ESM 244 Week 4: NLS

Nathan Grimes

1/17/2022

Quick Aside

I created this presentation in RMarkdown using the Beamer extension. If you want to see the code for making a presentation in markdown, I included it on gauchospace.

Pros

- Easily integrate R code
- Updates figures automatically
- Access to Latex and all mathematical formulations

Cons

- Define everything in code, no easy powerpoint tools
- Maybe not as “sexy” as other presentations

Packages to follow along with

```
library(tidyverse)
remotes::install_github("lter/lterdatasampler")
library(lterdatasampler)
library(knitr)
library(broom)
library(investr)
```

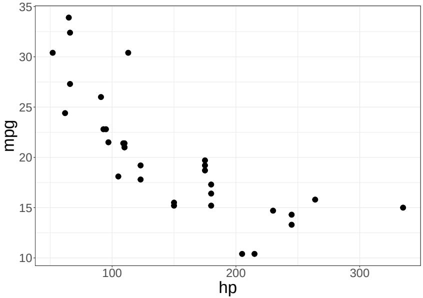
Section 1

What is NLS?

Remember what OLS does first

Fundamental objective of Ordinary Least Squares regression:

- Best fit a line to data

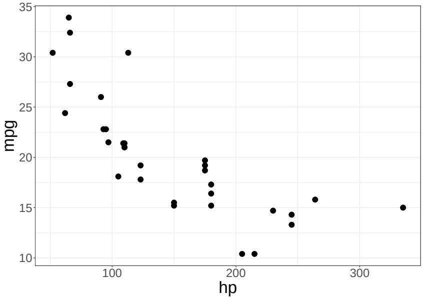


How does OLS fit the line?

Remember what OLS does first

Fundamental objective of Ordinary Least Squares regression:

- Best fit a line to data



How does OLS fit the line?

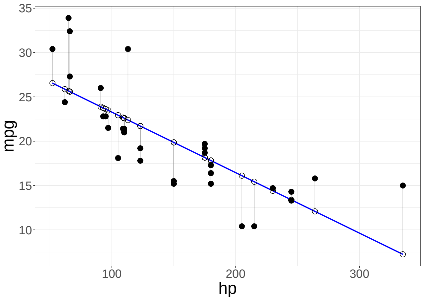
- Hint: $\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$

Remember what OLS does first

How does OLS fit the line?

- Minimize squared error (aka residuals)

$$\hat{\beta} = \min_{\beta} \sum_{i=1}^{n} \hat{\epsilon}_i^{\,2} = \sum_{i=1}^{n} (y_i - \beta x_i)^2$$

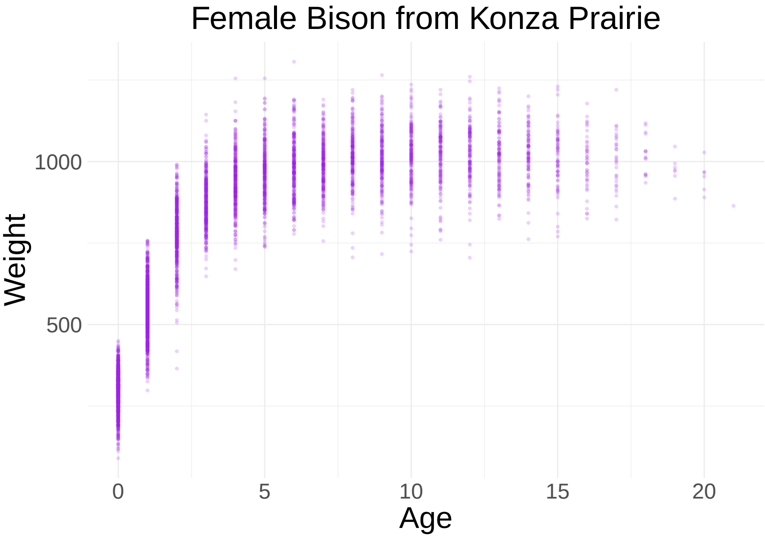


OLS is simple and powerful, but has limitations

- Linear relationship between predictor (y) and variables (x)
- Last week we started branching away from linear models with logistic regressions
- But the link functions typically still maintain a linear form

$$\ln(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...\beta_n x_n$$

But what if we get something like this?



Or this?

