Lecture 09

ESTIMATION

Probability & Statistics

Francisco Richter, Martina Boschi and Ernst Wit

1 Statistical Learning

In the theoretical framework of statistical modeling, our objective is to examine the relationship between a variable y and one or more variables $\mathbf{x}$. Here, y represents the outcome we seek to predict or explain, while the vector $\mathbf{x}$ comprises the explanatory or predictor variables that inform our predictions of y.
This relationship is theoretically expressed as:

$$y = f(\mathbf{x}) + \epsilon$$

where:

- $f(\mathbf{x})$ embodies the systematic information conveyed by $\mathbf{x}$ regarding y. This function f is unknown and reflects the true underlying process that we aim to understand.
- ϵ captures the random error, accounting for variations in y not explained by $\mathbf{x}$, with an expected value of zero.

Definition 1 (Statistical Model)**.** *A statistical model is a mathematical representation of observed data. In the context of regression, we often describe the model as $Y = f(X) + \epsilon$, where Y is the dependent variable, X is the independent variable, f is the function that represents the systematic relationship between X and Y, and ϵ represents the error term, capturing all other factors affecting Y that are not included in X.*

Transitioning from this theoretical construct to empirical application, we collect a dataset with n observations. Each observation i in the dataset comprises an actual outcome y_i and the corresponding values of variables $\mathbf{x}_i$. The practical challenge in regression analysis lies in estimating a function $\hat{f}$ from the observed data that serves as a surrogate for the true function f. This estimated function $\hat{f}$ is what we use to predict new values of y based on observed $\mathbf{x}$.
Accordingly, the estimated relationship is given by:

$$y_i = \hat{f}(\mathbf{x}_i) + \epsilon_i$$

where $\hat{f}(\mathbf{x}_i)$ is the predicted value of y based on the i-th observation's values of $\mathbf{x}$, and ϵ_i is the estimation error for that observation.
Naturally, to evaluate how well the model describes the data, we can consider metrics like the Mean Squared Error (MSE), which is computed as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ represents the predicted value of the dependent variable for the i-th observation, given by the model as $\hat{y}_i = \hat{f}(\mathbf{x}_i)$.
The goal of the regression analysis is, therefore, to find the estimated function $\hat{f}$ that minimizes the MSE, reflecting the closest approximation to the true function f that generated the observed data. In linear regression, we model the relationship between a dependent variable and one or more independent variables using a linear function:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where:

- $\mathbf{y}$ is an $n \times 1$ vector containing the observed values of the dependent variable.
- $\mathbf{X}$ is an $n \times k$ matrix of the independent variables (predictors), with n observations and k predictors.

- $\boldsymbol{\beta}$ is a $k \times 1$ vector of unknown parameters (coefficients) to be estimated.
- $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of error terms, representing the differences between the observed and predicted values.

Example. Suppose we have data from a gym recording the weight lifted and the number of repetitions performed by three individuals on a particular machine:

Observation	Weight Lifted (kg)	Repetitions
1	50	15
2	60	12
3	70	9

First, we construct the dependent variable vector **y**, which contains the number of repetitions:

$$\mathbf{y} = \begin{bmatrix} 15 \\ 12 \\ 9 \end{bmatrix}$$

Next, we construct the independent variable matrix **X**. Since we are including an intercept term, the first column consists of ones, and the second column contains the weights lifted:

$$\mathbf{X} = \begin{bmatrix} 1 & 50 \\ 1 & 60 \\ 1 & 70 \end{bmatrix}$$

The parameter vector $\boldsymbol{\beta}$ contains the coefficients to be estimated:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

We also have the error term vector $\boldsymbol{\epsilon}$:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}$$

Putting it all together, our linear regression model in matrix form becomes:

$$\begin{bmatrix} 15 \\ 12 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 & 50 \\ 1 & 60 \\ 1 & 70 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}$$

Expanding the matrix equation, we obtain three linear equations:

$$\begin{cases} 15 = \beta_0 + \beta_1 \times 50 + \epsilon_1 \\ 12 = \beta_0 + \beta_1 \times 60 + \epsilon_2 \\ 9 = \beta_0 + \beta_1 \times 70 + \epsilon_3 \end{cases}$$

In this model:

- β_0 is the intercept term, representing the predicted number of repetitions when the weight lifted is zero.
- β_1 is the slope coefficient, indicating how much the number of repetitions is expected to decrease for each additional kilogram lifted.
- ϵ_i accounts for the variability in repetitions not explained by the weight lifted.

□

To estimate β_0 and β_1, we can use the Ordinary Least Squares (OLS) method, which minimizes the sum of squared residuals (the differences between observed and predicted values). The OLS estimates are calculated using:

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

By plugging in the values from our example, we can compute the estimates for β_0 and β_1, allowing us to predict the number of repetitions based on the weight lifted.
Consider the problem of estimating the parameters $\boldsymbol{\beta}$ within a linear regression framework, where the goal is to minimize the Mean Squared Error (MSE) given by:

$$\mathsf{MSE}(\boldsymbol{\beta}) = \frac{1}{n}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

To derive the OLS estimator, we look for the value of $\boldsymbol{\beta}$ that minimizes the MSE. We start by setting the gradient of the MSE with respect to $\boldsymbol{\beta}$ equal to zero:

$$\frac{\partial}{\partial \boldsymbol{\beta}}\mathsf{MSE}(\boldsymbol{\beta}) = -\frac{2}{n}\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = 0$$

Solving for $\boldsymbol{\beta}$ yields the normal equations:

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

If $\mathbf{X}'\mathbf{X}$ is invertible, the OLS estimator $\hat{\boldsymbol{\beta}}$ is given by:

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

This expression for $\hat{\boldsymbol{\beta}}$ minimizes the MSE and is known as the OLS estimator.
The residuals of the model are defined as:

$$\boldsymbol{e} = \mathbf{y} - \hat{\mathbf{y}}$$

where:

- $\mathbf{y}$ is the vector of observed values of the dependent variable.
- $\hat{\mathbf{y}}$ is the vector of predicted values obtained from the model, given by $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$.

The residuals represent the portion of the dependent variable that is not explained by the model.
Note that the sum of the residuals is zero when the model includes an intercept term. This occurs because the OLS estimation ensures that the total predicted values equal the total observed values, resulting in the residuals summing to zero:

$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n}(y_i - \hat{y}_i) = \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \hat{y}_i = 0$$

To evaluate the performance of a linear regression model, we often use the coefficient of determination, denoted as R^2. The R^2 value measures the proportion of the variance in the dependent variable that is predictable from the independent variables.
The R^2 statistic is defined as:

$$R^2 = 1 - \frac{\mathsf{SS}_{\mathsf{res}}}{\mathsf{SS}_{\mathsf{tot}}}$$

where:

- $\mathsf{SS}_{\mathsf{res}} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ is the residual sum of squares, representing the unexplained variation after fitting the model.
- $\mathsf{SS}_{\mathsf{tot}} = \sum_{i=1}^{n}(y_i - \bar{y})^2$ is the total sum of squares, representing the total variation in the dependent variable.
- $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the mean of the observed values.

