# WDFW SOS Analysis

Thomas Buehrens

## Contents

---

This document was generated on 12/10/2020.

---

## Requirements

All analyses require the R software (v3.4.3) for data retrieval, data processing, and summarizing model results, and the Stan Software for Hamiltonian Monte Carlo (HMC) simulation.

## Functions

We also need a couple of helper functions which we will load from the functions folder, which we will load using the walk() function from the purrr package (which we will install if it is not already installed).

```
#=========================================
# Source function files in functions folder
#=========================================
if(!require("purrr")) {
  install.packages("purrr")
  library("purrr")
}else(library(purrr))
path <- "functions/"
files <- paste0(path, list.files(path))
purrr::walk(files, source)
```

## Packages

In addition to purr, We also need a few packages that are not included with the base installation of R, so we begin by installing them (if necessary) and then loading them.

```
#=================================================
# Load packages, install and load if not already
#=================================================
using("tidyverse",
      "rstan",
      "reshape2",
      "plotly",
      "reshape2",
      "metR",
      "directlabels",
      "RColorBrewer",
      "MASS",
      "tidyr",
      "ggrepel",
      "readxl",
      "ggforce",
      "readr",
      "ggplot2",
      "gridExtra",
      "tinytex"
      )
```

## User Inputs

We need to specify data source file names, list some manual data filter conditions, and specify which ESUs and DPSs we would like to analyze to estimate trends and smoothed abundances for:

```
#=========================
#designate data file names
#=========================
#CA flatfile name
CAfilename<-"ca-data-all 09-10-2020 15 36.xlsx"
#data not in CA
NONCAfilename<-"Raw data for pops not in CA_11.10.2020.csv"
```

```r
#ESU_DPS_list file name (also lists which ESUs to use CA for)
ESU_DPS_list<-"ESU_DPS_List_2020_09_17.csv"
#ESU_DPS_list file name
ESU_DIP_list_all<-"ESU_DIP_list_all.csv"
#Recovery_Goals file name
Recovery_Goals<-"Recovery goals_formatted_9.21.2020.csv"
#WDFW SCORE files to join based on pop name with SOS outputs
WDFW_Salmonid_Stock_Inventory_Population_Recovery_Goals<-
  "WDFW-Salmonid_Stock_Inventory_Population_Recovery_Goals.csv"
WDFW_Salmonid_Stock_Inventory_Populations<-
  "WDFW-Salmonid_Stock_Inventory_Populations.csv"
Recovery_Goals_LUT_edited<-
  "recoverygoals_LUT_edited.csv"

#=======================================
#set date stamp for file names and  plots
#=======================================
#set data date--this will name files and
#folders and tell the analysis code which
#date's data to use
data_date = "2020-11-10"



#=======================================
#set manual filter conditions for CA data
#=======================================
#list of pops by ESAPOPNAME that 1) do not
#have NOSAIJ or NOSAEJ data AND 2) TSAIJ
#and TSAEJ have too many hatchery fish to
#be treated as NOSA substitutes

HatchPops<-c(#total spawners includes substantial hatchery fish)
            "Steelhead (Puget Sound DPS) Green River - winter"
          #total spawners includes substantial hatchery fish)
           ,"Salmon, Chinook (Lower Columbia River ESU) North Fork Lewis River - spring"
           #total spawners includes substantial hatchery fish)
           ,"Salmon, Chinook (Puget Sound ESU) Mid-Hood Canal - fall"
          #total spawners include many late-timed supplementation fish from HC project
           ,"Steelhead (Puget Sound DPS) East Hood Canal Tributaries - winter"
          #total spawners include many late-timed supplementation fish from HC project
           ,"Steelhead (Puget Sound DPS) Skokomish River - winter"
          #total spawners include many late-timed supplementation fish from HC project
           ,"Steelhead (Puget Sound DPS) West Hood Canal Tributaries - winter"
)
#=====================================================================
# POPFIT_exceptions list of pops where POPFIT != 1 but is "close enough"
# that we should use anyway; listed by COMMONPOPNAME2
#=====================================================================
POPFIT_exceptions<-c(#no spawners in chinook river
                    "Grays and Chinook Rivers - fall Chinook salmon",
                   #ok to use only spawners above KFH
                    "Kalama River - spring Chinook salmon",
                   #assumes no mainstem spawners
```

```r
                   "Lower Cowlitz River - late Coho salmon",
                   #assumes no mainstem spawners
                   "Lower Cowlitz River - winter Steelhead",
                   #WA only; use 1/2 recovery goal
                   "Upper Gorge Tributaries - fall Chinook salmon",
                   #WA only; use 1/2 recovery goal
                   "Upper Gorge Tributaries - late Coho salmon",
                   #WA only; use 1/2 recovery goal
                   "Lower Gorge Tributaries - late Coho salmon",
                   #assumes no fish spawn in NF and mainstem Toutle below SRS
                   "Toutle River - fall Chinook salmon",
                   # some years mislabeled as partial instead of multiple
                   "Upper Cowlitz River - spring Chinook salmon"

)
#=============================================================================
#special cases to remove in order to get 1 abundance data pt per pop per year;
#these are supplied as a listed of quoted filter conditions
#=============================================================================
specialcaselist<-list(
  #1: this dataset is a duplicate Joseph Creek dataset; newer Nez Perce Tribe
     #PIT based estimate...use ODFW estimate for now
  quote(!COMMONPOPNAME%in%c("GRJOS-s")),
  #2: #below dams estimate is sketchy
  quote(!ESAPOPNAME=="Salmon, Chinook (Lower Columbia River ESU) Lower Gorge Tributaries - fall"),
  #3: #duplicate; use WDFW estimates rather than USFWS estimates
  quote(
    !(ESAPOPNAME=="Salmon, Chinook (Upper Columbia River spring-run ESU) Entiat River - spring"
      & CONTACTAGENCY=="U.S. Fish and Wildlife Service")
  ),
  #4: duplicate; use colville tribe estimate for 2005
  quote(
    !(ESAPOPNAME=="Steelhead (Upper Columbia River DPS) Okanogan River - summer"
      & METHODNUMBER==2
      & SPAWNINGYEAR == 2005)
  ),
  #5lower gorge coho use WDFW only, not ODFW data
  quote(
    !(ESAPOPNAME=="Salmon, coho (Lower Columbia River ESU) Lower Gorge Tributaries - late"
      & SUBMITAGENCY=="ODFW")
  ),
  #6: Entiat data lists both patch occupancy and old method as best.
     #use old method until they get patch occupancy cleaned up for upper C
  quote(
    !(ESAPOPNAME=="Steelhead (Upper Columbia River DPS) Entiat River - summer"
      & METHODNUMBER==2)
  ),
  #7: Elwha chinook data loaded as of 9/17/2020 is wrong
     #use data Neala Kendall got from NOAA/Tribe/Joe A instead
  quote(
    !(ESAPOPNAME=="Salmon, Chinook (Puget Sound ESU) Elwha River - fall")
  ),
  #8: Sammammish Fall chinook data loaded as of 9/17/2020 is wrong
```

```
    #popfit should be partial, therefore excluded
  quote(
    !(ESAPOPNAME=="Salmon, Chinook (Puget Sound ESU) Sammamish River - fall")
  )
)


#==========================================
#Set parameters for summarization of results
#==========================================
#Only include population in ESU summary that include recovery goals?
Withgoalsonly="yes"
#Number of years to calculate geomean of smoothed abundance
geomeanyears = 5
#last year in geomean abundance calculations
lastyear=2019
#set number of years for forward projection of ESUs
futureyears = 5
#exlcude populations from geomean smoothed abundance calculation
#that have no new observed data in period geomean is being calculated for?
filtergeomeansforpopswithnonewdata="No"
```

