Atefe Rajabi 40230563

1. Functionality

This exercise is structured to explore different error norms in linear regression. Below is a detailed documentation explaining each segment of the code:

1. Libraries:

- numpy for numerical purposes.

2. Data Generation

Sets a random seed for reproducibility and generates 100 random data points (`x`). It then generates corresponding `y` values based on a linear relationship with added Gaussian noise.

3. Error Norm Functions

Defines four functions, each calculating a different error norm (L0, L1, L2, Infinity) between true values (`y_true`) and predicted values (`y_pred`).

4. Error Calculation Function

Calculates the error for a given set of parameters (`alpha`, `beta`) and a chosen error norm function (`norm_func`).

5. Optimization Function

Iteratively searches for the optimal `alpha` and `beta` values that minimize the error calculated by `norm_func`. Searches within `alpha_range` and `beta_range`.

6. Norm Optimization Execution

Executes the optimization function for each error norm and prints the line equation and minimum error.

2. Result

y = 3 * x + 2 + np.random.randn(100) * 0.5

X:

```
x
```

```
array([0.5488135 , 0.71518937, 0.60276338, 0.54488318, 0.4236548 ,
       0.64589411, 0.43758721, 0.891773  , 0.96366276, 0.38344152,
       0.79172504, 0.52889492, 0.56804456, 0.92559664, 0.07103606,
       0.0871293 , 0.0202184 , 0.83261985, 0.77815675, 0.87001215,
       0.97861834, 0.79915856, 0.46147936, 0.78052918, 0.11827443,
       0.63992102, 0.14335329, 0.94466892, 0.52184832, 0.41466194,
       0.26455561, 0.77423369, 0.45615033, 0.56843395, 0.0187898 ,
       0.6176355 , 0.61209572, 0.616934  , 0.94374808, 0.6818203 ,
       0.3595079 , 0.43703195, 0.6976312 , 0.06022547, 0.66676672,
       0.67063787, 0.21038256, 0.1289263 , 0.31542835, 0.36371077,
       0.57019677, 0.43860151, 0.98837384, 0.10204481, 0.20887676,
       0.16130952, 0.65310833, 0.2532916 , 0.46631077, 0.24442559,
       0.15896958, 0.11037514, 0.65632959, 0.13818295, 0.19658236,
       0.36872517, 0.82099323, 0.09710128, 0.83794491, 0.09609841,
       0.97645947, 0.4686512 , 0.97676109, 0.60484552, 0.73926358,
       0.03918779, 0.28280696, 0.12019656, 0.2961402 , 0.11872772,
       0.31798318, 0.41426299, 0.0641475 , 0.69247212, 0.56660145,
       0.26538949, 0.52324805, 0.09394051, 0.5759465 , 0.9292962 ,
       0.31856895, 0.66741038, 0.13179786, 0.7163272 , 0.28940609,
       0.18319136, 0.58651293, 0.02010755, 0.82894003, 0.00469548])
```

Y:

```
y
```

```
array([3.06386559, 4.59598134, 4.04112135, 2.86652771, 4.01509049,
       4.88562693, 3.90215142, 4.58535658, 4.35561197, 3.67755042,
       4.17358664, 4.19790729, 3.80827117, 5.26510943, 2.39129137,
       2.61467448, 2.0659052 , 5.39079478, 4.3979263 , 4.81103113,
       5.87743038, 3.72359616, 2.74919559, 4.82628588, 1.76826158,
       4.89157366, 2.22325037, 4.46027935, 4.52701598, 3.98424322,
       3.72744632, 4.7757234 , 2.93783815, 4.66033432, 1.92236772,
       4.25413469, 4.30991315, 3.77329694, 5.13828392, 4.50656423,
       3.26673647, 2.76139547, 4.24201267, 2.84386936, 3.65301622,
       3.93709634, 2.41357091, 3.31141076, 3.28243243, 3.29486323,
       3.32563227, 3.58542914, 4.62795518, 2.32204971, 2.30870723,
       2.8221452 , 4.24762038, 2.65572543, 3.59693567, 2.18674602,
       1.73127995, 2.55082127, 4.05232552, 2.73206457, 3.78131947,
       3.57841526, 4.00656858, 2.84981197, 3.85588102, 2.05750292,
       4.89525759, 4.26262497, 4.55790585, 3.40131729, 4.16856448,
       1.78582423, 3.41173885, 1.82062393, 2.31468627, 2.13727313,
       2.70493331, 4.20755501, 2.66715289, 4.12119198, 3.0870866 ,
       3.21834996, 3.06963649, 1.50943598, 4.32185438, 4.9463599 ,
       3.41613627, 4.16159497, 2.82380889, 3.82346882, 2.35109686,
       2.89037135, 3.35783397, 1.71554775, 4.25905384, 2.02282601])
```

Result:

```
the line for L0_norm:  -5.0 * X + -5.0 and the min error is: 100

the line for L1_norm:  2.878787878787879 * X + 2.1717171717171713 and the min error is: 42.36090238016094

the line for L2_norm:  2.878787878787879 * X + 2.1717171717171713 and the min error is: 24.911088991921314
```

```
the line for infinity_norm:  3.0808080808080813 * X + 
2.1717171717171713 and the min error is: 1.0039697723467143
```

4. **Comparison and Explanation of Results:**

- L0 Norm: Looks for the line that has the least number of non-zero errors. The result might not be the best fit as it doesn't consider the magnitude of errors.

- L1 Norm: Minimizes the sum of absolute errors. Good for robustness against outliers.

- L2 Norm: Minimizes the sum of squared errors. Most common in regression, providing a balance between outlier sensitivity and fit.

- Infinity Norm: Minimizes the maximum error. The result is focused on the worst-case scenario rather than overall error, explaining why its minimum error can be significantly lower than other norms.

Each norm has its strengths and weaknesses, and their suitability varies based on the dataset and the goals of the regression analysis.

1. L0 Norm:

   - Strengths: Focuses on reducing the number of non-zero errors.

   - Weaknesses: Does not consider the magnitude of errors. It's generally not used in regression because minimizing the number of non-zero errors doesn't necessarily lead to a good fit.

   - Contextual Suitability: More theoretical than practical in regression. It's rarely the best choice for real-world datasets.

2. L1 Norm:

   - Strengths: Robust against outliers, as it doesn’t square error values.

   - Weaknesses: Can lead to multiple solutions, and it's less mathematically tractable than the L2 norm.

   - Contextual Suitability: Excellent for datasets with outliers or when a robust fit is needed.

3. L2 Norm:

   - Strengths: Balances outlier sensitivity and fit. It's the most commonly used norm in regression, providing a single, unique solution.

   - Weaknesses: Can be heavily influenced by outliers due to squaring the error terms.

- Contextual Suitability: Great for datasets without significant outliers and where a balanced fit is desired.

4. Infinity Norm:

   - Strengths: Focuses on the worst-case scenario by minimizing the largest error.

   - Weaknesses: Ignores the overall error distribution, focusing only on the single largest error.

   - Contextual Suitability: Useful when the largest error needs to be minimized, such as in systems where the worst-case scenario is critical.