Problem 1

(a)

i. Coefficients generated by BGD and SGD and the hyperparameters used

Method	α_0, k, epochs	Coefficients(w_0, w_1)	Error Rate
BGD	$\alpha_0 = 0.017, k = 7$, epochs=300	(1.9417, -2.8138)	0.198
SGD	$\alpha_0 = 0.017, k = 7$, epochs=300	(1.9331, -2.8055)	0.198

Table 1: Hyperparameters and Coefficients for BGD and SGD

Here both Batch gradient descent and stochastic gradient descent were run with the same parameters. Explanation of the hyperparameters used:
- Initial learning rate α_0
- Learning rate update for SGD uses the following equation: $\alpha = \frac{k}{\text{epoch number}} \times \alpha_0$. Here I used $k = 7$ - Both BGD and SGD were run for 300 epochs, however they convereged at different points as will be shown in (ii).

ii. Comparison of the two methods in terms of Epochs taken for convergence

To infer which method converges faster we compare the Epoch point where $E_{MS} \leq 0.2$. BGD took 166 epochs while SGD took 114 epochs. Thus, SGD converged faster. Figure 1 provides a visual comparison of the fit and training error. The hyperparameters were the same as (i). Same as before, the Learning Rate for BGD stays the same as the initial learning rate (α_0) while the SGD Learning rate updates after each epoch.

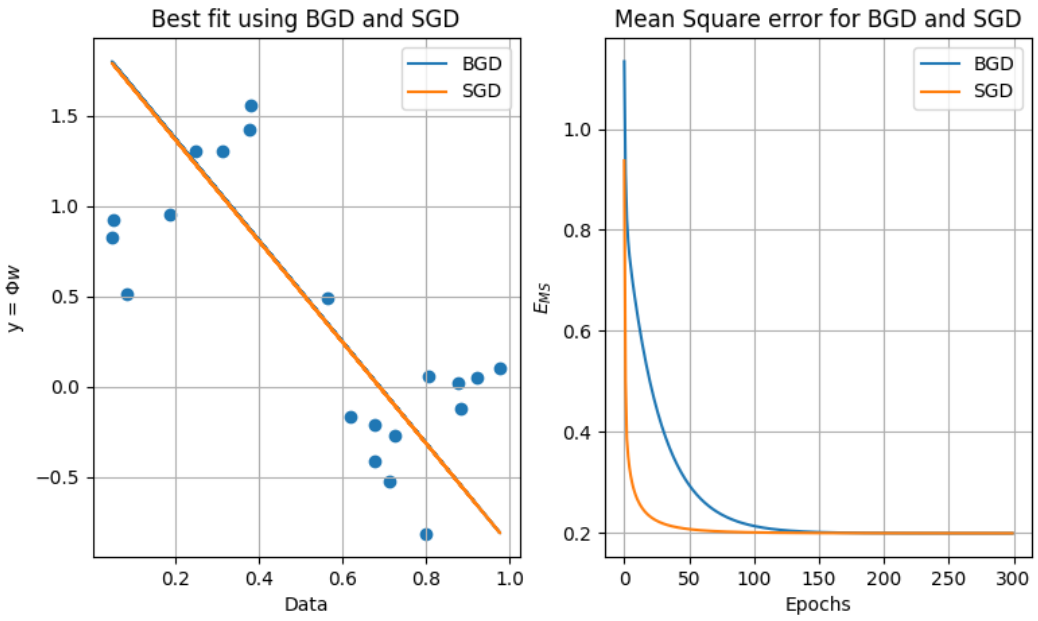

Figure 1: Linear Fit (left) and E_{MS} (right)

