BEE 4750 Lab 1: Julia and GitHub Basics

Name:

ID:

Due Date

Thursday, 8/28/25, 9:00pm

Setup

The following code should go at the top of most Julia scripts; it will load the local package environment and install any needed packages. You will see this often and shouldn't need to touch it.

```julia
import Pkg
Pkg.activate(".")
Pkg.instantiate()
```

```
  Activating project at `c:\Users\Lyy\lab1-132140636yyy`
   Installed libdecor_jll           v0.2.2+0
   Installed libfdk_aac_jll          v2.0.4+0
   Installed GR_jll                v0.73.17+0
   Installed LERC_jll               v4.0.1+0
   Installed JpegTurbo_jll            v3.1.1+0
   Installed x265_jll               v4.1.0+0
   Installed Libmount_jll            v2.41.0+0
   Installed Preferences             v1.4.3
   Installed Opus_jll               v1.5.2+0
   Installed LoggingExtras            v1.1.0
   Installed Xorg_xkbcomp_jll          v1.4.7+0
   Installed RelocatableFolders         v1.0.1
   Installed Unitful                v1.24.0
```

```
Installed Contour              v0.6.3
Installed Measures              v0.3.2
Installed Grisu              v1.0.2
Installed ConcurrentUtilities       v2.5.0
Installed Xorg_xcb_util_wm_jll       v0.4.2+0
Installed Xorg_xcb_util_image_jll      v0.4.1+0
Installed PlotUtils             v1.4.3
Installed RecipesPipeline         v0.6.12
Installed Xorg_libSM_jll          v1.2.6+0
Installed OpenSSL              v1.5.0
Installed DelimitedFiles          v1.9.1
Installed Xorg_xcb_util_jll         v0.4.1+0
Installed Cairo_jll             v1.18.5+0
Installed HTTP               v1.10.17
Installed Xorg_libXinerama_jll       v1.1.6+0
Installed Fontconfig_jll          v2.16.0+0
Installed Xorg_libxkbfile_jll       v1.1.3+0
Installed EpollShim_jll           v0.0.20230411+1
Installed ColorSchemes           v3.30.0
Installed Statistics            v1.11.1
Installed GR                v0.73.17
Installed Xorg_libXau_jll          v1.0.13+0
Installed IrrationalConstants       v0.2.4
Installed Missings              v1.2.0
Installed FFMPEG              v0.4.4
Installed Pango_jll             v1.56.3+0
Installed PtrArrays             v1.3.0
Installed xkbcommon_jll           v1.9.2+0
Installed Xorg_xcb_util_keysyms_jll    v0.4.1+0
Installed Showoff              v1.0.3
Installed Bzip2_jll             v1.0.9+0
Installed NaNMath              v1.1.3
Installed LZO_jll              v2.10.3+0
Installed XZ_jll              v5.8.1+0
Installed SimpleBufferStream        v1.2.0
Installed PlotThemes            v3.3.0
Installed HarfBuzz_jll           v8.5.1+0
Installed IJulia              v1.29.2
Installed fzf_jll              v0.61.1+0
Installed SoftGlobalScope         v1.1.0
Installed x264_jll              v10164.0.1+0
Installed FriBidi_jll            v1.0.17+0
Installed GLFW_jll              v3.4.0+2
Installed UnicodeFun            v0.4.1
Installed MbedTLS              v1.1.9
Installed TranscodingStreams        v0.11.3
Installed FreeType2_jll           v2.13.4+0
Installed StatsAPI             v1.7.1
Installed JLFzf               v0.1.11
Installed CodecZlib             v0.7.8
Installed Compat              v4.18.0
Installed DataStructures          v0.18.22
Installed StatsBase             v0.34.5
Installed Colors              v0.13.1
Installed libpng_jll            v1.6.50+0
Installed Xorg_libxcb_jll          v1.17.1+0
Installed mtdev_jll             v1.1.7+0
Installed libaom_jll            v3.12.1+0
Installed ExceptionUnwrapping        v0.1.11
Installed Dbus_jll             v1.16.2+0
Installed ColorTypes            v0.12.1
Installed Scratch              v1.3.0
