# ST239: Assignment 1

## Instructions

This assignment is worth 20% of the total module mark.

The assignment is composed of **4 Questions** worth at total of **60 points**, respectively, and a few optional questions.

**Submission Components** Your submission should include the following documents:

- Python Notebooks (with `.ipynb` extension) containing your answers for the exercises; including code and written text. You can choose to use one single notebook for the whole assignment or split it into multiple ones.

  The code should be fully reproducible (i.e. I should be able to run it in my local machine with no errors) and well-commented.

  **Style suggestions:**

  - Comments and written answers can be included directly on the Notebook using markdown cells. Alternatively, you can attach an accompanying text document (pdf, word) and refer to the question you are answering.
  - For each chunk of code or procedure used, you should add a small comment in code or a description on its functionality and how it is being used in a text cell.
  - DO NOT submit the datasets. I will place your code in a folder with them, so you should access them with a simple path command, e.g. `file-name.csv`

**!!! The integrity of the submission is the students' responsibility: make sure that your submission contains all the right files and that it is not corrupted !!**
**If you notice any issues with your submission: contact me asap.**

**The use of Generative AI Tools for this assignment is forbidden.**
If I suspect that you have used any such tools for the assessed coursework you risk being referred to an academic conduct panel.

## Q1. Warm up: Simple 'Linear' Regression

In recent labs, we have worked with the `'mpg'` dataset for the purpose of developing a multiple linear regression model. However, at the exploratory data analysis stage a preliminary finding was that the paiwise relationship between the target and a number of regressors was not quite linear.

- Give some context on how to perform simple linear regression on a transformation of the explanatory variable, suggest a plausible transformation for the regressors in the example and write down the new model mathematically.
- Fit a simple regression using the transformed regressor (one for each non-linear pair).
- Interpret the output of the regression analysis; explain how the parameters change meaning in light of the new model and discuss what simple 'linear' model best explains fuel efficiency (mpg).
- Compare your results with the simple linear regression model developed in previous labs using suitable metrics and criteria.
- For the 'best' transformed linear model, derive diagnostics and discuss them in comparison to the original linear case.

**[12 marks.]**

## Q2. Regression and MLE *from scratch*

1. Consider the Poisson model introduced in class and the MLE method.

   You are now provided with a record of weekly received emails over the course of one year in file `weekly_emails.csv`.

   - Plot the likelihood and log-likelihood curves with respect the unkown parameter $\lambda$ to the and explain your reasoning with a small text comment.
   - Highlight $\hat{\lambda}_{MLE}$ the maximum likelihood estimate for the observed data under the Maximum Likleihood Criterion.

   The library `optimize` in Python allows to optimize minimise or maximise a function created by the user using some built-in numerical methods as illustrated below

```
from scipy.optimize import minimize

def fun(x, beta)
    return x*beta+3

result = minimize(fun, x0, method="Nelder-Mead")

print(result)
```

   In the code above, the function `fun` represents the objective function to be optimized. It takes a parameter value `x` and returns the quantity $x \cdot \beta + 3$, which is treated as the function to be minimized.

   The optimization procedure starts from an initial guess `x0` and applies the Nelder–Mead algorithm, a gradient-free optimization method. At each iteration, the algorithm proposes a new value of the parameter `x` with the aim of reducing the value of the objective function. The iterations continue until convergence (the final value is found) is met.

   The output of the optimization is stored in the object `result`, which contains the final parameter estimate, the value of the objective function at the solution, and diagnostic information about the convergence.

   You should read more about this function in the aszsociated documentation.

- Implement a code to using the `minimise` function to find the MLE of the example above adapting the folllowing code:

  ```
  result = minimize(
          fun,
          x0,
          method="Nelder-Mead",
          callback=callback
      )
  ```

  and consider ways to illustrate the trajectory path from the initial guess to the final value. Think of comparing this visually with the result of the previous part.

2. In the lectures, we highlighted the limitations of the linear model when the relationship between the explanatory variable $X$ and the response $Y$ is in fact non-linear. We consider a transformation thereof in the Q1, another alternative *when sensible* is to consider a *polynomial* regression.

