APRIL

# Teden 1. 4. - 5. 4. 2020

**DATUM:** 1. 4., 2. 4., 3. 4., 4. 4.

**ŠT. UR:** 4 x 8 = 32

**DELO:** Nadaljevanje pisanja podpoglavja »Optimalno število stolpcev«. Poiskal sem formule za tri izbire: Rice, Square-root in Sturges. Malo sem razmislil, kaj je smisel teh metod in kako se razlikujejo med sabo. Te tri metode sem implementiral in testiral na normalni in Pareto porazdelitvi. Po tem sem formule in opise zapisal v omenjeno podpoglavje.

# Teden 6. 4. - 12. 4. 2020

**DATUM:** 6. 4., 7. 4., 8. 4., 9. 4., 10. 4.

**ŠT. UR:** 5 x 8 = 40

**DELO:** Nadaljevanje pisanja podpoglavja od zadnjič. Tokrat sem iskal in analiziral naslednje metode: Doane, Scoot, Freedman-Diaconis. Pri raziskovanju izbire Doane sem zasledil, da se tej izbiri izogibajo zaradi nejasnosti in zahtevnosti izračuna, torej sem jo črtal iz svojega seznama izbir. Poleg tega daje tudi zelo slabe rezultate. Vseeno pa sem implementiral vse tri metode, testiral pa sem le Scoot in Freedman-Diaconis izbiri na normalni in Pareto porazdelitvi. Te dve metodi sem navedel in opisal tudi v podpoglavju »Optimalno število stolpcev«.

# Teden 13. 4. - 19. 4. 2020

**DATUM:** 14. 4., 15. 4., 16. 4., 17. 4., 18. 4.

**ŠT. UR:** 5 x 8 = 40

**DELO:** Naslednje podpoglavje je o »kernel density estimation«. Predstavim idejo in implementiram vsa jedra. Po nekaj testih se izkaže, da je Gaussovo jedro najboljše za vse porazdelitve s »trebuhi« (npr. normalna porazdelitev, Rayleigh porazdelitev ...), tako da sem se odločil, da bom uporabljal kar Gaussian kernel density estimation (tudi optimizirano v Pythonu). V upanju, da bi še pohitril to metodo, sem napisal svojo implementacijo kernel estimacije, izkazalo pa se je, da je v splošnem tista vgrajena v knjižnjico »scipy« hitrejša.

# Teden 20. 4. - 26. 4. 2020

**DATUM:** 20. 4., 21. 4., 22. 4., 23. 4., 24. 4.

**ŠT. UR:** 5 x 8 = 40

**DELO:** Opis KL metode za optimalno število stolpcev v podpoglavju »KL metoda«. To smo že naredili, a potrebna je bila natančna formulacija in definicija ter ponovni razmisleki. Znova je bilo potrebno zagnati nekaj testov, ki so trajali kar precej časa. Veseli me, da metoda konkurira ostalim metodam, je pa zelo počasna v primerjavi z ostalimi, zato sem jo poskušal pohitriti – zaenkrat mi je uspelo le to, da sem integral zapisal kot vsoto dveh integralov, kar je naredilo algoritem 2x hitrejši (en člen se izračuna izven zanke).

# Teden 27. 4. – 30. 4. 2020

**DATUM:** 28. 4., 29. 4., 30. 4.

**ŠT. UR:** 3 x 8 = 24

**DELO:** Vse do zdaj omenjene metode so bile uporabne le za ekvidistantne stolpce. Mogoče pa sem pozabil na možnost, da bi lahko bili ekvikvantni stolpci boljši za predstavitev histograma. Implementiral sem algoritem, ki za vhod sprejme set podatkov in približno željeno število stolpcev, vrne pa vektor mej stolpcev za histogram. V maju bom testiral vse zgoraj omenjene metode, kjer bom za kriterij vzel neko divergenco med točno porazdelitvijo in nastalim histogramom (kjer bom vzel zelo veliko podatkov), in poizkusil to vrednost minimizirati.

**SKUPAJ ŠTEVILO UR: 176**

Priponke: test za različne izbire optimalnega števila stolpcev, kernel estimator lastna skripta, implementacija za ekvikvantalne stolpce