LAB 4 Likelihood Tools and Tool Likelihood

ER Deyle

Fall 2024; Marine Semester Block 3

Introduction

Fisheries Context

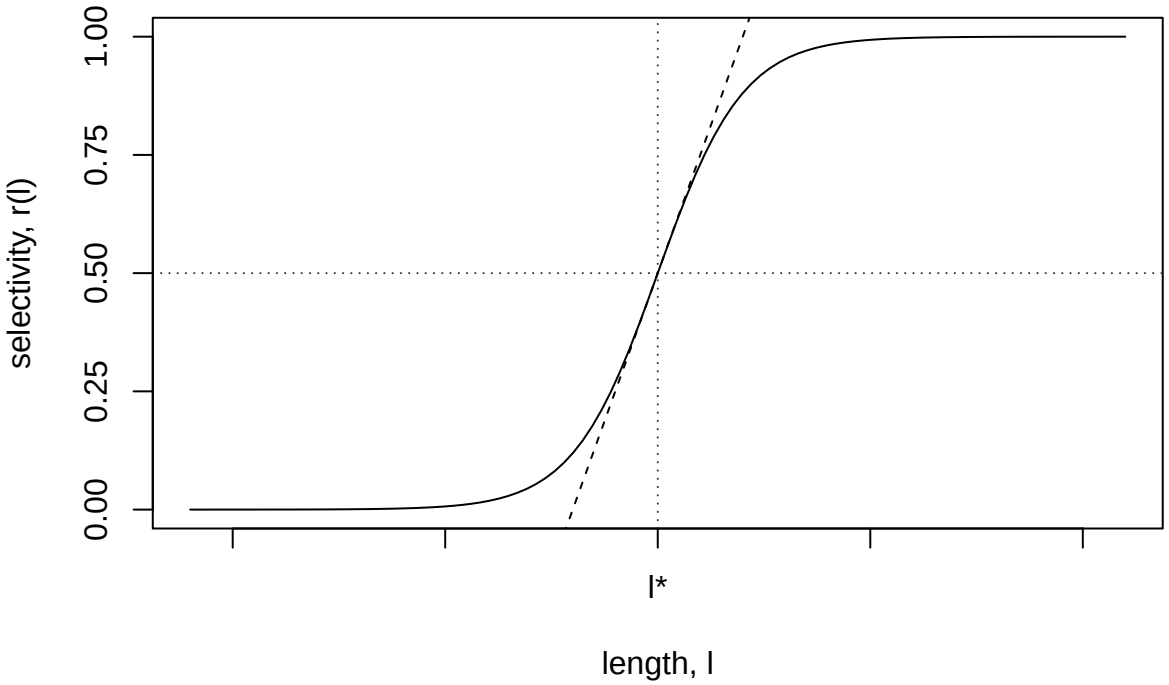
The fishery focus for today is gear selectivity. Gear selectivity is one of several pieces of machinery for translating catch data into age-structured estimates of the population abundance. Gear selectivity can be influenced by many factors from the physical construction of the gear to the physiology of the target organisms. Today, we are going to focus on scallop dredges, which more or less consist of a rake and a giant mesh bag behind it. Scallops smaller than the mesh size tend to fall out, while scallops of large size tend to be retained. For this reason, the selectivity has been modeled by a logistic function (S-curve). Yochum and DuPaul (2008) write this as:

$$r(l) = \frac{exp(a + bl)}{1 + exp(a + bl)}.$$

Notice that as $l \to -\infty$, $r(l) \to 0$ and as $l \to +\infty$, $r(l) \to 1$. That is to say, the selectivity tends to 0 (0% retention) as the length goes to 0 and selectivity tends to 1.0 (100% retention) as the length get very large. Moreover, if we rewrite the equation, the parameters become easier to interpret.

$$r(l|c, l^*) = \frac{exp(c(l - l^*))}{1 + exp(c(l - l^*))}.$$

When $l = l^*$, then $r(l) = 1/2$, making l^* the mid-point of retention. The variable c controls the steepness of the function on either side of the mid-point l^* (although the actual slope of the tangent line has a factor of $1/4$ hiding in it).



