The power of efficiency

I like the concept of efficiency a lot - probably because I am lazy. The benefits go beyond time saving though, it makes code much more robust to changes and it makes it much easier to transplant code into another project with minimal editing. Additonally, I have found the more efficient code makes it easier to get up to speed on what the code is doing when revisiting an old piece of code or reading someone else’s. My most significant encounter with efficiency this quarter was when I was revising my lab 4 assignment. The first issue I had to solve related to efficiency was to reduce the number of objects I was creating. My original code filled the environment with code every step along the way. Reducing the number of objects by writing over the object for each step severely reduced the amount of objects I had. This had the effect of making the code much easier to follow and reduced the number of times I would do a step such as filter. I was naming a lot objects with the suffix clean. Many of the objects would use the previous object with that suffix and add one step, but each time it was using the same dataset. By writing over the object when I made additional changes it was easier to track that it was the same data set the whole time, just slightly modified with each rewriting. Another revision I made to lab 4 to make it more efficient was for problem 4. In my first submission I had created to data sets containing two mutations I needed. In my revised lab I used a pivot step which allowed my to do the mutate function to one data set. The revised code was much shorter, and reduced the number of filter steps I was previously doing to two data set by only doing it to one, and I think it made it much easier to follow for a reader. After this revision I was very enthusiastic about reducing the number of repetitions in the rest of my code.