

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»  
Факультет физико-математических и естественных наук

ОТЧЕТ  
По лабораторной работе №5  
Построение графиков

Выполнил:  
Студент группы: НПИбд-02-21  
Студенческий билет: № 1032217060  
ФИО студента: Королев Адам Маратович

Москва 2024

Цели работы:

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

Выполнение работы:

1. Используя Jupyter Lab, повторите примеры из раздела 5.2. При этом дополните графики обозначениями осей координат, легендой с названиями траекторий, названиями графиков и т.п.

Код:

```
#####  
# Лабораторная работа № 5. Построение графиков  
#####  
  
using Pkg  
Pkg.update()  
Pkg.add("Plots")  
Pkg.add("PyPlot")  
Pkg.add("Plotly")  
Pkg.add("UnicodePlots")  
Pkg.add("Statistics")  
Pkg.add("Distributions")  
Pkg.add("PlotlyBase")  
Pkg.add("Kaleido")  
Pkg.add("IJulia")  
  
using Plots  
using Statistics  
using Distributions  
using PlotlyKaleido  
PlotlyKaleido.restart(timeout=30)  
  
println("Начало выполнения кода...")  
  
#####  
# Пример построения графика функции  $f(x) = (3x^2 + 6x - 9)e^{-0.3x}$   
#####  
  
f(x) = (3x.^2 + 6x .- 9).*exp.(-0.3x)  
x = collect(range(-5,10,length=151))  
y = f(x)  
  
# По умолчанию gr():  
gr()  
p_gr = plot(x, y,  
            title="A simple curve",  
            xlabel="Variable x",  
            ylabel="Variable y",  
            color="blue")  
display(p_gr)  
println("Построен график f(x) при помощи gr().")  
  
# Используем pyplot() с русскими надписями:
```

```

# Используем pyplot() с русскими надписями:
pyplot()
p_py = plot(x, y,
            title="Простая кривая",
            xlabel="Переменная x",
            ylabel="Переменная y",
            color="blue")
display(p_py)
println("Построен график f(x) при помощи pyplot() с русскими надписями.")

# График при помощи plotly() (требуется PlotlyBase и Kaleido):
plotly()
p_pl = plot(x, y,
            title="График при помощи plotly()",
            xlabel="Переменная x",
            ylabel="Переменная y",
            color="red")
display(p_pl)
println("Построен график f(x) при помощи plotly().")

# График при помощи unicodeplots() (требуется UnicodePlots):
unicodeplots()
p_un = plot(x, y,
            title="График при помощи unicodeplots()",
            xlabel="Переменная x",
            ylabel="Переменная y",
            color="green")
display(p_un)
println("Построен график f(x) при помощи unicodeplots().")

# Вернемся к pyplot()
pyplot()

#####
# 5.2.2. Опции при построении графика. Пример  $\sin(x)$  и разложение в ряд Тейлора
#####

pyplot()

sin_theor(x) = sin(x)
p_sin = plot(sin_theor, title="График sin(x)")
display(p_sin)

```

```

sin_theor(x) = sin(x)
p_sin = plot(sin_theor, title="График sin(x)")
display(p_sin)
println("Построен график sin(x).")

sin_taylor(x) = [(-1)^i*x^(2*i+1)/factorial(2*i+1) for i in 0:4] |> sum
p_taylor = plot(sin_taylor, title="График разложения sin(x) в ряд Тейлора")
display(p_taylor)
println("Построен график разложения sin(x) в ряд Тейлора.")

p_both = plot(sin_theor, label="sin(x), теоретическое")
plot!(p_both, sin_taylor, label="sin(x), разложение Тейлора")
display(p_both)
println("Выведены две функции на один график (sin и ее разложение).")

p_options = plot(
    sin_taylor,
    label = "sin(x), разложение в ряд Тейлора",
    line=(blue, 0.3, 6, :solid),
    size=(800, 500),
    xticks = (-5:0.5:5),
    yticks = (-1:0.1:1),
    xtickfont = font(12, "Times New Roman"),
    ytickfont = font(12, "Times New Roman"),
    ylabel = "y",
    xlabel = "x",
    title = "Разложение в ряд Тейлора",
    xrotation = rad2deg(pi/4),
    fillrange = 0,
    fillalpha = 0.5,
    fillcolor = :lightgoldenrod,
    background_color = :ivory
)
plot!(p_options,
    sin_theor,
    label = "sin(x), теоретическое значение",
    line=(black, 1.0, 2, :dash)
)
display(p_options)
println("Добавлены опции отображения (цвет, легенда, размер и т.д.).")

#####
# 5.2.3. Точечный график
#####

```

```

# 5.2.3. Точечный график
#####

x = range(1,10,length=10)
y = rand(10)
p_scatter = plot(x, y,
    seriestype = :scatter,
    title = "Точечный график")
display(p_scatter)
println("Построен простой точечный график.")

n = 50
x = rand(n)
y = rand(n)
ms = rand(50)*30
p_ms = scatter(x, y, markersize=ms, title="Точечный график с разными размерами маркеров")
display(p_ms)
println("Построен точечный график с разными размерами маркеров.")

n = 50
x = rand(n)
y = rand(n)
z = rand(n)
ms = rand(50)*30
p_3d = scatter(x, y, z, markersize=ms, title="3D точечный график")
display(p_3d)
println("Построен 3D точечный график.")

#####
# 5.2.4. Аппроксимация данных
#####

x = collect(0:0.01:9.99)
y = exp.(ones(1000)+x) + 4000*randn(1000)
p_data = scatter(x,y, markersize=3, alpha=.8, legend=false, title="Искусственно сгенерированные данные")
display(p_data)
println("Сгенерированы искусственные данные.")

A = [ones(1000) x x.^2 x.^3 x.^4 x.^5]
c = A\y
f_approx = c[1]*ones(1000) + c[2]*x + c[3]*x.^2 + c[4]*x.^3 + c[5]*x.^4 + c[6]*x.^5
plot!(p_data, x, f_approx, linewidth=3, color=:red, label="Аппроксимация полиномом")
display(p_data)
println("Проведена аппроксимация данных полиномом 5-й степени.")

```

```

display(p_data)
println("Проведена аппроксимация данных полиномом 5-й степени.")

#####
# 5.2.5. Две оси ординат
#####

p_twoaxes = plot(randn(100),
    ylabel="y1",
    leg=:topright,
    grid=:off,
    title="Первая траектория")
plot!(twinx(), randn(100)*10,
    c=:red,
    ylabel="y2",
    leg=:bottomright,
    grid=:off,
    box=:on)
display(p_twoaxes)
println("Построен график с двумя осями ординат.")

#####
# 5.2.6. Полярные координаты
#####

r_polar(θ) = 1 + cos(θ)*sin(θ)^2
θ = range(0, stop=2π, length=50)
p_polar = plot(θ, r_polar.(θ),
    proj=:polar,
    lims=(0,1.5),
    title="Полярный график r(θ)")
display(p_polar)
println("Построен график в полярных координатах.")

#####
# 5.2.7. Параметрический график
#####

x_param(t) = sin(t)
y_param(t) = sin(2t)
p_param = plot(x_param, y_param, 0, 2π, leg=false, fill=(0,:orange), title="Параметрический график на плоскости")
display(p_param)
println("Построен параметрический график на плоскости.")

```

```

p_param = plot(x_param, y_param, 0, 2л, leg=false, fill=(0,:orange), title="Параметрический график на плоскости")
display(p_param)
println("Построен параметрический график на плоскости.")

t = range(0, stop=10, length=1000)
x3d = cos.(t)
y3d = sin.(t)
z3d = sin.(5t)
p_3dparam = plot(x3d, y3d, z3d, title="Параметрический график в пространстве")
display(p_3dparam)
println("Построен параметрический график в 3D пространстве.")

#####
# 5.2.8. График поверхности
#####

f_surf(x,y) = x^2 + y^2
x_surf = -10:10
y_surf = x_surf
p_surf = surface(x_surf, y_surf, f_surf, title="График поверхности f(x,y)=x^2+y^2 (surface)")
display(p_surf)
println("Построен график поверхности (surface).")

p_wire = plot(x_surf, y_surf, f_surf,
    linestyle=:wireframe,
    title="График поверхности (wireframe)")
display(p_wire)
println("Построен график поверхности (wireframe).")

x_smooth = -10:0.1:10
y_smooth = x_smooth
p_smooth = plot(x_smooth, y_smooth, f_surf,
    linestyle=:surface,
    title="Сглаженный график поверхности")
display(p_smooth)
println("Построен сглаженный график поверхности.")

x_cam=range(-2,stop=2,length=100)
y_cam=range(sqrt(2),stop=2,length=100)
f_cam(x,y) = x*y - x - y + 1
p_cam = plot(x_cam, y_cam, f_cam,
    linestyle=:surface,
    c=cgrad([:red,:blue]),
    camera=(-30,30),

```

```

        linetype=:surface,
        c=cgrad([:red,:blue]),
        camera=(-30,30),
        title="Поверхность с изменённым углом зрения")
display(p_cam)
println("Построен график поверхности с изменённым углом зрения.")

#####
# 5.2.9. Линии уровня
#####

x_cont = 1:0.5:20
y_cont = 1:0.5:10
g(x,y) = (3x + y^2)*abs(sin(x) + cos(y))

p_gsurf = plot(x_cont,y_cont,g,
               linetype=:surface,
               title="График поверхности g(x,y)")
display(p_gsurf)
println("Построен график поверхности g(x,y).")

p_gcontour = contour(x_cont, y_cont, g,
                    title="Линии уровня g(x,y)")
display(p_gcontour)
println("Построены линии уровня g(x,y).")

p_gfill = contour(x_cont, y_cont, g,
                 fill=true,
                 title="Линии уровня с заполнением")
display(p_gfill)
println("Построены линии уровня с заполнением.")

#####
# 5.2.10. Векторные поля
#####

X = range(-2, stop=2, length=100)
Y = range(-2, stop=2, length=100)
h(x, y) = x^3 - 3x + y^2
p_hsurf = plot(X,Y,h,
               linetype=:surface,
               title="Поверхность h(x,y)=x^3-3x+y^2")
display(p_hsurf)
println("Построен график поверхности h(x,y).")

```



```

title="Поверхность  $h(x,y)=3x^2-3y$ ")
display(p_hsurf)
println("Построен график поверхности  $h(x,y)$ .")

p_hcontour = contour(X,Y,h,
    title="Линии уровня для  $h(x,y)$ ")
display(p_hcontour)
println("Построены линии уровня для  $h(x,y)$ .")

x_vec = range(-2,stop=2,length=12)
y_vec = range(-2,stop=2,length=12)
dh(x, y) = [3x^2 - 3; 2y]/25

p_quiver = quiver(x_vec, y_vec, quiver=dh, c=:blue, title="Векторное поле  $h(x,y)$ ")
xlims!(-2, 2)
ylims!(-2, 2)
display(p_quiver)
println("Построено векторное поле  $h(x,y)$ .")

#####
# Гипоциклоида
#####

r_h = 1
k_h = 3
n = 100
θ_h = collect(0:2π/100:2π+2π/100)
X_h = r_h*k_h*cos.(θ_h)
Y_h = r_h*k_h*sin.(θ_h)

p_hyp = plot(5,xlim=(-4,4),ylim=(-4,4), c=:red, aspect_ratio=1,
    legend=false, framestyle=:origin)
plot!(p_hyp, X_h,Y_h, c=:blue)

i = 50
t_h = θ_h[1:i]
x_h = r_h*(k_h-1)*cos.(t_h) + r_h*cos.((k_h-1)*t_h)
y_h = r_h*(k_h-1)*sin.(t_h) - r_h*sin.((k_h-1)*t_h)
plot!(p_hyp, x_h,y_h, c=:red)

xc = r_h*(k_h-1)*cos(t_h[end]) + r_h*cos.(θ_h)
yc = r_h*(k_h-1)*sin(t_h[end]) + r_h*sin.(θ_h)
plot!(p_hyp, xc,yc,c=:black)

x1 = transpose([r_h*(k_h-1)*cos(t_h[end]) x_h[end]])

```

```

plot!(p_hyp, xc,yc,c=:black)

x1 = transpose([r_h*(k_h-1)*cos(t_h[end]) x_h[end]])
y1 = transpose([r_h*(k_h-1)*sin(t_h[end]) y_h[end]])
plot!(p_hyp, x1,y1,markershape=:circle,markersize=4,c=:black)
scatter!(p_hyp,[x_h[end]],[y_h[end]],c=:red, markerstrokecolor=:red)
display(p_hyp)
println("Построена гипоциклоида.")

#####
# 5.2.12. Errorbars (Погрешности)
#####

sds = [1, 1/2, 1/4, 1/8, 1/16, 1/32]
n = 10
y_err = [mean(sd*randn(n)) for sd in sds]
errs = 1.96 .* sds ./ sqrt(n)

p_yerr = plot(y_err,
    ylims=(-1,1),
    title="График исходных значений")
display(p_yerr)
println("Построен график исходных значений для погрешностей.")

p_yerr2 = plot(y_err,
    ylims=(-1,1),
    err = errs,
    title="График с погрешностями")
display(p_yerr2)
println("Построен график с погрешностями (errorbars).")

p_yerr3 = plot(y_err, 1:length(y_err),
    xerr = errs,
    marker = stroke(3,:orange),
    title="Поворот графика с погрешностями")
display(p_yerr3)
println("Построен график с погрешностями, повернутый.")

p_yerr4 = plot(y_err,
    ribbon=errs,
    fill=:cyan,
    title="Заполнение области погрешностей цветом")
display(p_yerr4)
println("Область погрешностей заполнена цветом.")

