

# Stat 360 Project Rubric

Brad McNeney

## Project overview

The project is to create an R package that implements the Multivariate Adaptive Regression Splines (MARS) algorithm described in Friedman (1991):

Jerome H. Friedman

The Annals of Statistics , Mar., 1991, Vol. 19, No. 1 (Mar., 1991), pp. 1-67

The main function, `mars()`, takes a formula, data frame, and control object as input, fits the model by the forward and backward stepwise algorithms (Algorithms 2 and 3), and returns a `mars` object that contains the final least squares fit, along with a description of the basis functions from the final fit. You will also write `predict()`, `plot()`, `summary()`, and `print()` methods for `mars` objects. The `mars()` function and these four methods will be documented using `roxygen2` to produce R documentation files. Your package will include a test dataset that I will provide and unit tests of `mars()`, the `fwd_stepwise()` and `bwd_stepwise()` functions that implement the forward and backward algorithms, and the four methods. You will also include a package vignette that shows users how to use `mars()` and the four methods. In the vignette, include an analysis of one dataset that you find interesting (and that is not the test dataset). The dataset could be from another R package (easiest) or could be one that you have downloaded from another source and included as a package dataset.

The package will be made available in a `mars` directory within your group's project folder in the SFUStat360Projects GitHub repository. For example if your group's project folder is `MyGroup`, your R package will be at <https://github.com/SFUStatgen/SFUStat360Projects/tree/main/Projects/MyGroup/mars/>. You will also upload (i) a PDF of your vignette and (ii) a PDF of the package reference manual built with `devtools::build_manual()` to a Crowdmark assessment where they will be marked. The code on GitHub and documentation on Crowdmark are due at noon on Tuesday April 12.

## Grading Scheme

### Code on GitHub (25 marks)

1. REAMDE (1 marks): A README.md file should include your group members' names and student numbers.
2. R package structure (2 marks): The package should include DESCRIPTION, LICENCE and NAMESPACE files, and data, man, R, tests and vignettes directories.
3. `mars.R` (10 marks): The main `mars.R` file should include the `mars()` function and any others, such as `fwd_stepwise()`, that are called by `mars()`. Arrange your functions in a "top-down" manner, with higher-level functions appearing first, followed by successive levels of lower-level functions. You are graded on the following criteria:
  - Data structures (2): The input data structures should be a formula, data and `mars.control` object. The output data structure is an S3 object of class `mars` that inherits from class `lm`.
  - Correctness (5): The code should work correctly on the test dataset.
  - Readability (2): The steps and logic of your implementation should be clearly laid out. It should be easy for someone else in the class to read your code and understand what is going on.

- Efficiency (1): Take steps to avoid computational inefficiencies, such as excessive copying of large R objects.
4. Methods (12 marks): Include one file for each of the `plot()`, `predict()`, `print()` and `summary()` methods. For each method you are marked on:
- Correctness (2 marks): The method should work correctly on the test dataset.
  - Familiarity (1 marks): The method should look familiar to someone who has used the analogous method for `lm()` and `glm()`.

### Documentation (25 marks)

1. Vignette (10 marks): The vignette should include an analysis of one dataset that you find interesting (and that is not the test dataset). There is some overlap between the vignette and the documentation of the `mars()` function. Think of the vignette as long-form documentation that **teaches** a user how to use **all** of the features of your package, rather than the terse documentation of the individual functions that is a **reference** for users who already know something about the package and just want a refresher or a quick-start on specific functions. The vignette should:
  - (1 mark) be logically organized for the goal of introducing your package and its features to a user.
  - (2 marks) give a description of the MARS algorithm
  - (1 mark) show the user how to prepare the inputs and call `mars()`
  - (4 marks) shown the user how to use all four methods on a `mars` object
  - (2 marks) use the most interesting data you can find. Trivial examples will get no marks.
2. Documentation for `mars()` (11 marks): The documentation for your `mars()` function must be generated from roxygen2 comments in your `mars.R` source file. Marks are allocated as follows:
  - a. (2 marks) Description (brief) – a one- or two-line description of what the function does
  - b. (1 mark) Usage – how to call the function
  - c. (1 marks) Arguments – a list of arguments and their meaning
  - d. (2 marks) Details – a precise and detailed description of what the function does
  - e. (1 marks) Value – a description of the function’s return value
  - f. (1 mark) Author(s) – your name(s)
  - g. (1 mark) References – a reference to the Friedman paper and any other sources you think are necessary
  - h. (1 marks) See Also – a brief description of the methods written for MARS objects
  - i. (1 mark) Example – An example of how to use your function using the test dataset.
3. (4 marks) Documentation for the `predict()`, `plot()`, `summary()`, and `print()` methods will also be prepared as roxygen2 comments in each of the respective source files. You will receive one mark for each.