Tallinna Ülikool
Digitehnoloogiate Instituut

Lingvistika analüüs R-keele abil: Õppematerjal

Bakalaureusetöö

Autor: Magnus Kvell
Juhendaja: Jaagup Kippar

Autor: ....." ....."2016
Juhendaja: ....." ....."2016
Instituudi direktor: ....." ....."2016

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

…… …….(kuupäev) (autor)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina _____ (sünnikuupäev: _____)

(*autori nimi*)

1. annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
   _____
   _____
   _____

   (*lõputöö pealkiri*)

   mille juhendaja on _____,

   (*juhendaja nimi*)

   säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositooriumis.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tallinnas/Haapsalus/Rakveres/Helsingis, _____

(digitaalne) allkiri ja kuupäev

Sisukord

Sissejuhatus

Mujal maailmas populaarsust koguv statistika töötlemise jaoks mõeldud programmeerimiskeel R on Eestis seni vähetuntud. Samas saab keelt kasutada efektiivselt paljude sotsiaal ja reaal valdkonnas esinevate statistiliste andmete analüüsiks. Selle bakalaureusetöö eesmärk on anda ülevaade R-programmeerimiskeele kasutamisest lingvistika analüüside tegemisel õppematerjali koostamise näol.

Autorile teadaolevalt on käsitletud R-keele aluseid küll mõnes kõrgkooli õppekavas, kuid siiamaani eestikeele lingvistiliste analüüside tegemiseks, keelt laialdaselt ei ole kasutatud, pigem tehakse keeleteaduse statistilisi analüüse Python-i või mõne teise programmeerimiskeelega. Õppematerjali põhieesmärk on tuua välja kasutusvaldkonnad lingvistikas, kus oleks R-keele kasutamine efektiivsem, kui mõnes muus keeles või programmis. Peale selle tutvustab õppematerjal lingvistika analüüsides kasutatavaid meetodeid ja seaduspärasusi üleüldiselt.

Õppematerjali läbinud õppur saab sissejuhatuse lingvistikasse ja selle analüüsimisse R-programmeerimiskeele kaudu. Õppematerjali üheks lisaeesmärgiks tulevikus oleks leida R-programmeerimiskeele jaoks kasutus Eesti vahekeelekorpuse kodulehe tulevastes analüüsides, sest selle efektiivne täiendamine on hetkel läbiv teema. Teiseks lisaeesmärgiks on õppematerjali kasutamine tulevikus R-keele ja lingvistika õpetamisel mõnes ainekursuses.

Eesmärgi saavutamiseks tutvub autor nii olemasolevate võõrkeelsete õppematerjalidega, kui ka lingvistika üleüldise analüüsiga. Veel õpib autor selgeks R-keele piisaval tasemel, et lingvistilisi analüüse läbi viia ja selle kohta teadmisi edasi anda. Viimaseks tutvub autor õppematerjali loomise tavade ja protsessidega.

Bakalaureusetöö koosneb kolmest peatükist. Esimeses peatükis antakse ülevaade R-programmeerimiskeelest ning olemasolevatest õppematerjalidest. R-programmeerimiskeele ülevaates uuritakse programmeerimiskeelt ennast ja ka teisi vahendeid, millega saab teha lingvistika analüüse. Sarnaste õppematerjalide alapeatükis uuritakse teisi olemasolevaid võõrkeelseid materjale.

Teises peatükis analüüsib autor, kuidas ta ise omandas uue õppurina teadmisi, et saavutada lõppeesmärk ehk õppematerjal. Analüüsi käigus teeb autor järeldused, mis teed peaks uus õppur valima, et omandada teadmisi antud valdkonnas.

Kolmandas peatükis analüüsib autor loodavat õppematerjali. Analüüs toimub õppematerjali enda, õppematerjali vajaduste ja tulevaste õppematerjali kasutajate põhjal. Peale seda kirjeldab autor, kuidas võiks õppematerjali tulevikus täiendada ja kust selle protsessiga edasi minna.

Õppematerjaliga soovib autor kaasata uued õppurid keeleteaduse valdkonda ja selle analüüsi programmeerimiskeele kaudu.

1 R-programmeerimiskeel ja lingvistika

Selles peatükis tuleb juttu R-keskkonna omadustest ja miks on autor valinud lingvistilisteks analüüsideks selle programmeerimiskeele. Peale seda annab autor ülevaate teistest vahenditest mida saab kasutada statistiliste analüüside tegemiseks ja olemasolevast kirjandusest. Ülevaates analüüsib autor vahendite ja kirjanduse positiivseid ja negatiivseid külgi. Vahendite puhul tuleb analüüsi käsitlusse nende võrdlus keelega R.

1.1 R-programmeerimiskeele ja keskkonna tutvustus

R on keel ja keskkond statistilisteks arvutusteks ja graafilisteks analüüsideks. Keel R on GNU projekt, mis on sarnane S-keele keskkonnale ja mis arendati Bell Laboratiores poolt. Keelt R võib võtta, kui teistsugust keele S realiseerimist. Keelte R-i ja S-i vahel on suuri erinevusi, aga enamus R-i koodi jookseb ka S-i keskkonnas. Keel R pakub palju erinevaid valikuid statistiliseks ja graafiliseks andmete töötlemiseks ja analüüsimiseks ning on väga paindlik. Programmeerimiskeele tugevusteks on lihtsus graafiliste diagrammide loomisel ja matemaatiliste funktsioonide läbiviimisel. Keel on vabavaraliselt saadaval ja jookseb enamus UNIX-i platvormidel ja sarnastel süsteemidel, näiteks Linux, MacOS .

R-keskkond pakub sisse ehitatud vahendeid järgmisteks tegevusteks:

- Andmete hoiustamiseks.
- Operatsioonid massiividega.
- Andmete töötlus vahendeid.
- Graafilisi vahendeid andmete analüüsiks nii ekraani peal, kui ka failidesse kirjutatuna.
- Arendatud programmeerimiskeelt.

R-keele vahendeid saab täiustada loodud teekide juurde paigaldamisega, mis annavad lisavõimalusi erinevateks statistilisteks analüüsideks ja andmete töötlemiseks(What is R?, kuupäev puudub).

Lingvistika analüüs põhineb suurel hulgal statistilistes analüüsides, sellepärast on autor valinud selle programmeerimiskeele keeleteaduslike analüüside läbiviimiseks ja õpetamiseks.

1.2 R-keele kasutusvaldkonnad

R-keele põhifunktsionaalsus on statistilised uuringud ja statistiliste andmete töötlus. R-keelt saab kasutada ka muudes valdkondades. Mujal maailmas kasutatakse keelt suures ulatuses ärivaldkonnas. Näiteks kasutavad R-i paljud suured pangad krediidianalüüside tegemiseks (Companies Using R, kuupäev puudub). R-i saab kasutada igas valdkonnas, kus läbivaks teemaks on statistilised analüüsid.

Mille poolest keel erineb teistest programmeerimiskeeltest on see, et keele kasutusvaldkond on mõningal määral piiratud. R ei sobi suuremahuliste programmide kirjutamiseks, sest see kasutab suurel hulgal vahemälu. Sellest tulenevalt tuleb õppida keelt targalt kasutama, sest kogenuma R-i programmeerija ja algaja koodi kirjutaja programmi mälukasutus võib erineda mitmekordselt.

1.3 R-keele kasutus lingvistika analüüsides

Selles alapeatükis kirjeldab autor, kuidas saab kasutada programmeerimiskeelt R keeleteaduses. Keel ise on põhiliselt statistiliste analüüside läbiviimise jaoks, aga keelt saab kasutada ka suurtes kogustes andmete töötlemisseks. Andmeid saab R-i lugeda kõigist populaarsematest failitüüpidest. Lisa teekide installeerimisel, toetavate failide arv tõuseb.

Põhi ülesanded, millega keel R lingvistika analüüsides hakkama saab

- Lingvistika statistika uurimine.
- Statistika olulisuse testid.
- Sõnavara kasutuse uurimine.
- Lingvistika andmete töötlemine.
- Graafiline andmete analüüs, graafikute loomine.

Need on suuremad ja üldisemad teemad, mida saab uurida läbi R-programmeerimiskeele keeleteaduses. Autori loodavas õppematerjalis seletatakse need teemad pikemalt lahti, iga vastava teema kohta on harjutused ja ülesanded.

1.4 Alternatiivid keelele R

Otsest alternatiivi R-keelele ei ole, on vaid erinevad statistilised tarkvarad ja teised programmeerimiskeeled. Lingvistika analüüside läbi viimise võimalusi pakuvad erinevad veebilehed ja koostatud programmid, kuid nende funktsionaalsused on piiratud ja teatud analüüse nendega läbi viia on raske, sest erinevate hüpoteeside puhul on lahenduste leidmiseks erinevad meetodid. Siin toob autor välja kõige lähedasemad vahendid R-keelele, lingvistiliste analüüside tegemiseks.

Python

Python on programmeerimiskeel nagu R. Python on R-ile kõige lähedasem konkurent ja hea abivahend keeleteaduslikeks ja statistilisteks analüüsideks. Keel ise on lihtne ja laiaulatuslik, sellepärast sobib see hästi ka keeleteaduslikeks ning statistilisteks analüüsideks. Python-i ja R-keele vahe on selles, kui R-keel on fokuseeritud enamjaolt ainult statistilisteks arvutusteks, siis Python on üleüldiselt kodeerimiseks. Pythonit iseenesest kasutatakse laiaulatuslikumalt lingvistika uuringuteks, sest keel on populaarsem, kui R. R-keele tugevus üle Pythoni on koodi lihtsus ja graafilised abivahendid diagrammide ja jooniste näol.

Lokaalselt Eestis on Python-i õppimiseks olemas mitmeid õppematerjale samas, kui R-keele kohta eestikeelsed õppematerjalid praktiliselt puuduvad. Peale selle on Pythoni jaoks välja töötatud Tartu Ülikoolis eestikeelsete tekstide analüüsimiseks lisapakett Estnltk(EstNLTK). Sellest tulenevalt on Pythoniga eestikeelse teksti või korpuse analüüsimisel eelis, teatud analüüside tegemisel, näiteks morfosüntaktiline analüüs. Keeleteaduslike statistiliste analüüside puhul on R-keel lihtsam, sellepärast et enamik vajaminevaid valemeid saab välja kutsuda ühe käsu abil.

Matlab

R-keelt tihti võrreldakse Matlabiga, mis on ka keel ja keskkond. Matlab ei ole vabavaraline ja litsents on kallis. Matlab-i tugevuseks R-i üle on füüsikalise teadusega tegelemine, samas on R statistika osas tugevam. Kuna tegemist on kommertstarkvaraga, siis on Matlab-i lisapaketid kvaliteetsemalt arendatud, kuid need on ka tasulised. Keeleteadulikeks uuringuteks sobib

R-keel paremini, sest lingvistika uuringuteks on statistika analüüs R-keeles paindlikum. Matlab-i ja keeleteadusliku analüüsi kohta on õppematerjalid väga puudulikud. Autoril endal antud keskkonnas praktilised kogemused puuduvad. (MATLAB, 2016)

SPSS

SPSS on statistika tarkvara, mis on laialt kasutatav sotsiaalvaldkondades. SPSS-is on võimalik teha lingvistika analüüse läbi programmi sissehitatud funktsionaalsuste. Programm saab hakkama näiteks keskmiste arvutamisega, t-testidega ja sageduste leidmisega. SPSS annab nagu R lihtsa võimaluse luua diagramme ning läbi viia statistilisi arvutusi.

