Final Paper Code

Jack Magee

2022-12-01

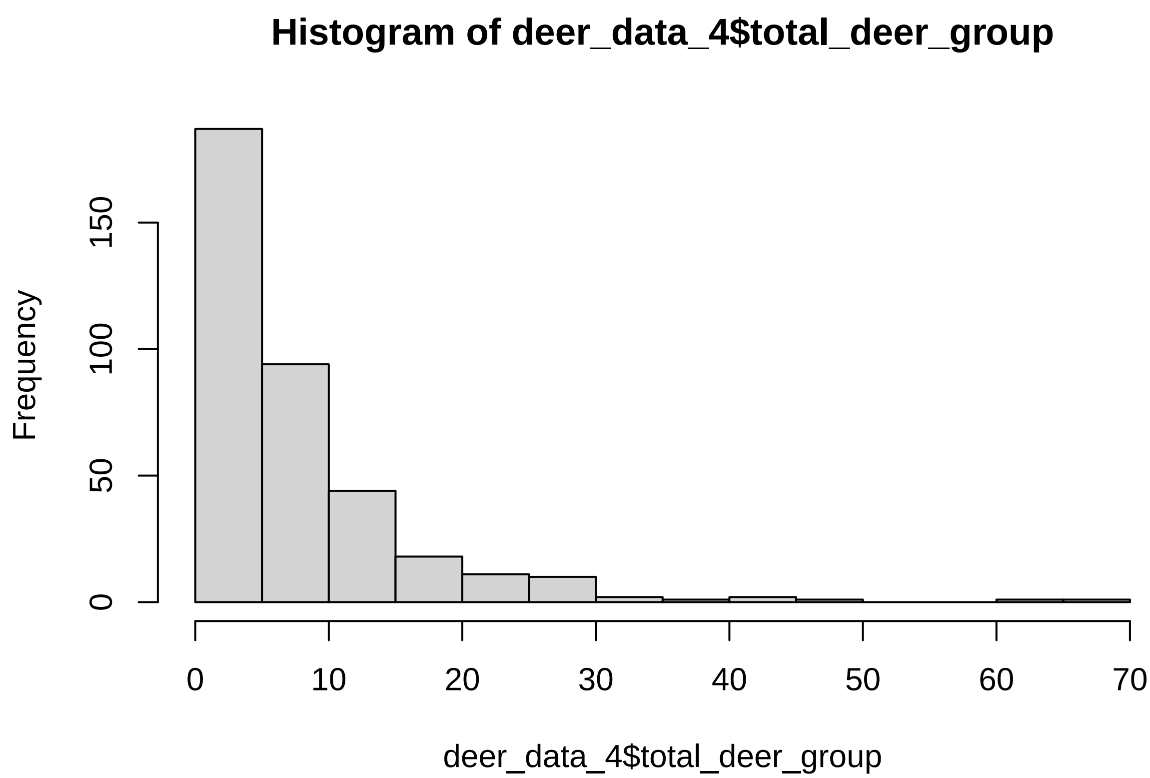
Required packages

```
#install.packages("MASS")
#install.packages("CAR")
#install.packages("ggplot2")
#install.packages("pacman")
```

```
#read in the data (Level 0) from GitHub
deer_data <- read.csv('https://raw.githubusercontent.com/mageejac/Eco-870-Final/main/DBS_2022_raw.csv')
```

Data Cleaning (Level 1 data)

```
#remove duplicated observations of the same group
deer_data_2 <- deer_data[!duplicated(deer_data$Group.GID), ]

#subset the data for group size, habitat, and location

deer_data_3 <- deer_data_2[,c("habitat_group","total_deer_group","obs.x","obs.y")]

#remove observations with unknown habitat type

deer_data_4 <- deer_data_3[!(deer_data_3$habitat_group=="unk" | deer_data_3$habitat_group=="tilled_unk"

# plot data distribution to choose distribution
hist(deer_data_4$total_deer_group)
```




```r
#choose negative binomial distribution because of large difference between variance and mean and over d
mean(deer_data_4$total_deer_group)
```

```
## [1] 8.153226
```

```r
var(deer_data_4$total_deer_group)
```

```
## [1] 72.91716
```

