General Notes

- You will submit a minimum of two files, the core files must conform to the following naming conventions (including capitalization and underscores). 123456789 is a placeholder, please replace these nine digits with your nine-digit Bruin ID. The files you must submit are:
  1. *123456789_stats102b_hw1.Rmd*: Your markdown file which generates the output file of your submission.
  2. *123456789_stats102b_hw1.html/pdf*: Your output file, either a PDF or an HTML file depending on the output you choose to generate.

  If you fail to submit any of the required core files you will receive ZERO points for the assignment. If you submit any files which do not conform to the specified naming convention, you will receive (at most) half credit for the assignment.

- **Your .Rmd file must knit**. If your .Rmd file does not knit you will receive (at most) half credit for the assignment.
  The two most common reason files fail to knit are because of workspace/directory structure issues and missing include files. To remedy the first, ensure all of the file paths in your document are relative paths pointing at the current working directory. To remedy the second, simply make sure you upload any and all files you source or include in your .Rmd file.

- Your coding should adhere to the tidyverse style guide: https://style.tidyverse.org/.

- Any functions/classes you write should have the corresponding comments as the following format.

```
my_function ← function(x, y, ...){
#A short description of the function
#Args:
#x: Variable type and dimension
#y: Variable type and dimension
#Return:
#Variable type and dimension
Your codes begin here
}
```

NOTE: *Everything* you need to do this assignment is here, in your class notes, or was covered in discussion or lecture.

- Please **DO NOT** look for solutions online.
- Please **DO NOT** collaborate with anyone inside (or outside) of this class.
- Please work **INDEPENDENTLY** on this assignment.
- Please use the functions only from **base** and **stats** packages.

1. Zero-order Coordinate Search

(a) Please devise a pseudo-code for the coordinate search algorithm and implement it as a function (coordinate_search()) in R for two-dimensional input. The function takes the following arguments:

> **g**: A cost function
>
> **alpha_choice**: If it is "diminishing", execute random search with diminishing steplength. If it is a numerical value, let it be the fixed steplength (default: 1).
>
> **max_its**: The maximum number of iteration (default: 100)
>
> **w0**: An initial point
>
> **tol**: The tolerance for the stop criteria (default: 10^{-5})

Your function should return a list object which includes

> **weight_history**: A $n \times 2$ matrix for weight history
>
> **cost_history**: A vector for corresponding cost function history

(b) Suppose we have $g(w_1, w_2) = w_1^2 + w_2^2 + 2$. Draw the plots of weight_history and cost_history to showcase your function with $\mathrm{w0}^T = [4\ 4]$.

(c) Repeat (b) with a steplength diminishing rule.

(d) Given $g(w_1, w_2) = 0.26(w_1^2 + w_2^2) - 0.48w_1w_2$, please find the global minimum of the given cost function and explain the way you do.

2. Please devise a pseudo-code for the zero-order coordinate descent algorithm and implement it as a function (coordinate_descent_0()) in R for two-dimensional input. Repeat (b), (c), and (d) in problem 1.

3. First-order Coordinate Descent

(a) Please express the coordinate descent algorithm as a local optimization scheme.

(b) Write R code for the particular case of a quadratic function from Example 2 of Chapter 5 and repeat the experiment.

(c) Repeat (b) with a steplength diminishing rule.

4. Convex function has the following two properties:

1. The sum of convex functions is always convex, and
2. The maximum of convex functions is convex.

Please prove these two results.

5. Gradient Descent

(a) Please devise a pseudo-code for the gradient descent algorithm and implement it as a function gd() in R. The function takes the following arguments:

> **g**: A cost function
>
> **g_prime**: The corresponding gradient
>
> **alpha_choice**: If it is "diminishing", execute random search with diminishing steplength. If it is a numerical value, let it be the fixed steplength (default: 1).
>
> **max_its**: The maximum number of iteration (default: 100)
>
> **w0**: An initial point
>
> **tol**: The tolerance for the stop criteria (default: 10^{-5})

Your function should return a list object which includes

> **weight_history**: A matrix for weight history
>
> **cost_history**: A vector for corresponding cost function history

(b) Suppose we have $g(w) = \frac{1}{50}(w^4 + w^2 + 10w)$. Make three separate runs with $\alpha = 1$, 0.1, and 0.01. Draw the plots of weight_history and cost_history of each run to showcase your function with an initial point 2. Please discuss your results.

(c) Repeat (b) with a steplength diminishing rule.

(d) Use your gd() to find the minimum of the following two functions:

$$g(x, y) = f(x) + f(y) = x^2 + y^2 - 2x - 2y + 6$$
$$G(x, y) = g(x) + g(y) = [(x^4 + y^4)/4] - [(x^3 + y^3)/3] - x^2 - y^2 + 4$$

Also, draw the plots of weight_history and cost_history.

6. Consider two $N = 2$ dimensional quadratic functions that take the general form $g(w) = a + b^T w + w^T C w$, where a and b are set to zero, and matrix C is set as follows:

$$C = \begin{bmatrix} .5 & 0 \\ 0 & 9 \end{bmatrix}$$
$$C = \begin{bmatrix} .1 & 0 \\ 0 & 9 \end{bmatrix}$$

In class, we demonstrate the zig-zag behavior of gradient descent with these two cases. Please implement the momentum-accelerated gradient descent scheme to repeat the experiments. Discuss your results using a cost function history plot.