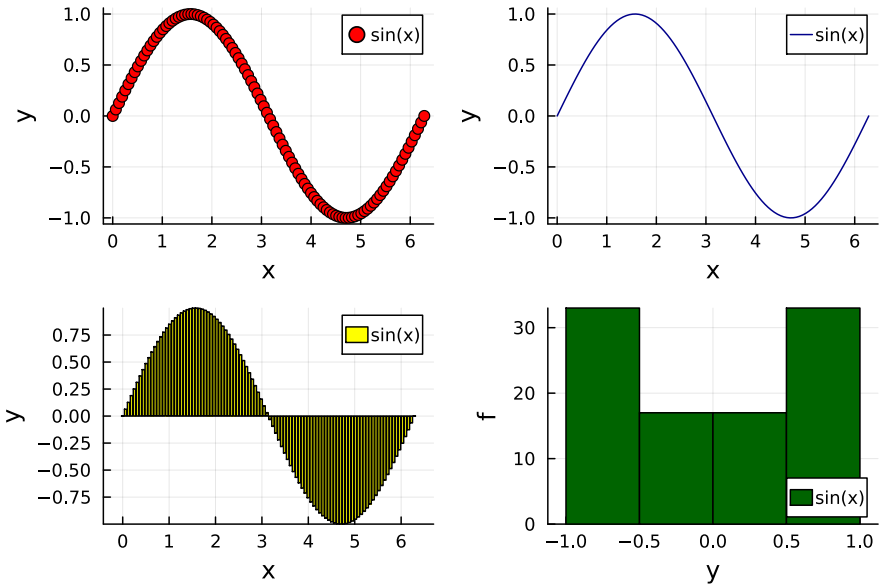
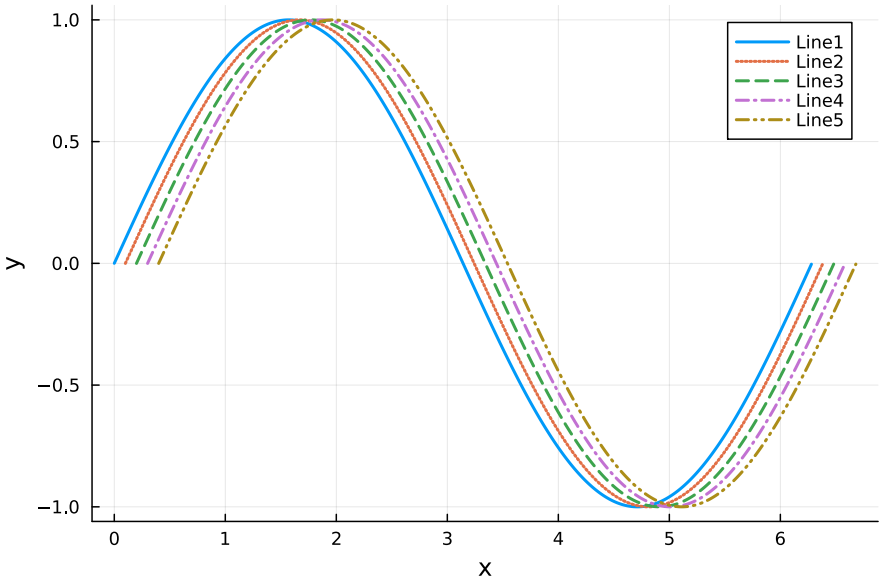
In [1]:
```
using Plots

X = range(0, stop=2π, length=100)
Y = sin.(X)

plt = scatter(X, Y, label="sin(x)", xlabel="x", ylabel="y", color="red")

plt2 = plot(X, Y, label="sin(x)", xlabel="x", ylabel="y", color="darkblue")

plt3 = bar(X, Y, label="sin(x)", xlabel="x", ylabel="y", color="yellow")

plt4 = histogram(X, Y, label="sin(x)", xlabel="y", ylabel="f", color="darkgreen")

plot(plt, plt2, plt3, plt4)
```

Out[1]:



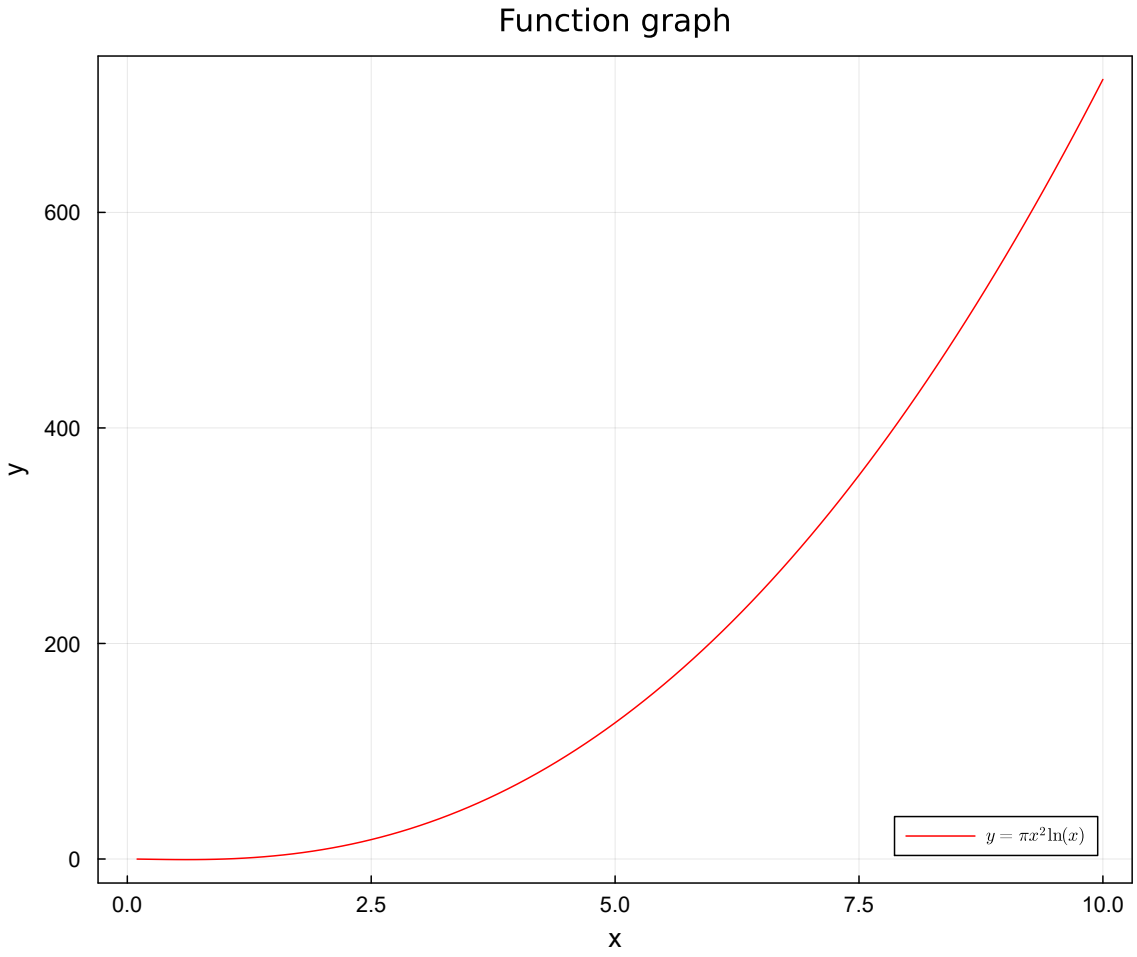
In [2]:
```
X = 0:0.01:2π
Y = sin.(X)

plt5 = plot(X, Y, label="Line1", xlabel="x", ylabel="y", linestyle=:solid, linewidth=2)
plt6 = plot!(X.+0.1, Y, label="Line2", linestyle=:dot, linewidth=2)
plt6 = plot!(X.+0.2, Y, label="Line3", linestyle=:dash, linewidth=2)
plt7 = plot!(X.+0.3, Y, label="Line4", linestyle=:dashdot, linewidth=2)
plt8 = plot!(X.+0.4, Y, label="Line5", linestyle=:dashdotdot, linewidth=2)
```