Modeling the data

```r
#load packages

library(MASS)
```

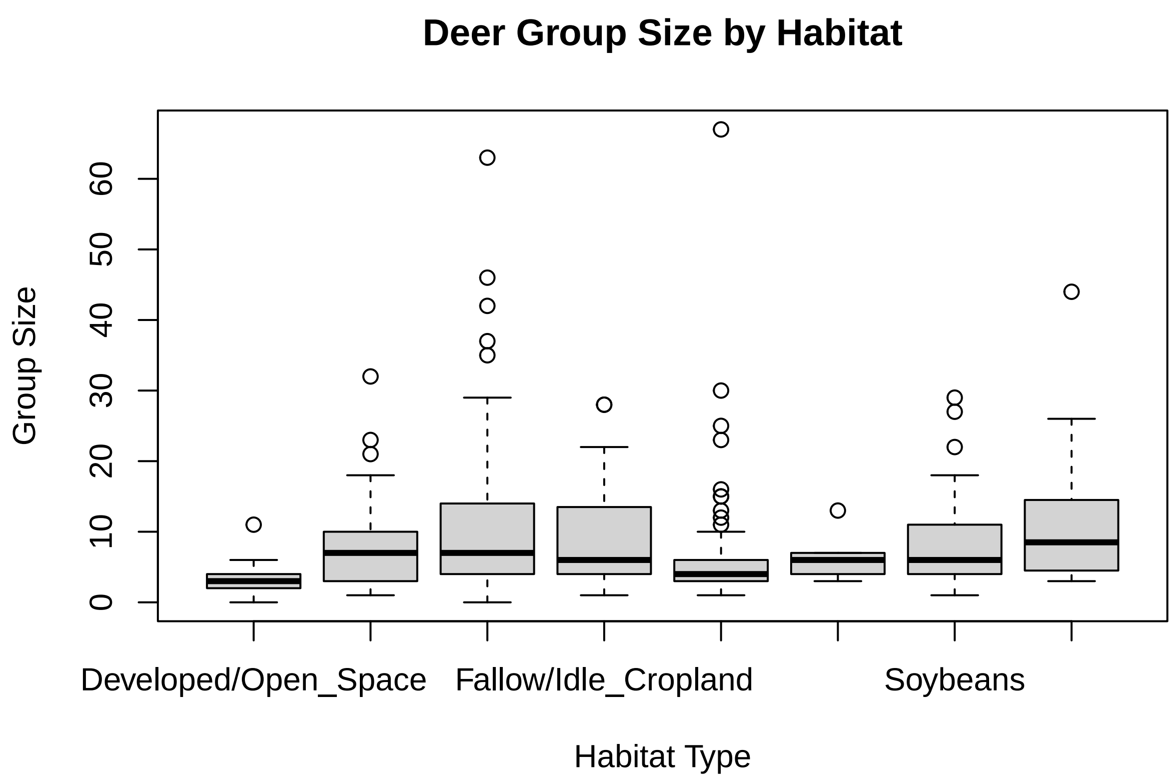
```
## Warning: package 'MASS' was built under R version 4.2.2
```

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.2
## Loading required package: carData
```

```r
#relevel data with Developed/Open Space habitat types as a control

deer_data_4$habitat_group = as.factor(deer_data_4$habitat_group)
deer_data_4$habitat_group_ref <- relevel(deer_data_4$habitat_group, ref = "Developed/Open_Space")

# create glm
model_1 <- glm.nb(total_deer_group ~ habitat_group_ref, data = deer_data_4)
```

```r
#generating summary statistics

boxplot(total_deer_group ~ habitat_group_ref, data = deer_data_4,
        main = "Deer Group Size by Habitat",
        xlab = "Habitat Type",
        ylab = "Group Size"
)
```




```r
Anova(model_1)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: total_deer_group
##                   LR Chisq Df Pr(>Chisq)
## habitat_group_ref   50.576  7  1.113e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(model_1)
```

```
##
## Call:
## glm.nb(formula = total_deer_group ~ habitat_group_ref, data = deer_data_4,
##     init.theta = 1.837655393, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.6332  -0.9686  -0.4332   0.3060   4.8434
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           1.1676     0.1748   6.681 2.38e-11 ***
## habitat_group_refAlfalfa              0.8486     0.2093   4.054 5.04e-05 ***
## habitat_group_refCorn                 1.1632     0.1896   6.136 8.47e-10 ***
## habitat_group_refFallow/Idle_Cropland   1.0260     0.2274   4.513 6.40e-06 ***
## habitat_group_refForest               0.6500     0.1962   3.314 0.000921 ***
## habitat_group_refShrubland            0.7195     0.4119   1.747 0.080709 .
## habitat_group_refSoybeans             1.0023     0.2247   4.460 8.18e-06 ***
## habitat_group_refWinter_Wheat         1.2962     0.2643   4.904 9.40e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.8377) family taken to be 1)
##
##     Null deviance: 431.68  on 371  degrees of freedom
## Residual deviance: 381.11  on 364  degrees of freedom
## AIC: 2288
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.838
##           Std. Err.:  0.159
##
##  2 x log-likelihood:  -2269.959
```

```
#URL for CropScapeR package: https://tmieno2.github.io/R-as-GIS-for-Economists/CropScapeR.html

if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman

## Warning: package 'pacman' was built under R version 4.2.2
```

```
pacman::p_load(
  stars, # spatiotemporal data handling
  terra, # raster data handling
  raster, # raster data handling
  sf, # vector data handling
  dplyr, # data wrangling
  stringr, # string manipulation
  lubridate, # dates handling
  data.table, # data wrangling
  tidyr, # reshape
  tidyUSDA, # download USDA NASS data
  keyring, # API key management
  FedData, # download Daymet data
  daymetr, # download Daymet data
  ggplot2, # make maps
  tmap, # make maps
  future.apply, # parallel processing
  CropScapeR, # download CDL data
  prism, # download PRISM data
```

```r
  exactextractr # extract raster values to sf
)

#set up map themes

theme_set(theme_bw())

theme_for_map <- theme(
  axis.ticks = element_blank(),
  axis.text= element_blank(),
  axis.line = element_blank(),
  panel.border = element_blank(),
  panel.grid.major = element_line(color='transparent'),
  panel.grid.minor = element_line(color='transparent'),
  panel.background = element_blank(),
  plot.background = element_rect(fill = "transparent",color='transparent')
)

#load/ install packages

library(CropScapeR)
library(devtools)
```

```
## Warning: package 'devtools' was built under R version 4.2.2

## Loading required package: usethis
```

```r
#devtools::install_github("cbw1243/CropScapeR")
library(CropScapeR)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v tibble  3.1.8     v purrr   0.3.5
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x data.table::between()    masks dplyr::between()
## x lubridate::date()        masks base::date()
## x tidyr::extract()         masks raster::extract(), terra::extract()
## x dplyr::filter()          masks stats::filter()
## x data.table::first()      masks dplyr::first()
## x data.table::hour()       masks lubridate::hour()
## x lubridate::intersect()   masks raster::intersect(), terra::intersect(), base::intersect()
## x data.table::isoweek()    masks lubridate::isoweek()
## x dplyr::lag()             masks stats::lag()
## x data.table::last()       masks dplyr::last()
## x data.table::mday()       masks lubridate::mday()
## x data.table::minute()     masks lubridate::minute()
## x data.table::month()      masks lubridate::month()
## x data.table::quarter()    masks lubridate::quarter()
## x dplyr::recode()          masks car::recode()
## x data.table::second()     masks lubridate::second()
## x dplyr::select()          masks raster::select(), MASS::select()
```

```
## x lubridate::setdiff()     masks base::setdiff()
## x purrr::some()            masks car::some()
## x purrr::transpose()       masks data.table::transpose()
## x lubridate::union()       masks raster::union(), terra::union(), base::union()
## x data.table::wday()       masks lubridate::wday()
## x data.table::week()       masks lubridate::week()
## x data.table::yday()       masks lubridate::yday()
## x data.table::year()       masks lubridate::year()
```

```
#Generate study area map and extent

Study_Area <- tigris::counties(state = "MI", cb = TRUE) %>%
  st_as_sf() %>%
  filter(NAME %in% c("Ionia", "Clinton", "Eaton", "Ingham", "Shiawassee"))
```

```
## Retrieving data for the year 2020
```

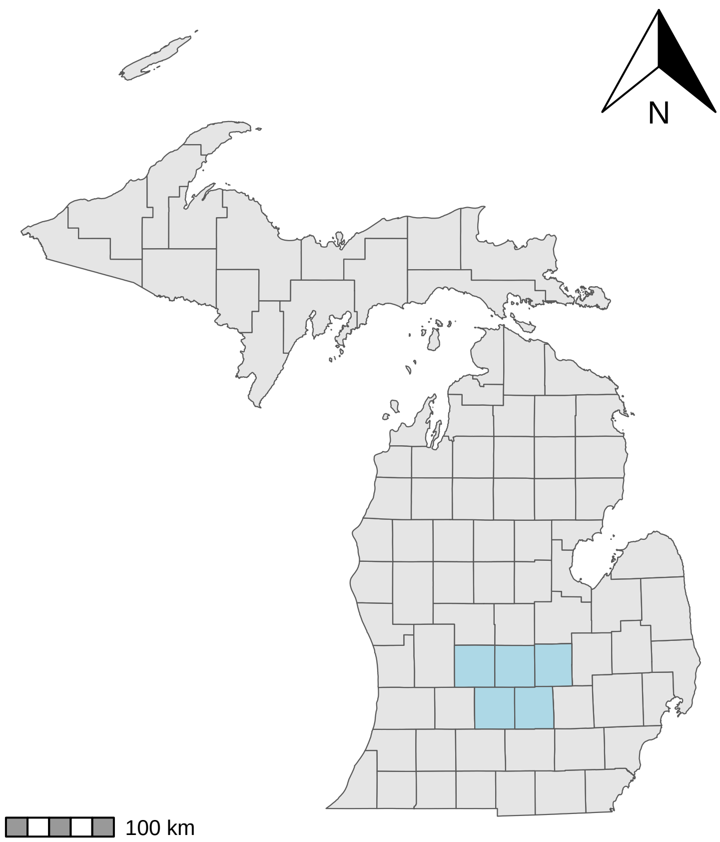
```
##   |                                                                              |
```

```
MICH_county <- tigris::counties("Michigan", cb = TRUE)
```

```
## Retrieving data for the year 2020
```

```
ggplot() +
  geom_sf(data = MICH_county) +
  geom_sf(data =Study_Area, fill = "lightblue") +
  theme_void()+
  ggspatial::annotation_scale(
    location = "bl",
    bar_cols = c("grey60", "white")) +
  ggspatial::annotation_north_arrow(location = "tr", which_north = "true")
```

```
## Scale on map varies by more than 10%, scale bar may be inaccurate
```




```
#pull in USDA crop cover data for study area

(
  cdl_MICH_SA <- GetCDLData(
    aoi = Study_Area,
    year = "2021",
    type = "b",
  )
)
```

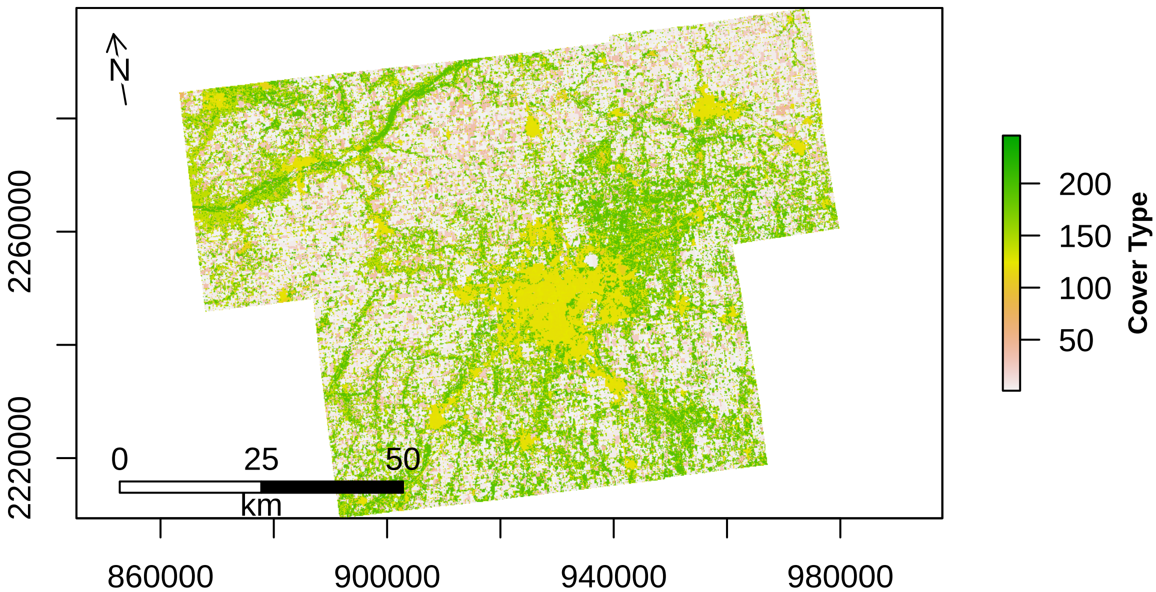
```
## class      : RasterLayer
## dimensions : 3005, 3884, 11671420  (nrow, ncol, ncell)
## resolution : 30, 30  (x, y)
## extent     : 863325, 979845, 2209365, 2299515  (xmin, xmax, ymin, ymax)
## crs        : +proj=aea +lat_0=23 +lon_0=-96 +lat_1=29.5 +lat_2=45.5 +x_0=0 +y_0=0 +datum=NAD83 +units
## source     : CDL_2021_clip_20221213193943_663425537.tif
## names      : Layer_1
```

```
#look up projections

#terra::crs(Study_Area)
#terra::crs(cdl_MICH_SA)
```

#clip crop cover data to study area

```
cdl_MICH_SA_masked <- Study_Area %>%
  #--- change the CRS first to that of the raster data ---#
  st_transform(., projection(cdl_MICH_SA)) %>%
```

```
  #--- mask the values outside the sf (turn them into NA) ---#
  raster::mask(cdl_MICH_SA, .)

#plot Crop cover data for the study area

plot(cdl_MICH_SA_masked,
     legend.args=list(text='Cover Type', side=4, font=2, line=2.5, cex=0.8))
scalebar(50000, type="bar", divs = 2, label = c(0,25,50),below="km")
north("topleft", angle = 350)
```




```
data("linkdata")

#pull in crop codes for USDA crop layer data

crop_codes <- read.csv("https://raw.githubusercontent.com/mageejac/Eco-870-Final/main/cdl_codes_names.cs

head(crop_codes)
```

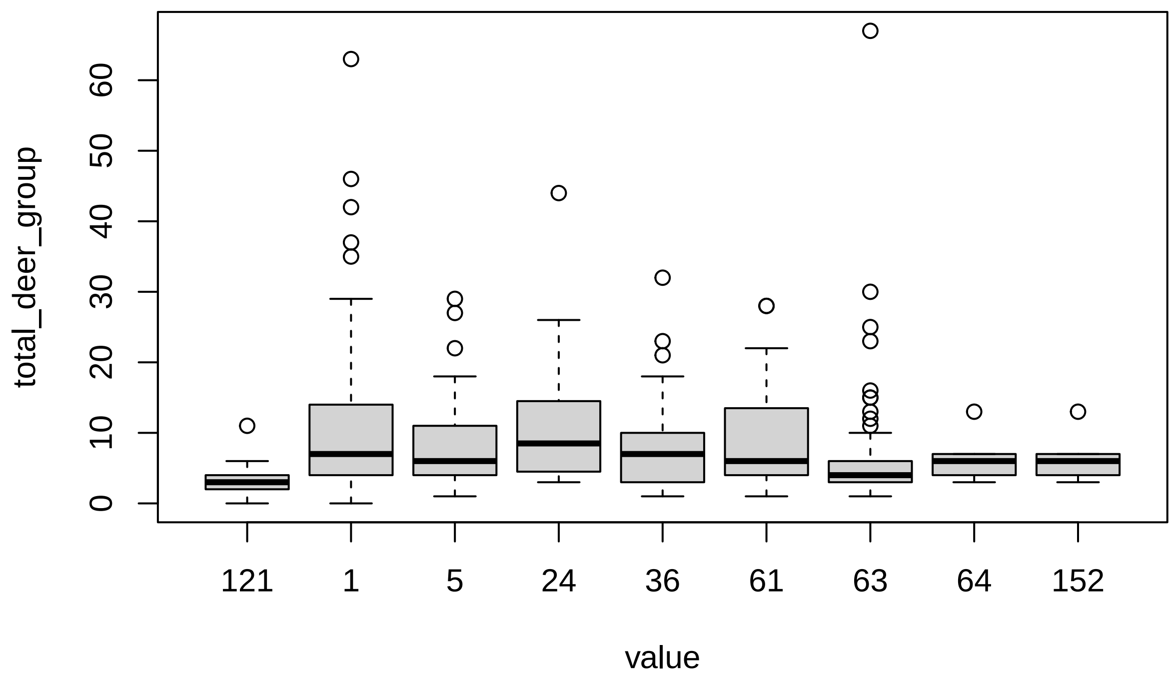
```
##   MasterCat     Crop
## 1         0   NoData
## 2         1     Corn
## 3         2   Cotton
## 4         3     Rice
## 5         4  Sorghum
## 6         5 Soybeans
```