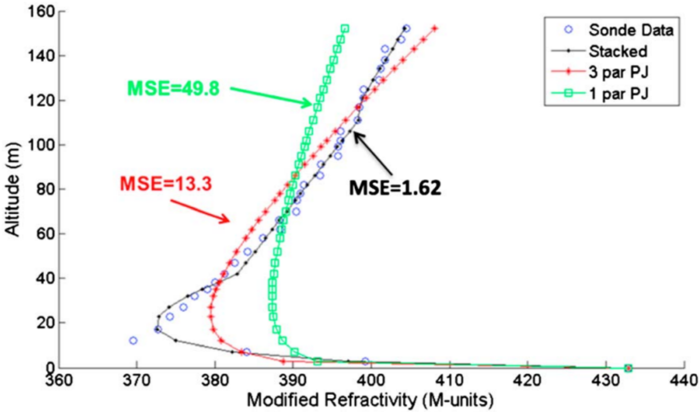


Figure 4. Modified refractivity versus height for an example of GA results for all three models (equations (1)–(3)) along with the corresponding measured refractivity data from the sonde (see legend). Corresponding MSE values (in M units^2) are also annotated in the figure.

In specific applications, accuracy greatly matters!

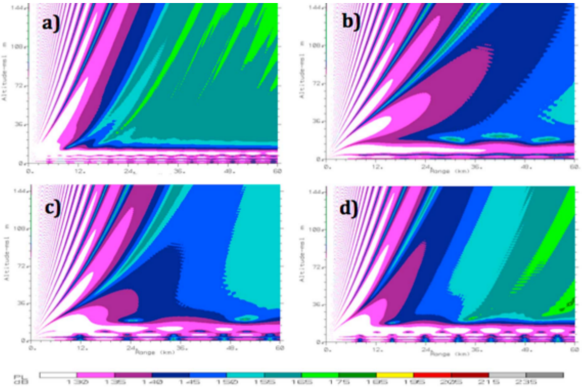


Figure 5. VTRPE simulation results using the *M* profiles shown in Figure 4. (a) Propagation loss (PL) pattern for the sonde data, (b) the 1-parameter Paulus-Jeske model, (c) the 3-parameter Paulus-Jeske model, and (d) the Stacked model.

Nonlinear Least Squares

Apply the same idea of least squares error minimization, but with any function

$$y_i = f(x_i, \beta) + \epsilon_i \tag{1}$$

$$\min_{\beta} = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - f(x_i, \beta))^2 \tag{2}$$

General idea is very similar, but implementation and use is quite different

How NLS works

No simple analytical solution like in OLS (Solve for $\hat{\beta}$)

Instead we iteratively approximate the solution through algorithms

- Gauss-Newton (Most Common)
- Levenberg-Marquardt (More flexible)

In general the algorithms make an approximation of the functions gradient (think derivative), then move along until some convergence criteria is met

$$\left| \frac{\overbrace{S^k}^{\text{Previous squared errror}} - \overbrace{S^{k+1}}^{\text{Updated squared error}}}{S^k} \right| < 0.0001$$

Demonstration of Gauss Newton Algorithm

When viewing the Gifs at the link, see how important the initial position of the guess is to avoid being trapped.

(A weakness in Beamer is not being able to include gifs, it is a pdf afterall!)

Why should we use NLS?

We need far fewer assumptions than even multiple regression

- Residuals do not have to be normally distributed
- No linear relationship required
- Don't care about homoscedasticity

If underlying model is smooth, can find solutions accurately and quickly compared to other methods

When to use NLS

Best suited for specific model parameterization given a collection of data

NLS excels when we have a known equation and want to fit parameters

There is no R^2 value to compare across model specifications, but we can still test model performance using AIC or Cross Fold Validation through RMSE (In lab this week!)

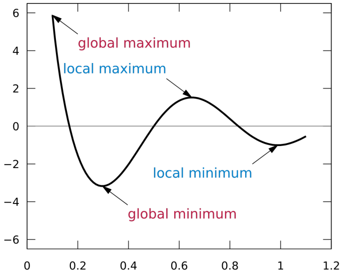
NLS is particularly useful for time series, but we will cover other methods for time series later in class.

