Homework Set #4 Problem 1 (Costa Huang)

Exponential Regrassion

We need to fit an exponential curve to the data for the years 1790 through 1900 and submit a graph of actual population and the predicted population. Let's first input the data

```
import numpy as np

# Data from year 1790 through 1900
x1 = np.array([
    0,10,20,30,40,50,
    60,70,80,90,100,110
])
y1 = np.array([
    3.929,5.308,7.240,9.638,12.866,17.069,
    23.192,31.443,38.558,50.156,62.948,76.094
])
```

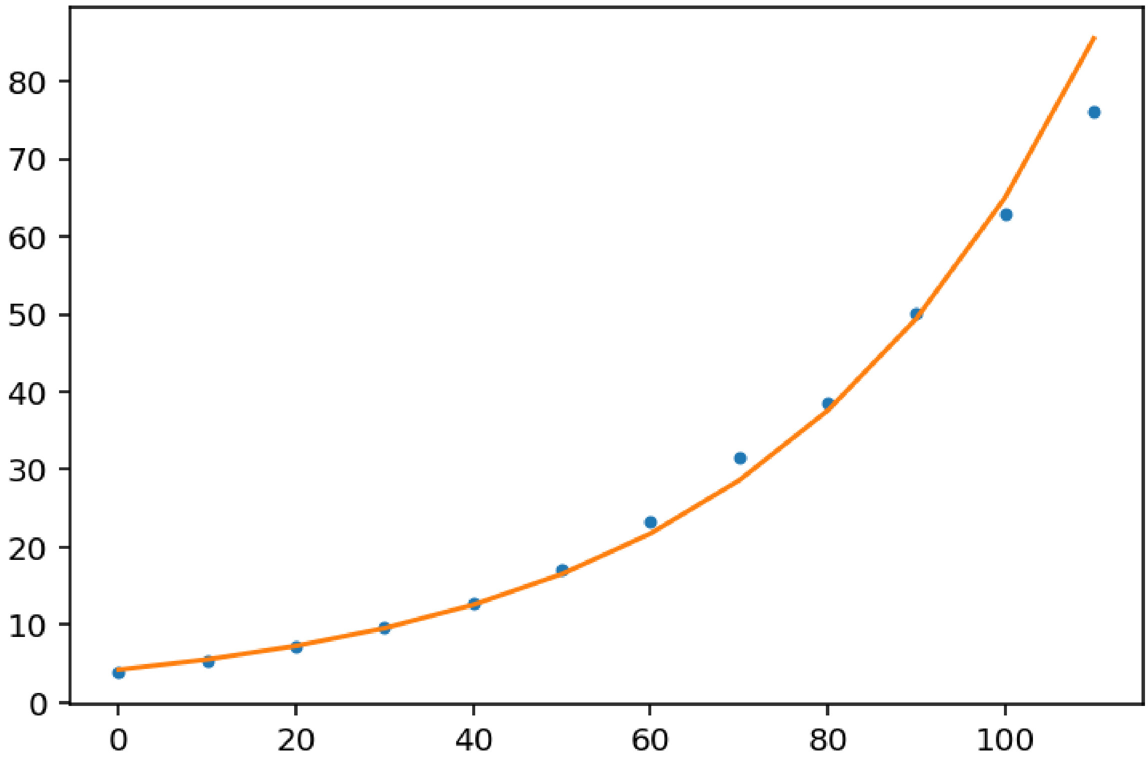
We want to use an exponential function $y = ae^{bx}$ to fit the curve. Notice it is necessarily true that

$$\ln y = \ln a + bx$$

which means $\ln y$ and x have a linear relationship. We could use the method of least square to calculate b and $\ln a$.