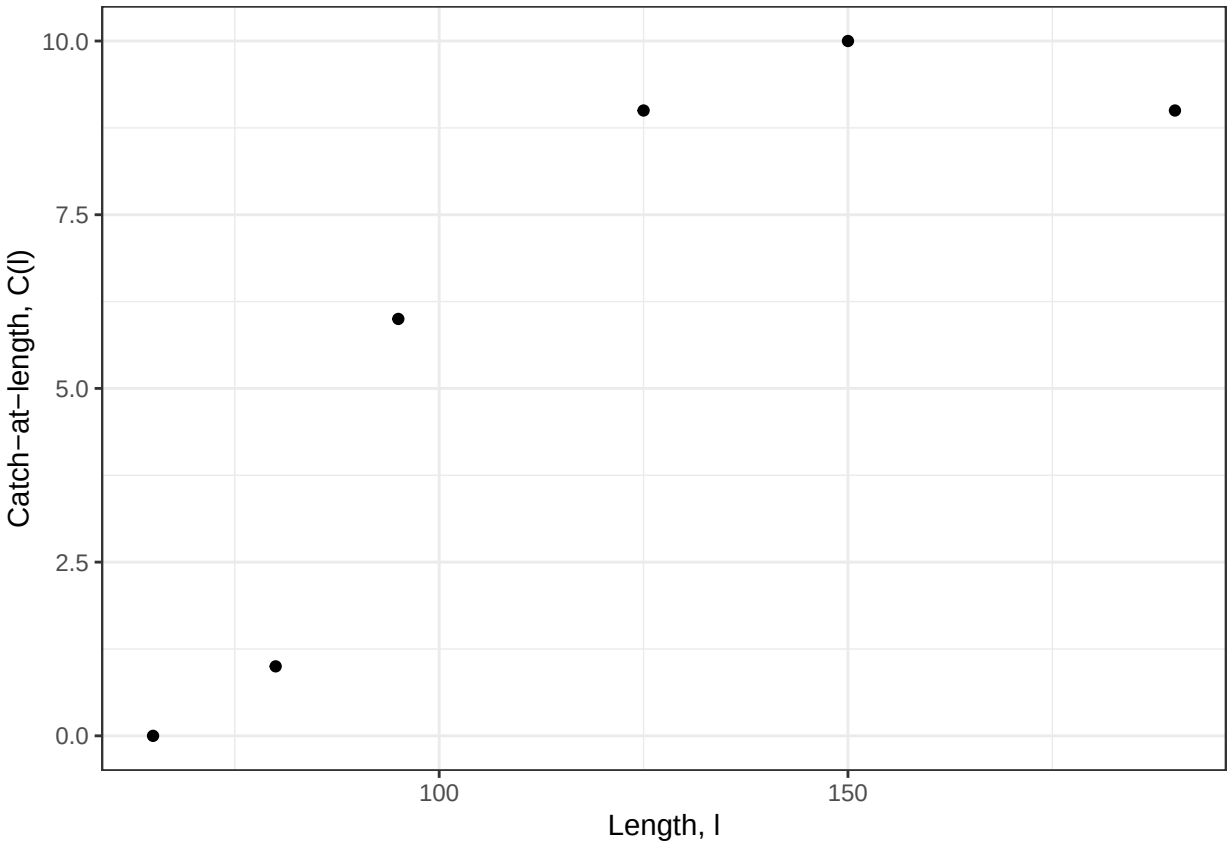
In general practice, both the steepness and mid-point would be fit, but in this lab we'll start by assuming the mid-point is the size of the mesh, $100mm$. Thus initially we just need to do a 1-dimensional optimization to fit the parameter c.

$$r_{100mm}(l|c) = \frac{exp(c(l-100mm))}{1+exp(c(l-100mm))}.$$

It's also a bit tricky relating the selectivity curve $r(l)$ to data. The data we'll be considering is catch-at-length from the gear. [Here's some sample data. We'll collect our own!]

```
scallop_data <- 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)
)

ggplot(scallop_data, aes(x=length_bin,y=catch_at_length)) +
  geom_point() +
  theme_bw() +
  labs(x='Length, l',y='Catch-at-length, C(l)')
```



That is because we only observed the individuals that are captured! If you catch no small individuals and many large individuals, does that mean your gear has strong selectivity or the natural population has a very disequal age distribution (differences in *selectivity* versus differences in *availability*)? For this lab, we're going to assume we have a very good idea of the length distribution of the population we're sampling.

We will start dealing with this in a more simple but less correct way, then attempt to be more rigorous. So for the first phase, we are going to assume that the population has a completely uniform size distribution. Thus, each length-class is equally *available* to catch. Then, the selectivity curve $r(l)$ should directly predict the catch-at-length data for a length-class, l_i, as the selectivity of that length-class times the total catch, N_C, divided by the number of length classes, n_i.

$$C(l_i) = r(l_i)N_C/n_i.$$

Therefore, we can fit parameters of the selectivity model to the catch fractions $n_i C(l_i)/N_C$. If the population does not have a uniform size-distribution, we can modify this so that the data we will fit are the fraction of a length-class in the catch compared to the fraction of the length-class in the population being sampled. However, this has the downside of weighting all length classes equally, even if the number of observations is very small. Properly dealing with a non-uniform size-distribution will require more machinery.

Tangent: In the field, the size-distribution of the natural population is generally something we want to get OUT of analysis, rather than something in hand at the start! In the Yochum & DuPaul (2008) paper, the data are from a paired trawl, where the catch of the commercial gear being studied is compared to catch of gear designed for minimal size selectivity (effectively acting as a control sample). Thus the data to fit are the fraction of catch in the commercial gear to the total catch for a given length-class. This adds an additional layer of modeling.

Computational Approach

Our first foray into model fitting will be with least-squares regression. The principle of least squares regression is to find the set of model parameters that produce the smallest residuals (discrepancies between the model prediction and the observed data), where the "size" of the residuals is measured by Euclidean distance (sum of squares). If the model predictions of the response variable (the selectivity) are denoted $\hat{s}_{(c,l*)}$ and the observations of the response variable are denoted s, then our goal will be to find the parameters c and l^* minimize the sum of squared differences:

$$SS(c, l^*) = \sum_i (\hat{s}_{(c,l^*)} - s)^2$$

We will then turn our attention to "Maximum Likelihood Estimation" (MLE). The principle is to find the set of model parameters that maximize how "likely" (i.e. the probability) the observed data is under that model. These approaches have a lot in common; the key difference in MLE is designed for random effects.

In this lab, the model gives the fraction of individuals of a length-class retained relative to the total individuals of that length-class encountered.

