Lab 5

Sawyer Balint

Fall 2024; Marine Semester Block 3

Contents

1 Introduction

This document is available at https://github.com/sjbalint/BI521/tree/main/scripts/labs

```
#import packages
library(tidyverse) #for data wrangling
library(here) #for filepath management
library(ggsci) #for colors
library(scales) #for log axis breaks

#custom graphing theme

update_geom_defaults("point", list(shape = 21, fill="grey", stroke=0.8))

theme <- list(
  theme_classic(),
  scale_color_viridis_d(option="inferno", end=0.8),
  scale_fill_viridis_d(option="inferno", end=0.8),
  theme(legend.position="right"),
  labs(x="Time")
)
```

2 Part I

2.1 Task 1

```r
#function for S-Ricker model
gR_SRicker <- function(S,mu){
  with(mu,S^(1+delta)*exp(r*(1-S/K)))
}

SRicker_sim <- function(S0, mu, n_iter){

  R0 <- gR_SRicker(S0, mu)

  df <- data.frame(S=S0, R=R0, t=0)

  #empty list to store results
  result.list <- list(df)

  i_model <- 1

  while(i_model<n_iter){

    i_model <- i_model + 1

    R <- gR_SRicker(S0, mu)
    S <- with(mu, R + Sa*S0)

    df <- data.frame(S=S, R=R, t=i_model)

    S0 <- S
    R0 <- R

    #log results
    result.list <- c(result.list, list(df))
  }

  #compile results
  result.df <- bind_rows(result.list)

  return(result.df)

}

#try it out
mu <- list(r=2.6, K=3/4, delta=1, Sa=0.6)
df <- SRicker_sim(S0=0.04, mu=mu, n_iter=50)

#looks good
head(df)
```

```
##            S         R t
## 1 0.04000000 0.01875269 0
## 2 0.04275269 0.01875269 2
## 3 0.04687066 0.02121905 3
## 4 0.05326450 0.02514210 4
## 5 0.06371637 0.03175768 5
## 6 0.08205658 0.04382675 6
```

```r
#loop over a range of S0
S0.list <- seq(0.01,0.1, by=0.01)

result.list <- list()

for (S0_i in S0.list){

  df <- SRicker_sim(S0=S0_i, mu=mu, n_iter=20) %>%
    mutate(S0 = S0_i)

  result.list <- c(result.list, list(df))

}

result.df <- bind_rows(result.list) %>%
  mutate(S0 = as.factor(S0))

ggplot(result.df, aes(t,S,color=S0, group=S0))+
  theme+
  geom_line()+
  geom_hline(yintercept=0, linetype="dashed")
```

