Structural independence in the logistic model

Ben Weinstein

Thursday, February 12, 2015

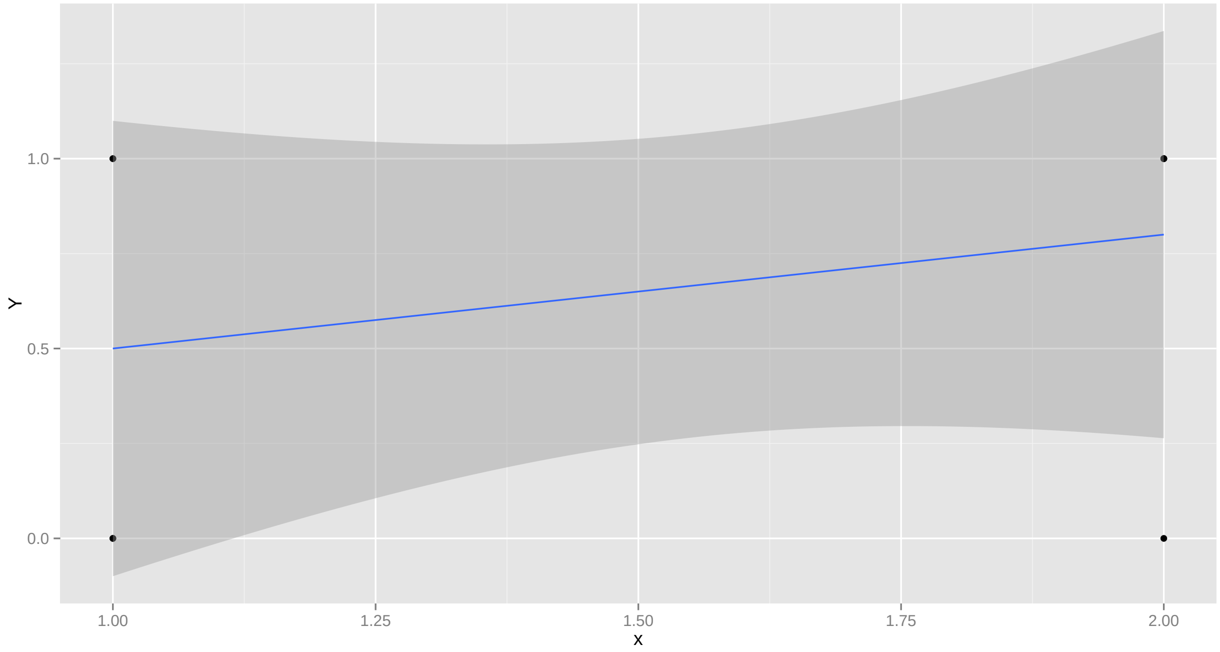
```
library(knitr)
library(pez)
library(ape)
library(dplyr)
library(ggplot2)
library(reshape2)
library(boot)
library(vegan)
library(RColorBrewer)
library(gridExtra)
library(scales)
library(stringr)
library(picante)
library(foreach)
library(doSNOW)
library(lme4)
opts_chunk$set(warning=FALSE,message=FALSE,echo=TRUE,eval=T)
opts_chunk$set(cache=TRUE, cache.path = 'Independence_cache/', fig.path='figure/',fig.width=11,fig.heig

load("Independence.Rdata")
```

Problem Statement

Suppose there are three species, 1, 2, and 3, with the distance between 1 and 2 = 1, and between 3 and

```
dat <- data.frame(sp=c(1,2,3,1,2,3,1,2,3), site=c(1,1,1,2,2,2,3,3,3), Y=c(0,1,1,1,0,1,1,1,0), x=c(1,2,2
```

```
ggplot(dat,aes(x=x,y=Y)) + geom_point() + geom_smooth(method="lm")
```



```
glm(formula = Y ~ x, family = "binomial", data = dat)
```

```
##
## Call:  glm(formula = Y ~ x, family = "binomial", data = dat)
##
## Coefficients:
## (Intercept)            x
##      -1.386        1.386
##
## Degrees of Freedom: 8 Total (i.e. Null);  7 Residual
## Null Deviance:       11.46
## Residual Deviance: 10.55     AIC: 14.55
```

So this gives a positive relationship between distance and presence, even though the presence of species is random under the constraint that there are two species in each site. Obviously, this isn't a desirable property.