The probability that an individual of length l is retained after it is encountered by the gear is

$$r_C(l|c, l^*) = \frac{exp(c(l - l^*))}{1 + exp(c(l - l^*))}.$$

And since the only possibilities we are entertaining is that an individual is either captured or not captured, the probability an encountered individual is not captured is just $1 - r_C(l)$. The likelihood of getting a set of data D_l consisting of n_c captures of N total individuals, all of the same length l is:

$$\mathcal{L}(\mathbf{D}_l|c, l^*) = (r_C(l))^{n_c}(1 - r_C(l))^{N-n_c}.$$

Of course, we are not interested in the likelihood for just a single length class, but across several different length classes. Recall that probabilities multiply:

$$p(A \cup B) = p(A)p(B).$$

So to get the likelihood of data across multiple length classes would involve multiplying them all together. This is doable. On the other hand, R is even better at adding a lot of things together than multiplying a lot of things together. For this reason, data fitting will often be done in the context of "Log-likelihood" instead. Recall that taking the $log()$ turns multiplications into additions

$$\log(AB) = \log(A) + \log(B),$$

and turns exponentiations into multipications

$$\log(A^b) = b \log(A).$$

Finally, the reason this is all useful to do is that the logrithm is minimum-preserving. Since it is "monotonic", if $a < b$ then

$$\log(a) < \log(b).$$

So when we put it all together, the log-likelihood, $\mathcal{LL}$ of our data D given a particular set of parameters c and l^* will be

$$\mathcal{LL}(\mathbf{D}|c,l^*) = \sum_{i=1}^{n} \left[n_c(l_i) \log(r_C(l_i)) + (N(l_i) - n_c(l_i)) \log(1 - r_C(l_i)) \right].$$

Where $n_C(l_i)$ is the number of individuals caught of length l_i (in the "ith" length-class) and $N(l_i)$ is the total individuals encountered of that length.

Task list:

- Take a least-squares approach to fitting the selectivity curve.
- Create a likelihood function for the selectivity model in 1D.
- Use R code to calculate the 1D-MLE of the selectivity steepness parameter assuming a value of the midpoint.
- Use R code to calculate the 2D-MLE of the selectivity steepness parameter and value of the midpoint.

1D parameter fitting

We are going to begin by fixing the mid-point of selectivity (l^* or l_{50}) to the physical measurement of the gear, $l^* = 100mm$. [If you collected your own data, measure the mesh size of your "scallop dredge" to get a value for l^*.]

(Nonlinear) Least Squares

TASK 1: First, write a function to represent the selectivity function $r(l|c)$ we are trying to fit. Then incorporate that into a second function for the sum-of-squares $SS(c,l^*)$ given observations of `length_class` and `catch_rate`. The selectivity function is a function of length (`l`), but we are also going to vary parameters. Thus your R function for selectivity needs to also be a function of the parameter `c`.

```
f_selectivity_1d <- function(l,c_){
  capture_rate <- exp(c_*(l - 100))/( 1 + exp(c_*(l - 100)) )
  return(capture_rate)
}
```

The model we are trying to fit is a function of l, the length. However, the thing we are trying to optimize is not a function of l, but a function of the model parameter, c.

```
SS_f_selectivity_1d <- function(c_,length_classes,catch_rates){
  squared_errors <- (f_selectivity_1d(length_classes,c_) - catch_rates)^2
  output <- sum(squared_errors)
  return(output)
}
```

TASK 2: First, plot the sum-of-squares (SS) as a function of the model parameter (c). Then use an exhaustive search over a range of values of `c` to estimate a minimum of the sum-of-squares function, and plot the selectivity curve for that value of `c`.

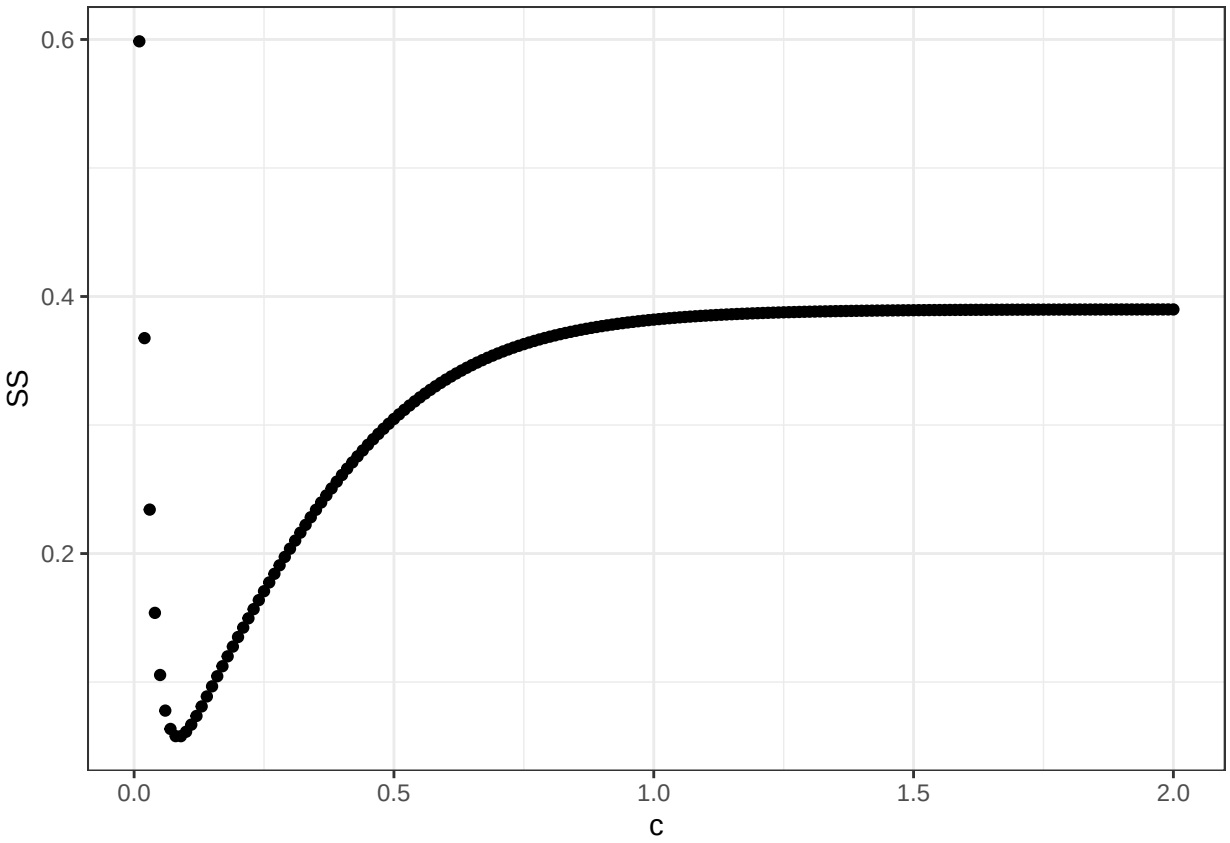
```
v_c_values <- seq(0.01,2,by=0.01)
v_SS_vs_c <- vector()
```

```
catch_rates <- scallop_data$catch_at_length / 10

for(c_i in v_c_values){
 v_SS_vs_c <- c(v_SS_vs_c,SS_f_selectivity_1d(c_i,scallop_data$length_bin,catch_rates))
  }
```

```
df_plot_SS <- data.frame(c=v_c_values,SS=v_SS_vs_c)

ggplot(df_plot_SS,aes(x=c,y=SS)) +
  geom_point() + theme_bw()
```