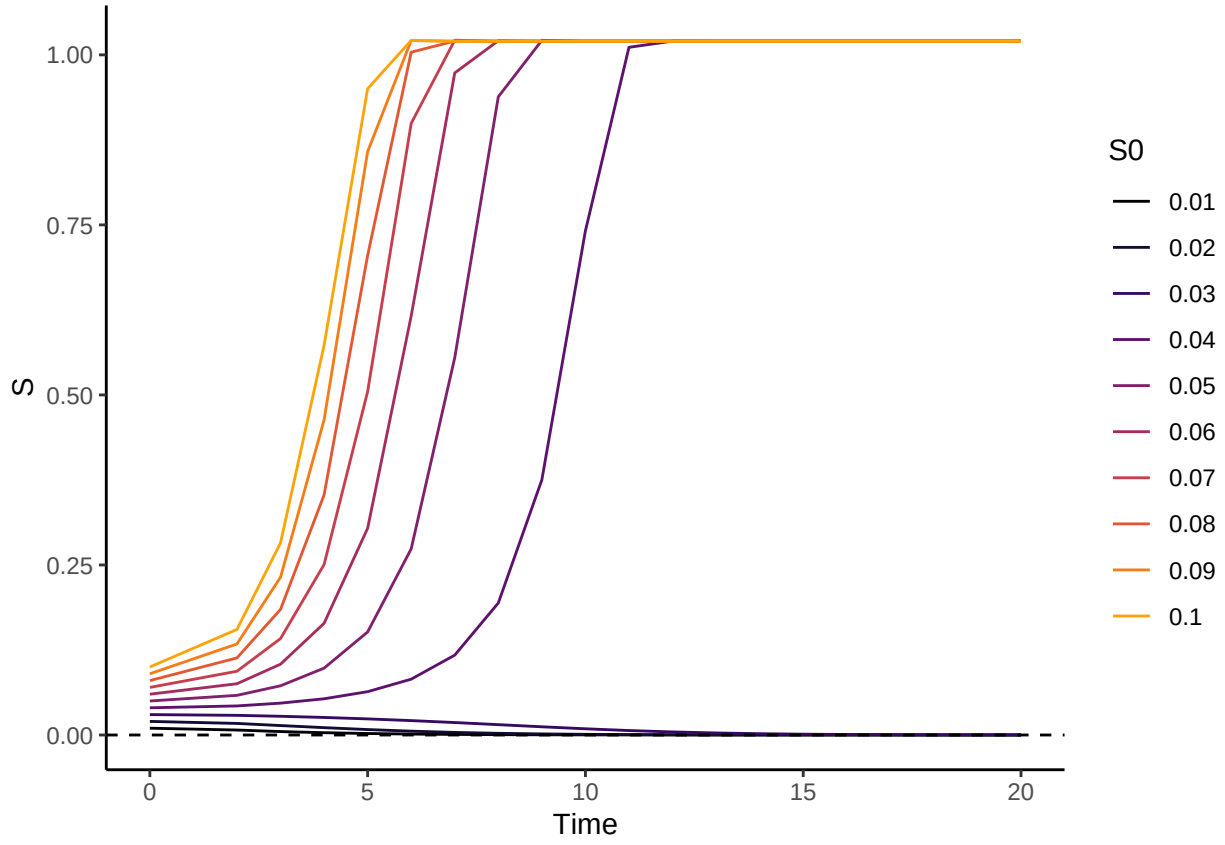

Fig. 1: Sigmodal Ricker with *S_0* values ranging from 0.01 to 0.1 (color). We find that S_0 equal to 0.01, 0.02, and 0.03 result to a population that crashes to zero.

2.2 Task 2

```r
#new function that includes fishing mortality
SRicker_Fm_sim <- function(S0, mu, Fm, n_iter){

  R0 <- gR_SRicker(S0, mu)

  df <- data.frame(S=S0, R=R0, t=0)

  #empty list to store results
  result.list <- list(df)

  i_model <- 1

  while(i_model<n_iter){

    i_model <- i_model + 1

    R <- gR_SRicker(S0, mu)
    S <- with(mu, (R + Sa*S0)*(1-Fm))

    df <- data.frame(S=S, R=R, t=i_model)

    S0 <- S
    R0 <- R

    #log results
    result.list <- c(result.list, list(df))
  }

  #compile results
  result.df <- bind_rows(result.list)

  return(result.df)

}

Fm.list <- seq(0,0.8, by=0.1)
S0=0.5

result.list <- list()

for (Fm_i in Fm.list){

  df <- SRicker_Fm_sim(S0=S0, mu=mu, Fm=Fm_i, n_iter=20) %>%
    mutate(Fm = Fm_i)

  result.list <- c(result.list, list(df))

}

result.df <- bind_rows(result.list) %>%
  mutate(Fm = as.factor(Fm))
```

```
#make a plot
ggplot(result.df, aes(t,S,color=Fm, group=Fm))+
  theme+
  geom_line()+
  geom_hline(yintercept=0, linetype="dashed")
```

