

Лабораторная работа №5

Построение графиков

Тазаева Анастасия Анатольевна

Содержание

1	Цель работы	6
2	Задание	7
3	Выполнение лабораторной работы	8
3.1	Примеры из раздела 5.2	8
3.2	Самостоятельная работа	35
4	Выводы	45

Список иллюстраций

3.1	Установка пакетов для julia	8
3.2	Примеры. Основные пакеты для работы с графиками в Julia. ч.1 . .	8
3.3	Примеры. Основные пакеты для работы с графиками в Julia. ч.2 . .	9
3.4	Примеры. Основные пакеты для работы с графиками в Julia. ч.3 . .	9
3.5	Примеры. Основные пакеты для работы с графиками в Julia. ч.4 . .	10
3.6	Примеры. Основные пакеты для работы с графиками в Julia. ч.5 . .	10
3.7	Примеры. Опции при построении графика. ч.1	11
3.8	Примеры. Опции при построении графика. ч.2	12
3.9	Примеры. Опции при построении графика. ч.3	13
3.10	Примеры. Простой точечный график.	14
3.11	Примеры. Точечный график с кодированием значения размером точки.	14
3.12	Примеры. 3-мерный точечный график с кодированием значения размером точки.	15
3.13	Примеры. Аппроксимация данных.	16
3.14	Примеры. Две оси ординат.	17
3.15	Примеры. Полярные координаты.	17
3.16	Примеры. Параметрический график кривой на плоскости и в пространстве	18
3.17	Примеры. График поверхности. ч.1	19
3.18	Примеры. График поверхности. ч.2	20
3.19	Примеры. Линии уровня. ч.1	21
3.20	Примеры. Линии уровня. ч.2	22
3.21	Примеры. Векторные поля	23
3.22	Примеры. Gif-анимация	24
3.23	Примеры. Гипоциклоида	25
3.24	Примеры. Errorbars. ч.1	26
3.25	Примеры. Errorbars. ч.2	27
3.26	Примеры. Errorbars. ч.3	28
3.27	Примеры. Errorbars. ч.4	29
3.28	Примеры. Использование пакета Distributions. ч.1	29
3.29	Примеры. Использование пакета Distributions. ч.2	30
3.30	Примеры. Использование пакета Distributions. ч.3	31
3.31	Примеры. Подграфики. ч.1	32
3.32	Примеры. Подграфики. ч.2	33
3.33	Примеры. Подграфики. ч.3	34
3.34	Примеры. Подграфики. ч.4	34

3.35 Задание 1	35
3.36 Задание 2	36
3.37 Задание 3	37
3.38 Задание 4	38
3.39 Задание 5	38
3.40 Задание 6	39
3.41 Задание 7	40
3.42 Задание 8	41
3.43 Задание 9	42
3.44 Задание 10	43
3.45 Задание 11	44

Список таблиц

1 Цель работы

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

2 Задание

1. Используя Jupyter Lab, повторите примеры из раздела 5.2.
2. Выполните задания для самостоятельной работы (раздел 5.4).

3 Выполнение лабораторной работы

3.1 Примеры из раздела 5.2

Примеры представлены на рис. 3.1 - 3.34. Сохраненные файлы по заданиям располагаются в смежном каталоге lab5+samost/examples

```
(@v1.10) pkg> add Plots.jl
Resolving package versions...
No Changes to `C:\Users\noname\.julia\environments\v1.10\Project.toml`
No Changes to `C:\Users\noname\.julia\environments\v1.10\Manifest.toml`
Precompiling project...
29 dependencies successfully precompiled in 16 seconds. 162 already precompiled.
```

Рис. 3.1: Установка пакетов для julia

Основные пакеты для работы с графиками в Julia

```
using Pkg
Pkg.add("Plots")
Pkg.add("PyPlot")
Pkg.add("Plotly")
Pkg.add("UnicodePlots")
# подключаем для использования Plots:
using Plots

@[32m@[1m Resolving@[22m@[39m package versions... ***

Рассмотрим построение графика функции  $f(x) = (3x^2 + 6x - 9)e^{[-0,3x]}$  разными способами. Фактически для построения графика функции требуется иметь массив соответствующих значений  $x$  и  $y$ .

# задание функции:
f(x) = (3x.^2 + 6x .- 9).*exp.(-0.3x)

f (generic function with 1 method)

# 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))

151-element Vector{Float64}: ***

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

151-element Vector{Float64}: ***
```

Рис. 3.2: Примеры. Основные пакеты для работы с графиками в Julia. ч.1


```
# 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  $f(x)=(3x^2+6x-9)e^{-0.3x}$  ",
      xlabel="Variable x",
      ylabel="Variable y",
      color="blue")
```

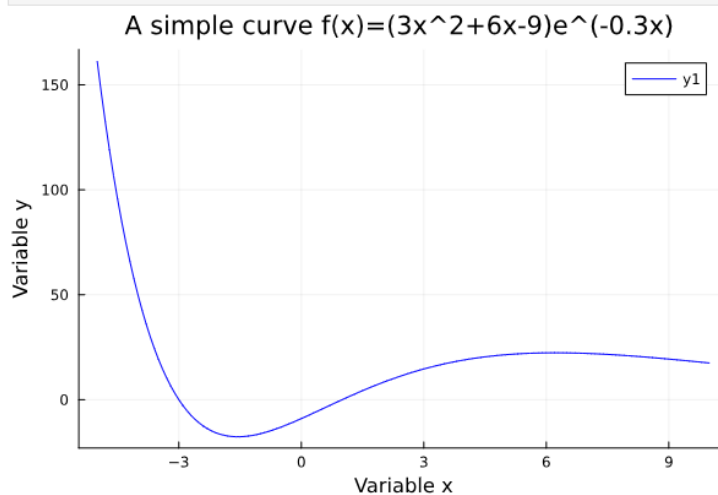


Рис. 3.3: Примеры. Основные пакеты для работы с графиками в Julia. ч.2

```
# 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")
```

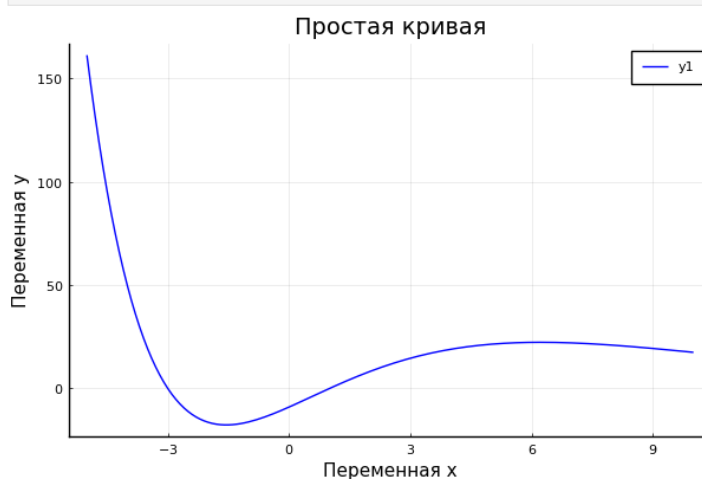


Рис. 3.4: Примеры. Основные пакеты для работы с графиками в Julia. ч.3

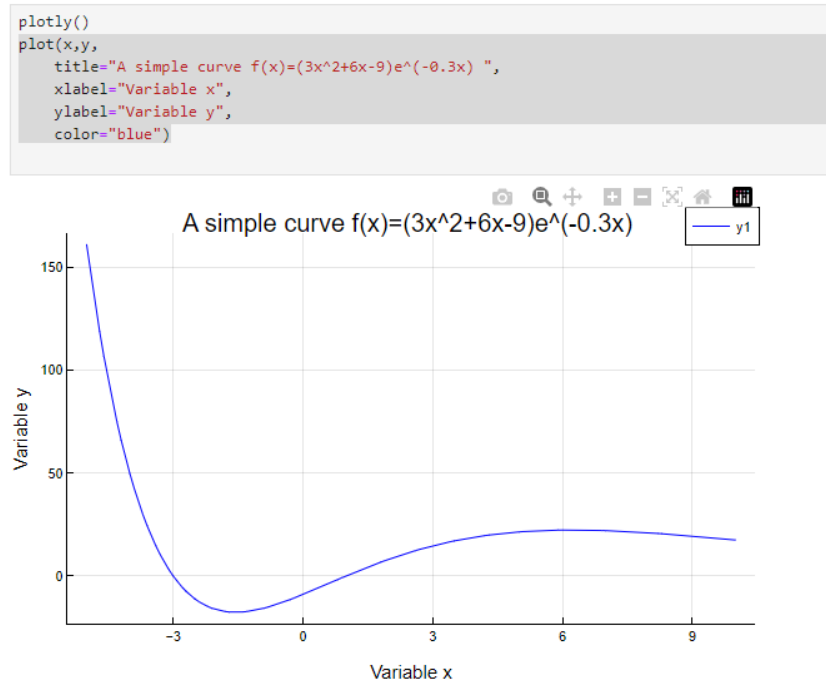


Рис. 3.5: Примеры. Основные пакеты для работы с графиками в Julia. ч.4

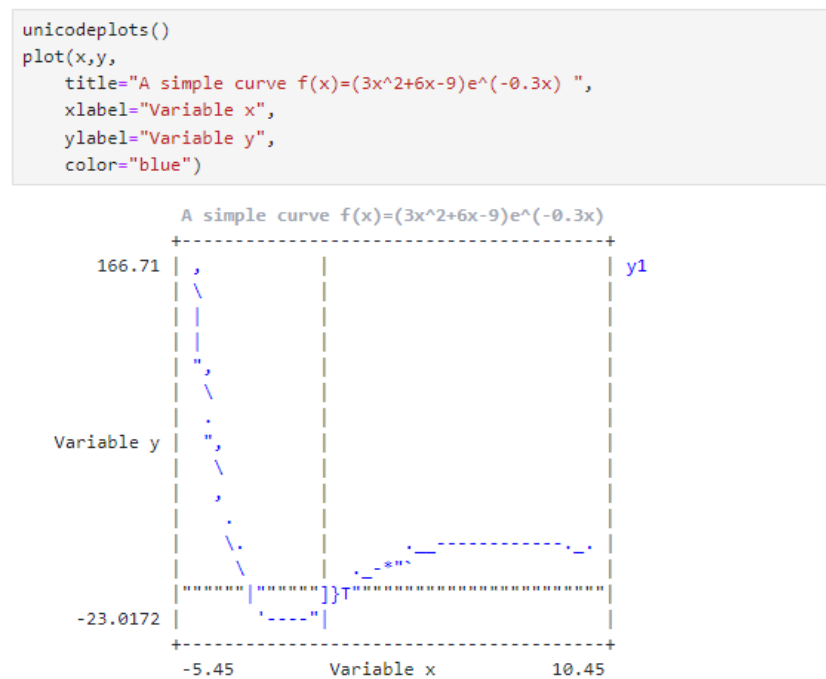


Рис. 3.6: Примеры. Основные пакеты для работы с графиками в Julia. ч.5

Опции при построении графика

Далее на примере графика функции $\sin(x)$ и графика разложения этой функции в ряд Тейлора:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots = \sum_{n=0}^{\infty} (-1)^n * \frac{x^{2n+1}}{(2n+1)!}, x \in C$$

```
# указывается, что для построения графика используется pyplot():
pyplot()
# задание функции sin(x):
sin_theor(x) = sin(x)
# построение графика функции sin(x):
plot(sin_theor,
title="График функции sin(x)",
xlabel="Переменная x",
color="orange")
```

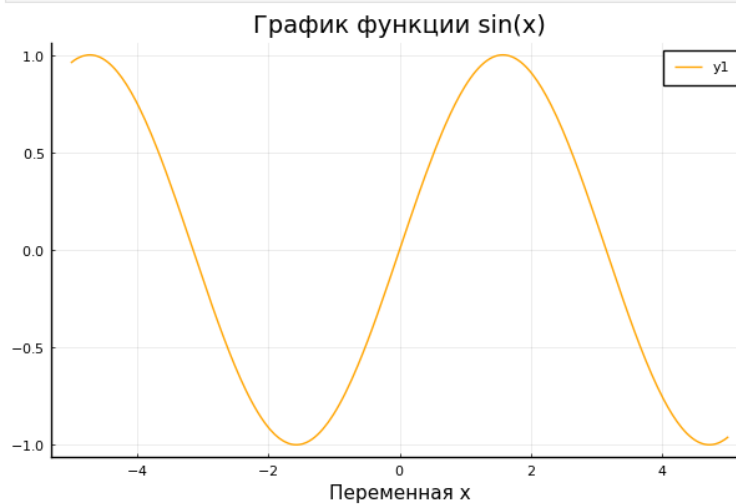
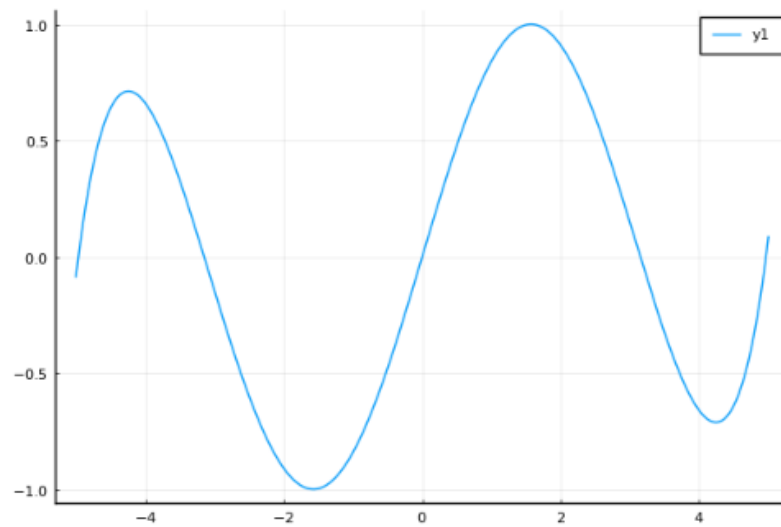


Рис. 3.7: Примеры. Опции при построении графика. ч.1

```
# задание функции разложения исходной функции в ряд Тейлора:
sin_taylor(x) = [(-1)^i*x^(2*i+1)/factorial(2*i+1) for i in 0:4] |> sum
plot(sin_taylor)
```



```
# построение двух функций на одном графике:
plot(sin_theor)
plot!(sin_taylor)
```

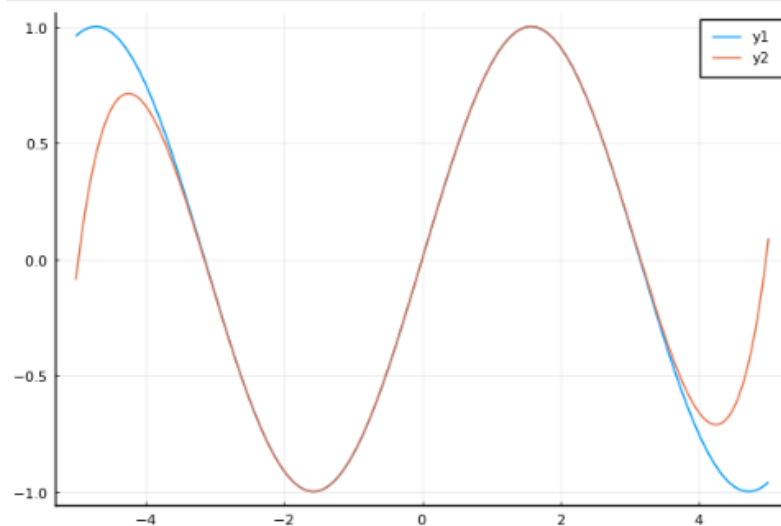


Рис. 3.8: Примеры. Опции при построении графика. ч.2

```

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))

```

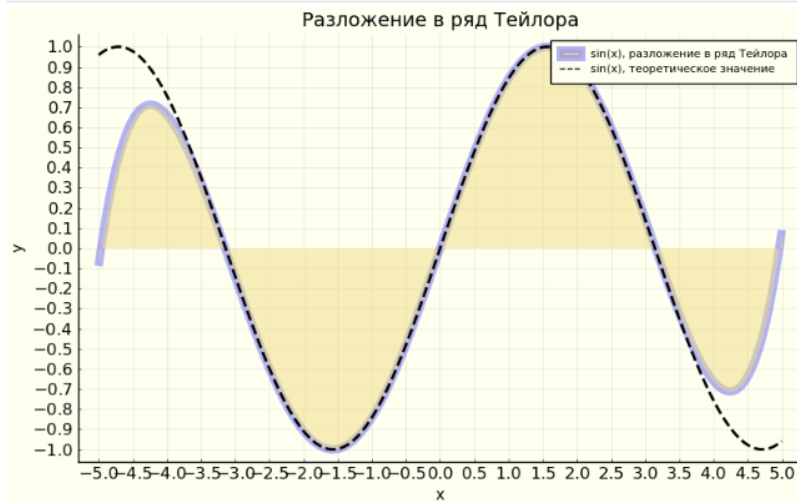


Рис. 3.9: Примеры. Опции при построении графика. ч.3

Простой 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"  
)
```

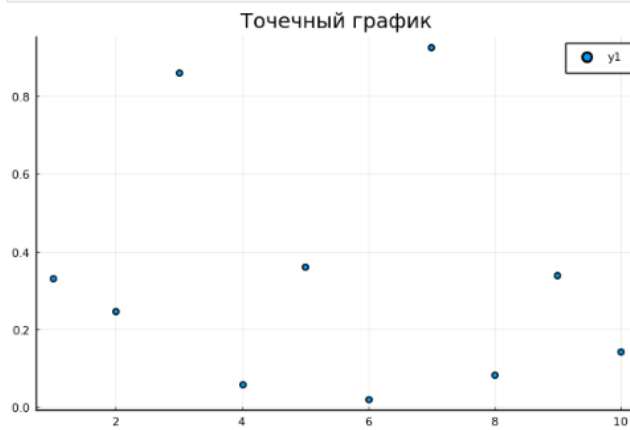


Рис. 3.10: Примеры. Простой точечный график.

Точечный график с кодированием значения размером точки

```
# 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)
```

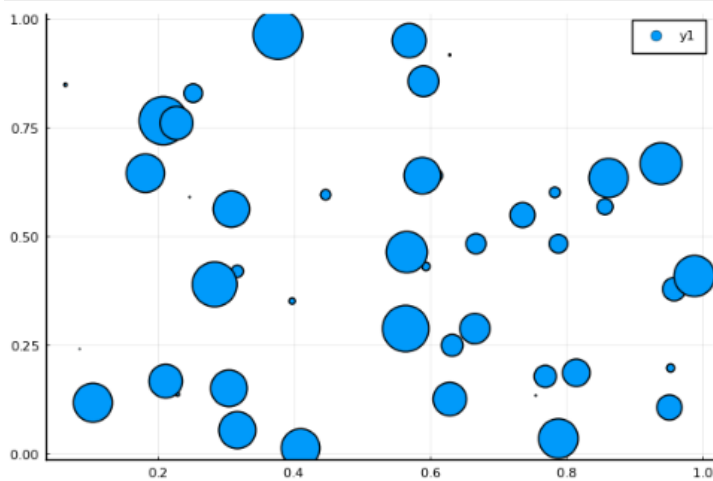


Рис. 3.11: Примеры. Точечный график с кодированием значения размером точки.

3-мерный точечный график с кодированием значения размером точки

