Sablefish Update Analysis

Introduction and Background

Back in the early summer of 2019, Chris Edwards sent us the updated ROMS output through 2018. At the time, Melissa and I were fully invested in getting the 2019 sablefish assessment done, including the Ecological Considerations section, and I did only a cursory analysis of the new data at the time. The present "document" serves two puposes. First, it lets me play around with RMarkdown. Second, it is an attempt to do some more detailed but still somewhat preliminary analyses of how the updated ROMS data affect the previous work Melissa and I and others did. Key questions are:

(1) Does the model still work?

(2) If not, why? What changed?

The short answer is:

- The model doesn't work, and
- My guess is that the blob years mess up the relationships

Previous work - a reminder

You will remember that we developed a literature-based life-history model for sablefish. We then used that model to select output from a ROMS model to use as predictors of sablefish recruitment, specifically residuals around the stock recruitment curve. We used stock-recruitment information from the 2016 sablefish assessment. The ROMS drivers were avilable for 1980 to 2010. After a bunch of model fitting and tons of model testing and diagnostics, the best fit model was:

residuals ~ DDpre + DDegg + CSTegg + LSTyolk + DDlarv

where:

DDpre = degree days pre-spawning (and indicator of potential female condition)

DDegg = degree days during the egg stage. Affects the rate of development.

CSTegg = cross shelf transport during the egg stage. Staying close to shore is good.

LSTyolk = Long shore transport during the yolksack larval stage. Transport to the north is good (at 1000 m)

DDlav = warmer temps lead to faster growth.

The r^2 for this model was 0.57, which is not great, but is better than the 0.5 needed to improve the stock assessment.

New data

Chris Edwards sent us update ROMS data for 2011 - 2018. At the time, he only sent updated data for the predictors in the best-fit model. We might ask for two things moving forwards: (1) updated data through 2019 and (2) all of the previously tested predictors to re-do the entire model fitting for 1981 - 2019.

There appears to be some change in median and variance between the old ROMS model and new ROMS model for the 1981-2010 and 2011- $-\infty$ ROMS data. Some of this difference, especially the degree days or temperature, is due to the blob, especially in 2016.

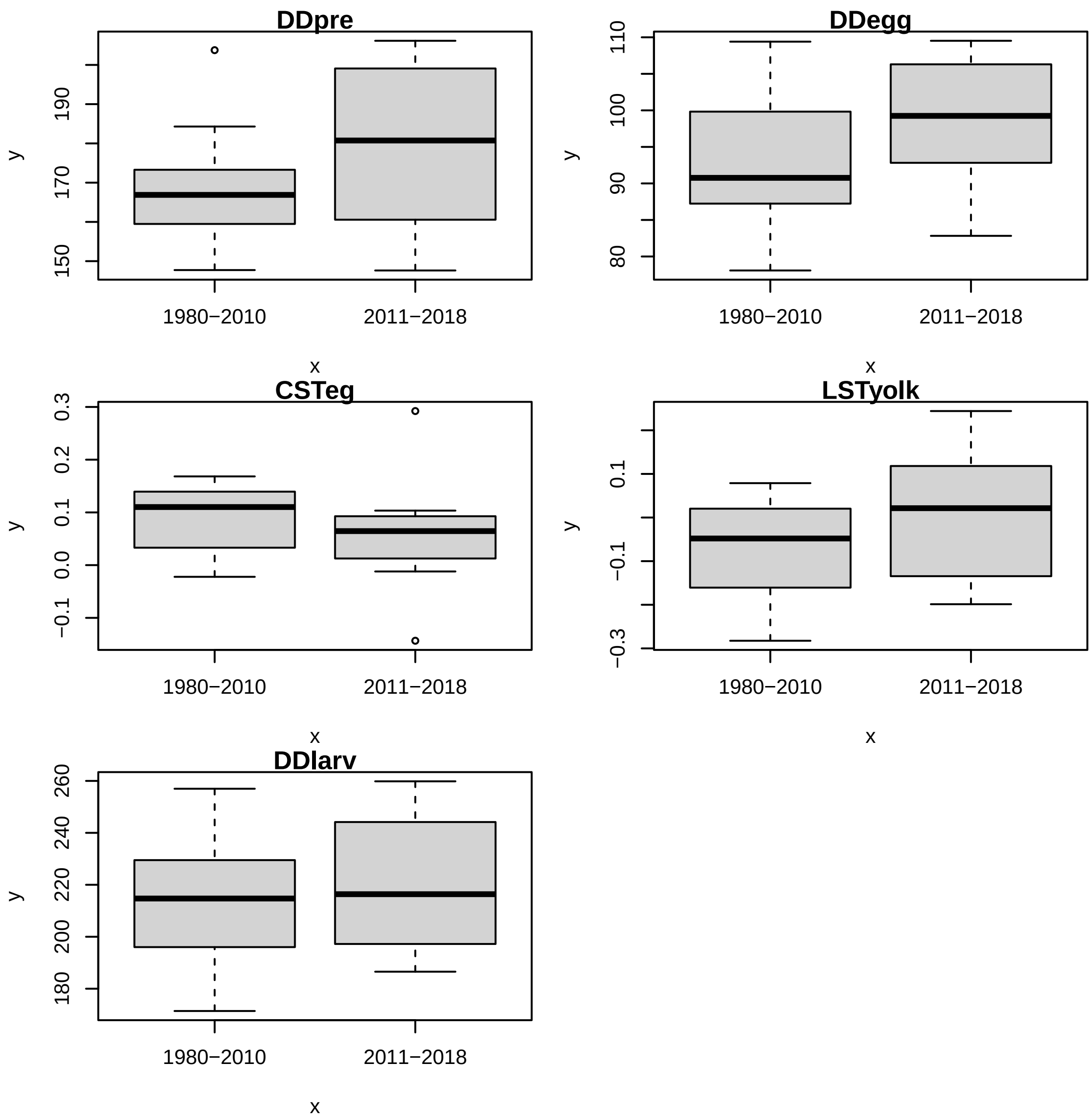


Figure 1: Box and whiskers plots of ROMS predictors using the the sablefish environment-recruitment model

For several of the ROMS timeseries, 2016 is a weird year. For the temperature related indices (DDpreD, Degg, DDLarv), there is an obvious increase in temperature and therefore DD associated with the blob years. However, there is also a substantial change in the directional trend for most indicators around 2016 or so. Both of the transport indictors show big changes in 2016 or slightly earlier at the onset of the blob.