Programmi eelis R-i ees on kasutajasõbralikus. Programmi puuduseks on see, et SPSS-is kasutatav programmeerimiskeel ei ole nii võimekas nagu R. Sellest tulenevalt on SPSS-i funktsionaalsused piiratud. Lihtsamaid analüüse on SPSS tarkvaraga kiirem läbi viia, keeruliste analüüside jaoks sobib R rohkem. (IBM, 2016)

SAS

SAS on statistika analüüsi ja andmete töötlemise programm. Programmi funktsionaalsuste hulka kuulub ka sisseehitatud programmeerimiskeel. SAS on selles peatükis olevatest vahenditest kõige populaarsem ärivaldkonnas. Programmiga on võimalik samamoodi lingvistika analüüse läbi viia nagu R-iga. SAS programmi eelis on kasutajasõbralikus ja põhifunktsionaalsuste rohkus, siinkohal tuleb keskkonna probleemiks samamoodi nagu SPSS tarkvara puhul see, et tegu ei ole programmeerimiskeelega vaid pigem programmiga. Sellest tulenevalt on programmis keeruliste analüüside läbiviimine ajakulukam.

SAS on R-keelest populaarsem vahend analüüside läbi viimiseks just ettevõtetes. Üks selle põhjustest on see, et R on vabavaraline ja ettevõtted ei usalda vabavaralist tarkvara. Teine põhjus tuleb traditsioonidest, sest programmi on ettevõtete poolt ressursse investeeritud. Kolmandaks põhjuseks on see, et R-keel viib läbi kõik operatsioonid vahemälu kasutades, samas SAS ei pruugi seda teha, sellest tulenevalt saavad programmis suure mahulised andmed kiiremini läbi töödeldud. (What Is the SAS System?, kuupäev puudub)

1.5 Inglisekeelsete õppematerjalide ülevaade

Statistics for Linguistics with R: A Practical Introduction

Autor: Gries, S. T

Aasta:2013

Tüüp:raamat

See raamat on esimene, mida autor kasutas lingvistika ja keele R õppimiseks, teos on inglisekeelne. Raamat on sobilik algajatele, kes tahavad lingvistika statistiliste uuringutega tegeleda. Raamatut autor otseselt enda töös eeskujuks ei võtnud, aga alustas sellest teema kohta algteadmiste saamiseks.

Raamat koosneb viiest põhipeatükist, kus antakse ülevaade järgmiste teemade kohta:'

- Empiiriliste uuringute alused
- R-keele alused
- Kirjeldav statistika
- Analüütiline statistika
- Välja valitud multifaktoriaalsed ja mitme muutujaga meetodid

Raamatu positiivseks küljeks on, teksti ja harjutuste lihtsus, negatiivseks raamatu liigne teoreetilisus. Mitmeid teoses olevaid ülesandeid ei saa eestikeelse teksti töötlemisel aluseks võtta, sellepärast et keele reeglistikud on teised. Seda õppematerjali võib pigem käsitleda, kui abivahendit keeleteaduslike hüpoteeside loomiseks. (Gries, S. T., 2013.)

Analyzing Linguistic Data: A Practical Introduction to Statistics using R

Autor: Baayen, R. H

Aasta: 2008

Tüüp: Raamat

Õppematerjali luues võttis autor eeskujuks selle raamatu, sest see on praktilisem, kui „*statistics for linguistics with" R*. Raamat sobib algajatele, aga ka edasijõudnud lingvistidele. Teose ülesehitus on kujul: teoreetiline osa ning pärast seda praktilised ülesanded. Raamatus antakse hea ülevaade, lingvistika statistilisest poolest.

Teos tekitas autoris lingvistika vastu huvi, sest selles antakse mahukas ülevaade seaduspärasustest ja eripäradest, mida lingvistika andmete analüüsimisel kasutada. Mitmed harjutused raamatus tehakse läbi tuntud ilukirjanduslike teostega. Teoses antakse lühiülevaade R-keele alustest, seda tehakse väga lühidalt ja raamatu keerukus on mõningate ülesannete puhul raske. Autor soovitab seda teost lugedes jälgida kasutatud kirjanduse viiteid või iseseisvalt leida läbivate teemade kohta materjali (Baayen, R. H.,2008).

Statistical Inference – a Gentle Introduction for Linguists(SIGIL)

Autorid: Baroni, Marco ja Evert, Stefan

Aasta:2007

Tüüp: e-õppematerjal

Õppematerjali kestvus ei ole väga pikk ja õppematerjal on sissejuhatav. Materjal koosneb slaididest, mis sisaldavad teooriat, läbitehtavate harjutuste koodi ja ülesandeid. Õppematerjal sobib algajale õppijale, kes tahab saada baas teadmisi antud valdkonnas.(Baroni,M., Evert.S., 2015)

Materjali leiab aadressilt: http://www.stefan-evert.de/SIGIL/ .

2 Praktilise informatsiooni kogumise analüüs

Enne teema käsitlemist, autoril puudus igasugune kogemus lingvistika analüüside käsitlemisel, sellepärast oli ettevalmistus periood kõige pikem protsess. Selles peatükis annab autor vastuse küsimusele, kuidas ta ise viis end uue õpilasena lingvistika valdkonnaga kurssi. Õppeprotsessi analüüsi viib autor läbi, et rakendada seda õppematerjali koostamisel.

2.1 Keeletehnoloogia konverents

Esimese sammuna, et ennast teemasse sisse viia osales autor Tartus AHHAA keskuses toimunud keeletehnoloogia konverentsil „Eesti keeletehnoloogia 2015“. Konverents oli kahepäevane ja andis ülevaate erinevatest keeletehnoloogilistest lahendustest, autor ise osales ühel päeval. Konverentsi lõpus sai praktikas proovida erinevaid arendatavaid lahendusi töötubades järgi. Konverentsi käigus omandas autor teadmisi keeleteaduses olevate eestikeelsete tehniliste lahenduste kohta ja probleemide kohta, mis tekivad programmide arenduse käigus. Näiteks suurematekst probleemideks on projektide rahastus ja pikk aeg, mis ühe keeletehnoloogilise abivahendi väljatöötamiseks läheb.

Kõige enam pakkus huvi autorile Pythonis valmistatav estNLTK teek, sest selle teegi funktsionaalsus on teksti analüüs. Tänu sellele sai autor jälgida, mis funktsionaalsused on juba välja töötatud, mida võib-olla saaks lihtsustada või täiendada R-keele abil, näiteks morfosüntaktiline analüüs. Lisapakett ise on veel arendamisjärgus, sellest tulenevalt on arendajatel kavas täiendada selle funktsionaalsusi. (EstNLTK, kuupäev puudub)

Ülejäänud konverentsil näidatud projektid puudutasid pigem üleüldisemaid tehnoloogilisi lahendusi, näiteks sõnaseletus kogumikke ja audio kõne uuringuid. Üleüldiselt oli autorile oli selline esmakogemus suur huvi tekitaja keeletehnoloogia valdkonna vastu ja hea lähtepunkt kust alustada.

2.2 R-keele õppimine

Uue programmeerimiskeele õppimine on ajakulukas ning nõuab pühendumust. Selle programmeerimiskeele põhitõdede õppimiseks kasutas autor internetis vabavaraliselt

saadaolevaid õppematerjali ja dokumentatsiooni, mis on üleval nii raamatu näol, kui ka keelekeskkonnas endas. Autoril oli vaja pühenduda ainult ühte valdkonda, sellepärast oli eesmärgiks ära õppida esmalt ainult põhifunktsionaalsused. Teine programmeerimiskeele õppimisperiood oli teema kohase analüüsitava kirjandusega tööd tehes.

Autori seisukohast uue programmeerimiskeele õppimiseks kõige efektiivsem moodus oleks käija praktilistel kursustel, sellepärast et seal saaks juhendaja käest küsida nõu, kui harjutuste arusaadavus tundub keeruline. Tänapäeval ei ole kirjanduse kaudu millegi õppimine enam väga keeruline, sest e-õppematerjalide rohkus järjest tõuseb ja lisanõuandeid saab küsida foorumitest.

2.3 Lingvistika analüüside õppimine

Selles etapis lähtus autor põhiliselt kahest raamatust „*Statistics for Linguistics with R: A Practical Introduction* ja *Analyzing Linguistic Data"* ja *A „Practical Introduction to Statistics using" R*. Mõlemad raamatud on õppematerjalid statistiliste analüüside läbiviimise kohta R-keeles. Raamatuid lugedes pidi autor kõrvalt informatsiooni otsima seaduspärasuste ja teoreemide kohta.

Materjali analüüsides tuli välja kaks suuremat probleemi, üks nendest oli see, et materjal oli inglisekeelne ja teine, et analüüsitavad tekstid olid võõrkeelsed. Autoril oli eesmärk teha eestikeelne õppematerjal kasutades eestikeelseid tekste, tänu sellele oli vaja analüüsida, mis funktsionaalsusi materjalist saab eeskujuks tuua.

3 Õppematerjali loomine

Bakalaureusetöö eesmärgiks on luua õppematerjal, mis annaks õppurile algteadmised praktiseerimaks lingvistika analüüse R-programmeerimiskeeles ja keskkonnas. Õppematerjali põhieesmärk on analüüsida R-keele funktsionaalsusi lingvistika uuringutes.

Bakalaureusetöös valminud õppematerjal ei käsitle R-keele kõiki algteadmisi, vaid tegeleb lingvistika analüüse puudutavate funktsionaalsustega. R-keelde sissejuhatuse saamiseks soovitab autor lugeda e-õppematerjali näiteks lehelt:
http://andmeteadus.github.io/2015/rakendustarkvara_R

Eesmärgi täitmiseks on võtnud autor eeskujuks samalaadsed inglisekeelsed materjalid, mis on sama teema kohta ja üleüldised eestikeelsed programmeerimiskeelte õppematerjalid. Õppematerjali tehes on autor lähtunud ADDIE mudelist.

Materjal koosneb kahest osast, harjutuste läbimisest ja pärast igat peatükki ülesannete lahendamisest. Ülesanded on kavandatud autori poolt vastavalt peatükis olevale teemale. Autor alustab harjutustega raskusastme järgi, eelnevalt sissejuhatuseks kergemad tekstifunktsioonid ja hiljem keerukamad statistilised analüüsid. Läbi ülesannete lahendamise omandab õppur peatükis läbitud teema.

Õppematerjali teemade valimisel pidas autor oluliseks järgmisi valikuid:

- teemade populaarsust
- teemade vajadust
- eestikeelseteksti analüüsimisvõimalust
- statistilist olulisust

Autor valis õppematerjali formaadiks algselt PDF formaadi ning hiljem on kavas töö lisada e-õppematerjalina. PDF formaat on hea algus, sest tänu sellele formaadile on töö levitamine lihtsam. Faili formaati saab kergelt printida või üleslaadida vabalt valitud keskkonda.

3.1 Õppematerjali vajadus

Õppematerjali vajadusest lähtus autor esiteks juhendaja soovist õppematerjali loomisel. Juhendaja tegeleb lingvistika alaste teemadega ning soovis R-programmeerimiskeele õppematerjali. Sellest tulenevalt tuli autoril idee teemad kokku panna ja luua lingvistika alane õppematerjal R-programmeerimiskeeles. Seni eestikeelsed õppematerjalid antud teema kohta puuduvad.

