

Лабораторная работа 5

Петрушов Дмитрий Сергеевич 1032212287

2024

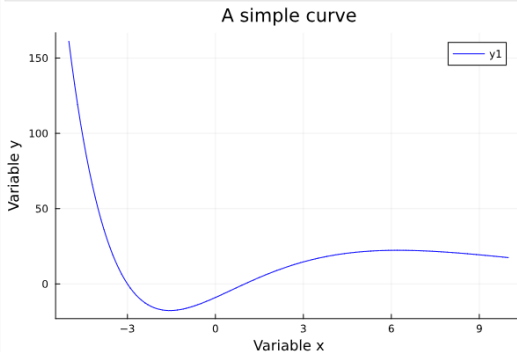
Российский университет дружбы народов, Москва, Россия

Основная цель работы — освоить синтаксис языка Julia для построения графиков.

Выполнение лабораторной работы

Основные пакеты для работы с графиками в Julia

```
# задание функции:  
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="blue")
```



Основные пакеты для работы с графиками в Julia

указывается, что для построения графика используется pyplot():

```
pyplot()
```

задание опций при построении графика

(название кривой, подписи по осям, цвет графика):

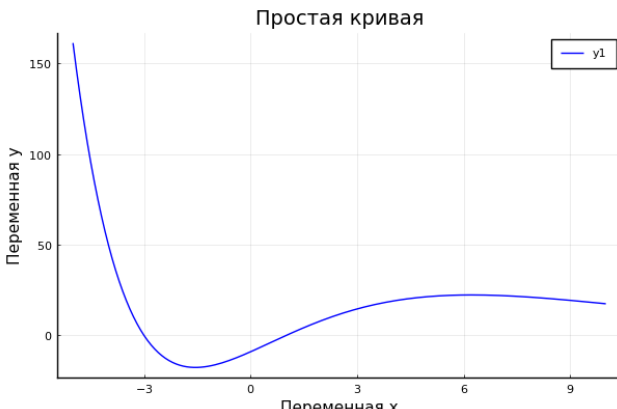
```
plot(x,y,
```

```
    title="Простая кривая",
```

```
    xlabel="Переменная x",
```

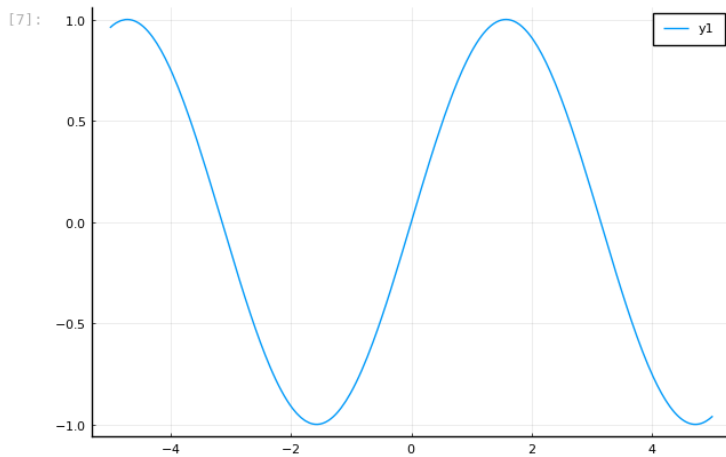
```
    ylabel="Переменная y",
```

```
    color="blue")
```



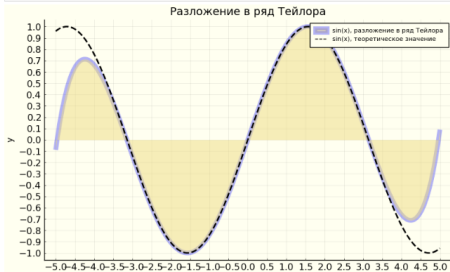
Опции при построении графика

```
[7]: # указывается, что для построения графика используется pyplot():  
plot()  
# задание функции sin(x):  
sin_theor(x) = sin(x)  
# построение графика функции sin(x):  
plot(sin_theor)
```