I also had a quick look at whether the years after 2011 were any different, in terms of ROMS predictors, by

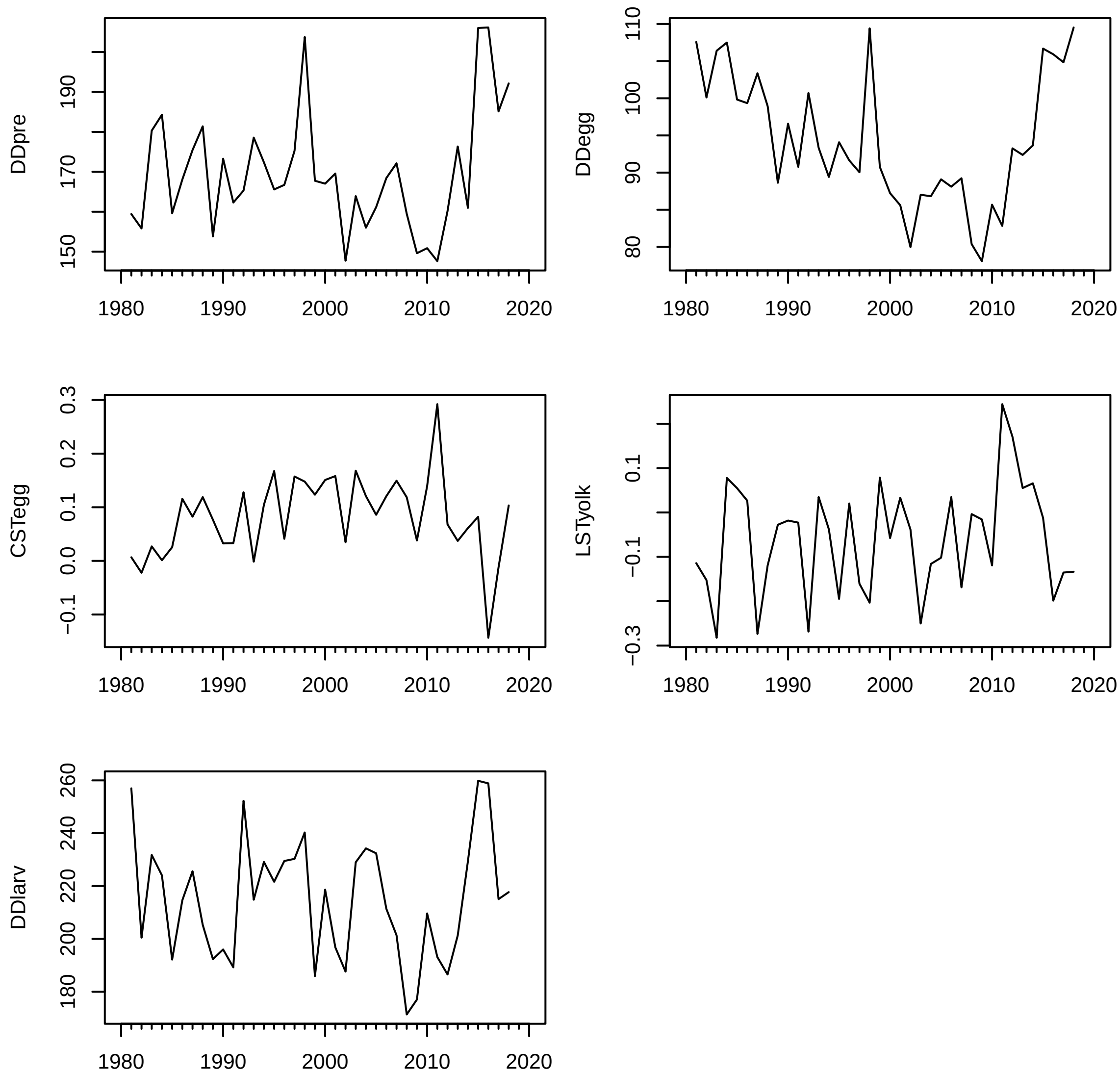


Figure 2: ROMS time series

running an nMDS on the five predictors. I did this for both the raw predictors and for normalized ROMS data. For the most part, the analysis suggests that the 'old' (1981-2010) ROMS data overlaps with the 'new' (2011-2018) data. However, for the untransformed data, 2015 and 2016 stood out as different from other years (off the the far left on the ordination plots). For the normalized data, 2016 was different (far lefft) as was 2011 (far right), but 2015 did not ordinate as extermely different. It was, however, still off to the left margin of all the data points.

