A Continuation of ggplot2: Mapping Probabilities

A Continuation of ggplot2: Mapping Probabilities

In this post, we will continue to explore the use of ggplot2 in the R universe, and look specifically at ways we can represent probabilities using a combination of ggplot2 and R's base functions. We will mainly use ggplot2's function `qplot`, which allows for quick creation of standard graphs, and we will focus on exploring the way we can create representations of various common probability distributions through these graphs. For this we will require the ggplot2 package.

```
library('ggplot2')
```

A quick look at `qplot`

`qplot` can be used to quickly make a graph. It has this format:

```
qplot(x=, y=, data=,color=, shape=, size=, alpha=, geom=
        , method=, formula=, facets=, xlim=, ylim= xlab=, ylab=, main=, sub=)
```

Some Statistical Distributions

Here are the distributions we will be representing using ggplot2:

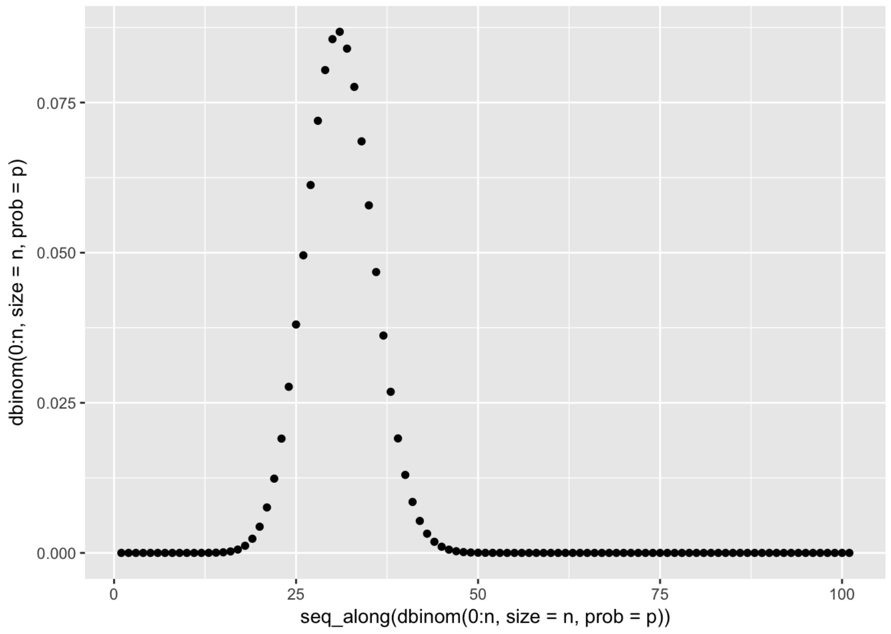
Distribution	**R name**
Binomial	binom
Beta	beta
Exponential	exp
Normal	norm

For a full list of distributions possible with R, head here: https://en.wikibooks.org/wiki/R_Programming/Probability_Distributions

Binomial Distribution

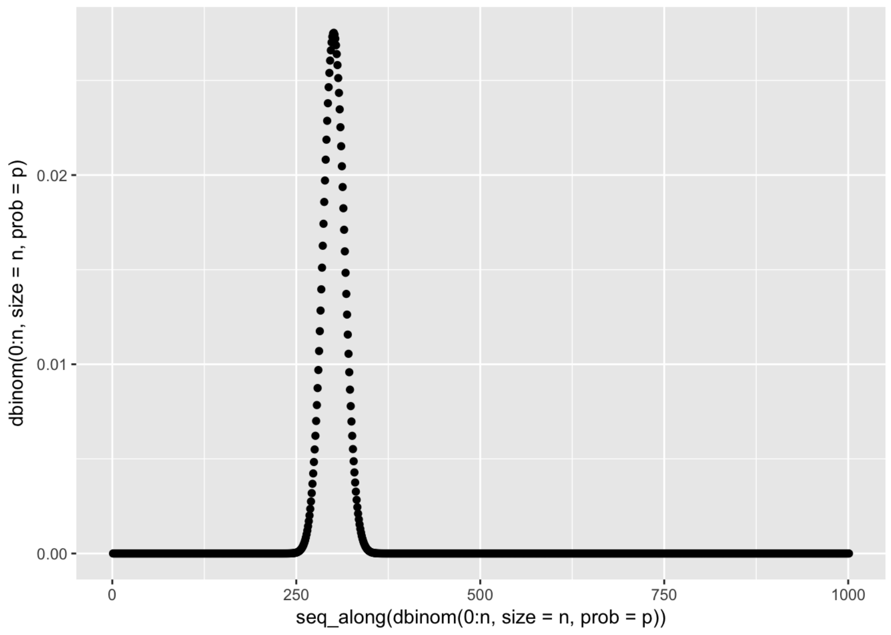
The binomial (n,p) distribution is used to model the number of occurences of an event with probability p occurring in n independent trials. The probability of k successes in n trials is $\binom{n}{k}p^k(1-p)^{n-k}$, where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

```
# representing the binomial distribution using qplot
n <- 100
p <- 0.3
qplot(y = dbinom(0:n,size = n, prob = p))
```



Looking at this graph, we can see the binomial distribution's shape, however, it does not look very nice at all! We will see as n becomes larger and larger, this density graph will become closer and closer to a smooth curve.

```
n <- 1000
qplot(y = dbinom(0:n,size = n, prob = p))
```

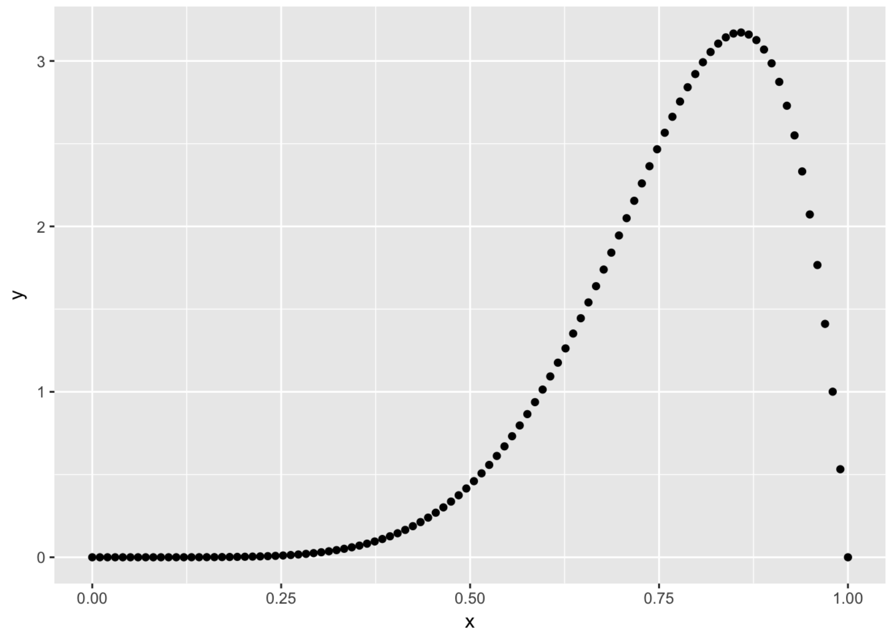


Here we see a much smoother looking graph. We will explore better ways to make the graphs look good in the upcoming distributions

The Beta Distribution

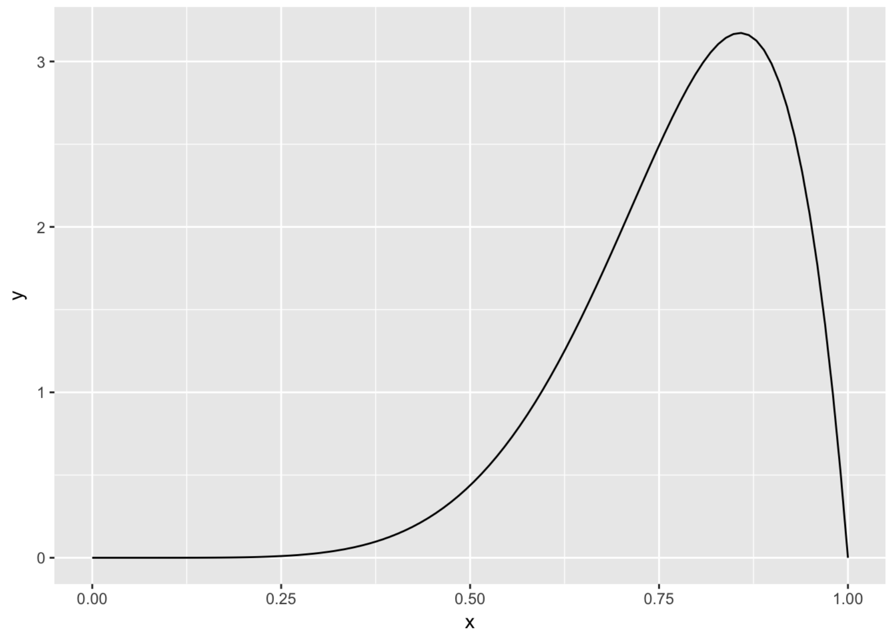
The beta distribution is a continuous distribution with two parameters. It is most commonly used to model uncertainty about the success of an experiment. We will make a preliminary graph below:

```
x <- seq(0,1,length=100)
y <- dbeta(x,7,2)
qplot(x, y)
```



Looking at this, we see that this is noncontinuous, as we modelled the distribution using a discrete sequence. In order to make this graph show a continuous line, we can use the argument `geom = 'line'`

```
qplot(x, y, geom = 'line')
```

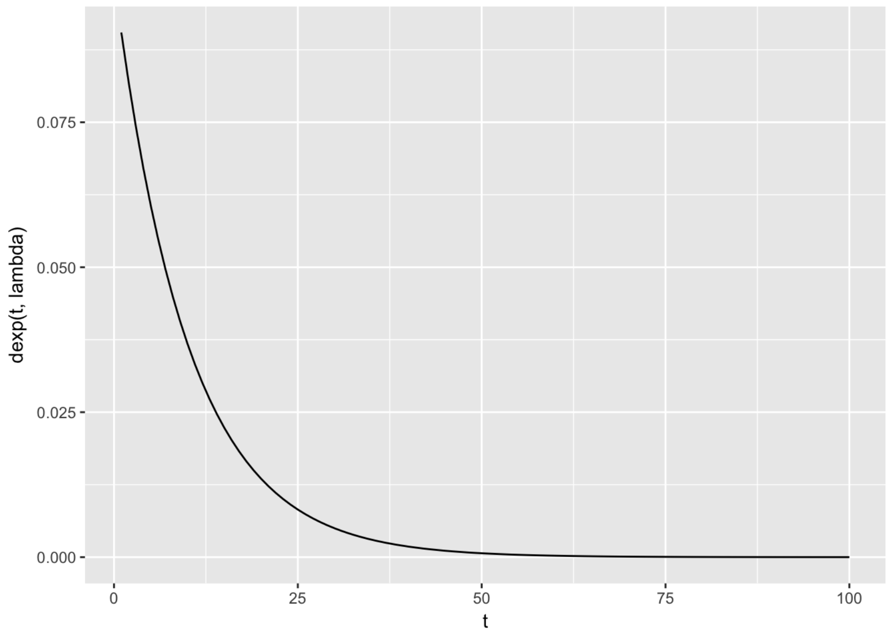


Now we have a graph with a smooth line through its data points, better representing the beta distribution.

Exponential Distribution

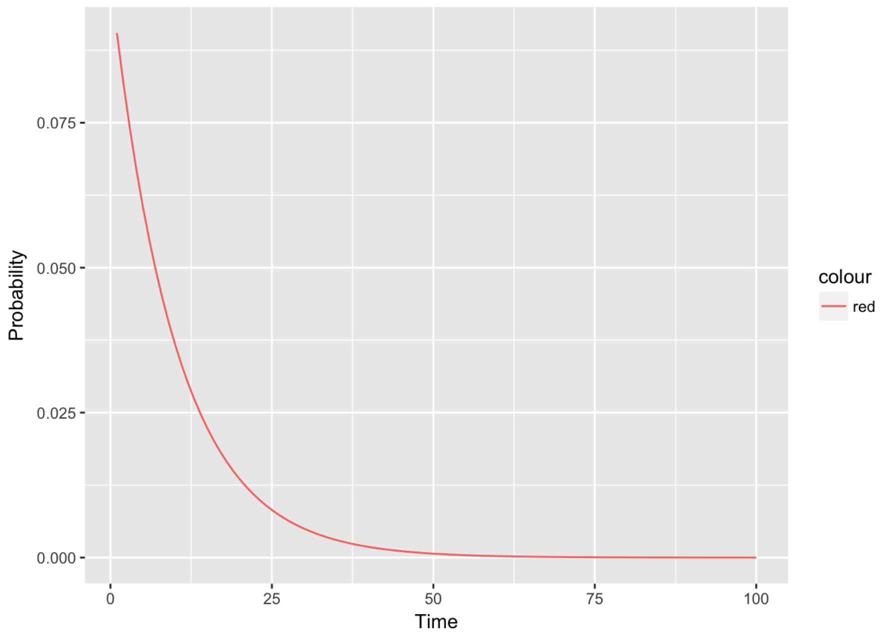
Moving on, we have the exponential distribution. The exponential distribution is used to model the time between independent events in a Poisson process (a series of independent events occurring continuously at a constant average rate). It has the equation λe^{-λt} where λ > 0. We will first create a line graph representing the exponential distribution.

```
lambda <- 0.1
t <- (1:100)
qplot(x = t, y = dexp(t,lambda), geom = 'line')
```



This graph is a good model of the exponential distribution, but it still isn't very pretty. To make it look nicer, we can alter the `xlab`, `ylab`, and `color` arguments.

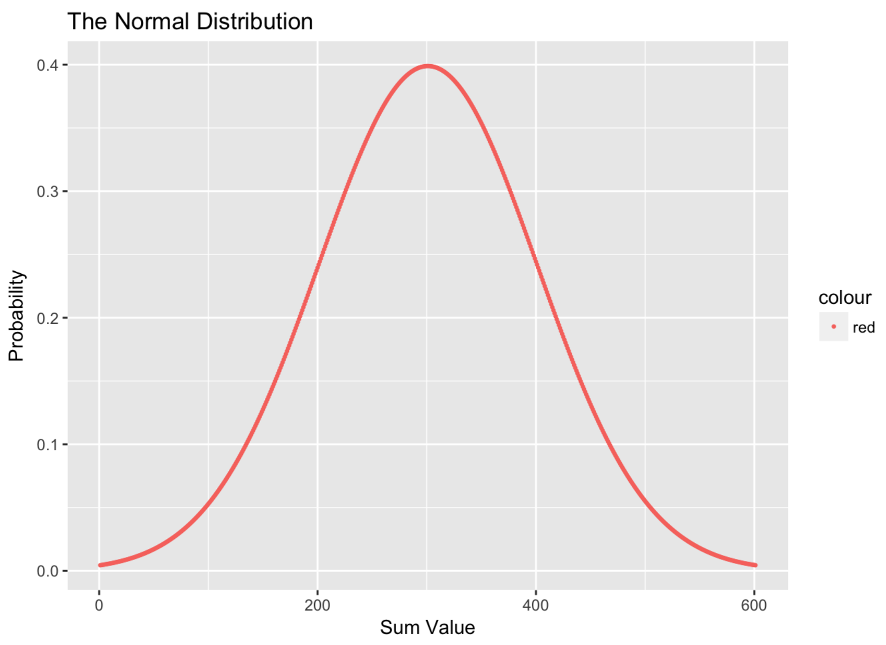
```
qplot(x = t, y = dexp(t,lambda), geom = 'line', xlab = 'Time', ylab = 'Probability', color = 'red')
```



The Normal Distribution

The distribution we are going to use for our final graph is one of the most recognizable in statistics: the normal distribution. This distribution is well known because of the Central Limit Theorem, in which if a sufficient number of independent random variables are added, their sum will approach the normal distribution. Our final graph will utilize everything learned above, but also include a title, and a thicker line. To do this we will use the `main` and `size` arguments. In the size argument, use `I(x)` to specify a numeric multiple of the default size. For example, `size = I(3)` will output at 3x the default size. Size can also be linked to other variables in order to display even more data on a single graph. This graph does not need to use the `geom` argument, as we will use sufficiently many points to make it appear almost continuous, using it as a representation of the Central Limit Theorem. This normal distribution represents the sum of the number of heads in 600 tosses, or more generally, the sum of 600 indicators with a 50% probability of success.

```
normdis <- dnorm(seq(-3, 3, 0.01))
qplot(y = normdis, xlab = 'Sum Value', ylab = 'Probability', main = 'The Normal Distribution', size = I(0.5), colo
r = 'red')
```



Conclusion

This post here shows how to use ggplot2, and more particularly the `qplot` function within it to quickly create representations of probability distributions. This also tries to use some logic in the presentation of data, with continuous data being represented by lines, and discrete data through points, and shows the importance of thinking about the data you are trying to represent.

References

- https://bitesizebio.com/23003/my-10-favorite-r-packages-and-the-cool-things-you-can-do-with-them/
- https://www.statmethods.net/advgraphs/probability.html
- https://en.wikibooks.org/wiki/R_Programming/Probability_Distributions
- https://www.statmethods.net/advgraphs/ggplot2.html
- http://rpubs.com/sinhrks/plot_dist
- http://t-redactyl.io/blog/2016/03/creating-plots-in-r-using-ggplot2-part-9-function-plots.html
- rs/ggplot2_intro/

Loading [MathJax]/jax/element/mml/optable/GreekAndCoptic.js

Loading [MathJax]/jax/element/mml/optable/GreekAndCoptic.js