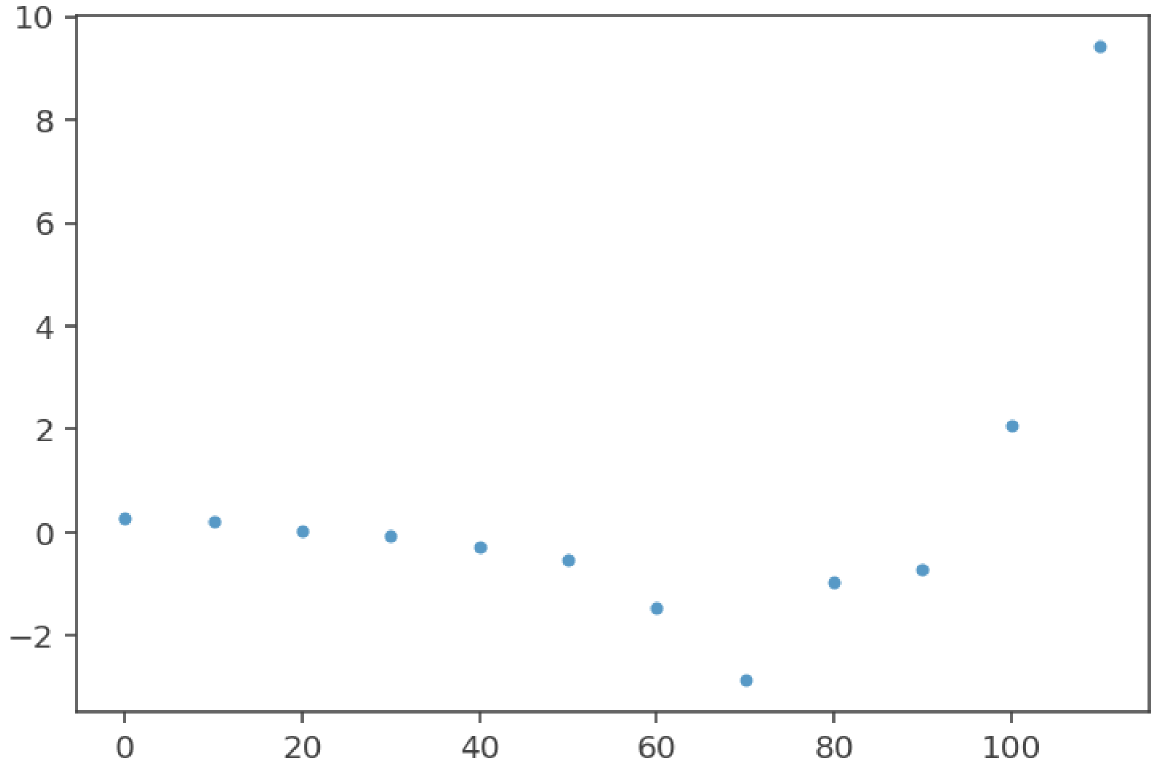
```
# np.polyfit(x, y, deg) returns a vector of coefficients
# that minimises the squared error.

%config InlineBackend.figure_format = 'svg'
import math
import matplotlib.pyplot as plt

b, lna = np.polyfit(x1, np.log(y1), 1)
a = math.exp(lna)
predicted_y1 = a * np.exp(b * x1)

plt.plot(x1, y1, '.')
plt.plot(x1, predicted_y1, '-')
```

```
[<matplotlib.lines.Line2D at 0x1e273487710>]
```




```
# Residual plot
plt.plot(x1, predicted_y1 - y1, '.')
```

```
[<matplotlib.lines.Line2D at 0x1e2735ebda0>]
```



Logistic Regression

We need to fit a logistic curve to the data for the years 1790 through 2010 and submit a graph of actual population and the predicted population. Let's first input the data

```
x2 = np.array([
    0,10,20,30,40,50,
    60,70,80,90,100,110,
    120,130,140,150,160,170,
    180,190,200,210,220
])
y2 = np.array([
    3.929,5.308,7.240,9.638,12.866,17.069,
    23.192,31.443,38.558,50.156,62.948,76.094,
    92.407,106.461,123.077,132.122,152.271,180.671,
    205.052,227.225,249.464,282.125,308.745
])
```

