Instructions

- Homework 1 is due November 30th at 16:00 Chicago Time.
  - We will not accept any submissions past 16:00:00, even if they are only one second late.
- You **must** upload the following files to the class Canvas:
  - `LASTNAME_FIRSTNAME.pdf`
  - `LASTNAME_FIRSTNAME.ipynb`
- Your code notebook **must** be runnable using my environment outlines in class 1 (Python 3.14, and the `requirements.txt`).
- You **must** use this template file and fill out your solutions for the written portion.
- Please note that your last name and first name should match what you appear on Canvas as.
- Include code snippets where required, as well as math and equations.
- Be *concise* where possible, all of the homework probelms can be answered in a few lines of math, code, and words.

Problem 1: One-Dimensional Data

Load in the data from the GitHub repository for this class.

```
import pandas as pd

df = pd.read_csv(
    "https://raw.githubusercontent.com/tobiasdelpozo/data-analysis-2025/refs/heads/master/homework/homework_1/homework_1_data.csv"
)
```

Problem 1.1

For the feature labeled `X1`, compute the mean, median, variance, and standard deviation. Report your numbers below (rounded to at least 4 decimal places).

Answer:

Statistics	Value
mean	12.3435
50%	11.9216
vars	38.8894
std	6.2361

Problem 1.2

Display a histogram of the feature `X1` using 50 bins. Do you think that the statistics you computed in 1.1 are good descriptors of the data? Include the graph below, and explain your reasoning in 1-2 sentences. [1]

Answer:

They are not a good descriptor of the data since the distribution has three peaks (trimodal), so the mean and median do not represent the data well.

[1] Hint: you can use `\includegraphics{}` to include images in LaTeX.

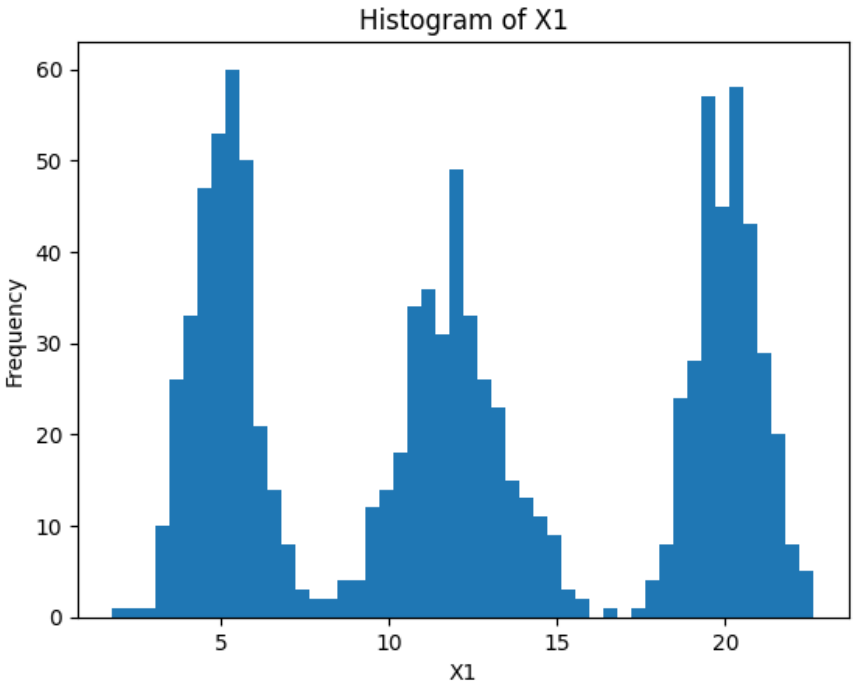

Figure 1: Histogram of feature X1 with 50 bins

Problem 1.3

Using the same feature `X1`, come up with some metrics that are descriptive of the distribution of the data. Note, this is open-ended, so think about what the data looks like, and how a human would describe it.

Answer:

As the data is trimodal, we can consider the three modes, which are approximately located at 5, 12, and 20. Computation can be done by splitting the data into three clusters and calculating the mean and standard deviation for each cluster. Another method is to use kernel estimation to estimate the density, and then find the local maxima to identify the modes.

Problem 2: kNN Regression

This problem uses the same dataset as Problem 1.

We're going to implement a k-Nearest Neighbors regression model. Unless otherwise specified, use an 80/20 train/test split for all parts of this problem.

Problem 2.1

Display a plot of `X2` versus the `target` variable. What do you notice about the relationship between these two variables?

Answer:
The target shows periodic behavior with respect to X2.

