

1.

podelila sam mapu na cetiri kvadratna podregiona kao i u proslom domacem, i onda sam ih plotovala
zatim sam dodelila podatke svakoj promenljivoj

```
treci_kvadrant = podaci[:,centar_y, :centar_x]  
cetvrti_kvadrant = podaci[:,centar_y, centar_x:]  
drugi_kvadrant = podaci[centar_y:, :centar_x]  
prvi_kvadrant = podaci[centar_y:, centar_x:]
```

Koristeci numpy metode izracunala sam medijanu, standardnu devijaciju i srednju vrednost, i to prikazala u tabeli

Izbran gornji levi i donji desni kvadrant - q2 i q4.

uradila primer za racunanje pirsonovog koeficijenta korelacije i p-vrednosti i zatim primenila na moje podatke. podaci su manipulisani metodom .flatten() jer su onda svi u istoj dimenziji pa na njima mogu da se primenjuju metode koje nam trebaju.

p-vrednost predstavlja verovatnocu da ce statisticki paramater biti veci ili jednak stvarno dobijenom rezultatu. Odnosno p-vrednost nam pokazuje da li je simulacija dovoljno verodostojna realnosti.

Pirsonov koeficijent nam pokazuje zavisnost dve velicine. U nasem slucaju dobili smo 0.1150 sto je vrlo mala pozitivna korelacija

izmedju dve promenljive.

2.

Definisali smo funkciju $f(x) = x + \sin(5\pi x) + 1$, konstantu i normalizovanu gustinu. Zatim racunamo max vrednost za $M > p(x)/g(x)$,

jer ce za takvo M sigurno vaziti nejednakost.

Uzorci x se prihvataju ako su generisani y uzorci manji od vrednosti ciljane gustine. Zatim se acceptance_ratio racuna kao broj povoljnih(len(y manji od vrednosti ciljane gustine))/broj mogucih ishoda(N).

za inverznu transformaciju koristimo metodu interp1d. Na kraju plotujemo dobijene vrednosti.

C= 1.6273239544735163

M= 1.7833107341438565

acceptance ratio= 0.56015