Out[2]:



In [70]:
```
y= ((X.^2).*π).*log.(X)
plot(X, y, color=:red, label="\$y = \\pi x^{2} \\ln(x)\$", xlabel="x", ylabel="y",  title="Function graph",
    framestyle=:box, framecolor="green", size=(768, 640),
    guidefont=font(12, "Arial"), tickfont=font(10, "Arial"))
```

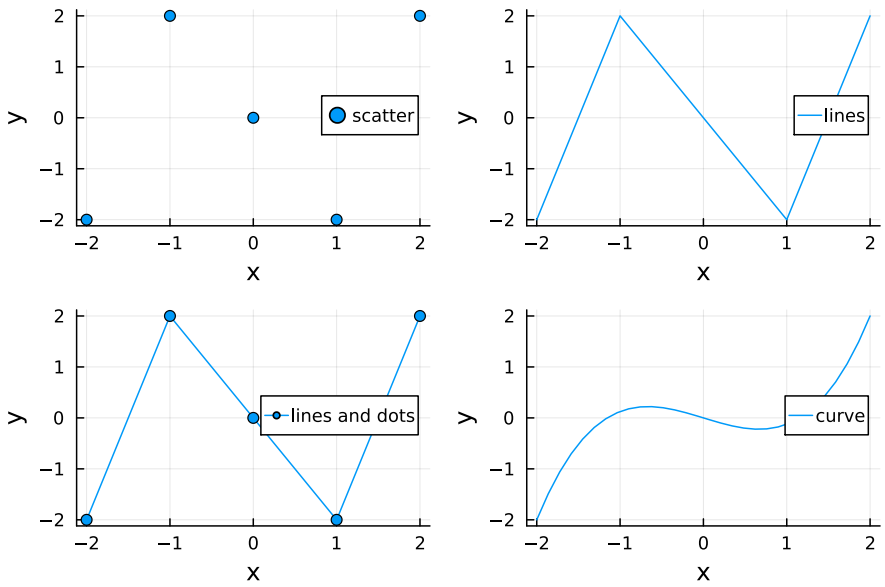
Out[70]:



In [4]:

```
X = [-2, -1, 0, 1, 2]
Y = (X.^3).-X.*3

plot1 = scatter(X, Y, label="scatter", xlabel="x", ylabel="y")
plot2 = plot(X, Y, line=:line, label="lines", xlabel="x", ylabel="y")
plot3 = plot(X, Y, line=:line, m=:circle, label="lines and dots", xlabel="x", ylabel="y")
plot4 = curves(X, Y, label="curve", xlabel="x", ylabel="y")

plot(plot1, plot2, plot3, plot4, legend=:right)
```

Out[4]:

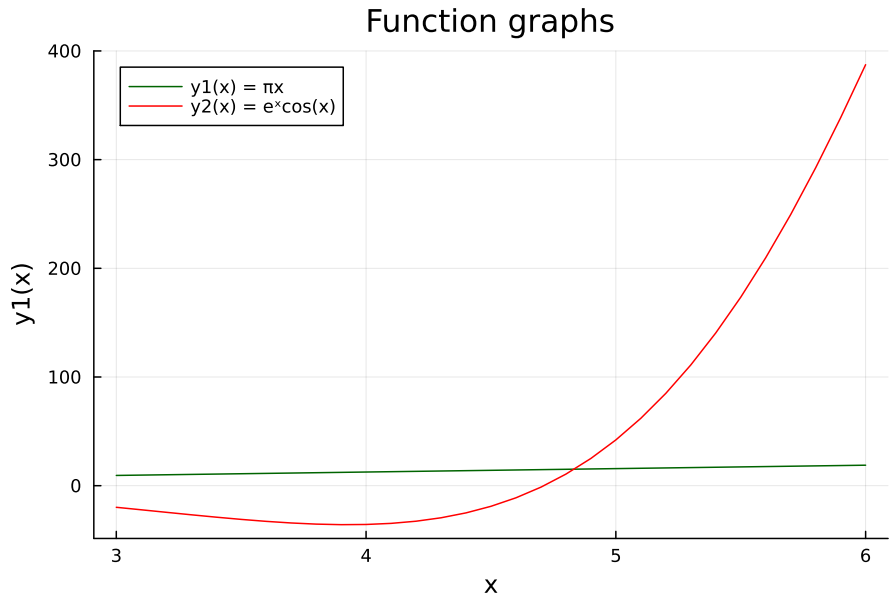


In [72]:

```
X = collect(3:0.1:6)
Y1 = X.*π
Y2= exp.(X).*cos.(X)
plot(X, Y1, label="y1(x) = πx", xlabel="x", ylabel="y1(x)", color="darkgreen",
```

```
    title="Function graphs", grid=:on, gridcolor=:black)
plot!(X, Y2, label="y2(x) = eˣcos(x)", color="red")
```

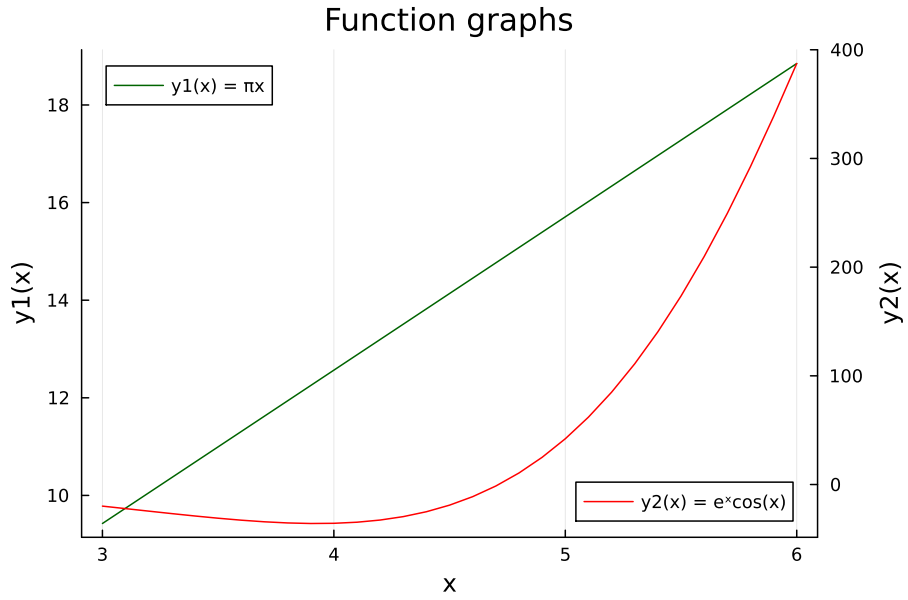
Out[72]:



In [42]:

```
X = collect(3:0.1:6)
Y1 = X.*π
Y2= exp.(X).*cos.(X)
plot(X, Y1, legend=:topleft, label="y1(x) = πx", xlabel="x", ylabel="y1(x)", title="Function graphs",
    color="darkgreen", grid=:on, gridcolor=:black)
plot!(twinx(), X, Y2, secondary=true, legend=:bottomright,
    label="y2(x) = eˣcos(x)", ylabel="y2(x)", color="red")
```

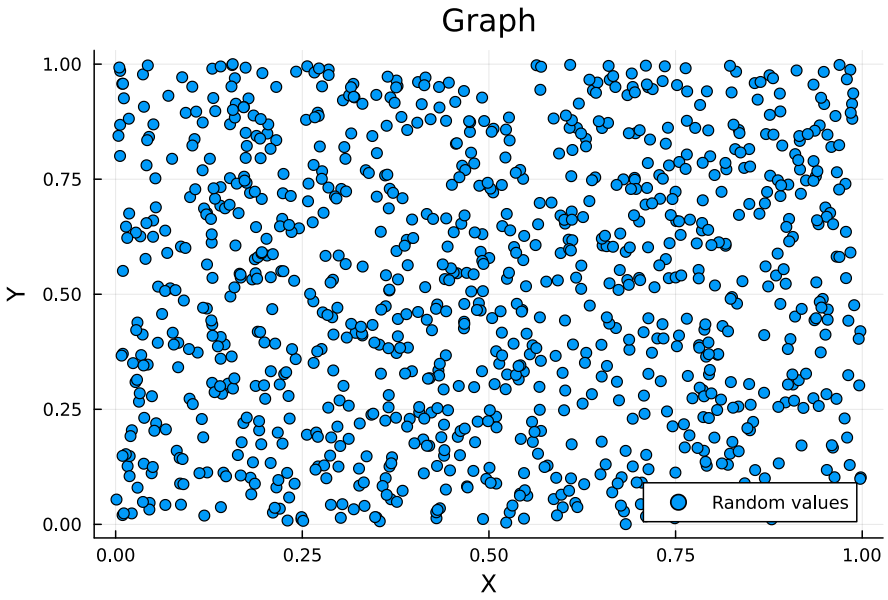
Out[42]:



In [73]:

```
using Random

x_data = rand(1000)
y_data = rand(1000)

scatter(x_data, y_data, label="Random values", xlabel="X", ylabel="Y", title="Graph")
```

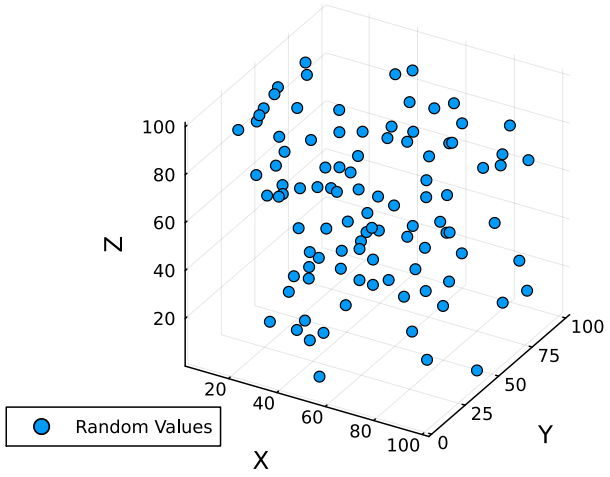
Out[73]:



In [47]:

```
x_data = rand(1:100,100)
y_data = rand(1:100,100)
z_data = rand(1:100,100)

plot(x_data, y_data, z_data, seriestype=:scatter, label="Random Values",
    xlabel="X", ylabel="Y", zlabel="Z", title="3D random values graph",)
```

Out[47]:

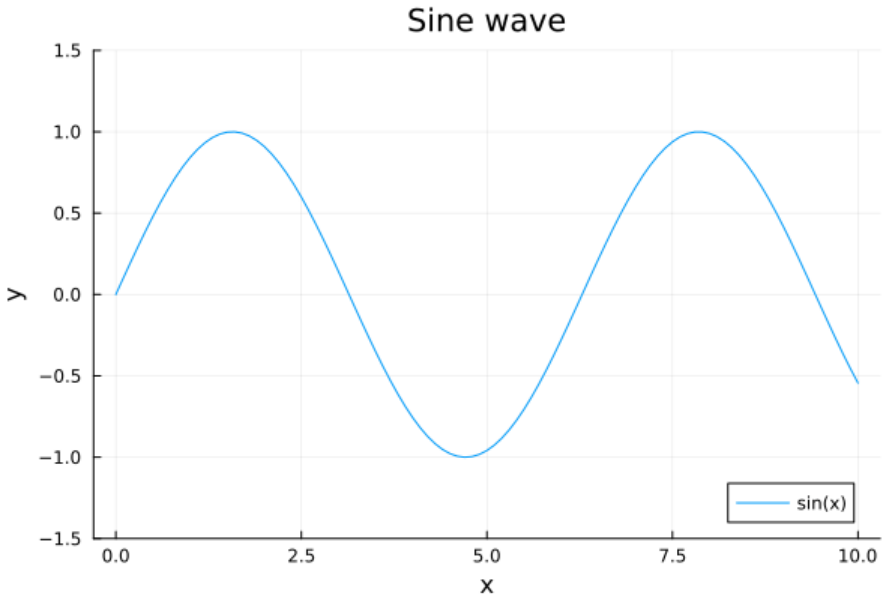


In [59]:

```
X = collect(0:0.1:10)
period = collect(0:0.1:2π)

anim = @animate for t in period
    y_values = sin.(X.+t)
    plot(x_values, y_values, label="sin(x)", xlabel="x", ylabel="y", title="Sine wave", ylims=(-1.5, 1.5))
end

gif(anim, "sine.gif", fps = 10)
```

```
[ Info: Saved animation to C:\Users\marin\Documents\UNI\DA\Lab5\sine.gif
```

Out[59]:

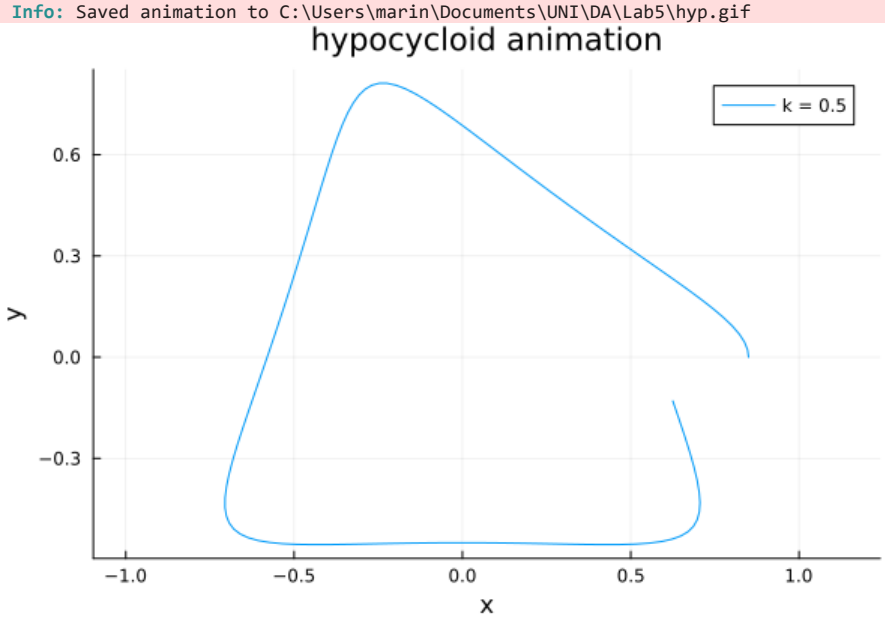


In [66]:

```
function hypocycloid(a, b, k)
    tetta  = range(0, stop=2π, length=100)
    x = (a - b) * cos.(tetta) .+ b * k * cos.((a - b) / b * tetta)
    y = (a - b) * sin.(tetta) .- b * k * sin.((a - b) / b * tetta)
    return x, y
end

k_values = collect(0.5:+0.5:2.5)

anim = @animate for k in k_values
    x, y = hypocycloid(1, 0.3, k)
    plot(x, y, aspect_ratio=:equal, label="k = $k", xlabel="x", ylabel="y", title="hypocycloid animation")
end

gif(anim, "hyp.gif", fps=5)
```

```
[ Info: Saved animation to C:\Users\marin\Documents\UNI\DA\Lab5\hyp.gif
```

Out[66]:



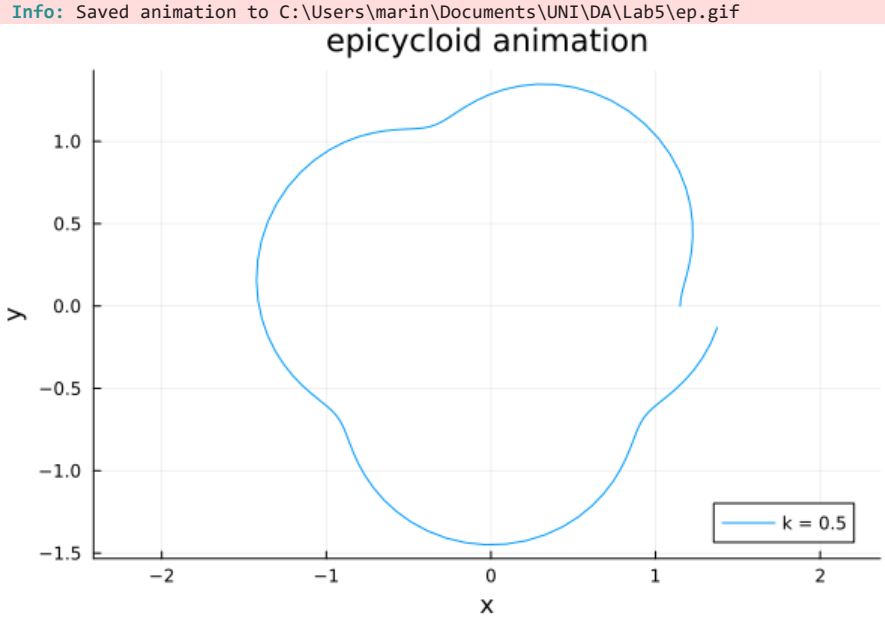
In [67]:

```
function epicycloid(a, b, k)
    tetta = range(0, stop=2π, length=100)
    x = (a + b) * cos.(tetta) .- b * k * cos.((a + b) / b * tetta)
    y = (a + b) * sin.(tetta) .- b * k * sin.((a + b) / b * tetta)
    return x, y
end

k_values = collect(0.5:+0.5:2.5)

anim = @animate for k in k_values
    x, y = epicycloid(1, 0.3, k)
    plot(x, y, aspect_ratio=:equal, label="k = $k", xlabel="x",
        ylabel="y", title="epicycloid animation")
```

```
end

gif(anim, "ep.gif", fps=5)
```

```
[ Info: Saved animation to C:\Users\marin\Documents\UNI\DA\Lab5\ep.gif
```

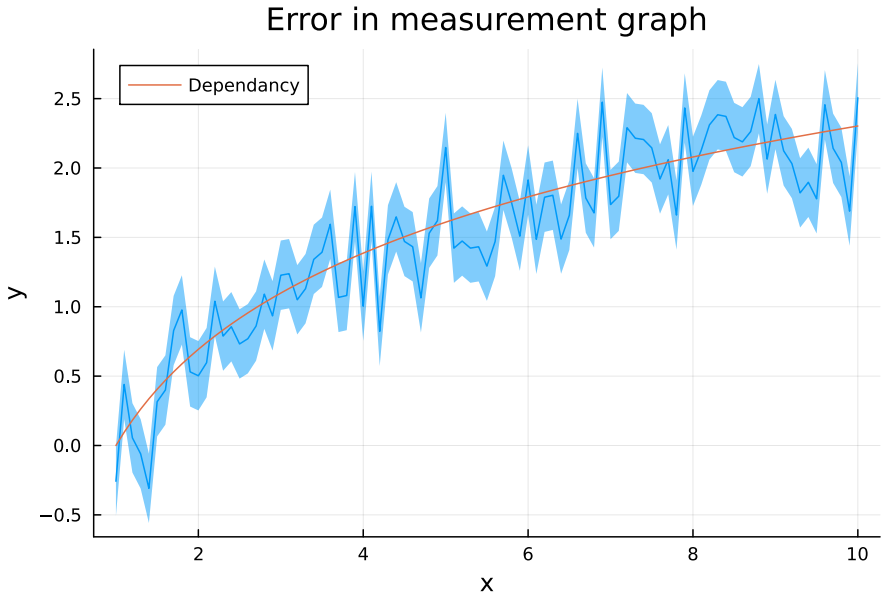
Out[67]:



In [84]:

```
x = 1:0.1:10
y_true =log.(x)

error_std = 0.25
y_pred = y_true.-randn(size(y_true))*error_std

plot(x, y_pred, ribbon=error_std, label="", xlabel="x", ylabel="y",
    title="Error in measurement graph")
plot!(x, y_true, label="Dependancy")
```

Out[84]:



In [ ]: