Lecture 10: Beginning optimization

Ciaran Evans

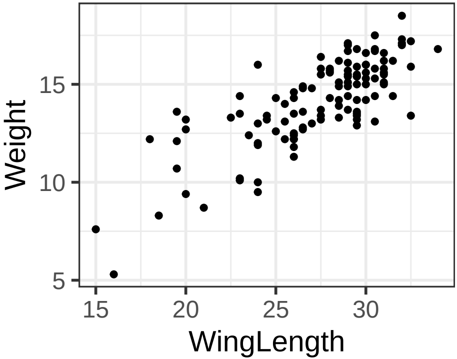
Course logistics

- Thanks for feedback!
- Tweak to grading policy: mastery at the question level, not the assignment level
  - Strictly better for your course grade
  - Won't penalize correct questions for mistakes on other questions
  - Resubmission still part of system
- Using other languages: if you would like to practice a different language for the R coding assignments, I am generally ok with that. Talk to me about it if you are interested
- Project 1 released

Motivation: regression models

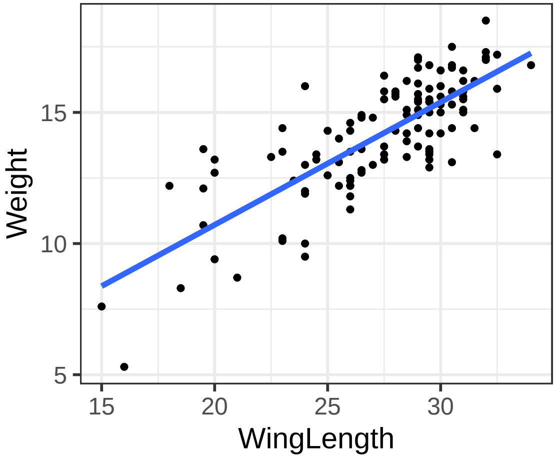
Data on 116 sparrows from Kent Island, New Brunswick.

- `Weight`: the weight of the sparrow (in grams)
- `WingLength`: the sparrow's wing length (in mm)



Question: How could I model the relationship between these two variables?

Motivation: linear regression



Question: How do I get the fitted regression line?

Motivation: linear regression

Population model: $\text{Weight}_i = \beta_0 + \beta_1 \text{WingLength}_i + \varepsilon_i$

Fitted model: $\widehat{\text{Weight}}_i = \widehat{\beta}_0 + \widehat{\beta}_1 \text{WingLength}_i$

In R:

```r
lm(Weight ~ WingLength, data = Sparrows)
```

```
Coefficients:
(Intercept)   WingLength
     1.3655       0.4674
```

Mathematically: $\widehat{\beta}_0, \widehat{\beta}_1$ are the values which *minimize* the residual sum of squares:

$$RSS(\beta_0, \beta_1) = \sum_{i=1}^{n} (\text{Weight}_i - \beta_0 - \beta_1 \text{WingLength}_i)^2$$

Overview: Optimization

Definition: *Optimization* is the problem of finding values that minimize or maximize some function.

Example:

$$RSS(\beta_0, \beta_1) = \sum_{i=1}^{n} (\text{Weight}_i - \beta_0 - \beta_1 \text{WingLength}_i)^2$$

- $RSS(\beta_0, \beta_1)$ is a function of β_0 and β_1
- We want to find the values of β_0 and β_1 that *minimize* this function

Question: How could we go about minimizing this function?

Overview: types of optimization methods

In this course, we will focus on two main types of optimization

- **Derivative-based methods:** use the derivative (and possibly higher-order derivatives too) to find a maximum/minimum.
- **Derivative-free methods:** do not use any derivatives (or approximations to derivatives).

We will begin with derivative-free methods.

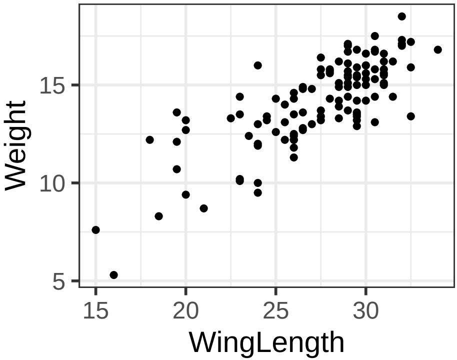
Optimization without a derivative

$$RSS(\beta_0, \beta_1) = \sum_{i=1}^{n} (\text{Weight}_i - \beta_0 - \beta_1 \text{WingLength}_i)^2$$

Question: How would you try to minimize $RSS(\beta_0, \beta_1)$ *without* taking a derivative? Brainstorm with your neighbor for 1-2 minutes, then we will discuss as a class.