## Data Preparation

Here we will use a data filtering algorithm described in the full report to select appropriate natural origin spawner abundance data for use in the status and trend analysis, we will prepare analysis input files, and we will plot the raw abundance data. DESCRIBE ALGORITHM HERE. the program will look for NOSAIJ or NOSAEJ data first and then proceed to TSAIJ and TSAEJ data only if NOSA data does not exist.

```
#prep and filter ca data
dat<-prepCAdata(CAfilename = CAfilename,
                data_date = data_date,
                ESU_DPS_list = ESU_DPS_list,
                Recovery_Goals = Recovery_Goals,
                POPFIT_exceptions = POPFIT_exceptions,
                specialcaselist = specialcaselist)

#prep NON CA data (data must be pre-filtered/final )
dat2<-prepNONCAdata(data_date=data_date,
                    NONCAfilename = NONCAfilename,
                    ESU_DPS_list = ESU_DPS_list)

#make analysis files for status and trend analysis
#(we use bind_rows to combine CA and Non-CA data)
makefiles(data=bind_rows(dat[,colnames(dat)%in%colnames(dat2)],dat2),
          data_date=data_date)

#make plots of raw abundance data by population and ESU/DPS
plotfunc(data=bind_rows(data=dat[,colnames(dat)%in%colnames(dat2)],dat2),
         data_date=data_date,
         ESU_DPS_list=ESU_DPS_list,
         Recovery_Goals=Recovery_Goals,
         Withgoalsonly=Withgoalsonly)
```