   The file `Q2_curve.csv` is a $N \times 2$-dimensional numpy array with two columns for $X$ and $Y$ with a complex pattern that you are asked to model with a non-linear statistical model.

   Your task is to find the relationship,

   $$Y = f(X) + \varepsilon.$$

   - Use the function `curve_fit` from the `scipy.optimize` library to automatically optimize the parameters of a custom model you define, based on the non-linear least squares criterion. This automated optimizer requires the following arguments: your custom model function, the explanatory data, the target data, and an initial guess for the parameters. For more details, refer to the `help` function.
   - Experiment with potential models that include higher-order terms of $X$.
   - Compare this with the output of a standard `sm` procedure and comment.

**[8 marks.]**

## Q3. Analysis of the ‘house efficiency’ dataset

Your goal is to develop and interpret a multiple linear regression model to explain monthly household electricity consumption (kwh_month) using building characteristics, energy-efficiency indicators, and a few additional variables.

| Variable | Type | Description |
|---|---|---|
| kwh_month | Continuous | Monthly household electricity consumption (kWh) |
| floor_area | Continuous | Total floor area of the dwelling (m$^2$) |
| insulation | Continuous | Insulation quality score (0 = very poor, 10 = excellent) |
| avg_temp | Continuous | Average outdoor temperature during the month (°C) |
| heat_pump | Binary | Indicator variable: 1 if a heat pump is installed, 0 otherwise |
| appliances | Integer | Number of major electrical appliances in the household |
| building_age | Continuous | Age of the building (years) |
| garden_size | Continuous | Size of the private garden area (m$^2$) |
| wall_colour_light | Binary | Indicator variable: 1 if the external wall colour is light, 0 otherwise |
| distance_city_km | Continuous | Distance of the dwelling from the city centre (km) |

Table 1: Description of variables in the synthetic house energy consumption dataset

1. **EDA -** Perform preliminary Exploratory Data Analysis:
   - Produce appropriate summary statistics and plots (histograms, scatter plots, boxplots).
   - Explore relationships between kwh_month and each predictor, and discuss any evident trends or anomalies.
2. **Fit -** Estimate a multiple linear regression model with the training data. Select the best model under the principles illustrated in the lecture:
   - Consider a few model selection strategies (backward elimination, forward selection) and illustrate the criteria you used to reach the final model.
   - Report the chosen final model, provide coefficient interpretations in the context of household energy use, and state which variables you decided to drop and why.

3 **Diagnostics -** Evaluate the quality of the final model using residual diagnostics:

- Residuals vs fitted values (linearity and heteroskedasticity).
- Q–Q plot (normality of errors).
- Leverage and influence (Cook's distance).
- Discuss whether the assumptions of linear regression appear reasonable and what remedies could be considered.
- The dataset contains a few influential observations. Explain how you would identify them, how they can affect coefficient estimates and inference, and how you would report results both *with* and *without* these observations.

4 **Collinearity -** Compute and comment on the VIF for the fitted model. Discuss whether multicollinearity is problematic and how it affects inference.

5 **Prediction and evaluation -**

- Use the fitted model to predict the monthly electricity consumption `kwh_month` for the following household that was not included in the original dataset:
  - Floor area: 160 m$^2$
  - Insulation score: 6
  - Average outdoor temperature: 8°C
  - Heat pump installed: Yes
  - Number of appliances: 12
  - Building age: 35 years
  - Garden size: 180 m$^2$
  - External wall colour is light
  - Distance from city centre: 7 km

  Briefly discuss which characteristics of this dwelling contribute the most to increasing or decreasing the predicted electricity consumption, based on the estimated coefficients.

