

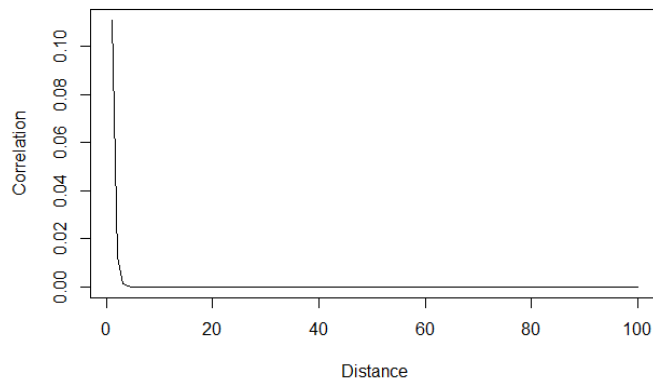
Step 1: Dragons

Step 2: Number of castles

Step 3: R code

- A) Dragons like to raid castles and feast on the humans that live around them, so more castles would predict more dragons because there is a large food source.
- B) My Phi value is 2.2 because castles are less spatially correlated because they are typically built out in open land not close to other castles. However, sometimes the castles cluster together if there is an abundance of resources in that area.

```
> rmvn <- function(n, mu = 0, V = matrix(1)) {  
+   p <- length(mu)  
+   if (any(is.na(match(dim(V), p))))  
+     stop("Dimension problem!")  
+   D <- chol(V)  
+   t(matrix(rnorm(n * p), ncol = p) %*% D + rep(mu, rep(n, p)))  
+ }  
> simgrid <- expand.grid(1:50, 1:50)  
> n <- nrow(simgrid)  
> distance <- as.matrix(dist(simgrid))  
> phi = 2.2  
> plot(1:100, exp(-phi * 1:100), type = "l", xlab = "Distance", ylab = "Correlation")
```

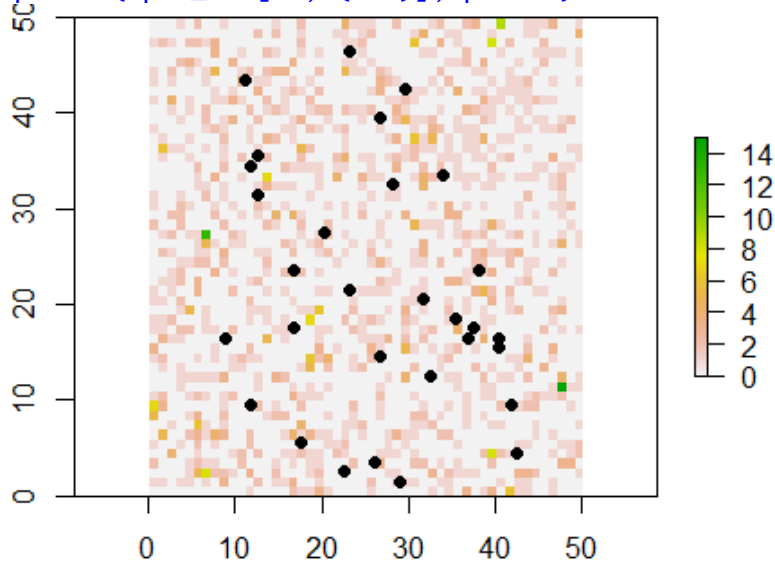


Step 4:

I chose to only sample 30 locations because hiking to all of these places is very difficult, time consuming and money intensive. Also dragons are large and only so many can live on the landscape and so 30 samples locations should decently represent the dragon population.

```
> X <- rpois(n, lambda=exp(-1+rmvn(1, rep(0, n), exp(-phi * distance))))  
> Xraster <- rasterFromXYZ(cbind(simgrid[, 1:2] - 0.5, X))  
> plot(Xraster)  
> spat_dat=rasterToPoints(Xraster)  
> head(spat_dat)  
      x    y X  
[1,] 0.5 49.5 0  
[2,] 1.5 49.5 0  
[3,] 2.5 49.5 0  
[4,] 3.5 49.5 1  
[5,] 4.5 49.5 0  
[6,] 5.5 49.5 2
```

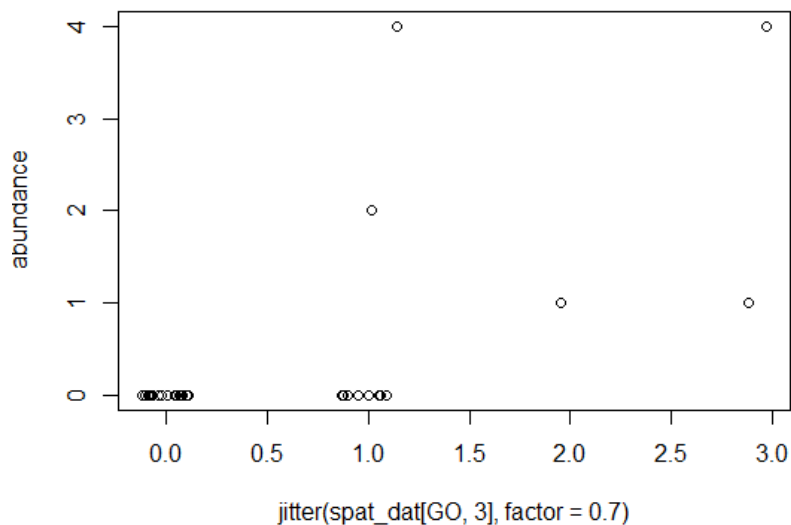
```
> GO=sample(x=c(1:nrow(spat_dat)),size=30)
> points(spat_dat[GO,c(1:2)], pch=19)
```



Step 5:

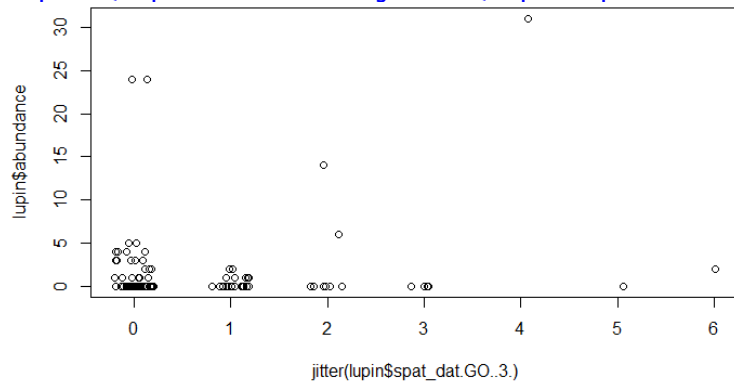
- A) I would use a hurdle model because dragons are rare species and dispersed over a large area, and so many sample sites visited may yield zero dragons.
- B) We are assuming that abundance is either zero if PA is zero or that abundance is equal to the count data when PA is 1.
- C) Plot

```
> presence_intercept=-3
> presence_slope=2.7
> PA=rbinom(30,plogis(presence_intercept+spat_dat[GO,3]*presence_slope),
+         size=1)
> count_intercept=0.2
> count_slope=0.33
> over_dispersion=0.9
> abundance=PA*rnbinom(30,mu=
+ exp(count_intercept+count_slope*spat_dat[GO,3]),
+ size=over_dispersion)
> plot(abundance~jitter(spat_dat[GO,3], factor=0.7))
```



Step 6: Looking at Anna's Data

```
> lupin<-read.csv("R/lupin_&pollinators.csv")
> str(lupin)
'data.frame': 100 obs. of 3 variables:
 $ X          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ spat_dat.GO..3.: int  1 0 3 3 0 0 0 0 1 1 ...
 $ abundance   : int  2 0 0 0 0 0 0 3 1 0 ...
> plot(lupin$abundance~jitter(lupin$spat_dat.GO..3.))
```



PA True Slope: 2

PA True Intercept: 2

```
> PA1=ifelse(lupin$abundance>0,1,0)
> glm(PA1~lupin$spat_dat.GO..3., family="binomial")
Call: glm(formula = PA1 ~ lupin$spat_dat.GO..3., family = "binomial")
Coefficients:
```

```
(Intercept)  lupin$spat_dat.GO..3.
-0.5934      -0.1031
```

Degrees of Freedom: 99 Total (i.e. Null); 98 Residual

Null Deviance: 128.2

Residual Deviance: 127.9 AIC: 131.9

```
> plogis(-0.59)
```

```
[1] 0.3566349
```

Count True Slope: 0.2

Count True Intercept: 0.1

```
> abundance2<-lupin$abundance[which(lupin$abundance>0)]
> factor<-lupin$spat_dat.GO..3.[which(lupin$abundance>0)]
> glm(abundance2~factor, family="poisson")
Call: glm(formula = abundance2 ~ factor, family = "poisson")
Coefficients:
```

```
(Intercept)      factor
1.3933         0.2142
```

Degrees of Freedom: 33 Total (i.e. Null); 32 Residual

Null Deviance: 223.4

Residual Deviance: 203.3 AIC: 305.4

```
> exp(1.39)
```

```
[1] 4.01485
```

**In order to assess if dragons are spatially auto correlated I would obtain phi by looking at dragon counts by location.