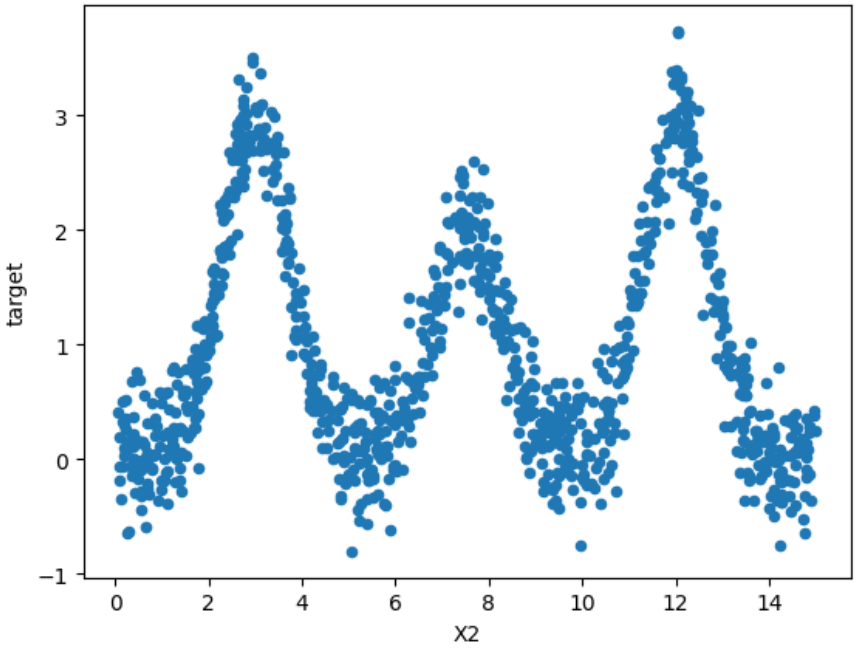

Figure 2: Scatter plot of X2 versus target variable

Problem 2.2

Implement a kNN regression model from scratch. You may use `numpy` and `pandas`, but you may not use any machine learning libraries (e.g. `scikit-learn`).

Your model should take in 4 parameters:

- `X_train`: training features
- `y_train`: training target variable
- `X_test`: testing features

- `k`: number of neighbors to use

And it should output the predicted values for `X_test`.

The algorithm you should use is as follows:

1. For each test point, compute the Euclidean distance to all training points.
2. Identify the k-nearest neighbors based on these distances.
3. Compute the predicted value as the mean of the target variable of these k-nearest neighbors.

Note that this we are only considering a single feature for this problem, so the Euclidean distance is simply $\sqrt{(x_{\text{test}} - x_{\text{train}})^2}$.

Include your code implementation below.[2]

Answer:

I attach two implementations: The first implementation uses a for-loop to compute the Euclidean distance for each test point individually. The second implementatin utilizes the matrix operation of numpy.

```python
import numpy as np
from numpy.typing import NDArray

def knn_regressor(
    X_train: NDArray[np.float64],
    y_train: NDArray[np.float64],
    X_test: NDArray[np.float64],
    k: int
) -> NDArray[np.float64]:

    y_pred = []
    for x_test in X_test:
        distances = np.linalg.norm(X_train - x_test, axis=1)
        knn_indices = np.argsort(distances)[:k]
        knn_values = y_train[knn_indices]
        y_pred.append(np.mean(knn_values))
    return np.array(y_pred)
```

Problem 2.3

Randomly split the data into training and testing sets (80/20 split), and report the Mean Squared Error (MSE) of your kNN regression model on the test set for $k = 5$.

Answer:

MSE for $k = 5$ is ≈ 0.0996.

The random split and MSE computation is as follows:

```python
ind = df.index.to_numpy()
np.random.shuffle(ind)
n = len(ind)
train_ind = ind[:int(n*0.8)]
```

[2]Hint: you can use `\lstlisting[language=python]` to include Python code snippets.

```
test_ind = ind[int(n*0.8):]

X_train = df.loc[train_ind, ["X2"]].to_numpy()
X_test = df.loc[test_ind, ["X2"]].to_numpy()
y_train = df.loc[train_ind, "target"].to_numpy()
y_test = df.loc[test_ind, "target"].to_numpy()

y_pred = knn_regressor(X_train, y_train, X_test, k=5)
mse_knn = ((y_test - y_pred)**2).mean()
mse_knn
```

Problem 2.4

For $k \in \{1, 5, 10, 20, 50, 100\}$, compute the MSE on the test set and plot the results (k values on the x-axis, MSE on the y-axis). What value of k gives the best performance on the test test?

