Rakendustarkvara: R. Kevad 2018, 8. praktikum*

1 Sõnetöötlus paketiga stringr

R- i baaspaketiga on kaasas mitmeid sõnede töötlemise käske, näiteks `grep(.)` ja `substr(.)`; pikemat loetelu näeb, kui trükkida konsooli `?grep` ja `?substr`. Kahjuks nende käskude süntaks pole päris ühesugune ning mõned neist ei ole täielikult vektoriseeritud.

Pakett **stringr** proovib seda puudust kõrvaldada, pakkudes sarnase süntaksiga rohkem vektoriseeritud käske (tegemist on nn *wrapper*-funktsioonidega baaspaketi sõnetöötluskäskudele).

```
#install.packages("stringr") #  kui paketti arvutis veel pole
library(stringr) # paketi aktiveerimine
```

1.1 Sõne pikkus, sõnede kokkukleepimine ja eraldamine, alamsõne eraldamine

- Sõnede **pikkust** saab teada käsuga `str_length(.)`.
- Sõnesid saab **kokku kleepida** üheks käsuga `str_c(.)`, millel saab argumendiga `sep` määrata, milline sümbol pannakse kokkukleebitavate sõnede vahele. Kui käsule `str_c(.)` kirjutada argumendi `collapse` väärtuseks mingi sümbol, siis kleebitakse kõik sõned üheks ainsaks sõneks, mis on selle sümboliga eraldatud.
- Sõne saab **tükeldada** käsuga `str_split(.)`, mille argumendiga `pattern` saab määrata, mis on tükkide eraldaja. Selle käsu tulemusena tekib **list** , mida saab vektoriks muuta käsuga `unlist(.)`, kui aga lisada käsku argument `simplify = TRUE` on tulemuseks **maatriks**. Kui `str_split(.)` käsule anda `pattern = ""`, siis tükeldatakse sõna üksikuteks tähtedeks.
- **Alamsõne eraldamiseks** on stringr paketis käsk `str_sub(.)`, mille argumentidega `start` ja `end` saab määrata alamsõne alguse ja lõpu tärgi indeksid. Andes neile argumentidele negatiivsed indeksi väärtused alustatakse loendamist sõne lõpust st indeks `-2` märgib sõne eelviimast kohta.

```
sõnad <- c("Õun", "Apelsin", "Porrulauk", NA, "")
str_length(sõnad)
```

```
## [1]  3  7  9 NA  0
```

```
str_c(sõnad, 1:5, sep = "=")
```

```
## [1] "Õun=1"       "Apelsin=2"   "Porrulauk=3" NA            "=5"
```

```
(x <- str_c(sõnad, 1:5, sep = "=", collapse = ". "))
```

```
## [1] NA
```

```
(x <- str_c(str_replace_na(sõnad), 1:5, sep = "=", collapse = ". "))
```

```
## [1] "Õun=1. Apelsin=2. Porrulauk=3. NA=4. =5"
```

```
str_split(x, ". ")
```

```
## [[1]]
## [1] "Õun=1"       "Apelsin=2"   "Porrulauk=3" "NA=4"        "=5"
```

```
unlist(str_split(x, ". "))
```

```
## [1] "Õun=1"       "Apelsin=2"   "Porrulauk=3" "NA=4"        "=5"
```

*Praktikumijuhendid põhinevad aine MTMS.01.092 Rakendustarkvara: R (2 EAP) materjalidel, mille autorid on Mait Raag ning Raivo Kolde

```r
str_split(sõnad, "")
```

```
## [[1]]
## [1] "Õ" "u" "n"
##
## [[2]]
## [1] "A" "p" "e" "l" "s" "i" "n"
##
## [[3]]
## [1] "P" "o" "r" "r" "u" "l" "a" "u" "k"
##
## [[4]]
## [1] NA
##
## [[5]]
## character(0)
```

```r
lause1 <- "see ja teine ja kolmas ja neljas"; lause2 <- "üks või kaks või kolm"
str_split(c(lause1, lause2), c("ja", "või"))
```

```
## [[1]]
## [1] "see "     " teine "  " kolmas " " nel"     "s"
##
## [[2]]
## [1] "üks "   " kaks " " kolm"
```

```r
str_sub(sõnad, 1:5, 3:7)
```

```
## [1] "Õun" "pel" "rru" NA    ""
```

```r
str_sub(sõnad, end = -3)
```

```
## [1] "Õ"       "Apels"   "Porrula" NA        ""
```

1.1.1 Ülesanded

1. Loe sisse Massatchusettsi andmestik:
   `link <- "http://kodu.ut.ee/~annes/Rkursus/"`
   `andmed <- read.table(str_c(link, "mass.txt"), sep = "\t", header = T)`
   Andmetabelis on veerus OCCP iga inimese amet, sealjuures kolme esimese tähega on kodeeritud vastav valdkond; näiteks kõik puhastusteenustega seotud ametid algavad tähtedega CLN. Mitu erinevat valdkonda on selles andmetabelis?
2. Kasutades sama tunnnust (`OCCP`) lisa andmestikku uus veerg `OCCP1`, mille väärtused oleks samad kui `OCCP`-l, kuid töötutel oleks tunnuse väärtus kujul *"UNE-UNEMPLOYED, LAST WORKED 5 YEARS AGO OR EARLIER OR NEVER"*.

1.2 Alamsõne otsimine ja muutmine

Et teada saada, kas üks sõne sisaldub teises sõnes, saab kasutada käsku `str_detect(.)` argumendiga `pattern`. Sisaldumiste esinemiste kokkuloendamiseks sobib käsk `str_count(.)`. Juhul, kui on soov otsitava alamsõne teksti kujul leida ja väljastada, saab kasutada käsku `str_extract(.)`. Et teada saada, millisel positsioonil asub otsitav alamsõne, võiks kasutada käsku `str_locate(.)`, mis tagastab kõige esimesel positsioonil leitud alamsõne (kui sellist alamsõne üldse leidub) algus- ja lõpuindeksid **matrix**-tüüpi objektina. Kui tahame kätte saada kõigil positsioonidel olevate alamsõnede algus- ja lõpuindeksid, sobib käsk `str_locate_all(.)`, mis tagastab listi.

```
# Vaatame üle oma vektori
sõnad
```

```
## [1] "Õun"       "Apelsin"   "Porrulauk" NA          ""
```

```
str_detect(sõnad, "r")
```

```
## [1] FALSE FALSE  TRUE    NA FALSE
```

```
str_count(sõnad, "r")
```

```
## [1]  0  0  2 NA  0
```

```
str_extract(sõnad, "r")
```

```
## [1] NA  NA  "r" NA  NA
```

```
str_extract_all(sõnad, "r")
```

```
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## [1] "r" "r"
##
## [[4]]
## [1] NA
##
## [[5]]
## character(0)
```

```
str_locate(sõnad, "r")
```

```
##      start end
## [1,]    NA  NA
## [2,]    NA  NA
## [3,]     3   3
## [4,]    NA  NA
## [5,]    NA  NA
```

```
str_locate_all(sõnad, "r")
```

```
## [[1]]
##      start end
##
## [[2]]
##      start end
##
## [[3]]
```

```
##      start end
## [1,]     3   3
## [2,]     4   4
##
## [[4]]
##      start end
## [1,]    NA  NA
##
## [[5]]
##      start end
```

Mõnikord on sõnede alguses või lõpus liiga palju tühikuid, neid saab eemaldada käsuga `str_trim(.)`. Käsuga `str_pad(.)` aga saab sõne algusesse või lõppu panna tühikuid (või muid sümboleid) juurde, nii et sõne saavutaks argumendiga `width` ette antud pikkuse.

```
str_trim("    siin on palju tühjust    ")
str_pad(sõnad, width = 9, side = "both", pad = "_")
```

```
## [1] "siin on palju tühjust"
## [1] "___Õun___" "_Apelsin_" "Porrulauk" NA          "_________"
```

Kõige üldisem sõnede muutmise käsk on `str_replace(.)`, mis proovib argumendiga `pattern` ette antud ja leitud mustrit asendada argumendiga `replacement` määratud mustriga; asendatakse ainult esimene leidumine. Kõiki leidumisi saab asendada käsuga `str_replace_all(.)`

```
str_replace(sõnad, "r", "l")
str_replace_all(sõnad, "r", "l")
```

```
## [1] "Õun"       "Apelsin"   "Polrulauk" NA          ""
## [1] "Õun"       "Apelsin"   "Pollulauk" NA          ""
```

Kõigile `stringr` paketi käskudele võib argumentidega `pattern` ja `replacement` ette anda ka regulaaravaldisi[1].

1.2.1 Ülesanded

1. Massatchusettsi andmestikus on veerus `COW` ära toodud, kelle heaks inimene töötab. Kui tegemist on palgatöötajaga, sisaldab `COW` väärtus vastava inimese puhul sõna *Employee* või *employee*. Milline on palgatöötajate keskmine palk (`WAGP`)? Andmestik:
   `link <- "http://kodu.ut.ee/~annes/Rkursus/"`
   `mass <- read.table(str_c(link, "mass.txt"), sep = "\t", header = T)`
2. Eesti isikukoodi formaat[2] on järgmine: abcdefghijk, kus a – sugu ja sajand (paaritu arv mees, paarisarv naine, 1,2 – 1800, 3,4 – 1900, 5,6 – 2000); bc – aasta, de – kuu, fg – päev, hij – (haigla) sel päeval (selles haiglas) sündimise järjekord, k – kontrollnumber.
   1. Loe sisse komaga eraldatud isikukoodide jada:
      `isikukoodid <- read.table(str_c(link, "isikukoodid.txt"))[1,]`
   2. Eralda isikukoodid üksteisest ja salvesta tekkiv vektor mingi nimega.
   3. Tekita uus andmetabel (`data.frame`), millesse lisa isikukoodide veerg.
   4. Lisa veerg, kus oleks kirjas, mis sugu iga inimene selles andmetabelis on.

2 Kuupäevadega töötamine

R-is on kuupäevade jaoks `Date` andmetüüp. See on omapärane andmetüüp: näiliselt on tegemist tekstiga, ent sisuliselt arvuga – kuupäevi saab liita-lahutada, arvutada keskmist. ISO standardile vastav kuupäev on kujul aasta-kuu-päev, sealjuures aasta on nelja numbriga, kuu ja päev kumbki kahe numbriga. Sageli on

[1] http://en.wikipedia.org/wiki/Regular_expression
[2] http://et.wikipedia.org/wiki/Isikukood

sisse loetavates andmestikes aga kuupäev teisiti vormindatud. Suvalises formaadis kuupäevalise sõna saab `Date`-tüüpi väärtuseks teisendada käsuga `as.Date(.)`, mille argumendiga `format` saab määrata, millises formaadis kuupäev ette antakse. Kui formaati ei ole käsus täpsustatud, siis proovib funktsioon esmalt formaadi `"%Y-%m-%d"` ehk *aasta-kuu-päev (2018-02-01)*, seejärel `"%Y/%m/%d"` ehk *aasta/kuu/päev (2018/03/01)* sobivust, kui kumbki ei klapi antakse veateade.

```
d1 <- as.Date("22.04.2009", "%d.%m.%Y")
d2 <- as.Date("30.04.2009", "%d.%m.%Y")
d2 - d1
```

```
## Time difference of 8 days
```

```
as.numeric(d2 - d1)
```

```
## [1] 8
```

Näites kasutatud formaadi tähiste `%d`, mis tähendab päeva numbrit, `%m` mis näitab kuu numbrit, `%Y` mis tähistab neljakohalist aastanumbrit ja teiste formaadi võimaluste kohta saab täpsemalt lugeda käsu `strptime(.)` abifailist `?strptime`.

Kuupäevaks formaaditud objektidega saab teha kõiki mõistlikke operatsioone, näiteks leida miinimum, keskmine, võrrelda hulki:

```
d3 <- Sys.Date()
paevad <- c(d1, d2, d3)
mean(paevad)
```

```
## [1] "2012-04-11"
```

```
d2 %in% paevad
```

```
## [1] TRUE
```

Küll aga ei saa näiteks leida kuupäevast logaritmi või ruutjuurt.Põhjus on selles, et R talletab kuupäevi tegelikult päevade arvuna alates nullpunktist, sealjuures nullpunktiks loetakse vaikimisi 1970-01-01. Selles võime veenduda `as.numeric(.)` käsku kasutades.

```
d1:d2
```

```
## [1] 14356 14357 14358 14359 14360 14361 14362 14363 14364
```

```
as.numeric(as.Date("1970-01-02"))
```

```
## [1] 1
```

```
as.numeric(as.Date("1969-12-30"))
```

```
## [1] -2
```

Date-tüüpi muutuja väärtuse saab sobival kujul sõneks teisendada käsuga `format(.)`, see võimaldab kombineerida teksti ja kuupäeva elemente:

```
format(Sys.Date(), "Kuupäev: %d. %B, (aastal %Y). Nädal nr %V.")
```

```
## [1] "Kuupäev: 13. märts, (aastal 2018). Nädal nr 11."
```

2.0.1 Ülesanded

Vaatame edasi isikukoodide vektori (`isikukoodid` eelmisest ülesandest) põhjal tekitatud isikute info andmestikku.

1. Lisa isikute andmestikku veerg, mis sisaldaks iga inimese sünnikuupäeva *Date*-tüüpi muutujana.
2. Lisa veerg, mis annab inimese vanuse täisaastates tänasel päeval (arvestades, et aastas on ~365,25 päeva).
3. Leia kuupäev, mil said/saad 10 000 päeva vanuseks.