Initial idea: grid search

- Define a set of values for β_0, β_1
- Calculate $RSS(\beta_0, \beta_1)$ for each pair of values
- Choose the values which minimize $RSS(\beta_0, \beta_1)$

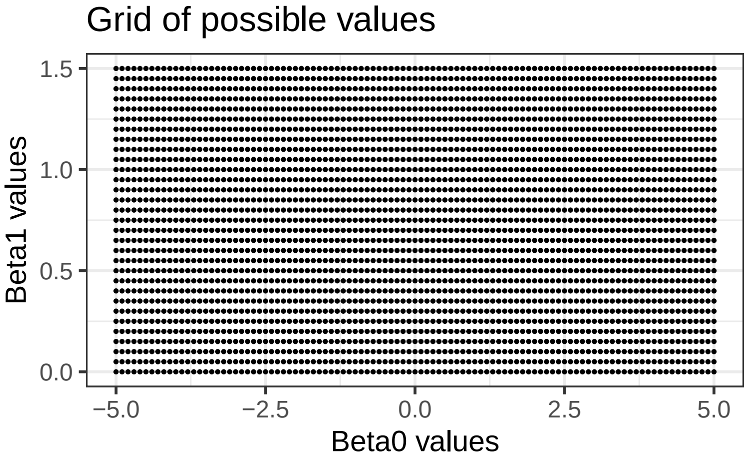


Question: What is a reasonable range of values to consider for β_0 and β_1?

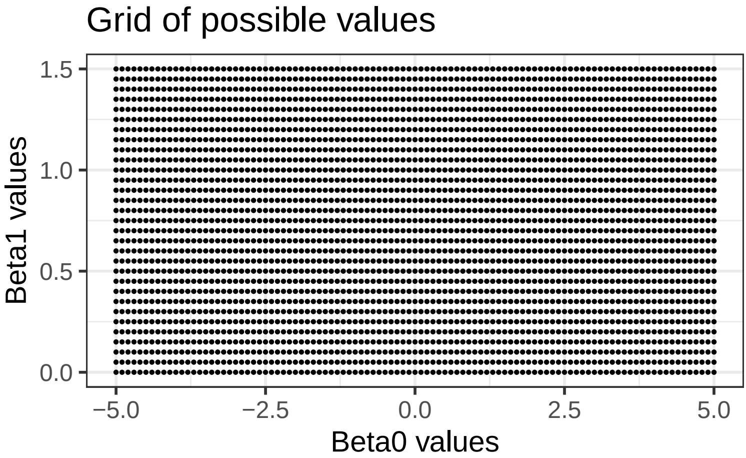
Initial idea: grid search

Consider values:

- $\beta_0 = -5,\ -4.9,\ -4.8, ...,\ 4.8,\ 4.9,\ 5$
- $\beta_1 = 0,\ 0.05, ...,\ 1.45,\ 1.5$