Õppematerjal on esmalt mõeldud inimestele, kellel on huvi keeleteaduse vastu ja kes soovivad katsetada, kas teksti või korpuste analüüsi kaudu erinevaid hüpoteese. Sellest tulenevalt annab R-programmeerimiskeel ühe lisavõimaluse lingvistika analüüside tegemiseks juurde, mida Eestis veel laiaulatuslikult ei kasutata.

3.2 Sihtrühm

Selles alapeatükis kirjeldab autor, millised eelnevad oskused oleksid vajalikud õppematerjali läbival õppuril. Veel tuleb jutuks, kellele on õppematerjal suunatud.

Enne õppematerjaliga tööd alustamist peaks õppur:

- huvi tundma keeleteaduslike analüüside vastu.
- omama algteadmisi keeles R.
- Omama ettekujutust lingvistika analüüsidest.
- Teadma lingvistika seaduspärasusi.
- Omama algteadmisi programmeerimiskeelte loogika kohta.

Ülal olevad punktid ei ole kohustuslikud, vaid aitavad õppuril materjali paremini mõista. Õppematerjal on suunatud kõigile, kellel on huvi lingvistiliste analüüside tegemise vastu.

3.3 Õppematerjali ülesehitus

Selles alapeatükis annab autor ülevaate õppematerjali üleseehitusest. Õppematerjal koosneb kuuest peatükist, igas peatükis käsitletakse ühte lingvistika analüüse puudutavat teemat, teoreetiliselt ja praktiliselt harjutuste näol, harjutusi võib vastavalt peatükile olla rohkem kui üks. Iga peatüki lõpus on ülesanne või ülesanded, mille peaks õppur lahendama.

Harjutused on loogilises järjestuses, iga eelnev harjutus täiendab järgnevat. Autor on valinud harjutused vastavalt olemasolevale kirjandusele ja programmeerimiskeelt arvestades koodi kirjutamise efektiivsusele. Koodi osas arvestas autor funktsionaalsusi valides koodi keerukust, autor lähtus sellest, kui palju on vaja koodi kirjutada, et vastav eesmärk täita.

3.4 Õppematerjali kujundus

Käesolevas alapeatükis kirjeldab autor, milliseid lahendusi kasutas ta valminud õppematerjali kujundamisel. Õppematerjal on kujundatud üldiselt samade reeglite järgi nagu käesolev bakalaureusetöö. Kujunduse kohapeal pani autor enim rõhku koodile. Programmeerimiskood on ära märgitud hele halli värvusega, et see oleks lugejale paremini märgatav harjutust ümber kirjutades või testides. Iga harjutuses saadud väljund on kujutatud koodinäite allosas, et õppur saaks koheselt võrrelda autori ja enda saadud tulemust. Õppematerjali lõpus märgib autor ära, millist lisakirjandust võiks õppur lugeda.

Enne igat koodinäidet on õppematerjalis teooria osa, mis annab ülevaate koodis kasutatud funktsioonidest ja funktsioonide tulemusest.

3.5 Õppematerjali ülevaade

Selles alapeatükis kirjeldab autor lühidalt, millistest peatükkidest valminud õppematerjal koosneb. Õppematerjali koosneb kuuest teemast ja õppematerjali läbimise eeldatav hetkepikkus on 1-3 tundi. Õppematerjali eeldatav pikkus on oletatav autori kogemustest.

Esimeseks peatükiks valis autor harjutused tekstifunktsioonidega. Inglisekeelsetes samalaadsetes materjalides on selle teema kohta vähe kirjutatud, kuid teema on tihedalt seotud tekstianalüüsimisega. See harjutus on mõeldud sissejuhatava peatükina R-keele loogikasse ja ülevaatusena R-keele tekstitöötlus võimekusse. Funktsioonid harjutustes leiab üles ka dokumentatsioonist, õppematerjalis näidatakse neid praktiliselt.

Teine peatükk annab ülevaate Text Mining teegist ja lühidalt selle teegi võimalustest. Autor valis selle teegi harjutuste jaoks, sest lisapakett on väga populaarne vahend korpustega töötamisel. Lisapakett annab võimaluse sisse lugeda korpuse kaustast ühe käsu abil. Korpuse

analüüsimiseks on teegis käsud, mille abil saab teha mitmeid analüüse kiiresti ja nende järgi luua graafikuid või maatrikseid. Harjutuste ülesanne on anda ettekujutus teegiga töötamiseks.

Kolmas peatükk käsitleb graafilisi tööriistu andmete vaatlemiseks. Kolmandas peatükis õpib õppematerjali lugeja, kuidas saada andmetest sõnade sagedused ja kuidas teha erinevaid diagramme. Peatükis on näidetena läbi tehtud kaks harjutust tulpdiagrammi ja punktdiagrammi näol. Harjutused läbinud õpilane saab ettekujutuse, kuidas töötada R-keele graafiliste vahenditega.

Neljandas peatükis käsitleb autor statistilist tõenäosust lingvistika analüüsides. Statistilise tõenäosuse puhul tulevad jutuks p-väärtused ja nende olulisus lingvistika statistikas. Veel annab autor ülevaate binoomjaotustest ja nende kasutusest lingvistika analüüsimisel. Peatüki läbinu saab sissejuhatuse lingvistika statistilisse analüüsi ja oskab iseseisvalt loodud hüpoteese läbi viia.

Viies peatükk käsitleb sõnavararikkuse seaduspärasusi. Viiendas peatükis lahkab autor kahte lingvistika seaduspära sõnavararikkuse analüüsimiseks Heapi ja Zipfi seaduseid. Seaduspärasuste analüüsimine toimub läbi erinevate sõnade sageduste mõõtmise ning nende võrdlemise. Peatüki läbinu oskab oma andmeid analüüsides kasutada seaduspärasusi teksti normaalsuse mõõtmiseks. Vastav teema on käsitletud õppematerjalis, sest seaduspärasuste kaudu teksti analüüs on üks lingvistiliste analüüside aluseid.

Kuues peatükk kasutab Eesti vahekeelekorpuse kodulehel olevat morfosüntaktilist analüüsivahendit, harjutuses leitakse analüüsi tulemustest sõnaliigid. Sarnase harjutuse on originaalselt koostanud õppejõud ja juhendaja Jaagup Kippar, autor võttis peatüki aluseks tema koostatud harjutuse. Harjutus annab ülevaate, kuidas saada teistest abivahenditest R-keele abil kätte statistikat ja seda R-keele keskkonnas analüüsida edasi.

3.6 Õpiväljundid

Selles alapeatükis kirjeldab autor, millised teadmised omandab õppematerjali läbinud õppur ja kuidas õpinguid jätkata.
Õppematerjal on loodud kas R-keskkonnas või ideaalis RStudio lisakeskkonnas kasutamiseks. Õppematerjali lugedes soovitab autor kasutada lisa materjalina interneti allikaid või leida kirjandust raamatute näol, mis on õppematerjalis ära märgitud.

Õppur, kes on materjali läbinud saab algteadmised järgmiste teemade kohta:

- Tekstifunktsioonid keeles R.
- Korpuste käsitlemine.
- Lingvistika analüüside tegemine, sõnaseguste järgi.
- Sõnavara rikkuse leidmine tekstist.
- Statistilistest seaduspärasustest Zipfi ja Heapi seaduse näol.

Õppur, kes on õppematerjali läbinud, võib teemaga edaspidiseks tutvumiseks lugeda soovitusliku kirjandust, mis on ära märgitud õppematerjalis. Õppematerjal annab teadmised, kuidas iseseisvalt lahendada hüpoteese ja tekste analüüsida, sellest tulenevalt võib õppur edasi minna iseseisvalt harjutuste lahendamisega. Eestikeelsed õppematerjalid antud teema kohta puuduvad, samas võib õppur uurida eestikeelseid loodud lingvistika analüüsimis õppematerjale, mis ei ole seotud R-programmeerimiskeelega ja neid kasutades proovida analüüse teha R-programmeerimisekeeles.

Õppematerjali jätkuks on autoril kavas viia õppematerjal, e-õppematerjali kujule ja tulevikus seda täiendada. Autor plaanib ise antud teemat edasi õppida, sellest tulenevalt ka õppematerjali lisada tulevikus läbi viidud harjutusi.

Hetkel on õppematerjal suuremas osas testimata, sest autor ei leidnud piisavalt testijaid õppematerjali jaoks enne bakalaureuse töö esitamist. Testimine on kavas tulevikus läbi viia, et õppematerjali koodi pooles vigu leida ja teooria osa paranda. Testijate leidmisega oli probleeme sellepärast, et R-keelt Tallinna Ülikooli õppekavas see aasta veel ei õpetata sellisel kujul, et loengutes saaks valminud õppematerjali läbi teha. 2016 kevadel oli R programmeerimiskeel küll läbiv teema XMLi loengutes, aga autori õppematerjali on seal

raske kasutada, sest see sisaldab pigem puhast R-keelt ja keele keskkonna kasutamist. Sügisel 2016 on kavas õpetada R-programmeerimiskeelt õppekavas lisaainena, kus võiksid mõned harjutused õppematerjalist kasutust leida.

Kokkuvõte

Käesoleva bakalaureusetöö põhieesmärgiks oli koostada õppematerjal, et anda sissejuhatus lingvistika analüüside tegemiseks R-programmeerimiskeele kaudu. Õppematerjali tegemise põhjuseks oli eestikeelsete õppematerjalide puudus antud teema kohta keele vähese populaarsuse tõttu Eestis. Peale selle on õppematerjali loomise eesmärgiks tulevikus kasutuse leidmine õppekursustel.

Bakalaureusetöö käigus valmis õppematerjal, milles autor käsitleb sissejuhatavalt lingvistika analüüside läbiviimist. Õppematerjalis annab autor lühiülevaate R-keele loogikast ja lingvistiliste analüüside ideedest.

Enne õppematerjali loomist analüüsis autor olemasolevat kirjandust ja R-keele kasutatavust lingvistikas. Autor tutvus erinevate teiste võimalustega, mida saaks kasutada lingvistika analüüside läbiviimisel. Lühiülevaatena kirjeldas autor bakalaureusetöö käigus toimunud õpiprotsessi.

Õppematerjali loomise käigus sai autor uusi kogemusi õpitava programmeerimiskeele ja lingvistika alaste analüüside tegemise näol. Bakalaureusetöös esinenud probleemideks õppematerjali koostamisel oli autori eelnev vähene kogemus teemaga tegelemisel ja töö testijate õigeaegne leidmine.

Bakalaureusetöö käigus loodud õppematerjal annab tulevastele keeleteaduse analüüsijatele ühe lisavõimaluse analüüside läbiviimiseks.

Summary

Title: Analysing Linguistics with R. A Learning Material

The main aim of this Bachelor Thesis was to create a learning material that provides an introduction to linguistics analyses with R programming language. Since Thesis author could not find any learning materials about the subject in Estonian language, author decided to greate a learning material himself. Another reason for creation the need for a learning material in upcoming R-language course.

In the course of this Bachelor Thesis, author will finish a study material, what provides a small introduction to linguistics analyzes with R language. In the study material author provides an introduction to R-language logic and linguistics data analyzing ideas.

