Model-Averaging-For-Sean

Cole

September 19, 2019

```
## 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_conflict
s() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## 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
```

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

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

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

```
##
## Attaching package: 'AICcmodavg'
```

```
## The following objects are masked from 'package:MuMIn':
##
##     AICc, DIC, importance
```

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

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

Models

Models (Crossed Effects)

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

AIC Tables

```
lepmod.crossed_dredge = MuMIn::dredge(lepmod.crossed, subset = (`cond(site.region)` && `cond(year)`))
```

```
## Fixed terms are "cond((Int))" and "disp((Int))"
```

```
lepmod.crossed_dredge
```

```
## Global model call: glmmTMB(formula = all.leps ~ spp * site.region + spp * year +
##     site.region * year + (1 | collection), data = mainlice, family = nbinom2,
##     ziformula = ~0, dispformula = ~1)
## ---
## Model selection table
##    cnd((Int)) dsp((Int)) cnd(sit.rgn) cnd(spp) cnd(yer) cnd(sit.rgn:spp)
## 24     -1.185          +            +        +        +
## 56     -1.469          +            +        +        +
## 32     -1.125          +            +        +        +                +
## 64     -1.427          +            +        +        +                +
## 8      -1.983          +            +        +        +
## 40     -2.244          +            +        +        +
## 16     -1.957          +            +        +        +                +
## 48     -2.227          +            +        +        +                +
## 22     -1.051          +            +                 +
## 6      -1.939          +            +                 +
##    cnd(sit.rgn:yer) cnd(spp:yer) df   logLik  AICc delta weight
## 24                +              12 -411.452 847.1  0.00  0.529
## 56                +            + 18 -406.204 848.8  1.71  0.225
## 32                +              14 -410.656 849.5  2.47  0.154
## 64                +            + 20 -405.297 851.1  3.98  0.072
## 8                                 9 -418.292 854.7  7.61  0.012
## 40                             + 15 -413.222 856.7  9.63  0.004
## 16                               11 -417.618 857.4 10.30  0.003
## 48                             + 17 -412.495 859.3 12.25  0.001
## 22                +              10 -444.677 909.5 62.40  0.000
## 6                                 7 -452.342 918.7 71.67  0.000
## Models ranked by AICc(x)
## Random terms (all models):
## 'cond(1 | collection)'
```

```
calmod.crossed_dredge = MuMIn::dredge(calmod.crossed, subset = (`cond(site.region)` && `cond(year)`))
```

```
## Fixed terms are "cond((Int))" and "disp((Int))"
```

```
calmod.crossed_dredge
```

```
## Global model call: glmmTMB(formula = all.cal ~ spp * site.region + spp * year +
##     site.region * year + (1 | collection), data = mainlice, family = nbinom2,
##     ziformula = ~0, dispformula = ~1)
## ---
## Model selection table
##    cnd((Int)) dsp((Int)) cnd(sit.rgn) cnd(spp) cnd(yer) cnd(sit.rgn:spp)
## 32    -0.7104          +            +        +        +                +
## 24    -0.7910          +            +        +        +
## 64    -0.8636          +            +        +        +                +
## 56    -0.9259          +            +        +        +
## 8     -1.1610          +            +        +        +
## 16    -1.0900          +            +        +        +                +
## 40    -1.2760          +            +        +        +
## 48    -1.2170          +            +        +        +                +
## 22    -0.3716          +            +                 +
## 6     -0.7379          +            +                 +
##    cnd(sit.rgn:yer) cnd(spp:yer) df    logLik   AICc delta weight
## 32                +              14 -1478.419 2985.1  0.00  0.410
## 24                +              12 -1480.773 2985.7  0.65  0.297
## 64                +            + 20 -1473.480 2987.4  2.35  0.126
## 56                +            + 18 -1475.829 2988.0  2.97  0.093
## 8                                 9 -1486.158 2990.4  5.35  0.028
## 16                               11 -1484.199 2990.5  5.47  0.027
## 40                             + 15 -1481.158 2992.6  7.51  0.010
## 48                             + 17 -1479.200 2992.7  7.67  0.009
## 22                +              10 -1493.316 3006.8 21.68  0.000
## 6                                 7 -1498.198 3010.5 25.39  0.000
## Models ranked by AICc(x)
## Random terms (all models):
## 'cond(1 | collection)'
```

Model Averaging

So the goal here is to get the model-averaged predictions and then the confidence intervals.

