**Answer 1**

The objective function (or loss function since minimizing) is:

SE = Squared Error

This represents the ordinary least squares problem of regression.

We have to find and which minimizes the loss function (minimizes the squared error to regression line).

So, we take the partial derivative of the loss function w.r.t. and and set it equal to 0.

and

or,

or,

or,

or,

The yellow highlighted values are the mean values or the expected values, so we can rewrite as:

or,

(1)

or,

or,

or,

or,

The yellow highlighted values are the mean values or the expected values, so we can rewrite as:

or,

(2)

Now we just need the solve the two simultaneous equations (1) and (2) to find and

Rewriting equation (2)

or,

(3)

Rewriting equation (1)

Substituting in the above,

or,

or,

or,

(4)

or,

or,

or,

Using the above, rewriting equation (4)

Substituting from above in (3)

**Answer 2**

From (4)

Previously, we divided by m to get mean, let us undo that (i.e. multiply numerator and denominator by m), then:

(5)

Given true linear model

Substituting the above in (5),

or,

or,

or,

or,

or,

But the mean of errors/noise is 0(given), therefore

and because

Therefore,

This means that is an unbiased estimate of w i.e. it is correct on average.

Now we know

or,

or,

or,

(6)

From (3),

or,

or,

or,

since and expected value of an expected value is just that.

Substituting from (6)

or,

**Alternatively**, in Answer 1, we proved the following:

or,

or,

or,

or,

or,

or,

or,

But , therefore,

This means that is an unbiased estimate of w i.e. it is correct on average.

From (3),

or,

or,

or (since ),

or,

The variances of these estimators are asked to be proven in the next question, so please refer to Answer 3

**Answer 3**

**Finding Variance of :**

or,

or (given ,

or,

or,

or,

(7)

or (m-1 for sample variance; for population it would have been m),

or,

or (for large m, m-1=m),

**Finding Variance of :**

or,

or,

or,

Evaluating the above yellow highlighted term

or,

or,

or,

Therefore,

or,

or,

**Answer 4**

After centering, new

is the mean of values, hence we are performing mean centering as discussed in the lecture.

Let the new error on be and be (after centering)

From (7)

or,

or,

or,

or,

or,

Hence the new error (after centering) is same as the previous error.

From the result in Answer 3, we have

or,

or,

or,

or,

But , therefore,

or,

Proof that new error on i.e. is minimized

or,

or,

Since is a square hence it is a positive quantity, therefore the numerator is smaller than the denominator.

Hence, .

**Answer 5**

**Code:**

**Importing Libs**

**import** numpy **as** np

**Functions to Generate Xs and Y**

# gen X (m datapoints (1 Dim))

**def** gen\_X(m): #https://numpy.org/doc/stable/reference/random/generated/numpy.random.uniform.html

X=np.random.uniform(low=100,high=102,size=m)

**return** X

# gen Y/target/labels/output

**def** gen\_Y(X,w,b,variance):

Y=np.empty((np.shape(X)[0]))

**for** i **in** **range**(np.shape(Y)[0]): #https://numpy.org/doc/stable/reference/random/generated/numpy.random.normal.html

#scale takes std. dev. hence we have to sqrt the variance

Y[i]=X[i]\*w + b + np.random.normal(loc=0, scale=**pow**(variance,0.5))

**return** Y

# gen new X i.e. X' (m datapoints (1 Dim))

**def** gen\_X\_dash(X):

X\_dash=np.empty((np.shape(X)[0]))

**for** i **in** **range**(np.shape(X\_dash)[0]):

X\_dash[i] = X[i] - 101

**return** X\_dash

**Helper functions**

**def** compute\_w\_and\_b(X, Y):

#https://numpy.org/doc/stable/reference/generated/numpy.cov.html

#https://stackoverflow.com/questions/15317822/calculating-covariance-with-python-and-numpy

#w\_hat=Cov(x,y)/Var(x) ... proven in Answer 1

w=(np.cov(X, Y, bias=True)[0][1])/np.var(X)

#b\_hat=y\_mean-w\_hat\*x\_mean ... equation (3)

b=np.mean(Y)-w\*np.mean(X)

**return** w, b

**def** sim(m, w, b, variance, iters=1000):

w\_hat\_list = []

b\_hat\_list = []

w\_hat\_dash\_list = []

b\_hat\_dash\_list = []

**for** i **in** **range**(iters):

X=gen\_X(m)

Y=gen\_Y(X,w,b,variance)

X\_dash=gen\_X\_dash(X)

w\_hat, b\_hat = compute\_w\_and\_b(X, Y)

w\_hat\_list.append(w\_hat)

b\_hat\_list.append(b\_hat)

w\_hat\_dash, b\_hat\_dash = compute\_w\_and\_b(X\_dash, Y)

w\_hat\_dash\_list.append(w\_hat\_dash)

b\_hat\_dash\_list.append(b\_hat\_dash)

**print**("Expected value of w\_hat: ",np.mean(w\_hat\_list))

**print**("Expected value of w\_hat\_dash: ",np.mean(w\_hat\_dash\_list))

**print**("Expected value of b\_hat: ",np.mean(b\_hat\_list))

**print**("Expected value of b\_hat\_dash: ",np.mean(b\_hat\_dash\_list))

**print**("Variance of w\_hat: ",np.var(w\_hat\_list))

**print**("Variance of w\_hat\_dash: ",np.var(w\_hat\_dash\_list))

**print**("Variance of b\_hat: ",np.var(b\_hat\_list))

**print**("Variance of b\_hat\_dash: ",np.var(b\_hat\_dash\_list))

**Running the sim**

m=200

w=1

b=5

variance=0.1

sim(m, w, b, variance, 1000)

**Output:**

Expected value of w\_hat: 0.9993276553126749

Expected value of w\_hat\_dash: 0.999327655312675

Expected value of b\_hat: 5.067982014234828

Expected value of b\_hat\_dash: 106.00007520081499

Variance of w\_hat: 0.0014741891806137291

Variance of w\_hat\_dash: 0.0014741891806137278

Variance of b\_hat: 15.034748195851964

Variance of b\_hat\_dash: 0.0004470032298959514

The results agree.

First, we are generating one dimensional data (**X**) uniformly distributed between 100 and 102.

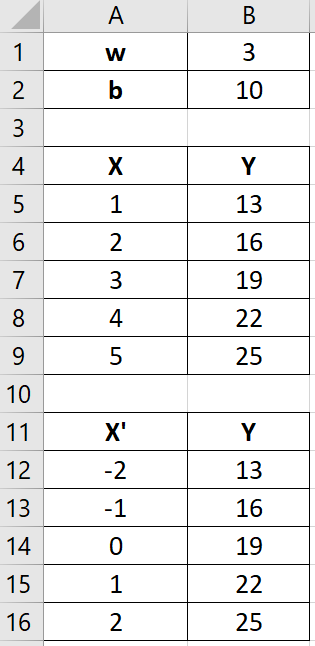
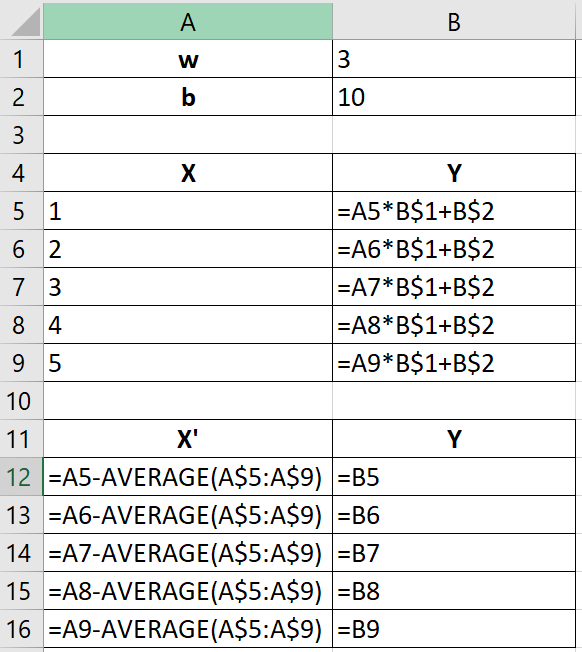
Then, we are generating the regression line (**Y**) based on **X**.

Then, we are generating **X’ (X\_dash)** i.e. X centered around the mean of X.

As, we have shown in Answer 4, that centering the data around the mean, produces the same error on the weights but reduces/minimizes the error on the bias. This is confirmed from the above simulation results. The variance of weights on non-centered data and centered data is almost the same (difference only in higher precision). The variance of bias on non-centered data and centered data is different. The error on bias is much reduced on the centered data.

**Answer 6**

Consider the following dataset (different w and b values from question 5):

And the resultant plots with trendline:

After generating the trendline in excel, we see that the coefficient of **x** in the equation of the line remains same even after centering the data around the mean.

This is because all the points are just re positioned by the same value. However, the bias (y-intercept) changes.

Here there is no error term but even if there was, then on average the slope (w) would remain the same. Hence the estimate or mean of the slope has no change when the data is shifted.

We can intuitively say that we are just grabbing the dataset and repositioning the whole thing to our wish (w will remain the same if centered around a value other than the mean as well). Hence the slope doesn’t change.

This is true for higher dimensional data as well. The structure remains the same, only the position of the whole thing is changed.

**Answer 7**

After augmenting (going from one dimension to two dimensions), the new X would be

Consequently,

Now,

or,

or,

or,

or,

To find the condition number of i.e. , we need to find the find the eigenvalues of (largest and smallest eigenvalue). *Later we will find to compare both.*

or,

or,

or,

or,

We have a quadratic equation in terms of , solving which will give us two eigenvalues of the 2\*2 matrix .

*We will consider the ‘+’ term as the largest eigenvalue and the ‘-’ term as the smallest eigenvalue.*

or,

References

1. Lecture Notes and Videos
2. <https://www.khanacademy.org/math/statistics-probability/describing-relationships-quantitative-data/more-on-regression/v/proof-part-4-minimizing-squared-error-to-regression-line>
3. <https://en.wikipedia.org/wiki/Covariance>
4. <https://math.stackexchange.com/questions/2947788/what-is-varx-y>
5. <http://www.unm.edu/~lspear/geog525/24linreg2.pdf>
6. <http://home.iitk.ac.in/~shalab/econometrics/Chapter2-Econometrics-SimpleLinearRegressionAnalysis.pdf>

\*\*\*