

MC meetodid, praktikum 3

Teema 1: Hii-ruut testiga jaotusele vastavuse kontrollimine

Hii-ruut test on oma olemuselt vahend selleks, et kontrollida lõpliku arvu erinevate väärtustega juhusliku suuruse korral katsetulemuste vastamist mingile fikseeritud jaotusele, mis diskreetsel juhul on kirjeldatud paaridena (v_i, p_i) , $i = 1, \dots, m$, kus v_i , $i = 1, \dots, m$ on võimalikud väärtused ja p_i , $i = 1, \dots, m$ on nende saamise tõenäosused. Näiteks juhul, kui katseks on täringuvise ja tulemuseks märgime silmade arvu, siis täringu aususe kontrollimiseks võime teha n katset ning seejärel kontrollida, kas kõikide tulemuste tõenäosus võiks olla $\frac{1}{6}$. R-is tuleb χ^2 testi rakendamiseks lugeda kokku erinevate väärtuste esinemiste arv (käsk `table`) ja siis rakendada testi käsuga `chisq.test(tabel, p=c(p1,...,pm))`, kus `tabel` on käsu `table` rakendamisel saadud tulemus. Tõenäosused võib ära jätta, kui need on kõik võrdsed.

Harjutus 1

Testime lineaarset kongruentset generaatorit parameetritega $a = 697$, $b = 67$, $m = 16777216$. Tekita seemnest 11 lähtuvalt 400 pseudojuhuslikku arvu. Korruta need kuuega, liida 1 ja võta täisosa (kasuks tuleb funktsioon `floor`). Esimesed 6 arvu peaks tulema

```
## [1] 1 2 6 4 3 1
```

Kontrolli χ^2 -testi abil, kas need võiksid olla ausa täringu abil saadud visete tulemuseks. (kontrolliks: testi p väärtuseks peaks tulema 0.4812). Mis oleks otsuseks juhul $\alpha = 0.5$? Kui sageli me teeksime sellise α korral ausa täringu visete põhjal vale otsuse? Mis on testi tulemuse põhjal otsuseks, kui nõuda, et ausa täringu puhul me teeks õige otsuse üheksal juhul kümnest (sama pika katseseeria korral)?

Kui me tahame χ^2 -testi kasutada selleks, et kontrollida katsetulemuste vastavust mingi konkreetse jaotusega juhusliku suuruse X sõltumatute katsete korral saadud väärtustele, siis valime mingid m vastastikku välistavat X -ga seotud sündmust, mis katavad kõik võimalused, ning võtame iga katse tulemuseks selle sündmuse järjekorranumbri, mis sellel katsel toimus. Näiteks võime valida sündmusteks $A_i = \{c_{i-1} < X \leq c_i\}$, kus

$$-\infty = c_0 < c_1 < \dots < c_{m-1} < c_m = \infty.$$

Sel juhul on sündmuse A_i tõenäosus arvutatav juhusliku suuruse X jaotusfunktsiooni abil kujul $p_i = P(A_i) = F(c_i) - F(c_{i-1})$. Nüüd võime χ^2 -testi abil kontrollida, kas vaadeldavate sündmuste toimumise sagedus on kooskõlas nende toimumise tõenäosustega.

Harjutus 2

Tekita käskudega

```
set.seed(20200918)
x <- rexp(500,rate=0.2)
```

pseudojuhuslikud arvud, mis peaks vastama eksponentjaotusele $Exp(\frac{1}{5})$. Kontrolli nende arvude vastavust sellele jaotusele, vaadeldes piirkondadesse $A_1 = (-\infty, 1]$, $A_2 = (1, 2]$, $A_3 = (2, 3]$, $A_4 = (3, 4]$, ..., $A_{10} = (9, 10]$, $A_{11} = (10, \infty)$ sattumise vastavust vaadeldava jaotuse poolt määratud tõenäosustele. Milline on otsus, kui olulisuse nivooks valida 0.05?

Lahenduse skeem:

- 1) kõigepealt tekita vektor jaotuspunktides, kuhu võiks lisada ka $-\infty$ ja ∞ , millele vastavad väärtused R-is on `-Inf` ja `Inf`. Arve ja vektoreid saab kokku ühendada käsuga `c()`
- 2) Leia järgmisena jaotuse `Exp(0.2)` jaotusfunktsiooni väärtused toodud punktides. Siin on kasuks teadmine, et jaotusfunktsiooni saamiseks paneme `p` tähe jaotuse nime ette, nt `punif` on ühtlase jaotuse jaotusfunktsioon, `pnorm` on normaaljaotuse jaotusfunktsioon, `pexp` on eksponentjaotuse jaotusfunktsioon jne. Infot käsu võimalike parameetrite kohta saab kujul `?kask`.
- 3) Piirkondadesse sattumise tõenäosused on järjestikuste väärtuste vahed. Jällegi on R-is mugav vahend vektori järjestikuste väärtuste vahede leidmiseks, selleks on käsk `diff`, mille rakendamisel saame sündmustele vastavate tõenäosuste vektori.
- 4) Siinkohal tuleks mõelda, kas vaadeldava vektori pikkus on piisavalt suur, et igasse piirkonda sattumiste oodatav arv oleks vähemalt 5. Seda tingimust saab kontrollida, kui leida tõenäosuste vektori minimaalne väärtus ning korrutada see vektori `x` pikkusega. Vektori pikkust saab leida `length()` käsuga ning miinimumi jaoks on `min()` funktsioon.
- 5) Nüüd tuleb leida igale vaatluse korral see sündmus, mis toimus. Vaadeldaval kujul olevate sündmuste korral on selliseks vaatluste "sildistamiseks" mugav käsk `cut`, millele tuleb anda ette sildistatav vektor ja piirkondi defineerivate punktide vektor.
- 6) Rakenda `cut` käsu tulemusele käsku `table`, et lugeda kokku igasse osalõiku sattumise tõenäosused.

Otsus valitud olulisuse nivool on ...

Teema 2: empiirilise jaotusfunktsiooni ja vaadeldavale jaotusele vastava jaotusfunktsiooni erinevuste mõõtmisel põhinevad testid

Empiiriline jaotusfunktsioon annab iga piirkonna $A_x = (-\infty, x)$ korral sinna piirkonda sattunud vaatluste arvu jagatise kõikide vaatluste arvuga,

$$F_n(x) := \frac{\#\{i \mid x_i \leq x\}}{n}$$

Juhusliku suuruse X jaotusfunktsioon aga annab iga x korral sellisesse piirkonda sattumise tõenäosus:

$$F(x) = P(X \in A_x)$$

Kolmogorov-Smirnovi test mõõdab maksimaalset erinevust, st leiab valimi põhjal

$$K_n := \sup_x |F_n(x) - F(x)|$$

ning annab ka tõenäosuse, et valitud jaotusele vastava juhusliku suuruse puhul n katse tegemisel saame vähemalt sama suure väärtuse.

Cramer - von Mises test ehk ω^2 -test mõõdab empiirilise jaotusfunktsiooni ja jaotusele vastava jaotusfunktsiooni (mille tihedusfunktsiooniks olgu f) vahelist erinevust, arvutades allpool toodud integraali erinevuse ruudust:

$$\omega_n^2 := n \int_{-\infty}^{\infty} (F(x) - F_n(x))^2 f(x) dx$$

Jällegi on teada, kuidas see erinevus käitub juhul, kui vaadeldakse juhuslikule suurusele X vastavate sõltumatute katsete tulemusi.

Harjutus 3

Kontrolli Kolmogorov-Smirnovi testiga, kas eelmises harjutuses tekitatud pseudojuhuslikud arvud võivad vastata jaotusele `Exp(0.2)`. Kontrolli samuti, kas need võivad vastata ka jaotusele `Exp(0.19)`

Vihjed: Kolmogorov-Smirnovi testi rakendamiseks on käsk `ks.test`, millele tuleb anda valim, vastava jaotuse jaotusfunktsiooni nimi R-is ning lisaks jaotusfunktsioonile täiendavad parameetrid (näiteks normaaljaotuse korral `mean=1, sd=2`).

Harjutus 4

Korda eelmise harjutuse tulemusi Cramer-von Mises testiga.

Vihjed: selle testi kasutamiseks peab R-is olema installeeritud lisapakett `gofest`, mis tuleb pärast installeerimist laadida. Testi sooritamise käsuks on `cvm.test`, mille kasutamine on sarnane Kolmogorov-Smirnovi testiga.

Teema 3: Sõltumatuse testimine

Eelnevalt vaadeldud testid kontrollisid ainult seda, kas juhusliku suuruse väärtused paiknevad võimalike tulemuste hulgas nii, nagu teoreetilise jaotuse järgi oodata võiks, kuid ei arvestanud seda, kuidas need väärtuse katseseerias esinesid. Seega võivad katsetulemuste vahelised sõltuvused avastamata jääda.

Siit tuleb idee vaadelda mitte ainult katsetulemusi ükshaaval, vaid mingite komplektidena ja vaadelda komplektidega seotud sündmuste vastavust sõltumatute katsete korral saadavatele tulemustele.

Esimene idee - **sõltumatute paaride test**. Selleks

- Fikseerime m vastastikku välistavat sündmust $A_1, \dots, A_m$, mille tõenäosused vaadeldava jaotuse korral on $p_1, \dots, p_m$ ja mis katavad ära kõik võimalused (täissüsteem)
- Igale katsetulemusele seame vastavusse sündmuse numbri, mis sellel katsel toimus
- Saame numbritest $1, \dots, m$ koosneva jada $k_1, k_2, \dots, k_n$
- Vaatleme paare $(k_1, k_2), (k_3, k_4), \dots$
- Kui tegemist sõltumatute katsetega, siis paari (i, j) saamise tõenäosus on $p_i \cdot p_j$
- Kasutame χ^2 testi kontrollimiseks, kas esinemiste arvud $n_{i,j}$ on piisavalt lähedased oodatavale arvule $\frac{n}{2} p_i p_j$
- **NB!** testi rakendamiseks sagedustabeli koostamisel peame olema hoolikad, et kõik paarid tabelis esineksid (ka siis, kui katsetulemuste põhjal sellist paari ei tekkinud)! Samuti peaks veenduma, et iga sündmuse paari korral on toimumiste oodatavad arvud $\frac{n}{2} p_i p_j$ piisavalt suured (vähemalt 5), vastasel korral ei ole õigustatud χ^2 -testi rakendamisel p -väärtuseid leida χ^2 -jaotuse abil vaid need tuleks leida näiteks simulatsioonide teel (suvand `simulate.p.value=TRUE` käsus `chisq.test`)

Harjutus 5

Rakenda sõltumatute paaride testi harjutuses nr 2 kasutatud jada järjestikuste vaatluste sõltumatuse kontrolliks samade sündmuste korral, mida kasutati selles harjutuses jaotusele vastavuse kontrollimisel.

Lahenduse skeem: Lähtu varasemas ülesandes leitud faktorist (olgu selle nimeks `sildistatud_x`) ja sündmuste tõenäosuste vektorist (olgu selle nimeks `p`). Tekita faktorist `sildistatud_x` kaks faktorit `esimene` ja `teine`, kus `esimene` vastab paaride esimestele elementidele ja `teine` paaride teistele elementidele. Selleks tuleb faktorisse `esimene` korjata `sildistatud_x` esimene, kolmas jne element; faktorisse `teine` aga teine, neljas jne element. Nii vektoritest kui faktoritest kindlate elementide väljanappimiseks võib indeksisse kirjutada vajaminevate koordinaatidega vektor ning praegusel juhul sobib selle tekitamiseks hästi käsk `seq`.

Seejärel saab lugeda moodustada tabeli, mille i -ndas reas ja j -ndas tulbas on paaride arv, kus esimene element vastas sündmusele A_i ja teine element sündmusele A_j . Selleks sobib jälle `table` käsk, millele tuleb nüüd anda kaks argumenti (faktorid `esimene` ja `teine`).

Nüüd tee valmis tõenäosuste maatriksi, mille element indeksiga i, j on $p_i \cdot p_j$. Sellise tulemuse annab p komponentidest moodustatud üheveerulise maatriksi korrutamine p komponentidest moodustatud üherealise maatriksiga. Sarnast tegevust maatriksi tekitamiseks kahest vektorist läheb mitmel pool vaja ja seetõttu on vastaval operatsioonil ka oma nimi - tegemist on nn tensorskorrutisega või väliskorrutisega (*outer product*) ning R-is on selleks oma tehtemärk `%o%`. Nii et tuleb leida `p %o% p`. Selleks, et testi tulemusena korrektsed numbrid saada, tuleb esimese argumendina anda sündmuste paaride toimumise

tabelist moodustatud vektor (mille saab käsuga `c(maatriks)`) ja tõenäosuste maatriksist samamoodi moodustatud tõenäosuste vektor kujul `p=c(p_maatriks)`.

Märkusena tasub ehk mainida, et kui käsule `chisq.test()` anda ette ainult sündmusepaaride toimumise tabel maatrikskujul, siis kontrollib test samuti katsete sõltumatust, aga tabeliruumidele vastavate sündmuste tõenäosusi hinnatakse andmete põhjal (st suvandiga `p=` antud väärtuseid ei arvestata). Kuna vaadeldavas situatsioonis me teame, mis tõenäosusega iga sündmustepaar peaks esinema, on eelnevas toodud test tugevam.

Kodutöö 3 (1p, tähtaeg 27. september 2020)

Moodle VPL (Virtual Programming Lab) harjutus “Kodu3”, ülesannet näha ja esitada saab alates 20.09.2020 Lahendus esitada R skriptina (.R fail). Eelnevalt kontrollige, et lahendus töötab nõ puhtalt lehelt alustades (RStudios valida eelnevalt Session -> Clear Wokspace). Palun jälgige hoolikalt ülesande juhiseid!