(b)

i. Plot for E_{RMS} vs. Degree of Polynomial

The closed form solution used is $\mathbf{w} = \Phi^{+}\mathbf{y}$

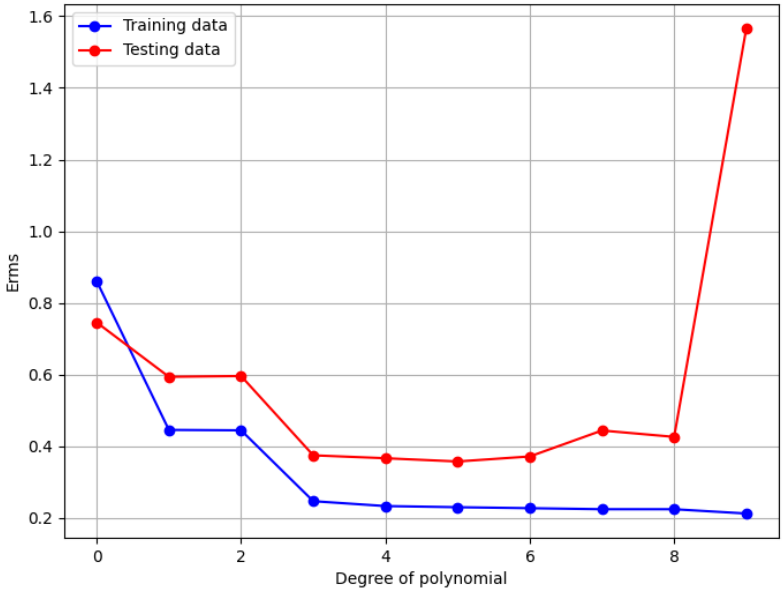

Figure 2: E_{RMS} vs. Degree of Polynomial

ii. Polynomial that fits best

From Figure 2 we can see that the Training and testing error is least at the 5^{th} (M=6) degree polynomial. Beyond that the training error reduces but the testing error increases fast. This shows that the model overfits after 6^{th} (M=7) degree polynomial. (Note: 0^{th} (M=1) degree polynomial is a constant).

(c)

i. Closed form solution for Ridge Regression

Given: $E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N} \mathbf{w}^T\Phi(\mathbf{x}^{(n)}) - y^{(n)} + \frac{\lambda}{2}\|\mathbf{w}\|^2$

$$= \frac{1}{2}\mathbf{w}^T\Phi^T(\Phi\mathbf{w} - \mathbf{y}) + \frac{1}{2}\mathbf{y}^T\mathbf{y} + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$$

Taking the gradient and setting it to 0.

$\implies \nabla_{\mathbf{w}} E(\mathbf{w}) = \Phi^T\Phi\mathbf{w} - \Phi^T\mathbf{y} + \lambda\mathbf{w} = 0$

