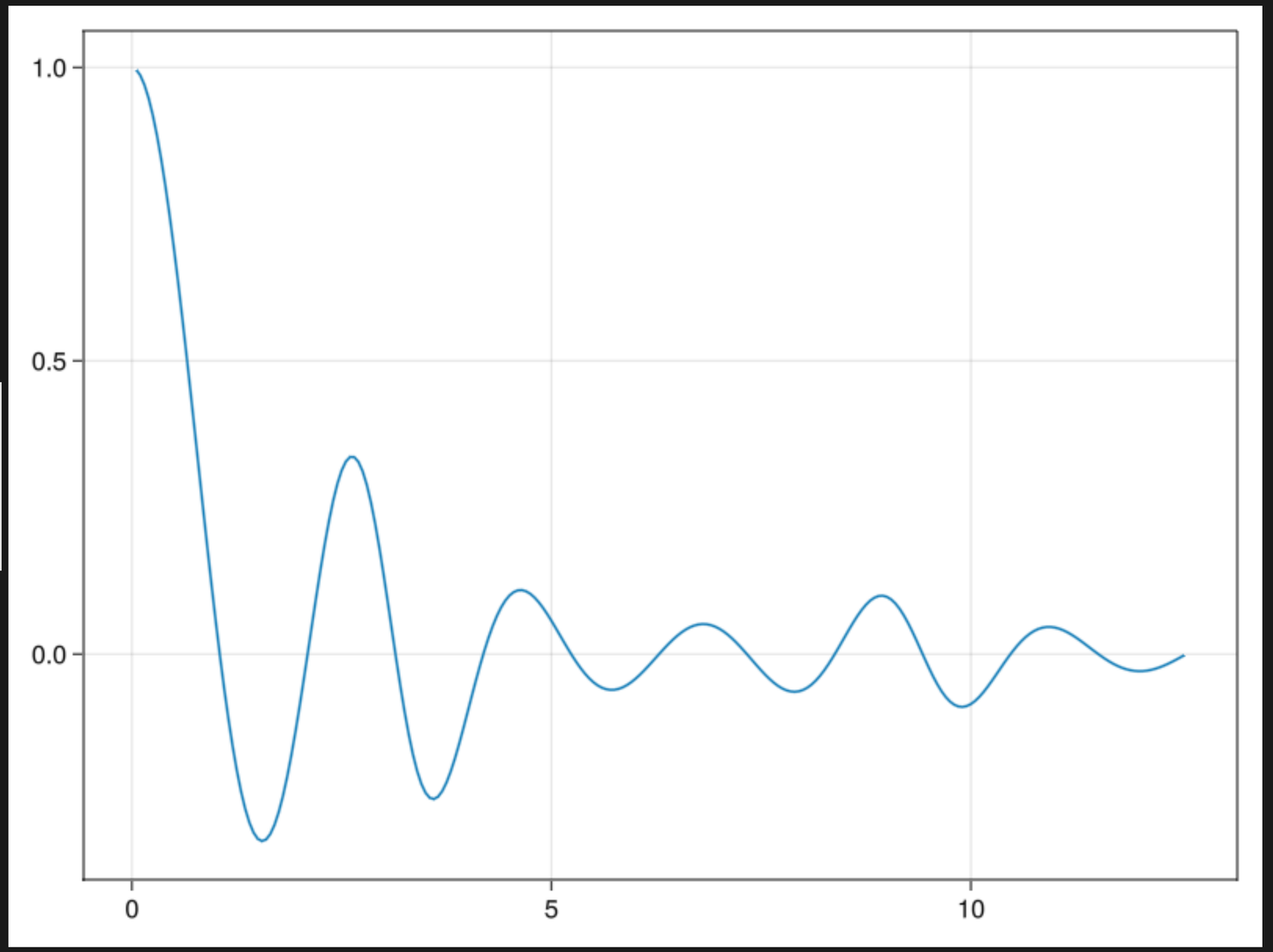
Makie

Basics: Axis

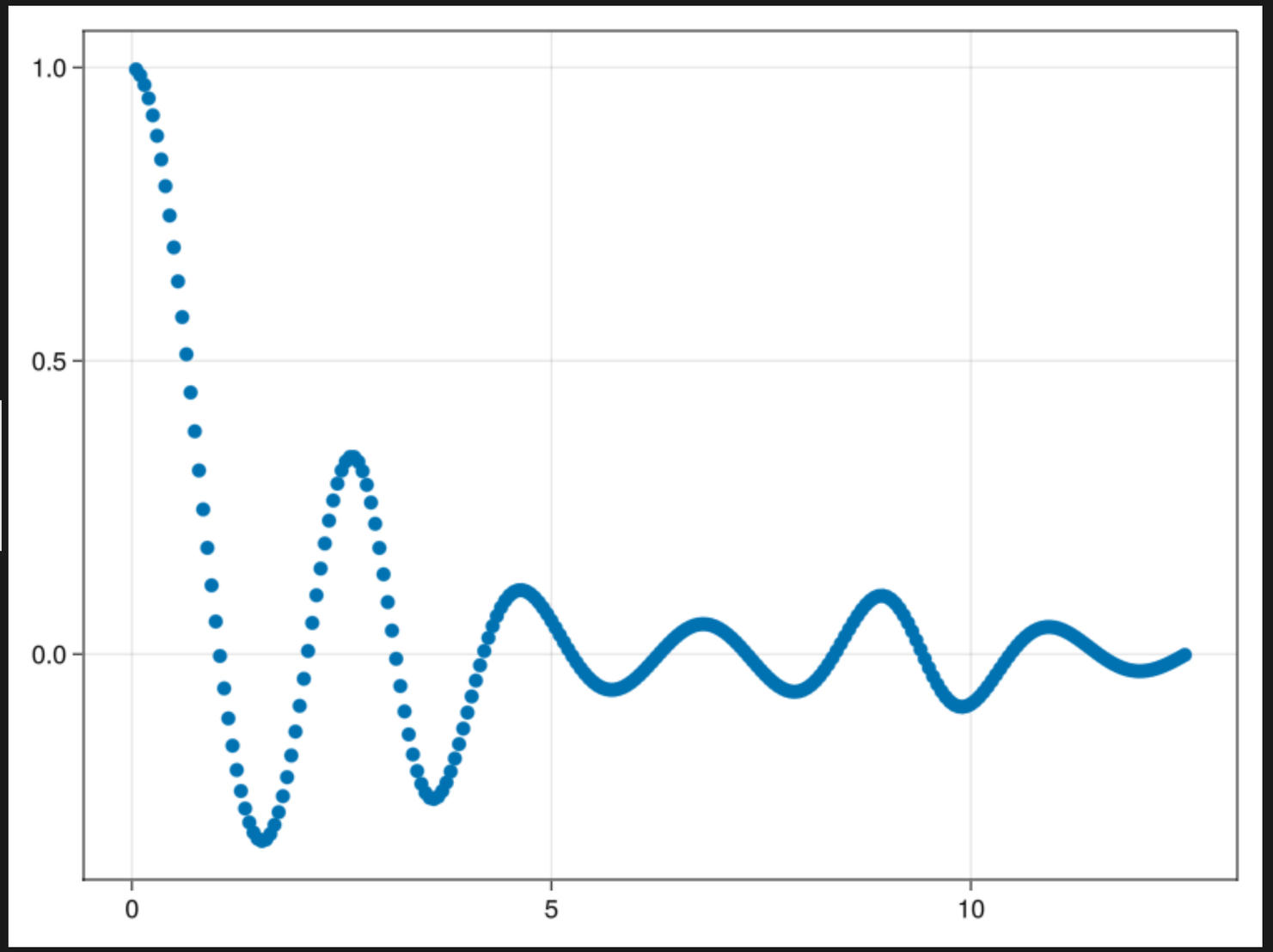
Danisch & Krumbiegel, (2021). Makie.jl: Flexible high-performance data visualization for Julia.
Journal of Open Source Software, 6(65), 3349

Lazaro Alonso @LazarusAlon
MPI-BGI

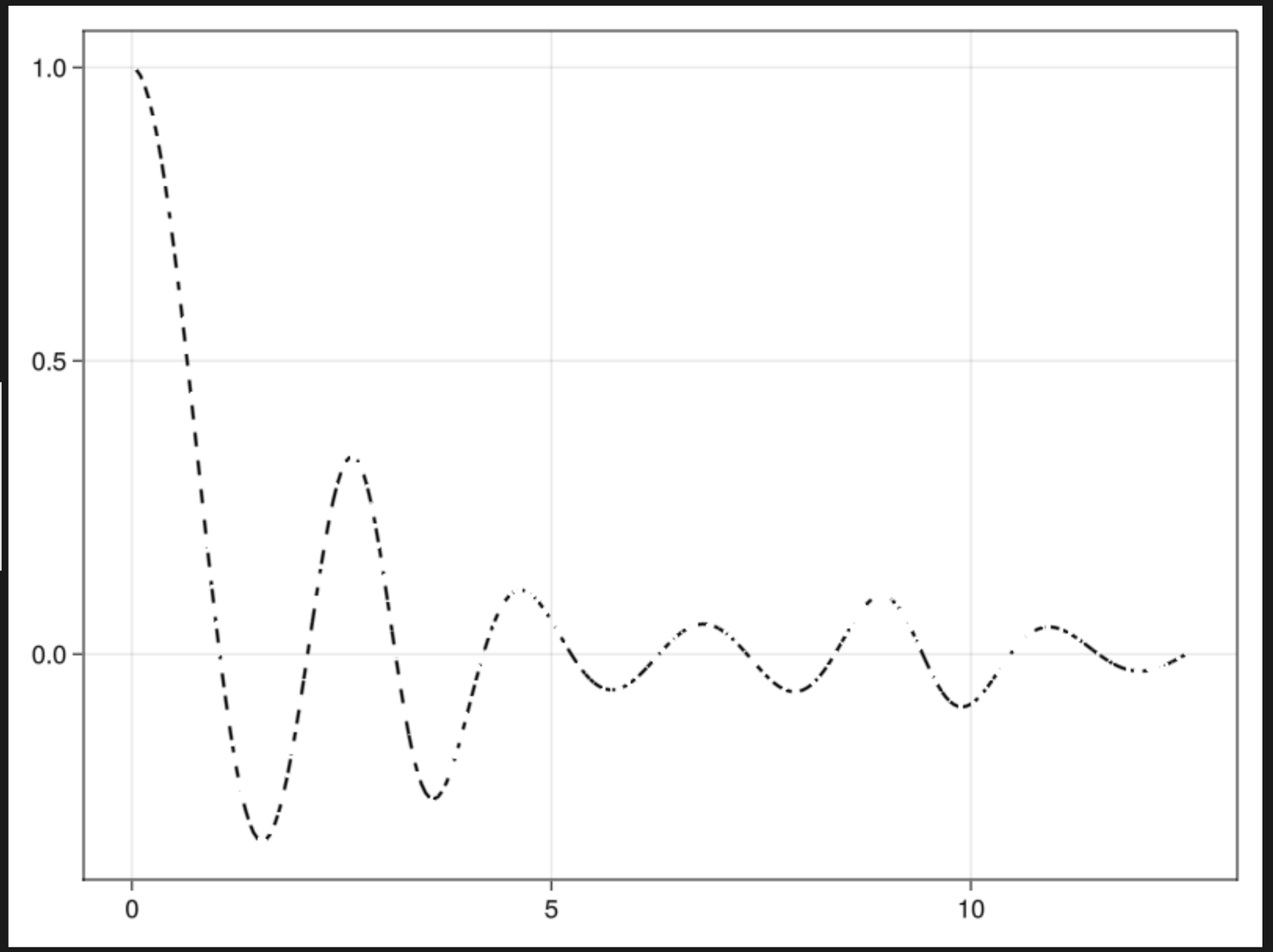
```
using GLMakie, Colors, Random, ColorSchemes
x = 0.05:0.05:4π
y = sin.(3x) ./ (cos.(x) .+ 2) ./ x