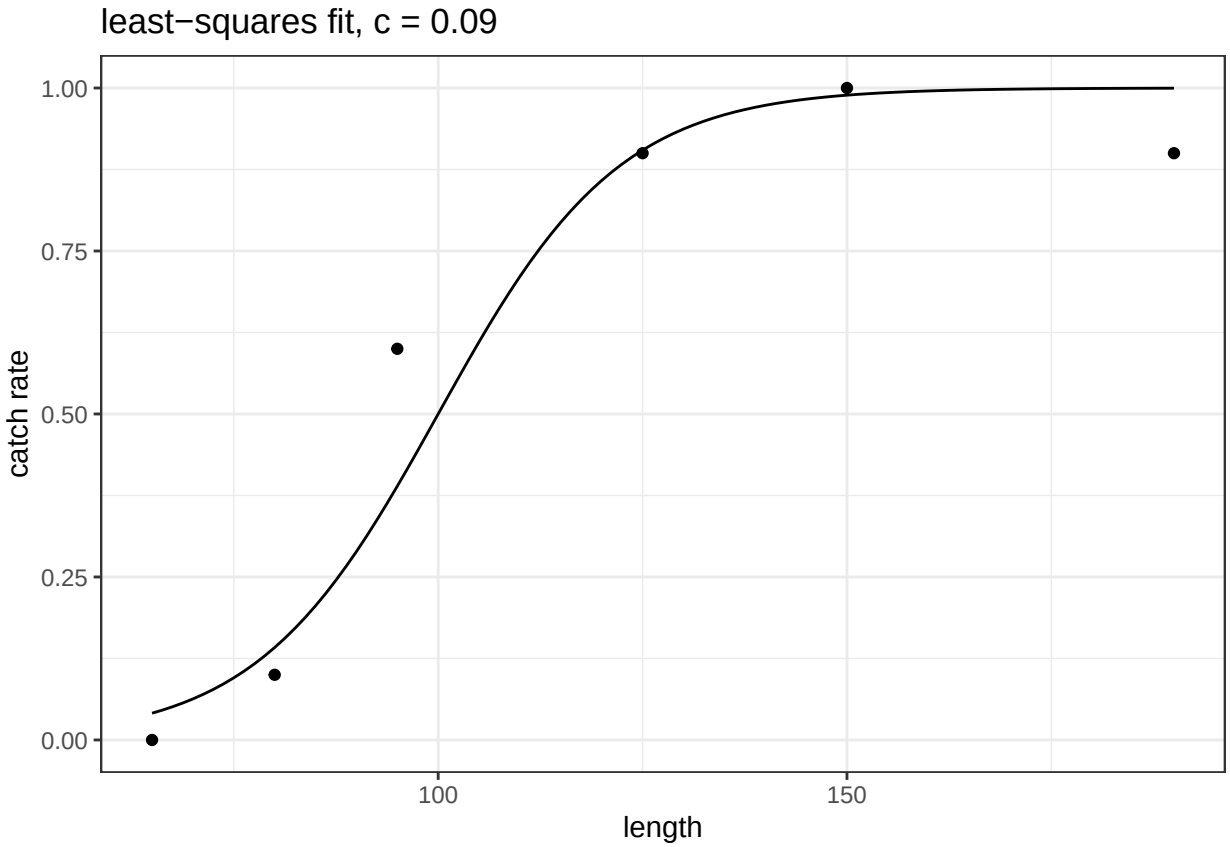


What value of c minimizes the sum-of squares?

```
c_exhaustive_search <- v_c_values[which.min(v_SS_vs_c)]
```

Plot the selectivity curve that minimizes the sum-of-squares, along with the data, to visualize the fit.

```
ggplot(scallop_data,aes(x=length_bin,y=catch_at_length/10)) +
  geom_point() +
  geom_function(fun = ~ f_selectivity_1d(.x,c_exhaustive_search)) +
  theme_bw() +
  labs(x="length",y="catch rate",title=paste("least-squares fit, c =",c_exhaustive_search))
```