Before creating a study material, author analysed literature about the subject and how programming language R could be useful in linguistics. Author also familiarized himself with other means for linguistics analyses.In the process of study material creation, author gained new experience about R-language and linguistics in general.

Main problems what accured during creation process were the authors lack of prior experience about the subject and finding testers for the study material. The study material, what was created during this Thesis gives an overview about the subject to future learners, who are intrested in analysing linguistics through a programming language.

First chapter of this Bachelor Thesis provides an overview about R-language usage, analyses alternatives and other available study materials. Second chapter introduces authors learning process about R-language and linguistic analyses. Third chapter provides an analyses of the Study material creation process.

Kasutatud kirjandus

R: What is R? .Kasutamise kuupäev 20.aprill 2016.a., allikas https://www.r-project.org/about.html

EstNLTK: Pythoni teegid eestikeelsete vabatekstide lihtsamaks töötlemiseks — Eesti keeles. Kasutamise kuupäev 25.aprill 2016.a., allikas https://www.keeletehnoloogia.ee/et/ekt-projektid/estnltk-pythoni-teegid-eestikeelsete-vabatektside-lihtsamaks-tootlemiseks

MATLAB. (2016)- MathWorks - MathWorks Nordic. (n.d.). Kasutamise kuupäev 21.aprill 2016.a., allikas http://se.mathworks.com/products/matlab/

Gries, S. T. (2013). *Statistics for Linguistics with R: A Practical Introduction*. De Gruyter.

Baayen, R. H. (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.

Companies Using R. Revolution Analytics. Kasutamise kuupäev 25.aprill 2016.a., allikas http://www.revolutionanalytics.com/companies-using-r

SAS Institute Inc.What Is the SAS System?: Introduction to the SAS System :: Step-by-Step Programming with Base SAS(R) Software. Kasutamise kuupäev 27.aprill 2016.a., allikas http://support.sas.com/documentation/cdl/en/basess/58133/HTML/default/viewer.htm#a002645411.htm

Raag,M.,Kolde,R. (2014/2015).Statistikatarkvara R õpetus · Statistiline andmeteadus ja visualiseerimine. Kasutamise kuupäev 21.aprill 2016.a., allikas http://andmeteadus.github.io/2015/rakendustarkvara_R/

IBM SPSS Statistics Features. (2016)IBM Corporation. Kasutamise kuupäev 27.aprill 2016.a., allikas http://www-01.ibm.com/software/analytics/spss/products/statistics/features.html

Baroni,M., Evert.S.(2015).SIGIL Homepage. Kasutamise küüpäev 29.aprill 2016.a., allikas http://www.stefan-evert.de/SIGIL/

Free Software Foundation, Inc. (2016).gnu.org. Kasutamise küüpäev 20.aprill 2016.a., allikas http://www.gnu.org/

Bakalaureusetöös kasutatud mõisted

Text Mining(tm)- Korpuste analüüsiks kasutatav R-programmeerimiskeele teek

GNU- Täielikult vabast tarkvarast koosnev Unix-i laadne operatsiooni süsteem. Kollektsioon paljudest programmidest, teekidest, mängudest. GNU arendus algas aastal 1984 ja on teatud kui GNU projekt.(Free Software Foundation, Inc., 2014)

UNIX- Operatsioonisüsteemide perekond.

Keelekorpus – Tavakõnes esinevate tekstide kogumik. Korpusi kasutatakse keele uurimise alusena statistilistes analüüsides.

Lisad

Lisa 1. Õppematerja

Tallinna Ülikool
Digitehnoloogiate Instituut

Lingvistika analüüs R-keele abil

Õppematerjal

Autor: Magnus Kvell

Tallinn 2016

Sisukord

Sissejuhatus

Käesolev õppematerjal annab lühiülevaate lingvistika analüüsimise võimalustest R-keeles. Õppematerjali eesmärgiks on õppur sisse juhatada R-keele kasutamisse ja selle keele loogikasse, keeleteaduslike analüüside tegemiseks. Õppematerjal on loodud inimestele, kellel on huvi viia läbi keeleteaduslike analüütilisi uuringuid, läbi programmeerimiskeele kasutamise. Materjal on koostatud R-keele versiooniga 3.2.4 ja RStudio versiooniga 0.99.896. Mõlemad programmid on vabavaralised ning tasuta kodulehtedelt kättesaadavad.

Enne materjaliga tutvumist oleks õppuril soovitatav tutvuda R-keele põhifunktsionaalsustega, et materjalist paremini aru saada. Materjal ei ole mõeldud täielikuks sissejuhatuseks R-programmeerimiskeelde, vaid ainult selle keele teatud funktsionaalsustesse, mis on seotud lingvistikaga. Õppematerjal on loodud inimestele, kellel on huvi keeleteaduse ning teksti statistilise analüüsimise vastu. Õppematerjal koosneb kuuest sissejuhatavast harjutusest, mida tutvustatakse lähemalt igas peatükis, nendeks on:

- Tekstifunktsioonid
- *Text mining* lisa teek
- Graafiline lingvistika andmete analüüs
- Tõenäosuse jaotus
- Sõnavara rikkuse mudel’
- Morfoloogiline analüüs

Harjutuste järjekord on pandud paika harjutuste raskusastmete järgi, sellest tulenevalt on soovituslik alustada algusest õppematerjali läbimisega, sest hilisemates peatükkides võivad operatsioonid arusaadamatuks jääda. Algajal õppuril võib mõne harjutuse jaoks vaja minna täiendavat lugemist, mis on ära mainitud õppematerjali lõpus. Iga harjutuse lõpus on ülesanded, mida õppur peab täitma, et materjalist paremini aru saada. Funktsioonid, mis ei ole täielikult lahti seletatud või mille puhul jäävad sisendid arusaamatuks, tuleb õppuril endal dokumentatsioonist järgi kontrollida.
Õppematerjaliga töötamise jaoks on materjaliga kaasas autori valitud tekstide kogumikud ja ka läbitavate harjuste kommenteeritud programmeerimiskood. Tekstid on valitud lähtuvalt

ülesannete vajadustest ning harjutused on valitud teiste materjalide analüüside tulemuse läbi ja autori isiklikust huvist lähtudes.

Autori kogemuste põhjal võib öelda, et keeleteadusliku materjali analüüsimine sellisel kujul võib algaja jaoks väga keeruline olla. Sellepärast soovitab autor enne materjaliga alustamist tutvuda erinevate lingvistika teoreetilisemate uuringutega ja reeglitega, et õppematerjalist aru saamine oleks lihtsam.

Tööriistad ja töö alustamine

Selles materjalis on kasutatud R-programmeerimiskeelega töötamiseks kahte vahendit, üks nendest on R-keel ja teine on RStudio nimeline lisakeskkond, mis on R-keelega töötamise mugavdamiseks. RStudio kasutamine ei ole kohustuslik, aga keskkond pakub võimalusi jälgida mugavalt oma jooksutatud koodi ajalugu. RStudio samuti annab funktsiooni ette, kui seda kirjutama hakata ning näitab ära koheselt funktsiooni sisendid.

Vajaminevad programmid leiduvad aadressidel:

- R-keel-https://www.r-project.org/
- Rstudio- https://www.rstudio.com/products/rstudio/

Töö alustamise puhul on mõistlik seada üles koht kuhu ja kust R-i failid liiguvad, see toimub käsu abil `setwd()`. Funktsiooni sisendiks on enda valitud töökoha asukoht. Käsk tuleb uuesti sisestada pärast keskkonna sulgemist.

```
setwd("C:/Users/Magnus/Desktop/Raamat")
```

Asukoha, kuhu R faile salvestab ja kust neid võtab saab järgmise käsu abil:

```
getwd()
```

Õppematerjali harjutustes oleva koodi ja tekstid leiab aadressilt:
http://www.tlu.ee/~manz/õppematerjal/õppematerjal.html

1 Tekstifunktsioonid

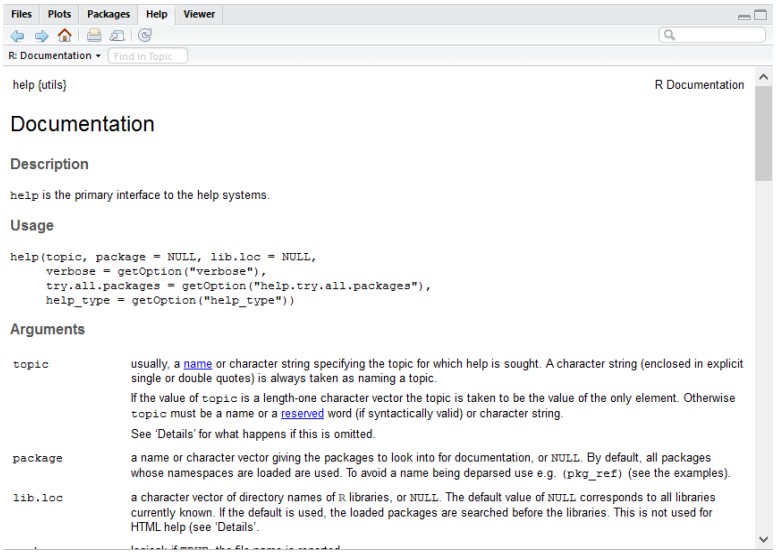
Selles õppematerjali osas antakse ülevaade tekstitöötlusfunktsioonidest. Funktsioonide läbi tegemine on vajalik edaspidistes peatükkides olevate harjutuste paremaks mõistmiseks. Autor soovitab tutvuda ja katsetada läbi kõik selles peatükis olevad funktsioonid. Teistes sarnastes materjalides ei ole pandud rõhku automaatsele tekstitöötlusele, millega keel R iseenesest saab hästi hakkama. Selles õppematerjalis on toodud välja sissejuhatuseks põhifunktsioonid, et kohaneda R-i keskkonna kasutamisega ja sealt edasi minna suuremas kogustes lingvistiliste andmete analüüsimisele.

1.1 Tekstitöötlusfunktsioonid

Dokumentatsioonist saab tekstitöötluse jaoks, R-i keskkonna siseselt funktsioonid kätte käsu `help()` abil:

```
help.search(keyword = "character", package = "base")
```

Kui sisesta ainult `help()`, siis konsooli ilmuvad konsooli kõrvale teised dokumentatsiooni käsud, kust saab edasi otsida ülesande lahendamise jaoks vajaminevaid funktsioone. See käsk üksi viib keskkonna kasutaja kogu R-i sisese dokumentatsiooni juurde. RStudiot kasutades ilmub dokumentatsioon all paremal nurgas.