```
# 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)
```

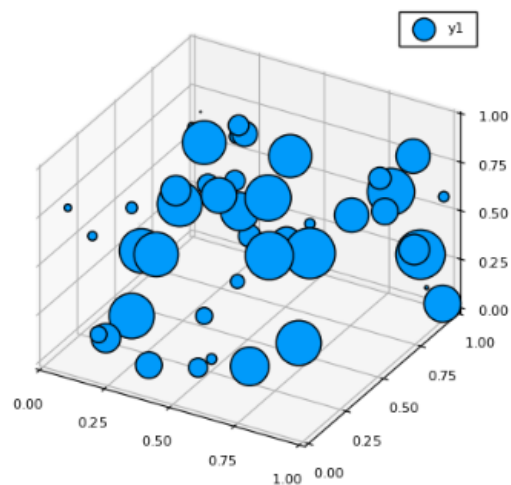
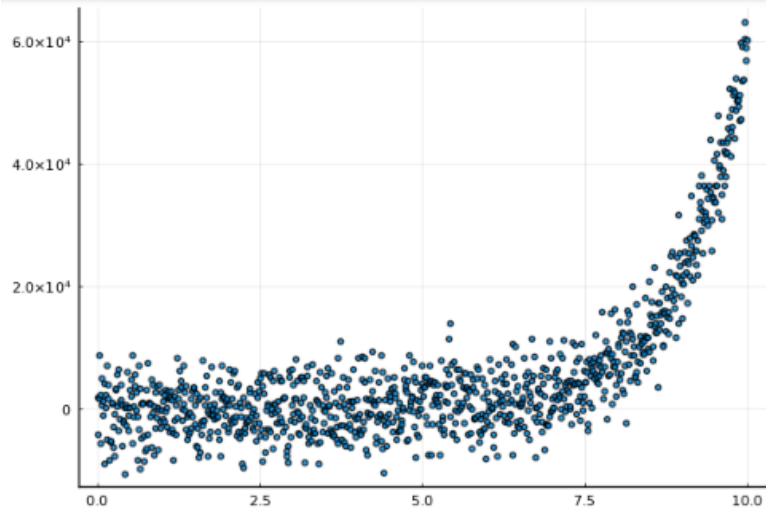


Рис. 3.12: Примеры. 3-мерный точечный график с кодированием значения размером точки.

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)
```



```
# 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:  
g = 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,g,linewidth=3,color=:red)
```

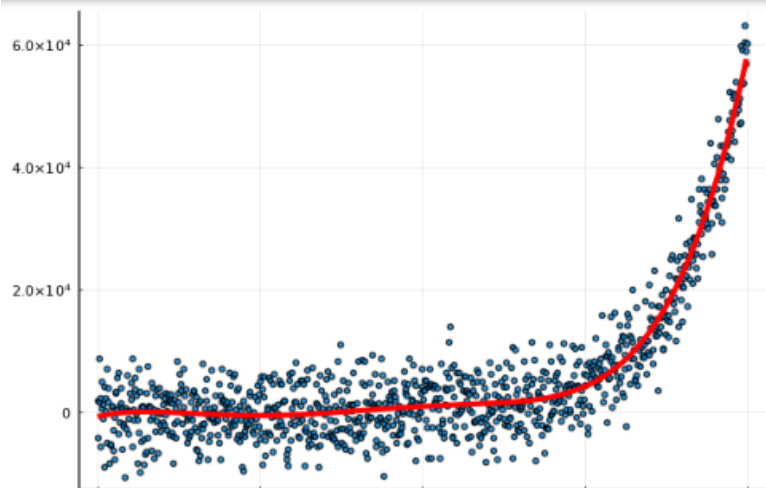


Рис. 3.13: Примеры. Аппроксимация данных.

Две оси ординат

```
# пример случайной траектории
# (заданы обозначение траектории, легенда вверху справа, без сетки)
plot(randn(100),
     ylabel="y1",
     leg=:topright,
     grid=:off,
)
```

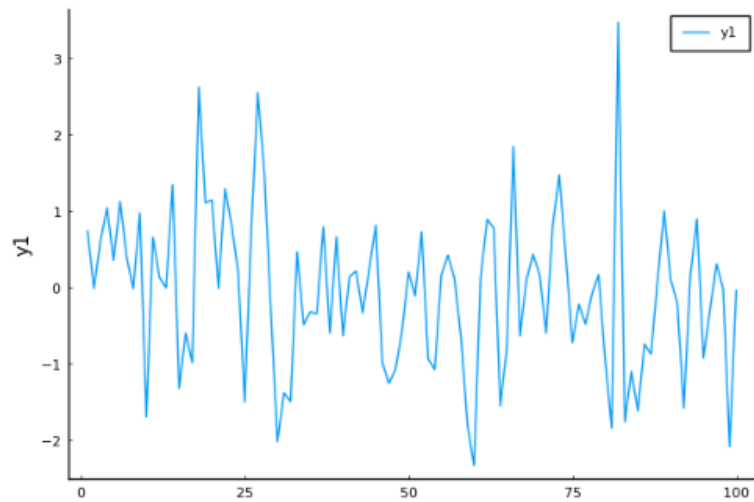


Рис. 3.14: Примеры. Две оси ординат.

Полярные координаты

Приведём пример построения графика функции

$$r(\theta) = 1 + \cos(\theta) + \sin^2(\theta)$$

в полярных координатах

```
# 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)
)
```

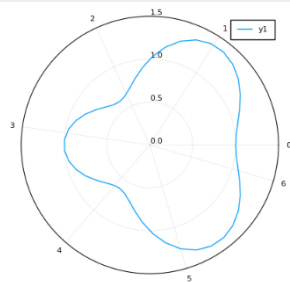
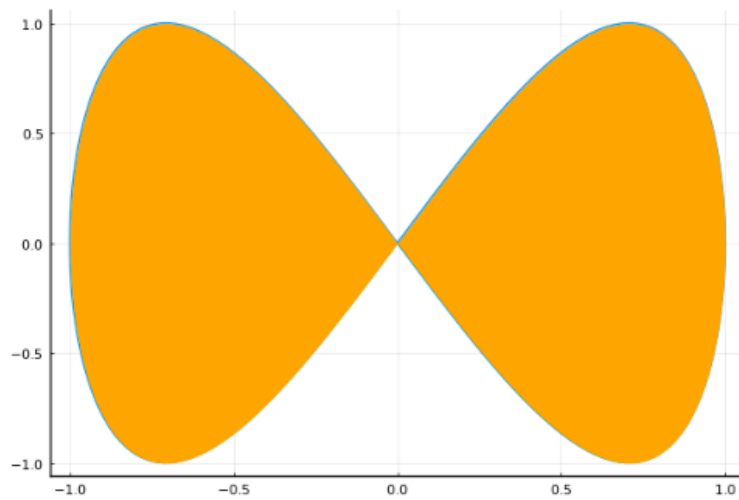


Рис. 3.15: Примеры. Полярные координаты.

Параметрический график кривой на плоскости

```
# параметрическое уравнение:  
x_f(t) = sin(t)  
y_f(t) = sin(2t)  
# построение графика:  
plot(x_f, y_f, 0, 2π, leg=false, fill=(0,:orange))
```



Параметрический график кривой в пространстве

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

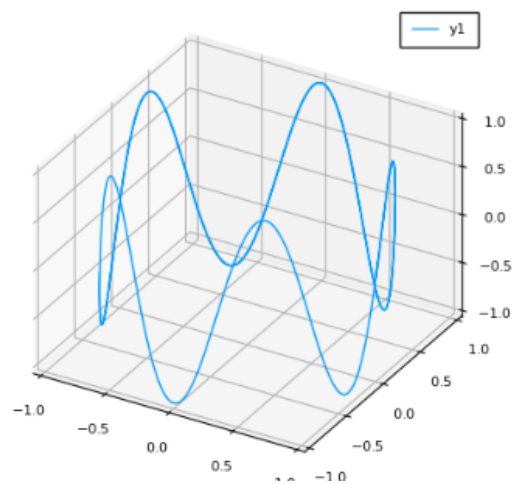
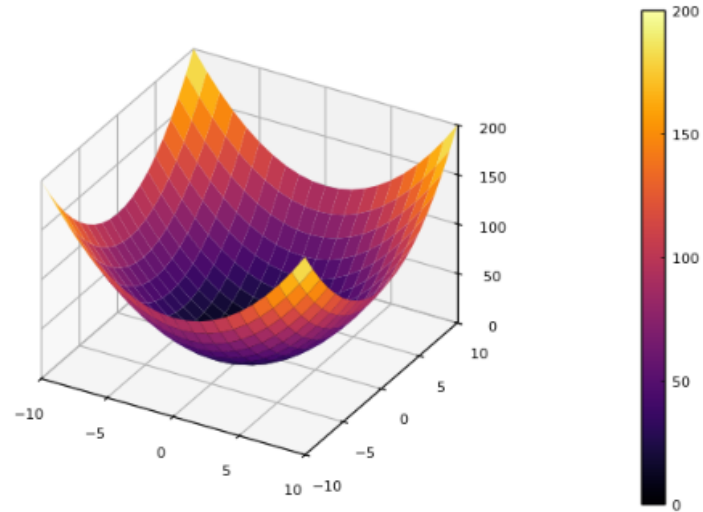


Рис. 3.16: Примеры. Параметрический график кривой на плоскости и в пространстве

График поверхности

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



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

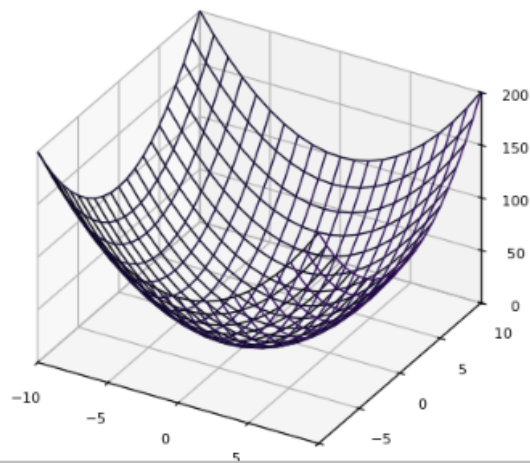
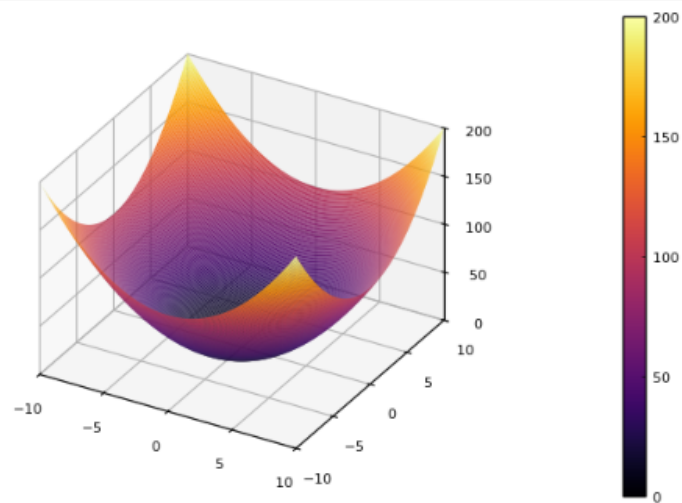