Installed eudev_jll             v3.2.14+0
Installed Xorg_libXext_jll         v1.3.7+0
Installed Xorg_xcb_util_cursor_jll     v0.1.5+0
Installed Expat_jll             v2.6.5+0
Installed Zstd_jll             v1.5.7+1
Installed Libtiff_jll            v4.7.1+0
Installed TensorCore            v0.1.1
Installed Libffi_jll            v3.4.7+0
Installed Format              v1.3.7
Installed Xorg_libXrender_jll        v0.9.12+0
Installed libinput_jll           v1.28.1+0
Installed Plots               v1.40.17
Installed OrderedCollections        v1.8.1
Installed libevdev_jll           v1.13.4+0
Installed ColorVectorSpace         v0.11.0
Installed Ogg_jll              v1.3.6+0
Installed Qt6ShaderTools_jll        v6.8.2+1
Installed Xorg_libXi_jll          v1.8.3+0
Installed Reexport             v1.2.2
Installed Qt6Declarative_jll        v6.8.2+1
Installed Vulkan_Loader_jll         v1.3.243+0
Installed AliasTables            v1.1.3
Installed LogExpFunctions          v0.3.29
Installed Libuuid_jll            v2.41.0+0
```

```
   Installed Xorg_libXcursor_jll       v1.2.4+0
   Installed Xorg_libICE_jll         v1.1.2+0
   Installed Xorg_xcb_util_renderutil_jll  v0.3.10+0
   Installed MacroTools            v0.5.16
   Installed Graphite2_jll          v1.3.15+0
   Installed StableRNGs            v1.0.3
   Installed DocStringExtensions       v0.9.5
   Installed libass_jll            v0.17.4+0
   Installed Xorg_xtrans_jll         v1.6.0+0
   Installed BitFlags             v0.1.9
   Installed Pixman_jll            v0.44.2+0
   Installed OpenSSL_jll           v3.5.1+0
   Installed Wayland_jll           v1.24.0+0
   Installed FFMPEG_jll            v7.1.1+0
   Installed Latexify             v0.16.8
   Installed Xorg_xkeyboard_config_jll    v2.44.0+0
   Installed LLVMOpenMP_jll          v18.1.8+0
   Installed DataAPI              v1.16.0
   Installed Xorg_libXrandr_jll        v1.5.5+0
   Installed RecipesBase           v1.3.4
   Installed FixedPointNumbers        v0.8.5
   Installed Xorg_libXfixes_jll        v6.0.1+0
   Installed LAME_jll             v3.100.3+0
   Installed Qt6Wayland_jll          v6.8.2+1
   Installed GettextRuntime_jll        v0.22.4+0
   Installed Qt6Base_jll           v6.8.2+1
   Installed Libiconv_jll           v1.18.0+0
   Installed LaTeXStrings           v1.4.0
   Installed URIs               v1.6.1
   Installed libvorbis_jll          v1.3.8+0
   Installed Glib_jll             v2.84.3+0
   Installed Libglvnd_jll           v1.7.1+1
   Installed Xorg_libXdmcp_jll        v1.1.6+0
   Installed Requires             v1.3.1
   Installed Xorg_libX11_jll         v1.8.12+0
   Installed Unzip               v0.2.0
   Installed UnitfulLatexify         v1.7.0
   Installed SortingAlgorithms        v1.2.1
    Building IJulia → `C:\Users\Lyy\.julia\scratchspaces\44cfe95a-1eb2-52ea-b672-e2afdf69b78f
Precompiling project...
   6719.3 ms   LaTeXStrings
   4732.8 ms   Contour
   5373.7 ms   StatsAPI
```

```
 5970.7 ms    TensorCore
 4214.3 ms    Format
 4761.0 ms    Measures
 5862.1 ms    Statistics
 3899.8 ms    OrderedCollections
 6055.7 ms    Grisu
 5369.6 ms    Requires
 8803.6 ms    MacroTools
 6033.5 ms    StableRNGs
 6676.2 ms    Unzip
 3954.7 ms    Reexport
 5243.8 ms    DocStringExtensions
 7474.7 ms    IrrationalConstants
 4771.1 ms    SimpleBufferStream
 5418.3 ms    URIs
 4172.2 ms    PtrArrays
 6679.0 ms    TranscodingStreams
 4885.9 ms    DelimitedFiles
 5567.9 ms    NaNMath
 4319.4 ms    DataAPI
 3694.5 ms    BitFlags
 6964.4 ms    ConcurrentUtilities
 3166.8 ms    Scratch
 4195.4 ms    LoggingExtras
