個數.csv')[, c(1,14)] ##尖尾鴨數量
Birdcount1$id = 1:nrow(Birdcount1) - 1
tw1 = merge(taiwan_map, Birdcount1, by = "id", all.x=T)
names(tw1)[8] = "NUMPOINTS"
# 畫圖
Birdmap1 = ggplot() + geom_polygon(data = tw1, aes(x = longitude ,y = latitude, group = group, fill = log(NUMPOINTS + 1)), col = 'azure2')
Birdmap1 + scale_fill_gradient(low = "white", high = "red") + theme_dark() + ggtitle('尖尾鴨分布')
```

尖尾鴨分布



bird species presence counts distribution

- 所需資料:
 - 臺灣環境陸域因子資料
 - 該鳥類近5年的次數(2015-2020年)

```
library(rgdal)
library(ggplot2)
library(dplyr)
```

```
# 資料讀取
```

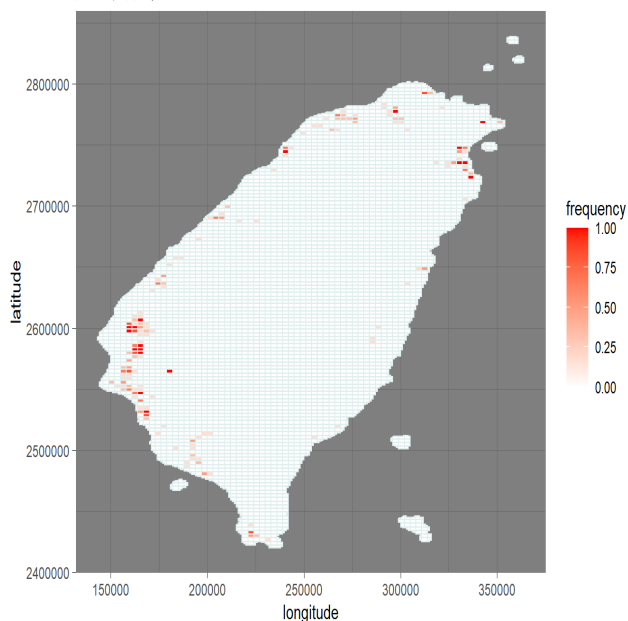
```
tw.map = readOGR('C:/Users/user/Desktop/R/data/G3km_3826_environment_dataset/G3km_3826_environmental_dataset.shp', layer = 'G3km_3826_environmental_dataset')
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\user\Desktop\R\data\G3km_3826_environment_dataset\G3km_3826_environmental_dataset.shp", layer: "G3km_3826_environmental_dataset"
## with 4762 features
## It has 112 fields
## Integer64 fields read as strings: id1 FID_G3km_3 id_2 ELEmax ELEmin ELERA PROAD
```

```
birdat = read.csv('E:/尖尾鴨/2015-2020.csv') #尖尾鴨近5年的次數
# 資料整理
taiwan_map = fortify(tw.map)
names(taiwan_map)[1] <- "longitude" #欄位命名
names(taiwan_map)[2] <- "latitude" #欄位命名
birdat <- birdat[,-1] #將資料內第1列的所有行值刪除
birdat[birdat>0]=1 #該資料內大於0的數值視為1
birdat$total <- birdat$X2015+birdat$X2016+birdat$X2017+birdat$X2018+birdat$X2019+birdat$X2020 #5年次數相加
frequency = birdat$total/ 6 #算出頻率

dat = data.frame(1:4762 - 1, frequency) #資料存入
names(dat) = c('id', 'frequency') #欄位命名
tw = merge(taiwan_map, dat, by = "id", all.x = T) #資料合併
# 畫圖
Birdmap = ggplot() + geom_polygon(data = tw, aes(x = longitude, y = latitude, group = group, fill=frequency), col='azure2')
Birdmap1 + scale_fill_gradient(low = "white", high = "red") + theme_dark() + ggtitle('尖尾鴨分布')
```

尖尾鴨頻率圖



wild bird distribution spatial probability predicted by ZIP/ZINB modeling

```
library(dplyr) #更有效率的資料處理
library(rgdal) #基本地圖資料處理
library(ggplot2) #視覺化分析
library(caTools) #資料集分割函式
library(glmnet) #廣義線性模型
library(caret) #對數據進行預處理包括特徵篩選和缺失值填充等·還能進行交叉驗證和各種模型的grid調參數
```

```
# 資料讀取
tw.map = readOGR('C:/Users/user/Desktop/R/CAR/G3km_3826_environment_dataset/G3km_3826_environmental_dataset.shp')
norbird = read.csv('C:/Users/user/Desktop/R/data/69水鳥個數.csv')
plist = read.csv('C:/Users/user/Desktop/R/data/水鳥p-list.csv')

# 資料整理
taiwan_map = fortify(tw.map) #將讀進來的shp檔內容轉成向量
names(taiwan_map)[1] <- "longitude"
names(taiwan_map)[2] <- "latitude"
ter = tw.map@data[, 9:110]
ter[ter < 0 | is.na(ter)] = 0 #ter裡有小於0或ter內有遺漏值全部改成0的數值
ter$ELEmax = as.numeric(ter$ELEmax) #取得ter表中ELEmax(最高海拔) ##as.numeric資料型別轉換-轉成數值
ter$ELEmin = as.numeric(ter$ELEmin) #取得ter表中ELEmin(最低海拔) ##as.numeric資料型別轉換-轉成數值
ter$ELERA = as.numeric(ter$ELERA) #取得ter表中ELERA(海拔跨幅) ##as.numeric資料型別轉換-轉成數值
plist[is.na(plist)] = '' #有遺漏值(na)的地方讓他為空

birdat = data.frame(plist$鳥名, plist$X13.尖尾鴨) #data.frame創建新的資料框 ##擷取plist資料裡的鳥名那一整行及尖尾鴨那一整行
names(birdat) = c('鳥名', '該鳥種') #命名birdat資料框內的欄位名稱
birdat = birdat[-which(birdat$該鳥種 == ''), ] #"- "意思為除去該後面指令範圍條件 ##該行條件意思為除去該鳥種這列沒有任何文字數字的資料
birdat$該鳥種 = as.numeric(as.character(birdat$該鳥種)) #由字串/元轉為數值 ##有可能是factor
birdat$該鳥種[is.na(birdat$該鳥種)] = 0 ##處理遺漏值

ybdat = data.frame(norbird$X13.尖尾鴨) #data.frame創建新的資料框·且將欄位名稱命名為尖尾鴨
names(ybdat) = 'Y' #行的名稱取為Y
for (k in 1:nrow(ybdat)){ #判斷數值
  if(ybdat[k, 1] > 0){ #k為每執行一迴圈增加1(行數)·1為固定列數(統一都取第1列)
    ybdat[k, 1] = 1 #大於0·自動將數值判斷為1
  }
}

for(i in 1:nrow(birdat)){ #nrow為行·意思為以行為主要判斷
  depdat = norbird[, which(names(norbird) == birdat[i, 1])]
  for (j in 1:length(depdat)) {
    if(depdat[j] > 0){
      depdat[j] = 1
    }
  }
  ybdat = cbind.data.frame(ybdat, depdat) #cbind為左右合併
  names(ybdat)[i + 1] = paste('X', i, sep = '') #欄位命名
}
```