Рис. 3.17: Примеры. График поверхности. ч.1

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



```
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,
      linestyle = :surface,
      c=cgrad([:red,:blue]),
      camera=(-30,30),
    )
```

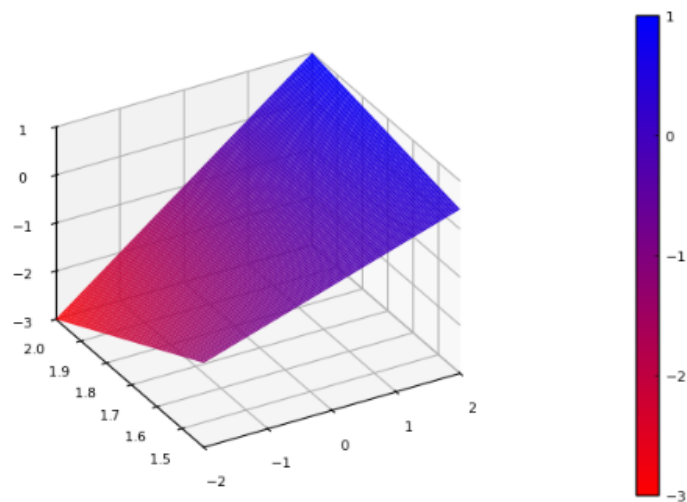
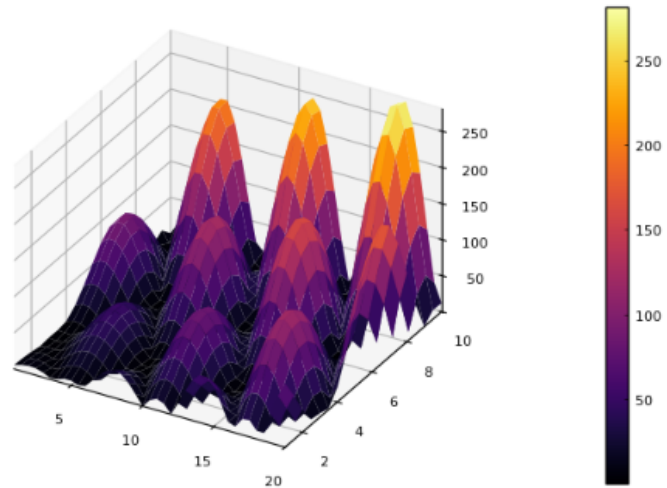


Рис. 3.18: Примеры. График поверхности. ч.2

Линии уровня

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



```
contour(x, y, g_f)
```

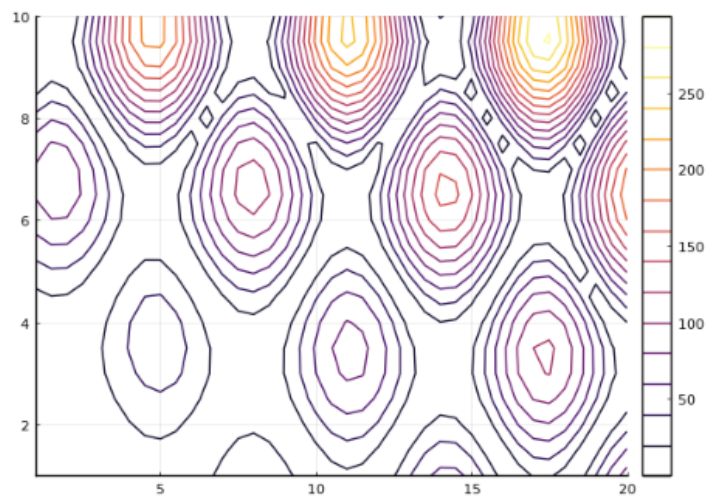


Рис. 3.19: Примеры. Линии уровня. ч.1

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

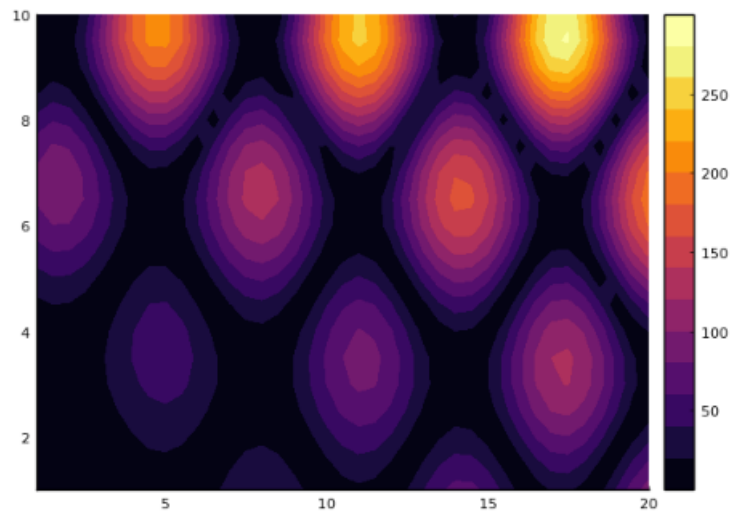
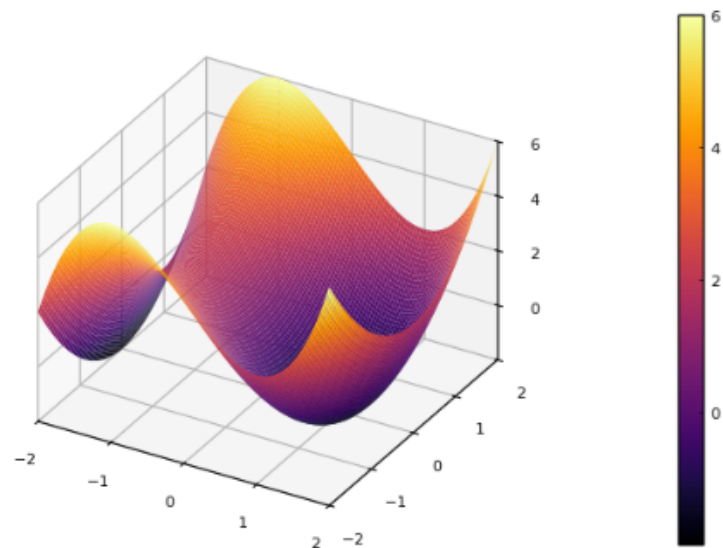


Рис. 3.20: Примеры. Линии уровня. ч.2

Векторные поля

```
# 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,  
      linestyle = :surface  
)
```



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

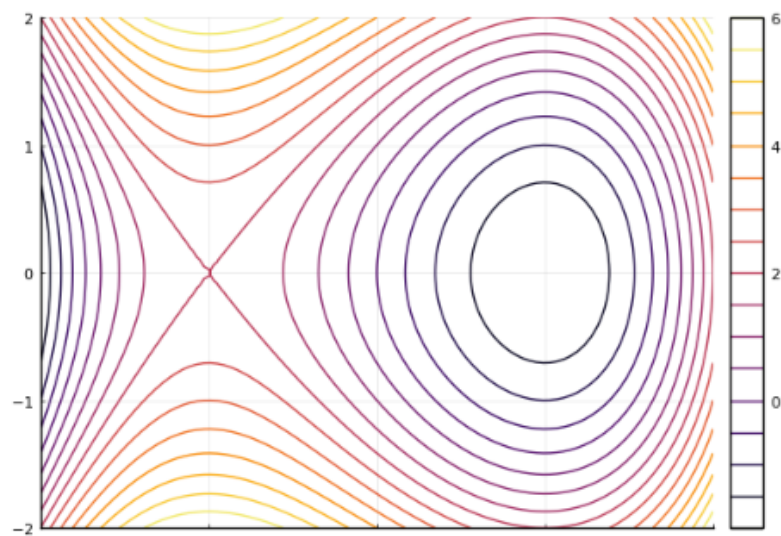
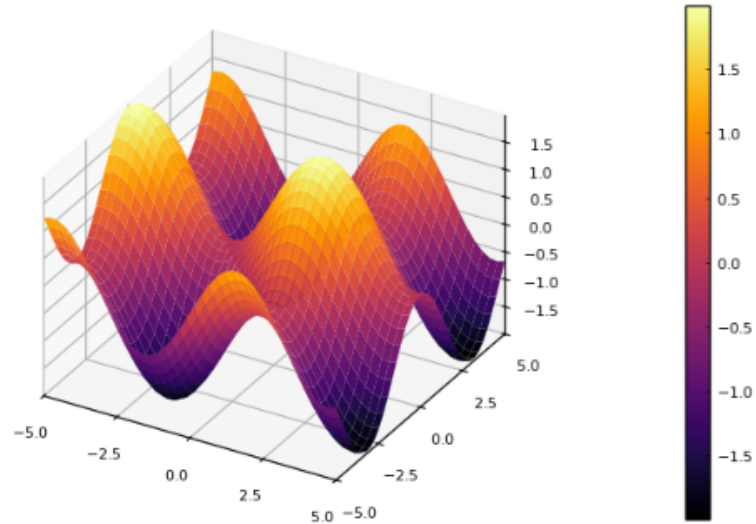


Рис. 3.21: Примеры. Векторные поля

Gif-анимация

```
pyplot()
# построение поверхности:
i = 0
X = Y = range(-5, stop=5, length=40)
surface(X, Y, (x,y) -> sin(x+10sin(i))+cos(y))
```