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## 'comm' has negative data: 'autotransform', 'noshare' and 'wascores' set to FALSE
```

```
## Run 0 stress 0.02979083
## Run 1 stress 0.02979085
## ... Procrustes: rmse 2.070218e-05  max resid 5.945009e-05
## ... Similar to previous best
## Run 2 stress 0.02979084
## ... Procrustes: rmse 1.279365e-05  max resid 3.488096e-05
## ... Similar to previous best
## Run 3 stress 0.02979083
## ... Procrustes: rmse 5.5616e-07  max resid 1.696027e-06
## ... Similar to previous best
## Run 4 stress 0.02979083
## ... New best solution
## ... Procrustes: rmse 1.07869e-05  max resid 3.220686e-05
## ... Similar to previous best
## Run 5 stress 0.02979083
## ... Procrustes: rmse 1.106535e-05  max resid 5.249233e-05
## ... Similar to previous best
## Run 6 stress 0.02979084
## ... Procrustes: rmse 2.4381e-05  max resid 6.834609e-05
## ... Similar to previous best
## Run 7 stress 0.02979083
## ... Procrustes: rmse 6.512538e-06  max resid 1.834912e-05
## ... Similar to previous best
## Run 8 stress 0.02979087
## ... Procrustes: rmse 4.132638e-05  max resid 0.0001130804
## ... Similar to previous best
## Run 9 stress 0.02979083
## ... Procrustes: rmse 1.439949e-05  max resid 4.042836e-05
## ... Similar to previous best
## Run 10 stress 0.02979084
## ... Procrustes: rmse 1.980068e-05  max resid 5.75638e-05
## ... Similar to previous best
## Run 11 stress 0.02979083
## ... Procrustes: rmse 4.76409e-06  max resid 1.281387e-05
## ... Similar to previous best
## Run 12 stress 0.02979084
## ... Procrustes: rmse 1.588137e-05  max resid 6.996127e-05
## ... Similar to previous best
## Run 13 stress 0.02979085
## ... Procrustes: rmse 1.433938e-05  max resid 4.174733e-05
```

```
## ... Similar to previous best
## Run 14 stress 0.02979085
## ... Procrustes: rmse 2.47088e-05  max resid 6.977706e-05
## ... Similar to previous best
## Run 15 stress 0.02979087
## ... Procrustes: rmse 3.48281e-05  max resid 9.204675e-05
## ... Similar to previous best
## Run 16 stress 0.02979084
## ... Procrustes: rmse 1.90907e-05  max resid 5.526063e-05
## ... Similar to previous best
## Run 17 stress 0.02979084
## ... Procrustes: rmse 2.477735e-05  max resid 7.293882e-05
## ... Similar to previous best
## Run 18 stress 0.02979085
## ... Procrustes: rmse 2.670266e-05  max resid 7.514977e-05
## ... Similar to previous best
## Run 19 stress 0.02979083
## ... Procrustes: rmse 1.075731e-05  max resid 3.069892e-05
## ... Similar to previous best
## Run 20 stress 0.02979083
## ... Procrustes: rmse 1.19367e-05  max resid 3.368212e-05
## ... Similar to previous best
## *** Best solution repeated 17 times

## Warning in par(mfrow = c(2, 2), par = c(2, 1, 1, 1)): "par" is not a graphical
## parameter

## species scores not available

## species scores not available

## 'comm' has negative data: 'autotransform', 'noshare' and 'wascores' set to FALSE

## Run 0 stress 0.1150941
## Run 1 stress 0.1173108
## Run 2 stress 0.1173107
## Run 3 stress 0.1173107
## Run 4 stress 0.1173107
## Run 5 stress 0.1173107
## Run 6 stress 0.1150941
## ... New best solution
## ... Procrustes: rmse 3.873328e-05  max resid 0.0001382618
## ... Similar to previous best
## Run 7 stress 0.1150941
## ... Procrustes: rmse 0.0001151741  max resid 0.0005907469
## ... Similar to previous best
## Run 8 stress 0.1150941
## ... Procrustes: rmse 8.661254e-05  max resid 0.0003509923
## ... Similar to previous best
## Run 9 stress 0.1150941
## ... New best solution
## ... Procrustes: rmse 2.511363e-05  max resid 9.006418e-05
## ... Similar to previous best
```

```
## Run 10 stress 0.1150941
## ... Procrustes: rmse 2.633617e-05  max resid 0.000105237
## ... Similar to previous best
## Run 11 stress 0.1173107
## Run 12 stress 0.1150941
## ... Procrustes: rmse 6.06961e-05  max resid 0.0003059508
## ... Similar to previous best
## Run 13 stress 0.1173107
## Run 14 stress 0.1173107
## Run 15 stress 0.1173107
## Run 16 stress 0.1173107
## Run 17 stress 0.1173107
## Run 18 stress 0.1173107
## Run 19 stress 0.1150941
## ... Procrustes: rmse 7.971864e-06  max resid 3.949539e-05
## ... Similar to previous best
## Run 20 stress 0.1150941
## ... Procrustes: rmse 2.357287e-05  max resid 8.49863e-05
## ... Similar to previous best
## *** Best solution repeated 5 times
```

```
## species scores not available
```

```
## species scores not available
```

Stock recruitment data.

The original analysis examined the recruitment-environment relationship for 1981 - 2010 based on the availability of ROMS predictors.
The data from the 2016 and 2019 assessments are not exactly the same. While correlated for 1981-2010 (r = 0.94), they diverge a bit at higher values. These data are the stock-recruitment residuals calculated from the S_R relationship in the 2016 and 2019 assessments. Some of the parameters differ in those relationships. For exmaple, steepness differs between the 2016 and 2019 assessment models. So the difference could be due to both differenes in the assessment predictions overall and to changes in the stock-recruitmetn relationship.

The main point here is that the response data change and we might want to know how that influences the model fits both going forward and over the original time period.

Model comparisons for changes in recrutiment residuals and ROMS input

I fit a bunch of models to look at how both changing the source of the resisuals (2016 or 2019 assessment) and updated the ROMS predictors affect the mode performance and results. First, fit model using the best-fit terms to:

(1) 2016 assessment data for 1981-2010. This is the original best-fit model. Included here for reference.

(2) 2016 assessment data for 1981-2015. This uses the updated ROMS predictors through 2015 to determine how the original data fit when adding years. That is: is the relationship stable if we move forward in time.

(3) 2019 assessment data for 1981-2018. Again, does the relationship stay stable given an update in the assessment data and the ROMS data (from 2011 forward)

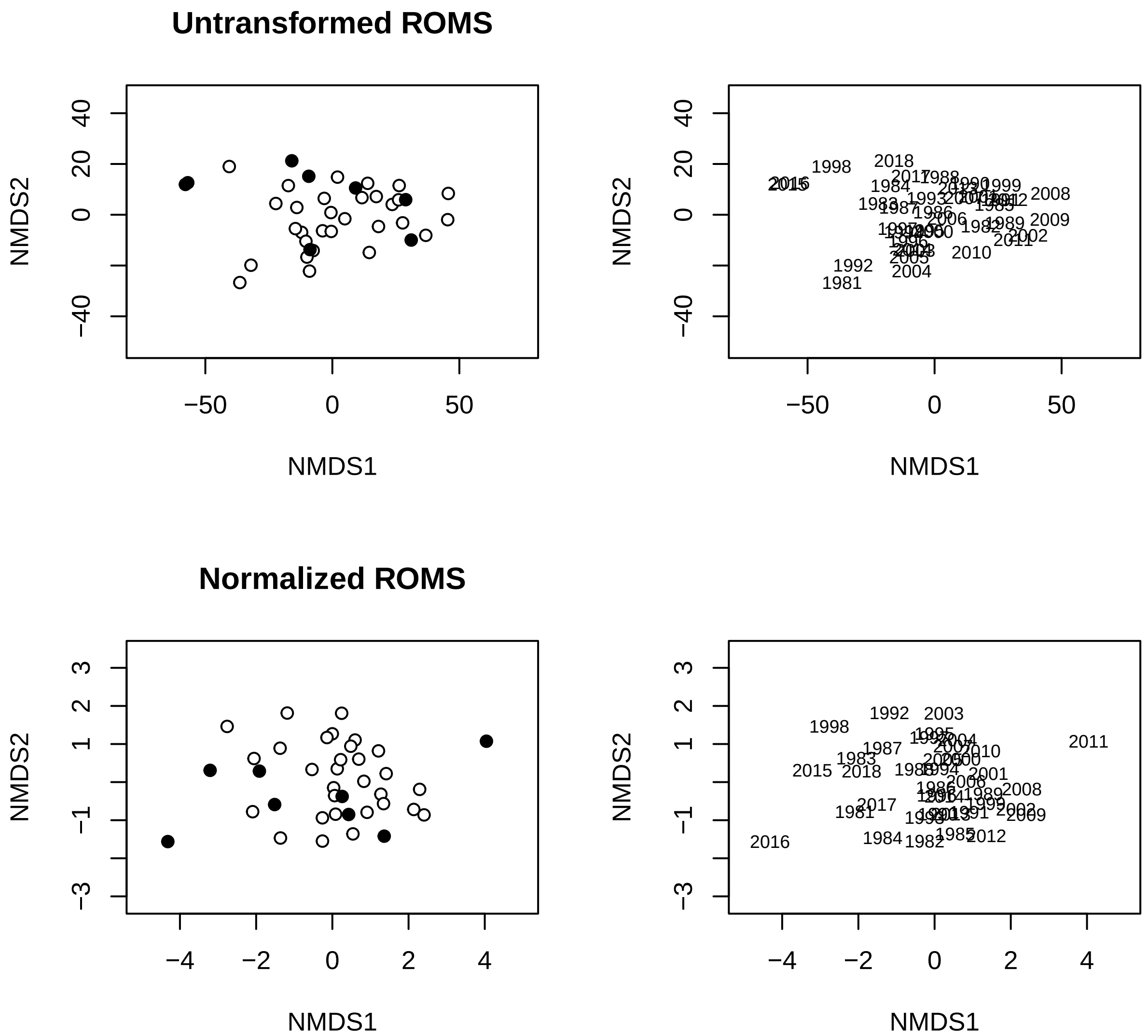


Figure 3: nMDS of the five ROMS parameters. The two upper panes show results for the raw data. In the two lower panes, the ROMS time series were normalized prior to the analysis

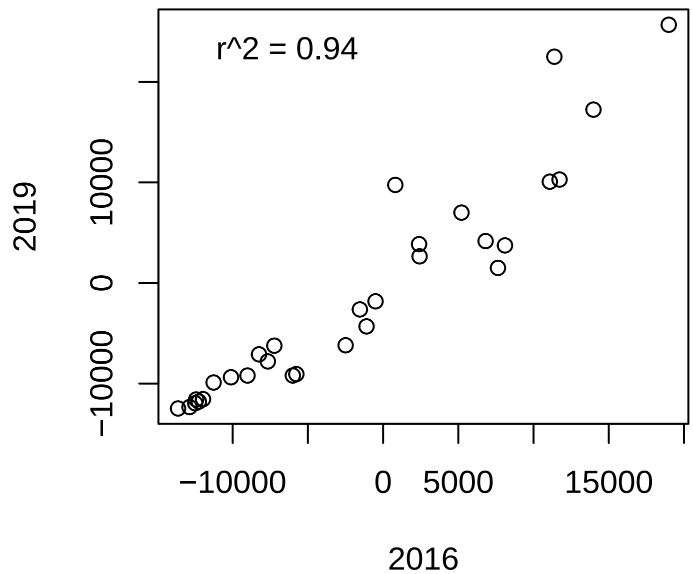


Figure 4: Correlations between residuals for the 2016 and 2019 stock assessments

The results are disappointing. The original best-fit model explained 57 % of the variation in recruitment residuals. Running the updated ROMS model through 2015 saw the r^2 value drop to $r^2 = 0.17$. Using the 2019 assessment data saw the r^2 drop to $r^2 = 0.16$.

I also looked at how well the models forecast from 2010 through 2018. Partly this examines whether the models do a good or crappy job of forecasting. Partly it looks at whether the environemnt-recruitment relationship holds up once we add new ROMS data (for 2011 - 2018).

Because updating the ROMS parameters involved an altered model (new inputs for 2011 forward), the ROMS data might differ in some way. A change in variance, for example might alter the relationship such that the slope of the relationship was qualitatively the same, but the exact value might differ, disrupting the relationship and lower the r^2. Therefore, I tried including a time blcok as a fixed, categorical factor with an interaction to allow different relationships (an coefficients) for each ROMS predictor before and after the ROMS model update. So, the time series are 1981-2010 and 2011 to NA

```
##
## Call:
## lm(formula = resids2019 ~ block + DDegg + DDpre + LSTyolk + block:DDegg +
##     block:DDpre + block:LSTyolk, data = fishb, na.action = na.fail)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14893  -5399  -1287   4825  24411
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        190221.2    95269.9   1.997 0.055007 .
```

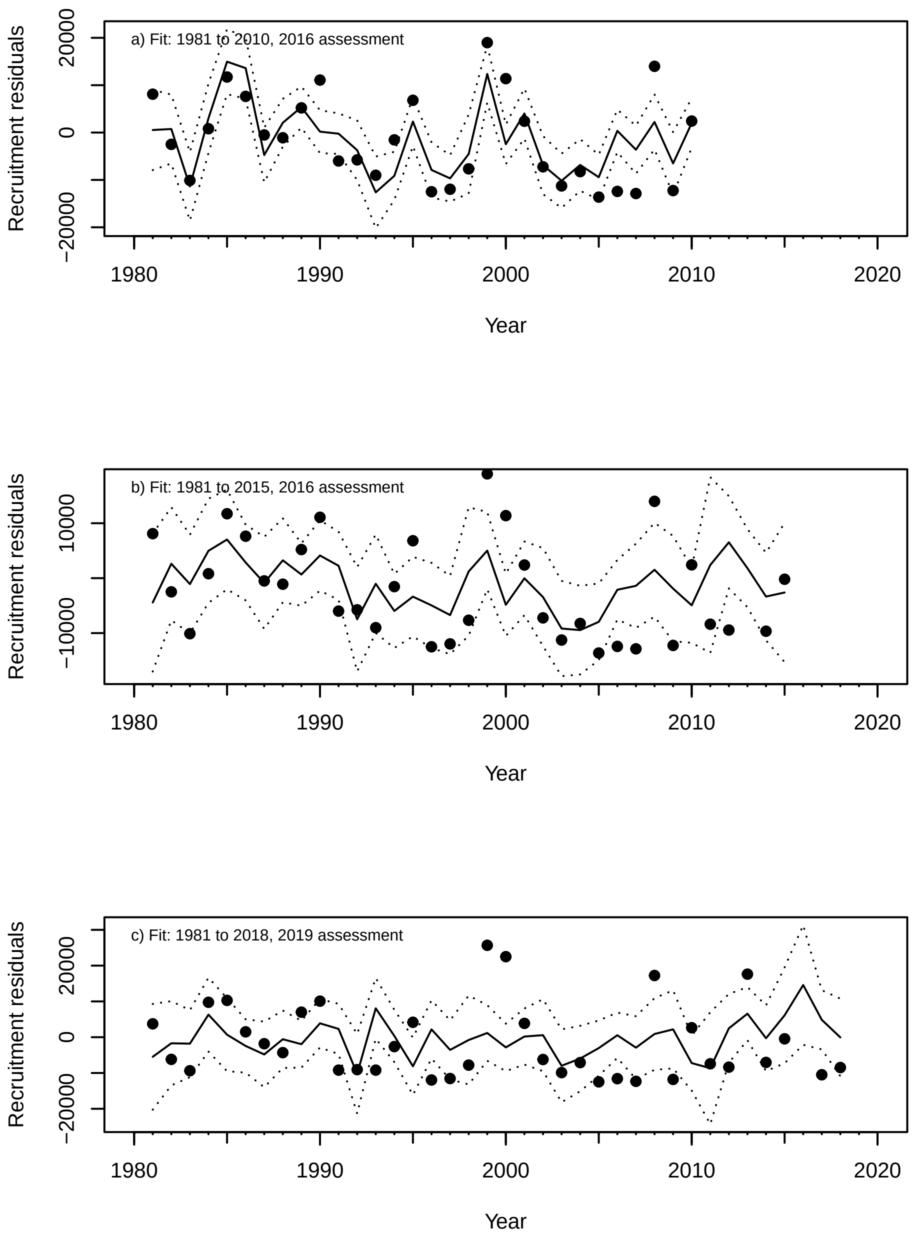
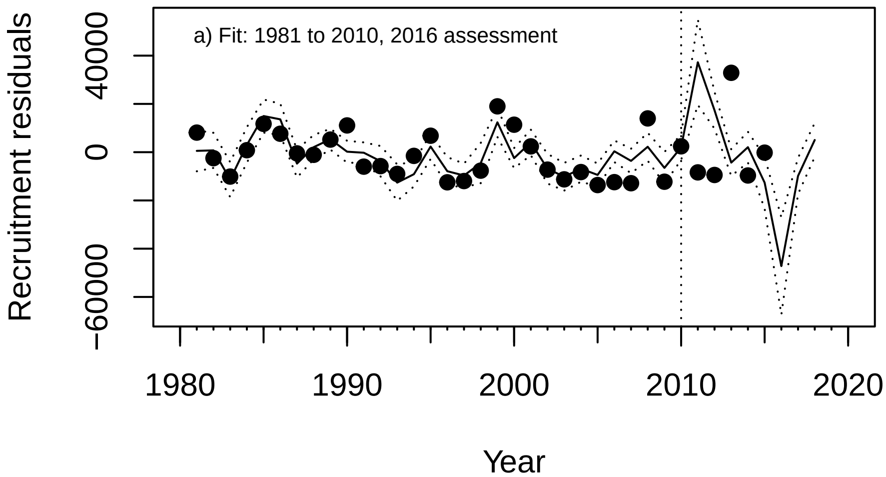


Figure 5: Model fits for three different models. Solid line is the model prediction +/- 95% CL. Points are residuals from the sablfish assessment



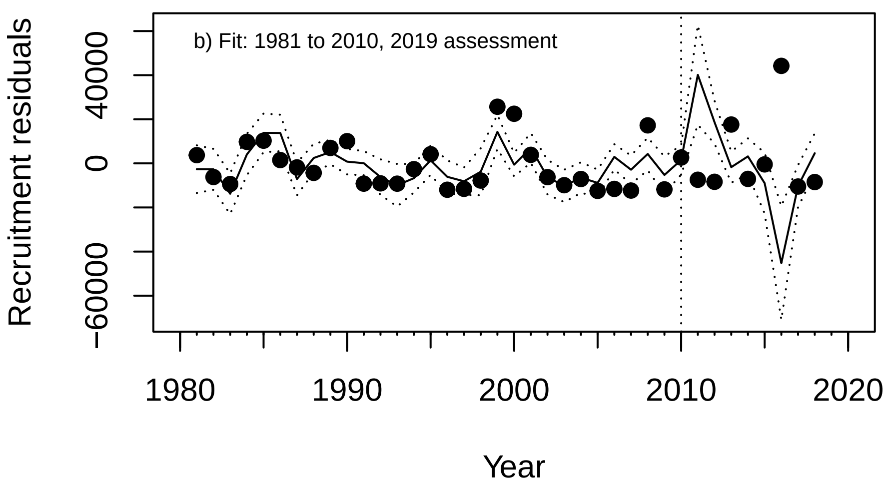


Figure 6: Model fits for two different models. Solid line is the model prediction +/- 95% CL. Points are residuals from the sablfish assessment