```

# 台灣網格判斷
num = c(-141:-139, -1, 1, 139:141) #c()為創建一個陣列/向量
y_nai = matrix(0, 4762, 8) #matrix()函數→matrix(可輸入一個值或一組值,矩陣列數,矩陣行/欄數)
for (s in 1:nrow(ybdat)) {
  testid = tw.map$id - tw.map$id[s]
  for (t in 1:length(num)) {
    if(length(ybdat[testid == num[t], 1]) == 1){
      y_nai[s, t] = ybdat[testid == num[t], 1]
    }else y_nai[s, t] = 0
  }
}
ybdat$naY = apply(y_nai, 1, sum)

#條件自相關模型(Conditional Autoregressive Model,CAR)

dat = cbind.data.frame(norbird$id, ybdat, ter) #合併資料框(左右合併)
names(dat)[1] = 'id'
obs = read.csv('C:/Users/user/Desktop/R/data/水鳥觀測者.csv')[1:4762, ] #將檔案匯入·並且取資料第1
行-4762行的所有列數值(存入)
birdobs = filter(obs, OBSERVER == 1) #filter函數:可以選要分析的觀察值·針對列作子集→filter(資料名稱,
篩選條件) ##篩選obs資料·並取有觀測者為1的資料
alldat = merge(birdobs, dat, by = 'id', all.x = T)[, -c(1, 2)]

split = sample.split(alldat$Y, SplitRatio = 0.8) ##測試集與訓練集
train_dat = subset(alldat, split == T)
test_data = subset(alldat, split == F)

trainControl = trainControl(method = 'repeatedcv', number = 5, repeats = 3, search = 'random', v
erboseIter = TRUE) #trainControl為挑選 $\alpha$ (alpha)· $\lambda$ (lambda)
elastic = train(as.factor(Y) ~., data = train_dat, method = 'glmnet', family = 'binomial', trCon
trol = trainControl, tuneLength = 50)

coef = 0
for (i in 1:length(coef(elastic$finalModel, elastic$bestTune$lambda))) {
  coef[i] = coef(elastic$finalModel, elastic$bestTune$lambda)[i]
}

dat = dat[, -1]
dat$PROAD = as.numeric(dat$PROAD)
BetaX = 0
for (m in 1:nrow(dat)) {
  x_ij = 0
  for (n in 2:ncol(dat)) {
    x_ij[n - 1] = coef[n] * dat[m, n]
  }
  BetaX[m] = coef[1] + sum(x_ij)
}
probability = exp(BetaX) / (1 + exp(BetaX))

#匯出CSV檔
write.csv(probability, 'C:/Users/user/Desktop/尖尾鴨PSM.csv')

#畫圖
precount = cbind.data.frame(1:nrow(ybdat) - 1, probability)
names(precount) = c('id', 'probability')
pretw = merge(taiwan_map, precount, by = "id", all.x = T)
preBirdmap = ggplot() + geom_polygon(data = pretw, aes(x = longitude,y = latitude, group = grou
p, fill = probability), col = 'azure2')
preBirdmap = preBirdmap + scale_fill_gradient(low = "white", high = "red") + theme_dark()
preBirdmap

```

ZIP(zero-inflated poisson model)

```
library(pscl)
library(dplyr)
library(rgdal)
library(ggplot2)
library(caTools)
library(glmnet)
library(caret)
```

```
# 讀取資料
tw.map = readOGR('C:/Users/user/Desktop/R/data/G3km_3826_environment_dataset/G3km_3826_environmental_dataset.shp')
taiwan_map = fortify(tw.map)
ter = tw.map@data[, 9:110]
ter[ter < 0 | is.na(ter)] = 0
ter$ELEmax = as.numeric(ter$ELEmax)
ter$ELEmin = as.numeric(ter$ELEmin)
ter$ELERA = as.numeric(ter$ELERA)

norbird = read.csv('C:/Users/user/Desktop/R/data/69水鳥個數.csv')
plist = read.csv('C:/Users/user/Desktop/R/data/水鳥p-list.csv')
plist[is.na(plist)] = ''

birdat = data.frame(plist$鳥名, plist$X13.尖尾鴨)
names(birdat) = c('鳥名', '該鳥種')
birdat = birdat[-which(birdat$該鳥種 == ''), ]
birdat$該鳥種 = as.numeric(as.character(birdat$該鳥種))
birdat$該鳥種[is.na(birdat$該鳥種)] = 0
dat = birdat

ybdat = data.frame(norbird$X13.尖尾鴨)
names(ybdat) = 'Y'
for (k in 1:nrow(ybdat)) {
  if(ybdat[k, 1] > 0){
    }
}
for(i in 1:nrow(dat)){
  depdat = norbird[, which(names(norbird) == dat[i, 1])]
  for (j in 1:length(depdat)) {
    if(depdat[j] > 0){
      depdat[j] = 1
    }
  }
  ybdat = cbind.data.frame(ybdat, depdat)
  names(ybdat)[i + 1] = paste('X', i, sep = '')
}
num = c(-141:-139, -1, 1, 139:141)
y_nai = matrix(0, 4762, 8)
for (s in 1:nrow(ybdat)) {
  testid = tw.map$id - tw.map$id[s]
  for (t in 1:length(num)) {
    if(length(ybdat[testid == num[t], 1]) == 1){
      y_nai[s, t] = ybdat[testid == num[t], 1]
    }else y_nai[s, t] = 0
  }
}
}
```

```

ybdatt$naY = apply(y_nai, 1, sum)

dat = cbind.data.frame(norbird$id, ybdatt, ter)
dat$num = 1:4762
names(dat)[1] = 'id'
birdobs = read.csv('C:/Users/user/Desktop/R/data/水鳥觀測者.csv')[1:4762, ]
alldat = merge(birdobs, dat, by = 'id', all.x = T)[, -1]
alldat = alldat %>% arrange(alldat$num)
split = sample.split(alldat$Y, SplitRatio = 0.8)
train_dat = subset(alldat, split == T)
test_data = subset(alldat, split == F)

null = glm(Y ~ 1, data = train_dat, family = 'poisson')
full = glm(Y ~ .- OBSERVER - num, data = train_dat, family = 'poisson')
forward.glm = step(null, scope = list(lower = null, upper = full), direction = 'forward')
Z = paste('Y ~', as.character(forward.glm$formula[3]), '+ naY | OBSERVER')

model = zeroinfl(as.formula(Z), data = train_dat, dist = 'poisson')
summary(model)

result = predict(model, alldat, type = 'prob')
zero = result[, 1]
true_z = result[birdobs$OBSERVER == 0, 1]
precount = cbind.data.frame(1:nrow(ybdatt) - 1, 1 - zero)
names(precount) = c('id', 'probability')
pretw = merge(taiwan_map, precount, by = "id", all.x = T)
preBirdmap = ggplot() + geom_polygon(data = pretw, aes(x = long,y = lat, group = group, fill = p
robability), col = 'azure2')
preBirdmap = preBirdmap + scale_fill_gradient(low = "white", high = "red") + theme_dark() + ggtitle('尖尾鴨(Anas acuta)ZIP')
preBirdmap

write.csv(1 - zero, 'C:/Users/user/Desktop/尖尾鴨(ZIP).csv')

```

ZINB(zero-inflated negative binomial model)