The R^2 value can be interpreted as the fraction of the total variation in the dependent variable that is explained by the model:

$$R^2 = \frac{\text{Explained Variation}}{\text{Total Variation}} = \frac{\mathsf{SS}_{\mathsf{tot}} - \mathsf{SS}_{\mathsf{res}}}{\mathsf{SS}_{\mathsf{tot}}}$$

An R^2 value of 1 indicates that the regression predictions perfectly fit the data, meaning the model explains all the variability in the dependent variable. An R^2 value of 0 indicates that the model does not explain any of the variability of the response data around its mean.

Example (Regression Analysis of Gym Data with Multiple Variables)**.** We analyze the relationship between the number of repetitions performed on a "Chest Press" machine and several factors such as weight lifted, age, and gender. Using synthetic data of 100 individuals, the dataset includes the following variables:

- *Repetitions*: The number of repetitions performed.

- *Weight Level*: The weight lifted in kilograms.
- *Age*: The age of the individual (in years).
- *Gender*: The gender of the individual (Male or Female).

To examine these relationships, we fit three models: a simple linear regression, gender-specific regressions, and a multivariate regression.
We first examine the relationship between repetitions and weight level using a simple linear regression:

$$\text{Repetitions} = \beta_0 + \beta_1 \times \text{Weight Level} + \varepsilon$$

After fitting the model, we obtain the following estimates:

$$\hat{\beta}_0 = 17.56, \quad \hat{\beta}_1 = -0.091$$

The negative slope indicates that as the weight level increases, the number of repetitions decreases, with a reduction of approximately 0.091 repetitions per kilogram. The residual standard error for this model is 1.32, and the R^2 value is 0.415, indicating that 41.5% of the variability in repetitions is explained by the weight level. This relationship is visualized in Figure 1.

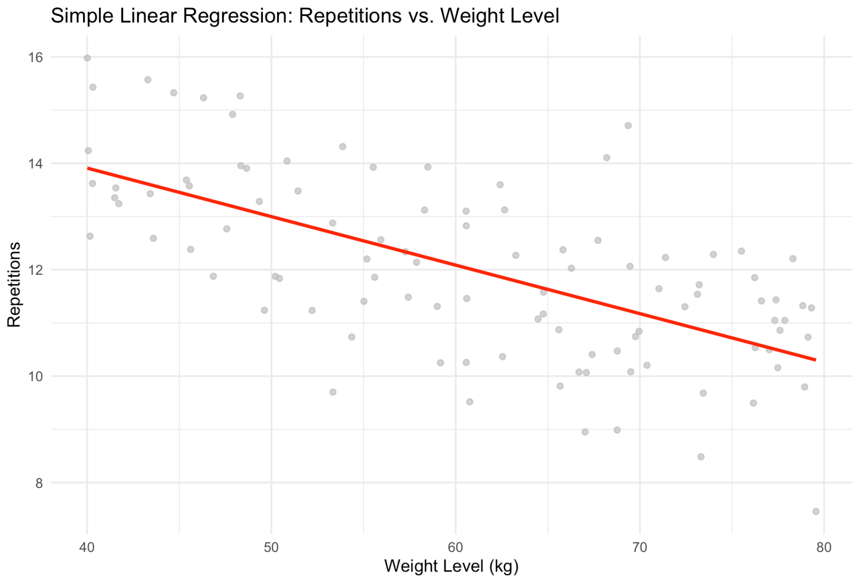

Figure 1: Simple Linear Regression: Repetitions vs. Weight Level. The regression line highlights the negative association between weight level and repetitions.

To account for potential differences between genders, we fit separate linear regression models for males and females:

$$\text{Repetitions} = \beta_0 + \beta_1 \times \text{Weight Level} + \varepsilon$$

For males, the regression estimates are:

$$\hat{\beta}_0 = 16.75, \quad \hat{\beta}_1 = -0.091, \quad \text{MSE} = 1.03, \quad R^2 = 0.528$$

For females, the regression estimates are:

$$\hat{\beta}_0 = 19.74, \quad \hat{\beta}_1 = -0.111, \quad \text{MSE} = 0.775, \quad R^2 = 0.696$$

The results suggest that while weight level negatively affects repetitions for both genders, the effect is slightly stronger for females. Additionally, the R^2 value for females (0.696) is higher than for males (0.528), indicating that weight level explains a greater proportion of the variance in repetitions for females. These gender-specific relationships are visualized in Figure 2.
Finally, we incorporate additional predictors, such as age and gender, into a multivariate regression model:

$$\text{Repetitions} = \beta_0 + \beta_1 \times \text{Weight Level} + \beta_2 \times \text{Age} + \beta_3 \times \text{Gender (Male)} + \varepsilon$$

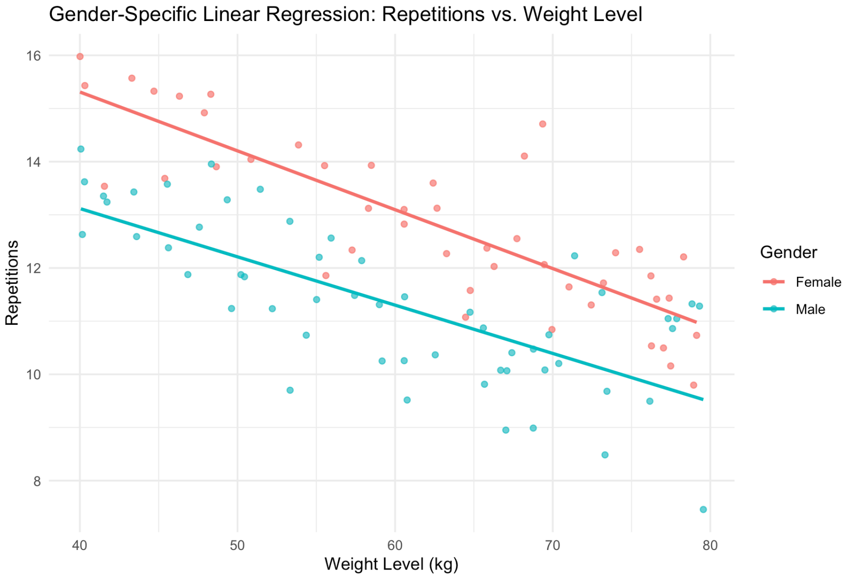

Figure 2: Gender-Specific Linear Regression: Repetitions vs. Weight Level. The blue line represents the male model, while the green line represents the female model.

