Statistical Modelling I

HES 505 Fall 2024: Session 21

Carolyn Koehn

Objectives

By the end of today you should be able to:

- Describe and implement overlay analyses
- Extend overlay analysis to statistical modeling
- Generate spatial predictions from statistical models

Overlay Analyses

Overlays

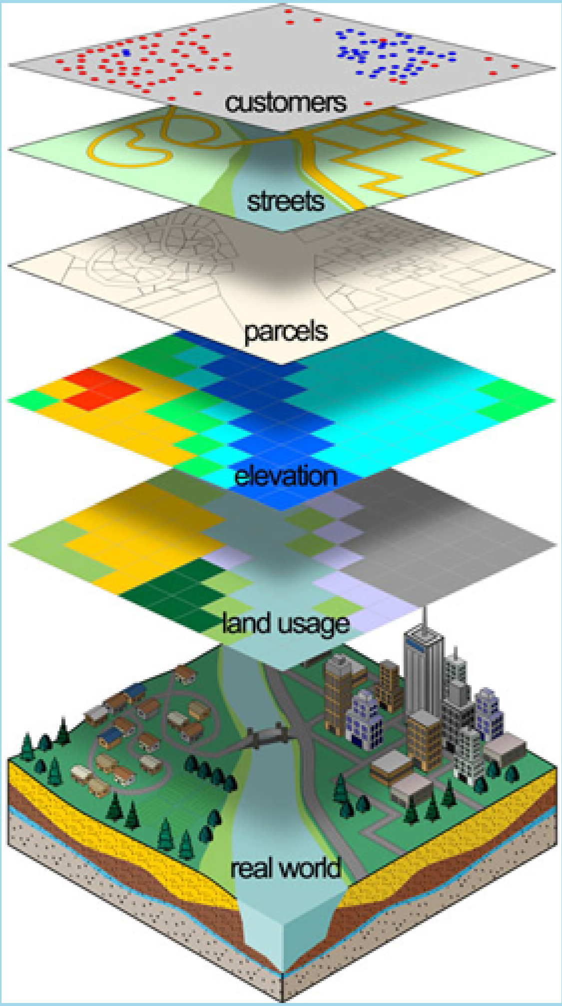
- Methods for identifying optimal site selection or suitability
- Apply a common scale to diverse or dissimilar outputs

Getting Started

1. Define the problem.
2. Break the problem into submodels.
3. Determine significant layers.
4. Reclassify or transform the data within a layer.
5. Add or combine the layers.
6. Verify

Boolean Overlays

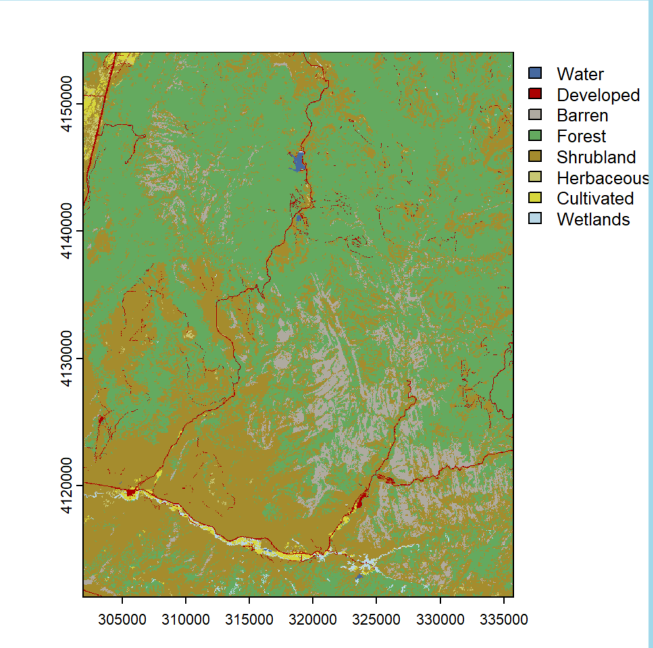
- Successive disqualification of areas
- Series of “yes/no” questions
- “Sieve” mapping



Boolean Overlays

- Reclassifying
- Which types of land are appropriate

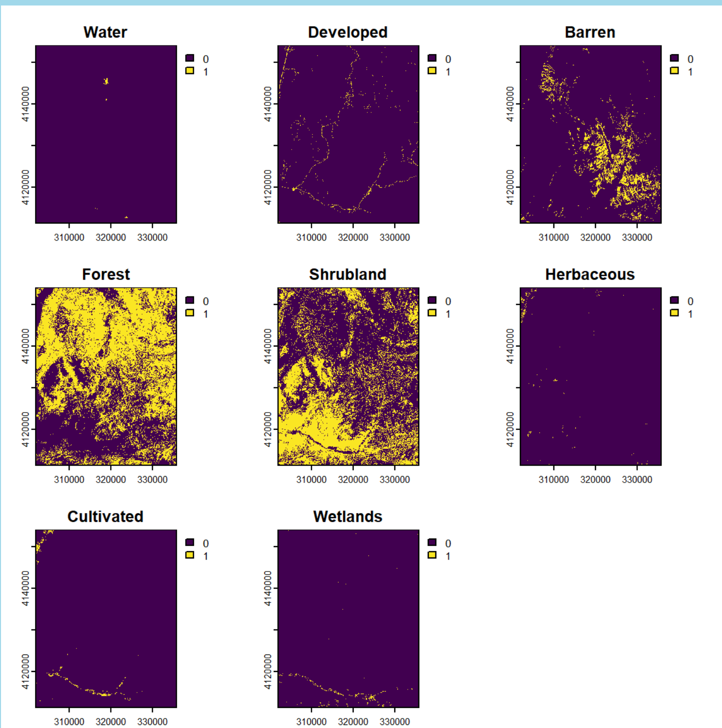
```
nlcd <-  rast(system.file("raster/nlcd.tif", package = "spDataLarge"))
plot(nlcd)
```



Boolean Overlays

- Which types of land are appropriate?

