Neural Network

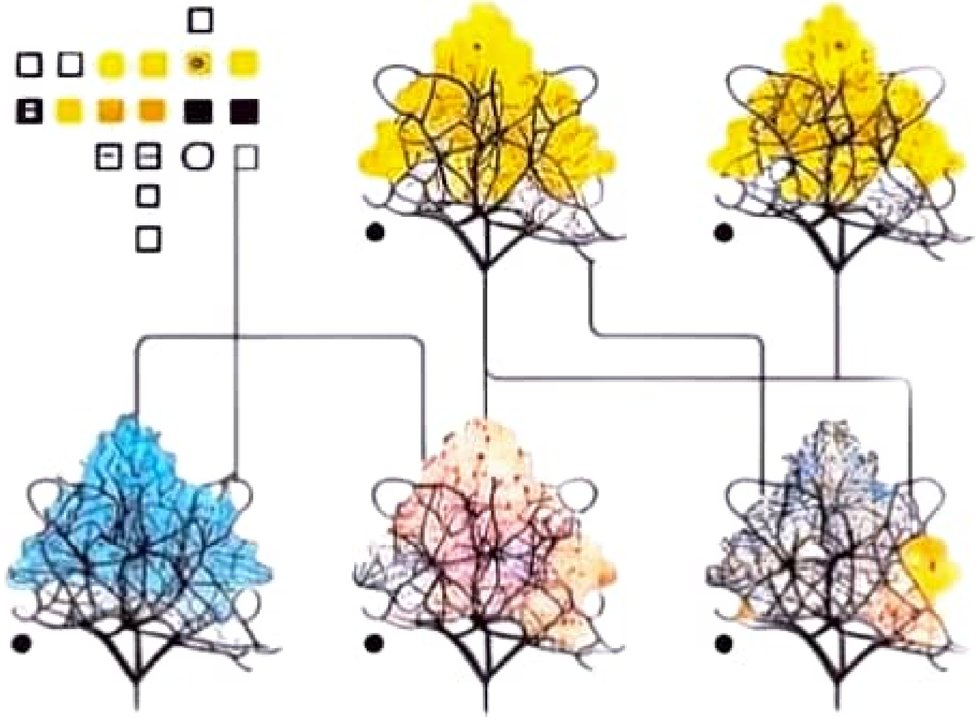
A neural network is a type of machine learning algorithm that is inspired by the structure and function of the human brain, and consists of interconnected processing nodes that are organized into layers.

Support Vector Machines

Support vector machines (SVMs) are a type of machine learning algorithm that is used for classification and regression tasks, and finds the hyperplane that maximally separates the classes in the data.

Nearest Neighbour

Nearest neighbor is a type of machine learning algorithm that makes predictions for a sample by finding the most similar samples in the training data and using their labels to make a prediction.



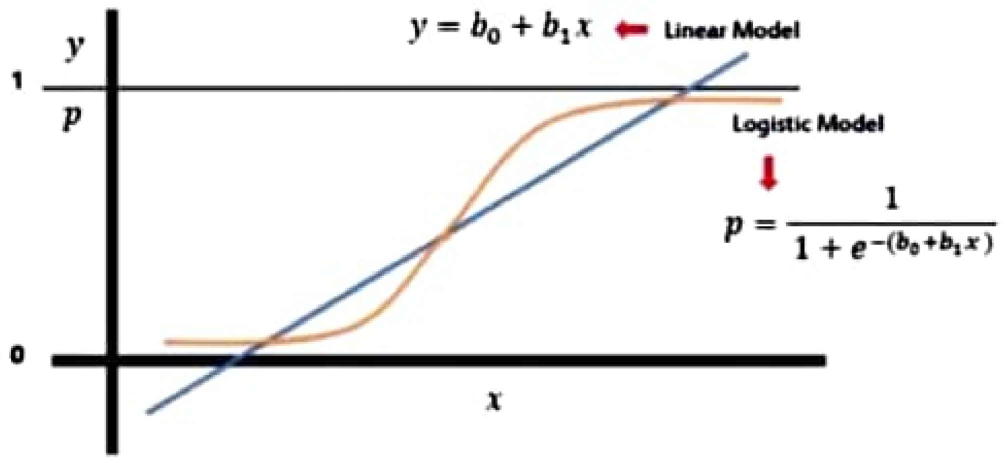
Random Forest

A random forest is a type of ensemble learning algorithm that trains multiple decision trees on random subsets of the data and then combines their predictions to make a final prediction. This can improve the performance of the model compared to using a single decision tree.

logistic regression

Logistic Regression is a Supervised statistical technique to find the probability of dependent variable.

The Logistic Regression instead for fitting the best fit line, condenses the output of the linear function between 0 and 1.



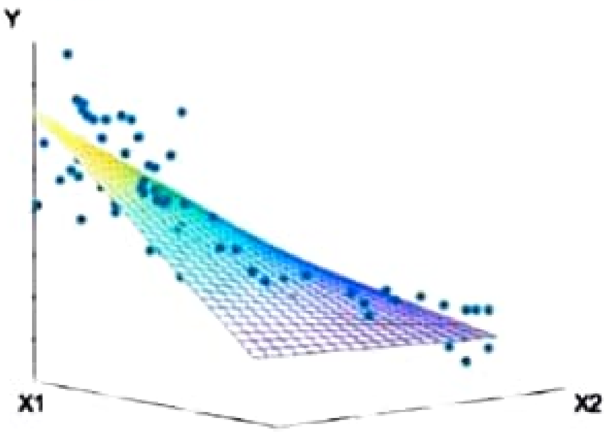
In the formula of the logistic model, when b0 + b1x = 0, then the p will be 0.5, similarly, b0 + b1x > 0, then the p will be going towards 1 and b0 + b1x < 0, then the p will be going towards 0.

linear regression

A Linear Regression model representation is a linear equation :

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_i x_i$$

β_0 is usually called intercept or bias coefficient. The dimension of the hyperplane of the regression is its complexity.



Learning a LR means estimating the coefficients from the training data. Common methods include Gradient Descent or Ordinary Least Squares.

Variations

Lasso Regression : where OLS is modified to minimize the sum of the coefficients (L1 regularization)

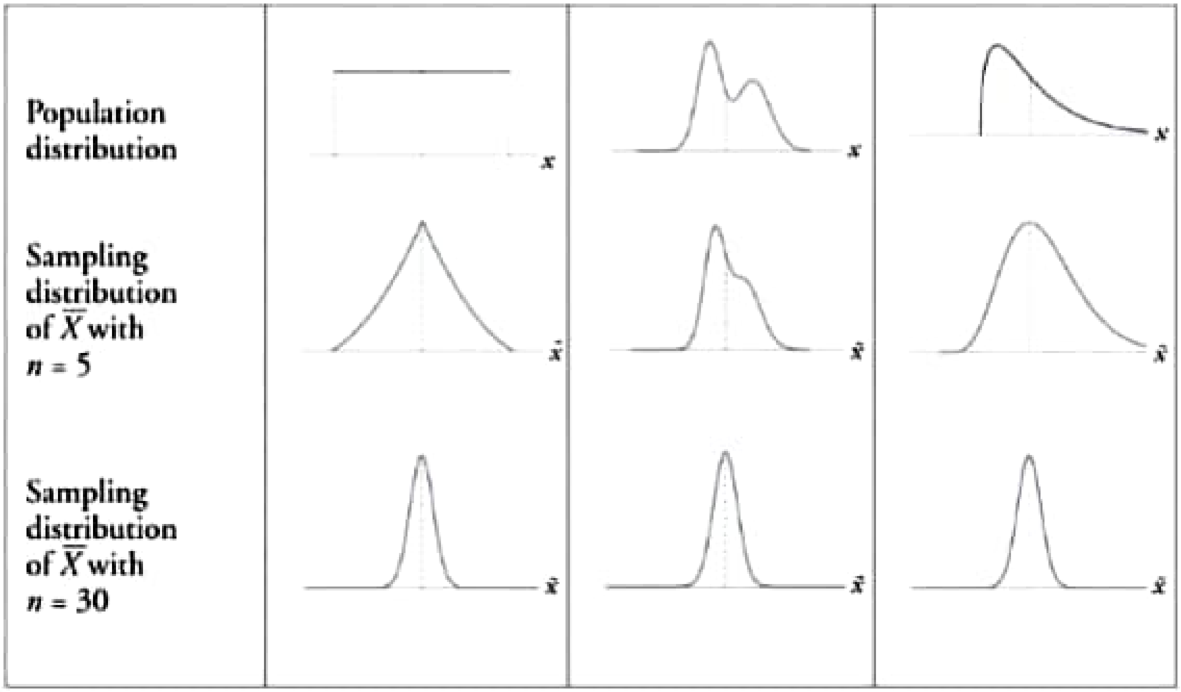
Ridge Regression : where OLS is modified to minimize the squared sum of the coefficients (L2 regularization)

central limit theorem

For large sample sizes, the sampling distribution of means will approximate to normal distribution even if the population distribution is not normal.

Mean of the sample means is computed as $\mu_{\bar{X}} = \mu$

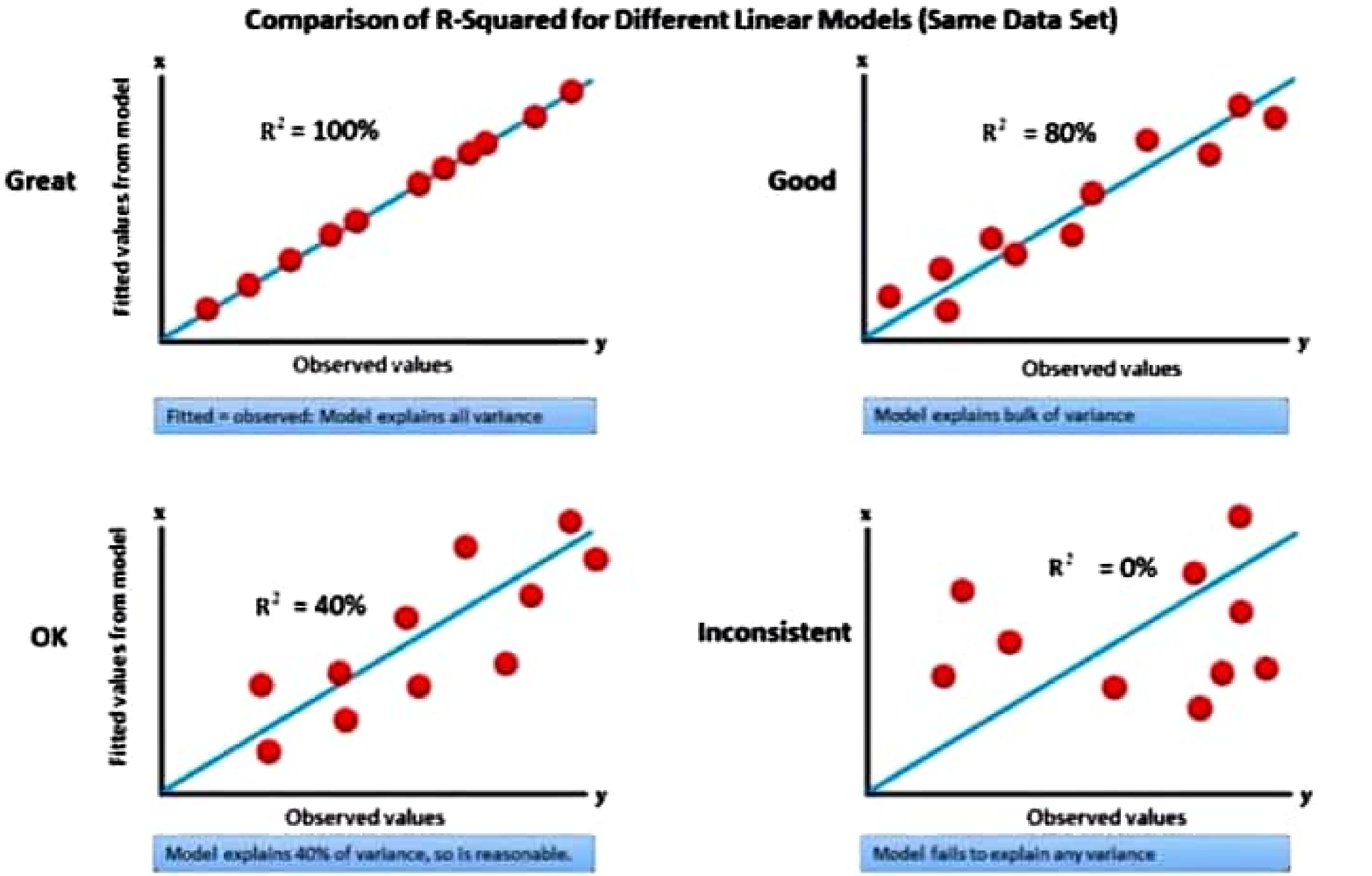
And the standard deviation of sample means $\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$