Here, Gender (Male) is a binary variable equal to 1 for males and 0 for females. The estimated coefficients for the multivariate regression are:

$$\hat{\beta}_0 = 20.11, \quad \hat{\beta}_1 = -0.101, \quad \hat{\beta}_2 = -0.024, \quad \hat{\beta}_3 = -1.814$$

These results indicate that:

- *Weight Level*: Higher weights are associated with fewer repetitions (-0.101 repetitions per kilogram).
- *Age*: Older individuals perform slightly fewer repetitions (-0.024 repetitions per year).
- *Gender*: Males perform approximately 1.814 fewer repetitions than females, holding other factors constant.

The residual standard error for this model is 0.941, and the R^2 value is 0.706, meaning that 70.6% of the variability in repetitions is explained by weight level, age, and gender. The predictions from the multivariate regression model are visualized in Figure 3.

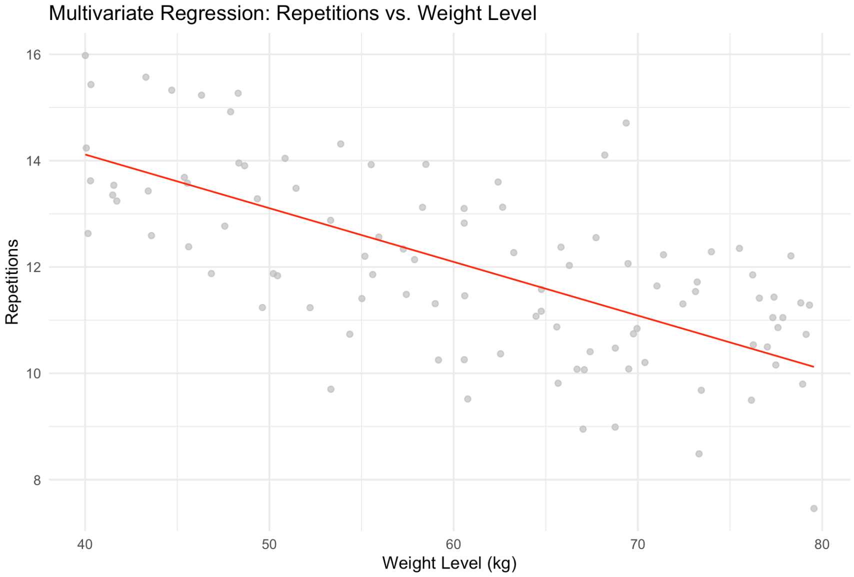

Figure 3: Multivariate Regression: Repetitions vs. Weight Level. The regression line is shown in red, indicating the combined effect of weight level, age, and gender.

To evaluate and compare the models, we summarize the Mean Squared Error (MSE) and R^2 values in Table 1.

Model	**MSE**	**R²**
Simple Linear Regression	1.696	0.415
Male Regression	1.033	0.528
Female Regression	0.775	0.696
Multivariate Regression	0.851	0.706

Table 1: Model Performance Comparison. The multivariate model achieves the highest R^2 value, indicating the best fit.

The multivariate model provides the best fit ($R^2 = 0.706$), demonstrating the importance of including additional predictors such as age and gender. Gender-specific models also perform well, particularly for females, where weight level explains a significant proportion of the variance in repetitions. These results underscore the value of tailoring regression models to capture the effects of relevant predictors, which can help gym trainers and managers design personalized training programs.

□

A high R^2 does not necessarily indicate that a regression model is appropriate, as it fails to account for overfitting. Overfitting occurs when a model captures noise or random variation in the data rather than meaningful patterns. This happens because adding predictors reduces the residual sum of squares (SS$_{\text{residual}}$) by solving the optimization problem:

$$\min_{\beta_0,\beta_1,\dots,\beta_p} \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})\right)^2 .$$

As the number of predictors increases, the model has more flexibility to fit the data, causing R^2 to always increase or remain constant, even if the additional predictors do not contribute meaningful information. This property makes R^2 prone to favoring overly complex models.

To address this limitation, the adjusted R^2 statistic is often used, which introduces a penalty for adding predictors. It is defined as:

$$R^2_{\text{adj}} = 1 - \left(\frac{\text{SS}_{\text{residual}}/(n-p-1)}{\text{SS}_{\text{total}}/(n-1)}\right),$$

where n is the number of observations and p is the number of predictors. Unlike R^2, adjusted R^2 can decrease when irrelevant predictors are added, making it a more robust measure for comparing models. While R^2 is useful for assessing the proportion of variance explained, adjusted R^2 provides a better balance between model fit and complexity.

Balancing the bias and variance is crucial in minimizing the MSE. Simpler models tend to have high bias but low variance, while complex models can have low bias but high variance. The irreducible error σ_ϵ^2 is independent of the model and sets a lower bound on the achievable MSE.

To analyze the sources of error in the Mean Squared Error (MSE), we assume the observed outcomes y_i can be expressed as:

$$y_i = f(x_i) + \epsilon_i,$$

where $f(x_i)$ is the true underlying function and ϵ_i is a noise term with $\mathbb{E}[\epsilon_i] = 0$ and variance σ_ϵ^2. Substituting this into the squared error term yields:

$$(y_i - \hat{y}_i)^2 = (f(x_i) + \epsilon_i - \hat{f}(x_i))^2.$$

Expanding the square, we get:

$$(y_i - \hat{y}_i)^2 = (f(x_i) - \hat{f}(x_i))^2 + 2\epsilon_i(f(x_i) - \hat{f}(x_i)) + \epsilon_i^2.$$

Taking the expectation over the data, and noting that $\mathbb{E}[\epsilon_i] = 0$, the cross-product term vanishes, simplifying to:

$$\mathbb{E}\left[(y_i - \hat{y}_i)^2\right] = \mathbb{E}\left[(f(x_i) - \hat{f}(x_i))^2\right] + \mathbb{E}[\epsilon_i^2].$$

The first term, $\mathbb{E}\left[(f(x_i) - \hat{f}(x_i))^2\right]$, captures the model error and can be decomposed further into bias and variance:

$$\mathbb{E}\left[(f(x_i) - \hat{f}(x_i))^2\right] = \text{Bias}(\hat{f}(x_i))^2 + \text{Variance}(\hat{f}(x_i)),$$

where:

$$\text{Bias}(\hat{f}(x_i))^2 = \left(\mathbb{E}[\hat{f}(x_i)] - f(x_i)\right)^2, \quad \text{Variance}(\hat{f}(x_i)) = \mathbb{E}[\hat{f}(x_i)^2] - (\mathbb{E}[\hat{f}(x_i)])^2.$$

The second term, $\mathbb{E}[\epsilon_i^2]$, represents the irreducible error, which is due to the noise ϵ inherent in the data:

$$\text{Irreducible Error} = \sigma_\epsilon^2.$$

Combining these components, the expected MSE can be written as:

$$\mathbb{E}\left[(y_i - \hat{y}_i)^2\right] = \text{Bias}(\hat{f}(x_i))^2 + \text{Variance}(\hat{f}(x_i)) + \sigma_\epsilon^2.$$

This decomposition illustrates that the total error has three components: bias, variance, and irreducible error. Reducing MSE involves balancing bias and variance, as simplifying a model increases bias while reducing variance, and overly complex models increase variance but reduce bias. The irreducible error σ_ϵ^2 cannot be reduced by any model, as it reflects inherent randomness in the data.

