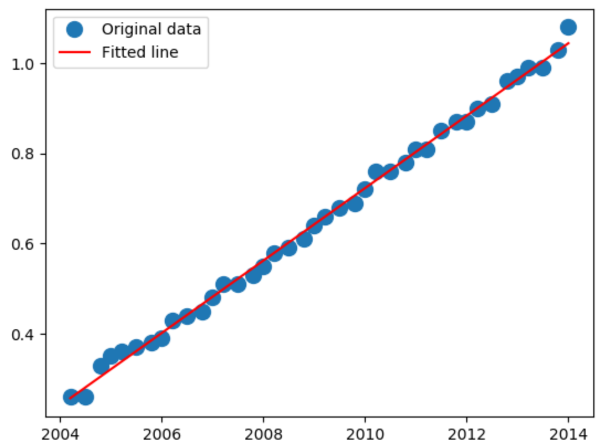
Exercise 3.1

Code:

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# plus time and loan index in csv file
# import student_debt.csv
df = pd.read_csv("./student_debt.csv")
print(df)

# change data frame to array
x = df["year"].to_numpy()
print(x)
y = df["loan"].to_numpy()
print(y)

# write the equation and plot it
A = np.vstack([x, np.ones(len(x))]).T
m, c = np.linalg.lstsq(A, y, rcond=None)[0]
_ = plt.plot(x, y, 'o', label='Original data', markersize=10)
_ = plt.plot(x, m*x + c, 'r', label='Fitted line')
_ = plt.legend()

# predict the student debt in 2050
y_2050 = m*2050 + c
print(y_2050)

plt.show()
```

Plot:



From above code, the student debt in 2050 is 3.93601 trillion of dollars.

EE 475 HW 2

Name: Zhisheng Lin
student #: 3126299

Exercise 3.3

(a) $g(\tilde{w}) = \sum_{p=1}^{P} (\tilde{x}_p^T \tilde{w} - y_p)^2 = \sum_{p=1}^{P} \left(\tilde{w}^T \tilde{x}_p \tilde{x}_p^T \tilde{w} - 2 y_p \tilde{x}_p \tilde{w} + \|y_p\|^2\right)$

$\therefore Q = 2(\tilde{x}_1 \tilde{x}_1^T + \cdots + \tilde{x}_p \tilde{x}_p^T) = 2\sum_{p=1}^{P} \tilde{x}_p \tilde{x}_p^T$

$r = 2(\tilde{x}_1 y_1 + \cdots + \tilde{x}_p y_p) = 2\sum_{p=1}^{P} \tilde{x}_p y_p$

$d = \|y_1\|^2 + \cdots + \|y_p\|^2 = \sum_{p=1}^{P} \|y_p\|^2$

(b) $Q = EDE^T = \sum_{n=1}^{N} e_n e_n^T d_n$

$\because \alpha_i^T Q \alpha_i \geq (\alpha_i^T x_i)(\alpha_i^T x_i)^T \geq 0$ where α_i is eigenvectors

$\therefore \lambda_i \alpha_i^T \alpha_i \geq 0$ where λ_i is eigenvalues

$\therefore \lambda \geq 0$

So, Q has all nonnegative eigenvalues

(c) $\because \frac{\partial}{\partial w_j} g(w) = \frac{1}{2}\left(\sum_{n=1}^{N} w_n Q_{nj} + \sum_{n=1}^{N} Q_{jn} w_n\right) + r_j$

$\therefore \nabla g(w) = Qw + r$

$\because \frac{\partial}{\partial w_i \partial w_j} g(w) = \frac{1}{2}(Q_{ij} + Q_{ji})$

$\therefore \nabla^2 g(w) = \frac{1}{2}(Q + Q^T) = Q$

Since Q has all nonnegative eigenvalues, $\nabla^2 g(\tilde{w}) > 0$, $g(\tilde{w})$ is convex.

(d) $\nabla^2 g(w^{k-1}) w^k = \nabla^2 g(w^{k-1}) w^{k-1} - \nabla g(w^{k-1})$

single step: $\nabla^2 g(w^0) w = \nabla^2 g(w^0) w^0 - \nabla g(w^0)$

$Qw = Q - Q + r$

$Qw = r$

$\left(\sum_{p=1}^{P} \tilde{x}_p \tilde{x}_p^T\right) w = \sum_{p=1}^{P} \tilde{x}_p y_p$

Exercise 3.10

(a) $\sigma(t) = \frac{1}{1+e^{-t}}$

assume $\sigma(t) = y$, $t = x$, then $y = \frac{1}{1+e^{-x}}$

$\therefore$ $x = \log\left(\frac{y}{1-y}\right)$

$\therefore$ $\sigma^{-1}(t) = \log\left(\frac{t}{1-t}\right)$, $0 < t < 1$

$\therefore$ $\sigma^{-1}(\sigma(t)) = \log\left(\frac{\frac{1}{1+e^{-t}}}{1-\frac{1}{1+e^{-t}}}\right) = \log e^{t} = t$

(b) $\sigma(b + x_p w) \approx y_p$

Since the sigmoid inverse is applied to the system

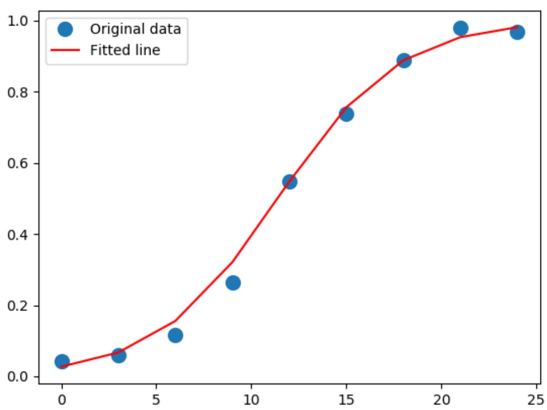
$b + x_p w \approx \sigma^{-1}(y_p) = \log\left(\frac{y_p}{1-y_p}\right)$

Exercise 3.10 (c)

Code:

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# plus x and y index in csv file
# import bacteria_data.csv
df = pd.read_csv("./bacteria_data.csv")
print(df)

# change data frame to array
x = df["x"].to_numpy()
print(x)
y = df["y"].to_numpy()
print(y)

# add 1 colum of ones
X = np.vstack([x, np.ones(len(x))]).T
print(X)

# get Y
Y = np.log(y/(1-y))
print(Y)

# get the inverse of xp*xpT and xpyp and omega
xpxp_inv = np.linalg.inv(X.T.dot(X))
print(xpxp_inv)
xpyp = X.T.dot(Y)
print(xpyp)
omega = xpxp_inv.dot(xpyp.T)
print(omega)

# get b and w from omega
b = omega[1]
w = omega[0]

# write the equation and plot it
_ = plt.plot(x, y, 'o', label='Original data', markersize=10)
_ = plt.plot(x, 1/(1+np.exp(- w*x - b)), 'r', label='Fitted line')
_ = plt.legend()
plt.show()
```

Plot:



Exercise 3.11

(a) $g(b, w) = \sum_{p=1}^{P} (\sigma(b + x_p^T w) - y_p)^2 = \sum_{p=1}^{P} (\sigma(\tilde{x}_p^T \tilde{w}) - y_p)^2$

$\nabla g(\tilde{w}) = 2 \sum_{p=1}^{P} (\sigma(\tilde{x}_p^T \tilde{w}) - y_p) \cdot \nabla(\sigma(\tilde{x}_p^T \tilde{w}) - y_p)$

$\because \sigma'(t) = \sigma(t)[1 - \sigma(t)]$

$\therefore \nabla g(\tilde{w}) = 2 \sum_{p=1}^{P} (\sigma(\tilde{x}_p^T \tilde{w} - y_p)) \cdot \sigma(\tilde{x}_p^T \tilde{w})(1 - \sigma(\tilde{x}_p^T \tilde{w})) \cdot \nabla(\tilde{x}_p^T \tilde{w})$