$\implies (\lambda I + \Phi^T\Phi)\mathbf{w} - \Phi^T\mathbf{y} = 0$

$\implies \boxed{\mathbf{w}^* = (\lambda I + \Phi^T\Phi)^{-1}\Phi^T\mathbf{y}}$

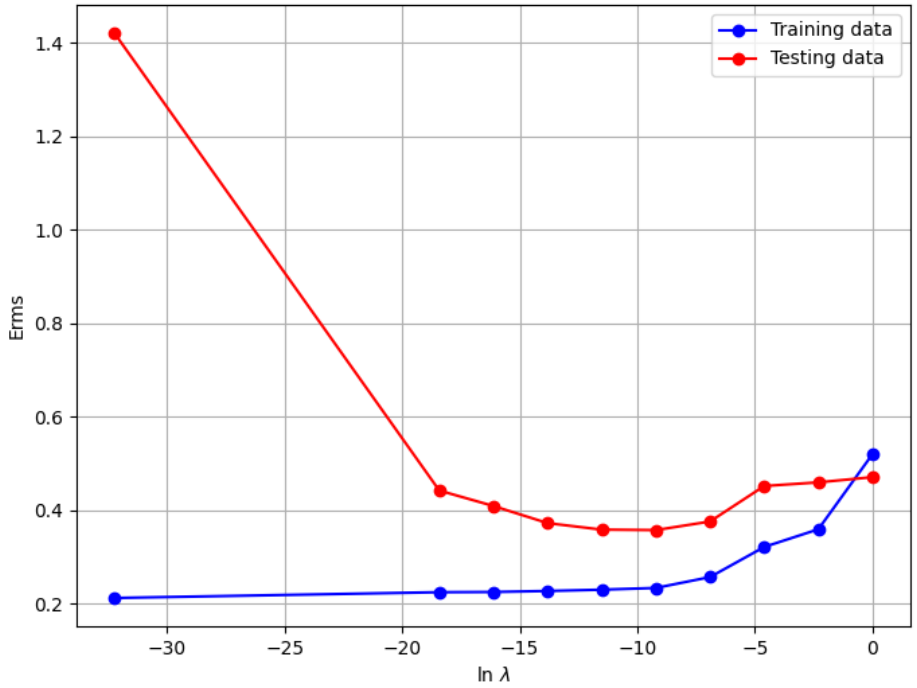

Figure 3: E_{RMS} vs. $ln\lambda$

ii. Best λ

The best λ value seems to be 10^{-4} with $ln(\lambda) = -9.2103$. This has the lowest testing error. Table 2 also summarizes the $w^{(i)}$ values for each λ. In particular the column with $\lambda = 10^{-4}$ summarizes the best **w**.

w_i^*	$\lambda = 0$	$\lambda = 10^{-8}$	$\lambda = 10^{-7}$	$\lambda = 10^{-6}$	$\lambda = 10^{-5}$	$\lambda = 10^{-4}$	$\lambda = 10^{-3}$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$
w_0^*	5.781	.8728	.908	.6778	.5911	.764	1.1478	1.615	1.719	1.229
w_1^*	-207	5.68	4.22	9.267	11.59	9.18	4.96	.248	-1.266	-.723
w_2^*	3396	-1.15	17.19	-13.1	-30.01	-22.62	-13.03	-4.18	-1.73	-.749
w_3^*	-2.7e4	-13.23	-104	-43.55	-0.612	-2.267	-3.52	-1.97	-1.01	-.49
w_4^*	1.2e5	-136	50.76	44.44	20.93	10.25	4.38	.259	-.274	-.25
w_5^*	-3.3e5	222	122	50.21	13.97	10.18	6.45	1.32	.227	-.083
w_6^*	5.6e5	140	9.433	-7.66	-3.13	4.703	4.81	1.50	.533	.0343
w_7^*	-5.6e5	-200	-119.4	-52.17	-13.13	-.877	1.64	1.217	.706	.113
w_8^*	3.1e5	-239	-99.87	-36.03	-8.934	-4.285	-1.75	.7502	.796	.165
w_9^*	-7.2e4	223	120.7	48.87	8.993	-5.148	-4.76	.254	.835	.198

Table 2: Optimal weight vector for each λ

Problem 2

(a) Derivation

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N} r^{(i)}(\mathbf{w}'\mathbf{x}^{(i)} - y^{(i)})^2$$

$$= \frac{1}{2}\sum_{i=1}^{N}(\mathbf{w}'\mathbf{x}^{(i)} - y^{(i)})r^{(i)}(\mathbf{w}'\mathbf{x}^{(i)} - y^{(i)})$$

$$= \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}'^{(i)}\mathbf{w} - y^{(i)})r^{(i)}(\mathbf{x}'^{(i)}\mathbf{w} - y^{(i)})$$

$$= (X\mathbf{w} - \mathbf{y})'R(X\mathbf{w} - \mathbf{y})$$

$$(\text{the } \frac{1}{2} \text{ gets absorbed in R})$$

(Note: I am using $'$ instead of T for transpose for the sake of convenience)

(b) Weighted closed form solution

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N} r^{(i)}(\mathbf{w}'\mathbf{x}^{(i)} - y^{(i)})^2$$

$$\implies \nabla_{\mathbf{w}} E_D(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{w}'\mathbf{x}^{(i)} - y^{(i)})r^{(i)}\mathbf{x}^{(i)}$$

