NewModelsForSeanToView

Cole

August 12, 2019

Preliminary Things

I thought this might be the easiest way to show you all the things without having to copy and paste everything into a word document, so here's an RMD instead lol.

Necessary packages etc:

```
## Warning: package 'tidyverse' was built under R version 3.5.3
```

```
## -- Attaching packages --------------------------------------------------------------------------------- tidyverse 1.2.1 -
-
```

```
## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
## Warning: package 'tibble' was built under R version 3.5.3
```

```
## Warning: package 'tidyr' was built under R version 3.5.3
```

```
## Warning: package 'readr' was built under R version 3.5.3
```

```
## Warning: package 'purrr' was built under R version 3.5.3
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
## Warning: package 'stringr' was built under R version 3.5.3
```

```
## Warning: package 'forcats' was built under R version 3.5.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------ tidyverse_conflicts() -
-
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Warning: package 'glmmTMB' was built under R version 3.5.2
```

```
## Warning: package 'ggeffects' was built under R version 3.5.3
```

```
## Warning: package 'DHARMa' was built under R version 3.5.3
```

```
## Warning: package 'MuMIn' was built under R version 3.5.3
```

Models

Now, first off the models:

```
lepmod.yrsrsp <- glmmTMB(all.leps ~ spp + site.region + year - 1 + (1|collection),
                      data = mainlice, family=nbinom2)
calmod.yrsrsp <- glmmTMB(all.cal ~ spp + site.region + year - 1 + (1|collection),
                      data = mainlice, family=nbinom2)
```

Model Diagnostics

A look at the summaries, and we'll use the DHARMa package to look at the diagnostic plots since the usual techniques don't work with TMB objects. This package implements simulated quantile residuals for glmms. One important note, is that the developers of this package note that Simulate() is unconditional, i.e. all random effects will be re-simulated, and thus all predictions and simulations are conditional on REs, which can sometimes result in a positive correlation between res and predicted when the random effects are strong in the model. So I'm not too worried about the fact that there's a slight correlation here for us as I don't think it's a problem fit-wise.

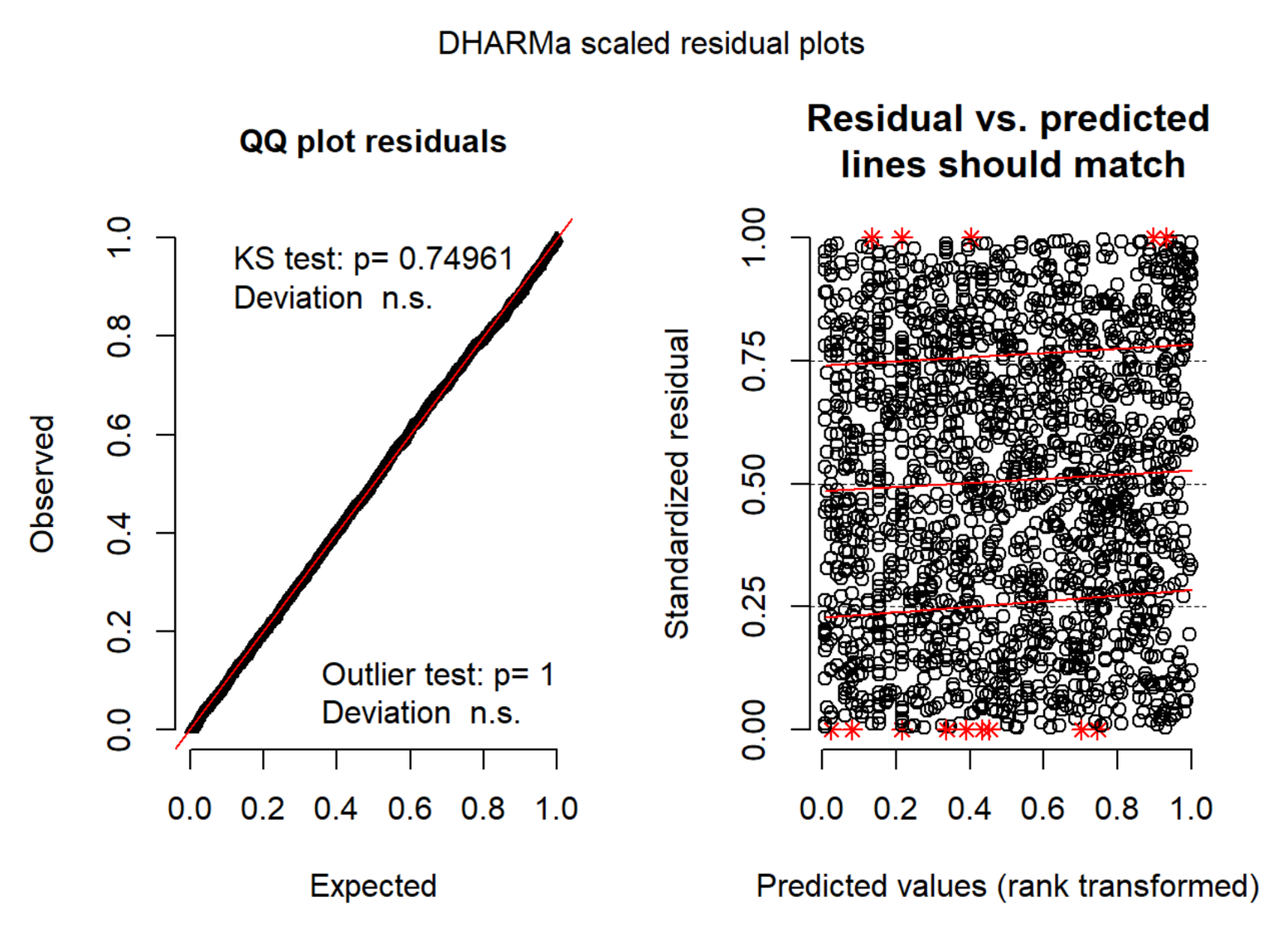
```
summary(lepmod.yrsrsp)
```

```
##  Family: nbinom2  ( log )
## Formula:
## all.leps ~ spp + site.region + year - 1 + (1 | collection)
## Data: mainlice
##
##      AIC      BIC   logLik deviance df.resid
##    854.6    904.2   -418.3    836.6     1826
##
## Random effects:
##
## Conditional model:
##  Groups     Name        Variance Std.Dev.
##  collection (Intercept) 0.6749   0.8215
## Number of obs: 1835, groups:  collection, 52
##
## Overdispersion parameter for nbinom2 family (): 0.41
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## sppCU         -1.9832     0.3980  -4.982 6.28e-07 ***
## sppPI         -1.1917     0.3658  -3.258  0.00112 **
## sppSO         -3.4837     0.4345  -8.017 1.08e-15 ***
## site.regionJ  -0.1716     0.3367  -0.510  0.61032
## year2016      -0.6586     0.3881  -1.697  0.08969 .
## year2017      -2.0926     0.8863  -2.361  0.01822 *
## year2018      -2.1216     0.5139  -4.129 3.65e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_lep = simulateResiduals(lepmod.yrsrsp)
```

```
## It seems you are diagnosing a glmmTBM model. There are still a few minor limitations associatd with this package. The mos
t important is that glmmTMB doesn't implement an option to create unconditional predictions from the model, which means that
predicted values (in res ~ pred) plots include the random effects. With strong random effects, this can sometimes create dia
gonal patterns from bottom left to top right in the res ~ pred plot. All other tests and plots should work as desired. Pleas
e see https://github.com/florianhartig/DHARMa/issues/16 for further details.
```

```
plot(res_lep)
```



DHARMa scaled residual plots
QQ plot residuals
KS test: p= 0.74961
Deviation n.s.
Outlier test: p= 1
Deviation n.s.
Observed
Expected
0.0
0.2
0.4
0.6
0.8
1.0
Residual vs. predicted
lines should match
Standardized residual
0.00
0.25
0.50
0.75
1.00
Predicted values (rank transformed)

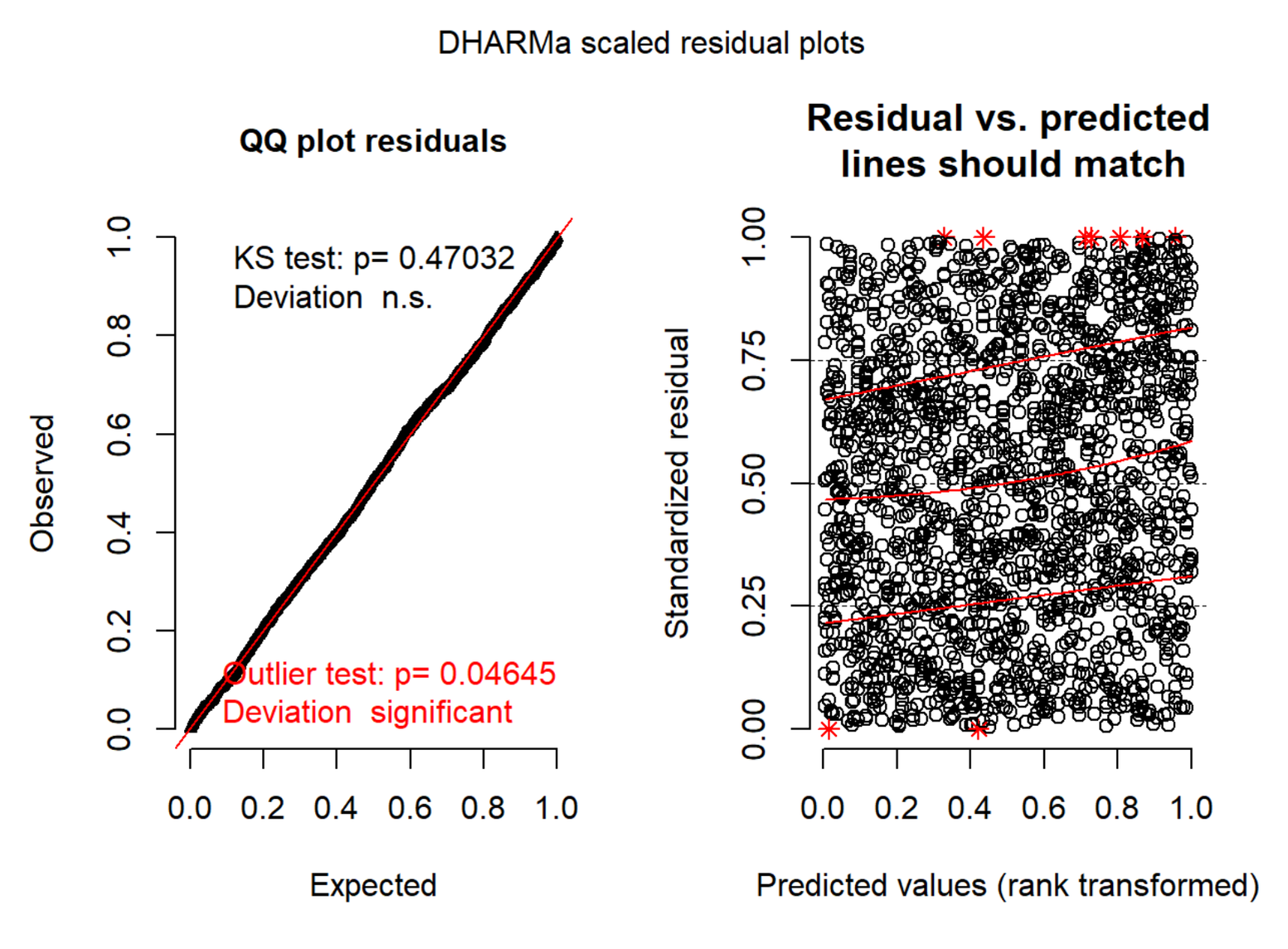

```
summary(calmod.yrsrsp)
```

```
##  Family: nbinom2  ( log )
## Formula:
## all.cal ~ spp + site.region + year - 1 + (1 | collection)
## Data: mainlice
##
##      AIC      BIC   logLik deviance df.resid
##   2990.3   3039.9  -1486.2   2972.3     1826
##
## Random effects:
##
## Conditional model:
##  Groups     Name        Variance Std.Dev.
##  collection (Intercept) 0.1011   0.3179
## Number of obs: 1835, groups:  collection, 52
##
## Overdispersion parameter for nbinom2 family (): 1.38
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## sppCU         -1.1611     0.1729  -6.714 1.89e-11 ***
## sppPI         -0.5805     0.1583  -3.667 0.000245 ***
## sppSO         -0.6864     0.1526  -4.497 6.88e-06 ***
## site.regionJ   0.4015     0.1265   3.175 0.001499 **
## year2016      -0.5786     0.1578  -3.667 0.000245 ***
## year2017      -0.7778     0.2848  -2.731 0.006307 **
## year2018      -0.5159     0.1770  -2.915 0.003557 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
res_cal = simulateResiduals(calmod.yrsrsp)
```

```
## It seems you are diagnosing a glmmTBM model. There are still a few minor limitations associatd with this package. The mos
t important is that glmmTMB doesn't implement an option to create unconditional predictions from the model, which means that
predicted values (in res ~ pred) plots include the random effects. With strong random effects, this can sometimes create dia
gonal patterns from bottom left to top right in the res ~ pred plot. All other tests and plots should work as desired. Pleas
e see https://github.com/florianhartig/DHARMa/issues/16 for further details.
```

```
plot(res_cal, rank = T)
```



DHARMa scaled residual plots
QQ plot residuals
KS test: p= 0.47032
Deviation n.s.
Outlier test: p= 0.04645
Deviation significant
Observed
Expected
0.0
0.2
0.4
0.6
0.8
1.0
Residual vs. predicted
lines should match
Standardized residual
0.00
0.25
0.50
0.75
1.00
Predicted values (rank transformed)

So we can see that the Caligus model as a significant value, but visually it looks good.

AIC table

I'll use dredging to get the full model set and see how it looks:

```
lepmod.yrsrsp_dredge = MuMIn::dredge(lepmod.yrsrsp)
```

```
## Fixed term is "disp((Int))"
```

```
## Warning in glmmTMB(formula = all.leps ~ 0, data = mainlice, family =
## nbinom2, : unused argument (`NA` = ~(1 | collection)) (model 0 skipped)
```

```
calmod.yrsrsp_dredge = MuMIn::dredge(calmod.yrsrsp)
```

```
## Fixed term is "disp((Int))"
```

```
## Warning in glmmTMB(formula = all.cal ~ 0, data = mainlice, family =
## nbinom2, : unused argument (`NA` = ~(1 | collection)) (model 0 skipped)
```

```
lepmod.yrsrsp_dredge
```

```
## Global model call: glmmTMB(formula = all.leps ~ spp + site.region + year - 1 + (1 |
##     collection), data = mainlice, family = nbinom2, ziformula = ~0,
##     dispformula = ~1)
## ---
## Model selection table
##   dsp((Int)) cnd(sit.rgn) cnd(spp) cnd(yer) df   logLik  AICc delta weight
## 7          +                     +        +  8 -418.419 852.9  0.00  0.706
## 8          +             +       +        +  9 -418.292 854.7  1.76  0.292
## 3          +                     +           5 -427.833 865.7 12.78  0.001
## 4          +             +       +           6 -427.751 867.5 14.63  0.000
## 5          +                              +  6 -452.366 916.8 63.86  0.000
## 6          +             +                +  7 -452.342 918.7 65.83  0.000
## 2          +             +                   4 -461.316 930.7 77.74  0.000
## Models ranked by AICc(x)
## Random terms (all models):
## 'cond(1 | collection)'
```

```
calmod.yrsrsp_dredge
```