```r
# merge with data set

colnames(crop_codes)[which(names(crop_codes) == "Crop")] <- "habitat_group_ref"

deer_data_5 <- merge(crop_codes, deer_data_4, by ='habitat_group_ref')

colnames(deer_data_5)[which(names(deer_data_5) == "MasterCat")] <- "value"

deer_data_5$value <- as.factor(deer_data_5$value)

deer_data_5$value <- relevel(deer_data_5$value, ref = "121")
```

```r
#generate second model to predict congregations across the landscape with crop code value

model_2 <- glm.nb(total_deer_group ~ value, data = deer_data_5)

summary(model_2)
```

```
##
## Call:
## glm.nb(formula = total_deer_group ~ value, data = deer_data_5,
##     init.theta = 1.856216391, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6405  -0.9713  -0.4349   0.3073   4.8642
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.1676     0.1742   6.702 2.06e-11 ***
## value1        1.1632     0.1889   6.156 7.44e-10 ***
## value5        1.0023     0.2239   4.476 7.59e-06 ***
## value24       1.2962     0.2633   4.923 8.54e-07 ***
## value36       0.8486     0.2086   4.068 4.75e-05 ***
## value61       1.0260     0.2266   4.529 5.93e-06 ***
## value63       0.6500     0.1955   3.325 0.000886 ***
## value64       0.7195     0.4104   1.753 0.079563 .
## value152      0.7195     0.4104   1.753 0.079563 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.8562) family taken to be 1)
##
##     Null deviance: 437.22  on 376  degrees of freedom
## Residual deviance: 385.93  on 368  degrees of freedom
## AIC: 2317.5
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.856
##           Std. Err.:  0.159
##
##  2 x log-likelihood:  -2297.490
```

```r
boxplot(total_deer_group ~ value, data = deer_data_5)
```




```r
# fitting predicted group values to raster

values_crop <-data.frame(value= c("121","1","5","24","36","61","63","64","152"))

p1<-predict(model_2, newdata=values_crop, se.fit=TRUE, type='response')

py2<- expand.grid(values_crop)
p2<-predict(model_2, newdata=py2, se.fit=TRUE, type='response')
predicted_group_size <- data.frame(py2, p2)

predicted_group_size
```

```
##   value       fit    se.fit residual.scale
## 1   121  3.214286 0.5599824              1
## 2     1 10.285714 0.7519229              1
## 3     5  8.757576 1.2318447              1
## 4    24 11.750000 2.3201354              1
## 5    36  7.509804 0.8619706              1
## 6    61  8.967742 1.2987916              1
## 7    63  6.157303 0.5465094              1
## 8    64  6.600000 2.4522272              1
## 9   152  6.600000 2.4522272              1
```

```r
#plotting new raster with predicted values
```

```
names(cdl_MICH_SA_masked) = "value"

group_raster<- predict(cdl_MICH_SA_masked, model_2, type="response")

plot(group_raster,
     col= heat.colors(8, rev=TRUE),
     legend.args=list(text='Group Size', side=4, font=2, line=2.5, cex=0.8))
scalebar(50000, type="bar", divs = 2, label = c(0,25,50),below="km")
north("topleft", angle = 350)
```