Since$(\mathbf{w}'\mathbf{x}^{(i)} - y^{(i)})r^{(i)}$ is a scalar we can rewrite the above as:

$$\nabla_{\mathbf{w}} E_D(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\mathbf{x}^{(i)}(\mathbf{w}'\mathbf{x}^{(i)} - y^{(i)})r^{(i)}$$

$$= X'R(X\mathbf{w} - \mathbf{y})$$

Set the gradient to 0

$$0 = X'R(X\mathbf{w} - \mathbf{y})$$

$$\implies \boxed{\mathbf{w} = (X'RX)^{-1}X'R\mathbf{y}}$$

(c) Reducing ML estimate of w to Weighted Linear Regression

Given:

$$p(y^{(i)}|\mathbf{x}^{(i)};\mathbf{w}) = \frac{1}{\sqrt{2\pi(\sigma^{(i)})^2}}\exp\left\{\frac{-(y^{(i)}-\mathbf{w}'\mathbf{x}^{(i)})^2}{2(\sigma^{(i)})^2}\right\}$$

We can rewrite this as:

$$p(y^{(i)}|\mathbf{x}^{(i)};\mathbf{w}) = \sqrt{\frac{\beta}{2\pi}}\exp\left\{\frac{-\beta(y^{(i)}-\mathbf{w}'\mathbf{x}^{(i)})^2}{2}\right\} \quad \text{Where } \beta = \frac{1}{(\sigma^{(i)})^2}$$

Taking log on both sides

$$\log p(y^{(i)}|\mathbf{x}^{(i)};\mathbf{w}) = \sum_{i=1}^{N}\log\left(\sqrt{\frac{\beta}{2\pi}}\exp\left\{\frac{-\beta(y^{(i)}-\mathbf{w}'\mathbf{x}^{(i)})^2}{2}\right\}\right)$$

$$= \underbrace{\sum_{i=1}^{N}\frac{1}{2}\log\beta - \frac{1}{2}\log 2\pi}_{\text{Constant terms}} - \sum_{i=1}^{N}\frac{\beta}{2}((y^{(i)}-\mathbf{w}'\mathbf{x}^{(i)})^2)$$

Take the gradient and set it to 0

$$\nabla_{\mathbf{w}}\log p(y^{(i)}|\mathbf{x}^{(i)};\mathbf{w}) = \sum_{i=1}^{N}\frac{\beta}{2}(y^{(i)}-\mathbf{w}'\mathbf{x}^{(i)})\mathbf{x}^{(i)} = 0$$

This form of the equation shows that the problem has reduced to a Weighted Linear Regression.

Here we can substitute $\beta = r^{(i)}$ which gives us the following form:

$$\implies 0 = \sum_{i=1}^{N} r^{(i)}(y^{(i)}-\mathbf{w}'\mathbf{x}^{(i)})\mathbf{x}^{(i)}$$

$$= X'R(\mathbf{y}-X\mathbf{w})$$

$$\implies \boxed{\mathbf{w} = (X'RX)^{-1}X'R\mathbf{y}}$$

This is the same form as (b). This also means $\boxed{r^{(i)} = \frac{1}{(\sigma^{(i)})^2}}$