Above picture, shows 3 different population distributions which are not normal. Sampling distribution of means gets a little closer to normal distribution when we take n =5 and almost normal distribution when n=30

R-Square

R-square(R^2) is also known as the **coefficient of determination**, It is the **proportion** of variation in **Y explained by** the independent variables **X**. It is the measure of goodness of fit of the model.



If R^2 is 0.8 it means **80% of the variation** in the output can be explained by the **input variable**.

bernoulli trial

In the theory of probability and statistics, a **Bernoulli trial** (or binomial trial) is a random experiment with exactly **two possible outcomes**, "success" and "failure", in which the probability of success is the **same** every time the experiment is conducted.

A random variable corresponding to a binomial experiment is denoted by $B(n,p)$, and is said to have a binomial distribution. The probability of exactly k successes in the experiment $B(n,p)$ is given by:

$$P(k) = \binom{n}{k} p^k q^{n-k}$$

where,

n : number of experiments
k : probability of success in k times
p : probability of single success
q : probability of single failure (1=1−p)

poisson distribution

The Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant mean rate and independently of the time since the last event.
A discrete random variable X is said to have a Poisson distribution, with parameter $\lambda > 0$ if it has a probability mass function given by:

$$f(k;\lambda) = \Pr(X{=}k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

where,

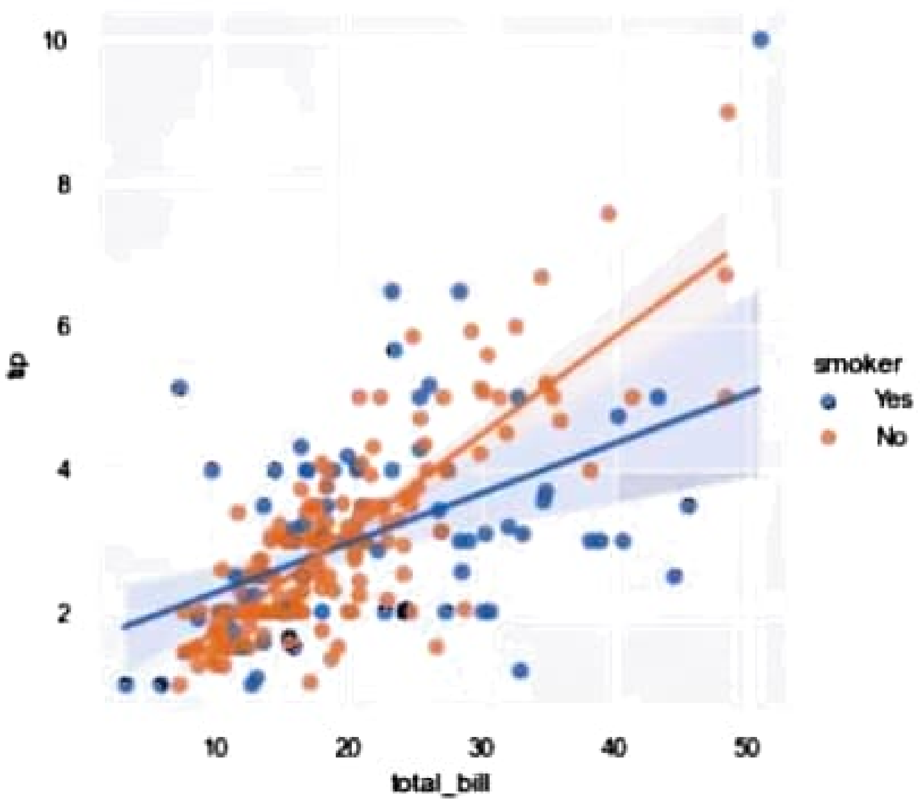
k is the number of occurrences ($k = 0, 1, 2, \ldots$)
e is Euler's number ($e = 2.71828\ldots$)
! is the factorial function.
The positive real number λ is equal to the expected value of X and also to its variance.

seaborn
scatter with regression

```
import seaborn as sns
# Load Your Dataframe
tips = sns.load_dataset("tips")
# Visualize
sns.set_theme()
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips)
```

executed in 479ms, finished 17:15:38 2022-04-24

```
<seaborn.axisgrid.FacetGrid at 0x1f389ef2b80>
```