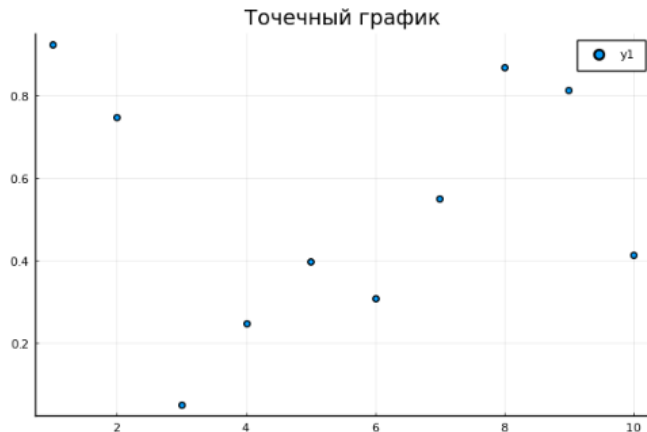
Опции при построении графика

```
plot(  
    # функция sin(x):  
    sin_taylor,  
    # подпись в легенде, цвет и тип линии:  
    label = "sin(x), разложение в ряд Тейлора",  
    line=(:blue, 0.3, 6, :solid),  
    # размер графика:  
    size=(800, 500),  
    # 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))
```



Точечный график

```
# параметры распределения точек на плоскости:  
x = range(1,10,length=10)  
y = rand(10)  
# параметры построения графика:  
plot(x, y,  
      seriestype = :scatter,  
      title = "Точечный график"  
)
```



Точечный график

```
# параметры распределения точек на плоскости:
```

```
n = 50
```

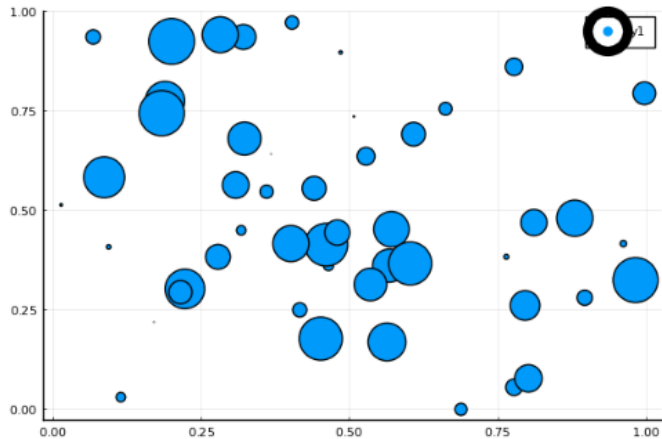
```
x = rand(n)
```

```
y = rand(n)
```

```
ms = rand(50) * 30
```

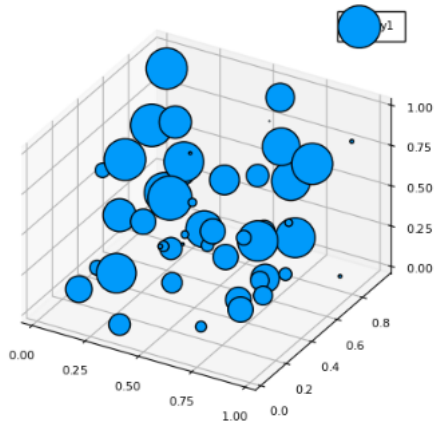
```
# параметры построения графика:
```

```
scatter(x, y, markersize=ms)
```



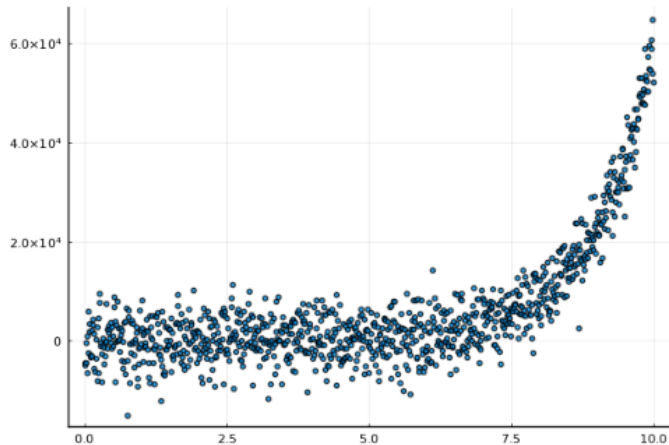
Точечный график

```
: # параметры распределения точек в пространстве:  
n = 50  
x = rand(n)  
y = rand(n)  
z = rand(n)  
ms = rand(50) * 30  
# параметры построения графика:  
scatter(x, y, z, markersize=ms)
```



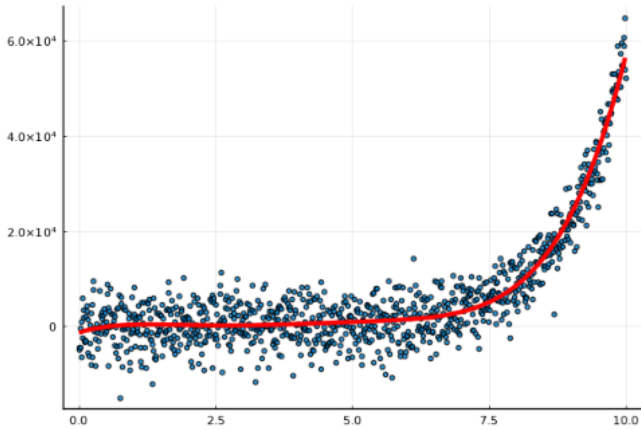
Аппроксимация данных

```
# массив данных от 0 до 10 с шагом 0.01:  
x = collect(0:0.01:9.99)  
# экспоненциальная функция со случайным сдвигом значений:  
y = exp.(ones(1000)+x) + 4000*randn(1000)  
# построение графика:  
scatter(x,y,markersize=3,alpha=.8,legend=false)
```



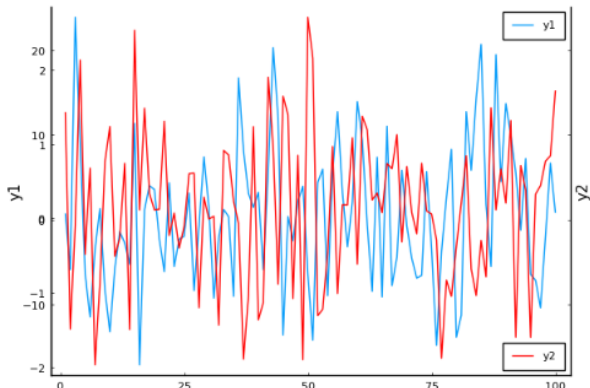
Аппроксимация данных

```
# определение массива для нахождения коэффициентов полинома:  
A = [ones(1000) x x.^2 x.^3 x.^4 x.^5]  
# решение матричного уравнения:  
c = A\y  
# построение полинома:  
f1 = c[1]*ones(1000) + c[2]*x + c[3]*x.^2 + c[4]*x.^3 + c[5]*x.^4 + c[6]*x.^5  
# построение графика аппроксимирующей функции:  
plot(x,f1,linewidth=3, color=:red)
```



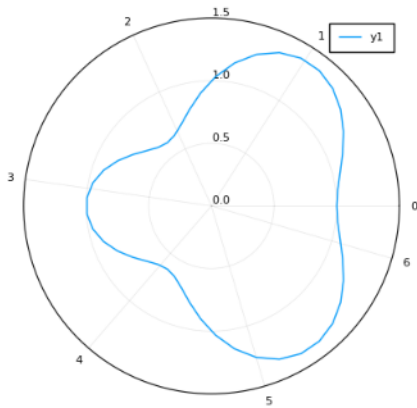
Две оси ординат

```
# 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)
      )
```



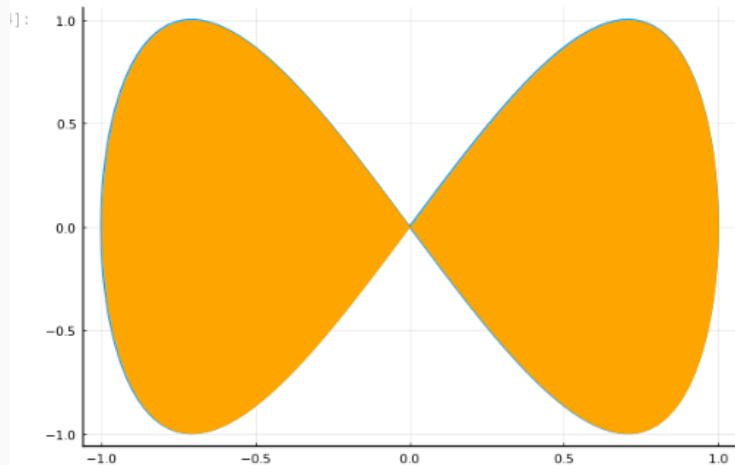
Полярные координаты

```
: # функция в полярных координатах:  
r(θ) = 1 + cos(θ) * sin(θ)^2  
# полярная система координат:  
θ = range(0, stop=2π, length=50)  
# график функции, заданной в полярных координатах:  
plot(θ, r.(θ),  
proj=:polar,  
lims=(0,1.5)  
)
```



Параметрический график

```
1]: # параметрическое уравнение:  
x1(t) = sin(t)  
y1(t) = sin(2t)  
# построение графика:  
plot(x1, y1, 0, 2π, leg=false, fill=(0,:orange))
```



Параметрический график

```
: # параметрическое уравнение  
t = range(0, stop=10, length=1000)  
x = cos.(t)  
y = sin.(t)  
z = sin.(5t)  
# построение графика:  
plot(x, y, z)
```

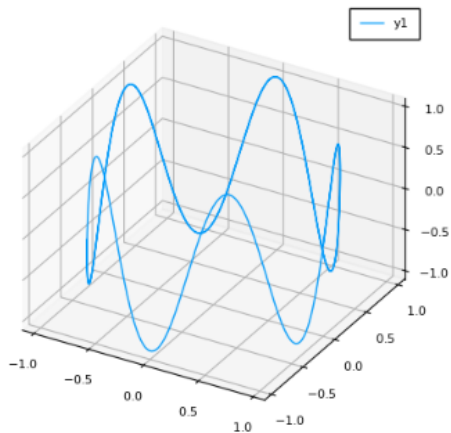


График поверхности

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

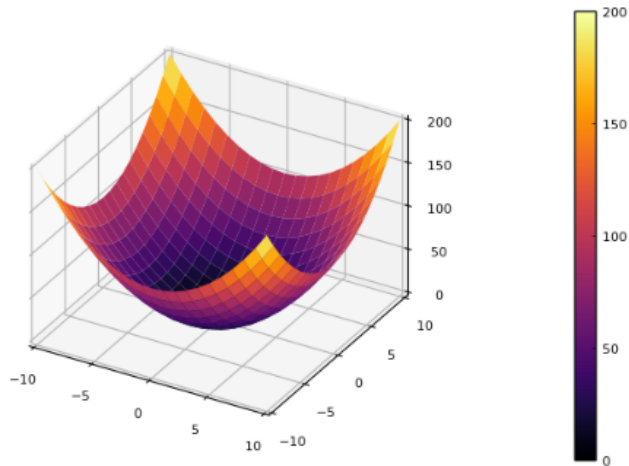
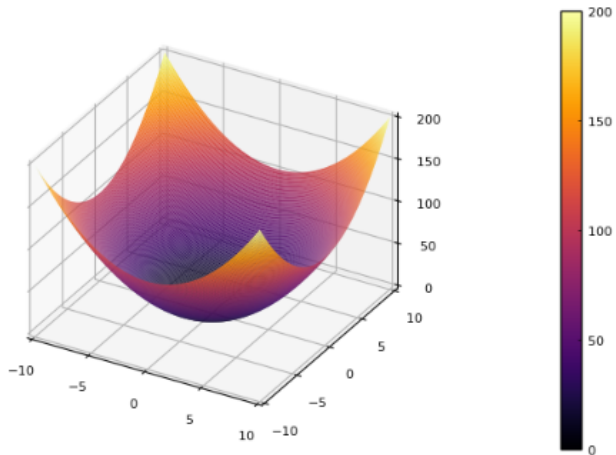


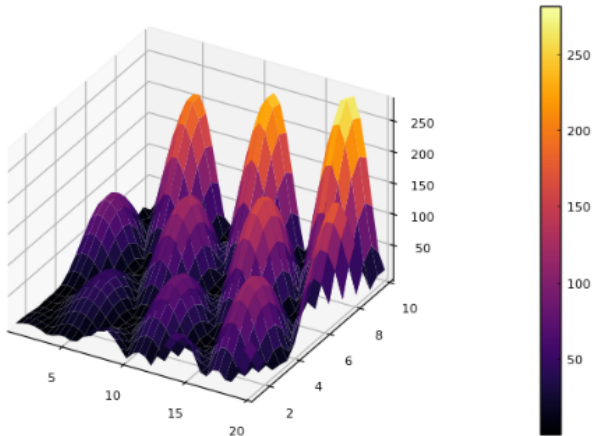
График поверхности

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

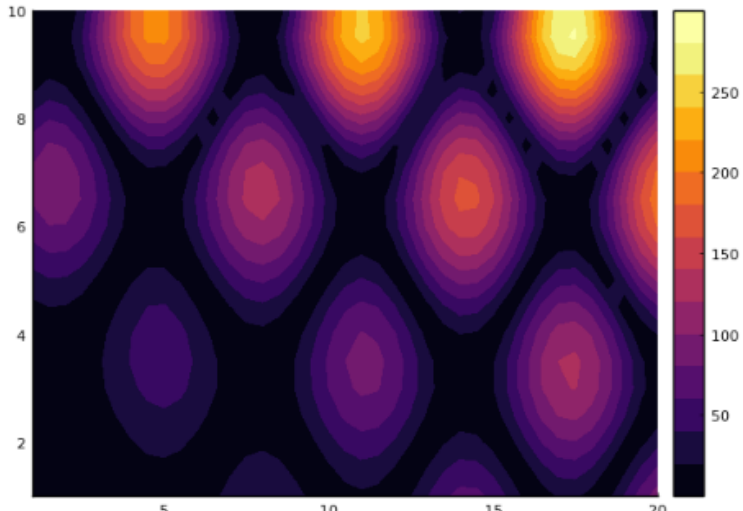


```
] : x = 1:0.5:20  
    y = 1:0.5:10  
    g(x, y) = (3x + y ^ 2) * abs(sin(x) + cos(y))  
    plot(x,y,g,  
         linestyle = :surface,  
         )
```

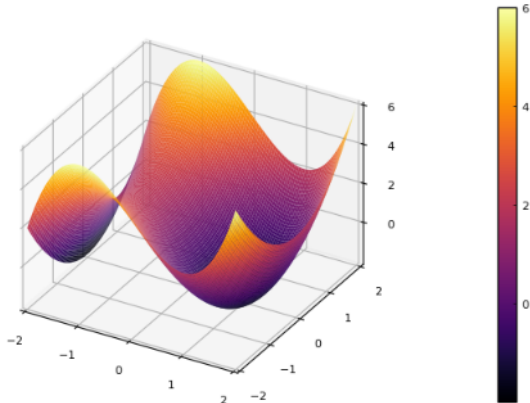
] :



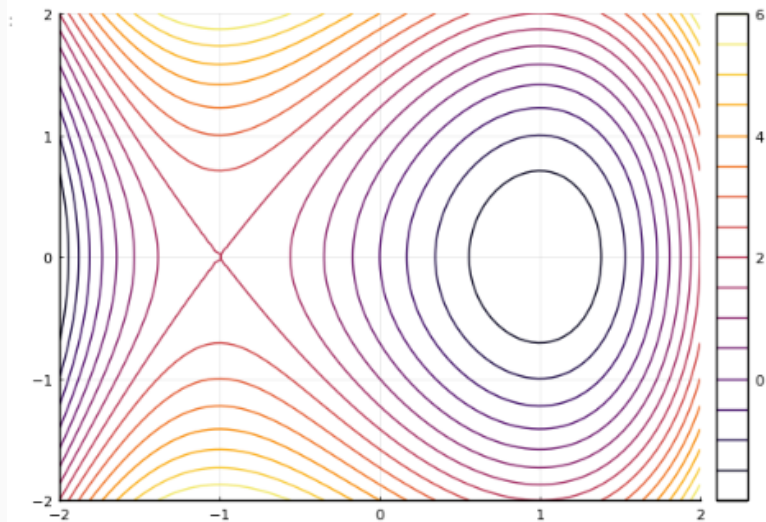
```
p = contour(x, y, g,  
fill=true)  
plot(p)
```

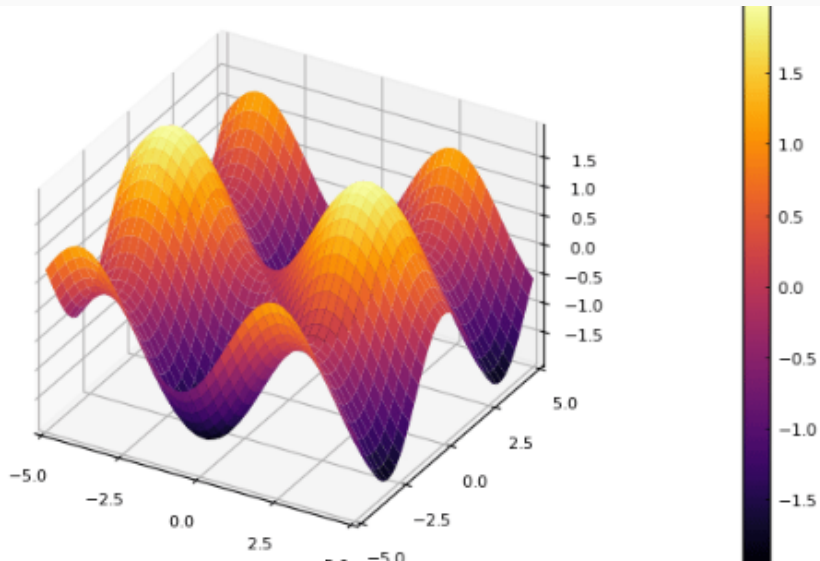


```
# определение переменных:  
X = range(-2, stop=2, length=100)  
Y = range(-2, stop=2, length=100)  
# определение функции:  
h(x, y) = x^3 - 3x + y^2  
# построение поверхности:  
plot(X,Y,h,  
linetype = :surface  
)
```



```
: # построение линий уровня:  
: contour(X, Y, h)
```

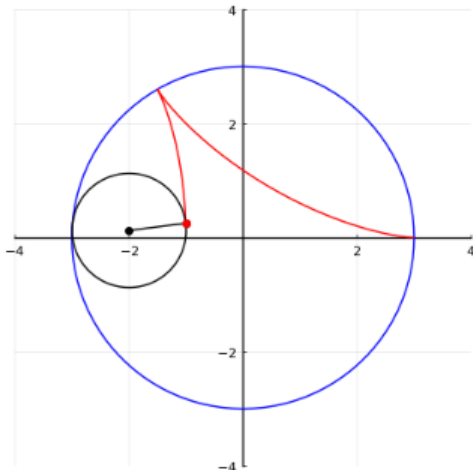




Гипоциклоида

радиус малой окружности:

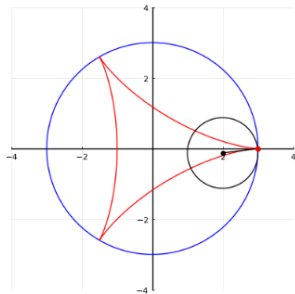
```
x1 = transpose([r1*(k-1)*cos(t[end]) x[end]])  
y1 = transpose([r1*(k-1)*sin(t[end]) y[end]])  
plot!(x1,y1,markershape=:circle,markersize=4,c=:black)  
scatter!([x[end]],[y[end]],c=:red, markerstrokecolor=:red)
```



Гипоциклоида

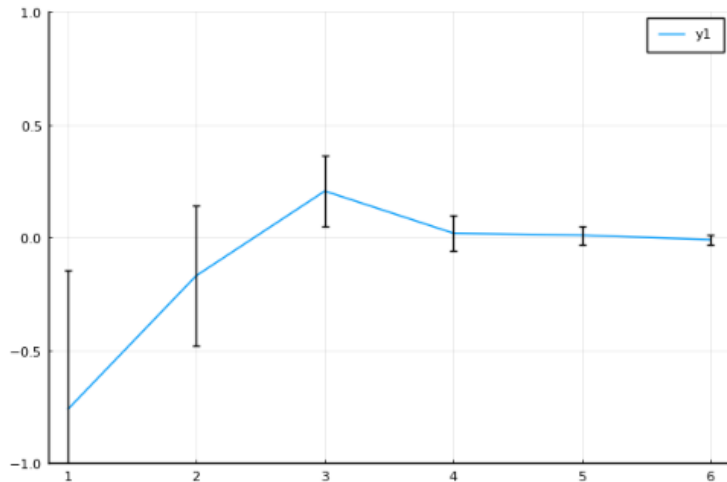
```
anim = @animate for i in 1:n
# задаём оси координат:
plt=plot(5,xlim=(-4,4),ylim=(-4,4), c=:red, aspect_ratio=1,legend=false, framestyle=:origin)
# большая окружность:
plot!(plt, X,Y, c=:blue, legend=false)
t = 0[1:i]
# гипоциклоида:
x = r1*(k-1)*cos.(t) + r1*cos.((k-1)*t)
y = r1*(k-1)*sin.(t) - r1*sin.((k-1)*t)
plot!(x,y, c=:red)
# малая окружность:
xc = r1*(k-1)*cos(t[end]) .+ r1*cos.(θ)
yc = r1*(k-1)*sin(t[end]) .+ r1*sin.(θ)
plot!(xc,yc,c=:black)
# радиус малой окружности:
xl = transpose([r1*(k-1)*cos(t[end]) x[end]])
yl = transpose([r1*(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,"hypocycloid.gif")
```

[Info: Saved animation to /home/hobo/hypocycloid.gif



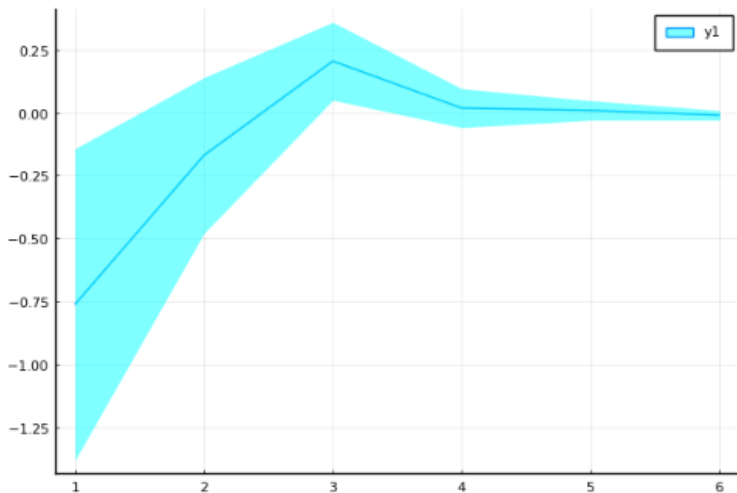
Errorbars

```
plot(y,  
     ylims = (-1,1),  
     err = errs  
)
```



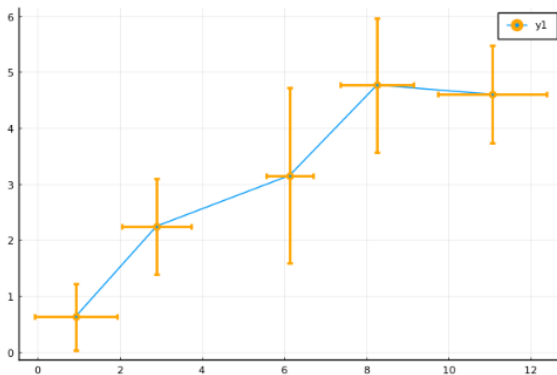
Errorbars

```
plot(y,  
     ribbon=errs,  
     fill=:cyan  
)
```

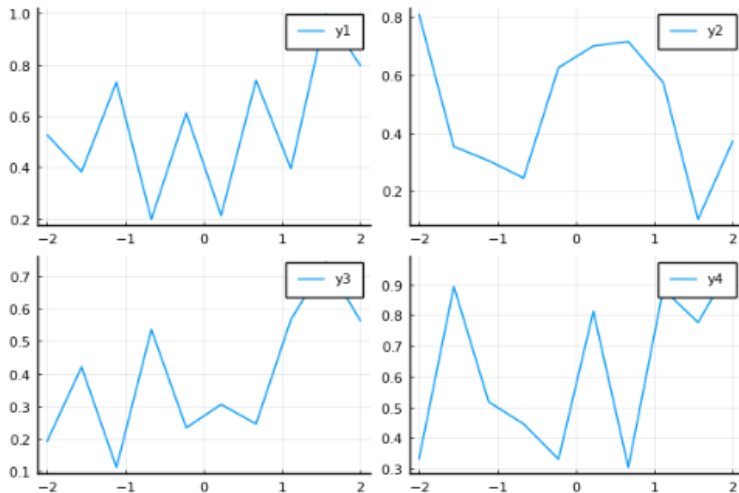


Errorbars

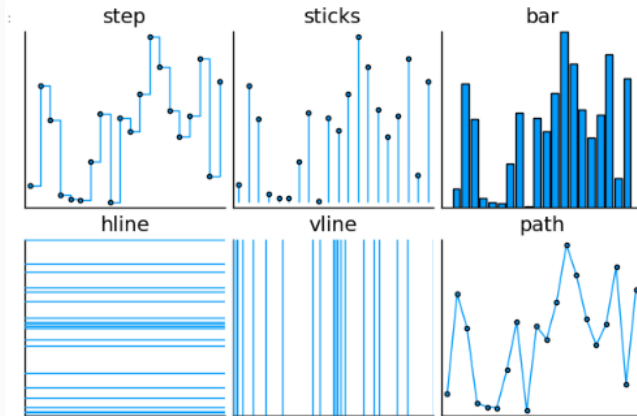
```
n = 10
x = [(rand()+1) .* randn(n) .+ 2i for i in 1:5]
y = [(rand()+1) .* randn(n) .+ i for i in 1:5]
f(v) = 1.96std(v) / sqrt(n)
xerr = map(f, x)
yerr = map(f, y)
x = map(mean, x)
y = map(mean, y)
plot(x, y,
     xerr = xerr,
     yerr = yerr,
     marker = stroke(2, :orange)
)
```



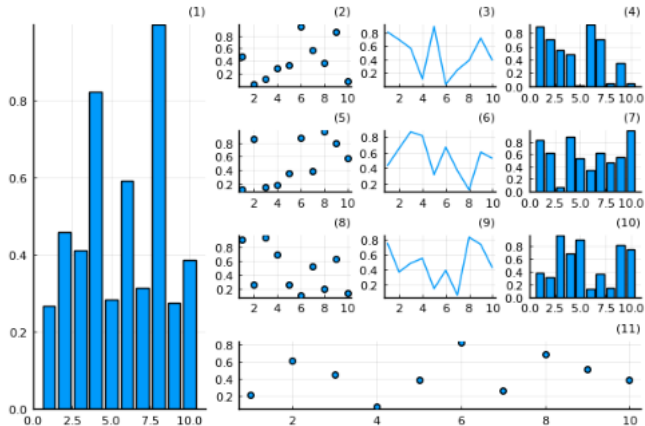
```
plot(x,y,  
      layout=4  
)
```



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



```
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)  
)
```



Вывод

В ходе выполнения лабораторной работы был освоен синтаксис языка Julia для построения графиков.