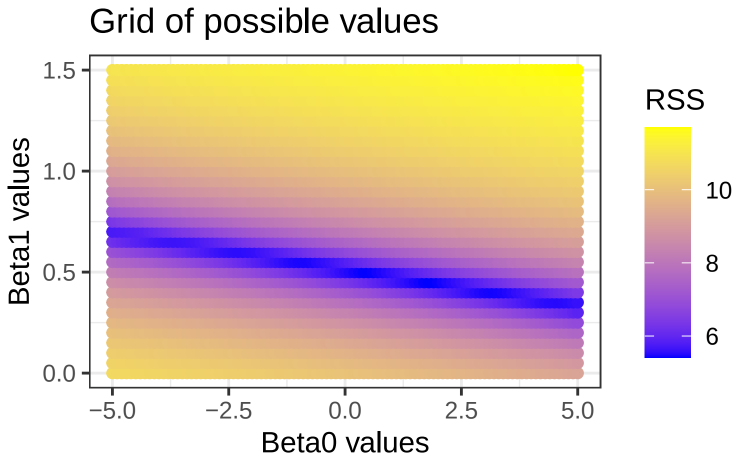


Initial idea: grid search

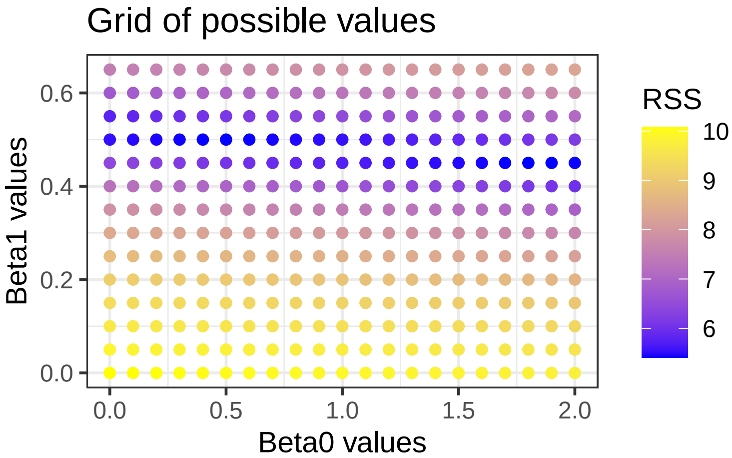


Now we calculate $RSS(\beta_0, \beta_1)$ for each possible pair in the grid.

Initial idea: grid search

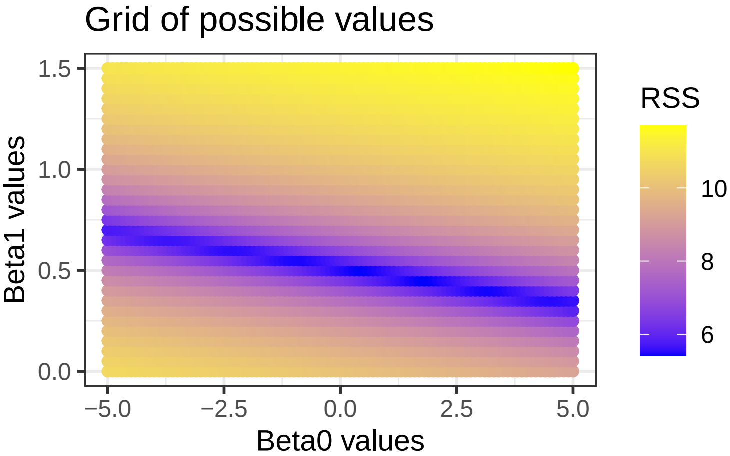


Initial idea: grid search



Combination with smallest RSS: $\beta_0 = 1.8,\ \beta_1 = 0.45$

Grid search: limitations



Question: What are some disadvantages of this grid search procedure?

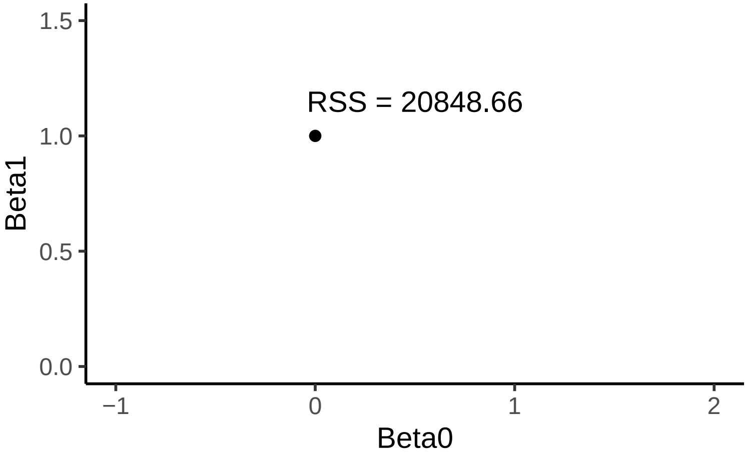
Grid search: limitations

For the basic grid search procedure described here:

- Does not scale well to higher dimensions (more coefficients)
- Requires a good selection of grid points
- Doesn't consider new values
- Can't tell when it is "close" to an optimal value

