Course Project 2 Guidelines

Jelena H. Pantel

2024-01-23 07:48:35

Overview

You will give solutions to five problems on the topic of fitting observed data to models. You can submit your solutions by emailing me the solutions at jelena.pantel@uni-due.de. The solutions can either be (1) a fully executable RMarkdown file (.Rmd) or (2) an R script (.R). You **must** make sure the file runs 100% without errors before submitting (or that the .Rmd file knits - you can knit to HTML, PDF, your choice, whatever works).

Please note that the following R packages should be installed to get everything I do here to work:

```
library(ggplot2)
library(ISwR)
library(gauseR)
library(vegan)
library(mapsFinland)
library(brms)
```

Problem 1. Fitting, interpreting, and predicting from a linear model

Description: We can use a *linear regression* to model the relationship between two (or more) continuous variables. We'll work with a linear regression in R, and how to interpret the output (you will have more of this later in a proper statistics class, but let's review for now).

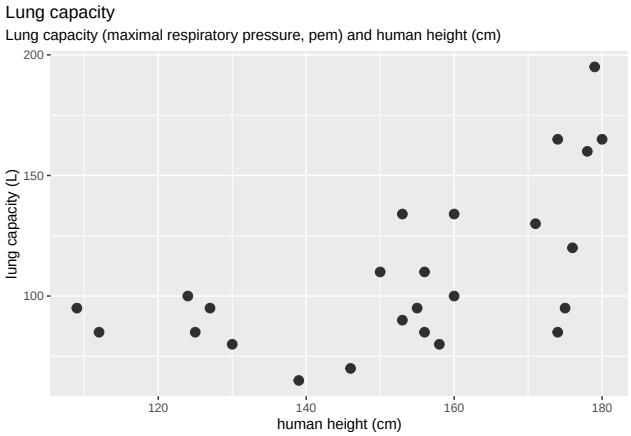
Data / Problem: We have some data about lung functioning, and we hypothesize that lung capacity (maximal respiratory pressure, measuring the strength of respiratory muscles, which I will refer to as *pem*) is driven by human height (cm). As a researcher, I collect data to test this hypothesize and see how well a linear model does describe the relationship, and I estimate the parameters in the linear model.

Our most basic linear model looks like this:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

$$pem_i = \beta_0 + \beta_1 height_i + \epsilon_i$$

This model has variables y and x, and parameters β_0 (the slope), β_1 (the y-intercept), and ϵ_i, the error term (the variance in y_i that is *not* explained by the linear model). Our goal is to **fit the data to the model**, and estimate the model parameters β_0, β_1, and ϵ_i that are most likely given the data. We use this existing dataset with observed values for `pemax` and `height`, fit a linear model to predict `pemax` from `height`, and then use the fitted model to make predictions for new cases.



The data can be found here:

```
library(ISwR)
data(cystfibr)
```

- Please use the `lm` command to fit a linear model to estimate y `pemax` as a function of x `height`: $y = mx + b$. Tell me the model estimates for the parameters b (*y-intercept*) and m (*slope*). Then, use the `predict` command to tell me the model-predicted values for lung capacity when the heights are 110, 140, and 170 cm.

Problem 2. Fitting, interpreting, and predicting from a logistic growth model using nls

Description: We considered a model for logistic growth:

$$\frac{dN}{dt} = rN(1 - \frac{N}{K})$$

where r is the population growth rate, K is the population carrying capacity, and N is the population size. We would like to evaluate some population data, fit it to a logistic growth model, and estimate parameters in that logistic growth model.

Data / Problem: We will use data for a Paramecium from Gause's experiments.

Figure 1: Paramecium aurelium (Artwork from PhyloPic, by Emily Jane McTavish, from http://chestofbooks.com/animals/Manual-Of-Zoology/images/I-Order-Ciliata-41.jpg)

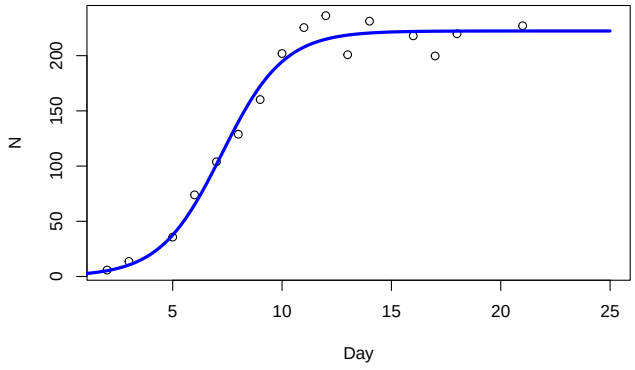
```
# load data
data(gause_1934_book_f22)
dat <- gause_1934_book_f22[gause_1934_book_f22$Treatment == "Pa",
    ]
```

- Create a plot of the population size over time (`dat$Days`) for species 2 only (`dat$Volume_Species2`). Then I would like you to use the `nls` command to fit this observed data to a logistic growth model (`Volume_Species2 ~ Day`).

- Recall that `nls` has some built-in models, and one of the was for logistic growth. You can find that via the function `SSlogis`. This will work similarly to the Michaelis-Menten (`SSmicmen`) example we did in class in Exercise 4.1 (E4.1).
- There is one very important consideration!! R's `nls` command does not use the exact same logistic model I show you above. Instead it fits the model as:

$$\frac{dN}{dt} = \frac{r}{1 + e^{(-K(N-N_m))}}$$

- This form is a bit different than what you are used to, but it still considers exponential growth at a rate r, then population size limitation by the carrying capacity K. There is a new parameter, N_m - this is the value of N at the inflection point of the logistic curve. So in this logistic model, there are **three** parameters you need to estimate. Please use the `nls` command to estimate the parameters r, K, and N_m.
- Report the parameter values given by the model fit.
- Please make a plot of the original data, and the curve fit by the logistic model produced by the `nls` command. Mine looks like this:



Problem 3. Fitting, interpreting, and predicting from a logistic growth model using gauseR

Data / Problem: For the same Paramecium dataset, use the function `gause_wrapper` in the R package `gauseR` to fit the data for Species 2 (the same data as above). This is yet another version of the logistic growth model. Instead of using carrying capacity K, it uses an *intraspecific competition coefficient* α_{ii}:

$$\frac{dN_i}{dt} = \frac{r}{1 + \alpha_{ii} N_i}$$

- So please estimate the values of r and α_{ii}, and show the plot with the raw data and curve fit. You will use the command `gause_wrapper`- note from E4.1 that you don't need to do much to get this to work!
- Please give me the model estimates for r (given in the output variable as `r1`) and for α_{ii} (given in the output variable as `a11`). Note that you can access this using the name of the output variable, then addending `$parameter_intervals`. So for example, if I assign the results of the `gause_wrapper` command to a variable called `gause_out`, then I can get the parameter values by typing `gause_out$parameter_intervals`. The values are given in the `mu` column.

Problem 4. Fitting, interpreting, and predicting from a species-area curve using vegan and nls

Description: A reliable pattern we can observe in nature is the relationship between area (of habitat) and number of species observed. You can read more about that in this paper, **Lomolino 2000**. The exact shape of this curve can take a few different forms. For example, here is a plot showing the number of terrestrial isopods on the central Aegean islands:

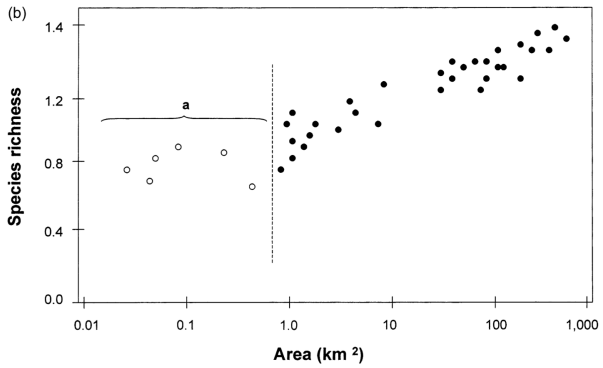


Figure 2: Species-area curve, terrestrial isopods, central Aegean islands. From Lomolino 2000.

One curve that can often describe the species-area relationship is an *Arrhenius curve*. The function for this is:

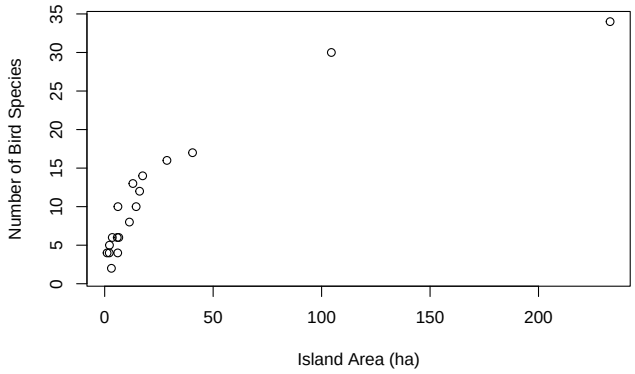
$$Species = k * area^z$$

Where parameter z is the steepness of the species-area curve, and k is the expected number of species in a unit area.

