Week 1 Module 5: Curves

CSCI E-5a Fall 2021

Let's clear the environment:

```
rm( list = ls() )
```

Module Overview and Learning Objectives

In this module, we'll develop methods for graphing mathematical curves.

- In Section 1, we'll learn about the `curve()` function.
- In Section 2, we'll see how to graph polynomial functions.
- In Section 3, we'll learn how to draw reference lines for a graph.
- In Section 4, we'll learn how to graph exponential functions.
- In Section 5, we'll see how to create a legend for a graph.

When you've completed this module, you'll be able to:

- Use the `curve()` function to create a new graphics plot and draw a curve on it.
- Add a curve to a pre-existing graphics plot.
- Create graphs of polynomial functions
- Create graphs of exponential functions
- Add a legend to a plot

We'll also meet two new functions:

- The `curve()` function
- The `legend()` function

All right! Let's get started by learning about the `curve()` function.

Section 1: The `curve()` Function

> **Main Idea:** *We can draw curves of mathematical functions by using the `curve()` function.*

In this section, we'll learn about the `curve()` function.

So far, we've drawn points and lines by providing explicit specifications of the coordinates.

The built-in R function `curve()` is another approach to drawing geometric objects.

With the `curve()` function, we provide a mathematical expression for a function $f(x)$, and then R plots that function.

In the simplest form of the `curve()` function, we specify one input argument, named `expr`, which is an algebraic expression in the variable x.

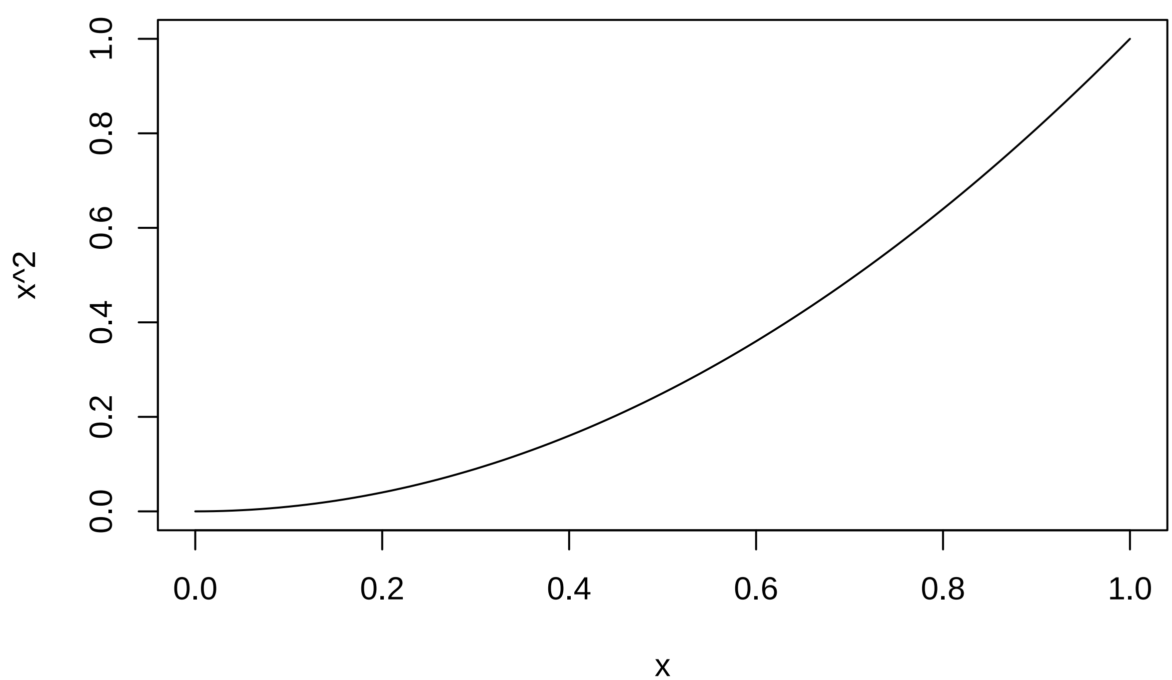
We'll explore algebraic expressions in greater detail in the next lecture, but here's one simple rule to get us started:

- To raise an expression to a power, use the ^ symbol.

Thus, to raise the expression `x` to the power 2, we write `x^2`.

Let's draw a plot of the function $f(x) = x^2$:

```
curve(
    expr = x^2
)
```



Notice in this example that the `curve()` function created a graphics plot.

So the `curve()` function is similar to the `plot()` function in that it can create a graphics plot.

In this respect the `curve()` function is different from the `points()` and `lines()` functions, which cannot create a graphics plot and require a pre-existing graphics plot.

However, the `curve()` function is different from the `plot()` function because we can use it to add curves to a pre-existing graphics plot.

Recall that the `plot()` function *always* creates a new graphics plot, so we can't use it to layer new graphics objects onto a pre-existing plot.

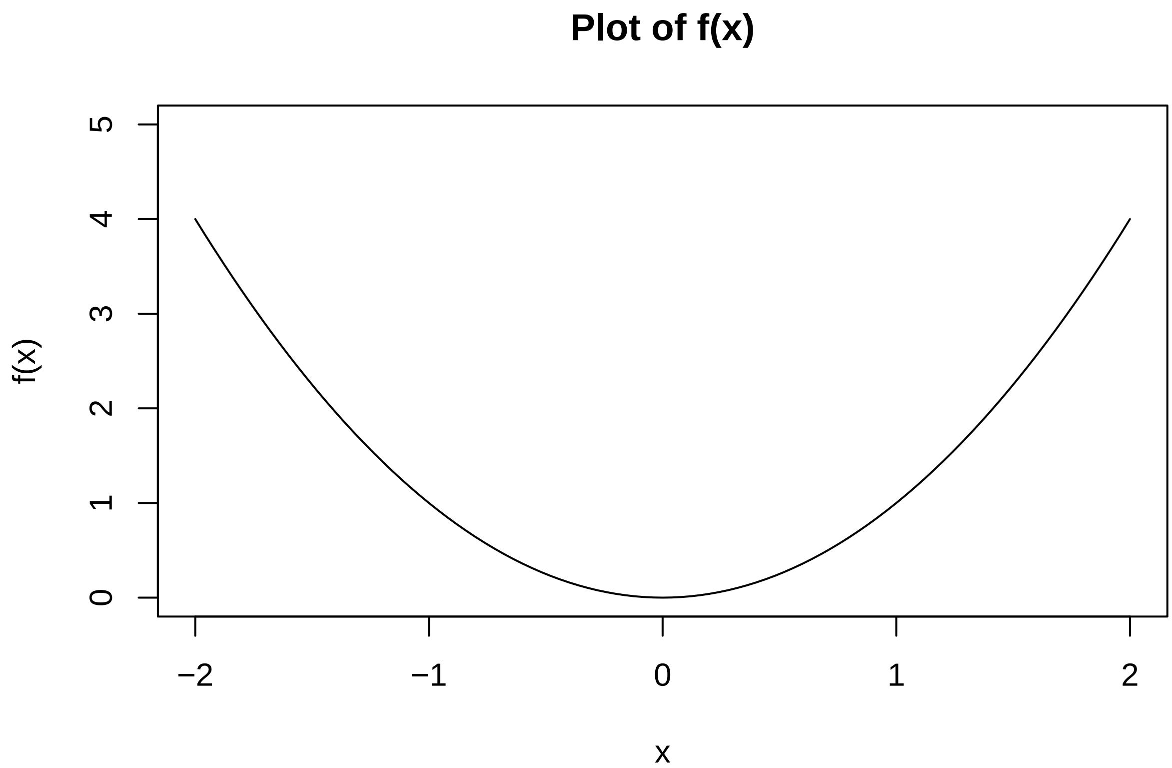
In other words, we can't "turn off" the feature of the `plot()` that creates a new graphics plot.

That's the whole point of the `plot()` function – it creates a new graphics plot!

With the `curve()` function we *can* turn off this feature, and thus we can add a curve to a pre-existing graphics plot without creating a new graphics plot.