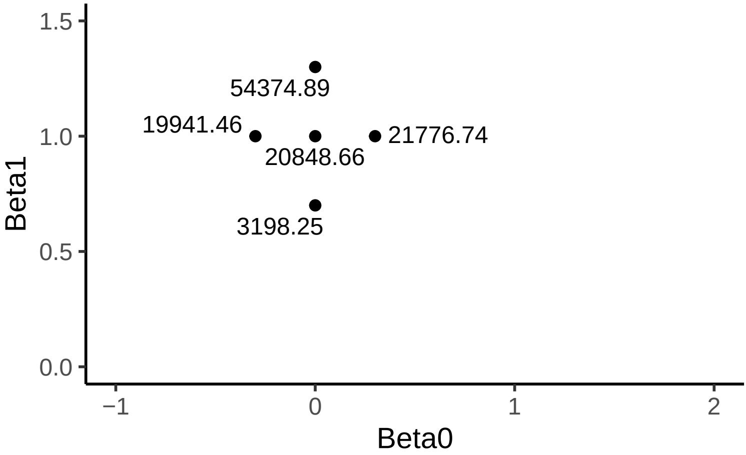
Better approach: compass search

Step 1: Start with an initial guess for β_0 and β_1, and calculate $RSS(\beta_0, \beta_1)$:



Better approach: compass search

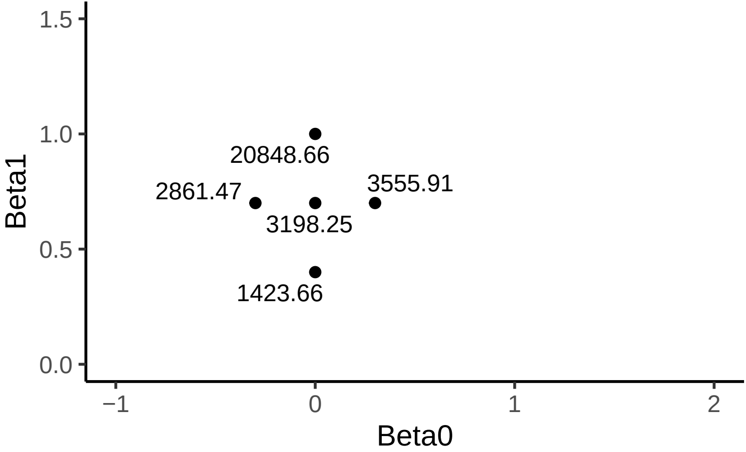
Step 2: Try test points in the four directions around the initial point:



Which of the 5 points is the best current guess for (β_0, β_1)?

Better approach: compass search

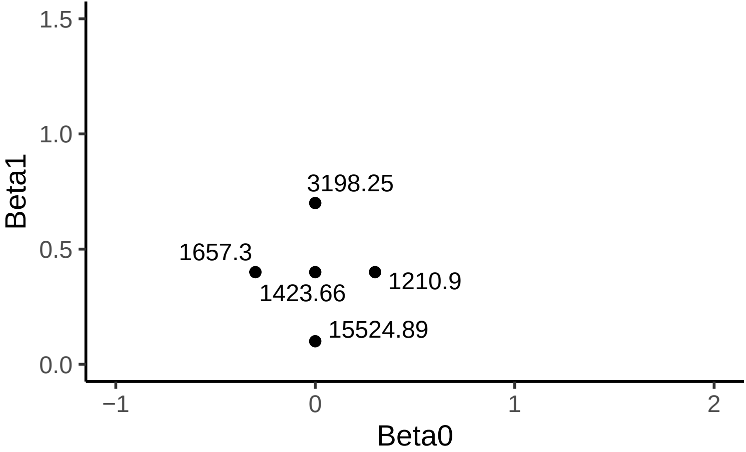
Step 3: If one of the four new points is better, move to the new best point:



Where do we move next?