2 Bootstrap

Bootstrap resampling is a non-parametric method used to estimate the sampling distribution of a statistic. It is particularly valuable when the theoretical distribution of the statistic is unknown or difficult to derive. This technique involves repeatedly resampling the data with replacement to generate a large number of bootstrap samples, each of the same size as the original dataset. By calculating the statistic of interest for each bootstrap sample, we can approximate its sampling distribution and compute metrics such as standard errors, confidence intervals, and bias estimates.

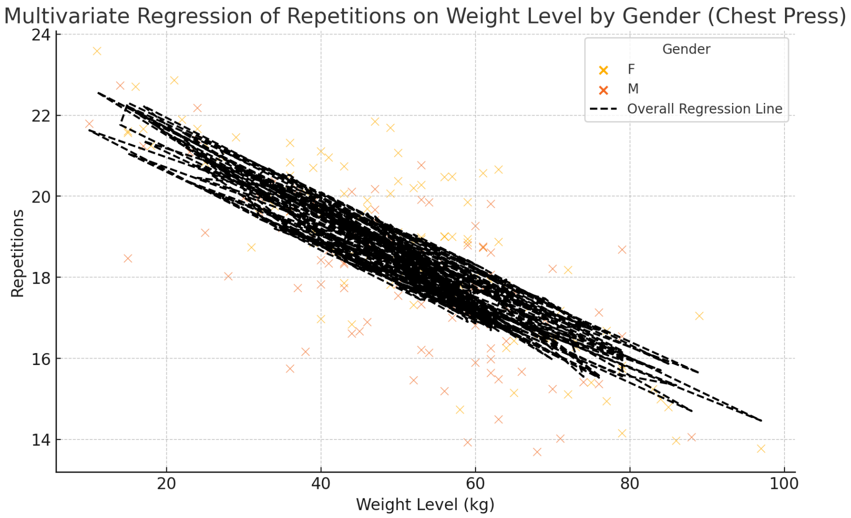

Figure 4: Bootstrap Distribution of MSE for Multivariate Model. The blue line indicates the observed model MSE, while the red and green lines represent the 95% confidence interval bounds.

Given an original dataset of size n, the bootstrap procedure is as follows:

1. Randomly draw n observations with replacement from the dataset to create a bootstrap sample X_b^*, where b indexes the bootstrap sample.
2. Compute the statistic of interest θ_b^* for the bootstrap sample:

   $$\theta_b^* = s(X_b^*), \quad b = 1, 2, \ldots, B$$

   Here, $s(\cdot)$ represents the function used to compute the statistic, such as the mean, median, or MSE, and B is the total number of bootstrap samples (commonly 1000 or more).
3. Use the B bootstrap replicates $\{\theta_1^*, \theta_2^*, \ldots, \theta_B^*\}$ to estimate the sampling distribution of θ.

This approach has several advantages:

- It is simple to implement and does not require strong distributional assumptions about the data.

- It can be applied to virtually any statistic, including those for which analytical derivations are challenging.
- It provides a direct way to compute measures of uncertainty such as standard errors and confidence intervals.

To illustrate, consider estimating the standard error of a statistic θ using bootstrap replicates. The bootstrap standard error (SE) is given by:

$$SE(\theta) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\theta_b^* - \overline{\theta^*}\right)^2}$$

where $\overline{\theta^*}$ is the mean of the bootstrap replicates.
Confidence intervals for θ can also be constructed from the bootstrap replicates. For a $(1-\alpha) \times 100\%$ confidence interval, the bootstrap percentiles are used:

$$CI_{(1-\alpha)\times 100\%} = \left(\theta^*_{(\alpha/2)}, \theta^*_{(1-\alpha/2)}\right)$$

where $\theta^*_{(\alpha/2)}$ and $\theta^*_{(1-\alpha/2)}$ are the lower and upper percentiles of the bootstrap replicates.

Example: Bootstrap Estimation of MSE for a Multivariate Model In our analysis of gym data, we applied the bootstrap method to estimate the sampling distribution of the Mean Squared Error (MSE) for the multivariate regression model. The original dataset consists of 100 observations, and the multivariate model predicts repetitions based on weight level, age, and gender.
Using $B = 1000$ bootstrap samples, we computed the MSE for each sample to generate a bootstrap distribution. From this distribution, we calculated the standard error and constructed a 95% confidence interval. The results are summarized in Table 2.

Metric	Value
Multivariate Model MSE	1.71
Bootstrap MSE SE	0.17
Bootstrap 95% CI Lower	1.36
Bootstrap 95% CI Upper	2.02

Table 2: Summary of MSE and Bootstrap Estimates for Multivariate Model.

Figure 4 shows the bootstrap distribution of MSE, where the observed model MSE (1.71) is marked by a blue line, and the red and green lines represent the bounds of the 95% confidence interval.
To further interpret these results, Figure 5 illustrates the multivariate regression predictions against the original observations, highlighting gender-specific trends for repetitions as a function of weight level.
These results highlight the reliability of the multivariate model's performance. The narrow confidence interval $(1.36, 2.02)$ suggests stability across different data samples, supporting the robustness of the model's predictions. Moreover, the visualization underscores the model's ability to capture the distinct effects of weight, age, and gender on repetitions.

3 Monte Carlo Estimation

Monte Carlo methods are a broad class of algorithms that rely on random sampling to obtain numerical results. One of the most classic applications of these methods is *Monte Carlo integration*.
Imagine you wish to find the area under a curve defined by a function $f(x)$ over an interval $[a, b]$. Traditional numerical integration methods, like the trapezoidal rule or Simpson's rule, divide the interval into small sub-intervals and approximate the area under the curve using geometric shapes. In contrast, Monte Carlo integration estimates this integral by taking random samples.
The process is as follows:

1. Define a rectangle that contains the region you wish to integrate. This rectangle should span from a to b in the x-direction and from 0 to a value M which is greater than the maximum value of $f(x)$ on the interval in the y-direction.

