Question 1 : Linear Regression

We will fit linear regression models to the data in file `regression_part1.csv`.

(a) Describe the main properties of the data, focusing on the size, data ranges, and data types.

The data consist of 50 rows and 2 columns, where the 1st column is the data which represents the ***revision time*** taken for the exam and 2nd column represents the ***exam score*** received. Thus, the data contains 50 individuals' ***revision time*** and ***exam score*** and both data types are numerical.
By using the function `describe()`, $\mu_1 = 22.22$ & $\sigma_1 = 13.97$ and $\mu_2 = 49.20$ & $\sigma_2 = 20.93$ where $(\mu_1, \sigma_1), (\mu_2, \sigma_2)$ are mean and standard deviation for ***revision time*** and ***exam score*** respectively

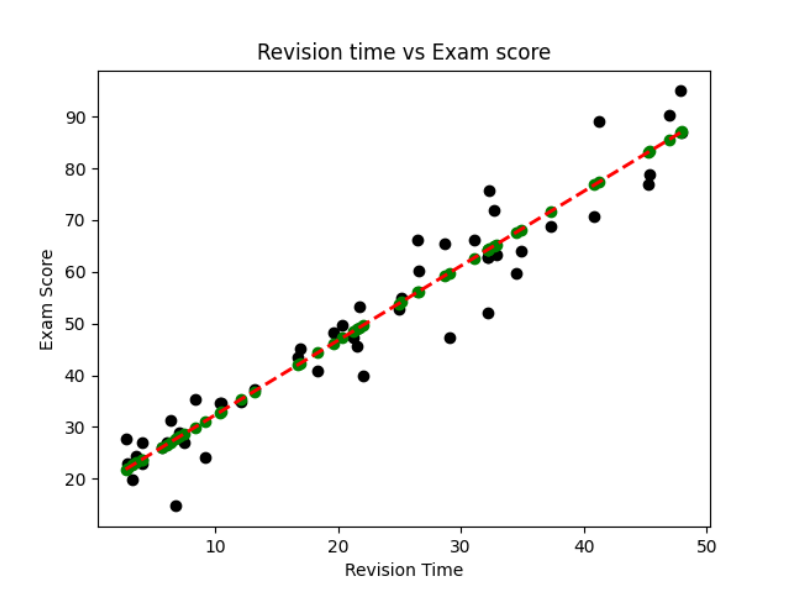
(b) Fit a linear model to the data so that we can predict `exam_score` from `revision_time`. Report the estimated model parameters $\mathbf{w}$. Describe what the parameters represent for this 1D data. For this part, you should use the sklearn implementation of Linear Regression.
Hint: By default in sklearn `fit_intercept = True`*. Instead, set* `fit_intercept = False` *and pre-pend* 1 *to each value of* x_i *yourself to create* $\boldsymbol{\phi}(x_i) = [1, x_i]$.

We have $y = w_0 + w_1 X$ where matrix X is 50×1 and $\mathbf{w} = [w_0, w_1]^T$ is 2×1. We are adding a column of 1 to ensure the dot product between matrices. Thus we can expresses y_i as $y_i = [1, x_i] \cdot [w_0, w_1]^T = \boldsymbol{\phi}(x_i) \cdot [w_0, w_1]$.

From the relationship of the equation we can derive in a way such that w_0 represents the intercept of the linear fit and also known as the bias term. w_1 represents the gradient of the linear fit.

(c) Display the fitted linear model and the input data on the same plot.

A fitted linear model with the scatter plot



After pre-pending 1 to each value of x_i and fitting a linear model to the data, the function `'lm.coef_'` printed the parameter values of 17.898 and 1.441 for w_0 & w_1 respectively.

As we can see from the graph, the fitted line represents the data well with no significant outlier. Black points represent the true value of ***Exam Score*** and the Green points represent the predicted value of ***Exam Score***. The red line is the linear regression being fitted on the ***Revision Time***.

(d) Instead of using sklearn, implement the closed-form solution for fitting a linear regression model yourself using numpy array operations. Report your code in the answer box. It should only take a few lines (i.e. <5).
Hint: Only report the relevant lines for estimating **w** *e.g. we do not need to see the data loading code. You can write the code in the answer box directly or paste in an image of it.*

As mentioned above, we denoted that $y = w_0 + w_1 X$ and we want to estimate $\boldsymbol{w} = [w_0, w_1]$ by implementing closed-form solution. Therefore, we can compute $\boldsymbol{w} = (X^T \cdot X)^{-1} \cdot X^T \cdot y$, where $(X^T \cdot X)^{-1} \cdot X^T$ is the pseudo-inverse of X

Then, we applied this formula using the numpy array operations shown below,

```
import numpy as np
from numpy.linalg import inv

w = inv(X.T.dot(X)).dot(X.T).dot(y)
```

Since we have the 1 by 2 matrix, X. The dimension of $\boldsymbol{w}$ is 2 by 1 matrix with the values of $[17.89768026, 1.44114091]$ respectively.

(e) Mean Squared Error (MSE) is a common metric used for evaluating the performance of regression models. Write out the expression for MSE and list one of its limitations.
Hint: For notation, you can use y for the ground truth quantity and $\hat{y}$ (`$\hat{y}$` in latex) in place of the model prediction.

$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \phi\hat{\boldsymbol{w}})^2$, where $\hat{y} = \phi\hat{\boldsymbol{w}}$
One limitation of Mean Squared Error is that it is very sensitive towards outliers. When there is a single outlier with a large error in a model, the value of the MSE can be much greater than those models with more outliers but with small errors. However, looking at the plot in the previous question, we can see that there is no significant outlier and therefore, MSE is a good metric to use for the evaluation of regression model in our case.

(f) Our next step will be to evaluate the performance of the fitted models using Mean Squared Error (MSE). Report the MSE of the data in `regression_part1.csv` for your prediction of `exam_score`. You should report the MSE for the linear model fitted using sklearn and the model resulting from your closed-form solution. Comment on any differences in their performance.

We acquired $\boldsymbol{w} = (X^T \cdot X)^{-1} \cdot X^T \cdot y$ in the closed-form solution. Then we know $\hat{y} = X \cdot \boldsymbol{w}$. Then MSE value of 30.985 was resulted.

Using `sklearn`, we employed `sklearn.metrics.mean_squared_error(y, reg.predict(x))` to get the MSE and the value is also 30.985.
There is no difference in their performance as they have the same vlaue for MSE