```
library(pscl)
library(dplyr)
library(rgdal)
library(ggplot2)
library(caTools)
library(MASS)
library(caret)
```

```
tw.map = readOGR('C:/Users/user/Desktop/R/data/G3km_3826_environment_dataset/G3km_3826_environmental_dataset.shp')
taiwan_map = fortify(tw.map)
names(taiwan_map)[1] <- "longitude"
names(taiwan_map)[2] <- "latitude"
ter = tw.map@data[, 9:110]
ter[ter < 0 | is.na(ter)] = 0
ter$ELEmax = as.numeric(ter$ELEmax)
ter$ELEmin = as.numeric(ter$ELEmin)
ter$ELERA = as.numeric(ter$ELERA)
```

```
norbird = read.csv('C:/Users/user/Desktop/R/data/39留鳥個數.csv')
plist = read.csv('C:/Users/user/Desktop/R/data/留鳥p-list.csv')
plist[is.na(plist)] = ''
```

```
birdat = data.frame(plist$鳥名, plist$X22.烏頭翁)
names(birdat) = c('鳥名', '該鳥種')
birdat = birdat[-which(birdat$該鳥種 == ''), ]
birdat$該鳥種 = as.numeric(as.character(birdat$該鳥種))
birdat$該鳥種[is.na(birdat$該鳥種)] = 0
dat = birdat
```

```
ybdat = data.frame(norbird$X22.烏頭翁)
names(ybdat) = 'Y'
for (k in 1:nrow(ybdat)) {
  if(ybdat[k, 1] > 0){
    ybdat[k, 1] = 1
  }
}
for(i in 1:nrow(dat)){
  depdat = norbird[, which(names(norbird) == dat[i, 1])]
  for (j in 1:length(depdat)) {
    if(depdat[j] > 0){
      depdat[j] = 1
    }
  }
  ybdat = cbind.data.frame(ybdat, depdat)
  names(ybdat)[i + 1] = paste('X', i, sep = '')
}
num = c(-141:-139, -1, 1, 139:141)
y_nai = matrix(0, 4762, 8)
for (s in 1:nrow(ybdat)) {
  testid = tw.map$id - tw.map$id[s]
  for (t in 1:length(num)) {
    if(length(ybdat[testid == num[t], 1]) == 1){
      y_nai[s, t] = ybdat[testid == num[t], 1]
    }else y_nai[s, t] = 0
  }
}
}
```

```

ybdat$naY = apply(y_nai, 1, sum)

dat = cbind.data.frame(norbird$id, ybdat, ter)
dat$num = 1:4762
names(dat)[1] = 'id'
birdobs = read.csv('C:/Users/user/Desktop/R/data/留鳥觀測者.csv')[1:4762, ]
alldat = merge(birdobs, dat, by = 'id', all.x = T)[, -1]
alldat = alldat %>% arrange(alldat$num)

split = sample.split(alldat$Y, SplitRatio = 0.8)
train_dat = subset(alldat, split == T)
test_data = subset(alldat, split == F)

null = glm.nb(Y ~ 1, data = train_dat)
full = glm.nb(Y ~ .- OBSERVER - num, data = train_dat)
forward.glm = step(null, scope = list(lower = null, upper = full), direction = 'forward')
Z = paste('Y ~', as.character(forward.glm$call$formula[3]), '| X5 + X67 + X12 + X57')

model = zeroinfl(as.formula(Z), data = train_dat, dist = 'negbin')
summary(model)

result = predict(model, alldat, type = 'prob')
zero = result[, 1]
true_z = result[birdobs$OBSERVER == 0, 1]
precount = cbind.data.frame(1:nrow(ybdat) - 1, 1 - zero)
names(precount) = c('id', 'probability')
pretw = merge(taiwan_map, precount, by = "id", all.x = T)
preBirdmap = ggplot() + geom_polygon(data = pretw, aes(x = longitude, y = latitude, group = group, fill = probability), col = 'azure2')
preBirdmap = preBirdmap + scale_fill_gradient(low = "white", high = "red") + theme_dark()
preBirdmap

write.csv(1 - zero, 'C:/Users/user/Desktop/ZINB/烏頭翁ZINB.csv')

```

ZIP-Logistic-Kendall

```
library(pscl) #zero-inflated Poisson model
```

```
# 讀取資料
poultry <- read.csv('C:/Users/user/Desktop/R/data2/禽場.csv') #禽場與爆發禽場資料

# 傾向分數資料叫進來
csvpath = 'C:/Users/user/Desktop/R/data2/PSM(鳥)/水鳥/' #目前存放要處理csv檔案的路徑
csvfilesn = list.files(path = csvpath) #存list.files()函數的結果 · list.files()會將每個檔案名稱處理成字串向量
tmpprt = function(rtcsv) { #自訂函數將csv檔讀進來
  read.csv(rtcsv)
}
data = lapply(paste(csvpath, csvfilesn, sep = ''), tmpprt)
#lapply()函數中的paste()會將csvpath(存有存放工作目錄路徑字串)與csvfilesn(存有檔案名稱的字串)組合成絕對路徑 · 使tmpprt()自訂函數可以大量匯入csv檔案
```

Zero-inflated Poisson regression(方法一)

```
z <- matrix(NA , length(csvfilesn) , 3) #matrix為矩陣→matrix(一個/組數值,矩陣列數,矩陣行/欄數)
for (i in 1:length(csvfilesn)) { #迴圈從1到csvfilesn裡的數量

  score <- data[[i]][2] 各鳥種傾向分數(將原先就有先做好的各鳥種傾向分數資料讀進來)、data資料第i列第2行
  dat <- data.frame(poultry$爆發禽場,score) # 爆發禽場跟傾向分數資料
  names(dat)[1] <- 'Y' #命名dat資料裡欄位名稱為Y
  zip <- zeroinfl(Y ~ x, data = dat , dist = "poisson") #爆發禽場對傾向分數做zip迴歸 #zeroinfl函數做零膨脹迴歸 ##dist用甚麼方法去配適
  b <- strsplit(csvfilesn[[i]] , "鳥")[[1]][1] # 字串分割 #strsplit用來分割字串 · 以"鳥"結尾做為字串分割的符號
  z[i,1] <- b #第一行 鳥名
  z[i,2] <- round(zip$coefficients$zero[2],3) #第二行 zip係數 ##論文p.14
  z[i,3] <- round(summary(zip)$coefficients$zero[8],3) #第三行 p-value ##論文p.14
}
write.csv(z , "C:/Users/user/Desktop/R/新資料夾/zip.csv") #檔案匯出
```

logistic regression ##邏輯斯迴歸(方法二)

