

```

1  """
2  Created on Mon Apr 29 17:15:27 2024
3  @author:pcpedrosa
4  """
5
6  import matplotlib.pyplot as plt
7  import numpy as np
8  import pandas as pd
9  import seaborn as sns
10
11
12  sns.set(style="darkgrid", palette="muted")
13
14
15  def simulate_linear_data(
16      start, stop, N, beta_0, beta_1, eps_mean, eps_sigma_sq
17  ):
18      """
19      Simulate a random dataset using a noisy
20      linear process.
21
22      Parameters
23      -----
24      N: 'int'
25          Number of data points to simulate
26      beta_0: 'float'
27          Intercept
28      beta_1: 'float'
29          Slope of univariate predictor, X

```

```

31      Returns
32      -----
33      df: 'pd.DataFrame'
34          A DataFrame containing the x and y values.
35      """
36      # Create a pandas DataFrame with column 'x' containing
37      # N uniformly sampled values between 0.0 and 1.0
38      df = pd.DataFrame(
39          {"x":
40              np.linspace(start, stop, num=N)
41          }
42      )
43      # Use a linear model ( $y \sim \beta_0 + \beta_1 x + \epsilon$ ) to
44      # generate a column 'y' of responses based on 'x'
45      df["y"] = beta_0 + beta_1*df["x"] + np.random.RandomState(s).normal(
46          eps_mean, eps_sigma_sq, N
47      )
48      return df
49
50
51  def plot_simulated_data(df):
52      """
53      Plot the simulated data with sns.lmplot()
54
55      Parameters
56      -----
57      df: 'pd.DataFrame'
58          A DataFrame containing the x and y values.
59      """

```

```

59
60 # Plot the data, and a frequentist linear regression fit
61 # using the seaborn package
62 sns.lmplot(x="x", y="y", data=df, height=10)
63 plt.xlim(0.0, 1.0)
64 plt.show()
65
66
67 if __name__ == "__main__":
68     # These are our "true" parameters
69     beta_0 = 1.0 # Intercept
70     beta_1 = 2.0 # Slope
71
72     # Simulate 100 data points between 0 and 1, with a variance of 0.5
73     start = 0
74     stop = 1
75     N = 100
76     eps_mean = 0.0
77     eps_sigma_sq = 0.5
78
79     # Fix Random Seed
80     s = 42
81
82     # Simulate the "linear" data using the above parameters
83     df = simulate_linear_data(
84         start, stop, N, beta_0, beta_1, eps_mean, eps_sigma_sq
85     )
86     # Plot the simulated data
87     plot_simulated_data(df)

```