lines(x,y)
```



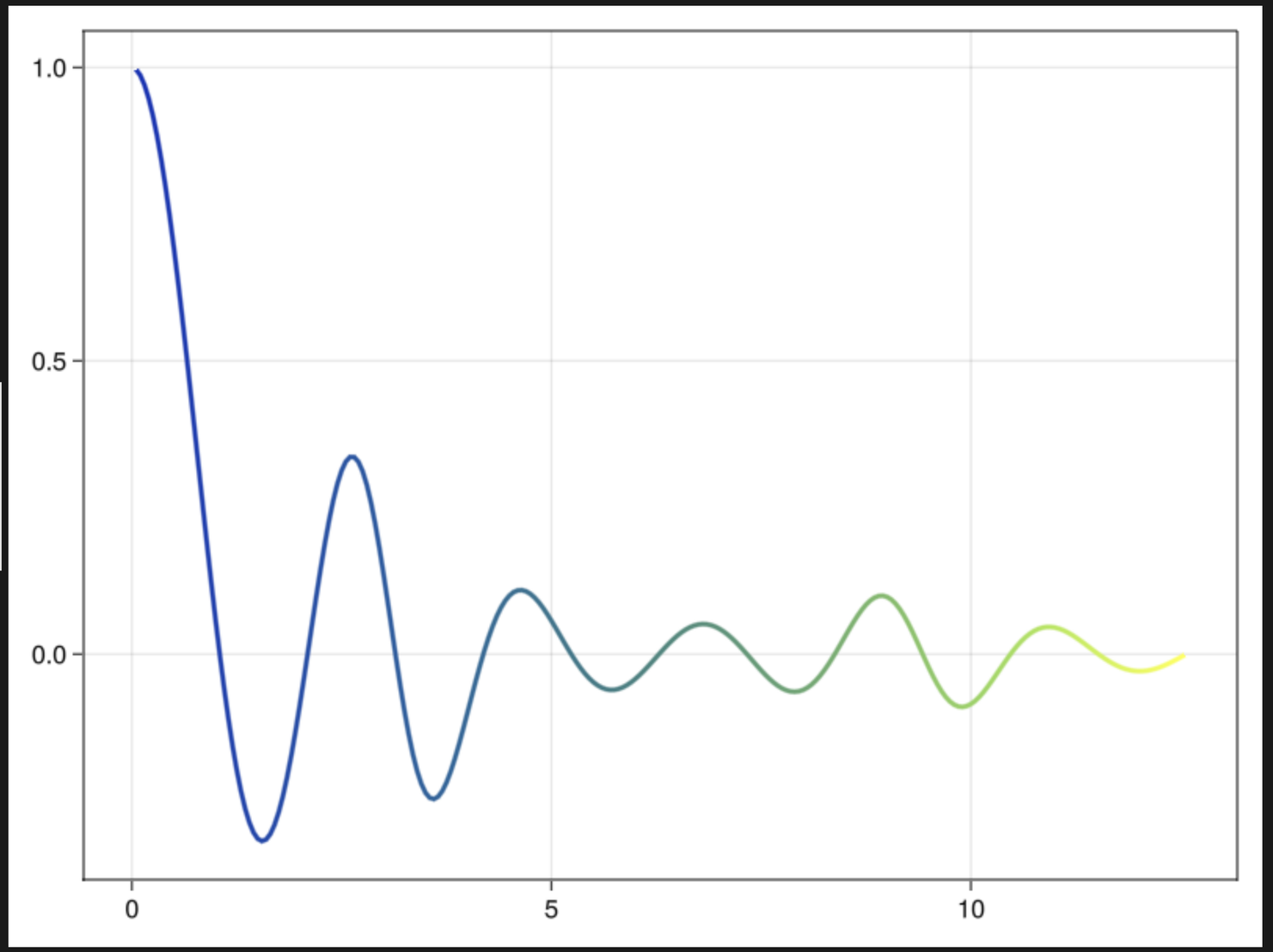
```
x = 0.05:0.05:4π
y = sin.(3x) ./ (cos.(x) .+ 2) ./ x

scatter(x,y)
```



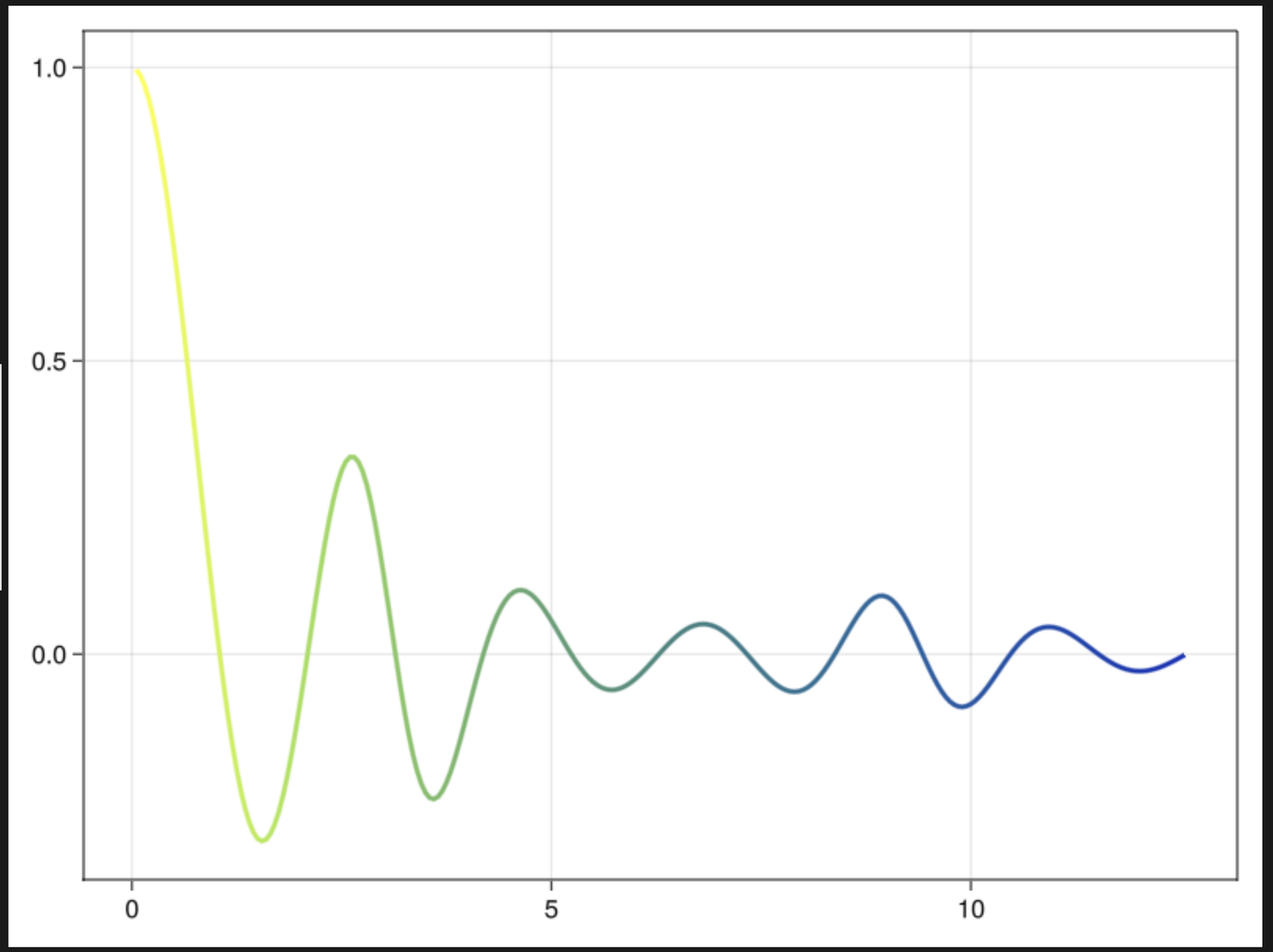
```julia
## Line attributes
x = 0.05:0.05:4π
y = sin.(3x) ./ (cos.(x) .+ 2) ./ x

lines(x, y; color = :black, linewidth = 2, linestyle = :dash)
```



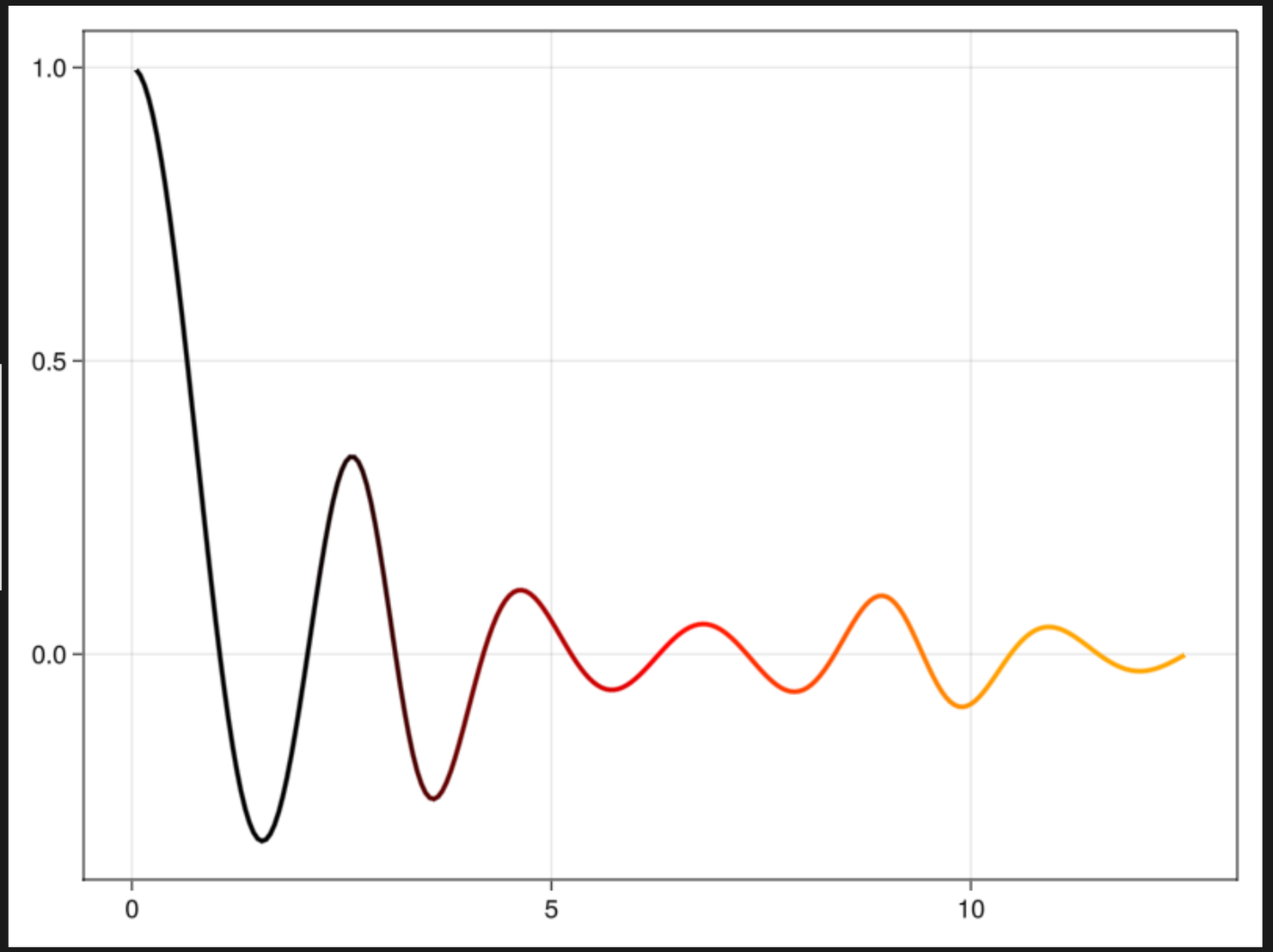
```
x = 0.05:0.05:4π