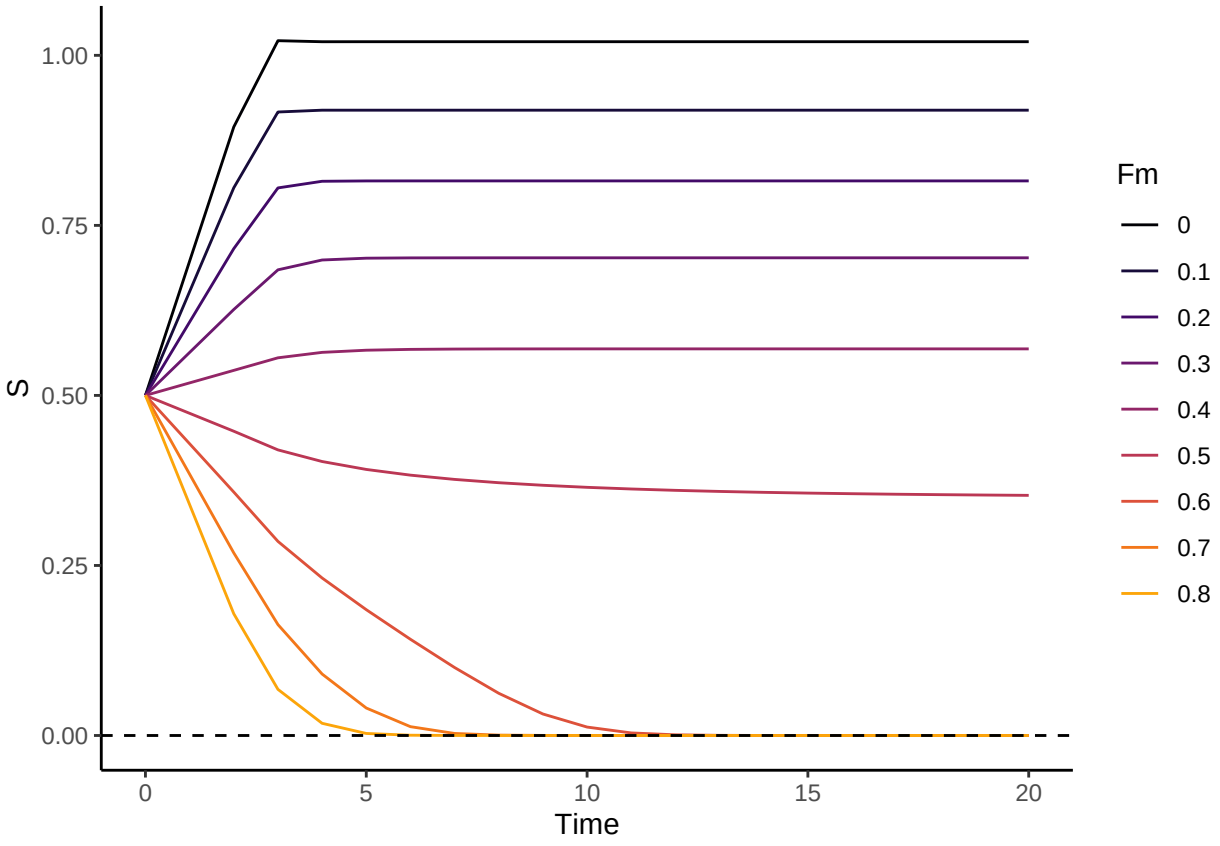

Fig. 2: Higher fishing mortality results in a lower equilibrium population level and a longer time to reach equilibrium. At mortalities of 0.6, 0.7, and 0.8, the population crashes to zero.

This simulation implements a constant effort model of fishing.

2.3 Task 3

```
#function for S-Ricker model with noise
gR_SRicker_noise <- function(S,mu){
  with(mu,rlnorm(1, meanlog=(1+delta)*log(S)+(r*(1-(S^(1))/K)), sdlog=tau_r))
}

#new function that includes noise
SRicker_Fm_noise_sim <- function(S0, mu, Fm, n_iter){

  R0 <- gR_SRicker_noise(S0, mu)
```

```
  df <- data.frame(S=S0, R=R0, t=0)

  #empty list to store results
  result.list <- list(df)

  i_model <- 1

  while(i_model<n_iter){

    i_model <- i_model + 1

    R <- gR_SRicker_noise(S0, mu)
    S <- with(mu, (R + Sa*S0)*(1-Fm))

    df <- data.frame(S=S, R=R, t=i_model)

    S0 <- S
    R0 <- R

    #log results
    result.list <- c(result.list, list(df))
  }

  #compile results
  result.df <- bind_rows(result.list)

  return(result.df)

}

S0=0.2

result.list <- list()

for (Fm_i in seq(0,0.6, by=0.2)){

  for (taur_r_i in c(0,0.1)){
  mu <- list(r=2.6, K=3/4, delta=1, Sa=0.6, tau_r=taur_r_i)

  df <- SRicker_Fm_noise_sim(S0, mu, Fm=Fm_i, n_iter=20) %>%
    mutate(tau_r=taur_r_i,
           Fm=Fm_i)

  result.list <- c(result.list, list(df))

  }

}

result.df <- bind_rows(result.list) %>%
  mutate(tau_r=factor(tau_r),
         Fm = factor(Fm))
```

```
#make a plot
ggplot(result.df, aes(t,S,color=Fm, linetype=tau_r))+
  theme+
  geom_line()+
  geom_hline(yintercept=0, linetype="dashed")
```

