1 Python 101: Homework

1.1 By Evelyn J. Boettcher

1.2 Week 3 Lesson 3: Solving Equations

1.2.1 Problem 1

Based on General Linear Least Squares from chapter 15 of Numerical˙recipes˙chap15.pdf Numerical Recipes in C with focus on Solution by Use of Singular Value Decomposition

HW: For the svd fit example, re-write the the function to allow for any n degree polynomial fit.

e.g.

```
def svg_fit(x,y,n):
   Your Code here
   return a_fit
```

SVD Fit Example:

```
def svd_fit_quad(x, y):
    """
     perform a linear regression using svd
    y = a_0 + a_1 * x + a_2 * x**2
    decompose the design matrix (A) from 15.4.4 of Numerical Recipes in C

    Inputs:
        x: Numpy array or pandas series
        y: Numpy array or pandas series

    Returns:
        a_0, a_1 and a_2
    """
    design_matrix = np.array([np.ones_like(x), x, x ** 2]).T  # len(x) x 3
    a_fit = np.zeros(3)  # We know we want a length 3
    u, s, vh = np.linalg.svd(design_matrix, full_matrices=False)
    for i in range(3):
        a_fit += u[:, i].dot(y) / s[i] * vh[i, :]  # 15.4.17
    print(a_fit)
    y_fit = design_matrix @ a_fit
    plt.figure()
    plt.plot(x, y, label="real")
    plt.plot(x, y_fit, label="fit")
    plt.legend()
    plt.grid()
    plt.show()
    return a_fit



if __name__ == '__main__':
    df = pd.read_csv('../data/co2_weekly_mlo.txt', skiprows=49,
                     names=['yr', 'mon', 'day', 'decimal', 'ppm', ' #days', '1 yr ago', '10 yr ago', 's
                     delim_whitespace=True)
    clean_df = df[df.ppm != -999.99]
    # clean_df.plot('decimal', 'ppm')
    afit = svd_fit_quad(clean_df['decimal'], clean_df['ppm'])
```

Solution

1.3 Problem 1

```
def svd_fit_quad(x, y, M):
    """
    Add Doc string here
    """
    if M < 2 :
        print("Nothing really to fit, please use a n > 1")
        return
    if type(M) != int:
        print('n needs to be an integer')
        return
    # Trick here is the design_matrix.
    design_matrix = np.ones_like(x)
    for ii in range(1, M):
        design_matrix = np.vstack([design_matrix, x**ii])
    design_matrix = design_matrix.T
    a_fit = np.zeros(M)  # We know we want a length n
    u, s, vh = np.linalg.svd(design_matrix, full_matrices=False)
    for i in range(M):
        a_fit += u[:, i].dot(y) / s[i] * vh[i, :]  # 15.4.17
    print(a_fit)
    y_fit = design_matrix @ a_fit
    plt.figure()
    plt.plot(x, y, label="real")
    plt.plot(x, y_fit, label="fit")
    plt.legend()
    plt.grid()
    plt.show()
    return a_fit



if __name__ == '__main__':
    df = pd.read_csv('../data/co2_weekly_mlo.txt', skiprows=49,
                     names=['yr', 'mon', 'day', 'decimal', 'ppm', ' #days', '1 yr ago', '10 yr ago', 's
                     delim_whitespace=True)
    clean_df = df[df.ppm != -999.99]
    # clean_df.plot('decimal', 'ppm')
    afit = svd_fit_quad(clean_df['decimal'], clean_df['ppm'])
```