y = sin.(3x) ./ (cos.(x) .+ 2) ./ x

lines(x, y; color = x, linewidth = 3, linestyle = :solid,
    colormap = :imola)
```



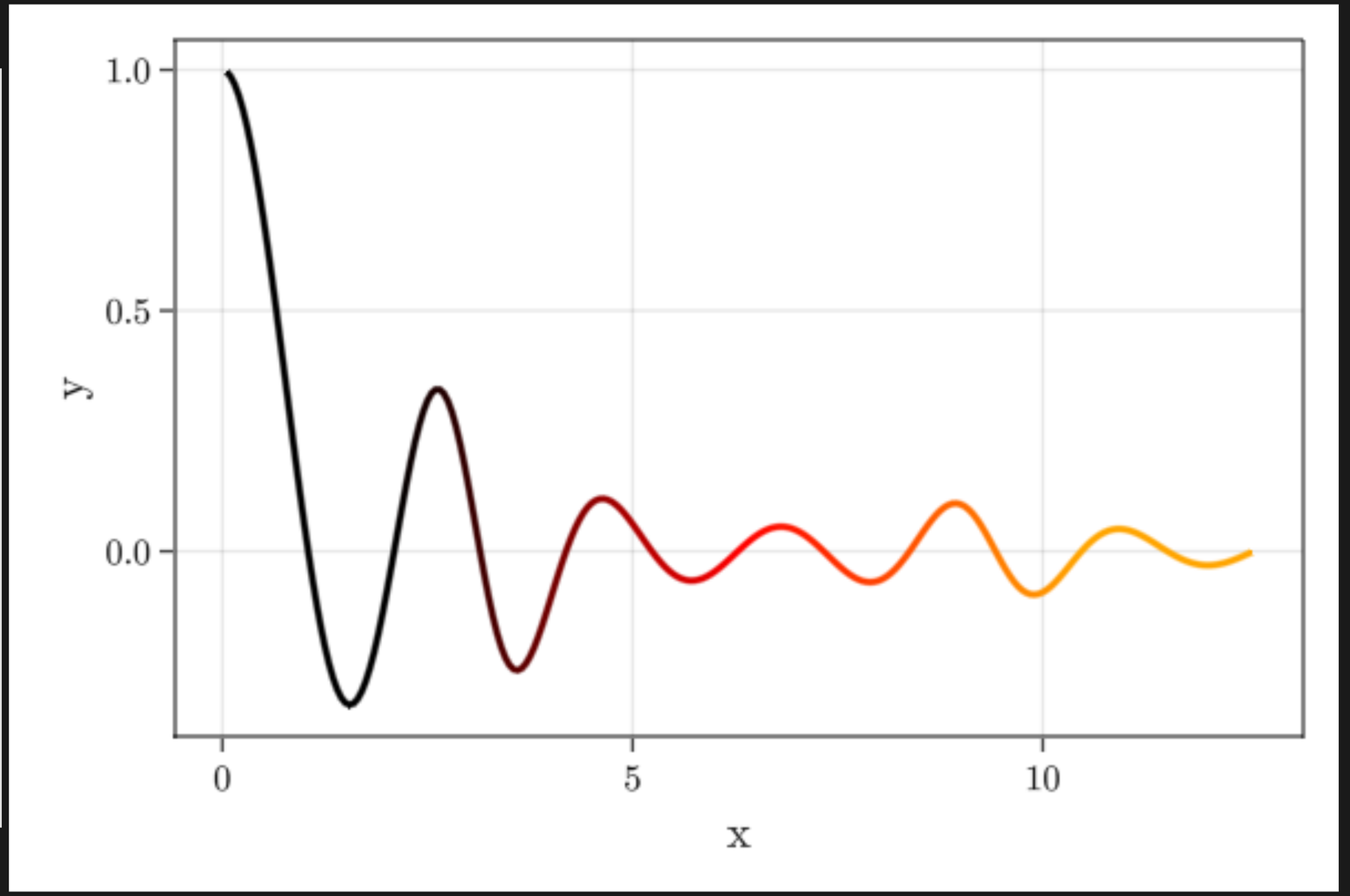
```julia
## Reversed colormap
x = 0.05:0.05:4π
y = sin.(3x) ./ (cos.(x) .+ 2) ./ x

lines(x, y; color = x, linewidth = 3, linestyle = :solid,
    colormap = Reverse(:imola))
```



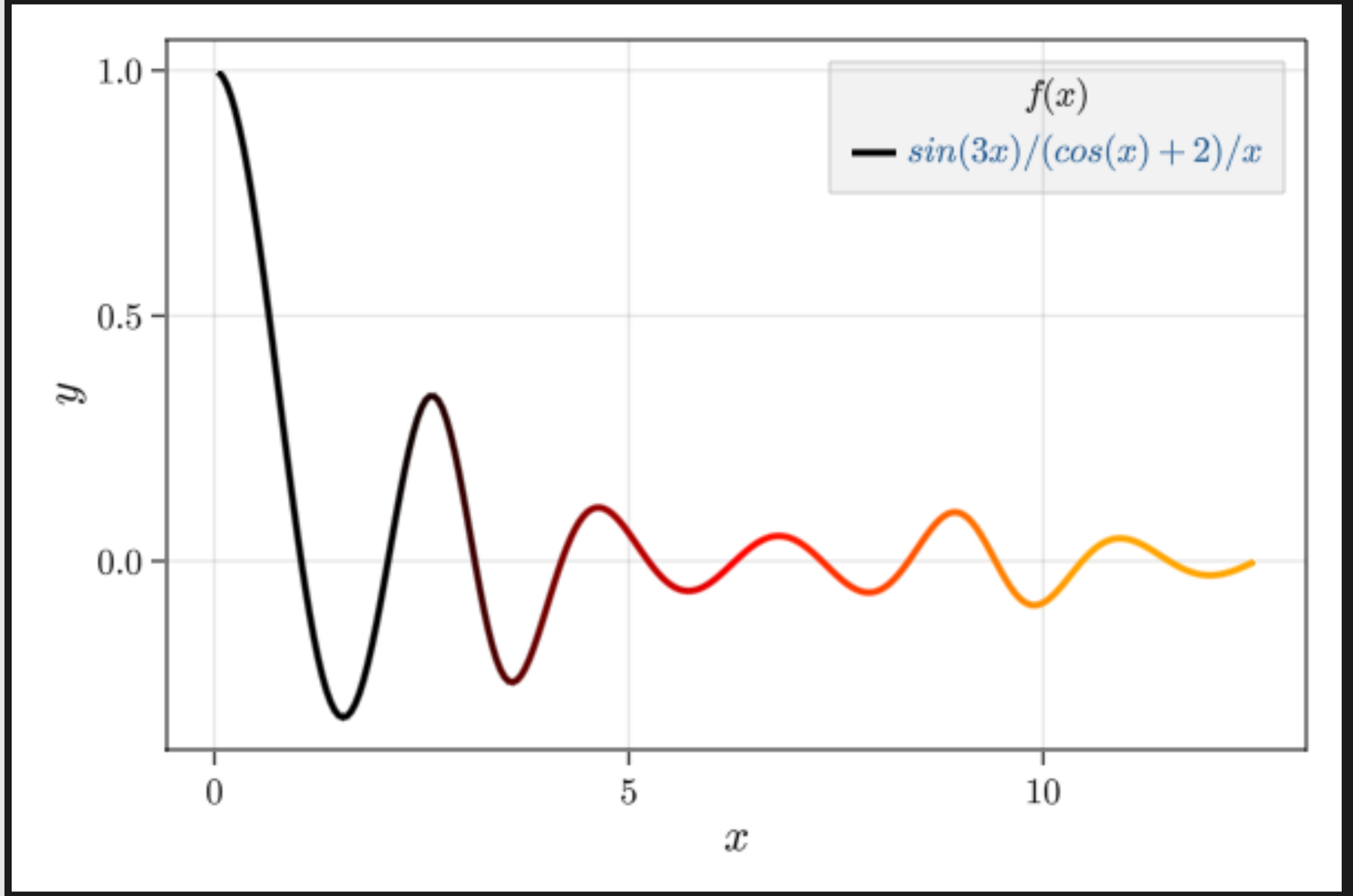
```
## Your own colors
x = 0.05:0.05:4π
y = sin.(3x) ./ (cos.(x) .+ 2) ./ x

lines(x, y; color = x, linewidth = 3, linestyle = :solid,
    colormap = [:black, :red, :orange])
```



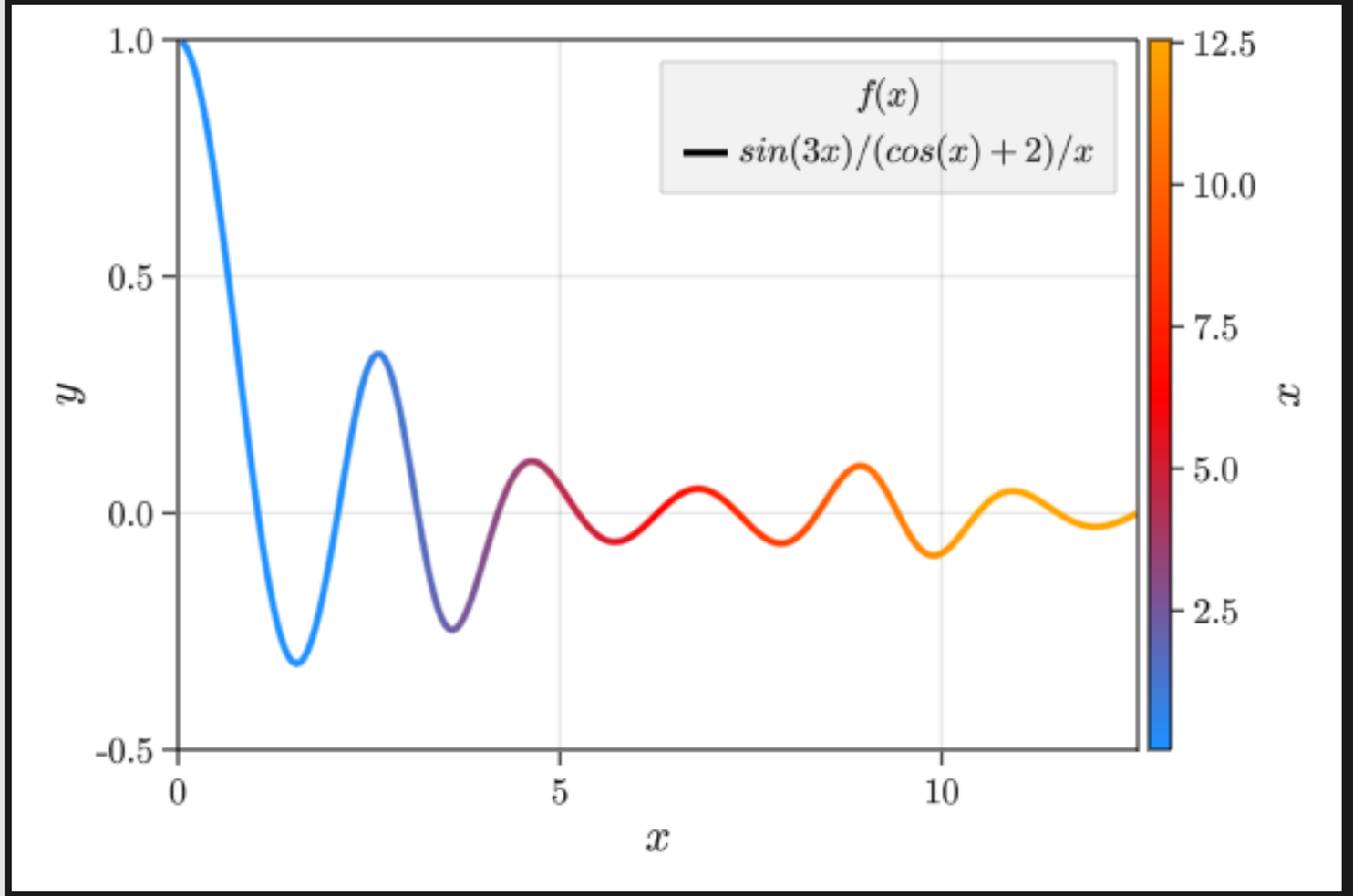
```julia
## Axis arguments
x = 0.05:0.05:4π
y = sin.(3x) ./ (cos.(x) .+ 2) ./ x

lines(x, y;
    color = x,
    linewidth = 3,
    linestyle = :solid,
    colormap = [:black, :red, :orange],
    axis = (;
        xlabel = "x",
        ylabel = "y",
        xlabelsize = 20,
        ylabelsize = 20,),
    figure = (;
        resolution = (600, 400),
        font = "CMU Serif"
        )
    )
```



```julia
## Adding a Legend
function plotlineLegend()
    lines(x, y;
        color = x,
        linewidth = 3,
        linestyle = :solid,
        colormap = [:black, :red, :orange],
        label = L"sin(3x)/(cos(x) +2)/x",
        axis = (;
            xlabel = L"x",
            ylabel = L"y",
            xlabelsize = 20,
            ylabelsize = 20,),
        figure = (;
            resolution = (600, 400),
            font = "CMU Serif"
            )
        )
    axislegend(L"f(x)";
        position = :rt,
        bgcolor = (:grey, 0.1),
        labelcolor = :dodgerblue4,
        framecolor = :snow3,
        )
    current_figure()
end
plotlineLegend()
```



```julia
## Adding a Legend and Colorbar
function plotlineCbar()
    fig, ax, obj = lines(x, y;
        color = x, linewidth = 3, linestyle = :solid,
        colormap = [:dodgerblue, :red, :orange],
        label = L"sin(3x)/(cos(x) +2)/x",
        axis = (;
            xlabel = L"x", ylabel = L"y",
            xlabelsize = 20, ylabelsize = 20,),
        figure = (;
            resolution = (600, 400),
            font = "CMU Serif"
            )
        )
    limits!(ax, 0, 4π, -0.5, 1)
    axislegend(L"f(x)"; position = :rt, bgcolor = (:grey, 0.1),
        framecolor = :snow3)
    Colorbar(fig[1,2], obj;
        label = L"x",
        width=10,
        labelsize = 20
        )
    colgap!(fig.layout, 5)
    fig
end
plotlineCbar()
```



```
with_theme(theme_ggplot2()) do
    plotlineCbar()
end
```