Answer:
$k = 20$ gives the best performance with minimum MSE 0.0926.

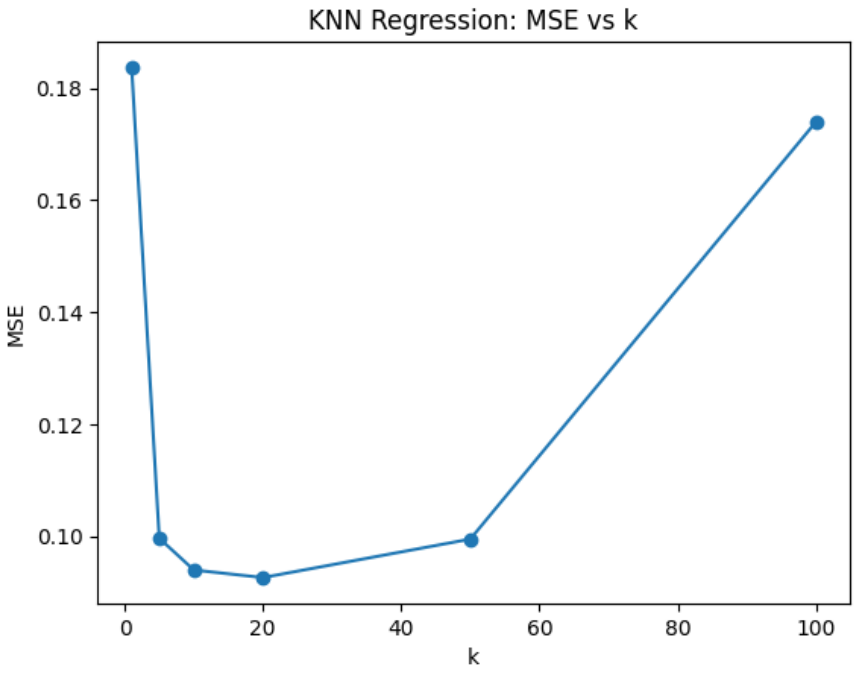

Figure 3: KNN Regression: MSE vs k

```
k_list = [1,5,10,20,50,100]
mse_list = []
for k in k_list:
    y_pred = knn_regressor(X_train, y_train, X_test, k=k)
    mse = ((res['y_test'] - y_pred)**2).mean()
    mse_list.append(mse)
mse_df = pd.Series(mse_list, index=k_list)
mse_df.plot(marker='o', xlabel='k', ylabel='MSE', title='KNN Regression: MSE vs k'
    )
print(f"minimum MSE: {mse_df.min()} at k={mse_df.idxmin()}")
```

Problem 2.5

Which value of k do you think has the highest bias? And which has the highest variance? Explain your reasoning in 1-2 sentences.

Answer:
$K = 100$ has the highest bias because it averages over a large number of neighbors, which smooths out the predictions and may miss local patterns. $K = 1$ has the highest variance because it relies on a single neighbor for each prediction, making it sensitive to noise and fluctuations in the training data.

Problem 3: Linear Regression

Using the same dataset as Problems 1 and 2, we are going to explore linear regression.

Problem 3.1

Using `statsmodels`, fit a linear regression model to predict `y` using `X3`. You should use **not** use an intercept term in your model. Report your β coefficient below:

Answer:
Running the regression, $\beta \approx 0.564234$.

Problem 3.2

Re-run the linear regression model from 3.1, but this time include an intercept term. What are your new β coefficients (intercept and slope)?

Answer:
With the intercept, $[\alpha, \beta] \approx [7.151452, -0.292625]$.

Problem 3.3

Do the following data transformations:

$$\tilde{y} = y - \bar{y} \qquad \tilde{X}_3 = X_3 - \bar{X}_3$$

Re-run the linear regression model using $\tilde{y}$ and $\tilde{X}_3$, without an intercept term. What is your β coefficient? How does it compare to your answer in 3.1?

Answer:
This result with the same coefficient as in 3.2, $\beta = -0.292625$. This is because the constant α term is achieved by

$$\alpha = \bar{y} - \beta\bar{X}_3$$

as such, centering the data removes the need for an intercept term.

Problem 3.4

Inspect your data. Display a scatter plot of `X3` versus `target`. What do you notice about the relationship between these two variables? Is a linear model appropriate for this data? Explain your reasoning in 1-2 sentences.

Answer:
The relationship between X3 and target is non-linear, showing a periodic pattern. The linear model is not appropriate as it cannot capture the oscillating behavior of the data.

