

# **Лабораторная Работа №5**

## Построение графиков

---

Козлов В.П.

Российский университет дружбы народов им. Патриса Лумумбы, Москва, Россия

## Докладчик

---

- Козлов Всеволод Павлович
- НФИбд-02-22
- Российский университет дружбы народов
- [1132226428@pfur.ru]

## **Выполнение лабораторной работы**

---

## Цель работы

---

Освоить синтаксис языка Julia для построения графиков

## Задание

---

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

## Задание 5.2.1

```
[4]: # 5.2.1 Основные пакеты для работы с графиками

[5]: f(x) = (3x^2 + 6x - 9).exp(-0.3x)
x = collect(range(-5, 10, length=151))
y = f(x)

# Построение с gr()
gr()
plot(x, y,
      title="График функции f(x)",
      xlabel="Переменная x",
      ylabel="Переменная y",
      color="blue",
      label="f(x) = (3x^2 + 6x - 9)e^(-0.3x)",
      linewidth=2
)
```

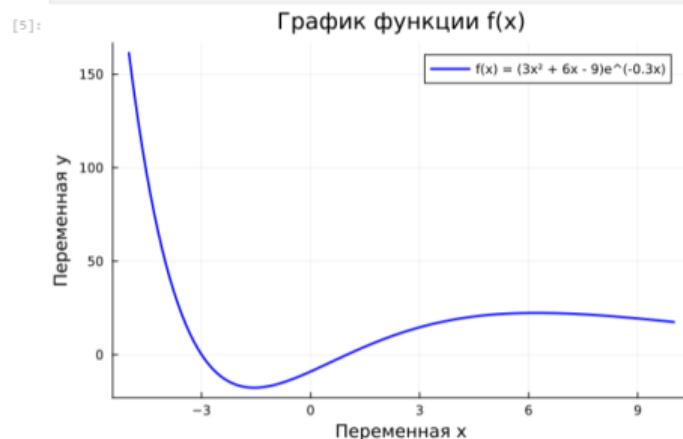
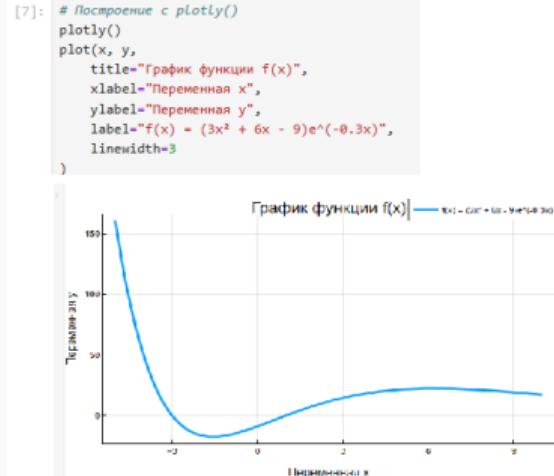


Figure 1: Задание 5.2.1

## Задание 5.2.1



**Figure 2:** Задание 5.2.1

## Задание 5.2.2

```
[12]: # 5.2.2 Сравнить sin(x) с разложением в ряд Тейлора
sin_theor(x) = sin(x)
sin_taylor(x) = sum([(-1)^i * x^(2*i+1) / factorial(2*i+1) for i in 0:4])
x_taylor = range(-2π, 2π, length=100)
plot(x_taylor, sin_theor.(x_taylor), label="sin(x) теор", linewidth=2)
plot!(x_taylor, sin_taylor.(x_taylor), label="ряд Тейлора", linestyle=:dash)
```

[12]:

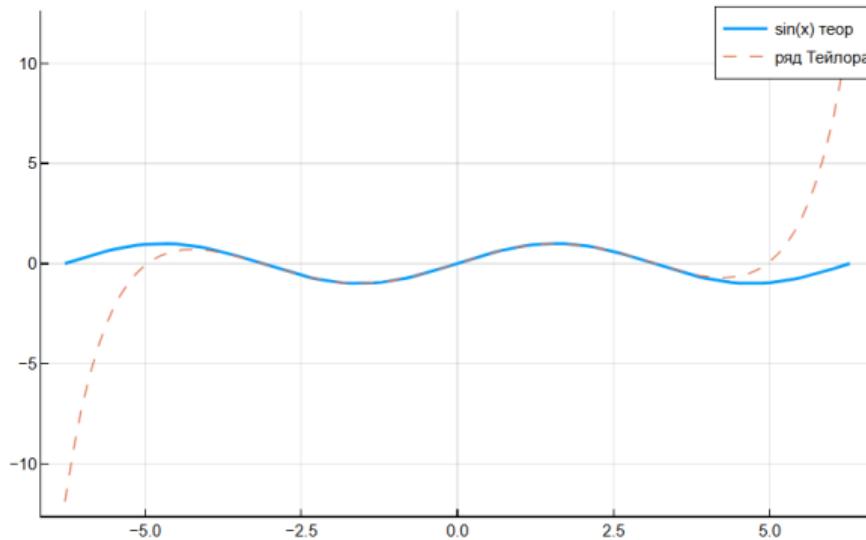


Figure 3: Задание 5.2.2

## Задание 5.2.3

```
[14]: # 5.2.3 Построить точечные графики 2D и 3D
x_scatter = range(1, 10, length=10)
y_scatter = rand(10)
scatter(x_scatter, y_scatter, title="Точечный график", xlabel="X", ylabel="Y")
```

```
[14]:
```