Joonis 1 Dokumentatsioon RStudios

1.2 Teksti ja failide sisselugemine keskkonda

Tavalise teksti sisestus toimub muutuja andmise ja seejärel stringi kirjutamise abil. Teksti välja lugemiseks kasutatakse muutujat, mis anti stringile.

```
tekst<-"Täna on ilus ilm"
tekst
 [1] "Täna on ilus ilm"
```

Failide sisestamine ja failidesse kirjutamine

Tekstfailide *.txt* ja ka teiste failide teksti stringidena sisselugemiseks saab kasutada kahte eri meetodit, üks nendest on `readLines()` ja teine `scan()`.

```
lugemine <- readLines("ilm.txt")
```

Sõnade eraldamine iga stringina, `character(0)` funktsioon loob sõna vektori, andes sisendi 0, anname me programmile teada, et igast eraldiolevast tähest luuakse vektor. Nendeks harjutusteks võib kasutada autori loodud faili „ilm.txt“

```
scan("ilm.txt", character(0))
[1] "Täna"    "on"      "väljas"  "ilus"    "ilm"
```

Lausete eraldamine iga stringina, funktsiooni argument `sep` tähendab, et vektor eraldatakse pärast igat punkti:

```
scan("ilm.txt", character(0), sep = ".")
[1] "Täna on väljas ilus ilmõüöä"
```

Failidesse kirjutamiseks on ka kaks meetodit üks nendest on `scan()` ja teine `writeLines.`

```
cat(lugemine,file="loe.txt",sep=" ")
writeLines(lugemine, con = "loe.txt", sep = " ", useBytes = FALSE)
```

Järgnev kood loeb R-i keskkonda sisse terve Tallinna Ülikooli veebilehe, sarnase koodi abil võib näiteks kätte saada andmeid veebikeskkondadest.

```
veebileht <- scan("http://www.tlu.ee", character(0))
veebileht[1: 6]
```

```
[1] "<!DOCTYPE"          "html>"              "<head"
[4] "lang=\"et\">"       "<meta"              "charset=\"utf-8\">"
```

1.3 Mitme stringi liitmine ja lahutamine

Stringide liitmiseks ilma lisapakettide installimiseta on kaks võimalust, kas `paste()` või `cat()`. Stringide eraldamiseks on käsk `strsplit()`.

```
paste("tere","olen","mina",sep=".")
[1] "tere.olen.mina"

paste("tere","olen","mina",sep='')
[1] "tereolenmina"

cat(c("Tere","olen","mina"), sep = "-")
Tere-olen-mina
unlist(strsplit("Tere olen mina", " "))
[1] "Tere" "olen" "mina"
```

Tähtede arvu leidmine sõnades

Stringi pikkuse annab `nchar()`.

```
nchar("Tere olen mina")
[1] 14
```

Stringi eraldamine

Käsk `substr()` saab hakkama mingi osa stringi eraldamisega, esimeseks argumendiks on string, teiseks eraldatav stringi osa ja kolmandaks stringide arv. Eraldamine toimub automaatselt tühiku järgi.

```
substr("õues on ilus ilm", 1, 4)
[1] "õues"
```

Tähemärgi või sõna leidmine stringis

`Grep()` funktsioon võtab esimeseks argumendiks tavalise argumendi ja teiseks argumendiks sisestatava vektori, kui kirjutada `value` võrduseks `FALSE`, siis tagastab `grep` uue vektori, milles on sisestatava vektori kohtade arv, sisestatud argumendile vastavate vastete järgi. Kui value on `TRUE`, siis kirjutab `grep` välja need vektorid, mis ühtivad argumendiga.

```
tekst<-"Täna on ilus ilm. See kass on armas"
sub(pattern = "on", replacement = "oli",  x = tekst)
grep("r+",   c("Tere",   "tore   tari",   "kollane",   "rrr"),   perl=TRUE,
value=FALSE)
[1] 1 2 4
grep("r+", c("Tere", "tore tari", "kollane", "rrr"), perl=TRUE, value=TRUE)
[1] "Tere"      "tore tari" "rrr"
```

Stringis mustri asendamine

Funktsioonide `sub()` ja `gsub()` abil on võimalik asendada stringis olevaid tähemärke uute vastu. Käsk `sub()` vahetab välja ühe sisendi teise antud sisendi vastu ja `gsub()` asendab kõik terves sisendis olevad sarnased mustrid.

Käsk `chartr()` on lühend sõnadest *character translation*, selle funktsiooni abil saab asendada stringis vastavad sisendiks antud tähemärgid.

```
 [1] "Täna oli ilus ilm. See kass on armas"
gsub(pattern = "on", replacement = "oli",  x = tekst)
[1] "Täna oli ilus ilm. See kass oli armas"
chartr(old="i",new="u",x="täna on ilus ilm")
[1] "täna on ulus ulm"
```

Tähe suuruse muutmine stringis

Tähemärgi suuruse muutmiseks on kaks käsku, üks neist `toupper()`, mis on väikse tähe suureks muutmiseks ja teine `tolower()`, mis on mõeldud suure tähe väikseks tegemiseks. Need käsud üksikuna võtavad terve stringis olevad tähed ja töötlevad need.

```
tolower("Tere TÄNA on Ilus Ilm")
[1] "tere täna on ilus ilm"
```

```
toupper("Tere TÄNA on Ilus Ilm")
[1] "TERE TÄNA ON ILUS ILM"
```

Stringide võrdlemine

Kahe stringi samaväärsuse saamiseks saab kasutada kahte võrdusmärki, vastused tagastatakse kas `TRUE` või `FALSE` argumendi näol.

```
"TERE"=="TERE"
[1] TRUE
"tere"=="TERE"
[1] FALSE
```

Käsk `agrep()` võrdleb stringe Levenshteini distantsi järgi. Levenshteini distants mõõdab kahe stringi seotust. Väljundi FALSE korral tagastatakse stringi asukoht vektoris.

```
agrep(pattern = "tere", x = c("tervist", "12334", "terve", "TERVIST"), max
= 3, value = TRUE)
[1] "tervist" "terve"
agrep(pattern = "tere", x = c("tervist", "12334", "terve", "TERVIST"), max
= 3, value = FALSE)
[1] 1 3
```

1.4 Teksti puhastamine

Enne mis iganes teksti või tabeliga töötama hakkamist, ei pruugi andmed alati olla R-i jaoks võrreldaval kujul, näiteks võib esineda tekstis jutumärke, komi, numbreid või mis iganes sümboleid, mis rikuvad sisendi olemust. Valesti sisse loetud ja töötlemata tekst võib rikkuda analüüsi tulemused. Selleks, et tulemused oleks võrreldavad tuleks tekst puhastada.

```
essee <- readLines("essee1.txt")
essee
```

Sellisel kujul teksti sisselugemine ja välja kirjutamine, nagu näha, ei anna teksti kujul, mis oleks analüüsitav. Näiteks kui oleks soov sõnade kasutatavust võrrelda, siis tuleks tekst sisse lugeda antud kujul:

```
essee <- scan("essee1.txt", character(0), quote = NULL)
```

R-i sisend `[[:punct:]]`, sisaldab enim kasutatavaid tähemärke. Tähemärkide eemaldamine sisse loetavast tekstist toimub järgneva koodi abil:

```
essee<- gsub("[[:punct:]]", " ", loe)
```

Lõplik kood, mis jagab kõik sõnad stringideks ja eemaldab tähemärgid ja väljatrükk:

```
loe <- scan("essee1.txt", character(0), quote = NULL)
essee<- gsub("[[:punct:]]", " ", loe)
essee
```

unikaalsete sõnade leidmine

Funktsioon `unique()`, leiab kõik kordumatud stringid tekstis ja väljastab need konsooli.

```
unique(essee)
```

1.5 Esimese tähe muutmine suureks täheks, igas stringis.

Selleks, et üks täht sisendis muuta, eraldi käsk puudub, aga tähe muutmise jaoks stringis saab teha käskude jada, millega probleem lahendada. Algoritmid, mida võib tihedamini vaja minna on soovitatav funktsiooniks teha, sest selle läbi saab sama algoritmi uuesti välja kutsuda samas projektis, mingis teises koodi osas, funktsiooni näol. `Strsplit()` kasutades saab ära eraldada stringist valitud sümboli, teise stringina. Käsu `paste()` abil saab uuesti sõnad liita. Pärast seda saab valminud funktsiooni teksti külge kinnitada näiteks `sapply()` abil.

```
esiTäht <- function(x) {
  s <- strsplit(x, " ")[[1]]
  paste(toupper(substring(s, 1,1)), substring(s, 2),
        sep="", collapse=" ")
}
suur<-sapply(essee, esiTäht)
suur
```

Ülesanded:

Loe sisse mõni teine tekst näiteks lehelt http://evkk.tlu.ee/, stringi haaval.

Eemalda tekstist kõik numbrid ja tähemärgid.

Muuda tekstis kolmas olev sümbol suureks täheks.

Kirjuta tekst .csv faili ja ava see konsoolis.

2 Text mining(tm) teek

Text mining(tm) teek annab võimaluse lugeda sisse suurel hulgal andmeid korraga ja neis teha kergemaid analüüse paari käsuga. Selles peatükis käsitleme sissejuhatavalt antud lisapaketi võimalusi. Teek tegeleb üldjuhul korpustega, näiteks võib sisse lugeda teeki erinevate suhtlusvõrkude kommentaarid ja neid uurida. Autor leiab, et see lisapakett on hea vahend suurema teksti või mitme teksti failide sisselugemiseks ja nende analüüsimiseks.

Esiteks läheb meil vaja harjutuse jaoks teatud hulgal andmeid, mida analüüsida, näidisfailid andmetega leiab autori kaasatud materjalist kaustast korpus. Andmed näidisülesannetes on saadud Eesti vahekeelekorpuse kodulehelt.

2.1 Vajaminevate lisade allalaadimine

Lisapakettide allalaadimine toimub `Install()` käsu abil, mitme paketi korraga allalaadimiseks võib teha muutuja ja selle `Install.packages()` käsku lisada.

```
Teegid <- c("tm", "wordcloud")
install.packages(Teegid, dependencies=TRUE)
install.packages("Rcampdf", repos = "http://datacube.wu.ac.at/", type = "source")
```

Edaspidi on näha, mille jaoks mingit teeki kasutame. Kõik lisad tuleb ainult korra installeerida R-i. Pärast saab neid välja kutsuda library() käsuga.

2.2 Korpuse sisselugemine

Korpuse sisselugemine toimub samamoodi nagu tavalise faili puhul, aga pärast sisselugemist, saame kasutada Text Mining teeki ja kasutada käsku `Corpus()`, et teha failidest ühine loetelu. Käsk `summary()` näitab andmeid sisse loetud korpuse kohta.

```
korpus <- file.path("korpus")
library(tm)
tekstid <- Corpus(DirSource(korpus))
summary(tekstid)
inspect(tekstid[1:5])
```

Teksti puhastamine toimub sama loogika alusel nagu eelmises peatükis, aga Text Mining annab puhastamise jaoks lisafunktsionaalsusi nagu näiteks `removePuntuation`, mis eemaldab kirjavahemärgid. Samamoodi leiduvad teegis funktsioonid nagu näiteks `removeWords()` ja `removeNumbers()`, mida saab kasutada jälgides samat loogikat.

```
tekstid <- tm_map(tekstid, removePunctuation)
```

Sõnasagedus maatriks annab ülevaate sageduste kohta igas failis. See on hea funktsionaalsus sageduste analüüsimiseks ja sellega saab kergelt analüüsida kõnekasutust, erinevates sarnastes tekstides, näiteks samateemalised esseed. Ühe variandina oleks näiteks võimalik kommentaarida puhul saada funktsiooni kaudu kätte negatiivsed sõnad ning neid võrrelda. Võrdluse tulemus annaks järeldusi, millised kommentaarid olid negatiivsed ja palju neid oli. Funktsiooni kasutamiseks tuleb `DocumentTermMatrix()` käsule anda sisse vastav tekst või korpus (Basic Text Mining in R., kuupäev puudub)..

```
dmatrix <- DocumentTermMatrix(tekstid)
dmatrix
#vastupidine
tmatrix <- TermDocumentMatrix(tekstid)
tmatrix
```

Sageduste järjekorda seadmine sõnade arvu populaarsuse järgi kahanevalt:

```
#erinevate sõnade sagedus
sagedus <- colSums(as.matrix(dmatrix))
length(sagedus)
järjekord <- order(sagedus)
   000    100   1869   1985   1994   2001    822  aasta aastad aastal
     4      1      2      1      1      1      1      1      1      9
```

2.3 Sõnapilve tegemine

Sõnapilve tegemiseks saab kasutada Wordcloud teeki, mis sai varem R-i lisatud. Selleks tuleb see lisa uuesti enne funktsiooni `library()` abil välja kutsuda. Pärast seda saame funktsiooni `wordcloud()` abil välja kutsuda graafilise sõnapilve. Sõna pilve analüüsides näeme, et sõna „eesti“ ja „see“ on sageduselt enim esinevad (Basic Text Mining in R., kuupäev puudub).

```
library(wordcloud)
```

```
wordcloud(names(sagedus), sagedus, min.freq=5)
```

Joonis 2 Sõnapilv

Ülesanded:

Koosta värviline sõnapilv. Kasuta selleks järgmist funktsiooni:

```
colors=brewer.pal(6, "Dark2")
```

3 Graafiline lingvistiline andmete analüüs

Enne järgnevate peatükkidega alustamist tuleb meil R-keele keskkonda lisada uus teek nimega languageR. Selle saame lisada samamoodi nagu eelmise harjutuse puhul ehk siis `install.packages()` käsu abil.

```
install.packages("languageR")
```

Tänu graafilisele analüüsile saab jälgida erinevaid andmeid visuaalselt. Selleks harjutuseks saame sisse lugeda autori kaustast fail tõde.txt, mis koosneb A.H.Tammsaare ilukirjandusliku teose tekstist „Tõde ja Õigus“. Teose, koos teiste eestikeelsete litsentsi vabade teostega võib leida aadressil: http://www.luts.ee/index.php/e-raamatud PDF failina. Faili tekst failiks tegemiseks võib kasutada näiteks R-keele mõnda teeki või internetis leitavaid failide konvertereid. (E-raamatud - Tartu Linnaraamatukogu, 2016)

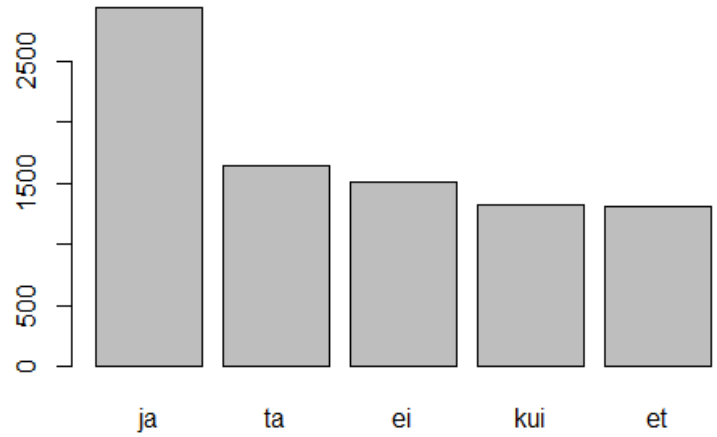
Faili sisselugemise järgselt koostame tabeli, kus saame sõnade sagedused `.table()` käsu abil. Pärast sisselugemist tuleb eemaldada tähemärgid.

```
tõde <- scan("tõde.txt" ,character(0), encoding = "UTF-8")
tõde<- gsub("[[:punct:]]", " ", tõde)
head(tõde[1:5])
[1] ""       "I"      "See"    "oli"    "läinud"
```

3.1 Sageduste leidmine

Käsk `table()` annab meile automaatselt sõnade sagedused. Kui `decreasing` väljundiks sisestada `TRUE`, siis saame sõnasagedused kahanevalt. Väljastame esimese viie sõna sõnasagedused ja teeme neist tulpdiagrammi.

```
tõde.table <- table(tõde)
tõde.table <- sort(tõde.table, decreasing = TRUE)
barplot(tõde.table[1:5])
[1] " "       "I"      "See"    "oli"    "läinud"
```

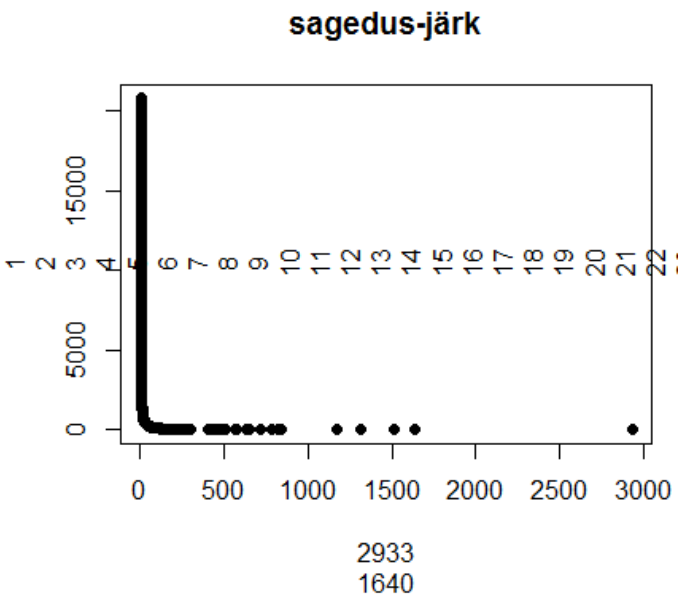


Tabelist on näha, et domineerivamad sõnad teoses on sidesõnad. Kõige populaarsemat sõna „ja“ on kasutatud teoses pea kaks korda enam, kui populaarsuselt teist sõna. Numbriliseks väljatrükiks võib kasutada lihtsalt väljatrükki `table.tõde [1:5]`.

3.2 Punktdiagramm sõnasageduste järgu järgi

Sõnasageduste järgu järkude järgi saab järeldusi teha, palju erinevaid sõnu on tekstis kasutatud. Selleks, et sageduste järgud saada, asendama `length()` funktsiooni abil sageduste asukohad. Sellega saame jälgida, kuidas on sõnad tekstis jaotatud. Diagrammi x osal on nähtavad sõnasagedused ja y osal sageduste järk.

```
järk <- 1:length(tõde.table)
järk[1:5]
plot(tõde.table, ranks, main="sagedus-järk",
     xlab=tõde.table, ylab=ranks, pch=19)
```



Joonis 3 Sagedusjärk

Tabelist saame näha, et populaarsuselt 5 kõige sagedasemalt esinenud sõna, esinevad tekstis palju sagedamini kui ülejäänud sõnad.

Ülesanded:

Otsi `help()` käsku kasutades üles erinevate diagrammide tegemiste võimalused, katseta need sagedustega läbi.

4 Tõenäosuse jaotus

Kui jälgida, kui palju mingit sõna kasutatakse või vokaalide pikkusi sõnades, siis viime me läbi statistilise analüüsi. Iga statistilise analüüsi tulemus on erinev. Näiteks sõnasagedused erinevad iga uue korpuse analüüsi puhul, samamoodi varieeruvad vokaalide pikkused erinevate silpide ja sõnade puhul. Iga suvalise muutuja puhul, mõned analüüsi tulemused võivad olla tõenäolisemad, kui teised. Siin peatükis saame tuua välja kaks tähtsat suvaliste muutujate kategooriat. Suvalisi muutujaid nagu sageduste arve nimetatakse diskreetseteks. Muutujaid, mis on kestvad kutsutakse pidevateks. Selles õppematerjalis toome näite diskreetsete muutujate jaotuse kohta.

4.1 Diskreetse jaotuse näide

Sõna esinemise tõenäosustega tegelemiseks defineerime kaks väärtust. Üks nendest on p-väärtus ehk õnnestumise tõenäosus, teine on väärtus q, ebaõnnestumis väärtus, mis on vastupidine. Oletame, et meil on suur korpus, kus on mainitud n = 1000000 korda sõna „ja“ ning selle sõna esinemise tõenäosus on näiteks 0.056428435, siin juhul saame võtta p-väärtuseks sõnaesinemise sageduse. Need arvud omavahel läbi korrutades saame vastuseks 56429. Nüüd oletame, et on lõpmatu arv korpuseid ja igas korpuses esineb sõna „ja“ keskmiselt 57000 korda, miljoni sõna puhul. Tänu nendele oletustele saame leida sõna esinemise keskmise tõenäosuse erinevuse tekstis, binoomjaotuse abil. Binoomjaotuse jaoks pakub keel R meile kolme erinevat funktsiooni `dbinom(), pbinom() ja rbinom()`. Selles harjutuses vaatleme neist kahte. Relatiivse tõenäosuse leidmise jaoks saame kasutada funktsiooni `dbinom()`. Funktsiooni sisendiks on kolm väärtust. Esimene neist on keskmine sõnade arv, teine on n ehk originaalne sõnade arv ja kolmas on esinemise sagedus (Baayen, R. H., 2008) .

```
bino <- dbinom(56000, 1000000, 0.056428435)
sprintf("%.10f",bino)
[1] 0.0003082753
```

Funktsioon `pbinom()` leiab `dbinomi()` summa. Ütleme, et meil on ühe sõna esinemise tõenäosus väga väike, näiteks 0,0000080, siit leiame, kui suur on tõenäosus, et see sõna ei esine miljoni sõnalises tekstis mitte kordagi ja peale seda esineb sõna korra.

Kordagi:

```
dbinom(0, size = 1000000, prob = 0.0000080)
[1] 0.0003354519
```

Korra:

```
dbinom(0, size = 1000000, prob = 0.0000080)+
dbinom(1, size = 1000000, prob = 0.0000080)
[1] 0.003019089
```

Eelneva liitmise asemel saaksime me kasutada funktsiooni `pbinom()` antud kujul:

```
pbinom(1, 1000000, 0.0000080)
[1] 0.003019089
```

`Rbinom()` genereerib suvalisi numbreid, see funktsioon aitab simuleerida olukordi, näiteks mingi sõna esinemise tõenäosuse leidmiseks.

