Beadandó feladatok R-ből
2018. tavasz

Tudnivalók

A megoldások során néhány rövid kódot és két függvényt kell írnia.

A beadás előtt ellenőrizze a következőket (különben a beadás nem lesz sikeres):

1. A két függvényt és a kódokat egy darab R scriptben küldje, ennek neve a saját neve legyen .R kiterjesztéssel. (Például az én esetemben ez CsimaJudit.R lenne).

2. A feladatmegoldás során csak az `rstudio` alap package-eit használja.

3. A scriptekben az adatfile-ok elérése relatív path-szal történjen. Ehhez az kell, hogy a letöltött `specdata.zip` és `korhaz.zip` file-okat a working directory-jába tömörítse ki, így ott létrehozva egy `specdata` directory-t és egy `outcome-of-care-measures.csv` file-t. (Ha simán kitömöríti a letöltött zip-file-okat, akkor ez így lesz.) Ezeket relatív hivatkozással érje el a scriptekből, amit ír, ez azt jelenti, hogy például az első feladatban a `read.csv` függvényt `read.csv("./specdata/093.csv")` alakban hívja meg, a `read.csv("C:/specdata/093.csv")` nem jó.

4. Ne használjon `for`-ciklust! Minden feladat megoldható enélkül, pusztán az órán tanult ismeretek (és egy-két új függvény) segítsével. Ha mégis `for`-ciklust használ, akkor az elért pontjainak 60%-a lesz a kapott pontszáma az adott részre.

5. Csak olyan függvényt illetve kódot küldjön be, ami hiba nélkül lefordult.

Beadási határidő május 2., szerda reggel 8 óra, de természetesen korábban is el lehet küldeni, amint időm engedi adok visszajelzést róla. A beadott kódokat az utolsó laboron kell elmagyaráznia, ennek célja az, hogy meggyőződjünk arról, hogy a kód saját munka. Ha ennek ellenkezője derül ki bármilyen módon, az elégtelen osztályzatot jelent az egész tárgyra nézve. Az utolsó laboron történő bemutató előtt nézze át újra a kódokat, hogy emlékezzen arra, mit is csinálnak az egyes kódrészletek és el tudja magyarázni őket.

Ha nem sikerül teljesen megoldani egy feladatot (pl. nem tud mindent a függvény, amit kellene neki), de ér el részeredményt, azt is érdemes beküldeni, részpontokat is lehet szerezni.

Összesen 16 pontot lehet szerezni, azaz ha mindent tökéletesen megold, akkor 1 extra pontot fog kapni. (A zhkból 70, a másik laborháziból 15 pont szerezhető, így összesen 101 pontot lehet kapni, de a ponthatárokat (85, 70, 55, 40) a 100 pontos maximumból számoljuk.)
Ha bármi kérdése van a beadással kapcsolatban, akkor emailben keressen nyugodtan: csima@cs.bme.hu.

Feladatok

Töltse le a `specdata.zip` file-t a weboldalról a saját working directoryjába. Ez a tömörített file 332 CSV file-t tartalmaz, melyek mindegyike egy-egy USA-beli mérőállomás adatait tartalmazza (a file-ok egyike volt a 2. laboron használt 001.csv is). Tömörítse ki a specdata.zip file-t a `specdata` directoryba (a working directory-n belül), az első három feladatnál ezekkel a file-okkal kell majd dolgoznia.

1. Ebben a feladatban rövid kódokat kell írnia, amikkel az alábbi feladatokat lehet megvalósítani:
   (a) (1 pont) Töltse be a `093.csv` file-t, majd határozza meg, hogy ebben az előforduló szulfát értékek közül mekkora a legnagyobb és a legkisebb.
   (b) (2 pont) Határozza meg a nitrát értékek átlagát azon sorokra, ahol sem a szulfát, sem a nitrát értéke nem hiányzik.

2. (4 pont) Írjon egy `mean_nitrate` nevű függvényt, melynek egy argumentuma van, ennek neve `threshold`, és a függvény azt számolja ki, hogy mekkora a nitrát értékek átlaga a `093.csv` file azon soraira, ahol a szulfát értéke nagyobb, mint `threshold`. Ha nincsen olyan sor, ahol a szulfát értéke a `threshold`-nál nagyobb, akkor írja ki azt, hogy `Nincs ilyen sor.`

   Néhány minta input-output pár, amit reprodukálnia kell a függvénynek:

   ```
   > mean_nitrate(0)
   [1] 2.842958
   > mean_nitrate(4)
   [1] 3.390235
   > mean_nitrate(45)
   [1] "Nincs ilyen sor"
   ```

3. (4 pont) Írjon egy `good_rows` nevű függvényt, aminek egy argumentuma van, ennek neve `sorszam`, ami egy mérőállomás azonosító (1 és 332 között). A függvény először ellenőrizze, hogy a megadott `sorszam` érték a kívánt 1-332 tartományból kerül-e ki, és ha nem, akkor írja ki, hogy `hibas sorszam`. Ha a sorszám megfelelő volt, akkor pedig írja ki az adott sorszámú csv file-ban szereplő teljes sorok számát.

   Segítség (egy lehetséges megoldáshoz):
   (1) a sorszám beolvasásához nézze meg a `paste` függvényt.
   (2) nézzen utána az %in% függvénynek
   (3) az egy- és kétjegyű sorszámok átalakításához is használja a paste függvényt.

   Néhány minta input-output pár, amit reprodukálnia kell a függvénynek:

   ```
   > good_rows(543)
   [1] Error in good_rows(543) :  hibas sorszam
   > good_rows(1)
   [1] 117
   > good_rows(101)
   [1] 64
   ```

4. (5 pont) Töltse le a `korhaz.zip` file-t a working directoryjába és tömörítse ki. Két file lesz benne,

- az `outcome-of-care-measures.csv` (sok minden más adat mellett) az összes amerikai kórházra megadja a kórházi felvételt követő 30 napon belüli halálozási adatokat heart attack, heart failure és pneumonia kategóriákban (11., 17. és 23. attribútum),
- a `Hospital_Revised_Flatfiles.pdf` leírja több, a kórházak összehasonlítására alkalmas adatbázis szerkezetét (változók neve, típusa, stb.), egy ilyen adatbázis az előbbi `outcome-of-care-measures.csv`, ennek leírása a 19. pontban található (17-20. oldal).

Az adatbázist a data.medicare.gov oldalról szedtem, itt található a pdf file-ban leírt további rengeteg adatbázis is, de ezekkel most semmi dolgunk nem lesz. (Sajnos magyarul semmi hasonlót nem találtam, ezért dolgozunk az amerikai adatokkal.)

Írjon egy olyan kódot, ami az `outcome-of-care-measures.csv` file adatait felhasználva elkészít egy három sorból álló data frame-et, ahol az oszlopok az egyes államok (az oszlopok nevei megegyeznek az államok neveivel), a sorok pedig a három kategória (heart attack, heart failure és pneumonia), adott államban vett halálozási adatainak átlagát tartalmazzák. A sorok nevei legyenek HA, HF és PN.

Segítség (egy lehetséges megoldáshoz):

- a read.csv hívásakor célszerű beállítani, hogy colClasses = “character” és aztán amikor majd az átlagot számoljuk, akkor gondoskodnunk kell arról, hogy ekkor már numeric típusú legyen a megfelelő oszlop