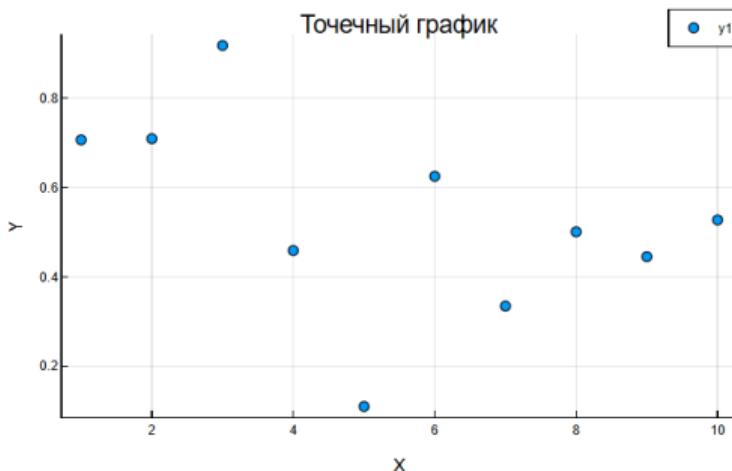
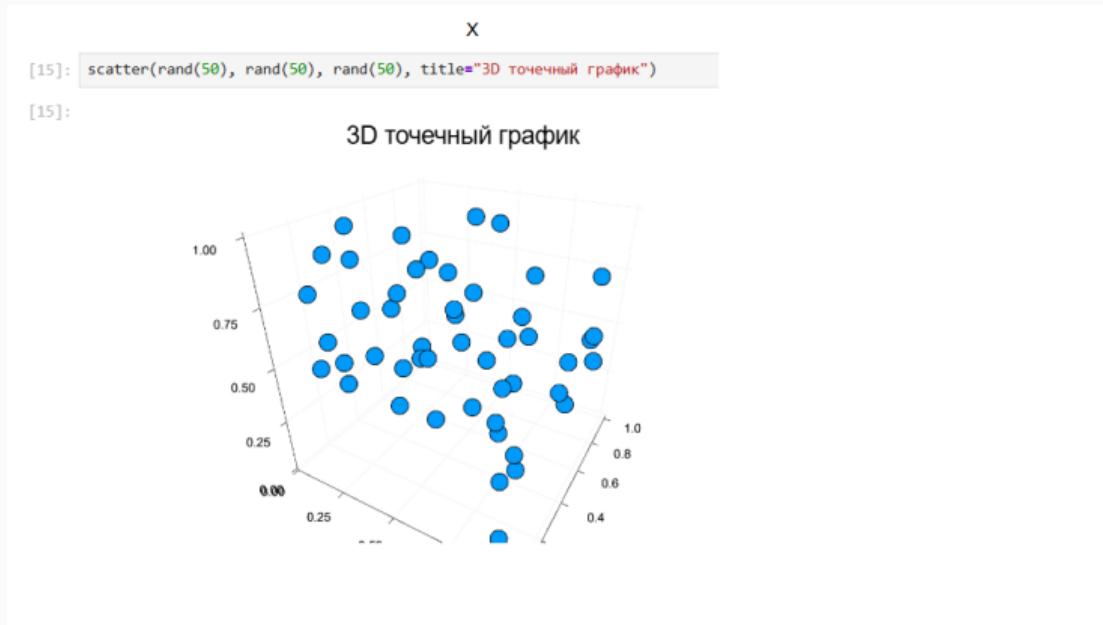


Figure 4: Задание 5.2.3

## Задание 5.2.3



**Figure 5:** Задание 5.2.3

## Задание 5.2.4

```
[16]: # 5.2.4 Аппроксимировать данные полиномом
x_app = collect(0:0.01:9.99)
y_app = exp(0.5 .* x_app) + 100 * randn(1000)
A = [ones(1000) x_app x_app.^2 x_app.^3]
c = A\y_app
f_app = c[1] .+ c[2]*x_app + c[3]*x_app.^2 + c[4]*x_app.^3
scatter(x_app, y_app, markersize=1, alpha=0.3, label="Данные")
plot!(x_app, f_app, linewidth=3, color=:red, label="Полином")
```

[16]:

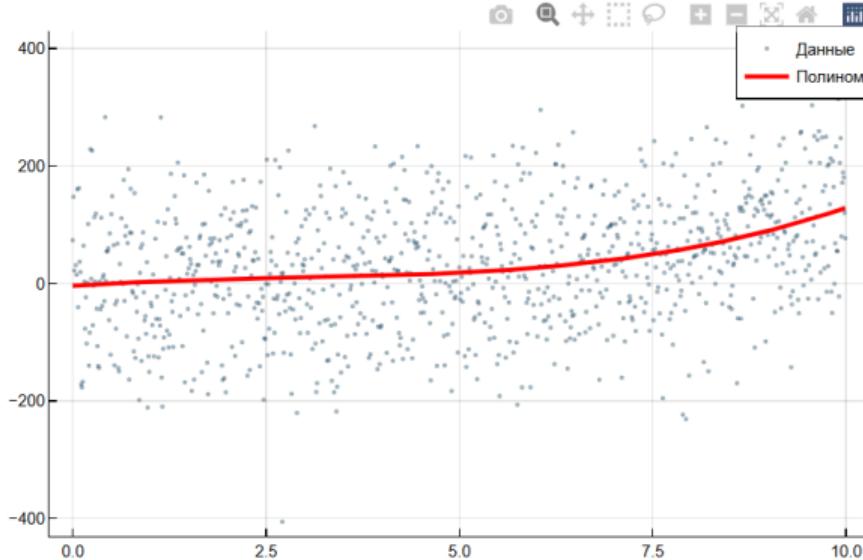


Figure 6: Задание 5.2.4

## Задание 5.2.5

```
[17]: # 5.2.5 Построить график с двумя осями ординат  
plot(randn(100), ylabel="y1", label="Траектория 1", color=:blue)  
plot!(randn(100) .+ 2, label="Траектория 2", color=:red)
```

```
[17]:
```