4.2 Mingi sõnakorduste leidmine teksti osas ja tõenäosus

Selles harjutuses leiame, kui mitu korda esineb sõna „Andres“ teksti erinevates osades. Tänu sellele saame jälgida, kus raamatu osades ilmub tegelane enim. Selleks saame kasutada `data.frame()` käsku, millega saame sõnade loendi tabeli kujul ja `cut()` käsu abil võime lõigata teksti näiteks neljakümneks võrdseks osaks. Kuna meil on 876267 sõna, siis neid ei saa võrdseteks osadeks jaotada. Sellepärast tuleb lõpust ära jätta 27 sõna, et me saaksime jaotada teksti võrdseks osaks (Baayen, R. H., 2008).

```
tõdeSag <- data.frame(sõna = tõde[1:87240],
                      tükk = cut(1:87240, breaks = 40, labels = F))
tõdeSag[1:5, ]
      sõna tükk
1 <U+FEFF>    1
2        I    1
3      See    1
4      oli    1
5   läinud    1
```

Pärast seda saame loendada sõnad, mis vastavad ette antud otsitavale sõnale. Käsk `tapply()` leiab üles kordumatud sõnad, mille taha tekib kas `TRUE` või `FALSE` väljund ja

käsu `sum()` abil, loendatakse need kõik kokku igas loodud sektsioonis. Väljatrükiks saame palju on mainitud otsitud tegelast igas sektsioonis.

```
tõdeSag$tõde <- tõdeSag$sõna == "Andres"
loendaAndres <- tapply(tõdeSag$tõde, tõdeSag$tükk, sum)
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 1 20  7  4 11  6  5  3  8 10  6 17 14  5  6 12 20 17 23  3  2  5  0  5
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
21 14 21 12 15  4  5  4 13  9  0  0 19 30 11  8
```

Järgmisena teeme `xtabs()` käsu abil sagedustabeli, et analüüsida, mitu korda sõna „Andres“ esineb igas ärajaotatud osas.

```
loendaAndres.tab <- xtabs(~loendaAndres)
loendaAndres.tab
0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 17 19 20 21 23 30
 3  1  1  2  3  5  3  1  2  1  1  2  2  1  2  1  2  1  2  2  1  1
```

Väljund tabelist on näha, et kolmes teksti lõigus ei esine sõna „Andres“ kordagi ja näiteks kahes lõigus esineb sõna 21 korda.

Nüüd võime leida näiteks tõenäosuste sageduse, kui tõenäoline on, et see sõna esineb nii mitu korda, mingis teksti lõigus. Sageduste summa on võrdne ühega.

```
loendaAndres.probs= xtabs(~loendaAndres)/nrow(loendaAndres)
round(loendaAndres.probs)
loendaAndres.probs
    0     1     2     3     4     5     6     7     8     9    10    11
12    13
0.075 0.025 0.025 0.050 0.075 0.125 0.075 0.025 0.050 0.025 0.025 0.050
0.050 0.025
   14    15    17    19    20    21    23    30
0.050 0.025 0.050 0.025 0.050 0.050 0.025 0.025
```

Veel võime arvutuste teel saada, kui tihedalt see sõna esineb tekstis.

```
n <- 87240
p <- mean(loendaAndres/n)
p
[1] 0.0001134801
```

Tulemusest on näha, et selle sõna esinemise tõenäosus on väga väike, arvestades koguteksti suurust.

Ülesanded:

Proovi läbi binoomjaotused teiste arvudega, analüüsi tulemusi.

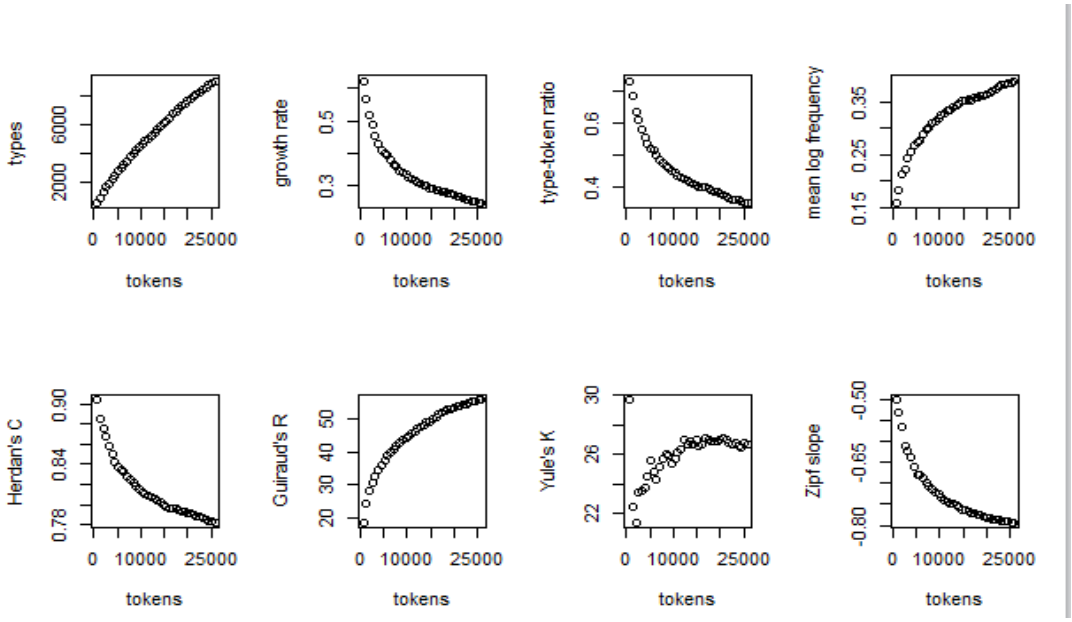
5 Leksikaalse rikkuse mudel

Sageduse mõõted nagu sõnade sageduste analüüsimine võib pakkuda lingvistidele suuri väljakutseid. Sõnavara rikkust, saab mõõta selle järgi, kui palju on tekstis kasutatud erinevaid sõnu ja ka kokku lugedes nende sagedusi. R-i keskkonnas on sõnavara rikkuste analüüsimeks mitmeid funktsioone.

Funktsiooni `growth.fnc()` abil saame välja lugeda tekstist tõde erinevaid andmeid sageduste ja teksti rikkuse kohta. Selle funktsiooni sisendiks anname tekstilõigud ja igas lõigus olevate sõnade arvud. Jagades 87240 ehk siis sõnade arvu neljakümneks erinevaks osaks, sellest järeldame, et igas osas peaks olema ligikaudu 2181 sõna, et raamat terviklikult sisse lugeda. Funktsiooni välja trükkides on näha hulk erinevaid andmeid.

Kolm esimest veergu annavad vastava arvu sõnu kumulatiivselt kasvavas järjekorras, igast tekstilõigust. Järgmised kolm veergu annavad sõnad, mida on loendatud ühel, kahel ja kolmel korral. Ülejäänud lahtrid annavad informatsiooni leksikaalse rikkuse kohta. Viimastes veergudes saadavate tulemuste kohta võib lisakirjandust uurida, sest selles õppematerjalis, nende tähendustel me pikemalt ei peatu. Tabeli tegemine funktsioonist käib `plot()` käsu abil sisendina kasutame esimest muutujat.

```
tõde.kasv= growth.fnc(text = tõde, size = 2181, nchunks = 40)
head.growth(tõde.kasv)
plot(tõde.kasv)
```



Joonis 4 Leksikaalse rikkuse mõõtmed

5.1 Sõnavara rikkuse võrdlemine

Lähemateks analüüsideks jaotame teksti kaheks ebavõrdseks osaks. Kahe teksti sõnade rikkust saab võrrelda funktsiooni `compare.richness.fnc()` abil. See käsk võrdleb sõnade sagedus mudeleid igas tekstis ja valib parima mudeli teksti analüüsimiseks. Saadud tulemuses näitab Z-veerg, teksti rikkuse kasvu. Selles ülesandes on esimene tekst pikem ja teine lühem, esimese Z-arvu positiivse, sest esimeses tekstis on suurem kasutatud sõnade arv ja teise negatiivse, sest erinevate sõnade tihedus ja arv pikemas tekstis tk1 on suurem. Esimene number on suur, sest see on seotud p-väärtusega, mis on nullilähedane.

```
tk1 = tõde[1:50000]
tk2 = tõde[50001:87240]
compare.richness.fnc(tk1, tk2)
comparison of lexical richness for tk1 and tk2
with approximations of variances based on the LNRE models
gigp (X2 = 64.39) and gigp (X2 = 39.87)
    Tokens Types HapaxLegomena GrowthRate
tk1  50000 14486          9992    0.19984
tk2  37240 11019          7595    0.20395
two-tailed tests:
                             Z      p
Vocabulary Size         28.291 0.0000
Vocabulary Growth Rate -1.469 0.1418
```

Kui peale eelmist võrdlust võrrelda sama teksti, aga kahe võrdse osana, siis saame tulemusteks palju väiksemad Z-i väärtused. Käsu `length()` abil saame teha mõlemad tükid sama pikaks, luues uue muutuja. Järgmisena võrdleme uuesti mõlemat teksti tükki.

```
tk1a = tk1[1:length(tk2)]
compare.richness.fnc(tk1a, tk2)
comparison of lexical richness for tk1a and tk2
with approximations of variances based on the LNRE models
gigp (X2 = 27.74) and gigp (X2 = 39.87)
     Tokens Types HapaxLegomena GrowthRate
tk1a  37240 11649          8097    0.21743
tk2   37240 11019          7595    0.20395
two-tailed tests:
                            Z p
Vocabulary Size        5.4407 0
Vocabulary Growth Rate 4.4082 0
```

Saadud tulemus on mõnevõrra üllatav, sest tabelit vaadates on näha, et teises tekstilõigus toimub sõnavara kasutuse kasv. Ebaregulaarne on see sellepärast, et teksti rikkuse kasv on selgesti märgatav, isegi pärast võrdsete osade võrdlemist. Enamus sellel kujul teksti võrdlustes peaks olema Z väärtused nullilähedased.

5.2 Sõnavara rikkuse seaduspärasused

Suuremate tekstide normaalsuse uurimiseks saame kasutada erinevaid seaduspärasusi. Üks nendest on Zipfi seadus, mis väidab, et kõige sagedamini esinev sõna esineb sageduselt kaks korda enam, kui sageduselt teine sõna, sama kehtib ka järgnevate sõnasagedustega kohta. Teine seaduspärasus, mida saame sageduste ja teksti rikkuse jaoks kasutada on Heapi seadus, tuntud ka nime all Herdani seadus. Heapi seadus kirjeldab erinevate sõnade ja teksti suuruse suhet (Lü, L., Zhang, Z.-K., & Zhou, T., 2013).

Heapi seadus saadakse järgmise valemi abil:

$V R (n) = Kn \beta$

Heapi seaduse valemis Vr(n) on erinevate väljatoodud sõnade sagedus. K ja β on vabad parameetrid, mis on empiiriliselt saadud, tavaliselt K jääb 10 ja 100 vahele ja β jääb 0.4 ja 0.6

vahele. Täht n, märgistab teksti suurust ehk siis sõnade arvu. Heapi seadus on asümptootiliselt võrdne Zipfi seadusega (Gell-Mann.M., 1994).

R-keeles saame neid analüüse graafiliselt läbi viia eelmises harjutuses saadud tabeli järgi. Selliste analüüside paremaks jälgimiseks kasutatakse lingvistikas logaritme. Logaritmide kasutamine annab paremini jälgitavad arvud ning väiksemad numbrid. Logaritmide kasutamisel saame seaduspärasusi statistiliselt täpsemalt mõõta.