```

```

        title="Заполнение области погрешностей цветом")
display(p_yerr4)
println("Область погрешностей заполнена цветом.")

N=10
x_2 = [(rand()+1).*randn(N).+2i for i=1:5]
y_2 = [(rand()+1).*randn(N).+i for i=1:5]
f_err(v) = 1.96*std(v)/sqrt(N)
xerr = map(f_err, x_2)
yerr = map(f_err, y_2)
x_m = map(mean, x_2)
y_m = map(mean, y_2)

p_xyerr = plot(x_m, y_m,
               xerr=xerr,
               yerr=yerr,
               marker=stroke(2,:orange),
               title="График ошибок по двум осям")
display(p_xyerr)
println("Построен график ошибок по двум осям.")

p_xyerr_as = plot(x_m, y_m,
                  xerr=(0.5xerr,2xerr),
                  yerr=(0.5yerr,2yerr),
                  marker=stroke(2,:orange),
                  title="Асимметричные ошибки")
display(p_xyerr_as)
println("Построен график асимметричных ошибок.")

#####
# 5.2.13. Использование пакета Distributions
#####

pyplot()

ages = rand(15:55,1000)
p_hist1 = histogram(ages, title="Гистограмма распределения возраста")
display(p_hist1)
println("Построена гистограмма распределения возраста.")

d=Normal(35.0,10.0)
ages = rand(d,1000)
p_hist2 = histogram(ages,
                    label="Распределение по возрастам (года)",

```

```

ages = rand(d,1000)
p_hist2 = histogram(ages,
    label="Распределение по возрастам (года)",
    xlabel="Возраст (лет)",
    ylabel="Количество",
    title="Гистограмма нормального распределения")
display(p_hist2)
println("Построена гистограмма нормального распределения.")

plotly()
d1=Normal(10.0,5.0)
d2=Normal(35.0,10.0)
d3=Normal(60.0,5.0)
N=1000
ages = Float64[]
append!(ages,rand(d1,Int64(ceil(N/2))))
append!(ages,rand(d2,N))
append!(ages,rand(d3,Int64(ceil(N/3))))
p_hist3 = histogram(ages,
    bins=50,
    label="Распределение по возрастам (года)",
    xlabel="Возраст (лет)",
    ylabel="Количество",
    title="Распределение по возрастам (года)")
display(p_hist3)
println("Построена гистограмма распределения по трем нормальным законам.")

pyplot()

#####
# 5.2.14. Подграфики
#####

x=range(-2,2,length=10)
y = rand(10,4)
p_sub1 = plot(x,y,
    layout=(4,1),
    title="Серия из 4-х графиков в один столбец")
display(p_sub1)
println("Построена серия из 4-х графиков в один столбец.")

p_sub2 = plot(x,y,
    layout=4,
    title="Серия из 4-х графиков в сетке")

```

```

p_sub2 = plot(x,y,
              layout=4,
              title="Серия из 4-х графиков в сетке")
display(p_sub2)
println("Построена серия из 4-х графиков в сетке.")

# Важно: суммы heights должны быть ровно 1.0!
p_sub3 = plot(x,y,
              size=(600,300),
              layout = grid(4,1,heights=[0.2,0.3,0.4,0.1]), # сумма = 1.0
              title="Серия из 4-х графиков разной высоты")
display(p_sub3)
println("Построена серия из 4-х графиков разной высоты.")

p1 = plot(x,y, title="Линейный график")
p2 = scatter(x,y, title="Точечный график")
p3 = plot(x,y[:,1:2],
          xlabel="Labelled plot of two columns",
          lw=2,
          title="Wide lines")
p4 = histogram(x,y, title="Гистограммы")

p_combined = plot(p1,p2,p3,p4,
                  layout=(2,2),
                  legend=false,
                  size=(800,600),
                  background_color = :ivory,
                  title="Объединение нескольких графиков")
display(p_combined)
println("Объединены несколько графиков в одной сетке.")

seriestypes = [:step, :sticks, :bar, :hline, :vline, :path]
titles = ["step" "sticks" "bar" "hline" "vline" "path"]
p_st = plot(rand(20,1), st = seriestypes,
            layout = (2,3),
            ticks=nothing,
            legend=false,
            title=titles,
            m=3)
display(p_st)
println("Показаны различные стили отображения данных (step, sticks, bar, ...).")

l = @layout [ a{0.3w} [grid(3,3)
b{0.2h} ]]
p_complex = plot(rand(10,11)

```

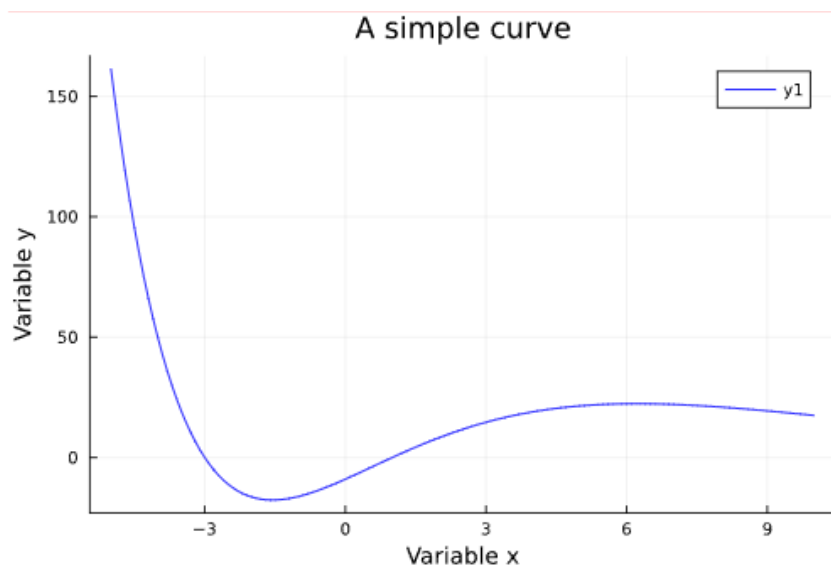
```

l = @layout [ a{0.3w} [grid(3,3)
b{0.2h} ]]
p_complex = 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))
display(p_complex)
println("Показан сложный макет для построения графиков.")

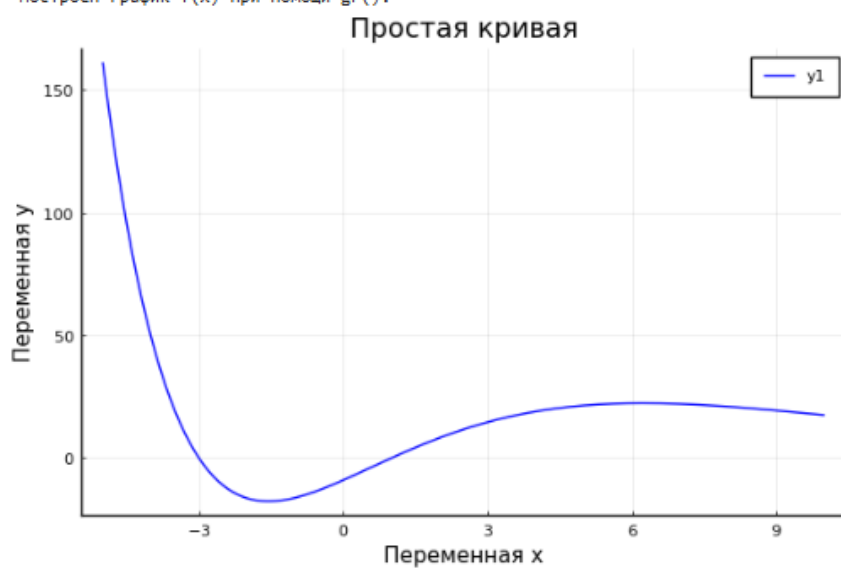
println("Все задания успешно выполнены!")

```

Результаты:

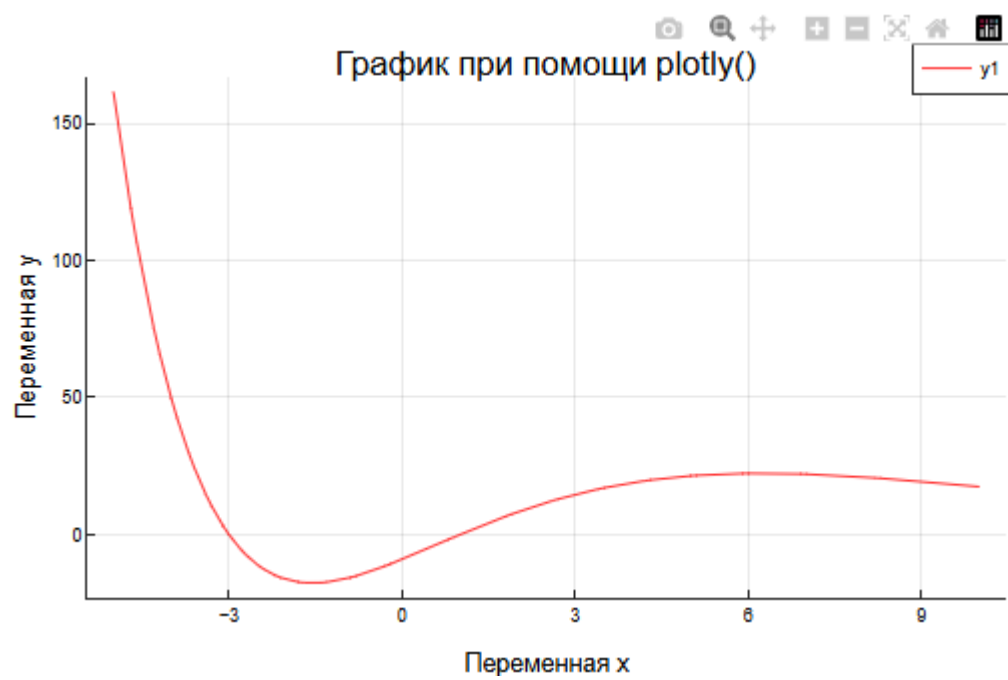


построен график  $f(x)$  при помощи `gr()`.



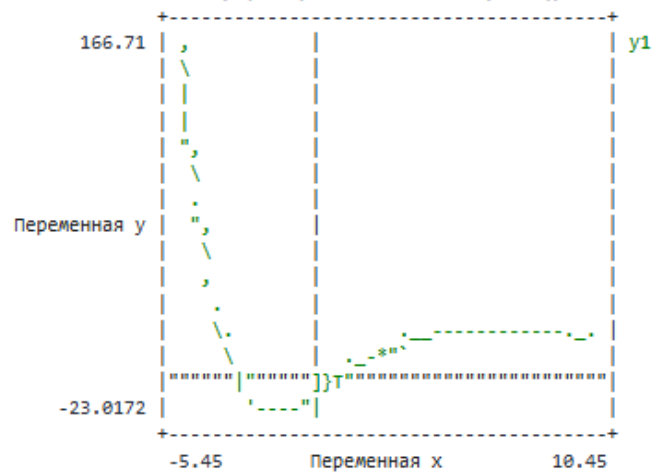
Построен график  $f(x)$  при помощи `ruplot()` с русскими надписями.

---



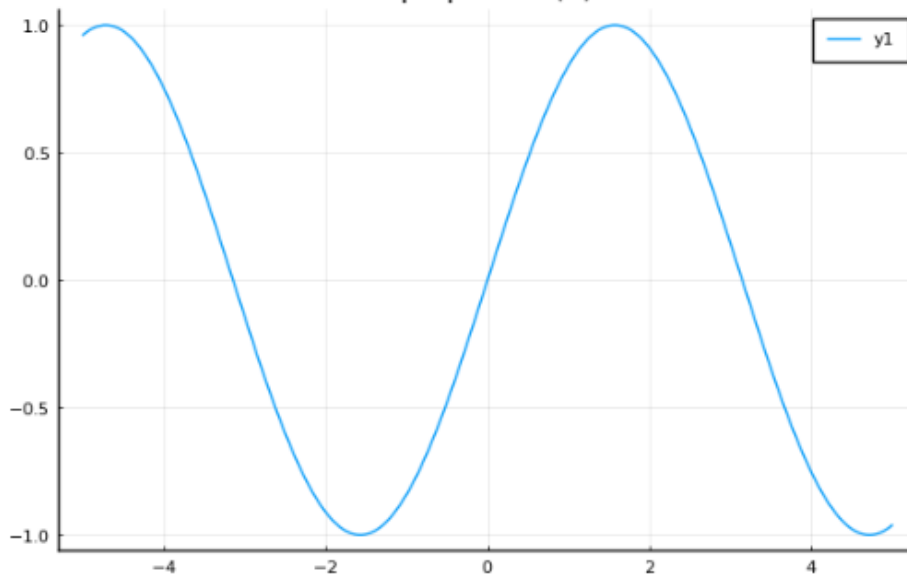
Построен график  $f(x)$  при помощи `plotly()`.

График при помощи `unicodeplots()`



Построен график  $f(x)$  при помощи `unicodeplots()`.

График  $\sin(x)$

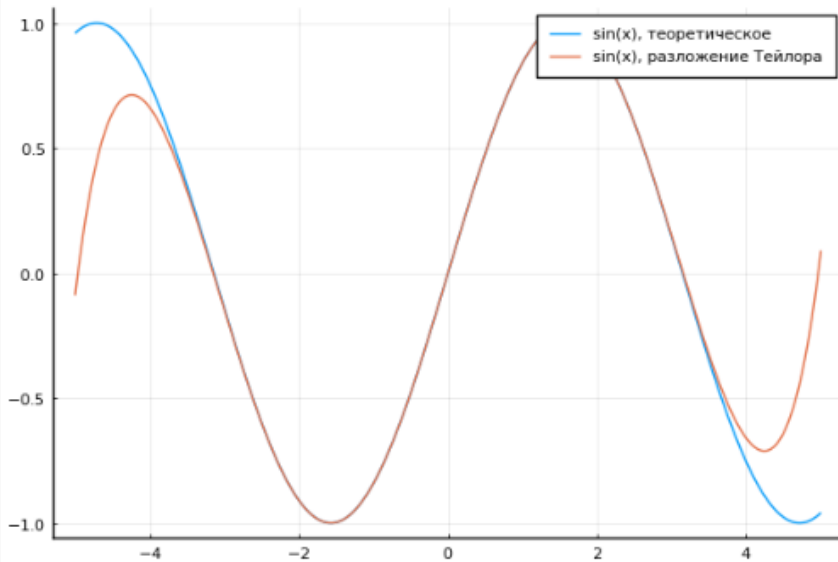


Построен график  $\sin(x)$ .



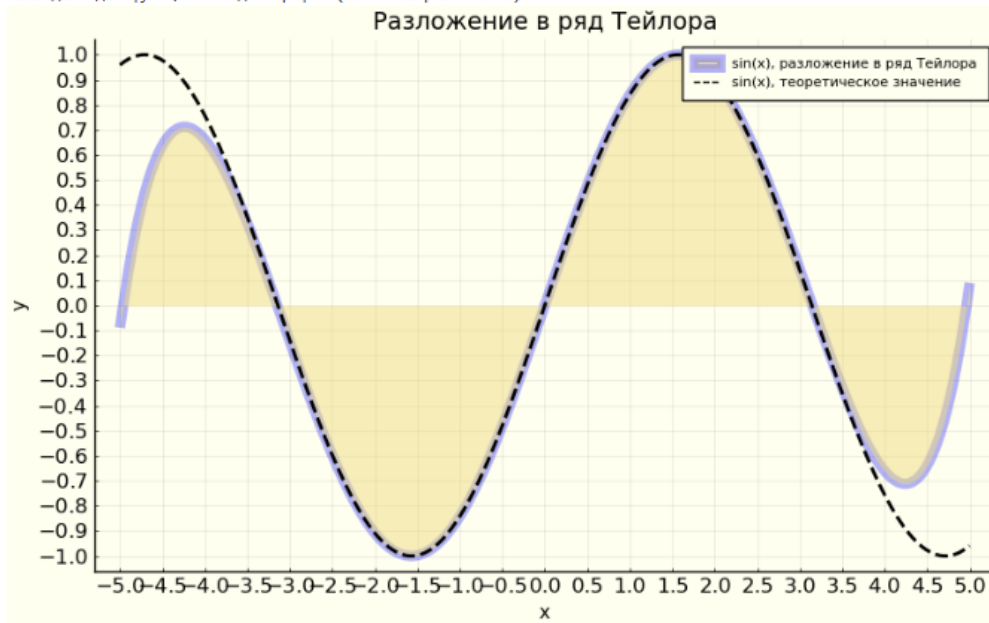


Построен график разложения  $\sin(x)$  в ряд Тейлора.



Выведены две функции на один график ( $\sin$  и ее разложение).

Выведены две функции на один график (sin и ее разложение).

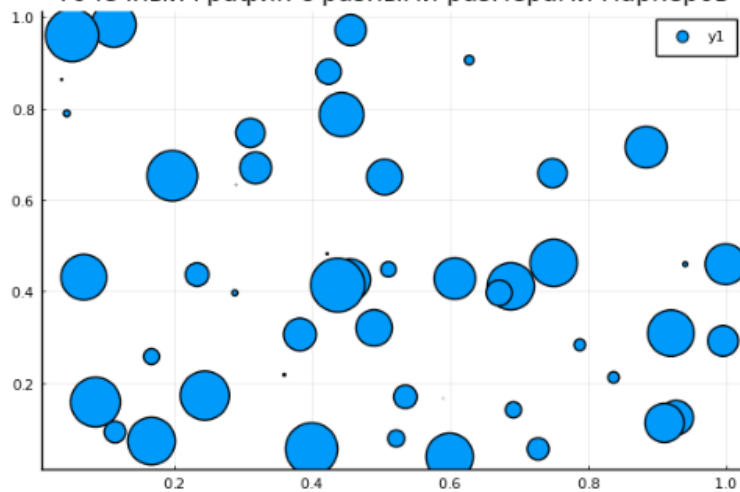


Добавлены опции отображения (цвет, легенда, размер и т.д.).



Построен простой точечный график.

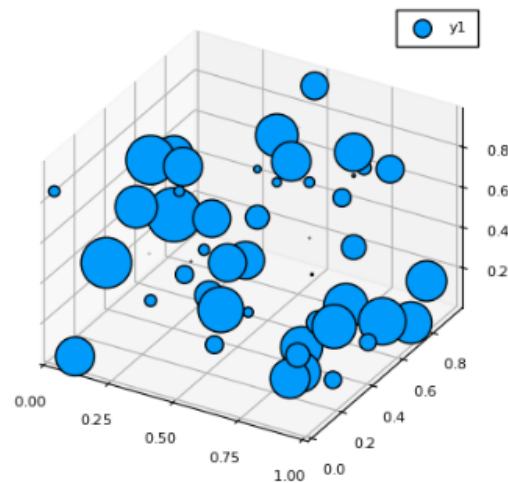
### Точечный график с разными размерами маркеров



```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
```

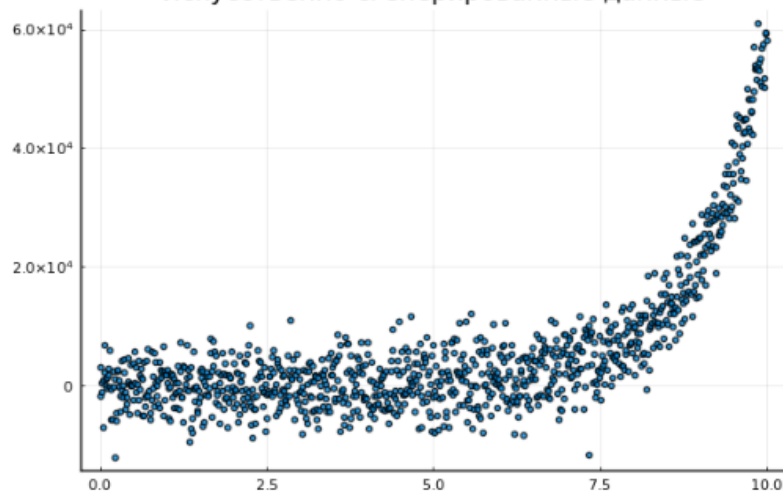
Построен точечный график с разными размерами маркеров.

### 3D точечный график



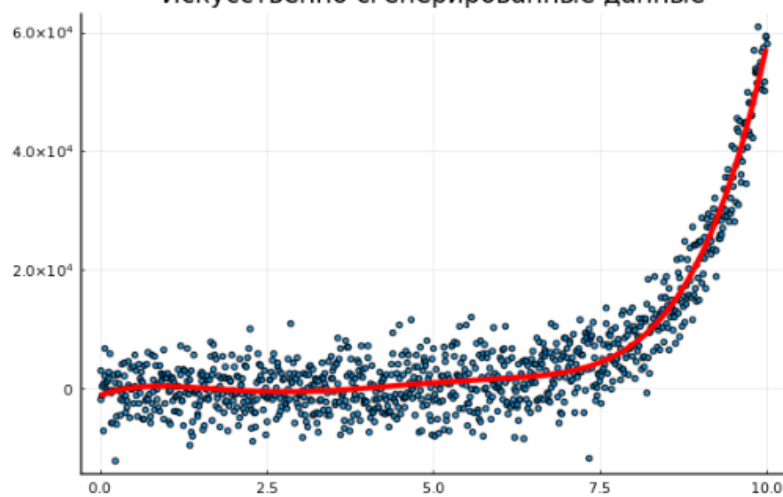
Построен 3D точечный график.

### Искусственно сгенерированные данные



```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
Сгенерированы искусственные данные.
```

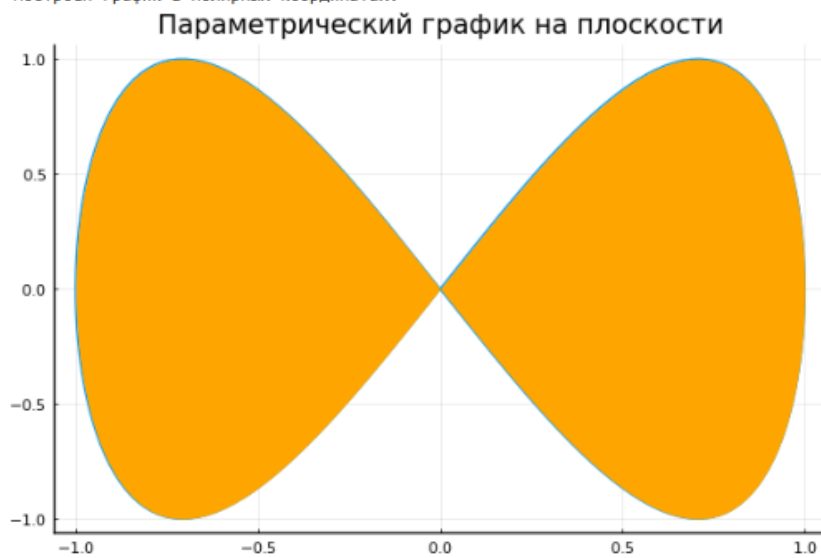
### Искусственно сгенерированные данные



Построен график с двумя осями ординат.



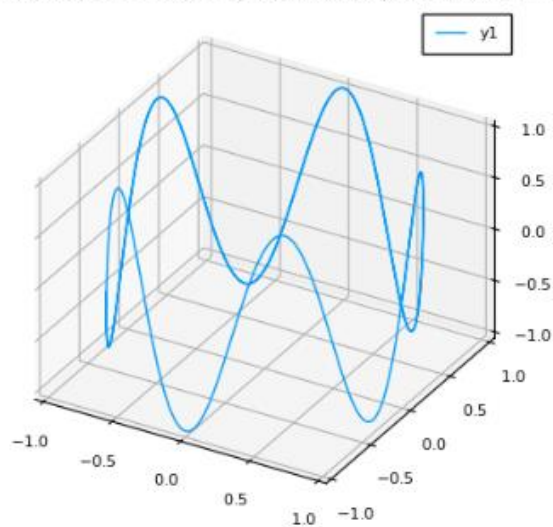
Построен график в полярных координатах.



Построен параметрический график на плоскости.

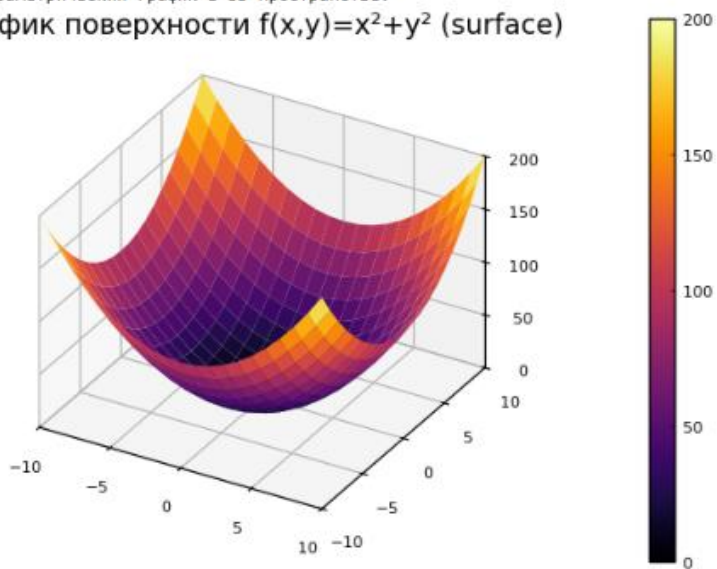
построен параметрический график на плоскости.

### Параметрический график в пространстве



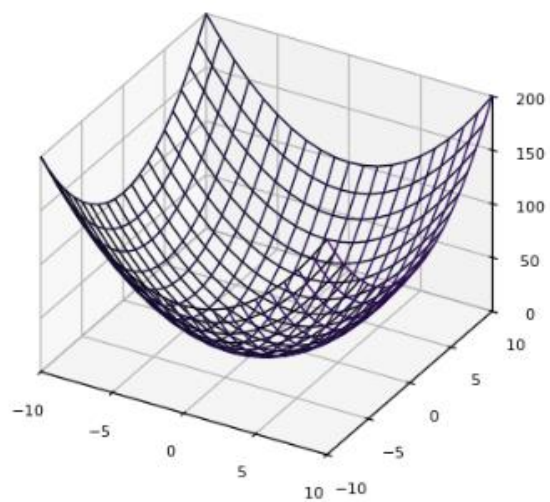
построен параметрический график в 3D пространстве.

### График поверхности $f(x,y)=x^2+y^2$ (surface)



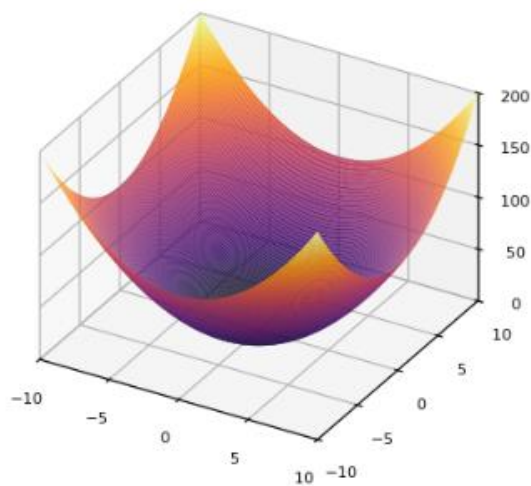
Построен график поверхности (surface).

График поверхности (wireframe)



Построен график поверхности (wireframe).

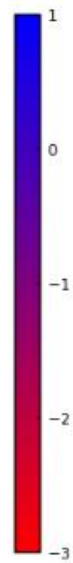
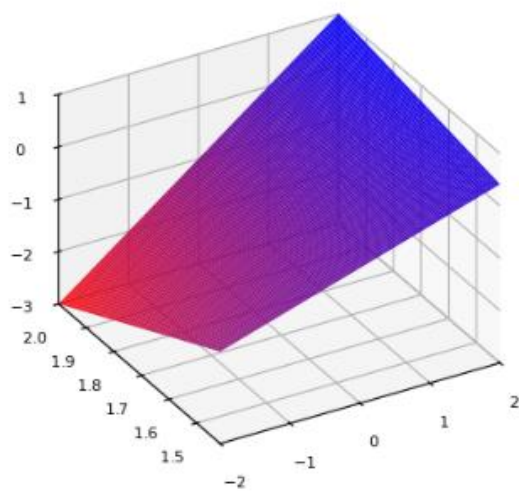
Сглаженный график поверхности



Построен сглаженный график поверхности.

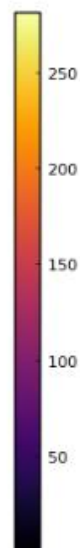
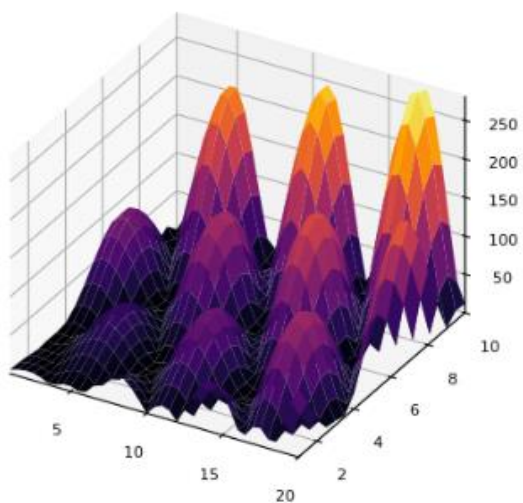
Построен сглаженный график поверхности.

Поверхность с изменённым углом зрения



Построен график поверхности с измененным углом зрения.

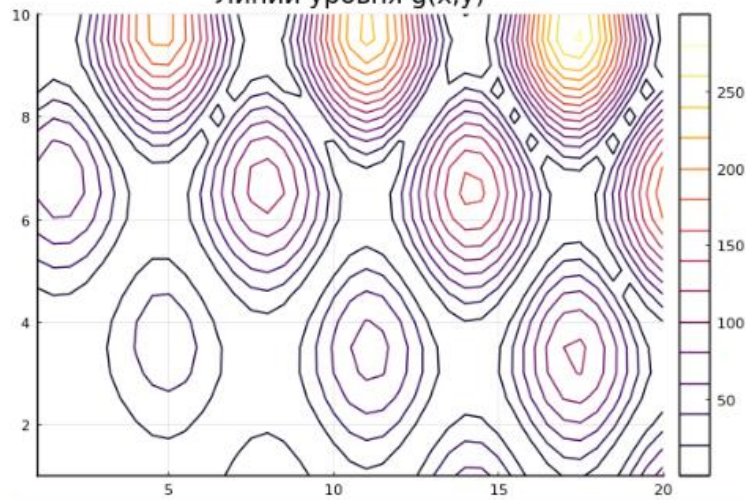
График поверхности  $g(x,y)$





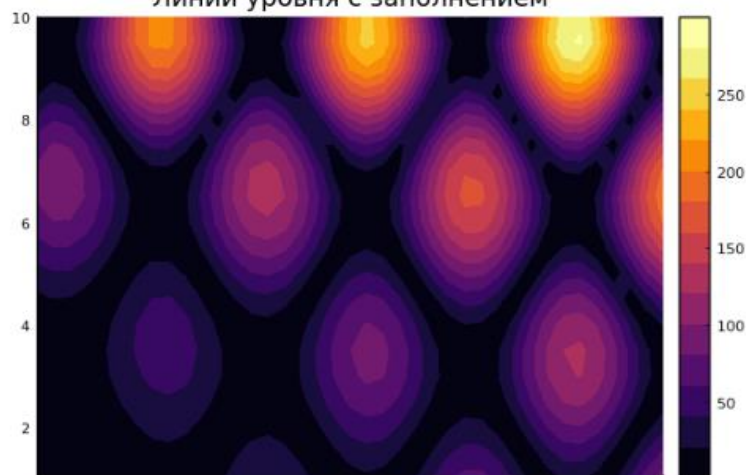
Построен график поверхности  $g(x,y)$ .

Линии уровня  $g(x,y)$



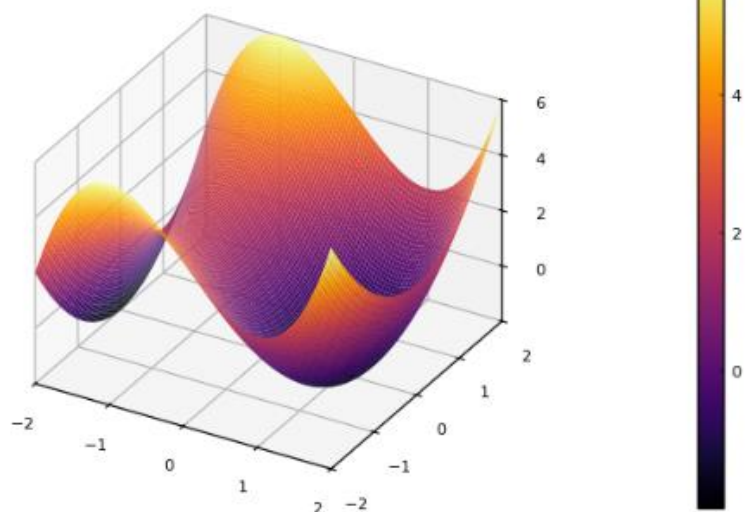
```
sys:1: UserWarning: The following kwargs were not used by contour: 'label'
sys:1: UserWarning: The following kwargs were not used by contour: 'label'
sys:1: UserWarning: The following kwargs were not used by contour: 'label'
sys:1: UserWarning: The following kwargs were not used by contour: 'label'
Построены линии уровня  $g(x,y)$ .
```

Линии уровня с заполнением



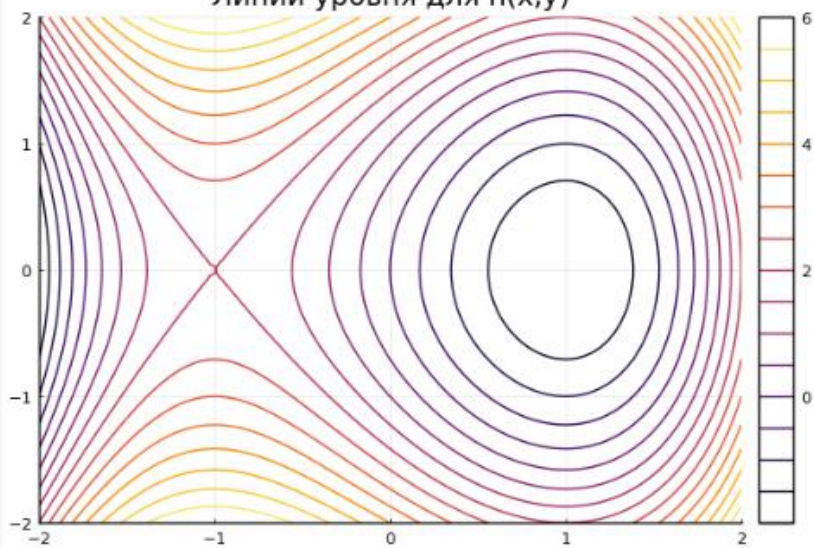
Построены линии уровня с заполнением.

Поверхность  $h(x,y)=x^3-3x+y^2$



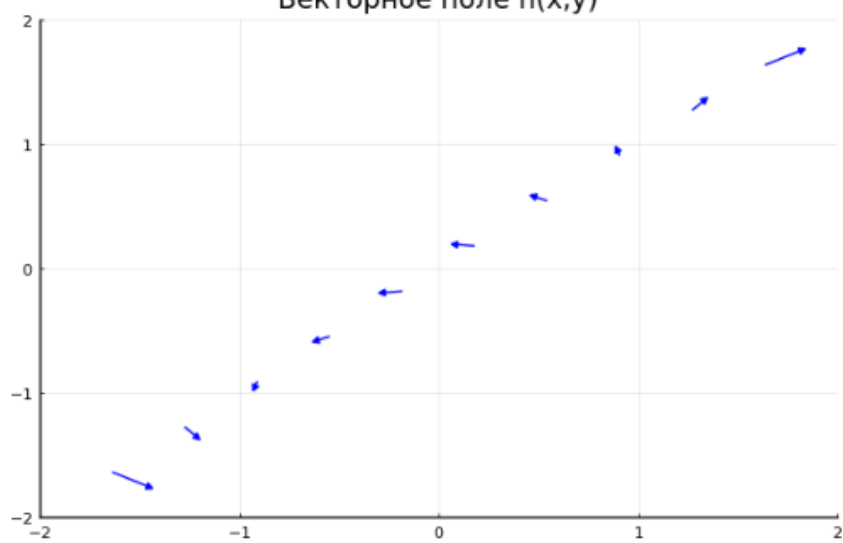
Построен график поверхности  $h(x,y)$ .