Better approach: compass search

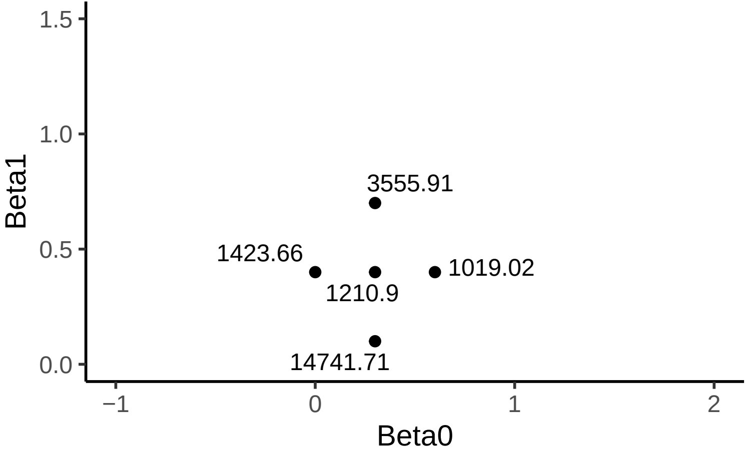
Step 3: If one of the four new points is better, move to the new best point:



Where do we move next?

Better approach: compass search

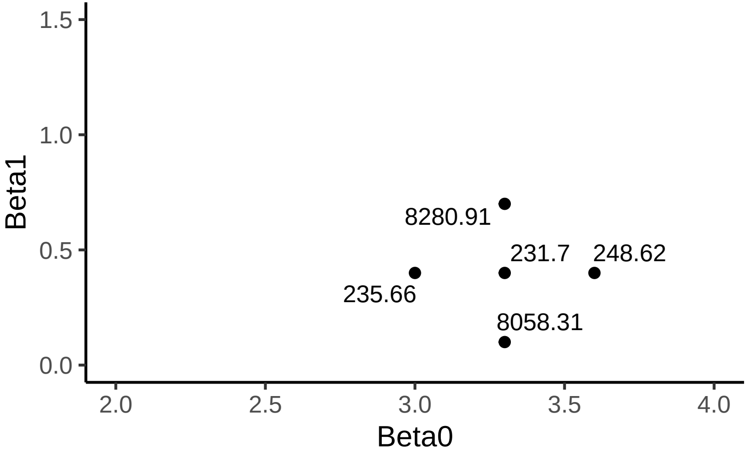
Step 3: If one of the four new points is better, move to the new best point:



Where do we move next?

Better approach: compass search

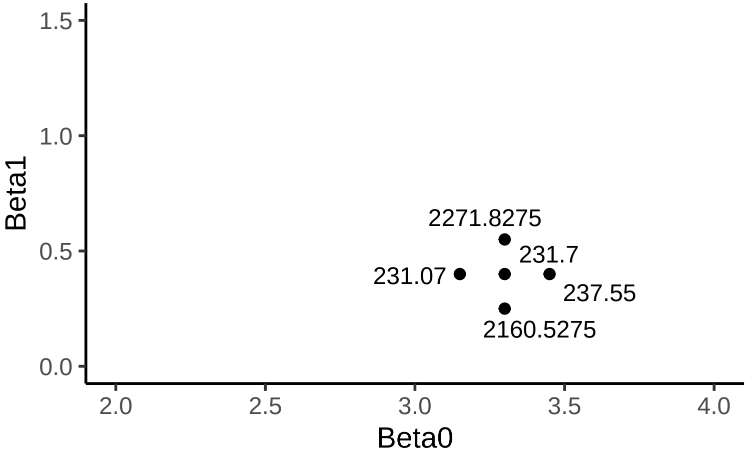
After a few more iterations, we end up here:



Where do we move next?

Better approach: compass search

Step 4: If none of the new points is an improvement, try again with half the distance:



Where do we move next?

Compass search overview (in 2 dimensions)

To minimize some function $f(\beta_0, \beta_1)$:

1. Choose an initial guess $(\beta_0^{(0)}, \beta_1^{(0)})$ and initial step size Δ_0
2. Evaluate f at the points
   - $(\beta_0^{(0)}, \beta_1^{(0)})$
   - $(\beta_0^{(0)}, \beta_1^{(0)} \pm \Delta_0)$
   - $(\beta_0^{(0)} \pm \Delta_0, \beta_1^{(0)})$
3. If f is smaller at one of the new points: move to the smallest value, update to $(\beta_0^{(1)}, \beta_1^{(1)})$
4. Otherwise: $\Delta_{k+1} = 0.5\Delta_k$ (shrink step size and try again)
5. Repeat

Your turn

Practice questions on the course website:

https://sta379-s25.github.io/practice_questions/pq_10.html

- Practice with compass search
- Start in class. You are welcome to work with others
- Practice questions are to help you practice. They are not submitted and not graded
- Solutions are posted on the course website