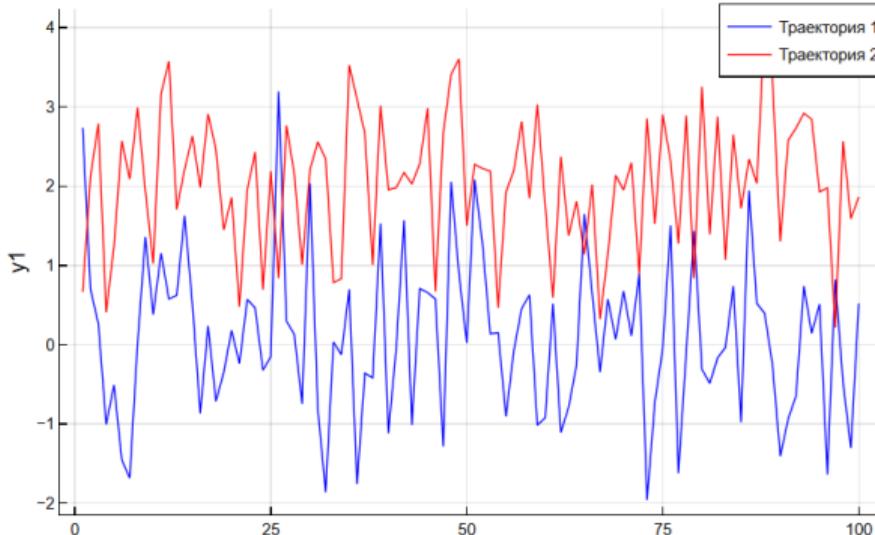
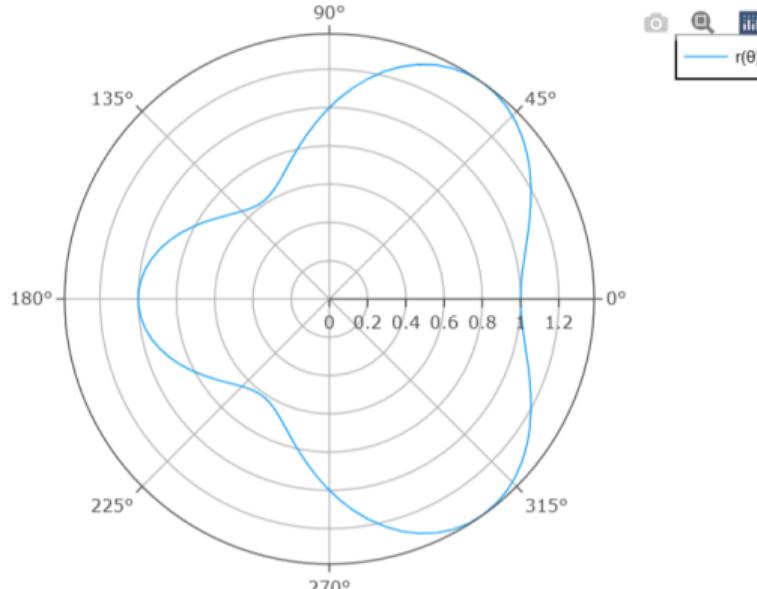


Figure 7: Задание 5.2.5

## Задание 5.2.6

```
[18]: # 5.2.6 Построить график в полярных координатах  
r(θ) = 1 + cos(θ) * sin(θ)^2  
θ_polar = range(0, stop=2π, length=100)  
plot(θ_polar, r_(θ_polar), proj=:polar, lims=(0, 1.5), label="r(θ)")
```

```
[18]:
```



**Figure 8:** Задание 5.2.6

## Задание 5.2.7

```
[19]: # 5.2.7 Построить параметрические кривые на плоскости и в пространстве  
t_plane = range(0, stop=2π, length=100)  
plot(sin.(t_plane), sin.(2 .* t_plane), label="x=sin(t), y=sin(2t)")  
plot(cos.(t_plane), sin.(t_plane), 0.1 .* t_plane, label="Сpirаль")
```

```
[19]:
```

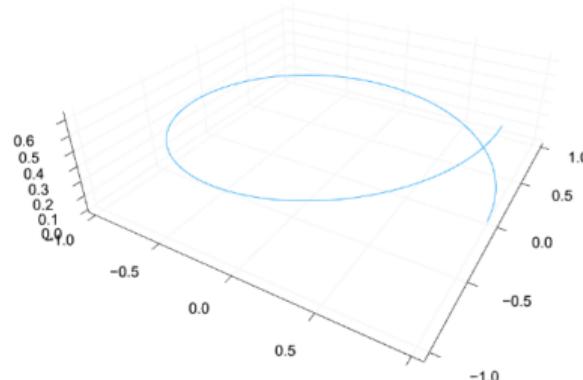


Figure 9: Задание 5.2.7

## Задание 5.2.8

```
[20]: # 5.2.8 Построить поверхности разными способами
f_surf(x,y) = x^2 + y^2
x_surf = -5:0.5:5
y_surf = -5:0.5:5
surface(x_surf, y_surf, f_surf, title="Поверхность")
plot(x_surf, y_surf, f_surf, linetype=:wireframe, title="Каркас")
```

```
[20]:
```

Каркас

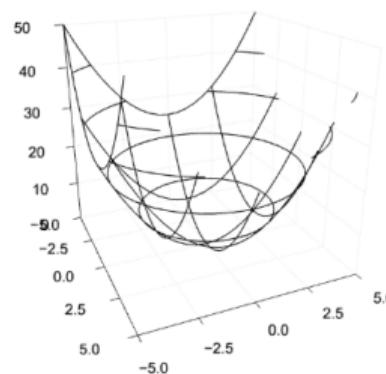


Figure 10: Задание 5.2.8

## Задание 5.2.9

```
[21]: # 5.2.9 Построить линии уровня с заполнением  
g(x, y) = (3x + y^2) * abs(sin(x) + cos(y))  
x_contour = 1:0.5:10  
y_contour = 1:0.5:8  
contour(x_contour, y_contour, g, fill=true, title="Линии уровня")
```

```
[21]:
```