## Run Status and Trend Analysis

Here we will run the status and trend analysis using Stan via rstan. We will then tidy up the results to prepare for summarization and plotting. You must have rtools installed and stan installed for this to work:

```r
#run analysis for any populations that is not yet complete
analyzedata(data_date=data_date,
            ESUsubset= c(# "Lower Columbia coho"
                # ,"Lower Columbia Chinook"
                # ,"Lower Columbia steelhead"
                # ,"Mid-Columbia steelhead"
                # ,"Snake River spring and summer Chinook"
                # ,"Snake River steelhead"
                # ,"Upper Columbia spring Chinook"
                # ,"Upper Columbia steelhead"
                # ,"Snake River fall Chinook"
                # ,"Puget Sound steelhead"
                # ,"Puget Sound Chinook"
                # ,"Ozette Lake sockeye"
                # ,"Hood Canal summer chum"
                # ,"Lower Columbia chum"
              ),
            ESU_DPS_list=ESU_DPS_list,
            cores=4,
            chains = 4,
            iter = 2000,
            warmup= 1000,
            thin = 1,
            control=list(adapt_delta=0.9995)
            )

#combine results output
combineresults(data_date=data_date,Recovery_Goals=Recovery_Goals)

#merge results with raw data
resultsdata<-makeresultsdata(data_date=data_date,
                             data=bind_rows(dat[,colnames(dat)%in%colnames(dat2)],dat2),
                             ESU_DPS_list=ESU_DPS_list,Recovery_Goals=Recovery_Goals,
                             Smoothed_Abundance=Smoothed_Abundance)
```

## Summarize Status and Trend Results

Here we will summarize our status and trend results, first at the population level, and then at the ESU/DPS level. This code may take up to 5-15 minutes to run; adjust expectations accordingly.

```r
#make plots of raw abundance data by population and ESU/DPS
#with smoothed abundances added as lines
resultsplots<-plotfunc2(resultsdata=resultsdata,
                        Recovery_Goals=Recovery_Goals,
                        Withgoalsonly=Withgoalsonly,
                        data_date=data_date
                        )
```

```
# #make ESU and DIP Results
ESU_DIP_results<-make_ESU_DIP_results(
  resultsdata=resultsdata,
  lastyear=lastyear,data_date =data_date,
  geomeanyears=geomeanyears,
  filtergeomeansforpopswithnonewdata=filtergeomeansforpopswithnonewdata
  )


#plot results
esu_results_plot(ESU_DIP_results=ESU_DIP_results,
                 data_date=data_date,
                 futureyears=futureyears,
                 resultsplots=resultsplots,
                 bypopsplots="No"
                 )
```

# Format Data for Data.wa.gov

Here we will merge our raw population name list, recovery goals, and final abundance data used for analysis with the population name list used by data.wa.gov in order to display the data on the web. Unfortunately, this portion of the program involves two manual steps. First, a lookup table ("recoverygoals_LUT_raw.csv") is created to compare the population set from the SOS analysis with that in data.wa.gov using the function make_data_wa_gov_recover_goals_lut. This lookup table uses character matching to match names and needs manual inspection and proofing. Once this function is done, open, inspect, and edit as necessary "recoverygoals_LUT_raw.csv" and re-save as "recoverygoals_LUT_edited.csv" in the data folder. You many then run the following 3 functions: make_data_wa_gov_recovery_goals, make_escapement_data_wa_gov,make_population_data_wa_gov. The final file these functions creates ("Population_Data_For_data_wa_gov_raw.csv") will have missing information for populations that are not listed in the inventory of populations on data.wa.gov. This file should be downloaded, inspected, edited, and re-sawed as "Population_Data_For_data_wa_gov.csv" and is then ready for upload to data.wa.gov

```
#==========================
#make tables for data.wa.gov
#==========================
make_data_wa_gov_recovery_goals_LUT(data_date = data_date,
                          Recovery_Goals=Recovery_Goals,
                          WDFW_Salmonid_Stock_Inventory_Population_Recovery_Goals=
                            WDFW_Salmonid_Stock_Inventory_Population_Recovery_Goals
                        )
make_data_wa_gov_recovery_goals(data_date = data_date,
                          Recovery_Goals=Recovery_Goals,
                          Recovery_Goals_LUT_edited=Recovery_Goals_LUT_edited,
                          ESU_DPS_list
                        )
make_escapement_data_wa_gov(data_date=data_date,
                          Recovery_Goals_LUT_edited=Recovery_Goals_LUT_edited
                        )
make_population_data_wa_gov(data_date=data_date,
                          Recovery_Goals=Recovery_Goals,
                          Recovery_Goals_LUT_edited=Recovery_Goals_LUT_edited,
                          WDFW_Salmonid_Stock_Inventory_Populations=
```

```
                        WDFW_Salmonid_Stock_Inventory_Populations
                    )
```

# Reproducing this pdf or html page

In order to reproduce this pdf or html page you need to have a LaTex application installed. Running this snippet of code will automatically install tinytex on your machine so you can render pdfs and html:

```
#tinytex::install_tinytex()
```