For the predictions, we have the set of R candidate models $M_1, \dots, M_n$ models and we want the model-averaged estimator $\hat{\theta}$ of θ as the weighted average from each model

$$\hat{\theta} = \sum_{i=1}^{R} w_i \hat{\theta}_i$$

. This can be used to average the predictions that we get using the ggpredict package.

First we need all our candidate models - for simplicity sake they've been organized by the ranking from the model comparison.

```
#omited the last two models in each set since their weights were 0.
lep1 = glmmTMB(all.leps ~ site.region + year + spp +
            site.region * year +
            (1 | collection), data = mainlice, family = nbinom2)
lep2 = glmmTMB(all.leps ~ site.region + year + spp +
            spp * year + site.region * year +
            (1 | collection), data = mainlice, family = nbinom2)
lep3 = glmmTMB(all.leps ~ site.region + year + spp +
            spp * site.region + site.region * year +
            (1 | collection), data = mainlice, family = nbinom2)
lep4 = glmmTMB(all.leps ~ site.region + year + spp +
            spp * site.region + spp * year + site.region * year +
            (1 | collection), data = mainlice, family = nbinom2)
lep5 = glmmTMB(all.leps ~ site.region + year + spp +
            (1 | collection), data = mainlice, family = nbinom2)
lep6 = glmmTMB(all.leps ~ site.region + year + spp +
            spp * year +
            (1 | collection), data = mainlice, family = nbinom2)
lep7 = glmmTMB(all.leps ~ site.region + year + spp +
            spp * site.region +
            (1 | collection), data = mainlice, family = nbinom2)
lep8 = glmmTMB(all.leps ~ site.region + year + spp +
            spp * site.region + spp * year +
            (1 | collection), data = mainlice, family = nbinom2)



cal1 = glmmTMB(all.cal ~ site.region + year + spp +
            site.region * year + site.region * spp +
            (1 | collection), data = mainlice, family = nbinom2)
cal2 = glmmTMB(all.cal ~ site.region + year + spp +
            site.region * year +
            (1 | collection), data = mainlice, family = nbinom2)
cal3 = glmmTMB(all.cal ~ site.region + year + spp +
            spp * site.region + site.region * year + spp * year +
            (1 | collection), data = mainlice, family = nbinom2)
cal4 = glmmTMB(all.cal ~ site.region + year + spp +
            spp * year + site.region * year +
            (1 | collection), data = mainlice, family = nbinom2)
cal5 = glmmTMB(all.cal ~ site.region + year + spp +
            (1 | collection), data = mainlice, family = nbinom2)
cal6 = glmmTMB(all.cal ~ site.region + year + spp +
            site.region * year +
            (1 | collection), data = mainlice, family = nbinom2)
cal7 = glmmTMB(all.cal ~ site.region + year + spp +
            spp * year +
            (1 | collection), data = mainlice, family = nbinom2)
cal8 = glmmTMB(all.cal ~ site.region + year + spp +
            spp * site.region + spp * year +
            (1 | collection), data = mainlice, family = nbinom2)
summary(cal1)
```

```
##  Family: nbinom2  ( log )
## Formula:
## all.cal ~ site.region + year + spp + site.region * year + site.region *
##     spp + (1 | collection)
## Data: mainlice
##
##      AIC      BIC   logLik deviance df.resid
##   2984.8   3062.0  -1478.4   2956.8     1821
##
## Random effects:
##
## Conditional model:
##  Groups     Name        Variance Std.Dev.
##  collection (Intercept) 0.06448  0.2539
## Number of obs: 1835, groups:  collection, 52
##
## Overdispersion parameter for nbinom2 family (): 1.41
##
## Conditional model:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.71037    0.21773  -3.263 0.001104 **
## site.regionJ          -0.37197    0.29207  -1.274 0.202823
## year2016              -1.13092    0.21375  -5.291 1.22e-07 ***
## year2017              -1.40737    0.35949  -3.915 9.05e-05 ***
## year2018              -0.88082    0.24456  -3.602 0.000316 ***
## sppPI                  0.33813    0.19955   1.694 0.090184 .
## sppSO                  0.49964    0.17934   2.786 0.005335 **
## site.regionJ:year2016  0.97337    0.28481   3.418 0.000632 ***
## site.regionJ:year2017  1.16990    0.51493   2.272 0.023088 *
## site.regionJ:year2018  0.63727    0.32298   1.973 0.048486 *
## site.regionJ:sppPI     0.40133    0.25662   1.564 0.117840
## site.regionJ:sppSO    -0.02155    0.23835  -0.090 0.927962
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now let's use the ggpredict package to get predictions:

```
lep1pred <- ggpredict(lep1, terms = c('spp', 'year', 'site.region'))
lep2pred <- ggpredict(lep2, terms = c('spp', 'year', 'site.region'))
lep3pred <- ggpredict(lep3, terms = c('spp', 'year', 'site.region'))
lep4pred <- ggpredict(lep4, terms = c('spp', 'year', 'site.region'))
lep5pred <- ggpredict(lep5, terms = c('spp', 'year', 'site.region'))
lep6pred <- ggpredict(lep6, terms = c('spp', 'year', 'site.region'))
lep7pred <- ggpredict(lep7, terms = c('spp', 'year', 'site.region'))
lep8pred <- ggpredict(lep8, terms = c('spp', 'year', 'site.region'))

cal1pred <- ggpredict(cal1, terms = c('spp', 'year', 'site.region'))
cal2pred <- ggpredict(cal2, terms = c('spp', 'year', 'site.region'))
cal3pred <- ggpredict(cal3, terms = c('spp', 'year', 'site.region'))
cal4pred <- ggpredict(cal4, terms = c('spp', 'year', 'site.region'))
cal5pred <- ggpredict(cal5, terms = c('spp', 'year', 'site.region'))
cal6pred <- ggpredict(cal6, terms = c('spp', 'year', 'site.region'))
cal7pred <- ggpredict(cal7, terms = c('spp', 'year', 'site.region'))
cal8pred <- ggpredict(cal8, terms = c('spp', 'year', 'site.region'))
```

So now that we have the predictions, we can use the weights to get model averaged results for each.

```
###start by getting them all in one dataframe with the weights

#pull the predicted values from each one
lepallpred = data.frame(cbind(lep1pred$predicted, lep2pred$predicted, lep3pred$predicted, lep4pred$predicted,
                   lep5pred$predicted, lep6pred$predicted, lep7pred$predicted, lep8pred$predicted)) %>%
                  rename(lep1 = X1, lep2 = X2, lep3 = X3, lep4 = X4, lep5 = X5, lep6 = X6, lep7 = X7, lep8 = X8)
calallpred = data.frame(cbind(cal1pred$predicted, cal2pred$predicted, cal3pred$predicted, cal4pred$predicted,
                   cal5pred$predicted, cal6pred$predicted, cal7pred$predicted, cal8pred$predicted)) %>%
                  rename(cal1 = X1, cal2 = X2, cal3 = X3, cal4 = X4, cal5 = X5, cal6 = X6, cal7 = X7, cal8 = X8)