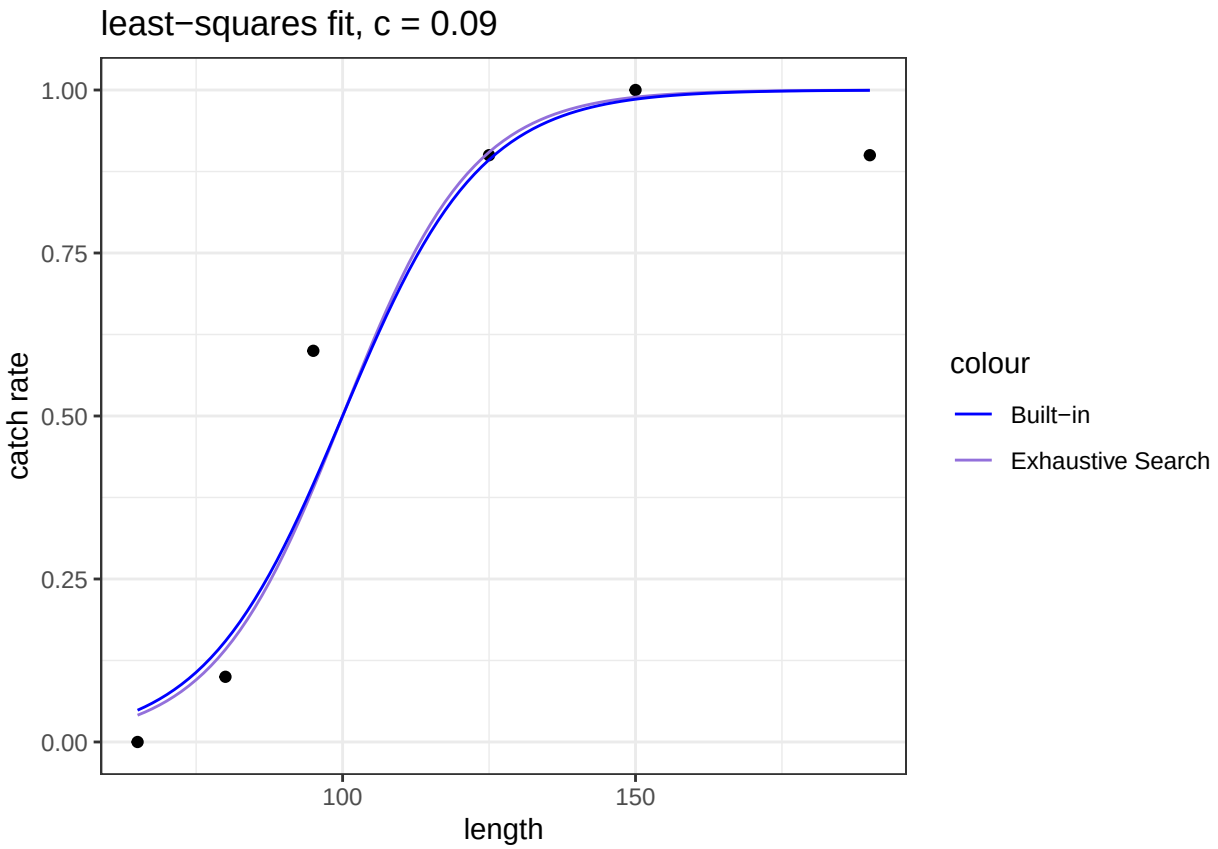


TASK 3: Use the built-in R function `optim` to explicitly optimize the sum of squares to get an estimate for the best value of `c`. Now you have two estimates of `c`: one from using exhaustive search, and one from R's `optim` function. Plot the selectivity curves that are "optimal" under these two different methods together on the same plot, along with the data. You need to tell it an initial guess for the variable you are minimizing over (`c`).

```
c_r_optim <- optim(0.1,
                   fn = function(c_) SS_f_selectivity_1d(c_,scallop_data$length_bin,catch_rates),
                   method="BFGS")$par
c_exhaustive_search <- v_c_values[which.min(v_SS_vs_c)]
```

Use the example of `geom_function()` below if you need a suggestion for plotting.

```
colors <- c('Exhaustive Search' = 'mediumpurple','Built-in'='blue1')
ggplot(data=scallop_data,aes(x = length_bin, y = catch_at_length/10)) +
  geom_point() +
  geom_function(fun = ~ f_selectivity_1d(.x, c_exhaustive_search),aes(color='Exhaustive Search')) +
  geom_function(fun = ~ f_selectivity_1d(.x, c_r_optim),aes(color='Built-in')) +
  theme_bw() +
  labs(x="length",y="catch rate",title=paste("least-squares fit, c =",c_exhaustive_search)) +
  scale_color_manual(values = colors)
```



Likelihood function in 1D

We now turn to the maximum likelihood approach of parameter fitting. First, review the explanation in the introduction. This gives a symbolic expression for the likelihood of the logistic selectivity.

```
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

  }

  return(LL_out)
}
```

TASK 4: Make a function in R to represent the log-likelihood of an arbitrary set of data, given parameter choice. This should have the same or similar inputs to the sum-of-squares error before.

Testing the Likelihood function.

Let's play around with the likelihood function a bit to make sure it works and to build a bit of intuition.