```
for (i in 1:nrow(poultry)) { #禽場數量大於1的改成1
  if(poultry[i,4] > 1){
    poultry[i,4] <- 1
  }
}

l <- matrix(NA , length(csvfilesn) , 3) #matrix為矩陣→matrix(一個/組數值,矩陣列數,矩陣行/欄數)
for (i in 1:length(csvfilesn)) { #各鳥種傾向分數(將原先就有先做好的各鳥種傾向分數資料讀進來)
  score <- data[[i]][2] #data資料第i列第2行
  dat <- data.frame(poultry$爆發禽場,score) #爆發禽場跟傾向分數資料
  names(dat)[1] <- "Y" #命名dat資料裡欄位名稱為Y
  logr <- glm(Y ~ x, data = dat , family = "binomial") #爆發禽場對傾向分數做邏輯斯迴歸 #family是指
  用甚麼方法去配適(參數) #binomial-邏輯斯迴歸參數
  b <- strsplit(csvfilesn[[i]] , "鳥")[[1]][1] #字串分割 #strsplit用來分割字串·以"鳥"結尾做為字串
  分割的符號
  l[i,1] <- b #第一行 鳥名
  l[i,2] <- round(logr$coefficients[2],3) #論文p.14上方
  l[i,3] <- round(summary(logr)$coefficient[8],3) #論文p.14上方
}
write.csv(l , "C:/Users/user/Desktop/R/新資料夾/logistic.csv") #檔案匯出
```

kendall's tau(方法三)

```
k <- matrix(NA , length(csvfilesn) , 2) #matrix為矩陣→matrix(一個/組數值,矩陣列數,矩陣行/欄數)
for (i in 1:length(csvfilesn)) { #各鳥種傾向分數(將原先就有先做好的各鳥種傾向分數資料讀進來)
  score <- data[[i]][2] #data資料第i列第2行
  dat <- data.frame(poultry$爆發禽場,score) #爆發禽場跟傾向分數資料
  names(dat)[1] <- "Y" #命名dat資料裡欄位名稱為Y
  ken <- cor.test(dat$Y, dat$x , method = "kendall") #計算kendall相關係數
  b <- strsplit(csvfilesn[[i]] , "鳥")[[1]][1] #字串分割 #strsplit用來分割字串·以"鳥"結尾做為字串
  分割的符號
  k[i,1] <- b #第一行 鳥名
  k[i,2] <- round(ken$estimate,3) #round小數點 #取到小數點第3位
}
write.csv(k , "C:/Users/user/Desktop/R/新資料夾/kendall's tau.csv") #檔案匯出
```

propensity score matching (PSM) and Bootstrapping

- 所需要的資料:
 - 臺灣陸域環境因子資料
 - 臺灣禽場資料
 - 候鳥或水鳥PSM資料
 - 候鳥或水鳥個數資料

```
library(dplyr) #用於filter
library(ggplot2)
```

```
# 讀取資料
csvpath = 'C:/Users/user/Desktop/R/data2/PSM(鳥+環境)/水鳥/'
csvfilesn = list.files(path = csvpath)
tmpprt = function(rtcsv){
  read.csv(rtcsv)
}
data = lapply(paste(csvpath, csvfilesn, sep = ''), tmpprt)

outbreak <- read.csv('C:/Users/user/Desktop/R/data2/禽場.csv') #禽場跟爆發禽場資料

chIQ <- c()
list.f <- c()
boots <- c()
inter.val <- c()
#for (q in 1:length(csvfilesn)) {
  score <- data[[5]][2] #選取PSM資料夾裡的鳥欄位
  #bird <- read.csv('C:/Users/user/Desktop/R/data2/36留鳥個數.csv')[5] # 讀留鳥檔案、[]為資料欄位
  bird <- read.csv('C:/Users//user/Desktop/R/data2/68水鳥個數.csv')[5] # 讀水鳥檔案、[]為資料欄位
  names(bird) <- "NUMPOINTS"
  outbreak$score <- score$x #將傾向分數合併到爆發的禽場資料
  outbreak$num <- bird$NUMPOINTS #鳥的個數
  for (i in 1:nrow(outbreak)) { #爆發禽場與鳥的個數大於0都會設定為1
    if(outbreak[i, 6] > 0){
      outbreak[i, 6] = 1
    }
    if(outbreak[i, 8] > 0){
      outbreak[i, 8] = 1
    }
  }
}
outbreak_1 <- filter(outbreak 爆發禽場 == 1) ##將爆發禽場為1的挑出來
outbreak_0 <- outbreak[-which(outbreak$All_PF == 0),] #將沒有禽場的部分去除
outbreak_0 <- filter(outbreak_0 爆發禽場 == 0)
diff_v <- c()
diff1 <- c()
a <- c()
b <- c()
c <- c()
d <- c()
for (j in 1:nrow(outbreak_1)) {
  for (k in 1:nrow(outbreak_0)) {
    diff. <- outbreak_1[j, 7] - outbreak_0[k, 7] ##爆發禽場傾向分數-沒有爆發禽場傾向分數
    diff_v[k] <- diff.
  }
}
dat <- rbind.data.frame(outbreak_1[j,], outbreak_0[which.min(abs(diff_v)),]) #找最小的傾向分數

if(dat$num[1] != 0 & dat$num[2] != 0){ #論文p.32 爆發禽場與有無鳥種出現
  a <- c(a, 1)
}
}
```



```

    b <- c(b ,1)
  }
  if(dat$num[1] == 0 & dat$num[2] != 0){
    c <- c(c ,1)
  }
  if(dat$num[1] == 0 & dat$num[2] == 0){
    d <- c(d ,1)
  }
  list <- matrix(c(length(a), length(c), length(b), length(d)), 2, 2)
}
chiq <- c(chiq , mcnemar.test(list)$statistic) #論文P.15
list.f <- rbind.data.frame(list.f,list) #將所有鳥種的2*2列聯表存入

```

Bootstrapping

```

diff1 <- c(diff(dat$score),diff1) #論文p.33設立區間(爆發禽場-無爆發禽場的傾向分數)
mx <- max(abs(max(diff1)),abs(min(diff1))) #論文p.33設立區間(取絕對值後找最大)
inter.val <- c(inter.val , mx) #論文p.33設立區間(每種鳥的區間存入)
chi_list <- c()
bb <- c()
cc <- c()
bbb <- c()
ccc <- c()
if(mx <= 0.01){
  for (i in 1:1000) {
    a <- c()
    b <- c()
    c <- c()
    d <- c()
    for (j in 1:nrow(outbreak_1)) { #在無爆發禽場的傾向分數去挑選屬於這個區間
      score1 <- outbreak_1[j,7]
      score1_u <- score1+0.01
      score1_l <- score1-0.01
      filt <- filter(outbreak_0 , score1_l < score & score < score1_u)
      dat <- rbind.data.frame(outbreak_1[j,] , filt[sample(nrow(filt) , 1),]) #sample隨機選取數
      量
      if(dat$num[1] != 0 & dat$num[2] != 0){
        a <- c(a ,1)
      }
      if(dat$num[1] != 0 & dat$num[2] == 0){
        b <- c(b ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] != 0){
        c <- c(c ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] == 0){
        d <- c(d ,1)
      }
      list = matrix(c(length(a), length(c), length(b), length(d)), 2, 2)
    }
    stasis <- mcnemar.test(list)$statistic ##檢定統計量
    if(is.na(stasis) == T){ ##有可能算出來為na值·就視為0
      stasis <- 0
    }
    chi_list <- c(chi_list , stasis)
    ##### 判斷1000次中為正相關或負相關
    if(length(b) > length(c)){ ##if else語法
      bb <- c(bb , 1) ##正相關
    }
    else{
      cc <- c(cc , 1) ##負相關
    }
  }
}

```