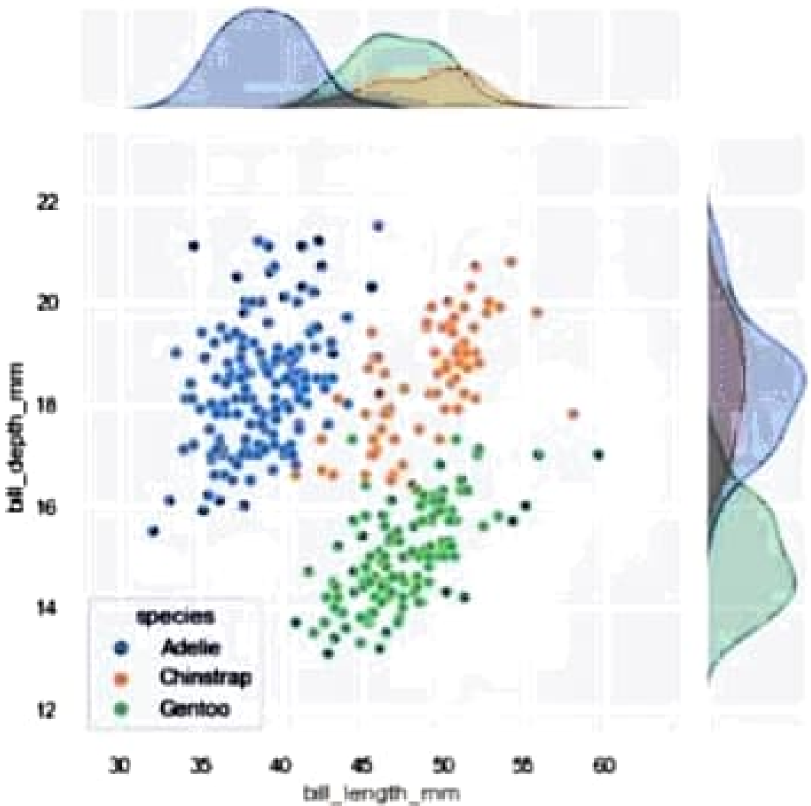


seaborn joint plot

```
import seaborn as sns
# Load Your Dataframe
penguins = sns.load_dataset("penguins")
# Visualize
sns.set_theme()
sns.jointplot(data=penguins, x="bill_length_mm", y="bill_depth_mm", hue="species")
```

executed in 381ms, finished 20 02 11 2022-05-10

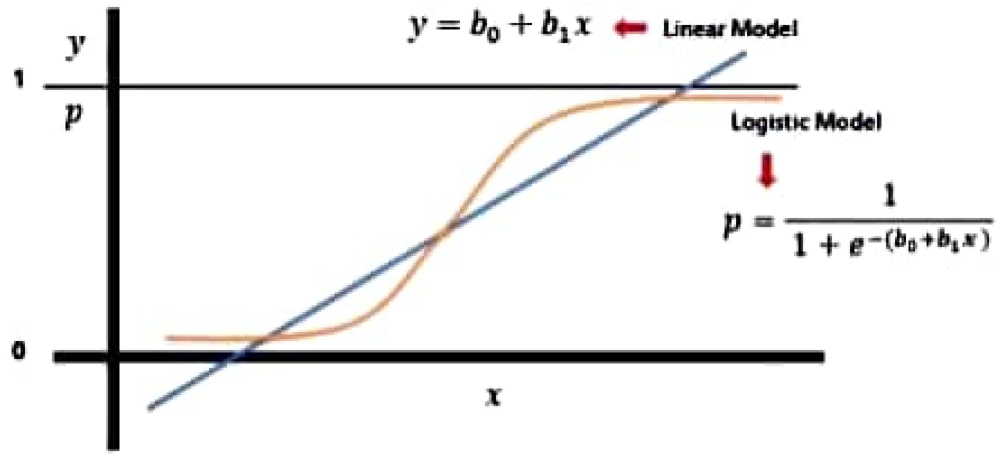
```
<seaborn.axisgrid.JointGrid at 0x18b496fb130>
```



logistic regression

Logistic Regression is a Supervised statistical technique to find the probability of dependent variable.

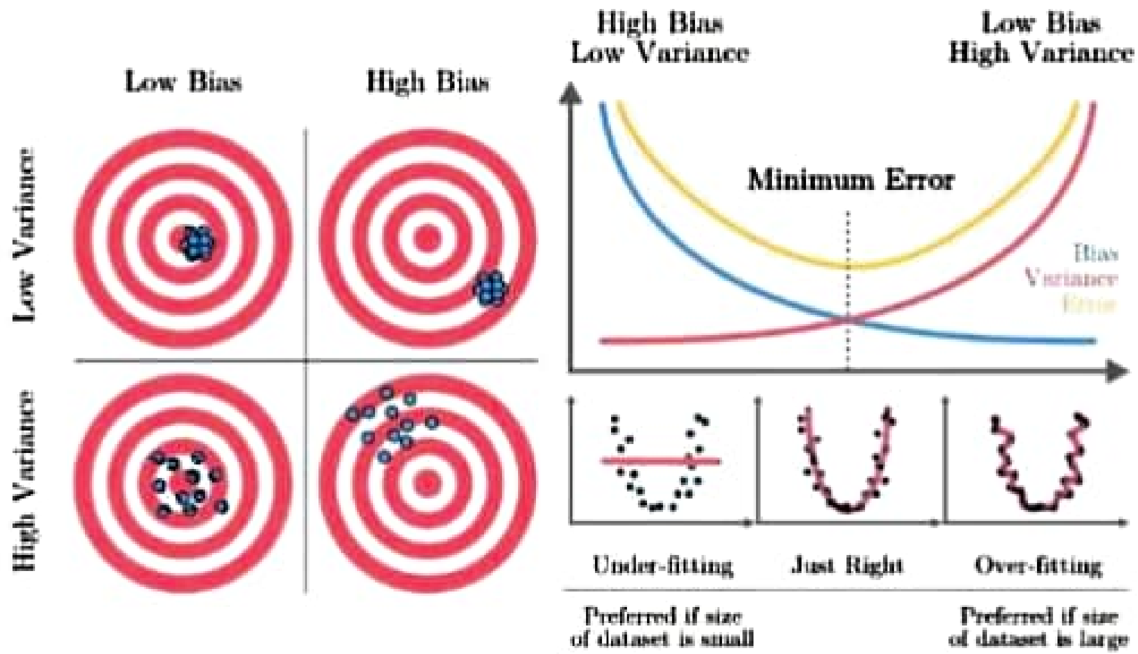
The Logistic Regression instead for fitting the best fit line, condenses the output of the linear function between 0 and 1.