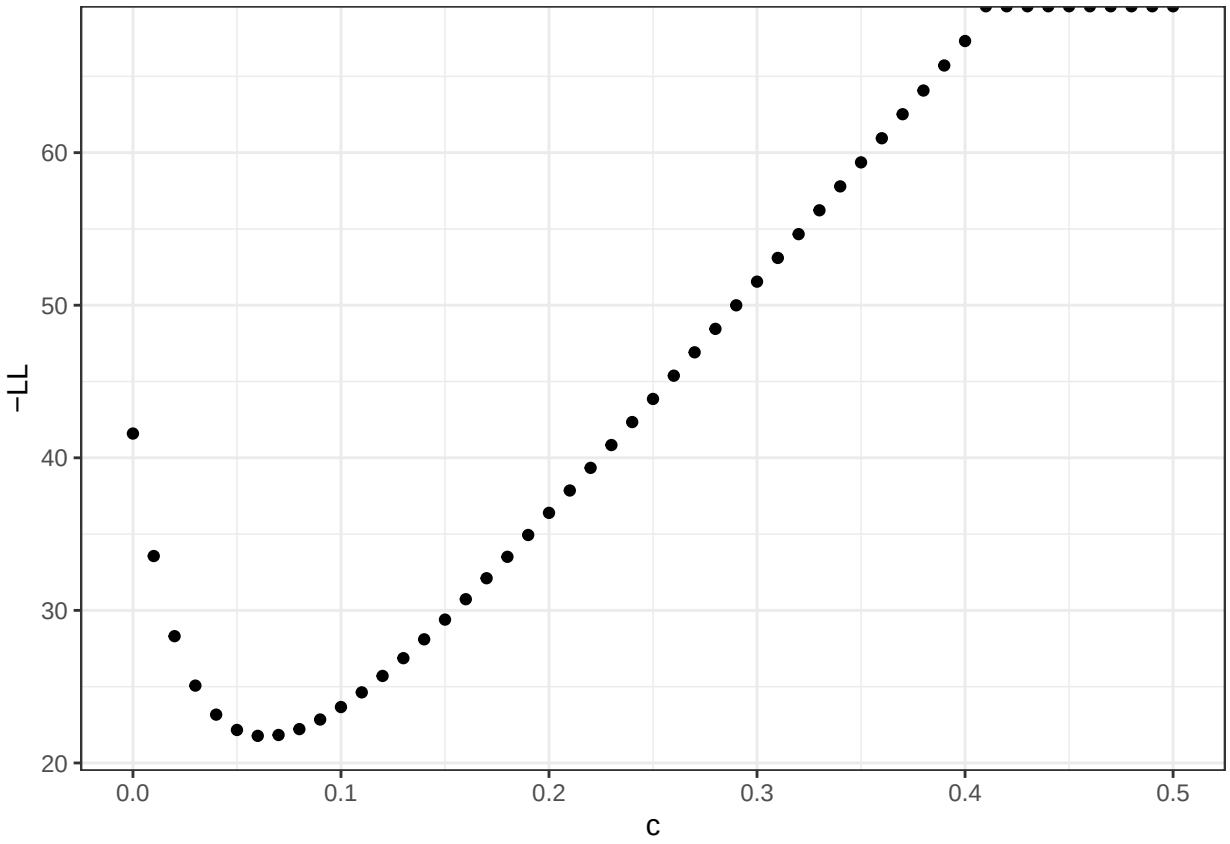
TASK 5: Plot the likelihood function for the same data used for the sum-of-squares exercise. The plot should have c on the x-axis (try a range of values from 0 to 0.5), and $\mathcal{L}(\mathbf{D}_l|c)$ on the y-axis. In theory we can do the above with the function `LL_f_selectivity_1d()` above, but it can be a little easier if you first define an intermediate function that allows you to vary the parameter, but keep the data arguments fixed.

```
f_LL_scallop_data <- function(x) LL_f_selectivity_1d(x,
                                                     scallop_data$length_bin,
                                                     scallop_data$catch_at_length,
                                                     scallop_data$escapes_at_length)
```

Then you can just calculate that function (the log-likelihood *given* the particular set of data, `scallop_data`) for a sequence of `c_` values. (And plot!).

```
v_c_values <- seq(0,0.5,by=0.01)
df_plot_LL <- data.frame(c=v_c_values,LL=f_LL_scallop_data(v_c_values))


ggplot(df_plot_LL,aes(x=c,y=-LL)) +
  geom_point() + theme_bw()
```