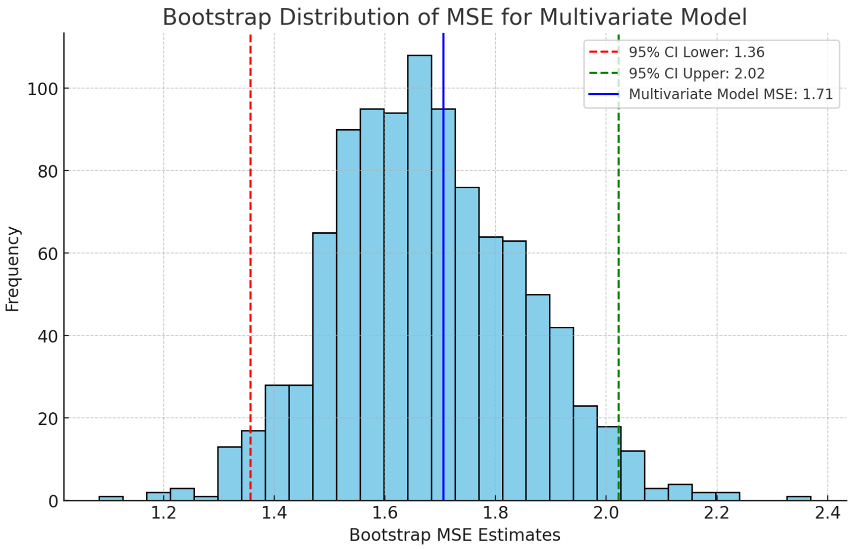

Figure 5: Multivariate Regression of Repetitions on Weight Level, Age, and Gender. The trends for males and females are shown separately, reflecting gender-specific effects in the predictions.

2. Generate a large number of random points inside this rectangle.
3. Count the number of points that fall below the curve $f(x)$.
4. The ratio of the number of points below the curve to the total number of points, multiplied by the area of the rectangle, gives an estimate of the integral.

Mathematically, the estimate I for the integral of $f(x)$ over $[a, b]$ is:

$$I = \left(\frac{\text{Number of points below } f(x)}{\text{Total number of points}} \right) \times \text{Area of rectangle}$$

The expectation of a random variable offers a theoretical average or "center of mass" for its distribution. A natural question arises: if we sample repeatedly from this distribution, how closely does the sample average approximate this theoretical expectation? The Law of Large Numbers provides an answer, guaranteeing that, under specific conditions, the sample average converges to the theoretical expectation as the number of samples increases. This law underpins the intuitive notion that as we collect more independent observations, their average tends towards the true mean of the distribution.

Theorem 2 (Law of Large Numbers)**.** *Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed (i.i.d.) random variables with a finite expectation denoted by $\mathbb{E}[X_i]$. Then, as n approaches infinity, the sample average converges to the expected value:*

$$\frac{1}{n} \sum_{i=1}^{n} X_i \xrightarrow[n \to \infty]{} \mathbb{E}[X_i].$$

In practical terms, the average outcome from a large number of trials will approximate the expected value, and this approximation improves with more trials.

Building on the Law of Large Numbers, Monte Carlo Integration approximates integrals by averaging function values at randomly chosen points.

Theorem 3 (Fundamental theorem of Monte Carlo Integration)**.** *Consider a real-valued function $f(x)$ defined over a domain D. The Monte Carlo estimate for the integral $\int_D f(x)\,dx$ is:*

$$\frac{1}{N} \sum_{i=1}^{N} f(x_i),$$

where x_i are random samples drawn uniformly from D. As N approaches infinity, and under certain conditions, this estimate converges to the true value of the integral, thanks to the Law of Large Numbers.

Monte Carlo Integration can be approached using indicator random variables, especially useful when the domain of integration, D, is complex or irregularly shaped. This method leverages random sampling and probability to provide an estimate for the integral.
The methodology is:

1. **Random Sampling**: Draw a random point (u_1, u_2) uniformly from a larger domain R that encompasses D.

2. **Indicator Variable**: Define a binary random variable I as:

$$I = \begin{cases} 1 & \text{if } u_1 \leq f(u_2) \text{ and } (u_1, u_2) \in D \\ 0 & \text{otherwise} \end{cases}$$

   This variable I is 1 if the point (u_1, u_2) lies below the curve of f within D, and 0 otherwise.

3. **Compute the Proportion**: After drawing N random points, compute the proportion $\hat{p}$ of points for which $I = 1$. This proportion estimates the ratio of the area under f in D to the area of R.

4. **Estimate the Integral**: The integral of f over D is approximately $\hat{p} \times |R|$.

The expectation of the indicator random variable I is given by:

$$\mathbb{E}[I] = P((u_1, u_2) \text{ is under } f \text{ and in } D)$$

This expectation is essentially the proportion of the area under f within D relative to R:

$$\mathbb{E}[I] = \frac{\text{Area under } f \text{ in } D}{|R|}$$

The Monte Carlo estimate for this expectation, after N trials, is:

$$\hat{p} = \frac{1}{N} \sum_{i=1}^{N} I_i$$

where I_i is the value of I for the i-th random sample. Thus, the Monte Carlo estimate for the integral of f over D becomes:

$$\hat{p} \times |R|$$

By the Law of Large Numbers, as N grows larger, $\hat{p}$ converges to $\mathbb{E}[I]$, making our integral estimate increasingly accurate.
Monte Carlo Integration can be applied to various problems, one of the most illustrative being the estimation of the mathematical constant π.

Example (Estimation of π using Monte Carlo)**.** Consider a unit circle inscribed in a unit square. If we uniformly sample random points within this square, the probability that a point lies inside the circle is equal to the ratio of the area of the circle to the area of the square. Given that the area of the unit circle is π and the area of the unit square is 1, this ratio is $\frac{\pi}{4}$.
Let's define an indicator random variable I:

$$I = \begin{cases} 1 & \text{if the point is inside the unit circle} \\ 0 & \text{otherwise} \end{cases}$$

After drawing N random points in the square, the proportion $\hat{p}$ of points for which $I = 1$ approximates the ratio of the area of the circle to the square. Therefore, an estimate of π is given by:

$$\pi \approx 4 \times \hat{p}$$

This method leverages the geometric interpretation of π and the probabilistic foundations of Monte Carlo to provide an estimate. As N grows larger, the estimate becomes more accurate due to the Law of Large Numbers.

□

Example. Consider the function $f(x) = x^2$ over the interval $[0, 1]$. The actual value of this integral is $\frac{1}{3}$. Using Monte Carlo integration, we can estimate this value.
As seen in the figure, the blue curve represents the function $f(x) = x^2$. The green dots represent the random points that fall below the curve, and the red dots represent the points that fall above the curve. Through this method, we estimated the value of the integral to be approximately 0.3349, which is close to the actual value of $\frac{1}{3} \approx 0.3333$. □

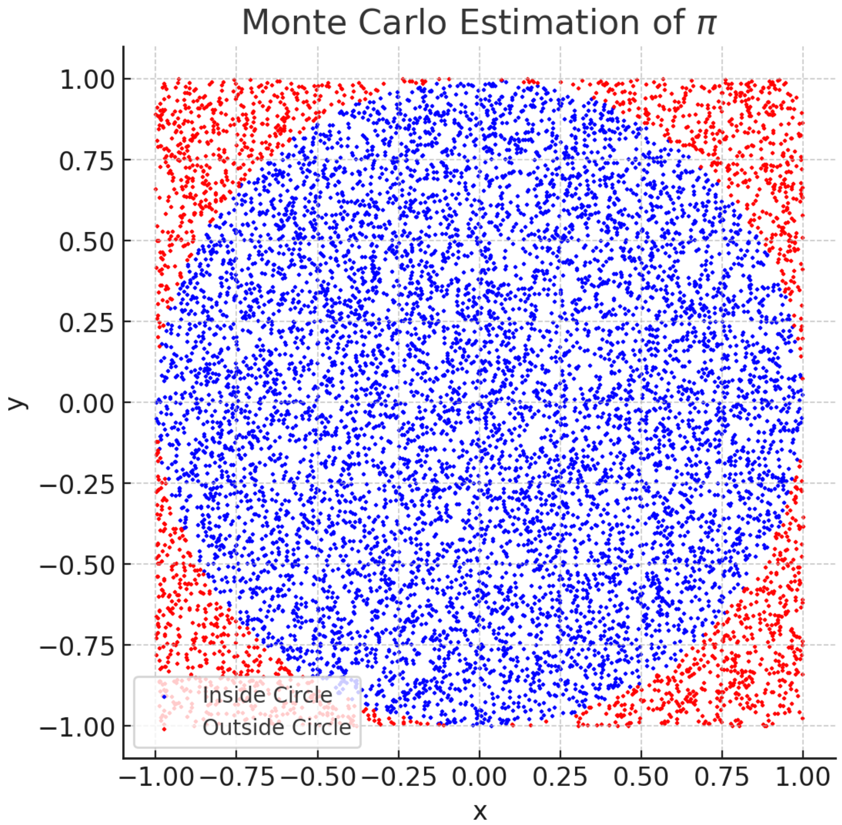

Figure 6: Monte Carlo Estimation of π. Blue dots represent random points inside the unit circle, while red dots represent points outside the circle but inside the unit square.

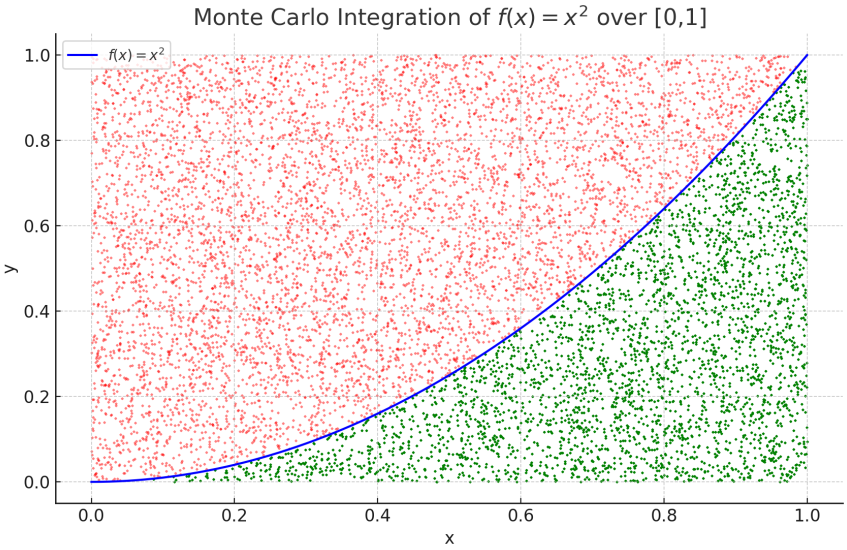

Figure 7: Monte Carlo Integration of $f(x) = x^2$ over [0,1]. Green dots represent points below the curve, and red dots represent points above the curve.

4 Stochastic Dynamics Inference

Stochastic reaction systems are dynamical processes where a number of agents, particles, or nodes interact in a stochastic way through reactions, generating different configurations for the studied system over time. These systems include applications such as protein dynamics, gene expression, ecological systems, kinetic systems, compartmental models, and actor-oriented systems, among others.

In many of these applications, statistical methods for the inference of kinetic rates dynamics are crucial. Kinetic systems can be mathematically modeled as Poisson processes, like the ones seen in lecture 6.

Before delving into the complexities of reaction systems, it's essential to understand the basics of combinatorics, which plays a critical role in these systems. Combinatorics, the branch of mathematics dealing with combinations and arrangements of objects, is fundamental in calculating reaction rates and understanding reaction dynamics.

Consider a simple scenario where we have a set of distinct objects, and we want to determine how many different ways we can arrange or select these objects. The principles of combinatorics allow us to calculate these possibilities. For example, if we have n different objects and want to choose r of them, the number of different combinations we can form is given by the binomial coefficient, denoted as $\binom{n}{r}$. This coefficient is calculated as:

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

where $n!$ (n factorial) is the product of all positive integers up to n. This combinatorial calculation is crucial in reaction systems, particularly when determining reaction rates and the number of possible interactions between different molecular species.

4.1 Generating Continuous Random Variables

Generating continuous random variables with specific probability distributions is a fundamental task in simulations and modeling. Two commonly used methods for this purpose are the *Inverse Transform Sampling* and *Rejection Sampling* techniques. Additionally, generating samples from the *Normal Distribution* is of paramount importance due to its ubiquitous presence in statistical modeling and the Central Limit Theorem. These methods rely on uniform random variables and mathematical transformations to generate samples from a desired distribution.

Inverse Transform Sampling

The inverse transform sampling method is a straightforward technique for generating random variables when the cumulative distribution function (CDF) of the desired distribution is invertible.

Theorem 4 (Inverse Transform Sampling)**.** *Let U be a uniform random variable in the interval $[0,1]$, and let $F_X^{-1}(u)$ be the inverse function of $F_X(x)$, the CDF of X. Then $X = F_X^{-1}(U)$ will have the PDF $f_X(x)$.*

Proof. Let $U \sim U(0,1)$ and $F_X(x)$ be the CDF of a random variable X. The proof follows from:

$$\begin{aligned} P(X \leq x) &= P(F_X^{-1}(U) \leq x) \\ &= P(U \leq F_X(x)) \\ &= F_U(F_X(x)) \\ &= F_X(x). \end{aligned}$$

Thus, $X = F_X^{-1}(U)$ follows the desired distribution. □

This theorem provides a practical method to simulate random variables from a given distribution by transforming a uniform random variable using the inverse CDF.

Example: Generating an Exponential Random Variable The exponential distribution, frequently used to model waiting times in Poisson processes, has the probability density function (PDF):

$$f_X(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Its CDF is:

$$F_X(x;\lambda) = \begin{cases} 1 - e^{-\lambda x} & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

To generate a random variable X with this distribution, we:

1. Draw a uniform random variable $U \sim U(0,1)$.

2. Compute the inverse CDF transformation:

$$X = F_X^{-1}(U) = -\frac{1}{\lambda}\ln(1-U).$$

This procedure ensures that X samples follow the exponential distribution. The simplicity of this method makes it highly effective when $F_X^{-1}(u)$ is analytically computable.

Rejection Sampling

When the CDF of a distribution is not invertible or the inverse is computationally expensive to calculate, rejection sampling provides an alternative approach. This method generates random samples by comparing points against the target probability density function.

Theorem 5 (Rejection Sampling with Uniform Distribution)**.** *Let $f(x)$ be a non-negative and bounded PDF defined on a domain $\mathcal{D}$ such that $0 \leq f(x) \leq M$ for all $x \in \mathcal{D}$, where M is the maximum value of $f(x)$. Samples from $f(x)$ can be generated as follows:*

1. *Generate $X \sim$ Uniform($\mathcal{D}$) and $Y \sim U(0, M)$.*
2. *If $Y \leq f(X)$, accept X as a sample from $f(x)$. Otherwise, reject X and repeat.*

Intuition and Acceptance Probability Rejection sampling works by constructing a bounding rectangle over the target distribution $f(x)$ using the uniform distribution. Accepted points (X, Y) lie below the curve $y = f(x)$, ensuring that the resulting samples follow the desired distribution. The probability of accepting a sample is proportional to the ratio of the area under $f(x)$ to the area of the bounding rectangle:

$$P(\text{Acceptance}) = \frac{\text{Area under } f(x)}{\text{Area of bounding rectangle}}.$$

Example: Rejection Sampling for a Custom Distribution Consider a distribution $f(x) \propto x^2$ on $[0, 1]$, normalized to form a valid PDF:

$$f(x) = 3x^2, \quad 0 \leq x \leq 1.$$

Using $M = 3$ (the maximum value of $f(x)$) and a uniform proposal distribution over $[0, 1]$, the rejection sampling steps are:

1. Generate $X \sim U(0, 1)$ and $Y \sim U(0, M)$.
2. Accept X if $Y \leq f(X) = 3X^2$; otherwise, reject and repeat.

This method generalizes to other distributions where direct sampling is not feasible.

Sampling from the Normal Distribution

The normal distribution, characterized by its bell-shaped probability density function, is one of the most important distributions in statistics due to its occurrence in natural phenomena and its central role in the Central Limit Theorem. Generating samples from a normal distribution is fundamental in simulations, statistical modeling, and various applications in engineering and the sciences. Several methods have been developed to achieve this, each leveraging different mathematical principles.

Properties of the Normal Distribution The normal distribution is defined by its mean μ and variance σ^2, with the probability density function (PDF) given by:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Key properties include symmetry around the mean, the empirical rule (68-95-99.7), and its role as the limiting distribution under the Central Limit Theorem.

Box-Muller Transform

The Box-Muller Transform is a widely used method for generating pairs of independent standard normal random variables from two independent uniform random variables. This method leverages polar coordinates and the properties of exponential and trigonometric functions.

Theorem 6 (Box-Muller Transform)**.** *Let U_1 and U_2 be independent uniform random variables on $(0,1]$. Then the transformations*

$$Z_1 = \sqrt{-2\ln U_1} \cdot \cos(2\pi U_2),$$

$$Z_2 = \sqrt{-2\ln U_1} \cdot \sin(2\pi U_2),$$

yield two independent standard normal random variables Z_1 and Z_2.

Derivation and Intuition The Box-Muller Transform utilizes the transformation from Cartesian to polar coordinates. By expressing the uniform variables in polar form and applying the properties of the exponential and trigonometric functions, the resulting variables adhere to the standard normal distribution.

Generating Normal Samples To generate a normal random variable with mean μ and standard deviation σ, the standard normal variables Z_1 and Z_2 obtained from the Box-Muller Transform can be scaled and shifted:

$$X = \mu + \sigma Z_1,$$

$$Y = \mu + \sigma Z_2.$$

This process ensures that X and Y follow the desired normal distribution.

Central Limit Theorem-Based Sampling

The Central Limit Theorem (CLT) states that the sum of a large number of independent and identically distributed random variables, each with finite mean and variance, will approximate a normal distribution regardless of the original distribution of the variables. This property can be exploited to generate normal random variables.

Methodology To generate a normal random variable using the CLT:

1. Generate n independent and identically distributed uniform random variables $U_1, U_2, \ldots, U_n$ on $(0,1)$.
2. Compute their sum:
$$S_n = \sum_{i=1}^{n} U_i.$$
3. Normalize the sum to have mean μ and variance σ^2:
$$Z = \frac{S_n - \frac{n}{2}}{\sqrt{\frac{n}{12}}}.$$

As n becomes large, Z approaches a standard normal distribution $N(0,1)$. For practical purposes, moderate values of n (e.g., $n = 12$) can provide reasonable approximations.

4.2 Reaction Systems

The system under study is characterized by a set of reactions occurring over time from 0 to T. These reactions involve various compartments or states in the system, each with its distinct dynamics.

Each reaction in the system is associated with a waiting time T_i, which follows an exponential distribution:

$$T_i \sim \text{Exp}(\lambda_i) \quad \text{for } i = 1, 2, \ldots$$

Here, λ_i is the rate parameter of the i-th reaction.

The count C indicates the number of occurrences of a particular event in a time interval Δt:

$$C = \# \text{ of occurrences in } \Delta t$$

For instance, the probability of no occurrences (i.e., $C = 0$) within Δt is:

$$P(C = 0) = 1 - (1 - e^{-\lambda \Delta t})$$

The probability of exactly one occurrence in Δt is:

$$P(C = 1) = e^{-\lambda \Delta t} \cdot \frac{(\lambda \Delta t)^1}{1!}$$

And for reaction i, the count C_i follows a Poisson distribution:

$$C_i \sim \text{Poisson}\,(\theta_i)$$

where θ_i is given by:

$$\theta_i = \lambda_i \prod_{j=1}^{p} \binom{Y(t)}{k_j} \Delta t$$

This equation models the frequency of each reaction, considering the current state $Y(t)$ of the system.

With the knowledge of C_i, we update the system states as follows:

$$\Delta Y(t) = Y(t + \Delta t) - Y(t) = V^T C$$

This expression calculates the change in state variables, $\Delta Y(t)$, based on the reactions that occurred.

Given observed data Y at specific time points, we aim to estimate the rate parameters λ that best describe the system dynamics. The dynamics can be represented as:

$$\Delta Y(t) = X(t)\lambda + \eta(t)$$

Here, $X(t)$ is the matrix capturing the influence of each reaction, and $\eta(t)$ represents the error term.