In the formula of the logistic model, when b0 + b1x = 0, then the p will be 0.5, similarly, b0 + b1x > 0, then the p will be going towards 1 and b0 + b1x < 0, then the p will be going towards 0.

bias-variance tradeoff

Bias–variance tradeoff is the property of a model that the variance of the parameter estimated across samples can be reduced by increasing the bias in the estimated parameters.



Bias–variance Tradeoff

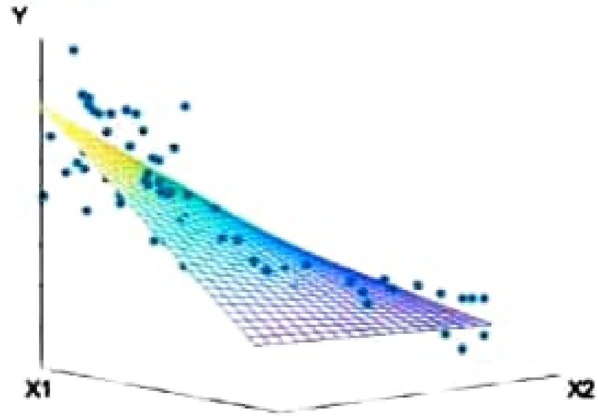
- Increasing bias (not always) reduces variance and vice-versa
- Error = bias2 + variance +irreducible error
- The best model is where the error is reduced
- Compromise between bias and variance

linear regression

A Linear Regression model representation is a linear equation :

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_i x_i$$

β_0 is usually called intercept or bias coefficient. The dimension of the hyperplane of the regression is its complexity.



Learning a LR means estimating the coefficients from the training data. Common methods include Gradient Descent or Ordinary Least Squares.

Variations

Lasso Regression : where OLS is modified to minimize the sum of the coefficients (L1 regularization)

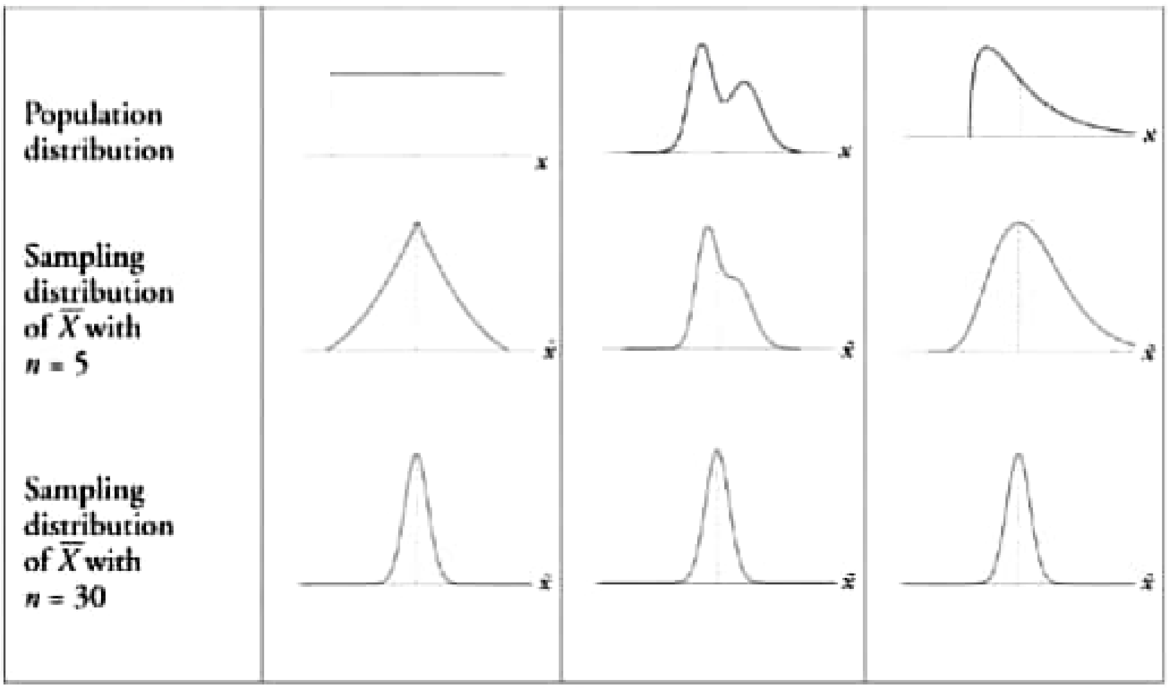
Ridge Regression : where OLS is modified to minimize the squared sum of the coefficients (L2 regularization)

central limit theorem

For large sample sizes, the sampling distribution of means will approximate to normal distribution even if the population distribution is not normal.