We want to use a logistic function

$$y = \frac{L}{1 + e^{a+bx}}$$

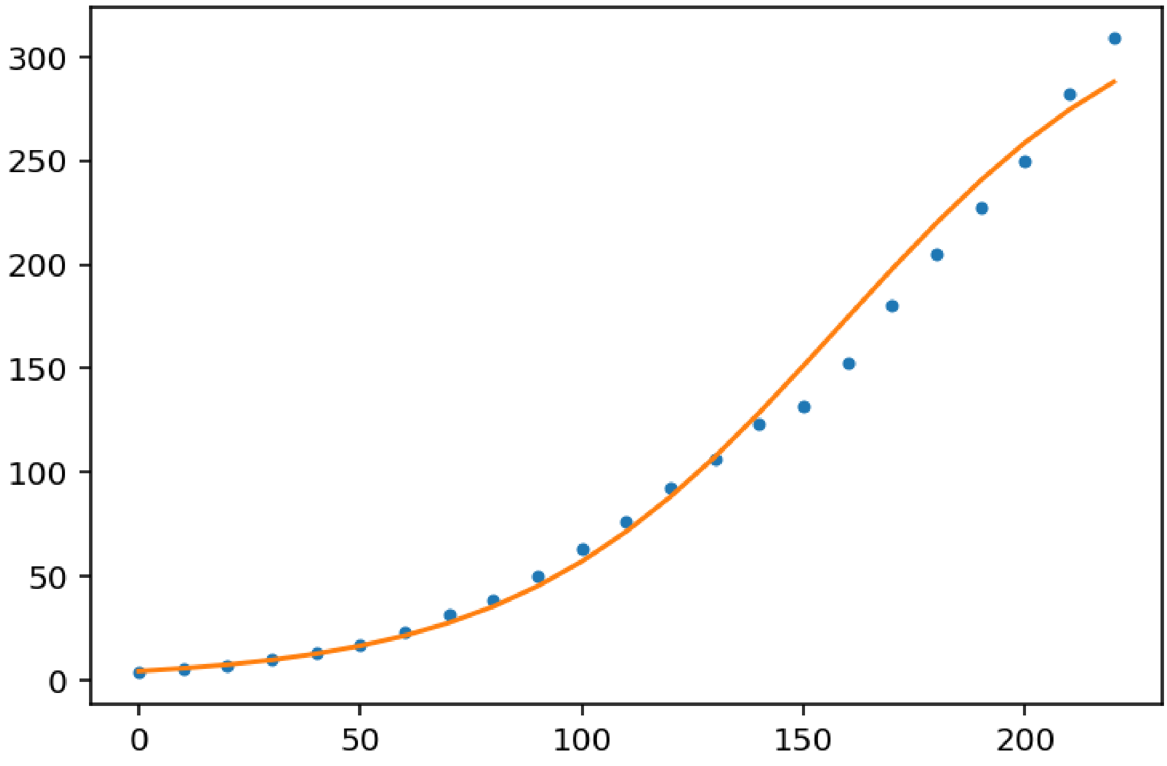
to model the data. Notice that, as the book points out, it is necessarily true that

$$\ln\left(\frac{L-y}{y}\right) = a + bx$$

which means, $\ln\left(\frac{L-y}{y}\right)$ is linear with x. In this problem, we assume the limit L is 340. Now we can find the values of a and b by similar process.

```
L = 340
b, a = np.polyfit(x2, np.log((L - y2) / y2), 1)
predicted_y2 = L / (1 + np.exp(a + b * x2))

plt.plot(x2, y2, '.')
plt.plot(x2, predicted_y2, '-')
```

```
[<matplotlib.lines.Line2D at 0x1e2734d7780>]
```




```
# Residual plot
plt.plot(x2, predicted_y2 - y2, '.')
```

```
[<matplotlib.lines.Line2D at 0x1e273713d68>]
```