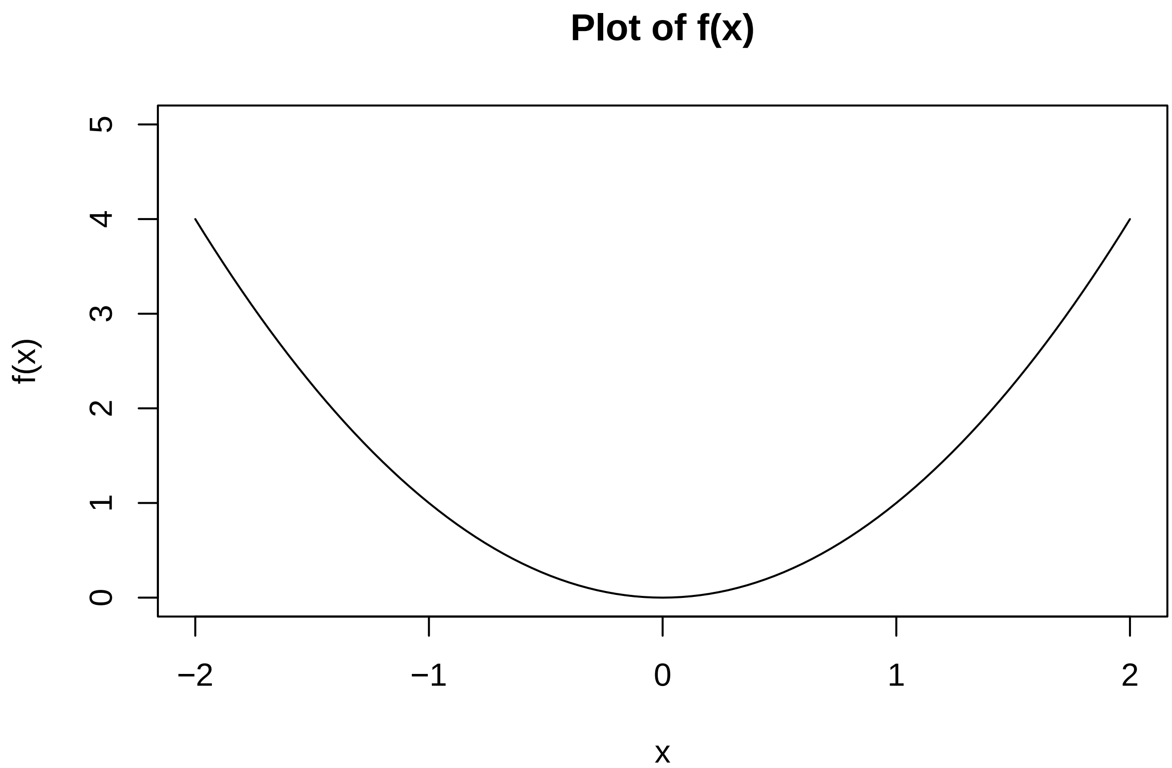
We can use the `curve()` function with a pre-existing plot by using the `add = TRUE` option.

```
plot(
    x = NULL,
    xlim = c(-2, 2),
    ylim = c(0, 5),
    main = "Plot of f(x)",
    xlab = "x",
    ylab = "f(x)"
)

# Now we can add a curve to the plot:

curve(
    expr = x^2,
    add = TRUE   # Remember this!
)
```



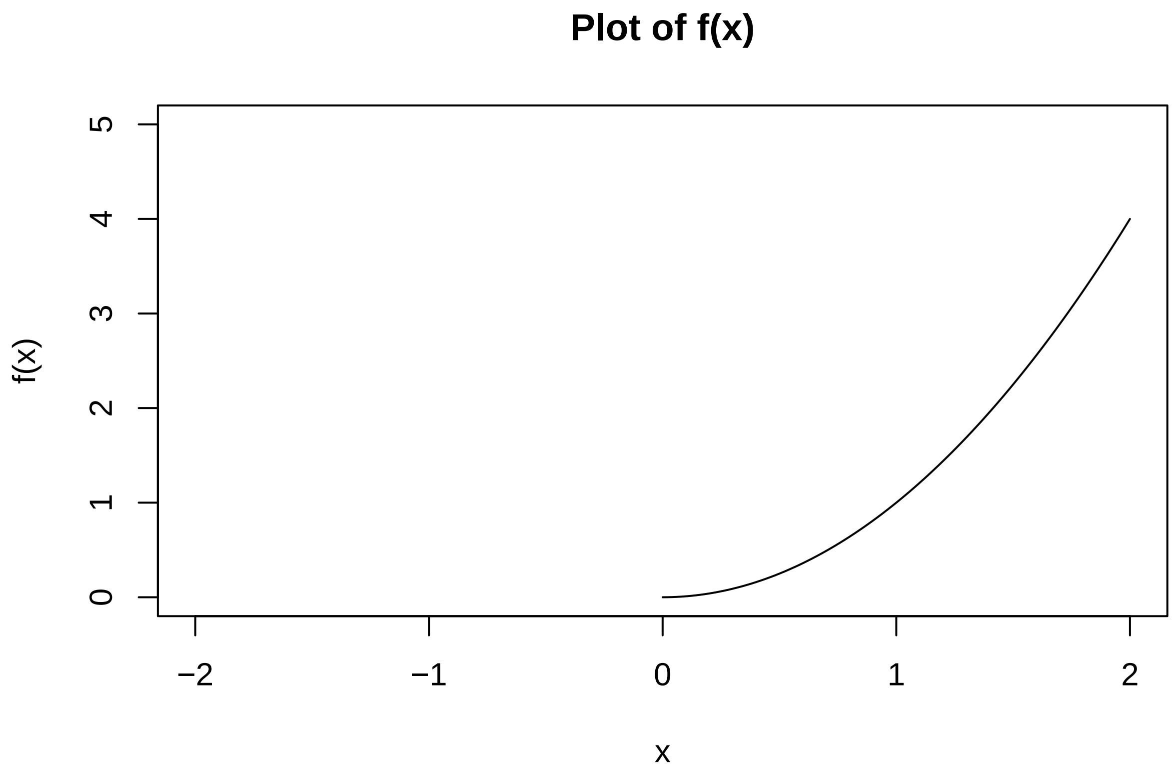
In fact, since the `curve()` function can create a new display, we can use all the standard options for modifying this:

```
curve(
    expr = x^2,
    xlim = c(-2, 2),
    ylim = c(0, 5),
    main = "Plot of f(x)",
    xlab = "x",
    ylab = "f(x)"
)
```



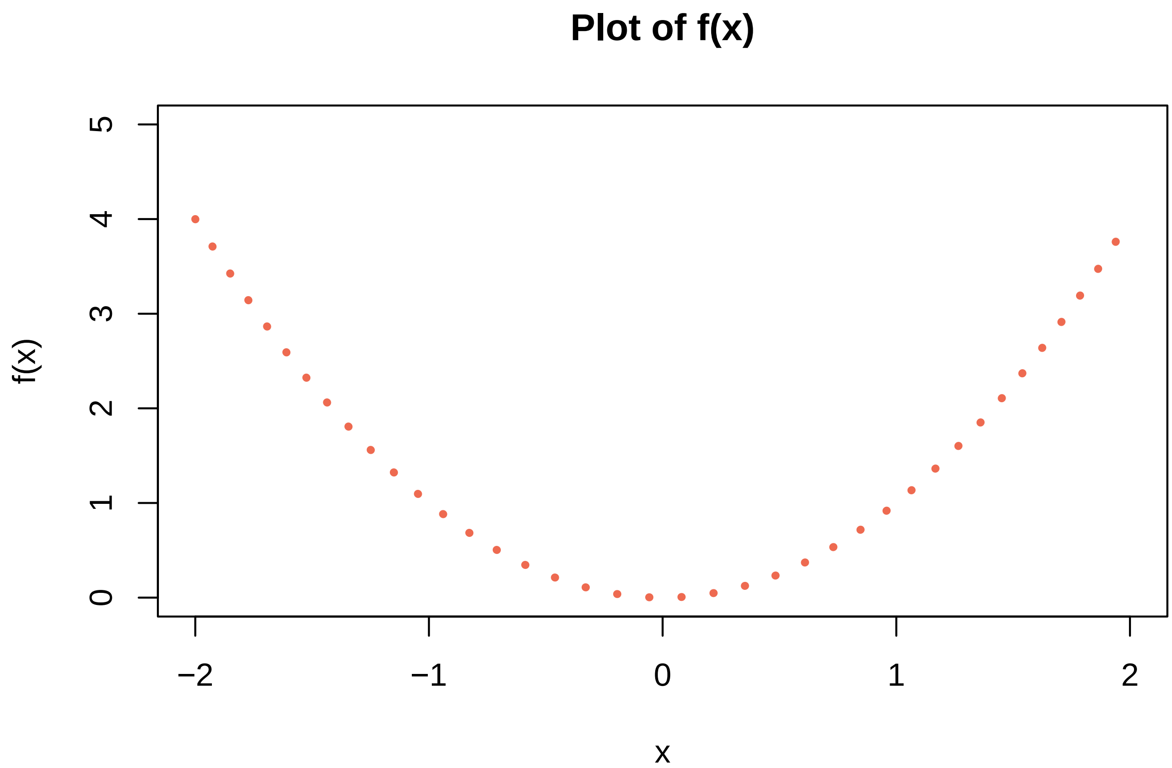
On the other hand, if you first create an empty graphics plot with the `plot()` function and then use an `xlim` argument inside the `curve()` function, you can specify different ranges for the graphics plot and the curve itself:

```
# Create an empty plotting region with no data

plot(
    x = NULL,
    xlim = c(-2, 2),
    ylim = c(0, 5),
    main = "Plot of f(x)",
    xlab = "x",
    ylab = "f(x)"
)


# Now draw the curve, but on a limited range of x values

curve(
    expr = x^2,
    xlim = c(0, 2),
    add = TRUE
)
```



