Графика в Julia

ТАУБЕР КИРИЛЛ ОЛЕГОВИЧ

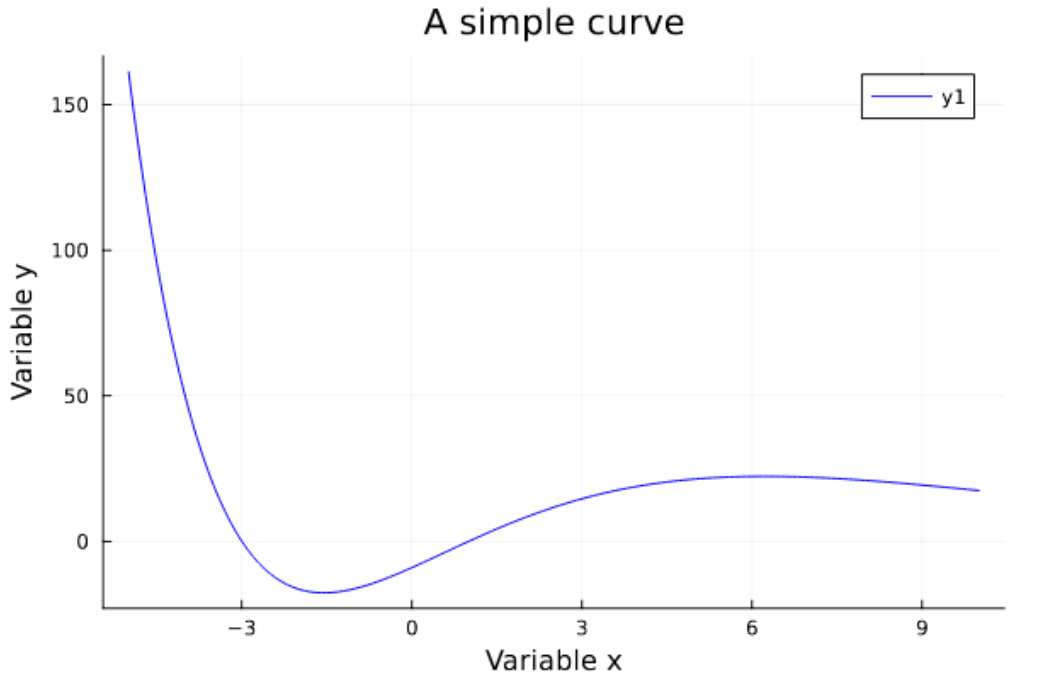
НПИБД-02-19

РУДН,2022

Ввод [95]:

```
using Plots
f(x) = (3x.^2 + 6x .- 9).*exp.(-0.3x)

# генерирование массива значений x в диапазоне от -5 до 10 с шагом 0,1
# (шаг задан через указание длины массива):
x = collect(range(-5,10,length=151))

# генерирование массива значений y:
y = f(x)

# указывается, что для построения графика используется gr():
gr()

# задание опций при построении графика
# (название кривой, подписи по осям, цвет графика):
plot(x,y,title="A simple curve",xlabel="Variable x",ylabel="Variable y",color
```

Out[95]:



Ввод [96]:

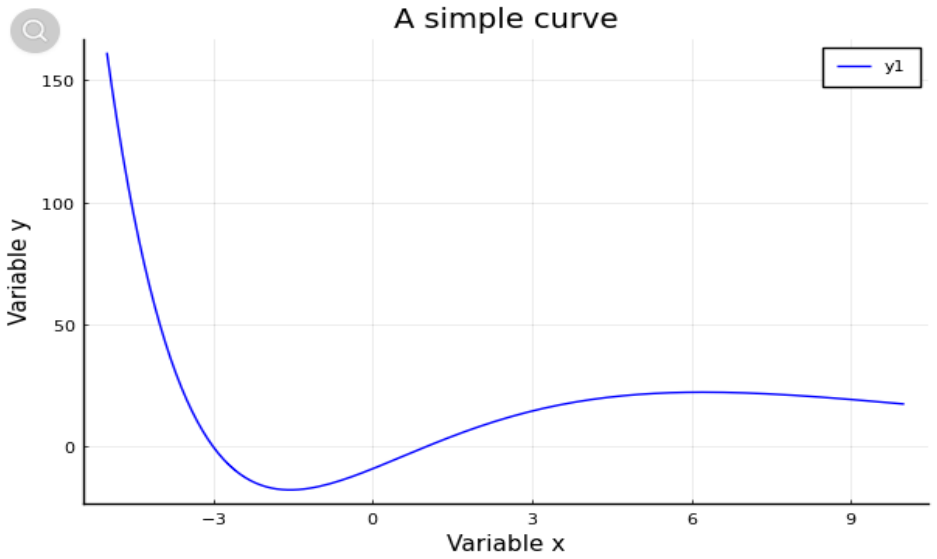
```
pyplot()
```

Out[96]: Plots.PyPlotBackend()

Ввод [97]:

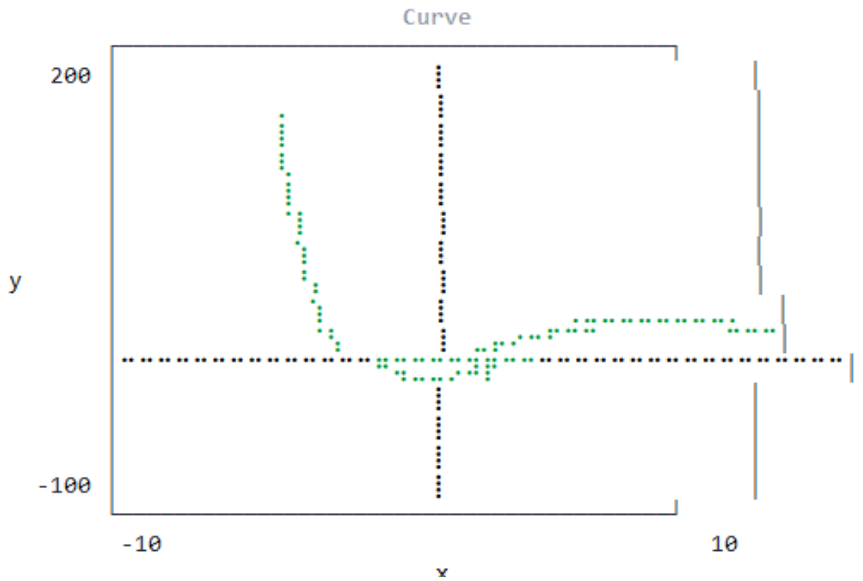
```
plot(x,y,title="A simple curve",xlabel="Variable x",ylabel="Variable y",color
```

Out[97]:



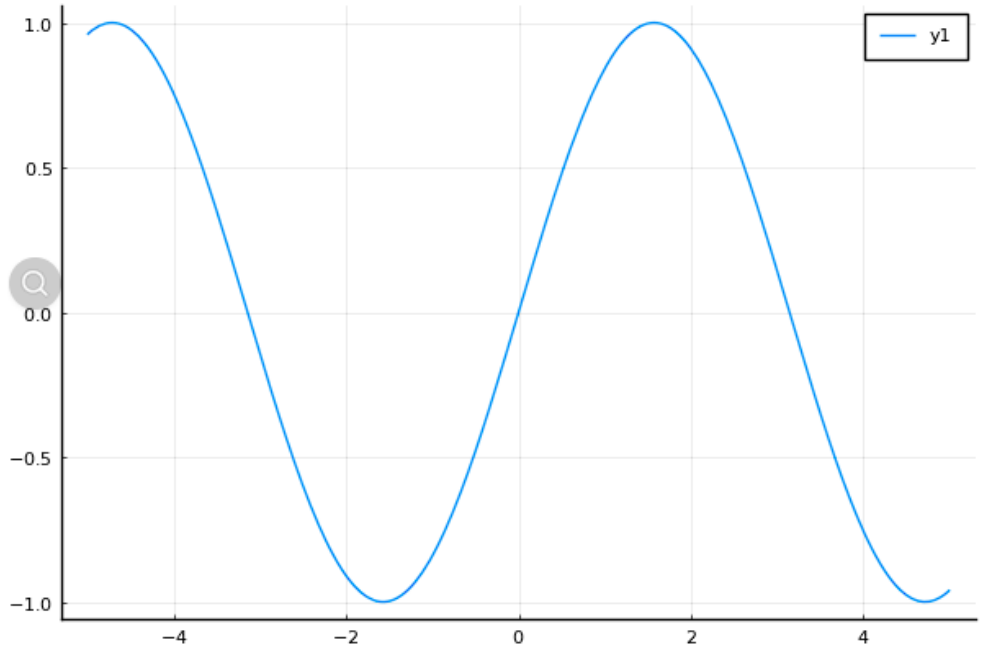
Ввод [157]:
```
using UnicodePlots
lineplot(x,y,title="Curve",xlabel="x",ylabel="y")
```

Out[157]:



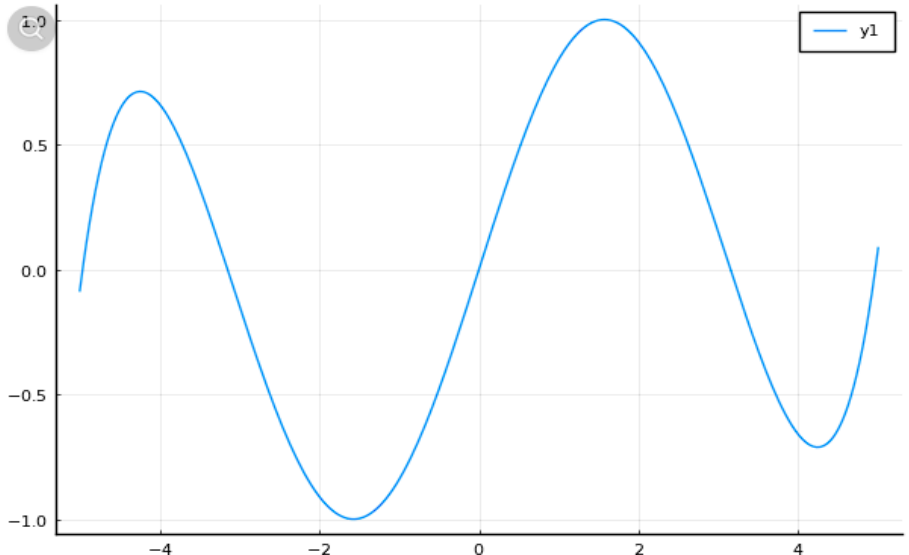
Ввод [158]:
```
pyplot()
sin_theor(x) = sin(x)
plot(sin_theor)
```

Out[158]:



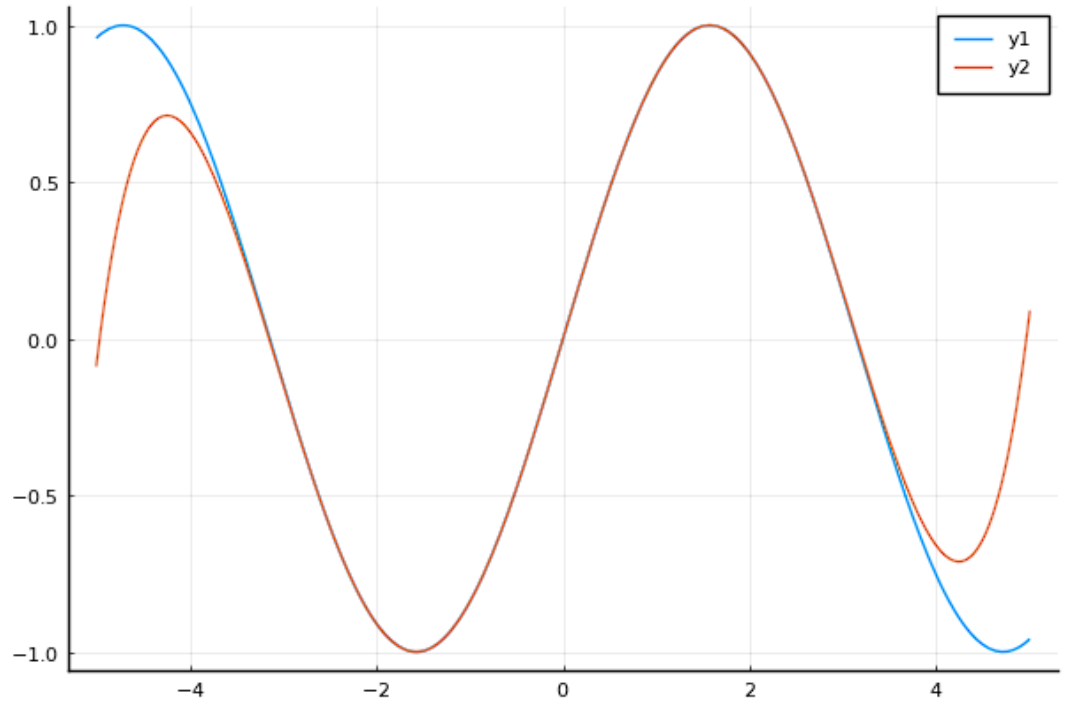
Ввод [159]:
```
sin_taylor(x) = [(-1)^i*x^(2*i+1)/factorial(2*i+1) for i in 0:4] |> sum
plot(sin_taylor)
```

Out[159]:



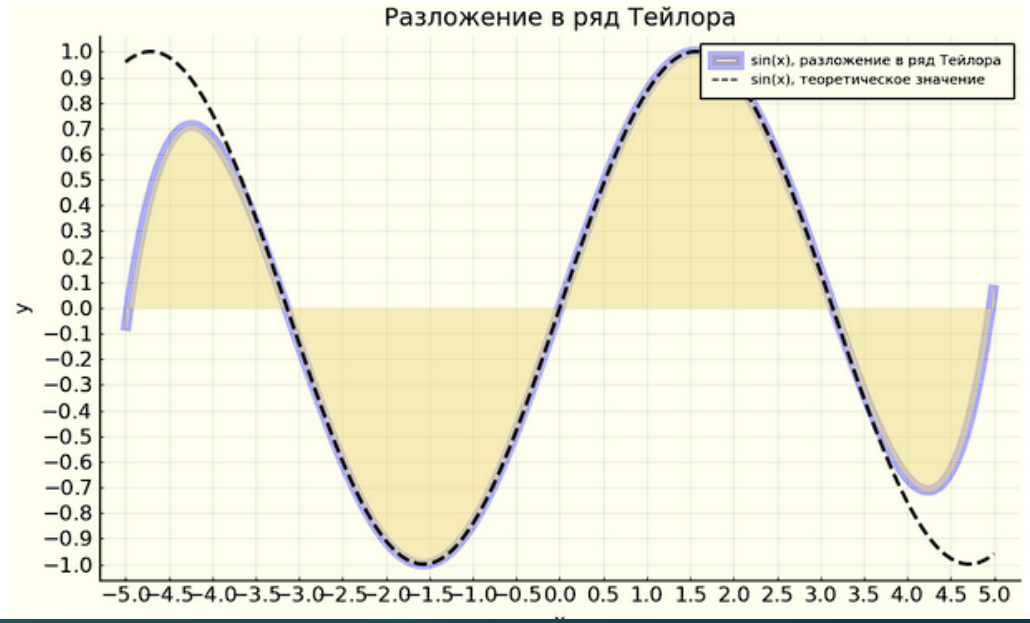
Ввод [160]:
```
plot(sin_theor)
plot!(sin_taylor)
```

Out[160]:




```
# параметры отображения значений по осям
xticks = (-5:0.5:5),
yticks = (-1:0.1:1),
xtickfont = font(12, "Times New Roman"),
ytickfont = font(12, "Times New Roman"),
# подписи по осям:
ylabel = "y",
xlabel = "x",
# название графика:
title = "Разложение в ряд Тейлора",
# поворот значений, заданный по оси x:
xrotation = rad2deg(pi/4),
# заливка области графика цветом:
fillrange = 0,
fillalpha = 0.5,
fillcolor = :lightgoldenrod,
# задание цвета фона:
background_color = :ivory
)
plot!(
# функция sin_theor:
sin_theor,
# подпись в легенде, цвет и тип линии:
label = "sin(x), теоретическое значение",
line=(:black, 1.0, 2, :dash))
```

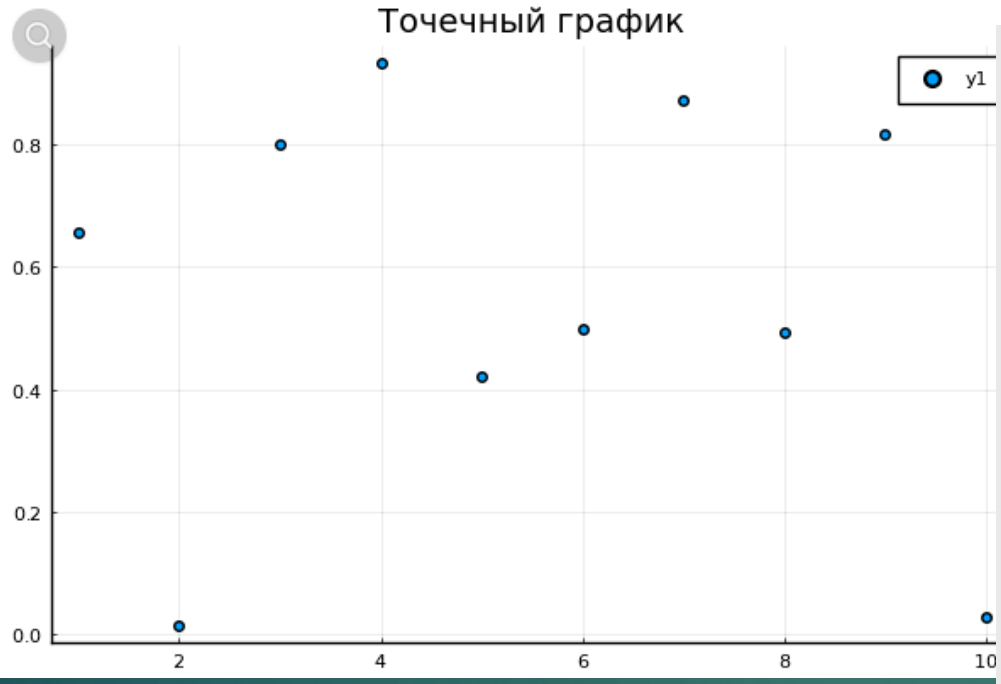
Out[161]:



Ввод [208]:

```
x = range(1,10,length=10)
y = rand(10)
plot(x, y, seriestype = :scatter, title = "Точечный график")
```

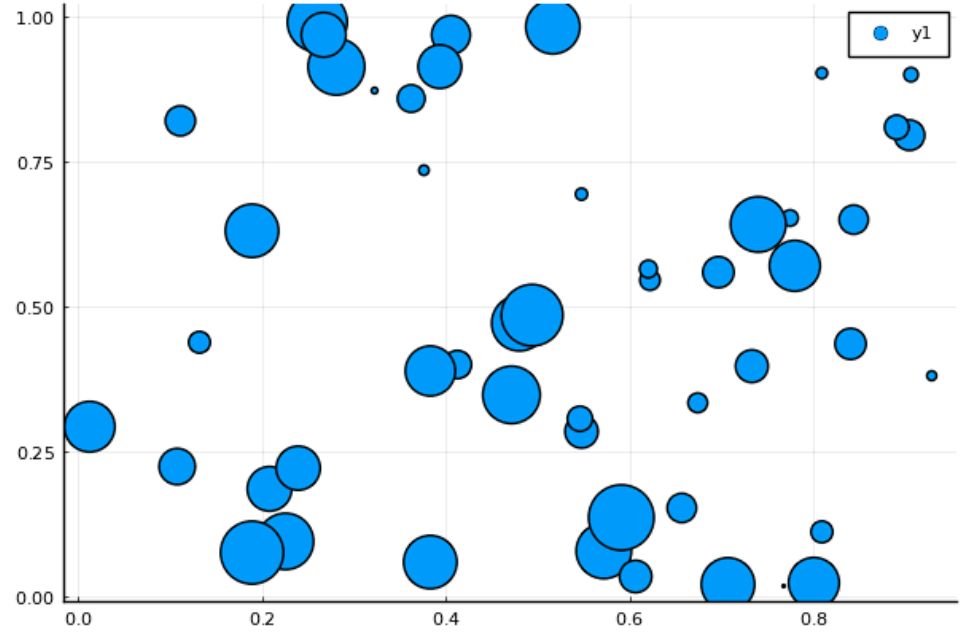
Out[208]:



Ввод [163]:

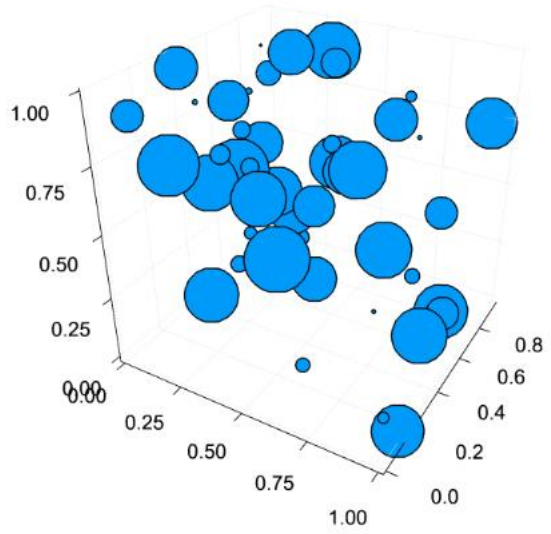
```
pyplot()
n = 50
x = rand(n)
y = rand(n)
ms = rand(50) * 30
scatter(x, y, markersize=ms)
```

Out[163]:



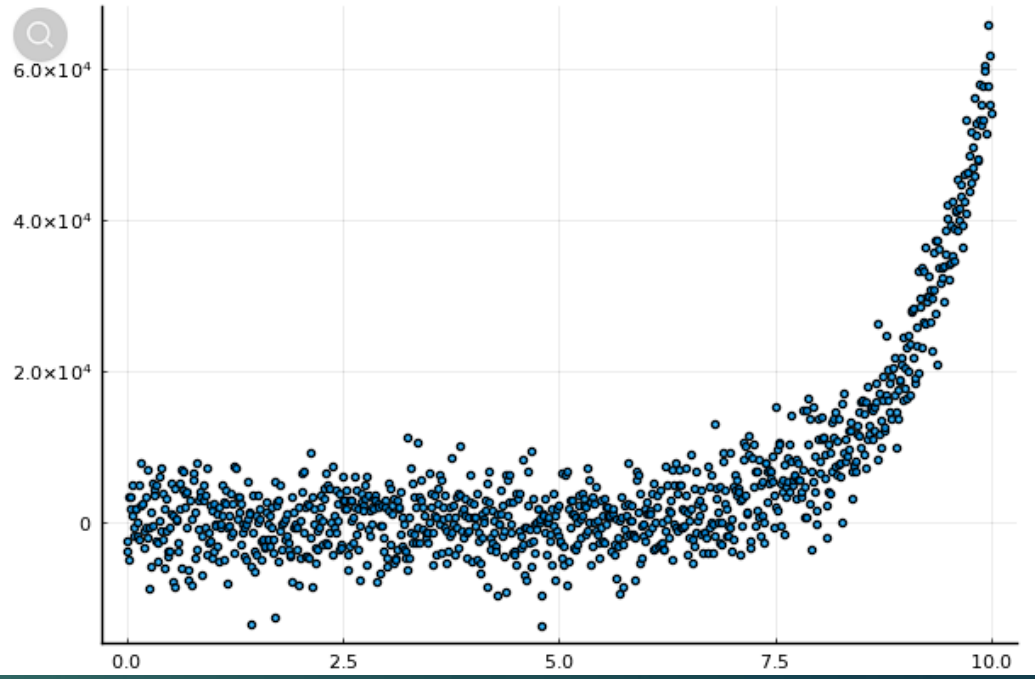
Ввод [164]:
```
plotly()
n = 50
x = rand(n)
y = rand(n)
z = rand(n)
ms = rand(n)*10
scatter(x, y, z, markersize=ms)
```

Out[164]:



Ввод [165]:
```
pyplot()
x = collect(0:0.01:9.99)
y = exp.(ones(1000)+x) + 4000*randn(1000)
scatter(x,y,markersize=3,alpha=.8,legend=false)
```

Out[165]:

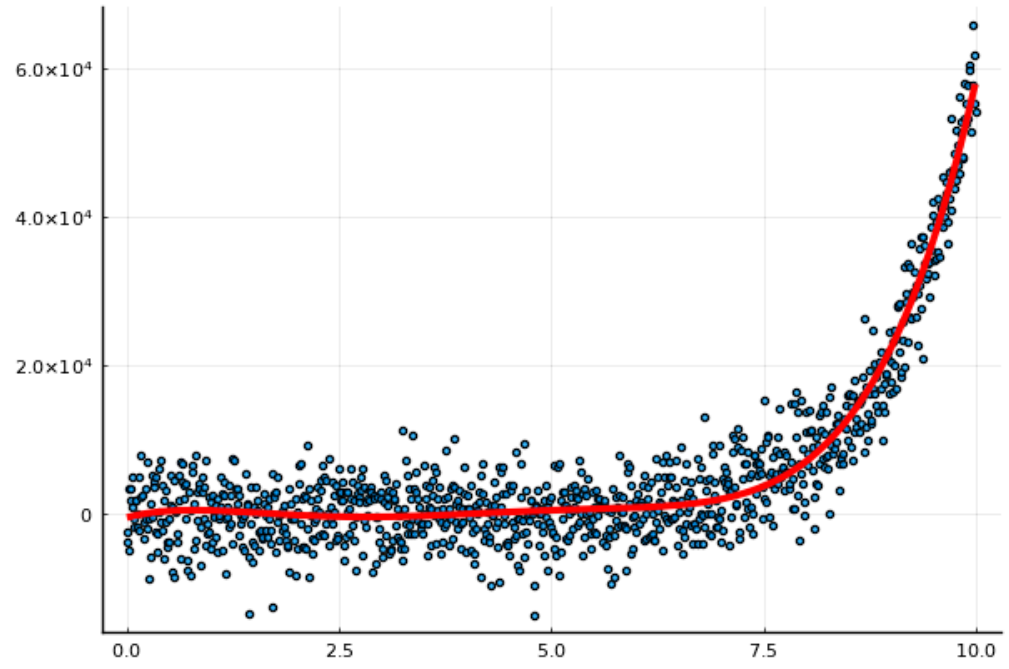


Ввод [166]:
```
A = [ones(1000) x x.^2 x.^3 x.^4 x.^5]
c = A\y
```

Out[166]:
```
6-element Vector{Float64}:
  -490.1042530295315
  3019.6578573556308
 -2967.1338981294025
  1043.6605015429384
  -153.45932170288876
     8.164354107272525
```

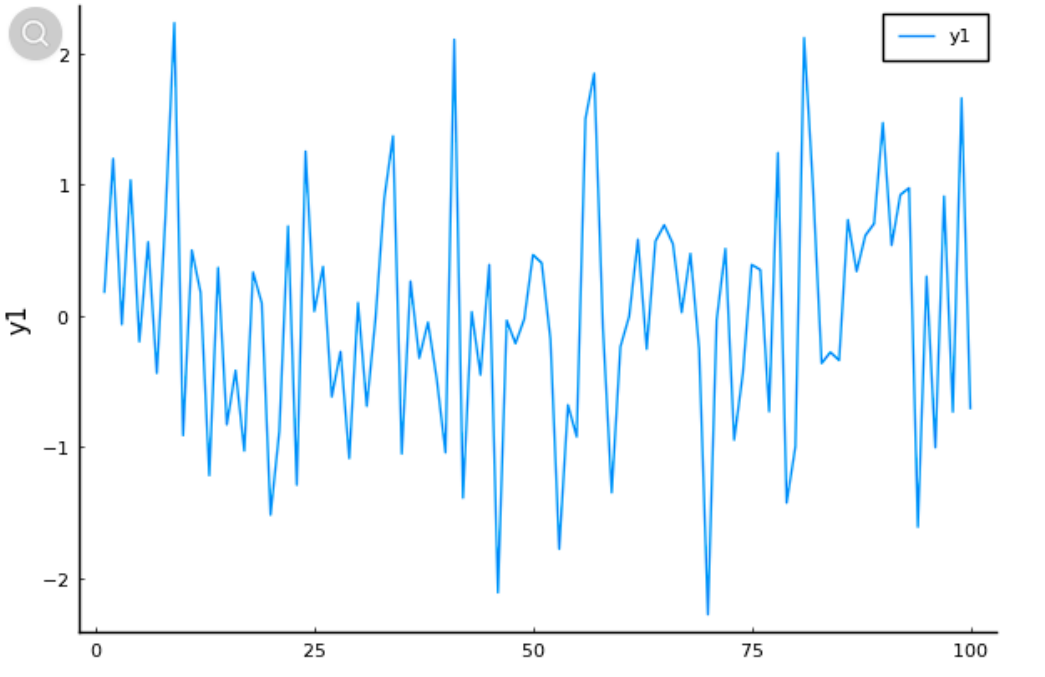
Ввод [167]:
```
F = c[1]*ones(1000) + c[2]*x + c[3]*x.^2 + c[4]*x.^3 + c[5]*x.^4 + c[6]*x.^5
plot!(x,F,linewidth=3, color=:red)
```

Out[167]:

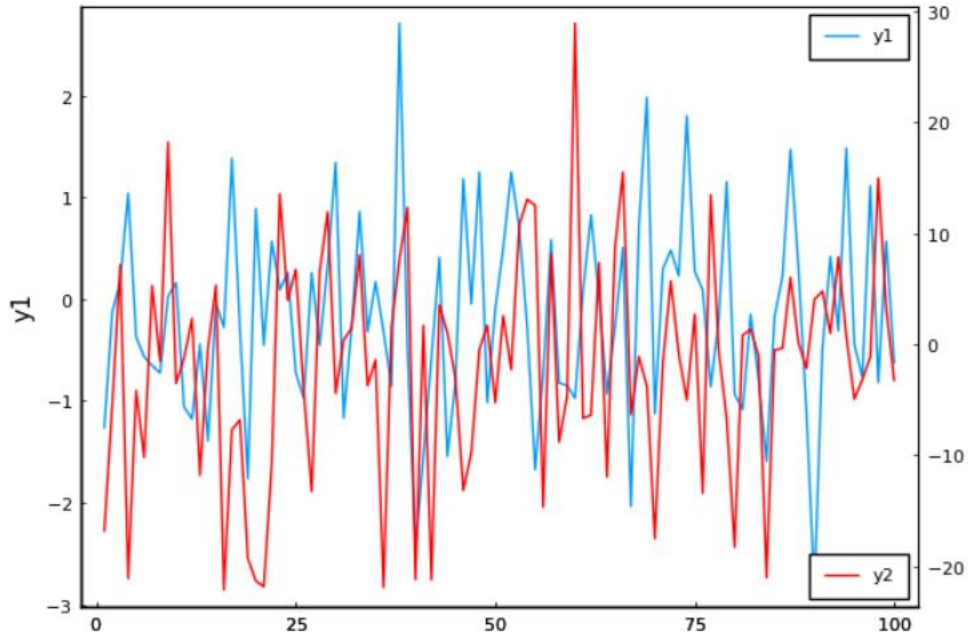


Ввод [168]:
```
plot(randn(100),ylabel="y1",leg=:topright,grid = :off,)
```

Out[168]:

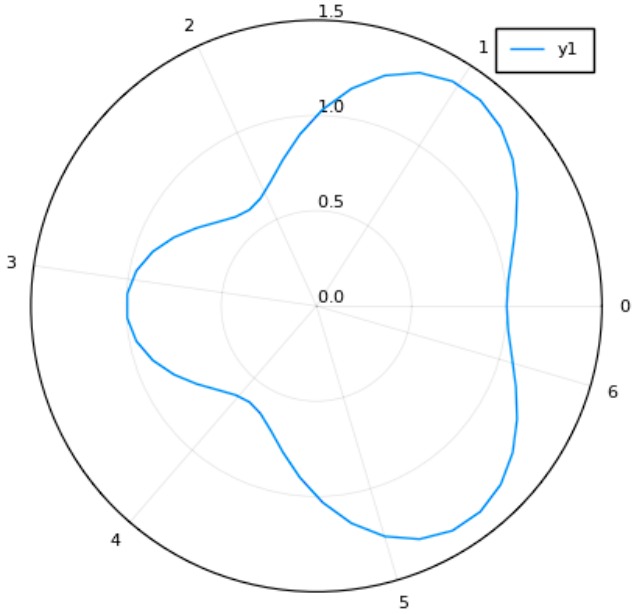


```
# пример добавления на график второй случайной траектории
# (задано обозначение траектории и её цвет, легенда снизу справа, без сетки)
# задана рамка графика
plot!(twinx(), randn(100)*10,
c=:red,
ylabel="y2",
leg=:bottomright,
grid = :off,
box = :on,
# size=(600, 400)
)
```



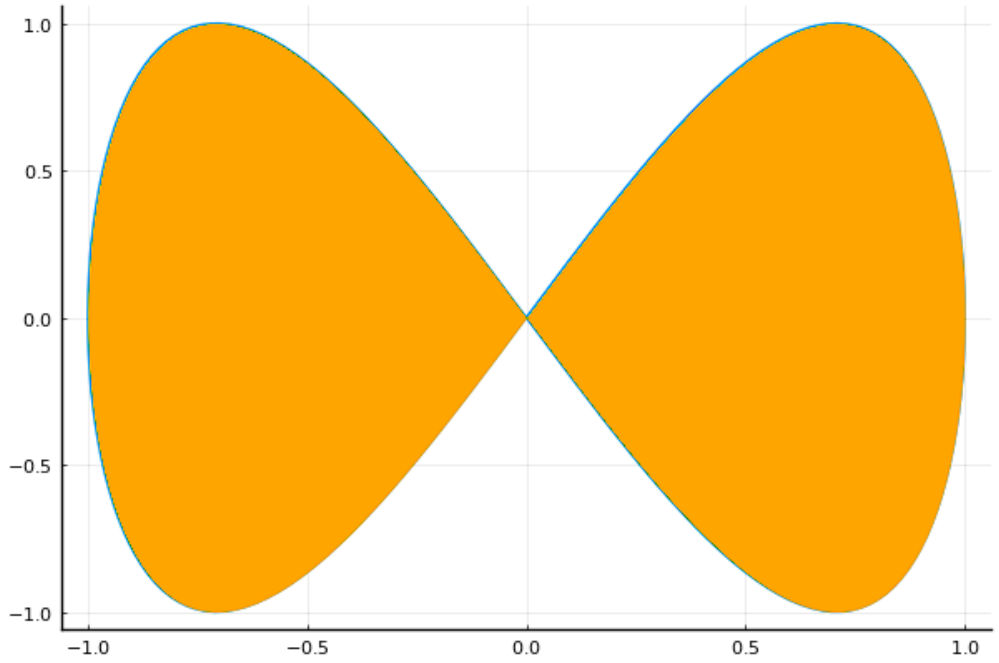
Ввод [170]:
```
r(θ) = 1 + cos(θ) * sin(θ)^2
θ = range(0, stop=2π, length=50)
plot(θ, r.(θ),
proj=:polar,
lims=(0,1.5)
)
```

Out[170]:



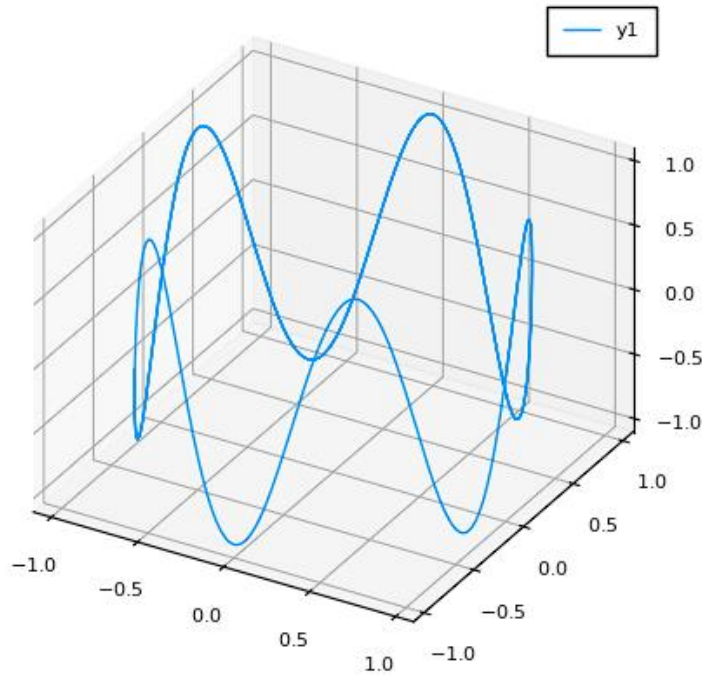
Ввод [171]:
```
X(t) = sin(t)
Y(t) = sin(2t)
plot(X, Y, 0, 2π, leg=false, fill=(0,:orange))
```

Out[171]:



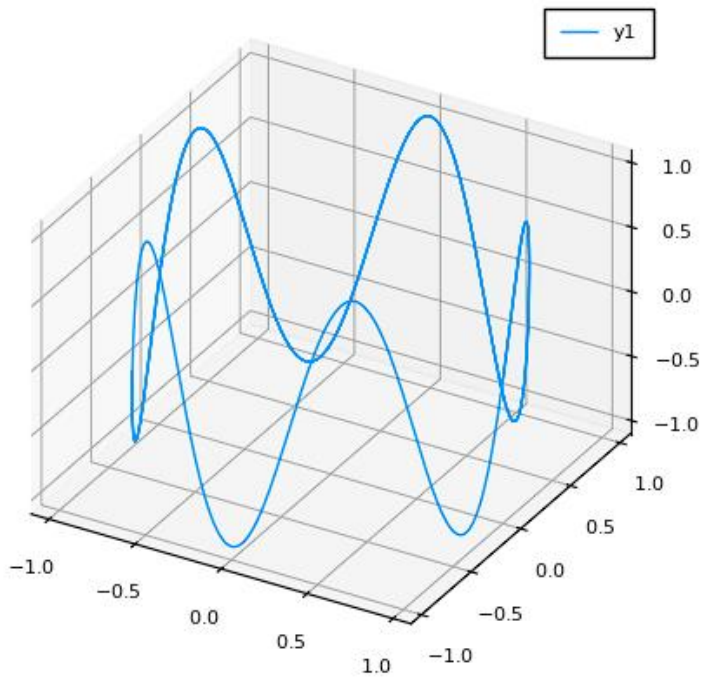
Ввод [172]:
```
t = range(0, stop=10, length=1000)
x = cos.(t)
y = sin.(t)
z = sin.(5t)
plot(x, y, z)
```

Out[172]:

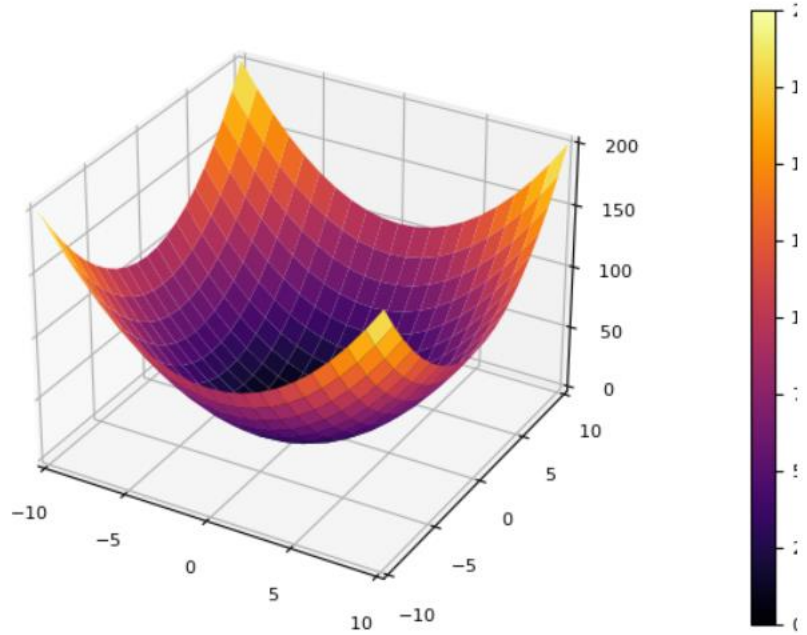


Ввод [172]:
```
t = range(0, stop=10, length=1000)
x = cos.(t)
y = sin.(t)
z = sin.(5t)
plot(x, y, z)
```

Out[172]:

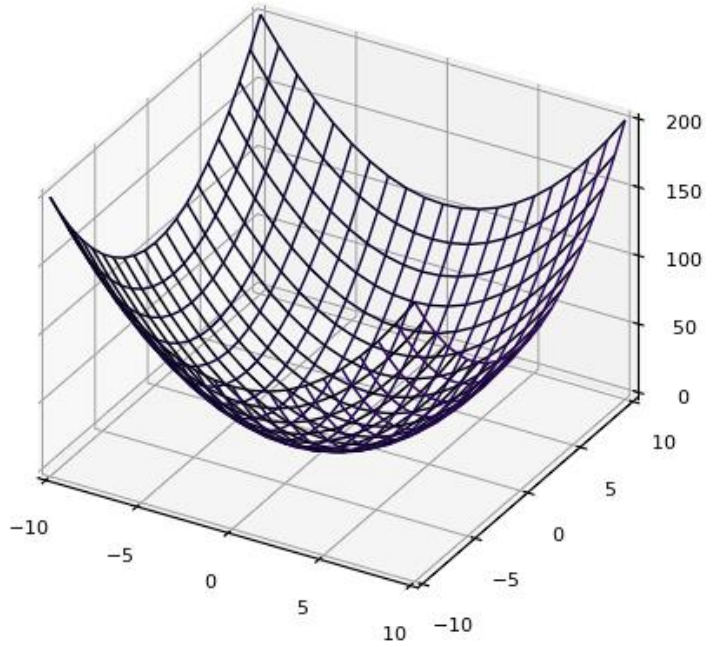



```
# построение графика поверхности:
f(x,y) = x^2 + y^2
x = -10:10
y = x
surface(x, y, f)
```



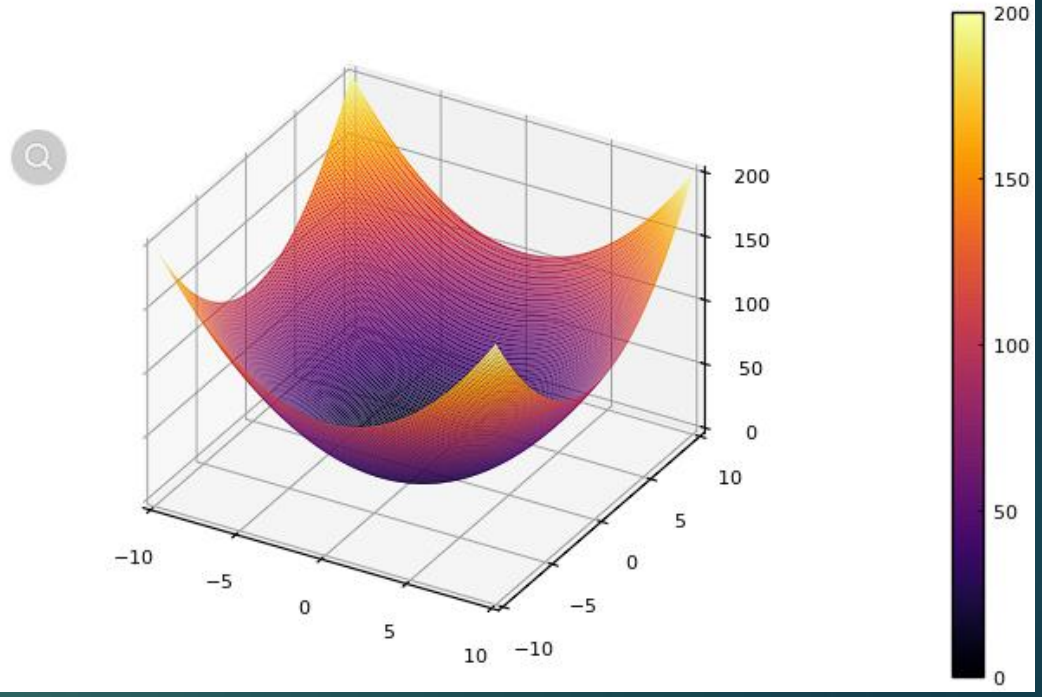
Ввод [174]:
```
f(x,y) = x^2 + y^2
x = -10:10
y = x
plot(x, y, f,
linetype=:wireframe
)
```

Out[174]:



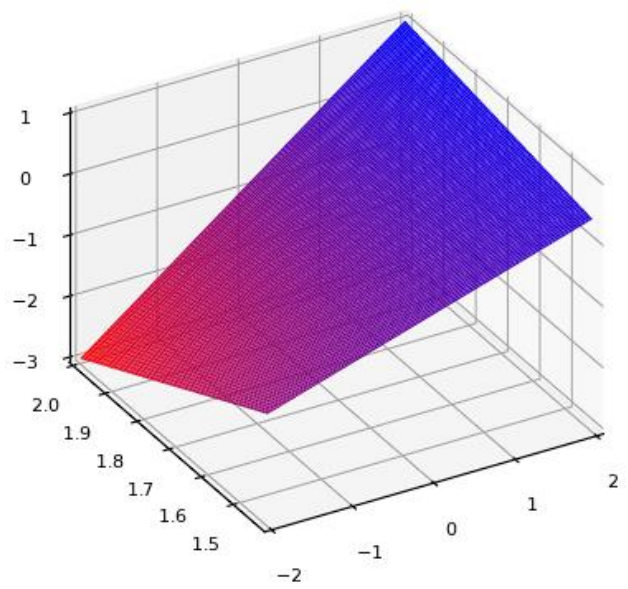
Ввод [175]:
```
f(x,y) = x^2 + y^2
x = -10:0.1:10
y = x
plot(x, y, f,
linetype = :surface
)
```

Out[175]:



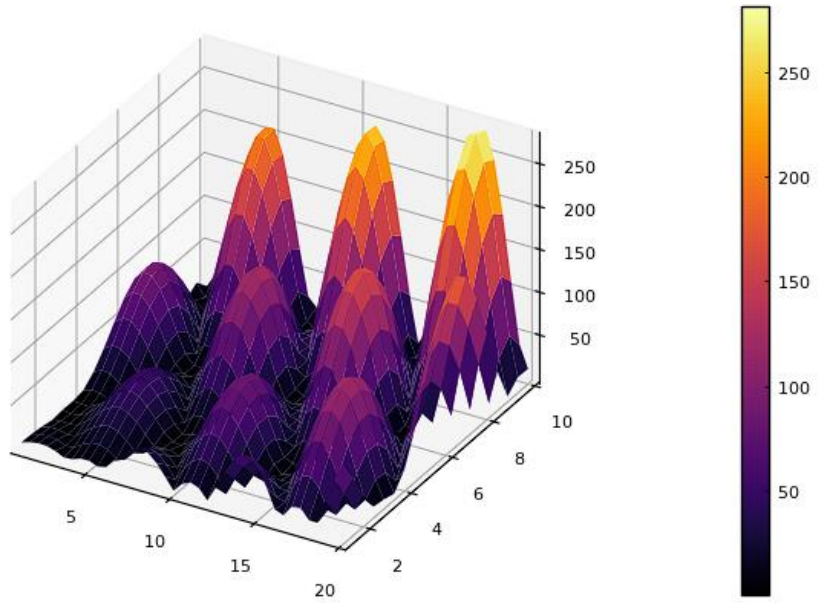
Ввод [176]:
```
x=range(-2,stop=2,length=100)
y=range(sqrt(2),stop=2,length=100)
f(x,y) = x*y-x-y+1
plot(x,y,f,
linetype = :surface,
c=cgrad([:red,:blue]),
camera=(-30,30),
)
```

Out[176]:

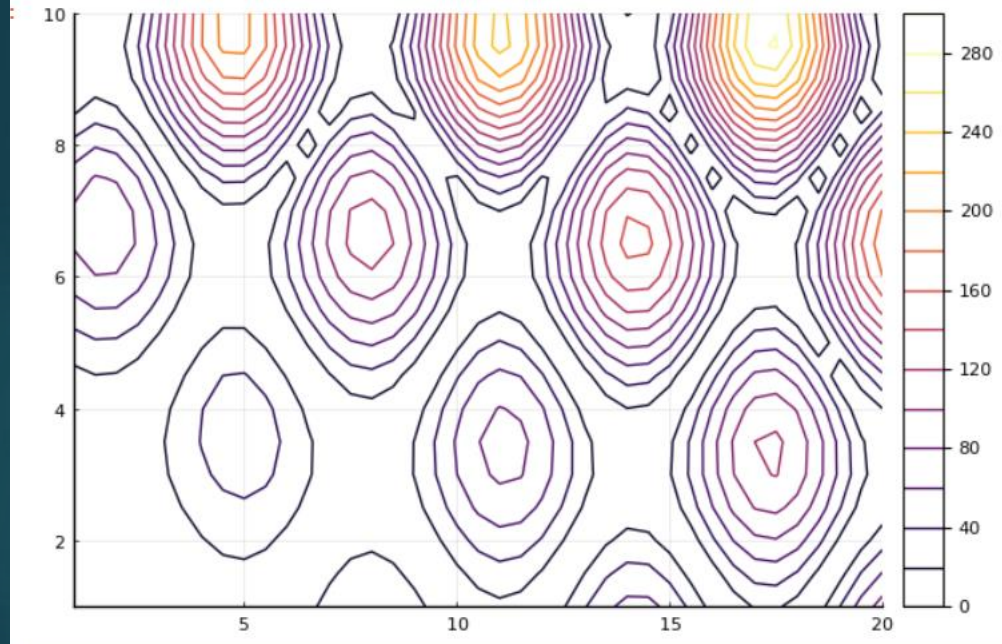


Ввод [177]:
```
x = 1:0.5:20
y = 1:0.5:10
g(x, y) = (3x + y ^ 2) * abs(sin(x) + cos(y))
plot(x,y,g,
linetype = :surface,
)
```

Out[177]:

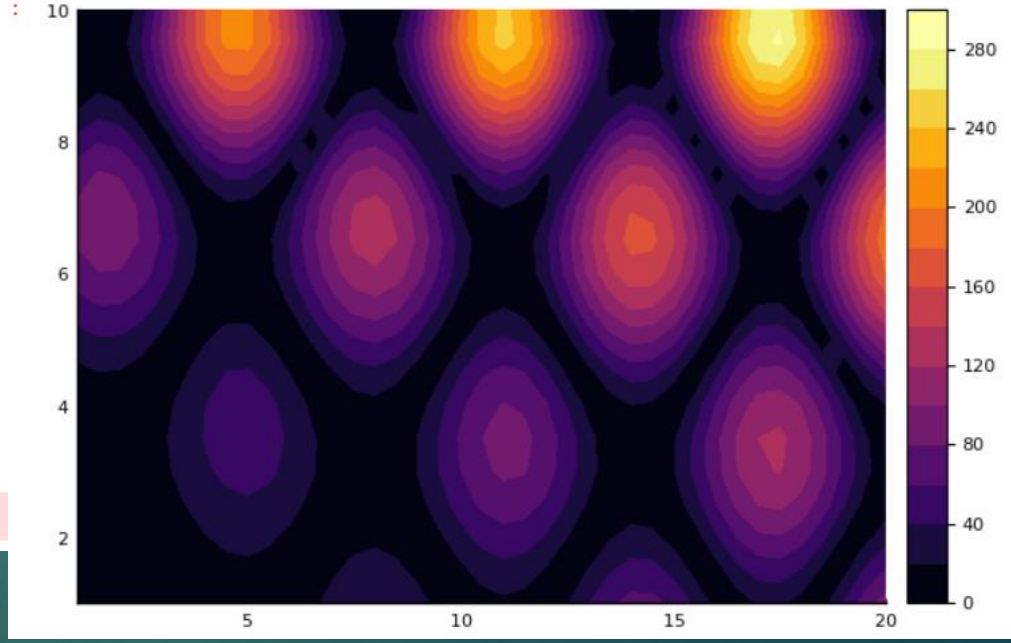


```
#Линии уровня можно построить, используя проекцию значений исходной функции на плоскость:
contour(x, y, g)
```




```
sys:1: UserWarning: The following kwargs were not used by contour: 'label'
```

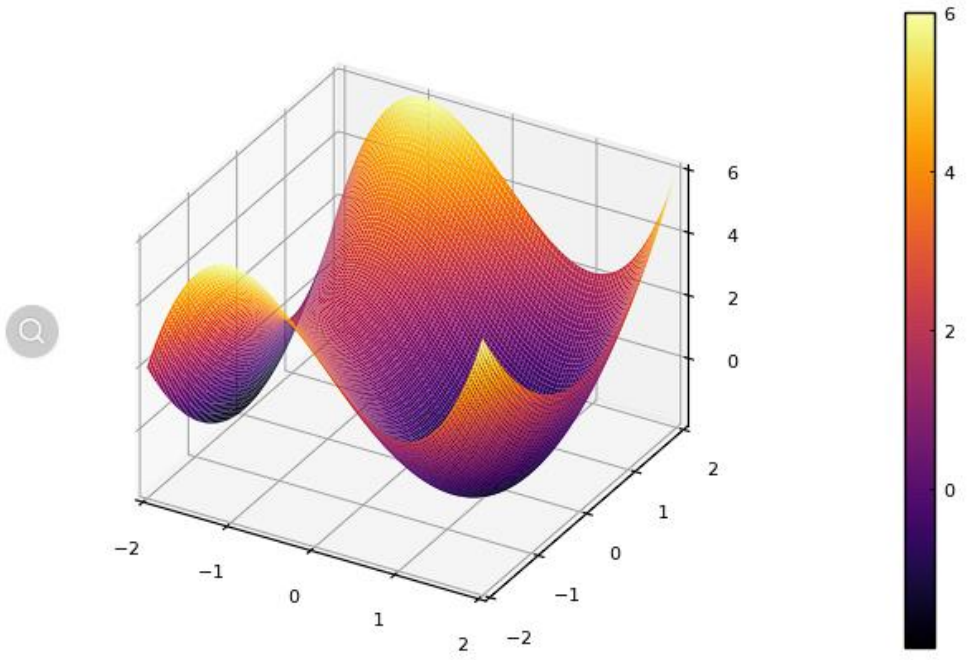
```
#Можно дополнительно добавить заливку цветом:
p = contour(x, y, g,
fill=true)
plot(p)
```



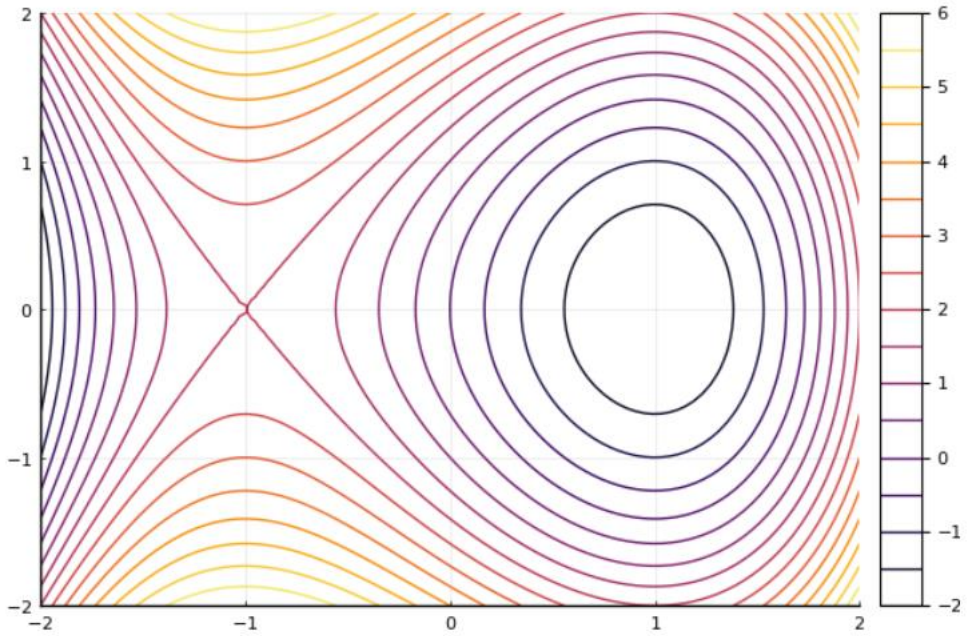
Ввод [180]:

```
X0 = range(-2, stop=2, length=100)
Y0 = range(-2, stop=2, length=100)
h(x, y) = x^3 - 3x + y^2
plot(X0,Y0,h,
linetype = :surface
)
```

Out[180]:




```
# построение линий уровня:
contour(X0, Y0, h)
```



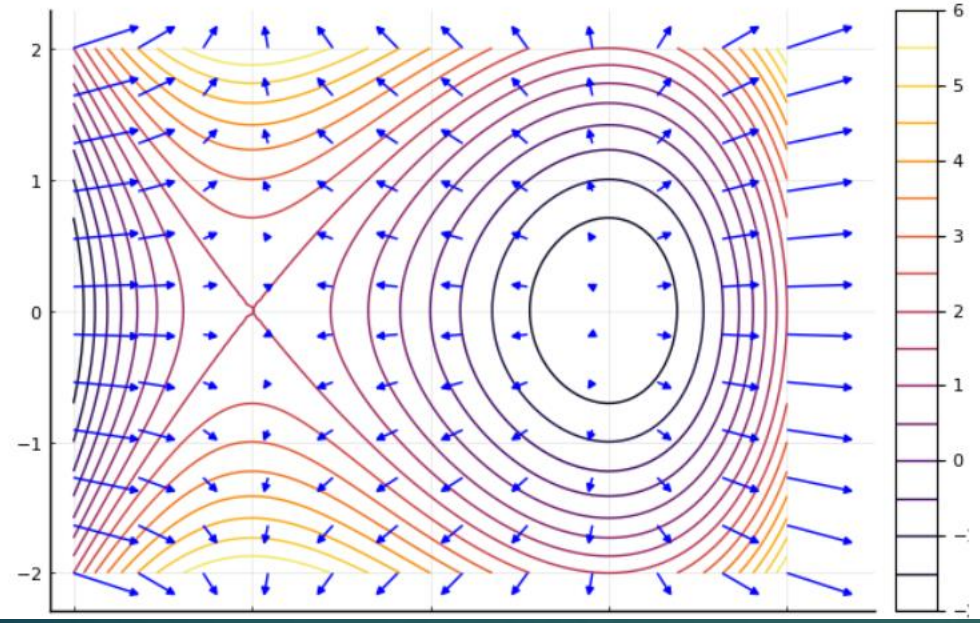
```
# градиент:
x = range(-2, stop=2, length=12)
y = range(-2, stop=2, length=12)
```

```
-2.0:0.36363636363636365:2.0
```

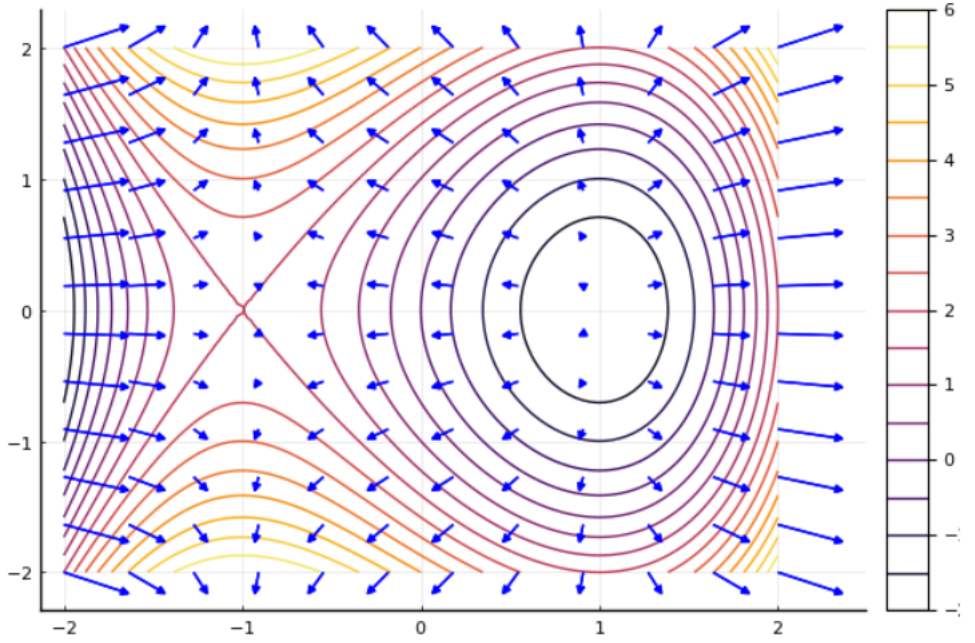
```
# производная от исходной функции:
dh(x, y) = [3x^2 - 3; 2y] / 25
```

```
dh (generic function with 1 method)
```

```
# построение векторного поля:
quiver!(x, y', quiver=dh, c=:blue)
```

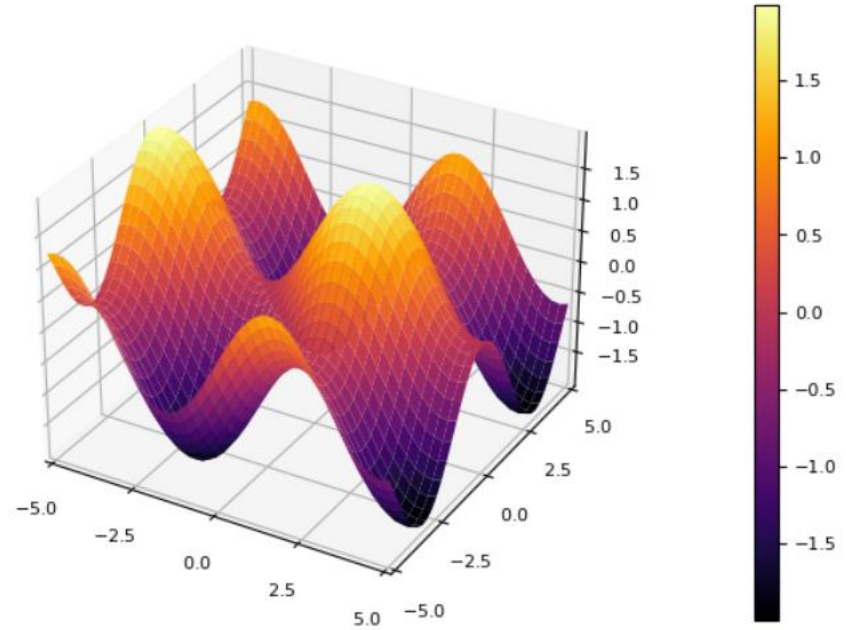



```
# построение векторного поля:
quiver!(x, y', quiver=dh, c=:blue)
```



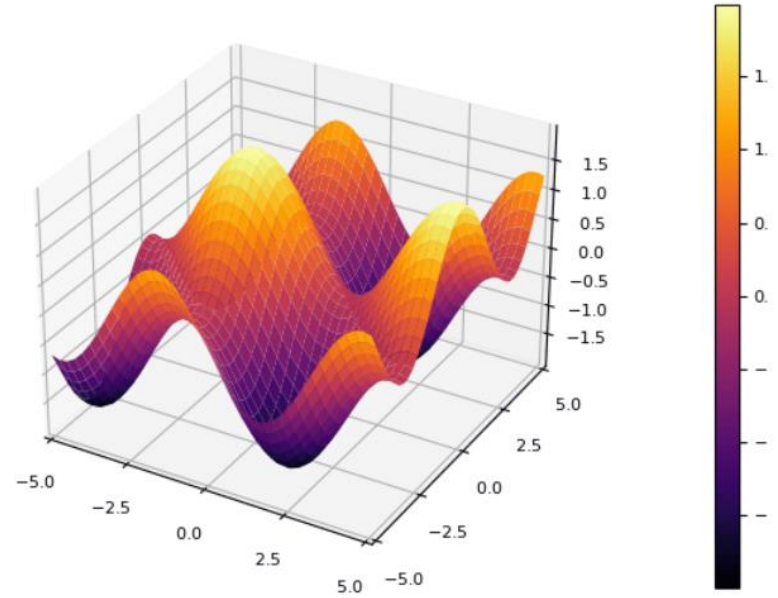

```
sys:1: UserWarning: The following kwargs were not used by contour: 'label'
```

```
# анимация:
X0 = Y0 = range(-5,stop=5,length=40)
@gif for i in range(0,stop=2π,length=100)
surface(X0, Y0, (x,y) -> sin(x+10sin(i))+cos(y))
end
```




```
Info: Saved animation to
  fn = C:\Users\Admin\tmp.gif
@ Plots C:\Users\Admin\.julia\packages\Plots\uCh2y\src\animation.jl:104
```

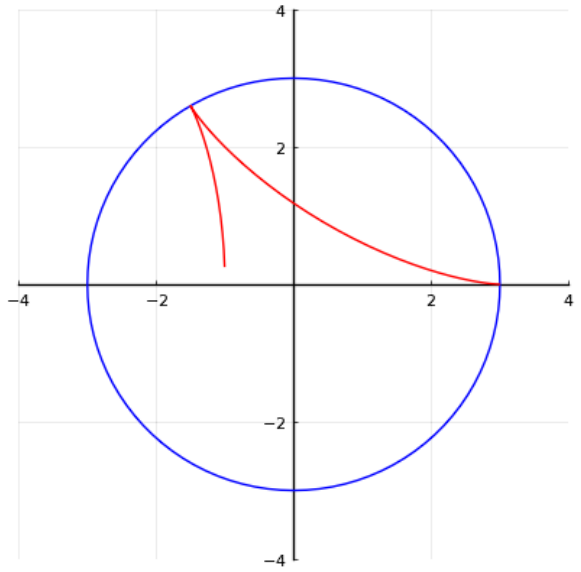
Out[87]:



Ввод [187]:

```
i = 50
t = θ[1:i]
# гипоциклоида:
x = r0*(k-1)*cos.(t) + r0*cos.((k-1)*t)
y = r0*(k-1)*sin.(t) - r0*sin.((k-1)*t)
plot!(x,y, c=:red)
```

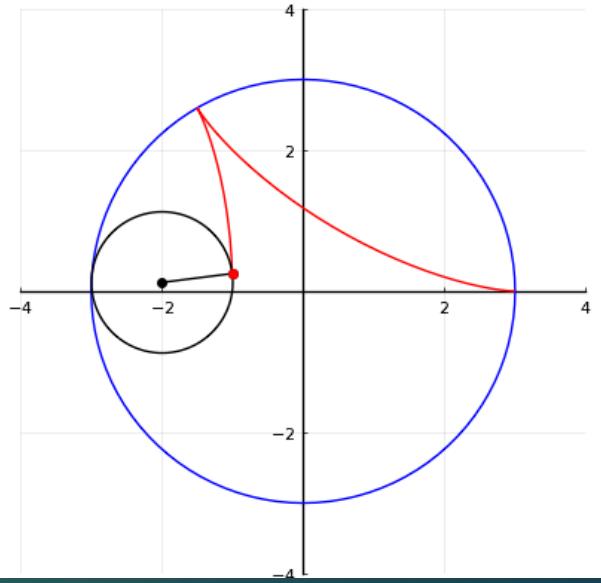
Out[187]:



Ввод [189]:

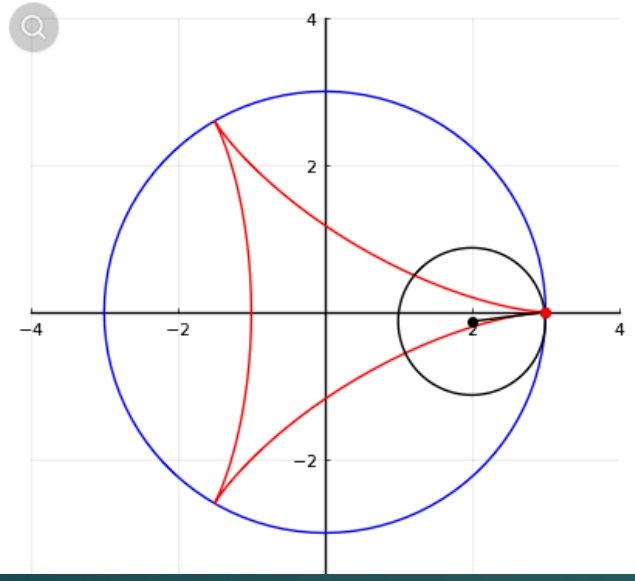
```
# радиус малой окружности:
xl = transpose([r0*(k-1)*cos(t[end]) x[end]])
yl = transpose([r0*(k-1)*sin(t[end]) y[end]])
plot!(xl,yl,markershape=:circle,markersize=4,c=:black)
scatter!([x[end]],[y[end]],c=:red, markerstrokecolor=:red)
```

Out[189]:



Ввод [190]:

```
#В конце сделаем анимацию получившегося изображения
anim = @animate for i in 1:n
# задаём оси координат:
plt=plot(5,xlim=(-4,4),ylim=(-4,4), c=:red, aspect_ratio=1,legend=false, fram
# большая окружность:
plot!(plt, X01,Y01, c=:blue, legend=false)
t = θ[1:i]
# гипоциклоида:
x = r0*(k-1)*cos.(t) + r0*cos.((k-1)*t)
y = r0*(k-1)*sin.(t) - r0*sin.((k-1)*t)
plot!(x,y, c=:red)
# малая окружность:
    xc = r0*(k-1)*cos(t[end]) .+ r0*cos.(θ)
    yc = r0*(k-1)*sin(t[end]) .+ r0*sin.(θ)
plot!(xc,yc,c=:black)
# радиус малой окружности:
xl = transpose([r0*(k-1)*cos(t[end]) x[end]])
yl = transpose([r0*(k-1)*sin(t[end]) y[end]])
plot!(xl,yl,markershape=:circle,markersize=4,c=:black)
scatter!([x[end]],[y[end]],c=:red, markerstrokecolor=:red)
end
```



Out[190]:

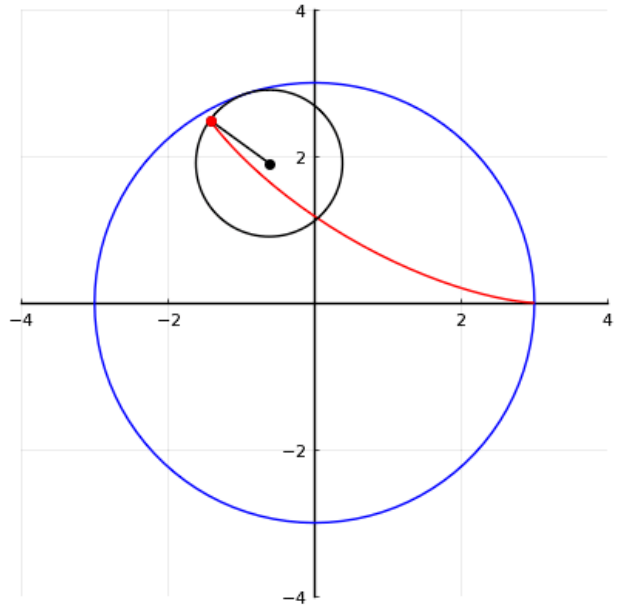
```
Animation("C:\\Users\\KIRR\\AppData\\Local\\Temp\\jl_XNYnpN", ["000001.png",
"000002.png", "000003.png", "000004.png", "000005.png", "000006.png", "00000
7.png", "000008.png", "000009.png", "000010.png"  …  "000091.png", "000092.p
ng", "000093.png", "000094.png", "000095.png", "000096.png", "000097.png", "
000098.png", "000099.png", "000100.png"])
```

Ввод [191]:

```
gif(anim,"hypocycloid.gif")
```

```
[ Info: Saved animation to C:\Users\KIRR\hypocycloid.gif
@ Plots C:\Users\KIRR\.julia\packages\Plots\M4dfL\src\animation.jl:156
```

Out[191]:



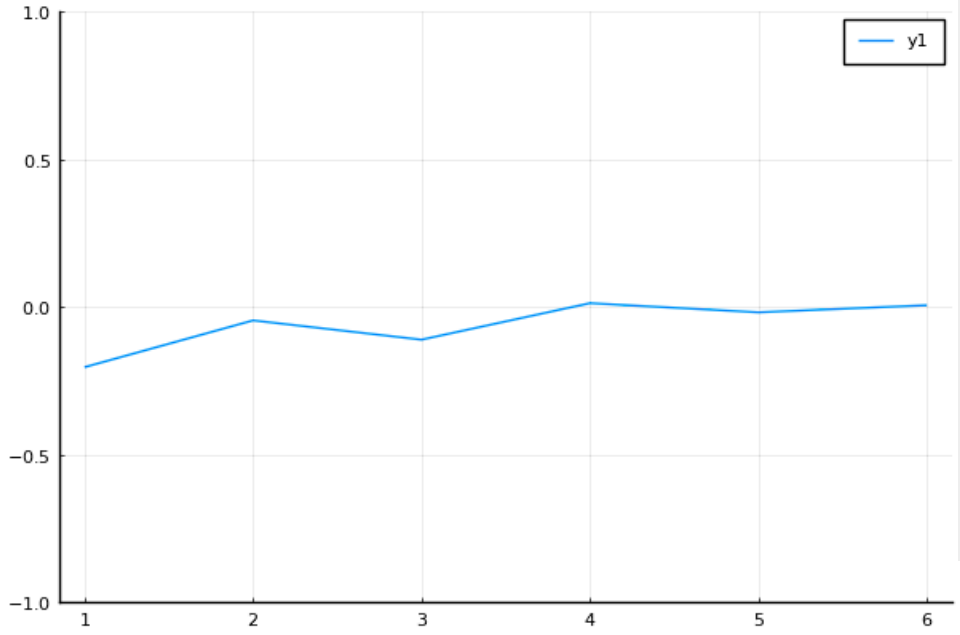
Ввод [192]:

```
Pkg.add("Statistics")
```

Ввод [192]:
```
Pkg.add("Statistics")
using Statistics
sds = [1, 1/2, 1/4, 1/8, 1/16, 1/32]
n = 10
y = [mean(sd*randn(n)) for sd in sds]
errs = 1.96 * sds / sqrt(n)
plot(y,
ylims = (-1,1),
)
```

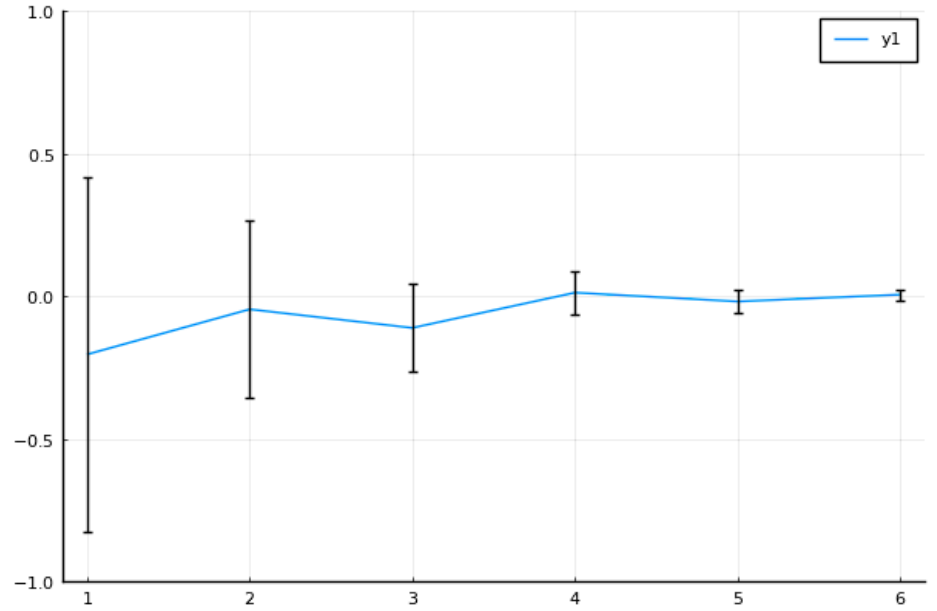
```
  Resolving package versions...
No Changes to `C:\Users\KIRR\.julia\environments\v1.8\Project.toml`
No Changes to `C:\Users\KIRR\.julia\environments\v1.8\Manifest.toml`
```

Out[192]:



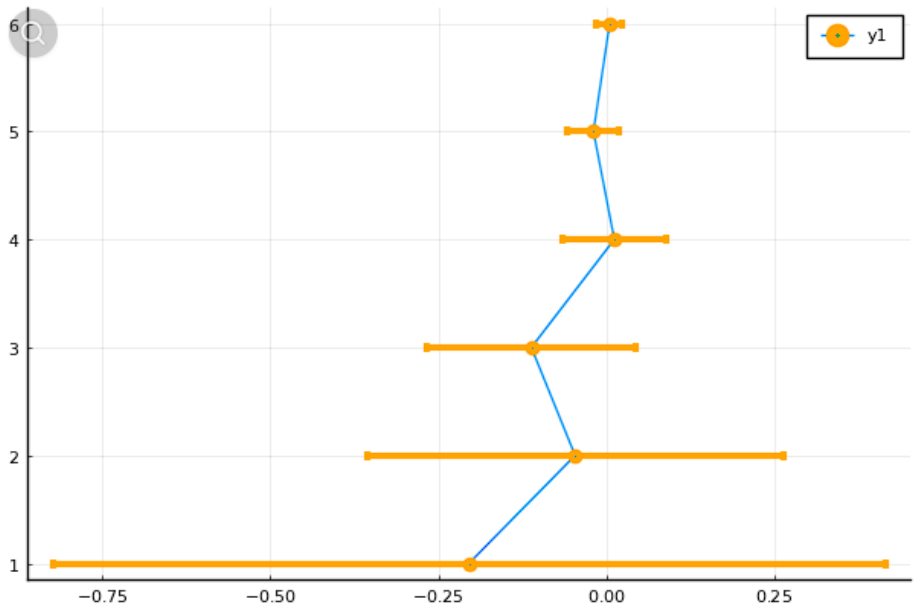
Ввод [193]:
```
plot(y,
ylims = (-1,1),
err = errs
)
```

Out[193]:



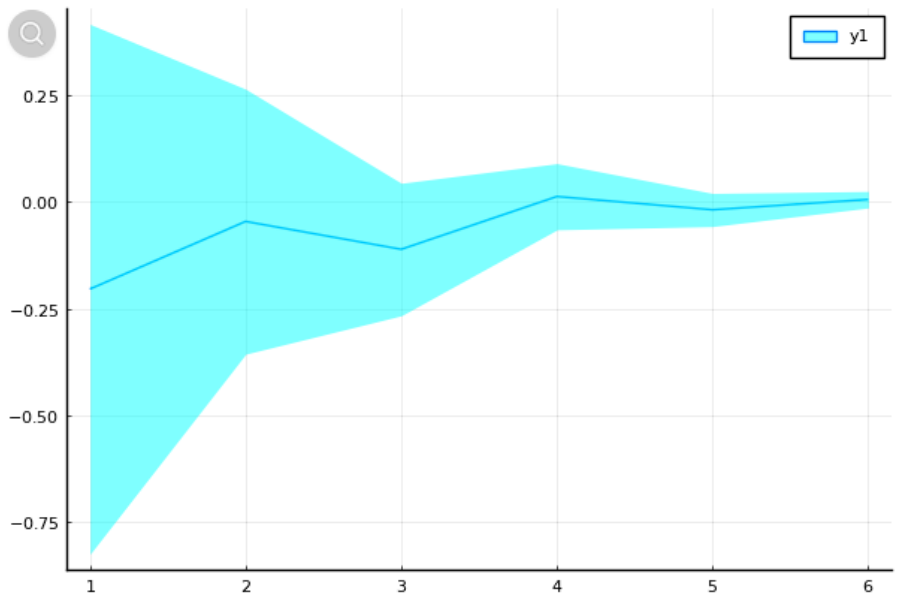
Ввод [194]:
```
plot(y, 1:length(y),
xerr = errs,
marker = stroke(3,:orange)
)
```

Out[194]:




```
fill=:cyan
)
```

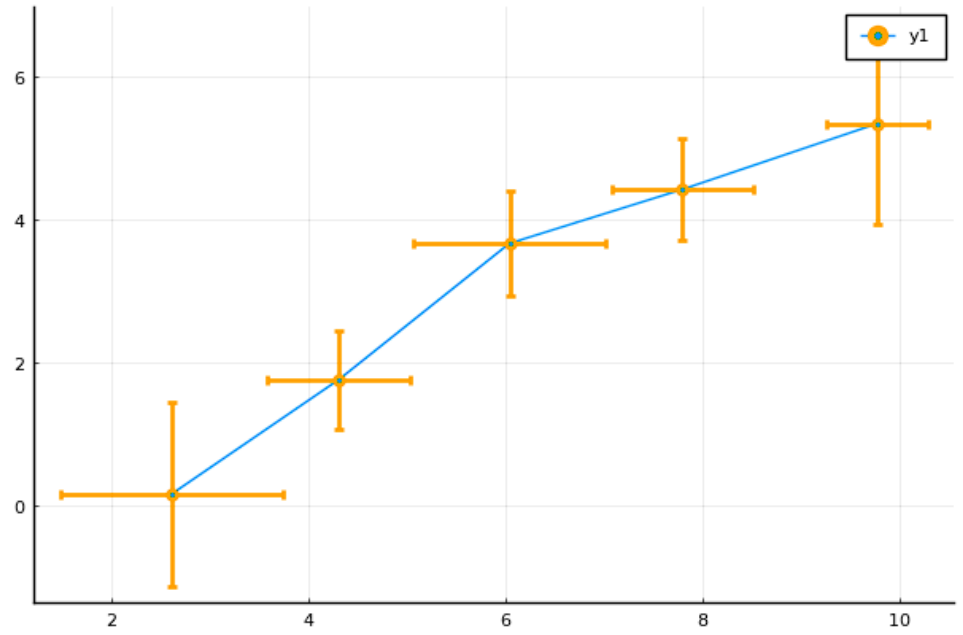
Out[195]:



Ввод [196]:

```
n = 10
x = [(rand()+1) .* randn(n) .+ 2i for i in 1:5]
y = [(rand()+1) .* randn(n) .+ i for i in 1:5]
f(v) = 1.96std(v) / sqrt(n)
xerr = map(f, x)
yerr = map(f, y)
x = map(mean, x)
y = map(mean, y)
plot(x, y,
xerr = xerr,
yerr = yerr,
marker = stroke(2, :orange)
)
```

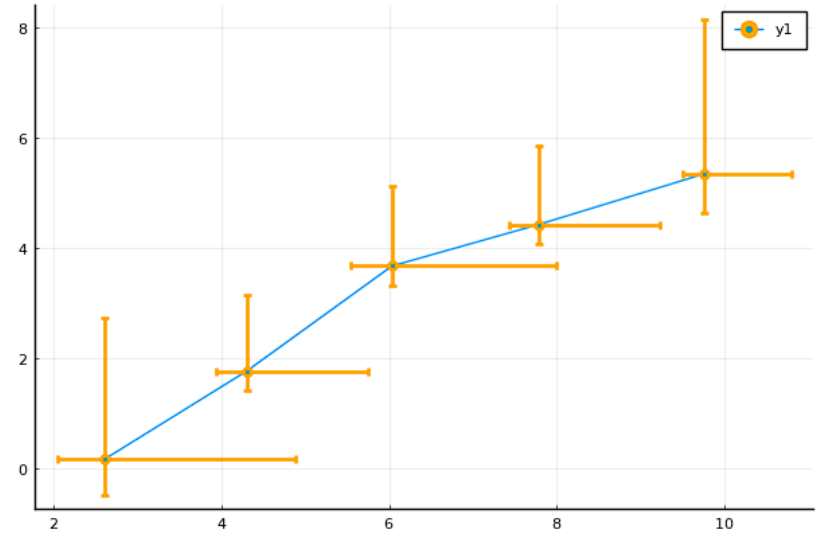
Out[196]:



Ввод [197]:

```
plot(x, y,
xerr = (0.5xerr,2xerr),
yerr = (0.5yerr,2yerr),
marker = stroke(2, :orange)
)
```

Out[197]:

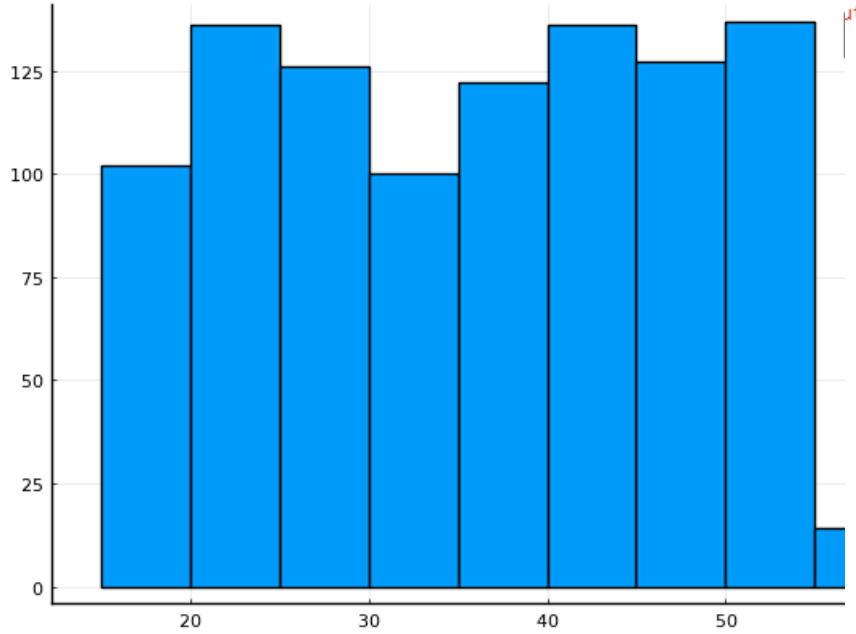


Ввод [198]:

```
Pkg.add("Distributions")
using Distributions
using Plots
pyplot()
ages = rand(15:55,1000)
#Строим гистограмму (рис. 5.41):
histogram(ages)
```

```
  Resolving package versions...
 No Changes to `C:\Users\KIRR\.julia\environments\v1.8\Project.to
 No Changes to `C:\Users\KIRR\.julia\environments\v1.8\Manifest.t
```

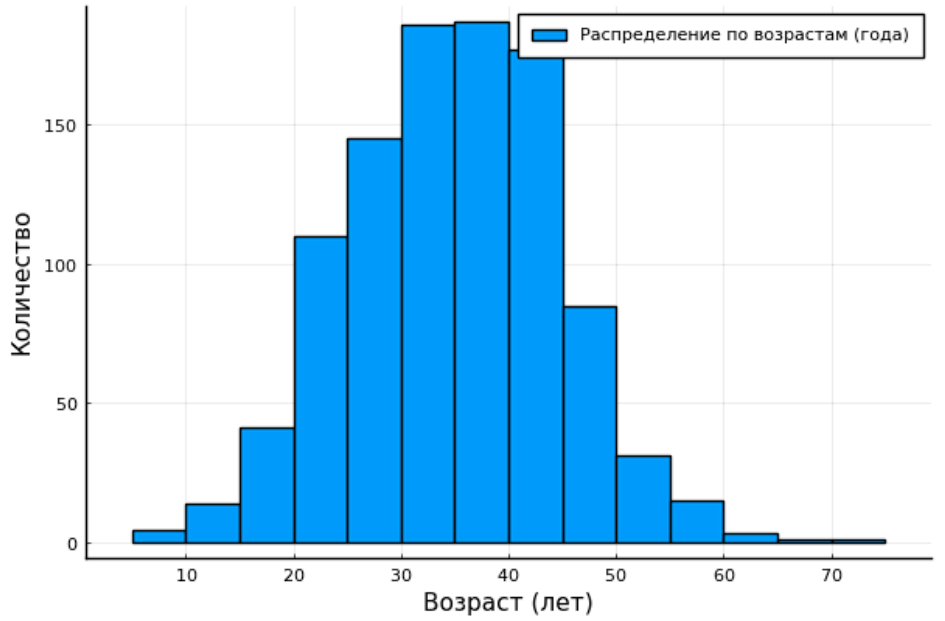
Out[198]:



[199]:

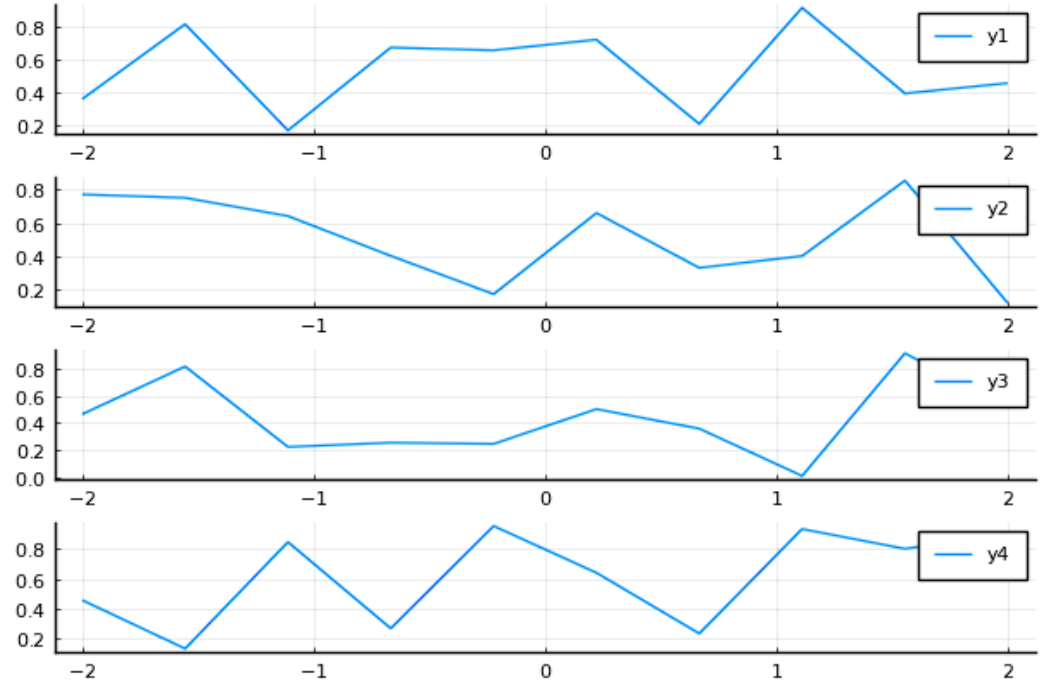
```
d=Normal(35.0,10.0)
ages = rand(d,1000)
histogram(
ages,
label="Распределение по возрастам (года)",
xlabel = "Возраст (лет)",
ylabel= "Количество"
)
```

ut[199]:

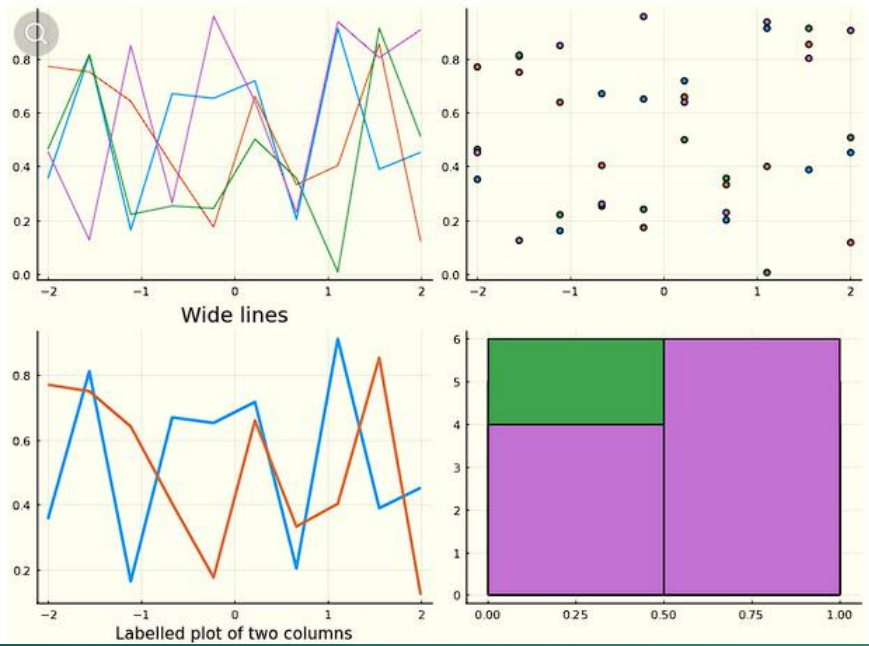



```
Ввод [201]: pyplot()
x=range(-2,2,length=10)
y = rand(10,4)
plot(x,y,
layout=(4,1)
)
```

Out[201]:



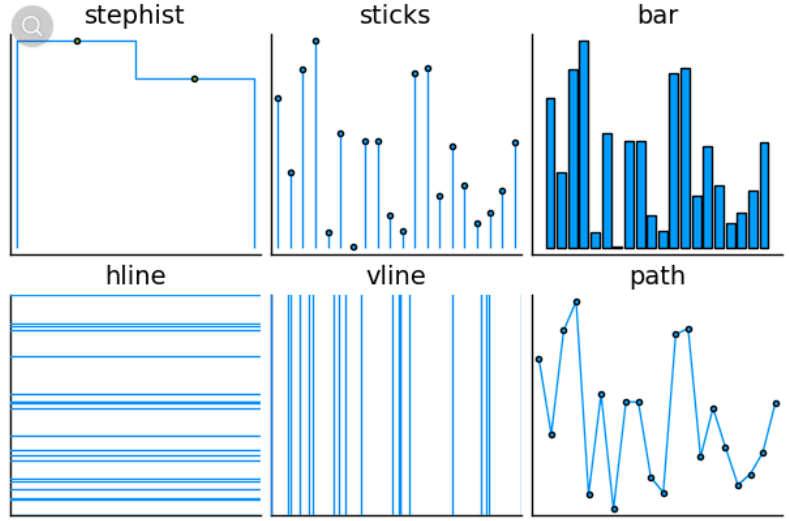
Out[204]:



Ввод [205]:

```
pyplot()
seriestypes = [:stephist, :sticks, :bar, :hline, :vline, :path]
titles =["stephist" "sticks" "bar" "hline" "vline" "path"]
plot(rand(20,1), st = seriestypes,
    layout = (2,3),
    ticks=nothing,
    legend=false,
    title=titles,
    m=3)
```

Out[205]:

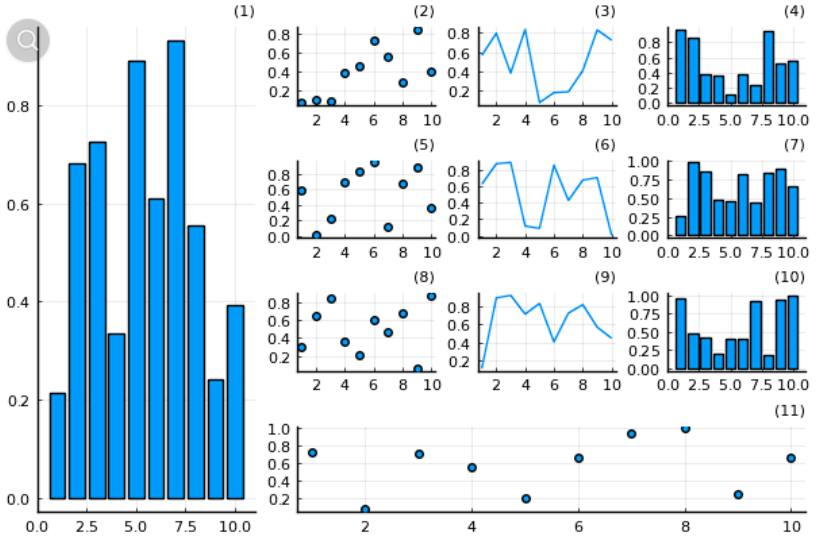


Ввод [206]:

Ввод [206]:

```
l = @layout [ a{0.3w} [grid(3,3)
b{0.2h} ]]
plot(
rand(10,11),
layout = l, legend = false, seriestype = [:bar :scatter :path],
title = ["($i)" for j = 1:1, i=1:11], titleloc = :right, titlefont = font(8)
)
```

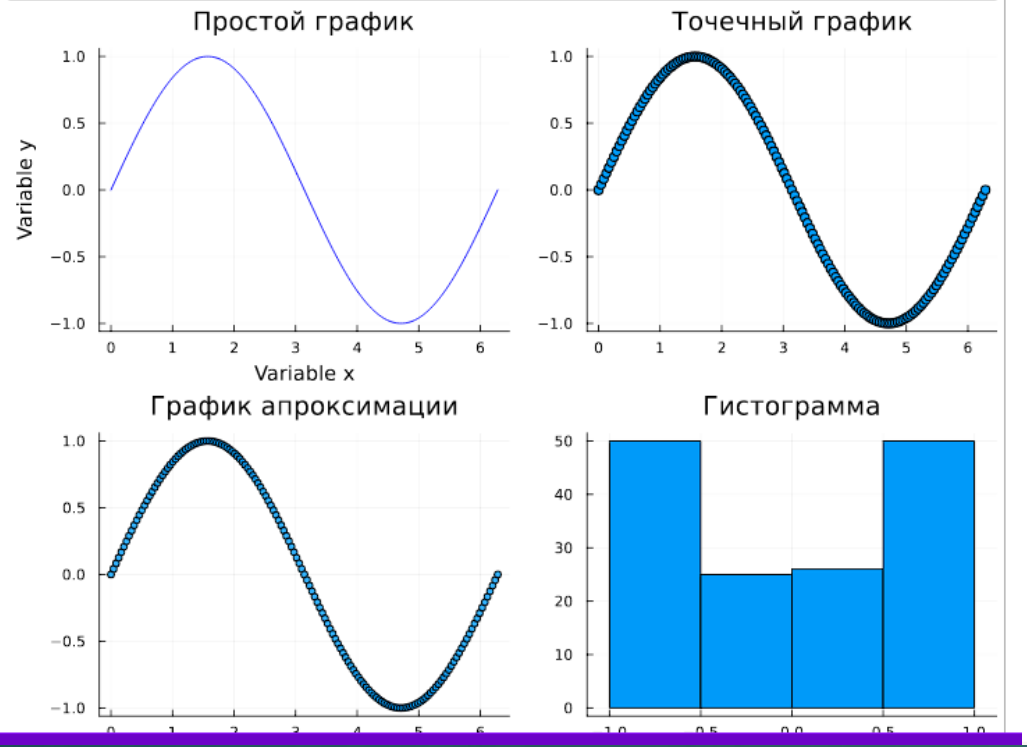
Out[206]:

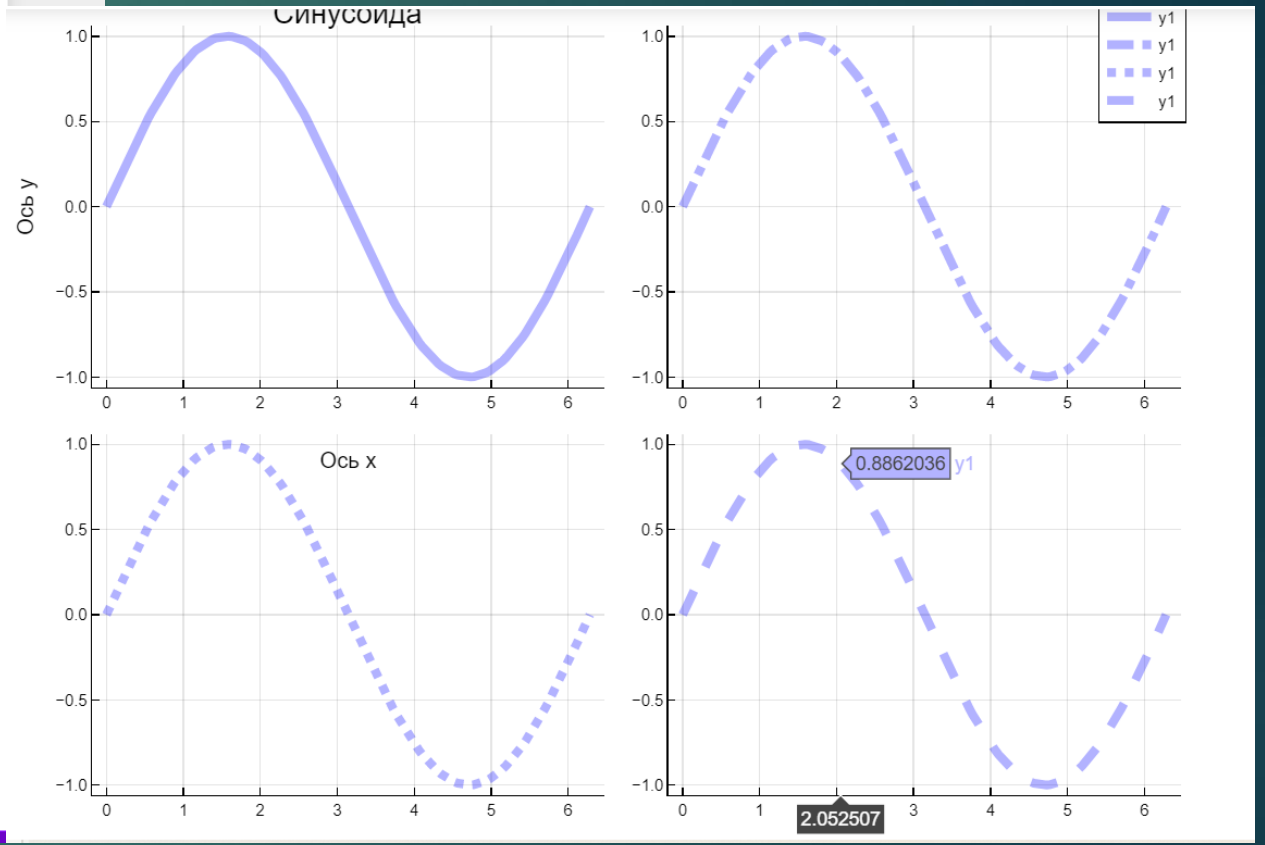


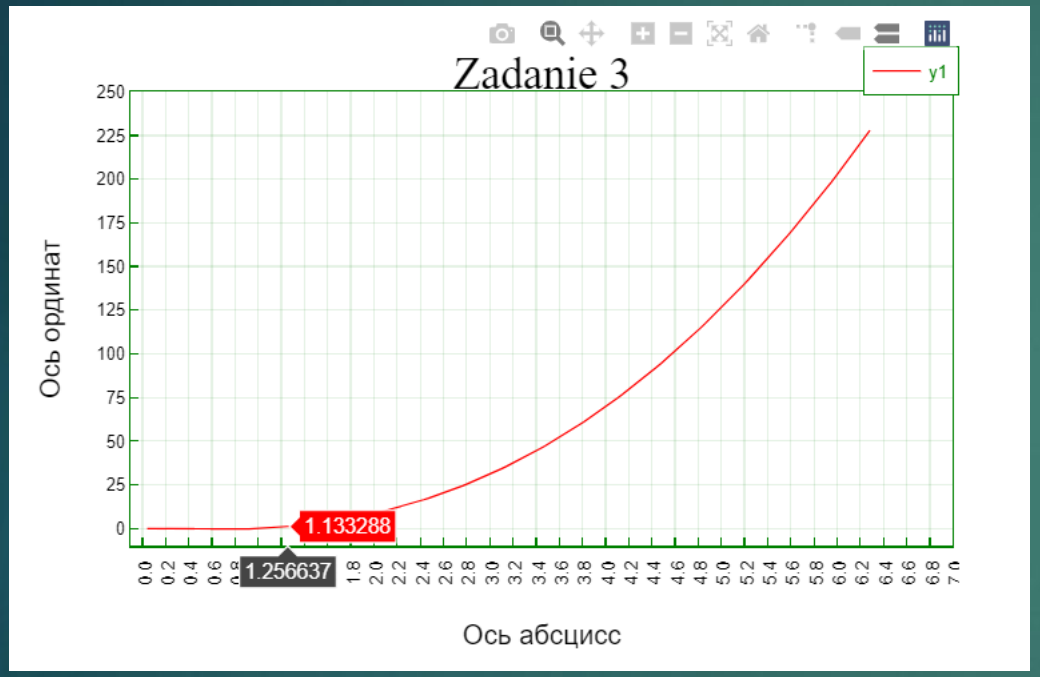
Ввод [210]:

```
f(x)=sin.(x)

x = collect(range(0,2*pi,length=151))

y = f(x)

gr()

p1=plot(x,y, title="Простой график", xlabel="Variable x", ylabel="Variable y",  color="blue")

p2=plot(x, y, seriestype = :scatter, title = "Точечный график")

p3=scatter(x,y,markersize=3,alpha=.8,legend=false, title = "График апроксимации")

p4=histogram(f(x), title = "Гистограмма")

plot(p1,p2,p3,p4, layout=(2,2), legend=false, size=(800,600))
```

Out[210]:





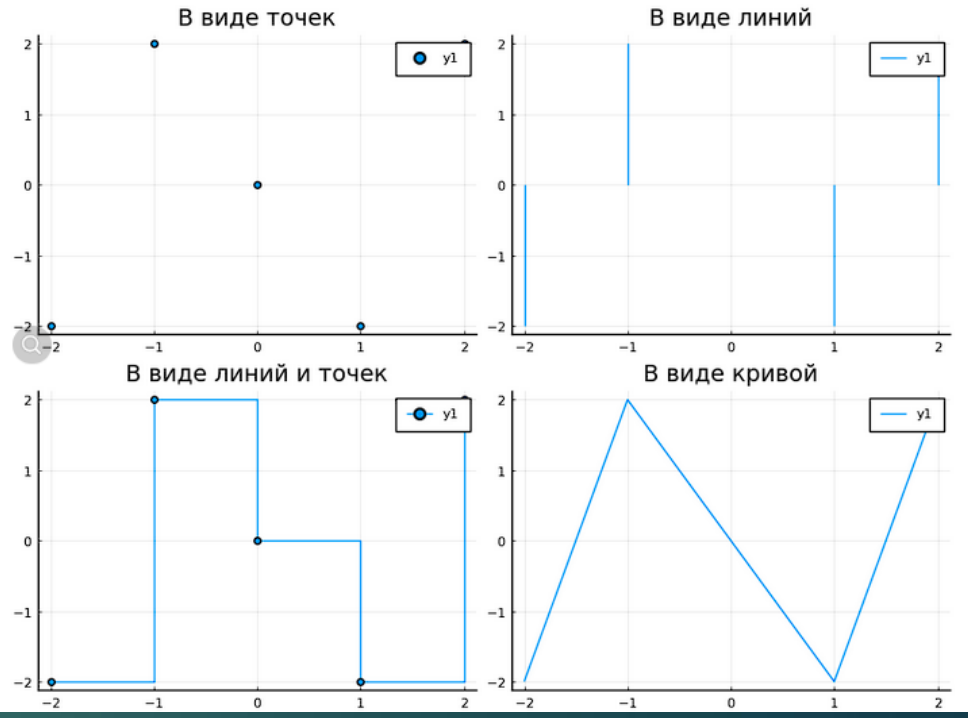


Ввод [215]:

```
pyplot()
f(x) = x.^3 - 3*x
x = [-2, -1, 0, 1, 2]
y = f(x)
p1=plot(x, y, seriestype = :scatter, title = "В виде точек")
p2=plot(x, y, seriestype = :sticks, title= "В виде линий")
p3=plot(x, y, seriestype = :step, marker = 'o', title = "В виде линий и точек")
p4=plot(x, y, title = "В виде кривой")
plot(p1,p2,p3,p4, layout=(2,2), legend=true, size=(800,600))
```

```
[ Warning: Skipped marker arg o.
@ Plots C:\Users\KIRR\.julia\packages\Plots\M4dfL\src\args.jl:1147
```

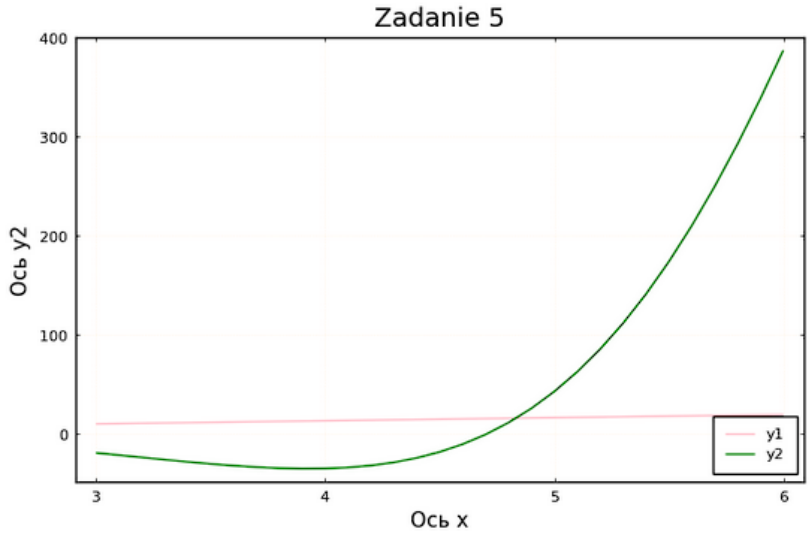
Out[215]:

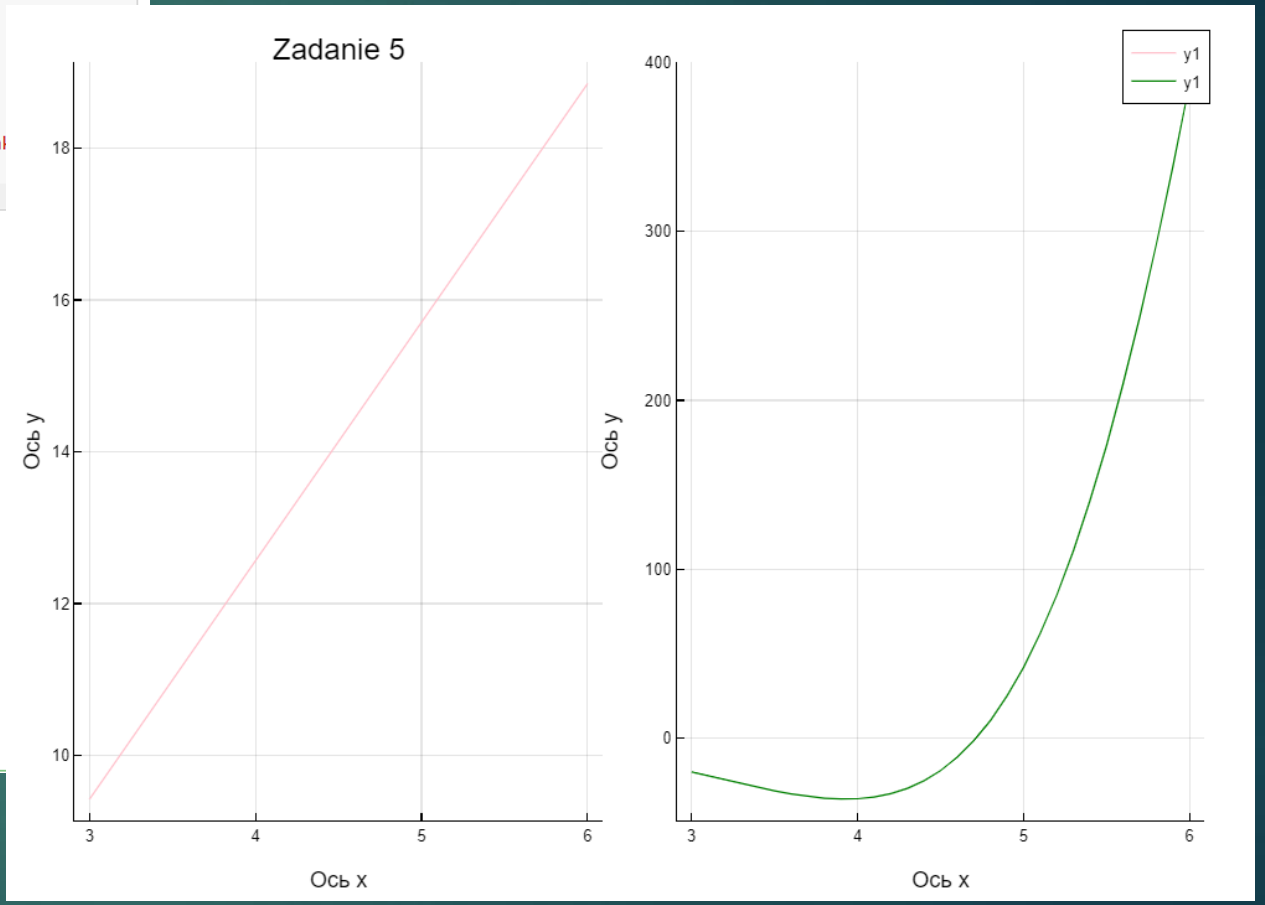


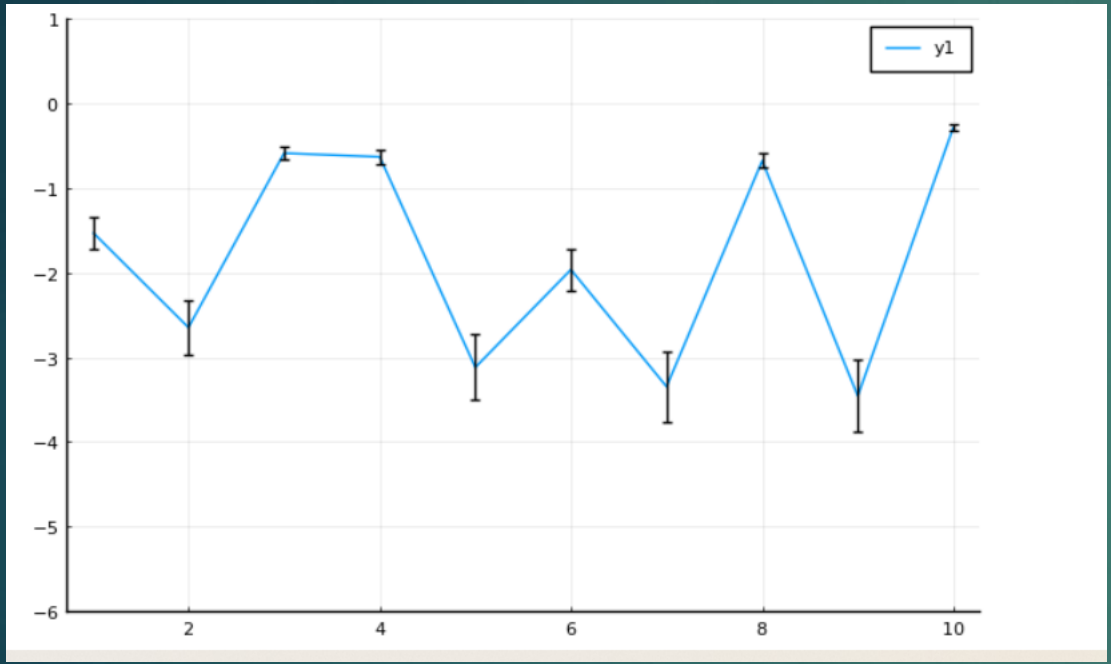
Ввод [217]:

```
pyplot()
f1(x)=pi*x
f2(x)=exp.(x).*cos.(x)
x=[i for i in 3:0.1:6]
y1=f1(x)
y2=f2(x)
plot(x,y1, title="Zadanie 5", xlabel="Ось x", ylabel="Ось y1", leg=:topleft,  color="pink
plot!(x,y2, color="green",  ylabel="Ось y2", leg=:bottomright, grid = :on, box = :on)
```

Out[217]:





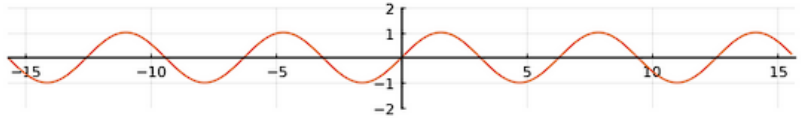

y1
1
0
−1
−2
−3
−4
−5
−6
2
4
6
8
10


Точечный 3d график случайных чисел
y1
Ось z
0.8
0.6
0.4
0.2
0.0
Ось x
0.2
0.4
0.6
0.8
Ось y
0.2
0.4
0.6
0.8

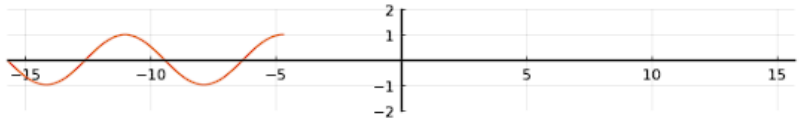
Ввод [223]:

```
pyplot()
n=250
x = collect(-5*pi:2*pi/50:5*pi+pi/100)
anim = @animate for i in 1:n
    plt=plot(1,xlim=(-5*pi,5*pi),ylim=(-2,2), c=:red, aspect_ratio=1,legend=false, framestyle=:origin
    t=x[1:i]
    y=sin.(t)
    plot!(t,y)
end
gif(anim,"sinusoida.gif")
```

```
Info: Saved animation to C:\Users\KIRR\sinusoida.gif
@ Plots C:\Users\KIRR\.julia\packages\Plots\M4dfL\src\animation.jl:156
```



Out[223]:

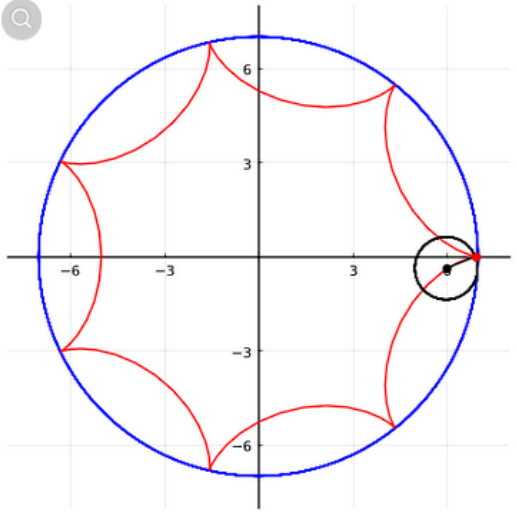


Ввод [224]:

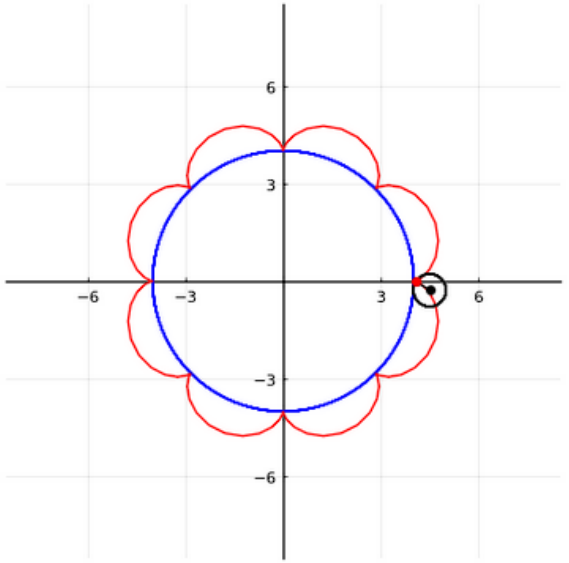
```
function hypocycloid(x, y)
pyplot()
r0 = 1
k = x
n = y
θ = collect(0:2*π/100:10*π+2*π/n)
X01 = r0*k*cos.(θ)
Y01 = r0*k*sin.(θ)
anim = @animate for i in 1:n
plt=plot(5,xlim=(-k-1,k+1),ylim=(-k-1,k+1), c=:red, aspect_ratio=1,legend=false, framestyle=:origin)
plot!(plt, X01,Y01, c=:blue, legend=false)
t = θ[1:i]
x = r0*(k-1)*cos.(t) + r0*cos.((k-1)*t)
y = r0*(k-1)*sin.(t) - r0*sin.((k-1)*t)
plot!(x,y, c=:red)
xc = r0*(k-1)*cos(t[end]) .+ r0*cos.(θ)
yc = r0*(k-1)*sin(t[end]) .+ r0*sin.(θ)
plot!(xc,yc,c=:black)
xl = transpose([r0*(k-1)*cos(t[end]) x[end]])
yl = transpose([r0*(k-1)*sin(t[end]) y[end]])
plot!(xl,yl,markershape=:circle,markersize=4,c=:black)
scatter!([x[end]],[y[end]],c=:red, markerstrokecolor=:red)
end
gif(anim,"hypocycloid1.gif")
end
```

Out[224]: hypocycloid (generic function with 1 method)

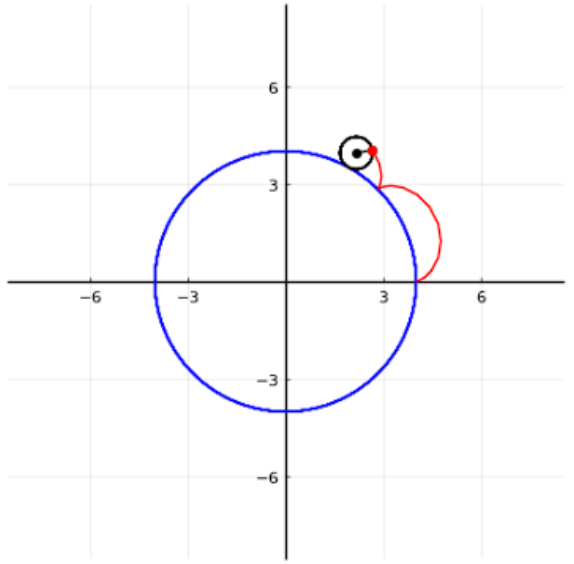
Ввод [226]:
```
hypocycloid(7, 100)
```
```
[ Info: Saved animation to C:\Users\KIRR\hypocycloid1.gif
└ @ Plots C:\Users\KIRR\.julia\packages\Plots\M4dfL\src\animation.jl:156
```



Ввод [234]:
```
epicycloid(8,100)
```
```
[ Info: Saved animation to C:\Users\KIRR\epicycloid.gif
└ @ Plots C:\Users\KIRR\.julia\packages\Plots\M4dfL\src\animation.jl:156
```



Out[234]:



Итог:
Освоил синтаксис языка Julia для построения графиков