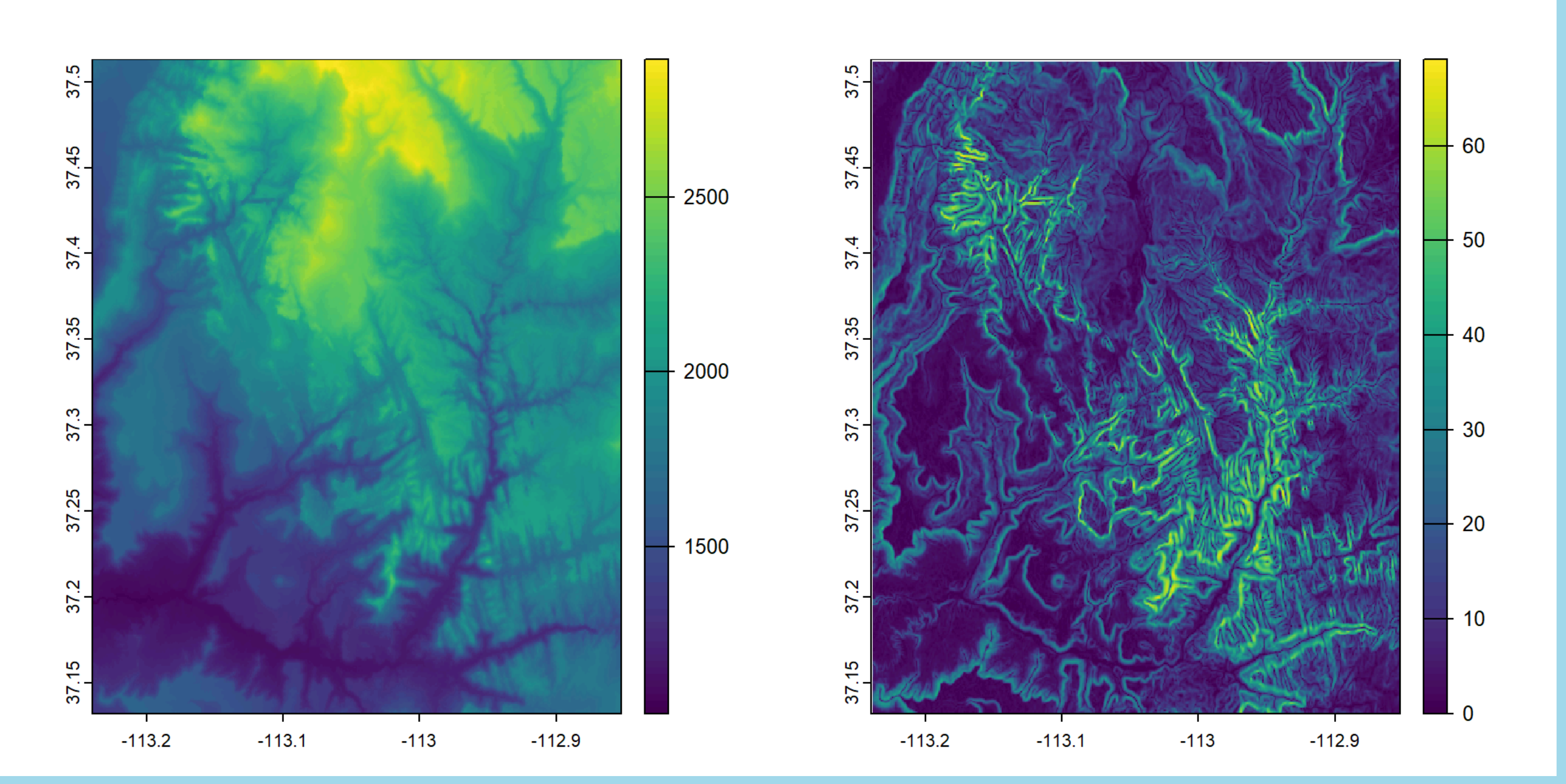
```
nlcd.segments <- segregate(nlcd)
names(nlcd.segments) <- levels(nlcd)[[1]][-1,2]
plot(nlcd.segments)
```



Boolean Overlays

- Which types of land are appropriate?

```
srtm <- rast(system.file("raster/srtm.tif", package = "spDataLarge"))
slope <- terrain(srtm, v = "slope")
```

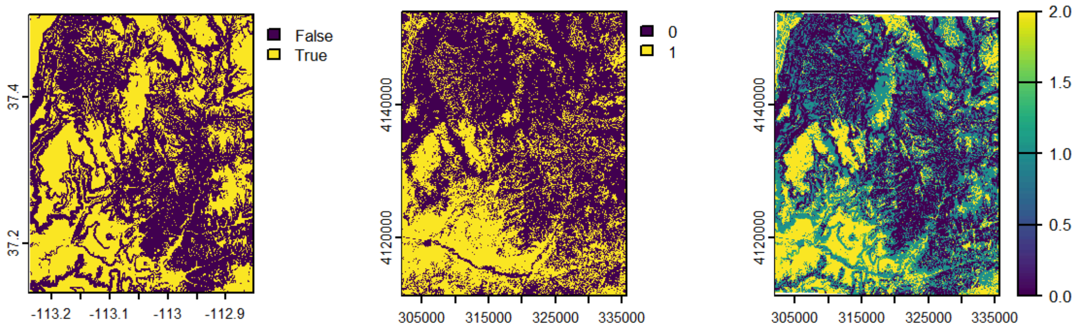


Boolean Overlays

- Make sure data is aligned!

```
suit.slope <- slope < 10
suit.landcov <- nlcd.segments["Shrubland"]
suit.slope.match <- project(suit.slope, suit.landcov)
suit <- suit.slope.match + suit.landcov
```

Boolean Overlays



Challenges with Boolean Overlays

1. Assume relationships are really Boolean
2. No measurement error
3. Categorical measurements are known exactly
4. Boundaries are well-represented

A more general approach

- Define a *favorability* metric

$$F(\mathbf{s}) = \prod_{M=1}^{m} X_m(\mathbf{s})$$

- Treat $F(\mathbf{s})$ as binary
- Then $F(\mathbf{s}) = 1$ if all inputs ($X_m(\mathbf{s})$) are suitable
- Then $F(\mathbf{s}) = 0$ if not

Estimating favorability

$$F(\mathbf{s}) = f(w_1 X_1(\mathbf{s}), w_2 X_2(\mathbf{s}), w_3 X_3(\mathbf{s}), \ldots, w_m X_m(\mathbf{s}))$$

- $F(\mathbf{s})$ does not have to be binary (could be ordinal or continuous)
- $X_m(\mathbf{s})$ could also be extended beyond simply 'suitable/not suitable'
- Adding weights allows incorporation of relative importance
- Other functions for combining inputs ($X_m(\mathbf{s})$)

Weighted Linear Combinations

$$F(\mathbf{s}) = \frac{\sum_{i=1}^{m} w_i X_i(\mathbf{s})}{\sum_{i=1}^{m} w_i}$$

- $F(s)$ is now an index based on the values of $X_m(\mathbf{s})$
- w_i can incorporate weights of evidence, uncertainty, or different participant preferences
- Dividing by $\sum_{i=1}^{m} w_i$ normalizes by the sum of weights