Линии уровня для  $h(x,y)$

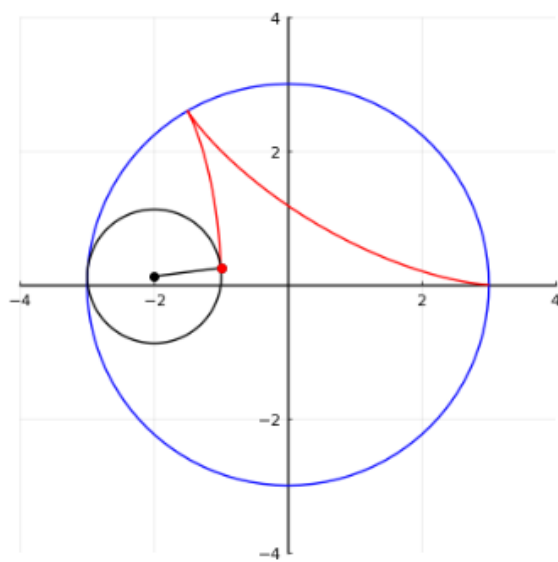


Построены линии уровня для  $h(x,y)$ .

Векторное поле  $h(x,y)$

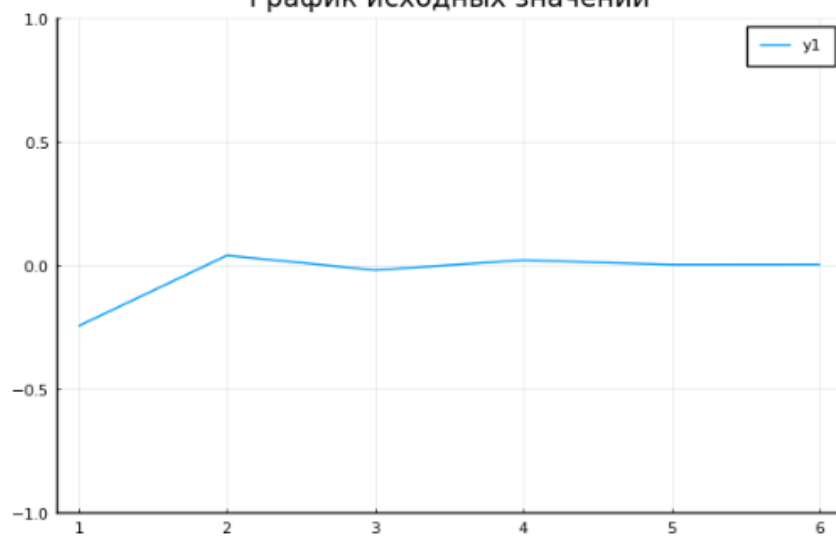


Построено векторное поле  $h(x,y)$ .



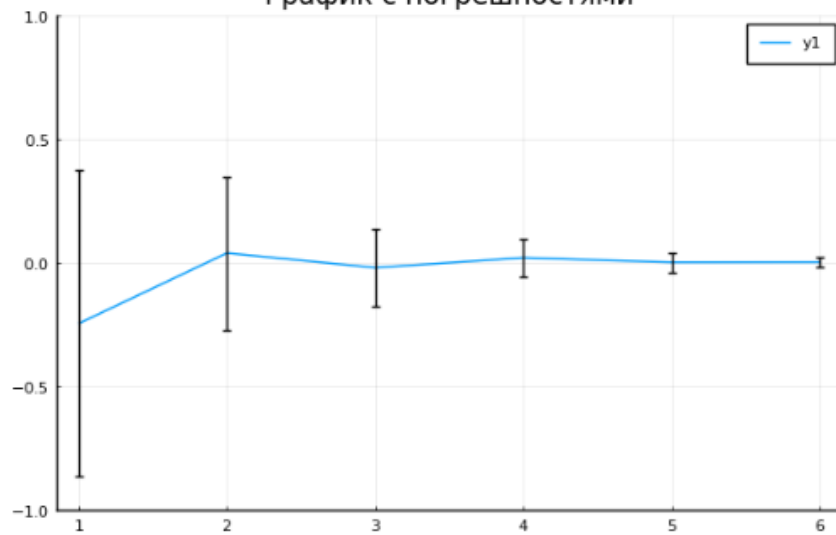
Построена гипоциклоида.

График исходных значений



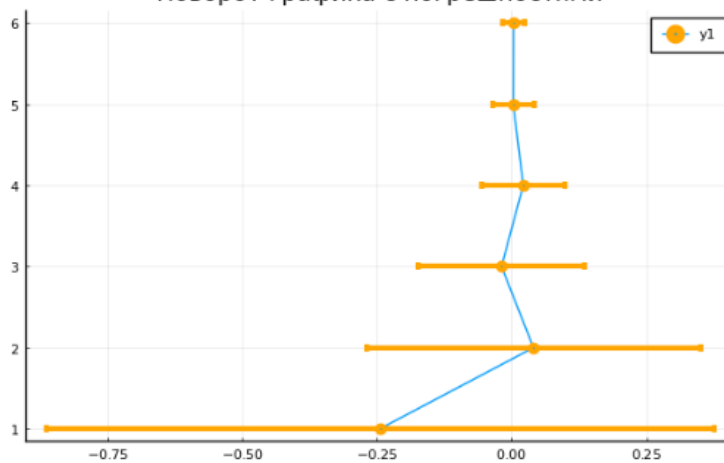
Построен график исходных значений для погрешностей.

График с погрешностями



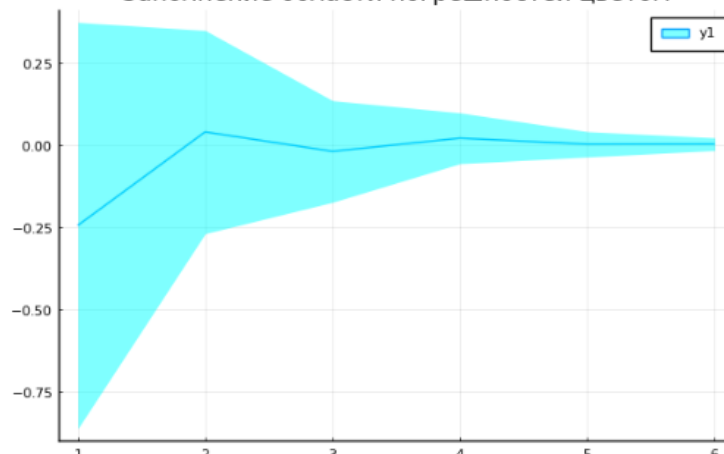
Построен график с погрешностями (errorbars).

### Поворот графика с погрешностями



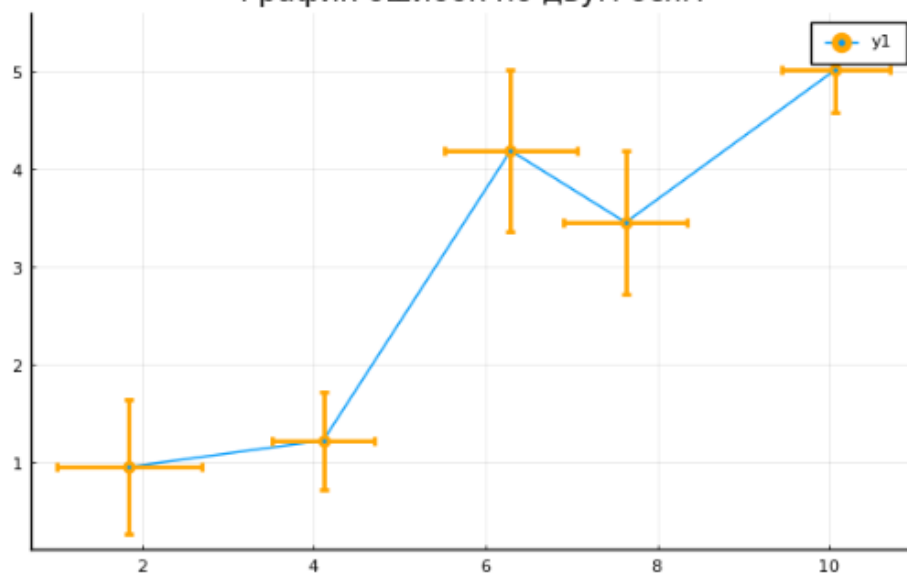
```
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: You passed a edgecolor/edgecolors ((1.0, 0.6470588235294118, 0.0, 1.0)) for an unfilled ma
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: You passed a edgecolor/edgecolors ((1.0, 0.6470588235294118, 0.0, 1.0)) for an unfilled ma
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: You passed a edgecolor/edgecolors ((1.0, 0.6470588235294118, 0.0, 1.0)) for an unfilled ma
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored
sys:1: UserWarning: You passed a edgecolor/edgecolors ((1.0, 0.6470588235294118, 0.0, 1.0)) for an unfilled ma
Построен график с погрешностями, повернутый.
```

### Заполнение области погрешностей цветом



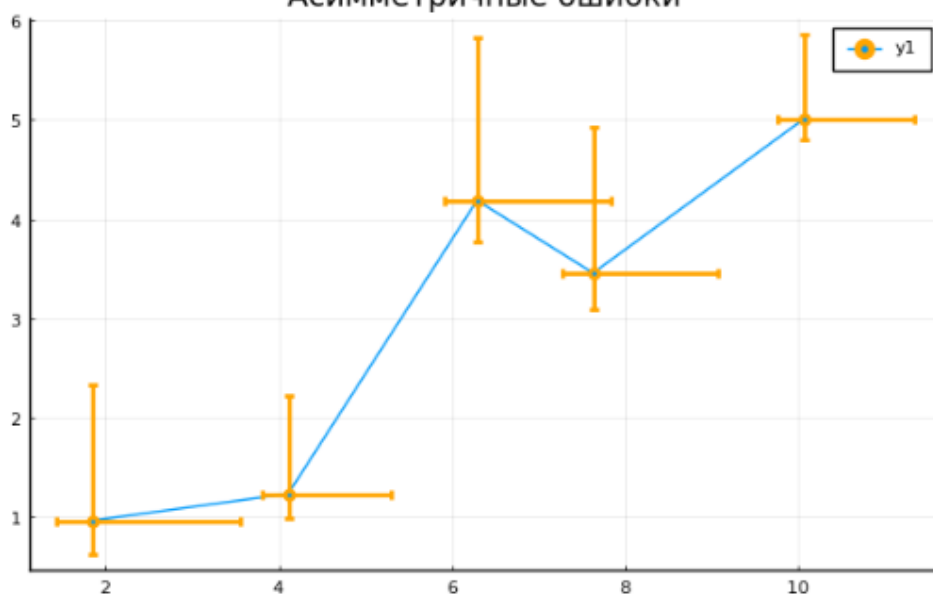
область погрешностей заполнена цветом.

График ошибок по двум осям

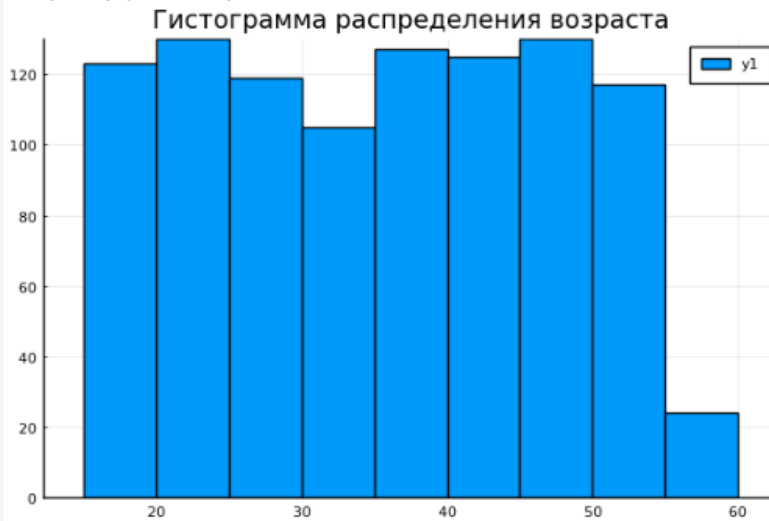


Этот код построит разрыв в `edgescolor` (`edgescolor = ((1,0), 0.167/0.333233234116, 0.16, 1.16)`) для `plt.plot`.  
Построен график ошибок по двум осям.

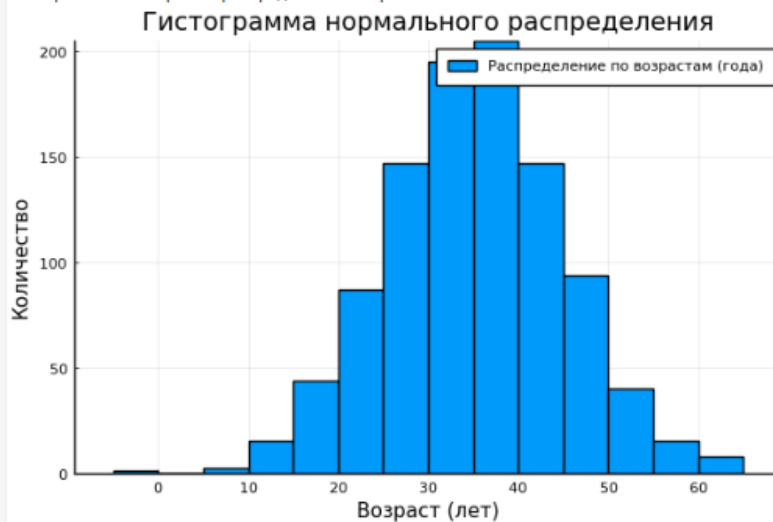
Асимметричные ошибки



Построен график асимметричных ошибок.



sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored  
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored  
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored  
sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored  
Построена гистограмма распределения возраста.

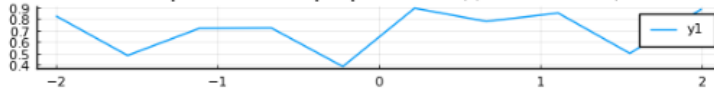




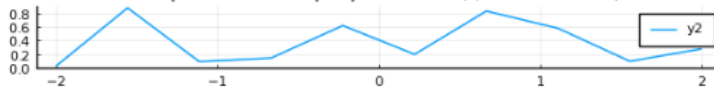
Построена гистограмма распределения по трем нормальным законам.



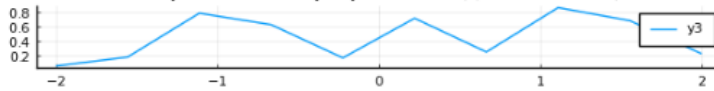
Серия из 4-х графиков в один столбец



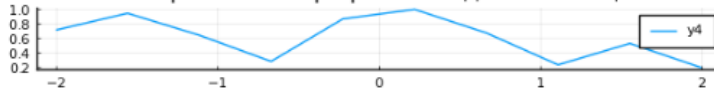
Серия из 4-х графиков в один столбец



Серия из 4-х графиков в один столбец

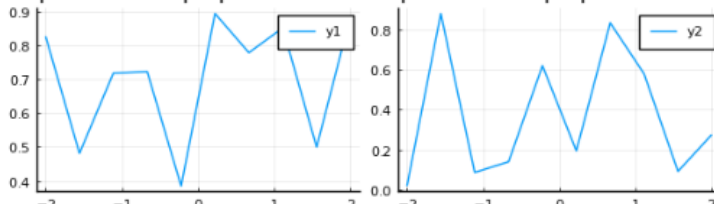


Серия из 4-х графиков в один столбец

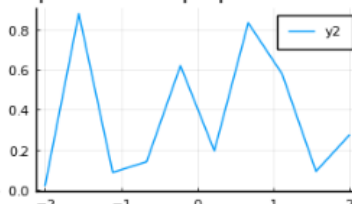


Построена серия из 4-х графиков в один столбец.

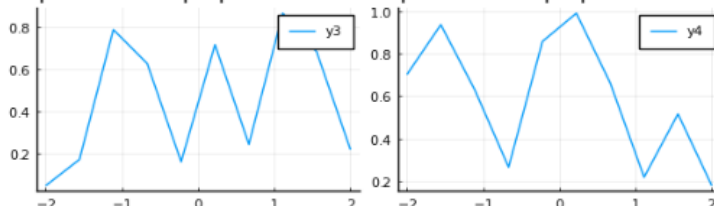
Серия из 4-х графиков в сетке



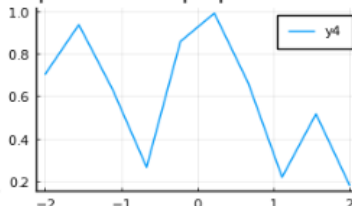
Серия из 4-х графиков в сетке



Серия из 4-х графиков в сетке

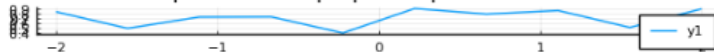


Серия из 4-х графиков в сетке

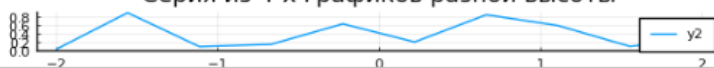


Построена серия из 4-х графиков в сетке.

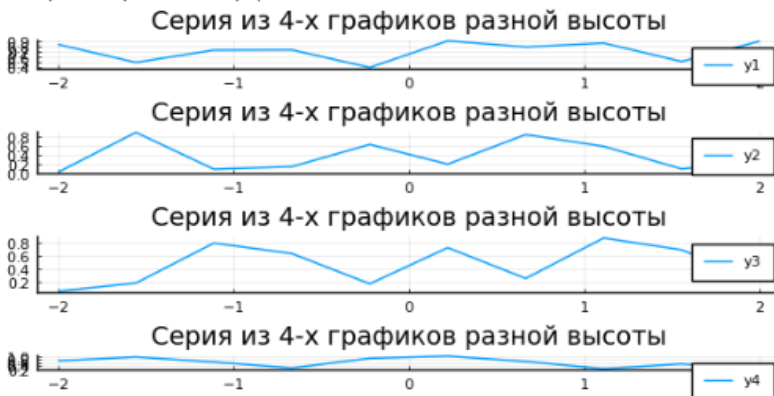
Серия из 4-х графиков разной высоты



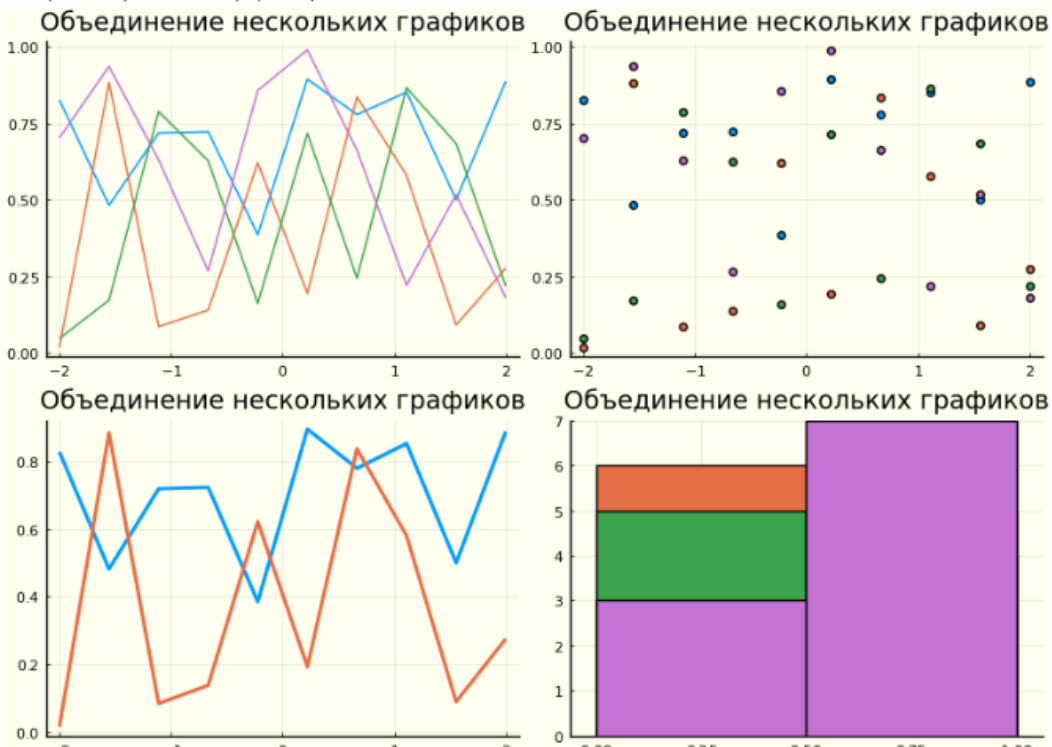
Серия из 4-х графиков разной высоты



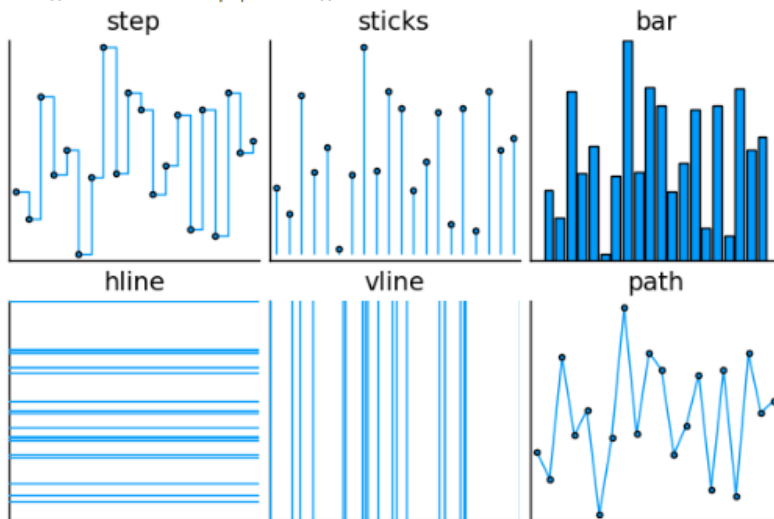
Построена серия из 4-х графиков в сетке.



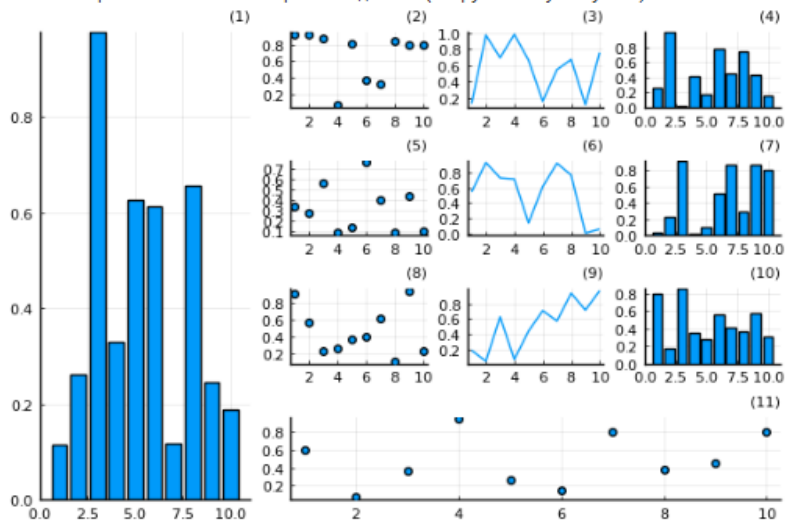
Построена серия из 4-х графиков разной высоты.



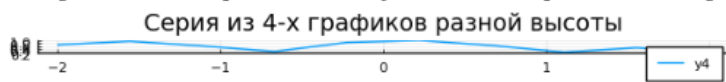
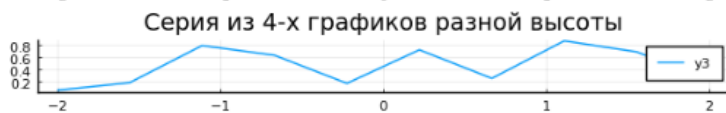
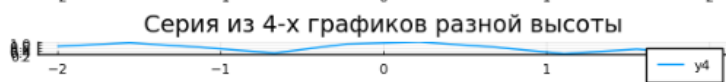
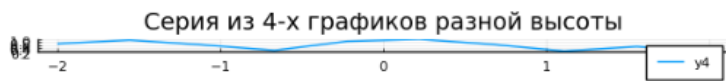
Объединены несколько графиков в одной сетке.

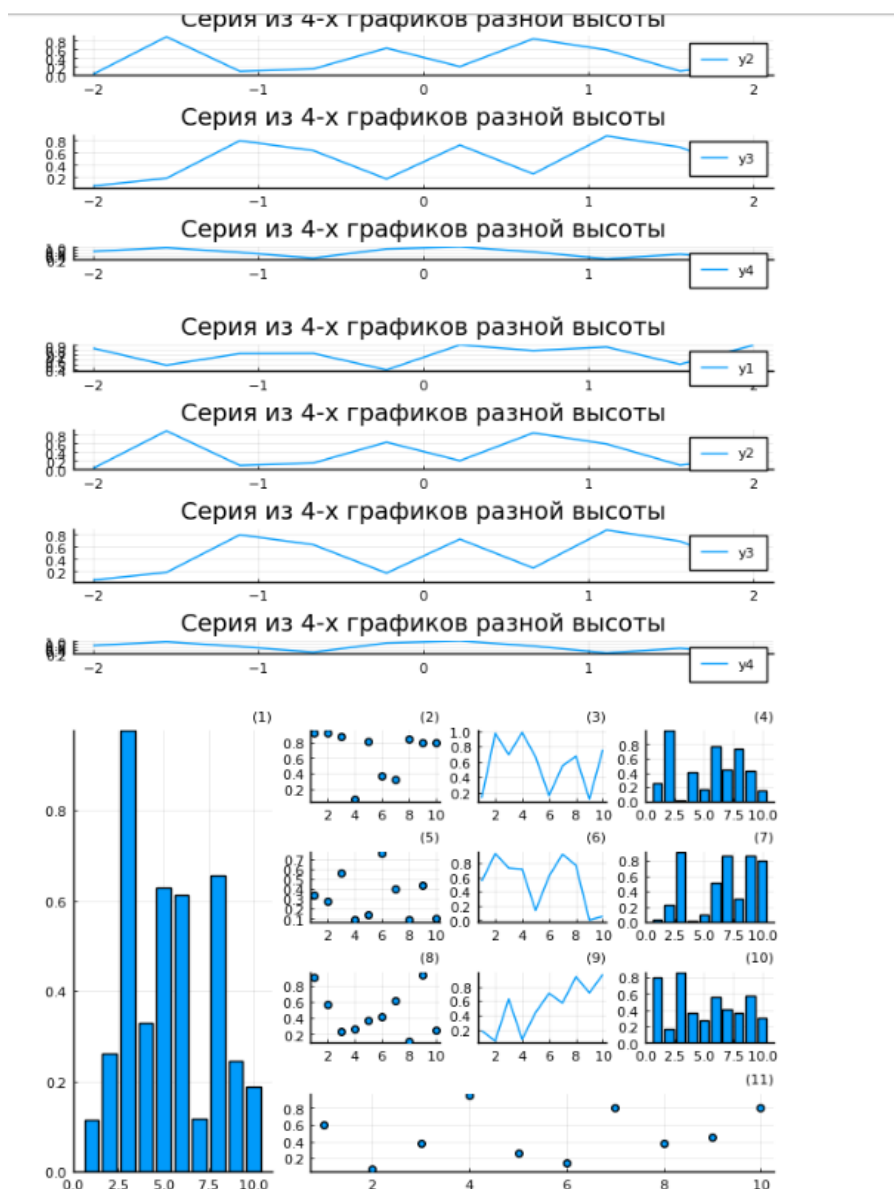


sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored  
 sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored  
 sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored  
 sys:1: UserWarning: No data for colormapping provided via 'c'. Parameters 'vmin', 'vmax' will be ignored  
 Показаны различные стили отображения данных (step, sticks, bar, ...).



Все задания успешно выполнены!





2. Выполните задания для самостоятельной работы (раздел 5.4).

Код:

```

using Pkg
Pkg.add("Plots")
Pkg.add("PyPlot")
Pkg.add("Plotly")
Pkg.add("UnicodePlots")
Pkg.add("Distributions")
Pkg.add("Statistics")
Pkg.add("FFMPEG")
Pkg.add("Measures")

using Plots
using Statistics
using Distributions
using Measures

println("Начинаем выполнение заданий из раздела 5.4...")

#####
# 1. Построить все возможные типы графиков функции  $y=\sin(x)$ ,  $x=0..2\pi$  в одном окне
#####
println("Задание 1...")
x = range(0, 2π, length=100)
y_sin = sin.(x)

p_task1 = plot(layout=(2,3), title="Различные типы графиков sin(x)")
plot!(p_task1, x, y_sin, seriestype=:line, title="line", subplot=1)
plot!(p_task1, x, y_sin, seriestype=:scatter, title="scatter", subplot=2)
histogram!(p_task1, y_sin, title="histogram", subplot=3)
plot!(p_task1, x, y_sin, seriestype=:sticks, title="sticks", subplot=4)
plot!(p_task1, x, y_sin, seriestype=:step, title="step", subplot=5)
bar!(p_task1, x, abs.(y_sin), title="bar(|sin(x)|)", subplot=6)
display(p_task1)

#####
# 2. Построить графики  $y=\sin(x)$ ,  $x=0..2\pi$  с различными типами линий, все в одном окне
#####
println("Задание 2...")
line_styles = [:solid, :dash, :dot, :dashdot, :dashdotdot]
p_task2 = plot(title="sin(x) с разными типами линий", xlabel="x", ylabel="sin(x)")
for (i, ls) in enumerate(line_styles)
    plot!(p_task2, x, y_sin .+ 0.1*(i-1), line=(ls, 2), label=string(ls))
end
display(p_task2)

#####
# 3. Построить график  $y=\pi x^2 \ln(x)$ . Настроить оси, цвета и шрифт
#####
println("Задание 3...")
x_task3 = range(0.1, 2, length=100)
y_task3 = π.*x_task3.^2.*log.(x_task3)
p_task3 = plot(x_task3, y_task3,
    title="y = πx² ln(x)",
    xlabel="x", ylabel="y",

```

```

# 5. два графика на одном и с двумя осями ординат
#####
println("Задание 5...")
x_task5 = 3:0.1:6
y1_task5 = pi.*x_task5
y2_task5 = exp.(x_task5).*cos.(x_task5)

# оба графика на одном рисунке
p_task5a = plot(x_task5, y1_task5, label="πx", xlabel="x", ylabel="y",
    title="Два графика на одном рисунке",
    linecolor=:blue, grid=true)
plot!(p_task5a, x_task5, y2_task5, label="exp(x)*cos(x)", linecolor=:red)
display(p_task5a)
println("Недостаток: разный масштаб по y, сложно сравнивать.")

# с двумя осями ординат
p_task5b = plot(x_task5, y1_task5, label="πx", xlabel="x", ylabel="y1",
    title="Две оси ординат",
    linecolor=:blue, grid=true)
plot!(twinx(), x_task5, y2_task5, label="exp(x)*cos(x)", linecolor=:red, ylabel="y2")
display(p_task5b)

#####
# 6. Экспериментальные данные с ошибками измерений
#####
println("Задание 6...")
n = 20
x_exp = range(1,20,length=n)
y_exp_true = log.(x_exp)
y_exp = y_exp_true .+ 0.2*randn(n)
errs = 0.2*ones(n)
p_task6 = plot(x_exp, y_exp, yerr=errs, seriestype=:scatter,
    xlabel="x", ylabel="измеренное значение",
    title="Экспериментальные данные с ошибками")
plot!(p_task6, x_exp, y_exp_true, label="Модель log(x)", linecolor=:red)
display(p_task6)

#####
# 7. Точечный график случайных данных
#####
println("Задание 7...")
n = 50
x_rand = rand(n)
y_rand = rand(n)
p_task7 = scatter(x_rand, y_rand,
    xlabel="Random X",
    ylabel="Random Y",
    title="Случайные данные",
    label="Data points",
    legend=:topright)
display(p_task7)

#####
# 8. 3D точечный график случайных данных
#####

```

```
#####
# 8. 3D точечный график случайных данных
#####
println("Задание 8...")
n = 50
x_3d = rand(n)
y_3d = rand(n)
z_3d = rand(n)
p_task8 = scatter(x_3d, y_3d, z_3d,
    xlabel="x",
    ylabel="y",
    zlabel="z",
    title="3D случайные данные",
    label="3D Points",
    legend=:top)
display(p_task8)

#####
# 9. Анимация синусоиды
#####
println("Задание 9...")
x_anim = range(0,2π,length=100)
anim = @animate for i in 1:20
    plot(x_anim, sin.(x_anim .+ 0.1*i),
        title="Анимация синусоиды (кадр $i)",
        xlabel="x", ylabel="sin(x+0.1*i)")
end
gif(anim, "sinus_anim.gif", fps=10)
println("Анимация синусоиды сохранена в sinus_anim.gif")

#####
# 10. Анимированная гипоциклоида
#####
println("Задание 10...")

function hypocycloid(r, k; frames=50)
    θ = range(0, stop=2π, length=300)
    anim = @animate for i in 1:frames
        t = θ[1:i]
        x = r*(k-1)*cos.(t) .+ r*cos.((k-1).*t)
        y = r*(k-1)*sin.(t) .- r*sin.((k-1).*t)
        plot(x,y, aspect_ratio=1, xlims=(-4,4), ylims=(-4,4),
            title="Гипоциклоида k=$k", legend=false)
    end
    gif(anim, "hypocycloid_k$(k).gif", fps=10)
end

r=1
k_ints = [2,3]
for k_ in k_ints
    println("Анимация гипоциклоиды для k=$k_")
    hypocycloid(r, k_, frames=50)
end
```



```

r=1
k_ints = [2,3]
for k_ in k_ints
    println("Анимация гипоциклоиды для k=$k_")
    hypocycloid(r, k_, frames=50)
end

k_rat = [2.5,1.5]
for k_ in k_rat
    println("Анимация гипоциклоиды для k=$k_")
    hypocycloid(r, k_, frames=50)
end

println("Гипоциклоиды анимированы.")

#####
# 11. Анимированная эпициклоида
#####
println("Задание 11...")

function epicycloid(r, k; frames=50)
    θ = range(0, stop=2π, length=300)
    anim = @animate for i in 1:frames
        t = θ[1:i]
        x = r*(k+1)*cos.(t) .- r*cos.((k+1).*t)
        y = r*(k+1)*sin.(t) .- r*sin.((k+1).*t)
        plot(x,y, aspect_ratio=1, xlims=(-4,4), ylims=(-4,4),
            title="Эпициклоида k=$(k)", legend=false)
    end
    gif(anim, "epicycloid_k$(k).gif", fps=10)
end

for k_ in [2,3]
    println("Анимация эпициклоиды для k=$k_")
    epicycloid(1, k_, frames=50)
end

for k_ in [1.5,2.5]
    println("Анимация эпициклоиды для k=$k_")
    epicycloid(1, k_, frames=50)
end

println("Эпициклоиды анимированы.")

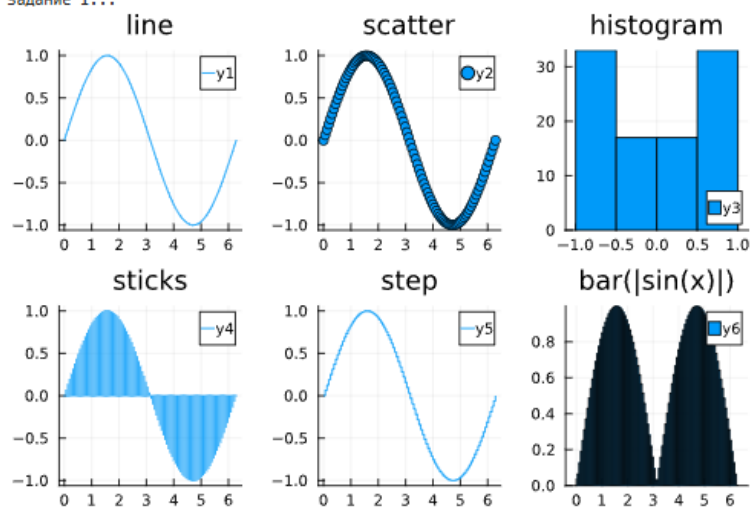
println("Все задания успешно выполнены!")

```

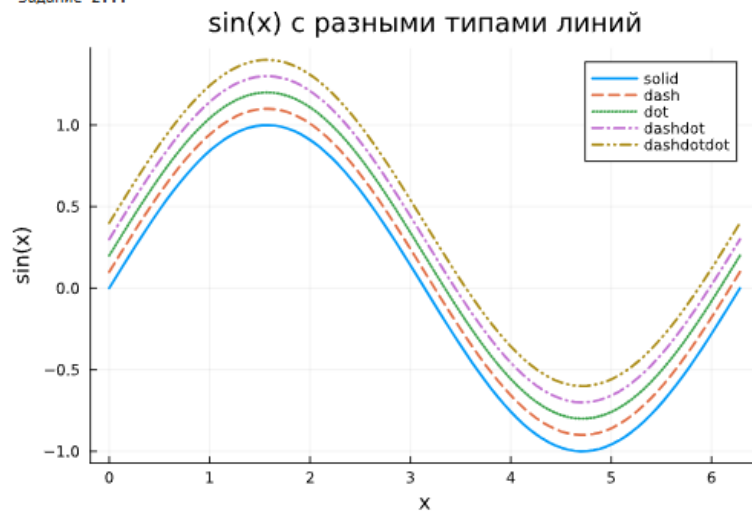
Результаты:

Начинаем выполнение заданий из раздела 5.4...

Задание 1...



Задание 2...

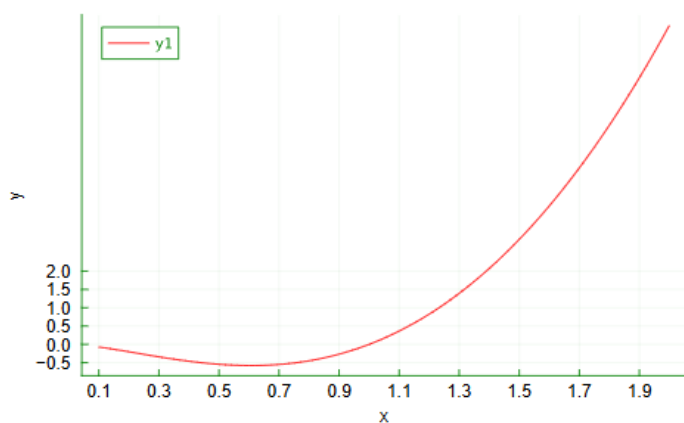


Задание 3...

$$y = \pi x^2 \ln(x)$$

Задание 3...

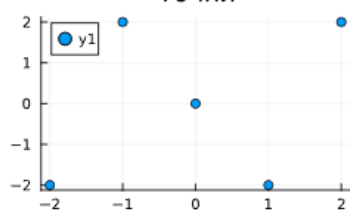
$$y = \pi x^2 \ln(x)$$



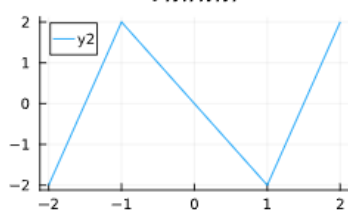
Задание 4...

GKS: could not find font Arial.ttf

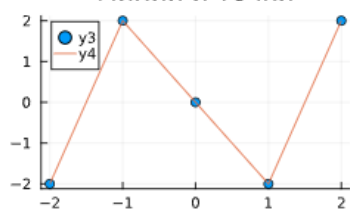
Точки



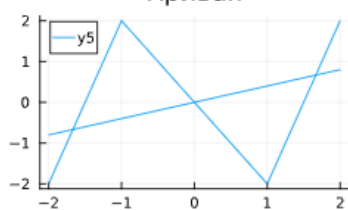
Линии



Линии и точки



Кривая



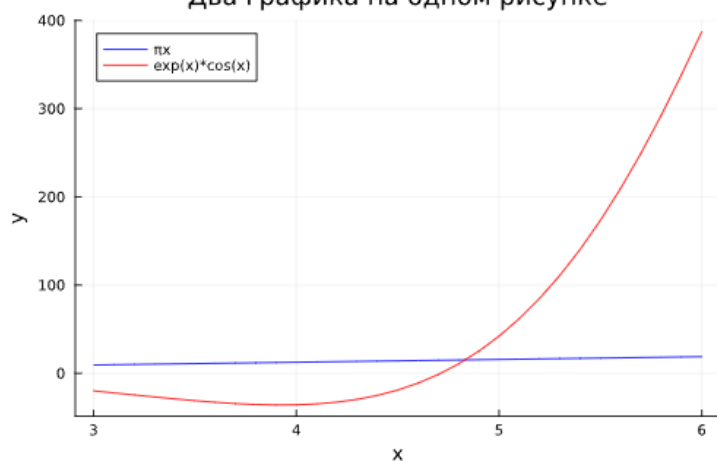
Задание 5...

Два графика на одном рисунке



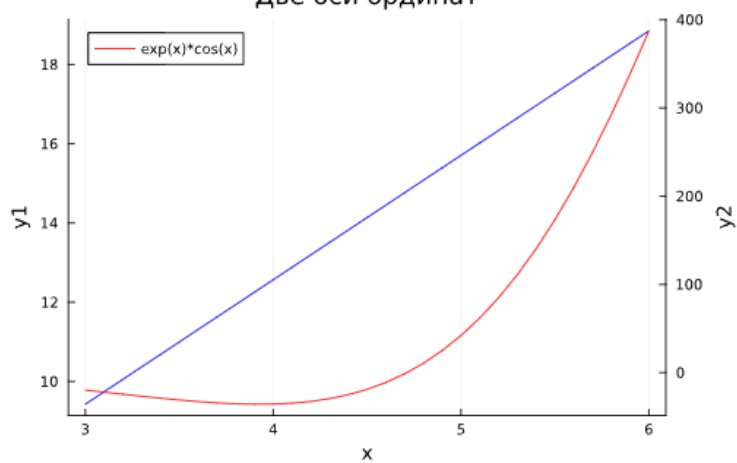
Задание 5...

### Два графика на одном рисунке



Недостаток: разный масштаб по  $y$ , сложно сравнивать.

### Две оси ординат

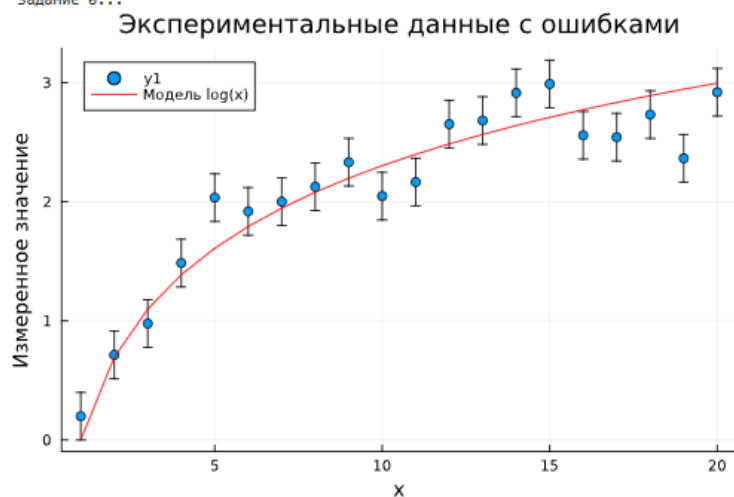


Задание 6...

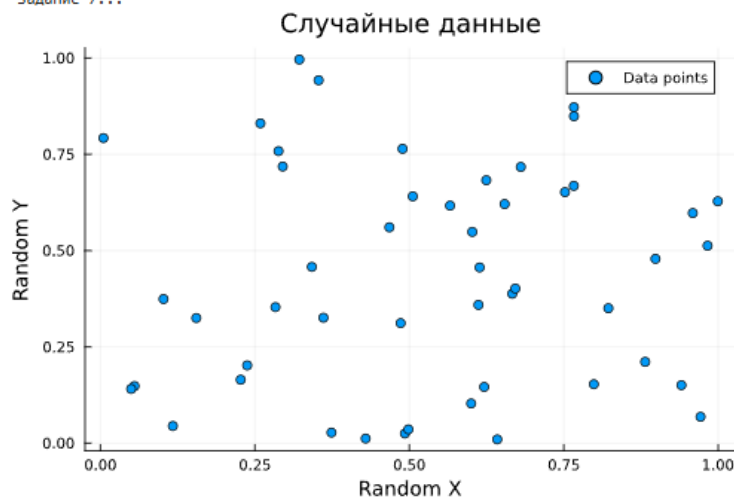
### Экспериментальные данные с ошибками



Задание 6...

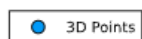


Задание 7...



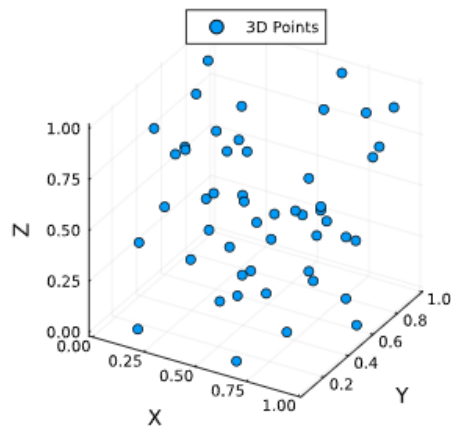
Задание 8...

### 3D случайные данные



Задание 8...

### 3D случайные данные



Задание 9...

Анимация синусоиды сохранена в sinus\_anim.gif

Задание 10...

Анимация гипоциклоиды для k=2

[ Info: Saved animation to C:\Users\Neox\sinus\_anim.gif

Анимация гипоциклоиды для k=3

[ Info: Saved animation to C:\Users\Neox\hypocycloid\_k2.gif

Анимация гипоциклоиды для k=2.5

[ Info: Saved animation to C:\Users\Neox\hypocycloid\_k3.gif

Анимация гипоциклоиды для k=1.5

[ Info: Saved animation to C:\Users\Neox\hypocycloid\_k2.5.gif

Гипоциклоиды анимированы.

Задание 11...

Анимация эпициклоиды для k=2

[ Info: Saved animation to C:\Users\Neox\hypocycloid\_k1.5.gif

Анимация эпициклоиды для k=3

[ Info: Saved animation to C:\Users\Neox\epicycloid\_k2.gif

[ Info: Saved animation to C:\Users\Neox\epicycloid\_k3.gif

Анимация эпициклоиды для k=1.5

Анимация эпициклоиды для k=2.5

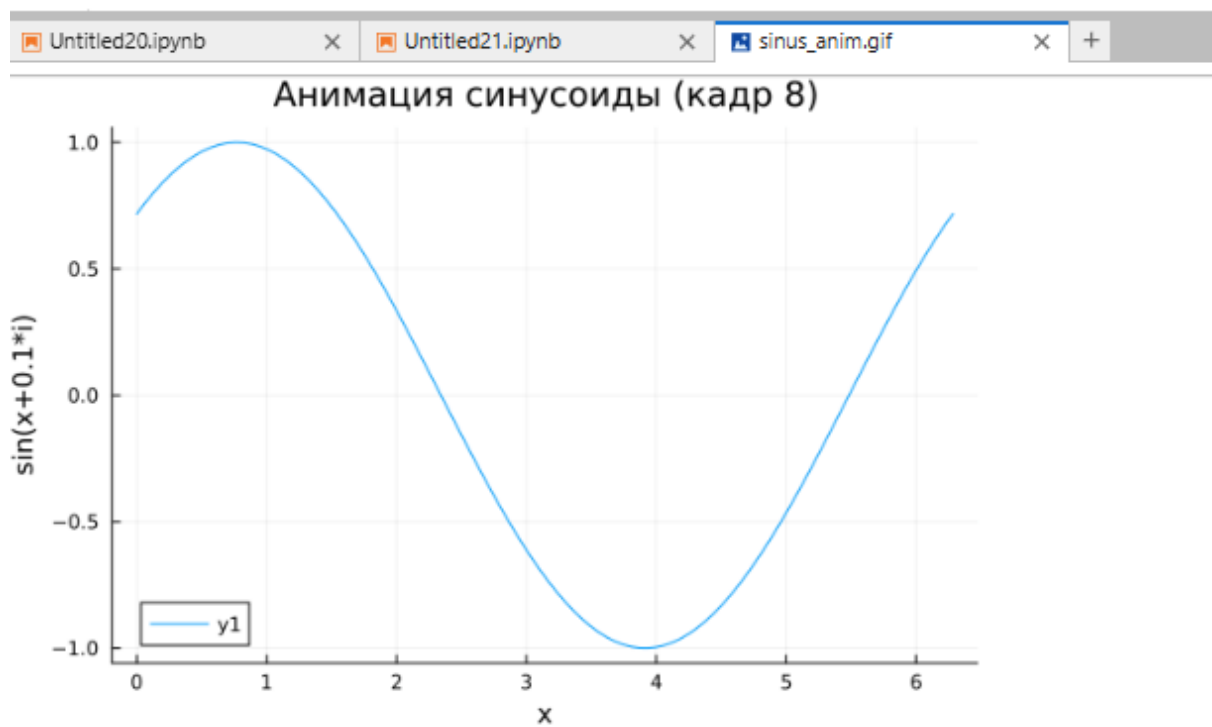
[ Info: Saved animation to C:\Users\Neox\epicycloid\_k1.5.gif

Эпициклоиды анимированы.

Все задания успешно выполнены!

[ Info: Saved animation to C:\Users\Neox\epicycloid\_k2.5.gif

Прим. Gif:



Выводы:

В процессе выполнения работы мною был освоен синтаксис языка Julia для построения графиков