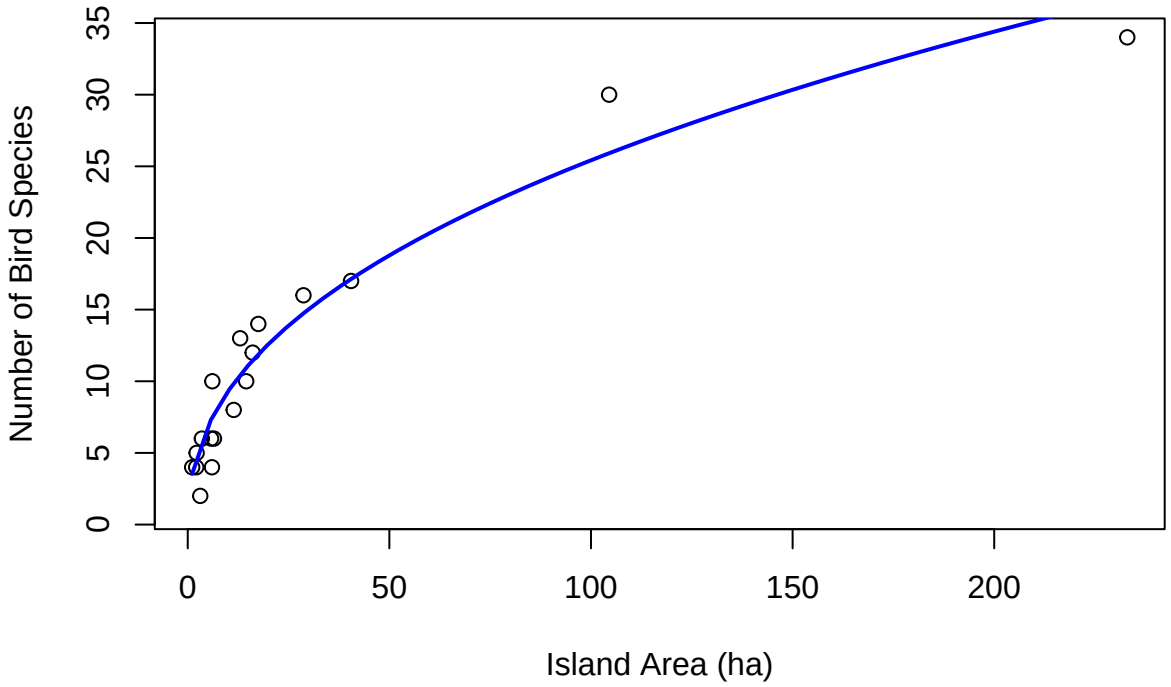
Data / Problem: We have some data for bird species on the Sipoo island archipelago (a district of Finland) (the data is using 'hectares' as units, a hectare is equal to 10,000 square meters).

```
data("seutukunnat2019")
ggplot(seutukunnat2019) + geom_sf() + ggtitle("Finland: Maps in R!")
```


```
## Get species area data: sipoo.map gives the areas of
## islands
data(sipoo, sipoo.map)
S <- specnumber(sipoo)
plot(S ~ area, sipoo.map, xlab = "Island Area (ha)", ylab = "Number of Bird Species",
    ylim = c(1, max(S)))
```



- An R package, `vegan` has some nls models included that are commonly used in ecology! After loading `library(vegan)`, please use the `nls` command and the `SSarrhenius` model to fit the model `S ~ area` and estimate the parameters `k` and `z`. Add the curve to the plot. Mine looks like this:



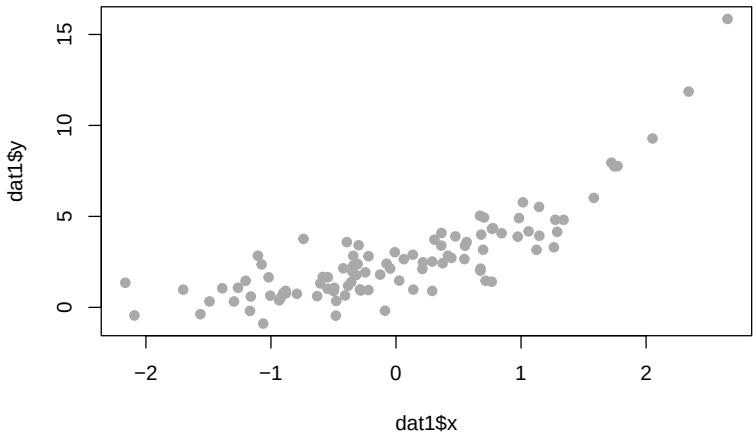
- Please give me the values for the estimated model parameters (you can use `summary` or `coef` on the fit nls model object to get these).

Problem 5. Bayesian non-linear modelling (new!)

Description: We recall from lecture 4.3 (L4.3) that we can use Bayesian estimation of model parameters. This works by specifying a *prior distribution* for each model parameter, sampling candidate values from this prior and calculating the *likelihood* of the observed data given those parameter values, accepting or rejecting samples, to produce an estimate of the *posterior distribution* for each model parameter. We will learn to do and interpret a Bayesian non-linear model here.

Data / Problem: To make sure this works well, you will simulate some data using the following code:

```
b <- c(2, 0.75)
x <- rnorm(100)
y <- rnorm(100, mean = b[1] * exp(b[2] * x))
dat1 <- data.frame(x, y)
plot(dat1$x,dat1$y,pch=19,col="darkgrey")
```



You can see directly that y is not a linear function of x, but instead an exponential function. Our goal is to use Bayesian inference, with the R package `brms`, to fit the data to the following model:

$$y_i = b_1 e^{b_2 x_i} + \sigma$$

And to estimate the values of the model parameters b_1 and b_2. We must specify a candidate *prior distribution* for the model parameters, then use the syntax of the R library `brms` to run the Bayesian estimation. Here I show the code for this, then ask you questions to interpret the model output.

```
library(brms)
# Set the parameter prior distribution
prior1 <- brms::prior(normal(1, 2), nlpar = "b1") + brms::prior(normal(0, 2), nlpar = "b2")
# Set up the brms model
fit1 <- brms::brm(bf(y ~ b1 * exp(b2 * x), b1 + b2 ~ 1, nl = TRUE),data = dat1, prior = prior1)
# Look at the Bayesian model results
summary(fit1)
# Plot the posterior distributions for the model parameters
plot(fit1)
# Plot the data with the model-fit curve
plot(conditional_effects(fit1), points = TRUE)
```

Questions to answer for homework:

1. Please state what *prior distribution* is used for the b_1 and b_2 parameters. Use the notation `b1 ~ DISTRIBUTION(dist parameters)`. So for example if the prior is a gamma distribution with rate and shape parameters 50 and 170, I would say `b1 ~ Gamma(50,170)`. I can write that in formula notation in RMarkdown by writing: `$$b_1 \sim Gamma(50,170)$$`.

2. Please give the *mean* and *95% HPDI (highest probability density interval)* for the three model parameters: b_1, b_2, and σ (*sigma*, the error term for the model). You can see this in the `summary(fit1` results. (the HPDI is the range that holds 95% of the probability density for estimates)

3. Bayesian estimation uses MCMC (*Markov Chain Monte Carlo*) to sample estimates from the prior distribution. The values of the model parameters selected in each step of the chain are shown as the *trace plots* to the right of the blue posterior distributions in the `plot(fit1)` command. Please use the following links to help answer the question - does it appear that the Markov chains in our Bayesian analysis have converged? (feel free to use eg the `bayesplot` library to make new trace plots if you desire)

Link 1: https://m-clark.github.io/bayesian-basics/model-exploration.html

Link 2: https://m-clark.github.io/bayesian-basics/diagnostics.html

4. Look at the help menu (`?brm`) for the model-fit command `brm` - how many Markov chains are used in this model (what is the default value for the argument `chains=x` ?)

5. Please use the library `bayesplot` to choose a new type of plot for analyzing your Bayesian model results. Create the figure, and explain what the figure tells us to help with the interpretation. Note that it is ok to choose a plot that re-draws some of the plots that result from the `plot(fit1)` command. You can read more about the `bayesplot` options here:

Link 3: http://mc-stan.org/bayesplot/