Model-driven overlay

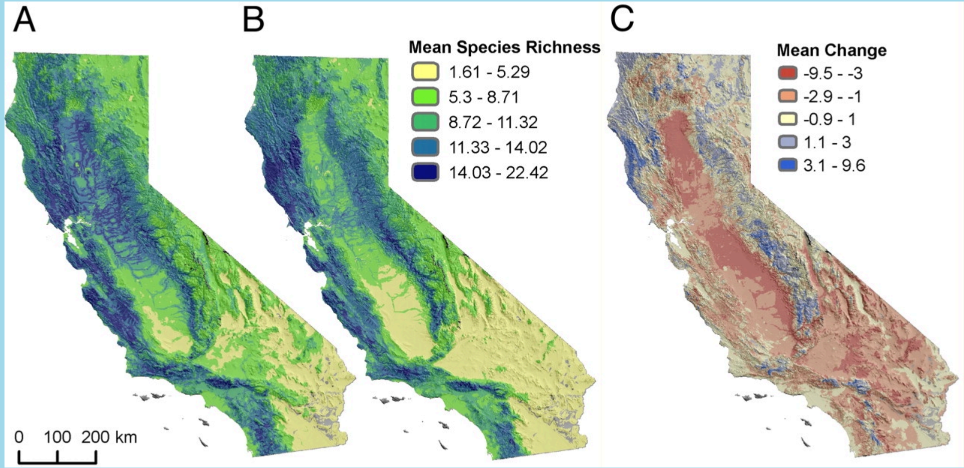
$$F(\mathbf{s}) = w_0 + \sum_{i=1}^{m} w_i X_i(\mathbf{s}) + \epsilon$$

- If we estimate w_i using data, we specify $F(s)$ as the outcome of regression
- When $F(s)$ is binary $\rightarrow$ logistic regression
- When $F(s)$ is continuous $\rightarrow$ linear (gamma) regression
- When $F(s)$ is discrete $\rightarrow$ Poisson regression
- Assumptions about ϵ matter!!

Logistic Regression and Distribution Models

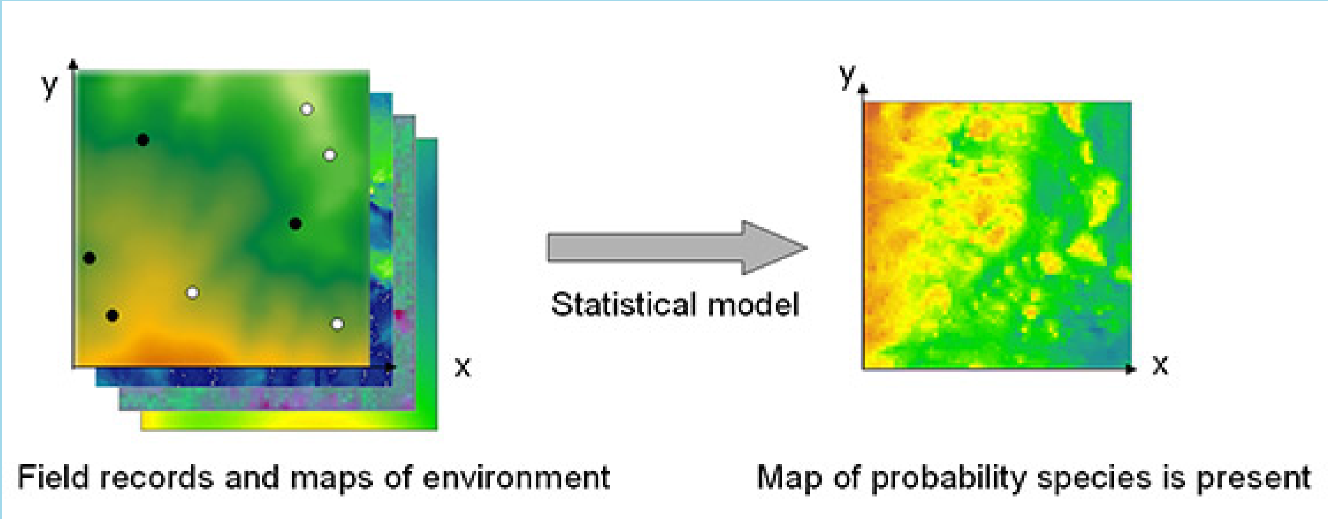
Why do we create distribution models?

- To identify important correlations between predictors and the occurrence of an event
- Generate maps of the 'range' or 'niche' of events
- Understand spatial patterns of event co-occurrence
- Forecast changes in event distributions



From Wiens et al. 2009

General analysis situation



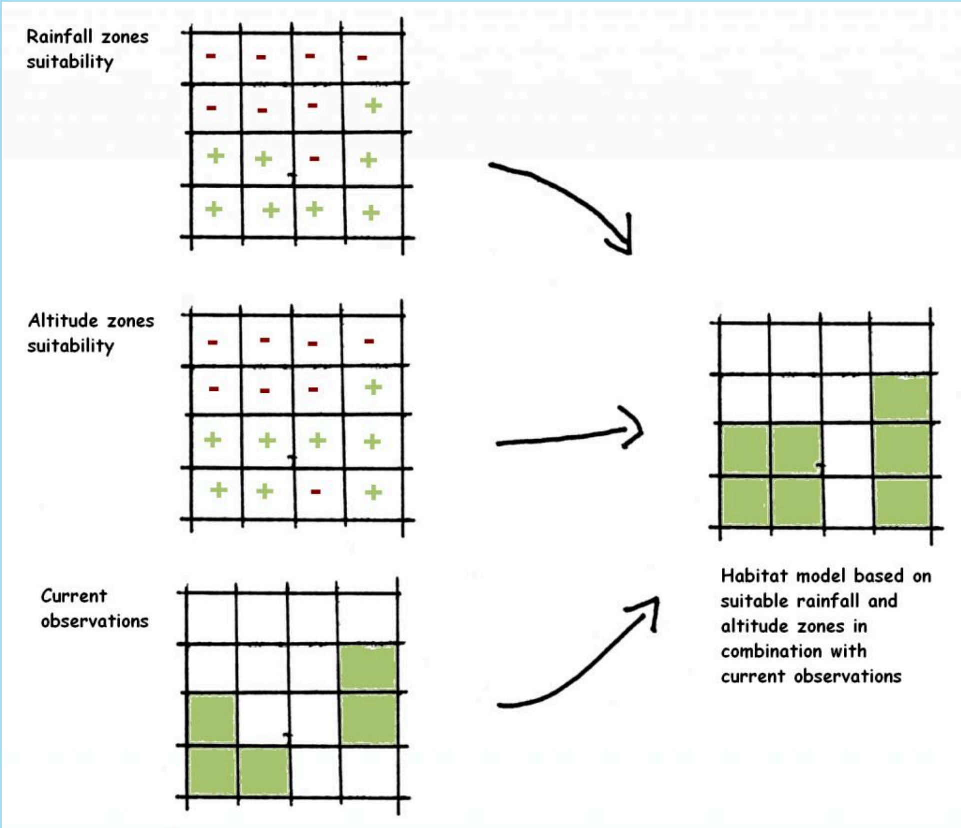
From Long

- Spatially referenced locations of events $(\mathbf{y})$ sampled from the study extent
- A matrix of predictors $(\mathbf{X})$ that can be assigned to each event based on spatial location

> **Goal**: Estimate the probability of occurrence of events across unsampled regions of the study area based on correlations with predictors

