Normal Distribution

In statistics, a normal distribution or Gaussian distribution is a type of continuous probability distribution for a real-valued random variable. The general form of its probability density function is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

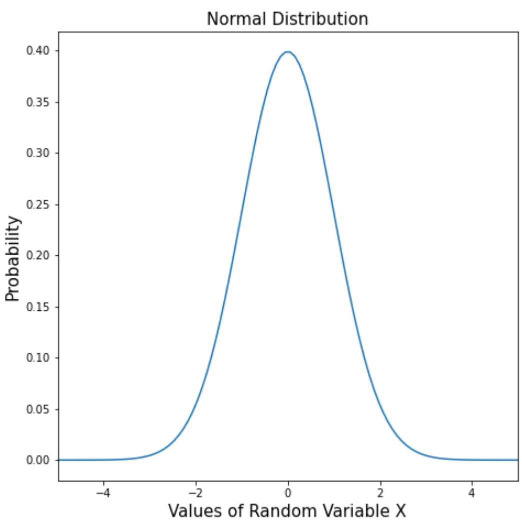
where:

x = value of the variable or data being examined and f(x) the probability function
μ = the mean
σ = the standard deviation \

As with any probability distribution, the normal distribution describes how the values of a variable are distributed. It is the most important probability distribution in statistics because it accurately describes the distribution of values for many natural phenomena. Characteristics that are the sum of many independent processes frequently follow normal distributions. For example, heights, blood pressure, measurement error, and IQ scores follow the normal distribution.

Visualization of Normal Distribution

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
#
# Create a standard normal distribution with mean as 0 and standard deviation as 1
#
mu = 0
std = 1
snd = stats.norm(mu, std)
#
# Generate 100 random values between -5, 5
#
x = np.linspace(-5, 5, 100)
#
# Plot the standard normal  distribution for different values of random variable
# falling in the range -5, 5
#
plt.figure(figsize=(7.5,7.5))
plt.plot(x, snd.pdf(x))
plt.xlim(-5, 5)
plt.title('Normal Distribution', fontsize='15')
plt.xlabel('Values of Random Variable X', fontsize='15')
plt.ylabel('Probability', fontsize='15')
plt.show()
```



Generate a random normal distribution of size 2x3 with mean at 1 and standard deviation of 2:

In [2]:
```
from numpy import random

x = random.normal(loc=1, scale=2, size=(2, 3))

print(x)
```

```
[[-1.81678029  0.83313863  2.56082605]
 [-0.92080741  0.41989958 -1.07173519]]
```

Probability Density Function (PDF)

The probability density function (PDF) is a statistical expression that defines a probability distribution (the likelihood of an outcome) for a discrete random variable as opposed to a continuous random variable. When the PDF is graphically portrayed, the area under the curve will indicate the interval in which the variable will fall.

A continuous random variable X is said to follow the normal distribution if it's probability density function (PDF) is given by:

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\cdot\sigma^2}}\cdot e^{-\frac{1}{2}\cdot\left(\frac{x-\mu}{\sigma}\right)^2} \qquad \text{Equation 3.1}$$

Normal Distribution with Real World Distribution

Example1 :Let's understand the use case of the PDF with an example. Let's assume that we are working with the heights of kids in the 1st grade. We know from experience that such heights, when sampled in significant quantities, are normally distributed. However, we are in learning mode. Let's not go out and actually measure the heights of 1st graders. Let's make some sample data that is normally distributed. How can we do that easily?

First, we need some reasonable numbers for μ and σ. Has someone already done data sampling work on the heights of 1st graders? Yes! Check out THIS STUDY. The researchers of that study found μ = 37 inches and σ = 2 inches. Let's use these parameters and some python code to create some sample data – a valuable skill to have when learning data science. We can create the PDF of a normal distribution using basic functions in Python.

In [13]:
```
import math

### Probability Density Function (PDF) from Scratch
def PDF(x, mean, std_dev):
    probability = 1.0 / math.sqrt(2 * 3.141592*(std_dev)**2)  # first part of equation
    probability *= math.exp(-0.5 * ((x - mean)/std_dev)**2)  # multiply first part to second part


    return probability

X = []

### Create Sample Normally Distributed Data with Mean of 37, and Std_Dev of 2 ###
for x in range(29, 46):
    y = PDF(x, 37, 2)

    ### We want to create fake data by replicating x according to it's probability ###
    N_vals_at_y = int(round(y * 1001, 0))
    for i in range(N_vals_at_y):
        X.append(x)


# Finding mean
mean = round(sum(X)/len(X), 4)

std_dev = 0.0
N = len(X)
for x in X:
    std_dev += (x - mean)**2

std_dev /= N
std_dev = math.sqrt(std_dev)
std_dev = round(std_dev, 2)
```

In [32]:
```
print("We sample measured the heights of "+ str(N) +" 1st graders.")
print("The mean is" ,mean)
print("The standard deviation is", std_dev)
```

```
We sample measured the heights of 1000 1st graders.
The mean is 37.0
The standard deviation is 1.99
```

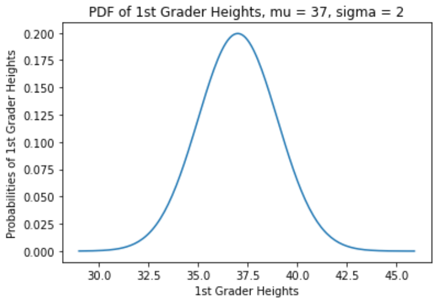
In [39]:

```
import scipy
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

figure,ax = plt.subplots()

X = []
Y = []

for x in np.arange(29, 46, 0.1):
    X.append(x)
    y = norm.pdf(x=x, loc=37.0, scale=2)
    Y.append(y)
ax.set_xlabel('1st Grader Heights')
ax.set_ylabel('Probabilities of 1st Grader Heights')
ax.set_title('PDF of 1st Grader Heights, mu = 37, sigma = 2')
plt.plot(X, Y)
plt.show()
```



Question: What is the probability of a 1st grader's height being 39 inches?

In [45]:

```
x = 39
p = norm.pdf(x=x, loc=37.0, scale=2)
print("The probability of a 1st grader's height being 39 inches is",p)
```

```
The probability of a 1st grader's height being 39 inches is 0.12098536225957168
```

Cumulative Distribution Function (CDF)

The cumulative distribution function, CDF, or cumulant is a function derived from the probability density function for a continuous random variable. It gives the probability of finding the random variable at a value less than or equal to a given cutoff, ie, P(X ≤ x).

Example 2: Let's use an example to help us understand the concepts of the cumulative distribution function (CDF). IQ scores are known to be normally distributed with a mean of 100 and a standard deviation of 15.

Question：what's the probability that someone's IQ score lower than 95 ?

In [41]:

```
import scipy.stats
mean = 100
sd = 15
dis = scipy.stats.norm(mean, sd)
```

In [44]:

```
## Probablity that someone's IQ lower than 95
p1 = dis.cdf(95)
print("The probability of a someone's IQ score lower than 95 is ",p1)
```

```
The probability of a someone's IQ score lower than 95 is  0.36944134018176367
```

In [ ]:

```

```