# Lab 4

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_jama(),
  scale_fill_jama(),
  theme(legend.position="inside",
        legend.position.inside=c(0.8,0.3))
)
```

```
#import data
scallops.df <- data.frame(
  length_bin=c(65,80,95,125,150,190),
  catch_at_length=c(0,1,6,9,10,9),
  escapes_at_length=c(10,9,4,1,0,1)
)
```

# 2 1D parameter fitting

## 2.1 Task 1

```
#selectivity function
f_selectivity_1d <- function(l,c){
  exp(c*(l-100))/(1 + exp(c*(l-100)))
}

#sum of squares
SS_f_selectivity_1d <- function(c, length_classes, catch_rates){
  squared_errors <- (f_selectivity_1d(length_classes, c) - catch_rates)^2
  sum_squares <- sum(squared_errors)
  return(sum_squares)
}
```

## 2.2 Task 2

```
c.v <- seq(0.01,2,0.01)
SS.v <- vector()

catch_rates <- scallops.df$catch_at_length/10

for (c in c.v){
  SS <- SS_f_selectivity_1d(c, scallops.df$length_bin, catch_rates)
  SS.v <- c(SS.v, SS)
}

result.df <- data.frame(c=c.v,
                        ss=SS.v)

ggplot(result.df, aes(c, ss))+
  theme+
  geom_point()+
```

```
  geom_line()+
  labs(x="C", y="Sum of Squares")
```

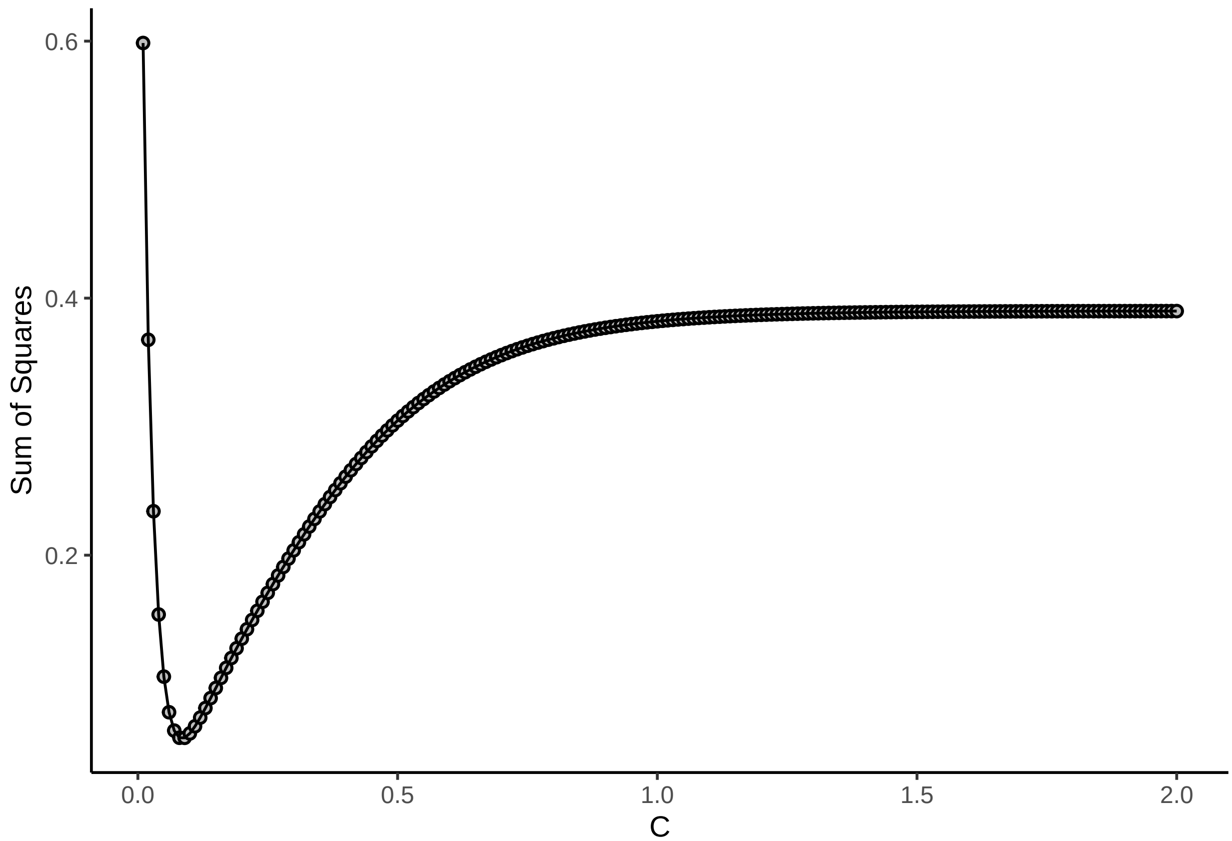



**Fig. 1:** There is a well-defined relative minimum in sum-of-squares at a $c$ value of around 0.1.

```
#exhaustive search
c_min_exh <- c.v[which.min(SS.v)]

ggplot(scallops.df, aes(length_bin, catch_at_length/10))+
  theme+
  geom_point()+
  geom_function(fun = ~f_selectivity_1d(.x, c_min_exh))+
  labs(x="Length", y="Catch Rate", title=paste("Least-squares fit, c=",c_min_exh))
```

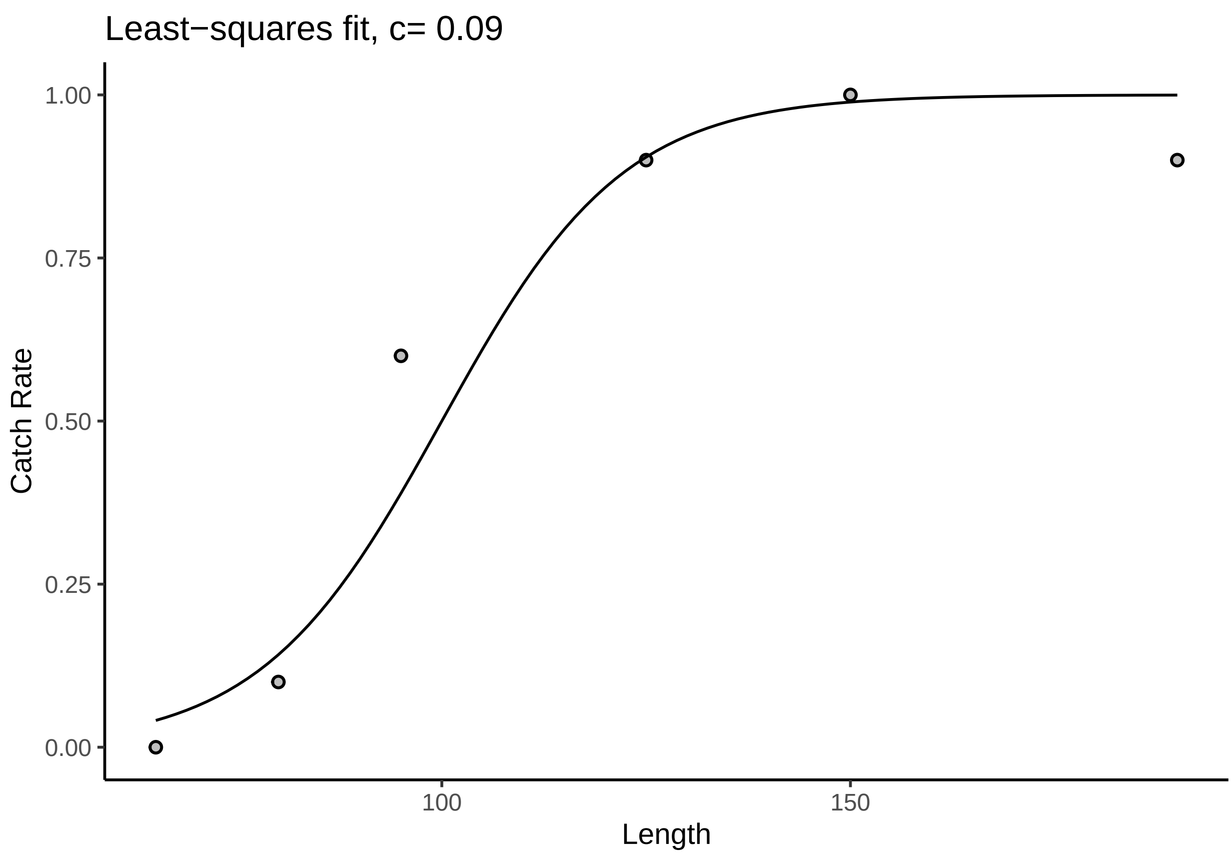



**Fig 2:** Using the exhaustive search, we now have a least-squared selectivity curve. We can observe that selectivity is near zero at lengths of aroung 50mm and near 100% ar lengths of 150mm.

### 2.3 Task 3

```
c_min_opt <- optim(0.1,fn=function(c) SS_f_selectivity_1d(c, scallops.df$length_bin, catch_rates),
                   method="BFGS")$par

c_min_exh <- c.v[which.min(SS.v)]

ggplot(scallops.df, aes(length_bin, catch_at_length/10))+
  theme+
  geom_point()+
  geom_function(fun = ~f_selectivity_1d(.x, c_min_opt), aes(color="optim()"))+
  geom_function(fun = ~f_selectivity_1d(.x, c_min_exh), aes(color="Exhaustive Search"))+
  labs(x="Length", y="Catch Rate", title=paste("Least-squares fit, c =",c_min_exh), color="Method")
```

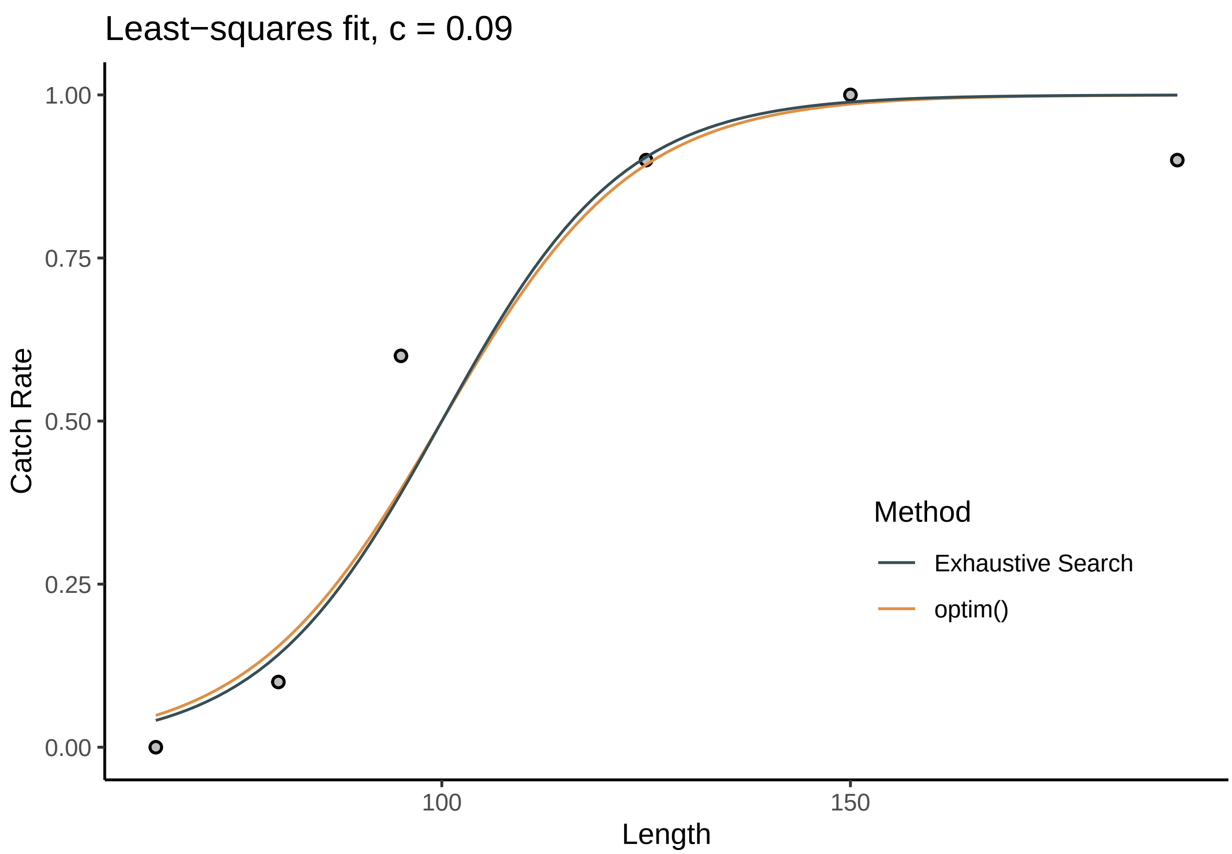



**Fig. 3:** Exhaustive search and optimized parameters provide comparible, albiet slightly different, selectivity curves.

## 2.4 Task 4

```
LL_f_selectivity_1d <- function(c, length_classes, catch_at_length, escape_at_length){

  LL_out <- 0

  for (i_length_class in 1:length(length_classes)){

    length_class_i <- length_classes[i_length_class]
    LL_i <- catch_at_length[i_length_class]*log(f_selectivity_1d(length_class_i,c))+
      escape_at_length[i_length_class]*log(1-f_selectivity_1d(length_class_i,c))

    LL_out <- LL_out + LL_i

  } #for length class

  return(LL_out)

} #function
```

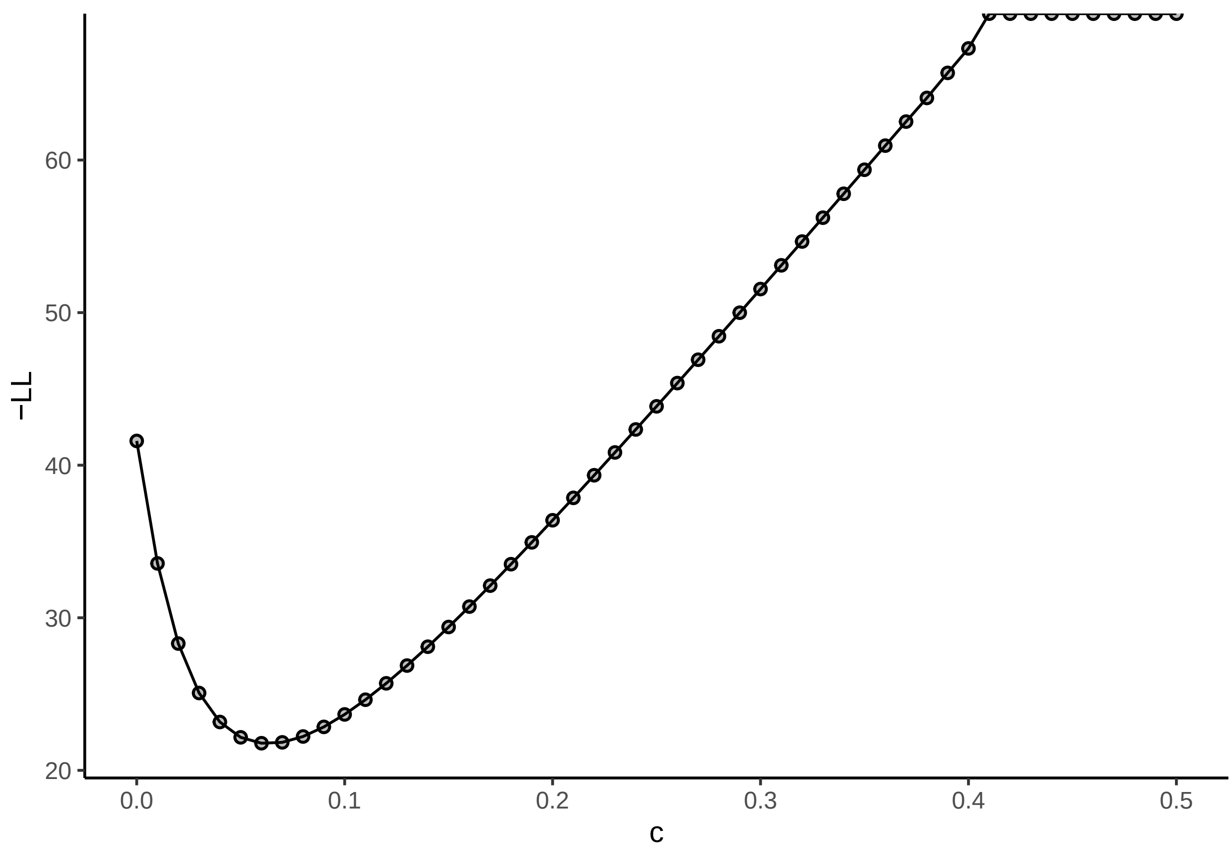
## 2.5 Task 5

```
f_LL_scallop_1d <- function(x) LL_f_selectivity_1d(x,
                                                  scallops.df$length_bin,
                                                  scallops.df$catch_at_length,
                                                  scallops.df$escapes_at_length)

c.v <- seq(0,0.5,0.01)

plot.df <- data.frame(c=c.v,LL=f_LL_scallop_1d(c.v))

ggplot(plot.df, aes(c,-LL))+
  theme+
  geom_point()+
  geom_line()
```




> **Fig. 4:** As with the sum-of-squares, there is a well-defined relative minimum of $LL$ at $c$ values around 0.06.

## 2.6 Task 6

```
c_r_opt <- optim(0.1, fn = function(c) -f_LL_scallop_1d(c), method="BFGS")$par
```

```
c_r_exh <- c.v[which.min(-f_LL_scallop_1d(c.v))]

ggplot(scallops.df, aes(length_bin, catch_at_length/10))+
  theme+
  geom_point()+
  geom_function(fun = ~f_selectivity_1d(.x, c_r_opt), aes(color="optim()"))+
  geom_function(fun = ~f_selectivity_1d(.x, c_r_exh), aes(color="Exhaustive Search"))+
  labs(x="Length", y="Catch Rate", title=paste("Maximum-likelihood fit, c =",c_r_exh), color="Method")
```

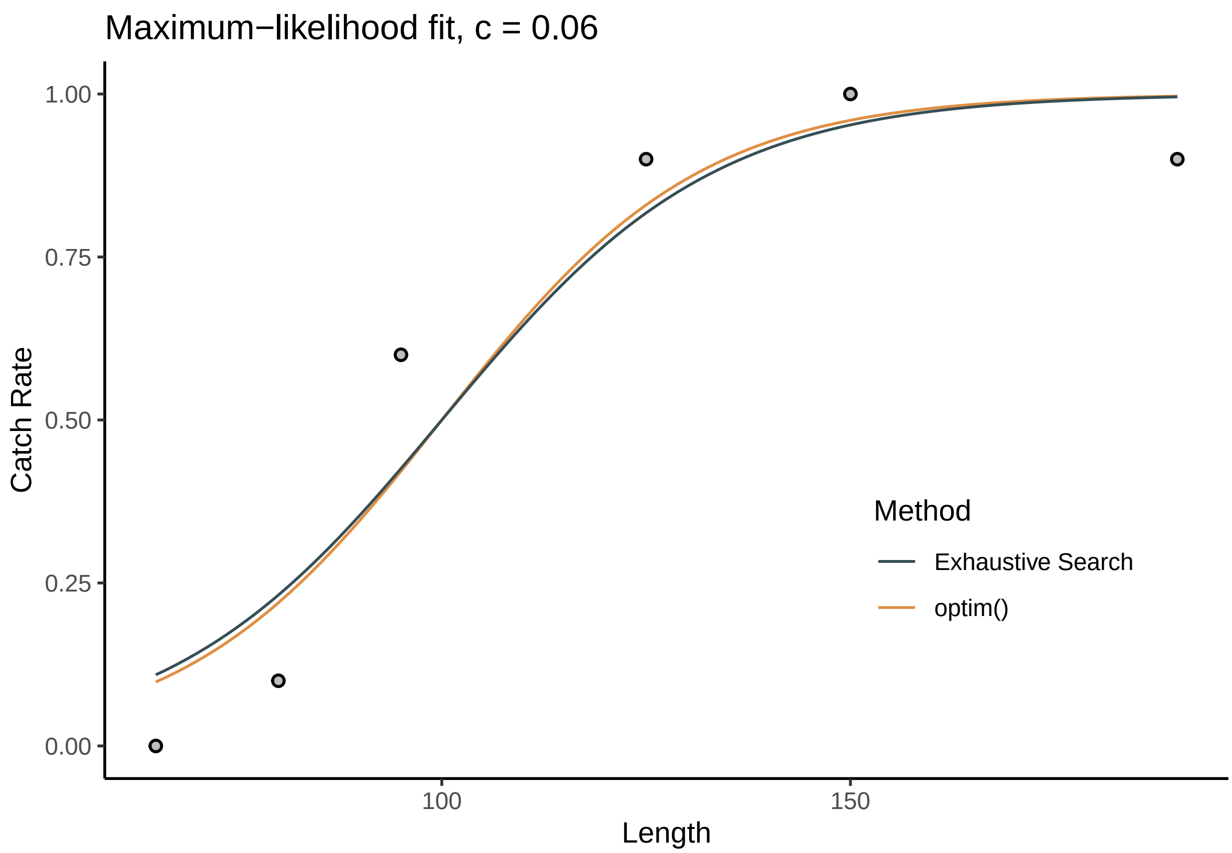



**Fig. 5:** As with sum-of-squares, the exhaustive search and optimized parameters provide similar selectivity curves. Additionally, these curves are similar to those produced using the sum-of-squares method.

# 3 2D fitting

## 3.1 Task 7

```
#selectivity function
f_selectivity_2d <- function(l,c,l_star=100){
  exp(c*(l-l_star))/(1 + exp(c*(l-l_star)))
}
```

```r
LL_f_selectivity_2d <- function(c, l_star, length_classes, catch_at_length, escape_at_length){

  LL_out <- 0

  for (i_length_class in 1:length(length_classes)){

    length_class_i <- length_classes[i_length_class]
    LL_i <- catch_at_length[i_length_class]*log(f_selectivity_2d(length_class_i,c, l_star))+
      escape_at_length[i_length_class]*log(1-f_selectivity_2d(length_class_i,c,l_star))

    LL_out <- LL_out + LL_i

  } #for length class

  return(LL_out)

}

f_LL_scallop_2d <- function(x, l_star) LL_f_selectivity_2d(x, l_star,
                                                          scallops.df$length_bin,
                                                          scallops.df$catch_at_length,
                                                          scallops.df$escapes_at_length)

plot.df <- expand_grid(c=seq(0,0.25,0.001),
                       l_star=seq(70,150,0.5)) %>%
  mutate(LL = f_LL_scallop_2d(c,l_star))

LL_exh.df <- plot.df[which.min(-plot.df$LL),]

ggplot(plot.df, aes(x=c,y=l_star))+
  theme+
  geom_raster(aes(fill=-LL))+
  geom_contour(aes(z=-LL), binwidth=2.5, color="black")+
  geom_point(data=LL_exh.df, fill="darkred", size=3)+
  scale_fill_viridis_c(option="mako", direction=-1)+
  scale_x_continuous(expand=c(0,0))+
  scale_y_continuous(expand=c(0,0))+
  theme(legend.position="right")
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

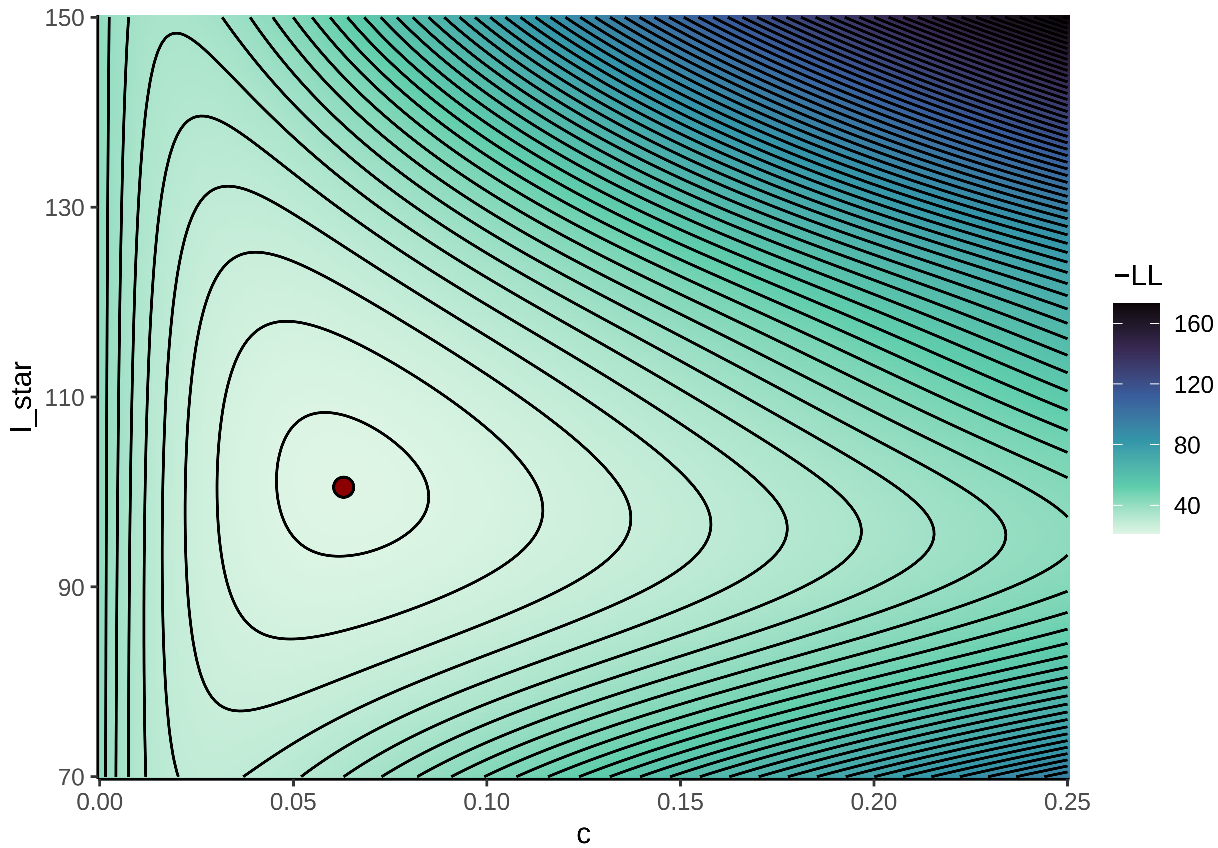
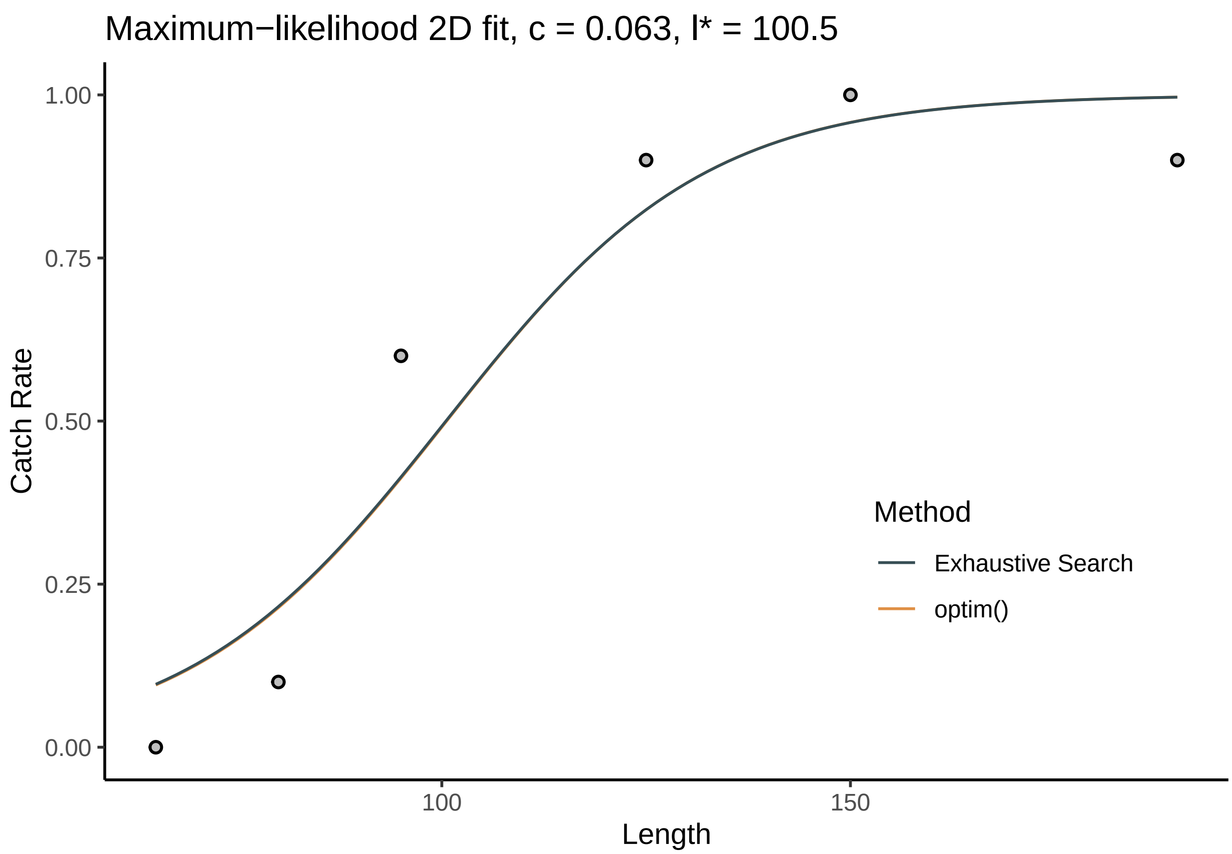



**Fig. 6:** Likelihood under a range of $c$ and $l\_star$, with maximum likihood of 21.7 at $c = 0.063$ and $l\_star = 100.5$ indicated by the red point.

```
LL_exh.df <- plot.df[which.min(-plot.df$LL),]

param_r_opt <- optim(c(LL_exh.df$c, LL_exh.df$l_star),
                     fn = function(params) -f_LL_scallop_2d(params[1],params[2]),
                     method="Nelder-Mead")

ggplot(scallops.df, aes(length_bin, catch_at_length/10))+
  theme+
  geom_point()+
  geom_function(fun = ~f_selectivity_2d(.x, param_r_opt$par[1], param_r_opt$par[2]), aes(color="optim()
  geom_function(fun = ~f_selectivity_2d(.x, LL_exh.df$c, LL_exh.df$l_star), aes(color="Exhaustive Searc
  #geom_abline()
  labs(x="Length", y="Catch Rate",
       title=paste0("Maximum-likelihood 2D fit, c = ",LL_exh.df$c,", l* = ",LL_exh.df$l_star), color="Me
```




> **Fig. 7:** Selectivity curve for *c* and *l_star* identified with exhaustive search (black) and `optim()` (orange) using the 2d method. Note that the two lines are plotting nearly on top of each other.

## 4 Reflection Questions:

The key difference is that Yochum and DuPaul (2008) use a survey dredge to determine the size distribution of the population, rather than assuming the size distribution as we have done here.

Earlier in this lab, we assumed that l_star was the size of the mesh (100mm) and it proved to be pretty accurate. The size of the rings in the commericial gear in Yochum and DuPaul (2008) was 102mm, and so I will use that as a prediction that here.

## 5 Extension

```
#import data
df <- read_csv(here("Labs/LAB 4/data-2024/whelk_catch_at_length_paired.csv")) %>%
  mutate(is_commercial=ifelse(Gear=="C", TRUE, FALSE))
```

```
## Rows: 25 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Gear
```

```
## dbl (2): Tow_ID, Length_mm
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#likelihood function for paired measurements
LL_f_splitprop_2d <- function(c, l_star, catch_lengths, is_commercial){

  p_C <- 0.5
  LL_out <- 0

  for (i_individual in 1:length(catch_lengths)){

    length_i <- catch_lengths[i_individual]
    is_commercial_i <- is_commercial[i_individual]

    phi_C_i <- p_C*f_selectivity_2d(length_i,c,l_star) /
      (p_C*f_selectivity_2d(length_i, c, l_star)+(1 - p_C))

    LL_i <- phi_C_i * is_commercial_i + (1 - phi_C_i)*(1-is_commercial_i)

    LL_out <- LL_out + LL_i

  } #for length class

  return(LL_out)

}

#stick it in another function
f_LL_whelk_paired_2d <- function(c, l_star) {
  LL_f_splitprop_2d(c, l_star,
                    df$Length_mm,
                    is_commercial=df$is_commercial)
}

#estimate LL for a range of c and l_star
plot.df <- expand_grid(c=seq(0,100,0.1),
                       l_star=seq(10,26,0.1)) %>%
  mutate(LL = f_LL_whelk_paired_2d(c,l_star)) %>%
  drop_na()

#exhaustive search
LL_exh.df <- plot.df[which.min(-plot.df$LL),]

#optim()
param_r_opt <- optim(c(LL_exh.df$c, LL_exh.df$l_star),
                     fn = function(params) -f_LL_whelk_paired_2d(params[1],params[2]),
                     method="Nelder-Mead")

#make a plot
ggplot(plot.df, aes(x=c,y=l_star))+
  theme+
  geom_raster(aes(fill=-LL))+
```

```
  geom_contour(aes(z=-LL), color="black")+
  geom_point(data=LL_exh.df, fill="darkred", size=3)+
  scale_fill_viridis_c(option="mako", direction=-1)+
  scale_x_continuous(expand=c(0,0))+
  scale_y_continuous(expand=c(0,0))+
  theme(legend.position="right")
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

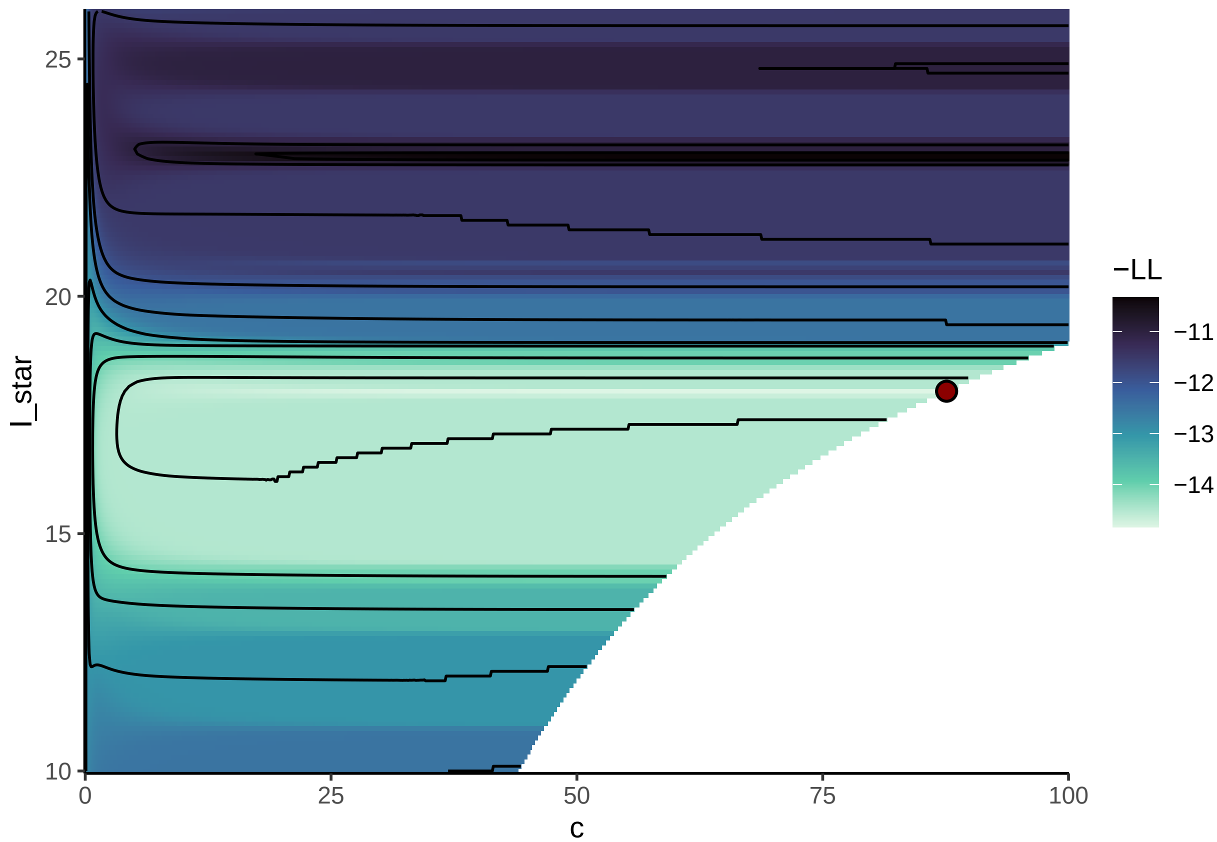



**Fig. 8:** This doesn't look good! The maximum likelihood is on a horizon, suggesting that the model is not performing well.

```
#make a 3d model with p_C
LL_f_splitprop_3d <- function(c, l_star, p_C, catch_lengths, is_commercial){

  LL_out <- 0

  for (i_individual in 1:length(catch_lengths)){

    length_i <- catch_lengths[i_individual]
    is_commercial_i <- is_commercial[i_individual]

    phi_C_i <- p_C*f_selectivity_2d(length_i,c,l_star) /
```

```r
      (p_C*f_selectivity_2d(length_i, c, l_star)+(1 - p_C))

    LL_i <- phi_C_i * is_commercial_i + (1 - phi_C_i)*(1-is_commercial_i)

    LL_out <- LL_out + LL_i

  } #for length class

  return(LL_out)

}

#wrap it in a function
f_LL_whelk_paired_3d <- function(c, l_star, p_C) {
  LL_f_splitprop_3d(c, l_star, p_C,
                    df$Length_mm,
                    is_commercial=df$is_commercial)
}

#try to optimize
param_r_opt <- optim(c(10,18,0.5),
                     fn = function(params) -f_LL_whelk_paired_3d(params[1],params[2],params[3]),
                     method="L-BFGS-B", lower=c(0,10,0), upper=c(100,26,1))

param_r_opt
```

```
## $par
## [1] 10.0314867 18.5539630  0.9977384
##
## $value
## [1] -17.22975
##
## $counts
## function gradient
##       43       43
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

Unfortunately, we are getting unrealistic estimates of $p\_C$. Something is seriously wrong here. . .