Modeling Presence-Absence Data

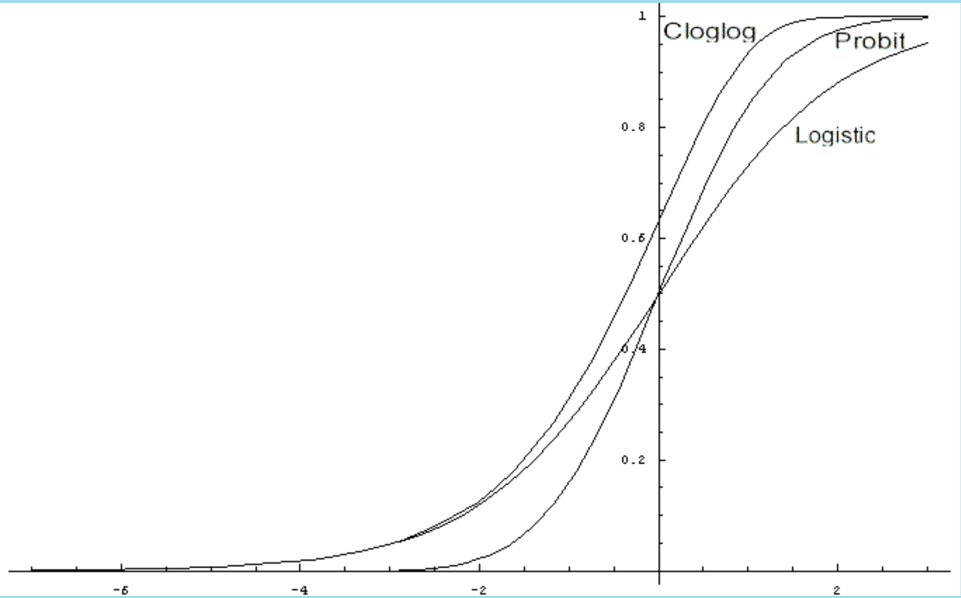
- Random or systematic sample of the study region
- The presence (or absence) of the event is recorded for each point
- Hypothesized predictors of occurrence are measured (or extracted) at each point



Logistic regression

- We can model favorability as the **probability** of occurrence using a logistic regression
- A *link* function maps the linear predictor $(\mathbf{x_i}'\beta + \alpha)$ onto the support (0-1) for probabilities
- Estimates of β can then be used to generate 'wall-to-wall' spatial predictions

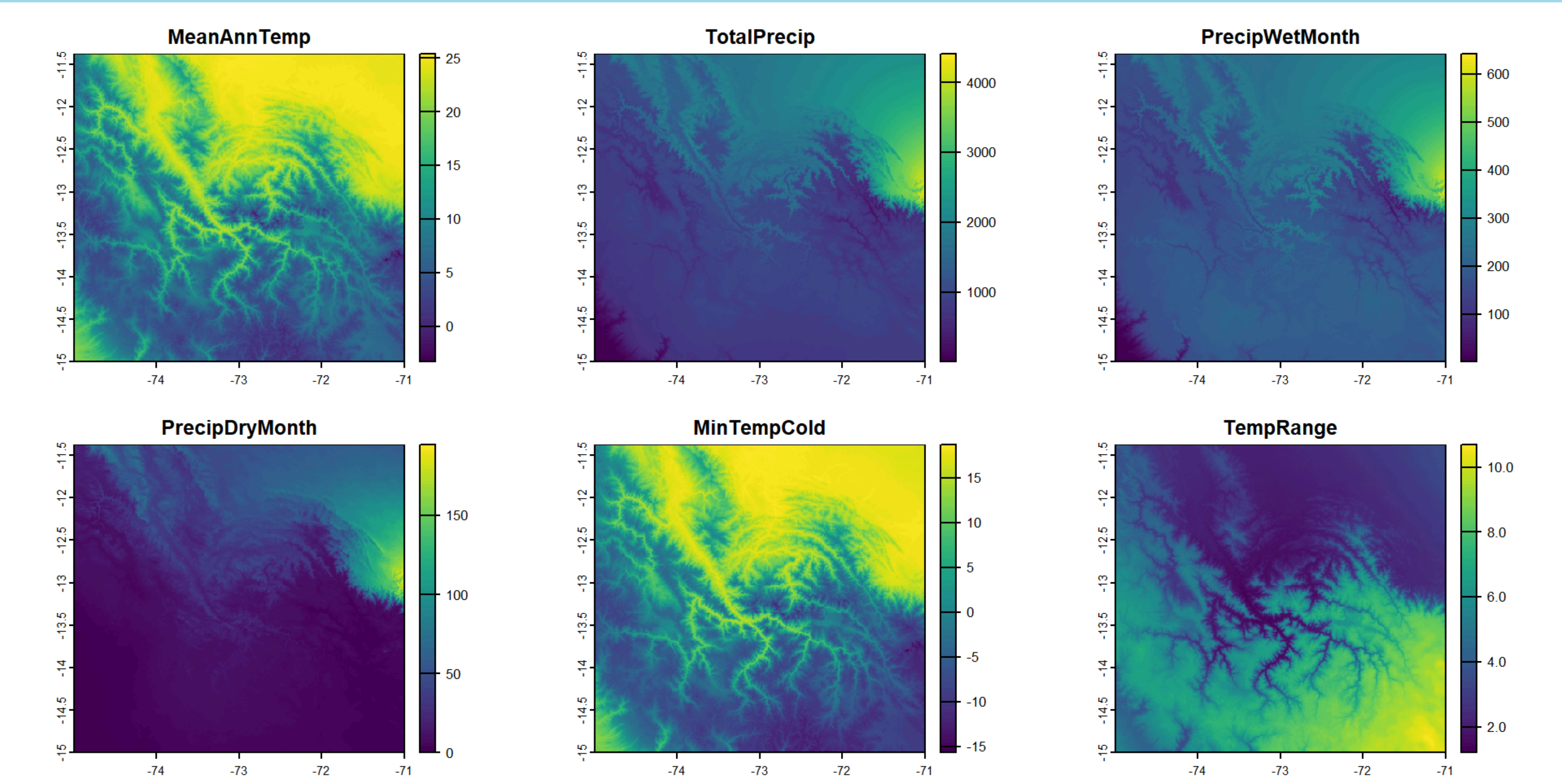
$$y_i \sim \mathrm{Bern}(p_i)$$
$$\mathrm{link}(p_i) = \mathbf{x_i}'\beta + \alpha$$



From Mendoza

An Example

Inputs from the `dismo` package



An Example

The sample data

An Example

Building our dataframe

```r
pts.df <- terra::extract(pred.stack, vect(pres.abs), df=TRUE)
head(pts.df)
```

```
  ID MeanAnnTemp TotalPrecip PrecipWetMonth PrecipDryMonth MinTempCold
1  1    4.975000         760            151              3        -5.9
2  2    6.391667         830            146             10        -4.1
3  3   11.816667         845            181              7         1.3
4  4   11.241667         694            150              4        -0.4
5  5    6.875000         909            199              4        -6.4
6  6    5.750000        1002            190             12        -5.4
  TempRange
1       6.3
2       5.7
3       4.6
4       6.6
5       8.3
6       6.4
```

