# Linear Regression Project

Eton Tackett

June 25 2025

## 1 Introduction and Problem Statement

We are given the problem statement (Taken from ISLR): Suppose that we are statistical consultants hired by a client to provide advice on how to improve sales of a particular product. The advertising data set consists of the sales of that product in 200 different markets, along with advertising budgets for the product in each of those markets for three different media: TV, radio, and newspaper. [...] It is not possible for our client to directly increase sales of the product. On the other hand, they can control the advertising expenditure in each of the three media. Therefore, if we determine that there is an association between advertising and sales, then we can instruct our client to adjust advertising budgets, thereby indirectly increasing sales. In other words, our goal is to develop an accurate model that can be used to predict sales on the basis of the three media budgets.

## 2 Key Observations

We analyze a dataset titled advertising.csv with data from 35 employees which includes their salary, location, department, years of experience, among others. We observed many outliers in the dataset which can impact our model. The data is structured, which means that it has a predefined format and all the data has consistent attributes. There is no missing data (NaN values).

## 3 Tools Used

We plan to use Python to implement the model. The libraries we plan to use include pandas, scikitlearn, numpy, matplotlib. This gives us tools for data management, machine learning, and data analysis via visualization.

## 4 Methods

We will use the EDA process throughout. The pandas library will help us load the data into Jupyter Notebook. We will then make use of pandas to confirm our observations. Bivariate analysis will be used to explore and uncover relationships in the data (Radio vs Sales, TV vs Sales, etc). I will use scatter plots, correlation coefficients, and covariance to better understand these relationships.

Once the EDA process is complete, we can then move to the preprocessing the data, building the pipeline, and evaluating the model. I think a Linear Regression model is suitable for this. Mathematically, the equation is $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $X \in \mathbb{R}^{m\times n}$ is our data/feature matrix, $\boldsymbol{\beta} \in \mathbb{R}^{n\times 1}$ is our coefficient vector. For simplicity, we assume $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 I)$ is noise. This assumption ensures the expectation of the errors is 0 and the variance is fixed (Unbiased model). If the relationships are non-linear, we can easily use polynomial regression instead.

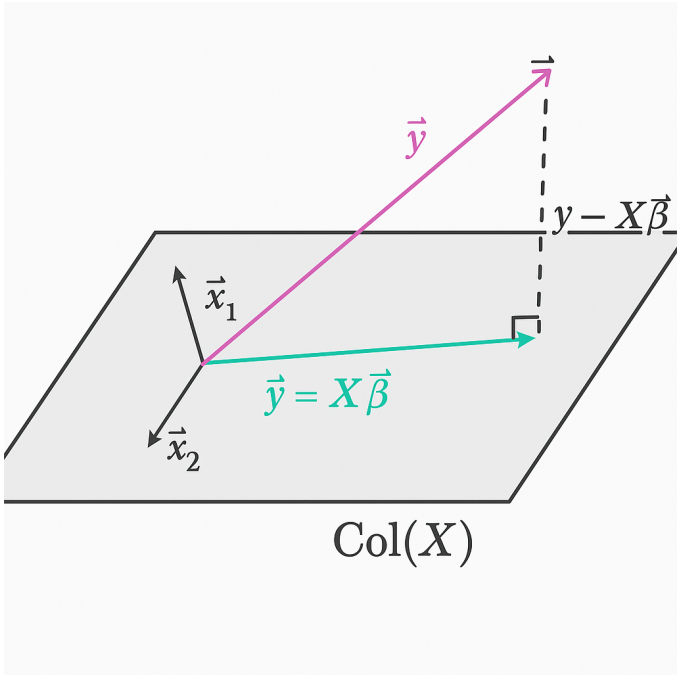
## 5 Mathematical Background

The goal mathematically is to find the optimal solution $\hat{\beta}$ such that the error vector $\min_{\boldsymbol{\beta}} \|\mathbf{y} - X\hat{\boldsymbol{\beta}}\|_2^2$. This $\hat{\beta}$ can be solved by using the normal equations:

$$X^\top X\hat{\boldsymbol{\beta}} = X^\top \mathbf{y}$$

Assuming $\left(X^T X\right)^{-1}$ has full rank, $\hat{\beta} = \left(X^T X\right)^{-1} X^T y$ is the optimal solution. The projection vector onto the column space of X is then given by

$$X\hat{\boldsymbol{\beta}} = X\left(X^\top X\right)^{-1} X^\top \mathbf{y} = \hat{\mathbf{y}}$$




Geometric visualization of Linear Least Squares (Green vector should be $X\hat{\boldsymbol{\beta}} = \hat{\mathbf{y}}$)

If we assume $X \in \mathbb{R}^{m \times n}$ has rank r then we know $col\,(X) \subset \mathbb{R}^m$ and the vector

$$X\hat{\boldsymbol{\beta}} = \hat{\mathbf{y}}$$

is the orthogonal projection of $\mathbf{y}$ onto some r-dimensional hyperplane of $\mathbb{R}^m$.

# 6 Results and Discussion

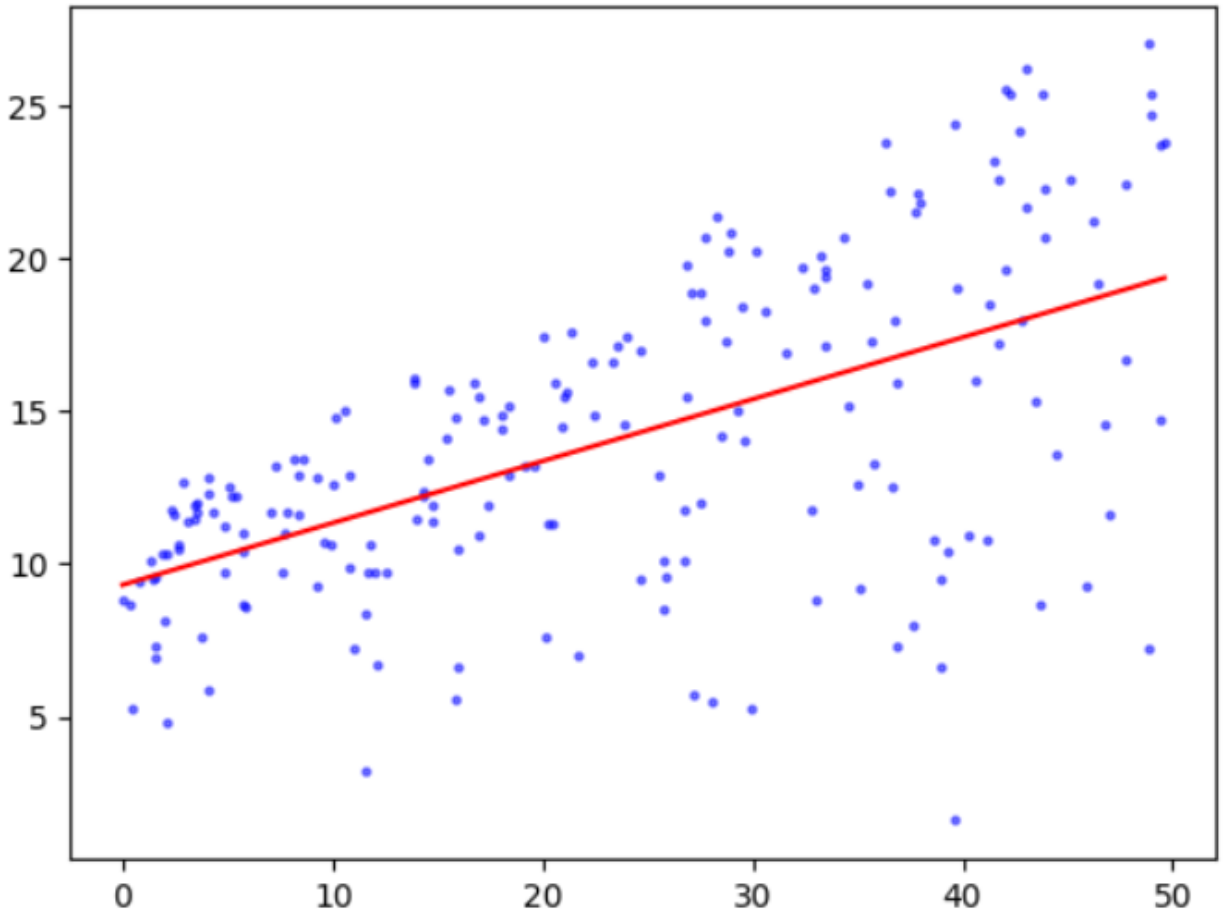
```
def gramschmidt(X):
    (r,p) = X.shape
    Q = np.zeros((r,p))
    R = np.zeros((p,p))
    for j in range(p):
        v = np.copy(X[:,j])
        for i in range(j):
            R[i,j] = np.dot(Q[:,i], X[:,j])
            v -= R[i,j]*Q[:,i]
        R[j,j] = LA.norm(v)
        Q[:,j] = v/R[j,j]
    return Q, R
```

```
[26] def backsubs(R,b):
        m = b.shape[0]
        x = np.zeros(m)
        for i in reversed(range(m)):
            x[i] = (b[i] - np.dot(R[i,i+1:m],x[i+1:m]))/R[i,i]
        return x
```

```
[27] def ls_by_qr(X, b):
        Q, R = gramschmidt(X)
        return backsubs(R, Q.T @ b)
```

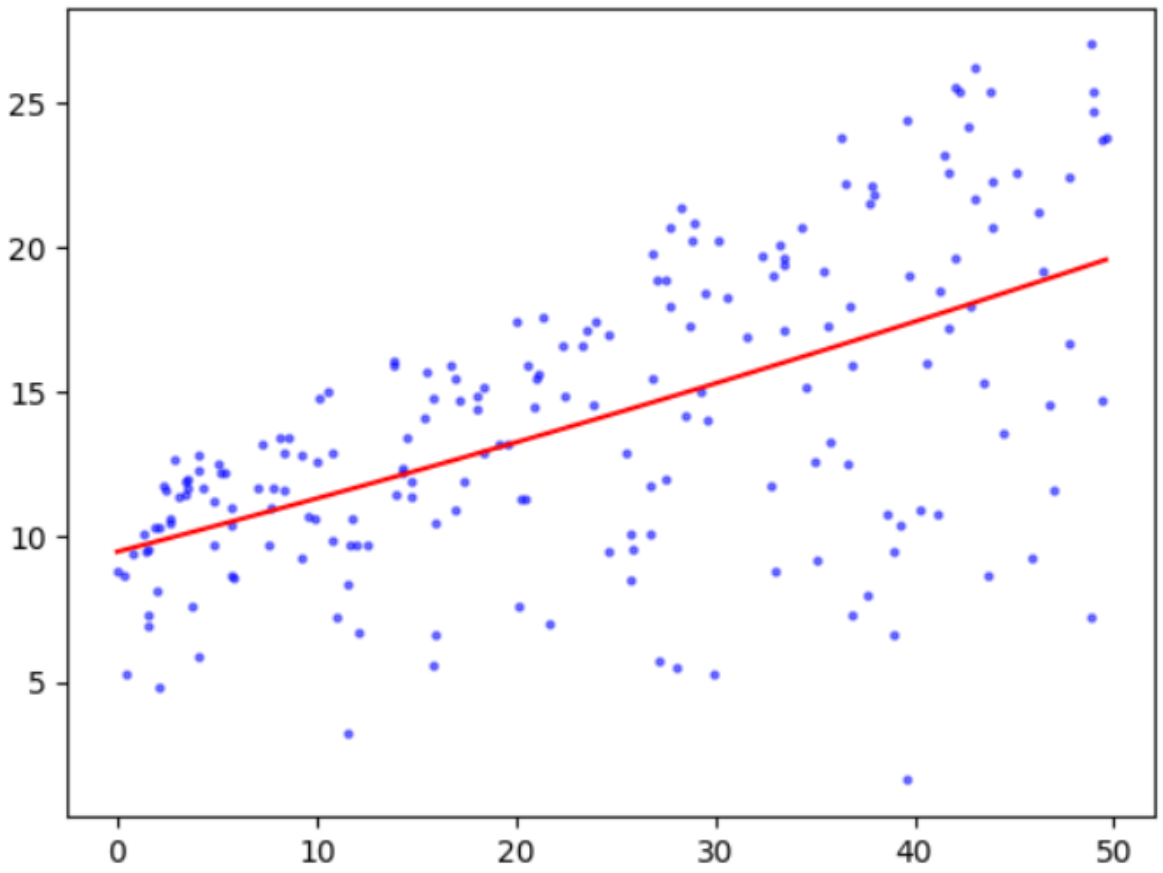
Functions taken from Math 535 Notebook

I decided to analyze the relationship between radio and sales and the linear regression plot is below




Regression plot Radio vs Sales

From this, a polynomial regression model might be better. We use a degree 2 polynomial to best capture the trends.




Polynomial Regression plot Radio vs Sales