

Лабораторная работа № 5

Построение графиков

Герра Гарсия Паола Валентина

Содержание

Цель работы	2
Задание	2
Теоретическое введение	2
Выполнение лабораторной работы	2
Задания для самостоятельного выполнения	12
Выводы	16
Список литературы.....	17

Список иллюстраций

Основные пакеты для работы с графиками в Julia	3
Основные пакеты для работы с графиками в Julia	3
Опции при построении графика	4
Опции при построении графика	4
Точечный график	5
Точечный график	5
Аппроксимация данных.....	6
Полярные координаты.....	6
График поверхности	7
Линии уровня	7
Векторные поля	8
Анимация	8
Гипоциклоида.....	9
Errorbars	9
Errorbars	10
Использование пакета Distributions	10
Подграфики	11
Подграфики	11
Подграфики	12
Задание №1 и №2	12
Задание №3	13
Задание №4	13
Задание №5	14

Задание №6	14
Задание №7 и №8	15
Задание №9	15
Задание №10.....	15
Задание №10.....	16
Задание №11.....	16
Задание №11.....	16

Цель работы

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

Задание

1. Используя JupyterLab, повторите примеры. При этом дополните графики обозначениями осей координат, легендой с названиями траекторий, названиями графиков и т.п.
2. Выполните задания для самостоятельной работы.

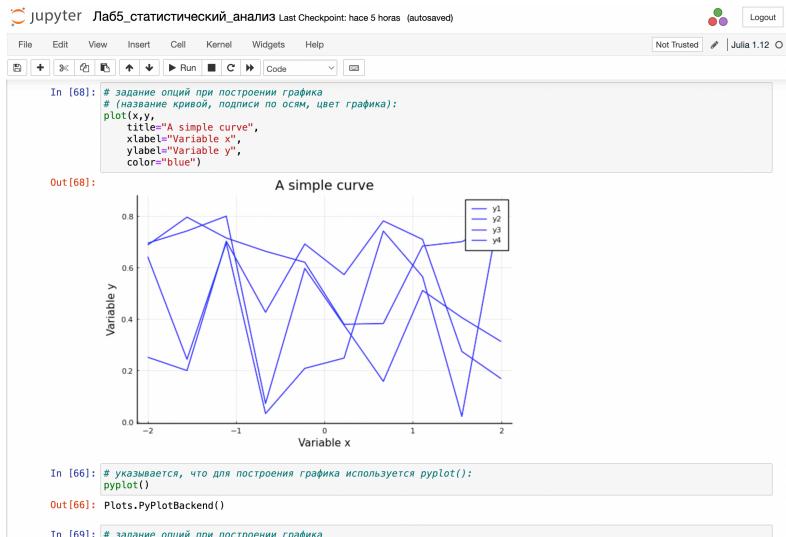
Теоретическое введение

Julia – высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений [[@julialang](#)]. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

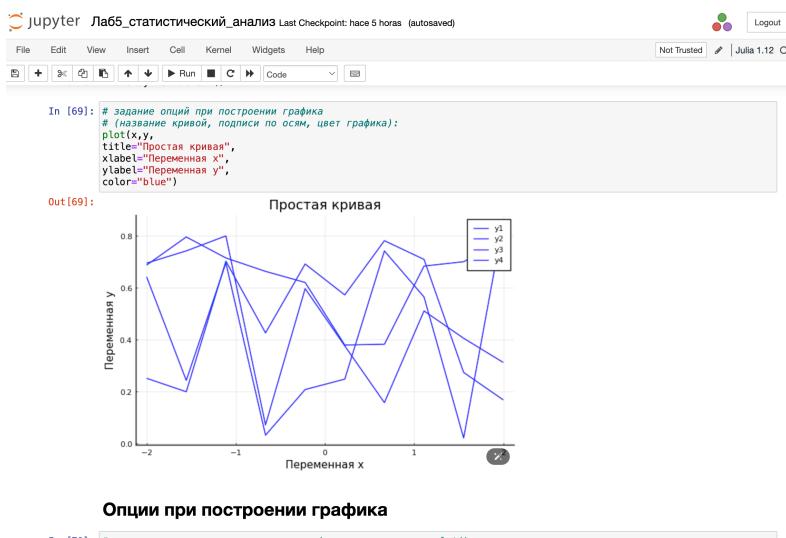
Для выполнения заданий была использована официальная документация Julia [[@juliadoc](#)].

Выполнение лабораторной работы

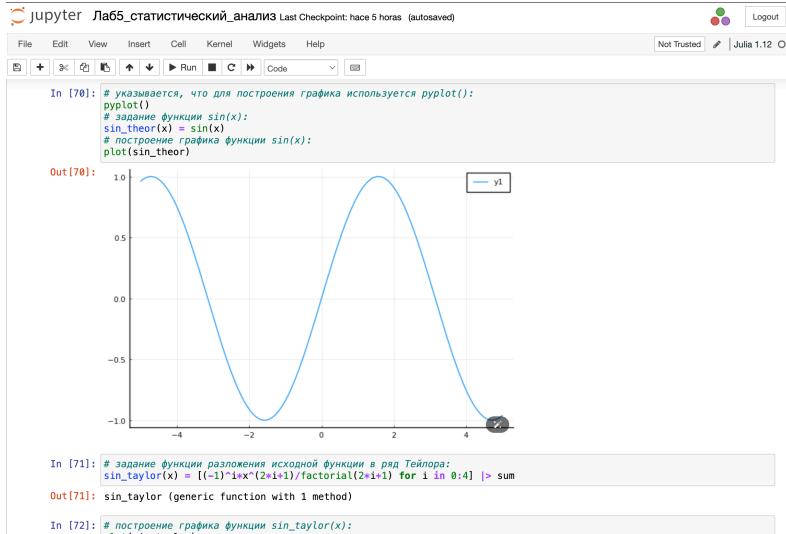
Выполним примеры из лабораторной работы для знакомства с пакетами по отрисовки графиков и их функциями (рис. [-@fig:001]-[-@fig:019]).