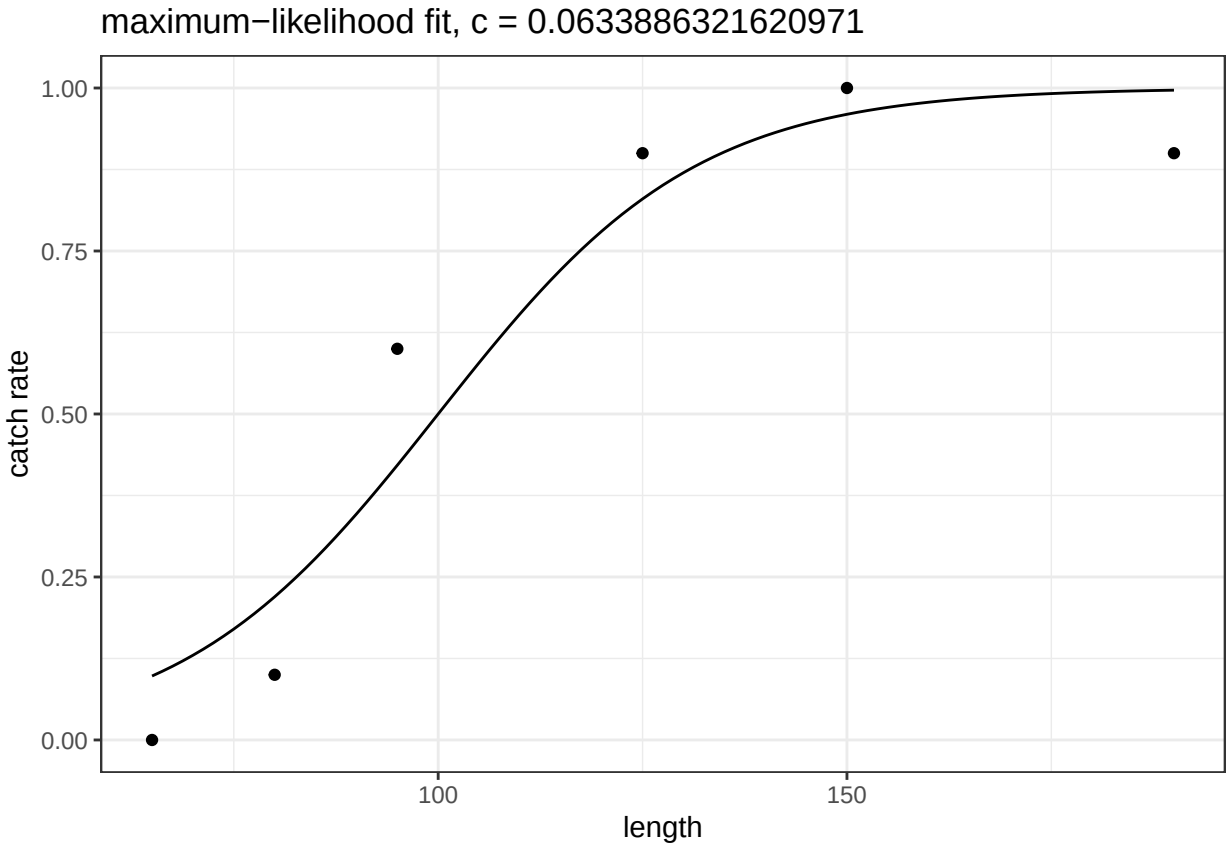


Fitting by optimizing the likelihood function

```
c_r_optim <- optim(0.1, fn = function(c_) -f_LL_scallop_data(c_),method="BFGS")$par
```

```
ggplot(scallop_data,aes(x=length_bin,y=catch_at_length/10)) +
  geom_point() +
  geom_function(fun = ~ f_selectivity_1d(.x,c_r_optim)) +
  theme_bw() +
  labs(x="length",y="catch rate",title=paste("maximum-likelihood fit, c =",c_r_optim))
```

TASK 6: Use exhaustive search and the R `optim()` function to get two values for the optimal parameter `c` of the logistic selectivity given the data `scallop_data`. Then plot the selectivity curves using these values of `c`, as you did before for least-squares.



This is a good time to note the difference between the model function and the likelihood function. Each point on the x-axis of the likelihood function is a different parameterization of the model function. Towards the left end are very shallow slopped selectivity models and towards the right end are very steep selectivity models.

More Maximum Likelihood Estimation

Likelihood in 2D

We need to rework the machinery we've developed to allow for fitting in the length mid-point as well as the slope of the "S".

TASK 7: Rewrite the log-likelihood function to include specification of both parameters `c` and `l_star`.

> This will also necessitate writing a new function for the selectivity curve that has `l_star` as a parameter, not fixed value.

```
f_selectivity_2d <- function(l,c_,l_star){
  capture_rate <- exp(c_*(l - l_star))/( 1 + exp(c_*(l - l_star)) )
  return(capture_rate)
}
```

```
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

  }

  return(LL_out)
}
```

Before, we could plot the 1D (log-)likelihood function as a curve in the x-y plane. Now, the likelihood function effectively generates a *surface* rather than just a *curve*. This makes visual inspection a bit more tricky and it makes exhaustive search a lot more work for the computer.

First, as a matter of good house-keeping, we define an intermediate function that supplies the log-likelihood with the general 2D LL function `LL_f_selectivity_2d()`, but sets the data inputs to be fixed. That is, a function of the LL specific to just our particular data, `scallop_data`.

```
f_LL_scallop_data <- function(c_,l_star) LL_f_selectivity_2d(c_,l_star,
                                                             scallop_data$length_bin,
                                                             scallop_data$catch_at_length,
                                                             scallop_data$escapes_at_length)
```

Now we need to do an exhaustive search over a grid. There's an R function that makes generating the gridded values easy (although we could do this with a `for` loop no problem!). It's called `expand.grid()`. Type `?expand.grid`. For example, if we want to evaluate a function of x and y on a 1 x 1 grid in the x-y plane, we'd type

```
search_grid <- expand.grid(x=0:10,y=0:10)
head(search_grid)
```

```
##   x y
## 1 0 0
## 2 1 0
## 3 2 0
## 4 3 0
## 5 4 0
## 6 5 0
```

Notice R gives us our pairs of (x,y) points in a data-frame. Convenient! We want to search a grid of `c` and `l_star` values, however.

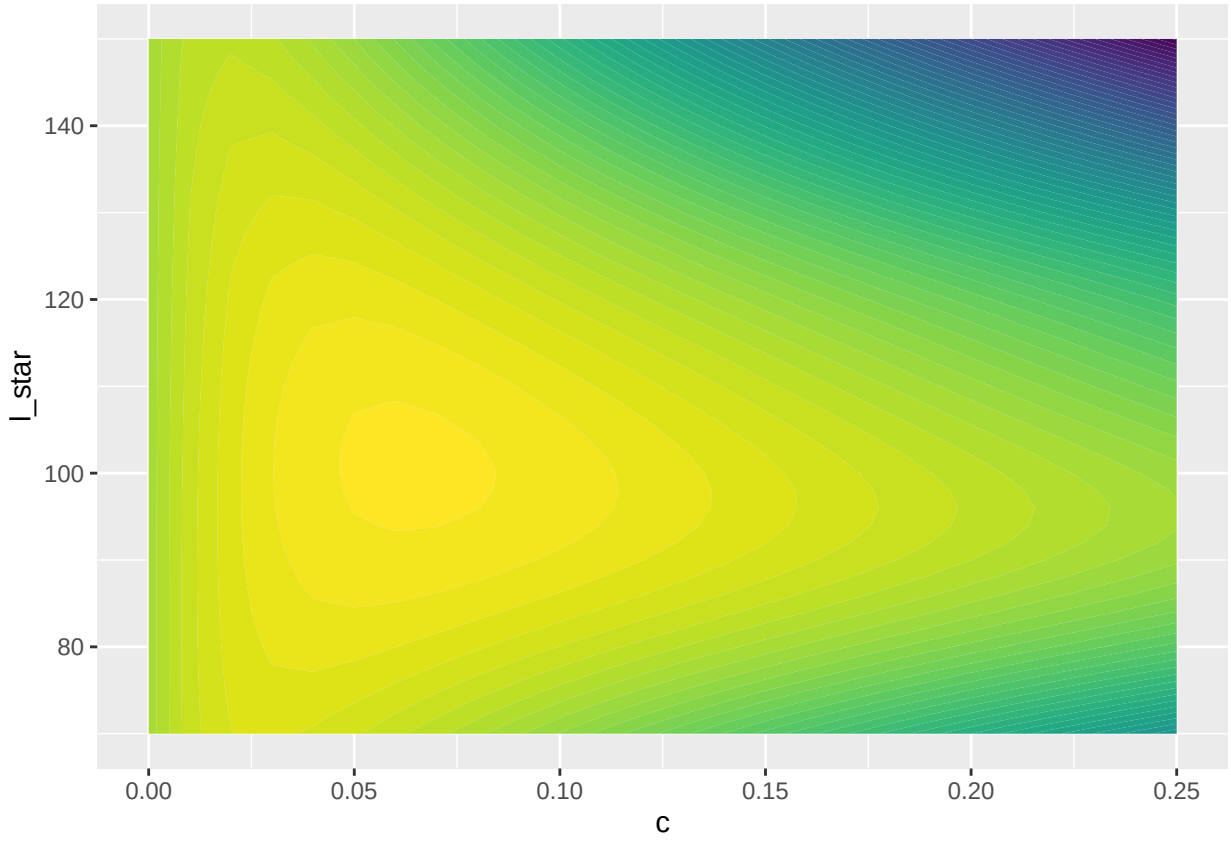
TASK 8: Use expand.grid and the LL function in 2D to calculate the likelihood of `scallop_data` under parameter combinations with `c` ranging from 0 - 0.25 and `l_star` ranging from 70 - 150.

