1) The object function that must be solved for linear regression in one-dimension is the following:

$$L = \min_{w_{cap}, b_{Cap}} E\left[\left(w_{cap}.X + b_{cap} - Y\right)^2\right]$$

But, since, $Y = w.x + b + \mathcal{E}$, we can rewrite the equation as:

$$L = \min_{w_{cap}, b_{Cap}} E\left[\left((w_{cap} - w)X + (b_{cap} - b) - \mathcal{E}\right)^2\right]$$

The solutions to this equation can be solved in the usual way by differentiating the function w.r.t the parameter and equating them to 0.

$$\frac{\partial L}{\partial b_{cap}} = E\left[2\left((w_{cap} - w)X + (b_{cap} - b) - Y\right)\right] = 0$$

$$\frac{\partial L}{\partial w_{cap}} = E\left[2\left((w_{cap}. - w)X + (b_{cap} - b) - Y\right).X)\right] = 0$$

Solving for the 2 equations, we get, $E\left[w_{cap}\right] = w$ and $E\left[b_{cap}\right] = b$

For calculating variance,
We know that:

$$w_{cap} = [X^T X]^{-1} X^T Y$$

where X is the augmented variable vector of length 2.

For truly linear model data we can write this as,

$$W_{cap} = [X^T X]^{-1} X^T (XW + \mathcal{E})$$

$$=> \ W_{cap} = W + [X^T X]^{-1} X^T \mathcal{E}$$

where W is the augmented linear coefficient vector of length 2.

This can be interpreted in the following way: $w_{cap}$ recovers the true value w with an error described by the right-hand term. We can express this relation as:

$$W_{cap} \sim W + N(0, ([X^T X]^{-1} X^T)([X^T X]^{-1} X^T)^T \sigma^2)$$

$$=> \ W_{cap} \sim W + N(0, [X^T X]^{-1} \, \sigma^2)$$

Since,

$$(X^T X)^{-1} = \begin{matrix} \frac{\sum x_i^2}{m \sum (x_i - E[x])^2} & \frac{-\sum x_i}{m \sum (x_i - E[x])^2} \\ \frac{\sum x_i}{m \sum (x_i - E[x])^2} & \frac{1}{\sum (x_i - E[x])^2} \end{matrix}$$

Also since, the var of the coefficient vector is equal to the var of the normal noise described in $W_{cap} \sim W + N(0, [X^T X]^{-1} \, \sigma^2)$

$$var\left(W_{Cap}\right) = \begin{matrix} var(b_{cap}) & cov(b_{cap}, w_{cap}) \\ cov(b_{cap}, w_{cap}) & var(w_{cap}) \end{matrix} = \sigma^2 (X^T X)^{-1}$$

Using this relation, we get,

$$var(b_{cap}) = \frac{\sigma^2 \sum x_i^2}{m \sum (x_i - E[x])^2}$$

$$var(w_{cap}) = \frac{\sigma^2}{m \sum (x_i - E[x])^2}$$

2) If the variance and mean of X are known from an underlying distribution, we can rewrite the above equations as:

$$var(w_{cap}) = \frac{\sigma^2}{m\, var(x)}$$

$$var(b_{cap}) = \frac{\sigma^2 E[x^2]}{m\, var(x)}$$

3) If the data is re-centered using $x_i' = x_i - \mu$,

The var(x')=var(x), therefore, var($w_{cap}$) or the error in $w_{cap}$ remains the same as before it was re-centered.

However,

$$\begin{aligned} E[x'^2] &= E[(x-\mu)^2] \\ &= E[x^2 + \mu^2 - 2x\mu] \\ &= E[x^2] - 2E[x]\mu + \mu^2 \\ &= [x^2] - \mu^2 \\ &< E[x^2] \end{aligned}$$

Therefore, since the numerator of var($b_{cap}$) reduces keeping everything else constant, the value of var($b_{cap}$) or the error in $b_{cap}$ reduces.

4) Please find the code in main.py.

Yes, the observed results agree with what was mentioned earlier. After re-centering, the error in $w_{cap}$ remains the same, but the error in $b_{cap}$ reduces from ~0.015 to ~0.00001.

5) There is no change in the slope, since the relative differences between the values of X or Y remain the same. Only the absolute value of X changes. Therefore, there is only a change in the intercept of our regression line.

6) We know that that,

$$\Sigma \rightarrow m\, E[X^T X]$$

On augmenting the feature set from 1 to 2 dimensions, X=[1 x].

We can then rewrite the above as:

$$\Sigma \rightarrow m\, E[\begin{bmatrix} 1 \\ x \end{bmatrix} [1\ x]]$$

$$\Sigma \rightarrow m\, E \begin{bmatrix} 1 & x \\ x & x^2 \end{bmatrix}$$

$$\Sigma \rightarrow m \begin{bmatrix} 1 & E[x] \\ E[x] & E[x^2] \end{bmatrix}$$

On re-centering the data, the above matrix changes to:

$$\Sigma \rightarrow m\begin{bmatrix} 1 & 0 \\ 0 & E[(x-\mu)^2] \end{bmatrix}$$

The eigenvalues of this matrix are $E[(x-\mu)^2]$ and 1. The value of $K(\Sigma)$ given by

$$K(\Sigma) = \frac{\text{Largest eigenvalue of } \Sigma}{\text{Smallest eigenvalue of } \Sigma}$$

Is always lower than the value we get using the non-centered data.