**[25 marks.]**

## Q4. Logistic regression to detect phishing emails

The dataset 'spam_phishing.csv' contains the following information:

| Variable name | Description | Type | Range / Values |
|---|---|---|---|
| `attachment_size_kb` | Size of email attachments in kilobytes. Equals 0 if the email has no attachment. | Numeric | $\geq 0$ (typically 0–2000 KB) |
| `special_char_ratio_pct` | Percentage of special characters relative to the total number of characters in the email body, multiplied by 100. | Numeric | 0–30 |
| `has_fishy_words` | Indicator for whether the email contains suspicious words such as "urgent", "win", "offer", etc. | Binary | 0 = No, 1 = Yes |
| `is_spam` | Target variable indicating whether the email is phishing/spam. | Binary | 0 = Non-spam, 1 = Spam |

Table 2: Description of variables used in the phishing email classification dataset.

Your goal is to develop and interpret a logistic regression model to classify email as **phishing/spam** or **non-spam** using a small set of features.

- Restrict your attention to the first 400 observations of the dataset for inference and leave the remaining 50 observations for future validation (until further indications).
- **EDA -** Develop some preliminary Exploratory Data Analysis: use appropriate plots to explore the variables and their relationships to one another. Add some preliminary insights or and discuss any comments you may have.
- **Fit -** Estimate a logistic regression model with the training data. Select the best model under the principles illustrated in the lecture. Write down the final model, interpret the outputs in Python with focus on the parameters and their meaning in terms of Odds Ratio for the context of spam detection.
- **Prediction -**
  - Use the fitted model to predict spam probabilities for the 50 test observations.
  - Use this quantity to classify the emails using a 0.5 probability threshold and a 0.7 one.
  - Compare the two threshold in terms of predictive performance; you may use the metric that you find most appropriate upon explaining your choice intuitively.

- **Question 1:** Under the final model, consider two emails:
  - Email A: very large attachment, very low special-character ratio (between 0-1), contains one fishy word.
  - Email B: even larger attachment (by 120KB), a few special characters (a ratio of 4), no fishy words.

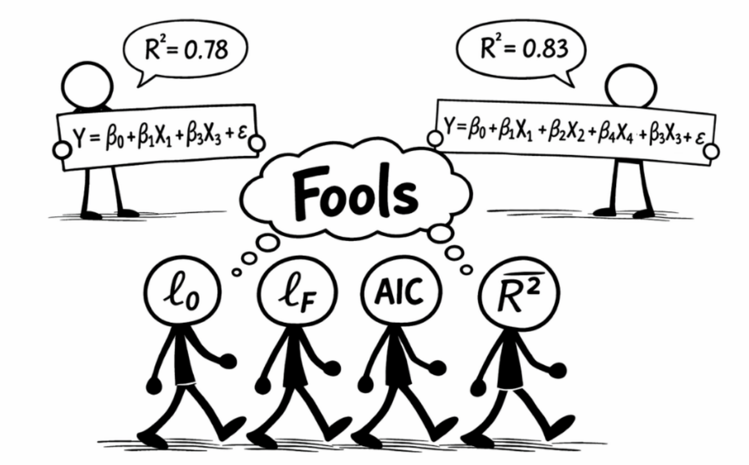
  According to the logistic regression model, who has a higher probability of having received a phishing email?

- **Question 2:** Suppose that the dataset, in addition to the variables already used, a further categorical variable called `‘sender_type’` is introduced. This has three levels: known contact (i.e friend, colleague, etc), unknown (no prior interaction), trusted entity (claims to be a bank, delivery service, platform etc).
  Explain how you would include this in the model, what would be the associated parameter, their interpretation in terms of Odds Ratio and what may be your pre-judgement on the effect of this variable in the context of spam detection.

**[15 marks.]**

## Bonus Question

Can you explain to a non-statistician the following vignette?


$R^2 = 0.78$
$Y = \beta_0 + \beta_1 X_1 + \beta_3 X_3 + \varepsilon$
$R^2 = 0.83$
$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_4 X_4 + \beta_3 X_3 + \varepsilon$
Fools
$\ell_0$
$\ell_F$
AIC
$\overline{R^2}$

You will still receive feedback on your explanation, which can be useful to improve communication/explanation skills. **[0 marks.]**