Data Science
Multiple linear regression

Manuel Pita, Universidade Lusófona. CICANT October 29, 2023

This is version 0.1 of this booklet. if you find any errors, please send and email to
manuel.pita@ulusofona.pt

Introduction

Recall that **Simple Linear Regression (SLR)** is a statistical method that models the relationship between a single independent variable and a dependent variable. In essence, it's like drawing a straight line (best fit) through data points on a two-dimensional plot. The mathematical equation for SLR can be expressed as:

$$y = \beta_0 + \beta_1 x_1 + \epsilon$$

Where:

- y is the dependent variable (what you're trying to predict).
- x_1 is the independent variable.
- β_0 is the y-intercept.
- β_1 is the slope of the line.
- ϵ represents the residuals or errors in the predictions.

Now we turn our focus to **Multiple Linear Regression (MLR)**, which extends the concept of simple linear regression to include two or more independent variables. Instead of fitting a straight line in a 2D space (as in SLR), in MLR, we are fitting a plane or a hyperplane in a higher-dimensional space. The mathematical equation for MLR can be expressed as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon$$

Where:

- $x_1, x_2, \ldots, x_k$ are the independent variables.
- β_0 is the intercept.
- $\beta_1, \beta_2, \ldots, \beta_k$ are the coefficients for the independent variables.
- ϵ represents the residuals or errors in the predictions.

Keep in mind that while SLR deals with a single predictor and a response, MLR deals with multiple predictors. MLR allows for a more comprehensive analysis by considering multiple factors simultaneously, but it also introduces complexities, like for example pre-existing relationships between predictors. However, do not worry about those complexities for now.

Obtaining a MLR Coefficient Estimates with Python

Multiple Linear Regression (MLR) is an extension of Simple Linear Regression that allows for modelling relationships between a dependent variable and multiple independent variables. In this section, we'll explore how to obtain the coefficient estimates for an MLR model using Python's `scikit-learn` library.

Step 1: Setting up the Environment

Firstly, ensure you have `scikit-learn` installed. If not, it can be installed using `pip`:

```
pip install scikit-learn
```

Step 2: Preparing the Data

For the sake of explanation, let's assume you have a dataset with a dependent variable y and two independent variables x_1 and x_2.

Load your data (for example, using `pandas`):

```
import pandas as pd

data = pd.read_csv('your_data_file.csv')
X = data[['x1', 'x2']]
y = data['y']
```

Step 3: Building the Model

Once the data is prepared, we can fit an MLR model:

```
from sklearn.linear_model import LinearRegression

model = LinearRegression().fit(X, y)
```

Step 4: Extracting Coefficient Estimates

After fitting the model, the coefficient estimates can be accessed using the `coef_` attribute for the independent variables and `intercept_` for the intercept:

```
intercept = model.intercept_
coef_x1, coef_x2 = model.coef_

print(f"Intercept (beta_0): {intercept}")
print(f"Coefficient for x1 (beta_1): {coef_x1}")
print(f"Coefficient for x2 (beta_2): {coef_x2}")
```

The output will provide the estimated coefficients, which represent the change in the dependent variable for a one-unit change in the respective independent variable, holding all else constant.

Concluding Notes

Using Python's `scikit-learn` library simplifies the process of estimating coefficients in MLR. Once the model is built, the coefficients provide valuable insights into the relationships between the dependent and independent variables. Remember to interpret the coefficients in the context of your data and domain knowledge.

For further details and advanced options, you can refer to the `scikit-learn` documentation at `https://scikit-learn.org/stable/`.