**Mathematics and Computer Science**

A good understanding of math is essential for every computer scientist, and the math requirement is starting to become more diverse.

* [Discrete Math](http://en.wikipedia.org/wiki/Discrete_mathematics) is the most important and basic class for computer science, and for this reason it is usually offered in CS departments instead of math departments. This class will underpin your intro to algorithms to class and teach you how to prove things mathematically and give you the fundamentals for analyzing data structures and algorithms.
* [Calculus](http://en.wikipedia.org/wiki/Calculus), while not directly used in intro-level computer science classes, is generally a sequence of courses offered by your university to buff up your math skills. As you start getting into things like numerical programming and machine learning, though, it will prove immensely useful. It's also a requirement for advanced probability/statistics courses.
* [Probability](http://en.wikipedia.org/wiki/Probability) is usually covered in some extent in your discrete math class, but you'll want to take a class on continuous probability distributions and [statistical inference](http://en.wikipedia.org/wiki/Statistical_inference), probably in the math and statistics department. This will give you a better understanding of how to do numerical computation and simulation, and is fundamentally necessary for [machine learning](http://en.wikipedia.org/wiki/Machine_learning), one of the most important applications of computer science.
* [Linear Algebra](http://en.wikipedia.org/wiki/Linear_algebra) is a class that you will find primarily useful for machine learning and (advanced) algorithms classes, but its importance in computer vision, computer graphics, machine learning, and other quantitative sub-disciplines is paramount.

That said, if an intro to machine learning class is available, they will probably cover enough linear algebra and other stuff that you can get by with a basic probability class. However, for graduate study in computer science, a good understanding of all areas of math above is essential.

Beyond undergraduate math, higher-level math courses are useful for certain theoretical areas of computer science (e.g. algorithmic game theory, which intersects with economics) and especially in going beyond being a machine learning practitioner to developing new algorithms. These courses include:

* [Real analysis](https://en.wikipedia.org/wiki/Real_analysis), including [measure theory](https://en.wikipedia.org/wiki/Measure_%28mathematics%29) where you'll find that if you study probability and calculus for long enough, they converge again. Analysis is generally a useful thing to know when you start working with algorithms involving numbers.
* [Optimization](https://en.wikipedia.org/wiki/Mathematical_optimization), including [linear optimization](https://en.wikipedia.org/wiki/Linear_programming), [convex optimization](https://en.wikipedia.org/wiki/Convex_optimization), [gradient descent](https://en.wikipedia.org/wiki/Gradient_descent), and so on. In many cases, "learning" a machine learning model basically boils down to optimizing an objective function, and properties of this function such as whether it is convex have a big impact on how easy it is to optimize.
* [Numerical methods](https://en.wikipedia.org/wiki/Numerical_analysis): some wouldn't consider this a math class *per se*, but in translating algorithms and theory into the imperfect representation of floating-point math, there are many practical problems to be solved. For example, the [log-sum-exp trick](https://stackoverflow.com/a/15278480/586086).
* For those who will be in "data science" and related fields, advanced statistics and especially [causal inference](https://en.wikipedia.org/wiki/Causal_inference) are very important. There are a lot of things to know, mostly because having access to a lot of data tempts [this problem](https://xkcd.com/552/) for the uninitiated.