Mean of the sample means is computed as $\mu_{\bar{X}} = \mu$

And the standard deviation of sample means $\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}}$



Above picture, shows 3 different population distributions which are not normal. Sampling distribution of means gets a little closer to normal distribution when we take n =5 and almost normal distribution when n=30

the law of large number

The law of large numbers is a theorem that describes the result of performing the same experiment a large number of times. According to the law, the **average** of the results obtained from a large number of trials should be **close** to the **expected value.**

as $n \to \infty$, the sample mean $\langle x \rangle$ equals the population mean μ of each variable.

$$\begin{aligned} \langle X \rangle &= \left\langle \frac{X_1 + \ldots + X_n}{n} \right\rangle \\ &= \frac{1}{n} \left(\langle X_1 \rangle + \ldots + \langle X_n \rangle \right) \\ &= \frac{n\,\mu}{n} \\ &= \mu. \end{aligned}$$

Example : Law of large numbers in **Insurance**

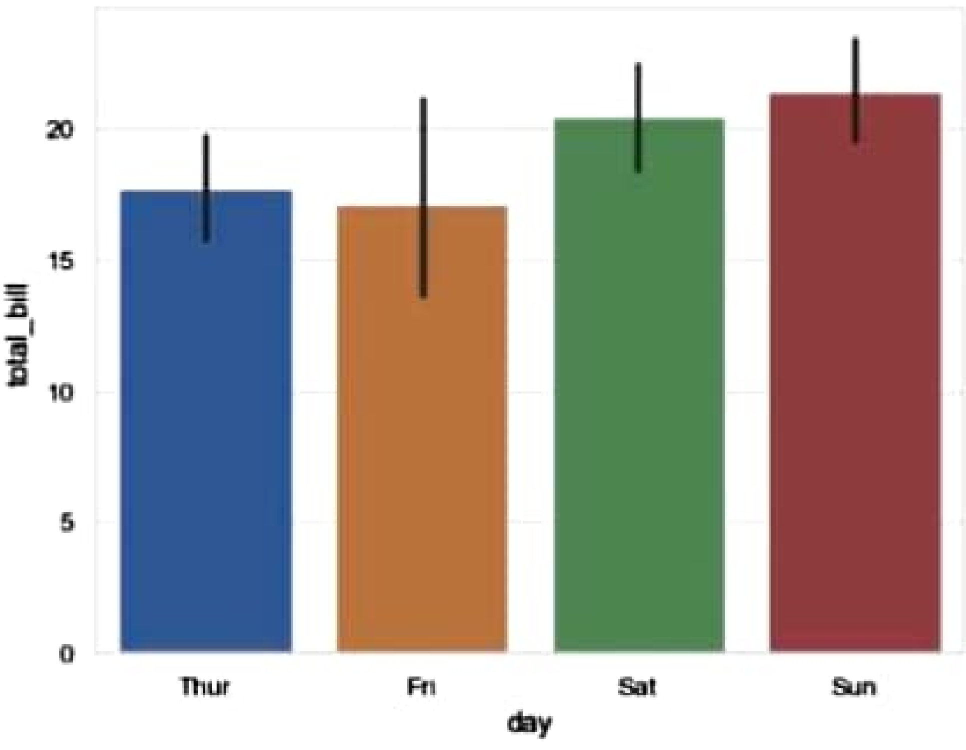
Imagine **500** people **paying** a **premium** for his property to the Insurance. Even though they pay you for the property damage that was caused by fire to **10%** of the subscriber i.e. **50** people they have the premium of **450 people still**. And that's how they calculate their risk management and the Law of Large Numbers comes in the role.

seaborn
simple barchart

Draw a set of vertical bar plots grouped by a categorical variable:

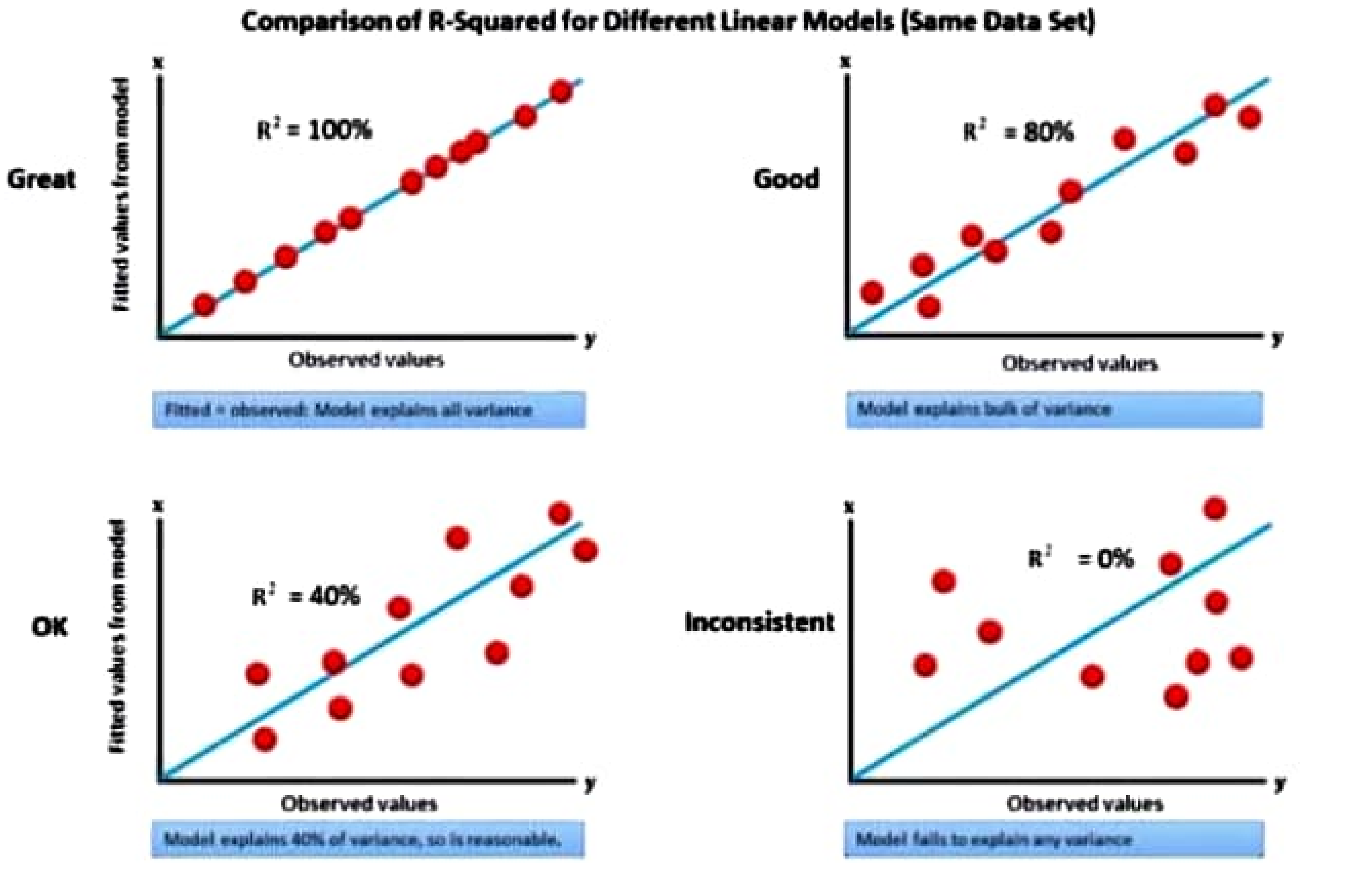
```
>>> import seaborn as sns
>>> sns.set_theme(style="whitegrid")
>>> tips = sns.load_dataset("tips")
>>> ax = sns.barplot(x="day", y="total_bill", data=tips)
```