(d)

i. Unweighted Linear Regression

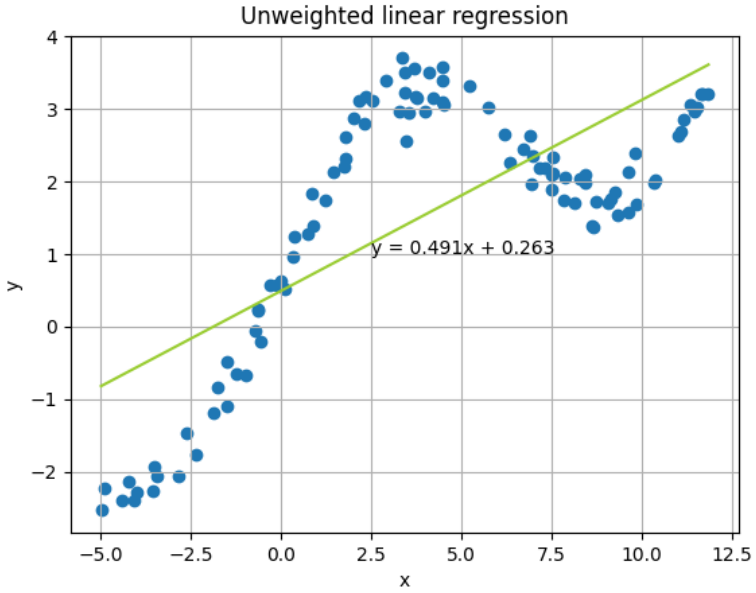

Figure 4: Unweighted Linear Regression

ii. Locally Weighted Linear Regression

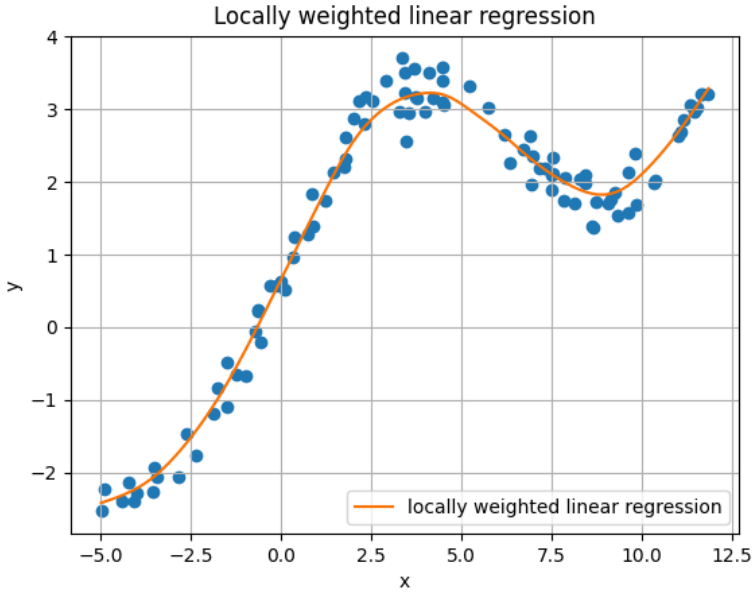

Figure 5: Locally weighted linear regression with $\tau = 0.8$

iii. Locally weighted Linear Regression with different τ

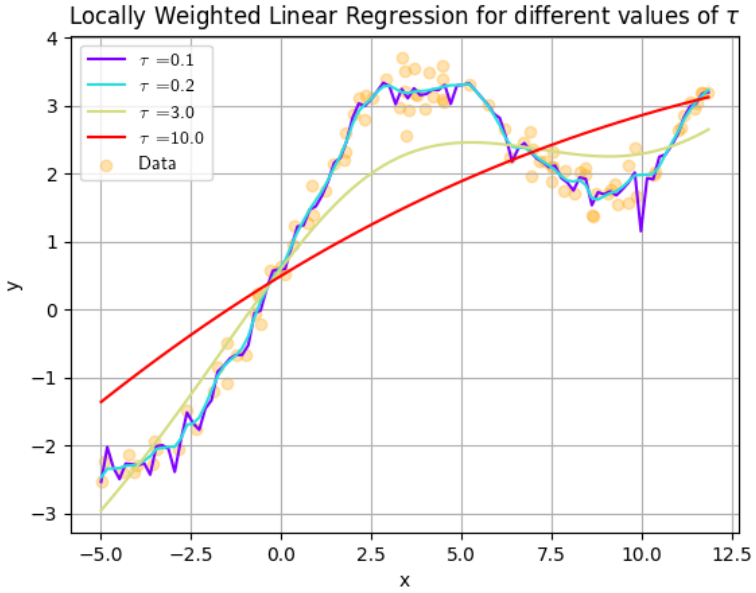

Figure 6: Locally weighted linear regression with different τ values

From the plot we can infer that if τ is too small then the curve will overfit, and if the τ is too small, then it will be too smooth and underfit. This is also apparent from the given equation of $r^{(i)} = \exp\left(\frac{-(x-x^{(i)})^2}{2\tau^2}\right)$. As $\tau \to \infty, r^{(i)} \to 1$, which is the same as linear regression and as $\tau \to 0$, weights overfit.

Problem 3

(a)

We can rewrite the given w_0, w_1 in matrix multiplication form:

$$w_1 = \frac{\frac{1}{N}\mathbf{y}'\mathbf{x} - \bar{X}\bar{Y}}{\frac{1}{N}\mathbf{x}'\mathbf{x} - \bar{X}^2} = \frac{\frac{1}{N}\mathbf{y}'\mathbf{x} - \frac{1}{N^2}(\mathbf{1}'_N\mathbf{x})(\mathbf{1}'_N\mathbf{y})}{\frac{1}{N}\mathbf{x}'\mathbf{x} - \frac{1}{N^2}(\mathbf{1}'_N\mathbf{x})^2} = \frac{N\mathbf{y}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})(\mathbf{1}'_N\mathbf{y})}{N\mathbf{x}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})^2} \tag{1}$$

$$w_0 = \frac{1}{N}\mathbf{1}'_N\mathbf{y} - w_1(\frac{1}{N}\mathbf{1}'_N\mathbf{x}) \tag{2}$$

Substituting the value of w_1 into (2) we get:

$$w_0 = \frac{1}{N}\mathbf{1}'_N\mathbf{y} - \left(\frac{N\mathbf{y}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})(\mathbf{1}'_N\mathbf{y})}{N\mathbf{x}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})^2}\right)\frac{1}{N}\mathbf{1}'_N\mathbf{x}$$

$$= \frac{1}{\cancel{N}}\left(\frac{\cancel{N}(\mathbf{1}'_N\mathbf{y})(\mathbf{1}'_N\mathbf{x}) - \cancel{(\mathbf{1}'_N\mathbf{y})(\mathbf{1}'_N\mathbf{x})^2} + \cancel{(\mathbf{1}'_N\mathbf{y})(\mathbf{1}'_N\mathbf{x})^2} - \cancel{N}(\mathbf{y}'\mathbf{x})(\mathbf{1}'_N\mathbf{x})}{N\mathbf{x}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})^2}\right)$$

$$= \frac{(\mathbf{x}'\mathbf{x})(\mathbf{1}'_N\mathbf{y}) - (\mathbf{1}'_N\mathbf{x})(\mathbf{x}'\mathbf{y})}{N\mathbf{x}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})^2} \tag{3}$$

Now we can minimize L and use the closed form solution to evaluate w_0 and w_1. Rewrite $h(x) = w_0 + w_1 x$ as $h(x) = \begin{bmatrix}\mathbf{1}_N & \mathbf{x}\end{bmatrix}\begin{bmatrix}w_0 \\ w_1\end{bmatrix} = \Phi\mathbf{w}$. This makes $L = \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \Phi(x^{(i)})\mathbf{w})$. The closed form solution is $\mathbf{w} = \Phi^{+}\mathbf{y} = (\Phi'\Phi)^{-1}\Phi'\mathbf{y}$.

$$\begin{bmatrix}w_0 \\ w_1\end{bmatrix} = \left(\begin{bmatrix}\mathbf{1}'_N \\ \mathbf{x}'\end{bmatrix}\begin{bmatrix}\mathbf{1}_N & \mathbf{x}\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{1}'_N \\ \mathbf{x}'\end{bmatrix}\mathbf{y}$$