```
## blockOld         -194738.9    99322.1  -1.961 0.059259 .
## DDegg              -4359.7     1185.4  -3.678 0.000918 ***
## DDpre               1355.8      425.2   3.188 0.003337 **
## LSTyolk          -126899.0    59017.6  -2.150 0.039723 *
## blockOld:DDegg      4647.4     1221.6   3.804 0.000652 ***
## blockOld:DDpre     -1474.8      472.7  -3.120 0.003973 **
## blockOld:LSTyolk  172237.8    61764.6   2.789 0.009103 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10260 on 30 degrees of freedom
## Multiple R-squared:  0.4736, Adjusted R-squared:  0.3507
## F-statistic: 3.855 on 7 and 30 DF,  p-value: 0.00419
```

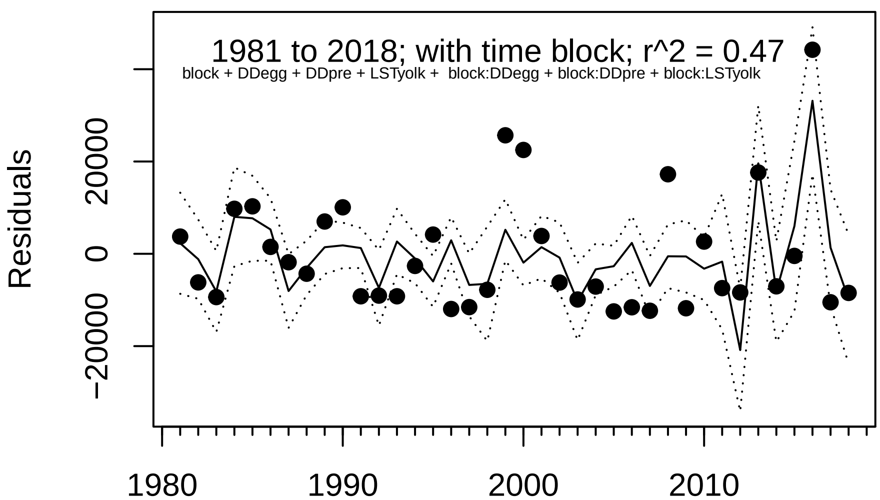


Figure 7: Solid line is the model prediction +/- 95% CL. Points are residuals from the sablfish assessment

Adding the block "helped" raising the r^2 to $r^2 = 0.47$. However, there are a lot of terms in the model, so the results are not totally sound from this analysis. I also looked at the effect in another way. Here, fit the best-fit model to the 2019 assessment residuals for 1981-2010. I then iteratively added one year at a time to look at when there might be any major changes in the coefficients and, therefore, the relationship between the ROMS drivers and sablefish recruitment. In particular, it allowed the model to predict high recruitment in 2016. The model still misses some low recruitments in 2006, 2007, 2011 and 2012, which is disappointing.

Around the blob years (or slightly earlier for some drivers), the signs of the coefficients switch for DDpre (2013), DDegg(2016), and CSTegg(2016). While the values change the signs for the coefficients for LSTyolk and DDlarv stay consistent from 2010 to 2018, when adding one year at a time to the data.

```
##                     Terms            Y2010       Y2011       Y2012       Y2013
## (Intercept) (Intercept)  3358.39608408277  1044.35902   261.73248 -4648.28535
```

```
## DDpre             DDpre  -402.70979578087   -87.38375   -56.80808    31.26063
## DDegg             DDegg  1101.16905325037   591.23204   491.61096   392.45979
## CSTegg           CSTegg  113057.935211422 27177.30036 25939.91602 15371.92087
## LSTyolk         LSTyolk  54101.3070652771 19378.17503 12040.62369 16496.41394
## DDlarv           DDlarv -213.424817238802  -206.98359  -187.73989  -181.95184
##                     Y2014       Y2015        Y2016        Y2017        Y2018
## (Intercept) -3719.43749 -6916.74163 -21225.39528 -18472.69666 -12711.85630
## DDpre          38.26439    55.98589    334.40438    308.27445    291.42168
## DDegg         398.96332   386.27961   -104.55869   -173.94362   -256.54885
## CSTegg      16708.15431 16139.23530 -48806.61746 -44274.36527 -49395.16570
## LSTyolk     14516.63196 15498.34998  11622.76400  14634.35073  15759.13395
## DDlarv       -196.37745  -188.93011    -95.57392    -60.71913    -36.93527
```

I also quickly looked at dropping 2016, which was a blob year with projected low recruitment, but in which the assessment observed high age-0 abundance. Thus, with the blob etc, this might be an anomalous year for some reason not caught in our ROMS predictors. Just dropping the 'blob year' (here defined as 2016 since it was an outlier on the nMDS) did not help.

```
##
## Call:
## lm(formula = resids2019 ~ DDpre + DDegg + CSTegg + LSTyolk +
##     DDlarv, data = fish[fish$Year %in% yrs, ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14355  -7623  -1720   6510  25750
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1810.45   25383.77   0.071    0.944
## DDpre          10.18     183.46   0.056    0.956
## DDegg         270.78     346.16   0.782    0.440
## CSTegg      15665.23   30553.75   0.513    0.612
## LSTyolk     18733.92   16218.82   1.155    0.257
## DDlarv       -144.82     107.60  -1.346    0.188
##
## Residual standard error: 10480 on 31 degrees of freedom
## Multiple R-squared:  0.1381, Adjusted R-squared:  -0.0009707
## F-statistic: 0.993 on 5 and 31 DF,  p-value: 0.4381
```

The r^2 remained low: $r^2 = 0.14$.

Allowing quadratic terms does not help. The quadratic term would allow for the relationshop to switch as the predictor got very big or small.

```
## Fixed term is "(Intercept)"
```

```
##   |                                                                      |
```

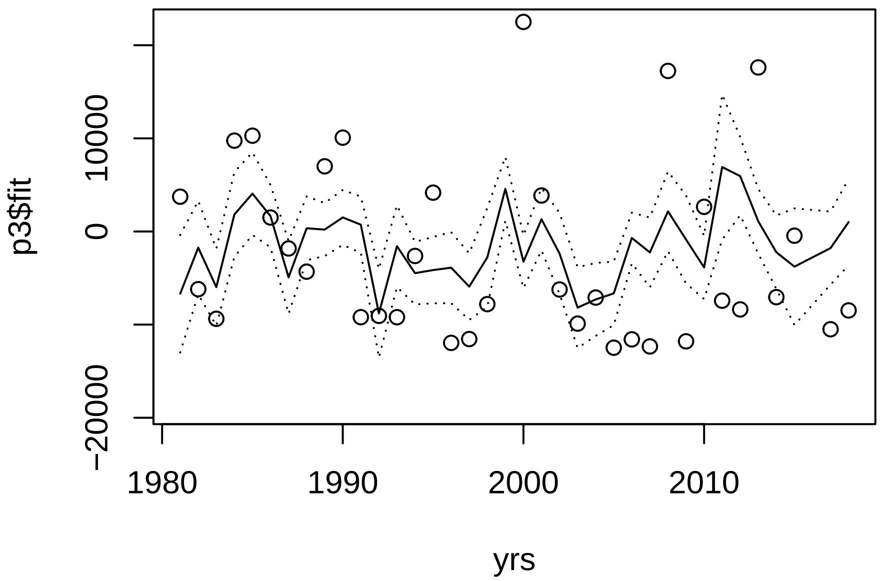


Figure 8: Model fit dropping 2016