Pitfalls (literally) and warnings

NLS is only as good as the underlying model. Bring your brain to the party and make sure the model you're fitting is appropriate

Follows gradient of steepest descent → local min/max valleys

- With n-parameters chances of local min/max rises



Requires good initial guesses

- Comes from the underpinning algorithms

Know when to use NLS or another option

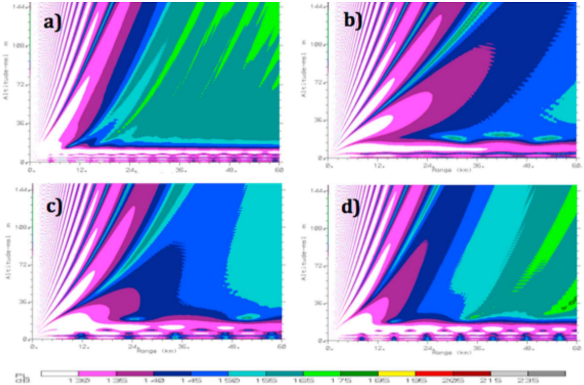


Figure 5. VTRPE simulation results using the *M* profiles shown in Figure 4. (a) Propagation loss (PL) pattern for the sonde data, (b) the 1-parameter Paulus-Jeske model, (c) the 3-parameter Paulus-Jeske model, and (d) the Stacked model.

In this research we ended up using Genetic Algorithms instead as they provide global solutions without having to guess

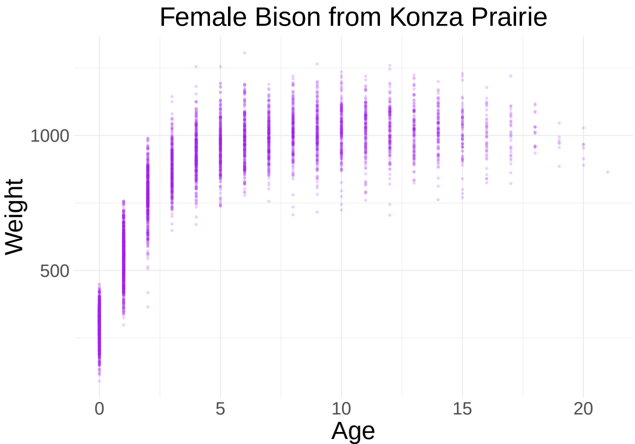
Operationally, the Navy doesn't have time to guess

Section 2

Using NLS in R

Let's apply NLS to our Female Bisons


```
knz_bison_age <- knz_bison %>%
  mutate(animal_age = rec_year - animal_yob) %>%
  filter(animal_sex=="F")
```



Use R Built in functions


```
df_nls<-nls(formula=   # Model we want to estimate,
            data    # Data we are evaluating,
            start   # Our initial guesses
            control # List of tolerance value, etc
            trace   # Do we want to see convergence
            upper   # Bounds on input parameters
            ... # some other useful stuff )
```

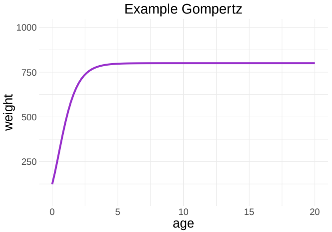
What Model to use?

Scour the literature or create your own (only with sufficient justification)

For our Bison, Martin and Barboza (2020) used a Gompertz model

$$BM = b1 * exp(-exp(-b2 * (age - b3))) \tag{3}$$

$b1$ = asymptotic body mass (pounds)
$b2$ = instantaneous growth-rate
$b3$ = age at inflection point years
age = Independent variable
BM = Body mass (pounds) Dependent variable



Create a function in R to test our model


```
gompertz<-function(b1,b2,b3,age){
 BM= b1*exp(-exp(-b2*(age-b3)))
return(BM)
}
```

Note: For the nls function it's okay to define all parameters like we did. In other optimization tools (e.g. optim) you would want to keep the first input index as a vector if you have multiple choice variables

Providing a guess is very important


```
df_nls<-nls(animal_weight~gompertz(b1,b2,b3,animal_ag
              data=knz_bison_age,
              start=list(b1=?,b2=?,b3=?),
              trace=TRUE )
```

The initial guesses and data also tell nls which variables are we trying to find and which data are we comparing

4 methods for providing guesses

1. Use past parameters from similar studies
2. Use data to internally define guesses (min, mean, max, etc.)
3. In 2-D, look at the graphs and estimate
4. In N-D, combine steps 1-2 then create a start grid to search over

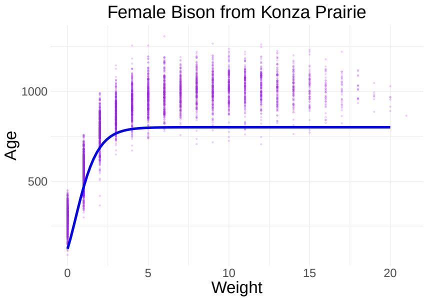
Applied guessing

How we could use step 2 and 3 in this case

Asymptotic body mass ($b1$) implies a max body length we could take the biggest observed female or generally look at the graph

Age inflection point ($b3$) is where the curve starts bending

Instantaneous growth-rate ($b2$) is kind of weird, but you could manipulate a set Gompertz model to see how it changes the shape and try to match it.