Solving this gives us

$$\begin{bmatrix}w_0 \\ w_1\end{bmatrix} = \left(\begin{bmatrix}\mathbf{1}'_N \\ \mathbf{x}'\end{bmatrix}\begin{bmatrix}\mathbf{1}_N & \mathbf{x}\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{1}'_N \\ \mathbf{x}'\end{bmatrix}\mathbf{y}$$

$$= \frac{1}{N\mathbf{x}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})^2}\begin{bmatrix}\mathbf{x}'\mathbf{x} & -\mathbf{1}'_N\mathbf{x} \\ -\mathbf{1}'_N\mathbf{x} & N\end{bmatrix}\begin{bmatrix}\mathbf{1}'_N\mathbf{y} \\ \mathbf{x}'\mathbf{y}\end{bmatrix}$$

$$\implies w_0 = \frac{(\mathbf{x}'\mathbf{x})(\mathbf{1}'_N\mathbf{y}) - (\mathbf{1}'_N\mathbf{x})(\mathbf{x}'\mathbf{y})}{N\mathbf{x}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})^2} \qquad \text{Same as (3)}$$

$$\implies w_1 = \frac{N\mathbf{y}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})(\mathbf{1}'_N\mathbf{y})}{N\mathbf{x}'\mathbf{x} - (\mathbf{1}'_N\mathbf{x})^2} \qquad \text{Same as (1)}$$

Hence proven.

(b)

i. To prove: A is PD $\iff$ $\lambda_i > 0$ for each i

Proof of $\implies$: A has unitary eigendecomposition $A = V\Lambda V'$

$$\mathbf{x}'A\mathbf{x} = \mathbf{x}'V\Lambda V'\mathbf{x} = z'\Lambda z = \sum_i \lambda_i |z_i|^2 \qquad \text{where } Z = V'\mathbf{x}$$

$$\therefore \text{ if } A \text{ has } \lambda_i > 0 \text{ then } \mathbf{x}'Ax > 0 \quad \forall x \in \mathbb{R} \implies A \succ 0$$

Proof of $\impliedby$ assume A has an eigenvalue $\lambda_i \leq 0$ then let $x = V\mathbf{e}_i$ where $\mathbf{e}_i$ is the elementary vector with 1 at position i (e.g. $\mathbf{e}_1 = \begin{bmatrix}1 \\ 0 \\ \vdots \\ 0\end{bmatrix}$). Then

$$\mathbf{x}'A\mathbf{x} = \mathbf{e}'_i V'V\Lambda V'V\mathbf{e}_i = \lambda_i \leq 0$$

$\implies$ A cannot be PD, thus A is PD $\iff$ $\lambda_i > 0$
Q.E.D.

ii. To prove: For any $\beta > 0$ ridge regression makes the matrix $\Phi'\Phi + \beta I$ PD

Let Φ have the SVD as $U\Sigma V'$ where U and V are unitary. Then,

$$\Phi'\Phi = V\Sigma' \underbrace{U'U}_{\text{I}} \Sigma V' = V\Sigma'\Sigma V'$$

This is a valid unitary eigen-decomposition, thus an SVD for $\Phi'\Phi$. with signular values as σ_i^2. Thus,

$$\begin{aligned}\Phi'\Phi + \beta I = V\Sigma'\Sigma V' + \beta \underbrace{VIV'}_{\text{Since } VV'=I} \\ = V(\Sigma'\Sigma + \beta I)V'\end{aligned}$$

This is a valid unitary eigen decomposition thus an SVD with singular values $\sigma_i^2 + \beta \implies$ Singular values of $\Phi'\Phi$ are shifted by β. Since $\sigma_i^2 > 0$ and it is given $\beta > 0$, then $\sigma_i^2 + \beta > 0 \implies$ $\Phi'\Phi + \beta I$ is PD.
Q.E.D.