

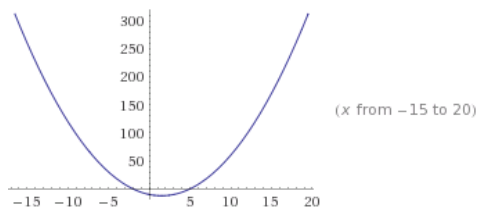
## Join the Stack Overflow Community

Stack Overflow is a community of 6.4 million programmers, just like you, helping each other.  
Join them; it only takes a minute:

[Sign up](#)

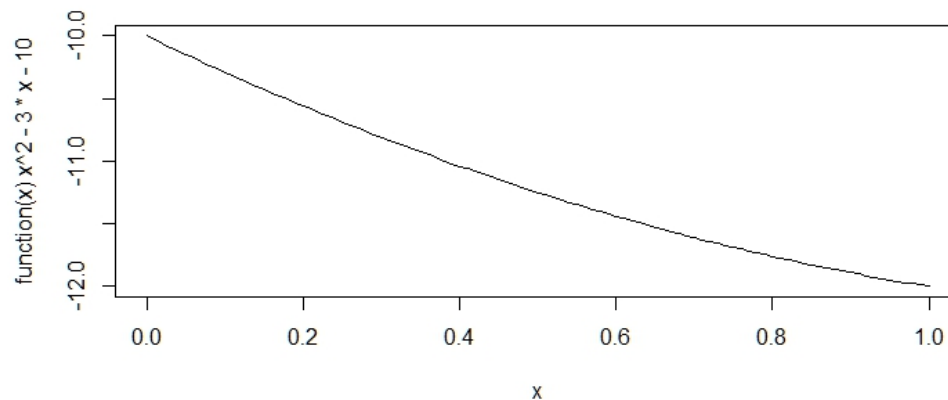
## Plot a polynomial function like on the wolframalpha-website so that it is easy to understand

Plotting a function in the wolfram-alpha-website looks like this:



<http://www.wolframalpha.com/link>

Plotting the same function in R looks like this:



```
plot( function(x) x^2 - 3*x - 10 )
```

The default plot from Wolfram is much easier to understand. I think this is because it shows the x-axis (at  $y=0$ ), and centers the parabola.

I am not good enough at math to just look at the formula of a function and see where I should center the plot, and I am plotting the functions to learn about how different functions create different lines, so I need this centering to be done automatically, because otherwise I might misunderstand a plot.

Is it possible to create the Wolfram-plot *automatically* i.e. without me telling R where it would be sensible to center the plot?

r function plot wolframalpha

edited Nov 27 '13 at 20:13

asked Nov 27 '13 at 19:44



Rasmus Larsen  
477 1 10 35

- 2 there is definitely not a built-in way to do this in base R. It's an interesting challenge to design an algorithm that automatically figures out the interesting range of a function, for some definition of "interesting" ... – Ben Bolker Nov 27 '13 at 21:06

The `polynom` package offers some functionality here. – [mnel](#) Nov 27 '13 at 22:05

- 2 Picking domains and ranges automatically is challenging. Lee Wilkinson proposes one approach in [cs.uic.edu/~wilkinson/Publications/plotfunc.pdf](http://cs.uic.edu/~wilkinson/Publications/plotfunc.pdf) – [hadley](#) Nov 27 '13 at 22:13

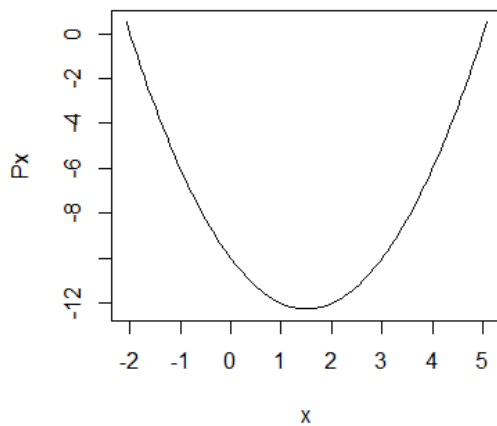
### 3 Answers

The `polynom` package will create some sensible defaults.

eg.

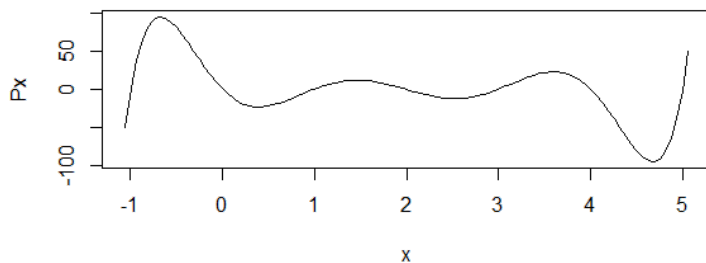
```
library(polynom)

# your polynomial (coefficients in ascending powers of x order)
p <- polynomial(c(-10,-3,1))
plot(p)
```



# a more complicated example, a polynomial crossing the x axis at -1,0,1,2,3,4,5

```
p2 <- poly.calc(-1:5)
p2
# -120*x + 154*x^2 + 49*x^3 - 140*x^4 + 70*x^5 - 14*x^6 + x^7
plot(p2)
```



answered Nov 27 '13 at 22:24



[mnel](#)

69.9k 8 165 174

You can set the desired interval to plot over, as described in `?plot.function`. Also see `curve` and `abline`.

```
plot(function(x) x^2 - 3*x - 10, -15, 15); abline(h=0,v=0,lty=3)
```

or

```
curve(x^2 - 3*x - 10, -15, 15); abline(h=0,v=0,lty=3)
```

answered Nov 27 '13 at 19:50



Aaron

22.2k 3 38 79

Thanks. I've edited the question so it is more clear that I am looking for a way to do this automatically. – [Rasmus Larsen](#) Nov 27 '13 at 20:14

- 2 As far as I know, R does not have built-in way to do this automatically. However, if you have some idea of what would be "sensible", a function could be written to change the plotting interval automatically. – [Aaron](#) Nov 27 '13 at 20:58

This is quite an old post but I was trying to fit a polynomial curve based on the coefficients of my model.

The original in base R :

```
plot(y ~ x)
curve(3*x - 2*x^2 + 2*x^3) ## random coefficients for easy example
```

I use ggplot2 - so I wanted to use the curve generated from the coefficients rather than a + geom\_smooth (This also works but I prefer the curve below)

```
bestfit <- geom_smooth(method = "loess", se = T, size = 1)

ggplot2
+ bestfit
```

Instead I created a function with the coefficients above

```
test
test <- function(x) {3*x - 2*x^2 + 2*x^3}
```

I have then added it as a layer in the ggplot

```
ggplot2
+ stat_function(fun = test)
```

It gives me the same curve as the base plot function but I can add all the additional layers in ggplot

edited Nov 4 '15 at 10:40



Tunaki

59.5k 18 93 127

answered Nov 4 '15 at 10:33



Julie

1

The code does not produce any ggplot (only a base plot). – [Rasmus Larsen](#) Aug 26 at 8:49