Apply guesses

```
b_gompertz<-nls(animal_weight~gompertz(b1,b2,b3,animal_age
                              data = knz_bison_age,
                              start = list(b1=1000,b2=1,b3=0.6),
                              trace = TRUE)
```

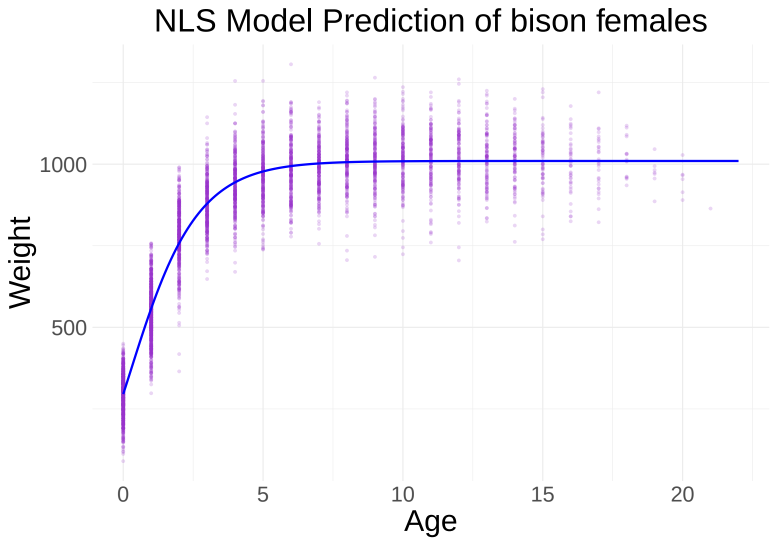
```
## 52917148 :  1e+03 1e+00 6e-01
## 34426629 :  1007.2008456    0.6930941    0.3444975
## 33207967 :  1009.6701568    0.7250854    0.2798164
## 33206308 :  1009.7635726    0.7275290    0.2832794
## 33206307 :  1009.7560207    0.7275973    0.2832921
```

What did the model find?


```
summary(b_gompertz)
```

```
##
## Formula: animal_weight ~ gompertz(b1, b2, b3, animal_age)
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
## b1 1.010e+03  1.903e+00  530.58   <2e-16 ***
## b2 7.276e-01  7.550e-03   96.38   <2e-16 ***
## b3 2.833e-01  8.064e-03   35.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 81.12 on 5046 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 3.106e-06
##   (236 observations deleted due to missingness)
```

How well does the model predict the data?



```r
model_aug<-broom::augment(b_gompertz)

sum_squared_error=sum((model_aug$.resid)^2)
print(sum_squared_error)
```

```
## [1] 33206307
```

If we compare different model runs from the trace output we can see this is the smallest sum of squared errors. No other model will get lower this number.

Adding confidence intervals is easy

```
conf<-as_tibble(predFit(b_gompertz,
            newdata = list(animal_age=age_series),
            interval="confidence"),
            level=0.95)

head(conf,n=4)
```

```
## # A tibble: 4 x 3
##      fit   lwr   upr
##    <dbl> <dbl> <dbl>
## 1  295.  291.  300.
## 2  322.  318.  327.
## 3  349.  345.  353.
## 4  376.  372.  380.
```

```
#plot+geom_ribbon(data=conf...)
```

Model fits so well, the confidence intervals don't even show on plot.

Section 3

Optimization across R

NLS falls under the optimization umbrella

Many flavors and varieties of optimization

As simply as possible

$$
\begin{aligned}
V(x,c) = \max_{c} f(x,c) \quad & \text{Subject to} \\
x &\geq 0 \\
c &\geq 0 \\
x &= g(x,c)
\end{aligned}
$$

Which method to use?

Depends on the question being asked

- What mathematical form is the optimization equation in?
    - Something like Maximum likelihood estimatation will be different than quadratic programming
- Do I need it to be fast or accurate?
- How many/form of parameters?

Best to use methods that you understand than ones you don't

Optimization toolkit highlights

optim/optimx: Workhorse functions in R and probably the ones you will use the most

-Article on why we should move towards optimx

quadprog: Used by a GP I advised in 2021

GA: Genetic algorithms are the best global tool that I know of

NLoptr: New project trying to keep syntax of alogrithms the same across computer languages

List of every optimization function with short descriptions