An Example

Building our dataframe

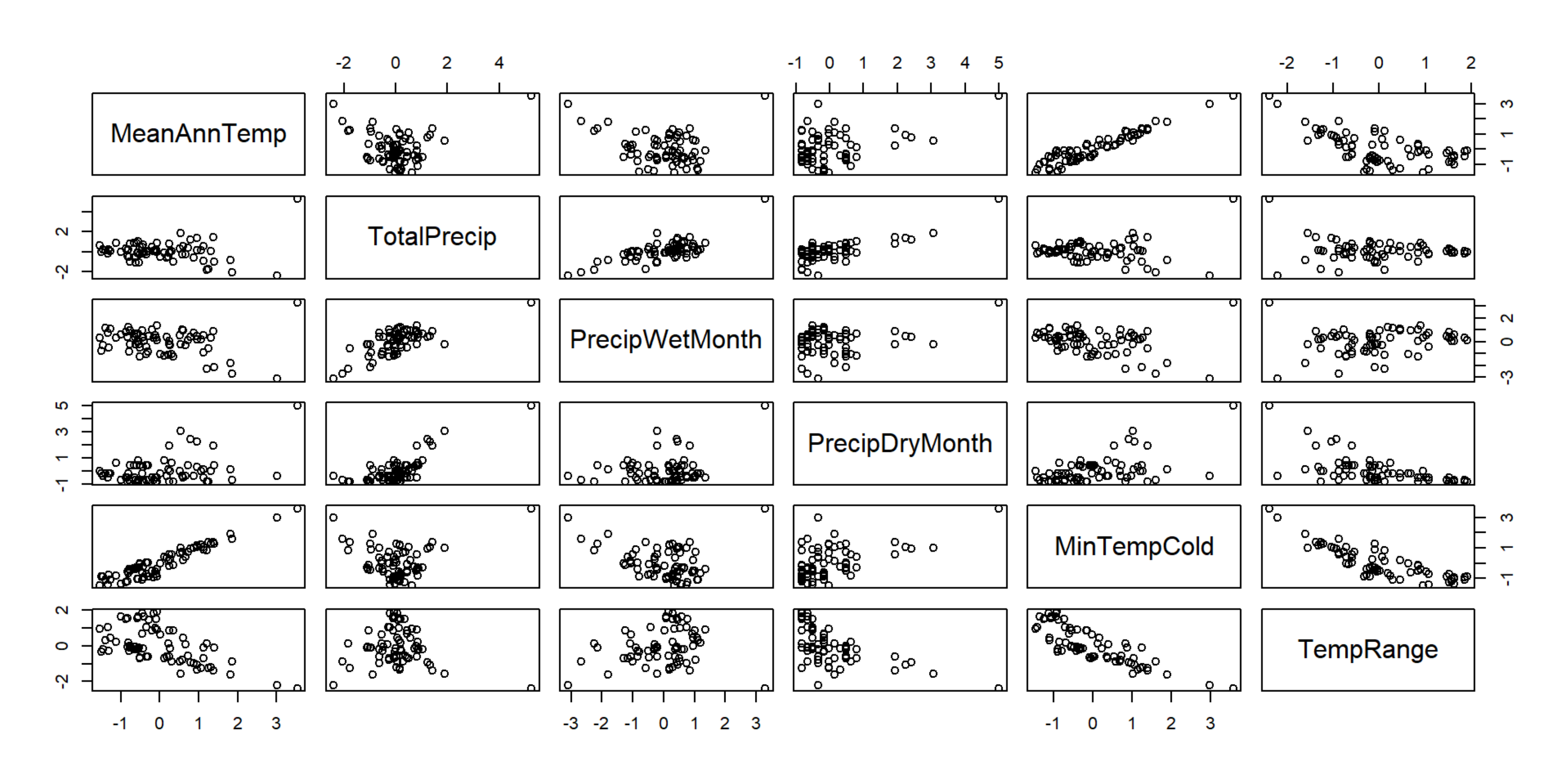
```
pts.df[,2:7] <- scale(pts.df[,2:7])
summary(pts.df)
```

```
       ID         MeanAnnTemp        TotalPrecip        PrecipWetMonth   
 Min.   : 1.00   Min.   :-1.5372   Min.   :-2.40213   Min.   :-3.1103  
 1st Qu.:19.25   1st Qu.:-0.6294   1st Qu.:-0.51638   1st Qu.:-0.4595  
 Median :37.50   Median :-0.2416   Median : 0.02767   Median : 0.1910  
 Mean   :37.50   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000  
 3rd Qu.:55.75   3rd Qu.: 0.5984   3rd Qu.: 0.41766   3rd Qu.: 0.5697  
 Max.   :74.00   Max.   : 3.5301   Max.   : 5.24357   Max.   : 3.2982  
 PrecipDryMonth     MinTempCold        TempRange       
 Min.   :-0.8282   Min.   :-1.4490   Min.   :-2.38574  
 1st Qu.:-0.6660   1st Qu.:-0.7159   1st Qu.:-0.69892  
 Median :-0.2607   Median :-0.2996   Median :-0.09947  
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000  
 3rd Qu.: 0.4283   3rd Qu.: 0.7033   3rd Qu.: 0.80668  
 Max.   : 5.0085   Max.   : 3.5710   Max.   : 1.90799  
```

An Example

Looking at correlations

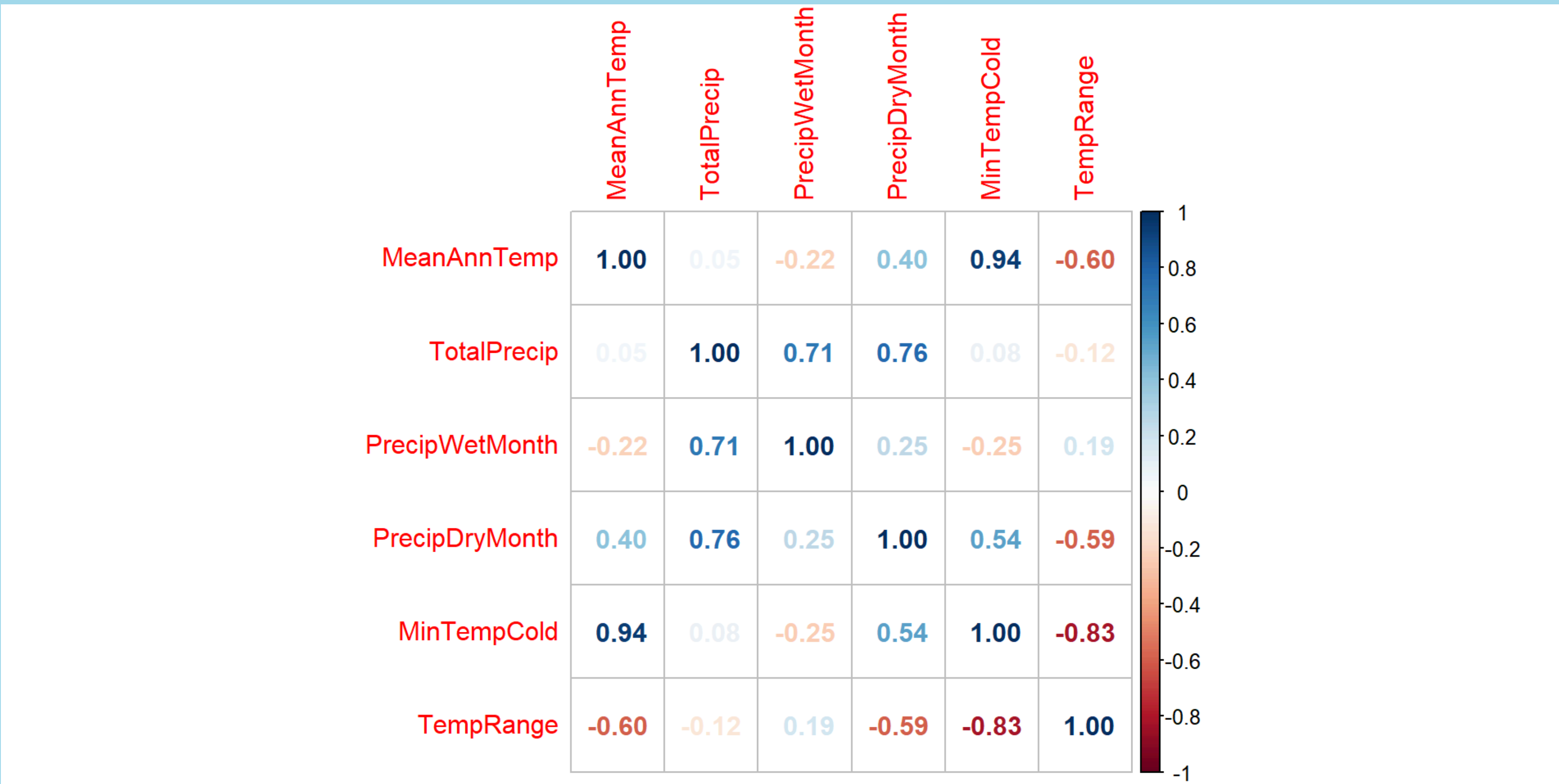
```
pairs(pts.df[,2:7])
```



An Example

Looking at correlations

```
corrplot(cor(pts.df[,2:7]), method = "number")
```



An Example

Fitting some models

```
pts.df <- cbind(pts.df, pres.abs$y)
colnames(pts.df)[8] <- "y"
logistic.global <- glm(y~., family=binomial(link="logit"), data=pts.df[,2:8
logistic.simple <- glm(y ~ MeanAnnTemp + TotalPrecip, family=binomial(link=
logistic.rich <- glm(y ~ MeanAnnTemp + PrecipWetMonth + PrecipDryMonth, fam
```

An Example

Checking out the results

```r
summary(logistic.global)
```

```
Call:
glm(formula = y ~ ., family = binomial(link = "logit"), data = pts.df[,
    2:8])

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.08264    0.25052  -0.330    0.741
MeanAnnTemp    -0.62712    3.36175  -0.187    0.852
TotalPrecip    -1.41121    0.86351  -1.634    0.102
PrecipWetMonth  0.71202    0.50360   1.414    0.157
PrecipDryMonth  1.06540    0.80825   1.318    0.187
MinTempCold     2.02792    4.88923   0.415    0.678
TempRange       1.84210    1.98680   0.927    0.354

(Dispersion parameter for binomial family taken to be 1)
```

An Example

Checking out the results

```r
summary(logistic.simple)
```

```
Call:
glm(formula = y ~ MeanAnnTemp + TotalPrecip, family = binomial(link =
"logit"),
    data = pts.df[, 2:8])

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.05085    0.23676  -0.215    0.830
MeanAnnTemp  0.35247    0.24825   1.420    0.156
TotalPrecip -0.18792    0.24423  -0.769    0.442

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 102.532  on 73  degrees of freedom
```

An Example

Checking out the results

```
summary(logistic.rich)
```

```
Call:
glm(formula = y ~ MeanAnnTemp + PrecipWetMonth + PrecipDryMonth,
    family = binomial(link = "logit"), data = pts.df[, 2:8])

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -0.05522    0.23813  -0.232   0.8166
MeanAnnTemp     0.51136    0.28979   1.765   0.0776 .
PrecipWetMonth  0.15989    0.27013   0.592   0.5539
PrecipDryMonth -0.33762    0.28827  -1.171   0.2415
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

An Example

Comparing models

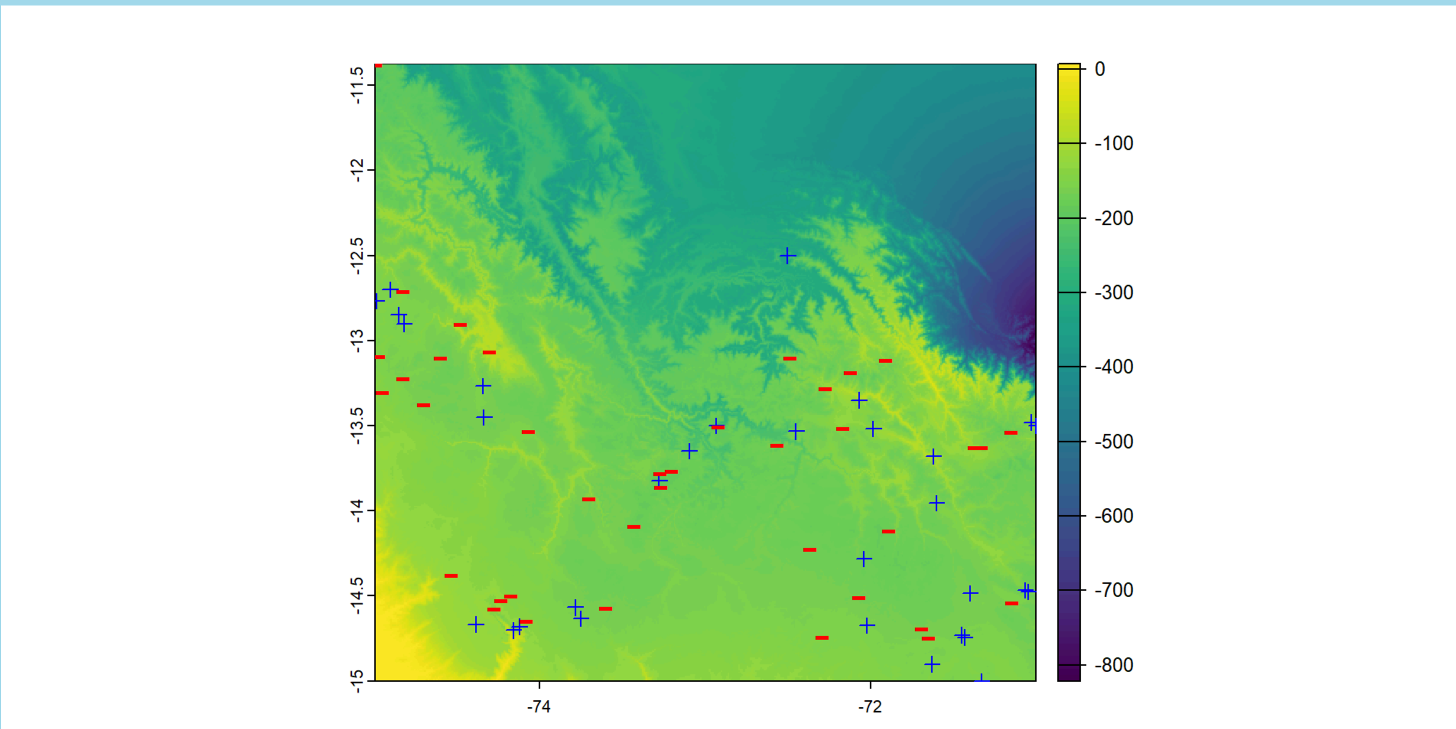
```
AIC(logistic.global, logistic.simple, logistic.rich)
```

```
                df      AIC
logistic.global  7 106.9931
logistic.simple  3 105.9526
logistic.rich    4 107.1040
```

An Example

Generating predictions

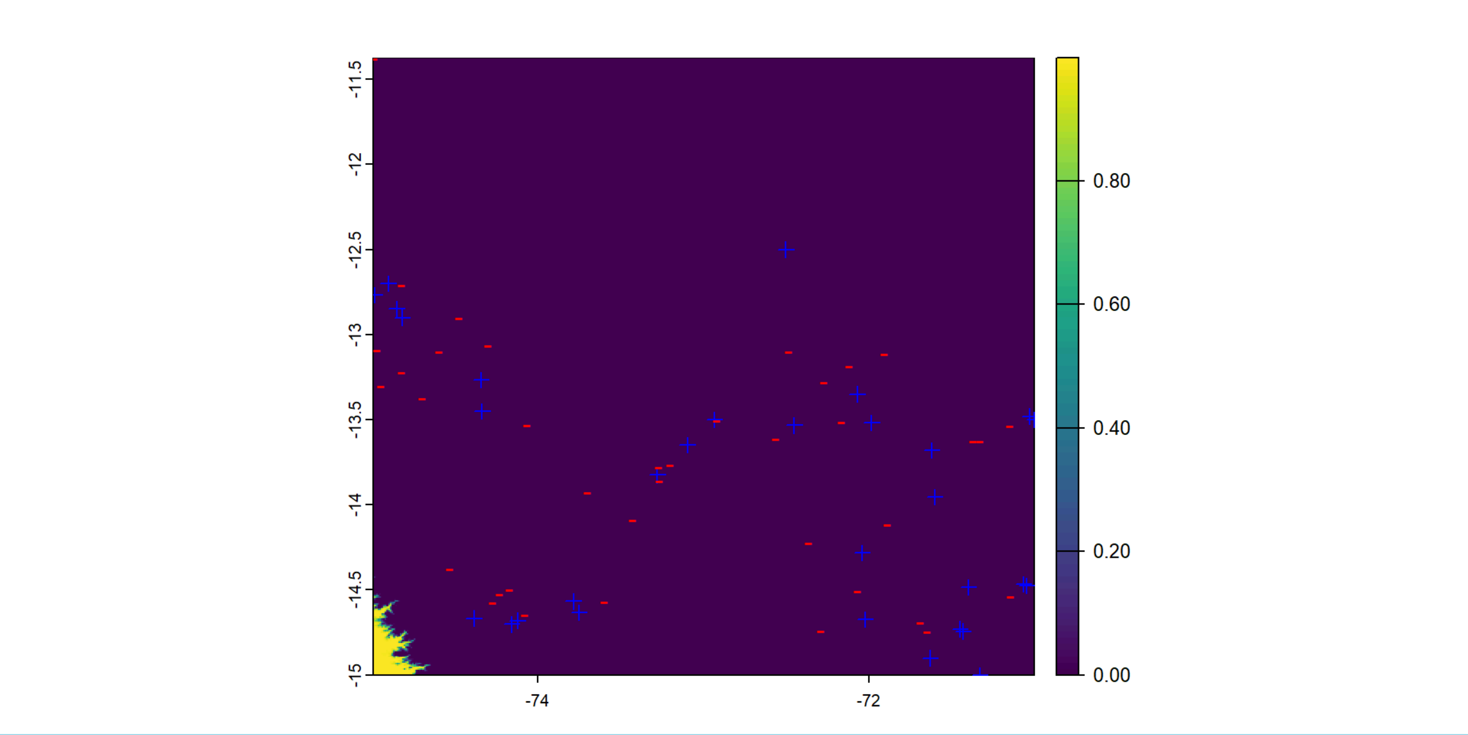
```
preds <- predict(object=pred.stack, model=logistic.simple)
plot(preds)
plot(pres.pts$geometry, add=TRUE, pch=3, col="blue")
plot(abs.pts$geometry, add=TRUE, pch ="-", col="red", cex=2)
```



An Example

Generating predictions

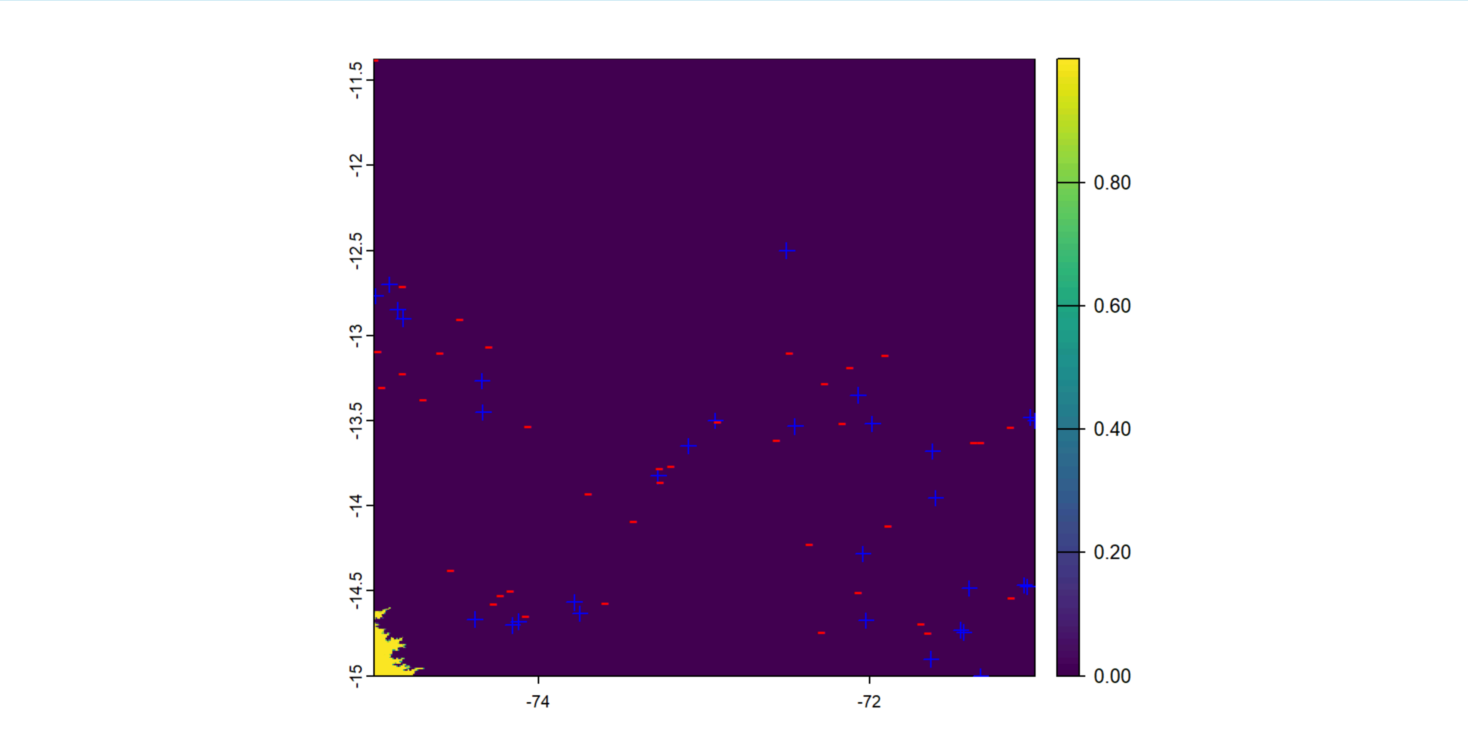
```r
preds <- predict(object=pred.stack, model=logistic.simple, type="response")
plot(preds)
plot(pres.pts$geometry, add=TRUE, pch=3, col="blue")
plot(abs.pts$geometry, add=TRUE, pch ="-", col="red")
```



An Example

Generating predictions

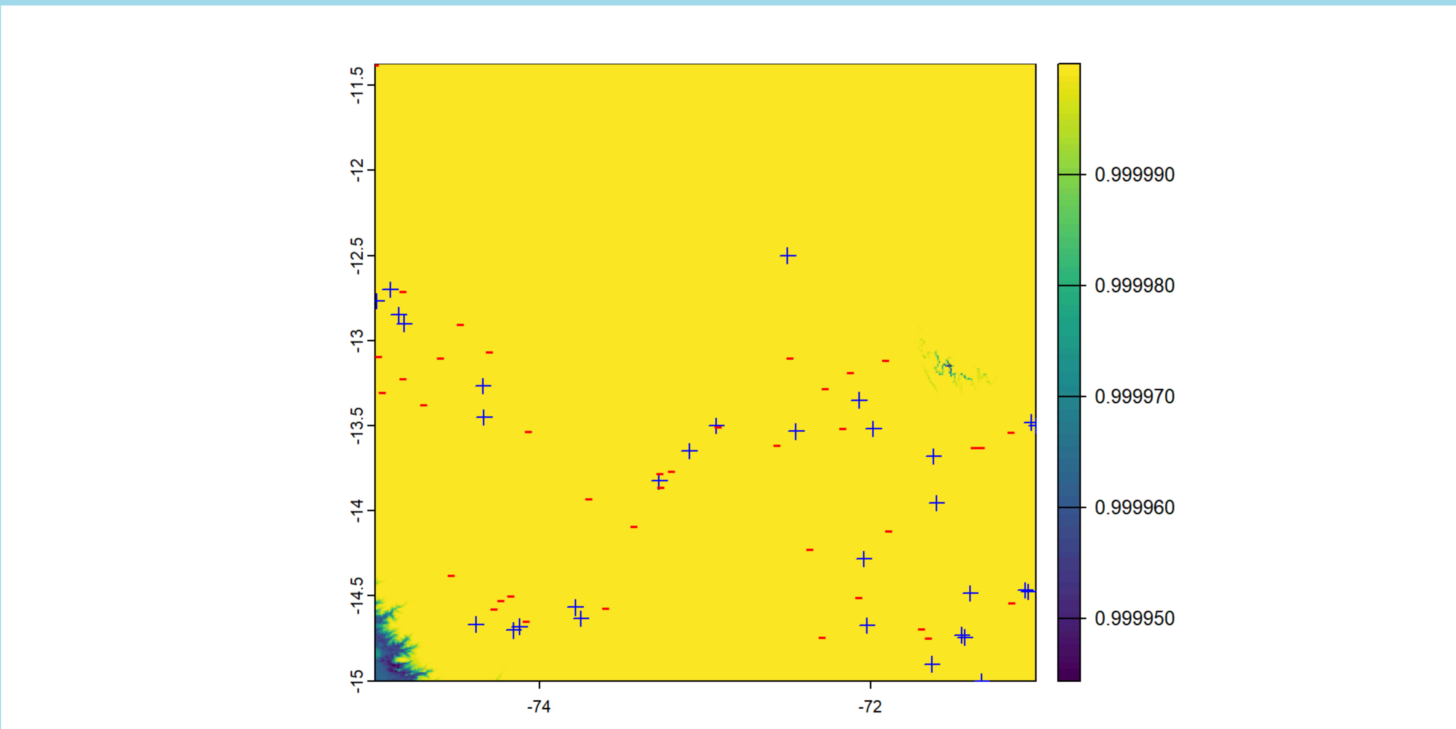
```
preds <- predict(object=pred.stack, model=logistic.global, type="response")
plot(preds)
plot(pres.pts$geometry, add=TRUE, pch=3, col="blue")
plot(abs.pts$geometry, add=TRUE, pch ="-", col="red")
```



An Example

Generating predictions

```
preds <- predict(object=pred.stack, model=logistic.rich, type="response")
plot(preds)
plot(pres.pts$geometry, add=TRUE, pch=3, col="blue")
plot(abs.pts$geometry, add=TRUE, pch ="-", col="red")
```



Key assumptions of logistic regression

- Dependent variable must be binary
- Observations must be independent (important for spatial analyses)
- Predictors should not be collinear
- Predictors should be linearly related to the log-odds
- **Sample Size**