```

#### 判斷顯著次數中為正相關或負相關
if(statis > 3.841){ ##p.15
  if(length(b) > length(c)){
    bbb <- c(bbb , 1)
  }
  else{
    ccc <- c(ccc, 1)
  }
}else next
}
chi_list <- data.frame(chi_list) #轉成資料型態
chi_list_sign <- filter(chi_list , chi_list > 3.841) #哪些是顯著
chi_list_nsign <- filter(chi_list , chi_list < 3.841) #哪些是不顯著
w <- rbind(length(chi_list_sign$chi_list),length(chi_list_nsign$chi_list)) #顯著的次數
x <- rbind(length(bb) , length(cc)) #正相關的次數
y <- rbind(length(bbb),length(ccc)) #負相關的次數
boots <- rbind(boots , cbind(w,x,y))
}
if(mx > 0.01 & mx <= 0.02){
  for (i in 1:1000) {
    a <- c()
    b <- c()
    c <- c()
    d <- c()
    for (j in 1:nrow(outbreak_1)) {
      score1 <- outbreak_1[j,7]
      score1_u <- score1+0.02
      score1_l <- score1-0.02
      filt <- filter(outbreak_0 , score1_l < score & score < score1_u)
      dat <- rbind.data.frame(outbreak_1[j,] , filt[sample(nrow(filt) , 1),])
      if(dat$num[1] != 0 & dat$num[2] != 0){
        a <- c(a ,1)
      }
      if(dat$num[1] != 0 & dat$num[2] == 0){
        b <- c(b ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] != 0){
        c <- c(c ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] == 0){
        d <- c(d ,1)
      }
    }
    list = matrix(c(length(a), length(c), length(b), length(d)), 2, 2)
  }
  statis <- mcnemar.test(list)$statistic
  if(is.na(statis) == T){
    statis <- 0
  }
  chi_list <- c(chi_list , statis)
  if(length(b) > length(c)){
    bb <- c(bb , 1)
  }
  else{
    cc <- c(cc , 1)
  }
  if(statis > 3.841){

```

```

    if(length(b) > length(c)){
      bbb <- c(bbb , 1)
    }
    else{
      ccc <- c(ccc, 1)
    }
  }else next
}
chi_list <- data.frame(chi_list)
chi_list_sign <- filter(chi_list , chi_list > 3.841)
chi_list_nsign <- filter(chi_list , chi_list < 3.841)
w <- rbind(length(chi_list_sign$chi_list),length(chi_list_nsign$chi_list))
x <- rbind(length(bb) , length(cc))
y <- rbind(length(bbb),length(ccc))
boots <- rbind(boots , cbind(w,x,y))
}
if(mx > 0.02 & mx <= 0.03){
  for (i in 1:1000) {
    a <- c()
    b <- c()
    c <- c()
    d <- c()
    for (j in 1:nrow(outbreak_1)) {
      score1 <- outbreak_1[j,7]
      score1_u <- score1+0.03
      score1_l <- score1-0.03
      filt <- filter(outbreak_0 , score1_l < score & score < score1_u)
      dat <- rbind.data.frame(outbreak_1[j,] , filt[sample(nrow(filt) , 1),])
      if(dat$num[1] != 0 & dat$num[2] != 0){
        a <- c(a ,1)
      }
      if(dat$num[1] != 0 & dat$num[2] == 0){
        b <- c(b ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] != 0){
        c <- c(c ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] == 0){
        d <- c(d ,1)
      }
    }
    list = matrix(c(length(a), length(c), length(b), length(d)), 2, 2)
  }
  stasis <- mcnemar.test(list)$statistic
  if(is.na(stasis) == T){
    stasis <- 0
  }
  chi_list <- c(chi_list , stasis)
  if(length(b) > length(c)){
    bb <- c(bb , 1)
  }
  else{
    cc <- c(cc , 1)
  }
  if(stasis > 3.841){

```

```

    if(length(b) > length(c)){
      bbb <- c(bbb , 1)
    }
    else{
      ccc <- c(ccc, 1)
    }
  }else next
}
chi_list <- data.frame(chi_list)
chi_list_sign <- filter(chi_list , chi_list > 3.841)
chi_list_nsign <- filter(chi_list , chi_list < 3.841)
w <- rbind(length(chi_list_sign$chi_list),length(chi_list_nsign$chi_list))
x <- rbind(length(bb) , length(cc))
y <- rbind(length(bbb),length(ccc))
boots <- rbind(boots , cbind(w,x,y))
}
if(mx > 0.03 & mx <= 0.04){
  for (i in 1:1000) {
    a <- c()
    b <- c()
    c <- c()
    d <- c()
    for (j in 1:nrow(outbreak_1)) {
      score1 <- outbreak_1[j,7]
      score1_u <- score1+0.04
      score1_l <- score1-0.04
      filt <- filter(outbreak_0 , score1_l < score & score < score1_u)
      dat <- rbind.data.frame(outbreak_1[j,] , filt[sample(nrow(filt) , 1),])
      if(dat$num[1] != 0 & dat$num[2] != 0){
        a <- c(a ,1)
      }
      if(dat$num[1] != 0 & dat$num[2] == 0){
        b <- c(b ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] != 0){
        c <- c(c ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] == 0){
        d <- c(d ,1)
      }
      list = matrix(c(length(a), length(c), length(b), length(d)), 2, 2)
    }
    stasis <- mcnemar.test(list)$statistic
    if(is.na(stasis) == T){
      stasis <- 0
    }
    chi_list <- c(chi_list , stasis)
    if(length(b) > length(c)){
      bb <- c(bb , 1)
    }
    else{
      cc <- c(cc , 1)
    }
  }
  if(stasis > 3.841){

```