The residuals η are the difference between observed and predicted changes in state variables:

$$\eta = \Delta Y - X\lambda$$

The objective is to minimize the sum of squared residuals, $\Sigma\eta^2$, which is equivalent to minimizing $\eta^T\eta$. By differentiating with respect to λ and setting the derivative to zero, we find the optimal λ:

$$\frac{\partial}{\partial \lambda} \Sigma \eta^2 = 0$$

Solving this equation yields the estimated rate parameters $\hat{\lambda}$:

$$\hat{\lambda} = (X^T X)^{-1} X^T \Delta Y$$

This optimized $\hat{\lambda}$ provides the best-fit parameters for our model, describing the underlying dynamics of the system.

Stoichiometry in reaction systems examines the quantitative relationships between reactants and products in chemical reactions. Consider a system with r different reactants and p different products involved in a series of reactions.

Each reaction can be described by an equation. For the j-th reaction, the equation is:

$$\sum_{i=1}^{r} k_{ij} R_i \xrightarrow{\theta_j} \sum_{i=1}^{p} s_{ij} P_i. \tag{1}$$

Here, R_i represents the i-th reactant and P_i represents the i-th product. The coefficients k_{ij} and s_{ij} are the stoichiometric coefficients for reactants and products, respectively, in the j-th reaction.

Example (Basic Predator-Prey Model)**.** Consider a simplified ecological model with two types of organisms: predators (denoted as P) and prey (denoted as Q). The dynamics of this system are captured by the following reactions, each with its associated rate:

$$Q \xrightarrow{\beta_{\text{repro}}} 2Q \quad \text{(Prey reproduction at rate } \beta_{\text{repro}}\text{)}, \tag{2}$$

$$P + Q \xrightarrow{\beta_{\text{pred}}} 2P \quad \text{(Predation and predator reproduction at rate } \beta_{\text{pred}}\text{)}, \tag{3}$$

$$P \xrightarrow{\beta_{\text{death}}} \emptyset \quad \text{(Predator death at rate } \beta_{\text{death}}\text{)}. \tag{4}$$

We define the reactants and products for each reaction as follows:

- Reactants: $R = \{Q, P, P + Q\}$,
- Products: $P = \{2Q, 2P, \emptyset\}$.

When does something happen:

$$\begin{aligned} \Delta T_{P\to\emptyset} &\sim \text{Exp}(\beta_{\text{death}}), \\ \Delta T_{P+Q\to 2P} &\sim \text{Exp}(\beta_{\text{pred}}), \\ \Delta T_{Q\to 2Q} &\sim \text{Exp}(\beta_{\text{repro}}). \end{aligned}$$

About reaction in Δt time:

- $P \to \emptyset$: will have happened approx $2\beta_{\text{death}}\Delta t$ times,
- $P + Q \to 2P$: $6\beta_{\text{pred}}\Delta t$,
- $Q \to 2Q$: $3\beta_{\text{repro}}\Delta t$.

We expect to see

$$\begin{aligned} E[Y_P(\Delta t)] &= Y_P(0) + 6\beta_{\text{pred}}\Delta t - 2\beta_{\text{death}}\Delta t, \\ E[Y_Q(\Delta t)] &= Y_Q(0) + 3\beta_{\text{repro}}\Delta t - 6\beta_{\text{pred}}\Delta t. \end{aligned}$$

We actually saw at $\Delta t = 2$:

$$\begin{aligned} Y_P(2) &= 5, \\ Y_Q(2) &= 4. \end{aligned}$$

Now we want to know what we expect to happen at $2 + \Delta t$:

$$\begin{aligned} R_1 &: 4\beta_{\text{repro}}\Delta t \text{ times}, \\ R_2 &: 20\beta_{\text{pred}}\Delta t \text{ times}, \\ R_3 &: 5\beta_{\text{death}}\Delta t \text{ times}. \end{aligned}$$

So we expect to see

$$\begin{aligned} E[Y_P(2 + \Delta t)] &= 5 + 20\beta_{\text{pred}}\Delta t - 5\beta_{\text{death}}\Delta t, \\ E[Y_Q(2 + \Delta t)] &= 4 + 3\beta_{\text{repro}}\Delta t - 20\beta_{\text{pred}}\Delta t. \end{aligned}$$

We actually see

$$\begin{aligned} Y_P(4) &= 8, \\ Y_Q(4) &= 1. \end{aligned}$$

What are the most likely values for β_{repro}, β_{pred}, β_{death}?
We saw 3×2 values of the states:
First interval:

$$\begin{aligned} 5 &\approx 2 + 6\beta_{\text{pred}} \cdot 2 - 2\beta_{\text{death}} \cdot 2, \\ 4 &\approx 3 + 3\beta_{\text{repro}} \cdot 2 - 6\beta_{\text{pred}} \cdot 2. \end{aligned}$$

Second interval:

$$\begin{aligned} 8 &\approx 5 + 20\beta_{\text{pred}} \cdot 2 - 5\beta_{\text{repro}} \cdot 2, \\ 1 &\approx 4 + 5\beta_{\text{repro}} \cdot 2 - 20\beta_{\text{pred}} \cdot 2. \end{aligned}$$

This leads to the matrix equation:

$$\begin{bmatrix} 3 \\ 1 \\ 3 \\ -3 \end{bmatrix} = \begin{bmatrix} 0 & 12 & -4 \\ 6 & -12 & 0 \\ 40 & -10 & 0 \\ 10 & -40 & 0 \end{bmatrix} \begin{bmatrix} \beta_{\text{repro}} \\ \beta_{\text{pred}} \\ \beta_{\text{death}} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{bmatrix}$$

To estimate the model parameters (β_{repro}, β_{pred}, β_{death}), we use the observed values from two intervals, forming the following matrix equation:

$$\begin{bmatrix} 3 \\ 1 \\ 3 \\ -3 \end{bmatrix} = \begin{bmatrix} 0 & 12 & -4 \\ 6 & -12 & 0 \\ 0 & 40 & -10 \\ 10 & -40 & 0 \end{bmatrix} \begin{bmatrix} \beta_{\text{repro}} \\ \beta_{\text{pred}} \\ \beta_{\text{death}} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{bmatrix}$$

The least squares estimate $\hat{\beta}$ is calculated as $\hat{\beta} = (X^T X)^{-1} X^T Y$. Using this formula, the estimated values are:

$$\hat{\beta}_{\text{repro}} \approx 0.36,$$
$$\hat{\beta}_{\text{pred}} \approx 0.155,$$
$$\hat{\beta}_{\text{death}} \approx 0.235.$$

To further assess the reliability of these estimates, a bootstrap analysis was performed. The table below provides the descriptive statistics for each bootstrapped beta coefficient:

Statistic	β_{repro}	β_{pred}	β_{death}
Mean	-0.344	-0.095	-0.809
Std	1.083	0.332	1.221
Min	-2.100	-0.450	-2.100
25%	-0.733	-0.450	-2.100
50%	0.360	0.155	-0.050
75%	0.633	0.233	0.235
Max	0.633	0.233	0.633

Table 3: Descriptive statistics for each bootstrapped beta coefficient.

□