When we use the `curve()` function we are effectively drawing a line, so we can modify the curve using our standard options for lines:

```
plot(
    x = NULL,
    xlim = c(-2, 2),
    ylim = c(0, 5),
    main = "Plot of f(x)",
    xlab = "x",
    ylab = "f(x)"
)

curve(
    expr = x^2,
    lty = "dotted",
    lwd = 4,
    col = "coral2",
    add = TRUE
)
```

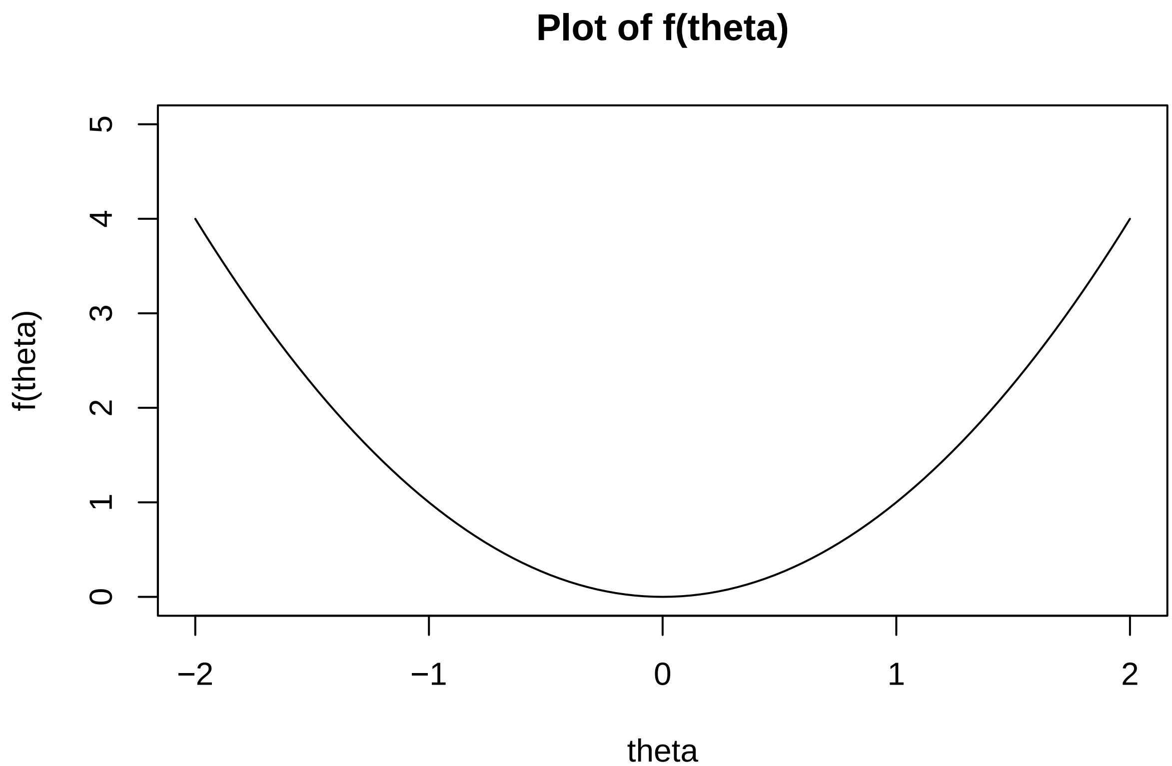


The first argument to `curve()` must be an expression that has the variable `x`.

R will then plot this function by evaluating the expression across the range of `x`.

If we want to plot across a different variable, we can use the `xname` option to specify the input variable.

```
plot(
    x = NULL,
    xlim = c(-2, 2),
    ylim = c(0, 5),
    main = "Plot of f(theta)",
    xlab = "theta",
    ylab = "f(theta)"
)

curve(
    expr = theta^2,
    xname = "theta",
    add = TRUE
)
```



So that's how to use the `curve()` function to plot mathematical functions.

Now let's see how to use this with polynomial functions.

Section 2: Graphing Polynomial Functions

Main Idea: *We can graph polynomial functions.*

In this section we'll learn how to graph polynomial functions.

To precisely define the concept of a polynomial, we can first define a "power term", which is a variable (typically x), raised to a fixed power, multiplied by a constant. For instance, this is a power term:

$$f(x) \;=\; 3 \cdot x^2$$

Here we start with the variable x, raise it to the fixed power 2, and then multiply it by the constant 3.

Thus, $f(x)$ satisfies our definition of a power term.

A polynomial is a sum of power terms.

For instance, consider the function $g(x)$:

$$g(x) \;=\; 3x^2 - 7x + 4$$

We can think of $g(x)$ as the sum of the three power terms: $3x^2$, $(-7x)$, and 4:

$$g(x) \;=\; (3x^2) + (-7x) + (4x^0)$$

Now let's see how to graph polynomial functions by using the built-in R function `curve()`.

To use the `curve()` function for polynomial functions, the first input argument is an expression for the polynomial in the variable x.

Previously, we saw that we could use the symbol '^' to express exponentiation (that means raising to a power).

Here are two new rules for algebraic expressions in R:

- To add two expressions, use the `+` symbol.
- To multiply two expressions, use the `*` symbol.

There's an important point about expressing a polynomial when using the `curve()` function: multiplication and exponentiation must be explicitly indicated.

In standard mathematical notation, we typically do *not* indicate multiplication explicitly.

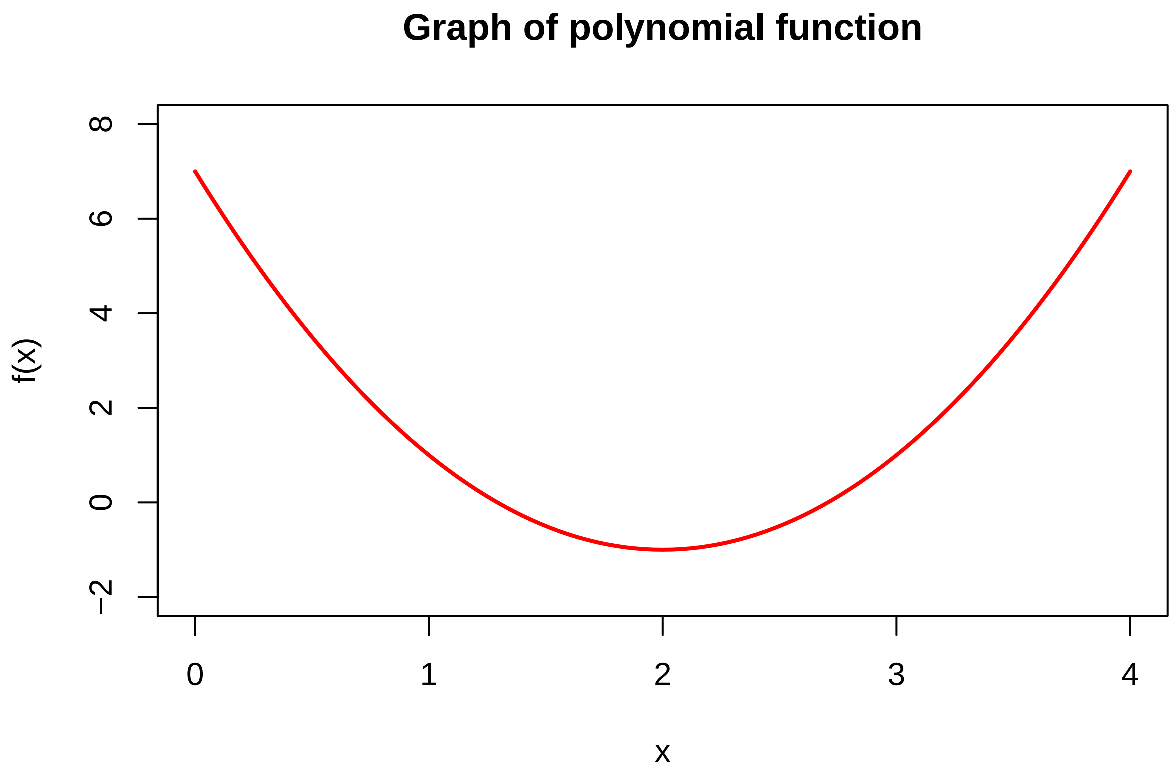
Thus, when we use standard mathematical notation to express the idea of 2 times the variable x, we simply write $2x$.

In R, this won't work, and we have to explicitly indicate the multiplication by writing $2 * x$.

Likewise, we have to explicitly indicate exponentiation by using the caret symbol `^`.

Let's graph the function $f(x) = 2x^2 - 8x + 7$:

```
curve(
    expr = 2*x^2 - 8*x + 7,
    xlim = c(0, 4),
    ylim = c(-2, 8),
    main = "Graph of polynomial function",
    xlab = "x",
    ylab = "f(x)",
    lty = "solid",
    lwd = 2,
    col = "red"
)
```



So that's how to graph polynomial functions by using the `curve()` function.

Now let's consider how to draw reference lines for our graphs.

Exercise 5.1: Cubic polynomial

Plot the function $f(x) = -2x^3 + 8x - 4$ for x ranging from -3 to +3.

Solution

```
# Type your answer here
```

Section 3: Reference Lines

> **Main Idea:** *We can add reference lines to a graph.*

In this section, we'll learn how to draw reference lines for a graph.

In mathematics, we refer to the horizontal line $y = 0$ as the "x-axis".

Likewise, the vertical line $x = 0$ is called the "y-axis".

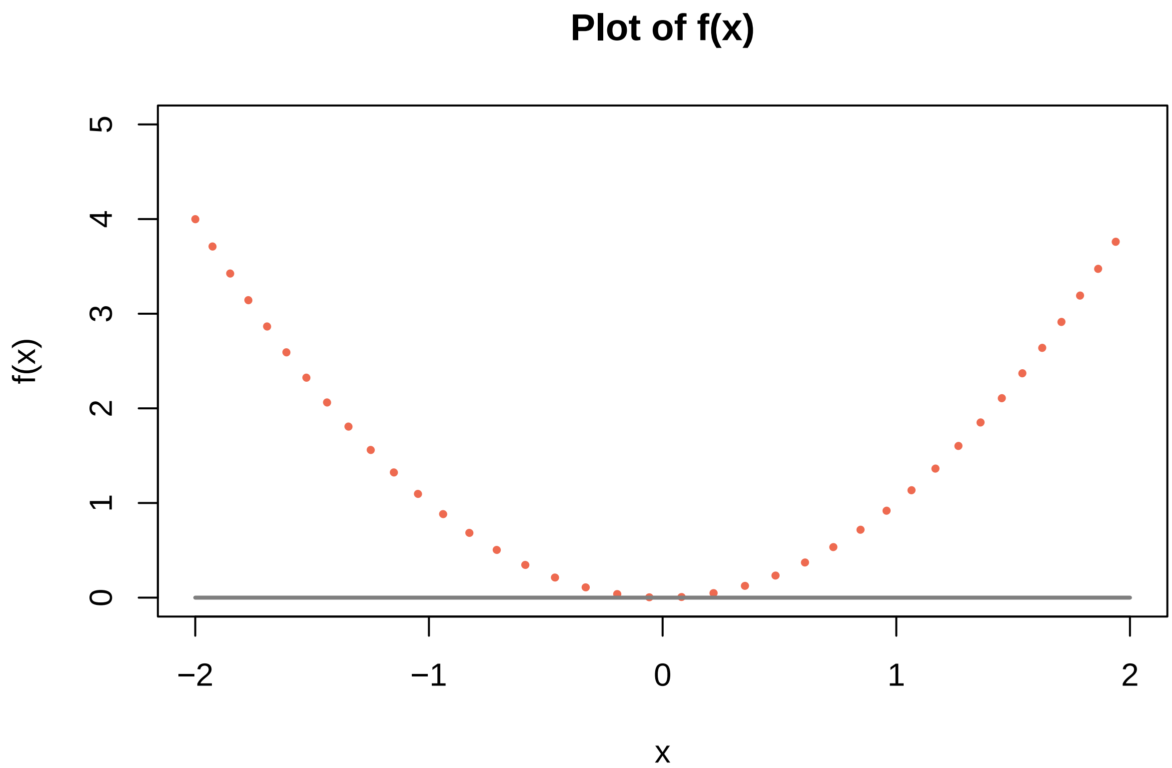
In R, the term "x-axis" means the thing at the bottom of the plotting region with tick marks and labels for the tick marks.

In general, the x-axis for the plotting region will not be drawn on the line $y = 0$, and instead there is a small offset.

This isn't necessarily a bad thing, but it would nice to see the line $y = 0$ on the graph just as a reference.

Of course, we can draw in the line $y = 0$ by using the `segments()` or the `lines()` functions.

```r
curve(
    expr = x^2,
    xlim = c(-2, 2),
    ylim = c(0, 5),
    main = "Plot of f(x)",
    xlab = "x",
    ylab = "f(x)",
    lty = "dotted",
    lwd = 4,
    col = "coral2"
)

# Draw a horizontal reference line

segments(
    x0 = -2,
    y0 = 0,
    x1 = 2,
    y1 = 0,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)
```



For this course you should always draw in a horizontal reference line if the range of the y-axis contains the value 0, including the endpoints.

In the graph above, the y-axis ranges from 0 to 5, so it does contain the value 0 at its lower endpoint, and thus we draw a horizontal reference line.

We can use the same method to draw in a vertical reference line.

For our plot, the y-axis ranges from 0 to 5, so we want a vertical reference from $y = 0$ to $y = 5$, and this would start at the point $(0, 0)$ and end at the point $(0, 5)$.

Remember: if it's the "vertical" reference axis, that means that the x-coordinate is *always* 0.

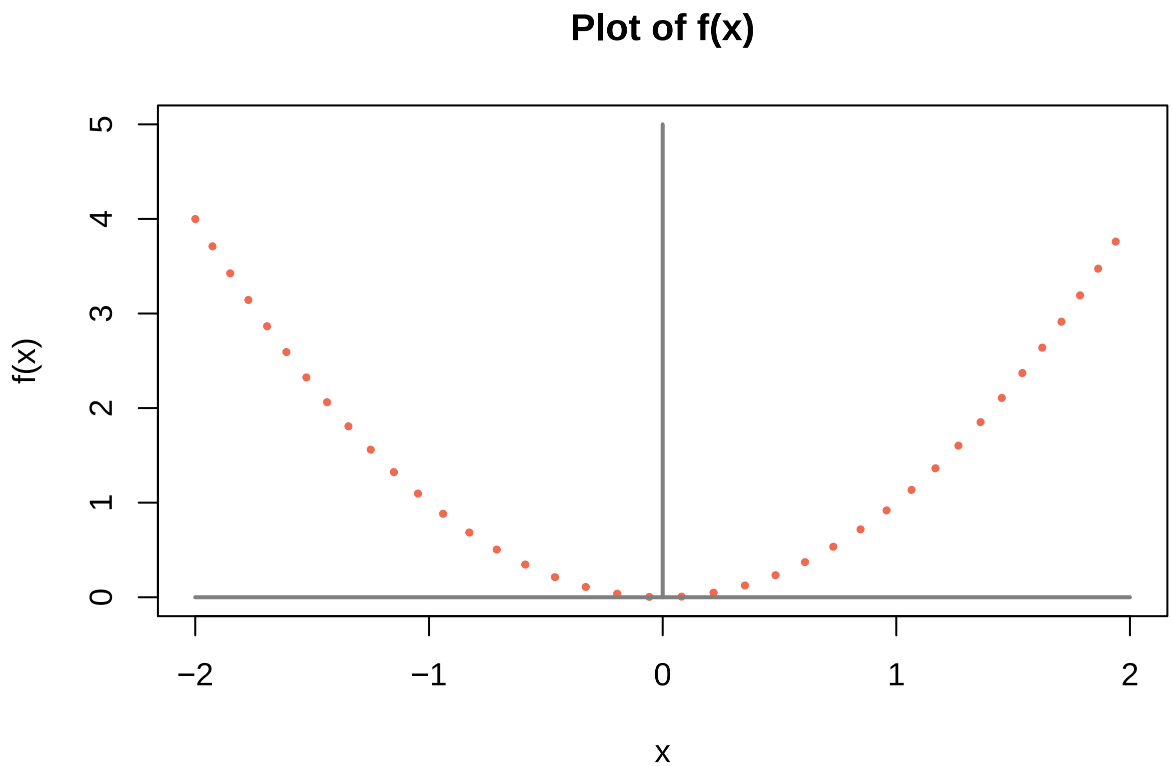
For this course, you should always draw in a vertical reference line if the range of the x-axis contains the value 0, including the endpoints.

In the graph above, the x-axis ranges from -2 to 2, so it does contain the value 0, and thus we draw a vertical reference line.

Let's draw in a vertical reference line for this graph:

```
curve(
    expr = x^2,
    xlim = c(-2, 2),
    ylim = c(0, 5),
    main = "Plot of f(x)",
    xlab = "x",
    ylab = "f(x)",
    lty = "dotted",
    lwd = 4,
```

```
    col = "coral2"
)

# Here's the horizontal reference line:

segments(
    x0 = -2,
    y0 = 0,
    x1 = 2,
    y1 = 0,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)


# Here's the vertical reference line:

segments(
    x0 = 0,
    y0 = 0,
    x1 = 0,
    y1 = 5,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)
```



There's a subtle issue with this graph.

We drew the curve first, and then drew the horizontal and vertical reference lines.

That means that we overlaid the reference lines on top of the curve.

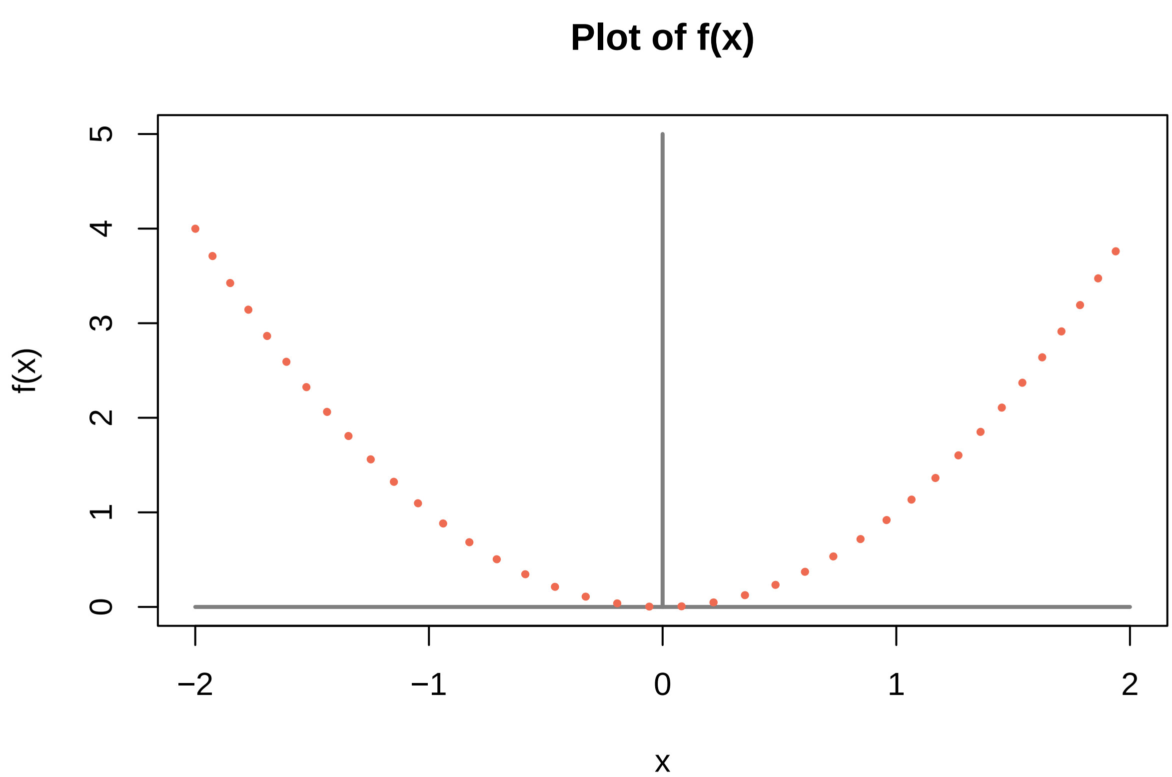
But I think it's nicer to draw the curve on top of the reference lines.

We can create an empty plotting region with no data, and then explicitly specify the sequence by which the graphics objects are drawn.

Thus, we can control which objects are overlaid on other objects.

```
plot(
    x = NULL,
    xlim = c(-2, 2),
    ylim = c(0, 5),
    main = "Plot of f(x)",
    xlab = "x",
    ylab = "f(x)"
)


segments(
    x0 = -2,
    y0 = 0,
    x1 = 2,
    y1 = 0,
    lty = "solid",
```

```
    lwd = 2,
    col = "gray50"
)


segments(
    x0 = 0,
    y0 = 0,
    x1 = 0,
    y1 = 5,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)

curve(
    expr = x^2,
    lty = "dotted",
    lwd = 4,
    col = "coral2",
    add = TRUE
)
```



This isn't a huge issue with this particular graph, but it does illustrate the idea of layering objects onto a graph, and how we can control the display by creating an empty plotting region with no data and then adding objects to the plot.

By the way, I've been calling these lines the "horizontal reference line" and the "vertical reference line", but you should be clear that this terminology is unique to our course, and nobody outside of this course ever speaks like this.

So that's how to draw reference lines for a graph.

Now let's learn how to graph exponential functions by using the `curve()` function.

Exercise 5.2: Cubic polynomial with reference lines

Plot the function $f(x) = -2x^3 + 8x - 4$ for x ranging from -3 to +3. Include the horizontal and vertical reference lines $y = 0$ and $x = 0$.

Solution

Section 4: Graphing Exponential Functions

> **Main Idea:** *We can graph exponential functions.*

In this section, we'll learn how to graph exponential functions.

Another common mathematical function is the *exponential* function.

The exponential function is defined for all real numbers, and this includes the negative numbers as well as the positive numbers.

It has the algebraic form:

$$f(x) \;=\; e^x$$

Here e is the positive constant $e \approx 2.718282$.

We can also incorporate a rate constant, denoted r:

$$g(x) = e^{rx}$$

Note that r can be negative as well as positive.

To calculate values of the exponential function, R provides the built-in function `exp()`, which takes a single numeric value as its input argument:

```
exp( x = 1 )
```

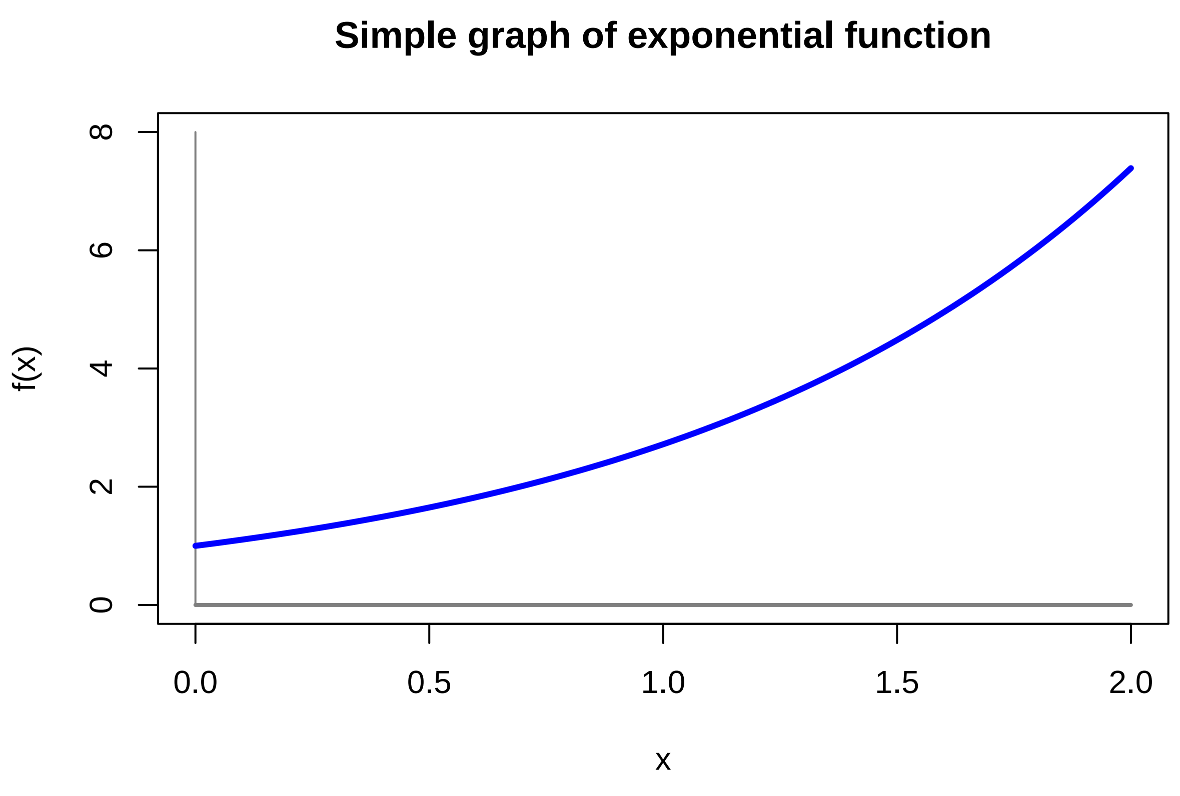
```
## [1] 2.718282
```

Remember, the first input argument for the `curve()` function is an expression using the variable x, thus to plot an exponential function we'll need to use an expression of the form `exp(x)`.

Let's make a graph of the exponential function $f(x) = e^x$:

```
plot(
    x = NULL,
    xlim = c(0, 2),
    ylim = c(0, 8),
    main = "Simple graph of exponential function",
    xlab = "x",
    ylab = "f(x)"
)


# Draw the horizontal reference line y = 0:

segments(
    x0 = 0,
    y0 = 0,
    x1 = 2,
    y1 = 0,
    lwd = 2,
    lty = "solid",
    col = "gray50"
)
```

```
# Draw the vertical reference line x = 0:

segments(
    x0 = 0,
    y0 = 0,
    x1 = 0,
    y1 = 8,
    lwd = 1,
    lty = "solid",
    col = "gray50"
)


curve(
    expr = exp(x),
    lwd = 3,
    lty = "solid",
    col = "blue",
    add = TRUE
)
```

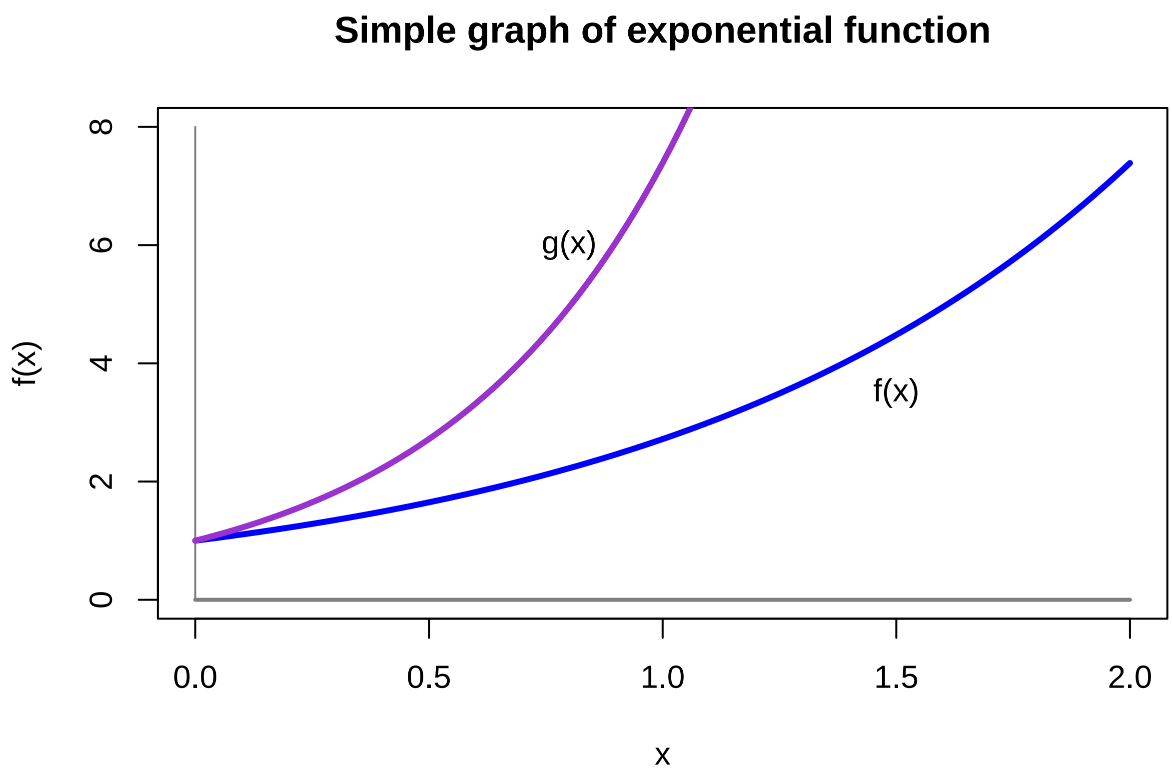


Now let's add in the function $g(x) = e^{2x}$.

```
plot(
    x = NULL,
    xlim = c(0, 2),
```

```
    ylim = c(0, 8),
    main = "Simple graph of exponential function",
    xlab = "x",
    ylab = "f(x)"
)

# Draw the horizontal reference line y = 0:

segments(
    x0 = 0,
    y0 = 0,
    x1 = 2,
    y1 = 0,
    lwd = 2,
    lty = "solid",
    col = "gray50"
)

# Draw the vertical reference line x = 0:

segments(
    x0 = 0,
    y0 = 0,
    x1 = 0,
    y1 = 8,
    lwd = 1,
    lty = "solid",
    col = "gray50"
)


curve(
    expr = exp(x),
    lwd = 3,
    lty = "solid",
    col = "blue",
    add = TRUE
)

text(
    x = 1.5,
    y = 3.5,
    labels = "f(x)"
)


curve(
    expr = exp(2 * x),
    lwd = 3,
    lty = "solid",
    col = "darkorchid3",
    add = TRUE
)
```

```
text(
    x = 0.8,
    y = 6,
    labels = "g(x)"
)
```



So that's how to graph exponential functions.

Now let's see how to add a legend to a graph.

Exercise 5.3: Negative exponentials

Draw a graph of two functions:

- The function $f(x) = e^{-x}$.
- The function $g(x) = e^{-2x}$.

Draw horizontal and vertical reference lines, and annotate the curves with text.

Solution

Section 5: The `legend()` Function

> **Main Idea:** *We can add a legend to a graph.*

In this section, we'll see how to create a legend for a graph.

In the previous section, we graphed two functions, and the only way that we could distinguish them was to use some text annotation.

That's OK, and I like text annotation.

But it's also nice to provide a legend for the graph.

The built-in R function `legend()` will add a legend to a pre-existing plotting region.

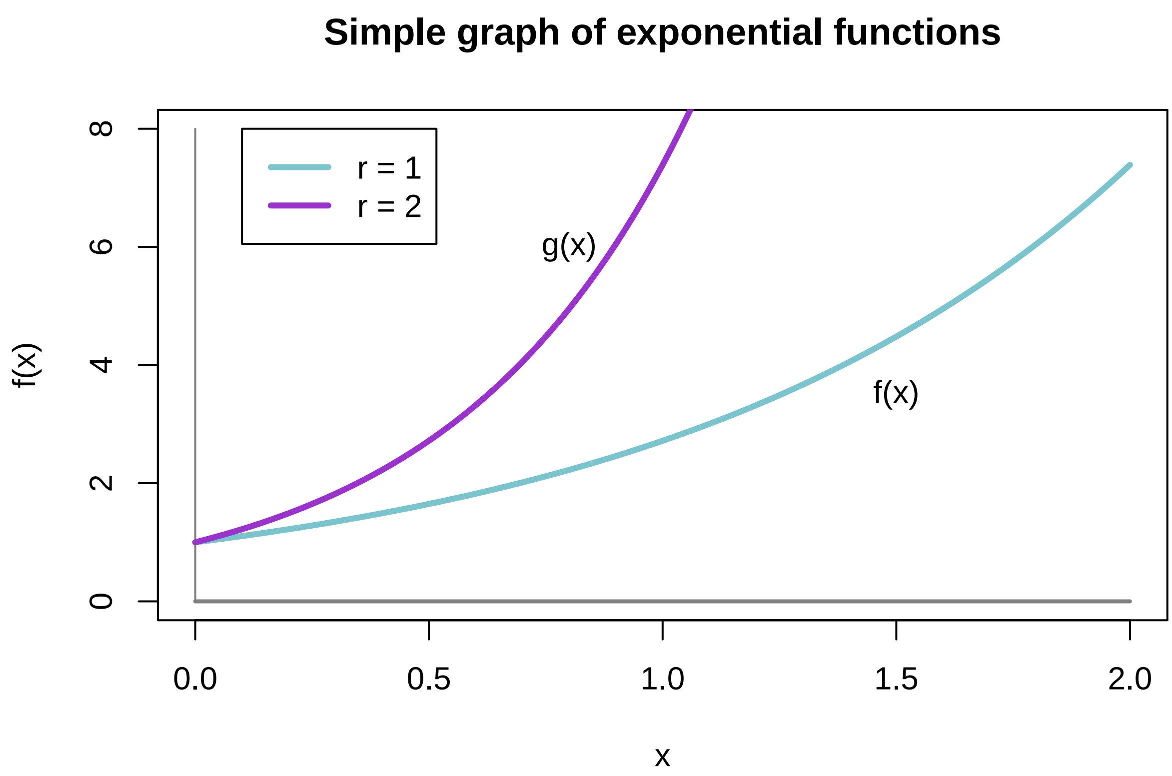
- The first argument, denoted `x`, indicates the x-coordinate of the upper left-hand corner of the legend.
- The second argument, denoted `y`, indicates the y-coordinate of the upper left-hand corner of the legend.
- The third argument, denoted `legend`, is a vector of character strings that are the text of the legend.
- The fourth argument, denoted `lty`, is a character string specification of the line type for the legend.
- The fifth argument, denoted `lwd`, is a character string specification of the line width for the lines of the legend.
- The sixth argument, denoted `col`, is a vector of character strings that are color names for the items in the legend.

That might not make a whole lot of sense to you, but I think it will become much more manageable if you look at some code.

Let's add a legend to our plot of the two exponential functions:

```
plot(
    x = NULL,
    xlim = c(0, 2),
    ylim = c(0, 8),
    main = "Simple graph of exponential functions",
    xlab = "x",
    ylab = "f(x)"
)

# Draw the horizontal reference line

segments(
    x0 = 0,
    y0 = 0,
    x1 = 2,
    y1 = 0,
    lwd = 2,
    lty = "solid",
    col = "gray50"
)

# Draw the vertical reference line
```

```r
segments(
    x0 = 0,
    y0 = 0,
    x1 = 0,
    y1 = 8,
    lwd = 1,
    lty = "solid",
    col = "gray50"
)

# Draw f(x) and annotate with text

curve(
    expr = exp(x),
    lwd = 3,
    lty = "solid",
    col = "cadetblue3",
    add = TRUE
)

text(
    x = 1.5,
    y = 3.5,
    labels = "f(x)"
)

# Draw the graph g(x) and annotate with text

curve(
    expr = exp(2 * x),
    lwd = 3,
    lty = "solid",
    col = "darkorchid3",
    add = TRUE
)

text(
    x = 0.8,
    y = 6,
    labels = "g(x)"
)

# Now draw the legend

legend(
    x = 0.1,
    y = 8,
    legend =
        c( "r = 1", "r = 2" ),
    lty = "solid",
    lwd = 3,
    col = c("cadetblue3", "darkorchid3")
)
```



So that's how to create a legend for a graph.

Now let's see how to graph normal density curves.

Exercise 5.4: Negative exponentials with a legend

Draw a graph of two functions:

- The function $f(x) = e^{-x}$.
- The function $g(x) = e^{-2x}$.

Draw horizontal and vertical reference lines, annotate the curves with text, and include a legend.

Solution

Module Review

In this module, we learned how to graph some special mathematical objects: polynomials, exponential functions, and normal density functions.

- In Section 1, we learned about the `curve()` function.
- In Section 2, we saw how to graph polynomial functions.
- In Section 3, we learned how to draw reference lines for our graphs.
- In Section 4, we learned how to graph exponential functions.
- In Section 5, we saw how to create a legend for a graph.

Now that you've completed this module, you should be able to:

- Use the `curve()` function to create a new graphics plot and draw a curve on it.
- Add a curve to a pre-existing graphics plot.
- Create graphs of polynomial functions
- Create graphs of exponential functions
- Add a legend to a plot

We also met two new functions:

- The `curve()` function
- The `legend()` function

All right! That's it for Module 5: Curves.

In fact, that's all the content for Week 1: Base R Graphics.

Now you can finish Problem Set 1!

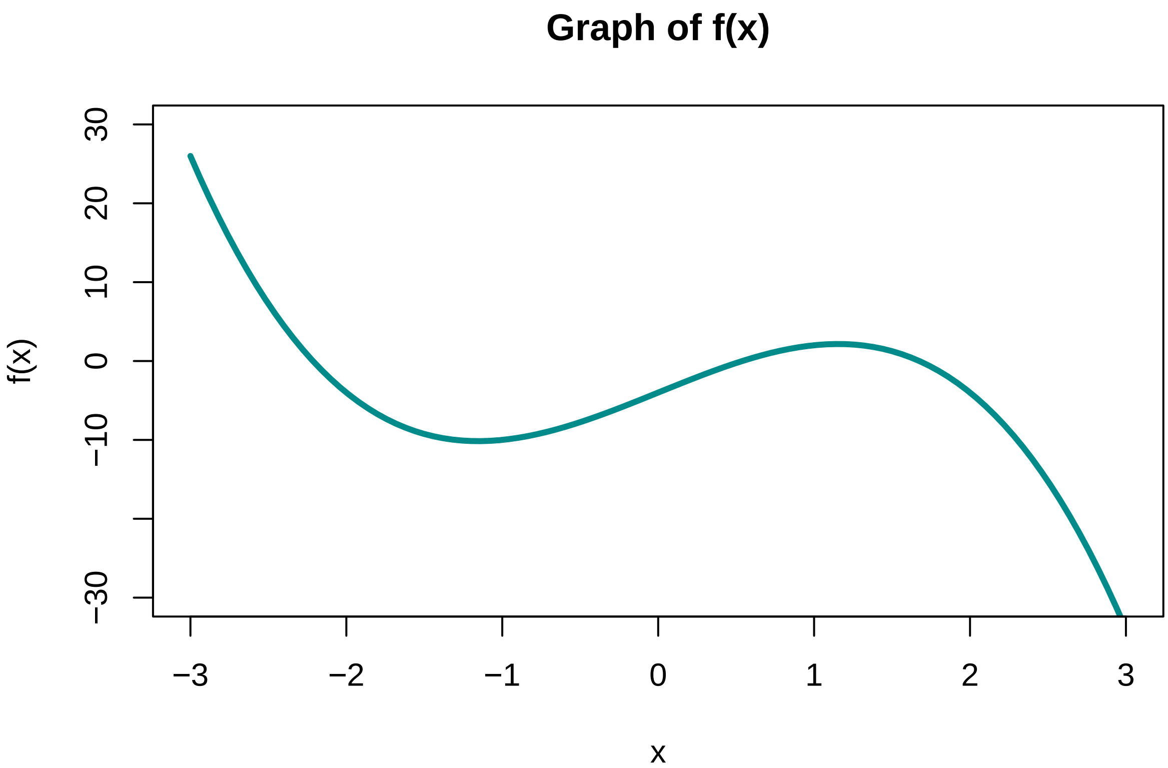
Solutions to the Exercises

Exercise 5.1: Cubic polynomial

Plot the function $f(x) = -2x^3 + 8x - 4$ for x ranging from -3 to +3.

Solution

```
curve(
    -2 * x^3 + 8 * x - 4,
    xlim = c(-3, 3),
    ylim = c(-30, 30),
    main = "Graph of f(x)",
    xlab = "x",
    ylab = "f(x)",
    lty = "solid",
    lwd = 3,
    col = "cyan4"
)
```



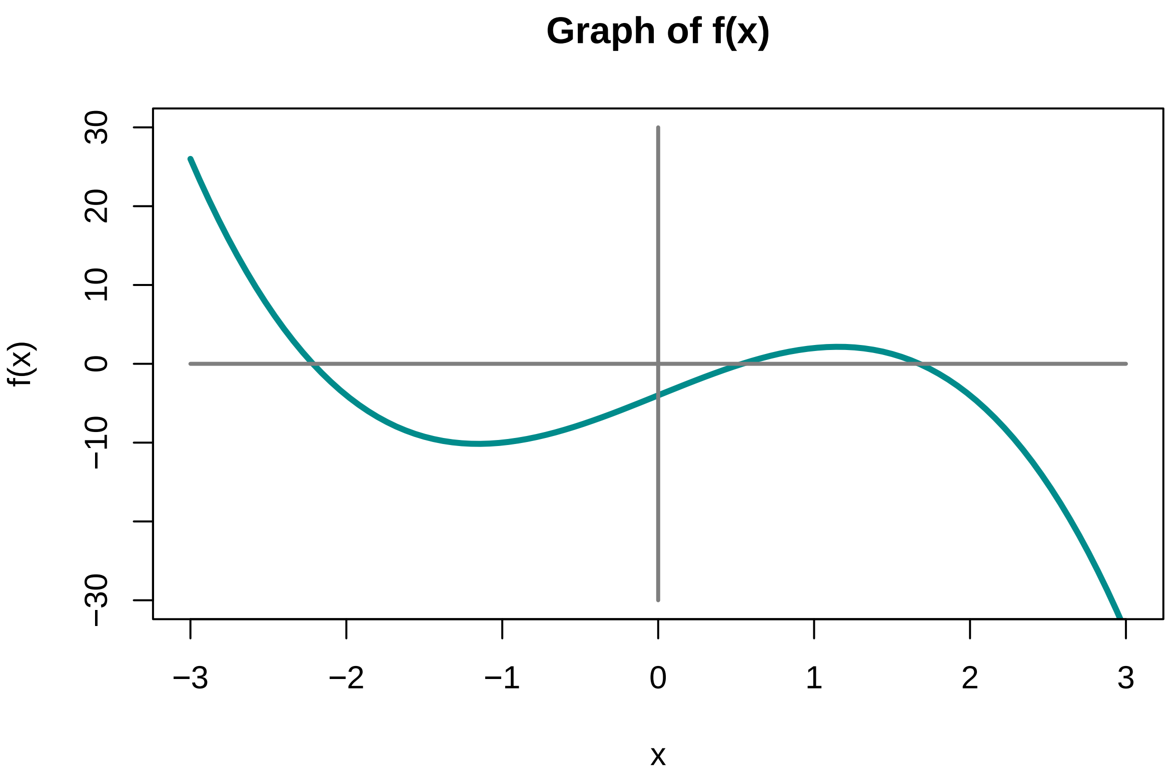
Exercise 5.2: Cubic polynomial with reference lines

Plot the function $f(x) = -2x^3 + 8x - 4$ for x ranging from -3 to +3. Include the horizontal and vertical reference lines $y = 0$ and $x = 0$.

Solution

```
curve(
    -2 * x^3 + 8 * x - 4,
    xlim = c(-3, 3),
    ylim = c(-30, 30),
    main = "Graph of f(x)",
    xlab = "x",
    ylab = "f(x)",
    lty = "solid",
    lwd = 3,
    col = "cyan4"
)

# Draw the horizontal reference line:

segments(
    x0 = -3,
    y0 = 0,
    x1 = 3,
```

```
    y1 = 0,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)


# Draw the vertical reference line:

segments(
    x0 = 0,
    y0 = -30,
    x1 = 0,
    y1 = 30,
    lty = "solid",
    lwd = 2,
    col = "gray50"
)
```



Exercise 5.3: Negative exponentials

Draw a graph of two functions:

- The function $f(x) = e^{-x}$.

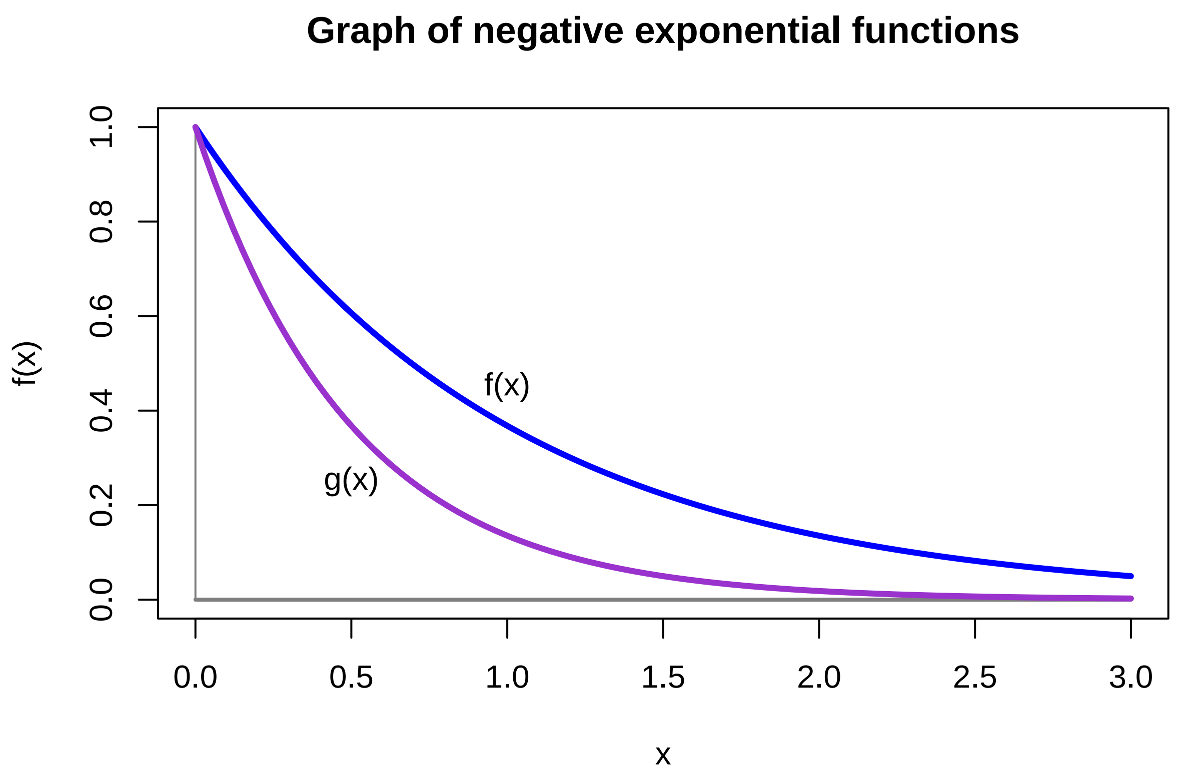
- The function $g(x) = e^{-2x}$.

Draw horizontal and vertical reference lines, and annotate the curves with text.

Solution

```
plot(
    x = NULL,
    xlim = c(0, 3),
    ylim = c(0, 1),
    main = "Graph of negative exponential functions",
    xlab = "x",
    ylab = "f(x)"
)

# Now we can draw in the horizontal reference axis y = 0

segments(
    x0 = 0,
    y0 = 0,
    x1 = 3,
    y1 = 0,
    lwd = 2,
    lty = "solid",
    col = "gray50"
)

# Finally we can draw in the vertical reference axis x = 0

segments(
    x0 = 0,
    y0 = 0,
    x1 = 0,
    y1 = 1,
    lwd = 1,
    lty = "solid",
    col = "gray50"
)


curve(
    exp(-x),
    lwd = 3,
    lty = "solid",
    col = "blue",
    add = TRUE
)

text(
    x = 1,
    y = 0.45,
    labels = "f(x)"
)
```

```
curve(
    exp(-2 * x),
    lwd = 3,
    lty = "solid",
    col = "darkorchid3",
    add = TRUE
)

text(
    x = 0.5,
    y = 0.25,
    labels = "g(x)"
)
```



Exercise 5.4: Negative exponentials with a legend

Draw a graph of two functions:

- The function $f(x) = e^{-x}$.
- The function $g(x) = e^{-2x}$.

Draw horizontal and vertical reference lines, annotate the curves with text, and include a legend.

Solution

```r
plot(
    x = NULL,
    xlim = c(0, 3),
    ylim = c(0, 1),
    main = "Graph of negative exponential functions",
    xlab = "x",
    ylab = "f(x)"
)

# Now we can draw in the horizontal reference axis y = 0

segments(
    x0 = 0,
    y0 = 0,
    x1 = 3,
    y1 = 0,
    lwd = 2,
    lty = "solid",
    col = "gray50"
)

# Finally we can draw in the vertical reference axis x = 0

segments(
    x0 = 0,
    y0 = 0,
    x1 = 0,
    y1 = 1,
    lwd = 1,
    lty = "solid",
    col = "gray50"
)


curve(
    exp(-x),
    lwd = 3,
    lty = "solid",
    col = "blue",
    add = TRUE
)

text(
    x = 1,
    y = 0.45,
    labels = "f(x)"
)


curve(
    exp(-2 * x),
    lwd = 3,
    lty = "solid",
```

```
    col = "darkorchid3",
    add = TRUE
)

text(
    x = 0.5,
    y = 0.25,
    labels = "g(x)"
)


legend(
    x = 2.4,
    y = 1,
    legend =
        c( "r = -1", "r = -2"),
    lty = "solid",
    lwd = 3,
    col =
        c( "blue", "darkorchid3")
)
```