```

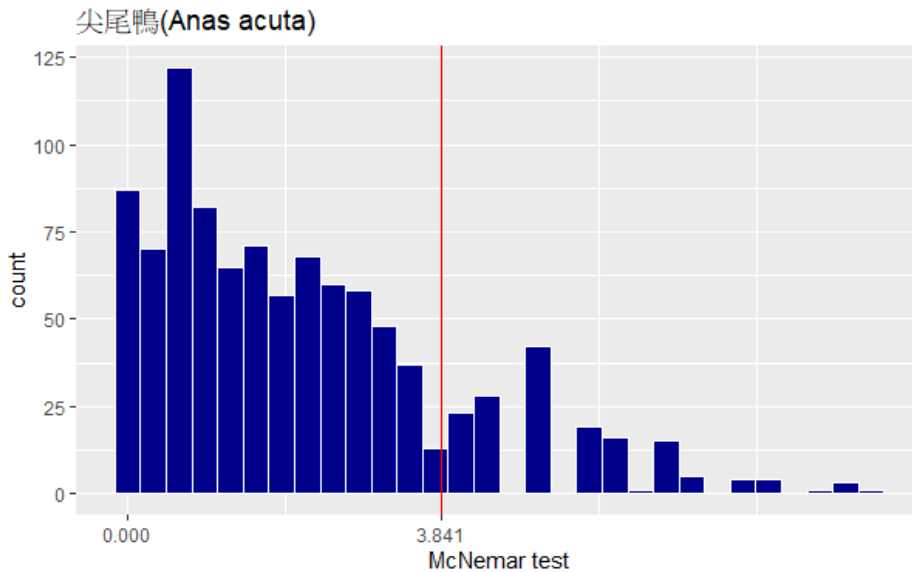
    if(length(b) > length(c)){
      bbb <- c(bbb , 1)
    }
    else{
      ccc <- c(ccc, 1)
    }
  }else next
}
chi_list <- data.frame(chi_list)
chi_list_sign <- filter(chi_list , chi_list > 3.841)
chi_list_nsign <- filter(chi_list , chi_list < 3.841)
w <- rbind(length(chi_list_sign$chi_list),length(chi_list_nsign$chi_list))
x <- rbind(length(bb) , length(cc))
y <- rbind(length(bbb),length(ccc))
boots <- rbind(boots , cbind(w,x,y))
}
if(mx > 0.04){
  for (i in 1:1000) {
    a <- c()
    b <- c()
    c <- c()
    d <- c()
    for (j in 1:nrow(outbreak_1)) {
      score1 <- outbreak_1[j,7]
      score1_u <- score1+0.05
      score1_l <- score1-0.05
      filt1 <- filter(outbreak_0 , score1_l < score & score < score1_u)
      dat <- rbind.data.frame(outbreak_1[j,] , filt[sample(nrow(filt) , 1),])
      if(dat$num[1] != 0 & dat$num[2] != 0){
        a <- c(a ,1)
      }
      if(dat$num[1] != 0 & dat$num[2] == 0){
        b <- c(b ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] != 0){
        c <- c(c ,1)
      }
      if(dat$num[1] == 0 & dat$num[2] == 0){
        d <- c(d ,1)
      }
      list = matrix(c(length(a), length(c), length(b), length(d)), 2, 2)
    }
    stasis <- mcnemar.test(list)$statistic
    if(is.na(stasis) == T){
      stasis <- 0
    }
    chi_list <- c(chi_list , stasis)
    if(length(b) > length(c)){
      bb <- c(bb , 1)
    }
    else{
      cc <- c(cc , 1)
    }
  }
  if(stasis > 3.841){

```

```

    if(length(b) > length(c)){
      bbb <- c(bbb , 1)
    }
    else{
      ccc <- c(ccc, 1)
    }
  }else next
}
chi_list <- data.frame(chi_list)
chi_list_sign <- filter(chi_list , chi_list > 3.841)
chi_list_nsign <- filter(chi_list , chi_list < 3.841)
w <- rbind(length(chi_list_sign$chi_list),length(chi_list_nsign$chi_list))
x <- rbind(length(bb) , length(cc))
y <- rbind(length(bbb),length(ccc))
boots <- rbind(boots , cbind(w,x,y))
}
}
# 畫圖
ggplot() + geom_histogram(aes(x = chi_list$chi_list) , col = "white" , fill = "darkblue") +
  ggtitle("尖尾鴨(Anas acuta)") + geom_vline(xintercept = 3.841 , col="red") +
  scale_x_continuous(name = "McNemar test",breaks = c(0,3.841,10,20))

```



wild bird risk mapping

```
# 讀取資料
poultry <- read.csv("C:/Users/user/Desktop/R/data2/禽場.csv")
score <- read.csv("C:/Users/user/Desktop/R/禽場風險地圖/全部鳥的PSM(有加環境因子).csv")
score2 <- read.csv("C:/Users/user/Desktop/R/禽場風險地圖/全部鳥的PSM(沒有加環境因子).csv")

# 開始計算風險值-1
poultry$id <- 1:nrow(poultry)
poultry$爆發比例 <- poultry$爆發禽場/poultry$All_PF
poultry$爆發比例[is.na(poultry$爆發比例)] <- 0
poultry$爆發比例[which(poultry$爆發比例 == Inf)] <- 1
poultry$面積 <- poultry$Tarea/9000000

#(方法一)顯著次數大於100次以上
## 水鳥(沒有環境因子)
poultry$risk <- ((0.544*score$長趾濱鵒+0.896*score$紅嘴鷗+0.116*score$小+
                0.171*score$太平洋金斑+0.138*score$高+0.416*score$小燕鷗+
                0.119*score$家燕)/7)*(poultry$爆發比例)*(poultry$面積)

## 水鳥(環境)
poultry$risk2 <- (((0.784*score2$X49.長趾濱鵒+0.482*score2$X8.紅嘴鷗+0.402*score2$X21.小..
                  +0.191*score2$X31.田鵒+0.174*score2$X13.尖尾鴨+
                  0.157*score2$X11.太平洋金斑.)/6)*(poultry$爆發比例)*(poultry$面積)

## 留鳥(沒有環境因子)
poultry$risk <- ((0.431*score$X4.紅鳩+0.355*score$X5.黃頭鷺+0.162*score$X9.紅嘴黑鴨+
                0.215*score$X14.紅冠水雞)/4)*(poultry$爆發比例)*(poultry$面積)

## 留鳥(環境)
poultry$risk <- ((0.990*score$X9.紅嘴黑鴨+0.489*score$X11.家八哥+
                0.127*score$X26.繡眼畫眉)/4)*(poultry$爆發比例)*(poultry$面積)

#(方法二)全部鳥都納入
## 水鳥(沒有環境因子)
poultry$risk <- (((0.668*score$X1.高蹺.+0.248*score$X11.太平洋金斑.+0.209*score$X21.小..+
                  0.211*score$X31.田鵒+0.104*score$X33.白翅黑燕鷗+0.489*score$X49.長趾濱鵒+
                  0.485*score$X55.家燕+0.020*score$X2.黑腹燕鷗+0.057*score$X4.黑腹濱鵒+
                  0.005*score$X7.小水鴨+0.063*score$X8.紅嘴鷗+0.001*score$X12.鳳頭潛鴨+
                  0.021*score$X13.尖尾鴨+0.039*score$X14.反嘴鷗+0.001*score$X16.紅胸濱鵒+
                  0.021*score$X17.裏海燕鷗+0.010*score$X19.小環頸.+0.053*score$X20.小燕鷗+
                  0.002*score$X24.大杓鷗+0.016*score$X32.白冠雞+0.012*score$X39.池鷺+
                  0.034*score$X40.灰斑.+0.004*score$X44.夜鷺+0.045*score$X45.東方鷺+
                  0.012*score$X60.絲光椋鳥+0.007*score$X65.遊隼+0.064*score$X75.鷹斑鷗)/27)*(pou
                  0.171*score$太平洋金斑+0.138*score$高+0.416*score$小燕鷗+
                  0.119*score$家燕)/7)*(poultry$爆發比例)*(poultry$面積)

