|  |
| --- |
| **Sahifa 1** |

|  |
| --- |
| **2. sahifa** |

**UDC 004.6**

**BBC: 32.973**

Devpractice jamoasi. Python. Ma'lumotlarni vizualizatsiya qilish. Matplotlib. Seaborn. Mayavi. - devpractice.ru.

2020. - 412 p.: Kasal.

Ushbu kitob tilda ma'lumotlarni vizualizatsiya qilish uchun kutubxonalarga qaratilgan

*Python* dasturlash : *Matplotlib* , *Seaborn* , *Mayavi* . Har bir kutubxonada mavjud

ma'lumotlarni vizualizatsiya qilish vositalarini batafsil tavsifi, tashqi o'rnatish uchun vositalar

grafiklarning turi va joylashuvi.

UDC 004.6

BBC: 32.973

*Materiallar quyidagilar tomonidan tuzilgan va tayyorlangan* :

Abdraxmanov M.I.

Barcha huquqlar himoyalangan. Ushbu kitobning biron bir qismini hech birida ko'paytirish mumkin emas

yozma ruxsatisiz har qanday shaklda va har qanday usulda

mualliflik huquqi egalari.

Kitobda keltirilgan materiallar ko'p marta tekshirilgan. Ammo inson bo'lganidan beri

va texnik xatolar hali ham mumkin, *devpractice.ru* loyihasi mualliflari va jamoasi *bunga* toqat qilmaydilar

foydalanish bilan bog'liq mumkin bo'lgan xatolar va oqibatlar uchun javobgarlik

ushbu kitobdan materiallar.

© devpractice.ru, 2020 yil

© Abdraxmanov M.I., 2020 yil

1

|  |
| --- |
| **3. sahifa** |

**Mundarija**

[*Matplotlib* kutubxonasi I. qism](https://translate.googleusercontent.com/translate_f#7).................................................. ......................................... 6

[Kirish ................................................. .................................................. ........................ 6](https://translate.googleusercontent.com/translate_f#7)

[1-bob. Tezkor boshlash ............................................. .................................................. .... 7](https://translate.googleusercontent.com/translate_f#8)

[1.1 O'rnatish ................................................ .................................................. .............. 7](https://translate.googleusercontent.com/translate_f#8)

[1.1.1](https://translate.googleusercontent.com/translate_f#8)*Matplotlib-*[ni o'rnatish parametrlari.................................................. .................... 7](https://translate.googleusercontent.com/translate_f#8)

1.1.2 [*Menejer*](https://translate.googleusercontent.com/translate_f#8) yordamida [*Matplotlib-ni* o'rnatish](https://translate.googleusercontent.com/translate_f#8)*pip*[.......................................... 7](https://translate.googleusercontent.com/translate_f#8)

[1.1.3 O'rnatishni tekshirish ............................................. .......................................... 7](https://translate.googleusercontent.com/translate_f#8)

[1.2 Tez boshlash ............................................... .................................................. ....... 8](https://translate.googleusercontent.com/translate_f#9)

[1.3 Belgilash ............................................... ............................................. o'n](https://translate.googleusercontent.com/translate_f#11)

[1.4 Bitta maydon bo'yicha bir nechta diagrammalar ............................................ ....................... o'n bir](https://translate.googleusercontent.com/translate_f#12)

[1.5 Turli sohalardagi grafikalar taqdimoti ............................................ ........... 12](https://translate.googleusercontent.com/translate_f#13)

[1.6 Kategorik ma'lumotlar jadvalini tuzish ... 14](https://translate.googleusercontent.com/translate_f#15)

[1.7 Jadvalning asosiy elementlari .............................................. ............................... 15](https://translate.googleusercontent.com/translate_f#16)

[2-bob. Modul bilan ishlash asoslari](https://translate.googleusercontent.com/translate_f#20)*piplot*[................................................. ................. o'n to'qqiz](https://translate.googleusercontent.com/translate_f#20)

[2.1 Grafiklarni qurish ............................................... ......................................... o'n to'qqiz](https://translate.googleusercontent.com/translate_f#20)

[2.2 Grafikdagi matn yorliqlari ............................................. ............................. 21](https://translate.googleusercontent.com/translate_f#22)

[2.2.1 Akslarning nomi ............................................. ........................................ 21](https://translate.googleusercontent.com/translate_f#22)

[2.2.2 Grafika sarlavhasi ............................................. .......................................... 22](https://translate.googleusercontent.com/translate_f#23)

[2.2.3 Matnli eslatma ............................................. .................................... 23](https://translate.googleusercontent.com/translate_f#24)

[2.2.4 Afsona .............................................. .................................................. .......... 23](https://translate.googleusercontent.com/translate_f#24)

[2.3 Chiziqli diagramma bilan ishlash ............................................. ............................... 24](https://translate.googleusercontent.com/translate_f#25)

[2.3.1 Uchastka chizig'i uslubi ............................................ ....................................... 24](https://translate.googleusercontent.com/translate_f#25)

[2.3.2 Chiziq rangi ............................................. .................................................. ..... 27](https://translate.googleusercontent.com/translate_f#28)

[2.3.3 Grafik turi ............................................. .................................................. ... 28](https://translate.googleusercontent.com/translate_f#29)

[2.4 Grafiklarni bir-biridan alohida joylashtirish ........................................... ...... o'ttiz](https://translate.googleusercontent.com/translate_f#31)

[2.4.1 Subplot () funktsiyasi bilan ishlash ......................................... ............................... o'ttiz](https://translate.googleusercontent.com/translate_f#31)

[2.4.2 Subplots () funktsiyasi bilan ishlash ......................................... ............................. 33](https://translate.googleusercontent.com/translate_f#34)

[3-bob. Jadval elementlarini sozlash ............................................ .......................... 34](https://translate.googleusercontent.com/translate_f#35)

[3.1 Afsona bilan ishlash .............................................. ................................................. 34](https://translate.googleusercontent.com/translate_f#35)

[3.1.1 Afsonani namoyish qilish ............................................. .................................... 34](https://translate.googleusercontent.com/translate_f#35)

[3.1.2 Afsonaning jadvaldagi joylashuvi ........................................... ................. 36](https://translate.googleusercontent.com/translate_f#37)

[3.1.3 Afsonani o'rnatish uchun qo'shimcha parametrlar ...................................... 38](https://translate.googleusercontent.com/translate_f#39)

[3.2 Grafiklarning joylashuvi ............................................... ......................................... 40](https://translate.googleusercontent.com/translate_f#41)

3.2.1 GridSpec vositasi ............ [................................. ...................................... 40](https://translate.googleusercontent.com/translate_f#41)

[3.3 Grafikning matnli elementlari .............................................. .............................. 45](https://translate.googleusercontent.com/translate_f#46)

[3.3.1 Rasm sarlavhasi va diagramma maydonlari .......................................... .................... 47](https://translate.googleusercontent.com/translate_f#48)

[3.3.2 Diagramma o'qi yorliqlari ............................................ ..................................... 48](https://translate.googleusercontent.com/translate_f#49)

[3.3.3 Matn bloki ............................................. ................................................ 50](https://translate.googleusercontent.com/translate_f#51)

[3.3.4 Izohlar .............................................. .................................................. ..... 52](https://translate.googleusercontent.com/translate_f#53)

3.4 Matn sinfining xususiyatlari ............ [.................................. ............................................. 59](https://translate.googleusercontent.com/translate_f#60)

[3.4.1 Matnni namoyish qilish uchun javobgar bo'lgan parametrlar ..................................... 59](https://translate.googleusercontent.com/translate_f#60)

[3.4.2 Yozuvning joylashishiga javobgar parametrlar ................................ 62](https://translate.googleusercontent.com/translate_f#63)

[3.4.3 Yozuvning fonini o'rnatishga mas'ul bo'lgan parametrlar ............... 64](https://translate.googleusercontent.com/translate_f#65)

[3.5 Ranglar paneli - rangli](https://translate.googleusercontent.com/translate_f#67)*panel*[.................................................. ............................. 66](https://translate.googleusercontent.com/translate_f#67)

3.5.1 inset\_locator () yordamida umumiy sozlash .......... [........................ 68](https://translate.googleusercontent.com/translate_f#69)

[3.5.2 O'lchovni o'rnatish va yozuvni o'rnatish .......................................... ............... 70](https://translate.googleusercontent.com/translate_f#71)

3.5.3 Qo'shimcha rang satrini sozlash parametrlari [................. 71](https://translate.googleusercontent.com/translate_f#72)

2018-04-02 121 2

|  |
| --- |
| **4. sahifa** |

[4-bob. Ma'lumotlarni vizuallashtirish ............................................. ........................................ 73](https://translate.googleusercontent.com/translate_f#74)

[4.1 Chiziqli grafik ............................................... .................................................. 73](https://translate.googleusercontent.com/translate_f#74)

[4.1.1 Rejalashtirish ............................................. ....................................... 73](https://translate.googleusercontent.com/translate_f#74)

4.1.1.1 fmt argumenti parametrlari ............. [............................... ........................ 75](https://translate.googleusercontent.com/translate_f#76)

[4.1.2 Grafika va o'qi orasidagi maydonni to'ldirish ......................................... ........ 77](https://translate.googleusercontent.com/translate_f#78)

[4.1.3 Grafik belgilarini belgilash ............................................ .................... 82](https://translate.googleusercontent.com/translate_f#83)

[4.1.4 Diagrammani qisqartirish ............................................. ............................................. 85](https://translate.googleusercontent.com/translate_f#86)

[4.2 Qadam, stek, nuqta va boshqa grafikalar ....................................... 86](https://translate.googleusercontent.com/translate_f#87)

[4.2.1 Bosqichli grafik ............................................. ....................................... 86](https://translate.googleusercontent.com/translate_f#87)

[4.2.2 Stack diagrammasi ............................................. ............................................. 87](https://translate.googleusercontent.com/translate_f#88)

[4.2.3](https://translate.googleusercontent.com/translate_f#89)*Stem-*[diagramma ............................................... .................................................. .88](https://translate.googleusercontent.com/translate_f#89)

4.2.4 nuqta uchastkasi (tarqoq uchastka) ............. [............................ ........ 91](https://translate.googleusercontent.com/translate_f#92)

[4.3 Bar va pirog grafikalar ............................................. ...................... 95](https://translate.googleusercontent.com/translate_f#96)

[4.3.1 Shtrixli jadvallar ............................................. .................................. 95](https://translate.googleusercontent.com/translate_f#96)

[4.3.1.1 Guruhlar jadvallari ............................................ ......... 98](https://translate.googleusercontent.com/translate_f#99)

[4.3.1.2 diagrammasi v](https://translate.googleusercontent.com/translate_f#100)*xatolar paneli*[elementi ............................................... .......... 99](https://translate.googleusercontent.com/translate_f#100)

[4.3.2 Dairesel jadvallar ............................................. .....................................yuz](https://translate.googleusercontent.com/translate_f#101)

[4.3.2.1 Klassik piroglar diagrammasi ............................................ ........yuz](https://translate.googleusercontent.com/translate_f#101)

[4.3.2.2 Ichki doiraviy diagrammalar ............................................ .......... 104](https://translate.googleusercontent.com/translate_f#105)

[4.3.2.3 Teshikli dairesel diagramma ........................................... ......... 105](https://translate.googleusercontent.com/translate_f#106)

[4.4 Rangli panjara ............................................... .................................................. ..106](https://translate.googleusercontent.com/translate_f#107)

[4.4.1](https://translate.googleusercontent.com/translate_f#107)*kolormaplar*[) ................................................. ..................... 106](https://translate.googleusercontent.com/translate_f#107)

[4.4.2 Rangli panjara qurish ............................................ ......................... 107](https://translate.googleusercontent.com/translate_f#108)

[5-bob.](https://translate.googleusercontent.com/translate_f#114)*3D*[qurish-grafikalar.](https://translate.googleusercontent.com/translate_f#114)*Mplot3d Toolkit*[bilan ishlash................................... 113](https://translate.googleusercontent.com/translate_f#114)

[5.1 Chiziqli diagramma ............................................... ................................................. 113](https://translate.googleusercontent.com/translate_f#114)

[5.2 Nuqta chizmasi (tarqalish chizmasi) ........................................... ............ 114](https://translate.googleusercontent.com/translate_f#115)

[5.3 Kadr yuzasi ............................................... ...................................... 116](https://translate.googleusercontent.com/translate_f#117)

[5.4 Yuzaki ................................................ .................................................. ..... 117](https://translate.googleusercontent.com/translate_f#118)

[II qism. Seaborn kutubxonasi ................................................ .......................................... 120](https://translate.googleusercontent.com/translate_f#121)

[Kirish ................................................. .................................................. .................... 120](https://translate.googleusercontent.com/translate_f#121)

[6-bob. Tezkor boshlash ............................................. .................................................. 122](https://translate.googleusercontent.com/translate_f#123)

[6.1 O'rnatish ................................................ .................................................. .......... 122](https://translate.googleusercontent.com/translate_f#123)

6.1.1 O'rnatish imkoniyatlari *dengiz dengizi*[.................................................. ................... 122](https://translate.googleusercontent.com/translate_f#123)

6.1.2 *dengiz qirg'og'ini*[o'rnatish](https://translate.googleusercontent.com/translate_f#123)menejerlar orqali [*pip*](https://translate.googleusercontent.com/translate_f#123) va*konda*[................................... 122](https://translate.googleusercontent.com/translate_f#123)

6.1.3 O'rnatishni tekshirish ............. [............................... ................ 123](https://translate.googleusercontent.com/translate_f#124)

[6.2 Tez boshlash ............................................... .................................................. ... 123](https://translate.googleusercontent.com/translate_f#124)

6.2.1 Nuqta uchastkasini qurish .............. [.............................. ..................... 124](https://translate.googleusercontent.com/translate_f#125)

6.2.2 Chiziqli grafikni qurish .............. [.............................. .................... 125](https://translate.googleusercontent.com/translate_f#126)

6.2.3 Kategorik ma'lumotlar bilan ishlash .............. [............................. ............. 126](https://translate.googleusercontent.com/translate_f#127)

[7-bob. Diagrammalar ko'rinishini moslashtirish ........................................... ................ 129](https://translate.googleusercontent.com/translate_f#130)

[7.1](https://translate.googleusercontent.com/translate_f#130)*dengiz bo'yidagi*[uslublar.................................................. .................................................. 129](https://translate.googleusercontent.com/translate_f#130)

[7.2](https://translate.googleusercontent.com/translate_f#135)*dengiz osti kontekstlari*[.................................................. ......................................... 134](https://translate.googleusercontent.com/translate_f#135)

[7.3 Panjara va o'qlarni o'rnatish ............................................. ........................................ 139](https://translate.googleusercontent.com/translate_f#140)

7.3.1 Grid ...... [........................................ .................................................. ............ 139](https://translate.googleusercontent.com/translate_f#140)

7.3.2 Grafik maydoni va o'qlari ............ [............................... ......................................... 141](https://translate.googleusercontent.com/translate_f#142)

[7.4 Afsona ................................................ .................................................. ............. 145](https://translate.googleusercontent.com/translate_f#146)

[7.5 Shrift ................................................ .................................................. ............... 147](https://translate.googleusercontent.com/translate_f#148)

[7.6 Rang bilan ishlash .............................................. .................................................. .148](https://translate.googleusercontent.com/translate_f#149)

[8-bob. Ma'lumotlardagi munosabatlarni vizualizatsiya qilish ........................................... ................. 153](https://translate.googleusercontent.com/translate_f#154)

[8.1 Funksiyalarning umumiy parametrlari .............................................. ................................ 153](https://translate.googleusercontent.com/translate_f#154)

3

|  |
| --- |
| **Sahifa 5** |

8.1.1 Asosiy argumentlar ............ [................................. ....................................... 154](https://translate.googleusercontent.com/translate_f#155)

8.1.2 Graflarning axborot tarkibini oshirish parametrlari ........ [.............. 154](https://translate.googleusercontent.com/translate_f#155)

8.2 Chiziqli diagramma. Lineplot () funktsiyasi ............ [.................................. ................. 156](https://translate.googleusercontent.com/translate_f#157)

8.2.1 lineplot () funktsiyasiga kirish ............ [............................. ..................... 157](https://translate.googleusercontent.com/translate_f#158)

8 [.2.2 Kutilayotgan qiymat va ishonchni aks ettirish](https://translate.googleusercontent.com/translate_f#161)

intervallar ........ [......................................... .................................................. ........ 160](https://translate.googleusercontent.com/translate_f#161)

8.2.3 Grafikning axborot tarkibini oshirish .............. [.............................. ....... 164](https://translate.googleusercontent.com/translate_f#165)

[8.2.3.1 Rang sxemasini o'rnatish ............................................ .................... 165](https://translate.googleusercontent.com/translate_f#166)

[8.2.3.2 Uslubni o'rnatish ............................................. ..................................... 166](https://translate.googleusercontent.com/translate_f#167)

[8.2.3.3 Chiziq qalinligini sozlash ............................................ ..................... 168](https://translate.googleusercontent.com/translate_f#169)

[8.2.4 Vaqt qatorlarini vizuallashtirish ............................................ ................... 170](https://translate.googleusercontent.com/translate_f#171)

8.3 Tarqoqlik diagrammasi. Scatterplot () funktsiyasi ............ [.................................. ..... 171](https://translate.googleusercontent.com/translate_f#172)

[8.3.1 scatterplot () funktsiyasiga kirish ......................................... ................ 171](https://translate.googleusercontent.com/translate_f#172)

[8.3.2](https://translate.googleusercontent.com/translate_f#173)*Tarqoq uchastkaning*[axborot](https://translate.googleusercontent.com/translate_f#173)*tarkibini*[oshirish................................. 172](https://translate.googleusercontent.com/translate_f#173)

[8.3.2.1 Rang sxemasini o'rnatish ............................................ .................... 173](https://translate.googleusercontent.com/translate_f#174)

[8.3.2.2 Marker uslubini o'rnatish ............................................ .................... 175](https://translate.googleusercontent.com/translate_f#176)

[8.3.2.3 Marker hajmini belgilash ............................................ .................. 177](https://translate.googleusercontent.com/translate_f#178)

[8.4 Diagramma maydonlari elementlarining ko'rinishini o'rnatish ..................................... 179](https://translate.googleusercontent.com/translate_f#180)

8.4.1 Afsona ........ [...................................... .................................................. ........ 179](https://translate.googleusercontent.com/translate_f#180)

[8.4.2 Eksa yorliqlari ............................................. ................................................. 182](https://translate.googleusercontent.com/translate_f#183)

[8.4.3 Ma'lumotlar to'plamini saralash ............................................ ........................... 183](https://translate.googleusercontent.com/translate_f#184)

8.5 Fonni o'rnatish bilan aloqalarni ingl. Relplot () funktsiyasi ... [........ 185](https://translate.googleusercontent.com/translate_f#186)

[9-bob. Kategorik ma'lumotlarning vizualizatsiyasi ............................................ ........... 189](https://translate.googleusercontent.com/translate_f#190)

[9.1 Funksiyalarning umumiy parametrlari .............................................. ................................ 190](https://translate.googleusercontent.com/translate_f#191)

9.1.1 Asosiy parametrlar ............. [................................ ...................................... 190](https://translate.googleusercontent.com/translate_f#191)

9.1.2 Graflarning axborot tarkibini oshirish parametrlari ........ [.............. 191](https://translate.googleusercontent.com/translate_f#192)

[9.2 Kategoriyali ma'lumotlarni tarqalish uchastkalari sifatida tasavvur qilish ... 192](https://translate.googleusercontent.com/translate_f#193)

[9.2.1 Striptiz () funktsiyasi ... ............................................. 192](https://translate.googleusercontent.com/translate_f#193)

[9.2.2 Swarmplot () funktsiyasi ........................................... ........................................ 202](https://translate.googleusercontent.com/translate_f#203)

[9.3 Kategorik ma'lumotlar taqsimotining vizualizatsiyasi ... 205](https://translate.googleusercontent.com/translate_f#206)

[9.3.1 Boxplot () funktsiyasi ........................................... ......................................... 206](https://translate.googleusercontent.com/translate_f#207)

[9.3.2 Skripka () funktsiyasi ........................................... ................................................. 214](https://translate.googleusercontent.com/translate_f#215)

[9.4 Kategorik ma'lumotlarning taxminlarini vizualizatsiya qilish ............................................. ..221](https://translate.googleusercontent.com/translate_f#222)

[9.4.1 Funktsiya nuqtasi () ........................................... ......................................... 222](https://translate.googleusercontent.com/translate_f#223)

9.4.2 Funktsiya barplot () .......... [................................. .............................................. 228](https://translate.googleusercontent.com/translate_f#229)

9.4.3 countplot () funktsiyasi .......... [................................. .......................................... 233](https://translate.googleusercontent.com/translate_f#234)

9.5 Shakl darajasida ishlash. Catplot () funktsiyasi ............ [.................................. ..... 236](https://translate.googleusercontent.com/translate_f#237)

[10-bob. Ma'lumotlardagi taqsimotlarni vizualizatsiya qilish ........................................... ........ 246](https://translate.googleusercontent.com/translate_f#247)

10.1 Distplot () funktsiyasi .......... [................................... ................................................. 246](https://translate.googleusercontent.com/translate_f#247)

10.2 funktsiyasi kdeplot () .......... [................................... ................................................ 253](https://translate.googleusercontent.com/translate_f#254)

10.3 Rugplot funktsiyasi () .......... [................................... ................................................. 262](https://translate.googleusercontent.com/translate_f#263)

[11-bob. Chiziqli regressiya modelining vizualizatsiyasi ........................................... ... 265](https://translate.googleusercontent.com/translate_f#266)

[11.1 Funksiyalarning umumiy parametrlari .............................................. .............................. 265](https://translate.googleusercontent.com/translate_f#266)

11.2 Funktsiyani regplot () .......... [................................... ................................................. 266](https://translate.googleusercontent.com/translate_f#267)

11.3 Funktsiya qoldig'i () .......... [................................... ............................................... 278](https://translate.googleusercontent.com/translate_f#279)

11.4 lmplot funktsiyasi () .......... [................................... .................................................. .281](https://translate.googleusercontent.com/translate_f#282)

[12-bob. Diagrammaning joylashishini boshqarish ............................................ .............. 288](https://translate.googleusercontent.com/translate_f#289)

[12.1](https://translate.googleusercontent.com/translate_f#289)*Facet-* mesh .... [........................................... .................................................. ...... 288](https://translate.googleusercontent.com/translate_f#289)

[12.2](https://translate.googleusercontent.com/translate_f#297)*Pair-* sof ... [............................................ .................................................. ......... 296](https://translate.googleusercontent.com/translate_f#297)

[12.2.1 funktsiya juftligi () ........................................... ......................................... 297](https://translate.googleusercontent.com/translate_f#298)

4

|  |
| --- |
| **6-bet** |

[12.2.2 Class PairPlot ............................................. ............................................... 302](https://translate.googleusercontent.com/translate_f#303)

[12.3](https://translate.googleusercontent.com/translate_f#307)*Birgalikdagi* mash ... [........................................... .................................................. ........ 306](https://translate.googleusercontent.com/translate_f#307)

[12.3.1 qo'shma uchastka () funktsiyasi ... .......................................... 307](https://translate.googleusercontent.com/translate_f#308)

[12.3.2 JointPlot klassi ............................................. .............................................. 313](https://translate.googleusercontent.com/translate_f#314)

[III qism. Kutubxona](https://translate.googleusercontent.com/translate_f#316)*Mayavi*[................................................. .......................................... 315](https://translate.googleusercontent.com/translate_f#316)

[Kirish ................................................. .................................................. .................... 315](https://translate.googleusercontent.com/translate_f#316)

[13-bob. Tezkor boshlash ............................................. ................................................ 317](https://translate.googleusercontent.com/translate_f#318)

[13.1 O'rnatish ................................................ .................................................. ........ 317](https://translate.googleusercontent.com/translate_f#318)

13.2 Tez boshlash ........... [.................................... .................................................. .319](https://translate.googleusercontent.com/translate_f#320)

[13.2.1](https://translate.googleusercontent.com/translate_f#320)*GUI*[bilan ishlash](https://translate.googleusercontent.com/translate_f#320)*Mayavi2*[dasturi.................................................. .... 319](https://translate.googleusercontent.com/translate_f#320)

[13.2.2 rivojlanish](https://translate.googleusercontent.com/translate_f#327)*Mayavi*[yordamida](https://translate.googleusercontent.com/translate_f#327)*Python modullari*[............................. 326](https://translate.googleusercontent.com/translate_f#327)

[13.2.3](https://translate.googleusercontent.com/translate_f#329)*Mayavi*[bilan](https://translate.googleusercontent.com/translate_f#329)*ishlash*[yilda](https://translate.googleusercontent.com/translate_f#329)*Yupyter daftarchasi*[................................................ ........ 328](https://translate.googleusercontent.com/translate_f#329)

[14-bob. Ko'rinishni sozlash ............................................. ............................. 330](https://translate.googleusercontent.com/translate_f#331)

[14.1 Shakl / Sahna nazorati ............................................. ............................. 330](https://translate.googleusercontent.com/translate_f#331)

[14.2 Loyihalash elementlarini sozlash .............................................. ................. 336](https://translate.googleusercontent.com/translate_f#337)

14.2.1 Sahna nomi ........... [.................................. .......................................... 337](https://translate.googleusercontent.com/translate_f#338)

[14.2.2 Modelning tashqi konturi ............................................ .............................. 339](https://translate.googleusercontent.com/translate_f#340)

[14.2.3 Koordinata o'qlarini o'rnatish ............................................ .......................... 340](https://translate.googleusercontent.com/translate_f#341)

[14.2.4 Rang satrini ( *colorbar* ) sozlash ......................................... .......... 344](https://translate.googleusercontent.com/translate_f#345)

[14.3 Kamerani boshqarish ............................................... ........................................ 346](https://translate.googleusercontent.com/translate_f#347)

[15-bob. Ma'lumotlarni vizuallashtirish ............................................. .................................... 349](https://translate.googleusercontent.com/translate_f#350)

[15.1 Bir o'lchovli ma'lumotlar to'plamlari bilan ishlash funktsiyalari ............................. 350](https://translate.googleusercontent.com/translate_f#351)

15.1.13-nuqta () funktsiyasi ........... [................................ ......................................... 351](https://translate.googleusercontent.com/translate_f#352)

[15.1.2 Funksiya fitnasi3d () ........................................... ......................................... 355](https://translate.googleusercontent.com/translate_f#356)

[15.2 Ikki o'lchovli ma'lumotlar to'plamlari bilan ishlash funktsiyalari .................................... 359](https://translate.googleusercontent.com/translate_f#360)

[15.2.1 imshow () funktsiyasi ........................................... .......................................... 360](https://translate.googleusercontent.com/translate_f#361)

15.2.2 surf () funktsiyasi ......... [.................................. ................................................. 362](https://translate.googleusercontent.com/translate_f#363)

15.2.3 funktsiyasi kontur\_surf () ........... [................................ ................................... 366](https://translate.googleusercontent.com/translate_f#367)

15.2.4 Mesh () funktsiyasi ........... [................................ .............................................. 368](https://translate.googleusercontent.com/translate_f#369)

[15.3 3D ma'lumotlar to'plamlari bilan ishlash funktsiyalari .............................. 370](https://translate.googleusercontent.com/translate_f#371)

15.3.1 funktsiya konturi3d () ........... [................................ ....................................... 371](https://translate.googleusercontent.com/translate_f#372)

[15.3.2 funktsiya quiver3d () ........................................... ......................................... 375](https://translate.googleusercontent.com/translate_f#376)

[15.3.3 funktsiyasi hajmi\_slice () ........................................... .................................. 376](https://translate.googleusercontent.com/translate_f#377)

16-bob. Bilan ishlash *quvur liniyasi*[.................................................. ........................................ 379](https://translate.googleusercontent.com/translate_f#380)

16.1 Tuzilishi *quvur liniyasi*[.................................................. ............................................ 382](https://translate.googleusercontent.com/translate_f#383)

16.2 Ma'lumot manbalari bilan ishlash .............. [............................... ........................... 384](https://translate.googleusercontent.com/translate_f#385)

[16.3 Filtrlar bilan ishlash .............................................. .......................................... 389](https://translate.googleusercontent.com/translate_f#390)

[16.4 Modullar bilan ishlash .............................................. ............................................ 402](https://translate.googleusercontent.com/translate_f#403)

[Xulosa ................................................. .................................................. .................... 412](https://translate.googleusercontent.com/translate_f#413)

besh

|  |
| --- |
| **Sahifa 7** |

***Matplotlib* kutubxonasi I. qism**

**Kirish**

*Matplotlib* kutubxonasi eng mashhur vositalardan biridir

*Python-da* ma'lumotlarni vizualizatsiya qilish . U ikkala ijod uchun ham juda yaxshi

statik tasvirlar, ham animatsion, ham interaktiv

echimlar.

*Matplotlib* qismi hisoblanadi *Ilmiy Python* , kutubxonalar bir to'plam uchun

*NumPy-* ni ham o'z ichiga olgan ilmiy hisoblash va ma'lumotlarni vizualizatsiya qilish[1](https://translate.googleusercontent.com/translate_f#7) ,

*SciPy*[2](https://translate.googleusercontent.com/translate_f#7), *Pandalar*[3](https://translate.googleusercontent.com/translate_f#7) ,*SymPy*[4](https://translate.googleusercontent.com/translate_f#7) va boshqa bir qator vositalar.

Ushbu kitob ma'lumotlarni vizualizatsiya qilish masalalarini va

bu chiziqli va qadam grafikalar, diagrammalar qurish

tarqalish, chiziqli va pirogli diagrammalar, chiziqli va *3D* grafikalar.

Grafiklarning ko'rinishini, ularning ko'rinishini moslashtirishga katta e'tibor berildi

elementlar va maket.

Funksiyalar parametrlarini tavsiflashda quyidagilardan foydalaniladi

format:

• **argname: turi (lar)**

◦ **tavsif**

Agar ma'lumotlar turida ixtiyoriy so'z bo'lsa, demak, bu degani

ushbu parametr standart qiymatga ega va ixtiyoriydir

aniq ko'rsatib bering.

1 [https://numpy.org/](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://numpy.org/)

2 [https://scipy.org/](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://scipy.org/)

3 [https://pandas.pydata.org/](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://pandas.pydata.org/)

4 [https://www.sympy.org/en/index.html](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://www.sympy.org/en/index.html)

6

|  |
| --- |
| **8-sahifa** |

**1-bob. Tezkor boshlash**

**1.1 O'rnatish**

**1.1.1 *Matplotlib-*ni o'rnatish parametrlari**

*Matplotlib-* ni o'rnatish uchun ikkita asosiy variant mavjud , birinchisi

*Anaconda* paketini tarqatish *,*

o'z ichiga oladi

bizni qiziqtirgan kutubxona, ikkinchisi - *Matplotlib* installyatsiyasi

dan

paket menejeridan foydalanish. O'rnatish haqida *Anaconda* mumkin

o'qing [*devpractice.ru*](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://devpractice.ru/python-lesson-1-install/)[5](https://translate.googleusercontent.com/translate_f#8)...

**1.1.2 *pip* menejeri yordamida *Matplotlib-ni* o'rnatish**

*Matplotlib-* ni *pip* to'plami menejeri turi yordamida o'rnatish uchun

operatsion tizimingizning buyruq satri quyidagi buyruqlar:

**python -m pip o'rnatish -U pip**

**python -m pip install -U matplotlib**

Birinchisi sizning yangilanadi *PIP* , ikkinchi o'rnatish *Matplotlib* barcha bilan

kerakli bog'liqliklar.

**1.1.3 O'rnatishni tekshirish**

Hammasi to'g'ri o'rnatilganligini tekshirish uchun ishga tushiring

*Python* tarjimoni va unga import buyrug'ini kiriting:

**import**matplotlib

Agar xato xabari bo'lmasa, u holda *Matplotlib kutubxonasi*

o'rnatilgan va ishlatilishi mumkin.

5 [https://devpractice.ru/python-lesson-1-install/](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://devpractice.ru/python-lesson-1-install/)

7

|  |
| --- |
| **9-bet** |

Keling, kutubxona versiyasini tekshirib ko'raylik, ehtimol u farq qiladi

quyida:

>>> matplotlib. nilufar\_\_

**'3.0.3'**

**1.2 Tez boshlash**

*Matplotlib* tafsilotlariga sho'ng'ishdan oldin , bir nechtasini ko'rib chiqing

misollar, qaysi uchun kutubxonadan foydalanishingiz mumkinligini o'rganib chiqdingiz

o'z muammolarini hal qilish va intuitiv tushunchaga erishish

ushbu vosita bilan ishlash tamoyillari.

Agar siz *Jupyter Notebook-* dan foydalanayotgan bo'lsangiz *,* uni olish uchun

kod bilan birga hujayralar yonidagi grafikalar, siz maxsus bajarishingiz kerak

*Matplotlib-ni* import *qilgandan* so'ng *sehrli* buyruq :

plt **sifatida import**matplotlib.pyplot

% matplotlib qatori

Ish natijasi 1.1-rasmda ko'rsatilgandek ko'rinadi.

**1.1-rasm - bilan ishlash misol *Matplotlib* yilda *Jupyter daftar***

8

|  |
| --- |
| **10-bet** |

Agar siz *py* faylida kod yozsangiz va uni qo'ng'iroq orqali *ishlatsangiz*

*Python* tarjimoni , sizga% matplotlib qatori kerak emas,

faqat kutubxona importidan foydalaning. Bunga o'xshash misol

1.1-rasmda ko'rsatilgan, chunki alohida *Python* fayli bo'ladi

shunga o'xshash:

plt **sifatida import**matplotlib.pyplot

plt.plot ([ 1 , 2 , 3 , 4 , 5 ], [ 1 , 2 , 3 , 4 , 5 ])

plt.show ()

Natijada siz izolyatsiya qilingan oynada grafikani olasiz (1.2-rasmga qarang).

**Shakl 1.2 - Izolyatsiya qilingan oynadagi grafik displey**

Bundan tashqari, biz foydalanish xususiyatlari haqida to'xtamaymiz

*sehrli* buyruqlar, agar siz *Jupyter* dan foydalanayotgan bo'lsangiz, shunchaki eslang

*notebook* bilan ishlashda *Matplotlib* , u holda, albatta, albatta, kerak

Inline buyruq% matplotlib.

to'qqiz

|  |
| --- |
| **11-bet** |

Endi to'g'ridan-to'g'ri *Matplotlib-* ga *boramiz* . Darsning maqsadi “ *Tez*

*start* ”- har xil turdagi jadvallarni yaratish, tashqi ko'rinishini sozlash

va ushbu vosita bilan qulay bo'ling.

**1.3 Plotirovka**

Birinchidan, oddiy chiziqli munosabatlarni o'rnatamiz, ism qo'yamiz

grafikalar, o'qlarni belgilang va katakchani ko'rsating:

# Mustaqil (x) va qaram (y) o'zgaruvchilar

x = np.linspace ( 0 , 10 , 50 )

y = x

# Belgilash

plt.title ( **'Lineer bog'liqlik y = x'** ) # sarlavha

plt.xlabel ( **'x'** ) # absissa o'qi

plt.ylabel ( **'y'** ) # ordinatalar o'qi

plt.grid () # panjara ekranini yoqish

plt.plot (x, y) # chizma

Natijada biz 1.3-rasmda ko'rsatilgan grafikani olamiz.

**1.3-rasm - Chiziqli grafik**

o'n

|  |
| --- |
| **Sahifa 12** |

Chiziq turini va uning rangini o'zgartiraylik, buning uchun plot () funktsiyasida, kabi

uchinchi parametr, biz ma'lum bir shakllangan qatorni o'tkazamiz

Shunday qilib, bizning holatimizda bu "r--", bu erda "r" qizil va

'-' - chiziq turi - nuqta chiziq:

# Belgilash

plt.title ( **'Lineer bog'liqlik y = x'** ) # sarlavha

plt.xlabel ( **'x'** ) # absissa o'qi

plt.ylabel ( **'y'** ) # ordinatalar o'qi

plt.grid ()

# panjara ekranini yoqish

plt.plot (x, y, **'r--'** ) # chizma

**1.4-rasm - O'zgartirilgan chiziqli grafik**

Rang va chiziq turini qanday belgilash haqida batafsilroq ma'lumot berilgan

" *2.3 Chiziqli diagramma bilan ishlash* " bo'limiga qarang .

**1.4 Bitta maydonda bir nechta diagrammalar**

Keling, bitta maydonda bir nechta grafikalar tuzamiz, buning uchun biz qo'shamiz

kvadratik qaramlik:

# Lineer bog'liqlik

x = np.linspace ( 0 , 10 , 50 )

y1 = x

o'n bir

|  |
| --- |
| **13-bet** |

# Kvadratik qaramlik

y2 = [i \*\* 2 **uchun** i **yilda** x]

# Belgilash

plt.title ( **'Bog'liqliklar: y1 = x, y2 = x ^ 2'** ) # sarlavha

plt.xlabel ( **'x'** )

# absissa o'qi

plt.ylabel ( **'y1, y2'** ) # ordinatalar o'qi

plt.grid ()

# panjara ekranini yoqish

plt.plot (x, y1, x, y2) # chizma

**1.5-rasm - Bitta maydonda bir nechta grafikalar**

Yuqoridagi misolda plot () ketma-ket ishlaydi

birinchi grafikni chizish uchun ikkita massiv, ikkitasi uchun

ikkinchisini qurish, ikkalasi uchun ham, siz ko'rib turganingizdek

mustaqil o'zgaruvchining *x* qiymatlari massivini *chizmalar* bir xil bo'ladi.

**1.5 Grafalarni turli sohalarda aks ettirish**

Uchinchi, juda keng tarqalgan vazifa - ikkitasini namoyish qilish

yoki aks ettirilishi mumkin bo'lgan turli xil maydonlar

bir nechta grafikalar.

12

|  |
| --- |
| **14-bet** |

Keling, turli sohalarda bizga ma'lum bo'lgan bog'liqliklarni yarataylik:

x = np.linspace ( 0 , 10 , 50 )

y1 = x

# Lineer bog'liqlik

y2 = [i \*\* 2 **uchun** i **yilda** x]

# Kvadratik qaramlik

# Belgilash

plt.figure ( figsize = ( 9 , 9 ))

plt.subplot ( 2 , 1 , 1 )

plt.plot (x, y1)

# fitna

plt.title ( **'Bog'liqliklar: y1 = x, y2 = x ^ 2'** ) # sarlavha

plt.ylabel ( **«y1»** , Shrift = 14 ) # muvofiqlashtirish o'qi

plt.grid ( **To'g'ri** )

# panjara ekranini yoqish

plt.subplot ( 2 , 1 , 2 )

plt.plot (x, y2)

# fitna

plt.xlabel ( **«x»** , Shrift = 14 ) # **x-** o'qi

plt.ylabel ( **«y2»** , Shrift = 14 ) # muvofiqlashtirish o'qi

plt.grid ( **To'g'ri** )

# panjara ekranini yoqish

**Shakl 1.6 - Grafiklar bilan bo'lingan maydonlar**

13

|  |
| --- |
| **Sahifa 15** |

Biz bu erda ikkita yangi xususiyatdan foydalandik:

• figurasi () - global parametrlarni o'rnatish funktsiyasi

grafiklarni namoyish qilish. Anjir parametri unga uzatiladi

shaklning umumiy hajmini belgilaydigan ikki elementli naycha.

• subplot () - maydonning joylashishini belgilaydigan funktsiya

jadval. Uchun maydonlarni aniqlashning bir necha yo'li mavjud

grafikalar chiqishi. Masalan, biz ushbu parametrdan foydalandik,

uchta argumentni o'z ichiga oladi: birinchi dalil

- qatorlar soni, ikkinchisi - hosil qilingan maydonda ustunlar,

uchinchisi - indeks (maydon raqami, yuqoridan pastga, chapga hisoblang

o'ng).

Qolgan funktsiyalar sizga tanish, bundan tashqari biz foydalanganmiz

hajmini belgilash uchun xlabel () va ylabel () funktsiyalarining fontsize parametri

shrift.

**1.6 Kategorik uchun diagramma yaratish**

**ma'lumotlar**

Bungacha biz raqamli ma'lumotlar, ya'ni qaramlik asosida grafikalar tuzdik

va mustaqil o'zgaruvchilar raqamli edi. Amalda, juda

ko'pincha siz raqamli bo'lmagan ma'lumotlar bilan ishlashingiz kerak - ismlar

odamlar, shahar nomlari va boshqalar. Keling, diagramma tuzamiz

do'kondagi mevalar soni ko'rsatiladi:

mevalar = [ **'olma'** , **'shaftoli'** , **'apelsin'** , **'banana'** , **'qovun'** ]

hisoblaydi = [ 34 , 25 , 43 , 31 , 17 ]

plt.bar (mevalar, sonlar)

plt.title ( **'Mevalar!'** )

plt.xlabel ( **"Meva"** )

plt.ylabel ( **'Count'** )

o'n to'rt

|  |
| --- |
| **16-bet** |

**1.7-rasm - Shtrixli jadval**

Diagrammani ko'rsatish uchun bar () funktsiyasidan foydalanganmiz.

Agar siz o'zingiz yugurishga harakat qilgan bo'lsangiz

yuqoridagi misollar, siz allaqachon shakllantirishingiz kerak edi

kutubxona qanday ishlashini bir oz tushunish

*Matplotlib* .

**1.7 Jadvalning asosiy elementlari**

Tasvir bilan bog'liq asosiy atamalar va tushunchalarni ko'rib chiqing

uchun siz tanishishingiz kerak bo'lgan grafikalar

kelajakda ushbu kitobning boblarini o'rganishda hech qanday qiyinchiliklar bo'lmaydi

*Matplotlib kutubxonasi* uchun hujjatlar .

Keyinchalik, biz barchaga murojaat qilish uchun "grafika" atamasidan foydalanamiz

*Matplotlib* ko'rsatadigan rasm (1.8-rasmga qarang) va chiziqlar,

berilgan ma'lumotlar to'plamidan tuzilgan.

15

|  |
| --- |
| **Sahifa 17** |

**1.8-rasm - Jadvalning asosiy elementlari**

*Matplotlib* tasvirni *chizgan* asosiy element

bo'ladi *-rasm* . 1.8-rasmda keltirilgan hamma narsa

rasm elementlari. Keling, uning tarkibiy qismlarini batafsil ko'rib chiqaylik.

**Jadval**

1.8-rasmda ikkita grafik - chiziq va nuqta ko'rsatilgan.

*Matplotlib* juda ko'p turli xil sozlamalarni taqdim etadi,

bu grafikani berish uchun ishlatilishi mumkin

talab qilinadigan ko'rinish: rang, qalinlik, tur, chiziq uslubi va boshqa ko'p narsalar.

**Akslar**

Shaklning ikkinchi muhim elementi - bu o'qlar. Har bir o'q uchun

yorliq (imzo), katta ( *asosiy* ) va qo'shimcha o'rnatishingiz mumkin

( *kichik* ) Shomil, ularning yorliqlari, kattaligi, qalinligi va diapazonlari.

o'n olti

|  |
| --- |
| **18-bet** |

**Panjara va afsona**

Panjara va afsona shakl elementlari bo'lib, ular sezilarli darajada

grafikning ma'lumot tarkibini oshirish. Tarmoq asosiy bo'lishi mumkin

( *katta* ) va qo'shimcha ( *kichik* ). Har bir mash turi o'rnatilishi mumkin

rang, chiziq kengligi va turi. Panjara va afsonani ko'rsatish uchun

tegishli buyruqlardan foydalaniladi.

Quyida grafik tuzilgan kod berilgan,

1.8-rasmda tasvirlangan:

plt **sifatida import**matplotlib.pyplot

**dan**matplotlib.ticker **import**(MultipleLocator, FormatStrFormatter,

AutoMinorLocator)

np **sifatida**numpy- **ni import qilish**

x = np.linspace ( 0 , 10 , 10 )

y1 = 4 \* x

y2 = [i \*\* 2 **uchun** i **yilda** x]

fig, ax = plt.subplots ( figsize = ( 8 , 6 ))

ax.set\_title ( **'Mustaqillik uchastkalari: y1 = 4 \* x, y2 = x ^ 2'** , fontsize = 16 )

ax.set\_xlabel ( **«x»** , Shrift = 14 )

ax.set\_ylabel ( **'y1, y2'** , fontsize = 14 )

ax.grid ( qaysi = **'katta'** , chiziq kengligi = 1,2 )

ax.grid ( qaysi = **'kichik'** , linestyle = **'-'** , color = **'kul'** , chiziq kengligi = 0,5 )

ax.scatter (x, y1, c = **'qizil'** , yorliq = **'y1 = 4 \* x'** )

ax.plot (x, y2, yorliq = **'y2 = x ^ 2'** )

ax.legend ()

ax.xaxis.set\_minor\_locator (AutoMinorLocator ())

ax.yaxis.set\_minor\_locator (AutoMinorLocator ())

ax.tick\_params ( qaysi = **'katta'** , uzunlik = 10 , kenglik = 2 )

ax.tick\_params ( qaysi = **'kichik'** , uzunlik = 5 , kenglik = 1 )

plt.show ()

17

|  |
| --- |
| **19-bet** |

Keyinchalik, biz konfiguratsiya va foydalanish xususiyatlarini batafsil tahlil qilamiz.

1.8-rasmda ko'rsatilgan barcha elementlar.

18

|  |
| --- |
| **20-bet** |

**2-bob. *Piplot* moduli bilan ishlash asoslari**

Diagramma bilan bog'liq deyarli barcha vazifalar bo'lishi mumkin

pyplot moduli tomonidan taqdim etilgan imkoniyatlardan foydalangan holda hal qilish.

Ko'rsatilgan har qanday misollarni ishlatish uchun

birinchi bob, siz ilgari piplotni import qilishingiz kerak edi

*Matplolib* kutubxonalari . Hozirda foydalanuvchilar orasida qabul qilingan

pyplot modulini quyidagi tarzda import qiling:

plt **sifatida import**matplotlib.pyplot

*Matplolib* ijodkorlari uni ishlatishda o'xshash qilishga harakat qilishdi

yilda *MATLAB* Agar ikkinchisi bilan tanish bo'lsangiz, Aql shunday

*Matplolib* osonroq bo'ladi.

**2.1 Rejalashtirish**

Pyplot quradi tasvir asosiy element, *,* deb

ustiga joylashtirilgan bir yoki bir nechta maydon shakl ( *shakl* )

grafikalar, koordinata o'qlari, matn yorliqlari va boshqalar. Qurilish uchun

plot () funktsiyasi ishlatiladi. Eng minimal versiyada

uni parametrlarsiz ishlatish mumkin:

plt **sifatida import**matplotlib.pyplot

% matplotlib qatori

plt.plot ()

Natijada bo'sh maydon ko'rsatiladi (2.1-rasmga qarang).

**2.1-rasm - Bo'sh maydon**

o'n to'qqiz

|  |
| --- |
| **21-bet** |

Keyingi import buyrug'i va *Yupyter* uchun *sehrli* buyruq (birinchi va ikkinchi)

yuqoridagi dastur satrlari) ko'rsatilmaydi.

Agar ro'yxat plot () funktsiyasiga parametr sifatida o'tkazilsa, u holda

ushbu ro'yxatdagi qiymatlar y o'qi (o'qi *y* ) va eksenel bo'yicha joylashtiriladi

abssissa ( *x-* o'qi ) massiv elementlari indekslari bo'ladi:

plt.plot ([ 1 , 7 , 3 , 5 , 11 , 1 ])

**2.2-rasm - uchun qiymatlar bo'yicha chizilgan chiziqli grafik**

***Y* o'qi**

*X* va *Y* o'qlari bo'ylab qiymatlarni o'rnatish uchun siz () chizishingiz kerak

ikkita ro'yxatni o'tkazing:

plt.plot ([ 1 , 5 , 10 , 15 , 20 ], [ 1 , 7 , 3 , 5 , 11 ])

**2.3-rasm - uchun qiymatlarga qarshi chizilgan chiziqli grafik**

***Y* va *X* o'qlari**

20

|  |
| --- |
| **Sahifa 22** |

**2.2 Diagrammadagi matn yorliqlari**

Diagrammadagi eng ko'p ishlatiladigan matn yorliqlari:

• o'qlarning nomlari;

• jadvalning o'zi nomi;

• grafik maydonidagi matnli yozuvlar;

• afsona.

Quyida yuqorida sanab o'tilgan narsalar haqida umumiy ma'lumot mavjud. Ko'proq

ular haqida batafsil hikoya « *3.3 Matn elementlari*

*grafika* ”deb nomlangan.

**2.2.1 O'qlarning nomi**

O'rnatish uchun uchun yorlig'ini *x* , o'qi uchun, XLabel () funktsiyasi foydalanish *y* o'qi -

ylabel () *.*Keling, ushbu funktsiyalarning argumentlari bilan shug'ullanamiz. Asosiy

xlabel () va ylabel () funktsiyalarining parametrlari:

• xlabel (yoki ylabel): str

◦ imzo matni.

• labelpad: raqamli qiymat yoki Yo'q; standart qiymat:

Yo'q

◦ o'qlarni o'z ichiga olgan uchastka maydoni orasidagi masofa

imzo matni.

Xlabel () va ylabel () funktsiyalari qo'shimcha sifatida qabul qiladi

matplotlib.text.Text sinfining konstruktorining parametrlari

(bundan keyin matn), ulardan ba'zilari:

• shrift yoki o'lcham *: ro'yxatdagi* raqam yoki qiymat: {'xx-small',

'x-kichik', 'kichik', 'o'rta', 'katta', 'x-katta', 'xx-

katta '}

◦ Shrift o'lchami.

21

|  |
| --- |
| **23-bet** |

• fontstyle: ro'yxatdagi qiymat: {'normal', 'kursiv',

'qiyshiq'}

◦ Shrift uslubi.

• shrift: 0 dan 1000 gacha bo'lgan raqam yoki

ro'yxat: {'ultralight', 'light', 'normal', 'regular',

"kitob", "o'rta", "roman", "yarim qavatli", "demibold",

'demi', 'qalin', 'og'ir', 'ortiqcha qalin', 'qora'}

◦ Shriftning vazni.

• rang: rangni sozlashning mavjud usullaridan biri (bo'limga qarang

" *2.3.2 Chiziq rangi* ").

◦ imzo matnining rangi.

Foydalanish misoli:

plt.xlabel ( **"Kun"** , Shrift = 15 , rang = **'ko'k'** )

Biz faqat ba'zi funktsiya argumentlarini ko'rib chiqdik

xlabel () va ylabel (), ular haqida batafsil ma'lumot bo'ladi

" *3.1.2 Diagramma o'qlarining yorliqlari* " bo'limida keltirilgan .

**2.2.2 Diagrammaning nomi**

Diagrammaning sarlavhasini o'rnatish uchun title () funktsiyasidan foydalaning:

plt.title ( **'Grafik narxi'** , shrift = 17 )

Uning parametrlari orasida biz quyidagilarni ta'kidlaymiz:

• yorliq: str

◦ sarlavha matni.

• loc: o'rnatilgan qiymat: {'center', 'left', 'right'}

◦ Sarlavhaning hizalanishi.

22

|  |
| --- |
| **24-bet** |

Sinf konstruktorining parametrlari title () funktsiyasi uchun ham mavjud.

Matn, ularning ba'zilari funktsiya argumentlarining tavsifida keltirilgan

xlabel () va ylabel ().

**2.2.3 Matnli eslatma**

Text () funktsiyasi matnni diagramma maydoniga joylashtirish uchun javobgardir.

Uning birinchi va ikkinchi argumentlari pozitsiya koordinatalari, uchinchisi

yorliq matni, foydalanish misoli:

plt.text ( 1 , 1 , **'turi: Chelik'** )

Matnli yozuv ko'rinishini aniq sozlash uchun

Text sinfining konstruktor parametrlaridan foydalaning.

**2.2.4 Afsona**

Afsona chart () funktsiyasini chaqirish orqali diagrammada joylashtiriladi,

ushbu bo'limda biz buning dalillarini ko'rib chiqmaymiz

funktsiyalari. Uni sozlash bo'yicha batafsil ma'lumot keltirilgan

" *3.1 Afsona bilan ishlash* " bo'limiga qarang .

Yuqorida aytib o'tilganlarni allaqachon tanish bo'lgan jadvalga joylashtiramiz

matn elementlari:

x = [ 1 , 5 , 10 , 15 , 20 ]

y = [ 1 , 7 , 3 , 5 , 11 ]

plt.plot (x, y, label = **'temir narxi'** )

plt.title ( **'Grafik narxi'** , shrift = 15 )

plt.xlabel ( **"Kun"** , Shrift = 12 , rang = **'ko'k'** )

plt.ylabel ( **«Narx»** , Shrift = 12 , rang = **'ko'k'** )

plt.legend ()

plt.grid ( **To'g'ri** )

plt.text ( 15 , 4 , **'** ulg'aying **!'** )

23

|  |
| --- |
| **25-bet** |

**2.4-rasm - Diagrammadagi matn yorliqlari**

Ro'yxatdagi variantlarga panjara qo'shdik

grid (True) funktsiyasidan foydalanish.

**2.3 Chiziqli diagramma bilan ishlash**

Ushbu xatboshida asosiy parametrlar muhokama qilinadi

chiziqli jadval ko'rinishini o'zgartirish uchun ishlatilishi mumkin.

Chiziqli diagramma plot () funktsiyasi, uning tashqi ko'rinishi yordamida tuziladi

to'g'ridan-to'g'ri ko'rsatilgan funktsiya argumentlari orqali tuzilishi mumkin,

masalan:

plt.plot (x, y, color = **'red'** )

Shu bilan bir qatorda, setp () funktsiyasidan foydalanishingiz mumkin:

plt.setp ( rang = **"qizil"** , chiziq kengligi = 1 )

**2.3.1 Uchastka chizig'i uslubi**

Grafikning chiziq uslubi linestyle parametri orqali o'rnatiladi, bu

2.1-jadvaldan qiymatlarni qabul qilishi mumkin.

24

|  |
| --- |
| **Sahifa 26** |

**2.1-jadval - chiziqli jadvalning chiziqli uslublari**

**Parametr qiymati**

**Tavsif**

'-' yoki 'qattiq'

Uzluksiz chiziq.

'-' yoki 'chiziqli'

Kesilgan chiziq.

'-.' yoki "chiziqcha"

Chiziq va nuqta chizig'i.

':' yoki 'nuqta'

Nuqta chiziq.

'Yo'q' yoki '' yoki ''

Chiziqni ko'rsatmang.

Chiziq uslubi koordinatalari bo'lmagan ro'yxatlardan so'ng darhol uzatilishi mumkin

bu linestyle parametri ekanligini ko'rsatuvchi:

x = [ 1 , 5 , 10 , 15 , 20 ]

y = [ 1 , 7 , 3 , 5 , 11 ]

plt.plot (x, y, **'-'** )

**Shakl 2.5 - Amaliy uslubga ega chiziq**

Yana bir variant - setp () funktsiyasidan foydalanish:

x = [ 1 , 5 , 10 , 15 , 20 ]

y = [ 1 , 7 , 3 , 5 , 11 ]

chiziq = plt.plot (x, y)

plt.setp (chiziq, linestyle = **'-'** )

25

|  |
| --- |
| **Sahifa 27** |

Natija 2.5-rasmdagi kabi bo'ladi.

Bitta maydonda bir nechta jadvallarni ko'rsatish uchun sizga kerak

tegishli qiymatlar to'plamini plot () funktsiyasiga o'tkazing.

Keling, bir nechta ma'lumotlar to'plamini tuzamiz va boshqasini ko'rsatish orqali ularni namoyish etamiz

chiziq uslublari:

x = [ 1 , 5 , 10 , 15 , 20 ]

y1 = [ 1 , 7 , 3 , 5 , 11 ]

y2 = [i \* 1,2 + 1 **uchun** i **yilda** Y1]

Y3 = [i \* 1,2 + 1 **uchun** i **yilda** Y2]

Y4 = [i \* 1,2 + 1 **uchun** i **yilda** Y3]

plt.plot (x, y1, **'-'** , x, y2, **'-'** , x, y3, **'-.'** , x, y4, **':'** )

**2.6-rasm - bitta funktsiya tomonidan qurilgan bir nechta grafikalar**

**fitna ()**

Xuddi shu natijani plot () ni har biri uchun alohida-alohida chaqirib olish mumkin

ma'lumotlar to'plami. Agar siz grafikalarni ajratib qo'yishni istasangiz do'stingiz

do'stingizdan (har biri o'z sohasida), keyin funktsiyadan foydalaning

subplot () (qarang " *2.4.1* Subplot () *funktsiyasi bilan ishlash* "):

plt.plot (x, y1, **'-'** )

plt.plot (x, y2, **'-'** )

26

|  |
| --- |
| **Sahifa 28** |

plt.plot (x, y3, **'-.'** )

plt.plot (x, y4, **':'** )

**Shakl 2.7 - Turli xil tomonidan qurilgan bir nechta grafikalar**

**funktsiyalar uchastkasi ()**

**2.3.2 Chiziq rangi**

Grafik chizig'ining rangi rang parametri orqali o'rnatiladi (yoki c, agar bo'lsa

qisqartirilgan versiyasidan foydalaning). Qiymat bo'lishi mumkin

quyidagi formatlardan birida taqdim etilgan:

• *RGB* yoki *RGBA:* o'zgaruvchan nuqta qiymatlari oralig'i

[0, 1] (misol: (0,1, 0,2, 0,3);

• *RGB* yoki *RGBA* : *hex* formatidagi qiymat (masalan: '# 0a0a0a');

• diapazonda suzuvchi nuqta raqamini mag'lubiyat bilan ko'rsatish

[0, 1] (rangni kulrang shkalada belgilaydi) (masalan: '0.7');

• to'plamdagi belgi: {'b', 'g', 'r', 'c', 'm', 'y', 'k', 'w'};

• *X11* / *CSS4 palitrasidagi* rangning nomi *;*

• *xkcd* palitrasidagi rang ([https://xkcd.com/color/rgb/](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://xkcd.com/color/rgb/) ) kerak

'xkcd:' prefiksidan boshlang;

• *Tableau Color* to'plamidagi *rang* (palitrasi *T10* ) bilan boshlanishi kerak

prefiks 'tab:'.

27

|  |
| --- |
| **Sahifa 29** |

Agar rang {'b', 'g', 'r' to'plamidagi belgi yordamida ko'rsatilgan bo'lsa,

'c', 'm', 'y', 'k', 'w'}, keyin uni uslub bilan birlashtirish mumkin

plot () funktsiyasining fmt parametri ichidagi chiziqlar. Masalan: kesilgan

qizil chiziq quyidagicha o'rnatiladi: '--r' va chiziqli yashil rang

shuning uchun '-.g':

x = [ 1 , 5 , 10 , 15 , 20 ]

y = [ 1 , 7 , 3 , 5 , 11 ]

plt.plot (x, y, **'-r'** )

**2.8-rasm - kesilgan qizil shaklida berilgan grafik**

**chiziqlar**

**2.3.3 Grafik turi**

Shu paytgacha biz faqat chiziqli jadvallar, funktsiya bilan ishladik

plot () fitna turini belgilashga imkon beradi: chiziqli yoki nuqta. Uchun

nuqta uchastkasi, siz qo'shimcha ravishda markerni belgilashingiz mumkin

ochkolar holati.

28

|  |
| --- |
| **Sahifa 30** |

Mana bir nechta misol:

plt.plot (x, y, **'ro'** )

**2.9-rasm - dumaloq qizil nuqtalardan iborat grafika**

plt.plot (x, y, **'bx'** )

**2.10-rasm - ko'k *x-* belgilaridan iborat grafika**

Nuqta ko'rinishini sozlash bo'yicha qo'shimcha ma'lumot olish uchun

grafiklarni " *4.2.4 nuqta grafigi (diagramma* ) bo'limida topish mumkin

*tarqatish)* ".

29

|  |
| --- |
| **Sahifa 31** |

**2.4 Grafiklarni bir-biridan alohida joylashtirish**

**do'stim**

Diagrammalarni boshqasiga joylashtirishning uchta asosiy yondashuvi mavjud

maydonlar:

• joylashishni aniqlash uchun subplot () funktsiyasidan foydalanish

maydonni grafik bilan joylashtirish;

• oldindan o'rnatish uchun pastki chiziqlar () funktsiyasidan foydalanish

maydonlar sig'adigan panjara;

• geometriyani yanada aniqroq aniqlash uchun GridSpec-dan foydalanish

katakchada grafikalar bilan maydonlarni joylashtirish.

Ushbu bobda dastlabki ikkita yondashuv ko'rib chiqiladi.

**2.4.1 Subplot () funktsiyasi bilan ishlash**

Grafiklarni alohida maydonlarda aks ettirishning eng oson usuli

ularning joylashishini aniqlash uchun subplot () funktsiyasidan foydalaning. Oldin

shu nuqtada biz to'g'ridan-to'g'ri *shakl* , uning qiymatlari bilan ishlamadik

sukut bo'yicha o'rnatilgan parametrlar, biz qoniqdik. Yechimlar uchun

joriy vazifa parametrlardan birini talab qiladi - fon hajmi,

qo'lda o'rnatish. Buning uchun fig () funktsiyasining figsize argumenti javob beradi,

unga ikkita suzuvchi *-* elementlardan tashkil topgan katak beriladi

substratning balandligi va kengligini aniqlash.

Hajmi o'rnatilgandan so'ng, joy ko'rsatiladi: qaerda bo'ladi

subplot () funktsiyasidan foydalangan holda uchastkali maydonni o'rnating.

o'ttiz

|  |
| --- |
| **Sahifa 32** |

Subplot () ga qo'ng'iroq qilishning quyidagi variantlari mavjud:

subplot (nrows, ncols, index)

• nrows: int

Lines qatorlar soni.

• ncols: int

◦ ustunlar soni.

• indeks: int

The buyumning joylashgan joyi.

subplot (pos)

• pos: int

Osition Lavozim. O'z ichiga olgan uch xonali raqam sifatida ko'rsatilgan

qatorlar, ustunlar va indekslar soni haqida ma'lumot, masalan

212 raqami quyidagilarni anglatadi: ikkita chiziq bilan belgi tayyorlang va

bitta ustunda, elementni ikkinchisidagi birinchi holatiga o'tkazing

torlar.

Agar qatorlar va ustunlar soni bo'lsa, ikkinchi variantdan foydalanish mumkin

panjaralar 10 dan oshmasligi kerak, aks holda birinchi variantga murojaat qilish yaxshiroqdir.

Keling, ushbu funktsiyalar bilan ishlashning bir misolini ko'rib chiqamiz:

# Asl ma'lumotlar to'plami

x = [ 1 , 5 , 10 , 15 , 20 ]

y1 = [ 1 , 7 , 3 , 5 , 11 ]

y2 = [i \* 1,2 + 1 **uchun** i **yilda** Y1]

Y3 = [i \* 1,2 + 1 **uchun** i **yilda** Y2]

Y4 = [i \* 1,2 + 1 **uchun** i **yilda** Y3]

# Astar o'lchamini belgilash

plt.figure ( figsize = ( 12 , 7 ))

# Grafik chiqishi

plt.subplot ( 2 , 2 , 1 )

plt.plot (x, y1, **'-'** )

31

|  |
| --- |
| **Sahifa 33** |

plt.subplot ( 2 , 2 , 2 )

plt.plot (x, y2, **'-'** )

plt.subplot ( 2 , 2 , 3 )

plt.plot (x, y3, **'-.'** )

plt.subplot ( 2 , 2 , 4 )

plt.plot (x, y4, **':'** )

**2.11-rasm - Grafiklarni alohida maydonlarga joylashtirish**

**(1-misol)**

Subplot () chaqiruvining ikkinchi variantidan foydalanib, xuddi shu muammoni hal qilaylik:

plt.subplot ( 221 )

plt.plot (x, y1, **'-'** )

plt.subplot ( 222 )

plt.plot (x, y2, **'-'** )

plt.subplot ( 223 )

plt.plot (x, y3, **'-.'** )

plt.subplot ( 224 )

plt.plot (x, y4, **':'** )

Natijada ko'rsatilganiga o'xshash grafik bo'ladi

2.11-rasm.

32

|  |
| --- |
| **Sahifa 34** |

**2.4.2 Subplots () funktsiyasi bilan ishlash**

Funktsional qo'ng'iroqlarning ketma-ketligini ishlatishda noqulaylik

subplot () - bu har safar belgilashingiz kerak

katakchadagi qatorlar va ustunlar soni. Bunga yo'l qo'ymaslik uchun

Agar subplots () funksiyasidan foydalanish mumkin *,* uning barcha parametrlari bo'yicha

bizni faqat dastlabki ikkitasi qiziqtiradi, ular orqali ularning soni

qatorning satrlari va ustunlari. Subplots () funktsiyasi ikkita ob'ektni qaytaradi,

birinchisi - bu *shakllar* , maydonlar joylashgan fon

grafikalar, ikkinchisi Axes ob'ekti (yoki ob'ektlar massivi), bu orqali

tashqi ko'rinishini sozlash uchun to'liq kirish huquqini olishingiz mumkin

ko'rsatilgan narsalar.

() Pastki chizmalaridan foydalanib to'rtta grafikni ko'rsatish masalasini hal qilaylik:

fig, axs = plt.subplots ( 2 , 2 , figsize = ( 12 , 7 ))

o'qlar [ 0 , 0 ] .plot (x, y1, **'-'** )

o'qlar [ 0 , 1 ] .plot (x, y2, **'-'** )

o'qlar [ 1 , 0 ] .plot (x, y3, **'-.'** )

boltlar [ 1 , 1 ] .plot (x, y4, **':'** )

**2.12-rasm - Grafiklarni alohida maydonlarga joylashtirish**

**(misol 2)**

33

|  |
| --- |
| **Sahifa 35** |

**3-bob. Grafik elementlarini sozlash**

**3.1 Afsona bilan ishlash**

Ushbu bo'lim quyidagi mavzularni qamrab oladi: displey

afsona, uning joylashuvini jadvalda belgilash, tashqi o'rnatish

afsonaning bir turi.

**3.1.1 Afsonani namoyish qilish**

Legend () funktsiyasi afsonani diagrammada aks ettirish uchun ishlatiladi *.*

Qo'ng'iroq qilishning quyidagi variantlari mavjud:

afsona ()

afsona (yorliqlar)

afsona (tutqichlar, yorliqlar)

Birinchi variantda afsonaning yorliqlaridan foydalaniladi

chizma funktsiyalarida ko'rsatilgan yorliqlar (yorliq parametri):

x = [ 1 , 5 , 10 , 15 , 20 ]

y1 = [ 1 , 7 , 3 , 5 , 11 ]

y2 = [ 4 , 3 , 1 , 8 , 12 ]

plt.plot (x, y1, **'o-r'** , label = **'qator 1'** )

plt.plot (x, y2, **'o-.g'** , label = **'1-satr'** )

plt.legend ()

**3.1-rasm - Diagrammadagi afsona (1-misol)**

34

|  |
| --- |
| **Sahifa 36** |

Ikkinchi variant sizga mustaqil ravishda matn yorlig'ini belgilashga imkon beradi

ko'rsatilgan ma'lumotlar:

plt.plot (x, y1, **'o-r'** )

plt.plot (x, y2, **'o-.g'** )

plt.legend ([ **'L1'** , **'L2'** ])

**3.2-rasm - Diagrammadagi afsona (2-misol)**

Uchinchi variantda siz satrlarning yozishmalarini qo'lda belgilashingiz mumkin

matn yorliqlari:

line1, = plt.plot (x, y1, **'o-b'** )

line2, = plt.plot (x, y2, **'o-.m'** )

plt.legend ((chiziq2, satr1), [ **'L2'** , **'L1'** ])

**3.3-rasm - Diagrammadagi afsona (3-misol)**

35

|  |
| --- |
| **37-bet** |

**3.1.2 Afsonaning jadvaldagi joylashuvi**

Afsonaning joylashuvi loc parametri bilan belgilanadi, bu

3.1-jadvalda ko'rsatilgan qiymatlardan birini olishi mumkin.

**3.1-jadval - Grafikdagi afsonaviy joylashuv parametrlari**

**String tavsifi**

**Kod**

"eng yaxshi"

0

"yuqori o'ng"

1

"yuqori chap"

2018-04-02 121 2

"pastki chap"

3

"pastki o'ng"

4

"to'g'ri"

besh

"markaz chap"

6

"markaz o'ng"

7

"pastki markaz"

8

"yuqori markaz"

to'qqiz

"markaz"

o'n

Quyida turli xil variantlarni ko'rsatadigan misol keltirilgan

lok parametri orqali afsonaning joylashuvi:

locs = [ **'best'** , **'yuqori o'ng'** , **'yuqori chap'** , **'pastki chap'** ,

**"pastki o'ng"**, **"o'ng"**, **"markaz chap"**, **"o'ng markaz"**,

**'pastki markaz'**, **'yuqori markaz'**, **'markaz'**]

plt.figure ( figsize = ( 12 , 12 ))

**uchun**I **-yilda** qator ( 3 ):

**uchun**j **yilda** qator ( 4 ):

**agar**i \* 4 + j < 11 bo'lsa :

plt.subplot ( 3 , 4 , i \* 4 + j + 1 )

plt.title (locs [i \* 4 + j])

plt.plot (x, y1, **'o-r'** , label = **'qator 1'** )

36

|  |
| --- |
| **38-bet** |

plt.plot (x, y2, **'o-.g'** , label = **'chiziq 2'** )

plt.legend ( loc = locs [i \* 4 + j])

**boshqa**:

**tanaffus**

**3.4-rasm - afsonaning joylashuvi uchun turli xil variantlar**

**grafikalar**

Afsonaning joylashgan joyini yanada moslashuvchan boshqarish uchun siz qilishingiz mumkin

legend () funktsiyasining bbox\_to\_anchor parametridan foydalaning .

37

|  |
| --- |
| **39-bet** |

Ushbu parametrga to'rt yoki ikkitadan korniş beriladi

elementlar:

bbox\_to\_anchor = (x, y, kenglik, balandlik)

bbox\_to\_anchor = (x, y)

bu erda x, y - afsonaviy joylashuv koordinatalari;

kenglik - kenglik;

balandlik - balandlik.

Bbox\_to\_anchor parametridan foydalanish misoli:

plt.plot (x, y1, **'o-r'** , label = **'qator 1'** )

plt.plot (x, y2, **'o-.g'** , label = **'1-satr'** )

plt.legend ( bbox\_to\_anchor = ( 1 , 0.6 ))

**3.5-rasm - afsonaning diagramma maydonidan tashqarida joylashishi**

**3.1.3 Qo'shimcha afsonaviy sozlamalar**

3.2-jadvalda qo'shimcha parametrlar keltirilgan

afsonani aniq sozlash uchun ishlatilishi mumkin.

38

|  |
| --- |
| **40-bet** |

**3.2-jadval - Afsonaviy displeyni sozlash variantlari**

**Parametr**

**Turi**

**Tavsif**

shrift

int *,* float yoki

{'xx-small', 'x-small',

"kichik", "o'rta",

'katta', 'x-katta', 'xx-

katta '}

Shrift kattaligi

afsonalar.

ramka

bool

Displey ramkasi.

ramkalfa

Hech kim yoki suzmaydi

Afsonaning shaffofligi.

yuz rangi

Yo'q yoki str

Rangni to'ldiring.

edgecolor

Yo'q yoki str

Asosiy rang.

sarlavha

Yo'q yoki str

Sarlavha matni.

title\_fontsize Yo'q yoki str

Shrift hajmi.

Afsonaviy parametrlar bilan ishlashga misol:

plt.plot (x, y1, **'o-r'** , label = **'qator 1'** )

plt.plot (x, y2, **'o-.g'** , label = **'1-satr'** )

plt.legend ( Shrift = 14 , soya = **True** , framealpha = 1 , facecolor = **'Y'** ,

edgecolor = **'r'** , title = **'Afsona'** )

**Shakl 3.6 - Afsonaning ko'rinishini o'rnatish misoli**

39

|  |
| --- |
| **41-bet** |

**3.2 Grafiklarning joylashuvi**

Eng oddiy va eng ko'p ishlatiladigan tartib variantlari

grafikalar « *2-bobda* muhokama qilingan. *Modul bilan ishlash asoslari*

piplot "deb nomlangan. Ushbu bo'limda biz GridSpec vositasini ko'rib chiqamiz

tartibni aniq sozlash.

***3.2.1* GridSpec *vositasi***

GridSpec klassi panjara geometriyasini va joylashishni belgilashga imkon beradi

uning dalalari grafikalar bilan. GridSpec bilan ishlash kabi ko'rinishi mumkin

juda noqulay va qo'shimcha kod yozishni talab qiladi, ammo agar shunday bo'lsa

maydonlarni grafika bilan ahamiyatsiz tarzda tartibga solish kerak, keyin

ushbu vosita ajralmas bo'lib qoladi. Bilan ishlashdan oldin

GridSpec *,* uni import qiling:

matplotlib.gridspec-ni gridspec **sifatida import**qiling

Birinchidan, ikkita maydonni grafikalar bilan namoyish etishning oddiy masalasini echaylik

GridSpec-dan foydalanish:

x = [ 1 , 2 , 3 , 4 , 5 ]

y1 = [ 9 , 4 , 2 , 4 , 9 ]

y2 = [ 1 , 7 , 6 , 3 , 5 ]

FG = plt.figure ( figsize = ( 7 , 3 ), constrained\_layout = **True** )

gs = gridspec.GridSpec ( ncols = 2 , nrows = 1 , figure = fg)

fig\_ax\_1 = fg.add\_subplot (gs [ 0 , 0 ])

plt.plot (x, y1)

fig\_ax\_2 = fg.add\_subplot (gs [ 0 , 1 ])

plt.plot (x, y2)

40

|  |
| --- |
| **42-bet** |

**3.7-rasm - Grafikli ikkita maydon**

Quyidagi qatorda GridSpec sinfining ob'ekti yaratilgan:

gridspec.GridSpec ( ncols = 2 , nrows = 1 , figure = fg)

Ustunlar, qatorlar va raqamlar soni sinf konstruktoriga beriladi,

hamma narsa ko'rsatiladigan joyda.

GridSpec ob'ektini yaratishning muqobil usuli quyidagicha:

gs = fg.add\_gridspec ( 1 , 2 )

Bu erda fg add\_gridspec () usuliga ega bo'lgan Shakl ob'ekti,

belgilangan parametrlarga ega bo'lgan mash qo'shishga imkon beradi (

bizning holatimizda bitta qator va ikkita ustun).

Maydon joylashgan panjara elementlarini o'rnatish uchun

grafik, GridSpec sizga o'xshash sintaksisdan foydalanishga imkon beradi

bu *Numpy-* da *tilimlarni* qurish uchun ishlatiladi .

Mavjud ma'lumotlarga yana bitta ma'lumotlar to'plamini qo'shaylik:

x = [ 1 , 2 , 3 , 4 , 5 ]

y1 = [ 9 , 4 , 2 , 4 , 9 ]

y2 = [ 1 , 7 , 6 , 3 , 5 ]

y3 = [- 7 , - 4 , 2 , - 4 , - 7 ]

41

|  |
| --- |
| **43-bet** |

Grafiklarni yangi tartibda tuzamiz:

FG = plt.figure ( figsize = ( 9 , 4 ), constrained\_layout = **True** )

gs = fg.add\_gridspec ( 2 , 2 )

fig\_ax\_1 = fg.add\_subplot (gs [ 0 ,: ])

plt.plot (x, y2)

fig\_ax\_2 = fg.add\_subplot (gs [ 1 , 0 ])

plt.plot (x, y1)

fig\_ax\_3 = fg.add\_subplot (gs [ 1 , 1 ])

plt.plot (x, y3)

**3.8-rasm - Bepul joylashish (1-misol)**

Quyida ma'lumotlarsiz yana bir misol keltirilgan (bo'sh joylar bilan),

bu GridSpec imkoniyatlarini aks ettiradi:

FG = plt.figure ( figsize = ( 9 , 9 ), constrained\_layout = **True** )

gs = fg.add\_gridspec ( 5 , 5 )

fig\_ax\_1 = fg.add\_subplot (GS [ 0 : 3 ])

fig\_ax\_1.set\_title ( **'gs [0** ,: **3]'** )

fig\_ax\_2 = fg.add\_subplot (gs [ 0 , 3 :])

fig\_ax\_2.set\_title ( **'gs [0, 3:]'** )

fig\_ax\_3 = fg.add\_subplot (gs [ 1 :, 0 ])

fig\_ax\_3.set\_title ( **'gs [1:, 0]'** )

42

|  |
| --- |
| **Sahifa 44** |

fig\_ax\_4 = fg.add\_subplot (gs [ 1 :, 1 ])

fig\_ax\_4.set\_title ( **'gs [1:, 1]'** )

fig\_ax\_5 = fg.add\_subplot (gs [ 1 , 2 :])

fig\_ax\_5.set\_title ( **'gs [1, 2:]'** )

fig\_ax\_6 = fg.add\_subplot (gs [ 2 : 4 , 2 ])

fig\_ax\_6.set\_title ( **'gs [2: 4, 2]'** )

fig\_ax\_7 = fg.add\_subplot (gs [ 2 : 4 , 3 :])

fig\_ax\_7.set\_title ( **'gs [2: 4, 3:]'** )

fig\_ax\_8 = fg.add\_subplot (gs [ 4 , 3 :])

fig\_ax\_8.set\_title ( **'gs [4, 3:]'** )

**3.9-rasm - Bepul joylashish (2-misol)**

43

|  |
| --- |
| **45-bet** |

Siz maydonlarning o'lchamlarini oldindan belgilashingiz va ularni quyidagicha o'tkazishingiz mumkin

parametrlar qatori sifatida:

FG = plt.figure ( figsize = ( 5 , 5 ), constrained\_layout = **True** )

kengliklari = [ 1 , 3 ]

balandliklar = [ 2 , 0,7 ]

gs = fg.add\_gridspec ( ncols = 2 , nrows = 2 , width\_ratios = widths ,

height\_ratios = balandliklar)

fig\_ax\_1 = fg.add\_subplot (gs [ 0 , 0 ])

fig\_ax\_1.set\_title ( **'w: 1, h: 2'** )

fig\_ax\_2 = fg.add\_subplot (gs [ 0 , 1 ])

fig\_ax\_2.set\_title ( **'w: 3, h: 2'** )

fig\_ax\_3 = fg.add\_subplot (gs [ 1 , 0 ])

fig\_ax\_3.set\_title ( **'w: 1, h: 0.7'** )

fig\_ax\_4 = fg.add\_subplot (gs [ 1 , 1 ])

fig\_ax\_4.set\_title ( **'w: 3, h: 0.7'** )

**3.10-rasm - Bepul joylashish (3-misol)**

44

|  |
| --- |
| **46-bet** |

Ushbu bo'lim va " *2-bob.* Ma'lumotlar *bilan ishlash asoslari*

pyplot *moduli* ”yaratish uchun etarli bo'lishi kerak

deyarli har qanday murakkablikning tartibi.

**3.3 Grafikning matn elementlari**

Grafik tuzishda matn tarkibidagi qismni ajratib ko'rsatish kerak

quyidagi komponentlar:

• maydon nomi (sarlavha);

• rasmning sarlavhasi (taglavha);

• eksa yorliqlari (xlabel, ylabel);

• grafik maydonidagi matn bloki (matn) yoki rasm (figurali matn);

• izoh - indeksli matn.

Muayyan narsalarga qo'shimcha ravishda matnni o'z ichiga olgan har bir element

uni sozlash uchun javobgar parametrlar - bu Text sinfining parametrlari,

juda ko'p sonli tashqi ko'rinish sozlamalariga kirishni ochadigan va

matn elementining joylashuvi. Batafsil tavsif

Matn sinfidan mavjud bo'lgan parametrlar “ *3.4*

Matn *sinfining xususiyatlari* ". Quyida barchasini ko'rsatadigan kod mavjud

yuqoridagi matn elementlari:

plt.figure ( figsize = ( 10 , 4 ))

plt.figtext ( 0,5 , - 0,1 , **'figtext'** )

plt.suptitle ( **'suptitle'** )

plt.subplot ( 121 )

plt.title ( **"sarlavha"** )

plt.xlabel ( **'** xlabel **'** )

plt.ylabel ( **'** ylabel **'** )

plt.text ( 0,2 , 0,2 , **'matn'** )

45

|  |
| --- |
| **Sahifa 47** |

plt.annotate ( **'izoh'** , xy = ( 0,2 , 0,4 ), xytext = ( 0,6 , 0,7 ),

arrowprops = dict ( facecolor = **'qora'** , shrink = 0,05 ))

plt.subplot ( 122 )

plt.title ( **"sarlavha"** )

plt.xlabel ( **'** xlabel **'** )

plt.ylabel ( **'** ylabel **'** )

plt.text ( 0,5 , 0,5 , **'matn'** )

**3.11-rasm - Diagrammaning matn elementlari**

Taqdim etilgan ba'zi matn elementlari bizda allaqachon mavjud

" *2-bob.* Piplot *moduli bilan ishlash asoslari* " da muhokama qilingan

dars ularni batafsil o'rganib chiqadi. O'z ichiga olgan grafik elementlari

matnda bir qator sozlash parametrlari mavjud, ular rasmiyda

hujjatlar \*\* kvarg sifatida belgilanadi. Bu sinfning xususiyatlari

matplotlib.text.Tashqi ko'rinishni boshqarish uchun ishlatiladi

matn.

46

|  |
| --- |
| **Sahifa 48** |

**3.3.1 Rasm sarlavhasi va diagramma maydonlari**

Diagramma maydonining sarlavhasi bilan boshlaymiz. Sarlavha matni bilan o'rnatiladi

quyidagi asosiy xususiyatga ega title () funktsiyasidan foydalangan holda

dalillar:

• yorliq: str

◦ sarlavha matni.

• shrift: dikt

The Yorliq ko'rsatilishini boshqarish uchun lug'at mavjud

quyidagi tugmalar:

▪ 'shrift': shrift hajmi;

▪ 'shrift': shrift;

▪ 'vertikal tekislash': vertikal tekislash;

▪ "gorizontal yo'nalish": gorizontal tekislash.

• loc: {'center', 'left', 'right'}, str, ixtiyoriy

◦ tekislash.

• pad: float

◦ Sarlavha sarlavhasi va uchastkasining yuqori qismi orasidagi bo'shliq.

Title () funktsiyasi shuningdek xususiyatlarni argument sifatida qo'llab-quvvatlaydi

sinf matni:

vazn = [ **'engil'** , **'odatiy'** , **'qalin'** ]

plt.figure ( figsize = ( 12 , 4 ))

**uchun**I, LC **yilda** enumerate ([ **"left"** , **"markazi"** , **"o'ng"** ]):

plt.subplot ( 1 , 3 , i + 1 )

plt.title ( label = LC, loc = LC, Shrift = 12 + i \* 5 , FontWeight [i] og'irlik =,

pad = 10 + i \* 15 )

47

|  |
| --- |
| **49-bet** |

**3.12-rasm - Diagrammaning sarlavhasi**

Shakl sarlavhasi suptitle () funktsiyasi, argumentlar yordamida o'rnatiladi

bu funktsiya sarlavha () uchun muhokama qilingan narsaga o'xshaydi.

Matn sinfining xususiyatlari orqali aniqroq sozlash mumkin.

**3.3.2 Eksa yorliqlari**

Piplot bilan ishlashda uchastka o'qi yorliqlarini o'rnatish,

ob'ekt bilan ishlashda labelx () va labely () *funktsiyalaridan foydalaniladi*

O'qlar - set\_xlabel () va set\_ylabel () funktsiyalari .

Funktsiyalarning asosiy dalillari deyarli to'liq mos keladi

title () funktsiyasi uchun berilgan:

• yorliq: str

◦ imzo matni.

• shrift: dikt

The Sarlavha ko'rsatilishini boshqarish uchun lug'at, o'z ichiga oladi

quyidagi tugmalar:

▪ 'shrift': shrift hajmi;

▪ 'shrift': shrift;

▪ 'vertikal tekislash': vertikal tekislash;

▪ "gorizontal yo'nalish": gorizontal tekislash.

48

|  |
| --- |
| **Sahifa 50** |

• labelpad: suzuvchi

Label Yorliq va eksa orasidagi bo'shliq.

Oddiy holatda, faqat imzo matnini o'tkazish kifoya

mag'lubiyat sifatida:

x = [i **uchun** I **-yilda** qator ( 10 )]

y = [i \* 2 **uchun** I **-yilda** qator ( 10 )]

plt.plot (x, y)

plt.xlabel ( **'X o'qi'** )

plt.ylabel ( **'Y o'qi'** )

**3.13-rasm - Diagramma o'qlari yorliqlari (1-misol)**

49

|  |
| --- |
| **Sahifa 51** |

Moslashtirish uchun ba'zi bir rivojlangan xususiyatlardan foydalanish

eksa belgilarining ko'rinishi:

plt.plot (x, y)

plt.xlabel ( **"X-o'qi \ n Mustaqil qiymati"** , Shrift = 14 , FontWeight = **'qalin'** )

plt.ylabel ( **"Y-o'qi \ n qaram qiymati"** , Shrift = 14 , FontWeight = **'qalin'** )

**3.14-rasm - Diagramma o'qlari yorliqlari (2-misol)**

**3.3.3 Matn bloki**

Funksiya diagramma maydoniga matn bloklarini o'rnatish uchun javobgardir

matn (). Ushbu funktsiyaning asosiy parametrlaridan foydalanish mumkin

joylashuvi, tarkibi va shrift sozlamalari:

• x: float

◦ yorliqning *x-* koordinatali qiymati .

• y: suzmoq

◦ *y-* qiymatini muvofiqlashtirish yorlig'i.

• s: str

Text Matn yorlig'i.

50

|  |
| --- |
| **Sahifa 52** |

Eng sodda shaklda () dan foydalanish quyidagi ko'rinishga ega bo'ladi:

plt.text ( 0 , 7 , **'SALOM!'** , fontsize = 15 )

plt.plot ( oraliq ( 0 , 10 ), oraliq ( 0 , 10 ))

**3.15-rasm - Matn bloki (1-misol)**

Text sinfining xususiyatlaridan foydalanish

buni o'zgartirish uchun

vakillik:

bbox\_properties = dict ( boxstyle = **'darrow, pad = 0.3'** , ec = **'k'** , fc = **'y'** , ls = **'-'** ,

lw = 3 )

plt.text ( 2 , 7 , **«HELLO!»** , Shrift = 15 , bbox = bbox\_properties)

plt.plot ( oraliq ( 0 , 10 ), oraliq ( 0 , 10 ))

**3.16-rasm - Matn bloki (2-misol)**

51

|  |
| --- |
| **Sahifa 53** |

**3.3.4 Izohlar**

Izohlash vositasi ko'rsatilgan blokirovkalarni o'rnatishga imkon beradi

jadvaldagi ma'lum bir joyga ishora qilish uchun tarkib va ​​o'q.

Izoh yaratish uchun annotate () funktsiyasidan foydalaning

uning dalillari:

• matn: str

Not Izoh matni.

• xy: (float, float)

The o'q ko'rsatadigan joyning koordinatalari.

• xytext: (float, float), ixtiyoriy

◦ Matn yorlig'i joylashgan joyning koordinatalari.

• xycoords: str

The Joy belgilanadigan koordinatalar tizimi

ko'rsatgich.

• textcoords: str

The Joy belgilanadigan koordinatalar tizimi

matn bloki.

• o'q o'qlari: diktator, ixtiyoriy

◦ Oklarni ko'rsatish parametrlari. Ushbu parametrlarning nomlari

(lug'at tugmachalari) - bu ob'ekt konstruktorining parametrlari

FancyArrowPatch klassi.

Quyida ishlashni ko'rsatadigan namunaviy kod mavjud

annotation () funktsiyasi bilan:

matematikani **import qilish**

x = ro'yxat ( oraliq (- 5 , 6 ))

y = [i \*\* 2 **uchun** i **yilda** x]

52

|  |
| --- |
| **Sahifa 54** |

plt.annotate ( **'min'** , xy = ( 0 , 0 ), xycoords = **'ma'lumotlar' '** , xytext = ( 0 , 10 ),

textcoords = **'ma'lumotlar'** , arrowprops = dict ( facecolor = **'g'** ))

plt.plot (x, y)

**3.17-rasm - izohlash () funktsiyasi bilan ishlashni namoyish etish**

Xycoords va textcoords parametrlariga qiymat berilishi mumkin

3.3-jadval.

**3.3-jadval - xycoords va textcoords parametrlarining qiymatlari**

**Qiymat**

**Tavsif**

"ko'rsatkich nuqtalari"

Kelib chiqishi shaklning pastki chap burchagi

(0, 0). Koordinatalar punktlarda ko'rsatilgan.

"rasm piksellari"

Kelib chiqishi shaklning pastki chap burchagi

(0, 0). Koordinatalar piksel bilan berilgan.

"figurali fraktsiya" kelib chiqishi rasmning pastki chap burchagidir

(0, 0) yuqori o'ng burchak nuqta bo'lsa

(o'n bir). Koordinatalar bitta qismda ko'rsatilgan.

"o'qlar nuqtalari"

Kelib chiqishi - maydonning pastki chap burchagi

grafikalar (0, 0). Koordinatalar punktlarda ko'rsatilgan.

53

|  |
| --- |
| **Sahifa 55** |

"o'qlar piksellari"

Kelib chiqishi - maydonning pastki chap burchagi

grafikalar (0, 0). Koordinatalar piksel bilan berilgan.

"o'qlar kasrlari"

Kelib chiqishi - maydonning pastki chap burchagi

maydonning yuqori o'ng burchagi bilan grafikalar (0, 0)

Bu nuqta (1, 1). Koordinatalar ulushlarda o'rnatiladi

bittadan.

"ma'lumotlar"

Koordinata tizimi turi: dekart. Ish

grafika maydonida amalga oshiriladi.

"qutbli"

Koordinatali tizim turi: qutbli. Ish

grafika maydonida amalga oshiriladi.

Yozuv ko'rinishini o'zgartirish uchun xususiyatlardan foydalaning

sinf Matn.

Izoh o'qining ko'rinishini moslashtirishni ko'rib chiqamiz. Orqasida

o'qni ko'rsatish konfiguratsiyasi arrowprops parametriga mos keladi,

bu kimning kalitlari bo'lgan lug'atni qiymat sifatida qabul qiladi

FancyArrowPatch klassi konstruktorining parametrlari bo'lib, ulardan

ikkitasini tanlang: o'q uslubi (o'q uslubi uchun javobgar) va

connectionstyle (ulagichning uslubi uchun javobgardir).

54

|  |
| --- |
| **Sahifa 56** |

**Ok uslubi**

Parametr: **o'q uslubi**

Turi: str *,* ArrowStyle *,* ixtiyoriy

Mavjud o'q uslublari 3.4-jadvalda va 3.18-rasmda keltirilgan.

**3.4-jadval - Izohli o'qlarning uslublari**

**Sinf**

**Ism**

**Xususiyatlar**

Egri chiziq

-

Yo'q

CurveB

->

bosh uzunligi = 0,4, bosh kengligi = 0,2

Qavs B

- [

kenglikB = 1,0, uzunlikB = 0,2,

angleB = Yo'q

CurveFilledB

- |>

bosh uzunligi = 0,4, bosh kengligi = 0,2

Egri A

<-

bosh uzunligi = 0,4, bosh kengligi = 0,2

CurveAB

<->

bosh uzunligi = 0,4, bosh kengligi = 0,2

CurveFilledA

<| -

bosh uzunligi = 0,4, bosh kengligi = 0,2

CurveFilledAB <| - |>

bosh uzunligi = 0,4, bosh kengligi = 0,2

Qavs

] -

kenglik A = 1,0, uzunlik A = 0,2,

burchak A = Yo'q

QavslarAB

] - [

kenglik A = 1,0, uzunlik A = 0,2,

burchak A = Yo'q, kenglikB = 1.0,

uzunlikB = 0,2, burchakB = Yo'q

Chiroyli

chiroyli

bosh uzunligi = 0,4, bosh kengligi = 0,4,

quyruq kengligi = 0,4

Oddiy

oddiy

bosh uzunligi = 0,5, bosh kengligi = 0,5,

quyruq kengligi = 0,2

Takoz

xanjar

tail\_width = 0,3, shrink\_factor = 0,5

55

|  |
| --- |
| **57-bet** |

**Shakl 3.18 - Izohli o'qlarning uslublari**

Tasvirni yaratish uchun dastur kodi

3.18-rasm:

plt.figure ( figsize = ( 7 , 5 ))

strelkalar = [ **'-'** , **'->'** , **'- ['** , **'| - |'**, **'- |>'** , **'<-'** , **'<->'** , **'<| -'** , **'<| - |>'** ,

**"xayoliy"**, **"sodda"**, **"xanjar"**]

bbox\_properties = dict (

boxstyle = **'round, pad = 0,2'** ,

ec = **'k'** ,

fc = **'w'** ,

ls = **'-'** ,

lw = 1

)

ofs\_x = 0

ofs\_y = 0

56

|  |
| --- |
| **Sahifa 58** |

**uchun**I, AR **ichida** Enumerate (strelkalar):

**agar**i == 6 : ofs\_x = 0,5 bo'lsa

plt.annotate (ar, xy = ( 0.4 + ofs\_x, 0.92 -ofs\_y), xycoords = **'ma'lumotlar'** ,

xytext = ( 0,05 + ofs\_x, 0,9 -ofs\_y), textcoords = **'ma'lumotlar "** , Shrift = 17 ,

bbox = bbox\_properties,

arrowprops = dict ( arrowstyle = ar)

)

**agar**ofs\_y == 0.75 : ofs\_y = 0 bo'lsa

**else**: ofs\_y + = 0.15

**Ulagich uslubi**

Parametr: **ulanish uslubi**

Turi: str *,* ConnectionStyle *,* None *,* ixtiyoriy

Ushbu parametr xy va nuqtalarini bog'laydigan chiziq uslubini o'rnatadi

xycoords. Qiymat sinf ob'ektini olishi mumkin

ConnectionStyle yoki satr uslubini belgilaydigan satr

vergul bilan ajratilgan parametrlarga ega ulanishlar.

**3.5-jadval - Izohlovchi ulagichining uslublari**

**Sinf**

**Ism**

**Xususiyatlar**

Burchak

burchak

burchak A = 90, burchakB = 0, rad = 0,0

Burchak3

burchak 3.

burchak A = 90, burchak B = 0

Ark

yoy

burchak A = 0, burchak B = 0, armA = Hech narsa,

armB = Yo'q, rad = 0,0

Arc3

arc3

rad = 0,0

Bar

bar

armA = 0.0, armB = 0.0, kasr = 0.3,

burchak = Yo'q

57

|  |
| --- |
| **Sahifa 59** |

Quyida ishlash imkoniyatlarini namoyish etuvchi misol keltirilgan

Connectionstyle parametri bilan:

matematikani **import qilish**

fig, axs = plt.subplots ( 2 , 3 , figsize = ( 12 , 7 ))

conn\_style = [

**'burchak, burchak A = 90, burchakB = 0, rad = 0,0'**,

**"burchak3, burchak A = 90, burchakB = 0"**,

**'yoy, burchak A = 0, burchakB = 0, armA = 0, armB = 40, rad = 0.0'**,

**'arc3, ​​rad = -1.0'**,

**'bar, armA = 0.0, armB = 0.0, kasr = 0.1, burchak = 70'**,

**'bar, kasr = -0.5, burchak = 180'**,

]

**uchun**I **-yilda** qator ( 2 ):

**uchun**j **yilda** qator ( 3 ):

axs [i, j] .text ( 0.1 , 0.5 , **' \ n '** .join (conn\_style [i \* 3 + j] .split ( **','** )))

axs [i, j] .annotate ( **'text'** , xy = ( 0.2 , 0.2 ), xycoords = **'data'** ,

xytext = ( 0.7 , 0.8 ), textcoords = **'ma'lumotlar'** ,

arrowprops = dict ( arrowstyle = **'->'** ,

connectionstyle = conn\_style [i \* 3 + j]))

**Shakl 3.19 - Izohli ulanish chizig'ining uslublari**

58

|  |
| --- |
| **Sahifa 60** |

**3.4 Matn sinfining xususiyatlari**

Matplotlib.text.Text sinfining xususiyatlarini ko'rib chiqing

matn elementlarini aniq sozlash uchun kirishni ta'minlash. Biz emas

biz Matn sinfining barcha xususiyatlarini ko'rib chiqamiz, umumiy nuqtai nazarni tuzamiz

eng ko'p ishlatiladigan.

**3.4.1 Matnni namoyish qilish uchun javobgar parametrlar**

Matnni namoyish qilish uchun javobgar bo'lgan parametrlar sizga sozlash imkonini beradi

shaffoflik, rang, shrift va boshqalar:

• alfa: float

◦ dekalning shaffoflik darajasi. Parametr raqam bilan belgilanadi

0 dan 1,0 gacha - to'liq shaffoflik, 1 - to'liq

xiralik.

• rang: rangni sozlashning mavjud usullaridan biri (bo'limga qarang

" *2.3.2 chiziq rangi* ")

◦ Matn rangi. Parametr qiymati xuddi shunday turga ega

uchastkaning rangi uchun mas'ul bo'lgan fitna funktsiyasining parametri.

• fontfamily (yoki oila): str

The To'plam qatori sifatida ko'rsatilgan matn shrifti: {'serif',

'sans-serif', 'cursive', 'fantasy', 'monospace'}. Mumkin

shriftingizdan foydalaning.

• shrift (yoki o'lcham): str, int

◦ Shrift kattaligi, siz quyidagilarni tanlashingiz mumkin: {'xx-small', 'x-

kichik ',' kichik ',' o'rta ',' katta ',' x-katta ',' xx-

katta '} yoki raqamli qiymat sifatida o'rnatiladi.

59

|  |
| --- |
| **Sahifa 61** |

• fontstyle (yoki uslub): str

The Shrift uslubi, to'plamdan: {'normal', 'kursiv',

'qiyshiq'}.

• fontvariant (yoki variant): str

Set Shrift uslubi, to'plamdan o'rnatilgan: {'normal', 'small-

bosh harflar '}.

• shrift (yoki vazn): str

The Shriftning to'yinganligi, to'plamdan: {'ultralight',

'engil', 'normal', 'muntazam', 'kitob', 'o'rta',

'roman', 'semibold', 'demibold', 'demi', 'qalin',

'og'ir', 'qo'shimcha qalin', 'qora'} yoki raqamli qiymat

0-1000 oralig'ida .

Ro'yxatda keltirilganlardan foydalanishni ko'rsatadigan bir misolni ko'rib chiqaylik

yuqoridagi parametrlar:

plt.title ( **"Sarlavha"** , alfa = 0,5 , rang = **«r»** , Shrift = 18 , FontStyle = **'kursiv'** ,

fontweight = **'qalin'** , chiziqlar oralig'i = 10 )

plt.plot ( oraliq ( 0 , 10 ), oraliq ( 0 , 10 ))

**3.20-rasm - Text sinfining xususiyatlaridan foydalanishga misol**

60

|  |
| --- |
| **62-bet** |

Xususiyatlarni guruhli sozlash uchun siz parametrdan foydalanishingiz mumkin

fontproperties yoki font\_properties, bu qiymat sifatida

font\_manager.FontProperties sinf ob'ekti o'tkazildi.

FontProperties sinf konstruktori quyidagicha ko'rinadi:

FontProperties (oila = **Yo'q** , uslub = **Yo'q** , variant = **Yo'q** , vazn = **Yo'q** ,

streç = **Yo'q** , o'lcham = **Yo'q** , fname = **Yo'q** )

Konstruktor parametrlari:

• oila: ko'ch

Ont Shrift nomi.

• stil str

◦ Shrift uslubi.

• variant str

Sc Yozuv.

• strest str

Ont Shrift kengligi.

• vazn st

◦ Shriftlarning to'yinganligi.

• hajmi str

◦ Shrift o'lchami.

FontProperties-ni ishlatishdan oldin uni unutmang

Import:

**dan**matplotlib.font\_manager **import**FontProperties

plt.title ( **'Sarlavha'** , fontproperties = FontProperties ( family = **'monospace'** ,

uslub = **'kursiv'** , vazn = **'og'ir'** , o'lcham = 15 ))

plt.plot ( oraliq ( 0 , 10 ), oraliq ( 0 , 10 ))

61

|  |
| --- |
| **63-bet** |

**3.21-rasm - fontproperties parametri bilan ishlashga misol**

**3.4.2 Yorliqning joylashuvi uchun javobgar parametrlar**

Yorliq uchun siz hizalama, holat, aylanish va *z* belgilashingiz mumkin -

buyurtma:

• gorizontal yo'nalish (yoki ha): str

◦ Gorizontal tekislash. To'plamdan o'rnatish: {'center',

"o'ng", "chap"} *.*

• vertikal tekislash (yoki va): str

◦ Vertikal tekislash. {'Center' to'plamidan o'rnatish,

'top', 'bottom', 'boshlang'ich', 'center\_baseline'} *.*

• pozitsiya: (float, float)

◦ Yorliqning holati. *X* va *y* ikkita koordinatalar bilan belgilanadi ,

pozitsiya parametriga kassetali sifatida uzatiladi

ikkita element.

• aylanish: float yoki str

◦ burilish. Yorliq yo'nalishi matn sifatida o'rnatilishi mumkin

{'vertikal', 'gorizontal'} yoki raqamli - qiymati

daraja.

62

|  |
| --- |
| **64-bet** |

• aylanish\_mode: str

Ation Aylantirish rejimi. Ushbu parametr tartibini belgilaydi

aylanish va tekislash. Agar u "standart" bo'lsa

birinchi navbatda aylanish, so'ngra tekislash amalga oshiriladi. Agar a

"langar" ga teng, keyin aksincha.

• zorder: suzuvchi

◦ Joylashtirish tartibi. Parametr qiymati aniqlanadi

elementlarning ko'rsatilish tartibi. Minimal element

birinchi navbatda zorder qiymati ko'rsatiladi.

Vazifa parametrlaridan foydalangan holda sarlavha misolini ko'rib chiqamiz

Manzil:

plt.title ( **"Sarlavha"** , Shrift = 17 , lavozim = ( 0,7 , 0,2 ), aylanish = **'vertikal'** )

plt.plot ( oraliq ( 0 , 10 ), oraliq ( 0 , 10 ))

**Shakl 3.22 - Parametrlarni sozlashdan foydalanish misoli**

**Manzil**

63

|  |
| --- |
| **Sahifa 65** |

**3.4.3 Fonni o'rnatish uchun javobgar parametrlar**

**yozuv**

Parametr yozuv fonini o'rnatish uchun javobgardir:

• orqa rang: rang

◦ Fon rangi.

Agar rang, qalinlik ko'rsatkichi bilan aniqroq sozlash zarur bo'lsa,

ramka turi, asosiy plomba rangi va boshqalar, keyin parametrdan foydalaning

bbox, uning qiymati kalitlari xossalari bo'lgan lug'atdir

sinf yamalari *.*FancyBboxPatch (3.6-jadvalga qarang) *.*

**3.6-jadval - yamalar sinfining xususiyatlari *.*FancyBboxPatch**

**Mulk**

**Qiymat turi**

**Tavsif**

boxstyle

str yoki

matplotlib.patches.

BoxStyle

Kadrlar uslubi. Sm.

3.7-jadval.

alfa

suzuvchi yoki Yo'q

Shaffoflik.

rang

rang

Rang.

edgecolor yoki ec Color, yo'q yoki "auto"

Ramka chegarasi rangi.

yuz rangi yoki fc rangi yoki yo'q

Rangni to'ldiring.

to'ldirish

bool

To'g'ri - foydalanish

to'ldirish, Yolg'on - yo'q.

lyuk

{'/', '\', '|', '-', '+',

'x', 'o', 'O', '.', '\*'}

Tugatish.

linestyle yoki ls {'-', '-', '-.', ':',

", (ofset, on-off-seq),

...}

Frame layn uslubi.

linewidth yoki lw float yoki Hech narsa

Chiziq kengligi.

64

|  |
| --- |
| **Sahifa 66** |

**3.7-jadval - Boxstyle parametrlari**

**Sinf**

**Ism**

**Xususiyatlar**

**Tashqi ko'rinish**

Doira

doira

pad = 0,3

DArrow

darrow

pad = 0,3

LArrow

tomoq

pad = 0,3

RArrow

siyrak

pad = 0,3

Dumaloq

dumaloq

yostiq = 0,3,

rounding\_size = Hech narsa

4-tur

dumaloq4

yostiq = 0,3,

rounding\_size = Hech narsa

Dumaloq tish

yostiq = 0,3,

tish\_size = Hech narsa

Sawtooth

arra tishlari

yostiq = 0,3,

tish\_size = Hech narsa

Kvadrat

kvadrat

pad = 0,3

Yozuvning foniga misol:

matplotlib.patches- **dan**FancyBboxPatch-ni **import**qiladi

bbox\_properties = dict (

boxstyle = **"rarrow, pad = 0.3"** ,

ec = **'g'** , fc = **'r'** ,

ls = **'-'** ,

lw = 3

)

(plt.title **"Sarlavha"** , Shrift = 17 , bbox = bbox\_properties, o'rnini = ( 0.5 ,

0.85 ))

plt.plot ( oraliq ( 0 , 10 ), oraliq ( 0 , 10 ))

65

|  |
| --- |
| **67-bet** |

**3.23-rasm - Yozuvning fonini o'rnatish misoli**

**3.5 Ranglar paneli - rangli *panel***

Agar siz ranglarni taqsimlashni rejalashtirmoqchi bo'lsangiz

colormesh (), pcolor (), imshow () va boshqalar, keyin namoyish qilish uchun

sizga kerak bo'lishi mumkin rang va raqamli qiymat

*Matplotlib-da colorbar* deb nomlangan afsonaning *analogi* . Yarataylik

np.random.rand () va displey yordamida tasodifiy tarqatish

pcolor () orqali:

np **sifatida**numpy- **ni import qilish**

np.random.seed ( 123 )

vals = np.random.randint ( 10 , size = ( 7 , 7 ))

plt.pcolor (vals)

**3.24-rasm - Ranglarni taqsimlash**

66

|  |
| --- |
| **Sahifa 68** |

*Keling* , ushbu to'plam uchun *ranglar panelini yarataylik* :

np.random.seed ( 123 )

vals = np.random.randint ( 10 , size = ( 7 , 7 ))

plt.pcolor (vals)

plt.colorbar ()

**3.25-rasm - Berilgan rang uchun ranglar paneli**

**tarqatish**

Rang satrida ranglarni alohida ajratish uchun sizga kerak

kerakli rang sxemasini taqdim etish

tegishli funktsiyaga cmap parametri (bizning holatimizda, pcolor ()):

np.random.seed ( 123 )

vals = np.random.randint ( 10 , size = ( 7 , 7 ))

plt.pcolor (vals, cmap = plt.get\_cmap ( **'viridis'** , 11 ))

plt.colorbar ()

**Shakl 3.26 - Diskret rang ajratish bilan ranglar paneli**

67

|  |
| --- |
| **69-bet** |

**3.5.1 inset\_locator () yordamida umumiy sozlash**

Rang satrini sozlash uchun variantlardan biri

asosiy Axes elementi asosida o'zingizni yarating va o'zgartiring

uning parametrlarining bir qismi. Buni funktsiya yordamida bajarish qulay

mpl\_toolkits.axes\_grid1.inset\_locator-dan inset\_axes (). Asosiy

uning dalillari 3.7-jadvalda keltirilgan.

**3.7-jadval - inset\_axes () funktsiyasining parametrlari**

**Parametr**

**Turi**

**Tavsif**

ota\_axslar

O'qlar

Ota-onalar o'qlari ob'ekti.

kengligi

float yoki str

Ob'ektning kengligi. O'rnatilgan

ota-onalarning ulushi

ob'ekt yoki mutlaq

raqam sifatida qiymat.

balandlik

float yoki str

Ob'ektning balandligi. O'rnatilgan

ota-onalarning ulushi

ob'ekt yoki mutlaq

raqam sifatida qiymat.

lok

int yoki string *,*

ixtiyoriy *,*

qiymati

standart: 1

Mulk joylashgan joy.

To'plamdan qiymatni oladi:

'yuqori o'ng': 1,

'yuqori chap': 2,

"pastki chap": 3,

"pastki o'ng": 4,

"o'ng": 5,

"chap markaz": 6,

"o'ng tomonda": 7,

"pastki markaz": 8,

'yuqori markaz': 9,

"markaz": 10

68

|  |
| --- |
| **70-bet** |

bbox\_to\_anchor tuple yoki

matplotlib.trans

shakllari.BboxBase

yoki ixtiyoriy

Joylashuvi va nisbati

ob'ektning tomonlari. O'rnatilgan

format (chap burchak, pastki qism)

burchak, kenglik, balandlik), yoki

(chap burchak, pastki burchak).

bbox\_transform matplotlib.trans

shakllari. Transformatsiya

yoki ixtiyoriy

Ob'ektni o'zgartirish.

chegara paneli

suzuvchi yoki

ixtiyoriy

Bbox\_to\_anchor va orasidagi bo'shliq

ob'ekt.

Keling, inset\_axes () bilan qanday ishlashni misol bilan namoyish qilaylik:

**dan**mpl\_toolkits.axes\_grid1.inset\_locator **import**inset\_axes

np.random.seed ( 123 )

vals = np.random.randint ( 11 , size = ( 7 , 7 ))

fig, ax = plt.subplots ()

gr = ax.pcolor (vals)

axins = inset\_axes (ax, width = **"7%"** , height = **"50%"** , loc = **'pastki chap' '** ,

bbox\_to\_anchor = ( 1.05 , 0. , 1 , 1 ), bbox\_transform = ax.transAxes,

chegara paneli = 0 )

plt.colorbar (gr, cax = axins)

**3.27-rasm - yordamida qurilgan ranglar paneli**

**inset\_axes ()**

69

|  |
| --- |
| **71-bet** |

Agar kerak bo'lsa, siz ranglar paneli shkalasini o'zgartirishingiz mumkin

Tick ​​sinfidagi ob'ektdan foydalanish.

**3.5.2 O'lchovni o'rnatish va yorliqni o'rnatish**

O'zingizning o'lchovingizni belgilash uchun siz ro'yxatni topshirishingiz kerak

parametr orqali colorbar () funktsiyasiga masshtab elementlarining qiymatlari

Shomil. O'lchov yorlig'i yorliq parametri yordamida o'rnatiladi

colorbar () funktsiyalari.

Keling, oldingi misoldagi oxirgi qatorni quyidagilar bilan o'zgartiramiz

yo'l:

plt.colorbar (gr, cax = axins, ticks = [ 0 , 5 , 10 ], label = **'Qiymat'** )

**Shakl 3.28 - O'zining shkalasi bilan rang paneli**

70

|  |
| --- |
| **72-bet** |

Agar matn yorliqlarini o'rnatishga ehtiyoj bo'lsa, unda

set\_yticklabels () funktsiyasidan foydalaning:

cbar = plt.colorbar (gr, cax = axins, ticks = [ 0 , 5 , 10 ], label = **'Qiymat'** )

cbar.ax.set\_yticklabels ([ **'Past'** , **'Medium'** , **'High'** ])

**3.29-rasm - Matn yorliqli ranglar paneli**

**3.5.3 Qo'shimcha rang sozlamalari**

**chiziqlar**

Rang ko'rinishini sozlash uchun bir qator parametrlarni ko'rib chiqamiz

colorbar () funktsiyasi uchun argument sifatida mavjud bo'lgan satrlar.

**3.8-jadval - colorbar () funktsiyasining parametrlari**

**Mulk**

**Turi**

**Tavsif**

yo'nalish vertikal yoki

gorizontal

Yo'nalish.

kichraytirish

suzmoq

Ranglarni masshtablash

chiziqlar.

uzaytirish

['na' | | "ikkalasi ham"

| 'min' | 'max']

Ko'rsatkich holati

o'lchovni kengaytirish.

71

|  |
| --- |
| **73-bet** |

kengaytirmoq

[Hech kim | 'avtomatik' |

uzunlik | uzunliklar]

Kengaytma ko'rsatkichining o'lchami

tarozi.

chizmalar

bool

Displey

bo'linadigan panjara

rang paneli.

np **sifatida**numpy- **ni import qilish**

np.random.seed ( 123 )

vals = np.random.randint ( 10 , size = ( 7 , 7 ))

plt.pcolor (vals, cmap = plt.get\_cmap ( **'viridis'** , 11 ))

plt.colorbar ( orientatsiya = **'gorizontal'** ,

shrink = 0,9 , kengaytirish = **'max'** , extensionfrac = 0,2 ,

extensionrect = **False** , chizmalar = **False** )

**Shakl 3.30 - Qo'shimcha parametrlarga ega rang paneli**

72

|  |
| --- |
| **74-bet** |

**4-bob. Ma'lumotlarni vizuallashtirish**

**4.1 Chiziqli diagramma**

Chiziqli diagramma eng ko'p ishlatiladigan turlardan biridir

ma'lumotlarni vizualizatsiya qilish uchun grafikalar. Biz buni ishlatganmiz

oldingi *darslarda Matplotlibning* imkoniyatlarini namoyish etish , bunda

darsda biz uni sozlash imkoniyatlarini batafsil ko'rib chiqamiz

tashqi ko'rinish.

**4.1.1 Plotirovka qilish**

Chiziqli grafikani chizish uchun, () funktsiyasidan foydalaning

quyidagi imzo bilan:

fitna ([x], y, [fmt], \*, ma'lumotlar = **Yo'q** , \*\* kvarglar)

fitna ([x], y, [fmt], [x2], y2, [fmt2], ..., \*\* kwargs)

Agar plot () funktsiyasini bitta argument bilan chaqirsangiz:

Uchastka (y), keyin biz ordinatali ( *y* o'qi ) grafigini olamiz

o'tkazilgan ro'yxatdagi qiymatlar abssissa o'qi bo'ylab ( *x* o'qi ) chizilgan -

massiv elementlari indekslari.

Plot () funktsiyasining argumentlarini ko'rib chiqing:

• x, x2, ...: massiv

First Abscissa ma'lumotlar to'plamlari ( *x* o'qi ) birinchi, ikkinchi va

va boshqalar. grafik san'at.

• y, y2, ...: qator

◦ *Y-* Ma'lumotlarni guruhlar eksa uchun birinchi, ikkinchi va

va boshqalar. grafik san'at.

73

|  |
| --- |
| **Sahifa 75** |

• fmt: str

Ot Plot formati. Qator sifatida belgilash: '[marker] [chiziq]

[rang] '.

• \*\* kvarg

◦ sinf xususiyatlari

Ga kirishni ta'minlaydigan Line2D

jadvalning ko'rinishi uchun juda ko'p sozlamalar,

eng foydali bo'lganlar 4.1-jadvalda keltirilgan.

**4.1-jadval - Line2D sinfining xususiyatlari**

**Mulk**

**Turi**

**Tavsif**

alfa

suzmoq

Shaffoflik.

rang yoki v

rang

Rang.

fillstyle

{'to'liq', 'chap',

"o'ng", "pastki",

'top', 'none'}

To'ldirish uslubi.

yorliq

ob'ekt

Matn yorlig'i.

linestyle yoki ls

{'-', '-', '-.',

':', '', (ofset,

off-seq), ...}

Chiziq uslubi.

chiziq kengligi yoki lw

suzmoq

Chiziq kengligi.

marker

matplotlib.markers

O'q uslubi.

markeredgecolor yoki

mec

rang

Chegaraning rangi

marker.

markeredgewidth yoki

mew

suzmoq

Chegaraning qalinligi

marker.

markerfacecolor yoki

mfc

rang

Rangni to'ldiring

marker.

markersize yoki ms

suzmoq

Marker hajmi.

74

|  |
| --- |
| **Sahifa 76** |

**4.1.1.1 fmt argumenti uchun parametrlar**

Fmt argumenti quyidagi formatga ega: '[marker] [line] [color]'

• marker: str

◦ marker turini belgilaydi, qiymatlardan birini olishi mumkin,

4.2-jadvalda keltirilgan.

**4.2-jadval - marker turi**

**Belgilar**

**Tavsif**

'.'

Nuqta *belgisi* .

','

Piksel ( *piksel belgisi* ).

"o"

Doira ( *doira belgisi* ).

"v"

Pastga *uchburchak* ( *triangle\_down*

*marker* ).

'^'

Uchburchak yuqoriga qarab ( *uchburchak\_up belgisi* ).

'<'

Chapga yo'naltirilgan *uchburchak* ( *triangle\_left marker* ).

'>'

To'g'ri *uchburchak* ( *triangle\_right*

*marker* ).

'1'

Pastga yo'naltirilgan uchburchak ( *tri\_down marker* ).

"2"

Uchburchak yuqoriga qarab ( *tri\_up belgisi* ).

"3"

Chapga yo'naltirilgan uchburchak ( *tri\_left marker* ).

"4"

To'g'ri uchburchak ( *tri\_right marker* ).

bu

Kvadrat ( *kvadrat marker* ).

"p"

Pentagon *markeri* .

'\*'

Yulduz ( *yulduz belgisi* ).

"h"

Olti burchakli (olti burchakli *marker* ).

"H"

Olti burchakli (olti burchakli *marker* )

"+"

Plyus ( *ortiqcha marker* ).

"x"

*X* shaklidagi marker ( *x marker* ).

"D"

Romb ( *olmos belgisi* ).

75

|  |
| --- |
| **Sahifa 77** |

"d"

*Romb* ( *ingichka\_ olmosli marker).*

'|'

Vertikal chiziq ( *vline marker* ).

'\_'

Landshaft chiziq ( *hline marker* ).

• qator: str

◦ Chiziq uslubi.

**4.3-jadval - Chiziq uslubi**

**Belgilar**

**Tavsif**

'-'

Qattiq *chiziq uslubi* .

'-'

Punktir *chiziq uslubi* .

'-.'

Buzilgan chiziq ( *dash-Access nuqta chizig'i uslubi* ).

':'

Kesik chiziq ( *nuqta chiziq uslubi* ).

• rang

◦ Grafik rangi.

**4.4-jadval - Grafik rangi**

**Belgilar**

**Tavsif**

"b"

Moviy.

"g"

Yashil.

"r"

Qizil.

"c"

Turkuaz.

"m"

Binafsha (qizil).

"y"

Sariq.

"k"

Qora.

"w"

Oq.

76

|  |
| --- |
| **78-bet** |

Keling, plot () ning imkoniyatlarini misol bilan namoyish etamiz:

x = [ 1 , 5 , 10 , 15 , 20 ]

y1 = [ 1 , 7 , 3 , 5 , 11 ]

y2 = [ 4 , 3 , 1 , 8 , 12 ]

plt.figure ( figsize = ( 12 , 7 ))

plt.plot (x, y1, **'o-r'** , alfa = 0.7 , label = **'first'** , lw = 5 , mec = **'b'** , mew = 2 ,

ms = 10 )

plt.plot (x, y2, **'v-.g'** , label = **'second'** , mec = **'r'** , lw = 2 , mew = 2 , ms = 12 )

plt.legend ()

plt.grid ( **To'g'ri** )

**4.1-rasm - Uchastka () yordamida qurilgan grafikalar**

Chiziqli diagramma bilan ishlashning turli xil variantlarini ko'rib chiqamiz.

**4.1.2 Grafika va o'qi orasidagi maydonni to'ldirish**

Fill\_between () funktsiyasi maydonlarni to'ldirish uchun ishlatiladi. Imzo

funktsiyalari:

fill\_between (x, y1, y2 = 0 , bu erda = **Yo'q** , interpolate = **False** , qadam = **Yo'q** , \*,

ma'lumotlar = **Yo'q** , \*\* kvarg)

77

|  |
| --- |
| **79-bet** |

Funktsiyaning asosiy parametrlari:

• x: *N* uzunlikdagi massiv

◦ Abscissa o'qi uchun ma'lumotlar to'plami ( *x-* o'qi ).

• y1: *N* uzunlikdagi massiv yoki skaler qiymat

◦ *Ordinata* o'qi ( *y* o'qi ) uchun ma'lumotlar to'plami birinchi egri *chiziqdir* .

• y2: *N* uzunlikdagi massiv yoki skaler qiymat

◦ *Ordinata* o'qi ( *y* o'qi ) uchun ma'lumotlar to'plami ikkinchi egri *chiziqdir* .

• bu erda: bool elementlari massivi (uzunligi *N* ), ixtiyoriy, qiymati bo'yicha

sukut bo'yicha: yo'q

◦ Belgilangan rang bilan to'ldirilgan mintaqani belgilaydi

koordinatalari x [qaerda]: interval x [i] va bilan to'ldiriladi

x [i + 1] agar bu erda [i] va qaerda [i + 1] bo'lsa, to'g'ri.

• qadam: {'pre', 'post', 'mid'} *,* ixtiyoriy

◦ Agar *qadam* funktsiyasi ishlatilgan bo'lsa, *qadamni* belgilaydi

grafik displey (ulardan birida muhokama qilinadi

quyidagi bo'limlar).

• \*\* kvarg

◦ Ko'pburchak sinfining xususiyatlari *.*

Keling, tajriba uchun ma'lumotlar to'plamini yarataylik:

np **sifatida**numpy- **ni import qilish**

x = np.aranjir ( 0,0 , 5 , 0,01 )

y = np.cos (x \* np.pi)

Grafani to'ldirish bilan namoyish qilaylik:

plt.plot (x, y, c = **'r'** )

plt.fill\_bet (x, y)

78

|  |
| --- |
| **80-bet** |

**4.2-rasm - To'ldirilgan grafik (1-misol)**

To'ldirish qoidalarini o'zgartiramiz:

plt.plot (x, y, c = **'r'** )

plt.fill\_bet (x, y, bu erda = (y> 0,75 ) | (y <- 0,75 ))

**4.3-rasm - To'ldirilgan grafik (2-misol)**

79

|  |
| --- |
| **81-sahifa** |

Y1 va y2 parametrlaridan foydalanib *,* siz murakkabroq bo'lishingiz mumkin

echimlar. *Y* > = 0 sharti bilan maydonni 0 dan *y* gacha to'ldirish :

plt.plot (x, y, c = **'r'** )

plt.fill\_between (x, y, bu erda = (y> 0 ))

**4.4-rasm - To'ldirilgan grafik (3-misol)**

*Y* > = 0,5 sharti bilan maydonni 0,5 dan *y* gacha to'ldiring :

plt.plot (x, y, c = **'r'** )

plt.grid ()

plt.fill\_between (x, 0,5 , y, bu erda = y> = 0,5 )

**4.5-rasm - To'ldirilgan grafik (4-misol)**

80

|  |
| --- |
| **Sahifa 82** |

*Y* va 1 orasidagi maydonni to'ldirish :

plt.plot (x, y, c = **'r'** )

plt.grid ()

plt.fill\_bet (x, y, 1 )

**4.6-rasm - To'ldirilgan grafik (5-misol)**

Ikki rangli to'ldirish opsiyasi:

plt.plot (x, y, c = **'r'** )

plt.grid ()

plt.fill\_between (x, y, bu erda = y> = 0 , color = **'g'** , alfa = 0.3 )

plt.fill\_between (x, y, bu erda = y <= 0 , color = **'r'** , alfa = 0.3 )

**4.7-rasm - To'ldirilgan grafik (6-misol)**

81

|  |
| --- |
| **83-bet** |

**4.1.3 Grafik belgilarini sozlash**

Ushbu bo'limning boshida biz qachon markerlar bilan ishlashga misol keltirdik

grafiklarni namoyish qilish. Keling, yana takrorlaymiz, ammo soddalashtirilgan usulda.

shakl:

x = [ 1 , 2 , 3 , 4 , 5 , 6 , 7 ]

y = [ 7 , 6 , 5 , 4 , 5 , 6 , 7 ]

plt.plot (x, y, marker = **'o'** , c = **'g'** )

**4.8-rasm - Belgilangan grafik**

Ma'lumotlar to'plamini yarataylik:

np **sifatida**numpy- **ni import qilish**

x = np.aranjir ( 0,0 , 5 , 0,01 )

y = np.cos (x \* np.pi)

Undagi ochkolar soni 500 tani tashkil etadi, shuning uchun yuqoridagilar

yondashuv qo'llanilmaydi: fikrlar bir-biriga to'g'ri keladi:

plt.plot (x, y, marker = **'o'** , c = **'g'** )

82

|  |
| --- |
| **84-bet** |

**4.9-rasm - Ko'p sonli markerlar bilan grafik**

Bunday holda siz markerlarni ko'rsatish oralig'ini o'rnatishingiz kerak

birini tanlashi mumkin bo'lgan markevery parametri ishlatiladi

quyidagi qiymatlar:

• Yo'q - har bir nuqta ko'rsatiladi;

• N - har bir *N-chi* nuqta ko'rsatiladi ;

• (start, N) - nuqtadan boshlab har bir *N-chi* nuqta ko'rsatiladi

*boshlash* ;

• tilim (start, end, N) - har *N-chi* nuqtani ko'rsatadi

bir interval *start* uchun *oxirida* ;

• [i, j, m, n] - faqat *i, j, m, n* nuqtalari ko'rsatiladi .

83

|  |
| --- |
| **Sahifa 85** |

Quyida markevery qanday ishlashini ko'rsatuvchi misol keltirilgan:

x = np.aranjir ( 0,0 , 5 , 0,01 )

y = np.cos (x \* np.pi)

m\_ev\_case = [ **Yo'q** , 10 , ( 100 , 30 ), bo'lak ( 100 , 400 , 15 ), [ 0 , 100 , 200 , 300 ],

[ 10 , 50 , 100 ]]

fig, ax = plt.subplots ( 2 , 3 , figsize = ( 10 , 7 ))

axs = [ax [i, j] **uchun** I **-yilda** qator ( 2 ) **uchun** J **yilda** qator ( 3 )]

**uchun**I, hodisa **ham** enumerate (m\_ev\_case):

akslar [i] .set\_title ( str (ish))

axs [i] .plot (x, y, **'o'** , ls = **'-'** , ms = 7 , markevery = case)

**Shakl 4.10 - Belgilashning turli xil variantlari**

84

|  |
| --- |
| **Sahifa 86** |

**4.1.4 Diagrammani kesish**

Grafikning faqat unga mos keladigan qismini ko'rsatish uchun

muayyan holat, oldindan maskalanishdan foydalaning

numpy paketidan masked\_where () funktsiyasidan foydalangan holda ma'lumotlar:

x = np.aranjir ( 0,0 , 5 , 0,01 )

y = np.cos (x \* np.pi)

y\_masked = np.ma.masked\_ qaerda (y <- 0,5 , y)

plt.ylim (- 1 , 1 )

plt.plot (x, y\_masked, chiziq kengligi = 3 )

**4.11-rasm - Grafikni qirqishga misol**

85

|  |
| --- |
| **Sahifa 87** |

**4.2 Step, stack, point va boshqalar**

**grafikalar**

**4.2.1 Bosqich grafigi**

Step () funktsiyasi yordamida qadam chizmasi tuziladi:

qadam (x, y, [fmt], \*, ma'lumotlar = **Yo'q** , bu erda = **'pre'** , \*\* kwargs)

Funktsiya parametrlari:

• x: qator

◦ Abscissa o'qi uchun ma'lumotlar to'plami ( *x-* o'qi ).

• y: qator

◦ *Ordinata* o'qi uchun ma'lumotlar to'plami (o'qi o'qi ).

• fmt: str *,* ixtiyoriy

◦ Chiziq formati (plot () funktsiyasiga qarang).

• ma'lumotlar: indekslangan ob'ekt *,* ixtiyoriy

◦ teglar.

• bu erda: {'pre', 'post', 'mid'} *,* ixtiyoriy; qiymati

standart: 'oldindan'

The Qadam o'rnatiladigan joyni aniqlaydi:

▪ 'pre': *y* qiymati *x* qiymatining chap tomoniga qo'yiladi , ya'ni. qiymat

y [i] (x [i-1]; x [i]) oralig'i uchun aniqlanadi *;*

▪ 'post': *y* qiymati *x* qiymatining o'ng tomoniga joylashtiriladi , ya'ni.

y [i] qiymati (x [i]; x [i + 1]) oralig'i uchun aniqlanadi *;*

▪ 'mid': *y* qiymati interval o'rtasiga joylashtirilgan.

86

|  |
| --- |
| **88-bet** |

x = np.arange ( 0 , 7 )

y = x

where\_set = [ **'pre'** , **'post'** , **'mid'** ]

fig, axs = plt.subplots ( 1 , 3 , figsize = ( 15 , 4 ))

**uchun**I, ax **yilda** Enumerate (axs):

ax.step (x, y, **'g-o'** , bu erda = qaerda\_set [i])

ax.grid ()

**4.12-rasm - Bosqichli grafik**

**4.2.2 Stack chart**

Stackplot () funktsiyasi stack fitnesini yaratish uchun ishlatiladi.

Uning mohiyati shundaki, grafikalar bir-birining ustiga va har birida namoyish etiladi

keyingisi oldingi va berilganlarning yig'indisi:

x = np.arange ( 0 , 11 , 1 )

y1 = np.array ([(- 0.2 ) \* i \*\* 2 + 2 \* i **uchun** i **yilda** x])

y2 = np.array ([(- 0,4 ) \* i \*\* 2 + 4 \* i **uchun** i **yilda** x])

Y3 = np.array ([ 2 \* i **uchun** i **yilda** x])

teglar = [ **'y1'** , **'y2'** , **'y3'** ]

fig, ax = plt.subplots ()

ax.stackplot (x, y1, y2, y3, teglar = yorliqlar)

ax.legend ( loc = **'yuqori chap'** )

87

|  |
| --- |
| **89-bet** |

**4.13-rasm - Stack chart**

*Y2* mintaqaning yuqori qirrasi to'plamlardan olingan qiymatlar yig'indisi sifatida aniqlanadi

*y1* va *y2* , *y3* mos ravishda *y1* , *y2* va *y3* yig‘indisi .

**4.2.3 *Poytaxt* diagrammasi**

Vizual ravishda, *ildiz* grafasi v nuqtadan olingan qatorlar to'plamiga o'xshaydi

koordinatalari ( *x* , *y* ) yuqori qismga, uning yuqori qismida joylashgan

marker:

x = np.arange ( 0 , 10.5 , 0.5 )

y = np.array ([(- 0.2 ) \* i \*\* 2 + 2 \* i **uchun** i **yilda** x])

plt.stem (x, y)

**Rasm 4.14 - *Stem-* jadvali**

88

|  |
| --- |
| **90-bet** |

Stem () funktsiyasi uchun qo'shimcha parametrlar:

• linefmt: str *,* ixtiyoriy

Line vertikal chiziq uslubi.

**Jadval 4.5 - vertikal chiziq uslubi**

**Belgilar**

**Chiziq uslubi**

'-'

Qattiq *chiziq uslubi* .

'-'

Punktir *chiziq uslubi* .

'-.'

Buzilgan chiziq ( *dash-Access nuqta chizig'i uslubi* ).

':'

Kesik chiziq ( *nuqta chiziq uslubi* ).

• markerfmt: str *,* ixtiyoriy

◦ Marker formati.

**4.6-jadval - Marker formati**

**Qiymat**

**Tavsif**

"o"

Doira ( *Doira* ).

"+"

*Plyus* belgisi ( *Plyus belgisi* ).

'\*'

Yulduzcha ( *yulduzcha* ).

'.'

Nuqta ( *nuqta* ).

"x"

Xoch ( *xoch* ).

'kvadrat' yoki 's'

Kvadrat ( *maydon* ).

"olmos" yoki "d" *olmos* .

'^'

Pastga *uchburchak* ( *triangle\_down* ).

"v"

Yuqoriga yo'naltirilgan *uchburchak* ( *triangle\_up* ).

'<'

Chapga yo'naltirilgan *uchburchak* ( *triangle\_left* ).

'>'

To'g'ri yo'naltirilgan *uchburchak* ( *triangle\_right* ).

'pentagram' yoki

"p"

Pentagon ( *Besh qirrali yulduz (pentagram)* ).

89

|  |
| --- |
| **91-bet** |

"olti burchakli" yoki "h" olti burchakli ( *Olti burchakli yulduz (olti* burchakli *)* ).

"yo'q"

Marker *yo'q* (Belgilagichlar *yo'q* ).

• basefmt: str *,* ixtiyoriy

◦ boshlang'ich formati.

• pastki: suzuvchi *,* ixtiyoriy; standart qiymat *: 0*

◦ *y-* boshlang'ich koordinatasi.

Qo'shimcha parametrlar bilan ishlashni ko'rsatadigan misol:

plt.stem (x, y, linefmt = **'r--'** , markerfmt = **'^'** , pastki = 1 )

**4.15-rasm - O'zgartirilgan *shtematik* jadval**

90

|  |
| --- |
| **92-bet** |

**4.2.4 nuqta uchastkasi (tarqoq uchastka)**

Scatter () funktsiyasi tarqalish chizmasini qurish uchun ishlatiladi.

Eng sodda ko'rinishda, siz uni scatter () funktsiyasidan o'tib olishingiz mumkin

*x* va *y* koordinatalari :

x = np.arange ( 0 , 10.5 , 0.5 )

y = np.cos (x)

plt.scatter (x, y)

**P isunok 4.16 - ajratish diagrammasi (1-misol)**

Tasvirni batafsilroq sozlash uchun sizga kerak

scatter () funktsiyasining qo'shimcha parametrlaridan foydalaning *.*

Funktsional qo'ng'iroq imzosi:

tarqalish (x, y, s = **yo'q** , c = **yo'q** , marker = **yo'q** , smap = **yo'q** , norm = **yo'q** ,

vmin = **Yo'q** , vmax = **Yo'q** , alfa = **Yo'q** , chiziq kengliklari = **Yo'q** , verts = **Yo'q** ,

edgecolors = **yo'q** , \*, plotnonfinite = **yolg'on** , ma'lumotlar = **yo'q** , \*\* kwargs)

Keling, uning ba'zi parametrlarini ko'rib chiqaylik:

• x: massiv *,* shakli (n,)

◦ Abscissa o'qi uchun ma'lumotlar to'plami ( *x-* o'qi ).

91

|  |
| --- |
| **93-bet** |

• y: massiv *,* shakli (n,)

◦ *Ordinata* o'qi uchun ma'lumotlar to'plami (o'qi o'qi ).

• lar: skalar yoki massiv *,* shakli (n,) *,* ixtiyoriy

◦ Ballar shkalasi.

• c: rang

[6](https://translate.googleusercontent.com/translate_f#93)yoki rangli elementlar to'plami *,* ixtiyoriy

◦ Rang.

• marker: MarkerStyle, ixtiyoriy

◦ nuqta uslubi.

• smap: str, Colormap [7](https://translate.googleusercontent.com/translate_f#93)*,* ixtiyoriy *,* sukut bo'yicha: yo'q

◦ Ranglar xaritasi (" *4.4.1 Rangli xaritalar (rang xaritalari)* " ga qarang).

• namuna: Normalize [8](https://translate.googleusercontent.com/translate_f#93)*,* ixtiyoriy *,* sukut bo'yicha: yo'q

◦ Ma'lumotlarni normalizatsiya qilish.

• alfa: skalar *,* ixtiyoriy *,* sukut bo'yicha:

Yo'q

◦ oshkoralik.

• kenglik: skalar yoki massiv *,* ixtiyoriy *,* qiymat

sukut bo'yicha: yo'q

The Belgilangan chegaraning kengligi.

• edgecolors: {'face', 'none', None}, rang

6 yoki rang to'plami

elementlar *,* ixtiyoriy *.*

Order Chegara rangi.

6 Rangni sozlashning mavjud usullaridan biri (" *2.3.2 Line color* " bo'limiga qarang )

7 [https://matplotlib.org/api/\_as\_gen/matplotlib.colors.Colormap.html](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://matplotlib.org/api/_as_gen/matplotlib.colors.Colormap.html)

8 [https://matplotlib.org/api/\_as\_gen/matplotlib.colors.Normalize.html](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://matplotlib.org/api/_as_gen/matplotlib.colors.Normalize.html)

92

|  |
| --- |
| **Sahifa 94** |

Scatter () funktsiyasi parametrlari bilan ishlashga misol:

x = np.arange ( 0 , 10.5 , 0.5 )

y = np.cos (x)

plt.scatter (x, y, b = 80 , S = **«r»** , marker = **'D'** , linewidths = 2 , edgecolors = **'g'** )

**4.17-rasm - Tarqatish diagrammasi (2-misol)**

Rang va hajm bilan ishlashni namoyish qiluvchi misol:

matplotlib.colorsni mcolors **sifatida import**qiling

bc = mcolors.BASE\_COLORS

x = np.arange ( 0 , 10.5 , 0.25 )

y = np.cos (x)

num\_set = np.random.randint ( 1 , len (mcolors.BASE\_COLORS), len (x))

registri = num\_set \* 35

color = [ list (bc.keys ()) [i] **for** i **for** num\_set]

plt.scatter (x, y, s = o'lchamlar, alfa = 0.4 , c = ranglar, chiziq kengliklari = 2 ,

edgecolors = **'yuz'** )

plt.plot (x, y, **'g--'** , alfa = 0.4 )

93

|  |
| --- |
| **Sahifa 95** |

**4.18-rasm - O'zgartirilgan tarqalish chizmasi (3-misol)**

94

|  |
| --- |
| **Sahifa 96** |

**4.3 Ustunlar va piroglar**

**4.3.1 Ustunlar jadvallari**

Ustunli, toifali ma'lumotlarni tasavvur qilish uchun juda mos keladi.

grafikalar. Ularni qurish uchun quyidagi funktsiyalar qo'llaniladi:

• bar () - vertikal chiziqli jadval;

• barh () - gorizontal chiziqli jadval.

Oddiy diagramma tuzamiz:

np.random.seed ( 123 )

Guruhlar = [ **f'P {** i **} " uchun** I **-yilda** qator ( 7 )]

hisoblaydi = np.random.randint ( 3 , 10 , len (guruhlar))

plt.bar (guruhlar, hisoblar)

**4.19-rasm - Vertikal chiziqli jadval**

95

|  |
| --- |
| **Sahifa 97** |

Agar biz bar () ni barh () bilan almashtirsak, biz gorizontal diagrammani olamiz:

plt.barh (guruhlar, hisoblar)

**Shakl 4.20 - Gorizontal chiziqli jadval**

Bar () funktsiyasi parametrlarini batafsil ko'rib chiqamiz:

*Asosiy parametrlar:*

• x: qator

◦ *x -* ustunlar koordinatalari.

• balandlik: skalar yoki massiv

◦ Ustun balandliklari.

• kenglik: skalar, massiv yoki ixtiyoriy

◦ Ustun kengligi.

Pastki: skalar, massiv yoki ixtiyoriy

◦ *y -* bazaning koordinatasi.

• hizalamak: {'center', 'edge'} *,* ixtiyoriy; standart qiymat *:*

"markaz"

◦ *x* koordinatasiga tekislang .

*Qo'shimcha variantlar:*

• rang: rang

[9](https://translate.googleusercontent.com/translate_f#97), rang to'plami yoki ixtiyoriy

Chart Diagramma ustunlarining rangi.

9 Rangni sozlashning mavjud usullaridan biri (" *2.3.2 Line color* " bo'limiga qarang )

96

|  |
| --- |
| **Sahifa 98** |

• edgecolor: rang

[10](https://translate.googleusercontent.com/translate_f#98), rang elementlari to'plami yoki ixtiyoriy

◦ Ustun chegarasi rangi.

• kenglik: skalar, massiv yoki ixtiyoriy

Order Chegara kengligi.

• tick\_label: str, array yoki ixtiyoriy

◦ ustun uchun yorliqlar.

• xerr *,* yerr: skalar, o'lchamdagi massiv (N,) *,*

shakli (2, N) yoki ixtiyoriy

◦ Grafik uchun xato miqdori. Aniq qiymat

yuqoriga qo'shilgan / olib tashlangan (o'ng - uchun

gorizontal grafik) chegara. Qabul qilishi mumkin

quyidagi qiymatlar:

▪ skalar: barcha chiziqlar uchun nosimmetrik +/-;

▪ shakli (N,): har bir satr uchun nosimmetrik +/-;

▪ shakli (2, N): odatiy - va har bir satr uchun +. Birinchi

satr pastki xato qiymatlarini o'z ichiga oladi, ikkinchi satr esa

yuqori;

▪ Yo'q: xato qiymatlarini ko'rsatmang. Ushbu qiymat

sukut bo'yicha ishlatiladi.

• rang: rang

10 , rang elementlari to'plami yoki ixtiyoriy; qiymat

standart: "qora"

The Xato chizig'ining rangi.

• log: bool *,* ixtiyoriy; standart qiymat: noto'g'ri

◦ *Y* o'qi uchun logaritmik o'lchovni *yoqadi.*

• yo'nalish: {'vertikal', 'gorizontal'} *,* ixtiyoriy

◦ Yo'nalish: vertikal yoki gorizontal.

10 Rangni sozlashning mavjud usullaridan biri (" *2.3.2 Chiziq rangi* " bo'limiga qarang )

97

|  |
| --- |
| **99-bet** |

Bar () parametrlari bilan ishlashni ko'rsatuvchi misol:

matplotlib.colorsni mcolors **sifatida import**qiling

bc = mcolors.BASE\_COLORS

np.random.seed ( 123 )

Guruhlar = [ **f'P {** i **} " uchun** I **-yilda** qator ( 7 )]

hisoblaydi = np.random.randint ( 0 , len (bc), len (guruhlar))

kenglik = hisoblash \* 0,1

ranglar = [[ **'r'** , **'b'** , **'g'** ] [ int (np.random.randint ( 0 , 3 , 1 ))] **uchun** \_ **in**

hisoblaydi]

plt.bar (guruhlar, hisoblar, kenglik = kenglik, alfa = 0,6 , pastki = 2 , rang = ranglar,

edgecolor = **'k'** , chiziq kengligi = 2 )

**4.21-rasm - O'zgartirilgan shtrixli jadval**

**4.3.1.1 guruhli jadvallar**

Ma'lum bir tarzda tayyorlangan ma'lumotlardan foydalanishingiz mumkin

guruh diagrammalarini yaratish:

cat\_par = [ **f'P {** i **} " uchun** I **-yilda** qator ( 5 )]

g1 = [ 10 , 21 , 34 , 12 , 27 ]

g2 = [ 17 , 15 , 25 , 21 , 26 ]

kengligi = 0,3

x = np.arange ( len (cat\_par))

98

|  |
| --- |
| **100-bet** |

fig, ax = plt.subplots ()

rects1 = ax.bar (x - width / 2 , g1, width, label = **'g1'** )

rects2 = ax.bar (x + width / 2 , g2, width, label = **'g2'** )

ax.set\_title ( **'Guruh** diagrammasi **misoli'** )

ax.set\_xticks (x)

ax.set\_xticklabels (cat\_par)

ax.legend ()

**4.22-rasm - Guruhlar jadvalining diagrammasi**

**4.3.1.2 *Xatolar paneli* elementi bilan *diagramma***

*Xatolar* elementi har biri uchun xato qiymatini belgilashga imkon beradi

grafik elementi. Buning uchun xerr, yerr va parametrlari

ecolor *,* dastlabki ikkitasi xatoning kattaligini, ikkinchisi rangni belgilaydi:

np.random.seed ( 123 )

rnd = np.random.randint

cat\_par = [ **f'P {** i **} " uchun** I **-yilda** qator ( 5 )]

g1 = [ 10 , 21 , 34 , 12 , 27 ]

xato = np.array ([[rnd ( 2 , 7 ), rnd ( 2 , 7 )] **uchun** \_ **yilda** qator ( Len . T (cat\_par))])

99

|  |
| --- |
| **101-sahifa** |

fig, axs = plt.subplots ( 1 , 2 , figsize = ( 10 , 5 ))

axs [ 0 ] .bar (cat\_par, g1, yerr = 5 , ecolor = **'r'** , alfa = 0.5 , edgecolor = **'b'** ,

chiziq kengligi = 2 )

axs [ 1 ] .bar (cat\_par, g1, yerr = error, ecolor = **'r'** , alfa = 0.5 , edgecolor = **'b'** ,

chiziq kengligi = 2 )

**4.23-rasm - *Xatolar paneli* elementi bilan ustunlar jadvali**

**4.3.2 Dairesel jadvallar**

**4.3.2.1 Klassik piroglar jadvali**

Dairesel jadvallar - bu mutanosiblikni ko'rsatishning ingl

to'plamdagi komponentlar. Ular hisobot berish uchun ideal,

taqdimotlar va boshqalar. *Matplotlib-da* pirog diagrammalarini *tuzish uchun*

pie () funktsiyasi ishlatiladi.

Misol diagrammasi:

vals = [ 24 , 17 , 53 , 21 , 35 ]

teglar = [ **'Ford'** , **'Toyota'** , **'BMW'** , **'AUDI'** , **'Jaguar'** ]

fig, ax = plt.subplots ()

ax.pie (vallar, teglar = yorliqlar)

ax.axis ( **'teng'** )

yuz

|  |
| --- |
| **Sahifa 102** |

**4.24-rasm - pirog diagrammasi**

Pie () funktsiyasi parametrlarini ko'rib chiqing:

• x *:* qator

Portion qism o'lchamlari bilan massiv.

• portlash: qator, ixtiyoriy; standart qiymat: Yo'q

◦ Agar parametr Hech kimga teng bo'lmasa, u holda zarbalarning qismi

uzatilgan qiymatda sanab o'tilgan

ma'lum bir masofa uchun jadvallar. Bunday diagrammaning namunasi:

**4.25-rasm - ajratilgan sektori bo'lgan doiraviy diagramma**

• yorliqlar *:* ro'yxat *,* ixtiyoriy; sukut bo'yicha: yo'q

Text Matn yorliqlarini urib qo'ying.

101

|  |
| --- |
| **103-sahifa** |

• ranglar: rang elementlari massivi

[o'n bir](https://translate.googleusercontent.com/translate_f#103), ixtiyoriy; qiymati

sukut bo'yicha: yo'q

Colors Ranglarni urish.

• autopct: str, function, ixtiyoriy; standart qiymat: Yo'q

The Beat ichidagi matn yorlig'i formati, matn sonli

ma'lum bir ulush bilan bog'liq indikatorning qiymati.

• pctdistance: float *,* ixtiyoriy; standart qiymat: 0.6

◦ Har bir mag'lubiyat markazi va matn boshi orasidagi masofa

autopct parametri tomonidan ko'rsatilgan yorliq.

• shadow *:* bool *,* ixtiyoriy, standart qiymat: False

◦ Diagramma uchun soyani ko'rsating.

• labeldistance: float *,* Yo'q, ixtiyoriy; standart qiymat: 1.1

Text Matn yorliqlari ko'rsatiladigan masofa

ulushlar. Agar parametr Yo'q bo'lsa, unda yorliqlar bo'lmaydi

ko'rsatiladi.

• startangle *:* suzuvchi *,* ixtiyoriy; sukut bo'yicha: yo'q

The Diagrammani soat sohasi farqli ravishda aylantirish uchun burchak

*x* o'qi atrofida o'qlar .

• radius *:* suzuvchi *,* ixtiyoriy; sukut bo'yicha: yo'q

◦ Diagramma radiusi miqdori.

• qarshi soat: bool *,* ixtiyoriy; standart qiymat *:* To'g'ri

◦ Aylanish yo'nalishi: soat yo'nalishi bo'yicha yoki teskari yo'nalishda.

• wedgeprops *:* dict *,* ixtiyoriy; sukut bo'yicha: yo'q

◦ Beatlarning ko'rinishini aniqlaydigan parametrlar lug'ati (qarang)

matplotlib.patches.Wedge).

• textprops *:* dict *,* ixtiyoriy; sukut bo'yicha: yo'q

11 Rangni sozlashning mavjud usullaridan biri (" *2.3.2 Chiziq rangi* " bo'limiga qarang )

102

|  |
| --- |
| **104-bet** |

◦ Matn ko'rinishini aniqlaydigan parametrlar lug'ati

yorliqlar (matplotlib.text.Text sinfiga qarang).

• markaz: qiymatlar ro'yxati suzadi, ixtiyoriy; standart qiymat:

(0, 0)

The Grafik markazi.

• ramka: bool *,* ixtiyoriy; standart qiymat: noto'g'ri

◦ Agar parametr rost bo'lsa, u holda bo'ladi

ramka ko'rsatiladi.

• rotatelabellar: bool *,* ixtiyoriy; standart qiymat: noto'g'ri

◦ Agar parametr rost bo'lsa, matn yorliqlari aylantiriladi

berilgan burchak ostida.

Pie () funktsiyasi parametrlari bilan ishlashni namoyish qiluvchi misol:

vals = [ 24 , 17 , 53 , 21 , 35 ]

teglar = [ **'Ford'** , **'Toyota'** , **'BMW'** , **'AUDI'** , **'Jaguar'** ]

portlash = ( 0,1 , 0 , 0,15 , 0 , 0 )

fig, ax = plt.subplots ()

ax.pie (vals, teglar = teglar, autopct = **'% 1.1f %%'** , shadow = **True** ,

explode = explode, wedgeprops = { **'lw'** : 1 , **'ls'** : **'-'** , **'edgecolor'** : **'k'** },

rotatelabels = **rost** )

ax.axis ( **'teng'** )

**4.26-rasm - O'zgartirilgan piroglar jadvali**

103

|  |
| --- |
| **Sahifa 105** |

**4.3.2.2 Ichki pirog diagrammasi**

Ichki pirog diagrammasi ikkita tarkibiy qismdan iborat:

uning ichki qismi ma'lumotlarning batafsil taqdimoti,

tashqi qismi esa berilgan maydonlar uchun jami. Har bir hudud

raqamli qiymatlarning ro'yxati bo'lib, ular birgalikda hosil bo'ladi

umumiy ma'lumotlar to'plami.

Misol:

fig, ax = plt.subplots ()

ofset = 0,4

data = np.array ([[ 5 , 10 , 7 ], [ 8 , 15 , 5 ], [ 11 , 9 , 7 ]])

cmap = plt.get\_cmap ( **'tab20b'** )

b\_colors = smap (np.array ([ 0 , 8 , 12 ]))

sm\_colors = smap (np.array ([ 1 , 2 , 3 , 9 , 10 , 11 , 13 , 14 , 15 ]))

ax.pie (data.sum ( o'q = 1 ), radius = 1 , ranglar = b\_colors,

wedgeprops = dict ( width = ofset, edgecolor = **'w'** ))

ax.pie (data.flatten (), radius = 1- ofset, ranglar = sm\_colors,

wedgeprops = dict ( width = ofset, edgecolor = **'w'** ))

**4.27-rasm - Ichki tortilgan diagramma**

104

|  |
| --- |
| **Sahifa 106** |

**4.3.2.3 Teshikli pirog diagrammasi**

Teshikli dairesel diagramma qurish uchun sizga kerak

aktsiyalar ko'rinishi uchun javobgar bo'lgan wedgeprops parametri o'rnatildi

dik qiymat (kenglik = 0,5):

vals = [ 24 , 17 , 53 , 21 , 35 ]

teglar = [ **'Ford'** , **'Toyota'** , **'BMV'** , **'AUDI'** , **'Jaguar'** ]

fig, ax = plt.subplots ()

ax.pie (vals, teglar = teglar, wedgeprops = dict ( kenglik = 0,5 ))

**4.28-rasm - Teshikli pirog diagrammasi**

105

|  |
| --- |
| **107-bet** |

**4.4 Rangli panjara**

Rang panjarasi bu rang bilan to'ldirilgan maydon

rang xaritasi va elementlarning raqamli qiymatlari bilan belgilanadi

o'tgan ikki o'lchovli qator.

**4.4.1 Rangli karta ( *kolormaplar* )**

Rangli xarita - bu bo'lishi mumkin bo'lgan tayyorlangan ranglar to'plami

ma'lumotlar to'plamlarini tasavvur qilish uchun foydalaning. Batafsil

[rasmiy](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://matplotlib.org/3.1.0/tutorials/colors/colormaps.html) kartada topishingiz mumkin bo'lgan rangli kartalar bo'yicha qo'llanma

[Onlayn Matplotlib](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://matplotlib.org/3.1.0/tutorials/colors/colormaps.html)[*12*](https://translate.googleusercontent.com/translate_f#107)*.*Agar mavjud bo'lsa, bunday xaritani o'zingiz yaratishingiz mumkin

mavjud emas. 4.29-rasmda misollar keltirilgan

*Matplotlib kutubxonasidagi* ba'zi rangli xaritalar .

**4.29-rasm - Rangli xaritalar**

12 [https://matplotlib.org/3.1.0/tutorials/colors/colormaps.html](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://matplotlib.org/3.1.0/tutorials/colors/colormaps.html)

106

|  |
| --- |
| **Sahifa 108** |

**4.4.2 Rangli tarmoqni yaratish**

Rangli panjara yaratish uchun siz funktsiyadan foydalanishingiz mumkin

imshow () yoki pcolormesh ().

**imshow ()**

Imshow () funktsiyasining asosiy maqsadi *2D tasvirlashdir*

rasters. Bu rasmlar, ikki o'lchovli ma'lumotlar massivlari, matritsalar bo'lishi mumkin

va h.k.

Internetdan rasm yuklab oladigan oddiy dastur yozamiz

berilgan *urlda* va uni kutubxona yordamida namoyish etadi

*Matplotlib* :

**dan**batareya **import**Image

**import**so'rovlari

**dan**io **import**BytesIO

response = Request.get ( **'https://matplotlib.org/\_static/logo2.png'** )

img = Image.open (BytesIO (response.content))

plt.imshow (img)

Uning ishi natijasi *Matplotlib* logotipi tasviri bo'ladi .

**4.30-rasm - *Matplotlib* logotipi *tasviri***

107

|  |
| --- |
| **Sahifa 109** |

Keling, 2 o'lchovli ma'lumotlar to'plamini yaratamiz va uni ko'rsatamiz

imshow ():

np.random.seed ( 19680801 )

ma'lumotlar = np.random.randn ( 25 , 25 )

plt.imshow (ma'lumotlar)

**4.31-rasm - bilan ikki o'lchovli ma'lumotlar to'plamini vizualizatsiya qilish**

**imshow () dan foydalanish**

Imshow () funktsiyasining ba'zi parametrlarini ko'rib chiqamiz:

• X: massiv yoki PIL tasviri

◦ Quyidagi massiv o'lchamlari qo'llab-quvvatlanadi:

▪ (M, N): skalar ma'lumotlari bo'lgan ikki o'lchovli massiv.

▪ (M, N, 3): *RGB* qiymatlari bo'lgan massiv (0-1 suzuvchi yoki 0-255

int).

▪ (M, N, 4): *RGBA* qiymatlari bo'lgan massiv (0-1 float yoki 0-255

int).

• smap: str yoki Colormap *,* ixtiyoriy

◦ tasvir uchun rangli xarita (qarang: " *4.4.1 rangli xaritalar*

*(kolormaplar)* ")

• norma: Normallashtirish *,* ixtiyoriy

◦ Normalizatsiya - skalar ma'lumotlarini diapazonga aylantirish

[0,1] rangli xaritani qoplashdan oldin. Ushbu parametr

*RGB (A)* ma'lumotlari uchun e'tiborsiz qoldirildi .

108

|  |
| --- |
| **Sahifa 110** |

• aspekt: ​​{'teng', 'avtomatik'} yoki suzuvchi *,* ixtiyoriy

Equal 'teng': tomonlarning nisbati 1 ga teng;

Auto 'avtomatik': nisbat o'zgarmaydi.

• interpolatsiya: str *,* ixtiyoriy

◦ Interpolatsiya algoritmi. Quyidagi qiymatlar mavjud:

'yo'q', 'eng yaqin', 'bilinear', 'bikubic', 'spline16',

'spline36', 'hanning', 'hamming', 'hermite', 'kaiser',

"quadric", "catrom", "gaussian", "bessel", "mitchell",

"sinc", "lanczos".

• alfa: raqamli qiymat *,* ixtiyoriy

◦ oshkoralik. Uni 0 dan 1 gacha oraliqda o'rnatish mumkin. Parametr

*RGBA* uchun e'tiborsiz qoldirildi .

• vmin, vmax: raqamli qiymat, ixtiyoriy

◦ vmin va vmax raqamli qiymatlari (agar norma ko'rsatilmagan bo'lsa

aniq) qamrab olinadigan ma'lumotlar doirasini aniqlang

rangli karta. Odatiy bo'lib, rang xaritasi butunlay qamrab oladi

ko'rsatilgan ma'lumotlarning qiymatlari oralig'i. Agar ishlatilgan bo'lsa

parametr normasi, keyin vmin va vmax e'tiborga olinmaydi.

• kelib chiqishi: {'yuqori', 'pastki'} *,* ixtiyoriy

Of Kelib chiqish joyi ([0,0] punktlar): 'yuqori' *-*

yuqori chap, 'pastki' *-* koordinataning pastki chap burchagi

samolyot.

• daraja: (chap, o'ng, pastki, yuqori) *,* ixtiyoriy

◦ Tasvirni *x, y* o'qlari bo'ylab o'lchamlarini o'zgartiring .

• filterrad: float> 0 *,* ixtiyoriy; standart qiymat: 4.0

◦ Filtrlar uchun filtr radiusi parametri

masalan, "sinc" *,* "lanczos" yoki "blackman" dan foydalaning.

109

|  |
| --- |
| **Sahifa 111** |

Yuqoridagi ro'yxatdagi parametrlardan foydalangan holda misol:

anjir, axs = plt.subplots ( 1 , 2 , figsize = ( 10 , 3 ), constrained\_layout = **True** )

p1 = axs [ 0 ] .imshow (ma'lumotlar, smap = **'qish'** , aspekt = **'teng'** , vmin = - 1 , vmax = 1 ,

kelib chiqishi = **"past"** )

fig.colorbar (p1, ax = eksa [ 0 ])

p2 = eksa [ 1 ] .imshow (ma'lumotlar, smap = **'plazma'** , aspekt = **'teng'** ,

interpolatsiya = **'gauss'** , kelib chiqishi = **'past'** , darajasi = ( 0 , 30 , 0 , 30 ))

fig.colorbar (p2, ax = axs [ 1 ])

**4.32-rasm - Ikki o'lchovli ma'lumotlar to'plamini tasavvur qilish imkoniyatlari**

**pcolormesh ()**

Biz *2 o'lchovli* ma'lumotlar to'plamini tasavvur qilish uchun keyingi funktsiya

pcolormesh () ni ko'rib chiqing. U erda yana bir biri *Matplotlib kutubxona*

o'xshash funktsiyaga ega vosita - pcolor (), aksincha,

biz ko'rib chiqayotgan pcolormesh () tezroq va yaxshiroqdir

aksariyat hollarda variant. Pcolormesh () funktsiyasi shunga o'xshash

imshow () da uning imkoniyatlari, ammo farqlar mavjud.

Pcolormesh () funktsiyasining parametrlarini ko'rib chiqing:

• C: qator

◦ *2D* skalar qadriyatlar majmuasini.

• smap: str yoki Colormap *,* ixtiyoriy

Cm imshow () dagi smap-ga qarang *.*

110

|  |
| --- |
| **112-bet** |

• norma: Normallashtirish *,* ixtiyoriy

Norm imshow () dagi normani ko'ring *.*

• vmin *,* vmax: raqamli qiymat *,* ixtiyoriy; qiymati

sukut bo'yicha: yo'q

V imshow () da vmin *,* vmax-ga qarang *.*

• edgecolors: {'none', None, 'face', color[13](https://translate.googleusercontent.com/translate_f#112), rang qatori

elementlar} *,* ixtiyoriy; standart qiymat: 'yo'q'

Order Chegara rangi. Quyidagi variantlar mumkin:

▪ 'yo'q' yoki '': chegara ekrani yo'q;

▪ yo'q: qora;

▪ "yuz": katak rangidan foydalaniladi;

▪ Rangni o'rnatishning mavjud usullaridan biri (" *2.3.2* " bo'limiga qarang

*Chiziq rangi* ").

• alfa: raqamli qiymat *,* ixtiyoriy; standart qiymat *:*

Yo'q

Im alfa-ni imshow-da ko'ring ().

• soya: {'tekis', 'gouraud'} *,* ixtiyoriy

◦ To'ldirish uslubi. Mavjud qiymatlar:

▪ "tekis": har bir kvadrat uchun qattiq rang;

▪ "gouraud": har bir kvadrat uchun usul qo'llaniladi

soyali *gurus* .

• snap: bool *,* ixtiyoriy; standart *:* noto'g'ri

◦ Piksel chegaralarini kesib oling.

13 Rangni sozlashning mavjud usullaridan biri (" *2.3.2 Line color* " bo'limiga qarang )

111

|  |
| --- |
| **113-bet** |

Pcolormesh () funktsiyasidan foydalanish misoli:

np.random.seed ( 123 )

ma'lumotlar = np.random.rand ( 5 , 7 )

plt.pcolormesh (ma'lumotlar, smap = **'plazma'** , edgecolors = **'k'** , shading = **'flat'** )

**4.33-rasm - bilan ikki o'lchovli ma'lumotlar to'plamini vizualizatsiya qilish**

**pcolormesh () yordamida**

112

|  |
| --- |
| **Sahifa 114** |

**5-bob. *3D-* grafikalarni qurish. Bilan ishlash**

***mplot3d Toolkit***

Shu paytgacha biz qurgan barcha grafikalar ikki o'lchovli edi va

*Matplotlib shuningdek 3D* grafika *chizish* imkoniyatini beradi . Biz import qilamiz

*3D* bilan ishlash uchun zarur bo'lgan modullar :

plt **sifatida import**matplotlib.pyplot

**dan**mpl\_toolkits.mplot3d **import**Axes3D

Keling, *3D* chizish uchun ba'zi vositalarni ko'rib chiqamiz.

**5.1 Chiziqli diagramma**

Chiziqli grafani chizish uchun dan () funktsiyasidan foydalaning

Axes3D:

Axes3D.plot (self, xs, ys, \* args, zdir = **'z'** , \*\* kwargs)

*Axes3D.plot* funktsiyasi *parametrlari* :

• xs, ys: *1D-* massivlar

◦ nuqtalarning *x* va *y* koordinatalari .

• *zs* : raqam yoki *1D-* qator

◦ *z* koordinatalari. Agar skaler qiymat berilsa, u holda bo'ladi

grafadagi barcha nuqtalarga tayinlangan.

• zdir *:* {'x', 'y', 'z'}; sukut bo'yicha: 'z'

◦ *z* yo'nalishi sifatida qabul qilinadigan o'q .

• \*\* kvarg

Used ishlatilganiga o'xshash qo'shimcha dalillar

2D grafikalarni chizish uchun plot () funktsiyasida.

113

|  |
| --- |
| **Sahifa 115** |

x = np.linspace (-np.pi, np.pi, 50 )

y = x

z = np.cos (x)

fig = plt.figure ()

ax = fig.add\_subplot ( 111 , proyeksiya = **'3d'** )

ax.plot (x, y, z, label = **'parametrik egri'** )

**5.1-rasm - Axes3D.plot () funktsiyasini namoyish etish**

**5.2 nuqta uchastkasi (tarqalish chizmasi)**

Scatter () funktsiyasi tarqalish chizmasini qurish uchun ishlatiladi

Axes3D dan:

Axes3D.scatter (o'zini, XS, HydrizBot, ZS = 0 , zdir = **'z'** , b = 20 , S = **hech** ,

deepshade = **rost** , \* args, \*\* kwargs)

Axes3D.scatter () funktsiya parametrlari:

• xs *,* ys: *1D* massivi

◦ nuqtalarning *x* va *y* koordinatalari .

• zs : raqam yoki *1D* -array, ixtiyoriy; standart qiymat: 0

◦ *Z-* punktlari koordinatalarini . Agar skaler qiymat berilgan bo'lsa,

keyin u grafikning barcha nuqtalariga tayinlanadi.

114

|  |
| --- |
| **Sahifa 116** |

• zdir *:* {'x', 'y', 'z', '-x', '-y', '-z'} *,* ixtiyoriy; qiymat

sukut bo'yicha 'z'

◦ *z* yo'nalishi sifatida qabul qilinadigan o'q .

• lar: raqam yoki qator, ixtiyoriy; standart 20

◦ Marker hajmi.

• c: rang [14](https://translate.googleusercontent.com/translate_f#116) , raqamlar massivi, rang elementlari massivi, ixtiyoriy

◦ marker rangi. Mumkin bo'lgan qiymatlar:

▪ barcha markerlar uchun rangning satr qiymati;

▪ qator ranglari qiymatlari to'plami;

▪ orqali ranglarda ko'rsatilishi mumkin bo'lgan raqamlar qatori

smap va norma funktsiyalari;

▪ *2D* - elementlari *RGB* yoki *RGBA* bo'lgan massiv ;

• chuqurlik *:* bool *,* ixtiyoriy

Depth Chuqurlik effekti uchun soyali belgilar.

• \*\* kvarg

Used ishlatilganiga o'xshash qo'shimcha dalillar

2-o'lchovli grafikani chizish uchun scatter () funktsiyasida.

np.random.seed ( 123 )

x = np.random.randint (- 5 , 5 , 40 )

y = np.random.randint ( 0 , 10 , 40 )

z = np.random.randint (- 5 , 5 , 40 )

s = np.random.randint ( 10 , 100 , 20 )

fig = plt.figure ()

ax = fig.add\_subplot ( 111 , proyeksiya = **'3d'** )

ax.scatter (x, y, z, s = s)

14 Rangni sozlashning mavjud usullaridan biri (" *2.3.2 Chiziq rangi* " bo'limiga qarang )

115

|  |
| --- |
| **Sahifa 117** |

**5.2-rasm - Axes3D.scatter () funktsiyasini namoyish etish**

**5.3 Kadr yuzasi**

Simli ramka yuzasini qurish uchun funktsiyadan foydalaning

Axes3D dan plot\_wireframe ():

Axes3D.plot\_wireframe (o'zini, X, Y, Z, \* arglar, \*\* kvarglar)

Axes3D.plot\_wireframe () funktsiya parametrlari:

• X, Y, Z: *2D* massivlari

Face Yuzaki qurilish ma'lumotlari.

• rcount *,* ccount: int, standart qiymat: 50

Will Bu ramka elementlarining maksimal soni

har bir yo'nalishda ishlatiladi.

• rstride *,* cstride: int

Which bilan qadamning o'lchamini aniqlaydigan parametrlar

o'tgan qatorlardan qator / ustun elementlarini oling.

Rstride, cstride va rcount, ccount parametrlari

o'zaro eksklyuziv.

• \*\* kvarg

◦ Parametr bo'lgan qo'shimcha dalillar

Line3DCollection sinfining konstruktori.

116

|  |
| --- |
| **Sahifa 118** |

u, v = np.mgrid [ 0 : 2 \* np.pi: 20j , 0 : np.pi: 10j ]

x = np.cos (u) \* np.sin (v)

y = np.sin (u) \* np.sin (v)

z = np.cos (v)

fig = plt.figure ()

ax = fig.add\_subplot ( 111 , proyeksiya = **'3d'** )

ax.plot\_wireframe (x, y, z)

ax.legend ()

**5.3-rasm - funktsiyani namoyish etish**

**Axes3D.plot\_wireframe ()**

**5.4 Yuzaki**

Sirtni chizish uchun, dan plot\_surface () funktsiyasidan foydalaning

Axes3D:

Axes3D.plot\_surface (o'zini, X, Y, Z, \* args, norm = **Yo'q** , vmin = **Yo'q** ,

vmax = **Hech** , lightsource = **hech** , \*\* kwargs)

Axes3D.plot\_surface () funktsiya parametrlari:

• X, Y, Z: *2D* massivlari

Face Yuzaki qurilish ma'lumotlari.

• rcount *,* ccount: int

117

|  |
| --- |
| **119-bet** |

◦ " *5.3 Wireframe Surface* " dan rcount, ccount-ga qarang .

• rstride *,* cstride *:* int

◦ " *5.3 Wireframe Surface* " dan rstride, cstride-ga qarang .

• rang *:* rangni sozlashning mavjud usullaridan biri (bo'limga qarang

" *2.3.2 Chiziq rangi* ").

Surface Sirt xususiyatlari uchun rang.

• smap: str yoki Colormap *,* ixtiyoriy

Surface Sirt uchun rangli xarita (qarang " *4.4.1 Rangli xaritalar"*

*(kolormaplar)* ")

• yuz ranglari: rang elementlari massivi

Each Har bir sirt xususiyati uchun alohida rang.

• norma: Normallashtirish

Col Kolormap uchun normalizatsiya.

• vmin *,* vmax *:* suzuvchi

◦ normallashtirish chegaralari.

• soya *:* bool; *s* Standart qiymatlar: To'g'ri

Fac Ranglar uchun soyadan foydalanish.

• yorug'lik manbasi: LightSource

◦ LightSource sinfining ob'ekti yorug'lik manbasini belgilaydi,

soya = To'g'ri bo'lsagina ishlatiladi.

• \*\* kvarg

◦ Parametr bo'lgan qo'shimcha dalillar

Poly3DCollection sinfining konstruktori.

u, v = np.mgrid [ 0 : 2 \* np.pi: 20j , 0 : np.pi: 10j ]

x = np.cos (u) \* np.sin (v)

y = np.sin (u) \* np.sin (v)

z = np.cos (v)

fig = plt.figure ()

118

|  |
| --- |
| **120-bet** |

ax = fig.add\_subplot ( 111 , proyeksiya = **'3d'** )

ax.plot\_surface (x, y, z, cmap = **'** xulosa **'** )

ax.legend ()

**5.4-rasm - funktsiyani namoyish etish**

**Axes3D.plot\_surface ()**

119

|  |
| --- |
| **121-bet** |

**II qism. Seaborn kutubxonasi**

**Kirish**

*Seaborn -* bu vizualizatsiya vazifalarini hal qilish uchun kutubxona

statistika sohasida ishlashga yo'naltirish. Base *seaborn* emas

birinchisida batafsil berilgan *Matplotlib kutubxonasi*

kitob qismlari.

*Seaborn* quyidagilarni ta'minlaydi:

• ma'lumotlar to'plamidagi munosabatlarni o'rganish uchun qulay *API* ;

• chiziqli modellarni avtomatik hisoblash va ko'rsatish

regressiya;

• ma'lumotlar ichki tuzilishini o'rganish vositalari;

• har xil jadvallarning joylashishini boshqarish vositalari

bitta maydonda turlar;

• Diagramma mavzulari.

Ma'lumotlarni vizualizatsiya qilish vositalari *seaborn* tomonidan taqdim etilgan

besh guruhga bo'lish mumkin:

• ma'lumotlardagi munosabatlarni vizualizatsiya qilish;

• toifadagi ma'lumotlarni vizualizatsiya qilish;

• tarqatishlarni vizualizatsiya qilish;

• chiziqli regressiyani vizualizatsiya qilish;

• matritsali ma'lumotlar to'plamlarini vizualizatsiya qilish.

Design *API* - ma'lumotlar vizuallashtirish uchun mo'ljallangan *dengiz funktsiyalari*

turli funktsiyalarning argumentlari mavjud bo'lgan tarzda

bir xil maqsad, bir xil nomlarga ega bo'lish. Shuning uchun, o'qiyotganda

120

|  |
| --- |
| **Sahifa 122** |

har xil vizualizatsiya vositalari biz oldindan beramiz

bu savolga qaytmaslik uchun umumiy dalillarning tavsifi

funktsiyalarni o'zlari ko'rib chiqishda. Ushbu bo'limlarni birinchi o'qiyotganda

funksionallikni o'rganish jarayonida ularni o'tkazib yuborishingiz va ularga murojaat qilishingiz mumkin

u yoki bu asbob.

121 2

|  |
| --- |
| **123-bet** |

**6-bob. Tezkor boshlash**

**6.1 O'rnatish**

**6.1.1 *dengiz qirg'og'ini* o'rnatish *imkoniyatlari***

*Seaborn* kutubxona qismi hisoblanadi *Anaconda* to'plami Agar uni bor, agar

o'rnatilgan, ushbu bo'limni o'tkazib yuborishingiz mumkin. Qo'llanma

*Anaconda* o'rnatilishi yoqilgan[devpractice.ru](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://devpractice.ru/python-lesson-1-install/)[15](https://translate.googleusercontent.com/translate_f#123)...

**6.1.2 *Dengiz dengizini quvur* va *konda* orqali o'rnatish**

Kutubxonani o'rnatish uchun siz to'plamdan foydalanishingiz mumkin

*pip* menejeri :

**quvurlarni o'rnatish dengiz**

yoki *conda* :

**conda install seaborn**

Yuqoridagi ikkala variantda ham siz eng so'nggi barqarorni olasiz

ozod qilish. Agar sizga eng so'nggi, hozirgi vaqtda versiyasi qiziq bo'lsa

( *Ishlab chiqish versiyasi* ), siz o'rnatishingiz mumkin *seaborn* to'g'ridan to'g'ri *GitHub* :

**pip install git + https: //github.com/mwaskom/seaborn.git#egg=seaborn**

Uning ish uchun *seaborn* quyidagi modullardan talab:

• *Python* 3.6+

• *uyqusirab*

• *jirkanch*

• *pandalar*

• *matplotlib*

• *statsmodels*

15 [https://devpractice.ru/python-lesson-1-install/](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://devpractice.ru/python-lesson-1-install/)

122

|  |
| --- |
| **Sahifa 124** |

Agar ulardan ba'zilari o'rnatilmagan bo'lsa, ular yuklanadi va

avtomatik ravishda tarqatiladi.

**6.1.3 O'rnatishni tekshirish**

*Dengiz sathining to'g'ri* o'rnatilganligini *tekshirish* uchun tarjimonni ishga tushiring

*Python* buyruq rejimida va quyidagi buyruqlarni kiriting:

>>> dengiz dengizidan import qilish

>>> dengiz tug'ilgan .\_\_ versiya\_\_

**"0.9.0"**

Agar natija versiya raqamini ko'rsatsa (yuqoridagi misolda u 0.9.0 bo'lsa),

u holda kutubxona o'rnatilgan va foydalanishga yaroqli deb taxmin qilish mumkin.

**6.2 Tez boshlash**

Ishni boshlashdan oldin, kerakli to'plamni import qilishingiz kerak

kutubxonalar:

pandalarni pd sifatida import qilish

np sifatida numpy- ni import qilish

import seaborn sifatida SNS

*Dengiz dengizini* olib kirishda unga " *sns"* taxallusi *beriladi* , shunday bo'ladi

ushbu kutubxona foydalanuvchilari orasida odatda qabul qilingan, sizga maslahat beramiz

ushbu parametrga rioya qiling.

Tajribalar uchun biz yuklangan ma'lumotlardan foydalanamiz

load\_dataset () funktsiyasidan foydalangan holda. Dalil sifatida u

kerakli ma'lumotlar to'plamining nomini oladi.

123

|  |
| --- |
| **Sahifa 125** |

**6.2.1 Tarqoqlik chizmasini chizish**

*Mpg* ma'lumotlar to'plamini yuklaylik :

mpg = sns.load\_dataset ( **"mpg"** )

Load\_dataset () funktsiyasi ma'lumotlar to'plamini hosil qiladi va uni qaytaradi

pandas.DataFrame ob'ekti sifatida. *Mpg* to'plami ma'lumotni o'z ichiga oladi

bir qator avtoulovlarning xususiyatlari haqida. Qo'shimcha ma'lumot

uni o'qishingiz mumkin [autompg-ma'lumotlar to'plami](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://www.kaggle.com/uciml/autompg-dataset)[16](https://translate.googleusercontent.com/translate_f#125) .

Ma'lumotlar to'plami tarkibini vizual ravishda baholash uchun biz foydalanamiz

DataFrame sinf ob'ektining head () usuli yordamida:

mpg.head ()

Tezlanish ( *tezlashuv* ) ning miqdorga bog'liqligini quraylik

ot kuchi ( *ot kuchi* ), bu erda nuqta kattaligi aniqlanadi

tsilindr soni:

sns.relplot ( x = **"ot kuchi"** , y = **"tezlanish"** , hajmi = **"silindrlar"** , ma'lumotlar = mpg)

16 [https://www.kaggle.com/uciml/autompg-dataset](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://www.kaggle.com/uciml/autompg-dataset)

124

|  |
| --- |
| **Sahifa 126** |

**6.1-rasm -** relplot () **funktsiyasini namoyish etish , variant:**

**tarqoq fitna**

**6.2.2 Chiziqli grafikni yaratish**

Chiziqli grafik funktsiyasi qanday ishlashini namoyish qilish

miqdor ma'lumotlarini o'z ichiga olgan *parvozlar* to'plamini yuklang

havo transportidan foydalangan yo'lovchilar:

reyslar = sns.load\_dataset ( **"parvozlar"** )

flight.head ()

125

|  |
| --- |
| **Sahifa 127** |

Tashilgan yo'lovchilar soniga bog'liqlikni yarataylik

( *yo'lovchilar* ) yil ( *yil* ):

sns.relplot ( x = **"yil"** , y = **"yo'lovchilar"** , turdagi = **"chiziq"** , afsona = **"to'liq"** ,

ma'lumotlar = reyslar);

**6.2-rasm - relplot () funktsiyasini namoyish etish, variant:**

**chiziqli grafik**

**6.2.3 Kategorik ma'lumotlar bilan ishlash**

*Iris* seti yordamida misol sifatida kategorik ma'lumotlar bilan ishlashni ko'rib chiqing

(Fisherning irislari):

ìrísí = sns.load\_dataset ( **"ìrísí"** )

iris.head ()

126

|  |
| --- |
| **Sahifa 128** |

Yuklangan ma'lumotlar to'plami ma'lumot olish uchun ma'lumotdir

tasniflash algoritmlari, u 150 ga yaqin ma'lumotlarni aks ettiradi

ìrísí namunalari, har bir alohida tur uchun 50 ta: Iris mo'rt

( *setosa* ), Iris *virginica* ( *virginica* ) va Iris rangli ( *versikolor* ).

Har bir misol uchun quyidagi parametrlar aniqlangan:

• tashqi perianth lobining uzunligi ( *sepal\_length* );

• tashqi perianth lobining kengligi ( *sepal\_width* );

• ichki perianth lobining uzunligi ( *petal\_length* );

• Ichki perianth lobining kengligi ( *petal\_width* ).

Ushbu to'plamni ko'rsatish uchun catplot () funktsiyasidan foydalanamiz:

sns.catplot ( x = **"turlari"** , y = **"sepal\_length"** , kind = **"to'dasi"** , ma'lumotlar = ìrísí);

**6.3-rasm - catplot () funktsiyasini namoyish etish**

127

|  |
| --- |
| **Sahifa 129** |

Diagramma ko'pincha toifali ma'lumotlar bilan ishlashda ishlatiladi

Mo'ylovli quti, u boxplot () funktsiyasi yordamida qurilgan:

sns.boxplot (x = **"turlari"** , y = **"sepal\_length"** , ma'lumotlar = ìrísí)

**6.4-rasm - boxplot () funktsiyasini namoyish etish**

Shu bilan *dengiz bo'yidagi* kutubxonaga qisqacha *sharhimiz tugadi* . Uning kaliti

o'ziga xos xususiyati shundaki, minimal sozlamalardan foydalanib, siz

grafikalar va jadvallarni ajoyib ingl

ro'yxatdan o'tish. Keyinchalik, biz vositalarni batafsil ko'rib chiqamiz

ma'lumotlar vizualizatsiyasi va jadvallarning ko'rinishini sozlash, bu

beradi *seaborn* .

128

|  |
| --- |
| **Sahifa 130** |

**7-bob. Diagramma ko'rinishini sozlash**

Dengiz *bo'yidagi* kutubxonada rang bilan ishlash vositalari mavjud

maket, uslubni sozlash, katak va diagramma o'qlari, maket

grafikalar, afsona va shriftlar bilan ishlash.

**7.1 *dengiz bo'yidagi*uslublar**

Dengiz *dengizining* o'ziga xos xususiyati *shundaki* , bu grafikalar va jadvallar

uni qurish, chiroyli ko'rinishga ega bo'lish uchun ishlatish mumkin.

Ishlab chiquvchilar foydalanuvchi ekanligiga ishonch hosil qilishga harakat qilishdi

natija ko'rinishini sozlash uchun minimal harakatlarni talab qildi

ularning ishi. Tashqi ko'rinishini o'rnatishning eng oson va tezkor usuli

bu oldindan tayyorlangan uslublardan birini ishlatishdir. Shuningdek, bor

shaxs bilan o'z uslublaringizni yaratish va ulardan foydalanish qobiliyati

ro'yxatdan o'tish.

Uslubni o'rnatish uchun set\_style () funktsiyasidan foydalaning

quyidagi imzo:

set\_style (uslub = Yo'q , rc = Yo'q )

Funktsiya parametrlari:

• uslub: dict, Hech narsa yoki { **'** darkgrid **'** to'plamidan qiymat ,

**'** whitegrid **'** , **'** dark **'** , **'** white **'** , **'** ticks **'** }

Parameters Tayyorlangan parametrlar yoki uslub nomi bilan lug'at

o'rnatilgan.

• rc: dict, ixtiyoriy

◦ O'tkazilgan parametrlarni bekor qilish uchun javobgardir

uslub uchun uslub argumenti.

129

|  |
| --- |
| **Sahifa 131** |

Diagrammaning joylashuvi uchun javobgar parametrlar ro'yxatini oling

axes\_style () funktsiyasidan foydalanishingiz mumkin:

axes\_style ( uslub = Yo'q , rc = Yo'q )

Ushbu funktsiya argumentlarining ro'yxati va funktsional maqsadi

set\_style () bilan bir xil.

Birinchidan, keling, mavjud bo'lgan uslublar to'plami bilan tanishib chiqamiz

foydalanish. Ushbu namoyish uchun *parvozlar* to'plamidan foydalanamiz .

Yuklaymiz:

reyslar = sns.load\_dataset ( **"parvozlar"** )

Tanlagan yo'lovchilar soniga bog'liqlikni yaratamiz

funktsiyasi yordamida yildan-yilga sayohat qilish uchun havo transporti

lineplot ().

*Seaborn* quyidagi uslublar to'plamini taqdim etadi: darkgrid, whitegrid,

quyuq, oq, shomil.

Darkgrid uslubi:

sns.set\_style ( **"darkgrid"** )

sns.lineplot (x = **'yil'** , y = **'yo'lovchilar'** , ma'lumotlar = reyslar)

**Shakl 7.1 - Darkgrid dizayn uslubi**

130

|  |
| --- |
| **132-bet** |

Whitegrid uslubi:

sns.set\_style ( **"whitegrid"** )

sns.lineplot ( x = **'yil'** , y = **'yo'lovchilar'** , ma'lumotlar = reyslar)

**Shakl 7.2 - Whitegrid dizayn uslubi**

To'q uslub:

sns.set\_style ( **"qorong'i"** )

sns.lineplot ( x = **'yil'** , y = **'yo'lovchilar'** , ma'lumotlar = reyslar)

**7.3-rasm - To'q uslub**

131

|  |
| --- |
| **133-bet** |

Oq uslub:

sns.set\_style ( **"oq"** )

sns.lineplot ( x = **'yil'** , y = **'yo'lovchilar'** , ma'lumotlar = reyslar)

**7.4-rasm - Oq dizayn uslubi**

Shomil uslubi:

sns.set\_style ( **"shomil"** )

sns.lineplot ( x = **'yil'** , y = **'yo'lovchilar'** , ma'lumotlar = reyslar)

**Shakl 7.5 - Shomillarning dizayni uslubi**

132

|  |
| --- |
| **Sahifa 134** |

O'zgartirish uchun mavjud bo'lgan parametrlar ro'yxatini olishingiz mumkin

axes\_style () funktsiyasidan foydalanib:

>>> sns.axes\_style ()

{ **'axes.axisbelow'** : To'g'ri ,

**'axes.edgecolor'**: **'oq'**,

**'axes.facecolor'**: **'# EAEAF2'**,

**'axes.grid'**: To'g'ri ,

**'axes.labelcolor'**: **'.15'**,

**'axes.spines.bottom'**: To'g'ri ,

**'axes.spines.left'**: To'g'ri ,

**'axes.spines.right'**: To'g'ri ,

**'axes.spines.top'**: To'g'ri ,

**'figure.facecolor'**: **'oq'**,

**'font.family'**: [ **'sans-serif'**],

**'font.sans-serif'**: [ **'Arial'**,

**"DejaVu Sans"**,

**"Ozodlik sansi"**,

**"Bitstream Vera Sans"**,

**'sans-serif'**],

**'grid.color'**: **'oq'**,

**'grid.linestyle'**: **'-'**,

**'image.cmap'**: **'raketa'**,

**'lines.solid\_capstyle'**: **'dumaloq'**,

**'patch.edgecolor'**: **'w'**,

**'patch.force\_edgecolor'**: To'g'ri ,

**'text.color'**: **'.15'**,

**'xtick.bottom'**: False , **'xtick.color'**: **'.15'**,

**'xtick.direction'**: **'out'**, **'xtick.top'**: False ,

**'ytick.color'**: **'.15'**, **'ytick.direction'**: **'out'**,

**'ytick.left'**: False , **'ytick.right'**: False }

133

|  |
| --- |
| **Sahifa 135** |

Barcha parametrlarda ularning funktsionalligini ko'rsatuvchi nomlar mavjud

uchrashuv. Whitegrid uslubining bir qator parametrlarini o'zgartiraylik:

sns.set\_style ( **"whitegrid"** , { **'axes.labelcolor'** : **"b"** , **'axes.edgecolor'** : **'r'** ,

**'xtick.color'**: **'g'**})

sns.lineplot (x = **'yil'** , y = **'yo'lovchilar'** , ma'lumotlar = reyslar)

**Shakl 7.6 - O'zgartirilgan Whitegrid uslubi**

**7.2 *dengiz osti kontekstlari***

*Dengiz dengizidagi* kontekstlar o'lchovni boshqarish uchun ishlatiladi

Tasvirlar. Grafik qaerda ishlatilishiga qarab:

u maqola yoki taqdimotning bir qismiga aylanadi, u yoki bu tanlanadi

uning elementlari ko'lami.

Kontekstni sozlash uchun set\_context () funktsiyasi ishlatiladi:

set\_context (kontekst = Yo'q , font\_scale = 1 , rc = Yo'q )

Funktsiya parametrlarini ko'rib chiqing:

• kontekst: dict, to'plamdagi parametr: { **'** paper **'** , **'** notebook **'** ,

**'** talk **'** , **'** poster **'** }, Hech narsa

Parameters Kontekstning parametrlari yoki ramziy nomi bilan lug'at.

134

|  |
| --- |
| **136-bet** |

• font\_scale: float, ixtiyoriy

◦ Shrift hajmini o'zgartirish uchun o'lchov koeffitsienti.

• rc: dict, ixtiyoriy

Properties xususiyatlarni bekor qilish parametrlari bilan lug'at,

kontekst argumenti orqali ko'rsatilgan kontekst.

Kontekst parametrlari ro'yxatini olish uchun funktsiyadan foydalaning

plotting\_context ():

plotting\_context ( kontekst = Yo'q , font\_scale = 1 , rc = Yo'q )

Funktsiya parametrlarining maqsadi xuddi shunday

set\_context ().

Kontekstlar bilan ishlashni namoyish etish uchun biz to'plamdan foydalanamiz

*ìrísí* ma'lumotlari :

ìrísí = sns.load\_dataset ( **"ìrísí"** )

Qog'oz mazmuni:

sns.set\_context ( **"qog'oz"** )

sns.scatterplot (x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí)

**7.7-rasm - Qog'oz konteksti**

135

|  |
| --- |
| **137-bet** |

Daftarning konteksti:

sns.set\_context ( **"daftar"** )

sns.scatterplot ( x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí)

**7.8-rasm - daftar konteksti**

Gapirish mazmuni:

sns.set\_context ( **"munozara"** )

sns.scatterplot ( x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí)

**7.9-rasm - nutq konteksti**

136

|  |
| --- |
| **Sahifa 138** |

Kontekst afishasi:

sns.set\_context ( **"plakat"** )

sns.scatterplot ( x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí)

**7.10-rasm - Kontekstli plakat**

O'zgartirish mumkin bo'lgan kontekst parametrlari ro'yxatini ko'rib chiqamiz.

Agar ma'lum bir kontekstning sozlamalarini topishingiz kerak bo'lsa, unda uning nomi

argument sifatida qabul qilinishi kerak:

>>> sns.plotting\_context ( **"daftar"** )

{ **'axes.labelsize'** : 12 ,

**'axes.linewidth'**: 1.25 ,

**'axes.titlesize'**: 12 ,

**'font.size'**: 12 ,

**'grid.linewidth'**: 1 ,

**'legend.fontsize'**: 11 ,

**'lines.linewidth'**: 1,5 ,

**'lines.markersize'**: 6 ,

**'patch.linewidth'**: 1 ,

**'xtick.labelsize'**: 11 ,

**'xtick.major.size'**: 6 ,

**'xtick.major.width'**: 1.25 ,

137

|  |
| --- |
| **139-bet** |

**'xtick.minor.size'**: 4 ,

**'xtick.minor.width'**: 1 ,

**'ytick.labelsize'**: 11 ,

**'ytick.major.size'**: 6 ,

**'ytick.major.width'**: 1.25 ,

**'ytick.minor.size'**: 4 ,

**'ytick.minor.width'**: 1 }

Keling, ba'zi parametrlarni o'zgartiraylik:

sns.set\_context ( **"daftar"** , font\_scale = 1.5 , rc = { **'lines.markersize'** : 15 ,

**'xtick.labelsize'**: 15.0 , **'ytick.labelsize'**: 15.0 })

sns.scatterplot ( x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí)

**7.11-rasm - o'zgartirilgan daftarning konteksti**

**parametrlar**

138

|  |
| --- |
| **140-bet** |

**7.3 Panjara va o'qlarni o'rnatish**

Masalan va uslublar bilan ishlashni namoyish etgan misollarda ko'rib turganingizdek

kontekstlar: grafika va o'qlar maydonida panjarani ingl.

uslubni (kontekstni) tanlash bilan o'zgaradi. Ushbu bo'lim bo'ladi

jadvalning ushbu xususiyatlarini aniq sozlash variantlari ko'rib chiqildi.

**7.3.1 Grid**

Taqdim etilgan uslub parametrlari

7.1-jadval va 7.2-jadvaldagi kontekst parametrlari.

**Jadval 7.1 - Panjara o'rnatish uchun uslub parametrlari**

**Parametr**

**Tavsif**

boltalar

Diagramma maydonida panjarani ko'rsatish uchun javobgardir.

To'g'ri - katakchani ko'rsating, False - yo'q.

grid.color

Tarmoq chizig'i rangi.

panjara.linestyle

Panjara chizig'i uslubi.

**Jadval 7.2 - Grid sozlamalari uchun kontekst parametrlari**

**Parametr**

**Tavsif**

grid.linewidth  panjara chizig'ining kengligi.

Keling, misollar yordamida panjara bilan ishlashni ko'rib chiqamiz. Birinchidan, uslubni o'rnatamiz

whitegrid va notebookning konteksti:

sns.set\_style ( **"whitegrid"** )

sns.set\_context ( **"daftar"** )

139

|  |
| --- |
| **141-bet** |

*Iris* to'plami uchun tarqaladigan joyni tuzamiz :

sns.scatterplot ( x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí)

**Shakl 7.12 - Amaliy uslub bilan grafik ko'rinish**

whitegrid **va** notebookning **konteksti**

Endi panjara parametrlarini o'zgartiramiz:

sns.set\_style ( **"whitegrid"** , rc = { **'grid.color'** : **'# ff0000'** , **'grid.linestyle'** : **'-'** })

sns.set\_context ( **"daftar"** , rc = { **'grid.linewidth'** : 3.0 })

Keling, nima bo'lganini ko'raylik:

sns.scatterplot (x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí)

**7.13-rasm - parametrlari o'zgartirilgan diagramma ko'rinishi**

**to'rlar**

140

|  |
| --- |
| **142-bet** |

**7.3.2 Uchastka maydoni va o'qlari**

Marja va o'qlarni o'rnatish uchun uslub va kontekst variantlari

grafikalar **"** o'qlar **"** prefiksidan boshlanadi (7.3 va 7.4-jadvallarga qarang).

**Jadval 7.3 - Panjara o'rnatish uchun uslub parametrlari**

**Parametr**

**Tavsif**

o'qlar

Tarmoqni (True / 'line') yoki ostiga qo'ying

(False) diagrammasi ustida.

eksa.edgecolor

Diagramma maydonining chegara rangi.

oqlari.facecolor

Diagramma maydonining rangi.

boltalar. labelcolor

Eksa yorliqlarining rangi.

boltalar. tikanlar. pastki

O'qni qutining pastki qismiga qo'ying (To'g'ri).

boltalar. tikanlar. chap

O'qni qutining chap tomoniga qo'ying (To'g'ri).

boltalar. tikanlar. to'g'ri

O'qni qutining o'ng tomoniga qo'ying (To'g'ri).

boltalar. tikanlar.top

O'qni qutining yuqori qismiga qo'ying (To'g'ri).

**Jadval 7.4 - Grid sozlamalari uchun kontekst parametrlari**

**Parametr**

**Tavsif**

o'qlar. chiziq kengligi

Grafika o'qlarining qalinligi.

o'qlar. sarlavha

Sarlavha hajmi.

141

|  |
| --- |
| **143-bet** |

Ushbu parametrlardan foydalanishga misol:

sns.set\_style ( **"whitegrid"** , rc = { **'axes.axisbelow'** : **'line'** ,

**'axes.edgecolor'**: **'qizil'**,

**'axes.facecolor'**: **'och yashil'**,

**'axes.labelcolor'**: **'qizil'**,

**'axes.spines.bottom'**: To'g'ri ,

**'axes.spines.left'**: To'g'ri ,

**'axes.spines.right'**: noto'g'ri ,

**'axes.spines.top'**: yolg'on })

sns.set\_context ( **"daftar"** , rc = { **'axes.labelsize'** : 15.0 ,

**'axes.linewidth'**: 2.5 ,

**'axes.titlesize'**: 20.0 })

sp = sns.scatterplot (x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí)

sp.set\_title ( **"Axes tune"** )

**7.14-rasm - parametrlari o'zgartirilgan diagramma ko'rinishi**

**chekka va o'qlar chizmasi**

142

|  |
| --- |
| **144-bet** |

Yana bir muhim elementi hisoblanadi **- Shomil** belgilari deb

grafik o'qi bo'yicha chizilgan. Ular bilan ishlash uchun parametrlardan foydalaniladi

"xtick" va "ytick" prefikslari bilan 7.5 va 7.6-jadvallarga qarang.

**7.5-jadval - Shomillarni o'rnatish uslubi parametrlari**

**Parametr**

**Tavsif**

xtick.bottom

Shomillarni pastki o'qga joylashtirish (To'g'ri).

xtick.top

Shomillarni yuqori o'qga joylashtirish (To'g'ri).

xtick.direction

Belgilangan chiziqlarning yo'nalishi ("tashqariga") yoki

*x* o'qi bo'yicha ichki ('in') *.*

xtick.color

*X* o'qi bo'yicha belgilash chiziqlarining rangi *.*

ytick.left

Shomillarni chap o'qga joylashtirish (To'g'ri).

ytick.right

Shomillarni o'ng o'qga joylashtirish (To'g'ri).

ytick.direction

Belgilangan chiziqlarning yo'nalishi ("tashqariga") yoki

*y* o'qi bo'yicha ichkariga ('in') kiriting *.*

ytick.color

*Y* o'qi bo'yicha belgilaydigan chiziqlarning rangi *.*

**7.6-jadval - Shomillarni o'rnatish uchun kontekst parametrlari**

**Parametr**

**Tavsif**

xtick.labelsize

*X* o'qi bo'yicha belgilash belgilarining kattaligi *.*

xtick.major.size

*X* o'qi bo'yicha asosiy belgilar yorlig'i uzunligi *.*

xtick.major.width

*X* o'qi bo'yicha asosiy belgilarning kengligi *.*

xtick.minor.size

*X* o'qi bo'yicha qo'shimcha belgilar yorlig'i uzunligi *.*

143

|  |
| --- |
| **Sahifa 145** |

xtick.minor.width

*X* o'qi bo'yicha qo'shimcha belgilash belgilarining kengligi *.*

ytick.labelsize

*Y* o'qi bo'yicha belgilash belgilarining kattaligi *.*

ytick.major.size

*Y* o'qi *ustidagi* asosiy belgilar yorlig'i uzunligi *.*

ytick.major.width

*Y* o'qi bo'yicha asosiy belgilar yorlig'i kengligi *.*

ytick.minor.size

*Y* o'qi *ustidagi* qo'shimcha belgilar yorlig'i uzunligi *.*

ytick.minor.width

*Y* o'qi bo'yicha qo'shimcha belgilash belgilarining kengligi *.*

Yordamchi qatorni ko'rsatish uchun ( *kichik* ) kerak

set\_minor\_locator () funktsiyasi yordamida lokatorni ulang. Bunday

manipulyatsiya allaqachon *Matplotlib* darajasidagi *ishdir* ( *seaborn*

buning ustiga amalga oshirildi), odatda *dengiz dengizlari* bilan *ishlashda* buni bajaring

kerak emas.

Yordamchi o'lchagichning dizaynini namoyish qilish uchun biz beramiz

misol:

sns.set\_style ( **"whitegrid"** , rc = { **'xtick.bottom'** : To'g'ri ,

**'xtick.color'**: **'qizil'**, **'xtick.direction'**: **'in'**,

**'xtick.top'**: To'g'ri ,

**'ytick.color'**: **'qizil'**, **'ytick.direction'**: **'in'**,

**'ytick.left'**: To'g'ri ,

**'ytick.right'**: To'g'ri })

sns.set\_context ( **"daftar"** , rc = { **'xtick.labelsize'** : 15.0 ,

**'xtick.major.size'**: 6.0 , **'xtick.major.width'**: 1.25 ,

**'xtick.minor.size'**: 4.0 , **'xtick.minor.width'**: 2.0 ,

**'ytick.labelsize'**: 15.0 ,

144

|  |
| --- |
| **146-bet** |

**'ytick.major.size'**: 6.0 , **'ytick.major.width'**: 1.25 ,

**'ytick.minor.size'**: 4.0 , **'ytick.minor.width'**: 1.0 })

mpl sifatida import matplotlib

sp = sns.scatterplot (x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí)

sp.get\_xaxis (). set\_minor\_locator (mpl.ticker.AutoMinorLocator ())

sp.get\_yaxis (). set\_minor\_locator (mpl.ticker.AutoMinorLocator ())

sp.grid (b = To'g'ri , bu = **'kichik'** , rang = **'och yashil'** , chiziq kengligi = 0,5 )

**Shakl 7.15 - O'zgartirilgan Shomil dizayni bilan diagramma**

**7.4 Afsona**

Agar siz foydalansangiz, jadvaldagi afsona avtomatik ravishda ko'rsatiladi

ma'lumotlarni bitta yoki bo'yicha guruhlash uchun qo'shimcha parametr

boshqa belgilar. Guruhlarni ajratib ko'rsatish uchun rangdan foydalanishingiz mumkin

(parametr tusi) yoki o'lchov (parametr hajmi).

Afsonasi bo'lgan grafikaga misol keltiramiz:

sns.scatterplot (x = **'sepal\_length'** , y = **'petal\_length'** , hue = **" types "** ,

ma'lumotlar = ìrísí)

145

|  |
| --- |
| **147-bet** |

**7.16-rasm - Afsonaviy grafik**

Dengiz *bo'yidagi* kutubxonaning *o'zi*

deyarli emas

vizual ko'rinishni sozlash uchun vositalarni taqdim etadi

afsonalar. Faqat parametr - legend.fontsize for

afsonaning shrift hajmini guruhdan ajratilgan boshqarish

kontekst sozlamalari:

sns.set\_style ( **"whitegrid"** )

sns.set\_context ( **"munozara"** , rc = { **'legend.fontsize'** : 10.0 })

**7.17-rasm - Shrift o'lchamlari o'zgartirilgan afsona**

146

|  |
| --- |
| **148-bet** |

Agar ko'proq aniq sozlash kerak bo'lsa: afsonaning joylashuvi, uning

hajmi, va hokazo. Buning uchun siz vositalarni ishlatishingiz kerak,

*Matplotlib quyidagilarni* taklif qiladi :

sns.set\_style ( **"whitegrid"** )

sns.set\_context ( **"daftar"** )

sp = sns.scatterplot (x = **'sepal\_length'** , y = **'petal\_length'** , hue = **" types "** ,

ma'lumotlar = ìrísí)

sp.legend (loc = **'center right'** , bbox\_to\_anchor = ( 1.35 , 0.5 ), ncol = 1 )

**Shakl 7.18 - foydalanib afsonaning joylashishini o'zgartirish**

***matplotlib* kutubxonasidan usul**

**7.5 shrift**

Font.family parametrlari (uslubdan) va

font\_scale (kontekstdan):

sns.set\_style ( **"whitegrid"** , rc = { **'font.family'** : [ **'fantaziya'** ]})

sns.set\_context ( **"daftar"** , font\_scale = 1,5 )

sp = sns.scatterplot (x = **'sepal\_length'** , y = **'petal\_length'** , hue = **" types "** ,

ma'lumotlar = ìrísí)

sp.legend (loc = **'markaz o'ng'** , bbox\_to\_anchor = ( 1.5 , 0.5 ), ncol = 1 )

147

|  |
| --- |
| **149-bet** |

**7.19-rasm - Diagrammadagi shriftni o'zgartirish**

**7.6 Rang bilan ishlash**

*Seaborn Matplolib tomonidan* taqdim etilgan rang sxemalaridan foydalanadi ,

bu **"***2.3.2 Chiziq rangi***"** bo'limida batafsil tavsiflangan . Bilan ishlash

ranglar sxemasi kutubxonasi bir qator funktsiyalarni taqdim etadi,

keyinchalik ushbu bo'limda batafsil muhokama qilinadi.

Color\_palette () funktsiyasi

Ranglar ro'yxatini qaytaradi.

Funktsiya prototipi:

color\_palette ( palitra = Yo'q , n\_colors = Yo'q , desat = Yo'q )

Funktsiya parametrlari:

• palitra: Yo'q, satr, ro'yxat, ixtiyoriy

◦ Paletaning nomi yoki ranglar to'plami. Agar qiymat Yo'q bo'lsa, unda

joriy palitrasi qaytariladi.

• n\_colors: int, ixtiyoriy

◦ Paletadagi ranglar soni.

148

|  |
| --- |
| **Sahifa 150** |

• desat: float, ixtiyoriy

◦ Doygunlikni boshqarish koeffitsienti, 1 asl

ranglarning namoyishi.

Set\_palette () funktsiyasi

Joriy rang palitrasini o'rnatadi. Uchrashuv

parametrlar color\_palette () funktsiyasi uchun ko'rsatilgan parametrlar bilan bir xil.

Funktsiya prototipi:

set\_palette (palitrasi, n\_colors = yo'q , desat = yo'q , color\_codes = noto'g'ri )

Qo'shimcha variantlar:

• color\_codes: bool

◦ Agar parametr True bo'lsa, unda bekor qilish amalga oshiriladi

qisqa nomlar bilan bog'liq ranglar ('r', 'g', ...) in

belgilangan palitraga muvofiq.

Keling, ba'zi rang to'plamlarini tanlab ko'rib chiqamiz.

Amaldagi rang sxemasi:

sns.palplot (sns.color\_palette ())

Accent palitrasi:

sns.set\_palette ( **"Urg'u"** )

sns.palplot (sns.color\_palette ())

149

|  |
| --- |
| **Sahifa 151** |

Accent palitrasidan atigi uchta rangni olamiz:

sns.set\_palette ( **"Accent"** , n\_colors = 3 )

sns.palplot (sns.color\_palette ())

Palitra tab10:

sns.set\_palette ( **"tab10"** , n\_colors = 3 , desat = 1 )

sns.palplot (sns.color\_palette ())

Doygunlikni o'zgartirish:

sns.set\_palette ( **"tab10"** , n\_colors = 3 , desat = 0,5 )

sns.palplot (sns.color\_palette ())

sns.set\_palette ( **"tab10"** , n\_colors = 3 , desat = 0,1 )

sns.palplot (sns.color\_palette ())

Set\_color\_codes () funktsiyasi

Yorqinlikni va to'yinganlikni o'zgartiradi, agar rangni belgilash uchun

bitta harfli qisqartmalar ishlatiladi.

150

|  |
| --- |
| **Sahifa 152** |

Funktsiya prototipi:

set\_color\_codes (palitrasi = **"chuqur"** )

Funktsiya parametri:

• palitra: {'chuqur', 'ovozsiz', 'pastel', 'qorong'i', 'yorqin',

'colorblind'}

◦ dan Palette nomi *seaborn* kutubxona .

Mana ba'zi bir misollar.

Chuqur palitrasi:

sns.set\_color\_codes ( **"chuqur"** )

sns.barplot (x = **'turlari'** , y = **'petal\_length'** , ma'lumotlar = ìrísí, palitrasi = [ **'r'** , **'g'** ,

**'b'**])

**Shakl 7.20 - chuqur palitrasi**

151

|  |
| --- |
| **Sahifa 153** |

Ovozsiz palitrasi:

sns.set\_color\_codes ( **"ovozsiz"** )

sns.barplot ( x = **'turlari'** , y = **'petal\_length'** , ma'lumotlar = ìrísí, palitrasi = [ **'r'** , **'g'** ,

**'b'**])

**7.21-rasm - tovushsiz palitrasi**

To'q palitrasi:

sns.set\_color\_codes ( **"qorong'i"** )

sns.barplot ( x = **'turlari'** , y = **'petal\_length'** , ma'lumotlar = ìrísí, palitrasi = [ **'r'** , **'g'** ,

**'b'**])

**Shakl 7.22 - qorong'i palitrasi**

152

|  |
| --- |
| **Sahifa 154** |

**8-bob. Ma'lumotlardagi munosabatlarni ingl**

*Seaborn* ko'rsatish uchun funktsiyalarni taqdim etadi (8.1-jadvalga qarang)

chiziqli grafikalar va sochilgan chiziqlar ko'rinishidagi ma'lumotlar.

**Jadval 8.1 - munosabatlarni vizualizatsiya qilish uchun *Seaborn* funktsiyalari**

**ma'lumotlar**

**Funktsiya**

**Tavsif**

relplot

Bilan munosabatlarni ko'rish uchun umumiy interfeys

tartibini sozlash uchun variantlar.

lineplot

Semantik guruhlash bilan chiziqli jadval.

sochilib ketish

Semantik guruhlash bilan tarqatish.

Taqdim etilgan funktsiyalar to'plamini shartli ravishda ikkiga bo'lish mumkin

guruhlar: birinchisi, qurilish uchun ixtisoslashtirilgan funktsiyalar

ma'lum bir turdagi uchastkalar, ularga lineplot () va kiradi

scatterplot (); ikkinchi guruh - figura darajasining funktsiyalari (substratlar? on)

qaysi uchastkalarni aks ettiradi): relplot (), beradi

birinchi guruhdagi vositalarning funktsionalligi va qo'shimcha ravishda

grafiklarni tuzish qobiliyatini beradi: imkon beradi

qiymatiga qarab grafiklarni alohida maydonlarda ko'rsatish

yoki boshqa biron bir belgi. Relplot () ko'proq deb ayta olamiz

lineplot () va scatterplot () uchun yuqori darajadagi interfeys.

**8.1 Funksiyalarning umumiy parametrlari**

Funktsiyalar uchun umumiy bo'lgan parametrlarni ko'rib chiqing

ma'lumotlardagi munosabatlarni ingl.

153

|  |
| --- |
| **Sahifa 155** |

**8.1.1 Asosiy dalillar**

Asosiy funktsiya argumentlari:

• x, y: ma'lumotlar to'plamidagi o'zgaruvchan nomlar, ixtiyoriy

◦ To'plamdan *x* va *y* o'qlarini o'ziga xos xususiyatlarga bog'lang

ma'lumotlar parametri orqali o'tgan ma'lumotlar. Ma'lumotlar kerak

raqamli turdagi bo'lishi. Lineplot () va scatterplot () funktsiyalari uchun

to'g'ridan-to'g'ri ma'lumotlar bilan vektorlarni uzatish joizdir.

Parametrlar Yo'q bo'lishi mumkin, bu holda shunday bo'ladi

butun ma'lumotlar to'plamini ma'lumotlardan yaratdi.

• ma'lumotlar: DataFrame, ixtiyoriy

◦ ustunlar joylashgan pandas.DataFrame ma'lumotlar to'plami

O'zgaruvchan nomlar, satrlar qiymatlardir. Ustun nomlari,

siz tasavvur qilmoqchi bo'lgan ma'lumotlar uzatiladi

x va y parametrlari.

**8.1.2 Axborot tarkibini oshirish parametrlari**

**grafikalar**

Grafiklarni yanada ma'lumotli qilish uchun funktsiya

ma'lumotlar munosabatlari vizuallashtirishlari quyidagilarni ta'minlaydi

parametrlari:

• ci: int, 'sd' yoki None, ixtiyoriy

Displayed Ko'rsatilgan ishonch oralig'ining o'lchamini aniqlaydi.

Agar qiymat 'sd' bo'lsa, unda ishonch oralig'i o'rniga

standart og'ish ko'rsatiladi.

• hue: ma'lumotlar to'plamidagi o'zgaruvchining nomi, ixtiyoriy

◦ Ma'lumotlar to'plamidagi atributni ishlatilishini o'rnatadi

ma'lumotlarning ranglarini ajratish uchun. Vizual ravishda guruhlar bo'ladi

alohida chiziqlar (nuqta) sifatida taqdim etilgan, har xil

154

|  |
| --- |
| **Sahifa 156** |

rang. Lineplot () va scatterplot () funktsiyalari imkon beradi

ma'lumotlar bilan vektorlarni to'g'ridan-to'g'ri parametrga o'tkazing, uchun

relplot () ga ruxsat berilmaydi.

• palitra: palitra nomi, ro'yxat, dikt, ixtiyoriy

◦ sxemasidan rang ajratish uchun foydalanish uchun

ma'lumotlar to'plami rangda ko'rsatilgan xarakteristikaning qiymatlari bo'yicha.

*Seaborn* turli xil rang sxemalarini taqdim etadi,

foydalanish mumkin bo'lgan, batafsilroq tavsiflangan

Qo'shimcha ma'lumot olish uchun " *7.5 Rang bilan ishlash* " bo'limiga qarang . Siz rangingizni belgilashingiz mumkin

o'yin aniqlangan lug'at orqali sxema

atributning ranglari va ranglari kutubxonadan olingan

*Matplotlib* .

• hue\_order: ro'yxat, ixtiyoriy

◦ Ma'lumotlar to'plamidagi ma'lumotlarga ranglarning qo'llanilish tartibini belgilaydi,

rang parametridan o'tdi.

• hue\_norm: tuple yoki Normalize sinf ob'ekti, ixtiyoriy

◦ DataSet bo'lsa Data normallashtirish qo'llanilishi mumkin

rang parametrida ko'rsatilgan raqamli turdagi. Uchun

hech qanday ma'lumotni normalizatsiya qilishdan foydalanilmaydi.

• hajmi: ma'lumotlar to'plamidan o'zgaruvchi nomi, ixtiyoriy

◦ Foydalaniladigan ma'lumotlar to'plamidagi atributni belgilaydi

ma'lumotlarni o'lchamlari bo'yicha ajratish. Vizual ravishda guruhlar bo'ladi

alohida chiziqlar (nuqta) sifatida taqdim etilgan, har xil

kengligi (hajmi). Lineplot () va scatterplot () funktsiyalari

ma'lumotlar bilan vektorlarni to'g'ridan-to'g'ri parametrga o'tkazishga imkon beradi,

relplot () uchun bu yaroqsiz.

• size\_order: ro'yxat, ixtiyoriy

155

|  |
| --- |
| **Sahifa 157** |

◦ Chiziq qalinligi o'rtasida taqsimlanish tartibini belgilaydi

o'lchov parametri orqali ko'rsatilgan ma'lumotlar to'plamidan elementlar.

• size\_norm: tuple, Normalize sinf ob'ekti, ixtiyoriy

◦ Belgilangan ma'lumotlar to'plami uchun ma'lumotlarning normalizatsiyasini belgilaydi

hajmi parametri orqali. Faqat uchun ishlatilishi mumkin

raqamli qiymatlar.

• uslub: ma'lumotlar to'plamidagi o'zgaruvchining nomi, ixtiyoriy

◦ Ma'lumotlar to'plamidagi atributni ishlatilishini o'rnatadi

ma'lumotlarni uslubi bo'yicha ajratish. Vizual ravishda guruhlar bo'ladi

alohida chiziqlar (nuqta) sifatida taqdim etilgan, har xil

chiziq uslubi va / yoki marker turi. Lineplot () va

scatterplot () sizga vektorlarni ma'lumotlar bilan uzatishga imkon beradi

to'g'ridan-to'g'ri parametr, relplot () uchun ruxsat berilmaydi.

• markerlar: bool, list, dict, ixtiyoriy

◦ Markerlar turini belgilaydi. Agar parametr False bo'lsa, u holda

markerlardan foydalanilmaydi, agar To'g'ri bo'lsa, ular qo'llaniladi

standart markerlardan foydalaniladi. Marker turi mumkin

o'rtasida yozishmalar o'rnatadigan lug'at orqali o'rnatiladi

parametrdan o'tgan ma'lumotlar to'plamidagi qiymatlar

uslub va marker kodi. Marker kodlari mos keladi

*Matplotlib-da* ishlatiladi .

• style\_order: ro'yxat, ixtiyoriy

◦ Uslublarning qo'llanilish tartibini belgilaydi.

**8.2 Chiziqli diagramma. Lineplot () funktsiyasi**

Ma'lumotlardagi munosabatlarni chiziqli grafik sifatida tasavvur qilish

lineplot () funktsiyasi qo'llaniladi. Ushbu turdagi grafikalar ko'pincha

vaqt qatorlari va orasidagi bog'liqliklarni tasavvur qilish uchun ishlatiladi

156

|  |
| --- |
| **Sahifa 158** |

doimiy bo'lgan o'zgaruvchilar, masalan, kabi

harorat, bosim, narx va boshqalar.

**8.2.1 Lineplot () funktsiyasi bilan tanishish**

Bir o'zgaruvchi bilan boshqasi o'rtasida oddiy munosabatlarni o'rnatish bilan boshlaymiz.

Birinchidan, kerakli kutubxonalarni import qilaylik:

import seaborn sifatida SNS

np sifatida numpy- ni import qilish

pandalarni pd sifatida import qilish

Grafiklarning uslubini o'rnating (" *7.1 uslublari* " bo'limiga qarang

*seaborn* "):

sns.set\_style ( **"darkgrid"** )

Tasodifiy ma'lumotlar to'plamini yarataylik va funktsiyadan foydalanamiz

lineplot () ular orasidagi bog'liqlikni ko'rsatish uchun:

np.random.seed ( 123 )

x = [i uchun I -yilda qator ( 10 )]

y = np.random.randint ( 10 , size = len (x))

sns.lineplot (x, y)

Birinchi satrda generatorning urug'i ko'rsatilgan, buning uchun buni qilish kerak

dastur har safar boshlanganda bir xil bo'ladi

ma'lumotlar. Shundan so'ng biz abssissa o'qi ( *x* o'qi ) uchun ma'lumotlar to'plamlarini yaratamiz -

0 dan 9 gacha ketma-ketlik, ordinatalar o'qlari ( *y* o'qi ) tasodifiy sonlar

*x* ro'yxatidagi elementlar soniga teng miqdorda 0 dan 9 gacha .

Keyinchalik, bog'liqlikni ko'rsatish uchun lineplot () funktsiyasini chaqiramiz

yaratilgan to'plamlar orasida. Bizning holatlarimizda biz foydalanganmiz

variant, ma'lumotlar funktsiyaga to'g'ridan-to'g'ri, formada uzatilganda

vektorlar. Dastur natijasi 8.1-rasmda keltirilgan.

157

|  |
| --- |
| **Sahifa 159** |

**8.1-rasm - lineplot () funktsiyasining natijasi**

Amalda, tahlil qilish uchun ma'lumotlar ko'pincha shaklda tayyorlanadi

Pandalar kutubxonasining ketma-ket yoki DataFrame tuzilmalari. Keling, boshqasini yarataylik

bir nechta ma'lumotlar to'plamlari:

np.random.seed ( 123 )

namuna = [i uchun I -yilda qator ( 10 )]

y = np.random.randint ( 10 , size = len (sample))

z = np.random.randint ( 4 , size = len (sample))

ma'lumotlar = [namuna, y, z]

Keling, ularni bitta DataFrame-ga birlashtiramiz, buni quyidagicha qilishingiz mumkin:

df = pd.DataFrame (ma'lumotlar) .transpose ()

df.columns = [ **'namuna'** , **'y\_val'** , **'z\_val'** ]

yoki quyidagicha:

df = pd.DataFrame ({ **'sample'** : sample, **'y\_val'** : y, **'z\_val'** : z})

Olingan ma'lumotlar to'plamini ko'rib chiqamiz: df strukturasining mazmuni

8.2-jadvalda keltirilgan

158

|  |
| --- |
| **Sahifa 160** |

**Jadval 8.2 - df strukturasining tarkibi**

**namuna**

**y\_val**

**z\_val**

0

2018-04-02 121 2

2018-04-02 121 2

1

2018-04-02 121 2

3

2018-04-02 121 2

6

1

3

1

0

4

3

2018-04-02 121 2

besh

to'qqiz

0

6

6

3

7

1

1

8

0

3

to'qqiz

1

2018-04-02 121 2

Yordamida df to'plamidan y\_valga bog'liqligini namoyish qilaylik

lineplot () funktsiyalari:

sns.lineplot (x = **'namuna'** , y = **'y\_val'** , ma'lumotlar = df)

**Shakl 8.2 - y\_val atributining namunaga bog'liqligini ingl**

159

|  |
| --- |
| **Sahifa 161** |

Parametr x o'q uchun ma'lumotlar to'plamidan xarakteristikaning nomini belgilaydi

absissa, y orqali - ordinatalar o'qi uchun ma'lumotlar DataFrame

ma'lumotlar olinadi.

Agar sizga DataFrame to'plamidan barcha elementlarni ko'rsatish kerak bo'lsa, u holda

uni lineplot () ga ma'lumotlar parametri orqali uzatishingiz mumkin

x va y ni belgilash orqali:

sns.lineplot ( ma'lumotlar = df)

**8.3-rasm - butun ma'lumotlar bazasini aks ettirish df**

**8.2.2 matematik kutishni aks ettirish va**

**ishonch oralig'i**

Oldingi misollarda to'plamdan olingan qiymatlarga e'tibor bering

biz abssissa o'qi uchun parametr sifatida ko'rsatgan ma'lumotlar

(x o'qi) takrorlanmaydi. Val\_y ga qarshi val\_z grafigini tuzamiz.

Ushbu ma'lumotlar uchun yuqoridagi xususiyat endi bajarilmaydi:

bir xil val\_z qiymati (masalan, 0) mos keladi

bir nechta val\_y qiymatlari (bizning holatlarimizda bu 1 va 9).

160

|  |
| --- |
| **Sahifa 162** |

Keling, chiziqli grafikni tuzamiz:

sns.lineplot ( x = **'z\_val'** , y = **'y\_val'** , data = df)

**8.4-rasm - y\_val parametrining z\_valga bog'liqligi**

Bunday holda, qiymatlar ordinatalar o'qi bo'ylab chizilmaydi

parametr *y* va uning matematik kutilishi. Masalan, z\_val = 0 uchun,

y\_val = (1 + 9) / 2 = 5, biz buni diagrammada ko'rib turibmiz. Yengilroq maydon

chiziq atrofida 95% ishonch oralig'i bo'lsa, uning o'rniga

Agar siz standart og'ishni ko'rsatishni xohlasangiz, tayinlashingiz kerak

lineplot () funktsiyasining ci parametri, 'sd' qiymati:

sns.lineplot ( x = **'z\_val'** , y = **'y\_val'** , ci = **'sd'** , data = df)

**Shakl 8.5 - standart displeyga bog'liqlik grafigi**

**og'ishlar**

161

|  |
| --- |
| **Sahifa 163** |

Ishonch oralig'ini ko'rsatish (yoki standart og'ish)

Hech kimni c ga o'tkazib o'chirib qo'yish mumkin:

sns.lineplot ( x = **'z\_val'** , y = **'y\_val'** , ci = Yo'q , ma'lumotlar = df)

**Shakl 8.6 - Ishonch oralig'ini ko'rsatishni o'chirib qo'yish**

" *Tez boshlash* " bobida biz misol keltirdik,

yo'lovchilar tashishning to'plamdan yilga bog'liqligini tasvirlab beradi

berilgan *reyslar* , keling, yana bir bor keltiraylik:

reyslar = sns.load\_dataset ( **"parvozlar"** )

sns.lineplot (x = **'yil'** , y = **'yo'lovchilar'** , ma'lumotlar = reyslar)

**8.7-rasm - yo'lovchilar tashish hajmining to'plamdan yilga bog'liqligi**

***reyslar***

162

|  |
| --- |
| **Sahifa 164** |

Ishonch oralig'ini ko'rsatishni boshqarish uchun

(qattiq to'ldirish, yorliqlar) err\_style parametridan foydalanadi, bu

'band', 'bar' yoki None bo'lishi mumkin. "Band" ning ma'nosi

yuqoridagi rasmda ko'rsatilgandek qattiq to'ldirishni belgilaydi.

"Barlar" qiymati ishonch oralig'ini chiziqlar sifatida ko'rsatadi:

sns.lineplot (x = **'yil'** , y = **'yo'lovchilar'** , err\_style = **'bar'** , ma'lumotlar = parvozlar)

**Shakl 8.8 - ishonch oralig'ini quyidagicha ko'rsatish**

**segmentlar**

Agar siz err\_style parametrini Yo'q deb belgilasangiz, u holda

ishonch oralig'i ko'rsatilmaydi:

**Shakl 8.9 - Ishonch oralig'ini ko'rsatishni o'chirib qo'yish**

**err\_style parametridan foydalanib**

163

|  |
| --- |
| **Sahifa 165** |

**8.2.3 Grafikning ma'lumot tarkibini oshirish**

Lineplot () funktsiyasi uchta parametrni taqdim etadi

qo'shimcha bog'lash orqali siz grafikaning ma'lumot tarkibini oshirishingiz mumkin

rang, chiziq uslubi kabi ingl

va hajmi. Parametrlar bunga javobgardir: rang (rang), uslub (chiziq uslubi) va

hajmi (hajmi). Keling, ular bilan misollar ustida ishlashni ko'rib chiqaylik.

Keling, kerakli kutubxonalar to'plamini yuklaymiz va uslubni o'rnatamiz:

import seaborn sifatida SNS

pandalarni pd sifatida import qilish

sns.set\_style ( **"darkgrid"** )

Ma'lumotlar to'plamini ish uchun yuklaymiz:

df = sns.load\_dataset ( **"mpg"** )

Keling, avtomobil quvvatining ishlab chiqarilgan yiliga bog'liqligini namoyish etamiz:

sns.lineplot (x = **'model\_year'** , y = **'ot kuchi'** , ma'lumotlar = df)

**8.10-rasm - Avtomobil quvvatining yilga bog'liqligi grafigi**

**ozod qilish**

164

|  |
| --- |
| **Sahifa 166** |

**8.2.3.1 Rang sxemasini sozlash**

Agar vazifa avtomobil kuchining yilga bog'liqligini o'rganish bo'lsa

AQSh va Yaponiya uchun alohida-alohida chiqaring, foydalanishingiz mumkin

rang parametrlari bilan mamlakatlarni rang bilan ajratib ko'rsatish:

sns.lineplot ( x = **'model\_year'** , y = **'ot kuchi'** , hue = **'origin'** ,

ma'lumotlar = df [df [ **'kelib chiqishi'** ]! = **'Evropa'** ])

**Shakl 8.11 - Bo'lish uchun rang parametridan foydalanish**

**mamlakat ma'lumotlari**

Rangni ajratishni yanada sozlash uchun siz qilishingiz mumkin

lineplot () funktsiyasining quyidagi parametrlaridan foydalaning: palitra,

hue\_order, hue\_norm.

Rangni o'zgartirib, tartibini o'rnatamiz:

df\_usa\_jp = df [df [ **'kelib chiqishi'** ]! = **'** evropa **'** ]

sns.lineplot ( x = **'model\_year'** , y = **'ot kuchi'** , hue = **'origin'** ,

palitra = { **'usa'** : **'r'** , **'japan'** : **'b'** }, hue\_order = [ **'japan'** , **'usa'** ],

ma'lumotlar = df\_usa\_jp)

165

|  |
| --- |
| **Sahifa 167** |

**8.12-rasm - va parametrlar palitrasidan foydalanishga misol**

**hue\_order**

**8.2.3.2 Uslublarni sozlash**

Chiziq uslubi yordamida ma'lumotlarni segmentlashtirish uchun parametrdan foydalaning

uslubi, u orqali atribut o'rnatiladi

ajratish:

sns.lineplot (x = **'model\_year'** , y = **'ot kuchi'** , style = **'origin'** ,

ma'lumotlar = df\_usa\_jp)

**Shakl 8.13 - ajratish uchun uslub parametridan foydalanish**

**mamlakat ma'lumotlari**

166

|  |
| --- |
| **Sahifa 168** |

Siz chiziq uslubini quyidagilar yordamida sozlashingiz mumkin

parametrlar: chiziqlar, markerlar, style\_order. Ikki kishi uchun tavsif

ikkinchisi " *8.1.2 oshirish parametrlari* " bo'limida tarjima qilingan

*diagrammalarning ma'lumotliligi* ", chiziqlar haqidagi ma'lumotlar keltirilgan

quyida:

• chiziqcha: bool, list, dict, ixtiyoriy

◦ Lyuk turini aniqlaydi. Agar parametr False bo'lsa, u holda

qattiq chiziq ishlatiladi, agar To'g'ri bo'lsa, unda turlari

standart lyuklar. Amaliy soyalash turi

parametrga o'tgan ma'lumotlar to'plamidan o'ziga xos qiymatlar

uslubi, uni o'zingiz belgilashingiz mumkin, buning uchun sizga kerak

parametr qiymatlari va lyuk kodlarining mosligini o'rnating.

Tomonidan ham belgilangan bo'ladi Enkübasyon *bo'yicha Umumiy bo'lim* quyidagi shaklda: (uzunligi

segment, bo'shliq kattaligi) yoki bo'sh chiziq, qaysi

qattiq chiziqni belgilaydi.

Chiziqlar parametri bilan ishlashni misol yordamida ko'rib chiqamiz:

sns.lineplot ( x = **'model\_year'** , y = **'ot kuchi'** , style = **'origin'** ,

tire = { **'usa'** : (2, 2), **'**Japan **'** : (5, 2)}, data = df\_usa\_jp)

**8.14-rasm - chiziqlar parametri bilan ishlashni namoyish etish**

167

|  |
| --- |
| **Sahifa 169** |

Keling, markerlarni qo'shamiz:

sns.lineplot ( x = **'model\_year'** , y = **'ot kuchi'** , style = **'origin'** ,

tire = False , markers = { **'usa'** : **'^'** , **'** Japan **'** : **'o'** }, data = df\_usa\_jp)

**8.15-rasm - markerlar parametri bilan ishlashni namoyish etish**

**8.2.3.3 Chiziq qalinligini sozlash**

Ma'lumotlarni segmentlarga ajratishning navbatdagi usuli - qalinlikni belgilash

satrlari, o'lchov parametri buning uchun javobgardir. To'plam u orqali uzatiladi

asosiy to'plamni ajratish uchun ishlatiladigan ma'lumotlar

har xil qalinlikdagi chiziqda.

Namoyish uchun biz bilan birga bo'lgan *mpg* to'plamini o'zgartiramiz

oldingi bo'limlarda ishlagan:

fn\_filter = lambda x: To'g'ri bo'lsa, agar x ichida [ 4 , 8 ] bo'lmasa False

fn\_mod = lambda x: { 4 : **'to'rt'** , 8 : **'sakkiz'** } [x]

df\_mod1 = df [df [ **'silindrlar'** ] .map (fn\_filter)]. nusxa ko'chirish ()

df\_mod1 [ **'silindrlar'** ] = df\_mod1 [ **'silindrlar'** ] .map (fn\_mod)

Biz grafada sakkizta va qatorli turli xil kenglikdagi avtoulovlarni namoyish qilamiz

to'rt silindrli dvigatellar:

sns.lineplot (x = **'model\_year'** ,

y = **"ot kuchi"** ,

hajmi = **'silindrlar'** ,

ma'lumotlar = df\_mod1)

168

|  |
| --- |
| **Sahifa 170** |

**8.16-rasm - uchun o'lcham parametridan foydalanishni namoyish etish**

**ma'lumotlarni tsilindr soniga bo'lish**

Bundan tashqari, chiziq kengligi bilan ishlash uchun siz foydalanishingiz mumkin

parametrlar size\_order va size\_norm (" *8.1.2 parametrlari* " bo'limiga qarang

*grafiklarning axborot tarkibini oshirish* ").

Chiziq qalinligini belgilash tartibini o'zgartiraylik:

sns.lineplot ( x = **'model\_year'** , y = **'ot kuchi'** , size = **'silindrlar'** ,

size\_order = ( **'to'rt'** , **'sakkiz'** ), ma'lumotlar = df\_mod1)

**8.17-rasm - size\_order parametri bilan ishlashni namoyish etish**

169

|  |
| --- |
| **Sahifa 171** |

**8.2.4 Vaqt qatorlarini vizualizatsiya qilish**

*Pandas* kutubxonasi ishlash uchun kuchli vositalarni taqdim etadi

vaqt qatorlari. Biz ularni to'plam yaratish uchun ishlatamiz

indeks sifatida yanvar vaqt tamg'alariga ega bo'lgan ma'lumotlar

2018 yildan 2019 yil yanvarigacha bir oylik muddat va ular uchun qiymatlar

Biz tasodifiy yorliqlar yaratamiz:

date\_index = pd.date\_range (start = **'2018'** , freq = **'M'** , davrlar = 12 )

chop etish (sana\_index)

DatetimeIndex ([ **'2018-01-31'** , **'2018-02-28'** , **'2018-03-31'** , **'2018-04-30'** ,

**'2018-05-31'**, **'2018-06-30'**, **'2018-07-31'**, **'2018-08-31'**,

**'2018-09-30'**, **'2018-10-31'**, **'2018-11-30'**, **'2018-12-31'**],

dtype = **'datetime64 [ns]'** , freq = **'M'** )

Ish uchun ma'lumotlar to'plami:

np.random.seed ( 123 )

data\_set = np.random.randint ( 5 , size = len (date\_index))

DataFrame tuzilishini yarataylik:

df = pd.DataFrame (ma'lumotlar = ma'lumotlar\_ to'plami, indeks = sana\_indeks, ustunlar = [ **'qiymat'** ])

Birinchi beshta tarkibiy element:

df.head ()

**qiymat**

2018-01-31

2018-04-02 121 2

2018-02-28

4

2018-03-31

2018-04-02 121 2

2018-04-30

1

2018-05-31

3

170

|  |
| --- |
| **Sahifa 172** |

Df dan olingan ma'lumotlarni *seaborn* yordamida chiziqli grafik sifatida *namoyish qilaylik* :

sns.lineplot (ma'lumotlar = df)

**8.18-rasm - Vaqt qatorlarini vizualizatsiya qilish**

Df to'plamining indekslari ishlatilganligini unutmang

*x* o'qi uchun ma'lumot sifatida .

**8.3 Tarqoqlik diagrammasi. Scatterplot () funktsiyasi**

*Tarqoqlik* diagrammasi (tarqalish *chizig'i* )

ajralmas

ma'lumotlar bilan ishlashda analitik vosita chiziqli bilan birga

grafikalar, chiziqli jadvallar va boshqalar. Bunday qurish uchun

*seaborn* chartlari scatterplot () funktsiyasini ta'minlaydi.

**8.3.1 scatterplot () funktsiyasiga kirish**

Boshlashdan oldin ko'rsatilgan kutubxonalar to'plamini import qiling

"6 *.2 Tez boshlash* " bo'limiga qarang . Darkgrid uslubini yarataylik:

sns.set\_style ( **"darkgrid"** )

*Iris* ma'lumotlar to'plamini yuklaylik :

df = sns.load\_dataset ( **"ìrísí"** )

171

|  |
| --- |
| **Sahifa 173** |

Sepal\_length parametrining petal\_length ga bog'liqligini ko'rsatamiz

nuqta uchastkasining shakli:

sns.scatterplot (x = **'sepal\_length'** , y = **'petal\_length'** , data = df)

**8.19-rasm - scatterplot () funktsiyasini namoyish etish**

**8.3.2 *Tarqoq uchastkaning*axborot *tarkibini*oshirish**

Tarqoq uchastkaning ma'lumot tarkibini oshirish uchun siz qilishingiz mumkin

biz ko'rib chiqqan narsalarga o'xshash parametrlardan foydalaning

lineplot () bilan ishlashda - bu rangni (rangni) sozlash uchun rang, o'lcham -

marker hajmi, uslub - marker uslubi.

Tajribalar uchun biz allaqachon ma'lum bo'lgan ma'lumotlar to'plamidan foydalanamiz

*mpg* :

mpg = sns.load\_dataset ( **"mpg"** )

Diapazonga bog'liqlikning nuqta grafigini tuzamiz

Avtomobilni yonilg'i *sarfidan* ( *joy almashtirish*) ( *mpg* ):

sns.scatterplot (x = **'mpg'** , y = **'siljish'** , ma'lumotlar = mpg)

172

|  |
| --- |
| **Sahifa 174** |

**Rasm 8.20 - bog'liqligi *olish* parametr ustida *mpg***

**8.3.2.1 Rang sxemasini sozlash**

Ishlab chiqaruvchining mamlakatini rang bilan belgilang:

sns.scatterplot ( x = **'mpg'** , y = **'siljish'** , rang = **'kelib chiqish'** , ma'lumotlar = mpg)

**Shakl 8.21 - Bo'lish uchun rang parametridan foydalanish**

**mamlakat ma'lumotlari**

Parametrlar rang sxemasini qo'shimcha sozlash uchun javobgardir.

palitra, hue\_order va hue\_norm, ularning maqsadi va foydalanish tartibi

xuddi shu nomdagi lineplot () funktsiyasi parametrlariga o'xshaydi (bo'limga qarang

173

|  |
| --- |
| **Sahifa 175** |

" *8.2.3.1 Chiziqli jadvalning rang sxemasini o'rnatish")* . Uchun

ular bilan qanday ishlashni namoyish qilish uchun biz bir nechta misollarni keltiramiz. O'zgartirish

rang palitrasi:

sns.scatterplot ( x = **'mpg'** , y = **'siljish'** , hue = **'kelib chiqish'** ,

palitrasi = **'plazma'** , ma'lumotlar = mpg)

**8.22-rasm -** plazma **rang palitrasi yordamida misol**

Keling, o'zimizga tegishli ranglar to'plamini o'rnataylik:

cmap = { **'usa'** : **'y'** , **'** Japan **'** : **'g'** , **'europe'** : **'r'** }

sns.scatterplot (x = **'mpg'** , y = **'siljish'** , rang = **'kelib chiqish'** , palitrasi = smap,

ma'lumotlar = mpg)

**Shakl 8.23 ​​- Oldindan tayyorlangan bilan ishlashga misol**

**rang sxemasi**

174

|  |
| --- |
| **Sahifa 176** |

Ranglarni qo'llash tartibini o'zgartiraylik:

order = [ **'yaponiya'** , **'evropa'** , **'AQSh'** ]

sns.scatterplot (x = **'mpg'** , y = **'siljish'** , hue = **'kelib chiqish'** , hue\_order = tartib,

ma'lumotlar = mpg)

**Shakl 8.24 - ranglarni qo'llash tartibini o'zgartirish**

Oldingi misollardan farqli o'laroq, afsonaga e'tibor bering

mamlakatlar endi o'zgaruvchida ko'rsatilgan tartibda ro'yxatga olingan

buyurtma.

**8.3.2.2 Marker uslubini sozlash**

Bizning misolimizdagi markerlarning uslubi uchun miqdor javobgar bo'ladi

tsilindr ( *tsilindr* ):

sns.scatterplot (x = **'mpg'** , y = **'siljish'** , hue = **'kelib chiqish'** ,

style = **'silindrlar'** , ma'lumotlar = mpg)

plt.legend (bbox\_to\_anchor = ( 1.05 , 1 ), loc = 2 , borderaxespad = 0. )

175

|  |
| --- |
| **Sahifa 177** |

**8.25-rasm - uslub parametri bilan ishlashni namoyish etish**

Bundan tashqari, biz afsonani jadval maydonidan ko'chirishga majbur bo'ldik, chunki u

muhim miqdordagi joyni egallay boshladi.

Taqqoslash uchun bu erda uchun uslub parametrini ishlatishga misol keltirilgan

*ìrísí* to'plami :

sns.scatterplot ( x = **'sepal\_length'** , y = **'petal\_length'** , style = **' types '** ,

ma'lumotlar = ìrísí)

**Shakl 8.26 - uchun uslub parametri bilan ishlashni namoyish etish**

***ìrísí* ma'lumotlar to'plami**

176

|  |
| --- |
| **Sahifa 178** |

Ushbu to'plamdagi ma'lumotlar ingl

rang bo'yicha qo'shimcha segmentatsiya, orasidagi farqlar

ob'ektlar.

Marker uslubini yanada sozlash uchun siz foydalanishingiz mumkin

parametrlari markerlari va style\_order, ularning maqsadi bir xil

lineplot () funktsiyasi uchun bir xil nomlar. Biz ulardan ko'proq foydalanamiz

batafsil rasm sozlamalari: markerlar uslubini o'zgartirish va buyurtma berish

belgilarning taqdimoti:

mrks = { **'virginica'** : **'o'** , **'setosa'** : **'D'** , **'versicolor'** : **'X'** }

order = [ **'virginica'** , **'setosa'** , **'versicolor'** ]

sns.scatterplot (x = **'sepal\_length'** , y = **'petal\_length'** , style = **' types '** ,

ma'lumotlar = ìrísí, style\_order = buyurtma, markerlar = mrks)

**8.27-rasm - parametrlar bilan ishlashni namoyish qilish va**

**style\_order**

**8.3.2.3 marker hajmini sozlash**

Hajmi parametri bilan ishlashni namoyish qilish uchun to'plamga qaytamiz

ma'lumotlar *mpg,* biz transport vositasining og'irligi ( *tortish* belgisi ) uchun foydalanamiz

markerlar hajmini boshqarish:

177

|  |
| --- |
| **Sahifa 179** |

sns.scatterplot (x = **'mpg'** , y = **'siljish'** , rang = **'kelib chiqishi'** , hajmi = **'vazn'** ,

ma'lumotlar = mpg)

**8.28-rasm - o'lchov parametri bilan ishlashni namoyish etish**

Size\_order va size\_norm parametrlari buyurtma uchun javobgardir va

markerni o'lchamini normallashtirish, o'lchamlari o'lchamini aniqlaydi

markerlar:

sns.scatterplot ( x = **'mpg'** , y = **'siljish'** , rang = **'kelib chiqishi'** , hajmi = **'vazn'** ,

o'lchamlari = (10, 150), ma'lumotlar = mpg)

**8.29-rasm - o'lchovlar parametri bilan ishlashni namoyish etish**

178

|  |
| --- |
| **Sahifa 180** |

Keling, uchta vositadan ham foydalanamiz: rang, o'lcham va uslub

bir vaqtning o'zida:

sns.scatterplot ( x = **'mpg'** , y = **'siljish'** , rang = **'kelib chiqishi'** , hajmi = **'vazn'** ,

s tyle = **'silindrlar'** , ma'lumotlar = mpg)

plt.legend (bbox\_to\_anchor = ( 1.05 , 1 ), loc = 2 , borderaxespad = 0. )

**Shakl 8.30 - Bir vaqtning o'zida foydalanishni namoyish etish**

**parametrlarning rangi** , **hajmi va uslubi**

**8.4 Dala elementlarining ko'rinishini sozlash**

**grafik san'at**

Elementlarni sozlash uchun ba'zi foydali vositalarni ko'rib chiqamiz

chiziq uchastkalari maydonlari, ya'ni afsonaviy va eksa teglari. Ko'proq

ushbu va boshqa vositalar haqida 7-bobda batafsil o'qishingiz mumkin

" *Grafik ko'rinishini sozlash"* .

**8.4.1 Afsona**

Afsonaviy parametr afsonani ko'rsatish uchun javobgardir, mumkin

quyidagi qiymatlarni qabul qiling: 'qisqa' - ishlatiladi

179

|  |
| --- |
| **Sahifa 181** |

qisqartirilgan ma'lumotlar to'plami, "to'liq" - to'liq ma'lumotlar to'plamidan foydalaniladi

ma'lumotlar, False - afsona ko'rsatilmaydi.

*Misolni* ko'rib chiqing: *mgp* ma'lumotlar to'plamida *silindrlar* xususiyati olinadi

{3, 4, 5, 6, 8} to'plamining qiymatlari :

df = sns.load\_dataset ( **"mpg"** )

o'rnatilgan (df [ **"silindrlar"** ])

{ 3 , 4 , 5 , 6 , 8 }

Ushbu xususiyatga muvofiq ranglarni ajratish bilan grafikani siqilgan holda namoyish qilaylik

afsonaviy tasvirning varianti:

sns.lineplot ( x = **'model\_year'** , y = **'ot kuchi'** , hue = **'silindrlar'** , ma'lumotlar = df,

afsona = **'qisqacha'** )

**8.31-rasm - kamaytirilgan qiymatlar to'plamini aks ettirish**

**afsonaga kiring**

Ko'rib turganingizdek, afsonada 2 dan 2 gacha bo'lgan qiymatlar oralig'i ko'rsatilgan

8-chi. Endi xuddi shu grafikani tuzamiz, lekin afsonaviy parametrni tayinlaymiz

"to'liq" qiymati:

sns.lineplot ( x = **'model\_year'** , y = **'ot kuchi'** , hue = **'silindrlar'** , ma'lumotlar = df,

afsona = **'to'liq'** )

180

|  |
| --- |
| **Sahifa 182** |

**8.32-rasm - xarakterli qiymatlarning to'liq to'plamini in**

**afsona**

Endi afsonada *silindrlar* xususiyatining barcha qiymatlari keltirilgan . Olib ketish

syujet maydonidan afsona, buning uchun biz afsona parametrini o'rnatdik

Yolg'on:

sns.lineplot ( x = **'model\_year'** , y = **'ot kuchi'** , hue = **'silindrlar'** , ma'lumotlar = df,

afsona = Yolg'on )

**Shakl 8.33 - Afsonasiz diagramma maydoni**

181

|  |
| --- |
| **Sahifa 183** |

**8.4.2 Eksa yorliqlari**

*Seaborn* kutubxona *bo'lgan* asosida qurilgan *yuqori Matplotlib* sozlash ba'zi shunday,

u orqali ishlab chiqarilishi mumkin. Bu bolta uchun yorliqlarni tayinlash bilan bog'liq

grafik koordinatalari.

Keling, " *8.2.4 Time Series Visualization"* bo'limidan misolni ko'rib chiqaylik :

np.random.seed ( 123 )

date\_index = pd.date\_range (start = **'2018'** , freq = **'M'** , davrlar = 12 )

data\_set = np.random.randint ( 5 , size = len (date\_index))

df = pd.DataFrame (ma'lumotlar = ma'lumotlar\_ to'plami, indeks = sana\_indeks, ustunlar = [ **'qiymat'** ])

Imzolarni o'rnatish uchun xlabel () va ylabel () funktsiyalaridan foydalaning

matplotlib.pyplot-dan:

plt sifatida import matplotlib.pyplot

sns.lineplot (ma'lumotlar = df)

plt.xlabel ( **"Tadbir kuni"** , fontsize = 14 )

plt.ylabel ( **"Graf"** , shrift hajmi = 14 )

**Shakl 8.34 - Diagramma o'qlari uchun yorliqlarni o'rnatish**

182

|  |
| --- |
| **Sahifa 184** |

**8.4.3 Ma'lumotlar to'plamini saralash**

Odatiy bo'lib, grafikani namoyish qilishdan oldin, *dengiz dengiz* ishlab chiqaradi

ma'lumotlar to'plamini saralash, ushbu parametrni sozlash orqali o'chirib qo'yish mumkin

sort parametri False.

Dvigatel kuchining avtomobil ishlab chiqarilgan yiliga bog'liqligini yarataylik:

sns.lineplot (x = **'model\_year'** , y = **'ot kuchi'** , ma'lumotlar = df)

**Rasm 8.35 - chizma qaram bo'lib *ot kuchi* parametr ustida**

**ma'lumotlar to'plamini aralashtirishdan oldin *model\_year***

*MPG* ma'lumotlar to'plamini aralashtiramiz :

df = sns.load\_dataset ( **'mpg'** )

df = df.sample (frac = 1 )

Tanlangan parametrlarga bog'liqlik grafigini yana tuzamiz:

sns.lineplot (x = **'model\_year'** , y = **'ot kuchi'** , ma'lumotlar = df)

183

|  |
| --- |
| **Sahifa 185** |

**Rasm 8.36 - chizma qaram bo'lib *ot kuchi* parametr ustida**

**ma'lumotlar to'plamini aralashtirgandan so'ng *model\_year***

Ko'rib turganingizdek, natijaga nisbatan hech narsa o'zgarmadi,

biz aralashtirmasdan oldik (8.36-rasmga qarang). O'chirish

oldindan saralash opsiyasi:

sns.lineplot ( x = **'model\_year'** , y = **'ot kuchi'** , ma'lumotlar = df, sort = False )

**Rasm 8,37 - chizma qaram bo'lib *ot kuchi* parametr ustida**

***model\_year* tartiblash o'chirilgan**

Bunday holda, grafik allaqachon yuqoridagilardan farq qiladi.

variant.

184

|  |
| --- |
| **Sahifa 186** |

**8.5 Moslashtirish bilan munosabatlarni ingl**

**substrat. Relplot () funktsiyasi**

*Seaborn* kutubxonasi boshqa ko'rsatish funktsiyasini ta'minlaydi

ma'lumotlardagi munosabatlar: relplot (). Lineplot () va dan farqli o'laroq

scatterplot (), bu nafaqat sozlash imkoniyatini beradi

grafaning o'zi, shuningdek shakli - substrat,

barcha grafik komponentlar joylashgan.

Tashqi ko'rinishini sozlash variantlari xuddi shunday

lineplot () va scatterplot () funktsiyalari uchun berilgan. Quyida

sozlash uchun ishlatilishi mumkin bo'lgan argumentlarni taqdim etdi

raqamlar:

• qator, qator: str

◦ bo'linishni amalga oshiradigan xususiyatlarning nomlari

qatorlar va ustunlarga shakllar. Faqat foydalanish mumkin

kategorik belgilar.

• col\_wrap: int, ixtiyoriy

◦ Birlashtiriladigan ustunlar soni.

• row\_order, col\_order: qatorlar ro'yxati, ixtiyoriy

Listed Ro'yxatdagi qatorlar va / yoki ustunlar tartibi

xarakterli qiymatlar.

• kind: str, ixtiyoriy

◦ Ko'rsatilgan grafik turi: 'chiziq' - chiziqli tasvirni ko'rsatish uchun

grafikalar, "tarqoq" - nuqta.

• balandlik: int, float, ixtiyoriy

◦ Diagramma chegarasining balandligi dyuym.

• aspekt: ​​int, float, ixtiyoriy

◦ Maydonning grafaga nisbati, kengligi hisoblanadi

formulasi bo'yicha: tomon \* balandlik.

185

|  |
| --- |
| **Sahifa 187** |

• facet\_kws: diktat, ixtiyoriy

◦ FacetGrid uchun qo'shimcha argumentlar bilan lug'at

(raqamni boshqarish uchun sinf).

Keling, relplot () bilan ishlashni misollar orqali ko'rib chiqamiz. Biz uslubdan foydalanamiz

oq tarmoq:

sns.set\_style ( **'whitegrid'** )

*Iris* to'plami bilan boshlaymiz :

ìrísí = sns.load\_dataset ( **'ìrísí'** )

sns.relplot ( x = **'sepal\_length'** , y = **'petal\_length'** , kind = **' spread '** ,

ma'lumotlar = ìrísí)

**8.38-rasm - relplot () funktsiyasini namoyish etish**

Jadval deyarli biz " *8.3* " bo'limida olingan jadvaldan farq qilmaydi

*Tarqoq fitna.*Scatterplot () *funktsiyasi* ".

186

|  |
| --- |
| **188-bet** |

Endi uchta alohida maydonda diagrammalar tuzamiz

ajratuvchi parametrdan biz ìrísí turini ( *turlarini* ) ishlatamiz :

sns.relplot (x = **'sepal\_length'** , y = **'petal\_length'** , hue = **' types '** ,

kind = **'tarqalish'** , ma'lumotlar = iris, col = **'turlar'** )

**8.39-rasm - funktsiya parametri bilan ishlashni namoyish etish**

**relplot ()**

Satr parametri va yordamida chiziqlarni bo'linishni amalga oshiramiz

maydonlarning o'lchamini balandligi va tomoni bo'yicha belgilaylik:

sns.relplot ( x = **'sepal\_length'** , y = **'petal\_length'** , hue = **' types '** ,

turi = **'tarqalish'** , ma'lumotlar = ìrísí, satr = **'turlari'** , balandligi = 3 , tomoni = 3 )

187

|  |
| --- |
| **189-bet** |

**8.40-rasm - satr, balandlik** , **parametrlari bilan ishlashni namoyish etish**

**aspekt relplot () funktsiyalari**

188

|  |
| --- |
| **190-bet** |

**9-bob. Kategorik ma'lumotlarni ingl**

Ma'lumotlarni tahlil qilishda siz ko'pincha kategorik bilan ishlashingiz kerak

belgilar, ularning o'ziga xos xususiyati shundaki, ular qadriyatlarni qabul qilishadi

tartibsiz to'plamdan. Ushbu guruhga quyidagilar kiradi

rang (mashina, meva), shakli (kvadrat, doira,

uchburchak) va boshqalar. Ushbu turdagi ma'lumotlar uchun juda muhim

buyurtma munosabati joriy etiladi, ya'ni. operatsiyani endi qo'llash mumkin emas

yoki taqqoslash uchun kamroq.

Kategorik belgilarning o'ziga xos turi mavjud - bu tartibli

belgilar. Ular tabiatan raqamlar emas, balki ular uchun allaqachon

taqqoslash operatsiyasini, bunday belgilarga misollarni belgilashingiz mumkin

avtomobil turi (avtomobil, yuk mashinasi), ta'lim (bakalavr,

magistr, fan nomzodi, fan doktori).

*Seaborn* to'plamlarni ko'rsatish uchun vositalarni taqdim etadi

tarqalish uchastkalari ko'rinishidagi kategorik ma'lumotlar va ularning taxminlari va

kategorik ma'lumotlarning taqsimlanishini namoyish etish uchun vositalar.

Catplot () funktsiyasi barcha funktsiyalar uchun umumiy interfeysdir

o'zgartirish qobiliyatiga ega bo'lgan toifali ma'lumotlar bilan ishlash

shakl (pastki chiziq), ideal holda relplot () funktsiyasini takrorlaydi

" *8.5 Fon sozlamalari bilan aloqalarni vizualizatsiya qilish* " bo'limiga qarang *.*

*Funktsiyalar* relplot () *". Funktsiyalar* ro'yxati 9.1-jadvalda keltirilgan.

189

|  |
| --- |
| **191-bet** |

**9.1-jadval - Kategorik ma'lumotlarni vizualizatsiya qilish funktsiyalari**

**Funktsiya nomi**

**Tavsif**

catplot ()

Kategorik bilan ishlash uchun umumiy interfeys

substratni o'zgartirish imkoniyati bilan ma'lumotlar.

[stripplot ()](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=http://seaborn.pydata.org/generated/seaborn.stripplot.html%23seaborn.stripplot#seaborn.stripplot)

Kategorik vizualizatsiya vositalari

tarqatish uchastkalari ko'rinishidagi ma'lumotlar.

[to'da uchastkasi ()](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=http://seaborn.pydata.org/generated/seaborn.swarmplot.html%23seaborn.swarmplot#seaborn.swarmplot)

[boxplot ()](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=http://seaborn.pydata.org/generated/seaborn.boxplot.html%23seaborn.boxplot#seaborn.boxplot)

Tarqatishni vizualizatsiya qilish vositalari

to'liq ma'lumotlar.

[skripka (](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=http://seaborn.pydata.org/generated/seaborn.violinplot.html%23seaborn.violinplot#seaborn.violinplot) )

[boxenplot ()](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=http://seaborn.pydata.org/generated/seaborn.boxenplot.html%23seaborn.boxenplot#seaborn.boxenplot)

[nuqta (](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=http://seaborn.pydata.org/generated/seaborn.pointplot.html%23seaborn.pointplot#seaborn.pointplot) )

Vizualizatsiya vositalarini baholash

to'liq ma'lumotlar.

[barplot (](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=http://seaborn.pydata.org/generated/seaborn.barplot.html%23seaborn.barplot#seaborn.barplot) )

[countplot ()](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=http://seaborn.pydata.org/generated/seaborn.countplot.html%23seaborn.countplot#seaborn.countplot)

**9.1 Funksiyalarning umumiy parametrlari**

9.1-jadvalda keltirilgan funktsiyalar umumiy parametrlar to'plamiga ega

diagrammalar ko'rinishini sozlash uchun ularning tavsifi quyida keltirilgan

ushbu bo'lim.

**9.1.1 Asosiy parametrlar**

• x, y: ma'lumotlar to'plamidagi o'zgaruvchan nomlar, ixtiyoriy

◦ To'plamdan *x* va *y* o'qlarini o'ziga xos xususiyatlarga bog'lang

ma'lumotlar parametri orqali o'tgan ma'lumotlar. Funktsiyalar uchun

catplot () dan tashqari, bilan vektorlarni o'tkazish joizdir

to'g'ridan-to'g'ri ma'lumotlar, catplot () uchun emas. Parametrlar bo'lishi mumkin

Agar to'liq ko'rsatishni xohlasangiz, qiymat Yo'q

ma'lumotlar to'plami.

190

|  |
| --- |
| **192-bet** |

• ma'lumotlar: DataFrame

◦ ustunlar joylashgan pandas.DataFrame turidagi ma'lumotlar to'plami

bu xususiyat nomlari, satrlar qiymatlardir. Ustun nomlari,

siz tasavvur qilmoqchi bo'lgan ma'lumotlar uzatiladi

*x* va *y* parametrlari orqali .

**9.1.2 Axborot tarkibini oshirish parametrlari**

**grafikalar**

Grafiklarning ma'lumot tarkibini oshirish uchun ulardan foydalanish mumkin

parametrlari:

• hue: ma'lumotlar to'plamidagi o'zgaruvchining nomi, ixtiyoriy

◦ Foydalaniladigan ma'lumotlar to'plamidagi atributni belgilaydi

ma'lumotlarning ranglarini ajratish uchun. Vizual ravishda guruhlar bo'ladi

farq qiluvchi alohida elementlar sifatida taqdim etilgan

rang. Catplot () dan tashqari barcha funktsiyalar imkon beradi

ma'lumotlar bilan vektorlarni to'g'ridan-to'g'ri parametrga o'tkazing, uchun

catplot () ga ruxsat berilmaydi.

• order, hue\_order: qator qiymatlari ro'yxati

Items Ob'ektlarni ko'rsatish tartibi (yoki ranglarni belgilash).

• orient: **'** v **'** | 'h', ixtiyoriy

◦ Uchastka yo'nalishi, 'v' - vertikal, 'h' -

gorizontal.

• rang: *Matplotlib-* rang[17](https://translate.googleusercontent.com/translate_f#192) , ixtiyoriy

◦ Rang Barcha uchun ma'lumotlar yoki don ko'rsatilgan

gradient palitrasi.

• palitra: palitraning nomi, ro'yxat yoki diktant

◦ Rang uchun ishlatiladigan palitra

rangni belgilangan atribut qiymatlari bo'yicha ma'lumotlarni ajratish.

17 Rangni sozlashning mavjud usullaridan biri (" *2.3.2 Line color* " bo'limiga qarang )

191

|  |
| --- |
| **193-bet** |

**9.2 toifali ma'lumotlarning ingl**

**tarqalish jadvallari**

Kategorik ma'lumotlarni vizualizatsiya qilish uchun vositalarning birinchi guruhiga

stripplot () va swarmplot () funktsiyalarini o'z ichiga oladi.

Ushbu funktsiyalar uchun kengroq umumiy parametrlar to'plami mavjud

bob boshidagi bilan taqqoslaganda. Ularni ko'proq ko'rib chiqing

batafsil:

• hajmi: suzuvchi, ixtiyoriy

Of markerlarning diametri.

• edgecolor: *Matplotlib-* rang[18](https://translate.googleusercontent.com/translate_f#193), "kul", ixtiyoriy

Of markerlar chegarasining rangi. "Kul" qiymati uchun yorqinlik bo'ladi

ishlatilgan rang sxemasiga qarab belgilanadi.

• chiziq kengligi: suzuvchi, ixtiyoriy

◦ chegara chizig'i kengligi.

• dodge: bool, ixtiyoriy

◦ Agar parametr True bo'lsa va rang ishlatilsa, u holda diagramma

ma'lumotlar vizual ravishda ajratiladigan sifatida taqdim etiladi

guruhlar.

**9.2.1 Striptiz () funktsiyasi**

Sayohatimizni stripplot () funktsiyasi bilan boshlaylik. Ushbu funktsiya tuziladi

tarqoq fitna. Shunga o'xshash vosita bilan - funktsiya

scatterplot (), biz " *8.3* Scatterplot" bobida uchrashdik *.*

Scatterplot () *funktsiyasi* ".

18 Rangni sozlashning mavjud usullaridan biri (" *2.3.2 Chiziq rangi* " bo'limiga qarang )

192

|  |
| --- |
| **194-bet** |

*Iris* ma'lumotlar to'plamini yuklaylik :

ìrísí = sns.load\_dataset ( **"ìrísí"** )

Oddiy tarqatish diagrammasini tuzamiz:

sns.stripplot (x = **'tur'** , y = **'sepal\_length'** , ma'lumotlar = ìrísí)

**9.1-rasm - funktsiya yordamida tuzilgan diagramma**

**stripplot ()**

Ma'lumotlar to'plami uchun tarqoq uchastkani tuzishingiz kerak bo'lsa,

uning elementlari mustaqil massivlar, keyin

ularni to'g'ridan-to'g'ri x, y parametrlari orqali shakllantirmasdan o'tkazishingiz mumkin

oldindan DataFrame:

np.random.seed ( 321 )

x\_vals = np.random.randint ( 3 , hajmi = 200 )

y\_vals = np.random.randn ( 1 , len (x\_vals)) [ 0 ]

sns.stripplot ( x = x\_vals, y = y\_vals)

193

|  |
| --- |
| **195-bet** |

**9.2-rasm - funktsiya yordamida tuzilgan diagramma**

**to'g'ridan-to'g'ri uzatilgan ma'lumotlar bo'yicha stripplot ()**

**x va y parametrlari orqali**

Diagrammaning axborot mazmunini oshirish uchun siz foydalanishingiz mumkin

ranglar sxemasini, o'lchamini va zarb rangini sozlash imkoniyatlari

markerlar.

*Maslahatlar* to'plamini yuklang :

maslahatlar = sns.load\_dataset ( **"maslahatlar"** )

Keling, head () usuli yordamida dastlabki beshta qatorni chop etamiz, natija

funktsiyaning bajarilishi 9.2-jadvalda keltirilgan:

tips.head ()

**9.2-jadval - *Maslahatlar* to'plamining birinchi besh qatori**

***total\_bill uchi***

***jinsiy aloqa***

***chekuvchi***

***kun***

***vaqt***

***hajmi***

16.99 1.01

*Ayol*

*Yo'q*

*Quyosh*

*Kechki ovqat*

2018-04-02 121 2

10.34 1.66

*Erkak*

*Yo'q*

*Quyosh*

*Kechki ovqat*

3

21.01 3.50

*Erkak*

*Yo'q*

*Quyosh*

*Kechki ovqat*

3

23.68 3.31

*Erkak*

*Yo'q*

*Quyosh*

*Kechki ovqat*

2018-04-02 121 2

24.59 3.61

*Ayol*

*Yo'q*

*Quyosh*

*Kechki ovqat*

4

194

|  |
| --- |
| **196-bet** |

Ushbu to'plam chap uchining kattaligi to'g'risidagi ma'lumotlarni o'z ichiga oladi,

quyidagi mezonlarga bo'linadi: *total\_bill -* jami hisob-kitob (yilda.)

USD), *uchi uchi* hajmi (AQSh dollarida), *jinsi -* jinsi,

*chekuvchi -* mijoz chekadimi yoki yo'qmi, *vaqt -* vaqt: tushlik yoki tushlik, *hajmi -*

kompaniya hajmi.

Keling, uchi o'lchamiga ( *uchlariga* ) bog'liqlikning nuqta grafigini tuzamiz

ta'tilda bo'lganlar soni ( *hajmi* ):

sns.stripplot (x = **"size"** , y = **'tip'** , data = tips)

**9.3-rasm - Maslahatlar ( *maslahatlar* ) hajmining miqdorga bog'liqligi**

**ta'til kompaniyasidagi shaxs ( *hajmi* )**

Ko'rib turganingizdek, har xil o'lchamdagi markerlar guruhlari

agar kerak bo'lsa rang berish uchun turli xil ranglarda ta'kidlangan

bir xil ishlatilgan bo'lsa, u rang parametri orqali o'rnatilishi mumkin:

sns.stripplot ( x = **"size"** , y = **'tip'** , color = **"g"** , data = tips)

195

|  |
| --- |
| **197-bet** |

**9.4-rasm - rang parametri bilan ishlashni namoyish etish**

Endi odamlarning jinsini ta'kidlab o'tamiz, buning uchun biz rang parametrini tayinlaymiz

*jinsiy* qiymati :

sns.stripplot (x = "size" , y = 'tip' , hue = "sex" , ma'lumotlar = maslahatlar)

**9.5-rasm - rang parametridan foydalanishni namoyish etish**

**(1-misol)**

Yildan *«hajmi 'bo'lgan* haqiqiy belgisi, buning uchun moslashtirish

buyurtma ma'nosiz. Agar biz qaramlik diagrammasini tuzadigan bo'lsak

mijozning jinsidan maslahat, keyin kerakli narsani belgilash mumkin bo'ladi

vakillik.

196

|  |
| --- |
| **198-bet** |

Odatiy bo'lib, biz quyidagi natijani olamiz:

sns.stripplot ( x = **"vaqt"** , y = **'uchi'** , hue = **'jinsiy aloqa'** , ma'lumotlar = maslahatlar)

**9.6-rasm - rang parametridan foydalanishni namoyish etish**

**(misol 2)**

Rangni belgilash tartibini o'zgartiraylik:

sns.stripplot ( x = **"vaqt"** , y = **'uchi'** , hue = **'jinsiy aloqa'** , hue\_order = [ **'ayol'** , **'erkak'** ],

ma'lumotlar = maslahatlar)

**9.7-rasm - hue\_order parametri bilan ishlashni namoyish etish**

197

|  |
| --- |
| **Sahifa 199** |

Palitra parametridan foydalanib rang palitrasini o'zgartiramiz:

color\_palette = { **"Erkak"** : **"r"** , **"Ayol"** : **"g"** }

sns.stripplot (x = **"size"** , y = **'tip'** , hue = **"sex"** , palitrasi = color\_palette,

ma'lumotlar = maslahatlar)

**9.8-rasm - rang sxemasini parametr orqali o'zgartirish misoli**

**palitrasi**

Keling, markerlarning hajmini oshiraylik, ularga rang va kenglik chegarasini bering:

sns.stripplot (x = **"vaqt"** , y = **'uchi "** , hajmi = 10 , edgecolor = **" kulrang "** , chiziq kengligi = 1 ,

data = tips.sample (frac = 1 , random\_state = 123 ) [: 30 ])

**9.9-rasm - parametrlar orqali markerlarning ko'rinishini o'rnatish**

**hajmi, edgecolor va chiziq kengligi**

198

|  |
| --- |
| **Sahifa 200** |

Yo'nalishni belgilash uchun javobgar bo'lgan parametrlar bilan ishlaylik

jadvallar, xarakterli qiymatlarni ko'rsatish tartibi, bo'linish

qo'shimcha ma'lumotlar qiymati bo'yicha asosiy ma'lumotlar to'plami.

Belgilarni ko'rsatish tartibini o'zgartiraylik: *kechki ovqat* va *tushlik* joylarini o'zgartiraylik :

sns.stripplot ( x = **"vaqt"** , y = **'uchi "** , hue\_order = [ **" Kechki ovqat "** , **" Tushlik "** ],

ma'lumotlar = maslahatlar)

**9.10-rasm - hue\_order parametri bilan ishlashni namoyish etish**

Diagramma yo'nalishi orientatsiya parametri orqali o'rnatiladi:

sns.stripplot ( x = **"uchi"** , y = **'vaqt'** , orient = **'h'** , ma'lumotlar = maslahatlar)

**9.11-rasm - Diagrammaning gorizontal yo'nalishini o'rnatish**

199

|  |
| --- |
| **Sahifa 201** |

X va y belgilarining nomlari ham bo'lganligini unutmang

qayta tashkil etilgan. Diagrammaning ravshanligini oshirish uchun

rang parametrlari orqali ranglarni ajratishni ishlatadigan siz qilishingiz mumkin

hosil bo'lgan to'plamlarni ingl

dodge parametridan foydalanib:

sns.stripplot ( x = **"vaqt"** , y = **'uchi "** , hue = **" jinsiy aloqa "** , dodge = rost , ma'lumotlar = maslahatlar)

**9.12-rasm - Dodge parametri bilan ishlashni namoyish etish**

Taqdim etilgan to'plamlarning zichligi jitter parametri tomonidan boshqariladi.

Ushbu parametr, agar siz bir-birining ustiga chiqishni bartaraf etishingiz kerak bo'lsa, foydali bo'lishi mumkin.

ma'lumotlar to'plamlari o'rtasida. Standart sozlamalar bilan namuna:

sns.stripplot ( x = **"size"** , y = **'tip'** , data = tips)

200

|  |
| --- |
| **Sahifa 202** |

**9.13-rasm - uchun standart qiymati bo'lgan diagramma**

**jitter parametri**

Jitter qiymatini 0,03 ga o'zgartiring:

sns.stripplot ( x = **"size"** , y = **'tip'** , size = 5, jitter = 0.03, data = tips)

**9.14-rasm - jitter qiymati 0,03 ga teng bo'lgan diagramma**

Agar jitter True-ga o'rnatilsa, u avtomatik ravishda mos keladi

eng maqbul qiymat.

201

|  |
| --- |
| **Sahifa 203** |

**9.2.2 swarmplot () funktsiyasi**

Swarmplot () funktsiyasi va uchun parametrlari jihatidan

displey sozlamalari stripplot () ga o'xshaydi, bundan mustasno

u yo'q bo'lgan parametr jitter. Swarmplot () ning g'oyasi

diagrammada ko'rsatilgan nuqtalar bir-biriga mos kelmasligi, bu imkon beradi

to'plamlarda ma'lum qiymatlarning tarqalishi haqida xulosa chiqarish

ularning vizual taqsimoti to'g'risidagi ma'lumotlar.

Qanday ishlashni ko'rsatadigan bir nechta misollarni ko'rib chiqamiz

swarmplot ():

sns.swarmplot (x = **"vaqt"** , y = **'maslahat "** , ma'lumotlar = maslahatlar)

**9.15-rasm - swarmplot () funktsiyasi tomonidan qurilgan diagramma**

Quyidagi diagrammadan xulosa qilishimiz mumkinki, tushlikda

ko'pincha ular ikki dollar, tushlik paytida esa tushlik sifatida qoldiradilar

ikki-to'rt dollar.

Rang sxemasini o'rnatish uchun biz mos keladigan xususiyatdan foydalanamiz

mijozlar chekish yoki chekmasliklari uchun:

sns.swarmplot ( x = **"vaqt"** , y = **'uchi "** , hue = **" chekuvchi "** , ma'lumotlar = maslahatlar)

202

|  |
| --- |
| **Sahifa 204** |

**9.16-rasm - atribut bo'yicha ranglarni ajratishni namoyish etish**

**chekuvchi**

Keling, qo'shimcha ravishda 50 ta to'plamdan namuna olaylik

markerlarning o'lchamlarini, ularning rangini va chegara kengligini sozlaylik:

df = tips.sample (frac = 1 , random\_state = 123 ) [: 50 ]

sns.swarmplot (x = **"vaqt"** , y = **'uchi "** , hajmi = 10 , edgecolor = **" kulrang "** , chiziq kengligi = 2 ,

ma'lumotlar = df)

**9.17-rasm - o'lchov hajmi, rangi va**

**markerlarning chegara kengligi**

203

|  |
| --- |
| **Sahifa 205** |

Keling, mustaqil rang palitrasini o'rnatamiz:

color\_palette = { **"Erkak"** : **"g"** , **"Ayol"** : **"y"** }

sns.swarmplot (x = **"vaqt"** , y = **'uchi "** , hue = **" jinsiy aloqa "** , hajmi = 10 , edgecolor = **" kulrang "** ,

palitra = color\_palette, chiziq kengligi = 2 , ma'lumotlar = df)

**9.18-rasm - rang palitrasi o'zgartirilgan diagramma**

Orient parametri diagrammaning yo'nalishi uchun javob beradi, buyurtma uchun -

buyurtma, dodge parametri ma'lumotlarning vizual ajratilishini boshqaradi.

Keling, ulardan foydalanishni misollar bilan keltiraylik. Yo'nalishni o'zgartirish

va *vaqt* atributining qiymatlarini ko'rsatish tartibi :

sns.swarmplot (x = **"tip"** , y = **'time'** , order = [ **'Kechki ovqat'** , **'Tushlik'** ], orient = **'h'** ,

hue = **"sex"** , ma'lumotlar = maslahatlar)

**9.19-rasm - tartib va ​​orientatsiya parametrlari bilan ishlashni namoyish etish**

204

|  |
| --- |
| **Sahifa 206** |

Dodge parametri yordamida tanlovni guruhlarga bo'ling:

sns.swarmplot ( x = **"vaqt"** , y = **'uchi'** , hue = **"chekuvchi"** , qochish = rost , ma'lumotlar = maslahatlar)

**9.20-rasm - Dodge parametri bilan ishlashni namoyish etish**

**9.3 Kategorik taqsimotlarni vizuallashtirish**

**ma'lumotlar**

Amalda, statistik masalalarni echishda ko'pincha qurish kerak bo'ladi

tarqalish diagrammasi va tarqatish diagrammasi. Siz ulardan foydalanishingiz mumkin

bir qator uchun o'rganilgan ma'lumotlar to'plamlari haqida tasavvurga ega bo'lish

qiymatlari ushbu turga mos keladigan raqamli xususiyatlar

diagrammalar.

*Seaborn* tarqatishni tasavvur qilish uchun uchta funktsiyani taqdim etadi

toifadagi ma'lumotlar:

• boxplot () - ba'zida quti va mo'ylovlar diagrammasini tuzadi

transkripsiya orqali chaqiriladi: boxplot, u aks etadi

median, kvartillar va tashqaridan chiquvchilar;

• skripka () - quti va mo'ylovga o'xshash diagramma tuzadi

yadro zichligini taxmin qilish;

205

|  |
| --- |
| **207-bet** |

• boxenplot () - to'rtburchaklardan diagramma tuzadi, yaxshi

katta ma'lumotlar to'plamlarini tasavvur qilish uchun mos.

Grafiklarning ko'rinishini sozlash uchun parametrlardan foydalaning

bobning boshida keltirilgan. Bunga qo'shimcha ravishda, funktsiyalar mavjud

umumiy argumentlar to'plami va individual sozlash elementlari.

Funktsiyalar uchun qo'shimcha parametrlarning umumiy to'plamini ko'rib chiqing

toifadagi ma'lumotlarning taqsimlanishini ingl.

• to'yinganlik: suzuvchi, ixtiyoriy

◦ Rangning to'yinganligi nisbati.

• kenglik: suzuvchi, ixtiyoriy

◦ Diagramma elementining kengligi (masalan, mo'ylovli quti).

• dodge: bool, ixtiyoriy

◦ Agar parametr True bo'lsa va rang parametridan foydalanilsa, u holda

jadval, ma'lumotlar ingl

ajratiladigan guruhlar.

• chiziq kengligi: suzuvchi, ixtiyoriy

◦ Elementlar chegarasining kengligi.

**9.3.1 Boxplot () funktsiyasi**

Dastlab, qanday ma'lumotlarni olishimiz mumkinligini aniqlaylik.

*quti* chizig'idan . Uning ko'rinishi 9.21-rasmda ko'rsatilgan. Ustida

diagrammada median (2-kvartil), 1 va 3-kvartillar ko'rsatilgan;

interkartil oralig'ini tashkil etuvchi; agar siz uni qo'ysangiz va

bir yarim interkartil intervalgacha, so'ngra oxirgi qiymatlar

ushbu oraliqdagi belgilar mo'ylov chegaralarini, tashqaridagi qiymatlarni tashkil qiladi

ushbu chegaralar haddan tashqari chegaralar sifatida belgilanadi, ular ko'rsatiladi

alohida fikrlar.

206

|  |
| --- |
| **Sahifa 208** |

**9.21-rasm - *Boxplot* diagrammasi**

Barcha ma'lumotlarni vizualizatsiya qilish funktsiyalari uchun umumiy parametrlardan tashqari

*seaborn* ( *8.1-ning umumiy funktsional parametrlariga* qarang ) va uchun vositalar

kategorik ma'lumotlar bilan ishlash, boxplot () funktsiyasi bir qatorga ega

noyob parametrlar:

• fliersize: suzuvchi, ixtiyoriy

◦ Chegaralarni ko'rsatish uchun ishlatiladigan markerlarning kattaligi.

• whis: suzuvchi, ixtiyoriy

◦ Mo'ylov uzunligini aniqlaydigan interkartil oralig'ining qiymati

grafikalar.

*Mpg* ma'lumotlar to'plami , displey uchun diagramma *tuzamiz*

mashina bosib o'tgan millar sonining taqsimlanishi

AQSh va Evropa uchun bir litr yoqilg'i:

mpg = sns.load\_dataset ( **"mpg"** )

mpg\_mod = mpg [mpg [ **"kelib chiqishi"** ]! = **"yaponiya"** ]

sns.boxplot (x = **"kelib chiqishi"** , y = **"mpg"** , ma'lumotlar = mpg\_mod)

207

|  |
| --- |
| **Sahifa 209** |

**9.22-rasm - boxplot () funktsiyasini namoyish etish**

Rangni bir xil rangga solamiz:

sns.boxplot (x = **"kelib chiqishi"** , y = **"mpg"** , color = **'g'** , ma'lumotlar = mpg\_mod)

**Shakl 9.23 - Yashil plombali *quti* diagrammasi**

Rangning to'yinganligini to'yinganlik parametri orqali o'zgartirish mumkin

sukut bo'yicha 0,75, ya'ni. rasm ko'rsatiladi

kam to'yinganlik, qoida tariqasida, bunday echim ko'proq narsani beradi

rangning ingl. Keling, qanday ko'rinishini taqqoslaylik

to'yinganlik parametrining turli qiymatlari uchun diagramma:

208

|  |
| --- |
| **Sahifa 210** |

plt.figure ( figsize = ( 15 , 5 ))

sat\_list = [ 1 , 0.75 , 0.5 , 0.25 ]

uchun i, p yilda Enumerate (sat\_list):

plt.subplot ( 1 , len (sat\_list), i + 1 )

sns.boxplot ( x = **"kelib chiqishi"** , y = **"mpg"** , to'yinganlik = s, ma'lumotlar = mpg\_mod)

**9.24-rasm - to'yinganlik parametri bilan ishlashni namoyish etish**

Vizual dizaynni o'zgartirish uchun yana bir vosita

diagramma - bu chiziqning kengligi parametri, bu orqali qalinligi o'rnatiladi

chiziqlar "mo'ylovli quti":

plt.figure ( figsize = ( 15 , 5 ))

lw\_list = [ 6 , 3 , 1 ]

uchun I, LW yilda enumerate (lw\_list):

plt.subplot ( 1 , len (lw\_list), i + 1 )

sns.boxplot ( x = **"kelib chiqishi"** , y = **"mpg"** , chiziq kengligi = lw, ma'lumotlar = mpg\_mod)

**9.25-rasm - chiziq kengligi parametri bilan ishlashni namoyish etish**

209

|  |
| --- |
| **211-bet** |

Miqdor bo'yicha qo'shimcha rang ajratishni qo'shing

rang parametridan foydalangan holda avtomobil dvigatelidagi silindrlar:

sns.boxplot (x = **"kelib chiqishi"** , y = **"mpg"** , hue = **"silindrlar"** , ma'lumotlar = mpg\_mod)

**9.26-rasm - rang parametrlari bilan ishlashni namoyish etish**

Rang sxemasini palitra parametri orqali o'rnatamiz:

sns.boxplot ( x = **"kelib chiqishi"** , y = **"mpg"** , hue = **"silindrlar"** , palitrasi = **'Pastel1'** ,

ma'lumotlar = mpg\_mod)

**9.27-rasm - palitrasi parametri bilan ishlashni namoyish etish**

210

|  |
| --- |
| **212-bet** |

Asosiy xarakteristikaning qiymatlarini ko'rsatish tartibi quyidagicha belgilanadi

hue\_order orqali rangni qo'llagan holda buyurtma parametri. Yo'nalish

barcha *dengiz sathidagi* funktsiyalar singari diagrammalar tomonidan boshqariladi

orientatsiya:

sns.boxplot ( x = **"mpg"** , y = **"kelib chiqish"** , orient = **'h'** , ma'lumotlar = mpg\_mod)

**9.28-rasm - orientatsiya parametri bilan ishlashni namoyish etish**

"Mo'ylovli quti" ning kengligi (yoki balandligi, agar yo'nalishi = 'h') tomonidan belgilanadi

parametr kengligi. Agar qiymat 1 ga teng bo'lsa, kataklar barchasini egallaydi

berilgan joy, masalan, funktsiya egallasa

faqat ikkita qiymat, keyin butun grafik maydoni ikkiga bo'linadi

kattalashgan yoki kamaygan qutilar qutining o'lchamini o'zgartiradi:

plt.figure ( figsize = ( 15 , 5 ))

w\_list = [ 1.25 , 1 , 0.75 , 0.5 ]

uchun w i, ham enumerate (w\_list):

plt.subplot ( 1 , len (w\_list), i + 1 )

sns.boxplot ( x = **"kelib chiqishi"** , y = **"mpg"** , kengligi = w, ma'lumotlar = mpg\_mod)

211

|  |
| --- |
| **213-bet** |

**9.29-rasm - kenglik parametri bilan ishlashni namoyish etish**

Parametr chegaralarni belgilaydigan markerlarning kattaligi uchun javobgardir.

varaqlash:

plt.figure ( figsize = ( 10 , 5 ))

plt.subplot ( 121 )

sns.boxplot ( x = **"kelib chiqishi"** , y = **"mpg"** , ma'lumotlar = mpg\_mod)

plt.subplot ( 122 )

sns.boxplot ( x = **"kelib chiqishi"** , y = **"mpg"** , fliersize = 10 , ma'lumotlar = mpg\_mod)

**9.30-rasm - Flyersize parametri bilan ishlashni namoyish etish**

212

|  |
| --- |
| **214-bet** |

Ushbu bobning boshida aytib o'tilganidek, qutining vertikal kattaligi (uning

balandlik) 1 va 3-chi kvartillar orasidagi intervalga teng, bir yarim shunday

interval mo'ylovning uzunligini, ma'lumotlar to'plamidagi barcha qiymatlarni belgilaydi,

ushbu intervaldan kattaroq kattaroq ko'rsatkich sifatida belgilanadi. Orqasida

interkartalar oralig'ining qiymati whis parametriga mos keladi, tekshirib ko'raylik

sukut bo'yicha bu qiymat 1,5 ga teng:

plt.figure ( figsize = ( 10 , 5 ))

plt.subplot ( 121 )

sns.boxplot ( x = **"kelib chiqishi"** , y = **"mpg"** , ma'lumotlar = mpg\_mod)

plt.subplot ( 122 )

sns.boxplot ( x = **"kelib"** , y = **"MPG"** , whis = 1,5 , ma'lumotlar = mpg\_mod)

**9.31-rasm - whis = 1.5 parametri bilan ishlashni namoyish etish**

213

|  |
| --- |
| **215-bet** |

Endi ushbu qiymatlarni o'zgartirib, diapazonini tuzamiz

ikkiga teng:

sns.boxplot (x = **"kelib chiqishi"** , y = **"mpg"** , whis = 2 , ma'lumotlar = mpg\_mod)

**9.32-rasm - whis = 2 parametri bilan ishlashni namoyish etish**

**9.3.2 skripka () funktsiyasi**

Kategorik ma'lumotlarni ko'rish uchun keyingi funktsiya,

biz ko'rib chiqadigan skripka () bo'ladi. Funktsional jihatdan

maqsadi va imkoniyatlari, u ilgari muhokama qilinganga o'xshaydi

boxplot (), qo'shimcha ravishda yadro zichligini baholaydi.

Biz tomonidan ishlatilgan ma'lumotlar to'plami bilan ishlaymiz

boxplot () ni tekshirishda:

mpg = sns.load\_dataset ( **"mpg"** )

*Keling* fitna *boxplot* va *Violine diagrammalar* uchun bir-biriga Keyingi

taqqoslashlar.

mpg\_mod = mpg [mpg [ **"kelib chiqishi"** ]! = **"yaponiya"** ]

plt.figure (figsize = ( 10 , 5 ))

plt.subplot ( 121 )

sns.boxplot (x = **"kelib chiqishi"** , y = **"mpg"** , ma'lumotlar = mpg\_mod)

214

|  |
| --- |
| **216-bet** |

plt.subplot ( 122 )

sns.violinplot (x = **"kelib chiqishi"** , y = **"mpg"** , ma'lumotlar = mpg\_mod)

**9.33-rasm - *Boxlot* va *skripka chizmalarini* taqqoslash**

Rang sxemasi bilan ishlash uchun bizga tanish bo'lgan parametrlarga qo'shimcha ravishda va

violineplot () bir qator noyob dalillarga ega,

ulardan ba'zilari quyida keltirilgan:

• bw: {'scott', 'silverman', float}, ixtiyoriy

◦ yadro *o'tkazuvchanligi* .

• o'lchov: {'maydon', 'hisoblash', 'kenglik'}, ixtiyoriy

The Diagramma kengligini masshtablash usuli.

• split: bool, ixtiyoriy

◦ Agar ma'lumotlar to'plamini rang parametrlari orqali ajratish

ikki guruhga bo'linadi, keyin split = True bilan bo'ladi

diagrammaning faqat yarmi ko'rsatiladi.

215

|  |
| --- |
| **217-bet** |

*Ranglar jadvali*

Rang parametri orqali bir xil rang oralig'i ko'rsatilgan:

sns.violinplot (x = **"kelib chiqishi"** , y = **"mpg"** , rang = **"sariq"** , ma'lumotlar = mpg\_mod)

**9.34-rasm - funktsiyaning rang parametri bilan ishlashni namoyish etish**

**skripka ()**

AQSh va *mpg* ma'lumotlar to'plamidan tanlang

4 yoki 6 tsilindrli Yaponiya:

fn\_filter = lambda x: To'g'ri bo'lsa, agar x in [ 4 , 6 ] bo'lsa False

fn\_mod = lambda x: { 4 : **'to'rt'** , 6 : **'olti'** , 8 : **'sakkiz'** } [x]

mpg\_country = mpg [mpg [ **"kelib chiqishi"** ]! = **"** evropa **"** ]

mpg\_demo = mpg\_country [mpg\_country [ **'silindrlar'** ] .map (fn\_filter)]. nusxa ko'chirish ()

mpg\_demo [ **'silindrlar'** ] = mpg\_demo [ **'silindrlar'** ] .map (fn\_mod)

Qo'shimcha rang ajratish bilan diagramma tuzamiz

tsilindr soni:

sns.violinplot ( x = **"kelib chiqishi"** , y = **"mpg"** , hue = **"silindrlar"** , ma'lumotlar = mpg\_demo)

216

|  |
| --- |
| **218-bet** |

**9.35-rasm - rang funktsiyasi parametri bilan ishlashni namoyish etish**

**skripka ()**

Shilinglar uchun rang sxemasini va ranglarni belgilash tartibini *aniqlaylik* :

color\_scheme = { **"to'rt"** : **"y"** , **"olti"** : **"binafsha"** }

color\_order = [ **"olti"** , **"to'rt"** ]

sns.violinplot (x = **"kelib chiqishi"** , y = **"mpg"** , hue = **"silindrlar"** ,

hue\_order = color\_order, palitrasi = color\_scheme, ma'lumotlar = mpg\_demo)

**9.36-rasm - hue\_order va parametrlari bilan ishlashni namoyish etish**

**palitra skripka () funktsiyalari**

217

|  |
| --- |
| **219-bet** |

Rangning to'yinganligi to'yinganlik parametri orqali o'rnatiladi:

plt.figure ( figsize = ( 15 , 5 ))

s\_list = [ 0,25 , 0,5 , 0,75 , 1 ]

uchun i, p yilda Enumerate (s\_list):

plt.subplot ( 1 , len (s\_list), i + 1 )

sns.violinplot ( x = **"kelib chiqishi"** , y = **"mpg"** , hue = **"silindrlar"** , to'yinganlik = s,

ma'lumotlar = mpg\_demo)

**9.37-rasm - to'yinganlik parametri bilan ishlashni namoyish etish**

**violinplot () funktsiyalari**

Ushbu diagrammalar nosimmetrik shaklga ega ekanligini unutmang; shuning uchun

tarqatish mohiyatini baholash uchun bu etarli

yarmi, siz ularni birlashtira olasiz - tarqatishdan chap qismini oling

silindrlarning soni oltitaga, o'ng - raqam bilan

to'rtga teng:

sns.violinplot (x = **"kelib chiqishi"** , y = **"mpg"** , hue = **"silindrlar"** ,

hue\_order = color\_order, palitrasi = color\_scheme, split = True , ma'lumotlar = mpg\_demo)

218

|  |
| --- |
| **Sahifa 220** |

**9.38-rasm - split funktsiya parametri bilan ishlashni namoyish etish**

**skripka ()**

Diagramma konturining chiziq kengligi linewidth orqali o'rnatiladi **:**

plt.figure ( figsize = ( 15 , 5 ))

lw\_list = [ 6 , 3 , 1 ]

uchun I, LW yilda enumerate (lw\_list):

plt.subplot ( 1 , len (lw\_list), i + 1 )

sns.violinplot ( x = **"kelib chiqishi"** , y = **"mpg"** , chiziq kengligi = lw, ma'lumotlar = mpg\_demo)

**9.39-rasm - chiziq kengligi parametri bilan ishlashni namoyish etish**

**violinplot () funktsiyalari**

219

|  |
| --- |
| **Sahifa 221** |

Yadro kengligi bw parametri orqali o'rnatiladi:

bw\_values ​​= [ **"** scott **"** , **"silverman"** , 0.25 , 1 ]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, B ham Enumerate (bw\_values):

plt.subplot ( 1 , len (bw\_values), i + 1 )

plt.title ( str (b))

sns.violinplot ( x = **"kelib chiqishi"** , y = **"mpg"** , bw = b, ma'lumotlar = mpg\_demo)

**9.40-rasm - bw funktsiyasi parametri bilan ishlashni namoyish etish**

**skripka ()**

Diagrammaning ko'lamini shkala bo'yicha belgilash mumkin, bu parametr

uchta qiymatdan birini oladi: {'maydon', 'hisoblash', 'kenglik'}:

scale\_values ​​= [ **"maydon"** , **"hisoblash"** , **"kenglik"** ]

plt.figure ( figsize = ( 15 , 5 ))

uchun i, p yilda Enumerate (scale\_values):

plt.subplot ( 1 , len (scale\_values), i + 1 )

plt.title ( str (lar))

sns.violinplot ( x = **"kelib chiqishi"** , y = **"mpg"** , shkalasi = s, ma'lumotlar = mpg\_demo)

220

|  |
| --- |
| **Sahifa 222** |

**9.41-rasm - funktsiyaning shkala parametri bilan ishlashni namoyish etish**

**skripka ()**

Buyurtma va yo'nalish parametrlari buyurtma va yo'nalish uchun javobgardir:

sns.violinplot (x = "mpg" , y = "kelib chiqish" , orient = 'h' ,

order = [ "usa" , " Japan " ], data = mpg\_demo)

**Rasm 9.42 - parametrlarni bilan ishlash namoyish** tartibi va

yo'naltirilgan skripka () **funktsiyalari**

**9.4 Ma'lumotlarning toifadagi natijalarini ingl**

Ushbu guruhga pointplot (), barplot () va funktsiyalari kiradi

countplot (), ularning maqsadi har xil qilishdir

ma'lumotlar to'plamlarining umumlashtirilgan xususiyatlari (biz ularni chaqiramiz

taxminlar), masalan, ma'lum bir narsaga tegishli narsalar soni

221

|  |
| --- |
| **Sahifa 223** |

guruh, standart og'ish va boshqalar. Ularning parametr guruhlari

funktsiyalar qisman bir-biriga to'g'ri keladi, ammo biz bunday qilmaymiz

alohida ro'yxat sifatida ko'rib chiqildi.

**9.4.1 Pointplot () funktsiyasi**

Pointplot () funktsiyasi ma'lumotlar to'plamini baholashni aks ettiradi

grafik chegaradagi nuqta va ishonch oralig'i chiziq sifatida,

uning markazi belgilangan nuqtada yotadi.

Namoyish amalga oshiriladi *nuqta* ma'lumotlar :

nuqta = sns.load\_dataset ( **"nuqta"** )

Kategorik xususiyat sifatida nuqta diagrammasi tuzamiz

*hizalamayı* tanlang , taxmin *firing\_rate* tomonidan *hisoblanadi* :

sns.pointplot (x = **'hizalamak'** , y = **'otashin\_rat'** , ma'lumotlar = nuqtalar)

***9.43-* rasm - *Pointplot***

222

|  |
| --- |
| **Sahifa 224** |

As siz arbobi ko'rishingiz mumkin: o'rtacha *firing\_rate* uchun

*align = nuqta* taxminan 40.25, *align = sacc* uchun 39.1.

Keling, ushbu qiymatlarning aniq qiymatlarini olaylik:

nuqta [nuqta [ **"align"** ] == **'nuqta'** ] [ **'firing\_rate'** ] **.mean** ()

40.23124948122005

nuqtalar [dots [ **"align"** ] == **'sacc'** ] [ **'firing\_rate'** ] **.mean** ()

39.083297066051394

Vertikal chiziqlar 95% ishonch oralig'ini anglatadi.

Yana bir ajratish darajasini qo'shaylik - *tanlov* asosida , tanlang

u rang parametridan foydalangan holda:

sns.pointplot (x = **'hizalamak'** , y = **'firing\_rate'** , hue = **'choice'** , data = nuqta)

**9.44-rasm - foydalanib ma'lumotlarni ajratishni namoyish etish**

**pointplot () funktsiyasining rang parametri**

223

|  |
| --- |
| **Sahifa 225** |

Ranglar palitrasini o'zgartiramiz:

sns.pointplot ( x = **'align'** , y = **'firing\_rate'** , hue = **'choice'** , palitrasi = **'GnBu'** ,

ma'lumotlar = nuqta)

**Shakl 9.45 - bilan rang palitrasini o'zgartirishni namoyish etish**

**pointplot () funktsiyasining palitrasi parametridan foydalangan holda**

Bashorat qiluvchi parametr, algoritm uchun javobgardir, orqali

bu vektorni qabul qiladigan statistikani hisoblash funktsiyasi va

skaler qiymatini qaytarish. Keling, boshqacha diagrammalar tuzamiz

ballni aniqlash usuli:

dan numpy import o'rtacha, Media, min, Maks

taxminchi = [o'rtacha, o'rtacha, min, maksimal]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, As -yilda Enumerate (kiritish):

plt.subplot ( 1 , len (taxminchi), i + 1 )

plt.title (es. \_\_name\_\_ )

sns.pointplot ( x = **'align'** , y = **'firing\_rate'** , taxminchi = es, ma'lumotlar = nuqta)

224

|  |
| --- |
| **Sahifa 226** |

**9.46-rasm - taxminiy parametr bilan ishlashni namoyish etish**

**nuqtalari () funktsiyalari**

Ci parametri ishonch oralig'ini o'rnatish uchun ishlatiladi (qarang.

" *8.1.2 Axborot tarkibini oshirish parametrlari* " bo'limiga qarang

*grafikalar* "). Qiymatni aniq belgilaydigan raqamli qiymatdan tashqari

ishonch oralig'i, agar kerak bo'lsa, unga "sd" ni topshirishingiz mumkin

standart og'ishni aks ettiring yoki bu holda Yo'q

vertikal chiziqlar ko'rsatilmaydi:

cis = [ Yo'q , 95 , **'sd'** ]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, c yilda enumerate (MDH):

plt.subplot ( 1 , len (cis), i + 1 )

plt.title ( **f "ci =** { c } **"** )

sns.pointplot ( x = **'hizalamak'** , y = **'otashin\_rat'** , ci = c, ma'lumotlar = nuqtalar)

**9.47-rasm - funktsiya ci parametri bilan ishlashni namoyish etish**

**nuqta ()**

225

|  |
| --- |
| **Sahifa 227** |

Parametrlar orqali markerlar va ulanish chizig'ining uslubi o'rnatiladi

markerlar va linestyle:

• markerlar: satr yoki qatorlar ro'yxati, ixtiyoriy

Each Har bir qiymat uchun ishlatiladigan markerlar

xususiyat tusdan o'tib ketdi.

• uslublar: satr yoki satrlar ro'yxati, ixtiyoriy

Values ​​Qadriyatlar uchun ishlatiladigan chiziq uslublari

xususiyat tusdan o'tib ketdi.

ms = [ **"s"** , **"^"** ]

ls = [ **"-"** , **"-".**]

sns.pointplot ( x = **'align'** , y = **'firing\_rate'** , hue = **'choice'** , markers = ms,

linestyles = ls, data = nuqta)

**Shakl 9.48 - parametrlar bilan ishlashni namoyish qilish va**

**linestyles pointplot () funktsiyalari**

Bundan tashqari, jadval ko'rinishini sozlash uchun siz qilishingiz mumkin

pointplot () funktsiyasining quyidagi parametrlari foydalidir:

• o'lchov: suzuvchi, ixtiyoriy

◦ Diagramma elementlari hajmini belgilaydigan multiplikator.

226

|  |
| --- |
| **Sahifa 228** |

• xatolik: suzuvchi, ixtiyoriy

Confidence Ishonch oraliqlarini ifodalovchi chiziqli og'irliklar va

ularning uzoq elementlari.

• kapsula: suzuvchi, ixtiyoriy

Represent ifodalaydigan chiziqlar avlodlari kengligi

ishonch oralig'i

Xatolik, to'siq parametrlarini ishlatish misoli:

xatolar = [ 1 , 2 , 4 ]

kapsulalar = [ 0 , 0.5 , 1 ]

plt.figure ( figsize = ( 15 , 5 ))

uchun i, p yilda enumerate ( zip (errwidthes, alabora)):

plt.subplot ( 1 , len (xatolar), i + 1 )

plt.title ( **f "errwidth =** { p [ 0 ] } **, kapsula =** { p [ 1 ] } **"** )

sns.pointplot ( x = **'hizalamak'** , y = **'otashin\_rat'** , xato = p [ 0 ],

kapsula = p [ 1 ], ma'lumotlar = nuqta)

**9.49-rasm - xato va parametrlari bilan ishlashni namoyish etish**

**pointplot () funktsiyasining ag'darilishi**

227

|  |
| --- |
| **Sahifa 229** |

O'lchov parametri bilan ishlashni namoyish etish:

tarozi = [ 0,5 , 1 , 2 ]

plt.figure ( figsize = ( 15 , 5 ))

uchun i, p yilda Enumerate (tarozi):

plt.subplot ( 1 , len (tarozi), i + 1 )

plt.title ( **f "masshtab =** { s } **"** )

sns.pointplot ( x = **'hizalamak'** , y = **'otashin\_rat'** , shkala = s, ma'lumotlar = nuqta)

**9.50-rasm - shkala parametri bilan ishlashni namoyish etish**

**nuqtalari () funktsiyalari**

**9.4.2 Barplot () funktsiyasi**

Barplot () uchastkalari jadvalini: satr balandligi

(ustun) xarakteristik balning raqamli qiymatini belgilaydi

(matematik kutish, o'rtacha va boshqalar), chiziqni kesib o'tish

satrining yuqori chegarasi ishonch oralig'i.

Funktsiya imkoniyatlarini o'rganish uchun *mpg* ma'lumotlar to'plamini yuklang

barplot ():

mpg = sns.load\_dataset ( **"mpg"** )

228

|  |
| --- |
| **Sahifa 230** |

Bunga qarab avtoulovlarning kuchini baholash uchun diagramma tuzamiz

ishlab chiqaruvchi mamlakat:

sns.barplot (x = **'kelib chiqishi'** , y = **'ot kuchi'** , ma'lumotlar = mpg)

**9.51-rasm -** Barplot **diagrammasi**

Ishonch oralig'i quyuq kulrang chiziqlar sifatida ko'rsatilgan

har bir satrning yuqori qismidagi ranglar. Hamma uchun bitta rangni sozlash uchun

satrlari, rang parametridan foydalanishingiz mumkin:

sns.barplot ( x = **'kelib chiqishi'** , y = **'ot kuchi'** , rang = **'to'q sariq'** , ma'lumotlar = mpg)

**9.51-rasm - rang parametri bilan ishlashni namoyish etish**

**barplot () funktsiyalari**

229

|  |
| --- |
| **231-bet** |

Hue argumenti yordamida ma'lumotlarni ajratishning yana bir darajasi aniqlanishi mumkin,

biz bundan raqam bo'yicha qo'shimcha segmentatsiya qilish uchun foydalanamiz

avtomobil dvigatelidagi shilinglar:

sns.barplot ( x = **'kelib chiqishi'** , y = **'ot kuchi'** , rang = **'silindrlar'** , ma'lumotlar = mpg)

**9.52-rasm - rang funktsiyasi parametri bilan ishlashni namoyish etish**

**barplot ()**

Ranglar palitrasini o'zgartiramiz:

sns.barplot ( x = **'kelib chiqishi'** , y = **'ot kuchi'** , hue = **'silindrlar'** , palitrasi = **'Set2'** ,

ma'lumotlar = mpg)

**9.53-rasm - Palitra parametri bilan ishlashni namoyish etish**

**barplot () funktsiyalari**

230

|  |
| --- |
| **232-bet** |

Kategoriya atributining *kelib chiqish* qiymatlarini ko'rsatish tartibini belgilaylik :

order = [ **'evropa'** , **'AQSh'** , **'yaponiya'** ]

sns.barplot (x = **'kelib chiqishi'** , y = **'ot kuchi'** , hue = **'silindrlar'** , palitrasi = **'Set2'** ,

buyurtma = buyurtma, ma'lumotlar = mpg)

**9.54-rasm - Buyurtma parametri bilan ishlashni namoyish etish**

**barplot () funktsiyalari**

Ballarni hisoblashning turli usullari bilan diagrammalar tuzamiz:

taxminchi = [o'rtacha, o'rtacha, min , maksimal ]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, As -yilda Enumerate (kiritish):

plt.subplot ( 1 , len (taxminchi), i + 1 )

plt.title (es. \_\_name\_\_ )

sns.barplot ( x = **'kelib chiqishi'** , y = **'ot kuchi'** , taxminchi = es, ma'lumotlar = mpg)

231

|  |
| --- |
| **233-bet** |

**9.55-rasm - taxminiy parametr bilan ishlashni namoyish etish**

**barplot () funktsiyalari**

Ishonch oralig'ini belgilash (yoki standart og'ish)

ci parametri orqali amalga oshiriladi:

cis = [ Yo'q , 80 , **'sd'** ]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, c yilda enumerate (MDH):

plt.subplot ( 1 , len (cis), i + 1 )

plt.title ( **f "ci =** { c } **"** )

sns.barplot ( x = **'kelib chiqishi'** , y = **'ot kuchi'** , ci = c, ma'lumotlar = mpg)

**9.56-rasm - funktsiya ci parametri bilan ishlashni namoyish etish**

**barplot ()**

232

|  |
| --- |
| **234-bet** |

Xatolik, to'siq parametrlari (" *9.4.1* Pointplot () *funktsiyasi* " ga qarang) va

errcolor satr xususiyatlarini o'rnatish uchun javobgardir. Birinchi ikkitasi bilan biz

allaqachon tanish, keling, errcolorni aniqlaymiz:

• errcolor: *Matplotlib* -color

◦ ishonch interval liniyasi rangi.

Quyida ushbu parametrlar bilan ishlashning namunasi keltirilgan:

errcolor = [ **'r'** , **'b'** , **'y'** ]

xatolar = [ 2 , 3 , 4 ]

kapsulalar = [ 0 , 0.3 , 0.7 ]

plt.figure ( figsize = ( 15 , 5 ))

uchun i, p yilda enumerate ( zip (errcolor, errwidthes, alabora)):

plt.subplot ( 1 , len (errcolor), i + 1 )

plt.title ( **f "errcolor =** { p [ 0 ] } **, errwidth =** { p [ 1 ] } **, burilish =** { p [ 2 ] } **"** )

sns.barplot ( x = **'kelib chiqishi'** , y = **'ot kuchi'** , errcolor = p [ 0 ], errwidth = p [ 1 ],

kapsula = p [ 2 ], ma'lumotlar = mpg)

**9.57-rasm - xatolik parametrlari bilan ishlashni namoyish etish,**

**barplot () funktsiyasining xiralashishi va xatoligi**

**9.4.3 countplot () funktsiyasi**

Countplot () funktsiyasi to'plamdagi elementlar sonini aniqlaydi

ma'lum bir toifaga tegishli ma'lumotlar va displeylar

natijada chiziqli jadval ko'rinishidagi qiymat.

233

|  |
| --- |
| **Sahifa 235** |

Countplot () bilan ishlashni namoyish qilish uchun biz to'plamdan foydalanamiz

ushbu *maslahatlar* :

maslahatlar = sns.load\_dataset ( **"maslahatlar"** )

*X* o'qi uchun xususiyat sifatida tarqatish diagrammasini tuzamiz

belgilash *kun* :

sns.countplot (x = **"kun"** , ma'lumotlar = maslahatlar)

**9.58-rasm - Countplot () diagrammasi**

Keling, barcha baralar uchun yashil rangni o'rnatamiz:

sns.countplot ( x = **"kun"** , color = **"g"** , ma'lumotlar = maslahatlar)

**9.59-rasm - rang parametri bilan ishlashni namoyish etish**

**countplot () funktsiyalari**

234

|  |
| --- |
| **236-bet** |

Keling, yana bitta parametrni - odamning jinsini tanishtiramiz:

sns.countplot ( x = **'kun'** , hue = **'jinsiy aloqa'** , ma'lumotlar = maslahatlar)

**9.60-rasm - rang funksiyasi parametri bilan ishlashni namoyish etish**

**countplot ()**

Ranglar palitrasini o'zgartiramiz:

sns.countplot ( x = **'kun'** , hue = **'jinsiy aloqa'** , palitra = **"PuOr"** , ma'lumotlar = maslahatlar)

**9.61-rasm - Palitra parametri bilan ishlashni namoyish etish**

**countplot () funktsiyalari**

235

|  |
| --- |
| **237-bet** |

Countplot () funktsiyasi to'yinganlik parametriga ega, u javob beradi

tanlangan ranglar diapazonining to'yinganligi:

plt.figure ( figsize = ( 15 , 5 ))

s\_list = [ 0,25 , 0,5 , 0,75 , 1 ]

uchun i, p yilda Enumerate (s\_list):

plt.subplot ( 1 , len (s\_list), i + 1 )

plt.title ( **f "to'yinganlik =** { s } **"** )

sns.countplot ( x = **'kun'** , hue = **'jins'** , to'yinganlik = s, palitrasi = **"Set1"** ,

ma'lumotlar = maslahatlar)

**9.62-rasm - to'yinganlik parametri bilan ishlashni namoyish etish**

**countplot () funktsiyalari**

**9.5 Shakl darajasida ishlash. Catplot () funktsiyasi**

Funktsional va maqsadga muvofiq, catplot () funktsiyasi

vizualizatsiya asboblar qutisidan relplot () funktsiyasiga o'xshash

ma'lumotlardagi aloqalar (" *8.5* bo'limiga qarang. *bilan munosabatlarni ingl*

*pastki qavatni o'rnatish.*Relplot () *funktsiyasi "* ). Uning o'ziga xos xususiyati

bu hamma uchun umumiy interfeysni taqdim etishidir

Kategorik ma'lumotlarni vizualizatsiya qilish guruhining funktsiyalari, u ham

boshqaradigan shakl darajasi parametrlari bilan ishlashga imkon beradi

236

|  |
| --- |
| **238-bet** |

diagrammalarni joylashtirish. Bu holda ko'rsatkich darajasi aniqlanadi

FacetGrid sinfining ob'ekti, u orqali u to'g'ridan-to'g'ri o'tadi

joylashtirishni boshqarish.

Keling, ushbu vositani yaxshiroq oldindan bilib olaylik

uslub va kontekst:

sns.set\_style ( **"whitegrid"** )

sns.set\_context ( **"daftar"** )

Keling yuk *Maslahatlar* DataSet tajribalar uchun:

maslahatlar = sns.load\_dataset ( **"maslahatlar"** )

Kategorik qiymatlarni taqsimlashning tarqalish diagrammasini tuzamiz

o'zgaruvchan *kun* :

sns.catplot ( x = **'kun'** , y = **'total\_bill'** , turdagi = **'chiziq'** , ma'lumotlar = maslahatlar)

**9.63-rasm - catplot () funktsiyasini namoyish etish**

237

|  |
| --- |
| **239-bet** |

Rasmdan ko'rinib turibdiki, biz shunga o'xshash diagrammani oldik

stripplot () tomonidan ishlab chiqarilgan. Catplot () parametri bilan,

bu orqali diagramma turi ko'rsatilgan:

• kind: str, ixtiyoriy

Art Diagramma turi. 9.5-jadvalda qiymatning mosligi ko'rsatilgan

uchun mos parametr va asboblar qutisi funktsiyasi

kategorik ma'lumotlarning ingl. Standart qiymat

*"Ip"* .

**Jadval 9.4 - parametr turi va funktsiyalari qiymatlarining mosligi**

***dengiz dengizi***

**Parametr qiymati**

***Seaborn* funktsiyasi**

*Ip*

stripplot ()

*to'da*

to'da uchastkasi ()

*quti*

boxplot ()

*skripka*

skripka ()

*boksen*

boxenplot ()

*nuqta*

nuqta ()

*bar*

barplot ()

*hisoblash*

countplot ()

X, y, data, hue, taxminiy, ci, order, parametrlarini tayinlash

tartib, rang, orientatsiya, rang va palitrasi shu nomdagi ranglarga o'xshash

ilgari muhokama qilingan toifali ma'lumotlarni vizualizatsiya qilish funktsiyalari

Biz ular haqida to'xtamaymiz. To'g'ridan-to'g'ri boramiz

238

|  |
| --- |
| **Sahifa 240** |

joylashuvni boshqarishimiz mumkin bo'lgan argumentlar

diagrammalar. Kol va qator bilan boshlaymiz:

• qator, kol: ma'lumotlar to'plamidagi o'zgaruvchan nomlar, ixtiyoriy

◦ kategorik belgilar

qatorlarga va ustunlarga taqsimlash.

Qolgan maslahatlar sonini taqsimlash sxemalarini tuzamiz

haftaning kuni, ustunlarga bo'lingan:

sns.catplot (x = 'kun' , y = 'total\_bill' , col = 'sex' , kind = 'strip' , ma'lumotlar = maslahatlar)

**Rasm 9.64 - bilan ishlash namoyish funktsiyasi** Polkovnik **parametr**

catplot ()

Keling, bu misolni o'zgartiramiz, shunday qilib bo'linish satrma-satr amalga oshiriladi:

sns.catplot ( x = **'kun'** , y = **'total\_bill'** , satr = **'jinsiy aloqa'** , turdagi = **'chiziq'** , ma'lumotlar = maslahatlar)

239

|  |
| --- |
| **Sahifa 241** |

**9.65-rasm - funktsiya qatori parametri bilan ishlashni namoyish etish**

**catplot ()**

Diagramma o'lchamlari va tomonlarning nisbatlarini boshqarish uchun

balandlik va tomon parametrlaridan foydalaning:

• balandlik: raqam, ixtiyoriy

◦ Grafik balandligi. Dyuymlarda ko'rsatilgan.

240

|  |
| --- |
| **Sahifa 242** |

• tomon: raqam, ixtiyoriy

Asp tomonlarning nisbati, kengligini ko'rsatuvchi omil

diagramma quyidagicha hisoblanadi: tomon \* balandlik

Boylik va bilan ishlashni ko'rsatadigan ba'zi bir misollar

jihati:

sns.catplot ( x = **'kun'** , y = **'total\_bill'** , col = **'jins'** , balandlik = 5 , aspekt = 0,5 ,

kind = **'strip'** , ma'lumotlar = maslahatlar)

**9.66-rasm - balandlik va parametrlari bilan ishlashni namoyish etish**

**catplot () funktsiyasining tomoni (1-misol)**

sns.catplot ( x = **'kun'** , y = **'total\_bill'** , col = **'jins'** , balandlik = 5 , aspekt = 1,5 ,

kind = **'strip'** , ma'lumotlar = maslahatlar)

241

|  |
| --- |
| **Sahifa 243** |

**9.67-rasm - balandlik va parametrlari bilan ishlashni namoyish etish**

**catplot () funktsiyasining tomoni (1-misol)**

Legend joylashuvi afsona va orqali boshqariladi

afsonaviy:

• afsona: bool, ixtiyoriy

Paramet Agar parametr True in bo'lsa, afsona ko'rsatiladi

aks holda, bu diagrammada ko'rinmaydi.

• legend\_out: bool, ixtiyoriy

◦ Afsonaning joylashishi: agar parametr rost bo'lsa, u holda afsona

diagrammalardan tashqarida (rasmning o'ng tomonida), False bo'ladi

- to'g'ridan-to'g'ri diagrammalarning birida ko'rsatiladi.

Afsonani diagrammada joylashtiramiz:

sns.catplot ( x = **'kun'** , y = **'total\_bill'** , col = **'sex'** , h ue = **'chekuvchi'** ,

legend\_out = False , kind = **'strip'** , data = maslahatlar)

242

|  |
| --- |
| **Sahifa 244** |

**9.68-rasm - legend\_out parametri bilan ishlashni namoyish etish**

**catplot () funktsiyalari**

Afsonani rasmdan olib tashlaymiz:

sns.catplot ( x = **'kun'** , y = **'total\_bill'** , col = **'jinsiy aloqa'** , hue = **'chekuvchi'** ,

legend = False , kind = **'strip'** , data = tips)

**9.69-rasm - afsonaviy parametr bilan ishlashni namoyish etish**

**catplot () funktsiyalari**

Afsonani diagramma maydonining tashqarisiga qo'yamiz:

sns.catplot ( x = **'kun'** , y = **'total\_bill'** , col = **'jinsiy aloqa'** , hue = **'chekuvchi'** ,

kind = **'strip'** , ma'lumotlar = maslahatlar)

243

|  |
| --- |
| **Sahifa 245** |

**Shakl 9.70 - Afsonani maydon tashqarisiga qo'yishni namoyish etish**

**grafik san'at**

Margin\_titles = False bilan diagramma 9.71-rasmda ko'rsatilgan:

sns.catplot ( x = **'kun'** , y = **'total\_bill'** , col = **'jinsiy aloqa'** , satr = **'chekuvchi'** ,

margin\_titles = Yolg'on , balandlik = 3 , turdagi = **'chiziq'** , ma'lumotlar = maslahatlar)

**9.71-rasm - parametr bilan ishlashni namoyish etish**

**margin\_titles = catplot uchun noto'g'ri ()**

244

|  |
| --- |
| **Sahifa 246** |

9.71-rasmdagi diagrammani qaerdagi holat bilan solishtiring

margin\_titles = To'g'ri:

sns.catplot ( x = **'kun'** , y = **'total\_bill'** , col = **'jinsiy aloqa'** , satr = **'chekuvchi'** , balandlik = 3 ,

margin\_titles = To'g'ri , turdagi = **'chiziq'** , ma'lumotlar = maslahatlar)

**9.72-rasm - parametr bilan ishlashni namoyish etish**

**margin\_titles = catplot uchun to'g'ri ()**

Shaklni aniq sozlash uchun siz foydalanishingiz mumkin

facet\_kws parametri, konstruktor argumentlari u orqali o'tkaziladi

FacetGrid klassi, kalitlari nomlari bo'lgan lug'at sifatida

dalillar.

245

|  |
| --- |
| **Sahifa 247** |

**10-bob. In tarqatishlarni ingl**

**ma'lumotlar**

Ma'lumotlar bo'yicha taqsimotlarni tasavvur qilish uchun asboblar qutisi quyidagilardan iborat

10.1-jadvalda keltirilgan uchta funktsiyadan.

**Jadval 10.1 - Tarqatishni vizualizatsiya qilish funktsiyalari**

**ma'lumotlar**

**Funktsiya**

**Tavsif**

distplot ()

Ish natijalari diagrammasi

funktsiyalari: hist (), kdeplot () va rugplot ().

kdeplot ()

Bir o'lchovli yoki ikki o'lchovli yadroni namoyish etadi

zichlikni baholash.

gilamcha ()

Ma'lumot elementlarini yonidagi qator sifatida aks ettiradi

koordinata o'qi.

**10.1 distplot () funktsiyasi**

Distplot () funktsiyasi taqsimotlarni tasavvur qilish uchun mo'ljallangan

bir o'lchovli ma'lumotlar to'plamlari. U chizgan diagramma mumkin

quyidagi tarkibiy qismlardan iborat:

• *Matplotlib-* dan hist () funktsiyasi *natijasi* ;

• dengiz dengizidan kdeplot () funktsiyasi *natijasi* ;

• dengiz tubidan rugplot () funktsiyasining *natijasi.*

Ushbu komponentlar ixtiyoriy: qaysi birini tanlashingiz mumkin

foydalanish uchun funktsiyalar to'plami.

Distplot () funktsiyasining eng muhim parametrlarini ko'rib chiqamiz:

• a: ketma- *ketliklar* , *1 o'lchovli* massiv yoki ro'yxat.

◦ grafigi chizmachilik uchun DataSet bo'lishi mumkin

ketma-ket tuzilma yoki bir o'lchovli massiv.

246

|  |
| --- |
| **Sahifa 248** |

• axlat qutilari: int, list, str, None, ixtiyoriy

◦ *Matplotlib kutubxonasining* hist () funktsiyasi uchun qutilar soni .

• tarix: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda histogram ko'rsatiladi

grafik maydon, agar False bo'lsa, unda yo'q.

• kde: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda KDE diagrammasi (natijasi

kdeplot ()) funktsiyasi uchastka maydonida ko'rsatiladi, agar

False, keyin yo'q.

• gilamcha: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda funktsiya natijasi

rugplot () fitna maydonida ko'rsatiladi, agar u noto'g'ri bo'lsa,

keyin yo'q.

Keling, tajribalar uchun ma'lumotlar bazasini yaratamiz va uni tasavvur qilamiz:

np.random.seed ( 123 )

x = np.random.chisquare ( 2 , 500 )

sns.distplot (x)

**10.1-rasm - distplot () funktsiyasini namoyish etish**

247

|  |
| --- |
| **Sahifa 249** |

Keling, *x* asosida pandalar quramiz va hosil bo'lgan ob'ektni uzatamiz

distplot ():

s = pd.Series (x)

sns.distplot (lar)

**10.2-rasm - bilan distplot () funktsiyasini namoyish etish**

**pandalar turidagi argument**

Keling, qutilar parametrining turli xil qiymatlari qanday ta'sir qilishini namoyish qilaylik

diagramma ko'rinishi:

bs = [ Yo'q , 1 , 3 , 5 , 15 ]

plt.figure ( figsize = ( 15 , 5 ))

uchun i, b yilda Enumerate (BS):

plt.subplot ( 1 , len (bs), i + 1 )

plt.title ( **f "axlat qutilari =** { b } **"** )

sns.distplot (s, axlat qutilari = b)

**10.3-rasm - axlat qutilari funktsiyasi parametri bilan ishlashni namoyish etish**

**distplot ()**

248

|  |
| --- |
| **Sahifa 250** |

Gistogrammani diagrammadan olib tashlash uchun parametrni belgilang

tarix False:

sns.distplot (s, hist = False )

**10.4-rasm - gistogrammasiz diagramma**

Parametr yadro zichligi smetasini ko'rsatish uchun javobgardir

kde:

sns.distplot (s, kde = False )

**10.5-rasm - kde holda diagramma**

249

|  |
| --- |
| **Sahifa 251** |

Rugplot () natijasini diagramaga qo'shaylik:

sns.distplot (s, gilamchali = rost )

**Shakl 10.6 - tarqatish qo'shilgan diagramma**

Ko'rinishni yanada sozlash uchun siz foydalanishingiz mumkin

quyidagi parametrlarga ega:

• rang: *Matplotlib -* rang , ixtiyoriy

Chart Diagramma elementlarining rangi. Turli xil elementlar uchun bo'ladi

tanlangan turli xil soyalar.

• vertikal: bool, ixtiyoriy

◦ Diagramma yo'nalishi: To'g'ri - vertikal, noto'g'ri (yoki yo'q)

- gorizontal.

• norm\_hist: bool, ixtiyoriy

◦ Agar qiymat To'g'ri bo'lsa, u holda gistogramma tuzishda

ustunlarning balandligi emas, balki zichlik qiymatini aks ettiradi

ma'lumotlar to'plamidagi elementlarning soni.

• axlabel: str, False, None, ixtiyoriy

◦ Absissa o'qining nomi (x o'qi). Agar qiymat Yo'q bo'lsa, unda bo'ladi

o'tgan strukturaning Name parametrining qiymati

argument a (agar u bo'lsa), aks holda imzo bo'lmaydi

etkazib berildi.

250

|  |
| --- |
| **Sahifa 252** |

• yorliq: satr, ixtiyoriy

Legend Afsona uchun yorliq.

Keling, amalda ushbu parametrlar bilan ishlaylik. Grafiklarning rangini o'zgartiring

yashil rangga:

sns.distplot (s, gilamcha = To'g'ri , rang = **"g"** )

**10.7-rasm - bilan distplot () funktsiyasini namoyish etish**

**parametr rangi**

Yo'nalishni o'zgartiraylik:

sns.distplot (s, gilam = rost , vertikal = rost )

**Shakl 10.8 - bilan distplot () funktsiyasini namoyish etish**

**vertikal parametr bilan**

251

|  |
| --- |
| **Sahifa 253** |

*X* o'qi uchun yorliqni o'rnatamiz :

sns.distplot (s, gilamcha = True , axlabel = **"chi-kvadrat qiymatlari"** )

**Shakl 10.9 - bilan distplot () funktsiyasini namoyish etish**

**parametr axlabel**

Grafiklarning taqdimotini yaxshi sozlash uchun foydalaning

hist\_kws, kde\_kws va rug\_kws parametrlari, ular orqali

tegishli funktsiyalarning parametrlari:

• hist\_kws: dikt, ixtiyoriy

◦ hist dalillar () funktsiyasi.

• kde\_kws: dikt, ixtiyoriy

◦ kdeplot dalillar () funktsiyasi.

• rug\_kws: dict, ixtiyoriy

◦ rugplot dalillar () funktsiyasi.

Keling, ular bilan qanday ishlashni misol yordamida namoyish etamiz:

h\_kws = { **"alfa"** : 0,3 , **"rang"** : **"r"** }

k\_kws = { **"soya"** : To'g'ri , **"rang"** : **"g"** }

r\_kws = { **"balandlik"** : 0.1 }

sns.distplot (s, axlat qutilari = 10 , gilamchali = rost , hist\_kws = h\_kws, kde\_kws = k\_kws,

rug\_kws = r\_kws)

252

|  |
| --- |
| **Sahifa 254** |

**Shakl 10.10 -** hist\_kws **parametrlari bilan ishlashni namoyish etish**,

kde\_kws **va** rug\_kws distplot () **funktsiyalari**

**10.2 kdeplot () funktsiyasi**

Kdeplot () funktsiyasi bir o'lchovli yoki ikki o'lchovli yadroni namoyish etadi

*zichlik smetasi* - *yadro zichligini baholash* (bundan keyin *KDE* ). Ma'lumotlar

qaysi grafikalar tuzilganligi ma'lumotlar va ma'lumotlar2 parametrlari orqali o'tadi (ichida

ikki o'lchovli baholash holati):

• ma'lumotlar: bir o'lchovli massiv

Plot Chizish uchun ma'lumotlar to'plami.

• ma'lumotlar2: bir o'lchovli qator, ixtiyoriy

Agar kerak bo'lsa qo'shimcha ma'lumotlar to'plami

2D *KDE* .

Keling yuk ma'lumotlar yig'indisi *maslahatlar* va *MPG* :

maslahatlar = sns.load\_dataset ( **"maslahatlar"** )

mpg = sns.load\_dataset ( **"mpg"** )

Keling qurish *KDE* uchun *olish* parametr :

sns.kdeplot (mpg [ **"joy almashtirish"** ] )

253

|  |
| --- |
| **Sahifa 255** |

**10.11-rasm -** kdeplot () **funktsiyasini namoyish etish .**

**Bir o'lchovli variant**

Misol 2D *KDE* :

sns.kdeplot (maslahatlar [ **"total\_bill"** ], maslahatlar [ **"maslahat"** ])

**10.12-rasm -** kdeplot () **funktsiyasini namoyish etish .**

**Ikki o'lchovli variant**

10.12-rasmda ko'rsatilganiga o'xshash diagramma tuzamiz

*mpg* to'plami :

x = mpg [ **"silindrlar"** ]

y = mpg [ **"joy almashtirish"** ]

sns.kdeplot (x, y)

254

|  |
| --- |
| **Sahifa 256** |

**10.13-rasm - kdeplot () funktsiyasini namoyish qilish**

***mpg* ma'lumotlar to'plami**

Diagrammaning ko'rinishini sozlash uchun siz foydalanishingiz mumkin

quyidagi parametrlarga ega:

• soya: bool, ixtiyoriy

◦ ostidagi plomba borligini (Haqiqiy) yoki yo'qligini (Noto'g'ri) aniqlaydi

1D yoki yo'lni to'ldirish uchun *KDE* egri chizig'i

ikki o'lchovli versiya uchun.

• vertikal: bool, ixtiyoriy

◦ Grafik joylashuvi. Agar qiymat rost bo'lsa

parametrning raqamli qiymati *x* o'qi bo'ylab chizilgan , *KDE -*

*y* o'qida , agar False bo'lsa, aksincha.

• rang: *Matplotlib -* rang , ixtiyoriy

◦ Grafik rangi va to'ldirilishi.

• smap: str yoki Colormap *,* ixtiyoriy

◦ grafik rangli xarita (bo'lim "ga qarang *4.4.1 Rang*

*xaritalar (kolormaplar)* ".

• afsona: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda afsona qo'shiladi

diagramma.

255

|  |
| --- |
| **257-bet** |

• cbar: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda diagramma qo'shiladi

ranglar paneli (2D *KDE* uchun ishlatiladi ).

• shadow\_lowest: bool, ixtiyoriy

The To'ldirishni boshqaradi. Agar parametr True bo'lsa, u yashiringan bo'ladi

eng past *KDE* qiymatiga ega maydon . Faqat ishlatilgan

2D *KDE uchun.*

• kümülatif: bool, ixtiyoriy

True Agar rost bo'lsa, tarqatish funktsiyasi ko'rsatiladi.

• gridsize: int, ixtiyoriy

◦ Tarmoqdagi ochkolar soni.

Yuqorida sanab o'tilgan parametrlar bilan ishlashni amalda ko'rib chiqamiz.

*KDE* ning bir o'lchovli variantini to'ldirishni ko'rsatamiz :

sns.kdeplot (y, soya = rost )

**10.14-rasm - soya parametri bilan ishlashni namoyish etish**

**kdeplot () funktsiyalari**

256

|  |
| --- |
| **Sahifa 258** |

Afsonani maydondan olib tashlaymiz:

sns.kdeplot (y, soya = To'g'ri , afsona = Yolg'on )

**10.15-rasm - afsonaviy parametr bilan ishlashni namoyish etish**

**kdeplot () funktsiyalari**

Rangni o'zgartiring:

sns.kdeplot (y, soya = rost , rang = **'r'** )

**10.16-rasm - rang parametri bilan ishlashni namoyish etish**

**kdeplot () funktsiyalari**

257

|  |
| --- |
| **Sahifa 259** |

Ikki o'lchovli versiya uchun to'ldirish bilan ishlashga misol:

sns.kdeplot (x, y, soya = rost )

**10.17-rasm - soya parametri bilan ishlashni namoyish etish**

**kdeplot () funktsiyalari (2D ma'lumotlar to'plami)**

Ko'rsatiladigan darajalar sonini belgilaylik:

sns.kdeplot (x, y, soya = rost , n\_levels = 3 )

**10.18-rasm - n\_levels parametri bilan ishlashni namoyish etish**

**kdeplot () funktsiyalari (2D ma'lumotlar to'plami)**

258

|  |
| --- |
| **Sahifa 260** |

Vizual taqdimotni yaxshiroq qilish uchun ranglar palitrasini o'zgartiring

" *plazma* ":

sns.kdeplot (x, y, soya = To'g'ri , smap = **'plazma'** )

**10.19-rasm - cmap funktsiyasi parametri bilan ishlashni namoyish etish**

**kdeplot () (2D ma'lumotlar to'plami)**

Eng past darajali to'yib oling *KDE* diagramma qiymati va

*colorbar* qo'shish :

sns.kdeplot (x, y, soya = True , cmap = **'plazma'** , shade\_lowest = FALSE ,

cbar = rost )

**Shakl 10.20 - shadow\_lowest parametri bilan ishlashni namoyish etish**

**va cbar kdeplot () funktsiyasi (2D ma'lumotlar to'plami)**

259

|  |
| --- |
| **Sahifa 261** |

Paletaning yana bir varianti:

sns.kdeplot (x, y, soya = rost , smap = **'kamalak'** , cbar = rost )

**10.21-rasm - *kamalak* palitrasi**

Keling, tarqatish funktsiyasini tuzamiz:

sns.kdeplot (y, shadow = True , kümülatif = True )

**10.22-rasm - Kümülatif parametr bilan ishlashni namoyish etish**

**kdeplot () funktsiyalari**

260

|  |
| --- |
| **Sahifa 262** |

Parametrlar hisob-kitoblarni boshqarish uchun ishlatilishi mumkin

yadro, bw, gridsize va kesilgan:

• yadro: {'gau' | 'cos' | 'biw' | 'epa' | 'uch' | 'triw'},

ixtiyoriy

◦ yadro funktsiyasi. Ikki o'lchovli versiya uchun faqat

Gauss yadrosi (gau)

• bw: {'scott' | 'silverman'}, raqamli qiymat, juft juft

(2D uchun), ixtiyoriy

◦ yadro kengligi. Agar siz sukutni qoldirsangiz, u shunday bo'ladi

eng yaxshi ingl. beradigan qiymat tanlanadi

vakillik.

sns.kdeplot (y, soya = True , kernel = **'cos'** )

sns.kdeplot (y, shadow = True , kernel = **'gau'** )

**10.23-rasm - yadro parametri bilan ishlashni namoyish etish**

**kdeplot () funktsiyalari**

261

|  |
| --- |
| **Sahifa 263** |

sns.kdeplot (y, soya = True , label = **"standart"** )

sns.kdeplot (y, shadow = True , bw = 10 , label = **"bw = 10"** )

sns.kdeplot (y, soya = To'g'ri , bw = 70 , yorliq = **"bw = 70"** )

**10.24-rasm - bw funktsiyasi parametri bilan ishlashni namoyish etish**

**kdeplot ()**

**10.3 Rugplot () funktsiyasi**

Rugplot () funktsiyasi elementlarni yonidagi qator sifatida aks ettiradi

koordinata o'qi. Funktsiya parametrlari:

• a: vektor

◦ Bir o'lchovli ma'lumotlar to'plami.

• balandlik: raqamli qiymat, ixtiyoriy

◦ Chiziq balandligi.

• o'qi: {'x' | 'y'}, ixtiyoriy

The Diagramma ko'rsatiladigan o'q.

*Mpg* ma'lumotlar to'plami bilan *ishlaylik* :

mpg = sns.load\_dataset ( **'mpg'** )

*Rugplot* diagrammasini *tuzamiz* :

sns.rugplot (mpg [ **"joy almashtirish"** ])

262

|  |
| --- |
| **Sahifa 264** |

**10.25-rasm - rugplot () funktsiyasini namoyish etish**

Chiziqlarning rangini va balandligini o'zgartiring:

sns.rugplot (mpg [ **"joy almashtirish"** ], balandligi = 0,5 , rang = **'to'q sariq'** )

**10.26-rasm - balandlik va rang parametri bilan ishlashni namoyish etish**

**kdeplot () funktsiyalari**

263

|  |
| --- |
| **Sahifa 265** |

Parametr qiymatlarini ko'rsatish uchun *y* o'qini tanlaymiz :

sns.rugplot (mpg [ **"siljish"** ], balandlik = 0,3 , rang = **'r'** , o'q = **'y'** )

**10.27-rasm - funktsiyaning o'qi parametri bilan ishlashni namoyish etish**

**kdeplot ()**

Bundan tashqari, ko'rinishni boshqarish uchun siz foydalanishingiz mumkin

quyidagi parametrlar:

• bolta: matplotlib o'qlari, ixtiyoriy

◦ Maydon (Axes sinfining ob'ekti), unda

diagramma.

• kvarglar: dikt

◦ Konstruktorga uzatiladigan parametrlarga ega lug'at

LineCollection sinfining.

264

|  |
| --- |
| **Sahifa 266** |

**11-bob. Chiziqli modelni ko'rsatish**

**regressiyalar**

Ushbu guruhning vositalari juda qiziqarli funktsiyalarga ega:

ular uzatilgan ma'lumotlardan chiziqli regressiya modelini tuzadilar va

uni asl to'plam bilan birga namoyish eting. Vazifalar

jadval 11.1.

**Jadval 11.1 - Chiziqli modelni vizualizatsiya qilish funktsiyalari**

**regressiyalar**

Funktsiya

Tavsif

lmplot ()

Ma'lumotlar to'plami va regressiya chizig'ini ko'rsatadi,

boshqarish qobiliyatiga ega, ularga qurilgan

maydonlarni fonda grafikalar bilan tartibga solish.

regplot ()

Ma'lumotlar to'plami va regressiya chizig'ini ko'rsatadi,

ularga qurilgan.

residplot ()

Asl nusxadagi elementlarning og'ishlarini aks ettiradi

regressiya modelidan ma'lumotlar to'plami,

ularga chizilgan diagramma shaklida chizilgan.

**11.1 Funksiyalarning umumiy parametrlari**

Vizualizatsiya funktsiyalarining ba'zi umumiy parametrlarini ko'rib chiqamiz

chiziqli regressiya modellari:

• x, y: ma'lumotlar to'plamidagi o'zgaruvchan nomlar, ixtiyoriy

◦ To'plamdan *x* va *y* o'qlarini o'ziga xos xususiyatlarga bog'lang

ma'lumotlar parametri orqali o'tgan ma'lumotlar. Funktsiyalar uchun,

catplot () dan tashqari, bilan vektorlarni o'tkazish joizdir

to'g'ridan-to'g'ri ma'lumotlar, catplot () uchun - yo'q. Parametrlar mumkin

agar kerak bo'lsa, Yo'q-ga o'rnatiladi

barcha ma'lumotlar to'plamini ko'rsatish.

265

|  |
| --- |
| **Sahifa 267** |

• ma'lumotlar: DataFrame

◦ ustunlar joylashgan pandas.DataFrame ma'lumotlar to'plami

Belgilar, satrlar qiymatlardir. Dan berilgan ustun nomlari

ko'rsatilishi kerak bo'lgan parametrlarga o'tkaziladi

*x* va *y* .

• pastlik: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda model

*statsmodels* , mahalliy darajada tortilgan chiziqli

regressiya. Bunday model uchun ishonch oralig'i emas

ko'rsatiladi.

• mustahkam: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda model

barqaror regressiyani baholash uchun *statsmodels* .

• {tarqalmoq, chiziq} \_kws: dict

Function Qo'shimcha funktsiya argumentlarini aniqlang

plt.scatter () va plt.plot ().

**11.2 regplot () funktsiyasi**

Regplot () funktsiyasi ma'lumotlar to'plamini va regressiya chizig'ini aks ettiradi,

ularga qurilgan. *Iris* va *mpg* ma'lumotlar to'plamlarini yuklaylik :

mpg = sns.load\_dataset ( **"mpg"** )

ìrísí = sns.load\_dataset ( **"ìrísí"** )

Mumkin bo'lgan masofa uchun regressiya chizig'ini quraylik

avtoulovni o'z kuchida boshqaring. X va y parametrlari orqali biz o'tamiz

funktsiyalar nomlari, ma'lumotlar orqali - ma'lumotlar to'plami.

sns.lmplot (x = **"ot kuchi"** , y = **"siljish"** , ma'lumotlar = mpg)

266

|  |
| --- |
| **Sahifa 268** |

**11.1-rasm - regplot () funktsiyasini namoyish etish**

*Iris* to'plamidan faqat sinfga tegishli bo'lgan irislarni tanlang

*setosa* :

ìrísí\_mod = ìrísí [iris [ **"turlari"** ] == **"setosa"** ]

Keling, kenglikning tashqi periant lobining uzunligiga bog'liqligini tuzaylik:

sns.regplot (x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod)

**11.2-rasm - *sepal\_length* parametrlariga bog'liqlikning vizualizatsiyasi**

**va *iris* ma'lumotlar to'plamidan *sepal\_width***

267

|  |
| --- |
| **Sahifa 269** |

Ikkala nuqtani (tarqatish parametri) diagramma maydonidan olib tashlashingiz mumkin, yoki

regressiya chizig'i (fit\_reg parametri). Keling, buni namoyish qilaylik,

tarqalish parametrini False-ga o'rnating:

sns.regplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod,

tarqalish = yolg'on )

**11.3-rasm - tarqalish parametri bilan ishlashni namoyish etish**

**regplot () funktsiyalari**

Parametrni "False" ga o'rnatamiz:

sns.regplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod,

fit\_reg = Soxta )

**11.4-rasm - fit\_reg parametri bilan ishlashni namoyish etish**

**regplot () funktsiyalari**

268

|  |
| --- |
| **Sahifa 270** |

Grafiklarning ko'rinishini sozlash uchun siz foydalanishingiz mumkin

parametrlari:

• rang: *Matplotlib -* rang , ixtiyoriy

◦ Ko'rsatilgan elementlarning rangi (nuqta va chiziq).

• marker: *Matplotlib-* dan marker kodi

Points Ko'rsatilgan markerning uslubi.

sns.regplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod, color = **'r'** )

**11.5-rasm - funktsiyaning rang parametri bilan ishlashni namoyish etish**

**regplot ()**

markerlar = [ **"^"** , **"\*"** , **"o"** ]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, M yilda Enumerate (marker):

plt.subplot ( 1 , len (markerlar), i + 1 )

plt.title ( **f "marker =** { m } **"** )

sns.regplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod,

rang = **'r'** , marker = m)

269

|  |
| --- |
| **Sahifa 271** |

**11.6-rasm - marker parametri bilan ishlashni namoyish etish**

**regplot () funktsiyalari**

Batafsil sozlash uchun scatter\_kws parametrlaridan foydalaning,

line\_kws:

s\_kws = { **"** linewidths **"** : 2 , **"edgecolors"** : **'g'** , **'color'** : **'r'** , **"s"** : 50 }

p\_kws = { **'ls'** : **'-.'**, **'lw'** : 3 }

sns.regplot (x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod,

scatter\_kws = s\_kws, line\_kws = p\_kws)

**11.7-rasm - scatter\_kws parametri bilan ishlashni namoyish etish va**

**line\_kws regplot () funktsiyasi**

270

|  |
| --- |
| **Sahifa 272** |

Replog () funktsiyasi tarqalish chizmasiga alternativa beradi -

ball qiymatlari va ishonch oraliqlari bilan jadval. Uchun

ushbu rejim sozlamalari quyidagi parametrlardan foydalaniladi:

• x\_estimator: funktsiya, ixtiyoriy

The Bal qiymatini hisoblash funktsiyasi.

• x\_bins : int yoki vektor , ixtiyoriy

◦ guruhlar sonini aniqlaydi

asl qadriyatlar to'plami.

• x\_ci: 'ci', 'sd', int [0, 100] oralig'ida yoki yo'q, ixtiyoriy

◦ ishonch oralig'i hajmi. Agar qiymat 'sd' bo'lsa,

keyin ishonch oralig'i o'rniga,

standart og'ish.

Median ball bilan diagramma tuzamiz:

dan numpy import Media

sns.regplot (x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod,

x\_estimator = median)

**Shakl 11.8 - O'rtacha baho bilan jadval**

271

|  |
| --- |
| **Sahifa 273** |

X\_bins parametri bilan ishlashga misol:

x\_bins = [ 5 , 10 , yo'q ]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, B ham Enumerate (x\_bins):

plt.subplot ( 1 , len (x\_bins), i + 1 )

plt.title ( **f "x\_bins =** { b } **"** )

sns.regplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod,

x\_bins = b)

**11.9-rasm - funktsiyaning x\_bins parametri bilan ishlashni namoyish etish**

regplot ()

X\_ci parametri bilan ishlashga misol:

x\_ci = [ 50 , 80 , **"sd"** ]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, xc yilda Enumerate (x\_ci):

plt.subplot ( 1 , len (x\_ci), i + 1 )

plt.title ( **f "x\_ci =** { xc } **"** )

sns.regplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod,

x\_ci = xc, x\_bins = 10 )

272

|  |
| --- |
| **Sahifa 274** |

**11.10-rasm - funktsiyaning x\_ci parametri bilan ishlashni namoyish etish**

**regplot ()**

Siz sezgan bo'lishingiz mumkin, xuddi shu regressiya chizig'i atrofida maydon mavjud

ranglar asosiy chiziq sifatida, ammo engilroq soyada. Bu

ushbu maydon uchun hisoblangan ishonch oralig'ini anglatadi

quyidagi parametrlar bilan boshqariladi:

• ci: int [0, 100] oralig'ida yoki yo'q, ixtiyoriy

The Ishonch oralig'ining o'lchami.

• n\_boot: int, ixtiyoriy

Ci ni baholash uchun namunalar soni.

Diagrammaning har xil qiymatlar bilan qanday ko'rinishini ko'rib chiqamiz.

ci:

ci = [ 50 , 99 , Hech narsa ]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, c yilda enumerate (CI):

plt.subplot ( 1 , len (ci), i + 1 )

plt.title ( **f "ci =** { c } **"** )

sns.regplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , data = iris\_mod, ci = c)

273

|  |
| --- |
| **Sahifa 275** |

**11.11-rasm - funktsiya ci parametri bilan ishlashni namoyish etish**

**regplot ()**

Regplot () funktsiyasining ba'zi parametrlarini ko'rib chiqamiz:

• logistik: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda taxmin qabul qilinadi,

bu *y* bir tomonlama o'zgarmaydigan va modelini qurish uchun

logistik regressiya modeli

*statsmodels* .

*Iris* to'plamiga yana bir xususiyat *qo'shaylik* : *is\_setosa* , qaysi

ìrísí *setoza* bo'lsa, to'g'ri bo'lsa , aks holda -

Noto'g'ri qiymat:

ìrísí [ **"is\_setosa"** ] = ìrísí [ **"turlari"** ] == **'setosa'**

sns.regplot (x = **"sepal\_length"** , y = **"is\_setosa"** , logistik = To'g'ri , ma'lumotlar = ìrísí)

**11.12-rasm - logistik parametr bilan ishlashni namoyish etish**

**regplot () funktsiyalari**

274

|  |
| --- |
| **Sahifa 276** |

• mustahkam: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda model

barqaror regressiyani baholash uchun statsmodels.

Sog'lom parametr bilan ishlashga misol:

sns.regplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , mustahkam = True ,

ma'lumotlar = iris\_mod)

**Shakl 11.13 - mustahkam parametr bilan ishlashni namoyish etish**

**regplot () funktsiyalari**

• logx: bool, ixtiyoriy

◦ Agar qiymat True bo'lsa, u holda chiziqli regressiya baholanadi

shakl *y ~ log (x)* . Parametr x kattaroq bo'lishi kerak

chizish.

Logx parametri bilan ishlashga misol:

sns.regplot ( x = **"mpg"** , y = **"siljish"** , logx = To'g'ri , ma'lumotlar = mpg)

275

|  |
| --- |
| **Sahifa 277** |

**Shakl 11.14 - logx funktsiyasi parametri bilan ishlashni namoyish etish**

**regplot ()**

• pastlik: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u mahalliy sifatida tuziladi

vaznli chiziqli regressiya. Uchun ishonch oralig'i

bunday model namoyish etilmaydi.

Past parametr bilan ishlashga misol:

sns.regplot ( x = **"mpg"** , y = **"siljish"** , lo wess = To'g'ri , ma'lumotlar = mpg)

**Shakl 11.15 - pastlik parametri bilan ishlashni namoyish etish**

**regplot () funktsiyalari**

276

|  |
| --- |
| **Sahifa 278** |

• order: int, ixtiyoriy

◦ Agar parametr qiymati 1 dan katta bo'lsa, u holda

funktsiya numpy.polyfit polinomni tuzish uchun

regressiya.

Buyurtma parametri bilan ishlashga misol:

buyurtma = [ 1 , 2 , 3 ]

plt.figure ( figsize = ( 15 , 5 ))

uchun I, o yilda Enumerate (tartibi):

plt.subplot ( 1 , len (buyurtma), i + 1 )

plt.title ( **f "order =** { o } **"** )

sns.regplot ( x = **"mpg"** , y = **"siljish"** , tartib = o, ma'lumotlar = mpg)

**Shakl 11.16 - Buyurtma parametri bilan ishlashni namoyish etish**

**regplot () funktsiyalari**

• qisqartirish: bool, ixtiyoriy

◦ Odatiy bo'lib, regressiya chizig'i hamma uchun ko'rsatiladi

x o'qi oralig'i. Agar parametr True bo'lsa, u holda satr bo'ladi

ma'lumotlar to'plamining haddan tashqari qiymatlari bilan cheklangan.

277

|  |
| --- |
| **Sahifa 279** |

Qisqartirilgan parametr bilan ishlashga misol:

sns.regplot (x = **"mpg"** , y = **"siljish"** , truncate = rost , ma'lumotlar = mpg)

**Shakl 11.17 - qisqartirilgan parametr bilan ishlashni namoyish etish**

**regplot () funktsiyalari**

**11.3 Funktsiyaning qoldiq joyi ()**

Residplot () asl nusxadagi elementlarning farqlarini aks ettiradi

ma'lumotlar bazasi, ularga o'rnatilgan regressiya modelidan, shaklda

tarqalish diagrammasi. Bunday diagrammaning har bir nuqtasi farqdir

elementning asl to'plamdagi qiymati va qiymati o'rtasidagi

o'sha paytda regressiya modelini qaytaradi. Funktsiya parametrlari

residplot () " *11.1* " bo'limida ko'rsatilgan to'plam bilan bir xil

*Umumiy funktsiya parametrlari* ".

Uning ishlatilishining ba'zi bir misollari. Biz bilan ishlaymiz

*mpg* to'plami :

mpg = sns.load\_dataset ( **"mpg"** )

Regplot () funktsiyasi yordamida chiziqli regressiya modelini yarataylik:

sns.regplot (x = **"mpg"** , y = **"siljish"** , ma'lumotlar = mpg)

278

|  |
| --- |
| **Sahifa 280** |

**11.18-rasm - qurilgan chiziqli regressiya modeli grafigi**

**regplot () yordamida**

Endi belgilangan qiymatlarning og'ishlarining tarqalish chizmasini tuzamiz

modeldagi ma'lumotlar:

sns.residplot (x = **"mpg"** , y = **"siljish"** , rang = **'g'** , ma'lumotlar = mpg)

**11.19-rasm - Belgilangan qiymatlar sapmalarining tarqalish diagrammasi**

**residplot () yordamida tuzilgan model ma'lumotlari**

279

|  |
| --- |
| **Sahifa 281** |

Uchinchi tartibli modelga misol:

sns.regplot (x = **"mpg"** , y = **"siljish"** , buyurtma = 3 , ma'lumotlar = mpg)

**11.20-rasm - Uchinchi darajali model grafigi**

Bunday model uchun og'ishlar taqsimoti quyidagicha bo'ladi:

sns.residplot (x = **"mpg"** , y = **"siljish"** , tartib = 3 , rang = **'g'** , ma'lumotlar = mpg)

**Shakl 11.21 - Uchinchi model uchun og'ishlar taqsimoti**

**buyurtma**

280

|  |
| --- |
| **Sahifa 282** |

**11.4 lmplot () funktsiyasi**

Lmplot () funktsiyasi regplot () ga o'xshashdir (qarang).

Boshqarish qobiliyatiga ega bo'lgan " *11.2* regplot () *funktsiyasi* ") bo'limiga qarang

maydonlarni fonda grafikalar bilan tartibga solish.

*Mpg* va *ìrísí* ma'lumotlar to'plamlarini yuklaylik :

mpg = sns.load\_dataset ( "mpg" )

ìrísí = sns.load\_dataset ( "ìrísí" )

Agar sozlamaga javob beradigan parametrlarni chetga surib qo'ysak

substratlar, aks holda, lmplot () funktsiyasining argumentlari deyarli amal qiladi

ko'rib chiqilganlar bilan to'liq nom va maqsadga to'g'ri keladi

regplot () funktsiyalari. Bularga x, y, ma'lumotlar, tartib, x\_bins, birliklar,

ci, x\_ci, n\_boot, {x, y} \_jitter, {x, y} \_qism, x\_estimator, scatter,

fit\_reg, logistik, mustahkam, logx, past, qisqartirilgan, {tarqalmoq,

line} \_kws, dropna.

Ma'lumotlarni guruhlarga ajratish va boshqarish uchun javobgar parametrlar

lmplot () funktsiyalari argumentlarga mos keladi

relplot () va catplot () funktsiyalari, ularga rang, qator, kol,

pallet, hue\_order, afsona, legend\_out, balandlik, tomon. Shuning uchun biz

biz ularga to'liq tavsif bermaymiz, o'zimizni ish namunasi bilan cheklaymiz

ular bilan.

Lmplot () funktsiyasi yordamida regressiya modelini tuzamiz,

regplot () funktsiyasi bilan olishimiz mumkin bo'lgan narsa.

sns.lmplot (x = **"ot kuchi"** , y = **"siljish"** , ma'lumotlar = mpg)

281

|  |
| --- |
| **Sahifa 283** |

**11.22-rasm - qurilgan chiziqli regressiya modeli grafigi**

**lmplot bilan ()**

Ma'lumotlarni kategorik asosda keyingi bilan ajratish

har bir guruh uchun funktsiyadagi o'z regressiya modelini yaratish

lmplot () quyidagi parametrlardan foydalanadi:

• rang: rang ajratish uchun;

• qator: diagrammalarning vertikal namoyishi uchun;

• kol: diagrammalarning gorizontal namoyishi uchun.

Paletli parametr rang sxemasini o'rnatish uchun javobgardir.

Keling qurish uchun diagramma *ìrísí* o'rnatish ìrísí turiga ko'ra bo'linish bilan

( *turlar* xususiyati ):

sns.lmplot ( x = **"sepal\_width"** , y = **"petal\_width"** , hue = **" types "** , data = iris)

282

|  |
| --- |
| **Sahifa 284** |

**11.23-rasm - rang funktsiyasi parametri bilan ishlashni namoyish etish**

**lmplot ()**

Paletani o'zgartiramiz:

sns.lmplot ( x = **"sepal\_width"** , y = **"petal\_width"** , hue = **" types "** ,

palitrasi = **"Dark2"** , ma'lumotlar = ìrísí)

**11.24-rasm - Palitra parametri bilan ishlashni namoyish etish**

**lmplot () funktsiyalari**

283

|  |
| --- |
| **Sahifa 285** |

Har bir uchun Keling bugungi grafiklari *turlari* qiymati haqida

gorizontal bo'linish bilan turli xil maydonlar:

sns.lmplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , hue = **" types "** ,

palitra = **"Dark2"** , col = **"turlari"** , ma'lumotlar = ìrísí)

**11.25-rasm - funktsiya parametri bilan ishlashni namoyish etish**

**lmplot ()**

Vertikal bo'linish:

sns.lmplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , hue = **" types "** ,

palitra = **"Dark2"** , satr = **"turlar"** , ma'lumotlar = ìrísí)

284

|  |
| --- |
| **Sahifa 286** |

**11.26-rasm - funktsiya qatori parametri bilan ishlashni namoyish etish**

**lmplot ()**

285

|  |
| --- |
| **Sahifa 287** |

Diagrammalar hajmini boshqarish uchun balandlik parametrlaridan foydalaning

(balandlik) va tomon (tomonlar nisbati):

sns.lmplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , hue = **" types "** ,

palitrasi = **"Dark2"** , col = **"turlari"** , balandligi = 3 , tomoni = 1,5 , ma'lumotlar = ìrísí)

**Shakl 11.27 - balandlik parametri bilan ishlashni namoyish etish va**

**aspekt funktsiyasi lmplot ()**

Sharex va sharey parametrlari barchada x va y o'qlarini aks ettirish uchun javobgardir

diagrammalar yoki faqat o'ta chap tomonda (gorizontal holda)

joylashuvi) yoki pastki qismida (vertikal tartibga solish uchun):

sns.lmplot ( x = **"sepal\_length"** , y = **"sepal\_width"** , col = **" types "** ,

sharey = **'kol'** , ma'lumotlar = ìrísí)

**Shakl 11.28 - Sharey parametri bilan ishlashni namoyish etish**

**lmplot () funktsiyalari**

Rang tartibi (agar rang parametridan foydalanilsa) yoki buyurtma

diagrammalarni ko'rsatish (agar col yoki bo'lsa

286

|  |
| --- |
| **Sahifa 288** |

qator) hue\_order, col\_order, row\_order yordamida o'rnatiladi. Keling, beraylik

hue\_order uchun misol:

h\_order = [ **"virginica"** , **"setosa"** , **"versicolor"** ]

sns.lmplot (x = **"sepal\_width"** , y = **"petal\_width"** , hue = **" types "** ,

hue\_order = h\_order, ma'lumotlar = ìrísí)

**11.29-rasm - hue\_order parametri bilan ishlashni namoyish etish**

**lmplot () funktsiyalari**

Afsonaviy parametr afsonani ko'rsatish uchun, uni olib tashlash uchun javobgardir

maydon chegaralari grafikalar bilan - legend\_out.

287

|  |
| --- |
| **Sahifa 289** |

**12-bob. Diagramma maketini boshqarish**

*Seaborn* uchta boshqaruv vositalarini taqdim etadi

tartib sxemalari:

• *Facet-* grid: grafiklarni joylashtirish uchun klassik panjara,

render to'plamlari funktsiyalari bilan qurilgan

ma'lumotlardagi munosabatlar (scatterplot (), lineplot ()), vizualizatsiya

kategorik ma'lumotlar (stripplot (), boxplot () va boshqalar) va

chiziqli regressiya modelining vizualizatsiyasi (regplot (),

residplot ()).

• *Pair-* grid: juftlik munosabatlarini namoyish etish uchun panjara

ma'lumotlar.

• *Joint-* grid: bilan ikkita o'zgaruvchiga diagramma ko'rsatadi

ularning tarqatilishini qo'shimcha ravishda ingl.

**12.1 *Facet* grid**

Biz FacetGrid klassi bilan tanishganimizda tanishgan edik

funktsiyalar relplot (), catplot (), umumiyni amalga oshirishdan tashqari

tegishli funktsiyalar guruhlari uchun interfeys, ular ruxsat berildi

grafikalar sxemasi bilan ishlash. FacetGrid klassi mavjud bo'lgani uchun

to'g'ridan-to'g'ri foydalaning, keyin siz mustaqil ravishda, asosida foydalanishingiz mumkin

mos keladigan ob'ektning tartibini yarating.

FacetGrid konstruktor parametrlarining ko'pchiligida siz shunday bo'lishingiz kerak

relplot () va catplot () tavsiflarini o'qib chiqsangiz, ularga tanish bo'lgansiz

ma'lumotlar, qator, qator, rang, col\_wrap, sharex, sharey, bo'y, tomon,

palitrasi, qator\_qatori, kolorizator, hue\_order, dropna, legend\_out,

margin\_titles.

288

|  |
| --- |
| **Sahifa 290** |

Grafiklarni ko'rsatish uchun ob'ektning map () usulini chaqirish kerak

FacetGrid klassi. Unda quyidagi argumentlar to'plami mavjud:

• funktsiya: qo'ng'iroq qilish mumkin

Ot Plotting funktsiyasi (masalan, tarqalish ()).

• args: torlar

◦ O'tkazilgan to'plamdagi ustunlar nomlari

FacetGrid ob'ektini yaratishda ma'lumotlar parametri.

• kvarglar: kalit so'z argumentlari

◦ funktsiya argumenti orqali o'tgan parametr parametrlari.

Ma'lumotlar bazasini *yuklash* :

nuqta = sns.load\_dataset ( **"nuqta"** )

Keling, dastlabki 250 ta elementni chiqaramiz:

dots\_mod = dots.sample ( frac = 1 ) [: 250 ]

Grafiklarni ustunlarga bo'lish uchun biz parametrdan foydalanamiz

*T1* yoki *T2* qiymatlarini qabul qilishi mumkin bo'lgan *tanlov* , *satrma-* satr -

*hizalamak* , bu *succ* , *nuqta* . Keling, sinf ob'ektini yarataylik

FacetGrid:

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , row = **"** hizalamoq **"** )

Endi, natijada paydo bo'lgan katakchaning har bir katakchasida biz diagrammani namoyish etamiz

*vaqt* uchun tarqalish va *firing\_rate xususiyatlari* :

fg.map (plt.scatter, **"firing\_rate"** , **"time"** )

289

|  |
| --- |
| **Sahifa 291** |

**12.1-rasm - kol** va **qator parametrlari bilan ishlashni namoyish etish**

**FacetGrid klassi**

Tarqoqlik chizmasi o'rniga biz tarqatish gistogrammalarini tuzamiz

*muvofiqlik* xususiyati :

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , row = **"** hizalamoq **"** )

fg.map (plt.hist, **"muvofiqlik"** )

290

|  |
| --- |
| **Sahifa 292** |

**12.2-rasm - kol** va **qator parametrlari bilan ishlashni namoyish etish**

**FacetGrid klassi**

Qo'shimcha rang ajratish uchun siz foydalanishingiz mumkin

rang parametrlari:

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , hue = **"** hizalamoq **"** )

fg.map (plt.scatter, **"firing\_rate"** , **"time"** )

**12.3-rasm - Hue class parametri bilan ishlashni namoyish etish**

**FacetGrid**

291

|  |
| --- |
| **Sahifa 293** |

Paletani o'zgartiramiz:

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , hue = **"align"** , palitra = **"Set2"** )

fg.map (plt.scatter, **"firing\_rate"** , **"time"** )

**12.4-rasm - sinf palitrasi parametri bilan ishlashni namoyish etish**

**FacetGrid**

Balandlik va tomon parametrlaridan foydalangan holda diagrammalarning o'lchamlarini belgilaylik:

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , hue = **"hizalama"** , balandlik = 5 ,

aspekt = 0,5 )

fg.map (plt.scatter, **"firing\_rate"** , **"time"** )

**12.5-rasm - balandlik va parametrlari bilan ishlashni namoyish etish**

**FacetGrid sinfining aspekti**

292

|  |
| --- |
| **Sahifa 294** |

Grafiklarni ko'rsatish tartibini o'zgartirib, afsonani qo'shamiz:

col\_ord = [ **"T1"** , **"T2"** ]

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , hue = **"align"** , col\_order = col\_ord)

fg.map (plt.scatter, **"firing\_rate"** , **"time"** ) .add\_legend ()

**12.6-rasm - col\_order parametri bilan ishlashni namoyish etish**

**FacetGrid klassi**

Parametrni belgilash orqali afsonani grafik bilan maydonga qo'yish mumkin

legend\_out noto'g'ri. Xlim, ylim parametrlari oraliqlarni o'rnatadi

*x* va *y* o'qlari uchun :

ylim = (- 1000 , 1000 )

xlim = ( 0 , 100 )

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , hue = **"align"** , xlim = xlim,

ylim = ylim)

fg.map (plt.scatter, **"firing\_rate"** , **"time"** )

**12.7-rasm - xlim va ylim parametrlari bilan ishlashni namoyish etish**

**FacetGrid klassi**

293

|  |
| --- |
| **Sahifa 295** |

Mana koordinata o'qlari va panjara bilan ishlashning ba'zi bir misollari. Shunga qaramay

Bir marta biz standart parametrlarga ega bo'lgan diagramma variantini taqdim etamiz:

fg.map (plt.scatter, **"firing\_rate"** , **"time"** )

**12.8-rasm - Standart parametrlarga ega diagramma**

Iltimos, e'tibor bering: *y* o'qi barcha chiziqli grafikalar uchun keng tarqalgan va *x*

*-* ustundagi barcha grafikalar uchun. Siz majburan yoqishingiz mumkin

yordamida har bir grafik uchun o'qlardagi belgilarni ko'rsatish

sharex va sharey parametrlari ularga noto'g'ri yuboriladi:

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , satr = **"hizalamak"** , sharex = False ,

sharey = Soxta )

fg.map (plt.scatter, **"firing\_rate"** , **"time"** )

294

|  |
| --- |
| **Sahifa 296** |

**12.9-rasm - sharex parametrlari bilan ishlashni namoyish etish va**

**FacetGrid sinfining sherigi**

Agar despine parametri standart qiymat bilan qoldirilgan bo'lsa, unda biz

biz diagramma maydonining o'ng va yuqori tomonlarini ochamiz:

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , hue = **"** hizalamoq **"** )

fg.map (plt.hist, **"firing\_rate"** )

**Shakl 12.10 - Despine parametri bilan ishlashni namoyish etish**

**FacetGrid klassi**

295

|  |
| --- |
| **Sahifa 297** |

Buni False-ga o'rnatamiz:

fg = sns.FacetGrid (dots\_mod, col = **"choice"** , hue = **"align"** , despine = False )

fg.map (plt.hist, **"firing\_rate"** )

**12.11-rasm - despine = False parametri bilan ishlashni namoyish etish**

**FacetGrid klassi**

Ko'rib turganingizdek, jadvallar bilan maydonlar hamma uchun cheklangan bo'lib qoldi

partiyalar.

**12,2 *Pair-* oro**

Keyingi tartibni boshqarish vositasi biz

o'ylab ko'ring - bu PairPlot. PairPlot klassi panjara hosil qiladi

dan tanlangan xususiyatlarni juft taqqoslash diagrammalarini tuzish

uzatilgan ma'lumotlar to'plami. *Seaborn*

qulaylikni ta'minlaydi

ushbu turdagi maket bilan ishlash vositasi funktsiyadir

juftlik (). Agar ko'proq aniq sozlash kerak bo'lsa, unda bu

holda siz PairPlot sinfidan to'g'ridan-to'g'ri foydalanishingiz kerak

uning usullari. Tanishligimizni juftlik () funktsiyasi bilan boshlaymiz.

296

|  |
| --- |
| **Sahifa 298** |

**12.2.1 pairplot () funktsiyasi**

Pairplot () funktsiyasi elementlari bo'lgan panjarani yaratadi

berilgan xususiyatlar to'plamini juft taqqoslash uchastkalari. Eng yaxshisi uchun

vizual namoyish, Shomil uslubini qo'llang:

sns.set (style = **"ticks"** )

*MPPG* to'plamini yuklang va undan pastki namunani chiqarib oling,

faqat mpg, ot kuchi, siljish belgilarini o'z ichiga olgan

kelib chiqishi:

mpg = sns.load\_dataset ( **"mpg"** )

mpg\_mod = mpg [[ **"mpg"** , **"ot kuchi"** , **"siljish"** , **"kelib chiqish"** ]]

Pairplot () funktsiyasidan foydalanib diagramma tuzamiz:

sns.pairplot (mpg\_mod)

**12.12-rasm - pairplot () funktsiyasi bilan qurilgan diagramma**

297

|  |
| --- |
| **Sahifa 299** |

Siz allaqachon funktsiya parametrlarining ko'pini bilishingiz kerak:

ma'lumotlar, hue, hue\_order, palitrasi, markerlar, balandlik, tomon, pasayish, biz

boshqa funktsiyalarni o'rganayotganda uchrashdi.

Ular bilan ishlashning bir nechta misollari. Bo'linishni qo'shish

Kategoriyali atribut *kelib chiqishi* , biz tarqatgan fitna uchun

uchburchak marker, shuningdek balandlik va tomonlarning nisbatlarini bildiradi:

sns.pairplot (mpg\_mod, hue = **"origin"** , markers = **"^"** , height = 2 , aspect = 1.1 )

**12.13-rasm - rang parametrlari bilan ishlashni namoyish etish,**

**belgilar, balandlik va tomon funktsiyalari juftligi ()**

Ro'yxatda keltirilganlardan tashqari, juftlik () funktsiyasi quyidagilarni o'z ichiga oladi

foydali parametrlar:

• vars: ro'yxat, ixtiyoriy

◦ Foydalaniladigan ma'lumotlar to'plamidagi xususiyatlarning nomlari

namoyish qilish.

298

|  |
| --- |
| **Sahifa 300** |

• {x, y} \_vars: ro'yxat, ixtiyoriy

◦ taqdim etiladigan ma'lumotlar to'plamidagi xususiyatlarning nomlari

qatorlarda (x\_vars) va ustunlarda (y\_vars).

• turdagi: {'tarqatish', 'reg'}, ixtiyoriy

◦ Magistraldan tashqarida joylashgan diagrammalar uchun diagramma turi

diagonallar.

• diag\_kind: {'auto', 'hist', 'kde', None}, ixtiyoriy

◦ Asosiy diagonal elementlari bo'yicha grafik turi.

• burchak: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda yuqori o'ngdagi diagrammalar

qismlar (asosiy diagonal ustidagi hamma narsa) ko'rsatilmaydi.

AQSh displey uchun tanlash bo'lsin sifatlari *mpg* va *ot kuchiga ega* to'plamining

parametrlari yordamida *mpg\_mod* :

sns.pairplot (mpg\_mod, hue = **"origin"** , vars = [ **"mpg"** , **"ot kuchi"** ))

**12.14-rasm - funktsiya bilan ishlashni namoyish etish parametri parametri**

**juftlik ()**

299

|  |
| --- |
| **Sahifa 301** |

To'plamlarni aniqlash uchun x\_vars va y\_vars parametrlaridan foydalanamiz

*x* va *y* o'qlari uchun xususiyatlar :

sns.pairplot (mpg\_mod, hue = **"origin"** , x\_vars = [ **"mpg"** , **"ot"** ),

y\_vars = [ **"joy almashtirish"** ])

**12.15-rasm - x\_vars va parametrlari bilan ishlashni namoyish etish**

**pairplot () funktsiyasining y\_vars**

Tarqoqlik o'rniga regressiya modeli bilan grafik tuzamiz:

sns.pairplot (mpg\_mod, vars = [ **"ot kuchi"** , **"siljish"** ], turdagi = **'reg'** )

**12.16-rasm - funktsiyaning turdagi parametri bilan ishlashni namoyish etish**

**juftlik ()**

300

|  |
| --- |
| **Sahifa 302** |

Asosiy diagonaldagi diagramma turini gistogrammadan *KDE ga o'zgartiramiz* :

sns.pairplot (mpg\_mod, vars = [ **"ot kuchi"** , **"siljish"** ),

diag\_kind = **'kde'** )

**12.17-rasm - diag\_kind parametri bilan ishlashni namoyish etish**

**juftlik () funktsiyalari**

Asosiy diagonal ustidagi barcha diagrammalarni olib tashlaymiz, buning uchun biz tayinlaymiz

burchak parametri To'g'ri:

sns.pairplot (mpg\_mod, vars = [ **"ot kuchi"** , **"siljish"** ], burchak = rost )

301

|  |
| --- |
| **Sahifa 303** |

**12.18-rasm - burchak parametri bilan ishlashni namoyish etish**

**juftlik () funktsiyalari**

**12.2.2 PairPlot klassi**

PairPlot sinfining konstruktori qisman parametrlarni o'z ichiga oladi

funktsiya uchun biz ko'rib chiqqan narsalarga to'g'ri keladi

pairplot (), ularga quyidagilar kiradi: ma'lumotlar, rang, hue\_order, palitrasi, hue\_kws,

vars, {x, y} \_vars, burchak, balandlik, aspekt, dropna, qo'shimcha ravishda

pastki qatlamni o'rnatish uchun ikkita variant:

• layout\_pad: int yoki float, ixtiyoriy

◦ Diagrammalar orasidagi masofa.

• Despine: bool, ixtiyoriy

◦ Agar parametr rost bo'lsa, u holda cheklovchi

uchastka qutisining yuqori va o'ng qismidagi chiziqlar.

Diagrammani FacetGird sinfiga o'xshash tarzda qurish uchun

map () usuli ishlatiladi, argument sifatida uzatiladi

302

|  |
| --- |
| **Sahifa 304** |

chizish uchun funktsiya. PairPlot ko'proq narsani taqdim etadi

asosiy diagrammalar taqdimotini boshqarish funktsiyalari

katakchaning yuqori va pastki qismidagi diagonali: map\_diag (), map\_offdiag (),

map\_lower (), map\_upper ().

Oldindan yuklangan *mpg-* dan ishlash uchun ma'lumotlar to'plamini tayyorlaylik :

mpg\_mod = mpg [[ **"mpg"** , **"vazn"** , **"siljish"** , **"kelib chiqish"** ]]

Uchastka tuzish uchun PairPlot sinf ob'ektining map () usulidan foydalanamiz

diagrammalar:

pg = sns.PairGrid (mpg\_mod)

pg.map (plt.scatter)

**12.19-rasm - PairGrid klassi bilan ishlashni namoyish etish**

303

|  |
| --- |
| **Sahifa 305** |

Keling, *kelib chiqishiga qarab* ajratishni *qo'shaylik* :

pg = sns.PairGrid (mpg\_mod, hue = **'origin'** )

pg.map (plt.scatter)

**Shakl 12.20 - Hue class parametri bilan ishlashni namoyish etish**

**PairGrid**

O'rnatish uchun map\_diag () va map\_offdiag () funktsiyalaridan foydalanamiz

asosiy diagonal va uning tashqarisidagi diagrammalar turlari:

pg = sns.PairGrid (mpg\_mod)

pg.map\_diag (sns.kdeplot)

pg.map\_offdiag (plt.scatter)

304

|  |
| --- |
| **Sahifa 306** |

**12.21-rasm - map\_diag () va funktsiyalari bilan ishlashni namoyish etish**

**map\_offdiag () PairGrid sinfining**

Map\_diag (), map\_lower () va map\_upper () bilan ishlash variantlari:

pg = sns.PairGrid (mpg\_mod)

pg.map\_diag (plt.hist)

pg.map\_lower (plt.scatter)

pg.map\_upper (sns.kdeplot)

305

|  |
| --- |
| **307-bet** |

**12.22-rasm - map\_diag () funktsiyalari bilan ishlashni namoyish etish** ,

**map\_lower () va map\_upper () PairGrid sinfiga kiradi**

**12.3 *qo'shma* mash**

Boshqarish vositalarini ko'rib chiqishning yakuniy mavzusi

tartibi JointGrid sinfining imkoniyatlari haqida umumiy ma'lumot bo'ladi. Bu sinf

asosiy bo'yicha ikki o'lchovli taqsimotni ko'rsatish uchun panjara quradi

diagramma va bir o'lchovli ikkita alohida diagramma

parametrlarning har birini alohida taqsimlash. Xuddi shunday

JointGrid kutubxonasi bilan ishlashni soddalashtirish uchun PairGrid bilan ish

*seaborn* birlashishi () funktsiyasini yanada nozikroq qilish uchun ta'minlaydi

sozlamalari, to'g'ridan-to'g'ri JointGrid ob'ektidan foydalaning.

306

|  |
| --- |
| **Sahifa 308** |

**12.3.1 qo'shma uchastka () funktsiyasi**

Jointplot () funktsiyasi sizga tanish bo'lgan quyidagi to'plamni o'z ichiga oladi

parametrlar: x, y, ma'lumotlar, rang, balandlik, pasayish, {x, y} lim.

*Iris* ma'lumotlar to'plamini yuklaylik :

ìrísí = sns.load\_dataset ( **"ìrísí"** )

Jointplot () funktsiyasi yordamida diagramma tuzamiz:

sns.jointplot (x = **'sepal\_length'** , y = **'sepal\_width'** , data = ìrísí)

**12.23-rasm - jointplot () funktsiyasini namoyish etish**

Diagrammaning rangi va hajmini o'zgartiraylik:

sns.jointplot ( x = **'sepal\_length'** , y = **'sepal\_width'** , data = ìrísí,

rang = **"to'q sariq"** , balandlik = 5 )

307

|  |
| --- |
| **Sahifa 309** |

**12.24-rasm - rang bilan ishlashni namoyish etish va**

**jointplot () funktsiyasining balandligi**

Keling, koordinata o'qlari oralig'ini ko'rsatamiz:

sns.jointplot ( x = **'sepal\_length'** , y = **'sepal\_width'** , data = ìrísí, xlim = ( 0 , 10 ),

ylim = ( 0 , 10 ))

**12.25-rasm - xlim va ylim parametrlari bilan ishlashni namoyish etish**

**jointplot () funktsiyalari**

308

|  |
| --- |
| **310-bet** |

Markaziy parametr turini belgilash uchun turdagi parametr ishlatiladi,

u quyidagi qiymatlardan biriga ega bo'lishi mumkin {'tarqalishi' | 'reg' |

'qoldiq' | 'kde' | 'hex'}.

Quyida diagrammalar ko'rinishini namoyish etish uchun misollar keltirilgan,

turdagi har xil qiymatlar bilan olingan. Variant kind = "scatter" bilan

oldingi misollarda ko'rgan edik.

Kind = 'reg' parametri bilan diagramma:

sns.jointplot ( x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí, kind = **"reg"** )

**12.26-rasm - kind = 'reg' parametri bilan ishlashni namoyish etish.**

**jointplot () funktsiyalari**

309

|  |
| --- |
| **Sahifa 311** |

Type = 'kde' parametri bilan diagramma:

sns.jointplot ( x = **'sepal\_length'** , y = **'petal\_length'** , data = ìrísí, kind = **"kde"** )

**12.27-rasm - kind = 'kde' parametri bilan ishlashni namoyish etish**

**jointplot () funktsiyalari**

Parametri bilan diagramma = = hex ':

sns.jointplot ( x = **'sepal\_length'** , y = **'sepal\_width'** , data = iris, kind = **"hex"** )

**12.28-rasm - kind = 'hex' parametri bilan ishlashni namoyish etish**

**jointplot () funktsiyalari**

310

|  |
| --- |
| **Sahifa 312** |

Kind = 'resid' parametri bilan diagramma:

sns.jointplot ( x = **'sepal\_length'** , y = **'sepal\_width'** , data = ìrísí, kind = **"qoldiq"** )

**12.29-rasm - kind = 'resid' parametri bilan ishlashni namoyish etish**

**jointplot () funktsiyalari**

Markaziy va yon tomonlarning o'lchamlarini boshqarish uchun

Parametr parametri diagrammalar uchun javobgardir:

sns.jointplot ( x = **'sepal\_length'** , y = **'sepal\_width'** , data = ìrísí, nisbat = 2 )

**Shakl 12.30 - nisbati parametri bilan ishlashni namoyish etish**

**jointplot () funktsiyalari**

311

|  |
| --- |
| **Sahifa 313** |

Bo'shliq parametri markaz va yon orasidagi bo'shliqni aniqlaydi

diagrammalar:

sns.jointplot ( x = **'sepal\_length'** , y = **'sepal\_width'** , data = ìrísí, bo'shliq = 2 )

**12.31-rasm - kosmik parametr bilan ishlashni namoyish etish**

**jointplot () funktsiyalari**

312

|  |
| --- |
| **Sahifa 314** |

**12.3.2 JointPlot sinfi**

JointPlot sinfining konstruktor parametrlari qism bilan bir xil

jointplot () funktsiyasining argumentlari: x, y, ma'lumotlar, balandlik, nisbat, bo'shliq,

dropna, {x, y} lim. Diagrammani aks ettirish uchun plot () usulidan foydalaning

yoki usullari plot\_joint (), plot\_marginal (). Uchastka () uslubiga

markazni qurish uchun funktsiyalar

va yon jadvallar:

jg = sns.JointGrid (x = **'sepal\_length'** , y = **'sepal\_width'** , data = ìrísí)

jg.plot (sns.scatterplot, sns.kdeplot)

**12.32-rasm - plot () funktsiyasini namoyish etish**

**JointGrid sinfi**

Plot\_joint () va plot\_marginals () funktsiyalari izolyatsiyaga imkon beradi

qurilish funktsiyalari uchun parametrlarni belgilang va o'tkazing markaziy va

yon jadvallar:

jp = sns.JointGrid (x = **'sepal\_length'** , y = **'sepal\_width'** , data = ìrísí)

jp.plot\_joint (sns.regplot, color = **"g"** )

jp.plot\_marginals (sns.distplot, color = **'r'** )

313

|  |
| --- |
| **Sahifa 315** |

**12.33-rasm - plot\_joint () funktsiyalari bilan ishlashni namoyish etish**

**va plot\_ marginals ()**

jp = sns.JointGrid ( x = **'sepal\_length'** , y = **'sepal\_width'** , data = ìrísí)

jp.plot\_joint (sns.regplot, color = **"g"** )

jp.plot\_marginals (sns.kdeplot, shadow = True , color = **'r'** )

**12.34-rasm - plot\_joint () va funktsiyalari bilan ishlashni namoyish etish**

**plot\_ marginals ()**

314

|  |
| --- |
| **Sahifa 316** |

**III qism. *Mayavi* kutubxonasi**

**Kirish**

*Mayavi* - muammolarni hal qilish uchun vositalar to'plami

*2D* / *3D* ko'rsatish. *Mayavi* tuzilmasi *grafikli vositani* o'z ichiga oladi

*2D / 3D* taqdimotni boshqarish qulay bo'lgan interfeys

ma'lumotlar, deyiladi *mayavi2* . Ikkinchi asosiy komponent

*Python tili* uchun paketlar va modullar to'plami , ular orqali siz

*2D / 3D* renderlashni yaratish, sozlash va ishga tushirishni boshqarish

modellar. Ushbu kutubxona bilan ishlashning kontseptual tamoyillari shunga o'xshashdir

biz *Matplotlib* yoki *dengiz dengizini* o'rganganimizda *amal qildik* . *Mayavi*

ustidan sarıcı *VTK* - *Visualization Toolkit* - kuchli ochiq manba

vizualizatsiya uchun kutubxona.

*Mayavi* quyidagi xususiyatlarni taqdim etadi:

• skaler, vektorli va tensorli ma'lumotlarni *2 o'lchovli ko'rinishda ko'rish*

yoki *3D* ko'rinish;

• *Python tilini* qo'llab-quvvatlash ;

• rivojlanish tufayli funktsional imkoniyatlarni kengaytirish imkoniyati

o'z modullari, filtrlari va boshqalar;

• turli xil fayl formatlarini qo'llab-quvvatlash: *VTK* ( *XML* va undan katta *versiyalar* )

variantlar), *PLOT3D* ;

• ish natijalarini har xil grafik fayllarda saqlash

formatlar;

• tezkor jadval tuzish uchun qulay vositalar

mlab.

315

|  |
| --- |
| **Sahifa 317** |

*Mayavi* bilan ishlashning mohiyati ma'lumotlar oqimlarini yaratishdir

( *quvurlar* ). Birinchi bosqichda ma'lumotlar maxsus moslamalarga yuklanadi

( *ma'lumotlar manbalari* ). To'g'ridan-to'g'ri *VTK* fayllari sifatida foydalanish mumkin,

va sonli massivlardan (yoki boshqasidan) *VTK* tuzilmalarini yaratish

buning imkoni bo'lgan tuzilmalar). Keyinchalik qayta ishlash amalga oshiriladi

ma'lumotlar (qayta ishlash) filtrlar yordamida, ishlov berish natijalari,

renderga uzatildi. Ko'rsatiladigan maydon

modelning ko'rsatish natijasi " *Sahna"* deb nomlanadi .

316

|  |
| --- |
| **Sahifa 318** |

**13-bob. Tezkor boshlash**

**13.1 O'rnatish**

*Mayavini* o'rnatish uchun siz *pip* to'plami menejeridan foydalanishingiz mumkin

yoki *conda* (agar sizda *Anaconda* o'rnatilgan bo'lsa ) yoki kutubxonani qurish

manbadan.

*Pip* orqali o'rnatish uchun quyidagi buyruqlardan foydalaning:

**pip o'rnatish mayavi**

**pip o'rnatish PyQt5**

Manbadan o'rnatish uchun:

**git klon https://github.com/enthought/mayavi.git**

**CD mayavi**

**pip install -r requirements.txt**

**pip o'rnatish PyQt5**

**python setup.py o'rnatish**

*Mayavi* kompyuteringizda to'g'ri o'rnatilganligini tekshirish uchun oching

konsol va unga buyruqni kiriting:

**mayavi2**

Natijada, qobiq yuklanishi kerak, uning ko'rinishi

13.1-rasmda keltirilgan.

317

|  |
| --- |
| **Sahifa 319** |

**13.1-rasm - grafik interfeysga ega bo'lgan *mayavi2* dasturi**

**uchun *2D / 3D* vizuallashtirish**

318

|  |
| --- |
| **320-bet** |

**13.2 Tez boshlash**

*Mayavi* u bilan ishlashning bir necha usullarini *taklif* qiladi:

1. *mayavi2 GUI* dasturi bilan ishlash *.*

2. Kutubxonadan foydalangan holda *Python* dasturlarini ishlab chiqish

*Mayavi* . Bu erda ikkita variant mavjud:

1. *Python* modullarini ishlab chiqish .

2. O'rnatilgan dvigatel bilan *GUI* dasturlarini ishlab chiqish

*Mayavi* .

3. *Yupyter daftaridan* foydalanish .

Biz o'rnatilgan *Mayavi* dvigateli bilan *GUI* dasturlarini ishlab chiqish

biz ko'rib chiqmaymiz. Muxtasar qilib aytganda, bu imkoniyatning mohiyati

yordamida o'zingizning ilovalaringizni yaratishingiz mumkin

bilan grafik interfeyslarni *Mayavi* ularning ichida ko'milgan elementlar uchun

ma'lumotlarni vizualizatsiya qilish. Shunday qilib, siz echimlarni olishingiz mumkin

aniq vazifalar uchun tayyorlangan. Elementlarni kiritish uchun

*Xususiyatlar* tomonidan ishlatiladigan *Mayavi*[*o'n to'qqiz*](https://translate.googleusercontent.com/translate_f#320)... Buni qanday qilish haqida ko'proq bilib oling

rasmiy hujjatlarda o'qishingiz mumkin [20](https://translate.googleusercontent.com/translate_f#320) .

*Mayavi2* dasturining *GUI* bilan tanishishni boshlaymiz .

***13.2.1 Mayavi2 GUI bilan ishlash***

*Mayavi2-* ni ishga tushirish uchun terminal oching va quyidagilarni kiriting:

**mayavi2**

Natijada, 13.2-rasmda ko'rsatilgan oyna paydo bo'ladi.

19 [https://docs.enthought.com/traits/](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://docs.enthought.com/traits/)

20 [https://docs.enthought.com/mayavi/mayavi/building\_applications.html](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://docs.enthought.com/mayavi/mayavi/building_applications.html)

319

|  |
| --- |
| **Sahifa 321** |

**13.2-rasm - Mayavi2 dastur oynasi elementlari bilan**

Rasmda dastur oynasining asosiy elementlari ko'rsatilgan:

• *Dvigatel daraxtlarini ko'rish;*

• *TVTK sahnasi;*

• *Ob'ekt muharriri;*

• *Python Interactive Shell;*

• *Logger yorlig'i.*

Dastur bilan tajriba o'tkazish uchun to'plamni yuklang

demo ma'lumotlar, buning uchun menyu satrida *File-> Load data ->* -ni tanlang

*Parametrik sirt manbasini yarating* . Shundan so'ng, " *Dvigatel daraxti*

*View* "ma'lumot manbai qo'shiladi:

320

|  |
| --- |
| **Sahifa 322** |

**Shakl 13.3 - *Dvigatel daraxtlari ko'rinishini* ma'lumotlar manbai bilan *ko'rish***

Element daraxtida " *Modul yoki filtr qo'shish* " -ni tanlang , so'ngra " *Ob'ekt*

*Muharriri* "siz ikkiga bo'lingan modullar va filtrlar ro'yxatini ko'rasiz

yorliqlar. " *Vizualizatsiya modullari* " yorlig'idan *Surface* modulini tanlang

(sichqonchaning chap tugmasi bilan elementni ikki marta bosishingiz kerak (rasmga qarang.)

13.4).

**Shakl 13.4 - *Yuzaki* modulni qo'shish**

321

|  |
| --- |
| **Sahifa 323** |

Agar siz hamma narsani to'g'ri bajargan bo'lsangiz, unda sahna ko'rinishi kerak

13.5-rasmdagi kabi tasvir.

**13.5-rasm - yuklangan demo-ma'lumotlar to'plamining vizualizatsiyasi**

" *Dvigatel daraxtlari ko'rinishida* " " *ParametricSurface* " bandiga qaytib , siz qilishingiz mumkin

*3D* modeli qurilgan ma'lumotlar to'plamini o'zgartirish .

322

|  |
| --- |
| **Sahifa 324** |

**Shakl 13.6 - Ma'lumotlar to'plamini o'zgartirish**

Bizning holatimizda *o'g'il bola* varianti tanlandi , uni *diniy qilib* o'zgartiring :

**Shakl 13.7 - sahnada *diniy* ma'lumotlar to'plamining vizualizatsiyasi**

323

|  |
| --- |
| **Sahifa 325** |

Mobius chizig'i misoli ( *mobius* ):

**Shakl 13.8 - Sahnada *mobius ma'lumotlar to'plamining* vizualizatsiyasi**

Modulning yuzasiga kelsak ( *sirt* ), unda u ham bor

tashqi ko'rinishini o'zgartirish uchun ishlatilishi mumkin bo'lgan bir qator parametrlar

raqamning sahnada ko'rinishi. Masalan, ma'lumotlar to'plamini *qaytarib bering* ,

" *Dvigatel daraxtlari ko'rinishida* " *Surface-* ni tanlang , " *Aktyor* " yorlig'iga o'ting

va " *Taqdim etish* " parametrini *sim ramka* qiymatiga o'zgartiring *.*IN

bu sirt turini simli ramkaga o'zgartiradi.

324

|  |
| --- |
| **Sahifa 326** |

**Shakl 13.9 - sirtni namoyish qilish turini o'zgartirish**

**simli ramka**

Natijada siz 13.10-rasmda ko'rsatilgan rasmni olasiz.

**Shakl 13.10 - Parametrli sirt**

***Vakolat = simli ramka***

325

|  |
| --- |
| **Sahifa 327** |

Shunday qilib, *Mayavi2 bilan ishlash* juda oddiy. " *Dvigatel*

*Tree View* "komponentlari quyidagilarda yaratilgan va tuzilgan

buyurtma: ***Ma'lumotlar manbai -> Filtrlar -> Model*** . " *Ob'ekt*

*Muharriri* "komponentlarning parametrlarini o'zgartiradi,

ish natijasi.

***13.2.2 Mayavi yordamida Python modullarini ishlab chiqish***

Matn muharriringizda *demo.py* , in deb nomlangan hujjat yarating

biz kutubxona bilan qanday ishlashni ko'rsatadigan kod yozamiz

*Mayavi* .

Birinchidan, kerakli kutubxonalarni import qilaylik:

np sifatida numpy- ni import qilish

dan mayavi import mlab

Siz numpy haqida ushbu kitobning oldingi qismlaridan bilib oldingiz. Of

Mayavi kutubxonasi, biz mlab modulini import qilamiz. Bu beradi

*Matplotlib* uslubida *2D / 3D* modellarini yaratish vositalari . Siz

siz o'zingizning modulingiz kodida ham mlab-dan foydalanishingiz mumkin

interaktiv muhitda.

Buning uchun *100* dona miqdorida *-5* dan *5 gacha* bo'lgan nuqtalar to'plamini *tuzamiz*

Keling, numpy dan linspace funktsiyasidan foydalanamiz:

t = np.linspace (- 5 , 5 , 100 )

Uch o'lchovli spiral uchun *x* , *y* , *z* koordinatali massivlar yarating

parametrli tenglamalar orqali:

x = 3 \* np.cos (t)

y = 3 \* np.sin (t)

z = t / np.pi

326

|  |
| --- |
| **Sahifa 328** |

*3D* tasvirni yaratish uchun kodni qo'shaylik:

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

mlab.show ()

Sizning *demo.py* faylingizda bo'lishi kerak bo'lgan barcha kodlar :

np sifatida numpy- ni import qilish

dan mayavi import mlab

t = np.linspace (- 5 , 5 , 100 )

x = 3 \* np.cos (t)

y = 3 \* np.sin (t)

z = t / np.pi

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

mlab.show ()

Terminali oching, *demo.py* moduli bilan *katalogga* o'ting va ishga tushiring

Ilova:

**python demo.py**

Natijada, " *Mayavi sahnasi* " oynasi model bilan ochilishi kerak ,

13.11-rasmda keltirilgan.

**Rasm 13.11 - bino misol *3D* modeli *Python* bilan**

***Mayavi* yordamida**

327

|  |
| --- |
| **Sahifa 329** |

*Mayavi sahnasi* - bu siz aylanadigan interaktiv muhit

tasvirlar, izometrik, saqlash va

uni sozlang.

***13.2.3 Mayvi bilan Yupyter daftarida ishlash***

AH ko'rsatish uchun *Mayavi* bir yilda *Jupyter tashrif buyurdingizmi*

uchun ishlatiladigan *backend-ni* belgilashingiz kerak

model ko'rinishlari. Qaysi birini xohlasangiz

*backend-dan* foydalaning , sizga o'rnatishingiz kerak bo'lishi mumkin

qo'shimcha komponentlar.

*"IPy" backend* bilan ishlash uchun sizga tavsiya etiladi (tavsiya etiladi)

*ipywidgets* va *ipyevents* paketlarini o'rnating :

**conda install -c conda-forge ipyevents**

**conda install -c conda-forge ipywidgets**

*Orqa tomoni 'png'* ko'rsatilayotgan natijani statik sifatida ifodalaydi

Tasvirlar. Siz u bilan interaktiv ravishda ishlay olmaysiz.

Yana bir *backend* bo'ladi *"x3d»* tomonidan qo'llab-quvvatlanadi *X3D-*

elementlari, undan foydalanish uchun quyidagi buyruqni bajaring:

**jupyter nbextension install --py mayavi --user**

Barcha shartlarni o'rnatgandan so'ng, *Yupyter-ni* ishga tushiring

*daftar* . Biz numpy va mayavilarni import qilamiz, ularga kerak bo'ladi

misol:

np sifatida numpy- ni import qilish

dan mayavi import mlab

Keyingi muhim qadam - *Mayavini o'tish* orqali sozlash

*jupyter notebooki* bilan ishlayotgan ma'lumotlar, buning uchun sizga kerak

init\_notebook () funktsiyasini chaqiring. Parametrlarsiz chaqirilganda

328

|  |
| --- |
| **330-bet** |

*backend 'ipy'* sukut bo'yicha tanlanadi, agar xohlasangiz

boshqasi, keyin uning nomini argument sifatida kiriting:

mlab.init\_notebook ()

Ushbu buyruqni ishga tushirgandan so'ng, quyidagilarni olishingiz kerak

xabar:

...

X3d backend bilan boshlangan daftar.

Keyingi katakka kodni kiriting:

t = np.linspace (- 5 , 5 , 100 )

x = 3 \* np.cos (t)

y = 3 \* np.sin (t)

z = t / np.pi

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

Keyin s o'zgaruvchisi bo'lgan katakchani bajaring:

**Rasm 13,12 - Misol *mayavi* yilda *Jupyter daftar***

Agar siz *"ipy"* yoki *"x3d" backend-* dan foydalanayotgan bo'lsangiz, u holda model (rasm)

interaktiv bo'ladi, uni aylantirish, masshtablash va h.k.

329

|  |
| --- |
| **Sahifa 331** |

**14-bob. Ko'rinishni sozlash**

*Pythonda* ishlashda *eng yuqori* darajadagi *Mayavi* elementi hisoblanadi

skript - bu *shakl* . Kelajakda sahna tushunchasi ( *Sahna* )

va *shakl* , agar alohida ko'rsatilmagan bo'lsa, tushuniladi

sinonim sifatida. Sahnada *2D / 3D* tasviri chizilgan.

Rasmning o'ziga qo'shimcha ravishda siz atrof-muhitni alohida sozlashingiz mumkin,

koordinata o'qlarini sozlash, sozlash va joylashtirishni o'z ichiga oladi

*ranglar paneli* elementi . Bundan tashqari, siz kamera boshqaruvini yoqishingiz mumkin

sahna. Keling, ushbu vositalarni batafsil ko'rib chiqamiz.

**14.1 Shakl / Sahnani boshqarish**

*Mayavi* tomonidan boshqariladigan barcha vositalar

sahna mayavi.mlab paketida joylashgan.

**Clf () funktsiyasi**

Voqeani tozalaydi.

Funktsiya prototipi:

clf (rasm = yo'q )

Funktsiya parametrlari:

• shakl

◦ Tarkibni tozalash istagan sahna. Agar a

qiymati Yo'q, joriy sahna tozalanadi.

Close **() funktsiyasi**

Sahnani yopadi.

Funktsiya prototipi:

yopish (sahna = Yo'q , barchasi = Yolg'on )

330

|  |
| --- |
| **Sahifa 332** |

Funktsiya parametrlari:

• sahna

Closed yopiladigan shaklning identifikatori bo'lishi mumkin

raqam, satr yoki *Scene* ob'ekti . Agar qiymat

Yo'q, hozirgi sahna yopilmaydi.

• barchasi

◦ Agar parametr rost bo'lsa, unda barchasi mavjud

sahnalar.

Draw **() funktsiyasi**

Sahnani qayta chizishga majbur qiladi.

Funktsiya prototipi:

chizish (rasm = Yo'q )

Funktsiya parametrlari:

• shakl

Red Qayta chiziladigan sahna. Agar qiymat Yo'q bo'lsa,

u holda hozirgi sahna qayta chiziladi.

**Gcf () funktsiyasi**

Joriy sahnaga ko'rsatkichni qaytaradi.

Funktsiya prototipi:

gcf (engin e = yo'q )

Funktsiya parametrlari:

• dvigatel

◦ Agar siz sahna olinadigan dvigateldan foydalanilsa

bir nechta dvigatellar bilan ishlash.

331

|  |
| --- |
| **Sahifa 333** |

**Savefig () funktsiyasi**

Sahna tarkibini tasvir sifatida saqlaydi. Funktsiya

birida rasmni saqlash qobiliyatini ta'minlaydi

quyidagi formatlar: *png, jpg, bmp, tiff, ps, eps, pdf, rib (renderman), oogl*

*(geomview), iv (OpenInventor), vrml, obj (wavefront)* .

Funktsiya prototipi:

savefig (fayl nomi, hajmi = yo'q , raqam = yo'q , kattalashtirish = **'avtomatik'** , \*\* kvargs)

Funktsiya parametrlari:

• Fayl nomi

. Saqlanadigan fayl nomi.

• hajmi

◦ Saqlangan rasm hajmi.

• shakl

◦ Rasm sifatida saqlanadigan sahna.

• kattalashtirish

Screen Ekran tarjimasi uchun ko'lamli omil

saqlangan rasmga.

**Skrinshot () funktsiyasi**

Sahna (rasm) tarkibini massiv sifatida qaytaradi.

Funktsiya prototipi:

skrinshot (rasm = Yo'q , mode = **'rgb'** , antialiased = False )

Funktsiya parametrlari:

• shakl

◦ Rasm olinadigan sahna.

• rejim

◦ Rasm rang rejimi {'rgb', 'rgba'}.

332

|  |
| --- |
| **Sahifa 334** |

• antialiased

◦ Agar qiymat True bo'lsa, u holda skrinshot paydo bo'lganda bo'ladi

anti-lizing ishlatiladi.

**Shakl () funktsiyasi**

Yangi sahna yaratadi yoki hozirgi holatini qaytaradi.

Funktsiya prototipi:

shakl (shakl = Yo'q , bgcolor = Yo'q , fgcolor = Yo'q , dvigatel = Yo'q , o'lcham = ( 400 ,

350 ))

Funktsiya parametrlari:

• shakl

◦ Agar parametr Yo'q bo'lsa, u holda joriy sahna qaytariladi,

aks holda yangisi yaratiladi.

• bgcolor

Scene Sahnaning fon rangi.

• fgcolor

Scene Sahnadagi matn elementlarining rangi.

• dvigatel

◦ *Mayavi shaklidagi* dvigatel .

• hajmi

◦ Sahna hajmi. Odatiy *(400, 350)* .

Sahna bilan ishlashning misollarini ko'rib chiqamiz. Birinchidan, import qilaylik

kerakli paketlar:

np sifatida numpy- ni import qilish

dan mayavi import mlab

plt sifatida import matplotlib.pyplot

333

|  |
| --- |
| **Sahifa 335** |

Keling, allaqachon tanish bo'lgan *3D* koordinatalarini o'z ichiga olgan massiv yarataylik

spirallar:

t = np.linspace (- 5 , 5 , 100 )

x = 3 \* np.cos (t)

y = 3 \* np.sin (t)

z = t / np.pi

Keling, yangi sahna (parametr) uchun och yashil rang fonini o'rnatamiz

bgcolor), matn yorliqlari uchun qizil (fgcolor parametri) va

hajmi *300x300* :

mlab.figure ( **"test\_fig"** , bgcolor = ( 0.56 , 0.93 , 0.56 ), fgcolor = ( 1 , 0 , 0 ),

hajmi = ( 300 , 300 ))

Keling , *3D* spiral quramiz va sahnani namoyish etamiz:

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

mlab.title ( **"Sinov rasm"** )

mlab.show ()

Natijada biz quyidagi sahnani olamiz:

**14.1-rasm - *Mayavi-ning* konfiguratsiya imkoniyatlarini namoyish etish**

**sahnalar**

334

|  |
| --- |
| **Sahifa 336** |

Agar siz sahnani rasm sifatida saqlamoqchi bo'lsangiz, unda qo'shishingiz kerak

mlab.show () funktsiyasidan oldin yoki uning o'rniga, agar xohlamasangiz

sahna ekranda ko'rsatildi, satr:

mlab.savefig ( **"test.png"** )

Sahna tarkibini rasm sifatida massiv sifatida olish mumkin

keyingi ishlov berish uchun. Qatorni olib tashlang yoki izoh bering:

mlab.show ()

Quyidagi kodni qo'shing:

f = mlab.gcf ()

f.scene.\_lift ()

img = mlab.screenshot ()

plt.imshow (img)

plt.show ()

Skrinshot funktsiyasi uchun dastlabki ikkita satr kerak

yaxshi ishladi. Ushbu yozuv paytida *Mayavida* xatolik yuz *berdi* ,

ularsiz skrinshot olishga imkon bermagan. Vizualizatsiya uchun

tasvir *Matplotlib kutubxonasidagi* imshow () funktsiyasidan foydalanadi . IN

Natijada biz 14.2-rasmda ko'rsatilgan oynani olamiz.

335

|  |
| --- |
| **Sahifa 337** |

**14.2-rasm - Tasvir sifatida model**

**14.2 Dizayn elementlarini sozlash**

Dizayn elementlari deganda biz grafik elementlarni,

to'g'ridan-to'g'ri grafaning o'zi emas. Ularga

sarlavha, koordinata o'qlari, yorliqlar, *rang paneli* elementlari va boshqalarni o'z ichiga oladi . Hammasi

dizayn elementlarini boshqarish funktsiyalari joylashgan

paketli mlab.

Funktsiyalar ro'yxati 14.1-jadvalda keltirilgan.

336

|  |
| --- |
| **Sahifa 338** |

**14.1-jadval - Dizayn elementlarini boshqarish funktsiyalari**

**Funktsiya**

**Tavsif**

sarlavha ()

Sahna nomi.

kontur ()

Modelning tashqi ko'rinishi.

o'qlar ()

Koordinata o'qlarining ko'rinishini boshqaradi.

xlabel ()

X o'qi yorlig'i.

ylabel ()

Y o'qi yorlig'i.

zlabel ()

Z o'qi yorlig'i.

rang paneli ()

Rangli chiziq.

**14.2.1 Sahna sarlavhasi**

Tanishimizni sahna nomi bilan boshlaymiz. Biz u bilan allaqachon uchrashganmiz

shaklni yaratishda va sozlashda oldingi qismda.

Sarlavhani boshqarish funktsiyasining prototipi:

sarlavha (\* args, \*\* kwargs)

Funktsiya parametrlari:

• rang

◦ Sarlavha rangi, har biri uchta elementdan tashkil topgan strelka sifatida ko'rsatilgan

ulardan 0 dan 1 gacha bo'lgan raqam.

• shakl

◦ Sarlavha ko'rsatiladigan sahna. Agar parametr bo'lsa

Hech kimga teng emas, sarlavha joriy sahnada ko'rsatiladi.

• balandlik

◦ Sarlavha qo'yilishi kerak bo'lgan balandlik o'rnatiladi

rasm o'lchamining mutanosibligi sifatida.

• chiziq\_ kengligi

Of Belgilarning satr kengligi. Standart qiymat: 2.

337

|  |
| --- |
| **Sahifa 339** |

• xiralik

◦ Shaffoflik, sukut bo'yicha 1.

• hajmi

◦ Sarlavha hajmi.

• ism

◦ Yaratiladigan ob'ekt nomi.

Sarlavha bilan ishlashga misol:

np sifatida numpy- ni import qilish

dan mayavi import mlab

t = np.linspace (- 5 , 5 , 100 )

x = 3 \* np.cos (t)

y = 3 \* np.sin (t)

z = t / np.pi

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

mlab.title ( **"Sinov** anjomi **"** , rang = ( 0 , 1 , 0 ), balandlik = 0,1 , xiralashganlik = 0,5 , o'lcham = 1,5 )

mlab.show ()

**14.3-rasm - title () funktsiyasi bilan ishlashni namoyish etish**

338

|  |
| --- |
| **340-bet** |

**14.2.2 Modelning tashqi konturi**

Sahnada siz buning uchun konturni model atrofida joylashtirishingiz mumkin

anahat () funktsiyasi ishlatiladi:

kontur (\* args, \*\* kwargs)

Funktsiya parametrlari:

• rang

◦ Har biri uchta elementdan tashkil topgan katakcha sifatida ko'rsatilgan chiziq rangi

bu 0 dan 1 gacha bo'lgan raqam.

• daraja

◦ Kontur geometriyasi [xmin, xmax, ymin, ymax, zmin, zmax],

modelning o'lchamlari sukut bo'yicha olinadi.

• shakl

◦ Tashqi kontur ko'rsatiladigan sahna. Agar a

parametr Yo'q, keyin yorliq joriy ko'rsatiladi

sahna.

• chiziq\_ kengligi

◦ Chiziq kengligi. Standart qiymat: 2.

• ism

◦ Yaratiladigan ob'ekt nomi.

• xiralik

◦ Shaffoflik, sukut bo'yicha 1.

Misol bilan ishlashni ko'rib chiqamiz (). Kerakli ma'lumotlarni import qilish uchun kod

kutubxonalar va koordinatalar massivini yarating: *t, x, y, z* emas

biz buni " *14.2.1 Sahna sarlavhasi* " bo'limidan olishingiz mumkin . Quyida

*3D* spiral atrofida ramka chizish dasturi taqdim etilgan :

339

|  |
| --- |
| **Sahifa 341** |

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

mlab.outline (rang = ( 0,9 , 0,9 , 0,9 ), chiziq\_ kengligi = 3 , xiralik = 0,5 )

mlab.show ()

Natijada 14.4-rasmda ko'rsatilgan sahna quriladi.

**14.4-rasm - anahat () funktsiyasi bilan ishlashni namoyish etish**

**14.2.3 Koordinata o'qlarini o'rnatish**

Funktsiya koordinata o'qlarining ko'rinishini boshqarish uchun javobgardir

o'qlar (), o'q belgilarini xlable (), ylabel (),

zlabel (). Keling, ularni batafsil ko'rib chiqaylik.

**Axes () funktsiyasi**

Koordinata o'qlarining ko'rinishini boshqaradi.

Funktsiya prototipi:

o'qlar (\* args, \*\* kwargs)

Funktsiya parametrlari:

340

|  |
| --- |
| **Sahifa 342** |

• rang

• Rang, har biri uchta elementdan tashkil topgan

0 dan 1 gacha bo'lgan raqam.

• daraja

• Geometriya [xmin, xmax, ymin, ymax, zmin, zmax], by

modelning o'lchamlari sukut bo'yicha olinadi.

• shakl

• Koordinata o'qlari ko'rsatiladigan sahna. Agar a

parametr Yo'q, keyin yorliq joriy ko'rsatiladi

sahna.

• chiziq\_ kengligi

• Chiziq kengligi. Standart qiymat: 2.

• ism

• Yaratiladigan ob'ekt nomi.

• nb\_labels

• O'qlarning har bir yo'nalishi bo'yicha yorliqlar soni.

• xiralik

• Shaffoflik, standart qiymat 1 ga teng.

• diapazonlar

• O'qlarda ko'rsatilgan yorliqlar qatori: [xmin, xmax, ymin,

ymax, zmin, zmax], model o'lchamlari sukut bo'yicha olinadi.

• x\_axis\_visibility

• *x* o'qi aks etadimi yoki yo'qmi *.*

• xlabel

• *x* o'qining imzosi .

• y\_axis\_visibility

• *Y* o'qi *aks etadimi* yoki yo'qmi *.*

341

|  |
| --- |
| **Sahifa 343** |

• ylabel

• *Y* o'qi *yorlig'i* .

• z\_axis\_visibility

• *z* o'qini aks ettiring yoki ko'rsatmang *.*

• zlabel

• *Z* o'qi imzosi *.*

Axes () funktsiyasi bilan ishlashga misol, model yaratish uchun ma'lumotlar

" *14.2.1 Sarlavha* " bo'limida keltirilgan misoldan olinishi kerak

*sahnalar* ":

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

mlab.axes (color = ( 0 , 1 , 0 ), nb\_labels = 5 , intervallar = [ 0 , 10 , 0 , 10 , 0 , 10 ],

z\_axis\_visibility = False , **ylabel** = **"y\_ax"** , **zlabel** = **"z\_ax"**

)

mlab.show ()

**14.5-rasm - o'qlar () funktsiyasi bilan ishlashga misol**

342

|  |
| --- |
| **Sahifa 344** |

Koordinata o'qlari uchun yorliqlarni o'rnatish uchun siz foydalanishingiz mumkin

xlabel (), ylable (), zlable () funktsiyalari. Ro'yxatdagi funktsiyalar

biz keltiradigan misol sifatida bir xil argumentlar to'plamiga ega bo'ling

xlabel () prototipi:

xlabel (matn, ob'ekt = Yo'q )

Funktsiya parametrlari:

• matn

◦ Eksa yorlig'i.

• ob'ekt

◦ Ushbu parametr qo'llaniladigan ob'ekt.

Xlabel (), ylable (), zlable () funktsiyalari bilan ishlashga misol:

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

mlab.xlabel ( **"x\_ax"** )

mlab.ylabel ( **"y\_ax"** )

mlab.zlabel ( **"z\_ax"** )

mlab.show ()

**14.6-rasm - xlabel (), ylable () funktsiyalari bilan ishlashga misol,**

**zlable ()**

343

|  |
| --- |
| **Sahifa 345** |

**14.2.4 Rang satrini ( *colorbar* ) sozlash**

Sahnada rang satrini joylashtirish uchun funktsiyadan foydalaning.

colorbar () rang satri kerakligini aniq ko'rsatmoqchi bo'lsangiz

skalar ma'lumotlari uchun, keyin vektor ma'lumotlari uchun skalarbar () dan foydalaning

vektor paneli ().

Funktsiya prototipi:

colorbar (ob'ekt = Yo'q , sarlavha = Yo'q , yo'nalish = Yo'q , nb\_labels = Yo'q ,

nb\_colors = Yo'q , label\_fmt = Yo'q )

Funktsiya parametrlari:

• ob'ekt

Color Ranglar paneli quriladigan ob'ekt.

• sarlavha

The Colorbar elementining sarlavhasi.

• yo'nalish

Entation Yo'nalish: 'gorizontal' - gorizontal, 'vertikal' -

vertikal.

• nb\_lables

◦ Rangda ko'rsatiladigan yorliqlar soni

Ip.

• label\_fmt

◦ Yorliq matni dizayni shabloni.

• nb\_colors

◦ Ko'rsatiladigan ranglarning maksimal soni

rang paneli.

344

|  |
| --- |
| **Sahifa 346** |

Colorbar () dan foydalanish misoli:

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

mlab.colorbar (s, title = **"Bar v1"** , orientation = **"vertikal"** ,

nb\_labels = 5 , label\_fmt = **"% .1f"** )

mlab.show ()

**14.7-rasm - colorbar () funktsiyasi bilan ishlashga misol**

Ranglar *panelidagi* ranglar sonini *cheklaylik* :

s = mlab.plot3d (x, y, z, z, tube\_radius = 0,1 )

mlab.colorbar (s, title = **"Bar v2"** , orientatsiya = **"vertikal"** , nb\_labels = 5 ,

label\_fmt = **"% .1f"** , nb\_colors = 10 )

mlab.show ()

345

|  |
| --- |
| **347-bet** |

**Shakl 14.8 - elementdagi ranglar sonini cheklash**

**rang paneli**

**14.3 Kamerani boshqarish**

*Mayavi* pozitsiyani boshqarish uchun bir qator vositalarni taqdim etadi

*3D* sahna *ko'riladigan* kamera , funktsiyalar ro'yxati

14.2-jadvalda keltirilgan.

**Jadval 14.2 - Kamerani boshqarish funktsiyalari**

**Funktsiya**

**Tavsif**

harakat qilish

Kamera va fokusni harakatga keltiradi.

balandlik

Kamerani mos keladigan o'q atrofida aylantiradi

"To'g'ri" yo'nalishga.

rulon

Kameraning mutlaq burilish burchagini o'rnatadi / qaytaradi.

ko'rinish

Kamera ko'rinishini o'rnatadi / qaytaradi.

346

|  |
| --- |
| **Sahifa 348** |

yaw

Kamerani mos keladigan o'q atrofida aylantiradi

Yuqoriga yo'nalish.

Ushbu barcha kameralarni boshqarish funktsiyalari haqida ko'proq ma'lumot olish uchun siz qilishingiz mumkin

rasmiy hujjatlarda o'qing [21](https://translate.googleusercontent.com/translate_f#348)...

Masalan, view () funktsiyasi bilan ishlashni ko'rib chiqing. Funktsiya prototipi:

ko'rinish (azimut = Yo'q , balandlik = Yo'q , masofa = Yo'q , fokal markaz = Yo'q ,

roll = Yo'q , reset\_roll = To'g'ri , raqam = Yo'q )

Funktsiya parametrlari:

• azimut

Degrees Azimut, darajalarda ko'rsatilgan (0 dan 360 gacha).

• balandlik

Degrees darajalarda ko'rsatilgan ko'tarilish (0 dan 180 gacha).

• masofa

Focus Fokusdan kameraga masofa. Agar siz qiymatni qo'ysangiz

"avtomatik", eng yaxshi qiymat topiladi.

• fokalpoint

◦ Kamera fokusi. Agar siz "avtomatik" qiymatini qo'ysangiz, u holda bo'ladi

eng yaxshi qiymat tanlanadi.

• rulon

◦ Kamera o'qi atrofida aylanishini o'rnatadi.

• reset\_roll

◦ Agar qiymat True bo'lsa, unda roll parametri e'tiborga olinmaydi.

• shakl

◦ Kamera boshqariladigan sahna.

21 [https://docs.enthought.com/mayavi/mayavi/auto/mlab\_camera.html](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://docs.enthought.com/mayavi/mayavi/auto/mlab_camera.html)

347

|  |
| --- |
| **349-bet** |

View () funktsiyasi bilan ishlashga misol:

v = mlab.view ()

mlab.view (azimut = 0 , balandlik = 0 , masofa = 20 , fokalpoint = [ 1 , 0 , 0 ])

mlab.show ()

Natijada biz quyidagi rasmni olamiz:

**14.9-rasm - view () funktsiyasi bilan ishlashga misol**

348

|  |
| --- |
| **Sahifa 350** |

**15-bob. Ma'lumotlarni vizuallashtirish**

Ushbu bo'limda ishlashning tavsifi va misollari keltirilgan

*Mayavi* tomonidan taqdim etilgan *2D / 3D* ko'rsatish uchun funktsiyalar . IN

keyingi qism: " *16. bob. Mayavi vositalari* "

yordamida bu muammoni hal qilishning umumiy yondashuvi

ma'lumotlarni qayta ishlash quvurlari.

*Mayavining* ma'lumotlarni vizualizatsiya qilish uchun barcha funktsiyalarini quyidagilarga bo'lish mumkin

ular ishlaydigan ma'lumotlar massivlarining o'lchamlari, ya'ni

bir o'lchovli ( *1D* ) vektorlar, ikki o'lchovli ( *2D* ) ma'lumotlar jadvallari va bo'lishi mumkin

uch o'lchovli ( *3D* ) to'plamlar. Bu erda keltirilgan barcha funktsiyalar

qismi mlab paketida joylashgan.

Funktsiyalarni ko'rsatish uchun umumiy parametrlar:

• rang

◦ Rang. Tople sifatida barcha elementlar uchun bitta rangni o'rnatadi

uchta elementdan iborat bo'lib, ularning har biri 0 dan oralig'idagi raqamdir

1 ga qadar.

• kolormap

◦ Ranglar palitrasi. Agar s yoki f ishlatilsa, u holda rang

elementlar ularning qiymatlari asosida tanlanadi.

• daraja

◦ Model hajmi. Ro'yxat sifatida ko'rsatilgan [xmin, xmax, ymin,

ymax, zmin, zmax]. Odatiy o'lcham: dan

x, y, z massivlari.

• shakl

◦ Model joylashtiriladigan bosqich.

349

|  |
| --- |
| **Sahifa 351** |

• chiziq\_ kengligi

◦ Chiziq kengligi. Standart qiymat: 2.0.

• ism

◦ Model nomi.

• xiralik

◦ oshkoralik.

• reset\_zoom

Sc Masshtabni tiklash.

• vmax, vmin

◦ Rang shkalasi uchun maksimal va minimal qiymatlar.

**15.1 Bir o'lchovli to'plamlar bilan ishlash funktsiyalari**

**ma'lumotlar**

Bir-o'lchovli Ma'lumotlarni guruhlar tasavvur qilish *3D* kosmik

15.1-jadvalda keltirilgan funktsiyalar mavjud.

**15.1-jadval - *Mayavi* bir o'lchovli funktsiyalar**

**ma'lumotlar to'plamlari**

**Funktsiya**

**Tavsif**

ball3d

Ma'lumotlar to'plamini ball sifatida ko'rish.

fitna3d

Ma'lumotlar to'plamining chiziqli vizualizatsiyasi.

quiver3d

Ma'lumotlar to'plamini vektor maydoni sifatida ingl

(otuvchi).

350

|  |
| --- |
| **Sahifa 352** |

**15.1.1 ballar3d () funktsiyasi**

Points3d () funktsiyasi maxsus rasm sifatida aks etadi

(gliflar) koordinatalari x, y va

z. 15-bobda keltirilgan *ma'lumotlarga* qo'shimcha ravishda *ma'lumotlar vizualizatsiyasi,* uchun

Points3d () funktsiyasi quyidagi parametrlarga ega:

• x, y, z: numpy qatori, ro'yxat

Points Ballar koordinatalari.

• s: *numpy* -array, ro'yxat

Size x, y, z bilan bir xil bo'lishi kerak bo'lgan massiv. Tarkibida

har bir nuqtani tavsiflovchi raqamlar. Foydalanish mumkin

gliflarning rangini yoki hajmini boshqarish uchun.

• f: callabel

◦ xarakteristikani qaytaradigan f (x, y, z) funktsiyasi

asl ma'lumotlar to'plamidagi har bir nuqta uchun raqam. Mumkin

gliflarning rangini yoki hajmini boshqarish uchun ishlatiladi.

• mask\_points

◦ o'tgan x to'plamlardan ma'lumotlarni olish bosqichini aniqlaydi,

y, z. Ko'rsatilgan sonni kamaytirish uchun ishlatiladi

katta ma'lumotlar to'plami bilan ishlashda elementlar.

• rejim

◦ Glif uslubini o'rnatadi: '2darrow', '2dcircle', '2dcross',

'2ddash', '2ddiamond', '2dhooked\_arrow', '2dsquare',

'2dthick\_arrow', '2dthick\_cross', '2dtriangle',

'2dvertex', 'o'q', 'o'qlar', 'konus', 'kub',

'silindr', 'nuqta', 'shar'

• qaror

◦ Glif o'lchamlari.

351

|  |
| --- |
| **Sahifa 353** |

Points3d () bilan ishlashning misollarini ko'rib chiqamiz. Kerakli narsalarni yuklang

kutubxonalar va tajribalar uchun ma'lumotlar bazasini yaratish:

np sifatida numpy- ni import qilish

dan mayavi import mlab

t = np.linspace ( 0 , 2 \* pi, 15 )

x = 3 \* np.cos (t)

y = 3 \* np.sin (t)

z = 0 \* t

s = 2 \* np.cos (t)

Biz ma'lumotlar bazasini *3D* maydonidagi sharlar sifatida tasavvur qilamiz :

mlab.points3d (x, y, z)

mlab.show ()

**15.1-rasm - points3d () funktsiyasini namoyish etish**

352

|  |
| --- |
| **Sahifa 354** |

Agar s qatorini to'rtinchi parametr sifatida o'tkazsangiz, u holda qiymatlar

undan gliflarning rangi va hajmini aniqlash uchun foydalaniladi:

mlab.points3d (x, y, z, s)

**15.2-rasm -** s **funktsiya parametri bilan ishlashni namoyish etish**

**ball3d ()**

S o'rniga siz f dan foydalanishingiz mumkin:

f\_attr = lambda x, y, z: np.array ([ 2 \* np.random.rand () uchun \_ yilda

qator (len (x))])

mlab.points3d (x, y, z, f\_attr)

353

|  |
| --- |
| **Sahifa 355** |

**15.3-rasm - funktsiyaning** f **parametri bilan ishlashni namoyish etish**

**ball3d ()**

Barcha gliflarga bir xil rang bering, shuningdek qiymatini o'zgartiring

rezolyutsiya parametri , avval biz unga 4, keyin 16,

kod 4-variant uchun berilgan:

mlab.points3d (x, y, z, rang = ( 0 , 1 , 0 ), rezolyutsiya = 4 )

**15.4-rasm - Qaror parametri bilan ishlashni namoyish etish**

**points3d () funktsiyalari**

354

|  |
| --- |
| **Sahifa 356** |

Ikkinchi variantda gliflarning sifatiga e'tibor bering (bilan

qaror parametrining katta qiymati ) . Keling, ishlatilganlarni o'zgartiraylik

rang palitrasi, glif ko'rinishi va shaffofligi:

mlab.points3d (x, y, z, s, colormap = **"plazma"** , mode = **'ballon "** ,

shaffoflik = 0,7 )

**15.5-rasm - colormap parametrlari bilan ishlashni namoyish etish** ,

**mod** , **points3d opaklığı () funktsiyasi**

**15.1.2 funktsiyasi plot3d ()**

Plot3d () funktsiyasi o'tgan bir o'lchovli massivlardan chiziqlarni chizadi

*3D* kosmosdagi ma'lumotlar . Bobning boshida sanab o'tilganlardan tashqari

umumiy parametrlar, plot3d () bir qator noyob argumentlarga ega:

• x, y, z: numpy qatori, ro'yxat

Points Ballar koordinatalari.

• vakillik

◦ Model sirtining turi: 'sirt', 'simli ramka', 'nuqtalar'.

355

|  |
| --- |
| **Sahifa 357** |

• tube\_radius

◦ Chiziqni ko'rsatish uchun ishlatiladigan quvur radiusi.

• siljishlar

◦ quvur komponentlar soni uchun ishlatiladigan

chiziq ko'rinishlari. Odatiy qiymati 6 ga teng.

Ish uchun ma'lumotlar to'plami:

t = np.linspace ( 0 , 5 \* np.pi, 100 )

x = t \* np.cos (t)

y = t \* np.sin (t)

z = t

Plot3d () yordamida oddiy modelni yarataylik:

s = mlab.plot3d (x, y, z)

mlab.show ()

**15.6-rasm - plot3d () funktsiyasini namoyish etish**

356

|  |
| --- |
| **Sahifa 358** |

Qanday qilib qo'shimcha ravishda ko'rsatib o'ting, radiusi 0,5 ga teng quvur shaklida model quramiz

parametr s, quvurning turli uchastkalari uchun rangni belgilash uchun

kamalak rang kartasini tanlang:

s = mlab.plot3d (x, y, z, z , colormap = **"kamalak"** , tube\_radius = 0,5 ,

tube\_sides = 3 )

**15.7-rasm - colormap parametrlari bilan ishlashni namoyish etish,**

**tube\_radius, plot3d () funktsiyasining tube\_sides**

Tube\_sides parametri, yuqorida aytib o'tilganidek, javobgardir

quvur yuzlari soni, ushbu misolda biz uchburchak shaklini yaratdik

quvur. Buni 15.8-rasmda ko'rsatilgan nuqtai nazardan yaqqol ko'rish mumkin.

357

|  |
| --- |
| **Sahifa 359** |

**Shakl 15.8 - Quvurlarning tuzilishi**

Tube\_sides parametrini 6 ga o'rnatamiz, endi quvur bor

olti burchakli shakli:

**15.9-rasm - tube\_sides = 6 parametri bilan ishlashni namoyish etish**

**vazifalari plot3d ()**

358

|  |
| --- |
| **Sahifa 360** |

Keling, trubaning hajmini oshiramiz va sirt uslubini simli ramkaga o'zgartiramiz:

s = mlab.plot3d (x, y, z, z, tube\_radius = 1,5 , tube\_sides = 10 ,

vakillik = **'simli ramka'** )

**15.10-rasm - parametrlar bilan ishlashni namoyish etish**

**plot3d () funktsiyasining vakili = 'simli ramka'**

**15.2 Ikki o'lchovli to'plamlar bilan ishlash funktsiyalari**

**ma'lumotlar**

*3D* maydonida 2 o'lchovli ma'lumotlar to'plamini yaratish

15.2-jadvalda keltirilgan funktsiyalar mavjud.

**Jadval 15.2 - 2D ko'rsatish uchun *Mayavi* funktsiyalari**

**ma'lumotlar to'plamlari**

**Funktsiya**

**Tavsif**

imshow

Ma'lumotlar to'plamini rasm sifatida taqdim etadi.

bemaqsad

O'tkazilgan *2 o'lchovli* balandlikdan sirtni quradi.

359

|  |
| --- |
| **Sahifa 361** |

kontur\_surf

Dan har bir balandlik darajasi uchun uchastkalarning kontur chiziqlari

uzatiladigan *2 o'lchovli* ma'lumotlar qatori.

mash

O'tkazilgan *2 o'lchovli* massivlardan sirt hosil qiladi

ma'lumotlar.

**15.2.1 imshow () funktsiyasi**

Imshow () funktsiyasi unga uzatilgan ikki o'lchovli massivni aks ettiradi

tasvir sifatida.

" *Ma'lumotlarni vizuallashtirish* " 15-bobida ko'rsatilganlardan tashqari

imshow () quyidagi argumentlar to'plamiga ega:

• interpolatsiya qilish

◦ Agar qiymat To'g'ri bo'lsa, u holda interpolatsiya rejimi yoqiladi.

Vizualizatsiya uchun ma'lumotlar bazasini yarataylik:

x, y = np.mgrid [- 2 : 2 : 0,1 , - 2 : 2 : 0,1 ]

z = np.cos (x \* y) \* np.sin (x \* y)

Keling, z matritsasini rasm sifatida namoyish etamiz:

mlab.imshow (z)

mlab.show ()

360

|  |
| --- |
| **Sahifa 362** |

**15.11-rasm - imshow () funktsiyasini namoyish etish**

Interpolatsiya rejimini o'chiring va ranglar palitrasini o'zgartiring:

mlab.imshow (z, interpolate = False , colormap = **'qish'** )

mlab.show ()

**15.12-rasm - Interpolat parametrlari bilan ishlashni namoyish etish**

**va colormap imshow () funktsiyalari**

361

|  |
| --- |
| **Sahifa 363** |

**15.2.2 surf () funktsiyasi**

Funktsiya sirtni o'tgan *2D* balandlik massividan hosil qiladi .

Sörf () ning eng ko'p ishlatiladigan variantlari

quyidagi ko'rinish:

bemaqsad (lar, ...)

bemaqsad (x, y, s, ...)

bemaqsad (x, y, f, ...)

X, y, s, f argumentlari pozitsion (ya'ni, ular bo'lishi mumkin emas)

parametr nomi bilan o'tish). S parametri *2D-*

har bir elementi balandlik qiymati bo'lgan ma'lumotlar to'plami

yuzalar; x, y - *1D* yoki *2D* bo'lishi mumkin , ular

s massivini indekslash uchun ishlatiladi; to'g'ridan-to'g'ri *2D-* o'rniga

ma'lumotlar bilan bir qatorda, siz to'plamlar uchun f funktsiyasini topshirishingiz mumkin

x va y koordinatalari sirt sathining balandlik qiymatlarini beradi.

Surf () ning qolgan argumentlari nomlangan. IN

bobning boshida ko'rsatilganlarga qo'shimchalar (qarang. " *15-bob. Vizualizatsiya*

*ma'lumotlar* ") funktsiyasi quyidagi parametrlarni ta'minlaydi:

• niqob

Elements Dastlabki elementlarni bostirish uchun bool qiymatlari massivi

ma'lumotlar to'plami.

• vakillik

◦ Yuzaki turi "sirt", "simli ramka" yoki bo'lishi mumkin

"ochkolar".

• warp\_scale

◦ Z o'qi miqyosi. Agar qiymat "avtomatik" bo'lsa

ta'minlash uchun shkala qiymati tanlanadi

eng yaxshi vizual taqdimot.

362

|  |
| --- |
| **Sahifa 364** |

Biz quyidagi ma'lumotlar to'plami bilan ishlaymiz:

x, y = mgrid [- 2 : 2 : 0.1 , - 2 : 2 : 0.1 ]

z = 5 \* cos (x \* y) \* sin (x \* y)

Keling, sirtni quramiz:

mlab.surf (z)

mlab.show ()

**15.13-rasm - surf () funktsiyasini namoyish etish**

Surf () ga z ga qo'shimcha ravishda x va y massivlarini o'tkazamiz:

mlab.surf (x, y, z)

mlab.show ()

15.14-rasmda ko'rsatilgan quyidagi rasmni olamiz.

363

|  |
| --- |
| **Sahifa 365** |

**15.14-rasm - x, y, z parametrlari bilan ishlashni namoyish etish**

**surf () funktsiyalari**

Bu faqat bitta qatorni topshirganimizda sodir bo'ldi

z, koordinatalarning qiymati sifatida (x, y), bu indekslar

qator, ikkinchi versiyada, z ga qo'shimcha ravishda uzatildi

oldindan tuzilgan x va y massivlari. Eslatma,

ushbu massivlardagi qo'shni qiymatlar orasidagi qadam 1 emas, balki 0,1 ga teng

birinchi versiyada edi, shuning uchun balandliklar va vodiylar ko'proq narsalarga ega

aniq ko'rinish.

Sirt uchun parametr yordamida bitta rang o'rnatishingiz mumkin

rang. Vizual taqdimotni sozlash uchun ham foydali

z o'qi o'lchovini boshqaradigan warp\_scale parametri.

Biz qachonki oldingi misoldan sirt balandligini solishtiring

warp\_scale = 'auto' da parametr bilan x va y o'tdi:

mlab.surf (x, y, z, color = ( 0 , 0 , 1 ), warp\_scale = **'auto'** )

364

|  |
| --- |
| **Sahifa 366** |

**15.15-rasm - rang parametrlari bilan ishlashni namoyish etish,**

**surf () funktsiyasining warp\_scale**

Paletani "bahor" ga o'zgartiring, sirt turini "simli ramka" ga o'rnating,

shaffoflikni 50% ga sozlang:

mlab.surf (z, colormap = **'bahor "** , shaffoflik = 0,5 , vakillik = **' simli ramka '** ,

warp\_scale = **'avtomatik'** )

**15.16-rasm - Shaffoflik parametrlari bilan ishlashni namoyish etish,**

**surf () funktsiyasining namoyishi va kolormapi**

365

|  |
| --- |
| **Sahifa 367** |

**15.2.3 contour\_surf () funktsiyasi**

Contour\_surf () funktsiyasi, ilgari muhokama qilingan surf () ga o'xshaydi

Undan farqli o'laroq, u sirtni emas, balki kontur chiziqlarini quradi

uzatilgan *2D* ma'lumotlar to'plamidan har bir balandlik darajasi .

Funktsional qo'ng'iroq variantlari:

kontur\_surf (lar, ...)

kontur\_surf (x, y, s, ...)

kontur\_surf (x, y, f, ...)

X, y, s, f parametrlari pozitsion, maqsadga muvofiq keladi

bemaqsad uchun o'xshashlari bilan (). Contour\_surf () funktsiyasini qo'llab-quvvatlaydi

umumiy to'plamdagi parametrlar (" *Ma'lumotlarni vizualizatsiya* " 15-bobga qarang ) va

ularga qo'shimcha:

• warp\_scale

◦ z o'qi bo'ylab masshtab, surf () dan farqli o'laroq, bunday emas

"avtomatik" qiymatini qo'llab-quvvatlaydi.

• konturlar

Display Ko'rsatiladigan konturlar ro'yxati.

Biz allaqachon tanish bo'lgan ma'lumotlar to'plami bilan ishlaymiz:

x, y = np.mgrid [- 2 : 2 : 0,1 , - 2 : 2 : 0,1 ]

z = np.cos (x \* y) \* np.sin (x \* y)

Keling, kontur yuzasini quramiz:

mlab.contour\_surf (x, y, z)

mlab.show ()

366

|  |
| --- |
| **Sahifa 368** |

**15.17-rasm - contour\_surf () funktsiyasini namoyish etish**

Parametr yordamida ko'rsatilgan konturlar sonini ko'paytiramiz

konturlar , palitrani o'zgartiring va z o'qi uchun o'lchovni o'rnating:

mlab.contour\_surf (z, colormap = **'kuz'** , konturlar = 11 , warp\_scale = 10 )

**15.18-rasm - konturlar parametrlari bilan ishlashni namoyish etish,**

**colormap va warp\_scale funktsiyalari contour\_surf ()**

367

|  |
| --- |
| **Sahifa 369** |

**15.2.4 mesh () funktsiyasi**

Mesh () funktsiyasi o'tgan *2D* massivlaridan sirt hosil qiladi

ma'lumotlar. Funktsiya imzosi quyidagicha:

mash (x, y, z, ...)

Bu erda x, y, z - qurish uchun ishlatiladigan *2 o'lchovli* ma'lumotlar to'plamlari

sirt. Ulardan keyin siz boshida ko'rsatilgan parametrlarni qo'shishingiz mumkin

boblar yoki quyidagi ro'yxatdan (ro'yxat to'liq emas):

• vakillik

◦ Yuzaki turi, quyidagilar bo'lishi mumkin: 'sirt', 'simli ramka',

"ochko", "mash" yoki "xayolparast". Agar "ball" turi tanlansa,

unda siz gliflar bilan ishlash parametrlaridan foydalanishingiz mumkin.

• rejim

◦ Glif turi.

• o'lchov\_faktor

◦ o'lchov omili.

Torus sirtini belgilaydigan ma'lumotlar to'plamini yarataylik:

u, v = np.mgrid [-np.pi: np.pi / 2 : 0.05 , -np.pi: np.pi + 0.1 : 0.1 ]

x = np.cos (u) \* (np.cos (v) + 3 )

y = np.sin (u) \* (np.cos (v) + 3 )

z = np.sin (v)

Mesh () funktsiyasi yordamida sirtni quramiz:

mlab.mesh (x, y, z)

mlab.show ()

368

|  |
| --- |
| **Sahifa 370** |

**15.19-rasm - mesh () funktsiyasini namoyish etish**

Rang sxemasini va sirt turini 'simli ramka' ga o'zgartiraylik:

mlab.mesh (x, y, z, colormap = **"bahor"** , vakillik = **"simli ramka"** )

**15.20-rasm - parametr bilan ishlashni namoyish etish**

**mesh () funktsiyasining vakili = 'simli ramka'**

369

|  |
| --- |
| **Sahifa 371** |

"Mesh" sirt opsiyasi:

mlab.mesh (x, y, z, colormap = **"yoz"** , vakillik = **"mesh"** )

**15.21-rasm - parametr bilan ishlashni namoyish etish**

**vakillik = 'mesh' funktsiyasi mesh ()**

**15.3 3D to'plamlar bilan ishlash funktsiyalari**

**ma'lumotlar**

3D ma'lumotlar bazasini *3D* bo'shliqda tasavvur qilish uchun

15.3-jadvalda keltirilgan funktsiyalar mavjud.

**15.3-jadval - 2D ko'rsatish uchun *Mayavi* funktsiyalari**

**ma'lumotlar to'plamlari**

**Funktsiya**

**Tavsif**

kontur3d

O'tkazilgan *3D* to'plamidan izosurflar hosil qiladi

ma'lumotlar

quiver3d

O'tkazilgan koordinatalardan vektor maydonini quradi

vektorlar.

volume\_slice O'tkazilgan *3D* ma'lumotlar to'plamlari uchun tilimlarni yaratadi .

370

|  |
| --- |
| **Sahifa 372** |

3D to'plam funktsiyalari uchun umumiy parametrlar

ma'lumotlar:

• rang

◦ Rang. Bitta rangni uchta elementning katakchasi sifatida o'rnatadi,

ularning har biri 0 dan 1 gacha bo'lgan raqamlar.

• kolormap

◦ Ranglar palitrasi. Agar s yoki f ishlatilsa, u holda rang

elementlar ularning qiymatlari asosida tanlanadi.

• daraja

◦ Model hajmi. Ro'yxat sifatida ko'rsatilgan [xmin, xmax, ymin,

ymax, zmin, zmax]. Odatiy o'lcham: dan

x, y, z massivlari.

• shakl

◦ Model joylashtiriladigan bosqich.

• chiziq\_ kengligi

◦ Chiziq kengligi. Standart qiymat: 2.0.

• ism

◦ Model nomi.

• xiralik

◦ oshkoralik.

• reset\_zoom

Sc Masshtabni tiklash.

• vmax, vmin

◦ Rang shkalasi uchun maksimal va minimal qiymatlar.

**15.3.1 kontur3d () funktsiyasi**

Contour3d () funktsiyasi o'tgan *3D* to'plamdan izosurflarni hosil qiladi

ma'lumotlar. Kontseptual jihatdan u tuzadigan contour\_surf () ga o'xshaydi

371

|  |
| --- |
| **Sahifa 373** |

kontur chiziqlari (" *15.2.3* contour\_surf () *funktsiyasi* " bo'limiga qarang ). Ikkita mumkin

asosiy qo'ng'iroq imkoniyatlari:

• koordinatalarini ko'rsatmasdan: contour3d (skalar, ...).

• koordinatalarni tayinlash bilan: kontur3d (x, y, z, skalar,…).

Scalars parametri - bu ma'lumotlarning *3D* massivi

izosurfalar qurish. Qolgan parametrlar butunlay bir xil.

contour\_surf () bilan, warp\_scale bundan mustasno, funktsiya unga ega

contour3d () yo'q.

Keling, kosmosdagi konus tenglamasi asosida ma'lumotlar qatorini yarataylik:

x, y, z = np.ogrid [- 7 : 7 : 0.1 , - 7 : 7 : 0.1 , - 1 : 7 : 0.1 ]

skalar = x \* x + y \* y - z \* z

Standart parametr qiymatlari bo'lgan model:

mlab.contour3d (skalar)

**15.22-rasm - contour3d () funktsiyasini namoyish etish**

372

|  |
| --- |
| **Sahifa 374** |

Keling, izosurfalar sonini ko'paytiraylik:

mlab.contour3d (skalar, konturlar = 10 )

**15.23-rasm - "Contrours" parametri bilan ishlashni namoyish etish**

**contour3d () funktsiyalari**

Keling, shaffoflik bilan qanday ishlashni namoyish qilaylik, ikkita variant mavjud -

bu shaffoflikni modelning barcha elementlariga yoki shaffoflikka o'rnatish uchun

ma'lumotlar to'plamidagi skalar qiymati bilan aniqlanadi. Birinchisi uchun

opatsizlik parametriga javob beradi:

mlab.contour3d (skalar, konturlar = 10 , xiralik = 0,5 )

373

|  |
| --- |
| **Sahifa 375** |

**15.24-rasm -** Shaffoflik **parametri bilan ishlashni namoyish etish**

contour3d () **funktsiyalari**

Ikkinchi variant shaffof parametr orqali faollashtiriladi:

**15.25-rasm - shaffof parametr bilan ishlashni namoyish etish**

**contour3d () funktsiyalari**

374

|  |
| --- |
| **Sahifa 376** |

Dan past qiymatlarga ega bo'lgan maydonlarga e'tibor bering

uzatilgan to'plamning (ko'k) shaffofligi oshib ketdi

eng yuqori qiymatlarga ega bo'lgan joylar (qizil).

**15.3.2 Quiver3d () funktsiyasi**

Quiver3d () funktsiyasi o'tganlardan vektor maydonini yaratadi

vektorlarning koordinatalari. Kutubxonada uchta asosiy ma'lumotlar mavjud

funktsiya bilan ishlash variantlari:

quiver3d (u, v, w, ...)

quiver3d (x, y, z, u, v, w, ...)

quiver3d (x, y, z, f, ...)

Birinchisida faqat vektorning tarkibiy qismlari quiver3d (), in-ga uzatiladi

vektorning ikkinchi koordinatalari va tarkibiy qismlari, uchinchi koordinatalarida va

ularni qabul qiladigan va vektor tarkibiy qismlarini qaytaradigan funktsiya.

Komponentlar va koordinatalar - bu *3D* ma'lumotlar to'plami.

Quiver3d () funktsiyasining parametrlari parametrlari bilan bir xil

contour3d () ixtiyoriy skalarar argumenti bilan

bir qator skaler qiymatlarni topshirishingiz mumkin.

Ma'lumotlar to'plami va koordinatalari bo'yicha funktsiyani tayyorlaylik.

vektorning tarkibiy qismlarini qaytaradi:

x, y, z = np.mgrid [ 0 : 3 , 0 : 3 , 0 : 3 ]

u = x \* x

v = y

w = z

def f (x, y, z):

x \* x, y, z ni qaytaring

375

|  |
| --- |
| **Sahifa 377** |

Vektorli maydonni quiver3d () funktsiyasini chaqirish orqali qurish mumkin

o'tgan komponentlar va koordinatalar:

mlab.quiver3d (x, y, z, u, v, w, scale\_factor = 0,2 )

mlab.show ()

yoki koordinatalari va funktsiyasi bilan f:

mlab.quiver3d (x, y, z, f, scale\_factor = 0,2 )

mlab.show ()

Ikkala holatda ham biz quyidagi shaklning modelini olamiz:

**15.26-rasm - quiver3d () funktsiyasini namoyish etish**

**15.3.3 funktsiyasi hajmi\_slice ()**

Volume\_slice () funktsiyasi o'tgan *3D* to'plamlar uchun tilimlarni yaratadi

ma'lumotlar. Ikkita asosiy qo'ng'iroq opsiyalari mavjud:

• skalar to'plamini o'tkazish bilan:

hajm\_slice (skalar, ...)

• koordinatalar va skalerlarni ko'rsatish:

hajm\_slice (x, y, z, skalar, ...)

376

|  |
| --- |
| **Sahifa 378** |

Bobning boshida keltirilgan parametrlardan tashqari, uchun

hajmi\_slice () mavjud:

• samolyot\_opacity

◦ Kesuvchi tekislikning shaffofligi.

• tekislik yo'nalishi

◦ Dilim tekisligining joylashuvi, sukut bo'yicha "x\_axes".

• slice\_index

◦ Dilim tekisligi joylashgan indeks.

Demo ma'lumotlar to'plami:

x, y, z = ogrid [- 7 : 7 : 0.1 , - 7 : 7 : 0.1 , - 1 : 7 : 0.1 ]

skalar = x \* x + y \* y - z \* z

Keling, *x* o'qi bo'ylab tilim *quramiz* :

mlab.volume\_slice (skalar, tekislik\_orientation = 'x\_axes')

mlab.show ()

**15.27-rasm - hajmi\_slice () funktsiyasini namoyish etish**

377

|  |
| --- |
| **Sahifa 379** |

Yana bitta o'qga kesilgan tekislikni qo'shing va uni o'rtasiga o'tkazing:

mlab.volume\_slice (skalar, slice\_index = round (len (x) / 2 )),

tekislik\_orientation = **'x\_axes'** )

mlab.volume\_slice (skalar, tekislik\_orientation = **'y\_axes'** )

mlab.outline ()

mlab.show ()

**15.28-rasm - ikkita kesilgan tekislik bilan ishlashni namoyish etish**

378

|  |
| --- |
| **380-bet** |

**16-bob. *Quvur liniyasi* bilan ishlash**

Biz kuzatgan *Mayavi* kutubxonasi bilan ishlashning varianti

15-bob " *Ma'lumotlarni vizuallashtirish* " aslida

juda cheklangan va barcha imkoniyatlardan foydalanishga imkon bermaydi,

ushbu vosita tomonidan taqdim etilgan. Aslida, ular bilan funktsiyalar

biz tanishgan stsenariylar

(shablonlar) ma'lum bir natija beradigan ish. Misol uchun,

agar biz surf () funktsiyasi bilan ishlasak, biz uni quyidagicha o'tkazamiz

kirish ma'lumotlari massiv massivlardir va biz shunday sirtni olamiz

uzatilgan ma'lumotlarga asoslanib ma'lum bir tarzda quriladi va hokazo. Ko'proq ma'lumot olish uchun

chuqur darajadagi *Mayavining* ishi bir *qatorni* bajarishdir

operatsiyalari *quvuri* rejimida. *Quvur liniyasi* taxmin qiladi

uch bosqich ketma-ketligi:

• ma'lum bir manbadan *ma'lumotlarni manba* ob'ektiga yuklash ;

• to'plamdagi vositalar yordamida ma'lumotlarni o'zgartirish

*Filtrlar*[*22*](https://translate.googleusercontent.com/translate_f#380)*-*bu ixtiyoriy qadam;

• *Modullar* to'plamidan modul yordamida model yaratish[*23*](https://translate.googleusercontent.com/translate_f#380)...

*Quvur liniyasini* qurish uchun *Mayavi* kutubxonasi vosita *beradi*

mlab paketidagi quvur liniyasi.

15-bobda muhokama qilingan funktsiyalardan biriga qo'ng'iroq qilganingizda,

*Mayavi* o'z ichiga tegishli *quvur liniyasini* quradi

kerakli ma'lumotlar manbai bilan ishlashni ta'minlaydigan ob'ektlar va

tegishli turdagi modelni yaratadigan modul.

22 [https://docs.enthought.com/mayavi/mayavi/filters.html#filters](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://docs.enthought.com/mayavi/mayavi/filters.html%23filters#filters)

23 [https://docs.enthought.com/mayavi/mayavi/modules.html#modules](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://docs.enthought.com/mayavi/mayavi/modules.html%23modules#modules)

379

|  |
| --- |
| **Sahifa 381** |

Konveyer konstruktsiyasini ko'rish uchun tugmani bosing

Voqea joyidagi asboblar panelidagi *"Mayavi quvuri"* (16.1-rasmga qarang). Uchun

Masalan, funktsiya uchun hosil bo'lgan *quvur liniyasini* ko'rib chiqamiz

imshow (). Quyidagi dastur kodini ishga tushiring:

np sifatida numpy- ni import qilish

dan mayavi import mlab

x, y = np.mgrid [- 2 : 2 : 0,1 , - 2 : 2 : 0,1 ]

z = np.cos (x \* y) \* np.sin (x \* y)

mlab.imshow (z)

Bu quyidagi rasmda ko'rsatilgan sahnani ochadi.

**16.1-rasm - imshow () funktsiyasi yordamida qurilgan model**

Asboblar panelidagi ko'rsatilgan tugmani bosing. Derazada " *Mayavi*

*quvuri* "Quvur liniyasi oynaning chap tomonida daraxt sifatida ko'rsatiladi.

380

|  |
| --- |
| **Sahifa 382** |

**Shakl 16.2 - *Mayavi quvur liniyasi* oynasi**

Bizning holatimizda konveyer ikkita elementdan iborat:

• *Array2DSource - ma'lumotlar manbai* ob'ektini yaratadi ;

• *ImageActor -* sahnada namoyish etish uchun model yaratadi.

Tegishli *quvurlarni* qo'lda quramiz :

src = mlab.pipeline.array2d\_source (z)

img = mlab.pipeline.image\_actor (src)

mlab.show ()

Ushbu kodni bajarish natijasi bilan variantga o'xshash bo'ladi

mlab.imshow (). Aslida satrlar:

mlab.imshow (z)

va

src = mlab.pipeline.array2d\_source (z)

img = mlab.pipeline.image\_actor (src)

ekvivalentlar.

381

|  |
| --- |
| **Sahifa 383** |

Keling, yanada murakkab *quvur liniyasini* ko'rib chiqaylik , buning uchun biz uni funktsiyaga o'tkazamiz

surf () ma'lumotlar *z* :

mlab.surf (z)

mlab.show ()

Quvur liniyasi quyidagicha bo'ladi:

Keling, uni kodda qo'llaymiz:

src = mlab.pipeline.array2d\_source (z)

çözgü = mlab.pipeline.warp\_scalar (src)

norm = mlab.pipeline.poly\_data\_normals (çözgü)

surf = mlab.pipeline.sface (norma)

mlab.show ()

**16.1 *quvur liniyasi* tuzilishi**

*Mayavi quvuri* necha qatlamdan iborat. Eng yuqori qismida

Bu dvigatel ( *Dvigatel* ). U element daraxtida ko'rinmaydi

*quvuri* , vosita sahnani yaratish va yo'q qilish uchun javobgardir

modeli ko'rsatiladi. Keyingi *sahna* , u o'z ichiga oladi

ma'lumotlar manbalari ( *Ma'lumotlar manbai* ), filtrlar ( *Filtrlar* ), modul menejeri

( *Modullar menejeri* ) va vizualizatsiya modullari ( *Modullar* ). *Dvigatel* va *sahna*

bitta nusxada, zanjirlarda quyidagilar mavjud

tuzilishi: ***ma'lumotlar manbai -> filtrlar (noldan n donagacha) ->***

***Modullar menejeri -> vizualizatsiya moduli*** bir nechta bo'lishi mumkin

bitta sahna ichida.

382

|  |
| --- |
| **Sahifa 384** |

Ma'lumot manbalari *numpy-* dan ma'lumotlarni yuklash uchun -

massivlar, ro'yxatlar va boshqalar. maxsus narsalarga. Filtrlar ruxsat beradi

ma'lumotlar, modul menejeri bo'yicha turli xil o'zgarishlarni amalga oshirish

(sifatida ko'rsatiladi *rang va afsonalar* ham *Mayavi quvuri* ) rangini nazorat

va ma'lumotlar taqdimoti. Modullarni to'g'ridan-to'g'ri tasvirlash

chiziqlar, yuzalar, nuqtalar va shaklida sahnada vakolatxonani shakllantirish

va boshqalar.

*Quvurni* foydalanuvchi interfeysi orqali qo'lda yig'ish mumkin

yoki yaratish funktsiyalarini chaqiradigan dastur yozing

quvur liniyasining elementlari va ularga kerakli dalillarni uzatadi. Uchinchidan

variant - oldindan tayyorlangan funktsiyadan foydalanish

*Mayavi* kutubxonasi , bu aniq echim uchun *quvur liniyasini* quradi

sirtlarni, chiziqlarni va boshqalarni namoyish qilish kabi vazifalar. Har bir

*quvur liniyasiga* kiradigan element asosiy elementga havolaga ega,

u oladigan ma'lumotlar.

Funktsiyaning asosiy elementlarini ko'rsatish uchun misol yozamiz

bemaqsad ():

obj = mlab.surf (z)

while ( To'g'ri ):

chop etish ( turi (obj))

if hasattr (obj, **"ota-ona"** ):

obj = obj.parent

boshqa :

tanaffus

Keling, ushbu dasturni ishga tushiramiz:

<sinf 'mayavi.modules.surface.Surface'>

<sinf 'mayavi.core.module\_manager.ModuleManager'>

<sinf 'mayavi.filters.poly\_data\_normals.PolyDataNormals'>

<sinf 'mayavi.filters.warp\_scalar.WarpScalar'>

383

|  |
| --- |
| **Sahifa 385** |

<sinf 'mayavi.sources.array\_source.ArraySource'>

<sinf 'mayavi.core.scene.Scene'>

<sinf 'mayavi.core.engine.Engine'>

Iltimos, ilova qiling, agar siz dasturiy ravishda *quvur liniyasini* qurishda , keyin aniq

siz *ModuleManager-* ni, shuningdek *Scene* va *Engine-* ni ko'rsatishingiz *shart* emas .

**16.2 Ma'lumot manbalari bilan ishlash**

Vizualizatsiya muammosini hal qilish uchun *quvur liniyasini* yaratishda birinchi narsa

nima qilish kerak - bu ma'lumot manbai to'g'risida qaror qabul qilish

quvur liniyasida ishlatiladi. Bilan ko'rsatiladigan ma'lumotlar

*Mayavidan* foydalanish ikkita katta guruhga bo'linishi mumkin - bu

bog'liq va bog'liq bo'lmagan.

Aloqasi bo'lmagan bir qator *ochko* bo'ladi

koordinatalari tasodifiy taqsimlanishi mumkin bo'lgan nuqtalar to'plami

bo'shliq, ular orasidagi masofa va raqamli qiymatlar

nuqtalardagi parametrlar ham tasodifiy bo'lishi mumkin. Bunday

to'siq, ma'lum bir ma'lumot elementi haqida hech narsa deyish mumkin emas

qo'shni qiymatlar.

Majmui *ulangan ma'lumotlar nuqtalari* bo'ladi

ma'lumotlar tizimli, teng taqsimlangan blok.

Uning o'ziga xos xususiyati shundaki, qo'shni elementlar (nuqtalar)

funktsional jihatdan bir-biri bilan bog'liq (shuning uchun ism), siz qilishingiz mumkin

ular interpolatsiya natijasidir. Bunday to'plamlar hali ham mavjud

maydon deb nomlangan. Bog'langan ma'lumotlar *aniq* va ga bo'linadi

yopiq ( *yopiq* ) bog'liq. Birinchi holda, bu taxmin qilinadi

ma'lumotlar panjarali tuzilishda joylashtirilgan, shuning uchun

384

|  |
| --- |
| **Sahifa 386** |

masalan, *2D* tasvirlar tasvirlangan . Ikkinchisida, orasidagi bog'liqlik

nuqtalar aniq ko'rsatilgan, bu ma'lumotlarga chiziqlar chizish imkonini beradi,

raqamlar hajmini aniqlash va h.k.

Qurilish manbalari uchun eng ko'p ishlatiladigan funktsiyalar

bog'liq bo'lmagan ma'lumotlar quyida keltirilgan:

• scalar\_scatter ()

• vector\_scatter ()

Tegishli to'plamlarni yaratish uchun quyidagi guruhlardan foydalaniladi

funktsiyalari:

• bevosita bog'langan holda yaratish:

• scalar\_field ()

• vector\_field ()

• array2d\_source ()

• aniq bog'liqlik yaratish:

• line\_source ()

• triangular\_mesh\_source ()

Dastlabki ma'lumotlar to'plamlari sifatida yuqorida keltirilgan funktsiyalar quyidagilardir

*noaniq* qatorlarni qabul qilish .

Keling, ushbu funktsiyalarni batafsil ko'rib chiqamiz.

**Scalar\_scatter () funktsiyasi** siyrak bog'lanmaganlikni hosil qiladi

ma'lumotlar manbai quyidagi qo'ng'iroq imzolariga ega:

scalar\_scatter (lar, ...)

scalar\_scatter (x, y, z, s, ...)

scalar\_scatter (x, y, z, f, ...)

385

|  |
| --- |
| **Sahifa 387** |

Parametr s - ma'lumotlar bazasini belgilaydi: bo'ladigan qator

ingl. Agar x, y, z parametrlari ko'rsatilmagan bo'lsa, ularning o'rniga, in

koordinatalar sifatida s to'plamidan indekslardan foydalaniladi. Buning o'rniga

s, koordinatalarni va qabul qiladigan f funktsiyani belgilashingiz mumkin

tegishli qiymatni qaytaring.

Scalar\_scatter () bilan ishlashga misol:

s = [ 1 , 2 , 3 , 4 , 5 ]

src\_data = mlab.pipeline.scalar\_scatter (lar)

mlab.pipeline.glyph (src\_data)

mlab.show ()

**16.3-rasm - funktsiyani namoyish etish**

**mlab.pipeline.scalar\_scatter ()**

**Vector\_scatter () funktsiyasi** vektorning siyrak manbasini yaratadi

ma'lumotlar. Funktsiya imzolari:

vector\_scatter (u, v, w, ...)

vector\_scatter (x, y, z, u, v, w, ...)

vector\_scatter (x, y, z, f, ...)

U, v, w parametrlari xuddi funktsiyadagi kabi vektorlarning tarkibiy qismlari

scalar\_scatter (), agar x, y, z ko'rsatilmagan bo'lsa, koordinatalar bo'ladi

u, v, w to'plamlaridan indekslar qabul qilinadi. F funktsiyasi qaytishi kerak

berilgan koordinatalar to'plami uchun vektor komponentlari.

386

|  |
| --- |
| **Sahifa 388** |

**Scalar\_field () funktsiyasi** skalar maydonini yaratadi, funktsiya imzosi:

scalar\_field (s, ...)

scalar\_field (x, y, z, s, ...)

scalar\_field (x, y, z, f, ...)

S, x, y, z parametrlari *2D* , *3D* ma'lumotlar to'plamidir, masalan,

numpy.ogird () funktsiyalari yordamida tuzilishi mumkin,

numpy.mgird ().

Foydalanish misoli:

x, y, z = np.ogrid [ 0 : 5 , 0 : 5 , 0 : 5 ]

s = x \* y \* z

src = mlab.pipeline.scalar\_field (lar)

hajmi = mlab.pipeline.volume (src, vmin = 0 , vmax = 10 )

mlab.show ()

**16.4-rasm - scalar\_field () va namoyish**

**hajmi ()**

**Vector\_filed ()** funktsiyasi **vector\_scatter ()** funktsiyasiga o'xshaydi

faqat vektor ma'lumotlari uchun.

387

|  |
| --- |
| **Sahifa 389** |

**Array2d\_source () funktsiyasi***2 o'lchovli* ma'lumot manbasini yaratadi

*2D* manba qatori. Funktsiya imzolari:

array2d\_source (lar, ...)

array2d\_source (x, y, s, ...)

array2d\_source (x, y, f, ...)

S, x, y parametrlari - bu bo'lishi mumkin bo'lgan kabi *2 o'lchovli* massivlar

numpy.ogird (), numpy.mgird () bilan qurilgan.

Foydalanish misoli:

x, y = np.ogrid [- 2 : 2 : 0,1 , - 2 : 2 : 0,1 ]

z = np.cos (x \* y)

src = mlab.pipeline.array2d\_source (z)

img = mlab.pipeline.image\_actor (src)

mlab.show ()

**Shakl 16.5 - array2d\_source () va funktsiyalarining ishlashini namoyish etish**

**image\_actor ()**

**Line\_source () funktsiyasi** chiziqli ma'lumotlar to'plamini yaratadi.

Funktsiya imzolari:

line\_source (x, y, z, ...)

line\_source (x, y, z, s, ...)

388

|  |
| --- |
| **390-bet** |

line\_source (x, y, z, f, ...)

Parametrlar x, y, z - chiziq nuqtalarining koordinatalari, s - qator

chiziqning xususiyatlarini belgilaydigan qiymatlar.

t = np.linspace ( 0 , 5 \* np.pi, 100 )

x = t \* np.cos (t)

y = t \* np.sin (t)

z = t

line = mlab.pipeline.line\_source (x, y, z)

tube = mlab.pipeline.tube (chiziq, tube\_radius = 1 )

bemaqsad = mlab.pipeline.surface (tube)

mlab.show ()

**Shakl 16.6 - line\_source () va namoyish**

**sirt ()**

**16.3 Filtrlar bilan ishlash**

Yilda Filtrlar *quvuri* 'e *Mayavi*[*24*](https://translate.googleusercontent.com/translate_f#390) faqat transformatsiya uchun mo'ljallangan

ma'lumotlar, ular ma'lumot bermaydi. Ko'rib chiqing

sizning ishingizda foydali bo'lishi mumkin bo'lgan ba'zi filtrlar.

24[https://docs.enthought.com/mayavi/mayavi/filters.html#filters](https://translate.google.com/translate?hl=uz&prev=_t&sl=auto&tl=uz&u=https://docs.enthought.com/mayavi/mayavi/filters.html%23filters#filters)

389

|  |
| --- |
| **391-bet** |

***CellDerivatives***

Kirish ma'lumotlar to'plamining hosilalarini hisoblab chiqadi (skalar yoki

vektorli ma'lumotlar), qiymatlari bo'ladigan hujayralar to'plamini yaratadi

hosilalar.

Misol:

x, y = np.mgrid [- 2 : 2 : 0,1 , - 2 : 2 : 0,1 ]

z = np.cos (x \* y) \* np.sin (x \* y)

src = mlab.pipeline.array2d\_source (z)

cell\_drv = mlab.pipeline.cell\_derivatives (src)

mlab.pipeline.glyph (cell\_drv, mode = **'cube'** )

mlab.pipeline.image\_actor (src)

Shakl 16.7 da ko'rsatilgan rasm uchun,

16.8-rasmda keltirilgan lotin qiymatlari bilan maydon.

**Shakl 16.7 - Original rasm**

390

|  |
| --- |
| **392-bet** |

**16.8-rasm - *CellDerivatives* filtrining namoyishi**

***CellToPointData***

Ma'lumotlar bo'lgan *hujayra* atributini *nuqta* atributiga, *nuqta* qiymatiga o'zgartiradi

bu nuqta tegishli bo'lgan hujayralarning o'rtacha qiymati sifatida aniqlanadi.

*PointToCellData* teskari filtri *mavjud.*

***Kontur***

Izosurfalar yaratish uchun kontur filtri.

***CutPlane***

Imkoniyat beradigan sahnada *3D* vositasi bo'lgan interaktiv filtr

ma'lumotlar to'plamlari uchun kesuvchilarni yaratish.

Misol:

x, y, z = np.ogrid [- 7 : 7 : 0.5 , - 7 : 7 : 0.5 , - 1 : 7 : 0.5 ]

skalar = x \* x + y \* y - z \* z

src = mlab.pipeline.scalar\_field (skalar)

cut\_pl = mlab.pipeline.cut\_plane (src)

mlab.pipeline.surface (cut\_pl)

mlab.pipeline.iso\_surface (src, konturlar = 3 , shaffof = rost )

391

|  |
| --- |
| **393-bet** |

**Shakl 16.9 - *CutPlane* filtri *ishini* namoyish etish**

***DataSetClipper***

Filtr ma'lumotlar bazasini ma'lum bir sohaga o'rnatadi:

x, y, z = np.ogrid [- 7 : 7 : 0.5 , - 7 : 7 : 0.5 , - 1 : 7 : 0.5 ]

skalar = x \* x + y \* y - z \* z

src = mlab.pipeline.scalar\_field (skalar)

dsc = mlab.pipeline.data\_set\_clipper (src)

mlab.pipeline.glyph (dsc)

**Shakl 16.10 - *DataSetClipper* filtrini namoyish etish**

392

|  |
| --- |
| **394-bet** |

***ExtractEdges***

Ma'lumotlar to'plamidan hujayra chegaralarini oladi:

x, y, z = np.ogrid [- 7 : 7 : 0.5 , - 7 : 7 : 0.5 , - 1 : 7 : 0.5 ]

skalar = x \* x + y \* y - z \* z

src = mlab.pipeline.scalar\_field (skalar)

qirralar = mlab.pipeline.extract\_edges (src)

mlab.pipeline.sface (qirralari)

mlab.show ()

**16.11-rasm - *ExtractEdges* filtrining namoyishi**

***ExtractGrid***

Dastlabki to'plamdan ma'lumotlarning bir qismini ajratib olishga imkon beradi:

x, y, z = np.ogrid [- 7 : 7 : 0.5 , - 7 : 7 : 0.5 , - 1 : 7 : 0.5 ]

skalar = x \* x + y \* y - z \* z

src = mlab.pipeline.scalar\_field (skalar)

panjara = mlab.pipeline.extract\_grid (src)

grid.x\_min = 5

grid.x\_max = 20

grid.y\_min = 5

grid.y\_max = 20

mlab.pipeline.surface (grid)

mlab.show ()

393

|  |
| --- |
| **395-bet** |

**16.12-rasm - *ExtractGrid* filtrining namoyishi**

***ExtractVectorNorm***

Filtr vektorning normasini (uzunligini) hisoblab chiqadi. Bilan ishlashda ishlatiladi

vektorli ma'lumotlar. Masalan, agar siz vektorni ko'rsatadigan bo'lsangiz

uzunliklarni chizmasdan ma'lumotlar, natijada quyidagicha bo'ladi:

x, y, z = np.mgrid [ 0 : 3 , 0 : 3 , 0 : 3 ]

u = x \* x

v = y

w = z

src = mlab.pipeline.vector\_scatter (u, v, w)

mlab.pipeline.glyph (src, mode = **"2darrow"** )

mlab.show ()

**Shakl 16.13 - Boshlang'ich model**

394

|  |
| --- |
| **396-bet** |

Bunday holda, biz faqat vektorlarning yo'nalishlarini ko'ramiz. Qo'shaylik

vektor uzunligini hisoblash:

src = mlab.pipeline.vector\_scatter (u, v, w)

normalar = mlab.pipeline.extract\_vector\_norm (src)

mlab.pipeline.glyph (normalar, rejim = **"2darrow"** )

mlab.show ()

**16.14-rasm - *ExtractVectorNorm* filtrini namoyish etish**

***ExtractVectorComponentlari***

Vektorlarning tarkibiy qismlarini oladi:

*Y-* komponentlarini uzatilgan vektorlardan ajratib oling :

x, y, z = np.mgrid [ 0 : 3 , 0 : 3 , 0 : 3 ]

u = x \* x

v = y

w = z

src = mlab.pipeline.vector\_scatter (u, v, w)

comps = mlab.pipeline.extract\_vector\_components (src)

comps.component = **'y-komponent'**

mlab.pipeline.glyph (komps, mode = **"2darrow"** )

mlab.show ()

395

|  |
| --- |
| **397-bet** |

**Shakl 16.15 - Filtrni ishlashini namoyish etish**

***ExtractVectorComponentlari***

***Gaussian Splatter***

Filtr nuqtalarni elliptik guss hajmiga joylashtiradi

tarqatish. Bu zichlik maydonini to'plam bo'yicha taxmin qilish uchun foydalidir

tarqoq nuqtalar.

16.16-rasmga o'xshash to'plam uchun.

**Shakl 16.16 - boshlang'ich model**

*GaussianSplatter* filtri bilan oldindan ishlov *beramiz* :

src = mlab.pipeline.array2d\_source (z)

gsp = mlab.pipeline.gaussian\_splatter (src)

mlab.pipeline.glyph (gsp, mode = **'konus'** )

mlab.show ()

Biz quyidagi modelni olamiz:

396

|  |
| --- |
| **398-bet** |

**Shakl 16.17 - *GaussianSplatter* filtrining namoyishi**

Vizualizatsiya uchun boshqa modullardan foydalanib, siz juda ko'p narsalarni olishingiz mumkin

qiziqarli modellar:

**16.18-rasm - mode = 'Volume' parametri bilan ishlashni namoyish etish**

***GaussianSplatter* filtri**

397

|  |
| --- |
| **399-bet** |

**16.19-rasm - parametr bilan ishlashni namoyish etish**

**mode = 'isosurface' *GaussianSplatter* filtri**

***GreedyTerrainDecimation***

Agar yaqin *2D* ularni tutib, uchburchak, bir qator tasvirni

minimal miqdor:

x, y = np.mgrid [- 2 : 2 : 0,1 , - 2 : 2 : 0,1 ]

z = np.cos (x \* y) \* np.sin (x \* y)

src = mlab.pipeline.array2d\_source (z)

gtd = mlab.pipeline.greedy\_terrain\_decimation (src)

mlab.pipeline.surface (gtd)

mlab.show ()

**Shakl 16.20 - Filtrni ishlashini namoyish etish**

***GreedyTerrainDecimation***

398

|  |
| --- |
| **400-bet** |

***ImageChangeInformation***

Filtr koordinata o'qlarining joylashishini, cho'zilishini belgilashga imkon beradi

eksa bo'ylab va boshqalar. ma'lumotlarning o'zi o'zgartirmasdan.

***Mask nuqtalari***

Ma'lumotlarning pastki namunalarini shakllantirish uchun filtr.

***PolyDataNormals***

Tuzatish uchun ishlatiladigan kirish ma'lumotlari uchun normalarni hisoblab chiqadi

tasvirlar va boshqalar.

Masalan, ko'plab qoidabuzarliklarga ega modelni yarataylik,

buning uchun biz *WarpScalar* filtridan foydalanamiz :

z = np.random.rand ( 15 , 15 )

src = mlab.pipeline.array2d\_source (z)

çözgü = mlab.pipeline.warp\_scalar (src)

mlab.pipeline.surface (çözgü)

mlag.show ()

**16.21-rasm - *WarpScalar* filtrining namoyishi**

Endi filtrlashning yana bir darajasini qo'shib, usulsizliklarni yaxshilaymiz

*PolyDataNormals* yordamida :

399

|  |
| --- |
| **Sahifa 401** |

z = np.random.rand ( 15 , 15 )

src = mlab.pipeline.array2d\_source (z)

çözgü = mlab.pipeline.warp\_scalar (src)

poly = mlab.pipeline.poly\_data\_normals (çözgü)

mlab.pipeline.surface (poly)

mlab.show ()

**16.22-rasm - *PolyDataNormals* filtrining namoyishi**

***QuadricDecimation***

Asl meshdagi uchburchaklar sonini kamaytiradi, hosil qiladi

modelni yaxshiroq namoyish etish.

***Chiqish-ni tanlang***

Uchun ko'p blokli ma'lumotlar manbalari bilan ishlashda foydalaniladi

chiqish tanlovi.

***Stripper***

Filtr yuzalardagi bo'shliqlarni to'ldirish uchun ishlatiladi. Yarataylik

trubka *striptiz filtrisiz* :

t = np.linspace ( 0 , 5 \* np.pi, 100 )

x = t \* np.cos (t)

y = t \* np.sin (t)

z = t

400

|  |
| --- |
| **Sahifa 402** |

src = mlab.pipeline.line\_source (x, y, z, z)

tb = mlab.pipeline.tube (src, tube\_radius = 1,5 , tube\_sides = 10 )

mlab.pipeline.surface (tb)

mlab.show ()

**Shakl 16.23 - *Stripper* filtri bo'lmagan model**

src = mlab.pipeline.line\_source (x, y, z, z)

stp = mlab.pipeline.stripper (src)

tb = mlab.pipeline.tube (stp, tube\_radius = 1,5 , tube\_sides = 10 )

mlab.pipeline.surface (tb)

mlab.show ()

**16.24-rasm - *Stripper* filtrining namoyishi**

***Eshik***

Ma'lumotlarni kiritish uchun chegara o'rnatish uchun filtr.

401

|  |
| --- |
| **Sahifa 403** |

***TransformData***

Kirish ma'lumotlari bo'yicha chiziqli o'zgarishlarni amalga oshiradi.

***Naycha***

Chiziqlarni naychalarga o'zgartiradi.

***UserDefined***

Maxsus filtrlardan foydalanishga imkon beradi.

***Vortisit***

Kirish vektor maydonining girdobini hisoblab chiqadi.

***WarpScalar***

Belgilangan yo'nalish bo'yicha kiritishni o'zgartiradi (buradi)

berilgan o'lchov bilan yo'nalishlar.

***WarpVector***

Vektor bo'ylab kiritishni o'zgartiradi (çözgü).

**16.4 Modullar bilan ishlash**

*Mayavi quvur liniyasidagi* modullar ma'lumotlarni taqdim etish *uchun* javobgardir va

voqea joyida qo'shimcha ma'lumotlarni namoyish qilish. Ma'lumotlar bo'lishi mumkin

to'g'ridan-to'g'ri va bir qator filtrlar orqali modulga o'tdi, agar

qo'shimcha konversiyalar talab qilinadi. Guruhlashaylik

modullarni funktsional asosda, shuning uchun ularni o'rganish osonroq bo'ladi.

**Izohlarni sahnada joylashtirish uchun modullar.**

Ushbu modullarning o'ziga xos xususiyati shundaki, ular joylashgan

mlab.pipeline-da emas, paket mlab. Ushbu modullarga quyidagilar kiradi:

• *O'qlar*

• Modelga koordinata o'qlarini qo'shadi.

402

|  |
| --- |
| **Sahifa 404** |

• *Xulosa*

• Model atrofida qutichani chizadi.

• *Orientatsiya yo'nalishlari*

• Sahnaga eksa yo'nalishlari bilan element qo'shadi.

• *Matn* , *Text3D*

• Sahnaga matn qo'shadi.

x, y, z = np.ogrid [- 7 : 7 : 0.1 , - 7 : 7 : 0.1 , - 1 : 7 : 0.1 ]

skalar = x \* x + y \* y - z \* z

src = mlab.pipeline.scalar\_field (skalar)

mlab.pipeline.iso\_surface (src)

mlab.axes ()

mlab.outline ()

mlab.orientation\_axes ()

mlab.text ( 0 , 0 , **"Past daraja"** )

**Shakl 16.25 - *Mayavi* modullari**

403

|  |
| --- |
| **Sahifa 405** |

**Skalyar ma'lumotlar bilan ishlash modullari.**

***Glif***

Ma'lumotlarni glif sifatida taqdim etadi:

x, y, z = np.ogrid [- 7 : 7 : 0.5 , - 7 : 7 : 0.5 , - 1 : 7 : 0.5 ]

skalar = x \* x + y \* y - z \* z

src = mlab.pipeline.scalar\_field (skalar)

mlab.pipeline.glyph (src)

mlab.show ()

**16.26-rasm - *Glif* filtrining namoyishi**

***IsoSurface***

Ma'lumotlarni izosurfalar sifatida ko'rsatish:

src = mlab.pipeline.scalar\_field (skalar)

mlab.pipeline.iso\_surface (src)

mlab.show ()

404

|  |
| --- |
| **Sahifa 406** |

**16.27-rasm - *IsoSurface* filtrining namoyishi**

***ScalarCutPlane***

Ma'lumotlar to'plamining faqat sekant tomonidan ajratilgan qismini aks ettiradi

samolyot:

src = mlab.pipeline.scalar\_field (skalar)

mlab.pipeline.scalar\_cut\_plane (src)

mlab.show ()

**16.28-rasm - *ScalarCutPlane* modulining namoyishi**

405

|  |
| --- |
| **Sahifa 407** |

***Yuzaki***

Berilgan ma'lumotlar to'plamidan sirt hosil qiladi:

src = mlab.pipeline.scalar\_field (skalar)

mlab.pipeline.surface (src)

mlab.show ()

**Shakl 16.29 - *Surface* modulining namoyishi**

**Vektorli ma'lumotlarni vizualizatsiya qilish uchun modullar.**

***VectorCutplane***

Vektorli ma'lumotlar uchun bo'laklarni yaratadi:

x, y, z = np.mgrid [ 0 : 3 : 0.5 , 0 : 3 : 0.5 , 0 : 3 : 0.5 ]

u = x \* x

v = y

w = z

src = mlab.pipeline.vector\_field (u, v, w)

mlab.pipeline.vector\_cut\_plane (src)

mlab.show ()

406

|  |
| --- |
| **Sahifa 408** |

**Shakl 16.30 - *VectorCutplane* modulining namoyishi**

***Vektorlar***

O'tkazilgan vektorlar to'plamini beradi:

x, y, z = np.mgrid [ 0 : 3 : 0.5 , 0 : 3 : 0.5 , 0 : 3 : 0.5 ]

u = x \* x

v = y

w = z

src = mlab.pipeline.vector\_field (u, v, w)

mlab.pipeline.vectors (src)

mlab.show ()

**Shakl 16.31 - *Vektorlar* modulining namoyishi**

407

|  |
| --- |
| **Sahifa 409** |

***Tartibga solish***

Oqim chiziqlarini o'rganishga imkon beruvchi interaktiv vosita,

o'tkazilgan vektorlar to'plamidan tuzilgan:

x, y, z = np.mgrid [ 0 : 3 : 0.5 , 0 : 3 : 0.5 , 0 : 3 : 0.5 ]

u = x \* x

v = y

w = z

src = mlab.pipeline.vector\_field (u, v, w)

mlab.pipeline.streamline (src, seedtype = **'tekislik'** )

mlab.show ()

**Shakl 16.32 - *Streamline* modulining namoyishi**

**Ma'lumotlar bilan ishlash modullari.**

***GridPlane***

Modul modelga panjara bilan tekislikni joylashtiradi:

x, y, z = np.ogrid [- 7 : 7 : 0.5 , - 7 : 7 : 0.5 , - 1 : 7 : 0.5 ]

skalar = x \* x + y \* y - z \* z

src = mlab.pipeline.scalar\_field (skalar)

mlab.pipeline.grid\_plane (src)

mlab.show ()

408

|  |
| --- |
| **410-bet** |

**Shakl 16.33 - *GridPlane* modulining namoyishi**

***ContourGridPlane***

Ma'lumotlar to'plami uchun konturlar to'plami bo'lgan tekislik.

***CustomGridPlane***

Tarmoqli foydalanuvchi tomonidan sozlanishi tekislikni yaratishga imkon beradi.

***ImagePlaneWidget***

Berilgan to'plam uchun interaktiv chiqib ketish tekisligini quradi

ma'lumotlar:

x, y, z = np.ogrid [- 7 : 7 : 0.5 , - 7 : 7 : 0.5 , - 1 : 7 : 0.5 ]

skalar = x \* x + y \* y - z \* z

src = mlab.pipeline.scalar\_field (skalar)

mlab.pipeline.image\_plane\_widget (src)

mlab.pipeline.volume (src)

mlab.show ()

409

|  |
| --- |
| **411-bet** |

**16.34-rasm - *ImagePlaneWidget* modulining namoyishi**

***ImageActor***

Ma'lumotlar to'plamini 2 o'lchovli tasvir sifatida ifodalaydi:

x, y = np.mgrid [- 2 : 2 : 0,1 , - 2 : 2 : 0,1 ]

z = np.cos (x \* y) \* np.sin (x \* y)

src = mlab.pipeline.scalar\_field (z)

mlab.pipeline.image\_actor (src)

mlab.show ()

**16.35-rasm - *ImageActor* modulining namoyishi**

***Tovush***

3D modelini yaratadi:

410

|  |
| --- |
| **412-bet** |

x, y, z = np.ogrid [- 7 : 7 : 0.5 , - 7 : 7 : 0.5 , - 1 : 7 : 0.5 ]

skalar = x \* x + y \* y - z \* z

src = mlab.pipeline.scalar\_field (skalar)

mlab.pipeline.volume (src)

mlab.show ()

**Shakl 16.36 - *Volume* modulining namoyishi**

411

|  |
| --- |
| **413-bet** |

**Xulosa**

Kutubxonalar haqida iloji boricha ko'proq ma'lumot berishga harakat qildik

*Matplotlib* , *Seaborn* va *Mayavi* , umid qilamanki sizga ma'lumot

ushbu kitob sahifalarida uchrashgan, foydali ekanligi isbotlangan. agar sizda bo'lsa

sharhlar yoki tarkib bo'yicha takliflar, keyin bizga yozing

devpractice.mail@gmail.com, biz siz bilan gaplashishni istaymiz.

412