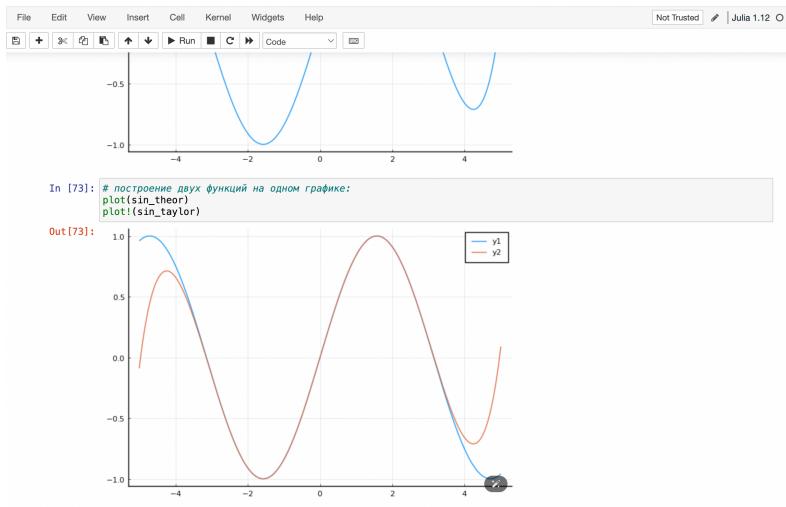
Основные пакеты для работы с графиками в Julia



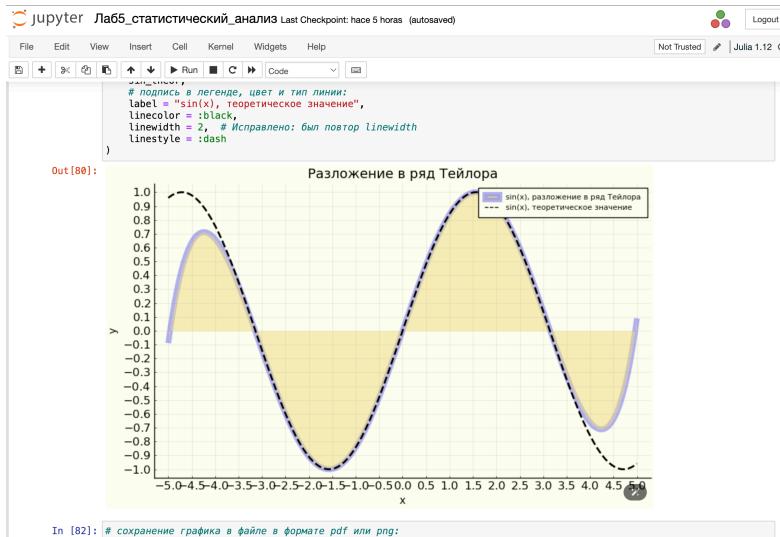
Основные пакеты для работы с графиками в Julia



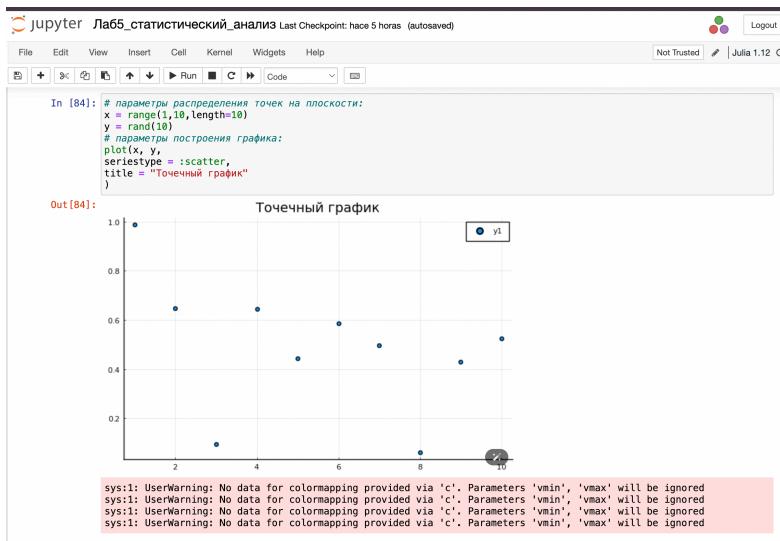
Опции при построении графика



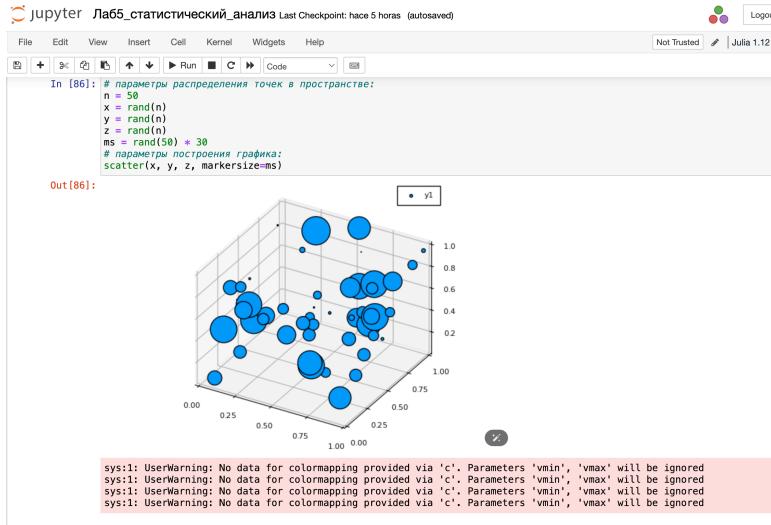
Опции при построении графика



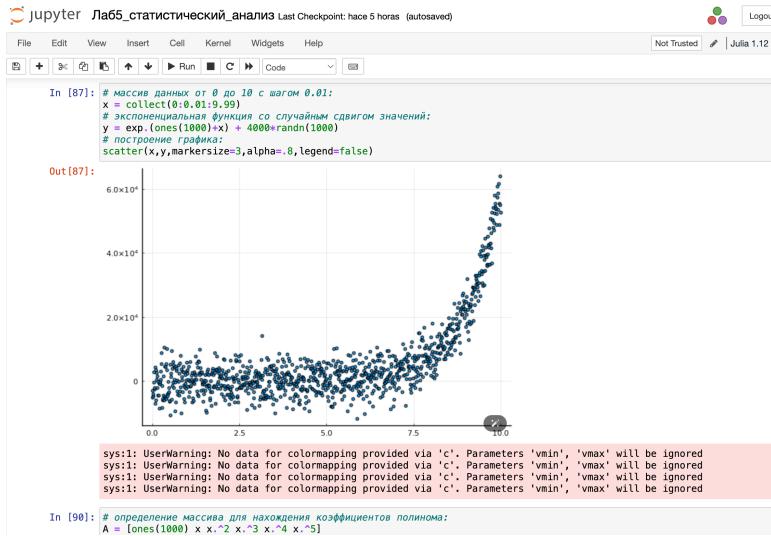
Точечный график



Точечный график



Аппроксимация данных



Полярные координаты

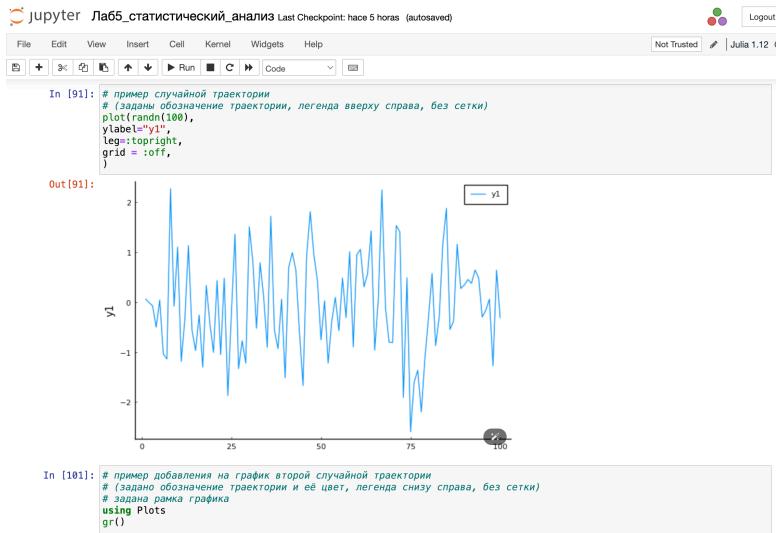
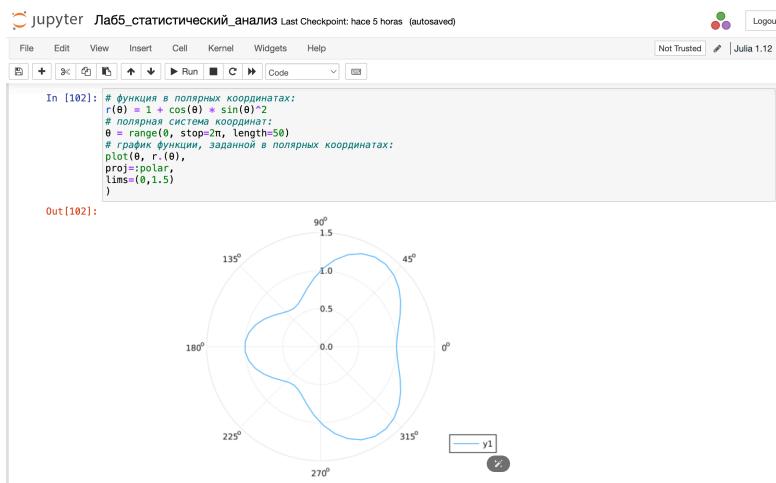
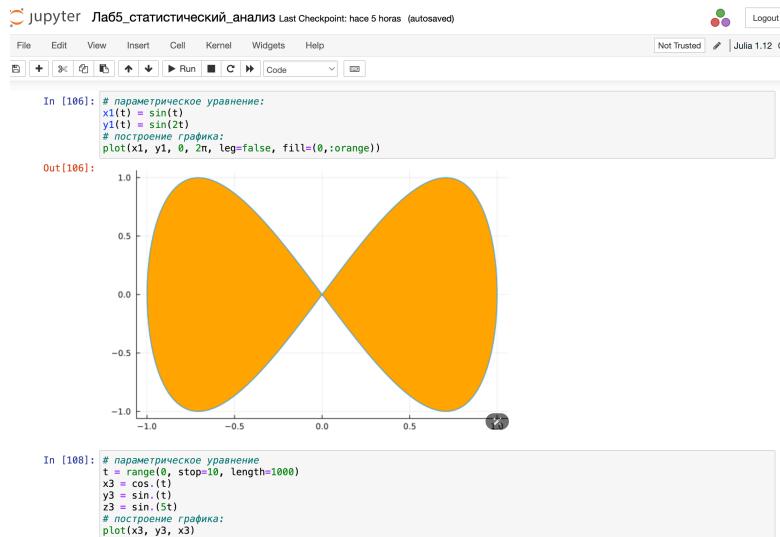


График поверхности

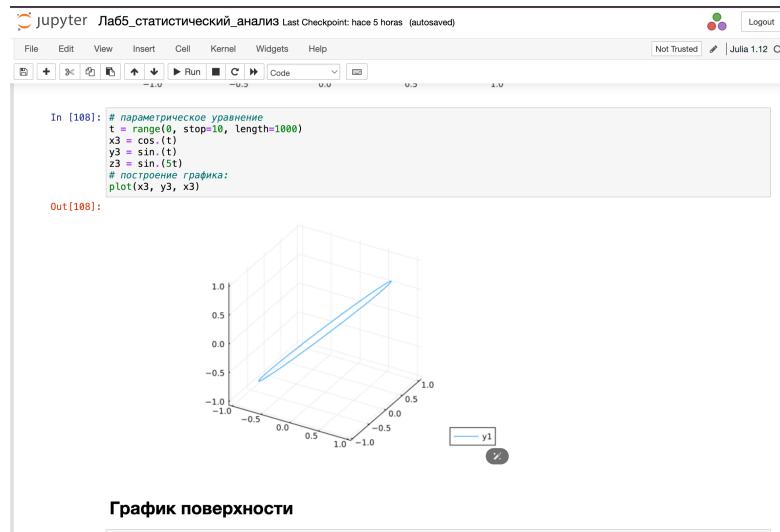


Параметрический график кривой на плоскости

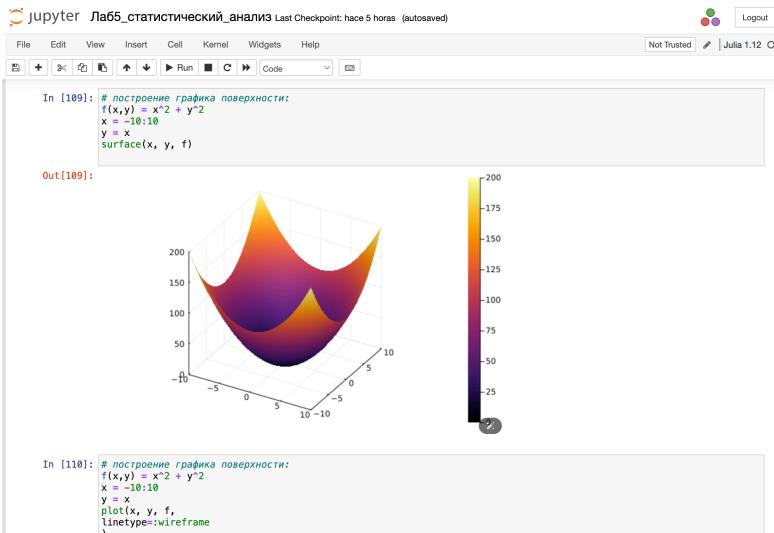
Линии уровня



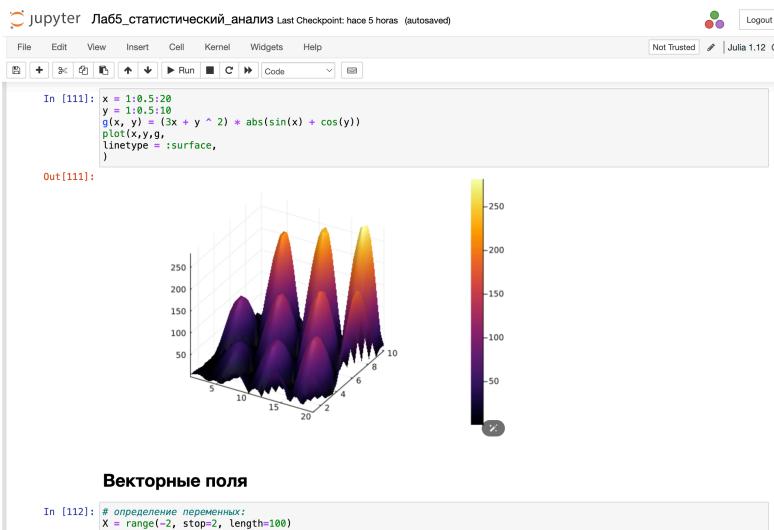
Векторные поля



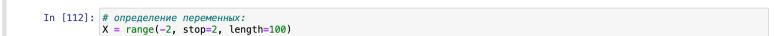
Анимация



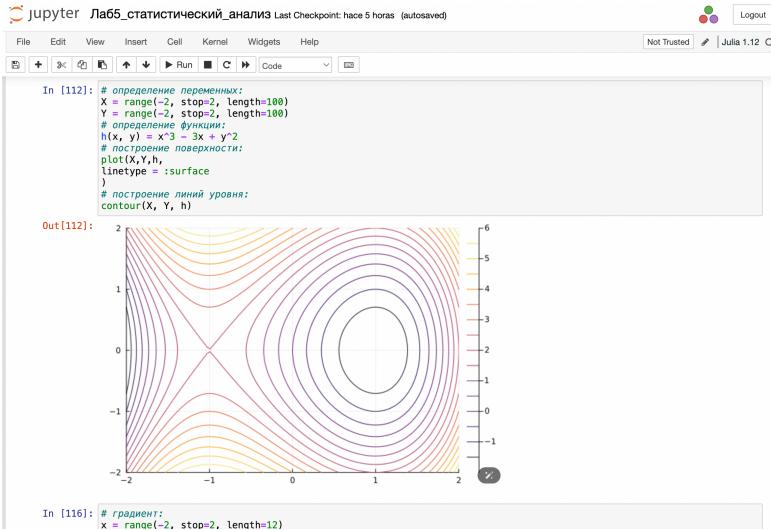
Гипоциклоида



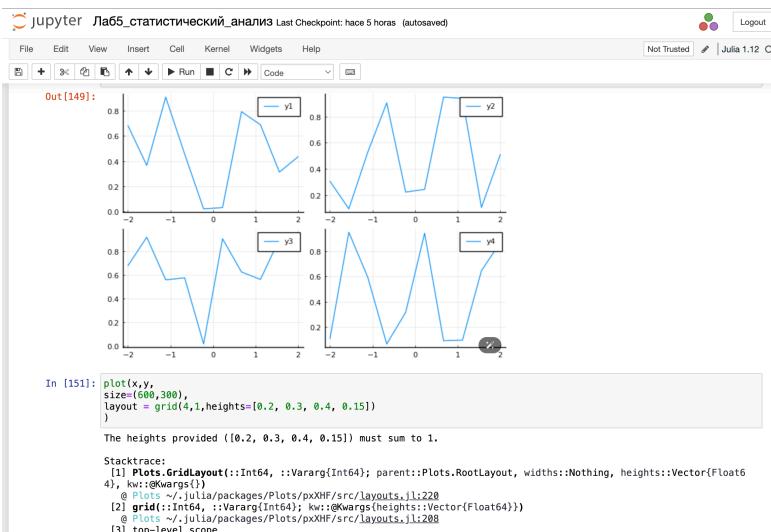
Векторные поля



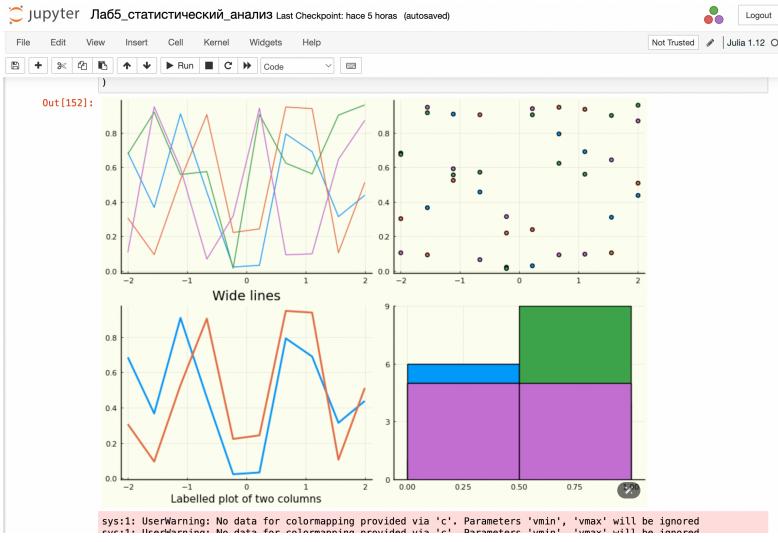
Errorbars



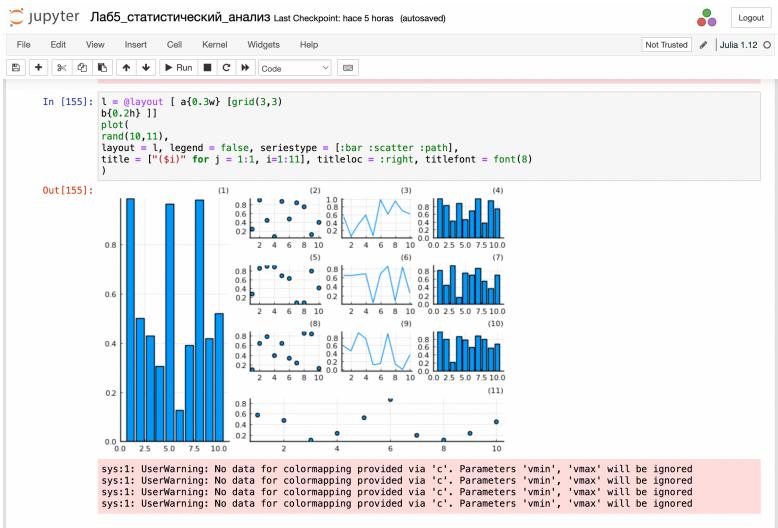
Errorbars



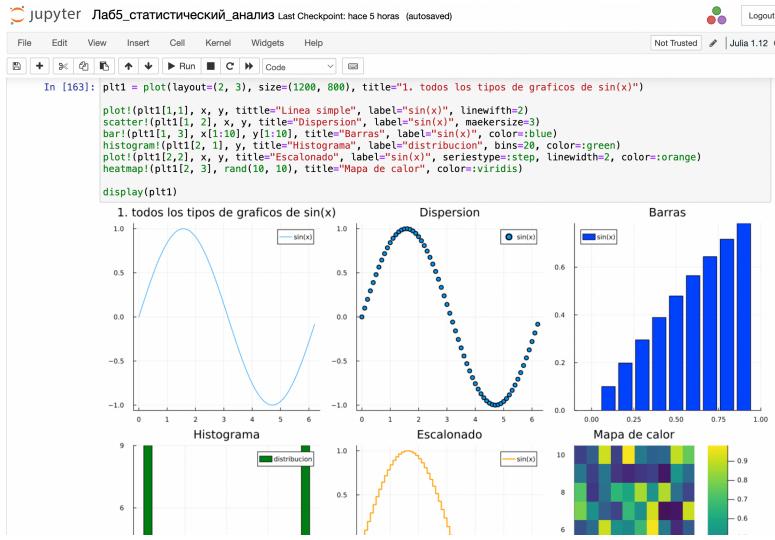
Использование пакета Distributions



Подграфики



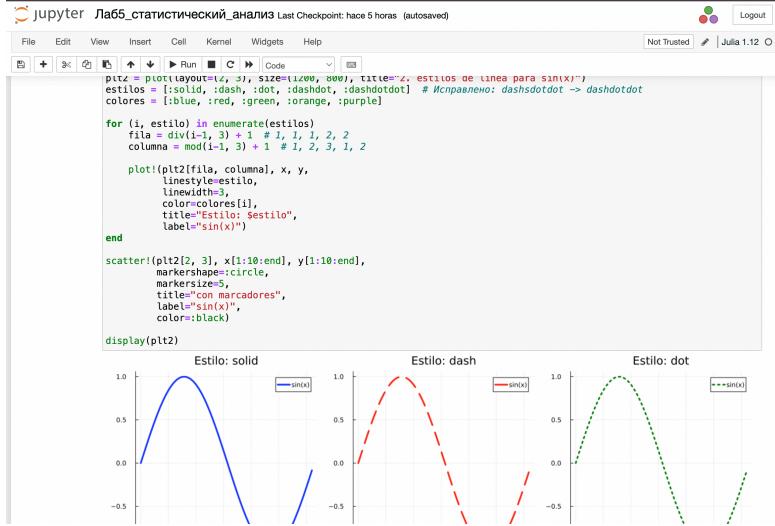
Подграфики



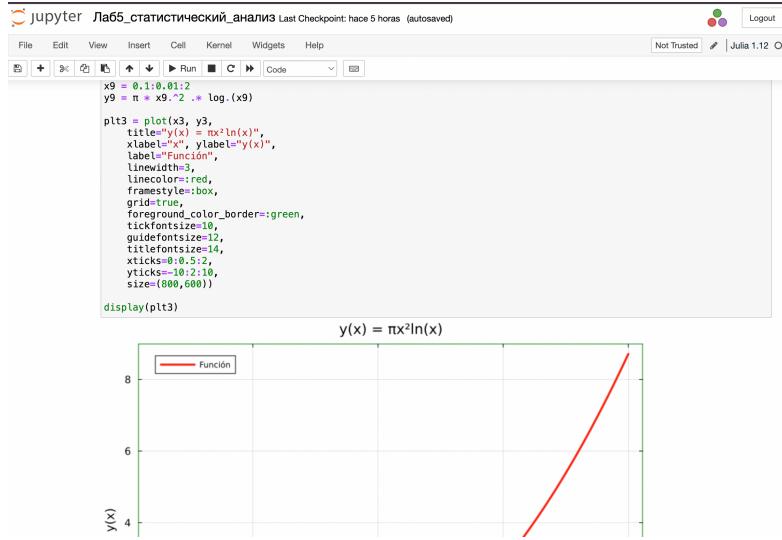
Подграфики

Задания для самостоятельного выполнения

Выполним задания (рис. [-@fig:020]-[-@fig:030]).



Задание №1 и №2



Задание №3

Jupyter Lab65_статистический_анализ Last Checkpoint: hace 5 horas (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Julia 1.12 O

```
#4 cuatro tipos de graficos en subplots
x6 = [-2, -1, 0, 1, 2]
y6 = x6.^6 .-3x6

plt4 = plot(layout=(2, 2), size=(1000, 800), title="4 graficos de y = x^3")

#Puntos
scatter!(plt4[1, 1], x6, y6, title="Puntos", label="", markersize=8, color=:blue)

#lineas
plot!(plt4[1, 2], x6, y6, title="lineas", label="", linewidth=2, color=:red)

#lineas y puntos
plot!(plt4[2, 1], x6, y6, title="lineas y puntos", label="", linewidth=2, color=:green, marker=:circle, markersize=6

#curva suave
xs = range(-2, 2, length=100)
ys = xs.^3 .-3xs
plot!(plt4[2, 2], xs, ys, title="curva suave", label="", linewidth=3, color=:purple)

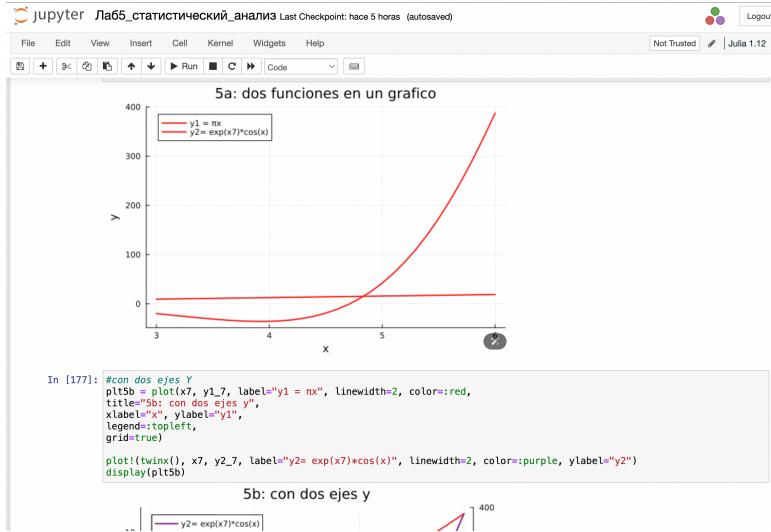
savefig(plt4, "figure_paola.png")
println("grafico 4 guardado como figure_paola.png")

grafico 4 guardado como figure_paola.png

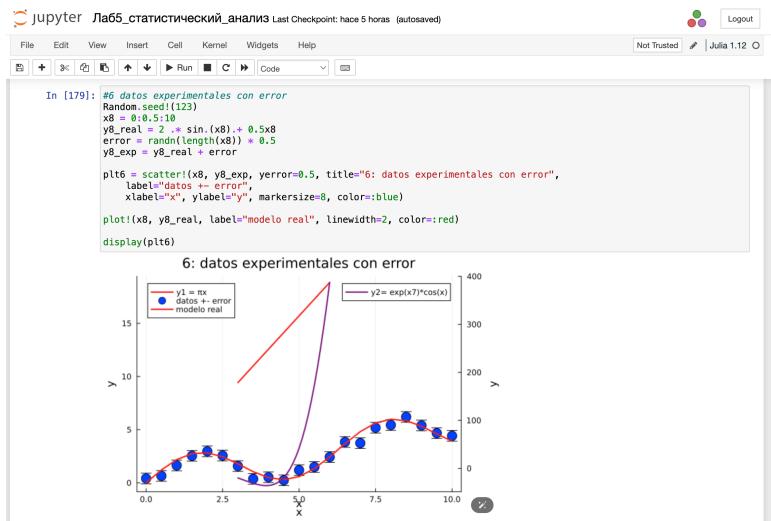
#5 dos graficos en uno
x7 = 3:0.1:6
y1_7 = π * x7
y2_7 = exp.(x7).*cos.(x7)

#grafico simple
plt5a = plot(x7, y1_7, label="y1 = nx", linewidth=2, color=:red,
title="5: dos funciones en un grafico",
xlabel="x", ylabel="y",
legend=:topleft,
grid=true)
```

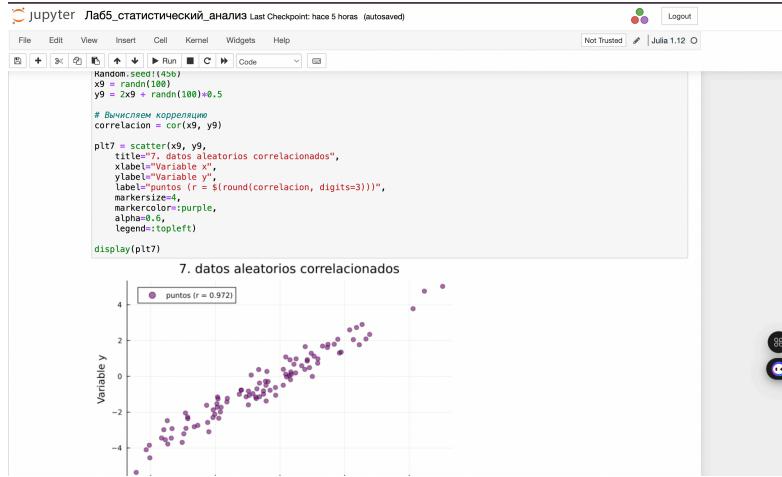
Задание №4



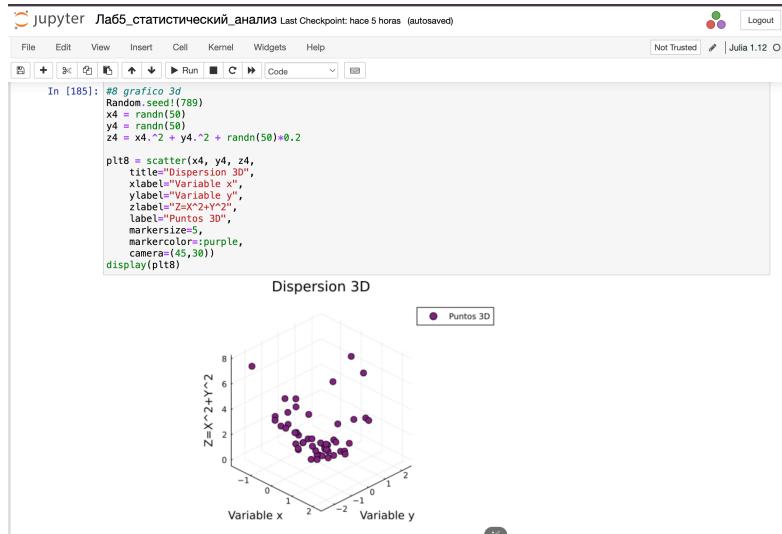
Задание №5



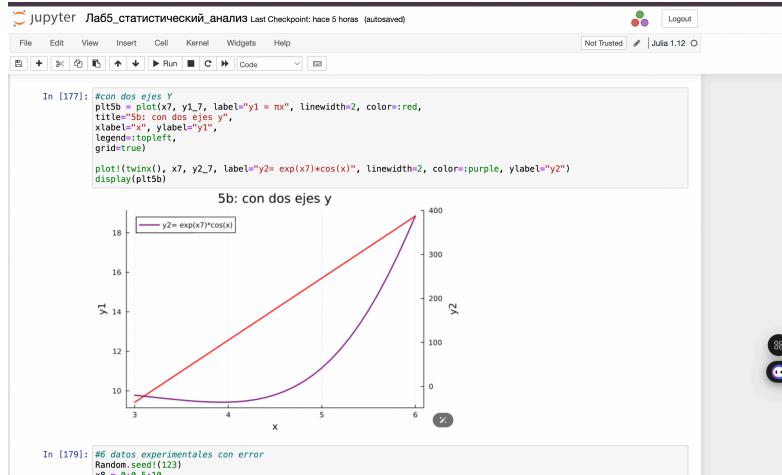
Задание №6



Задание №7 и №8



Задание №9



Задание №10

```
In [186]: #9 animacion de sin(x)
anim = @animate for i in 1:50
    x_anim = range(0, 2*pi*i/50), length=100
    y_anim = sin.(x_anim)

    plot(x_anim, y_anim,
        title="animacion: sin(x)",
        xlabel="x",
        ylabel="sin(x)",
        label="",
        xlims=(0, 2pi),
        ylims=(-1.2, 1.2),
        linewidth=3,
        color=:pink)
end

gif(anim, "sin_animation.gif", fps=15)
println("animacion guardada como sin_animation.gif")

[ Info: Saved animation to /Users/paovalentinag10/sin_animation.gif
animacion guardada como sin_animation.gif
```

Задание №10

```
Jupyter Lab6_статистический_анализ Last Checkpoint: hace 5 horas (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted Julia 1.12 O

In [189]: #10 hipocicloide animada
function hipocicloide(R, r, t)
    x = (R - r)*cos.(t) + r*cos.((R - r)/r * t)
    y = (R - r)*sin.(t) - r*sin.((R - r)/r * t)
    return x, y
end

println("\nCreando animación de hipocicloide...")
valores_k = [3, 4, 5/2, 7/3] # 2 enteros, 2 racionales
R = 5

anim_h = @animate for k in valores_k
    r = R/k
    t = range(0, 2pi*k, length=500)
    x_h, y_h = hipocicloide(R, r, t)

    plot(x_h, y_h,
        title="hipocicloide k = $k",
        xlabel="",
        aspect_ratio=:equal,
        linewidth=2,
        color=:red,
        xlims=(-R-1, R+1),
        ylims=(-R-1, R+1))
end

gif(anim_h, "hipocicloide.gif", fps=2)
println("Hipocicloide guardada como hipocicloide.gif")

[ Info: Saved animation to /Users/paovalentinag10/hipocicloide.gif
Creando animación de hipocicloide...
Hipocicloide guardada como hipocicloide.gif
```

In [192]: #11 epicicloide animada

Задание №11

```
Jupyter Lab6_статистический_анализ Last Checkpoint: hace 5 horas (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Not Trusted Julia 1.12 O

function epicicloide(R, r, t)
    x = (R + r)*cos.(t) - r*cos.((R + r)/r * t)
    y = (R + r)*sin.(t) - r*sin.((R + r)/r * t)
    return x, y
end

anim_e = @animate for k in valores_k
    r = R/k
    t = range(0, 2pi*k, length=500)
    x_e, y_e = epicicloide(R, r, t)

    plot(x_e, y_e,
        title="Epicicloide k = $k",
        xlabel="",
        aspect_ratio=:equal,
        linewidth=2,
        color=:red,
        xlims=(-2R-1, 2R+1),
        ylims=(-2R-1, 2R+1))
end

gif(anim_e, "epicicloide.gif", fps=2)
println("Epicicloide guardada como epicicloide.gif")

print("archivos guardados:")
print("- figure_pola.png (4)")
print("- sin_animation.gif(9)")
print("- hipocicloide.gif(10)")
print("- epicicloide.gif(11)")

[ Info: Saved animation to /Users/paovalentinag10/epicicloide.gif
Epicicloide guardada como epicicloide.gif
archivos guardados:
- figure_pola.png (4)
- sin_animation.gif(9)
- hipocicloide.gif(10)
- epicicloide.gif(11)
```

Задание №11

Выводы

В результате выполнения данной лабораторной работы я освоила синтаксис языка Julia для построения графиков.

Список литературы