 6877.3 ms    MbedTLS
 4823.3 ms    Preferences
 4273.6 ms    Compat
 4180.6 ms    SoftGlobalScope
 4206.8 ms    ExceptionUnwrapping
 7185.0 ms    UnicodeFun
 6267.2 ms    Statistics → SparseArraysExt
 4365.7 ms    Showoff
 9209.1 ms    FixedPointNumbers
 4207.8 ms    LogExpFunctions
 3571.9 ms    AliasTables
 4367.2 ms    CodecZlib
10482.3 ms    Latexify
 4085.0 ms    Missings
 5211.8 ms    RelocatableFolders
 4476.1 ms    JLLWrappers
 4246.1 ms    PrecompileTools
 3677.1 ms    Compat → CompatLinearAlgebraExt
 5230.4 ms    Latexify → SparseArraysExt
```

```
  6484.1 ms   ColorTypes
  4076.1 ms   OpenSSL_jll
  5187.3 ms   Graphite2_jll
  4015.0 ms   Libmount_jll
  3474.7 ms   EpollShim_jll
  4841.5 ms   LLVMOpenMP_jll
  4349.9 ms   Bzip2_jll
  5623.4 ms   libsodium_jll
  4995.0 ms   Xorg_libICE_jll
  4362.9 ms   Xorg_libXau_jll
  6335.1 ms   libpng_jll
 46580.0 ms   Unitful
  6488.9 ms   libfdk_aac_jll
  5876.9 ms   LAME_jll
  4590.0 ms   LERC_jll
  6349.5 ms   fzf_jll
  3719.1 ms   XZ_jll
  5047.7 ms   JpegTurbo_jll
  5784.9 ms   Ogg_jll
  4604.3 ms   mtdev_jll
  6325.8 ms   Xorg_libXdmcp_jll
  5681.0 ms   x265_jll
  4394.7 ms   x264_jll
  6144.1 ms   Zstd_jll
  5563.4 ms   libaom_jll
  3705.3 ms   Expat_jll
  5627.2 ms   LZO_jll
  5758.6 ms   Opus_jll
  3938.3 ms   Xorg_xtrans_jll
  5717.2 ms   libevdev_jll
  5882.2 ms   Libiconv_jll
  3886.3 ms   Libffi_jll
  5757.3 ms   eudev_jll
  5152.9 ms   Libuuid_jll
  4201.7 ms   FriBidi_jll
  5598.9 ms   RecipesBase
  6041.9 ms   ColorTypes → StyledStringsExt
  7255.7 ms   DataStructures
  8001.3 ms   ColorVectorSpace
  3361.9 ms   Pixman_jll
  4740.4 ms   FreeType2_jll
  6705.9 ms   OpenSSL
  4293.4 ms   ZeroMQ_jll
```

```
14921.7 ms    Colors
 4805.7 ms    Xorg_libSM_jll
 4637.1 ms    Unitful → PrintfExt
 6347.0 ms    JLFzf
 3876.8 ms    libvorbis_jll
 5179.5 ms    Xorg_libxcb_jll
 4684.2 ms    Libtiff_jll
 4221.9 ms    Dbus_jll
 7034.6 ms    GettextRuntime_jll
 6369.5 ms    Wayland_jll
 5036.4 ms    libinput_jll
31639.2 ms    Parsers
 4251.3 ms    SortingAlgorithms
 5535.6 ms    Fontconfig_jll
 6308.6 ms    UnitfulLatexify
 3575.8 ms    Xorg_xcb_util_jll
 8915.1 ms    ZMQ
 2909.5 ms    Xorg_libX11_jll
12429.8 ms    ColorSchemes
 5817.5 ms    Glib_jll
 7021.6 ms    JSON
 4553.9 ms    Xorg_xcb_util_image_jll
 4669.1 ms    Xorg_xcb_util_keysyms_jll
 8785.6 ms    StatsBase
 5060.6 ms    Xorg_xcb_util_renderutil_jll
 3957.8 ms    Xorg_xcb_util_wm_jll
 5927.6 ms    Xorg_libXrender_jll
 4445.5 ms    Xorg_libXext_jll
 5005.0 ms    Xorg_libXfixes_jll
 4857.4 ms    Xorg_libxkbfile_jll
 4118.2 ms    Xorg_xcb_util_cursor_jll
 5406.3 ms    Conda
 4693.4 ms    Xorg_libXinerama_jll
 3526.5 ms    Xorg_libXrandr_jll
 3714.7 ms    Libglvnd_jll
 6195.6 ms    Cairo_jll
 5529.6 ms    Xorg_libXcursor_jll
 5584.0 ms    Xorg_libXi_jll
 4654.5 ms    Xorg_xkbcomp_jll
 7602.1 ms    HarfBuzz_jll
 5503.1 ms    Xorg_xkeyboard_config_jll
51180.2 ms    HTTP
 7067.7 ms    libass_jll
```

```
   5583.9 ms  xkbcommon_jll
   6342.8 ms  Pango_jll
  33011.8 ms  PlotUtils
  17962.0 ms  IJulia
   4374.9 ms  Vulkan_Loader_jll
   4519.2 ms  libdecor_jll
   6961.0 ms  FFMPEG_jll
   3572.8 ms  GLFW_jll
   6672.0 ms  Qt6Base_jll
   5248.2 ms  FFMPEG
  11235.7 ms  PlotThemes
  16678.5 ms  RecipesPipeline
   8169.3 ms  Qt6ShaderTools_jll
   9796.7 ms  GR_jll
   9478.6 ms  Qt6Declarative_jll
   3377.0 ms  Qt6Wayland_jll
  13118.4 ms  GR
 121885.2 ms  Plots
  12647.2 ms  Plots → UnitfulExt
  14043.0 ms  Plots → IJuliaExt
  152 dependencies successfully precompiled in 384 seconds. 36 already precompiled.
```

This next cell loads packages which are required for the rest of the code evaluation. In this case, we only need to load the `Plots.jl` plotting package, but you will see others over the course of the semester (and can add more if desired; just make sure that you've added the new packages to the environment). Standard Julia practice is to load all of the needed packages at the top of the file.

Warning

Loading packages can take a while, especially the first time! Julia tries to precompile all of the packages you're using so repeat use is faster, but this can be quite slow at first.

```julia
using Plots
```

Introduction

Julia

Julia is an up-and-coming language, originally developed for scientific programming. While learning a new programming language always has its hiccups, the good news is that if you've programmed in a high-level language such as Python or MATLAB, most Julia concepts should look familiar.

If you have not successfully set up Julia, follow the instructions in Tools Setup and/or ask for help.

You can use other editors for this course, but our recommendation is Visual Studio Code with the Julia extension, which will make life a *lot* simpler! You should have set this up by following the Tools Setup instructions, but if not, do so now and/or ask for help.

Jupyter Notebooks

Jupyter notebooks integrate text and equations in Markdown with Julia (or Python, or R) code. To do this, Jupyter notebooks consist of two types of "cells": code cells and Markdown (text) cells.

Click once on this section of text. A box will appear around this text (and some areas above/below it) - all of that is within this cell.

Markdown is a text markup framework for formatting language that makes things look pretty when viewed across different platforms: web browsers, notebooks, and so forth. Text written in Markdown can also include hyperlinks, LaTeX equations, section headers, and images, among other features (most of the course website and the lecture notes were all written in Markdown!). Here is a basic Markdown cheat sheet.

