SEAK Pink Salmon Forecasting Process

Sara Miller

August 2023

Data

The data needed to run the code are updated in the file var*yyyy*_final.csv. The CPUE, harvest, and ISTI variables are collated by the ADF&G Ketchikan staff (Andy Piston and Teresa Fish). The satellite sea surface temperature variables are created by running the code satellite_data_monthly.R. The process for the temperature variables are then written up in the satellite_SST_process_YYYY.Rmd file. Therefore, run the code satellite_data_monthly.R and then add these temperature variables to the var*yyyy*_final.csv file. JYear is the juvenile year. The index variable stays the same unless the pink salmon forecasting group decides to change the process of the CPUE calculation for pink salmon. See the document calibration_coefficient_discussion_Nov_2020.pdf in the folder 2021_forecast. The weight_values variable was originally used to calculate a weighted MAPE and aimed to weight the current years greater than the former. This is not used and the 5-year and 10-year MAPE are used to compare the various models.

The satellite SST data, ISTI20_MJJ, and CPUEcal variables should follow the JYear from 1997 on. The SEAK catch should follow the Year variable from 1998 on.

Code

satellite_data_monthly

First, run the satellite_data_monthly.R code in the code folder. This code script will create the environmental variables needed to fill in the var*yyyy*_final.csv sheet. The monthly data (referenced in the code) needs to be manually downloaded from the site https://coastwatch.pfeg.noaa.gov/erddap/griddap/NOAA_DHW_monthly.html. Once at the site, the time is set to 1997-04-16T00:00:00Z and YYYY-07-16T00:00:00Z (where YYYY is the data year), the latitude is set to 54 and 60, and the longitude is set to -137.2 and -130. In the file type, choose '.nc-Download a NetCDF-3 binary file with COARDS/CF/ACDD metadata.' Place the file in the data folder for the current forecast year, and change the file name to 'NOAA_DHW_monthly_97_YY.nc' where YY is the data year.

The top of the script needs to be updated each year.

```
# create a folder for temperature_data
out.path <- paste0("2024_forecast/results/temperature_data/") # update year
if(!exists(out.path)){dir.create(out.path)}

# set up directories----
year.forecast <- "2024_forecast" # update year
data.directory <- file.path(year.forecast, 'data', '/')
results.directory <- file.path(year.forecast,  'results/temperature_data', '/')
```

Update all occurrences of the year. Example: NOAA_DHW_monthly_97_YY.nc where YY needs to be the current year; sst_oisst_97_YY_monthly_data.csv where YY needs to be the data year. This includes updating the figures to be 1997:YYYY where YYYY is the data year, and the file varYYYY_final.csv needs to contain the YYYY variables for the forecast year.

```
read.csv(paste0(data.directory, 'var2023_final.csv')) %>% # update the year to the data year
```

The satellite SST variables will be output into the file results/temperature_data/sst_regions_oisst_97_*yy*_monthly_data_s Then, these variables need to be copied and pasted into the var*yyyy*_final.csv sheet (the variables Chatham_SST_MJJ, Chatham_SST_May, Chatham_SST_AMJJ, Chatham_SST_AMJ, Icy_Strait_SST_MJJ, Icy_Strait_SST_May, Icy_Strait_SST_AMJJ, Icy_Strait_SST_AMJ, NSEAK_SST_MJJ, NSEAK_SST_May, NSEAK_SST_AMJJ, NSEAK_SST_AMJ, SEAK_SST_MJJ, SEAK_SST_May, SEAK_SST_AMJJ, SEAK_SST_AMJ). This seems a little backwards since this file is used in the satellite_data_monthly code, but it is only because an ISTI figure is created. The ISTI variable is not a satellite SST variable and is a SECM temperature variable. The file satellite_SST_process.Rmd does not need much updating if the same process as the prior year was used (e.g., the same latitude and longitude coordinates are used for the region of the satellite SST variables). It is helpful to run this file (satellite_SST_process.Rmd) every year so there is a record of the process. Save the outputted pdf file with a date so it does not get rewritten.

model code

To create the 18 models, the code is run in the following order;

1. 1_summarize_models.R;
2. 2_diagnostics.R;
3. 2a_diagnostics.R;
4. 3_sensitivity.R; and
5. 4_retro_analysis.R

1_summarize_models.R script

This script creates the model_summary_table1.csv, model_summary_table2.csv, model_summary_table3.csv, model_summary_table4.csv, seak_model_summary.csv, data_used_a.csv, data_used_b.csv, and a separate results_m*xx*.csv file for each model run. The columns 'model1_sim' and 'sigma' in the results_m*xx*.csv files need to be copied to the excel workbook model_summary_table_September_2022.xlsx (into each model) in the summary tables folder so that the one-step-ahead MAPE for 5 and 10 years is calculated correctly. The forecasts.csv file in the data folder is created from the results in the model_summary_table_September_2022.xlsx file. The model_summary_table5.csv file is also created from the excel workbook model_summary_table_September_2022.xlsx (although the adjusted R squared values are from the model_summary_table2.csv file). The forecast_models.png figure is also produced from this script.

The top of the script needs to be updated each year.

```
year.forecast <- "2023_forecast" # forecast year
year.data <- 2022 # last year of data
year.data.one <- year.data - 1
sample_size <-  (year.data-1998)+1 # number of data points in model (this is used for Cook's distance)
```

```
forecast2022 <- 15.6 # input last year's forecast for the forecast plot
data.directory <- file.path(year.forecast, 'data', '/')
results.directory <- file.path(year.forecast,'results', '/')
results.directory.MAPE <- file.path(year.forecast,  'results/MAPE', '/')
results.directory.retro <- file.path(year.forecast,  'results/retro', '/')
source('2023_forecast/code/functions.r')
```

In order to correctly calculate the one-step-ahead MAPE for each of the 18 models, the bias-corrected forecast needs to be calculated for each forecast of the MAPE. This is one step I have thought about deleting and just going with the non-bias corrected MAPE for the 18 models (for simplicity). So there are two choices. The first choice is not entirely correct, but it is simpler.

The two options:

(1) use the function f_model_one_step_ahead_multiple which outputs the 5-year and 10-year MAPE to the csv file seak_model_summary_one_step_ahead5.csv and seak_model_summary_one_step_ahead5.csv; or

(2) run the function f_model_one_step_ahead for each of the 18 models (which produces a results_m*x*.csv file each each model in the results folder), and take the forecast and sigma from this file (for each model) and paste it in the excel file model_summary_Table_*month_year*.xlsx. Then the bias-corrected MAPE for the 18 models is calculated in the spreadsheet.The only reason the calculation is done in excel is that I haven't figured out how to do to bias-corrected one-step-ahead MAPE in R.

Option #1 is saved in the file model_summary_table2 and option #2 is saved in the file model_summary_table3 (in the results/summary folder).

2_diagnostics.R

This script is used to explore the best model (based on the lowest one-step-ahead MAPE and group discussions). The outputs include model_summary_table4_*best_model*.csv. This csv files includes the residuals, hat values, cook's distance values, standardized residuals, and fitted values that are used to create the diagnostic figures catch_plot_pred_m*xx*.png, fitted_m*xx*.png, general_diagnostics_m*xx*.png, and influential_m*xx*.png. In addition, the top of the script outputs the lack of fit test (Bonferroni p-values), and the lack of fit curvature test.

The top of the script needs to be updated each year.

```
fit_value_model<-18.841 #best model outputs (bias-corrected); value of forecast (from model_summary_tabl
lwr_pi_80<-12.273 # 80% PI from model_summary_table2
upr_pi_80<-28.922 # 80% PI from model_summary_table2
best_model<-m11
model<-'m11'
year.forecast <- "2023_forecast" # forecast year
year.data <- 2022 #last year of data
year.data.one <- year.data - 1

# source code and functions
source('2023_forecast/code/1_summarize_models.r') # current forecast year folder
source('2023_forecast/code/functions.r') # current forecast year folder

# best model based on performance metrics
lm(SEAKCatch_log ~ CPUE + NSEAK_SST_May, data = log_data_subset) -> m11
```

2a_diagnostics.R

This script is used to explore an alternative best model. The script is very similar to the 2_diagnostics.R script.

3_sensitivity.R

This code is used to filter out certain influential years (to see the effect on the model results) but was not used in the 2023 forecast process.

4_retro_analysis.R

This script creates model hindcasts for the best models and combines them with the forecasts.csv file. The result is a data frame with hindcasts (for each model) using data up to a certain year, and then the forecast for that reduced data set. The data frame is then used to create multiple figures (e.g., year_minus_5.png) that help show how the MAPE is calculated. For example, the figure year_minus_5.png shows the hindcasts for models m1, m2, and m11 using data from 1998 to 2017 only, and the 2018 forecast based on these three models (and only using data from 1998-2017). The figures are output into the folder results/retro/figs. This script also produces the MAPE_forecasts.png figure for the best (or chosen) models which are the one-step-ahead MAPE forecasts for the chosen models.

The forecasts.csv files in the data folder needs to be created manually from the spreadsheet model_summary_Table_*month_y* in the results/summary_tables folder.

This script is very long, but is basically just repeating the process for the three models (CPUE-only model and two best models).