your dataframe



R-Square

R-square(R^2) is also known as the **coefficient of determination**, It is the **proportion** of variation in **Y explained by** the independent variables **X**. It is the measure of goodness of fit of the model.



If R^2 is 0.8 it means **80% of the variation** in the output can be explained by the **input variable**.

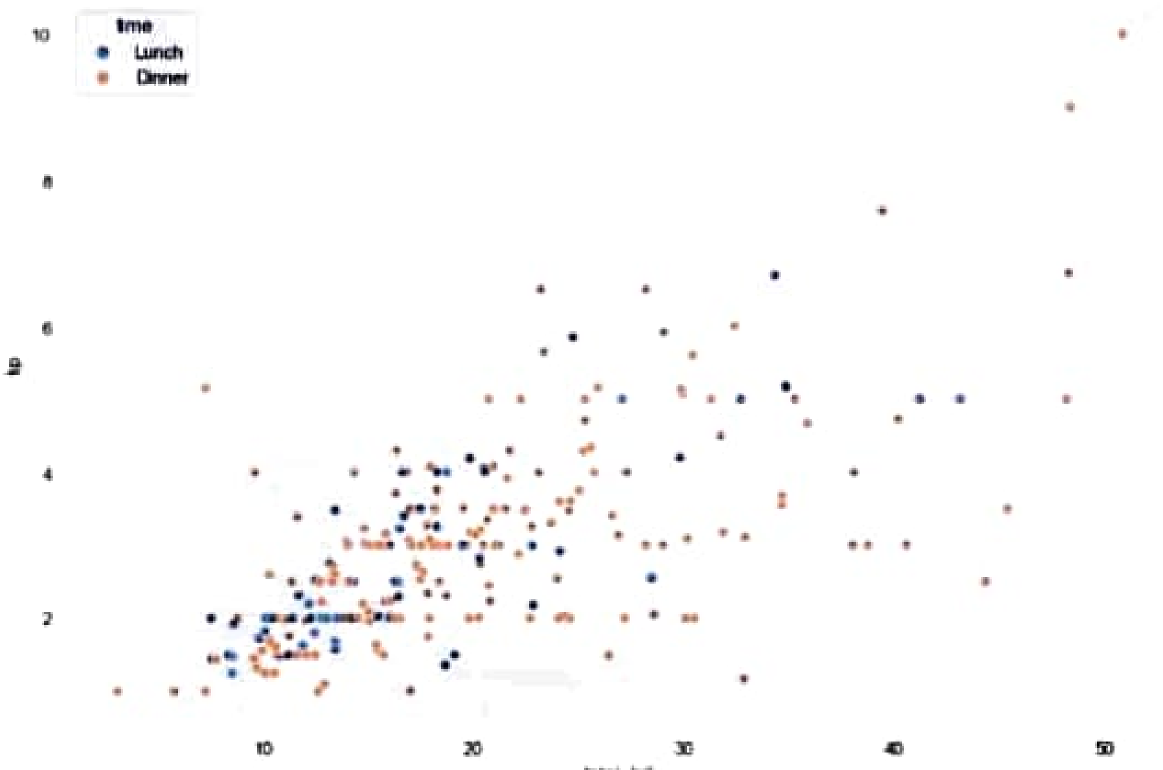
seaborn
basic scatterplot

```
import seaborn as sns

# Load Your Dataframe
tips = sns.load_dataset("tips")

# Plot Scatter with Hue

# Set Theme
sns.set_theme()
# Set figure size
sns.set(rc={'figure.figsize':(12,8)})
# Visualize
sns.scatterplot(data=tips, x="total_bill", y="tip", hue="time")
```

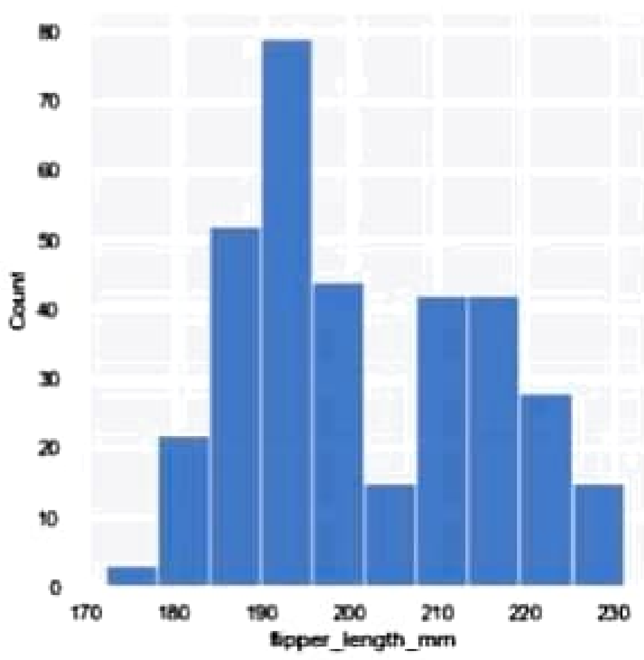


seaborn
basic histogram

```
import seaborn as sns
# Load Your Dataframe
penguins = sns.load_dataset("penguins")
# Visualize
sns.set_theme()
sns.displot(penguins, x="flipper_length_mm")
```

executed in 200ms, finished 17:41:58 2022-04-24

```
<seaborn.axisgrid.FacetGrid at 0x1bd3632d250>
```



bayes theorem

In probability theory and statistics, Bayes' theorem named after Thomas Bayes, describes the probability of an event, based on prior knowledge of conditions that might be related to the event. (Wikipedia)

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

where

A, B	= events
$P(A\|B)$	= probability of A given B is true
$P(B\|A)$	= probability of B given A is true
$P(A), P(B)$	= the independent probabilities of A and B

mean squared error

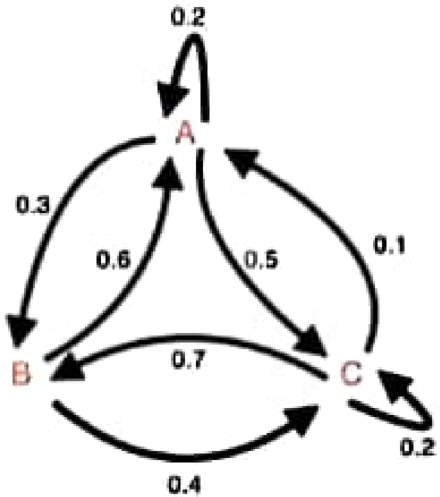
Mean Squared Error / MSE is one of the most common regression loss functions. Mean Squared Error also known as L2 loss, we calculate the error by squaring the difference between the predicted value and actual value and averaging it across the dataset.

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

MSE is also known as Quadratic loss as the penalty is not proportional to the error but to the square of the error. Squaring the error gives higher weight to the outliers, which results in a smooth gradient for small errors.
A good model will have MSE value closer to zero.

markov chain model

A **Markov chain** is a stochastic model created by Andrey Markov, which outlines the **probability** associated with a **sequence** of **events occurring** based on the state in the **previous event.**



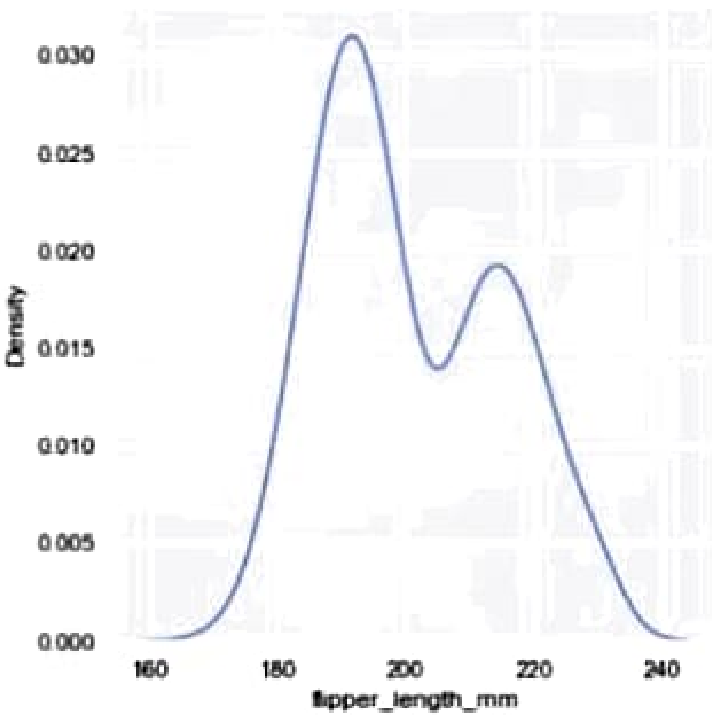
A very common **application** of Markov chains in data science is **text prediction**. It's an **area of NLP** which is commonly used in the tech industry by companies like Google, Linkedin and Instagram. When you're writing emails, google predicts and **suggests you words/phrases** to autocomplete your email, when you receive messages on Instagram or Linkedin, the app suggests some potential replies.

seaborn
density plot (KDE)

```
import seaborn as sns
# Load Your Dataframe
penguins = sns.load_dataset("penguins")
# Visualize
sns.set_theme()
sns.displot(penguins, x="flipper_length_mm", kind="kde")
```

executed in 174ms, finished 17:11:20 2022-04-24

```
<seaborn.axisgrid.FacetGrid at 0x20b1ff64490>
```



seaborn
KDE plot with hue

```
import seaborn as sns
# Load Your Dataframe
penguins = sns.load_dataset("penguins")
# Visualize
sns.set_theme()
sns.displot(penguins, x="flipper_length_mm", hue="species", kind="kde", multiple="stack")
```

executed in 289ms, finished 17:49:08 2022-04-24

```
<seaborn.axisgrid.FacetGrid at 0x1bd3630eee0>
```