What you are looking at right now is the formatted text after the Markdown is processed. To see the raw Markdown, do one of:

- press `Enter` while that cell is selected, or
- double-click on that cell.

A couple of the features you will see in this Markdown cell:

- The `---` command creates a horizontal line. This is also nice for separating sections.
- Backticks (`` `...` ``) can be used to format and highlight code, keystrokes, etc.
- The `#` sign is used to create a new section header; two `#` signs (`##`) is used to create a new subsection header; `###` creates a subsubsection, and so on.
- You can create a bulleted list by using the asterisk `*` or a dash `-` and a space.
- You can create regular text by just typing as usual.
- You can create **bold-faced text** by wrapping it with two asterisks on both sides.
- You can create *italicized text* by wrapping it with a single asterisk on both sides.

- To create a new paragraph, you must include a blank line between the old and new paragraphs.

At this point you might be wondering how to turn this cell back into the fully formatted Markdown text instead of the raw Markdown you're probably still looking at. You have a couple of options, depending on your platform, but the most consistent is to type `Shift + Enter` to **execute** the cell (this is also how to run code, but more on that later).

Additionally, you will frequently need to create new cells in your Jupyter notebooks. How you do this will depend on how you interact with the notebook, but try to figure this out now.

One tip is to think carefully about what bits of code should be in the same cell, as you typically only see output from the last command in a cell. For example, compare the following:

```julia
x = 5
sin(x)
```

```
-0.9589242746631385
```

with

```julia
x = 5
```

```
5
```

```julia
sin(x)
```

```
-0.9589242746631385
```

In Julia, you can also suppress the output of a command with a semi-colon:

```julia
sin(x);
```

which can help if you want to split some code out for clarity or to insert some text prior to it, but don't want to clutter the notebook with its output.

For code cells, to execute the commands within the cell, we also press `Shift+Enter`.

Finally, **make sure that you evaluate all of the code cells in order before submitting**. One bad outcome with notebooks occurs when cells are evaluated out of order, so fixed bugs and edits in previous cells do not get a chance to propagate down. You can do this with the `Run All` command in whichever interface you're using to edit your notebook.

Julia Basics

There are many tutorials and references for Julia, including a basics overview on the class website. Please feel free to reference these as you work through any part of the course.

Formatting Math

It will often be helpful to include nicely-formatted mathematics in a notebook. Markdown accomodates this using LaTeX syntax. A LaTeX cheatsheet is available on the class website, and many other guides exist online.

Below is an example of a formatted equation:

$$x = 5.$$

Looking For Help

There is no shame in using Google, or other resources, for help when programming. There are many, many times when you can't quite get the syntax to work, can't quite figure out the right package or command to use, or are feeling too lazy or overwhelmed (I'm not judging either of those!) to dig through the documentation. Some good resources include:

- Stack Overflow is a treasure trove of answers;
- The official Julia forum and the Julia Subreddit are also very useful.

You are also highly encouraged to post on Ed Discussion, though getting a response might be less immediate. Just be mindful that to get good answers, you have to help people help you, and **make sure to give credit to any resources that were helpful**!

Exercises (3 points)

Use your understanding of Julia syntax and the GitHub workflow to complete the following (hopefully short) exercises. Convert your completed lab assignment to a PDF and submit it to the Gradescope Assignment "Lab 1".

Remember to:

- Include a (succint but clear) writeup of the core idea underlying your code, through some combination of equations, text, and algorithms. As you are not required to submit your code, we will not be looking at it in detail, and instead will rely on those writeups to assess whether your approach is correct.

- If using the notebook, evaluate all of your code cells, in order (using a `Run All` command). This will make sure all output is visible and that the code cells were evaluated in the correct order.
- Tag each of the problems when you submit to Gradescope; a 10% penalty will be deducted if this is not done.

Computing a Dot Product

Given two numeric arrays `x` and `y`, write a function to compute their dot product if they have equal length, and return an error if not (this is useful for debugging!). Use the following code as a starting point. **Do not use a built-in function**.

```julia
function dot_product(x, y)
    if ... # insert test condition for equal lengths
        # compute and return dot product
    else
        throw(DimensionMismatch("length of x not equal to length of y"))
    end
end
```

```
Base.Meta.ParseError: ParseError:
# Error @ c:\Users\Lyy\lab1-132140636yyy\jl_notebook_cell_df34fa98e69747e1a8f8a730347b8e2f_X1
function dot_product(x, y)
    if ... # insert test condition for equal lengths
#           invalid identifier
ParseError:

# Error @ c:\Users\Lyy\lab1-132140636yyy\jl_notebook_cell_df34fa98e69747e1a8f8a730347b8e2f_X1

function dot_product(x, y)

    if ... # insert test condition for equal lengths

#           invalid identifier




Stacktrace:

 [1] top-level scope

   @ c:\Users\Lyy\lab1-132140636yyy\jl_notebook_cell_df34fa98e69747e1a8f8a730347b8e2f_X15sZm
```

Here are some tests to make sure your code works as intended. Tests like these are useful to make sure everything works as intended. One reason to split your code up into functions is that it makes it straightforward to write tests to make sure each piece of your code works, which makes it easier to identify where errors are occuring.

```
dot_product([1 2 3], [4 5 6])
```

```
UndefVarError: UndefVarError: `dot_product` not defined in `Main`
Suggestion: check for spelling errors or missing imports.
UndefVarError: `dot_product` not defined in `Main`

Suggestion: check for spelling errors or missing imports.



Stacktrace:

 [1] top-level scope

   @ c:\Users\Lyy\lab1-132140636yyy\jl_notebook_cell_df34fa98e69747e1a8f8a730347b8e2f_X20sZm
```

If you know the value you should get, you can write a more formal test using the `@assert` macro, which is a good way to "automate" checking (since you get an obvious error if the code doesn't work as desired):

```
@assert dot_product([1 2 3], [4 5 6]) == 32
```

Let's also make sure we get an error when the dimensions of the two vectors don't match:

```
dot_product([1 2 3], [4 5])
```

Making a Plot

Write a function to compute the square of an integer `x`. Evaluate this function for integers between $x = -5$ and $x = 5$ and make a plot of the squared values (you can find a quick guide to making various types of plots here). Make sure to label your axes.

```
# insert your code here
x = 5
sin(x)
```

```
-0.9589242746631385
```

Commit and Push Your Changes to GitHub

After completing the previous two exercises, commit your solution file (notebook or otherwise) and push to GitHub. Use an informative commit message which makes it clear what changes you've made. The specific workflow for this will vary depending on how you're writing up your solutions; please search for specifics and ask for help as needed!

Useful Commit Sizes

Ideally, you'd commit whenever you make a "substantial" enough change that you want to lock in, such as writing the core code for a problem or completing a problem, if you're preparing code to be used elsewhere (by yourself or others), or if you want to ask for help. `git` lets you revert changes back to a previous commit, so it's easy to undo changes or updates which broke something that was previously working, so changing too many things at once can make it hard to keep track of what worked when.

But in this case, go ahead and just commit after finishing the problems.

Push the repository with these commits to GitHub and take a screenshot of the repository page (`https://github.com/BEE4750-FA25/<username>/lab01`) which shows the updated repository. Include that screenshot in your submission as the solution to this problem.

Submitting PDF

Important

These submission instructions will not be repeated on future assignments!

Export your writeup as a PDF (this doesn't have to be a notebook if you're struggling to export: you can do your writeup in *e.g.* Microsoft Word) and submit it to the "Lab 1" assignment on Gradescope. **Make sure that you tag pages corresponding to relevant problems to avoid a 10% penalty**.

Printing Code to PDF

You are not required to submit your code when submitting assignments. However, when printing a notebook to PDF (whether you do this locally or via the GitHub Action), long lines will run off the edge of code cells, which may result in comments or code being hidden. If you see this, go back to the notebook and break up long lines into shorter onces (for example, see the comment in the above code cell) to ensure key parts of your results aren't missing.