#add the weights from the model selection object
lepallpred = lepallpred %>%
  mutate(w1 = rep(lepmod.crossed_dredge$weight[1], nrow(lepallpred)),
         w2 = rep(lepmod.crossed_dredge$weight[2], nrow(lepallpred)),
         w3 = rep(lepmod.crossed_dredge$weight[3], nrow(lepallpred)),
         w4 = rep(lepmod.crossed_dredge$weight[4], nrow(lepallpred)),
         w5 = rep(lepmod.crossed_dredge$weight[5], nrow(lepallpred)),
         w6 = rep(lepmod.crossed_dredge$weight[6], nrow(lepallpred)),
         w7 = rep(lepmod.crossed_dredge$weight[7], nrow(lepallpred)),
         w8 = rep(lepmod.crossed_dredge$weight[8], nrow(lepallpred)))

calallpred = calallpred %>%
  mutate(w1 = rep(calmod.crossed_dredge$weight[1], nrow(calallpred)),
         w2 = rep(calmod.crossed_dredge$weight[2], nrow(calallpred)),
         w3 = rep(calmod.crossed_dredge$weight[3], nrow(calallpred)),
         w4 = rep(calmod.crossed_dredge$weight[4], nrow(calallpred)),
         w5 = rep(calmod.crossed_dredge$weight[5], nrow(calallpred)),
         w6 = rep(calmod.crossed_dredge$weight[6], nrow(calallpred)),
         w7 = rep(calmod.crossed_dredge$weight[7], nrow(calallpred)),
         w8 = rep(calmod.crossed_dredge$weight[8], nrow(calallpred)))

#now make averaged predictions!
lepallpred = lepallpred %>%
  mutate(lep1w = lep1*w1, lep2w = lep2*w2, lep3w = lep3*w3, lep4w = lep4*w4, lep5w = lep5*w5, lep6w = lep6*w6, lep7w = lep7*
w7, lep8w = lep8*w8) %>%
  mutate(avg = lep1w + lep2w + lep3w + lep4w + lep5w + lep6w + lep7w + lep8w)

calallpred = calallpred %>%
  mutate(cal1w = cal1*w1, cal2w = cal2*w2, cal3w = cal3*w3, cal4w = cal4*w4, cal5w = cal5*w5, cal6w = cal6*w6, cal7w = cal7*
w7, cal8w = cal8*w8) %>%
  mutate(avg = cal1w + cal2w + cal3w + cal4w + cal5w + cal6w + cal7w + cal8w)

#keep just the averaged predictions and the relevant grouping info
lepavgpred = lepallpred %>%
  select(avg) %>%
  mutate(sal = lep1pred$x, reg = lep1pred$facet, yr = lep1pred$group)
lepavgpred$sal = factor(lepavgpred$sal, levels = c(1, 2, 3), labels = c('CU', 'PI', 'SO'))

calavgpred = calallpred %>%
  select(avg) %>%
  mutate(sal = cal1pred$x, reg = cal1pred$facet, yr = cal1pred$group)
calavgpred$sal = factor(calavgpred$sal, levels = c(1, 2, 3), labels = c('CU', 'PI', 'SO'))
```

Confidence Intervals

Now for the frustratingly simple solution to a problem that took way too many hours to solve.

```
#### THIS WORKS
devtools::install_github("glmmTMB/glmmTMB/glmmTMB")
```

```
## Skipping install of 'glmmTMB' from a github remote, the SHA1 (f84f3e95) has not changed since last install.
##   Use `force = TRUE` to force installation
```

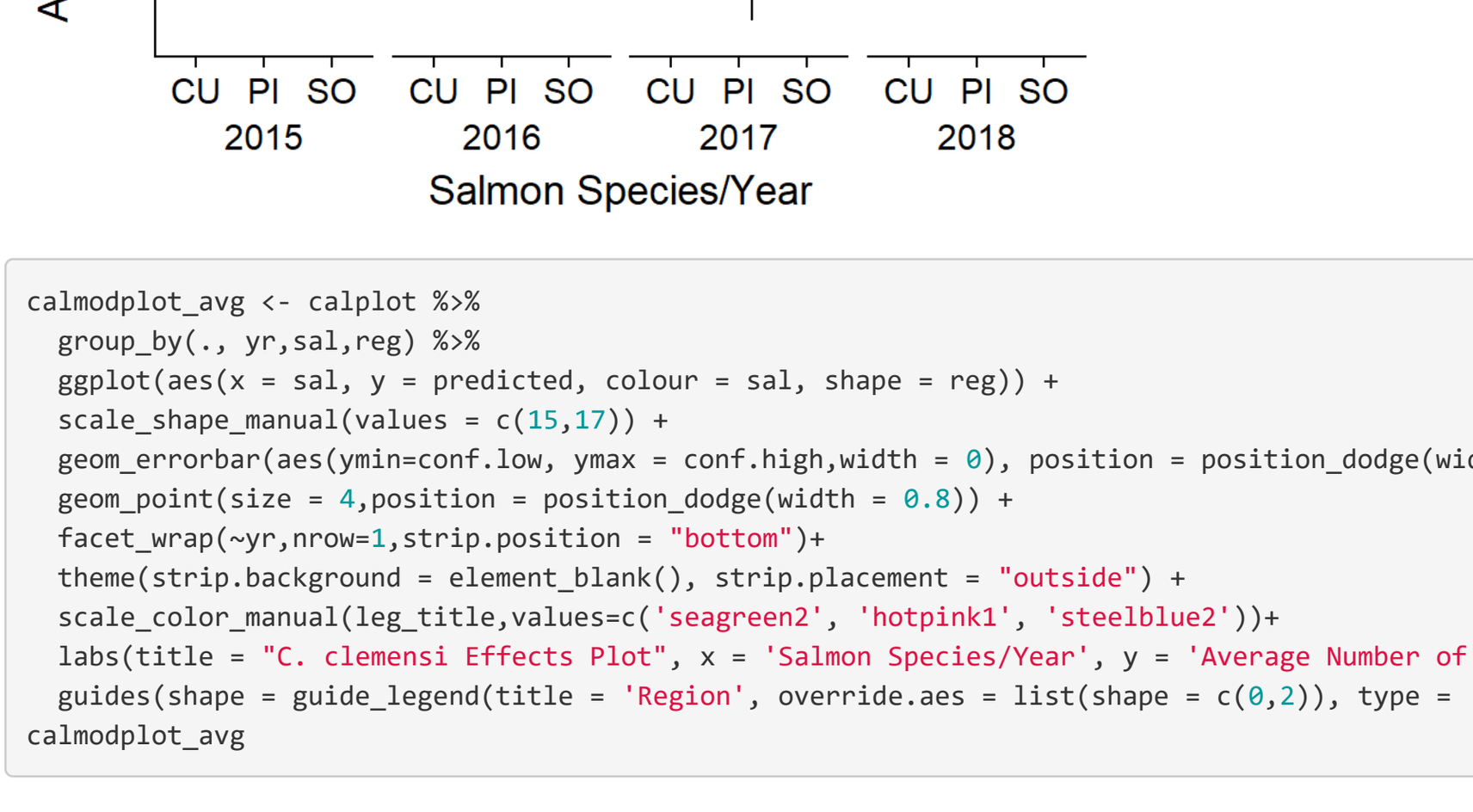
```
library(glmmTMB)

DI = as.vector(rep('D', 12)); JS = as.vector(rep('J', 12))
f = as.vector(rep('2015', 3)); si = as.vector(rep('2016', 3)); sv = as.vector(rep('2017', 3)); e = as.vector(rep('2018', 3))
year = as.vector(cbind(f, si, sv, e)); year = as.vector(replicate(2, year))
CU = 'CU'; PI = 'PI'; SO = 'SO'
spp = as.vector(rbind(CU, PI, SO)); spp = as.vector(replicate(8, spp))

newdata <- matrix(nrow = 24, ncol = 4)
newdata[c(1:12), 1] = DI; newdata[c(13:24), 1] = JS
newdata[c(1:24), 2] = year
newdata[c(1:24), 3] = spp

newdata = data.frame(newdata)
newdata = newdata %>%
  rename(site.region = `X1`, year = `X2`, spp = `X3`, collection = `X4`)
newdata$collection = NA

#for leps
lepconfset1 = get.models(lepmod.crossed_dredge, subset = TRUE)
lepmodavg1 = model.avg(lepconfset1, subset = TRUE)

newdata$collection = NA
leppredict1 = data.frame(
  model = sapply(lepconfset1, predict, newdata = newdata),
  averagedfull = predict(lepmodavg1, type = 'response', newdata = newdata, se.fit = TRUE))
leppredict1$low = leppredict1$averagedfull.fit - (1.96*leppredict1$averagedfull.se.fit)
leppredict1$up = leppredict1$averagedfull.fit + (1.96*leppredict1$averagedfull.se.fit)
leppredict1$mine = lepallpred$avg
leppredict1$mine.low = leppredict1$mine - (1.96*leppredict1$averagedfull.se.fit)
leppredict1$mine.up = leppredict1$mine + (1.96*leppredict1$averagedfull.se.fit)

#for cal
calconfset1 = get.models(calmod.crossed_dredge, subset = TRUE)
calmodavg1 = model.avg(calconfset1, subset = TRUE)
newdata <- matrix(nrow = 24, ncol = 4)
newdata[c(1:12), 1] = DI; newdata[c(13:24), 1] = JS
newdata[c(1:24), 2] = year
newdata[c(1:24), 3] = spp
newdata = data.frame(newdata)
newdata = newdata %>%
  rename(site.region = `X1`, year = `X2`, spp = `X3`, collection = `X4`)
newdata$collection = NA
calpredict1 = data.frame(
  model = sapply(calconfset1, predict, newdata = newdata),
  averagedfull = predict(calmodavg1, type = 'response', newdata = newdata, se.fit = TRUE))
calpredict1$low = calpredict1$averagedfull.fit - (1.96*calpredict1$averagedfull.se.fit)
calpredict1$up = calpredict1$averagedfull.fit + (1.96*calpredict1$averagedfull.se.fit)
calpredict1$mine = calallpred$avg
calpredict1$mine.low = calpredict1$mine - (1.96*calpredict1$averagedfull.se.fit)
calpredict1$mine.up = calpredict1$mine + (1.96*calpredict1$averagedfull.se.fit)
```

Effects Plot

```
reg = data.frame(newdata$site.region)
yr = data.frame(newdata$year)
sal = data.frame(newdata$spp)
lepplot = cbind(data.frame(cbind(leppredict1$averagedfull.fit, leppredict1$averagedfull.se.fit)),
                          reg, yr, sal) %>%
  rename(predicted = X1, se = X2, reg = newdata.site.region, yr =  newdata.year, sal = newdata.spp)
lepplot$conf.low = lepplot$predicted - (1.96*lepplot$se)
lepplot$conf.high = lepplot$predicted + (1.96*lepplot$se)

calplot = cbind(data.frame(cbind(calpredict1$averagedfull.fit, calpredict1$averagedfull.se.fit)),
                          reg, yr, sal) %>%
  rename(predicted = X1, se = X2, reg = newdata.site.region, yr =  newdata.year, sal = newdata.spp)
calplot$conf.low = calplot$predicted - (1.96*calplot$se)
calplot$conf.high = calplot$predicted + (1.96*calplot$se)

str(lepplot)
```

```
## 'data.frame':    24 obs. of  7 variables:
##  $ predicted: num  0.2834 0.6876 0.0711 0.0685 0.1251 ...
##  $ se       : num  0.1263 0.2732 0.0356 0.0269 0.044 ...
##  $ reg      : Factor w/ 2 levels "D","J": 1 1 1 1 1 1 1 1 1 1 ...
##  $ yr       : Factor w/ 4 levels "2015","2016",..: 1 1 1 2 2 2 3 3 3 4 ...
##  $ sal      : Factor w/ 3 levels "CU","PI","SO": 1 2 3 1 2 3 1 2 3 1 ...
##  $ conf.low : num  0.03593 0.15203 0.00123 0.01576 0.03883 ...
##  $ conf.high: num  0.531 1.223 0.141 0.121 0.211 ...
```

```
## Make the plots

leg_title <- 'Salmon Species'
lepsmodplot_avg <- lepplot %>%
  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'))
lepsmodplot_avg
```




```
calmodplot_avg <- calplot %>%
  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'))
calmodplot_avg
```