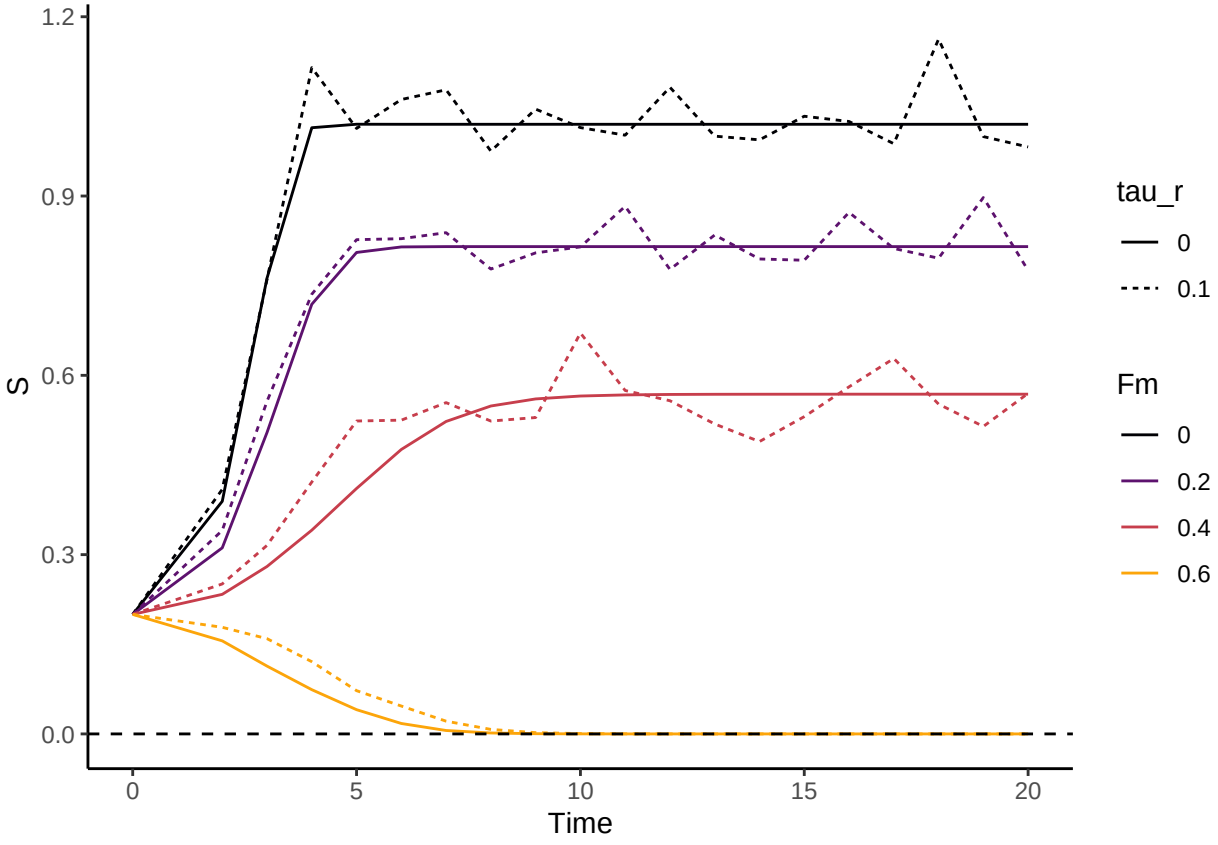

Fig. 3: I modified S0 and the range of fishing mortality tested. The additional "noise" of *tau_r* = 1 is observed in the dashed lines.

This version is including process error - variations in the actual population of the fishery, rather than noise stemming from incomplete counting.