## 水鳥(環境)
poultry$risk2 <- (((0.784*score2$X49.長趾濱鵒+0.482*score2$X8.紅嘴鷗+0.402*score2$X21.小..
                  +0.191*score2$X31.田鵒+0.174*score2$X13.尖尾鴨+
                  0.157*score2$X11.太平洋金斑.)/6)*(poultry$爆發比例)*(poultry$面積)
```

留鳥(沒有環境因子)

```
poultry$risk <- ((0.431*score$X4.紅鳩+0.355*score$X5.黃頭鷺+0.162*score$X9.紅嘴黑鴨+
0.215*score$X14.紅冠水雞)/4)*(poultry$爆發比例)*(poultry$面積)
```

留鳥(環境)

```
poultry$risk <- ((0.990*score$X9.紅嘴黑鴨+0.489*score$X11.家八哥+
0.127*score$X26.繡眼畫眉)/4)*(poultry$爆發比例)*(poultry$面積)
```

#(方法二)全部鳥都納入

水鳥(沒有環境因子)

```
poultry$risk <- ((0.668*score$X1.高蹺.+0.248*score$X11.太平洋金斑.+0.209*score$X21.小.+
0.211*score$X31.田鷺+0.104*score$X33.白翅黑燕鷗+0.489*score$X49.長趾濱鷺+
0.485*score$X55.家燕+0.020*score$X2.黑腹燕鷗+0.057*score$X4.黑腹濱鷺+
0.005*score$X7.小水鴨+0.063*score$X8.紅嘴鷗+0.001*score$X12.鳳頭潛鴨+
0.021*score$X13.尖尾鴨+0.039*score$X14.反嘴鷺+0.001*score$X16.紅胸濱鷺+
0.021*score$X17.裏海燕鷗+0.010*score$X19.小環頸.+0.053*score$X20.小燕鷗+
0.002*score$X24.大杓鷺+0.016*score$X32.白冠雞+0.012*score$X39.池鷺+
0.034*score$X40.灰斑.+0.004*score$X44.夜鷺+0.045*score$X45.東方鷺+
0.012*score$X60.絲光棕鳥+0.007*score$X65.遊隼+0.064*score$X75.鷹斑鷺)/27)*(pou
ltry$爆發比例)*(poultry$面積)
```

水鳥(環境)

```
poultry$risk <- ((0.112*score$X1.高蹺.+0.120*score$X8.紅嘴鷗+0.138*score$X11.太平洋金斑.+
0.107*score$X13.尖尾鴨+0.261*score$X20.小燕鷗+0.464*score$X21.小.+
0.398*score$X31.田鷺+0.775*score$X41.灰鵲鷺+0.146*score$X45.東方鷺+
0.652*score$X49.長趾濱鷺+0.050*score$X2.黑腹燕鷗+0.013*score$X4.黑腹濱鷺+
0.002*score$X7.小水鴨+0.003*score$X12.鳳頭潛鴨+0.014*score$X14.反嘴鷺+
0.001*score$X16.紅胸濱鷺+0.056*score$X17.裏海燕鷗+0.013*score$X24.大杓鷺+
0.014*score$X29.中杓鷺+0.010*score$X32.白冠雞+0.018*score$X33.白翅黑燕鷗+
0.014*score$X39.池鷺+0.014*score$X44.夜鷺+0.009*score$X54.紅頭潛鴨+
0.033*score$X55.家燕+0.042*score$X60.絲光棕鳥+0.006*score$X63.黑尾鷺+
0.004*score$X65.遊隼+0.071*score$X70.燕鷗+0.001*score$X74.彎嘴濱鷺)/30)*(pou
ltry$爆發比例)*(poultry$面積)
```

留鳥(沒有環境因子)

```
poultry$risk <- ((0.431*score$X4.紅鳩+0.355*score$X5.黃頭鷺+0.162*score$X9.紅嘴黑鴨+
0.215*score$X14.紅冠水雞+0.003*score$X1.麻雀+0.003*score$X2.白頭翁+
0.007*score$X8.珠頸斑鳩+0.019*score$X11.家八哥+0.028*score$X13.小雨燕+
0.004*score$X17.樹鵲+0.004*score$X18.大卷尾+0.011*score$X24.金背鳩+
0.016*score$X29.棕沙燕+0.002*score$X32.白腰文鳥+0.012*score$X33.黑鷺+
0.038*score$X34.黑頭文鳥+0.007*score$X39.鵲鷺)/17)*(poultry$爆發比例)*(poultry
$面積)
```

留鳥(環境)

```
poultry$risk <- ((0.990*score$X9.紅嘴黑鴨+0.489*score$X11.家八哥+0.127*score$X26.繡眼畫眉+
0.005*score$X4.紅鳩+0.011*score$X5.黃頭鷺+0.005*score$X7.野鴿+
0.011*score$X8.珠頸斑鳩+0.001*score$X12.洋燕+0.002*score$X14.紅冠水雞+
0.004*score$X16.埃及聖.+0.009*score$X17.樹鵲+0.010*score$X20.白鵲鷺+
0.020*score$X21.五色鳥+0.010*score$X24.金背鳩+0.057*score$X31.巨嘴鴉+
0.005*score$X32.白腰文鳥+0.009*score$X33.黑鷺+0.011*score$X34.黑頭文鳥+
0.049*score$X39.鵲鷺)/19)*(poultry$爆發比例)*(poultry$面積)
```



```
# 開始計算風險值-2
poultry$risk <- (poultry$risk)*10000
poultry$risk2 <- (poultry$risk2)*10000
poultry$total <- poultry$risk+poultry$risk2
write.csv(poultry, 'C:/Users/user/Desktop/poultry.csv')

#與台灣網格合併
tw.map = st_read("C:/Users/user/Desktop/R/data/G3km_3826_environment_dataset/G3km_3826_environmental_dataset.shp") ##將檔案讀進來
tw.map_2 <- data.frame(tw.map[,1:5])
poultry2 <- merge(tw.map_2,poultry,by="id")
poultry3 <- poultry[-which(poultry2$total == 0),] ##剔除風險值為0的值
poultry3<- (poultry3[order(poultry3$total,decreasing = T),]) #按照順序大小排序
result <- poultry3[1:60,] ##取前20%
result2 <-poultry3[61:296,] ##剩下80%
write.csv(result, 'C:/Users/user/Desktop/R/禽場風險地圖/20220215/20220216_前20%.csv')
write.csv(result2, 'C:/Users/user/Desktop/R/禽場風險地圖/20220215/20220216_剩下的80%.csv')

