# Homework 1

**The following notebook is meant to help you work through Problems 1, 2, and 3 on Homework 1. You are by no means required to use it, nor are you required to fill out/use any of the boilerplate code/functions. You are welcome to implement the functions however you wish.**

[1]:
```
# Loading data; no changes needed.

import numpy as np
import matplotlib.pyplot as plt
from cmath import exp
train_data = np.genfromtxt("data/earth_temperature_sampled_train.csv", delimiter = ',')
year_train = train_data[:, 0] / 1000
temp_train = train_data[:, 1]
test_data = np.genfromtxt("data/earth_temperature_sampled_test.csv", delimiter = ',')
year_test = test_data[:, 0] / 1000
temp_test = test_data[:, 1]
```

[2]:
```
# These are test functions for your implementations in Questions 2.1 and 3.1; no changes needed.

def test_p2(kernel_regressor):
    """
    Run this only after you have implemented the function compute_loss, which returns the loss for a tau.
    Note that the test cases this Autograder uses are distinct from the tau values specified in the homework.
    """

    tau1, y1 = 3, [1.57, -5.04, -4.98924104, -2.74994907, -8.88]
    tau2, y2 = 90, [ 1.228045, -5.16138536, -4.16187715, -2.83729799, -8.31847509]
    tau3, y3 = 2700, [-3.46763865, -5.71861367, -4.87566622, -5.56020686, -5.18940151]

    test_pts = np.array([400, 500, 600, 700, 800])
    train_data = np.genfromtxt("data/earth_temperature_sampled_train.csv", delimiter = ',')[1:]
    year_train = train_data[:, 0] / 1000
    temp_train = train_data[:, 1]

    for tau, y in zip([tau1, tau2, tau3], [y1, y2, y3]):
        assert np.allclose(y, kernel_regressor(test_pts, tau, year_train, temp_train)), f"Failed for tau={tau}"

    print("Passed")

def test_p3(predict_knn):
    """
    Run this only after you have implemented the functions predict_kernel and predict_knn.
    """
    k1, y1 = 1, [1.57, -5.04, -4.99, -2.75, -8.88]
    k2, y2 = 3, [0.37333333, -5.19, -4.13, -2.63333333, -4.5]
    k3, y3 = 55, [-5.22981818, -5.22981818, -5.22981818, -5.41981818, -5.41981818]


    test_pts = np.array([400, 500, 600, 700, 800])
```