$\therefore \nabla g(\tilde{w}) = 2 \sum_{p=1}^{P} (\sigma(\tilde{x}_p^T \tilde{w}) - y_p) \cdot \sigma(\tilde{x}_p^T \tilde{w})(1 - \sigma(\tilde{x}_p^T \tilde{w})) \tilde{x}_p$

Exercise 3.13

(a) $g(b, w) = \sum_{p=1}^{P} (\sigma(b + x_p^T w) - y_p)^2 + \lambda \|w\|_2^2$

$= \sum_{p=1}^{P} (\sigma(\tilde{x}_p^T \tilde{w}) - y_p)^2 + \lambda w^T w$

$\because \nabla(\sum_{p=1}^{P} (\sigma(\tilde{x}_p^T \tilde{w}) - y_p)^2) = 2 \sum_{p=1}^{P} (\sigma(\tilde{x}_p^T \tilde{w}) - y_p) \cdot \sigma(\tilde{x}_p^T \tilde{w})(1 - \sigma(\tilde{x}_p^T \tilde{w})) \tilde{x}_p$

$\because \nabla(\lambda w^T w) = \lambda \cdot 2w$

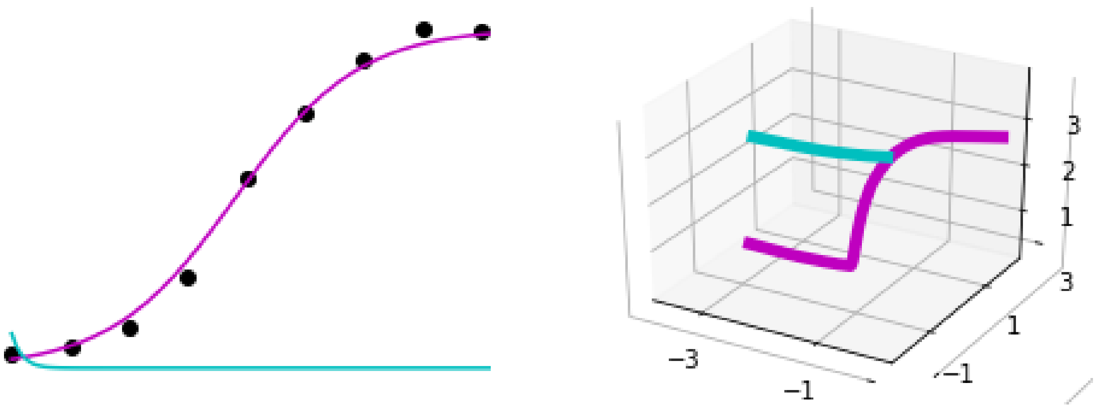
$\therefore \nabla g(w) = 2 \sum_{p=1}^{P} (\sigma(\tilde{x}_p^T \tilde{w}) - y_p) \cdot \sigma(\tilde{x}_p^T \tilde{w})(1 - \sigma(\tilde{x}_p^T \tilde{w})) \tilde{x}_p + 2\lambda \begin{bmatrix} 0 \\ w \end{bmatrix}$

Exercise 3.11 (b)

Code:

```
r = 2*(1/(1+np.exp(-X.dot(w)))-y)*(1/(1+np.exp(-X.dot(w))))*(1-(1/(1+np.exp(-X.dot(w)))))
grad = X.T.dot(r)
```

Plot:



Exercise 3.13 (b)

Code:

```
r = 2*(1/(1+np.exp(-X.dot(w)))-y)*(1/(1+np.exp(-X.dot(w))))*(1-
(1/(1+np.exp(-X.dot(w)))))
grad = X.T.dot(r)+2*lam*np.array([[0],[w[1,0]]])
```

Plot: