Computational Linear Algebra Final Project:

Least Square Methods with Application in Regression Analysis

Veronika Titarchuk, Lizzy Eccles, Mingyu Huang

Contents

1 Introduction

Regression is a technique in statistics used for determining and modeling the relationships between variables. Regression is used in many fields including engineering, multiple sciences, economics, management, and almost every other field imaginable. An example referred to many times in this process involves a drink bottler. Suppose an industrial engineer employed by a soft drink beverage bottler is analyzing the operations involving product delivery for a vending machine. The engineer suspects that the time required to load and service a machine by a deliveryman is related to the number of cases of soft drinks delivered and the distance walked by the driver in feet and meters. He randomly chose 25 vending machines and the delivery time in minutes and the volume of product delivered in cases were recorded for each machine.

The objective of linear regression is to represent the relationship between variables as a straight-line function. This function takes the form outlined in Equation 1. This project will aim to estimate the β_k parameters by $\hat{\beta}_k$ using least square techniques where Equation 2 represents this estimated function. These $\hat{\beta}_k$ values are estimated such that the sum of squares of the differences between the observations y_i and the straight line made is minimized. The topics discussed are going to be the least squares method, generalized inverse, singular value decomposition, generalized least squares method, condition number, and convergence and divergence.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \epsilon \tag{1}$$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_k x_k \tag{2}$$

2 Techniques

2.1 Least Squares

2.1.1 Simple Linear Regression

To start off simple, let's consider a simple linear regression model. This model focuses on the relationship between a response variable y and a single regressor variable x. The reduced straight-line model is represented below with the estimated simple linear regression line given by Equation 4.

$$y = \beta_0 + \beta_1 x + \epsilon \tag{3}$$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{4}$$

The least squares method is used to estimate β_0 and β_1. To minimize the sum of squares difference, a criterion needs to be defined

$$S(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \tag{5}$$

The least squares estimators for β_0 and β_0, must satisfy

$$\left.\frac{\partial S}{\partial \beta_0}\right|_{\hat{\beta}_0,\hat{\beta}_1} = -2\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \tag{6}$$

$$\left.\frac{\partial S}{\partial \beta_1}\right|_{\hat{\beta}_0,\hat{\beta}_1} = -2\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)x_i = 0 \tag{7}$$

If we simplify the equations above, we end up with the least squares normal equations

$$n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \tag{8}$$

$$\hat{\beta}_0 \sum_{i=1}^{n} x_i + \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i \tag{9}$$

Finding the solution to the normal equations yields

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{10}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} y_i x_i - \frac{(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}} \tag{11}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. The $\hat{\beta}_0$ and $\hat{\beta}_1$ are the least squares estimators for the intercept and slope of the $\hat{y}$ function. A more simplistic form of the $\hat{\beta}_1$ function is denoted using S_{xx} and S_{xy} as follows.

$$S_{xx} = \sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n} = \sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{12}$$

$$S_{xy} = \sum_{i=1}^{n} y_i x_i - \frac{(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{n} = \sum_{i=1}^{n} y_i(x_i - \bar{x}) \tag{13}$$

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} \tag{14}$$

Shear Strength Example

A motor manufacturer uses a bonding igniter propellant and a sustainer propellant combined inside a metal housing. The shear strength between the two propellants is an important factor for quality. A quality control expert suspects that the shear strength is related to the age of the propellant in weeks. They collected 20 batches of propellant and the age of the batch and the shear strength were recorded in Figure 1.

$$S_{xx} = 4677.69 - \frac{71422.56}{20} = 1106.56$$

$$S_{xy} = 528492.64 - \frac{(267.25)(42627.15)}{20} = -41112.65$$

$$\hat{\beta}_1 = \frac{-41112.65}{1106.56} = -37.15$$

$$\hat{\beta}_0 = 2131.3575 - (-37.15)13.3625 = 2627.82$$

This implies that the least squares fit is $\hat{y} = 2627.82 - 37.15x$.

Shear Strength, y_i (psi)	Age of Propellant, x_i (weeks)
2158.70	15.50
1678.15	23.75
2316.00	8.00
2061.30	17.00
2207.50	5.50
1708.30	19.00
1784.70	24.00
2575.00	2.50
2357.90	7.50
2256.70	11.00
2165.20	13.00
2399.55	3.75
1779.80	25.00
2336.75	9.75
1765.30	22.00
2053.50	18.00
2414.40	6.00
2200.50	12.50
2654.20	2.00
1753.70	21.50

Figure 1: The shear strength of the bond between two types of propellant with age of propellant.

```
Python code:

A = np.vstack([x, np.ones(len(x))]).T

# turn y into a column vector
y = y[:, np.newaxis]

alpha = np.dot((np.dot(np.linalg.inv(np.dot(A.T,A)),A.T)),y)
print("Coefficients:\nIntercept: ", alpha[1][0],"\nAge of Propellant: ", alpha[0][0])
```

```
# plot the results
plt.figure(figsize = (10,8))
plt.plot(x, y, 'b.')
plt.plot(x, alpha[0]*x + alpha[1], 'r')
plt.xlabel('x')
plt.ylabel('y')
plt.show()

Python output:
Coefficients:
Intercept:  2627.8223590012963
Age of Propellant:  -37.153590944905226
```

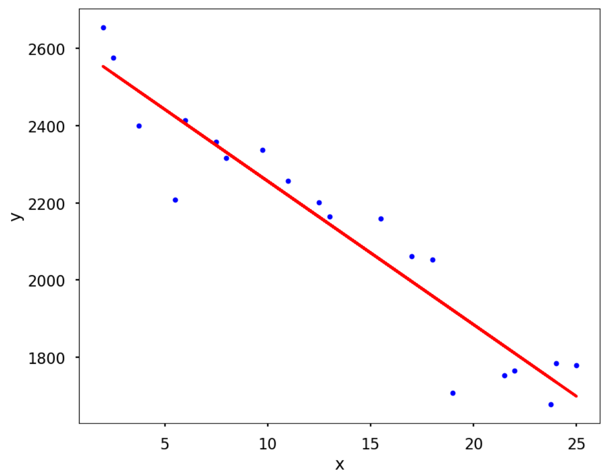

Figure 2: Simple Linear Regression of shear strength based on the age of the propellant in weeks.

2.1.2 Multiple Linear Regression

Refer back to the example explained involving delivery time depending on the number of cases and distance walked. The data for this section is given in Figure 2 and the multiple regression model showing the relationship between these variables is Equation 15.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon \tag{15}$$

In this case, y would represent the delivery time in minutes, x_1 is the number of cases delivered, and x_2 is the distance traveled in feet. The unknown β_0, β_1, and β_2 will be estimated using $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$. The multiple linear regression estimation will then be

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 \tag{16}$$

To minimize the sum of squares difference, the criterion for multiple linear regression is

Deliver time, y (min)	Number of Cases, x_1	Distance (ft), x_2
16.68	7	560
11.50	3	220
12.03	3	340
14.88	4	80
13.75	6	150
18.11	7	330
8.00	2	110
17.83	7	210
79.24	30	1460
21.50	5	605
40.33	16	688
21.00	10	215
13.50	4	255
19.75	6	462
24.00	9	448
29.00	10	776
15.35	6	200
19.00	7	132
9.50	3	36
35.10	17	770
17.90	10	140
52.32	26	810
18.75	9	450
19.83	8	635
10.75	4	150

Figure 3: Full delivery time data set with number of cases and distances traveled in feet.

$$S(\beta_0, \beta_1, \beta_2, ...\beta_k) = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{i=1}^{k} \beta_j x_{ij})^2 \tag{17}$$

The least squares estimators for β_0, β_1, β_2, ..., β_k must satisfy the following two equations.

$$\left.\frac{\partial S}{\partial \beta_0}\right|_{\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k} = -2\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \sum_{i=1}^{k} \hat{\beta}_j x_{ij}) = 0 \tag{18}$$

$$\left.\frac{\partial S}{\partial \beta_j}\right|_{\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k} = -2\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \sum_{i=1}^{k} \hat{\beta}_j x_{ij}) x_{ij} = 0 \tag{19}$$

where $j = 1, 2, ..., k$. From this, the normal equations are made and are as follows

$$\begin{aligned}
n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik} &= \sum_{i=1}^{n} y_i \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1}^2 + \hat{\beta}_2 \sum_{i=1}^{n} x_{i1} x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{i1} x_{ik} &= \sum_{i=1}^{n} x_{i1} y_i \\
\vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad &\qquad \vdots \\
\hat{\beta}_0 \sum_{i=1}^{n} x_{ik} + \hat{\beta}_1 \sum_{i=1}^{n} x_{ik} x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{ik} x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 &= \sum_{i=1}^{n} x_{ik} y_i
\end{aligned} \tag{20}$$

The solution of these equations will yield $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k$. A more simplistic way to represent these equations is in matrix form.

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon} \tag{21}$$

where

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{bmatrix}, \boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & ... & x_{1k} \\ 1 & x_{21} & x_{22} & ... & x_{2k} \\ ... & & & & \\ 1 & x_{n1} & x_{n2} & ... & x_{nk} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ ... \\ \beta_k \end{bmatrix}, \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ ... \\ \epsilon_n \end{bmatrix}$$

This notation requires $\boldsymbol{y}$ to be $n \times 1$, $\boldsymbol{X}$ to be $n \times p$, $\boldsymbol{\beta}$ to be $p \times 1$, and $\boldsymbol{\epsilon}$ to be $n \times 1$. To find the vector of the $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k$, the least square estimators $\hat{\boldsymbol{\beta}}$, minimize

$$\boldsymbol{S(\beta) = (y - X\beta)'(y - X\beta) = y'y - 2\beta'y + \beta'X'X\beta} \tag{22}$$

The estimators must satisfy

$$\left.\frac{\partial S}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}} = -2\boldsymbol{X'y} + 2\boldsymbol{X'X\hat{\beta}} = \boldsymbol{0} \tag{23}$$

and simplifies to

$$\boldsymbol{X'X\hat{\beta}} = \boldsymbol{X'y} \tag{24}$$

solving the normal equations gives us the least squares estimates vector shown in equation 25.

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y} \tag{25}$$

If the regressors or x variables are linearly independent, the $(\boldsymbol{X'X})^{-1}$ will always exist. Later in the paper, it will be discussed what to do if the columns are not linearly independent.

The following matrices represent if Equation 24 were to be written out explicitly.

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i2} & \cdots & \sum_{i=1}^{n} x_{ik} \\ \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i1}^2 & \sum_{i=1}^{n} x_{i1}x_{i2} & \cdots & \sum_{i=1}^{n} x_{i1}\boldsymbol{x}_{ik} \\ \vdots & \vdots & \vdots & & \vdots \\ \sum_{i=1}^{n} x_{ik} & \sum_{i=1}^{n} x_{ik}x_{i1} & \sum_{i=1}^{n} x_{ik}x_{i2} & \cdots & \sum_{i=1}^{n} x_{ik}^2 \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_{i1}y_i \\ \vdots \\ \sum_{i=1}^{n} x_{ik}y_i \end{bmatrix} \tag{26}$$

This indicates the fitted regression model matrix form

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X'X})^{-1}\boldsymbol{X'y} \tag{27}$$

Delivery Time Example

Using the data from Figure 2, the $\boldsymbol{X}$ and $\boldsymbol{y}$ matrices are defined as follows where $\boldsymbol{y}$ is the delivery time in minutes and each row of the $\boldsymbol{X}$ represents the the x_k values for each $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ estimates, in that order.

$$\mathbf{X} = \begin{bmatrix} 1 & 7 & 560 \\ 1 & 3 & 220 \\ 1 & 3 & 340 \\ 1 & 4 & 80 \\ 1 & 6 & 150 \\ 1 & 7 & 330 \\ 1 & 2 & 110 \\ 1 & 7 & 210 \\ 1 & 30 & 1460 \\ 1 & 5 & 605 \\ 1 & 16 & 688 \\ 1 & 10 & 215 \\ 1 & 4 & 255 \\ 1 & 6 & 462 \\ 1 & 9 & 448 \\ 1 & 10 & 776 \\ 1 & 6 & 200 \\ 1 & 7 & 132 \\ 1 & 3 & 36 \\ 1 & 17 & 770 \\ 1 & 10 & 140 \\ 1 & 26 & 810 \\ 1 & 9 & 450 \\ 1 & 8 & 635 \\ 1 & 4 & 150 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 16.68 \\ 11.50 \\ 12.03 \\ 14.88 \\ 13.75 \\ 18.11 \\ 8.00 \\ 17.83 \\ 79.24 \\ 21.50 \\ 40.33 \\ 21.00 \\ 13.50 \\ 19.75 \\ 24.00 \\ 29.00 \\ 15.35 \\ 19.00 \\ 9.50 \\ 35.10 \\ 17.90 \\ 52.32 \\ 18.75 \\ 19.83 \\ 10.75 \end{bmatrix}$$

$$\boldsymbol{X'y} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ 7 & 3 & \ldots & 4 \\ 560 & 220 & \ldots & 150 \end{bmatrix} \begin{bmatrix} 16.68 \\ 11.50 \\ \ldots \\ 10.75 \end{bmatrix} = \begin{bmatrix} 559.60 \\ 7375.44 \\ 337072.00 \end{bmatrix}$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'y}$$

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} = \begin{bmatrix} 25 & 219 & 10232 \\ 219 & 3055 & 133899 \\ 10232 & 133899 & 6725688 \end{bmatrix}^{-1} \begin{bmatrix} 559.60 \\ 7375.44 \\ 337072.00 \end{bmatrix} = \begin{bmatrix} 2.3412 \\ 1.6159 \\ 0.01438 \end{bmatrix}$$

$$\hat{y} = 2.3412 + 1.6159x_1 + 0.01438x_2$$

Python code:

```
A2 = np.concatenate((np.ones((xm.shape[0], 1)), xm), axis=1)
# turn y into a column vector
y2 = y2[:, np.newaxis]
```

```
alpha2 = np.dot((np.dot(np.linalg.inv(np.dot(A2.T,A2)),A2.T)),y2)
print("Coefficients:\nIntercept: ", alpha2[0][0],"\nNumber of cases: ", alpha2[1][0],
      "\nDistance: ", alpha2[2][0])
```

```
Python output:

Coefficients:
Intercept:  2.3412311451921566
Number of cases:  1.6159072106092514
Distance:  0.014384826255548131
```

2.1.3 Multiple Linear Regression Geometrical Interpretation of Least Squares

When we use the method of least squares to interpret multiple linear regression (MLR) in a geometric sense, we aim to understand how the regression line fits the data points in multidimensional space. In MLR, each independent variable represents a dimension in space, and the dependent variable is represented as the outcome along a particular axis.
To determine the best-fitting plane (or hyperplane in higher dimensions), our goal is to minimize the squared vertical distances between the observed data points and the plane. We can visualize this process as fitting a plane through a cloud of data points in a multidimensional space. The fitted plane should minimize the sum of the squared vertical distances (residuals) from each data point to the plane.

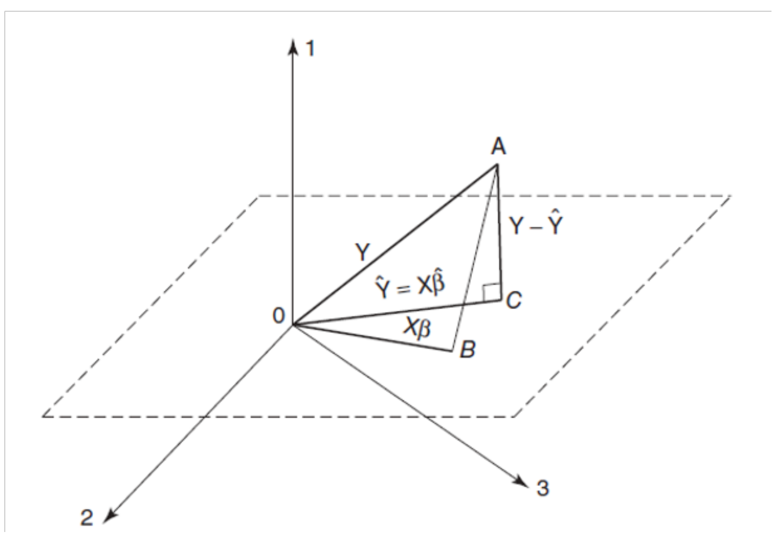

Figure 4: Geometrical Interpretation of Least Square

The coefficients of the plane are determined by the method of least squares, which minimizes the sum of the squared vertical distances, equivalent to minimizing the sum of the squared residuals. This ensures that the plane is positioned to best represent the overall trend in the data.

Overall, interpreting MLR geometrically with the least squares method involves finding the plane that best fits the data points in a multidimensional space by minimizing the sum of squared vertical distances between the observed data points and the plane. This approach allows us to understand and quantify the relationship between multiple independent variables and a dependent variable in a geometric context.

2.1.4 Multiple Linear Regression Special Case of Orthogonal Columns

Consider the model

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon} = \boldsymbol{X_1\beta_1} + \boldsymbol{X_2\beta_2} + \boldsymbol{\epsilon} \tag{28}$$

The extra-sum-of-squares method allows us to measure the effect of the regressors in $\boldsymbol{X_2}$conditional on those in $\boldsymbol{X_1}$ by computing $\boldsymbol{SS_R(\beta_2|\beta_1)}$. In general, we cannot talk about finding the sum of squares due to $\boldsymbol{\beta_2, SS_R(\beta_2)}$, without accounting for the dependence of this quantity on the regressors in $\boldsymbol{X_1}$. However, if the columns in $\boldsymbol{X_1}$ are orthogonal to the columns in $\boldsymbol{X_2}$, we can determine a sum of squares due to $\boldsymbol{\beta_2}$ that is free of any dependence on the regressors in $\boldsymbol{X_1}$.
To demonstrate this, form the normal equations $\boldsymbol{(X'X)\hat{\beta} = X'y}$ for the model. Moreover, the normal equations can be decomposed to be

$$\left[\begin{array}{c|c} \mathbf{X}_1'\mathbf{X}_1 & \mathbf{X}_1'\mathbf{X}_2 \\ \hline \mathbf{X}_2'\mathbf{X}_1 & \mathbf{X}_2'\mathbf{X}_2 \end{array}\right]\left[\begin{array}{c} \hat{\boldsymbol{\beta}}_1 \\ \hline \hat{\boldsymbol{\beta}}_2 \end{array}\right] = \left[\begin{array}{c} \mathbf{X}_1'\mathbf{y} \\ \hline \mathbf{X}_2'\mathbf{y} \end{array}\right]$$

Now if the columns of $\boldsymbol{X_1}$ are orthogonal to the columns in $\boldsymbol{X_2}$, $\boldsymbol{X_1X_2 = 0}$ and $\boldsymbol{X_2'X_1 = 0}$. Then the normal equations become

$$\boldsymbol{X_1X_1\beta_1 = X_1y}, \quad \boldsymbol{X_2X_2\beta_2 = X_2y} \tag{29}$$

with solution

$$\boldsymbol{\hat{\beta}_1 = (X_1'X_1)^{-1}X_1'y}, \quad \boldsymbol{\hat{\beta}_2 = (X_2'X_2)^{-1}X_2'y} \tag{30}$$

Note that the least-squares estimator of $\boldsymbol{\beta_1}$ is $\boldsymbol{\hat{\beta}_1}$ regardless of whether or not X_2 is in the model, and the least-squares estimator of $\boldsymbol{\beta_2}$ is $\boldsymbol{\hat{\beta}_2}$ regardless of whether or not X_1 is in the model.

2.2 Generalized Inverse

Generalized inverses, also known as pseudo inverses, are important when working with unbalanced data which implies a matrix that is not full rank. When a matrix is not full rank, the inverse is impossible to find. However, a generalized inverse gives the analyst a way to still calculate problems when the data is not of full rank.

For those full-rank matrices $\boldsymbol{X}$, the estimators are minimized in this form

$$\boldsymbol{\hat{\beta}} = \boldsymbol{(X'X)^{-1}X'y} \tag{31}$$

When $\boldsymbol{X}$ is not full rank, the least squares estimators take the form

$$\hat{\boldsymbol{\beta}} = \boldsymbol{G}\boldsymbol{X}'\boldsymbol{y} \tag{32}$$

where $\boldsymbol{G}$ is the generalized inverse of $\boldsymbol{X}'\boldsymbol{X}$. It is important for the reader to know that there are infinitely many generalized inverses depending on the route decided. Three ways to calculate this generalized inverse will be discussed in this section. The first uses a reduced matrix approach, the second uses the matrix diagonalization method, and the third uses singular value decomposition.

Generalize inverse of a matrix $\boldsymbol{X}$ is any matrix $\boldsymbol{G}$ that satisfies the equation

$$\boldsymbol{X}\boldsymbol{G}\boldsymbol{X} = \boldsymbol{X} \tag{33}$$

The equivalent diagonal form of $\boldsymbol{X}$ first needs to be defined. If $\boldsymbol{X}$ has order $p \times q$, the reduced form can be written as

$$\boldsymbol{P}_{p\times p}\boldsymbol{X}_{p\times q}\boldsymbol{Q}_{q\times q} = \boldsymbol{\Delta}_{p\times q} \equiv \begin{bmatrix} \boldsymbol{D}_{r\times r} & \boldsymbol{0}_{r\times(q-r)} \\ \boldsymbol{0}_{(p-r)\times r} & \boldsymbol{0}_{(p-r)\times(q-r)} \end{bmatrix} \tag{34}$$

$\boldsymbol{P}$ and $\boldsymbol{Q}$ are products of elementary operators, the matrix $\boldsymbol{X}$ has rank r, $\boldsymbol{D}_r$ is a diagonal matrix of order r, and the $\boldsymbol{0}$ represents a null matrix. $\boldsymbol{G}$ is easily determined from the $\boldsymbol{\Delta}$ matrix as outlined below.

$$\boldsymbol{\Delta}^{-1} = \begin{bmatrix} \boldsymbol{D}_r^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{35}$$

$$\boldsymbol{G} = \boldsymbol{Q}\boldsymbol{\Delta}^{-1}\boldsymbol{P} \tag{36}$$

Note that

$$\boldsymbol{\Delta}\boldsymbol{\Delta}^{-1}\boldsymbol{\Delta} = \boldsymbol{\Delta} \tag{37}$$

It can be said that $\boldsymbol{\Delta}^{-1}$ is a generalized inverse of $\boldsymbol{\Delta}$. This helps to solidify the idea that $\boldsymbol{G}$ is a generalized inverse of $\boldsymbol{X}$ since it can be observed that

$$\boldsymbol{X} = \boldsymbol{P}^{-1}\boldsymbol{\Delta}\boldsymbol{Q}^{-1} \tag{38}$$

Since $\boldsymbol{P}$ and $\boldsymbol{Q}$ are products of elementary matrices, this implies that their inverses do exist as well. Putting everything together, it can be concluded that, indeed, $\boldsymbol{G}$ is a generalized inverse of $\boldsymbol{X}$.

$$\boldsymbol{X}\boldsymbol{G}\boldsymbol{X} = \boldsymbol{P}^{-1}\boldsymbol{\Delta}\boldsymbol{Q}^{-1}\boldsymbol{Q}\boldsymbol{\Delta}^{-1}\boldsymbol{P}\boldsymbol{P}^{-1}\boldsymbol{\Delta}\boldsymbol{Q}^{-1} = \boldsymbol{P}^{-1}\boldsymbol{\Delta}\boldsymbol{\Delta}^{-1}\boldsymbol{\Delta}\boldsymbol{Q}^{-1} = \boldsymbol{P}^{-1}\boldsymbol{\Delta}\boldsymbol{Q}^{-1} = \boldsymbol{X} \tag{39}$$

Generalized Inverse Example Using Reduced Matrix

Begin by analyzing the matrix $\boldsymbol{X}$ and noticing that it is not of full rank, specifically, rank of 2.

$$\boldsymbol{X} = \begin{bmatrix} 1 & 2 & 5 & 2 \\ 3 & 7 & 12 & 4 \\ 0 & 1 & -3 & -2 \end{bmatrix}$$

Due to $\boldsymbol{X}$ being rank 2, choose any 2×2 submatrix of $\boldsymbol{X}$. Say matrix $\boldsymbol{M}$,

$$M = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix}$$

From here, take the inverse and the transpose of your selected submatrix $\boldsymbol{M}$

$$\boldsymbol{M} = \begin{bmatrix} 1 & 2 \\ 3 & 7 \end{bmatrix} \implies \boldsymbol{M}^{-1} = \begin{bmatrix} 7 & -2 \\ -3 & 1 \end{bmatrix} \implies (\boldsymbol{M}^{-1})' = \begin{bmatrix} 7 & -3 \\ -2 & 1 \end{bmatrix}$$

Define a matrix $\boldsymbol{H}$ as such

$$\boldsymbol{H} = \begin{bmatrix} 7 & -3 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

This will imply that the generalized inverse $\boldsymbol{G}$ is

$$\boldsymbol{G} = \boldsymbol{H}' = \begin{bmatrix} 7 & -2 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Generalized Inverse Example Using Matrix Diagonalization

For

$$\boldsymbol{X} = \begin{bmatrix} 4 & 1 & 2 & 0 \\ 1 & 1 & 5 & 15 \\ 3 & 1 & 3 & 5 \end{bmatrix}$$

The diagonal form $\boldsymbol{\Delta}$ is made using $\boldsymbol{P}$ and $\boldsymbol{Q}$

$$\boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 0 \\ -\frac{2}{3} & -\frac{1}{3} & 1 \end{bmatrix}$$

and

$$\boldsymbol{Q} = \begin{bmatrix} 1 & -1 & 1 & 5 \\ 0 & 1 & -6 & -20 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{PXQ} = \boldsymbol{\Delta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\boldsymbol{\Delta}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{1}{3} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{G} = \boldsymbol{Q\Delta P^{-1}} = \begin{bmatrix} \frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & \frac{4}{3} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

It is also possible to obtain the generalized inverse using singular value decomposition (SVD). Let $\boldsymbol{X}$ be a matrix of rank r. Let $\boldsymbol{\Lambda}$ be $r \times r$ the diagonal matrix of the ordered, non-zero eigenvalues of $\boldsymbol{X'X}$ and $\boldsymbol{XX'}$ from largest to smallest. If the eigenvalues of $\boldsymbol{X'X}$ and $\boldsymbol{XX'}$ are the same, the decomposition is

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{S}' & \boldsymbol{T}' \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}^{1/2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}' \\ \boldsymbol{V}' \end{bmatrix} = \boldsymbol{S'\Lambda^{1/2}U'} \tag{40}$$

where $\begin{bmatrix} \boldsymbol{S}' & \boldsymbol{T}' \end{bmatrix}$ and $\begin{bmatrix} \boldsymbol{U} & \boldsymbol{V} \end{bmatrix}$ are orthogonal matrices in the SVD. Notice that $\boldsymbol{S'S} + \boldsymbol{T'T} = \boldsymbol{I}$, $\boldsymbol{UU'} + \boldsymbol{VV'} = \boldsymbol{I}$, $\boldsymbol{SS'} = \boldsymbol{I}$, $\boldsymbol{TT'} = \boldsymbol{I}$, $\boldsymbol{S'T} = \boldsymbol{0}$, $\boldsymbol{T'S} = \boldsymbol{0}$, $\boldsymbol{UU'} = \boldsymbol{I}$, $\boldsymbol{U'V} = \boldsymbol{0}$, and $\boldsymbol{V'U} = \boldsymbol{0}$. Additionally, $\boldsymbol{X'X} = \boldsymbol{U\Lambda U'}$ and $\boldsymbol{XX'} = \boldsymbol{S'\Lambda S}$. The generalized inverse of $\boldsymbol{X}$ is then

$$\boldsymbol{G} = \boldsymbol{U\Lambda^{1/2}S} \tag{41}$$

<u>Generalized Inverse Example Using Singular Value Decomposition</u>

Define

$$\boldsymbol{X} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

then

$$\boldsymbol{X'X} = \begin{bmatrix} 4 & 2 & 2 \\ 2 & 2 & 0 \\ 2 & 0 & 2 \end{bmatrix}$$

and

$$\boldsymbol{XX'} = \begin{bmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \end{bmatrix}$$

To find the eigenvalues of $\boldsymbol{X'X}$ and $\boldsymbol{X'X}$, solve the equations $det(\boldsymbol{X'X}) = 0$ and $det(\boldsymbol{X'X}) = 0$. The corresponding eigenvalues are $\lambda = 6, 2, 0$ and $\lambda = 6, 2, 0, 0$, respectively. This results in the following eigenvectors and $\begin{bmatrix}\boldsymbol{S}' & \boldsymbol{T}'\end{bmatrix}$, $\begin{bmatrix}\boldsymbol{U} & \boldsymbol{V}\end{bmatrix}$ matrices.

$$\begin{bmatrix}\boldsymbol{U} & \boldsymbol{V}\end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{6}} & 0 & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{bmatrix}$$

$$\begin{bmatrix}\boldsymbol{S}' & \boldsymbol{T}'\end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{2} & -\frac{1}{2} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

Using this information, the singular value decomposition for $\boldsymbol{X}$ is

$$\boldsymbol{X} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{2} & -\frac{1}{2} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} \sqrt{6} & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix}$$

$$\boldsymbol{S}' = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

$$\boldsymbol{\Lambda^{1/2}} = \begin{bmatrix} \frac{1}{\sqrt{6}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\boldsymbol{U}' = \begin{bmatrix} \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

The generalized inverse can then be calculated

$$\boldsymbol{G} = \begin{bmatrix} \frac{2}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{3} & -\frac{1}{6} & -\frac{1}{6} \\ -\frac{1}{6} & -\frac{1}{6} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

2.3 Generalized Least Squares

One of the main assumptions needed to fit a linear regression model is that there are constant error variances. How would the estimators be found in the case of non-constant error variances? A weighted least squares method is needed where the deviation between the observed and the expected values of y_i is multiplied by a weight w_i chosen inversely proportional to the variance of y_i. For the simple linear regression case, the weighted least squares function is

$$S(\beta_0, \beta_1) = w_i \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 \tag{42}$$

with resulting least squares normal equations

$$\hat{\beta}_0 \sum_{i=1}^{n} w_i + \hat{\beta}_1 \sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} w_i y_i \tag{43}$$

$$\hat{\beta}_0 \sum_{i=1}^{n} w_i x_i + \hat{\beta}_1 \sum_{i=1}^{n} w_i x_i^2 = \sum_{i=1}^{n} w_i x_i y_i \tag{44}$$

Solve for β_0 and β_1 to get the estimate equations. In this paper, the focus will be heavy on the more general case. When the model is $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$ while $E(\boldsymbol{\epsilon}) = \boldsymbol{0}$ and $Var(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{V}$ where $\boldsymbol{V}$ is a known $n \times n$ matrix. With these conditions, the ordinary least squares estimator $\boldsymbol{\hat{\beta} = (X'X)^{-1}X'y}$ no longer works. The first step is to transform the data such that the new observations satisfy the least squares assumptions, from there, the ordinary least squares method is applied. $\boldsymbol{V}$ is a nonsingular, positive definite matrix due to $\sigma^2 \boldsymbol{V}$ representing the covariance matrix of errors. Because of this, there exists an $n \times n$ matrix $\boldsymbol{K}$ that is nonsingular and symmetric, where $\boldsymbol{K'K = KK' = V}$. Note that $\boldsymbol{K}$ is usually the $\sqrt{\boldsymbol{V}}$.

Let new variables be defined as

$$\boldsymbol{z} = \boldsymbol{K^{-1}y} \tag{45}$$

$$\boldsymbol{B} = \boldsymbol{K^{-1}X} \tag{46}$$

$$\boldsymbol{g} = \boldsymbol{K^{-1}\epsilon} \tag{47}$$

Replacing the appropriate symbol from the regression model with those for generalized least squares, the regression equation now becomes

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon} \implies \boldsymbol{K^{-1}y} = \boldsymbol{K^{-1}X\beta} + \boldsymbol{K^{-1}\epsilon} \tag{48}$$

or simply

$$\boldsymbol{z} = \boldsymbol{B\beta} + \boldsymbol{g} \tag{49}$$

It can be left up to the reader to realize that the elements of the covariance matrix, $\boldsymbol{g}$, has mean 0, constant variance, and are uncorrelated. Since $\boldsymbol{g}$ satisfies the ordinary least squares assumptions, the least squares function is

$$S(\boldsymbol{\beta}) = \boldsymbol{g'g} = \boldsymbol{\epsilon' V^{-1} \epsilon} = (\boldsymbol{y} - \boldsymbol{X\beta})' \boldsymbol{V^{-1}} (\boldsymbol{y} - \boldsymbol{X\beta}) \tag{50}$$

The least squares normal equations are

$$(\boldsymbol{X'V^{-1}X})\boldsymbol{\hat{\beta}} = \boldsymbol{X'V^{-1}y} \tag{51}$$

with the solution

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X'V^{-1}X})^{-1}\boldsymbol{X'V^{-1}y} \tag{52}$$

where $\hat{\boldsymbol{\beta}}$ is the generalized least-squares estimator of $\boldsymbol{\beta}$

Generalized Least Squares Apple Time Series Example

Suppose a businessman wants to analyze the adjusted stock price of Apple stock. They analyzed 15 days and recorded the open price (x_1), the close price (x_2), how many stocks were bought or sold (x_3), and the adjusted close price based on the volume, open, and close price (y) for each day. It is important to note that stock prices are time-related which breaks the ordinary least squares linear regression assumption of independent observations/residuals.

Apple Open (x_1)	Apple Close (x_2)	Apple Volume (x_3)	Adj_Close (y)
42.763	42.357	83973600	40.830
42.525	42.722	89134000	41.181
42.847	42.545	89960800	41.010
42.428	42.700	87342800	41.160
42.813	42.605	98507200	41.068
42.428	42.732	75891200	41.191
42.798	43.007	104457600	41.456
42.950	42.765	68998800	41.222
42.895	43.243	75652800	41.683
43.540	43.557	87493600	41.986
43.428	43.583	68280800	42.010
43.303	43.717	111341600	42.140
43.580	43.287	112861600	41.726
43.570	43.743	103544800	42.165
43.923	43.963	109744800	42.377

Figure 5: Full delivery time data set with number of cases and distances traveled.

$$
\boldsymbol{X} = \begin{bmatrix}
1 & 42.763 & 42.357 & 83973600 \\
1 & 42.525 & 42.722 & 89134000 \\
1 & 42.847 & 42.545 & 89960800 \\
1 & 42.428 & 42.700 & 87342800 \\
1 & 42.813 & 42.605 & 98507200 \\
1 & 42.428 & 42.732 & 75891200 \\
1 & 42.798 & 43.007 & 104457600 \\
1 & 42.950 & 42.765 & 68998800 \\
1 & 42.895 & 43.243 & 75652800 \\
1 & 43.540 & 43.557 & 87493600 \\
1 & 43.428 & 43.583 & 68280800 \\
1 & 43.303 & 42.717 & 111341600 \\
1 & 43.580 & 43.287 & 112861600 \\
1 & 43.570 & 43.743 & 103544800 \\
1 & 43.923 & 43.963 & 109744800
\end{bmatrix}
\qquad
\boldsymbol{y} = \begin{bmatrix}
42.357 \\ 42.722 \\ 42.545 \\ 42.700 \\ 42.605 \\ 42.732 \\ 43.007 \\ 42.765 \\ 43.243 \\ 43.557 \\ 43.583 \\ 43.717 \\ 43.287 \\ 43.743 \\ 43.963
\end{bmatrix}
$$

Python code:

```
glsApple = df.to_numpy()
glsA = glsApple[:,0:3]
glsA = np.concatenate((np.ones((glsA.shape[0], 1)), glsA), axis=1)
glsb = glsApple[:,3]


def GLS(A,b):
    Xt = A.T

    XtX = Xt @ A
    H = A @ np.linalg.inv(XtX) @ Xt
    V = np.eye(H.shape[0]) - H
    V_inv = np.linalg.inv(V)

    eigenvalues, eigenvectors = np.linalg.eig(V)

    # Take square root of eigenvalues
    sqrt_eigenvalues = np.sqrt(np.around(eigenvalues, decimals = 0))
    sqrt_eigenvalues[np.isnan(sqrt_eigenvalues)] = 0

    new_beta = np.linalg.inv(Xt @ V_inv @ A)@ Xt @ V_inv @ glsb

    return new_beta

result = GLS(glsA,glsb)
print("Coefficients:\nIntercept: ", result[0],"\nOpen: ",result[1],"\nClose: ",
      result[2],"\nVolume: ", result[3])
```

Python output:

Norm Type	Kappa
1-norm	2131162740.203
2-norm	523926887.709
Inf-norm	1437589964.344

Figure 6: Kappa by using different norm in Apple Stock Example

```
Coefficients:
Intercept:  -1.8061331086617154
Open:  0.0010493428342614798
Close:  1.004517218418194
Volume:  -3.238861507657687e-09
```

3 Condition Number

When working with least squares to produce a linear model, the columns of $\boldsymbol{X}$ should not be linearly dependent. This produces a problem in the computation process. When $\boldsymbol{X}$ has nearly linearly dependent, $\boldsymbol{X}$ is described as ill-conditioned. To determine the condition of the matrix, a condition number, $\kappa(\boldsymbol{X})$, is calculated using the following formula

$$\kappa(\boldsymbol{X}) = \|\boldsymbol{X}\| \left\|\boldsymbol{X}^{-1}\right\| \tag{53}$$

In specific cases, if $\|\cdot\| = \|\cdot\|_2$, then $\kappa(\boldsymbol{X}) = \frac{\sigma_1}{\sigma_m}$ where σ_m is the minimal singular value and σ_1 is the maximal singular value. If $\boldsymbol{X}$ is singular, then $\kappa(\boldsymbol{X}) = \infty$. If κ is small, $\boldsymbol{X}$ is said to be well-conditioned. If κ is large, $\boldsymbol{X}$ is said to be ill-conditioned. The meaning of "small" and "large" depends on the application, but generally "small" is determined to be $1, 10, 10^2$ and "large" is determined to be $10^6, 10^{16}$.

Condition Number for Generalized Least Squares Apple Stock Example

The $\boldsymbol{X}$ matrix is the same matrix defined in the previous example. The norms of $\boldsymbol{X}$ and $\boldsymbol{X}^{-1}$ are computed by built-in function in Python Numpy.linalg package. Furthermore, $\kappa(\boldsymbol{X})$ is computed as below:

```
Python code:

con1 = np.linalg.norm(glsA, ord = 1)
con2 = np.linalg.norm(glsA, ord = 2)
coninf = np.linalg.norm(glsA, ord = np.inf)
```

```
glsAinv = np.linalg.pinv(glsA)

con1inv = np.linalg.norm(glsAinv, ord = 1)
con2inv = np.linalg.norm(glsAinv, ord = 2)
coninfinv = np.linalg.norm(glsAinv, ord = np.inf)

print("Condition number using 1-norm: ",con1*con1inv)
print("Condition number using 2-norm: ",con2*con2inv)
print("Condition number using inf-norm: ",coninf*coninfinv)

Python output:

Condition number using 1-norm:  2131162740.2032871
Condition number using 2-norm:  523926887.7091318
Condition number using inf-norm:  1437589964.3436596
```

4 Stability

Method	**Cholesky Factorization**	**QR Factorization**	**SVD**
Complexity (flops)	$mn^2 + n^3/3$	$2mn^2 - 2n^3/3$	$2mn^2 + 11n^3$
Stability	Unstable	Stable	Numerically stable most of the time, even for ill-conditioned. Unstable when dealing with small singular values.
Advantage	Fastest	Stable even ill-conditioned	Can be used for any matrix
Disadvantage	Numerically stability is an issue if the matrix is ill-conditioned or nearly singular	May fail when X is nearly rank-deficient, or a large matrix	Expensive

5 Conclusion

In our project, we explored various techniques for estimating linear regression parameters and dealing with challenges such as singular matrices and autocorrelation in data. We began with the ordinary least squares method, which is a fundamental approach to estimating regression parameters. By using singular value decomposition (SVD), we were able to handle singular matrices efficiently, which helped us find a generalized inverse and made our estimation process more robust. Next, we explored the generalized least squares method, which helps deal with data or models exhibiting autocorrelation or multicollinearity. We used this technique to analyze Apple stock time series data and successfully mitigated the autocorrelation issue that we encountered with ordinary least squares. Even though we encountered an ill-conditioned problem, indicated by the kappa index, our code demonstrated resilience and stability, thanks to the assistance of SVD. This highlights the practical significance of using advanced mathematical tools to improve the performance and reliability of regression analysis methodologies. Overall, our project showed that it's important to use sophisticated statistical techniques to address real-world challenges in regression analysis. By combining theoretical understanding with practical application, we demonstrated the efficacy of least square methods in extracting meaningful insights from complex datasets.

6 References

[1]. Montgomery, Douglas C., Elizabeth A. Peck, and G. Geoffrey Vining. Introduction to Linear Regression Analysis. 5th ed. Hoboken, NJ: Wiley, 2012.

[2]. Searle, S. R. (Shayle R.), and Marvin H. J. Gruber. Linear Models. Hoboken, New Jersey: Wiley, 2017.

[3]. Weisberg, Sanford. Applied Linear Regression. Fourth edition. Hoboken, New Jersey: Wiley, 2014.

[4]. Madsen, Henrik, and Poul Thyregod. Introduction to General and Generalized Linear Models. First edition. Boca Raton, FL: CRC Press, an imprint of Taylor and Francis, 2010. https://doi.org/10.1201/9781439891148.

[5]. Trefethen, Lloyd N., and David Bau. Numerical linear algebra. Society for Industrial and Applied Mathematics, 2022.

[6]. Computational Linear Algebra Aspects. Differential Algebraic Equations: A Projector Based Analysis, 2013.

[7]. Zhang, Fuzhen. Matrix Theory Basic Results and Techniques. 2nd ed. 2011. New York, NY: Springer New York, 2011. https://doi.org/10.1007/978-1-4614-1099-7.

[8]. Benner, Peter., Matthias. Bollhöfer, Daniel. Kressner, Christian. Mehl, and Tatjana. Stykel, eds. Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory Festschrift in Honor of Volker Mehrmann. 1st ed. 2015. Cham: Springer International Publishing, 2015. https://doi.org/10.1007/978-3-319-15260-8.

[9]. Wagner, Kevin., and Miloš. Doroslovački. Proportionate-Type Normalized Least Mean Square Algorithms. 1st ed. London: ISTE, 2013.

[10]. Liu, Qiaohua, Qian Zhang, and Dongmei Shen. "Condition Numbers of the Mixed Least Squares-Total Least Squares Problem Revisited." Linear & Multilinear Algebra 71, no. 13 (2023): 2144–62. https://doi.org/10.1080/03081087.2022.2094861.

[11]. Hogben, Leslie. Handbook of Linear Algebra. Boca Raton: Chapman & Hall/CRC, 2007.

[12]. Boyd, Stephen P., and Lieven Vandenberghe. Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares. Cambridge; Cambridge University Press, 2018.