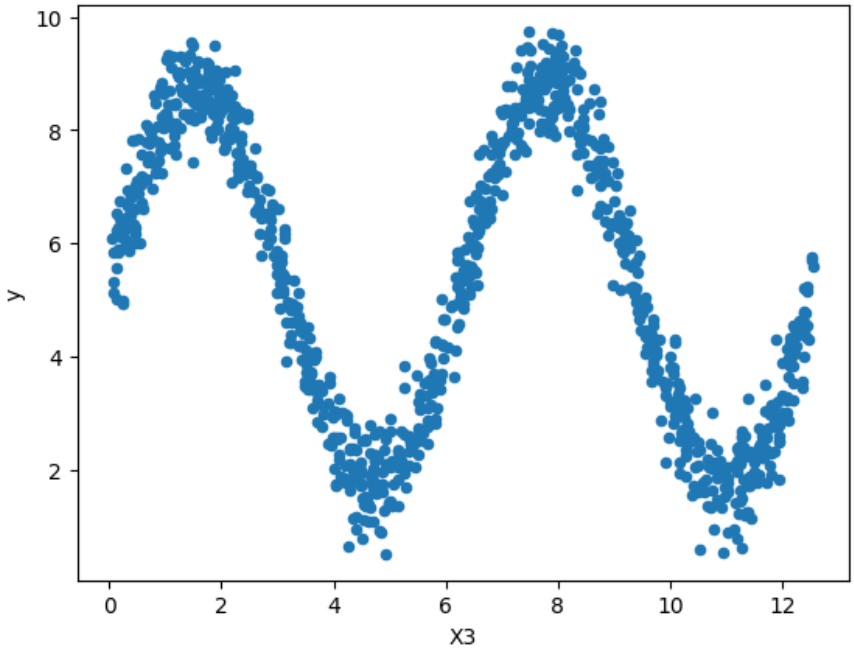

Figure 4: Scatter plot of X3 versus target variable

Problem 3.5

Define a new feature `X3_sin` as follows:

$$\texttt{X3\_sin} = \sin(\texttt{X3})$$

Fit a linear regression model to predict `target` using `X3_sin`, including an intercept term. Report your β coefficients (intercept and slope) below:

Answer:
Using the new feature `X3_sin`, the coefficients are $[\alpha, \beta] \approx [5.2484, 3.5362]$. Previously from 3.2 the R^2 was 0.171, but now it has improved to approximately 0.964, indicating a better fit.

Problem 3.6

Display a plot of `X3_sin` versus `target`. Do you think a linear model is appropriate for this data? Explain your reasoning in 1-2 sentences.

Answer:

After the sinusodial transformation, the relationship between `X3_sin` and `target` appears more linear. Thus, a linear model is more appropriate for this transformed data.

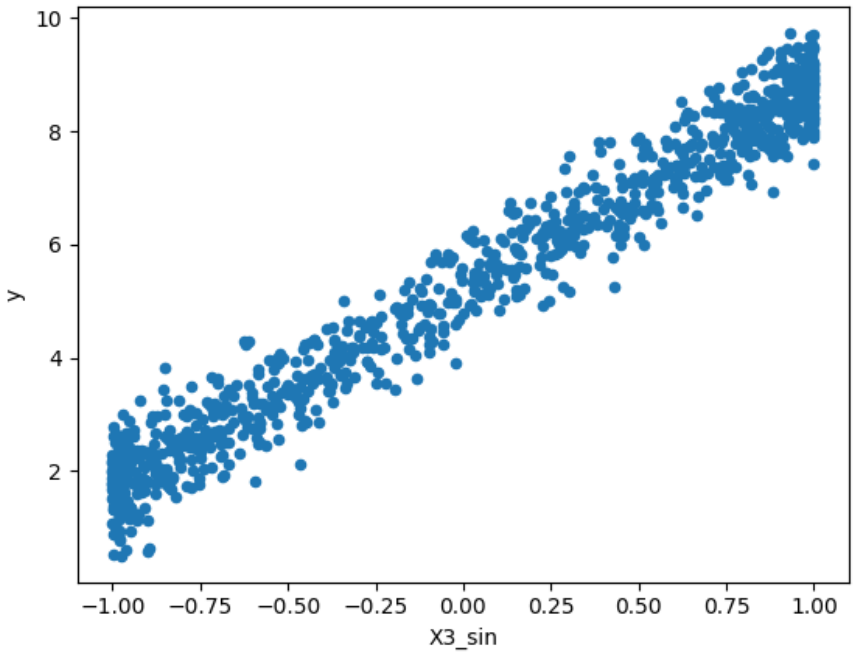

Figure 5: Scatter plot of X3_sin versus target variable