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

[Info: Saved animation to C:\Users\noname\tmp.gif

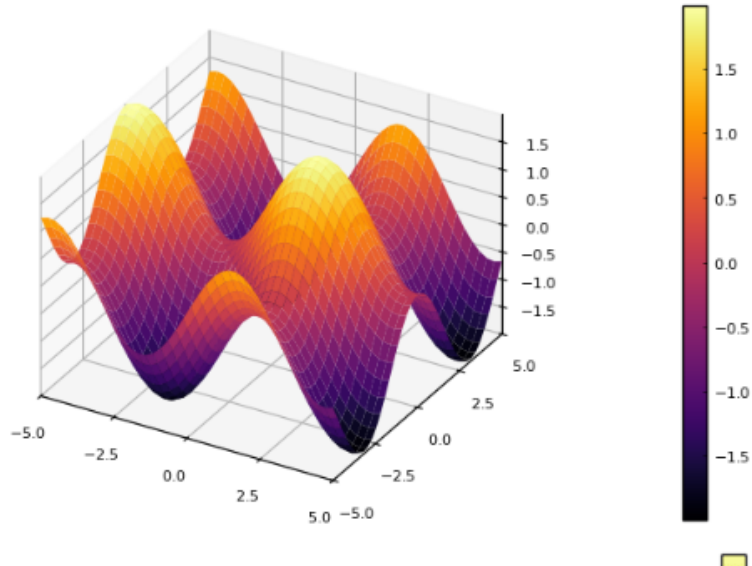


Рис. 3.22: Примеры. Gif-анимация

Гипоциклоида

```
# радиус малой окружности:
rr = 1
# коэффициент для построения большой окружности:
k = 3
# число отсчётов:
n = 100
# массив значений угла θ:
# theta from 0 to 2pi ( + a little extra)
θ = collect(0:2*π/100:2*π+2*π/100)
# массивы значений координат:
X = rr*k*cos.(θ)
Y = rr*k*sin.(θ)
# 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)
i = 50
t = θ[1:i]
# гипоциклоида:
x = rr*(k-1)*cos.(t) + rr*cos.((k-1)*t)
y = rr*(k-1)*sin.(t) - rr*sin.((k-1)*t)
plot!(x,y, c=:red)
# малая окружность:
xc = rr*(k-1)*cos(t[end]) + rr*cos.(θ)
yc = rr*(k-1)*sin(t[end]) + rr*sin.(θ)
plot!(xc,yc,c=:black)
# радиус малой окружности:
x1 = transpose([rr*(k-1)*cos(t[end]) x[end]])
y1 = transpose([rr*(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)
```

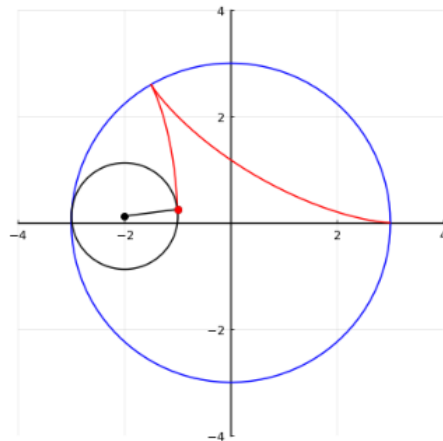


Рис. 3.23: Примеры. Гипоциклоида

Errorbars

```
# подключение пакета Statistics:
```

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

```
Resolving package versions...
```

```
No Changes to `C:\Users\noname\.julia\environments\v1.10\Project.toml`
```

```
No Changes to `C:\Users\noname\.julia\environments\v1.10\Manifest.toml`
```

```
#Зададим массив значений:
```

```
sds = [1, 1/2, 1/4, 1/8, 1/16, 1/32]
```

```
#Затем сгенерируем массив ошибок (отклонений от исходных значений):
```

```
n = 10
```

```
y = [mean(sd*randn(n)) for sd in sds]
```

```
errs = 1.96 * sds / sqrt(n)
```

```
#Построим график исходных значений
```

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

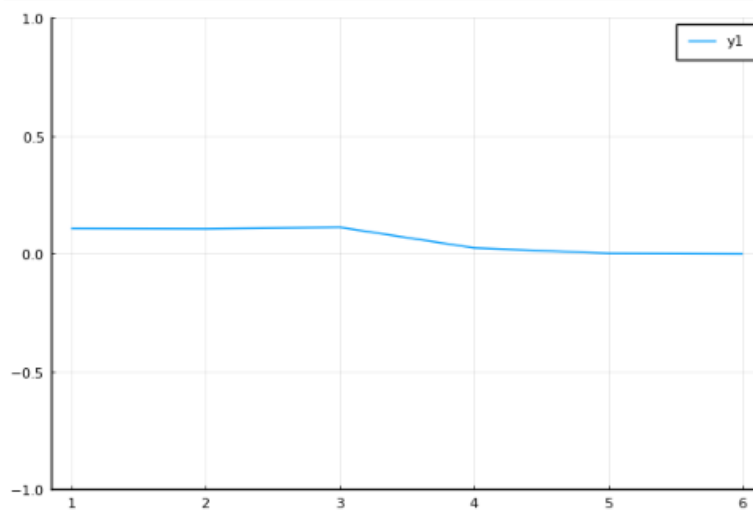


Рис. 3.24: Примеры. Errorbars. ч.1

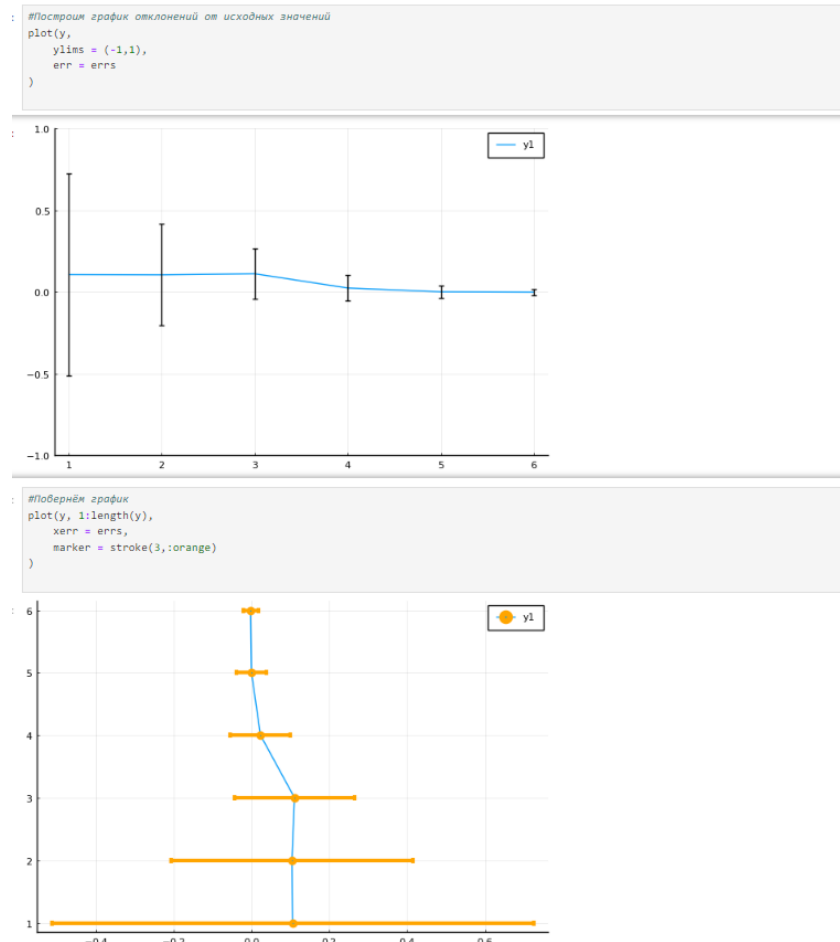
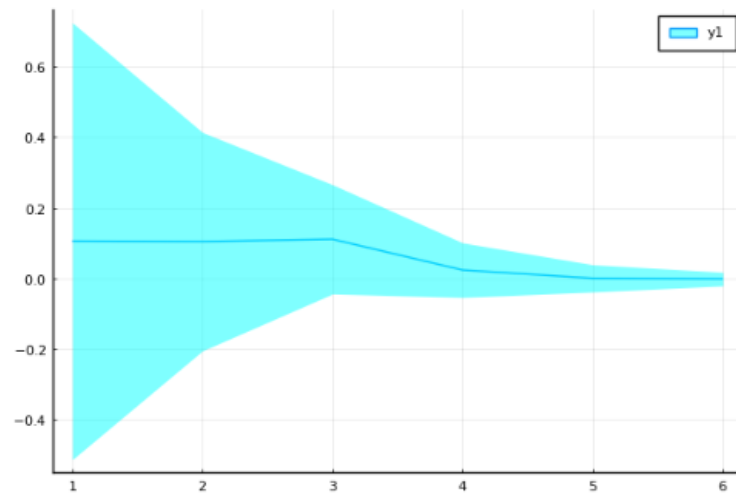


Рис. 3.25: Примеры. Errorbars. ч.2

```
#Заполним область цветом
plot(y,
      ribbon=errs,
      fill=:cyan
    )
```



```
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)
    )
```

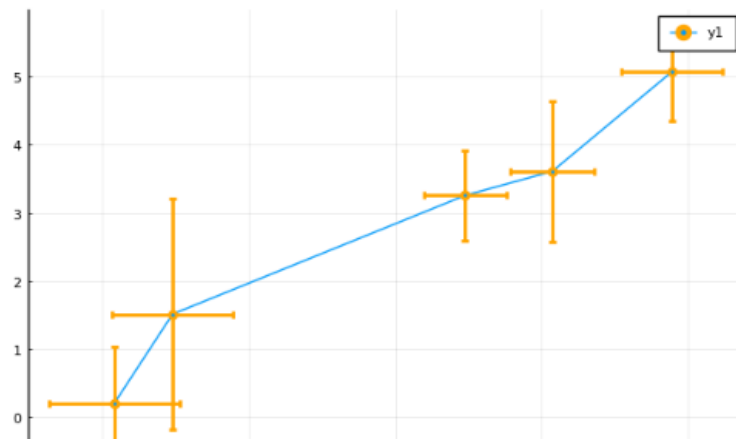


Рис. 3.26: Примеры. Errorbars. ч.3

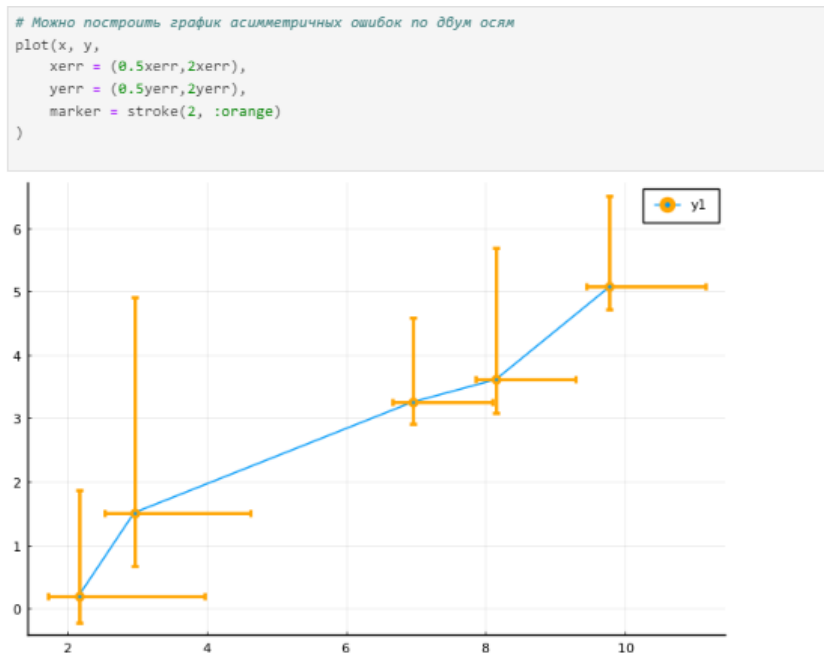


Рис. 3.27: Примеры. Errorbars. ч.4

Использование пакета Distributions

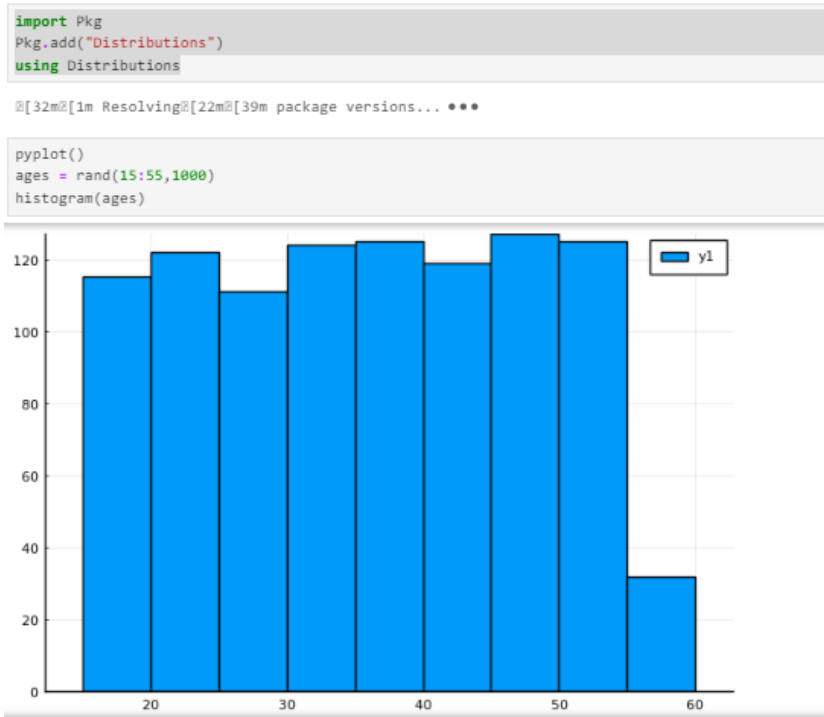


Рис. 3.28: Примеры. Использование пакета Distributions. ч.1

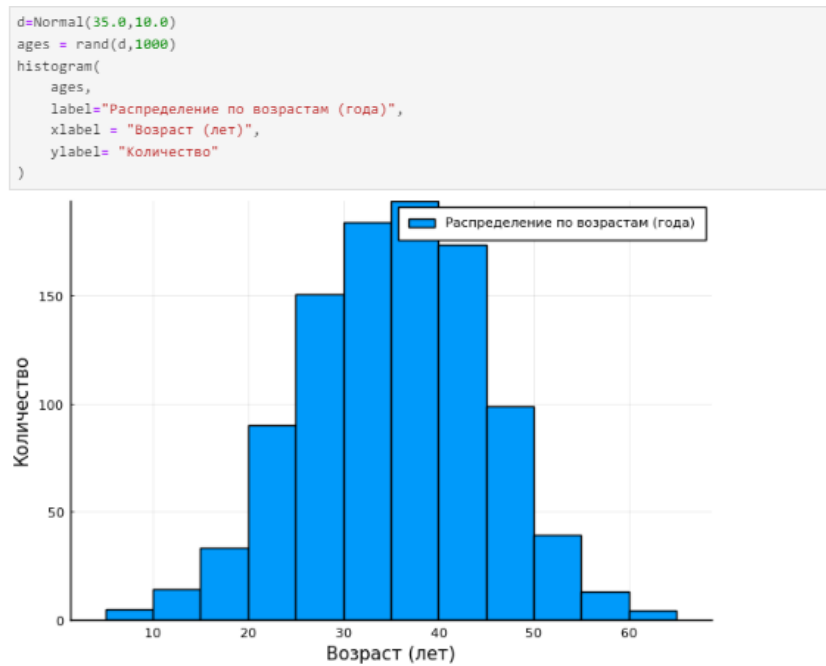


Рис. 3.29: Примеры. Использование пакета Distributions. ч.2

```

plotly()
d1=Normal(10.0,5.0);
d2=Normal(35.0,10.0);
d3=Normal(60.0,5.0);
N=1000;
ages = (Float64)[];
ages = append!(ages,rand(d1,Int64(ceil(N/2))));
ages = append!(ages,rand(d2,N));
ages = append!(ages,rand(d3,Int64(ceil(N/3))));
histogram(
    ages,
    bins=50,
    label="Распределение по возрастам (года)",
    xlabel = "Возраст (лет)",
    ylabel= "Количество",
    title = "Распределение по возрастам (года)"
)

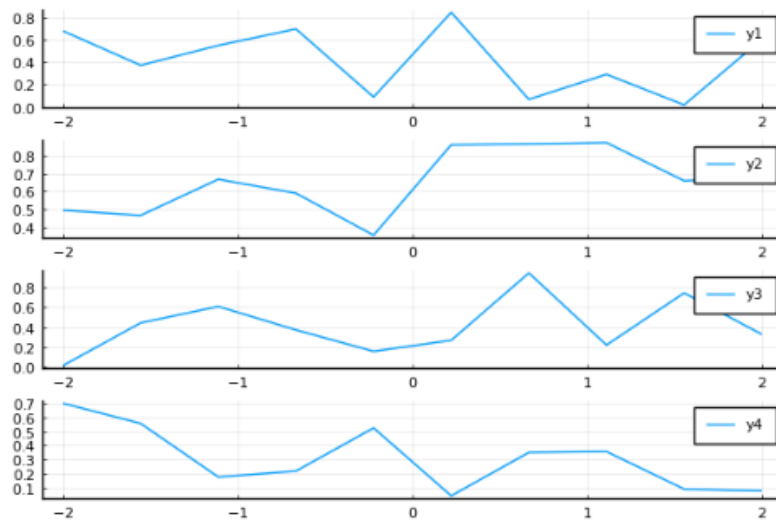
```



Рис. 3.30: Примеры. Использование пакета Distributions. ч.3

Подграфики

```
# загружаем pyplot():
pyplot()
# построение серии графиков:
x=range(-2,2,length=10)
y = rand(10,4)
plot(x,y,
      layout=(4,1)
)
```



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

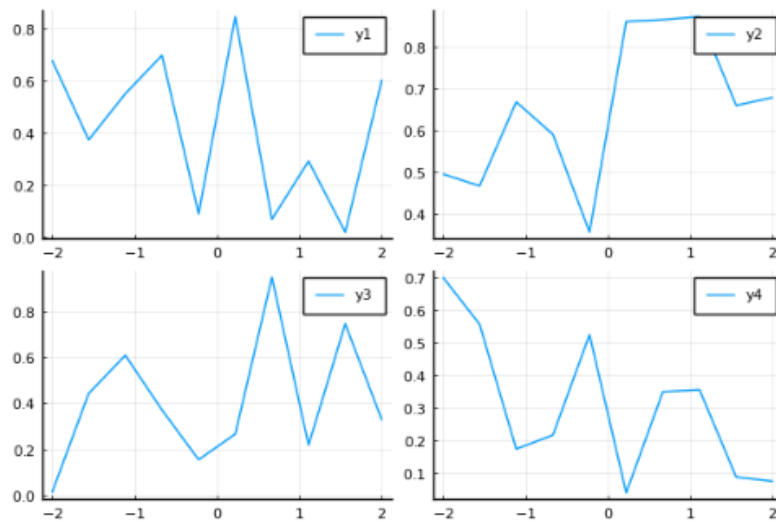


Рис. 3.31: Примеры. Подграфики. ч.1

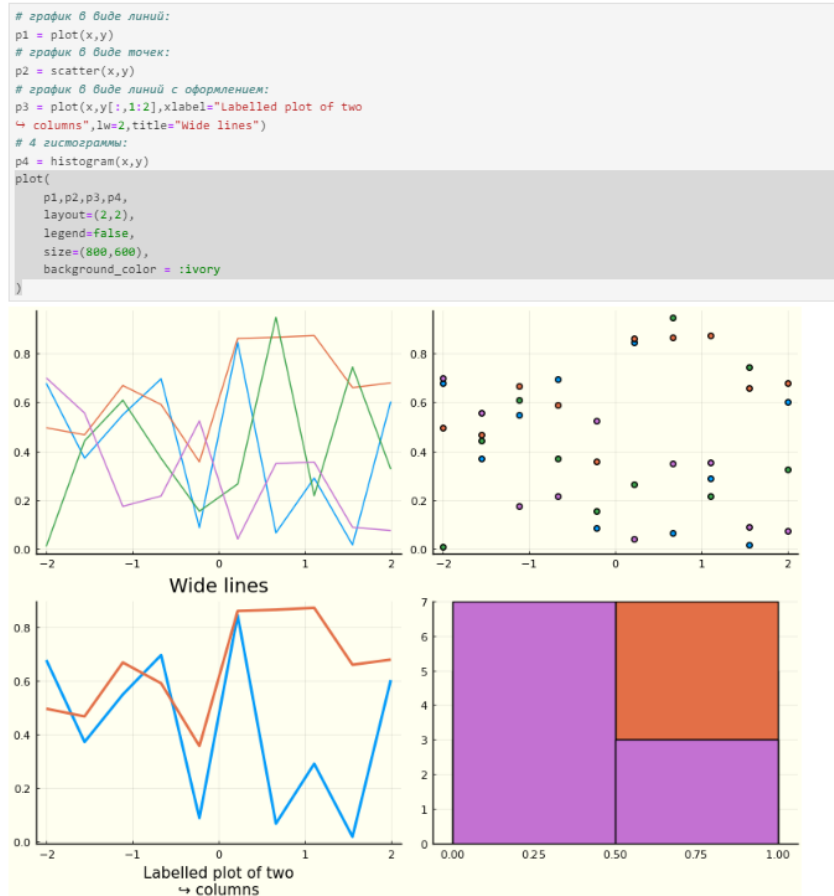


Рис. 3.32: Примеры. Подграфики. ч.2

```

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
)

```

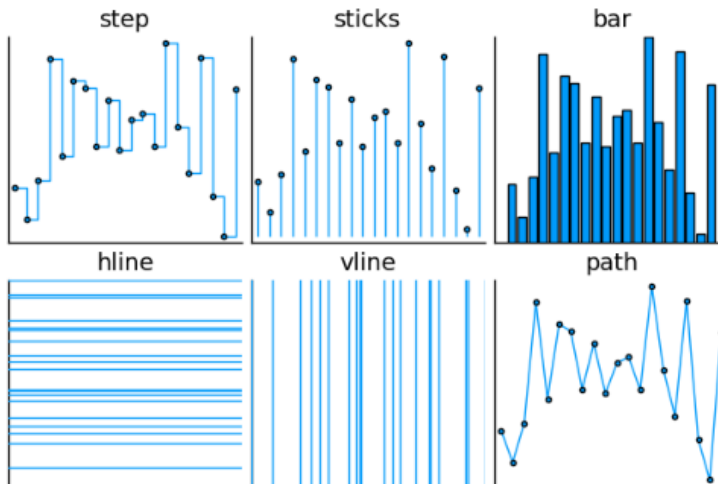


Рис. 3.33: Примеры. Подграфики. ч.3

```

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)
)

```

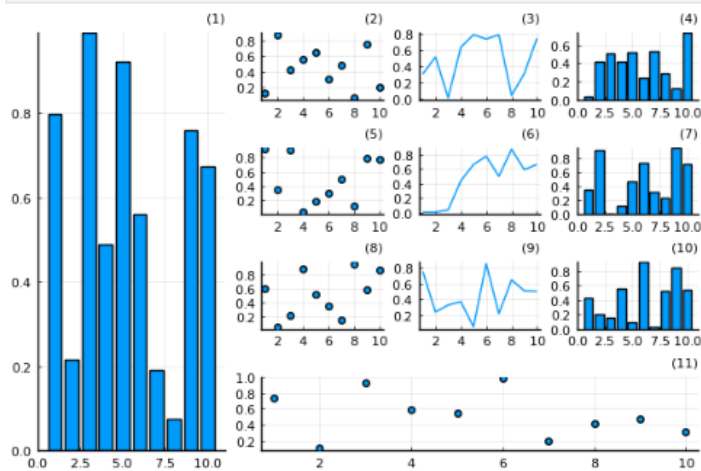


Рис. 3.34: Примеры. Подграфики. ч.4

3.2 Самостоятельная работа

Примеры представлены на рис. 3.35 - 3.45. Сохраненные файлы по заданиям располагаются в смежном каталоге lab5+samost/samost.

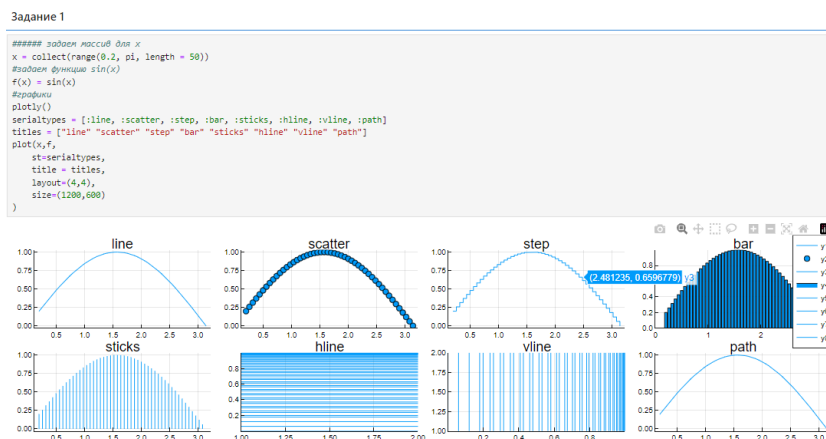


Рис. 3.35: Задание 1

Задание 2

```
titles = ["dot" "dashdot" "dash" "solid"]  
  
p1=plot(x,f, linestyle=:dot)  
p2=plot(x,f, linestyle=:dashdot)  
p3=plot(x,f, linestyle=:dash)  
p4=plot(x,f, linestyle=:solid)  
plot(p1,p2,p3,p4,  
      title = titles,  
      layout=(2,2)  
)
```

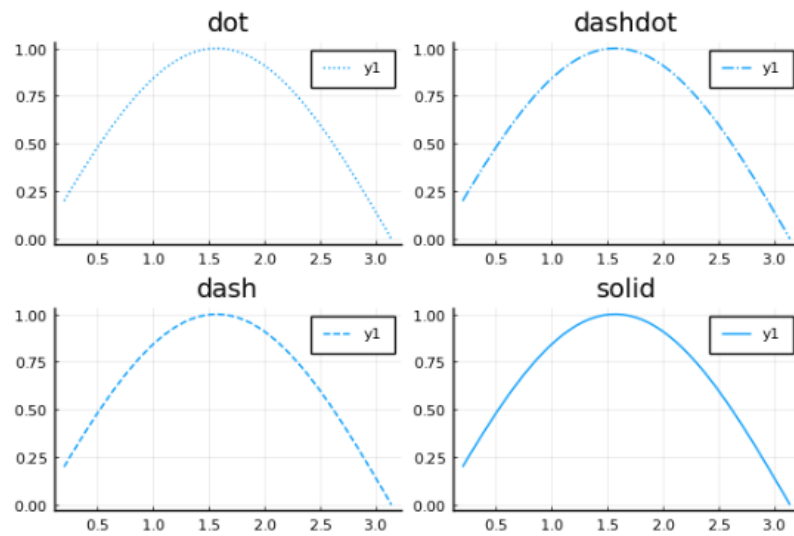


Рис. 3.36: Задание 2

Задание 3

```
#задаем шрифт надписей
using LaTeXStrings
#задаем функцию
x=collect(range(0,10,100))
f(x) = pi*x^2*log(x)
plot(x,f,
      xlabel="x",
      ylabel=L"\pi*x^2*\ln{x}",
      background_color_outside = :limegreen,
      linecolor = :red,
      xticks=(0:1:10),
      yticks=(0:100:1000)
)
```

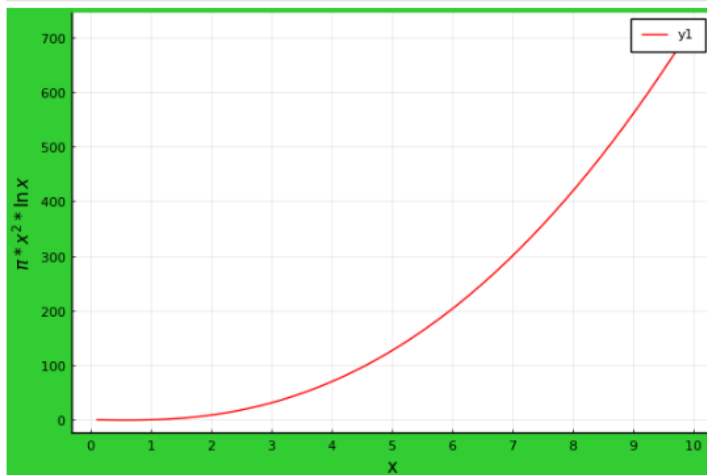
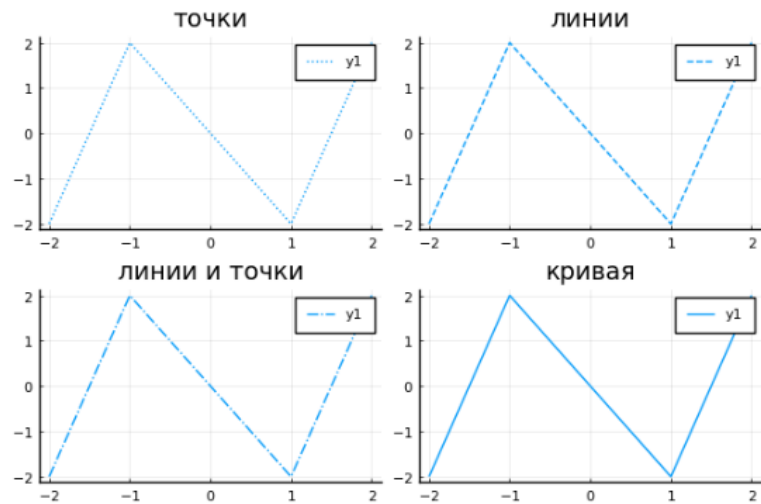


Рис. 3.37: Задание 3

Задание 4

```
x = [-2, -1, 0, 1, 2]
f(x) = x^3-3*x
titles = ["точки" "линии" "линии и точки" "кривая"]
p1=plot(x,f, line=:dot)
p2=plot(x,f, line=:dash)
p3=plot(x,f, line=:dashdot)
p4=plot(x,f, line=:solid)
plot(p1,p2,p3,p4,
     title = titles,
     layout=(2,2)
    )
savefig("figure_tazaeva.png")
```



"C:\\Users\\noname\\figure_tazaeva.png"

Рис. 3.38: Задание 4

Задание 5

```
x=collect(3:0.1:6)
y1(x)=pi*x
y2(x)=exp(x)*cos(x)
plot(x,[y1,y2],legend=true,grid=true, layout=(1,2), size=(1000,400))
```

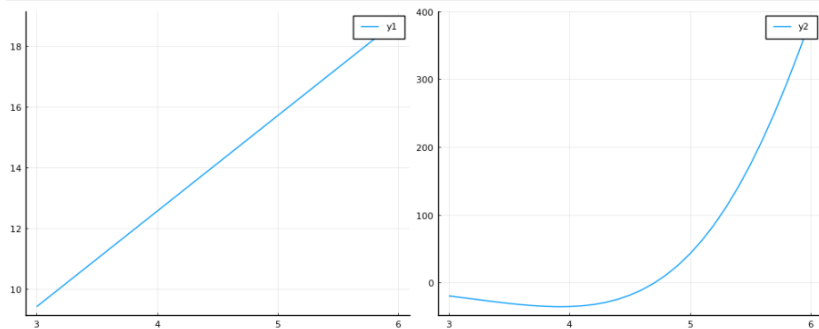


Рис. 3.39: Задание 5

Задание 6

```
sds = [1, 1/2, 1/4, 1/8, 1/16, 1/32]
n = 10
y = [mean(sd*randn(n)) for sd in sds]
errs = 2.45 * sds / n*0.01
plot(y,
      ylims = (-1,1),
)
plot(y,
      ylims = (-1,1),
      err = errs
)
```

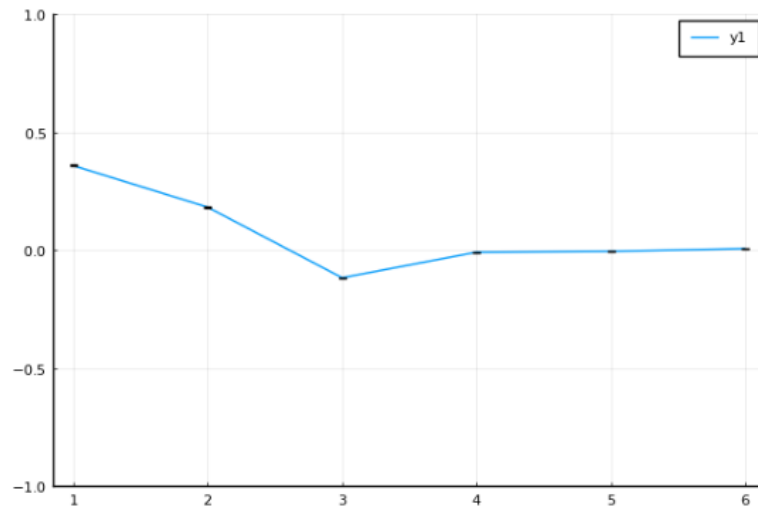


Рис. 3.40: Задание 6

Задание 7

```
n = 50
x = rand(1:20, 20)
y = rand(1:20, 20)
scatter(x,y,
        legend=true,
        label = "точки",
        xlabel="x",
        ylabel="y",
        title="Задание 7. Точечный график случайных чисел")
```



Рис. 3.41: Задание 7

Задание 8

```
n = 50
x = rand(1:20, 20)
y = rand(1:20, 20)
z = rand(1:20, 20)
scatter(x,y,z,
        legend=true,
        label = "точки",
        xlabel="x",
        ylabel="y",
        title="Задание 8. 3D - Точечный график случайных чисел")
```

Задание 8. 3D - Точечный график случайных чисел

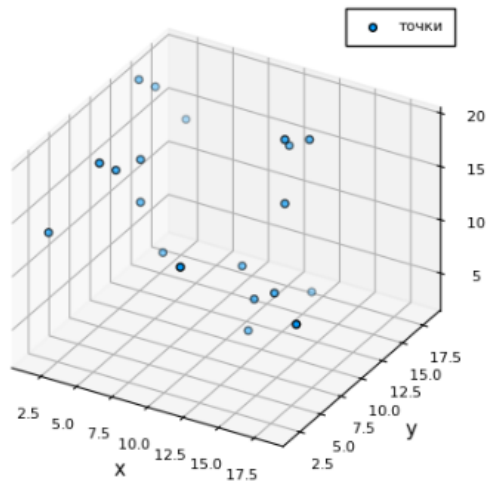


Рис. 3.42: Задание 8

Задание 9

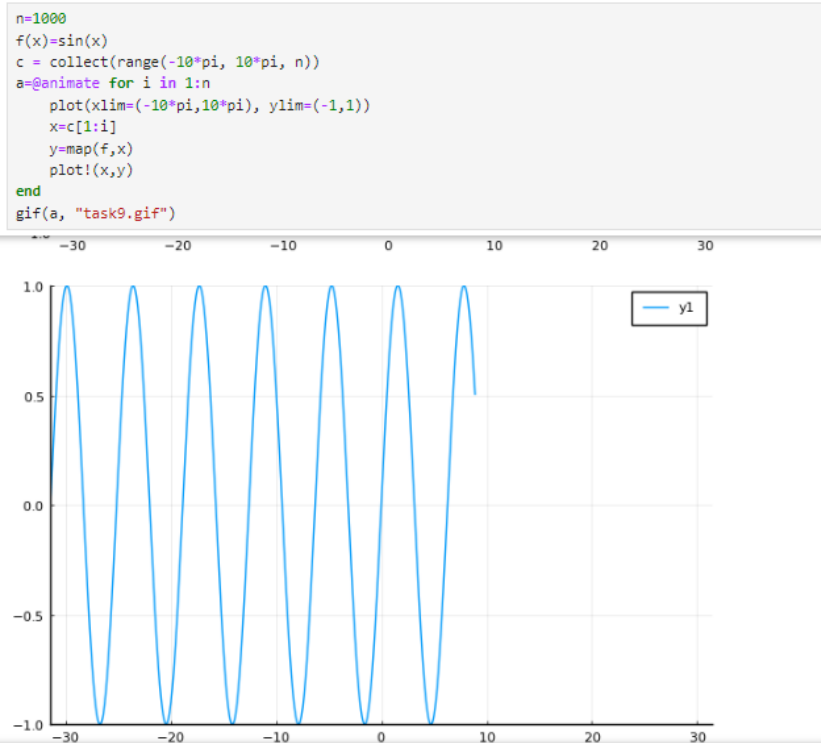


Рис. 3.43: Задание 9

```

# радиус малой окружности:
rr = 0.5
# коэффициент для построения большой окружности:
k = 3
# число отсчётов:
n = 100
# массив значений угла θ:
# theta from 0 to 2π ( + a little extra)
θ = collect(0:2*π/100:2*π+2*π/100)
# массивы значений координат:
X = rr*k*cos.(θ)
Y = rr*k*sin.(θ)
a=@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)
    i = 50
    t = θ[1:i]
    # гипотенуза:
    x = rr*(k-1)*cos.(t) + rr*cos.((k-1)*t)
    y = rr*(k-1)*sin.(t) - rr*sin.((k-1)*t)
    plot!(x,y, c=:red)
    # малая окружность:
    xc = rr*(k-1)*cos(t[end]) .+ rr*cos.(θ)
    yc = rr*(k-1)*sin(t[end]) .+ rr*sin.(θ)
    plot!(xc,yc,c=:black)
    # радиус малой окружности:
    x1 = transpose([rr*(k-1)*cos(t[end]) x[end]])
    y1 = transpose([rr*(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)
end
gif(a,"task10.gif")

```

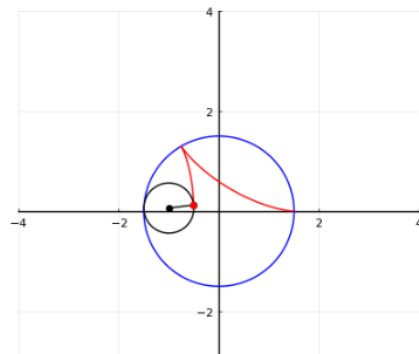


Рис. 3.44: Задание 10

Задание 11

```

# радиус малой окружности:
rr = 1
# коэффициент для построения большой окружности:
k = 5
# число отсчётов:
n = 100
# массив значений угла θ:
# theta from 0 to 2pi ( + a little extra)
θ = collect(0:2π/100:2π+2π/100)
# массивы значений координат:
X = rr*k*cos.(θ)
Y = rr*k*sin.(θ)
a=animate for i in 1:n
    # задаём оси координат:
    plt=plot(5,xlim=(-10,10),ylim=(-10,10), c=:red, aspect_ratio=1, legend=false, framestyle=:origin)
    # большая окружность:
    plot!(plt, X,Y, c=:blue, legend=false)
    i = 50
    t = θ[1:i]
    # эволюционда:
    x = rr*(k-1)*cos.(t) + rr*cos.((k-1)*t)
    y = rr*(k-1)*sin.(t) - rr*sin.((k-1)*t)
    plot!(x,y, c=:red)
    # малая окружность:
    xc = rr*(k-1)*cos(t[end]) + rr*cos.(θ)
    yc = rr*(k-1)*sin(t[end]) + rr*sin.(θ)
    plot!(xc,yc,c=:black)
    # радиус малой окружности:
    x1 = transpose([rr*(k-1)*cos(t[end]) x[end]])
    y1 = transpose([rr*(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)
end
gif(a,"task11.gif")

```

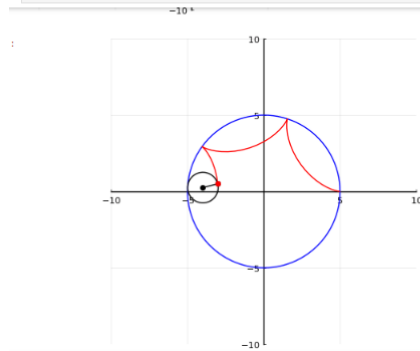


Рис. 3.45: Задание 11

4 Выводы

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