```
## Global model call: lm(formula = resids2019 ~ DDpre + DDegg + CSTegg + LSTyolk +
##     DDlarv + I(DDpre^2) + I(DDegg^2) + I(CSTegg) + I(LSTyolk^2) +
##     I(DDlarv^2), data = dx, na.action = na.fail)
## ---
## Model selection table
##     (Intrc)  CSTgg  CSTgg DDlrv DDlrv^2 DDpre LSTyl      R2     F df   logLik
## 2      3583 -48630                                  0.08210 3.220  3 -410.955
## 3      3583        -48630                           0.08210 3.220  3 -410.955
## 4      3583 -48630                                  0.08210 3.220  3 -410.955
## 113  316200               -3258   7.269 260.7       0.18490 2.570  5 -408.700
## 1      -435                                         0.00000        2 -412.583
## 49   347200               -3227   7.407             0.11770 2.336  4 -410.203
## 65   -31260                             181.6       0.04556 1.719  3 -411.697
## 258    4728 -51010                            15450 0.10450 2.043  4 -410.485
## 259    4728        -51010                         15450 0.10450 2.043  4 -410.485
## 260    4728 -51010                            15450 0.10450 2.043  4 -410.485
## 50   300400 -38450        -2743   6.248             0.16510 2.241  5 -409.155
## 51   300400        -38450 -2743   6.248             0.16510 2.241  5 -409.155
## 52   300400 -38450        -2743   6.248             0.16510 2.241  5 -409.155
## 66   -18070 -41860                      124.3       0.10190 1.985  4 -410.542
## 67   -18070        -41860               124.3       0.10190 1.985  4 -410.542
## 68   -18070 -41860                      124.3       0.10190 1.985  4 -410.542
##      AICc delta weight
## 2   828.6  0.00  0.105
## 3   828.6  0.00  0.105
## 4   828.6  0.00  0.105
```

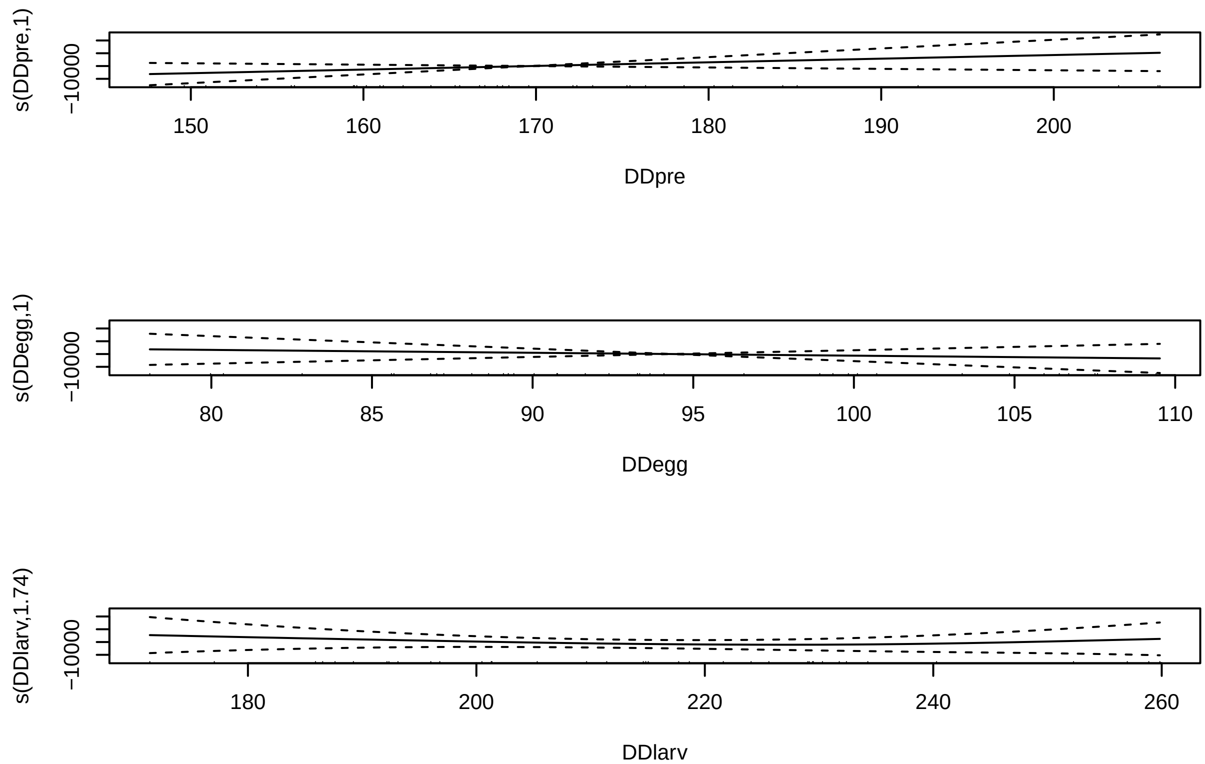
```
## 113 829.3  0.66  0.076
## 1   829.5  0.89  0.068
## 49  829.6  1.00  0.064
## 65  830.1  1.48  0.050
## 258 830.2  1.57  0.048
## 259 830.2  1.57  0.048
## 260 830.2  1.57  0.048
## 50  830.2  1.57  0.048
## 51  830.2  1.57  0.048
## 52  830.2  1.57  0.048
## 66  830.3  1.68  0.046
## 67  830.3  1.68  0.046
## 68  830.3  1.68  0.046
## Models ranked by AICc(x)
```

Likewise, adding smoothed terms in a GAM does not improve the model, nor does it suggest that the terms are non-linear.

```
## Loading required package: nlme
```

```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## resids2019 ~ s(DDpre) + s(DDegg) + CSTegg + LSTyolk + s(DDlarv)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3866       3411   1.133    0.266
## CSTegg        -41939      30410  -1.379    0.178
## LSTyolk        13628      19048   0.715    0.480
##
## Approximate significance of smooth terms:
##             edf Ref.df     F p-value
## s(DDpre)  1.000  1.000 2.084   0.159
## s(DDegg)  1.000  1.000 0.363   0.551
## s(DDlarv) 1.736  2.167 0.784   0.519
##
## R-sq.(adj) =  0.075   Deviance explained = 21.8%
## GCV = 1.8224e+08  Scale est. = 1.4993e+08  n = 38
```



What's next?

I'm not really sure what analyses to do next with the existing data. I would appreciate any thoughts. Some ideas:

(1) An obvious thing to do would be to check in with Chris Edwards (hence the email) regarding the state of the ROMS model. As I remember it, the ROMS output we received at the time was somewhat preliminary and an attempt to get us something before the stock assessment was due. The data at the time also only ran through 2018. So, if the data available, it would be worthwhile to get the updated ROMS predictors through 2019, as we have sablefish recruitment recruitment info through then.

(2) It might be worthwhile to look at all the original predictors, not just the final five in the recriutment model. We could then re-do the model fitting process and see if we come up with the same best-fit model (unlikely given that the current update through 2018 fits poorly), or whether the blob years upset the relatinship and changed the base predictors. This change might be informative for sablefish ecology in some way.

(3) We might also check with Chris E regarding any thoughts he has on changes in the model output. There do seem to be some changes in the median/mean values, but the blob is an obvious explanation for those changes. I don't see beg changes in variance or something like that, which would suggest some sort of fundamental change in the model, but might be good to talk it out. My guess is that the failure of the updated ROMS time series to predict sablefish recruitment is due to entirely different oceanic conditions around the blob years, not changes in ROMS model.

(4) One area that continues to bother me is 2006, 2007 and now 2011 and 2012. The model overpredicts these years. The SSH model also overpredicts 2006 and 2007. It would be nice to figure out why. Female

(age7+) condition was low in 2006 and 2007, but that wouldn't seem to explain the low recruitment as their condition is low after the fact.

End of my thoughts.