5.3 Heapi seadus

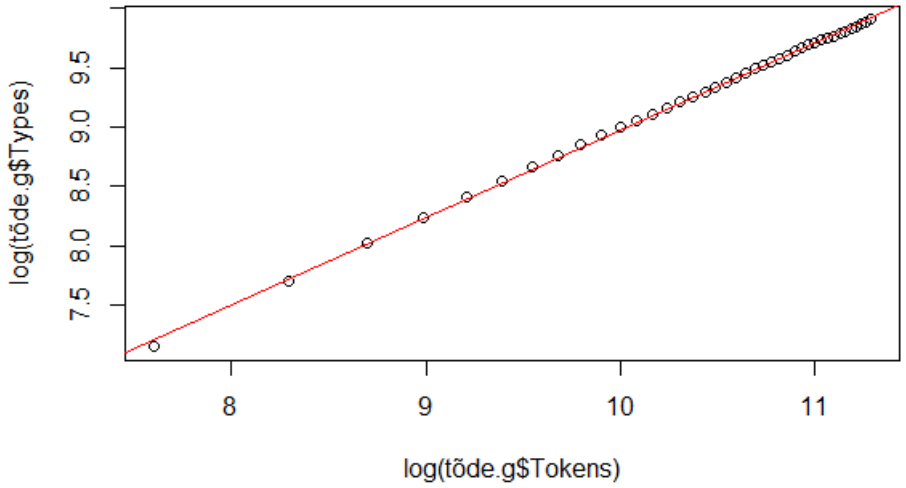
Heapi seaduse saame nii, et võtame leksikaalse rikkuse mudelist järjest andmeid välja. Selle jaoks teeme uue muutuja `tõde.g`, mille abil eraldame eelmisest muutujast andmed vastaval kujul.

```
tõde.g = tõde.kasv@data$data
head(tõde.g, 3)
```

Selles harjutuses saame jälgida sõnavara kasvu läbi Heapi seaduse. Esimene vastav väärtus läbi selle seaduse, peaks olema järgmisest ligikaudu kaks korda kaugemal, sellele järgneva sageduse vahe peaks olema jällegi kaks korda lühem. Analüüsi tulemuseks peaks seaduspärasuse järgi tekkima keskele enamvähem diagonaalne joon. Esiteks, võtame vajalikud väärtused.

Pärast seda saame teha logaritme kasutades graafiku, millele tõmbame keskele võrreldava joone. Graafiku tulemuseks peaks olema diagonaalselt läbiv joon, millel on kujutatud sõnade sagedused. Saadud uuest mudelist võrdleme tüüpi(Type) ja sõnade arvu(Tokens).

```
plot(log(tõde.g$Tokens), log(tõde.g$Types))
tõde.g.lm = lm(log(tõde.g$Types)~log(tõde.g$Tokens))
abline(tõde.g.lm, col="red")
```



Joonis 5 Heapi seadus

Graafikult on näha, et tekst on seaduspärane. Peale tabeli tegemist võime välja kutsuda saadud mudeli. `Summary()` käsu abil.

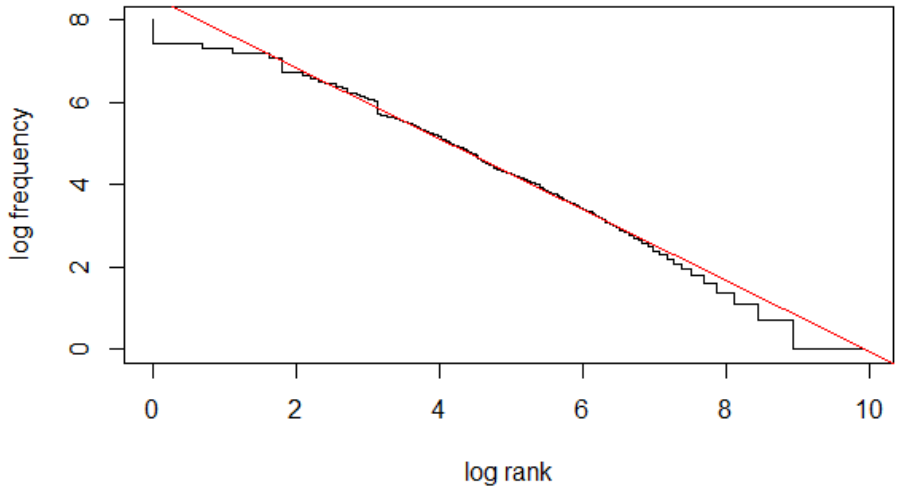
```
summary(tõde.g.lm)
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        1.623092   0.034340   47.27   <2e-16 ***
log(tõde.g$Tokens) 0.735243   0.003304  222.56   <2e-16 ***
```

5.4 Zipfi seaduspärasus

Zipfi seaduspärasuse ehk Zipfi kallaku eraldamiseks saame kasutada funktsiooni `zipf.fnc()`. Funktsiooni väljundiks on sõnade sagedused, nende sageduste sagedused ja sõnade sageduste vastav positsioon. Funktsiooni väljatrükkides on näha tabel ülalmainitud väärtustega.

```
z = zipf.fnc(tõde, plot = T)
head(z, n = 3)
    frequency freqOfFreq rank
180      2933          1    1
179      1640          1    2
178      1512          1    3
```

Järgmisena saame teha lineaarse mudeli ennustamaks funktsiooni tulemust, selleks teeme tabeli ja tõmbame keskele uuesti andmetest sõltuvuses oleva joone.



Joonis 6 Zipfi seadus

Ülesanded:

Vii läbi uuring läbi mõne väiksemas mahus oleva tekstiga, näiteks „essee1.txt“ autori kaasatud tekstitest, võrdle ja analüüsi tulemusi.

6 Morfoloogiline analüüs

Morfoloogia ehk vormiõpetus on lingvistika osa, mis uurib sõnavorme. Eesti keelseks morfoloogia analüüsiks on olemas erinevad vahendid, näiteks morfosüntaktilise analüüsi leiab lehelt http://evkk.tlu.ee/Search/search_reeglid.html. Antud õppematerjalis kasutame ühte teksti, millega on sama analüüs läbi viidud. Morfosüntakti näidet saab vaadata kas Eesti vahekeelekorpuse kodulehelt või autori materjalis olevast teksti failist „morfol.txt“.

Näite pealt nähtuna, annab analüüs hulga erinevaid sümboleid ja lühendeid, sealhulgas sõna algvormi, sõnaliigi jne. Selle harjutuse eesmärk on leida sõnaliigid läbiviidud analüüsist. Sõnaliikide tähistusi saab näha failist „sõnaliigid.txt“. ’

6.1 Sõnaliikide eraldamine

Esimese sammuna on vaja sisselugeda morfosüntaktiliselt analüüsitud fail. Kui morfoanalüüsi tulemusi vaadata, siis on näha, et kolmas, siis teises stringis, pärast analüüsitud sõna on sõnavormi tähis. Sellest lähtuvalt tuleb eraldada esimene ja teine string ning teisest stringist võtta kolmas tähis. Funktsioon `as.character()` võtab ühest reast iga valitud sümboli.

```
morfol1 <- readLines(paste("morfol.txt", sep=""))
morfol2 <- sapply(strsplit(morfol1[substr(morfol1, 1, 1)=="\t"], "  "),
function(v) as.character(v[3]))´
morfol2
```

6.2 Sõnaliikide sageduse leidmine

Selleks, et saadud sõnaliike ja sõnaliikide tähiseid, mis on teises failis võrrelda, tuleks viia mõlemad tabelid samale kujule. Selle jaoks saab kasutada käsku `as.data.frame(table)`. Saadud tabelis tuleb ära defineerida tabeli päise pealkirjad, et neid hiljem võrrelda sisse loetava sõnaliikide tähistuse failiga.

```
morfol3 <- as.data.frame(table(morfol2[nchar(morfol2)==1]))
names(morfol3)<-c("sõnaliik", "kogus")
```

6.3 Kahe tabeli liitmine

Enne lõpptulemuse saamist, kus on toodud välja sõnaliigi tähis ja sagedus, tuleb sisse lugeda keskkonda sõnaliikide tähistuste tabel „sõnaliigid.txt“. Pärast seda saab tabelid liita `merge()` funktsiooni abil. Kui tabelid on liidetud võib tulemused panna sageduste järjekorras käsu order järgi.

```
morfol4<-merge(morfol3, liigid, by.x="sõnaliik", by.y="lühend")
morfo5<- morfol4[rev(order(morfol4[2])), ]
morfo5
```

6.4 Graafiku tegemine

Graafiku tegemiseks on olenevalt graafikust erinevad käsud, selle ülesande juures teeme sektordiagrammi `pie()`, et näidata palju on mingeid sõnaliike tabelis. Diagrammi esimeseks sisendiks on arvuline väärtus ja teiseks sektori nimi.

Lõplik ülesande kood näeb välja selline:

```
morfol1 <- readLines(paste("morfol.txt", sep=""))
morfol2 <- sapply(strsplit(morfol1[substr(morfol1, 1, 1)=="\t"], "  "),
function(v) as.character(v[3]))
morfol3<-as.data.frame(table(morfol2[nchar(morfol2)==1]))
names(morfol3) <- c("sõnaliik", "kogus")
liigid <- read.table("sõnaliigid.txt", header=TRUE);
morfol4 <- merge(morfol3, liigid, by.x="sõnaliik", by.y="lühend")
morfo5 <- morfol4[rev(order(morfol4[2])), ]
morfo5
pie(morfo5$kogus, morfo5$kirjeldus)
```

Ülesanded:

Vii analüüs läbi „tõde.txt“ failiga ja tee sellest sõnapilv(`wordCloud()`).

Materjalid edasiseks õppimiseks:

Gries, S. T. (2013). *Statistics for Linguistics with R: A Practical Introduction*. De Gruyter.

Baayen, R. H. (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.

Jockers, M. (2014). *Text Analysis with R for Students of Literature*. Springer.

Kasutatud kirjandus

Müller.F.,Waibe.B. (2016). Corpus linguistics - an introduction — Englisches Seminar. Kasutamise kuupäev 11.aprill 2016.a., allikas http://www.anglistik.uni-freiburg.de/seminar/abteilungen/sprachwissenschaft/ls_mair/corpus-linguistics

R Programming/Text Processing - Wikibooks, open books for an open world. Kasutamise kuupäev 16.aprill 2016.a., allikas https://en.wikibooks.org/wiki/R_Programming/Text_Processing

E-raamatud - Tartu Linnaraamatukogu. (2016). Kasutamise kuupäev 17.aprill 2016.a., allikas http://www.luts.ee/index.php/e-raamatud

Gries, S. T. (2013). *Statistics for Linguistics with R: A Practical Introduction*. De Gruyter.

Baayen, R. H. (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.

Õõpik.(2012).Statistika sõnaraamat - Ökoloogia: skaalad, uurimismeetodid ja tulemuste teaduslik esinduslikkus. Kasutamise kuupäev 19.aprill 2016.a., allikas http://skaaladjameetodid.weebly.com/statistika-sotildenaraamat.html

Korpus: Otsing. Kasutamise kuupäev 15.aprill 2016.a., allikas http://evkk.tlu.ee/Search

Korpus: Otsingu tulemused. Kasutamise kuupäev 18.aprill 2016.a., allikas http://evkk.tlu.ee/Search/search_reeglid.html

Text Mining Package [R package tm version 0.6-2]. Kasutamise kuupäev 18.aprill 2016.a., allikas https://cran.r-project.org/web/packages/tm/index.html

Basic Text Mining in R. Kasutamise kuupäev 14.aprill 2016.a., allikas, 2016, from https://rstudio-pubs-static.s3.amazonaws.com/31867_8236987cf0a8444e962ccd2aec46d9c3.html

Gell-Mann.M.(1994).Zipf's Law, Benford's Law. Kasutamise küüpäev 18.aprill 2016.a., allikas http://www.cut-the-knot.org/do_you_know/zipfLaw.shtml

Lü, L., Zhang, Z.-K., & Zhou, T. (2013). Deviation of Zipf's and Heaps' Laws in human languages with limited dictionary sizes. *Scientific Reports*, *3*, 1082. doi:10.1038/srep01082