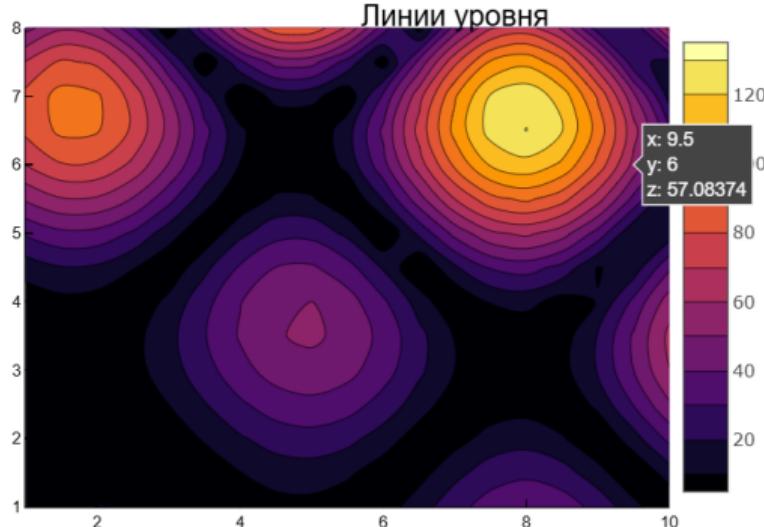


Figure 11: Задание 5.2.9

## Задание 5.2.12

```
[22]: # 5.2.12 Построить график с errorbars
sds = [1, 1/2, 1/4, 1/8, 1/16, 1/32]
y_err = [mean(sd * randn(10)) for sd in sds]
errs = 1.96 * sds / sqrt(10)
plot(y_err, err=errs, label="Данные с ошибками", marker=stroke(3, :orange))
```

[22]:

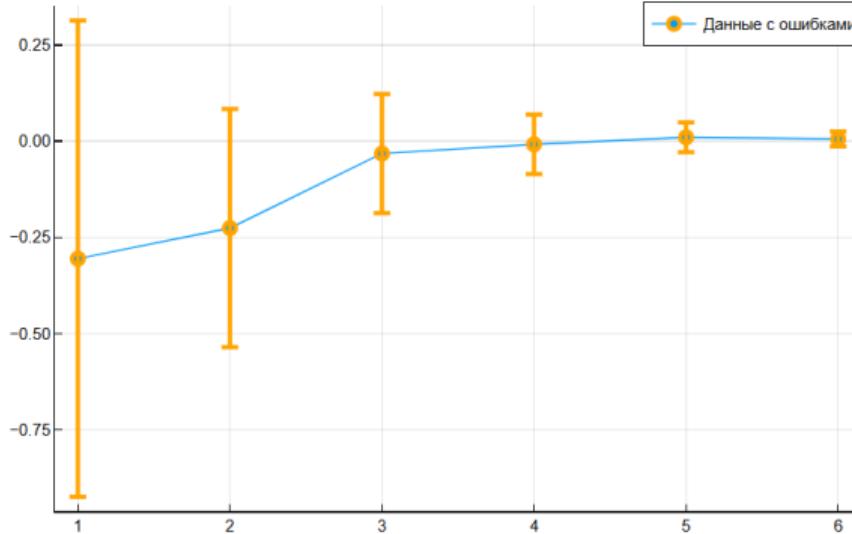


Figure 12: Задание 5.2.12

## Задание 5.2.13

```
[24]: # 5.2.13 Построить гистограмму нормального распределения  
d = Normal(35.0, 10.0)  
ages = rand(d, 1000)  
histogram(ages, label="Распределение", xlabel="Возраст", ylabel="Количество")
```

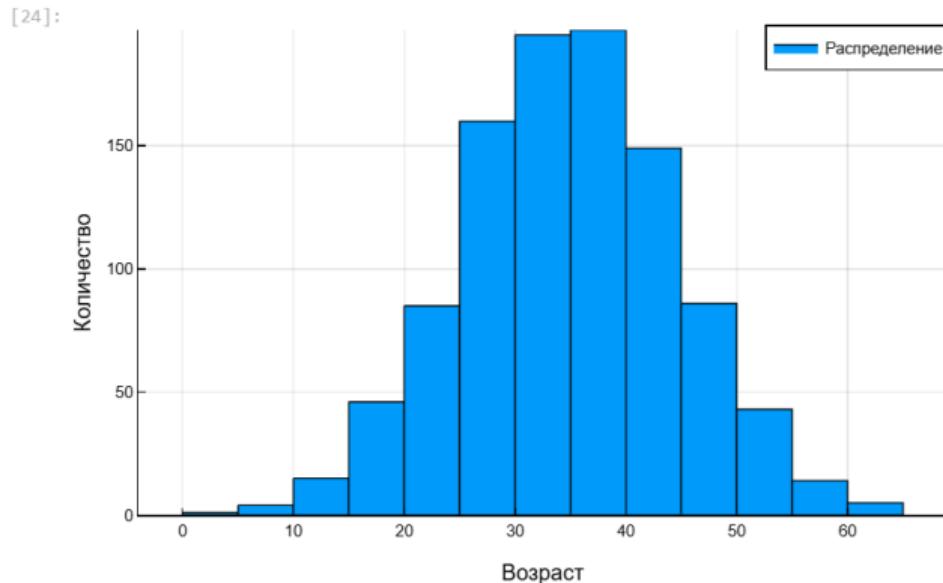


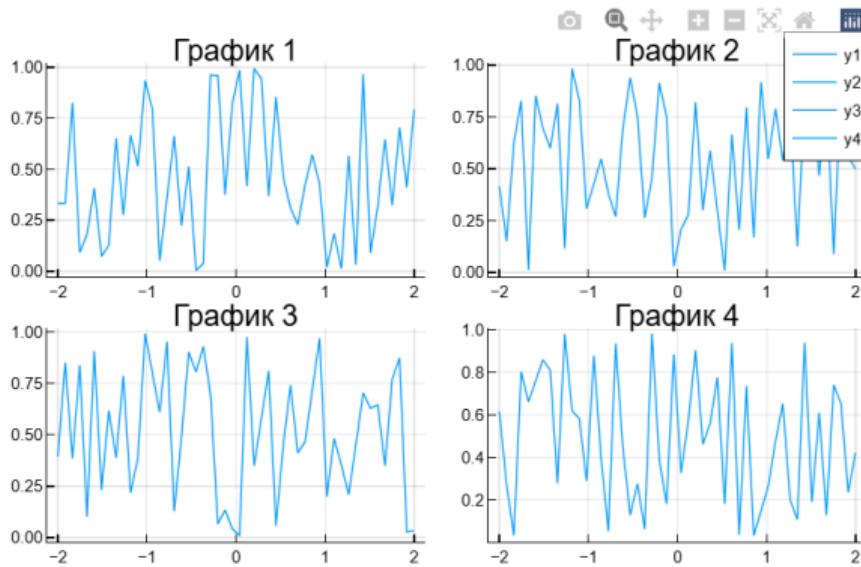
Figure 13: Задание 5.2.13

## Задание 5.2.14

[25]:

```
# 5.2.14 Создать подграфики в разных Layouts  
x_sub = range(-2, 2, length=50)  
y_sub = rand(50, 4)  
plot(x_sub, y_sub, layout=[2,2], title=["График 1" "График 2" "График 3" "График 4"])
```

[25]:



**Figure 14:** Задание 5.2.14

## Задачи для самостоятельной работы

---

## Задание 5.4.1

```
[26]: # 5.4.1 Построить все типы графиков sin(x) в одном окне
x_sin = range(0, 2π, length=100)
y_sin = sin.(x_sin)
p1 = plot(x_sin, y_sin, title="Линейный")
p2 = scatter(x_sin[1:5:end], y_sin[1:5:end], title="Точечный")
p3 = plot(x_sin, y_sin, seriestype=:sticks, title="Палочки")
p4 = plot(x_sin, y_sin, seriestype=:step, title="Ступенчатый")
p5 = histogram(y_sin, title="Гистограмма")
p6 = bar(1:10, abs.(y_sin[1:10:100]), title="Столбчатая")
plot(p1, p2, p3, p4, p5, p6, layout=(3,2), legend=false)
```

[26]:

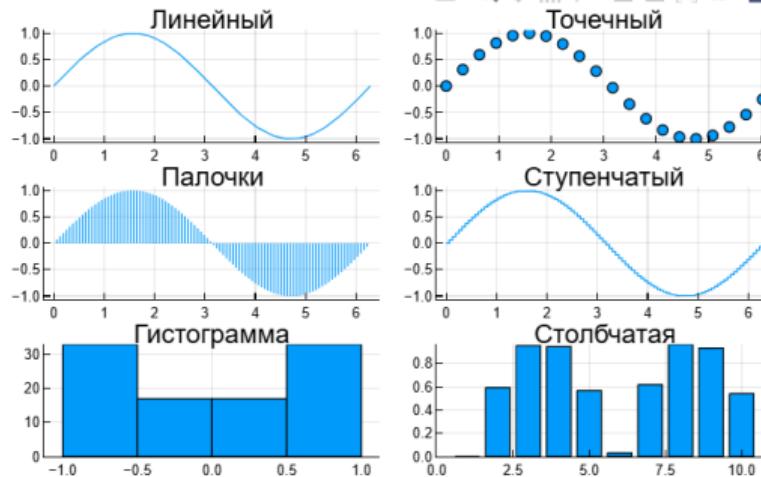


Figure 15: Задание 5.4.1

## Задание 5.4.2

```
[31]: # 5.4.2 Построить sin(x) со всеми типами оформления линий
plot(x_sin, y_sin, linestyle=:solid, label="Стиль 1", title="Разные стили линий")
plot!(x_sin, y_sin .+ 0.3, linestyle=:dash, label="Стиль 2")
plot!(x_sin, y_sin .+ 0.6, linestyle=:dot, label="Стиль 3")
plot!(x_sin, y_sin .+ 0.9, linestyle=:dashdot, label="Стиль 4")
```

```
[31]:
```

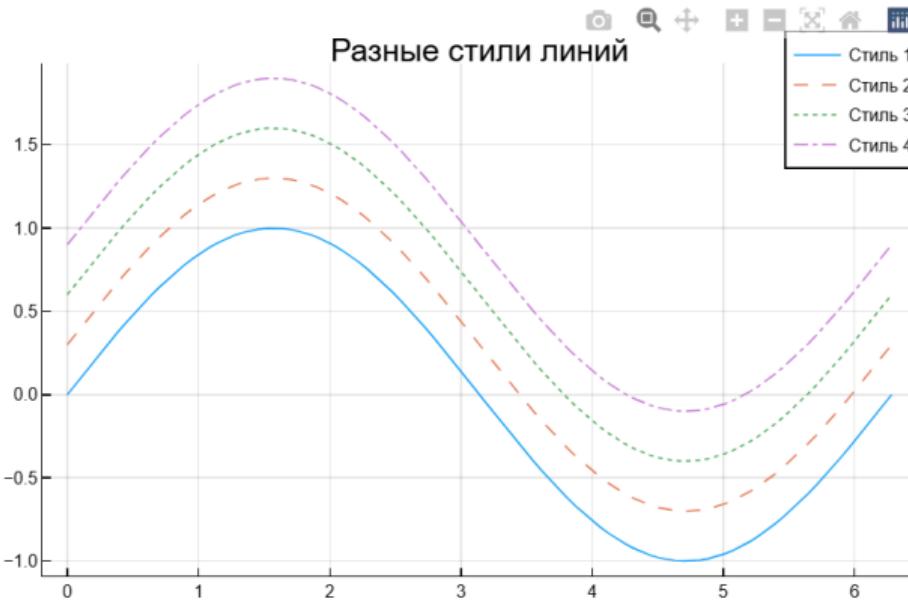


Figure 16: Задание 5.4.2

## Задание 5.4.3

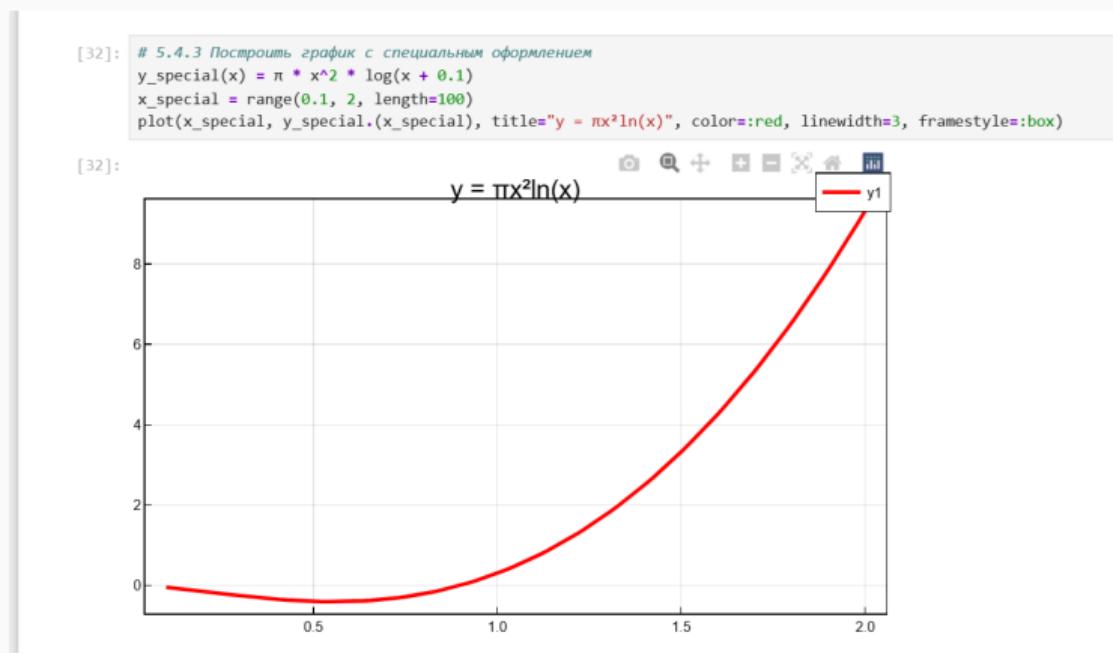


Figure 17: Задание 5.4.3

## Задание 5.4.4

```
[36]: # 5.4.4 Изобразить 4 типа графиков по точкам в подикнах
using Plots
gr()
x_family = [-2, -1, 0, 1, 2]
y_family = x_family.^3 - 3x_family
p1_fam = scatter(x_family, y_family, title="Точки")
p2_fam = plot(x_family, y_family, title="Линии")
p3_fam = plot(x_family, y_family, marker=:circle, title="Линии и точки")
x_curve = range(-2, 2, length=100)
y_curve = x_curve.^3 - 3x_curve
p4_fam = plot(x_curve, y_curve, title="Кривая")
plot(p1_fam, p2_fam, p3_fam, p4_fam, layout=(2,2))
savefig("vsevolod_kozlov.png")
```

[36]: "c:\\studies\\RUDDN\\linux\_stoch\\lab5\\vsevolod\_kozlov.png"

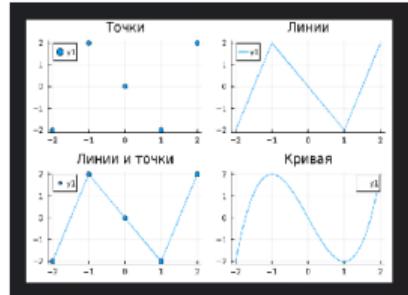


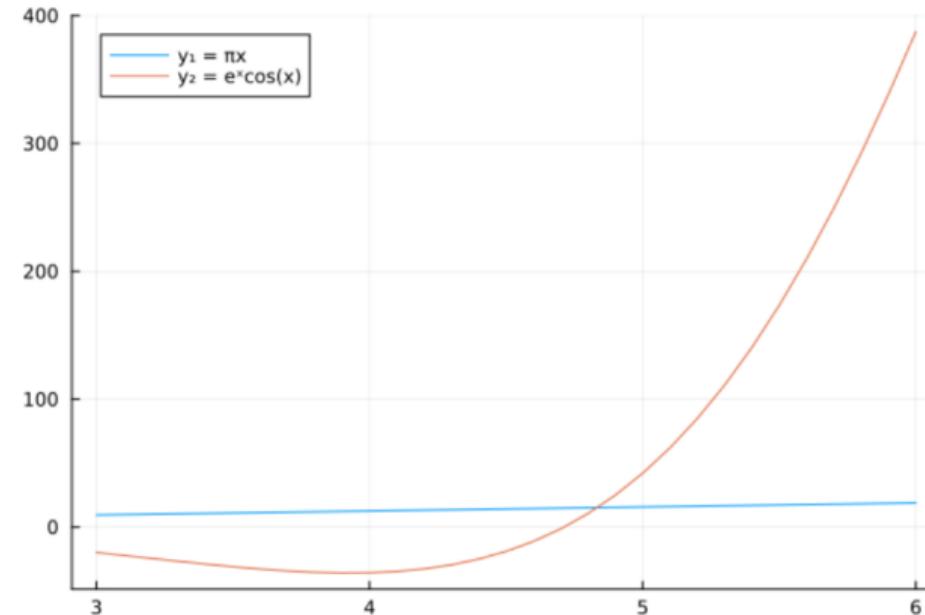
Figure 18: Задание 5.4.4

## Задание 5.4.5

[37]:

```
# 5.4.5 Построить графики двух функций с общей осью и с двумя осями
x_range = 3:0.1:6
plot(x_range, pi * x_range, label="y1 = πx")
plot!(x_range, exp.(x_range) .* cos.(x_range), label="y2 = excos(x)")
```

[37]:



## Задание 5.4.6

```
[38]: # 5.4.6 Построить график экспериментальных данных с ошибками
x_exp = 1:10
y_exp = [2.1, 3.9, 6.2, 8.1, 9.8, 12.1, 13.9, 16.2, 18.0, 19.9]
y_err = [0.3, 0.4, 0.5, 0.4, 0.6, 0.5, 0.4, 0.6, 0.5, 0.7]
scatter(x_exp, y_exp, yerr=y_err, label="Экспериментальные данные")
```

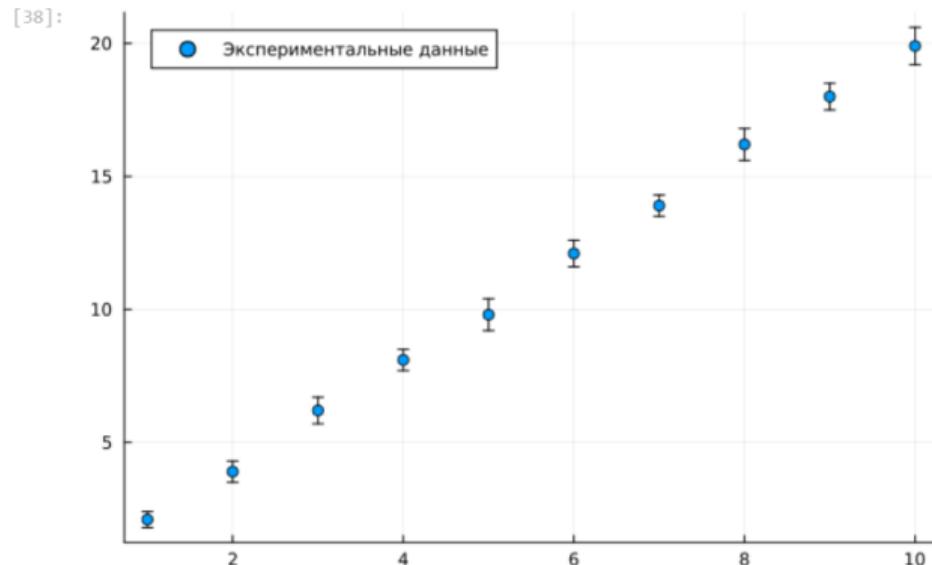


Figure 20: Задание 5.4.6

## Задание 5.4.7

```
[39]: # 5.4.7 Построить точечный график случайных данных  
scatter(rand(50) * 10, rand(50) * 10, title="Случайные данные", xlabel="X", ylabel="Y")
```

```
[39]:
```

Случайные данные

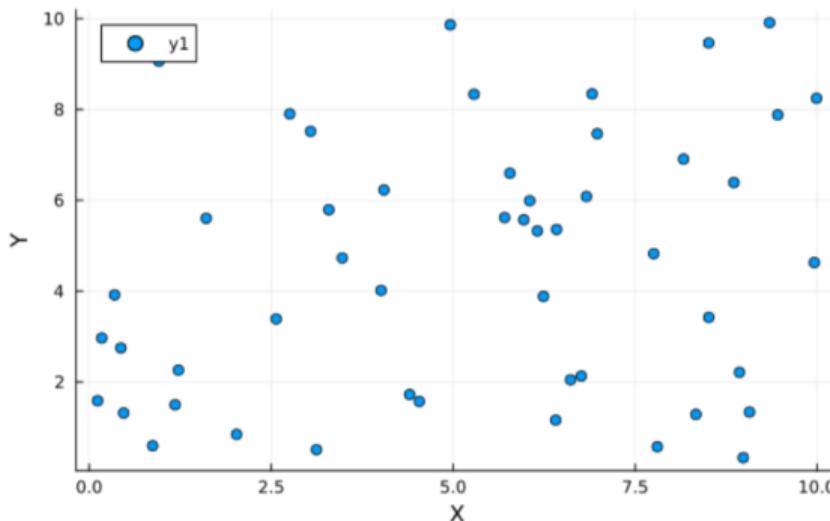


Figure 21: Задание 5.4.7

## Задание 5.4.8

```
[40]: # 5.4.8 Построить 3D точечный график случайных данных  
scatter(rand(100) * 10, rand(100) * 10, rand(100) * 10, title="3D случайные данные")
```

```
[40]:
```

3D случайные данные

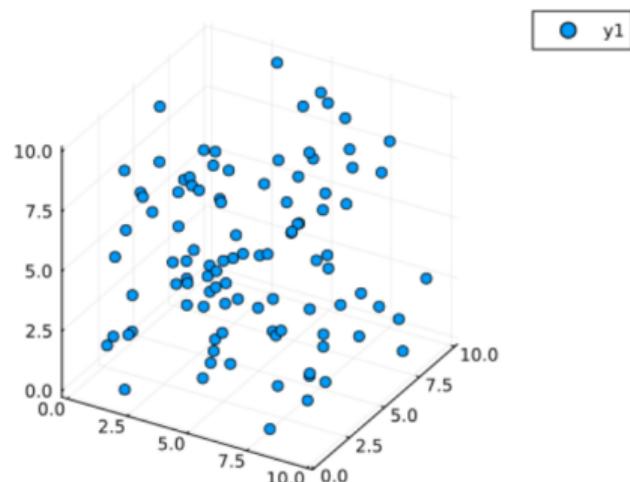


Figure 22: Задание 5.4.8

## Задание 5.4.9

```
11]: # 5.4.9 Создать анимацию построения синусоиды
anim_sin = @animate for i in 1:100
    x_anim = range(0, i/10, length=100)
    plot(x_anim, sin.(x_anim), xlim=(0, 10), ylim=(-1.5, 1.5), title="Анимация синусоиды")
end
gif(anim_sin, "sine_animation.gif", fps=10)
```

[ Info: Saved animation to C:\studies\RUDN\linux\_stoch\lab5\sine\_animation.gif

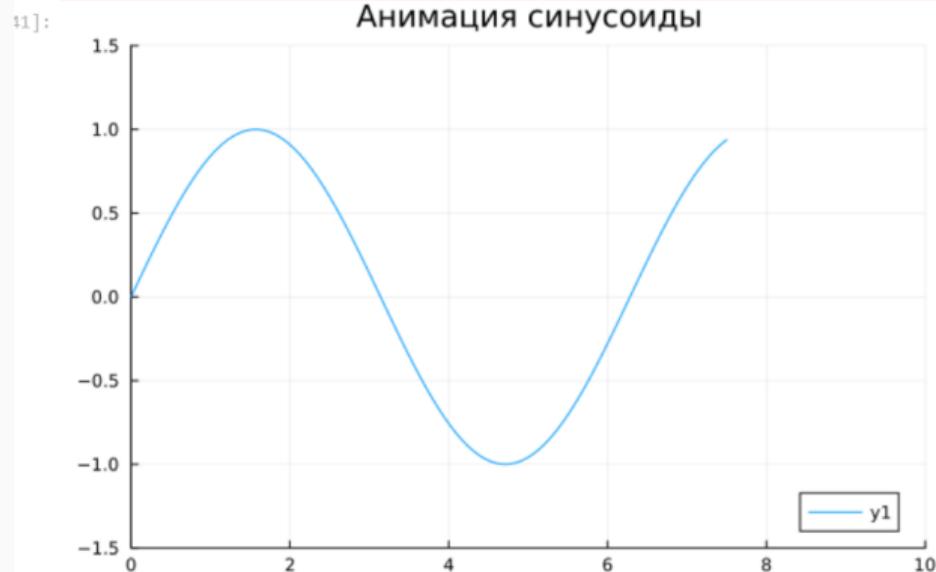


Figure 23: Задание 5.4.9

## Задание 5.4.10

```
[42]: # 5.4.10 Создать анимацию гипоциклоиды для разных k
function hypocyloid_animation(k)
    r = 1
    n = 100
    theta = range(0, 2pi, length=n)
    anim = @animate for i in 1:n
        t = theta[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, xlim=(-4,4), ylim=(-4,4), title="Гипоциклоида k=$k")
    end
    return anim
end
hypocyloid_animation(3, "hypocyloid_k3.gif", fps=10)
hypocyloid_animation(4, "hypocyloid_k4.gif", fps=10)

[ Info: Saved animation to C:\studies\RUDN\linux_stoch\lab5\hypocyloid_k3.gif
[ Info: Saved animation to C:\studies\RUDN\linux_stoch\lab5\hypocyloid_k4.gif
```

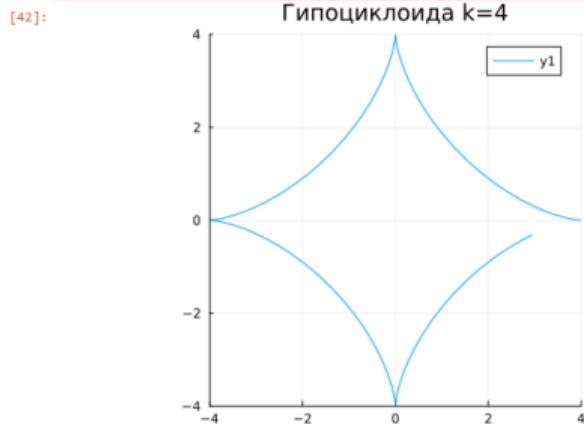


Figure 24: Задание 5.4.10

## Задание 5.4.11

[43]:

```
# 5.4.11 Создать анимацию эпициклоиды для разных k
function epicycloid_animation(k)
    r = 1
    n = 100
    theta = range(0, 2π, length=n)
    anim = @animate for i in 1:n
        t = theta[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, xlim=(-8,8), ylim=(-8,8), title="Эпициклоида k=$k")
    end
    return anim
end
gif(epicycloid_animation(3), "epicycloid_k3.gif", fps=10)
gif(epicycloid_animation(4), "epicycloid_k4.gif", fps=10)

[ Info: Saved animation to C:\studies\RUDN\linux_stoch\lab5\epicycloid_k3.gif
[ Info: Saved animation to C:\studies\RUDN\linux_stoch\lab5\epicycloid_k4.gif
```

[43]:

Эпициклоида k=4

30/31

Figure 25: Задание 5.4.11

## Выводы

Освоил синтаксис языка Julia для построения графиков.