The reason that you get this is that the values of Y and x are structurally intertwined, so the values of Y aren't independent.

I'm wondering if there is a simpler way to do this

Aim

To address Tony's concern, i want to evaluate:

- How this potential non-independence could effect results
- What is the range of covariates we could expect by random chance

What does the relationship look like with increasing species and sites

Let's generalize this structure and see how this relationship changes with more sites and species, does it flatten out?

- Each species draw is bernoulli trial
- There is no min or max richness

Simulate phylogeny * Start with a balanced tree, we can check other topologies as well

Build a function to compute logistic regression for any given number of species in a phylogeny and sites

- Occupancy is drawn from a bernoulli trial

```r
simL<-function(species,sites){

#Simulate a balanced phylogeny
trx<-compute.brlen(stree(species,"balanced",tip.label=1:species))

#Cophenetic matrix
ctrx<-cophenetic(trx)

#Create data.frame
sp=rep(1:species,sites)
site=as.vector(sapply(1:sites,function(x) rep(x,species)))
dat<-data.frame(sp,site)

#Presence/absence
dat$Y<-rbinom(sites*species,1,.5)

#Compute co-occurrence
dat<-co_occur(dat,trx,ctrx)
return(dat)}
```

Iterate over a large parameter space

```r
species.space<-rep(2^(3:7))
site.space<-rep(c(20,40,60,80,100),200)
```

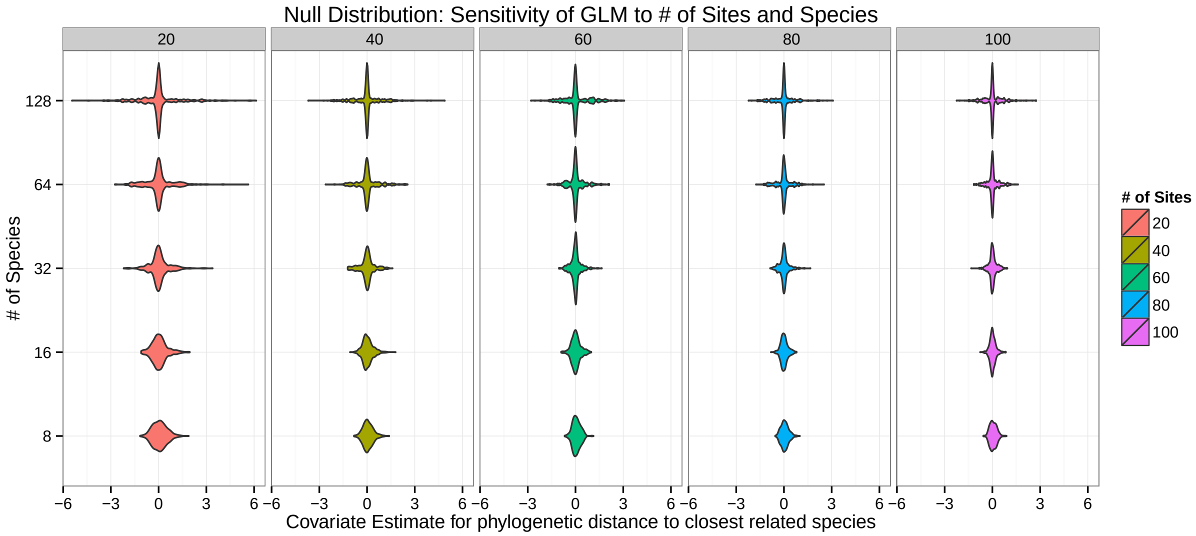
Simulate Data

- For each site and species combination draw random occurrence

```
simdat<-lapply(species.space,function(x){
    lapply(site.space,function(y){
      l<-simL(x,y)
  mod<-glm(data=l,formula = Y ~ x, family = "binomial")
  mode<-summary(mod)$coefficients[,"Estimate"]
  data.frame(Intercept=mode[[1]],x=mode[[2]])
      })
})

#name layers
names(simdat)<-species.space

for(x in 1:length(simdat)){
  names(simdat[[x]])<-site.space
}
```

View regression

```
mdat<-melt(simdat)
colnames(mdat)<-c("variable","Estimate","Sites","Species")

mdat$Sites<-as.numeric(mdat$Sites)
mdat$Species<-as.numeric(mdat$Species)

ggplot(mdat,aes(x=as.factor(Species),y=Estimate,fill=as.factor(Sites))) + geom_violin() + theme_bw() +
```



Conclusion

The relationship between x and y is indeed structured, since a species occurring at a site changes the phylogenetic composition at the site. This relationship is largely the same across number of sites and number of species. This suggestions its the topology of the tree that controls the null distribution of potential covariates. The way forward is then to randomize the tips of the phylogeny by creating random assemblages of the same size and species prevalence and compare the observed glm estimate with the null distribution of covariate estimates and consider how probability of observing our estimate given the topology of our tree.

Observed versus null distribution

Read in data

```
#read in tree
trx<-read.tree("InputData\\hum294.tre")

new<-str_extract(trx$tip.label,"(\\w+).(\\w+)")
#get duplicates
trx<-drop.tip(trx,trx$tip.label[duplicated(new)])

#name tips.
trx$tip.label<-str_extract(trx$tip.label,"(\\w+).(\\w+)")

ctrx<-cophenetic(trx)

#standardize the distances, just to avoid rounding error.
ctrx<-ctrx/max(ctrx)

siteXspp<-read.csv("C:/Users/Ben/Dropbox/Thesis/Pred_Realized/Assemblages/SiteXsppraster.csv",row.names=

dim(siteXspp)
```

```
## [1] 133 200
```

```
print("133 Species at 201 Sites")
```

```
## [1] "133 Species at 201 Sites"
```

```
source("SpeciesOverlapSourceFunctions.R")
```

```
#definite function to randomize assemblage and compute glm
randomCo<-function(){
  r<-randomizeMatrix(siteXspp,"independentswap")
  l<-co_occur(melt(r))
  mod<-glm(data=l,formula = P_A ~ poly(Phylo.Relatedness,2,raw=TRUE), family = "binomial")
  mode<-summary(mod)$coefficients[,"Estimate"]
  data.frame(Intercept=mode[[1]],x=mode[[2]],x2=mode[[3]])
}
```

```
cl<-makeCluster(20,"SOCK")
registerDoSNOW(cl)
out<-foreach(x=1:5000,.packages=c("picante","reshape2")) %dopar% {
  randomCo()
  }

stopCluster(cl)
```

```r
#melt data frame
dat<-melt(out)
#remove last column
dat<-dat[,-3]
```

Observed Relationship

What is our observed relationship between probability of presence and * Fit a glm without a random effect * Fit a mixed model (glmer) with a species level random effect

Since i have to compute 5000 models, its not really feasible to make a null distribution in the bayesian approach, but we can compare our results here to the predicted function at the end.

GLM

```r
l<-co_occur(melt(as.matrix(siteXspp)))

#glm observed
mod<-glm(data=l,formula = as.factor(P_A) ~ poly(Phylo.Relatedness,2,raw=TRUE), family = "binomial")
summary(mod)
```

```
##
## Call:
## glm(formula = as.factor(P_A) ~ poly(Phylo.Relatedness, 2, raw = TRUE),
##     family = "binomial", data = l)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.5090  -0.4778  -0.4059  -0.3463   2.8120
##
## Coefficients:
##                                         Estimate Std. Error z value
## (Intercept)                             -2.19045    0.08878 -24.673
## poly(Phylo.Relatedness, 2, raw = TRUE)1  1.71281    0.39412   4.346
## poly(Phylo.Relatedness, 2, raw = TRUE)2 -3.45675    0.40232  -8.592
##                                         Pr(>|z|)
## (Intercept)                              < 2e-16 ***
## poly(Phylo.Relatedness, 2, raw = TRUE)1 1.39e-05 ***
## poly(Phylo.Relatedness, 2, raw = TRUE)2  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 14773  on 25999  degrees of freedom
## Residual deviance: 14386  on 25997  degrees of freedom
## AIC: 14392
##
## Number of Fisher Scoring iterations: 6
```

```
mode<-summary(mod)$coefficients[,"Estimate"]
observed<-melt(data.frame(Intercept=mode[[1]],x=mode[[2]],x2=mode[[3]]))
confint.glm<-confint(mod)
predict.glm<-data.frame(x=l$Phylo.Relatedness,y=predict(mod))
```

GLMM (1|Species)

```
mod2<-glmer(data=l,formula = as.factor(P_A) ~ poly(Phylo.Relatedness,2,raw=T) + (1|Species), family = "
smod2<-summary(mod2)
smod2
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## as.factor(P_A) ~ poly(Phylo.Relatedness, 2, raw = T) + (1 | Species)
##    Data: l
##
##      AIC      BIC   logLik deviance df.resid
##  12488.9  12521.6  -6240.4  12480.9    25996
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4668 -0.3068 -0.1879 -0.1173 16.9612
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  Species (Intercept) 1.642    1.282
## Number of obs: 26000, groups:  Species, 130
##
## Fixed effects:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           -2.2496     0.1592 -14.131  < 2e-16
## poly(Phylo.Relatedness, 2, raw = T)1   1.2601     0.5061   2.490   0.0128
## poly(Phylo.Relatedness, 2, raw = T)2  -4.1911     0.5138  -8.157 3.44e-16
##
## (Intercept)                          ***
## poly(Phylo.Relatedness, 2, raw = T)1 *
## poly(Phylo.Relatedness, 2, raw = T)2 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##               (Intr) p(P.R,2,r=T)1
## p(P.R,2,r=T)1 -0.638
## p(P.R,2,r=T)2  0.563 -0.966
```

```
coef.glmer<-smod2$coefficients
observed.glmer<-melt(data.frame(Intercept=coef.glmer[1,1],x=coef.glmer[2,1],x2=coef.glmer[3,1]))
confint.glmer<-confint(mod2)
predict.glmer<-data.frame(x=l$Phylo.Relatedness,y=predict(mod2))
```

```
observed.bayes<-melt(data.frame(Intercept=-1.96,x=6.64,x2=-11.69))

#bind results
coef.dat<-melt(list(GLM=observed,GLMM=observed.glmer,Bayesian=observed.bayes))
```

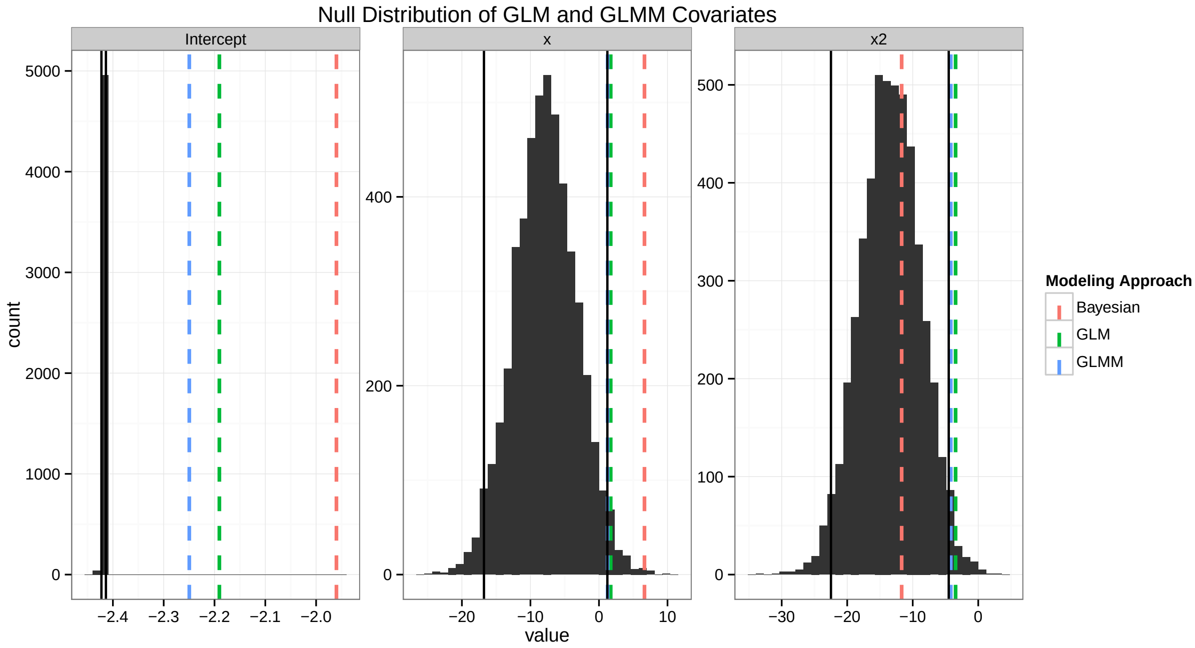
Interpretation:

Both the glm and glmer results suggest that the polynomial 'fixed' effect is signficant. Normally i'd be concerned with high corerlation between fixed effects, but it makes sense how the linear and polynomial terms are linked.

Visualize null distribution versus observed value

- View the mean estimates against the null distribution

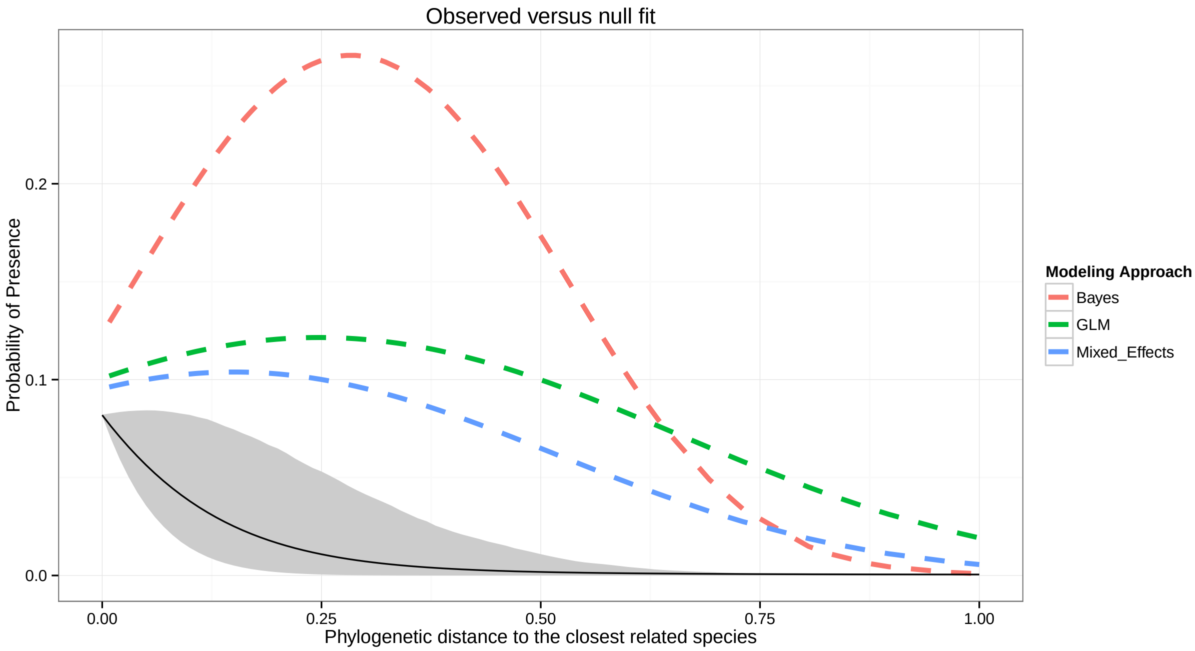
```
quants<-group_by(dat,variable) %>% summarize(lower=quantile(value,0.025),upper=quantile(value,0.975))

ggplot(dat,aes(x=value)) + geom_histogram() + facet_wrap(~variable,scales="free") + theme_bw() + ggtitl
```



The thin black line are the central 95th quantiles of the null distribution for each parameter based on 5000 glm models for randomized data. We compare this null distrubtion to the observed mean covariate estimates for glm glmm, and bayesian models. From this figure I infer that the the probability of getting our result by the intertwined structure of the model is very low. This is because to get our result we need to get both the x and xˆ2 terms, they are not independent, so the joint probability of getting our result is even more remote than just $\alpha = 0.05$ would suggest.

Simulated Trajectories

```
#define trajectory function
trajF<-function(intercept,linear,polynomial,x){
  p<-inv.logit(intercept + linear * x  + polynomial * x^2)
  return(p)
}

s<-seq(0,1,0.01)

#Simulated trajectories
ynew<-lapply(out,function(x){
  y<-trajF(x$Intercept,x$x,x$x2,s)
  data.frame(s,y)
  })

ynew<-rbind_all(ynew)

conf<-group_by(ynew,s) %>% summarise(mean=mean(y),upper=quantile(y,0.975),lower=quantile(y,0.025))


#Observed glm
yobs.glm<-data.frame(x=l$Phylo.Relatedness,y=trajF(observed[1,2],observed[2,2],observed[3,2],l$Phylo.Rel

#observed glmer (1|Species)
yobs.glmer<-data.frame(x=l$Phylo.Relatedness,y=trajF(observed.glmer[1,2],observed.glmer[2,2],observed.gl

#observed bayesian with species level effect
confint.bayes<-data.frame("2.5%"=c(-2.44,4.72,-9.37),"97.5%"=c(-1.48,8.62,-14.71))
colnames(confint.bayes)<-colnames(confint.glm)

yobs.bayes<-data.frame(x=l$Phylo.Relatedness,y=trajF(observed.bayes[1,2],observed.bayes[2,2],observed.ba

obs<-melt(list(GLM=yobs.glm,Mixed_Effects=yobs.glmer,Bayes=yobs.bayes),id.var=c("x","y"))

ggplot(data=conf,aes(x=s,y=mean)) + geom_ribbon(aes(ymin=lower,ymax=upper),fill='gray80') + theme_bw()
```

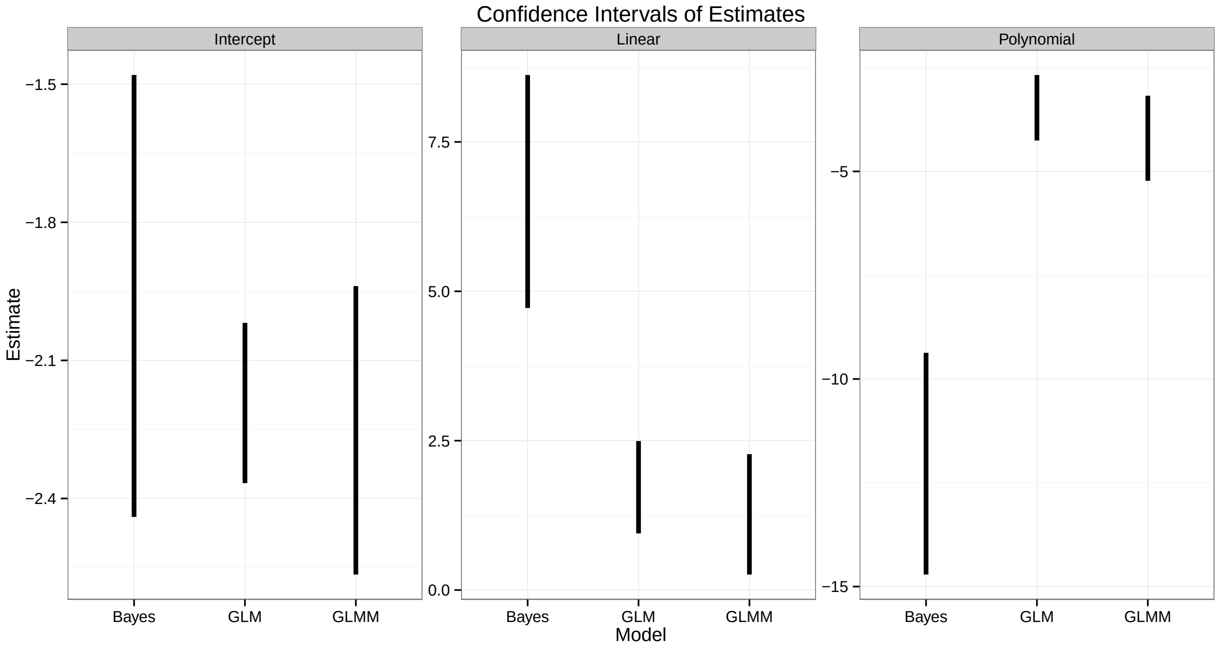


The grey area is the null distrubition of glm estimates. The red line is the Bayesian estimate, teal is the glm mixed effects (1|Species), and the green is glm without accounting for species differences.

CI Intervals

The confidence intervals are not show here, and there is quite a bit of variance in the bayesian thats not quite as extensive in the glmm.

```
co<-list(GLM=confint.glm,GLMM=confint.glmer[-1,],Bayes=confint.bayes)
#standardize row and columne names
co<-lapply(co,function(x){
  rownames(x)<-c("Intercept","Linear","Polynomial")
  colnames(x)<-c("Lower","Upper")
  return(as.matrix(x))
})
co<-melt(co)
co<-dcast(co,...~Var2)
ggplot(co,aes(x=L1,ymin=Lower,ymax=Upper))  + facet_wrap(~Var1,scales="free") + geom_linerange(size=1.4
```



```r
save.image("Independence.Rdata")
#load image if desired
load("Independence.Rdata")
```