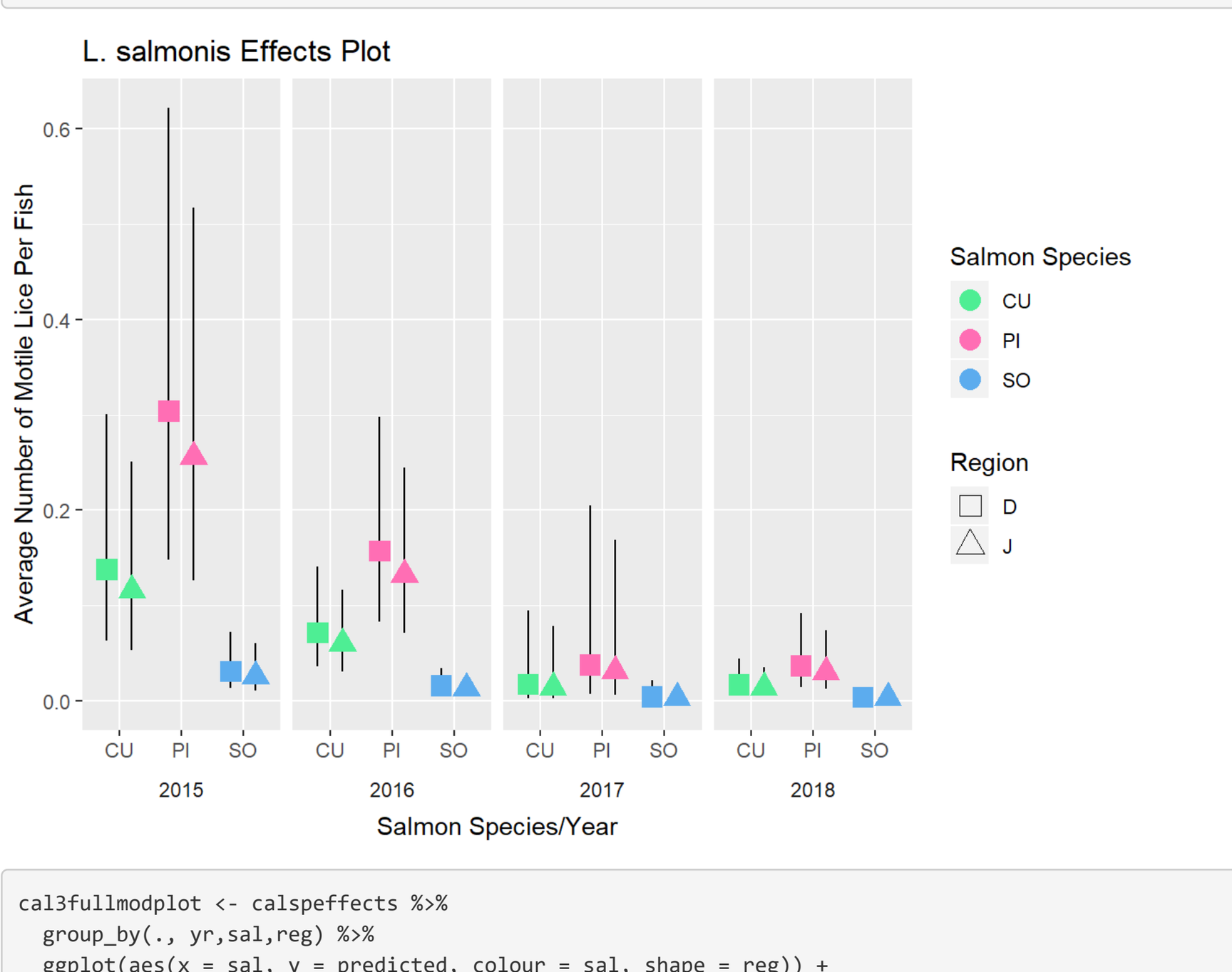
```
## Global model call: glmmTMB(formula = all.cal ~ spp + site.region + year - 1 + (1 |
##     collection), data = mainlice, family = nbinom2, ziformula = ~0,
##     dispformula = ~1)
## ---
## Model selection table
##   dsp((Int)) cnd(sit.rgn) cnd(spp) cnd(yer) df    logLik   AICc delta
## 8          +             +       +        +  9 -1486.158 2990.4  0.00
## 7          +                     +        +  8 -1490.784 2997.6  7.23
## 4          +             +       +           6 -1493.201 2998.4  8.03
## 3          +                     +           5 -1497.330 3004.7 14.28
## 6          +             +                +  7 -1498.198 3010.5 20.04
## 5          +                              +  6 -1502.828 3017.7 27.29
## 2          +             +                   4 -1505.653 3019.3 28.91
##   weight
## 8  0.956
## 7  0.026
## 4  0.017
## 3  0.001
## 6  0.000
## 5  0.000
## 2  0.000
## Models ranked by AICc(x)
## Random terms (all models):
## 'cond(1 | collection)'
```

This was our problem, that the best model for the leps was not the full model and so we couldn't get at region-level differences here.

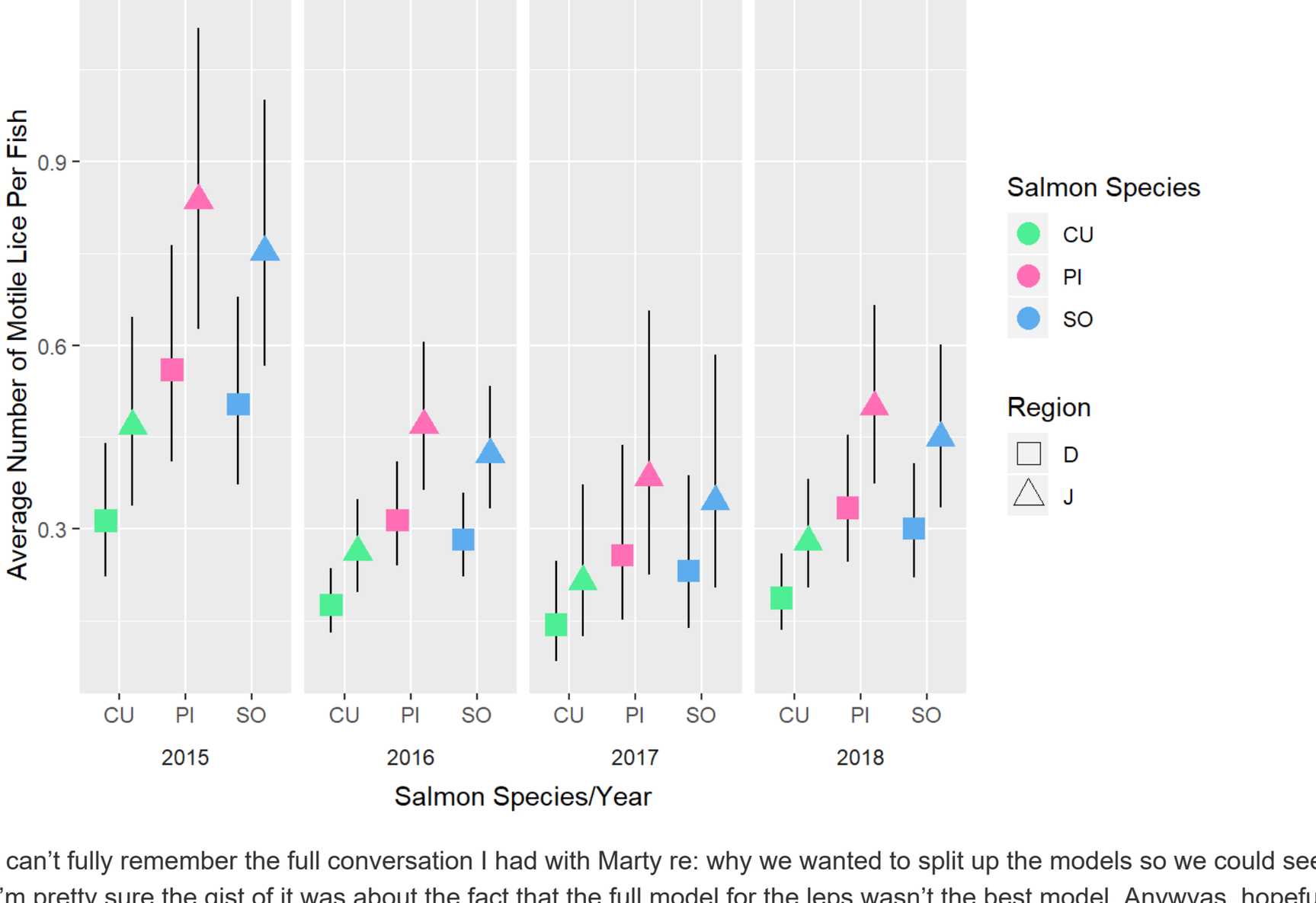
Effects Plots

Using the ggeffects package we can get the predicted estimates/95% CIs and take a look at what those look like.

```
calspeffects <- ggpredict(calmod.yrsrsp, terms = c('spp', 'year', 'site.region'))
lepspeffects <- ggpredict(lepmod.yrsrsp, terms = c('spp', 'year', 'site.region'))

calspeffects = calspeffects %>%
  rename(sal = x, reg = facet, yr = group)

calspeffects$sal = factor(calspeffects$sal, levels = c(1, 2, 3), labels = c('CU', 'PI', 'SO'))

lepspeffects = lepspeffects %>%
  rename(sal = x, reg = facet, yr = group)

lepspeffects$sal = factor(lepspeffects$sal, levels = c(1, 2, 3), labels = c('CU', 'PI', 'SO'))

## Make the plots

leg_title <- 'Salmon Species'
leps3fullmodplot <- lepspeffects %>%
  group_by(., yr,sal,reg) %>%
  ggplot(aes(x = sal, y = predicted, colour = sal, shape = reg)) +
  scale_shape_manual(values = c(15,17)) +
  geom_errorbar(aes(ymin=conf.low, ymax = conf.high,width = 0), position = position_dodge(width = 0.8),colour = 'Black')+
  geom_point(size = 4,position = position_dodge(width = 0.8)) +
  facet_wrap(~yr,nrow=1,strip.position = "bottom")+
  theme(strip.background = element_blank(), strip.placement = "outside") +
  scale_color_manual(leg_title,values=c('seagreen2', 'hotpink1', 'steelblue2'))+
  labs(title = "L. salmonis Effects Plot", x = 'Salmon Species/Year', y = 'Average Number of Motile Lice Per Fish') +
  guides(shape = guide_legend(title = 'Region', override.aes = list(shape = c(0,2)), type = 'b'))
leps3fullmodplot
```




```
cal3fullmodplot <- calspeffects %>%
  group_by(., yr,sal,reg) %>%
  ggplot(aes(x = sal, y = predicted, colour = sal, shape = reg)) +
  scale_shape_manual(values = c(15,17)) +
  geom_errorbar(aes(ymin=conf.low, ymax = conf.high,width = 0), position = position_dodge(width = 0.8),colour = 'Black')+
  geom_point(size = 4,position = position_dodge(width = 0.8)) +
  facet_wrap(~yr,nrow=1,strip.position = "bottom")+
  theme(strip.background = element_blank(), strip.placement = "outside") +
  scale_color_manual(leg_title,values=c('seagreen2', 'hotpink1', 'steelblue2'))+
  labs(title = "C. clemensi Effects Plot", x = 'Salmon Species/Year', y = 'Average Number of Motile Lice Per Fish') +
  guides(shape = guide_legend(title = 'Region', override.aes = list(shape = c(0,2)), type = 'b'))
cal3fullmodplot
```



I can't fully remember the full conversation I had with Marty re: why we wanted to split up the models so we could see the region-level effects, but I'm pretty sure the gist of it was about the fact that the full model for the leps wasn't the best model. Anywyas, hopefully this is helpful! Let me know what you think about all of this and how you'd like me to move forwards! I'll go and start addressing some of your comments in the MS that don't necessarily rely on the exact results and I guess we can go from there!