What pair of `c` and `l_star` values resulted in the maximum LL?

```r
v_c_values <- seq(0,0.25,by=0.01)
v_l_values <- seq(70,150,by=2)
df_plot_LL <- expand.grid(c=v_c_values,l_star=v_l_values)
df_plot_LL$LL <- NA

for(i_row in 1:nrow(df_plot_LL)){
df_plot_LL$LL[i_row] <- f_LL_scallop_data(df_plot_LL$c[i_row],df_plot_LL$l_star[i_row])
}

ggplot(df_plot_LL,aes(x=c,y=l_star,z=LL)) +
  geom_contour_filled(binwidth = 2.5,show.legend=FALSE)
```




```r
# geom_raster(data=df, aes(fill=z)) +
#      scale_fill_gradient(limits=range(df$z), high = 'white', low = 'red')
```

```r
params_LL_exhaustive <- df_plot_LL[which.min(-df_plot_LL$LL),c("c","l_star")]
```

We can also use the R `optim` function in a 2-dimensional optimization setting. Just always keep in mind that R `optim` actually does **minimization** by default.

```r
param_r_optim <- optim(c(0.1,100),
                       fn = function(params) -f_LL_scallop_data(params[1],params[2]),
                       method="Nelder-Mead")
```

TASK 9: Plot the selectivity curve for the pairs of `c` and `l_star` found with exhaustive search and the R `optim()` function.

Reflection Questions:

> Look again at the analysis in Yochum and DuPaul 2008 (in Quant_Fish_2023/LAB 4/). They use maximum likelihood to fit parameters of the New Bedford scallop dredge. However, their approach has an extra layer. What are these differences?

Yochum and DuPaul use the same equation for dredge selectivity (using `a` and `b` however rather than `c` and `l_star`). However, in a real world (survey) setting we do not actually know the number of individuals that escape capture. An easy way around this is to use a paired approach with the normal gear ("Commercial" gear) deployed along side a dredge that does not have length selectivity ("Survey" gear) but captures all individuals encountered. We could work out an expression for the number of captures and estimated number of escapes from this, but an equivalent thing to do is to think about how the selectivity of the Commercial gear (r_C) will drive the proportion of scallops (of length l) that are caught in the commercial gear out of the total catch from both gears (Fc(l)).

Thus, now instead of trying to fit our captures and escapes to the function `r`, we instead fit the proportions between Commercial and Survey dredges $\Phi_C(l)$. In terms of the function `r` we were working with, we model this proportion as:

$$\Phi_C(l) = \frac{p_C r_C(l)}{p_C r_C(l) + (1 - p_C)}$$

.

We've introduced another parameter here, p_C which is the "split probability". In effect, this parameter can account for the Commercial and Survey dredges encountering different numbers of individuals irrespective of their selectivity. Why? Well the obvious case would be if one has a wider mouth than the other. If the dredges are nearly identical aside from mesh, we'd expect p_C to be 1/2.

We've already decided how we want to model the retention probability $r(l)$ so we can substitute this in to get the full model equation:

$$\Phi_C(l) = \frac{p_C \ exp(c(l - l^*))}{(1 - p_C) + exp(c(l - l^*))}$$

With this equation, we can set about estimating the selectivity of the "Commercial dredge" we have in our desktop scallop dredging experiment. Since you already have written a function for $r_C(l)$ you can use that in your definition of $\Phi_C(l)$ ... or not!

Reflection Question: Before you start, make a prediction for l^*. What's your basis for that prediction?

Now we want to calculate the log-likelihood function for this model. Yochum and DuPaul give it to us but it's something in principal we could work out for ourselves given that we have the expression above.

So when we put it all together, the log-likelihood, $\mathcal{LL}$ of our data D given a particular set of parameters c, l^*, and p_C along with data of the number caught in the Commercial Dredge of a given length class, $n_C(l_i)$, and the number caught in the Survey Dredge of that lnegth, $n_S(l_i)$:

$$\mathcal{LL}(\mathbf{D}|c, l^*, p_C) = \sum_{i=1}^{n} \left[n_C(l_i) \log(\Phi_C(l_i)) + (n_S(l_i)) \log(1 - \Phi_C(l_i)) \right].$$

Where $n_C(l_i)$ is the number of individuals caught of length l_i (in the "ith" length-class) and $N(l_i)$ is the total individuals encountered of that length.