#再用qgis畫圖
```

t_sne_cluster

```
library(Rtsne) ##tsne package
```

```
## Warning: 套件 'Rtsne' 是用 R 版本 4.1.2 來建造的
```

```
library(ggplot2)  
library(ggthemes) ##沒有安裝過要先安裝-install.packages("ggthemes")
```

```
## Warning: 套件 'ggthemes' 是用 R 版本 4.1.2 來建造的
```

```
land <- read.csv('E:/20220310/9個陸域因子.csv') #檔案路徑  
land$id <- as.character(land$id)  
land$sum <- rowSums(land[,2:10])  
land <- land[!(rownames(land) %in% rownames(land[land$sum %in% 0,])),]  
land <- land[,-c(ncol(land))]  
land[,2:10] <- land[,2:10]/(3000*3000)*100  
  
set.seed(65)  
tSNE_land <- land %>%  
  select(where(is.numeric)) %>%  
  scale() %>%  
  Rtsne(check_duplicates=F)  
  
tSNE_df_land <- tSNE_land$Y %>%  
  as.data.frame() %>%  
  rename(tSNE1="V1", tSNE2="V2")  
tSNE_df_land$id <- land$id  
write.table(x = tSNE_df_land, file = 'E:/e_bird/land_tsne.txt', quote = F) ##存入檔案  
  
#png(paste(path, 'row land tsne.png', sep = ''), width = 1600, height = 900)  
ggplot(tSNE_df_land, aes(x = tSNE1, y = tSNE2,)) + geom_point(size=2) +  
  ggtitle(label = 'Land t-SNE') + theme_light()  
dev.off()  
  
land <- read.csv('E:/20220310/9個陸域因子.csv') #檔案路徑  
rownames(land) <- land$id  
land <- land[,-c(1)]  
land$sum <- rowSums(land)  
land <- land[!(rownames(land) %in% rownames(land[land$sum %in% 0,])),]  
land <- land[,-c(ncol(land))]  
bird <- read.delim('E:/20220310/全部鳥的PSM(有環境因子).txt', encoding = 'BIG-5')  
bird$id <- as.character(bird$id)  
rownames(bird) <- bird$id  
bird <- bird[(rownames(bird) %in% rownames(land)),]  
  
set.seed(142)  
tSNE_fit <- bird %>%  
  select(where(is.numeric)) %>%  
  scale() %>%  
  Rtsne(check_duplicates=F)
```

```

tSNE_df <- tSNE_fit$Y %>%
  as.data.frame() %>%
  rename(tSNE1="V1",tSNE2="V2")
tSNE_df[,c('id')] <- bird[,c('id')]
write.table(tSNE_df, 'E:/e_bird/bird_tsne.txt') ##存入檔案

bird_tsne <- read.delim('E:/e_bird/bird_tsne.txt', sep = ' ') ##將檔案讀進來

gg_theme= theme(
  axis.title.x = element_text(size = 36),
  axis.text.x = element_text(size = 34),
  axis.title.y = element_text(size = 36),
  axis.text.y = element_text(size = 34),
  title = element_text(size = 40),
  plot.caption = element_text(size = 32),
  legend.text = element_text(size = 14),
  legend.key.size = unit(2, 'lines'))

#png(paste(path, 'bird tsne_kmean_top3.png', sep = ''), width = 1600, height = 900)
set.seed(123)
result <- ggplot(bird_tsne, aes(x = tSNE1, y = tSNE2))+geom_point(size=3)+
  ggtitle(label = 'Bird t-SNE with cluster', subtitle = 'Label the top5 bird prediction (mean, n
o minus outliers)') +
  theme_light()+guides(color = guide_legend(override.aes = list(size = 3))) +
  labs( colour = "top3")+gg_theme+scale_color_manual(values = c(topo.colors(nrow(as.data.frame(
table(bird_tsne$top5))))))
result
dev.off()

# lable by x% land factor
land <- read.csv('E:/20220310/9個陸域因子.csv')

land$sum <- rowSums(land[,2:10])
land <- land[!(rownames(land) %in% rownames(land[land$sum %in% 0,])),]
land <- land[,-c(ncol(land))]
land[,2:10] <- land[,2:10]/(3000*3000)*100
land$cluster <- 'mix'
for(i in seq(2,10)) {
  row <- rownames(land[land[,i] >= 90,])
  land[row,'cluster'] <- colnames(land)[i]
}
land <- land[order(land$id, decreasing = F),]

land_tsne <- read.delim(file = 'E:/e_bird/land_tsne.txt', sep = ' ')

land_tsne <- land_tsne[order(land_tsne$id, decreasing = F),]
land_tsne[,c('id_2', 'cluster')] <- land[,c('id', 'cluster')]

sum(as.numeric(land_tsne$id_2)[1:nrow(land_tsne)]-as.numeric(land_tsne$id)[1:nrow(land_tsne)])

#path <- c('~/.Analysis/e_bird/land_plot/')
#png(paste(path, 'row land tsne_land_factor.png', sep = ''), width = 1600, height = 900)
set.seed(123)

```

```

ggplot(land_tsne, aes(x = tSNE1, y = tSNE2, color=cluster))+geom_point(size=2)+
  ggtitle(label = 'Land t-SNE with clustering', subtitle = 'Lable the 90% land factor') +
  theme_light()+gg_theme+guides(color = guide_legend(override.aes = list(size = 5))) +
  gg_theme+scale_color_manual(values = c('#F2A405' , '#288C73', '#87EFEE', '#8CC051',
'#A1ACB1', '#967BDC', '#4B89DC'))
dev.off()

bird_tsne <- read.delim(file = 'E:/e_bird/bird_tsne.txt', sep = ' ')

bird_tsne <- bird_tsne[order(bird_tsne$id, decreasing = F),]
bird_tsne[,c('id_2', 'cluster')] <- land[,c('id', 'cluster')]

sum(as.numeric(bird_tsne$id_2)[1:nrow(bird_tsne)]-as.numeric(bird_tsne$id)[1:nrow(bird_tsne)])

gg_theme= theme(
  axis.title.x = element_text(size = 36),
  axis.text.x = element_text(size = 34),
  axis.title.y = element_text(size = 36),
  axis.text.y = element_text(size = 34),
  title = element_text(size = 40),
  plot.caption = element_text(size = 32),
  legend.text = element_text(size = 14),
  legend.key.size = unit(2, 'lines'))

#png(paste(path, 'bird tsne_land_factor.png', sep = ''), width = 1600, height = 900)
set.seed(123)
ggplot(bird_tsne, aes(x = tSNE1, y = tSNE2,color = cluster))+geom_point(size=2)+
  ggtitle(label = 'Bird t-SNE with cluster', subtitle = 'Lable the 90% land factor') +
  theme_light()+guides(color = guide_legend(override.aes = list(size = 3))) +labs( colour = "cluster")+
  gg_theme+scale_color_manual(values = c('#F2A405' , '#288C73', '#87EFEE', '#8CC051' , '#A1ACB
1', '#967BDC', '#4B89DC'))
dev.off()

```